IDENTIFICATION

PRODUCT CODE: AC-S942A-MC

PRODUCT NAME: CNKXAA0 KXT-11 PROC DIAG

DATE CREATED: MARCH 1982

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: G. PASQUANTONIO

1.0 GENERAL INFORMATION

THIS DIAGNOSTIC IS A RAM-BASED VERSION OF "CIKXAA0" AND IS
INTENDED FOR FIELD SUPPORT OF THE KXT-11 SBC (M8063-AA).

1.1 ABSTRACT

THE KXT11-AA IS A T11 BASED SINGLE BOARD COMPUTER (SBC-11).
IT INCLUDES A T11 CPU, 2KW 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 2KW RAM TESTS.
4. LOCAL 1KW 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 1KW MACRO-ODT ROM.
KXT11-LP (H3275) DUAL LOOP-BACK CONNECTOR.
16KW (OR MORE) Q-BUS MEMORY (MSV11-D).
XXDP+ LOAD DEVICE AND MEDIA (RX01, RX02, OR TU58).
CONSOLE TERMINAL (VT52, VT100, ETC...).
Q-BUS EXERCISER (G5281) MODULE (OPTIONAL).

1.3 RELATED DOCUMENTS AND STANDARDS

KXT11-AA ENGINEERING SPEC.
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)
LOCAL RAM ADDRESS/SIZE	160010	2KW-4	
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 ABOUT 6 SECONDS PER PASS, 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) POWER-FAIL,
BEVNT, AND Q-BUS INTERRUPTS WILL ALSO BE TESTED HERE.

5.3 SEGMENT 3 -- LOCAL 2KW RAM TESTS.

VERIFY READ/WRITE ACCESS TO THE ON-BOARD 2KW RAM USING VARIOUS
ADDRESS AND DATA PATTERNS. IF A Q-BUS-EXERCISER MODULE IS
INSTALLED (AT 174020), TEST READ/WRITE DMA ACCESS AS WELL.

NOTE: MACRO-ODT UTILIZES A PORTION OF THE LOCAL RAM.
IF A BREAK-AND-PROCEED SEQUENCE IS SERVICED DURING THIS SEGMENT
ERRORS MAY OCCUR IF/WHEN RAM BACKGROUND DATA PATTERNS ARE
DESTROYED (BY ODT ITSELF).

5.4 SEGMENT 4 -- LOCAL 1KW 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). SEQ 0006

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:

NOTE THAT THE TABLE OF CONTENTS INCLUDES A SEQUENTIAL LIST OF ALL ERROR HALTS AND AN INDICATION OF THE NATURE OF THE ERROR ENCOUNTERED AT THAT LOCATION.

```
                        .TITLE  CNKXAA -- KXT-11 (FALCON) VERIFICATION TESTS (FIELD VERSION).
                        ;
                        ; REV B -- FIX RAM ADDRESS TEST.
                        ;          FIX SLU SPEED DIFFERENTIAL TEST.
                        ;          FIX PIO RESET TEST.
                        ;
                        ; PART 1 -- TEST THE T11 INSTRUCTION SET.
                        ; PART 2 -- TEST THE T11 TRAPS AND INTERRUPTS.
                        ; PART 3 -- TEST THE LOCAL 2K 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
                                .NLIST  MD,MC,CND,TOC
                                .LIST   ME
                        ;*****************************************************************************
000076                  PRGSIZ= <^H<$LSTAD>>+1  ; PROGRAM SIZE IN 1/8 K UNITS.
                        .SBTTL  SWITCH REGISTER OPTIONS
000176                  SWR=    176             ; THE STANDARD SOFT SWITCH REGISTER ADDRESS.
                        ;       SWR<15> 100000  HALT ON ERROR.
                        ;       SWR<14>
                        ;       SWR<13> 020000  INHIBIT ERROR MESSAGES.
                        ;       SWR<12> 010000  PRINT ERROR SUMMARY AT END-PASS.
                        ;       SWR<11>
                        ;       SWR<10>
                        ;       SWR<09> 001000  LOOP ON ERROR.
                        ;       SWR<08>
                        ;       SWR<02:00>   N  RUN SEGMENT N (1 THRU 7) ONLY.
                        ;*****************************************************************************
000000                  TN$=    0               ; INIT TEST NUMBER SEQUENCE.
                        ;
                        ; T11 CPU EQUATES.
                        ;
000006                  R6=     SP
000007                  R7=     PC
000004                  ERRVEC= 4               ; ILLEGAL INSTRUCTION VECTOR.
000010                  RESVEC= 10              ; RESERVED OPCODE VECTOR.
000014                  BPTVEC= 14              ; BREAKPOINT VECTOR...
000014                  TRTVEC= BPTVEC          ;...TRACE TRAP TOO.
000020                  IOTVEC= 20              ; IOT VECTOR.
000024                  PWRVEC= 24              ; POWER FAIL VECTOR.
000030                  EMTVEC= 30              ; EMT VECTOR.
000034                  TRPVEC= 34              ; TRAP VECTOR.
000340                  PRI7=   340
000300                  PRI6=   300
000240                  PRI5=   240
000200                  PRI4=   200
000140                  PRI3=   140
000100                  PRI2=   100
000040                  PRI1=   040
000000                  PRI0=   000
                        ;
000010                  RESRVD= 000010          ; RESERVED (UNDEFINED) OPCODE.
```

```
                        MFPT=   000007          ; MFPT OPCODE (NEW INSTRUCTION)...
                        KXT11=  4               ;...SHOULD RETURN #4 IN R0.
                        CLNZ=   CLN!CLZ
                        SEVC=   SEV!SEC
                        SENVC=  SEN!SEV!SEC
                        NOP1=   260
                        SKP1=   BR+1
                        SKP2=   BR+2
                        SKP3=   BR+3
104400                  CHECKCC= TRAP           ; CC BIT CHECK CALL.
104420                  BEGINTEST= TRAP+20
104421                  CHECKSWR= TRAP+21       ; CHECK FOR <^G>.
104422                  GETSWR= TRAP+22         ; GET NEW SWR.
104423                  ERROR=  TRAP+23         ; ERROR CALL.
                        ;
                        .SBTTL  KXT-11 (FALCON) DEFAULT CONFIGURATION
                        ;
172000                  START=  ODT             ; ON POWER-UP, START MACRO-ODT IN PROM...
172004                  RESTART= ODTRE          ;...TRAP TO "RESTART" ON TIME-OUT OR HALT...
000000                  EHALT=  HALT            ;...AND HALT ON PROGRAM ERROR.

170000                  ROMADR= 170000          ; LOCAL ROM SOCKETS STRAPPED...
002000                  ROMSIZ= 1024.           ;...TO RESPOND AT 30K TO 31K...
                                                ;...MACRO-ODT PROM MUST BE INSTALLED.
172000                  ODT=    172000          ; IT STARTS HERE...
172004                  ODTRE=  172004          ;...AND RESTARTS HERE.
170000                  ODTBRK= 170000          ; "BREAK" HALT IS SERVICED HERE...
000140                  ODTIN=  BHALT           ;...AND THE VECTOR SERVES AS A HANDY FLAG.

160010                  RAMADR= 160010          ; LOCAL 2K RAM STRAPPED...
003764                  RAMSIZ= 2048.-4.-8.     ;...TO RESPOND AT 28K(+4) TO 30K...
                                                ;...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).

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                  PIOM=   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
001034                  SPD7=   K7
001040                  DFPBK=  ADR+40          ; DEFAULT PBR BITS...
001042                  DFSPD=  ADR+42          ;...AND THE CORRESPONDING SPEED COUNTER.

001044                  RCSR=   ADR+44          ; DLART REGISTER POINTERS.
001046                  RBUF=   ADR+46
001050                  XCSR=   ADR+50
001052                  XBUF=   ADR+52
001054                  RVEC=   ADR+54
001056                  XVEC=   ADR+56

001060                  PPA=    ADR+60          ; PARALLEL I/O REGISTER POINTERS.
001062                  PPB=    ADR+62
001064                  PPC=    ADR+64
001066                  PPCW=   ADR+66
001070                  PPAV=   ADR+70
001072                  PPBV=   ADR+72

                                ;  \\
                                ;   >> REMAINDER OF 1ST K IS OUR STACK.
00377o                  STACK=  3776            ;  //
                                ;
                                ; NOW THE ENTIRE 2N. K OF RAM IS USED AS AN "ERROR LOG".
                                ; ALL ERRORS ARE LOGGED AS THEY OCCUR.
                                ;
004000                  ERRCNT= 4000            ; THE 1ST WORD IS A TOTAL ERROR COUNT...
004002                  TPASS=  4002            ;...THEN A "TOTAL" PASS COUNT...
004004                  GPASS=  4004            ;...AND A "CONSECUTIVE-GOOD" PASS COUNT.
004006                  EPCTBL= 4006            ; THE REST FORMS A TABLE OF 2 WORD ENTRIES...
```

```
  194                                                ;...SHOWING ERROR PC (WD1) AND THE NUMBER...
  195                                                ;...OF OCCURRANCES OF THAT ERROR (WD2).
  196          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.
010134  010132                  .PRINT .-2                      ; *** DEBUG, 100000 ***
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  035336  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  025370                MOV     INSEG,R0        ; SINGLE SEGMENT SELECTED ??
010274  001420                        BEQ     DOALL           ;...IF NOT, RUN 'EM ALL...
010276  010007                        MOV     R0,PC           ;...ELSE, GO THERE.

010300  005015  045516  040530  HELLO:  .ASCIZ  <15><12>'NKXAA0 KXT-11 DIAGNOSTIC'<15><12>
        010336                  .EVEN
                                .SBTTL
```

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

010336                                  DOALL:
010336  012767  036202  025634        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$
```

```
   357  010506  003002                    BGT     2$
   358  010510                      1$:
   (2)  010510  104423  010344            ERROR,  T1              ;; *** BRANCH FAILURE OR CC NOT 1010 ***
   359  010514                      2$:
   360  010514  000270              CBIT:  SEN                     ; SET N...
   361  010516  000262                     SEV                     ;...V...
   362  010520  000261                     SEC                     ;...AND C, NZVC=1011
   363  010522  001406                     BEQ     1$
   364  010524  100005                     BPL     1$
   365  010526  102004                     BVC     1$
   366  010530  002403                     BLT     1$
   367  010532  003402                     BLE     1$
   368  010534  101001                     BHI     1$
   369  010536  002002                     BGE     2$
   370  010540                      1$:
   (2)  010540  104423  010344            ERROR,  T1              ;; *** BRANCH FAILURE OR CC NOT 1011 ***
   371  010544                      2$:
   372  010544  000270              ZBIT:  SEN                     ; SET N...
   373  010546  000262                     SEV                     ;...V...
   374  010550  000261                     SEC                     ;...C...
   375  010552  000264                     SEZ                     ;...AND Z, NZVC = 1111.
   376  010554  001007                     BNE     1$
   377  010556  100006                     BPL     1$
   378  010560  102005                     BVC     1$
   379  010562  103004                     BCC     1$
   380  010564  002403                     BLT     1$
   381  010566  003002                     BGT     1$
   382  010570  101001                     BHI     1$
   383  010572  001402                     BEQ     2$
   384  010574                      1$:
   (2)  010574  104423  010344            ERROR,  T1              ;; *** BRANCH FAILURE OR CC NOT 1111 ***
   385  010600                      2$:
   386
   387                                    ;*******************************************************************************
   (2)                                    .SBTTL      T2 -- SET AND CLEAR ALL CONDITION CODES
   (2)                                    ;*******************************************************************************
   (2)  010600  104420  000002       T2:  BEGINTEST, 2            ;; CHECK SWITCH REGISTER OPTIONS.
   388  010604  000277              ALLCC: SCC                     ; NZVC = 1111
   389  010606  104417                     CHECKCC+17              ; CHECK ALL BITS SET.
   390  010610  104423  010600            ERROR,  T2              ;; *** CC BITS WRONG AFTER SCC ***
   391
   392  010614  000257                     CCC                     ; NZVC = 0000
   393  010616  104400                     CHECKCC+0
   394  010620  104423  010600            ERROR,  T2              ;; *** CC BITS WRONG AFTER CCC ***
   395
   396                                    ;*******************************************************************************
   (2)                                    .SBTTL      T3 -- TEST REGISTER SELECTION
   (2)                                    ;*******************************************************************************
   (2)  010624  104420  000003       T3:  BEGINTEST, 3            ;; CHECK SWITCH REGISTER OPTIONS.
   397  010630  012700  000001     REGS:  MOV     #1,R0            ; LOAD UP R0 THRU R4.
   398  010634  012701  000004            MOV     #4,R1
   399  010640  012702  000020            MOV     #20,R2
   400  010644  012703  000100            MOV     #100,R3
   401  010650  012704  000400            MOV     #400,R4
   402  010654  012705  001000            MOV     #1000,R5
   403  010660  060600                     ADD     R6,R0           ; R0 + #STACK...
```

```
                        ADD     R5,R0           ;...+ 1000...
                        ADD     R4,R0           ;...+ 400...
                        ADD     R3,R0           ;...+ 100...
                        ADD     R2,R0           ;...+ 20...
                        ADD     R1,R0           ;...+ 4...
                        CMP     R0,#STACK+1525  ;...SHOULD = THIS.
                        BEQ     1$              ; BR IF OK.
                        ERROR,  T3              ;; *** REGISTER SELECTION FAILURE ***
                1$:

                ;*******************************************************************
                .SBTTL      T4 -- JMP INSTRUCTION -- MODE 1
                ;*******************************************************************
                T4:     BEGINTEST, 4            ;; CHECK SWITCH REGISTER OPTIONS.
                JMP1:   MOV     #1$,R0          ; TEST JUMP INSTRUCTION MODE 1
                        SCC
                        JMP     (R0)
                        ERROR,  T4              ;; *** MODE 1 JUMP FAILED ***
                1$:     CHECKCC+17
                        ERROR,  T4              ;; *** CC BITS ALTERED ON JMP ***
                        CMP     R0,#1$
                        BEQ     2$              ; CONTINUE IF R0 IS OK
                        ERROR,  T4              ;; *** R0 INCORRECT AFTER JUMP ***
                2$:

                ;*******************************************************************
                .SBTTL      T5 -- JMP INSTRUCTION -- MODES 2 AND 3
                ;*******************************************************************
                T5:     BEGINTEST, 5            ;; CHECK SWITCH REGISTER OPTIONS.
                JMP2:   MOV     #1$,R0          ; TEST JUMP INSTRUCTION MODE 2
                        SCC
                        JMP     (R0)+
                        ERROR,  T5              ;; *** MODE 2 JUMP FAILED ***
                1$:     CHECKCC+17
                        ERROR,  T5              ;; *** CC BITS ALTERED AFTER JMP ***
                        CMP     R0,#1$+2        ; IS THERE AUTO INC.?
                        BEQ     JMP3
                        ERROR,  T5              ;; *** MODE 2 FAILED ON JUMP ***

                JMP3:   MOV     #3$,TEMP        ; TEST JUMP INSTRUCTION MODE 3
                        MOV     #4$,TEMP+2      ;
                        MOV     #TEMP,R0
                        SCC
                        JMP     @(R0)+          ;
                        ERROR,  T5              ;; *** MODE 3 JUMP FAILED ***
                3$:     CMP     @(R0),4$        ; IS THERE AUTO INC.?
                        BEQ     4$
                        ERROR,  T5              ;; *** MODE 3 JUMP FAILED ***
                4$:

                ;*******************************************************************
                .SBTTL      T6 -- JMP INSTRUCTION -- MODES 4 AND 5
                ;*******************************************************************
                T6:     BEGINTEST, 6            ;; CHECK SWITCH REGISTER OPTIONS.
                JMP4:   MOV     #3$,R0          ; TEST JUMP INSTRUCTION MODE 4
                        SCC
```

```
  451  011064  000140                        JMP     -(R0)
  452  011066  104423  011052                ERROR,  T6              ;; *** DIDN'T JUMP AT ALL ***
  453  011072  000402                        BR      4$              ; JUMP SHOULD LAND HERE
  454  011074                          3$:
  (2)  011074  104423  011052                ERROR,  T6              ;; *** MODE 4 JUMP FAILED. ***
  455  011100  022700  011072          4$:   CMP     #3$-2,R0        ; CHECK R0
  456  011104  001402                        BEQ     JMP5
  457  011106  104423  011052                ERROR,  T6              ;; *** R0 INCORRECT AFTER JUMP ***
  458
  459  011112  012767  011140  167670  JMP5: MOV     #3$,TEMP1       ; TEST JUMP INSTRUCTION MODE 5
  460  011120  012700  001010                MOV     #TEMP1,R0
  461  011124  012767  011144  167654        MOV     #4$,TEMP1-2     ;
  462  011132  000150                        JMP     @-(R0)
  463  011134  104423  011052                ERROR,  T6              ;; *** MODE 5 JUMP DIDN'T ***
  464  011140                          3$:
  (2)  011140  104423  011052                ERROR,  T6              ;; *** AUTO-DECR FAILED ON MODE 5 JUMP ***
  465  011144  022700  001006          4$:   CMP     #TEMP1-2,R0     ; CHECK R0
  466  011150  001402                        BEQ     5$
  467  011152  104423  011052                ERROR,  T6              ;; *** R0 INCORRECT AFTER JMP5 ***
  468  011156                          5$:
  469
  470                                  ;*******************************************************************
  (2)                                  .SBTTL      T7 -- JMP INSTRUCTION -- MODES 6 AND 7
  (2)                                  ;*******************************************************************
  (2)  011156  104420  000007          T7:     BEGINTEST, 7            ;; CHECK SWITCH REGISTER OPTIONS.
  471  011162  012703  011204          JMP6:   MOV     #1$+6,R3
  472  011166  000163  177772                  JMP     -6(R3)
  473  011172  104423  011156                  ERROR,  T7              ;; *** MODE 6 JUMP FAILED ***
  474  011176  020327  011204          1$:     CMP     R3,#1$+6        ; CHECK R3
  475  011202  001402                          BEQ     2$
  476  011204  104423  011156                  ERROR,  T7              ;; *** R3 WRONG AFTER JMP6 ***
  477  011210  000167  000004          2$:     JMP     3$-.-4(PC)      ; TEST JUMP INSTRUCTION MODE 6
  478  011214  104423  011156                  ERROR,  T7              ;; *** JUMP FAILED ***
  479
  480  011220  012703  011234          3$:     MOV     #JMP7,R3        ; JUMP SHOULD LAND HERE
  481  011224  000163  000000                  JMP     0(R3)
  482  011230  104423  011156                  ERROR,  T7              ;; *** JUMP FAILED ***
  483
  484  011234  012703  001006          JMP7:   MOV     #TEMP,R3
  485  011240  012713  011254                  MOV     #1$,(R3)
  486  011244  000173  000000                  JMP     @(R3)
  487  011250  104423  011156                  ERROR,  T7              ;; *** MODE 7 JUMP FAILED ***
  488  011254  012713  011274          1$:     MOV     #3$,(R3)        ; TEST JUMP INSTRUCTION MODE 7
  489  011260  012700  001002                  MOV     #TEMP-4,R0
  490  011264  000170  000004                  JMP     @4(R0)
  491  011270  104423  011156                  ERROR,  T7              ;; *** JUMP FAILED ***
  492  011274  012767  011316  167504  3$:     MOV     #4$,TEMP        ; CONTINUE
  493  011302  012700  001006                  MOV     #TEMP,R0
  494  011306  000170  000000                  JMP     @0(R0)
  495  011312  104423  011156                  ERROR,  T7              ;; *** MODE 7 JUMP FAILED. ***
  496  011316                          4$:
  497
  498                                  ;*******************************************************************
  (2)                                  .SBTTL      T10 -- JSR AND RTS INSTRUCTIONS
  (2)                                  ;*******************************************************************
  (2)  011316  104420  000010          T10:    BEGINTEST, 10           ;; CHECK SWITCH REGISTER OPTIONS.
```

```
   499  011322  012706  003776          JSRTST: MOV     #STACK,SP       ; SET UP STACK POINTER.
   500  011326  000277                          SCC
   501  011330  004767  000004                  JSR     PC,2$
   502  011334                          1$:
   (2)  011334  104423  011316                  ERROR,  T10             ;; *** JSR INSTRUCTION FAILED ***
   503  011340  104417                  2$:     CHECKCC+17
   504  011342  104423  011316                  ERROR,  T10             ;; *** CC BITS CHANGED ON JSR ***
   505  011346  022706  003774          4$:     CMP     #STACK-2,SP
   506  011352  001402                          BEQ     5$              ; BR IF STACK IS RIGHT.
   507  011354  104423  011316                  ERROR,  T10             ;; *** STACK WRONG AFTER JSR ***
   508  011360  022716  011334          5$:     CMP     #1$,(SP)
   509  011364  001402                          BEQ     6$              ; BR IF STACKED PC IS RIGHT.
   510  011366  104423  011316                  ERROR,  T10             ;; *** SP DID NOT HAVE CORRECT RETURN ADDRESS ***
   511
   512  011372  012716  011404          6$:     MOV     #7$,(SP)        ; CHANGE RETURN PC.
   513  011376  000207                          RTS     PC
   514  011400  104423  011316                  ERROR,  T10             ;; *** RTS INSTRUCTION FAILED ***
   515  011404  022706  003776          7$:     CMP     #STACK,SP
   516  011410  001402                          BEQ     8$              ; BR IF STACK PROPERLY RESTORED.
   517  011412  104423  011316                  ERROR,  T10             ;; *** SP WAS NOT RESTORED BY RTS INSTRUCTION ***
   518  011416                          8$:
   519                                  ;
   520                                  .SBTTL
   521                                  .SBTTL  DESTINATION MODE 0
   522                                  ;
   523                                  ;*******************************************************************
   (2)                                  .SBTTL      T11 -- TSTB, CLRB, AND MOVB
   (2)                                  ;*******************************************************************
   (2)  011416  104420  000011          T11:    BEGINTEST, 11           ;; CHECK SWITCH REGISTER OPTIONS.
   524  011422  012706  003776          TSTB0:  MOV     #STACK,SP       ; INIT THE STACK.
   525  011426  000277                          SCC
   526  011430  105000                          CLRB    R0              ; CLEAR THE REGISTER
   527  011432  104404                          CHECKCC+4               ; CHECK FOR CC = 4
   528  011434  104423  011416                  ERROR,  T11             ;; *** CLRB FAILED ***
   529  011440  105700                          TSTB    R0              ; CHECK IT
   530  011442  104404                          CHECKCC+4               ; CHECK FOR CC = 4
   531  011444  104423  011416                  ERROR,  T11             ;; *** TSTB FAILED. ***
   532  011450  112701  000377                  MOVB    #377,R1         ; LOAD THE REGISTER
   533  011454  104410                          CHECKCC+10              ; CHECK FOR CC = 10
   534  011456  104423  011416                  ERROR,  T11             ;; *** INCORRECT CC BITS ***
   535  011462  105701                          TSTB    R1              ; CHECK IT
   536  011464  104410                          CHECKCC+10              ; CHECK FOR CC = 10
   537  011466  104423  011416                  ERROR,  T11             ;; *** INCORRECT CC BITS ***
   538
   539                                  ;*******************************************************************
   (2)                                  .SBTTL      T12 -- CMPB AND BISB
   (2)                                  ;*******************************************************************
   (2)  011472  104420  000012          T12:    BEGINTEST, 12           ;; CHECK SWITCH REGISTER OPTIONS.
   540  011476  000277                  CMPB0:  SCC
   541  011500  152702  000377                  BISB    #377,R2         ; LOAD REGISTER
   542  011504  104411                          CHECKCC+11              ; CHECK FOR CC = 11
   543  011506  104423  011472                  ERROR,  T12             ;; *** INCORRECT CC BITS ***
   544  011512  122702  000377                  CMPB    #377,R2         ; CHECK COMPARE
   545  011516  001402                          BEQ     2$              ; CONTINUE IF OK
   546  011520  104423  011472                  ERROR,  T12             ;; *** BISB OR CMPB INSTRUCTION FAILED ***
   547  011524  112700  000077          2$:     MOVB    #77,R0
```

```
548  011530  120002                        CMPB    R0,R2           ; CHECK IT AGAIN
549  011532  100002                        BPL     3$              ; CONTINUE IF OK
550  011534  104423  011472                ERROR,  T12             ;; *** CMPB INSTRUCTION FAILED (WRONG CC) ***
551  011540  120200                3$:     CMPB    R2,R0           ; ONCE MORE
552  011542  100402                        BMI     4$              ; CONTINUE IF OK
553  011544  104423  011472                ERROR,  T12             ;; *** CMPB INSTRUCTION FAILED (WRONG CC) ***
554  011550  112702  000377        4$:     MOVB    #377,R2         ; LOAD REGISTER, SIGN EXTEND
555  011554  122702  000377                CMPB    #377,R2         ; CHECK IF BYTE INSTRUCTION
556  011560  001402                        BEQ     5$              ; CONTINUE IF OK
557  011562  104423  011472                ERROR,  T12             ;; *** CMPB BECAME CMP INSTRUCTION ***
558  011566  112702  000377        5$:     MOVB    #377,R2         ; LOAD REGISTER, SIGN EXTEND
559  011572  120227  000377                CMPB    R2,#377         ; CHECK IF BYTE INSTRUCTION
560  011576  001402                        BEQ     6$              ; CONTINUE IF OK
561  011600  104423  011472                ERROR,  T12             ;; *** MOVB OR CMPB FAILED ***
562  011604                         6$:
563
564                                 ;*******************************************************************************
(2)                                 .SBTTL      T13 -- BICB AND BITB
(2)                                 ;*******************************************************************************
(2)  011604  104420  000013        T13:    BEGINTEST, 13           ;; CHECK SWITCH REGISTER OPTIONS.
565  011610  112703  000377        BICB0:  MOVB    #377,R3         ; LOAD REGISTER
566  011614  112700  000252                MOVB    #252,R0         ; PLACE #252 IN R0
567  011620  000277                        SCC
568  011622  140003                        BICB    R0,R3           ; CLEAR EVERY OTHER BIT
569  011624  104401                        CHECKCC+1               ; CHECK FOR CC = 1
570  011626  104423  011604                ERROR,  T13             ;; *** INCORRECT CC BITS ***
571  011632  130003                        BITB    R0,R3           ; CHECK IT
572  011634  001402                        BEQ     4$              ; CONTINUE IF OK
573  011636  104423  011604                ERROR,  T13             ;; *** BICB OR BITB INSTRUCTION FAILED ***
574  011642  132703  000125        4$:     BITB    #125,R3         ; CHECK IT
575  011646  104401                        CHECKCC+1               ; CHECK FOR CC = 1
576  011650  104423  011604                ERROR,  T13             ;; *** INCORRECT CC BITS ***
577  011654  150003                        BISB    R0,R3           ; SET THE BITS THAT WERE CLEARED
578  011656  100402                        BMI     6$
579  011660  104423  011604                ERROR,  T13             ;; *** BISB INSTRUCTION FAILED ***
580  011664  142703  000177        6$:     BICB    #177,R3         ; CLEAR ALL THE BITS EXCEPT FOR SIGN
581  011670  104411                        CHECKCC+11              ; CHECK FOR CC = 11
582  011672  104423  011604                ERROR,  T13             ;; *** INCORRECT CC BITS ***
583  011676  132703  000377                BITB    #377,R3         ; CHECK IT
584  011702  104411                        CHECKCC+11              ; CHECK FOR CC = 11
585  011704  104423  011604                ERROR,  T13             ;; *** INCORRECT CC BITS ***
586
587                                 ;*******************************************************************************
(2)                                 .SBTTL      T14 -- INCB AND DECB
(2)                                 ;*******************************************************************************
(2)  011710  104420  000014        T14:    BEGINTEST, 14           ;; CHECK SWITCH REGISTER OPTIONS.
588  011714  112704  000177        INCB0:  MOVB    #177,R4         ; R4 = 177
589  011720  000261                        SEC
590  011722  105204                        INCB    R4              ; ADD ONES INTO REG. 4
591  011724  104413                        CHECKCC+13              ; CHECK FOR CC = 13
592  011726  104423  011710                ERROR,  T14             ;; *** INCORRECT CC BITS ***
593  011732  112704  000376                MOVB    #376,R4
594  011736  105204                        INCB    R4
595  011740  104411                        CHECKCC+11              ; CHECK FOR CC = 11
596  011742  104423  011710                ERROR,  T14             ;; *** INCORRECT CC BITS ***
597  011746  105204                        INCB    R4
```

```
  598  011750  104405                        CHECKCC+5               ; CHECK FOR CC = 5
  599  011752  104423  011710                ERROR,  T14             ;; *** INCORRECT CC BITS ***
  600  011756  105204                        INCB    R4
  601  011760  104401                        CHECKCC+1               ; CHECK FOR CC = 1
  602  011762  104423  011710                ERROR,  T14             ;; *** INCORRECT CC BITS ***
  603  011766  122704  000001                CMPB    #1,R4           ; CHECK IT
  604  011772  001402                        BEQ     2$              ; CONTINUE IF OK
  605  011774  104423  011710                ERROR,  T14             ;; *** INCB INSTRUCTION FAILED ***
  606  012000  000261                2$:     SEC
  607  012002  105304                        DECB    R4              ; SUBTRACT ONES FROM REG. 4
  608  012004  104405                        CHECKCC+5               ; CHECK FOR CC = 5
  609  012006  104423  011710                ERROR,  T14             ;; *** INCORRECT CC BITS ***
  610  012012  105304                        DECB    R4
  611  012014  104411                        CHECKCC+11              ; CHECK FOR CC = 11
  612  012016  104423  011710                ERROR,  T14             ;; *** INCORRECT CC BITS ***
  613  012022  012704  000200                MOV     #200,R4
  614  012026  105304                        DECB    R4
  615  012030  104403                        CHECKCC+3               ; CHECK FOR CC = 3
  616  012032  104423  011710                ERROR,  T14             ;; *** INCORRECT CC BITS ***
  617  012036  105304                        DECB    R4
  618  012040  104401                        CHECKCC+1               ; CHECK FOR CC = 1
  619  012042  104423  011710                ERROR,  T14             ;; *** INCORRECT CC BITS ***
  620
  621                                ;*******************************************************************
  (2)                                .SBTTL      T15 -- COMB AND NEGB
  (2)                                ;*******************************************************************
  (2)  012046  104420  000015        T15:    BEGINTEST, 15           ;; CHECK SWITCH REGISTER OPTIONS.
  622  012052  112703  000252        COMB0:  MOVB    #252,R3         ; LOAD EVERY OTHER BIT
  623  012056  000277                        SCC
  624  012060  105103                        COMB    R3              ; 1'S COMPLEMENT
  625  012062  104401                        CHECKCC+1               ; CHECK FOR CC = 1
  626  012064  104423  012046                ERROR,  T15             ;; *** INCORRECT CC BITS ***
  627  012070  122703  000125                CMPB    #125,R3         ; CHECK IT
  628  012074  001402                        BEQ     2$              ; CONTINUE IF OK
  629  012076  104423  012046                ERROR,  T15             ;; *** COMB INSTRUCTION FAILED ***
  630  012102  000277                2$:     SCC
  631  012104  105103                        COMB    R3              ; COMPLEMENT BACK
  632  012106  104411                        CHECKCC+11              ; CHECK FOR CC = 11
  633  012110  104423  012046                ERROR,  T15             ;; *** INCORRECT CC BITS ***
  634  012114  122703  000252                CMPB    #252,R3         ; CHECK IT
  635  012120  001402                        BEQ     3$              ; CONTINUE IF OK
  636  012122  104423  012046                ERROR,  T15             ;; *** COMB INSTRUCTION FAILED ***
  637  012126  012703  000377        3$:     MOV     #377,R3
  638  012132  000277                        SCC
  639  012134  105103                        COMB    R3
  640  012136  104405                        CHECKCC+5               ; CHECK FOR CC = 5
  641  012140  104423  012046                ERROR,  T15             ;; *** INCORRECT CC BITS ***
  642
  643  012144  112700  000001        NEGB0:  MOVB    #1,R0           ; LOAD THE REGISTER
  644  012150  105400                        NEGB    R0              ; 2'S COMPLEMENT
  645  012152  104411                        CHECKCC+11              ; CHECK FOR CC = 11
  646  012154  104423  012144                ERROR,  NEGB0           ;; *** INCORRECT CC BITS ***
  647  012160  122700  000377                CMPB    #377,R0         ; CHECK IT
  648  012164  001402                        BEQ     2$              ; CONTINUE IF OK
  649  012166  104423  012144                ERROR,  NEGB0           ;; *** NEGB INSTRUCTION FAILED ***
  650  012172  012700  000200        2$:     MOV     #200,R0
```

```
   651  012176  105400                        NEGB    R0              ; 2'S COMPLEMENT
   652  012200  104413                        CHECKCC+13              ; CHECK FOR CC = 13
   653  012202  104423  012144                ERROR,  NEGB0           ;; *** INCORRECT CC BITS ***
   654  012206  122700  000200                CMPB    #200,R0         ; CHECK IT
   655  012212  001402                        BEQ     3$              ; CONTINUE IF OK
   656  012214  104423  012144                ERROR,  NEGB0           ;; *** NEGB FAILED ***
   657  012220                          3$:
   658
   659                                  ;*******************************************************************************
   (2)                                  .SBTTL      T16 -- ROLB AND RORB
   (2)                                  ;*******************************************************************************
   (2)  012220  104420  000016          T16:    BEGINTEST, 16           ;; CHECK SWITCH REGISTER OPTIONS.
   660  012224  112701  000040          ROLB0:  MOVB    #40,R1          ; LOAD REGISTER
   661  012230  000257                        CCC                     ; CLEAR FLAGS
   662  012232  106101                        ROLB    R1              ; SHIFT
   663  012234  106101                        ROLB    R1              ;
   664  012236  104412                        CHECKCC+12              ; CHECK FOR CC = 12
   665  012240  104423  012220                ERROR,  T16             ;; *** INCORRECT CC BITS ***
   666  012244  122701  000200                CMPB    #200,R1         ; CHECK IT
   667  012250  001402                        BEQ     1$              ; CONTINUE IF OK
   668  012252  104423  012220                ERROR,  T16             ;; *** ROLB INSTRUCTION FAILED ***
   669  012256  106101                  1$:   ROLB    R1              ; SHIFT
   670  012260  104407                        CHECKCC+7               ; CHECK FOR CC = 7
   671  012262  104423  012220                ERROR,  T16             ;; *** INCORRECT CC BITS ***
   672  012266  106101                        ROLB    R1              ; SHIFT
   673  012270  122701  000001                CMPB    #1,R1           ; CHECK IT
   674  012274  001402                        BEQ     2$              ; CONTINUE IF OK
   675  012276  104423  012220                ERROR,  T16             ;; *** ROLB FAILED. ***
   676  012302                          2$:
   677  012302  112702  000004          RORB0:  MOVB    #4,R2           ; LOAD REGISTER
   678  012306  000257                        CCC                     ; CLEAR FLAGS
   679  012310  106002                        RORB    R2              ; SHIFT
   680  012312  106002                        RORB    R2              ;
   681  012314  122702  000001                CMPB    #1,R2           ; CHECK IT
   682  012320  001402                        BEQ     1$              ; CONTINUE IF OK
   683  012322  104423  012302                ERROR,  RORB0           ;; *** RORB INSTRUCTION FAILED ***
   684  012326  106002                  1$:   RORB    R2              ; SHIFT
   685  012330  104407                        CHECKCC+7               ; CHECK FOR CC = 7
   686  012332  104423  012302                ERROR,  RORB0           ;; *** INCORRECT CC BITS ***
   687  012336  106002                        RORB    R2              ; SHIFT
   688  012340  104412                        CHECKCC+12              ; CHECK FOR CC = 12
   689  012342  104423  012302                ERROR,  RORB0           ;; *** INCORRECT CC BITS ***
   690  012346  122702  000200                CMPB    #200,R2         ; CHECK IT
   691  012352  001402                        BEQ     2$              ; CONTINUE IF OK
   692  012354  104423  012302                ERROR,  RORB0           ;; *** RORB FAILED ***
   693  012360                          2$:
   694
   695                                  ;*******************************************************************************
   (2)                                  .SBTTL      T17 -- ASLB AND ASRB
   (2)                                  ;*******************************************************************************
   (2)  012360  104420  000017          T17:    BEGINTEST, 17           ;; CHECK SWITCH REGISTER OPTIONS.
   696  012364  112703  000040          ASLB0:  MOVB    #40,R3          ; LOAD REGISTER
   697  012370  000257                        CCC                     ; CLEAR FLAGS
   698  012372  106303                        ASLB    R3              ; SHIFT
   699  012374  106303                        ASLB    R3              ;
   700  012376  104412                        CHECKCC+12              ; CHECK FOR CC = 12
```

```
   701  012400  104423  012360                 ERROR,  T17          ;; *** INCORRECT CC BITS ***
   702  012404  122703  000200                 CMPB    #200,R3      ; CHECK IT
   703  012410  001402                          BEQ     4$           ; CONTINUE IF OK
   704  012412  104423  012360                 ERROR,  T17          ;; *** ASLB INSTRUCTION FAILED ***
   705  012416  106303                 4$:      ASLB    R3           ; SHIFT
   706  012420  104407                          CHECKCC+7            ; CHECK FOR CC = 7
   707  012422  104423  012360                 ERROR,  T17          ;; *** INCORRECT CC BITS ***
   708  012426  106303                          ASLB    R3           ; SHIFT
   709  012430  104404                          CHECKCC+4            ; CHECK FOR CC = 4
   710  012432  104423  012360                 ERROR,  T17          ;; *** INCORRECT CC BITS ***
   711
   712  012436  112704  000004         ASRB0:   MOVB    #4,R4        ; LOAD REGISTER
   713  012442  000257                          CCC                  ; CLEAR FLAGS
   714  012444  106204                          ASRB    R4           ; SHIFT
   715  012446  106204                          ASRB    R4           ;
   716  012450  122704  000001                 CMPB    #1,R4        ; CHECK IT
   717  012454  001402                          BEQ     2$           ; CONTINUE IF OK
   718  012456  104423  012436                 ERROR,  ASRB0        ;; *** ASRB INSTRUCTION FAILED ***
   719  012462  106204                 2$:      ASRB    R4           ; SHIFT
   720  012464  104407                          CHECKCC+7            ; CHECK FOR CC = 7
   721  012466  104423  012436                 ERROR,  ASRB0        ;; *** INCORRECT CC BITS ***
   722  012472  106204                          ASRB    R4           ; SHIFT
   723  012474  104404                          CHECKCC+4            ; CHECK FOR CC = 4
   724  012476  104423  012436                 ERROR,  ASRB0        ;; *** INCORRECT CC BITS ***
   725  012502  112703  000202                 MOVB    #202,R3      ; LOAD REGISTER
   726  012506  106203                          ASRB    R3           ; SHIFT
   727  012510  106203                          ASRB    R3           ;
   728  012512  104411                          CHECKCC+11           ; CHECK FOR CC = 11
   729  012514  104423  012436                 ERROR,  ASRB0        ;; *** INCORRECT CC BITS ***
   730  012520  122703  000340                 CMPB    #340,R3      ; CHECK IT
   731  012524  001402                          BEQ     3$           ; CONTINUE IF OK
   732  012526  104423  012436                 ERROR,  ASRB0        ;; *** ASRB FAILED ***
   733  012532                          3$:
   734
   735                                  ;*******************************************************************
   (2)                                  .SBTTL      T20 -- ADCB AND SBCB
   (2)                                  ;*******************************************************************
   (2)  012532  104420  000020          T20:     BEGINTEST, 20        ;; CHECK SWITCH REGISTER OPTIONS.
   736  012536  105000                  ADCB0:   CLRB    R0           ; CLEAR THE REGISTER
   737  012540  000257                          CCC                  ; CLEAR FLAGS
   738  012542  105500                          ADCB    R0           ; ADD C BIT = 0
   739  012544  104404                          CHECKCC+4            ; CHECK FOR CC = 4
   740  012546  104423  012532                 ERROR,  T20          ;; *** INCORRECT CC BITS ***
   741  012552  000261                          SEC                  ; C=1
   742  012554  105500                          ADCB    R0           ; ADD C BIT=1
   743  012556  000261                          SEC                  ; C=1
   744  012560  105500                          ADCB    R0           ; AGAIN
   745  012562  104400                          CHECKCC+0            ; CHECK FOR CC = 0
   746  012564  104423  012532                 ERROR,  T20          ;; *** INCORRECT CC BITS ***
   747  012570  122700  000002                 CMPB    #2,R0        ; CHECK IT
   748  012574  001402                          BEQ     4$           ; CONTINUE IF OK
   749  012576  104423  012532                 ERROR,  T20          ;; *** ADCB INSTRUCTION FAILED ***
   750  012602  112700  000177         4$:      MOVB    #177,R0      ; LOAD LARGEST POSITVE NUMBER
   751  012606  000261                          SEC                  ; C=1
   752  012610  105500                          ADCB    R0           ; ADD C BIT=1
   753  012612  104412                          CHECKCC+12           ; CHECK FOR CC = 12
```

```
                                        ERROR,  T20             ;; *** INCORRECT CC BITS ***
012614  104423  012532                  CMPB    #200,R0         ; CHECK IT
012620  122700  000200                  BEQ     6$              ; CONTINUE IF OK
012624  001402                          ERROR,  T20             ;; *** ADCB INSTRUCTION FAILED ***
012626  104423  012532                  MOVB    #377,R0         ; LOAD -1
012632  112700  000377          6$:     SEC                     ; C=1
012636  000261                          ADCB    R0              ; ADD C BIT=1
012640  105500                          CHECKCC+5               ; CHECK FOR CC = 5
012642  104405                          ERROR,  T20             ;; *** INCORRECT CC BITS ***
012644  104423  012532
                                        MOVB    #3,R1           ; LOAD REGISTER
012650  112701  000003          SBCB0:  CCC                     ; CLEAR FLAGS
012654  000257                          SBCB    R1              ; SUBTRACT C BIT=0
012656  105601                          CHECKCC+0               ; CHECK FOR CC = 0
012660  104400                          ERROR,  SBCB0           ;; *** INCORRECT CC BITS ***
012662  104423  012650                  CMPB    #3,R1           ; CHECK IT
012666  122701  000003                  BEQ     2$              ; CONTINUE IF OK
012672  001402                          ERROR,  SBCB0           ;; *** SBCB INSTRUCTION FAILED ***
012674  104423  012650          2$:     SEC                     ; C=1
012700  000261                          SBCB    R1              ; SUBTRACT C BIT=1
012702  105601                          SEC                     ; C=1
012704  000261                          SBCB    R1              ;
012706  105601                          CHECKCC+0               ; CHECK FOR CC = 0
012710  104400                          ERROR,  SBCB0           ;; *** INCORRECT CC BITS ***
012712  104423  012650                  CMPB    #1,R1           ; CHECK IT
012716  122701  000001                  BEQ     3$              ; CONTINUE IF OK
012722  001402                          ERROR,  SBCB0           ;; *** SBCB INSTRUCTION FAILED ***
012724  104423  012650          3$:     SEC                     ; C=1
012730  000261                          SBCB    R1              ; SUBTRACT C BIT=1
012732  105601                          CHECKCC+4               ; CHECK FOR CC = 4
012734  104404                          ERROR,  SBCB0           ;; *** INCORRECT CC BITS ***
012736  104423  012650                  SEC                     ; C=1
012742  000261                          SBCB    R1              ; SUBTRACT C BIT = 1
012744  105601                          CHECKCC+11              ; CHECK FOR CC = 11
012746  104411                          ERROR,  SBCB0           ;; *** INCORRECT CC BITS ***
012750  104423  012650                  CMPB    #377,R1         ; CHECK IT
012754  122701  000377                  BEQ     4$              ; CONTINUE IF OK
012760  001402                          ERROR,  SBCB0           ;; *** SBCB INSTRUCTION FAILED ***
012762  104423  012650          4$:     MOVB    #200,R1         ; LOAD R1
012766  112701  000200                  SEC                     ; C=1
012772  000261                          SBCB    R1              ; SUBTRACT C BIT = 1
012774  105601                          CHECKCC+2               ; CHECK FOR CC = 2
012776  104402                          ERROR,  SBCB0           ;; *** INCORRECT CC BITS ***
013000  104423  012650

;*******************************************************************************
.SBTTL      T21 -- TST, CLR, AND MOV
;*******************************************************************************
013004  104420  000021          T21:    BEGINTEST, 21           ;; CHECK SWITCH REGISTER OPTIONS.
013010  000277                  MOV0:   SCC
013012  005000                          CLR     R0              ; CLEAR THE REGISTER
013014  104404                          CHECKCC+4               ; CHECK FOR CC = 4
013016  104423  013004                  ERROR,  T21             ;; *** INCORRECT CC BITS ***
013022  005700                          TST     R0              ; CHECK IT
013024  104404                          CHECKCC+4               ; CHECK FOR CC = 4
013026  104423  013004                  ERROR,  T21             ;; *** INCORRECT CC BITS ***
013032  012704  177777                  MOV     #177777,R4      ; LOAD THE REGISTER
```

```
  807  013036  010401                        MOV     R4,R1
  808  013040  104410                        CHECKCC+10              ; CHECK FOR CC = 10
  809  013042  104423  013004                ERROR,  T21             ;; *** INCORRECT CC BITS ***
  810  013046  005701                        TST     R1              ; CHECK IT
  811  013050  104410                        CHECKCC+10              ; CHECK FOR CC = 10
  812  013052  104423  013004                ERROR,  T21             ;; *** INCORRECT CC BITS ***
  813  013056  020401                        CMP     R4,R1           ; CHECK R1 TO CONTAIN PROPER DATA
  814  013060  001402                        BEQ     2$
  815  013062  104423  013004                ERROR,  T21             ;; *** R1 WRONG AFTER MOV ***
  816  013066  000263                2$:     SEVC                    ; SET V & C BITS
  817  013070  010000                        MOV     R0,R0
  818  013072  104405                        CHECKCC+5
  819  013074  104423  013004                ERROR,  T21             ;; *** INCORRECT CC BITS ***
  820
  821                                ;*******************************************************************
                                     .SBTTL      T22 -- CMP AND BIS
                                     ;*******************************************************************
  (2)  013100  104420  000022        T22:    BEGINTEST, 22           ;; CHECK SWITCH REGISTER OPTIONS.
  822  013104  000277                CMP0:   SCC
  823  013106  012700  177777                MOV     #177777,R0      ; LOAD REGISTER
  824  013112  050002                        BIS     R0,R2           ; CHECK THE BIS INSTRUCTION
  825  013114  104411                        CHECKCC+11              ; CHECK FOR CC = 11
  826  013116  104423  013100                ERROR,  T22             ;; *** INCORRECT CC BITS ***
  827  013122  020002                        CMP     R0,R2           ; CHECK COMPARE
  828  013124  001402                        BEQ     2$              ; CONTINUE IF OK
  829  013126  104423  013100                ERROR,  T22             ;; *** BIS OR CMP INSTRUCTION FAILED ***
  830  013132  022702  000077        2$:     CMP     #77,R2          ; CHECK IT AGAIN
  831  013136  100002                        BPL     3$              ; CONTINUE IF OK
  832  013140  104423  013100                ERROR,  T22             ;; *** CMP INSTRUCTION FAILED (WRONG CC) ***
  833  013144  020227  000077        3$:     CMP     R2,#77          ; ONCE MORE
  834  013150  100402                        BMI     4$              ; CONTINUE IF OK
  835  013152  104423  013100                ERROR,  T22             ;; *** BIC OR CMP FAILED ***
  836  013156                         4$:
  837
  838                                ;*******************************************************************
                                     .SBTTL      T23 -- BIC AND BIT
                                     ;*******************************************************************
  (2)  013156  104420  000023        T23:    BEGINTEST, 23           ;; CHECK SWITCH REGISTER OPTIONS.
  839  013162  012703  177777        BIC0:   MOV     #177777,R3      ; LOAD REGISTER
  840  013166  012700  001006                MOV     #TEMP,R0        ; PLACE THE ADDRESS OF TEMP IN R0
  841  013172  012710  125252                MOV     #125252,(R0)    ; SET (R0)
  842  013176  000277                        SCC
  843  013200  041003                        BIC     (R0),R3         ; CLEAR EVERY OTHER BIT
  844  013202  104401                        CHECKCC+1               ; CHECK FOR CC = 1
  845  013204  104423  013156                ERROR,  T23             ;; *** INCORRECT CC BITS ***
  846  013210  031003                        BIT     (R0),R3         ; CHECK IT
  847  013212  001402                        BEQ     1$              ; CONTINUE IF OK
  848  013214  104423  013156                ERROR,  T23             ;; *** BIC OR BIT INSTRUCTION FAILED ***
  849  013220  032703  052525        1$:     BIT     #52525,R3       ; CHECK IT
  850  013224  104401                        CHECKCC+1               ; CHECK FOR CC = 1
  851  013226  104423  013156                ERROR,  T23             ;; *** INCORRECT CC BITS ***
  852  013232  052703  125252                BIS     #125252,R3      ; SET THE BITS THAT WERE CLEARED
  853  013236  100402                        BMI     2$              ; CONTINUE IF OK
  854  013240  104423  013156                ERROR,  T23             ;; *** BIT OR BIS INSTRUCTION FAILED ***
  855  013244  042703  077777        2$:     BIC     #77777,R3       ; CLEAR ALL THE BITS EXCEPT FOR SIGN
  856  013250  104411                        CHECKCC+11              ; CHECK FOR CC = 11
```

```
  857  013252  104423  013156                ERROR,  T23             ;; *** INCORRECT CC BITS ***
  858  013256  012700  177777                MOV     #177777,R0
  859  013262  030003                        BIT     R0,R3           ; CHECK IT
  860  013264  104411                        CHECKCC+11              ; CHECK FOR CC = 11
  861  013266  104423  013156                ERROR,  T23             ;; *** INCORRECT CC BITS ***
  862  013272  000263                        SEVC                    ; SET V & C BITS
  863  013274  040000                        BIC     R0,R0
  864  013276  104405                        CHECKCC+5               ; CHECK CC = 5
  865  013300  104423  013156                ERROR,  T23             ;; *** INCORRECT CC BITS ***
  866  013304  005700                        TST     R0              ; CHECK R0 TO CONTAIN 0
  867  013306  001402                        BEQ     3$
  868  013310  104423  013156                ERROR,  T23             ;; *** BIC FAILED ***
  869  013314                        3$:
  870
  871                                ;*****************************************************************
  (2)                                .SBTTL      T24 -- INC AND DEC
  (2)                                ;*****************************************************************
  (2)  013314  104420  000024        T24:    BEGINTEST, 24           ;; CHECK SWITCH REGISTER OPTIONS.
  872  013320  012704  077777        INC0:   MOV     #77777,R4       ; R4=77777
  873  013324  000261                        SEC
  874  013326  005204                        INC     R4              ; ADD ONES INTO REG. 4
  875  013330  104413                        CHECKCC+13              ; CHECK FOR CC = 13
  876  013332  104423  013314                ERROR,  T24             ;; *** INCORRECT CC BITS ***
  877  013336  012704  177776                MOV     #177776,R4
  878  013342  005204                        INC     R4
  879  013344  104411                        CHECKCC+11              ; CHECK FOR CC = 11
  880  013346  104423  013314                ERROR,  T24             ;; *** INCORRECT CC BITS ***
  881  013352  005204                        INC     R4
  382  013354  104405                        CHECKCC+5               ; CHECK FOR CC = 5
  883  013356  104423  013314                ERROR,  T24             ;; *** INCORRECT CC BITS ***
  884  013362  005204                        INC     R4
  885  013364  104401                        CHECKCC+1               ; CHECK FOR CC = 1
  886  013366  104423  013314                ERROR,  T24             ;; *** INCORRECT CC BITS ***
  887  013372  022704  000001                CMP     #1,R4           ; CHECK IT
  888  013376  001402                        BEQ     4$              ; FAILED
  889  013400  104423  013314                ERROR,  T24             ;; *** INC INSTRUCTION FAILED ***
  890  013404  000261                4$:     SEC
  891  013406  005304                        DEC     R4              ; SUBTRACT ONES FROM REG. 4
  892  013410  104405                        CHECKCC+5               ; CHECK FOR CC = 5
  893  013412  104423  013314                ERROR,  T24             ;; *** INCORRECT CC BITS ***
  894  013416  005304                        DEC     R4
  895  013420  104411                        CHECKCC+11              ; CHECK FOR CC = 11
  896  013422  104423  013314                ERROR,  T24             ;; *** INCORRECT CC BITS ***
  897  013426  012704  100000                MOV     #100000,R4
  898  013432  005304                        DEC     R4              ;
  899  013434  104403                        CHECKCC+3               ; CHECK FOR CC = 3
  900  013436  104423  013314                ERROR,  T24             ;; *** INCORRECT CC BITS ***
  901  013442  005304                        DEC     R4              ;
  902  013444  104401                        CHECKCC+1               ; CHECK FOR CC = 1
  903  013446  104423  013314                ERROR,  T24             ;; *** INCORRECT CC BITS ***
  904
  905                                ;*****************************************************************
  (2)                                .SBTTL      T25 -- COM AND NEG
  (2)                                ;*****************************************************************
  (2)  013452  104420  000025        T25:    BEGINTEST, 25           ;; CHECK SWITCH REGISTER OPTIONS.
  906  013456  012703  125252        COM0:   MOV     #125252,R3      ; LOAD EVERY OTHER BIT
```

```
                                        SCC
                                        COM     R3              ; 1'S COMPLEMENT
                                        CHECKCC+1               ; CHECK FOR CC = 1
                                        ERROR,  T25             ;; *** INCORRECT CC BITS ***
                                        CMP     #52525,R3       ; CHECK IT
                                        BEQ     2$              ; CONTINUE IF OK
                                        ERROR,  T25             ;; *** COM INSTRUCTION FAILED ***
                                2$:     SCC
                                        COM     R3              ; COMPLEMENT BACK
                                        CHECKCC+11              ; CHECK FOR CC = 11
                                        ERROR,  T25             ;; *** INCORRECT CC BITS ***
                                        CMP     #125252,R3      ; CHECK IT
                                        BEQ     3$              ; CONTINUE IF OK
                                        ERROR,  T25             ;; *** COM INSTRUCTION FAILED ***
                                3$:     MOV     #177777,R3
                                        SCC
                                        COM     R3
                                        CHECKCC+5               ; CHECK FOR CC = 5
                                        ERROR,  T25             ;; *** INCORRECT CC BITS ***

                                NEG0:   MOV     #1,R0           ; LOAD THE REGISTER
                                        NEG     R0              ; 2'S COMPLEMENT
                                        CHECKCC+11              ; CHECK FOR CC = 11
                                        ERROR,  NEG0            ;; *** INCORRECT CC BITS ***
                                        CMP     #177777,R0      ; CHECK IT
                                        BEQ     2$              ; CONTINUE IF OK
                                        ERROR,  NEG0            ;; *** NEG INSTRUCTION FAILED ***
                                2$:     MOV     #100000,R0
                                        NEG     R0              ; 2'S COMPLEMENT
                                        CHECKCC+13              ; CHECK FOR CC = 13
                                        ERROR,  NEG0            ;; *** INCORRECT CC BITS ***
                                        CMP     #100000,R0      ; CHECK IT
                                        BEQ     3$              ; CONTINUE IF OK
                                        ERROR,  NEG0            ;; *** NEG FAILED ***
                                3$:

                                ;*******************************************************************
                                .SBTTL      T26 -- ROL AND ROR
                                ;*******************************************************************
                                T26:    BEGINTEST, 26           ;; CHECK SWITCH REGISTER OPTIONS.
                                ROL0:   MOV     #20000,R1       ; LOAD REGISTER
                                        CCC                     ; CLEAR FLAGS
                                        ROL     R1              ; SHIFT
                                        ROL     R1              ;
                                        CHECKCC+12              ; CHECK FOR CC = 12
                                        ERROR,  T26             ;; *** INCORRECT CC BITS ***
                                        CMP     #100000,R1      ; CHECK IT
                                        BEQ     1$              ; CONTINUE IF OK
                                        ERROR,  T26             ;; *** ROL INSTRUCTION FAILED ***
                                1$:     ROL     R1              ; SHIFT
                                        CHECKCC+7               ; CHECK FOR CC = 7
                                        ERROR,  T26             ;; *** INCORRECT CC BITS ***
                                        ROL     R1              ; SHIFT
                                        CMP     #1,R1           ; CHECK IT
                                        BEQ     2$              ; CONTINUE IF OK
                                        ERROR,  T26             ;; *** ROL FAILED ***
```

```
                                        2$:
013706                          ROR0:   MOV     #4,R2           ; LOAD REGISTER
013706  012702  000004                  CCC                     ; CLEAR FLAGS
013712  000257                          ROR     R2              ; SHIFT
013714  006002                          ROR     R2              ;
013716  006002                          CMP     #1,R2           ; CHECK IT
013720  022702  000001                  BEQ     1$              ; CONTINUE IF OK
013724  001402                          ERROR,  ROR0            ;; *** ROR INSTRUCTION FAILED ***
013726  104423  013706          1$:     ROR     R2              ; SHIFT
013732  006002                          CHECKCC+7               ; CHECK FOR CC = 7
013734  104407                          ERROR,  ROR0            ;; *** INCORRECT CC BITS ***
013736  104423  013706                  ROR     R2              ; SHIFT
013742  006002                          CHECKCC+12              ; CHECK FOR CC = 12
013744  104412                          ERROR,  ROR0            ;; *** INCORRECT CC BITS ***
013746  104423  013706                  CMP     #100000,R2      ; CHECK IT
013752  022702  100000                  BEQ     2$              ; CONTINUE IF OK
013756  001402                          ERROR,  ROR0            ;; *** ROR FAILED ***
013760  104423  013706          2$:
013764
                                        ;*******************************************************************
                                        .SBTTL      T27 -- ASL AND ASR
                                        ;*******************************************************************
013764  104420  000027          T27:    BEGINTEST, 27           ;; CHECK SWITCH REGISTER OPTIONS.
013770  012703  020000          ASL0:   MOV     #20000,R3       ; LOAD REGISTER
013774  000257                          CCC                     ; CLEAR FLAGS
013776  006303                          ASL     R3              ; SHIFT
014000  006303                          ASL     R3              ;
014002  104412                          CHECKCC+12              ; CHECK FOR CC = 12
014004  104423  013764                  ERROR,  T27             ;; *** INCORRECT CC BITS ***
014010  022703  100000                  CMP     #100000,R3      ; CHECK IT
014014  001402                          BEQ     4$              ; CONTINUE IF OK
014016  104423  013764                  ERROR,  T27             ;; *** ASL INSTRUCTION FAILED ***
014022  006303                  4$:     ASL     R3              ; SHIFT
014024  104407                          CHECKCC+7               ; CHECK FOR CC = 7
014026  104423  013764                  ERROR,  T27             ;; *** INCORRECT CC BITS ***
014032  006303                          ASL     R3              ; SHIFT
014034  104404                          CHECKCC+4               ; CHECK FOR CC = 4
014036  104423  013764                  ERROR,  T27             ;; *** INCORRECT CC BITS ***
014042
014042  012704  000004          ASR0:   MOV     #4,R4           ; LOAD REGISTER
014046  000257                          CCC                     ; CLEAR FLAGS
014050  006204                          ASR     R4              ; SHIFT
014052  006204                          ASR     R4              ;
014054  022704  000001                  CMP     #1,R4           ; CHECK IT
014060  001402                          BEQ     2$              ; CONTINUE IF OK
014062  104423  014042                  ERROR,  ASR0            ;; *** ASR INSTRUCTION FAILED ***
014066  006204                  2$:     ASR     R4              ; SHIFT
014070  104407                          CHECKCC+7               ; CHECK FOR CC = 7
014072  104423  014042                  ERROR,  ASR0            ;; *** INCORRECT CC BITS ***
014076  006204                          ASR     R4              ; SHIFT
014100  104404                          CHECKCC+4               ; CHECK FOR CC = 4
014102  104423  014042                  ERROR,  ASR0            ;; *** INCORRECT CC BITS ***
014106  012703  100002                  MOV     #100002,R3      ; LOAD REGISTER
014112  006203                          ASR     R3              ; SHIFT
014114  006203                          ASR     R3              ;
014116  104411                          CHECKCC+11              ; CHECK FOR CC = 11
```

```
1013  014120  104423  014042                ERROR,  ASR0            ;; *** INCORRECT CC BITS ***
1014  014124  022703  160000                CMP     #160000,R3      ; CHECK IT
1015  014130  001402                        BEQ     3$              ; CONTINUE IF OK
1016  014132  104423  014042                ERROR,  ASR0            ;; *** ASR FAILED ***
1017  014136                          3$:
1018
1019                                  ;*******************************************************************************
 (2)                                  .SBTTL      T30 -- ADC AND SBC
 (2)                                  ;*******************************************************************************
 (2)  014136  104420  000030          T30:    BEGINTEST, 30           ;; CHECK SWITCH REGISTER OPTIONS.
1020  014142  005000                  ADC0:   CLR     R0              ; CLEAR THE REGISTER
1021  014144  000257                          CCC                     ; CLEAR FLAGS
1022  014146  005500                          ADC     R0              ; ADD C BIT = 0
1023  014150  104404                          CHECKCC+4               ; CHECK FOR CC = 4
1024  014152  104423  014136                  ERROR,  T30             ;; *** INCORRECT CC BITS ***
1025  014156  000261                          SEC                     ; C=1
1026  014160  005500                          ADC     R0              ; ADD C BIT=1
1027  014162  000261                          SEC                     ; C=1
1028  014164  005500                          ADC     R0              ; AGAIN
1029  014166  104400                          CHECKCC+0               ; CHECK FOR CC = 0
1030  014170  104423  014136                  ERROR,  T30             ;; *** INCORRECT CC BITS ***
1031  014174  022700  000002                  CMP     #2,R0           ; CHECK IT
1032  014200  001402                          BEQ     4$              ; CONTINUE IF OK
1033  014202  104423  014136                  ERROR,  T30             ;; *** ADC INSTRUCTION FAILED ***
1034  014206  012700  077777          4$:     MOV     #77777,R0       ; LOAD LARGEST POSITVE NUMBER
1035  014212  000261                          SEC                     ; C=1
1036  014214  005500                          ADC     R0              ; ADD C BIT=1
1037  014216  104412                          CHECKCC+12              ; CHECK FOR CC = 12
1038  014220  104423  014136                  ERROR,  T30             ;; *** INCORRECT CC BITS ***
1039  014224  022700  100000                  CMP     #100000,R0      ; CHECK IT
1040  014230  001402                          BEQ     6$              ; FAILED
1041  014232  104423  014136                  ERROR,  T30             ;; *** ADC INSTRUCTION FAILED ***
1042  014236  012700  177777          6$:     MOV     #-1,R0          ; LOAD -1
1043  014242  000261                          SEC                     ; C=1
1044  014244  005500                          ADC     R0              ; ADD C BIT=1
1045  014246  104405                          CHECKCC+5               ; CHECK FOR CC = 5
1046  014250  104423  014136                  ERROR,  T30             ;; *** INCORRECT CC BITS ***
1047
1048  014254  012701  000003          SBC0:   MOV     #3,R1           ; LOAD REGISTER
1049  014260  000257                          CCC                     ; CLEAR FLAGS
1050  014262  005601                          SBC     R1              ; SUBTRACT C BIT=0
1051  014264  104400                          CHECKCC+0               ; CHECK FOR CC = 0
1052  014266  104423  014254                  ERROR,  SBC0            ;; *** INCORRECT CC BITS ***
1053  014272  022701  000003                  CMP     #3,R1           ; CHECK IT
1054  014276  001402                          BEQ     2$              ; CONTINUE IF OK
1055  014300  104423  014254                  ERROR,  SBC0            ;; *** SBC INSTRUCTION FAILED ***
1056  014304  000261                  2$:     SEC                     ; C=1
1057  014306  005601                          SBC     R1              ; SUBTRACT C BIT=1
1058  014310  000261                          SEC                     ; C=1
1059  014312  005601                          SBC     R1              ;
1060  014314  104400                          CHECKCC+0               ; CHECK FOR CC = 0
1061  014316  104423  014254                  ERROR,  SBC0            ;; *** INCORRECT CC BITS ***
1062  014322  022701  000001                  CMP     #1,R1           ; CHECK IT
1063  014326  001402                          BEQ     3$              ; CONTINUE IF OK
1064  014330  104423  014254                  ERROR,  SBC0            ;; *** SBC INSTRUCTION FAILED ***
1065  014334  000261                  3$:     SEC                     ; C=1
```

```
 1066  014336  005601                        SBC     R1              ; SUBTRACT C BIT=1
 1067  014340  104404                        CHECKCC+4               ; CHECK FOR CC = 4
 1068  014342  104423  014254                ERROR,  SBCO            ;; *** INCORRECT CC BITS ***
 1069  014346  000261                        SEC                     ; C=1
 1070  014350  005601                        SBC     R1              ; SUBTRACT C BIT = 1
 1071  014352  104411                        CHECKCC+11              ; CHECK FOR CC = 11
 1072  014354  104423  014254                ERROR,  SBCO            ;; *** INCORRECT CC BITS ***
 1073  014360  022701  177777                CMP     #-1,R1          ; CHECK IT
 1074  014364  001402                        BEQ     4$              ; CONTINUE IF F OK
 1075  014366  104423  014254                ERROR,  SBCO            ;; *** SBC INSTRUCTION FAILED ***
 1076  014372  012701  100000        4$:     MOV     #100000,R1      ; LOAD R1
 1077  014376  000261                        SEC                     ; C=1
 1078  014400  005601                        SBC     R1              ; SUBTRACT C BIT = 1
 1079  014402  104402                        CHECKCC+2               ; CHECK FOR CC = 2
 1080  014404  104423  014254                ERROR,  SBCO            ;; *** INCORRECT CC BITS ***
 1081
 1082                                ;*******************************************************************************
  (2)                                .SBTTL      T31 -- SXT, SWAB AND XOR
  (2)                                ;*******************************************************************************
  (2)   014410  104420  000031        T31:    BEGINTEST, 31           ;; CHECK SWITCH REGISTER OPTIONS.
 1083  014414  005002                SXT0:   CLR     R2              ; CLEAR REGISTER
 1084  014416  000277                        SCC
 1085  014420  000254                        CLNZ
 1086  014422  006702                        SXT     R2              ; SIGN EXTEND
 1087  014424  104405                        CHECKCC+5               ; CHECK FOR CC = 5
 1088  014426  104423  014410                ERROR,  T31             ;; *** INCORRECT CC BITS ***
 1089  014432  005702                        TST     R2              ; REG. 2 SHOULD STILL BE 0
 1090  014434  001402                        BEQ     2$              ; CONTINUE IF OK
 1091  014436  104423  014410                ERROR,  T31             ;; *** SXT INSTRUCTION FAILED ***
 1092  014442  000273                2$:     SENVC                   ; SET N, V & C BITS
 1093  014444  006702                        SXT     R2              ; SIGN EXTEND
 1094  014446  104411                        CHECKCC+11              ; CHECK FOR CC = 11
 1095  014450  104423  014410                ERROR,  T31             ;; *** INCORRECT CC BITS ***
 1096  014454  022702  177777                CMP     #-1,R2          ; REG. 2 SHOULD NOW HAVE -1
 1097  014460  001402                        BEQ     SWAB0           ; CONTINUE IF OK
 1098  014462  104423  014410                ERROR,  T31             ;; *** SXT FAILED ***
 1099
 1100  014466  012703  125125        SWAB0:  MOV     #125125,R3      ; LOAD BIT PATTERN INTO REGISTER
 1101  014472  000277                        SCC
 1102  014474  000250                        CLN
 1103  014476  000303                        SWAB    R3              ; SWAP BYTES OF REGISTER
 1104  014500  104410                        CHECKCC+10              ; CHECK FOR CC = 10
 1105  014502  104423  014466                ERROR,  SWAB0           ;; *** INCORRECT CC BITS ***
 1106  014506  022703  052652                CMP     #52652,R3       ; CHECK IT
 1107  014512  001402                        BEQ     1$              ; CONTINUE IF OK
 1108  014514  104423  014466                ERROR,  SWAB0           ;; *** SWAB INSTRUCTION FAILED ***
 1109  014520  012703  000377        1$:     MOV     #377,R3
 1110  014524  000277                        SCC
 1111  014526  000244                        CLZ
 1112  014530  000303                        SWAB    R3
 1113  014532  104404                        CHECKCC+4               ; CHECK FOR CC = 4
 1114  014534  104423  014466                ERROR,  SWAB0           ;; *** INCORRECT CC BITS ***
 1115  014540  022703  177400                CMP     #177400,R3
 1116  014544  001402                        BEQ     XOR0
 1117  014546  104423  014466                ERROR,  SWAB0           ;; *** SWAB FAILED ***
 1118
```

```
 1119  014552  012704  177777          XOR0:   MOV     #-1,R4          ; LOAD REGISTERS
 1120  014556  012703  177777                  MOV     #-1,R3          ;
 1121  014562  000277                          SCC
 1122  014564  074403                          XOR     R4,R3           ; SHOULD PRODUCE 0'S IN REG. 3
 1123  014566  104405                          CHECKCC+5               ; CHECK FOR CC = 5
 1124  014570  104423  014552                  ERROR,  XOR0            ;; *** INCORRECT CC BITS ***
 1125  014574  012703  077777                  MOV     #77777,R3
 1126  014600  010400                          MOV     R4,R0           ; PLACE A -1 IN R0
 1127  014602  000263                          SEVC                    ; SET V & C BITS
 1128  014604  000244                          CLZ
 1129  014606  074003                          XOR     R0,R3
 1130  014610  104411                          CHECKCC+11              ; CHECK FOR CC = 11
 1131  014612  104423  014552                  ERROR,  XOR0            ;; *** INCORRECT CC BITS ***
 1132  014616  012702  125252                  MOV     #125252,R2      ; LOAD REGISTERS
 1133  014622  012704  052525                  MOV     #52525,R4       ;
 1134  014626  000277                          SCC
 1135  014630  074204                          XOR     R2,R4           ; SHOULD PRODUCE ALL 1'S IN REG. 4
 1136  014632  104411                          CHECKCC+11              ; CHECK FOR CC = 11
 1137  014634  104423  014552                  ERROR,  XOR0            ;; *** INCORRECT CC BITS ***
 1138  014640  022704  177777                  CMP     #-1,R4          ; CHECK IT
 1139  014644  001402                          BEQ     1$              ; CONTINUE IF OK
 1140  014646  104423  014552                  ERROR,  XOR0            ;; *** XOR FAILED ***
 1141  014652                          1$:
 1142
 1143                                  ;*******************************************************************************
                                       .SBTTL      T32 -- ADD AND SUB
                                       ;*******************************************************************************
 1147  014652  104420  000032          T32:    BEGINTEST, 32           ;; CHECK SWITCH REGISTER OPTIONS.
 1148  014656  012701  021421          ADD0:   MOV     #21421,R1       ; LOAD REGISTERS
 1149  014662  060101                          ADD     R1,R1           ; ADD
 1150  014664  104400                          CHECKCC+0               ; CHECK FOR CC = 0
 1151  014666  104423  014652                  ERROR,  T32             ;; *** INCORRECT CC BITS ***
 1152  014672  022701  043042                  CMP     #43042,R1       ; CHECK IT
 1153  014676  001402                          BEQ     1$              ; CONTINUE IF OK
 1154  014700  104423  014652                  ERROR,  T32             ;; *** ADD INSTRUCTION FAILED ***
 1155  014704  012700  156357          1$:     MOV     #-21421,R0      ; LOAD REGISTERS
 1156  014710  060000                          ADD     R0,R0           ; ADD
 1157  014712  104411                          CHECKCC+11              ; CHECK FOR CC = 11
 1158  014714  104423  014652                  ERROR,  T32             ;; *** INCORRECT CC BITS ***
 1159  014720  022700  134736                  CMP     #-43042,R0      ; CHECK IT
 1160  014724  001402                          BEQ     2$              ; CONTINUE IF OK
 1161  014726  104423  014652                  ERROR,  T32             ;; *** ADD INSTRUCTION FAILED ***
 1162  014732  012702  100000          2$:     MOV     #100000,R2      ; LOAD REGISTERS
 1163  014736  060202                          ADD     R2,R2           ; ADD SHOULD RESULT AS 0'S
 1164  014740  104407                          CHECKCC+7               ; CHECK FOR CC = 7
 1165  014742  104423  014652                  ERROR,  T32             ;; *** INCORRECT CC BITS ***
 1166  014746  012704  021421                  MOV     #21421,R4       ; LOAD REGISTERS
 1167  014752  012701  156357                  MOV     #-21421,R1      ;
 1168  014756  060401                          ADD     R4,R1           ; ADD SHOULD RESULT AS 0'S
 1169  014760  001402                          BEQ     3$              ; CONTINUE IF OK
 1170  014762  104423  014652                  ERROR,  T32             ;; *** ADD INSTRUCTION FAILED ***
 1171  014766  005404                  3$:     NEG     R4              ; SWITCH SOURCE AND DESTINATION
 1172  014770  012701  021421                  MOV     #21421,R1       ;
 1173  014774  060104                          ADD     R1,R4           ; SHOULD RESULT AS 0'S
 1174  014776  001402                          BEQ     SUB0            ; CONTINUE IF OK
 1175  015000  104423  014652                  ERROR,  T32             ;; *** ADD FAILED ***
```

```
1172
1173  015004  012702  021421          SUB0:   MOV     #21421,R2       ; LOAD REGISTERS
1174  015010  012703  156357                  MOV     #-21421,R3      ;
1175  015014  160203                          SUB     R2,R3           ; RESULT SHOULD=-43042
1176  015016  104410                          CHECKCC+10              ; CHECK FOR CC = 10
1177  015020  104423  015004                  ERROR,  SUB0            ;; *** INCORRECT CC BITS ***
1178  015024  022703  134736                  CMP     #-43042,R3      ; CHECK IT
1179  015030  001402                          BEQ     4$              ; CONTINUE IF OK
1180  015032  104423  015004                  ERROR,  SUB0            ;; *** SUB INSTRUCTION FAILED ***
1181  015036  012703  021421          4$:     MOV     #21421,R3       ; LOAD REGISTER
1182  015042  010204                          MOV     R2,R4           ; NOW R4 = #21421
1183  015044  160403                          SUB     R4,R3           ; RESULT SHOULD=0
1184  015046  001402                          BEQ     6$
1185  015050  104423  015004                  ERROR,  SUB0            ;; *** SUB NSTRUCTION FAILED ***
1186  015054  012703  177777          6$:     MOV     #-1,R3          ; LOAD REGISTERS
1187  015060  012702  077777                  MOV     #77777,R2       ; LOAD REGISTERS
1188  015064  160302                          SUB     R3,R2           ; RESULT SHOULD BE 100000 AND OVERFLOW
1189  015066  104413                          CHECKCC+13              ; CHECK FOR CC = 13
1190  015070  104423  015004                  ERROR,  SUB0            ;; *** INCORRECT CC BITS ***
1191  015074  022702  100000                  CMP     #100000,R2      ; CHECK IT
1192  015100  001402                          BEQ     8$              ; CONTINUE IF OK
1193  015102  104423  015004                  ERROR,  SUB0            ;; *** SUB INSTRUCTION FAILED ***
1194  015106  012704  177777          8$:     MOV     #-1,R4
1195  015112  160304                          SUB     R3,R4
1196  015114  104404                          CHECKCC+4               ; CHECK FOR CC = 4
1197  015116  104423  015004                  ERROR,  SUB0            ;; *** INCORRECT CC BITS ***
1198
1199                                  ;*******************************************************************************
 (2)                                .SBTTL      T33 -- MTPS AND MFPS
 (2)                                ;*******************************************************************************
 (2)  015122  104420  000033          T33:    BEGINTEST, 33           ;; CHECK SWITCH REGISTER OPTIONS.
1200  015126  012701  177777          TPSW:   MOV     #177777,R1
1201  015132  005000                          CLR     R0
1202  015134  106400                          MTPS    R0              ; SET PSW TO 0
1203  015136  104400                          CHECKCC+0               ; CHECK FOR CC = 0
1204  015140  104423  015122                  ERROR,  T33             ;; *** INCORRECT CC BITS ***
1205  015144  106701                          MFPS    R1              ; MOVE PSW TO R1
1206  015146  001402                          BEQ     2$              ; BR IF BIT 8 OF PSW WAS EXTENDED IN R1
1207  015150  104423  015122                  ERROR,  T33             ;; *** MTPS OR MFPS INSTRUCTION FAILED ***
1208  015154  104404                  2$:     CHECKCC+4               ; CHECK FOR CC = 4
1209  015156  104423  015122                  ERROR,  T33             ;; *** INCORRECT CC BITS ***
1210  015162  012700  000337                  MOV     #337,R0
1211  015166  106400                          MTPS    R0              ; EXPECT 317 SINCE MTPS DOESN'T SET T BIT
1212  015170  104417                          CHECKCC+17              ; CHECK FOR CC = 17
1213  015172  104423  015122                  ERROR,  T33             ;; *** INCORRECT CC BITS ***
1214  015176  106701                          MFPS    R1              ; MOVE PSW TO R1
1215  015200  104411                          CHECKCC+11              ; CHECK FOR CC   11 (C BIT SHOULD NOT BE EFFECTED BY MFP
1216  015202  104423  015122                  ERROR,  T33             ;; *** INCORRECT CC BITS ***
1217  015206  022701  177717                  CMP     #177717,R1      ; CHECK TO SEE IF BIT 8 OF PSW WAS EXTENDED THRU R1
1218  015212  001402                          BEQ     3$
1219  015214  104423  015122                  ERROR,  T33             ;; *** MTPS OR MFPS INSTRUCTION FAILED ***
1220  015220                          3$:
1221                                  ;
1222                                  .SBTTL
1223                                  .SBTTL  NON-ZERO ADDRESS MODES
1224                                  ;
```

```
                        ;*****************************************************************************
                        .SBTTL      T34 -- TEST MODES 0 AND 1 USING MOVB AND MOV
                        ;*****************************************************************************
015220  104420  000034  T34:    BEGINTEST, 34           ;; CHECK SWITCH REGISTER OPTIONS.
015224  112700  000252  MODE0:  MOVB    #252,R0         ; LOAD REGISTERS
015230  110001           MOVB    R0,R1           ;
015232  110102           MOVB    R1,R2           ;
015234  122702  000252   CMPB    #252,R2         ; CHECK IT
015240  001402           BEQ     1$              ; OK, CONTINUE
015242  104423  015220   ERROR,  T34             ;; *** MOVB INSTRUCTION FAILED IN MODE 0 ***
015246  012700  125252   1$:     MOV     #125252,R0      ; LOAD REGISTERS
015252  010001           MOV     R0,R1           ;
015254  010102           MOV     R1,R2           ;
015256  022702  125252   CMP     #125252,R2      ; CHECK IT
015262  001402           BEQ     MODE1           ; OK, CONTINUE
015264  104423  015220   ERROR,  T34             ;; *** MOV INSTRUCTION FAILED IN MODE 0 ***

015270  012700  001006   MODE1:  MOV     #TEMP,R0        ; LOAD ADDRESSES INTO REGS.
015274  012701  001010           MOV     #TEMP1,R1       ;
015300  012702  001012           MOV     #TEMP2,R2       ;
015304  005067  163502           CLR     TEMP2           ; START CLEAN
015310  112710  000125           MOVB    #125,(R0)       ; LOAD THE LOCATIONS
015314  111011           MOVB    (R0),(R1)       ; TEMP => TEMP1
015316  111112           MOVB    (R1),(R2)       ; TEMP1 => TEMP2
015320  122767  000125  163464   CMPB    #125,TEMP2      ; CHECK IT
015326  001402           BEQ     1$              ; OK, CONTINUE
015330  104423  015220   ERROR,  T34             ;; *** MOV INSTRUCTION FAILED IN MODE 1 ***
015334  012710  052525   1$:     MOV     #52525,(R0)     ; LOAD THE LOCATIONS
015340  011011           MOV     (R0),(R1)       ; TEMP => TEMP1
015342  011112           MOV     (R1),(R2)       ; TEMP1 => TEMP2
015344  022767  052525  163440   CMP     #52525,TEMP2    ; CHECK IT
015352  001402           BEQ     2$              ; OK, CONTINUE
015354  104423  015220   ERROR,  T34             ;; *** MOV INSTRUCTION FAILED IN MODE 1 ***
015360                   2$:

                        ;*****************************************************************************
                        .SBTTL      T35 -- TEST MODE 2 USING MOVB AND MOV
                        ;*****************************************************************************
015360  104420  000035  T35:    BEGINTEST, 35           ;; CHECK SWITCH REGISTER OPTIONS.
015364  012700  001006   MODE2:  MOV     #TEMP,R0        ; LOAD ADDRESSES
015370  012701  001010           MOV     #TEMP1,R1       ;
015374  012702  001012           MOV     #TEMP2,R2
015400  105022           CLRB    (R2)+           ; START CLEAN
015402  112710  000252           MOVB    #252,(R0)       ; LOAD THE LOCATIONS
015406  112021           MOVB    (R0)+,(R1)+     ; TEMP => TEMP1
015410  105201           INCB    R1              ; MAKE IT EVEN
015412  111167  163370           MOVB    (R1),TEMP       ; MORE 0'S INTO TEMP
015416  105200           INCB    R0              ; MAKE IT EVEN
015420  112021           MOVB    (R0)+,(R1)+     ; TEMP1 => TEMP2
015422  124227  000252           CMPB    -(R2),#252      ; CHECK IT
015426  001003           BNE     1$              ; FAILED
015430  105767  163352           TSTB    TEMP            ; CHECK IT
015434  001402           BEQ     2$              ; OK, CONTINUE
015436                   1$:
015436  104423  015360           ERROR,  T35             ;; *** INSTRUCTIONS FAILED IN MODE 2 ***
```

```
 1276  015442  005741                  2$:     TST     -(R1)
 1277  015444  005022                          CLR     (R2)+           ; START CLEAN
 1278  015446  012740  125252                  MOV     #125252,-(R0)   ; LOAD LOCATIONS
 1279  015452  012020                          MOV     (R0)+,(R0)+     ; TEMP => TEMP1
 1280  015454  011067  163326                  MOV     (R0),TEMP       ; 0 => TEMP
 1281  015460  012121                          MOV     (R1)+,(R1)+     ; 125252 => TEMP2
 1282  015462  024227  125252                  CMP     -(R2),#125252   ; CHECK IT
 1283  015466  001003                          BNE     3$              ; FAILED
 1284  015470  005767  163312                  TST     TEMP            ; CHECK IT
 1285  015474  001402                          BEQ     4$              ; OK, CONTINUE
 1286  015476                          3$:
 (1)   015476  104423  015360                  ERROR,  T35             ;; *** INSTRUCTIONS FAILED IN MODE 2 ***
 1288  015502                          4$:
 1289
 1290                                  ;*******************************************************  ****************
 (2)                                    .SBTTL      T36 -- TEST MODE 3 USING MOVB AND MOV
 (3)                                    ;*************************************************************************
 (4)   015502  104420  000036          T36:    BEGINTEST, 36           ;; CHECK SWITCH REGISTER OPTIONS.
 1294  015506  012767  001006  163264  MODE3:  MOV     #TEMP,ADR       ; LOAD ADDRESSES
 1295  015514  012767  001010  163260          MOV     #TEMP1,ADR1     ;
 1296  015522  012767  001012  163254          MOV     #TEMP2,ADR2     ;
 1297  015530  012700  001000                  MOV     #ADR,R0         ; LOAD ADDRESSES OF ADDRESSES
 1298  015534  012701  001002                  MOV     #ADR1,R1        ;
 1299  015540  105067  163246                  CLRB    TEMP2           ; START CLEAN
 1300  015544  112767  000125  163234          MOVB    #125,TEMP       ;
 1301  015552  113031                          MOVB    @(R0)+,@(R1)+   ; TEMP => TEMP1
 1302  015554  113167  163226                  MOVB    @(R1)+,TEMP     ; TEMP2 => TEMP
 1303  015560  113030                          MOVB    @(R0)+,@(R0)+   ; TEMP1 => TEMP2
 1304  015562  122767  000125  163222          CMPB    #125,TEMP2      ; CHECK IT
 1305  015570  001003                          BNE     1$              ; FAILED
 1306  015572  105767  163210                  TSTB    TEMP            ; CHECK IT
 1307  015576  001402                          BEQ     2$              ; OK, CONTINUE
 1308  015600                          1$:
 (5)   015600  104423  015502                  ERROR,  T36             ;; *** INSTRUCTIONS FAILED IN MODE 3 ***
 1310
 1311  015604  005067  16320a          2$:     CLR     TEMP2           ; START CLEAN
 1312  015610  012767  052525  163170          MOV     #52525,TEMP     ; LOAD LOCATIONS
 1313  015616  012700  001000                  MOV     #ADR,R0         ; LOAD ADDRESSES OF ADDRESSES
 1314  015622  012701  001002                  MOV     #ADR1,R1        ;
 1315  015626  013030                          MOV     @(R0)+,@(R0)+   ; TEMP => TEMP1
 1316  015630  013067  163152                  MOV     @(R0)+,TEMP     ; TEMP2 => TEMP
 1317  015634  013131                          MOV     @(R1)+,@(R1)+   ; TEMP1 => TEMP2
 1318  015636  022767  052525  163146          CMP     #52525,TEMP2    ; CHECK IT
 1319  015644  001003                          BNE     3$              ; FAILED
 1320  015646  005767  163134                  TST     TEMP            ; CHECK IT
 1321  015652  001402                          BEQ     4$              ; OK, CONTINUE
 1322  015654                          3$:
 (6)   015654  104423  015502                  ERROR,  T36             ;; *** INSTRUCTIONS FAILED IN MODE 3 ***
 1324  015660                          4$:
 1325
 1326                                  ;*************************************************************************
 (7)                                    .SBTTL      T37 -- TEST MODE 4 USING MOVB AND MOV
 (8)                                    ;*************************************************************************
 (9)   015660  104420  000037          T37:    BEGINTEST, 37           ;; CHECK SWITCH REGISTER OPTIONS.
 1329  015664  105067  163116          MODE4:  CLRB    TEMP            ; START CLEAN
 1330  015670  012700  001006                  MOV     #TEMP,R0        ; LOAD ADDRESSES
```

```
015674  012701  001010            MOV     #TEMP1,R1       ;
015700  012702  001012            MOV     #TEMP2,R2       ;
015704  005202                    INC     R2              ; ADJUST THE POINTER
015706  021267  163101            CMP     (R2),TEMP2+1
015712  001402                    BEQ     1$
015714  104423  015660            ERROR,  T37             ;; *** INSTRUCTIONS FAILED IN MODE 4 ***
015720  112742  000252    1$:     MOVB    #252,-(R2)      ; LOAD TEMP2
015724  005201                    INC     R1              ; ADJUST THE POINTERS
015726  005202                    INC     R2              ;
015730  114241                    MOVB    -(R2),-(R1)     ; TEMP2 => TEMP1
015732  005200                    INC     R0              ; ADJUST THE POINTERS
015734  005202                    INC     R2              ;
015736  114042                    MOVB    -(R0),-(R2)     ; TEMP => TEMP2
015740  105200                    INCB    R0              ; ADJUST THE POINTERS
015742  021067  163041            CMP     (R0),TEMP+1
015746  001402                    BEQ     2$
015750  104423  015660            ERROR,  T37             ;; *** INSTRUCTIONS FAILED IN MODE 4 ***
015754  105201            2$:     INCB    R1              ;
015756  114140                    MOVB    -(R1),-(R0)     ; TEMP1 => TEMP
015760  122767  000252  163020    CMPB    #252,TEMP       ; CHECK IT
015766  001003                    BNE     3$              ; FAILED
015770  105767  163016            TSTB    TEMP2           ; CHECK IT
015774  001402                    BEQ     4$              ; OK, CONTINUE
015776                    3$:
015776  104423  015660            ERROR,  T37             ;; *** INSTRUCTIONS FAILED IN MODE 4 ***

016002  005067  163000    4$:     CLR     TEMP            ; START CLEAN
016006  012700  001006            MOV     #TEMP,R0        ; LOAD ADDRESSES
016012  012701  001010            MOV     #TEMP1,R1       ;
016016  012702  001012            MOV     #TEMP2,R2       ;
016022  005722                    TST     (R2)+           ; ADJUST THE POINTER
016024  021267  162764            CMP     (R2),TEMP2+2
016030  001402                    BEQ     5$
016032  104423  015660            ERROR,  T37             ;; *** INSTRUCTIONS FAILED IN MODE 4 ***
016036  012742  125252    5$:     MOV     #125252,-(R2)   ; LOAD TEMP2
016042  005721                    TST     (R1)+           ; ADJUST THE POINTERS
016044  005722                    TST     (R2)+           ;
016046  014241                    MOV     -(R2),-(R1)     ; TEMP2 => TEMP1
016050  005720                    TST     (R0)+           ; ADJUST POINTERS
016052  005722                    TST     (R2)+           ;
016054  014042                    MOV     -(R0),-(R2)     ; TEMP => TEMP2
016056  005720                    TST     (R0)+           ; ADJUST THE POINTERS
016060  005721                    TST     (R1)+           ;
016062  014140                    MOV     -(R1),-(R0)     ; TEMP1 => TEMP
016064  022767  125252  162714    CMP     #125252,TEMP    ; CHECK IT
016072  001003                    BNE     6$              ; FAILED
016074  005767  162712            TST     TEMP2           ; CHECK IT
016100  001402                    BEQ     7$              ; OK, CONTINUE
016102                    6$:
016102  104423  015660            ERROR,  T37             ;; *** INSTRUCTIONS FAILED IN MODE 4 ***
016106                    7$:

                        ;*******************************************************************
                        .SBTTL      T40 -- TEST MODE 5 USING MOVB AND MOV
                        ;*******************************************************************
016106  104420  000040  T40:    BEGINTEST, 40           ;; CHECK SWITCH REGISTER OPTIONS.
```

```
1372  016112  105067  162670          MODE5:  CLRB    TEMP            ; START CLEAN
1373  016116  012767  001006  162654          MOV     #TEMP,ADR       ; LOAD ADDRESSES
1374  016124  012767  001010  162650          MOV     #TEMP1,ADR1     ;
1375  016132  012767  001012  162644          MOV     #TEMP2,ADR2     ;
1376  016140  012700  001000                  MOV     #ADR,R0         ; LOAD ADDRESSES OF ADDRESSES
1377  016144  012701  001002                  MOV     #ADR1,R1        ;
1378  016150  012702  001004                  MOV     #ADR2,R2        ;
1379  016154  005722                          TST     (R2)+           ; ADJUST THE POINTER
1380  016156  112752  000125                  MOVB    #125,@-(R2)     ; LOAD TEMP2
1381  016162  022122                          CMP     (R1)+,(R2)+     ; ADJUST THE POINTERS
1382  016164  115251                          MOVB    @-(R2),@-(R1)   ; TEMP2 => TEMP1
1383  016166  022022                          CMP     (R0)+,(R2)+     ; ADJUST THE POINTERS
1384  016170  115052                          MOVB    @-(R0),@-(R2)   ; TEMP => TEMP2
1385  016172  022022                          CMP     (R0)+,(R2)+     ; ADJUST THE POINTERS
1386  016174  125052                          CMPB    @-(R0),@-(R2)   ;CHECK IT
1387  016176  001402                          BEQ     1$
1388  016200  104423  016106                  ERROR,  T40             ;; *** INSTRUCTION FAILED IN MODE 5 ***
1389  016204  022120                  1$:     CMP     (R1)+,(R0)+     ; ADJUST THE POINTERS
1390  016206  115150                          MOVB    @-(R1),@-(R0)   ; TEMP1 => TEMP
1391  016210  122767  000125  162570          CMPB    #125,TEMP       ; CHECK IT
1392  016216  001003                          BNE     2$              ; FAILED
1393  016220  105767  162566                  TSTB    TEMP2           ; CHECK IT
1394  016224  001402                          BEQ     3$              ; OK, CONTINUE
1395  016226                          2$:
1396  016226  104423  016106                  ERROR,  T40             ;; *** INSTRUCTIONS FAILED IN MODE 5 ***
1397
1398  016232  005067  162550          3$:     CLR     TEMP            ; START CLEAN
1399  016236  012700  001000                  MOV     #ADR,R0         ; LOAD ADDRESSES OF ADDRESSES
1400  016242  012701  001002                  MOV     #ADR1,R1        ;
1401  016246  012702  001004                  MOV     #ADR2,R2        ;
1402  016252  005722                          TST     (R2)+           ; ADJUST THE POINTER
1403  016254  012752  052525                  MOV     #52525,@-(R2)   ; LOAD TEMP2
1404  016260  022122                          CMP     (R1)+,(R2)+     ; ADJUST THE POINTERS
1405  016262  015251                          MOV     @-(R2),@-(R1)   ; TEMP2 => TEMP1
1406  016264  022022                          CMP     (R0)+,(R2)+     ; ADJUST THE POINTERS
1407  016266  015052                          MOV     @-(R0),@-(R2)   ; TEMP => TEMP2
1408  016270  022021                          CMP     (R0)+,(R1)+     ; ADJUST THE POINTERS
1409  016272  015150                          MOV     @-(R1),@-(R0)   ; TEMP1 => TEMP
1410  016274  022767  052525  162504          CMP     #52525,TEMP     ; CHECK IT
1411  016302  001003                          BNE     4$              ; FAILED
1412  016304  005767  162502                  TST     TEMP2           ; CHECK IT
1413  016310  001402                          BEQ     5$              ; OK, CONTINUE
1414  016312                          4$:
1415  016312  104423  016106                  ERROR,  T40             ;; *** INSTRUCTIONS FAILED IN MODE 5 ***
1416  016316                          5$:
1417
1418                                  ;*******************************************************************************
1419                                  .SBTTL      T41 -- TEST MODE 6 USING MOVB AND MOV
1420                                  ;*******************************************************************************
1421  016316  104420  000041          T41:    BEGINTEST, 41           ;; CHECK SWITCH REGISTER OPTIONS.
1422  016322  005067  162464          MODE6:  CLR     TEMP2           ; START CLEAN
1423  016326  012700  001006                  MOV     #TEMP,R0        ; LOAD ADDRESSES
1424  016332  012701  001010                  MOV     #TEMP1,R1       ;
1425  016336  012702  001012                  MOV     #TEMP2,R2       ;
1426  016342  112760  000252  000000          MOVB    #252,0(R0)      ; LOAD TEMP (LOW BYTE)
1427  016350  112760  000252  000001          MOVB    #252,1(R0)      ; LOAD TEMP (HIGH BYTE)
```

```
1423  016356  022767  125252  162422          CMP     #125252,TEMP    ; CHECK IT
1424  016364  001012                          BNE     1$              ; FAILED
1425  016366  116062  000001  000000          MOVB    1(R0),0(R2)     ; TEMP(H) => TEMP2(L)
1426  016374  116160  000002  000005          MOVB    2(R1),5(R0)     ; TEMP2(L) => TEMP2(H)
1427  016402  022767  125252  162402          CMP     #125252,TEMP2   ; CHECK IT
1428  016410  001402                          BEQ     2$              ; OK, CONTINUE
1429  016412                          1$:
  (2) 016412  104423  016316                  ERROR,  T41             ;; *** INSTRUCTIONS FAILED IN MODE 6 ***
1430
1431  016416  005067  162366          2$:     CLR     TEMP1           ; START CLEAN
1432  016422  012760  052525  000000          MOV     #52525,0(R0)    ; LOAD TEMP
1433  016430  016260  177774  000002          MOV     -4(R2),2(R0)    ; TEMP => TEMP1
1434  016436  022767  052525  162344          CMP     #52525,TEMP1    ; CHECK IT
1435  016444  001402                          BEQ     3$              ; OK, CONTINUE
1436  016446  104423  016316                  ERROR,  T41             ;; *** INSTRUCTIONS FAILED IN MODE 6 ***
1437  016452                          3$:
1438
1439                                  ;*******************************************************************************
  (2)                             .SBTTL      T42 -- TEST MODE 7 USING MOVB AND MOV
  (2)                             ;*******************************************************************************
  (2) 016452  104420  000042      T42:    BEGINTEST, 42           ;; CHECK SWITCH REGISTER OPTIONS.
1440  016456  005067  162326      MODE7:  CLR     TEMP1           ; START CLEAN
1441  016462  012767  001006  162310        MOV     #TEMP,ADR       ; LOAD ADDRESSES
1442  016470  012767  001010  162304        MOV     #TEMP1,ADR1     ;
1443  016476  012767  001012  162300        MOV     #TEMP2,ADR2     ;
1444  016504  012700  001000                MOV     #ADR,R0         ; LOAD ADDRESSES OF ADDRESSES
1445  016510  012701  001002                MOV     #ADR1,R1        ;
1446  016514  012702  001004                MOV     #ADR2,R2        ;
1447  016520  112770  000252  000000        MOVB    #252,@0(R0)     ; LOAD TEMP
1448  016526  117270  177774  000002        MOVB    @-4(R2),@2(R0)  ; TEMP => TEMP1
1449  016534  122767  000252  162246        CMPB    #252,TEMP1      ; CHECK IT
1450  016542  001402                        BEQ     1$              ; OK, CONTINUE
1451  016544  104423  016452                ERROR,  T42             ;; *** MODE 7 IS FAILING ***
1452  016550  012770  125252  000000  1$:   MOV     #125252,@0(R0)  ; LOAD TEMP
1453  016556  017270  177774  000002        MOV     @-4(R2),@2(R0)  ; TEMP => TEMP1
1454  016564  022767  125252  162216        CMP     #125252,TEMP1   ; CHECK IT
1455  016572  001402                        BEQ     2$              ; OK, CONTINUE
1456  016574  104423  016452                ERROR,  T42             ;; *** INSTRUCTIONS FAILED IN MODE 7 ***
1457
1458  016600                          2$:
1459                                  ;*******************************************************************************
  (2)                             .SBTTL      T43 -- TSTB, CLRB, AND MOVB
  (2)                             ;*******************************************************************************
  (2) 016600  104420  000043      T43:    BEGINTEST, 43           ;; CHECK SWITCH REGISTER OPTIONS.
1460  016604  012700  001006      TSTB1:  MOV     #TEMP,R0        ; LOAD ADDRESSES
1461  016610  012701  001010              MOV     #TEMP1,R1       ;
1462  016614  000277                       SCC
1463  016616  105010                       CLRB    (R0)            ; CLEAR THE LOCATION
1464  016620  104404                       CHECKCC+4               ; CHECK FOR CC = 4
1465  016622  104    016600                ERROR,  T43             ;; *** INCORRECT CC BITS ***
1466  016626  105 10                       TSTB    (R0)            ; CHECK IT
1467  016630  104 04                       CHECKCC+4               ; CHECK FOR CC = 4
1468  016632  104423  016600               ERROR,  T43             ;; *** INCORRECT CC BITS ***
1469  016636  112711  000377              MOVB    #377,(R1)       ; LOAD THE LOCATION
1470  016642  104410                       CHECKCC+10              ; CHECK FOR CC = 10
1471  016644  104423  016600               ERROR,  T43             ;; *** INCORRECT CC BITS ***
```

```
1472  016650  105711                          TSTB    (R1)            ; CHECK IT
1473  016652  104410                          CHECKCC+10              ; CHECK FOR CC = 10
1474  016654  104423  016600                  ERROR,  T43             ;; *** INCORRECT CC BITS ***
1475  016660  010002                          MOV     R0,R2           ; R2 IS NOW POINTING TO LOCATION TEMP
1476  016662  112762  000200  000000          MOVB    #200,0(R2)      ; PLACE #200 IN LOCATION TEMP
1477  016670  112241                          MOVB    (R2)+,-(R1)     ; MOVE #200 TO LOCATION TEMP+1
1478  016672  026127  177777  100200          CMP     -1(R1),#100200  ; CHECK THE DATA IN LOCATION TEMP
1479  016700  001402                          BEQ     4$
1480  016702  104423  016600                  ERROR,  T43             ;; *** MOVB INSTRUCTION FAILED ***
1481  016706  020102                  4$:     CMP     R1,R2           ; CHECK THE REGISTERS FOR PROPER VALUE
1482  016710  001402                          BEQ     5$
1483  016712  104423  016600                  ERROR,  T43             ;; *** "    INSTRUCTION FAILED ***
1484  016716                          5$:
1485
1486                                  ;*******************************************************************************
 (2)                                .SBTTL      T44 -- CMPB AND BISB
 (2)                                ;*******************************************************************************
 (2)  016716  104420  000044          T44:    BEGINTEST, 44           ;; CHECK SWITCH REGISTER OPTIONS.
1487  016722  012701  001012          CMPB1:  MOV     #TEMP2,R1
1488  016726  012702  001006                  MOV     #TEMP,R2        ; LOAD ADDRESS
1489  016732  012711  000077                  MOV     #77,(R1)        ; PLACE 77 IN LOCATION TEMP2
1490  016736  112704  000377                  MOVB    #377,R4         ; R4 SHOULD CONTAIN #177777
1491  016742  150412                          BISB    R4,(R2)         ; LOAD LOCATION
1492  016744  104410                          CHECKCC+10              ; CHECK FOR CC = 10
1493  016746  104423  016716                  ERROR,  T44             ;; *** INCORRECT CC BITS ***
1494  016752  120412                          CMPB    R4,(R2)         ; CHECK COMPARE
1495  016754  001402                          BEQ     2$              ; CONTINUE IF OK
1496  016756  104423  016716                  ERROR,  T44             ;; *** BISB OR CMPB INSTRUCTION FAILED ***
1497  016762  121112                  2$:     CMPB    (R1),(R2)       ; CHECK IT AGAIN
1498  016764  100002                          BPL     3$              ; CONTINUE IF OK
1499  016766  104423  016716                  ERROR,  T44             ;; *** CMPB INSTRUCTION FAILED (WRONG CC) ***
1500  016772  121211                  3$:     CMPB    (R2),(R1)       ; ONCE MORE
1501  016774  100402                          BMI     4$              ; CONTINUE IF OK
1502  016776  104423  016716                  ERROR,  T44             ;; *** INSTR FAILED ***
1503  017002                          4$:
1504
1505                                  ;*******************************************************************************
 (2)                                .SBTTL      T45 -- BICB AND BITB
 (2)                                ;*******************************************************************************
 (2)  017002  104420  000045          T45:    BEGINTEST, 45           ;; CHECK SWITCH REGISTER OPTIONS.
1506  017006  012703  001006          BICB1:  MOV     #TEMP,R3        ; LOAD ADDRESS
1507  017012  112713  000377                  MOVB    #377,(R3)       ; LOAD LOCATION
1508  017016  012700  001010                  MOV     #TEMP1,R0       ;PLACE THE ADDRESS OF TEMP1 IN R0
1509  017022  010001                          MOV     R0,R1           ;AND R1
1510  017024  112721  000252                  MOVB    #252,(R1)+      ; PLACE #252 IN TEMP1
1511  017030  000277                          SCC
1512  017032  146013  000000                  BICB    0(R0),(R3)      ; CLEAR EVERY OTHER BIT
1513  017036  104401                          CHECKCC+1               ; CHECK FOR CC = 1
1514  017040  104423  017002                  ERROR,  T45             ;; *** INCORRECT CC BITS ***
1515  017044  136113  177777                  BITB    -1(R1),(R3)     ; CHECK IT
1516  017050  001402                          BEQ     4$              ; CONTINUE IF OK
1517  017052  104423  017002                  ERROR,  T45             ;; *** BICB OR BITB INSTRUCTION FAILED ***
1518  017056  132713  000125          4$:     BITB    #125,(R3)       ; CHECK IT
1519  017062  104401                          CHECKCC+1               ; CHECK FOR CC = 1
1520  017064  104423  017002                  ERROR,  T45             ;; *** INCORRECT CC BITS ***
1521  017070  154113                          BISB    -(R1),(R3)      ; SET THE BITS THAT WERE CLEARED
```

```
1522  017072  100402                        BMI     6$              ; CONTINUE IF OK
1523  017074  104423  017002                ERROR,  T45             ;; *** BITB OR BISB INSTRUCTION FAILED ***
1524  017100  012746  000177        6$:     MOV     #177,-(SP)      ; STORE #177 ON THE STACK
1525  017104  142613                        BICB    (SP)+,(R3)      ; CLEAR ALL THE BITS EXCEPT SIGN BIT
1526  017106  104411                        CHECKCC+11              ; CHECK FOR CC = 11
1527  017110  104423  017002                ERROR,  T45             ;; *** INCORRECT CC BITS ***
1528  017114  132713  000377                BITB    #377,(R3)       ; CHECK IT
1529  017120  104411                        CHECKCC+11              ; CHECK FOR CC = 11
1530  017122  104423  017002                ERROR,  T45             ;; *** INCORRECT CC BITS ***
1531  017126  010300                        MOV     R3,R0           ; PLACE THE ADDRESS OF TEMP IN R0
1532  017130  012710  001010                MOV     #TEMP1,(R0)     ; PLACE THE ADDRESS OF TEMP1 IN TEMP
1533  017134  012730  000377                MOV     #377,@(R0)+     ; WRITE A 377 IN LOCATION TEMP1
1534  017140  000263                        SEVC                    ; SET V & C BITS
1535  017142  145070  000000                BICB    @-(R0),@(R0)    ; BIT CLEAR THE CONTENTS
1536                                                                ; OF TEMP1 TO THE CONTENTS OF TEMP1
1537  017146  104405                        CHECKCC+5               ; CHECK FOR CC = 5
1538  017150  104423  017002                ERROR,  T45             ;; *** INCORRECT CC BITS ***
1539  017154  022027  001010                CMP     (R0)+,#TEMP1    ; MAKE SURE THAT (R0) IS POINTING TO LOCATION TEMP1
1540  017160  001402                        BEQ     8$
1541  017162  104423  017002                ERROR,  T45             ;; *** BICB OR CMP INSTRUCTION FAILED IN THE SPECIFIC MO
1542  017166  005750                8$:     TST     @-(R0)          ; TEST LOCATION TEMP1
1543  017170  001402                        BEQ     10$
1544  017172  104423  017002                ERROR,  T45             ;; *** BICB INSTRUCTION FAILED ***
1545  017176  000257                10$:    CCC
1546  017200  141010                        BICB    (R0),(R0)       ; CLEAR THE LOCATION TEMP
1547  017202  104404                        CHECKCC+4               ; CHECK FOR CC = 4
1548  017204  104423  017002                ERROR,  T45             ;; *** INCORRECT CC BITS ***
1549
1550                                ;*******************************************************************************
 (2)                                .SBTTL      T46 -- INCB AND DECB
 (2)                                ;*******************************************************************************
 (2)  017210  104420  000046        T46:    BEGINTEST, 46           ;; CHECK SWITCH REGISTER OPTIONS.
1551  017214  012704  001006        INCB1:  MOV     #TEMP,R4        ; LOAD ADDRESS
1552  017220  112714  000177                MOVB    #177,(R4)       ; TEMP LOCATION=177
1553  017224  000261                        SEC
1554  017226  105214                        INCB    (R4)            ; ADD ONES INTO LOCATION
1555  017230  104413                        CHECKCC+13              ; CHECK FOR CC = 13
1556  017232  104423  017210                ERROR,  T46             ;; *** INCORRECT CC BITS ***
1557  017236  012714  000376                MOV     #376,(R4)
1558  017242  105224                        INCB    (R4)+           ;
1559  017244  104411                        CHECKCC+11              ; CHECK FOR CC = 11
1560  017246  104423  017210                ERROR,  T46             ;; *** INCORRECT CC BITS ***
1561  017252  105744                        TSTB    -(R4)           ; DECREMENT R4 BY 1
1562  017254  005746                        TST     -(SP)           ; AND SP BY 2
1563  017256  010426                        MOV     R4,(SP)+        ; PLACE THE ADDRESS OF TEMP ON THE STACK
1564  017260  000241                        CLC                     ; CLEAR C BIT
1565  017262  105256                        INCB    @-(SP)          ; INCREMENT THE CONTENTS OF LOCATION TEMP
1566  017264  104404                        CHECKCC+4               ; CHECK FOR CC = 4
1567  017266  104423  017210                ERROR,  T46             ;; *** INCORRECT CC BITS ***
1568  017272  123634                        CMPB    @(SP)+,@(R4)+   ; RESTORE STACK POINTER
1569  017274  000261                        SEC                     ; SET C BIT
1570  017276  105264  177777                INCB    -1(R4)
1571  017302  104401                        CHECKCC+1               ; CHECK FOR CC = 1
1572  017304  104423  017210                ERROR,  T46             ;; *** INCORRECT CC BITS ***
1573  017310  124427  000001                CMPB    -(R4),#1        ; CHECK IT
1574  017314  001402                        BEQ     2$              ; CONTINUE IF OK
```

```
 1575  017316  104423  017210                 ERROR,  T46             ;; *** INCB INSTRUCTION FAILED ***
 1576  017322  000261                 2$:     SEC
 1577  017324  105314                         DECB    (R4)            ; SUBTRACT ONES FROM LOCATION
 1578  017326  104405                         CHECKCC+5               ; CHECK FOR CC = 5
 1579  017330  104423  017210                 ERROR,  T46             ;; *** INCORRECT CC BITS ***
 1580  017334  105324                         DECB    (R4)+
 1581  017336  104411                         CHECKCC+11              ; CHECK FOR CC = 11
 1582  017340  104423  017210                 ERROR,  T46             ;; *** INCORRECT CC BITS ***
 1583  017344  112764  000200  177777         MOVB    #200,-1(R4)
 1584  017352  105344                         DECB    -(R4)           ;
 1585  017354  104403                         CHECKCC+3               ; CHECK CC = 3
 1586  017356  104423  017210                 ERROR,  T46             ;; *** INCORRECT CC BITS ***
 1587  017362  105364  000000                 DECB    0(R4)           ;
 1588  017366  104401                         CHECKCC+1               ; CHECK FOR CC = 1
 1589  017370  104423  017210                 ERROR,  T46             ;; *** INCORRECT CC BITS ***
 1590  017374  126427  000000  000176         CMPB    0(R4),#176
 1591  017402  001402                         BEQ     3$
 1592  017404  104423  017210                 ERROR,  T46             ;; *** DECB INSTRUCTION FAILED ***
 1593  017410                   3$:
 1594
 1595                                 ;*******************************************************************************
  (2)                                .SBTTL      T47 -- COMB AND NEGB
  (2)                                ;*******************************************************************************
  (2)  017410  104420  000047         T47:    BEGINTEST, 47           ;; CHECK SWITCH REGISTER OPTIONS.
 1596  017414  012703  001006         COMB1:  MOV     #TEMP,R3        ; LOAD ADDRESS
 1597  017420  012704  001010                 MOV     #TEMP1,R4
 1598  017424  012714  000252                 MOV     #252,(R4)
 1599  017430  112413                         MOVB    (R4)+,(R3)      ; LOAD EVERY OTHER BIT
 1600  017432  000277                         SCC
 1601  017434  105113                         COMB    (R3)            ; 1'S COMPLEMENT
 1602  017436  104401                         CHECKCC+1               ; CHECK FOR CC = 1
 1603  017440  104423  017410                 ERROR,  T47             ;; *** INCORRECT CC BITS ***
 1604  017444  122713  000125                 CMPB    #125,(R3)       ; CHECK IT
 1605  017450  001402                         BEQ     2$              ; CONTINUE IF OK
 1606  017452  104423  017410                 ERROR,  T47             ;; *** COMB INSTRUCTION FAILED ***
 1607  017456  000277                 2$:     SCC
 1608  017460  105113                         COMB    (R3)            ; COMPLEMENT BACK
 1609  017462  104411                         CHECKCC+11              ; CHECK FOR CC = 11
 1610  017464  104423  017410                 ERROR,  T47             ;; *** INCORRECT CC BITS ***
 1611  017470  010400                         MOV     R4,R0
 1612  017472  126013  177777                 CMPB    -1(R0),(R3)     ; CHECK IT
 1613  017476  001402                         BEQ     3$              ; CONTINUE IF OK
 1614  017500  104423  017410                 ERROR,  T47             ;; *** COMB INSTRUCTION FAILED ***
 1615  017504  112724  000377         3$:     MOVB    #377,(R4)+
 1616  017510  114413                         MOVB    -(R4),(R3)      ; PLACE #377 IN (R3)
 1617  017512  000277                         SCC
 1618  017514  105113                         COMB    (R3)
 1619  017516  104405                         CHECKCC+5               ; CHECK FOR CC = 5
 1620  017520  104423  017410                 ERROR,  T47             ;; *** INCORRECT CC BITS ***
 1621
 1622  017524  012700  001006         NEGB1:  MOV     #TEMP,R0        ; LOAD ADDRESS
 1623  017530  112710  000001                 MOVB    #1,(R0)         ; LOAD THE LOCATION
 1624  017534  105410                         NEGB    (R0)            ; 2'S COMPLEMENT
 1625  017536  104411                         CHECKCC+11              ; CHECK FOR CC = 11
 1626  017540  104423  017524                 ERROR,  NEGB1           ;; *** INCORRECT CC BITS ***
 1627  017544  122710  000377                 CMPB    #377,(R0)       ; CHECK IT
```

```
1628  017550  001402                  BEQ     2$              ; CONTINUE IF OK
1629  017552  104423  017524          ERROR,  NEGB1           ;; *** NEGB INSTRUCTION FAILED ***
1630  017556  012710  000200  2$:     MOV     #200,(R0)
1631  017562  105410                  NEGB    (R0)            ; 2'S COMPLEMENT
1632  017564  104413                  CHECKCC+13              ; CHECK FOR CC = 13
1633  017566  104423  017524          ERROR,  NEGB1           ;; *** INCORRECT CC BITS ***
1634  017572  122710  000200          CMPB    #200,(R0)       ; CHECK IT
1635  017576  001402                  BEQ     3$              ; CONTINUE IF OK
1636  017600  104423  017524          ERROR,  NEGB1           ;; *** NEGB FAILED. ***
1637  017604                  3$:
1638
1639                          ;*******************************************************************************
 (2)                          .SBTTL      T50 -- ROLB AND RORB
 (2)                          ;*******************************************************************************
 (2)  017604  104420  000050  T50:    BEGINTEST, 50           ;; CHECK SWITCH REGISTER OPTIONS.
1640  017610  012701  001010  ROLB1.  MOV     #TEMP1,R1       ; LOAD ADDRESS
1641  017614  112711  000040          MOVB    #40,(R1)        ; LOAD LOCATION
1642  017620  000257                  CCC                     ; CLEAR FLAGS
1643  017622  106111                  ROLB    (R1)            ; SHIFT
1644  017624  106111                  ROLB    (R1)            ;
1645  017626  104412                  CHECKCC+12              ; CHECK FOR CC = 12
1646  017630  104423  017604          ERROR,  T50             ;; *** INCORRECT CC BITS ***
1647  017634  122711  000200          CMPB    #200,(R1)       ; CHECK IT
1648  017640  001402                  BEQ     1$              ; CONTINUE IF OK
1649  017642  104423  017604          ERROR,  T50             ;; *** ROLB INSTRUCTION FAILED ***
1650  017646  106111          1$:     ROLB    (R1)            ; SHIFT
1651  017650  104407                  CHECKCC+7               ; CHECK FOR CC = 7
1652  017652  104423  017604          ERROR,  T50             ;; *** INCORRECT CC BITS ***
1653  017656  106111                  ROLB    (R1)            ; SHIFT
1654  017660  122711  000001          CMPB    #1,(R1)         ; CHECK IT
1655  017664  001402                  BEQ     RORB1           ; CONTINUE IF OK
1656  017666  104423  017604          ERROR,  T50             ;; *** ROLB FAILED ***
1657
1658  017672  012702  001010  RORB1:  MOV     #TEMP1,R2       ; LOAD ADDRESS
1659  017676  112712  000004          MOVB    #4,(R2)         ; LOAD LOCATION
1660  017702  000257                  CCC                     ; CLEAR FLAGS
1661  017704  106012                  RORB    (R2)            ; SHIFT
1662  017706  106012                  RORB    (R2)            ;
1663  017710  122712  000001          CMPB    #1,(R2)         ; CHECK IT
1664  017714  001402                  BEQ     1$              ; CONTINUE IF OK
1665  017716  104423  017672          ERROR,  RORB1           ;; *** RORB INSTRUCTION FAILED ***
1666  017722  106012          1$:     RORB    (R2)            ; SHIFT
1667  017724  104407                  CHECKCC+7               ; CHECK FOR CC = 7
1668  017726  104423  017672          ERROR,  RORB1           ;; *** INCORRECT CC BITS ***
1669  017732  106012                  RORB    (R2)            ; SHIFT
1670  017734  104412                  CHECKCC+12              ; CHECK FOR CC = 12
1671  017736  104423  017672          ERROR,  RORB1           ;; *** INCORRECT CC BITS ***
1672  017742  122712  000200          CMPB    #200,(R2)       ; CHECK IT
1673  017746  001402                  BEQ     2$              ; CONTINUE IF OK
1674  017750  104423  017672          ERROR,  RORB1           ;; *** RORB FAILED ***
1675  017754                  2$:
1676
1677                          ;*******************************************************************************
 (2)                          .SBTTL      T51 -- ASLB AND ASRB
 (2)                          ;*******************************************************************************
 (2)  017754  104420  000051  T51:    BEGINTEST, 51           ;; CHECK SWITCH REGISTER OPTIONS.
```

```
 1678  017760  012703  001010          ASLB1:  MOV     #TEMP1,R3       ; LOAD ADDRESS
 1679  017764  112713  000040                  MOVB    #40,(R3)        ; LOAD LOCATION
 1680  017770  000257                          CCC                     ; CLEAR FLAGS
 1681  017772  106313                          ASLB    (R3)            ; SHIFT
 1682  017774  106313                          ASLB    (R3)            ;
 1683  017776  104412                          CHECKCC+12              ; CHECK FOR CC = 12
 1684  020000  104423  017754                  ERROR,  T51             ;; *** INCORRECT CC BITS ***
 1685  020004  122713  000200                  CMPB    #200,(R3)       ; CHECK IT
 1686  020010  001402                          BEQ     4$              ; CONTINUE IF OK
 1687  020012  104423  017754                  ERROR,  T51             ;; *** ASLB INSTRUCTION FAILED ***
 1688  020016  106313                  4$:     ASLB    (R3)            ; SHIFT
 1689  020020  104407                          CHECKCC+7               ; CHECK FOR CC = 7
 1690  020022  104423  017754                  ERROR,  T51             ;; *** INCORRECT CC BITS ***
 1691  020026  106313                          ASLB    (R3)            ; SHIFT
 1692  020030  104404                          CHECKCC+4               ; CHECK FOR CC = 4
 1693  020032  104423  017754                  ERROR,  T51             ;; *** INCORRECT CC BITS ***
 1694
 1695  020036  012704  001010          ASRB1:  MOV     #TEMP1,R4       ; LOAD ADDRESSES
 1696  020042  012703  001012                  MOV     #TEMP2,R3       ;
 1697  020046  112714  000004                  MOVB    #4,(R4)         ; LOAD LOCATION
 1698  020052  000257                          CCC                     ; CLEAR FLAGS
 1699  020054  106214                          ASRB    (R4)            ; SHIFT
 1700  020056  106214                          ASRB    (R4)            ;
 1701  020060  122714  000001                  CMPB    #1,(R4)         ; CHECK IT
 1702  020064  001402                          BEQ     2$              ; CONTINUE IF OK
 1703  020066  104423  020036                  ERROR,  ASRB1           ;; *** ASRB INSTRUCTION FAILED ***
 1704  020072  106214                  2$:     ASRB    (R4)            ; SHIFT
 1705  020074  104407                          CHECKCC+7               ; CHECK FOR CC = 7
 1706  020076  104423  020036                  ERROR,  ASRB1           ;; *** INCORRECT CC BITS ***
 1707  020102  106214                          ASRB    (R4)            ; SHIFT
 1708  020104  104404                          CHECKCC+4               ; CHECK FOR CC = 4
 1709  020106  104423  020036                  ERROR,  ASRB1           ;; *** INCORRECT CC BITS ***
 1710  020112  112713  000202                  MOVB    #202,(R3)       ; LOAD LOCATION
 1711  020116  106213                          ASRB    (R3)            ; SHIFT
 1712  020120  106213                          ASRB    (R3)            ;
 1713  020122  104411                          CHECKCC+11              ; CHECK FOR CC = 11
 1714  020124  104423  020036                  ERROR,  ASRB1           ;; *** INCORRECT CC BITS ***
 1715  020130  122713  000340                  CMPB    #340,(R3)       ; CHECK IT
 1716  020134  001402                          BEQ     3$              ; CONTINUE IF OK
 1717  020136  104423  020036                  ERROR,  ASRB1           ;; *** ASRB FAILED ***
 1718  020142                          3$:
 1719
 1720                                  ;*******************************************************************************
  (2)                                  .SBTTL      152 -- ADCB AND SBCB
  (2)                                  ;*******************************************************************************
  (2)  020142  104420  000052          T52:    BEGINTEST, 52           ;; CHECK SWITCH REGISTER OPTIONS.
 1721  020146  012700  001012          ADCB1:  MOV     #TEMP2,R0       ; LOAD ADDRESS
 1722  020152  105010                          CLRB    (R0)            ; CLEAR THE LOCATION
 1723  020154  000257                          CCC                     ; CLEAR FLAGS
 1724  020156  105510                          ADCB    (R0)            ; ADD C BIT = 0
 1725  020160  104404                          CHECKCC+4               ; CHECK FOR CC = 4
 1726  020162  104423  020142                  ERROR,  T52             ;; *** INCORRECT CC BITS ***
 1727  020166  000261                          SEC                     ; C=1
 1728  020170  105510                          ADCB    (R0)            ; ADD C BIT=1
 1729  020172  000261                          SEC                     ; C=1
 1730  020174  105510                          ADCB    (R0)            ; AGAIN
```

```
 1731  020176  104400                          CHECKCC+0           ; CHECK FOR CC = 0
 1732  020200  104423  020142                  ERROR,  T52         ;; *** INCORRECT CC BITS ***
 1733  020204  122710  000002                  CMPB    #2,(R0)     ; CHECK IT
 1734  020210  001402                          BEQ     4$          ; CONTINUE IF OK
 1735  020212  104423  020142                  ERROR,  T52         ;; *** ADCB INSTRUCTION FAILED ***
 1736  020216  112710  000177          4$:     MOVB    #'77,(R0)   ; LOAD LARGEST POSITVE BYTE
 1737  020222  000261                          SEC                 ; C=1
 1738  020224  105510                          ADCB    (R0)        ; ADD C BIT=1
 1739  020226  104412                          CHECKCC+12          ; CHECK FOR CC = 12
 1740  020230  104423  020142                  ERROR,  T52         ;; *** INCORRECT CC BITS ***
 1741  020234  122710  000200                  CMPB    #200,(R0)   ; CHECK IT
 1742  020240  001402                          BEQ     6$          ; CONTINUE IF OK
 1743  020242  104423  020142                  ERROR,  T52         ;; *** ADCB INSTRUCTION FAILED ***
 1744  020246  112710  000377          6$:     MOVB    #377,(R0)   ; LOAD -1
 1745  020252  000261                          SEC                 ; C=1
 1746  020254  105510                          ADCB    (R0)        ; ADD C BIT=1
 1747  020256  104405                          CHECKCC+5           ; CHECK FOR CC = 5
 1748  020260  104423  020142                  ERROR,  T52         ;; *** INCORRECT CC BITS ***
 1749
 1750  020264  012701  001012          SBCB1:  MOV     #TEMP2,R1   ; LOAD ADDRESS
 1751  020270  112711  000003                  MOVB    #3,(R1)     ; LOAD LOCATION
 1752  020274  000257                          CCC                 ; CLEAR FLAGS
 1753  020276  105611                          SBCB    (R1)        ; SUBTRACT C BIT=0
 1754  020300  104400                          CHECKCC+0           ; CHECK FOR CC = 0
 1755  020302  104423  020264                  ERROR,  SBCB1       ;; *** INCORRECT CC BITS ***
 1756  020306  122711  000003                  CMPB    #3,(R1)     ; CHECK IT
 1757  020312  001402                          BEQ     2$          ; CONTINUE IF OK
 1758  020314  104423  020264                  ERROR,  SBCB1       ;; *** SBCB INSTRUCTION FAILED ***
 1759  020320  000261                  2$:     SEC                 ; C=1
 1760  020322  105611                          SBCB    (R1)        ; SUBTRACT C BIT=1
 1761  020324  000261                          SEC                 ; C=1
 1762  020326  105611                          SBCB    (R1)        ;
 1763  020330  104400                          CHECKCC+0           ; CHECK FOR CC = 0
 1764  020332  104423  020264                  ERROR,  SBCB1       ;; *** INCORRECT CC BITS ***
 1765  020336  122711  000001                  CMPB    #1,(R1)     ; CHECK IT
 1766  020342  001402                          BEQ     3$          ; CONTINUE IF OK
 1767  020344  104423  020264                  ERROR,  SBCB1       ;; *** SBCB INSTRUCTION FAILED ***
 1768  020350  000261                  3$:     SEC                 ; C=1
 1769  020352  105611                          SBCB    (R1)        ; SUBTRACT C BIT=1
 1770  020354  104404                          CHECKCC+4           ; CHECK FOR CC = 4
 1771  020356  104423  020264                  ERROR,  SBCB1       ;; *** INCORRECT CC BITS ***
 1772  020362  000261                          SEC                 ; C=1
 1773  020364  105611                          SBCB    (R1)        ; SUBTRACT C BIT = 1
 1774  020366  104411                          CHECKCC+11          ; CHECK FOR CC = 11
 1775  020370  104423  020264                  ERROR,  SBCB1       ;; *** INCORRECT CC BITS ***
 1776  020374  122711  000377                  CMPB    #377,(R1)   ; CHECK IT
 1777  020400  001402                          BEQ     4$          ; CONTINUE IF OK
 1778  020402  104423  020264                  ERROR,  SBCB1       ;; *** SBCB INSTRUCTION FAILED ***
 1779  020406  112711  000200          4$:     MOVB    #200,(R1)   ; LOAD R1
 1780  020412  000261                          SEC                 ; C=1
 1781  020414  105611                          SBCB    (R1)        ; SUBTRACT C BIT = 1
 1782  020416  104402                          CHECKCC+2           ; CHECK FOR CC = 2
 1783  020420  104423  020264                  ERROR,  SBCB1       ;; *** INCORRECT CC BITS ***
 1784
 1785                          ;*******************************************************************
  (2)                          .SBTTL  T53 -- TST, CLR, AND MOV
```

```
  (2)                                   ;*****************************************************************************
  (2)  020424  104420  000053           T53:    BEGINTEST, 53           ;; CHECK SWITCH REGISTER OPTIONS.
1786   020430  012701  001006           MOV1:   MOV     #TEMP,R1        ; LOAD ADDRESSES
1787   020434  012700  001010                   MOV     #TEMP1,R0       ;
1788   020440  000277                           SCC
1789   020442  005010                           CLR     (R0)            ; CLEAR THE LOCATION
1790   020444  104404                           CHECKCC+4               ; CHECK FOR CC = 4
1791   020446  104423  020424                   ERROR,  T53             ;; *** INCORRECT CC BITS ***
1792   020452  005720                           TST     (R0)+           ; CHECK IT
1793   020454  104404                           CHECKCC+4               ; CHECK FOR CC = 4
1794   020456  104423  020424                   ERROR,  T53             ;; *** INCORRECT CC BITS ***
1795   020462  010040                           MOV     R0,-(R0)
1796   020464  012730  177777                   MOV     #177777,@(R0)+
1797   020470  016011  177776                   MOV     -2(R0),(R1)     ; LOAD THE LOCATION.
1798   020474  104410                           CHECKCC+10              ; CHECK FOR CC = 10
1799   020476  104423  020424                   ERROR,  T53             ;; *** INCORRECT CC BITS ***
1800   020502  005711                           TST     (R1)            ; CHECK IT
1801   020504  104410                           CHECKCC+10              ; CHECK FOR CC = 10
1802   020506  104423  020424                   ERROR,  T53             ;; *** INCORRECT CC BITS ***
1803
1804                                   ;*****************************************************************************
  (2)                                   .SBTTL      T54 -- CMP AND BIS
  (2)                                   ;*****************************************************************************
  (2)  020512  104420  000054           T54:    BEGINTEST, 54           ;; CHECK SWITCH REGISTER OPTIONS.
1805   020516  012702  001010           CMP1:   MOV     #TEMP1,R2       ; LOAD ADDRESS
1806   020522  012700  001006                   MOV     #TEMP,R0        ; PLACE THE ADDRESS OF TEMP IN R0
1807   020526  000257                           CCC                     ; CLEAR CC BITS.
1808   020530  012720  177777                   MOV     #177777,(R0)+   ; PLACE #177777 IN TEMP AND INC R0 BY 2
1809   020534  054012                           BIS     -(R0),(R2)      ; LOAD LOCATION
1810   020536  104410                           CHECKCC+10              ; CHECK FOR CC = 10
1811   020540  104423  020512                   ERROR,  T54             ;; *** INCORRECT CC BITS ***
1812   020544  022227  177777                   CMP     (R2)+,#177777   ; CHECK COMPARE
1813   020550  001402                           BEQ     2$              ; CONTINUE IF OK
1814   020552  104423  020512                   ERROR,  T54             ;; *** CMP OR BIS INSTRUCTION FAILED ***
1815   020556  020227  001012           2$:     CMP     R2,#TEMP1+2     ; CHECK R2 TO CONTAIN ADDRESS OF TEMP1+2
1816   020562  001402                           BEQ     3$
1817   020564  104423  020512                   ERROR,  T54             ;; *** NO AUTO INCREMENT ***
1818   020570  022742  000077           3$:     CMP     #77,-(R2)       ; CHECK IT AGAIN
1819   020574  104401                           CHECKCC+1               ; CHECK FOR CC = 1
1820   020576  104423  020512                   ERROR,  T54             ;; *** INCORRECT CC BITS ***
1821   020602  022722  077777                   CMP     #77777,(R2)+
1822   020606  104413                           CHECKCC+13              ; CHECK FOR CC = 13
1823   020610  104423  020512                   ERROR,  T54             ;; *** INCORRECT CC BITS ***
1824   020614  022227  077777                   CMP     -(R2),#77777    ; ONCE MORE
1825   020620  104410                           CHECKCC+10              ; CHECK FOR CC = 10
1826   020622  104423  020512                   ERROR,  T54             ;; *** INCORRECT CC BITS ***
1827   020626  012767  052525  160156           MOV     #52525,TEMP2    ; SET EVERY OTHER BIT IN TEMP2
1828   020634  012767  001012  160146           MOV     #TEMP2,TEMP1    ; PLACE THE ADDRESS OF TEMP2 IN TEMP1
1829   020642  012704  001000                   MOV     #ADR,R4
1830   020646  012714  001002                   MOV     #ADR1,(R4)      ; PLACE THE ADDRESS OF ADR1 IN ADR POINTED BY R4
1831   020652  012734  125252                   MOV     #125252,@(R4)+  ; PLACE THE #125252 IN LOCATION ADR1
1832   020656  057432  177776                   BIS     @-2(R4),@(R2)+  ; SET EVERY OTHER BIT AT LOCATION TEMP2
1833                                                                    ; AND INCREMENT R2 BY 2
1834   020662  010200                           MOV     R2,R0           ; PLACE ADDRESS OF TEMP2 IN R0
1835   020664  025027  177777                   CMP     @-(R0),#177777  ; TEMP2 SHOULD CONTAIN ALL 1'S
1836   020670  001402                           BEQ     4$
```

```
 1837  020672  104423  020512                 ERROR,  T54             ;; *** CMP OR BIS INSTUCTIONS FAILED IN MODES OTHER THAN
 1838  020676  020227  001012          4$:    CMP     R2,#TEMP1+2     ; R2 SHOULD CONTAIN THE ADDRESS OF TEMP2
 1839  020702  001402                         BEQ     5$
 1840  020704  104423  020512                 ERROR,  T54             ;; *** MODE 5 IS FAILING ***
 1841  020710  005040                   5$:   CLR     -(R0)           ; PLACE A 0 IN TEMP
 1842  020712  010067  160074                 MOV     R0,TEMP2        ; PLACE ADDRESS OF TEMP IN TEMP2
 1843  020716  022020                         CMP     (R0)+,(R0)+     ; BUMP R0 BY 4
 1844  020720  055070  000002                 BIS     @-(R0),@2(R0)   ; PLACE THE CONTENTS OF TEMP2 AT TEMP
 1845  020724  022767  001006  160054         CMP     #TEMP,TEMP      ; TEMP SHOULD CONTAIN ITS OWN ADDRESS
 1846  020732  001402                         BEQ     6$
 1847  020734  104423  020512                 ERROR,  T54             ;; *** CMP OR BIS INSTRUCTIONS FAILED ***
 1848  020740                           6$:
 1849
 1850                                  ;*******************************************************************************
   (2)                                 .SBTTL      T55 -- BIC AND BIT
   (2)                                 ;*******************************************************************************
   (2)  020740  104420  000055          T55:    BEGINTEST, 55           ;; CHECK SWITCH REGISTER OPTIONS.
 1851  020744  012703  001006          BIC1:   MOV     #TEMP,R3        ; LOAD ADDRESS
 1852  020750  012713  177777                  MOV     #177777,(R3)    ; LOAD LOCATION
 1853  020754  012704  001000                  MOV     #ADR,R4         ; PLACE THE ADDRESS OF ADR IN R4
 1854  020760  012714  001002                  MOV     #ADR1,(R4)      ; PLACE THE ADDRESS OF ADR1 IN ADR
 1855  020764  011334                          MOV     (R3),@(R4)+     ; LOAD LOCATION ADR1 WITH #177777
 1856  020766  012700  001010                  MOV     #TEMP1,R0       ; PLACE THE ADDRESS OF TEMP1 IN R0
 1857  020772  012710  125252                  MOV     #125252,(R0)    ; SET EVERY OTHER BIT AT LOCATION TEMP1
 1858  020776  000277                          SCC
 1859  021000  042013                          BIC     (R0)+,(R3)      ; CLEAR EVERY OTHER BIT
 1860  021002  104401                          CHECKCC+1               ; CHECK FOR CC = 1
 1861  021004  104423  020740                  ERROR,  T55             ;; *** INCORRECT CC BITS ***
 1962  021010  034013                          BIT     -(R0),(R3)      ; CHECK IT
 1863  021012  001402                          BEQ     1$              ; CONTINUE IF OK
 1864  021014  104423  020740                  ERROR,  T55             ;; *** BIC OR BIT INSTRUCTION FAILED ***
 1865  021020  032713  052525           1$:    BIT     #52525,(R3)     ; CHECK IT
 1866  021024  104401                          CHECKCC+1               ; CHECK FOR CC = 1
 1867  021026  104423  020740                  ERROR,  T55             ;; *** INCORRECT CC BITS ***
 1868  021032  056013  000000                  BIS     0(R0),(R3)      ; SET THE BITS THAT WERE CLEARED
 1869  021036  100402                          BMI     2$              ; CONTINUE IF OK
 1870  021040  104423  020740                  ERROR,  T55             ;; *** BIT OR BIS INSTRUCTION FAILED ***
 1871  021044  012720  077777           2$:    MOV     #77777,(R0)+    ; SET ALL THE BITS AT TEMP1 EXCEPT SIGN
 1872  021050  010002                          MOV     R0,R2
 1873  021052  046213  177776                  BIC     -2(R2),(R3)     ; TRY CLEARING THE OTHER BITS
 1874  021056  104411                          CHECKCC+11              ; CHECK FOR CC = 11
 1875  021060  104423  020740                  ERROR,  T55             ;; *** INCORRECT CC BITS ***
 1876  021064  020027  001012                  CMP     R0,#TEMP1+2     ; R0 SHOULD CONTAIN THE ADDRESS OF TEMP1+2
 1877  021070  001402                          BEQ     3$
 1878  021072  104423  020740                  ERROR,  T55             ;; *** R0 WRONG ***
 1879  021076  010010                   3$:    MOV     R0,(R0)         ; PLACE ADDRESS OF TEMP2 IN TEMP2.
 1880  021100  000263                          SEVC                    ; SET V AND C BITS.
 1881  021102  044000                          BIC     -(R0),R0        ; SHOULD CLEAR R0.
 1882  021104  104405                          CHECKCC+5               ; CHECK FOR CC = 5
 1883  021106  104423  020740                  ERROR,  T55             ;; *** INCORRECT CC BITS ***
 1884  021112  037413  177776                  BIT     @-2(R4),(R3)    ; CHECK IT
 1885  021116  104411                          CHECKCC+11              ; CHECK FOR CC = 11
 1886  021120  104423  020740                  ERROR,  T55             ;; *** INCORRECT CC BITS ***
 1887  021124  012746  125252                  MOV     #125252,-(SP)   ; SET EVERY OTHER BIT ON THE STACK
 1888  021130  017423  177776                  MOV     @-2(R4),(R3)+   ; SET ALL THE BITS AT LOCATION TEMP
 1889  021134  046643  000000                  BIC     0(SP),-(R3)     ; CLEAR EVERY OTHER BIT AT LOCATION TEMP
```

```
 1890  021140  022327  052525                 CMP     (R3)+,#52525    ; TEMP SHOULD CONTAIN # 52525
 1891  021144  001402                         BEQ     4$
 1892  021146  104423  020740                 ERROR,  T55             ;; *** BIC FAILED IN MODE 6 ***
 1893  021152  012700  001014          4$:    MOV     #TEMP2+2,R0     ; PLACE THE ADDRESS OF TEMP2+2 IN R0
 1894  021156  010340                         MOV     R3,-(R0)        ; PLACE THE ADDRESS OF TEMP1 IN TEMP2
 1895  021160  014330                         MOV     -(R3),@(R0)+    ; MOVE # 52525 IN LOCATION TEMP1
 1896  021162  000263                         SEVC                    ; SET V & C BITS
 1897  021164  035026                         BIT     @-(R0),(SP)+    ; BIT TEST TEMP1 WITH STACK AND RESTORE STACK POINTER
 1898  021166  104405                         CHECKCC+5               ; CHECK FOR CC = 5
 1899  021170  104423  020740                 ERROR,  T55             ;; *** INCORRECT CC BITS ***
 1900  021174  020627  003776                 CMP     SP,#STACK       ; MAKE SURE THAT THE SP IS OK
 1901  021200  001402                         BEQ     5$
 1902  021202  104423  020740                 ERROR,  T55             ;; *** STACK POINTER FOULED UP ***
 1903  021206                          5$:
 1904
 1905                                  ;*******************************************************************************
  (2)                                  .SBTTL      T56 -- INC AND DEC
  (2)                                  ;*******************************************************************************
  (2)  021206  104420  000056          T56:    BEGINTEST, 56           ;; CHECK SWITCH REGISTER OPTIONS
 1906  021212  012704  001010          INC1:   MOV     #TEMP1,R4       ; LOAD ADDRESS
 1907  021216  012714  077777                  MOV     #77777,(R4)     ; TEMP1 = 77777
 1908  021222  000261                         SEC
 1909  021224  005214                         INC     (R4)            ; ADD ONES INTO LOCATION
 1910  021226  104413                         CHECKCC+13              ; CHECK FOR CC = 13
 1911  021230  104423  021206                 ERROR,  T56             ;; *** INCORRECT CC BITS ***
 1912  021234  012714  177776                 MOV     #177776,(R4)
 1913  021240  012700  001006                 MOV     #TEMP,R0        ; R0 IS POINTING TO LOCATION TEMP
 1914  021244  005214                         INC     (R4)            ;
 1915  021246  104411                         CHECKCC+11              ; CHECK CC = 11.
 1916  021250  104423  021206                 ERROR,  T56             ;; *** INCORRECT CC BITS ***
 1917  021254  005214                         INC     (R4)            ;
 1918  021256  104405                         CHECKCC+5               ; CHECK FOR CC = 5
 1919  021260  104423  021206                 ERROR,  T56             ;; *** INCORRECT CC BITS ***
 1920  021264  005214                         INC     (R4)
 1921  021266  104401                         CHECKCC+1               ; CHECK FOR CC = 1
 1922  021270  104423  021206                 ERROR,  T56             ;; *** INCORRECT CC BITS ***
 1923  021274  026427  000000  000001         CMP     0(R4),#1        ; CHECK IT
 1924  021302  001402                         BEQ     4$              ; CONTINUE IF OK
 1925  021304  104423  021206                 ERROR,  T56             ;; *** INC INSTRUCTION FAILED ***
 1926  021310  000261                  4$:    SEC
 1927  021312  005314                         DEC     (R4)            ; SUBTRACT ONES FROM LOCATION
 1928  021314  104405                         CHECKCC+5               ; CHECK FOR CC = 5
 1929  021316  104423  021206                 ERROR,  T56             ;; *** INCORRECT CC BITS ***
 1930  021322  005314                         DEC     (R4)
 1931  021324  104411                         CHECKCC+11              ; CHECK CC = 11.
 1932  021326  104423  021206                 ERROR,  T56             ;; *** INCORRECT CC BITS ***
 1933  021332  012714  100000                 MOV     #100000,(R4)
 1934  021336  005314                         DEC     (R4)            ;
 1935  021340  104403                         CHECKCC+3               ; CHECK FOR CC = 3
 1936  021342  104423  021206                 ERROR,  T56             ;; *** INCORRECT CC BITS ***
 1937  021346  005314                         DEC     (R4)            ;
 1938  021350  104401                         CHECKCC+1               ; CHECK FOR CC = 1
 1939  021352  104423  021206                 ERROR,  T56             ;; *** INCORRECT CC BITS ***
 1940
 1941                                  ;*******************************************************************************
  (2)                                  .SBTTL      T57 -- COM AND NEG
```

```
  (2)
  (2)  021356  104420  000057          ;*****************************************************************************
                                       T57:    BEGINTEST, 57           ;; CHECK SWITCH REGISTER OPTIONS.
 1942  021362  012703  001010          COM1:   MOV     #TEMP1,R3       ; LOAD ADDRESS
 1943  021366  012713  125252                  MOV     #125252,(R3)    ; LOAD EVERY OTHER BIT
 1944  021372  000277                          SCC
 1945  021374  005163  000000                  COM     0(R3)           ; 1'S COMPLEMENT
 1946  021400  104401                          CHECKCC+1               ; CHECK FOR CC = 1
 1947  021402  104423  021356                  ERROR,  T57             ;; *** INCORRECT CC BITS ***
 1948  021406  022713  052525                  CMP     #52525,(R3)     ; CHECK IT
 1949  021412  001402                          BEQ     2$              ; CONTINUE IF OK
 1950  021414  104423  021356                  ERROR,  T57             ;; *** COM INSTRUCTION FAILED ***
 1951  021420  000277                  2$:     SCC
 1952  021422  005123                          COM     (R3)+           ; COMPLEMENT BACK
 1953  021424  104411                          CHECKCC+11              ; CHECK FOR CC = 11
 1954  021426  104423  021356                  ERROR,  T57             ;; *** INCORRECT CC BITS ***
 1955  021432  022743  125252                  CMP     #125252,-(R3)   ; CHECK IT
 1956  021436  001402                          BEQ     3$              ; CONTINUE IF OK
 1957  021440  104423  021356                  ERROR,  T57             ;; *** COM INSTRUCTION FAILED ***
 1958  021444  010300                  3$:     MOV     R3,R0           ; R0 IS NOW POINTING TO LOCATION TEMP1
 1959  021446  012710  177777                  MOV     #177777,(R0)
 1960  021452  000277                          SCC
 1961  021454  005110                          COM     (R0)
 1962  021456  104405                          CHECKCC+5               ; CHECK FOR CC = 5
 1963  021460  104423  021356                  ERROR,  T57             ;; *** INCORRECT CC BITS ***
 1964
 1965  021464  012704  001010          NEG1:   MOV     #TEMP1,R4       ; LOAD ADDRESS
 1966  021470  012724  000001                  MOV     #1,(R4)+                ; LOAD THE LOCATION
 1967  021474  010402                          MOV     R4,R2
 1968  021476  012762  100000  000000          MOV     #100000,0(R2)
 1969  021504  005444                          NEG     -(R4)           ; 2'S COMPLEMENT
 1970  021506  104411                          CHECKCC+11              ; CHECK FOR CC = 11
 1971  021510  104423  021464                  ERROR,  NEG1            ;; *** INCORRECT CC BITS ***
 1972  021514  022724  177777                  CMP     #177777,(R4)+   ; CHECK IT
 1973  021520  001402                          BEQ     2$              ; CONTINUE IF OK
 1974  021522  104423  021464                  ERROR,  NEG1            ;; *** NEG INSTRUCTION FAILED ***
 1975  021526  016444  000000          2$:     MOV     0(R4),-(R4)     ; TEMP1 HOLDS LARGEST NEGATIVE NUMBER
 1976  021532  005414                          NEG     (R4)            ; 2'S COMPLEMENT
 1977  021534  104413                          CHECKCC+13              ; CHECK FOR CC = 13
 1978  021536  104423  021464                  ERROR,  NEG1            ;; *** INCORRECT CC BITS ***
 1979  021542  026214  000000                  CMP     0(R2),(R4)      ; CHECK IT
 1980  021546  001402                          BEQ     3$              ; CONTINUE IF OK
 1981  021550  104423  021464                  ERROR,  NEG1            ;; *** NEG FAILED. ***
 1982  021554                          3$:
 1983
 1984                                  ;*****************************************************************************
  (2)                                  .SBTTL      T60 -- ROL AND ROR
  (2)                                  ;*****************************************************************************
  (2)  021554  104420  000060          T60:    BEGINTEST, 60           ;; CHECK SWITCH REGISTER OPTIONS.
 1985  021560  012701  001012          ROL1:   MOV     #TEMP2,R1       ; LOAD ADDRESS
 1986  021564  012711  020000                  MOV     #20000,(R1)     ; LOAD LOCATION
 1987  021570  000257                          CCC                     ; CLEAR FLAGS
 1988  021572  006121                          ROL     (R1)+           ; SHIFT
 1989  021574  006141                          ROL     -(R1)           ;
 1990  021576  104412                          CHECKCC+12              ; CHECK FOR CC = 12
 1991  021600  104423  021554                  ERROR,  T60             ;; *** INCORRECT CC BITS ***
 1992  021604  022711  100000                  CMP     #100000,(R1)    ; CHECK IT
```

```
 1993  021610  001402                        BEQ     1$              ; CONTINUE IF OK
 1994  021612  104423  021554                ERROR,  T60             ;; *** ROL INSTRUCTION FAILED ***
 1995  021616  006161  000000        1$:     ROL     0(R1)           ; SHIFT
 1996  021622  104407                        CHECKCC+7               ; CHECK FOR CC = 7
 1997  021624  104423  021554                ERROR,  T60             ;; *** INCORRECT CC BITS ***
 1998  021630  010102                        MOV     R1,R2           ; R2 IS NOW POINTING TO LOCATION TEMP2
 1999  021632  006112                        ROL     (R2)            ; SHIFT
 2000  021634  022711  000001                CMP     #1,(R1)         ; CHECK IT
 2001  021640  001402                        BEQ     ROR1            ; CONTINUE IF OK
 2002  021642  104423  021554                ERROR,  T60             ;; *** ROL FAILED. ***
 2003
 2004  021646  012702  001012        ROR1:   MOV     #TEMP2,R2       ; LOAD ADDRESS
 2005  021652  012712  000004                MOV     #4,(R2)         ; LOAD LOCATION
 2006  021656  000257                        CCC                     ; CLEAR FLAGS
 2007  021660  006012                        ROR     (R2)            ; SHIFT
 2008  021662  006012                        ROR     (R2)            ;
 2009  021664  022712  000001                CMP     #1,(R2)         ; CHECK IT
 2010  021670  001402                        BEQ     1$              ; CONTINUE IF OK
 2011  021672  104423  021646                ERROR,  ROR1            ;; *** ROR INSTRUCTION FAILED ***
 2012  021676  006012        1$:     ROR     (R2)            ; SHIFT
 2013  021700  104407                        CHECKCC+7               ; CHECK FOR CC = 7
 2014  021702  104423  021646                ERROR,  ROR1            ;; *** INCORRECT CC BITS ***
 2015  021706  006012                        ROR     (R2)            ; SHIFT
 2016  021710  104412                        CHECKCC+12              ; CHECK FOR CC = 12
 2017  021712  104423  021646                ERROR,  ROR1            ;; *** INCORRECT CC BITS ***
 2018  021716  022712  100000                CMP     #100000,(R2)    ; CHECK IT
 2019  021722  001402                        BEQ     2$              ; CONTINUE IF OK
 2020  021724  104423  021646                ERROR,  ROR1            ;; *** ROR FAILED. ***
 2021  021730                        2$:
 2022
 2023                                       ;*************************************************************************
  (2)                                       .SBTTL      T61 -- ASL AND ASR
  (2)                                       ;*************************************************************************
  (2)   021730  104420  000061      T61:    BEGINTEST, 61           ;; CHECK SWITCH REGISTER OPTIONS.
 2024  021734  012703  001012      ASL1:   MOV     #TEMP2,R3       ; LOAD ADDRESS
 2025  021740  012713  020000              MOV     #20000,(R3)     ; LOAD LOCATION
 2026  021744  000257                      CCC                     ; CLEAR FLAGS
 2027  021746  006313                      ASL     (R3)            ; SHIFT
 2028  021750  006313                      ASL     (R3)            ;
 2029  021752  104412                      CHECKCC+12              ; CHECK FOR CC = 12
 2030  021754  104423  021730              ERROR,  T61             ;; *** INCORRECT CC BITS ***
 2031  021760  022713  100000              CMP     #100000,(R3)    ; CHECK IT
 2032  021764  001402                      BEQ     4$              ; CONTINUE IF OK
 2033  021766  104423  021730              ERROR,  T61             ;; *** ASL INSTRUCTION FAILED ***
 2034  021772  006313        4$:     ASL     (R3)            ; SHIFT
 2035  021774  104407                      CHECKCC+7               ; CHECK FOR CC = 7
 2036  021776  104423  021730              ERROR,  T61             ;; *** INCORRECT CC BITS ***
 2037  022002  006313                      ASL     (R3)            ; SHIFT
 2038  022004  104404                      CHECKCC+4               ; CHECK FOR CC = 4
 2039  022006  104423  021730              ERROR,  T61             ;; *** INCORRECT CC BITS ***
 2040
 2041  022012  012704  001012      ASR1:   MOV     #TEMP2,R4       ; LOAD ADDRESSES
 2042  022016  012703  001006              MOV     #TEMP,R3        ;
 2043  022022  012714  000004              MOV     #4,(R4)         ; LOAD LOCATION
 2044  022026  000257                      CCC                     ; CLEAR FLAGS
 2045  022030  006214                      ASR     (R4)            ; SHIFT
```

```
 2046  022032  006214                        ASR     (R4)            ;
 2047  022034  022714  000001                CMP     #1,(R4)         ; CHECK IT
 2048  022040  001402                        BEQ     2$              ; CONTINUE IF OK
 2049  022042  104423  022012                ERROR,  ASR1            ;; *** ASR INSTRUCTION FAILED ***
 2050  022046  006214                2$:     ASR     (R4)            ; SHIFT
 2051  022050  104407                        CHECKCC+7               ; CHECK FOR CC = 7
 2052  022052  104423  022012                ERROR,  ASR1            ;; *** INCURRECT CC BITS ***
 2053  022056  006214                        ASR     (R4)            ; SHIFT
 2054  022060  104404                        CHECKCC+4               ; CHECK FOR CC = 4
 2055  022062  104423  022012                ERROR,  ASR1            ;; *** INCORRECT CC BITS ***
 2056  022066  012713  100002                MOV     #100002,(R3)    ; LOAD LOCATION
 2057  022072  006213                        ASR     (R3)            ; SHIFT
 2058  022074  006213                        ASR     (R3)            ;
 2059  022076  104411                        CHECKCC+11              ; CHECK FOR CC = 11
 2060  022100  104423  022012                ERROR,  ASR1            ;; *** INCORRECT CC BITS ***
 2061  022104  022713  160000                CMP     #160000,(R3)    ; CHECK IT
 2062  022110  001402                        BEQ     3$              ; CONTINUE IF OK
 2063  022112  104423  022012                ERROR,  ASR1            ;; *** ASR FAILED ***
 2064  022116                        3$:
 2065
 2066                                  ;*******************************************************************************
  (2)                                  .SBTTL      T62 -- ADC AND SBC
  (2)                                  ;*******************************************************************************
  (2)  022116  104420  000062  T62:    BEGINTEST, 62                   ;; CHECK SWITCH REGISTER OPTIONS.
 2067  022122  012700  001006  ADC1:   MOV     #TEMP,R0                ; LOAD ADDRESS
 2068  022126  005010                        CLR     (R0)            ; CLEAR THE LOCATION
 2069  022130  000257                        CCC                     ; CLEAR FLAGS
 2070  022132  005510                        ADC     (R0)            ; ADD C BIT = 0
 2071  022134  104404                        CHECKCC+4               ; CHECK FOR CC = 4
 2072  022136  104423  022116                ERROR,  T62             ;; *** INCORRECT CC BITS ***
 2073  022142  000261                        SEC                     ; C=1
 2074  022144  005510                        ADC     (R0)            ; ADD C BIT=1
 2075  022146  000261                        SEC                     ; C=1
 2076  022150  005510                        ADC     (R0)            ; AGAIN
 2077  022152  104400                        CHECK   +0              ; CHECK FOR CC = 0
 2078  022154  104423  022116                ERROR,  T62             ;; *** INCORRECT CC BITS ***
 2079  022160  022710  000002                CMP     #2,(R0)         ; CHECK IT
 2080  022164  001402                        BEQ     4$              ; CONTINUE IF OK
 2081  022166  104423  022116                ERROR,  T62             ;; *** ADC INSTRUCTION FAILED ***
 2082  022172  012[illegible]  077777  4$:   MOV     #77777,(R0)     ; LOAD LARGEST POSITVE NUMBER
 2083  022176  000261                        SEC                     ; C=1
 2084  022200  005510                        ADC     (R0)            ; ADD C BIT=1
 2085  022202  104412                        CHECKCC+12              ; CHECK FOR CC = 12
 2086  022204  104423  022116                ERROR,  T62             ;; *** INCORRECT CC BITS ***
 2087  022210  022710  100000                CMP     #100000,(R0)    ; CHECK IT
 2088  022214  001402                        BEQ     6$              ; CONTINUE IF OK
 2089  022216  104423  022116                ERROR,  T62             ;; *** ADC INSTRUCTION FAILED ***
 2090  022222  012710  177777  6$:   MOV     #-1,(R0)                ; LOAD -1
 2091  022226  000261                        SEC                     ; C=1
 2092  022230  005510                        ADC     (R0)            ; ADD C BIT=1
 2093  022232  104405                        CHECKCC+5               ; CHECK FOR CC = 5
 2094  022234  104423  022116                ERROR,  T62             ;; *** INCORRECT CC BITS ***
 2095
 2096  022240  012701  001006  SBC1:   MOV     #TEMP,R1                ; LOAD ADDRESS
 2097  022244  012711  000003                MOV     #3,(R1)         ; LOAD LOCATION
 2098  022250  000257                        CCC                     ; CLEAR FLAGS
```

```
 2099  022252  005611                          SBC     (R1)            ; SUBTRACT C BIT=0
 2100  022254  104400                          CHECKCC+0               ; CHECK FOR CC = 0
 2101  022256  104423  022240                  ERROR,  SBC1            ;; *** INCORRECT CC BITS ***
 2102  022262  022711  000003                  CMP     #3,(R1)         ; CHECK IT
 2103  022266  001402                          BEQ     2$              ; CONTINUE IF OK
 2104  022270  104423  022240                  ERROR,  SBC1            ;; *** SBC INSTRUCTION FAILED ***
 2105  022274  000261                  2$:     SEC                     ; C=1
 2106  022276  005611                          SBC     (R1)            ; SUBTRACT C BIT=1
 2107  022300  000261                          SEC                     ; C=1
 2108  022302  005611                          SBC     (R1)            ;
 2109  022304  104400                          CHECKCC+0               ; CHECK FOR CC = 0
 2110  022306  104423  022240                  ERROR,  SBC1            ;; *** INCORRECT CC BITS ***
 2111  022312  022711  000001                  CMP     #1,(R1)         ; CHECK IT
 2112  022316  001402                          BEQ     3$
 2113  022320  104423  022240                  ERROR,  SBC1            ;; *** SBC INSTRUCTION FAILED ***
 2114  022324  000261                  3$:     SEC                     ; C=1
 2115  022326  005611                          SBC     (R1)            ; SUBTRACT C BIT=1
 2116  022330  104404                          CHECKCC+4               ; CHECK FOR CC = 4
 2117  022332  104423  022240                  ERROR,  SBC1            ;; *** INCORRECT CC BITS ***
 2118  022336  000261                          SEC                     ; C=1
 2119  022340  005611                          SBC     (R1)            ; SUBTRACT C BIT = 1
 2120  022342  104411                          CHECKCC+11              ; CHECK FOR CC = 11
 2121  022344  104423  022240                  ERROR,  SBC1            ;; *** INCORRECT CC BITS ***
 2122  022350  022711  177777                  CMP     #-1,(R1)        ; CHECK IT
 2123  022354  001402                          BEQ     4$              ; CONTINUE IF OK
 2124  022356  104423  022240                  ERROR,  SBC1            ;; *** SBC INSTRUCTION FAILED ***
 2125  022362  012711  100000          4$:     MOV     #100000,(R1)    ; LOAD R1
 2126  022366  000261                          SEC                     ; C=1
 2127  022370  005611                          SBC     (R1)            ; SUBTRACT C BIT = 1
 2128  022372  104402                          CHECKCC+2               ; CHECK FOR CC = 2
 2129  022374  104423  022240                  ERROR,  SBC1            ;; *** INCORRECT CC BITS ***
 2130
 2131                                  ;*******************************************************************************
  (2)                                  .SBTTL      T63 -- SXT, SWAB, AND XOR
  (2)                                  ;*******************************************************************************
  (2)   022400  104420  000063          T63:    BEGINTEST, 63           ;; CHECK SWITCH REGISTER OPTIONS.
 2132   022404  012702  001010          SXT1:   MOV     #TEMP1,R2       ; LOAD ADDRESS
 2133   022410  005012                          CLR     (R2)            ; CLEAR LOCATIONS
 2134   022412  000277                          SCC
 2135   022414  000254                          CLNZ
 2136   022416  006712                          SXT     (R2)            ; SIGN EXTEND
 2137   022420  104405                          CHECKCC+5               ; CHECK FOR CC = 5
 2138   022422  104423  022400                  ERROR,  T63             ;; *** INCORRECT CC BITS ***
 2139   022426  005712                          TST     (R2)            ; LOCATION SHOULD STILL BE 0
 2140   022430  001402                          BEQ     2$              ; CONTINUE IF OK
 2141   022432  104423  022400                  ERROR,  T63             ;; *** SXT INSTRUCTION FAILED ***
 2142   022436  000273                  2$:     SENVC                   ; SET N, V & C BITS
 2143   022440  006712                          SXT     (R2)            ; SIGN EXTEND
 2144   022442  104411                          CHECKCC+11              ; CHECK FOR CC = 11
 2145   022444  104423  022400                  ERROR,  T63             ;; *** INCORRECT CC BITS ***
 2146   022450  022712  177777                  CMP     #-1,(R2)        ; LOCATION SHOULD NEW HAVE -1
 2147   022454  001402                          BEQ     SWAB1           ; CONTINUE IF OK
 2148   022456  104423  022400                  ERROR,  T63             ;; *** SXT FAILED ***
 2149
 2150   022462  012703  001012          SWAB1:  MOV     #TEMP2,R3       ; LOAD ADDRESS
 2151   022466  012713  125125                  MOV     #125125,(R3)    ; LOAD BIT PATTERN INTO LOCATION
```

```
 2152  022472  000277                          SCC
 2153  022474  000250                          CLN
 2154  022476  000313                          SWAB    (R3)            ; SWAP BYTES OF LOCATIONS
 2155  022500  104410                          CHECKCC+10              ; CHECK FOR CC = 10
 2156  022502  104423  022462                  ERROR,  SWAB1           ;; *** INCORRECT CC BITS ***
 2157  022506  022713  052652                  CMP     #52652,(R3)     ; CHECK IT
 2158  022512  001402                          BEQ     1$              ; CONTINUE IF OK
 2159  022514  104423  022462                  ERROR,  SWAB1           ;; *** SWAB INSTRUCTION FAILED ***
 2160  022520  012713  000377          1$:     MOV     #377,(R3)
 2161  022524  000277                          SCC
 2162  022526  000244                          CLZ
 2163  022530  000363  000000                  SWAB    0(R3)
 2164  022534  104404                          CHECKCC+4               ; CHECK FOR CC = 4
 2165  022536  104423  022462                  ERROR,  SWAB1           ;; *** INCORRECT CC BITS ***
 2166  022542  022713  177400                  CMP     #177400,(R3)
 2167  022546  001402                          BEQ     XOR1
 2168  022550  104423  022462                  ERROR,  SWAB1           ;; *** SWAB FAILED. ***
 2169
 2170  022554  012704  177777          XOR1:   MOV     #-1,R4          ; LOAD LOCATIONS
 2171  022560  012767  177777  156222          MOV     #-1,TEMP1       ;
 2172  022566  000277                          SCC
 2173  022570  074467  156214                  XOR     R4,TEMP1        ; SHOULD PRODUCE 0'S IN TEMP1
 2174  022574  104405                          CHECKCC+5               ; CHECK FOR CC = 5
 2175  022576  104423  022554                  ERROR,  XOR1            ;; *** INCORRECT CC BITS ***
 2176  022602  012767  077777  156200          MOV     #77777,TEMP1
 2177  022610  012700  001010                  MOV     #TEMP1,R0       ; PLACE THE ADDRESS OF TEMP1 IN R0
 2178  022614  000263                          SEVC                    ; SET V & C BITS
 2179  022616  000244                          CLZ
 2180  022620  074410                          XOR     R4,(R0)
 2181  022622  104411                          CHECKCC+11              ; CHECK FOR CC = 11
 2182  022624  104423  022554                  ERROR,  XOR1            ;; *** INCORRECT CC BITS ***
 2183  022630  012701  125252                  MOV     #125252,R1      ; LOAD LOCATIONS
 2184  022634  012720  052525                  MOV     #52525,(R0)+    ;
 2185  022640  000277                          SCC
 2186  022642  074140                          XOR     R1,-(R0)        ; SHOULD PRODUCE ALL 1'S IN TEMP1
 2187  022644  104411                          CHECKCC+11              ; CHECK FOR CC = 11
 2188  022646  104423  022554                  ERROR,  XOR1            ;; *** INCORRECT CC BITS ***
 2189  022652  022767  177777  156130          CMP     #-1,TEMP1       ; CHECK IT
 2190  022660  001402                          BEQ     1$              ; CONTINUE IF OK
 2191  022662  104423  022554                  ERROR,  XOR1            ;; *** XOR FAILED. ***
 2192  022666                          1$:
 2193
 2194                                  ;*******************************************************************************
  (2)                                 .SBTTL      T64 -- ADD, SUB, AND SOB
  (2)                                 ;*******************************************************************************
  (2)  022666  104420  000064          T64:    BEGINTEST, 64           ;; CHECK SWITCH REGISTER OPTIONS.
 2195  022672  012700  001012          ADD1:   MOV     #TEMP2,R0       ; LOAD ADDRESSES
 2196  022676  012701  001006                  MOV     #TEMP,R1        ;
 2197  022702  012767  021421  156102          MOV     #21421,TEMP2    ; LOAD LOCATIONS
 2198  022710  011011                          MOV     (R0),(R1)       ;
 2199  022712  061011                          ADD     (R0),(R1)       ; ADD
 2200  022714  104400                          CHECKCC+0               ; CHECK FOR CC = 0
 2201  022716  104423  022666                  ERROR,  T64             ;; *** INCORRECT CC BITS ***
 2202  022722  022767  043042  156056          CMP     #43042,TEMP     ; CHECK IT
 2203  022730  001402                          BEQ     1$              ; CONTINUE IF OK
 2204  022732  104423  022666                  ERROR,  T64             ;; *** ADD INSTRUCTION FAILED ***
```

```
2205  022736  005010                    1$:     CLR     (R0)            ; CLEAR LOCATION TEMP2
2206  022740  060020                            ADD     R0,(R0)+        ; PLACE THE ADDRESS OF TEMP2 IN TEMP2
2207  022742  024027  001014                    CMP     -(R0),#TEMP2+2  ; CHECK IT.
2208  022746  001402                            BEQ     2$
2209  022750  104423  022666                    ERROR,  T64             ;; *** ADD INSTRUCTION FAILED IN MODE 2 ***
2210  022754  012767  156357  156030    2$:     MOV     #-21421,TEMP2   ; LOAD LOCATIONS
2211  022762  012011                            MOV     (R0)+,(R1)      ;
2212  022764  064011                            ADD     -(R0),(R1)      ; ADD
2213  022766  104411                            CHECKCC+11              ; CHECK FOR CC = 11
2214  022770  104423  022666                    ERROR,  T64             ;; *** INCORRECT CC BITS ***
2215  022774  022767  134736  156004            CMP     #-43042,TEMP    ; CHECK IT
2216  023002  001402                            BEQ     3$              ;CONTINUE IF OK
2217  023004  104423  022666                    ERROR,  T64             ;; *** ADD INSTRUCTION FAILED ***
2218  023010  012767  100000  155774    3$:     MOV     #100000,TEMP2   ; LOAD LOCATIONS
2219  023016  011061  000000                    MOV     (R0),0(R1)      ;
2220  023022  066011  000000                    ADD     0(R0),(R1)      ; ADD SHOULD RESULT AS 0'S
2221  023026  104407                            CHECKCC+7               ; CHECK FOR CC=7
2222  023030  104423  022666                    ERROR,  T64             ;; *** INCORRECT CC BITS ***
2223  023034  012767  021421  155746            MOV     #21421,TEMP1    ; LOAD LOCATION TEMP1
2224  023042  012760  001010  000000            MOV     #TEMP1,0(R0)    ; PLACE THE ADDRESS OF TEMP1 IN TEMP2
2225  023050  012711  156357                    MOV     #-21421,(R1)    ; LOAD LOCATION TEMP
2226  023054  010004                            MOV     R0,R4           ; MAKE R4 POINT TO LOCATION TEMP2
2227  023056  067411  000000                    ADD     @0(R4),(R1)     ; ADD SHOULD RESULT AS 0'S
2228  023062  104405                            CHECKCC+5               ; CHECK FOR CC=5
2229  023064  104423  022666                    ERROR,  T64             ;; *** INCORRECT CC BITS ***
2230  023070  005430                            NEG     @(R0)+          ; NEGATE THE CONTENTS OF TEMP1
2231  023072  012746  021421                    MOV     #21421,-(SP)    ; PLACE # 21421 ON THE STACK
2232  023076  065066  000000                    ADD     @-(R0),0(SP)    ; ADD, SHOULD=0'S
2233  023102  104405                            CHECKCC+5               ; CHECK FOR CC=5
2234  023104  104423  022666                    ERROR,  T64             ;; *** INCORRECT CC BITS ***
2235  023110  005726                            TST     (SP)+           ; CHECK THE STACK TO CONTAIN 0, ALSO
2236                                                                    ; RESTORE THE STACK POINTER
2237  023112  001402                            BEQ     4$
2238  023114  104423  022666                    ERROR,  T64             ;; *** ADD INSTRUCTION FAILED IN MODE 5 ***
2239  023120  012767  137777  155664    4$:     MOV     #137777,TEMP2
2240  023126  062767  137777  155656            ADD     #137777,TEMP2
2241  023134  104403                            CHECKCC+3               ; CHECK CC=3
2242  023136  104423  022666                    ERROR,  T64             ;; *** INCORRECT CC BITS ***
2243  023142  022767  077776  155642            CMP     #77776,TEMP2
2244  023150  001402                            BEQ     SUB1
2245  023152  104423  022666                    ERROR,  T64             ;; *** ADD FAILED ***
2246
2247  023156  012702  001006            SUB1:   MOV     #TEMP,R2        ; LOAD ADDRESSES
2248  023162  012703  001010                    MOV     #TEMP1,R3       ;
2249  023166  012767  021421  155612            MOV     #21421,TEMP     ; LOAD LOCATIONS
2250  023174  012767  156357  155606            MOV     #-21421,TEMP1   ;
2251  023202  161213                            SUB     (R2),(R3)       ; RESULT SHOULD=-43042
2252  023204  104410                            CHECKCC+10              ; CHECK FOR CC = 10
2253  023206  104423  023156                    ERROR,  SUB1            ;; *** INCORRECT CC BITS ***
2254  023212  022767  134736  155570            CMP     #-43042,TEMP1   ; CHECK IT
2255  023220  001402                            BEQ     1$              ;CONTINUE IF OK
2256  023222  104423  023156                    ERROR,  SUB1            ;; *** SUB INSTRUCTION FAILED ***
2257  023226  012767  021421  155554    1$:     MOV     #21421,TEMP1    ; LOAD LOCATION
2258  023234  161213                            SUB     (R2),(R3)       ; RESULT SHOULD=0
2259  023236  001402                            BEQ     2$
2260  023240  104423  023156                    ERROR,  SUB1            ;; *** SUB INSTRUCTION FAILED ***
```

```
 2261  023244  012767  177777  155536  2$:     MOV     #-1,TEMP1       ; LOAD LOCATIONS
 2262  023252  012767  077777  155526          MOV     #77777,TEMP     ; LOAD LOCATIONS
 2263  023260  161312                          SUB     (R3),(R2)       ; RESULT SHOULD GIVE 100000 AND OVERFLOW
 2264  023262  104413                          CHECKCC+13              ; CHECK FOR CC = 13
 2265  023264  104423  023156                  ERROR,  SUB1            ;; *** INCORRECT CC BITS ***
 2266  023270  022767  100000  155510          CMP     #100000,TEMP    ; CHECK IT
 2267  023276  001402                          BEQ     3$              ; CONTINUE IF OK
 2268  023300  104423  023156                  ERROR,  SUB1            ;; *** SUB INSTRUCTION FAILED ***
 2269  023304  012712  177777          3$:     MOV     #-1,(R2)
 2270  023310  161312                          SUB     (R3),(R2)
 2271  023312  104404                          CHECKCC+4               ; CHECK FOR CC = 4
 2272  023314  104423  023156                  ERROR,  SUB1            ;; *** INCORRECT CC BITS ***
 2273  023320  012767  077777  155460          MOV     #77777,TEMP
 2274  023326  162767  077777  155452          SUB     #77777,TEMP
 2275  023334  104404                          CHECKCC+4               ; CHECK FOR CC=4
 2276  023336  104423  023156                  ERROR,  SUB1            ;; *** INCORRECT CC BITS ***
 2277  023342  005767  155440                  TST     TEMP
 2278  023346  001402                          BEQ     SOB1            ; TEMP SHOULD BE =0
 2279  023350  104423  023156                  ERROR,  SUB1            ;; *** SUB INSTRUCTION FAILED ***
 2280
 2281  023354  012700  000012          SOB1:   MOV     #10.,R0         ; LOAD REGISTERS
 2282  023360  005001                          CLR     R1              ;
 2283  023362  005201                  1$:     INC     R1              ; KEEP COUNT
 2284  023364  020127  000012                  CMP     R1,#10.
 2285  023370  003402                          BLE     2$
 2286  023372  104423  023354                  ERROR,  SOB1            ;; *** SOB INSTRUCTION FAILED ***
 2287  023376  000277                  2$:     SCC
 2288  023400  077010                          SOB     R0,1$           ; SUB. 1 FROM REG. 0, GO BACK TO 1$
 2289  023402  104417                          CHECKCC+17              ; CHECK FOR CC = 17
 2290  023404  104423  023354                  ERROR,  SOB1            ;; *** INCORRECT CC BITS ***
 2291  023410  005700                          TST     R0              ; REG. 0 = 0 ?
 2292  023412  001402                          BEQ     3$              ; NO, FAILED
 2293  023414  104423  023354                  ERROR,  SOB1            ;; *** SOB INSTRUCTION FAILED ***
 2294  023420  022701  000012          3$:     CMP     #10.,R1         ; DID IT GO THRU 10 TIMES ?
 2295  023424  001402                          BEQ     4$              ; CONTINUE IF OK
 2296  023426  104423  023354                  ERROR,  SOB1            ;; *** SOB INSTRUCTION FAILED ***
 2297  023432  012704  000010          4$:     MOV     #10,R4          ; PLACE #10 IN R4
 2298  023436  077401                  5$:     SOB     R4,5$           ; STAY HERE UNTILL R4 = 0
 2299  023440  005704                          TST     R4
 2300  023442  001402                          BEQ     6$              ; CONTINUE IF OK
 2301  023444  104423  023354                  ERROR,  SOB1            ;; *** SOB FAILED ***
 2302  023450                          6$:
 2303
 2304                                  ;*******************************************************************************
  (2)                                  .SBTTL      T65 -- MTPS AND MFPS
  (2)                                  ;*******************************************************************************
  (2)  023450  104420  000065  T65:     BEGINTEST, 65           ;; CHECK SWITCH REGISTER OPTIONS.
 2305  023454  012700  001006  TPSW1:   MOV     #TEMP,R0        ; PUT THE ADDRESS OF TEMP IN R0
 2306  023460  012701  001010          MOV     #TEMP1,R1       ; PUT THE ADDRESS OF TEMP1 IN R1
 2307  023464  012711  177777          MOV     #177777,(R1)    ; TEMP1 = 177777
 2308  023470  005010                  CLR     (R0)            ; TEMP  = 0
 2309  023472  106410                  MTPS    (R0)            ; PSW = 0
 2310  023474  104400                  CHECKCC+0               ; CHECK FOR CC = 0
 2311  023476  104423  023450          ERROR,  T65             ;; *** INCORRECT CC BITS ***
 2312  023502  106711                  MFPS    (R1)            ; MOVE PSW TO TEMP1
 2313  023504  104404                  CHECKCC+4               ; CHECK FOR CC = 4
```

```
2314  023506  104423  023450                  ERROR,  T65             ;; *** INCORRECT CC BITS ***
2315  023512  022711  177400                  CMP     #177400,(R1)    ; CHECK TEMP1 TO MAKE SURE THAT  ONLY
2316                                                                    ; THE LOWER BYTE WAS AFFECTED BY MFPS
2317  023516  001402                          BEQ     1$
2318  023520  104423  023450                  ERROR,  T65             ;; *** MTPS OR MFPS INSTRUCTION FAILED ***
2319  023524  005011                  1$:     CLR     (R1)
2320  023526  106427  000337                  MTPS    #337            ; SET PSW = 317 SINCE T BIT CAN NOT BE SET BY MTPS
2321  023532  104417                          CHECKCC+17              ; CHECK FOR CC = 17
2322  023534  104423  023450                  ERROR,  T65             ;; *** INCORRECT CC BITS ***
2323  023540  106767  155244                  MFPS    TEMP1           ; MOVE PSW TO TEMP1
2324  023544  104411                          CHECKCC+11              ; CHECK FOR CC = 11 (C BIT SHOULD NOT BE EFFECTED BY MFP
2325  023546  104423  023450                  ERROR,  T65             ;; *** INCORRECT CC BITS ***
2326  023552  022767  000317  155230          CMP     #317,TEMP1
2327  023560  001402                          BEQ     2$
2328  023562  104423  023450                  ERROR,  T65             ;; *** MFPS INSTRUCTION FAILED IN MODE 6 ***
2329  023566                          2$:
2330
2331                                  ;*******************************************************************************
  (2)                               .SBTTL      T66 -- BYTE INSTRUCTIONS AFFECTING FULL WORDS
  (2)                               ;*******************************************************************************
  (2)   023566  104420  000066      T66:    BEGINTEST, 66           ;; CHECK SWITCH REGISTER OPTIONS.
2332  023572  005000                BTWRD:  CLR     R0
2333  023574  000277                        SCC
2334  023576  112700  000200                MOVB    #200,R0         ; SET THE HIGHEST BIT OF THE
2335                                                                  ; LOWER BYTE
2336  023602  104411                        CHECKCC+11              ; CHECK FOR CC=11
2337  023604  104423  023566                ERROR,  T66             ;; *** INCORRECT CC BITS ***
2338  023610  022700  177600                CMP     #177600,R0      ; CHECK FOR SIGN EXTENSION IN R0
2339  023614  001402                        BEQ     1$
2340  023616  104423  023566                ERROR,  T66             ;; *** SIGN WAS NOT EXTENDED IN R0 ***
2341  023622  000277                  1$:   SCC
2342  023624  012700  177777                MOV     #177777,R0
2343  023630  112700  000000                MOVB    #0,R0           ;CLEAR THE LOWER BYTE OF R0.
2344  023634  104405                        CHECKCC+5               ; CHECK FOR CC=5
2345  023636  104423  023566                ERROR,  T66             ;; *** INCORRECT CC BITS ***
2346  023642  005700                        TST     R0              ; CHECK R0 FOR SIGN EXTENTION
2347  023644  001402                        BEQ     2$
2348  023646  104423  023566                ERROR,  T66             ;; *** SIGN WAS NOT EXTENDED IN R0. ***
2349  023652  012704  001012          2$:   MOV     #TEMP2,R4       ; R4 IS POINTING TO TEMP2
2350  023656  012714  000377                MOV     #377,(R4)       ; PLACE #377 IN LOCATION TEMP2
2351  023662  012706  003774                MOV     #STACK-2,R6
2352  023666  116426  000000                MOVB    0(R4),(R6)+     ; PUSH # 377 ON STACK
2353  023672  022706  003776                CMP     #STACK,R6
2354  023676  001402                        BEQ     3$
2355  023700  104423  023566                ERROR,  T66             ;; *** R6 DID NOT GET INCREMENTED ***
2356  023704  124627  000377          3$:   CMPB    -(R6),#377      ; CHECK LOCATION STACK-2 TO
2357                                                                  ; CONTAIN PROPER DATA
2358  023710  001402                        BEQ     4$
2359  023712  104423  023566                ERROR,  T66             ;; *** BYTE INSTPUCTION IS FAILING WITH R6 ***
2360  023716  022706  003774          4$:   CMP     #STACK-2,R6     ; CHECK THAT R6 WAS DECREMENTED
2361                                                                  ; BY 2 BY A BYTE INSTRUCTION
2362  023722  001402                        BEQ     5$
2363  023724  104423  023566                ERROR,  T66             ;; *** R6 WAS NOT DECREMENTED ***
2364  023730  016467  000000  155C50  5$:   MOV     0(R4),TEMP      ; SET THE LOWER BYTE OF LOCATION TEMP
2365  023736  005726                        TST     (R6)+           ; RESTORE STACK POINTER
2366  023740  000277                        SCC
```

```
2367  023742  114667  155041                  MOVB    -(SP),TEMP+1    ; SET THE HIGHER BYTE OF LOCATION TEMP
2368  023746  104411                          CHECKCC+11              ; CHECK FOR CC=11
2369  023750  104423  023566                  ERROR,  T66             ;; *** INCORRECT CC BITS ***
2370  023754  022767  177777  155024          CMP     #177777,TEMP    ; CHECK TEMP FOR THE CORRECT VALUE
2371  023762  001402                          BEQ     6$
2372  023764  104423  023566                  ERROR,  T66             ;; *** TEMP FOULED UP ***
2373  023770  005067  155012          6$:     CLR     TEMP
2374  023774  000241                          CLC
2375  023776  105167  155005                  COMB    TEMP+1          ; WRITE 1'S IN THE HIGHER BYTE OF TEMP
2376  024002  104411                          CHECKCC+11              ; CHECK FOR CC=11
2377  024004  104423  023566                  ERROR,  T66             ;; *** INCORRECT CC BITS ***
2378  024010  022767  177400  154770          CMP     #177400,TEMP
2379  024016  001402                          BEQ     7$
2380  024020  104423  023566                  ERROR,  T66             ;; *** TEMP FOULED UP ***
2381
2382  024024  004767  010700          7$:     CALL    ENDSEG          ; CHECK FOR LOOP-IN-SEG...
2383                                                                  ;...RETURN AND FALL THRU IF NOT.
2384                                  .SBTTL
```

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

024030  012767  036222  012142  S2:     MOV     #SEG2,SEGN      ; SEGMENT 2 TEXT.

                                        ;*************************************************************************
                                        .SBTTL      T67 -- TEST AUTO INCREMENT/DECREMENT OF STACK POINTER (R6)
                                        ;*************************************************************************
024036  104420  000067          T67:    DEFINTEST, 67           ;; CHECK SWITCH REGISTER OPTIONS.
024042  012700  001026          R6TST:  MOV     #K4,R0          ; USE K4 AS PSUEDO STACK...
024046  010006                          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 ***
```

```
2436
2437  024256  012767  123456  154544  1$:     MOV     #123456,K5
2438  024264  012767  050505  154526          MOV     #050505,K1
2439  024272  012701  001020                  MOV     #K1,R1          ;R1(050505)K1
2440  024276  012706  001032                  MOV     #K6,R6          ;R6(123456)K5
2441  024302  114621                          MOVB    -(R6),(R1)+     ;LOW .BYTE OF R6 TO R1 (DECREMENT)
2442  024304  026727  154510  050456          CMP     K1,#050456
2443  024312  001402                          BEQ     2$
2444  024314  104423  024210                  ERROR,  T70             ;; *** FALSE R6 .BYTE TRANSFER ***
2445
2446  024320  012767  123456  154472  2$:     MOV     #123456,K1
2447  024326  012767  050505  154474          MOV     #050505,K5
2448  024334  012701  001020                  MOV     #K1,R1          ;(123456)
2449  024340  012706  001030                  MOV     #K5,R6          ;(050505)
2450  024344  112126                          MOVB    (R1)+,(R6)+     ;LOW OF R1 TO LOW OF R6
2451  024346  022767  050456  154454          CMP     #050456,K5
2452  024354  001402                          BEQ     3$
2453  024356  104423  024210                  ERROR,  T70             ;; *** FALSE R6 .BYTE TRANSFER ***
2454
2455  024362  012767  123456  154430  3$:     MOV     #123456,K1
2456  024370  012767  050505  154432          MOV     #050505,K5
2457  024376  012701  001021                  MOV     #K1+1,R1        ;123456
2458  024402  012706  001030                  MOV     #K5,R6          ;050505
2459  024406  112126                          MOVB    (R1)+,(R6)+     ;HIGH OF R1 TO LOW OF R6
2460  024410  026727  154414  050647          CMP     K5,#050647
2461  024416  001402                          BEQ     4$
2462  024420  104423  024210                  ERROR,  T70             ;; *** FALSE R6 .BYTE TRANSFER ***
2463
2464  024424  012767  123456  154366  4$:     MOV     #123456,K1
2465  024432  012767  050505  154370          MOV     #050505,K5
2466  024440  012701  001021                  MOV     #K1+1,R1        ;R1-123456-ODD ADDRESS
2467  024444  012706  001030                  MOV     #K5,R6          ;R6-050505--.EVEN ADDRESS
2468  024450  112521                          MOVB    (R6)+,(R1)+     ;LOW OF R6 TO HIGH OF R1
2469  024452  022767  042456  154340          CMP     #042456,K1
2470  024460  001402                          BEQ     5$
2471  024462  104423  024210                  ERROR,  T70             ;; *** FAILED LOW OF 6 TO HIGH OF 1 ***
2472  024466                          5$:
2473
2474                                  ;*************************************** ***********************************
  (2)                               .SBTTL      T71 -- TEST BYTE COMPARES ON SEQUENTIAL ODD/EVEN ADDRESSES
  (2)                               ;*************************************************************************
  (2)   024466  104420  000071      T71:    BEGINTEST, 71           ;; CHECK SWITCH REGISTER OPTIONS.
2475  024472  005067  154320                  CLR     K0              ; K0 <= 000000
2476  024476  012767  052525  154314          MOV     #52525,K1       ; K1 <= 052525
2477  024504  012767  052400  154310          MOV     #52400,K2       ; K2 <= 052400
2478  024512  126767  154302  154301          CMPB    K1,K1+1
2479  024520  001402                          BEQ     1$
2480  024522  104423  024466                  ERROR,  T71             ;; *** LOW TO HIGH IN SAME WORD FAILS ***
2481  024526  126767  154267  154264  1$:     CMPB    K1+1,K1
2482  024534  001402                          BEQ     2$
2483  024536  104423  024466                  ERROR,  T71             ;; *** HIGH TO LOW IN SAME WORD FAILS ***
2484  024542  126767  154255  154250  2$:     CMPB    K2+1,K1
2485  024550  001402                          BEQ     3$
2486  024552  104423  024466                  ERROR,  T71             ;; *** HIGH TO LOW IN DIFFERENT WORDS FAILS ***
2487  024556  126767  154240  154232  3$:     CMPB    K2,K0
2488  024564  001402                          BEQ     4$
```

```
 2489  024566  104423  024466                       ERROR,  T71             ;; *** EVEN TO EVEN FAILED ***
 2490  024572  126767  154223  154223  4$:          CMPB    K1+1,K2+1
 2491  024600  001402                               BEQ     5$
 2492  024602  104423  024466                       ERROR,  T71             ;; *** ODD TO ODD FAILED ***
 2493  024606  126767  154210  154207  5$:          CMPB    K2,K2+1
 2494  024614  001002                               BNE     6$
 2495  024616  104423  024466                       ERROR,  T71             ;; *** LOW TO HIGH IN SAME WORD FAILED ***
 2496  024622  126767  154175  154173  6$:          CMPB    K2+1,K2+1
 2497  024630  001402                               BEQ     7$
 2498  024632  104423  024466                       ERROR,  T71             ;; *** HIGH TO HIGH IN SAME WORD FAILED ***
 2499  024636  126767  154160  154155  7$:          CMPB    K2,K1+1
 2500  024644  001002                               BNE     8$
 2501  024646  104423  024466                       ERROR,  T71             ;; *** EVEN TO ODD FAILED ***
 2502  024652                          8$:
 2503
 2504                                  ;****************** ************************************************************
  (2)                                  .SBTTL      T72 -- TEST THAT BUS TIME-OUT TRAPS TO THE "RESTART" ADDRESS
  (2)                                  ;************************************************************************
  (2)  024652  104420  000072          T72:    BEGINTEST, 72           ;; CHECK SWITCH REGISTER OPTIONS.
 2505                                  ;
 2506                                  ; T11 TRAPS TO ODT "RESTART" ON TIME-OUT.  THE CODE THERE WILL
 2507                                  ; EMULATE A TRAP THRU THE TRADITIONAL BUS ERROR VECTOR (4).
 2508                                  ;
 2509  024656  012706  003776                       MOV     #STACK,SP       ; SET STACK POINTER.
 2510  024662  012767  024712  153114               MOV     #3$,ERRVEC      ; SET TRAP VECTOR.
 2511  024670  106427  000000                       MTPS    #PRIO
 2512  024674  005737  160000          1$:          TST     @#160000        ; NEXM -- TIME-OUT AND TRAP.
 2513  024700  000257                               CCC                     ; SHOULD EXECUTE THIS...
 2514  024702  000240                  2$:          NOP                     ;...THEN TRAP AND STACK THIS PC.
 2515  024704  104423  024652                       ERROR,  T72             ;; *** BUS TIME-OUT DIDN'T TRAP AT ALL ***
 2516  024710  000414                               BR      5$
 2517  024712  011600                  3$:          MOV     (SP),R0         ; GET STACKED PC...
 2518  024714  016601  000002                       MOV     2(SP),R1        ;...AND PSW.
 2519  024720  020027  024702                       CMP     R0,#2$
 2520  024724  001402                               BEQ     4$              ; BR IF PC IS RIGHT.
 2521  024726  104423  024652                       ERROR,  T72             ;; *** STACKED PC INCORRECT ON TIME-OUT TRAP ***
 2522
 2523  024732  005701                  4$:          TST     R1
 2524  024734  001402                               BEQ     5$              ; BR IF PSW IS RIGHT.
 2525  024736  104423  024652                       ERROR,  T72             ;; *** STACKED PSW INCORRECT ON TIME-OUT TRAP ***
 2526
 2527  024742  012767  000006  153034  5$:          MOV     #ERRVEC+2,ERRVEC ; RESET TRAP CATCHER.
 2528
 2529                                  ;************************************************************************
  (2)                                  .SBTTL      T73 -- TEST THAT AN "ILLEGAL" INSTRUCTION TRAPS TO 4
  (2)                                  ;************************************************************************
  (2)  024750  104420  000073          T73:    BEGINTEST, 73           ;; CHECK SWITCH REGISTER OPTIONS.
 2530                                  ;
 2531                                  ; DO BOTH "JMP R" AND "JSR R,R"
 2532                                  ; ON ERROR, R0 POINTS TO THE OFFENDER.
 2533                                  ;
 2534  024754  012700  025012                       MOV     #1$,R0          ; 1ST PASS USES A "JMP R"
 2535  024760  000402                               SKP2
 2536  024762  012700  025016          8$:          MOV     #2$,R0          ; 2ND PASS USES A "JSR R,R"
 2537  024766  012706  003776                       MOV     #STACK,SP       ;STACK POINTER SETUP
 2538  024772  012767  025030  153004               MOV     #3$,ERRVEC      ; SET TRAP PC...
```

```
 2539  025000  005067  153002                  CLR     ERRVEC+2        ;...AND PSW.
 2540  025004  106427  000317                  MTPS    #PRI6!17        ; SET CURRENT (OLD) PSW.
 2541  025010  000110                          JMP     (R0)            ; DISPATCH TO ONE OR THE OTHER.
 2542  025012  000101                  1$:     JMP     R1              ; ILLEGAL JMP -- SHOULD TRAP.
 2543  025014  000401                          SKP1
 2544  025016  004101                  2$:     JSR     R1,R1           ; ILLEGAL JSR -- SHOULD TRAP.
 2545  025020  000240                          NOP
 2546  025022  104423  024750                  ERROR,  T73             ;; *** ILLEGAL JMP/JSR DIDN'T TRAP ***
 2547  025026  000427                          BR      7$
 2548
 2549  025030  106701                  3$:     MFPS    R1              ; OK, GET CURRENT (NEW) PSW.
 2550  025032  001402                          BEQ     4$              ; BR IF IT'S RIGHT.
 2551  025034  104423  024750                  ERROR,  T73             ;; *** PSW INCORRECT AFTER ILLEGAL INSTR TRAP TO 4 ***
 2552  025040  020627  003772          4$:     CMP     SP,#STACK-4
 2553  025044  001402                          BEQ     5$              ; BR IF STACK IS RIGHT.
 2554  025046  104423  024750                  ERROR,  T73             ;; *** STACK POINTER INCORRECT ***
 2555  025052  021627  025014          5$:     CMP     (SP),#1$+2
 2556  025056  001405                          BEQ     6$              ; BR IF STACKED PC IS RIGHT (JMP).
 2557  025060  021627  025020                  CMP     (SP),#2$+2
 2558  025064  001402                          BEQ     6$              ; BR IF STACKED PC IS RIGHT (JSR).
 2559  025066  104423  024750                  ERROR,  T73             ;; *** INCORRECT PC SAVED ON STACK ***
 2560  025072  026627  000002  000317  6$:     CMP     2(SP),#PRI6!17
 2561  025100  001402                          BEQ     7$              ; BR IF STACKED (OLD) PSW IS RIGHT.
 2562  025102  104423  024750                  ERROR,  T73             ;; *** INCORRECT PSW SAVED ON STACK ***
 2563
 2564  025106  020027  025012          7$:     CMP     R0,#1$          ; IS THIS THE 1ST PASS ??
 2565  025112  001723                          BEQ     8$              ; LOOP ONCE IF SO...
 2566  025114  012767  000006  152662          MOV     #ERRVEC+2,ERRVEC ;...OTHERWISE, RESET VECTOR...
 2567
 2568                                  ;*****************************************************************************
  (2)                                  .SBTTL      T74 -- TEST THAT A RESERVED (UNDEFINED) INSTRUCTION TRAPS TO 10
  (2)                                  ;*****************************************************************************
  (2)  025122  104420  000074          T74:    BEGINTEST, 74           ;; CHECK SWITCH REGISTER OPTIONS.
 2569  025126  012706  003776                  MOV     #STACK,SP       ;STACK POINTER SETUP
 2570  025132  012767  025162  152650          MOV     #2$,RESVEC      ; SET TRAP PC...
 2571  025140  005067  152646                  CLR     RESVEC+2        ;...AND PSW.
 2572  025144  106427  000317                  MTPS    #PRI6!17        ; SET CURRENT (OLD) PSW.
 2573  025150  000010                          RESRVD                  ; RESERVED INSTRUCTION -- SHOULD TRAP
 2574  025152  000240                  1$:     NOP
 2575  025154  104423  025122                  ERROR,  T74             ;; *** RESERVED OPCODE DIDN'T TRAP ***
 2576  025160  000424                          BR      6$
 2577
 2578  025162  010700                  2$:     MFPS    R0
 2579  025164  001402                          BEQ     3$              ; BR IF NEW PSW IS RIGHT.
 2580  025166  104423  025122                  ERROR,  T74             ;; *** NEW PSW INCORRECT AFTER TRAP TO 10 ***
 2581  025172  020627  003772          3$:     CMP     SP,#STACK-4
 2582  025176  001402                          BEQ     4$              ; BR IF SP PUSHED DOWN RIGHT.
 2583  025200  104423  025122                  ERROR,  T74             ;; *** STACK POINTER INCORRECT ***
 2584  025204  021627  025152          4$:     CMP     (SP),#1$
 2585  025210  001402                          BEQ     5$              ; BR IF STACKED PC IS RIGHT.
 2586  025212  104423  025122                  ERROR,  T74             ;; *** INCORRECT PC SAVED ON STACK ***
 2587  025216  026627  000002  000317  5$:     CMP     2(SP),#PRI6!17
 2588  025224  001402                          BEQ     6$              ; BR IF STACKED (OLD) PSW IS RIGHT.
 2589  025226  104423  025122                  ERROR,  T74             ;; *** INCORRECT PSW SAVED ON STACK ***
 2590
 2591  025232  012767  000012  152550  6$:     MOV     #RESVEC+2,RESVEC ; RESET VECTOR.
```

```
                                ;*****************************************************************************
                                .SBTTL      T75 -- TEST THAT A "BPT" INSTRUCTION TRAPS TO 14
                                ;*****************************************************************************
025240  104420  000076          T75:    BEGINTEST, 75           ;; CHECK SWITCH REGISTER OPTIONS.
025244  012706  003776                  MOV     #STACK,SP       ;STACK POINTER SETUP
025250  012767  025300  152536          MOV     #2$,BPTVEC      ; SET TRAP PC...
025256  005067  152534                  CLR     BPTVEC+2        ;...AND PSW.
025262  106427  000317                  MTPS    #PRI6!17        ; SET CURRENT (OLD) PSW.
025266  000003                          BPT                     ; BPT -- SHOULD TRAP.
025270  000240                  1$:     NOP
025272  104423  025240                  ERROR,  T75             ;; *** BPT DIDN'T TRAP ***
025276  000424                          BR      6$

025300  106700                  2$:     MFPS    R0
025302  001402                          BEQ     3$              ; BR IF NEW PSW IS RIGHT.
025304  104423  025240                  ERROR,  T75             ;; *** PSW INCORRECT AFTER BPT TRAP TO 14 ***
025310  020627  003772          3$:     CMP     SP,#STACK-4
025314  001402                          BEQ     4$              ; BR IF STACK IS RIGHT.
025316  104423  025240                  ERROR,  T75             ;; *** STACK POINTER INCORRECT ***
025322  021627  025270          4$:     CMP     (SP),#1$
025326  001402                          BEQ     5$              ; BR IF STACKED PC IS RIGHT.
025330  104423  025240                  ERROR,  T75             ;; *** INCORRECT PC SAVED ON STACK ***
025334  026627  000002  000317  5$:     CMP     2(SP),#PRI6!17
025342  001402                          BEQ     6$              ; BR IF STACKED (OLD) PSW IS RIGHT.
025344  104423  025240                  ERROR,  T75             ;; *** INCORRECT PSW SAVED ON STACK ***

025350  012767  000016  152436  6$:     MOV     #BPTVEC+2,BPTVEC ; RESET VECTOR.

                                ;*****************************************************************************
                                .SBTTL      T76 -- TEST THAT AN "IOT" INSTRUCTION TRAPS TO 20
                                ;*****************************************************************************
025356  104420  000076          T76:    BEGINTEST, 76           ;; CHECK SWITCH REGISTER OPTIONS.
025360  012706  003776                  MOV     #STACK,SP       ;STACK POINTER SETUP
025366  012767  025416  152424          MOV     #2$,IOTVEC      ; SET TRAP PC...
025374  005067  152422                  CLR     IOTVEC+2        ;...AND PSW.
025400  106427  000317                  MTPS    #PRI6!17        ; SET CURRENT (OLD) PSW.
025404  000004                          IOT                     ; IOT -- SHOULD TRAP.
025406  000240                  1$:     NOP
025410  104423  025356                  ERROR,  T76             ;; *** IOT DID NOT TRAP ***
025414  000424                          BR      6$

025416  106700                  2$:     MFPS    R0
025420  001402                          BEQ     3$              ; BR IF NEW PSW IS RIGHT.
025422  104423  025356                  ERROR,  T76             ;; *** PSW INCORRECT AFTER IOT TRAP TO 20 ***
025426  020627  003772          3$:     CMP     SP,#STACK-4
025432  001402                          BEQ     4$              ; BR IS STACK IS RIGHT.
025434  104423  025356                  ERROR,  T76             ;; *** STACK POINTER INCORRECT ***
025440  021627  025406          4$:     CMP     (SP),#1$
025444  001402                          BEQ     5$              ; BR IF STACKED PC IS RIGHT.
025446  104423  025356                  ERROR,  T76             ;; *** INCORRECT PC SAVED ON STACK ***
025452  026627  000002  000317  5$:     CMP     2(SP),#PRI6!17
025460  001402                          BEQ     6$              ; BR IF STACKED (OLD) PSW IS RIGHT.
025462  104423  025356                  ERROR,  T76             ;; *** INCORRECT PSW SAVED ON STACK ***

025466  012767  000022  152324  6$:     MOV     #IOTVEC+2,IOTVEC ; RESET VECTOR.
```

```
                                        ;*****************************************************************************
                                        .SBTTL      T77 -- TEST THAT 'POWER-FAIL' TRAPS TO 24 (REQUIRES Q-BUS EXERCISER)
                                        ;*****************************************************************************
025474  104420  000077          T77:    BEGINTEST, 77           ;; CHECK SWITCH REGISTER OPTIONS.
025500  012706  003776                  MOV     #STACK,SP       ;STACK POINTER SETUP
025504  004767  007552                  CALL    QBXIN           ; Q-BUS EXERCISOR AVAILABLE ??
025510  001456                          BEQ     7$              ; BYPASS TEST IF NOT.
025512  010004                          MOV     R0,R4           ; YES, MOVE POINTER TO R4 AND FALL THRU.
                                        ;
                                        ; QBX CSR1<13> IS USED TO EMULATE A POWER DOWN/UP SEQUENCE.
                                        ; CAUTION:
                                        ; WE ONLY WANT TO RECOGNIZE THE 'DOWN' HALF OF THE POWER SEQUENCE.
                                        ; ABORT THE PWR-CYCLE AT THAT POINT SO WE DON'T CRASH TO ODT.
                                        ;
025514  012767  025564  152302          MOV     #2$,PWRVEC      ; SET POWER TRAP PC...
025522  005067  152300                  CLR     PWRVEC+2        ;...AND PSW
025526  106427  000300                  MTPS    #PR16           ; SET CURRENT (OLD) PSW.
025532  012700  140000                  MOV     #140000,R0      ; IF LOOPING, WE HAVE TO WAIT LONG...
025536  077001                          SOB     R0,.            ;...ENOUGH (==250MSEC) TO INSURE THE...
                                                                ;...PREVIOUS BPOK CYCLE WAS FINISHED...
                                                                ;...(IN QBX) BEFORE TRYING ANOTHER.
025540  012700  000005                  MOV     #5,R0           ; OK, SET A 25 USEC TIMER.
025544  052714  020000                  BIS     #20000,(R4)     ; SET PWR-CYCLE REQUEST...
025550  000257                          CCC                     ;...SHOULD SEE IT AFTER THIS...
025552  077001          1$:     SOB     R0,.                    ;...AND STACK THIS PC.
025554  005014                  CLR     (R4)                    ; TRAP NOT RECEIVED, CLEAR THE REQUEST.
025556  104423  025474                  ERROR,  T77             ;; *** POWER-FAIL DID NOT TRAP ***
025562  000426                          BR      6$
                                        
025564  106700          2$:     MFPS    R0                      ; OK, GET NEW PSW...
025566  005014                  CLR     (R4)                    ;...AND CLEAR PWR-CYCLE REQUEST.
025570  005700                  TST     R0
025572  001402                  BEQ     3$                      ; BR IF NEW PSW IS RIGHT.
025574  104423  025474                  ERROR,  T77             ;; *** PSW INCORRECT AFTER PWR-FAIL TRAP TO 24 ***
025600  020627  003772          3$:     CMP     SP,#STACK-4
025604  001402                  BEQ     4$                      ; BR IS STACK IS RIGHT.
025606  104423  025474                  ERROR,  T77             ;; *** STACK POINTER INCORRECT ***
025612  021627  02555           4$:     CMP     (SP),#1$
025616  001402                  BEQ     5$                      ; BR IF STACKED PC IS RIGHT.
025620  104423  025474                  ERROR,  T77             ;; *** INCORRECT PC SAVED ON STACK ***
025624  026627  000002  000300  5$:     CMP     2(SP),#PR16
025632  001402                  BEQ     6$                      ; BR IF STACKED (OLD) PSW IS RIGHT.
025634  104423  025474                  ERROR,  T77             ;; *** INCORRECT PSW SAVED ON STACK ***
                                        
025640  012767  000026  152156  6$:     MOV     #PWRVEC+2,PWRVEC ; RESTORE VECTOR.
025646                          7$:
                                        
                                        ;*****************************************************************************
                                        .SBTTL      T100 -- TEST THAT AN 'EMT' INSTRUCTION TRAPS TO 30
                                        ;*****************************************************************************
025646  104420  000100          T100:   BEGINTEST, 100          ;; CHECK SWITCH REGISTER OPTIONS.
                                        ;
                                        ; NOW EMT'S HAVE TO BE OK OR WE'D NEVER HAVE GOTTEN STARTED IN THE FIRST
                                        ; PLACE.  WE'VE ALREADY VERIFIED ALL VARIATIONS OF THE EMT OPCODE,
                                        ; SO WE'LL JUST TEST FOR CORRECT STACK MANIPULATION HERE.
```

```
2692                                    ;
2693  025652  012706  003776            MOV     #STACK,SP       ;STACK POINTER SETUP
2694  025656  016701  152146            MOV     EMTVEC,R1       ; SAVE EMT VECTOR IN R1.
2695  025662  012767  025706  152140    MOV     #2$,EMTVEC      ; SET TRAP PC...
2696  025670  005067  152136            CLR     EMTVEC+2        ;...AND PSW.
2697  025674  106427  000317            MTPS    #PRI6!17        ; SET CURRENT (OLD) PSW.
2698  025700  104000                    EMT                     ; EMT -- SHOULD TRAP...
2699  025702  000240            1$:     NOP                     ;...AND STACK THIS PC.
2700  025704  000240                    NOP                     ; IF IT DOESN'T, JUST FALL THRU ANYWAY.
2701
2702  025706  106700            2$:     MFPS    R0              ; GET NEW PSW...
2703  025710  010167  152114            MOV     R1,EMTVEC       ;...AND RESTORE VECTOR.
2704  025714  005700                    TST     R0
2705  025716  001402                    BEQ     3$              ; BR IF NEW PSW IS RIGHT.
2706  025720  104423  025646            ERROR,  T100            ;; *** PSW INCORRECT AFTER EMT TRAP TO 30 ***
2707  025724  020627  003772    3$:     CMP     SP,#STACK-4
2708  025730  001402                    BEQ     4$              ; BR IF SP IS RIGHT.
2709  025732  104423  025646            ERROR,  T100            ;; *** STACK POINTER INCORRECT ***
2710  025736  021627  025702    4$:     CMP     (SP),#1$
2711  025742  001402                    BEQ     5$              ; BR IF STACKED PC IS RIGHT.
2712  025744  104423  025646            ERROR,  T100            ;; *** INCORRECT PC SAVED ON STACK ***
2713  025750  026627  000002  000317    5$:     CMP     2(SP),#PRI6!17
2714  025756  001402                    BEQ     6$              ; BR IF STACKED (OLD) PSW IS RIGHT.
2715  025760  104423  025646            ERROR,  T100            ;; *** INCORRECT PSW SAVED ON STACK ***
2716  025764                    6$:
2717
2718                                    ;*******************************************************************************
 (2)                                    .SBTTL      T101 -- TEST THAT A "TRAP" INSTRUCTION TRAPS TO 34
 (2)                                    ;*******************************************************************************
 (2)  025764  104420  000101            T101:   BEGINTEST, 101          ;; CHECK SWITCH REGISTER OPTIONS.
2719                                    ;
2720                                    ; WE ALREADY KNOW THAT ALL VARIATIONS OF THE TRAP OPCODE WORK.
2721                                    ; SO WE'LL JUST CHECK THAT THE STACK IS HANDLED PROPERLY.
2722                                    ;
2723  025770  012706  003776            MOV     #STACK,SP       ;STACK POINTER SETUP
2724  025774  016701  152034            MOV     TRPVEC,R1       ; SAVE TRAP VECTOR IN R1.
2725  026000  012767  026024  152026    MOV     #2$,TRPVEC      ; SET TRAP PC...
2726  026006  005067  152024            CLR     TRPVEC+2        ;...AND PSW.
2727  026012  106427  000317            MTPS    #PRI6!17        ; SET CURRENT (OLD) PSW.
2728  026016  104400                    TRAP+0                  ; TRAP -- SHOULD TRAP...
2729  026020  000240            1$:     NOP                     ;...AND STACK THIS PC.
2730  026022  000240                    NOP                     ; IF IT DOESN'T, JUST FALL THRU ANYWAY.
2731
2732  026024  106700            2$:     MFPS    R0              ; GET NEW PSW...
2733  026026  010167  152002            MOV     R1,TRPVEC       ;...AND RESTORE VECTOR.
2734  026032  005700                    TST     R0
2735  026034  001402                    BEQ     3$              ; BR IF NEW PSW IS RIGHT.
2736  026036  104423  025764            ERROR,  T101            ;; *** PSW INCORRECT AFTER TRAP TO 34 ***
2737  026042  020627  003772    3$:     CMP     SP,#STACK-4
2738  026046  001402                    BEQ     4$              ; BR IF STACK IS RIGHT.
2739  026050  104423  025764            ERROR,  T101            ;; *** STACK POINTER INCORRECT ***
2740  026054  021627  026020    4$:     CMP     (SP),#1$
2741  026060  001402                    BEQ     5$              ; BR IF STACKED PC IS RIGHT.
2742  026062  104423  025764            ERROR,  T101            ;; *** INCORRECT PC SAVED ON STACK ***
2743  026066  026627  000002  000317    5$:     CMP     2(SP),#PRI6!17
2744  026074  001402                    BEQ     6$              ; BR IF STACKED (OLD) PSW IS RIGHT.
```

```
                                        PC 26076 = INCORRECT PSW SAVED ON STACK.

026076  104423  025764                         ERROR,  T101            ;; *** INCORRECT PSW SAVED ON STACK ***
026102                                6$:

                                      ;*************************************************************************
                                      .SBTTL      T102 -- TEST THAT THE "T" BIT CAUSES A TRACE TRAP
                                      ;*************************************************************************
026102  104420  000102                T102:   BEGINTEST, 102          ;; CHECK SWITCH REGISTER OPTIONS.
                                      ;
                                      ; FIRST, SET "T" VIA RTI -- SHOULD TRAP TO 14 BEFORE NEXT INSTRUCTION.
                                      ;
026106  012706  003776                        MOV     #STACK,SP
026112  012767  026150  151674                MOV     #3$,TRTVEC      ; SET TRAP PC...
026120  005067  151672                        CLR     TRTVEC+2        ;...AND NEW PSW.
026124  012746  000020                        MOV     #20,-(SP)       ; PUSH OLD PSW WITH T BIT...
026130  012746  026136                        MOV     #.+6,-(SP)      ;...AND PC.
026134  000002                              RTI                     ; SHOULD SET T BIT...
026136  000240                        1$:   NOP                     ;...AND TRAP BEFORE THIS EXECUTES.
026140  000240                        2$:   NOP
026142  104423  026102                        ERROR,  T102            ;; *** TRACE BIT DID NOT TRAP ***
026146  000417                              BR      6$

026150  106700                        3$:   MFPS    R0                ; TRAPPED OK, T SHOULD BE CLEAR NOW.
026152  001402                              BEQ     4$                ; BR IF SO.
026154  104423  026102                        ERROR,  T102            ;; *** NEW PSW INCORRECT AFTER TRACE TRAP TO 14 ***
026160  021627  026136                4$:   CMP     (SP),#1$          ; IT SHOULD HAVE TRAPPED BEFORE...
026164  001402                              BEQ     5$                ;...THE INSTRUCTION AT 1$.
026166  104423  026102                        ERROR,  T102            ;; *** TRACE TRAP HAPPENED AT WRONG TIME ***
026172  026627  000002  000020        5$:   CMP     2(SP),#20         ; AND T BIT SHOULD HAVE BEEN STACKED...
026200  001402                              BEQ     6$                ;...ALONG WITH THE OLD PSW.
026202  104423  026102                        ERROR,  T102            ;; *** T BIT WASN'T SAVED IN STACKED PSW ***
                                      ;
                                      ; NOW SET "T" VIA RTT -- SHOULD ALLOW NEXT INSTRUCTION, THEN TRAP.
                                      ;
026206  012767  026240  151600        6$:   MOV     #8$,TRTVEC        ; CHANGE VECTOR.
026214  012746  000020                      MOV     #20,-(SP)         ; PUSH OLD PSW WITH T BIT...
026220  012746  026226                      MOV     #.+6,-(SP)        ;...AND PC.
026224  000006                              RTT                     ; SHOULD SET T BIT...
026226  000240                              NOP                     ;...EXECUTE THIS...
026230  000240                        7$:   NOP                     ;...AND TRAP BEFORE THIS.
026232  104423  026102                      ERROR,  T102            ;; *** TRACE TRAP DIDN'T HAPPEN ***
026236  000417                              BR      11$

026240  106700                        8$:   MFPS    R0                ; OK, T SHOULD BE CLEAR NOW.
026242  001402                              BEQ     9$
026244  104423  026102                      ERROR,  T102            ;; *** NEW PSW INCORRECT AFTER TRACE TRAP TO 14 ***
026250  021627  026230                9$:   CMP     (SP),#7$          ; AND IT SHOULD HAVE ALLOWED...
026254  001402                              BEQ     10$               ;...THE 1ST NOP TO EXECUTE.
026256  104423  026102                      ERROR,  T102            ;; *** RTT DIDN'T ALLOW NEXT INSTRUCTION TO EXECUTE ***
026262  026627  000002  000020        10$:  CMP     2(SP),#20         ; AND THE T BIT SHOULD BE ON STACK.
026270  001402                              BEQ     11$
026272  104423  026102                      ERROR,  T102            ;; *** T BIT WASN'T SAVED IN STACKED PSW ***
026276                                11$:

                                      ;*************************************************************************
                                      .SBTTL      T103 -- TEST THAT TRACE TRAP ON A TRAP IS INHIBITED
                                      ;*************************************************************************
```

```
  (2)  026276  104420  000103          T103:  BEGINTEST, 103          ;; CHECK SWITCH REGISTER OPTIONS.
2796   026302  012706  003776                 MOV     #STACK,SP
2797   026306  012767  026360  151500         MOV     #1$,TRTVEC      ; SET TRACE VECTOR...
2798   026314  005067  151476                 CLR     TRTVEC+2        ;...AND PSW.
2799   026320  012767  026370  151472         MOV     #2$,IOTVEC      ; SET IOT VECTOR...
2800   026326  005067  151470                 CLR     IOTVEC+2        ;...AND PSW.
2801   026332  012746  000020                 MOV     #20,-(SP)       ; PUSH PSW WITH T BIT...
2802   026336  012746  026344                 MOV     #.+6,-(SP)      ;...AND PC.
2803   026342  000006                         RTT                     ; SET T BIT...
2804   026344  000004                         IOT                     ;...AND XCT ANOTHER TRAP...
2805   026346  000240                         NOP                     ; THE IOT SHOULD WIN...
2806   026350  000240                         NOP                     ;...AND INHIBIT THE TRACE.
2807   026352  104423  026276                 ERROR,  T103            ;; *** NEITHER TRAP OCCURRED ??? ***
2808   026356  000404                         BR      2$
2809
2810   026360  000240          1$:    NOP                             ; IOT SHOULD HAVE CLEARED T...
2811   026362  000240                 NOP                             ;...AND INHIBITED THE TRACE.
2812   026364  104423  026276                 ERROR,  T103            ;; *** TRACE TRAP ON TRAP WASN'T INHIBITED ***
2813
2814   026370  012767  000022  151422  2$:    MOV     #IOTVEC+2,IOTVEC ; RESET IOT VECTOR.
2815
2816                                  ;*******************************************************************
  (2)                                .SBTTL      T104 -- TEST THAT RESET HAS NO EFFECT ON A TRACE TRAP
  (2)                                ;*******************************************************************
  (2)  026376  104420  000104          T104:  BEGINTEST, 104          ;; CHECK SWITCH REGISTER OPTIONS.
2817   026402  012706  003776                 MOV     #STACK,SP       ;SET STACK
2818   026406  012767  026446  151400         MOV     #1$,TRTVEC      ; SET TRACE VECTOR.
2819   026414  005067  151376                 CLR     TRTVEC+2
2820   026420  012746  000020                 MOV     #20,-(SP)       ; PUSH PSW WITH T BIT...
2821   026424  012746  026432                 MOV     #.+6,-(SP)      ;...AND PC.
2822   026430  000006                         RTT                     ; SET "T" BIT...
2823   026432  000005                         RESET                   ;...AND XCT BUS RESET.
2824   026434  000240                         NOP                     ; SHOULD HAVE NO EFFECT...
2825   026436  000240                         NOP                     ;... I.E. TRAP SHOULD STILL OCCUR...
2826   026440  000240                         NOP                     ;...AFTER THE RESET INSTRUCTION.
2827   026442  104423  026376                 ERROR,  T104            ;; *** BUS-RESET DISABLED OR INHIBITED TRACE TRAP ***
2828
2829   026446  106427  000000          1$:    MTPS    #PRIO           ; WE'RE ALL DONE.
2830   026452  012767  000016  151334         MOV     #TRTVEC+2,TRTVEC ; RESET TRACE VECTOR.
2831
2832                                  ;*******************************************************************
  (2)                                .SBTTL      T105 -- TEST THAT ODD ADDRESS TRAPS ARE NOT IMPLEMENTED
  (2)                                ;*******************************************************************
  (2)  026460  104420  000105          T105:  BEGINTEST, 105          ;; CHECK SWITCH REGISTER OPTIONS.
2833   026464  012706  003776                 MOV     #STACK,SP       ; SET STACK.
2834   026470  012767  026514  151306         MOV     #2$,ERRVEC      ; IN CASE ODD ADDRESS TRAPS.
2835   026476  000167  000007                 JMP     1$+1
2836   026502  000240                         NOP                     ; IT DIDN'T DO ANYTHING AT ALL !!
2837   026504  104423  026460                 ERROR,  T105            ;; *** JUMP TO AN ODD ADDRESS DID NOTHING ***
2838   026510  012707  026520          1$:    MOV     #3$,PC          ; ADJUST ODD PC AND EXIT.
2839   026514                         2$:
  (2)  026514  104423  026460                 ERROR,  T105            ;; *** JMP TO AN ODD ADDRESS TIMED OUT ***
2840   026520  012767  000006  151256  3$:    MOV     #ERRVEC+2,ERRVEC ; RESET THE TRAP CATCHER.
2841
2842                                  ;*******************************************************************
  (2)                                .SBTTL      T106 -- TEST THE CPU TYPE INSTRUCTION (MFPT)
```

```
  (2)                                   ;*****************************************************************************
  (2)  026526  104420  000106           T106:   BEGINTEST, 106          ;; CHECK SWITCH REGISTER OPTIONS.
 2843  026532  012706  003776                   MOV     #STACK,SP       ; SET STACK.
 2844  026536  012767  026602  151244           MOV     #2$,RESVEC      ; SET TRAP CATCHER.
 2845  026544  005000                           CLR     R0
 2846  026546  000257                           CCC                     ; CLEAR CC BITS.
 2847  026550  000007                           MFPT                    ; MOVE FROM PROC TYPE (TO R0).
 2848  026552  106701                           MFPS    R1
 2849  026554  032701  000017                   BIT     #17,R1
 2850  026560  001402                           BEQ     1$
 2851  026562  104423  026526                   ERROR,  T106            ;; *** CC BITS CHANGED ON MFPT INSTRUCTION ***
 2852  026566  020027  000004           1$:     CMP     R0,#KXT11
 2853  026572  001405                           BEQ     3$
 2854  026574  104423  026526                   ERROR,  T106            ;; *** RETURNED VALUE INCORRECT ON MFPT ***
 2855  026600  000402                           BR      3$
 2856  026602                           2$:
  (2)  026602  104423  026526                   ERROR,  T106            ;; *** MFPT TRAPPED TO 10 ***
 2857  026606  012767  000012  151174   3$:     MOV     #RESVEC+2,RESVEC
 2858
 2859                                   ;*****************************************************************************
  (2)                                   .SBTTL      T107 -- TEST EXTERNAL Q-BUS INTERRUPT (REQUIRES Q-BUS EXERCISER)
  (2)                                   ;*****************************************************************************
  (2)  026614  104420  000107           T107:   BEGINTEST, 107          ;; CHECK SWITCH REGISTER OPTIONS.
 2860  026620  012706  003776                   MOV     #STACK,SP
 2861  026624  004767  006432                   CALL    QBXIN           ; Q-BUS EXERCISOR AVAILABLE ??
 2862  026630  001444                           BEQ     4$              ; IF NOT, BYPASS TEST.
 2863  026632  010004                           MOV     R0,R4           ; YES, CSR1 POINTER => R4 AND FALL THRU.
 2864                                   ;
 2865                                   ; USE QBX TO FAKE A Q-BUS INTERRUPT (NO DMA).
 2866                                   ;
 2867  026634  012703  000110                   MOV     #QBXVEC,R3      ; GET VECTOR POINTER.
 2868  026640  005014                           CLR     (R4)            ; CLEAR CSR1 (NO DMA REQUIRED).
 2869  026642  012713  026706                   MOV     #1$,(R3)        ; SET Q-BUS VECTOR...
 2870  026646  012763  000200  000002           MOV     #PRI4,2(R3)     ;...AND PRIORITY.
 2871  026654  106427  000200                   MTPS    #PRI4           ; RAISE CPU TO Q-BUS LEVEL.
 2872  026660  012764  000001  000002           MOV     #1,2(R4)        ; SET QBX GO (CLEAR DONE)...
 2873  026666  052764  000020  000002           BIS     #20,2(R4)       ;...ENABLE INT 4 ON DONE...
 2874  026674  042764  000001  000002           BIC     #1,2(R4)        ;...AND CLEAR GO (SET DONE).
 2875  026702  000241                           240                     ; INTERRUPT SHOULD BE HELD OFF.
 2876  026704  000401                           BR      2$
 2877  026706                           1$:
  (2)  026706  104423  026614                   ERROR,  T107            ;; *** Q-BUS INTERRUPTS AT WRONG LEVEL ***
 2878  026712  000407                           BR      3$
 2879  026714  012713  026732           2$:     MOV     #3$,(R3)        ; CHANGE THE VECTOR.
 2880  026720  106427  000000                   MTPS    #PRI0           ; LOWER CPU...
 2881  026724  000240                           240                     ;...INTERRUTPT SHOULD COME IN.
 2882  026726  104423  026614                   ERROR,  T107            ;; *** Q-BUS INTERRUPT NOT RECEIVED ***
 2883  026732  005014                   3$:     CLR     (R4)            ; THAT'S ALL THERE IS TO IT.
 2884  026734  012723  000112                   MOV     #QBXVEC+2,(R3)+ ; RESET Q-BUS VECTOR.
 2885  026740  005013                           CLR     (R3)
 2886  026742                           4$:
 2887
 2888                                   ;*****************************************************************************
  (2)                                   .SBTTL      T110 -- TEST THE "BEVNT" INTERRUPT (REQUIRES Q-BUS EXERCISER)
  (2)                                   ;*****************************************************************************
  (2)  026742  104420  000110           T110:   BEGINTEST, 110          ;; CHECK SWITCH REGISTER OPTIONS.
```

```
 2889  026746  012706  003776                MOV     #STACK,SP       ; SET STACK POINTER.
 2890  026752  004767  006304                CALL    QBXIN           ; Q-BUS EXERCISOR AVAILABLE ??
 2891  026756  001433                       BEQ     4$              ; IF NOT, BYPASS TEST.
 2892  026760  010004                       MOV     R0,R4           ; YES, CSR1 POINTER => R4 AND FALL THRU.
 2893                                ;
 2894                                ; QBX CSR1<11> EMULATES A "BEVNT" INTERRUPT.
 2895                                ;
 2896  026762  106427  000300               MTPS    #PRI6           ; SET PRI 06
 2897  026766  012767  027006  151104       MOV     #1$,BEVNT       ; SET BEVNT VECTOR.
 2898  026774  005000                       CLR     R0
 2899  026776  012714  004000               MOV     #4000,(R4)      ; SET "BEVNT" REQUEST.
 2900  027002  077001                       SOB     R0,.            ; DELAY ABOUT 250 MSEC, INTERRUPT...
 2901  027004  000403                       BR      2$              ;...SHOULD BE MASKED AT THIS LEVEL.
 2902  027006                        1$:
  (2)  027006  104423  026742               ERROR,  T110            ;; *** BEVNT INTERRUPT LEVEL INCORRECT ***
 2903  027012  000410                       BR      3$
 2904  027014  012767  027034  151056  2$:  MOV     #3$,BEVNT       ; OK, CHANGE VECTOR...
 2905  027022  106427  000240               MTPS    #PR15           ;...AND LOWER CPU TO PRI 5.
 2906  027026  077001                       SOB     R0,.            ; DELAY AGAIN, INT SHOULD COME IN.
 2907  027030  104423  026742               ERROR,  T110            ;; *** BEVNT INTERRUPT NOT RECEIVED ***
 2908  027034  012767  000102  151036  3$:  MOV     #BEVNT+2,BEVNT  ; RESET VECTOR.
 2909  027042  106427  000000               MTPS    #PRI0           ; LOWER CPU TO 0.
 2910
 2911  027046  004767  00565o          4$:  CALL    ENDSEG          ; CHECK FOR LOOP-IN-SEGMENT...
 2912                                                               ;...RETURN AND FALL THRU IF NOT.
 2913                                .SBTTL
```

```
                                        .SBTTL  SECTION 3 -- LOCAL 2K RAM TESTS.

027052  012767  036245  007120  S3:     MOV     #SEG3,SEGN      ; SEGMENT 3 TEXT.
027060  012767  160010  151720          MOV     #RAMADR,TEMP    ; SAVE RAM ADDRESS...
027066  012767  003764  151714          MOV     #RAMSIZ,TEMP1   ;...AND SIZE.

                                        ;*******************************************************************************
                                        .SBTTL      T111 -- LOCAL 2K RAM ADDRESS TEST
                                        ;*******************************************************************************
027074  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.
                                        ;
027100  012706  003776                  MOV     #STACK,SP
027104  012767  027206  150672          MOV     #7$,ERRVEC      ; SET TIME-OUT VECTOR.
027112  016700  151670                  MOV     TEMP,R0         ; GET RAM ADDRESS...
027116  016701  151666                  MOV     TEMP1,R1        ;...AND SIZE.
027122  010146                          MOV     R1,-(SP)        ; SAVE A COPY.

027124  010010          1$:     MOV     R0,(R0)         ; LOAD ADDRESSES (FROM BOTTOM UP).
027126  005720                  TST     (R0)+           ; BUMP
027130  077103                  SOB     R1,1$           ; LOOP 'TIL DONE.

027132  011602                  MOV     (SP),R2         ; RECOVER SIZE K.
027134  014001          2$:     MOV     -(R0),R1        ; READ 'EM BACK (FROM TOP DOWN).
027136  020100                  CMP     R1,R0           ; DATA = ADDRESS ??
027140  001402                  BEQ     3$              ; BR IF SO.
027142  104423  027074          ERROR,  T111            ;; *** RAM DATA INCORRECT AT ADDRESS IN R0 ***
027146  077206          3$:     SOB     R2,2$           ; LOOP 'TIL DONE.

027150  011602                  MOV     (SP),R2         ; RECOVER SIZE K.
027152  010001          4$:     MOV     R0,R1           ; ADDRESS => R1...
027154  000301                  SWAB    R1              ;...SWAP BYTES...
027156  010120                  MOV     R1,(R0)+        ;...AND LOAD SWAPPED ADDRESSES.
027160  077204                  SOB     R2,4$

027162  011603                  MOV     (SP),R3         ; RECOVER SIZE K.
027164  014001          5$:     MOV     -(R0),R1        ; GET DATA.
027166  010002                  MOV     R0,R2
027170  000302                  SWAB    R2
027172  020102                  CMP     R1,R2           ; DATA = SWAPPED ADDRESS ??
027174  001402                  BEQ     6$              ; BR IF SO.
027176  104423  027074          ERROR,  T111            ;; *** RAM DATA (SWAPPED) INCORRECT AT ADDRESS IN R0 ***
027202  077310          6$:     SOB     R3,5$
027204  000402                  BR      8$              ; DONE, EXIT.

027206          7$:
027206  104423  027074          ERROR,  T111            ;; *** BUS TIME-OUT -- RAM ADDRESS IN R0 ***
027212          8$:

                                        ;*******************************************************************************
                                        .SBTTL      T112 -- LOCAL 2K RAM DATA TEST
                                        ;*******************************************************************************
027212  104420  000112          T112:   BEGINTEST, 112          ;; CHECK SWITCH REGISTER OPTIONS.
                                        ;
```

```
                                        ; FILL RAM WITH A BACKGROUND PATTERN OF 0'S.
                                        ; FLOAT AND VERIFY A 1 BIT THRU EACH WORD, AND WHEN DONE
                                        ; RESTORE THAT WORD TO THE BACKGROUND.
                                        ; WHEN ALL WORDS DONE, VERIFY THAT THE ENTIRE RAM STILL CONTAINS
                                        ; THE CORRECT BACKGROUND DATA.
                                        ;
                                        ; REPEAT A SECOND TIME WITH BACKGROUND 1, AND A FLOATING 0.
                                        ;
027216  012706  003776                  MOV     #STACK,SP
027222  012767  027376  150554          MOV     #10$,ERRVEC     ; SET TIME-OUT VECTOR.
027230  005046                          CLR     -(SP)           ; BACKGROUND = 0'S FIRST...
027232  000402                          SKP2
027234  012746  177777          1$:     MOV     #-1,-(SP)       ;...THEN 1'S.
027240  016700  151542                  MOV     TEMP,R0         ; SET RAM ADDRESS...
027244  016701  151540                  MOV     TEMP1,R1        ;...AND SIZE.
027250  011620                  2$:     MOV     (SP),(R0)+      ; FILL RAM WITH BACKGROUND DATA.
027252  077102                          SOB     R1,2$
                                        
027254  016700  151526                  MOV     TEMP,R0         ; RESET ADDRESS...
027260  016701  151524                  MOV     TEMP1,R1        ;...AND WORD COUNT...
027264  011004                  3$:     MOV     (R0),R4         ;...AND EXERCISE 1ST/NEXT LOCATION.
027266  020416                          CMP     R4,(SP)
027270  001402                          BEQ     4$              ; BR IF BACKGROUND IS RIGHT.
027272  104423  027264                  ERROR,  3$              ;; *** INITIAL BACKGROUND INCORRECT AT ADDRESS IN R0 ***
027276  012702  000001          4$:     MOV     #1,R2
027302  011603                          MOV     (SP),R3         ; COPY CURRENT BACKGROUND TO R3.
027304  074203                          XOR     R2,R3           ; SET OR CLEAR BIT 0.
027306  012702  000020                  MOV     #16.,R2         ; SET BIT COUNTER.
027312  000402                          SKP2
027314  006303                  5$:     ASL     R3              ; ROTATE DATA...
                                          ADC   R3              ;...BIT 15 => BIT 0.
027320  010310                  6$:     MOV     R3,(R0)         ; WRITE FLOATING DATA...
027322  011004                          MOV     (R0),R4         ;...AND READ IT BACK.
027324  020304                          CMP     R3,R4
027326  001402                          BEQ     7$              ; BR IF ITS RIGHT.
027330  104423  027320                  ERROR,  6$              ;; *** FLOATING DATA INCORRECT AT ADDRESS IN R0 ***
027334  077211                  7$:     SOB     R2,5$           ; LOOP FOR ALL BITS.
027336  011620                          MOV     (SP),(R0)+      ; THEN RESTORE, BUMP ADDRESS...
027340  077127                          SOB     R1,3$           ;...AND LOOP FOR ALL WORDS.
                                        
027342  016700  151440                  MOV     TEMP,R0         ; NOW, RESET POINTERS...
027346  016701  151436                  MOV     TEMP1,R1        ;...AND VERIFY THAT RESTORED..
027352  011004                  8$:     MOV     (R0),R4         ;...BACKGROUND IS STILL CORRECT.
027354  020416                          CMP     R4,(SP)
027356  001402                          BEQ     9$              ; BR IF SO.
027360  104423  027352                  ERROR,  8$              ;; *** BACKGROUND CHANGED AT ADDRESS IN R0 ***
027364  005720                  9$:     TST     (R0)+           ; BUMP...
027366  077107                          SOB     R1,8$           ;...AND LOOP 'TIL DONE.
                                        
027370  005726                          TST     (SP)+           ; FINALLY, IF BACKGROUND IS 0...
027372  001720                          BEQ     1$              ;...LOOP ONCE MORE...
027374  000402                          BR      11$             ;...OTHERWISE, WE'RE ALL DONE.
                                        
027376                          10$:
(2) 027376  104423  027212              ERROR,  T112            ;; *** BUS TIME-OUT -- RAM ADDRESS IN R0 ***
027402                          11$:
```

```
                        ;*****************************************************************************
                        .SBTTL      T113 -- OPTIONAL DMA TEST (REQUIRES Q-BUS EXERCISER)
                        ;*****************************************************************************
027402  104420  000113  T113:   BEGINTEST, 113          ; CHECK SWITCH REGISTER OPTIONS.
027406  012706  003776          MOV     #STACK,SP
027412  004767  005644          CALL    QBXIN           ; Q-BUS EXERCISOR AVAILABLE ??
027416  001504                  BEQ     10$             ; IF NOT, BYPASS TEST.
027420  010004                  MOV     R0,R4           ; YES, CSR1 POINTER => R4 AND FALL THRU.
                        ;
                        ; TRANSFER EACH RAM WORD TO THE QBX AND BACK.
                        ; VERIFY THAT RETURNED DATA MATCHES THAT WHICH WAS SENT.
                        ;
027422  012767  027624  150354          MOV     #9$,ERRVEC      ; SET TRAP CATCHER.
027430  016701  151352                  MOV     TEMP,R1         ; RAM ADDRESS POINTER...
027434  016702  151350                  MOV     TEMP1,R2        ;...AND WORD COUNT.

027440  010111          2$:     MOV     R1,(R1)         ; SET 1ST/NEXT RAM WORD.
027442  012714  140407          MOV     #140407,(R4)    ; DATI (MEM => QBX) IN "HOG" MODE.
027446  010164  000004          MOV     R1,4(R4)        ; FROM RAM (R1) TO DATA REGISTER...
027452  012764  177774  000006  MOV     #-4,6(R4)       ;...4 TIMES.
027460  012764  000401  000002  MOV     #401,2(R4)      ; GO, INHIBIT BA+.
027466  005000                  CLR     R0
027470  105764  000002  3$:     TSTB    2(R4)
027474  100401                  BMI     33$             ; PROCEED WHEN DONE...
027476  077004                  SOB     R0,3$
027500  032764  040000  000002  33$:    BIT     #40000,2(R4)    ;...AND CHECK FOR NXM ERROR.
027506  001402                  BEQ     4$              ; BR IF OK.
027510  104423  027440          ERROR,  2$              ;; *** NXM ERROR IN QBX ON DATI (MEM TO QBX) ***
027514  026401  000010  4$:     CMP     10(R4),R1       ; DATA REGISTER CORRECT ??
027520  001402                  BEQ     5$
027522  104423  027440          ERROR,  2$              ;; *** DATA INCORRECT IN QBX ON DATI (MEM TO QBX) ***

027526  012714  000601  5$:     MOV     #000601,(R4)    ; DATO (QBX => MEM) IN "HOG" MODE.
027532  012764  001012  000004  MOV     #TEMP2,4(R4)    ; FROM DATA REGISTER TO TEMP2...
027540  012764  177774  000006  MOV     #-4,6(R4)       ;...4 TIMES.
027546  012764  000401  000002  MOV     #401,2(R4)      ; GO, INHIBIT BA+.
027554  005000                  CLR     R0
027556  105764  000002  6$:     TSTB    2(R4)
027562  100401                  BMI     66$             ; PROCEED WHEN DONE...
027564  077004                  SOB     R0,6$
027566  032764  040000  000002  66$:    BIT     #40000,2(R4)    ;...AND CHECK FOR NXM.
027574  001402                  BEQ     7$              ; BR IF OK.
027576  104423  027526          ERROR,  5$              ;; *** NXM ERROR IN QBX ON DATO (QBX TO MEM) ***
027602  026701  151204  7$:     CMP     TEMP2,R1        ; RETURNED DATA CORRECT ??
027606  001402                  BEQ     8$
027610  104423  027526          ERROR,  5$              ;; *** DATA INCORRECT IN TEMP2 ON DATO (QBX TO MEM) ***

027614  005721          8$:     TST     (R1)+           ; SET NEXT ADDRESS...
027616  077270                  SOB     R2,2$           ;...AND LOOP 'TIL DONE.
027620  005014                  CLR     (R4)            ; CLEAR QBX...
027622  000402                  BR      10$             ;...AND EXIT.

027624          9$:
027624  104423  027402          ERROR,  T113            ;; *** BUS TIME-OUT -- QBX (R4) OR RAM (R1) ***
```

```
 3071  027630  012767  000006  150146  10$:    MOV     #ERRVEC+2,ERRVEC ; RESET ERROR VECTOR.
 3072  027636  004767  005066                  CALL    ENDSEG           ; CHECK FOR LOOP-IN-SEGMENT...
 3073                                                                   ;...RETURN AND FALL THRU IF NOT.
 3074                                  .SBTTL
```

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

027642  012767  036263  006330  S4:     MOV     #SEG4,SEGN      ; SEGMENT 4 TEXT.
027650  012767  170000  151130          MOV     #ROMADR,TEMP    ; SAVE ROM ADDRESS...
027656  012767  002000  151124          MOV     #ROMSIZ,TEMP1   ;...AND SIZE.

                                        ;*******************************************************************************
                                        .SBTTL      T114 -- LOCAL PROM/RAM ADDRESS TEST
                                        ;*******************************************************************************
027664  104420  000114          T114:   BEGINTEST, 114          ;; CHECK SWITCH REGISTER OPTIONS.
                                        ;
                                        ; PROM/RAM SOCKETS CONTAIN THE 1KW MACRO ODT PROM SET.
                                        ; ATTEMPT TO WRITE (123456) INTO EACH LOCATION.
                                        ; IF AN ADDRESS HAPPENS TO CONTAIN THAT VALUE, THEN WRITE AGAIN
                                        ; WITH THE COMPLIMENT OF THAT VALUE.
                                        ;
027670  012706  003776                  MOV     #STACK,SP
027674  012767  027746  150102          MOV     #3$,ERRVEC      ; SET TIME-OUT VECTOR.
027702  016700  151100                  MOV     TEMP,R0         ; SET ADDRESS...
027706  016701  151076                  MOV     TEMP1,R1        ;...AND SIZE.

027712  012702  123456          1$:     MOV     #123456,R2
027716  010210                          MOV     R2,(R0)         ; T   TO WRITE DATA (SHOULDN'T TRAP).
027720  021002                          CMP     (R0),R2         ; D   IT WRITE ??
027722  001006                          BNE     2$              ; BR IF NOT.
027724  005102                          COM     R2              ; MAYBE...
027726  010210                          MOV     R2,(R0)         ;...TRY COMPLIMENT.
027730  021002                          CMP     (R0),R2         ; HOW NOW ??
027732  001002                          BNE     2$              ; BR IF OK.
027734  104423  027712                  ERROR,  1$              ;; *** WRITE TO ROM ALTERED DATA ????? ***

027740  005720          2$:     TST     (R0)+           ; BUMP POINTER.
027742  077115                  SOB     R1,1$           ; LOOP 'TIL DONE...
027744  000402                  BR      4$              ;...AND PROCEED.

027746          3$:
027746  104423  027664                  ERROR,  T114            ;; *** BUS TIME-OUT -- ROM ADDRESS IN R0 ***

027752  012767  000006  150024  4$:     MOV     #ERRVEC+2,ERRVEC
027760  004767  004744                  CALL    ENDSEG          ; CHECK FOR LOOP-IN-SEGMENT...
                                                                ;...RETURN AND FALL THRU IF NOT.

                                        .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 SLUT).
                                        ;
027764  012767  036301  006206  S5:     MOV     #SEG5,SEGN      ; SEGMENT 5 TEXT.
027772  012700  177560                  MOV     #SLU1,R0        ; 1ST UNIT CSR...
027776  012701  000060                  MOV     #SLU1V,R1       ;...AND VECTOR(S).
030002  012702  000200                  MOV     #SLU1P,R2       ; SLU1 INTERRUPTS AT LEVEL 4.
030006  012703  030152                  MOV     #BADU1,R3       ; TIME-OUT VECTOR.
030012  000413                          BR      DLSET           ; SET-UP POINTERS AND XCT.
                                        ;
                                        ; AND THEN ON SLU2.
                                        ;
030014  012767  036317  006156  S6:     MOV     #SEG6,SEGN      ; SEGMENT 6 TEXT.
030022  012700  176540                  MOV     #SLU2,R0        ; 2ND UNIT CSR...
030026  012701  000120                  MOV     #SLU2V,R1       ;...AND VECTOR.
030032  012702  000240                  MOV     #SLU2P,R2       ; SLU2 INTERRUPTS AT LEVEL 5.
030036  012703  030162                  MOV     #BADU2,R3       ; TIME-OUT VECTOR.

030042                          DLSET:
                                ;*******************************************************************
                                .SBTTL      T115 -- TEST SERIAL LINE REGISTER ADDRESSES
                                ;*******************************************************************
030042  104420  000115          T115:   BEGINTEST, 115          ;; CHECK SWITCH REGISTER OPTIONS.
030046  012706  003776                  MOV     #STACK,SP       ; INIT THE STACK.
030052  010367  147726                  MOV     R3,ERRVEC       ; SET BUS TIME OUT TRAP CATCHER.
030056  010067  150762                  MOV     R0,RCSR         ; LOAD UP THE REGISTER POINTERS.
030062  005720                          TST     (R0)+           ; IF ANY OF THESE TRAP, WE'RE DEAD !!
030064  010067  150776                  MOV     R0,RBUF
030070  005720                          TST     (R0)+
030072  010067  150752                  MOV     R0,XCSR
030076  005720                          TST     (R0)+
030100  010067  150746                  MOV     R0,XBUF
030104  010167  150744                  MOV     R1,RVEC         ; SAVE RCVR VECTOR POINTER...
030110  012721  030172                  MOV     #BADRI,(R1)+    ;...AND INIT VECTOR.
030114  010221                          MOV     R2,(R1)+
030116  010167  150734                  MOV     R1,XVEC         ; SAVE XMTR VECTOR POINTER...
030122  012721  030202                  MOV     #BADTI,(R1)+    ;...AND INT VECTOR.
030126  010221                          MOV     R2,(R1)+
030130  017700  150714                  MOV     @XCSR,R0        ; NOW GET THE DEFAULT XMIT STATUS...
030134  042700  177705                  BIC     #^C72,R0        ;...AND STRIP THE BAUD RATE BITS.
```

```
3168  030140  052700  000002              BIS     #2,R0           ; INSURE 'PBRE' IS SET...
3169  030144  010067  150670              MOV     R0,DFPBR        ;...AND SAVE AS THE DEFAULT BAUD RATE.
3170  030150  000432                      BR      DLTSTS          ; START 'EM UP.
3171                                      ;
3172                                      ; COME HERE IF BUS TIME-OUT OR UNEXPECTED INTERRUPT.
3173                                      ;
3174  030152                            BADU1:
 (1)  030152  104423  027764                      ERROR,  S5              ;; *** BUS TIME-OUT ON SLU1 ADDRESS ***
3175  030156  000167  003270                      JMP     DLDUN
3176  030162                            BADU2:
 (1)  030162  104423  030014                      ERROR,  S6              ;; *** BUS TIME-OUT ON SLU2 ADDRESS ***
3177  030166  000167  003260                      JMP     DLDUN
3178  030172                            BADRI:
 (1)  030172  104423  030236                      ERROR,  DLTSTS          ;; *** UNEXPECTED RCVR INTERRUPT. ***
3179  030176  000167  003250                      JMP     DLDUN
3180  030202                            BADTI:
 (1)  030202  104423  030236                      ERROR,  DLTSTS          ;; *** UNEXPECTED XMTR INTERRUPT. ***
3181  030206  000167  003240                      JMP     DLDUN
3182                                      ;
3183                                      ; FLOATING 1 AND 0 DATA TABLE, USED IN VARIOUS TESTS.
3184                                      ;
3185  030212     015     012          SYNC:   .BYTE   015, 012                        ; SYNC PAIR.
3186  030214     001     002     004  FLT1:   .BYTE   001, 002, 004, 010, 020, 040, 100, 200
3187  030224     376     375     373  FLT0:   .BYTE   376, 375, 373, 367, 357, 337, 277, 177
3188  030234  000000                          .WORD   0                               ; TERMINATOR.
3189
3190  030236                            DLTSTS:
3191                                    ;*****************************************************************************
 (2)                                    .SBTTL      T116 -- TEST THAT BUS-RESET CLEARS THE RIGHT BITS
 (2)                                    ;*****************************************************************************
 (2)  030236  104420  000116            T116:   BEGINTEST, 116          ;; CHECK SWITCH REGISTER OPTIONS.
3192                                    ;
3193                                    ; SET ALL WRITABLE BITS AND RESET.  CHECK THAT ONLY
3194                                    ; RCV-IE, XMIT-IE, MAINT, AND XMIT-BRK GET CLEARED.
3195                                    ;
3196  030242  106427  000300                      MTPS    #PRI6           ; RAISE CPU
3197  030246  012777  000177  150574              MOV     #177,@XCSR      ; SET WRITABLE BITS IN XCSR...
3198  030254  012777  000377  150570              MOV     #377,@XBUF      ;...AND XBUF...
3199  030262  012777  000100  150554              MOV     #100,@RCSR      ;...AND RCSR.
3200  030270  000240  000240                      240,240                 ; IN CASE WE NEED A DELAY.
3201  030274  000005                              RESET
3202  030276  000240  000240                      240,240
3203  030302  017700  150536                      MOV     @RCSR,R0        ; GET RCSR...
3204  030306  017701  150534                      MOV     @RBUF,R1        ;...RBUF (DUMMY READ)...
3205  030312  017702  150532                      MOV     @XCSR,R2        ;...XCSR...
3206  030316  017703  150530                      MOV     @XBUF,R3        ;...AND XBUF.
3207  030322  016777  150512  150520              MOV     DFPBR,@XCSR     ; RESTORE DEFAULT XMIT...
3208  030330  005077  150510                      CLR     @RCSR           ;...AND RCVR STATUS.
3209  030334  032700  000100                      BIT     #100,R0         ; NOW, DID RCV-IE CLEAR ??
3210  030340  001402                              BEQ     1$
3211  030342  104423  030236                      ERROR,  T116            ;; *** BUS-RESET DIDN'T CLEAR RCV-IE BIT IN RCSR ***
3212  030346  032702  000105            1$:       BIT     #105,R2         ; DID XIE, MAINT AND XBRK CLEAR ??
3213  030352  001402                              BEQ     2$
3214  030354  104423  030236                      ERROR,  T116            ;; *** BUS-RESET DIDN'T CLEAR XIE, MAINT, AND/OR XBRK **
3215  030360  120227  000272            2$:       CMPB    R2,#272         ; ALL OTHERS SET (INCLUDING XMTRDY) ??
3216  030364  001402                              BEQ     3$
```

```
3217  030366  104423  030236            ERROR,  T116            ;; *** BUS-RESET CLEAREL WRONG BITS IN XCSR ***
3218  030372  120327  177777  3$:       CMPB    R3,#-1          ; ALL BITS STILL IN XBUF ??
3219  030376  001402                    BEQ     4$
3220  030400  104423  030236            ERROR,  T116            ;; *** BUS-RESET CLEARED BITS IN XBUF ***
3221  030404  106427  000000  4$:       MTPS    #PRIO           ; LOWER THE CPU.
3222
3223                                    ;*****************************************************************
 (2)                                    .SBTTL      T117 -- TEST THAT ALL UNDEFINED BITS ARE ZERO
 (2)                                    ;*****************************************************************
 (2)  030410  104420  000117  T117:     BEGINTEST, 117          ;; CHECK SWITCH REGISTER OPTIONS.
3224  030414  017700  150424            MOV     @RCSR,R0        ; GET REGISTERS AGAIN.
3225  030420  017701  150422            MOV     @RBUF,R1
3226  030424  017702  150420            MOV     @XCSR,R2
3227  030430  017703  150416            MOV     @XBUF,R3
3228  030434  032700  173477            BIT     #173477,R0      ; RCSR, UNDEFINED BITS SET ??
3229  030440  001402                    BEQ     1$
3230  030442  104423  030410            ERROR,  T117            ;; *** UNDEFINED BITS SET IN RCVR STATUS ***
3231  030446  032701  013400  1$:       BIT     #13400,R1       ; RBUF, UNDEFINED BITS SET ??
3232  030452  001402                    BEQ     2$
3233  030454  104423  030410            ERROR,  T117            ;; *** UNDEFINED BITS SET IN RCVR DATA BUFFER ***
3234  030460  032702  177400  2$:       BIT     #177400,R2      ; XCSR, UNDEFINED BITS SET ??
3235  030464  001402                    BEQ     3$
3236  030466  104423  030410            ERROR,  T117            ;; *** UNDEFINED BITS SET IN XMTR STATUS ***
3237  030472  032703  177400  3$:       BIT     #177400,R3      ; XBUF, UNDEFINED BITS SET ??
3238  030476  001402                    BEQ     4$
3239  030500  104423  030410            ERROR,  T117            ;; *** UNDEFINED BITS SET IN XMTR DATA BUFFER ***
3240  030504  105777  150340  4$:       TSTB    @XCSR           ; XMTR READY ??
3241  030510  100402                    BMI     5$
3242  030512  104423  030410            ERROR,  T117            ;; *** XMITTER NOT READY ***
3243  030516  030516            .PRINT  .                       ; *** DEBUG, NOP ***
3244  030516  000402            5$:     SKP2
3245  030520  000167  002726            JMP     DLDUN
3246                                    .SBTTL
```

```
                                .SBTTL  TRANSMITTERS
030524                          XMTRS:                          ; TRANSMITTER TEST SECTION.
                                ;*******************************************************************
                                .SBTTL      T120 -- FLOAT 1 AND 0 THRU THE XMIT DATA BUFFER
                                ;*******************************************************************
030524  104420  000120          T120:   BEGINTEST, 120          ;; CHECK SWITCH REGISTER OPTIONS.
030530  012700  030214                  MOV     #FLT1,R0        ; TABLE POINTER.
030534  112001                          MOVB    (R0)+,R1        ; GET 1ST CHARACTER.
030536  010177  150310          1$:     MOV     R1,@XBUF        ;LOAD DATA INTO "XMIT DATA BUF"...
030542  017702  150304                  MOV     @XBUF,R2        ;...AND READ IT BACK.
030546  120102                          CMPB    R1,R2
030550  001402                          BEQ     2$
030552  104423  030536                  ERROR,  1$              ;; *** FLOATING 1 OR 0 INCORRECT IN XBUF ***
030556  112001                  2$:     MOVB    (R0)+,R1        ; GET NEXT CHAR...
030560  001366                          BNE     1$              ;...AND LOOP 'TIL DONE.

                                ;*******************************************************************
                                .SBTTL      T121 -- TEST THAT "XMIT-BRK" CAN BE SET AND CLEARED
                                ;*******************************************************************
030562  104420  000121          T121:   BEGINTEST, 121          ;; CHECK SWITCH REGISTER OPTIONS.
030566  106427  000340                  MTPS    #PRI7           ; RAISE CPU.
030572  052777  000001  150250          BIS     #1,@XCSR        ;SET "XMIT-BRK" BIT...
030600  017700  150244                  MOV     @XCSR,R0        ;...AND SAVE IN R0.
030604  042777  000001  150236          BIC     #1,@XCSR        ; CLEAR IT...
030612  017701  150232                  MOV     @XCSR,R1        ;...AND SAVE IN R1.
030616  016777  150216  150224          MOV     DFPBR,@XCSR     ; RESTORE DEFAULT XMTR.
030624  106427  000000                  MTPS    #PRI0           ; AND PRIORITY.
030630  032700  000001                  BI      #1,R0
030634  001002                          BNE     1$              ;BR IF IT SET
030636  104423  030562                  ERROR,  T121            ;; *** XMIT-BRK BIT FAILED TO SET ***
030642  032701  000001          1$:     BIT     #1,R1
030646  001402                          BEQ     2$              ;BR IF IT CLEARED
030650  104423  030562                  ERROR,  T121            ;; *** XMIT-BRK BIT FAILED TO CLEAR ***
030654                          2$:

                                ;*******************************************************************
                                .SBTTL      T122 -- TEST THAT "XMIT-MAINT" CAN BE SET AND CLEARED
                                ;*******************************************************************
030654  104420  000122          T122:   BEGINTEST, 122          ;; CHECK SWITCH REGISTER OPTIONS.
030660  052777  000004  150162          BIS     #4,@XCSR        ;SET "XMIT-MAINT" BIT...
030666  017700  150156                  MOV     @XCSR,R0        ;...AND SAVE IN R0.
030672  042777  000004  150150          BIC     #4,@XCSR        ; CLEAR IT...
030700  017701  150144                  MOV     @XCSR,R1        ;...AND SAVE IN R1.
030704  016777  150130  150136          MOV     DFPBR,@XCSR
030712  032700  000004                  BIT     #4,R0
030716  001002                          BNE     1$              ;BR IF IT SET
030720  104423  030654                  ERROR,  T122            ;; *** XMIT-MAINT BIT FAILED TO SET ***
030724  032701  000004          1$:     BIT     #4,R1
030730  001402                          BEQ     2$              ;BR IF IT CLEARED
030732  104423  030654                  ERROR,  T122            ;; *** XMIT-MAINT BIT FAILED TO CLEAR ***
030736                          2$:

                                ;*******************************************************************
                                .SBTTL      T123 -- TEST THAT "XMIT-PBRE" CAN BE SET AND CLEARED
                                ;*******************************************************************
030736  104420  000123          T123:   BEGINTEST, 123          ;; CHECK SWITCH REGISTER OPTIONS.
```

```
3292  030742  052777  000002  150100          BIS     #2,@XCSR        ;SET ''XMIT-PBRE'' BIT...
3293  030750  017700  150074                  MOV     @XCSR,R0        ;...AND SAVE IN R0.
3294  030754  042777  000002  150066          BIC     #2,@XCSR        ; CLEAR IT...
3295  030762  017701  150062                  MOV     @XCSR,R1        ;...AND SAVE IN R1.
3296  030766  016777  150046  150054          MOV     DFPBR,@XCSR
3297  030774  032700  000002                  BIT     #2,R0
3298  031000  001002                          BNE     1$              ;BR IF IT SET
3299  031002  104423  030736                  ERROR,  T123            ;; *** XMIT-PBRE BIT FAILED TO SET ***
3300  031006  032701  000002          1$:     BIT     #2,R1
3301  031012  001402                          BEQ     2$              ;BR IF IT CLEARED
3302  031014  104423  030736                  ERROR,  T123            ;; *** XMIT-PBRE BIT FAILED TO CLEAR ***
3303  031020                          2$:
3304
3305                                  ;*******************************************************************
  (2)                               .SBTTL      T124 -- TEST THAT ''XMIT-PBR0'' CAN BE SET AND CLEARED
  (2)                               ;*******************************************************************
  (2)   031020  104420  000124      T124:   BEGINTEST, 124          ;; CHECK SWITCH REGISTER OPTIONS.
3306  031024  052777  000010  150016          BIS     #10,@XCSR       ;SET ''XMIT-PBR0'' BIT...
3307  031032  017700  150012                  MOV     @XCSR,R0        ;...AND SAVE IN R0.
3308  031036  042777  000010  150004          BIC     #10,@XCSR       ; CLEAR IT...
3309  031044  017701  150000                  MOV     @XCSR,R1        ;...AND SAVE IN R1.
3310  031050  016777  147764  147772          MOV     DFPBR,@XCSR
3311  031056  032700  000010                  BIT     #10,R0
3312  031062  001002                          BNE     1$              ;BR IF IT SET
3313  031064  104423  031020                  ERROR,  T124            ;; *** XMIT-PBR0 BIT FAILED TO SET ***
3314  031070  032701  000010          1$:     BIT     #10,R1
3315  031074  001402                          BEQ     2$              ;BR IF IT CLEARED
3316  031076  104423  031020                  ERROR,  T124            ;; *** XMIT-PBR0 BIT FAILED TO CLEAR ***
3317  031102                          2$:
3318
3319                                  ;*******************************************************************
  (2)                               .SBTTL      T125 -- TEST THAT ''XMIT-PBR1'' CAN BE SET AND CLEARED
  (2)                               ;*******************************************************************
  (2)   031102  104420  000125      T125:   BEGINTEST, 125          ;; CHECK SWITCH REGISTER OPTIONS.
3320  031106  052777  000020  147734          BIS     #20,@XCSR       ;SET ''XMIT-PBR1'' BIT...
3321  031114  017700  147730                  MOV     @XCSR,R0        ;...AND SAVE IN R0.
3322  031120  042777  000020  147722          BIC     #20,@XCSR       ; CLEAR IT...
3323  031126  017701  147716                  MOV     @XCSR,R1        ;...AND SAVE IN R1.
3324  031132  016777  147702  147710          MOV     DFPBR,@XCSR
3325  031140  032700  000020                  BIT     #20,R0
3326  031144  001002                          BNE     1$              ;BR IF IT SET
3327  031146  104423  031102                  ERROR,  T125            ;; *** XMIT-PBR1 BIT FAILED TO SET ***
3328  031152  032701  000020          1$:     BIT     #20,R1
3329  031156  001402                          BEQ     2$              ;BR IF IT CLEARED
3330  031160  104423  031102                  ERROR,  T125            ;; *** XMIT-PBR1 BIT FAILED TO CLEAR ***
3331  031164                          2$:
3332
3333                                  ;*******************************************************************
  (2)                               .SBTTL      T126 -- TEST THAT ''XMIT-PBR2'' CAN BE SET AND CLEARED
  (2)                               ;*******************************************************************
  (2)   031164  104420  000126      T126:   BEGINTEST, 126          ;; CHECK SWITCH REGISTER OPTIONS.
3334  031170  052777  000040  147652          BIS     #40,@XCSR       ;SET ''XMIT-PBR2'' BIT...
3335  031176  017700  147646                  MOV     @XCSR,R0        ;...AND SAVE IN R0.
3336  031202  042777  000040  147640          BIC     #40,@XCSR       ; CLEAR IT...
3337  031210  017701  147634                  MOV     @XCSR,R1        ;...AND SAVE IN R1.
3338  031214  016777  147620  147626          MOV     DFPBR,@XCSR
```

```
 3339  031222  032700  000040              BIT     #40,R0
 3340  031226  001002                      BNE     1$                ;BR IF IT SET
 3341  031230  104423  031164              ERROR,  T126              ;; *** XMIT-PBR2 BIT FAILED TO SET ***
 3342  031234  032701  000040        1$:   BIT     #40,R1
 3343  031240  001402                      BEQ     2$                ;BR IF IT CLEARED
 3344  031242  104423  031164              ERROR,  T126              ;; *** XMIT-PBR2 BIT FAILED TO CLEAR ***
 3345  031246                        2$:
 3346
 3347                                      ;*****************************************************************************
  (2)                                      .SBTTL      T127 -- TEST THAT "XMIT-IE" CAN BE SET AND CLEARED
  (2)                                      ;*****************************************************************************
  (2)  031246  104420  000127        T127: BEGINTEST, 127            ;; CHECK SWITCH REGISTER OPTIONS.
 3348  031252  106427  000300              MTPS    #PRI6             ;RAISE CPU PRIORITY
 3349  031256  052777  000100  147564      BIS     #100,@XCSR        ;SET "XMIT-IE" BIT...
 3350  031264  017700  147560              MOV     @XCSR,R0          ;...AND SAVE IN R0.
 3351  031270  042777  000100  147552      BIC     #100,@XCSR        ; CLEAR IT...
 3352  031276  017701  147546              MOV     @XCSR,R1          ;...AND SAVE IN R1.
 3353  031302  016777  147532  147540      MOV     DFPBR,@XCSR       ; RESTORE XMTR...
 3354  031310  106427  000000              MTPS    #PRI0             ;...AND CPU TOO.
 3355  031314  032700  000100              BIT     #100,R0
 3356  031320  001002                      BNE     1$                ;BR IF IT SET
 3357  031322  104423  031246              ERROR,  T127              ;; *** XMIT-IE BIT FAILED TO SET ***
 3358  031326  032701  000100        1$:   BIT     #100,R1
 3359  031332  001402                      BEQ     2$                ; BR IF IT CLEARED.
 3360  031334  104423  031246              ERROR,  T127              ;; *** XMIT-IE BIT FAILED TO CLEAR ***
 3361  031340                        2$:
 3362
 3363                                      ;*****************************************************************************
  (2)                                      .SBTTL      T130 -- TEST THAT "XMIT-RDY" CAN BE SET AND CLEARED
  (2)                                      ;*****************************************************************************
  (2)  031340  104420  000130        T130: BEGINTEST, 130            ;; CHECK SWITCH REGISTER OPTIONS.
 3364  031344  016777  147470  147476      MOV     DFPBR,@XCSR
 3365  031352  005001                      CLR     R1                ; DELAY COUNT.
 3366  031354  005002                      CLR     R2                ; DITTO
 3367  031356  105777  147466        1$:   TSTB    @XCSR             ;WAIT FOR "XMIT-RDY" TO SET
 3368  031362  100403                      BMI     2$
 3369  031364  077104                      SOB     R1,1$             ;KEEP TRYING.
 3370  031366  104423  031340              ERROR,  T130              ;; *** XMIT-RDY FAILED TO SET ***
 3371  031372  012777  000000  147452  2$: MOV     #0,@XBUF          ;TRANSMIT A NULL CHARACTER
 3372  031400  017700  147444              MOV     @XCSR,R0          ;READ XMIT STATUS REG
 3373  031404  032700  000200              BIT     #XMTRDY,R0        ; READY SHOULD BE GONE.
 3374  031410  001402                      BEQ     3$                ; BR IF SO.
 3375  031412  104423  031340              ERROR,  T130              ;; *** XMIT-RDY BIT FAILED TO CLEAR ***
 3376  031416  105777  147426        3$:   TSTB    @XCSR             ;WAIT FOR "XMIT-RDY" TO SET
 3377  031422  100403                      BMI     4$
 3378  031424  077204                      SOB     R2,3$             ; KEEP TRYING
 3379  031426  104423  031340              ERROR,  T130              ;; *** XMIT-RDY FAILED TO SET AFTER XMITTING CHARACTER *
 3380  031432                        4$:
 3381
 3382                                      ;*****************************************************************************
  (2)                                      .SBTTL      T131 -- TEST THAT WE CAN INTERRUPT ON "XMT-RDY"
  (2)                                      ;*****************************************************************************
  (2)  031432  104420  000131        T131: BEGINTEST, 131            ;; CHECK SWITCH REGISTER OPTIONS.
 3383  031436  016777  147376  147404      MOV     DFPBR,@XCSR
 3384  031444  016701  147406              MOV     XVEC,R1           ; GET VECTOR POINTER.
 3385  031450  106461  000002              MTPS    2(R1)             ; RAISE CPU TO XMTR LEVEL.
```

```
3386  031454  012711  031500                  MOV     #2$,(R1)        ; SET XMITTER VECTOR.
3387  031460  105777  147364          1$:     TSTB    @XCSR           ;WAIT FOR ''XMIT-RDY''
3388  031464  100375                          BPL     1$
3389  031466  052777  000100  147354          BIS     #100,@XCSR      ;SET ENABLE ''XMIT-IE'' BIT
3390  031474  000240                          240                     ; INTERRUPT SHOULD BE HELD OFF...
3391  031476  000406                          BR      3$              ;...BR IF SO.
3392  031500  022626                  2$:     CMP     (SP)+,(SP)+     ; IT WASN'T.
3393  031502  042777  000100  147340          BIC     #100,@XCSR      ; REMOVE ''XMIT-IE''
3394  031510  104423  031432                  ERROR,  T131            ;; *** XMT-RDY INTERRUPT AT WRONG LEVEL (TOO HIGH) ***
3395
3396  031514  012711  031542          3$:     MOV     #4$,(R1)        ;CHANGE THE VECTOR.
3397  031520  106427  000000                  MTPS    #PRIO           ;LOWER CPU PRIORITY
3398  031524  000240                          240                     ; INTERRUPT SHOULD HAVE COME IN.
3399  031526  042777  000100  147314          BIC     #100,@XCSR      ; IT DIDN'T, REMOVE ''XMIT-IE'' BIT
3400  031534  104423  031432                  ERROR,  T131            ;; *** XMT-RDY INTERRUPT NOT RECEIVED AT CPU LEVEL 0 ***
3401  031540  000415                          BR      6$
3402                                  ;
3403                                  ; NOW, ON INTERRUPT, WE'RE HERE AT CPU LEVEL 4 (5 IF SLU2).
3404                                  ; LOWER CPU AGAIN AND CHECK THAT WE DON'T GET ANOTHER INTERRUPT.
3405                                  ;
3406  031542  022626                  4$:     CMP     (SP)+,(SP)+
3407  031544  012711  031560                  MOV     #5$,(R1)        ; CHANGE THE VECTOR.
3408  031550  106427  000000                  MTPS    #PRIO           ;LOWER THE CPU PRIORITY AGAIN.
3409  031554  000240                          240                     ; SHOULDN'T GET ANOTHER...
3410  031556  000406                          BR      6$              ;...OK, EXIT.
3411  031560  022626                  5$:     CMP     (SP)+,(SP)+     ; 2ND INTERRUPT -- BAD NEWS.
3412  031562  042777  000100  147260          BIC     #100,@XCSR      ;REMOVE ''XMIT-IE''
3413  031570  104423  031432                  ERROR,  T131            ;; *** XMT INTERRUPT ACK FAILED TO REMOVE THE REQUEST **
3414
3415  031574  042777  000100  147246  6$:     BIC     #100,@XCSR      ; BE SURE THE ENABLE IS REMOVED.
3416  031602  012711  030202                  MOV     #BADTI,(R1)     ;RESET FALSE INTR VECTOR
3417
3418                                  ;******************************************************************************
 (2)                                  .SBTTL      T132 -- TEST THAT EACH XMIT BAUD RATE SETS A DIFFERENT SPEED
 (2)                                  ;******************************************************************************
 (2)  031606  104420  000132          T132:   BEGINTEST, 132          ;; CHECK SWITCH REGISTER OPTIONS.
3419                                  ;
3420                                  ; FIRST DETERMINE THE RELATIVE SPEED OF EACH BAUD RATE.
3421                                  ; TRANSMIT A CHARACTER AND COUNT CPU BUS TRANSACTION CYCLES UNTIL
3422                                  ; TRANSMISSION IS COMPLETE (XMT-RDY ASSERTS).
3423                                  ; SAVE THE OBSERVED CYCLE COUNT IN THE SPEED TABLE.
3424                                  ; IF NO FLAG BY THE TIME THE COUNT OVERFLOWS (65536. CYCLES), SAVE
3425                                  ; THE ZERO, WHICH IMPLIES XMT RDY NEVER SET AT THAT SPEED.
3426                                  ;
3427                                  ; NOTE:  ALL FOLLOWING RECEIVER TESTS WILL USE THE SPEED VALUES
3428                                  ; OBSERVED HERE AS LOOP CONTROL COUNTERS.
3429                                  ; NOTE:NOTE:  IF THERE'S A TTY ATTACHED, YOU'LL SEE SOME GARBAGE
3430                                  ; CHARACTERS ON THE SCREEN.
3431                                  ;
3432  031612  012700  000002                  MOV     #2,R0           ; START WITH SPEED 0 (300 BAUD).
3433  031616  012701  001016                  MOV     #SPD0,R1        ; TABLE POINTER.
3434  031622  005002                  1$:     CLR     R2              ; CLEAR THE ACCUMULATOR.
3435  031624  010077  147220                  MOV     R0,@XCSR        ; SET PBR BITS.
3436  031630  012777  000000  147214          MOV     #0,@XBUF        ; XMIT A CHAR.
3437  031636  105777  147206          2$:     TSTB    @XCSR           ; ON READY, CHAR HAS BEEN COPIED...
3438  031642  100375                          BPL     2$              ;...INTO THE SERIAL OUTPUT REGISTER.
```

```
3439  031644  012777  000000  147200          MOV     #0,@XBUF        ; XMIT AGAIN AND MEASURE THE TIME...
3440                                                                  ;...UNTIL READY RETURNS WHICH IS THE...
3441                                                                  ;...SERIAL OUTPUT TIME.
3442  031652  105777  147172          3$:     TSTB    @XCSR           ; [ 4 ]
3443  031656  100403                          BMI     4$              ; [ 1 ] WE'RE DONE ON XMT-RDY.
3444  031660  062702  000010                  ADD     #8.,R2          ; [ 2 ] COUNT 8 CYCLES/LOOP...
3445  031664  001372                          BNE     3$              ; [ 1 ] ...UNITL DONE (OR OVERFLOW).
3446  031666  010221                  4$:     MOV     R2,(R1)+        ; OK, R2 = TX TIME IN BUS CYCLES...
3447                                                                  ;...AT APPROX 3 USEC/COUN ...
3448                                                                  ;...SAVE IT IN THE SPEED TABLE.
3449  031670  020067  147144                  CMP     R0,DFPBR        ; CURRENT RATE = DEFAULT ???
3450  031674  001002                          BNE     5$              ; SKIP IF NOT.
3451  031676  010267  147140                  MOV     R2,DFSPD        ; YES, SAVE AS DEFAULT SPEED.
3452  031702  062700  000010          5$:     ADD     #10,R0          ; INCREMENT PBR BITS.
3453  031706  020027  000072                  CMP     R0,#72
3454  031712  101743                          BLOS    1$              ; LOOP FOR ALL 8 SETTINGS.
3455  031714  016777  147120  147126          MOV     DFPBR,@XCSR     ; DONE, RESTORE DEFAULT XMTTR...
3456                                                                  ;...AND FALL THRU.
3457                                  ;
3458                                  ; NOW THE RELATIVE SPEED AT EACH SETTING HAS BEEN DETERMINED.
3459                                  ; CHECK THAT THE DIFFERENTIAL OF EACH ONE TO THE NEXT LOWER
3460                                  ; IS 2:1 (MORE OR LESS).  THIS IS A GROSS SPEED CHECK SINCE
3461                                  ; WE CAN'T RELY ON A REAL CLOCK FOR TIMING.
3462                                  ;
3463                                  ; DIFFERENTIAL IS CALCULATED BY:
3464                                  ;       (LOWER/2)-HIGHER = 0 +/- 25.0% OF THE LOWER.
3465                                  ;
3466                                  ; ON ERROR, R0 DISPLAYS THE HIGHER OF THE TWO RATES UNDER TEST.
3467                                  ;
3468  031722  012700  000072                  MOV     #72,R0          ; WORK FROM THE HIGHEST DOWN.
3469  031726  012701  001034                  MOV     #SPD7,R1        ; TABLE POINTER.
3470  031732  011102                  10$:    MOV     (R1),R2         ; GET HIGHER (SMALLER) VALUE.
3471  031734  001002                          BNE     11$
3472  031736  104423  031606                  ERROR,  T132            ;* *** XMT-RDY NEVER SET AT PBR IN R0 ***
3473  031742  020127  001016          11$:    CMP     R1,#SPD0        ; IF PBR 0, THERE'S NOTHING LOWER...
3474  031746  001421                          BEQ     14$             ;...SO JUST QUIT.
3475
3476  031750  014103                          MOV     -(R1),R3        ; GET NEXT LOWER (LARGER) VALUE.
3477  031752  010304                          MOV     R3,R4           ; SAVE A COPY FOR THE TOLERANCE.
3478  031754  006203                          ASR     R3              ; NOW, 1/2 THE LARGER...
3479  031756  160203                          SUB     R2,R3           ;...MINUS THE SMALLER = DELTA.
3480  031760  100001                          BPL     12$
3481  031762  005403                          NEG     R3              ; MAKE ABSOLUTE IF NECESSARY.
3482
3483  031764  006204                  12$:    ASR     R4
3484  031766  006204                           ASR    R4              ; SET ALLOWABLE TOLERANCE...
3485  031770  000240                            NOP                   ;... = 25.0% OF THE LARGER.
3486  031772  000240                             NOP
3487  031774  020304                          CMP     R3,R4           ; DELTA <= 25.0% ???
3488  031776  003402                          BLE     13$             ; BR IF SO.
3489  032000  104423  031606                  ERROR,  T132            ;; *** SPEED DIFFERENTIAL ERROR ***
3490  032004  162700  000010          13$:    SUB     #10,R0          ; DEC RATE COUNT...
3491  032010  100350                          BPL     10$             ;...AND LOOP 'TIL DONE.
3492  032012                          14$:
3493                                  .SBTTL
```

```
                                        .SBTTL  RECEIVERS
032012                                  RCVRS:                          ; RECEIVER TEST SECTION.
                                        ;*******************************************************************
                                        .SBTTL      T133 -- TEST THAT 'RCV-IE" CAN BE SET AND CLEARED
                                        ;*******************************************************************
032012  104420  000133                  T133:   BEGINTEST, 133          ;; CHECK SWITCH REGISTER OPTIONS.
032016  106427  000300                          MTPS    #PRI6           ;RAISE CPU PRIORITY
032022  012777  000100  147014                  MOV     #100,@RCSR      ;SET 'RCV-IE" BIT...
032030  017700  147010                          MOV     @RCSR,R0        ;...AND SAVE IN R0.
032034  005077  147004                          CLR     @RCSR           ; CLEAR IT...
032040  017701  147000                          MOV     @RCSR,R1        ;...AND SAVE IN R1.
032044  032700  000100                          BIT     #100,R0
032050  001002                                  BNE     1$              ; BR IF IT SET.
032052  104423  032012                          ERROR,  T133            ;; *** RCV-IE BIT FAILED TO SET ***
032056  032701  000100                  1$:     BIT     #100,R1
032062  001402                                  BEQ     2$              ; BR IF IT CLEARED.
032064  104423  032012                          ERROR,  T133            ;; *** RCV-IE BIT FAILED TO CLEAR ***
032070  106427  000000                  2$:     MTPS    #PRI0

                                        ;*******************************************************************
                                        .SBTTL      T134 -- TEST THAT 'RCV-ACT" AND 'RCV-DONE" CAN SET AND CLEAR
                                        ;*******************************************************************
032074  104420  000134                  T134:   BEGINTEST, 134          ;; CHECK SWITCH REGISTER OPTIONS.
                                        ;
                                        ;THE TEST ENSURES THAT:
                                        ;       1. THE INTERNAL 'MAINT" PATH IS FUNCTIONAL
                                        ;       2. RCV-ACT CLEARS BEFORE RCV-DONE SETS.
                                        ;       3. RCV-ACT CAN SET AND CLEAR
                                        ;       4. READING RCV DATA REGISTER CLEARS RCV DONE
                                        ;
032100  016777  146734  146742                   MOV     DFPBR,@XCSR
032106  052777  000004  146734                   BIS     #4,@XCSR        ; SET MAINT BIT.
032114  105777  146730                  1$:     TSTB    @XCSR           ;INSURE WE'RE 'XMIT-RDY"
032120  100375                                  BPL     1$
032122  016700  146714                          MOV     DFSPD,R0
032126  006300                                  ASL     R0              ; SET A 2 CHAR (20 BITS) DELAY COUNT...
032130  010001                                  MOV     R0,R1
032132  010002                                  MOV     R0,R2           ;...AND SAVE 3 COPIES FOR LATER.
032134  010003                                  MOV     R0,R3
032136  077001                                  SOB     R0,.            ; WAIT 20 BIT TIMES...
032140  017700  146702                          MOV     @RBUF,R0        ;...AND DUMMY READ.
032144  017700  146674                          MOV     @RCSR,R0        ; GET RCVR STATUS.
032150  032700  004200                          BIT     #RCVACT!RCVDUN,R0 ; BOTH ACTIVE AND DONE SHOULD BE CLEAR
032154  001402                                  BEQ     2$
032156  104423  032074                          ERROR,  T134            ;; *** RCV-ACT AND/OR RCV-DUN ARE SET IN ERROR ***

032162  012777  000000  146662  2$:     MOV     #0,@XBUF        ;NOW, XMIT A NULL CHARACTER
032170  000240                          240
032172  017700  146646                  3$:     MOV     @RCSR,R0        ;GET RCVR STATUS
032176  032700  004000                          BIT     #RCVACT,R0
032202  001003                                  BNE     4$              ;BR IF AND WHEN ACTIVE SETS.
032204  077106                                  SOB     R1,3$
032206  104423  032074                          ERROR,  T134            ;; *** RCV-ACT FAILED TO SET ***
032212  032700  000200                  4$:     BIT     #RCVDUN,R0      ;TEST THAT 'RCV-DUN" IS NOT SET YET.
032216  001402                                  BEQ     5$
032220  104423  032074                          ERROR,  T134            ;; *** RCV-DUN SET WHEN RCV-ACT WAS SET ***
```

```
3545  032224  017700  146614          5$:     MOV     @RCSR,R0        ;READ RCVR STATUS AGAIN.
3546  032230  032700  004000                  BIT     #RCVACT,R0
3547  032234  001403                          BEQ     6$              ; BR IF AND WHEN ACTIVE CLEARS.
3548  032236  077206                          SOB     R2,5$
3549  032240  104423  032074                  ERROR,  T134            ;; *** RCV-ACT SET BUT FAILED TO CLEAR ***
3550  032244  017700  146574          6$:     MOV     @RCSR,R0        ;READ RCVR STATUS
3551  032250  032700  000200                  BIT     #RCVDUN,R0
3552  032254  001003                          BNE     7$              ; BR IF AND WHEN DONE SETS.
3553  032256  077306                          SOB     R3,6$
3554  032260  104423  032074                  ERROR,  T134            ;; *** RCV-DUN FAILED TO SET WHEN RCV-ACT CLEARED ***
3555  032264  017700  146556          7$:     MOV     @RBUF,R0        ; OK, READ SHOULD LOWER DONE FLAG.
3556  032270  017700  146550                  MOV     @RCSR,R0        ; GET STATUS.
3557  032274  032700  000200                  BIT     #RCVDUN,R0
3558  032300  001402                          BEQ     8$              ; BR IF DONE IS CLEAR.
3559  032302  104423  032074                  ERROR,  T134            ;; *** RCV-DUN FAILED TO CLEAR AFTER READING DATA ***
3560  032306                          8$:
3561
3562                                  ;*************************************************************************
  (2)                             .SBTTL      T135 -- TEST THE 'RCV-BRK', 'RCV-ERR', AND 'FR-ERR' BITS
  (2)                             ;*************************************************************************
  (2)   032306  104420  000135          T135.   BEGINTEST, 135          ;; CHECK SWITCH REGISTER OPTIONS.
3563                                    ;
3564                                    ; GENERATE A FRAMING ERROR BY SETTING THE RCV-BRK BIT.
3565                                    ; ON SLU1 -- EXPECT THAT BRK INVOKES A MASKABLE LEVEL 7 HALT REQUEST.
3566                                    ;
3567  032312  013746  000140                  MOV     @#BHALT,-(SP)   ; SAVE BREAK VECTOR.
3568  032316  106427  000340          1$:     MTPS    #PR17           ; RAISE CPU.
3569  032322  016777  146512  146520          MOV     DFPBR,@XCSR
3570  032330  052777  000004  146512          BIS     #4,@XCSR        ; SET MAINT BIT.
3571  032336  017700  146504                  MOV     @RBUF,R0        ; DUMMY READ.
3572  032342  012767  032476  145570          MOV     #6$,BHALT       ; SET THE HALT VECTOR (FOR SLU1).
3573  032350  012737  000340  000142          MOV     #PR17,@#BHALT+2
3574  032356  016701  146460                  MOV     DFSPD,R1
3575  032362  006301                          ASL     R1              ; SET A 2 CHAR (20 BITS) DELAY.
3576  032364  052777  000001  146456          BIS     #XMTBRK,@XCSR   ; SET "XBRK"...
3577  032372  012777  000025  146452          MOV     #25,@XBUF       ;...START XMITTER...
3578  032400  077101                          SOB     R1,.            ;...AND DELAY.  IF SLU1, INTERRUPT...
3579                                                                  ;...SHOULD BE HELD OFF (CPU IS AT 7).
3580  032402  105777  146436                  TSTB    @RCSR           ; DONE SHOLD BE THERE BY NOW.
3581  032406  100402                          BMI     2$              ; BR IF SO.
3582  032410  104423  032316                  ERROR,  1$              ;; *** NO RCV-DUN AFTER XMIT WITH XMT-BRK SET ***
3583  032414  026727  146424  176540  2$:     CMP     RCSR,#SLU2      ; IF SLU2...
3584  032422  001430                          BEQ     7$              ;...JUST VERIFY STATUS/DATA.
3585                                    ;
3586                                    ; SLU1, VERIFY THAT "BHALT" INTERRUPT DOES THE RIGHT THINGS.
3587                                    ;
3588  032424  012767  032446  145506  3$:     MOV     #4$,BHALT       ; SO FAR, SO GOOD, CHANGE THE VECTOR...
3589  032432  106427  000300                  MTPS    #PR16           ;...AND LOWER CPU TO 6...
3590  032436  000240                          240                     ;...INTERRUPT SHOULD COME IN.
3591  032440  104423  032316                  ERROR,  1$              ;; *** HALT INTERRUPT NOT RECEIVED AT CPU LEVEL 6 ***
3592  032444  000417                          BR      7$
3593  032446  022626                  4$:     CMP     (SP)+,(SP)+     ; OK, INT RECEIVED, FIX STACK.
3594  032450  012767  032466  145462          MOV     #5$,BHALT       ; NOW, CHANGE THE VECTOR...
3595  032456  106427  000300                  MTPS    #PR16           ;...LOWER CPU AGAIN...
3596  032462  000240                          240                     ;...AND SEE THAT WE DON'T GET ANOTHER.
3597  032464  000407                          BR      7$              ; TERRIFIC -- PROCEED.
```

```
3598  032466  022626                  5$:     CMP     (SP)+,(SP)+     ; 2ND HALT TRAP -- TOUGH LUCK !!
3599  032470  104423  032316                  ERROR,  1$              ;; *** DOUBLE HALT TRAP RECEIVED ***
3600  032474  000403                          BR      7$
3601  032476  022626                  6$:     CMP     (SP)+,(SP)+     ; WE'RE HERE IF HALT WASN'T MASKED !!
3602  032500  104423  032316                  ERROR,  1$              ;; *** HALT TRAP INTERRUPT NOT MASKED BY CPU LEVEL 7 ***
3603                                  ;
3604                                  ; BOTH LINES VERIFY CORRECT STATUS AND DATA BITS.
3605                                  ;
3606  032504  017700  146336          7$:     MOV     @RBUF,R0        ; READ (IF SLU1, LOWERS BRK REQ)..
3607  032510  012637  000140                  MOV     (SP)+,@#BHALT   ;...AND RESTORE BREAK VECTOR.
3608  032514  032700  004000                  BIT     #RCVBRK,R0      ; RCV-BRK SHOULD BE SET.
3609  032520  001002                          BNE     8$
3610  032522  104423  032316                  ERROR,  1$              ;; *** RCV-BRK BIT DIDN'T SET ***
3611  032526  032700  100000          8$:     BIT     #RCVERR,R0      ; SO SHOULD RCV-ERR...
3612  032532  001403                          BEQ     9$
3613  032534  032700  020000                  BIT     #FRERR,R0       ;...AND FR-ERR.
3614  032540  001002                          BNE     10$
3615  032542                          9$:
 (1)  032542  104423  032316                  ERROR,  1$              ;; *** RCV-ERR AND/OR FR-ERR BITS DIDN'T SET ***
3616  032546  105700                  10$:    TSTB    R0              ; RCVD DATA SHOULD BE ZERO.
3617  032550  001402                          BEQ     11$
3618  032552  104423  032316                  ERROR,  1$              ;; *** RECEIVED DATA NON-ZERO ON "BRK" ***
3619                                  ;
3620                                  ; AND FINALLY, WE SHOULD BE ABLE TO RESTORE A NORMAL RECEIVER.
3621                                  ;
3622  032556  042777  000001  146264  11$:     BIC     #XMTBRK,@XCSR   ; NOW, CLEAR THE BREAK BIT...
3623  032564  016700  146252                  MOV     DFSPD,R0
3624  032570  077001                          SOB     R0,.            ;...DELAY...
3625  032572  017700  146250                  MOV     @RBUF,R0        ;...DUMMY READ.
3626  032576  012777  000025  146246          MOV     #25,@XBUF       ; XMIT A CHAR...
3627  032604  105777  146234          12$:    TSTB    @RCSR
3628  032610  100375                          BPL     12$             ;...WAIT FOR RCVR DONE...
3629  032612  017700  146230                  MOV     @RBUF,R0        ;...AND READ.
3630  032616  032700  124000                  BIT     #RCVERR!FRERR!RCVBRK,R0 ; ERRORS SHOULD BE GONE.
3631  032622  001402                          BEQ     13$             ; BR IF SO.
3632  032624  104423  032556                  ERROR,  11$             ;; *** RCV-ERR, FR-ERR, AND/OR BRK BITS DIDN'T CLEAR ***
3633  032630  120027  000025          13$:    CMPB    R0,#25
3634  032634  001402                          BEQ     14$             ; AND CORRECT DATA RECEIVED.
3635  032636  104423  032520                  ERROR,  11$             ;; *** DATA INCORRECT AFTER "BRK" SEQUENCE ***
3636  032642  106427  000000          14$:    MTPS    #PR10           ; DONE, LOWER THE CPU.
3637
3638                                  ;******************************************************************************
 (2)                                .SBTTL      T136 -- TEST THAT "RCV-ERR" AND "OR-ERR" BITS SET AND CLEAR
 (2)                                ;******************************************************************************
 (2)  032646  104420  000136          T136:   BEGINTEST, 136          ;; CHECK SWITCH REGISTER OPTIONS.
3639  032652  016777  146162  146170          MOV     DFPBR,@XCSR
3640  032660  052777  000004  146162          BIS     #4,@XCSR        ; SET MAINT BIT.
3641  032666  017700  146154                  MOV     @RBUF,R0        ; DUMMY READ.
3642  032672  012701  000004                  MOV     #4,R1
3643  032676  012777  000000  146146  1$:     MOV     #0,@XBUF        ;XMIT A CHARACTER
3644  032704  105777  146140          2$:     TSTB    @XCSR           ;WAIT FOR XMIT-RDY
3645  032710  100375                          BPL     .-4
3646  032712  077107                          SOB     R1,1$           ; LOOP 4 TIMES...
3647                                                                  ;...SHOULD CAUSE AN OVER-RUN ERROR.
3648  032714  016700  146122                  MOV     DFSPD,R0
3649  032720  077001                          SOB     R0,.            ; WAIT 1 ADDITIONAL CHAR TIME...
```

```
3650  032722  017700  146120                  MOV     @RBUF,R0        ;...AND READ RCV DATA BUFFER.
3651  032726  100003                          BPL     3$              ; BR IF NO "RCV-ERR".
3652  032730  052700  0[illegible]00          BIT     #ORERR,R0       ; "OR-ERR" SHOULD BE SET AS WELL.
3653  032734  001002                          BNE     4$              ; BOTH SET -- TERRIFIC.
3654  032736                          3$:
 (2)  032736  104423  [illegible]             ERROR,  T136            ;; *** RCV-ERR AND/OR OR-ERR NOT SET ON FORCED OVER-RUN
3655
3656  032742  012777  000000  146102  4$:     MOV     #0,@XBUF        ; NOW XMIT AGAIN...
3657  032750  105777  146070          5$:     TSTB    @RCSR           ;...WAIT FOR RCVR DUN...
3658  032754  100375                          BPL     5$
3659  032756  017700  146066                  MOV     @RBUF,R0        ;...AND READ, ERROR SHOULD BE GONE.
3660  032762  032700  140000                  BIT     #RCVERR!ORERR,R0
3661  032766  001402                          BEQ     6$              ; BR IF SO.
3662  032770  104423  032646                  ERROR,  T136            ;; *** RCV-ERR AND/OR OR-ERR BITS FAILED TO CLEAR ***
3663  032774                          6$:
3664
3665                                  ;*******************************************************************
 (2)                                .SBTTL      T137 -- TEST THAT WE CAN INTERRUPT ON "RCV-DUN"
 (2)                                ;*******************************************************************
 (2)  032774  104420  000137          T137:   BEGINTEST, 137          ;; CHECK SWITCH REGISTER OPTIONS.
3666  033000  016777  146034  146042          MOV     DFPBR,@XCSR
3667  033006  052777  000004  146034          BIS     #4,@XCSR        ; SET MAINT BIT.
3668  033014  016701  146034                  MOV     RVEC,R1         ; GET RCVR VECTOR POINTER.
3669  033020  106461  000002                  MTPS    2(R1)           ; RAISE CPU TO RCVR LEVEL.
3670  033024  012711  033056                  MOV     #2$,(R1)        ; SET RCVR VECTOR.
3671  033030  012777  000000  146014          MOV     #0,@XBUF        ;TRANSMIT A CHARACTER
3672  033036  105777  146002          1$:     TSTB    @RCSR           ;WAIT FOR "RCV-DUN"
3673  033042  100375                          BPL     1$
3674  033044  052777  000100  145772          BIS     #100,@RCSR      ; SET ENABLE "RCV-IE" BIT
3675  033052  000240                          240                     ; INTERRUPT SHOULD BE HELD OFF...
3676  033054  000406                          BR      3$              ;...BR IF SO.
3677  033056  022626                  2$:     CMP     (SP)+,(SP)+     ; BUT IT WASN'T.
3678  033060  042777  000100  145756          BIC     #100,@RCSR      ;REMOVE "RCV-IE"
3679  033066  104423  032774                  ERROR,  T137            ;; *** RCV-DUN INTERRUPT AT WRONG LEVEL (TOO HIGH) ***
3680
3681  033072  012711  033120          3$:     MOV     #4$,(R1)        ; CHANGE VECTOR.
3682  033076  106427  000000                  MTPS    #PRI0           ;LOWER CPU PRIORITY
3683  033102  000240                          240                     ; INTERRUPT SHOULD COME IN.
3684  033104  042777  000100  145732          BIC     #100,@RCSR      ; BUT IT DIDN'T, REMOVE "RCV-IE" BIT
3685  033112  104423  032774                  ERROR,  T137            ;; *** RCV-DUN INTERRUPT NOT RECEIVED AT ***
3686  033116  000415                          BR      6$
3687                                  ;
3688                                  ; NOW, ON INTERRUPT, WE'RE HERE AT LEVEL 4 (5 IF SLU2).
3689                                  ; LOWER THE CPU AND VERIFY THAT WE DON'T GET ANOTHER ONE.
3690                                  ;
3691  033120  022626                  4$:     CMP     (SP)+,(SP)+
3692  033122  012711  033136                  MOV     #5$,(R1)        ; CHANGE THE VECTOR.
3693  033126  106427  000000                  MTPS    #PRI0           ;LOWER THE CPU PRIORITY AGAIN.
3694  033132  000240                          240                     ; SHOULDN'T GET ANOTHER...
3695  033134  000406                          BR      6$              ;...OK, EXIT.
3696  033136  022626                  5$:     CMP     (SP)+,(SP)+     ; 2ND INTERRUPT -- TOO BAD !!
3697  033140  042777  000100  145676          BIC     #100,@RCSR      ;REMOVE "RCV-IE"
3698  033146  104423  032774                  ERROR,  T137            ;; *** INTERRUPT ACK FAILED TO REMOVE THE REQUEST ***
3699
3700  033152  042777  000100  145664  6$:     BIC     #100,@RCSR      ; INSURE THE ENABLE IS OFF.
3701  033160  017700  145662                  MOV     @RBUF,R0        ; READ
```

```
3702  033164  012711  030172                  MOV     #BADRI,(R1)     ;RESET FALSE INTR VECTOR
3703
3704                                  ;*****************************************************************
 (2)                                  .SBTTL      T140 -- FLOAT 1 AND 0 THRU THE "MAINT" DATA LOOP (INTERNAL)
 (2)                                  ;*****************************************************************
 (2)  033170  104420  000140          T140:   BEGINTEST, 140          ;; CHECK SWITCH REGISTER OPTIONS.
3705                                  ;
3706                                  ; NOTE:  IF THERE'S A TERMINAL ATTACHED, YOU'LL SEE GARBAGE
3707                                  ; CHARACTERS ON THE SCREEN -- DON'T WORRY ABOUT IT !!
3708                                  ;
3709  033174  016777  145640  145646          MOV     DFPBR,@XCSR
3710  033202  052777  000004  145640          BIS     #4,@XCSR        ; SET MAINT BIT.
3711  033210  012700  030212                  MOV     #SYNC,R0        ; DATA TABLE POINTER.
3712  033214  112077  145632          1$:     MOVB    (R0)+,@XBUF     ; XMIT THE SYNC PAIR <CR>...
3713  033220  105777  145624          2$:     TSTB    @XCSR
3714  033224  100375                          BPL     2$
3715  033226  112077  145620                  MOVB    (R0)+,@XBUF     ;...AND <LF>.
3716  033232  105777  145606          3$:     TSTB    @RCSR           ; WAIT FOR RCVR DONE...
3717  033236  100375                          BPL     3$
3718  033240  127727  145602  000012          CMPB    @RBUF,#12       ; AND PROCEED ON THE RCVD <LF>.
3719  033246  001371                          BNE     3$
3720
3721  033250  112001                  4$:     MOVB    (R0)+,R1        ; NOW FLOAT DATA THRU THE LOOP.
3722  033252  010177  145574          5$:     MOV     R1,@XBUF        ; XMIT A BYTE...
3723  033256  105777  145562          6$:     TSTB    @RCSR
3724  033262  100375                          BPL     6$
3725  033264  017702  145556                  MOV     @RBUF,R2        ;...AND READ IT BACK.
3726  033270  120102                          CMPB    R1,R2
3727  033272  001402                          BEQ     7$
3728  033274  104423  033252                  ERROR,  5$              ;; *** RCVD DATA INCORRECT, MAINT DATA PATH FAILURE ***
3729  033300  105701                  7$:     TSTB    R1              ; TERMINATOR DONE ??
3730  033302  001362                          BNE     4$              ; NOT YET, LOOP.
3731
3732                                  ;*****************************************************************
 (2)                                  .SBTTL      T141 -- FLOAT 1 AND 0 THRU THE LOOP-BACK (EXTERNAL, SLU2 ONLY)
 (2)                                  ;*****************************************************************
 (2)  033304  104420  000141          T141:   BEGINTEST, 141          ;; CHECK SWITCH REGISTER OPTIONS.
3733  033310  026727  145550  177560          CMP     RCSR,#SLU1      ; CONSOLE LINE ??
3734  033316  001455                          BEQ     DLDUN           ; DON'T IF SO.
3735
3736  033320  016777  145514  145522          MOV     DFPBR,@XCSR     ; DEFAULT XMIT STATUS (NO MAINT).
3737  033326  017700  145514                  MOV     @RBUF,R0        ; DUMMY READ.
3738  033332  016700  145504                  MOV     DFSPD,R0
3739  033336  006300                          ASL     R0
3740  033340  006300                          ASL     R0              ; 4 CHAR LOOP TIMER.
3741  033342  010001                          MOV     R0,R1           ; SAVE A COPY.
3742  033344  112777  000033  145500          MOVB    #33,@XBUF       ; XMIT AN "ESC".
3743  033352  105777  145466          1$:     TSTB    @RCSR           ; IF RCVR-DUN SETS NOW, WE CAN ASSUME...
3744  033356  100413                          BMI     3$              ;...THAT THE LOOP-BACK IS INSTALLED.
3745  033360  077001                          SOB     R0,1$
3746  033362  112777  000132  145462          MOVB    #'Z,@XBUF       ; IF NOT, XMIT "Z".
3747  033370  105777  145450          2$:     TSTB    @RCSR           ; IF IT SETS NOW, WE HAVE A TTY...
3748  033374  100416                          BMI     DLDUN           ;...RESPONDING TO THE "ESC Z" SEQUENCE.
3749  033376  077104                          SOB     R1,2$           ; IF NEITHER, THE LOOP MUST BE OPEN.
3750  033400  104423  033304                  ERROR,  T141            ;; *** NO RCV-DUN -- IS LOOP CONNECTOR INSTALLED ??? ***
3751  033404  000422                          BR      DLDUN           ; QUIT (WHILE WE'RE AHEAD) !!
```

```
3752
3753  033406  017700  145334          3$:     MOV     @RBUF,R0        ; OK, DUMMY READ...
3754  033412  012700  030214                  MOV     #FLT1,R0        ;...SET TABLE POINTER...
3755  033416  112001                  4$:     MOVB    (R0)+,R1        ;...AND FLOAT DATA THRU THE LOOP.
3756  033420  010177  145426          5$:     MOV     R1,@XBUF        ; XMIT A BYTE...
3757  033424  105777  145414          6$:     TSTB    @RCSR
3758  033430  100375                          BPL     6$
3759  033432  017702  145410                  MOV     @RBUF,R2        ;...AND READ IT BACK.
3760  033436  120102                          CMPB    R1,R2
3761  033440  001402                          BEQ     7$
3762  033442  104423  033420                  ERROR,  5$              ;; *** RCVD DATA INCORRECT, EXTERNAL DATA PATH FAILURE *
3763  033446  105701                  7$:     TSTB    R1              ; TERMINATOR DONE ??
3764  033450  001362                          BNE     4$              ; NOT YET, LOOP.
3765
3766  033452  016777  145362  145370  DLDUN:  MOV     DFPBR,@XCSR     ; INSURE A DEFAULT XMITTER...
3767  033460  017700  145362                  MOV     @RBUF,R0        ;...AND RECEIVER.
3768  033464  012767  000006  144312          MOV     #ERRVEC+2,ERRVEC
3769  033472  004767  001232                  CALL    ENDSEG          ; CHECK FOR LOOP-IN-SEGMENT...
3770                                                                  ;...RETURN AND FALL THRU IF NOT.
3771  033476  026727  145342  176540          CMP     RCSR,#SLU2      ; LINE 2 ??
3772  033504  001402                          BEQ     S7              ; BR IF SO...
3773  033506  000167  174302                  JMP     S6              ;...ELSE, GO 'ROUND ONCE.
3774                                  .SBTTL
```

```
                                        .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.
                                        ;
033512  012767  036335  002460  S7:     MOV     #SEG7,SEGN      ; SEGMENT 7 TEXT.

                                        ;*******************************************************************************
                                        .SBTTL      T142 -- TEST PIO REGISTER ADDRESSES
                                        ;*******************************************************************************
033520  104420  000142          T142:   BEGINTEST, 142          ;; CHECK SWITCH REGISTER OPTIONS.
033524  000240  000240                  240 240                 ; *** TRY A RESET HERE ***
033530  012700  176200                  MOV     #PIOA,R0        ; A IS 1ST OF 4 REGISTERS.
033534  012701  000130                  MOV     #PIOBV,R1       ; B IS 1ST OF 2 VECTOR PAIRS.
033540  012702  000240                  MOV     #PIOP,R2        ; PRIORITY (5) FOR BOTH.
033544  012767  033634  144232          MOV     #BADPIO,ERRVEC  ; SET TIME-OUT VECTOR.
033552  012706  003776                  MOV     #STACK,SP
033556  010067  145276                  MOV     R0,PPA          ; SET REGISTER POINTERS.
033562  005720                          TST     (R0)+           ; IF ANY TRAP -- TROUBLE !@!
033564  010067  145272                  MOV     R0,PPB
033570  005720                          TST     (R0)+
033572  010067  145266                  MOV     R0,PPC
033576  005720                          TST     (R0)+
033600  010067  145262                  MOV     R0,PPCW
033604  005710                          TST     (R0)
033606  010167  145260                  MOV     R1,PPBV         ; SAVE B VECTOR POINTER.
033612  012721  033644                  MOV     #BADBI,(R1)+    ; SET TO CATCH UNEXPECTED INTERRUPTS.
033616  010221                          MOV     R2,(R1)+
033620  010167  145244                  MOV     R1,PPAV         ; DITTO FOR PORT A VECTOR.
033624  012721  033654                  MOV     #BADAI,(R1)+
033630  010221                          MOV     R2,(R1)+
033632  000414                          BR      PIOTST
                                        ;
                                        ; COME HERE IF TIME-OUT OR UNEXPECTED PIO INTERRUPTS.
                                        ;
033634                          BADPIO:
033634  104423  033520                  ERROR,  T142            ;;  ** BUS TIME-OUT ON PIO ADDRESS ***
033640  000167  001050                  JMP     PIODUN
033644                          BADBI:
033644  104423  033664                  ERROR,  PIOTST          ;; *** UNEXPECTED PORT B INTERRUPT ***
033650  000167  001040                  JMP     PIODUN
033654                          BADAI:
033654  104423  033664                  ERROR,  PIOTST          ;; *** UNEXPECTED PORT A INTERRUPT ***
033660  000167  001030                  JMP     PIODUN
```

```
 3826  033664                          PIOTST:
 3827                                  ;*************************************************************************
  (2)                                  .SBTTL      T143 -- TEST THAT RESET INVOKES MODE 0 AND CLEARS THE CHIP
  (2)                                  ;*************************************************************************
  (2)  033664  104420  000143          T143:   BEGINTEST, 143          ;; CHECK SWITCH REGISTER OPTIONS.
 3828                                  ;
 3829                                  ; VERIFY THAT PORT A (AND IT'S BUFFER) IS IN THE "RESET" STATE.
 3830                                  ; PORTS B AND C WILL BE INDETERMINATE DUE TO THE LOOP-BACK INTERCONNECTS
 3831                                  ; AND THE CONTROL WORD PORT IS "WO", SO DON'T BOTHER WITH THOSE.
 3832                                  ;
 3833  033670  000240  000240                 240,240
 3834  033674  000005                         RESET                   ; LED (ON C<7>) SHOULD BE OFF.
 3835  033676  005000                         CLR     R0              ; DELAY, CHIP SEEMS TO TAKE QUITE...
 3836  033700  077001                         SOB     R0,.            ;...A WHILE (200USEC) TO GET "RESET".
 3837  033702  117700  145152                 MOVB    @PPA,R0         ; READ PORT A (INPUT).
 3838  033706  120027  177777                 CMPB    R0,#-1          ; LO BYTE ONLY -- SHOULD BE HIGH.
 3839  033712  001402                         BEQ     1$
 3840  033714  104423  033664                 ERROR,  T143            ;; *** PORT A RESET STATE INCORRECT ***
 3841  033720                          1$:
 3842
 3843                                  ;*************************************************************************
  (2)                                  .SBTTL      T144 -- TEST MODE 1 I/O THRU THE LOOP CONNECTOR (FLAG MODE)
  (2)                                  ;*************************************************************************
  (2)  033720  104420  000144          T144:   BEGINTEST, 144          ;; CHECK SWITCH REGISTER OPTIONS.
 3844  033724  112777  000264  145134         MOVB    #PPM1,@PPCW     ; SET MODE 1, A IN, B OUT, C<7:6> OUT.
 3845  033732  112777  000016  145126         MOVB    #<7_1.>!0,@PPCW ; CLEAR C(7) TO TURN ON THE TINY LED.
 3846  033740  112777  000014  145120         MOVB    #<6_1.>!0,@PPCW ; CLEAR C(6) JUST FOR THE HELL OF IT.
 3847  033746  012700  000012                 MOV     #10.,R0
 3848  033752  077001                         SOB     R0,.            ; DELAY ABOUT 50 USEC...
 3849  033754  117703  145100                 MOVB    @PPA,R3         ;...AND DUMMY READ TO DROP "INRDY".
 3850  033760  117703  145100                 MOVB    @PPC,R3         ; NOW, GET STATUS.
 3851  033764  120327  000002                 CMPB    R3,#OUTRDY      ; SHOULD HAVE THIS STATUS SET.
 3852  033770  001402                         BEQ     1$
 3853  033772  104423  033720                 ERROR,  T144            ;; *** PORT C MODE 1 STATUS INCORRECT ***
 3854
 3855  033776  005001                  1$:    CLR     R1              ; SEND A BINARY COUNT FROM R1...
 3856  034000  012702  177777                 MOV     #-1,R2          ;...TO R2 THRU THE 8255.
 3857  034004  005000                  2$:    CLR     R0              ; LOOP DELAY.
 3858  034006  110177  145050                 MOVB    R1,@PPB         ; XMIT DATA OUTPUT (THRU B).
 3859  034012  117703  145046          3$:    MOVB    @PPC,R3         ; GET STATUS.
 3860  034016  032703  000002                 BIT     #OUTRDY,R3      ; OUTRDY SHOULD BE GONE.
 3861  034022  001402                         BEQ     4$
 3862  034024  104423  034004                 ERROR,  2$              ;; *** OUTRDY STILL SET AFTER OUTPUT XFER ***
 3863  034030  117703  145030          4$:    MOVB    @PPC,R3         ; GET STATUS AGAIN...
 3864  034034  120327  000042                 CMPB    R3,#INRDY!OUTRDY ;...AND WAIT FOR THIS.
 3865  034040  001403                         BEQ     5$              ; WONDERFUL, MOVE ON.
 3866  034042  077006                         SOB     R0,4$
 3867  034044  104423  034004                 ERROR,  2$              ;; *** STATUS INCORRECT AFTER OUTPUT XFER ***
 3868  034050  117702  145004          5$:    MOVB    @PPA,R2         ; DATA INPUT (THRU A).
 3869  034054  117703  145004                 MOVB    @PPC,R3         ; GET STATUS
 3870  034060  120327  000002                 CMPB    R3,#OUTRDY      ; EXPECT THIS STATUS.
 3871  034064  001402                         BEQ     6$              ; BR IF SO.
 3872  034066  104423  034004                 ERROR,  2$              ;; *** STATUS INCORRECT AFTER INPUT XFER ***
 3873  034072  120201                  6$:    CMPB    R2,R1           ; DATA LOOP INTACT ??
 3874  034074  001402                         BEQ     7$              ; BR IF SO.
 3875  034076  104423  034004                 ERROR,  2$              ;; *** DATA INCORRECT AFTER OUT/IN EXCHANGE ***
```

```
 3876  034102  105201                  7$:     INCB    R1
 3877  034104  001337                          BNE     2$                      ; LOOP 'TIL DONE.
 3878
 3879                                  ;*******************************************************************************
  (2)                                  .SBTTL      T145 -- TEST THE PORT B (OUTPUT) INTERRUPT
  (2)                                  ;*******************************************************************************
  (2)  034106  104420  000145          T145:   BEGINTEST, 145                  ;; CHECK SWITCH REGISTER OPTIONS.
 3880  034112  016704  144754                  MOV     PPBV,R4                 ; GET B VECTOR POINTER.
 3881  034116  106464  000002                  MTPS    2(R4)                   ; RAISE CPU TO PIO LEVEL...
 3882  034122  012714  034140                  MOV     #1$,(R4)                ;...AND SET VECTOR.
 3883  034126  112777  000005  144732          MOVB    #<2_1.>!1,@PPCW         ; SET C<2> B INTR ENABLE.
 3884  034134  000240                          240                             ; OUTRDY IS ALREADY THERE, SO...
 3885  034136  000403                          BR      2$                      ;...INTERRUPT REQUEST SHOULD BE THERE...
 3886                                                                          ;...BUT HELD OFF BY THE CPU LEVEL.
 3887  034140  022626                  1$:     CMP     (SP)+,(SP)+             ; BUT IT WASN'T !!
 3888  034142  104423  034106                  ERROR,  T145                    ;; *** PORT B INTERRUPT LEVEL INCORRECT ***
 3889  034146  117703  144712          2$:     MOVB    @PPC,R3                 ; GET STATUS.
 3890  034152  120327  000007                  CMPB    R3,#OUTENA!OUTRDY!OUTINT
 3891  034156  001402                          BEQ     3$
 3892  034160  104423  034106                  ERROR,  T145                    ;; *** STATUS INCORRECT WITH PORT B INTERRUPT PENDING **
 3893
 3894  034164  012714  034204          3$:     MOV     #4$,(R4)                ; OK, CHANGE VECTOR.
 3895  034170  106427  000200                  MTPS    #PRI4                   ; LOWER CPU TO LEVEL 4...
 3896  034174  000240                          240                             ;...INTERRUPT SHOULD COME IN.
 3897  034176  104423  034106                  ERROR,  T145                    ;; *** PORT B INTERRUPT NOT RECEIVED ***
 3898  034200  000401                          SKP1
 3899
 3900  034204  022626                  4$:     CMP     (SP)+,(SP)+             ; WE'RE HERE ON INTERRUPT, FIX STACK.
 3901  034206  012714  034222                  MOV     #5$,(R4)                ; CHANGE THE VECTOR...
 3902  034212  106427  000200                  MTPS    #PRI4                   ;...AND LOWER CPU AGAIN...
 3903  034216  000240                          240                             ;...SHOULDN'T GET ANOTHER INTERRUPT.
 3904  034220  000403                          BR      6$
 3905  034222  022626                  5$:     CMP     (SP)+,(SP)+             ; BUT WE DID -- TROUBLE.
 3906  034224  104423  034106                  ERROR,  T145                    ;; *** DOUBLE PORT B INTERRUPT ***
 3907
 3908  034230  112777  000004  144630  6$:     MOVB    #<2_1.>!0,@PPCW         ; OK, CLEAR THE ENABLE...
 3909  034236  112777  000001  144616          MOVB    #1,@PPB                 ;...WRITE -- SHOULD LOWER INT REQUEST.
 3910  034244  005000                          CLR     R0
 3911  034246  117703  144612          7$:     MOVB    @PPC,R3                 ; GET STATUS
 3912  034252  032703  000001                  BIT     #OUTINT,R3
 3913  034256  001403                          BEQ     8$                      ; BR WHEN REQUEST CLEARS.
 3914  034260  077006                          SOB     R0,7$
 3915  034262  104423  034106                  ERROR,  T145                    ;; *** PORT B INT REQUEST DIDN'T CLEAR AFTER WRITE ***
 3916
 3917  034266  132777  000040  144570  8$:     BITB    #INRDY,@PPC
 3918  034274  001001                          BNE     9$                      ; BR WHEN "INRDY" COMES UP...
 3919  034276  077005                          SOB     R0,8$                   ;...BUT DON'T WAIT FOREVER.
 3920  034300  117700  144554          9$:     MOVB    @PPA,R0                 ; DUMMY READ.
 3921  034304  012714  033644                  MOV     #BADBI,(R4)             ; RESET VECTOR.
 3922
 3923                                  ;*******************************************************************************
  (2)                                  .SBTTL      T146 -- TEST THE PORT A (INPUT) INTERRUPT
  (2)                                  ;*******************************************************************************
  (2)  034310  104420  000146          T146:   BEGINTEST, 146                  ;; CHECK SWITCH REGISTER OPTIONS.
 3924  034314  016704  144550                  MOV     PPAV,R4                 ; GET PORT A VECTOR POINTER.
 3925  034320  106464  000002                  MTPS    2(R4)                   ; RAISE CPU TO PIO LEVEL.
```

```
3926  034324  112777  000011  144534          MOVB    #<4_1.>!1,@PPCW ; SET C<4> A INTR ENABLE.
3927  034332  117703  144526                  MOVB    @PPC,R3
3928  034336  120327  000022                  CMPB    R3,#INENA!OUTRDY ; EXPECT TO SEE THIS.
3929  034342  001402                          BEQ     1$
3930  034344  104423  034310                  ERROR,  T146            ;; *** STATUS INCORRECT ***
3931
3932  034350  012714  034372          1$:     MOV     #2$,(R4)        ; SET THE VECTOR.
3933  034354  005000                          CLR     R0
3934  034356  012701  000123                  MOV     #123,R1
3935  034362  110177  144474                  MOVB    R1,@PPB         ; SEND DATA (B => A) THRU THE LOOP.
3936  034366  077001                          SOB     R0,.            ; INTERRUPT SHOULD BE HELD OFF.
3937  034370  000403                          BR      3$
3938  034372  022626                  2$:     CMP     (SP)+,(SP)+     ; BUT IT WASN'T.
3939  034374  104423  034310                  ERROR,  T146            ;; *** PORT A INTERRUPT LEVEL INCORRECT ***
3940
3941  034400  117703  144460          3$:     MOVB    @PPC,R3         ; GET STATUS.
3942  034404  120327  000072                  CMPB    R3,#INRDY!INENA!ININT!OUTRDY
3943  034410  001402                          BEQ     4$              ; BR IF STATUS IS OK.
3944  034412  104423  034310                  ERROR,  T146            ;; *** STATUS INCORRECT WITH PORT A INTERRUPT PENDING **
3945
3946  034416  012714  034436          4$:     MOV     #5$,(R4)        ; CHANGE VECTOR.
3947  034422  106427  000200                  MTPS    #PRI4           ; LOWER CPU TO LEVEL 4...
3948  034426  000240                          240                     ;...INTERRUPT SHOULD COME IN.
3949  034430  104423  034310                  ERROR,  T146            ;; *** PORT A INTERRUPT NOT RECEIVED ***
3950  034434  000401                          SKP1
3951
3952  034436  022626                  5$:     CMP     (SP)+,(SP)+     ; WE'RE HERE ON INTERRUPT, FIX STACK.
3953  034440  012714  034454                  MOV     #6$,(R4)        ; CHANGE VECTOR...
3954  034444  106427  000200                  MTPS    #PRI4           ;...AND LOWER AGAIN...
3955  034450  000240                          240                     ;...SHOULDN'T GET ANOTHER.
3956  034452  000403                          BR      7$
3957  034454  022626                  6$:     CMP     (SP)+,(SP)+     ; 2ND INTERRUPT, FIX STACK.
3958  034456  104423  034310                  ERROR,  T146            ;; *** DOUBLE PORT A INTERRUPT ***
3959
3960  034462  112777  000010  144376  7$:     MOVB    #<4_1.>!0,@PPCW ; OK, CLEAR THE ENABLE...
3961  034470  117702  144364                  MOVB    @PPA,R2         ;...READ DATA (SHOULD LOWER FLAGS).
3962  034474  005000                          CLR     R0
3963  034476  117703  144362          8$:     MOVB    @PPC,R3         ; READ STATUS.
3964  034502  032703  000010                  BIT     #ININT,R3
3965  034506  001403                          BEQ     9$              ; BR WHEN INT REQUEST CLEARS.
3966  034510  077006                          SOB     R0,8$
3967  034512  104423  034310                  ERROR,  T146            ;; *** PORT A INT REQUEST DIDN'T CLEAR AFTER READ ***
3968
3969  034516  012714  033654          9$:     MOV     #BADAI,(R4)     ; RESET PORT A VECTOR.
3970
3971                                  ;*******************************************************************************
 (2)                                .SBTTL      T147 -- TEST THAT B PRIORITY IS HIGHER THAN A
 (2)                                ;*******************************************************************************
 (2)  034522  104420  000147          T147:   BEGINTEST, 147          ;; CHECK SWITCH REGISTER OPTIONS.
3972  034526  016704  144340                  MOV     PPBV,R4         ; GET B VECTOR POINTER.
3973  034532  106464  000002                  MTPS    2(R4)           ; RAISE CPU TO DEVICE LEVEL.
3974  034536  012714  034632                  MOV     #3$,(R4)        ; SET B VECTOR.
3975  034542  012764  034622  000004          MOV     #2$,4(R4)       ; SET A VECTOR.
3976  034550  112777  000005  144310          MOVB    #<2_1.>!1,@PPCW ; ENABLE B...
3977  034556  112777  000011  144302          MOVB    #<4_1.>!1,@PPCW ;...AND A INTERRUPTS.
3978  034564  112777  000001  144270          MOVB    #1,@PPB         ; OUTPUT DATA...
```

```
3979  034572  005000                          CLR     R0
3980  034574  132777  000040  144262  1$:     BITB    #INRDY,@PPC     ;...AND WAIT 'TIL DONE.
3981  034602  001001                          BNE     .+4
3982  034604  077005                          SOB     R0,1$           ; KEEP-ALIVE.
3983
3984  034606  106427  000200                  MTPS    #PRI4           ; LOWER CPU, B SHOULD COME IN FIRST.
3985  034612  000240                          240
3986  034614  104423  034522                  ERROR,  T147            ;; *** NEITHER ONE INTERRUPTED -- WHAT THE HELL !! ***
3987  034620  000415                          BR      4$
3988  034622  022626                  2$:     CMP     (SP)+,(SP)+     ; WE'RE HERE IF A CAME IN FIRST.
3989  034624  104423  034522                  ERROR,  T147            ;; *** A INTERRUPTED BEFORE B -- BAD NEWS !! ***
3990  034630  000411                          BR      4$
3991
3992  034632  022626                  3$:     CMP     (SP)+,(SP)+     ; WE'RE HERE IF B INTERRUPTED FIRST.
3993  034634  012764  034654  000004          MOV     #4$,4(R4)       ; CHANGE THE A VECTOR...
3994  034642  106427  000200                  MTPS    #PRI4           ;...AND LET IT IN NOW.
3995  034646  000240                          240
3996  034650  104423  034522                  ERROR,  T147            ;; *** A INT NOT RECEIVED AFTER B INT ***
3997
3998  034654  112777  000004  144204  4$:     MOVB    #<2_1.>!0,@PPCW ; CLEAR THE ENABLES.
3999  034662  112777  000010  144176          MOVB    #<4_1.>!0,@PPCW
4000  034670  112777  000001  144164          MOVB    #1,@PPB         ; LOWER REQUESTS.
4001  034676  117700  144156                  MOVB    @PPA,R0
4002  034702  012714  033644                  MOV     #BADBI,(R4)     ; RESET VECTORS.
4003  034706  012764  033654  000004          MOV     #BADAI,4(R4)
4004
4005  034714  012767  000006  143062  PIODUN: MOV     #ERRVEC+2,ERRVEC
4006  034722  004767  000002                  CALL    ENDSEG          ; CHECK FOR LOOP-IN-SEGMENT...
4007  034726  000410                          BR      ENDPAS          ;...RETURN AND DO ENDPASS IF NOT.
4008                                  .SBTTL
```

```
                                        .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 RC,. 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.
                                        ;
034730  005767  000730        ENDSEG: TST     INSEG           ; LOOP-IN-SEGMENT ??
034734  001001                        BNE     1$              ; YES.
034736  000207                        RETURN                  ; JUST RETURN IF NOT.

034740  104007                1$:     EMT+7                   ; <CRLF>
034742  016700  001232                MOV     SEGN,R0         ; CURRENT SEGMENT DONE.
034746  000402                        SKP2
034750  012700  035130        ENDPAS: MOV     #EOP,R0         ; ALL SEGMENTS DONE.
034754  104003                        EMT+3
034756  012700  035141                MOV     #EOP1,R0
034762  104003                        EMT+3                   ; END PASS.
034764  005267  147012                INC     TPASS           ; BUMP TOTAL PASS COUNT.
034770  026727  147004                CMP     ERRCNT,(PC)+    ; ANY ERRORS THIS PASS ??
034774  000000                1$:     0
034776  001002                        BNE     2$              ; SKIP IF SO.
035000  005267  147000                INC     GPASS           ; NO. BUMP CONSECUTIVE-GOOD PASS COUNT.
035004  016767  146770  177762  2$:   MOV     ERRCNT,1$       ; UPDATE SAVED COUNT.
035012  016700  146764                MOV     TPASS,R0
035016  104424                        TRAP+24                 ; PASS COUNT...
035020  012700  035155                MOV     #EOPE,R0
035024  104003                        EMT+3
035026  016700  146746                MOV     ERRCNT,R0
035032  104424                        TRAP+24                 ;...AND TOTAL ERROR COUNT.
035034  104007                        EMT+7                   ; <CRLF>
035036  032767  010000  143132        BIT     #10000,SWR      ; ERROR SUMMARY ??
035044  001402                        BEQ     3$
035046  004767  000124                CALL    ERRSUM          ; PRINT IT IF SO.
035052  106427  000000        3$:     MTPS    #PRIO           ; LOWER THE CPU.
035056  005000                        CLR     R0
035060  012701  000004                MOV     #4,R1
035064  077001                4$:     SOB     R0,.            ; ENDPASS DELAY...
035066  077102                        SOB     R1,4$           ;...ABOUT 1 SECOND.

035070  013700  000042                MOV     @#42,R0
035074  001405                        BEQ     5$              ; BR IF NOT CHAINING.
035076  000240                        NOP                     ; OR RESET.
035100  004710                        CALL    (R0)            ; SYSMAC STYLE CHAIN RETURN.
035102  000240  000240  000240        240,240,240

035110  016700  000550        5$:     MOV     INSEG,R0        ; GET ITERATION ADDRESS.
035114  001002                        BNE     6$
035116  012700  010336                MOV     #DOALL,R0       ; IF NONE, BACK TO SQUARE 1.
035122  012706  003776        6$:     MOV     #STACK,SP       ; RESET STACK...
035126  000110                        JMP     (R0)            ;...AND LOOP FOREVER.

035130  005015  045516  040530  EOP:    .ASCIZ  <15><12>'NKXAA0'
035141     054  042440  042116  EOP1:   .ASCIZ  ', END PASS '
```

```
 4066  035155     054  052040  052117  EOPE:   .ASCIZ  ', TOTAL ERRORS '
 4067          035176                          .EVEN
 4068
 4069                                  ;*****************************************************************************
 4070                                  ; ERROR SUMMARY REPORT.
 4071                                  ; CALLED IN "END-PASS" IF SWR<12> IS SET.
 4072                                  ;
 4073  035176  012702  004006          ERRSUM: MOV     #EPCTBL,R2
 4074  035202  005712                          TST     (R2)
 4075  035204  001414                          BEQ     2$              ; BR IF NO ERROR DATA LOGGED.
 4076  035206  012700  035240                  MOV     #ERHEAD,R0
 4077  035212  104003                          EMT+3                   ; HEADER.
 4078  035214                          1$:
 4082  035214  104010                          EMT+10                  ;; <TAB>
  (1)  035216  012200                          MOV     (R2)+,R0
  (1)  035220  104424                          TRAP+24                 ;; PRINT (R2)+
  (1)  035222  104010                          EMT+10                  ;; <TAB>
  (1)  035224  012200                          MOV     (R2)+,R0
  (1)  035226  104424                          TRAP+24                 ;; PRINT (R2)+
 4083  035230  104007                          EMT+7                   ; <CRLF>
 4084  035232  005712                          TST     (R2)
 4085  035234  001367                          BNE     1$              ; LOOP 'TIL DONE.
 4086  035236  000207                  2$:     RETURN
 4087
 4088  035240  005015  042411  051122  ERHEAD: .ASCIZ  <15><12><11>'ERRPC'<11>' TIMES'<15><12>
 4089                                          .EVEN
 4090
 4091                                  ;*****************************************************************************
 4092                                  ; SUBROUTINE TO CHECK FOR Q-BUS EXERCISOR.
 4093                                  ; RETURN QBX CSR1 ADDRESS IN R0 IF IT'S THERE.
 4094                                  ;
 4095  035262  003767  143550          QBXIN:  TST     QBXA            ; QBX INSTALLED ??
 4096  035266  001020                          BNE     3$              ; BR IF SO.
 4097  035270  012767  035314  142506          MOV     #1$,ERRVEC      ; DON'T KNOW YET, SET TRAP CATCHER...
 4098  035276  012700  174020                  MOV     #QBXCSR,R0      ;...AND A POINTER.
 4099  035302  005010                          CLR     (R0)            ; CLEAR QBX CSR1 (IF POSSIBLE)...
 4100  035304  000240                          240                     ;...TRAP TO 1$ IF NOT.
 4101  035306  010067  143524                  MOV     R0,QBXA         ; IT'S THERE, SET FLAG = CSR ADDRESS...
 4102  035312  000403                          BR      2$              ;...AND RETURN.
 4103
 4104  035314  005067  143516          1$:     CLR     QBXA            ; NOT THERE, CLEAR FLAG.
 4105  035320  022626                          CMP     (SP)+,(SP)+     ; FIX STACK.
 4106  035322  012767  000006  142454  2$:     MOV     #ERRVEC+2,ERRVEC ; RESET ERROR VECTOR.
 4107  035330  016700  143502          3$:     MOV     QBXA,R0         ; RETURN CSR POINTER OR ZERO.
 4108  035334  000207                          RETURN
```

```
                                .SBTTL  TRAP HANDLER
                                ;*****************************************************************
                                ; TRAP HANDLER
                                ;
035336  010046                  TRAPHAN: MOV     R0,-(SP)        ; SAVE CALLERS R0...
035340  010146                           MOV     R1,-(SP)        ;...AND R1.
035342  016600  000004                   MOV     4(SP),R0        ; GET TRAP PC.
035346  116000  177776                   MOVB    -2(R0),R0       ; GET TRAP CODE.
035352  010001                           MOV     R0,R1           ; WORKING COPY.
035354  162701  000017                   SUB     #17,R1
035360  100001                           BPL     1$              ; BR IF TRAP+20 OR HIGHER.
035362  005001                           CLR     R1              ; TRAP+0 THRU TRAP+17 ARE COMMON.
035364  006301                  1$:      ASL     R1              ; SHIFT UP TO WORD OFFSET...
035366  016107  035372                   MOV     2$(R1),PC       ;...AND DISPATCH THERE.

035372  035406                  2$:      CCBIT                   ; TRAP0 - TRAP17 = CC BIT CHECK.
035374  035436                           BGNTST                  ; TRAP20 = BEGIN NEW TEST.
035376  035460                           CHKSWR                  ; TRAP21 = CHECK SWR.
035400  035470                           NEWSWR                  ; TRAP22 = GET NEW SWR.
035402  035666                           LOGERR                  ; TRAP23 = LOG AND REPORT ERROR.
035404  036464                           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.
                                ;
035406  016601  000006          CCBIT:   MOV     6(SP),R1        ; GET PSW TO CHECK.
035412  042701  177760                   BIC     #^C17,R1        ; KEEP CC BITS ONLY.
035416  020001                           CMP     R0,R1
035420  001003                           BNE     1$              ; BR IF CC BITS INCORRECT.
035422  062766  000004  000004           ADD     #4,4(SP)        ; OK, ADJUST RETURN PC.
035430  012601                  1$:      MOV     (SP)+,R1
035432  012600                           MOV     (SP)+,R0
035434  000002                           RTI

                                ;*****************************************************************
                                ; BEGIN TEST ROUTINE.
                                ; CALL VIA TRAP+20
                                ;
035436  104421                  BGNTST:  CHECKSWR
035440  012601                           MOV     (SP)+,R1        ; RESTORE CALLERS REGISTERS.
035442  012600                           MOV     (SP)+,R0
035444  017667  000000  000524           MOV     @(SP),TSTNUM    ; SAVE TEST NUMBER...
                                                                 ;...FOR THE ERROR HANDLER...
035452  062716  000002                   ADD     #2,(SP)         ;...BUMP RETURN PC...
035456  000002                           RTI                     ;...AND RETURN.
```

```
                        ;*****************************************************************************
                        ; SWITCH REGISTER HANDLER.
                        ; CALLED VIA TRAP+21 (CHECKSWR) AND TRAP+22 (GETSWR).
                        ;
                                .ENABL  LSB
035460  104005          CHKSWR: EMT+5                   ; CHECK KEYBOARD ( => R0 ).
035462  120027  000007          CMPB    R0,#7           ; <^G> ??
035466  001024                  BNE     1$              ; JUST RETURN IF NOT.

035470  005737  000042  NEWSWR: TST     @#42            ; CHAINING ??
035474  001021                  BNE     1$              ; DON'T IF SO.
035476  012700  035546          MOV     #OSWR,R0
035502  104003                  EMT+3                   ; OLD SWR...
035504  016700  142466          MOV     SWR,R0
035510  104424                  TRAP+24                 ;...= THIS.
035512  012700  035557          MOV     #NSWR,R0
035516  104003                  EMT+3                   ; NEW SWR ...
035520  104000                  EMT+0                   ;...AND WAIT FOR INPUT.
035522  104011                  EMT+11                  ; PARSE OCTAL NUMBER.
035524  000405                  BR      1$              ; JUST EXIT IF NO CHANGE.
035526  120067  142444          CMPB    R0,SWR          ; NEW SEGMENT SELECTED ??
035532  001016                  BNE     NEWSEG          ; BR IF SO.
035534  010067  142436          MOV     R0,SWR          ; OTHERWISE, SAVE NEW SWR...
035540  012601          1$:     MOV     (SP)+,R1        ;...RESTORE...
035542  012600                  MOV     (SP)+,R0
035544  000002                  RTI                     ;...AND RETURN TO CALLER.
                                .DSABL  LSB
035546  005015  053523  020122 OSWR:   .ASCIZ  <15><12>'SWR = '
035557     040  047040  053505 NSWR:   .ASCIZ  '  NEW = '
                                .EVEN
                        ;
                        ; NEW SEGMENT REQUIRED.  GET ADDRESS AND DO A RESTART-IN-SEGMENT.
                        ;
035570  010067  142402  NEWSEG: MOV     R0,SWR          ; SAVE NEW SWR.
035574  042700  177770          BIC     #^C7,R0         ; STRIP OFF SEGMENT NUMBER.
035600  006300                  ASL     R0              ; SHIFT NUMBER UP TO WORD OFFSET...
035602  016067  035644  000054        MOV     3$(R0),INSEG    ;...AND SET/CLEAR "INSEG" FLAG.
035610  012700  004000          MOV     #ERRCNT,R0
035614  012701  002000          MOV     #1024.,R1
035620  005020          1$:     CLR     (R0)+           ; CLEAR THE ERROR TABLE AND COUNTERS.
035622  077102                  SOB     R1,1$
035624  016700  000034          MOV     INSEG,R0        ; GET LOOP-SEGMENT ADDRESS.
035630  001002                  BNE     2$
035632  012700  010336          MOV     #DOALL,R0       ; IF ZERO, RESTORE "RUN-ALL" MODE.
035636  012706  003776  2$:     MOV     #STACK,SP       ; NOW, RESET STACK...
035642  000110                  JMP     (R0)            ;...AND RESTART-IN-SEGMENT.
035644  000000          3$:     0                       ; SEGMENT = 0 = RUN 'EM ALL...
035646  010336  024030  027052        S1,S2,S3        ;...OTHERWISE, RUN JUST ONE OF THESE.
035654  027642  027764  030014        S4,S5,S6
035662  033512                  S7
035664  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.
                                ;
035666  012700  004000        LOGERR: MOV     #ERRCNT,R0      ; GET TABLE POINTER...
035672  016601  000004                MOV     4(SP),R1        ;...AND CURRENT (ERROR) PC.
035676  005220                        INC     (R0)+           ; BUMP TOTAL ERROR COUNT.
035700  122020                        CMPB    (R0)+,(R0)+     ; SKIP OVER TOTAL PASS COUNT...
035702  005020                        CLR     (R0)+           ;...AND CLEAR "CONSECUTIVE-GOOD" COUNT.
035704  020110                1$:     CMP     R1,(R0)         ; THIS PC ALREADY LOGGED ??
035706  001410                        BEQ     3$              ; JUST BUMP ITS COUNT IF SO.
035710  005710                        TST     (R0)            ; NO, END OF TABLE ??
035712  001405                        BEQ     2$              ; STORE PC HERE IF SO.
035714  020027  007772                CMP     R0,#LOGEND-4    ; TABLE FULL ??
035720  103005                        BHIS    4$              ; YES, FORGET IT !!!
035722  022020                        CMP     (R0)+,(R0)+     ; NONE OF THE ABOVE, BUMP POINTER...
035724  000767                        BR      1$              ;...AND LOOP 'TIL WE FIND ONE...
                                                              ;...OR THE OTHER.
035726  010110                2$:     MOV     R1,(R0)         ; STORE ERROR PC...
035730  005260  000002        3$:     INC     2(R0)           ;...AND BUMP ITS COUNTER...
035734                        4$:                             ;...AND FALL THRU.

035734  032767  020000  142234  PNTERR: BIT     #20000,SWR      ; ERROR MESSAGES INHIBITED ??
035742  001062                        BNE     1$              ; BR IF SO.
035744  010667  000224                MOV     SP,ESP          ; GET STACK POINTER...
035750  062767  000010  000216        ADD     #10,ESP         ;...ADJUST TO SP AT TIME OF ERROR.
035756  104007                        EMT+7                   ; <CRLF>
035760  016700  000214                MOV     SEGN,R0
035764  104003                        EMT+3                   ; CURRENT SEGMENT...
035766  012700  036357                MOV     #FINTST,R0
035772  104003                        EMT+3                   ;...AND TEST...
035774  016700  000176                MOV     TSTNUM,R0
036000  104424                        TRAP+24                 ;...NUMBER.
036002  012700  036377                MOV     #HDR,R0
036006  104003                        EMT+3                   ;...HEADER.
036010  016600  000004                MOV     4(SP),R0
(1) 036014  104424                    TRAP+24                 ;; PRINT 4(SP)
(1) 036016  104010                    EMT+10                  ;; <TAB>
(1) 036020  016600  000006            MOV     6(SP),R0
(1) 036024  104424                    TRAP+24                 ;; PRINT 6(SP)
(1) 036026  104010                    EMT+10                  ;; <TAB>
(1) 036030  016600  000002            MOV     2(SP),R0
(1) 036034  104424                    TRAP+24                 ;; PRINT 2(SP)
(1) 036036  104010                    EMT+10                  ;; <TAB>
(1) 036040  011600                    MOV     (SP),R0
(1) 036042  104424                    TRAP+24                 ;; PRINT (SP)
(1) 036044  104010                    EMT+10                  ;; <TAB>
(1) 036046  010200                    MOV     R2,R0
(1) 036050  104424                    TRAP+24                 ;; PRINT R2
(1) 036052  104010                    EMT+10                  ;; <TAB>
(1) 036054  010300                    MOV     R3,R0
```

```
   (1)  036056  104424                        TRAP+24                 ;; PRINT R3
   (1)  036060  104010                        EMT+10                  ;; <TAB>
   (1)  036062  010400                        MOV     R4,R0
   (1)  036064  104424                        TRAP+24                 ;; PRINT R4
   (1)  036066  104010                        EMT+10                  ;; <TAB>
   (1)  036070  010500                        MOV     R5,R0
   (1)  036072  104424                        TRAP+24                 ;; PRINT R5
   (1)  036074  104010                        EMT+10                  ;; <TAB>
   (1)  036076  016700  000072                MOV     ESP,R0
   (1)  036102  104424                        TRAP+24                 ;; PRINT ESP
   (1)  036104  104010                        EMT+10                  ;; <TAB>
 4262   036106  104007                        EMT+7                   ; <CRLF>
 4263   036110  012601                1$:     MOV     (SP)+,R1        ; RESTORE REGISTERS.
 4264   036112  012600                        MOV     (SP)+,R0
 4265   036114  005767  142056                TST     SWR             ; HALT-ON-ERROR ( SWR<15> ) ??
 4266   036120  100012                        BPL     2$              ; BR IF NOT.
 4267   036122  013767  177564  000042        MOV     @#177564,SAVTTY ; YES, SAVE CONSOLE TTY STATUS...
 4268   036130  042737  000004  177564        BIC     #4,@#177564     ;...AND INSURE 'MAINT' IS NOT SET.
 4269   036136  000000                        EHALT                   ; ERROR HALT.
 4270   036140  016737  000026  177564        MOV     SAVTTY,@#177564 ; ON PROCEED, RESTORE TTY STATUS.
 4271   036146  032767  001000  142022  2$:   BIT     #1000,SWR       ; LOOP-ON-ERROR ( SWR<9> ) ??
 4272   036154  001003                        BNE     3$              ; BR IF SO.
 4273   036156  062716  000002                ADD     #2,(SP)         ; NO, ADJUST RETURN PC...
 4274   036162  000002                        RTI                     ;...AND RETURN TO CALLER.
 4275   036164  017616  000000        3$:     MOV     @(SP),(SP)      ; REPLACE RETURN PC WITH LOOP ADDRESS...
 4276   036170  000002                        RTI                     ;...AND DO IT.
 4277
 4278   036172  000000                SAVTTY: 0
 4279   036174  000000                ESP:    0
 4280   036176  000000                TSTNUM: 0
 4281   036200  036202                SEGN:   SEG1                    ; POINTS TO ONE OF THE FOLLOWING:
 4282   036202  030524  020061  047111 SEG1:   .ASCIZ  'T11 INSTRUCTION'
 4283   036222  030524  020061  051124 SEG2:   .ASCIZ  'T11 TRAP/INTERRUPT'
 4284   036245     114  041517  046101 SEG3:   .ASCIZ  'LOCAL 2KW RAM'
 4285   036263     114  041517  046101 SEG4:   .ASCIZ  'LOCAL 1KW ROM'
 4286   036301     123  051105  040511 SEG5:   .ASCIZ  'SERIAL LINE 1'
 4287   036317     123  051105  040511 SEG6:   .ASCIZ  'SERIAL LINE 2'
 4288   036335     120  051101  046101 SEG7:   .ASCIZ  'PARALLEL I/O PORT'
 4289   036357     040  040506  046111 FINTST: .ASCIZ  ' FAILS IN TEST '
 4290   036377     015  020012  041520 HDR:    .ASCII  <15><12>' PC+2'<11>
 4291   036407     040  050040  053523         .ASCII  '  PSW'<11>
 4292   036415     040  051040  004460         .ASCII  '  R0'<11>
 4293   036422  020040  030522     011         .ASCII  '  R1'<11>
 4294   036427     040  051040  004462         .ASCII  '  R2'<11>
 4295   036434  020040  031522     011         .ASCII  '  R3'<11>
 4296   036441     040  051040  004464         .ASCII  '  R4'<11>
 4297   036446  020040  032522     011         .ASCII  '  R5'<11>
 4298   036453     122  024066  050123         .ASCIZ  'R6(SP)'<15><12>
 4299                                           .EVEN
```

```
4301                                    ;*****************************************************************************
4302                                    ; CONVERT INTEGER TO ASCII AND PRINT.
4303                                    ; CALL VIA TRAP+24, NUMBER TO CONVERT IS 2ND WORD ON THE STACK.
4304                                    ;
4305                                    ; CODE ADDED BECAUSE EMT+30 (INTEGER TO ASCII) FUNCTIONALITY IS
4306                                    ; DIFFERENT BETWEEN XXDP+ V1.0 AND V1.1.  THIS ROUTINE MAKES THAT
4307                                    ; DIFFERENCE TRANSPARENT TO THE CALLER.
4308                                    ;
4309  036464  016600  000002           ITOA:   MOV     2(SP),R0        ; NUMBER => R0.
4310  036470  012701  036522                   MOV     #2$,R1          ; IF V1.1, PACK THE ASCII STRING AT 2$.
4311  036474  005011                           CLR     (R1)
4312  036476  104030                           EMT+30                  ; CONVERT I TO A.
4313  036500  005767  000016                   TST     2$              ;
4314  036504  001403                           BEQ     1$              ; IF V1.0, IT'S ALREADY BEEN PRINTED.
4315  036506  012700  036522                   MOV     #2$,R0          ; IF V1.1, PRINT IT NOW.
4316  036512  104003                           EMT+3
4317  036514  012601                   1$:     MOV     (SP)+,R1        ; RESTORE...
4318  036516  012600                           MOV     (SP)+,R0
4319  036520  000002                           RTI                     ;...AND RETURN.
4320
4321  036522  031061  032063  033065  2$:     .ASCIZ  /123456/
4322          036532                           .EVEN
4323
4324  036532  036532                   $LSTAD: .
4325
4326          010000                           .END    BEGIN
```

```
                    1094    1104    1113    1123    1130    1136    1146    1153    1160    1176    1189    1196    1203
                    1208    1212    1215    1464    1467    1470    1473    1492    1513    1519    1526    1529    1537
                    1547    1555    1559    1566    1571    1578    1581    1585    1588    1602    1609    1619    1625
                    1632    1645    1651    1667    1670    1683    1689    1692    1705    1708    1713    1725    1731
                    1739    1747    1754    1763    1770    1774    1782    1790    1793    1798    1801    1810    1819
                    1822    1825    1860    1866    1874    1882    1885    1898    1910    1915    1918    1921    1928
                    1931    1935    1938    1946    1953    1962    1970    1977    1990    1996    2013    2016    2029
                    2035    2038    2051    2054    2059    2071    2077    2085    2093    2100    2109    2116    2120
                    2128    2137    2144    2155    2164    2174    2181    2187    2200    2213    2221    2228    2233
                    2241    2252    2264    2271    2275    2289    2310    2313    2321    2324    2336    2344    2368
                    2376
CHECKS= 104421       93#    4156
CHKSWR  035460     4127     4169#
CLNZ  = 000254       84#    1085    2135
CMPB0   011476      540#
CMPB1   016722     1487#
CMP0    013104      822#
CMP1    020516     1805#
COMB0   012052      622#
COMB1   017414     1596#
COM0    013456      906#
COM1    021362     1942#
DFPBR = 001040      167#    3169*   3207    3267    3282    3296    3310    3324    3338    3353    3364    3383    3449
                   3455     3519    3569    3639    3666    3709    3736    3766
DFSPD = 001042      168#    3451*   3523    3574    3623    3648    3738
DLDUN   033452     3175     3177    3179    3181    3245    3734    3748    3751    3766#
DLSET   030042     3139     3149#
DLTSTS  030236     3170     3178    3180    3190#
DJALL   010336      299      308#   4060    4207
EHALT = 000000      101#     228    4269
EMTTRP  010140      261#
EMTVEC= 000030       70#     261     262*    289*    293*    2694    2695*   2696*   2703*
ENDPAS  034750     4007     4026#
ENDSEG  034730     2382*    2911*   3072*   3111*   3769*   4006*   4019#
EOP     035130     4026     4064#
EOPE    035155     4038     4066#
EOP1    035141     4028     4065#
EPCTBL= 004006      193#    4073
ERHEAD  035240     4076     4088#
ERRCNT= 004000      190#    4031    4035    4040    4201    4226
ERROR = 104423       95#     314     318     334     345     358     370     384     390     394     411     418     420
                    423      430     432     435     442     445     452     454     457     463     464     467     473
                    476      478     482     487     491     495     502     504     507     510     514     517     528
                    531      534     537     543     546     550     553     557     561     570     573     576     579
                    582      585     592     596     599     602     605     609     612     616     619     626     629
                    633      636     641     646     649     653     656     665     668     671     675     683     686
                    689      692     701     704     707     710     718     721     724     729     732     740     746
                    749      754     757     762     768     771     777     780     784     788     791     796     802
                    805      809     812     815     819     826     829     832     835     845     848     851     854
                    857      861     865     868     876     880     883     886     889     893     896     900     903
                    910      913     917     920     925     930     933     937     940     949     952     955     959
                    967      970     973     976     985     988     991     994    1002    1005    1008    1013    1016
                   1024     1030    1033    1038    1041    1046    1052    1055    1061    1064    1068    1072    1075
                   1080     1088    1091    1095    1098    1105    1108    1114    1117    1124    1131    1137    1140
                   1147     1150    1154    1157    1161    1166    1171    1177    1180    1185    1190    1193    1197
                   1204     1207    1209    1213    1216    1219    1231    1237    1248    1254    1272    1284    1302
```

```
                    1315    1326    1337    1344    1353    1368    1388    1395    1413    1429    1436    1451    1456
                    1465    1468    1471    1474    1480    1483    1493    1496    1499    1502    1514    1517    1520
                    1523    1527    1530    1538    1541    1544    1548    1556    1560    1567    1572    1575    1579
                    1582    1586    1589    1592    1603    1606    1610    1614    1620    1626    1629    1633    1636
                    1646    1649    1652    1656    1665    1668    1671    1674    1684    1687    1690    1693    1703
                    1706    1709    1714    1717    1726    1732    1735    1740    1743    1748    1755    1758    1764
                    1767    1771    1775    1778    1783    1791    1794    1799    1802    1811    1814    1817    1820
                    1823    1826    1837    1840    1847    1861    1864    1867    1870    1875    1878    1883    1886
                    1892    1899    1902    1911    1916    1919    1922    1925    1929    1932    1936    1939    1947
                    1950    1954    1957    1963    1971    1974    1978    1981    1991    1994    1997    2002    201
                    2014    2017    2020    2030    2033    2036    2039    2049    2052    2055    2060    2063    207
                    2078    2081    2086    2089    2094    2101    2104    2110    2113    2117    2121    2124    2129
                    2138    2141    2145    2148    2156    2159    2165    2168    2175    2182    2188    2191    2201
                    2204    2209    2214    2217    2222    2229    2234    2238    2242    2245    2253    2256    2260
                    2265    2268    2272    2276    2279    2286    2290    2293    2296    2301    2311    2314    2318
                    2322    2325    2328    2337    2340    2345    2348    2355    2359    2363    2369    2372    2377
                    2380    2396    2401    2406    2412    2415    2421    2424    2435    2444    2453    2462    2471
                    2480    2483    2486    2489    2492    2495    2498    2501    2515    2521    2525    2546    2551
                    2554    2559    2562    2575    2580    2583    2586    2589    2600    2605    2608    2611    2614
                    2625    2630    2633    2636    2639    2666    2673    2676    2679    2682    2706    2709    2712
                    2715    2736    2739    2742    2745    2760    2765    2768    2771    2781    2786    2789    2792
                    2807    2812    2827    2837    2839    2851    2854    2856    2877    2882    2902    2907    2940
                    2955    2959    2987    2999    3009    3017    3044    3047    3059    3062    3069    3102    3108
                    3174    3176    3178    3180    3211    3214    3217    3220    3230    3233    3236    3239    3242
                    3257    3271    3274    3285    3288    3299    3302    3313    3316    3327    3330    3341    3344
                    3357    3360    3370    3375    3379    3394    3400    3413    3472    3489    3505    3508    3533
                    3541    3544    3549    3554    3559    3582    3591    3599    3602    3610    3615    3618    3632
                    3635    3654    3662    3679    3685    3698    3728    3750    3762    3820    3822    3824    3840
                    3853    3862    3867    3872    3875    3888    3892    3897    3906    3915    3930    3939    3944
                    3949    3958    3967    3986    3989    3996
ERRSUM  055176      4045*   4073#
ERRVEC= 000004        64#   2510*   2527*   2558*   2559*   2566*   2834*   2840*   2927*   2973*   3029*   3071*   3090*
                    3110*   3152*   3768*   3800*   4005*   4097*   4106*
ESP     036174      4246*   4247*   4261    4279#
FINTST  036357      4251    4289#
FLT0    030224      3187#
FLT1    030214      3186#   3251    3754
FRERR = 020000      3127#   3613    3630
GETSWR= 104422        94#    297
GPASS = 004004       192#   4034*
HDR     036377      4255    4290#
HELLO   010300       295     302#
INCB0   011714       588#
INCB1   017214      1551#
INC0    013320       872#
INC1    021212      1906#
INENA = 000020      3786#   3928    3942
ININT = 000010      3787#   3942    3964
INIT    010000       220#
INIT1   010072       235     241#
INRDY = 000040      3785#   3864    3917    3942    3980
INSEG   035664       298    4019    4058    4200*   4205    4214#
IOTVEC= 000020        68#   2620*   2621*   2641*   2799*   2800*   2814*
ITOA    036464      4130    4309#
JMP1    010712       415#
JMP2    010752       427#
```

```
JMP3    011006            434     437#
JMP4    011056            449#
JMP5    011112            456     459#
JMP6    011162            471#
JMP7    011234            480     484#
JSRTST  011322            499#
KXT11 = 000004             83#   2852
K0    = 001016            148#    159   2475*  2487
K1    = 001020            149#    160   2429*  2430  2433  2438*  2439  2442  2446*  2448  2455*  2457  2464*
                         2466   2469   2476*  2478  2481  2484   2490  2499
K2    = 001022            150#    161   2477*  2484  2487  2490   2493  2496  2499
K3    = 001024            151#    162
K4    = 001026            152#    163   2391   2394  2399  2404   2410  2422
K5    = 001030            153#    164   2428*  2431  2437*  2447*  2449  2451  2456*  2458  2460  2465*  2467
K6    = 001032            154#    165   2440
K7    = 001034            155#    166
LEDBIT= 000200           3784#
LOGEND= 007776            196#   4235
LOGERR  035666           4129    4226#
MFPT  = 000007             82#   2847
MODE0   015224           1226#
MODE1   015270           1236    1239#
MODE2   015364           1258#
MODE3   015506           1288#
MODE4   015664           1319#
MODE5   016112           1372#
MODE6   016322           1417#
MODE7   016456           1440#
MOV0    013010            799#
MOV1    020430           1786#
NBIT    010434            336#
NEGB0   012144            643#    646    649    653    656
NEGB1   017524           1622#   1626   1629   1633   1636
NEG0    013550            927#    930    933    937    940
NEG1    021464           1965#   1971   1974   1978   1981
NEWSEG  035570           4185    4197#
NEWSWR  035470           4128    4173#
NOBIT   010374            320#
NOP1  = 000260             87#    325
NSWR    035557           4179    4192#
ODT   = 172000             99     106#
ODTBRK= 170000            108#    246
ODTIN = 000140            109#
ODTRE = 172004            100     107#
ORERR = 040000           3128#   3652   3660
OSWR    035546           4175    4191#
OUTENA= 000004           3789#   3890
OUTINT= 000001           3791#   3890   3912
OUTRDY= 000002           3790#   3851   3860   3864   3870   3890   3928   3942
PIOA  = 176200            128#   3797
PIOAV = 000134            129#
PIOB  = 176202            130#
PIOBV = 000130            131#   3798
PIOC  = 176204            133#
PIODUN  034714           3821    3823   3825   4005#
PIOMC = 176206            134#
```

```
PIOP   = 000240   132#  3799
PIOTST  033664   3816   3822   3824   3826#
PNTERR  035734   4244#
PPA    = 001060   177#  3802*  3837   3849   3868   3920   3961   4001
PPAV   = 001070   181#  3813*  3924
PPB    = 001062   178#  3804*  3858*  3909*  3935*  3978*  4000*
PPBV   = 001072   182#  3810*  3880   3972
PPC    = 001064   179#  3806*  3850   3859   3863   3869   3889   3911   3917   3927   3941   3963   3980
PPCW   = 001066   180#  3808*  3844*  3845*  3846*  3883*  3908*  3926*  3960*  3976*  3977*  3998*  3999*
PPM0   = 000253   3781#
PPM1   = 000264   3782#  3844
PRGSIZ= 000076    19#
PRI0   = 000000    79#   250*  2511*  2829*  2880*  2909*  3221*  3268*  3354*  3397*  3408*  3509*  3636*
                 3682*  3693*  4046*
PRI1   = 000040    78#
PRI2   = 000100    77#
PRI3   = 000140    76#
PRI4   = 000200    75#   123   2870   2871*  3895*  3902*  3947*  3954*  3984*  3994*
PRI5   = 000240    74#   126    132   2905*
PRI6   = 000300    73#  2540*  2560   2572*  2587   2597*  2612   2622*  2637   2656*  2680   2697*  2713
                 2727*  2743   2896*  3196*  3348*  3498*  3589*  3595*
PRI7   = 000340    72#   247   3262*  3568*  3573
PWRVEC= 000024    69#  2654*  2655*  2684*
QBXA   = 001036   157#  4095   4101*  4104*  4107
QBXCSR= 174020   136#  4098
QBXIN   035262   2645*  2861*  2890*  3022*  4095#
QBXVEC= 000110   137#  2867   2884
RAMADR= 160010   111#  2918
RAMSIZ= 003764   112#  2919
RBUF   = 001046   171#  3155*  3204   3225   3529   3555   3571   3606   3625   3629   3641   3650   3659
                 3701   3718   3725   3737   3753   3759   3767
RCSR   = 001044   170#  3153*  3199*  3203   3208*  3224   3499*  3500   3501*  3502   3530   3537   3545
                 3550   3556   3580   3583   3627   3657   3672   3674*  3678*  3684*  3697*  3700*  3716
                 3723   3733   3743   3747   3757   3771
RCVACT= 004000   3125#  3531   3538   3546
RCVBRK= 004000   3126#  3608   3630
RCVDUN= 000200   3124#  3531   3542   3551   3557
RCVERR= 100000   3129#  3611   3630   3660
RCVIE  = 000100   3123#
RCVRS   032012   3496#
REGS    010630    397#
RESRVD= 000010    81#  2573
RESTAR= 172004   100#
RESVEC= 000010    65#  2570*  2571*  2591*  2844*  2857*
ROLB0   012224   660#
ROLB1   017610   1640#
ROLO    013630   944#
ROL1    021560   1985#
ROMADR= 170000   103#  3079
ROMSIZ= 002000   104#  3080
RORB0   012302   677#   683    686    689    692
RORB1   017672   1655   1658#  1665   1668   1671   1674
RORO    013706   961#   967    970    973    976
ROR1    021646   2001   2004#  2011   2014   2017   2020
RVEC   = 001054   174#  3160*  3668
R6TST   024042   2391#
```

```
SAVTTY  036172          4267*  4270   4278#
SFCBO   012650           764#   768    771    777    780    784    788    791    796
SBCB1   020264          1750#  1755   1758   1764   1767   1771   1775   1778   1783
SBC0    014254          1048#  1052   1055   1061   1064   1068   1072   1075   1080
SBC1    022240          2096#  2101   2104   2110   2113   2117   2121   2124   2129
SEGN    036200           309*  2388*  2917*  3078*  3134*  3143*  3793*  4024   4249   4281#
SEG1    036202           309    4281   4282#
SEG2    036222          2388    4283#
SEG3    036245          2917    4284#
SEG4    036263          3078    4285#
SEG5    036301          3134    4286#
SEG6    036317          3143    4287#
SEG7    036335          3793    4288#
SENVC = 000273            86#  1092   2142
SEVC  = 000263            85#   816    862   1127   1534   1880   1896   2178
SKP1  = 000401            88#  2543   3898   3950
SKP2  = 000402            89#  2535   2975   2992   3244   4025
SKP3  = 000403            90#
SLU1  = 177560           121#  3135   3733
SLU1P = 000200           123#  3137
SLU1V = 000060           122#  3136
SLU2  = 176540           124#  3144   3583   3771
SLU2P = 000240           126#  3146
SLU2V = 000120           125#  3145
SOB1    023354          2278   2281#  2286   2290   2293   2296   2301
SPD0  = 001016           159#  3433   3473
SPD1  = 001020           160#
SPD2  = 001022           161#
SPD3  = 001024           162#
SPD4  = 001026           163#
SPD5  = 001030           164#
SPD6  = 001032           165#
SPD7  = 001034           166#  3469
STACK = 003776           185#   245    409    499    505    515    524   1900   2351   2353   2360   2509   2537
                        2552   2569   2581   2594   2606   2619   2631   2644   2674   2693   2707   2723   2737
                        2752   2796   2817   2833   2843   2860   2889   2926   2972   3021   3089   3151   3801
                        4061   4208
START = 172000            99#
SUB0    015004          1170   1173#  1177   1180   1186   1190   1193   1197
SUB1    023156          2244   2247#  2253   2256   2260   2265   2268   2272   2276   2279
SWAB0   014466          1097   1100#  1105   1108   1114   1117
SWAB1   022462          2147   2150#  2156   2159   2165   2168
SWR   = 000176            21#  4043   4177   4184   4186*  4197*  4244   4265   4271
SXT0    014414          1083#
SXT1    022404          2132#
SYNC    030212          3185#  3711
S1      010336           309#  4211
S2      024030          2388#  4211
S3      027052          2917#  4211
S4      027642          3078#  4212
S5      027764          3134#  3174   4212
S6      030014          3143#  3176   3773   4212
S7      033512          3772   3793#  4213
TEMP  = 001006           144#   437*   438*   439    484    489    492*   493    840   1239   1258   1265*  1270
                        1278*  1282   1288   1294*  1296*  1300   1305*  1309*  1313   1319*  1320   1335   1340
                        1346*  1347   1364   1372*  1373   1391   1397*  1409   1418   1423   1441   1460   1488
```

```
                      1506  1551  1596  1622  1786  1806  1845  1851  1913  2042  2067  2096  2196
                      2202  2215  2247  2249* 2262* 2266  2273* 2274* 2277  2305  2364* 2367* 2370
                      2373* 2375* 2378  2393* 2398* 2918* 2928  2977  2982  3004  3030  3079* 3091
TEMP1 = 001010         145#  459*  460   461*  465  1240  1259  1289  1321  1348  1374  1419  1431*
                      1434  1440* 1442  1449  1454  1461  1508  1532  1539  1597  1640  1653  1678
                      1695  1787  1805  1815  1828* 1838  1856  1876  1906  1942  1965  2132  2171*
                      2173* 2176* 2177  2189  2223* 2224  2248  2250* 2254  2257* 2261* 2306  2323*
                      2326  2919* 2929  2978  2983  3005  3031  3080* 3092
TEMP2 = 001012         146#  1241  1242* 1246  1252  1260  1290  1293* 1298  1304* 1311  1322  1324
                      1342  1349  1351  1366  1375  1393  1411  1417* 1420  1427  1443  1487  1696
                      1721  1750  1827* 1828  1842* 1893  1985  2004  2024  2041  2153  2195  2197*
                      2207  2210* 2218* 2239* 2240* 2243  2349  3050  3060
TEMP3 = 001014         147#
TN$   = 000147          32#  311#  314   318   334   345   358   370   384   387#  390   394   396#
                       411  414#  418   420   423   426#  430   432   435   442   445   448#  452
                       454  457   463   464   467   470#  473   476   478   482   487   491   495
                       498# 502   504   507   510   514   517   523#  528   531   534   537   539#
                       543  546   550   553   557   561   564#  570   573   576   579   582   585
                       587# 592   596   599   602   605   609   612   616   619   621#  626   629
                       633  636   641   659#  665   668   671   675   695#  701   704   707   710
                       735# 740   746   749   754   757   762   798#  802   805   809   812   315
                       819  821#  826   829   832   835   838#  845   848   851   854   857   861
                       865  868   871#  876   880   883   886   889   893   896   900   903   905#
                       910  913   917   920   925   943#  949   952   955   959   979#  985   988
                       991  994   1019# 1024  1030  1033  1038  1041  1046  1082# 1088  1091  1095
                      1098 1143# 1147  1150  1154  1157  1161  1166  1171  1199# 1204  1207  1209
                      1213 1216  1219  1225# 1231  1237  1248  1254  1257# 1272  1284  1287# 1302
                      1315 1318# 1326  1337  1344  1353  1368  1371# 1388  1395  1413  1416# 1429
                      1436 1439# 1451  1456  1459# 1465  1468  1471  1474  1480  1483  1486# 1493
                      1496 1499  1502  1505# 1514  1517  1520  1523  1527  1530  1538  1541  1544
                      1548 1550# 1556  1560  1567  1572  1575  1579  1582  1586  1589  1592  1595#
                      1603 1606  1610  1614  1620  1639# 1646  1649  1652  1656  1677# 1684  1687
                      1690 1693  1720# 1726  1732  1735  1740  1743  1748  1785# 1791  1794  1799
                      1802 1804# 1811  1814  1817  1820  1823  1826  1837  1840  1847  1850# 1861
                      1864 1867  1870  1875  1878  1883  1886  1892  1899  1902  1905# 1911  1916
                      1919 1922  1925  1929  1932  1936  1939  1941# 1947  1950  1954  1957  1963
                      1984# 1991 1994  1997  2002  2023# 2030  2033  2036  2039  2066# 2072  2078
                      2081 2086  2089  2094  2131# 2138  2141  2145  2148  2194# 2201  2204  2209
                      2214 2217  2222  2229  2234  2238  2242  2245  2304# 2311  2314  2318  2322
                      2325 2328  2331# 2337  2340  2345  2348  2355  2359  2363  2369  2372  2377
                      2380 2390# 2396  2401  2406  2412  2415  2421  2424  2427# 2435  2444  2453
                      2462 2471  2474# 2480  2483  2486  2489  2492  2495  2498  2501  2504# 2515
                      2521 2525  2529# 2546  2551  2554  2559  2562  2568# 2575  2580  2583  2586
                      2589 2593# 2600  2605  2608  2611  2614  2618# 2625  2630  2633  2636  2639
                      2643# 2666 2673  2676  2679  2682  2687# 2706  2709  2712  2715  2718# 2736
                      2739 2742  2745  2748# 2760  2765  2768  2771  2781  2786  2789  2792  2795#
                      2807 2812  2816# 2827  2832# 2837  2839  2842# 2851  2854  2856  2859# 2877
                      2882 2888# 2902  2907  2921# 2940  2955  2959  2962# 3017  3020# 3069  3082#
                      3108 3150# 3191# 3211  3214  3217  3220  3223# 3230  3233  3236  3239  3242
                      3250# 3261# 3271  3274  3277# 3285  3288  3291# 3299  3302  3305# 3313  3316
                      3319# 3327  3330  3333# 3341  3344  3347# 3357  3360  3363# 3370  3375  3379
                      3382# 3394  3400  3413  3418# 3472  3489  3497# 3505  3508  3511# 3533  3541
                      3544  3549  3554  3559  3562# 3638# 3654  3662  3665# 3679  3685  3698  3704#
                      3732# 3750  3795# 3820  3827# 3840  3843# 3853  3879# 3888  3892  3897  3906
                      3915  3923# 3930  3939  3944  3949  3958  3967  3971# 3986  3989  3996
TPASS = 004002         191#  4030* 4036
```

```
TPSW    015126          1200#
TPSW1   023454          2305#
TRAPHA  035336           294    4114#
TRAP4 = 167776           116#    248*
TRPVEC= 000034            71#    263*    290*    294*   2724    2725*   2726*   2733*
TRTVEC= 000014            67#   2753*   2754*   2775*   2797*   2798*   2818*   2819*   2830*
TSTB0   011422           524#
TSTB1   016604          1460#
TSTNUM  036176          4159*   4253    4280#
T1      010344           311#    314     318     334     345     358     370     384
T10     011316           498#    502     504     507     510     514     517
T100    025646          2687#   2706    2709    2712    2715
T101    025764          2718#   2736    2739    2742    2745
T102    026102          2748#   2760    2765    2768    2771    2781    2786    2789    2792
T103    026276          2795#   2807    2812
T104    026376          2816#   2827
T105    026460          2832#   2837    2839
T106    026526          2842#   2851    2854    2856
T107    026614          2859#   2877    2882
T11     011416           523#    528     531     534     537
T110    026742          2888#   2902    2907
T111    027074          2921#   2940    2955    2959
T112    027212          2962#   3017
T113    027402          3020#   3069
T114    027664          3082#   3108
T115    030042          3150#
T116    030236          3191#   3211    3214    3217    3220
T117    030410          3223#   3230    3233    3236    3239    3242
T12     011472           539#    543     546     550     553     557     561
T120    030524          3250#
T121    030562          3261#   3271    3274
T122    030654          3277#   3285    3288
T123    030736          3291#   3299    3302
T124    031020          3305#   3313    3316
T125    031102          3319#   3327    3330
T126    031164          3333#   3341    3344
T127    031246          3347#   3357    3360
T13     011604           564#    570     573     576     579     582     585
T130    031340          3363#   3370    3375    3379
T131    031432          3382#   3394    3400    3413
T132    031606          3418#   3472    3489
T133    032012          3497#   3505    3508
T134    032074          3511#   3533    3541    3544    3549    3554    3559
T135    032306          3562#
T136    032646          3638#   3654    3662
T137    032774          3665#   3679    3685    3698
T14     011710           587#    592     596     599     602     605     609     612     616     619
T140    033170          3704#
T141    033304          3732#   3750
T142    033520          3795#   3820
T143    033664          3827#   3840
T144    033720          3843#   3853
T145    034106          3879#   3888    3892    3897    3906    3915
T146    034310          3923#   3930    3939    3944    3949    3958    3967
T147    034522          3971#   3986    3989    3996
T15     012046           621#    626     629     633     636     641
```

T16	012220	659#	665	668	671	675						
T17	012360	695#	701	704	707	710						
T2	010600	387#	390	394								
T20	012532	735#	740	746	749	754	757	762				
T21	013004	798#	802	805	809	812	815	819				
T22	013100	821#	826	829	832	835						
T23	013156	838#	845	848	851	854	857	861	865	868		
T24	013314	871#	876	880	883	886	889	893	896	900	903	
T25	013452	905#	910	913	917	920	925					
T26	013624	943#	949	952	955	959						
T27	013764	979#	985	988	991	994						
T3	010624	396#	411									
T30	014136	1019#	1024	1030	1033	1038	1041	1046				
T31	014410	1082#	1088	1091	1095	1098						
T32	014652	1143#	1147	1150	1154	1157	1161	1166	1171			
T33	015122	1199#	1204	1207	1209	1213	1216	1219				
T34	015220	1225#	1231	1237	1248	1254						
T35	015360	1257#	1272	1284								
T36	015502	1287#	1302	1315								
T37	015660	1318#	1326	1337	1344	1353	1368					
T4	010706	414#	418	420	423							
T40	016106	1371#	1388	1395	1413							
T41	016316	1416#	1429	1436								
T42	016452	1439#	1451	1456								
T43	016600	1459#	1465	1468	1471	1474	1480	1483				
T44	016716	1486#	1493	1496	1499	1502						
T45	017002	1505#	1514	1517	1520	1523	1527	1530	1538	1541	1544	1548
T46	017210	1550#	1556	1560	1567	1572	1575	1579	1582	1586	1589	1592
T47	017410	1595#	1603	1606	1610	1614	1620					
T5	010746	426#	430	432	435	442	445					
T50	017604	1639#	1646	1649	1652	1656						
T51	017754	1677#	1684	1687	1690	1693						
T52	020142	1720#	1726	1732	1735	1740	1743	1748				
T53	020424	1785#	1791	1794	1799	1802						
T54	020512	1804#	1811	1814	1817	1820	1823	1826	1837	1840	1847	
T55	020740	1850#	1861	1864	1867	1870	1875	1878	1883	1886	1892	1899	1902
T56	021206	1905#	1911	1916	1919	1922	1925	1929	1932	1936	1939	
T57	021356	1941#	1947	1950	1954	1957	1963					
T6	011052	448#	452	454	457	463	464	467				
T60	021554	1984#	1991	1994	1997	2002						
T61	021730	2023#	2030	2033	2036	2039						
T62	022116	2066#	2072	2078	2081	2086	2089	2094				
T63	022400	2131#	2138	2141	2145	2148						
T64	022666	2194#	2201	2204	2209	2214	2217	2222	2229	2234	2238	2242	2245
T65	023450	2304#	2311	2314	2318	2322	2325	2328				
T66	023566	2331#	2337	2340	2345	2348	2355	2359	2363	2369	2372	2377	2380
T67	024036	2390#	2396	2401	2406	2412	2415	2421	2424			
T7	011156	470#	473	476	478	482	487	491	495			
T70	024210	2427#	2435	2444	2453	2462	2471					
T71	024466	2474#	2480	2483	2486	2489	2492	2495	2498	2501		
T72	024652	2504#	2515	2521	2525							
T73	024750	2529#	2546	2551	2554	2559	2562					
T74	025122	2568#	2575	2580	2583	2586	2589					
T75	025240	2593#	2600	2605	2608	2611	2614					
T7o	025356	2618#	2625	2630	2633	2636	2639					
T77	025474	2643#	2666	2673	2676	2679	2682					

```
VBIT    010462  347#
XBUF  = 001052  173#  3159*  3198*  3206   3227   3253*  3254   3371*  3436*  3439*  3535*  3577*  3626*
               3643*  3656*  3671*  3712*  3715*  3722*  3742*  3746*  3756*
XCSR  = 001050  172#  3157*  3166   3197*  3205   3207*  3226   3240   3263*  3264   3265*  3266   3267*
               3278*  3279   3280*  3281   3282*  3292*  3293   3294*  3295   3296*  3306*  3307   3308*
               3309   3310*  3320*  3321   3322*  3323   3324*  3334*  3335   3336*  3337   3338*  3349*
               3350   3351*  3352   3353*  3364*  3367   3372   3376   3383*  3387   3389*  3393*  3399*
               3412*  3415*  3435*  3437   3442   3455*  3519*  3520*  3521   3569*  3570*  3576*  3622*
               3639*  3640*  3644   3666*  3667*  3709*  3710*  3713   3736*  3766*
XMTBRK= 000001  3119#  3576   3622
XMTIE = 000100  3120#
XMTRDY= 000200  3121#  3373
XMTRS   030524  3249#
XOR0    014552  1116   1119#  1124   1131   1137   1140
XOR1    022554  2167   2170#  2175   2182   2188   2191
XVEC  = 001056  175#   3163*  3384
ZBIT    010546  372#
$LSTAD  036536  19     4324#
.       036534  218#   249    271    280    303#   314    318    334    345    358    370    384    390
               394    411    418    420    423    430    432    435    442    445    452    454    457
               463    464    467    473    476    477    478    482    487    491    495    502    504
               507    510    514    517    528    531    534    537    543    546    550    553    557
               561    570    573    576    579    582    585    592    596    599    602    605    609
               612    616    619    626    629    633    636    641    646    649    653    656    665
               668    671    675    683    686    689    692    701    704    707    710    718    721
               724    729    732    740    746    749    754    757    762    768    771    777    780
               784    788    791    796    802    805    809    812    815    819    826    829    832
               835    845    848    851    854    857    861    865    868    876    880    883    886
               889    893    896    900    903    910    913    917    920    925    930    933    937
               940    949    952    955    959    967    970    973    976    985    988    991    994
              1002   1005   1008   1013   1016   1024   1030   1033   1038   1041   1046   1052   1055
              1061   1064   1068   1072   1075   1080   1088   1091   1095   1098   1105   1108   1114
              1117   1124   1131   1137   1140   1147   1150   1154   1157   1161   1166   1171   1177
              1180   1185   1190   1193   1197   1204   1207   1209   1213   1216   1219   1231   1237
              1248   1254   1272   1284   1302   1315   1326   1337   1344   1353   1368   1388   1395
              1413   1429   1436   1451   1456   1465   1468   1471   1474   1480   1483   1493   1496
              1499   1502   1514   1517   1520   1523   1527   1530   1538   1541   1544   1548   1556
              1560   1567   1572   1575   1579   1582   1586   1589   1592   1603   1606   1610   1614
              1620   1626   1629   1633   1636   1646   1649   1652   1656   1665   1668   1671   1674
              1684   1687   1690   1693   1703   1706   1709   1714   1717   1726   1732   1735   1740
              1743   1748   1755   1758   1764   1767   1771   1775   1778   1783   1791   1794   1799
              1802   1811   1814   1817   1820   1823   1826   1837   1840   1847   1861   1864   1887
              1870   1875   1878   1883   1886   1892   1899   1902   1911   1916   1919   1922   1925
              1929   1932   1936   1939   1947   1950   1954   1957   1963   1971   1974   1978   1981
              1991   1994   1997   2002   2011   2014   2017   2020   2030   2033   2036   2039   2049
              2052   2055   2060   2063   2072   2078   2081   2086   2089   2094   2101   2104   2110
              2113   2117   2121   2124   2129   2138   2141   2145   2148   2156   2159   2165   2168
              2175   2182   2188   2191   2201   2204   2209   2214   2217   2222   2229   2234   2238
              2242   2245   2253   2256   2260   2265   2268   2272   2276   2279   2286   2290   2293
              2296   2301   2311   2314   2318   2322   2325   2328   2337   2340   2345   2348   2355
              2359   2363   2369   2372   2377   2380   2396   2401   2406   2412   2415   2421   2424
              2435   2444   2453   2462   2471   2480   2483   2486   2489   2492   2495   2498   2501
              2515   2521   2525   2546   2551   2554   2559   2562   2575   2580   2583   2586   2589
              2600   2605   2608   2611   2614   2625   2630   2633   2636   2639   2658   2664   2666
              2673   2676   2679   2682   2706   2709   2712   2715   2736   2739   2742   2745   2756
              2760   2765   2768   2771   2777   2781   2786   2789   2792   2802   2807   2812   2821
```

2827	2837	2839	2851	2854	2856	2877	2882	2900	2902	2906	2907	2940
2955	2959	2987	2999	3009	3017	3044	3047	3059	3062	3069	3102	3108
3174	3176	3178	3180	3211	3214	3217	3220	3230	3233	3236	3239	3242
3243	3257	3271	3274	3285	3288	3299	3302	3313	3316	3327	3330	3341
3344	3357	3360	3370	3375	3379	3394	3400	3413	3472	3489	3505	3508
3528	3533	3541	3544	3549	3554	3559	3578	3582	3591	3599	3602	3610
3615	3618	3624	3632	3635	3649	3654	3662	3679	3685	3698	3728	3750
3762	3820	3822	3824	3836	3840	3848	3853	3862	3867	3872	3875	3888
3892	3897	3906	3915	3930	3936	3939	3944	3949	3958	3967	3981	3986
3989	3996	4049	4067#	4322#	4324							

```
ERCALL    45#   314   318   334   345   358   370   384   390   394   411   418   420   423   430
           432   435   442   445   452   454   457   463   464   467   473   476   478   482   487
           491   495   502   504   507   510   514   517   528   531   534   537   543   546   550
           553   557   561   570   573   576   579   582   585   592   596   599   602   605   609
           612   616   619   626   629   633   636   641   646   649   653   656   663   668   671
           675   683   686   689   692   701   704   707   710   718   721   724   729   732   740
           746   749   754   757   762   768   771   777   780   784   788   791   796   802   805
           809   812   815   819   826   829   832   835   845   848   851   854   857   861   865
           868   876   880   883   886   889   893   896   900   903   910   913   917   920   925
           930   933   937   940   949   952   955   959   967   970   973   976   985   988   991
           994  1002  1005  1008  1013  1016  1024  1030  1033  1038  1041  1046  1052  1055  1061
          1064  1068  1072  1075  1080  1089  1091  1095  1098  1105  1108  1114  1117  1124  1131
          1137  1140  1147  1150  1154  1157  1161  1166  1171  1177  1180  1185  1190  1193  1197
          1204  1207  1209  1213  1216  1219  1231  1237  1248  1254  1272  1284  1302  1315  1326
          1337  1344  1353  1368  1388  1395  1413  1429  1436  1451  1456  1465  1468  1471  1474
          1480  1483  1493  1496  1499  1502  1514  1517  1520  1523  1527  1530  1538  1541  1544
          1548  1556  1560  1567  1572  1575  1579  1582  1586  1589  1592  1603  1606  1610  1614
          1620  1626  1629  1633  1636  1646  1649  1652  1656  1665  1668  1671  1674  1684  1687
          1690  1693  1703  1706  1709  1714  1717  1726  1732  1735  1740  1743  1748  1755  1758
          1764  1767  1771  1775  1778  1783  1791  1794  1799  1802  1811  1814  1817  1820  1823
          1826  1837  1840  1847  1861  1864  1867  1870  1875  1878  1883  1886  1892  1899  1902
          1911  1916  1919  1922  1925  1929  1932  1936  1939  1947  1950  1954  1957  1963  1971
          1974  1978  1981  1991  1994  1997  2002  2011  2014  2017  2020  2030  2033  2036  2039
          2049  2052  2055  2060  2063  2072  2078  2081  2086  2089  2094  2101  2104  2110  2113
          2117  2121  2124  2129  2138  2141  2145  2148  2156  2159  2165  2168  2175  2182  2188
          2191  2201  2204  2209  2214  2217  2222  2229  2234  2238  2242  2245  2253  2256  2260
          2265  2268  2272  2276  2279  2286  2290  2293  2296  2301  2311  2314  2318  2322  2325
          2328  2337  2340  2345  2348  2355  2359  2363  2369  2372  2377  2380  2396  2401  2406
          2412  2415  2421  2424  2435  2444  2453  2462  2471  2480  2483  2486  2489  2492  2495
          2498  2501  2515  2521  2525  2546  2551  2554  2559  2562  2575  2580  2583  2586  2589
          2600  2605  2608  2611  2614  2625  2630  2633  2636  2639  2666  2673  2676  2679  2682
          2706  2709  2712  2715  2736  2739  2742  2745  2760  2765  2768  2771  2781  2786  2789
          2792  2807  2812  2827  2837  2839  2851  2854  2856  2877  2882  2902  2907  2940  2955
          2959  2987  2999  3009  3017  3044  3047  3059  3062  3069  3102  3108  3174  3176  3178
          3180  3211  3214  3217  3220  3230  3233  3236  3239  3242  3257  3271  3274  3285  3288
          3299  3302  3313  3316  3327  3330  3341  3344  3357  3360  3370  3375  3379  3394  3400
          3413  3472  3489  3505  3508  3533  3541  3544  3549  3554  3559  3582  3591  3599  3602
          3610  3615  3618  3632  3635  3654  3662  3679  3685  3698  3728  3750  3762  3820  3822
          3824  3840  3853  3862  3867  3872  3875  3888  3892  3897  3906  3915  3930  3939  3944
          3949  3958  3967  3986  3989  3996

NEWTES    33#   311   387   396   414   426   448   470   498   523   539   564   587   621   659
           695   735   798   821   838   871   905   943   979  1019  1082  1143  1199  1225  1257
          1287  1318  1371  1416  1439  1459  1486  1505  1550  1595  1639  1677  1720  1785  1804
          1850  1905  1941  1984  2023  2066  2131  2194  2304  2331  2390  2427  2474  2504  2529
          2568  2593  2618  2643  2687  2718  2748  2795  2816  2832  2842  2859  2888  2921  2962
          3020  3082  3150  3191  3223  3250  3261  3277  3291  3305  3319  3333  3347  3363  3382
          3418  3497  3511  3562  3638  3665  3704  3732  3795  3827  3843  3879  3923  3971

. ABS.  036534     000     OVR   RO    ABS   GBL   I

 ERRORS DETECTED:  0

 CNKXAA,CNKXAA/LI:TOC/CR:=CNKXAA
 RUN-TIME: 16 21 1 SECONDS
```

```
B  1
C  1  INFORMATION
D  1  DOCUMENTS AND STANDARDS
E  1  NCE AND PROGRESS REPORTS
F  1   TEST DESCRIPTION
G  1  5 -- SERIAL LINE 1 TESTS (DEC
H  1    (FALCON) VERIFICATION TESTS (
I  1    (FALCON) VERIFICATION TESTS (
J  1    (FALCON) VERIFICATIO.I TESTS (
K  1    (FALCON) VERIFICATION TESTS (
L  1    (FALCON) VERIFICATION TESTS (
M  1    (FALCON) VERIFICATION TESTS (
N  1    (FALCON) VERIFICATION TESTS (
B  2    (FALCON) VERIFICATION TESTS (
C  2    (FALCON) VERIFICATION TESTS (
D  2    (FALCON) VERIFICATION TESTS (
E  2    (FALCON) VERIFICATION TESTS (
F  2    (FALCON) VERIFICATION TESTS (
G  2    (FALCON) VERIFICATION TESTS (
H  2    (FALCON) VERIFICATION TESTS (
I  2    (FALCON) VERIFICATION TESTS (
J  2    (FALCON) VERIFICATION TESTS (
K  2    (FALCON) VERIFICATION TESTS (
L  2    (FALCON) VERIFICATION TESTS (
M  2    (FALCON) VERIFICATION TESTS (
N  2    (FALCON) VERIFICATION TESTS (
B  3    (FALCON) VERIFICATION TESTS (
C  3    (FALCON) VERIFICATION TESTS (
D  3    (FALCON) VERIFICATION TESTS (
E  3    (FALCON) VERIFICATION TESTS (
F  3    (FALCON) VERIFICATION TESTS (
G  3    (FALCON) VERIFICATION TESTS (
H  3    (FALCON) VERIFICATION TESTS (
I  3    (FALCON) VERIFICATION TESTS (
J  3    (FALCON) VERIFICATION TESTS (
K  3    (FALCON) VERIFICATION TESTS (
L  3    (FALCON) VERIFICATION TESTS (
M  3    (FALCON) VERIFICATION TESTS (
N  3    (FALCON) VERIFICATION TESTS (
B  4    (FALCON) VERIFICATION TESTS (
C  4    (FALCON) VERIFICATION TESTS (
D  4    (FALCON) VERIFICATION TESTS (
E  4    (FALCON) VERIFICATION TESTS (
F  4    (FALCON) VERIFICATION TESTS (
G  4    (FALCON) VERIFICATION TESTS (
H  4    (FALCON) VERIFICATION TESTS (
I  4    (FALCON) VERIFICATION TESTS (
J  4    (FALCON) VERIFICATION TESTS (
K  4    (FALCON) VERIFICATION TESTS (
L  4    (FALCON) VERIFICATION TESTS (
M  4    (FALCON) VERIFICATION TESTS (
N  4    (FALCON) VERIFICATION TESTS (
B  5    (FALCON) VERIFICATION TESTS (
C  5    (FALCON) VERIFICATION TESTS (
D  5    (FALCON) VERIFICATION TESTS (
E  5    (FALCON) VERIFICATION TESTS (
F  5    (FALCON) VERIFICATION TESTS (
G  5    (FALCON) VERIFICATION TESTS (
H  5    (FALCON) VERIFICATION TESTS (
I  5    (FALCON) VERIFICATION TESTS (
J  5    (FALCON) VERIFICATION TESTS (
K  5    (FALCON) VERIFICATION TESTS (
L  5    (FALCON) VERIFICATION TESTS (
M  5    (FALCON) VERIFICATION TESTS (
N  5    (FALCON) VERIFICATION TESTS (
B  6    (FALCON) VERIFICATION TESTS (
C  6    (FALCON) VERIFICATION TESTS (
D  6    (FALCON) VERIFICATION TESTS (
E  6    (FALCON) VERIFICATION TESTS (
F  6    (FALCON) VERIFICATION TESTS (
G  6    (FALCON) VERIFICATION TESTS (
H  6    (FALCON) VERIFICATION TESTS (
I  6    (FALCON) VERIFICATION TESTS (
J  6    (FALCON) VERIFICATION TESTS (
K  6    (FALCON) VERIFICATION TESTS (
L  6    (FALCON) VERIFICATION TESTS (
M  6    (FALCON) VERIFICATION TESTS (
N  6    (FALCON) VERIFICATION TESTS (
B  7    (FALCON) VERIFICATION TESTS (
C  7    (FALCON) VERIFICATION TESTS (
D  7    (FALCON) VERIFICATION TESTS (
E  7    (FALCON) VERIFICATION TESTS (
F  7    (FALCON) VERIFICATION TESTS (
G  7    (FALCON) VERIFICATION TESTS (
H  7    (FALCON) VERIFICATION TESTS (
I  7    (FALCON) VERIFICATION TESTS (
J  7    (FALCON) VERIFICATION TESTS (
K  7    (FALCON) VERIFICATION TESTS (
L  7    (FALCON) VERIFICATION TESTS (
M  7    (FALCON) VERIFICATION TESTS (
N  7    (FALCON) VERIFICATION TESTS (
B  8    (FALCON) VERIFICATION TESTS (
C  8    (FALCON) VERIFICATION TESTS (
D  8    (FALCON) VERIFICATION TESTS (
E  8    (FALCON) VERIFICATION TESTS (
F  8    (FALCON) VERIFICATION TESTS (
G  8    (FALCON) VERIFICATION TESTS (
H  8    (FALCON) VERIFICATION TESTS (
I  8    (FALCON) VERIFICATION TESTS (
J  3    (FALCON) VERIFICATION TESTS (
K  8    (FALCON) VERIFICATION TESTS (
L  8    (FALCON) VERIFICATION TESTS (
M  8    (FALCON) VERIFICATION TESTS (
N  8    (FALCON) VERIFICATION TESTS (
B  9    (FALCON) VERIFICATION TESTS (
C  9    (FALCON) VERIFICATION TESTS (
D  9    (FALCON) VERIFICATION TESTS (
E  9    (FALCON) VERIFICATION TESTS (
F  9    (FALCON) VERIFICATION TESTS (
G  9    (FALCON) VERIFICATION TESTS (
H  9    (FALCON) VERIFICATION TESTS (
I  9    (FALCON) VERIFICATION TESTS (
J  9    (FALCON) VERIFICATION TESTS (
K  9    (FALCON) VERIFICATION TESTS (
L  9    (FALCON) VERIFICATION TESTS (
M  9    (FALCON) VERIFICATION TESTS (
N  9    (FALCON) VERIFICATION TESTS (
B 10    (FALCON) VERIFICATION TESTS (
C 10    (FALCON) VERIFICATION TESTS (
D 10    (FALCON) VERIFICATION TESTS (
E 10    (FALCON) VERIFICATION TESTS (
```