

KW11-L

LINE TIME CLOCK
MD-11-DZKWA-D

EP-DZKWA-D-DL-B

Copyright © 1975

FICHE 1 OF 1

SEP 1975

digital

Made In U.S.A.

DZKWA-D
SEQ

The grid contains 60 individual diagrams or data tables, organized in 10 rows and 6 columns. Each diagram is a small-scale technical drawing, likely representing a component of a larger system or a specific data set. The diagrams vary in complexity, with some showing simple flowcharts and others displaying more intricate schematic-like structures with numerous lines and text labels. The overall appearance is that of a technical manual or a series of data sheets for a specific piece of equipment.

11

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZKMA-D
PRODUCT NAME: LINE FREQUENCY CLOCK TEST
DATE REVISED: MAY 20, 1975
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: J RODENHISER/J LACEY/J COMEAU

COPYRIGHT (C) 1975
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DIGITAL EQUIPMENT CORPORATION.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

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

1.0 GENERAL PROGRAM INFORMATION

- 1.1 ABSTRACT
THIS PROGRAM TESTS THE K111L LINE FREQUENCY CLOCK. IT VALIDATES PROPER OPERATION UNDER BOTH INTERRUPT AND NON-INTERRUPT MODES.
- 1.2 SYSTEM REQUIREMENTS
THIS PROGRAM IS DESIGNED TO RUN ON ANY PDP-11 WITH 4K OF MEMORY AND A K111 LINE FREQUENCY CLOCK.

2.0 OPERATING INSTRUCTIONS

- 2.1 LOADING
PROCEDURE FOR NORMAL BINARY TAPES SHOULD BE FOLLOWED
 - A ABSOLUTE LOADER PROGRAM MUST BE IN MEMORY
 - B PLACE THE BINARY TAPE IN THE PAPER TAPE READER
 - C LOAD ADDRESS 17500
 - D DEPRESS START (TAPE SHOULD READ IN)
- 2.2 STARTING
PROGRAM STARTING ADDRESS IS 000200
 - A LOAD ADDRESS 000200
 - B SELECT SWITCH REGISTER OPTIONS (SEE SECTION 2.3)
 - C DEPRESS START (PROGRAM SHOULD START RUNNING)
- 2.3 SWITCH REGISTER OPTIONS
HERE IS A LIST OF CONSOLE SWITCHES AND THEIR EFFECT ON THE PROGRAM...

SWITCH	ACTION IF SET
15	HALT ON ERROR
14	LOOP ON CURRENTLY EXECUTING TEST
13	INHIBIT ERROR PRINTOUTS
12	(UNUSED)
11	INHIBIT ITERATIONS
10	BELL ON ERROR
9	LOOP ON ERROR
8	LOOP ON TEST SPECIFIED IN SWR(7:0)
7-0	8 OF TEST TO LOOP ON (ONLY WHEN SWRB = 1)

- 2.4 EXECUTION TIMES
 EXECUTION TIME FOR THIS PROGRAM IS DEPENDENT ON THE MODEL OF PDP-11 IT IS BEING RUN ON. FOR A PDP-11/40 ABOUT 5 SECONDS IS NECESSARY TO DO 1 PASS OF THE PROGRAM WITHOUT ITERATIONS.

3.0 ERROR INFORMATION

- 3.1 STANDARD ERROR REPORTING PROCEDURES
 ERROR PRINTOUTS CONSIST OF FROM 4 TO 8 COLUMNS OF DATA, A DATA HEADER, AND POSSIBLY A SHORT ERROR MESSAGE DESCRIBING THE ERROR. FOR EXAMPLE...

```
CLOCK FAILED TO INTERRUPT
PC      PS      SP      TEST#  LKS
002262 000344 000764 000007 000300
```

THE FIRST 4 COLUMNS OF OF THE ERROR MESSAGE ALWAYS SHOW THE CONTENTS OF THE PC, PS, SP, AND THE TEST NUMBER. MORE COLUMNS OF DATA ARE ADDED WHERE THEY MIGHT BE RELEVANT TO A PARTICULAR ERROR.

- 3.2 UNEXPECTED TRAP ERROR REPORTING
 AN UNEXPECTED TRAP TO ADDRESS 4 CAUSES THE FOLLOWING MESSAGE TO BE PRINTED OUT...

```
TRAPPED TO LOC 4 FROM LOCATION "XXXXXX"
RESTARTING PROGRAM
```

IN THE ACTUAL MESSAGE THE "XXXXXX" IS REPLACED BY THE PC ADDRESS PUSHED ONTO THE STACK WHEN THE UNEXPECTED TRAP OCCURS. THE PROGRAM THEN TRYs TO RESTART ITSELF DESPITE SWITCH REGISTER SETTINGS.

- 3.3 POWER FAIL
 IF A POWER FAIL CONDITION IS DETECTED THE FOLLOWING MESSAGE IS PRINTED...

```
POWER
```

```
AFTER PRINTING OUT THE MESSAGE THE PROGRAM TRYs TO
RESTART ITSELF.
```

5.0 DEVICE INFORMATION

- 5.1 GENERAL INFORMATION

THE LINE CLOCK INTERRUPT VECTOR ADDRESS IS 100
THE LINE CLOCK PRIORITY LEVEL IS BR6

5.2 REGISTERS

LINE CLOCK STATUS REGISTER (LKS) 777546

!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!
!	!	!	!	!	!	!	!	7	6	!	!	!	!	!	!	!

BIT6 IF SET MONITOR=1 CAUSES AN INTERRUPT
BIT7 MONITOR BIT. SET BY CLOCK, CLEARED BY USER

7.0 FLOW CHARTS

FLOW CHART

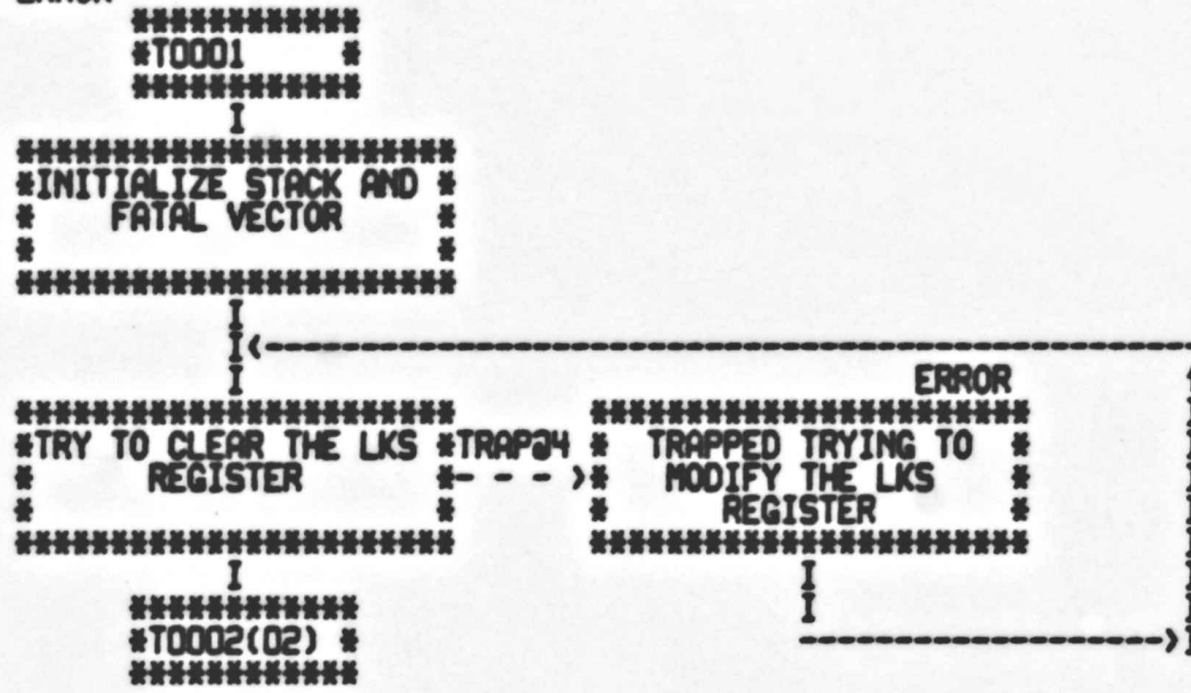
LINE CLOCK PROGRAM FLOW CHART

COPYRIGHT 1975
DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASS. 01754

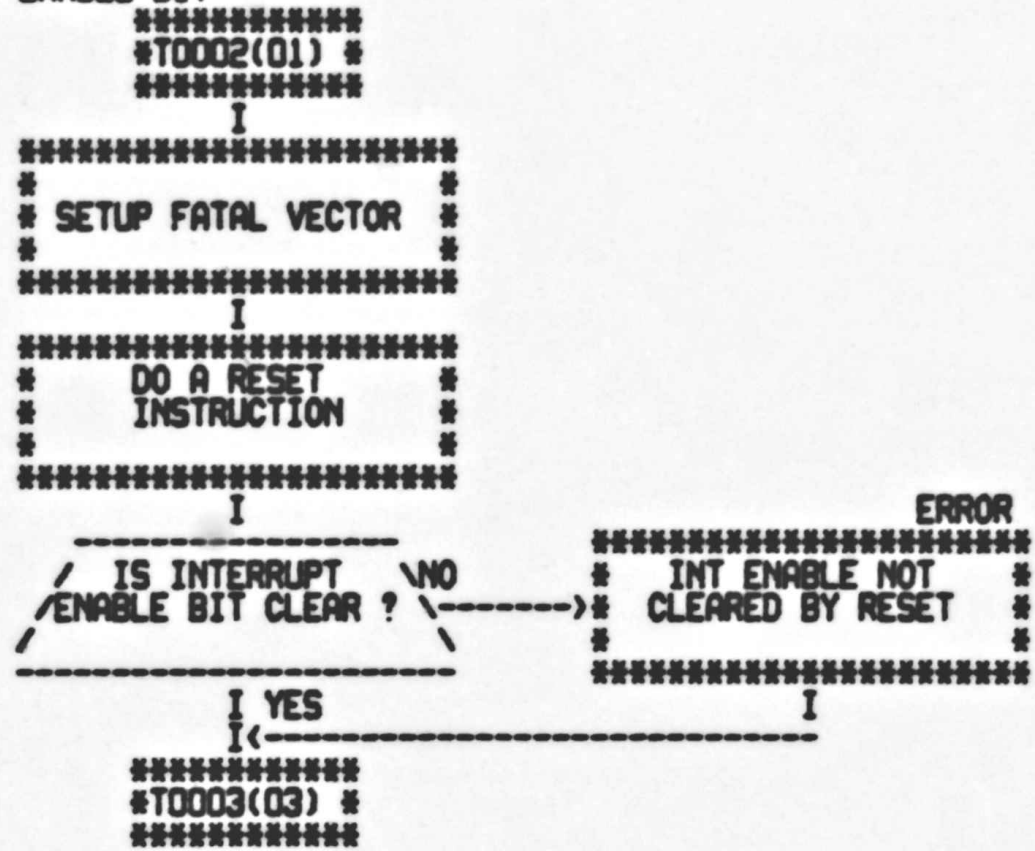
TABLE OF CONTENTS

PAGE 01	TEST THAT THE LKS CAN BE ACCESSED WITHOUT A BUS ERROR
PAGE 02	TEST THAT RESET CLEARS THE LINE CLOCK INTERRUPT ENABLE BIT
PAGE 03	TEST THAT INIT SETS CLOCK FLAG BIT
PAGE 04	TEST THAT CLOCK FLAG WILL SET BY ITSELF WITHIN A CERTAIN PERIOD OF TIME
PAGE 05	TEST THAT INTERRUPT ENABLE BIT MAY BE SET
PAGE 06	TEST THAT INTERRUPT ENABLE BIT MAY BE CLEARED
PAGE 07	TEST THAT CLOCK INTERRUPTS USING THE CORRECT VECTOR
PAGE 08	TEST THAT CLOCK WILL INTERRUPT WITH PS AT LEVEL 5
PAGE 09	TEST THAT CLOCK WILL NOT INTERRUPT WITH PROCESSOR AT PRIORITY 4
PAGE 10	TEST THAT RESET SETS THE CLOCK FLAG
PAGE 11	TEST CLOCK REPEATABILITY OVER 2 EQUAL TIME PERIODS
PAGE 12	LKS DUAL ADDRESSING TESTS T0014-T0021
PAGE 13	LKS REGISTER HIGH BYTE TEST
PAGE 14	CLOCK FLAG BIT TEST
PAGE 15	INTERRUPT TEST
PAGE 16	"NO SACK" TIMEOUT TEST
PAGE 17	RESET TEST
PAGE 18	MAKE SURE THAT THE CLOCK FLAG BIT SETS OK
PAGE 19	MULTIPLE INTERRUPT TEST
PAGE 20	NO INTERRUPT AT PRIO LEVEL 7 TEST
PAGE 21	CC PUSH TEST
PAGE 22	PC PUSH TEST

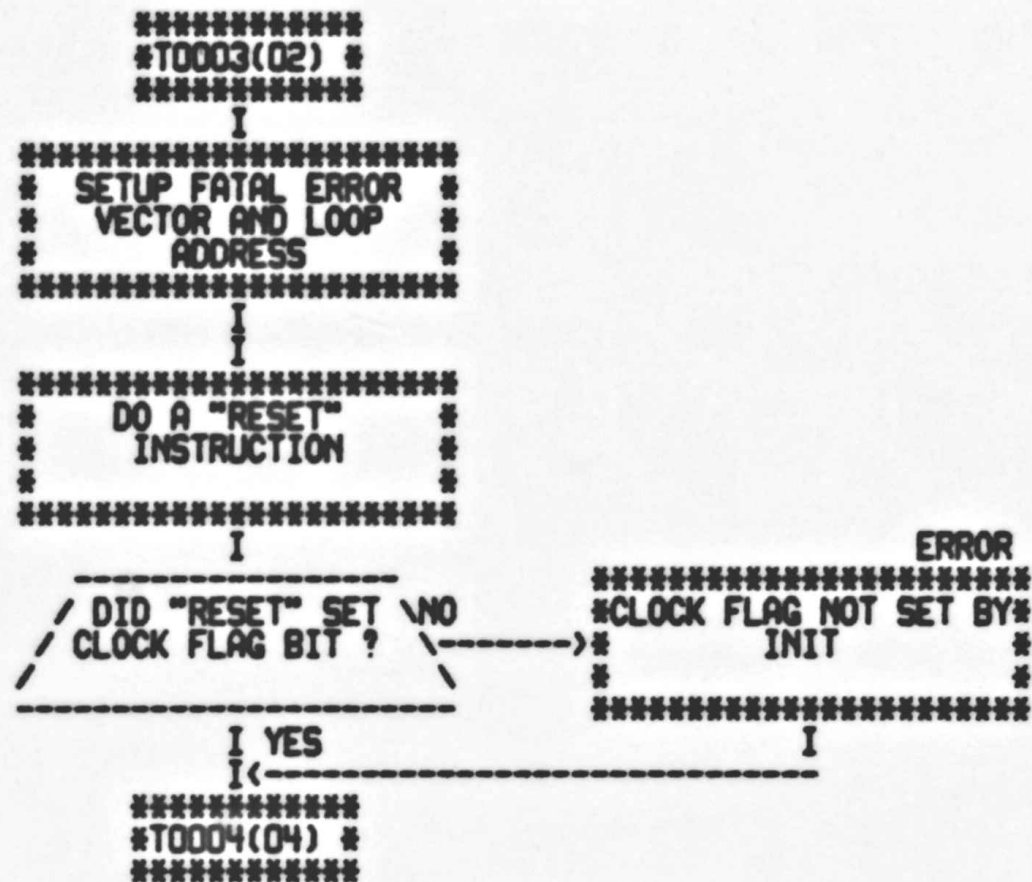
LINE CLOCK PROGRAM FLOW CHART
TEST THAT THE LKS CAN BE ACCESSED WITHOUT A BUS ERROR



LINE CLOCK PROGRAM FLOW CHART
TEST THAT RESET CLEARS THE LINE CLOCK INTERRUPT ENABLE BIT



LINE CLOCK PROGRAM FLOW CHART
TEST THAT INIT SETS CLOCK FLAG BIT



#T0004(03) #

I

SETUP FATAL VECTOR #

I

WAIT FOR CLOCK FLAG ##
TO SET ##

I

/ DID CLOCK FLAG \ NO
SET IN TIME ? >

ERROR

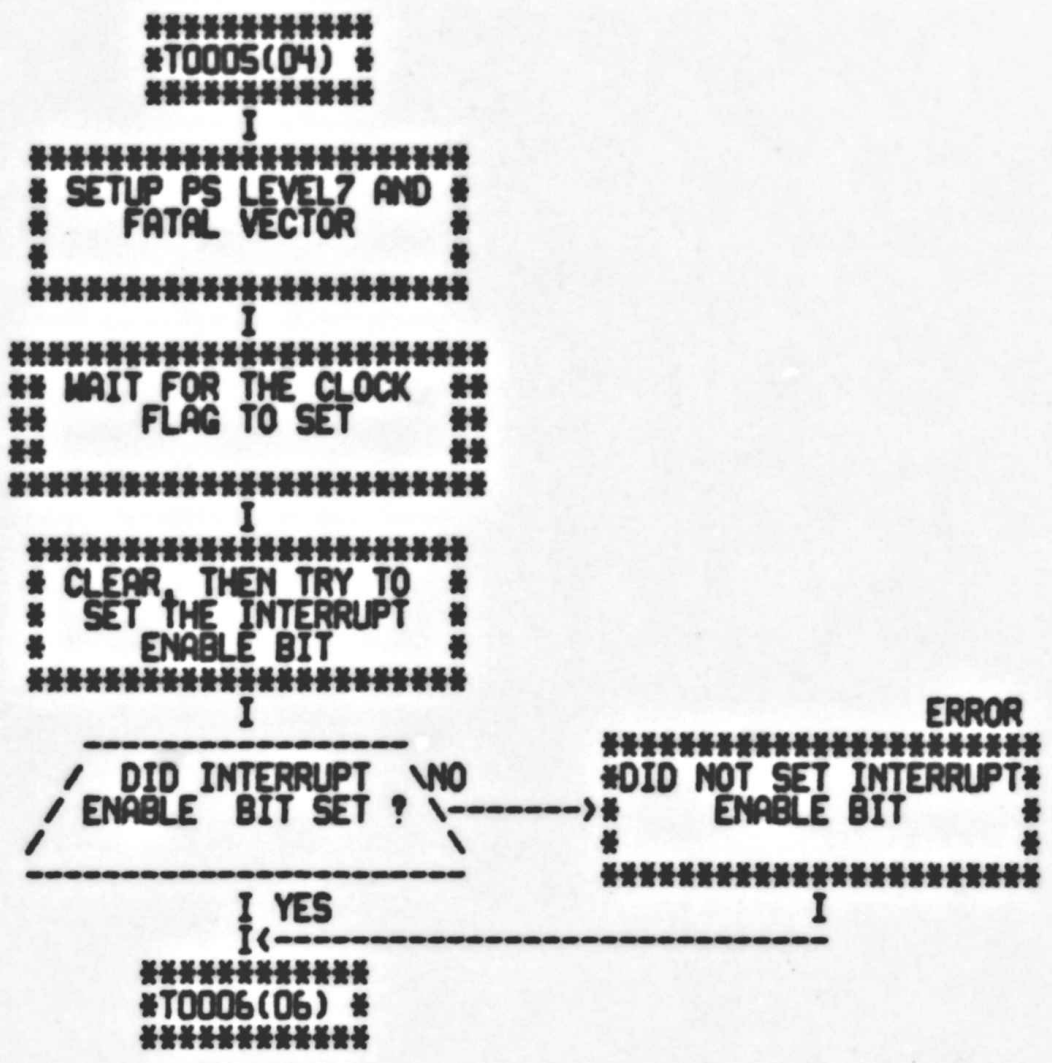
CLOCK FLAG DID NOT #
SET #

I YES

I

#T0005(05) #

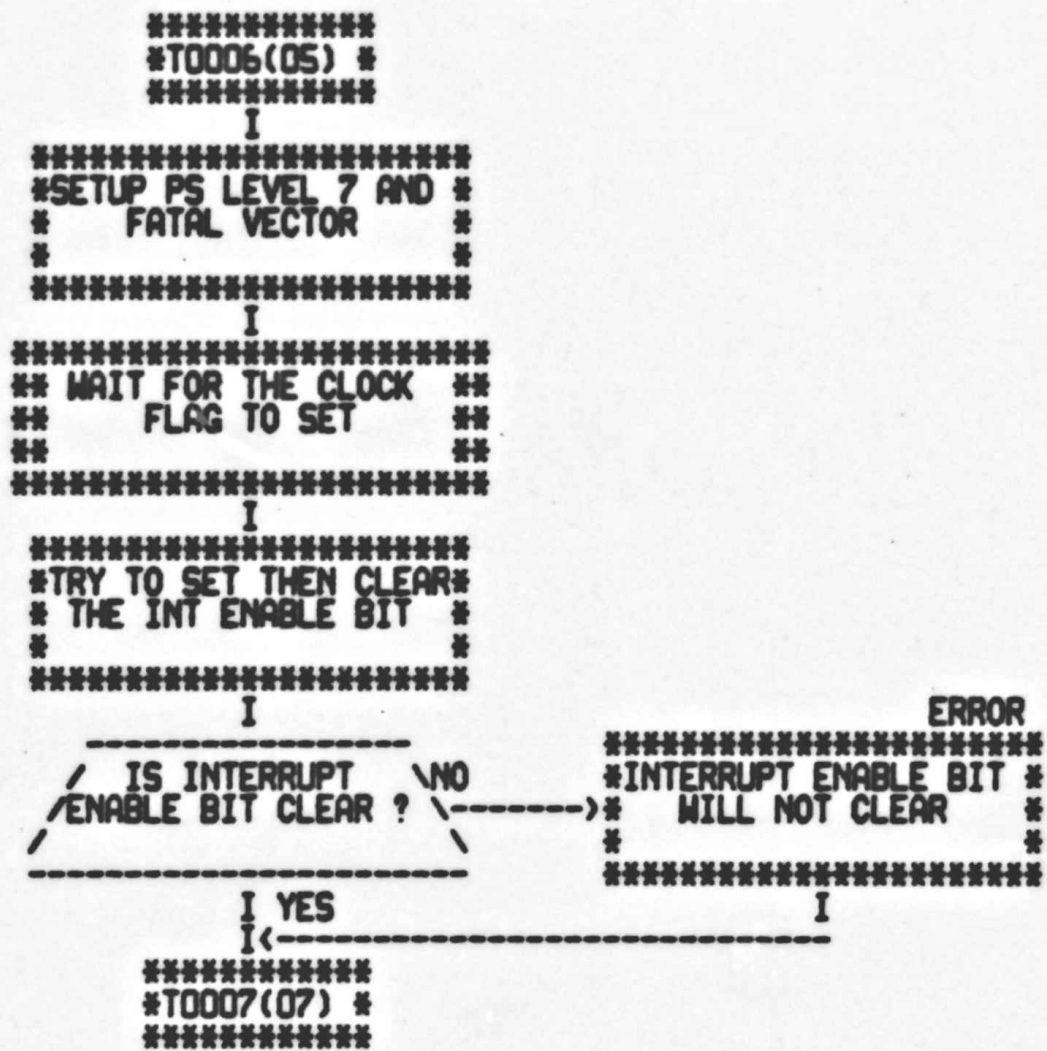
LINE CLOCK PROGRAM FLOW CHART
TEST THAT INTERRUPT ENABLE BIT MAY BE SET



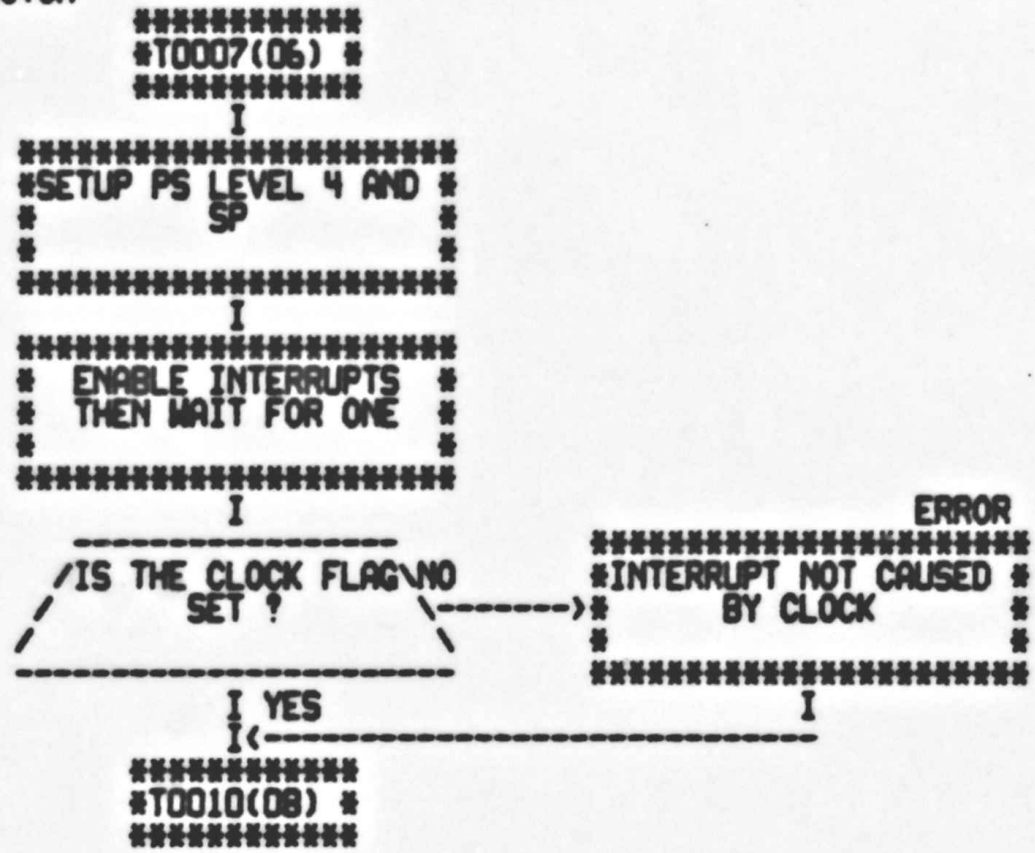
ERROR

#DID NOT SET INTERRUPT#
ENABLE BIT #

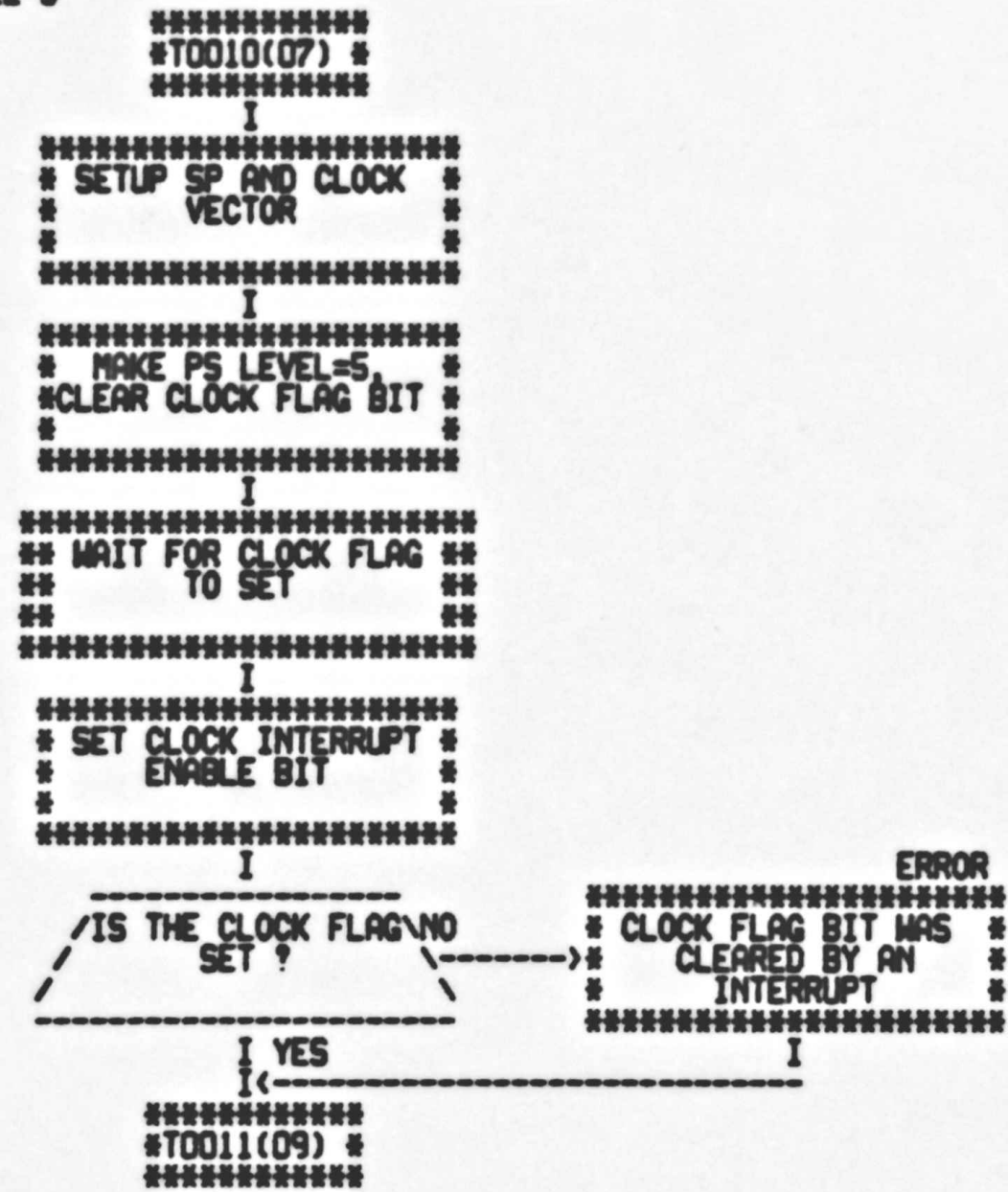
LINE CLOCK PROGRAM FLOW CHART
TEST THAT INTERRUPT ENABLE BIT MAY BE CLEARED



LINE CLOCK PROGRAM FLOW CHART
TEST THAT CLOCK INTERRUPTS USING THE CORRECT VECTOR



LINE CLOCK PROGRAM FLOW CHART
TEST THAT CLOCK WILL INTERRUPT WITH PS AT LEVEL 5



LINE CLOCK PROGRAM FLOW CHART
TEST THAT CLOCK WILL NOT INTERRUPT WITH PROCESSOR AT PRIORITY 4

#T0011(08) #

I

SETUP FATAL TRAP #
VECTOR #

I

WAIT FOR CLOCK FLAG #
TO SET #

I

SETUP SP, SET PS #
LEVEL TO 6 #

I

ENABLE CLOCK #
INTERRUPTS #

I

MAKE SURE THAT THE #
CLOCK FLAG IS STILL #
SET #

I

/ DID ME GET AN \ YES
/ INTERRUPT ? \

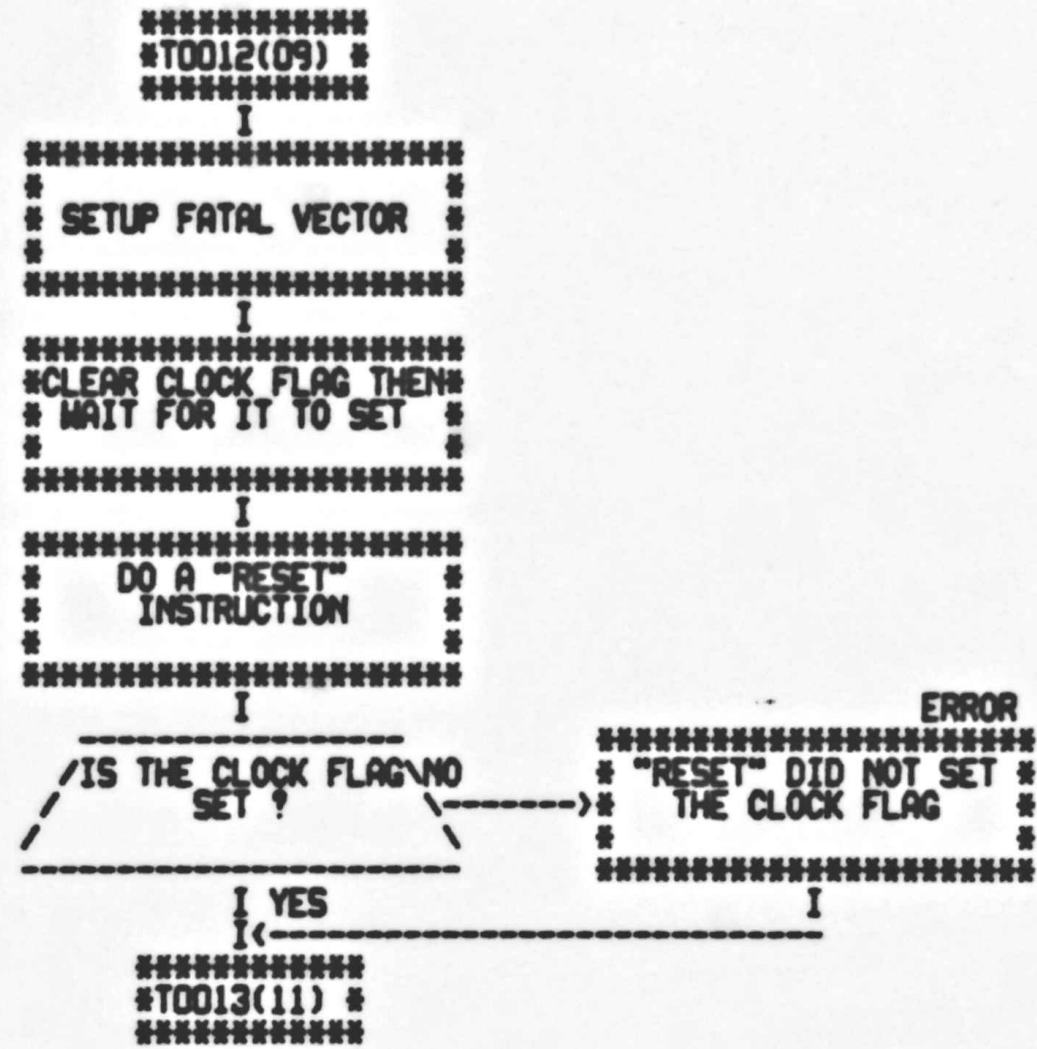
***** ERROR *****
CLOCK INTERRUPTED #
WITHOUT A HIGH ENOUGH #
PRIORITY #

I NO

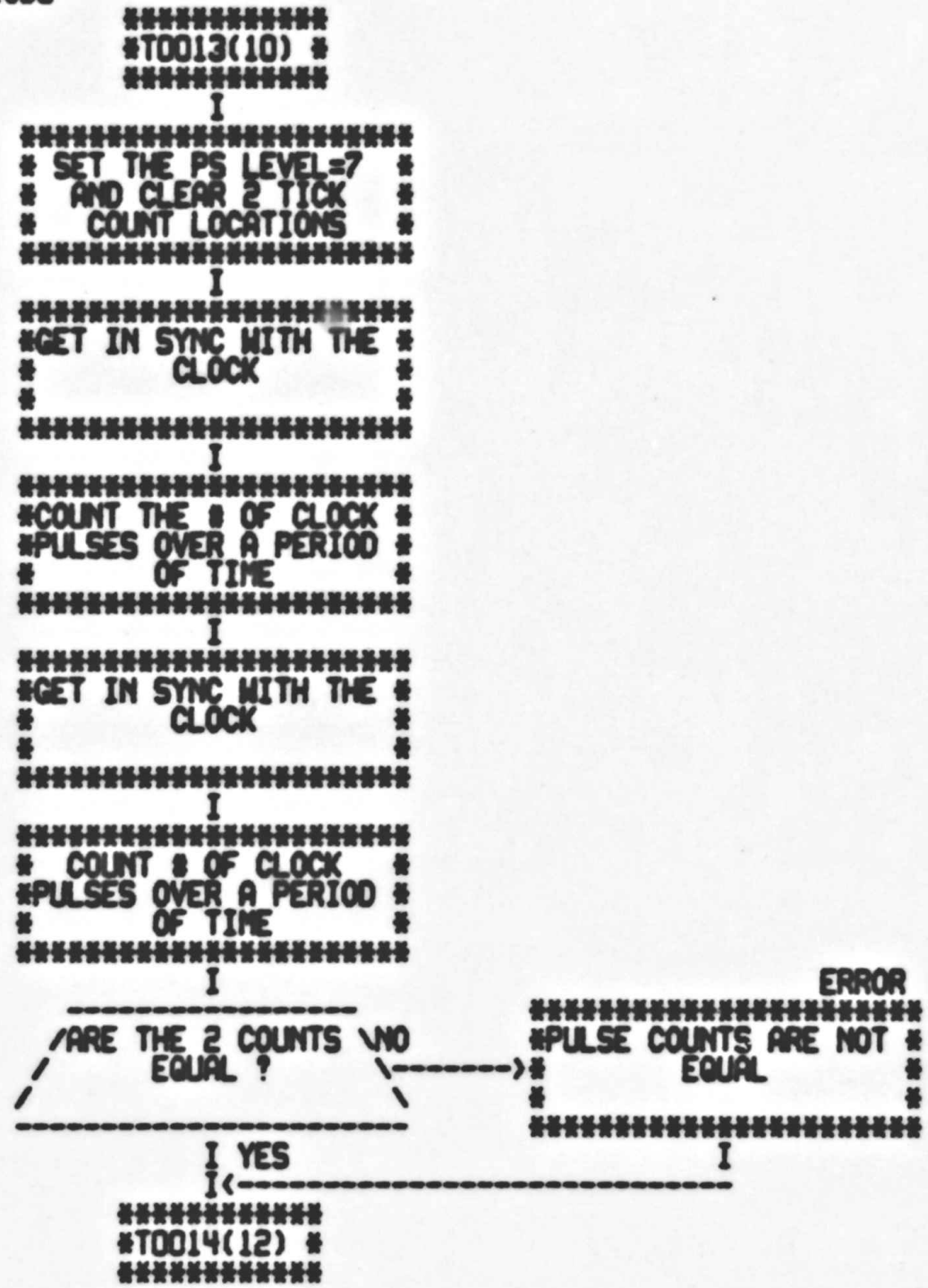
I <

#T0012(10) #

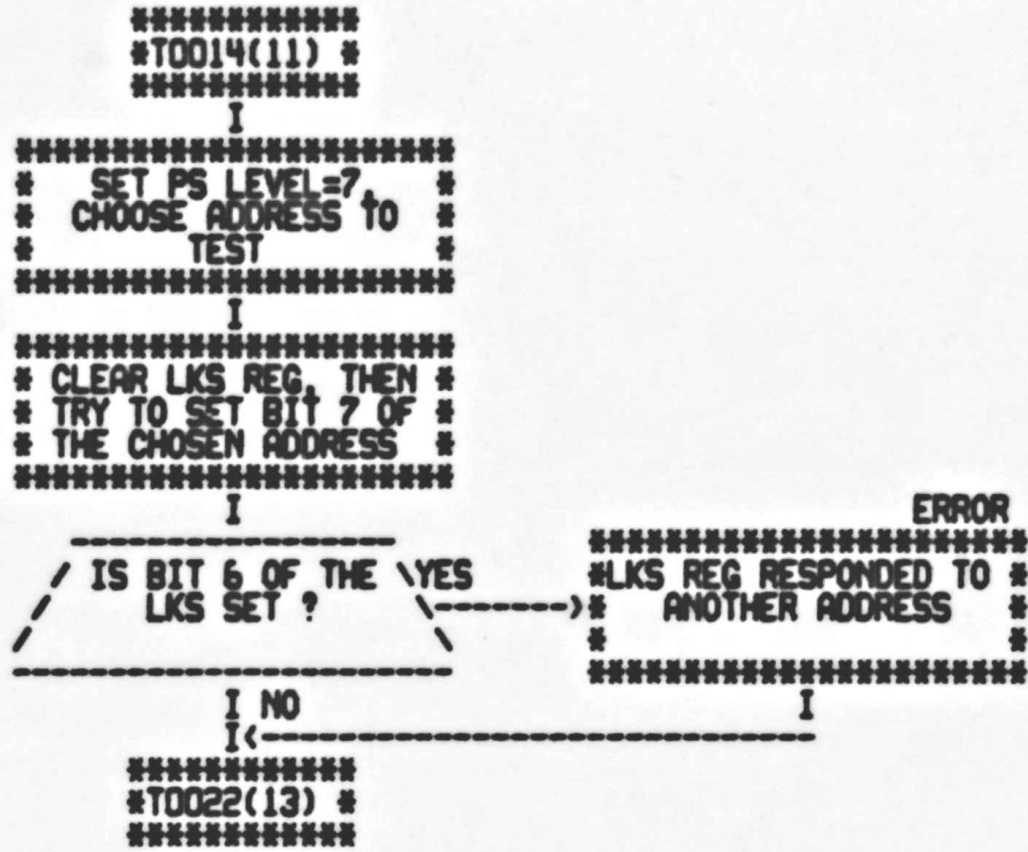
LINE CLOCK PROGRAM FLOW CHART
TEST THAT RESET SETS THE CLOCK FLAG

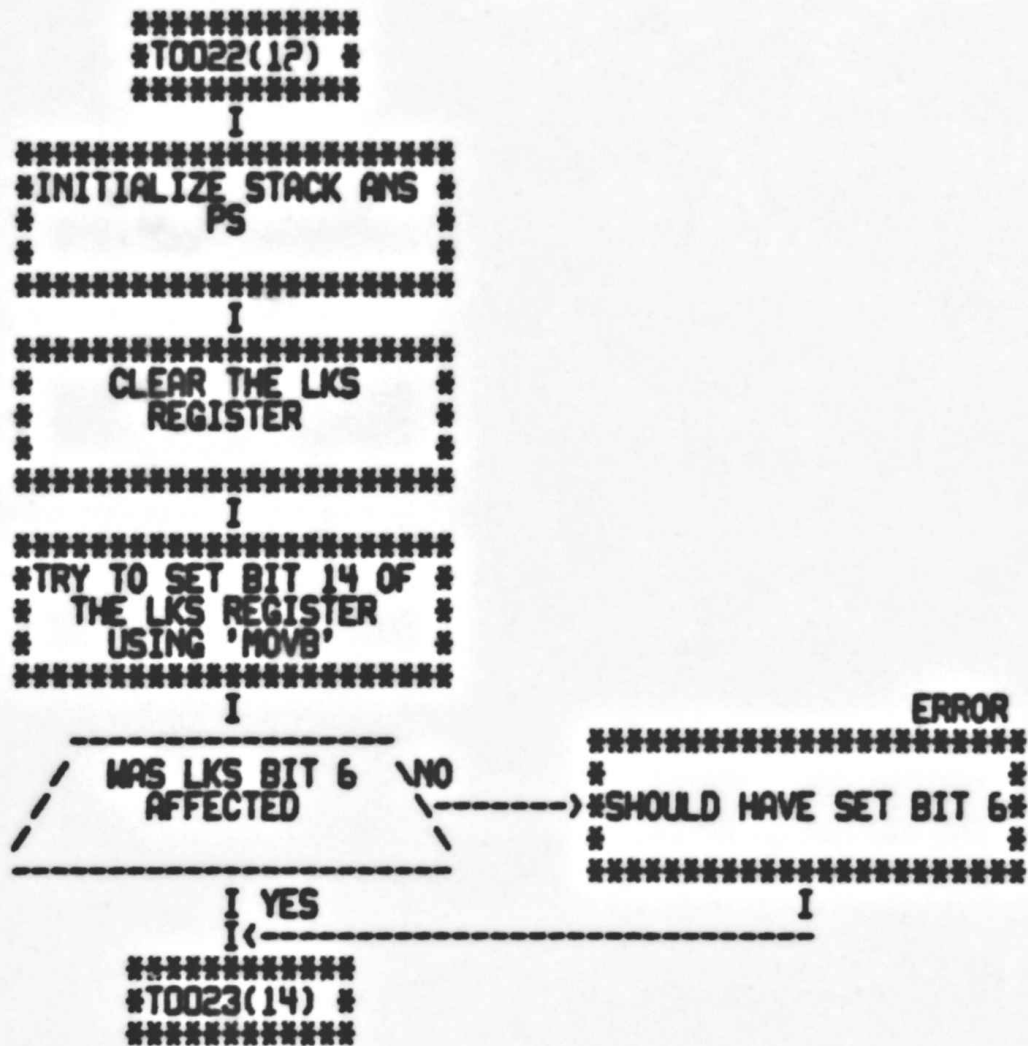


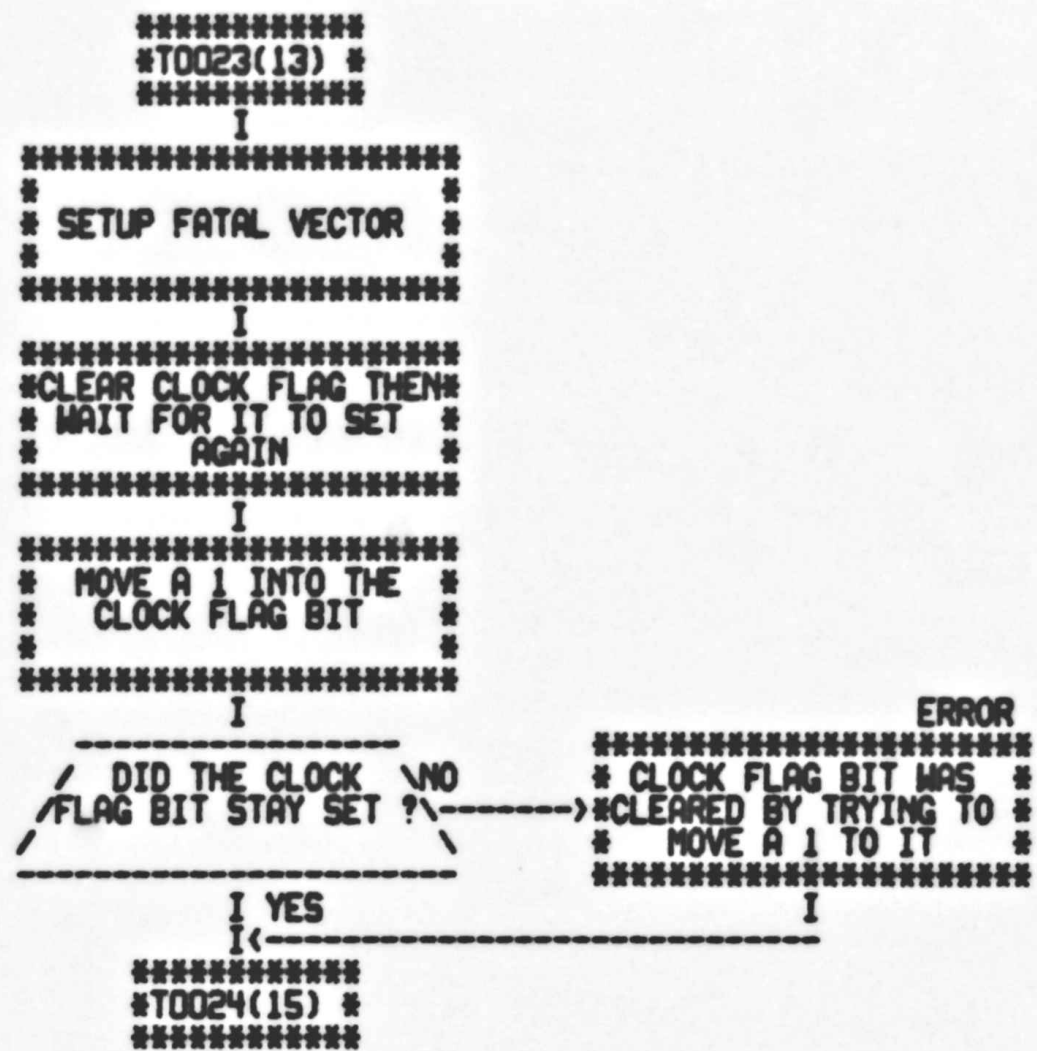
LINE CLOCK PROGRAM FLOW CHART
TEST CLOCK REPEATABILITY OVER 2 EQUAL TIME PERIODS



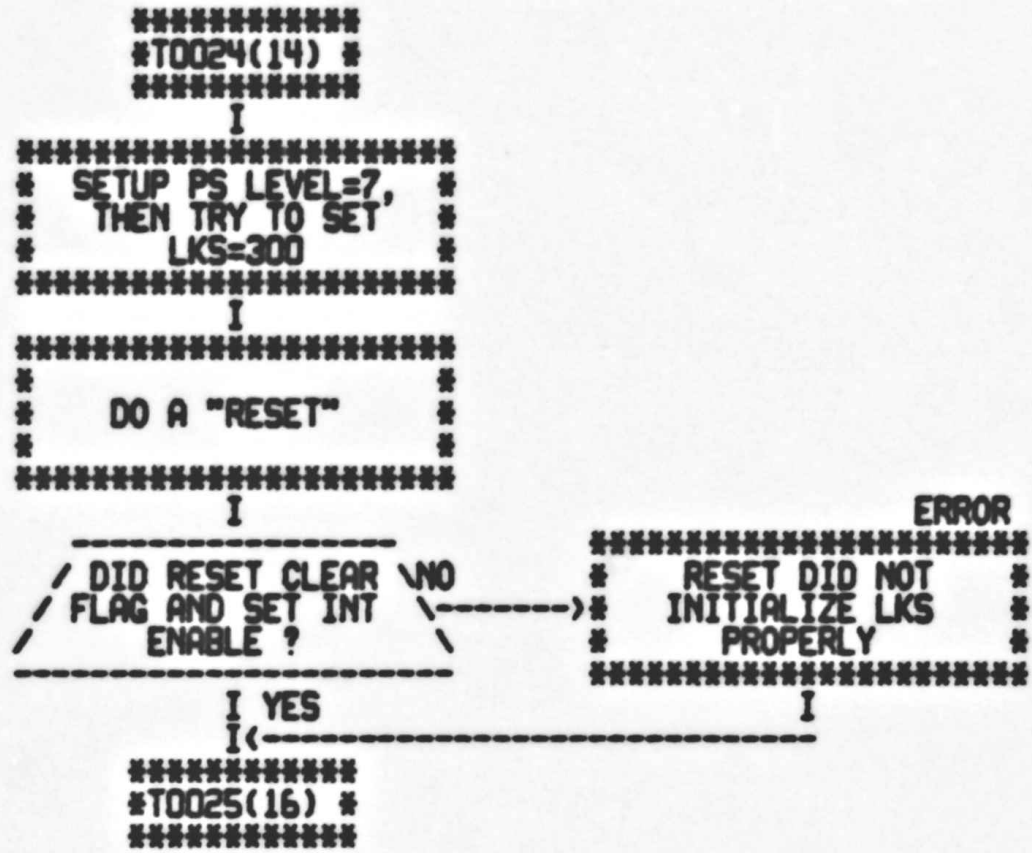
LINE CLOCK PROGRAM FLOW CHART
LKS DUAL ADDRESSING TESTS T0014-T0021



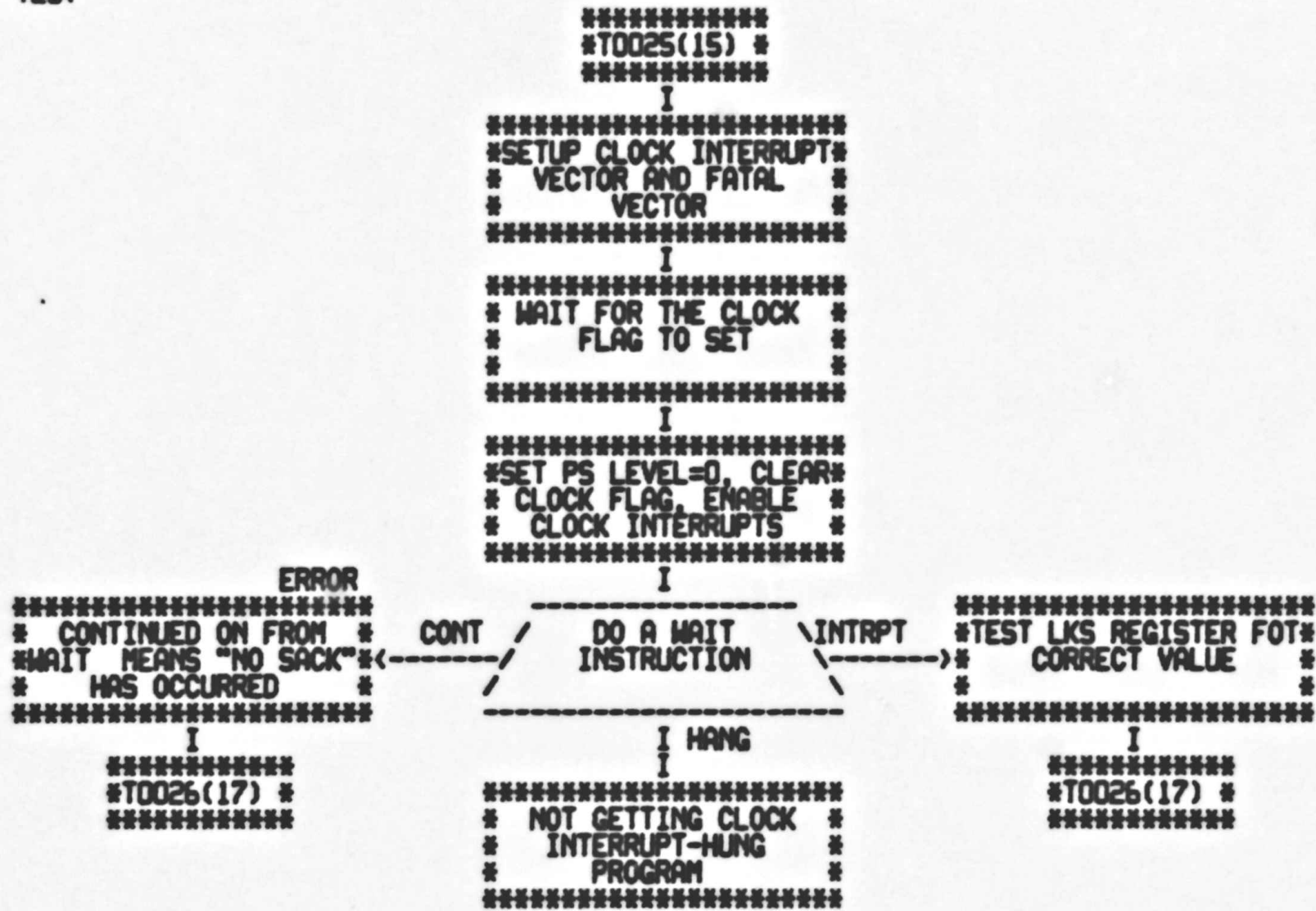




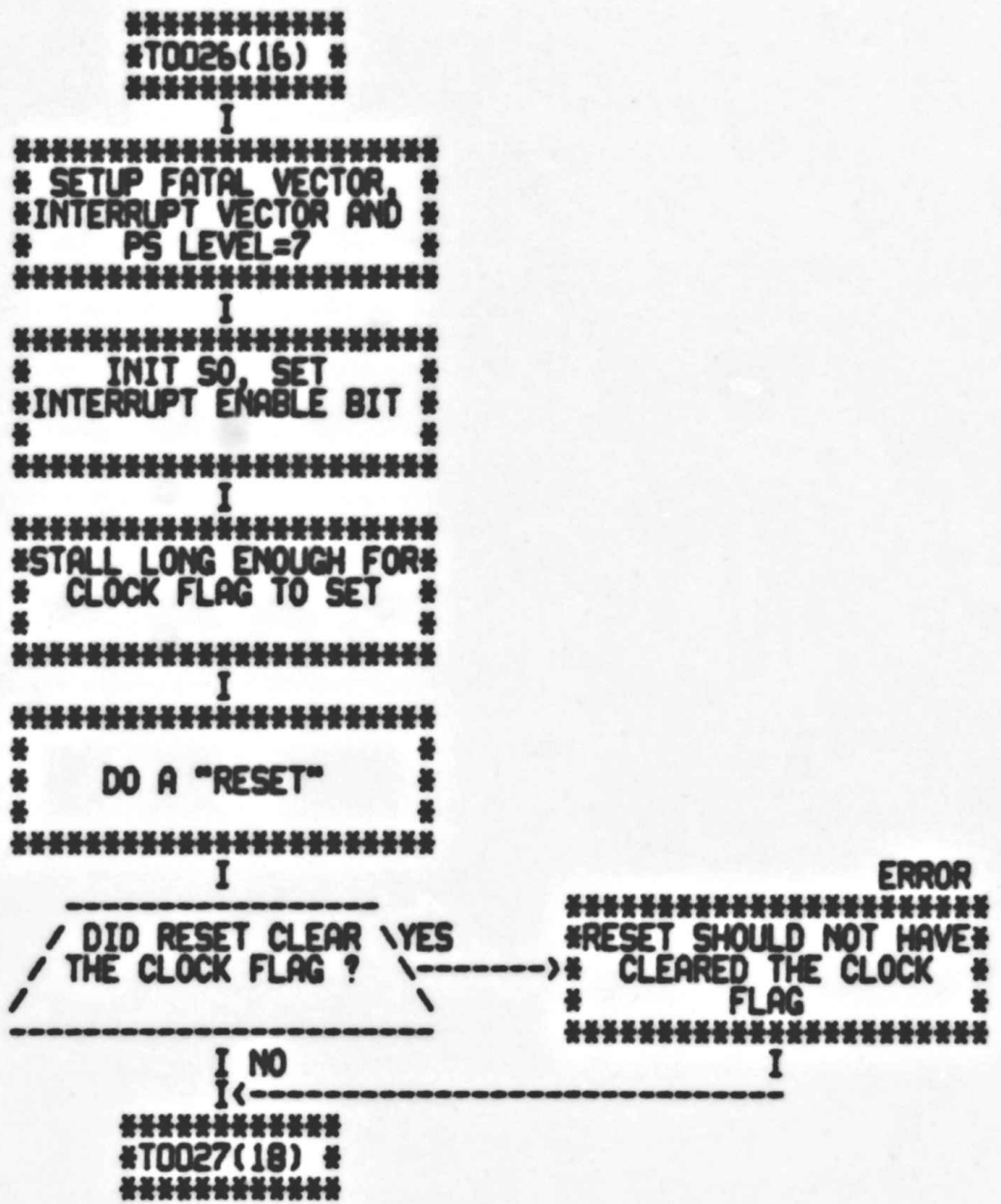
LINE CLOCK PROGRAM FLOW CHART
INTERRUPT TEST



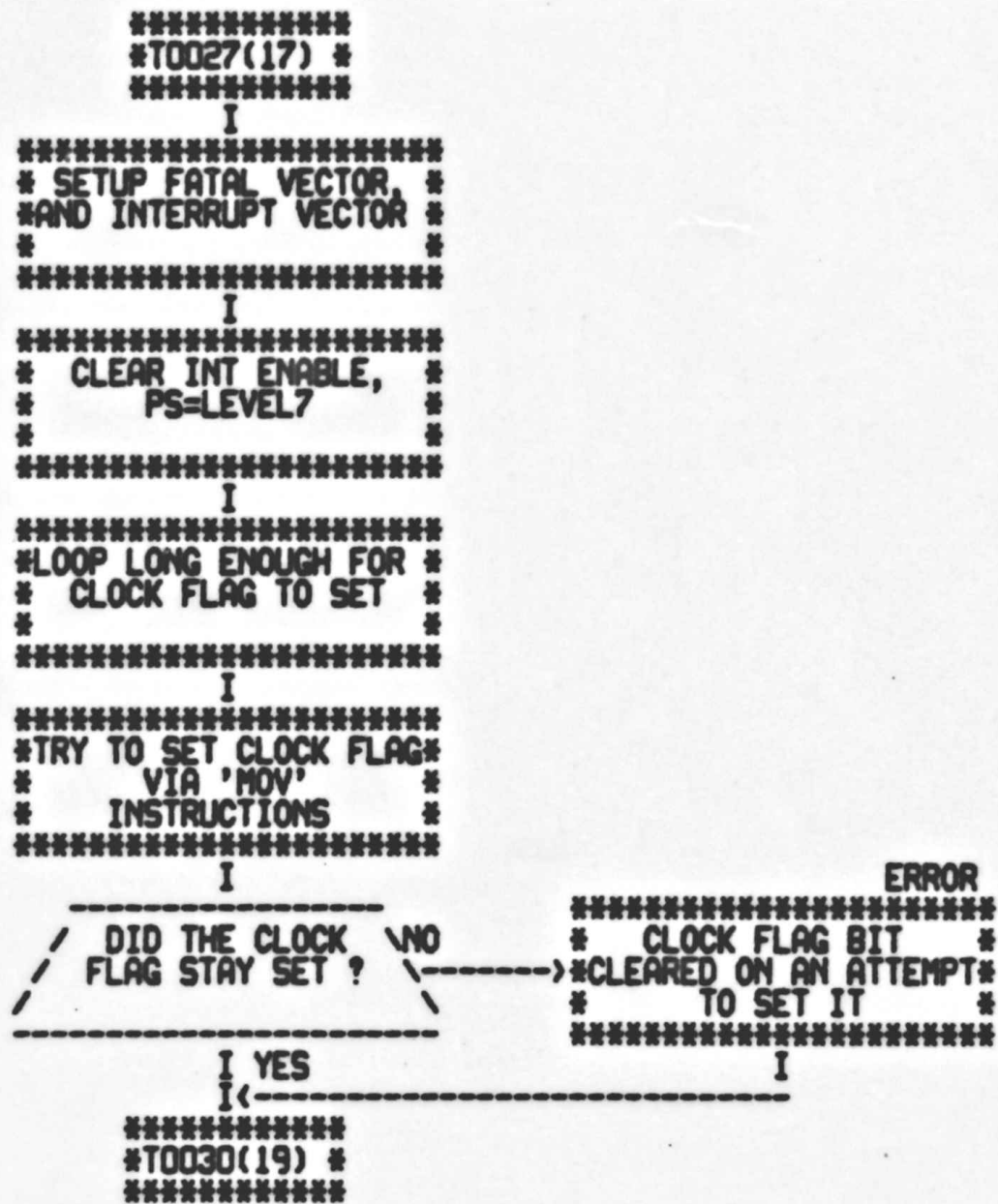
LINE CLOCK PROGRAM FLOW CHART
"NO SACK" TIMEOUT TEST



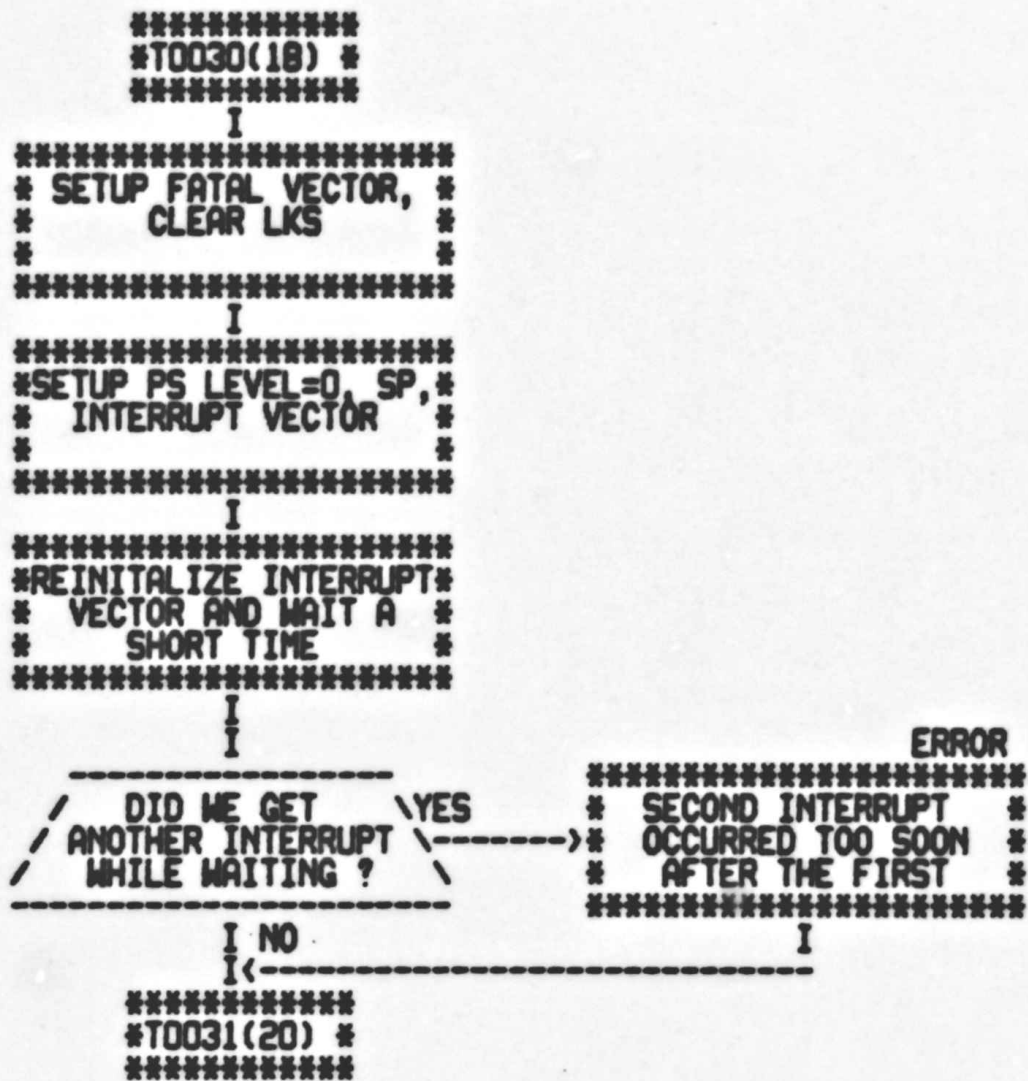
LINE CLOCK PROGRAM FLOW CHART
RESET TEST



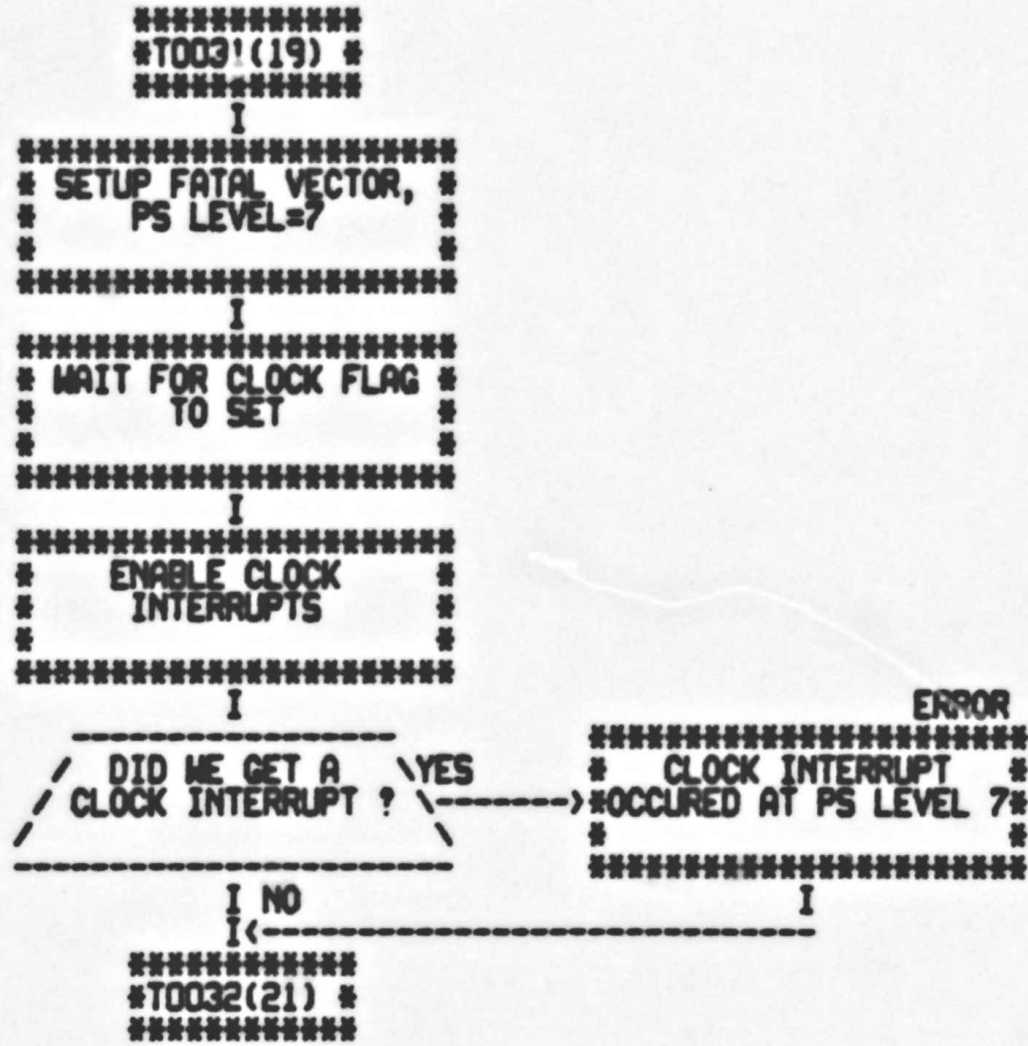
LINE CLOCK PROGRAM FLOW CHART
MAKE SURE THAT THE CLOCK FLAG BIT SETS OK



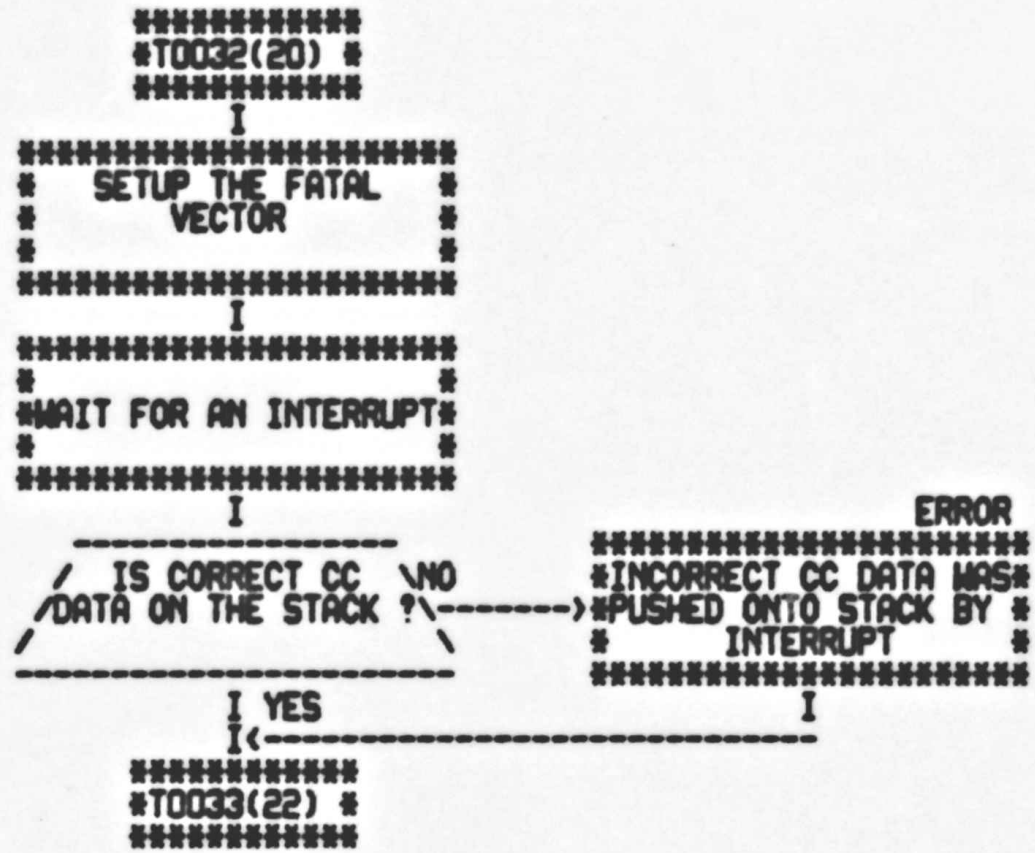
LINE CLOCK PROGRAM FLOW CHART
MULTIPLE INTERRUPT TEST

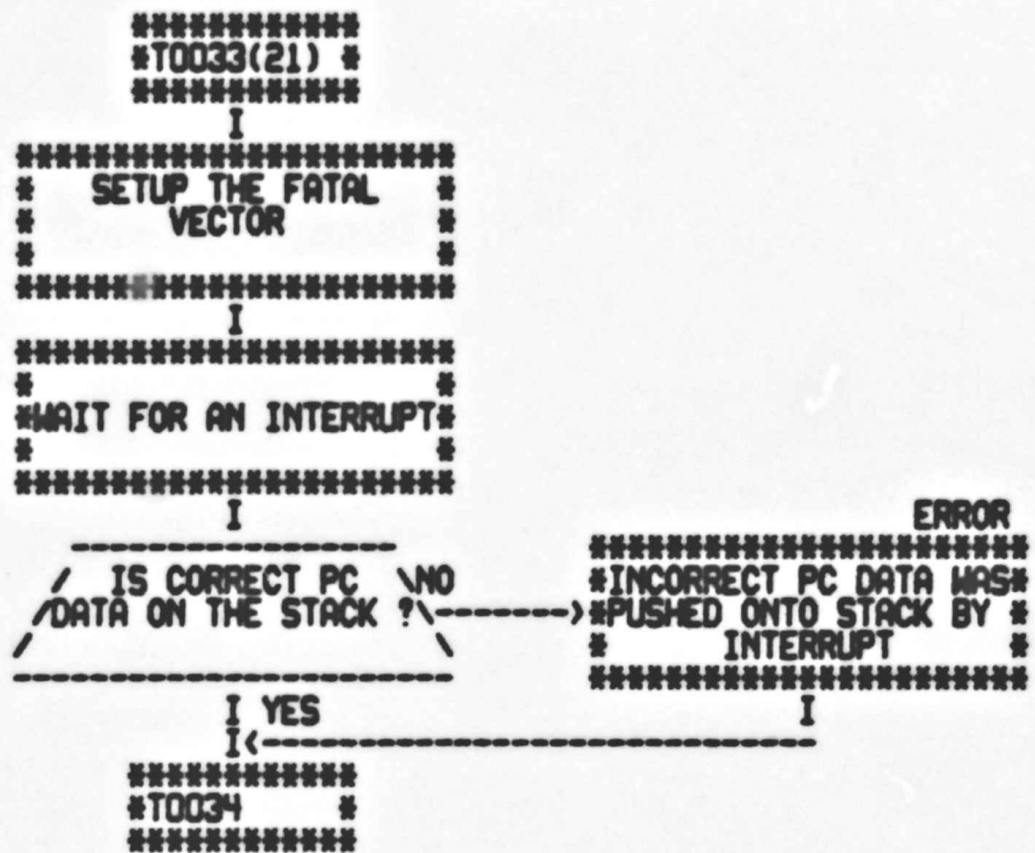


LINE CLOCK PROGRAM FLOW CHART
NO INTERRUPT AT PRIO LEVEL 7 TEST



LINE CLOCK PROGRAM FLOW CHART
CC PUSH TEST





26	OPERATIONAL SWITCH SETTINGS
28	BASIC DEFINITIONS
34	TRAP CATCHER
(1)	STARTING ADDRESS(ES)
41	COMMON TAGS
(1)	ERROR POINTER TABLE
101	TEST THAT THE LKS CAN BE REFERENCED WITHOUT A BUS ERROR
114	TEST THAT START CLEARS LINE CLOCK INTERRUPT ENABLE BIT
127	TEST THAT START SETS CLOCK FLAG
140	TEST THAT CLOCK FLAG WILL SET AFTER SUFFICIENT PERIOD OF TIME (20 MS MIN)
156	TEST THAT INTERRUPT ENABLE BIT MAY BE SET
172	TEST THAT INTERRUPT ENABLE BIT MAY BE CLEARED
189	TEST THAT CLOCK INTERRUPTS TO CORRECT VECTOR ADDRESS
214	TEST THAT CLOCK WILL INTERRUPT WITH PROCESSOR AT PRIORITY 5
238	TEST THAT CLOCK WILL NOT INTERRUPT WITH PROCESSOR PRIORITY 6
257	TEST THAT RESET SETS CLOCK FLAG
273	TEST LINE CLOCK REPEATABILITY
321	LINE CLOCK REGISTER ADDRESSING TEST
345	LINE CLOCK REGISTER ADDRESSING TEST
366	LINE CLOCK REGISTER ADDRESSING TEST
387	LINE CLOCK REGISTER ADDRESSING TEST
408	LINE CLOCK REGISTER ADDRESSING TEST
429	LINE CLOCK REGISTER ADDRESSING TEST
450	LINE CLOCK REGISTER HIGH BYTE TEST
470	CLOCK FLAG BIT TEST
485	INTERRUPT TEST
503	NO SACK TIMEOUT TEST
525	RESET TEST
546	CLOCK FLAG BIT TEST
569	CLOCK FLAG AFTER INTERRUPT TEST
594	NO INTERRUPT AT PRIORITY 7 TEST
613	CC PUSH TEST FOR CLOCK INTERRUPTS
636	PC PUSH TEST FOR CLOCK INTERRUPTS
658	END OF PASS INDICATING
663	END OF PASS ROUTINE
680	SCOPE HANDLER ROUTINE
681	TYPE ROUTINE
682	CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
683	BINARY TO OCTAL (ASCII) AND TYPE
684	ERROR MESSAGE TIMEOUT ROUTINE
693	ERROR HANDLER ROUTINE
694	TRAP DECODER
(3)	TRAP TABLE
695	POWER DOWN AND UP ROUTINES


```

(1)      000007      R7=      X7      ;GENERAL REGISTER
(1)      .EQUIV     R6,SP      ;STACK POINTER
(1)      .EQUIV     R7,PC      ;PROGRAM COUNTER
(1)
(1)      .# "SWITCH REGISTER" SWITCH DEFINITIONS
(1)      100000     SW15=     100000
(1)      040000     SW14=     40000
(1)      020000     SW13=     20000
(1)      010000     SW12=     10000
(1)      004000     SW11=     4000
(1)      002000     SW10=     2000
(1)      001000     SW09=     1000
(1)      000400     SW08=     400
(1)      000200     SW07=     200
(1)      000100     SW06=     100
(1)      000040     SW05=     40
(1)      000020     SW04=     20
(1)      000010     SW03=     10
(1)      000004     SW02=     4
(1)      000002     SW01=     2
(1)      000001     SW00=     1
(1)      .EQUIV     SW09,SW9
(1)      .EQUIV     SW08,SW8
(1)      .EQUIV     SW07,SW7
(1)      .EQUIV     SW06,SW6
(1)      .EQUIV     SW05,SW5
(1)      .EQUIV     SW04,SW4
(1)      .EQUIV     SW03,SW3
(1)      .EQUIV     SW02,SW2
(1)      .EQUIV     SW01,SW1
(1)      .EQUIV     SW00,SW0
(1)
(1)      .#DATA BIT DEFINITIONS (BIT00 TO BIT15)
(1)      100000     BIT15=    100000
(1)      040000     BIT14=    40000
(1)      020000     BIT13=    20000
(1)      010000     BIT12=    10000
(1)      004000     BIT11=    4000
(1)      002000     BIT10=    2000
(1)      001000     BIT09=    1000
(1)      000400     BIT08=    400
(1)      000200     BIT07=    200
(1)      000100     BIT06=    100
(1)      000040     BIT05=    40
(1)      000020     BIT04=    20
(1)      000010     BIT03=    10
(1)      000004     BIT02=    4
(1)      000002     BIT01=    2
(1)      000001     BIT00=    1
(1)      .EQUIV     BIT09,BIT9
(1)      .EQUIV     BIT08,BIT8
(1)      .EQUIV     BIT07,BIT7
(1)      .EQUIV     BIT06,BIT6
(1)      .EQUIV     BIT05,BIT5

```

```

(1)      .EQUIV BIT04,BIT4
(1)      .EQUIV BIT03,BIT3
(1)      .EQUIV BIT02,BIT2
(1)      .EQUIV BIT01,BIT1
(1)      .EQUIV BIT00,BIT0

(1)      ;#BASIC "CPU" TRAP VECTOR ADDRESSES
(1)      000004 ERRVEC= 4 ;TIME OUT AND OTHER ERRORS
(1)      000010 RESVEC= 10 ;RESERVED AND ILLEGAL INSTRUCTIONS
(1)      000014 TBITVEC=14 ;"T" BIT
(1)      000014 TRIVEC= 14 ;TRACE TRAP
(1)      000014 BPTVEC= 14 ;BREAKPOINT TRAP (BPT)
(1)      000020 IOTVEC= 20 ;INPUT/OUTPUT TRAP (IOT) ##SCOPE##
(1)      000024 PWRVEC= 24 ;POWER FAIL
(1)      000030 EMTVEC= 30 ;EMULATOR TRAP (EMT) ##ERROR##
(1)      000034 TRAPVEC=34 ;"TRAP" TRAP
(1)      000060 TKVEC= 60 ;TTY KEYBOARD VECTOR
(1)      000064 TPVEC= 64 ;TTY PRINTER VECTOR
(1)      000240 PIRQVEC=240 ;PROGRAM INTERRUPT REQUEST VECTOR
(1)      ;MISCELLANEOUS EQUATES
(1)      177546 LKS=177546
(1)      000240 NOP=240
(1)      000774 BUF2=774
(1)      000776 BUF1=776

(1)      .SBTTL TRAP CATCHER
(1)      .=0
(1)      000000 ;#ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
(1)      ;#SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
(1)      ;#LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS

(1)      .SBTTL STARTING ADDRESS(ES)
(1)      000200 .=200

(1)      000200 000137 001400 JMP @#KSTART ;JUMP TO STARTING ADDRESS OF PROGRAM
(1)      38 000050 000050 .=50
(1)      39 000050 000000 .WORD 0
(1)      40 000052 000000 .WORD 0
    
```

```

41      ;*****
(1)      ;.SBTTL COMMON TAGS
(1)      ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
(1)      ;*USED IN THE PROGRAM.
(1)      000046      000046      .=46
(1)      000046      005640      SENDAD      ;LOGICAL END OF PROGRAM
(1)      001100      001100      .=1100
(1)      001100      SCMTAG:      ;START OF COMMON TAGS
(1)      001100      SPASS: .WORD      0      ;CONTAINS PASS COUNT
(1)      001102      STSTNM: .BYTE      000      ;CONTAINS THE TEST NUMBER
(1)      001103      SERFLG: .BYTE      000      ;CONTAINS ERROR FLAG
(1)      001104      SICNT: .WORD      00000      ;CONTAINS SUBTEST ITERATION COUNT
(1)      001106      SLPADR: .WORD      00000      ;CONTAINS SCOPE LOOP 1100
(1)      001110      SLPERR: .WORD      00000      ;CONTAINS SCOPE RETURN FOR ERRORS
(1)      001112      SERTTL: .WORD      00000      ;CONTAINS TOTAL ERRORS DETECTED
(1)      001114      SITEMB: .BYTE      000      ;CONTAINS ITEM CONTROL BYTE
(1)      001115      SERMAX: .BYTE      001      ;CONTAINS MAX. ERRORS PER TEST
(1)      001116      SERRPC: .WORD      00000      ;CONTAINS PC OF LAST ERROR INSTRUCTION
(1)      001120      SGDADR: .WORD      00000      ;CONTAINS 1100 OF 'GOOD' DATA
(1)      001122      SBDADR: .WORD      00000      ;CONTAINS 1100 OF 'BAD' DATA
(1)      001124      SGDDAT: .WORD      00000      ;CONTAINS 'GOOD' DATA
(1)      001126      SBDAT: .WORD      00000      ;CONTAINS 'BAD' DATA
(1)      001130      000000 000000 .WORD      0,0,0      ;RESERVED--NOT TO BE USED
(1)      001136      177560      STKS: 177560      ;TTY KBD STATUS
(1)      001140      177562      STKB: 177562      ;TTY KBD BUFFER
(1)      001142      177564      STPS: 177564      ;TTY PRINTER STATUS REG. 1100
(1)      001144      177566      STPB: 177566      ;TTY PRINTER BUFFER REG. 1100
(1)      001146      000      SNULL: .BYTE      0      ;CONTAINS NULL CHARACTER FOR FILLS
(1)      001147      002      SFILLS: .BYTE      2      ;CONTAINS # OF FILLER CHARACTERS REQUIRED
(1)      001150      012      SFILLC: .BYTE      12      ;INSERT FILL CHARS. AFTER A "LINE FEED"
(1)      001151      000      STPFLG: .BYTE      0      ;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
(1)      001152      000000      SREGAD: .WORD      0      ;CONTAINS THE 1100 FROM WHICH (SREGD) WAS OBTAINED
(3)      001154      000000      SREGD: .WORD      0      ;CONTAINS ((SREGAD)+0)
(3)      001156      000000      SREG1: .WORD      000      ;CONTAINS ((SREGAD)+2)
(3)      001160      000000      SREG2: .WORD      000      ;CONTAINS ((SREGAD)+4)
(3)      001162      000000      SREG3: .WORD      000      ;CONTAINS ((SREGAD)+6)
(3)      001164      000000      SREG4: .WORD      000      ;CONTAINS ((SREGAD)+10)
(3)      001166      000000      SREG5: .WORD      000      ;CONTAINS ((SREGAD)+12)
(3)      001170      000000      SREG6: .WORD      000      ;CONTAINS ((SREGAD)+14)
(3)      001172      000000      SREG7: .WORD      000      ;CONTAINS ((SREGAD)+16)
(3)      001174      000000      STMP0: .WORD      00000      ;USER DEFINED
(3)      001176      000000      STMP1: .WORD      00000      ;USER DEFINED
(3)      001200      000000      STMP2: .WORD      00000      ;USER DEFINED
(3)      001202      000000      STMP3: .WORD      00000      ;USER DEFINED
(3)      001204      000000      STMP4: .WORD      00000      ;USER DEFINED
(3)      001206      000000      STMP5: .WORD      00000      ;USER DEFINED
(3)      001210      000000      STMP6: .WORD      00000      ;USER DEFINED
(3)      001212      000000      STMP7: .WORD      00000      ;USER DEFINED

```


(1)	001214	000000	
(1)	001216	000000	
(1)	001220	177607	000377
(1)	001224	077	
(1)	001225	015	
(1)	001226	000012	
(3)	001230	000000	

STIMES:	0
SESCAPE:	0
SBELL:	.ASCIZ <207><377><377>
SQUES:	.ASCIZ /?/
SCRLF:	.ASCIZ <15>
SLF:	.ASCIZ <12>
WORD:	000000

:	MAX. NUMBER OF ITERATIONS
:	ESCAPE ON ERROR 1100
:	CODE FOR BELL
:	QUESTION MARK
:	CARRIAGE RETURN
:	LINE FEED

```

(2) ;*****
(1)
(1) .SBTTL ERROR POINTER TABLE
(1)
(1) ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
(1) ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
(1) ;*LOCATION SITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
(1) ;*NOTE1: IF SITEMB IS 0 THE ONLY PERTINENT DATA IS (SERRPC).
(1) ;*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
(1)
(1) ;* EM ;POINTS TO THE ERROR MESSAGE
(1) ;* DH ;POINTS TO THE DATA HEADER
(1) ;* DT ;POINTS TO THE DATA
(1) ;* DF ;POINTS TO THE DATA FORMAT
(1)
(1) SERRTB:
(1) 42 001232 007752 EM1 ;"PC PS SP TEST# LKS LKS
(1) 43 001234 010033 DH1 ;"PC PS SP TEST# WAS S/B
(1) 44 001236 010114 DT1 ;SERRPC, SREG7, SREG6, SREG5, LKS, SGDDAT
(1) 45 001240 000000 0
(1) 47 001242 010132 EM2 ;"CLOCK FAILED TO INTERRUPT"
(1) 48 001244 010164 DH2 ;"PC PS SP TEST# LKS
(1) 49 001246 010236 DT2 ;SERRPC, SREG7, SREG6, SREG5, LKS
(1) 50 001250 000000 0
(1) 51 001252 010252 EM3 ;"CLOCK INTERRUPTED WHEN THE PROCESSOR PRIORITY WAS TOO HIGH"
(1) 52 001254 010345 DH3 ;"PC PS SP TEST# LKS
(1) 53 001256 010416 DT3 ;SERRPC, SREG7, SREG6, SREG5, LKS
(1) 54 001260 000000 0
(1) 57 001262 010432 EM4 ;"CLOCK GIVES UNEQUAL # OF PULSES OVER TWO EQUAL PERIODS OF TIME"
(1) 58 001264 010531 DH4 ;"PC PS SP TEST# 1ST 2ND"
(1) 59 001266 010666 DT4 ;SERRPC, SREG7, SREG6, SREG5, SREG1, SREG0
(1) 60 001270 000000 0
(1) 62 001272 010704 EM5 ;"LKS REGISTER RESPONDS TO ANOTHER ADDRESS"
(1) 63 001274 010755 DH5 ;"PC PS SP TEST# ADDRESS"
(1) 64 001276 011026 DT5 ;SERRPC, SREG7, SREG6, SREG5, SGADR
(1) 65 001300 000000 0
(1) 66 001302 011042 EM6 ;"A NO SACK TIMEOUT HAS OCCURED"
(1) 67 001304 011100 DH6 ;"PC PS SP TEST# LKS
(1) 68 001306 011152 DT6 ;SERRPC, SREG7, SREG6, SREG5, LKS
(1) 69 001310 000000 0
(1) 71 001312 011166 EM7 ;"WRONG CONDITION CODES WERE PUT ONTO STACK BY INTERRUPT"
(1) 72 001314 011255 DH7 ;"PC PS SP TEST# CC CC"
(1) 73 001316 011334 DT7 ;SERRPC, SREG7, SREG6, SREG5, BUF1, SGDDAT
(1) 74 001320 000000 0
(1) 75 001322 011352 EM10 ;"WRONG PC PUT ONTO THE STACK BY AN INTERRUPT"
(1) 76 001324 011426 DH10 ;"PC PS SP TEST#

```

```

79 001326 011512 DT10 ;SERRPC,SREG7,SREG6,SREG5,BUF2,SGDDAT
80 001330 000000 0
81
82 001332 011530 EM11 ;"TRAPPED TRYING TO ACCESS LKS REGISTER"
83 001334 011606 DH11 ;"(PC) (PS) (SP) TEST#"
84 001336 011644 DT11 ;SERRPC,SREG7,SREG6,SREG5
85 001340 000000 0
86
87
88 ;STARTUP CODE
89 .=1400
90 001400 012706 001000 KSTART: MOV #1000,SP ;INITIALIZE THE STACK SO WE CAN CALL THE TYPEOUT
91 001404 013746 177776 MOV PS,-(SP) ;SAVE STATUS
92 001410 004737 006216 JSR PC,STYPE ;PRINTOUT STARTUP MESSAGE
93 001414 007604 STMS ;ADDRESS OF MESSAGE "MAINDEC-11-DZKWA-D"
94 001416
(1) 001416 012706 001100 START: MOV #SCHTAG,R6 ;FIRST LOCATION TO BE CLEARED
(1) 001422 005026 CLR (R6)+ ;CLEAR MEMORY LOCATION
(1) 001424 022706 001136 CMP #STKS,R6 ;DONE?
(1) 001430 001374 BNE .-6 ;LOOP BACK IF NO
(1) 001432 012706 001000 MOV #1000,SP ;SETUP THE STACK POINTER
(1) 001436 012737 005746 000020 MOV #SCOPE,@IOTVEC ;IOT VECTOR FOR SCOPE ROUTINE
(1) 001444 012737 000340 000022 MOV #340,@IOTVEC+2 ;LEVEL 7
(1) 001452 012737 007132 000030 MOV #SERROR,@EMTVEC ;EMT VECTOR FOR ERROR ROUTINE
(1) 001460 012737 000340 000032 MOV #340,@EMTVEC+2 ;LEVEL 7
(1) 001466 012737 007324 000034 MOV #STRAP,@TRAPVEC ;TRAP VECTOR FOR TRAP CALLS
(1) 001474 012737 000340 000036 MOV #340,@TRAPVEC+2 ;LEVEL 7
(1) 001502 012737 007356 000024 MOV #SPWRON,@PMRVEC ;POWER FAILURE VECTOR
(1) 001510 012737 000340 000026 MOV #340,@PMRVEC+2 ;LEVEL 7
(1) 001516 013737 005610 005602 MOV SENDCT,SEOPCT ;SETUP END-OF-PROGRAM COUNTER
(1) 001524 112737 000001 001115 MOV #1,SERMAX ;ALLOW ONE ERROR PER TEST
(1) 001532 012737 001532 001106 MOV #,SLPADR ;INITIALIZE THE LOOP ADDRESS FOR SCOPE
95 001540 005737 000042 TST 42 ;LOADED BY A MONITOR
96 001544 001401 BEQ T0001 ;BR IF NO
97 001546 000005 RESET ;YES--GENERATE AN INIT
98
99
100
101 .SBTTL TEST THAT THE LKS CAN BE REFERENCED WITHOUT A BUS ERROR
102 :LKS ACCESS TEST
103 T0001: SCOPE
104 001550 000004 MOV #E0001,@#4 ;PREPARE FOR ADDRESSING THE LKS REGISTER. BAD HARDWARE
105 001552 012737 001606 000004 MOV #340,@#6 ;COULD CAUSE A TRAP TO 4
106 001560 012737 000340 000006 MOV #R0001,SLPADR ;TIGHTEN UP THE SCOPE LOOP A BIT IN CASE OF AN ERROR
107 001566 012737 001574 001106 R0001: MOV #1000,SP ;SETUP THE STACK POINTER IN CASE OF AN ERROR
108 001574 012706 001000 I0001: CLR @LKS ;JUST REFERENCE LKS. DONT WORRY IF IT DIDNT CLEAR YET
109 001600 005037 177546 BR T0002 ;WE DIDNT TRAP IF WE REACH HERE. GO ON TO NEXT TEST
110 001604 000401 E0001: ERROR 11 ;ERROR:::TRAPED TRYING TO ACCESS THE LKS REGISTER
111 001606 104011
112
113
114 .SBTTL TEST THAT START CLEARS LINE CLOCK INTERRUPT ENABLE BIT
115 :TEST THAT START CLEARS LINE CLOCK INTERRUPT ENABLE BIT
116 001610 000004 T0002: SCOPE

```



```

117 001612 012737 005674 000004      MOV      #TRAP0,2#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
(1) 001620 012737 000340 000006      MOV      #340,2#6       ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
118 001626 012737 001634 001106      MOV      #R0002,$LPADR  ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
119 001634 000005      R0002:  RESET
120 001636 012737 000200 001124      MOV      #200,$GDDAT    ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
121 001644 032737 000100 177546      BIT      #100,LKS       ;TEST THE INTERRUPT ENABLE BIT
122 001652 001401      BEQ      T0003
123 001654 104001      E0002:  ERROR 1        ;ERROR, CLOCK INTERRUPT ENABLE NOT CLEARED BY INIT
124
125
126
127
128      .SBTTL  TEST THAT START SETS CLOCK FLAG
      :TEST THAT START SETS CLOCK FLAG
129      T0003:  SCOPE
130 001656 000004      MOV      #TRAP0,2#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
(1) 001660 012737 000340 000006      MOV      #340,2#6       ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
131 001666 012737 000200 001124      MOV      #200,$GDDAT    ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
132 001674 012737 000170 001106      MOV      #R0003,$LPADR  ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
133 001702 012737 001710 000005      R0003:  RESET          ;SHOULD SET THE CLOCK FLAG
134 001712 105737 177546      TSTB    LKS             ;FIND OUT IF IT DID
135 001716 100401      BMI     T0004           ;GO ON TO THE NEXT TEST IF IT SET THE CLOCK FLAG
136 001720 104001      E0003:  ERROR 1        ;ERROR, CLOCK FLAG NOT SET BY INIT
137
138
139
140      .SBTTL  TEST THAT CLOCK FLAG WILL SET AFTER SUFFICIENT PERIOD OF TIME (20 MS MIN)
      :TEST THAT CLOCK FLAG WILL SET AFTER SUFFICIENT PERIOD OF TIME (20 MS MIN)
141      T0004:  SCOPE
142 001722 000004      MOV      #TRAP0,2#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
(1) 001724 012737 000340 000006      MOV      #340,2#6       ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
143 001732 012737 001746 001106      MOV      #R0004,$LPADR  ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
144 001740 012737 000174 001106      R0004:  MOV      #200,$GDDAT ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
145 001746 012737 000200 001124      CLR     LKS             ;CLEAR THE CLOCK FLAG
146 001754 005037 177546      CLR     R0              ;AND A TIMER LOCATION
147 001760 005000      R0004:  CLR     LKS             ;IS CLOCK FLAG SET
148 001762 105737 177546      TSTB    LKS
149 001766 100403      BMI     T0005
150 001770 005200      INC     R0              ;NO, INCREMENT COUNT 003 WAIT FOR SOMEMORE
151 001772 001373      BNE     R0004           ;WAIT SUFFICIENT AMOUNT OF TIME FOR CLOCK
152 001774 104001      E0004:  ERROR 1        ;ERROR, CLOCK FLAG FAILED TO SET
153
154
155
156      .SBTTL  TEST THAT INTERRUPT ENABLE BIT MAY BE SET
      :TEST THAT INTERRUPT ENABLE BIT MAY BE SET
157      T0005:  SCOPE
158 001776 000004      MOV      #TRAP0,2#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
(1) 002000 012737 000340 000006      MOV      #340,2#6       ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
159 002006 012737 000100 001124      MOV      #100,$GDDAT    ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
160 002014 012737 000100 001106      MOV      #R0005,$LPADR  ;SETUP LOOP BACK ADDRESS IN CASE OF ERROR
161 002022 012737 002036 001106      MOV      #340,$P5       ;SET PRIORITY TO LEVEL 7, NO INTERRUPTS
162 002030 012737 000340 177776      R0005:  MOV      #340,$P5
163 002036 005037 177546      CLR     LKS
164 002042 005003      CLR     R3              ;INITIALIZE A COUNTER LOCATION
(1) 002044 105737 177546      R0005:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
(1) 002050 100404      BMI     B0005           ;IF SO, CONTINUE ON WITH THE TEST

```

TEST THAT INTERRUPT ENABLE BIT MAY BE SET

```

(1) 002052 005203          INC      R3          ;IF NOT INCREMENT THE COUNTER LOCATION
(1) 002054 001373          BNE     A0005        ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
(1) 002056 104001          E0005:  ERROR 1      ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
(1) 002060 000410          BR      T0006
(1) 002062 012737 000100 177546 B0005:  MOV      #100,LKS    ;CLEAR CLOCK FLAG AND SET INTERRUPT ENABLE
165 002062 012737 000100 177546      BIT     #100,LKS    ;IS INTERRUPT ENABLE SET?
166 002070 032737 000100 177546          BNE     T0006
167 002076 001001          E1005:  ERROR 1      ;ERROR INTERRUPT ENABLE NOT SET
168 002100 104001
169
170
171

```

.SBTTL TEST THAT INTERRUPT ENABLE BIT MAY BE CLEARED
:TEST THAT INTERRUPT ENABLE BIT MAY BE CLEARED

```

172
173
174 002102 000004          T0006:  SCOPE
175 002104 012737 005674 000004      MOV     #TRAP0,2#4   ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
(1) 002112 012737 000340 000006      MOV     #340,2#6    ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
176 002120 012737 000000 001124      MOV     #0,$GDDAT   ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
177 002126 012737 002142 001106      MOV     #R0006,SLPADR ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
178 002134 012737 000340 177776      MOV     #340,PS     ;SET PRIORITY LEVEL TO 7. NO INTERRUPTS
179 002142 005037 177546          R0006:  CLR     LKS
180 002146 005003          CLR     R3          ;INITIALIZE A COUNTER LOCATION
(1) 002150 105737 177546          A0006:  TSTB   LKS     ;IS THE CLOCK FLAG SET?
(1) 002154 100404          BMI     B0006        ;IF SO, CONTINUE ON WITH THE TEST
(1) 002156 005203          INC     R3          ;IF NOT INCREMENT THE COUNTER LOCATION
(1) 002160 001373          BNE     A0006        ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
(1) 002162 104001          E0006:  ERROR 1      ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
(1) 002164 000412          BR      T0007
(1) 002166 012737 000100 177546      B0006:  MOV     #100,LKS    ;CLEAR CLOCK FLAG AND SET INTERRUPT ENABLE
181 002166 012737 000100 177546      CLR     LKS        ;CLEAR INTERRUPT ENABLE
182 002174 005037 177546          BIT     #100,LKS    ;TEST THE INTERRUPT ENABLE BIT
183 002200 032737 000100 177546          BEQ    T0007        ;IS INTERRUPT ENABLE CLEARED
184 002206 001401          E10006: ERROR 1      ;ERROR, ERROR INTERRUPT BIT CAN NOT BE CLEARED
185 002210 104001
186
187
188

```

.SBTTL TEST THAT CLOCK INTERRUPTS TO CORRECT VECTOR ADDRESS
:TEST THAT CLOCK INTERRUPTS TO CORRECT VECTOR ADDRESS

```

189
190
191 002212 000004          T0007:  SCOPE
192 002214 012737 005674 000004      MOV     #TRAP0,2#4   ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
(1) 002222 012737 000340 000006      MOV     #340,2#6    ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
193 002230 012737 000300 001124      MOV     #300,$GDDAT ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
194 002236 012737 002260 001106      MOV     #R0007,SLPADR ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
195 002244 012737 002340 000100      MOV     #00007,100  ;SET UP VECTOR RETURN POINTER
196 002252 012737 000340 000102      MOV     #340,2#102  ;1 INTERRUPT IS ENOUGH
197 002260 012706 001000          R0007:  MOV     #1000,SP   ;GET STACK READY FOR INTERRUPTS
198 002264 012737 000200 177776      MOV     #200,PS     ;SET PROCESSOR PRIORITY 4
199 002272 005037 177546          CLR     LKS
200 002276 005003          CLR     R3          ;INITIALIZE A COUNTER LOCATION
(1) 002300 105737 177546          A0007:  TSTB   LKS     ;IS THE CLOCK FLAG SET?
(1) 002304 100404          BMI     B0007        ;IF SO, CONTINUE ON WITH THE TEST
(1) 002306 005203          INC     R3          ;IF NOT INCREMENT THE COUNTER LOCATION
(1) 002310 001373          BNE     A0007        ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...

```


TEST THAT CLOCK INTERRUPTS TO CORRECT VECTOR ADDRESS

```

(1) 002312 104001 E0007: ERROR 1 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
(1) 002314 000415 BR T0010
(1) 002316 B0007: MOV #100,LKS ;ENABLE INTERRUPT
201 002316 012737 000100 177546 CLR R0
202 002324 005000 C0007: INC R0
203 002326 005200 ;STALL FOR TIME
204 002330 000240 ;WAIT FOR INTERRUPT
205 002332 001375 BNE C0007 ;ERROR, DIDNT GET INTERRUPT
206 002334 104002 E10007: ERROR 2
207 002336 000404 BR T0010 ;ENTER HERE IF INTERRUPTED
208 002340 105737 177546 D0007: TSTB LKS
209 002344 100401 BMI T0010
210 002346 104001 E20007: ERROR 1 ;ERROR, INTERRUPT NOT CAUSED BY CLOCK

```

.SBTTL TEST THAT CLOCK WILL INTERRUPT WITH PROCESSOR AT PRIORITY 5
:TEST THAT CLOCK WILL INTERRUPT WITH PROCESSOR AT PRIORITY 5

```

216 002350 000004 T0010: SCOPE
217 002352 012737 005674 000004 MOV #TRAP0,2#4 ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
(1) 002360 012737 000340 000006 MOV #340,2#6 ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
218 002366 012737 002410 001106 MOV #R0010,SLPADR ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
219 002374 012737 002470 000100 MOV #00010,100 ;SET UP VECTOR RETURN POINTER
220 002402 012737 000340 000102 MOV #340,2#102 ;NO INTERRUPTS ALLOWED AFTER THE FIRST ONE
221 002410 005037 177546 R0010: CLR LKS
222 002414 012737 000240 177776 MOV #240,PS ;SET PRIORITY 5
223 002422 012706 001000 MOV #1000,SP ;INITIALIZE THE STACK POINTER
224 002426 005003 CLR R3 ;INITIALIZE A COUNTER LOCATION
(1) 002430 105737 177546 A0010: TSTB LKS ;IS THE CLOCK FLAG SET?
(1) 002434 100404 BMI B0010 ;IF SO, CONTINUE ON WITH THE TEST
(1) 002436 005203 INC R3 ;IF NOT INCREMENT THE COUNTER LOCATION
(1) 002440 001373 BNE A0010 ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
(1) 002442 104001 E0010: ERROR 1 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
(1) 002444 000415 BR T0011
(1) 002446 B0010: MOV #100,LKS ;ENABLE INTERRUPT
225 002446 012737 000100 177546 CLR R0
226 002454 005000 C0010: INC R0
227 002456 005200 ;STALL FOR SOME TIME
228 002460 000240 ;WAIT FOR INTERRUPT
229 002462 001375 BNE C0010 ;ERROR, INTERRUPT FAILED TO OCCUR
230 002464 104002 E10010: ERROR 2
231 002466 000404 BR T0011 ;ENTER HERE IF INTERRUPTED
232 002470 105737 177546 D0010: TSTB LKS
233 002474 100401 BMI T0011
234 002476 104001 E20010: ERROR 1 ;ERROR, INTERRUPTDID NOT CLEAR THE CLOCK FLAG

```

.SBTTL TEST THAT CLOCK WILL NOT INTERRUPT WITH PROCESSOR PRIORITY 6
:TEST THAT CLOCK WILL NOT INTERRUPT WITH PROCESSOR PRIORITY 6

```

240 002500 000004 T0011: SCOPE
(1) 002502 012737 005674 000004 MOV #TRAP0,2#4 ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
(1) 002510 012737 000340 000006 MOV #340,2#6 ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
242 002516 012737 002524 001106 MOV #R0011,SLPADR ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR

```



```

(1) 002524 005037 177546 R0011: CLR LKS
(1) 002530 005003 CLR R3 ;INITIALIZE A COUNTER LOCATION
(1) 002532 105737 177546 A0011: TSTB LKS ;IS THE CLOCK FLAG SET?
(1) 002536 100404 BMI B0011 ;IF SO, CONTINUE ON WITH THE TEST
(1) 002540 005203 INC R3 ;IF NOT INCREMENT THE COUNTER LOCATION
(1) 002542 001373 BNE A0011 ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
(1) 002544 104001 E0011: ERROR 1 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
(1) 002546 000427 BR T0012
(1) 002550 B0011: MOV #1000,SP ;INITIALIZE THE STACK POINTER
(1) 002550 012706 001000 MOV #300,PS ;SET PRIORITY 6
(1) 002554 012737 000300 177776 MOV #E1011,100 ;SET UP VECTOR RETURN
(1) 002562 012737 002612 000100 MOV #100,LKS ;ENABLE INTERRUPT
(1) 002570 012737 000100 177546 CLR R3 ;INITIALIZE A COUNTER LOCATION
(1) 002576 005003 C0011: TSTB LKS ;IS THE CLOCK FLAG SET?
(1) 002600 105737 177546 BMI D0011 ;IF SO, CONTINUE ON WITH THE TEST
(1) 002604 100404 INC R3 ;IF NOT INCREMENT THE COUNTER LOCATION
(1) 002606 005203 BNE C0011 ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
(1) 002610 001373 E1011: ERROR 1 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
(1) 002612 104001 BR T0011
(1) 002614 000731 D0011: NOP
(1) 002616 000240 NOP ;GIVE CLOCK EXTRA TIME TO INTERRUPT
(1) 002620 000240 BR T0012
(1) 002622 000401 E20011: ERROR 3 ;ERROR, CLOCK INTERRUPTED WITHOUT HAVING PRIORITY
(1) 002624 104003

.SBTTL TEST THAT RESET SETS CLOCK FLAG
;TEST THAT RESET SETS CLOCK FLAG
T0012: SCOPE
MOV #200,SGDDAT ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
MOV #TRAP0,2#4 ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
MOV #340,2#6 ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
MOV #R0012,SLPADR ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
R0012: CLR LKS ;CLEAR CLOCK FLAG
CLR R3 ;INITIALIZE A COUNTER LOCATION
A0012: TSTB LKS ;IS THE CLOCK FLAG SET?
BMI B0012 ;IF SO, CONTINUE ON WITH THE TEST
INC R3 ;IF NOT INCREMENT THE COUNTER LOCATION
BNE A0012 ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
E0012: ERROR 1 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
BR T0013
B0012: CLR LKS
RESET ;SHOULD SET CLOCK FLAG
TSTB LKS
BMI T0013
E10012: ERROR 1 ;ERROR, RESET DIDN'T SET CLOCK FLAG

.SBTTL TEST LINE CLOCK REPEATABILITY
;TEST LINE CLOCK REPEATABILITY

```

```

275 ;MAKE SURE THAT OVER TWO EQUAL PERIODS OF TIME
276 ;THE CLOCK PUTS OUT THE SAME NUMBER OF PULSES
277 002722 000004 T0013: SCOPE
278 002724 005000 R0013: CLR R0 ;CLEAR 1ST TIME COUNT
279 002726 005001 R1013: CLR R1 ;CLEAR 1ST CLOCK COUNT
280 002730 012737 000340 177776 R1013: MOV #340,PS ;SET PRIORITY 7
281 002736 012737 002736 001106 R1013: MOV #R1013,SLPADR ;ERROR IN NEXT FEW INSTRUCTIONS CAUSES A SHORT SCOPE LO
282 002744 005037 177546 R1013: CLR LKS
283 ;SYNC ON CLOCK FLAG A COUPLE OF TIMES
284 002750 005003 R0013: CLR R3 ;INITIALIZE A COUNTER LOCATION
(1) 002752 105737 177546 R0013: TSTB LKS ;IS THE CLOCK FLAG SET?
(1) 002756 100404 R0013: BMI B0013 ;IF SO, CONTINUE ON WITH THE TEST
(1) 002760 005203 R0013: INC R3 ;IF NOT INCREMENT THE COUNTER LOCATION
(1) 002762 001373 R0013: BNE A0013 ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
(1) 002764 104001 E0013: ERROR 1 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
(1) 002766 000510 E0013: BR T0014
(1) 002770 000510 B0013: B0013:
285 002770 012737 002770 001106 R2013: MOV #R2013,SLPADR ;MAKE SCOPE LOOP SHORT IN CASE OF AN ERROR
286 002776 005037 177546 R2013: CLR LKS
287 003002 005003 C0013: CLR R3 ;INITIALIZE A COUNTER LOCATION
(1) 003004 105737 177546 C0013: TSTB LKS ;IS THE CLOCK FLAG SET?
(1) 003010 100404 C0013: BMI D0013 ;IF SO, CONTINUE ON WITH THE TEST
(1) 003012 005203 C0013: INC R3 ;IF NOT INCREMENT THE COUNTER LOCATION
(1) 003014 001373 C0013: BNE C0013 ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
(1) 003016 104001 E10013: ERROR 1 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
(1) 003020 000473 E10013: BR T0014
(1) 003022 000473 D0013: D0013:
288 003022 005037 177546 F0013: CLR LKS ;IS CLOCK FLAG SET
289 003026 105737 177546 F0013: TSTB LKS ;NO
290 003032 100003 F0013: BPL G0013 ;+1 TO CLOCK COUNT
291 003034 005201 F0013: INC R1 ;CLEAR CLOCK IF SET
292 003036 005037 177546 G0013: CLR LKS ;+1 TO TIME COUNT
293 003042 005200 G0013: INC R0 ;REPEAT UNTIL R0=0
294 003044 001370 G0013: BNE F0013 ;CLEAR 2ND TIME COUNT
295 003046 005000 G0013: CLR R0 ;CLEAR 2ND CLOCK COUNT
296 003050 005002 G0013: CLR R2 ;INSURE A SHORT SCOPE LOOP
297 003052 012737 003052 001106 R3013: MOV #R3013,SLPADR
298 003060 005037 177546 R3013: CLR LKS
299 ;SYNC ON CLOCK FLAG TWICE
300 003064 005003 H0013: CLR R3 ;INITIALIZE A COUNTER LOCATION
(1) 003066 105737 177546 H0013: TSTB LKS ;IS THE CLOCK FLAG SET?
(1) 003072 100404 H0013: BMI J0013 ;IF SO, CONTINUE ON WITH THE TEST
(1) 003074 005203 H0013: INC R3 ;IF NOT INCREMENT THE COUNTER LOCATION
(1) 003076 001373 H0013: BNE H0013 ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
(1) 003100 104001 E20013: ERROR 1 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
(1) 003102 000442 E20013: BR T0014
(1) 003104 000442 J0013: J0013:
301 003104 012737 003104 001106 R4013: MOV #R4013,SLPADR ;INSURE A SHORT SCOPE LOOP
302 003112 005037 177546 R4013: CLR LKS
303 003116 005003 K0013: CLR R3 ;INITIALIZE A COUNTER LOCATION
(1) 003120 105737 177546 K0013: TSTB LKS ;IS THE CLOCK FLAG SET?
(1) 003124 100404 K0013: BMI L0013 ;IF SO, CONTINUE ON WITH THE TEST
(1) 003126 005203 K0013: INC R3 ;IF NOT INCREMENT THE COUNTER LOCATION
(1) 003130 001373 K0013: BNE K0013 ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...

```



```

(1) 003132 104001 E30013: ERROR 1 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
(1) 003134 000425 BR T0014
(1) 003136 012737 002724 001106 L0013: MOV #R0013,SLPADR ;MUST LOOP BACK TO BEGINING OF TEST IF EEROR COMES NOW
003138 005037 177546 CLR LKS
003140 105737 177546 M0013: TSTB LKS ;IS CLOCK FLAG SET
003142 100003 BPL N0013 ;NO
003144 005202 INC R2 ;+1 TO CLOCK COUNT
003146 005037 177546 CLR LKS ;CLEAR CLOCK IF SET
003148 005200 N0013: INC R0 ;+1 TO TIME COUNT
003150 001370 BNE M0013 ;REPEAT UNTIL R0=0
003152 020102 CMP R1,R2 ;IS 1ST CLOCK COUNT EQUAL TO 2ND CLOCK COUNT?
003154 001406 BEQ T0014 ;YES
003156 010137 001156 E40013: MOV R1,SREG1 ;GET R1 READY FOR PRINTOUT
003158 010237 001160 MOV R2,SREG2 ;GET R2 READY FOR PRINTOUT
003160 104004 ERROR 4 ;ERROR, CLOCK FLAG OCCURRED DIFFERENT
003162 000240 NOP ;NUMBER OF TIMES IN EQUAL PERIODS

```

```

.SBTTL LINE CLOCK REGISTER ADDRESSING TEST
;LINE CLOCK REGISTER ADDRESSING TEST
;TEST THAT THE "LKS" REGISTER CAN NOT BE ADDRESSED AS ANYTHING BUT 177546
;SET A LOCATION THAT IS CLOSE(DIFFERS BY 1 BIT) TO THE LKS REGISTER
;TO 100. IF THE LKS ALSO CHANGES, THEN SIGNAL AN ERROR

```

```

003210 000004 T0014: SCOPE
003212 005037 001124 CLR SGDDAT
003214 012737 003254 001106 MOV #R0014,SLPADR ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
003216 012737 157546 001120 MOV #157546,SGDADR ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 13 CLEAR
003218 012737 003272 000004 MOV #A0014,4 ;SETUP VECTOR IN CASE IT IS NONEXISTANT
003220 012737 000340 000006 MOV #340,6
003222 012737 000340 177776 MOV #340,2#PS ;NO INTERRUPTS NOW
003224 012706 001000 R0014: MOV #1000,SP ;SETUP THE STACK
003226 005037 177546 CLR LKS
003228 012777 000100 175626 I0014: MOV #100,2SGDADR ;SET THE "CLOSE" ADDRESS TO = 100
003230 032737 000100 177546 A0014: BIT #100,LKS ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
003232 001401 BEQ B0014
003234 104005 E0014: ERROR 5 ;IT AFFECTED "LKS" -- ERROR
003236 005037 177546 B0014: CLR LKS
003238 012737 003322 000004 MOV #T0015,4
003240 005077 175576 CLR 2SGDADR ;CLEAR OUT THE "CLOSE" ADDRESS

```

```

.SBTTL LINE CLOCK REGISTER ADDRESSING TEST
;LINE CLOCK REGISTER ADDRESSING TEST

```

```

003322 000004 T0015: SCOPE
003324 012737 003404 000004 MOV #A0015,4 ;SETUP VECTOR IN CASE IT IS NONEXISTANT
003326 012737 000340 000006 MOV #340,6
003328 005037 001124 CLR SGDDAT
003330 012737 003366 001106 MOV #R0015,SLPADR ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
003332 012737 177146 001120 MOV #177146,SGDADR ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 8 CLEAR
003334 012737 000340 177776 MOV #340,2#PS ;NO INTERRUPTS NOW
003336 012706 001000 R0015: MOV #1000,SP ;SETUP THE STACK

```



```

355 003372 005037 177546          CLR      LKS
356 003376 012777 000100 175514 I0015:  MOV    #100,2SGDADR ;SET THE "CLOSE" ADDRESS TO = 100
357 003404 032737 000100 177546 A0015:  BIT    #100,LKS ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
358 003412 001401          BEQ    B0015
359 003414 104005          E0015:  ERROR 5 ;IT AFFECTED "LKS" -- ERROR
360 003416 005037 177546          B0015:  CLR    LKS
361 003422 012737 003434 000004      MOV    #T0016 ,4
362 003430 005077 175464          CLR    2SGDADR ;CLEAR OUT THE "CLOSE" ADDRESS
363
364
365
366
367 .SBTTL LINE CLOCK REGISTER ADDRESSING TEST
:LINE CLOCK REGISTER ADDRESSING TEST
368 003434 000004          T0016:  SCOPE
369 003436 012737 003516 000004      MOV    #A0016,4 ;SETUP VECTOR IN CASE IT IS NONEXISTANT
370 003444 012737 000340 000006      MOV    #340,6
371 003452 005037 001124          CLR    SGDADR
372 003456 012737 003500 001106      MOV    #R0016,SLPADR ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
373 003464 012737 177446 001120      MOV    #177446,SGDADR ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 6 CLEAR
374 003472 012737 000340 177776      MOV    #340,2#PS ;NO INTERRUPTS NOW
375 003500 012706 001000          R0016:  MOV    #1000,SP ;SETUP THE STACK
376 003504 005037 177546          CLR    LKS
377 003510 012777 000100 175402 I0016:  MOV    #100,2SGDADR ;SET THE "CLOSE" ADDRESS TO = 100
378 003516 032737 000100 177546 A0016:  BIT    #100,LKS ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
379 003524 001401          BEQ    B0016
380 003526 104005          E0016:  ERROR 5 ;IT AFFECTED "LKS" -- ERROR
381 003530 005037 177546          B0016:  CLR    LKS
382 003534 012737 003546 000004      MOV    #T0017 ,4
383 003542 005077 175352          CLR    2SGDADR ;CLEAR OUT THE "CLOSE" ADDRESS
384
385
386
387 .SBTTL LINE CLOCK REGISTER ADDRESSING TEST
:LINE CLOCK REGISTER ADDRESSING TEST
388
389 003546 000004          T0017:  SCOPE
390 003550 012737 003630 000004      MOV    #A0017,4 ;SETUP VECTOR IN CASE IT IS NONEXISTANT
391 003556 012737 000340 000006      MOV    #340,6
392 003564 005037 001124          CLR    SGDADR
393 003570 012737 003612 001106      MOV    #R0017,SLPADR ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
394 003576 012737 177556 001120      MOV    #177556,SGDADR ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 3 SET
395 003604 012737 000340 177776      MOV    #340,2#PS ;NO INTERRUPTS NOW
396 003612 012706 001000          R0017:  MOV    #1000,SP ;SETUP THE STACK
397 003616 005037 177546          CLR    LKS
398 003622 012777 000100 175270 I0017:  MOV    #100,2SGDADR ;SET THE "CLOSE" ADDRESS TO = 100
399 003630 032737 000100 177546 A0017:  BIT    #100,LKS ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
400 003636 001401          BEQ    B0017
401 003640 104005          E0017:  ERROR 5 ;IT AFFECTED "LKS" -- ERROR
402 003642 005037 177546          B0017:  CLR    LKS
403 003646 012737 003660 000004      MOV    #T0020 ,4
404 003654 005077 175240          CLR    2SGDADR ;CLEAR OUT THE "CLOSE" ADDRESS
405
406
407
408 .SBTTL LINE CLOCK REGISTER ADDRESSING TEST

```

```

409 :LINE CLOCK REGISTER ADDRESSING TEST
410 T0020: SCOPE
411 003660 000004 :MOV #A0020,4 ;SETUP VECTOR IN CASE IT IS NONEXISTANT
412 003662 012737 003742 000004 :MOV #340,6
413 003670 012737 000340 000006 :CLR $GDDAT
414 003702 012737 003724 001106 :MOV #R0020,$LPADR ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
415 003710 012737 177566 001120 :MOV #177566,$GDADR ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 4 SET
416 003716 012737 000340 177776 :MOV #340,$PS ;NO INTERRUPTS NOW
417 003724 012706 001000 R0020: :MOV #1000,$P ;SETUP THE STACK
418 003730 005037 177546 :CLR LKS
419 003734 012777 000100 175156 I0020: :MOV #100,$SGDADR ;SET THE "CLOSE" ADDRESS TO = 100
420 003742 032737 000100 177546 A0020: :BIT #100,LKS ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
421 003750 001401 :BEQ B0020
422 003752 104005 E0020: :ERROR 5 ;IT AFFECTED "LKS" -- ERROR
423 003754 005037 177546 B0020: :CLR LKS
424 003760 012737 003772 000004 :MOV #T0021,4
425 003766 005077 175126 :CLR $SGDADR ;CLEAR OUT THE "CLOSE" ADDRESS

```

```

426
427
428
429
430 .SBTTL LINE CLOCK REGISTER ADDRESSING TEST
431 :LINE CLOCK REGISTER ADDRESSING TEST
432 T0021: SCOPE
433 003772 000004 :MOV #A0021,4 ;SETUP VECTOR IN CASE IT IS NONEXISTANT
434 003774 012737 004054 000004 :MOV #340,6
435 004002 012737 000340 000006 :CLR $GDDAT
436 004010 005037 001124 :MOV #R0021,$LPADR ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
437 004014 012737 004036 001106 :MOV #177746,$GDADR ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 7 SET
438 004022 012737 177746 001120 :MOV #340,$PS ;NO INTERRUPTS NOW
439 004030 012737 000340 177776 :MOV #1000,$P ;SETUP THE STACK
440 004036 012706 001000 R0021: :MOV #1000,$P
441 004042 005037 177546 :CLR LKS
442 004046 012777 000100 175044 I0021: :MOV #100,$SGDADR ;SET THE "CLOSE" ADDRESS TO = 100
443 004054 032737 000100 177546 A0021: :BIT #100,LKS ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
444 004062 001401 :BEQ B0021
445 004064 104005 E0021: :ERROR 5 ;IT AFFECTED "LKS" -- ERROR
446 004066 005037 177546 B0021: :CLR LKS
447 004072 012737 004104 000004 :MOV #T0022,4
448 004100 005077 175014 :CLR $SGDADR ;CLEAR OUT THE "CLOSE" ADDRESS

```

```

449
450 .SBTTL LINE CLOCK REGISTER HIGH BYTE TEST
451 :LINE CLOCK REGISTER HIGH BYTE TEST
452 :MAKE SURE THE LKS REGISTER LOW BYTE RESPONDS TO THE HIGH BYTES ADDRESS
453 T0022: SCOPE
454 004104 000004 :MOV #TRAP0,$4 ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
455 004106 012737 005674 000004 :MOV #340,$6 ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
456 (1) 004114 012737 000340 000006 :MOV #100,$GDDAT
457 004122 012737 000100 001124 :MOV #R0022,$LPADR ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
458 004130 012737 004152 001106 :MOV #177547,$GDADR ;HIGH BYTE OF THE LKS REGISTER
459 004136 012737 177547 001120 :MOV #340,$PS ;NO INTERRUPTS NOW
460 004144 012737 000340 177776 :MOV #1000,$P ;SETUP THE STACK
461 004152 012706 001000 R0022: :MOV #1000,$P
462 004156 005037 177546 :CLR LKS
463 004162 112777 000100 174730 I0022: :MOVB #100,$SGDADR ;SET THE HIGH BYTE ADDRESS TO = 100

```



```

462 004170 032737 000100 177546 BIT #100,LKS ;MAKE SURE THAT "LKS" LOW BYTE WAS AFFECTED
463 004176 001001 BNE A0022
464 004200 104005 E0022: ERROR 5 ;SHOULD HAVE SET BIT 7. ERROR
465 004202 005037 177546 A0022: CLR LKS
466 004206 000005 RESET
467
468
469
470 .SBTTL CLOCK FLAG BIT TEST
471 :CLOCK FLAG BIT TEST
472 004210 000004 T0023: SCOPE
473 004212 012737 005674 000004 MOV #TRAP0,2#4 ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
(1) 004220 012737 000340 000006 MOV #340,2#6 ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
474 004226 012737 000200 001124 MOV #200,$GDDAT
475 004234 012737 004242 001106 MOV #R0023,SLPADR ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
476 004242 005037 177546 R0023: CLR LKS
477 004246 005003 CLR R3 ;INITIALIZE A COUNTER LOCATION
(1) 004250 105737 177546 A0023: TSTB LKS ;IS THE CLOCK FLAG SET?
(1) 004254 100404 BMI B0023 ;IF SO, CONTINUE ON WITH THE TEST
(1) 004256 005203 INC R3 ;IF NOT INCREMENT THE COUNTER LOCATION
(1) 004260 001373 BNE A0023 ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
(1) 004262 104001 E0023: ERROR 1 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
(1) 004264 000410 BR T0024
(1) 004266 B0023:
478 004266 012737 000200 177546 T0023: MOV #200,LKS ;MOVE A 1 INTO THE CLOCK FLAG BIT
479 004274 023737 177546 001124 CMP LKS,$GDDAT ;SHOULD NOT AFFECT THE FLAG BIT
480 004302 001401 BEQ T0024
481 004304 104001 E10023: ERROR 1 ;CLOCK FLAG DID NOT CLEAR
482
483
484
485 .SBTTL INTERRUPT TEST
486 004306 000004 T0024: SCOPE
487 004310 012737 005674 000004 MOV #TRAP0,2#4 ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
(1) 004316 012737 000340 000006 MOV #340,2#6 ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
488 004324 012737 000200 001124 MOV #200,$GDDAT ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
489 004332 012737 004352 001106 MOV #R0024,SLPADR ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
490 004340 012737 004406 000100 MOV #E0024,100
491 004346 005037 177546 CLR LKS ;ALLOW CLOCK INTERRUPTS
492 004352 012737 000340 177776 R0024: MOV #340,2#PS ;NO INTERRUPTS NOW
493 004360 012737 000300 177546 MOV #300,LKS
494 004366 005227 000000 A0024: INC #0 ;WAIT FOR 20+ MS
495 004372 001375 BNE A0024 ;LOOP BACK IF NOT DONE WAITING
496 004374 000005 RESET ;RESET SHOULD CLEAR INTERRUPT ENABLE
497 004376 023737 001124 177546 CMP $GDDAT,LKS ;AND LEAVE THE CLOCK FLAG SET
498 004404 001401 BEQ T0025 ;GO ON TO THE NEXT TEST IF IT DID
499 004406 104001 E0024: ERROR 1 ;RESET SET INTERRUPT ENABLE OR CLEARED CLOCK FLAG
500
501
502
503 .SBTTL NO SACK TIMEOUT TEST
504 :NO SACK TIMEOUT TEST
505 004410 000004 T0025: SCOPE
506 004412 012737 005674 000004 MOV #TRAP0,2#4 ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS

```



```

(1) 004420 012737 000340 000006      MOV      #340, @#6      ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
507 004426 012737 000300 001124      MOV      #300, $GDDAT  ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
508 004434 012737 004456 001106      MOV      @R0025, $LPADR ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
509 004442 012737 000340 000102      MOV      #340, 102     ;NO INTERRUPTS AFTER THE FIRST ONE
510 004450 012737 004530 000100      MOV      @C0025, 100
511 004456 005037 177546      R0025:  CLR      LKS
512 004462 012737 000360 177776      MOV      #360, PS
513 004470 005003      CLR      R3            ;INITIALIZE A COUNTER LOCATION
(1) 004472 105737 177546      A0025:  TSTB     LKS     ;IS THE CLOCK FLAG SET?
(1) 004476 100404      BMI     B0025         ;IF SO, CONTINUE ON WITH THE TEST
(1) 004500 005203      INC     R3            ;IF NOT INCREMENT THE COUNTER LOCATION
(1) 004502 001373      BNE     A0025         ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
(1) 004504 104001      E0025:  ERROR 1      ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
(1) 004506 000415      BR      T0026
(1) 004510      B0025:
514 004510 005037 177776      CLR     PS            ;MAKE ALL INTERRUPTS OK
515 004514 012737 000100 177546      MOV     #100, LKS    ;ENABLE CLOCK INTERRUPTS
516 004522 000001      WAIT
517 004524 104006      E10025: ERROR 6      ;THE ONLY WAY TO LEAVE HERE WITHOUT ERROR IS TO INTERRUPT
518 004526 000405      BR      T0026        ;IF IT ERROR IS HERE ITS BECAUSE OF A "NO-SACK" TIMEOUT
519 004530 022737 000300 177546      C0025:  CMP     #300, LKS
520 004536 001401      BEQ     T0026
521 004540 104001      E20025: ERROR 1      ;FIND OUT WHAT THE INTERRUPT DID TO THE CLOCK STATUS REG
;IT CLEARED THE INTERRUPT ENABLE OR THE FLAG BIT

;SBTTL RESET TEST
;RESET TEST
527 004542 000004      T0026:  SCOPE
528 004544 012737 005674 000004      MOV     @TRAP0, @#4   ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
(1) 004552 012737 000340 000006      MOV     #340, @#6    ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
529 004560 012737 000200 001124      MOV     #200, $GDDAT ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
530 004566 012737 004616 001106      MOV     @R0026, $LPADR ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
531 004574 012737 000340 177776      MOV     #340, @#PS   ;NO INTERRUPTS NOW
532 004602 012737 004654 000100      MOV     @E0026, 100
533 004610 012737 000140 000102      MOV     #140, 102   ;SETUP STATUS FOR AFTER THE INTERRUPT
534 004616 005037 177546      R0026:  CLR     LKS
535 004622 012706 001000      MOV     #1000, SP    ;SETUP THE STACK
536 004626 012737 000100 177546      MOV     #100, LKS    ;SET INTERRUPT ENABLE BIT NOW
537 004634 005227 000000      A0026:  INC     #0     ;WAIT FOR CLOCK FLAG TO SET
538 004640 001375      BNE     A0026
539 004642 000005      I0026:  RESET
540 004644 023737 177546 001124      CMP     LKS, $GDDAT  ;RESET SHOULD NOT CLEAR THE FLAG
541 004652 001401      BEQ     T0027        ;FIND OUT IF ID DIT DID OR NOT
542 004654 104001      E0026:  ERROR 1      ;RESET DID NOT INITIALIZE THE LKS WORD CORRECTLY

;SBTTL CLOCK FLAG BIT TEST
;CLOCK FLAG BIT TEST
;MAKE SURE IT DOESNT CLEAR WHEN YOU TRY TO SET IT VIA A 'MOV' INSTRUCTION
549 004656 000004      T0027:  SCOPE
550 004660 012737 005674 000004      MOV     @TRAP0, @#4   ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
(1) 004666 012737 000340 000006      MOV     #340, @#6    ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE

```

```

551 004674 012737 000200 001124      MOV      #200,$GDDAT      ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
552 004702 012737 004722 001106      MOV      #R0027,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
553 004710 005037 000102                CLR      102
554 004714 012737 005000 000100      MOV      #T0030 ,100
555 004722 005037 177546      R0027:  CLR      LKS
556 004726 012737 000340 177776      MOV      #340,PS
557 004734 012706 001000                MOV      #1000,SP      ;SETUP THE STACK
558 004740 005037 001230                CLR      WORD          ;SETUP A COUNTER LOCATION TO = 0
559 004744 005237 001230      A0027:  INC      WORD
560 004750 001375                BNE     A0027          ;WASTE TIME LOOPING UNTIL THE COUNTER REACHES 0
561 004752 012737 000300 177546      MOV      #300,LKS      ;SET INTERRUPT ENABLE AND TRY TO SET THE CLOCK FLAG
562 004760 012737 000200 177546      MOV      #200,LKS      ;TRY TO SET IT AGAIN
563 004766 023737 177546 001124      CMP     LKS,$GDDAT     ;DID THE CLOCK FLAG STAY SET?
564 004774 001401                BEQ     T0030          ;IF NOT GO ON TO THE NEXT TEST
565 004776 104001                E0027:  ERROR 1        ;ERROR - MOVED A '1' INTO THE CLOCK FLAG BIT AND IT STAY
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
(1)
(1)
(1)

```

.SBTTL CLOCK FLAG AFTER INTERRUPT TEST
:SEE IF AN INTERRUPT CLEARS THE CLOCK FLAG

```

T0030:  SCOPE
571 005000 000004
572 005002 012737 005674 000004      MOV      #TRAPO,2#4     ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
(1) 005010 012737 000340 000006      MOV      #340,2#6     ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
573 005016 005037 177546                CLR      LKS          ;NO CLOCK INTERRUPTS BEFORE WE ARE READY
574 005022 012737 000100 001124      MOV      #100,$GDDAT   ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
575 005030 012737 005036 001106      MOV      #R0030,$LPADR ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
576 005036 012737 005100 000100      R0030:  MOV      #A0030,100 ;SETUP CLOCK INTERRUPT VECTOR
577 005044 005037 000102                CLR      102         ;PRIORITY LEVEL WILL ALLOW FURTHER INTERRUPTS
578 005050 005037 177776                CLR      PS
579 005054 012706 001000                MOV      #1000,SP     ;SETUP THE STACK
580 005060 005037 177546                CLR      LKS
581 005064 105037 001230                CLR     WORD          ;CLEAR OUT A COUNTER LOCATION
582 005070 012737 000100 177546      MOV      #100,LKS     ;ENABLE CLOCK INTERRUPTS NOW
583 005076 000001                WAIT
584 005100 012737 005126 000100      A0030:  MOV      #E0030,100 ;ERROR IF WE INTERRUPT AGAIN
585 005106 005037 177776                CLR      PS          ;LET INTERRUPTS HAPPEN NOW
586 005112 105237 001230      B0030:  INC     WORD     ;DO A NOTHING LOOP FOR A VERY SHORT PERIOD OF TIME
587 005116 001375                BNE     B0030        ;WE SHOULD INCREMENT TO 0 LONG BEFORE AN INTERRUPT COMES
588 005120 005037 177546                CLR      LKS
589 005124 000401                BR      T0031
590 005126 104001                E0030:  ERROR 1        ;INTERRUPT DID NOT CLEAR THE CLOCK FLAG
591
592
593
594
595
596
597
598
599
600
(1)
(1)
(1)

```

.SBTTL NO INTERRUPT AT PRIORITY 7 TEST
:TEST THAT CLOCK WILL NOT INTERRUPT WITH PROCESSR AT PRIORITY 7

```

T0031:  SCOPE
596 005130 000004
597 005132 012737 005674 000004      MOV      #TRAPO,2#4     ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
(1) 005140 012737 000340 000006      MOV      #340,2#6     ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
598 005146 012737 005154 001106      MOV      #R0031,$LPADR ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
599 005154 005037 177546      R0031:  CLR      LKS
600 005160 005003                CLR      R3          ;INITIALIZE A COUNTER LOCATION
(1) 005162 105737 177546      A0031:  TST     LKS      ;IS THE CLOCK FLAG SET?
(1) 005166 100404                BMI     B0031        ;IF SO, CONTINUE ON WITH THE TEST

```



```

(1) 005170 005203          INC      R3          ;IF NOT INCREMENT THE COUNTER LOCATION
(1) 005172 001373          BNE     A0031        ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
(1) 005174 104001          E0031:  ERROR 1      ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
(1) 005176 000427          BR      T0032
(1) 005200          B0031:
601 005200 012706 001000          MOV     #1000,SP      ;INITIALIZE THE STACK POINTER
602 005204 012737 000340 177776          MOV     #340,PS      ;SET PRIORITY 7
603 005212 012737 005174 000100          MOV     #E0031,100   ;SET UP VECTOR RETURN
604 005220 012737 000100 177546          MOV     #100,LKS     ;ENABLE INTERRUPT
605 005226 005003          CLR     R3          ;INITIALIZE A COUNTER LOCATION
(1) 005230 105737 177546          C0031:  TSTB    LKS     ;IS THE CLOCK FLAG SET?
(1) 005234 100404          BMI     D0031        ;IF SO, CONTINUE ON WITH THE TEST
(1) 005236 005203          INC     R3          ;IF NOT INCREMENT THE COUNTER LOCATION
(1) 005240 001373          BNE     C0031        ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
(1) 005242 104001          E10031: ERROR 1      ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
(1) 005244 000404          BR      T0032
(1) 005246          D0031:
606 005246 000240          NOP
607 005250 000240          NOP          ;GIVE CLOCK EXTRA TIME TO INTERRUPT
608 005252 000401          BR      T0032
609 005254 104003          E20031: ERROR 3      ;ERROR, CLOCK INTERRUPTED WITHOUT HAVING PRIORITY
610
611
612
613          .SBTTL  CC PUSH TEST FOR CLOCK INTERRUPTS
614          ;TEST THAT CLOCK INTERRUPT PUSHES CONDITION CODES ONTO STACK
615 005256 000004          T0032:  SCOPE
616 005260 012737 005674 000004          MOV     #TRAP0,3#4   ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
(1) 005266 012737 000340 000006          MOV     #340,3#6     ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
617 005274 012737 005302 001106          MOV     #R0032,SLPADR ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
618 005302 005037 177546          R0032:  CLR     LKS
619 005306 005003          CLR     R3          ;INITIALIZE A COUNTER LOCATION
(1) 005310 105737 177546          A0032:  TSTB    LKS     ;IS THE CLOCK FLAG SET?
(1) 005314 100404          BMI     B0032        ;IF SO, CONTINUE ON WITH THE TEST
(1) 005316 005203          INC     R3          ;IF NOT INCREMENT THE COUNTER LOCATION
(1) 005320 001373          BNE     A0032        ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
(1) 005322 104001          E0032:  ERROR 1      ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
(1) 005324 000432          BR      T0033
(1) 005326          B0032:
620 005326 012706 001000          MOV     #1000,SP      ;INITIALIZE THE STACK POINTER
621 005332 005037 000776          CLR     BUF1
622 005336 005037 000774          CLR     BUF2
623 005342 012737 005370 000100          MOV     #C0032,100   ;SET UP VECTOR RETURN
624 005350 012737 000100 177546          MOV     #100,LKS     ;ENABLE INTERRUPT
625 005356 012737 000200 177776          MOV     #200,PS      ;SET PRIORITY 4
626 005364 000277          +SEC!SEV!SEZ!SEN    ;SET ALL CONDITION CODES
627 005366 000001          WAIT
628 005370 022737 000217 000776          C0032:  CMP     #217,BUF1
629 005376 001405          BEQ     T0033
630 005400 012737 000017 001124          MOV     #17,$GDDAT
631 005406 104007          ERROR 7
632 005410 000722          BR      T0032
633
634

```



```

(1) 005606 012737          MOV      (PC)+,2(PC)+      ;RESTORE COUNTER
(1) 005610 000001 SENDCT: .WORD      1
(1) 005612 005602          SEOPCT
(1) 005614 104400 005654          TYPE      SENDMG          ;TYPE "END PASS #"
(2) 005620 013746 001100          MOV      $PASS,-(SP)      ;SAVE $PASS FOR TYPEOUT
(2) 005624 104410          TYPDS          ;GO TYPE--DECIMAL ASCII WITH SIGN
(1) 005626 104400 005671          TYPE      SENULL          ;TYPE A NULL CHARACTER
(1) 005632 013700 000042 SGET42: MOV      @#42,R0      ;GET MONITOR ADDRESS
(1) 005636 001404          BEQ      SDOAGN          ;IF NONE
(1) 005640 004710 SENDAD: JSR      PC,(R0)      ;GO TO MONITOR
(1) 005642 000240          NOP
(1) 005644 000240          NOP          ;SAVE ROOM
(1) 005646 000240          NOP          ;FOR
(1) 005650 000137 001550 SDOAGN: JMP      @#T0001      ;ACT11
(1) 005654 005015 047105 020104 SENDMG: .ASCIZ  <15><12>/END PASS #/ ;RETURN
(1) 005662 040520 051523 021440
(1) 005670 000
(1) 005671 377 377 000 SENULL: .BYTE  -1,-1,0      ;NULL CHARACTER STRING
664 005674 005046 TRAPO: CLR      -(SP)
665 005676 004737 006216          JSR      PC,$TYPE          ;PRINTOUT "TRAPPED TO 4 FROM "
666 005702 007661          TRPMES          ;ADDRESS OF THE MESSAGE
667 005704 011646          MOV      (SP),-(SP)      ;GET THE ADDRESS WHERE THE TRAP OCCURED
668 005706 162716 000002          SUB      #2,(SP)          ;MAKE IT RIGHT
669 005712 104402          TYPDC          ;TYPE OUT ADDRESS IN OCTAL
670 005714 104400 001225          TYPE      ,SCLF          ;PRINTOUT A CARRIAGE RETURN-LINE FEED
671 005720 005046          CLR      -(SP)
672 005722 004737 006216          JSR      PC,$TYPE          ;PRINTOUT RESTARTING MESSAGE
673 005726 007722          TRPM2S          ;ADDRESS OF RESTART MESSAGE
674 005730 000240          NOP
675 005732 000240          NOP
676 005734 000240          NOP
677 005736 000240          NOP
678 005740 000005          RESET
679 005742 000137 001416          JMP      START
680
;*****
(1)
(1) .SBTTL SCOPE HANDLER ROUTINE
(1)
(1) :#SM14=1 LOOP ON TEST
(1) :#SM11=1 INHIBIT ITERATIONS
(1) :#SM09=1 LOOP ON ERROR
(1) :#SM08=1 LOOP ON TEST IN SMR<7:0>
(1) :#THE TEST NUMBER ($STSTNM) IS INCREMENTED AND DISPLAYED IN DISPLAY<7:0>
(1) :#AND THE ERROR FLAG ($ERFLG) IS DISPLAYED IN DISPLAY<15:08>
(1)
(1) $SCOPE:
(2) 005746 000240          NOP
(1) 005750 006137 177570          ROL      @#SMR          ;LOOP ON PRESENT TEST?
(1) 005754 100511          BMI      $OVER          ;YES IF SM14=1
(1) :####START OF CODE FOR THE XOR TESTER####
(1) 005756 000416          SXTSTR: BR      65          ;IF RUNNING ON THE "XOR" TESTER CHANGE
(1) :THIS INSTRUCTION TO A "NOP" (NOP=240)
(1) 005760 013746 000004          MOV      @#ERRVEC,-(SP)      ;SAVE THE CONTENTS OF THE ERROR VECTOR
(1) 005764 012737 006004 000004          MOV      #55,@#ERRVEC      ;SET FOR TIMEOUT

```



```

(1) 005772 005737 177060      TST      @#177060      ;TIME OUT ON XOR?
(1) 005776 012637 000004      MOV      (SP)+,@#ERRVEC ;RESTORE THE ERROR VECTOR
(1) 006002 000463              BR      $SVLAD        ;GO TO THE NEXT TEST
(1) 006004 022626              5S:     CMP      (SP)+,(SP)+ ;CLEAR THE STACK AFTER A TIME OUT
(1) 006006 012637 000004      MOV      (SP)+,@#ERRVEC ;RESTORE THE ERROR VECTOR
(1) 006012 000423              BR      7S           ;LOOP ON THE PRESENT TEST
(1) 006014              6S:     ;#####END OF CODE FOR THE XOR TESTER#####
(1) 006014 032737 000400 177570      BIT      #SM08,@#SMR   ;LOOP ON SPEC. TEST?
(1) 006022 001404              BEQ     2S           ;BR IF NO
(1) 006024 123737 177570 001102      CMPB    @#SMR,STSTNM  ;ON THE RIGHT TEST? SMR<7:0>
(1) 006032 001462              BEQ     $OVER        ;BR IF YES
(1) 006034 105737 001103              2S:     TSTB    SERFLG    ;HAS AN ERROR OCCURRED?
(1) 006040 001421              BEQ     3S           ;BR IF NO
(1) 006042 123737 001115 001103      CMPB    SERMAX,SERFLG ;MAX. ERRORS FOR THIS TEST OCCURRED?
(1) 006050 101015              BHI     3S           ;BR IF NO
(1) 006052 032737 001000 177570      BIT      #SM09,@#SMR   ;LOOP ON ERROR?
(1) 006060 001404              BEQ     4S           ;BR IF NO
(1) 006062 013737 001110 001106      7S:     MOV      $LPERR,$LPADR ;SET LOOP ADDRESS TO LAST SCOPE
(1) 006070 000443              BR      $OVER
(1) 006072 105037 001103              4S:     CLRB    SERFLG    ;ZERO THE ERROR FLAG
(1) 006076 005037 001214              CLR     $TIMES       ;CLEAR THE NUMBER OF ITERATIONS TO MAKE
(1) 006102 000415              BR      1S           ;ESCAPE TO THE NEXT TEST
(1) 006104 032737 004000 177570      3S:     BIT      #SM11,@#SMR  ;INHIBIT ITERATIONS?
(1) 006112 001011              BNE     1S           ;BR IF YES
(1) 006114 005737 001100              TST     $PASS        ;IF FIRST PASS OF PROGRAM
(1) 006120 001406              BEQ     1S           ;INHIBIT ITERATIONS
(1) 006122 005237 001104              INC     $ICNT        ;INCREMENT ITERATION COUNT
(1) 006126 023737 001214 001104      CMP     $TIMES,$ICNT ;CHECK THE NUMBER OF ITERATIONS MADE
(1) 006134 002021              BGE     $OVER        ;BR IF MORE ITERATION REQUIRED
(1) 006136 012737 000001 001104      1S:     MOV     @1,$ICNT    ;REINITIALIZE THE ITERATION COUNTER
(1) 006144 013737 006214 001214      MOV     $SMXCNT,$TIMES ;SET NUMBER OF ITERATIONS TO DO
(1) 006152 105237 001102      $SVLAD: INCB    $STSTNM   ;COUNT TEST NUMBERS
(1) 006156 011637 001106      MOV     (SP),$LPADR   ;SAVE SCOPE LOOP ADDRESS
(1) 006162 011637 001110      MOV     (SP),$LPERR  ;SAVE ERROR LOOP ADDRESS
(1) 006166 005037 001216      CLR     $ESCAPE      ;CLEAR THE ESCAPE FROM ERROR ADDRESS
(1) 006172 112737 000001 001115      MOVB   @1,SERMAX     ;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
(1) 006200 013737 001102 177570      $OVER:  MOV     $STSTNM,@#DISPLAY ;DISPLAY TEST NUMBER
(1) 006206 013716 001106      MOV     $LPADR,(SP)  ;FUDGE RETURN ADDRESS
(1) 006212 000002              RTI
(1) 006214 000010      $MXCNT: 10          ;FIXES PS
(1) 681 ;#####
(1) ;.SBTTL TYPE ROUTINE
(1) ;#ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
(1) ;#THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
(1) ;#NOTE1: SNUL contains the character to be used as the filler character.
(1) ;#NOTE2: SFILLS contains the number of filler characters required.
(1) ;#NOTE3: SFILLC contains the character to fill after.
(1) ;#
(1) ;#CALL:
(1) ;#1) USING A TRAP INSTRUCTION
(1) ;# TYPE ,MESADR ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
(1) ;#OR

```



```

(1)      ;#      TYPE
(1)      ;#      MESADR
(1)      ;#
(1)      ;#2) USING A JSR INSTRUCTION
(1)      ;#      MOV      PS,-(SP)      ;PUSH PROCESSOR STATUS WORD ON THE STACK
(1)      ;#      JSR      PC,STYPE      ;CALL TYPE ROUTINE
(1)      ;#      MESADDR      ;FIRST ADDRESS OF MESSAGE
(1)
(1) 006216 105737 001151 STYPE: TSTB STPFLG      ;IS THERE A TERMINAL?
(1) 006222 100002      BPL      1$      ;BR IF YES
(1) 006224 000000      HALT      ;HALT HERE IF NO TERMINAL
(1) 006226 000407      BR      3$      ;LEAVE
(1) 006230 010046      1$: MOV      RO,-(SP)      ;SAVE RO
(1) 006232 017600 000002      MOV      @2(SP),RO      ;GET ADDRESS OF ASCIZ STRING
(1) 006236 112046      2$: MOVB      (RO)+,-(SP)      ;PUSH CHARACTER TO BE TYPED ONTO STACK
(1) 006240 001005      BNE      4$      ;BR IF IT ISN'T THE TERMINATOR
(1) 006242 005726      TST      (SP)+      ;IF TERMINATOR POP IT OFF THE STACK
(1) 006244 012600      MOV      (SP)+,RO      ;RESTORE RO
(1) 006246 062716 000002      3$: ADD      #2,(SP)      ;ADJUST RETURN PC
(1) 006252 000002      RTI      ;RETURN
(1) 006254 004737 006306      4$: JSR      PC,7$      ;GO TYPE THIS CHARACTER
(1) 006260 123726 001150      5$: CMPB      $FILLC,(SP)+      ;IS IT TIME FOR FILLER CHARS.?
(1) 006264 001364      BNE      2$      ;IF NO GO GET NEXT CHAR.
(1) 006266 013746 001146      MOV      $NULL,-(SP)      ;GET # OF FILLER CHARS. NEEDED
(1)      ;AND THE NULL CHAR.
(1) 006272 105366 000001      6$: DECB      1(SP)      ;DOES A NULL NEED TO BE TYPED?
(1) 006276 002770      BLT      5$      ;BR IF NO--GO POP THE NULL OFF OF STACK
(1) 006300 004737 006306      JSR      PC,7$      ;GO TYPE A NULL
(1) 006304 000772      BR      6$      ;LOOP
(1) 006306 105777 172630      7$: TSTB      @STPS      ;WAIT UNTIL PRINTER IS READY
(1) 006312 100375      BPL      7$
(1) 006314 116677 000002 172622      MOVB      2(SP),@STPB      ;LOAD CHAR TO BE TYPED INTO DATA REG.
(1) 006322 000207      RTS      PC
682 ;;*****
(1)
(1) .SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
(1)
(1) ;#CALL:
(1) ;#      MOV      NUM,-(SP)      ;PUT THE BINARY NUMBER ON THE STACK
(1) ;#      TYPDS      ;GO TO THE ROUTINE
(1)
(1) STYPDS:
(2) 006324      MOV      RO,-(SP)      ;PUSH RO ON STACK
(3) 006324 010046      MOV      R1,-(SP)      ;PUSH R1 ON STACK
(3) 006326 010146      MOV      R2,-(SP)      ;PUSH R2 ON STACK
(3) 006330 010246      MOV      R3,-(SP)      ;PUSH R3 ON STACK
(3) 006332 010346      MOV      R5,-(SP)      ;PUSH R5 ON STACK
(3) 006334 010546      MOV      R5,-(SP)      ;PUSH R5 ON STACK
(1) 006336 012746 020200      MOV      #20200,-(SP)      ;SET BLANK SWITCH AND SIGN
(1) 006342 016605 000020      MOV      20(SP),R5      ;GET THE INPUT NUMBER
(1) 006346 100004      BPL      1$      ;BR IF INPUT IS POS.
(1) 006350 005405      NEG      R5      ;MAKE THE BINARY NUMBER POS.
(1) 006352 112766 000055 000001      1$: MOVB      #'-,1(SP)      ;MAKE THE ASCII NUMBER NEG.
(1) 006360 005000      CLR      RO      ;ZERO THE CONSTANTS INDEX
(1) 006362 012703 006540      MOV      #SDBLK,R3      ;SETUP THE OUTPUT POINTER

```

```

(1) 006366 112723 000040          MOVB      #' ,(R3)+      ;SET THE FIRST CHARACTER TO A BLANK
(1) 006372 005002          25: CLR      R2          ;CLEAR THE BCD NUMBER
(1) 006374 016001 006530          MOV      $DTBL(R0),R1  ;GET THE CONSTANT
(1) 006400 160105          35: SUB      R1,R5      ;FORM THIS BCD DIGIT
(1) 006402 002402          BLT      45           ;BR IF DONE
(1) 006404 005202          INC      R2          ;INCREASE THE BCD DIGIT BY 1
(1) 006406 000774          BR       35
(1) 006410 060105          45: ADD      R1,R5      ;ADD BACK THE CONSTANT
(1) 006412 005702          TST      R2          ;CHECK IF BCD DIGIT=0
(1) 006414 001002          BNE      55           ;FALL THROUGH IF 0
(1) 006416 105716          TSTB     (SP)        ;STILL DOING LEADING 0'S?
(1) 006420 100407          BMI      75           ;BR IF YES
(1) 006422 106316          55: ASLB     (SP)        ;MSD?
(1) 006424 103003          BCC      65           ;BR IF NO
(1) 006426 116663 000001 177777  MOVB      1(SP),-1(R3)  ;YES--SET THE SIGN
(1) 006434 052702 000060          65: BIS      #'0,R2      ;MAKE THE BCD DIGIT ASCII
(1) 006440 052702 000040          75: BIS      #' ,R2      ;MAKE IT A SPACE IF NOT ALREADY A DIGIT
(1) 006444 110223          MOVB      R2,(R3)+    ;PUT THIS CHARACTER IN THE OUTPUT BUFFER
(1) 006446 005720          TST      (R0)+       ;JUST INCREMENTING
(1) 006450 020027 000010          CMP      R0,#10      ;CHECK THE TABLE INDEX
(1) 006454 002746          BLT      25           ;GO DO THE NEXT DIGIT
(1) 006456 003002          BGT      85           ;GO TO EXIT
(1) 006460 010502          MOV      R5,R2       ;GET THE LSD
(1) 006462 000764          BR       65           ;GO CHANGE TO ASCII
(1) 006464 105726          85: TSTB     (SP)+      ;WAS THE LSD THE FIRST NON-ZERO?
(1) 006466 100003          BPL      95           ;BR IF NO
(1) 006470 116663 177777 177776  95: MOVB      -1(SP),-2(R3) ;YES--SET THE SIGN FOR TYPING
(1) 006476 105013          CLRB     (R3)        ;SET THE TERMINATOR
(3) 006500 012605          MOV      (SP)+,R5    ;POP STACK INTO R5
(3) 006502 012603          MOV      (SP)+,R3    ;POP STACK INTO R3
(3) 006504 012602          MOV      (SP)+,R2    ;POP STACK INTO R2
(3) 006506 012601          MOV      (SP)+,R1    ;POP STACK INTO R1
(3) 006510 012600          MOV      (SP)+,R0    ;POP STACK INTO R0
(1) 006512 104400 006540          TYPE     $DBLK        ;NOW TYPE THE NUMBER
(1) 006516 016666 000002 000004  MOV      2(SP),4(SP)  ;ADJUST THE STACK
(1) 006524 012616          MOV      (SP)+,(SP)
(1) 006526 000002          RTI
(1) 006530 023420          $DTBL: 10000.
(1) 006532 001750          1000.
(1) 006534 000144          100.
(1) 006536 000012          10.
(1) 006540 000004          $DBLK: .BLKM 4
683 ;*****
(1) ;.SBTTL BINARY TO OCTAL (ASCII) AND TYPE
(1) ;#STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
(1) ;#CALL:
(1) ;# MOV      NUM,-(SP) ;NUMBER TO BE TYPED
(1) ;# TYPOS ;CALL FOR TYPEOUT
(1) ;# .BYTE N ;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
(1) ;# .BYTE M ;M=1 OR 0
(1) ;# ;1=TYPE LEADING ZEROS
(1) ;# ;0=SUPPRESS LEADING ZEROS

```


E05

```

(1)      ;*
(1)      ;*STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
(1)      ;*STYPOS OR STYPOC
(1)      ;*CALL:
(1)      ;*      MOV      NUM,-(SP)          ;NUMBER TO BE TYPED
(1)      ;*      TYPON          ;CALL FOR TYPEOUT
(1)      ;*
(1)      ;*STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
(1)      ;*CALL:
(1)      ;*      MOV      NUM,-(SP)          ;NUMBER TO BE TYPED
(1)      ;*      TYPOC          ;CALL FOR TYPEOUT
(1)      ;*
(1) 006550 017646 000000          STYPOS: MOV      0(SP),-(SP)          ;PICKUP THE MODE
(1) 006554 116637 000001 006773  MOVB     1(SP),SOFILL        ;LOAD ZERO FILL SWITCH
(1) 006562 112637 006775          MOVB     (SP)+,SOMODE+1      ;NUMBER OF DIGITS TO TYPE
(1) 006566 062716 000002          ADD      02,(SP)          ;ADJUST RETURN ADDRESS
(1) 006572 000406                   BR       STYPON
(1) 006574 112737 000001 006773  STYPOC: MOVB     01,SOFILL        ;SET THE ZERO FILL SWITCH
(1) 006602 112737 000006 006775  MOVB     06,SOMODE+1      ;SET FOR SIX(6) DIGITS
(1) 006610 112737 000005 006772  STYPON: MOVB     05,SOCNT        ;SET THE ITERATION COUNT
(1) 006616 010346                   MOV      R3,-(SP)          ;SAVE R3
(1) 006620 010446                   MOV      R4,-(SP)          ;SAVE R4
(1) 006622 010546                   MOV      R5,-(SP)          ;SAVE R5
(1) 006624 113704 006775          MOVB     SOMODE+1,R4      ;GET THE NUMBER OF DIGITS TO TYPE
(1) 006630 005404                   NEG      R4
(1) 006632 062704 000006          ADD      06,R4            ;SUBTRACT IT FOR MAX. ALLOWED
(1) 006636 110437 006774          MOVB     R4,SOMODE        ;SAVE IT FOR USE
(1) 006642 113704 006773          MOVB     SOFILL,R4        ;GET THE ZERO FILL SWITCH
(1) 006646 016605 000012          MOV      12(SP),R5       ;PICKUP THE INPUT NUMBER
(1) 006652 005003                   CLR      R3                ;CLEAR THE OUTPUT WORD
(1) 006654 006105                   1S:    ROL      R5          ;ROTATE MSB INTO "C"
(1) 006656 000404                   BR       3S                ;GO DO MSB
(1) 006660 006105                   2S:    ROL      R5          ;FORM THIS DIGIT
(1) 006662 006105                   ROL      R5
(1) 006664 006105                   ROL      R5
(1) 006666 010503                   MOV      R5,R3
(1) 006670 006103                   3S:    ROL      R3          ;GET LSB OF THIS DIGIT
(1) 006672 105337 006774          DECB     SOMODE           ;TYPE THIS DIGIT?
(1) 006676 100016                   BPL      7S                ;BR IF NO
(1) 006700 042703 177770          BIC      #177770,R3       ;GET RID OF JUNK
(1) 006704 001002                   BNE     4S                ;TEST FOR 0
(1) 006706 005704                   TST     R4                ;SUPPRESS THIS 0?
(1) 006710 001403                   BEQ     5S                ;BR IF YES
(1) 006712 005204                   4S:    INC     R4          ;DON'T SUPPRESS ANYMORE 0'S
(1) 006714 052703 000060          BIS     #'0,R3           ;MAKE THIS DIGIT ASCII
(1) 006720 052703 000040          5S:    BIS     #' ,R3       ;MAKE ASCII IF NOT ALREADY
(1) 006724 110337 006770          MOVB     R3,R5           ;SAVE FOR TYPING
(1) 006730 104400 006770          TYPE     05             ;GO TYPE THIS DIGIT
(1) 006734 105337 006772          7S:    DECB     SOCNT        ;COUNT BY 1
(1) 006740 003347                   BGT     2S                ;BR IF MORE TO DO
(1) 006742 002402                   BLT     6S                ;BR IF DONE
(1) 006744 005204                   INC     R4                ;INSURE LAST DIGIT ISN'T A BLANK
(1) 006746 000744                   BR      2S                ;GO DO THE LAST DIGIT
(1) 006750 012605          6S:    MOV      (SP)+,R5    ;RESTORE R5

```


F05

```

(1) 006752 012604      MOV      (SP)+,R4      ;RESTORE R4
(1) 006754 012603      MOV      (SP)+,R3      ;RESTORE R3
(1) 006756 016666 000002 000004  MOV      2(SP),4(SP)  ;SET THE STACK FOR RETURNING
(1) 006764 012616      MOV      (SP)+,(SP)
(1) 006766 000002      RTI                ;RETURN
(1) 006770      000      BS:      .BYTE      0      ;STORAGE FOR ASCII DIGIT
(1) 006771      000      .BYTE      0      ;TERMINATOR FOR TYPE ROUTINE
(1) 006772      000      SOCNT:   .BYTE      0      ;OCTAL DIGIT COUNTER
(1) 006773      000      $OFILL: .BYTE      0      ;ZERO FILL SWITCH
(1) 006774 000000      $ONODE: 0            ;NUMBER OF DIGITS TO TYPE
684 ;*****
(1)
(1) .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
(1)
(1) ;#THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
(1) ;#ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
(1) ;#AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
(1)
(1) $ERRTYP:
(1) 006776 104400 001225      TYPE      $SCRLF      ;"CARRIAGE RETURN" & "LINE FEED"
(1) 007002 010046      MOV      R0,-(SP)     ;SAVE R0
(1) 007004 005000      CLR      R0           ;PICKUP THE ITEM INDEX
(1) 007006 153700 001114      BISB     @($ITEMB,R0
(1) 007012 001004      BNE      IS          ;IF ITEM NUMBER IS ZERO, JUST
(1)
(2) 007014 013746 001116      MOV      $ERRPC,-(SP) ;TYPE THE PC OF THE ERROR
(2)
(1) 007020 104402      TYPOC     ;SAVE $ERRPC FOR TYPEOUT
(1) 007022 000426      BR       6$         ;ERROR ADDRESS
(1) 007024 005300      1$:      DEC      R0       ;GO TYPE--OCTAL ASCII(ALL DIGITS)
(1) 007026 006300      ASL      R0       ;GET OUT
(1) 007030 006300      ASL      R0       ;ADJUST THE INDEX SO THAT IT WILL
(1) 007032 006300      ASL      R0       ;WORK FOR THE ERROR TABLE
(1) 007034 062700 001232      ADD      @($ERRTB,R0
(1) 007040 012037 007050      MOV      (R0)+,2$
(1) 007044 001404      BEQ      3$
(1) 007046 104400      TYPE     ;FORM TABLE POINTER
(1) 007050 000000      .WORD   0          ;PICKUP "ERROR MESSAGE" POINTER
(1) 007052 104400 001225      TYPE     $SCRLF    ;"CARRIAGE RETURN" & "LINE FEED"
(1) 007056 012037 007066      MOV      (R0)+,4$   ;PICKUP "DATA HEADER" POINTER
(1) 007062 001404      BEQ      5$
(1) 007064 104400      TYPE     ;SKIP TYPEOUT IF 0
(1) 007066 000000      .WORD   0          ;TYPE THE "DATA HEADER"
(1) 007070 104400 001225      TYPE     $SCRLF    ;"DATA HEADER" POINTER GOES HERE
(1) 007074 011000      MOV      (R0),R0    ;"CARRIAGE RETURN" & "LINE FEED"
(1) 007076 001004      BNE      7$
(1) 007100 012600 001225      MOV      (SP)+,R0   ;PICKUP "DATA TABLE" POINTER
(1) 007102 104400      TYPE     $SCRLF    ;GO TYPE THE DATA
(1) 007106 000207      RTS      PC        ;RESTORE R0
(2) 007110      000      7$:      .BYTE      0      ;"CARRIAGE RETURN" & "LINE FEED"
(2) 007110 013046      MOV      @2(R0)+,-(SP) ;SAVE @2(R0)+ FOR TYPEOUT
(2) 007112 104402      TYPOC     ;GO TYPE--OCTAL ASCII(ALL DIGITS)
(1) 007114 005710      TST      (R0)       ;IS THERE ANOTHER NUMBER?
(1) 007116 001770      BEQ      6$        ;BR IF NO

```

```

(1) 007120 104400 007126          TYPE      BS          ;TYPE TWO(2) SPACES
(1) 007124 000771                BR        7S          ;LOOP
(1) 007126 020040 000           BS:      .ASCIZ  / /      ;TWO(2) SPACES
(1) 007132 007132                .EVEN
693 ;*****
(1)
(1) .SBTTL  ERROR HANDLER ROUTINE
(1)
(1) ;#SM15=1      HALT ON ERROR
(1) ;#SM13=1      INHIBIT ERROR TYPEOUTS
(1) ;#SM10=1     BELL ON ERROR
(1) ;#SM09=1     LOOP ON ERROR
(1) ;#GO TO SERRTYP ON ERROR
(1)
(1) 007132          ERROR:
(3) 007132 010637 001170          MOV      SP, SREG6      ;GET THE CURRENT STACK POINTER VALUE
(3) 007136 162737 000004 001170  SUB      #4, SREG6     ;RESTORE IT TO ITS "PRE ERROR TRAP" VALUE FOR PR
(3) 007144 016637 000002 001172  MOV      2(SP), SREG7  ;GET THE PS OFF OF THE STACK
(3) 007152 005037 001166          CLR      SREG5        ;PREPARE "SREG5" TO HOLD THE TEST #
(3) 007156 113737 001102 001166  MOVB    STSTNM, SREG5 ;TEST # IS HELD IN THE LOW BYTE OF "TSTNM"
(3) 007164 010037 001154          MOV      R0, SREG0    ;MOST OF THE TIME R0 HAS GOOD STUFF IN IT ALSO
(1) 007170 105237 001103 7S:    INCB    SERFLG        ;SET THE ERROR FLAG
(1) 007174 001775          BEQ     7S           ;DON'T LET THE FLAG GO TO ZERO
(1) 007176 013737 001102 177570  MOV     STSTNM, @DISPLAY ;DISPLAY TEST NUMBER AND ERROR FLAG
(1) 007204 032737 002000 177570  BIT     #SM10, @SMR    ;BELL ON ERROR?
(1) 007212 001402          BEQ     1S           ;NO - SKIP
(1) 007214 104400 001220          TYPE    SBELL        ;RING BELL
(1) 007220 005237 001112 1S:    INC     SERTTL       ;COUNT THE NUMBER OF ERRORS
(1) 007224 011637 001116          MOV     (SP), SERRPC  ;GET ADDRESS OF ERROR INSTRUCTION
(1) 007230 162737 000002 001116  SUB     #2, SERRPC
(1) 007236 117737 171654 001114  MOVB    @SERRPC, $ITEMB ;STRIP AND SAVE THE ERROR ITEM CODE
(1) 007244 032737 020000 177570  BIT     #SM13, @SMR   ;SKIP TYPEOUT IF SET
(1) 007252 001004          BNE     2S           ;SKIP TYPEOUTS
(1) 007254 004737 006776          JSR    PC, @SERRTYP  ;GO TO USER ERROR ROUTINE
(1) 007260 104400 001225          TYPE    SCRLF
(1) 007264 005737 177570 2S:    TST    @SMR
(1) 007270 100001          BPL     3S           ;HALT ON ERROR
(1) 007272 000000          HALT
(1) 007274 032737 001000 177570 3S:    BIT     #SM09, @SMR  ;SKIP IF CONTINUE
(1) 007302 001402          BEQ     4S           ;HALT ON ERROR!
(1) 007304 013716 001110          MOV     SLPERR, (SP) ;LOOP ON ERROR SWITCH SET?
(1) 007310 005737 001216 4S:    TST    $ESCAPE
(1) 007314 001402          BEQ     5S           ;BR IF NO
(1) 007316 013716 001216          MOV     $ESCAPE, (SP);FLDGE RETURN FOR LOOPING
(1) 007322 000002          RTI
(1) ;*****
(1)
(1) .SBTTL  TRAP DECODER
(1)
(1) ;#THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
(1) ;#AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
(1) ;#OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
(1) ;#GO TO THAT ROUTINE.

```



```

(1) 007324 010046 STRAP: MOV R0,-(SP) ;SAVE R0
(1) 007326 016600 000002 MOV 2(SP),R0 ;GET TRAP ADDRESS
(1) 007332 005740 TST -(R0) ;BACKUP BY 2
(1) 007334 111000 MOVB (R0),R0 ;GET RIGHT BYTE OF TRAP
(1) 007336 016000 007344 MOV STRPAD(R0),R0 ;INDEX TO TABLE
(1) 007342 000200 RTS R0 ;GO TO ROUTINE

```

.SBTTL TRAP TABLE

;;THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
;;BY THE "TRAP" INSTRUCTION.

ROUTINE

```

(3) 007344 STRPAD:
(3) 007344 006216 STYPE ;CALL=TYPE TRAP+0(104400) TTY TYPEOUT ROUTINE
(3) 007346 006574 STYPOC ;CALL=TYPOC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING
(3) 007350 006550 STYPOS ;CALL=TYPOS TRAP+4(104404) TYPE OCTAL NUMBER (NO LEADING ZE
(3) 007352 006610 STYPON ;CALL=TYPON TRAP+6(104406) TYPE OCTAL NUMBER (AS PER LAST C
(3) 007354 006324 STYPDS ;CALL=TYPDS TRAP+10(104410) TYPE DECIMAL NUMBER (WITH SIGN)

```

;;*****

.SBTTL POWER DOWN AND UP ROUTINES

:POWER DOWN ROUTINE

```

(1) 007356 012737 007520 000024 $PWRDN: MOV $SILLUP,3$PWRVEC ;SET FOR FAST UP
(1) 007364 012737 000340 000026 MOV $340,3$PWRVEC+2 ;PRIO:7
(3) 007372 010046 MOV R0,-(SP) ;PUSH R0 ON STACK
(3) 007374 010146 MOV R1,-(SP) ;PUSH R1 ON STACK
(3) 007376 010246 MOV R2,-(SP) ;PUSH R2 ON STACK
(3) 007400 010346 MOV R3,-(SP) ;PUSH R3 ON STACK
(3) 007402 010446 MOV R4,-(SP) ;PUSH R4 ON STACK
(3) 007404 010546 MOV R5,-(SP) ;PUSH R5 ON STACK
(3) 007406 013746 007746 MOV POMPUS,-(SP) ;PUSH POMPUS ON STACK
(3) 007412 013746 007750 MOV POMPOP,-(SP) ;PUSH POMPOP ON STACK
(3) 007416 013746 007536 MOV POMMES,-(SP) ;PUSH POMMES ON STACK
(3) 007422 013746 001550 MOV T0001,-(SP) ;PUSH T0001 ON STACK
(1) 007426 010637 007524 MOV SP,$SAVR6 ;SAVE SP
(1) 007432 012737 007444 000024 MOV $SPWRUP,3$PWRVEC ;SET UP VECTOR
(1) 007440 000000 HALT
(1) 007442 000776 BR -2 ;HANG UP

```

:POWER UP ROUTINE

```

(1) 007444 013706 007524 $PWRUP: MOV $SAVR6,SP ;GET SP
(1) 007450 005037 007524 CLR $SAVR6 ;WAIT LOOP FOR THE TTY
(1) 007454 005237 007524 IS: INC $SAVR6 ;WAIT FOR THE INC
(1) 007460 001375 BNE IS ;OF <POMPUS>, <POMPOP>, <POMMES>, <T0001> WORD
(3) 007462 012605 MOV (SP)+,R5 ;POP STACK INTO R5
(3) 007464 012604 MOV (SP)+,R4 ;POP STACK INTO R4
(3) 007466 012603 MOV (SP)+,R3 ;POP STACK INTO R3
(3) 007470 012602 MOV (SP)+,R2 ;POP STACK INTO R2
(3) 007472 012601 MOV (SP)+,R1 ;POP STACK INTO R1
(3) 007474 012600 MOV (SP)+,R0 ;POP STACK INTO R0

```



```

(1) 007476 012737 007356 000024      MOV      #SPWRDN,2#PMRVEC      ;SET UP THE POWER DOWN VECTOR
(1) 007504 012737 000340 000026      MOV      #340,2#PMRVEC+2    ;PRIO:7
(1) 007512 104400 007526              TYPE      ,SPWR             ;POWER FAIL MESSAGE
(1) 007516 000002                      RTI
(1) 007520 000000                      SILLUP: HALT                ;THE POWER UP SEQUENCE WAS STARTED
(1) 007522 000776                      BR      .-2                  ;BEFORE THE POWER DOWN WAS COMPLETE
(1) 007524 000000                      $SAVR6: 0                    ;PUT THE SP HERE
(1) 007526 005015 047520 042527      SPWR:    .ASCIZ <15><12>"POWER"
(1) 007534 000122                      .EVEN
696 007536 005015 042522 052123      POMMES: .ASCIZ <15> <12> "RESTARTING AFTER A POWER FIALURE" <15> <12> <12>
      007544 051101 044524 043516
      007552 040440 052106 051105
      007560 040440 050040 053517
      007566 051105 043040 040511
      007574 052514 042522 005015
      007602 000012
697 007604 005015 046412 026504      STMES:  .ASCIZ <15><12><12>"MD-11-DZKMA-D LINE FREQUENCY CLOCK TEST"<15><12>
      007612 030461 042055 045532
      007620 040527 042055 046040
      007626 047111 020105 051106
      007634 050505 042525 041516
      007642 020131 046103 041517
      007650 020113 042524 052123
      007656 005015      000
698
699 007661      124 040522 050120      TRPMES: .ASCIZ "TRAPPED TO LOC 4 FROM LOCATION "
      007666 042105 052040 020117
      007674 047514 020103 020064
      007702 051106 046517 046040
      007710 041517 052101 047511
      007716 020116 000040
700 007722 042522 052123 051101      TRPM2S: .ASCIZ "RESTARTING PROGRAM"
      007730 044524 043516 050040
      007736 047522 051107 046501
      007744      000
701
702 007746 001230                      .EVEN
703 007750 001230      POMPU:  WORD
704 007752 020040 020040 020040      POMPO:  WORD
      007760 020040 020040 020040      EN1:    .ASCIZ "                LKS    LKS    "
      007766 020040 020040 020040
      007774 020040 020040 020040
      010002 020040 020040 020040
      010010 020040 045514 020123
      010016 020040 020040 045514
      010024 020123 020040 020040
      010032      000
705 010033      050 041520 020051      DH1:    .ASCIZ "(PC) (PS) (SP) TEST# WAS S/B "
      010040 020040 024040 051520
      010046 020051 020040 024040
      010054 050123 020051 020040
      010062 052040 051505 021524
      010070 020040 053440 051501

```

	010076	020040	020040	051440	
	010104	041057	020040	020040	
	010112	000040			
706					.EVEN
707	010114	001116	001172	001170	DT1: SERRPC, SREG7, SREG6, SREG5, LKS, SGDOAT
	010122	001166	177546	001124	
708	010130	000000			0
709	010132	046103	041517	020113	EM2: .ASCIZ "CLOCK FAILED TO INTERRUPT"
	010140	040506	046111	042105	
	010146	052040	020117	047111	
	010154	042524	051122	050125	
	010162	000124			
710	010164	050050	024503	020040	DH2: .ASCIZ "(PC) (PS) (SP) TEST# (LKS) "
	010172	020040	050050	024523	
	010200	020040	020040	051450	
	010206	024520	020040	020040	
	010214	042524	052123	020043	
	010222	020040	046050	051513	
	010230	020051	020040	000	
711		010236			.EVEN
712	010236	001116	001172	001170	DT2: SERRPC, SREG7, SREG6, SREG5, LKS
	010244	001166	177546		
713	010250	000000			0
714	010252	046103	041517	020113	EM3: .ASCIZ "CLOCK INTERRUPTED WHEN THE PROCESSOR PRIORITY WAS TOO HIGH"
	010260	047111	042524	051122	
	010266	050125	042524	020104	
	010274	044127	047105	052040	
	010302	042510	050040	047522	
	010310	042503	051523	051117	
	010316	050040	044522	051117	
	010324	052111	020131	040527	
	010332	020123	047524	020117	
	010340	044510	044107	000	
715	010345	050	041520	020051	DH3: .ASCIZ "(PC) (PS) (SP) TEST# (LKS) "
	010352	020040	024040	051520	
	010360	020051	020040	024040	
	010366	050123	020051	020040	
	010374	052040	051505	021524	
	010402	020040	024040	045514	
	010410	024523	020040	000040	
716					.EVEN
717	010416	001116	001172	001170	DT3: SERRPC, SREG7, SREG6, SREG5, LKS
	010424	001166	177546		
718	010430	000000			0
719	010432	046103	041517	020113	EM4: .ASCIZ "CLOCK GIVES UNEQUAL # OF PULSES OVER TWO EQUAL PERIODS OF TIME"
	010440	044507	042526	020123	
	010446	047125	050505	040525	
	010454	020114	020043	043117	
	010462	050040	046125	042523	
	010470	020123	053117	051105	
	010476	052040	047527	042440	
	010504	052521	046101	050040	
	010512	051105	047511	051504	
	010520	047440	020106	044524	

```
720 010526 042515 000
    010531 050 041520 020051 DM4: .ASCIZ "(PC) (PS) (SP) TEST# 1ST 2ND"<15><12>"
    010536 020040 024040 051520
    010544 020051 020040 024040
    010552 050123 020051 020040
    010560 052040 051505 021524
    010566 020040 030440 052123
    010574 020040 020040 031040
    010602 042116 005015 020040
    010610 020040 020040 020040
    010616 020040 020040 020040
    010624 020040 020040 020040
    010632 020040 020040 020040
    010640 020040 020040 020040
    010646 042520 044522 042117
    010654 020040 042520 044522
    010662 042117 000
721 010666 010666 .EVEN
722 010666 001116 001172 001170 DT4: SERRPC, SREG7, SREG6, SREG5, SREG1, SREG2
    010674 001166 001156 001160
723 010702 000000 0
724 010704 045514 020123 042522 EMS: .ASCIZ "LKS REGISTER RESPONDS TO ANOTHER ADDRESS"
    010712 044507 052123 051105
    010720 051040 051505 047520
    010726 042116 020123 047524
    010734 040440 047516 044124
    010742 051105 040440 042104
    010750 042522 051523 000
725 010755 050 041520 020051 DM5: .ASCIZ "(PC) (PS) (SP) TEST# ADDRESS"
    010762 020040 024040 051520
    010770 020051 020040 024040
    010776 050123 020051 020040
    011004 052040 051505 021524
    011012 020040 040440 042104
    011020 042522 051523 000
726 011026 011026 .EVEN
727 011034 001116 001172 001170 DT5: SERRPC, SREG7, SREG6, SREG5, SGDADR
    011034 001166 001120
728 011040 000000 0
729 011042 020101 047516 051440 EM6: .ASCIZ "A NO SACK TIMEOUT HAS OCCURED"
    011050 041501 020113 044524
    011056 042515 052517 020124
    011064 040510 020123 041517
    011072 052503 042522 000104
730 011100 050050 024503 020040 DM6: .ASCIZ "(PC) (PS) (SP) TEST# (LKS) "
    011106 020040 050050 024523
    011114 020040 020040 051450
    011122 024520 020040 020040
    011130 042524 052123 020043
    011136 020040 046050 051513
    011144 020051 020040 000
731 011152 011152 .EVEN
732 011152 001116 001172 001170 DT6: SERRPC, SREG7, SREG6, SREG5, LKS
    011160 001166 177546
```



```

733 011164 000000 0
734 011166 051127 047117 020107 EM7: .ASCIZ "WRONG CONDITION CODES WERE PUT ONTO STACK BY INTERRUPT"
    011174 047503 042116 052111
    011202 047511 020116 047503
    011210 042504 020123 042527
    011216 042522 050040 052125
    011224 047440 052116 020117
    011232 052123 041501 020113
    011240 054502 044440 052116
    011246 051105 052522 052120
    011254 000
735 011255 050 041520 020051 DH7: .ASCIZ "(PC) (PS) (SP) TEST# CC WAS CC S/B"
    011262 020040 024040 051520
    011270 020051 020040 024040
    011276 050123 020051 020040
    011304 052040 051505 021524
    011312 020040 041440 020103
    011320 040527 020123 041440
    011326 020103 027523 000102
736 011334 001116 001172 001170 .EVEN
737 011342 001166 000776 001124 DT7: SERRPC, SREG7, SREG6, SREG5, BUF1, SGDDAT
738 011350 000000 0
739 011352 051127 047117 020107 EM10: .ASCIZ "WRONG PC PUT ONTO THE STACK BY AN INTERRUPT"
    011360 041520 050040 052125
    011366 047440 052116 020117
    011374 044124 020105 052123
    011402 041501 020113 054502
    011410 040440 020116 047111
    011416 042524 051122 050125
    011424 000124
740 011426 050050 024503 020040 DH10: .ASCIZ "(PC) (PS) (SP) TEST# 2(SP)WAS 2(SP)S/B "
    011434 020040 050050 024523
    011442 020040 020040 051450
    011450 024520 020040 020040
    011456 042524 052123 020043
    011464 040040 051450 024520
    011472 040527 020123 024100
    011500 050123 051451 041057
    011506 020040 000
741 011512 001116 001172 001170 .EVEN
742 011520 001166 000774 001124 DT10: SERRPC, SREG7, SREG6, SREG5, BUF2, SGDDAT
743 011526 000000 0
744 011530 051124 042531 020104 EM11: .ASCIZ "TRYED TO ACCESS THE LKS REGISTER, AND TRAPPED"
    011536 047524 040440 041503
    011544 051505 020123 044124
    011552 020105 045514 020123
    011560 042522 044507 052123
    011566 051105 020054 047101
    011574 020104 051124 050101
    011602 042520 000104
745 011606 050050 024503 020040 DH11: .ASCIZ "(PC) (PS) (SP) TEST#"
    011614 020040 050050 024523

```

	011622	020040	020040	051450	
	011630	024520	020040	020040	
	011636	042524	052123	000043	
746					.EVEN
747	011644	001116	001172	001170	DT11: SERRPC, SREG7, SREG6, SREG5
	011652	001166			
748	011654	000000			0
749		000001			.END

R0027	004722	R0030	005036	R0031	005154	R0032	005302
R0033	005436	R1	=X000001	R1013	002736	R2	=X000002
R2013	002770	R3	=X000003	R3013	003052	R4	=X000004
R4013	003104	R5	=X000005	R6	=X000006	R7	=X000007
SP	=X000006	STACK	= 001100	START	001416	STKLMT	= 177774
STMES	007604	SMR	= 177570	SMD	= 000001	SM00	= 000001
SM01	= 000002	SM02	= 000004	SM03	= 000010	SM04	= 000020
SM05	= 000040	SM06	= 000100	SM07	= 000200	SM08	= 000400
SM09	= 001000	SM1	= 000002	SM10	= 002000	SM11	= 004000
SM12	= 010000	SM13	= 020000	SM14	= 040000	SM15	= 100000
SM2	= 000004	SM3	= 000010	SM4	= 000020	SM5	= 000040
SM6	= 000100	SM7	= 000200	SM8	= 000400	SM9	= 001000
TBITVE	= 000014	TKVEC	= 000060	TPVEC	= 000064	TRAPVE	= 000034
TRAP0	005674	TRPHES	007661	TRPH25	007722	TRTVEC	= 000014
TYPOS	= 104410	TYPE	= 104400	TYPOC	= 104402	TYPON	= 104406
TYP05	= 104404	T0001	001550	T0002	001610	T0003	001656
T0004	001722	T0005	001776	T0006	002102	T0007	002212
T0010	002350	T0011	002500	T0012	002626	T0013	002722
T0014	003210	T0015	003322	T0016	003434	T0017	003546
T0020	003660	T0021	003772	T0022	004104	T0023	004210
T0024	004306	T0025	004410	T0026	004542	T0027	004656
T0030	005000	T0031	005130	T0032	005256	T0033	005412
T0034	005544	WORD	001230	SBDADR	001122	SBDAT	001126
SBELL	001220	SCHTAG	001100	SCH1	= 000010	SCH2	= 000020
SCM3	= 000010	SCH4	= 000010	SCRLF	001225	SDBLK	006540
SDOAGN	005650	SOTBL	006530	SENDAD	005640	SENDCT	005610
SENDNG	005654	SENULL	005671	SEOP	005554	SEOPCT	005602
SERFLG	001103	SERMAX	001115	SERROR	007132	SERRPC	001116
SERRTB	001232	SERTY	006776	SERTTL	001112	SESCAP	001216
SFILLC	001150	SFILLS	001147	SGDADR	001120	SGDAT	001124
SGET42	005632	SHD	= 000000	SICNT	001104	SILLUP	007520
SITEMB	001114	SLF	001226	SLPADR	001106	SLPERR	001110
SMXCNT	006214	SNULL	001146	SOCNT	006772	SOMODE	006774
SOVER	006200	SPASS	001100	SPOMER	007526	SPMRDN	007356
SPMFLP	007444	SOUES	001224	SREGAD	001152	SREG0	001154
SREG1	001156	SREG2	001160	SREG3	001162	SREG4	001164
SREG5	001166	SREG6	001170	SREG7	001172	SSAVR6	007524
SSCOPE	005746	SSETUP	= 000037	SSTUP	= 177777	SSVLAD	006152
SSMR	= 167400	STIMES	001214	STKB	001140	STKS	001136
STMP0	001174	STMP1	001176	STMP2	001200	STMP3	001202
STMP4	001204	STMP5	001206	STMP6	001210	STMP7	001212
STN	= 000000	STP8	001144	STPFLG	001151	STPS	001142
STRAP	007324	STRP	= 000012	STRPAD	007344	STSTNM	001102
STYPOS	006324	STYPE	006216	STYPOC	006574	STYPON	006610
STYPOS	006550	SXTSTR	005756	SOFILL	006773	.	= 011656

ERRORS DETECTED: 0

#DZKMA.D,DZKMA.D+DZKMA.D.P11
RUN-TIME: 40 21 0 SECONDS
CORE USED: 14K

