DATA GENERAL
CORPORATION
Southboro,
Massachusetts 01772
(617) 485-9100

PROGRAM

Nova 800 Teletype Test

TAPES

Binary 095-000048-01

ABSTRACT

The Nova 800 Teletype Test is a maintenance program designed to detect malfunctions in the teletype logic, interrupt system, and the I/O bus logic. The program may be used to test teletype models ASR33, KSR33, and KSR35. The program contains routines to punch and read random data, echo typed keys, punch from the switch register, etc.

097-000025-01

```
!                       NOVA-800 TELETYPE TEST

!1.     ABSTRACT
!           THIS IS A MAINTENANCE PROGRAM DESIGNED
!           TO DETECT MALFUNCTIONS IN THE TELETYPE LOGIC,
!           INTERRUPT SYSTEM, AND THE I/O BUS LOGIC. THE
!           PROGRAM MAY BE USED TO TEST TELETYPE MODELS
!           ASR33, KSR33, AND KSR35. THE PROGRAM CONTAINS
!           ROUTINES TO PUNCH AND READ RANDOM DATA, ECHO
!           TYPED KEYS, PUNCH FROM THE SWITCH REGISTER, ETC.

!2.     MACHINE REQUIREMENTS
!2.1        NOVA-800 PROCESSOR
!2.2        2K READ/WRITE MEMORY
!2.3        ASR33, KSR33, OR KSR35 TELETYPE

!3.     SWITCH SETTINGS
!3.0            000006      =NEW DEVICE CODE STARTING ADDRESS
!3.1            000052      =DIAGNOSTIC STARTING ADDRESS
!3.2            000051      =PRINT C(SWITCHS)R ON TELETYPE
!3.3            000052      =ECHO TYPED INPUT
!3.4            000053      =PRINT CHARACTOR SET
!3.5            000054      =PUNCH AND READ RANDOM DATA
!3.6            000055      =PUNCH AND READ COUNTER
!3.7            SWITCH 0(1) =PROCEED FROM A ERROR
!3.8            SWITCH 1(1) =SET FOR KSR TELETYPES
!3.9            SWITCH 2(1) =DON'T CHECK KEYBOARD PARITY

!4.     OPERATING PROCEEDURE
!4.0        LOAD THE PROGRAM VIA THE BINARY LOADER, USE
!           THE HIGH SPEED READER IF AVAILABLE.

!4.1        IF THE DEVICE CODES OF THE UNIT UNDER TEST
!              ARE NOT 10 AND 11, ENTER THE EVEN NUMBER
!              IN REGISTER 000005, AND START AT 000006.
!              THE PROGRAM WILL MODIFY ITSELF AND HALT
!              READY FOR THE NEXT STEP.

!4.2        TESTING A ASR33 TELETYPE
!4.2.1          SET THE TELETYPE TO THE LOCAL POSITION
!4.2.2          PRESS THE TAPE PUNCH ON BUTTON
!4.2.3          PRESS THE "HERE IS" KEY. THE PUNCH SHOULD
!               FEED BLANK TAPE.
!4.2.4          PRESS THE "REPT" KEY, THEN THE "RUB OUT"
!               KEY. EXAMINE THE TAPE FOR ALL HOLES .
!4.2.5          PERFORM THE ABOVE TWO STEPS UNTILL ABOUT
!               10 INCHES OF TAPE HAVE BEEN PUNCHED.
!4.2.6          SET THE TELETYPE SWITCH TO LINE.
!4.2.7          PLACE THE END OF THE PUNCHED TAPE IN THE
!               READER. A TAPE LOOP ABOUT ONE FOOT LONG
!               IS THUS FORMED BETWEEN THE READER AND
!               THE PUNCH.
!4.2.8          SET THE READER SWITCH TO THE START POSITION.
```

```
                    14.2.9          SET THE SWITCH REGISTER TO 000050
                    14.2.10         PRESS START
                    14.2.11         THE PROGRAM WILL RUN UNTILL MANUALLY
                    )               STOPPED OR A ERROR IS DETECTED. THE WORD
                    )               "PASS" WILL BE PRINTED AT THE END OF EACH
                    )               PASS. OTHER INDETERMINATE CODES WILL ALSO
                    )               BE PRINTED.
                    14.2.12         WHEN THE PROGRAM HAS COMPLETED SEVERAL
                    )               PASSES SET THE DATA SWITCHES TO 000053.
                    14.2.13         PRESS RESET
                    14.2.14         SET THE READER SWITCH TO FREE.
                    14.2.15         PRESS THE PUNCH OFF BUTTON
                    14.2.16         PRESS START
                    14.2.17         A SUBSET OF THE ASCII CHARACTOR SET WILL
                    )               BE PRINTED. THE SET BEGINS WITH THE SEQ-
                    )               UENCE "MMM" TO PERMIT CARRIAGE RETURN
                    )               ADJUSTMENTS ON A WIDE LETTER.
                    14.2.18         AFTER SEVERAL LINES HAVE BEEN PRINTED
                    )               PRESS RESET AND EXAMINE THE PRINTED DATA.
                    14.2.19         SET THE DATA SWITCHES TO 000052
                    14.2.20         PRESS START
                    14.2.21         STRIKE EACH KEY ON THE KEYBOARD AND CHECK
                    )               TO INSURE THAT IT IS ECHOED PROPERLY.
                    14.2.22         SET THE DATA SWITCHES TO 000054
                    14.2.23         PRESS RESET.
                    14.2.24         REMOVE THE PUNCHED TAPE FROM THE
                    )               READER AND PUNCH.
                    14.2.25         PRESS THE PUNCH ON BUTTON
                    14.2.26         PRESS START
                    14.2.27         INSERT THE LEADER OF ALL ZEROS INTO
                    )               THE READER.
                    14.2.28         SET THE READER SWITCH TO START.
                    14.2.29         THE PROGRAM SHOULD PUNCH AND READ RANDOM
                    )               DATA UNTILL MANUALLY STOPPED OR A ERROR
                    )               IS DETECTED.
                    14.3          TELETYPES WHICH DO NOT CONTAIN THE READER AND
                    )             PUNCH OPTIONS MAY BE TESTED BY SETTING DATA
                    )             SWITCH 1(1) AND OMITTING THOSE STEPS PERTAIN-
                    )             ING TO THE READING AND PUNCHING OF TAPE.

                    15.         PROGRAM OUTPUT/ERROR DISCRIPTION
                    15.1            WHEN A ERROR IS DETECTED BY THE DIAGNOSTIC
                    )               THE PROGRAM WILL HALT AT LOCATION ERR+6.
                    )               EXAMINE AC3 TO OBTAIN THE ADDRESS OF THE
                    )               FAILING ROUTINE.
                    )               CONSULT THE LISTING TO DETERMINE THE CAUSE
                    )               OF THE ERROR.
                    )               PRESSING CONTINUE WILL CAUSE THE PROGRAM
                    )               TO ENTER A FAILING LOOP SUITABLE FOR SCOPING.
                    )               SETTING SWITCH 0(1) WILL CAUSE THE PROGRAM
                    )               TO PROCEED FROM THE ERROR.
                    15.2            WHEN A HALT IS EXECUTED IN ROUTINES OTHER
                    )               THAN THE DIAGNOSTIC CONSULT THE LISTING FOR
                    )               THE CAUSE OF ERROR.
```

```
)5.3        PUNCH SWITCHES ROUTINE (START ADDRESS 51)
)           THIS ROUTINE MAY BE USED FOR SCOPING THE
)           TELETYPE OUTPUT LOGIC. THE ROUTINE TRANS-
)           MITTS A CHARACTOR EVERY 102MS INDEPENDENT
)           OF THE TELETYPE BUSY OR DONE FLAG. THE
)           CHARACTOR TRANSMITTED IS A FUNCTION OF THE
)           RIGHT HALF DATA SWITCHES.
)5.4        TERMINATING A LINE INPUTTED IN THE ECHO
)           TEST WITH "CTRL Z" WILL CAUSE THE LINE TO
)           BE REPEATED UNTILL ANOTHER KEY IS STRUCK.
)           THIS PERMITS SPECIAL CODE COMBINATIONS,
)           FOR MECHANICAL ADJUSTMENTS, TO BE TYPED.

)6.     PROGRAM DISCRIPTION/THEORY OF OPERATION
)6.1        TELETYPE DIAGNOSTIC (START ADDRESS 50)
)           THIS ROUTINE WILL TEST THE TELETYPE INTER-
)           FACE LOGIC. THE ROUTINE WILL ALSO TEST
)           INPUT-OUTPUT BUS AND INTERRUPT LOGIC
)           CONTAINED IN THE CPU. AT THE END OF EACH
)           PROGRAM PASS THE WORD PASS IS PRINTED.
)           IF AFTER A FEW MINUTES THE WORD PASS IS
)           NOT PRINTED THE PROGRAM MAY BE IN A LOOP
)           OR THERE MAY BE A TELETYPE MALFUNCTION.
)6.2        PUNCH SWITCHES (START ADDRESS 51)
)           THE PUNCH SWITCHES ROUTINE PERMITS THE
)           OPERATOR TO SCOPE THE TELETYPE OUTPUT
)           LOGIC. THE ROUTINE ISSUES A I/O RESET
)           PULSE, SENDS THE CONTENTS OF THE SWITCHES
)           RIGHT HALF, WAITS 102MS, AND ITERATES THE
)           SEQUENCE.
)6.3        ECHO TEST (START ADDRESS 52)
)           THE ECHO TEST PROVIDES A MEANS OF DETER-
)           MINING IF THE CODES PRODUCED BY THE KEY-
)           BOARD ARE CORRECT. WHEN THE OPERATOR STRIKES
)           A KEY THE PROGRAM RECEIVES THE CHARACTOR AND
)           CHECKS IT FOR EVEN PARITY.... THE CHARAC-
)           TOR IS STORED IN A BUFFER AND ECHOED BACK
)           TO THE OPERATOR. IF A TAB HAS BEEN TYPED
)           A RUBOUT IS ALSO ECHOED. THE OPERATOR MUST
)           CHECK THE CHARACTOR PRINTED TO BE THE SAME
)           AS THE KEY DEPRESSED. WHEN THE CARRIAGE
)           RETURN KEY IS STRUCK THE PROGRAM CLEARS
)           THE BUFFER AND ECHOES A CARRIAGE RETURN
)           LINE FEED SEQUENCE. IF THE OPERATOR TERMIN-
)           ATES A LINE WITH A "CTRL Z" KEY THE PROGRAM
)           WILL INSERT A CARRIAGE RETURN AND ECHO THE
)           LINE UNTILL ANOTHER KEY IS STRUCK.
```

```
;6.4         PRINT CHARACTOR SET (START ADDRESS 53)
;            A PORTION OF THE ASCII CHARACTOR SET IS
;            PEPEATLY TYPED.
;6.5         PUNCH AND READ RANDOM (START ADDRESS 54)
;            THIS IS THE PRIME TEST OF  THE TELETYPE
;            READER AND PUNCH IN FULL DUPLEX OPERATION.
;            THE MSKO INSTRUCTION AND THE INTERRUPT SYS-
;            TEM ARE USED TO PROVIDE RANDOM START-STOP
;            SEQUENCES TO THE READER AND PUNCH. THE DATA
;            PUNCHED IS EVEN PARITY... RANDOM NUMBERS.
;6.6         PUNCH AND READ COUNT (START ADDRESS 55)
;            THIS TEST IS SIMULAR TO THE PREVIOUS TEST
;            EXCEPT THAT A COUNT PATTERN IS USED.

;7.      RESTRICTIONS/MISC
;7.1         THE DIAGNOSTIC ROUTINE WILL PRINT VARIOUS
;            INDETERMINATE CHARACTORS.
;7.2         THE DIAGNOSTIC ROUTINE WILL NOT OPERATE
;            PROPERLY WITH THE MODEL 37 TELETYPE.


.EOT
```

```
0005  .MAIN

       000001 .LOC 1
00001  000347         INTR              ;INTERRUPT SERVICE ADDRESS
00002  002105         JMP @BEG          ;MOTHER HEN START

       000005 .LOC 5
00005  177777 XSAV:   -1
00006  002007         JMP @.+1
00007  000153         START

       000045         .LOC 45
00045  000310         EGGS

       000050         .LOC 50

00050  002105         JMP @BEG          ;DIAGNOSTIC
00051  000225         JMP SWITCH        ;PRINT FROM SWITCH
00052  000240         JMP ECHO          ;ECHO THE INPUT
00053  002104         JMP @ALPHA        ;PRINT CHARACTOR SET
00054  002102         JMP @MAINR        ;RANDOM READ/PUNCH
00055  002103         JMP @MAINC        ;COUNTER READ/PUNCH

00056  060010 CDVCD:  060010
00057  000010 DVCD:   10
00060  060010 C60K:   060010
00061  000153 FIRST:  START
00062  002036 LAST:   CRLF
00063  000000 DEVRET: 0
00064  160077 CIOT:   160077
00065  177700 M100:   -100
00066  177634 CM100:  -144
00067  100004 IND4:   @4
00070  001762 ER:     ERR
       006070 EHALT=JSR @ER
00071  001746 SETUP:  ENTER
00072  001772 LOOP:   CYCLE
00073  001734 TIME:   TIMER
00074  057664 CTIML:  24500.   ;98 MS
00075  061634 CTIMH:  25500.   ;102MS
00076  060650 CTIM:   25000.   ;100MS
00077  000764 TWO:    500.     ;2 PER CENT OF 25000
00100  000000 TIMEX:  0
00101  000000 PASS:   0
00102  000463 MAINR:  MAIN
00103  000464 MAINC:  MAIN+1
00104  002044 ALPHA:  ALPH
00105  000546 BEG:    BEGIN
00106  000011 T94A:   .TTO
```

```
00107 000000 SAV0:    0
00110 000000 SAV1:    0
00111 000000 SAV2:    0
00112 000000 SAV3:    0
00113 000000 SAVC:    0
00114 000011 TIO:     .TTO
00115 000010 TIOX:    .TTI
00116 000000 TINCH:   0
00117 135525 C13555:  135525
00120 000000 RANI:    0
00121 000000 BROKEN:  0
00122 000000 ICTR:    0
00123 000000 CRA:     0
00124 000523 RAND:    XRAND
00125 000000 OCTR:    0
00126 000000 RAND0:   0
00127 000100 LT:      100
00130 000000 TYO:     0
00131 000000 TIMDEL:  0
00132 000004 C4:      4
00133 000177 C177:    177
00134 000000 MESSR:   0
00135 000377 C377:    377
00136 000015 C215:    15
00137 000012 C12:     12
00140 004000 C4000:   4000
00141 002342 PC5:     1250.    ;5 PERCENT OF 25000
00142 000347 CINTR:   INTR

00143 002220 CTABL:   TABLE+100
00144 002120 CTABF:   TABLE
00145 000000 CNTR:    0
00146 000032 C232:    32
00147 020000 C20K:    20000
00150 002036 ICRLF:   CRLF
00151 002013 IMESS:   MESS
00152 000011 C11:     11
      000011 .TTO=11
      000010 .TTI=10
```

```
A 0007  .MAIN

00153 030005 START:  LDA 2,XSAV
00154 151400         INC 2,2
00155 024060         LDA 1,C60K
00156 125400         INC 1,1
00157 044056         STA 1,CDVCD
00160 050106         STA 2,T04A
00161 050114         STA 2,TTO
00162 050057         STA 2,DVCD
00163 004201         JSR DEVCD        ;CHANGE THE .TTO DEVICE CODES

00164 024060         LDA 1,C60K
00165 044056         STA 1,CDVCD
00166 030005         LDA 2,XSAV
00167 050115         STA 2,TTOX
00170 050057         STA 2,DVCD
00171 004201         JSR DEVCD        ;CHANGE THE .TTT DEVICE CODES

00172 024065         LDA 1,M100       ;CHANGE CONSTANTS FOR THE NEXT RUN.
00173 020060         LDA 0,C60K
00174 123400         AND 1,0
00175 024057         LDA 1,DVCD
00176 123000         ADD 1,0
00177 040060         STA 0,C60K
00200 063077         HALT

00201 054063 DEVCD:  STA 3,DEVRET
00202 030061         LDA 2,FTRST
00203 021000         LDA 0,0,2        ;FIX THE DEVICE CODE
00204 024064         LDA 1,CIOT       ;163077
00205 123400         AND 1,0
00206 024056         LDA 1,CDVCD      ;INITIALLY 60010
00207 106415         SUB# 0,1,SNR
00210 000216         JMP DEVC2
00211 151400 DEVC1:  INC 2,2
00212 020062         LDA 0,LAST
00213 112414         SUB# 0,2,SZR
00214 000203         JMP DEVCD+2
00215 002063         JMP @DEVRET
00216 021000 DEVC2:  LDA 0,0,2        ;CHANGE THE INSTRUCTION
00217 024065         LDA 1,M100
00220 123400         AND 1,0          ;MASK THE DEVICE CODE
00221 024057         LDA 1,DVCD
00222 123000         ADD 1,0          ;NEW DEVICE CODE
00223 041000         STA 0,0,2
00224 000211         JMP DEVC1
```

```
A ????  .MAIN

??225 ?62677 SWITCH: IORST              ;TYPE FROM SWITCHES
??226 ?2?075         LDA ?,CTIMH
??227 1?2?2?         NEGZ 2,2
??230 1?2411         SUB# 2,2,SKP
??231 1?2410         SUB# 2,2
??232 ?63622?        SKPDN 2            ;DELAY ABOUT 100 MS
??233 1?1?24         INC ?,?,SZR
??234 ??2232         JMP .-4
??235 ?74477         READS 3
??236 ?75111         DOAS 3,.TTO        ;SEND C(SWITCH)R TO
??237 ??2226         JMP SWITCH+1       ;THE TELETYPE.

??240 ??6152 ECHO:   JSR @ICRLF         ;ECHO ON OUTPUT THE
??241 ?32143         LDA 2,CTABL        ;CHARACTORS RECEIVED
??242 ?34144         LDA 3,CTABF        ;ON INPUT.
??243 1?2???         SUB ?,?
??244 ?41???         STA ?,?,3
??245 175???         INC 3,3
??246 156414         SUB# 2,3,SZR
??247 ??2244         JMP .-3
??250 ?32144         LDA 2,CTABF
??251 ?2?145         STA ?,CNTR         ;SET COUNTER TO ?
??252 ??4314 ECHO1:  JSR TIN            ;LOOK FOR INPUT
??253 ?24135         LDA 1,C215
??254 1?6?25         SUB  ?,1,SNR
??255 ??2243         JMP ECHO           ;CARRIAGE TYPED
??256 ?24146         LDA 1,C232
??257 1?6414         SUB# ?,1,SZR
??260 ??2267         JMP ECHO3          ;STORE THE CHARACTOR
??261 ??6152 ECHO2:  JSR @ICRLF         ;CNTL Z TYPED
??262 ?63712?        SKPDZ .TTI         ;KEEP SENDING THE
??263 ??2336         JMP ECHO4          ;LINE TYPED UNTIL
??264 ??6151         JSR @TMESS         ;A KEY IS STRUCK.
??265 ??2122         TABLE
??266 ??2261         JMP ECHO2
??267 ?24135 ECHO3:  LDA 1,C377         ;STORE THE CHARACTOR
??270 ?32145         LDA 2,CNTR
??271 ?12145         ISZ CNTR
??272 ?34144         LDA 3,CTABF        ;BEGIN OF TABLE
??273 151223         MOVZR 2,2,SNC
??274 123421         AND 1,?,SKP
??275 123722         ANDS 1,?
??276 157???         ADD 2,3
??277 ?254??         LDA 1,?,3
??300 107???         ADD ?,1
??301 ?45???         STA 1,?,3
??302 ?32143         LDA 2,CTABL
??303 156414         SUB# 2,3,SZR       ;CHECK FOR
??304 ??2252         JMP ECHO1          ;BUFFER FULL
??305 ??2243         JMP ECHO
??306 ?62212 ECHO4:  NIOC .TTI
??307 ??2243         JMP ECHO
??310 ?????? EGGS:   ?          ;HEN FLAG
??311 ??????         ?          ;DEVICE CODE
??312 ??????         ?          ;ASR37 FLAG
??313 ??????         ?          ;PASS COUNTER
```

```
A 0000  .MAIN

00314 063610 TIN:   SKPDN .TTI          ;INPUT A CHARACTOR
00315 000314        JMP .-1             ;FROM THE KEYBOARD.
00316 060610        DIAC 0,.TTI         ;CHAR TO AC0
00317 063711        SKPDZ .TTI
00320 063077        HALT                ;FLAG CHECK!
00321 061111        DOAS 0,.TTO         ;XMITT CHARACTOR
00322 111000        MOVZ 0,2
00323 126000        ADC 1,1
00324 147000        ADD 2,1
00325 133404        AND 1,2,SZR
00326 000323        JMP .-3
00327 024147        LDA 1,C200K         ;LOOK AT SWITCH 2
00330 070477        READS 2
00331 133404        AND 1,2,SZR
00332 000335        JMP .+3             ;DONT CHECK PARITY.
00333 101012        MOV# 0,0,SZC        ;C(0)=CHARACTOR
00334 063077        HALT
00335 024133        LDA 1,C177          ;STRIP THE PARITY BIT.
00336 123400        AND 1,0
00337 024152        LDA 1,C11
00340 106414        SUB# 0,1,SZR        ;CHECK FOR TAB
00341 001400        JMP 0,3             ;EXIT NOT A TAB.
00342 126000        ADC 1,1
00343 063611        SKPDN .TTO          ;INSERT A RUBOUT
00344 000343        JMP .-1
00345 065111        DOAS 1,.TTO
00346 001400        JMP 0,3

00347 044107 INTR:  STA 0,SAV0          ;INTERRUPT SERVICE
00350 044110        STA 1,SAV1
00351 050111        STA 2,SAV2
00352 054112        STA 3,SAV3
00353 175200        MOVR 3,3
00354 054113        STA 3,SAVC          ;SAVE AC+CARRY.
00355 063777        SKPDZ CPU
00356 063077        HALT                ;POWER FAIL FLAG?
00357 024114        LDA 1,TIO
00360 071477        INTA 2
00361 132415        SUB# 1,2,SNR
00362 000447        JMP TYPE            ;OUTPUT
00363 024115        LDA 1,TIOX
00364 132414        SUB# 1,2,SZR
00365 063077        HALT                ;UNKOWN INTERRUPT! SEE C(2)

00366 063610 XTIN:  SKPDN .TTI          ;READER INPUT
00367 063077        HALT                ;RD DONE FLAG FAILED.
00370 064510        DIAS 1,.TTI         ;GET CHARACTOR
00371 063710        SKPDZ .TTI
00372 063077        HALT                ;FLAG CHECK
00373 063410        SKPBN .TTI
00374 063077        HALT                ;FLAG CHECK
00375 044116        STA 1,TTNCH         ;SAVE INPUT CHARACTOR.
```

```
▲ 0010  .MAIN

000376 125004 TIN1:   MOV 1,1,SZR
000377 000410         JMP TIN2
000400 020117         LDA 0,C1355       ;ON ZERO INPUT
000401 040120         STA 0,RANI        ;INIT RANDOM NUMBER
000402 010121         ISZ BROKEN        ;IF TO MUCH LEADER
000403 102401         SUB 0,0,SKP       ;OR ALL CHARACTORS 0
000404 063077         HALT              ;ERROR.
000405 040122         STA 0,ICTR        ;INIT COUNT PATTERN.
000406 000525         JMP DISMIS        ;EXIT INTERRUPT.

000407 030066 TIN2:   LDA 2,CM100
000410 050121         STA 2,BROKEN
000411 034123         LDA 3,CRA
000412 175005         MOV 3,3,SNR
000413 000410         JMP TIN3          ;RANDOM DATA
000414 010122         ISZ ICTR          ;COUNTER
000415 101010         MOV# 0,0
000416 020122         LDA 0,ICTR
000417 106414         SUB# 0,1,SZR      ;C(0)=GOOD,C(1)=BAD
000420 063077         HALT              ;READ OR PUNCH ERROR.
000421 044122         STA 1,ICTR        ;RESET COUNTER
000422 000471         JMP DISMIS        ;EXIT INTERRUPT.

000423 006124 TIN3:   JSR @RAND
000424 000120         RANI
000425 024116         LDA 1,TINCH
000426 106414         SUB# 0,1,SZR      ;C(0)=GOOD,C(1)=BAD
000427 063077         HALT              ;READ OR PUNCH ERROR.
000430 000463         JMP DISMIS

000431 063611 TYPE:   SKPDN .TTO        ;OUTPUT COUNT OR RANDOM
000432 063077         HALT              ;PATTERN, FLAG CHECK.
000433 034123         LDA 3,CRA         ;SWITCH
000434 175005         MOV 3,3,SNR
000435 000405         JMP TYPE1         ;RANDOM
000436 010125         ISZ OCTR          ;INC OUT COUNTER
000437 101010         MOV# 0,0
000440 020125         LDA 0,OCTR
000441 000403         JMP .+3
000442 006124 TYPE1:  JSR @RAND
000443 000126         RAND
000444 024127         LDA 1,LT          ;LEADER CHECK
000445 030130         LDA 2,TYO
000446 132032         ADCZ# 1,2,SZC
000447 000406         JMP TYPE2
000450 020117         LDA 0,C1355       ;RESET RANDOM IN LEADER
000451 040126         STA 0,RAND
000452 010130         ISZ TYO           ;+1 TO LEADER COUNT
000453 102400         SUB 0,0
000454 040125         STA 0,OCTR
000455 061111 TYPE2:  DOAS 0,.TTO       ;SEND THE CHARACTOR
000456 063411         SKPBN .TTO
000457 063077         HALT              ;FLAG CHECK
000460 063711         SKPDZ .TYO
000461 063077         HALT              ;FLAG CHECK
000462 000431         JMP DISMIS        ;EXIT INTERRUPT.
```

```
A 00011  .MAIN

00463 102401 MAIN:   SUB 0,0,SKP      ;READ/PUNCH RANDOM
00464 102000         ADC 0,0          ;READ/PUNCH COUNTER
00465 040123         STA 0,CRA        ;INITIALIZE
00466 062677         IORST
00467 020056         LDA 0,CM100
00470 040121         STA 0,BROKEN
00471 102400         SUB 0,0
00472 040130         STA 0,TYO
00473 024142         LDA 1,CINTR
00474 044001         STA 1,1
00475 060177         INTEN            ;ENABLE INTERRUPTS
00476 061111         DOAS 0,.TTO
00477 060511         DIAS 0,.TTI

00500 006124 MAIN1:  JSR @RAND        ;MAIN LOOP CNTL
00501 000131         TIMDEL           ;OF READER/PUNCH
00502 152400         SUB 2,2          ;INTERRUPT.
00503 103300         COMS 0,1
00504 072077         MSKO 2
00505 125424         INC 1,1,SZR
00506 000777         JMP .-1
00507 151400         INC 2,2
00510 151237         MOVZRS 2,2,SBN
00511 000772         JMP MAIN1+3
00512 000766         JMP MAIN1

00513 020107 DISMIS: LDA 0,SAV0       ;DISMISS A INTERRUPT
00514 024113         LDA 1,SAV1       ;RESTORE MACHINE STATE.
00515 030111         LDA 2,SAV2
00516 034113         LDA 3,SAVC
00517 175100         MOVL 3,3
00520 034112         LDA 3,SAV3
00521 060177         INTEN
00522 000000         JMP @0

00523 027400 XRAND:  LDA 1,@0,3       ;GENERATE A 8 BIT
00524 020133         LDA 0,C177       ;EVEN PARITY RANDOM
00525 131120         MOVZL 1,2        ;NUMBER.
00526 151120         MOVZL 2,2
00527 133000         ADD 1,2
00530 024403         LDA 1,.+3
00531 147000         ADD 2,1
00532 101300         MOVS 0,0
00533 123725         ANDZS 1,0,SNR
00534 000770         JMP XRAND+1
00535 047400         STA 1,@0,3       ;STORE A 16 BIT RANDOM
00536 111300         MOVS 0,0
00537 126000         ADC 1,1          ;PARITY GENERATOR
00540 107000         ADD 0,1
00541 123424         AND 1,0,SZR
00542 000775         JMP .-3
00543 101200         MOVR 0,0
00544 143300         ADDS 2,0         ;C(0)R=EVEN PARITY#
00545 001401         JMP 1,3
```

```
A 0012  .MAIN

000546 102400 BEGIN:  SUB 0,0          ;DIAGNOSTIC STARTS HERE.
000547 040131         STA 0,PASS

000550 102623 T00:    SUBZR 0,3        ;TNTT FOR POSSIABLE INTERRUPT.
000551 040001         STA 0,1
000552 006071         JSR @SETUP       ;THE SELB LINE TS
000553 063520         SKPBZ 0          ;GROUNDED.
000554 006070         EHALT            ;CHECK (AB2)
000555 006072         JSR @LOOP        ;AND INPUT TO SKIP LOGIC.

000556 006071 T02:    JSR @SETUP       ;THE SELD LINE TS
000557 063720         SKPDZ 0          ;GROUNDED.
000560 006070         EHALT            ;CHECK (AB0).
000561 006072         JSR @LOOP        ;CHECK SKIP LOGIC.

000562 006071 T04:    JSR @SETUP       ;TTO SHOULD NOT BE
000563 060211         NIOC .TTO        ;BUSY,ITS BEEN CLEARED.
000564 062677         IORST            ;IN TO DIFFIRENT WAYS.
000565 063511         SKPBZ .TTO       ;CHECK TTO BUSY(1) INPUT
000566 006070         EHALT            ;TO SELB LINE (AB2).
000567 006072         JSR @LOOP

000570 006071 T06:    JSR @SETUP       ;TROUBLE IN CPU.
000571 063411         SKPBN .TTO       ;SELECTION OF SKIP SEY.
000572 101001         MOV 0,0,SKP      ;CHECK SKIP INVERT LOGIC
000573 006070         EHALT
000574 006072         JSR @LOOP

000575 006071 T08:    JSR @SETUP       ;TTT SHOULD NOT BE BUSY
000576 060213         NIOC .TTI        ;TT HAS BEEN CLEARED IN
000577 062677         IORST            ;TWO WAYS. CHECK TTT INPUT
000600 063513         SKPBZ .TTT       ;TO SELB LINE (AB2).
000601 006070         EHALT            ;CHECK THE FLOP.
000602 006072         JSR @LOOP

000603 006071 T10:    JSR @SETUP       ;TTO SHOULD NOT BE DONE
000604 060211         NIOC .TTO        ;TT HAS BEEN CLEARED
000605 062677         IORST            ;IN TWO WAYS. CHECK
000606 063711         SKPDZ .TTO       ;TTO INPUT TO SELD (AB0)
000607 006070         EHALT            ;ALSO THE TTO DONE FLOP.
000610 006072         JSR @LOOP

000611 006071 T12:    JSR @SETUP       ;TTT SHOULD NOT BE DONE
000612 060213         NIOC .TTI        ;TT HAS BEEN CLEARED
000613 062677         IORST            ;IN TWO WAYS. CHECK
000614 063713         SKPDZ .TTT       ;TTT INPUT TO SELD(AB0).
000615 006070         EHALT            ;ALSO THE TTI DONE FLOP.
000616 006072         JSR @LOOP
```

```
A 0013  .MAIN

00617 006071 T26:   JSR @SETUP      ;THE TTI BUSY FLOP
00620 060110        NIOS .TTI       ;WAS NOT CLEARED BY
00621 060210        NIOC .TTI       ;A CLEAR PULSE OR
00622 062677        IORST           ;I/O RESET, CHECK CLEAR
00623 063510        SKPBZ .TTI      ;INPUT TO FLOP, ALSO
00624 006070        EHALT           ;E39 8-9,E41 8-9-10
00625 006072        JSR @LOOP

00626 006071 T28:   JSR @SETUP      ;THE TTI BUSY FLOP
00627 060110        NIOS .TTI       ;WAS NOT CLEARED BY
00630 062677        IORST           ;I/O RESET, CHECK
00631 063510        SKPBZ .TTI      ;E41 8-9, I/O RESET
00632 006070        EHALT           ;INPUT TO CLEAR SIDE
00633 006072        JSR @LOOP       ;OF TTI BUSY.

00634 006071 T30:   JSR @SETUP      ;THE TTI BUSY FLOP
00635 060110        NIOS .TTI       ;WAS NOT CLEARED BY
00636 060210        NIOC .TTI       ;TTI SELECT AND A
00637 063510        SKPBZ .TTI      ;CLEAR PULSE, CHECK
00640 006070        EHALT           ;AND GATE E42 8-9-10
00641 006072        JSR @LOOP       ;AND THE INPUT TO E41.

00642 006071 T32:   JSR @SETUP      ;SELECTING THE TTI
00643 060010        NIO .TTI        ;WITHOUT A START
00644 063510        SKPBZ .TTI      ;PULSE SET TTI
00645 006070        EHALT           ;BDP BUSY, CHECK
00646 006072        JSR @LOOP       ;E42 11-12-13

00647 020426        LDA 0,K40       ;A TEST THE INSURE THAT THE
00650 040426        STA 0,XORDEV    ;.TTO RESPONDS ONLY TO ONE
00651 006071 T36:   JSR @SETUP      ;DEVICE CODE.
00652 060111        NIOS .TTO
00653 006073        JSR @TIME
00654 063511        SKPBZ .TTO
00655 060111        NIOS .TTO
00656 020421        LDA 0,CNIOC     ;NIOC .TTO
00657 024417        LDA 1,XORDEV
00660 131000        MOV 1,2
00661 113520        ANDZL 0,2
00662 107000        ADD 0,1         ;C(1)=A NIOC INSTRUCTION
00663 146400        SUB 2,1         ;TO A DEVICE OTHER THAN
00664 044401        STA 1,.+1       ;THE .TTO, IF THE .TTO
00665 000000        0               ;RESPONDS CHECK THE DEVICE
00666 063411        SKPBN .TTO      ;SELECTION LOGIC AND
00667 006070        EHALT           ;JUMPERS.
00670 006072        JSR @LOOP
00671 020405        LDA 0,XORDEV
00672 101224        MOVZR 0,0,SZR
00673 000755        JMP T36-1
```

```
00674 000404          JMP .+4
00675 000040 K40:     40
00676 000000 XORDEV:  0
00677 060211 CNIOC:   NIOC .TTO

00700 006071 T54:     JSR @SETUP      ;START AND TTO SELECT
00701 060111          NIOS .TTO       ;FAILED TO SET TTO
00702 063411          SKPBN .TTO      ;BUSY. CHECK SELB (AB2)
00703 006070          EHALT           ;BUSY FLOP, ETC,ETC.
00704 006072          JSR @LOOP

00705 006071 T56:     JSR @SETUP      ;THE BUSY FLOP (TTO)
00706 060111          NIOS .TTO       ;WAS GATED ON TO THE
00707 063500          SKPBZ 0         ;SELB LINE (AB2)
00710 006070          EHALT           ;WITHOUT TTO SELECT
00711 006072          JSR @LOOP       ;BEING PRESENT.

00712 006071 T58:     JSR @SETUP      ;THE TTO BUSY FLOP
00713 060111          NTOS .TTO       ;WAS CLEARED VIA A
00714 060200          NTOC 0          ;START PULSE WITH-
00715 063411          SKPBN .TTO      ;OUT TTO SELECT.
00716 006070          EHALT
00717 006072          JSR @LOOP
```

```
A 0015  .MAIN

00720 006071 T60:   JSR @SETUP      ;THE TTO BUSY FLOP
00721 060111        NIOS .TTO       ;WAS CLEARED VIA
00722 060011        NIO .TTO        ;SELECTIOG THE TTO
00723 063411        SKPBN .TTO      ;WITHOUT A CLEAR PULSE.
00724 006070        EHALT
00725 006072        JSR @LOOP

00726 006071 T64:   JSR @SETUP      ;THE TTI BUSY FLOP
00727 060110        NIOS .TTI       ;FAILED TO SET VIA
00730 063410        SKPBN .TTI      ;A START PULSE.
00731 006070        EHALT           ;CHECK THE READER
00732 006073        JSR @TIME       ;START LEVEL,ETC,ETC.
00733 063510        SKPBZ .TTI
00734 006072        JSR @LOOP

00735 006071 T66:   JSR @SETUP      ;SETTING TTI BUSY
00736 060110        NIOS .TTI       ;CAUSED SELR
00737 063500        SKPBZ 0         ;LINE (A82) TO GO LOW
00740 006070        EHALT           ;WITHOUT TTI SELECT.
00741 006073        JSR @TIME       ;CHECK AND GATE OF
00742 063510        SKPBZ .TTI      ;(TT RDR BUSY(1),
00743 006072        JSR @LOOP       ;TTI SELECT). E29 1-2-3

00744 006071 T68:   JSR @SETUP      ;SETTING TTO DONE
00745 060111        NIOS .TTO       ;CAUSED ALL DONE
00746 006073        JSR @TIME       ;FLAGS TESTING TO SKIP.
00747 063511        SKPBZ .TTO      ;DONE IS GATED ONTO
00750 063700        SKPDZ 0         ;SELD (A80) WITHOUT
00751 006070        EHALT           ;DEVICE SELECT.
00752 006072        JSR @LOOP

00753 006071 T70:   JSR @SETUP      ;BOTH A I/O RESET
00754 060111        NIOS .TTO       ;AND A CLEAR PULSE
00755 006073        JSR @TIME       ;FAILED TO CLEAR
00756 063511        SKPBZ .TTO      ;THE DONE FLAG(TTO).
00757 060211        NIOC .TTO       ;CHECK THE START+CLR+RST
00760 062677        IORST           ;PULSE, THE FOLLOWING
00761 063711        SKPDZ .TTO      ;INVERTER, THE
00762 006070        EHALT           ;DONE FLOP ITSELF.
00763 006072        JSR @LOOP

00764 006071 T72:   JSR @SETUP      ;A I/O RESET FAILED
00765 060111        NIOS .TTO       ;TO CLEAR THE TTO
00766 006073        JSR @TIME       ;DONE FLOP, INPUT
00767 063511        SKPBZ .TTO      ;TO OR GATE E40 FAILED
00770 062677        IORST           ;CHECK LEAD 10, THEN
00771 063711        SKPDZ .TTO      ;REPLACE E40.
00772 006070        EHALT
00773 006072        JSR @LOOP
```

```
00774 006071 T74:   JSR @SETUP          ;A TTO CLEAR PULSE
00775 060111        NIOS .TTO           ;FAILED TO RESET
00776 006073        JSR @TIME           ;TTO DONE. CHECK
00777 063511        SKPBZ .TTO          ;F40=9,THE OR GATE
01000 060211        NIOC .TTO           ;THAT PRODUCES THE
01001 063711        SKPDZ .TTO          ;(STRT+CLR+RST) PULSE.
01002 006070        EHALT
01003 006072        JSR @LOOP

01004 006071 T76:   JSR @SETUP          ;A TTO STRT PULSE
01005 060111        NIOS .TTO           ;FAILED TO CLEAR TTO
01006 006073        JSR @TIME           ;DONE. CHECK F40=11
01007 063511        SKPBZ .TTO          ;THE OR GATE THAT
01010 060111        NIOS .TTO           ;PRODUCES THE (STRT
01011 063711        SKPDZ .TTO          ;+CLR+RST) PULSE.
01012 006070        EHALT
01013 006072        JSR @LOOP

01014 006071 T78:   JSR @SETUP          ;THE TTO FINISH PULSE
01015 102000        ADC 0,0             ;FAILED TO CLEAR BUSY.
01016 061111        DOAS 0,.TTO         ;SCOPE TO FIND OUT WHY.
01017 006073        JSR @TIME
01020 063511        SKPBZ .TTO
01021 063511        SKPBZ .TTO          ;A LARGE NUMBER OF THINGS
01022 006070        EHALT               ;COULD HAVE FAILED.
01023 006072        JSR @LOOP

01024 006071 T80:   JSR @SETUP          ;TRY TO SET DONE
01025 102000        ADC 0,0             ;VIA TTO FINISH. CHECK
01026 061111        DOAS 0,.TTO         ;SELD (A80)
01027 006073        JSR @TIME           ;INPUTS AND
01030 063511        SKPBZ .TTO          ;THE TTO DONE FLOP
01031 063611        SKPDN .TTO          ;OUTPUT,RESET,AND
01032 006070        EHALT               ;CLOCK INPUTS.
01033 006072        JSR @LOOP

01034 006071 T82:   JSR @SETUP          ;THE CPU FAILED
01035 102000        ADC 0,0             ;TO READ IN ANYTHING
01036 061477        INTA 0              ;ON INTA. CHECK CPU.
01037 100015        COM# 0,0,SNR
01040 006070        EHALT
01041 006072        JSR @LOOP

01042 006071 T84:   JSR @SETUP
01043 102520        SUBZL 0,0           ;A BIT 15 WAS READ
01044 065477        INTA 1              ;IN ON INTA.
01045 107414        AND# 0,1,SZR
01046 006070        EHALT
01047 006072        JSR @LOOP
```

```
001050 006071 T86:    JSR @SETUP        ;AFTER A I/O RESET
001051 061477         INTA 0            ;(ISSUED BY SETUP) INTA
001052 101004         MOV 0,0,SZR       ;SHOULD READ BACK NO
001053 006070         EHALT             ;BITS, BUT IT DID
001054 006072         JSR @LOOP         ;GET BITS BACK.

001055 006071 T88:    JSR @SETUP        ;SET THE TTO
001056 102000         ADC 0,0           ;DONE FLAG, THEN
001057 062077         MSKO 0            ;PERFORM A INTA
001060 062677         IORST             ;AND CHECK IF
001061 061111         DOAS 0,.TTO       ;ANY BITS WERE
001062 006073         JSR @TIME         ;RECEIVED BY THE
001063 063511         SKPBZ .TTO        ;CPU.
001064 061477         INTA 0            ;SUGGEST TTO INT
001065 101005         MOV 0,0,SNR       ;REQUEST FLOP FAILED.
001066 006070         EHALT
001067 006072         JSR @LOOP

001070 006071 T90:    JSR @SETUP        ;WITH THE DONE
001071 060111         NIOS .TTO         ;FLAG SET INTA
001072 006073         JSR @TIME         ;SHOULD READ BACK
001073 063511         SKPBZ .TTO        ;BIT 15, CHECK
001074 061477         INTA 0            ;INTA INPUT TO BIT 15
001075 126520         SUBZL 1,1         ;F31,F32,E12
001076 107405         AND 0,1,SNR
001077 006070         EHALT
001100 006072         JSR @LOOP

001101 006071 T92:    JSR @SETUP        ;A MSKO WITH BIT 15
001102 060111         NIOS .TTO         ;ZERO SHOULD NOT
001103 006073         JSR @TIME         ;CHANGE THE STATE OF
001104 063511         SKPBZ .TTO        ;TTO INT DISABLT.
001105 102120         ADCZL 0,0                 ;NO BITS WERE READ
001106 062077         MSKO 0            ;BACK ON INTA HOW-
001107 061477         INTA 0            ;EVER, CHECK DATA=15
001110 101005         MOV 0,0,SNR       ;INPUT TO TTO INT
001111 006070         EHALT             ;DISABLE FLOP.
001112 006072         JSR @LOOP

001113 006071 T94:    JSR @SETUP        ;THE DEVICE CODE
001114 060111         NIOS .TTO         ;READ BACK FOR
001115 006073         JSR @TIME         ;THE TTO SHOULD
001116 063511         SKPBZ .TTO        ;BE 11 (FIRST TTO) OR
001117 061477         INTA 0            ;51 (SECOND TTO),
001120 024106         LDA 1,T94A        ;SOME OTHER NUMBER
001121 106414         SUB# 0,1,SZR      ;WAS READ BACK.
001122 006070         EHALT             ;EXAMINE C(0).
001123 006072         JSR @LOOP
```

```
▲ 001B  .MAIN

 001124 006071 T96:  JSR @SETUP      ;THE TTO INT DISABLE
 001125 102520       SUBZL 0,0       ;FLOP IS SET VIA
 001126 062077       MSKO 0          ;MSKO THEN CLEARED
 001127 062677       IORST           ;VIA I/O RESET. NO
 001130 060111       NIOS .TTO       ;BITS WERE READ BACK.
 001131 006073       JSR @TIME       ;SUGGEST I/O RESET
 001132 063511       SKPBZ .TTO      ;FAILED TO CLEAR
 001133 061477       INTA 0          ;TTO INT DISABLE FLOP.
 001134 101005       MOV 0,0,SNR     ;OR ITS INPUT TO TTO
 001135 006070       EHALT           ;INT REQ FAILED.
 001136 006072       JSR @LOOP

 001137 006071 T98:  JSR @SETUP      ;THE TTO INT DISABLE
 001140 102520       SUBZL 0,0       ;FLOP EITHER FAILED TO
 001141 062077       MSKO 0          ;SET VIA MSKO OR
 001142 060111       NIOS .TTO       ;THE AND GATE
 001143 006073       JSR @TIME       ;(TTO DONE(1),TTO INT DIS(0))
 001144 063511       SKPBZ .TTO      ;FAILED, BECAUSE IT
 001145 061477       INTA 0          ;DID NOT PREVENT
 001146 101004       MOV 0,0,SZR     ;TTO INT REQ FROM
 001147 006070       EHALT           ;SETTING.
 001150 006072       JSR @LOOP

 001151 006071 U00:  JSR @SETUP      ;CHECK THE TIME IT TAKES
 001152 102000       ADC 0,0         ;TO PRINT A CHARACTOR. CHECK
 001153 061111       DOAS 0,.TTO     ;AND GATE TO MAKE UP FINISH
 001154 006073       JSR @TIME       ;IF TIME IS  0MS,STOP(1)
 001155 063511       SKPBZ .TTO      ;"   "    " 18MS,TTO0(1)
 001156 102000       ADC 0,0         ;"   "    " 27MS,"  1(1)
 001157 061111       DOAS 0,.TTO     ;"   "    " 36MS,"  2(1)
 001160 006073       JSR @TIME       ;"   "    " 45MS,"  3(1)
 001161 063511       SKPBZ .TTO      ;"   "    " 54MS,"  4(1)
 001162 024074       LDA 1,CTIML     ;"   "    " 63MS,"  5(1)
 001163 106432       SUBZ# 0,1,SZC   ;"   "    " 72MS,"  6(1)
 001164 006070       EHALT           ;"   "    " 81MS,"  7(1)
 001165 006072       JSR @LOOP       ;"   "    " 90MS,"  8(1)

 001166 006071 U02:  JSR @SETUP      ;A CHECK TO INSURE THAT
 001167 102000       ADC 0,0         ;DOA TO THE TTO DOES
 001170 061111       DOAS 0,.TTO     ;NOT SHIFT THE REGISTER
 001171 006073       JSR @TIME       ;IF TTO IS BUSY
 001172 063511       SKPBZ .TTO      ;CHECK AND GATE OF
 001173 102000       ADC 0,0         ;(DATA  OUT=A,TTO SELECT,
 001174 061111       DOAS 0,.TTO     ;TTO BUSY(0)) THE TTO
 001175 061011       DOA 0,.TTO      ;BUSY 0 INPUT.
 001176 061011       DOA 0,.TTO
 001177 061011       DOA 0,.TTO
 001200 006073       JSR @TIME
 001201 063511       SKPBZ .TTO
 001202 024074       LDA 1,CTIML     ;FAILURE IF TIME SHORT.
 001203 106432       SUBZ# 0,1,SZC   ;C(CTIML)=98% OF TOTAL.
 001204 006070       EHALT
 001205 006072       JSR @LOOP
```

```
A 0010  .MAIN

 01206 006071 U04:  JSR @SETUP        ;TEST THE TIME FOR TO
 01207 102000       ADC 0,0           ;LONG. CHECK AND GATE
 01210 061111       DOAS 0,.TTO       ;(DATA OUTA,TTO SELECT,
 01211 006073       JSR @TIME         ;TTO BUSY (0)).
 01212 063511       SKPBZ .TTO
 01213 102000       ADC 0,0
 01214 061111       DOAS 0,.TTO
 01215 006073       JSR @TIME
 01216 063511       SKPBZ .TTO
 01217 024075       LDA 1,CTIMH
 01220 106433       SUBZ# 0,1,SNC
 01221 006070       EHALT
 01222 006072       JSR @LOOP

 01223 006071 U06:  JSR @SETUP        ;TEST FOR SHORT TIME
 01224 061111       DOAS 0,.TTO       ;IF START FAILED TO
 01225 006073       JSR @TIME         ;CLEAR OUTPUT BUFFER
 01226 063511       SKPBZ .TTO        ;IT WILL CONTAIN ALL
 01227 060111       NIOS .TTO         ;ONES. CHARACTOR WILL
 01230 006073       JSR @TIME         ;END QUICKLY.
 01231 063511       SKPBZ .TTO
 01232 024074       LDA 1,CTIML
 01233 106432       SUBZ# 0,1,SZC
 01234 006070       EHALT
 01235 006072       JSR @LOOP

 01236 006071 U10:  JSR @SETUP        ;TEST FOR LONG TIME
 01237 061111       DOAS 0,.TTO       ;IF STOP1 WAS 0 AND
 01240 006073       JSR @TIME         ;START STILL GOT THROUGH
 01241 063511       SKPBZ .TTO        ;THE AND GATE TO
 01242 061011       DOA 0,.TTO        ;RESET THE BUFFER
 01243 060111       NIOS .TTO         ;THE TIME WOULD BE
 01244 006073       JSR @TIME         ;LONG.
 01245 063511       SKPBZ .TTO
 01246 024075       LDA 1,CTIMH
 01247 106433       SUBZ# 0,1,SNC
 01250 006070       EHALT
 01251 006072       JSR @LOOP
```

```
A 0020  .MAIN

 01252 006071 U14:  JSR @SETUP       ;TIME TO PRINT A
 01253 061111       DOAS 0,.TTO      ;CHARACTOR IS LESS
 01254 006073       JSR @TIME        ;THAN 2% OFF FROM 100MS.
 01255 063511       SKPBZ .TTO       ;OR A ERROR IS ASSUMED.
 01256 061111       DOAS 0,.TTO
 01257 006073       JSR @TIME
 01260 063511       SKPBZ .TTO
 01261 024076       LDA 1,CTIM
 01262 106423       SUBZ 0,1,SNC
 01263 124400       NEG 1,1
 01264 030077       LDA 2,TWO
 01265 132433       SUBZ# 1,2,SNC
 01266 006070       EHALT
 01267 006072       JSR @LOOP

 01270 006071 U16:  JSR @SETUP       ;THE MSKO LEVEL
 01271 102520       SUBZL 0,0        ;WAS RAISED AT DTOB
 01272 062000       DOB 0,0          ;TIME EVEN THOUGH
 01273 060111       NIOS .TTO        ;IT WAS NOT A CPU INST.
 01274 006073       JSR @TIME        ;CHECK MSKO (A38).
 01275 063511       SKPBZ .TTO
 01276 061477       INTA 0
 01277 101005       MOV 0,0,SNR
 01300 006070       EHALT
 01301 006072       JSR @LOOP

 01302 006071 U18:  JSR @SETUP       ;THE MSKO LEVEL
 01303 102520       SUBZL 0,0        ;WAS RAISED AT
 01304 060077       NIO CPU          ;CPU INST TIME
 01305 060111       NIOS .TTO        ;EVEN THOUGH NO
 01306 006073       JSR @TIME        ;DTOB WAS GIVEN.
 01307 063511       SKPBZ .TTO
 01310 061477       INTA 0
 01311 101005       MOV 0,0,SNR
 01312 006070       EHALT
 01313 006072       JSR @LOOP

 01314 006071 U20:  JSR @SETUP       ;THE INTA LEVEL (A40)
 01315 060111       NIOS .TTO        ;WAS RAISED AT DTIB
 01316 006073       JSR @TIME        ;TIME EVEN THOUGH NO
 01317 063511       SKPBZ .TTO       ;CPU INSTRUCTION WAS
 01320 061400       DIB 0,0          ;GIVEN.
 01321 024005       LDA 1,U20A
 01322 106415       SUB# 0,1,SNR
 01323 006070       EHALT
 01324 006072       JSR @LOOP
 01325 101011       MOV# 0,0,SKP
 01326 000011 U20A: .TTO

                .EOT
```

```
0021  .MAIN

01327 006071 B10:    JSR @SETUP          ;GROUND THE INTR LINE.
01330 102620         SUBZR 0,0           ;NO INTERRUPT SHOULD
01331 040000         STA 0,0             ;OCCURE BECAUSE ION
01332 040001         STA 0,1             ;IS NOT SET.
01333 061111         DOAS 0,.TTO
01334 063511         SKPBZ .TTO          ;CHECK SET GATE ON
01335 000777         JMP .-1 ;INTRQ.      TON SHOULD BE OFF
01336 024000         LDA 1,0
01337 106414         SUB# 0,1,SZR
01340 006070         EHALT
01341 102000         ADC 0,0
01342 006072         JSR @LOOP

01343 006071 B11:    JSR @SETUP          ;GROUND THE INTR LINE.
01344 102620         SUBZR 0,0           ;TURN THE INTERRUPT
01345 040000         STA 0,0             ;SYSTEM ON AND OFF.
01346 040001         STA 0,1             ;NO INTERRUPT SHOULD
01347 061111         DOAS 0,.TTO         ;OCCUR.
01350 063511         SKPBZ .TTO          ;CHECK INTRQ AND
01351 000777         JMP .-1             ;PTG2 SET GATE
01352 060177         NIOS CPU
01353 060277         NIOC CPU
01354 024000         LDA 1,0
01355 106414         SUB# 0,1,SZR
01356 006070 B11A:   EHALT
01357 006072         JSR @LOOP

01360 006071 B12:    JSR @SETUP          ;ATTEMPT PI-DEFER SEQUENCE
01361 152001         ADC 2,2,SKP
01362 001375         B12A                ;INTERRUPT RETURN ADDR
01363 020777         LDA 0,.-1
01364 040001         STA 0,1
01365 024407         LDA 1,B12A-1
01366 044002         STA 1,2             ;EHALT IN LOC 2
01367 071111         DOAS 2,.TTO
01370 063611         SKPON .TTO          ;GROUND THE INTR LINE
01371 000777         JMP .-1
01372 060177         NIOS CPU            ;TON
01373 176400         SUB 3,3             ;NO INTERRUPT. CHECK
01374 006070         EHALT               ;INTRQ,PI OR INTERRUPT
01375 006072 B12A:   JSR @LOOP           ;DEFERRED THRU 0 NOT 1
                                         ;CHECK "PLUS ONE" FOR MA=1
                                         ;IN DEFER CYCLE
```

```
A 0022  .MAIN

 01376 006071 B13:    JSR @SETUP          ;TEST FOR ION RESET
 01377 102001         ADC 0,0,SKP         ;BY PI CYCLE
 01400 001410         B13A
 01401 024777         LDA 1,.-1
 01402 044001         STA 1,1
 01403 061111         DOAS 0,.TTO
 01404 063511         SKPBZ .TTO
 01405 000777         JMP .-1
 01406 060177         NIOS CPU            ;ION STILL ON FOLLOWING
 01407 101000         MOV 0,0             ;INTERRUPT
 01410 063577 B13A:   SKPBZ CPU           ;CHECK PTG1=PI RESET
 01411 006070         EHALT               ;GATE ON ION
 01412 006072         JSR @LOOP

 01413 006071 B14:    JSR @SETUP          ;TEST FOR ONE INSTRUCTION
 01414 102001         ADC 0,0,SKP         ;TO BE PERFORMED BEFORE
 01415 001426         B14A                ;INTERRUPT OCCURS
 01416 024777         LDA 1,.-1
 01417 044001         STA 1,1
 01420 061111         DOAS 0,.TTO
 01421 063511         SKPBZ .TTO
 01422 000777         JMP .-1
 01423 060177         NIOS CPU
 01424 102620         SUBZR 0,0           ;NO INTERRUPT, CHECK ION,
 01425 006070         EHALT               ;INTRQ, PI
 01426 101125 B14A:   MOVZL 0,0,SNR
 01427 101003         MOV 0,0,SNC         ;AC0 NOT CORRECT FOLLOWING
 01430 006070         EHALT               ;INTERRUPT.  CHECK INTRQ
 01431 006072         JSR @LOOP           ;SYNC LOGIC FOR 1 INSTRUCTION
                                          ;WAIT AFTER INTEN AND
                                          ;BEFORE INTERRUPT

 01432 006071 B15:    JSR @SETUP          ;SEE IF PC STORED IN
 01433 102620         SUBZR 0,0           ;LOC 0 DURING PI CYCLE
 01434 040000         STA 0,0
 01435 040001         STA 0,1
 01436 061111         DOAS 0,.TTO
 01437 063511         SKPBZ .TTO
 01440 000777         JMP .-1
 01441 060177         NIOS CPU
 01442 101001         MOV 0,0,SKP         ;IF ERROR
 01443 001444         .+1                 ;CHECK PC TO MEM
 01444 020777         LDA 0,.-1           ;DURING PI CYCLE
 01445 024000         LDA 1,0
 01446 106414         SUB# 0,1,SZR
 01447 006070         EHALT
 01450 006072         JSR @LOOP
```

```
01451 006071 B16:   JSR @SETUP        ;DEFER FAILED TO SET
01452 020067        LDA 0,IND4
01453 040001        STA 0,1
01454 102001        ADC 0,0,SKP       ;AT THE END OF PI-2
01455 001465        B16A              ;CYCLE, CHECK 3 INPUT
01456 024777        LDA 1,.-1         ;OR GATE TO PRODUCE
01457 044004        STA 1,4           ;DEFER SET.
01460 061111        DOAS 0,.TTO
01461 063511        SKPBZ .TTO
01462 000777        JMP .-1
01463 060177        NIOS CPU
01464 101010        MOV* 0,0
01465 103014 B16A:  COM* 0,0,SZR
01466 006070        EHALT
01467 060277        NIOC CPU
01470 006072        JSR @LOOP

01471 006071 B17:   JSR @SETUP        ;A INTERRUPT DEFERED
01472 102001        ADC 0,0,SKP       ;THROUGH LOCATION 0
01473 001505        B17A              ;NOT LOCATION 1, CHECK
01474 024777        LDA 1,.-1         ;1 TO MA AT PI-2 TIME.
01475 044004        STA 1,4           ;CHECK ADD ONE LOGIC
01476 061111        DOAS 0,.TTO       ;(A91) AND GATE
01477 063511        SKPBZ .TTO        ;(PI2,TS3).
01500 000777        JMP .-1
01501 060177        NIOS CPU
01502 101010        MOV* 0,0
01503 063477        SKPBN CPU
01504 006070        EHALT
01505 060277 B17A:  NIOC CPU
01506 006072        JSR @LOOP

01507 006071 B18:   JSR @SETUP        ;A INTERRUPT OCCURED
01510 102001        ADC 0,0,SKP       ;BUT FAILED TO STORE
01511 001522        B18A              ;ANYTHING IN LOC 0.
01512 024777        LDA 1,.-1         ;CHECK INPUTS TO
01513 044004        STA 1,4           ;MEM MOD LOGIC.
01514 040000        STA 0,0
01515 061111        DOAS 0,.TTO
01516 063511        SKPBZ .TTO
01517 000777        JMP .-1
01520 060177        NIOS CPU
01521 000404        JMP .+4
01522 024000 B18A:  LDA 1,0
01523 106415        SUB* 0,1,SNR
01524 006070        EHALT
01525 006072        JSR @LOOP
```

```
01526 006071 V001:    JSR @SETUP          ;DIA DID NOT
01527 102000          ADC 0,0             ;CHANGE THE CONTENTS
01530 060413          DIA 0,.TTI          ;OF AC0. CHECK READ IO
01531 100015          COM# 0,0,SNR        ;THRU MEM TO AR=ACD
01532 006070          EHALT
01533 006072          JSR @LOOP

01534 074477          READS 3             ;IF KSR TTY DON'T
01535 175100          MOVL 3,3
01536 175102          MOVL 3,3,SZC        ;DO THIS SECTION
01537 000561          JMP LE025


01540 006071 V002:    JSR @SETUP          ;STARTING THE PAPER
01541 060110          NIOS .TTI           ;TAPE READER DID NOT
01542 006073          JSR @TIME           ;CAUSE DONE TO SET.
01543 063610          SKPDN .TTI          ;IS THE READER IN THE
01544 101002          MOV 0,0,SZC         ;START POSITION? TAPE IN
01545 006070          EHALT               ;READER? SWITCH IN THE
01546 006072          JSR @LOOP           ;LINE POSITION?

01547 006071 V004:    JSR @SETUP          ;THE SETTING OF
01550 060110          NIOS .TTI           ;TTI DONE DID NOT
01551 006073          JSR @TIME           ;CLEAR TT RDR BUSY.
01552 063610          SKPDN .TTI          ;CHECK TTI DONE(1)
01553 063510          SKPBZ .TTI          ;INPUT TO CLOCK. ALSO
01554 006070          EHALT               ;DATA INPUT SHOULD BE
01555 006072          JSR @LOOP           ;GROUND.

01556 006071 V006:    JSR @SETUP          ;I/O RESET FAILED
01557 060110          NIOS .TTI           ;TO CLEAR THE TTI
01560 006073          JSR @TIME           ;DONE FLOP. CHECK
01561 063510          SKPBZ .TTI          ;E40 5-6. ALSO INVERTER
01562 062677          IORST               ;E39 10-11. TO THE
01563 063710          SKPDZ .TTI          ;RESET SIDE OF THE
01564 006070          EHALT               ;DONE FLOP.
01565 006072          JSR @LOOP

01566 006071 V008:    JSR @SETUP          ;A CLEAR PULSE TO
01567 060110          NIOS .TTI           ;THE TTI LOGIC FAILED
01570 006073          JSR @TIME           ;TO CLEAR DONE.
01571 063510          SKPBZ .TTI          ;CHECK E40 4-6.
01572 060210          NIOC .TTI           ;THE OF GATE TO
01573 063710          SKPDZ .TTI          ;RESET SIDE OF DONE.
01574 006070          EHALT
01575 006072          JSR @LOOP
```

```
001576 006071 V10:    JSR @SETUP        ;A START PULSE TO
001577 060110         NIOS .TTI         ;THE TTI LOGIC FAILED
001600 006073         JSR @TIME         ;TO CLEAR DONE.
001601 063510         SKPBZ .TTI        ;CHECK E40 3-6, THE
001602 060110         NIOS .TTI         ;OR GATE TO RESET SIDE
001603 063710         SKPDZ .TTI        ;OF DONE.
001604 006070         EHALT
001605 006073         JSR @TIME
001606 063510         SKPBZ .TTI
001607 006072         JSR @LOOP

001610 006071 V12:    JSR @SETUP        ;AFTER A I/O
001611 061477         INTA 0            ;RESET INTA SHOULD
001612 101004         MOV 0,0,SZR       ;READ NO BITS.
001613 006070         EHALT
001614 006072         JSR @LOOP

001615 006071 V14:    JSR @SETUP        ;WITH THE DONE FLAG
001616 102400         SUB 0,0           ;SET INTA SHOULD
001617 062077         MSKO 0            ;READ BACK A DEVICE
001620 062677         IORST             ;CODE, CHECK TTI
001621 060110         NIOS .TTI         ;INT REQ FLOP.
001622 006073         JSR @TIME
001623 063510         SKPBZ .TTI
001624 061477         INTA 0
001625 101005         MOV 0,0,SNR
001626 006070         EHALT
001627 006072         JSR @LOOP


001630 006071 V16:    JSR @SETUP        ;CHECK THAT I/O RESET
001631 102520         SUBZL 0,0         ;WILL RESET TTI INT
001632 101100         MOVL 0,0          ;DISABLE.
001633 062077         MSKO 0            ;SET THE FLOP
001634 062677         IORST             ;TRY TO CLEAR
001635 060110         NIOS .TTI         ;SET DONE.
001636 006073         JSR @TIME
001637 063510         SKPBZ .TTI
001640 061477         INTA 0            ;EXPECT DEVICE CODE.
001641 101005         MOV 0,0,SNR
001642 006070         EHALT
001643 006072         JSR @LOOP

001644 006071 V18:    JSR @SETUP        ;A MSKO INSTRUCTION
001645 102400         SUB 0,0           ;WITHOUT A BIT 14
001646 062077         MSKO 0            ;SHOULD NOT SET
001647 060110         NIOS .TTI         ;TTI INT DISABLE, NO
001650 006073         JSR @TIME         ;DEVICE CODE WAS READ
001651 063510         SKPBZ .TTI        ;BACK ON INTA HOWEVER.
001652 061477         INTA 0            ;SUGGEST THE DATA INPUT
001653 101005         MOV 0,0,SNR       ;TO TTI INT DISABLE
001654 006070         EHALT             ;FAILED.
001655 006072         JSR @LOOP
```

```
A 0026  .MAIN

 01656 006071 V20:    JSR @SETUP        ;TTI INT DISABLE FAILED
 01657 102520         SUBZL 0,0         ;TO SET ON A MSKO
 01660 101120         MOVZL 0,0         ;INSTRUCTION OR , THE
 01661 062077         MSKO 0            ;AND GATE (TTI INT DISABLE
 01662 060110         NIOS .TTI         ;(0),TTI DONE(1)) FAILED.
 01663 006073         JSR @TIME         ;MSKO SHOULD PREVENT
 01664 063510         SKPBZ .TTI        ;INTA FROM READING
 01665 061477         INTA 0            ;DEVICE CODE.
 01666 101004         MOV 0,0,SZR
 01667 006070         EHALT
 01670 006072         JSR @LOOP

 01671 006071 V22:    JSR @SETUP        ;CHECK THAT I/O RESET
 01672 102620         SUBZR 0,0         ;WILL CLEAR TTI
 01673 040001         STA 0,1           ;INT REQ.
 01674 060110         NIOS .TTI
 01675 006073         JSR @TIME         ;WAIT FOR DONE
 01676 063510         SKPBZ .TTI
 01677 060177         NIOS CPU          ;ENABLE INTERRUPT
 01700 062477         DIC 0,CPU         ;I/O RESET
 01701 063477         SKPBN CPU         ;CHECK THAT NO
 01702 006070         EHALT             ;INTERRUPT OCCURED.
 01703 060277         NIOC CPU
 01704 006072         JSR @LOOP

 01705 006071 V24:    JSR @SETUP        ;CHECK THE READ A
 01706 060110         NIOS .TTI         ;CHARACTOR TIME TO
 01707 006073         JSR @TIME         ;BE WITHIN 5% OF
 01710 063510         SKPBZ .TTI        ;100MS.
 01711 024076         LDA 1,CTIM
 01712 106423         SUBZ 0,1,SNC
 01713 124400         NEG 1,1
 01714 030141         LDA 2,PC5
 01715 132433         SUBZ# 1,2,SNC
 01716 006070         EHALT
 01717 006072         JSR @LOOP

       001720         LE025=.
 01720 006150 V26:    JSR @ICRLF        ;END OF PASS MESSAGE
 01721 006150         JSR @ICRLF
 01722 006151         JSR @IMESS        ;MESSAGE PASS
 01723 002115         PMESS
 01724 006150         JSR @ICRLF
 01725 063611         SKPDN .TTO
 01726 000777         JMP .-1
 01727 034045         LDA 3,45          ;MOTHER HEN STUFF
 01730 011403         ISZ 3,3
 01731 000401         JMP .+1
 01732 002401         JMP @.+1
 01733 000550         T20
```

```
A 0027  .MAIN

01734 021400 TIMER: LDA 0,0,3        ;TIME THE INST
01735 040405        STA 0,.+5        ;FOLLOWING THE CALL.
01736 102040        ADCO 0,0
01737 102410        SUB# 0,0
01740 101402        INC 0,0,SZC
01741 001401        JMP 1,3
01742 000000        0
01743 000774        JMP .-4
01744 040100        STA 0,TIMEX
01745 001401        JMP 1,3

01746 054413 ENTER: STA 3,LOOPR      ;ITERATION RETURN
01747 034406        LDA 3,ITR        ;THIS TEST INITTALIZES
01750 054406        STA 3,ITRCT      ;EACH ROUTINE.
01751 176400        SUB 3,3
01752 054405        STA 3,ESWIT
01753 062677        IORST            ;I/O RESET
01754 002405        JMP @LOOPR

01755 000003 ITR:   3
01756 000000 ITRCT: 0
01757 000000 ESWIT: 0
01760 000000 RETURN: 0
01761 000000 LOOPR: 0

01762 054776 ERR:   STA 3,RETURN     ;ERROR
01763 034774        LDA 3,ESWIT
01764 175004        MOV 3,3,SZR
01765 002773        JMP @RETURN      ;NOT FIRST ERROR
01766 034772        LDA 3,RETURN     ;SET ERROR SWITCH
01767 054770        STA 3,ESWIT      ;3(3)=PC+1 OF THE
01770 063077        HALT             ;ERROR
01771 002767        JMP @RETURN
```

```
A 0028  .MAIN

01772 054766 CYCLE: STA 3,RETURN     ;ITERATION ROUTINE
01773 014763        DSZ ITRCT        ;END OF EACH TEST.
01774 000410        JMP CYCTS        ;NOT 10 ITERATIONS
01775 034760        LDA 3,ITR        ;RESET ITERATION
01776 054760        STA 3,ITRCT      ;COUNTER. IF ERROR
01777 034760        LDA 3,ESWIT      ;OCCURED STAY IN LOOP.
02000 062677        IORST
02001 175004        MOV 3,3,SZR
02002 002757        JMP @LOOPR       ;ITERATE PROGRAM
02003 002755        JMP @RETURN      ;GO TO NEXT PROG.

02004 034753 CYCTS: LDA 3,ESWIT      ;IF A ERROR
02005 175004        MOV 3,3,SZR      ;LOOK AT SWITCH 0
02006 074477        READS 3
02007 062677        IORST            ;IF SWITCH SET  PROCEED.
02010 175113        MOVL# 3,3,SNC
02011 002750        JMP @LOOPR       ;ITERATE
02012 002746        JMP @RETURN      ;NEXT TEST.

02013 054134 MESS:  STA 3,MESSR      ;PRINT A TEXT MESSAGE
02014 010134        ISZ MESSR
02015 031400        LDA 2,0,3        ;C(2) POINTS TO MESSAGE
02016 024135        LDA 1,C377       ;8 BIT MASK
02017 021000        LDA 0,0,2        ;C(2)=DATA WORD
02020 125112        MOVL# 1,1,SZC
02021 123701        ANDS 1,0,SKP
02022 123401        AND 1,0,SKP      ;C(0)=DATA CHARACTOR
02023 151400        INC 2,2          ;INC TO NEXT WORD
02024 124000        COM 1,1          ;FLIP MASK
02025 004403        JSR CHAR         ;PRINT
02026 000771        JMP MESS+4       ;ANOTHER
02027 002134        JMP @MESSR       ;LAST

02030 101005 CHAR:  MOV 0,0,SNR      ;PRINT A CHARACTOR FROM
02031 001401        JMP 1,3          ;C(0)R. EXIT +2 IF NULL
02032 063511        SKPBZ .TTO
02033 000777        JMP .-1          ;WAIT IF TTO BUSY
02034 061111        DOAS 0,.TTO      ;XMITT CHARACTOR
02035 001400        JMP 0,3 ;EXIT +1

02036 054134 CRLF:  STA 3,MESSR      ;PRINT A CARRIAGE
02037 020136        LDA 0,C215       ;RETURN AND A LINE
02040 004770        JSR CHAR         ;FEED SEQUENCE.
02041 020137        LDA 0,C12
02042 004766        JSR CHAR
02043 002134        JMP @MESSR

02044 006150 ALPH:  JSR @ICRLF       ;ALPHA TEST MESSAGE
02045 006151        JSR @IMESS
02046 002050        ALP              ;MESSAGE STORAGE AREA
02047 000775        JMP ALPH
```

```
▲ 0020  .MAIN

                ALP!    .TXTE |
    02050 046515 MMM."#$%&'()*
    02051 027115
    02052 121442
    02053 122444
    02054 023646
    02055 124450
    02056 025652 +,-./01234
    02057 026654
    02060 127456
    02061 130460
    02062 031662
    02063 032664 56789:;<=
    02064 133466
    02065 034670
    02066 135472
    02067 037675 ?@ABCDEFGHIJ
    02070 040700
    02071 141502
    02072 142504
    02073 043706
    02074 144510
    02075 045712 KLMNOPQRST
    02076 046714
    02077 147516
    02100 150520
    02101 051722
    02102 052724 UVWXYZ[\] ▲
    02103 153526
    02104 054730
    02105 156532
    02106 116333
    02107 057736 0@?@?@?@?@!
    02110 037700
    02111 037700
    02112 037700
    02113 037700
    02114 000300

                PMESS!  .TXTE |PASS|
    02115 040520
    02116 051523
    02117 000000


    02120 000000 TABLE!  0
                .END
```