



DATA GENERAL
CORPORATION

Southboro,
Massachusetts 01772
(617) 485-9100

PROGRAM

Supernova Instruction Timer

TAPES

Binary: 085-000003

ABSTRACT

Supernova Instruction Timer is a maintenance program designed to test the CPU clock circuits by timing the instruction set. The 100 ms teletype clock is used for calibration and is assumed accurate.

111

000002 *LOC 2
00002 002050 JMP 9B1
00003 002051 JMP 9B2

000050 *LOC 50

00050 000400 B1: BEGIN
00051 000401 B2: BEGIN+1
00052 000000 PSWIT: 0
00053 000000 CALIBR: 0
00054 003127 ICRLF: CRLF
00055 003004 IMESS: MESS
00056 000551 CITAB: ITABL-1
00057 001026 IBUFF: BBUF
00060 002777 FBUFF: FBUF
00061 177766 M12: -12
00062 023420 C23420: 23420
00063 000000 TIMEX: 0
00064 000012 C12: 12
00065 003026 IPDEC: PDEC
00066 000000 MSAV: 0
00067 177760 M20: -20
00070 100000 ZPOINT: 00
00071 000060 C60: 60
00072 000024 C24: 24
00073 073101 CDIV: DOCS 2,1
00074 073301 CMUL: DOCP 2,1
00075 000000 MDOPT: 0
00076 000000 TEN0: 0
00077 125252 C5252: 125252

000400 *LOC 400

00400 102401 BEGIN: SUB 0,0,SKP ;PRINT
00401 102000 ADC 0,0 ;DON'T PRINT
00402 040052 STA 0,PSWIT
00403 102400 SUB 0,0
00404 126520 SUBZL 1,1 ;LOOK FOR M/D OPTION
00405 131120 MOVZL 1,2
00406 073301 DOCP 2,1 ;MULTIPLY
00407 146400 SUB 2,1
00410 044075 STA 1,MDOPT ;C(1)=0 IF PRESENT.
00411 004510 JSR TIMER
00412 152000 ADC 2,2
00413 071111 DOAS 2,TTO
00414 004505 JSR TIMER
00415 071111 DOAS 2,TTO
00416 004503 JSR TIMER ;CALIBRATE ON
00417 040053 STA 0,CALIBR ;100 MS CLOCK.

SUPERNOVA INSTRUCTION TIMER

11. ABSTRACT

SUPERNOVA INSTRUCTION TIMER IS A MAINTENANCE PROGRAM DESIGNED TO TEST THE CPU CLOCK CIRCUITS BY TIMING THE INSTRUCTION SET. THE 100 MS TELETYPE CLOCK IS USED FOR CALIBRATION AND IS ASSUMED ACCURATE.

12. MACHINE REQUIREMENTS

SUPERNOVA PROCESSOR
A TYPE 33 OR 35 TELETYPE

13. SWITCH SETTINGS

STARTING ADDRESS=2 PRINT TIMES
STARTING ADDRESS=3 PRINT TIMES IF IN ERROR

14. OPERATING PROCEDURE

14.1 LOAD THE PROGRAM VIA THE BINARY LOADER
14.2 SET SWITCHES TO 3
14.3 PRESS START
14.4 THE PROGRAM WILL PRINT RUB-OUT CHARACTERS AND ANY INSTRUCTION TIME IN ERROR.

15. PROGRAM OUTPUT

15.1 WHEN THE PROGRAM IS STARTED AT LOCATION THREE, ONLY THOSE INSTRUCTIONS WITH TIMING MALFUNCTIONS WILL BE PRINTED. WHEN THE PROGRAM IS STARTED AT LOCATION TWO EACH INSTRUCTION TESTED ALONG WITH ITS EXECUTION TIME WILL BE PRINTED.

15.2 FOR EXAMPLE:

INSTRUCTION EXECUTION TIMES	
MOV 0,0	801
ADD 0,0	801
AND 0,0	800
LDA 0,0	1599
STA 0,0	1599
ISZ 0	1798
DSZ 0	1798
JMP ++1	801
JSR ++1	1399
LDA 0,90	2396
LDA 0,021	2596
LDA 0,031	2596
LDA 0,0(00)	3194
DIA 0,0	2895
DOA 0,0	3293
INTA 0	3294
SKPBN 0	2895
DIVIDE	6982
MULTIPLY	3891

```

36. PROGRAM DESCRIPTION
36.1 THE FOLLOWING PROCEEDURE IS USED TO CALCULATE
; THE INSTRUCTION TIMES. THE TELETYPE IS COM-
; MANED TO PRINT A CHARACTER. A "INC"
; INSTRUCTION THEN RECORDS THE NUMBER OF TIMES
; A SMALL LOOP IS ITERATED BEFORE THE TELETYPE
; BUSY FLAG IS ZERO. THIS COUNT REPRESENTS 100
; MILLISECONDS, AND IS USED FOR CALIBRATION.
; A 1000 WORD BUFFER IS FILLED WITH THE INST-
; RUCTION TO BE TIMED. A CHARACTER IS AGAIN SENT
; TO THE TELETYPE AND PROGRAM CONTROL IS TRANS-
; FERED TO THE BUFFER. THE BUFFER IS EXECUTED 10
; TIMES. WHEN THE INSTRUCTION IN QUESTION HAS
; BEEN EXECUTED 10000 TIMES (1000*10) THE PRO-
; GRAM THEN TIMES THE REMAINDER OF THE TELETYPE
; BUSY FLAG. THE VALUE THUS RECORDED IS SUBTRACTED
; FROM THE 100MS CALIBRATION TIME. THE DIFFERENCE
; REPRESENTS THE TIME FOR 10000 INSTRUCTIONS.
; THIS TIME IS CONVERTED TO NANO SECONDS
; FOR PRINTING.
36.2 THE NUMBER +1 IS MULTIPLIED BY +2. IF THE
; RESULT IS +2 THE MUL/DIV OPTION IS ASSUMED TO
; EXIST. IF THE RESULT IS NOT +2 THE MUL/DIV
; TIMES ARE NOT TESTED. THE TIMES REPRESENT THE
; NUMBER (125252) MULTIPLIED AND DIVIDED BY THE
; NUMBER +1.

37. LIMITATIONS/MISC
; THIS PROGRAM WILL NOT FUNCTION PROPERLY WITH
; A TYPE 37 TELETYPE.

```

```

00420 020052 BEG2: LDA 0,PSWIT
00421 101004      MOV 0,0,SZR
00422 000405      JMP BEG1          ;DON'T PRINT
00423 006054      JSR @ICRLF      ;HEADER
00424 006054      JSR @ICRLF
00425 006055      JSR @IMESS
00426 000644      HEADER

00427 020056 BEG1: LDA 0,CITAB      ;INITIALIZE INST
00430 040020      STA 0,20        ;TO BE TIMED.
00431 034075 BEG:  LDA 3,MDOPT
00432 030074      LDA 2,CMUL
00433 024073      LDA 1,CDIV
00434 022020      LDA 0,@20
00435 106404      SUB 0,1,SZR
00436 112405      SUB 0,2,SNR
00437 175005      MOV 3,3,SNR    ;A M/D INSTRUCTION
00440 101005      MOV 0,0,SNR
00441 000757      JMP BEG2        ;END OF TABLE
00442 030057      LDA 2,IBUFF    ;FILL A 1K BUFFER
00443 034060      LDA 3,FBUFF    ;WITH INSTRUCTIONS.
00444 041000      STA 0,0,2
00445 151400      INC 2,2
00446 156404      SUB 2,3,SZR
00447 000774      JMP *-4

00450 054000 INIT: STA 3,0          ;INIT FOR @ TEST.
00451 054021      STA 3,21
00452 152220      ADCZR 2,2
00453 151220      MOVZR 2,2
00454 050031      STA 2,31
00455 024061      LDA 1,M12
00456 044076      STA 1,TENO
00457 004442      JSR TIMER      ;WAIT AF TTO BUSY.
00460 071111      DOAS 2,TTO
00461 004440      JSR TIMER
00462 102400      SUB 0,0
00463 024077      LDA 1,C5252
00464 034057      LDA 3,IBUFF
00465 152520      SUEZL 2,2
00466 071111      DOAS 2,TTO      ;START THE TTO
00467 001447      JMP 47,3      ;EXIT TO BUFFER

```

111

```
00470 004431 TINSR: JSR TIMER           ;TIME TTD
00471 024053 LDA 1,CALIBR
00472 106400 SUB 0,1           ;C(1)=10K INST TIME
00473 030062 LDA 2,C23020
00474 004432 JSR MULT
00475 030053 LDA 2,CALIBR     ;C(1)=TIME PER INST
00476 004442 JSR DIV          ;IN NANO SECONDS.

00477 044063 TLOOK: STA 1,TIMEX        ;C(1)=ACTUAL TIME
00500 034020 LDA 3,20         ;C(2)=CORRECT TIME
00501 031424 LDA 2,ITABE-ITABL,3
00502 132423 SUBZ 1,2,SNC      ;THEORY-ACTUAL
00503 150400 NEG 2,2         ;C(2)=+DIFFERENCE
00504 020072 LDA 0,C24        ;C(0)=20 NANO SECONDS
00505 112440 SUBQ 0,2
00506 030052 LDA 2,PSWIT
00507 151006 MOV 2,2,SEZ
00510 000721 JMP BEG

00511 035447 LDA 3,IMEST-ITABL,3
00512 054403 STA 3,0+3
00513 006054 JSR @ICRLF
00514 006055 JSR @IMESS      ;MESSAGE ABOUT
00515 000000 C              ;INSTRUCTION TIMED.
00516 024063 LDA 1,TIMEX
00517 006065 JSR @IFDEC      ;PRINT THE TIME.
00520 000711 JMP BEG

00521 102000 TIMER: ADC 0,0           ;RECORD THE TIME
00522 101400 INC 0,0           ;FOR TTD TO FINISH.
00523 063511 SKPBZ TTD
00524 000776 JMP 0-2
00525 001400 JMP 0,3
```

111

```
00526 102460 MULT:   SUBC 0,0           ;C(1)*C(2)
00527 054066         STA 3,MSAV          ;RESULT TO C(0),C(1)
00530 034067         LDA 3,M20
00531 125203 MLOOP:  MOVR 1,1,SNC
00532 101201         MOVR 0,0,SKP
00533 143220         ADDZR 2,0
00534 175404         INC 3,3,SZR
00535 000774         JMP MLOOP
00536 125260         MOVCR 1,1
00537 002066         JMP @MSAV

00540 054066 DIV:   STA 3,MSAV          ;C(0),C(1)/C(2)
00541 034067         LDA 3,M20          ;C(0)=REMAINDER
00542 125120         MOVZL 1,1         ;C(1)=QUOIENT
00543 101100 DLOOP:  MOVL 0,0
00544 142412         SUB# 2,0,SZC
00545 142400         SUB 2,0
00546 125100         MOVL 1,1
00547 175404         INC 3,3,SZR
00550 000773         JMP DLOOP
00551 002066         JMP @MSAV

00552 101000 ITABL:  MOV 0,0           ;INST TO BE TIMED
00553 103000         ADD 0,0
00554 103400         AND 0,0
00555 020000         LDA 0,0
00556 040000         STA 0,0
00557 010000         ISZ 0
00560 014000         DSZ 0
00561 000401         JMP ++1
00562 004401         JSR ++1
00563 022000         LDA 0,@0
00564 022021         LDA 0,@21
00565 022031         LDA 0,@31
00566 022070         LDA 0,@ZPOINT
00567 060400         DIA 0,0
00570 061000         DOA 0,0
00571 061477         INTA 0
00572 063400         SKPBN 0
00573 073101         DOCS 2,1
00574 073301         DOCP 2,1
00575 000000         0
```

000012 *RDX 10

00576	001440	ITABE:	800	
00577	001440		800	
00600	001440		800	
00601	003100		1600	
00602	003100		1600	
00603	003410		1800	
00604	003410		1800	
00605	001440		800	
00606	002570		1400	
00607	004540		2400	
00610	005050		2600	
00611	005050		2600	
00612	006200		3200	
00613	005524		2900	
00614	006344		3300	
00615	006344 007164		3300 3700	
00616	005524		2900	
00617	015504		6980	
00620	007462		3890	

000010 *RDX 8

00621	000662	IMEST:	I1
00622	000667		I2
00623	000674		I3
00624	000701		I4
00625	000706		I5
00626	000713		I6
00627	000717		I7
00630	000723		I8
00631	000730		I9
00632	000735		I10
00633	000743		I11
00634	000751		I12
00635	000757		I13
00636	000766		I14
00637	000773		I15
00640	001000		I16
00641	001005		I17
00642	001012		I18
00643	001017		I19

00644	047311	HEADER:	*TXTE IIN
00645	152123	ST	
00646	052722	RU	
00647	152303	CT	
00650	147711	IO	
00651	120116	N	
00652	154305	EX	
00653	141705	EC	
00654	152125	UT	
00655	147711	IO	
00656	120116	N	
00657	144724	TI	
00660	148515	ME	
00661	000123	SI	

00662	147515	11:	*TXTE IMD
00663	120126	V	
00664	126060	O,	
00665	004460	O	
00666	000011		!
00667	042101	12:	*TXTE IAD
00670	120104	D	
00671	126060	O,	
00672	004460	O	
00673	000011		!
00674	047101	13:	*TXTE IAN
00675	120104	D	
00676	126060	O,	
00677	004460	O	
00700	000011		!
00701	042314	14:	*TXTE ILD
00702	120101	A	
00703	126060	O,	
00704	004460	O	
00705	000011		!
00706	152123	15:	*TXTE IST
00707	120101	A	
00710	126060	O,	
00711	004460	O	
00712	000011		!
00713	051711	16:	*TXTE IIS
00714	120132	Z	
00715	004460	O	
00716	000011		!
00717	051504	17:	*TXTE IDS
00720	120132	Z	
00721	004460	O	
00722	000011		!
00723	046712	18:	*TXTE IJM
00724	120120	P	
00725	025456	*+	
00726	004661	1	
00727	000011		!
00730	051712	19:	*TXTE IJS
00731	120322	R	
00732	025456	*+	
00733	004661	1	
00734	000011		!
00735	042314	110:	*TXTE ILD
00736	120101	A	
00737	126060	O,	
00740	030300	00	
00741	120240		
00742	000011		!

00743 042314 111: *TXTE 1LD
00744 120101 A
00745 126060 O,
00746 131300 @2
00747 004661 1
00750 000000 1

00751 042314 112: *TXTE 1LD
00752 120101 A
00753 126060 O,
00754 031700 @3
00755 004661 1
00756 000000 1

00757 042314 113: *TXTE 1LD
00760 120101 A
00761 126060 O,
00762 024300 @C
00763 030300 @0
00764 004651 >
00765 000000 1

00766 144504 114: *TXTE 1DI
00767 120101 A
00770 126060 O,
00771 004460 O
00772 000011 1

00773 147504 115: *TXTE 1DO
00774 120101 A
00775 126060 O,
00776 004460 O
00777 000011 1

01000 047311 116: *TXTE 1IN
01001 040724 TA
01002 030240 O
01003 004411
01004 000000 1

01005 045323 117: *TXTE 1SK
01006 041120 PB
01007 120116 N
01010 004460 O
01011 000011 1

01012 144504 118: *TXTE 1DI
01013 144526 VI
01014 142504 DE
01015 004411
01016 000000 1

01017 052515 119: *TXTE 1MU
01020 152314 LT
01021 050311 IP
01022 054714 LY
01023 120240

01024 004640
01025 000000 !

01026 000000 BBUF: 0
000012 *RDX 10
002777 *LOC **1000
000010 *RDX 8

02777 010076 PRUF: 1SZ TEN0
03000 002402 JMP @.+2
03001 002402 JMP @.+2
03002 001026 BBUF
03003 000470 TINSR

```

;TTO NON INTERRUPT PACKAGE
;"MESS" PRINTS ASCII MESSAGES AS SPECIFIED BY ASSEMBLR
;"CHAR" PRINTS ASCII CHARACTER, C(0)R,C(0)L MUST BE 0
;WILL RETURN +2 IF C(0)R=0,CORRECTS THE PARITY,33 SIMULR
;"TYPE" PRINTS C(0)R, MUST HAVE PROPER PARITY, RETURN IS
;TO CALL+1,REPLACE THIS ROUTINE WITH INTERRUPT TYPE IF S
;"CRLF" PRINTS A CARRIAGE RETURN
;"POCT" PRINTS C(1) IN OCTAL FOLLOWED BY A TAB
;"PDEC" PRINTS C(1) IN DECIMAL,LEADING ZEROS SUPPRESSED,
;FOLLOWED BY A TAB.

```

```

03004 054545 MESS:   STA 3,MESSR      ;PRINT A TEXT MESSAGE
03005 010544       ISZ MESSR
03006 031400       LDA 2,0,3      ;C(2) POINTS TO MESSAGE
03007 024541       LDA 1,C377    ;A 8 BIT MASK
03010 021000       LDA 0,0,2      ;C(2)=DATA WORD
03011 125112       MOVL# 1,1,SZC
03012 123701       ANDS 1,0,SKP
03013 123401       AND 1,0,SKP    ;C(0)=DATA CHARACTER RIGHT
03014 151400       INC 2,2      ;INC TO NEXT WORD
03015 124000       COM 1,1      ;FLIP MASK
03016 004462       JSR CHAR      ;PRINT
03017 000771       JMP MESS+4    ;ANOTHER
03020 002531       JMP @MESSR    ;LAST

03021 020525 ZOCT:  LDA 0,CH240
03022 101001       MOV 0,0,SKP

03023 020071 FOCT:  LDA 0,C60
03024 030433       LDA 2,OCTAB    ;PRINT C(1) IN OCTAL
03025 000403       JMP +3
03026 030441 PDEC:  LDA 2,DECTB    ;PRINT C(1) IN DECIMAL
03027 020517       LDA 0,CH240    ;SUPPRESS LEADING ZEROS
03030 054447       STA 3,RADRET    ;BOTH ENTRYS PRINT NUMBER
03031 040445       STA 0,ZSUPP    ;THEN TAB TO NEXT POSITION
03032 050401       STA 2,+1
03033 000000 DECOCT: 0      ;A"LDA 2,TABLE" INSTRUCTION
03034 010777       ISZ +,-1
03035 034442       LDA 3,RADRET    ;SETUP "TAB" AT END
03036 020503       LDA 0,CHTAB
03037 151005       MOV 2,2,SNR    ;IF TABLE ENTRY=0
03040 000440       JMP CHAR      ;EXIT WITH TAB
03041 034435       LDA 3,ZSUPP    ;ZEROS SUPPRESS STUF
03042 102400       SUB 0,0
03043 146512 DECOT:  SUBL# 2,1,SZC
03044 000405       JMP DECF
03045 146400       SUB 2,1      ;FORM THE DIGIT
03046 034071       LDA 3,C60
03047 101400       INC 0,0
03050 000773       JMP DECOT
03051 151235 DECF:  MOVZR# 2,2,SNR
03052 034071       LDA 3,C60
03053 054423       STA 3,ZSUPP    ;C(0)=DIGIT
03054 163000       ADD 3,0      ;MAKE ASCII
03055 004423       JSR CHAR      ;PRINT
03056 000755       JMP DECOCT    ;GET NEXT DIGIT

```

```

03057 030425 CCTAB: LDA 2,0+1+0-DECOCT
03060 100000          100000
03061 010000          10000
03062 001000          1000
03063 000100          100
03064 000010          10
03065 000001          1
03066 000000          0

03067 030435 DECTB: LDA 2,0+1+0-DECOCT
          000012 -RDX 10
03070 023420          10000
03071 001750          1000
03072 000144          100
03073 000012          10
03074 000001          1
03075 000000          0
          000010 -RDX 8

03076 000000 ZSUPP: 0
03077 000000 RADRET: 0

03100 054442 CHAR: STA 3,CHRET          ;PRINT C(0) RIGHT
03101 101325          MOVZS 0,0,SNR          ;RETURN +2 IF NULL
03102 001401          JMP 1,3
03103 040440          STA 0,CHSAV
03104 176000          ADC 3,3          ;COMPUTE THE PARITY
03105 117000          ADD 0,3
03106 163404          AND 3,0,5ZR
03107 000775          JMP *-3
03110 176660          SUBCR 3,3          ;COMBIND PARITY WITH CHAR
03111 020432          LDA 0,CHSAV
03112 163300          ADDS 3,0

03113 034426 CHAR1: LDA 3,CHTAB          ;IS THIS A TAB
03114 116405          SUB 0,3,SNR
03115 000407          JMP +7          ;YES
03116 004434          JSR TYPE          ;NO PRINT IT
03117 002423          JMP @CHRET          ;EXIT

03120 020424          LDA 0,CHORZ          ;SIMULATE A TAB
03121 034424          LDA 3,CHAR7          ;VIA 1 TO 8 SPACES
03122 117405          AND 0,3,SNR
03123 002417          JMP @CHRET
03124 020422          LDA 0,CH240
03125 004425          JSR TYPE
03126 000772          JMP *-6

```

```

03127 054420 CRLF:   STA 3,CRLF      ;SAVE RETURN
03130 020410        LDA 0,C215
03131 004747        JSR CHAR      ;PRINT CARRIAGE AND LF
03132 020405        LDA 0,C212
03133 004745        JSR CHAR
03134 102400        SUB 0,0
03135 040407        STA 0,CHORZ   ;CLEAR MORZ POSISTION
03136 002411        JMP 0CRLF     ;EXIT

```

```

03137 000212 C212:   212
03140 000215 C215:   215
03141 000011 CHTAB:  11
03142 000000 CKRET:  0
03143 000000 CHSAV:  0
03144 000000 CHORZ:  0
03145 000007 CHART:  7
03146 000240 CH240: 240
03147 000000 CRLF:   0

```

```

03150 000377 C377:   377
03151 000000 MESSR:  0
03152 054406 TYPE1:  STA 3,TYPRET  ;TYPE THE C(O)R IF
03153 010771        ISZ CHORZ
03154 063511        SKPEZ T10
03155 000777        JMP *-1
03156 081111        DOAS 0,T10
03157 002401        JMP 0TYPRET
03160 000000 TYPRET: 0

```

```

03161 000000 LAST:   0

```

•END