



DATA GENERAL  
CORPORATION

Southboro,  
Massachusetts 01772  
(617) 485-9100

NOVA DEBUGGER I MANUAL

TAPES

ASCII Source: 090-000043

Binary : 091-000069-00

July, 1969

## TABLE OF CONTENTS

1. Introduction
  2. Commands
    - 2.1 Notation
      - 2.2.1 Examine and Modify Commands
      - 2.2.2 Breakpoint Commands
      - 2.2.3 Run Commands
  3. Operating Procedure
- Appendix A: Command Summary

1. Introduction

The NOVA Debugger I is a program used to interface with user routines as an aid in debugging. It provides for one breakpoint within the user's routines. The accumulators, Carry, and memory can be examined and modified from the teletype after a breakpoint has occurred. The machine state can be monitored during execution of a routine using simple commands to Debugger I from the teletype. The Debugger interfaces with any type routine, including those using the NOVA interrupt structure.

## 2. Commands

### 2.1 Format

Commands to the Debugger are of the general form:

(argument)command

Commands are single teletype letters. The argument may be null, a single digit, an address, or an octal number.

A carriage return will be indicated by the symbol " ) ".

A line feed will be indicated by the symbol " / ".

### 2.2 Descriptions

#### 2.2.1 Examine and Modify Commands

Various commands are available which enable the user to examine and change both memory locations and accumulators. If we denote all of these as "registers", a register is said to be "open" after it has been examined. The option is available to simply "close" the register or to type an octal number which is used to replace the current contents of the register before closing it.

The accumulators can be examined by the command,

nA

where "n" specifies the specific accumulator. For example, if "1A" is typed, the Debugger will respond DDDDDD, where the D's represent the octal digits of the contents of AC1. AC1 is now opened and may be modified or simply closed. A ↓ closes the register. The contents may be modified by typing an octal

number followed by a  $\downarrow$  . The status of interrupt enable and Carry can be examined and modified by typing "4A". Bit 15 will be set if interrupts were enabled when the Debugger was entered because of a breakpoint, reset otherwise. Bit  $\emptyset$  will be set if Carry was set when the Debugger was entered.

Any memory location can be opened by typing

adr/

where "adr" is the octal address of the memory location. The contents of the location will be printed. The contents of any address except the breakpoint address can be modified by typing an octal number and closing the register. In addition to using a carriage return to close the register, a line feed will close the register and open the succeeding location.

### 2.2.3 Breakpoint Commands

Breakpoints are the key elements of the Debugger. A breakpoint is an instruction address at which the user would like to stop execution and examine the current state of his routine. Debugger I provides for one breakpoint. To change or activate a given breakpoint, the format is

adrB

where "adr" is the octal address desired for the breakpoint. The breakpoint is deactivated by moving it to an address outside the users program. This address may be in nonexistent memory if desired.

Upon execution, if the instruction at the breakpoint is encountered, execution halts before the breakpoint instruction is executed and control is transferred to the Debugger. The occurrence of a breakpoint causes the breakpoint address to be printed. The user may examine the current machine state by using any of the Debugger commands. If the user should examine the breakpoint address, he will find the proper contents. However, if the user routine executes a HALT or transfers wildly, the breakpoint can be seen by examining the address from the NOVA control console. The breakpoint will appear as the instruction

JMP @1Ø

The Debugger uses location 1Ø of page Ø to store an address entry point to itself.

Some caution should be used in the placement of breakpoints. If an arbitrarily complex subroutine could be transferred to at the breakpoint, the user should have no difficulty (since this is actually what happens). The following restrictions should, however, be noted.

1. The breakpoint should never be placed at a data word (these words are never executed).
2. The breakpoint should never be placed at an instruction which is modified during execution.
3. The breakpoint should not be placed at a point where interrupts cannot be held off for a long period of time, since the Debugger executes an INTDS upon entry.

4. A breakpoint should not be placed at a JSR instruction when the called subroutine requires arguments from the main sequence or when the return is not to call+1 or call+2.

### 2.2.3 Run Commands

The command issued to cause the Debugger to transfer control to the user is

adrR.

If "adr" is specified, the Debugger will transfer to that address. If no "adr" is specified, the Debugger will print a "?" and a line feed.

To return control after a breakpoint, the command

P

is given. The breakpoint instruction is executed at a point in the Debugger and control is returned to the user.

### 3. Operating Procedure

Debugger I requires only 192 (decimal) memory locations. Debugger I, although more limited in scope than Debugger II, is especially suited for users having memory capacity limitations.

An ASCII tape of Debugger I is provided with the standard NOVA software. This tape is originated at 7300 and ends at 7577. If this origin is not convenient for the user, the ASCII tape should be edited to reflect a new origin. In either case, the user can assemble Debugger I to produce a suitable binary tape.

The binary of Debugger I is loaded using the Binary Loader and started at its origin (e.g. 7300). If control is lost between the user routines and Debugger I, the Debugger can be restarted from the control console using its starting address.



APPENDIX A

Command Summary

Command	Effect
nA	Open accumulator n
adrB	Place a breakpoint at adr
P	Proceed from breakpoint
adrR	Begin execution at adr
†	Close current register (if open) and open next register
↓	Close current register
adr/	Open register at adr

APPENDIX B

Listing

007300 .LOC 7300

```
07300 014555 A:      DSZ BPSWT      ;START HERE
07301 040466          STA 0,ACS      ;SAVE ACS
07302 044466          STA 1,ACS+1
07303 050466          STA 2,ACS+2
07304 054466          STA 3,ACS+3
07305 020470          LDA 0,INST      ;SETUP BREAK INSTRUCTION
07306 040470          STA 0,PINST
07307 020465          LDA 0,BADYR      ;SET PROCEED ADDRESS SO
07310 040550          STA 0,PADDR      ;BREAK MAY BE CHANGED.
07311 152660          SUBCR 2,2      ;SAVE CARRY
07312 063577          SKPEZ CPU      ;REMEMBER STATE OF
07313 151400          INC 2,2        ;INTERRUPT FLOP.
07314 060277 C77:    NIOC CPU        ;TURN OF INTERRUPT.
07315 050456          STA 2,ACS+4      ;CARRY SAVE
07316 063511          SKPEZ TTO      ;REMEMBER THE STATE OF
07317 000777          JMP -1         ;TTO DONE FLAG.
07320 063711          SKPDZ TTO
07321 152000          ADC 2,2
07322 050540          STA 2,TFLAG

07323 024451          LDA 1,PADDR      ;IF ENTERED VIA
07324 010531          ISZ BPSWT      ;BREAK POINT TYPE THE
07325 004502 AAA:    JSR FOCT        ;BREAK POINT ADDRESS.
07326 020537          LDA 0,C15
07327 004521          JSR TYPE        ;ECHO CARRIAGE RETURN.

07330 020534 AA:     LDA 0,C12        ;TYPE LINE FEED
07331 004517          JSR TYPE
07332 040531          STA 0,OPEN      ;REGISTER OPEN/CLOSE
07333 152400          SUB 2,2
07334 050521          STA 2,BPSWT      ;BREAK POINT SWITCH.
07335 050524          STA 2,DIN        ;DATA TYPED SWITCH
07336 063610          SKPDN TTI
07337 000777          JMP -1
07340 060610          DIAC 0,TTI      ;INPUT A CHAR
07341 004507 BB:    JSR TYPE        ;ECHO
07342 034575          LDA 3,C177
07343 163400          AND 3,0        ;MASK PARITY

07344 034513          LDA 3,C70
07345 116032          ADCZ# 0,3,SZC
07346 034525          LDA 3,CM60
07347 117046          ADDO 0,3,SEZ
07350 004410          JSR SERH        ;NOT A DIGIT 0-7

07351 010510 CC:    ISZ DIN          ;DIGIT SWITCH
07352 151120          MOVZL 2,2      ;ASSEMBLE DIGITS
07353 153120          ADDZL 2,2
07354 173003          ADD 3,2,SNC      ;SKIP IF EXTRA DIGIT.
07355 000761          JMP BB-3
07356 020736 GOOF:  LDA 0,C77        ;OPERATOR ERROR.
07357 000750          JMP AA-1
```

```

07360 025513 SERH: LDA 1,X,3 ;SPECIAL CHARACTER SEARCH
07361 125112 MOVL# 1,1,SZC
07362 000774 JMP GOOF ;OPERATOR GOOF
07363 175440 C40: INCO 3,3
07364 106424 SUBZ 0,1,SZR ;CHECK CHAR WITH TABLE
07365 000773 JMP SERH ;NOT YET FOUND
07366 007522 JSR #7+X,3 ;A FIND
07367 000000 ACS: 0 ;AC STORAGE AREA
07370 000000 0
07371 000000 0
07372 000000 0
07373 000000 0
07374 007375 BADDR: INST
07375 000761 INST: JMP GOOF
07376 000000 PINST: 0

07377 101020 LINE: MOVZ 0,0 ;LINE FEED TYPED
07400 020465 CARR: LDA 0,C15 ;CARRIAGE TYPED
07401 004447 JSR TYPE ;TIME DELAY FOR ITTY.
07402 004446 JSR TYPE
07403 034535 LDA 3,EXAM
07404 024457 LDA 1,OPEN
07405 020454 LDA 0,DIN
07406 107032 ADDZ# 0,1,SZC ;TEST FOR OPEN REGISTER,DATA
07407 051400 STA 2,0,3
07410 165402 INC 3,1,SZC
07411 000717 JMP AA ;CARRIAGE EXIT
07412 044526 STA 1,EXAM ;LINE FEED CONT
07413 004414 JSR POCT
07414 034524 LDA 3,EXAM
07415 173000 ATYP: ADD 3,2 ;LETTER "A" TYPED
07416 050522 SLASH: STA 2,EXAM ;SLASH TYPED
07417 020755 LDA 0,BADDR ;IF OPENED REGISTER
07420 024755 LDA 1,INST ;IS LAST BREAK POINT
07421 112415 SUB# 0,2,SNR ;DON'T OPEN BUT
07422 000703 JMP AAA ;PRINT CORRECT INST.
07423 025000 LDA 1,0,2
07424 004403 JSR POCT
07425 102000 ADC 0,0 ;OPEN A REGISTER
07426 000704 JMP AA+2

07427 054432 POCT: STA 3,DIN ;OCTAL PRINT C(1)
07430 020733 LDA 0,C40
07431 004417 JSR TYPE ;PRINT SPACE.
07432 152621 SUBZR 2,2,SKP
07433 146401 SUB 2,1,SKP
07434 020435 LDA 0,C57
07435 101400 C1400: INC 0,0
07436 146432 SUBZ# 2,1,SZC
07437 000774 JMP POCT+4
07440 004410 JSR TYPE
07441 151220 MOVZR 2,2
07442 151220 MOVZR 2,2
07443 151224 MOVZR 2,2,SZR
07444 000770 JMP POCT+5 ;ANOTHER DIGIT
07445 020716 LDA 0,C40 ;EXIT WITH SPACE
07446 004402 JSR TYPE
07447 002412 JMP #DIN

```

```

07450 061111 TYPE: DOAS 0,TTO ;SEND CHAR TO TYPE
07451 063511 SKPBZ TTO
07452 000777 C777: JMP --1
07453 001400 JMP 0,3

07454 007301 CA: A+1
07455 000000 BPSWT: 0
07456 002010 CJMP4: JMP #10
07457 000070 C70: 70
07460 000000 PADDR: 0
07461 000000 DIN: 0
07462 000000 TFLAG: 0
07463 000000 OPEN: 0

07464 000012 C12: 12 ;LINE FEED
07465 000015 C15: 15 ;CARRIAGE
07466 000101 101 ;A
07467 000102 102 ;BREAK
07470 000120 120 ;PROCEED
07471 000057 C57: 57 ;SLASH
07472 000122 122 ;RUN
07473 177720 CM60: -60 ;TERMINATOR
07474 007377 LINE
07475 007400 CARR
07476 007415 ATYP
07477 007503 B
07500 007544 PRO
07501 007416 SLASH
07502 007515 R

07503 020672 B: LDA 0,INST ;BREAK LOGIC
07504 042670 STA 0,@BADDR ;RESTORE PREVIOUS BREAK
07505 021000 LDA 0,0,2
07506 040667 STA 0,INST ;INST AT BREAK POINT
07507 050665 STA 2,BADDR
07510 020746 LDA 0,CJMP4
07511 042663 STA 0,@BADDR
07512 000614 JMP AA-2

000113 X=C12-CC

07513 030745 LDA 2,PADDR
07514 125521 INCZL 1,1,SNP ;C(1)=+2
07515 020426 R: LDA 0,CEK ;GO COMMAND
07516 133005 ADD 1,2,SNR
07517 000637 JMP GOOF ;RUN NEEDS A ADDRESS!
07520 050741 STA 2,DIN
07521 040417 STA 0,EXAM

07522 020732 LDA 0,CA
07523 040010 STA 0,10 ;SET BREAK POINT RETURN.

```

```

07524 020643 LDA 0,ACS ;RESTORE AC 0-3
07525 024643 LDA 1,ACS+1
07526 030643 LDA 2,ACS+2
07527 034644 LDA 3,ACS+4
07530 010732 ISZ TFLAG
07531 060211 NIOC TTO ;CLEAR TTO DONE
07532 175141 MOVOL 3,3,SKP
07533 000000 RELADR: 0 ;ADDRESS FOR PC. RELATIVE.
07534 054637 STA 3,ACS+4
07535 034635 LDA 3,ACS+3 ;RESTORE AC-3
07536 014635 DSZ ACS+4 ;PERHAPS TURN ON
07537 060177 C177: NIOS CPU ;INTERRUPT SYSTEM
07540 000000 EXAM: 0 ;JMP IF G,INST IF P.
07541 014720 DSZ DIN ;-1 ADDRESS IF NO SKIP.
07542 002717 EXIT: JMP @DIN
07543 002721 CEX: DIN-EXAM&2777 ;A JMP @DIN INSTRUCTION

07544 020632 PRO: LDA 0,PINST ;PROCEED FROM BREAK
07545 030670 LDA 2,C1400 ;CHECK IF INST IS
07546 034704 LDA 3,C777 ;RELATIVE TO PC.
07547 113523 ANDZL 0,2,SNC
07550 172004 ADC 3,2,SZR
07551 000742 JMP R-2 ;NOT PC RELATIVE.

07552 165220 PRO1: MOVZR 3,1 ;DISPLACEMENT SIGN
07553 123400 AND 1,0 ;EXTENSION.
07554 125620 INCZR 1,1 ;C(1)=200,=BIT 8(1)
07555 107524 ANDZL 0,1,SZR ;C(3)=777,C(1)=400
07556 122400 SUB 1,0 ;C(0)=+,- DISPLACEMENT
07557 030701 LDA 2,PADDR ;PROCEED ADDRESS
07560 113100 ADDL 0,2
07561 020615 PRO2: LDA 0,PINST ;MAKE PROCEED INST
07562 164000 COM 3,1 ;INDIRECT AND RELATIVE
07563 175520 INCZL 3,3 ;TO CALCULATED ADDRESS.
07564 123440 ANDO 1,0
07565 117415 AND# 0,3,SNR
07566 163020 ADDZ 3,0 ;INSERT 0 BIT IN INST.
07567 151200 MOVR 2,2 ;IF 0 INSERT 0 IN
07570 050743 STA 2,RELADR ;THE EFFECTIVE ADDRESS.
07571 030404 LDA 2,CM5
07572 143000 ADD 2,0 ;THE ADDRESS PART TO INST.
07573 126400 SUB 1,1
07574 000717 JMP R-2

07575 000773 CM5: 773
07576 007300 A
07577 002777 JMP @.-1 ;ANOTHER PLACE TO START.

```

007300 .END A

000356 .LOC 356  
006356 .DUSR POS = JSR0 .  
00356 000476 POS1  
006357 .DUSR N = JSR0 .  
00357 000412 N1  
006360 .DUSR NE = JSR0 .  
00360 000421 NE1  
006361 .DUSR E = JSR0 .  
00361 000427 E1  
006362 .DUSR SE = JSR0 .  
00362 000436 SE1  
006363 .DUSR S = JSR0 .  
00363 000444 S1  
006364 .DUSR SW = JSR0 .  
00364 000453 SW1  
006365 .DUSR W = JSR0 .  
00365 000461 W1  
006366 .DUSR NW = JSR0 .  
00366 000470 NW1

```

00400 060323 DISP:  NIOP 23; TRY REMOVING THIS LATER
00401 061323      DOAP 0,23
00402 066123      DOBS 1,23
00403 060023      NIO 23
00404 143000      ADD 2,0
00405 167000      ADD 3,1
00406 014403      DSZ COUNT
00407 000771      JMP DISP
00410 002355      JMP 0 RET
00411 000411 COUNT: .

```

```

000355          RET = 355

```

```

00412 031400 N1:   LDA 2,0,3; GET LENGTH
00413 175400      INC 3,3; BUMP RETURN
00414 054355      STA 3,RET
00415 050774      STA 2, COUNT
00416 152400      SUB 2,2; 0 TO DX
00417 176000      ADC 3,3; -1 TO DY
00420 000760      JMP DISP

```

```

00421 031400 NE1:  LDA 2,0,3
00422 175400      INC 3,3
00423 054355      STA 3,RET
00424 050765      STA 2, COUNT
00425 152000      ADC 2,2; -1 TO DX
00426 000771      JMP NE1-2

```

```

00427 031400 E1:   LDA 2,0,3
00430 175400      INC 3 3
00431 054355      STA 3 RET
00432 050757      STA 2, COUNT.
00433 176400      SUB 3 3; 0 TO DY
00434 152000      ADC 2,2;-1 TO DX
00435 000743      JMP DISP

```

```

00436 031400 SE1:  LDA 2 0 3
00437 175400      INC 3 3
00440 054355      STA 3 RET
00441 050750      STA 2, COUNT
00442 176520      SUBEL 3 3; 1 TO DY
00443 000771      JMP E1+5

```

```

00444 031400 S1:   LDA 2 0 3
00445 175400      INC 3 3
00446 054355      STA 3 RET
00447 050742      STA 2 COUNT
00450 152400      SUB 2,2; 0 TO DX
00451 176520      SUBEL 3,3; 1 TO DY
00452 000726      JMP DISP

```



00453 031400 SW1: LDA 2,0,3  
00454 175400 INC 3,3  
00455 054355 STA 3, RET  
00456 050733 STA 2, COUNT  
00457 152520 SUBZL 2,2; 1 TO DX  
00460 000771 JMP SW1-2 ; 1 TO DY

00461 031400 W1: LDA 2,0,3  
00462 175400 INC 3,3  
00463 054355 STA 3 RET  
00464 050725 STA 2, COUNT  
00465 176400 SUB 3,3; 0 TO DY  
00466 152520 SUBZL 2,2; 1 TO DX  
00467 000711 JMP DISP

00470 031400 NW1: LDA 2,0,3  
00471 175400 INC 3,3  
00472 054355 STA 3,RET  
00473 050716 STA 2, COUNT  
00474 152520 SUBZL 2,2; 1 TO DX  
00475 000722 JMP NE1-2; -1 TO DY

00476 021400 POS1: LDA 0,0,3; GET X  
00477 025401 LDA 1,1,3; GET Y  
00500 100400 NEG 0,0  
00501 124400 NEG 1,1  
00502 001402 JMP 2,3

00600 062677 START: IORST  
00601 006356 R: POS  
00602 177140 -640  
00603 177600 -200  
00604 006357 N  
00605 000400 400  
00606 006361 E  
00607 000140 140  
00610 006362 SE  
00611 000040 40  
00612 006363 S  
00613 000100 100  
00614 006364 SW  
00615 000040 40  
00616 006365 W  
00617 000140 140  
00620 006356 POS  
00621 177160 -620  
00622 000000 0  
00623 006362 SE  
00624 000170 170  
  
00625 006356 I: POS  
00626 177460 -320  
00627 000200 200  
00630 006363 S  
00631 000400 400  
00632 006356 POS  
00633 177440 -340  
00634 000200 200  
00635 006361 E  
00636 000040 40  
00637 006356 POS  
00640 177440 -340  
00641 177600 -200  
00642 006361 E  
  
00643 000040 40

00644	006356	G:	POS
00645	000120		120
00646	000140		140
00647	006366	NW	
00650	000040	40	
00651	006365	W	
00652	000140	140	
00653	006364	SW	
00654	000040	40	
00655	006363	S	
00656	000300	300	
00657	006362	SE	
00660	000040	40	
00661	006361	E	
00662	000140	140	
00663	006360	NE	
00664	000040	40	
00665	006356	POS	
00666	000020	20	
00667	177760	-20	
00670	006361	E	
00671	000100	100	
00672	006363	S	
00673	000120	120	

00674	006356	B:	POS
00675	000240		240
00676	000200		200
00677	006361	E	
00700	000140	140	
00701	006362	SE	
00702	000040	40	
00703	006363	S	
00704	000060	60	
00705	006364	SW	
00706	000040	40	
00707	006362	SE	
00710	000040	40	
00711	006363	S	
00712	000120	120	
00713	006364	SW	
00714	000040	40	
00715	006356	POS	
00716	000240	240	
00717	000200	200	
00720	006363	S	
00721	000400	400	
00722	006356	POS	
00723	000240	240	
00724	000020	20	
00725	006361	E	
00726	000140	140	
00727	006356	POS	
00730	000240	240	
00731	177600	-200	
00732	006361	E	
00733	000140	140	

00734 006356 Y: POS  
00735 000520 520  
00736 000200 200  
00737 006362 SE  
00740 000120 120  
00741 006360 NE  
00742 000120 120  
00743 006356 POS  
00744 000640 640  
00745 000060 60  
00746 006363 S  
00747 000260 300-20

00750 000630 JMP START  
000600 .END START