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

PROGRAM

Core Compare

TAPES

Binary: 091-000007
ASCII Source: 090-000010

ABSTRACT

The Core Compare program is a NOVA utility routine which compares a binary format tape read in on either the paper tape reader or the ASR reader with the contents of core memory. If there are any differences, the address, tape contents and core contents are printed on the Teletype. Core contents are left unchanged.

093-000004-00

1. REQUIREMENTS

1.1 Memory

1K or larger alterable memory.

1.2 Equipment

Teleprinter output and either paper tape reader or Teletype reader.

1.3 External Subroutines

None

1.4 Other

None

2. OPERATING PROCEDURE

2.1 Calling Sequence

The Core Compare program must be loaded in the standard manner (see Binary Loader, 093-000003).

a. Place the binary tape to be compared in the input device (Teletype ASR reader or high-speed reader).

b. Set the data switches to 07400 and press RESET.

c. If using the high-speed reader, set data switch 0 to a 1.

d. Press START. The program will type a carriage return, line feed, and begin reading the tape.

2.2 Input Format

Standard binary format tape.

2.3 Output Format

The output of the Core Compare program is a listing of all differences between the contents of the tape being read and the contents of core memory in the following format:

AAAAA/ TTTTTT ← CCCCCC

where: A's represent the address
T's represent the contents of the tape
C's represent the contents of core

All numbers are printed in octal.

2.4 Error Returns

The checksum information on the binary tape is verified by the Core Compare program. If a checksum error is detected, the program will execute a HALT instruction at location 07474 and 07475 will be displayed in the Address Lights. The block in error may be reread by repositioning the tape in the reader and pressing CONTINUE.

When the tape has been completely read, the program will execute a HALT instruction at location 07433 and 07434 will be displayed in the Address Lights.

2.5 State of Active Registers upon Exit

Not applicable

2.6 Cautions to User

None

3. DISCUSSION

3.1 Algorithms

The Compare program reads the binary input tape in a manner identical to that used by the Binary Loader.

Instead of storing the word read from tape into core, it is compared with the word in core. If the two words are unequal, control passes to a routine which prints the output line on the Teleprinter. After completion, this routine goes to read the next word from tape.

3.2 Limitations and Accuracy

Not applicable

3.3 Size and Timing

The routine requires 173 (octal) words of storage. Its speed is limited by the input and output devices.

3.4 References

See the Binary Loader write-up (093-000003) for a description of the binary tape format.

3.5 Flow Diagrams

Not applicable

4. EXAMPLES AND APPLICATIONS

During program debugging, it is often useful to know if any program locations have been altered. The Core Compare program provides a means for obtaining this information, i.e. for obtaining information for a "post-mortem" analysis of program failures.

The ASCII source tape (090-000010) is provided to enable the user to create versions of the program with different origins.

5. PROGRAM LISTING

A listing of the program with an origin of 07400 follows.

```
                ;CORE COMPARE
                ;COMPARE CORE WITH A BINARY TAPE
                ;SW0(0)=TTI
                ;SW0(1)=PTR

      007400    .LOC 7400
07400 000443 START:  JMP BEGIN          ;START THE PROGRAM
                ;SUBROUTINE TO BUILD A WORD IN AC2
07401 054423 BUILD:  STA 3,TEMP1        ;SAVE RETURN
07402 004406         JSR GTCHR          ;GET FIRST BYTE
07403 171300         MOVS 3,2           ;PUT INTO LH OF AC2
07404 004404         JSR GTCHR          ;GET NEXT BYTE
07405 173300         ADDS 3,2           ;FORM WORD IN AC2
07406 143000         ADD 2,0            ;ADD INTO CHECKSUM
07407 002415         JMP @TEMP1         ;AND RETURN

                ;READ A BYTE INTO AC3
07410 054415 GTCHR:  STA 3,TEMP2        ;SAVE RETURN
07411 034415         LDA 3,SAVE         ;TEST WHICH DEVICE
07412 175103         MOVL 3,3,SNC
07413 000405         JMP .+5            ;TTI
07414 063612         SKPDN PTR          ;PTR
07415 000777         JMP .-1
07416 074512         DIAS 3,PTR         ;READ AND START
07417 002406         JMP @TEMP2         ;AND RETURN

07420 063610         SKPDN TTI
07421 000777         JMP .-1
07422 074510         DIAS 3,TTI         ;READ AND START
07423 002402         JMP @TEMP2         ;AND RETURN
07424 000000 TEMP1:  0
07425 000000 TEMP2:  0
07426 000000 SAVE:   0

                ;TEST BLOCK TYPE
07427 125224 TEST:   MOVZR 1,1,SZR      ;1=START BLOCK
07430 000405         JMP IGNOR          ;NO, IGNORE BLOCK
07431 101004         MOV 0,0,SZR        ;TEST THE CHECKSUM
07432 000442         JMP CHKER          ;ERROR
07433 063077         HALT               ;HALT
07434 000777         JMP .-1            ;DON'T PROCEED

                ;IGNORE BLOCK
07435 004753 IGNOR:  JSR GTCHR          ;READ UNTIL AN ALL
07436 020404         LDA 0,C377         ;ONES BYTE IS SEEN
07437 106404         SUB 0,1,SZR
07440 000775         JMP IGNOR
07441 000410         JMP BLOCK
07442 000377 C377:   377
```

```
                    ;START OF PROGRAM
07443 062677 BEGIN: IORST           ;RESET
07444 060477        READS 0         ;READ THE SWITCH REGISTER
07445 040761        STA 0,SAVE      ;AND SAVE IT FOR "GTCHR"
07446 060110        NIOS TTI        ;START TTI
07447 060112        NIOS PTR        ;AND PTR
07450 004454        JSR CRLF        ;TYPE CARRIAGE RETURN, L.F.
07451 004737 BLOCK: JSR GTCHR       ;GET A BYTE
07452 165305        MOVS 3,1,SNR    ;AND TEST FOR 0
07453 000776        JMP BLOCK       ;YES, KEEP READING
07454 004734        JSR GTCHR       ;OK, GET NEXT BYTE
07455 167300        ADDS 3,1        ;AND FORM COUNT IN AC1
07456 121000        MOV 1,0         ;SET CHECKSUM
07457 004722        JSR BUILD       ;GET ADDRESS
07460 050443        STA 2,ADDRS
07461 004720        JSR BUILD       ;ADD IN THE CHECKSUM
07462 125113        MOVL# 1,1,SNC   ;TEST BLOCKTYPE
07463 000744        JMP TEST        ;NOT A DATA BLOCK
07464 004715 DATA:  JSR BUILD       ;OK, GET DATA
07465 036436        LDA 3,@ADDRS    ;GET FROM MEMORY
07466 156414        SUB# 2,3,SZR    ;AND COMPARE
07467 000407        JMP ERROR       ;THEY ARE DIFFERENT
07470 010433 DATA0: ISZ ADDRS       ;INCREMENT ADDRESS
07471 125404        INC 1,1,SZR     ;TEST THE COUNT
07472 000772        JMP DATA        ;OK, GET NEXT
07473 101004        MOV 0,0,SZR     ;OK, TEST CHECKSUM
07474 063077 CHKER: HALT            ;CHECKSUM ERROR
07475 000754        JMP BLOCK       ;OK, GET NEXT BLOCK
```

```
                ;A DIFFERENCE WAS FOUND, PRINT IT
07476 040472 ERROR: STA 0,ACSV       ;SAVE THE AC'S
07477 044472        STA 1,ACSV+1
07500 050472        STA 2,ACSV+2
07501 020422        LDA 0,ADDRS      ;GET ADDRESS
07502 004442        JSR OCTAL        ;AND PRINT IT
07503 000002        7-5              ;5 OCTAL DIGITS
07504 024436        LDA 1,SLASH      ;TYPE "/"
07505 004426        JSR TYPE
07506 004424        JSR SPACE        ;TYPE A SPACE
07507 020463        LDA 0,ACSV+2     ;GET CORRECT VALUE
07510 004434        JSR OCTAL
07511 000001        7-6              ;AND TYPE AS 6 DIGITS
07512 024431        LDA 1,ARROW      ;TYPE "←"
07513 004420        JSR TYPE
07514 022407        LDA 0,@ADDRS     ;GET WHAT IS NOW IN CORE
07515 004427        JSR OCTAL        ;AND TYPE AS
07516 000001        7-6              ;6 DIGITS
07517 004405        JSR CRLF         ;TYPE CR, LF
07520 020450        LDA 0,ACSV
07521 024450        LDA 1,ACSV+1     ;RESTORE AC0,AC1
07522 000746        JMP DATA0        ;AND PROCEED
07523 000000 ADDRS: 0

                ;TYPE CR,LF
07524 171000 CRLF:  MOV 3,2          ;SAVE RETURN
07525 024412        LDA 1,C15        ;GET RETURN
07526 004405        JSR TYPE         ;AND PRINT IT
07527 024411        LDA 1,C12        ;GET LINE-FEED
07530 004403        JSR TYPE         ;AND TYPE IT
07531 001000        JMP 0,2          ;AND RETURN

                ;TYPE A SPACE
07532 024407 SPACE: LDA 1,C40        ;GET VALUE

                ;TYPE C(AC1)
07533 063511 TYPE:  SKPBZ TTO        ;WAIT FOR TTO DONE
07534 000777        JMP .-1
07535 065111        DOAS 1,TTO       ;TYPE C(AC1)
07536 001400        JMP 0,3          ;AND RETURN
07537 000015 C15:   15
07540 000012 C12:   12
07541 000040 C40:   40
07542 000057 SLASH: "/
07543 000137 ARROW: "←
```

```
                ;OCTAL PRINTER
                ;AC0=NUMBER TO BE PRINTED
                ;CALL+1=7-NUMBER OF OCTAL DIGITS
                ;CALL+2=RETURN

07544 025400 OCTAL: LDA 1,0,3
07545 044657        STA 1,TEMP1      ;SAVE C(CALL+1)
07546 175400        INC 3,3
07547 054656        STA 3,TEMP2      ;SAVE RETURN
07550 152621        SUBZR 2,2,SKP    ;GET A BIT IN LOCATION 0
07551 142401        SUB 2,0,SKP
07552 024770 OCTL1: LDA 1,SLASH      ;GET "0-1
07553 125400        INC 1,1
07554 142432        SUBZ# 2,0,SZC    ;COMPARE
07555 000774        JMP .-4          ;TOO SMALL, ADD 1 TO DIGIT
07556 004406        JSR OCTL2        ;OUTPUT THIS DIGIT
07557 151220       MOVZR 2,2        ;AND MOVE THE OUTPUT
07560 151220        MOVZR 2,2        ;1 OCTAL DIGIT TO
07561 151224        MOVZR 2,2,SZR    ;THE RIGHT
07562 000770        JMP OCTL1        ;DO THE NEXT
07563 002642        JMP @TEMP2       ;ALL DONE, EXIT

                ;PRINT AC1 IF NOT IGNORING
07564 014640 OCTL2: DSZ TEMP1        ;TEST FOR IGNORING
07565 001400        JMP 0,3          ;YES, RETURN
07566 010636        ISZ TEMP1        ;OK, SET SWITCH
07567 000744        JMP TYPE         ;AND TYPE IT

      000003 ACSV:  .BLK 3

                    .END
```