CO 600 NP
LINC TAPE SYSTEM
FOR NOVA COMPUTERS

REFERENCE MANUAL

COMPUTER OPERATIONS, INC.
10774 TUCKER STREET
BELTSVILLE, MARYLAND 20705

NOVEMBER 24, 1971
DOC. NO. 112
TABLE OF CONTENTS

INTRODUCTION
1.0 GENERAL DESCRIPTION
2.0 SPECIFICATION
3.0 INSTALLATION
4.0 LINCTAPE OPERATION
5.0 BOOTSTRAP
6.0 KEYBOARD EXECUTIVE ROUTINE
7.0 LINCTAPE UTILITIES
8.0 LINCTAPE I/O INSTRUCTIONS
9.0 TAPE FORMAT
10.0 THEORY OF OPERATION
11.0 SYSTEM COMPONENTS
12.0 POWER SUPPLY
13.0 MAINTENANCE

APPENDIX
A BOOTSTRAP LISTING
B UTILITY LISTING
C KEYBOARD EXEC LISTING
D STAND-RD TAPE MAP

REFERENCES

(1) "HOW TO USE THE NOVA COMPUTERS", DATA GENERAL CORPORATION, DG-NM
(2) "SCHEMATIC DIAGRAMS OF THE LINCTAPE/NOVA SYSTEM", COMPUTER OPERATIONS, INC.

COI DR. NO SCHEMATIC
D-10230-01 LINCTAPE MASTER CONTROLLER
 (D-10144-01 FOR S/N BELOW 1016)
B-10164-01 LINCTAPE DRIVE CONTROL BOARD
D-10198-01 LINCTAPE READER/WRITER BOARD
 (D-10232-01 FOR S/N BELOW 1016)
D-10244-01 LINCTAPE POWER SUPPLY
INTRODUCTION

THIS MANUAL IS DESIGNED TO ASSIST OPERATORS, PROGRAMMERS, AND MAINTENANCE PERSONNEL IN THE USE OF THE LINCTAPE SYSTEM WITH A NOVA, SUPER NOVA, NOVA 800 SERIES, OR NOVA 1200 SERIES COMPUTER. A KNOWLEDGE OF THE OPERATION OF THE NOVA IS ASSUMED.

NO DATA GENERAL OPTIONS, EXCEPT THE I/O CONNECTOR ARE REQUIRED. HOWEVER, IT IS ASSUMED THROUGHOUT THIS MANUAL, THAT A TELETYPE * OR EQUIVALENT I/O DEVICE IS AVAILABLE. THE I/O CONNECTOR IS STANDARD ON THE 1210, 1220, AND 820 NOVA COMPUTERS.

* TELETYPE IS A TRADEMARK OF THE TELETYPE CORPORATION.
1.0 GENERAL DESCRIPTION

THE CO-600-NP LINCTAPE SYSTEM CONSISTS OF TWO (OR MORE) DRIVES AND A COMMUN ELECTRONICS SYSTEM. EACH DRIVE HANDLES ONE REEL OF LINCTAPE.

EACH TAPE IS DIVIDED INTO SECTORS OR BLOCKS. THERE ARE 400 (620 OCTAL) SUCH BLOCKS PER TAPE, NUMBERED 0 THRU 399. EACH BLOCK CONTAINS 256 WORDS OF 16 BITS EACH. EACH BLOCK IS ADDRESSABLE AND THE TRANSFERRING OF DATA (READING OR WRITING) IS DONE ONE OR MORE BLOCKS AT A TIME. THUS, LINCTAPE IS MORE AKIN TO A DISC THAN IT IS TO INDUSTRY COMPATIBLE TAPE.

FOR INSTANCE, ASSUME THAT IT IS DESIRABLE TO SAVE A DATA BUFFER THAT EXISTS IN CORE FROM LOCATION 3000 THRU 3777 (OCTAL). IT CAN BE WRITTEN OUT ONTO LINCTAPE AS FOLLOWS:

```
SUB 0,0
DOB 0,LINC SELECT DRIVE NO 0
LDA 0,BLKNO LOAD AC0 WITH 1ST BLOCK NUMBER
LDA 1,NBLKS LOAD AC1 WITH NUMBER OF BLOCKS
LDA 2,FCORE LOAD AC2 WITH 1ST CORE LOCATION
JSR@ WLINC JUMP TO THE WRITE UTILITY SUBROUTINE
```

```
BLKNO: 100
NBLKS: 2
FCORE: 3000
WLINC: X7406
RLINC: X7403
```

THIS ROUTINE WILL WRITE 2 BLOCKS (1000 OCTAL WORDS), STARTING AT CORE LOCATION 3000, ONTO THE LINCTAPE THAT IS ON DRIVE 0, STARTING WITH BLOCK NUMBER 100.

TO READ THE DATA BACK INTO CORE, THE SAME PROCEDURE WOULD BE USED, EXCEPT THAT THE LAST INSTRUCTION WOULD BE "JSR@ RLINC".

ONE OF THE ADVANTAGES OF LINCTAPE OVER INDUSTRY COMPATIBLE TAPE SYSTEMS IS THE ABILITY TO OVERWRITE A BLOCK. THE SELF-SYNCHRONIZING FEATURES OF LINCTAPE ALLOW INFORMATION TO BE READ, UPDATED, AND RE-WRITTEN BACK ON THE SAME PLACE IN TAPE.
Among the major uses of LinCTape are:

1. Bootstrapping. This allows starting the computer from scratch by loading the Linc utilities and other desired programs into core.

2. Library storage. Library programs, such as the assembler debug routines, and diagnostics, can be stored on tape and loaded into core quickly at any time.

3. Program storage. User programs can be stored on LinCTape for rapid, direct access when needed. Three parameters: block location, core location, and number of blocks will specify all that is needed to load any of the programs.

4. Program debugging. When debugging programs, the current program can be saved on LinCTape. If the program destroys core, it can be quickly reloaded. If it requires updating, it can be loaded into core, updated, and written back onto tape easily. If experimental program modifications are to be tried, the original, and later iterations, of the program can be saved.

5. Program overlay. Large programs can be broken into smaller sections and loaded into core a section at a time. Thus it is often possible to utilize a machine with limited core to run large programs using LinCTape.

6. Data storage. Data blocks can be stored on, or read from LinCTape in the standard manner. Writing "in place" often allows the use of one tape, instead of the usual two that are required for industry compatible systems.

7. Data merging. Merging two data buffers can usually be done with one dual LinCTape system, rather than three industry compatible systems.
2.0 SPECIFICATIONS

2.1 SYSTEM
DUAL TAPE DRIVE
EXPANDABLE TO 16 DRIVES (8 DUAL)
60 IPS TAPE SPEED IN EITHER DIRECTION
BI-DIRECTIONAL BLOCK SEARCH AT FULL SPEED
UPDATE ANY SECTOR(S) DIRECTLY
WRITE PROTECT ON EACH DRIVE
25 SECONDS END TO END TRAVERSE TIME
8.5 SECONDS AVERAGE ACCESS TIME
130 MS START/STOP/REVERSE TIME
16 BIT PARALLEL INTERFACE
ALGEBRAIC CHECKSUM FOR EACH BLOCK
PERMANENT, PRE-RECORDED SECTOR ADDRESSES

2.2 TAPE
SCOTCH CAT. NO. 481-3/4-150-R42 (UNMARKED)
150 FEET LONG
3/4 INCH WIDE
10 TRACKS, 35 MIL, FULLY REDUNDANT
SANDWICH TAPE FOR LONG TAPE AND HEAD LIFE
3-13/16 INCH DIAMETER REEL
400 BPI PHASE RECORDING
16 BIT WORD
102,400 WORDS PER REEL (204,800 BYTES)
4200 WORDS/SECOND TRANSFER RATE (8400 BYTES/SECOND)
400 DATA BLOCKS
256 WORDS/BLOCK

2.3 PHYSICAL
8-3/4"H X 19"W X 12-3/16"D RACK MOUNT
2-1/8 FRONT PROJECTION
10-1/16 REAR PROJECTION (EXCLUDING PLUGS)
105-125 VOLTS, 60 HZ, 100 WATTS (50 HZ AVAILABLE)
35 POUNDS

2.4 MOUNTING
THE LINCTAPE IS DESIGNED FOR STANDARD RACK MOUNTING. THE DIMENSIONS ARE SHOWN IN FIGURE 2.1. IT IS ADVISABLE TO PROTECT THE HEADS, GUIDES, AND TAPE FROM DIRT AND DUST BY KEEPING THE FRONT DOOR CLOSED IN OPERATION. HOWEVER, IN A LABORATORY ENVIRONMENT, THE DOOR MAY BE REMOVED BY TWO SCREWS NEAR THE HINGE.

POWER, I/O, AND SLAVE CONNECTIONS ARE MADE IN THE REAR OF THE UNIT. ACCESS TO THE ELECTRONIC BOARDS IS MADE BY SWINGING THE REAR PANEL OR BY REMOVING THE REAR COVER OR THE TOP COVER.
Figure 2.1
Outline Dimensions
2.5 PACKING LIST

CO-600-6P MASTER LINCTAPE UNIT
1 EA DUAL LINCTAPE DRIVE SYSTEM WITH HEAD/WRITE ELECTRONICS
1 EA LINE CORD
1 EA I/O CABLE
2 EA REELS OF MARKED LINCTAPE
1 EA INSTRUCTION BOOK

CO-605 SLAVE UNIT
1 EA DUAL LINCTAPE SLAVE DRIVE SYSTEM
1 EA LINE CORD
1 EA SLAVE ADAPTER CABLE
2 EA REELS OF MARKED TAPE
1 EA INSTRUCTION BOOK

NOTE: ADDITIONAL LINCTAPES, PRE-MARKED AND CERTIFIED, MAY BE OBTAINED FROM COMPUTER OPERATIONS, INC.
3.0 INSTALLATION

THE CPU MUST HAVE THE I/O CONNECTOR OPTION FOR PROPER INSTALLATION:
- NUVA
- DATA GENERAL TYPE 4022
- SUPERNova " " " 8022
- NUVA 800 " " " 8222
- NUVA 1200 " " " 8122
- STANDARD ON THE 1210, 1220, AND 820.

3.1 MOUNT THE LINCTAPE. IF THE LINCTAPE IS TO BE RACK MOUNTED:
A. REMOVE THE DOOR BY REMOVING THE TWO SCREWS HOLDING THE HINGE BLOCK TO THE FRONT PANEL. BE CAREFUL NOT TO SCRATCH THE HEADS OR THE GUIDES.
B. PLACE THE LINCTAPE IN POSITION AND FASTEN FIRMLY IN PLACE USING FOUR SCREWS THRU THE FLANGE SLOTS. IT IS ADVISABLE TO PROTECT THE FRONT PANEL FINISH BY USING A FIBER OR PLASTIC WASHER UNDER THE SCREW HEADS.
C. REPLACE THE DOOR, BEING SURE IT IS POSITIONED TO CLOSE PROPERLY.

3.2 CONNECT THE POWER CABLES. IF IT IS DESIRED TO HAVE THE COMPLETE SYSTEM TURN ON AND OFF WITH THE COMPUTER POWER SWITCH:
A. TURN OFF ALL POWER TO THE SYSTEM.
B. PLUG THE LINCTAPE POWER CABLE INTO THE OUTLET IN THE BACK OF THE COMPUTER.
C. PLUG THE NEXT PERIPHERAL DEVICE (SUCH AS THE TELETYPE) INTO THE OUTLET IN THE BACK OF THE LINCTAPE. REFER TO THE DATA GENERAL MANUAL TO DETERMINE THE PERMISSIBLE LOAD. LINCTAPE DRAWS ABOUT 1 AMP.

3.3 CONNECT THE I/O CABLES. THIS CONNECTION DEPENDS ON THE OTHER PERIPHERALS IN THE SYSTEM. ONE OF THE FOLLOWING CONNECTIONS SHOULD BE MADE:
A. IF THE LINCTAPE IS THE ONLY PERIPHERAL ON THE I/O CONNECTOR, SIMPLY CONNECT THE I/O CABLE (SUPPLIED WITH LINCTAPE) BETWEEN THE COMPUTER I/O CONNECTOR AND EITHER OF THE LINCTAPE I/O CONNECTORS. LEAVE THE REMAINING CONNECTOR ON THE LINCTAPE OPEN.
B. IF OTHER PERIPHERALS ARE TO BE CONNECTED ON THE SAME LINE, THEY CAN BE "DAISY CHAINED" BY USING BOTH CONNECTORS ON THE LINCTAPE. IN THIS CASE, CONNECT THE I/O CABLE BETWEEN THE COMPUTER AND LINCTAPE AS IN "A" ABOVE. THEN CONNECT THE NEXT PERIPHERAL CABLE TO THE REMAINING I/O CONNECTOR ON THE LINCTAPE. AT THE END OF THIS CHAIN, AN I/O TERMINATOR SHOULD BE USED. POWER FOR THE TERMINATOR MUST BE SUPPLIED BY THE LAST PERIPHERAL. SEE THE "HOW TO" MANUAL, REFERENCE 1, APPENDIX A.
TO NEXT PERIPHERAL (IF ANY)

I/O TO TELETYP Et

I/O CABLE

I/O LINE CORD

I/O CONNECTOR*

I/O TO TELETYP E

I/O INPUT

* NOVA 1200 OPTION 8122
NOVA 800 OPTION 8222
SUPER NOVA OPTION 8022
NOVA OPTION 4022

FIGURE 3.1
CABLE DIAGRAM
3.4 Slave Unit Installation. If one or more slave units are to be connected, they may be strung in "Daisy Chain" fashion. As many as seven (7) slaves may be so connected, or a total of 16 drives. See Figure 3.2.

The two drives on the master unit are connected as numbers 0 and 1. (Even numbered drives are always on the left and odd numbered ones on the right when facing the front.) Slave units are normally wired to be numbers 2 and 3 at the factory. If additional slaves are added, it may be necessary to change the jumpers on the drive control board to suit. It is important that no two units have the same drive numbers. See Figure 3.3.

To change the drive number for any unit, the following steps are required:
1. Disconnect all power;
2. Remove the 6 screws on the top of the linetape and remove the cover;
3. Disconnect the three edge connectors from the top of the drive control board and carefully remove the board;
4. Note the position of the three jumper wires near the top center of the board. They are marked, and the sum of the markings indicates the drive number of the left hand drive. The right drive is, of course, one number higher. See Figure 3.3.
5. Carefully unsolder and change the necessary jumpers to suit. Be careful not to damage the pads.
6. Replace the board, the connectors, and the top cover.
FIGURE 3.2
SLAVE CONNECTIONS
Drive numbers 4 & 5 shown

Figure 3.3
Drive number jumpers
4.0 LINCTAPE OPERATION

THE LINCTAPE SYSTEM HAS TWO DRIVES WHICH MAY BE OPERATED SEPARATELY. THE LEFT DRIVE IS NUMBERED 0 (ZERO) AND THE RIGHT DRIVE IS NUMBERED 1 (ONE). THE TAPE IS ALWAYS MOUNTED ON THE RIGHT (OR SOURCE) HUB OF EACH DRIVE. THE LEFT HUB HAS THE TAKEUP REEL WHICH SHOULD NOT BE REMOVED.

EACH DRIVE HAS THREE BUTTONS ASSOCIATED WITH IT: LOAD, REWIND, AND WRITE PROTECT. NEGLECTING WRITE PROTECT, THERE ARE FOUR MODES OF OPERATION POSSIBLE FOR EACH DRIVE: OFF, FORWARD, REVERSE, AND TENSION. WHEN POWER IS FIRST APPLIED, BOTH DRIVES WILL BE IN THE OFF MODE.

THE LOAD (OR LEFT) BUTTON IS THE WHITE MOMENTARY CONTACT BUTTON ON THE LEFT SIDE OF EACH DRIVE. IT WILL MOVE THE TAPE FORWARD. WHEN RELEASED, THE DRIVE WILL BE IN THE TENSION MODE. THE DRIVE MUST BE IN THE TENSION MODE BEFORE THE COMPUTER CAN ACCESS IT. THE OPERATOR SHOULD CHECK THE TENSION IF THE LINCTAPE FAILS TO RESPOND TO COMPUTER COMMANDS.

THE REWIND (OR RIGHT) BUTTON IS THE WHITE MOMENTARY CONTACT BUTTON ON THE RIGHT SIDE OF EACH DRIVE. IT WILL MOVE THE TAPE IN THE REVERSE DIRECTION. WHEN RELEASED, THE DRIVE WILL BE IN THE OFF MODE.

THE WRITE PROTECT BUTTON IS THE RED ALTERNATE ACTION BUTTON LOCATED UNDER EACH HEAD. WHEN IT IS LIGHTED, IT IS IMPOSSIBLE FOR THE COMPUTER TO WRITE ON THE TAPE, AND THE DRIVE IS THUS PROTECTED FROM ACCIDENTAL WRITING. THE BUTTON HAS NO EFFECT ON READING.

TO MOUNT A TAPE ON EITHER DRIVE, PRESS THE REEL FIRMLY OVER THE HUB UNTIL IT SNAPS INTO PLACE. PASS THE END OF THE TAPE OVER THE GUIDES AND HEAD AND LAY IT ON THE TAKEUP REEL. HOLD THE TAPE AGAINST THE TAKEUP REEL AND WIND IT ON A COUPLE OF TURNS. TURN THE REELS TO TAKE UP ANY SLACK, AND PRESS THE LOAD BUTTON BRIEFLY. THE REELS SHOULD STOP WITH THE TAPE IN TENSION.

TO UNLOAD A TAPE, HOLD THE REWIND (OR RIGHT) BUTTON UNTIL THE TAPE UNWINDS. PULL THE REEL OFF BY PRESSING AGAINST THE HUB AND PULLING ON THE REEL UNTIL THE REEL SNAPS OFF.
CAUTION: CERTAIN PRECAUTIONS SHOULD BE OBSERVED CONCERNING ANY MAGNETIC TAPE SYSTEM, LINCTAPE INCLUDED:
(1) BE CAREFUL OF THE HEADS AND GUIDES. DO NOT SCRATCH THEM;
(2) KEEP THE TAPES, HEADS, AND GUIDES CLEAN.
SEE SECTION 13;
(3) KEEP TAPES AWAY FROM STRONG MAGNETIC FIELDS, SUCH AS MIGHT BE FOUND NEAR TRANSFORMERS, MOTORS, FLUORESCENT LIGHT BALLASTS, ETC.
(4) DO NOT USE STICKY TAPE, SUCH AS "SCOTCH" TAPE OR ADHESIVE LABELS, ON THE TAPE.
IN TIME, THE STICKY MATERIAL TENDS TO SEEP OUT, AND DISTRIBUTE ITSELF OVER THE TAPE AND GUIDES, CAUSING DROPOUTS. TO IDENTIFY A TAPE, PUT LABELS ON THE REEL.
FIGURE 4.1
FRONT PANEL CONTROLS
5.0 BOOTSTRAP

THE LINCTAPE BOOTSTRAP CAN BE USED WITH ANY OF THE NOVA SERIES COMPUTERS. IT ALLOWS QUICK AND EASY LOADING OF PROGRAMS FROM LINCTAPE.

THERE ARE TWO DIFFERENT PROCEDURES, DEPENDING ON WHETHER THE COMPUTER HAS A HARDWARE PROGRAM LOADER OR NOT.

THE PROCEDURE FOR THESE BOOTSTRAPS ARE DESCRIBED IN DETAIL IN APPENDIX A.
6.0 KEYBOARD EXECUTIVE ROUTINE.

THE KEYBOARD EXECUTIVE ROUTINE IS DESIGNED TO TRANSFER DATA AND PROGRAMS BETWEEN CORE AND LINCTAPE VIA THE KEYBOARD WITH A MINIMUM OF OPERATOR EFFORT. IT HAS TWO MODES OF OPERATION: GENERAL AND AUTOMATIC.

6.1 GENERAL MODE.

THE GENERAL MODE ALLOWS ANY NUMBER OF CONTIGUOUS BLOCKS OF DATA OR INFORMATION TO BE READ FROM OR WRITTEN ONTO LINCTAPE. THE FORMAT, WHICH REQUIRES 5 PARAMETERS, IS:

```
*1000,10,4,0R
```

WHERE

* IS THE RESPONSE GIVEN BY THE KEYBOARD EXEC WHEN WAITING FOR AN OPERATOR INPUT
1000 IS THE FIRST CORE LOCATION
10 IS THE FIRST BLOCK NUMBER
4 IS THE NUMBER OF BLOCKS
0 IS THE DRIVE NUMBER
R MEANS READ FROM TAPE INTO CORE.

THIS STRING, WHEN TYPED INTO THE KEYBOARD EXEC, WILL READ THE CONTENTS OF BLOCKS 10 THRU 13 FROM THE TAPE ON DRIVE 0 INTO CORE LOCATIONS 1000 THRU 2777. (NOTE THAT ALL NUMBERS ARE IN OCTAL.) COMMAS ARE USED TO SEPARATE THE PARAMETERS, EXCEPT NO SEPARATOR IS NEEDED BETWEEN THE DRIVE NUMBER AND THE COMMAND LETTER.

READ (R) MEANS TO TRANSFER BLOCKS OF DATA FROM LINCTAPE AND STORE THEM IN CORE. PREVIOUS DATA IN THE SPECIFIED AREAS OF CORE ARE, OF COURSE, LOST. THE DATA ON TAPE IS NOT ALTERED.

WRITE (W) MEANS TO TRANSFER BLOCKS OF INFORMATION FROM CORE AND WRITE THEM ON LINCTAPE. PREVIOUS DATA IN THE SPECIFIED BLOCKS OF TAPE ARE LOST, BUT CORE IS NOT ALTERED. AFTER WRITING, THE TAPE IS CHECKED ON A SECOND PASS.

CHECK (C) DOES NOT TRANSFER INFORMATION, BUT IT DOES CHECK AREAS OF TAPE FOR PROPER CHECKSUMS. IT IS PRIMARILY USED FOR TESTING PURPOSES.

THE KEYBOARD EXEC REQUIRES THE UTILITY SUBROUTINES TO BE IN CORE. NORMALLY, BOTH ARE RESIDENT. TO RE-ENTER THE KEYBOARD EXEC AT ANY TIME, START THE COMPUTER AT THE ENTRY LOCATION X7000 (X7777 IS THE HIGHEST CORE LOCATION).
PRECAUTIONS AND LIMITATIONS

1. THERE ARE THREE COMMANDS: READ(R), WRITE(W), AND CHECK(C);
2. ALL PARAMETERS MUST BE IN OCTAL;
3. EACH BLOCK CONTAINS 400 (256 DECIMAL) WORDS. WHOLE BLOCKS ARE ALWAYS TRANSFERRED;
4. NEGATIVE NUMBERS MUST BE ENTERED AS TWO'S COMPLEMENT NUMBERS:
   \( 177776 = -2 \)
5. BLOCK NUMBERS BELOW -6 (177772) ARE NOT ACCEPTABLE TO THE KEYBOARD EXEC, NOR ARE BLOCKS ABOVE 617 (OCTAL);
6. OVERFLOW BITS ARE IGNORED. THAT IS, ONLY THE LOWER 16 BITS OF THE TYPED NUMBER ARE USED:
   \( 17777776 = 177776 = -2 \)
7. NUMBERS NOT TYPED ARE ASSUMED TO BE ZERO:
   \( 1000, 2, R \) MEANS CORE LOC 1000, STARTING BLOCK ZERO 2 BLOCKS, DRIVE 0, READ.
8. IT IS THE OPERATOR'S RESPONSIBILITY NOT TO READ OVER THE KEYBOARD EXEC OR THE UTILITIES. THESE RESIDE IN LOCATIONS X7000 THRU X7377 AND X7400 THRU X7777 RESPECTIVELY.
6.2 AUTOMATIC MODE

Clearly, in the General Mode, it is necessary that the operator know where particular programs or data are stored on tape. The Automatic Mode allows frequently used programs to be read (but not written) quickly, without the need to know exactly where they are stored on tape. For instance, typing "A" into the Keyboard Exec could load the assembler.

Each type of tape (System, program, etc.) may have different programs, each with its own code letters. The core location and block numbers are stored in a table in the Keyboard Exec. The proper tape must be on drive 0 when using the Automatic Mode. This Automatic Mode will load, but will not execute the called program. The operator must do this separately. This allows several programs to be loaded at one time and executed selectively.

The command letters and their associated parameters may be modified as programs are added or deleted from the tape. See the Exec Listing for details.

Linctapes, when supplied by COI, normally contain no programs on tape other than the bootstrap, utilities, and Keyboard Exec. The Automatic Mode table is empty.

6.3 COMPUTER RESPONSE.

After the transfer takes place, the teletype will respond with another "*", indicating that the previous command was executed, and the Exec is waiting for another operator command.

If an error occurs, the teletype will print a "?" and then the "*". The possible errors that may occur include:

1. Drive not ready (not in tension mode),
2. A write attempt was made on a protected drive,
3. The command letter, either General or Automatic, is not valid,
4. A non-octal number was typed,
5. A non-valid character (such as space) was typed,
6. A non-valid block was requested (valid blocks are 177772 thru 617, inclusive),
7. The tape is bad, containing erroneous checksum, block number, etc.
7.0 LINCTAPE UTILITIES

THE LINCTAPE UTILITY SUBROUTINES ARE A SET OF SOFTWARE WHICH MAKE IT EASY FOR THE PROGRAMMER TO COMMUNICATE WITH THE LINCTAPE. THEY ALLOW READING AND WRITING WITH A FEW SIMPLE INSTRUCTIONS, AND RELIEVE THE PROGRAMMER OF THE DETAILS OF TIMING, BLOCK SEARCHING, ETC. ONLY DRIVE SELECTION IS LEFT UP TO THE PROGRAMMER.

NORMALLY, THE UTILITIES ARE LOADED NEAR THE TIP OF CORE, AND ARE NEVER CHANGED (THEY ARE SAID TO BE RESIDENT). THEY OCCUPY LOCATIONS X7400-X7577 (X7777 IS THE LAST LOCATION IN CORE). ALL CALLS MUST BE MADE WITH A 'JSR#' STATEMENT TO ONE OF THE FOLLOWING ENTRY POINTS:

CLINC  X7400  CHECK BLOCKS
RLINC  X7403  READ & CHECK BLOCKS
WLINC  X7406  WRITE & CHECK BLOCKS

WITH THE REGISTERS LOADED AS FOLLOWS:
AC0 = FIRST BLOCK NUMBER TO BE PROCESSED,
AC1 = NUMBER OF BLOCKS TO BE PROCESSED, AND
AC2 = FIRST CORE LOCATION.

IF AC1 = 0, THE TAPE WILL PRE-POSITION ITSELF NEAR THE BLOCK NUMBER SPECIFIED IN AC0.
IF AC2 IS NEGATIVE, THE DRIVE WILL START BACKWARD (THUS SAVING TIME IF IT IS KNOWN THAT THE BLOCK TO BE FOUND HAS BEEN PASSED), AND IT WILL TAKE THE 1'S COMPLEMENT (NOT THE NEGATIVE) OF THE NUMBER IN AC2 AS THE FIRST CORE LOCATION.

THE UTILITIES WILL RETURN TO THE PROGRAM WITH THE DRIVE STOPPED, AND THE REGISTERS CONTAINING THE FOLLOWING INFORMATION:
AC1 = 0 FOR NORMAL (NON-ERROR) RETURN
AC2 = NEXT BLOCK NUMBER
AC3 = NEXT CORE LOCATION

IF THERE IS AN ERROR, AC1 CONTAINS THE ERROR CODE:
AC1 = 1 IF THERE WAS A CHECKSUM ERROR:
AC0 = BAD BLOCK NUMBER
AC1 = 2 IF THERE WAS A BLOCK SIZE ERROR:
AC0 = BAD BLOCK NUMBER
AC2 = EXCESS OF WORDS IN BLOCK
AC3 = EXPECTED NUMBER OF WORDS
AC1 = 4 IF THERE WAS AN ILLEGAL BLOCK CALLED FOR
(*6 THRU 617 ARE LEGAL NORMALLY):
AC0 = TARGET BLOCK
AC2 = NEXT CORE LOCATION
AC3 = HIGHEST LEGAL BLOCK
AC1 = 8 IF THERE WAS A DRIVE STATUS ERROR:
AC3 = DRIVE STATUS
BIT 15 ON = DRIVE NOT READY (TENSION)
BIT 14 ON = WRITE ATTEMPTED ON PROTECTED DRIVE
THE CALLING SEQUENCE IS (ASSUMING .RDX = 8):

LINC  = 74 ; LINCTAPE DEVICE NUMBER

LDA  0,DRVNO ; SELECT DRIVE NUMBER
DOB  0,LINC

LDA  0,FBLKN ; SET AC0 = 1ST BLOCK NO
LDA  1,NBLKS ; SET AC1 = NO OF BLOCKS
LDA  2,FCORE ; SET AC2 = 1ST CORE LOC

JSR@ RLINC ; READ THE BLOCKS

MOV  1,1,SZR ; TEST FOR ERROR

WLINC: X7406 ; WRITE POINTER
RLINC: X7403 ; READ POINTER
FBLKN: 100
NBLKS: 10
FCORE: 1000
DRVNO: 1

THE ABOVE WILL READ THE CONTENTS OF BLOCK NUMBERS 100 THRU 107 FROM THE TAPE ON DRIVE 1 INTO CORE LOCATIONS 1000 THRU 4777 (ALL IN OCTAL, AND ASSUMING STANDARD 400 WORD BLOCKS).
8.0 LINCTAPE I/O INSTRUCTIONS

USUALLY, THE STANDARD LINCTAPE UTILITIES WILL HANDLE ALL
NECESSARY COMMUNICATION BETWEEN THE LINCTAPE AND THE COMPUTER.
IT IS RECOMMENDED THAT, WHENEVER POSSIBLE, THESE BE USED.

HOWEVER, IN THOSE CASES WHERE THEY ARE INADEQUATE, OR WHERE SPECIAL
CONDITIONS REQUIRE CHANGES, THE ACTUAL I/O COMMANDS AND LIMITATIONS
ARE GIVEN IN THIS SECTION.

THE LINCTAPE IS A PERIPHERAL DEVICE, WHICH IS ADDRESSED IN THE
SAME MANNER AS ALL OTHER PERIPHERALS, THRU THE I/O COMMANDS.
REFER TO THE "HOW TO USE" MANUAL, REFERENCE 1. THE COMMANDS
NECESSARY FOR OPERATION OF THE LINCTAPE INCLUDE THE FOLLOWING:

DRIVE COMMANDS
  SELECT DRIVE NUMBER
  START DRIVE FORWARD
  START DRIVE BACKWARD
  STOP DRIVE

STATUS
  DRIVE READY
  WRITE PROTECT
  BLOCK NUMBER READY
  DATA WORD READY
  CHECKSUM READY

DATA TRANSFER
  INPUT DATA WORD
  OUTPUT DATA WORD
  TURN WRITERS ON

SPECIFICALLY, THE FOLLOWING INSTRUCTIONS ARE APPLICABLE:

LINCE = DEVICE NUMBER 74 (OCTAL) NORMALLY

DATA I/O.
  DIA X,LINCE INPUT THE 16 BIT DATA WORD TO ACX.
  DOA X,LINCE OUTPUT THE 16 BIT WORD IN ACX TO THE LINCTAPE.
  DIB X,LINCE INPUT THE STATUS BITS TO ACX.
    BIT 15 ON = NOT READY
    BIT 14 ON = WRITE PROTECTED
  DOB X,LINCE SELECT DRIVE NUMBER. THE DRIVE NUMBER MUST BE IN
    THE 4 LOW ORDER BITS OF ACX.
  DOC X,LINCE TURN WRITERS ON. THE WRITERS ARE AUTOMATICALLY
    TURNED OFF AT THE END OF EACH BLOCK. THE VALUE
    OF X IS ARBITRARY. ANY ACX MAY BE USED.

CONTROL PULSES. THESE MAY BE COMBINED WITH ANY OF THE I/O
COMMANDS IN THE USUAL MANNER:

S START THE SELECTED DRIVE FORWARD;
P START THE SELECTED DRIVE BACKWARD;
C STOP THE DRIVE. THE DRIVE DOES NOT STOP AUTOMATICALLY
  AT THE END OF A BLOCK. THIS ALLOWS RAPID TRANSFER OF
  SUCCESSIVE BLOCKS, BUT IT MEANS THE PROGRAM MUST STOP THE
  DRIVE. IT SHOULD NOT BE STOPPED UNTIL THE NEXT BLOCK
  NUMBER IS READY, ESPECIALLY ON WRITING.
SKIP LINES. THERE ARE FOUR POSSIBLE CONDITIONS: (1) BLOCK NUMBER READY, (2) DATA READY, (3) CHECKSUM READY, OR (4) NONE OF THEM READY.

SKIP LINC SKIP IF DATA OR CHECKSUM IS READY;
SKIPZ LINC SKIP IF BLOCK OR NONE IS READY;
SKIPN LINC SKIP IF BLOCK NUMBER OR CHECKSUM IS READY;
SKPHZ LINC SKIP IF DATA OR NONE READY.

NOTE THAT IT REQUIRES TWO SKIP COMMANDS TO DETERMINE WHETHER A SPECIFIC TYPE OF WORD IS READY. NOTE ALSO THAT ALL COMMANDS EXCEPT DRIVE SELECT, APPLY TO THE SELECTED DRIVE ONLY.

SPECIAL

IORST STOP DRIVE AND SELECT DRIVE 0;
FRONT PANEL RESET IS THE SAME AS IORST.

FOR EXAMPLES OF THIS PROGRAMMING, REFER TO THE LISTINGS OF THE LINCTAPE UTILITIES.

TIMING

IN MOST CASES, TIMING IS NOT CRUCIAL. OCCASIONALLY, HOWEVER IT IS IMPORTANT THAT THE PROGRAMMER KNOW SOME OF THE BASIC TIMES INVOLVED. THE FOLLOWING ARE APPROXIMATE, AND MAY VARY BY 10% IN THE FORWARD DIRECTION AND 30% IN THE REVERSE DIRECTION. SEE THE DIAGRAM IN FIGURE 10.2.

- TAPE END TO END: 25 SECONDS
- START/TURN AROUND TIME: 130 MILLISECONDS
- BLOCK TO BLOCK: 63 MILLISECONDS (256 WORD BLOCKS)
- WORD TO WORD: 240 MICROSECONDS
- READY (BLOCK,DATA,CKSM TRANSFER): 40 MICROSECONDS (RESET BY DATA TRANSFER). MUST INPUT OR OUTPUT THE WORD DURING THIS TIME.
IN GENERAL, THE FOLLOWING RULES SHOULD BE OBSERVED:

1. DRIVE SELECTION SHOULD PRECEDE ANY MOTION;

2. DRIVE SELECTION SHOULD BE FOLLOWED BY A STATUS CHECK TO SEE IF (1) THE DRIVE IS READY AND (2) IT IS NOT PROTECTED WHEN WRITING;

3. DRIVE MAY BE STARTED FORWARD OR BACKWARD;

4. AN ACCELERATION DELAY OF 130 MS ALWAYS OCCURS WHEN STARTING OR CHANGING DIRECTION;

5. MARKS ARE TRUE FOR 36 TO 44 US FORWARD, AND 30 TO 50 US WHEN MOVING BACKWARD;

6. "BUSY" AND "DONE" ARE ENCODED FOR MARK SENSING. THEY ARE RESET WHENEVER A DIA OR DOA COMMAND IS SENT, SO A MARK CANNOT BE DOUBLY DETECTED.

7. FOLLOWING DETECTION OF ANY MARK IN A SKIP LOOP, A DIA OR DOA MUST BE ISSUED TO CLEAR THE DONE AND BUSY LINE, EVEN IF THE MARK WAS NOT THE TYPE DESIRED.

8. MOTION COMMANDS MAY BE REPEATED WHILE MOVING WITHOUT CAUSING ACCELERATION DELAY. MOTION MAY BE REVERSED WITHOUT STOPPING.

9. "WRITERS ON" COMMAND MUST NEVER BE GIVEN WHILE THE DRIVE IS GOING BACKWARD;

10. "WRITERS ON" COMMAND MUST BE GIVEN AFTER THE BLOCK MARK IS SENSED, BUT BEFORE THE FIRST DATA MARK COMES TRUE. THIS IS 220 US NOMINALLY.

11. AFTER A WRITE, MOTION MAY NOT BE CHANGED OR STOPPED UNTIL THE NEXT BLOCK MARK IS DETECTED. IN GENERAL, MOTION DECISIONS SHOULD BE MADE ONLY ON BLOCK MARKS.

12. SHOULD THE DRIVE STATUS CHANGE TO "NOT READY" DURING THE PROGRAM OPERATION, TAPE MOTION IS CLEARED AND MARKS CANNOT BE DETECTED UNTIL THE DRIVE IS RE-TENSIONED AND FORWARD OR START BACKWARD COMMAND IS GIVEN;
9.0 LINCTAPE FORMAT

To understand the operation of the linctape, it is necessary to visualize the format on the tape. This section describes the linctape itself.

9.1 Physical Description
The tape is 150 feet long, and \( \frac{3}{4} \) inch wide. It is about 0.0015 inches thick, and the oxide is sandwiched between two layers of mylar.

The tape is divided, by its pre-written markings, into three sections: front leader, data blocks, and trailer. See Figure 9.1.

9.2 Front Leader
The front leader has three parts: blank area, end zone, and preliminary blocks.

9.2.1 Blank Area
This is a small area at the beginning of the tape which has no marks of any kind. It is about two feet long.

9.2.2 End Zone
The end marks at the front of the tape are used to assure that the tape, once up to speed, has synchronizing information. It is essential, when bootstrapping, that the tape be started in this area so that the first block will be properly loaded. Thus this end zone must be long enough that the operator will not easily overshoot it when manually loading. A visual marker on the tape helps to assure this. This end zone is several feet long.

9.2.3 Preliminary Blocks
The first few blocks are used for bootstrapping and for storing special programs. They are given negative block numbers, so the programmer can normally use all positive blocks without destroying this area. Standard tapes have 8 such blocks, numbered -8 thru -1. Otherwise, these blocks are identical to data blocks.

9.3 Data Blocks
The main data on tape is contained in 400 addressable blocks, each containing 256 (400 octal) 16 bit words. Each block also contains its checksum, its own block number, and other housekeeping information. Each block is about 4 inches long.

9.3.1 Block Numbers
There are two block numbers for each block. One is at the beginning of each block and can be read in the forward direction. The other is at the end of each block and can be read only in the reverse direction.

9.3.2 Data Word
Each 16 bit data word occupies six longitudinal characters on tape. There are three data tracks and one mark track. This allows for 18 data bits, of which only 16 are used, and for six mark bits.
9.3.3 CHECKSUM
This is the sum, modulo 65,536 (2**16), of all words in a given block. It is obtained by simply adding each word in the computer and ignoring any overflow. This word is written on the tape immediately following the last data word in each block.

9.4 TRAILER
The trailer, like the leader, has three zones: final blocks, end zone, and blank tape.

9.4.1 FINAL BLOCKS
The final blocks, numbered 620 thru 627 are necessary for turn-around space. Since the tape "coasts" one or two blocks past the last block addressed, and since the tape usually starts in the forward direction to locate its position, several inches of identified space is needed after the last usable block.

9.4.2 END ZONE
This is used for identification when marking tapes, and is not normally used by the programmer.

9.4.3 BLANK ZONE
This is the end of the tape with no marks.

9.5 TRACKS
There are 10 tracks: one timing, one mark, three data, and these same tracks repeated (redundant).

9.5.1 TIMING
The timing tracks contain a synchronizing square wave which is used to clock the rest of the system.

9.5.2 MARK
The mark tracks contain a unique set of codes which is used to identify the information on the corresponding data tracks.

9.5.3 DATA TRACKS
The three sets of data tracks contain the low, medium, and high order six bits of information (the two high order bits are not used).

9.5.4 BIT PLACEMENT
Figure 9.1 shows the arrangement of bits on the tape. It illustrates the track redundancy and the relationship of the mark track.
Figure 9.1
LINC Tape Format

Notes:
1. All dimensions approximate
2. Based on 256 words per block
10.0 THEORY OF OPERATION

LINCTAPE, UNLIKE INDUSTRY COMPATIBLE TAPE, DOES NOT HAVE TO STOP SUDDENLY IN A PARTICULAR RECORD GAP. SINCE LINCTAPE HAS CODED MARKS ON THE TAPE ITSELF, IT IS ALWAYS POSSIBLE TO KNOW EXACTLY WHERE ON THE TAPE THE DATA IS BEING READ OR WRITTEN. THIS LEADS TO SEVERAL ADVANTAGES:

- THE DRIVE SYSTEM IS SIMPLE WITH VERY FEW MOVING PARTS;
- THE TAPE ACCELERATION CAN BE SLOWER, THUS HANDLING THE TAPE MORE GENTLY;
- THE COMPUTER DOES NOT NEED TO KEEP TRACK OF THE TAPE POSITION, SINCE THIS CAN BE FOUND ON THE TAPE ITSELF;
- DATA CAN BE WRITTEN OVER OLD DATA PRECISELY, WITHOUT DANGER OF UNDER- OR OVER-WRITING ADJACENT RECORDS;
- BLOCK SEARCHING CAN BE DONE IN EITHER DIRECTION.

THE LINCTAPE SYSTEM CONSISTS OF MAGNETIC TAPE WITH FIVE EFFECTIVE TRACKS. FIGURE 10.1 SHOWS THE BASIC BLOCK DIAGRAM, AND FIGURE 9.1 ILLUSTRATES THE TAPE PATTERNS.

THE TIMING TRACK IS USED TO SYNCHRONIZE THE INFORMATION FROM ALL OTHER TRACKS. IT IS A SIMPLE SQUARE WAVE, 90 DEGREES OUT OF PHASE FROM THE OTHER TRACKS. THIS ALLOWS THE EDGES OF THE TIMING SIGNAL TO STROBE ALL OTHER DATA.

THE MARK TRACK IS THE KEY TO THE LINCTAPE SYSTEM. IT IS A SERIES OF UNIQUE, SIX BIT CODES WHICH IDENTIFY THE INFORMATION IN THE CORRESPONDING DATA TRACKS. THESE CODES SPECIFY WHEN THE DATA, CHECKSUM, OR BLOCK NUMBERS ARE AVAILABLE IN THE DATA REGISTERS.

THE THREE DATA TRACKS CONTAIN 6 BITS OF DATA PER TRACK FOR A TOTAL OF 18 BITS. TWO BITS ARE UNUSED. THESE TRACKS ALSO CONTAIN THE CHECKSUM (FOLLOWING THE LAST DATA WORD), AND THE FORWARD AND REVERSE BLOCK NUMBERS, AT THE BEGINNING AND END OF EACH BLOCK, RESPECTIVELY.


DURING WRITING, THE NEXT DATA WORD TO BE WRITTEN IS JAMMED INTO THE SHIFT REGISTERS WHEN THE "DATA READY" COMES TRUE, AND THE WORD IS SHIFTED, BIT BY BIT, INTO THE DATA WRITERS. SPECIAL "FINAL" MARK CODES ARE USED TO DIFFERENTIATE READING AND WRITING TIMING. SEE FIGURE 10.2.

Figure 10.1 Basic LINC Tape Block Diagram
NOTE: THIS TIMING DIAGRAM REPRESENTS POSITIVE TRUE LINES. THE LINCTAPE I/O LINES ARE GROUND TRUE.

FIGURE 10.2 TIMING FOR FORWARD MOTION.
The five head signals are derived from ten actual tracks on the tape. All data is recorded redundantly; i.e., there are two clock channels, two mark channels, and three pairs of data channels. Identical information is recorded on each corresponding track. See Figure 9.1. Within the head, the corresponding head pairs are wired in series. If a dropout should occur in a given track due to dirt, poor oxide, etc., the signal from the other track is sufficiently large to insure proper reading of the data. The clock and mark channels are at the extreme edges of the tape to minimize skew problems. The data tracks are positioned to maintain maximum separation between corresponding channels. It is the fully redundant nature of the recording process that is responsible for the high reliability of the linctape system.

A tape must be pre-marked before it can be used. This pre-marking writes the necessary data into the timing and mark tracks, and writes the block numbers. In operation, this pre-marked data cannot be changed. Since the tape is marked and each block is identified, it is unnecessary to know where the tape is to locate a specific block. The program (utilities) simply reads where it is and takes appropriate action.

A more detailed block diagram of the linctape system is shown in Figures 10.3 and 10.4.
Figure 10.3
LINC Tape Block Diagram: Data Registers
Figure 10.4
LINC Tape Block Diagram: Control, Mark, & Timing
11.0 SYSTEM COMPONENTS

THIS SECTION DESCRIBES THE MAJOR COMPONENTS OF THE LINCTAPE SYSTEM AND ILLUSTRATES THEIR LOCATION AND INTER-RELATION. SEE FIGURE 11.1 FOR THE PHYSICAL LAYOUT.

11.1 MASTER CONTROL BOARD

11.2 DRIVE CONTROL BOARD
THE DRIVE CONTROL BOARD IS LOCATED DIRECTLY BEHIND THE FRONT PANEL. IT CONTAINS MUCH OF THE MOTOR CONTROL LOGIC, THE HEAD SWITCHING DIODES, AND THE FRONT PANEL SWITCH LOGIC. THREE CONNECTORS AT THE TOP OF THE BOARD CONNECT TO 115 VAC, TO LOGIC SIGNALS FROM THE MASTER CONTROL BOARD, AND HEAD SIGNALS RESPECTIVELY. THIS BOARD IS IDENTICAL (EXCEPT FOR JUMPER WIRES) IN THE MASTER AND ALL SLAVE UNITS. THE SCHEMATICS FOR THIS BOARD ARE IN COI DRAWING NUMBER B-10164-01.

11.3 READER/WRITER BOARD

11.4 POWER SUPPLY

11.5 INTERCONNECTION
THE INTERCONNECTION CABLING BETWEEN THE BOARDS AND CONNECTORS IS SHOWN IN FIGURE 11.2.
12.0 POWER SUPPLY

THE POWER SUPPLY FURNISHES D.C. POWER FOR THE COMPLETE LINCTAPE SYSTEM. IT IS SUFFICIENT TO SUPPLY ANY NUMBER OF SLAVE UNITS.

<table>
<thead>
<tr>
<th>INPUT</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>115 V, 50/60 Hz</td>
<td>+5 V @ 2.5 AMPS</td>
</tr>
<tr>
<td></td>
<td>-5 V @ 0.15 AMPS</td>
</tr>
<tr>
<td></td>
<td>+15 V @ 2.0 AMPS</td>
</tr>
</tbody>
</table>

IN ADDITION, THERE IS A LOGIC LEVEL OUTPUT WHICH INDICATES THE LINE STATUS AND IS USED TO PROTECT THE TAPE WHEN POWERING UP OR DOWN, OR WITH POWER FAILURE.

THE +15 VOLT SUPPLY IS DELAYED UPON POWER UP UNTIL THE +5 VOLT SOURCE HAS STABILIZED AND THE LOGIC LEVEL HAS RESET THE CRITICAL REGISTERS TO A SAFE STATE.

WHEN POWERING DOWN, (OR UPON POWER FAILURE) THE +15 VOLT SUPPLY IS CROWBARRED AND THE LOGIC LEVEL IS PULLED DOWN BEFORE THE +5 VOLT SUPPLY CAN FALL. THIS PREVENTS ANY ACCIDENTAL WRITING ON TAPE. THE -5 VOLT SUPPLY IS NOT CRITICAL, AND IS NOT DELIBERATELY SEQUENCED.

FIGURE 12.1 SHOWS THE APPROXIMATE POWER ON AND OFF SEQUENCING.
POWER ON SEQUENCE

POWER OFF SEQUENCE

FIGURE 12.1
13.CII MAINTENANCE

ALTHOUGH LINE TAPE IS RELATIVELY MAINTENANCE-FREE, THERE ARE A FEW MINOR MAINTENANCE STEPS WHICH WILL HELP ASSURE LONG, TROUBLE-FREE SERVICE:

1. PERIODICALLY, CLEAN THE HEADS AND TAPE GUIDES. USE A SOFT, CLEAN CLOTH (JOHNSON STERI-PAD GAUZE PADS ARE GOOD) AND AN APPROVED, UNCONTAMINATED HEAD CLEANING SOLVENT (SUCH AS AMPEX HEAD CLEANER, PART NUMBER 7010110). THIS SHOULD BE DONE FROM ONCE A DAY TO ONCE A WEEK, DEPENDING ON USE AND THE ENVIRONMENTAL CONDITIONS. DO NOT LET THE SOLVENT CONTACT THE TAPE. BE SURE THE HEAD AND GUIDES ARE DRY BEFORE MOUNTING THE TAPE. DO NOT SCRATCH THE HEADS OR THE GUIDES.

2. THE TAPE THEMSELVES OCCASIONALLY GET DIRTY. THIS IS THE PROBABLE CAUSE OF A TAPE "HUNTING" FOR A BLOCK. ONE CAN BE CLEANED BY PASSING IT BETWEEN TWO CLEAN GAUZE PADS. MOUNT THE TAPE IN THE USUAL MANNER, PLACE ONE PAD ON THE HEAD AND HOLD THE OTHER ON THE TAPE, AND WIND THE TAPE SO THE WHOLE LENGTH PASSES BETWEEN THE PADS. BE CAREFUL NOT TO DAMAGE THE EDGES OF THE TAPE.

CAUTION: NEVER USE STICKY TAPE, SUCH AS "SCOTCH" TAPE OR ADHESIVE LABELS ON THE TAPE ITSELF. IN TIME, THE GUMMY MATERIAL TENDS TO DISTRIBUTE ITSELF OVER THE TAPE AND CAUSE DROPOUTS. PUT LABELS ON THE REEL, NOT THE TAPE. ALSO, KEEP TAPES AWAY FROM STRONG MAGNETIC FIELDS, SUCH AS NEAR TRANSFORMERS, MOTORS, OR FLUORESCENT LIGHT BALLASTS.

WHEN DIAGNOSING TROUBLES, THE FOLLOWING CHECKS ARE HELPFUL:

1. DO THE WRITE PROTECT LIGHTS WORK? IF SO, THE +15 VOLT IS PROBABLY OK.
3. ARE THE SHAFTS FREE TO TURN WITHOUT BEING SLOPPY? ARE ALL FOUR OF THEM ABOUT THE SAME? DO THE REELS SNAP ONTO THE HUBS PROPERLY? ARE THEY SO LOOSE THAT THE REELS SQUEAK WHEN MOTION STARTS?
4. ARE THE HEADS, GUIDES, AND TAPE CLEAN? IF THESE ARE DIRTY, THEY WILL CAUSE DROPOUTS.
5. IS THE TROUBLE INTERMITTENT? DOES IT COME AND GO WHEN WIRES ARE MOVED? IS IT ASSOCIATED WITH ONE DRIVE? ONE TAPE?
6. DOES THE CPU WORK? WITH OTHER PERIPHERALS?
7. DOES THE CPU TRANSFER DATA IN AND OUT OF THE REGISTERS PROPERLY?
8. DOES THE CPU START AND STOP TAPE? CAN THE RELAYS BE HEARD?
9. DOES THE CPU SELECT DRIVES? CAN THE RELAYS BE HEARD?
10. DOES THE CPU FIND BLOCK MARKS? DATA MARKS? CHECK MARKS?
Based on the above observations, the following faults might occur:

1. Power off or system not plugged in.
2. Fuse blown: There are four fuses in the power supply and one in each of the drive control boards (master and slaves).
3. Loose connections: Check the connectors on the rear panel, and all connectors inside the units. Are all terminals tight and clean? Sometimes terminals can be squeezed slightly to make them tighter. Edge terminal fingers can be cleaned with an eraser.
4. Loose IC's: Remove the IC cover and press all IC's firmly in place. Be sure none are missing.
5. Power fail crowbar hung up: Turn off power for a few seconds and try again.
6. Mechanical failure: Check to be sure the hubs and sprockets are firmly fastened to the shafts. Inspect the belt for wear. Be sure they are mechanically free. The hub spring tension can be changed by moving or replacing the rubber band under the spring.

Some important timing points that can be observed while the tape is moving forward are:

- TREA+ 40 US square wave
- All other reader signals are rectangular waves with transitions 20 or 40 US apart
- ACIP+ 130 MS pulse when starting motors or changing their direction
- BMRK+, GMRK+ 40 US pulse every 63 MS
- DMRK+ 255 40 US pulses each block, 240 US apart
- FMTN+ must be on for marks to decode, but the reader signals can be monitored by manually moving tape.
APPENDIX A. BOOTSTRAP

LINCTAPE NOVA BOOTSTRAP ROUTINE 9/27/71

There are three methods of bootstrapping, depending upon the hardware configuration:

1. Supernova Program Loader,
2. NOVA 800/1200 Series Autoprogram Option, or
3. NOVA or 800/1200 Series Manual Bootstrap.

The Linctape Bootstrap can be used with any of these to load in the Linctape utilities and load and execute the executive system routine.

This executive system routine will depend on the type of tape being used. The keyboard executive routine is one example of an executive system routine.

<<<<<< PROCEDURES >>>>>

The procedure for bootstrapping with the Supernova program loader or the NOVA 800/1200 series autoprogram option is:

1. Set the data switches to the Linctape device number (usually 74),
2. Put a Linctape with the NOVA bootstrap routine on drive 0, with the marker to the right of the head. Leave in tension mode,
3. Press reset,
4. Press program load. The Linctape will move and stop, and the teletype will respond with the appropriate executive system response.

The procedure for bootstrapping when using the manual bootstrap is:

1. Be sure the manual bootstrap is in core. It is listed below,
2. Set the data switches to X7770, where X7777 is the highest core location,
3. Put a Linctape with the NOVA loader on drive 0, with the marker to the right of the head. Leave in tension mode,
4. Press reset,
5. Press start. The Linctape will move and stop, and the teletype will respond with the appropriate executive system response.
; TO LOAD THE MANUAL BOOTSTRAP INTO CORE:
; (1) SET THE DATA SWITCHES TO X7755 (1ST LOC),
; (2) PRESS EXAMINE,
; (3) SET THE DATA SWITCHES TO 177737 (1ST WORD),
; (4) PRESS DEPOSIT,
; (5) SET THE DATA SWITCHES TO THE NEXT WORD,
; (6) PRESS DEPOSIT NEXT,
; (7) REPEAT (5) AND (6) UNTIL ALL WORDS ARE LOADED.

; LOC WORD
; ----- -----
; X7755 177737 COUNT: -41 ; WORDS LOADED
; X7756 126420 GET:  SUBZ 1,1 ; CLEAR AC1 & SET C
; X7757 0636YY SKPDN LINC ; BYTE READY?
; X7760 000777 JMP  -1 ; NO: WAIT
; X7761 0603YY DIAS 0,LINC ; YES: GET BYTE
; X7762 107363 ADDCS 0,1,SNR ; MERGE: ANOTHER?
; X7763 000774 JMP  -4 ; YES: GET IT
; X7764 125305 MOVS 1,1,SNR ; NO: SWAP BACK: ZERO?
; X7765 000771 JMP  GET ; YES: TRY AGAIN
; X7766 045041 STA 1,41,2 ; STORE IT
; X7767 001400 JMP  0,3 ; RETURN WITH WORD
; X7768 0605YY 177737 ; START LINCTAPE
; X7769 030764 LDA 2,COUNT ; SET WORD COUNTER
; X7770 004764 JSR  GET ; GET A WORD
; X7771 151404 INC 2,2,SZR ; INCR & TEST
; X7772 000776 JMP  -2 ; NOT THRU
; X7773 000002 JMP  MANLU ; GO TO MAN PRELOAD
; X7774 000000 ; RESERVE FOR BINARY LOADER ADDRESS
; X7775 000000 ; RESERVE FOR BINARY LOADER JUMP

WHERE:
X7777 = HIGHEST CORE LOCATION

; CORE X CORE X
; 4 K 0 20 K 4
; 8 K 1 24 K 5
; 12 K 2 28 K 6
; 16 K 3 32 K 7

YY = LINCTAPE DEVICE NUMBER (USUALLY 74)
<<<<< PROGRAM SEQUENCE >>>>

THERE ARE FOUR SECTIONS TO THE COMPLETE BOOTSTRAP ROUTINE, NOT INCLUDING THE HARDWARE ROUTINES. THESE FOUR SECTIONS RESIDE IN THE FIRST THREE BLOCKS OF LINCTAPE:

1. PRELOADER
2. UTILITY LOADER
3. LINCTAPE UTILITIES
4. EXECUTIVE SYSTEM ROUTINE.

REFER TO THE "HOW TO USE THE NOVA COMPUTERS" MANUAL, SECTIONS 2.3 AND 3.3 FOR DETAILS ON THE HARDWARE LOADERS.


THE SUPERNOVA OR THE MANUAL BOOTSTRAP SKIPS LEADING ZEROES, AND LOADS THE PRELOADER PROGRAM FROM LINCTAPE BLOCK -10 INTO CORE AT LOCATIONS 0 THRU 40. THE SUPERNOVA THEN JUMPS TO LOC 40, WHILE THE MANUAL BOOTSTRAP JUMPS TO LOC 2 OF THE PRELOADER.

THE PRELOADER SETS THE DEVICE NUMBER AND IN TURN LOADS THE UTILITY LOADER FROM BLOCK -7 INTO CORE LOCATIONS 142 THRU 216, AND EXECUTES AT LOC 216.

THE NOVA 800/1200 AUTOLOADER LOADS AND EXECUTES ITS OWN BOOTSTRAP FROM HARDWARE INTO LOCATIONS 0 THRU 37. IT THEN LOADS THE PRELOADER (WHICH IS IGNORED) AND THE UTILITY LOADER INTO CORE LOCATIONS 100 THRU 216, AND EXECUTES THE UTILITY LOADER AT LOCATION 216.

THE UTILITY LOADER IS THUS LOADED AND EXECUTED AT 216 BY ANY OF THE BOOTSTRAP CONFIGURATIONS. IT SIZES CORE, AND LOADS THE UTILITIES JUST BELOW THE BINARY TAPE LOADER AT X7600.

THE LINC UTILITIES THEN REST AT X7400 THRU X7577. THE PROGRAM THEN JUMPS TO "START" IN THE UTILITIES, READS THE EXECUTIVE SYSTEM FROM BLOCK -6 INTO CORE LOCATIONS X7000 THRU X7377, AND FINALLY STOPS THE TAPE AND JUMPS TO LOCATION X7377 WITH AC3 = ADDRESS OF "CLINC" (X7400) OF THE UTILITIES.

SOME LOCATIONS NEAR THE BOTTOM OF CORE ARE WIPED OUT BY THE BOOTSTRAP PROCESS. THEY ARE:

SUPERNOVA MANUAL AUTOLOADER
0- 40 0- 40 0-37
142-216 142-216 100-216
377 377
; NOW THE PRELOADER PROGRAM

000010   ,RDX  8
000012   ,LOC  2
000014   @21775  MANLD: LDA  0,-3,3 ; MANUAL ENTRY LOC: GET D.N.
000016   000005   JMP  .+2
000018   06477  PRELOD: READS 0 ; GET DEVICE NUMBER
00001A   024026   LDA  1,C77
00001C   125400   AND  1,0 ; MASK 6 BITS
00001E   024030   LDA  1,GET+1
000020   107900   ADD  0,1
000022   040330   STA  1,GET+1 ; SET SKIP COMMAND
000024   024032   LDA  1,GET+3
000026   107000   ADD  0,1
000028   040332   STA  1,GET+3 ; SET INPUT COMMAND
00002A   125440   SUBO  1,1 ; CLEAR AC1 & RESET CARRY
00002C   000040   JMP  .-2 ; YES: IGNORE IT
00002E   101065   MOVCL 0,0,SNR ; ZERO BYTE?
000030   000016   JMP  .-3 ; NO: GET NEXT FULL WORD
000032   040327   JSR  GET ; NO: GET NEXT FULL WORD
000034   046025   STA  1,@PTR1 ; STORE THE WORD
000036   010142   ISZ  142 ; LAST WORD?
000038   000021   JMP  .-4 ; NO: GET ANOTHER WORD
00003A   00141   PTR1:  141 ; POINTER TO STORE & JUMP
00003C   000077   C77:  77 ; GET A WORD SUBR. GET A BYTE SUBR
00003E   125420   GET:  SUBZ  1,1 ; CLEAR AC1 & SET CARRY
000040   063600   SKPDN  0 ; BUFFER READY?
000042   000030   JMP  .-1 ; NO: WAIT
000044   265500   DIAS  0,0 ; YES: INPUT THE BYTE
000046   107363   ADDCS 0,1,SNC ; SWAP BYTES. NEED ANOTHER?
000048   000030   JMP  .-4 ; YES: GO GET IT
00004A   125300   MOVSL 1,1 ; NO: SWAP BACK
00004C   001400   JMP  0,3 ; RETURN WITH WORD/BYTE
00004E   177777   -1
00004F   000004   FORTY: JMP  PRELOD ; SUPERNOVA ENTRY LOCATION
; Now the utility loader

RDX 8
.LOC 77

SYNC: 377 ; Sync byte
COUNT: SYNC-END ; Word counter
.LOC 142
-END=1

; Set device number

GOGO: LDA 0,GET+1 ; Get I/O word
LDA 1,K77
AND 1,0 ; Get device number
LDA 1,GETW
ADD 0,1
STA 1,GETW+2 ; Set skip command
LDA 1,GETW+2
ADD 0,1
STA 1,GETW+2 ; Set input command

; Size core

LDA 0,ONEK ; Memory size increment
LDA 3,HIGH ; Highest open memory
SUB 0,3 ; Move down to next core
STA 3,1,3 ; Trial storage
LDA 2,1,3 ; Get it again
SUB# 2,3,SR ; Test for match
JMP .-4 ; No match: try again

; Skip zero bytes

JSR GETW ; AC2 = highest open core
MOVZ 0,0,SNR ; Ignore leading zeroes
.JMP .-2

; Load the utilities

STA 0,ONEK ; Save count "LTSIZ"
ADD 0,2 ; Point to load addr
STA 2,K77 ; Save pointer
JSR GETW ; Get a word
STA 0,0,2 ; Store it
INC 2,2 ; Incr pointer
ISZ ONEK ; Incr counter
.JMP .-4 ; If not done, go back

; Set up to load keyboard exec

LDA 2,K77 ; If done, get 1st util loc
STA 2,LOC-CLINC,2 ; & put in LOC
LDA 1,SIZE-CLINC,2 ; Get words/block
SUB 1,2 ; Set 1st core LOC
SUBZL 1,1 ; Set no of blocks to 1
LDA 0,BLOCK ; Set next blk no
LDA 3,K77
.JMP START-CLINC,3 ; Goto start

; Get word subroutine

GETW: SKPDN 0 ; Buffer ready?
.JMP .-1 ; No: wait
DIAS 0,0 ; Read word into AC0
.JMP 0,3 ; Return with full word

; Block:

.BLOCK: 6

K77: 77
ONEK: 1000
HIGH: 77600+1000 ; Save binary loader
.END: JMP GOGO ; Enter bootstrap here
APPENDIX B. LINCTAPE UTILITIES

These utilities are shown assembled for a 4K machine. However, they are position independent, and therefore will be identical for any size machine.

; Now the Linctape Utilities -------
;
; Block 7 must contain zeros between the last word of the utility loader and "LTSIZ".

000010
007377
07377 177600 LTSIZ: -start ; Linctape utility size

; Entries to Linctape Utilities
; WLINC: Write & check
; RLINC: Read & check
; CLINC: Check only

*** User must select drive before call with "DOB -J74"

- All calls made by 'JSR' to one of the above,
  with AC0= first block #
  AC1= number of blocks to be processed
  AC2= first core address
  AC1 may =0. this is the "find" function.
  AC2 if negative, represents the one's compl of the real address, and causes
  the block search to start in reverse.

- Normal returns are indicated by AC1=0
- AC2= next core address
- AC0= next block #

- Abnormal returns have the error code in AC1:
  AC1=1 checksum error. AC0= bad block #
  AC1=2 block size error. AC0= bad block.
  AC2= excess of marks in block.
  AC3= expected #.
  AC1=4 illegal block. AC0= target block.
  AC2= next core addr.
  AC3= highest legal block.
  AC1=8 drive status error. AC3= drive status,
    bit 1= protected, bit 0= not ready.

07400 054431 CLINC: STA 3,SAC3
07401 152400 SUB 2,2 ; Address doesn't matter
07402 004917 JMP CHKZ
07403 054426 RLINC: STA 3,SAC3
07404 034430 LDA 3,D2R ; Set read ktne. To store data
07405 004915 JMP READZ
07406 054423 WLINC: STA 3,SAC3
07407 034423 LDA 3,D1W
07410 054510 STA 3,D1XX ; Set up for write
07411 044501 STA 1,D2XX ; Save params
07412 054416 STA 2,SAC2
07413 004423 JSR DO ; Find & write blocks
07414 024476 RAW: LDA 1,D2XX ;RESTORE PARAMS
07415 122400 SUB 1,0
07416 034412 LDA 2,SAC2
07417 151113 MOVl# 2,2,SNC ;MAKE ADDR. NEG FOR REVERSE
07420 150000 COM 2,2
07421 034473 CHKZ: LDA 3,D2C ; NO STORE ON CHECK

07422 054470 READZ: STA 3,D2XX
07423 034410 LDA 3,D1RC
07424 054474 STA 3,D1XX ; SET UP FOR READ OR CHECK
07425 004411 JRSP DO ; FIND & READ BLOCKS
07426 060274 EXIT: NIOC LINC ; STOP DRIVE
07427 002402 JMP @SAC3 ; RETURN TO CALLER
07430 000000 SAC2: 0
07431 000000 SAC3: 0

07432 024700 D1W: LDA 0,0,2 ; DATA FOR WRITE SWITCH
07433 004750 D1RC: JMP READ-D1XX,1 ; FOR READ SWITCH
07434 132512 D2R: SUBL# 1,2,SZC ;DO THIS FOR READ, NOT CHECK
07435 000000 RETU: 0

07436 054777 D0: STA 3,RETU
07437 075474 DIB 3,LINC ; CHECK DRIVE READY
07440 175112 MOVl# 3,3,SZC
07441 004446 JMP E4
07442 151113 MOVl# 2,2,SNC ; LOOK AT ADDR.
07443 004470 JMP FINDF ; IF POS, START FORWARD
07444 150000 COM 2,2 ; IF NEG, START REVERSE

07445 175400 FINDR: SUB 3,3 ; ENTER HERE FOR REV.
07446 162000 ADC 3,0 ;POINT TO TARGET-1
07447 060374 NIOP LINC ;START REV
07450 004467 JSR GETBLOCK
07451 101401 FINDN: INC 0,0,SKP
07452 000776 JMP -2 ; KEEP GOING IF ABOVE OR ON
07453 060174 FINDF: NIOS LINC ; ENTER HERE FOR FWD.
07454 004463 JSR GETBLOCK
07455 000777 JMP -1 ; KEEP GOING IF BELOW
07456 175224 MOVZR 3,3,SZC ; FOUND TARGET IF =0
07457 000766 JMP FINDR
07460 125005 FOUND: MOV 1,1,SNR ;LAST BLOCK?
07461 002754 JMP @RETU ; YES, EXIT FROM "DO"
07462 160000 ADC 3,1 ;AC3=0. DECR BLOCK COUNTER
07463 040474 STA 0,TEMP1
07464 044474 STA 1,TEMP2
07465 024476 LDA 1,SIZE
07466 147000 ADD 2,1 ;POINT TO END OF BLOCK.
07467 000431 JMP D1XX
07470 063474 READ: SKPDN LINC ; WAIT FOR DATA
07471 000777 JMP -1
07472 063474 SKPBH LINC
07473 000416 JMP RDAT
07474 063474 RCHK: DIA 0,LINC ; INPUT CHECKSUM
07475 114405 SUB 0,3,SNR
07476 000434 JMP SCHK ; SHOULD = ACCUM, CHKSM
07477 024465 E1: LDA 1,C1 ; CHECKSUM ERROR
07478 000403 JMP *+3
07479 034463 E2: LDA 3,SIZE
07480 024463 LDA 1,C2
07481 024454 LDA 0,TEMP1
07482 000722 JMP EXIT
07483 024461 E3: LDA 1,C4 ; ILLEGAL BLOCK NUMBER
07484 000720 JMP EXIT
07485 024460 E4: LDA 1,C8 ; DRIVE STATUS ERROR
07501 024463 JMP EXIT
07502 024463 JMP EXIT
07503 024454 JMP EXIT
07504 000722 JMP EXIT
07505 024461 E3: LDA 1,C4 ; ILLEGAL BLOCK NUMBER
07506 000720 JMP EXIT
07507 024460 E4: LDA 1,C8 ; DRIVE STATUS ERROR
07511 060474 RDAT: DIA 0,LINC ; INPUT DATA WORD
07512 135212 D2XX: SUBL# 1,2,SZC ; "JMP *+2" FOR CHECK
07513 041000 STA 0,0,2
07514 000402 D2C: JMP *+2
07515 060474 WDAT: DIA 0,LINC ; WRITE DATA
07516 111000 BLOOP: ADD 0,3 ; UPDATE CHKSM ACCUM.
07517 111000 INC 0,3 ; UPDATE POINTER
07518 021000 D1XX: LDA 0,0,2 ; FOR READ & CHECK, "JMP READ"
07519 060474 DOC 0,LINC ; WRITERS ON
07520 060474 SKPDN LINC
07521 060474 JMP -1 ; WAIT FOR DATA, CHECK MARK
07522 060474 SKPBH LINC
07523 000777 JMP E4 ; MUST BE READY & UNPROTECTED
07524 060474 SKPBH LINC
07525 000770 JMP WDAT ; DATA MARK
07526 075074 WCHK: DIA 3,LINC ; WRITE CHECKSUM
07527 075074 DIB 3,LINC ; INPUT DRIVE STATUS
07528 175004 MOV 3,3,SR
07529 000756 JMP E4 ; MUST BE READY & UNPROTECTED
07530 134414 SCHK: SUB# 1,2,SZC ; WAS BLOCK RIGHT SIZE?
07531 000746 JMP E2
07532 134414 SCHK: SUB# 1,2,SZC ; WAS BLOCK RIGHT SIZE?
07533 000746 JMP E2
07534 024423 NEXT: LDA 0,TEMP1 ; BLOCK FINISHED
07535 024423 LDA 1,TEMP2 ; RESTORE BLOCK COUNTER
07536 000713 JMP FINDDN ; NEXT BLOCK
07537 054420 GETBLOCK: STA 3, TEMP1
07540 034421 LDA 3, MLIM ; CHECK TARGET LIMITS
07541 162432 SUBZ# 3, 0, SNC ; OK IF BETWEEN MLIM & PLIM
07542 000405 JMP WAIT
07543 034417 LDA 3, PLIM
07544 162032 ADCZ# 3, 0, SNC
07545 000740 JMP E3 ; NO SUCH BLOCK
07546 074474 DIA 3, LINC ; CLEAR SYNC FLOP
07547 063474 WAIT: SKPBN LINC ; GET BLOCK #
07550 000777 JMP WAIT
07551 065774 SKPDZ LINC
07552 000774 JMP WAIT-1 ; WAS A CHECK MARK
07553 074474 DIA 3, LINC ; INPUT IT
07554 116543 SUBOL 0, 3, SNC ; SKIP IF BELOW BLOCK WANTED
07555 010402 ISZ TEMP1
07556 002401 JMP @TEMP1
07557 000000 TEMP1: 0
07560 000000 TEMP2: 0
07561 177770 MLIM: 177770 ; LOWEST BLOCK
07562 000620 PLIM: 620 ; HIGHEST BLOCK
07563 000400 SIZE: 400 ; BLOCK LENGTH
07564 000001 C1: 1
07565 000002 C2: 2
07566 000004 C4: 4
07567 000010 C8: 10 ;
WHAT FOLLOWS IS A BOOT, WHICH CAN BE USED TO LOAD ALL BUT ONE BLOCK OF MEMORY, FROM DRIVE 0. THIS PROCEDURE IS USED BY THE BOOT PROG.

JUMP TO "START" WITH:

- AC0 = FIRST BLOCK TO BE READ,
- AC1 = # BLOCKS TO READ,
- AC2 = LOAD ADDR.

A HALT @ "PAUSE" INDICATES ERROR.

IF NO ERROR, CONTROL IS PASSED TO THE LAST WORD LOADED, WITH:

- AC2 = ADDR OF LAST WURD LOADED +1,
- AC3 = ADDR OF "CLINC" IN UTILITIES.

```
07570 000000 LLOC: 0
07571 004612 JSR KLINC
07572 034776 LDA 3,LLUC
07573 125005 MOV 1,1,SNR
07574 001377 JMP -1,2
07575 062677 PAUSE: IORST ; STOP & SELECT DRIVE 0
07576 065077 HALT ; IF ERROR, SET AC'S & CONTINUE
07577 000772 START: JMP -.6 ; SYSTEM START BLOCK
.END
```
APPENDIX C. KEYBOARD EXECUTIVE

LINCTAPE KEYBOARD EXECUTIVE ROUTINE 5/71

THE LINC EXECUTIVE INTERPRETS TYPED COMMANDS, AND PERFORMS LINCTAPE OPERATIONS ACCORDINGLY. THE PROGRAM MANIPULATES INTEGRAL NUMBERS OF LINC BLOCKS (256 WORDS EACH) BETWEEN TAPE AND ANY CONTIGUOUS CORE LOCATIONS.

COMMAND STRUCTURE

COMMANDS ARE OF TWO MODES:
A, B, C, D, X GENERAL MODE
AND Y AUTOMATIC MODE.

SYMBOL DEFINITIONS:

A = ADDRESS, AN OCTAL DIGIT STRING, FOLLOWED BY A COMMA. THE LAST 5 DIGITS TYPED REPRESENT THE FIRST CORE ADDRESS. IF ONLY A COMMA IS TYPED, THE VALUE WILL BE INTERPRETED AS 0.

B = BLOCK, THE FIRST BLOCK NUMBER; SAME GENERAL DEFINITION AS 'A', EXCEPT THE STRING IS INTERPRETED AS A 16-BIT SIGNED NUMBER. (2'S COMPLEMENT CONVENTION)

C = COUNT, THE COUNT OF BLOCKS TO BE PROCESSED; SAME DEFINITION, BUT A 16-BIT UNSIGNED NUMBER.

D = DRIVE, THE DRIVE NUMBER; AN OCTAL DIGIT STRING NOT FOLLOWED BY A COMMA. ONLY THE LEAST SIGNIFICANT (LAST TYPED) 4 BITS ARE USED.

X = ONE OF THE FOLLOWING SINGLE CHARACTERS:
C = CHECK - FIND AND VERIFY BLOCK NUMBERS AND CHECKSUMS.
R = READ - FIND, CHECK AND TRANSFER INTO CORE.
W = WRITE - FIND, WRITE FROM CORE, AND CHECK (TWO PASSES).
THE INTERPRETER CATCHES INVALID CHARACTERS TYPED, AND CHECKS COMMAND FORMAT, BUT IT IS POSSIBLE TO "BUMP" THE SYSTEM BY LOADING OVER THE EXECUTIVE PROGRAM, FOR EXAMPLE. THE EXECUTIVE RESIDES AT X7000-X7377 OCTAL (WHERE X7777 IS THE HIGHEST CORE LOCATION), AND IT CALLS ON THE LINC UTILITIES AT X7400-X7577 OCTAL. IN OTHER WORDS, THE MAXIMUM NUMBER OF BLOCKS YOU CAN READ IS 16 BLOCKS IN A 4K MACHINE, OR ALL BUT THE HIGHEST 1000 LOCATIONS OF CORE. THERE IS NO RESTRICTION ON CHECKING.

THE DRIVE MUST BE IN TENSION MODE AND, IF WRITING, THE DRIVE PROTECT LIGHT MUST BE OFF. IF ON, THE TAPE WILL NOT BE WRITTEN, AND AN ERROR WILL BE INDICATED.

THE EXECUTIVE MAY BE RESTARTED AS FOLLOWS:
PRESS 'RESET'
SET DATA SWITCHES TO X7000 OCTAL
PRESS 'START'
AUTOMATIC MODE

IN THE AUTOMATIC MODE, TYPING A SINGLE CHARACTER WILL READ IN THE DESIRED PROGRAM. THIS MODE WILL ONLY READ FROM DRIVE 0. IT WILL NOT WRITE, NOR WILL IT ACCESS ANY OTHER DRIVE.

FOR INSTANCE, TYPING "A" CAN READ THE ASSEMBLER FROM THE TAPE ON DRIVE 0 INTO CORE. AS WITH THE GENERAL MODE, THE SYSTEMS RETURNS TO THE KEYBOARD EXEC AFTER READING, AND THE NORMAL START PROCEDURE FOR THAT PROGRAM MUST BE FOLLOWED TO EXECUTE IT.

THE NECESSARY INFORMATION FOR READING IS STORED IN A TABLE IN THE KEYBOARD EXEC. IT IS ASSUMED THAT THE PROPER PROGRAM WILL BE FOUND ON THE TAPE. IN GENERAL, IT IS ADVISABLE TO MAINTAIN THE TABLE IN THE KEYBOARD EXEC TO MATCH THE PROGRAMS ON THAT PARTICULAR TAPE. IN THIS WAY, THE SYSTEM, AFTER BOOTSTRAPPING, WILL FUNCTION PROPERLY.

WHEN PROGRAMS ARE ADDED TO OR DELETED FROM THE TAPE, THE KEYBOARD EXEC MUST BE UPDATED IF THE AUTOMATIC MODE IS TO HANDLE IT PROPERLY. THIS TABLE STARTS AT LOCATION YY212 (WHERE YY000 IS THE FIRST LOCATION IN THE EXEC). THE TABLE REQUIRES A GROUP OF FOUR PARAMETERS FOR EACH PROGRAM:

1. KEYBOARD LETTER
2. FIRST CORE LOCATION
3. FIRST BLOCK NUMBER
4. NUMBER OF BLOCKS

THE SEQUENCE OF THE GROUPS DOES NOT MATTER, BUT THE SEQUENCE OF THE PARAMETERS WITHIN THE GROUP DOES. THERE IS ROOM IN THE TABLE FOR 23 LETTERS.

THE PROCEDURE FOR CHANGING THE TABLE IS:

1. ENTER THE EXECUTIVE ROUTINE
2. LOAD THE DEBUG III ROUTINE
3. PUT THE TAPE WITH THE EXECUTIVE ROUTINE THAT IS TO BE UPDATED ON DRIVE 0
4. READ A FRESH COPY OF THE EXEC INTO WORKING CORE (THE EXEC IS ON BLOCK -6, OR 177772):
   1000, 177772, 1, 00 NEVER TRY TO UPDATE THE EXEC "IN PLACE".
5. ENTER THE DEBUG ROUTINE
6. LIST THE TABLE, FROM LOCATION 1212 UNTIL THE FIRST (ASCII LETTER) WORD IN A GROUP IS ZERO, INDICATING THE END OF THE TABLE (A LOCATION ENDING IN -2 OR -6.)
(7) ADU, CHANGE, OR DELETE GROUPS IN THE TABLE AS REQUIRED. 
BE SURE TO PUT THE NUMBERS IN THE CORRECT SEQUENCE, AND 
TO PUT THE "END OF TABLE" ZERO AFTER THE LAST SET. NOTF 
ALSO THAT THE ASCII IS 7, NOT 8 BIT CODE. A TYPICAL 
SEQUENCE IS:

```
001242 000113  ASCII LETTER K
001243 001000  1ST CORE LOCATION
001244 000030   1ST BLOCK NUMBER
001245 000014 NUMBER OF BLOCKS
001246 000000 END OF TABLE
```

THE TABLE CAN, OF COURSE, BE UPDATED BY OTHER 
MEANS, SUCH AS THE FRONT PANEL SWITCHES.

(8) WRITE THE UPDATED VERSION BACK ONTO LINCTAPE:
1000,177772,1,0w

(9) REBOOT, AND TEST THE NEW ROUTINE.

TO EXECUTE ANY LOADED PROGRAM, YOU MUST FOLLOW 
NORMAL INSTRUCTIONS FOR THAT PROGRAM. THE 
LINC EXEC DOES NOT START PROGRAMS.

ERRORS ARE INDICATED BY A RETURN TO THE 
EXECUTIVE WITH ? TYPED, THEN *

ERRORS CAN BE :
1. TAPE NOT IN TENSION.
2. TAPE PROTECTED AGAINST WRITE.
3. CHECKSUM ERROR; YOU HAVE A BAD BLOCK ON TAPE,
   OR YOU PRESSED PROTECT SWITCH WHILE WRITING. RETRY.
4. BAD TAPE, NEEDS REMARKING.
5. YOU TRIED TO FIND A BLOCK NOT ON THE TAPE.
   LIMITS ARE 177772 (-6) THROUGH 0, TO 617, OCTAL.
6. INVALID COMMAND.

SUCCESSFUL COMPLETION OF A COMMAND IS INDICATED BY 
A RETURN TO THE EXECUTIVE WITH * TYPED.

EXAMPLES

```
*100,2,1,0r
* READ, STARTING AT CORE ADDRESS 100,
   ONE BLOCK, STARTING AT BLOCK 2, DRIVE 0.

*0,77777777777,1,1w
* WRITE BLOCK -1 FROM CORE ADDR 0, DRIVE 1

*0,177777,1,1w
? ERROR. SAME COMMAND AS ABOVE.

*0,0,620,0c
* CHECK TAPE BLOCKS 0 THROUGH 617.

*0,620,0c
* SAME THING.

*A
* LOAD EXTENDED ASSEMBLER.
```
THIS ROUTINE IS LISTED AS ASSEMBLED AT LOCATION 7000.
EXCEPT FOR THE POINTERS, WHICH ARE PRESET AT BOOT TIME, IT IS
POSITION INDEPENDENT, AND WILL BE THE SAME FOR ANY SIZE CORE.

; KEYBOARD EXEC 5/71/JJM

; THESE ROUTINES ALLOW THE USER TO USE THE LINC
; TAPE UTILITIES FROM THE KEYBOARD.
; THIS PROGRAM ACCEPTS TWO TYPES OF COMMANDS
; TYPE 1 IS THE SINGLE LETTER TYPE WHICH IS USED
; PRIMARILY TO READ IN SYSTEM PROGRAMS FROM
; DRIVE 0 ONLY. THE PROGRAM CONTAINS A LOOKUP
; TABLE FOR THESE COMMANDS WHICH CAN BE EASILY
; UPDATED WITH DEBUGER.
; THE TYPE TWO COMMANDS ARE READ, WRITE AND
; CHECK. THESE COMMANDS REQUIRE 4 PARAMETERS
; AS EXPLAINED IN THE INSTRUCTION BOOK.

007000 .LOC 7000

07000 00504 EXEC: 0704 006 007000 004054
07001 102400 JMP ELIN ; OUTPUT CR "*
07002 040571 SUB 0,0 ; CLEAR AC0
07003 040571 STA 0,TEM1 ; CLEAR INPUT ARRAY
07004 040571 STA 0,TEM2
07005 040571 STA 0,TEM3
07006 020552 STA 0,TEM4
07007 040571 LDA 0,C4 ; SET RE-TRY COUNT
07008 040571 STA 0,ERCO
07009 04503 JSR INOC ; INPUT OCTAL NUMBER
07010 04462 STA 1,TEM1
07011 03552 LDA 1,C54 ; COMA??
07012 142415 SUB# 2,0,SNR
07013 00422 JMP COMA

07015 024543 LDA 1,C4 ; LOOK UP INPUT CHARACTER
07016 034573 LDA 3,TABL
07017 031400 EAGN: LDA 2,0,3
07018 151005 MOV 2,2,SNR ; IF END OF TABLE GRIPE
07019 004571 JMP NOGO
07020 142415 SUB# 2,0,SNR ; IF MATCH GO TO IT
07021 00403 JMP FNDI
07022 13700 ADD 1,3 ; UPDATE POINTER
07023 00772 JMP EAGN ; AND TRY AGAIN

07026 021401 FNDI: LDA 0,1,3 ; GET PARAMETERS AND SAVE THEM
07027 040544 STA 0,TEM1
07028 021402 LDA 0,2,3
07029 040543 STA 0,TEM2
07030 021403 LDA 0,3,3
07031 040542 STA 0,TEM3
07032 02536 LDA 0,CR
07033 00414 JMP COM1-1 ; GO TO IT
07036 004455 COMA: JSR INUC ; GET NEXT NUMBER
07037 044535 STA 1,TEM2 ; SAVE IT
07040 030524 LDA 2,C54 ; CHECK FOR COMA
07041 142414 SUB# 2,0,SR
07042 000436 JMP NOGO
07043 004450 JSR INUC ; GET NEXT NUMBER
07044 044531 STA 1,TEM3
07045 142404 SUB# 2,0,SR
07046 000432 JMP NOGO
07047 004444 JSR INUC
07050 044526 STA 1,TEM4
07051 034530 LDA 3,TABL1 ; LOOK UP CHARACTER
07052 031400 COM1: LDA 2,0,3
07053 175400 INC 3,3
07054 175400 INC 3,3
07055 151015 MOV# 2,2,SR
07056 000422 JMP NOGO ; IF END OF TABLE GRIPE
07057 142414 SUB# 2,0,SR
07060 000772 JMP COM1 ; IF NOT MATCH TRY AGAIN
07061 035777 LDA 3,-1,3 ; GET EXECUTION ADDRESS
07062 054515 STA 3,FDDR ; SAVE EXEC ADDRESS

07063 020511 TRYAGN: LDA 0,TEM2 ; STARTING BLOCK #
07064 024511 LDA 1,TEM3 ; NUMBER OF BLOCKS
07065 030511 LDA 2,TEM4 ; DRIVE #
07066 072074 DOB 2,74 ; SELECT DRIVE
07067 030504 LDA 2,TEM1 ; STARTING CORE LOCATION
07070 034507 LDA 3,FDDR ; GET EXECUTION ADDRESS
07071 000400 JSR 0,3 ; GO TO IT
07072 120101 MOV# 1,1,SR ; IF NO ERROR RETURN TO START
07073 000705 JMP EXEC
07074 125223 MOVZER 1,1,SNC ; IF CHECKSUM TRY AGAIN
07075 000403 JMP NOGO ; ELSE GIVE UP
07076 014502 DSZ ERCO
07077 000764 JMP TRYAGN ; TRY THREE TIMES

07100 020462 NOGO: LDA 0,C15
07101 004447 JSR OA0
07102 020466 LDA 0,CQUES
07103 004445 JSR OA0
07104 020456 ELIN: LDA 0,C15
07105 004443 JSR OA0
07106 020463 LDA 0,CSTAR
07107 004441 JSR OA0
07110 020453 LDA 0,C40
07111 004437 JSR OA0
07112 000667 JMP EXEC+1
; CONVERT AN ASCII OCTAL CHARACTER STRING TO A BINARY NUMBER IN AC1, AND A BREAK CHARACTER IN AC0

07113 054425 INOC: STA 3,RTRN ; SAVE RETURN ADDRESS
07114 050425 STA 2,TEM0 ; SAVE AC2
07115 102400 SUB 0,0
07116 040424 STA 0,OCTL ; CLEAR RESULT WORD
07117 004424 INOC1: JSR IA0 ; GET A CHARACTER
07120 030445 LDA 2,C60
07121 034445 LDA 3,C67
07122 162033 ADCZ# 3,0,SNC ; TEST FOR 60<=N<=67
07123 112032 ADCZ# 0,2,SZC ; NO MUST BE BREAK CHARACTER
07124 000411 JMP EINOC ; MAKE IT OCTAL
07125 142400 SUB 2,0
07126 024414 LDA 1,OCTL
07127 125120 MOVZL 1,1 ; OLD TIMES 8
07128 125120 MOVZL 1,1
07130 125120 MOVZL 1,1
07131 125120 MOVZL 1,1
07132 107000 ADD 0,1 ; NEW PLUS OLD
07133 044407 STA 1,OCTL
07134 000763 JMP INOC1 ; LOOP UNTIL BREAK CHARACTER
07135 030404 EINOC: LDA 2,TEM0 ; RESTORE AC2
07136 024404 LDA 1,OCTL ; RESULT TO AC1
07137 002401 JMP @RTRN

07140 000000 RTRN: 0 ; SAVE RETURN LOCATION
07141 000000 TEM0: 0 ; STORAGE FOR AC2
07142 000000 OCTL: 0 ; RESULT

; INPUT AND ECHO TELETYPER CHARACTER AND MASK PARITY

07143 063610 IA0: SKP DN TTI ; WAIT FOR INPUT CHARACTER
07144 000777 JMP -1
07145 060610 DIAC 0,TTI ; INPUT CHARACTER TO AC0
07146 030421 LDA 2,C177 ; MASK OFF PARITY BIT
07147 143400 AND 2,0

07150 063511 OA0: SKPB Z TTO ; WAIT FOR OUTPUT READY
07151 000777 JMP -1
07152 061111 DOAS 0,TTO ; OUTPUT AC0 AND START
07153 030407 LDA 2,C15
07154 142434 SUBZ# 2,0,SZR ; IF CR OUTPUT LF
07155 014000 JMP 0,3
07156 024403 LDA 0,C12
07157 000771 JMP @A0
; CONSTANTS

07160 000004 C4: 4
07161 000012 C12: 12
07162 000015 C15: 15
07163 000040 C40: 40
07164 000054 C54: 54
07165 000060 C60: 60
07166 000067 C67: 67
07167 000177 C177: 177
07170 000077 CQUES: *?
07171 000052 CSTAR: * *
07172 000122 CR: *R
07173 000000 TEM1: 0
07174 000000 TEM2: 0
07175 000000 TEM3: 0
07176 000000 TEM4: 0
07177 000000 FDDR: 0
07200 000000 ERCO: 0
; VALUES IN THIS TABLE WILL BE FILLED IN BY THE SIZE ROUTINE AT LOAD TIME
; ADDRESS OF TABLE
07201 000000 TABL: 0
07202 00122 LR: *R
07203 000000 LC: 0
07204 00103 "C
07205 000000 LW: 0
07206 00127 0
07207 000000 0
07210 000000 ; END OF TABLE

; AUTOMATIC MODE TABLE
; THIS TABLE IS USED TO LOOK UP THE STARTING CORE LOCATION, STARTING BLOCK # AND THE NUMBER OF BLOCKS FOR THE SINGLE LETTER COMMANDS.

; THE FORM IS:
; COMMAND LETTER
; STARTING CORE LOCATION
; STARTING BLOCK NUMBER
; NUMBER OF BLOCKS

; THUS:
; 07212 00104 "D
; 07213 =01000 1000
; 07214 000000 0
; 07215 00014 14
; 07216 000000 0
; WOULD LOAD LOCATIONS 1000 THRU 6177 INTO CORE FROM BLOCKS 0 THRU 13 ON THE TAPE ON DRIVE 0, WHEN THE LETTER "D" WAS TYPED.

; THERE IS ROOM IN THE TABLE FOR 21 (DECIMAL) COMMAND LETTERS.

; ADDRESS OF TABLE
07211 000000 TABL: 0
07212 000000 ; END OF TABLE
; THIS ROUTINE DETERMINES THE SIZE OF CORE AND
; SETS ALL OF THE POINTERS.

007352 .LOC EXEC+352
07352 024421 SIZE: LDA 1, C400
07353 132400 SUB 1, 2
07354 141000 MOV 2, 0
07355 024421 LDA 1, TABP
07356 133000 ADD 1, 2
07357 051377 STA 2, -1, 2
07358 024415 LDA 1, TAB2P
07359 133000 ADD 1, 2
07360 051377 STA 2, -1, 2
07361 054622 STA 3, LC
07362 024410 LDA 1, C3
07363 137000 ADD 1, 3
07364 054615 STA 3, LR
07365 137000 ADD 1, 3
07366 054617 STA 3, LW
07367 111000 MOV 0, 2
07368 010000 JMP 0, 2
07369 000400 C400: 400
07370 000003 C3: 3
07371 000100 TAB2P: TABL-TABL1
07372 000202 TABP: TABL1-EXEC+1
07373 000753 JMP SIZE

.END
APPENDIX D. LINCTAPE MAP

STANDARD LINCTAPES FOR USE WITH THE NOVA COMPUTERS CONTAIN THE FOLLOWING PROGRAMS:

<table>
<thead>
<tr>
<th>BLOCK NUMBER</th>
<th>PROGRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>-10</td>
<td>PRELOADER (BYTE FORM)</td>
</tr>
<tr>
<td>-7</td>
<td>UTILITY LOADER (BYTE FORM)</td>
</tr>
<tr>
<td></td>
<td>LINCTAPE UTILITIES</td>
</tr>
<tr>
<td>-6</td>
<td>KEYBOARD EXECUTIVE</td>
</tr>
<tr>
<td>-5 THRU -1</td>
<td>ZEROS</td>
</tr>
</tbody>
</table>