CO 6ØØ NP
LINC TAPE SYSTEM
FOR NOVA COMPUTERS

REFERENCE MANUAL

COMPUTER OPERATIONS, INC.
1Ø774 TUCKER STREET
BELTSVILLE, MARYLAND 2Ø7Ø5

NOVEMBER 24, 1971
DOC. NO. 112

TABLE OF CONTENTS

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-1Ø23Ø-Ø1	LINCTAPE MASTER CONTROLLER (D-1Ø144-Ø1 FOR S/N BELOW 1Ø16)
B-1Ø164-Ø1	LINCTAPE DRIVE CONTROL BOARD
D-1Ø198-Ø1	LINCTAPE READER/WRITER BOARD (D-1Ø232-Ø1 FOR S/N BELOW 1Ø16)
D-1Ø244-Ø1	LINCTAPE POWER SUPPLY

CO 600

LINC TAPE SYSTEM

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 8ØØ SERIES, OR NOVA 12ØØ 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 THRUOUT THIS MANUAL, THAT A TELETYPE * OR EQUIVALENT I/O DEVICE IS AVAILABLE. THE I/O CONNECTOR IS STANDARD ON THE 121Ø, 122Ø, AND 82Ø NOVA COMPUTERS.

* TELETYPE IS A TRADEMARK OF THE TELETYPE CORPORATION.

1.Ø GENERAL DESCRIPTION

THE CO-6ØØ-NP LINCTAPE SYSTEM CONSISTS OF TWO (OR MORE) DRIVES AND A COMMON ELECTRONICS SYSTEM. EACH DRIVE HANDLES ONE REEL OF LINCTAPE.

EACH TAPE IS DIVIDED INTO SECTORS OR BLOCKS. THERE ARE 4ØØ (62Ø OCTAL) SUCH BLOCKS PER TAPE, NUMBERED Ø 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 3ØØØ THRU 3777 (OCTAL). IT CAN BE WRITTEN OUT ONTO LINCTAPE AS FOLLOWS:

```
        SUB  Ø,Ø
        DOB  Ø,LINC    SELECT DRIVE NO Ø
        LDA  Ø,BLKNO   LOAD ACØ 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: 1ØØ
NBLKS: 2
FCORE: 3ØØØ
WLINC: X74Ø6
RLINC: X74Ø3
```

THIS ROUTINE WILL WRITE 2 BLOCKS (1ØØØ OCTAL WORDS), STARTING AT CORE LOCATION 3ØØØ, ONTO THE LINCTAPE THAT IS ON DRIVE Ø, STARTING WITH BLOCK NUMBER 1ØØ.

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 LINC TAPE OVER INDUSTRY COMPATIBLE TAPE SYSTEMS IS THE ABILITY TO OVER-WRITE 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.Ø SPECIFICATIONS

2.1 SYSTEM

DUAL TAPE DRIVE
EXPANDABLE TO 16 DRIVES (8 DUAL)
6Ø 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
13Ø 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. 481-3/4-15Ø-R42 (UNMARKED)
15Ø FEET LONG
3/4 INCH WIDE
1Ø TRACKS, 35 MIL, FULLY REDUNDANT
SANDWICH TAPE FOR LONG TAPE AND HEAD LIFE
3-13/16 INCH DIAMETER REEL
4ØØ BPI PHASE RECORDING
16 BIT WORD
1Ø2,4ØØ WORDS PER REEL (2Ø4,8ØØ BYTES)
42ØØ WORDS/SECOND TRANSFER RATE (84ØØ BYTES/SECOND)
4ØØ DATA BLOCKS
256 WORDS/BLOCK

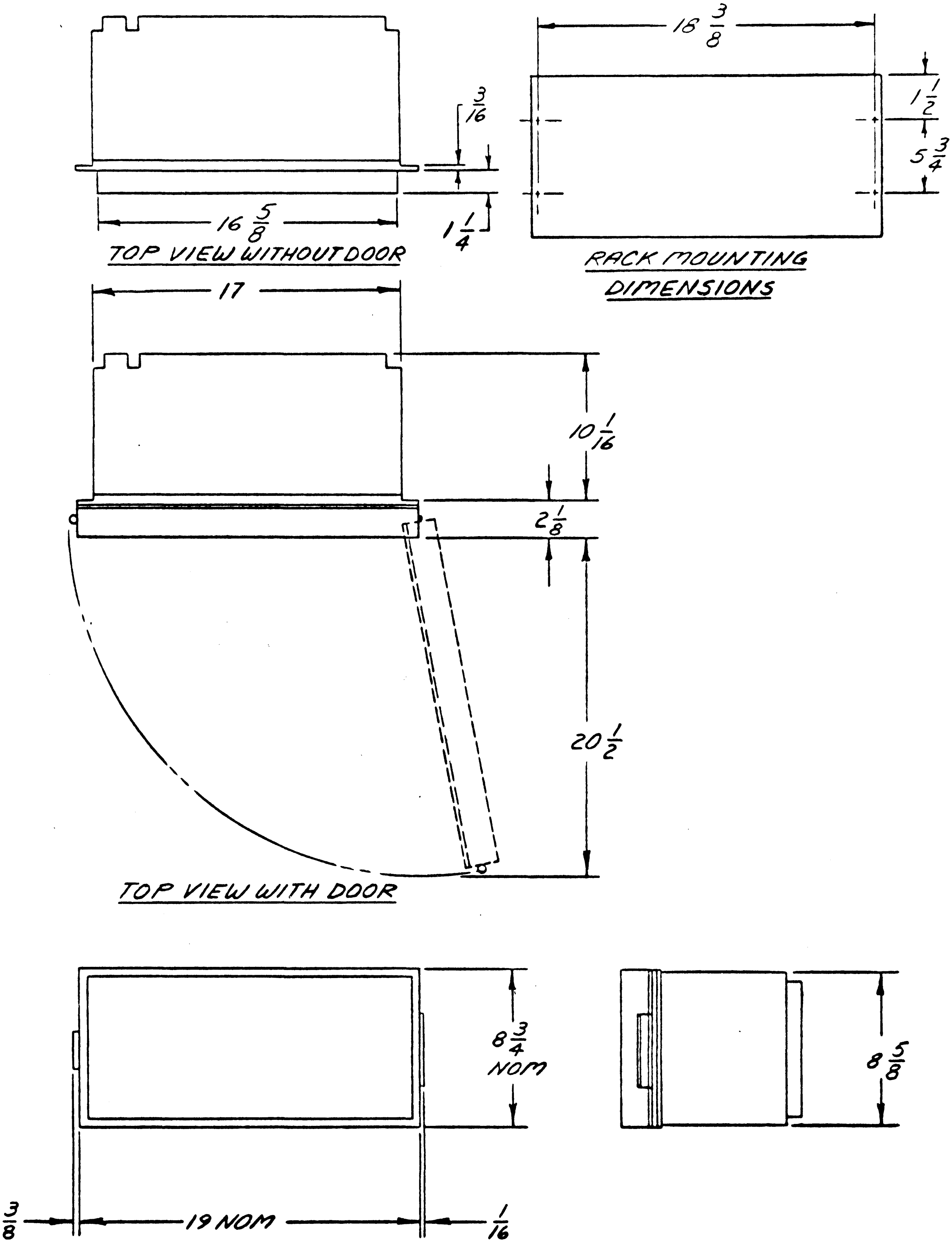
2.3 PHYSICAL

8-3/4"H X 19"W X 12-3/16"D RACK MOUNT
2-1/8 FRONT PROJECTION
1Ø-1/16 REAR PROJECTION (EXCLUDING PLUGS)
1Ø5-125 VOLTS, 6Ø HZ, 1ØØ WATTS (5Ø 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.



TOP VIEW WITHOUT DOOR

RACK MOUNTING DIMENSIONS

TOP VIEW WITH DOOR

FIGURE 2.1
OUTLINE DIMENSIONS

2.5 PACKING LIST

CO-6ØØ-NP MASTER LINCTAPE UNIT

1 EA	DUAL LINCTAPE DRIVE SYSTEM WITH READ/WRITE ELECTRONICS
1 EA	LINE CORD
1 EA	I/O CABLE
2 EA	REELS OF MARKED LINCTAPE
1 EA	INSTRUCTION BOOK

CO-6Ø5 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.Ø INSTALLATION

THE CPU MUST HAVE THE I/O CONNECTOR OPTION FOR PROPER INSTALLATION:

NOVA	DATA	GENERAL	TYPE	4Ø22
SUPERNOVA	"	"	"	8Ø22
NOVA 8ØØ	"	"	"	8222
NOVA 12ØØ	"	"	"	8122

STANDARD ON THE 121Ø, 122Ø, AND 82Ø.

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.

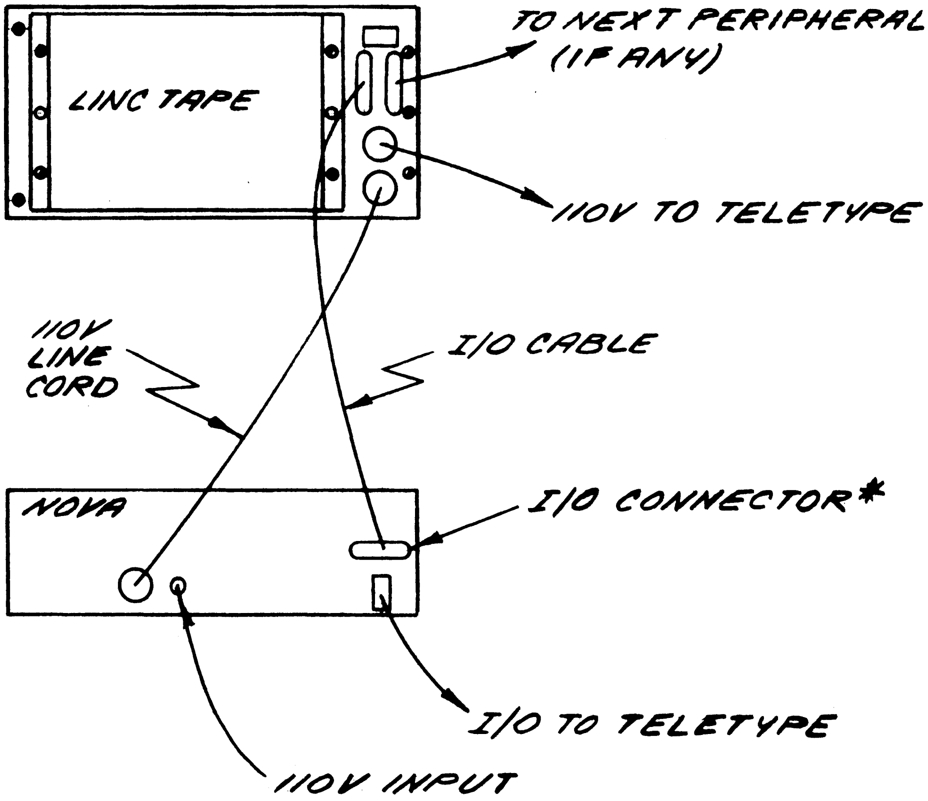
B. PLUG THE LINCTAPE POWER CABLE INTO THE OUTLET IN THE BACK OF THE COMPUTER.

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.



* 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 Ø 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 LINCTAPE 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.

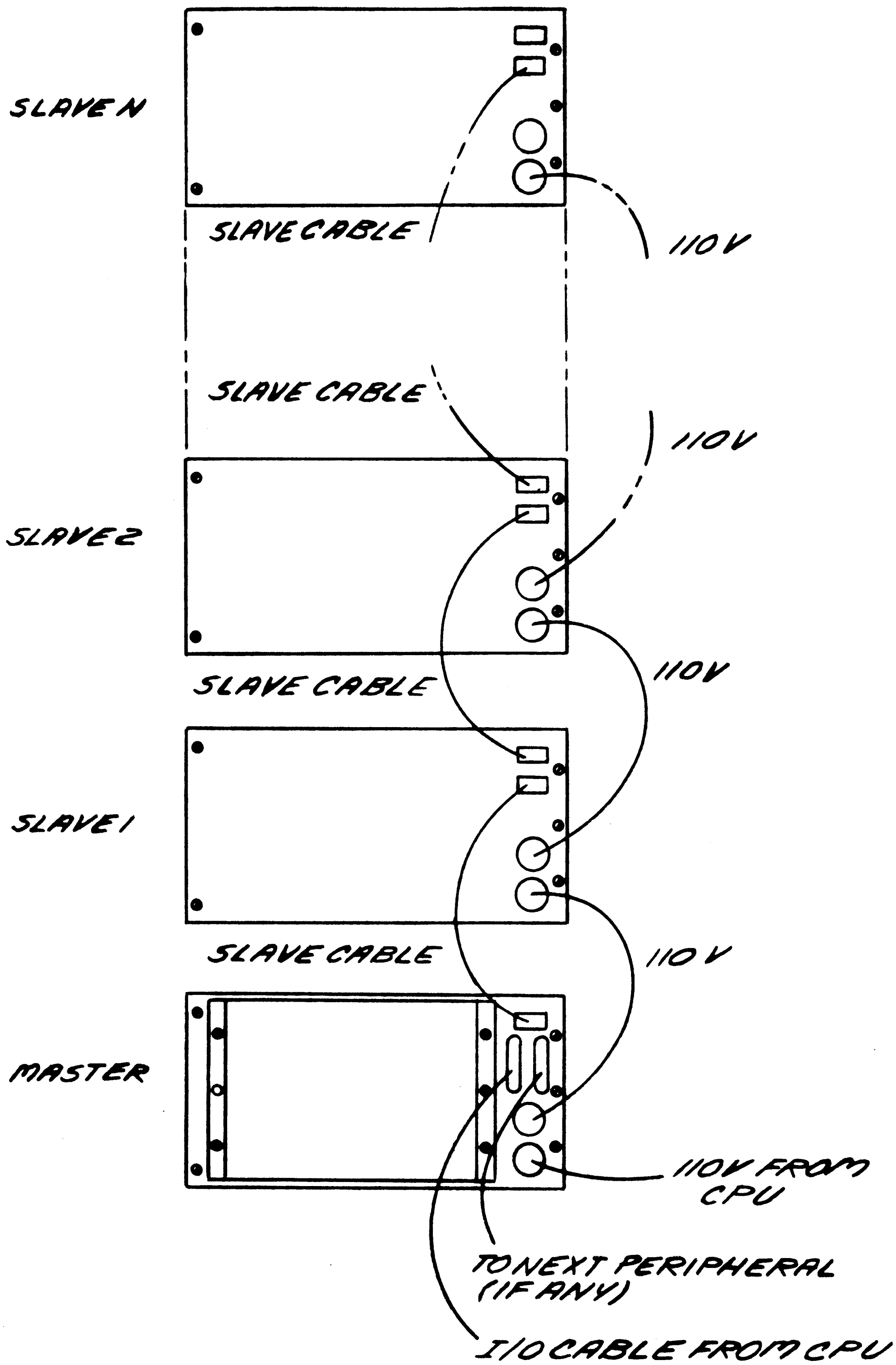
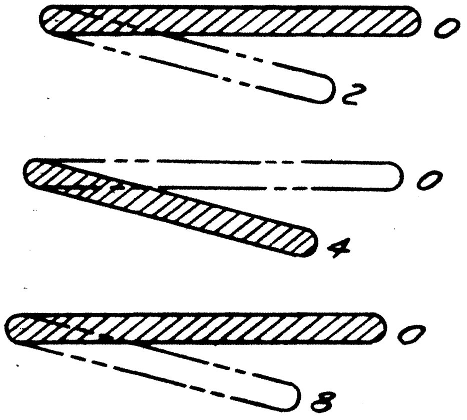


FIGURE 3.2
SLAVE CONNECTIONS



DRIVE NUMBERS 4 & 5 SHOWN

FIGURE 3.3
DRIVE NUMBER JUMPERS

4.Ø LINCTAPE OPERATION

THE LINCTAPE SYSTEM HAS TWO DRIVES WHICH MAY BE OPERATED SEPARATELY. THE LEFT DRIVE IS NUMBERED Ø (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.

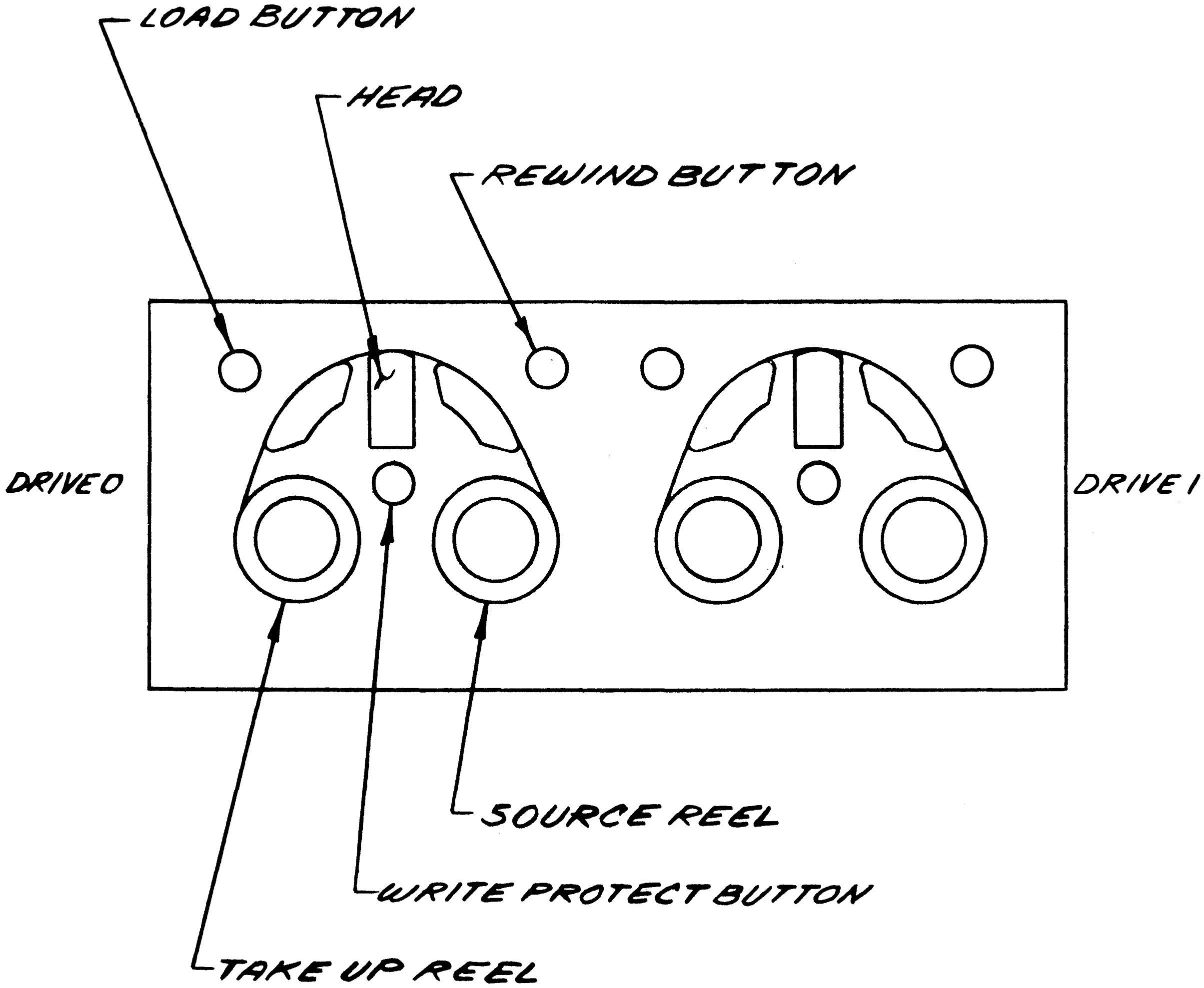


FIGURE 4.1
FRONT PANEL CONTROLS

5.Ø 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.Ø 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:

```
*1ØØØ,1Ø,4,ØR
   WHERE      *  IS THE RESPONSE GIVEN BY THE KEYBOARD
                 EXEC WHEN WAITING FOR AN OPERATOR INPUT
           1ØØØ  IS THE FIRST CORE LOCATION
             1Ø  IS THE FIRST BLOCK NUMBER
              4  IS THE NUMBER OF BLOCKS
              Ø  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 1Ø THRU 13 FROM THE TAPE ON DRIVE Ø INTO CORE LOCATIONS 1ØØØ 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 X7ØØØ (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 4ØØ (256 DECIMAL) WORDS. WHOLE BLOCKS ARE ALWAYS TRANSFERRED.
4. NEGATIVE NUMBERS MUST BE ENTERED AS TWOS 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:

       77777776 = 177776 = -2

7. NUMBERS NOT TYPED ARE ASSUMED TO BE ZERO:

       1ØØØ,,2,R MEANS CORE LOC 1ØØØ, STARTING BLOCK ZERO
       2 BLOCKS, DRIVE Ø, READ.

8. IT IS THE OPERATORS RESPONSIBILITY NOT TO READ OVER THE KEYBOARD EXEC OR THE UTILITIES. THESE RESIDE IN LOCATIONS X7ØØØ THRU X7377 AND X74ØØ 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 Ø 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.Ø 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 TOP OF CORE, AND ARE NEVER CHANGED (THEY ARE SAID TO BE RESIDENT). THEY OCCUPY LOCATIONS X74ØØ-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	X74ØØ	CHECK BLOCKS
RLINC	X74Ø3	READ & CHECK BLOCKS
WLINC	X74Ø6	WRITE & CHECK BLOCKS

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

IF AC1 = Ø, THE TAPE WILL PRE-POSITION ITSELF NEAR THE BLOCK NUMBER SPECIFIED IN ACØ.
IF AC2 IS NEGATIVE, THE DRIVE WILL START BACKWARD (THUS SAVING TIME IS 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 = Ø 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:
ACØ = BAD BLOCK NUMBER

AC1 = 2 IF THERE WAS A BLOCK SIZE ERROR:
ACØ = 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):
ACØ = 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   Ø,DRVNO
           DOB   Ø,LINC     ;SELECT DRIVE NUMBER
           ---
           LDA   Ø,FBLKN    ;SET ACØ = 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: X74Ø6            ;WRITE POINTER
    RLINC: X74Ø3            ;READ POINTER
    FBLKN: 1ØØ
    NBLKS: 1Ø
    FCORE: 1ØØØ
    DRVNO: 1
```

THE ABOVE WILL READ THE CONTENTS OF BLOCK NUMBERS 1ØØ THRU 1Ø7 FROM THE TAPE ON DRIVE 1 INTO CORE LOCATIONS 1ØØØ THRU 4777 (ALL IN OCTAL, AND ASSUMING STANDARD 4ØØ WORD BLOCKS).

8.Ø 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:

```
    LINC = DEVICE NUMBER 74 (OCTAL) NORMALLY

  DATA I/O.
    DIA  X,LINC   INPUT THE 16 BIT DATA WORD TO ACX.
    DOA  X,LINC   OUTPUT THE 16 BIT WORD IN ACX TO THE LINCTAPE.
    DIB  X,LINC   INPUT THE STATUS BITS TO ACX
                     BIT 15 ON = NOT READY
                     BIT 14 ON = WRITE PROTECTED
    DOB  X,LINC   SELECT DRIVE NUMBER.  THE DRIVE NUMBER MUST BE IN
                  THE 4 LOW ORDER BITS OF ACX.
    DOC  X,LINC   TURN WRITER 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.

SKPDN LINC	SKIP IF DATA OR CHECKSUM IS READY;
SKPDZ LINC	SKIP IF BLOCK OR NONE IS READY;
SKPBN LINC	SKIP IF BLOCK NUMBER OR CHECKSUM IS READY;
SKPBZ 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 Ø;
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. OCCASSIONALLY, HOWEVER IT IS IMPORTANT THAT THE PROGRAMMER KNOW SOME OF THE BASIC TIMES INVOLVED. THE FOLLOWING ARE APPROXIMATE, AND MAY VARY BY 1Ø % IN THE FORWARD DIRECTION AND 3Ø % IN THE REVERSE DIRECTION. SEE THE DIAGRAM IN FIGURE 1Ø.2.

TAPE END TO END	25 SECONDS
START/TURN AROUND TIME	13Ø MILLISECONDS
BLOCK TO BLOCK	63 MILLISECONDS (256 WORD BLOCKS)
WORD TO WORD	24Ø MICROSECONDS
READY (BLOCK,DATA,CKSM)	4Ø 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 13Ø MS ALWAYS OCCURS WHEN STARTING OR CHANGING DIRECTION;
5. MARKS ARE TRUE FOR 36 TO 44 US FORWARD, AND 3Ø TO 5Ø 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 22Ø 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.Ø 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 15Ø FEET LONG, AND 3/4 INCH WIDE. IT IS ABOUT Ø.ØØ15 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 4ØØ ADDRESSABLE BLOCKS, EACH CONTAINING 256 (4ØØ 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 62Ø 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 1Ø 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

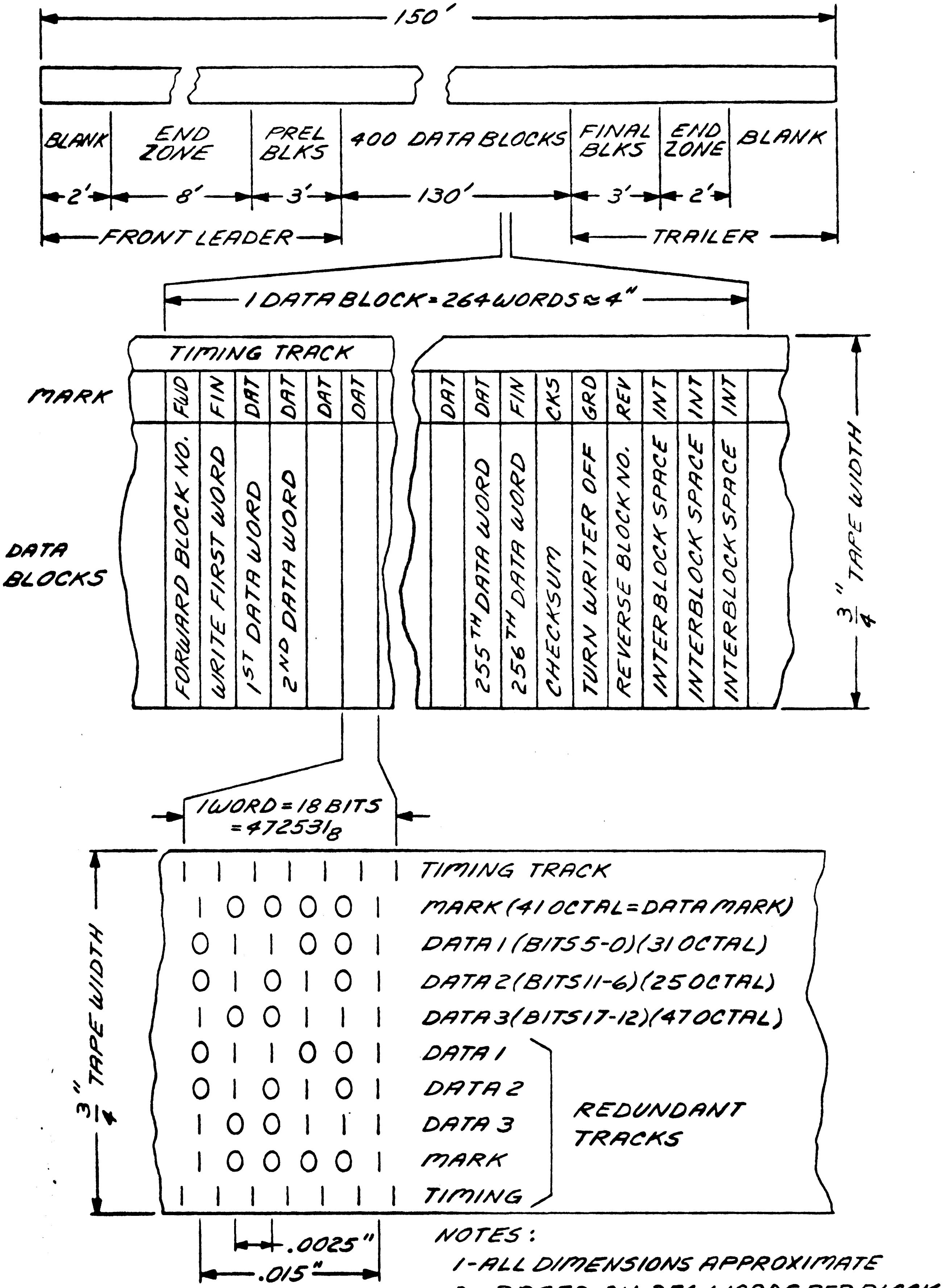
THE MARK TRACKS CONTAIN A UNIQUE SET OF CODES WHICH IS USED TO IDENTIFIY THE INFORMATION ON THE CORRESPONDING DATA TRACKS.

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.



NOTES:

1 - ALL DIMENSIONS APPROXIMATE

2 - BASED ON 256 WORDS PER BLOCK

FIGURE 9.1
LINC TAPE FORMAT

10.Ø 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 OF 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 1Ø.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, 9Ø 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.

AS THE TAPE MOVES WHILE READING, THE BITS FROM TAPE ARE SHIFTED SERIALLY INTO THE FOUR SHIFT REGISTERS AT A 25 KHZ RATE. WHEN THE PATTERN IN THE MARK REGISTER IS 41(OCTAL), THE MARK DECODING CIRCUITRY SETS THE "DATA READY" LINE. DURING THIS TIME, THE DATA REGISTERS CONTAIN THE CORRESPONDING DATA WORD. OTHER BIT PATTERNS DEFINE THE CHECKSUM AND THE BLOCK NUMBER. THESE PATTERNS ARE SIX BITS, OR 24Ø MICROSECONDS APART.

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 1Ø.2.

A CHECKSUM IS WRITTEN AT THE END OF EACH BLOCK. THIS SUM IS THE ALGEBRAIC SUM, MODULO 65,536, OF ALL THE WORDS IN THE BLOCK. THE CHECKSUM IS CALCULATED AND WRITTEN BY THE WRITE PROGRAM (UTILITIES) DURING THE WRITING OF EACH BLOCK. DURING THE READING OF EACH BLOCK, THE SUM OF THE INCOMING DATA IS CALCULATED AND COMPARED WITH THE CHECKSUM ON THE TAPE.

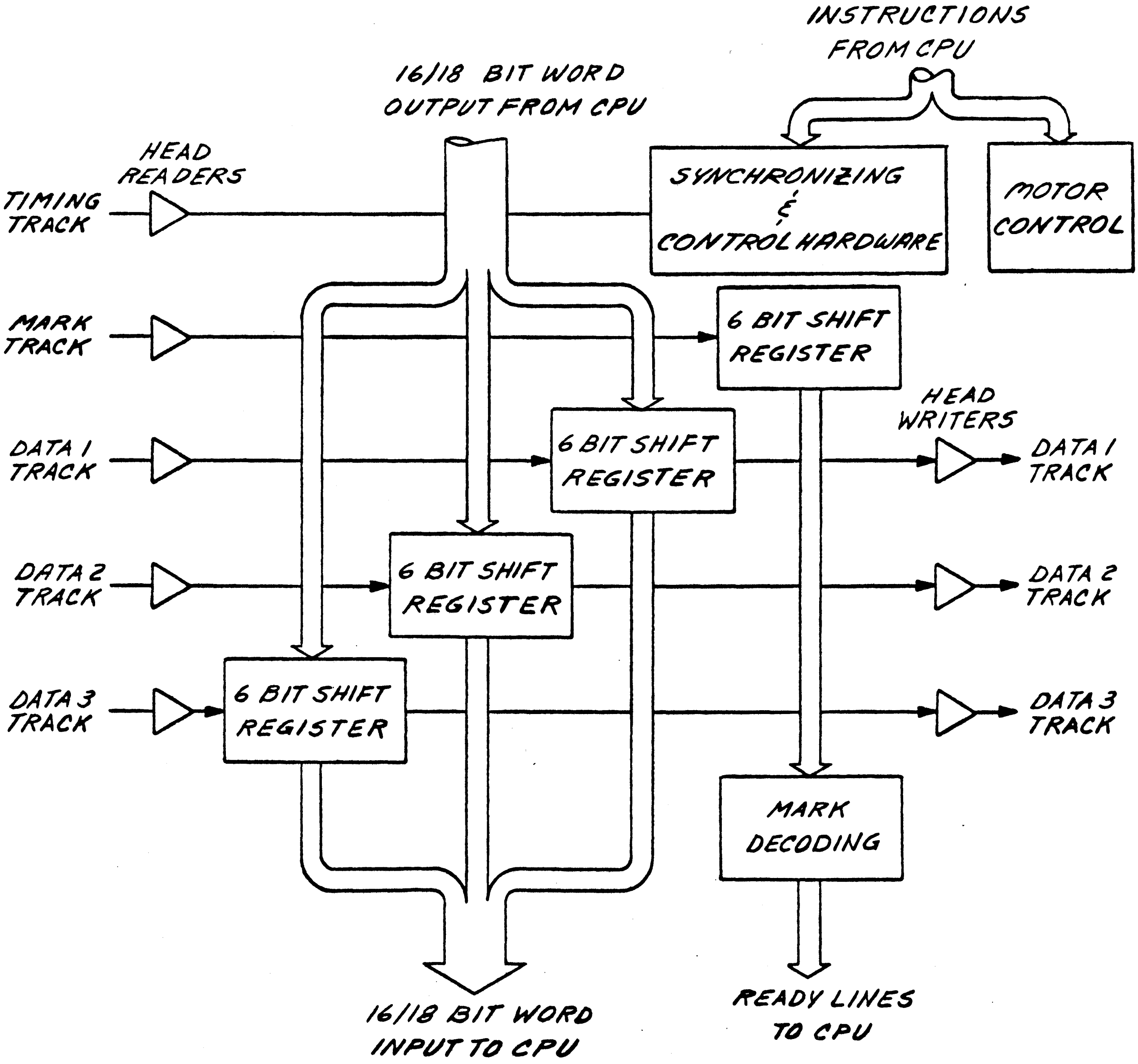
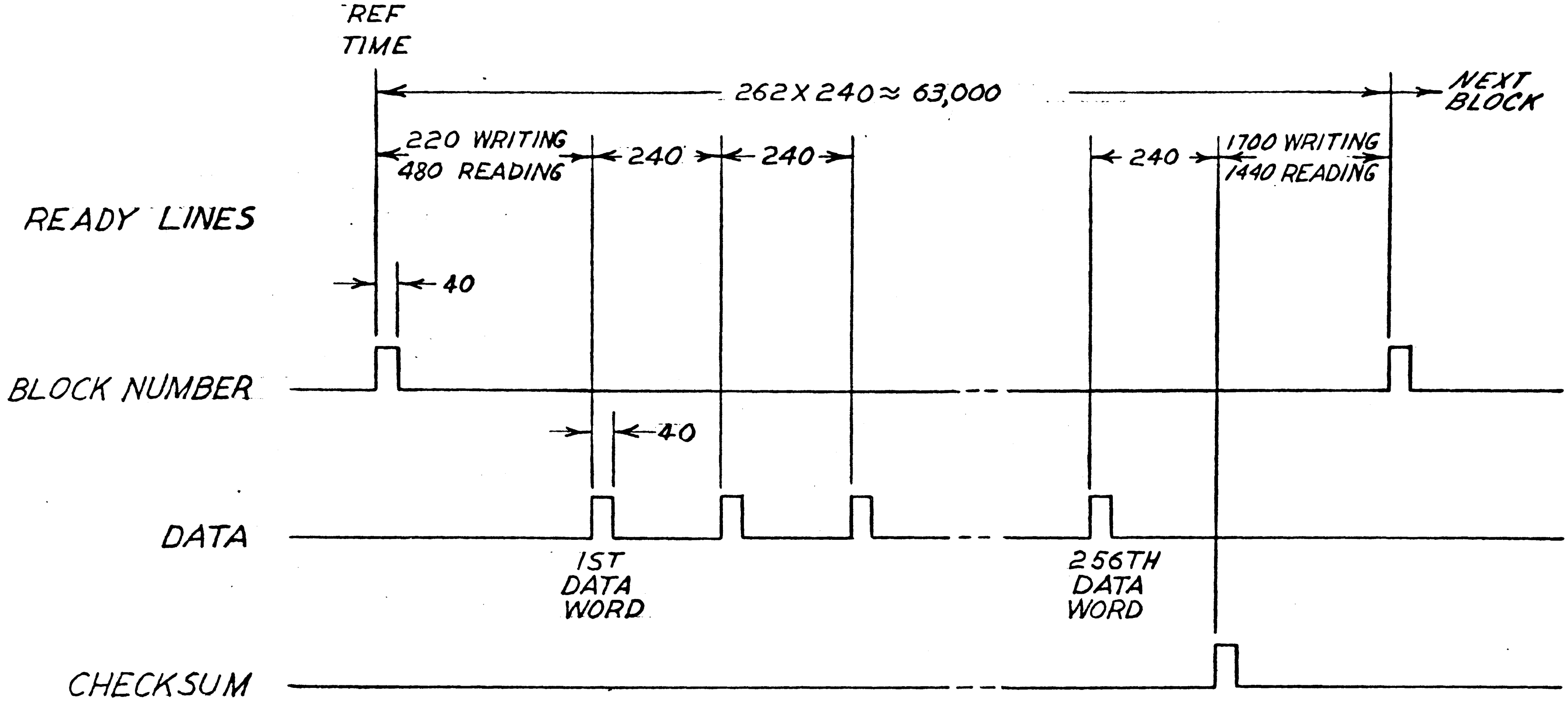


Figure 10.1 Basic LINC Tape Block Diagram



ALL TIMES IN MICROSECONDS ±10%

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 1Ø.3 AND 1Ø.4.

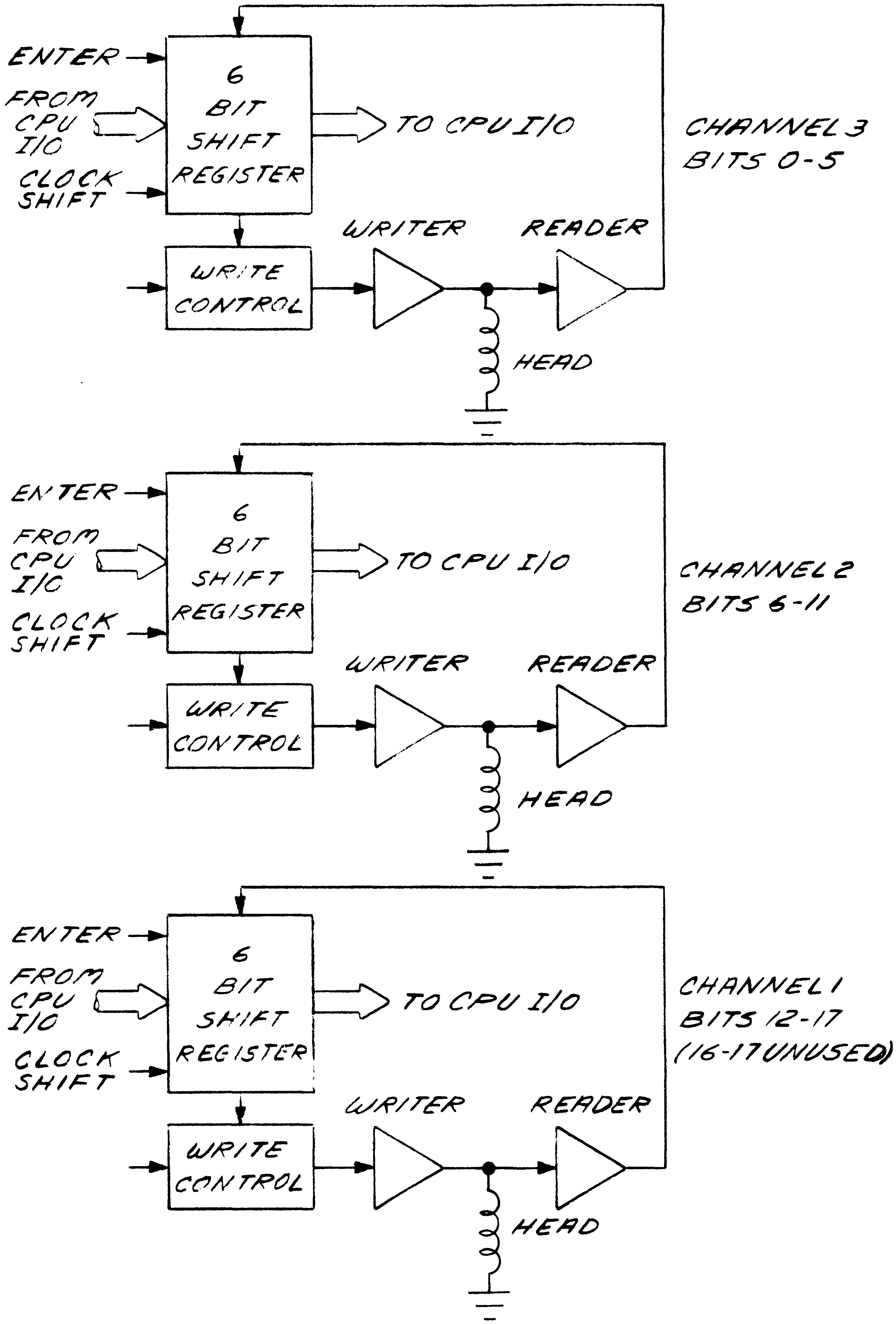


FIGURE 10.3
LINC TAPE BLOCK DIAGRAM:
DATA REGISTERS

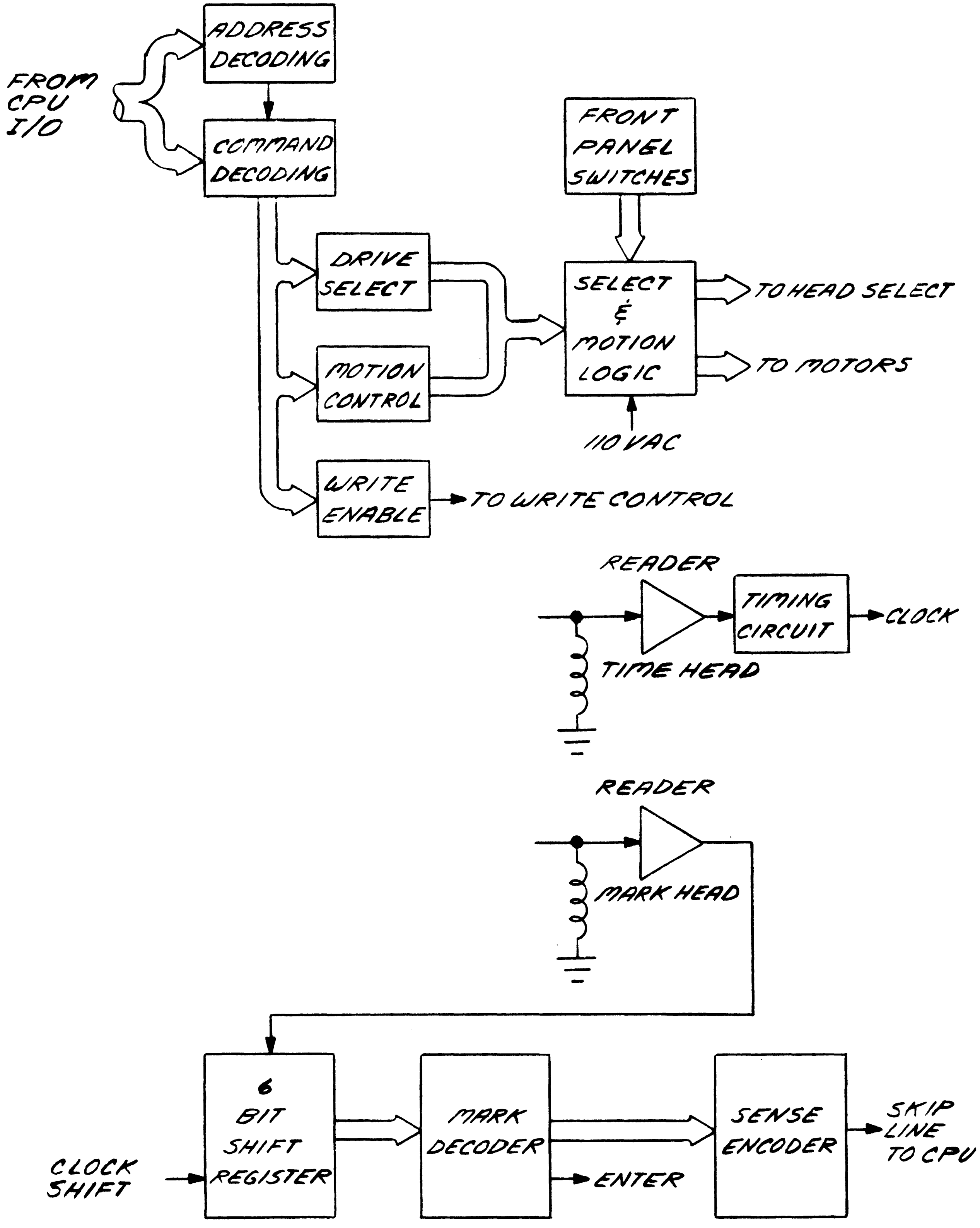


FIGURE 10.4
LINC TAPE BLOCK DIAGRAM:
CONTROL, MARK, & TIMING

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

THE MASTER CONTROL BOARD IS THE CENTRAL LOGIC BOARD FOR THE COMPLETE LINCTAPE SYSTEM. IT INTERFACES TO THE COMPUTER, AND TO ALL SLAVE UNITS. IT ALSO CONNECTS TO THE DRIVE CONTROL BOARD AND THE READER-WRITER BOARD. IT IS THE "WIRE-WRAP" BOARD ON THE BACK DOOR OF THE MASTER UNIT. ONLY ONE OF THESE BOARDS IS REQUIRED, REGARDLESS OF THE NUMBER OF SLAVE UNITS ON THE SYSTEM. THE SCHEMATICS FOR THIS BOARD ARE SHOWN IN COI DRAWING NUMBER D-10230-01 (D-10144-01 FOR S/N BELOW 1016).

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

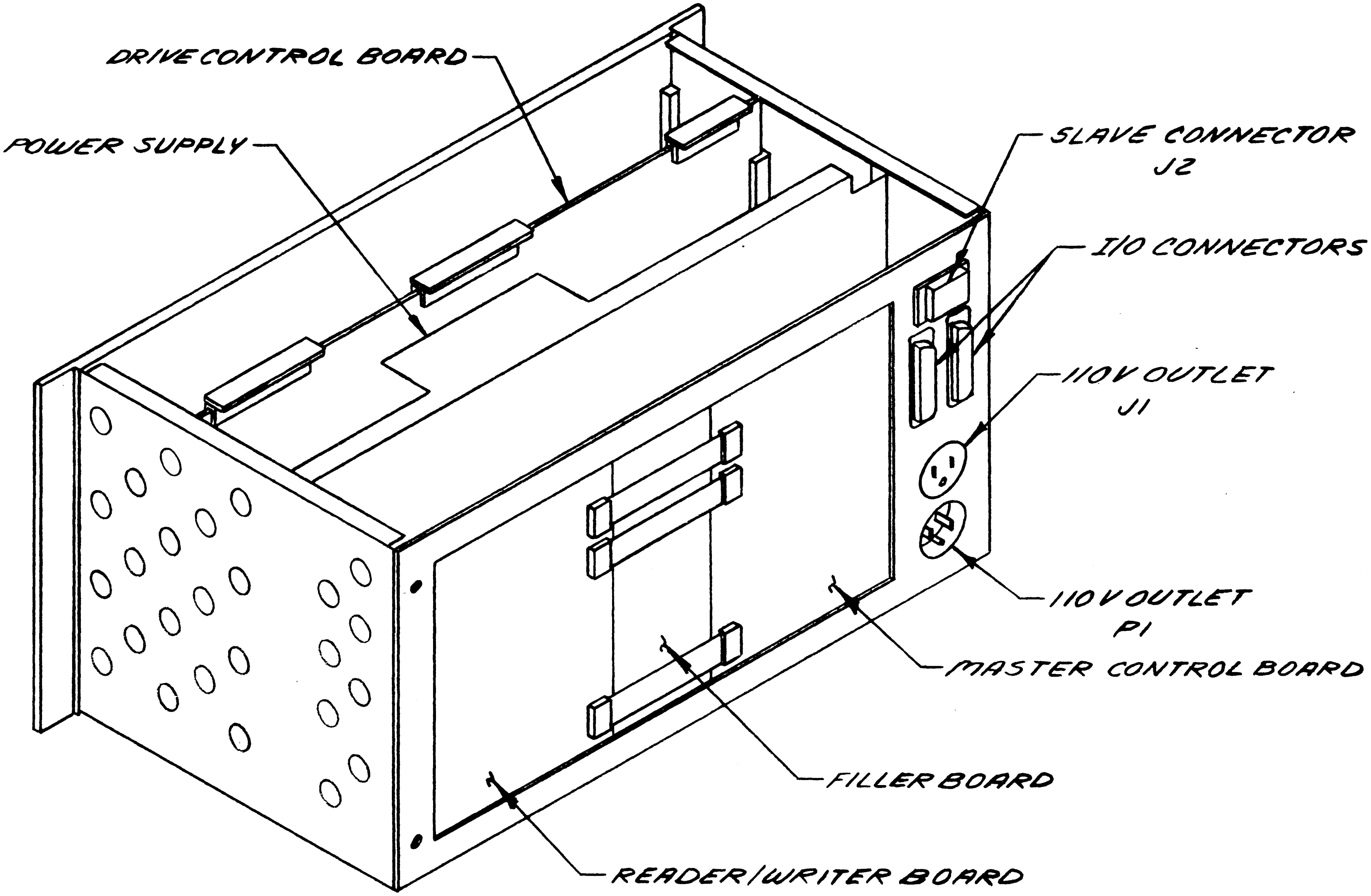
THIS P.C. BOARD, MOUNTED ON THE BACK DOOR OF THE MASTER UNIT, CONTAINS THE HEAD WRITER DRIVERS, AND THE HEAD READER AMPLIFIERS. ONLY ONE READER/WRITER BOARD IS REQUIRED PER SYSTEM, REGARDLESS OF THE NUMBER OF SLAVE UNITS. INTERCONNECTIONS TO THE R/W BOARD ARE MADE THRU THREE 16 PIN CONNECTORS WHICH PLUG INTO THE MAIN CONTROL BOARD. THE SCHEMATIC FOR THIS BOARD IS SHOWN IN COI DRAWING NUMBER D-10198-01 (D-10232-01 FOR S/N BELOW 1016).

11.4 POWER SUPPLY

THE POWER SUPPLY IS BEHIND THE DRIVE CONTROL BOARD. THE TERMINALS CAN BE ACCESSED BY OPENING THE BACK DOOR. IT SUPPLIES THREE D.C. VOLTAGES AND A "POWER FAIL" LOGIC SIGNAL. SEE COI DRAWING NUMBER D-10244-01.

11.5 INTERCONNECTION

THE INTERCONNECTION CABLING BETWEEN THE BOARDS AND CONNECTORS IS SHOWN IN FIGURE 11.2.



SHOWN WITH REAR COVER & TOP REMOVED

FIGURE 11.1
LOCATION OF MAJOR COMPONENTS
OF THE
MASTER LINC TAPE UNIT

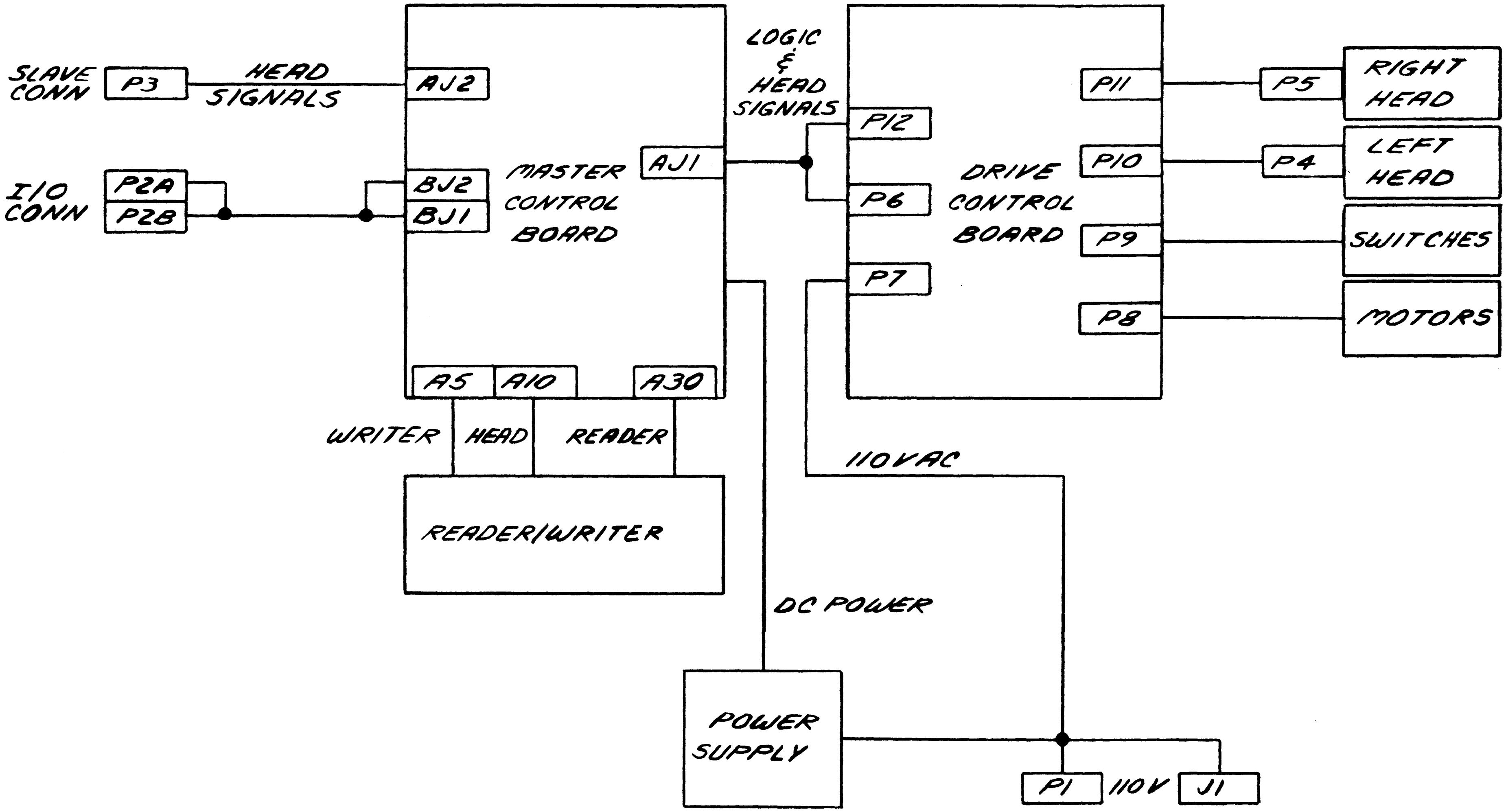


FIGURE 11.2
CABLE INTERCONNECTION

12.Ø POWER SUPPLY

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

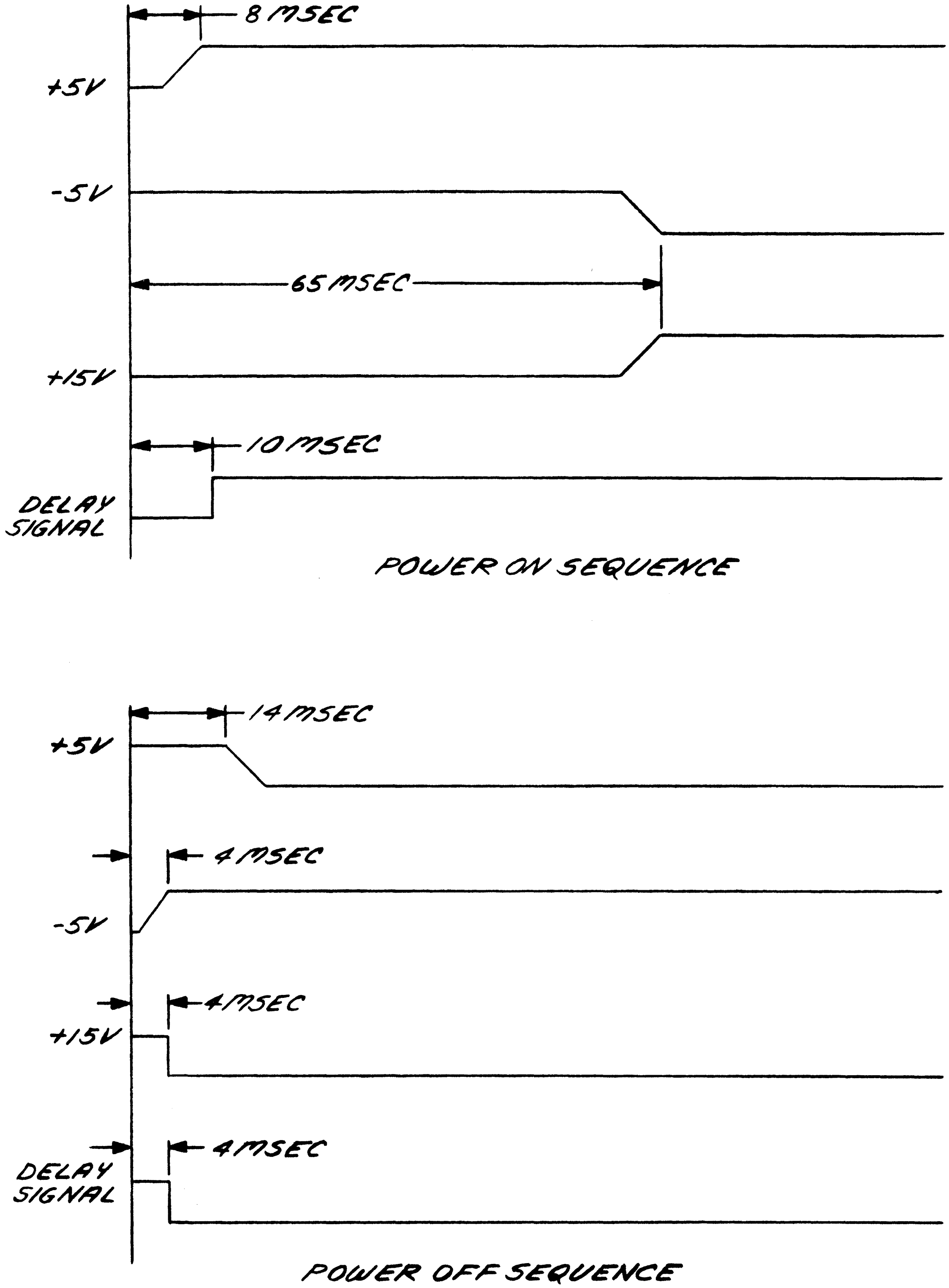
INPUT	115 V, 5Ø/6Ø HZ
OUTPUT	+5 V @ 2.5 AMPS
	-5 V @ Ø.15 AMPS
	+15 V @ 2.Ø AMPS

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.Ø MAINTENANCE

ALTHOUGH LINCTAPE 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 7Ø1Ø11Ø). 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 TAPES THEMSELVES OCCASSIONALLY 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.
2. DO THE LOAD AND REWIND BUTTONS WORK? FOR BOTH DRIVES? CAN THE RELAYS BE HEARD? IF SO, THE 11ØV AND +5 VOLT SUPPLY ARE 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 I.C. 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 OBSEVED 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  12642Ø  GET:   SUBZ  1,1        ;CLEAR AC1 & SET C
; X7757  Ø636YY         SKPDN LINC       ;BYTE READY?
; X776Ø  ØØØ777         JMP   .-1        ;NO: WAIT
; X7761  Ø6Ø5YY         DIAS  Ø,LINC     ;YES: GET BYTE
; X7762  1Ø7363         ADDCS Ø,1,SNC    ;MERGE: ANOTHER?
; X7763  ØØØ774         JMP   .-4        ;YES: GET IT
; X7764  1253Ø5         MOVS  1,1,SNR    ;NO: SWAP BACK: ZERO?
; X7765  ØØØ771         JMP   GET        ;YES: TRY AGAIN
; X7766  Ø45Ø41         STA   1,41,2     ;STORE IT
; X7767  ØØ14ØØ         JMP   Ø,3        ;RETURN WITH WORD
; X777Ø  Ø6Ø5YY  BOOT:  DIAS  Ø,LINC     ;START LINCTAPE
; X7771  Ø3Ø764         LDA   2,COUNT    ;SET WORD COUNTER
; X7772  ØØ4764         JSR   GET        ;GET A WORD
; X7773  1514Ø4         INC   2,2,SZR    ;INCR & TEST
; X7774  ØØØ776         JMP   .-2        ;NOT THRU
; X7775  ØØØØØ2         JMP   MANLD      ;GO TO MAN PRELOAD
; X7776          ; RESERVE FOR BINARY LOADER ADDRESS
; X7777          ; RESERVE FOR BINARY LOADER JUMP
;
;        WHERE:
;          X7777 = HIGHEST CORE LOCATION
;                  CORE  X    CORE  X
;                   4 K  Ø    2Ø 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 PRELOADER IS NEAR THE END OF BLOCK -1Ø OF THE
; LINCTAPE IN BYTE FORM.  THE BEGINNING OF BLOCK
; -1Ø IS ALL ZEROES.  THE UTILITY LOADER, ALSO
; IN BYTE FORM, IS AT THE BEGINNING OF BLOCK -7.
; THE UTILITIES ARE IN WORD FORM AT THE END OF BLOCK
; -7, AND THE EXECUTIVE SYSTEM, IN WORD FORM, FILLS
; BLOCK -6.
;
; THE SUPERNOVA OR THE MANUAL BOOTSTRAP SKIPS
; LEADING ZEROES, AND LOADS THE PRELOADER
; PROGRAM FROM LINCTAPE BLOCK -1Ø INTO CORE
; AT LOCATIONS Ø THRU 4Ø.  THE SUPERNOVA
; THEN JUMPS TO LOC 4Ø, 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 8ØØ/12ØØ AUTOLOADER LOADS AND EXECUTES
; ITS OWN BOOTSTRAP FROM HARDWARE INTO LOCATIONS
; Ø THRU 37.  IT THEN LOADS THE PRELOADER (WHICH
; IS IGNORED) AND THE UTILITY LOADER INTO CORE
; LOCATIONS 1ØØ 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 X76ØØ.
;
; THE LINC UTILITIES THEN REST AT X74ØØ THRU X7577.
; THE PROGRAM THEN JUMPS TO "START" IN THE UTILITIES,
; READS THE EXECUTIVE SYSTEM FROM BLOCK -6 INTO CORE
; LOCATIONS X7ØØØ THRU X7377, AND FINALLY STOPS
; THE TAPE AND JUMPS TO LOCATION X7377 WITH
; AC3 = ADDRESS OF "CLINC" (X74ØØ) OF THE UTILITIES.
;
; SOME LOCATIONS NEAR THE BOTTOM OF CORE ARE
; WIPED OUT BY THE BOOTSTRAP PROCESS. THEY ARE:
;   SUPERNOVA      MANUAL    AUTOLOADER
;      Ø- 4Ø        Ø- 4Ø       Ø-37
;     142-216     142-216     1ØØ-216
;      377                     377
```

^^^

```
                  ; NOW THE PRELOADER PROGRAM
                  ;
      ØØØØ1Ø               .RDX   8
      ØØØØØ2               .LOC   2
ØØØØ2 Ø21775  MANLD:  LDA    Ø,-3,3     ; MANUAL ENTRY LOC: GET D.N.
ØØØØ3 ØØØØØ5          JMP    .+2
ØØØØ4 Ø6Ø477  PRELOD: READS  Ø          ; GET DEVICE NUMBER
ØØØØ5 Ø24Ø26          LDA    1,C77
ØØØØ6 1234ØØ          AND    1,Ø        ; MASK 6 BITS
ØØØØ7 Ø24Ø3Ø          LDA    1,GET+1
ØØØ1Ø 1Ø7ØØØ          ADD    Ø,1
ØØØ11 Ø44Ø3Ø          STA    1,GET+1    ; SET SKIP COMMAND
ØØØ12 Ø24Ø32          LDA    1,GET+3
ØØØ13 1Ø7ØØØ          ADD    Ø,1
ØØØ14 Ø44Ø32          STA    1,GET+3    ; SET INPUT COMMAND
ØØØ15 12644Ø          SUBO   1,1        ; CLEAR AC1 & RESET CARRY
ØØØ16 ØØ4Ø3Ø          JSR    GET+1      ; GO AFTER BYTE
ØØØ17 1Ø1Ø65          MOVC   Ø,Ø,SNR    ; ZERO BYTE?
ØØØ2Ø ØØØØ16          JMP    .-2        ; YES: IGNORE IT
ØØØ21 ØØ4Ø27          JSR    GET        ; NO: GET NEXT FULL WORD
ØØØ22 Ø46Ø25          STA    1,@PTR1    ; STORE THE WORD
ØØØ23 Ø1Ø142          ISZ    142        ; LAST WORD?
ØØØ24 ØØØØ21          JMP    .-3        ; NO: GET ANOTHER WORD
ØØØ25 ØØØ141  PTR1:   141               ; POINTER TO STORE & JUMP
ØØØ26 ØØØØ77  C77:    77
              ; GET A WORD SUBR. GET A BYTE SUBR
ØØØ27 12642Ø  GET:    SUBZ   1,1        ; CLEAR AC1 & SET CARRY
ØØØ3Ø Ø636ØØ          SKPDN  Ø          ; BUFFER READY?
ØØØ31 ØØØØ3Ø          JMP    .-1        ; NO: WAIT
ØØØ32 Ø6Ø5ØØ          DIAS   Ø,Ø        ; YES: INPUT THE BYTE
ØØØ33 1Ø7363          ADDCS  Ø,1,SNC    ; SWAP BYTES. NEED ANOTHER?
ØØØ34 ØØØØ3Ø          JMP    .-4        ; YES: GO GET IT
ØØØ35 1253ØØ          MOVS   1,1        ; NO: SWAP BACK
ØØØ36 ØØ14ØØ          JMP    Ø,3        ; RETURN WITH WORD/BYTE
ØØØ37 177777          -1
ØØØ4Ø ØØØØØ4  FORTY:  JMP    PRELOD     ; SUPERNOVA ENTRY LOCATION
```

```
^^^

             ; NOW THE UTILITY LOADER
             ;
      ØØØØ1Ø         .RDX  8
      ØØØØ77         .LOC  77
ØØØ77 ØØØ377 SYNC:   377               ; SYNC BYTE
ØØ1ØØ 177661 COUNT:  SYNC-END          ; WORD COUNTER
      ØØØ142         .LOC  142
ØØ142 177723         .-END-1
             ; SET DEVICE NUMBER
ØØ143 Ø2ØØ3Ø GOGO:   LDA   Ø,GET+1     ; GET I/O WORD
ØØ144 Ø24213         LDA   1,K77
ØØ145 1234ØØ         AND   1,Ø         ; GET DEVICE NUMBER
ØØ146 Ø242Ø6         LDA   1,GETW
ØØ147 1Ø7ØØØ         ADD   Ø,1
ØØ15Ø Ø442Ø6         STA   1,GETW      ; SET SKIP COMMAND
ØØ151 Ø2421Ø         LDA   1,GETW+2
ØØ152 1Ø7ØØØ         ADD   Ø,1
ØØ153 Ø4421Ø         STA   1,GETW+2    ; SET INPUT COMMAND
             ; SIZE CORE
ØØ154 Ø2Ø214         LDA   Ø,ONEK      ; MEMORY SIZE INCREMENT
ØØ155 Ø34215         LDA   3,HIGH      ; HIGHEST OPEN MEMORY
ØØ156 1164ØØ         SUB   Ø,3         ; MOVE DOWN TO NEXT CORE
ØØ157 Ø55777         STA   3,-1,3      ; TRIAL STORAGE
ØØ16Ø Ø31777         LDA   2,-1,3      ; GET IT AGAIN
ØØ161 156414         SUB#  2,3,SZR     ; TEST FOR MATCH
ØØ162 ØØØ156         JMP   .-4         ; NO MATCH: TRY AGAIN
             ; SKIP ZERO BYTES
ØØ163 ØØ42Ø6         JSR   GETW        ; AC2 = HIGHEST OPEN CORE
ØØ164 1Ø1Ø25         MOVZ  Ø,Ø,SNR     ; IGNORE LEADING ZEROES
ØØ165 ØØØ163         JMP   .-2
             ; LOAD THE UTILITIES
ØØ166 Ø4Ø214         STA   Ø,ONEK      ; SAVE COUNT "LTSIZ"
ØØ167 113ØØØ         ADD   Ø,2         ; POINT TO LOAD ADDR
ØØ17Ø Ø5Ø213         STA   2,K77       ; SAVE POINTER
ØØ171 ØØ42Ø6         JSR   GETW        ; GET A WORD
ØØ172 Ø41ØØØ         STA   Ø,Ø,2       ; STORE IT
ØØ173 1514ØØ         INC   2,2         ; INCR POINTER
ØØ174 Ø1Ø214         ISZ   ONEK        ; INCR COUNTER
ØØ175 ØØØ171         JMP   .-4         ; IF NOT DONE, GO BACK
             ; SET UP TO LOAD KEYBOARD EXEC
ØØ176 Ø3Ø213         LDA   2,K77       ; IF DONE, GET 1ST UTIL LOC
ØØ177 Ø5117Ø         STA   2,LLOC-CLINC,2 ; & PUT IN LLOC
ØØ2ØØ Ø25163         LDA   1,SIZE-CLINC,2 ; GET WORDS/BLOCK
ØØ2Ø1 1324ØØ         SUB   1,2         ; SET 1ST CORE LOC
ØØ2Ø2 12652Ø         SUBZL 1,1         ; SET NO OF BLOCKS TO 1
ØØ2Ø3 Ø2Ø212         LDA   Ø,BLOCK     ; SET NEXT BLK NO
ØØ2Ø4 Ø34213         LDA   3,K77
ØØ2Ø5 ØØ1577         JMP   START-CLINC,3 ; GOTO START
             ; GET WORD SUBROUTINE
ØØ2Ø6 Ø636ØØ GETW:   SKPDN Ø           ; BUFFER READY?
ØØ2Ø7 ØØØ2Ø6         JMP   .-1         ; NO: WAIT
ØØ21Ø Ø6Ø5ØØ         DIAS  Ø,Ø         ; READ WORD INTO ACØ
ØØ211 ØØ14ØØ         JMP   Ø,3         ; RETURN WITH FULL WORD
ØØ212 177772 BLOCK:  -6
ØØ213 ØØØØ77 K77:    77
ØØ214 ØØ1ØØØ ONEK:   1ØØØ
ØØ215 1ØØ6ØØ HIGH:   776ØØ+1ØØØ        ; SAVE BINARY LOADER
ØØ216 ØØØ143 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 ZEROES BETWEEN THE
                ; LAST WORD OF THE UTILITY LOADER ADN "LTSIZ".
       ØØØØ1Ø           .RDX  8
       ØØ7377           .LOC  7377
 Ø7377 1776ØØ   LTSIZ:  .-START    ; LINCTAPE UTILITY SIZE
                ;
                ;   ENTRIES TO LINC UTILITIES
                ;     WLINC:  WRITE & CHECK
                ;     RLINC:  READ & CHECK
                ;     CLINC:  CHECK ONLY
                ;*** USER MUST SELECT DRIVE BEFORE CALL WITH "DOB -,74"
                ;  ALL CALLS MADE BY 'JSR' TO ONE OF THE ABOVE,
                ;      WITH ACØ= FIRST BLOCK #
                ;           AC1= NUMBER OF BLOCKS TO BE PROCESSED
                ;           AC2= FIRST CORE ADDRESS
                ;
                ;           AC1 MAY =Ø. 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=Ø.
                ;         AC2= NEXT CORE ADDRESS
                ;         ACØ= NEXT BLOCK #
                ;
                ;  ABNORMAL RETURNS HAVE THE ERROR CODE IN AC1:
                ;        AC1=1   CHECKSUM ERROR. ACØ= BAD BLOCK #
                ;
                ;        AC1=2   BLOCK SIZE ERROR. ACØ= BAD BLOCK.
                ;                AC2= EXCESS OF MARKS IN BLOCK.
                ;                AC3= EXPECTED #.
                ;
                ;        AC1=4   ILLEGAL BLOCK.  ACØ= TARGET BLOCK.
                ;                AC2= NEXT CORE ADDR.
                ;                AC3= HIGHEST LEGAL BLOCK.
                ;
                ;        AC1=8   DRIVE STATUS ERROR. AC3= DRIVE STATUS,
                ;                BIT 1= PROTECTED, BIT Ø= NOT READY.
                ;
 Ø74ØØ Ø54431   CLINC:  STA 3,SAC3
 Ø74Ø1 1524ØØ           SUB 2,2    ;ADDRESS DOESN'T MATTER
 Ø74Ø2 ØØØ417           JMP CHKZ
 Ø74Ø3 Ø54426   RLINC:  STA 3,SAC3
 Ø74Ø4 Ø3443Ø           LDA 3,D2R  ; SET READ RTNE. TO STORE DATA
 Ø74Ø5 ØØØ415           JMP READZ
 Ø74Ø6 Ø54423   WLINC:  STA 3,SAC3
 Ø74Ø7 Ø34423           LDA 3,D1W
 Ø741Ø Ø5451Ø           STA 3,D1XX ; SET UP FOR WRITE
 Ø7411 Ø445Ø1           STA 1,D2XX   ;SAVE PARAMS
 Ø7412 Ø5Ø416           STA 2,SAC2
 Ø7413 ØØ4423           JSR DO       ; FIND & WRITE BLOCKS
```

^^^

```
Ø7414 Ø24476  RAW:    LDA 1,D2XX  ;RESTORE PARAMS
Ø7415 1224ØØ          SUB 1,Ø
Ø7416 Ø3Ø412          LDA 2,SAC2
Ø7417 151113          MOVL# 2,2,SNC  ;MAKE ADDR. NEG FOR REVERSE
Ø742Ø 15ØØØØ          COM 2,2
Ø7421 Ø34473  CHKZ:   LDA 3,D2C   ; NO STORE ON CHECK
              ;
Ø7422 Ø5447Ø  READZ:  STA 3,D2XX
Ø7423 Ø3441Ø          LDA 3,D1RC
Ø7424 Ø54474          STA 3,D1XX   ; SET UP FOR READ OR CHECK
Ø7425 ØØ4411          JSR DO       ; FIND & READ BLOCKS
Ø7426 Ø6Ø274  EXIT:   NIOC LINC    ; STOP DRIVE
Ø7427 ØØ24Ø2          JMP @SAC3    ; RETURN TO CALLER
Ø743Ø ØØØØØØ  SAC2:   Ø
Ø7431 ØØØØØØ  SAC3:   Ø
              ;
Ø7432 Ø21ØØØ  D1W:    LDA Ø,Ø,2    ; DATA FOR WRITE SWITCH
Ø7433 ØØØ75Ø  D1RC:   JMP READ-D1XX,1    ; FOR READ SWITCH
Ø7434 132512  D2R:    SUBL# 1,2,SZC  ;DO THIS FOR READ, NOT CHECK
Ø7435 ØØØØØØ  RETU:   Ø
              ;
Ø7436 Ø54777  DO:     STA 3,RETU
Ø7437 Ø75474          DIB 3,LINC      ; CHECK DRIVE READY
Ø744Ø 175112          MOVL# 3,3,SZC
Ø7441 ØØØ446          JMP E4
Ø7442 151113          MOVL# 2,2,SNC ; LOOK AT ADDR.
Ø7443 ØØØ41Ø          JMP FINDF       ; IF POS, START FORWARD
Ø7444 15ØØØØ          COM 2,2      ; IF NEG, START REVERSE
              ;
Ø7445 1764ØØ  FINDR:  SUB 3,3         ; ENTER HERE FOR REV.
Ø7446 162ØØØ          ADC 3,Ø         ;POINT TO TARGET-1
Ø7447 Ø6Ø374          NIOP LINC       ;START REV
Ø745Ø ØØ4467          JSR GETBLOCK
Ø7451 1Ø14Ø1  FINDN:  INC Ø,Ø,SKP
Ø7452 ØØØ776          JMP .-2         ; KEEP GOING IF ABOVE OR ON
Ø7453 Ø6Ø174  FINDF:  NIOS LINC       ; ENTER HERE FOR FWD.
Ø7454 ØØ4463          JSR GETBLOCK
Ø7455 ØØØ777          JMP .-1         ; KEEP GOING IF BELOW
Ø7456 175224          MOVZR 3,3,SZR ; FOUND TARGET IF =Ø
Ø7457 ØØØ766          JMP FINDR
Ø746Ø 125ØØ5  FOUND:  MOV 1,1,SNR  ;LAST BLOCK?
Ø7461 ØØ2754          JMP @RETU       ; YES, EXIT FROM "DO"
Ø7462 166ØØØ          ADC 3,1         ;AC3=Ø. DECR BLOCK COUNTER
Ø7463 Ø4Ø474          STA Ø,TEMP1
Ø7464 Ø44474          STA 1,TEMP2
Ø7465 Ø24476          LDA 1,SIZE
Ø7466 147ØØØ          ADD 2,1      ;POINT TO END OF BLOCK.
Ø7467 ØØØ431          JMP D1XX
```

^^^

```
Ø747Ø Ø63674  READ:   SKPDN LINC    ; WAIT FOR DATA
Ø7471 ØØØ777          JMP .-1
Ø7472 Ø63474          SKPBN LINC
Ø7473 ØØØ416          JMP RDAT
Ø7474 Ø6Ø474  RCHK:   DIA Ø,LINC    ; INPUT CHECKSUM
Ø7475 1164Ø5          SUB  Ø,3,SNR
Ø7476 ØØØ434          JMP SCHK      ; SHOULD = ACCUM. CHKSM
Ø7477 024465  E1:     LDA 1,C1      ; CHECKSUM ERROR
Ø75ØØ ØØØ4Ø3          JMP .+3
Ø75Ø1 Ø34462  E2:     LDA 3,SIZE
Ø75Ø2 Ø24463          LDA 1,C2
Ø75Ø3 Ø2Ø454          LDA Ø,TEMP1
Ø75Ø4 ØØØ722          JMP EXIT
Ø75Ø5 Ø24461  E3:     LDA 1,C4      ; ILLEGAL BLOCK NUMBER
Ø75Ø6 ØØØ72Ø          JMP EXIT
Ø75Ø7 Ø2446Ø  E4:     LDA 1,C8      ; DRIVE STATUS ERROR
Ø751Ø ØØØ716          JMP EXIT
Ø7511 Ø6Ø474  RDAT:   DIA Ø,LINC    ;INPUT DATA WORD
Ø7512 132512  D2XX:   SUBL# 1,2,SZC  ;"JMP .+2" FOR CHECK
Ø7513 Ø41ØØØ          STA Ø,Ø,2
Ø7514 ØØØ4Ø2  D2C:    JMP .+2
Ø7515 Ø61Ø74  WDAT:   DOA Ø,LINC    ; WRITE DATA
Ø7516 117ØØØ  BLOOP:  ADD Ø,3       ; UPDATE CHKSM ACCUM.
Ø7517 1514ØØ          INC 2,2       ; UPDATE POINTER
Ø752Ø Ø21ØØØ  D1XX:   LDA Ø,Ø,2     ; FOR READ & CHECK, "JMP READ"
Ø7521 Ø65Ø74          DOC Ø,LINC   ;WRITERS ON
Ø7522 Ø63674          SKPDN LINC
Ø7523 ØØØ777          JMP .-1       ; WAIT FOR DATA, CHECK MARK
Ø7524 Ø63474          SKPBN LINC
Ø7525 ØØØ77Ø          JMP WDAT      ; DATA MARK
Ø7526 Ø75Ø74  WCHK:   DOA 3,LINC    ; WRITE CHECKSUM
Ø7527 Ø75474          DIB 3,LINC   ;INPUT DRIVE STATUS
Ø753Ø 175ØØ4          MOV 3,3,SZR
Ø7531 ØØØ756          JMP E4       ;MUST BE READY & UNPROTECTED
Ø7532 132414  SCHK:   SUB# 1,2,SZR ;WAS BLOCK RIGHT SIZE?
Ø7533 ØØØ746          JMP E2
Ø7534 Ø2Ø423  NEXT:   LDA Ø,TEMP1   ;BLOCK FINISHED
Ø7535 Ø24423          LDA 1,TEMP2   ;RESTORE BLOCK COUNTER
Ø7536 ØØØ713          JMP FINDN  ; NEXT BLOCK
              ;
              ;
```

```
^^^

07537 054420  GETBLOCK: STA 3,TEMP1
07540 034421            LDA 3,MLIM    ;CHECK TARGET LIMITS
07541 162432            SUBZ# 3,0,SZC ; OK IF BETWEEN MLIM & PLIM
07542 000405            JMP WAIT
07543 034417            LDA 3,PLIM
07544 162032            ADCZ# 3,0,SZC
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 063774            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 Ø. THIS PROCEDURE IS USED BY
                        ;   THE BOOT PROG.
                        ;   JUMP TO "START" WITH:
                        ;         ACØ=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 WORD LOADED +1,
                        ;         AC3= ADDR OF "CLINC" IN UTILITIES.
                        ;
Ø757Ø ØØØØØØ  LLOC:     Ø
Ø7571 ØØ4612            JSR   RLINC
Ø7572 Ø34776            LDA   3,LLOC
Ø7573 125ØØ5            MOV   1,1,SNR
Ø7574 ØØ1377            JMP   -1,2
Ø7575 Ø62677  PAUSE:   IORST          ;STOP & SELECT DRIVE Ø
Ø7576 Ø63Ø77           HALT           ;IF ERROR, SET AC'S & CONTINUE
Ø7577 ØØØ772  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,DX  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 Ø.

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 SIGNFICANT (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 "BOMB" THE SYSTEM BY LOADING OVER THE EXECUTIVE PROGRAM, FOR EXAMPLE. THE EXECUTIVE RESIDES AT X7ØØØ-X7377 OCTAL (WHERE X7777 IS THE HIGHEST CORE LOCATION), AND IT CALLS ON THE LINC UTILITIES AT X74ØØ-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 1ØØØ 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 X7ØØØ 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 Ø. IT WILL NOT WRITE, NOR WILL IT ACCESS ANY OTHER DRIVE.

FOR INSTANCE, TYPING "A" CAN READ THE ASSEMBLER FROM THE TAPE ON DRIVE Ø 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 YYØØØ 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 Ø

(4) READ A FRESH COPY OF THE EXEC INTO WORKING CORE (THE EXEC IS ON BLOCK -6, OR 177772):

```
1ØØØ,177772,1,ØR
```

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) ADD, 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:

```
        ØØ1242  ØØØ113   ASCII LETTER K
        ØØ1243  ØØ1ØØØ   1ST CORE LOCATION
        ØØ1244  ØØØØ3Ø   1ST BLOCK NUMBER
        ØØ1245  ØØØØ14   NUMBER OF BLOCKS
        ØØ1246  ØØØØØØ   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:

```
        1ØØØ,177772,1,ØW
```

(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 Ø, TO 617, OCTAL.
6.INVALID COMMAND.

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

EXAMPLES

```
*1ØØ,2,1,ØR
*                READ, STARTING AT CORE ADDRESS 1ØØ,
                 ONE BLOCK, STARTING AT BLOCK 2, DRIVE Ø.

*Ø,77777777777,1,1W
*                        WRITE BLOCK -1 FROM CORE ADDR Ø,DRIVE 1

*Ø,177777,1,1W
?
*                ERROR. SAME COMMAND AS ABOVE.

*Ø,Ø,62Ø,ØC      CHECK TAPE BLOCKS Ø THROUGH 617.
*,,62Ø,ØC        SAME THING.

*A
*                LOAD EXTENDED ASSEMBLER.
```

THIS ROUTINE IS LISTED AS ASSEMBLED AT LOCATION 7ØØØ.
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 COMANDS
                        ;       TYPE 1 IS THE SINGLE LETTER TYPE WHICH IS USED
                        ;       PRIMARILY TO READ IN SYSTEM PROGRAMS FROM
                        ;       DRIVE Ø ONLY.   THE PROGRAM CONTAINS A LOOKUP
                        ;       TABLE FOR THESE COMMANDS WHICH CAN BE EASILY
                        ;       UPDATED WITH DEBUGER.
                        ;       THE TYPE TWO COMANDS ARE READ, WRITE AND
                        ;       CHECK.  THESE COMMANDS REQUIRE 4 PARAMETERS
                        ;       AS EXPLAINED IN THE INSTRUCTION BOOK


        ØØ7ØØØ                  .LOC    7ØØØ

  Ø7ØØØ ØØØ5Ø4  EXEC:   JMP     ELIN     ; OUTPUT CR "*
  Ø7ØØ1 1024ØØ          SUB     Ø,Ø      ; CLEAR ACØ
  Ø7ØØ2 Ø4Ø571          STA     Ø,TEM1   ; CLEAR INPUT ARRAY
  Ø7ØØ3 Ø4Ø571          STA     Ø,TEM2
  Ø7ØØ4 Ø4Ø571          STA     Ø,TEM3
  Ø7ØØ5 Ø4Ø571          STA     Ø,TEM4
  Ø7ØØ6 Ø2Ø552          LDA     Ø,C4     ; SET RE-TRY COUNT
  Ø7ØØ7 Ø4Ø571          STA     Ø,ERCO
  Ø7Ø1Ø ØØ45Ø3          JSR     INOC     ; INPUT OCTAL NUMBER
  Ø7Ø11 Ø44562          STA     1,TEM1
  Ø7Ø12 Ø3Ø552          LDA     2,C54    ; COMA??
  Ø7Ø13 142415          SUB#    2,Ø,SNR
  Ø7Ø14 ØØØ422          JMP     COMA


  Ø7Ø15 Ø24543          LDA     1,C4     ; LOOK UP INPUT CHARACTER
  Ø7Ø16 Ø34573          LDA     3,TABL
  Ø7Ø17 Ø314ØØ  EAGN:   LDA     2,Ø,3
  Ø7Ø2Ø 151ØØ5          MOV     2,2,SNR  ; IF END OF TABLE GRIPE
  Ø7Ø21 ØØØ457          JMP     NOGO
  Ø7Ø22 142415          SUB#    2,Ø,SNR  ; IF MATCH GO TO IT
  Ø7Ø23 ØØØ4Ø3          JMP     FNDI
  Ø7Ø24 137ØØØ          ADD     1,3      ; UPDATE POINTER
  Ø7Ø25 ØØØ772          JMP     EAGN     ; AND TRY AGAIN


  Ø7Ø26 Ø214Ø1  FNDI:   LDA     Ø,1,3    ; GET PARAMETERS AND SAVE THEM
  Ø7Ø27 Ø4Ø544          STA     Ø,TEM1
  Ø7Ø3Ø Ø214Ø2          LDA     Ø,2,3
  Ø7Ø31 Ø4Ø543          STA     Ø,TEM2
  Ø7Ø32 Ø214Ø3          LDA     Ø,3,3
  Ø7Ø33 Ø4Ø542          STA     Ø,TEM3
  Ø7Ø34 Ø2Ø536          LDA     Ø,CR
  Ø7Ø35 ØØØ414          JMP     COM1+1   ; GO TO IT
```

^^^

```
Ø7Ø36 ØØ4455  COMA:   JSR     INOC      ; GET NEXT NUMBER
Ø7Ø37 Ø44535          STA     1,TEM2    ; SAVE IT
Ø7Ø4Ø Ø3Ø524          LDA     2,C54     ; CHECK FOR COMA
Ø7Ø41 142414          SUB#    2,Ø,SZR
Ø7Ø42 ØØØ436          JMP     NOGO
Ø7Ø43 ØØ445Ø          JSR     INOC      ; GET NEXT NUMBER
Ø7Ø44 Ø44531          STA     1,TEM3
Ø7Ø45 142404          SUB     2,Ø,SZR
Ø7Ø46 ØØØ432          JMP     NOGO
Ø7Ø47 ØØ4444          JSR     INOC
Ø7Ø5Ø Ø44526          STA     1,TEM4
Ø7Ø51 Ø3453Ø          LDA     3,TABL1 ; LOOK UP CHARACTER
Ø7Ø52 0314ØØ  COM1:   LDA     2,Ø,3
Ø7Ø53 1754ØØ          INC     3,3
Ø7Ø54 1754ØØ          INC     3,3
Ø7Ø55 151015          MOV#    2,2,SNR
Ø7Ø56 ØØØ422          JMP     NOGO      ; IF END OF TABLE GRIPE
Ø7Ø57 142414          SUB#    2,Ø,SZR
Ø7Ø6Ø ØØØ772          JMP     COM1      ; IF NOT MATCH TRY AGAIN
Ø7Ø61 Ø35777          LDA     3,-1,3    ; GET EXECUTION ADDRESS
Ø7Ø62 Ø54515          STA     3,FDDR    ; SAVE EXEC ADDRESS


Ø7Ø63 Ø2Ø511  TRYAGN: LDA     Ø,TEM2    ; STARTING BLOCK #
Ø7Ø64 Ø24511          LDA     1,TEM3    ; NUMBER OF BLOCKS
Ø7Ø65 Ø3Ø511          LDA     2,TEM4    ; DRIVE #
Ø7Ø66 Ø72Ø74          DOB     2,74      ; SELECT DRIVE
Ø7Ø67 Ø3Ø5Ø4          LDA     2,TEM1    ; STARTING CORE LOCATION
Ø7Ø7Ø Ø345Ø7          LDA     3,FDDR    ; GET EXECUTION ADDRESS
Ø7Ø71 ØØ54ØØ          JSR     Ø,3       ; GO TO IT
Ø7Ø72 125Ø15          MOV#    1,1,SNR ; IF NO ERROR RETURN TO START
Ø7Ø73 ØØØ7Ø5          JMP     EXEC
Ø7Ø74 125223          MOVZR   1,1,SNC ; IF CHECKSUM TRY AGAIN
Ø7Ø75 ØØØ4Ø3          JMP     NOGO      ; ELSE GIVE UP
Ø7Ø76 Ø145Ø2          DSZ     ERCO
Ø7Ø77 ØØØ764          JMP     TRYAGN    ; TRY THREE TIMES


Ø71ØØ Ø2Ø462  NOGO:   LDA     Ø,C15
Ø71Ø1 ØØ4447          JSR     OAØ
Ø71Ø2 Ø2Ø466          LDA     Ø,CQUES
Ø71Ø3 ØØ4445          JSR     OAØ
Ø71Ø4 Ø2Ø456  ELIN:   LDA     Ø,C15
Ø71Ø5 ØØ4443          JSR     OAØ
Ø71Ø6 Ø2Ø463          LDA     Ø,CSTAR
Ø71Ø7 ØØ4441          JSR     OAØ
Ø711Ø Ø2Ø453          LDA     Ø,C4Ø
Ø7111 ØØ4437          JSR     OAØ
Ø7112 ØØØ667          JMP     EXEC+1
```

^^^

```
                     ;        CONVERT AN ASCII OCTAL CHARACTER STRING TO A BINARY
                     ;        NUMBER IN AC1, AND A BREAK CHARACTER IN ACØ

Ø7113 Ø54425   INOC:    STA      3,RTRN   ; SAVE RETURN ADDRESS
Ø7114 Ø50425            STA      2,TEMØ   ; SAVE AC2
Ø7115 1Ø24ØØ            SUB      Ø,Ø
Ø7116 Ø4Ø424            STA      Ø,OCTL   ; CLEAR RESULT WORD
Ø7117 ØØ4424   INOC1:   JSR      IAØ      ; GET A CHARACTER
Ø712Ø Ø3Ø445            LDA      2,C6Ø
Ø7121 Ø34445            LDA      3,C67
Ø7122 162Ø33            ADCZ#    3,Ø,SNC  ; TEST FOR 6Ø<=N<=67
Ø7123 112Ø32            ADCZ#    Ø,2,SZC
Ø7124 ØØØ411            JMP      EINOC    ; NO MUST BE BREAK CHARACTER
Ø7125 1424ØØ            SUB      2,Ø      ; MAKE IT OCTAL
Ø7126 Ø24414            LDA      1,OCTL
Ø7127 12512Ø            MOVZL    1,1      ; OLD TIMES 8
Ø713Ø 12512Ø            MOVZL    1,1
Ø7131 12512Ø            MOVZL    1,1
Ø7132 1Ø7ØØØ            ADD      Ø,1      ; NEW PLUS OLD
Ø7133 Ø444Ø7            STA      1,OCTL
Ø7134 ØØØ763            JMP      INOC1    ; LOOP UNTIL BREAK CHARACTER

Ø7135 Ø3Ø4Ø4   EINOC:   LDA      2,TEMØ   ; RESTORE AC2
Ø7136 Ø244Ø4            LDA      1,OCTL   ; RESULT TO AC1
Ø7137 ØØ24Ø1            JMP      @RTRN

Ø714Ø ØØØØØØ   RTRN:    Ø                 ; SAVE RETURN LOCATION
Ø7141 ØØØØØØ   TEMØ:    Ø                 ; STORAGE FOR AC2
Ø7142 ØØØØØØ   OCTL:    Ø                 ; RESULT



                     ;        INPUT AND ECHO TELETYPE CHARACTER AND MASK PARITY

Ø7143 Ø6361Ø   IAØ:     SKPDN    TTI      ; WAIT FOR INPUT CHARACTER
Ø7144 ØØØ777            JMP      .-1
Ø7145 Ø6Ø61Ø            DIAC     Ø,TTI    ; INPUT CHARACTER TO ACØ
Ø7146 Ø3Ø421            LDA      2,C177   ; MASK OFF PARITY BIT
Ø7147 1434ØØ            AND      2,Ø

Ø715Ø Ø63511   OAØ:     SKPBZ    TTO      ; WAIT FOR OUTPUT         READY
Ø7151 ØØØ777            JMP      .-1
Ø7152 Ø61111            DOAS     Ø,TTO    ; OUTPUT ACØ AND START
Ø7153 Ø3Ø4Ø7            LDA      2,C15
Ø7154 142434            SUBZ#    2,Ø,SZR  ; IF CR OUTPUT LF
Ø7155 ØØ14ØØ            JMP      Ø,3
Ø7156 Ø2Ø4Ø3            LDA      Ø,C12
Ø7157 ØØØ771            JMP      OAØ
```

^^^

```
                    ;        CONSTANTS
Ø716Ø ØØØØØ4  C4:      4
Ø7161 ØØØØ12  C12:     12
Ø7162 ØØØØ15  C15:     15
Ø7163 ØØØØ4Ø  C4Ø:     4Ø
Ø7164 ØØØØ54  C54:     54
Ø7165 ØØØØ6Ø  C6Ø:     6Ø
Ø7166 ØØØØ67  C67:     67
Ø7167 ØØØ177  C177:    177
Ø717Ø ØØØØ77  CQUES:   "?
Ø7171 ØØØØ52  CSTAR:   "*
Ø7172 ØØØ122  CR:      "R
Ø7173 ØØØØØØ  TEM1:    Ø
Ø7174 ØØØØØØ  TEM2:    Ø
Ø7175 ØØØØØØ  TEM3:    Ø
Ø7176 ØØØØØØ  TEM4:    Ø
Ø7177 ØØØØØØ  FDDR:    Ø
Ø72ØØ ØØØØØØ  ERCO:    Ø
```

```
^^^



                      ;        VALUES IN THIS TABLE WILL BE FILLED IN BY THE
                      ;        SIZE ROUTINE AT LOAD TIME
Ø72Ø1 ØØØØØØ   TABL1:  Ø                       ; ADDRESS OF TABLE
Ø72Ø2 ØØØ122           "R                      ; READ LINC TAPE
Ø72Ø3 ØØØØØØ   LR:     Ø
Ø72Ø4 ØØØ1Ø3           "C
Ø72Ø5 ØØØØØØ   LC:     Ø
Ø72Ø6 ØØØ127           "W
Ø72Ø7 ØØØØØØ   LW:     Ø
Ø721Ø ØØØØØØ           Ø                       ; 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:
                      ;         Ø7212  ØØØ1Ø4  "D
                      ;         Ø7213  ØØ1ØØØ  1ØØØ
                      ;         Ø7214  ØØØØØØ   Ø
                      ;         Ø7215  ØØØ14   14
                      ;         Ø7216  ØØØØØØ  Ø
                      ;        WOULD LOAD LOCATIONS 1ØØØ THRU 6777 INTO CORE
                      ;        FROM BLOCKS Ø THRU 13 ON THE TAPE ON DRIVE Ø,
                      ;        WHEN THE LETTER "D" WAS TYPED.
                      ;
                      ;        THERE IS ROOM IN THE TABLE FOR 23 (DECIMAL)
                      ;        COMMAND LETTERS.
                      ;



Ø7211 ØØØØØØ   TABL:   Ø                       ; ADDRESS OF TABLE
Ø7212 ØØØØØØ           Ø                       ; END OF TABLE
```

^^^

```
                  ;        THIS ROUTINE DETERMINES THE SIZE OF CORE AND
                  ;        SETS ALL OF THE POINTERS.

      ØØ7352               .LOC    EXEC+352
Ø7352 Ø24421  SIZE:        LDA     1,C4ØØ
Ø7353 1324ØØ               SUB     1,2
Ø7354 141ØØØ               MOV     2,Ø
Ø7355 Ø24421               LDA     1,TABP
Ø7356 133ØØØ               ADD     1,2
Ø7357 Ø51377               STA     2,-1,2
Ø736Ø Ø24415               LDA     1,TAB2P
Ø7361 133ØØØ               ADD     1,2
Ø7362 Ø51377               STA     2,-1,2
Ø7363 Ø54622               STA     3,LC
Ø7364 Ø2441Ø               LDA     1,C3
Ø7365 137ØØØ               ADD     1,3
Ø7366 Ø54615               STA     3,LR
Ø7367 137ØØØ               ADD     1,3
Ø737Ø Ø54617               STA     3,LW
Ø7371 111ØØØ               MOV     Ø,2
Ø7372 ØØ1ØØØ               JMP     Ø,2
Ø7373 ØØØ4ØØ  C4ØØ:        4ØØ
Ø7374 ØØØØØ3  C3:          3
Ø7375 ØØØØ1Ø  TAB2P:       TABL-TABL1
Ø7376 ØØØ2Ø2  TABP:        TABL1-EXEC+1
Ø7377 ØØØ753               JMP     SIZE
                           .END
```

APPENDIX D. LINCTAPE MAP

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

BLOCK NUMBER	PROGRAM
-1Ø	PRELOADER (BYTE FORM)
-7	UTILITY LOADER (BYTE FORM)
	LINCTAPE UTILITIES
-6	KEYBOARD EXECUTIVE
-5 THRU -1	ZEROES