1972

A hardware and software interface between a magnetic tape unit and a Nova mini-computer

Thomas Dean Steury

Follow this and additional works at: http://scholarsmine.mst.edu/masters_theses

Part of the Electrical and Computer Engineering Commons

Recommended Citation

Steury, Thomas Dean, "A hardware and software interface between a magnetic tape unit and a Nova mini-computer" (1972). Masters Theses. Paper 4190.

This Thesis - Open Access is brought to you for free and open access by the Student Research & Creative Works at Scholars' Mine. It has been accepted for inclusion in Masters Theses by an authorized administrator of Scholars' Mine. For more information, please contact weaverjr@mst.edu.
A HARDWARE AND SOFTWARE INTERFACE
BETWEEN A MAGNETIC TAPE UNIT
AND A NOVA MINI-COMPUTER

BY

THOMAS DEAN STEURY, 1948-

A THESIS

Presented to the Faculty of the Graduate School of the

UNIVERSITY OF MISSOURI-ROLLA

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

1972

Approved by

[Signatures]
ABSTRACT

This thesis is concerned with the general operation of the Kennedy Model 1400R incremental write/continuous read magnetic tape unit and the hardware design and software implementation needed to make it compatible for use with the Data General Nova 800 mini-computer.

The hardware design was developed around a General Purpose Interface Board which provides a standard set of logic needed for interfacing any external device to the Nova. The hardware consists of decode networks, mode selection networks, and I/O timing networks. Special circuits needed to implement the hardware design and perform voltage conversions between the tape unit and the Nova computer are included. The complete hardware design utilizes TTL Integrated Circuit logic.

A software package was written in Nova assembler language which allows the user to write programs or data on magnetic tape and be able to read the programs or data off the tape when needed. Portions of these programs are discussed in this thesis.

The result of this hardware design and software implementation is a device by which faster input and output can be obtained.
ACKNOWLEDGEMENT

The author wishes to express his sincere appreciation to Dr. James Tracey for his patience and guidance in the preparation of this thesis and to Dr. Hugh Spence for his technical assistance. Appreciation is also extended to Wayne Omohundro for his assistance in testing both the hardware design and the software implementation.

Special thanks goes to my wife Janet for her patience in this undertaking and for her excellent typing ability.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>ii</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENT</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF ILLUSTRATIONS</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>vii</td>
</tr>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II. TAPE DECK OPERATION</td>
<td>3</td>
</tr>
<tr>
<td>A. Incremental Write Description</td>
<td>3</td>
</tr>
<tr>
<td>B. Continuous Read Description</td>
<td>7</td>
</tr>
<tr>
<td>C. Pushbutton Controls</td>
<td>9</td>
</tr>
<tr>
<td>1. Load Forward</td>
<td>9</td>
</tr>
<tr>
<td>2. Ready</td>
<td>10</td>
</tr>
<tr>
<td>3. File Gap</td>
<td>10</td>
</tr>
<tr>
<td>4. Rewind</td>
<td>10</td>
</tr>
<tr>
<td>D. Loading Procedure</td>
<td>10</td>
</tr>
<tr>
<td>III. HARDWARE DESIGN</td>
<td>13</td>
</tr>
<tr>
<td>A. General Description of Nova Computer</td>
<td>13</td>
</tr>
<tr>
<td>B. General Purpose Interface</td>
<td>14</td>
</tr>
<tr>
<td>C. Special Circuits</td>
<td>19</td>
</tr>
<tr>
<td>1. Clamping Circuit</td>
<td>20</td>
</tr>
<tr>
<td>2. Monostable Multivibrator</td>
<td>22</td>
</tr>
<tr>
<td>3. Delay Circuit</td>
<td>26</td>
</tr>
<tr>
<td>4. Open Collector Inverters</td>
<td>28</td>
</tr>
</tbody>
</table>
Table of Contents (continued)  

D. I/O Control Networks ........................ 29  
1. Decode Network .......................... 29  
2. Read/Write Select Network ............... 32  
3. Input/Output Timing Networks .......... 34  

IV. SOFTWARE ........................................ 41  
A. Basic I/O Instructions .................. 41  
B. Program Techniques ....................... 43  

V. CONCLUSION ...................................... 48  

BIBLIOGRAPHY ....................................... 49  
VITA .................................................. 50  
APPENDICES ........................................ 51  
A. Signal Abbreviation List ............... 51  
B. Software Examples ....................... 53
# LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figures</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. NRZI Recording</td>
<td>4</td>
</tr>
<tr>
<td>2. End-of-Record and End-of-File Gaps</td>
<td>6</td>
</tr>
<tr>
<td>3. Skew Delay</td>
<td>8</td>
</tr>
<tr>
<td>4. Tape Path</td>
<td>12</td>
</tr>
<tr>
<td>5. Device Select Logic and Control Line Buffer</td>
<td>16</td>
</tr>
<tr>
<td>6. I/O Data Buffer and Registers</td>
<td>17</td>
</tr>
<tr>
<td>7. Interrupt Control and Acknowledge</td>
<td>18</td>
</tr>
<tr>
<td>8. Clamping Circuit</td>
<td>21</td>
</tr>
<tr>
<td>9. Monostable Multivibrator and Signals</td>
<td>24</td>
</tr>
<tr>
<td>10. Delay Circuit and Signals</td>
<td>27</td>
</tr>
<tr>
<td>11. Decode Network</td>
<td>30</td>
</tr>
<tr>
<td>12. Read/Write Select Network</td>
<td>33</td>
</tr>
<tr>
<td>13. Input Timing Circuit and Signals</td>
<td>35</td>
</tr>
<tr>
<td>14. Output Timing Circuit</td>
<td>37</td>
</tr>
<tr>
<td>15. Output Timing Signals</td>
<td>38</td>
</tr>
<tr>
<td>16. Flow Chart Showing Format of Data on Tape</td>
<td>44</td>
</tr>
<tr>
<td>Table</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>I. Pulse Duration vs. Resistance and Capacitance</td>
<td>25</td>
</tr>
<tr>
<td>II. Bit Assignments for Control Signals</td>
<td>31</td>
</tr>
<tr>
<td>III. Bit Assignments for Status Signals</td>
<td>42</td>
</tr>
<tr>
<td>IV. Signal Abbreviation List</td>
<td>51</td>
</tr>
</tbody>
</table>
I. INTRODUCTION

The University of Missouri-Rolla Electrical Engineering Department is working on the development of a complete computer system for their facilities. This system is based around a Nova 800 general purpose computer with a 16K, 16 bit word memory and a cycle time of 800 nanoseconds. The system presently consists of the Nova 800 computer, digital-to-analog and analog-to-digital converters, a graphics display terminal, a high speed data line which provides access to an IBM 360 computer through another Nova 800 located within the school's computer center, and an ASR-33 teletype containing a keyboard, printer, and paper-tape punch and reader.

The programs which are used with the Nova in the Electrical Engineering Department are punched out on paper tape through the teletype punch. The only local method of entering these programs into the computer is through the teletype reader. The teletype reader and punch are slow input/output devices which transfer data at the rate of 10, eight-bit characters per second. Some of the larger programs on tape take over 20 minutes to read in through the teletype reader. A faster input/output device which can be used locally is clearly needed.

Available for use was a Kennedy incremental write/continuous read tape unit. With this unit, programs could be stored on magnetic tape and later read off the tape when needed. The advantages of storing programs on magnetic
tape instead of paper tape are substantial. Programs stored on magnetic tape take up less space and require less time to be read into memory. A program on paper tape requiring 20 minutes to be read into memory would take approximately 20 seconds if it were stored on magnetic tape.

This thesis is concerned with the general operation of the Kennedy tape unit and the hardware design and software implementation to make it compatible for use with the Data General Nova 800 general purpose computer. A similar thesis was written which describes the interface logic needed to connect the same tape unit to an SCC-650 computer [1]. There is an interesting difference between the two hardware designs. The software written for the SCC-650 interface is discussed in [2]. The software written for each interface, Nova and SCC-650, have only slight differences.
II. TAPE DECK OPERATION

A. Incremental Write Description

Of several schemes for digital magnetic recording, the Kennedy 1400R tape deck uses the most widely adopted IBM NRZI system (Figure 1). Under this system, digital information is recorded as transitions between saturation states of the tape magnetic coating. A transition between saturation states in either direction records a logic one, while no transition records a logic zero.

The Kennedy tape deck has seven recording tracks. A character is represented by a combination of six bits which are recorded simultaneously in six of the seven tracks. For purposes of error detection, a parity bit is recorded in the seventh track such that an odd number of logic ones are recorded for each character. If one bit of a character is written wrong, the character will then have even parity and the error will be detected.

Characters are spaced 0.005 inches apart or at a density of 200 per inch. A character is recorded onto the tape upon receiving a Write-Step command which steps the drive motor one increment of 0.005 inches. Depending on the rate at which the Write-Step command is received, the write rate of the tape unit is 0-300 characters per second.

At intervals, a gap must be inserted to allow the tape transport to stop. These gaps are designated as record gaps because they separate data into records (Figure 2). A
Figure 1. NRZI Recording
record gap contains a longitudinal check character (LCC) which is internally generated by the tape deck so that even parity will exist in each track for that record. This check character is separated from the last character in the record by 0.023 inches or approximately five blank characters. The total length of the record gap is three-fourths inch.

Files on tape are separated by file gaps (Figure 2). File gaps contain a file mark (15 in binary) and its LCC (15 in binary also) which are both internally generated by the tape unit. The file mark occurs approximately 2.7 inches from the last character in the last file. The LCC occurs five character spaces later or approximately 0.025 inches from the file mark. The LCC is followed by three-fourths inch of blank tape. The total length of the file gap is 3.5 inches.

A beginning-of-tape gap is 3.5 inches of blank tape starting at the beginning-of-tape marker. This gap is internally generated when the tape unit is in the Write mode and the beginning-of-tape marker is sensed while the stepper motor is driving the tape forward. No file marks or LCC's are written.

A write ring is located on the back of every reel of tape. If this ring is in place the magnetic tape may be used for recording data. If the ring is removed, the data which was previously recorded on the tape is protected and cannot be erased. This feature allows data to be stored on tape which cannot be erased accidentally.
Figure 2. End-of-Record and End-of-File Gaps
B. Continuous Read Description

The Kennedy 1400R tape deck is a continuous read tape unit. As long as a Load Forward signal is applied, data will appear as a continuous stream of pulses together with a clock signal. The reading rate is dependent upon the drive rate of the stepper motor which is 1,000 characters per second. During the read operation the stepper motor moves in a smooth motion and not in steps as in the write operation.

In reading magnetic tape, presence of a character is indicated by at least one bit in the logical one state being read. Because of tolerances, not all bits in a character will be read simultaneously (Figure 3). In order to assure that all bits in a character have been sensed before data is gated out, a skew delay is provided. When the first bit of the character is read, the delay allows approximately one-half character spacing for the remaining bits to be sensed. This is the maximum available time under worst case conditions. At the end of the skew delay, the character read is transmitted to the data lines.

Under the Incremental Write Description it was pointed out that a parity bit was used with each six bit character to provide odd parity over the seven tracks. The tape deck can provide even parity for each character but then there is a problem. When a character is zero (000000) the parity bit for even parity is also zero. Since there are no bits in the logical one state, the presence of the character will
Skew delay - starts when first bit in character is sensed. Maximum delay = $\frac{1}{2}T_c$

Figure 3. Skew Delay
C. Pushbutton Controls

Four pushbutton controls are located on the front panel for local control of the tape deck. These controls allow the user to load and remove tapes easily and to place the unit in any condition required to transfer data. The functions of each of these pushbuttons can be duplicated by computer control when the Read-Write-Remote selector switch is in the Remote position. The four pushbutton controls and their functions are as follows.

1. Load Forward

With power on, pressing the Load Forward button causes tape to be advanced at the rate of 1,000 steps per second in search of the beginning-of-tape marker. If the selector switch is in the Write mode, a 3.5 inch beginning-of-tape gap is produced immediately upon the sensing of the beginning-of-tape marker. If the selector switch is in the Read mode, the tape will stop upon sensing the beginning-of-tape marker. At any time after the beginning-of-tape marker has been sensed, pressing the Load Forward button causes tape to be advanced at the rate of 1,000 steps per second as long as the button is depressed. If the tape deck is ready and the selector switch is in the Write mode, tape is erased as it advances.
2. Ready

The Ready light indicates that the tape deck is ready to write data. After the beginning-of-tape marker has been sensed the Ready condition is automatically achieved and the Ready light will be on until a Rewind command is given. If desired, pressing Ready while the Load Forward action described is in process will place the tape unit in the Ready condition without the presence of the beginning-of-tape marker.

3. File Gap

If the tape unit is Ready and in the Write mode a file gap with file mark is written on tape when the button is depressed.

4. Rewind

Once depressed, Rewind cannot be stopped until the beginning-of-tape marker is sensed. Upon the sensing of the beginning-of-tape marker, the driver motor will stop but inertia will carry the marker several inches into the supply reel. To remove the tape depress Load Forward and Ready together momentarily and then depress Rewind.

D. Loading Procedure

Open the dust cover by pulling gently on the left side.

With the main power on and the toggle switch located below the left hub off, place a reel of tape on the supply
hub (left). The reel of tape should have a beginning-of-tape marker located on the surface of the tape. Make sure the reel is on perfectly straight before the knob in the center of the tape reel is tightened by turning it in a clockwise direction.

Thread the tape around the tension ring, through the photo-cell, over the head, between the pinch roller and the capstan, and around the tension ring onto the take-up reel (Figure 4).

Wind about two turns onto the take-up reel.

Turn the toggle switch below the left hub to the on position. Make sure the Read-Write-Remote selector switch is in the Remote position.

Turn the Data General Nova computer on. Flip the Reset-Stop toggle switch on the front panel of the computer to Reset.

Depress the Load Forward button on the tape unit and release. The tape should move forward until the beginning-of-tape marker is directly under the photocell. The Ready light should now be on. The tape deck is ready for operation.

For further information about the Kennedy 1400R magnetic tape unit, see the unit's operation manual [3]. Information about other magnetic tape units that use the IBM NRZI mode of recording are discussed in [4]. Machine parameters are given and the IBM NRZI system of recording is described in detail.
III. HARDWARE DESIGN

A. General Description of Nova Computer

The Data General Corporation Nova computer comes in two models, the table top and rack-mounted models. The rack-mounted model is the one that will be discussed. The model comes in two different size chassis. The five and one-quarter inch unit has seven slots for the mounting of 15 x 15 inch printed circuit boards. The ten and one-half inch unit has provisions for mounting 17 printed circuit boards of size 15 x 15 inches. These printed circuit boards may be memory modules, central processing units, or input/output control boards. All memory boards are connected to the CPU through the memory bus. All input/output control boards (otherwise known as interfaces) are connected to the CPU through the I/O bus. External lines from the interfaces to their respective peripheral devices are connected through the back panel of the computer. The back panel of the computer connects directly to the edge connectors of the printed circuit boards.

The I/O bus consists of sixteen bidirectional data lines, six device selection lines and 19 control lines from the CPU to the interfaces, and six control lines from the interfaces to the CPU. Signals on the control lines from the processor synchronize the transfer on the data lines, start and stop the device, and control the program interrupt. Signals on the control lines from the interface to the
processor indicate the states of its control flip flops, Busy, Done, and Interrupt Request. The bus signals and their description can be found in the Nova manual [5] page A3.

B. General Purpose Interface

Before proceeding it may be helpful to refer to Appendix A for the meaning of the abbreviations used in all the sections that follow. The 15 x 15 inch printed circuit board for I/O control mentioned in the previous section has a standard set of logic and a basic configuration. This standard set of logic which is needed by every external device to be used with the Nova makes up what is known as the General Purpose Interface. Approximately half of the board is consumed by this standard set of logic. The remainder of the board is configured to accept 65 (14 or 16 pin) integrated circuits, or 50 (14 or 16 pin) plus nine 24 pin integrated circuits of the designers logic. The board is layed out such that I.C. sockets and wire wrap pins may be used.

A series of some 200 wire wrap pins separates the two halves of the board. Some of these wire wrap pins connect to the standard logic and CPU signals. Others are connected to the back panel to be used for external connection of the device.

The Device Select logic, part of the General Purpose Interface, is controlled by a series of jumpers which are
set such that a particular six-bit device code will be recognized (Figure 5). The output of the Device Select logic is connected to the control line buffer to gate the control signals STRT, CLR, IOPLS, DATIA, DATOA, DATIB, DATOB, DATIC, and DATOC in from the CPU when the device is selected. The other control signals RQENB, MSKO, INTA, and IORST are input to the control buffer directly without dependence upon the device selection.

The I/O portion of the General Purpose Interface consists of the I/O Data Buffer and the Input and Output Data Registers (Figure 6). Data which is computer output from an accumulator is passed through the I/O Data Buffer and clocked into the Output Data Register. As soon as the data has been clocked in the Output Data Register it appears on the external data lines to the device. Data which is input to the computer is first clocked into the Input Data Register from the external lines. A signal from the CPU then gates the data through the I/O Buffer to the proper accumulator.

The interrupt logic of the General Purpose Interface consists of an Interrupt Control circuit and an Interrupt Chain and Acknowledge circuit (Figure 7). The Interrupt Control circuit consists of four D-type flip flops BUSY, DONE, INT DIS, and INT REQ. The BUSY and DONE flip flops monitor the state of the device. If a STRT pulse is given, the BUSY flip flop is set which starts the device. When
Figure 5. Device Select Logic and Control Line Buffer
Figure 6. I/O Data Buffer and Registers
Figure 7. Interrupt Control and Acknowledge
the device completes its operation, DONE is set and BUSY is clear.

To request an interrupt, DONE must be set and the INT DIS flip flop must be clear. This flip flop is controlled by a particular mask bit assigned for that device. It is set only when the proper mask bit in the accumulator is set and a MSKO signal is generated. With DONE set and INT DIS clear, the INT REQ flip flop is set when a RQENB signal is received. The RQENB signal is generated every machine cycle.

An interrupt chain is then used to determine if a device of higher priority has interrupted. The chain is interconnected to all the other interfaces in the order of the interrupt priority of the devices. As an INTP signal is passed through the chain, if an INT REQ flip flop of a device is set the chain will be broken, the INTP signal will not pass to the next device, and that device can now interrupt. Upon the receipt of an INTA signal, the device code is put onto data lines 10-15. These data lines can be decoded to determine what device has interrupted.

The General Purpose Interface logic was only covered in a general sense in this thesis. For more detailed drawings and descriptions, see the Nova manual [5].

C. Special Circuits

As was mentioned above, the logic of the General Purpose Interface was already provided by Data General
Corporation for use with their Nova computer. The General Purpose Interface only controlled signals between the CPU and the interface. Networks must be designed and built to control signals between the interface and a specific external device. What follows is a discussion of special circuits that are only parts of certain networks in the total interface design. These circuits warrant special attention because of their importance. The networks which comprise the total interface design will be discussed in the next section and can also be found in [6]. Further information about switching circuits like the ones in this section can be found in [7].

1. Clamping Circuit

The Kennedy tape unit is powered by a 10 volt supply. As a consequence, most of the signals output from the tape deck are 10 volts when at a logical one state. These same signals are outputs from circuits with a 1K or lower impedance. This means at least 10 milliamps of current is drawn from a tape deck signal when it is at a logical one state.

The General Purpose Interface Board is made up of integrated circuits which are powered by a five volt supply. At the input of an I.C. the maximum voltage allowed is 5.5 volts and the maximum current allowed is one milliamp. Because of these limitations, the tape deck signals are not input directly to the Input Data Register but are passed through a Clamping Circuit first (Figure 8). The tape deck signals are input to the cathode of diode 1. The anodes of
Figure 8. Clamping Circuit
diodes 1 and 2 are connected to an input of the Input Data Register. The cathode of diode 2 is connected to a five volt supply.

Diode 1 is in the Clamping Circuit to block the large current flow from the tape deck. When the tape deck signal is high, diode 1 is reversed bias and the voltage on the input of the Data Register is dependent upon the internal circuitry of the Data Register I.C. itself. Since TTL logic is used, the input is applied at the emitter of an NPN transistor. The voltage at the emitter will be the supply voltage of the I.C. minus the voltage drop across the base resistor and base-emitter junction. This voltage will be approximately 3.3 volts. When the tape deck signal is low, diode 1 is forward biased and the voltage at the input of the Data Register will be approximately the same voltage as the signal.

Diode 2 is in the Clamping Circuit to guarantee that the voltage at the input of the Data Register will not exceed five volts. If diode 1 breaks down, the large amount of current from the tape deck signal will be absorbed by the five volt power supply. The input will then be within the voltage and current limitations allowed by the I.C., and the I.C. will not be damaged.

2. Monostable Multivibrator

Some of the signals from the computer to the tape deck are voltage levels which only need to hold their value for
a minimum period of time. Other signals output from the computer are required to be pulses of certain duration in order for the tape deck circuitry to receive them properly. All of the computer pulse output signals originate from pulse signals from the CPU. The pulse signals from the CPU all have a constant duration. In order to make the CPU pulses the proper duration for input to the tape deck, they must pass through a monostable multivibrator (basically a flip flop with only one stable state). A configuration using one 7400 Quad two-input NAND gate I.C., a resistor, a capacitor, and a diode was developed to perform as a monostable multivibrator (Figure 9).

The input is at gate 1, and the output is at gate 3. The diode is placed in the circuit so that the output of the monostable will not trigger until the input pulse drops its voltage from high to low. The output pulse will drop its voltage from high to low upon triggering and stay low until the capacitor charges up. At this time the output pulse will return to a high voltage level. The duration of the output pulse is dependent upon the values of the resistor and capacitor.

In Table I, the output pulse duration is given for different values of resistors and capacitors. These values were measured using the same I.C. and may be slightly different when using another I.C. due to the internal resistance and capacitance of each I.C. Therefore, use the values in the table only as approximations. If the pulse duration
Figure 9. Monostable Multivibrator and Signals
<table>
<thead>
<tr>
<th>Resistance in Kilohms</th>
<th>1.0</th>
<th>5.3</th>
<th>18.0</th>
<th>24.3</th>
<th>47.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacitance in Microfarads</td>
<td>.01</td>
<td>.05</td>
<td>.1</td>
<td>.5</td>
<td>1</td>
</tr>
<tr>
<td>1.0</td>
<td>0.6</td>
<td>3.5</td>
<td>7</td>
<td>38</td>
<td>80</td>
</tr>
<tr>
<td>5.3</td>
<td>3.0</td>
<td>15.2</td>
<td>30</td>
<td>180</td>
<td>330</td>
</tr>
<tr>
<td>18.0</td>
<td>3.7</td>
<td>18.8</td>
<td>38</td>
<td>210</td>
<td>400</td>
</tr>
<tr>
<td>24.3</td>
<td>3.9</td>
<td>20.0</td>
<td>40</td>
<td>230</td>
<td>430</td>
</tr>
<tr>
<td>47.0</td>
<td>4.3</td>
<td>22.0</td>
<td>43</td>
<td>250</td>
<td>460</td>
</tr>
</tbody>
</table>

Pulse Duration in Microseconds

Table I. Pulse Duration vs. Resistance and Capacitance
needed is a critical value, it is best to measure the output pulse while trying various combinations of resistors and capacitors.

3. Delay Circuit

As was mentioned in the section Continuous Read Description, when the tape unit is reading, the data appears on the external line as a continuous stream of pulses coincident with a clock pulse. To clock the data into the Input Data Register of the General Purpose Interface, the data has to be on the input lines of the Data Register for at least 30 nanoseconds before the clock pulse comes up. This means that the clock pulse has to be delayed at least 30 nanoseconds. The clock pulse cannot be delayed too long because it must come up before the data on the input of the Data Register falls. The data remains on the data line for 20 microseconds.

The Delay Circuit (Figure 10) consists of a two-input NAND gate and resistor and capacitor. When the input pulse voltage goes high, the output pulse voltage will go low after the capacitor has charged up. Once the input pulse voltage goes low, the output pulse voltage will go high immediately. Therefore the leading edge of the input pulse is delayed on the output, but the trailing edge remains the same.
Figure 10. Delay Circuit and Signals
4. Open Collector Inverters

As was discussed in the subsection Clamping Circuit, there is a difference in voltage operation levels between the tape deck and the Nova computer. The Clamping Circuit succeeded in converting the tape deck signals to the operation level of the Nova computer. Some type of circuit is needed to convert the computer signals to the operation level of the tape deck. Among the 7400 series I.C.'s, there is a circuit which can be used to raise the voltage level of signals. This I.C. is the 7416 Open Collector Hex Inverter. When using the Open Collector Inverter, the user must furnish a pull-up resistor and supply voltage that will be connected to the output of the gate.

The input signals to the tape deck are required to have voltages from five to 15 volts. The Nova computer has voltage supplies of five and 15 volts. Since the use of the five volt supply at the output of the Open Collector Inverters would generate some signals with voltages below the necessary range, it was decided to use the 15 volt supply.

Pull-up resistors had to be chosen such that the rise and fall times of the pulses were less than two microseconds. Also the pull-up resistors had to be large enough so that the current from the supply-resistor combination was not greater than the breakdown current of 40 milliamps. Those signals that were normally at a low voltage required small pull-up resistors to decrease the rise time of the pulse.
Those signals that were normally at a high voltage required large pull-up resistors to decrease the fall time of the pulse.

D. I/O Control Networks

What follows are the three basic networks needed to interface the Nova Computer with the Kennedy 1400R magnetic tape unit. All of the interface networks are built from 7400 series TTL Integrated Circuit Logic. The only logic gates used in the design are NAND gates and Inverters.

1. Decode Network

There are certain signals which must be generated by the computer to provide control over the tape deck while it is in the Remote mode. Certain bits of the accumulator are assigned as the control word. Bit 13 of the accumulator is used to place the tape unit in either the Read or Write mode. A zero in bit 13 is used for Read, and a one in bit 13 is used for Write. The other control signals are decoded from bits 10 through 12 of the accumulator (Figure 11). The bit assignments for each control signal are shown in Table II.

As can be seen, no two control signals decoded from accumulator bits 10 through 12 can be output simultaneously. Since bit 13 is used for determining the Read-Write mode, any control signal which is output must also select either the Read or Write mode.
Figure 11. Decode Network
Table II. Bit Assignments for Control Signals

<table>
<thead>
<tr>
<th>Accumulator Bits</th>
<th>Control Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>10   11  12</td>
<td></td>
</tr>
<tr>
<td>0    0   1</td>
<td>Remote Rewind</td>
</tr>
<tr>
<td>0    1   0</td>
<td>End of Record</td>
</tr>
<tr>
<td>0    1   1</td>
<td>Remote Load Forward</td>
</tr>
<tr>
<td>1    0   0</td>
<td>Reverse Step</td>
</tr>
<tr>
<td>1    1   0</td>
<td>End of File</td>
</tr>
<tr>
<td>1    1   1</td>
<td>Remote Ready</td>
</tr>
</tbody>
</table>

In order to distinguish when bits 10 through 13 of the accumulator are used for control instead of data, a DATOA pulse from the CPU is used to gate the control words out to the tape deck. At any other time bits 10 through 13 of the accumulator are considered as data.

Four of the above mentioned control signals are required to be voltage levels. Those signals are Read, Write, Remote Load Forward, and Reverse Step. The rest of the control signals are pulses. Making the four control signals into levels is accomplished by setting a flip flop when the proper accumulator bits are one and the DATOA pulse is generated. The DATOA pulse serves as a clock input to a D-type flip flop. To drop the level control signals Reverse Step and Remote Load Forward, either another control signal must be output or the IO RESET pulse must be generated. Either the Read or Write control signal will always be output, but the IO RESET pulse sets the Read mode.
2. Read/Write Select Network

All of the control signals generated from the Decode Network are output directly to the tape deck except the Read and Write signals. With the Read-Write-Remote selector switch on the front panel of the tape deck, manual selection of the Read or Write mode may be made. When the selector switch is in the Remote mode, then the computer should have full control over the selection of the Read or Write mode.

Remote selection of the Read or Write mode is accomplished by connecting either the Remote Read Select line (RRSL) or the Remote Write Select line (RWSL) to the Selector Common line (SLCM). The Selector Common line has a 10 volt level signal when the Read-Write-Remote selector switch is in the Remote position. If the selector switch is in any other position the 10 volt level on the Selector Common line drops and the selector switch controls the mode of operation.

To make the connections of the Remote Read Select line or Remote Write Select line with the Selector Common line, a relay was used. To be compatible with the General Purpose Interface board, the relay had to be in a dual in-line package that could be placed into an I.C. socket and could be driven by five volts (Figure 12).

To control the switching of the relay, the Mag Tape Read (MTPR) signal from the Decode Circuit was used. The contact for the Remote Write Select line is normally closed, and the contact for the Remote Read Select line is normally open. When the Mag Tape Read control signal is output from
Figure 12. Read/Write Select Network
the Decode Circuit it causes current to flow through the coil which closes the Remote Read Select contact and opens the Remote Write Select contact. The diode is placed in the circuit so that when the output of the Inverter gate goes low the surge current will pass through the diode instead of the Inverter. This keeps the Inverter from being damaged.

3. Input/Output Timing Networks

During a Read operation the data appears on the external lines as a stream of pulses coincident with a clock pulse. To clock the data into the Input Data Register, the leading edge of the clock pulse must be delayed at least 30 nanoseconds. Therefore the clock pulse is input to the Delay Circuit (Figure 13). The output of the Delay Circuit could be used to clock the data into the Data Register. Instead the signal is passed through a monostable multivibrator to decrease the duration of the pulse. The output signal of the monostable is used to clock the data into the Data Register. Also this signal is used as a device completion signal to set the DONE flip flop of the General Purpose Interface.

During a Write operation the data is written on tape upon the receipt of a Write-Step pulse. Both the data and the Write-Step pulse are controlled by the DATOB CPU pulse. The DATOB pulse clocks the data in the accumulator into the Output Data Register. Once the data has entered the register it appears on the data lines as voltage levels. Also
Figure 13. Input Timing Circuit and Signals
the DATOB pulse is passed through a monostable to increase the pulse duration needed for the Write-Step pulse (WSTP). The Write-Step pulse is sent directly to the tape deck and the data is written on tape.

Approximately 60 microseconds after the leading edge of the Write-Step pulse, the tape deck generates a parity check signal. This signal originates from a Parity Check network within the tape deck. Upon writing the data, the normal parity bit is generated and written in track seven. The Parity Check circuit generates its own parity bit from the data and compares it with the parity bit written in track seven. If the two parity bits agree, no error occurred and no signal is generated. If the two parity bits disagree, a writing error has occurred and signal is sent to the computer. The signal is a pulse of 15 microseconds duration. To hold the signal, it is clocked into a flip flop. The output of the flip flop is used as a status signal.

In order to secure the proper timing, the Write-Step pulse (WSTP) is passed through a monostable to produce a delay pulse (DLY1) of such duration that the combined time between the leading edge of the Write-Step pulse and the trailing edge of the delay pulse is between 60 and 75 microseconds (Figures 14 and 15). The delay pulse (DLY1) is passed through a monostable to produce a short duration pulse (EPCP) to clock the Parity Check pulse into the flip flop.
Figure 14. Output Timing Circuit
Figure 15. Output Timing Signals
Another timing consideration is the duration between successive Write-Step pulses. The speed of the tape deck while in the Write mode is 300 characters per second maximum. This means that a delay of 3.3 milliseconds must occur between successive Write-Step pulses. To accomplish this the first delay pulse (DLY1) is inverted and passed through a monostable. By using the inverted delay signal (DLY1) on the input of the monostable, the output delay pulse (DLY2) will trigger correspondingly to the trailing edge of the Write-Step pulse.

The output delay pulse (DLY2) is passed through a monostable to produce another delay pulse (DLY3). The two monostables are set such that the combined delay of the two pulses (DLY2 and DLY3) is four milliseconds. Two monostables were needed instead of one because a four millisecond pulse could not be obtained from any resistor-capacitor combination with the DLY1 signal on the input. The DLY1 signal did not allow enough time for the capacitor in the monostable to recover.

The DLY3 signal is then passed through a monostable to produce a short duration pulse (STDN) used as the device completion signal to set the DONE flip flop of the General Purpose Interface. Once DONE is set another Write-Step pulse may be output.

Only the major portions of the hardware interface have been presented here. To get a more complete and detailed
description of the entire interface design and the external connections needed, see the maintenance manual on the inter-
face design [6].
IV. SOFTWARE

A. Basic I/O Instructions

After completion of the hardware design, it was necessary to develop the system software so that the hardware could be used. All of the signals used in the hardware design are controlled by signals which originate in the CPU of the computer. These signals are generated by the CPU upon command of the programmer. These commands are in the form of instructions which the computer recognizes.

The instructions in question are the input-output type. All words in the computer consist of 16 bits numbered zero to 15 from left to right. For the I/O instructions the last six bits (bits 10-15) are used for the device code. For the magnetic tape unit the device code is 22 in base eight. Since the computer has four accumulators, bits three and four of the I/O instruction are used to specify the accumulator that will be used for the operation. Bits five, six, and seven are used to designate the specific I/O instruction. These instructions are NIO, DIA, DOA, DIB, DOB, DIC, DOC, and SKP. Bits eight and nine when used with all the instructions except SKP specify the START, CLEAR, and IO PULSE signals. When used with the SKP instruction bits eight and nine test the state of the BUSY or DONE flip flops. Uses of the I/O instructions are demonstrated in the next few paragraphs and in Appendix B.
As was mentioned in the subsection Decode Network, accumulator bits 10 through 13 are assigned as a control word. The bit assignments for each control signal can be found in that subsection. To generate the proper control signal, the programmer must load the proper bits of the accumulator and then execute a DOA instruction.

There is also a status word which is assigned accumulator bits zero through seven. Each bit is assigned a status signal from the tape deck. The bit assignments for each status signal are shown below in Table III. In order to bring the status signals into the Data Register, an IO PULSE signal is needed. After the status is in the Data Register a DIA instruction is needed to bring the status into the accumulator.

<table>
<thead>
<tr>
<th>Accumulator Bits</th>
<th>Status Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Gap Detect</td>
</tr>
<tr>
<td>1</td>
<td>Echo Parity</td>
</tr>
<tr>
<td>2</td>
<td>Ready</td>
</tr>
<tr>
<td>3</td>
<td>Remote Status</td>
</tr>
<tr>
<td>4</td>
<td>Gap in Process</td>
</tr>
<tr>
<td>5</td>
<td>File Protect</td>
</tr>
<tr>
<td>6</td>
<td>End of Tape</td>
</tr>
<tr>
<td>7</td>
<td>Load Point</td>
</tr>
</tbody>
</table>

Table III. Bit Assignments for Status Signals

To output data to the tape deck a DOB instruction must be used. To sense when the data has been written on tape,
the DONE flip flop should be checked. The next DOB instruction should not be executed until DONE is nonzero.

To bring data into the computer, the DONE flip flop must be checked to be nonzero. This state occurs when the data has been brought into the Data Register. After DONE is found to be nonzero, a DIB instruction should be executed to bring the data into the accumulator.

The instruction sequences needed for executing any input or output (whether data, control or status) between the magnetic tape unit and the Nova computer are included in the examples in Appendix B.

B. Program Techniques

In this section the basic operation of the read and write routines is discussed. Along with the discussion, some subroutines contained in Appendix B are referenced. For a complete listing of the Mag Tape Write and Mag Tape Read routines see the interface users manual [6].

The main objective for building and designing the hardware interface was so that the magnetic tape unit could be used to write a library of programs on tape and to read programs from the library when needed. Software was developed that would write programs on magnetic tape in a certain format and use that format for reading programs from the tape. The format in which a program is written on tape is shown in flow table form in Figure 16.
Figure 16. Flow Chart Showing Format of Data on Tape
The basic idea behind the write routine is to dump the contents of memory between two locations onto the magnetic tape. Since the tape unit can only write six bits at a time, a word in memory is written as three characters on tape. A subroutine to write a word on tape is shown in Appendix B, example one. Included in the example is the subroutine to bring in status.

It should be noticed in the subroutine that each time a character is written, the status of the tape unit is checked. This is to see if an error occurred in writing the character. If an error is detected, the program halts. Notice also in the subroutine that a checksum is computed as each character is written. This checksum is written at the end of the record. The purpose of the checksum is for detecting reading errors.

The process of converting three characters into a word while reading is a little more difficult than writing the word into three characters. Example two in Appendix B is a subroutine to format three characters into a word. Included in the example is a subroutine to read a character and a subroutine to compute the checksum.

The complete data cannot be written continuously because, at intervals, a gap must be inserted to allow the tape transport to stop. The write routine is set up to write 200 words in a record. Each record is separated by a record gap. Example three in Appendix B is a subroutine used to write gaps. A delay is provided before and after
the gap is written to allow the capstan to stop completely. Status is checked during the writing of the gap to detect when the gap stops.

The characters 77 and 66 are written after each gap as control characters. When these characters are read from the tape, the read routine knows that the characters that follow are useful information. With the use of these control characters, the LCC written during the record gap does not affect the read operation. Two control characters are used because there is a possibility that the LCC written in the record gap might be the same as a control character. If this occurred while one control character was being used, all useful information would be phased together by one six-bit character. Example four in Appendix B is a subroutine to detect the control characters. Example five in Appendix B is a subroutine to write a control character.

The first useful information that is written on tape is the name of the program. The name of a program is specified in a word or 16 bits. The first part of the word (bits zero to seven) is the assigned tape number. The second part of the word (bits eight to 15) is the number assigned to that program. For example, the first program on tape 5000 would probably have the name 5001. The second program on the same tape would probably have the name 5002. During a read operation, the name of the desired program is specified by the user. Only that program will be stored into memory.
The two words written before each record, the memory address of the first data word in the record and the number of data words contained in the record, are used by the read routine to store the data which is read into the proper memory location. Also, knowing the number of words in the record allows the read routine to recognize when the checksum has been read. It then compares the checksum that was read to the one that was computed by the routine as each character was read. If the two checksums compare, no reading error occurred. If the two checksums are not equal the program is halted.

After all the data has been written, a record gap and the two control characters 77 and 66 are written. The starting address of the program is then written followed by an end-of-file control character 177777. During the read routine when the end-of-file control character is read, the tape rewinds and then loads forward back to the load point or beginning-of-tape marker. Example six in Appendix B is a subroutine to rewind the tape and load forward to the load point. In the subroutine, a delay is used to guarantee that the tape will completely rewind before the load forward signal is transmitted to the tape unit.

During the read routine, the starting address of the program that was read is stored in a specific location. When the read routine halts, pressing the Continue toggle switch on the computer console causes execution of the program which was just read off tape.
V. CONCLUSION

The hardware interface that was built has very few limitations that would hinder the complete utilization of all the functions available on the magnetic tape unit. The software interface that was written is the limiting factor. A much more complex software package could be written. For example, the hardware interface provides the circuitry to drive the tape in reverse while writing. This could be used to correct a writing error. Software could be written that would read data from the magnetic tape unit and punch it out on paper tape. Almost anything practical can be done with the magnetic tape unit if the software is provided.
BIBLIOGRAPHY


VITA

Thomas Dean Steury was born on November 23, 1948 in Kansas City, Missouri and received his primary education from there. He received his secondary education in Raytown, Missouri. He received his college education from Metropolitan Junior College in Kansas City, Missouri and the University of Missouri-Rolla in Rolla, Missouri, where he received his Bachelor of Science degree in Electrical Engineering in August 1970.

He has been enrolled in the Graduate School of the University of Missouri-Rolla since September 1970 but missed the spring semester 1971 because of active duty in the Armed Forces. He has been a Graduate Assistant in the Electrical Engineering Department during his enrollment as a graduate student.
## APPENDIX A

<table>
<thead>
<tr>
<th>Signal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDLY</td>
<td>Clock Delay</td>
</tr>
<tr>
<td>CLR</td>
<td>Clear</td>
</tr>
<tr>
<td>DATA0 to DATA15</td>
<td>Data</td>
</tr>
<tr>
<td>DIA or DATIA</td>
<td>Data In A Buffer</td>
</tr>
<tr>
<td>DOA or DATOA</td>
<td>Data Out A Buffer</td>
</tr>
<tr>
<td>DIB or DATIB</td>
<td>Data In B Buffer</td>
</tr>
<tr>
<td>DOB or DATOB</td>
<td>Data Out B Buffer</td>
</tr>
<tr>
<td>DIC or DATIC</td>
<td>Data In C Buffer</td>
</tr>
<tr>
<td>DOC or DATOC</td>
<td>Data Out C Buffer</td>
</tr>
<tr>
<td>DLY1</td>
<td>Delay One</td>
</tr>
<tr>
<td>DLY2</td>
<td>Delay Two</td>
</tr>
<tr>
<td>DLY3</td>
<td>Delay Three</td>
</tr>
<tr>
<td>DS0 to DS5</td>
<td>Device Selection</td>
</tr>
<tr>
<td>ECPL</td>
<td>External Clock Pulse</td>
</tr>
<tr>
<td>EOF</td>
<td>End of File</td>
</tr>
<tr>
<td>EOR</td>
<td>End of Record</td>
</tr>
<tr>
<td>EPCP</td>
<td>Echo Parity Check Pulse</td>
</tr>
<tr>
<td>EXCL</td>
<td>External Clock</td>
</tr>
<tr>
<td>IC</td>
<td>Integrated Circuit</td>
</tr>
<tr>
<td>INTA</td>
<td>Interrupt Acknowledge</td>
</tr>
<tr>
<td>INTP</td>
<td>Interrupt Priority</td>
</tr>
<tr>
<td>INTR</td>
<td>Interrupt Request</td>
</tr>
</tbody>
</table>

Table IV. Signal Abbreviation List
<table>
<thead>
<tr>
<th>Signal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOPLS</td>
<td>IO Pulse</td>
</tr>
<tr>
<td>IORST</td>
<td>IO Reset</td>
</tr>
<tr>
<td>MSKO</td>
<td>Mask Out</td>
</tr>
<tr>
<td>MTPR</td>
<td>Mag Tape Read</td>
</tr>
<tr>
<td>MTPW</td>
<td>Mag Tape Write</td>
</tr>
<tr>
<td>NIO</td>
<td>No IO</td>
</tr>
<tr>
<td>RLDLP</td>
<td>Remote Load Forward</td>
</tr>
<tr>
<td>RQENB</td>
<td>Request Enable</td>
</tr>
<tr>
<td>RRDY</td>
<td>Remote Ready</td>
</tr>
<tr>
<td>RRSL</td>
<td>Remote Read Select</td>
</tr>
<tr>
<td>RRWD</td>
<td>Remote Rewind</td>
</tr>
<tr>
<td>RVSTP</td>
<td>Reverse Step</td>
</tr>
<tr>
<td>RWSL</td>
<td>Remote Write Select</td>
</tr>
<tr>
<td>SELB</td>
<td>Selected Busy</td>
</tr>
<tr>
<td>SELD</td>
<td>Selected Done</td>
</tr>
<tr>
<td>SKP</td>
<td>Skip</td>
</tr>
<tr>
<td>SLCM</td>
<td>Selector Common</td>
</tr>
<tr>
<td>STDN</td>
<td>Set Done</td>
</tr>
<tr>
<td>STRT</td>
<td>Start</td>
</tr>
<tr>
<td>TTL</td>
<td>Transistor Transistor Logic</td>
</tr>
</tbody>
</table>

Table IV. Signal Abbreviation List (continued)
APPENDIX B

Software Examples

Example 1. Assume that a 16-bit word has been loaded into accumulator zero.

; WRTWD: WRITES THREE CHARACTERS OR A WORD ON TAPE IN THE ORDER
; BITS 10-15 FIRST, BITS 4-9 SECOND, BITS 0-3 THIRD

WRTWD: STA 3, RTRN
DOBS 0, 22
SKPDN 22
JMP .-1
JSR DELAY
JSR STAT
LDA 2, STAT2 ; 041000
AND 2, 1, SZR
HALT ; PARITY ERROR OR END OF TAPE
MOV 0, 3
LDA 2, EORM ; 77
AND 2, 3 ; KEEP SIX DATA BITS
LDA 1, CHECK
ADD 3, 1 ; COMPUTE CHECKSUM
AND 2, 1
STA 1, CHECK
MOVZR 0, 0
DSZ CW1 ; SIX SHIFTS FOR A NEW CHARACTER
JMP .-2
LDA 2,ICW1 ;RESET COUNT 1
STA 2,CW1
DSZ CW2 ;THREE CHARACTERS PER WORD
JMP WRTWD+1
LDA 2,ICW2 ;RESET COUNT 2
STA 2,CW2
JMP@ RTRN

DELAY: DSZ DLY
JMP .-1
LDA 2,IDLY ;20
STA 2,DLY ;RESET DLY
JMP 0,3

;STAT - BRINGS IN STATUS OF TAPE DECK
STAT: NIOP 22
DIA 1,22
JMP 0,3

RTRN: 0
STAT2: 041000
EORM: 77
CHECK: 0
CW1: 6
ICW1: 6
CW2: 3
ICW2: 3
DLY: 20
IDLY: 20
Example 2.

;RDFMT - FORMATS THREE CHARACTERS INTO A WORD

RDFMT:  STA 3, RTRN1
JSR RDCH
JSR CCHS
MOV 0,1 ; AC BITS 10-15
JSR RDCH
JSR CCHS
ADDZL 0,0 ; SHIFT SIX PLACES LEFT
ADDZL 0,0 ; TO FORM
ADDZL 0,0 ; AC BITS 4-9
ADD 0,1 ; 12 BITS FORMED
JSR RDCH
JSR CCHS
MOVS 0,0 ; SWAP HALVES OF AC
ADDZL 0,0 ; SHIFT FOUR PLACES LEFT
ADDZL 0,0 ; TO FORM AC BITS 0-3
ADD 0,1 ; FORM 16 BIT WORD
JMP@ RTRN1

;RDCH - READS A CHARACTER FROM A TAPE

RDCH:  NIOS 22
SKPDN 22 ; WAIT FOR DATA
JMP .-1
DIB 0,22 ; STROBE IN DATA
LDA 2, DBR ; 77
AND 2,0 ; KEEP AC BITS 10-15
JMP 0,3
;CCHS - COMPUTES THE CHECKSUM

CCHS: STA 3,RTRN2
LDA 3,CHECK
ADD 0,3
AND 2,3
STA 3,CHECK
JMP@ RTRN2

RTRN1: 0
DBR: 77
RTRN2: 0
CHECK: 0

Example 3. Assume accumulator one has been loaded with either 24 for record gap or 64 for file gap.

;GAP - WRITES A GAP ON TAPE

GAP: STA 3,RTRN
JSR DELAY
DOA 1,22
JSR STAT
LDA 0,RBOUT ;177400
AND 0,1 ;KEEP ONLY STATUS BITS
LDA 2,STAT3 ;130000
SUB# 2,1,SZR ;WAIT FOR GAP TO STOP
JMP -.5
JSR DELAY
JMP@ RTRN

;DELAY - CAUSES A DELAY SO CAPSTAN WILL STOP
DELAY:    LDA 2,ICW2 ;400
STA 2,CW4 ;RESET COUNT 4
DSZ CW4
JMP .-1
DSZ CW5
JMP .-5
LDA 2,ICW5 ;1000
STA 2,CW5 ;RESET COUNT 5
JMP 0,3
STAT:    NIOP 22
DIA 1,22
JMP 0,3
RTRN:    0
RBOUT: 177400
STAT3: 130000
ICW4: 400
CW4: 400
ICW5: 1000
CW5: 1000

Example 4.

;BORMK - IDENTIFIES THE MARKS AT BEGINNING OF RECORD
BORMK:    STA 3,RTRN1
JSR RDCH
SUB# 0,2,SNR ;CHARACTER IS 77?
JMP .+11 ;YES
MOV 0,0,SZR ;CHARACTER IS 00?
JMP .+3 ;NO
ISZ LPF ;YES
JMP .+5
DSZ BID ;4
JMP BORMK+1
IORST
HALT ;NO MARK FOUND
JSR RDCH
LDA 2,SBR ;66
SUB# 0,2,SNR ;IS CHARACTER 66?
JMP@ RTRN1 ;YES
DSZ BID ;NO
JMP -.5
IORST
HALT ;NO MARK FOUND
RDCH: NIOS 22
SKPDN 22 ;WAIT FOR DATA
JMP -.1
DIB 0,22 ;STROBE IN DATA
LDA 2,DBR ;77
AND 2,0 ;KEEP AC BITS 10-15
JMP 0,3
RTRN1: 0
LPF: 0
BID: 4
SBR: 66
DBR: 77
Example 5. Assume accumulator one has been loaded with either 77 or 66.

;SPCH - WRITES A SPECIAL CHARACTER ON TAPE

SPCH: DOBS 1,22  
      SKPDN 22  
      JMP .-1  
      STA 3,RTRN  
      JSR DELAY  
      JSR STAT  
      LDA 2,STAT2 ;041000  
      AND 2,1,SZR  
      HALT ;PARITY ERROR OR END OF TAPE  
      JMP@ RTRN

DELAY: DSZ DLY  
      JMP .-1  
      LDA 2,IDLY ;20  
      STA 2,DLY ;RESET DLY  
      JMP 0,3

STAT: NIOP 22  
      DIA 1,22  
      JMP 0,3

RTRN: 0

STAT2: 041000

DLY: 20

IDLY: 20
Example 6.

;RWND - REWINDS TAPE LOADS FORWARD TO LOAD POINT

RWND:  LDA 1,RWD ;10 FOR REWIND
       DOA 1,22
       LDA 2,ICW1 ;RESET COUNT 1
       STA 2,CW1
       DSZ CW1 ;7000
       JMP ..-1
       DSZ CW2 ;1000
       JMP ..-5
       LDA 2,ICW2 ;RESET COUNT 2
       STA 2,CW2

LFLP:  LDA 1,LDF ;30 FOR LOAD FORWARD
       DOA 1,22
       IORST
       JMP 0,3

RWD:  10
ICW1:  7000
CW1:  7000
ICW2:  1000
CW2:  1000
LDF:  30