AN AUTOMATED LABORATORY TEST SYSTEM

by

NEUFITO L. FERNANDES

B.S., Indian Institute of Technology - Bombay, 1979

A MASTER'S REPORT

Submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1983

Approved by

M S P Lucas

Major Professor

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

A conventional electronic laboratory bench typically consists of several instruments and accessories such as power supplies, voltmeters, signal generators, oscilloscopes, resistance substitution boxes, capacitance substitution boxes etcetera. And a typical laboratory experiment involves many manual operations like making/breaking interconnections, varying input signals, recording input/output signals, processing and analysing the recorded data and so forth. These operations are extremely time consuming and highly prone to human errors. Besides, once the principles of the experiment and the test set-up are understood, very little useful experience results from manually collecting and processing the data. It is therefore desirable to have an automated test bench that can perform most of the functions of the manual test bench but under program control. Towards realizing this objective a relatively simple and inexpensive Automated Test System has been developed for student use in the Electrical Engineering laboratory at Kansas State University.

This report provides brief descriptions of the major components and various hardware modules made available on the system, and their programming procedure. A block schematic of the system is shown in fig. 1.1. The key component in the system is the HP Model 6940B Multiprogrammer which is a special purpose mainframe that provides power supplies, interconnections, address and control logic, and phy-

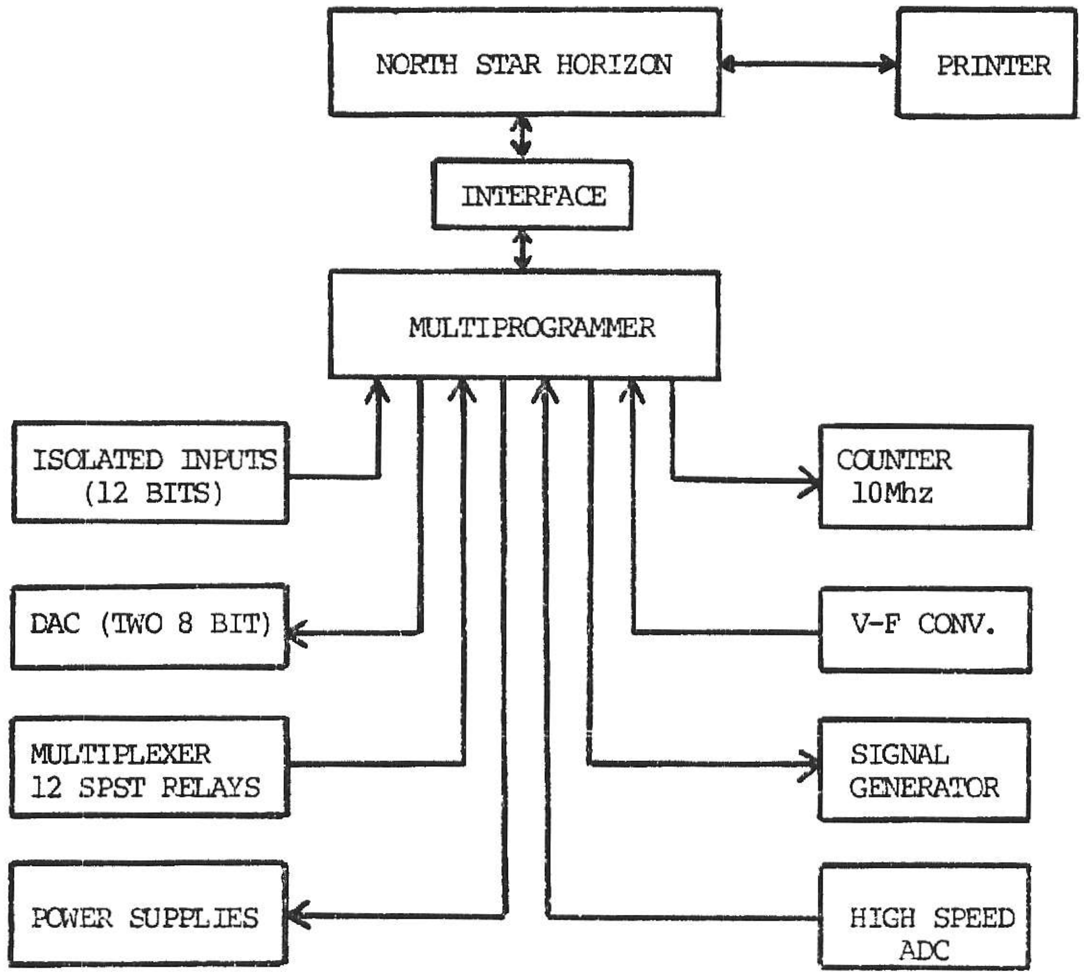


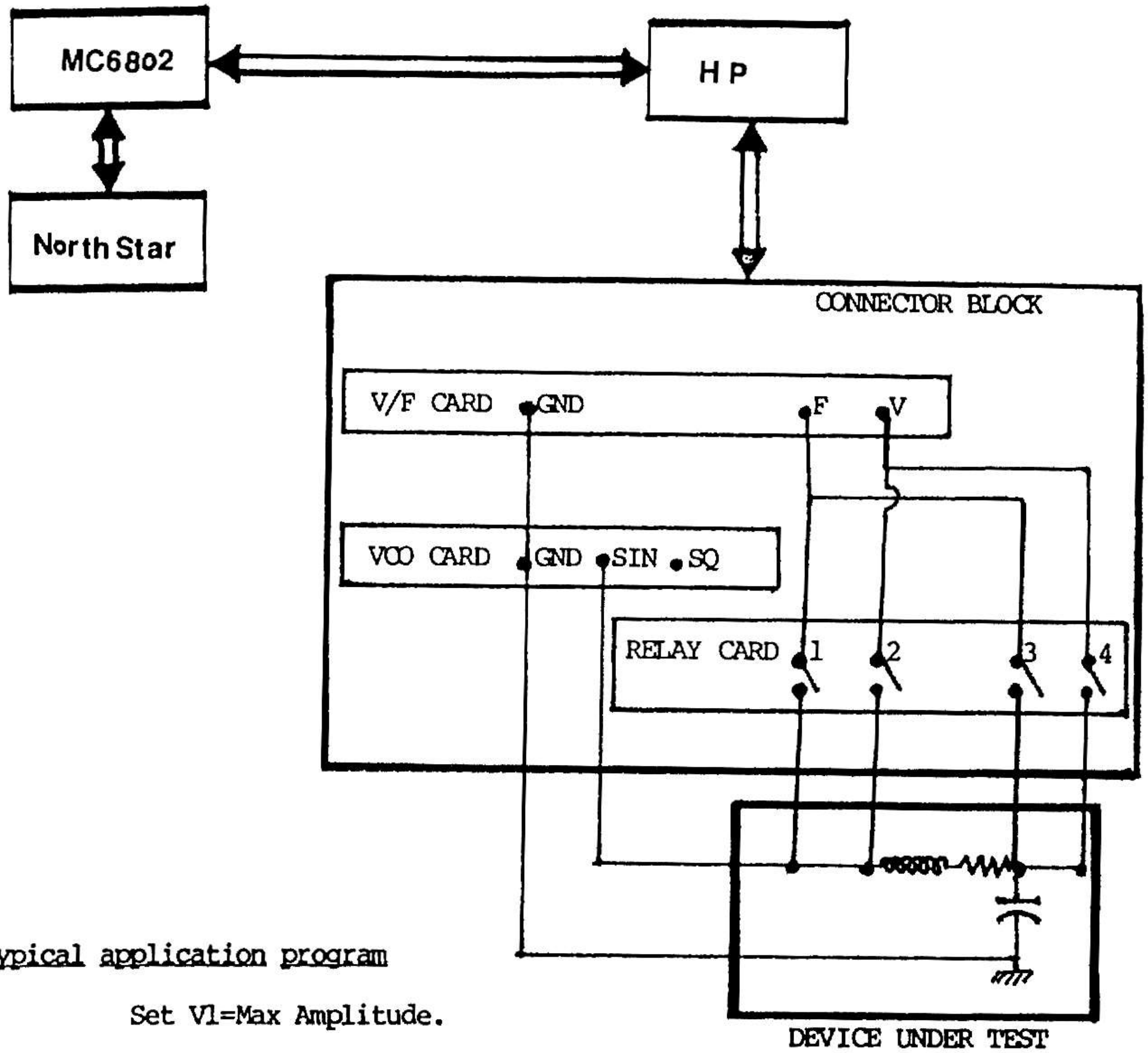
fig. 1.1 Block schematic of the automated test system

sical locations for a variety of input/output accessory modules. The multiprogrammer is program controlled from the Microcomputer - North Star Horizon II (using North Star BASIC), via a MC6802 based interface. This interface serves to provide a conversion of ASCII coded serial data (from the North Star) to Binary coded parallel data (required by the Multiprogrammer), and vice versa. In addition, it also performs logic level translation from RS232 (at the Horizon end) to TTL/DTL logic (at the Multiprogrammer). The Multiprogrammer and the MC6802 based interface are discussed in Chapter II and Chapter III respectively. Two of the accessory cards (the V/F card, and the VCO card) were designed and implemented by students. The hardware and the programming for the V/F card are described in chapter IV and the VCO card in chapter V. Three factory built cards; the Relay card, The High Speed Analog to Digital Conversion card, and the Isolated digital input card, have also been interfaced with the multiprogrammer system. A brief description and the programming procedure for these cards is given in chapters VI, VII, and VIII respectively. An Automated Diagnostic feature is provided using a simple plug-in module. This will ensure proper functioning of the various accessory cards, and is recommended before starting any lengthy experiment. This feature is discussed in chapter IX. System operation is outlined in Chapter X. A typical application of the system and the program logic required to control it is shown in fig. 1.2. A plot of the frequency response for this application is given in fig. 1.3.

An intelligent use of this system will eliminate a lot of hardwork and drudgery associated with the manual collection and processing of data during an experiment. An attempt has been made to provide simple and user friendly commands for the various accessory cards available on the system. The scope of using the system for more interesting and innovative experiments is unlimited. It is hoped that this system will make experimentation a more interesting and enjoyable experience for the students.

THIS BOOK CONTAINS NUMEROUS PAGES WITH DIAGRAMS THAT ARE CROOKED COMPARED TO THE REST OF THE INFORMATION ON THE PAGE.

THIS IS AS RECEIVED FROM CUSTOMER.



A typical application program

Set V1=Max Amplitude.

Set F1= 10 hz.

LOOP OUTPUT Waveform of V1 volts, F1 hz.

CLOSE Relays 1 and 2, OPEN Relays 3 and 4.

READ V_{in}, F_{in} and Store these values.

MEASURE V_{out}, F_{out} and Store these values.

Increment F1 by 100.

If F1 is less than 1 Mhz then go back to LOOP.

PLOT frequency response.

fig. 1.2 A typical application of the system and the program logic.

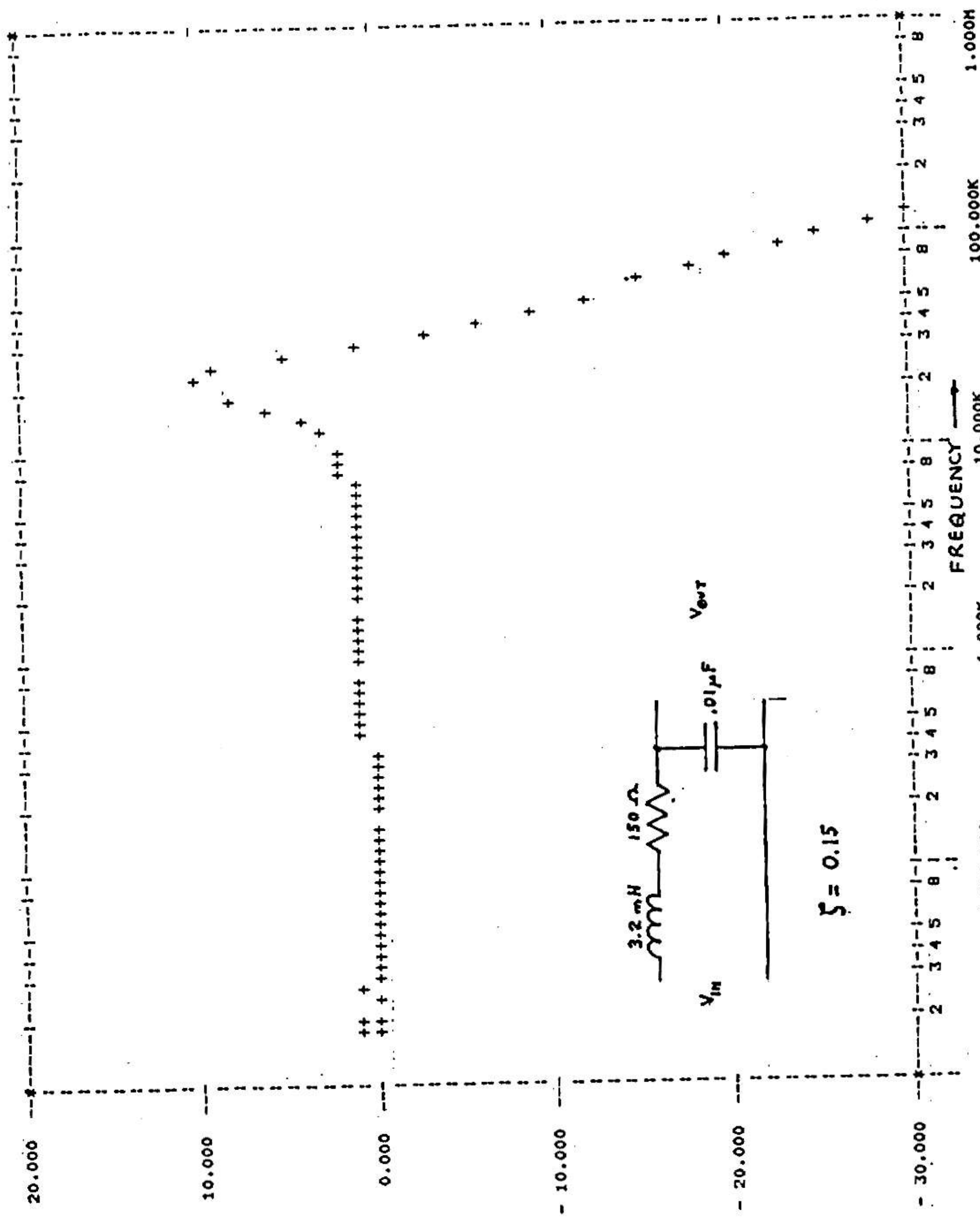


fig. 1.3 Frequency response of the system in fig. 1.2.

CHAPTER II

HP MULTIPROGRAMMER

The 6940B multiprogrammer is a master control unit for bidirectional (output data distribution/input data multiplexing) systems [2]. It can be used in a single unit system or multi-unit system consisting of one 6940B master unit and up to fifteen 6941B extender units. The configuration used in this project is a single unit system which can hold from one to fifteen plug-in input/output user accessory cards.

The function of the output cards is to develop an output quantity proportional to programmed data and to deliver this quantity to the user's system. The output cards are similar to one another in that each contains address gate, data storage and output address conversion circuits. The nature of the output conversion circuits determines the card type. An input card receives data from the user's system and makes it available to the computer. Any type of accessory card may be used in any of the 15 card slots, but when a card is assigned to a particular slot it assumes the address of that slot.

The system can be controlled by a digital computer via a 16 bit parallel binary word. The various communication protocols are provided by the following four types of words.

1. CONTROL WORD.

This word establishes the mode of operation of the system (input-/output, Timing/handshake etc). It is identified by all 1's in the four most significant bits and the word format is as shown in fig. 2.1.

The decimal equivalent of the 4 least significant bits represent the unit number of the unit being selected. In the case of a single unit system it is 00 (ie the 6940B master control unit). Bits 4 to 8 are mode control signals (TME, SYS, DTE, ISL, and IEN) which select one or more control modes (SYS, DTE, or TME for data output transmission, or ISL, IEN, or TME for data input reception). A brief explanation of the control mode signals is given in Appendix 8.

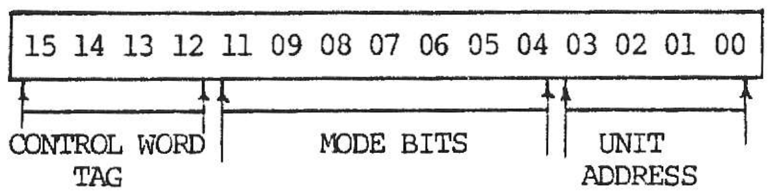


fig. 2.1 Control word format.

2. OUTPUT DATA WORD.

This word identifies the slot specified by the decimal equivalent of the 4 most significant bits and directs the data in the remaining 12 bits to the card housed in this slot. The output card then develops an output quantity proportional to the programmed data and delivers this quantity to the user. This could function as logic levels, or contact closures, or D/A converter output etc, depending on the type of the output card. Prior to sending a data word the system must be programmed for output mode by the control word. The data word format is shown below in fig. 2.2.

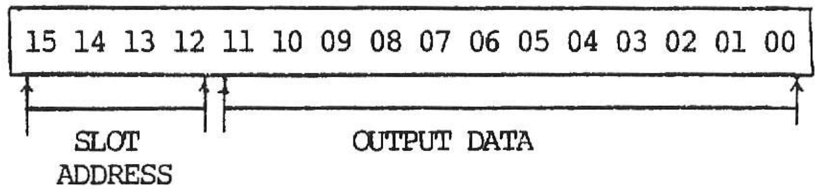


fig. 2.2 Data Word Format.

3. INPUT ADDRESS WORD.

This word selects a card that is to send data (a return data word) back to the computer. The slot address of the input card is specified by the 4 most significant bits as shown in fig. 2.3. The remaining 12 bits are not used. The selected input card makes the corresponding user data available to the computer on bits 0-11 of the multiprogrammer return data lines. Before programming an address word the multiprogrammer must be placed in the input mode and the corresponding unit selected by the previous control word.

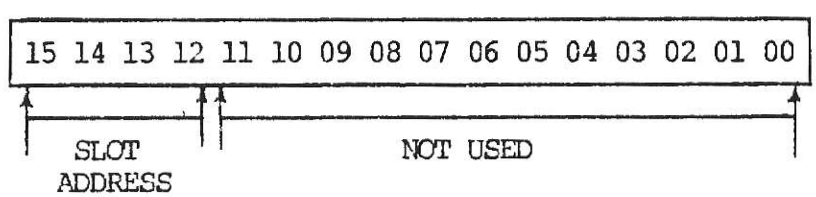


fig. 2.3 Address Word Format.

4. RETURN DATA WORD

This word returns information from an input card previously selected by an address word. The most significant bit of this word is the interrupt request bit. A "1" in this bit indicates that the card has returned a flag and its input data is valid or that this card has generated an interrupt. A "0" indicates that the card is still busy or has interrupted and its data is not valid. Bits 12 through 14 are not used and bits 0 through 11 contain the return data. If the input mode is not selected then these bits are an echo back of bits 0 to 11 received from the computer and can be used to check for validity of data transmitted to the multiprogrammer.

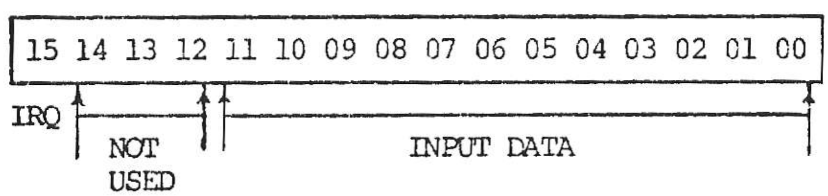


fig. 2.4 Return Data Word Format.

The control words for various modes of operation are given in fig. 2.5. The timing diagram for the HP Multiprogrammer I/O is shown in Appendix 7. The plug-in input/output accessory cards designed and implemented inhouse, as well as the factory built cards made available on the system, are discussed in chapters IV to VIII. A brief functional description and the programming sequence associated with these cards is provided with sample test programs.

Multiprogrammer Operating modes		CONTROL WORD (in HEX)
OUTPUT MODE	INPUT MODE	
SYE off		F000
SYE on		F020
DTE, SYE on		F060
DTE, SYE, TME on		F070
	ISL, SYE on	F0A0
	ISL, SYE, TME on	F0B0
	IEN, SYE, TME on	F130

fig. 2.5 Control words for various operating modes.

CHAPTER III

THE MC6802 BASED INTERFACE

3.1. HARDWARE DESCRIPTION

The heart of the system is the MC6802 microprocessor with on-board clock and 128 bytes of on-board RAM starting at location 0000H. Additional 1K memory is provided beginning at location 6000H. The monitor program, which decodes and executes user commands to control communications between the North Star and the HP Multiprogrammer, is stored in a 2716 EPROM (starting at location E000). A block diagram of the system is shown in fig. 3.1 and the memory map of the various modules in the system is illustrated in fig. 3.2. The schematic diagram of the system is shown in Appendix 4 [3].

A 3.6864 Mhz crystal provides the time base for the internal clock of the MPU. A divide-by-four circuit, internal to the MPU, gives an effective clock rate of 920.8 Khz. The RS232 to TTL conversion is performed by the 1488 and 1489 (sender/receiver) quad line driver. Parallel communication with the HP Multiprogrammer is handled by the 2 PIAs (PIA1 and PIA2). The serial to parallel conversion is accomplished by the MC6850 ACIA. The transmit and receive clock signals are generated by the MC6840 programmable timer module (PTM). Switches S1 and S2 (refer schematic diagram in Appendix 4) provide a choice of four different baud rates for the MC6840 PTM. These switches are read every time the system is switched on or RESET, and the baud rate is decided by the corresponding value in the baud-rate-table stored at the start

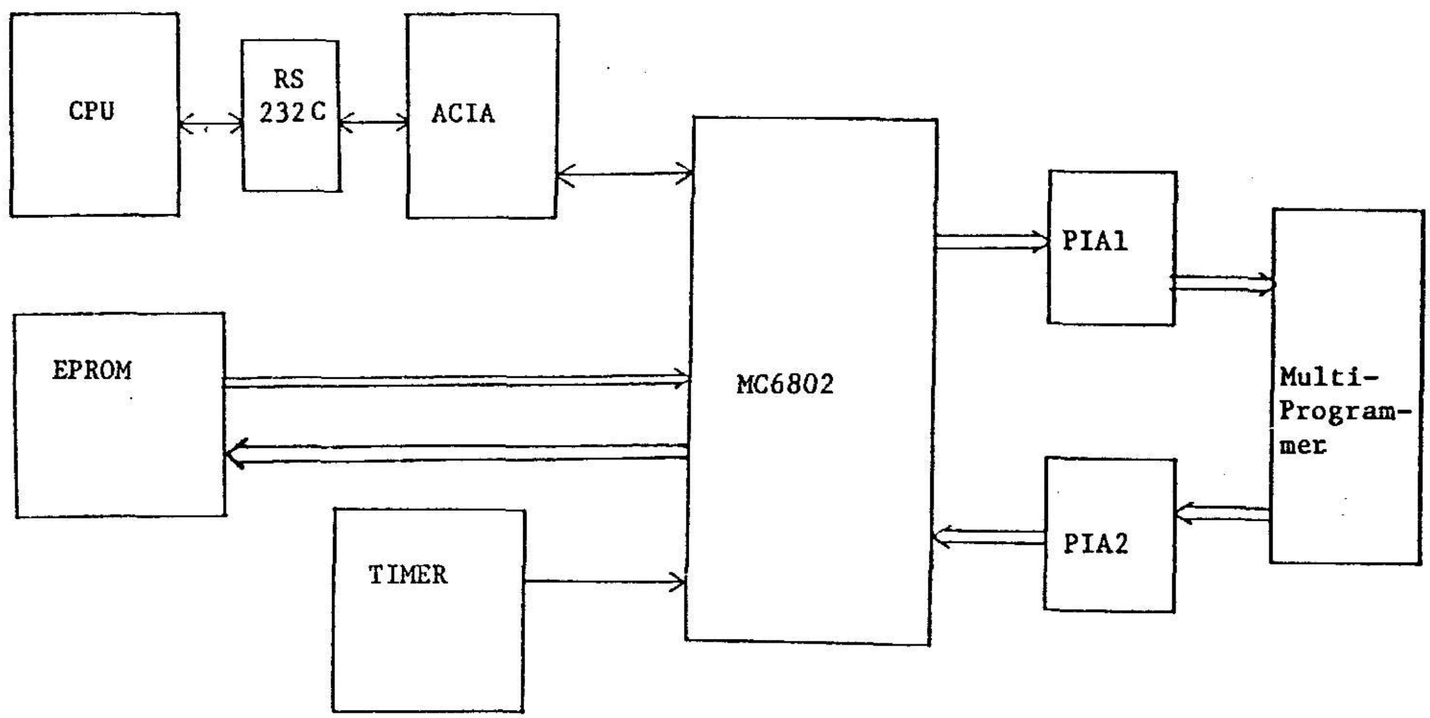


fig 3.1

INTERFACE USING THE MC6802 MICROPROCESSOR

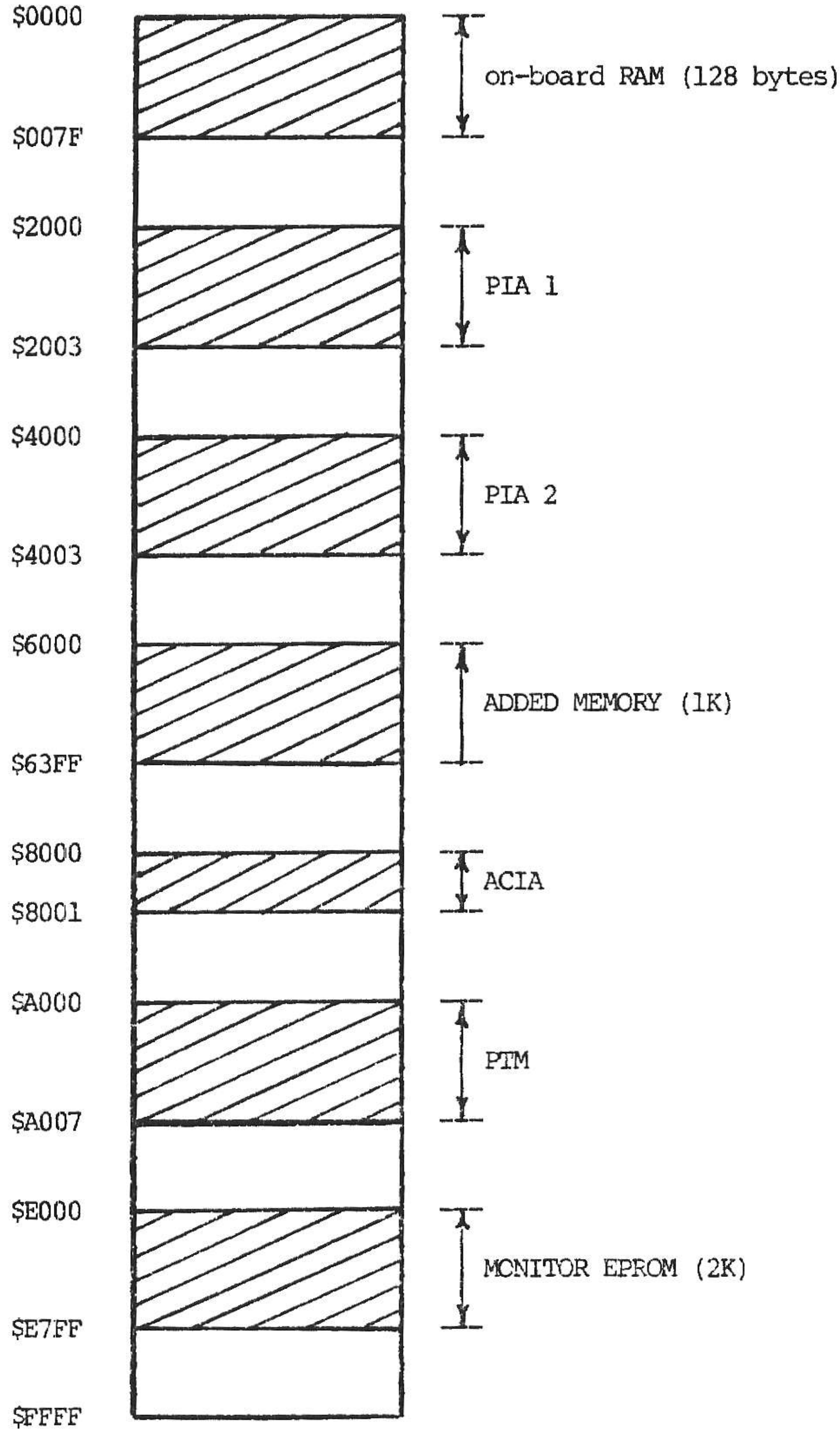


fig. 3.2 Memory map of the MC6802 system.

of the EPROM. The table content for different baud rates is shown in fig. 3.3.

S1	S2	Address in Memory	Contents	Baud rate
0	0	E000, E001	00, 5F	110
0	1	E002, E003	00, 17	300
1	0	E004, E005	00, 05	4800
1	1	E006, E007	00, 02	9600

fig. 3.3 Table of baud rates

For any other baud rate the value N to be stored in memory can be calculated as follows.

$$N = \frac{\text{System Clock Frequency}}{2 \times (\text{desired baud rate})} - 1$$

N should be converted to Hexadecimal before storing it in memory.

The above system hardware provides the bidirectional, asynchronous, serial to parallel data conversion appropriate for use with an HP 6940B Multiprogrammer. It can accept serial RS232 data from a modem or a computer and convert it to parallel DTL/TTL compatible logic signals.

The ASCII to Binary and Binary to ASCII conversions are handled by the software described in the following section.

3.2. SOFTWARE DESCRIPTION

The monitor program, written in MC6802 Assembly language, is stored in the 2716 EPROM. It consists of a main driver routine which decodes the command symbols (&, ?, #, %, !, A) from the user and calls the appropriate routine to perform the required task for the user. Flow-charts and detailed description of every routine is given in Appendix 1. The program listing is shown in Appendix 2.

The "&" and "?" commands provide the basic (one word at a time) communication from the North Star to the HP Multiprogrammer and from the HP Multiprogrammer to the North Star respectively. The remaining commands (#, %, !, A) are dedicated commands for use with a specific card in a specific slot. The function performed by each one of these commands can be alternately performed (in several steps) by the "&" and "?" commands. These command formats are explained and illustrated in Appendix 11. Alternate procedure for programming the same card, using the "&" and "?" commands, is given in Appexdix 12.

The monitor program is organized in a structured and modular fashion so as to permit easy addition of new commands for additional accessory cards. However, strict checks for illegal characters are not provided and therefore it is important to adhere to the command formats strictly.

CHAPTER IV

VOLTAGE/FREQUENCY MEASUREMENT CARD

4.1. V/F CARD DESCRIPTION.

The V/F card is an input card which provides for frequency measurement (of a pulse train meeting TTL requirement) or voltage measurements for voltages between -15 to +15 volts peak to peak [8]. The V/F measurement circuit is built on the HP Multiprogrammer breadboard input card 69480A. The circuit diagram for this V/F measurement circuit is shown in fig. 4.1. The breadboard input card allows the user to design and implement special input circuits. It includes basic address and read back circuit. The block diagram of the 69480A breadboard input card and its input connector are shown in fig. 4.2. The schematic diagram of the breadboard input card is shown in Appendix 5.

The heart of the V/F measurement circuit is the 7060 counter (Z11). It also has a voltage to frequency converter AD 537 (Z12), and a true RMS to DC converter AD 536 (Z13) to allow voltage measurements. The input train of pulses from the frequency source or the voltage (converted to frequency) source are fed to the 7060 counter which must first be reset. The counter is then permitted to count for a specific period of time at the end of which counting is stopped and the 32 bit counter read back by the computer a byte at a time. The 8 bit latch (Z8,Z9) and the address gate (Z10) are similar to the circuitary on the breadboard output card and allow the V/F card to be used as an output card from the programmer's point of view. The output functions required to be performed by this card include resetting of the 7060

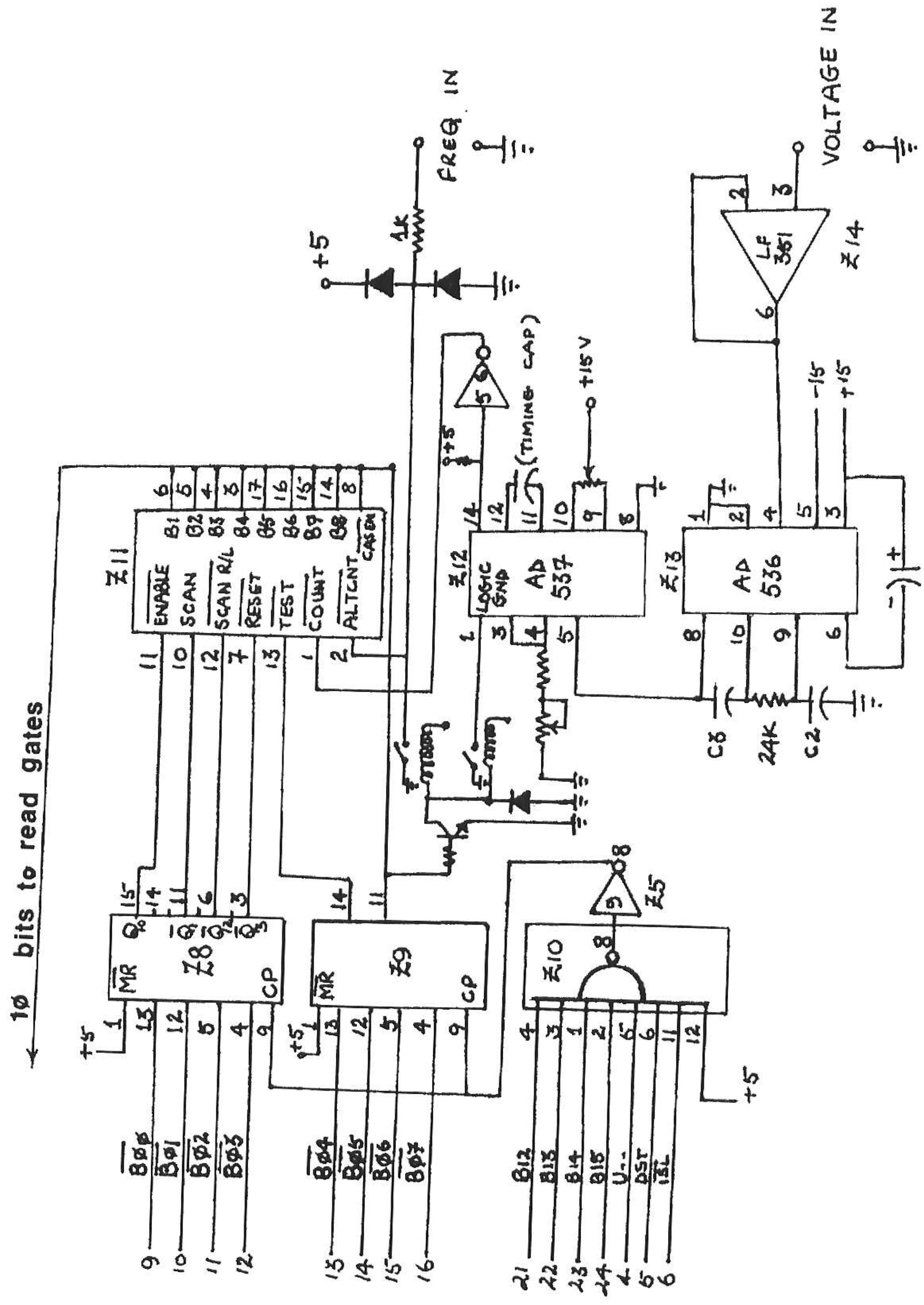
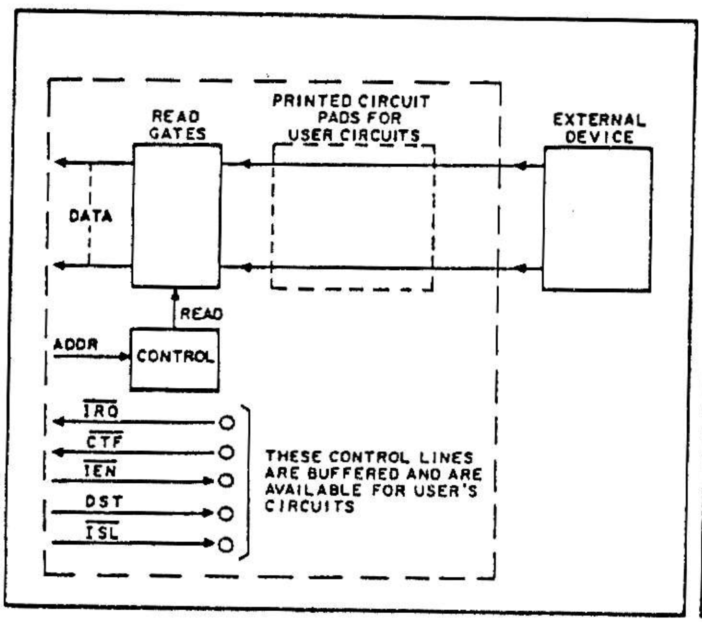
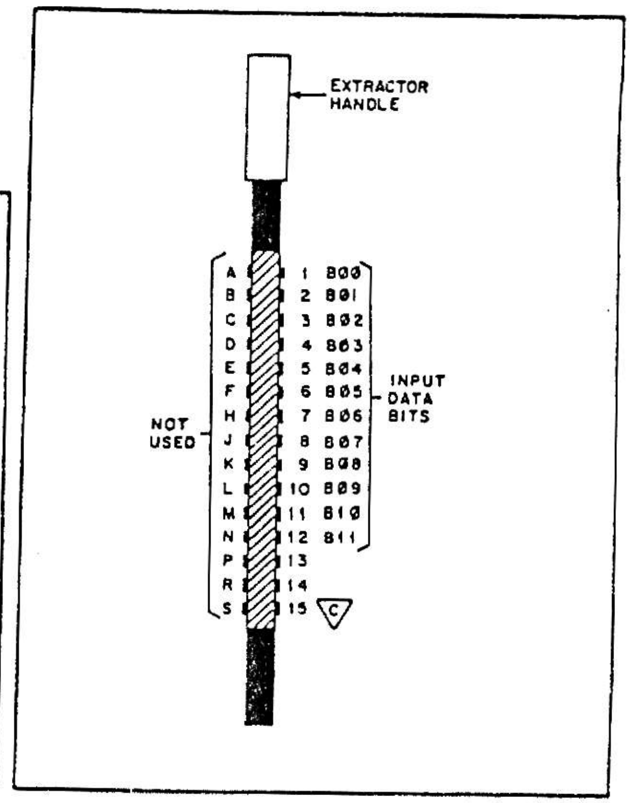


fig. 4.1 Schematic diagram of the V/F measurement circuit



69480A Block Diagram



69480A Input Connector

fig. 4.2 Block diagram of 69480A input card and its input connector.

counter, initializing it for counting and scanning the counter for read back operation a byte at a time. The functions of the various output data bits connected to the corresponding control pins of the 7060 counter are shown in fig. 4.3. The input data bits received from the V/F card, by the HP Multiprogrammer, are explained in fig. 4.4. Three control words are required to reset and initiate counting on the 7060 counter. The scanning and read back is achieved by 2 control words at the end of the time out period. The programming of the V/F card is explained in the following section.

4.2. V/F CARD PROGRAMMING.

The program to utilize the V/F card has to send 3 data words to provide the required transitions on the control lines to set it up for V/F measurements. A software time delay must then set up the period during which the input will be measured. Finally the 7060 counter must be scanned and its contents read back 8 bits at a time. The data read back can then be used to determine the voltage or frequency of the input. MC6802 has been programmed to recognize the symbol "#" as a command to perform the voltage or frequency measurement on the V/F card located in slot 413 (ie slot address = "1101" in binary) by following the above sequence of resetting the counter, time delay and read back operations. The command format and the code to read back the bytes returned from the counter is shown in appendix 11. The corresponding MC6802 assembly routine which performs this operation is the TIMER routine (see listing in Appendix 2 and flowchart in Appendix 1).

In case the V/F card is located in any other slot (other than slot

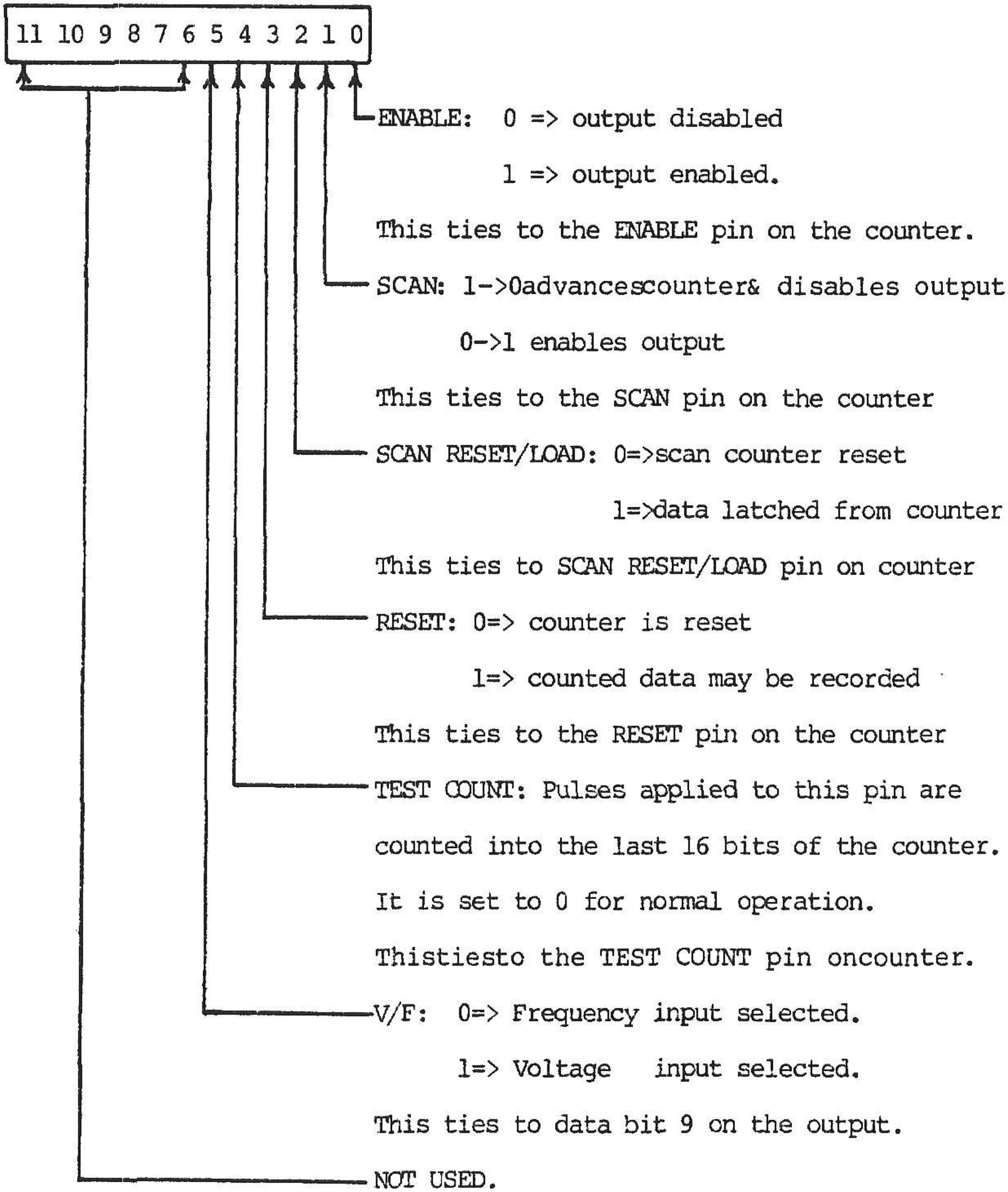


fig. 4.3 functions of the output data bits connected to the various control pins of 7060 counter.

413), then the "#" command cannot be used to perform voltage/frequency measurements. This is so because the current monitor program assumes the card to be housed in slot 413. Voltage or frequency measurements can be made through the V/F card in any other slot by following the steps shown in Appendix 12. This measurement will however take about 6 times longer than before because of the need for communicating each control word serially over the ACIA. Alternately the timer routine can be modified to accept the slot address where the V/F card is housed, in which case the user will be required to be aware of the slot position of the card.

The connections for the frequency or voltage source under test are brought out to the connector block. These are shown in Appendix 10.

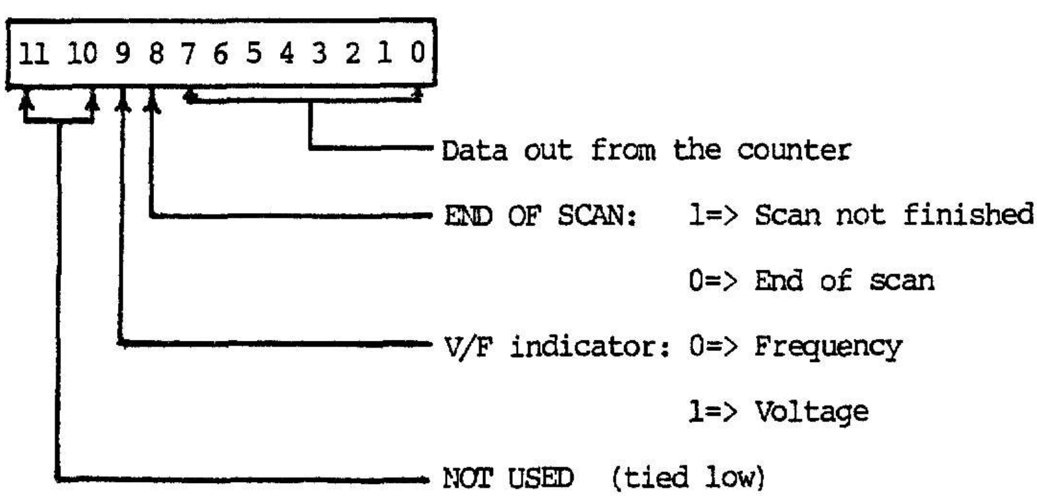


fig. 4.4 functions of the data bits read back from the V/F card

CHAPTER V

VOLTAGE CONTROLLED OSCILLATOR CARD

5.1. VCO CARD DESCRIPTION.

The voltage controlled oscillator (VCO) card is an output card which provides a digitally controlled function generator implemented on the HP Multiprogrammer's breadboard output card - 69380A. The breadboard output card allows the multiprogrammer user to design, build and control special output circuits through the multiprogrammer system. It includes the basic address storage and control signal buffer circuits. The block diagram of the 69380A breadboard output card and its output connector are shown in fig. 5.2. The schematic diagram of the breadboard output card is given in Appendix 6.

The schematic block diagram of the circuitry added to the breadboard output card to implement the VCO is shown in fig. 5.1 [7]. It consists of mainly a XR2206 function generator and two Digital to Analog Converters (DACs - for amplitude and frequency control) with associated sub-circuits, AMP voltage conditioning circuitry, output buffering and DC level adjusting circuits. The data lines of the AMPlitude control and FREQuency control DACs are connected to the corresponding data lines (Bits 0 to 7) of the latch circuits on the breadboard output card. The CS lines are connected to DTE from the multiprogrammer. Data lines 8 and 9 (D8 and D9) drive the WR lines of the AMP DAC and FREQ DAC respectively. The capacitor switching network employs switching to provide minimum path resistance. Four

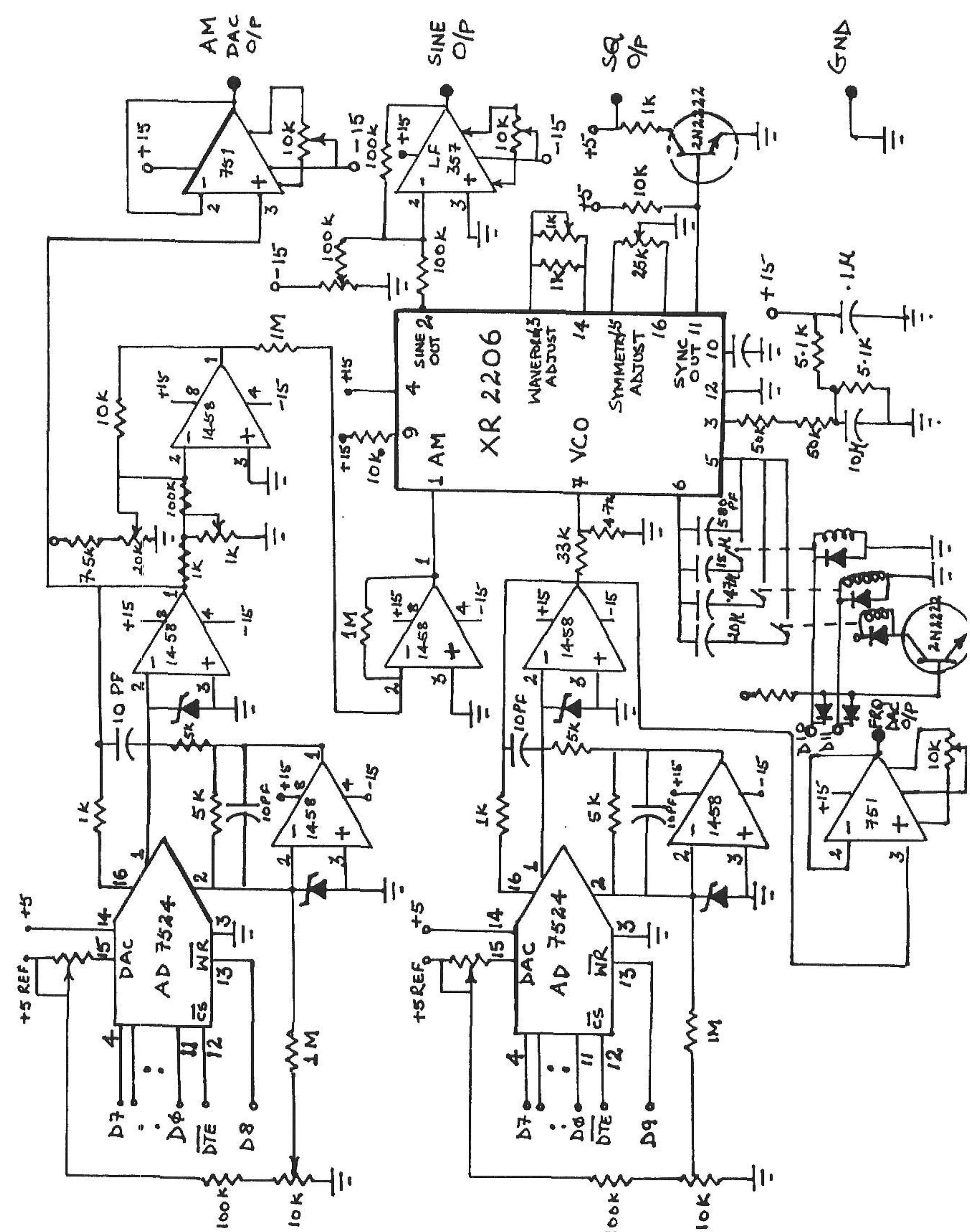
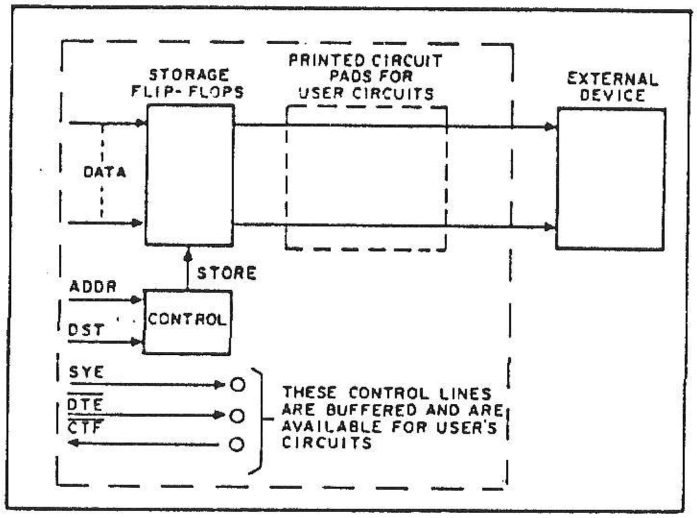
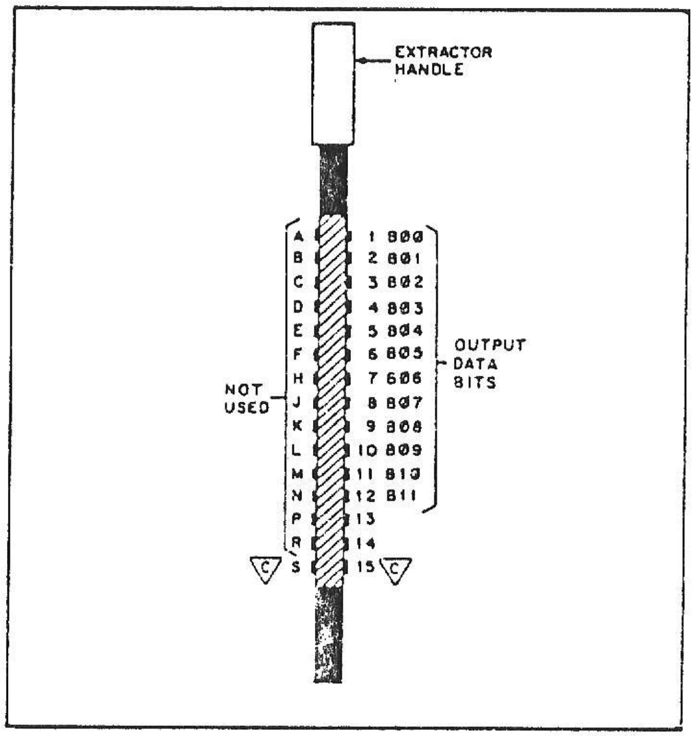


fig. 5.1 schematic diagram of the VCO circuit.



69380A Block Diagram



69380A Output Connector

fig. 5.2 Block diagram of the 69380A card and its connector frequency ranges are provided by controlling data bits 10 and 11. These ranges are as shown in fig. 5.3.

BIT 10	BIT 11	Frequency Range
0	0	.3 to 37 hz
0	1	37.0 to 1880 hz
1	0	1.09 to 48.5 Khz
1	1	.408 to 1.3 Mhz

fig. 5.3 Table of frequency ranges on the VCO

Both AMP and FREQ DAC outputs are brought out to the connector block for external use. The square wave output is buffered to provide additional drive current for the TTL output.

5.2. VCO CARD PROGRAMMING

The VCO output card receives control parameter data for voltage and frequency control from the user program and it develops a TTL compatible square wave in the frequency range of 1 Hz to 1.3 MHz. It also develops a sine wave output of amplitude between 10 mV to 6 Volts peak to peak, with a frequency range identical to the square wave output. The program to use the VCO card therefore has to send a control word to set up the programmer for output mode. The control

THIS BOOK CONTAINS NUMEROUS PAGES WITH THE ORIGINAL PRINTING ON THE PAGE BEING CROOKED.

THIS IS THE BEST IMAGE AVAILABLE.

word must then be followed by data words for frequency control (Bit 9 = 1) and amplitude control (Bit 8 = 1). The data words for amplitude control and frequency control must each be followed by data words for amplitude hold (Bits 8 and 9 = 0) and frequency hold (Bits 8 and 9 = 0) respectively, in order to meet the DAC interface timing shown in Appendix 9. All the data words need to be directed to the corresponding slot which holds the VCO card.

The MC6802 monitor has been programmed to recognize the symbol "!" as the command to control the VCO (if the VCO card is placed in slot 411 ie slot address = "1011" in binary). The command format to program the VCO, through BASIC, for a waveform of desired amplitude and frequency, is shown in Appendix 11. The assembly routine which recognizes the VCO command "!" and provides the required control to produce the desired waveform is the VCO routine (see listing in Appendix 2 and flowcharts in Appendix 1). If the VCO card is placed in any other slot other than slot 411 (K), then the command symbol "!" cannot be used to control the VCO card. In this case the 5-Step procedure, outlined in Appendix 12, will have to be followed in the program that controls the VCO card.

CHAPTER VI

RELAY OUTPUT CARD WITH READBACK, 69433A

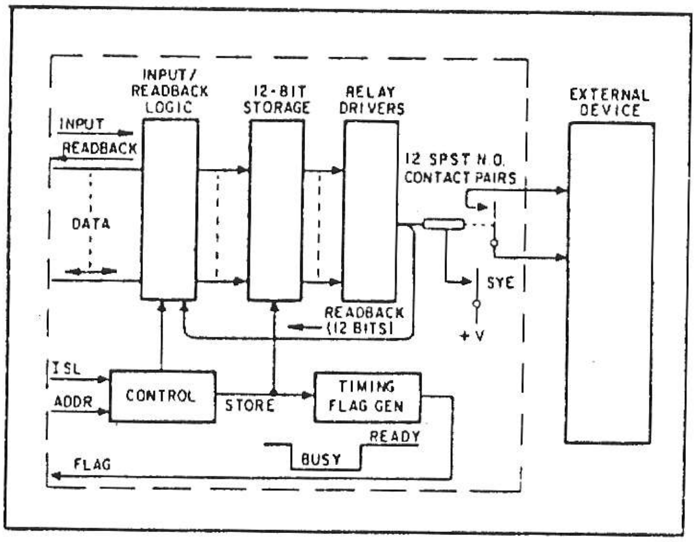
6.1. RELAY CARD DESCRIPTION

The 69433A relay output card is a factory built card. It provides 12 independent, SPST, normally open, isolated relay contacts controlled individually by the system user through a program. The 69433A relay output card also allows the user program to examine the status of the relay coil drive circuits on the card before and/or after the contacts are changed [10].

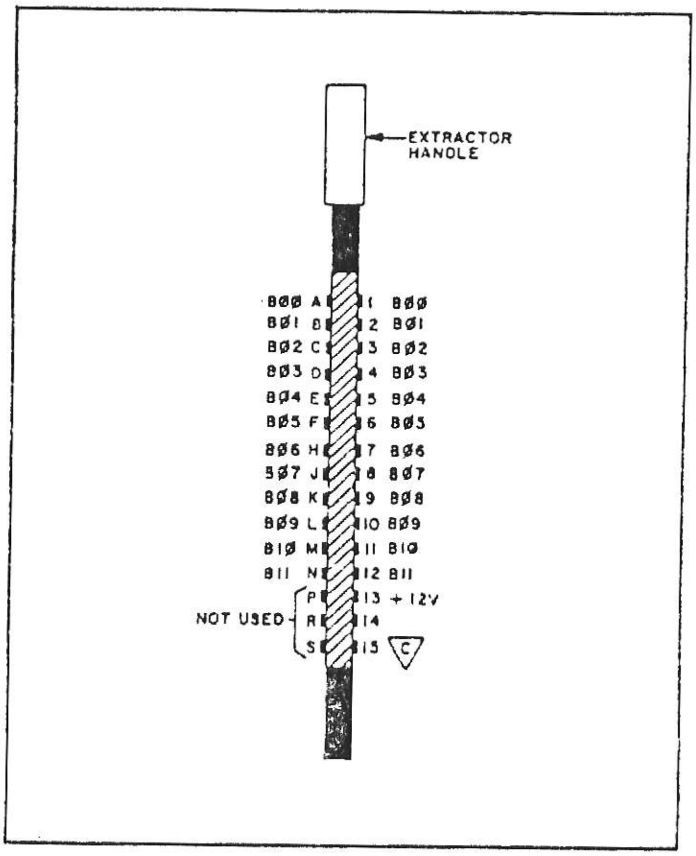
This card is placed in slot 401 (A) of the HP Multiprogrammer system. A block diagram of the card and its output connector are shown in fig. 6.1. Connections to either side of each contact point are brought out to the connector block as shown in the terminal assignments for the connector block in Appendix 10. Programming of this card is explained in the following section.

6.2. RELAY CARD PROGRAMMING.

To program the Relay card for contact closure, the system must first be configured for output mode by sending a control word with SYE, DTE, and TME on. The control word must be followed by a data word (output data word) containing the slot address and the output data. The output data is a 12 bit binary number with a "1" in the corresponding relay position which needs to be closed and a "0" in the relay position which needs to be kept open. The relay status may be examined at any time by the user program. To examine the status the system must



69433A Block Diagram



69433A Output Connector

fig. 6.1 Block diagram of the 69433A Relay Card and its output connector first be set up for an input mode by a control word with ISL and SYE on. This control word must be followed by an Address Word addressing the slot holding the relay card. The relay status bits are then available at the HP Multiprogrammer return data lines, and may be read back by the user program to examine the status of the relays.

The MC6802 interface has been programmed to provide the user with the "%W" command (in BASIC) to close the required contacts, and the "%R" command to examine the status of the relays. The corresponding Assembly routine which sends the required control and data words, upon recognizing the above commands, is the RELAY routine (see Appendix 1 for flowcharts and Appendix 2 for listing). The command formats are explained with illustrations in Appendix 11. If the relay card is placed in any other slot, other than slot 401 (A), then the user program will be required to send the appropriate control word followed by the corresponding data word, as outlined in the procedure in Appendix 12.

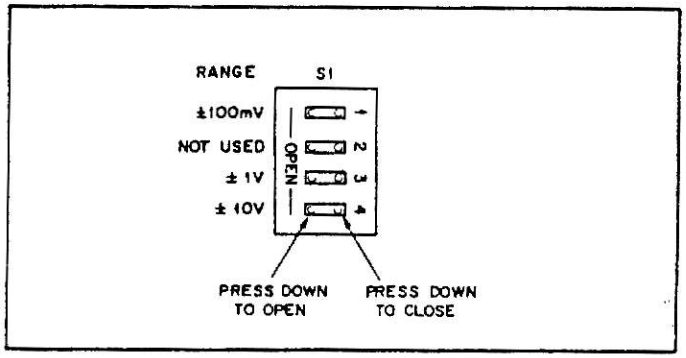
CHAPTER VII

HIGH SPEED ANALOG TO DIGITAL CONVERTER CARD, 69422A

7.1. A/D CARD DESCRIPTION.

The 69422A card is also a factory built card. It monitors bipolar dc levels in the ± 100 mv, ± 1 v, ± 10 v, or ± 100 v range, and returns a 12 bit 2's complement digital word to the MC6802 to indicate the magnitude and sign of the measured voltage [14]. Voltages in the ± 100 V range are connected to a divide-by-ten attenuator network on the card. The required range can be selected by the range switches (4 individual OPEN/CLOSE type switches), on the card, to the applicable input range. The range switches and the switch settings are shown in fig. 7.1. A block diagram of the 69422A card is shown in fig. 7.2.

The input voltage is converted to an equivalent 12-Bit digital word upon selecting the HP multiprogrammer for the Input Select mode (ie ISL on) and addressing the 69422A card with a gate signal. The digital word is stored in the card and is available to the MC6802 via the multiprogrammer return data lines. This word remains on the return data lines as long as the card is addressed and the input select mode is programmed ON. It is also retained by the card until the next conversion. The minimum conversion time through the backplane of the multiprogrammer is 50 micro seconds (ie 20 K conversions per second). However, due to program controlled acquisition of the samples, the actual conversion rate is 9.12 K samples per second. The card can also be used for conversion through an external trigger. Output data and



Range Switches

Range Switch Settings

INPUT VOLTAGE RANGE	RANGE SWITCH SETTINGS**			LSB
	S1-1 (±100mV)	S1-3 (±1V)	S1-4 (±10V)	(Min. Volt.Step)
±100mV (+0.10235V to −0.10240V)	CLOSE	OPEN	OPEN	50µV
±1V (+1.0235V to −1.0240V)	OPEN	CLOSE	OPEN	500µV
* ±10V ±10.235V to −10.240V)	OPEN	OPEN	CLOSE	5mV
* ±100V (+102.35V to −102.40V)	OPEN	OPEN	CLOSE	50mV

fig. 7.1 A/D card range switches and a table of the range switch settings

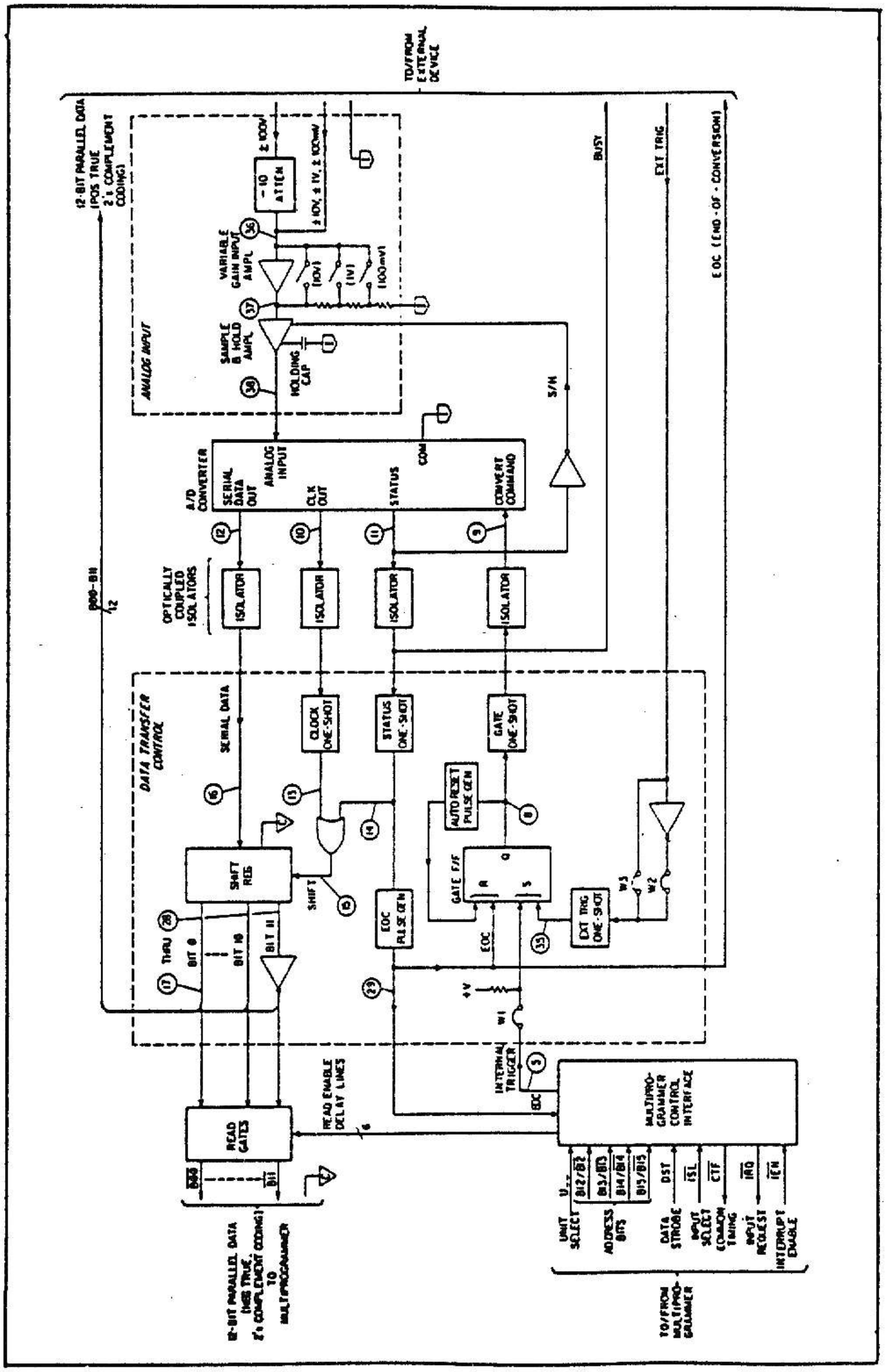


fig. 7.2 Block diagram of the high speed A/D card

control signals to synchronize data transfers with an external device are available at the card's edge connector. The minimum conversion time for external data output is 30 micro seconds (ie 33 K samples per second). The programming of the 69422A card via the MC6802 and the calculation of voltage values, from the digital word returned by the multiprogrammer, is explained in the following section.

7.2. A/D CARD PROGRAMMING.

The program to sample voltage signals through the 69422A card must first select the HP Multiprogrammer for an input mode by a control word with ISL and TME on. This control word is then to be followed by an address word with a gate signal. The gate signal initiates conversion, and the end of conversion is indicated by the receipt of the flag from the multiprogrammer. The 12 bit binary word corresponding to the voltage is then available on the return data lines as long as the card is addressed. This word can be converted to the corresponding voltage value by multiplying it with the appropriate step value in volts (see fig. 7.2 for step values for different ranges). If the data word is negative (ie its value is greater than or equal to 2048), then the two's complement of the number obtained must be taken (by subtracting 4096 from it) before converting it to volts.

The MC6802 is programmed to provide the command "A", to the BASIC user, to collect 512 samples via the 69422A card placed in slot 405 (E). Upon recognizing this command "A" the monitor program passes control to the ADC routine (see flowcharts in Appendix 1 and listing in Appendix 2). This routine collects 512 samples, by providing the required control and address word with the gate to the multiprogrammer, and sends them to the North Star. The number of samples collected is restricted to 512 by the limited memory currently available on the MC6802. Also the number of samples collected on every command is fixed at 512 in order to achieve faster sampling rate by saving on additional instructions required to provide the user with the option of variable samples. The user may ignore the extra samples in his BASIC program. The command format and the code required to calculate the voltage value from the binary word is illustrated in Appendix 11.

If the 69422A card is placed in any other slot (other than slot 405), then the above command cannot be used to sample voltage signals through the card. In this case the procedure outlined in Appendix 12 will have to be followed to sample the voltage signal. However, this procedure would yield a very slow conversion rate owing to the serial communication of each control and address word required for every sample.

The 69422A card is a very sophisticated tool for use with digital signal processing type application. A sinewave sampled at 9.12 K samples per second is shown in fig. 7.3. The program for sampling and analyzing a signal connected to the ADC card is listed in Appendix 14. The spectrum of the above sinewave obtained from this program is shown in fig. 7.4.

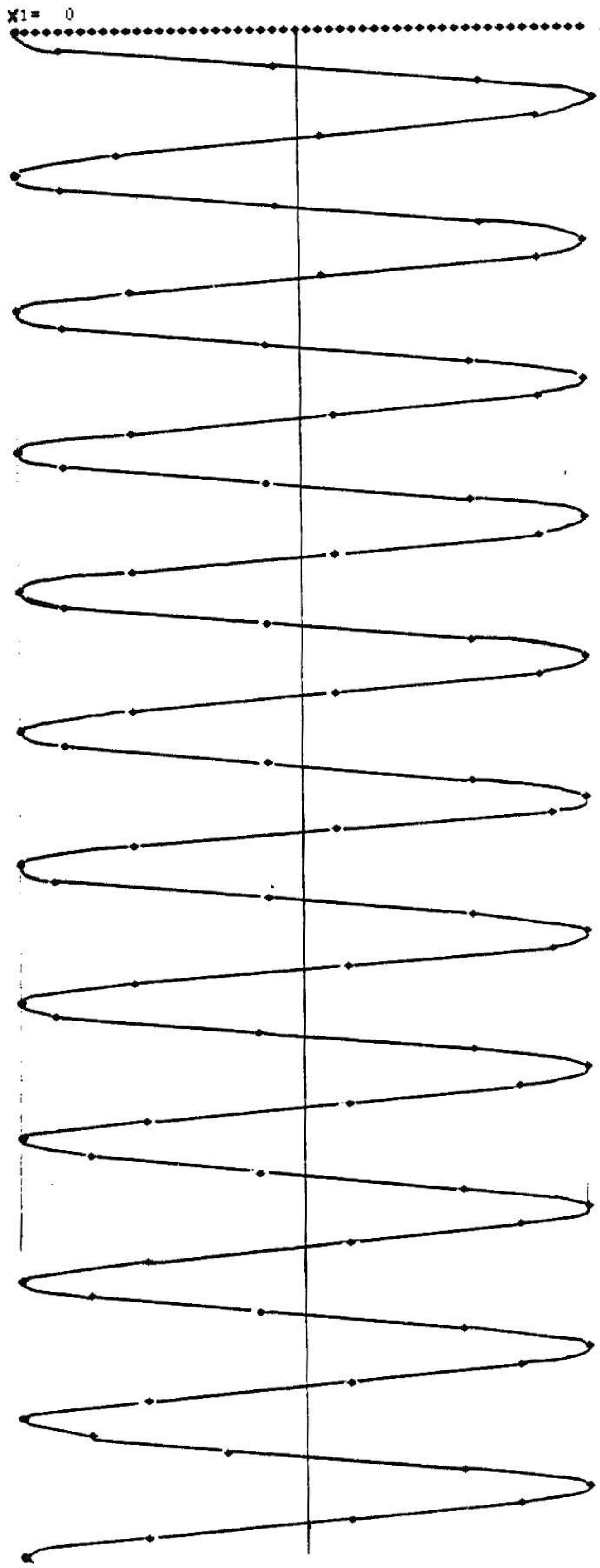


fig. 7.3 Plot of a sinewave sampled by the A/D Card.

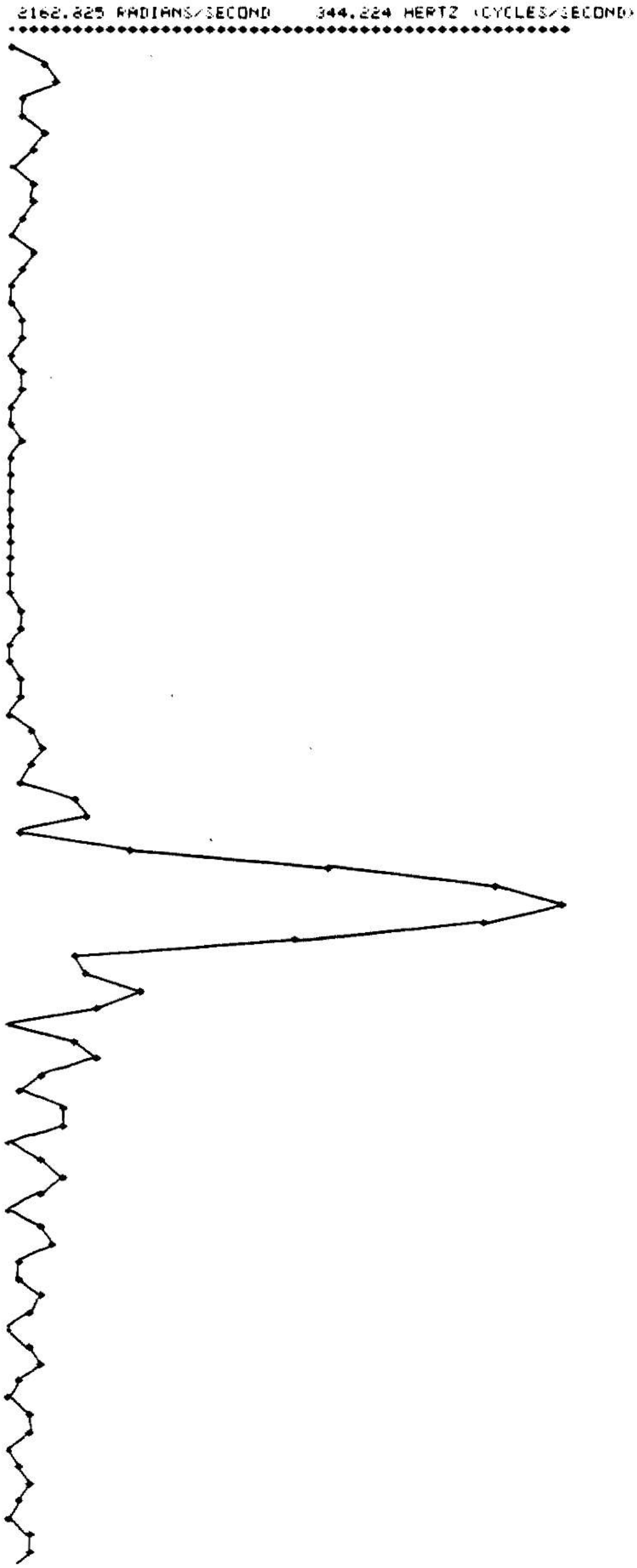


fig. 7.4 Frequency spectrum of the signal shown in fig. 7.3.

CHAPTER VIII

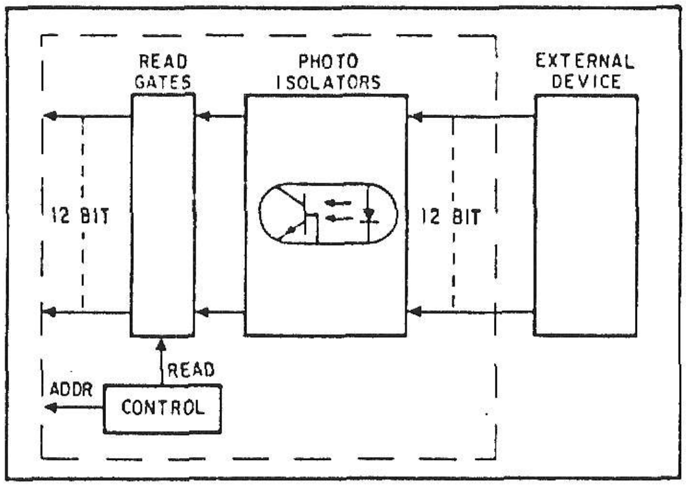
ISOLATED DIGITAL INPUT CARD, 69430A

8.1. ISO CARD DESCRIPTION

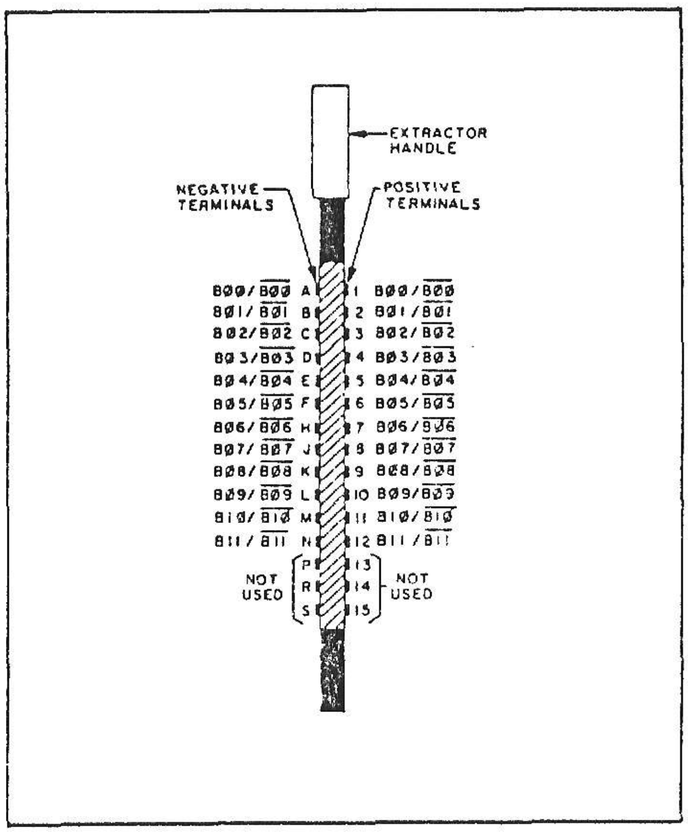
The Isolated digital input card allows the North Star user to read 12 bits of logic level data that are isolated from ac earth ground. It thus eliminates ground loop problems encountered in an automated tests and control system [5]. The card is placed in slot 400 (A) (ie its slot address = "0000"). A block diagram of this card and its input connector are shown in fig. 8.1. The negative and positive terminals are brought out to the connector block as shown in the connector block terminal assignment in Appendix 10.

8.2. ISO CARD PROGRAMMING

To read the digital inputs, via the Isolated Digital Input card, the HP Multiprogrammer must first be set up for input. This is done by a control word with ISL and SYE on. The control word is followed by address word addressing the slot holding the Isolated Digital Input card. The 12 bit binary word, corresponding to the states of the digital inputs at the input connector, is then returned by the Multiprogrammer on its return data lines. The ON state of an input is indicated by a "1" in the corresponding bit position, while the OFF state is indicated by a "0". No special command has been provided to communicate with this card as it can be read by the simple 3 steps procedure outlined in Appendix 11.



69430A Block Diagram



69430A Input Connector

fig. 8.1 Block diagram of the ISO card and its input connector.

CHAPTER IX

AUTOMATED DIAGNOSTICS

This automated diagnostics feature has been added to test for the proper functioning of all the modules in the system. The wiring to connect all the modules via the relays is provided on a single board with spring loaded contacts. The schematic diagram of this feature is shown in fig. 9.1.

The program controlling the tests first checks the V/F card. for this purpose the 110 baud rate clock from the North Star is used as the test input. The signal frequency measured by the V/F card should be between 1.745 and 1.765 Khz and the measured RMS voltage should lie between 3.6 and 3.7 volts. If the measured values fall outside these expected ranges, then further tests cannot be conducted and the diagnostics program is aborted. After a successful test on the V/F card, a 86.6 Khz and 3.4 volts rms signal is output on the VCO. The frequency and voltage of this signal are then measured via the V/F card. If the measured values are not as expected, then the measured values are printed out along with a message indicating the erroneous operation of the VCO. The FREQ DAC and the AMP DAC outputs are tested next. These outputs are then used as inputs to the Isolated Digital Input card and the high speed Analog to Digital Conversion card. The required connection to test each of the modules is made through pre-assigned relays. If a relay should malfunction, the test for the corresponding module is skipped and a message printed out to that effect. The listing for this program is given Appendix 13.

To use the Diagnostics feature, the diagnostics board with spring loaded contacts must first be mounted on the connector terminal block and then the program "DIAG" loaded and executed. The program prints out the descrepancies, if any, and stops at the end with the "END OF DIAGNOSTICS" message. It is recommended that diagnostics be performed before starting any lengthy experiment.

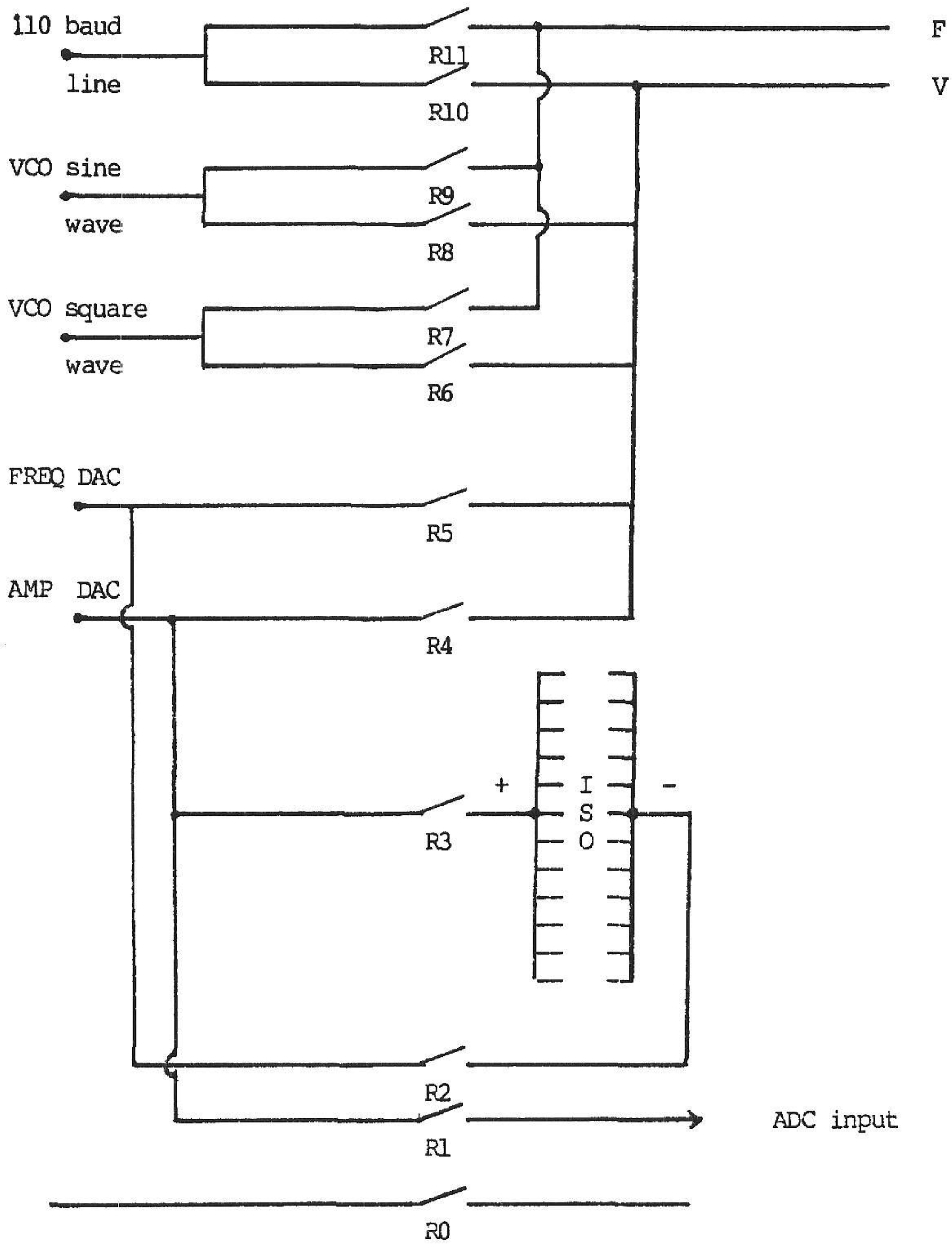


fig. 9.1 Schematics for diagnostics

CHAPTER X

OPERATING PROCEDURE

The procedure to conduct an automated test on this Automated Laboratory-tests System is as outlined below.

A. Connect the device under test to the required instruments (I/O cards) available on the automated system. All the points for connecting an external device to any I/O card are brought out on a connector block. The connector block terminal assignments are shown in Appendix 10.

B. Start up the North Star Horizon as follows.

1. Turn on the silent 700.

   (Push back the black switch on the upper right rear corner.)

2. Insert floppy disk labeled SYSTEM PROGRAMS into left hand disk drive (ie drive 1) of the Horizon.

3. Turn on the North Star Horizon: use black switch on the back side of the Horizon, next to the fan.

   The silent 700 will now print out:

   NORTH STAR DOS 5.2 AT 100
   +

   The + sign is the DOS prompt for the next command. To get into the BASIC programming mode, enter the GO BASIC command. ie.

   +GO BASIC (hit return)

The silent 700 will respond with READY on the next line.

4. Load the required test program from the system disk usingthe LOAD command. ie.

   LOAD XXXXX (hit return)

   XXXXX stands for the name of the test program.

C. Turn on the Multiprogrammer system. Use LINE ON switch on front panel.

D. Turn on the MC6802 interface. Use switch with red LED on left front.

At this point the system is set up and ready to go. Use theRUN command to execute the LOADed program. The program will control the experiment, print out the results, plot graphs etc. If the program to control a particular experiment does not exist, then a new program may be written using the commands explained in Appendices 11 and 12 to control the different instruments used in the experiment.

CHAPTER XI

CONCLUSIONS

The major components of the system and the various input/output accessory cards available on the system have been discussed in sufficient detail in chapters II to IX. To give a good understanding of the system and its operation these chapters are fairly complete. Extensive documentation is available on each one of the major components and the input/output accessory cards. A bibliography of these references is provided.

All the input/output accessory cards have been tested and found to function accurately within their operating limits. A table of the operating limits of the V/F card and the VCO card is provided in fig. 11.1. The high speed A/D is programmed to sample at the rate of 9120 samples per second. Therefore the maximum frequency of the signal under study should not exceed 4560 Hz in order to be able to perform any meaningful analysis of the signal. Also the number of samples is fixed at 512. So, at the above rate sufficient resolution of a slow signal cannot be obtained. The frequency of the signal under study should be greater than 50 Hz. To study slower signals a software delay will have to be introduced to decrease the sampling rate and thus provide better resolution.

	voltage	frequency
V/F CARD	11 Volts p-p (max)	100 Khz (max)
VCO SINE WAVE	6 Volts p-p (max)	1.3 Mhz (max)
VCO SQUARE WAVE	6 Volts p-p (max)	1.3 Mhz (max)
V/F CARD (COUNTER)	5.5 Volts p-p (min)	0.0 hz (min)

fig 11.1. A table of operating limits for the V/F and the VCO cards.

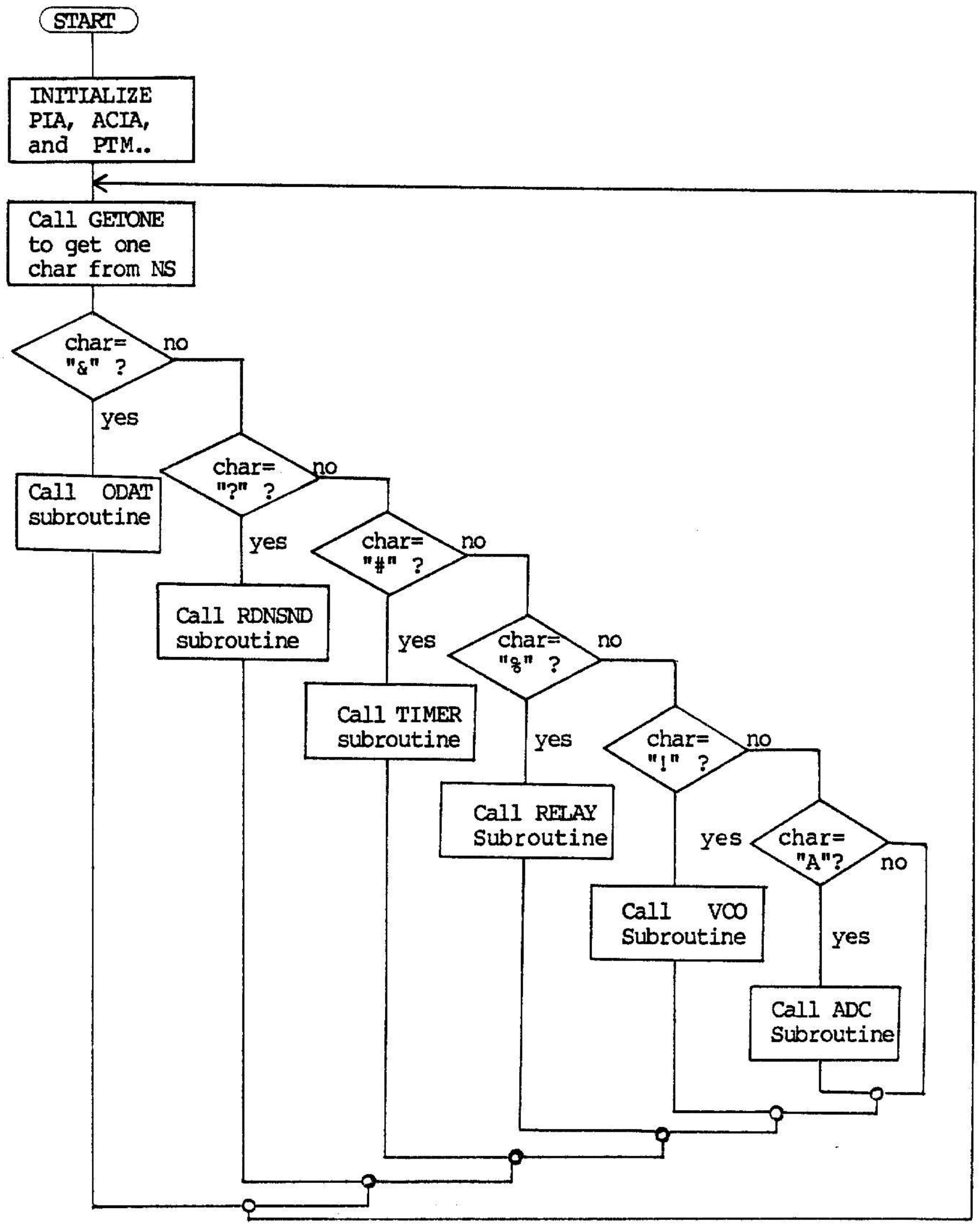
APPENDIX 1

Flowcharts and program description.

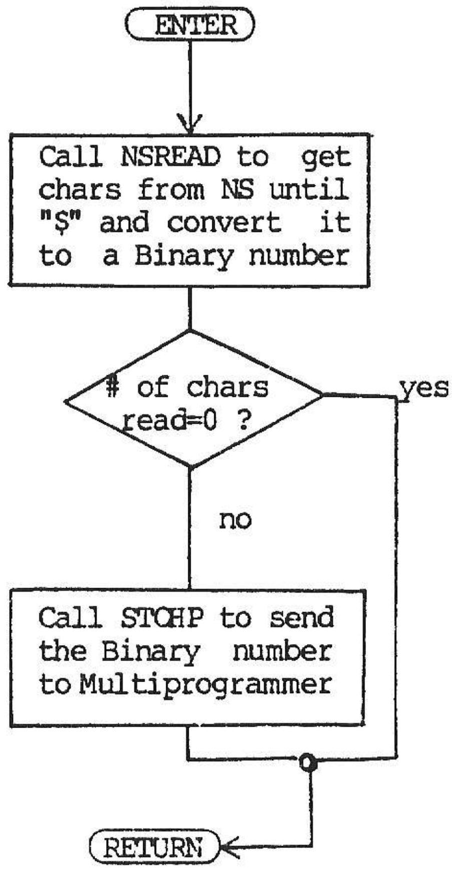
1. MAIN PROGRAM

This is the driver routine which accepts and decodes the command symbols (&, ?, #, %, !, A). It then calls the appropriate routine (ODAT, RDNSND, TIMER, RELAY, VCO, or ADC) to receive the corresponding command parameters and perform the required task. A typical task performed by one of these routines involves - providing the required control and data words to the HP Multiprogrammer, receiving the return data from the corresponding slot in the Multiprogrammer, converting the return data form Binary to ASCII, and sending it back to the North Star. On completing a particular command control is passed back to the start of the main program for receiving and decoding the next command (when sent). It is important that the command format is strictly adhered to since thorough checks for illegal characters etc have not been implemented.

Main program flowchart.



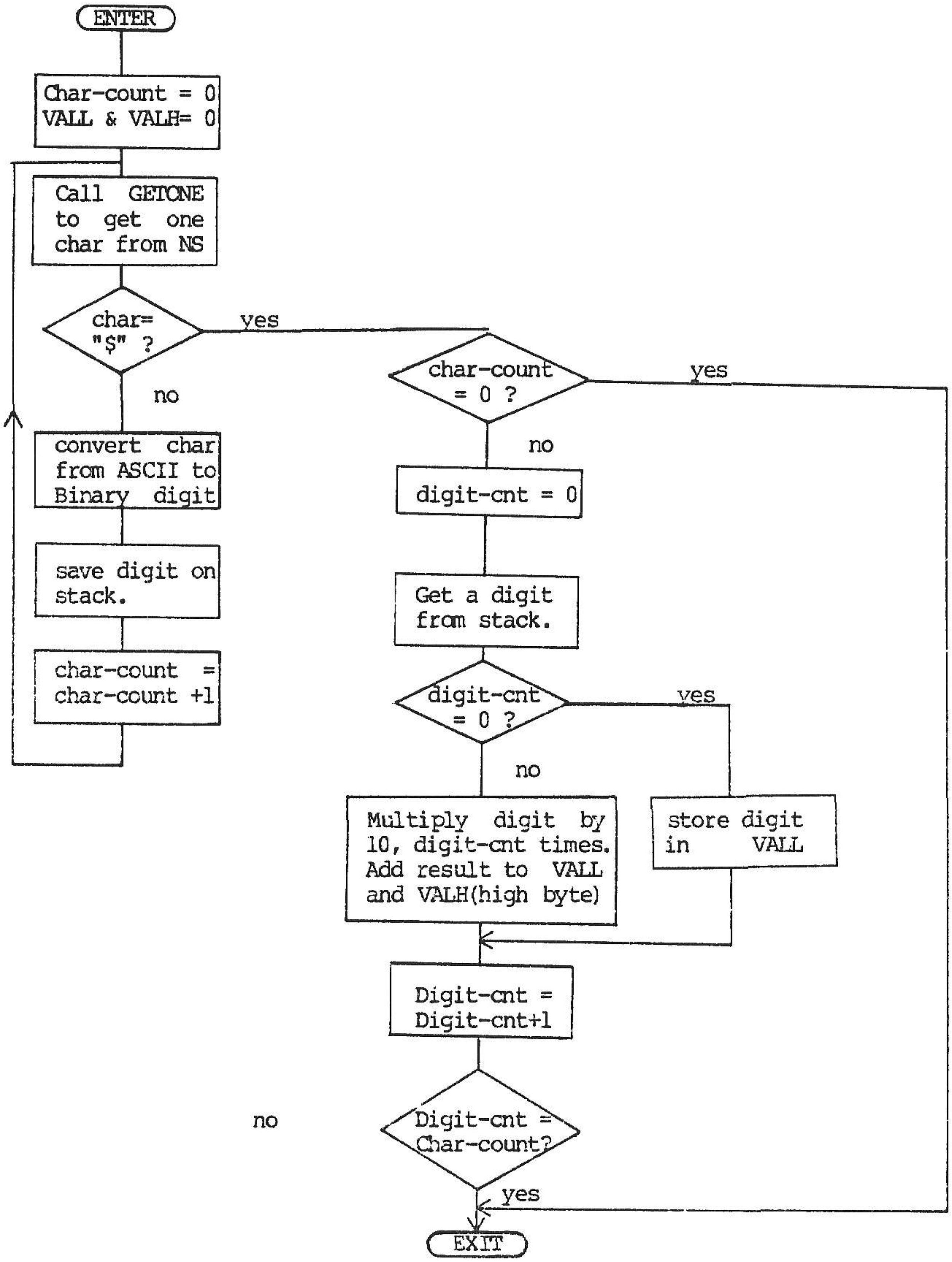
2. ODAT ROUTINE



ODAT routine description.

This routine calls NSREAD routine to accept an ASCII coded decimal number from the North Star, and converts it to a 16 bit Binary number. It then calls STOHP routine which sends the Binary number (stored at VALL & VALH by NSREAD) to the 16 bit input port of the HP Multiprogrammer. The ODAT routine is called from the Main Program by the command symbol "&". This command symbol provides a one way communication from theNorth Star to the Multiprogrammer (via the MC6802) and can be used along with the "?" command to simulate all other commands.

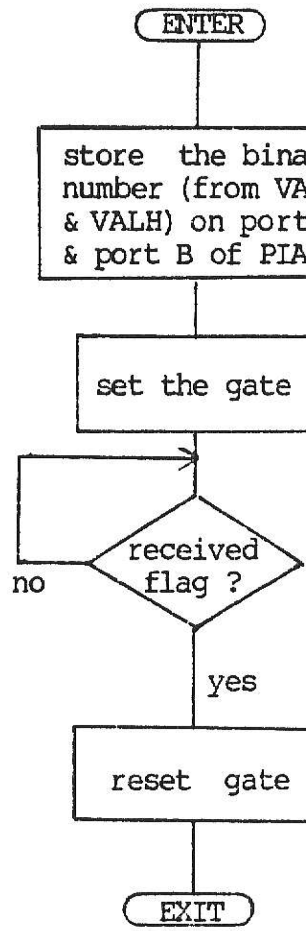
3. NSREAD routine flowchart.



NSREAD routine description.

This routine is called from various other routines to accept ASCII coded decimal number from the North Star and convert it to a 16 bit Binary number. It is important that only decimal digits be sent from the North Star and the last digit is followed by the character "$". On receipt of the "$" symbol the digits received are converted to Binary and control is passed back to the calling routine. The corresponding 16 bit binary number is passed back in memory locations VALH (High byte) and VALL (Low byte). (See page 4 of listing in Appendix 2).

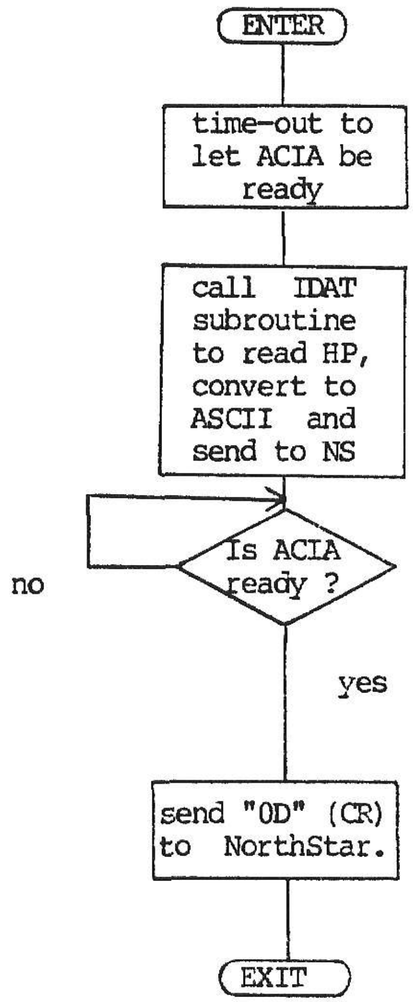
4. STOHP routine flowchart.



STOHP routine description.

This routine takes the binary number stored at VALH and VALL (by NSREAD or any other routine) and outputs it to the PIA ports connected to the HP Multiprogrammer input port. It then sets the gate and awaits the flag from the multiprogrammer before resetting the gate and passing control back to the calling routine.

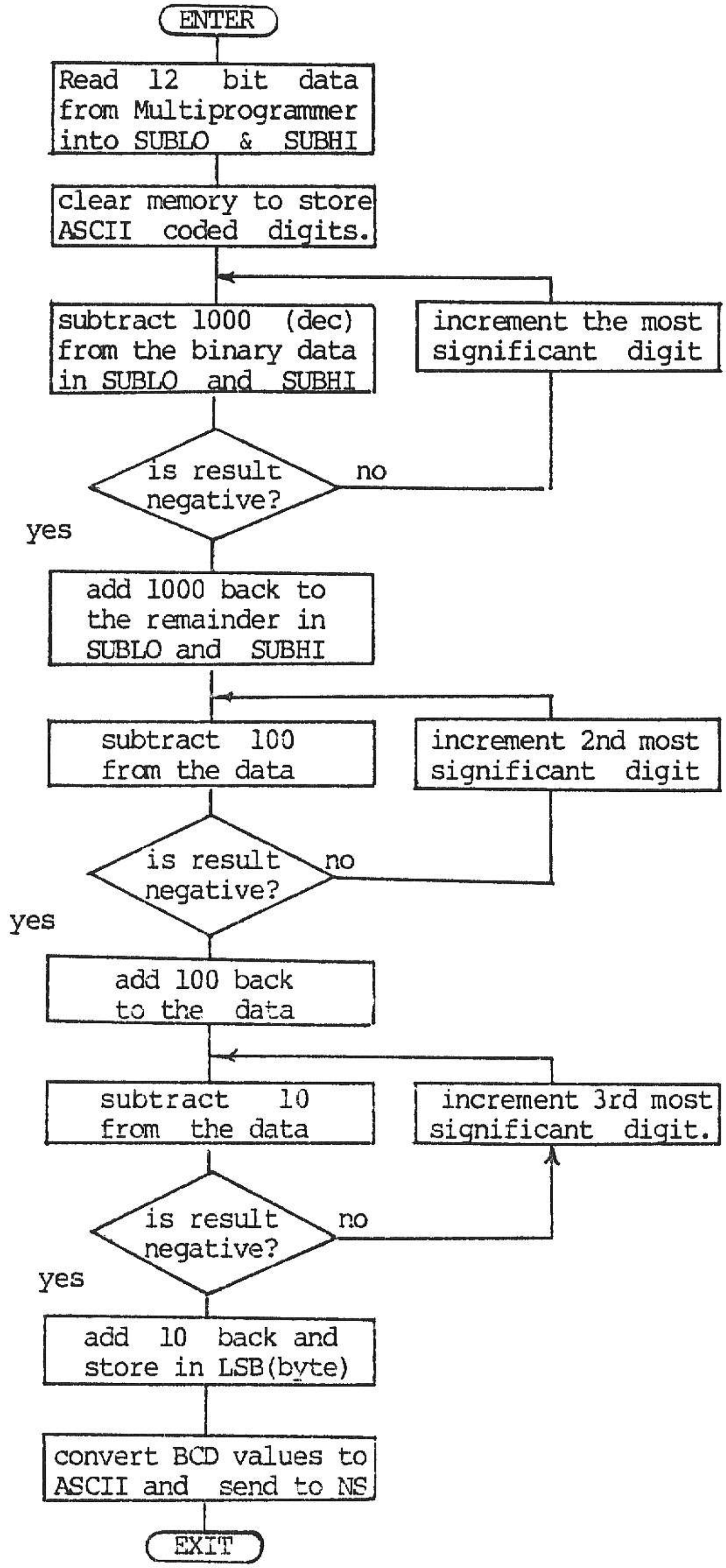
5. RDNSND routine description.



RDNSND routine description.

This routine is called from the main program on the receipt of the command symbol "?". It then calls IDAT routine to read the return data from the multiprogrammer, convert it to ASCII coded decimal digits, and send each digit to the North Star. The last digit is terminated by a carriage return character and control is passed back to the main program.

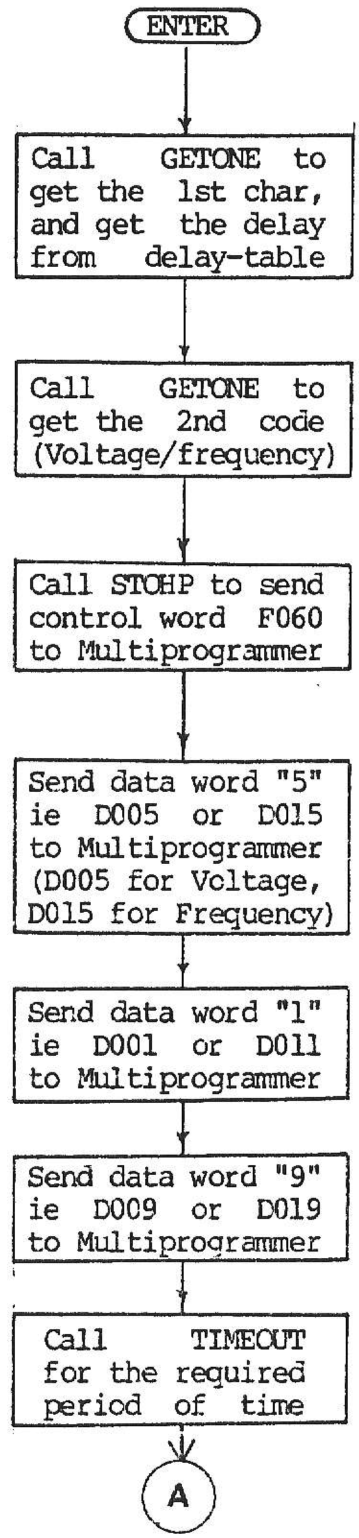
6. IDAT routine flowchart.

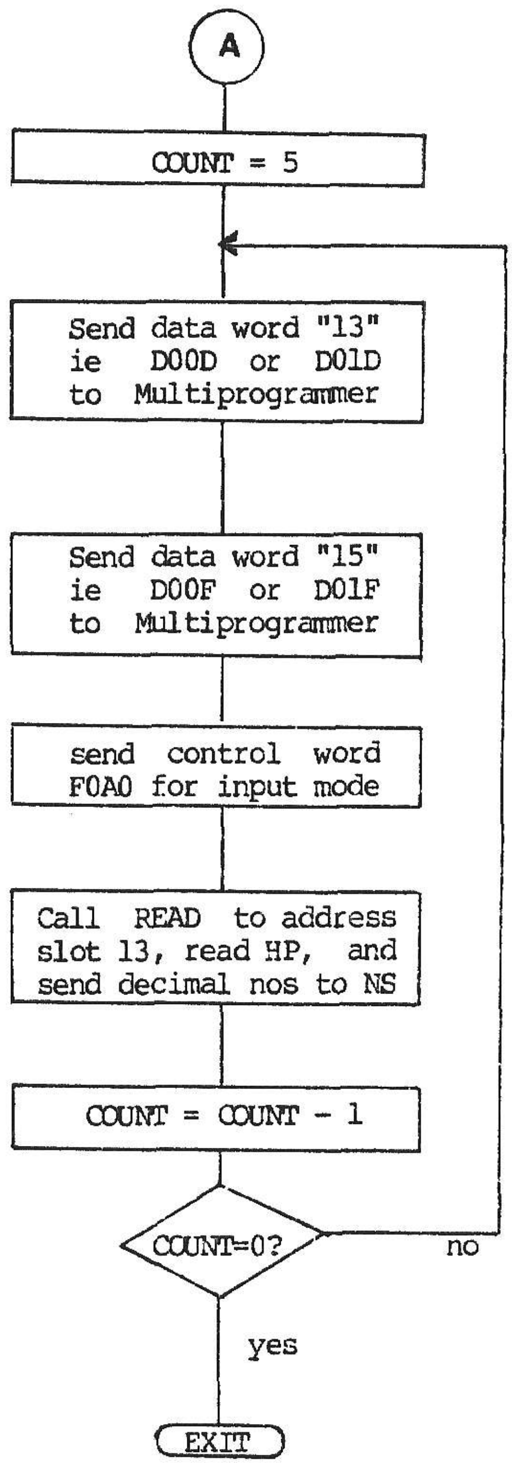


IDAT routine description.

This routine reads the return data from the HP multiprogrammer, converts it to ASCII coded binary digits, and sends each digit to the North Star. The return data may be from an input card (if the HP was set up for an input mode) or it may be the echo back of the output data sent to the HP. This routine may be called from the entry point WSEND (see page 7 of listing), to send data from memory to the North Star. The data from memory should be passed to this routine through locations SUBLO and SUBHI. It calls the routine SUB16 to perform the 16 bit binary to ASCII conversion.

7. TIMER routine flowchart.

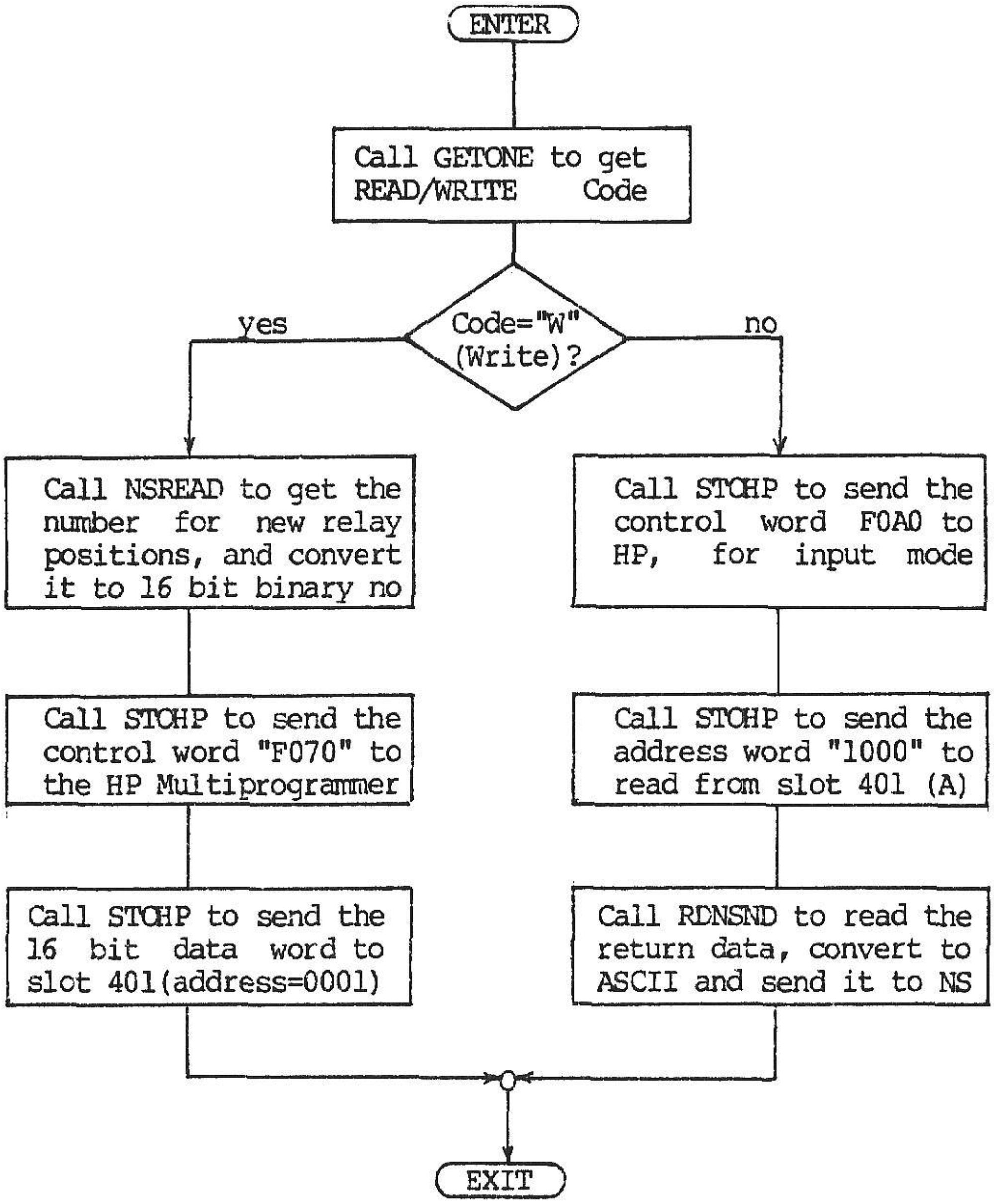



A
COUNT = 5
Send data word "13"
ie D00D or D01D
to Multiprogrammer
Send data word "15"
ie D00F or D01F
to Multiprogrammer
send control word
F0A0 for input mode
Call READ to address
slot 13, read HP, and
send decimal nos to NS
COUNT = COUNT - 1
COUNT=0?
no
yes
EXIT

TIMER routine description.

This routine controls the communication between the North Star and the V/F card in slot 413 (M) of the HP Multiprogrammer. It is called by the main program on recognizing the command symbol "#". It accepts two codes (one digit each) from the North Star user program. The first code indicates the time duration for the measurement of voltage or frequency. The second code indicates whether voltage or frequency is to be measured. The routine provides the control and data words to initiate counting on the counter and times out for the required period of time. At the end of this period it reads back the 5 byte data from the counter, converts it to decimal coded ASCII data, and sends it to the North Star. The command format and the code required to follow the command is explained and illustrated in Appendix 11.

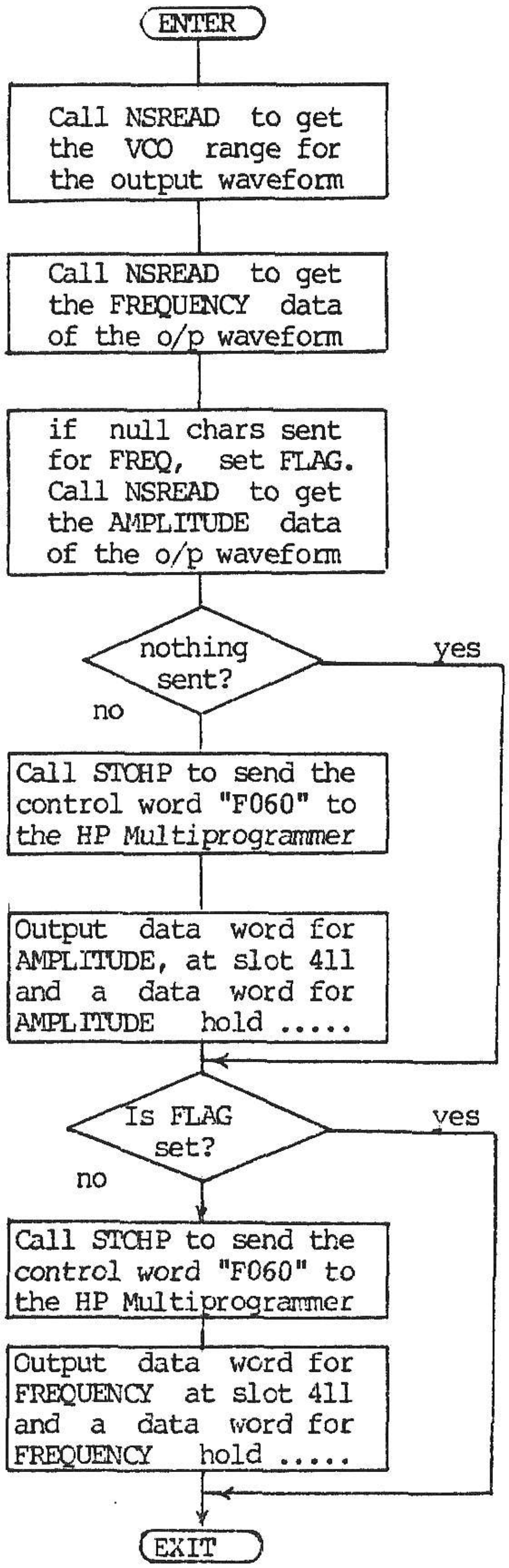
8. RELAY routine flowchart.



RELAY routine description.

This routine controls or examines the relay card in slot 401, for the North Star user. It is called from the main program by the command symbol "%". the command format for this "%" command is shown in Appendix 11. The routine first accepts a one-character code for reading (R) from, or writing (W) onto the relay card placed in slot 401. To read the relay card it sends the required input control word to the HP Multiprogrammer, reads the return data word, and sends the ASCII coded decimal equivalent to the user program. To establish (write) new relay positions it first accepts the decimal number corresponding to the binary pattern for the desired relay positions (a "1" for a closed relay, and a "0" for a open relay). The multiprogrammer is then set up for the output mode, and the required data word output to the relay card.

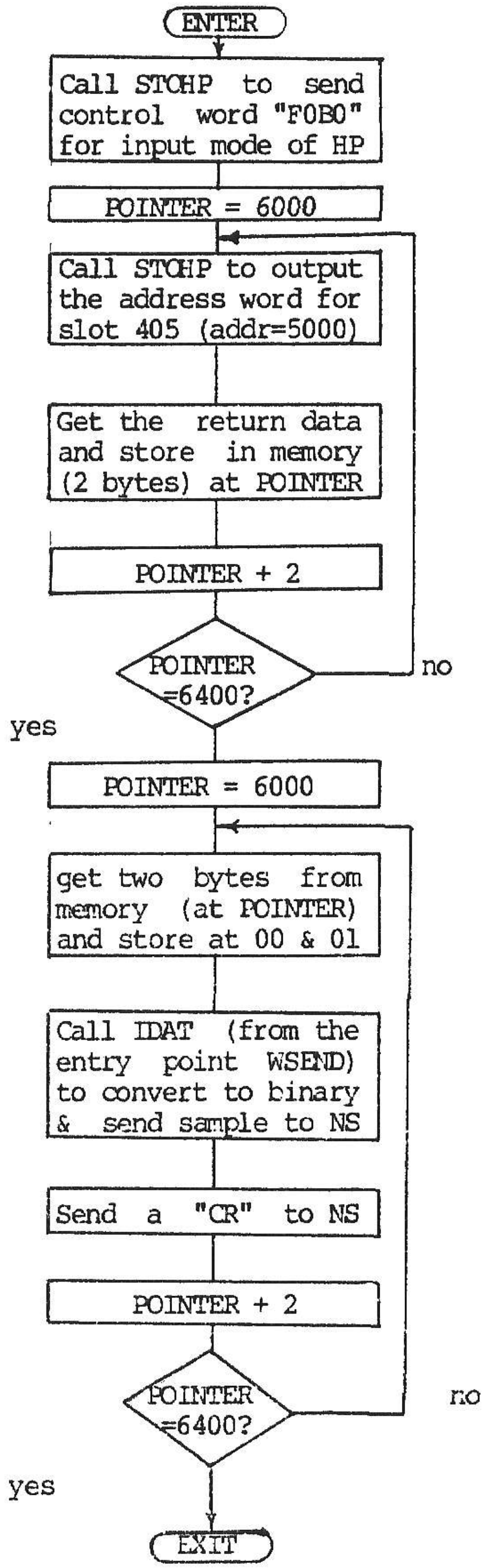
9. VCO ROUTINE



VCO routine description.

This routine accepts the Amplitude and frequency data from the user, and outputs the corresponding signal on the VCO card placed in slot 411. The routine is called from the main program by the command symbol "!". The command format is shown in Appendix 11. The routine first accepts the range in which the VCO is to operate. Then it gets the data for frequency and amplitude control and outputs the required control and data words to get the desired signal on the VCO. If no data is sent for frequency or amplitude control then the frequency or amplitude is left unchanged from its earlier setting.

10. ADC routine flowchart.



ADC routine description.

This routine is called from the main program by the command symbol "A". It configures the HP Multiprogrammer for an input mode and collects 512 samples of the signal at the ADC card placed in slot 405. The samples, temporarily stored in memory, are then sent back to the user program in the North Star. The command format and the code required to receive the samples is shown in Appendix 11.

APPENDIX 2

6802 Monitor Program Listing.

MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.1

```
                         NAM    MICRO1
           0050   STAKPT EQU    $50
           A000   PTMC1  EQU    $A000
           A004   PTMB   EQU    $A004
           A001   PTMC2  EQU    $A001
           0000   INDL   EQU    $00
           00E0   INDH   EQU    $E0
           0011   MEM    EQU    $11
           0012   MEM1   EQU    $12
           2001   CRA1   EQU    $2001
           2003   CRB1   EQU    $2003
           2000   DDRA1  EQU    $2000
           2002   DDRB1  EQU    $2002
           4001   CRA2   EQU    $4001
           4003   CRB2   EQU    $4003
           4000   DDRA2  EQU    $4000
           4002   DDRB2  EQU    $4002
           8000   ACIAS  EQU    $8000
           8000   ACIAC  EQU    ACIAS
           8001   ACIARD EQU    $8001
           8001   ACIAT  EQU    ACIARD
           0010   CP     EQU    $10
           000E   VALL   EQU    $0E
           000F   VALH   EQU    $0F
           000C   TMPL   EQU    $0C
           000D   TMPH   EQU    $0D
           000B   KNT    EQU    $0B
           000A   MARK   EQU    $0A
           0006   LSD3   EQU    $06
           0007   LSD2   EQU    $07
           0008   LSD1   EQU    $08
           0009   LSD    EQU    $09
           0001   SUBHI  EQU    $01
           0000   SUBLO  EQU    $00
           0003   MINHI  EQU    $03
           0002   MINLO  EQU    $02
           0004   NEG    EQU    $04
           0005   CARRY  EQU    $05
           0014   TEMP   EQU    $14
           0016   TYPE   EQU    $16
           0017   TVALL  EQU    $17
           0018   TVALH  EQU    $18
           0019   FFLAG  EQU    $19
      E000               ORG    $E000
      E000 00            FCB    $00,$5F,$00,$17,$00,$05,$00,$02
      E001 5F
      E002 00
      E003 17
      E004 00
      E005 05
```

```
E006 00
E007 02
                  ****************************************************************
                  *                                                              *
                  *         PIA,ACIA, AND PTM INITIALIZATION                     *
                  *                                                              *
                  ****************************************************************
E008 8E 0050            LDS     #STAKPT
E00B 86 00              LDA A   #$00
E00D B7 2001            STA A   CRA1       POINT CRA1 AT DDRA1
E010 B7 2003            STA A   CRB1
E013 86 FF              LDA A   #$FF
E015 B7 2000            STA A   DDRA1
E018 B7 2002            STA A   DDRB1
E01B 86 04              LDA A   #$04
E01D B7 2003            STA A   CRB1
E020 86 3C              LDA A   #$3C
E022 B7 2001            STA A   CRA1
E025 86 00              LDA A   #$00
E027 B7 4001            STA A   CRA2
E02A B7 4003            STA A   CRB2
E02D B7 4000            STA A   DDRA2
E030 B7 4002            STA A   DDRB2
E033 86 04              LDA A   #$04
E035 B7 4003            STA A   CRB2
E038 86 3C              LDA A   #$3C
E03A B7 4001            STA A   CRA2
E03D 86 03              LDA A   #$03
E03F B7 8000            STA A   ACIAC
E042 86 01              LDA A   #$01
E044 B7 8000            STA A   ACIAS
E047 B6 4002            LDA A   DDRB2
E04A 84 30              AND A   #$30
E04C 44                 LSR A
E04D 44                 LSR A
E04E 44                 LSR A
E04F 8B 00              ADD A   #INDL
E051 97 12              STA A   MEM1
E053 86 E0              LDA A   #INDH
E055 89 00              ADC A   #$00
E057 97 11              STA A   MEM
E059 DE 11              LDX     MEM
E05B EE 00              LDX     0,X
E05D 86 93              LDA A   #$93
E05F B7 A001            STA A   PTMC2
E062 86 00              LDA A   #$00
E064 B7 A000            STA A   PTMC1
E067 FF A004            STX     PTMB
E06A 86 83              LDA A   #$83
E06C B7 A001            STA A   PTMC2
                  *
                  *
                  *
                  *
                  *
```

```
                        ***********************************************************************
                        *                                                                     *
                        *          MAIN ROUTINE TO CHECK FOR VARIOUS COMMANDS &,?,#,%, ,'A'   *
                        *                                                                     *
                        ***********************************************************************
E06F BD E18F LOOKCO JSR      GETONE    GET THE NEXT CHAR FROM NS WHEN SENT
E072 81 26          CMP A    #$26
E074 26 05          BNE      CKQM
E076 BD E100        JSR      ODAT
E079 20 F4          BRA      LOOKCO
E07B 81 3F   CKQM   CMP A    #$3F
E07D 26 05          BNE      CKLB      IF NOT ? THEN CHECK IF IT IS #
E07F BD E200        JSR      RDNSND    READ  HP   AND SEND  RESULT TO N.STAR
E082 20 EB          BRA      LOOKCO
E084 81 23   CKLB   CMP A    #$23      CHECK IF CHAR EQUALS #
E086 26 05          BNE      CKREL     IF NOT "#" THEN CHECK IF RELAY COM "%"
E088 BD E400        JSR      TIMER     INITIALIZE V/F CARD & TIME IT
E08B 20 E2          BRA      LOOKCO
E08D 81 25   CKREL  CMP A    #$25      CHECK IF IT IS A RELAY COMMAND
E08F 26 05          BNE      CKVCO     IF NOT  RELAY THEN CHECK IF IT IS VCO COM
E091 BD E500        JSR      RELAY
E094 20 D9          BRA      LOOKCO
E096 81 21   CKVCO  CMP A    #$21      CHECK IF VCO COMMAND IE " "
E098 26 05          BNE      CADC      IF NOT VCO COMMAND THEN CHECK IF ADC COM
E09A BD E600        JSR      VCO
E09D 20 D0          BRA      LOOKCO
E09F 81 41   CADC   CMP A    #$41      CHECK IF A/D CONVERTER COMMAND. IE "A"
E0A1 26 CC          BNE      LOOKCO
E0A3 BD E19B        JSR      ADC
E0A6 20 C7          BRA      LOOKCO
                        ***********************************************************************
                        *                                                                     *
                        *                  E N D   OF   MAIN  ROUTINE                         *
                        *                                                                     *
                        ***********************************************************************
                        *
                        *
                        *
                        *
                        *
                        *
                        *
                        *
                        ***********************************************************************
                        *                                                                     *
                        *      ODAT SUBROUTINE  - TO ACCEPT CHARACTERS FROM N.STAR UNTIL $    *
                        *                      CONVERT TO BINARY, AND SEND RESULT TO HP       *
                        *                                                                     *
                        ***********************************************************************
E100                ORG      $E100
E100 BD E10B ODAT   JSR      NSREAD    GET CHARS FROM N.STAR UNTIL "$"
E103 96 10          LDA A    CP        IF NOTHING SENT THEN EXIT
E105 27 03          BEQ      XODAT
E107 BD E16B        JSR      STOHP     SEND VALUE IN VALL & VALH TO HP
E10A 39      XODAT  RTS
```

```
                   ****************************************************************
                   *                                                              *
                   *     NSREAD - SUBROUTINE TO READ N.STAR & CONVERT TO BINARY   *
                   *  THIS ROUTINE WILL RECEIVE ASCII DIGITS FROM THE N.STAR UNTIL A"$" *
                   *  CONVERT  TO 16 BIT BINARY & STORE RESULT IN VALL & VALH.    *
                   *                                                              *
                   ****************************************************************
E10B 86 00         NSREAD LDA A  #$00
E10D 97 10                STA A  CP
E10F 97 0E                STA A  VALL
E111 97 0F                STA A  VALH
E113 97 0C                STA A  TMPL
E115 97 0D                STA A  TMPH
E117 97 0B                STA A  KNT
E119 4C                   INC A
E11A 97 0A                STA A  MARK
E11C 8D E18F       LOOKC2 JSR    GETONE    GET THE NEXT CHAR FROM NS WHEN SENT
E11F 81 24                CMP A  #$24
E121 27 08                BEQ    CHECK     IF "$" THEN GET OUT
E123 80 30                SUB A  #$30
E125 36                   PSH A
E126 7C 0010              INC    CP
E129 20 F1                BRA    LOOKC2
E12B 96 10         CHECK  LDA A  CP
E12D 27 3B                BEQ    XNSRD
E12F 32            NEXT   PUL A
E130 D6 0B                LDA B  KNT
E132 26 0D                BNE    KNTPOS
E134 97 0E                STA A  VALL
E136 7C 000B              INC    KNT
E139 96 10                LDA A  CP
E13B 91 0B                CMP A  KNT
E13D 27 2B                BEQ    XNSRD
E13F 20 EE                BRA    NEXT
E141 5F            KNTPOS CLR B
E142 0C            X10    CLC
E143 48                   ASL A
E144 59                   ROL B
E145 97 0C                STA A  TMPL
E147 D7 0D                STA B  TMPH
E149 48                   ASL A
E14A 59                   ROL B
E14B 48                   ASL A
E14C 59                   ROL B
E14D 0C                   CLC
E14E 9B 0C                ADD A  TMPL
E150 D9 0D                ADC B  TMPH
E152 7A 000B              DEC    KNT
E155 26 EB                BNE    X10
E157 9B 0E                ADD A  VALL
E159 D9 0F                ADC B  VALH
E15B 97 0E                STA A  VALL
E15D D7 0F                STA B  VALH
E15F 7C 000A              INC    MARK
E162 96 0A                LDA A  MARK
```

```
E164 97 0B             STA A   KNT
E166 91 10             CMP A   CP
E168 26 C5             BNE     NEXT
E16A 39         XNSRD  RTS
                ***********************************************************************
                *      STOHP - SEND TO HP SUBROUTINE                                  *
                *  THIS SUBROUTINE WILL SEND 2 BYTES STORED IN VALL & VALH TO         *
                *  THE 16 BIT IN PORT OF HP - VALH TO HIGHER BYTE & VALL TO LOWER...  *
                ***********************************************************************
E16B 96 0E      STOHP  LDA A   VALL
E16D B7 2000           STA A   DDRA1
E170 96 0F             LDA A   VALH
E172 B7 2002           STA A   DDRB1
E175 02                NOP
E176 02                NOP
E177 B6 4001           LDA A   CRA2
E17A 84 F7             AND A   #$F7
E17C B7 4001           STA A   CRA2
E17F B6 4002    LOOKC3 LDA A   DDRB2
E182 84 40             AND A   #$40     CHECK FOR FLAG FROM MULTIPROGRAMMER
E184 27 F9             BEQ     LOOKC3
E186 B6 4001           LDA A   CRA2
E189 8A 08             ORA A   #$08
E18B B7 4001           STA A   CRA2     WHEN FOUND, RESET THE GATE
E18E 39         XSTOHP RTS
                ***********************************************************************
                *                                                                     *
                *    GETONE - SUBROUTINE TO GET 1 CHAR (IN REG A) FROM NS, WHEN SENT  *
                *                                                                     *
                ***********************************************************************
E18F B6 8000    GETONE LDA A   ACIAS
E192 47                ASR A
E193 24 FA             BCC     GETONE   WAIT UNTIL A CHRACTER IS SENT
E195 B6 8001           LDA A   ACIARD   READ IT INTO REG A WHEN SENT
E198 39         XGETON RTS
                ***********************************************************************
                *                                                                     *
                *      ADC - SUBROUTINE TO COLLECT 512 (1/2 K ) SAMPLES AND           *
                *      SEND THEM TO THE NORTH STAR.   THE ADC CARD IS ASSUMED TO      *
                *      BE IN SLOT 405 OF THE MULTIPROGRAMMER                          *
                *                                                                     *
                ***********************************************************************
E19B                   ORG     $E19B
E19B CE B0F0    ADC    LDX     #$B0F0   SEND CONTROL WORD "1111 0000 1011 0000"
E19E DF 0E             STX     VALL
E1A0 BD E16B           JSR     STOHP
E1A3 CE 0050           LDX     #$0050   ADC CARD IS IN SLOT 405
E1A6 DF 0E             STX     VALL
E1A8 CE 6000           LDX     #$6000
E1AB BD E16B    ANEXT  JSR     STOHP    ADDRESS SLOT 5 WITH GATE SIGNAL
E1AE B6 4000           LDA A   DDRA2    GET THE SAMPLE
E1B1 A7 00             STA A   0,X      STORE IT AT LOCATIONS 6000 ONWARDS
E1B3 08                INX
E1B4 B6 4002           LDA A   DDRB2    GET THE UPPER BYTE OF THE SAMPLE
E1B7 A7 00             STA A   0,X
```

```
E1B9 08               INX
E1BA 8C 6400          CPX     #$6400    COLLECT 512 SAMPLES IN ALL.  IE 1K BYTES
E1BD 26 EC            BNE     ANEXT
                *
                *
                *
E1BF CE 6000          LDX     #$6000    GET READY TO SEND THE SAMPLES TO NORTH STAR
E1C2 A6 00      AAGAIN LDA A  0,X       GET THE SAMPLE STORED IN MEMORY
E1C4 43               COM A             TAKE 1'S COMPLEMENT OF THE LOWER BYTE
E1C5 97 00            STA A   00
E1C7 08               INX
E1C8 A6 00            LDA A   0,X       GET THE HIGHER BYTE OF THE SAMPLE
E1CA 43               COM A
E1CB 84 0F            AND A   #$0F      MASK OFF THE HIGHER 4 BITS
E1CD 97 01            STA A   01
E1CF 08               INX
E1D0 DF 15            STX     $15       SAVE THE SAMPLE POINTER
E1D2 CE 003F          LDX     #$003F
E1D5 BD E47E          JSR     TIMOUT    WAIT FOR ACIA TO GET READY
E1D8 BD E224          JSR     WSEND     SEND THE SAMPLE
E1DB BD E209          JSR     LOOKC1    SEND A CARRIAGE RETURN
E1DE DE 15            LDX     $15
E1E0 8C 6400          CPX     #$6400
E1E3 26 DD            BNE     AAGAIN    REPEAT UNTIL ALL 512 SAMPLES ARE SENT
E1E5 39         XADC  RTS
E200                  ORG     $E200
                *******************************************************************
                *                                                                 *
                *     RDNSND - READ AND SEND TO N.STAR SUBROUTINE                 *
                *    THIS ROUTINE WILL READ HP ADDRESSED PORT AND SEND THE CORRESP *
                *  DECIMAL VALUE -(IN ASCII)- TO THE NORTH STAR FOLLOWED BY "CR"  *
                *                                                                 *
                *******************************************************************
E200 CE 0034    RDNSND LDX    #$0034    GET A DELAY COUNT
E203 BD E47E          JSR     TIMOUT    GIVE N.STAR HANDICAP OF 1/10 SEC
E206 BD E216          JSR     IDAT      READ HP AND SEND TO N.STAR
E209 B6 8000    LOOKC1 LDA A  ACIAS
E20C 46               ROR A
E20D 46               ROR A
E20E 24 F9            BCC     LOOKC1
E210 86 0D            LDA A   #$0D
E212 B7 8001          STA A   ACIAT
E215 39         XRDNSN RTS
                *******************************************************************
                *                                                                 *
                *   ROUTINE TO READ 16 BITS OF HP, CONVERT 12 BIT BINARY TO BCD TO *
                *    ASCII, AND SEND IT SERIALLY TO NORTH STAR                    *
                *                                                                 *
                *******************************************************************
E216 B6 4000    IDAT  LDA A   DDRA2     READ LOWER BYTE OF MULTIPROGRAMMER
E219 43               COM A             1'S COMPLEMENT OF LOW BYTE
E21A 97 00            STA A   SUBLO     LOW BYTE OF SUBTRAHEND
E21C B6 4002          LDA A   DDRB2     READ UPPER BYTE OF MULTIPROGRAMMER
E21F 43               COM A             1'S COMPLEMENT OF HIGH BYTE
E220 84 0F            AND A   #$0F      MASK OFF UPPER 4 BITS
```

```
E222 97 01          STA A  SUBHI    HIGH BYTE OF SUBTRAHEND
E224 CE 0006 WSEND  LDX    #LSD3
E227 86 03          LDA A  #$03
E229 C6 00          LDA B  #$00
E22B E7 00   INC    STA B  0,X      CLEAR CURRENT DIGIT
E22D 08             INX
E22E 4A             DEC A
E22F 2A FA          BPL    INC
E231 86 E8          LDA A  #$E8
E233 97 02          STA A  MINLO
E235 86 03          LDA A  #$03
E237 97 03          STA A  MINHI    LOAD 1000 INTO MINUEND
E239 BD E300 POS1   JSR    SUB16    BRANCH TO 16 BIT SUBTRACTION ROUTINE
E23C 96 04          LDA A  NEG      CHECK TO SEE IF SUBTRACTION RESULT IS NEGATIVE
E23E 81 00          CMP A  #$00
E240 26 05          BNE    NEGR1
E242 7C 0006        INC    LSD3
E245 20 F2          BRA    POS1
E247 96 00   NEGR1  LDA A  SUBLO    RESULT WAS NEGATIVE, ADD 1000 BACK
E249 8B E8          ADD A  #$E8
E24B 97 00          STA A  SUBLO
E24D 96 01          LDA A  SUBHI
E24F 89 03          ADC A  #$03
E251 97 01          STA A  SUBHI
E253 86 64          LDA A  #$64
E255 97 02          STA A  MINLO
E257 86 00          LDA A  #$00
E259 97 03          STA A  MINHI    LOAD 100 INTO MINUEND
E25B BD E300 POS2   JSR    SUB16    BRANCH TO 16 BIT SUBTRACT ROUTINE
E25E 96 04          LDA A  NEG
E260 81 00          CMP A  #$00
E262 26 05          BNE    NEGR2
E264 7C 0007        INC    LSD2
E267 20 F2          BRA    POS2     RESULT WAS POSITIVE, GO SUBTRACT ANOTHER 100
E269 86 64   NEGR2  LDA A  #$64     RESULT WAS NEGATIVE, ADD 100 BACK
E26B 9B 00          ADD A  SUBLO
E26D 97 00          STA A  SUBLO
E26F 96 01          LDA A  SUBHI
E271 89 00          ADC A  #$00
E273 97 01          STA A  SUBHI
E275 86 0A          LDA A  #$0A     LOAD 10 INTO MINUEND
E277 97 02          STA A  MINLO
E279 7F 0003        CLR    MINHI
E27C BD E300 POS3   JSR    SUB16    BRANCH TO 16 BIT SUBTRACT ROUTINE
E27F 96 04          LDA A  NEG
E281 81 00          CMP A  #$00     CHECK TO SEE IF THE RESULT WAS NEGATIVE
E283 26 05          BNE    NEGR3
E285 7C 0008        INC    LSD1
E288 20 F2          BRA    POS3     RESULT WAS POSITIVE, SUBTRACT ANOTHER 10
E28A 96 00   NEGR3  LDA A  SUBLO
E28C 8B 0A          ADD A  #$0A     RESULT WAS NEGATIVE, ADD 10 BACK
E28E 97 09          STA A  LSD      RESULT IS LSD
E290 C6 04          LDA B  #$04
E292 09      ASCII  DEX
E293 86 30          LDA A  #$30
```

```
E295 AB 00          ADD A  0,X       BCD TO ASCII
E297 A7 00          STA A  0,X
E299 5A             DEC B
E29A 26 F6          BNE    ASCII
E29C CE 0006        LDX    #LSD3     GET READY TO SEND IT BACK
E29F C6 04          LDA B  #$04
E2A1 D7 0B          STA B  KNT       INITALIZE COUNTER KNT
E2A3 A6 00   NEXT1  LDA A  0,X       COMPARE THE CHARACTER WITH "0"
E2A5 81 30          CMP A  #$30
E2A7 27 11          BEQ    ZERO
E2A9 F6 8000 LOOKC4 LDA B  ACIAS     READ ACIA STATUS REG
E2AC 56             ROR B
E2AD 56             ROR B
E2AE 24 F9          BCC    LOOKC4
E2B0 B7 8001        STA A  ACIAT     WRITE CHARACTER TO ACIA
E2B3 08             INX
E2B4 7A 000B        DEC    KNT
E2B7 26 11          BNE    PATCH     NO MORE ZEROS CAN BE SUPPRESSED
E2B9 39      XIDAT  RTS
E2BA D6 0B   ZERO   LDA B  KNT
E2BC C1 01          CMP B  #$01
E2BE 26 04          BNE    NOT
E2C0 86 30          LDA A  #$30
E2C2 20 E5          BRA    LOOKC4
E2C4 7A 000B NOT    DEC    KNT
E2C7 08             INX
E2C8 20 D9          BRA    NEXT1     ZEROS STILL BEING SUPPRESSED
E2CA A6 00   PATCH  LDA A  0,X       NC FURTHER ZERO SUPPRESSION
E2CC 20 DB          BRA    LOOKC4
E300                ORG    $E300
             ***********************************************************************
             *                                                                     *
             *      16 BIT SUBTRACTION SUBRCUTINE                                  *
             *                                                                     *
             ***********************************************************************
E300 96 03   SUB16  LDA A  MINHI
E302 36             PSH A
E303 96 02          LDA A  MINLO
E305 36             PSH A            PUT HIGH AND LOW BYTE OF MINUEND ON THE STACK
E306 7F 0005        CLR    CARRY
E309 7F 0004        CLR    NEG
E30C 43             COM A
E30D 0C             CLC
E30E 8B 01          ADD A  #$01      TWOS'S COMPLIMENT OF MINLO
E310 24 03          BCC    L1
E312 7C 0005        INC    CARRY
E315 D6 03   L1     LDA B  MINHI
E317 53             COM B
E318 DB 05          ADD B  CARRY     TWO'S COMPLIMENT OF MINHI
E31A 97 02          STA A  MINLO
E31C D7 03          STA B  MINHI
E31E 7F 0005        CLR    CARRY
E321 0C             CLC
E322 96 00          LDA A  SUBLO
E324 9B 02          ADD A  MINLO     SUBTRACT THE LOW BYTES
```

```
E326 24 03            BCC    L2
E328 7C 0005          INC    CARRY
E32B D6 01     L2     LDA B  SUBHI
E32D DB 03            ADD B  MINHI      SUBTRACT THE HIGH BYTES
E32F DB 05            ADD B  CARRY
E331 97 00            STA A  SUBLO
E333 D7 01            STA B  SUBHI      STORE THE RESULT
E335 2A 03            BPL    L3
E337 73 0004          COM    NEG        SET NEG TO FF IF THE RESULT IS NEGATIVE
E33A 32        L3     PUL A
E33B 97 02            STA A  MINLO
E33D 32               PUL A
E33E 97 03            STA A  MINHI      RETRIEVE THE ORIGINAL MINUEND
E340 39        XSUB16 RTS
E400                  ORG    $E400
               ********************************************************************
               *                                                                  *
               *      ROUTINE TO MONITOR V/F CARD FOR SPECIFIED TIME  AND READ  IT *
               *      BACK, AND SEND THE RESULT BACK TO THE NORTH STAR             *
               *                                                                  *
               ********************************************************************
E400 7F 0016   TIMER  CLR    TYPE
E403 BD E18F          JSR    GETONE     GET THE NEXT CHAR FROM NS WHEN SENT
E406 80 30            SUB A  #$30       GET THE HEX EQUIVALENT
E408 48               ASL A             AND MULTIPLY IT BY 2
E409 CE E6EE          LDX    #CADD-2     GET INTO X ADDRESS TO 1ST COUNT MINUS 2
E40C DF 14            STX    TEMP
E40E 9B 15            ADD A  TEMP+1
E410 97 15            STA A  TEMP+1
               *
E412 BD E18F          JSR    GETONE     GET THE NEXT CHAR FROM NS WHEN SENT
E415 81 30            CMP A  #$30       CHECK IF IT EQUALS ZERO
E417 27 04            BEQ    ARND
E419 86 20            LDA A  #$20
E41B 97 16            STA A  TYPE
E41D CE 60F0   ARND   LDX    #$60F0     GET THE CONTROL WORD
E420 DF 0E            STX    VALL
E422 BD E16B          JSR    STOHP      SEND VALUE IN VALL & VALH TO HP
               *
E425 86 D0            LDA A  #$D0       0D = 13 = SLOT ADDRESS FOR V/F CARD
E427 97 0F            STA A  VALH
E429 86 05            LDA A  #$05       SEND A  05 TO  V/F CARD
E42B BD E476          JSR    SEND
               *
E42E 86 01            LDA A  #$01       SEND A 01 TO V/F CARD
E430 BD E476          JSR    SEND
               *
E433 86 09            LDA A  #$09       SEND A 09 TO V/F CARD
E435 BD E476          JSR    SEND
               *
E438 DE 14            LDX    TEMP       GET THE POINTER TO THE COUNT
E43A EE 00            LDX    0,X        GET THE CORRESPONDING COUNT IN X
E43C BD E47E          JSR    TIMOUT
               *
E43F C6 05            LDA B  #$05       7060 COUNTER TO BE READ 5 TIMES
```

```
E441 D7 14              STA B   TEMP
                *
E443 86 0D      AGAIN   LDA A   #$0D        SEND A 13 TO V/F CARD
E445 BD E476            JSR     SEND
                *
E448 86 0F              LDA A   #$0F
E44A BD E476            JSR     SEND
                *
E44D CE A0F0            LDX     #$A0F0      SEND ISL AND SYE FOR READING
E450 DF 0E              STX     VALL
E452 BD E16B            JSR     STOHP       SEND VALUE IN VALL & VALH TO HP
                *
E455 CE 00D0            LDX     #$00D0      TO TRIGGER  INPUT FROM V/F CARD
E458 DF 0E              STX     VALL
E45A BD E16B            JSR     STOHP       SEND A 5 TO THE V/F SLOT
                *
E45D BD E466            JSR     READ
E460 7A 0014            DEC     TEMP
E463 26 DE              BNE     AGAIN
E465 39         XTIMER  RTS
                ************************************************************************
                *                                                                      *
                *     ROUTINE USED BY TIMER TO READ A BYTE FROM V/F CARD & SEND TO NS *
                *                                                                      *
                ************************************************************************
E466 BD E200    READ    JSR     RDNSND      TIMEOUT, READ HP AND SEND TO NSTAR
E469 CE 60F0            LDX     #$60F0
E46C DF 0E              STX     VALL
E46E BD E16B            JSR     STOHP       SEND VALUE IN VALL & VALH TO HP
E471 86 D0              LDA A   #$D0
E473 97 0F              STA A   VALH
E475 39         XREAD   RTS
                ************************************************************************
                *                                                                      *
                *      ROUTINE USED BY TIMER TO ADDRESS THE V/F SLOT                   *
                *                                                                      *
                ************************************************************************
E476 9A 16      SEND    ORA A   TYPE
E478 97 0E              STA A   VALL
E47A BD E16B            JSR     STOHP       SEND A 5 TO THE V/F SLOT
E47D 39         XSEND   RTS
                ************************************************************************
                *      TIMER SUBROUTINE TO TIMECUT FOR 1/10, U, OR  10 SEC DEPENDING   *
                *                           ON THE CONTENTS OF REG X                   *
                *                                                                      *
                ************************************************************************
E47E 86 5D      TIMOUT  LDA A   #$5D
E480 4A         BACK    DEC A
E481 26 FD              BNE     BACK
E483 09                 DEX
E484 26 F8              BNE     TIMOUT
E486 39         XTIMOU  RTS
                *
                *
                *
```

```
                         *
                         ***************************************************************************
                         *                                                                         *
                         *       RELAY  SUBROUTINE                                                 *
                         *  THIS ROUTINE WILL ESTABLISH(WRITE) -OR- READ BACK RELAY POSITIONS      *
                         * TO WRITE :- IT ACCEPTS NEW POSITION FROM N.STAR ,CONVERTS TO            *
                         *  BINARY, AND SENDS IT OUT ON SLOT A  (0001) OF HP INPUT PORT            *
                         *  TO  READ:- FETCHES CONTENTS OF SLOT A OF HP AND SENDS IT TO NS         *
                         *                                                                         *
                         ***************************************************************************
E500                     ORG     $E500
E500 BD E18F  RELAY      JSR     GETONE      GET THE NEXT CHAR FROM NS WHEN SENT
E503 81 57               CMP A   #$57        IF CHARACTER IS NOT "W" THEN READ RELAY
E505 26 20               BNE     RDREL
E507 BD E10B             JSR     NSREAD      ELSE GET NEW RELAY PCSITIONS  FROM   NS
E50A DE 0E               LDX     VALL        SAVE THE RELAY POSITIONS FOR LATER USE
E50C DF 17               STX     TVALL       IF NOTHING SENT THEN OPEN ALL RELAYS
                         *
E50E CE 70F0             LDX     #$70F0      SEND DTE SYE ANC TME TO HP
E511 DF 0E               STX     VALL
E513 BD E16B             JSR     STOHP       SEND VALUE IN VALL & VALH TO HP
                         *
E516 DE 17               LDX     TVALL       GET THE SAVED RELAY POSITIONS
E518 DF 0E               STX     VALL
E51A 96 0F               LDA A   VALH
E51C 84 0F               AND A   #$0F
E51E 8A 10               ORA A   #$10        ATTACH SLOT A ADDRESS TO DATA
E520 97 0F               STA A   VALH
E522 BD E16B             JSR     STOHP       SEND VALUE IN VALL & VALH TO HP
E525 20 13               BRA     XREL        DONE - SO EXIT FROM HERE
                         *
E527 CE A0F0  RDREL      LDX     #$A0F0      SEND ISL AND SYE TO TRIGGER HP FOR INPUT
E52A DF 0E               STX     VALL
E52C BD E16B             JSR     STOHP       SEND VALUE IN VALL & VALH TO HP
                         *
E52F CE 0010             LDX     #$0010      READ SLCT A (0001-CONTROL)
E532 DF 0E               STX     VALL
E534 BD E16B             JSR     STOHP       SEND VALUE IN VALL & VALH TO HP
E537 BD E200             JSR     RDNSND      READ RELAY AND SEND  RESULT TO N.STAR
E53A 39       XREL       RTS
E600                     ORG     $E600
                         ***************************************************************************
                         *                         V C O   SUBROUTINE                              *
                         *  THIS SUBROUTINE WILL GET THE RANGE, FREQUENCY, AND AMPLITUDE           *
                         *  FROM THE NORTH STAR AND OUTPUT CONTROL TO THE VCO CARD IN SLOT K       *
                         ***************************************************************************
E600 BD E10B  VCO        JSR     NSREAD      PERFORM GET CHARACTERS SUBROUTINE
E603 96 0E               LDA A   VALL        IF RANGE NOT BETWEEN 0 AND 3
E605 84 03               AND A   #$03        THEN RETAIN ONLY THE LEAST 2 BITS
E607 48                  ASL A
E608 48                  ASL A
E609 8A B0               ORA A   #$B0        PCSITION RANGE AND COMBINE IT WITH SLOT ADDR
E60B 97 16               STA A   TYPE        SAVE IT FOR LATER USE
E60D 7F 0019             CLR     FFLAG       FLAG INDICATING NO CHANGE IN FREQUENCY
E610 BD E10B  GETFRQ     JSR     NSREAD      GET THE MAGNITUDE OF FREQ IN BANARY
```

```
00594 E613 96 10              LDA A   CP
00595 E615 26 03              BNE     AROUN      IF NOTHING SENT THEN ASSUME
00596 E617 7C 0C19            INC     FFLAG      NO CHANGE IN FREQUENCY OF VCO
00597 E61A 96 0E      AROUN   LDA A   VALL
00598 E61C 97 17              STA A   TVALL      SAVE IT FCR NOW AND
00599 E61E BD E10B    GETAMP  JSR     NSREAD     GET THE AMPLITUDE FROM NORTH STAR
00600 E621 96 10              LDA A   CP         IF NOTHING SENT THEN
00601 E623 27 1F              BEQ     NOCHNG     DO NOT CHANGE THE AMPLITUDE OF VCO OUTPUT
00602 E625 D6 0E              LDA B   VALL       SAVE ONLY LAST 8 BITS OF AMPLITUDE
00603 E627 37                 PSH B              ON STACK
00604 E628 CE 60F0            LDX     #$60F0     SEND  SYE AND DTE TO HP
00605 E62B DF 0E              STX     VALL
00606 E62D BD E16B            JSR     STOHP      SEND VALUE IN VALL & VALH TO HP
00607                 *       .
00608 E630 33                 PUL B              GET AMPLITUDE IN VALL
00609 E631 02                 NOP                MAY REQUIRE A COM B HERE
00610 E632 D7 0E              STA B   VALL
00611 E634 96 16              LDA A   TYPE       GET SLOT ADDRESS WITH RANGE
00612 E636 8A 01              ORA A   #$01       MAKE WR(AMP) LOW
00613 E638 97 0F              STA A   VALH
00614 E63A BD E16B            JSR     STOHP      SEND VALUE IN VALL & VALH TO HP
00615                 *
00616 E63D 96 16              LDA A   TYPE
00617 E63F 97 0F              STA A   VALH       HOLD AMPLITUDE DATA
00618 E641 BD E16B            JSR     STOHP      SEND VALUE IN VALL & VALH TO HP
00619                 *
00620 E644 96 19      NOCHNG  LDA A   FFLAG
00621 E646 26 1D              BNE     XVCO       IF NO CHANGE IN FRQ THEN QUIT
00622 E648 CE 60F0            LDX     #$60F0
00623 E64B DF 0E              STX     VALL
00624 E64D BD E16B            JSR     STOHP      SEND VALUE IN VALL & VALH TO HP
00625                 *
00626 E650 96 17              LDA A   TVALL      GET THE GREQUENCY MAGNITUDE
00627 E652 43                 COM A
00628 E653 97 0E              STA A   VALL
00629 E655 96 16              LDA A   TYPE       GET SLOT ADDRESS ABD RANGEE
00630 E657 8A 02              ORA A   #$02       MAKE WR(FRQ) LOW
00631 E659 97 0F              STA A   VALH
00632 E65B BD E16B            JSR     STOHP      SEND VALUE IN VALL & VALH TO HP
00633                 *
00634 E65E 96 16              LDA A   TYPE
00635 E660 97 0F              STA A   VALH       HOLD FREQUENCY DATA
00636 E662 BD E16B            JSR     STOHP      SEND VALUE IN VALL & VALH TO HP
00637 E665 39         XVCO    RTS
00638 E6F0                    ORG     $E6F0
00639 E6F0 0288       CADD    FDB     $0288
00640 E6F2 1958               FDB     $1958
00641 E6F4 FD7D               FCB     $FD7D
00642 E7FE                    ORG     $E7FE
00643 E7FE E0                 FCB     $E0,$00
      E7FF 00
00644                         END
```

SYMBOL TABLE

STAKPT	0050	PTMC1	A000	PTMB	A004	PTMC2	A001	INDL	0000	INDH	00E0	MEM	0011	MEM1	C012	CRA1	2001
CRB1	2003	DDRA1	2000	DDRB1	2002	CRA2	4001	CRB2	4003	DDRA2	4000	DDRB2	4002	ACIAS	8000	ACIAC	8000
ACIARD	8001	ACIAT	8001	CP	0010	VALL	000E	VALH	000F	TMPL	000C	TMPH	000D	KNT	000B	MARK	000A
LSD3	0006	LSD2	0007	LSD1	0008	LSD	0009	SUBHI	0001	SUBLO	0000	MINHI	0003	MINLO	0002	NEG	0004
CARRY	0005	TEMP	0014	TYPE	0016	TVALL	0017	TVALH	0018	FFLAG	0019	LOOKC0	E06F	CKQM	E07B	CKLB	E084
CKREL	E08D	CKVCO	E096	CADC	E09F	ODAT	E100	XODAT	E10A	NSREAD	E10B	LOOKC2	E11C	CHECK	E12B	NEXT	E12F
KNTPOS	E141	X10	E142	XNSRD	E16A	STOHP	E16B	LOOKC3	E17F	XSTOHP	E18E	GETONE	E18F	XGETON	E198	ADC	E19B
ANEXT	E1AB	AAGAIN	E1C2	XADC	E1E5	RDNSND	E200	LOOKC1	E209	XRDNSN	E215	IDAT	E216	WSEND	E224	INC	E22B
POS1	E239	NEGR1	E247	POS2	E25B	NEGR2	E269	POS3	E27C	NEGR3	E28A	ASCII	E292	NEXT1	E2A3	LOCKC4	E2A9
XIDAT	E2B9	ZERO	E2BA	NOT	E2C4	PATCH	E2CA	SUB16	E300	L1	E315	L2	E32B	L3	E33A	XSUB16	E340
TIMER	E400	ARND	E41D	AGAIN	E443	XTIMER	E465	READ	E466	XREAD	E475	SEND	E476	XSEND	E47D	TIMOUT	E47E
BACK	E480	XTIMOU	E486	RELAY	E500	RDREL	E527	XREL	E53A	VCO	E600	GETFRQ	E610	AROUN	E61A	GETAMP	E61E
NOCHNG	E644	XVCO	E665	CADD	E6F0												

APPENDIX 3

Listing of the Machine Code in 2716 EPROM

000	00	5F	00	17	00	05	00	02	8E	00	50	86	00	B7	20	01
010	B7	20	03	86	FF	B7	20	00	B7	20	02	86	04	B7	20	03
020	86	3C	B7	20	01	86	00	B7	40	01	B7	40	03	B7	40	00
030	B7	40	02	86	04	B7	40	03	86	3C	B7	40	01	86	03	B7
040	80	00	86	01	B7	80	00	B6	40	02	84	30	44	44	44	8B
050	00	97	12	86	E0	89	00	97	11	DE	11	EE	00	86	93	B7
060	A0	01	86	00	B7	A0	00	FF	A0	04	86	83	B7	A0	01	BD
070	E1	8F	81	26	26	05	BD	E1	00	20	F4	81	3F	26	05	BD
080	E2	00	20	EB	81	23	26	05	BD	E4	00	20	E2	81	25	26
090	05	BD	E5	00	20	D9	81	21	26	05	BD	E6	00	20	D0	81
0A0	41	26	CC	BD	E1	9B	20	C7	FF	FF	FF	FF	FF	FF	FF	FF
0B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
100	BD	E1	0B	96	10	27	03	BD	E1	6B	39	86	00	97	10	97
110	0E	97	0F	97	0C	97	0D	97	0B	4C	97	0A	BD	E1	8F	81
120	24	27	08	80	30	36	7C	00	10	20	F1	96	10	27	3B	32
130	D6	0B	26	0D	97	0E	7C	00	0B	96	10	91	0B	27	2B	20
140	EE	5F	0C	48	59	97	0C	D7	0D	48	59	48	59	0C	9B	0C
150	D9	0D	7A	00	0B	26	EB	9B	0E	D9	0F	97	0E	D7	0F	7C
160	00	0A	96	0A	97	0B	91	10	26	C5	39	96	0E	B7	20	00
170	96	0F	B7	20	02	02	02	B6	40	01	84	F7	B7	40	01	B6
180	40	02	84	40	27	F9	B6	40	01	8A	08	B7	40	01	39	B6
190	80	00	47	24	FA	B6	80	01	39	FF	FF	CE	B0	F0	DF	0E
1A0	BD	E1	6B	CE	00	50	DF	0E	CE	60	00	BD	E1	6B	B6	40
1B0	00	A7	00	08	B6	40	02	A7	00	08	8C	64	00	26	EC	CE
1C0	60	00	A6	00	43	97	00	08	A6	00	43	84	0F	97	01	08
1D0	DF	15	CE	00	3F	BD	E4	7E	BD	E2	24	BD	E2	09	DE	15
1E0	8C	64	00	26	DD	39	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
1F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

ILLEGIBLE

THE FOLLOWING DOCUMENT (S) IS ILLEGIBLE DUE TO THE PRINTING ON THE ORIGINAL BEING CUT OFF

ILLEGIBLE

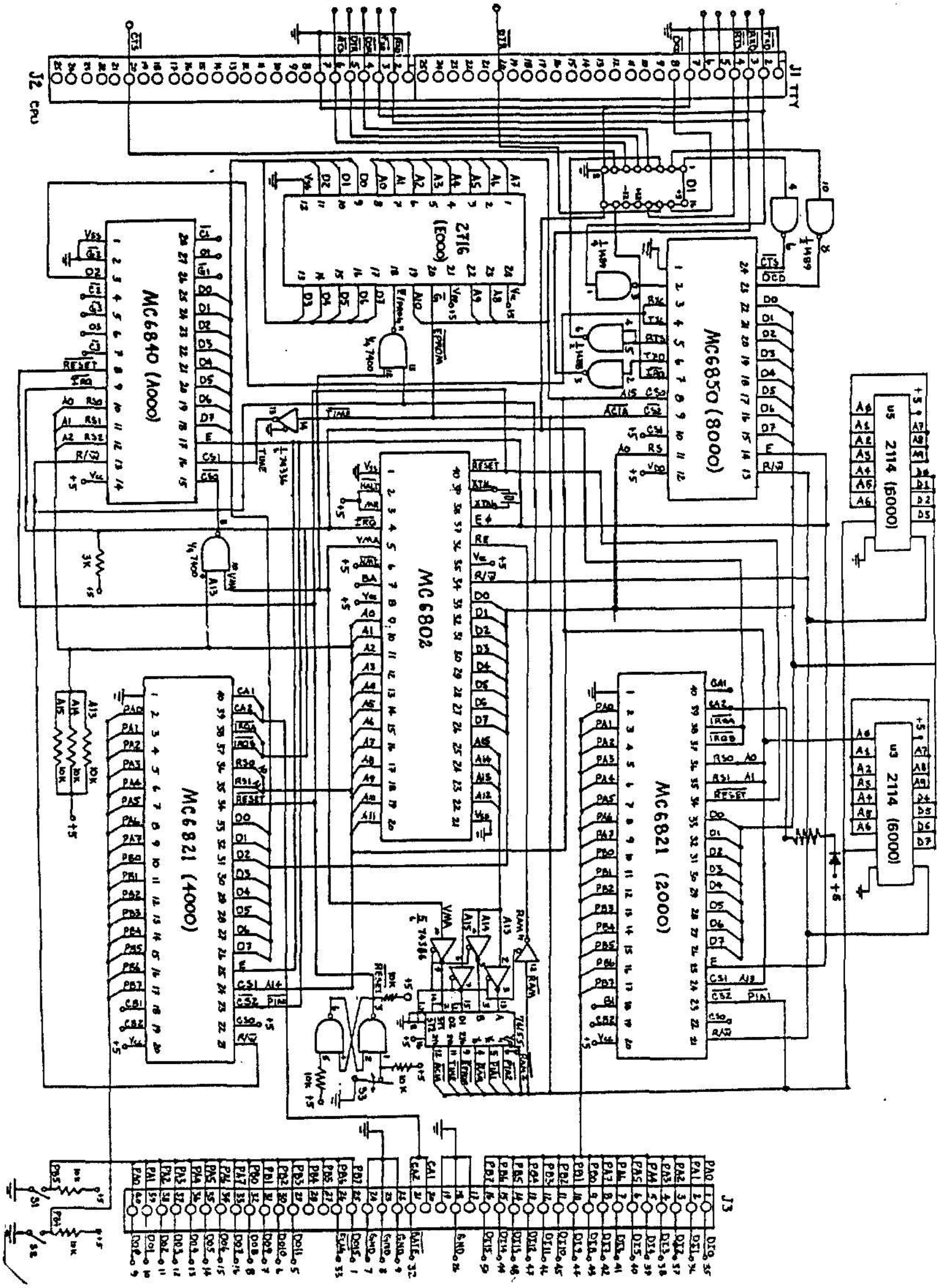
200	CE	00	34	BD	E4	7E	BD	E2	F6	B6	80	00	46	46	24	F9
210	86	0D	B7	80	01	39	B6	40	00	43	97	00	B6	40	02	43
220	84	0F	97	01	CE	00	06	86	03	C6	00	E7	00	08	4A	2A
230	FA	86	E8	97	02	86	03	97	03	BD	E3	00	96	04	81	00
240	26	05	7C	00	06	20	F2	96	00	8B	E8	97	00	96	01	89
250	03	97	01	86	64	97	02	86	00	97	03	BD	E3	00	96	04
260	81	00	26	05	7C	00	07	20	F2	86	64	9B	00	97	00	96
270	01	89	00	97	01	86	0A	97	02	7F	00	03	BD	E3	00	96
280	04	81	00	26	05	7C	00	08	20	F2	96	00	8B	0A	97	09
290	C6	04	09	86	30	AB	00	A7	00	5A	26	F6	CE	00	06	C6
2A0	04	D7	0B	A6	00	81	30	27	11	F6	80	00	56	56	24	F9
2B0	B7	80	01	08	7A	00	0B	26	11	39	D6	0B	C1	01	26	04
2C0	86	30	20	E5	7A	00	0B	08	20	D9	A6	00	20	DB	FF	FF
2D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
2E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
2F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
300	96	03	36	96	02	36	7F	00	05	7F	00	04	43	0C	8B	01
310	24	03	7C	00	05	D6	03	53	DB	05	97	02	D7	03	7F	00
320	05	0C	96	00	9B	02	24	03	7C	00	05	D6	01	DB	03	DB
330	05	97	00	D7	01	2A	03	73	00	04	32	97	02	32	97	03
340	39	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
350	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
360	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
370	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
380	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
390	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
3A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
3B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
3C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
3D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
3E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
3F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

400	7F	00	16	BD	E1	8F	80	30	48	CE	E6	EE	DF	14	9B	15
410	97	15	BD	E1	8F	81	30	27	04	86	20	97	16	CE	60	F0
420	DF	0E	BD	E1	6B	86	D0	97	0F	86	05	BD	E4	76	86	01
430	BD	E4	76	86	09	BD	E4	76	DE	14	EE	00	BD	E4	7E	C6
440	05	D7	14	86	0D	BD	E4	76	86	0F	BD	E4	76	CE	A0	F0
450	DF	0E	BD	E1	6B	CE	00	D0	DF	0E	BD	E1	6B	BD	E4	66
460	7A	00	14	26	DE	39	BD	E2	00	CE	60	F0	DF	0E	BD	E1
470	6B	86	D0	97	0F	39	9A	16	97	0E	BD	E1	6B	39	86	5D
480	4A	26	FD	09	26	F8	39	FF	FF	FF	FF	FF	FF	FF	FF	FF
490	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
500	BD	E1	8F	81	57	26	20	BD	E1	0B	DE	0E	DF	17	CE	70
510	F0	DF	0E	BD	E1	6B	DE	17	DF	0E	96	0F	84	0F	8A	10
520	97	0F	BD	E1	6B	20	13	CE	A0	F0	DF	0E	BD	E1	6B	CE
530	00	10	DF	0E	BD	E1	6B	BD	E2	00	39	FF	FF	FF	FF	FF
540	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
550	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
560	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
570	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
580	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
590	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

600	BD	E1	0B	96	0E	84	03	48	48	8A	B0	97	16	7F	00	19
610	BD	E1	0B	96	10	26	03	7C	00	19	96	0E	97	17	BD	E1
620	0B	96	10	27	1F	D6	0E	37	CE	60	F0	DF	0E	BD	E1	6B
630	33	02	D7	0E	96	16	8A	01	97	0F	BD	E1	6B	96	16	97
640	0F	BD	E1	6B	96	19	26	1D	CE	60	F0	DF	0E	BD	E1	6B
650	96	17	43	97	0E	96	16	8A	02	97	0F	BD	E1	6B	96	16
660	97	0F	BD	E1	6B	39	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
670	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
680	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
690	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
6A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
6B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
6C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
6D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
6E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
6F0	FF	88	19	58	FD	7D	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
700	02	88	19	58	FD	7D	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
710	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
720	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
730	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
740	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
750	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
760	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
770	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
780	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
790	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
7A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
7B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
7C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
7D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
7E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
7F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	E0	00

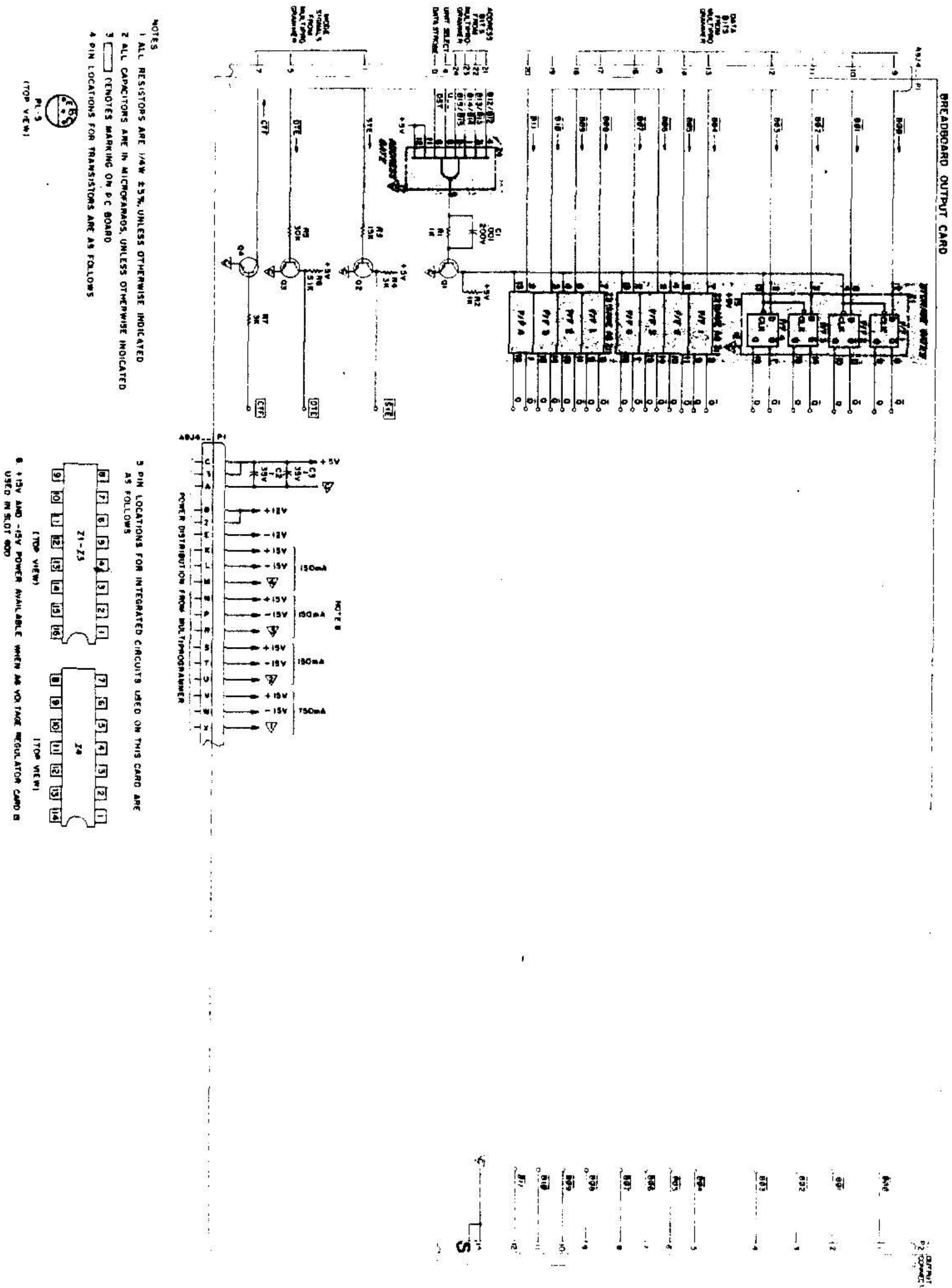
APPENDIX 4

Schematic diagram of the M6802 Based Interface



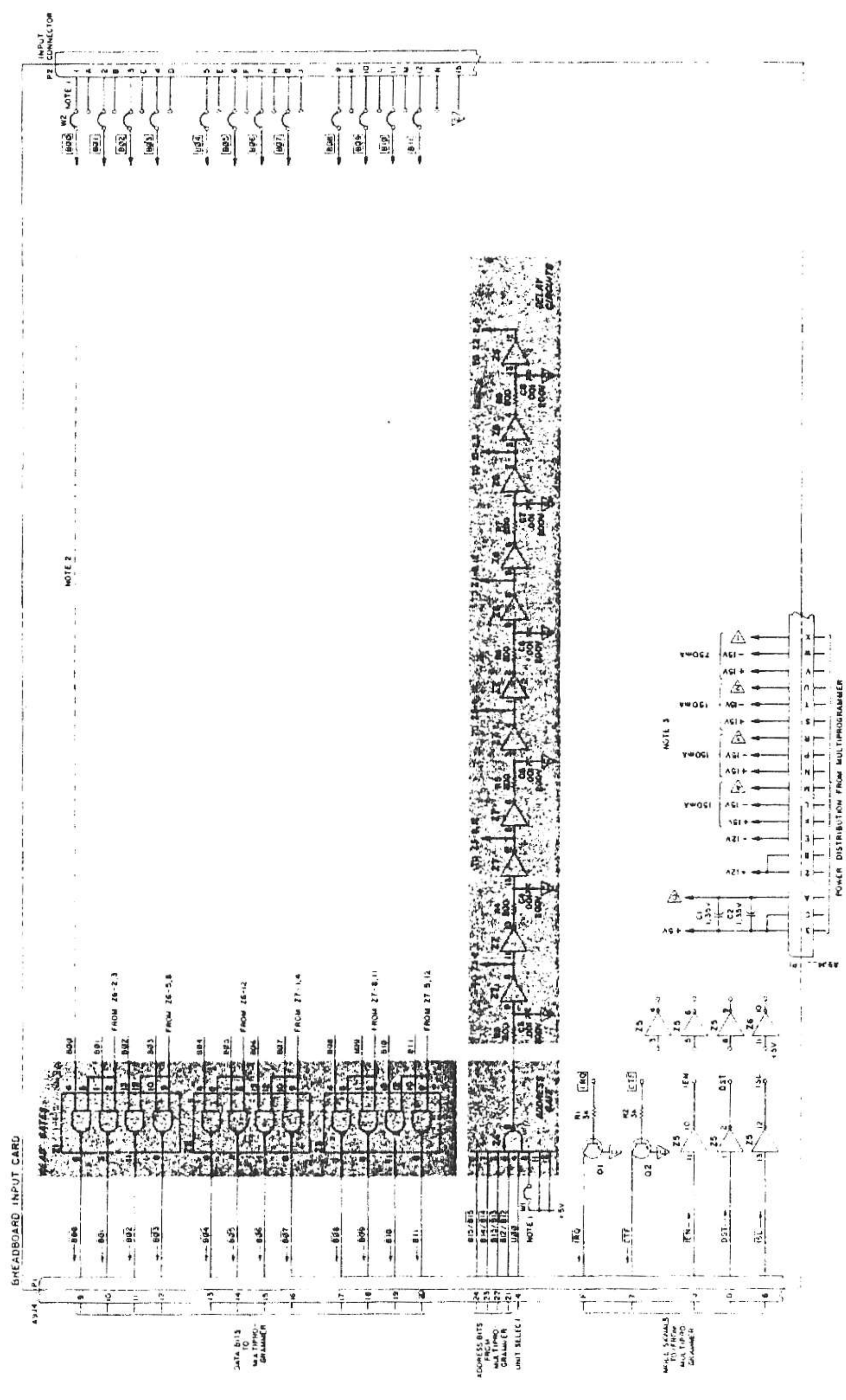
APPENDIX 6

Schematic diagram of the Breadboard Output Card 69380A



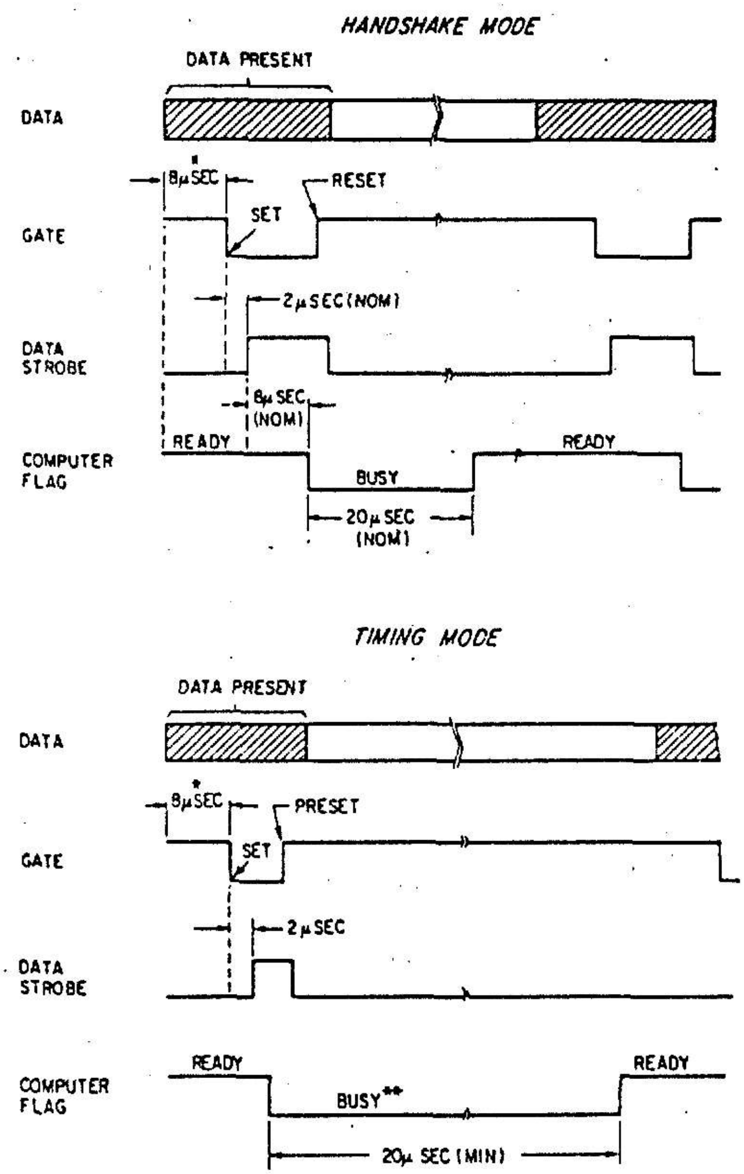
APPENDIX 5

Schematic diagram of the Breadboard Input Card 69480A



APPENDIX 7

Timing diagram of the HP Multiprogrammer I/O



* SOFTWARE DELAY

** DURATION OF BUSY FLAG DEPENDS ON THE TIMING CIRCUITS ON THE INDIVIDUAL OUTPUT CARDS OR RELATED EXTERNAL DEVICES. THE 6940B ENSURES A MINIMUM DURATION OF 20μSEC FOR THE BUSY FLAG.

APPENDIX 8

Control Mode Signals of the HP Multiprogrammer

SYE This SYSTEM ENABLE signal controls the output of the output cards. It is used as a part of an initial turn-on procedure. When the Multiprogrammer is first turned on (and before SYE is programmed on) the output circuits of all output cards are disabled, resistance output shorted, voltage outputs held at zero, and digital outputs held in the open or zero state. This feature protects the external system from potentially damaging outputs from random states at system turn on. So, while the output circuits are still disabled, each output card can be addressed and loaded with data representing a set of initial conditions before turning SYE on.

DTE This DATA TRANSFER ENABLE mode applies to cards having either external gate/flag capability or dual rank storage. It serves to strobe data into external devices and is used for simultaneous transfer of data from all output cards to the external system. Data is gated only when DTE is turned on.

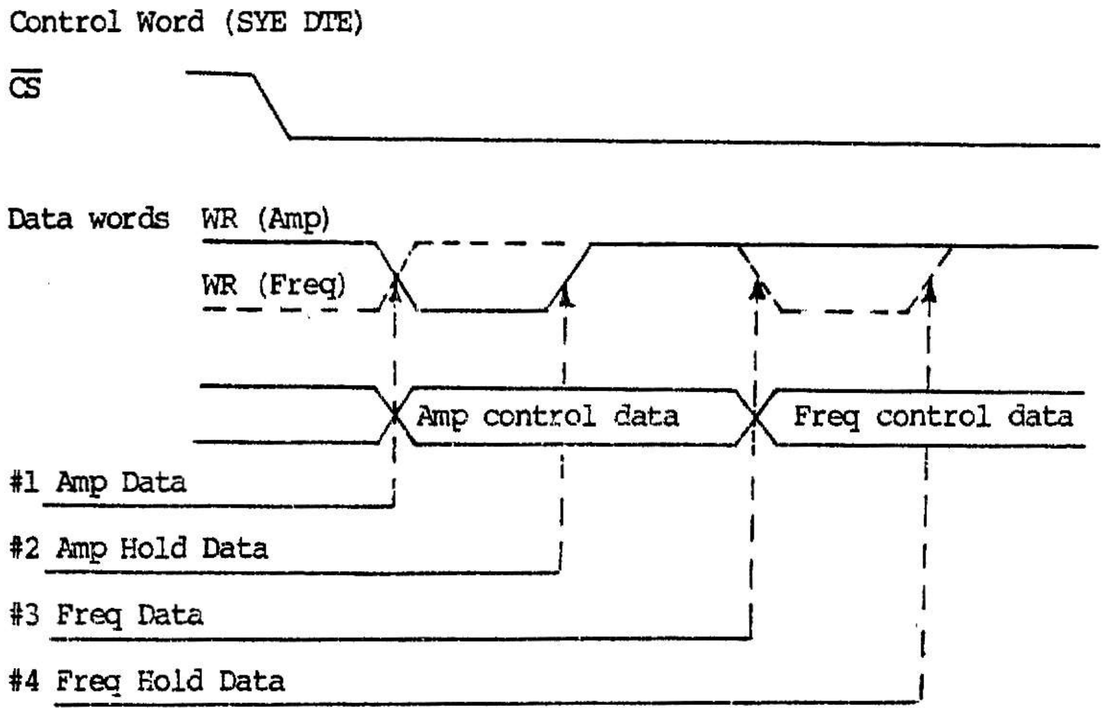
TME This TIMING MODE signal specifies whether or not the computer data output will wait for the addressed output card to signal ready for more data. This is done by controlling the Multiprogrammer flag.

ISL This INPUT SELECT mode signal, when programmed on, enables the Multiprogrammer return data lines (15 and 0 to 11) to carry subsequently addressed input card data and card input request identification - IRQ (if applicable) - on bit 15. Also, for the input cards that interface data from an external device to the computer, ISL is used to individually activate a card for data input operations with its external device.

IEN This INTERRUPT ENABLE mode signal establishes the interrupt search mode of operation, in which several input cards (having interrupt capability) can interface asynchronously with the computer. It can also be used to activate a group of input cards (if a hardware option is installed on the card), thereby eliminating the need for individually addressing input cards in the ISL mode to activate them.

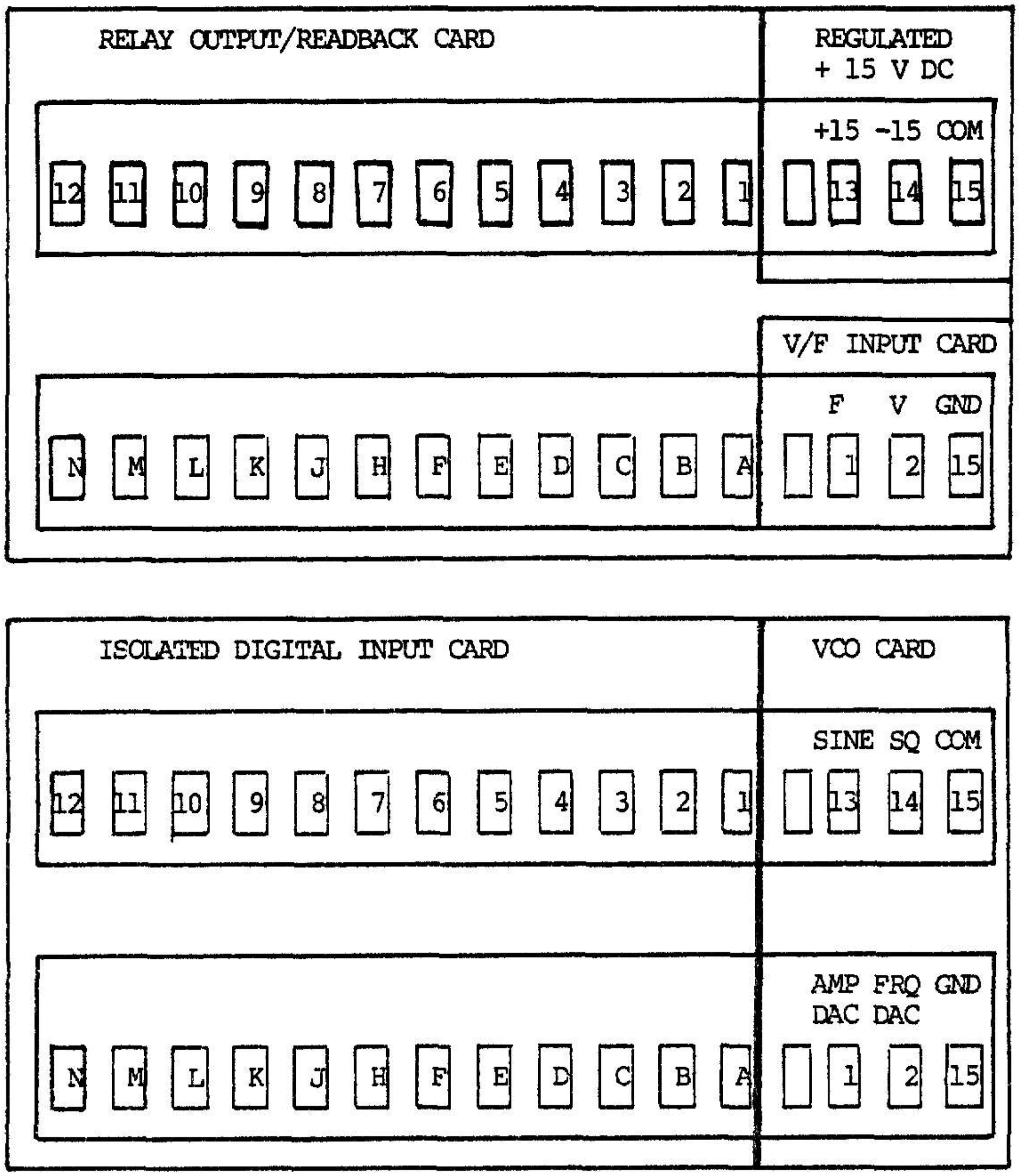
APPENDIX 9

VCO-DAC Interface Timing Diagram.



APPENDIX 10

Connector Block Terminal Assignments



APPENDIX 11

Command formats for the V/F, VCO, RELAY, A/D, and ISO cards

List of commands available to control communications with V/F, VCO, RELAY, A/D and ISO cards are explained and illustrated here in Appendix 11. Alternate procedure for programming these cards using only "&" and "?" commands is given in Appendix 12.

A. V/F CARD COMMAND.

command

```
PRINT#1 "#sV"
FOR L=0 TO 4
INPUT#1  D$
D(L)= (VAL(D$)-256-512*V)
NEXT L
```

where s = 1, 2, or 3 corresponding to a time period of 1/10th of a sec, 1 sec, or 10 seconds respectively, for which the voltage or frequency (as determined by **V**) will be monitored for measurement.

V = 1 for Voltage measurement. (voltage source must be connected between the correct pins on the terminal block)

0 for Frequency measurement. (freq source must be connected between the correct pins on the terminal block)

note The code following the command is essential as the 6802 will read back the counter on the V/F card and send the 5 bytes to the North Star at the end of the specified period. Therefore if the code to accept these bytes does not immediately follow this command then the count bytes sent by the 6802 may be missed by the North Star.

Illustration 1:

```
100  PRINT#1 "#21"
200  FOR L=0 TO 3
250  INPUT#1, D$
300  D(L)=(VAL(D$)-256-512)
400  NEXT L
500  C3=4025
600  V=(D(0)+D(1)*256+D(2)*256*256+D(3)*256*256*256)/C3
700  V=INT((V+.0005)*1000)/1000
800  PRINT "VOLTAGE = ",V," VOLTS."
900  END
```

This program will measure voltage of the source at the V/F voltage input terminals on terminal block, after monitoring the voltage for 1 sec (**s=2**), and print out the measured voltage. C3 is the calibration factor corresponding to voltage measurement for 1 second.

Illustration 2:

```
100 REM ******************************************************
101 REM * V/F PROGRAM TO MEASURE VOL  OR FREQUENCY ON THE V/F CA
102 REM * VERSION 3.0  WRITTEN BY N.FERNANDES.
104 REM * DATE WRITTEN 1-26-83
105 REM ******************************************************
430 REM
440 DIM D(4)
450 INPUT "VOLTAGE (V) OR FREQUENCY (F) ",B$
460 IF ASC(B$)=86 THEN O=1 ELSE O=0
470 GOSUB 510
480 IF O=0 THEN PRINT " FREQUENCY =  ",    ELSE PRINT "VOLTAGE =
481 PRINT INT((F+.0005)*1000)/1000,
482 IF O=0 THEN PRINT "   KHZ.  "          ELSE PRINT "   VOLTS.
493 INPUT "MORE MEASUREMENTS?-VOLTAGE(V)-FREQUENCY(F)-NONE(N).",
495 IF ASC(B$)=86 OR ASC(B$)=70 THEN 460
499 END
500 REM******************************************************
501 REM**  READ SUBROUTINE TO READ THE MULTIPROGRAMMER
502 REM**  O=1 FOR VOLTAGE & O=0 FOR FREQ-IS THE PARAMETER TO BE
504 REM**       PASSED TO THE SUBROUTINE.
505 REM**  VOLTAGE OR FREQUENCY IS PASSED BACK THROUGH  F...
506 REM******************************************************
507 REM
510 INPUT "1/10TH OF A SEC (1) OR 1 SEC (2) OR 10 SEC (3) ? ",A$
520 IF ASC(A$)<49 OR ASC(A$)>51 THEN 510
530 IF O=0 THEN A$=A$+"0"     \  C3=399.21
540 IF O=1 THEN A$=A$+"1"     \  C3=402
550 IF ASC(A$)=49 THEN IF O=0 THEN C3=399.21    ELSE C3=402
560 IF ASC(A$)=50 THEN IF O=0 THEN C3=3992.5    ELSE C3=4025
570 IF ASC(A$)=51 THEN IF O=0 THEN C3=39925     ELSE C3=40250
630 REM
640 PRINT#1,"#",A$
650 FOR L=0 TO 4
660 INPUT#1,D$
820 D(L)=VAL(D$)
835 NEXT L
840 REM **** CHECK FOR FORMAT AND SEQUENCE ERRORS  ***
850 IF D(4)>=2^8+2^9*O THEN PRINT "**SEQUENCE ERROR**"
860 FOR I=0 TO 3
870 IF D(I)<2^8+2^9*O THEN PRINT "** DATA FORMAT ERROR **"
880 NEXT I
890 D(0) = D(0)-(256+512*O)
891 D(1) = D(1)-(256+512*O)
892 D(2) = D(2)-(256+512*O)
893 D(3) = D(3)-(256+512*O)
920 F=((D(0)+D(1)*256+D(2)*256*256+D(3)*256*256*256)/C3)
970 RETURN
```

D. <u>VCO COMMAND FORMAT</u>

command **PRINT#1 "!rfffaaa$"**

where r = a number from 0 to 3 for ranges 0,1,2,3 respectively. If **r** is any other value, then only the 2 least significant bits of the resulting binary number will decide the range.

fff= ASCII coded decimal number from 0 to 255, corresponding to the relative frequency within the selected range. **fff** = 0 is the lower end and **fff** = 255 is the upper end frequency of the range selected by **r.** If **fff** is outside the range then the resulting binary number will be truncated to 8 bits, and used for relative frequency within that range.

aaa= ASCII coded decimal number from 0 to 255 for the relative amplitude of the VCO output (**aaa**=0 for min and **aaa**=255 for max amplitude). The 8 bit truncated result will be used if **aaa** is outside the range.

$ = Delimiter for **r, fff,** and **aaa.** The delimiters are mandatory but **r, fff,** and **aaa** are optional and if omitted then that parameter will not be changed from its previous value.

Illustration: The following program will output a waveform continuously varying in frequency from 0 to max. Try it out and observe the wave form on the oscilloscope.

```
100    FOR R=0 TO 3
110    FOR F=0 TO 255
120    R$  =  STR$(R)  \    R$=R$(2)+"$"
130    F$  =  STR$(F)  \    F$=F$(2)+"$"
140    PRINT#1, "!",R$,F$,"255$"
150    NEXT F
160    NEXT R
170    END
```

C. <u>RELAY CARD COMMAND.</u>

command 1 PRINT#1 "%Wdddd$"

This command will close (W for write) the relays whose positions are represented by 1's in the binary equivalent of the decimal number **dddd**, and it must be in the range 0 to 4095. If **dddd** is outside the range then the 12 bit truncated result will be used to represent the relay positions.

command 2 PRINT#1 "%R",D$

This command will read (R for read) the current relay settings. The result is obtained in D$ as an ASCII coded decimal number whose binary equivalent indicates the corresponding relay settings (1=closed, 0=open).

illustration 1:

```
100  INPUT "GIVE THE DECIMAL NO TO CLOSE DESIRED RELAYS",N
200  N$=STR$(N)      \      N$=N$(2)+"$"
300  PRINT#1 "%W",N$
400  END
```

This program will close the relays whose positions are determined by the 1's in the binary equivalent of the number N, and open the remaining relays. For eg if N=3 the the first 2 relays will be closed. This is confirmed by the first 2 bits on the HP multiprogrammer panel being turned on. The following program will illustrate the command to read back the existing relay settings. Therefore if executed immediately after executing the above program then it should print out the same number that was input to the first program.

Illustration 2:

```
500  INPUT#1 "%R",D$
600  PRINT   " RELAY POSITIONS IN DECIMAL = ",D$
700  END
```

Illustration 3:

```
100 REM*** TITLE OF PROGRAM --- RELAY1
170 REM ***************************************************************
180 REM *THIS PROGRAM WILL CONTROL THE RELAY OUTPUT/READBACK CARD. *
190 REM *THE OUTPUT RELAYS K1A1,K2A2,...,K12A12 MAY BE TURNED ON IN*
200 REM *ANY ORDER. THE USER MAY SET THE OUTPUT PATTERN IN A BINARY*
210 REM *OR BASE 10 FORMAT. THE OUTPUT DATA WILL ALSO BE READ BACK.*
220 REM ***************************************************************
230 REM
432 DIM B$(13)
620 INPUT "GIVE RELAY POSITIONS. IF IN BINARY, PRECEDE IT WITH %",B$
640 IF B$(1,1)="%" THEN GOSUB 6000  \REM **CHECK FOR BINARY & FIX IT
660 IF VAL(B$))4095 OR VAL(B$)(0 THEN PRINT "DATA OUT OF RANGE."
665 IF VAL(B$))4095 OR VAL(B$)(0 THEN 620
680 A$=STR$(VAL(B$))
690 A$=A$(2)                          \REM **REMOVE LEADING SPACE
700 PRINT#1,"%W",A$,"$"
710 INPUT#1,"%R",B$
820 D$=B$
830 PRINT "RELAY POSITION IN BASE TEN.  =  ",B$
840 GOSUB 6500                        \REM **CONVERT B$ TO BINARY
870 PRINT "RELAY POSITION IN BINARY.    =  %",B$
880 INPUT   "DO YOU WANT TO PLAY AROUND MORE (Y/N)",D$
881 IF ASC(D$)=89 THEN 620
900 END                               \REM **END OF PROGRAM**
6000 REM **************************************************************
6020 REM *    ROUTINE TO CONVERT BINARY STRING TO A DECIMAL NUMBER *
6030 REM **************************************************************
6050 B$=B$(2)\ T=0                    \REM **REMOVE % MARKER
6060 FOR I=0 TO LEN(B$)-1
6070     IF B$(LEN(B$)-I,LEN(B$)-I)="1" THEN T=T+2^I
6080     NEXT I
6090 B$=STR$(T)
6110 B$=B$(2)                         \REM **REMOVE LEADING SPACE
6120 RETURN
6500 REM **************************************************************
6520 REM *   ROUTINE TO CONVERT DECIMAL NUMBER TO A BINARY STRING   *
6530 REM **************************************************************
6540 T=VAL(B$)
6545 B$=""
6560 FOR I=11 TO 0 STEP -1
6570     IF (2^I)(=T THEN B$=B$+"1"
6580     IF (2^I))T THEN 6610
6590         T=T-2^I
6600         GOTO 6620
6610     B$=B$+"0"
6620     NEXT I
6630 RETURN
```

D. <u>COMMAND FORMAT FOR THE A/D CARD</u>

Command **PRINT#1,"A"**

```
DIM D(512)
FOR L=1 TO 512
INPUT#1,D$
D(L)=VAL(D$)
NEXT L
```

This command will cause the M6802 to collect 512 samples of the signal at the high speed Analog to Digital conversion card. These samples are then sent to the North Star, in form of 12 bit binary words. The code following the command receives the samples and stores them in the array D.

illustration:

```
100 DIM D(512)
110 PRINT#1,"A"
120 FOR L=1 TO 512
130  INPUT#1,D$
140  D(L)=VAL(D$)
150 NEXT L
160 FOR L=1 TO 512
170 IF D(L) < 2048 THEN 190
180  D(L)=D(L)-4096
190 D(L)=D(L)*.005  \ REM ASSUME A RENGE OF + 10 VOLTS
200 PRINT L,"TH SAMPLE  = ",D(L)," ",
210 NEXT L
220 END
```

The above program illustrates the use of the command to collect 512 samples through the fast A/D card, and the calculation of the voltage value from the 12 bit binary word returned for each sample. The sampling rate is 9120 samples per second. The user may choose to receive only the required number of signals and ignore the rest. In this case a dummy loop will be required to clear up the input port, as shown below.

```
300 FOR L=1 TO 10
310  INPUT#1,D$
320 NEXT L
```

E. ISO Card Programming

The Isolated Digital Input card can be read following the 3-step procedure outlined below.

STEP 1. Output a control word with ISL and SYE on. This configures the Multiprogrammer for input mode.

```
100     PRINT#1, "&", "61600", "$" \ REM CONTROL WORD "F0A0" IN HEX
```

STEP 2. Address the slot holding the Isolated Digital input card.

```
200     PRINT#1, "&", "0", "$"     \ REM CARD IS IN SLOT 400
```

STEP 3. Read back the data on the return data lines.

```
300     INPUT#1, "?",D$
310     D=VAL(D$)
```

D is now the equivalent decimal number which can be converted to binary to give the state of the digital inputs.

APPENDIX 12

Alternate Programming for the V/F, VCO, RELAY and A/D cards

A. ALTERNATE PROGRAMMING FOR V/F CARD

The V/F card can alternately be programmed for frequency measurement by the following 8 step procedure. For Voltage measurement "32" has to be added to all the output control words. Here the card is assumed to be in slot 413 (the slot address = "1101" in binary, "53248" in decimal).

STEP 1. send a "5" (decimal 53248+5) to the card using the following code. This enables the scan counter and sets the counter so it can be cleared with the next set of data.

```
100    PRINT#1,"&","61536","$"  \REM SEND CONTROL WORD="F060" (HEX)
150    PRINT#1, "&","53253","$"  \REMSEND5 TOSLOT413(IE 53248+5)
```

STEP 2. Send a "1" (decimal 53248+1) to the card. This resets the scan counter.

```
200    PRINT#1,"&","61536","$"  \REM SEND CONTROL WORD="F060" (HEX)
250    PRINT#1, "&","53249","$"  \REMSEND1 TOSLOT413(IE 53248+1)
```

STEP 3. Send a "9" (decimal 53248+9) to the card. This lets the pulses be counted at the frequency input.

```
300    PRINT#1,"&","61536","$" \REM SEND CONTROL WORD="F060" (HEX)
350    PRINT#1, "&","53257","$"  \REMSEND 9TOSLOT413(IE 53248+9)
```

STEP 4. Wait for a specified period of time for the counter to count.

```
400    FOR L=0 TO 1000
450    NEXT L
```

STEP 5. Send a "13" (decimal 53248+13) to the card. This latches the data from the counter so it may be read out.

```
500    PRINT#1,"&","61536","$"  \REM SEND CONTROL WORD="F060" (HEX)
550    PRINT#1,"&","53261","$"  \REMSEND13 TOSLOT413(53248+13)
```

STEP 6. Send a "15" (decimal 53248+15) to the card. This enables the counter output so that the data may be read.

```
600    PRINT#1,"&","61536","$"  \REM SEND CONTROL WORD="F060" (HEX)
650    PRINT#1,"&","53263","$"  \REM SEND 15 TO SLOT 413 (53248+15)
```

STEP 7. Send control word for input to the V/F card. Gate the input card and read data. Bit 8 will be "1", indicating scan not finished. Bit 9 will be "0", indicating frequency input is on.Bits 10 & 11 tied low. Subtract 256 from this value to getthe 8 bit value. If voltage input is on then 512 also needs to be subtracted.

```
700    PRINT#1,"&","61600","$"  \REM SEND INPUT CONTROL WORD"F0A0"
710    INPUT#1, "?",D$
720    D(L) = VAL(D$) - 256
```

STEP 8. Repeat steps 5 through 7, 5 times. At the fifth time Bit 8 will be zero indicating that the scan is over.

B. <u>ALTERNATE PROGRAMMING FOR THE VCO CARD.</u>

The VCO card can alternately be programmed to produce a desired waveform by following the 5 Step procedure outlined below. Here the VCO card is assumed to be placed in slot 411 (K) (ie the slot address = "1011" in binary, "45056" in decimal).

STEP 1. Set bits 10 and 12 depending on the range selected.

```
100   INPUT "RANGE ? (0,1,2, or 3) ",R
150   R=INT(R)
160   IF R=1 THEN R=1024   \ REM Set Bit 10 = 1 Bit 11 = 0
170   IF R=2 THEN R=2048   \ REM Set Bit 10 = 0 Bit 11 = 1
180   IF R=3 THEN R=3072   \ REM Set Bit 10 = 1 Bit 11 = 1
```

STEP 2. Accept the AMPLITUDE from the user and send it to the HP.

```
200   INPUT "GIVE AMPLITUDE ? (0 TO 256) ",A
210   A=INT(A)
220   PRINT#1,"&","61536","$" \REM Output Control word "F060"
250   PRINT#1,"&",STR$(45056+256+R+A),"$"
```

STEP 3. Hold the AMPLITUDE data.

```
300   PRINT#1,"&","61536","$"  \REM Output Control word "F060"
350   PRINT#1,"&",STR$(45056+R+A),"$"
```

STEP 4. Accept the FREQUENCY from the user and send it to the HP.

```
400   INPUT "GIVE FREQUENCY ? (0 TO 256) ",F
410   F=INT(F)
420   PRINT#1,"&","61536","$" \REM Output Control word "F060"
450   PRINT#1,"&",STR$(45056+256+R+F)
```

STEP 5. Hold the FREQUENCY data.

```
500   PRINT#1,"&","61536","$" \REM Output Control word "F060"
550   PRINT#1,"&",STR$(45056+F+A),"$"
```

C. ALTERNATE PROGRAMMING FOR THE RELAY CARD

If the relay card is placed in any other slot, other than slot 401 (A), then the commands "%W" and "%R" cannot be used to control the relay card. In this case the user is required to follow the procedure outlined below.

TO CLOSE THE REQUIRED RELAYS.

STEP 1. Accept the user data - a decimal number corresponding to the required 12 bit Binary pattern (with a "1" for a closed relay, and a "0" for an open relay).

```
100    INPUT#1, R
120    R=INT(R)
```

The user may be permitted to enter the binary pattern itself, in which case, the program will have to convert the binary number to a decimal number to be sent to the M6802.

STEP 2. Send the control word with DTE, SYE, and TME on, followed by the corresponding data word with slot address.

```
200    PRINT#1,"&","61552","$"
210    PRINT#1,"&",STR$(R+4096)\REMRELAY CARDINSLOT 401,(4096)
```

TO EXAMINE THE RELAY STATUS.

STEP 1. set the multiprogrammer for input mode by sending a control word with ISL, and SYE on.

```
100    PRINT#1,"&","61600","$"
```

STEP 2. Trigger the multiprogrammer for input from slot 401 (A) by an address word, and read the corresponding return data word and examine it.

```
200    PRINT#1,"&","4096","$"  \REM SLOT ADDRESS = 4096 (slot 401)
210    INPUT#1,"?",D$
220    D=VAL(D$)
```

D is the decimal value of the corresponding Binary number represinting the relay status ("1" for a closed, and "0" for a open relay.

D. ALTERNATE PROGRAMMING FOR THE A/D CARD

If the A/D card is not placed in slot 405, then the command "A", described in Appendix 11, cannot be used to measure voltage signals through the A/D card. In this case the card will have to be addressed directly, through BASIC, and the samples collected by following the procedure outlined below. However, the rate of sampling, using this procedure, will be very slow.

STEP 1. Send a control word with ISL, SYE, and TME on.

```
100 PRINT#1,"&","61616","$" \ REM CONTROL WORD "F0B0" IN HEX
```

STEP 2. Send the address word corresponding to the slot holding the A/D card.

```
200 PRINT#1,"&","20480","$" \ REM ASSUME CARD IS IN SLOT 405
```

STEP 3. Read back the return data and convert the binary word to the corresponding voltage value.

```
300 INPUT#1,"?",D$
310 D=VAL(D$)
320 IF D < 2048 THEN 340
330 D=D-4096
340 D=D*.005          \REM ASSUMING THE RANGE SETTING IS ±10V
```

D now represents the value of the signal, present at the A/D card input at the time of sampling, in volts.

APPENDIX 13

Automated Diagnostics Program Listing.

```
100 REM ************************************************************
110 REM *DIAGNOSTICS PROGRAM TO TEST FOR THE PROPER FUNCTIONING OF*
112 REM *-THE RELAY CARD, THE V/F CARD, THE VCO CARD, THE ADC CARD*
114 REM *AND THE ISOLATED DIGITAL INPUT CARD  -.  BEFORE EXECUTING*
116 REM *PROGRAM, THE DIAGNOSTIC MODULE MUST BE MOUNTED ON THE    *
118 REM *CONNECTOR BLOCK.........................................*
120 REM ************************************************************
130   N=2048          \   DIM B$(13)
140   GOSUB 1000      \   REM CLOSE RELAY 11
150   IF E = 0  THEN   200
160     PRINT "*****      ABORTING DIAGNOSTICS      *****"
170     GOTO   990
200   O=0
210   GOSUB 2000      \   REM MEASURE FREQUENCY FROM NORTH STAR
220   IF F>1.745  AND F<1.765 THEN GOTO 250
230     PRINT "FREQ MEASURED BY V/F CARD ",F, " KHZ IS NOT OK"
231     PRINT "IT SHOULD HAVE BEEN BETWEEN 1.745 AND 1.755 KHZ"
240     GOTO 160
250   N=1024
252   GOSUB 1000
254   IF E<>0 THEN GOTO 160
258   O=1
260   GOSUB 2000
270   IF F>3.6 AND F<3.7 THEN 300
280     PRINT "VOLTAGE MEASURED BY V/F CARD ",F," VOLTS IS NOT OK"
281     PRINT "IT SHOULD HAVE BEEN BETWEEN 3.6 AND 3.7 VOLTS"
290     GOTO 160
300 REM ************************************************************
301 REM *  TESTING VCO - SINE (FREQ & VOLTS) AND SQ (FREQ & VOLTS)*
302 REM ************************************************************
310   PRINT#1, "!2$255$25E$"         \ REM OUTPUT A WAVEFORM ON VCO
320   N=512                          \ REM CLOSE RELAY 9
321   GOSUB 1000
322   IF E=0   THEN 330
324     PRINT "SKIPPING VCO SINEWAVE FREQUENCY TEST"
326     GOTO   350
330   O=0                            \ REM VCO SINEWAVE FREQ TEST
331   GOSUB 2000
340   IF F>86.4 AND F<86.8 THEN 350
342     PRINT "VCO SINE WAVE FREQUENCY IS NOT OK"
344     PRINT "SHOULD BE BETWEEN 86.6 AND 86.8 KHZ FOR THIS TEST"
345     PRINT "IT ACTUALLY IS  ",F, "  KHZ"
346     PRINT "CHECK DIAGNOSTICS CONNECTIONS OR THE VCO CARD"
348     PRINT "*****      PROCEEDING WITH DIAGNOSTICS      *****"
```

```
350    N=256                          \ REM RELAY NO 8 FOR SINE VOLTAGE
351    GOSUB 1000
352    IF E=0 THEN 360
353       PRINT "SKIPPING VCO SINEWAVE VOLTAGE TEST "
354       GOTO 400
360    O=1                            \ REM VCO SINEWAVE VOLTAGE TEST
361    GOSUB 2000
370    IF F>3.3 AND F<3.6   THEN 400
372       PRINT "VCO SINE WAVE VOLTAGE NOT OK"
374       PRINT "SHOULD BE BETWEEN 3.1 & 3.4 V. IT IS ",F," VOLTS"
376       PRINT "CHECK DIAGNOSTICS CONNECTIONS OR THE VCO CARD"
378       PRINT "*****      PROCEEDING WITH DIAGNOSTICS      *****"
420    N=128                          \ REM RELAY 6 FOR SQ WAVE VOLTAGE
421    GOSUB 1000
422    IF E=0   THEN 430
424       PRINT "SKIPPING VCO SQUAREWAVE FREQUENCY TEST"
426       GOTO   450
430    O=0                            \ REM VCO SQUARE WAVE FREQ TEST
431    GOSUB 2000
440    IF F>86.4 AND F<86.9 THEN 450
442       PRINT "VCO SQUARE WAVE FREQUENCY IS NOT OK"
444       PRINT "SHOULD BE BETWEEN 86.6 AND 86.8 KHZ FOR THIS TEST"
445       PRINT "THE ACTUAL FREQUENCY IS ",F,"  KHZ"
446       PRINT "CHECK DIAGNOSTICS CONNECTIONS OR THE VCO CARD"
448       PRINT "*****      PROCEEDING WITH DIAGNOSTICS      *****"
450    N=64                           \ REM RELAY 7 FOR SQUARE VOLTAGE
451    GOSUB 1000
452    IF E=0 THEN 460
453       PRINT "SKIPPING VCO SQUARE WAVE VOLTAGE TEST "
454       GOTO 500
460    O=1                            \ REM VCO SQ
461    GOSUB 2000
470    IF F>2.3   AND F<2.5   THEN 500
472       PRINT "VCO SQUARE WAVE VOLTAGE NOT OK"
474       PRINT "SHOULD BE BETWEEN 2.3 & 2.5 V. IT IS ",F," VOLTS"
476       PRINT "CHECK DIAGNOSTICS CONNECTIONS OR THE VCO CARD"
478       PRINT "*****      PROCEEDING WITH DIAGNOSTICS      *****"
```

```
500 REM ************************************************************
501 REM *    TEST THE FREQ DAC AND AMP DAC OUTPUTS                  *
502 REM ************************************************************
520   N=32
521   GOSUB 1000
522   IF E=0   THEN 530
524      PRINT "SKIPPING FREQ-DAC-OUTPUT TEST"
526      GOTO 550
530   O=1
531   GOSUB 2000
540   IF F>4.9  AND F<5.1  THEN 550
542      PRINT "     FREQ-DAC OUTPUT          IS NOT OK"
544      PRINT "SHOULD  BE BETWEEN 4.9  AND 5.1 V FOR THIS TEST"
546      PRINT "CHECK DIAGNOSTICS CONNECTIONS OR THE VCO CARD"
548      PRINT "*****      PROCEEDING WITH DIAGNOSTICS      *****"
550   N=16
551   GOSUB 1000
552   IF E=0 THEN 560
554      PRINT "SKIPPING  AMP-DAC-OUTPUT TEST"
556      GOTO 600
560   O=1
561   GOSUB 2000
570   IF F>4.9  AND F<5.1  THEN 600
572      PRINT "     AMP-DAC OUTPUT           IS NOT OK"
574      PRINT "SHOULD BE BETWEEN 4.9  AND 5.1 V FOR THIS TEST"
576      PRINT "CHECK DIAGNOSTICS CONNECTIONS OR THE VCO CARD"
578      PRINT "*****      PROCEEDING WITH DIAGNOSTICS      *****"

600 REM ************************************************************
601 REM *    TEST ALL THE 12 ISOLATED DIGITAL INPUT PORTS           *
602 REM ************************************************************
610   N=8
611   GOSUB 1000
612   IF E=0 THEN 620
614      PRINT "WILL TRY AGAIN. IF IT FAILS ISO TEST TO BE SKIPPED"
620   N=4
621   GOSUB 1000
622   IF E=0 THEN 630
624      PRINT "WILL TRY AGAIN. IF IT FAILS ISO TEST TO BE SKIPPED"
630   N=12
631   GOSUB 1000
632   IF E=0 THEN 640
634      PRINT "IGNORE THE ABOVE MESSAGE....IT ONLY MEANS THAT "
636      PRINT " RELAYS 3 AND/OR 2 HAVE FAILED. SKIPPING ISO TEST"
638      PRINT "**** AND PROCEEDING WITH FURTHER DIAGNOSTICS  ****"
639      GOTO 700
640   PRINT#1,"&61600$"
641   PRINT#1,"&0$"
642   INPUT#1,"?",B$
650   B=VAL(B$)
651   IF B=4095 THEN 700
660      PRINT "ONE OR MORE OF THE ISO INPUTS INDICATED BY ZERO(0)"
662      PRINT "IN THE FOLLOWING BINARY WORD,  DO NOT WORK ...   "
670      GOSUB 3000
680      PRINT "*******        ",B$,"        ******"
```

```
700 REM ***********************************************************
701 REM * TEST THE ADC CARD BY MEASUREING 5 V FROM AMP DAC OUTPUT *
702 REM ***********************************************************
710   N=2
711   GOSUB 1000
712   IF E=0 THEN 720
714       PRINT "SKIPPING THE ADC TEST  AND PROCEEDING"
716       GOTO 800
720   PRINT#1,"&61584$"
721   PRINT#1,"&20480$"
722   INPUT#1,"?",D$
730   D=VAL(D$)*.005
740   IF D>4.8 AND D<5.1   THEN 800
750       PRINT " ADC CARD DOES NOT MEASURE ACCURATELY  "
760       PRINT " IT MEASURED  ",D," VOLTS FROM A 5VOLTS INPUT"
770       PRINT "*****     PROCEEDING WITH DIAGNOSTICS     *****"
800 REM
810   N=1
811   GOSUB 1000
812   IF E=0 THEN 900
820       PRINT "RELAY 0 .IE. THE 1ST RELAY IS NOT OK"
830       PRINT "*****     PROCEEDING WITH DIAGNOSTICS     *****"
900 REM
910   PRINT "**********END OF DIAGONISTICS**********"


1000 REM **********************************************************
1010 REM *  ROUTINE TO TEST FOR THE PROPER FUNCTIONING OF A RELAY *
1020 REM *           AND TO CLOSE IT & OPEN ALL OTHER RELAYS      *
1030 REM **********************************************************
1040 E=0
1055 N$=STR$(N)   \   N$ = N$(2)
1060 PRINT#1,   "%W",N$,"$"
1070 INPUT#1,  "%R",B$
1080 B=VAL(B$)
1090 IF B=N THEN 1170
1100    PRINT "RELAY NO ", LOG(N)/LOG(2), " IS NOT OK.
1110    E=1
1170 REM
1180 RETURN
```

```
1190 REM
2000 REM **********************************************************
2010 REM * THIS ROUTINE MEASURES VOLTAGE (VOLT) OR FREQUENCY (KHZ)*
2012 REM *   PARAMETER  PASSED IS  O.. RESULT IS PASSED BACK IN F *
2020 REM **********************************************************
2030 REM
2031   C$="#20"
2032   C3=3992.5
2034   IF O=0 THEN 2040
2036      C$= "#21"
2038      C3= 4025
2040   PRINT#1,C$
2050   FOR L=0 TO 4
2060       INPUT#1,D$
2070       D(L) = VAL(D$) - 256 - O*512
2080       NEXT L
2090   F = ((D(0)+D(1)*256+D(2)*256*256+D(3)*256*256*256)/C3)
2100   RETURN
3000 REM ***********************************************************
3001 REM * ROUTINE TO CONVERT A DECIMAL NUMBER TO A BINARY STRING   *
3002 REM ***********************************************************
3010   T=VAL(B$)
3015   B$=""
3020   FOR I=11 TO 0 STEP -1
3030       IF (2^I) <=T THEN B$=B$+"1"
3040       IF (2^I) > T THEN 3070
3050          T = T-2^I
3060          GOTO 3080
3070       B$=B$+"0"
3080       NEXT I
3090   RETURN
```

APPENDIX 14

Program listing for sampling and analyzing a signal.

```
10 REM ***********************************************************
20 REM *                  FREQUENCY ANALYZER                       *
60 REM * THIS PROGRAM COLLECTS SAMPLES OF DATA FROM                *
61 REM * THE SIGNAL CONNECTED TO THE FAST A/D INPUT                *
62 REM * AND TRANSFORMS IT INTO FREQUENCY SPACE....                *
63 REM * FOURIER TRANSFORM IS TAKEN 50 POINTS ON EITHER SIDE       *
64 REM * OF THE INPUT FREQUENCY.. WHICH MUST BE INPUT.......       *
66 REM ***********************************************************
70 PRINT     "***FOURIER SPECTRUM ANALYZER***"
100 PRINT "THE USER INPUTS THE NUMBER OF DATA POINTS TO BE TAKEN"
130 PRINT "THE PROGRAM WILL THEN PLOT THE DATA AND ITS SPECTRUM "
140 PRINT
150 INPUT "GIVE THE FREQUENCY OF THE INPUT SIGNAL",F1
160 INPUT "INPUT TERMINAL WIDTH:  ",L
190 INPUT "NUMBER OF DATA POINTS: ",N
200 INPUT "INPUT SCALE FACTOR:  ",I
205 IF I <1 THEN 200
210 X1=0 \ X2=(N-1)/9120  \  D=(X2-X1)/(N-1)
212 REM *********************************************************
213 REM * CLEAR THE INPUT PORT IF REQUIRED, BY A DUMMY READ    *
214 REM *********************************************************
215 FOR I=1 TO 10
220 INPUT#1,D$
225 NEXT I
230 PRINT "GOING TO COLLECT SAMPLES"
240 DIM D(N)
250 PRINT#1,"A"
260 FOR I=1 TO N
270 INPUT#1,D$
280 D(I)=VAL(D$)
285 NEXT I
340 FOR I=1 TO N
345 IF D(I) < 2048 THEN 355
350 D(I)=D(I)-4096
355 D(I)=D(I)*.005
360 NEXT I
370 B=0
379 REM **********************************************************
380 REM *    SHIFT DATA VALUES TO NON NEGATIVE                    *
381 REM **********************************************************
390 FOR I=1 TO N
400 IF B>D(I) THEN B=D(I)
410 NEXT I
420 FOR I=1 TO N
430 D(I)=D(I)-B
440 NEXT I
450 B=ABS(B)
460 PRINT\PRINT\PRINT
```

```
469 REM ************************************************************
470 REM *       FIND THE MAX. DATA VALUE                           *
471 REM ************************************************************
480 T=0
490 FOR I=1 TO N
500 IF T<D(I) THEN T=D(I)
510 NEXT I
519 REM *
520 REM   PLOT DATA
530 S=L/T
540 E=S*B
550 PRINT\PRINT
560 PRINT "    DATA PLOT (NORMALIZED)    "
570 PRINT\PRINT
580 PRINT "X1= ",X1
590 GOSUB 1190
600 FOR I=1 TO N
610 IF S*D(I)<1 THEN GOTO 620
620 IF S*D(I)<1 THEN PRINT "*"
630 IF S*D(I)>=1 THEN PRINT TAB(S*D(I)),"*"
640 IF S*B<1 THEN GOTO 660
650 REM PRINT TAB(E),":"
660 NEXT I
670 GOSUB 1200
680 PRINT "X2= ",X2
689 REM ************************************************************
690 REM *          START   FOURIER   TRANSFORM                      *
691 REM ************************************************************
700 DIM F(100)
702 W1=2*3.14159*F1-5000  \W2=2*3.14159*F1+5000   \W3=1
710 FOR I=1 TO 100
720 W=W1+(I-1)*100
730 C1=0\S1=0
740 FOR M=1 TO N
750 X=X1+(M-1)*D
760 G=W*X
770 C1=C1+D(M)*COS(G)
780 S1=S1+D(M)*SIN(G)
790 NEXT M
800 F(I)=SQRT(S1*S1+C1*C1)*D
820 NEXT I
830 REM TRANSFORM DATA TABLE
860 PRINT "   FREQ(RAD)    MODULUS"
870 PRINT "   ---------    -------"
880 FOR I=1 TO 100
890 PRINT "     ",INT((1000*W1+.5)/1000+(I-1)*100),"  ",
900 PRINT "    ",INT(1000*F(I)+.5)/1000,
910 NEXT I
920 PRINT\PRINT
```

```
929 REM *********************************************************
930 REM *              TRANSFORM PLOT                           *
931 REM *********************************************************
940 PRINT "    FREQUENCY PLOT (NORMALIZED)"
950 PRINT\PRINT
960 PRINT INT(1000*W1+.5)/1000,
970 PRINT " RADIANS/SECOND   ",
980 PRINT INT(500*W1/3.141595+.5)/1000,
990 PRINT " HERTZ (CYCLES/SECOND)"
1010 GOSUB 1200
1020 T=0
1030 FOR I=1 TO 100
1040 IF T<F(I) THEN T=F(I)
1050 NEXT I
1060 S=L/T
1070 FOR I=1 TO 100
1080 IF S*F(I)<1 THEN GOTO 1090
1090 IF S*F(I)<1 THEN PRINT "*"
1100 IF S*F(I)>=1 THEN PRINT TAB(S*F(I)-.5),"*"
1110 REM  PRINT ".",TAB(L-1),"."
1120 NEXT I
1130 GOSUB 1200
1140 PRINT INT(1000*W2+.5)/1000,
1150 PRINT " RADIANS/SECOND  ",
1160 PRINT INT(500*W2/3.141595+.5)/1000,
1170 PRINT " HERTZ (CYCLES/SECOND)"
1180 END
1190 REM BORDER LINE PLOT
1200 FOR I=1 TO L
1210 PRINT "*",
1220 NEXT I
1230 PRINT
1240 RETURN
```

BIBLIOGRAPHY

[1]. NorthStar System Software Manual #25013B, Revision 2.1,1980. NorthStar Computers, Inc. 1440 Fourth St., Berkeley, CA 94710.

[2]. Multiprogrammer Model 6940B Operating and Service Manual. June 1979. Hewlett-packard, New Jersey Division, Green Pond Road, Rockaway, New Jersey 07866.

[3]. E.E. Creel and J. Roberts "A Dedicated Interface for the HP6940B Multiprogrammer.", Department of Electrical Engineering, Kansas State University, Manhattan, Kansas, Fall 1980.

[4]. Voltage regulator card 69351B, Hewlett-packard, New Jersey Division, Green Pond Road, Rockaway, New Jersey 07866.

[5]. Isolated Digital Input Card 69430A, Hewlett-packard, New Jersey Division, Green Pond Road, Rockaway, New Jersey 07866.

[6]. Breadboard Input Card 69480A, Hewlett-packard, New Jersey Division, Green Pond Road, Rockaway, New Jersey 07866.

[7]. G. Krohe, "Digitally Driven Voltage Controlled Oscillator.", Department of Electrical Engineering, Kansas State University, Manhattan, Kansas, December 1981.

[8]. G. Sherwood, "Voltage/Frequency Input Card.", Department of Electrical Engineering, Kansas State University, Manhattan, Kansas, 1980.

[9]. Breadboard Output Card 69380A, Hewlett-packard, New Jersey Division, Green Pond Road, Rockaway, New Jersey 07866.

[10]. Relay Output/Readback Card 69433A, Hewlett-packard,New Jersey Division, Green Pond Road, Rockaway, New Jersey 07866.

[11]. TI 700 Silent Printer, Model 743 KSR Data Terminal Operating Instructions, Texas Instruments Inc., Digital Systems Division, P.O. Box 1444, Houston, Texas 77001.

[12]. OKIDATA Microline 82A User's Manual,Okidata Corp., Mt. Laurel, New Jersey 08054.

[13]. HEATHKIT Manual for Video Terminal Model H-19A Heath Company, Benton Harbor, Michigan 49022.

[14]. High Speed A/D Converter Card 69422A, Hewlett-packard, New Jersey Division, Green Pond Road, Rockaway, New Jersey 07866.

[15]. M. S. P. Lucas, "An Introduction to Computer Aided Instrumentation.", Department of Electrical Engineering, Kansas State University, Manhattan, Kansas.

[16]. W. S. Park, "An Automatic Test System.", Department of Electrical Engineering, Kansas State University, Manhattan, Kansas, July 1982.

ACKNOWLEDGEMENTS

I take this opportunity to express my sincere thanks to my major advisor Dr. M. S. P. Lucas. His guidance, stimulating discussions, enthusiasm and encouragement have contributed abundantly towards this project.

To Dr. D. Hummels and Dr. P. Richard, thanks for your valuable suggestions and support. Special thanks to Dr. G. Johnson for initiating me on the spinwriter, which made the typing of this report a pleasure.

I also express my gratitude towards my parents whose encouragement and constant support has contributed in a big way towards the successful completion of my graduate studies. Last but not the least, my sincere thanks to Sheila for her patience and understanding through this process.

AN AUTOMATED LABORATORY TEST SYSTEM

by

NEUFITO L. FERNANDES

B.S., Indian Institute of Technology - Bombay, 1979

AN ABSTRACT OF A MASTER'S REPORT

Submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1983

ABSTRACT

A typical electronic laboratory experiment involves many manual operations such as making/breaking interconnections, varying input signals, recording input/output signals, processing and analyzing the recorded data and so forth. These operations are extremely time consuming, highly prone to human error, and do not add to any useful experience after the initial principles and the experimental set-up are understood. To help overcome this problem an Automated Laboratory Test System has been designed and built.

Each system consists of a HP Multiprogrammer model 6940B, a North Star Horizon microcomputer, and an MC6802 based interface. The microcomputer follows the RS232 standard for its I/O which is ASCII coded. It is connected to the MC6802 based interface via a serial port. This interface converts the ASCII coded serial data to Binary coded parallel data required by the HP Multiprogrammer. The level translation, from RS232 to TTL and vice versa, is handled by MC1488 and MC1489 (sender-/receiver) quad line drivers. Various input/output accessory cards are provided on the HP Multiprogrammer to control the device under test. The Voltage/Frequency measurement card and the Voltage Controlled Oscillator card were designed and built by students. The remaining three cards, the Relay card, the High Speed Analog to Digital Conversion card and the Isolated Digital Inputs card, are factory built cards for use with the HP Multiprogrammer system.

The MC6802 monitor program controls the communications between the North Star and the HP Multiprogrammer. It has been designed to provide

simple and user friendly commands to interact with each of the HP Multiprogrammer cards directly from the user program in BASIC. It also performs the ASCII to BCD and BCD to ASCII conversions.

This project involved the efforts of several students under the guidance of Professor M. S. P. Lucas. The Author is responsible for the new software, The High Speed A/D Converter card, and the Automated Diagnostics feature.