NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

H3255

MULTICHANNEL DATA TRANSMISSION THROUGH A FIBER OPTIC CABLE

by

Fokion Hatzidakis

December 1987

Thesis Advisor: J. P. Powers

Approved for public release; distribution is unlimited

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; Distribution is unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b OFFICE SYMBOL (If applicable) 62
7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School	
6c. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b OFFICE SYMBOL (If applicable)
9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification)
MULTICHANNEL DATA TRANSMISSION THROUGH A FIBER OPTIC CABLE

12 PERSONAL AUTHOR(S)
HATZIDAKIS, Fokion

13a. TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM ______ TO ______	1987 December	92

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Fiber optic cable, optical source, detector, encoder, decorder, Manchester, controller, CPU

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

The objective of this thesis was to design and construct a system which transmits and receives data from different analog channels through a single fiber optic cable link. The system uses a microprocessor controlled multiplexer and a high speed analog-to-digital converter. Tests and measurements were performed to examine the restrictions, problems and different options on the various components of the system. Based on the above, a trade-off analysis was performed for the bandwidth and the number of channels.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT [✓ UNCLASSIFIED/UNLIMITED] ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a NAME OF RESPONSIBLE INDIVIDUAL J.P. POWERS	22b TELEPHONE (Include Area Code) (408) 646-2082	22c. OFFICE SYMBOL 62Po

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

Multichannel Data Transmission through a Fiber Optic Cable System Link

by

Fokion Hatzidakis
Lieutenant, Hellenic Navy
B.S., Hellenic Naval Academy, 1978

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1987

ABSTRACT

The objective of this thesis was to design and construct a system which transmits and receives data from different analog channels through a single fiber optic cable link. The system uses a microprocessor controlled multiplexer and a high speed analog-to-digital converter. Tests and measurements were performed to examine the restrictions, problems and different options on the various components of the system. Based on the above, a trade-off analysis was performed for the bandwidth and the number of channels.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENT

I would like to thank John Glenn, Optics-Lab Manager,for his technical support. His assistance during the design and construction of this system is greatly appreciated. I also wish to extend my gratitude to Professor J.P. Powers for introducing me into the fiber optics world. Finally, I express my appreciation to my wife Peggy for her support throughout my studies in Naval Postgraduate School.

I. INTRODUCTION

The increasing sophistication of multiple-function local area networks, the growth of personal computing in the business marketplace and the light weight demand are forcing system developers into an examination of not only what operating systems to use, but also what would be the optimum cable/system design. The growing requirements for a means of data transmission having light weight, wide bandwidth, immunity from electromagnetic interference and small size have all increased the demand for optical fiber.

Because of their large comparative bandwidths, fibers can carry large amounts of information. However, bandwidth capacity limits exist for all types of fibers and depend on the fiber, type of emitter and hardware employed. Furthermore in data acquisition systems, where we usually have more than one channels, the relation between bandwidth limits and the number of channels is of great importance. While complete ready-made systems are commercially available, for an experimental design, one usually has to conduct a trade-off analysis while meeting the specifications. The results and the performance of such a system would greatly depend on the design philosophy and the hardware used. In addition, the choice between analog and digital transmission and how the synchronization between input and output should be done will determine the parameters involved in designing a system link.

The purpose of this investigation was to design, construct and test of a system which transmits and receives data from different channels through a single fiber optic cable. Also, based on the above tests and measurements, a secondary goal will perform a trade-off analysis for the bandwidth and the number of channels and the system performance. This system required that the input and output data would be in analog form. Various other design requirements for this system included the following:

- The input to the system should have a maximum of about 20 KHz bandwidth.
- The input could have a possible DC component.
- The analog data should be converted to digital data, transmitted through the fiber optic cable and then converted to analog data again.
- The transmitter should be a light emitting diode (LED) or a semiconductor laser.
- The system should have the minimum hardware possible.

Even though this fiber optic system supposed to be designed for a specific analysis and limited to a certain hardware, there still are possibilities for expansion. The system is easily modified to include other applications. Also, there are more than one way to design and construct this system which gives the designer the flexibility to choose among different considerations.

II. SYSTEM DESIGN AND HARDWARE DEVELOPMENT

A. GENERAL SYSTEM DESCRIPTION

In order to design a system with the previous requirements, one has to consider the fact that the analog signal should be converted to a digital signal and reconverted at the output. In addition, when data is transmitted from one channel we have to make sure that these data will be received in the same channel number. Time division multiplexing systems are greatly depend on system synchronization. This proved the biggest problem in designing and building this system. The problem was solved by designing a controller which was able to control the input and output hardware. In order to gain insight into designing a controller and to have a comparison of differing performance, two different controllers were designed and constructed. The first one uses the NSC800 microprocessor made by National Semiconductor Corporation and the second one uses the Z8400H microprocessor made by Zilog Incorporation. Full description of these controllers will be given in a later section.

The first part of the design involves the transmitting circuit, a block diagram of which can be seen in Figure 2.1. An analog multiplexer was used to select the channel from which the data will be transmitted. The original design provided 16 channels. Most of the work, though, was done with one channel. The input bandwidth started with 22 KHz. A sample and hold circuit provided the data sampling before the conversion. The conversion from analog to digital data, was done by an analog to digital converter. A fast conversion time

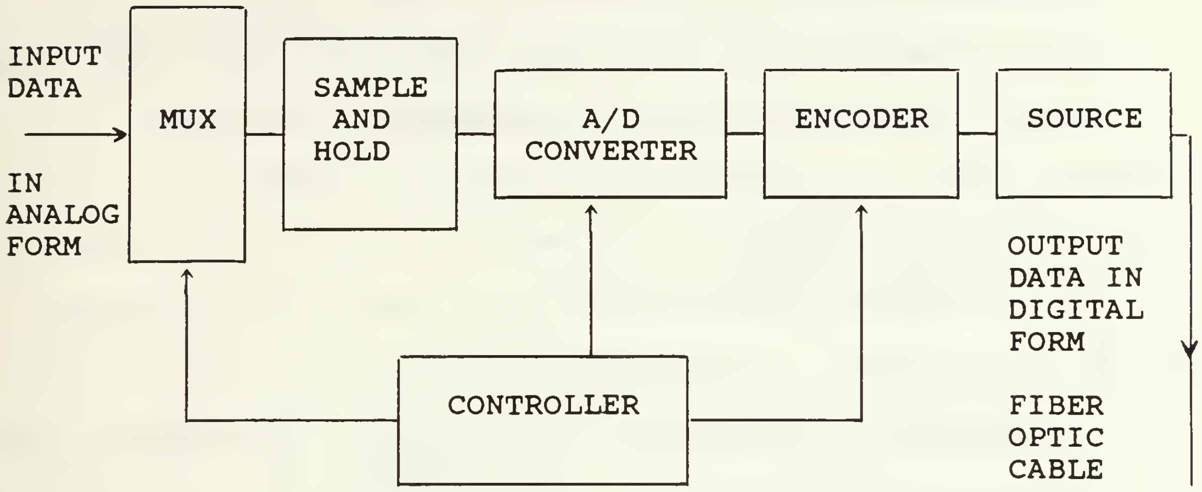


Figure 2.1 Basic Transmitter Block Diagram

was needed there and 8-bit resolution was chosen. Because in most fiber optic applications it is the Manchester scheme that has been used to encode data, this technique also used Manchester encoding. The Manchester encoder provides the needed synchronization pulse and the parity bit as well as the encoding of the data bits. The last stage in the transmission circuit is the source which transmits the data. The source can be a light emitting diode (LED) or a semiconductor laser. The connection between the transmitter and the receiver was done by a plastic fiber optic cable.

The second part of the system involves the receiving circuit, a block diagram of which is given in Figure 2.2. The the fiber optic cable connects directly to the detector. The optical detector is a pin photodiode which linearly convert the optical signal to an electronic signal. Continuing the Manchester scheme, a Manchester decoder recognizes the synchronization pulse and identifies it as well as decoding the data bits and checking the parity.

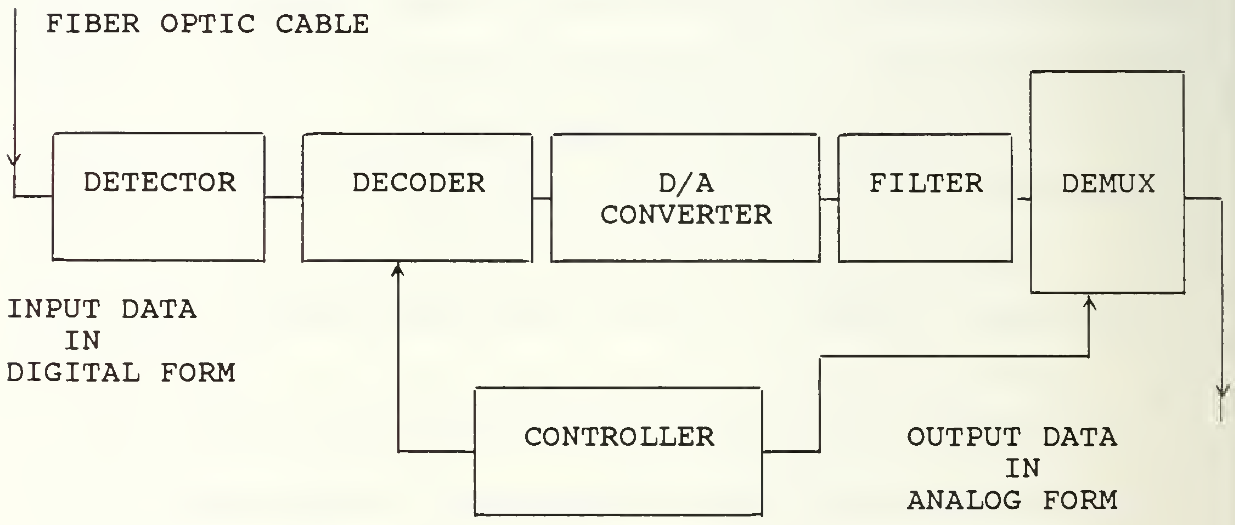


Figure 2.2 Basic Receiver Block Diagram

Having in mind that the output should be in analog form, a digital to analog converter is used for this conversion. Due to the quantization and the hardware nonlinearities, the output waveform is not exactly the same as the input waveform. It suffers amplitude and phase distortion so a filter circuit was needed in order to smooth these problems. The final stage in the receiver is the demultiplexer circuit from which the analog data is coming. The receiver is controlled by the second controller. The selection of the hardware was such that the relating parts were compatible with each other.

B. FIBER OPTIC DATA LINK

1. Optical Transmitter

The HFBR-1402 fiber optic transmitter, made by Hewlett Packard, was selected because of its capability of efficiently launching optical power into five different optical fiber sizes: 100/140 µm, 50/125 µm, 62.5/125 µm, 85/125 µm, and 200 µm PCS [Ref. 1]. This allows the designer flexibility in choosing the fiber optic size. The HFBR-1402 transmitter's high coupling efficiency allows the emitter to be driven at low current levels resulting in low power consumption and increased reliability of the transmitter. The high speed GaAlAs emitter operates at a wavelength of 820 nm. The transmitter can be used to design fiber optic data link at rates up to 5 Megabaud and it has a typical rise/fall time of 4.0 ns. The transmitter's circuit can be seen in Figure 2.3. In order to have low pulse distortion, the transmitter is driven with a 5.34 mA current at pin number 2 by setting the regulating resistor at 500Ω . The circuit also uses a DS75451 dual peripheral positive AND driver, made by Texas Instruments to be used as a power driver. In conclusion the circuit receives the digital data from the encoder and translates the electronic signal to light pulses which then are transmitted. [Ref. 1]

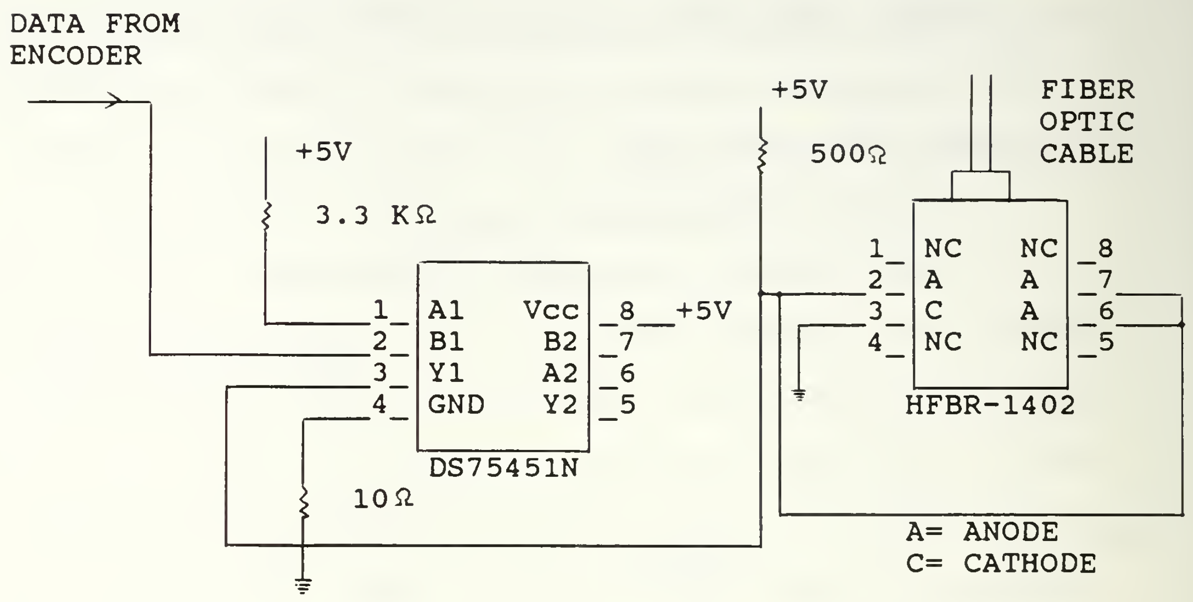


Figure 2.3 Optical Transmitter Circuit [Ref. 1]

2. Optical Receiver

For compatibility purposes, the HFBR-2402 Fiber optic receiver, made by Hewlett Packard, was chosen as the receiving device at the other end of the fiber optic cable. The receiver incorporates a monolithic photo-IC which contains a photodetector and a dc amplifier. It is essential that a bypass capacitor, 0.01 µF, be connected from pin 2 to pin 7 to remove the glitches. The receiver is designed to operate at rates up to 5 MBaud. The transmitter-receiver system has a typical pulse width distortion of 25 nsec and a Bit Error Rate of 10^{-9} with minimum receiver power of -24 dBm (4µW). The optical receiver circuit can be seen in Figure 2.4. [Ref. 1]

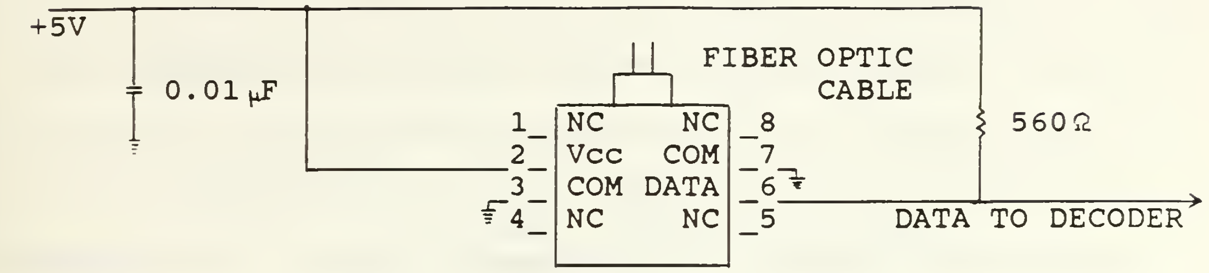


Figure 2.4 Optical Receiver Circuit [Ref.1]

3. <u>Other Circuit Options</u>

An other way to design the optical transmitter and the optical receiver circuits is illustrated in the next two figures. Figure 2.5 uses a SN74LS1000A quadruple 2-input

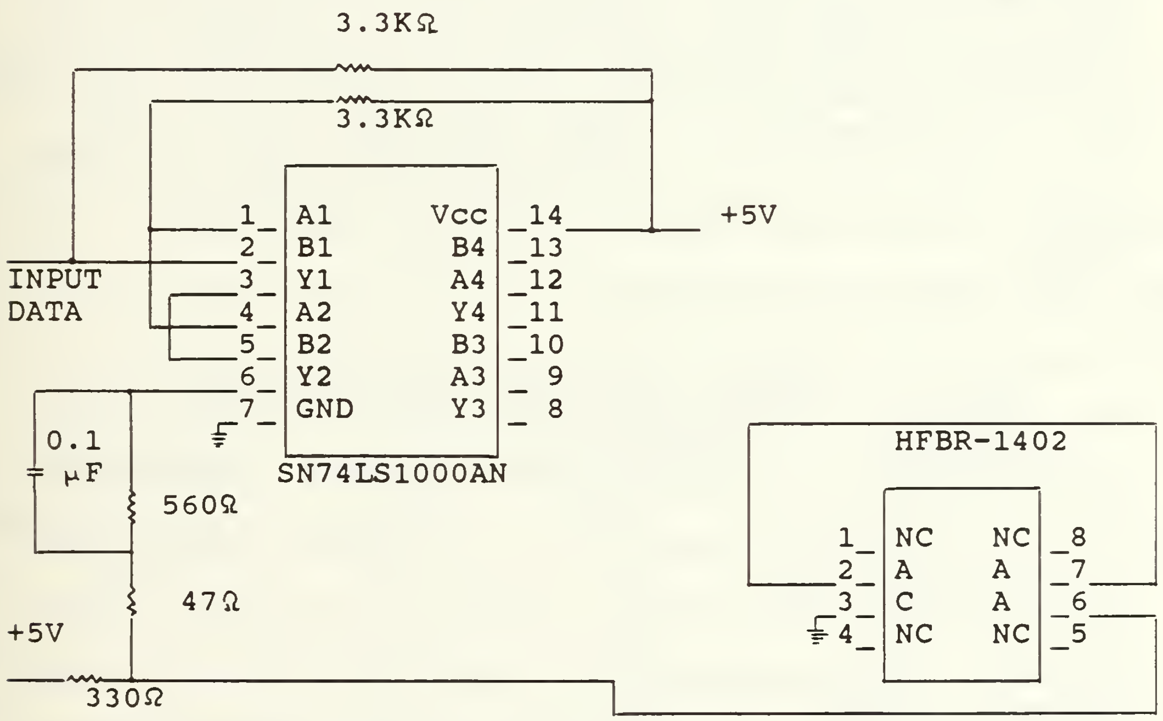


Figure 2.5 Optical Transmitter Circuit [Ref.1]

positive NAND gate used as a buffer. Figure 2.6 is a simple circuit which has only one resistor.

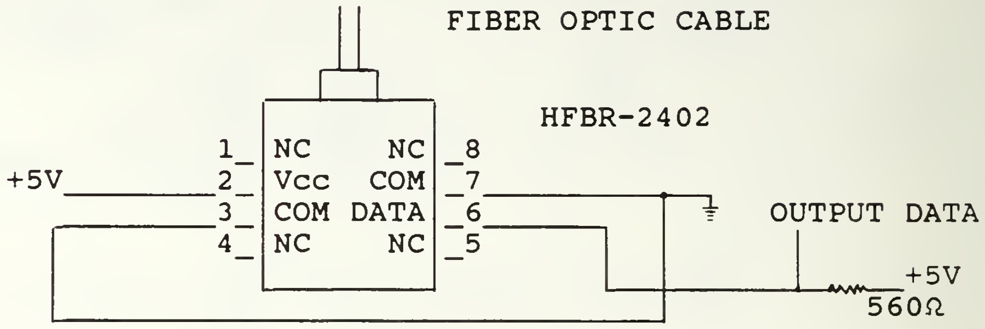


Figure 2.6 Optical Receiver Circuit [Ref. 1]

4. Fiber Optic Cable

The high performance HFBR-3510 plastic fiber optic cable, made by Hewlett Packard, was chosen because it was available at that time and because it was compatible with the transmitter-receiver system. It is a standard low loss cable and is constructed of a single step index plastic fiber. A table with the fiber optic cable specifications is given below.

TABLE 2.1 FIBER OPTIC CABLE SPECIFICATIONS [Ref.1]

Parameter	Typical Value	Units	Conditions
Cable Attenuation	0.31	dB/m	at 665nm Source NA = 0.5 l > 2m
Numerical Aperture	0.5	-	
Diameter, Core	1.0	mm	-
Diameter, Jacket	2.2	mm	-
Travel Time Constant	5.0	nsec/m	-

C. CODING

1. Encoder

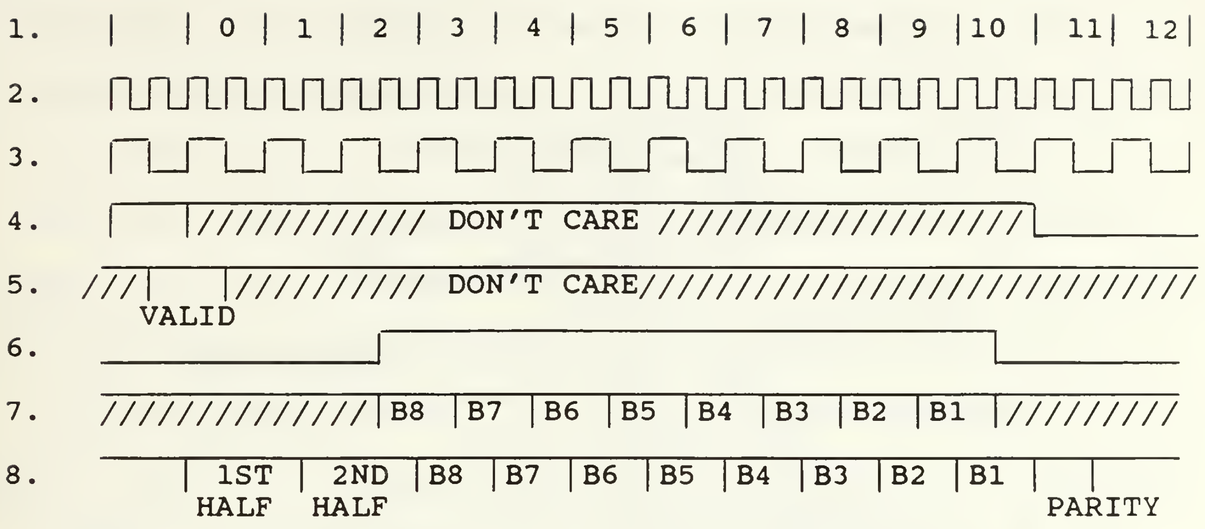
The following section was taken from Reference 2. The Harris HD-15531 is a high performance CMOS device inteneded to service the requirements of encoding and decoding the data of this system. This LSI chip is divided into two sections, an Encoder and a Decoder. These sections operate independently of each other, except for the MASTER RESET and word length functions. The HD-15531 allows the word length to be programmable (from 2 to 28 data bits). A frame consists of three bits for synchronization followed by the data word (2 to 28 data bits) followed by one bit of parity, thus the length will vary from 6 to 32 bit periods. This chip also allows selection of either even or odd parity for the Encoder and Decoder separately. This low power integrated circuit can support a 2.5 Megabit/sec data rate. The circuit used in this design was taken from Reference 2.

The Encoder requires a single clock with a frequency of twice the desired data rate applied at the SEND CLOCK input. An auxiliary divide by six counter is provided on chip which can be utilized to produce the SEND CLOCK by dividing the DECODER CLOCK. The frame length was set by programming the COUNT inputs. Having 8 bits, the frame length will be 12 bit periods. Parity was selected by programming ENCODER PARITY SELECT low for even parity. The Encoder's cycle begins when ENCODER ENABLE is high during a falling edge of ENCODER SHIFT

CLOCK. This cycle lasts for one word length or twelve ENCODER SHIFT CLOCK periods. At the next low-to-high transition of the ENCODER SHIFT CLOCK, a high SYNC SELECT input actuates a Command synchronization or a low will produce a Data synchronization for the word. When the Encoder is ready to accept data, the SEND DATA output will go high for eight ENCODER SHIFT CLOCK periods. During these eight periods the data should be clocked into the SERIAL DATA input with every high-to-low transition of the ENCODER SHIFT CLOCK so it can be sampled on the low-to-high transition. After the synchronization and Manchester encoded data are transmitted through the $\overline{\text{BIPOLAR}}$ $\overline{\text{ONE}}$, the Encoder adds on an additional bit with the parity for that word. To abort the Encoder transmission a positive pulse must applied at MASTER RESET. Any time after or during this pulse, a low-to-high transition on SEND CLOCK clears the internal counters and initializes the Encoder for a new word. Figure 2.7 shows the Encoder's timing diagram. Figure 2.8 shows the timing diagram of the Encoder's circuit. The Encoder's circuit, Figure 2.9, uses the SN74LS165 8-bit parallel to serial converter in order to convert the data from parallel to serial form. Parallel inputting occurs asynchronously when the Parallel Load ($\overline{\text{PL}}$) input is low. With $\overline{\text{PL}}$ high, serial shifting occurs on the rising edge of the clock.

2. <u>Decoder</u>

A second HD-15531 Manchester Encoder-Decoder chip had to be used at the other side of the receiving circuit to



1. TIMING
2. SEND CLOCK
3. ENCODER SHIFT CLOCK
4. ENCODER ENABLE
5. SYNC SELECT
6. SEND DATA
7. SERIAL DATA IN
8. $\overline{\text{BIPOLAR}}$ $\overline{\text{ONE}}$ $\overline{\text{OUT}}$

Figure 2.7 Encoder's Timing Diagram [Ref.2]

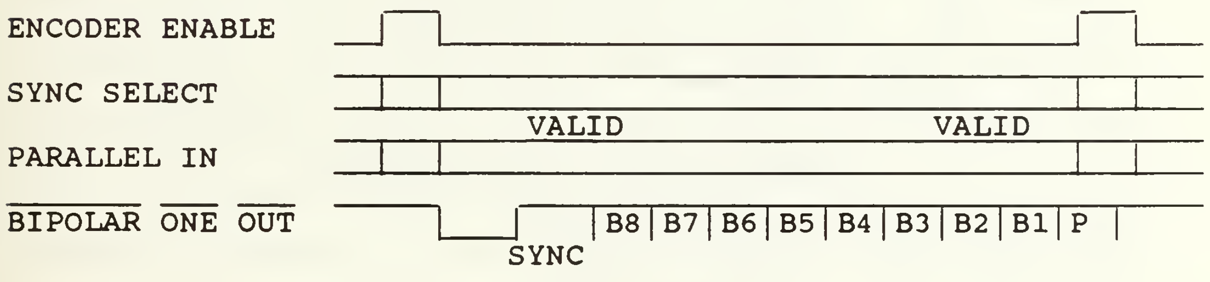
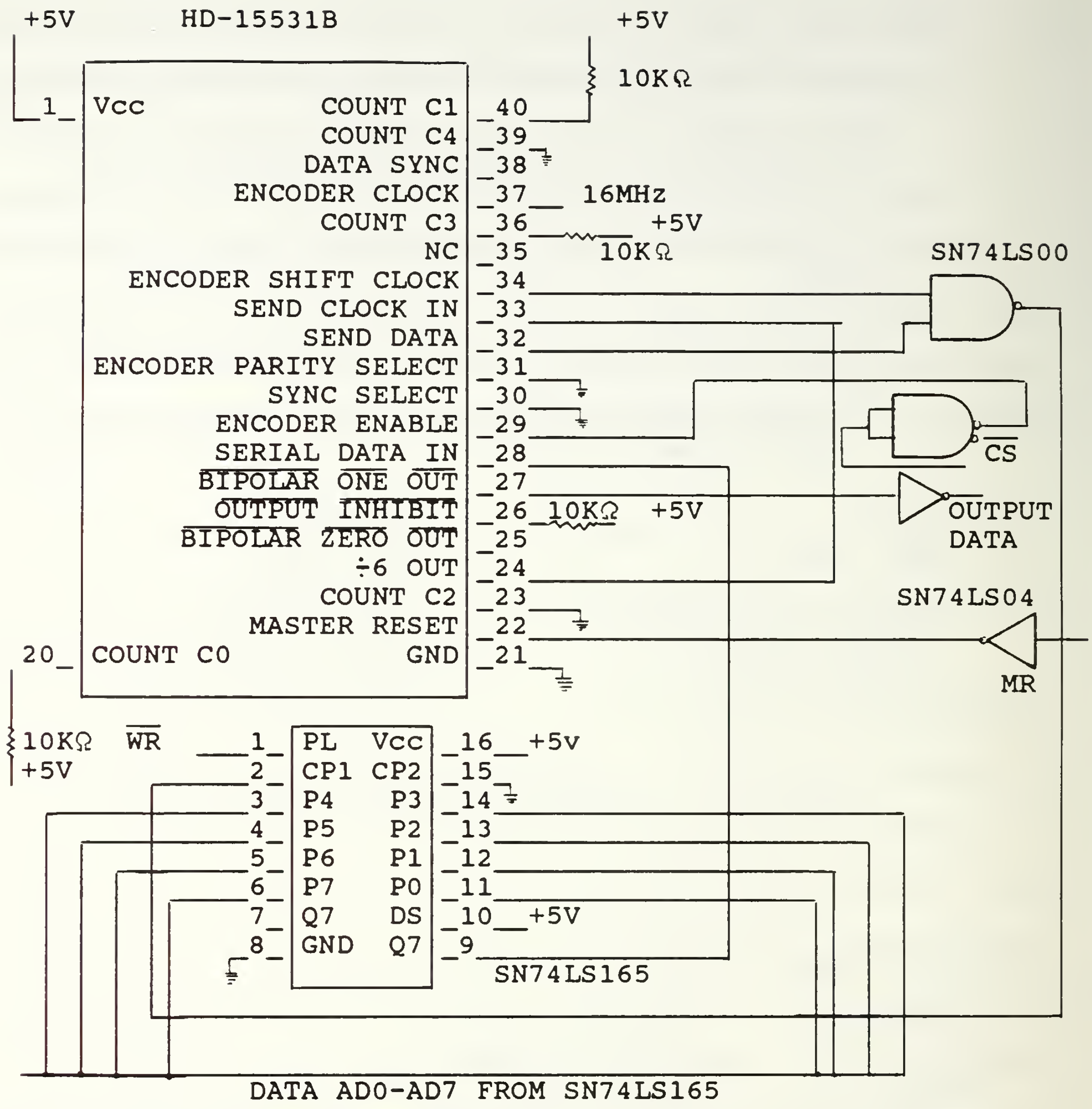


Figure 2.8 Encoder's Circuit Timing Diagram [Ref. 2]



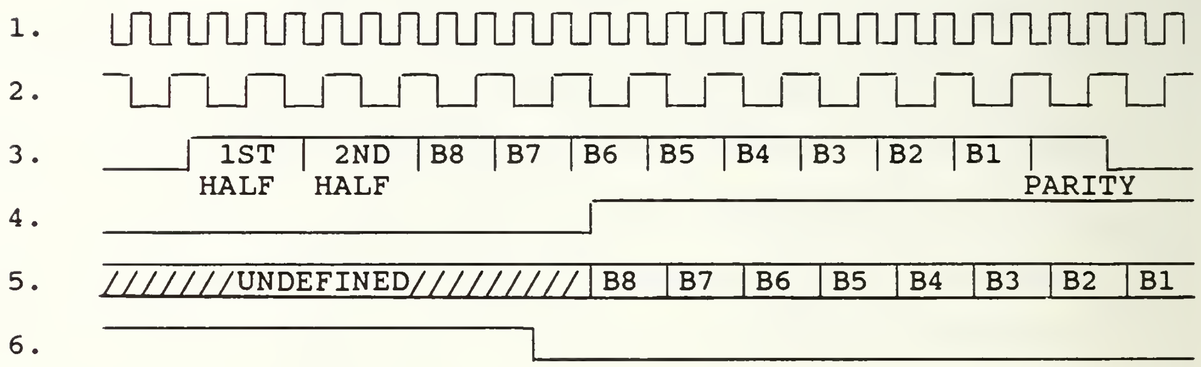
WR: FROM PIN 10 OF MM74HC138N-A

CS: FROM PIN 7 OF MM74HC138N-A

MR: FROM PIN 33 OF NSC800

Figure 2.9 Encoder's Circuit [Ref. 2]

decode the encoded data. The following section was taken from Reference 2. To operate the Decoder asynchronously requires a single clock with a frequency of twelve times the desired data rate applied at the DECODER CLOCK input. The Manchester II coded data can be presented to the Decoder in the following way. The signal from $\overline{\text{BIPOLAR}}$ $\overline{\text{ONE}}$ $\overline{\text{OUT}}$ on the Encoder through an inverter to UNIPOLAR DATA INPUT. The Decoder is free running and continuously monitors its data input for valid synchronization character and two valid Manchester data bits to start an output cycle. When a valid synchronization is recognized, the type of synchronization is indicated by the level at COMMAND SYNC which will go high and remain high for eight SHIFT CLOCK periods. The TAKE DATA output will go high and remain high while the Decoder is transmitting the decoded data through SERIAL DATA OUT. The decoded data available at SERIAL DATA OUT is in non return to zero (NRZ) format. The DECODER SHIFT CLOCK is provided so that the decoded bits can get shifted into an external register on every low-to-high transition of this clock. Note that DECODER SHIFT CLOCK may adjust its phase up until the time that TAKE DATA goes high. After all 8 decoded bits have been transmitted, the data is checked for parity. A high input on DECODER PARITY SELECT is set the Decoder to check for even parity. A high on VALID WORD output indicates a successful reception of a word without any Manchester or parity errors. At this time the Decoder is looking for a new synchronization character to start another output sequence. VALID WORD will go low approximately twelve DECODER SHIFT CLOCK periods after it goes high if not reset low sooner by a valid synchronization and two valid Manchester bits. At any time in the above sequence a high input on DECODER RESET during a low-to-high transition of DECODER SHIFT CLOCK will abort decoding and initialize the Decoder to start looking for a new synchronization character. Figure 2.10 shows



1. SYNCHRONOUS CLOCK
2. DECODER SHIFT CLOCK
3. BIPOLAR ONE IN
4. TAKE DATA-COMMAND SYNC-DATA SYNC
5. SERIAL DATA OUT
6. VALID WORD

Figure 2.10 Decoder's Timing Diagram [Ref. 2]

the Decoder's timing diagram. The Decoder's circuit, Figure 2.11, uses the high speed 8 bit serial-in parallel-out shift register. Serial data is entered through a 2-input AND gate synchronous with the low to high transition of the clock. Each low to high transition on the clock (CP) shifts data one place to the right and enters the Q outputs. Figure 2.12 shows the timing diagram of the Decoder's circuit.

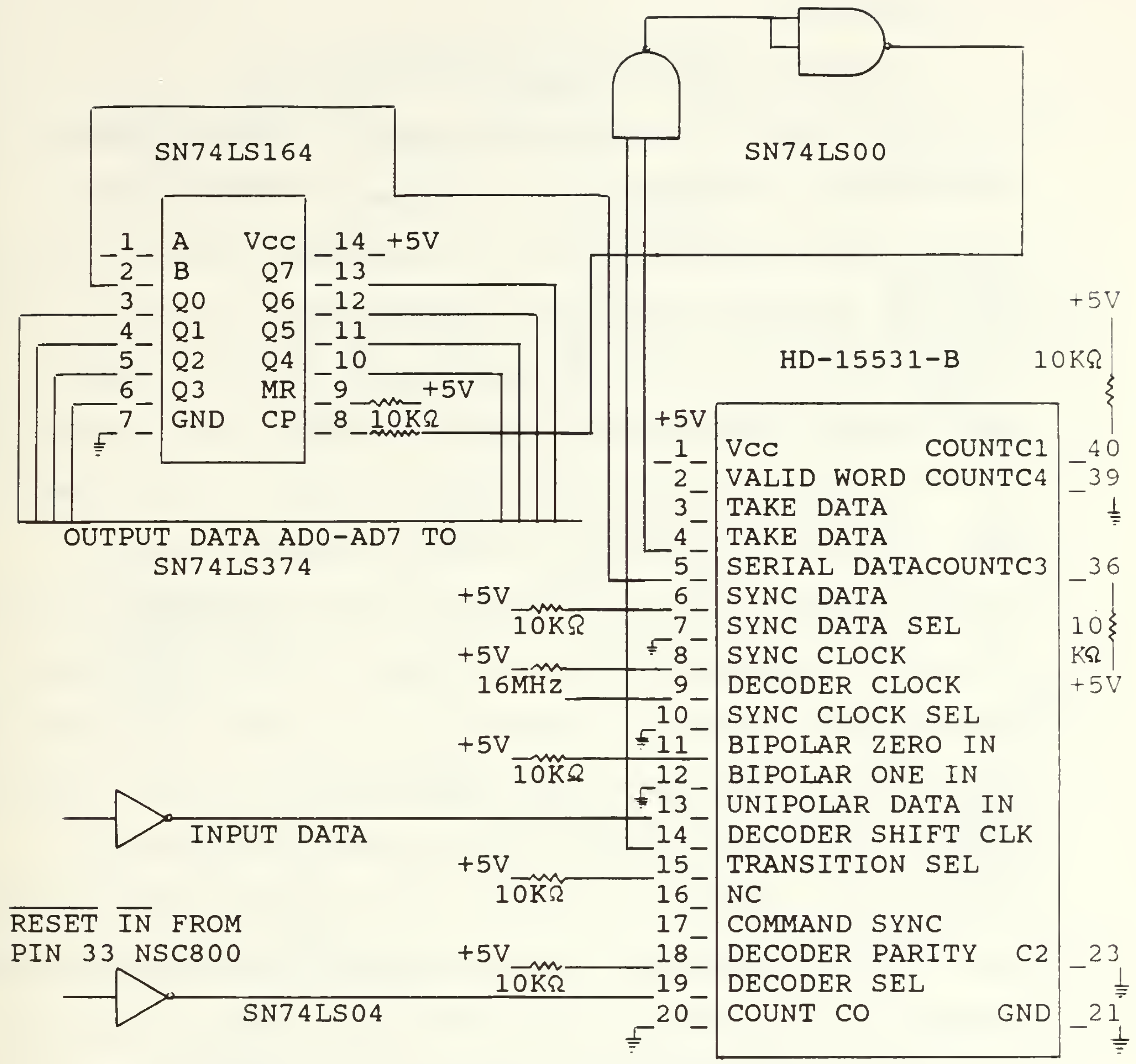


Figure 2.11 Decoder's circuit [Ref. 2]

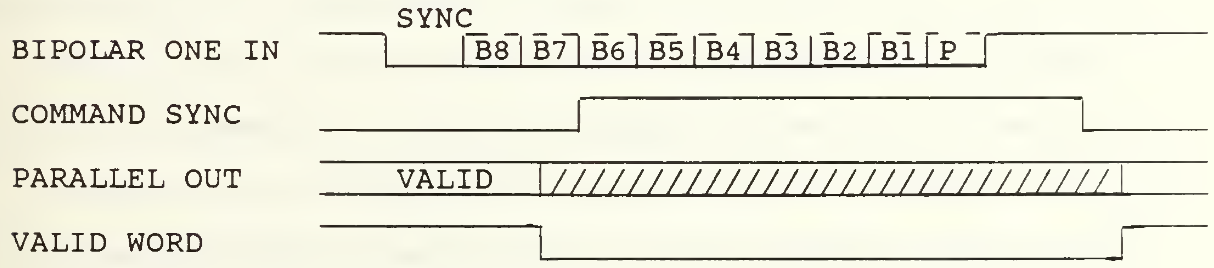


Figure 2.12 Decoder's Circuit Timing Diagram [Ref. 2]

D. CONTROLLER

1. General Description of the Controller

In order to develop and keep the correct timing of the chip selection/enable and all the other instructions a programmable controller had to be designed. Availability of parts and designing literature was a primary criterion for this purpose. The NSC800 Microprocessor chip was readily available and provides most of the requisite facilities. Also, the Z8400H Microprocessor chip was in order and was used to build a second controller. In both the controllers the parallel standard mode was used in which the parallel data is buffered internally in the controller. In this way slower devices were allowed to interface to the fiber optic link. The controller provides data flow buffering and control. The data paths provide 8-bit parallel data channels between the input/output interface, the controller and the shift register just before the encoder, and the control logic provides proper strobing for the various instructions of the A/D converter, the encoder and the channel selection. A block diagram of the controller is given in Figure 2.13. As can be seen, it contains a Central Processing Unit (CPU) for 8-bit processing, control latches for the various chip selections, data bus latches in the input/output interface, an Erasable Read Only Memory (EPROM) (used as program storage) and for faster processing a Random Access Memory (RAM). Detailed design of the controllers will be given as we examine each one

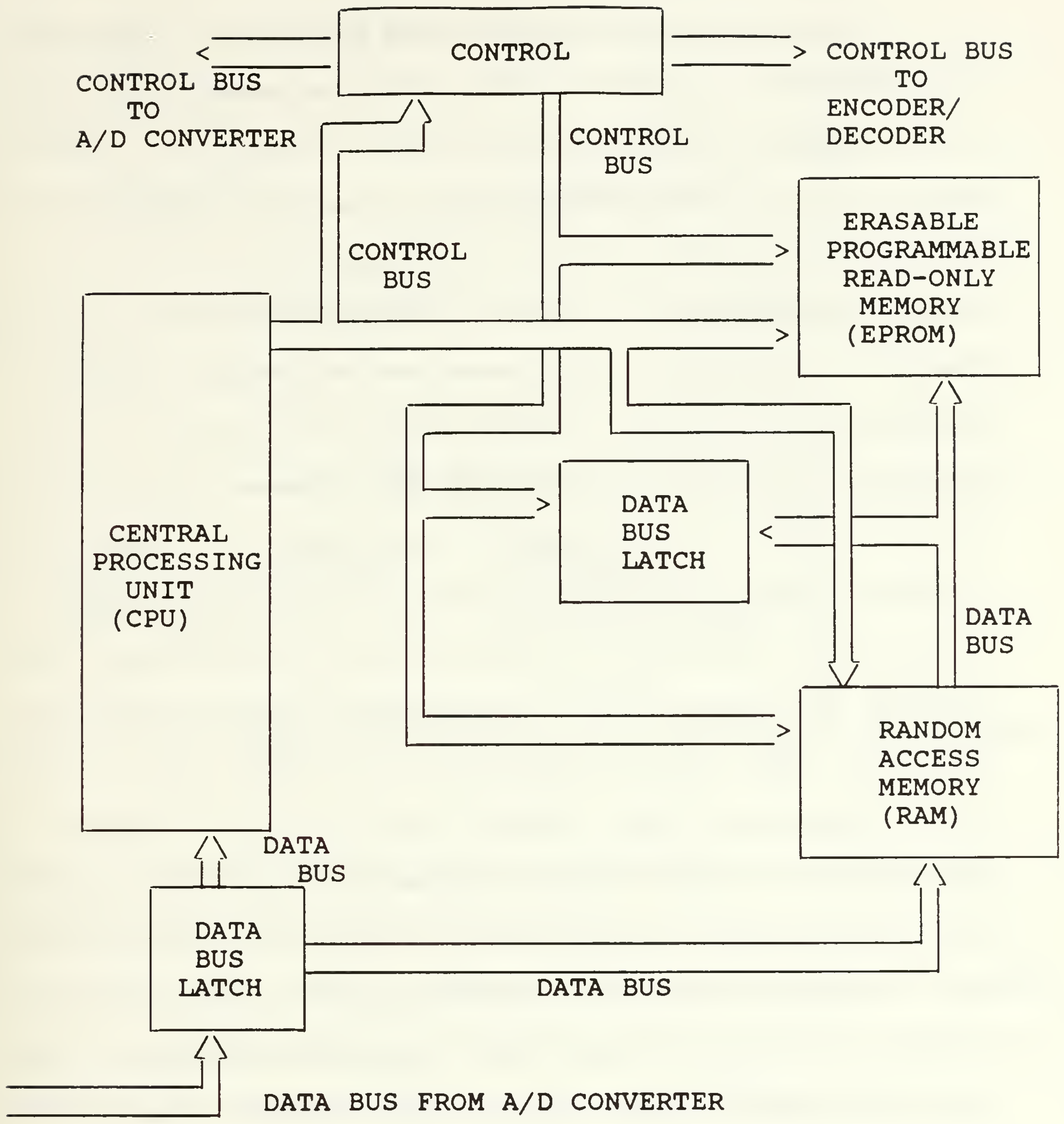

CONTROL
CONTROL BUS TO A/D CONVERTER
CONTROL BUS TO ENCODER/ DECODER
CONTROL BUS
CONTROL BUS
ERASABLE PROGRAMMABLE READ-ONLY MEMORY (EPROM)
CENTRAL PROCESSING UNIT (CPU)
DATA BUS LATCH
DATA BUS
RANDOM ACCESS MEMORY (RAM)
DATA BUS
DATA BUS LATCH
DATA BUS
DATA BUS FROM A/D CONVERTER

Figure 2.13 Block Diagram of the Controller

individually. This design was not the optimum solution but allowed completion of this effort with reasonable resources.

2. The NSC800 Microprocessor

This section was taken from Reference 3. The NSC800 Microprocessor is an 8-bit CMOS microprocessor that functions as the central processing unit (CPU) in the first controller design. It is fully compatible with the Z80 instructions set featuring a powerful set of 158 instructions with 10 addressing modes and 22 internal registers. It features unique power-save function, multiplexed bus structure, Schmitt trigger input on reset, on-chip bus controller and clock generator, on-chip 8-bit dynamic RAM refresh circuitry and finally a speed of 1 s instruction cycle at 4.0 MHz. It is capable of addressing 64 kilobytes of memory and 256 input/output (I/O) devices and it has five interrupt lines on-chip. The NSC800 uses a multiplexed bus for data and addresses. The 16-bit address bus is divided into a high-order 8-bit address bus that handles bits 8-15 of the address, and a low-order 8-bit multiplexed address/data bus that handles bits 0-7 of the address and bits 0-7 of the data. Strobe outputs from the NSC800 (ALE, $\overline{RD}$ and $\overline{WR}$) indicate when a valid address or data is present on the bus. Input/Output/Memory (IO/$\overline{M}$) indicates whether the ensuing cycle accesses memory or I/O. During an input or output instruction, the CPU duplicates the lower half of the address [AD(0-7)] onto the upper half [A(8-15)]. The eight bits of address will stay on A(8-15) for the entire machine cycle. Figure 2.14 shows the timing diagram for the opcode fetch cycles with and without a wait state.

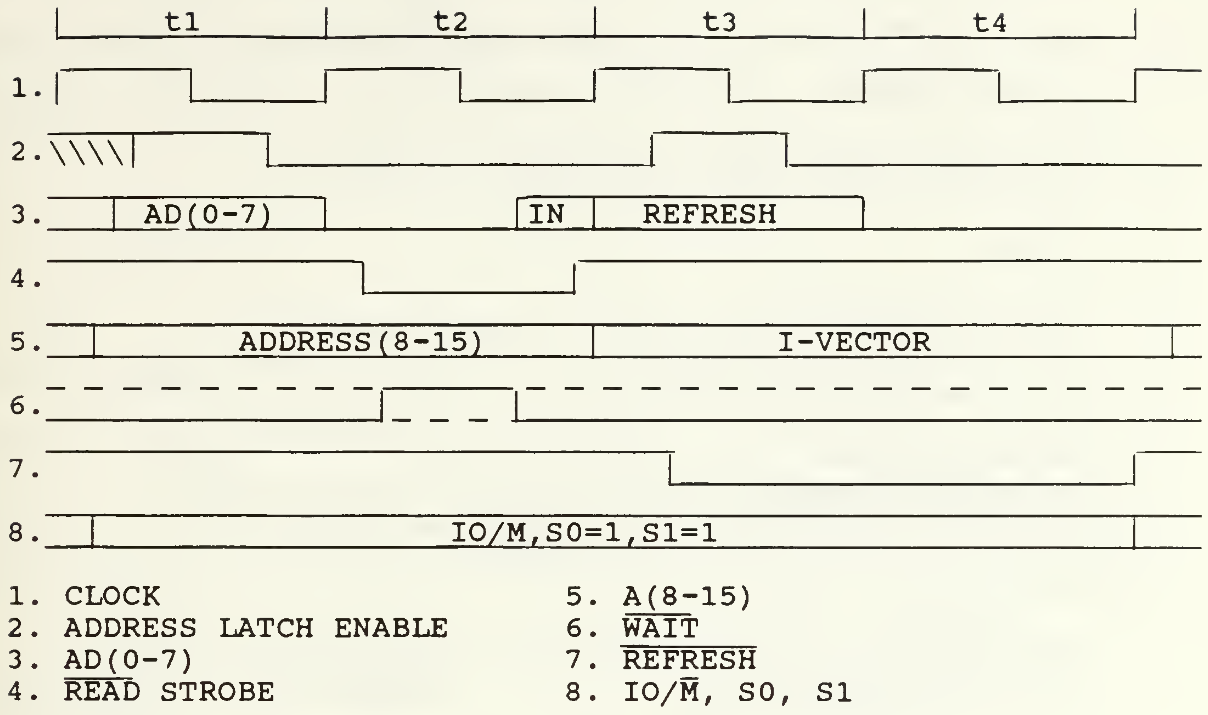
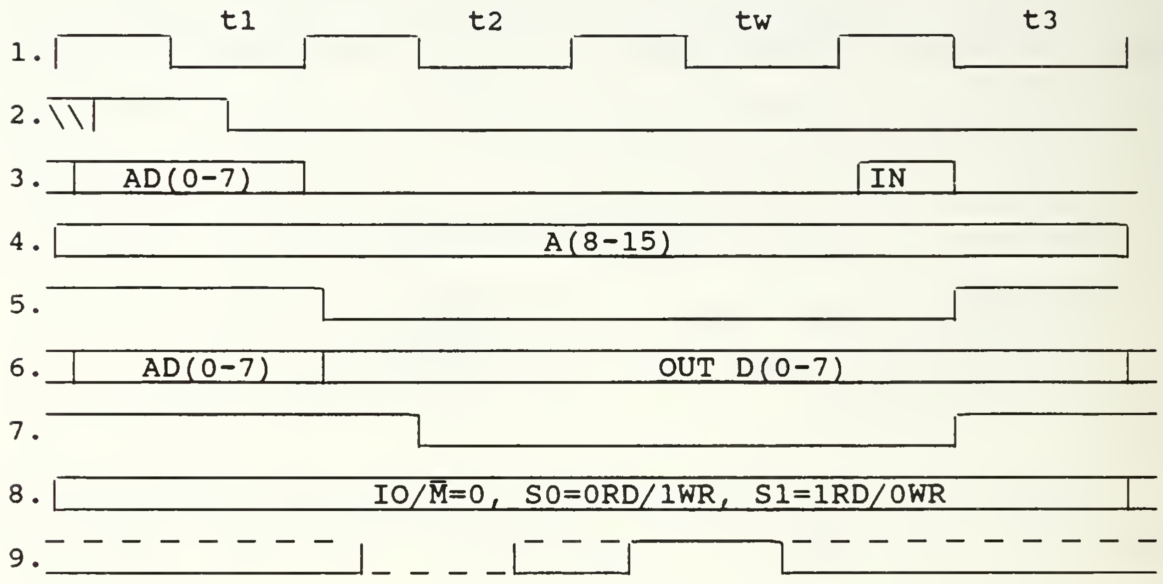

t1
t2
t3
t4
1.
2.
3.
4.
5.
6.
7.
8.
AD(0-7)
IN
REFRESH
ADDRESS(8-15)
I-VECTOR
IO/M,S0=1,S1=1
1. CLOCK
2. ADDRESS LATCH ENABLE
3. AD(0-7)
4. $\overline{\text{READ}}$ STROBE
5. A(8-15)
6. $\overline{\text{WAIT}}$
7. $\overline{\text{REFRESH}}$
8. IO/$\overline{\text{M}}$, S0, S1

Figure 2.14 Opcode Fetch Cycles with and without $\overline{\text{WAIT}}$ States [Ref. 3]

During the opcode fetch, the CPU places the contents of the PC on the address bus. The falling edge of Address Latch Enable (ALE) indicates a valid address on the AD(0-7) lines. The $\overline{\text{WAIT}}$ input is sampled during t2 and if active causes the NSC-800 to insert a wait state. $\overline{\text{WAIT}}$ is sampled again during this wait state so that when it goes inactive the CPU continues its opcode fetch by latching in the data on the rising edge of $\overline{\text{RD}}$ from the AD(0-7) lines. During t3, $\overline{\text{RFSH}}$ goes active and AD(0-7) has the dynamic RAM refresh address from register R and A(8-15) the interrupt vector from register I (see Appendix A).

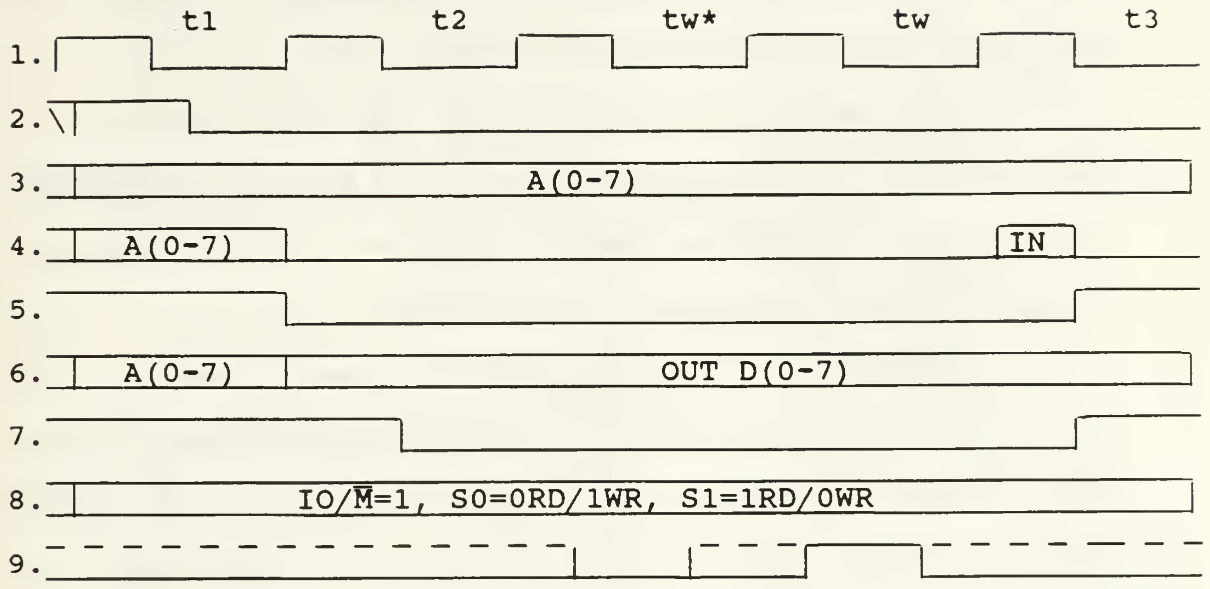
Figure 2.15 shows the timing for memory read and write cycles with and without a wait state. The $\overline{RD}$ strobe is widened by half the machine state for memory reads so that the actual latching of the input data occurs later.



1. CLOCK
2. ADDRESS LATCH ENABLE
3. AD (0-7)
4. A (8-15)
5. $\overline{\text{READ}}$ STROBE
6. AD (0-7)
7. $\overline{\text{WRITE}}$ STROBE
8. IO/$\overline{\text{M}}$, S0, S1
9. $\overline{\text{WAIT}}$

Figure 2.15 Memory Read and Write with and without $\overline{\text{WAIT}}$ States [Ref. 3]

Figure 2.16 shows the timing for input and output cycles with and without wait states. The CPU automatically inserts one wait state into each I/O instruction to allow sufficient time for an I/O port to decode the address.



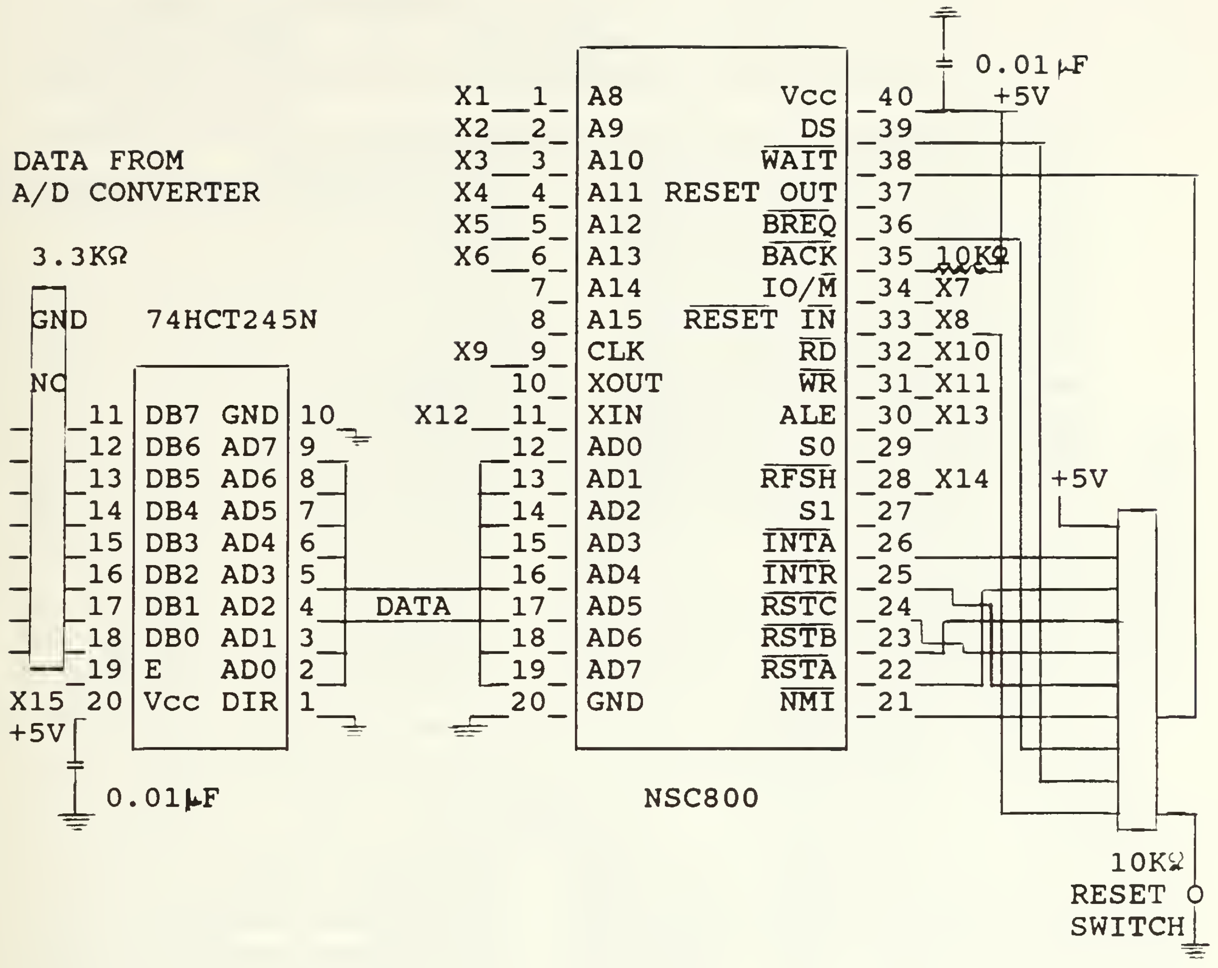
* WAIT state automatically inserted during IO operation

1. CLOCK	6. AD(0-7)
2. ADDRESS LATCH ENABLE	7. $\overline{\text{WRITE}}$ STROBE
3. A(8-15)	8. IO/$\overline{\text{M}}$, S0, S1
4. AD(0-7)	9. $\overline{\text{WAIT}}$
5. $\overline{\text{READ}}$ STROBE	

Figure 2.16 Input and Output Cycles with and without $\overline{\text{WAIT}}$ States [Ref. 3]

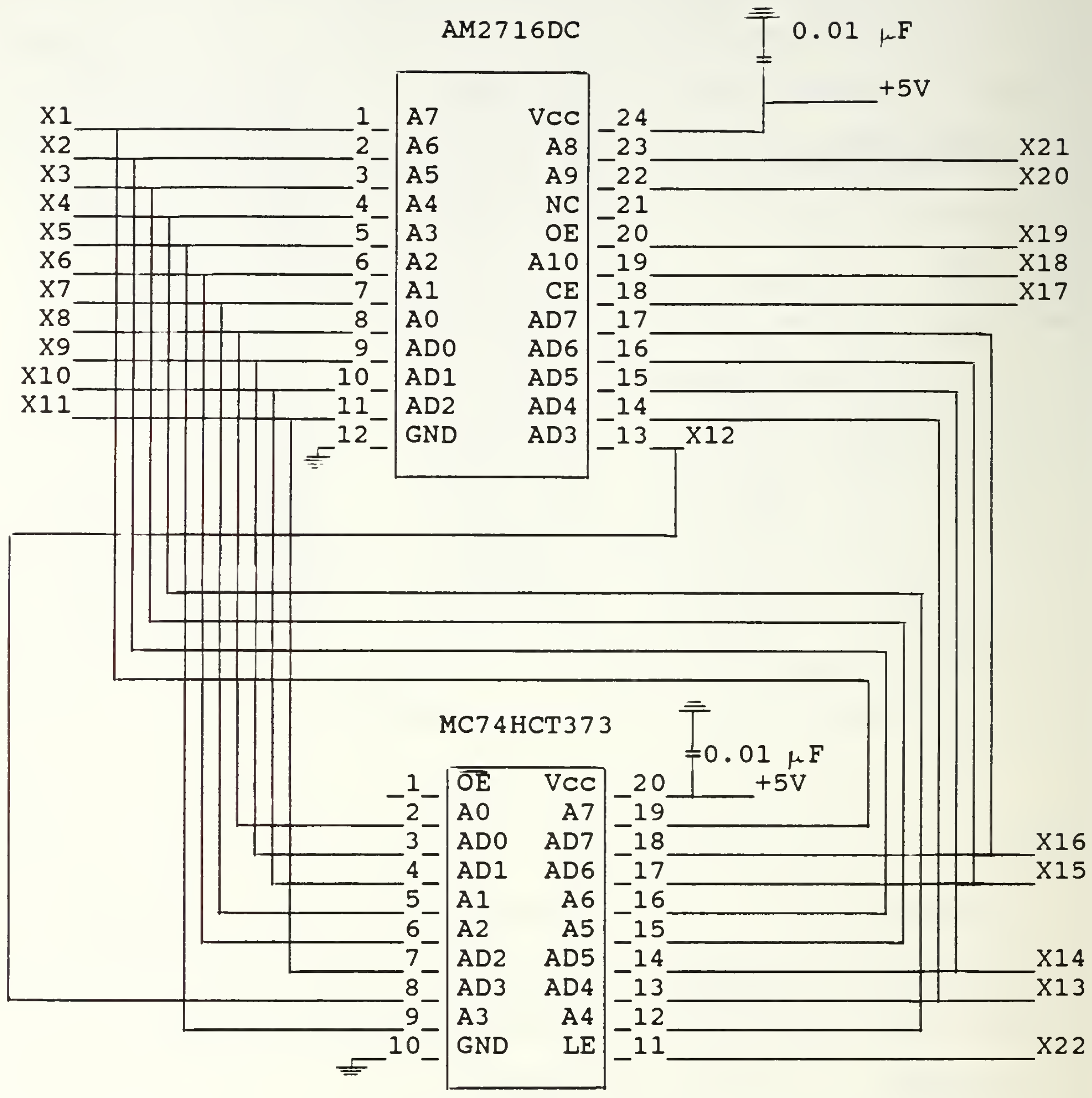
The NSC800 has five interrupt/restart inputs, four are maskable ($\overline{\text{RSTA}}$, $\overline{\text{RSTB}}$, $\overline{\text{RSTC}}$ and $\overline{\text{INTR}}$) and one is non-maskable ($\overline{\text{NMI}}$). $\overline{\text{NMI}}$ has the highest priority of all interrupts. $\overline{\text{NMI}}$ cannot be disabled. After recognizing an active input on $\overline{\text{NMI}}$, the CPU stops before the next instruction, pushes the Program Counter (PC) onto the stack, and jumps to address X'0066, where the user's interrupt service routine is located. $\overline{\text{NMI}}$ is intended for interrupts requiring immediate attention, (power down), $\overline{\text{RSTA}}$, $\overline{\text{RSTB}}$ and $\overline{\text{RSTC}}$ are restart inputs, which if enabled, execute a restart to memory location X'003C, X'0034 and X'002C, respectively. Note that the CPU response to the $\overline{\text{NMI}}$ and $\overline{\text{RST}}$ ($\bar{\text{A}}$, $\bar{\text{B}}$, $\bar{\text{C}}$) request input is basically identical, except for the restored memory location. Unlike $\overline{\text{NMI}}$, however, restart request input must be enabled.

Referring to the controller circuit, Figure 2.17 shows the detailed circuit wiring for the Microprocessor section. The data bits from the A/D converter are driven to the NSC800 Microprocessor through the 74HCT245N chip which is an octal bus transmitter/receiver designed for 8 line asynchronous 2-way data communication between data buses. The controller's memory section is illustrated in Figure 2.18. The AM2716DC is a 16,384-bit Electrically Erasable Read Only Memory (EPROM) which has a fast read access time of 250 ns maximum. The entire memory can be erased in 9 ms allowing the total time to rewrite all 2 kilobytes to be cut by 50%. In order for the controller to function as wanted, a program had to be stored inside the EPROM (see Appendix B). The MC74HCT373 chip consists of eight latches with 3-state outputs for bus organized system applications. The flip-flops appear transparent to the data (data changes asynchronously) when Latch Enable (LE) is high. When LE is low, the data that meets the setup times is latched. Data appears on the bus when the Output Enable ($\overline{\text{OE}}$) is low. When $\overline{\text{OE}}$ is high the bus output is in the high impudence state. Because the EPROM is a relatively



X1-X2-X3.	TO PINS 23-22-19 OF THE EPROM
X4-X5-X6-X7-X14-X15.	TO PINS 1-2-3-5-10 OF THE MM74HC138N-B
X8.	TO PIN 22 OF THE ENCODER
X9-X12.	CLOCKS
X10.	TO PIN 15 OF THE A/D CONVERTER
X11.	TO PIN 27 OF THE RAM
X13.	TO PIN 11 OF THE MC74HCT373

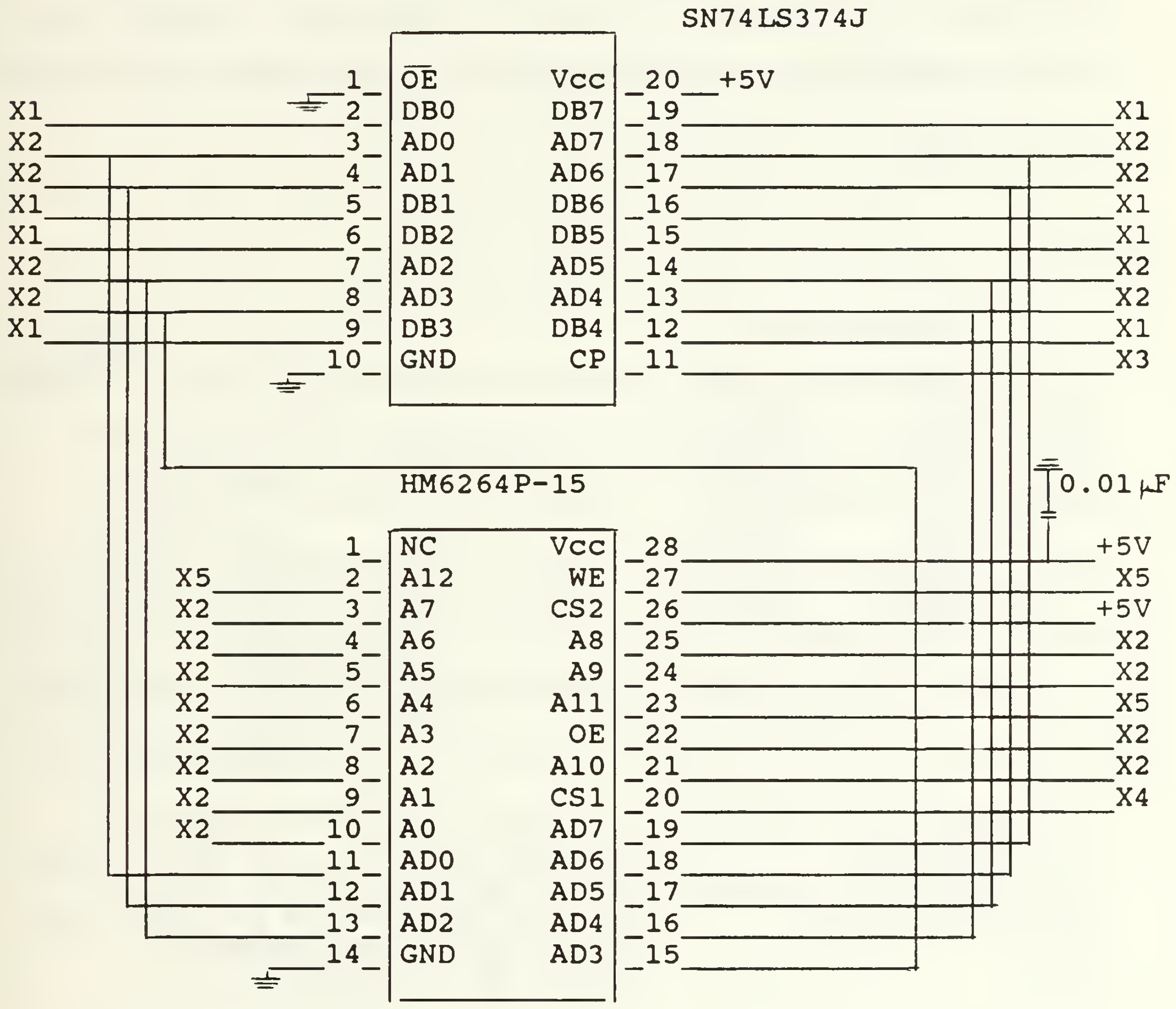
Figure 2.17 NSC800 Microprocessor Section



X1-X2-X3-X4.	TO PINS 4-3-2-1 OF THE MM74HC138-A
X9-X10-X11-X12-X13.	TO PINS 2-3-4-5-6 OF THE 74HCT245N
X14-X15-X16.	TO PINS 7-8-9 OF THE 74HCT245N
X18-X19-X20-X21-X22.	TO PINS 9-32-2-1-30 OF THE NSC800
X17.	TO PIN 15 OF THE MM74HC138N-B
X1-16.	TO PINS (3-13) & (15-19) OF THE RAM
X18-X19-X20-X21.	TO PINS 21-22-24-25 OF THE RAM

Figure 2.18 Controller's Memory Section

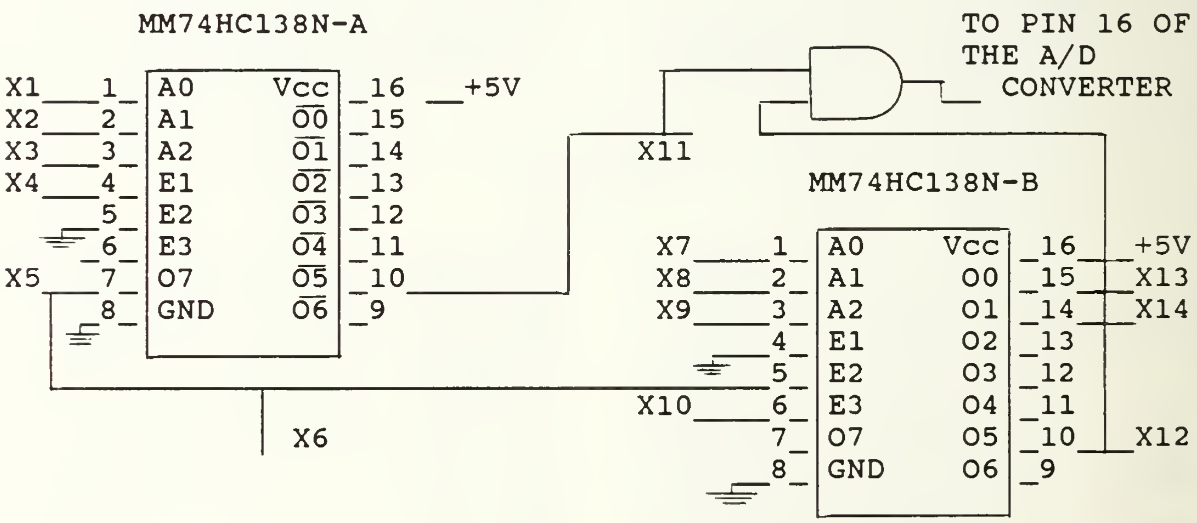
slow processing device and this would affect the final results (bandwidth) the HM6264P-15, an 8,192-word by 8-bit high static CMOS Random Access Memory chip was used because of the faster access time, 100-150 ns. Figure 2.19 illustrates the hard-wiring of the RAM plus the output data latch.


SN74LS374J
OE Vcc DB0 DB7 AD0 AD7 AD1 AD6 DB1 DB6 DB2 DB5 AD2 AD5 AD3 AD4 DB3 DB4 GND CP
+5V
X1 X2 X3
HM6264P-15
NC Vcc A12 WE A7 CS2 A6 A8 A5 A9 A4 A11 A3 OE A2 A10 A1 CS1 A0 AD7 AD0 AD6 AD1 AD5 AD2 AD4 GND AD3
X2 X4 X5
0.01 μF

X1. OUTPUT DATA BITS X2. TO EPROM
X3-X4. TO PINS 10-14 OF THE MM74HC138N-A X5. TO NSC800

Figure 2.19 Controller's Memory and Output Latch

The SN74LS374J is a high speed, low power octal D-type flip-flop featuring separate D-type inputs for each flip-flop and 3-state outputs for bus oriented applications. The last section in the controller is the control section, Figure 2.20. The circuit consists of two MM74HC138N and a SN74HC08N chips. The MM74HC138N is a high speed 1-of-8 Decoder/Demultiplexer. This device is ideally suited for high speed bipolar memory chip select address decoding and this was the reason for its selection.



X1-X2-X3-X4.	TO PINS 4-3-2-1-18 OF THE EPROM
X5-X6-X7-X8-X9-X10-X11.	TO THE ENCODER
X6-X7-X8-X9-X10.	TO PINS 34-4-5-6-28 OF THE NSC800
X12.	TO PIN 19 OF THE 74HCT245
X14.	TO PIN 20 OF THE RAM

Figure 2.20 Control Section

3. The Z8400 Microprocessor

The second controller uses the Z8400 Microprocessor, made by Zilog, as CPU. The following text was taken from Reference 4. It features an instruction set which contains 158 instructions. Its 2.5 MHz clock results in rapid instruction execution with consequent high data throughput. The extensive instruction set includes string, bit, byte and word operations. Block searches and block transfers together with indexed and relative addressing result in the most powerful data handling capabilities. The Microprocessor is linked by a vector interrupt system. This system may be daisy-chained to allow implementation of a priority interrupt scheme. Little, if any, additional logic is required for daisy-chaining. Duplicate sets of both general-purpose and flag registers are provided, easing the design and operation of system software through single-context switching, background-foreground programming, and single-level interrupt processing. In addition, two 16-bit index registers facilitate program processing of tables and arrays. There are three modes of high speed interrupt processing. The Z8400 CPU executes instructions through a specific sequence of operations:

- Memory read or write
- I/O device read or write
- Interrupt acknowledge

The basic clock period is referred to as a T time or cycle and three or more T cycles make up a machine cycle (M). The CPU places the contents of the Program Counter (PC) on the address bus at the start of the cycle, see Figure 2.21.

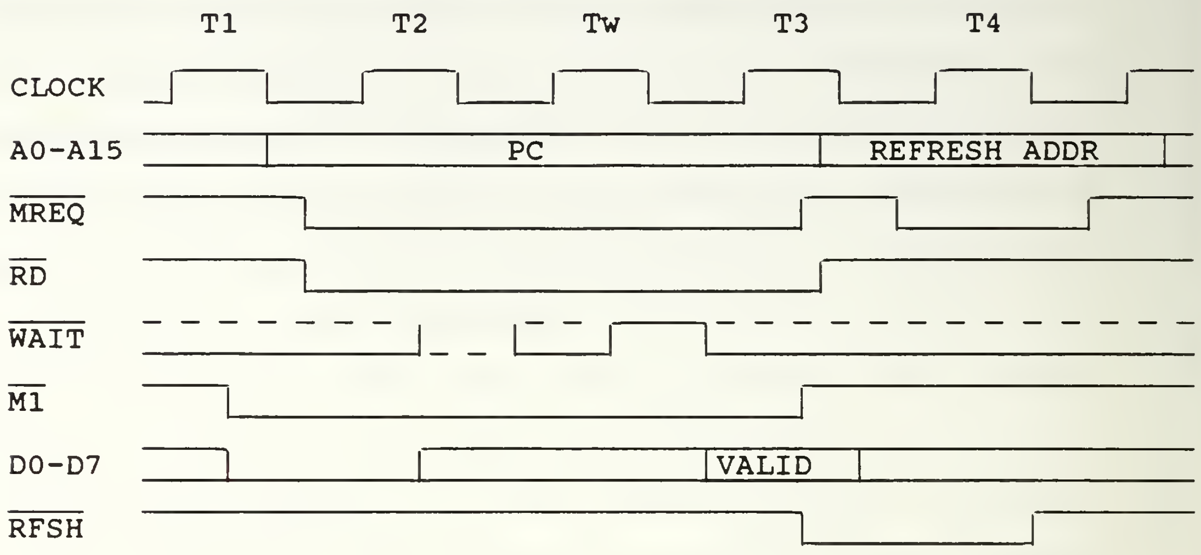


Figure 2.21 Instruction Opcode Fetch [Ref. 4]

Approximately one-half clock cycle later, $\overline{\text{MREQ}}$ goes active. When active, $\overline{\text{RD}}$ indicates that the memory data can be enabled onto the CPU data bus. The CPU samples the $\overline{\text{WAIT}}$ input with the falling edge of clock state T2. During clock states T3 and T4 of an $\overline{\text{M1}}$ cycle dynamic RAM refresh can occur while the CPU starts decoding and executing the instruction. When the Refresh Control signal becomes active, refreshing of dynamic memory can take place. Figure 2.22 shows the timing of memory read or write cycles other than opcode fetch ($\overline{\text{M1}}$) cycle. The $\overline{\text{MREQ}}$ and $\overline{\text{RD}}$ signals function exactly as in the fetch cycle. In cycle, $\overline{\text{MREQ}}$ also becomes active when the address bus is stable, so that it can be used directly as an R/W pulse to most memories. Figure 2.23 shows the timing for an I/O read or I/O write operation.

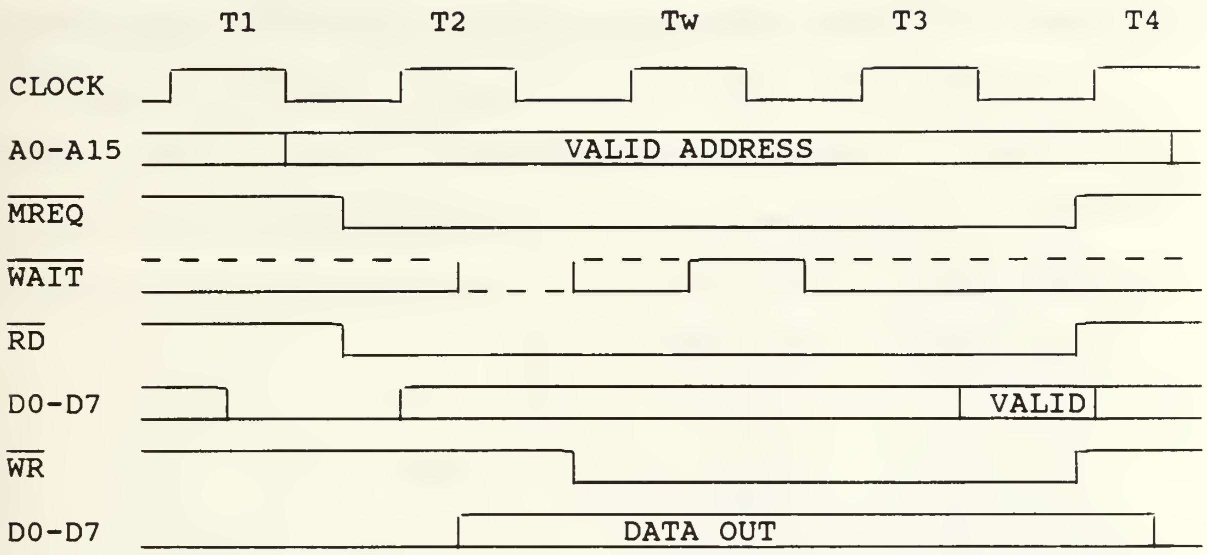


Figure 2.22 Memory Read or Write Cycles [Ref. 4]

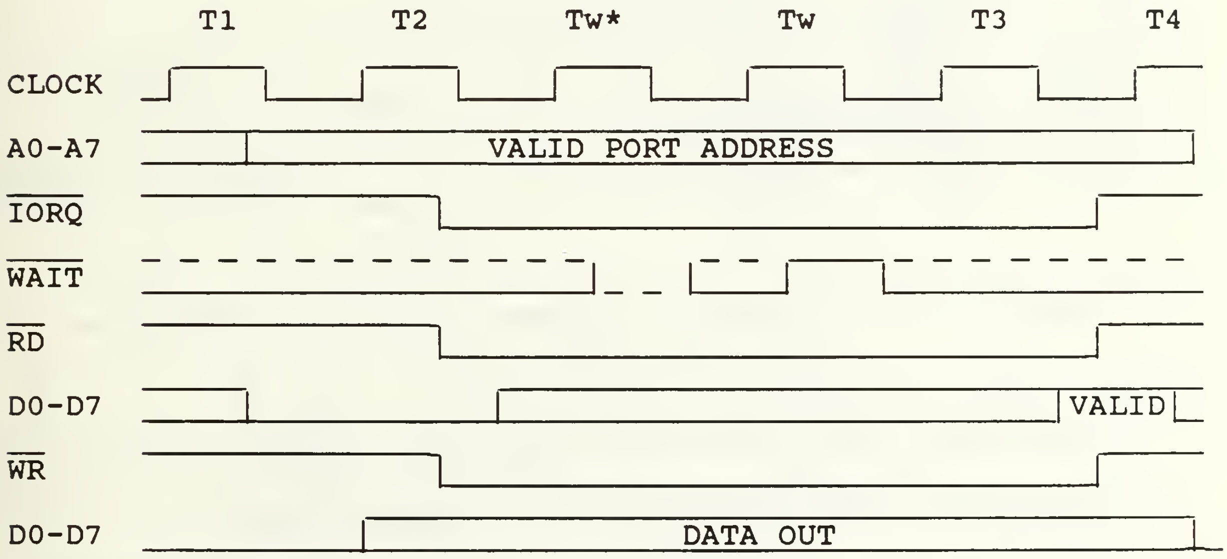


Figure 2.23 Input or Output Cycles [Ref. 4]

During I/O operations, the CPU automatically inserts a single wait state Tw. This extra wait state allows sufficient time for an I/O port to decode the address from the port address lines. The CPU samples the interrupt signal with the rising edge of the last clock cycle at the end of any instruction, see Figure 2.24. When an interrupt is accepted, a special $\overline{M1}$ cycle is generated. During this $\overline{M1}$ cycle, $\overline{IORQ}$ becomes active to indicate that the interrupting device can place an 8-bit vector on the data bus. The CPU automatically adds two wait states to this cycle. [Ref. 4]

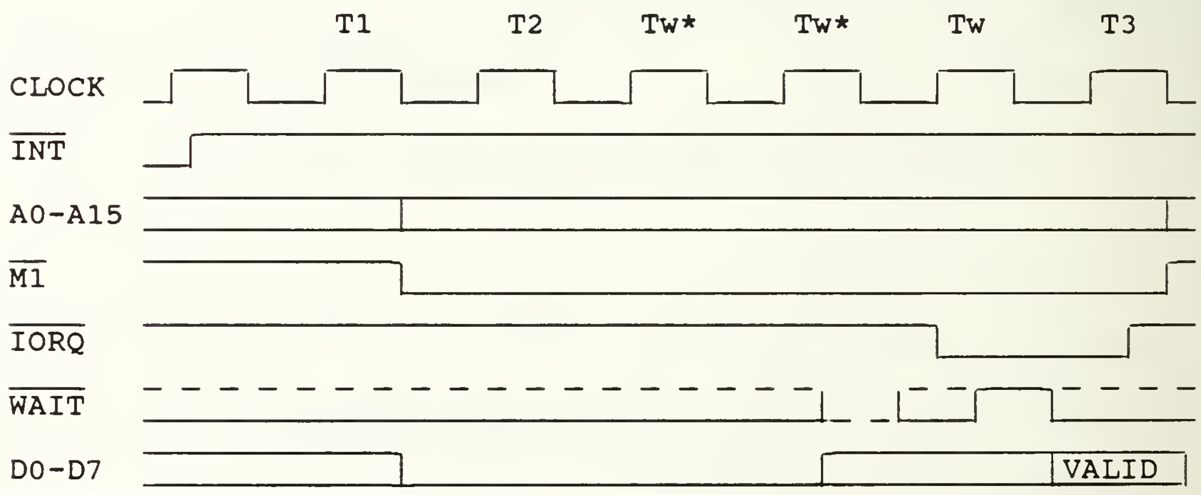
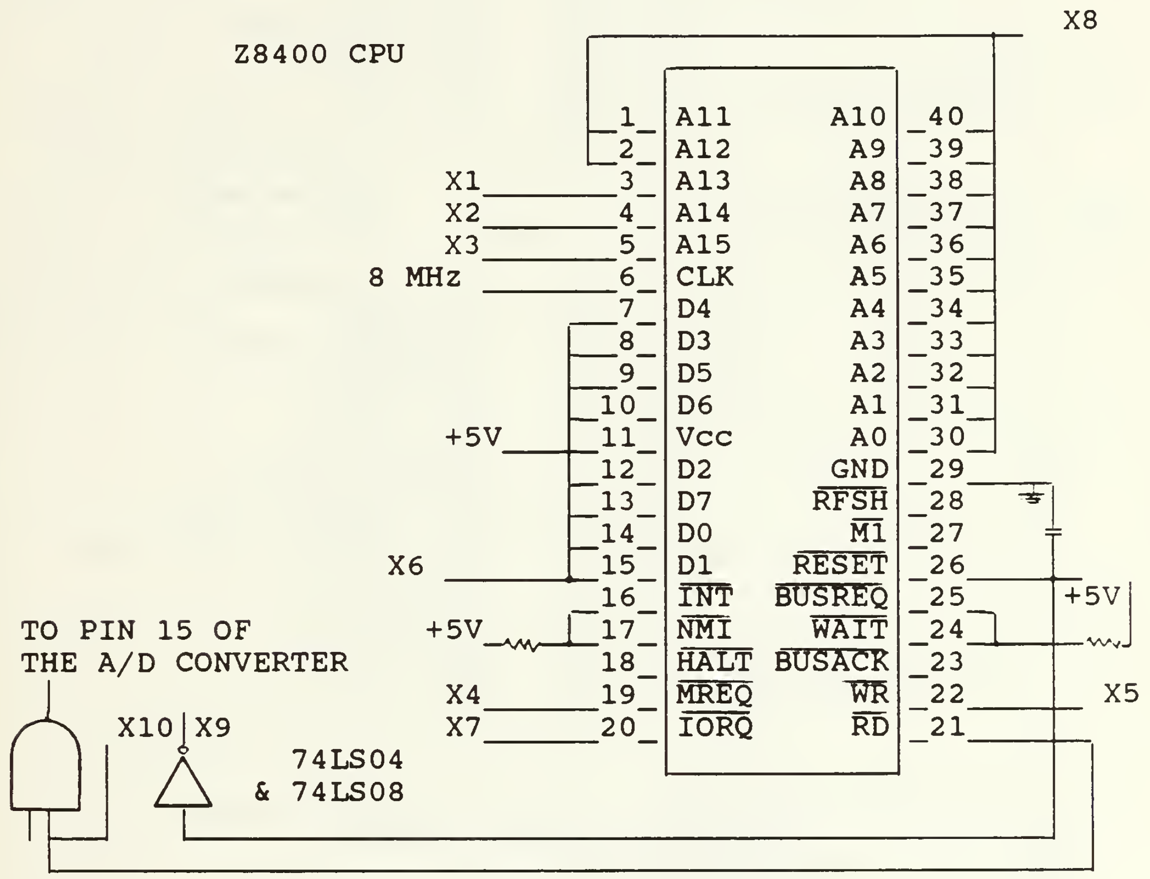


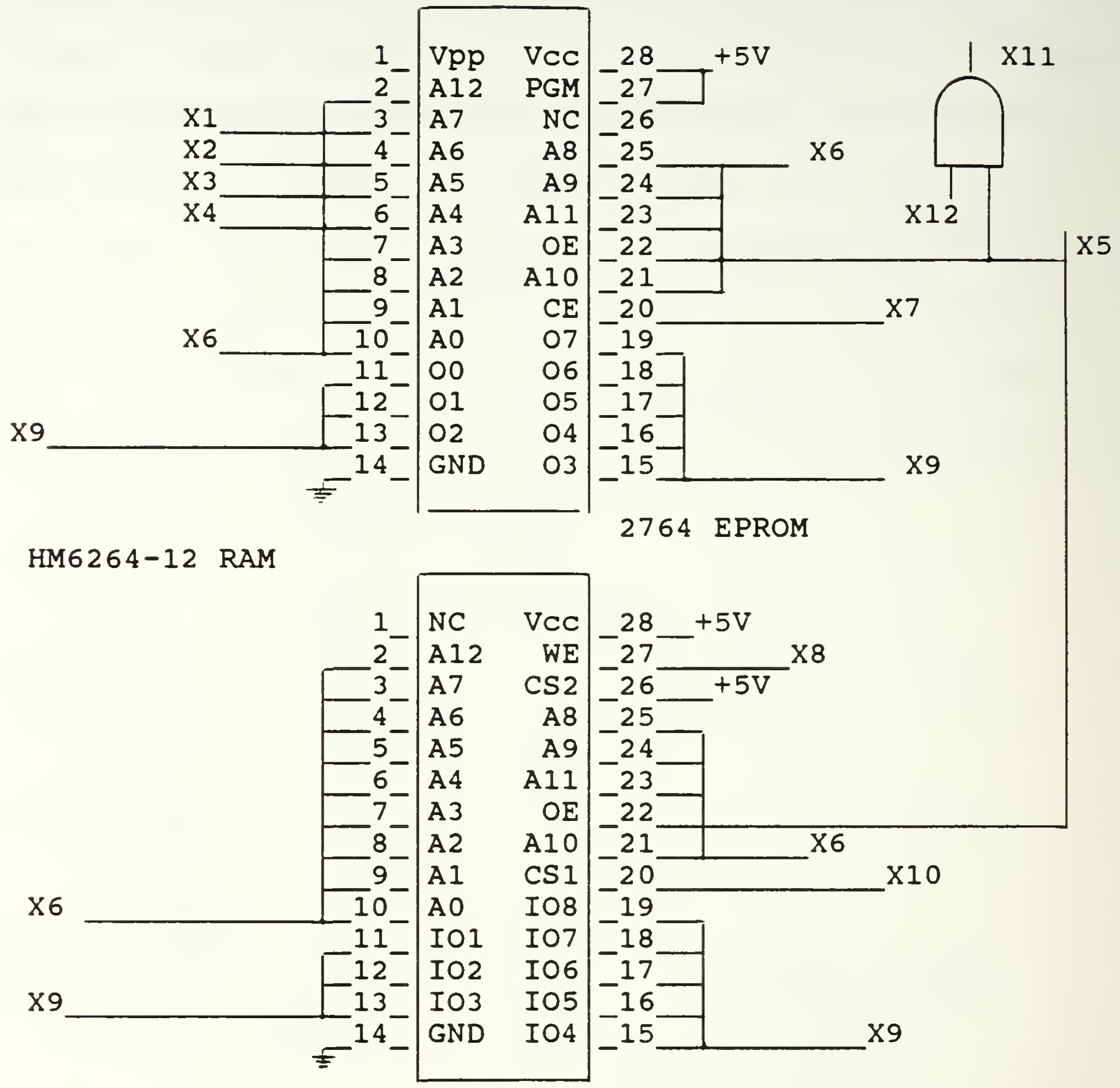
Figure 2.24 Interrupt Request/Acknowledge Cycle [Ref. 4]

Referring to the controller's circuit, Figure 2.25 shows the Z8400 CPU section and wiring. The memory section uses an 2764 EPROM with memory capacity of 65,536-bit and provides an access time of 180 ns. It also has the HM6264-12 RAM for faster processing and features the same specifications as the previous HM6264-12 RAM (see II-D.2). Figure 2.26 shows the memory section chips and wiring. The last section in this controller is the control section which has two SN74LS138 chips and a SN74LS032. Figure 2.27 shows the control section chips and wiring.



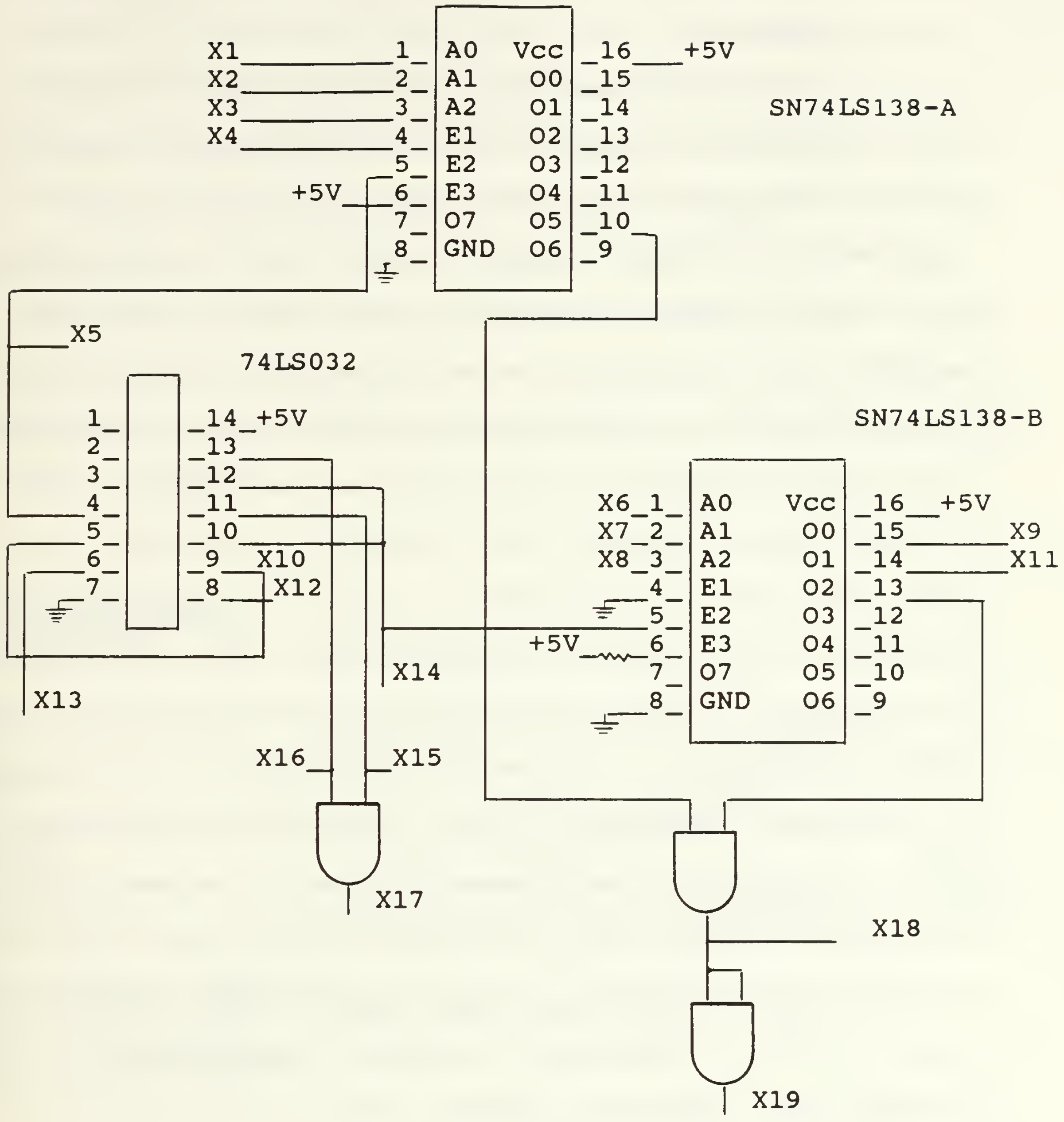
X1-X2-X3.	TO PINS 1-2-3 OF THE SN74LS138-B
X4.	TO PIN 5 OF THE SN74LS138-B & PIN 12 OF THE 74LS032
X5.	TO PINS 5-9 OF THE 74LS032
X6.	DIGITAL DATA BITS
X7.	TO PIN 5 OF THE SN74LS138-A & PIN 4 OF THE 74LS032
X8.	ADDRESS BITS
X9.	TO PINS 19-22 OF THE HD-15531
X10.	TO PIN 13 OF THE 74LS032

Figure 2.25 Microprocessor Section



X1-X2-X3-X4.	TO PINS 4-3-2-1 OF THE SN74LS138-A
X5.	TO PIN 11 OF THE 74LS032
X6.	ADDRESS BITS
X7.	TO PIN 15 OF THE SN74LS138-B
X8.	TO PIN 8 OF THE 74LS032
X9.	DATA BITS
X10.	TO PIN 14 OF THE SN74LS138-B
X11.	TO PIN 15 OF THE A/D CONVERTER
X12.	FROM PIN 21 OF THE Z8400 CPU

Figure 2.26 Memory Section



X1-X2-X3-X4-X9-X15.	TO PINS 6-5-4-3-20-22 OF THE 2764 EPROM
X5-X6-X7-X8-X10.	TO PINS 20-3-4-5-22 OF THE Z8400 CPU
X14-X16.	TO PINS 19-21 OF THE Z8400 CPU
X11-X12.	TO PINS 20-27 OF THE HM6264-12 RAM
X13.	TO PIN 1 OF THE SN74LS165 ENCODER SHIFT REGISTER
X17-X18.	TO PINS 15-16 OF THE A/D CONVERTER
X19.	TO PIN 29 OF THE HD-15531

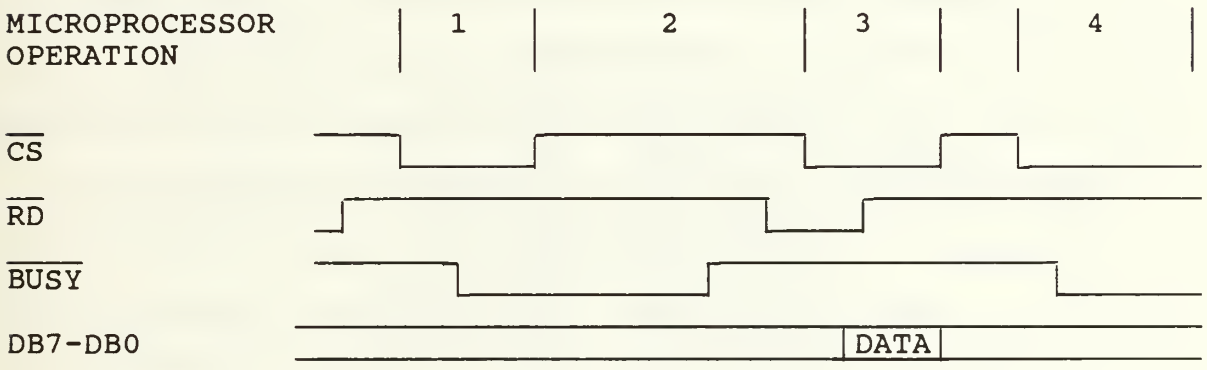
Figure 2.27 Control Section

E. CONVERSION

1. ADC-9008

This section was taken from Reference 5. The ADC-9008 is a monolithic successive-approximation analog-to-digital converter. The ADC-9008 outputs use three-state logic, allowing direct connection to the data bus or system input port. With its improved bus relinquish time, the ADC-9008 permits faster bus operation and consequent software economy. The ADC-9008 uses active-low chip select (CS) and READ/WRITE (RD) control signals for all operation. Depending on the control timing waveforms, the ADC-9008 is interfaced like static RAM, ROM or slow memory. With its on-board comparator, interface logic, optional internal clock, and +5V operation, the ADC-9008 is a low cost solution for microprocessor based 8-bit A/D systems. In all modes, the ADC-9008 is initialized by executing a READ instruction to the ADC-9008 address. The data obtained should be ignored. In static RAM mode, input $\overline{CS}$ is derived from the ADC-9008 address decoder and input $\overline{RD}$ is derived from the active low memory READ signal, see Figure 2.28. To start a conversion, execute a memory WRITE to the ADC-9008. The completed conversion data is obtained by executing a memory READ to the ADC-9008. During conversion, output $\overline{BUSY}$ will be low. We can read the data until $\overline{BUSY}$ returns high. The required minimum time between WRITE and READ is obtained by including NOP or other program instruction. It is important that WRITE and READ be alternately executed. A

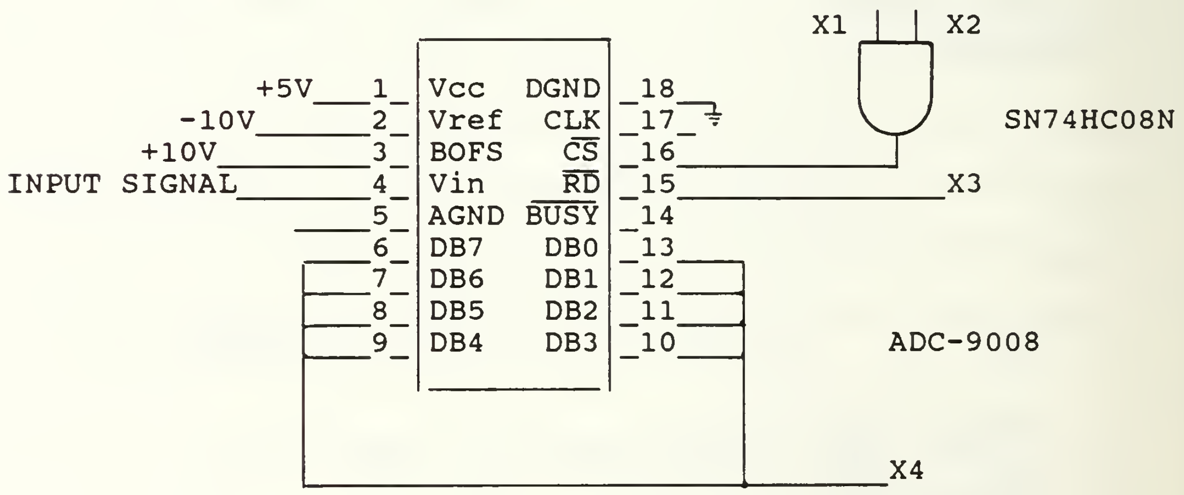
WRITE instruction has no effect unless the results of the previous WRITE have already been read. Once data has been read, the ADC-9008 is internally reset. A new conversion must be started using WRITE, and the conversion completed, before a new READ will produce valid data. The static RAM interface mode offers advantages in a tightly controlled software environment, where the relationship between WRITE and READ instruction pairs is certain. As long as minimum timing is satisfied, converted data may be read at any convenient time after conversion.



1. MEMORY WRITE
2. NOP OR OTHER INSTRUCTIONS UNTIL $\overline{\text{BUSY}}$ IS HIGH
3. MEMORY READ
4. MEMORY WRITE

Figure 2.28 Static RAM Mode Timing Diagram [Ref. 5]

The ADC-9008 actually converts the average voltage of inputs Vin and BOFS. This average voltage must be between 0V and the positive magnitude of the reference voltage Vref. If the analog input voltage range is 0V to | Vref |, then tie the Vin and BOFS pins together. Input 0V will correspond to code 00000000; input full scale will be code 11111111. Bipolar operation is obtained by using the BOFS input to offset the Vin input voltage. For example, with a -10V Vref, an offset voltage of +10V may be applied to BOFS. This offset technique will permit an analog signal range of -10V to +10V, with -10V corresponding to code 00000000 and positive full scale at code 11111111. [Ref. 5] Figure 2.29 shows the wiring circuit of the ADC-9008.



X1. TO PIN 10 OF THE MM74HC138N-A

X2. TO PIN 10 OF THE MM74HC138N-B

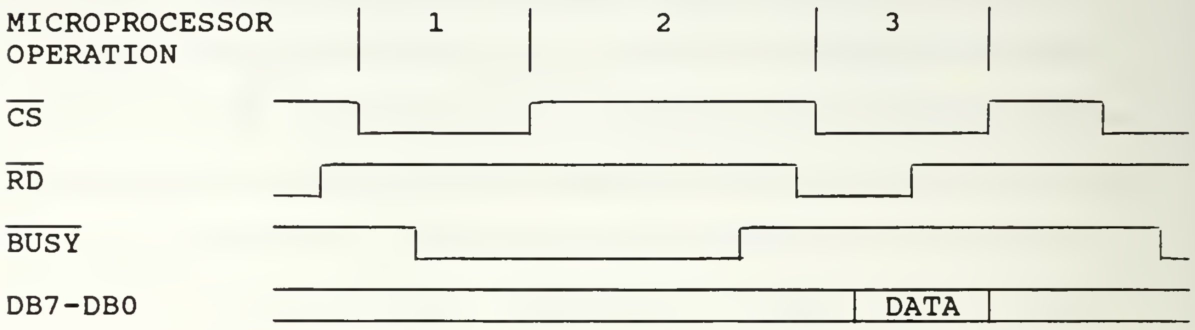
X3. TO PIN 32 OF THE NSC800 CPU

X4. DIGITAL DATA BITS

Figure 2.29 ADC-9008 Wiring Circuit

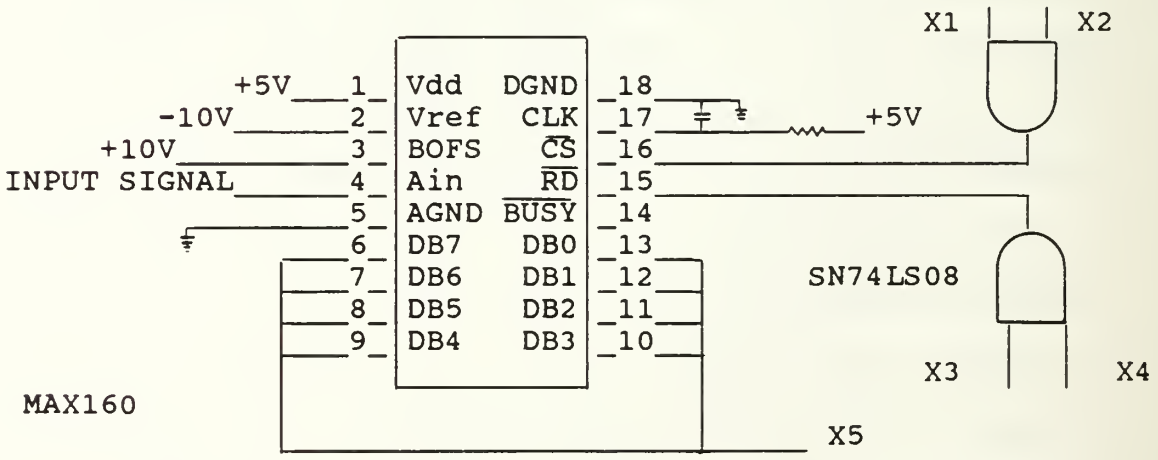
2. <u>MAX160CPN</u>

This section was taken from Reference 6. The MAX160 is a low cost, microprocessor compatible 8 bit analog to digital converter which use the successive approximation technique to achieve conversion times of 4 µs. This converter is an alternative design to that of the ADC-9008. It can easily interface with microprocessors by appearing as a memory location or I/O port without the need for external interfacing logic. Data outputs use latched, three-state buffer circuitry to allow direct connection to a microprocessor data bus or system input port. Operation is simplified by an on-chip clock, +5V power supply and low supply current (5mA maximum). The MAX160 uses the successive approximation technique to convert an unknown analog input to an 8 bit digital output code. When a start command is received from $\overline{\text{CS}}$ or $\overline{\text{RD}}$, $\overline{\text{BUSY}}$ goes low indicating that the conversion is in progress. Successive bits, starting with the most significant bit (MSB), are applied to the input of a DAC. The comparator determines whether the addition of the bit causes the DAC output to be larger or smaller than the analog input Ain. If the DAC output is greater than Ain, the trial bit is turned OFF, otherwise it is kept ON. Each successively smaller bit is tried and compared to Ain, in this manner until the least significant bit (LSB) decision has been made. When all bits have been tried, $\overline{\text{BUSY}}$ goes high, indicating that the conversion is ended and that the successive approximation register contains a valid representation of the analog input. The data can then be read using the $\overline{RD}$ input. Figure 2.30 shows static RAM mode timing diagram and Figure 2.31 the MAX160 wiring circuit.



1. MEMORY WRITE
2. NOP OR OTHER INSTRUCTIONS UNTIL $\overline{BUSY}$ IS HIGH
3. MEMORY READ

Figure 2.30 Static RAM Mode Timing Diagram [Ref. 6]

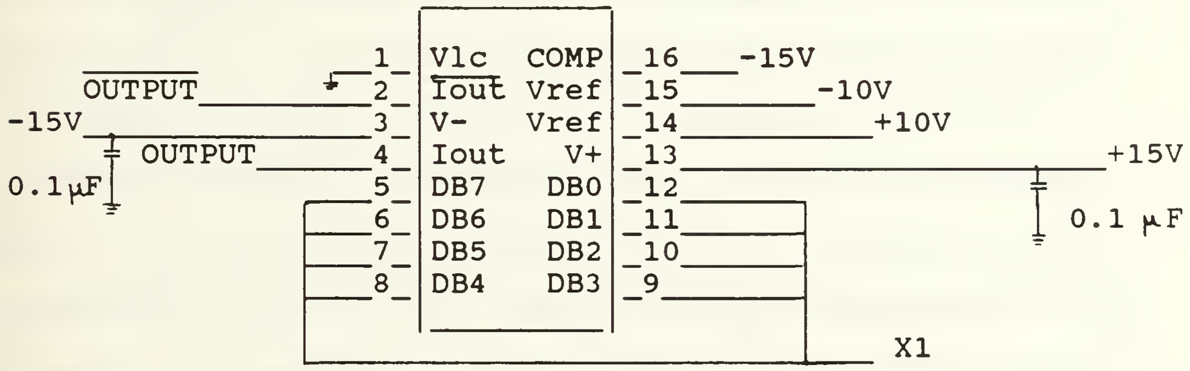


X1. FROM PIN 10 OF THE 74LS138-A
X2. FROM PIN 13 OF THE 74LS138-B
X3. FROM PIN 22 OF THE 2764 EPROM
X4. FROM PIN 21 OF THE Z8400 CPU
X5. DIGITAL DATA BITS

Figure 2.31 MAX160CPN Wiring Circuit

3. DAC0800

The following information was taken from Reference 7. The DAC08 is a monolithic 8-bit high speed current output digital to analog converter (DAC) featuring typical settling times of 100 ns. When used as a multiplying DAC, monotonic performance over a 40 to 1 reference current range is possible. The DAC08 also features high compliance complementary current outputs to allow differential output voltages of 20 Vp-p with simple resistor loads. The reference to full scale current matching of better than ±1 LSB eliminates the need for full scale trims in most applications while the nonlinearities of better than ±0.1% over temperature minimizes system error accumulations. The noise immune inputs of the DAC08 will accept TTL levels with the logic threshold pin, Vlc, pin 1 grounded. Simple adjustments of the Vlc potential allow direct interface to all logic families. Figure 2.32 shows the pin connections of the DAC0800.



X1. DIGITAL DATA BITS FROM SN74LS374J OUTPUT DATA LATCH

Figure 2.32 DAC0800 Wiring Circuit

F. SAMPLE AND HOLD CIRCUIT

The sample and hold circuit idea and the following section was taken from Reference 8, see Figure 2.33. The circuit uses the LT1016's high speed to improve upon a standard circuit function. The 200 ns acquisition time is well beyond monolithic sample and hold capabilities. This circuit also gets around many of the problems associated with standard sample and hold approaches, including FET switch errors and amplifier settling time. To achieve this, the LT1016's high speed is used in a circuit which completely abandons traditional sample and hold methods. When the sample and hold command line is high, Q2 conducts current, biasing Q3 on and forcing the 1000pF capacitor to discharge toward Q4's emitter potential. Q4's emitter, in turn, sits at a potential slightly below Q3's collector voltage. Q5 and the LT1009, biased from the input voltage, drive Q4. Concurrently, the TTL gate at the LT1016 grounds the latch pin and the comparator's inverting output goes high. When the sample and hold command line falls, Q2 and Q3 go off and the Q1 current source charge the 1000pF unit with fast linear ramp. This capacitor is buffered by Q7, a current sink loaded source follower. When Q7's output reaches the circuit's input value, the LT1016's inverting output switches low. The Q1 current source cuts off in about 2 ns and capacitor charging ceases. The LT1016's low state also means the NOR gate's output is high, latching the comparator's output. This prevents input line noise or a change in signal

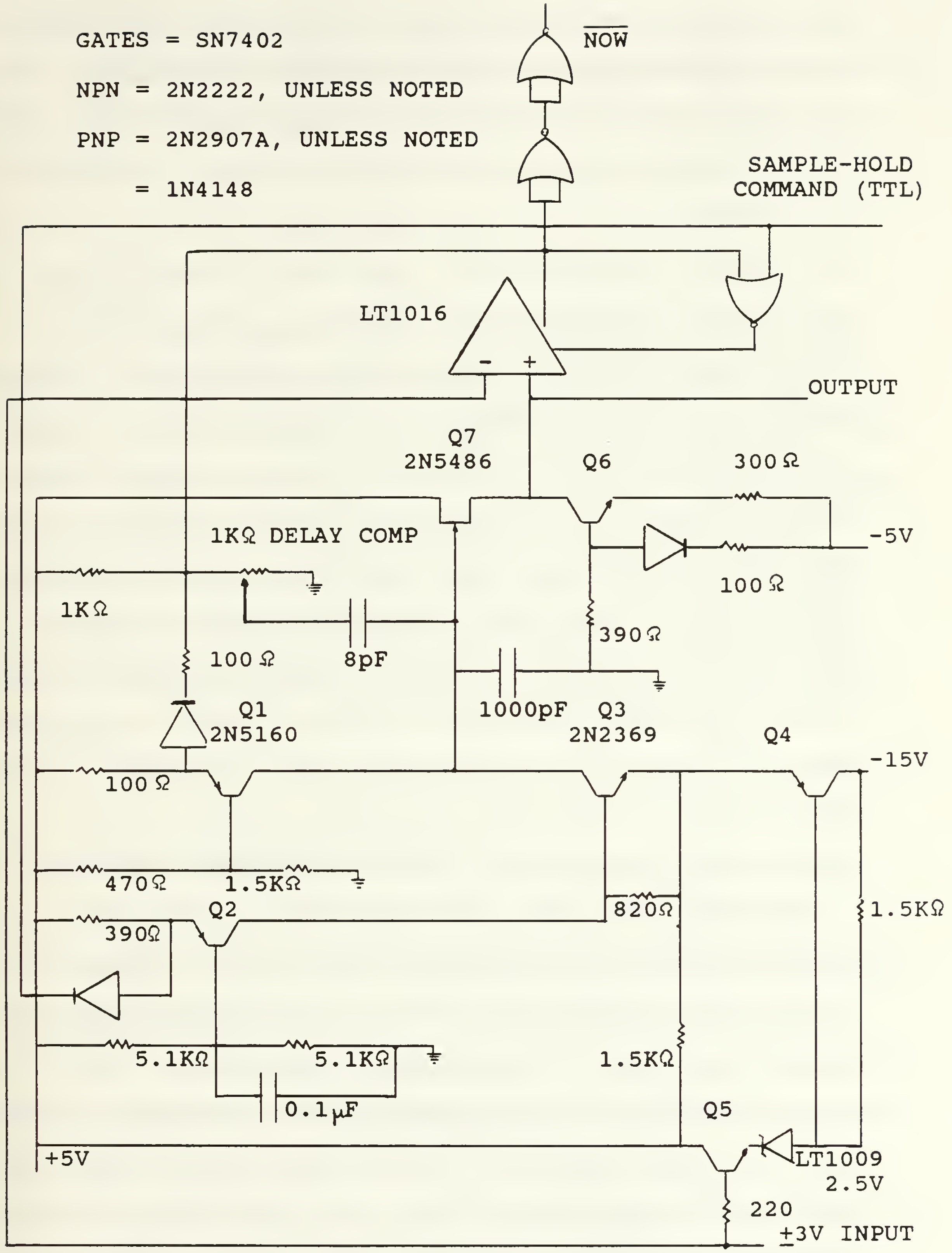


Figure 2.33 Sample and Hold Circuit [Ref. 8]

from affecting the stored value in the 1000pF hold capacitor. Ideally, Q7's output now sits at exactly the sampled value of the input voltage. In practice, a slight error exists because of the LT1016's delay and the turn-off time of Q1 (total 12 ns). Because of these delays, the capacitor is able to charge to a higher voltage than the input before current stops. This error term is compensated by removing a small quantity of charge from the 1000pF capacitor when the LT1016's inverting output goes low. The charge is removed through the 8pF-1KΩ potentiometer network. Because the charging ramp's slope is fixed, the error term is constant and the compensation works over the circuit's ±3V input common-mode range. When the LT1016 goes off, the ramp is seen to slightly overshoot its final value. The 1KΩ-8pF combination pulls enough charge out of the 1000pF hold capacitor to bring it back to the correct value. $\overline{\text{NOW}}$ falls low 2 gate delays after the LT1016 inverting output goes low. When this line goes low, the circuit's sampled output has settled from the correction transient and is valid data. The total time from the falling of the sample and hold line to the $\overline{\text{NOW}}$ output going low will always be inside 200 ns. The circuit's 200 ns acquisition time is due to the high slew rate of the charging ramp and the action of Q4, Q5 and the LT1009. These components form a wideband tracking amplifier whose output is always a fixed amount below the input. Q7's current source load (Q6) ensures that its Vgs does not change. Thus, Q3 will always reset the capacitor a small, relatively constant amount below any circuit input. In this way, the ramp does not have to run very long before it crosses the input value and acquisition time versus input voltage is constant.

G. FILTER

The filter idea was taken from Reference 9. The filter uses six operational amplifiers, the LF 356N, to filter the output from the DAC0800,the digital to analog converter. The design itself is not the best, but at least smooths out the output waveform and makes evaluation easier. The output waveform has twice the amplitude of the input in order to distinguish the noise components, amplitude distortion and phase distortion better. Figure 2.34 illustrates the filter circuit.

H. CHANNEL SELECTION CIRCUIT

The channel selection can be made by the DG 506A multiplexer. It provides 16-channel single ended multiplexing and demultiplexing of +15 volt analog signals. True bi-directional switch action takes place over the full analog signal, with Break-Before-Make operation to prevent momentary shorting of signal input. Thus, the input channel selection circuit is the same as the output channel selection circuit. A 4-bit binary counter, the SN74LS93N, provides the channel

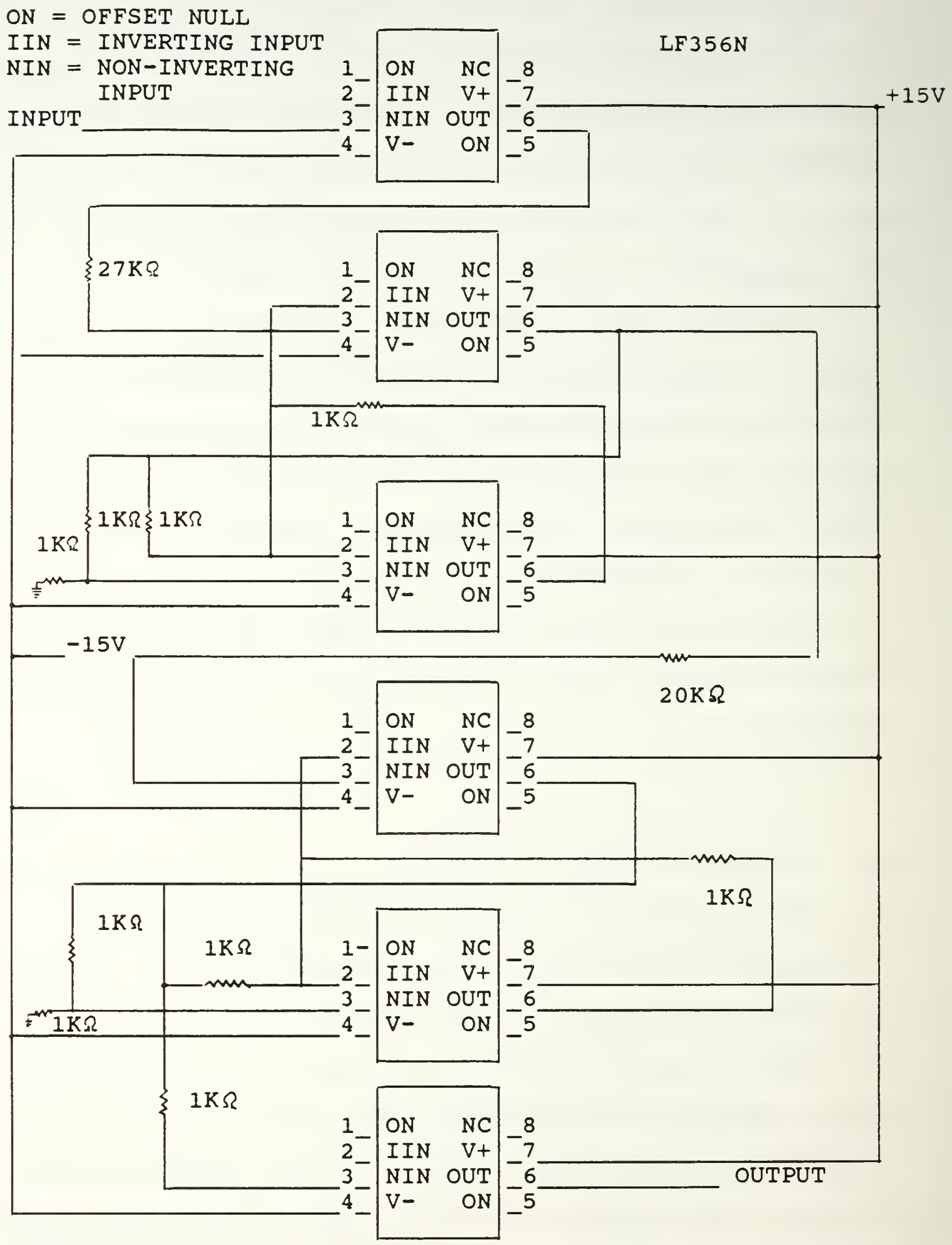


Figure 2.34   Filter Circuit

addresses and chip enable. The channel synchronization between the input and the output can be controlled by the controller. By initiating the operation of the system each channel is multiplexed and send through a buffer amplifier, the LH0033CJ, to the sample and hold circuit. At the output, the same signal is demultiplexed and send to the proper channel. Instructions given by the controllers assure the channel identification. Figure 2.35 illustrates the input/output channel selection circuit.

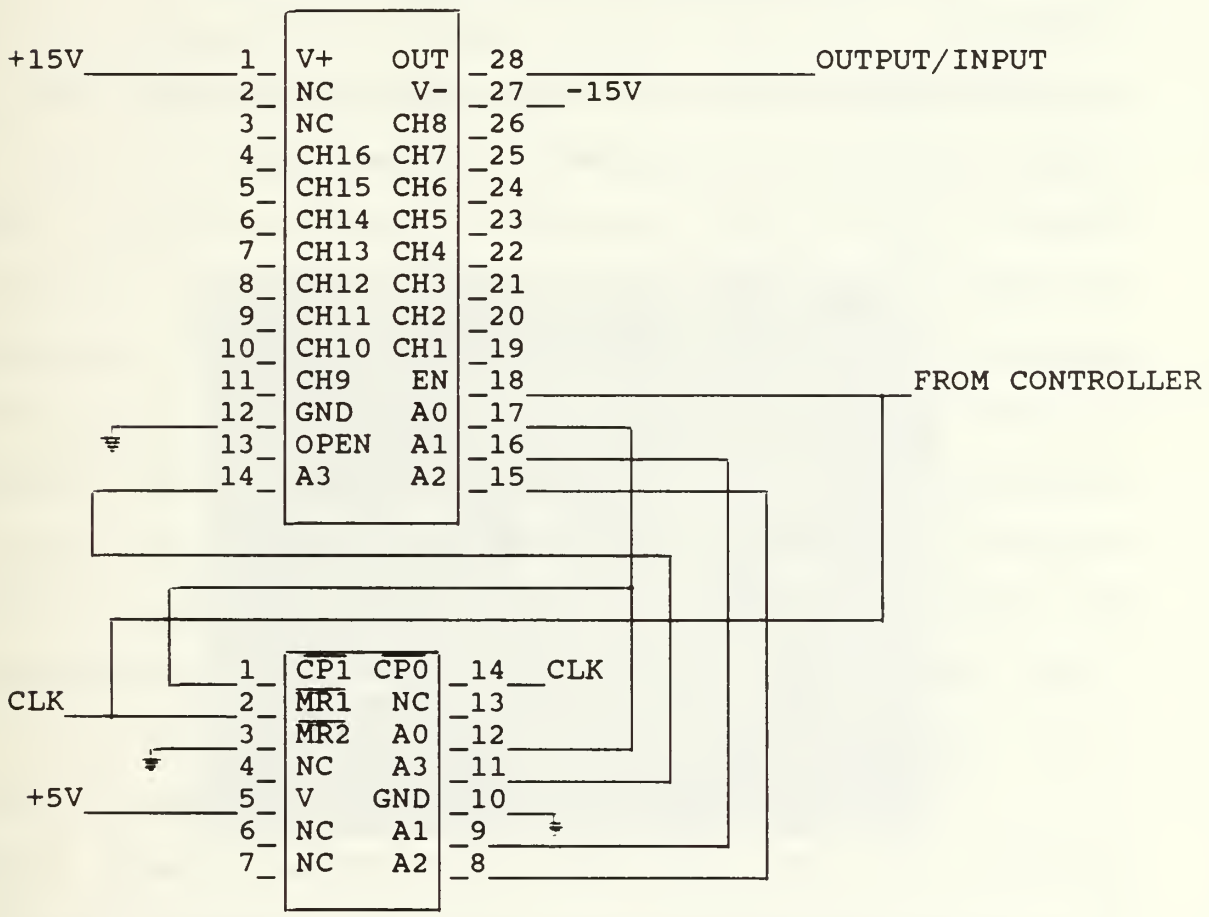


Figure 2.35 Channel Selection Circuit

III. TEST AND EVALUATION

A. TEST AND MEASUREMENTS

Due to the limited time at the end of this investigation, measurements were taken only for one channel. The test signals were the sinusoid, the triangular and the square waveforms. Measurements started from 1 KHz up to 100 KHz frequency settings. The system functioned very well up to 11 KHz with a slight distortion. Improvement in the shape of the output waveform was achieved when the first controller, using the NSC800 CPU, was replaced by a similar to the second one which had the Z8400 as CPU. For frequency settings above 11 KHz, the output waveform was distorted with an exception at 48 KHz where it seems to be a bandwidth of 2 KHz in which the output waveform can be seen clearly, although distorted. For test purposes the RAMs were disconnected from both the controller circuits in order to see how they were affecting the system's performance. The system functioned well up to 4 KHz frequency settings. Pictures were taken for the 9 KHz, 11 KHz and 48 KHz frequencies. Figure 3.1a illustrates the 9 KHz sinusoid input signal on the top and the filtered output underneath. Apart from the amplitude amplification, introduced by the filter, a slight clipping can be seen at the bottom of the waveform. Also when increasing the frequency, a delay is been introduced which for the 9 KHz causes the output waveform to be shifted by 170 degrees. Figure 3.1b shows the unfiltered

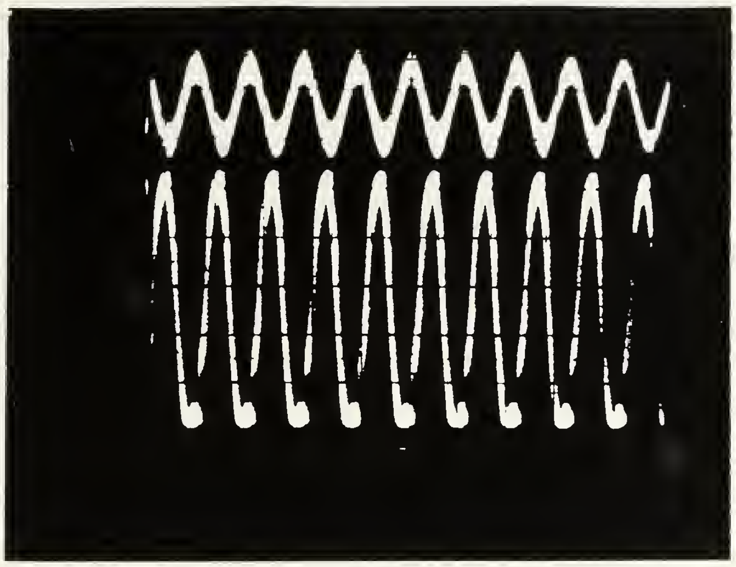


Figure 3.1a Input and Filtered Output Waveforms for 9 KHz Sinusoid Signal

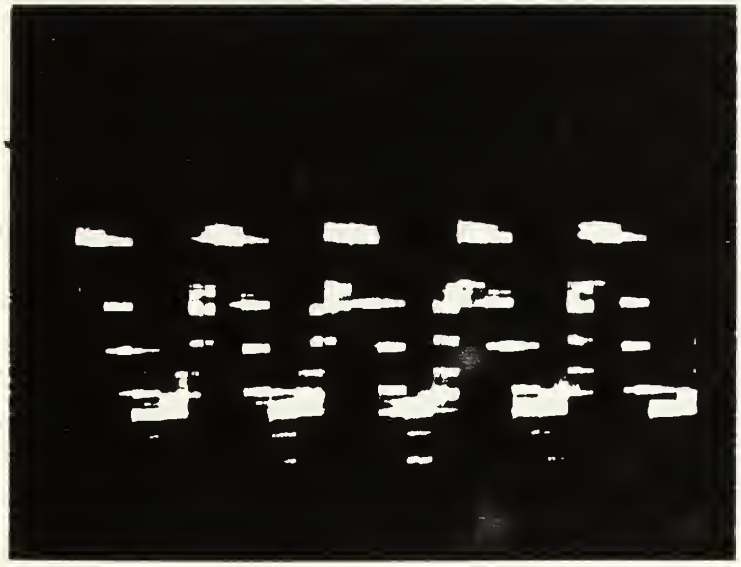


Figure 3.1b Unfiltered Output Waveform for 9 KHz Sinusoid Signal

output for the same input. Figure 3.2a shows the input on the top and the filtered output at the bottom, for the 9 KHz triangular signal. The system’s response is good for this kind of signal. There is the same amount of delay also here. Figure 3.2b shows the unfiltered output for the same input. Figure 3.3a shows the input and the filtered output at the bottom, for the 9 KHz square signal. The beginning of noise components can be seen clearly for this signal. The lower part of the output waveform is pretty much distorted probably because of the filter, because in the next figure, Figure 3.3b, the unfiltered output for the same input is very good. Figure 3.4a shows the input on the top and the filtered output at the bottom for the 11 KHz sinusoid signal. The output is almost inverted due to the delay. In Figure 3.4b the unfiltered output can be seen. Distortion slightly affects the shape of the waveform. Figure 3.5a shows the input and the filtered output at the bottom, for the 11 KHz triangular signal. Distortion is getting stronger, affecting the output’s shape. Figure 3.5b shows the unfiltered output for the same signal. Finally Figure 3.6a shows the severely affected filtered output at the bottom, for the 11 KHz square signal. Figure 3.5b shows the unfiltered output for the same signal. Six more figures, Figure 3.7a, 3.7b, 3.8a, 3.8b, 3.9a and 3.9b, show the filtered and unfiltered outputs for the three test signals at 48 KHz. The relatively good shape of the sinusoid and the badly distorted shape of the square signal are noted here.

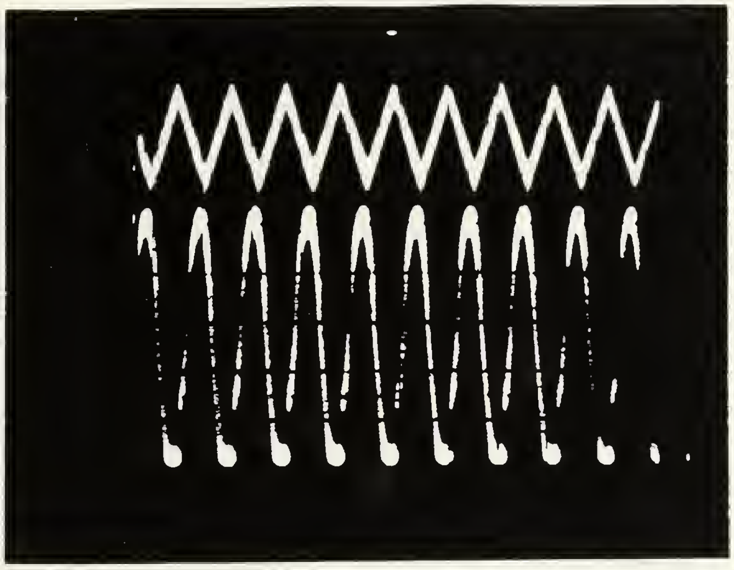


Figure 3.2a Input and Filtered Output Waveforms for 9 KHz Triangular Signal

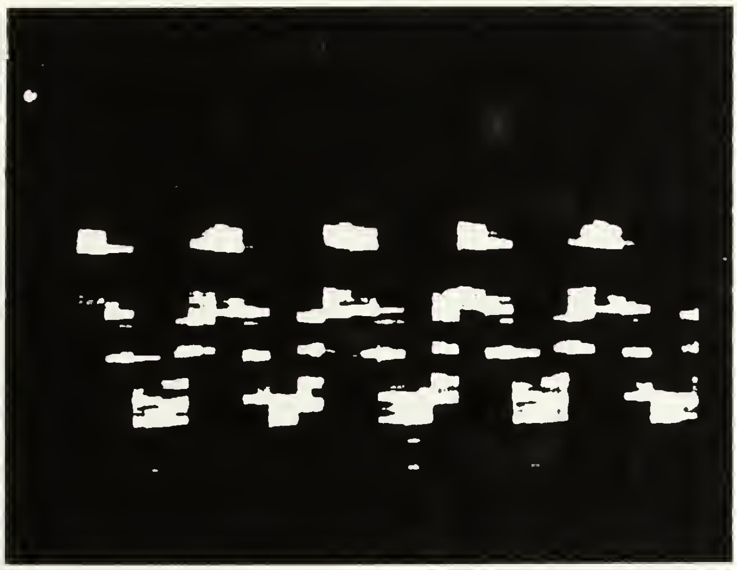


Figure 3.2b Unfiltered Output Waveform for 9 KHz Triangular Signal

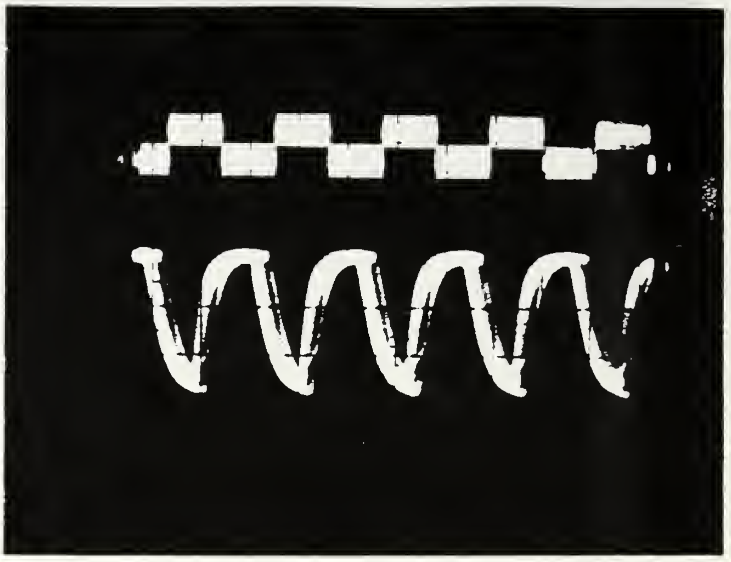


Figure 3.3a Input and Filtered Output Waveforms for 9 KHz Square Signal

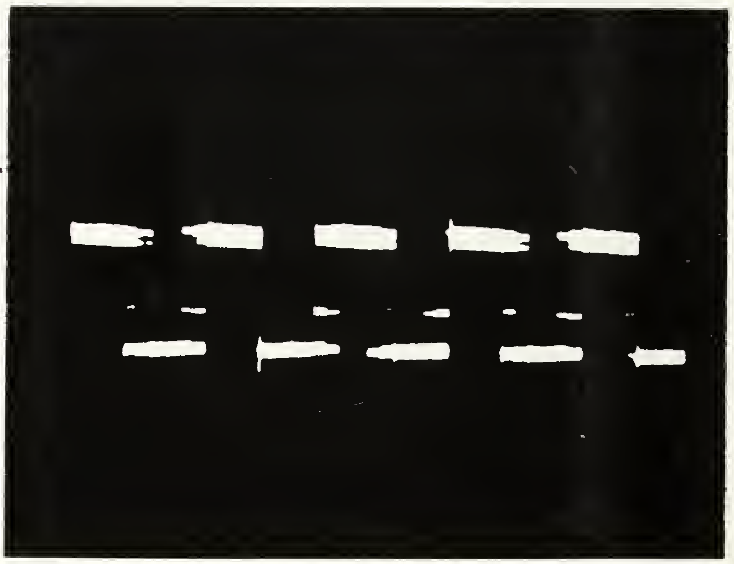


Figure 3.3b Unfiltered Output Waveform for 9 KHz Square Signal

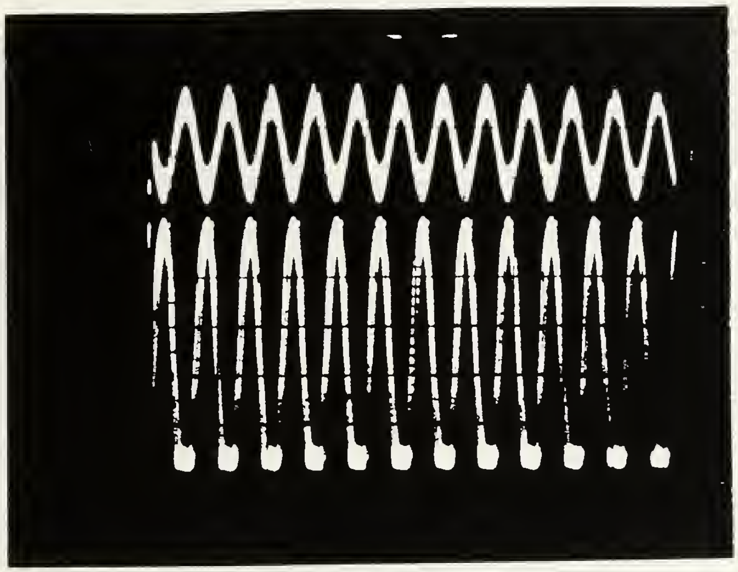


Figure 3.4a Input and Filtered Output Waveforms for 11 KHz Sinusoid Signal

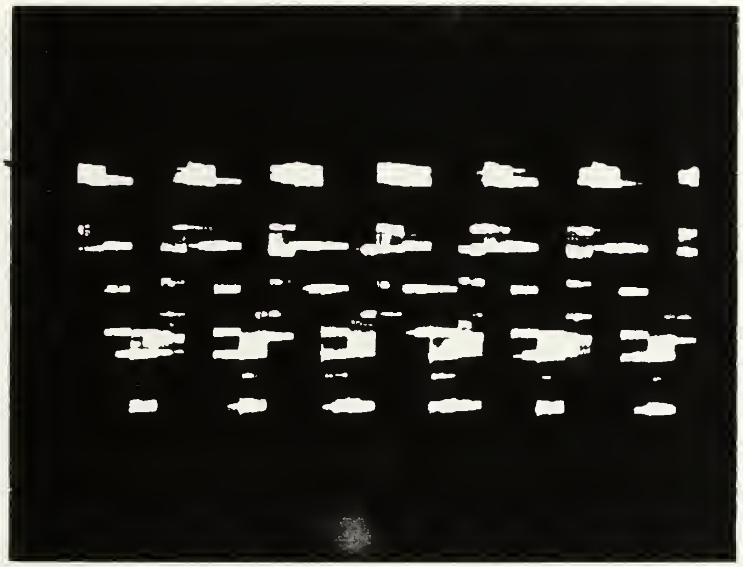


Figure 3.4b Unfiltered Output Waveform for 11 KHz Sinusoid Signal

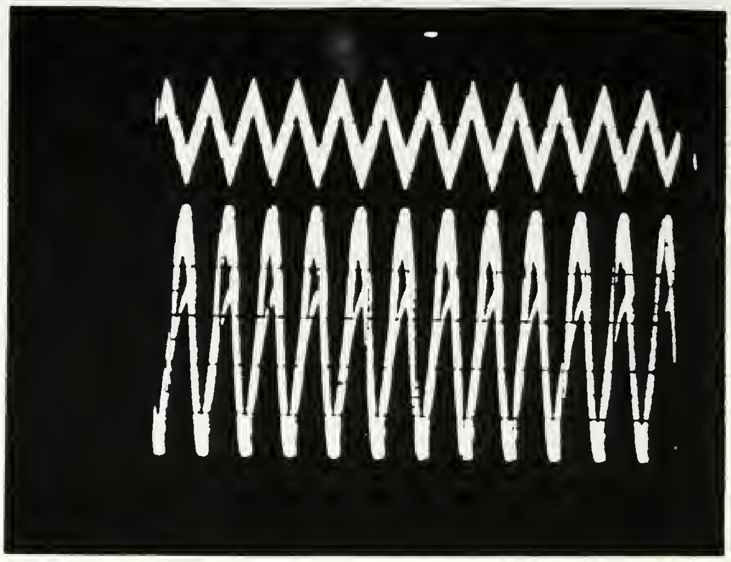


Figure 3.5a Input and Filtered Output Waveforms for 11 KHz Triangular Signal

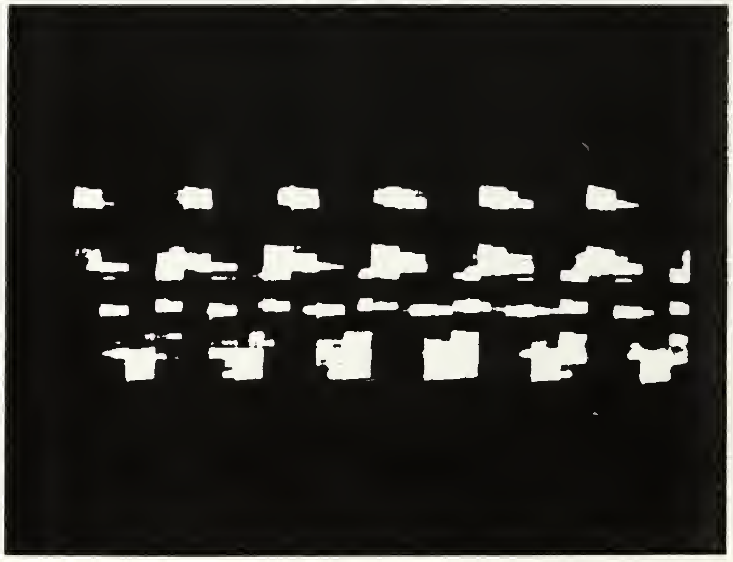


Figure 3.5b Unfiltered Output Waveform for 11 KHz Triangular Signal

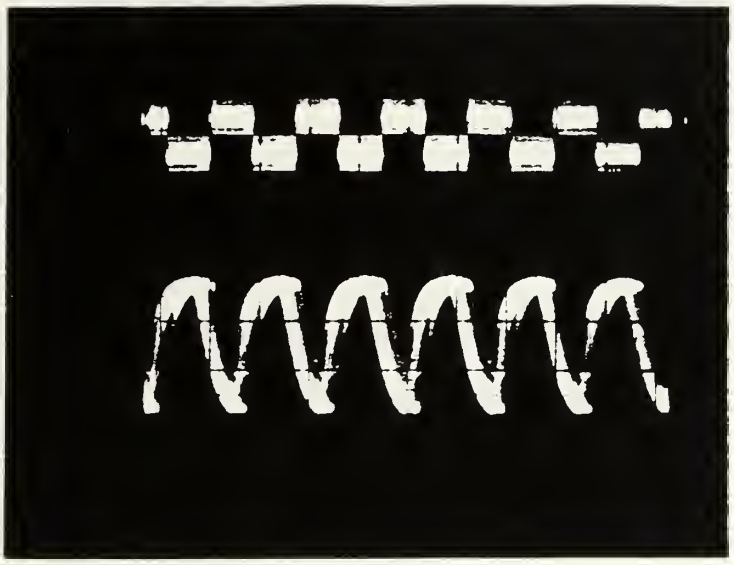


Figure 3.6a Input and Filtered Output Waveforms for 11 KHz Square Signal

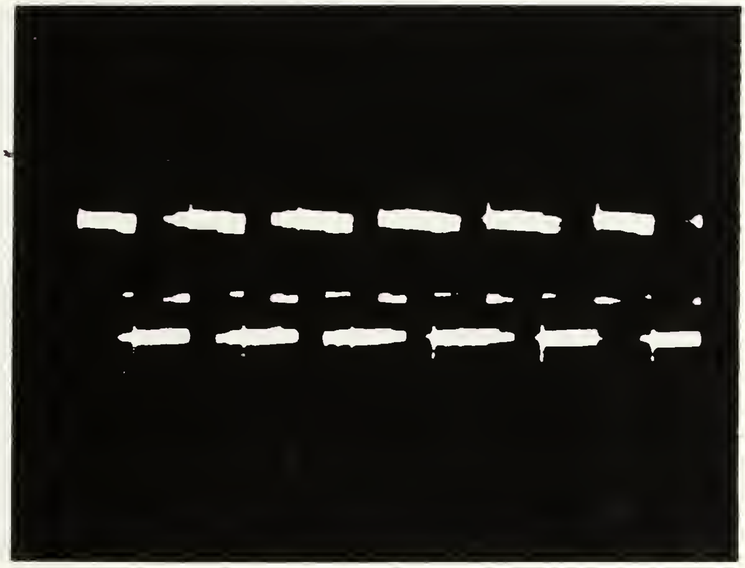


Figure 3.6b Unfiltered Output Waveform for 11 KHz Square Signal

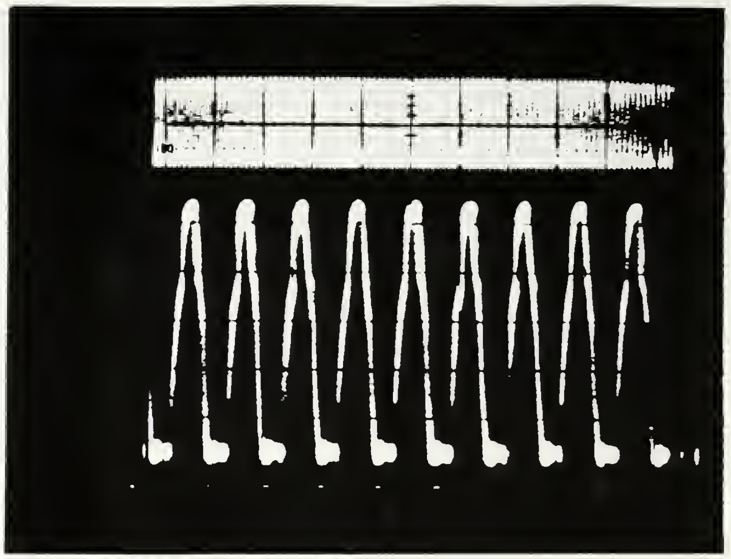


Figure 3.7a   Input and Filtered Output Waveforms for 48 KHz Sinusoid Signal

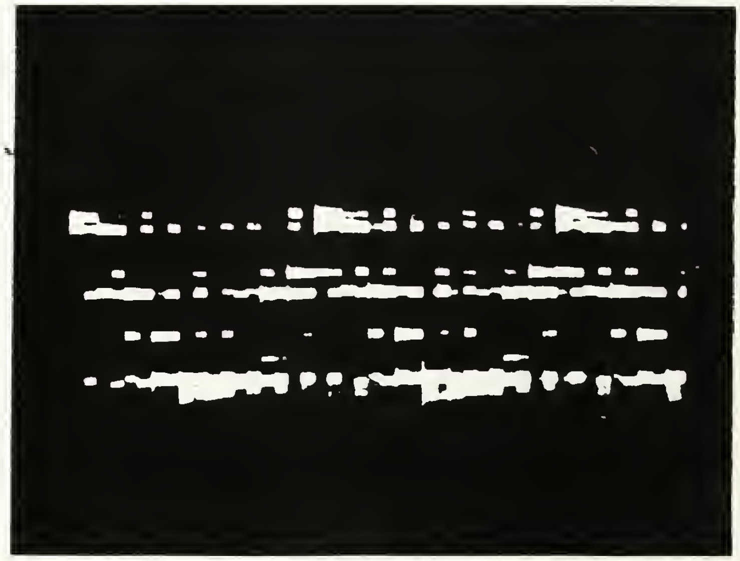


Figure 3.7b   Unfiltered Output Waveform for 48 KHz Sinusoid Signal

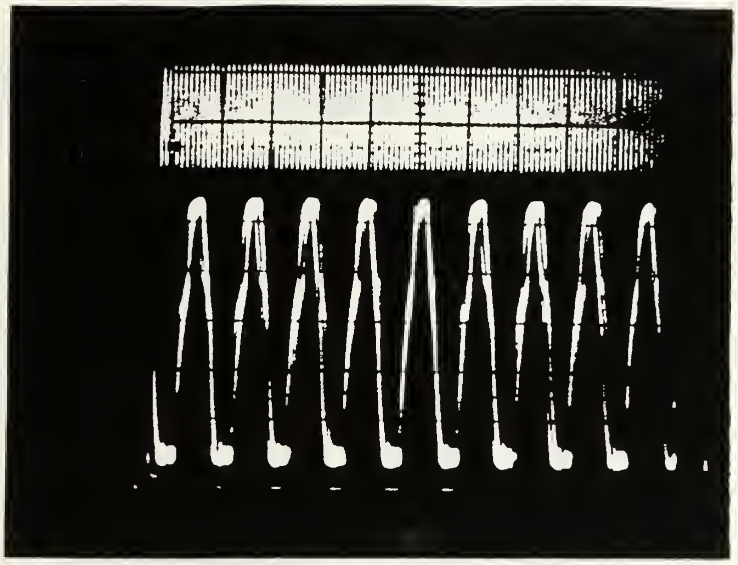


Figure 3.8a Input and Filtered Output Waveforms for 48 KHz Triangular Signal

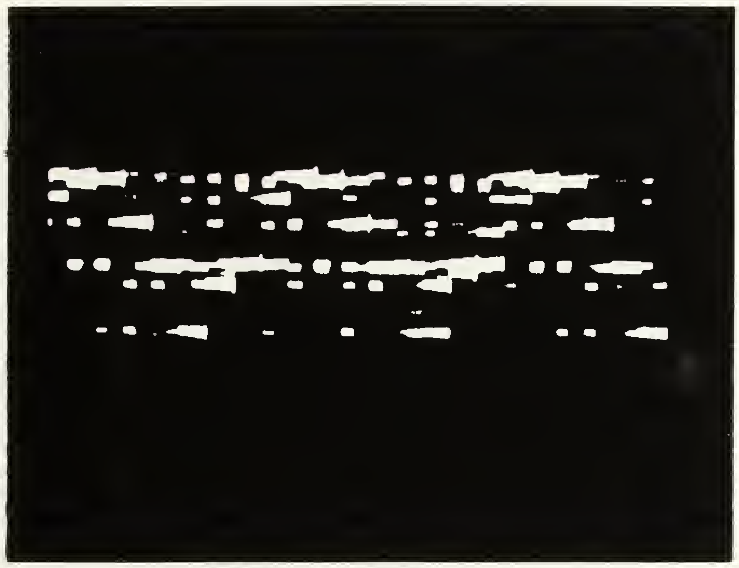


Figure 3.8b Unfiltered Output Waveform for 48 KHz Triangular Signal

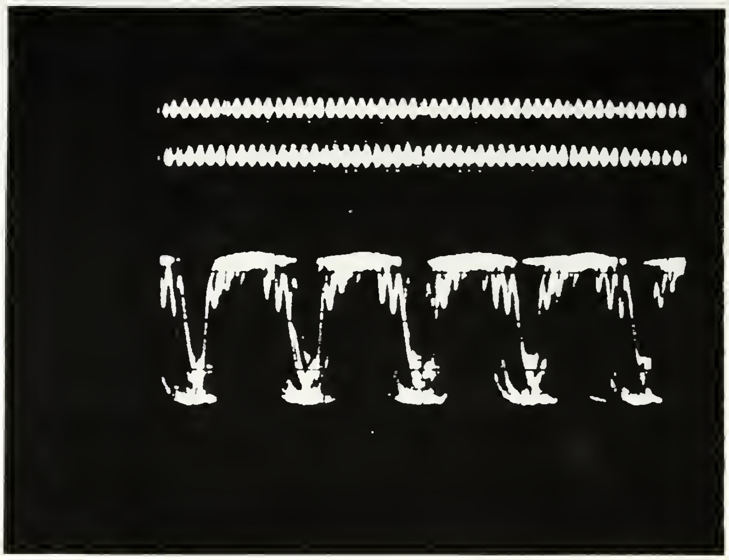


Figure 3.9a Input and Filtered Output Waveforms for 48 KHz Square Signal

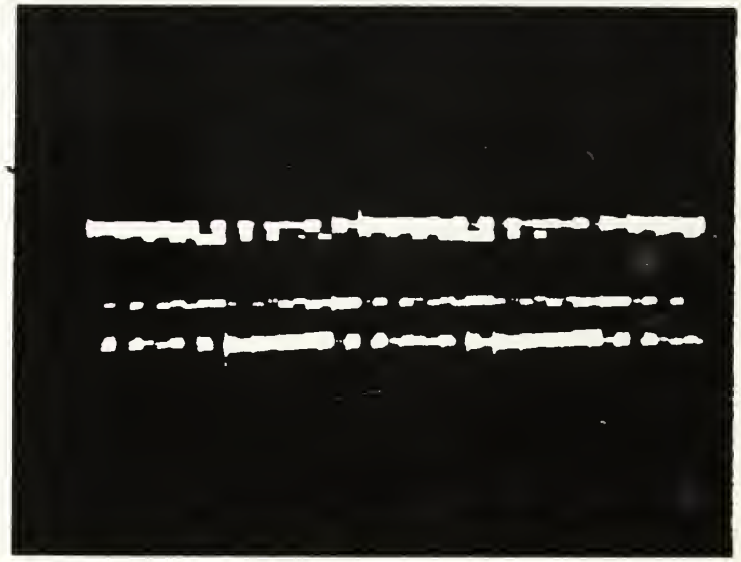


Figure 3.9b Unfiltered Output Waveform for 48 KHz Square Signal

B. CONCLUSIONS AND RECOMMENDATIONS

The goal, that is to design and build a system which transmits and receives data from different channels through a fiber optic link, was achieved for the most part. Even though the basic work was done with one channel the results can be applied to more than one channels. There is no need for more hardware, only a modification in the program which is stored in the EPROM. The multiplexer and the demultiplexer will be enabled from the controllers and in the same time each channel will be assigned a time window. Thus input and output synchronization will be achieved. The Manchester scheme will ensure always, that there will no delays introduced by the fiber optic cable. By using the Parallel Standard mode, to introduce the data to the input shift register for the Encoder through the controller, slower data transfer was performed and thus only up to 11 KHz was achieved. This was an expected trade-off. The presence of the RAM was a positive feature for the system's performance by increasing the bandwidth by 7 KHz. The system performed well for the sinusoid test signal but there was a distortion for the square test signal. The filter was not the best design but it served its purpose. Also, during the various frequency settings there was an increasing phase shift. There are faster processing multiplexer and demultiplexer chips on the market than the one used, which can be employed in a feature design. The system was low cost and had low power consumption.

The frequency can be increased by using the Parallel Bypass mode in which the data are transferred directly to the input shift register for the Encoder. Faster processing chips should be employed to minimize the processing delay. An idea to implement the design specifications would be to connect the multiplexer directly with the source. A latch would select the channels and data, in analog form, would be transmitted through the fiber optic cable. This would introduce a delay depending on the distance. The hardest part is to find this delay and use it to design a delay device just before the data go to the demultiplexer. Finally there are ready-made components on the market, though expensive, with which a more thorough investigation can be made. In essence, this thesis is the beginning point of a possible inexpensive fiber optic data link.

A. FUNCTIONAL DESCRIPTION OF THE NSC800 MICROPROCESSOR

This section was taken from Reference 3 and shows the NSC800 CPU register architecture as can be seen in Figure A1. The NSC800 is an 8-bit parallel device. The major functional areas are: the Arithmetic Logic Unit (ALU), register array, interrupt control, timing and control logic. These areas are connected via the 8-bit internal data bus.

1. Register Array

The NSC800 register array is divided into two parts: the dedicated registers and the working registers, as shown in Table A1.

TABLE A1 NSC800 REGISTER ARRAY [Ref. 3]

Main Register Set		Alternate Register Set		
Accumulator	Flags	Accumulator	Flags	
A	F	A′	F′	
B	C	B′	C′	Working
D	E	D′	E′	Registers
H	L	H′	L′	
Interrupt Vector I		Memory Refresh R		
Index Register IX				
Stack Pointer SP				
Program Counter PC				

2. Dedicated Registers

There are 6 dedicated registers in the NSC800: two 8-bit and four 16-bit registers, see Table A2.

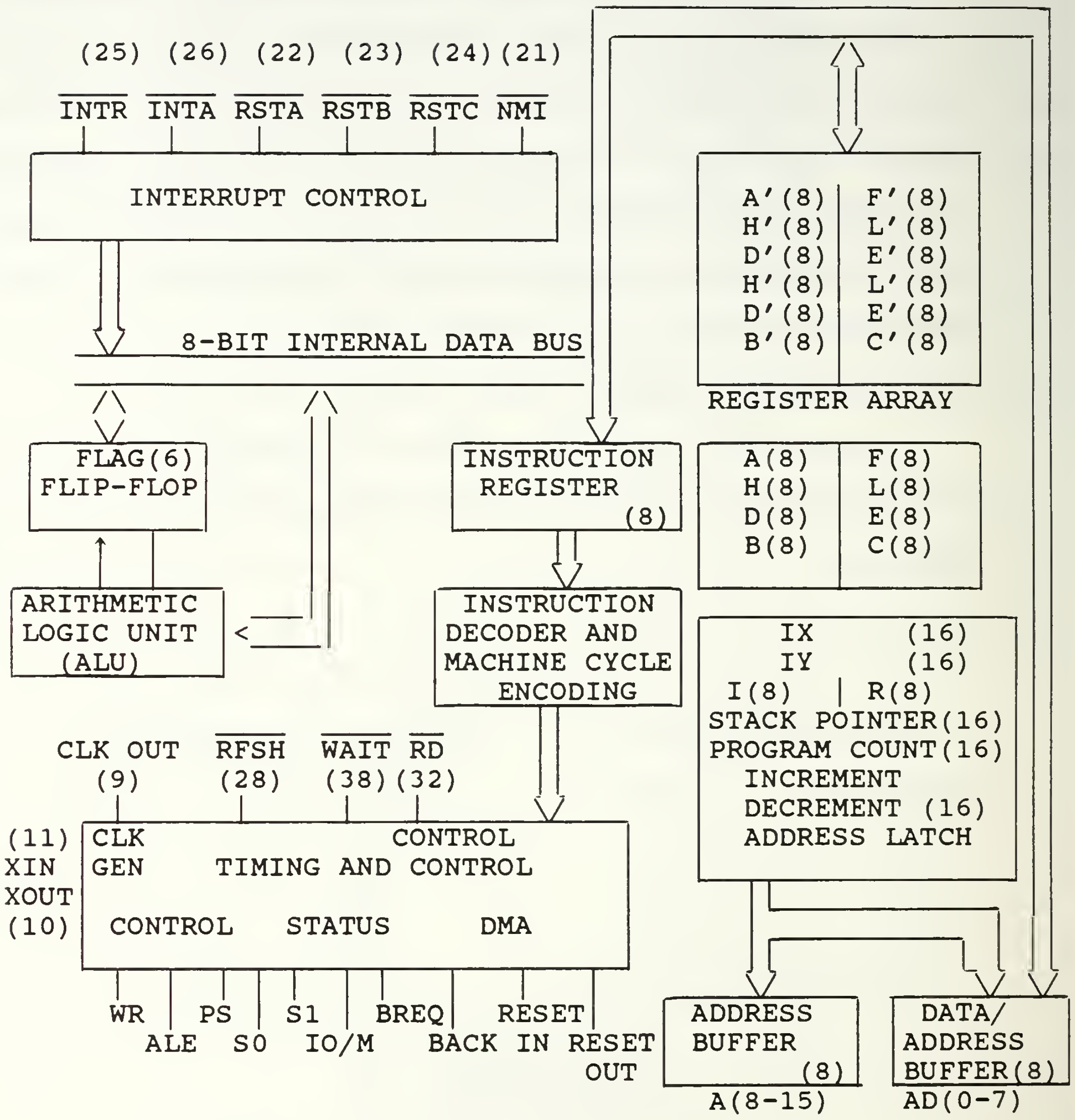


Figure A1 NSC800 CPU Functional Block Diagram [Ref. 3]

TABLE A2 DEDICATED REGISTERS [Ref. 3]

CPU Dedicated Registers

Program Counter PC	(16)
Stack Pointer SP	(16)
Index Register IX	(16)
Index Register IY	(16)
Interrupt Vector Register I	(8)
Memory Refresh Register R	(8)

a. Program Counter (PC)

The program counter contains the 16-bit address of the current instruction being fetched from memory. The PC increments after its contents have been transferred to the address lines. When a program jump occurs, the PC receives the new address which overrides the incrementer. There are many conditional and unconditional jumps, calls and return instructions in the NSC800's instruction repertoire that allow easy manipulation of this register in controlling the program execution.

b. Stack Pointer (SP)

The 16-bit stack pointer contains the address of the current top of stack that is located in external system RAM. The stack is organized in a last-in, first-out (LIFO) structure. The pointer decrements when data is pushed onto the stack and increments when data is popped from the stack. Various operations store or retrieve data on the stack. This along with the usage of subroutine calls and interrupts allows simple implementation of subroutine and interrupt nesting as well as alleviating many of data manipulation.

c. Index Register (IX and IY)

The NSC800 contains two index registers to hold independent 16-bit address used in the indexed addressing mode. In this mode an index register, either IX or IY, contains a base address of an area in memory making it a pointer for data tables. In all instructions employing indexed modes of operation, another byte acts as a signed two's complement displacement. This addressing mode enables easy data table manipulations.

d. Interrupt Register (I)

When the NSC800 provides a Mode 2 response to $\overline{\text{INTR}}$, the action taken is an indirect call to the memory location containing the service routine address. The pointer to the address of the service routine is formed by two bytes, the high-byte is from the I Register and the low-byte is from the interrupting peripheral. The peripheral always provides an even address for the lower byte (LSB=0). When the processor receives the wide lower byte from the peripheral it concatenates it in the following manner:

I Register	External byte
8-bits	0

The even location contains the low-order byte, the next consecutive location contains the high-order byte of the pointer to the beginning address of the interrupt service routine.

e. Refresh Register (R)

For systems that use dynamic memories rather than static RAM's, the NSC-800 provides an integral 8-bit memory refresh counter. The contents of the register are incremented after each opcode fetch and are sent out on the lower portion of the address bus, along with a refresh control signal. This provides a totally transparent refresh cycle and does not slow down CPU operation. The program can read and write to the R register, although this is usually done only for test purposes.

3. <u>CPU Working and Alternate Register Sets</u>

a. CPU Working Registers

The portion of the register array shown in Table A3, represents the CPU working registers.

TABLE A3 CPU MAIN WORKING REGISTER SET [Ref. 3]

Accumulator	A	(8)	Flags F	(8)
Register	B	(8)	Register C	(8)
Register	D	(8)	Register E	(8)
Register	H	(8)	Register L	(8)

These sixteen 8-bit registers are general purpose registers because they perform a multiple of functions, depending on the instruction being executed. They are grouped together also because of the manipulative types of instruction that they perform, particularly alternate set operations.

b. Alternate Registers

The NSC800 registers designated as CPU working registers have one common feature: the existence of a duplicate register in an alternate register set. This duplicate register in an alternate register set. This architectural concept simplifies programming during operations such as interrupt response, when the machine status represented by the contents of the registers must be saved. The alternate register concept makes one set of registers available to the programmer at any given time. Two instructions (EX AF, A'F' and EXX), exchange the current working set of registers with their alternate set. One exchange between the A and F registers and their respective duplicates (A'and F') saves the primary status information contained in the accumulator and the flag register. The second exchange instruction performs the exchange between the remaining registers B, C, D, E, H, and L, and their respective alternates B', C', D', E', H', and L'. This essentially saves the contents of the original complement of registers while providing the programmer with a usable alternate set. The alternate registers are given in Table A4.

TABLE A4 CPU ALTERNATE WORKING REGISTERS [Ref. 3]

Accumulator	A'	(8)	Flags	F'	(8)
Register	B'	(8)	Register	C'	(8)
Register	D'	(8)	Register	E'	(8)
Register	H'	(8)	Register	L'	(8)

4. <u>Register Functions</u>

a. Accumulator (A Register)

The A register serves as a source or destination register for data manipulation instructions. In addition, it serves as the accumulator for the results of 8-bit arithmetic and logic operations. The A register also has a special status in some types of operations; that is, certain addressing modes are reserved for the A register only, although the function is available for all the other registers. For example. any register can be loaded by immediate, register indirect, or indexed source addressing modes. The A register, however, can be loaded with an additional register indirect capability or by extended direct source addressing. Another special feature of the A register is that it produces more efficient memory coding than equivalent instruction functions directed to other registers. Any register can be rotated; however, while it requires a two-byte instruction to normally rotate any register, a single-byte instruction is available for rotating the contents of the accumulator.

b. F Register-Flags

The NSC800 flag register consists of six status bits that contain information regarding the results of previous CPU operations. The register can be read by pushing the contents onto the stack and then reading them, however, it cannot be written to. It is classified as a register because of its affiliation with the accumulator and the existence of a duplicate register for use in exchange instructions with the accumulator. Of the six flags shown in Table A5, only four can be directly tested by the programmer via conditional jump, call and return instructions. They are the Sign (S), Zero (Z), Parity/Overflow (P/V), and Carry (C) flags. The Half Carry (H) and Add/Subtract (N) flags are used for internal operations related to BCD arithmetic.

TABLE A5 FLAG REGISTER [Ref. 3]

```
BIT 7                                              BIT 0

  S        Z      \    H     \      P/V      N          C
  |        |           |             |       |          |
SIGN     ZERO     HALF  CARRY  PARITY     ADD/       CARRY
                               OVERFLOW   SUBTRACT
```

c. Carry (C)

A carry from the highest order bit of the accumulator during an add instruction, or a borrow generated during a subtraction instruction sets the carry flag. Specific shift and rotate instructions also affect this bit. Two specific instructions in the NSC800 instruction repertoire set (SCF) or complement (CCF) the carry flag. Other operations that affect the C flag are as follows:

- Adds
- Subtracts
- Logic Operations (always resets C flag)
- Rotate Accumulator
- Decimal Adjust
- Negation of Accumulator

Other operations do not affect the C flag.

d. Adds/Subtract (N)

This flag is used in conjunction with the H flag to ensure that the proper BCD correction algorithm is used during the decimal adjust instruction (DAA). The correction algorithm depends on whether an add or subtract was previously done with BCD operants. The operations that set the N flag are:

- Subtractions
- Decrements (8-bit)
- Complementing of the Accumulator
- Block I/O
- Block Searches
- Negation of the Accumulator.

The operations that reset the N flag are:

- Adds
- Increments
- Logic Operations
- Rotates
- Set and Complement Carry
- Input Register Indirect
- Block Transfers
- Load of the I or R Registers
- Bit Tests

Other operations do not affect the N flag.

e. Parity/Overflow (P/V)

The Parity/Overflow flag is a dual purpose flag that indicates results of logic and arithmetic operations. In logic operations, the P/V flag indicates the parity of the result; the flag is set (high) if the result is even, reset (low) if the result is odd. In arithmetic operations, it represents an overflow condition when the result, interpreted as signed two's complement arithmetic, is out of range for the eight-bit accumulator. The following operations affect the P/V flag according to the parity of the result of the operation:

- Logic Operations
- Rotate Digits
- Decimal Adjust
- Rotate and Shift
- Input Register Indirect

The following operations affect the P/V flag according to the overflow result of the operation.

- Adds (16 bit with carry, 8-bit with/without carry)
- Subtracts (16 bit with carry, 8-bit with/without carry)
- Increments and Decrements
- Negation of Accumulator

The P/V flag has no significance immediately after the following operations.

- Block I/O
- Bit Tests

In block transfers and compares, the P/V flag indicates the status of the BC register, always ending in the reset state after an auto repeat of a block move. Other operations do not affect the P/V flag.

This flag indicates a BCD carry, or borrow, result from the low-order four bits of operation. It can be used to correct the results of a previously packed decimal add, or subtract, operation by use of the Decimal Adjust Instruction (DAA). The following operations affect the H flag:

- Adds (8-bit)
- Subtracts (8-bit)
- Increments and Decrements
- Decimal Adjust
- Negation of Accumulator
- Always Set by: Logic AND
  Complement Accumulator
  Bit Testing
- Always Reset by: Logic OR's and XOR's
  Rotates and Shifts
  Set Carry
  Input Register Indirect
  Block Transfers
  Loads of I and R Registers

The H flag has no significance immediately after the following operations:

- 16-bit Adds with/without carry
- 16-bit Subtracts with carry
- Complement of the carry
- Block I/O
- Block Searches

Other operations do not affect the H flag.

g. Zero Flag (Z)

Loading a zero in the accumulator or when a zero results from an operation sets the zero flag. The following operations affect the zero flag:

- Adds (16-bit with carry, 8-bit with/without carry)
- Subtracts (16-bit with carry, 8-bit with/without carry)
- Logic Operations
- Increments and Decrements
- Rotate and Shifts
- Rotate Digits
- Decimal Adjust
- Input Register Indirect
- Block I/O (always set after auto repeat block I/O)
- Block Searches
- Load of I and R Registers
- Bit Tests
- Negation Transfers

The Z flag has no significance immediately after the following operations:

- Block Transfers

Other operations do not affect the zero flag.

h. Sign Flag (S)

The sign flag stores the state of the bit 7 (the most-significant bit and sign bit) of the accumulator following an arithmetic operation. This flag is of use when following an arithmetic operation. This flag is of use when dealing with signed numbers. The sign flag is affected by the following operation according to the result:

- Adds (16-bit with carry, 8-bit with/without carry)
- Subtracts (16-bit with carry, 8-bit with/without carry)
- Logic Operations
- Increments and Decrements
- Rotate and Shifts
- Rotate Digits
- Decimal Adjust
- Input Register Indirect
- Block Searches
- Load of I and R Registers
- Negation of Accumulator

The S flag has no significance immediately after the following operations:

- Block I/O
- Block Transfers
- Bit Tests

Other operations do not affect sign bit.

i. Additional General-Purpose Registers

The other general-purpose registers are the B, C, D, E, H and L registers and their alternate register set B′, C′, D′, E′, H′and L′. The general-purpose registers can be used interchangeably. In addition, the Band C registers perform special functions in the NSC800 expanded I/O capabilities, particularly block I/O operations. In these functions, the C register can address I/O ports; the B register provides a counter function when used in the register indirect address mode. When used with the special condition jump instruction (DJNZ) the B register again provides the counter function.

j. Alternate Configurations

The six 8-bit general-purpose registers (B, C, D, E, H, L) will combine to form three 16-bit registers. This occurs by concatenating the B and C registers to form the BC register, the D and E registers form the DE register, and the H and L registers form the HL register. Having these 16-bit registers allows 16-bit data handling, thereby expanding the number of 16-bit registers available for memory addressing modes. The HL register typically provides the pointer address for use in register indirect addressing of the memory. The DE register provides a second memory pointer register for the NSC800's block transfer operations. The BC register also provides an assist to the block transfer operations by acting as a byte-counter for these operations.

5. Arithmetic Logic Unit (ALU)

The arithmetic, logic and rotate instructions are performed by the ALU. The ALU internally communicates with the registers and data buffer on the 8-bit internal data bus.

6. Instruction Register and Decoder

During an opcode fetch, the first byte of an instruction is transferred from the data buffer to the instruction register. The instruction register feeds the instruction decoder, which gated by timing signals, generates the control signals that read or write data from or to the registers, control the ALU and provide all required external control signals. For more details about the NSC800's functions see Reference 3, from which most of the information came.

B. Z8400 REGISTER GENERAL DESCRIPTION

The Z8400 CPU offer higher system throughput and more efficient memory utilization than the previous NSC800. The internal registers contain 208 bits of read/write memory that are accessible. These registers include two sets of six general-purpose registers which may be used individually as either 8-bit registers or as 16-bit register pairs. In addition, there are two sets of accumulator and flag registers. A group of exchange instructions makes either set of main or alternate registers accessible. The alternate set allows operation in foreground-background mode or it may be reserved for very fast interrupt response. The Z8400 also contains a Stack Pointer, Program Counter, two index registers, a Refresh register (counter), and an Interrupt register. All output signals are fully decoded and timed to control standard memory or peripheral circuits. The internal block diagram, Figure A2, shows the primary functions of the Z8400 processor.

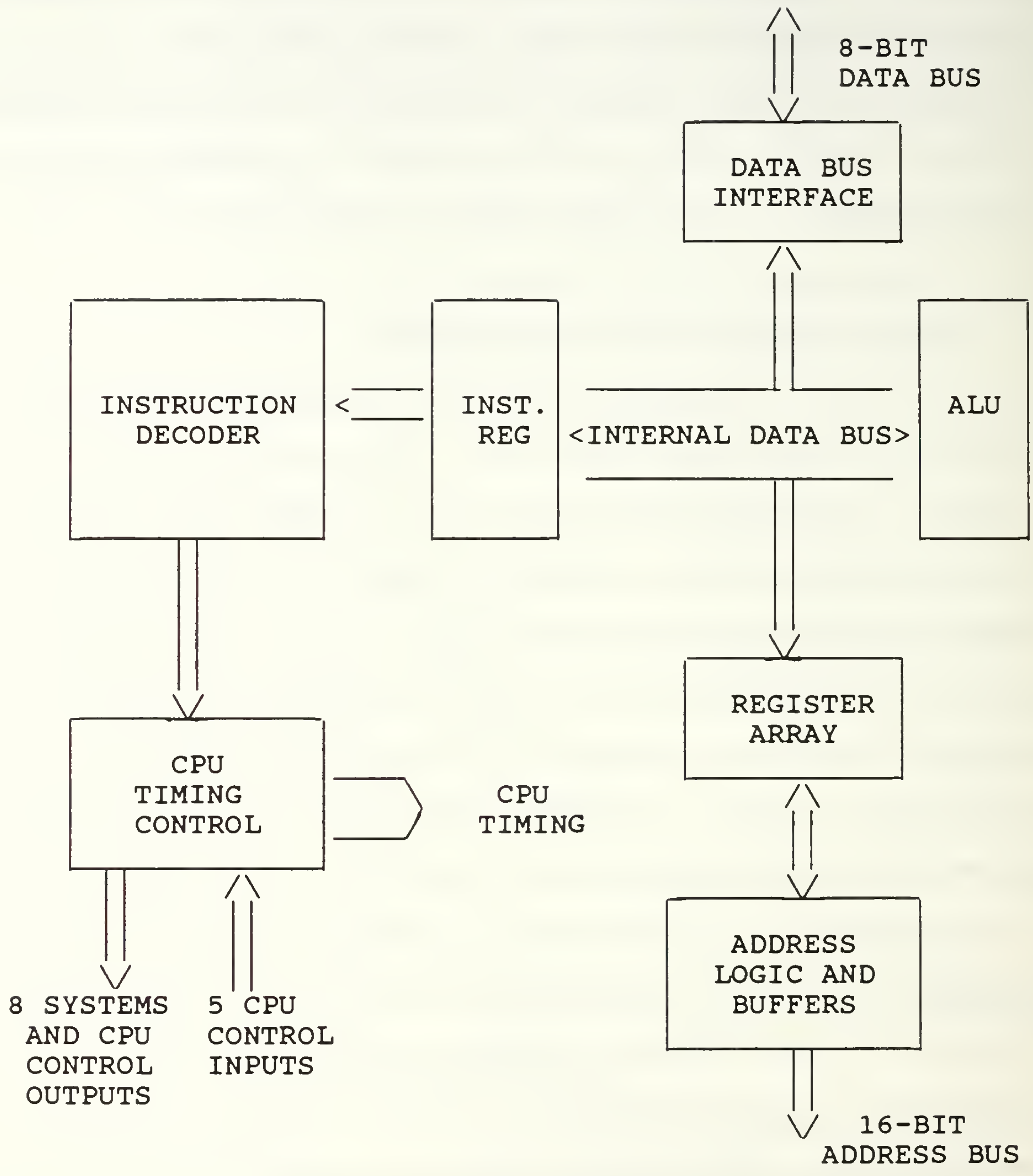


Figure A2 Z8400 CPU Block Diagram [Ref. 4]

Table A6 shows three groups of registers within the Z8400 CPU. The first group consists of duplicate sets of 8-bit registers: a principal set and an alternate set. Both sets consist of the Accumulator Register, the Flag Register and six general-purpose registers. Transfer of data between these duplicate sets of registers is accomplished by use of exchange instructions. The result is faster response to interrupts and easy, efficient implementation of such versatile programming techniques as background-foreground data processing. The second set of registers consists of six registers with assigned functions. These are the I (Interrupt Register), the R (Refresh Register), the IX and IY (Index Registers), the SP (Stack Pointer), and the PC (Program Counter). The third group consists of two interrupt status flip-flops, plus an additional pair of flip-flops which assists in identifying the interrupt mode at any particular time. Table A7 provides further information on these registers. The CPU accepts two interrupt signals: $\overline{\text{NMI}}$ and $\overline{\text{INT}}$. The $\overline{\text{NMI}}$ is a non-maskable interrupt and has the highest priority. $\overline{\text{INT}}$ is a lower priority interrupt and it requires that interrupts be enabled in software in order to operate. $\overline{\text{INT}}$ can be connected to multiple peripheral devices in a wired-OR configuration. The Z8400 has a single response mode for interrupt service for the non-maskable interrupt. The maskable interrupt, $\overline{\text{INT}}$, has three programmable response modes available. The CPU services interrupts by sampling the $\overline{\text{NMI}}$ and $\overline{\text{INT}}$ signals at the rising edge of the last clock of an instruction. More details about the Z8400 CPU are available in Reference 4 from which most of the above information came.

TABLE A6 Z8400 CPU REGISTERS [Ref. 4]

MAIN REGISTER SET		ALTERNATE REGISTER SET	
A ACCUMULATOR	F FLAG REG.	A′ ACCUMULATOR	F′ FLAG REGISTER
B GEN. PURPOSE	C GEN. PURPOSE	B′ GEN. PURPOSE	C′ GEN. PURPOSE
D GEN. PURPOSE	E GEN. PURPOSE	D′ GEN. PURPOSE	E′ GEN. PURPOSE
H GEN. PURPOSE	L GEN. PURPOSE	H′ GEN. PURPOSE	L′ GEN. PURPOSE
<---8 BITS---->			

IX INDEX REGISTER
IY INDEX REGISTER
SP STACK POINTER
PC PROGRAM COUNTER
<-----16 BITS---->

I INTERRUPT VECTOR
R MEMORY REFRESH
<-----8 BITS---->

INTERRUPT FLIP-FLOPS STATUS

IFF1	IFF2
0= INT. DISABLED 1= INT. ENABLED	STORES IFF1 DURING NMI SERVICE

INTERRUPT MODE FLIP-FLOPS

IMFa	IMFb	
0	0	INTERRUPT MODE
0	1	NOT USED
1	0	INTERRUPT MODE 1
1	1	INTERRUPT MODE 2

TABLE A7 Z8400 CPU REGISTERS [Ref. 4]

	Register	Bits	Remarks
A,A'	Accumulator	8	Stores an operand or the results of an operation
F,F'	Flags	8	[Ref. 4] Instruction Set
B,B'	Gen. Purpose	8	Can be used separately or as a 16 bit register with C
C,C'	Gen. Purpose	8	See register B above
D,D'	Gen. Purpose	8	Can be used separately or as a 16 bit register with E
E,E'	Gen. Purpose	8	See register D above
H,H'	Gen. Purpose	8	Can be used separately or as a 16 bit register with L
L,L'	Gen. Purpose	8	See register H above
I	Interrupt Reg.	8	Stores upper eight bits of memory address for vectored interrupt processing
R	Refresh Reg.	8	Provides user-transparent dynamic memory refresh. Automatically incremented and placed on the address bus during each instruction fetch cycle
IX	Index Reg.	16	Used for indexed addressing
IY	Index Reg.	16	Same as IX register above
SP	Stack Pointer	16	Holds address of the top of the stack
PC	Program Counter	16	Holds address of next instruction
IFF1 IFF2	Interrupt Enable	Flip Flops	Set or reset to indicate interrupt status
IMFa IMFb	Interrupt Mode	Flip Flops	Reflect Interrupt mode

APPENDIX B

SOFTWARE DEVELOPMENT

A simple program was written in the EPROM in order to initiate the reception of the data bits and to output the addresses. The chip selection was done through these addresses.

Program

Address	Opcode/Address	Mnemonic	Comment
0000	D3	OUT +	Output next byte
0001	51	ADDRS	Output address 51
0002	00	NOP	No operation
0003	00	NOP	No operation
0004	00	NOP	No operation
0005	00	NOP	No operation
0006	00	NOP	No operation
0007	00	NOP	No operation
0008	00	NOP	No operation
0009	3A	LDA ++	Load the next two bytes
000A	00	ADDRS	---
000B	40	ADDRS	Load address 4000
000C	C3	JMP ++	Jump to the next two bytes
000D	00	ADDRS	---
000E	00	ADDRS	Jump to address 0000

LIST OF REFERENCES

1. Hewlett-Packard Company, Optoelectronics Designer's Catalog, pages 4.20, 4.26-4.36, Palo Alto, California, 1986.

2. Harris Corporation, CMOS Digital Data Book, pages 4.47-4.54, Palm Bay, Florida, 1986.

3. NATIONAL SEMICONDUCTOR, NSC800 Microprocessor Family Databook, pages 1.3-1.23, Santa Clara, California, 1985.

4. Zilog, Data Book, pages 5-22, Cambell, California, 1982/1983.

5. Precision Monolithics Inc., Data Book and Applications Handbook, ADC-9008, Santa Clara, California, 1986.

6. MAXIM, Data Converters and Voltage References, pages 1.47-1.58, Sunnyvale, California, 1987.

7. National Semiconductor Corporation, Linear Databook, pages 8.118-8.125, Santa Clara, California, 1982.

8. LINEAR TECHNOLOGY, High Speed Comparator Techniques-Application Note 13, pages 14-16, Milpitas, California, 1985.

9. Sherif Michael, ECE 4100 Notes, page 28, 1987.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5000	2
2.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
3.	Department Chairman, Code 62 Department of Elecrical and Computer Engineering Naval Postgraduate School Monterey, California 93943	1
4.	Professor J.P. Powers, Code 62 Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943	2
5.	Professor Sherif Michael, Code 62 Mi Department of Elecrtical and Computer Engineering Naval Postgraduate School Monterey, California 93943	1
6.	Lieutenant Fokion Hatzidakis, Hellenic Navy Thiras 62 Athens 11252, Greece	6

Thesis
H3255 Hatzidakis
c.1 Multichannel data transmission through a fiber optic cable.