AUTOMATION OF ULTRASOUND DOSIMETRY EXPERIMENT

BY

STEVEN THEO VAUGHN
B.S., University of Illinois, 1977
B.S., Western Illinois University, 1977

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1980

Urbana, Illinois

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

THE GRADUATE COLLEGE

AUGUST 1980

WE HEREBY RECOMMEND THAT THE THESIS BY

STEVEN THEO VAUGHN

ENTITLED AUTOMATION OF ULTRASOUND DOSIMETRY EXPERIMENT

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF MASTER OF SCIENCE

William D. O'Brien, Jr
Director of Thesis Research

Head of Department

Committee on Final Examination†

Chairman

† Required for doctor's degree but not for master's.

O-517

UNIVERSITY OF ILLINOIS
at Urbana-Champaign
The Graduate College
330 Administration Building

FORMAT APPROVAL

To the Graduate College:

The format of the thesis submitted by STEVEN THEO VAUGHN for the degree of M.S. is acceptable to the department of ELEC ENG.

7/22/80
Date

(Signed) ____________________
Departmental Representative

ACKNOWLEDGEMENTS

The author wishes to express his grateful appreciateion to his advisor, Professor William D. O'Brien, for guidance and encouragement offered throughout this projects. He would also like to express a special thanks to the supporting staff of the Bioacoustics Research Laboratory. These thanks are extended to Mr. Joe Cobb for his advise in electronic matters, to Mr. Bill McNeill for building the mechanical systems in the experiment and to Mrs. Wanda Elliott for typing the thesis and providing welcomed advise on all administrative matters. The author also thanks Dr. Ronald Johnston and Mr. David Duback for contributing their expertise to the software development associated with this project. The author finally thanks his parents and family for their support throughout his educational career, and his future wife, Connie, for her support and encouragement during this difficult period.

Table of Contents

Chapter 1

Introduction

The purpose of this chapter is to provide insight into the dosimetry experiment and to outline the organization of the remaining chapters.

Studies have shown that increasing numbers of the population are being exposed to ultrasound every year (6). With this increasing use of ultrasound in medical diagnostic equipment it becomes attractive to consider an ultrasonic dosimetric parameter. This parameter would be useful for patient monitoring as well as being a useful tool for manufacturers and purchasers of ultrasonic equipment, possibly providing output information for comparing different devices.

An experiment has been proposed and is under way that may result in the definition of such a dosimetry parameter. This experiment uses the transient thermoelectric technique (TTT) to map the _in situ_ distribution of sound intensity in a mouse specimen. To accomplish this, a copper and constantin thermocouple junction is inserted into the specimen. The specimen is irradiated with ultrasound and the thermocouple response measured. The thermocouple response is related to the intensity by the equation:

$$I = \frac{\rho c}{\mu} (\frac{dT}{dt})_o$$

where I represents the intensity, μ is the acoustic intensity absorption coefficient of the embedding medium, the product ρc is the heat capacity of the embedding medium per unit volume, and $(\frac{dT}{dt})_o$ represents the time rate of change of the temperature of the embedding medium (3). In this manner, if the acoustic absorption coefficient (μ) and the heat capacity of the embedding medium are known, the slope of the thermocouple response yields $(\frac{dT}{dt})_o$ and the intensity can be found. After the collection of a large number of data points the specimen is histologically examined to identify the embedding material at each point. This technique provides information to develop an <u>in situ</u> field distribution map, one of the immediate goals of this experiment. For a complete explanation of the experimental procedures developed for this project, reference is made to the thesis work of Grady (5).

The major disadvantage of the transient thermoelectric technique is the time consuming procedure required to collect a sufficient number of data points. To ease the difficulty of this tedious task it was felt that the experiment could be automated. The automation of the experiment incorporates much of the existing equipment in the mouse irradiation room but required the construction of some new devices as well. A new mouse stage was constructed to enable movement of the thermocouple via a computer controlled stepping motor. A stepping motor indexer was constructed to control the thermocouple motor and an interface was built to allow communication to and from the irradiation facility to the Perkin Elmer 7/32 minicomputer. An analog system was also developed to allow direct recording of the thermocouple response into the 7/32 (2).

Through the use of this equipment the operator is relieved of the tedious work involved in positioning the thermocouple and transducer. Data analysis can be performed by FORTRAN programs on the 7/32 allowing for a great simplification and time reduction in the analysis.

This thesis describes the design and construction of the devices build specifically for the dosimetry project. Chapter 2 offers an overall view of the system with an explanation of the existing components. Chapter 3 gives the construction details of the mouse stage. Chapter 4 outlines the design and construction of the stepping motor indexer along with introductory remarks on the operation of the stepping motor. Chapter 5 describes the mouse room interface and includes the details required to program the interface from the 7/32. Chapter 6 illustrates some software that has been developed to control the mouse irradiation facility and Chapter 7 gives ideas on system improvements. The appendices include detailed schematics of the indexer and interface along with a reference of symbol definitions used throughout this thesis.

Chapter 2

Overall System Concept

This chapter will briefly describe the existing system configuration and also describe the changes to the system as detailed in later chapters.

2.1 Existing System

The existing system used for ultrasonic irradiation of tissue consists of a tank mounted on a millbase with three degrees of freedom (x, y, and z), controlled by stepping motors. One stepping motor for each degree of freedom allows this tank to move in relation to a statically mounted transducer. Mounting a specimen in a fixed position in the tank allows the transducer to scan the specimen along any of the three directions of movement. The motors are controlled by Slo-Syn SP1800 Indexers enabling accurate movement of the tank. Linear, optical encoders with digital read-outs are employed allowing for accurate measurement of movement in any direction.

2.1.1 RF System

A block diagram of the radio frequency (RF) system required to drive the transducer is given in Figure 2-1. The RF signal originates from the Hewlett Packard 8660A Synthesized Signal Generator and is fed into a double balanced mixer. A tone burst signal is generated by keying the mixer, the length of the envelope being determined by a timer and gating circuit. The gating circuit can also be driven by an external device such as the computer interface as explained in section 5.2. The tone burst signal is routed to an Amplifier Research model 1000L amp capable of 1 KW continuous output and 4 KW pulsed output. The amp output is fed into a network that provides matching to the quartz crystal transducer. The matching network routes the signal to the transducer.

The intensity of the ultrasonic field produced by the transducer is governed by the matching network. This network provides a capacitance voltage divider to determine the voltage drive to the transducer. The voltage can be adjusted by changing a variable capacitor (CS) in the matching network. The voltage across the transducer is related to CS by the equation:

$$V = 2.86 \times 10^{2} + CS \times 4.98$$

where V is in volts and CS is read directly from the angular encoder and scale. The intensity at a specific point in the

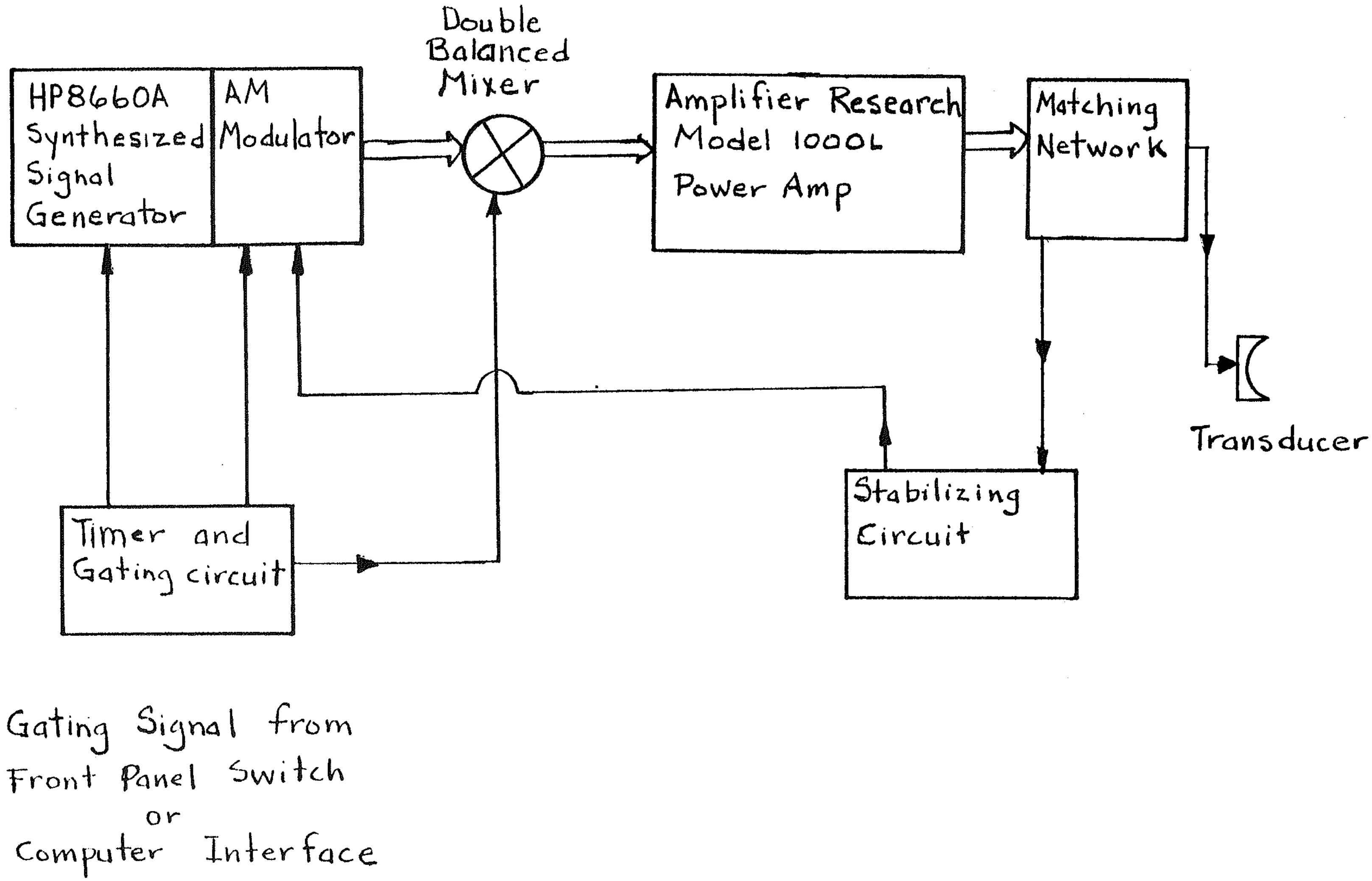

Double Balanced Mixer
HP8660A Synthesized Signal Generator
AM Modulator
Amplifier Research Model 1000L Power Amp
Matching Network
Transducer
Timer and Gating circuit
Stabilizing Circuit
Gating Signal from Front Panel Switch or Computer Interface

Figure 2-1. Sound Generation System--Block Diagram

field can be related to the voltage across the transducer by:

$$I = \left(\frac{V}{VCAL}\right)^2 \times ICAL$$

where I is intensity in W/sq cm and V is in volts. VCAL and ICAL are calibration constants determined for a particular transducer, VCAL is in volts and ICAL is W/sq cm.

The stabilizing circuit provides feedback to the AM modulation section of the HP 8660A. In this manner the circuit can fine tune the amount of drive delivered to the power amp during each shot. The operator can obtain course setting of the attenuation through manual adjustments made during initialization of the experiment.

2.1.2 Computer System

Existing computer control for this system originates with a PDP-8 minicomputer. The PDP-8 is interfaced with the irradiation facility allowing for control of the three directions of movement of the tank and the intensity, through the adjustment of the capacitor (CS). It can also turn the ultrasound on and control the shot length (the time the sound remains on). Figure 2-2 is a block diagram outlining the existing computer control. The PDP-8 can also read the tank

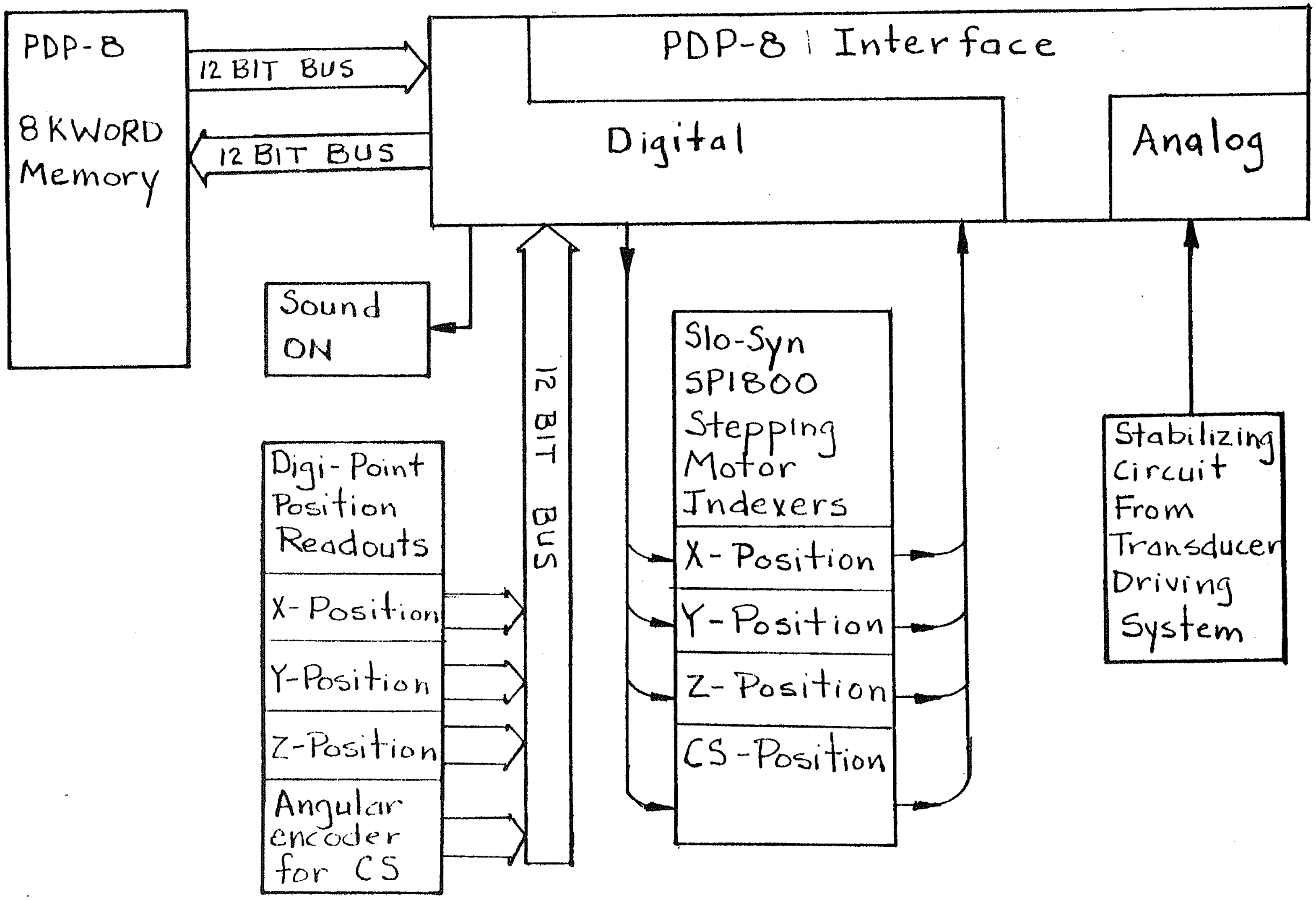


Figure 2-2. Existing Computer Control (PDP-8)--Block Diagram

position encoders and a position encoder on the capacitor (CS).

2.1.3 Thermocouple Circuit

The dosimetry project utilizes the transient thermoelectric technique as outlined in Chapter 1. A modified Keithley 149 milli-microvoltmeter is used to feed the thermocouple response into a galvonometer causing a light beam to deflect on photographic paper, thus recording the response. Figure 2-3 shows a block diagram for the existing thermocouple circuit.

2.2 New System

The new system will interface the existing system and some additional features required for the dosimetry experiment with the Perkin Elmer 7/32 minicomputer. In the early stages of this work it was felt that the irradiation facility should have the capability to be controlled by either the PDP-8 or the 7/32. The interface was built with the two machine capability as a design contraint, but, as work progressed with this project it became apparent that the PDP-8 would be retired and control would move exclusively to the 7/32. Therefore, the interface has the two machine capability, realizing only the 7/32 will be used for

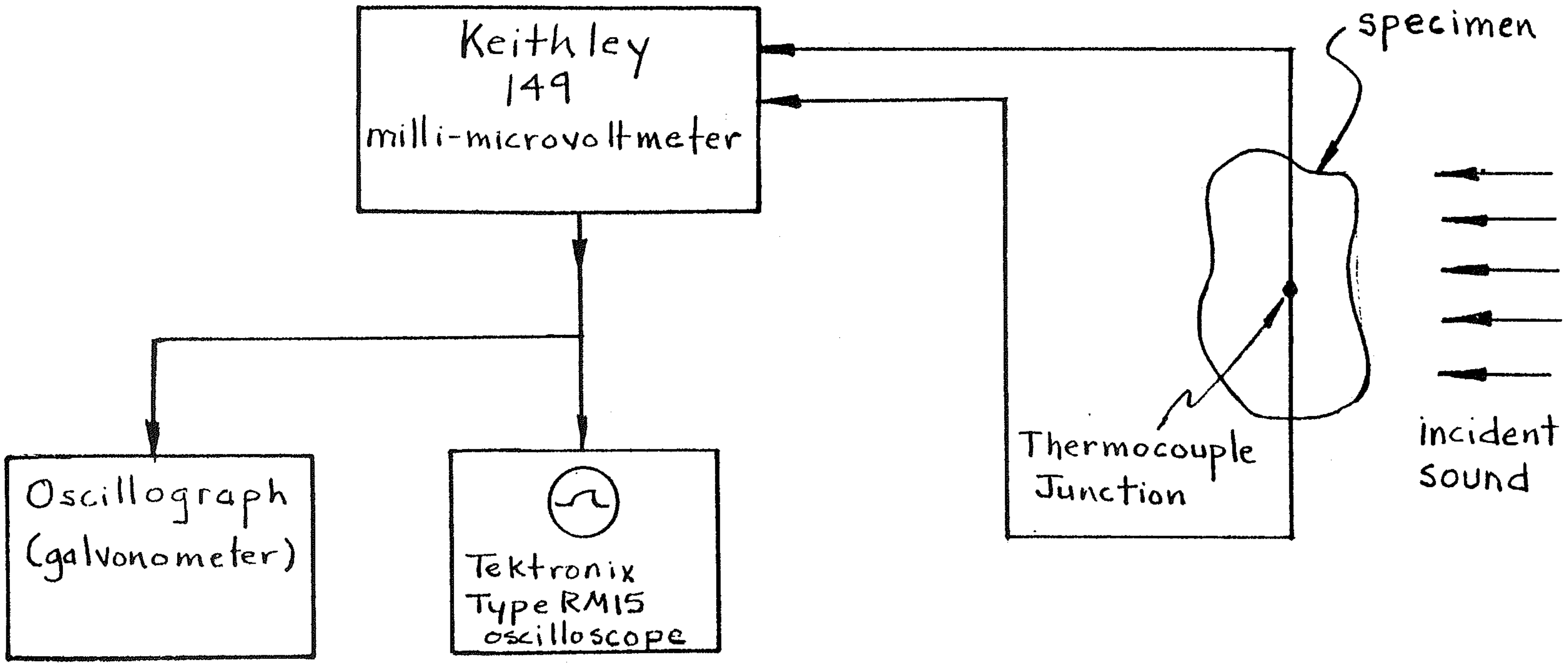


Figure 2-3. Existing Thermocouple Circuit--Block Diagram

controlling the experiments. Figure 2-4 is a block diagram showing the configuration of the new control system. General and detailed descriptions of this interface are given in Chapter 5 and will not be pursued here.

The new system will also have the added capability of thermocouple position control via a stepping motor. The motor can be controlled by an indexer or the interface. The stepping motor indexer is described in Chapter 4 and the movement mechanism is described in Chapter 3.

The new system will also use an instrumentation amplifier in conjunction with the A/D converter on the 7/32 to digitize the thermocouple response enabling computer analysis of the results. This instrumentation amplifier is presented in the thesis work of Duback (2) and will not be detailed in this presentation.

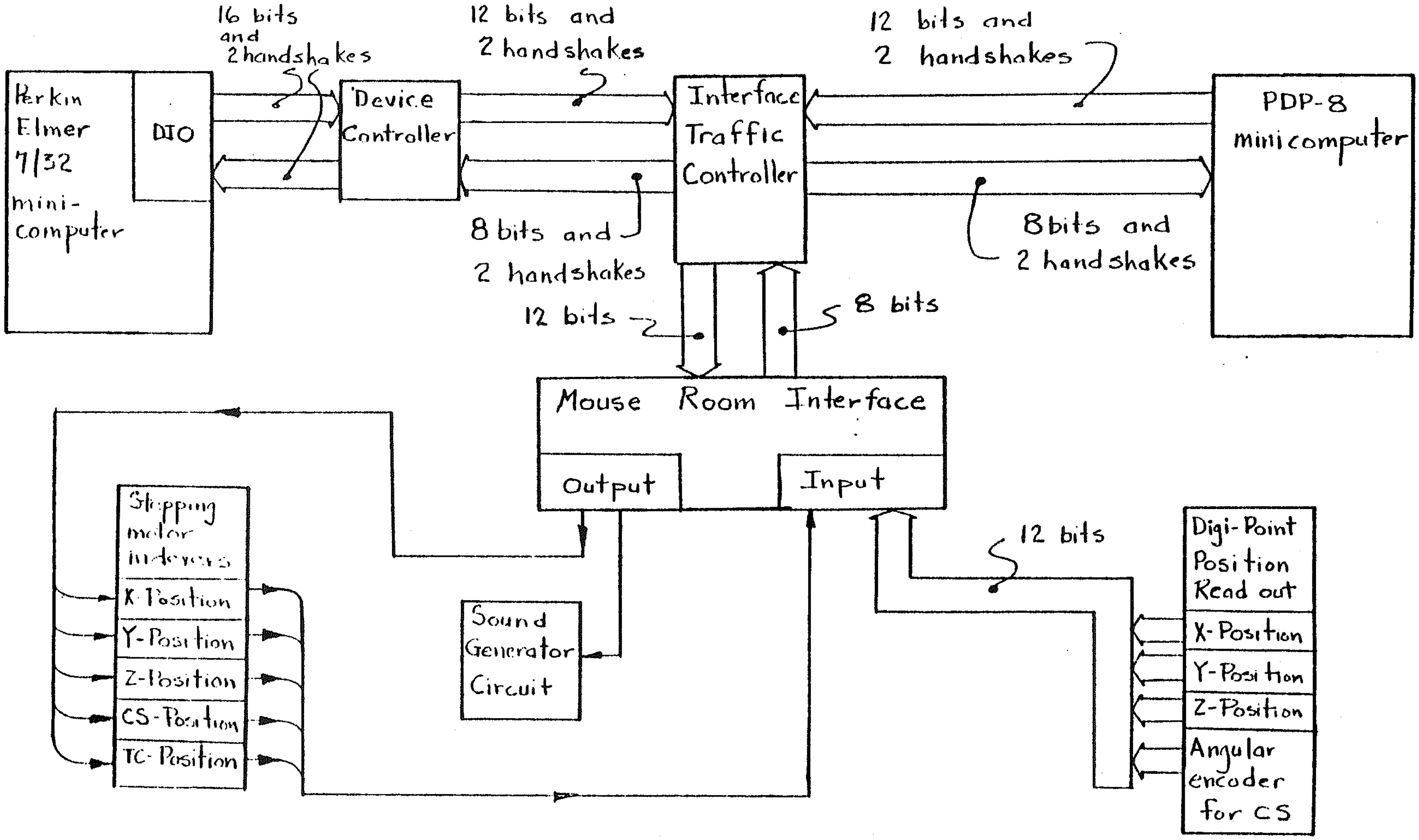


Figure 2-4. New Computer Control System--Block Diagram

Chapter 3

Description of Mechanical System

3.1 Design Considerations and Construction Details

This chapter will describe the thermocouple movement mechanism referred to as the mouse stage. The constraints considered in the design of the mouse stage as well as a description of the actual design are presented.

The basic purposes of the mouse stage for the dosimetry experiment are to:

a) horizontally move one or more thermocouple wires through a sacrificed or anesthetized mouse under stepping motor control.

b) Enable the operator to identify the thermocouple position and align the transducer accordingly during the set-up of the experiment.

c) Maintain an accurate record of the horizontal movement with an encoder or scale arrangement.

Along with these basic goals of the design it was considered important to keep the stage portable and to allow easy accessibility to the specimen for the purpose of inserting the thermocouples.

An important design constraint was the resolution of movement required to obtain accurate results in the experiment. During the development of the experimental procedures the thermocouple was usually moved in increments of 1 mm (5). One goal of the automated experiment would be to obtain a higher density of data points and a resolution better than 1 mm would be required. After consideration of the histological limitations in the data evaluation, 0.5 mm was considered an acceptable range of resolution. This resolution is exceeded through the use of a Warner Electric model M-305 ballbearing lead screw. The specifications (9) of this lead screw are given in Table 3-1. As seen from Table 3-1, the lead (linear travel per one revolution of the screw) is 0.050 inches or 1.27 mm and the backlash is 0.002 inches or 0.0508 mm. As discussed in Chapter 4, the stepping motor that drives this screw has 200 steps per revolution and one step is the smallest increment the motor can be driven to. Therefore in one step the lead screw could be moved 0.00635 mm, but the uncertainty of the backlash forces us to consider it as the limiting factor. From this we see that the resolution is limited to 0.0508 mm, well within our range of acceptable resolution. This analysis of the resolution

Table 3-1. Specification of Warner Electric Minature Ball Bearing Screw Model No M-305

Screw diameter	3/16 inch
Lead	0.050 inch
Pitch	20/inch
Operating Load Capacity	20 lbs.
Static Load Capacity	74 lbs.
Screw and Nut Material	Stainless steel
Lead Error	0.0005 inch/inch
Backlash	0.002 inch
Temperature Range	65 to 150 F

does not take into account the effects of the stage mass and friction encountered in the linear movement, but it does present an approximation to the magnitude of resolution possible with the chosen gearing arrangements.

Photographs of the constructed mouse stage are given in Figure 3-1 and Figure 3-2. The linear movement system will be discussed with reference to these photographs. The rearmost section of the stage that supports the motor and the specimen is stationary with the thermocouple movement coming from the Plexiglas frame in the forefront. This frame moves along two cylindrical rod runners. To enable ease of this linear movement, linear ball bushings (Thompson Industries Model A-4812-SS) are employed and are inserted into the Plexiglas frame.

The experimental procedure (5) requires that the mouse be frozen immediately after the conclusion of the irradiation procedure. It is important for the histological procedure that the mouse and thermocouple remain in the same orientation as when gathering irradiation data. Therefore the procedure is to leave the mouse, thermocouple and stage intact and place the stage into a freezing bath, consisting of ice, water, and salt. Within ten minutes enough freezing has occurred that the stage can be removed from the ice bath

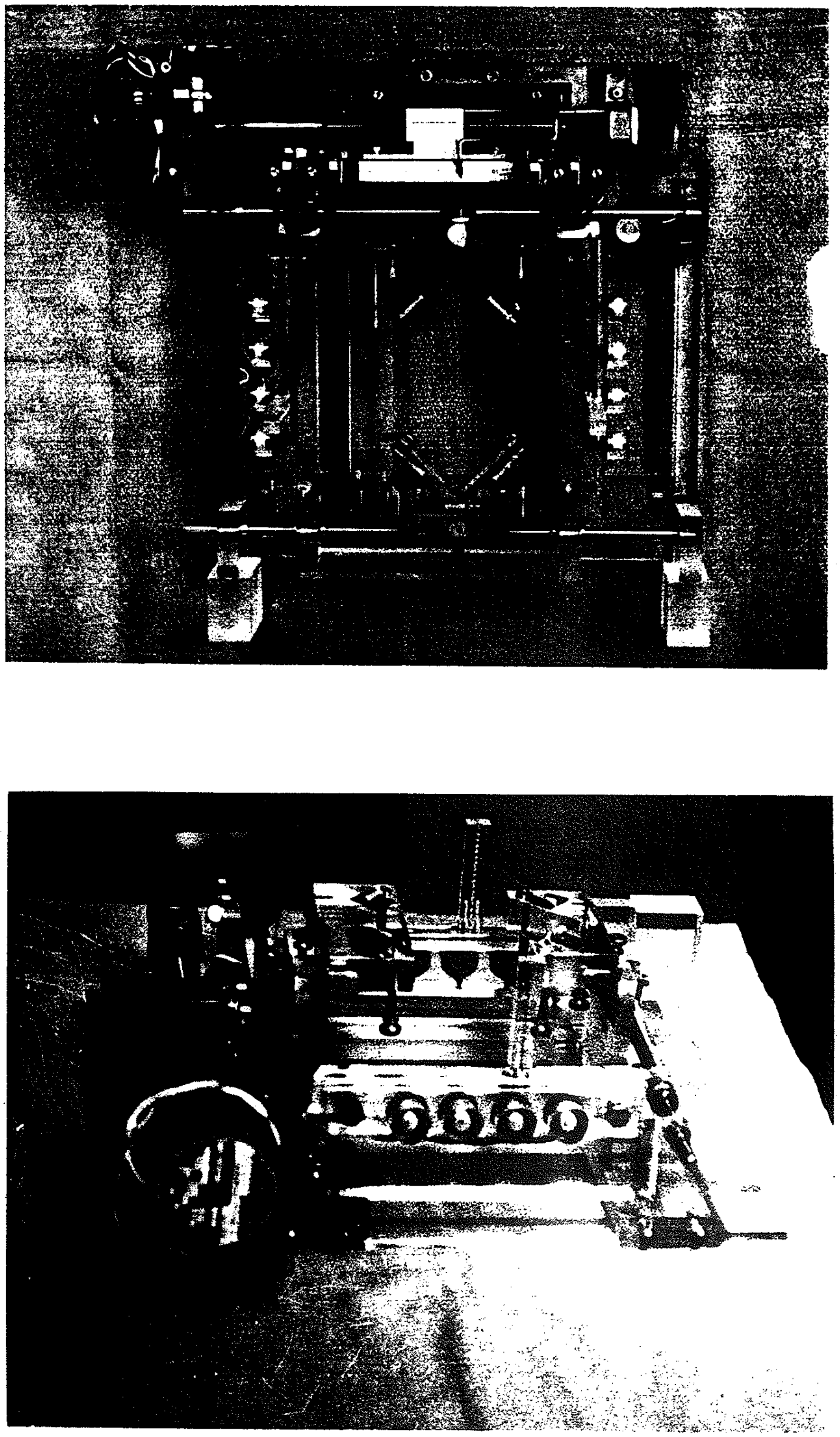


Figure 3-1. Mouse stage

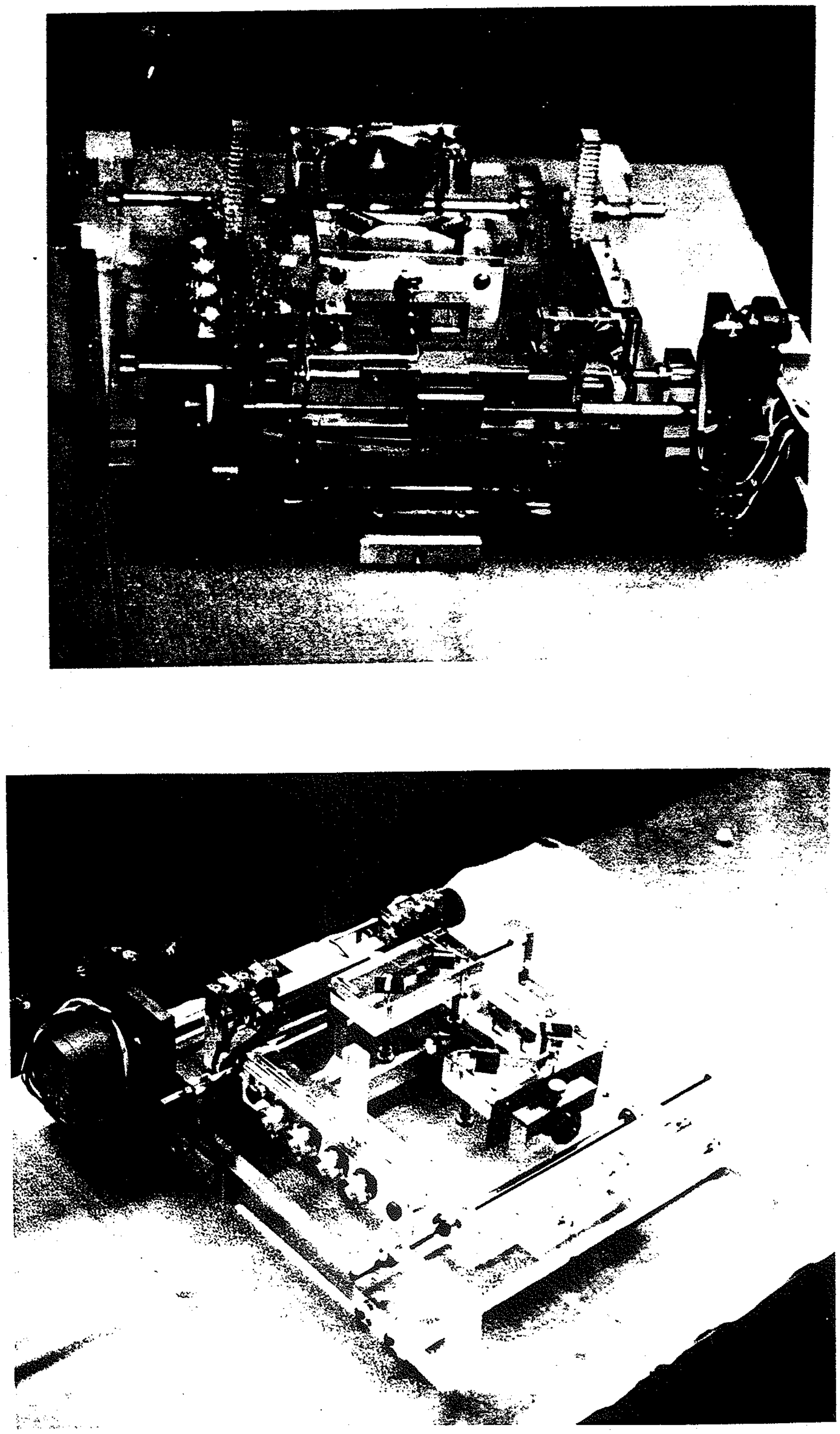


Figure 3-2. Mouse Stage

and the mouse removed from the stage. From this procedure it is evident that much of the stage will be subject to corrosive conditions. For these reasons the linear movement bushings and the rod are constructed of 440-C stainless steel to help prevent corrosion and allow for ease of movement. In other experiments the irradiation procedure requires saline solutions to act as the coupling medium between the transducer and the specimen, therefore this stage could be used for those experiments with less corrosion damage.

As mentioned earlier, one goal in a new mouse stage design would be to allow for multiple thermocouples to be inserted and moved in the specimen at the same time. The constructed stage has electrical connections for four thermocouples which will suffice considering four instrumentation amps will be built (2).

The electrical connections on the stage are designed and constructed taking into consideration the generation of thermal noise at dissimilar metal junctions (the thermoelectric effect). This effect is minimized by using crimped copper to copper connections wherever possible and where soldering is unavoidable, using Leeds and Northrup low thermal noise solder (L and N 107-1-0-1). The thermocouple connection on the moveable frame is shown in Figure 3-3. The

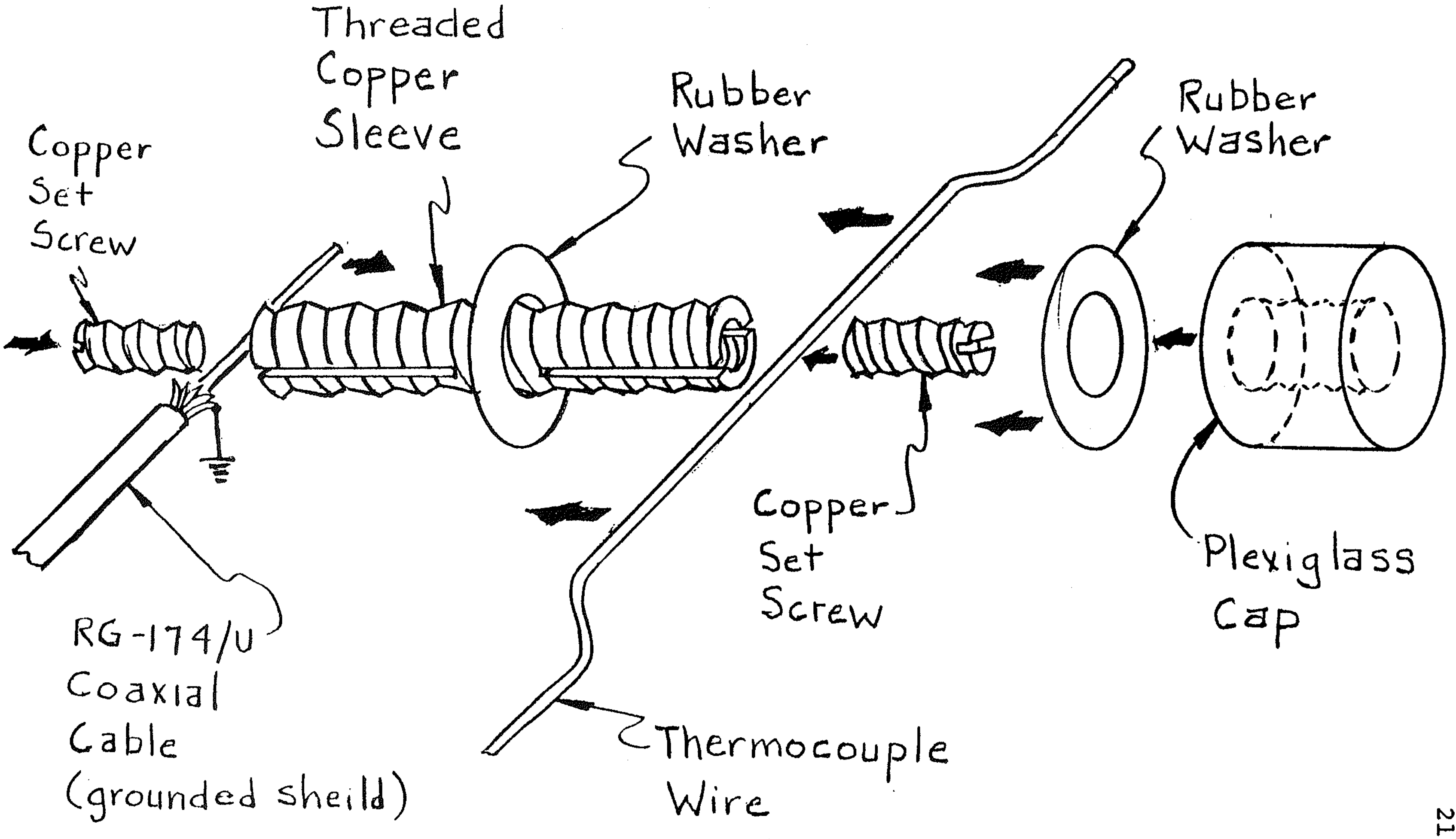


Figure 3-3. Mouse Stage Thermocouple Electrical Connectors--Expanded View

thermocouple is inserted into the slot and a copper set screw is tightened into the middle cavity. A rubber gasket is placed around the post and a Plexiglas cap is screwed onto the post. This forms a watertight seal to help eliminate any biasing problems that may arise if work in saline is pursued (2). The other connection at this post, to carry the signal to the Amphenol Hexagon connectors on the top of the stage, is made by a copper set screw abutting against the inner conductor of RG-174/U coaxial cable. This coaxial cable is terminated at the other end into the hexagon connectors. The ground side of this cable is left open at the thermocouple end but is connected through to the amplifier at the hexagon connector. The center conductor of the coaxial cable is crimped for two of the connectors and soldered (L and N 107-1-0-1) into the other two connectors.

It has been shown (4) that to obtain accurate acoustic absorption measurments using the thermoelectric technique the thermocouple junction and wire diameter size should be minimized. Therefore the mouse stage design should be capable of accommodating small diameter wire. In the experimental procedures (5) it was found that for this experiment a wire size of 76 micron diameter is chosen to obtain accurate absorption measurements and ensure mechanical strength for pulling the junction through the specimen. For the stage to accomodate this small, bare wire it is required to have easy accessibility to the thermocouple connectors and for the connectors to hold this wire without breaking it. The connectors explained earlier accomplish this. The stage also allows for smooth movement of the thermocouple through the specimen, eliminating the problem of a jerking motion breaking the junction.

Another important constraint borne out by the experimental procedure (5) is easy accessibility to the specimen during the set-up procedure. During the set-up, the mouse is sacrificed or anesthetized and all hair is removed from the area to be irradiated. The specimen can then be mounted on the stage and the thermocouples inserted. Accessibility for this purpose is maintained through a tiered design in the stage. The stationary part of the stage is the rearmost, the frame for moving the thermocouples is on the next tier and the mouse is mounted on the top tier. This keeps the mouse in front of the other parts of the stage and assures accessibility.

As mentioned earlier, this stage may be used for other experiments. The completed apparatus can now accommodate only a mouse, but the center portion of the stage that holds the specimen is removalbe allowing flexibility. Another center portion could be built to accommodate soft tissues and inserted into the stage.

The stage is also portable to enable movement from set-up table to water tank and then to the ice bath at the conclusion of the experiment.

The movement of the thermocouple can be monitored by a scale and pointer arrangement mounted on the moveable portion of the stage. Although this method provides for no direct thermocouple position feedback to the computer, it provides sufficient information for initial operation. Even with two or more thermocouples inserted into the specimen there is ease of the thermocouple movement and the stepping motor develops more than enough torque to accurately move the thermocouples to the requested position. As discussed earlier the backlash associated with the gearing is limited to 0.0508 mm, an order of magnitude smaller than the acceptable resolution. From these two considerations it can be concluded that during the course of an experiment negligible error would be introduced by not having closed loop type of control via the computer.

The computer can maintain position information on the thermocouple through software. The operator can enter the starting position and all subsequent moves logged with an ongoing record of the present position. See Chapter 6 for programming details.

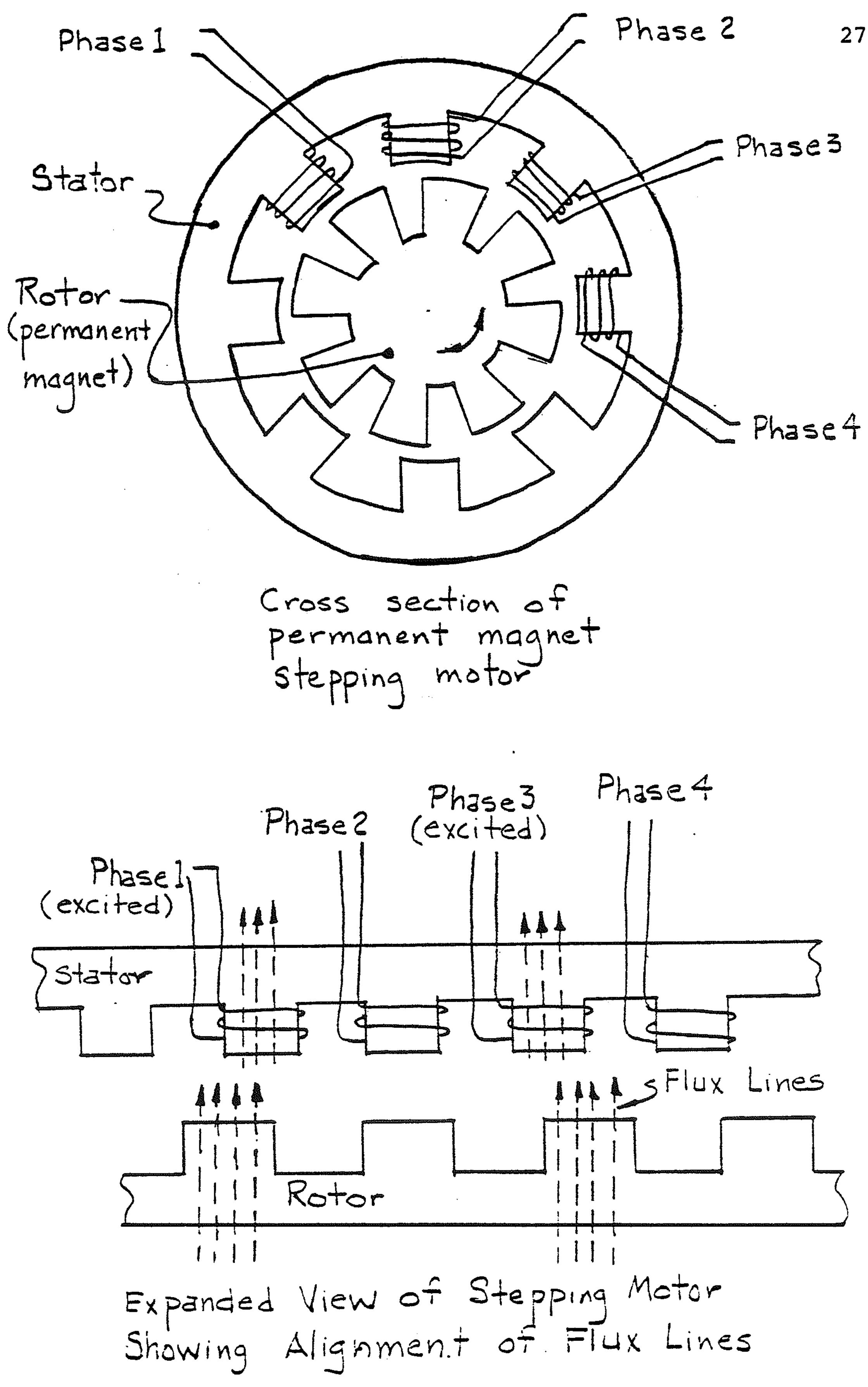
Chapter 4

Description of Stepping Motor Indexer

4.1 Principles of Stepping Motor Operation

To assure close tolerance and accuracy in the thermocouple movement, a DC stepping motor was chosen as the prime mover on the mouse stage. This section introduces the operating principles of the stepping motor and describes the motor chosen for the mouse stage construction.

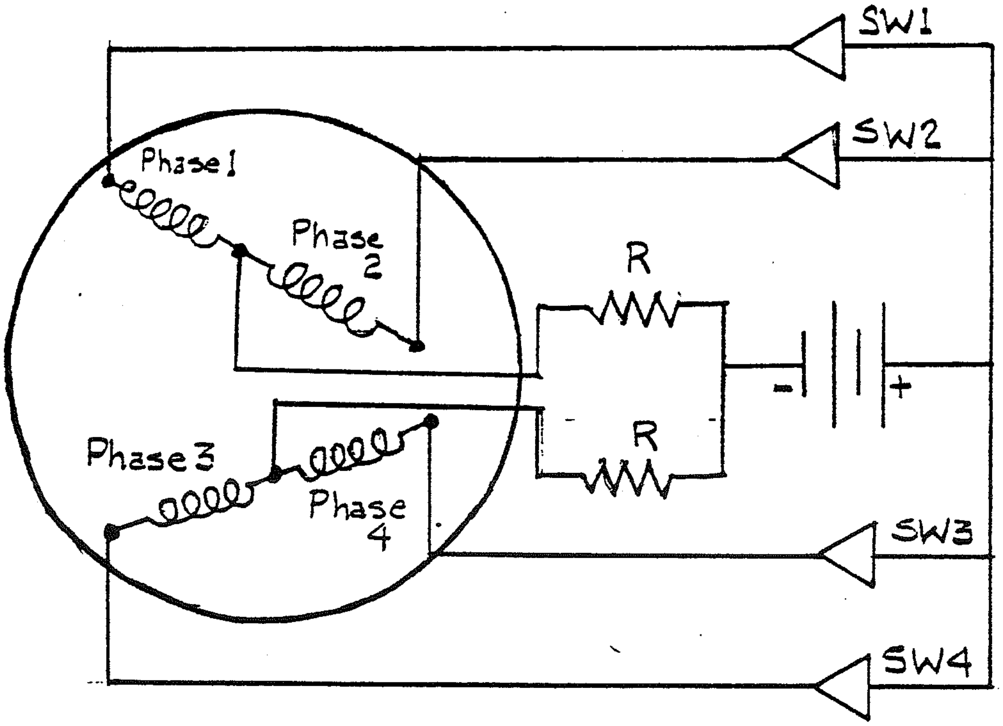
The stepping motor is a device that converts digital pulse inputs into a motor shaft movement. This movement can in turn yield accurate angular or linear motion. The motor is constructed such that one input pulse will yield a movement of one step or in some motors, one half step. From this it is seen that the motor has the ability to move to discrete positions in the angular revolution. Figures 4-1 and 4-2 help to illustrate the stepping operation. Figure 4-1 helps to conceptualize the construction of the stepping motor. This figure illustrates a 4-phase permanent magnet stepping motor similar to the one used in the mouse stage. To produce full step motions, two of the phases are excited and the motor assumes the position where the flux lines from the rotor and stator are maximally aligned. The excitation



Cross section of permanent magnet stepping motor

Expanded View of Stepping Motor Showing Alignment of Flux Lines

Figure 4-1. Stepping Motor--Cross Section and Expanded View



Schematic of 4-phase permanent magnet stepping motor and switching circuit

	STEP	SW1	SW2	SW3	SW4
Clock-wise	1	ON	OFF	ON	OFF
	2	ON	OFF	OFF	ON
	3	OFF	ON	OFF	ON
	4	OFF	ON	ON	OFF
	1	ON	OFF	ON	OFF
Counter Clock-wise	1	ON	OFF	ON	OFF
	4	OFF	ON	ON	OFF
	3	OFF	ON	OFF	ON
	2	ON	OFF	OFF	ON
	1	ON	OFF	ON	OFF

Stepping Sequence

Figure 4-2. Stepping Motor--Schematic and Step Sequence

is then moved onto another phase, resulting in a new position for maximum alignment. As this is repeated, the motor can rotate in a complete revolution. If the sequence is reversed the rotor rotates in the opposite direction. Figure 4-2 shows a schematic diagram of the stepping motor and the switching sequence required for clockwise (CW) and counterclockwise (CCW) rotation. To provide accurate position control an indexer is required to supply the correct sequence and number of steps (section 4.2).

The stepping motor used in the mouse stage is a Superior Electric, Slo-Syn model M061-FD301 stepping motor. This motor has a permanent magnet rotor and a 4-phase stator with a step resolution of 1.8 or 200 steps per revolution. The electrical characteristics are given in Table 4-1 (8).

TABLE 4-1
Electrical Specifications
Superior Electric Slo-Syn M061-FD301

1.	Time for single step. . . . . (24 volt. d-c drive)	7.5	msec
2.	DC Volts.	11.8	volts
3.	Amperes per winding	0.44	amps
4.	Winding Resistance. . . . . (25 C)	230	ohms per winding
5.	Inductance.	43	mH per winding

4.2 Indexer General Description

Figure 4-3 illustrates a stepping motor indexer that was constructed to control the Slo-Syn motor described above. This motor drives the thermocouple movement on the mouse stage.

The indexer can be controlled by its own front panel or by the 7/32 computer interface. Under front panel control, the user dials the number of steps required and depresses the "INDEX" button. This will move the stepping motor the requested number of steps in the direction dictated by the direction switch. Another mode of operation is to depress the "RUN" button. The motor will run in the direction determined by the direction switch as long as the "RUN" button is depressed. The "RUN" button has the added feature that if it is depressed during a move initiated by the "INDEX" button the move will stop. This feature can be used as a panic stop.

Upon the receipt of an index command the dial settings are latched and loaded into the counters. The clock circuit is then enabled through to the stepping sequence generator and the counter circuits. The loading and starting sequence is synchronous to assure that the correct number of steps are

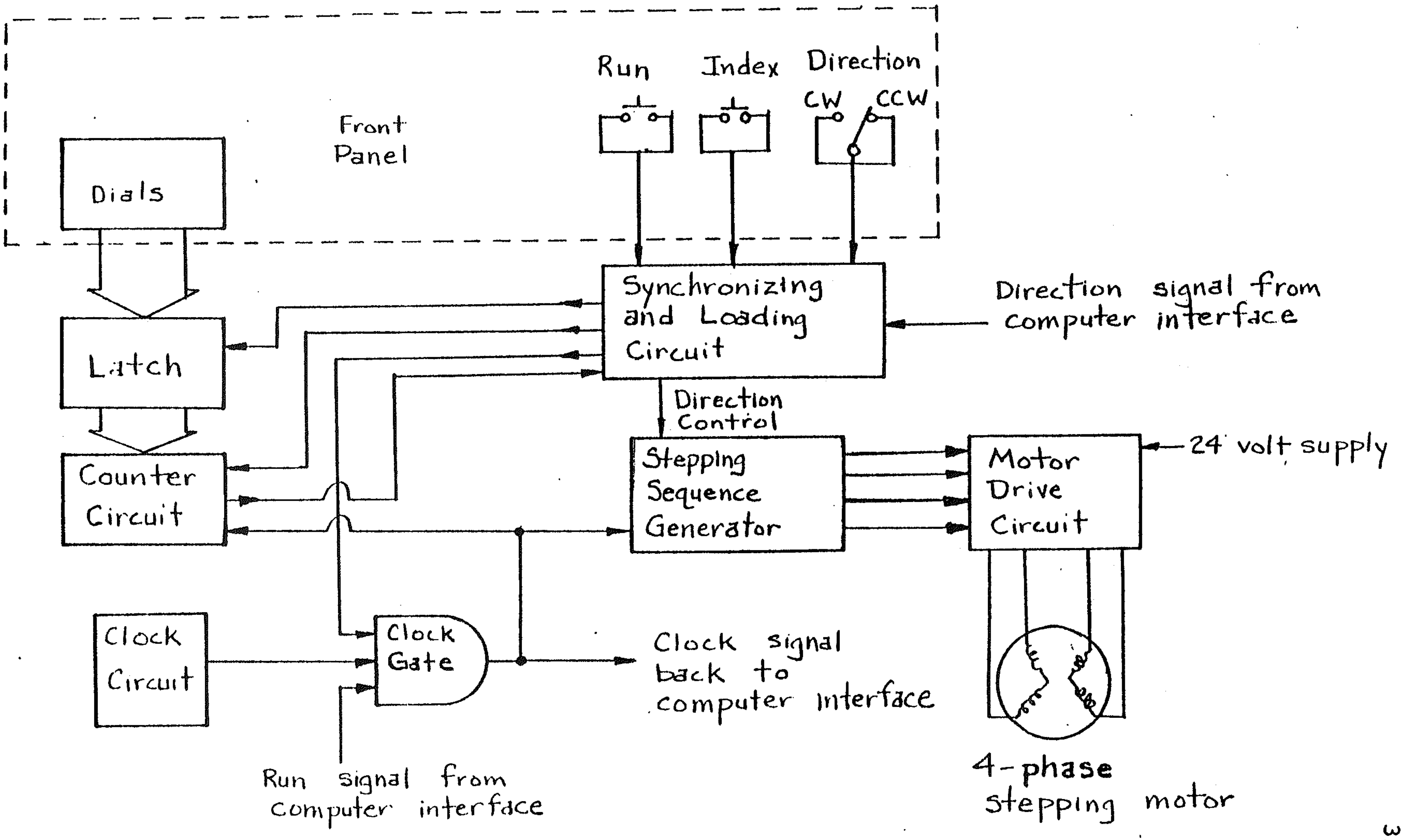

Run
Index
Direction
CW
CCW
Front Panel
Dials
Latch
Counter Circuit
Synchronizing and Loading Circuit
Direction signal from computer interface
Direction Control
Stepping Sequence Generator
Motor Drive Circuit
24 volt supply
Clock Circuit
Clock Gate
Clock signal back to computer interface
Run signal from computer interface
4-phase stepping motor

Figure 4-3. Stepping Motor Indexer--Block Diagram

always counted by the counting circuit. The dial information is latched enabling movement of dials during an indexed operation without effecting the ongoing movement.

The step sequence generating circuit produces the sequence as discussed in section 4.1. This sequence switches the 24 volt supply to the appropriate phases of the stepping motor.

The indexer can also be controlled by the computer interface. This is accomplished by supplying an enable signal from the interface that enables the indexer clock circuit through to the step sequencing circuit. The indexer clock is routed back to the interface where the steps are counted. The interface also supplies a signal used for direction control.

4.3 Description of Indexer Circuits

This section will give a detailed description of the stepping motor indexer circuits. Schematic drawings of the circuits are contained in Appendix A.

The stepping motor indexer consists of two wire wrap boards containing the digital logic and separately mounted motor drive electronics. A 5 volt supply for the logic circuits and a 24 volt supply for the motor drive are also contained in the indexer. TTL type devices are used exclusively throughout the logic design of the indexer. TTL is chosen because no special speed constraints are inherent in the design and these devices were readily available during the contruction of the indexer.

In the schematic drawings (Appendix A) every device is identified by an alphanumeric. For example, in Figure A-1 the device 1D13 is identified. The 1 identifies the board the device resides on and the D13 is a row and column identifier on the board. This technique is used in all the schematics in Appendix A and B.

4.3.1 Dial Circuits

Figure A-1 shows the circuit that contains the front panel dials and their associated interfacing circuitry. The dials are twelve position, non-shorting, ganged, rotary switches with the center wiper grounded. The switches are connected to board 1 through a Scotchflex connector on the end of the board. All the switch positions are tied high through 1000 ohm resistors on board 1. Each switch is decoded by a 10 line to 4 line encoder that produces a BCD output. This BCD output is inverted and latched into the 8-bit bistable latches if LTCHEN is high. The LTCHEN signal is generated by the sychronizing and loading circuit (section 4.3.3). Data is latched only when the "INDEX" button is depressed, enabling movement of the front panel dials during a move operation without adverse effects on the present operation.

4.3.2 Counter and Clock Circuit

The counter circuit is shown in Figure A-2 and the clock circuit is shown in Figure A-3. Also shown in Figure A-3 is the power on delay (POD) circuit.

The POD circuit provides a high output that is delayed 1 second when the power is switched on to the indexer. This signal is used to set flip-flops in the proper state at the initial turn-on.

The clock circuit is built around a 555 timing device. The values of the resistances (R1 and R2) and the capacitor (C1) determine the frequency of operation and the duty cycle. The frequency in hertz is given by

$$f = \frac{1.443}{(R1 + 2 \times R2) \times C1}$$

and the duty cycle (ratio of high time to the total cycle) is

$$D = (R1 + R2)/(R1 + 2 \times R2)$$

These equations would predict a frequency of 485 Hz and a duty cycle of 52.4% for the constructed circuit. The measured frequency of the system is 384 Hz.

The speed at which the stepping motor operates is determined by the clock circuit frequency. The clock signal is routed to the step sequence generator circuit and the speed of this step sequence determines the motor speed. Therefore, consideration must be given to the mechanical properties of the motor in determining the clock frequency. From the motor specifications (8) it is found that to maximize the motor torque it would be best to operate at 2000 steps per second, but high torque can also be obtained when operating at 300-1000 steps per second. The torque falls off when operating above 4000 steps per second.

It was found experimentally for the motor used in this application that at speeds above 600 steps per second unacceptable operation resulted if added measures were not taken. At the higher step speeds the motor's inertia could not be overcome and faulty operation resulted. To operate at these higher speeds adjustments must be made to the clock frequency during the start and stop portions of a move cycle. During the start portion of the cycle a slower step frequency is applied to the motor, allowing for the motor to overcome its inertia. As the motor increases in speed the step frequency is increased, until the desired speed is reached. The process is reversed to stop the motor.

For the movement of the thermocouple in the mouse stage it was not necessary to operate at speeds above 600 steps per second. Therefore the clock could be routed directly to the step sequence generator circuit without an interposing ramp-up and ramp-down circuit. As mentioned in section 4, the stepping motor has 200 steps per revolution and from Chapter 3 the lead screw gearing produces a linear movement of 1.27 mm per revolution. The measured frequency of the constructed indexer is 384 step per second which translates into an acceptable linear speed of 2.43 mm per second.

The counter consists of 5-74LS190 synchronous up/down counters. The up/down option is not utilized because the up/down input (pin 5) is tied high forcing the counter into the downcount mode. The data is loaded into the counters from the latches (section 4.3.1) when the CNTLD line is held low. The CNTLD is generated by the synchronizing and loading circuit (section 4.3.3). In this manner the counters are programmable with the data originating from the front panel dials.

The counters are connected in a parallel enable configuration. The ripple carry output (pin 13) of one stage is routed to the the clock input (pin 14) of the next stage. In this manner a ripple count is generated. The count enable input (pin 4) of all stages but the first are tied to ground. These stages are enabled and will count upon receipt of a low to high transition at their clock input. The count enable signal for the first stage (CNTEN) is generated by the synchronizing and loading circuit (section 4.3.3). A high to low pulse at the ripple carry of the last stage (2A7) signals that the programmed count is over and is fed back into circuitry to disable the counter (section 4.3.3).

4.3.3 Synchronizing and Loading Circuit

The synchronizing and loading circuit is shown in Figure A-2. The main function of this circuit is to provide enable or disable signals to the latch and counter circuits. It is desirable for these operations to be synchronous to assure that the proper information is loaded into the counters when the "INDEX" button is depressed. This circuit also allows the front panel dial settings to be changed during an ongoing index operation. Timing diagrams are given in Figures A-4 and A-5 to assist in the explanation of these circuits.

The timing diagram for the "INDEX" button buffering is given in Figure A-4. This circuit generates a pulse that is one clock period long each time the "INDEX" button is depressed. This assures that the initiation of an indexing operation is not dependent upon the length of time the button is depressed. The preset (PR) pin of the first D flip-flop is biased high but swings low when the switch is depressed. This brings Q1 high (2A1 pin 5) which is the D input to the second flip-flop. Upon receipt of a low-to-high transition of the clock Q2 comes high and stays high for one clock period, at which time the two flip-flops resume their original state, ready for another signal from the "INDEX" switch.

The timing diagram for the synchronizing and loading circuit is given in Figure A-5. This diagram shows that Q2 triggers Q3 to high, which enables the J-K flip-flops to begin toggling. The output of these flip-flops generate the signals LTCHEN, START, and Q7. The LTCHEN signal is used to latch the dial input data into the bistable latches (section 4.3.1) and is inverted to generate CNTLD which loads the output of the latches into the counters. The START signal is routed to a flip-flop that sets to generate the CNTEN signal. START is also inverted and sets a flip-flop to generate the GO signal. Q7 causes the CLR signal to go low, forcing Q3 low and disabling the J-K flip-flop. The CNTEN signal enables the counter and the clock transitions present at the first counter stage are counted until a ripple carry appears at the last stage. When RIP5 goes low the one-shot (2A4) is triggered, forcing GO low. GO triggers the one-shot (2A4), forcing CNTEN high which disables the counter. The circuit is now in the original state, ready for another signal to commence an index operation. The RUN signal can also force GO low, disabling the counters. This allows the operator to stop an index move by depressing the "RUN" button.

4.3.4 Step Sequence Generator Circuit

Figure A-6 shows the step sequence generator circuit. The two J-K flip-flops generate the signals used to produce the proper sequence of steps required to move the motor in the clockwise (CW) or counterclockwise (CCW) direction. The sequence of steps appear as the signals SW1, SW2, SW3, and SW4 (Figure 4-2) and are fed into the motor drive circuit.

The sequencing is started on the receipt of a high signal from GO (section 4.3.3) or RUN. RUN comes high anytime the "RUN" button on the front panel is depressed, thus the motor is driven as long as the "RUN" button is held down. The sequencing can also be started upon the receipt of a high signal from the computer interface (Chapter 5).

Anytime the motor is under interface control a front panel indication is given.

The direction of the move can be selected from the front panel or governed by the interface. If the indexer is under interface control the manual direction select is blocked from the sequencer and direction control must come from the interface. A high signal on 2C19 pin 1 dictates a CW move and a high on 2C19 pin 12 dictates a CCW move.

4.3.5 Motor Drive Circuit

The motor drive circuit is shown in Figures A-7 and A-8. Figure A-8 is a simplified schematic of the circuit, not showing the details of the connection scheme. For completeness Figure A-7 is included to show the details of the connections.

The signals SW1, SW2, SW3, and SW4 are fed into open collector buffers which drive the base of the 2N3055 transistors. When one of these signals goes high the transistor saturates, connecting the motor coil through to ground. The limit switches on the mouse stage must be in the proper state for the circuit to be complete. As seen in Figure 4-2 two of the phases will be on at any one time, thus the 50 ohm resistor must be capable of handling the appropriate current. The maximum current rating for each motor coil is 0.44 amps but in this circuit each coil is limited to approximately 0.2 amp. Therefore the 50 ohm resistor must be capable of handling about 8 Watts.

When the driving transistor turns off, the inductance inherent to the motor coil will generate a back emf that can be limited by the diode and 25 ohm resistor. The diode is commonly called a freewheeling diode and acts to short the back emf to the 24 volt supply. This arrangement provides assurance that an overvoltage condition will not appear across the motor coil. This feature can also increase the operating speed by quickly dissipating the energy stored in the coil.

Chapter 5

Description of Interface with 7/32 Computer

5.1 General Description

To provide control of the mouse room by the Perkin Elmer 7/32 minicomputer a new interface was constructed. As mentioned earlier, this interface has the capability of being controlled by two minicomputers but initially only the 7/32 will be in operation. The PDP-8 is used as a second machine in this presentation for illustrative purposes, realizing it is near retirement and the replacement status unknown. The interface accepts data from the 7/32 and decodes this information to provide appropriate signals for position control of the millbase (x, y, z), variable capacitor (CS) and the thermocouple (TC). The interface can also accept data from the mouse room and convey this information back to the 7/32.

An overall system configuration is given in Figure 2.4. Start-up operation initiates with the 7/32, putting it in control of the traffic controller. The traffic controller directs information to the mouse room or the PDP-8. After initiation, control can be transferred to the PDP-8 and returned to the 7/32 when needed. The configuration of control and interface output routing can all be accomplished through software. The 7/32 and PDP-8 both output 12 bits and 2 handshake lines which are all brought in parallel to the interface. The interface can output 8 bits and 2 handshake lines in parallel. The 12 bits selected by the traffic controller are routed to the decoding logic that selects which operation is to be performed. If this operation calls for an interface output, the 8 bits of information are fed back through the traffic controller to the appropriate machine.

Figure 5-1 is a block diagram of the interface. Twelve bits from the 7/32 and PDP-8 are fed into a 2 to 1 multiplexer which selects one as the source of information. The multiplexer outputs 8 bits of data onto a data bus and 4 bits of function code. The function code is decoded and this signal routed to the decode logic. The data bus feeds into the downcounters and the decode logic, enabling the counter to be programmed and started at the appropriate time. The clock bus is input to the counter clock providing the pulses that are counted. There are a number of inputs tri-stated onto this clock bus including pulses from all the positioning motors and the different clock frequencies generated by an on-board crystal controlled oscillator.

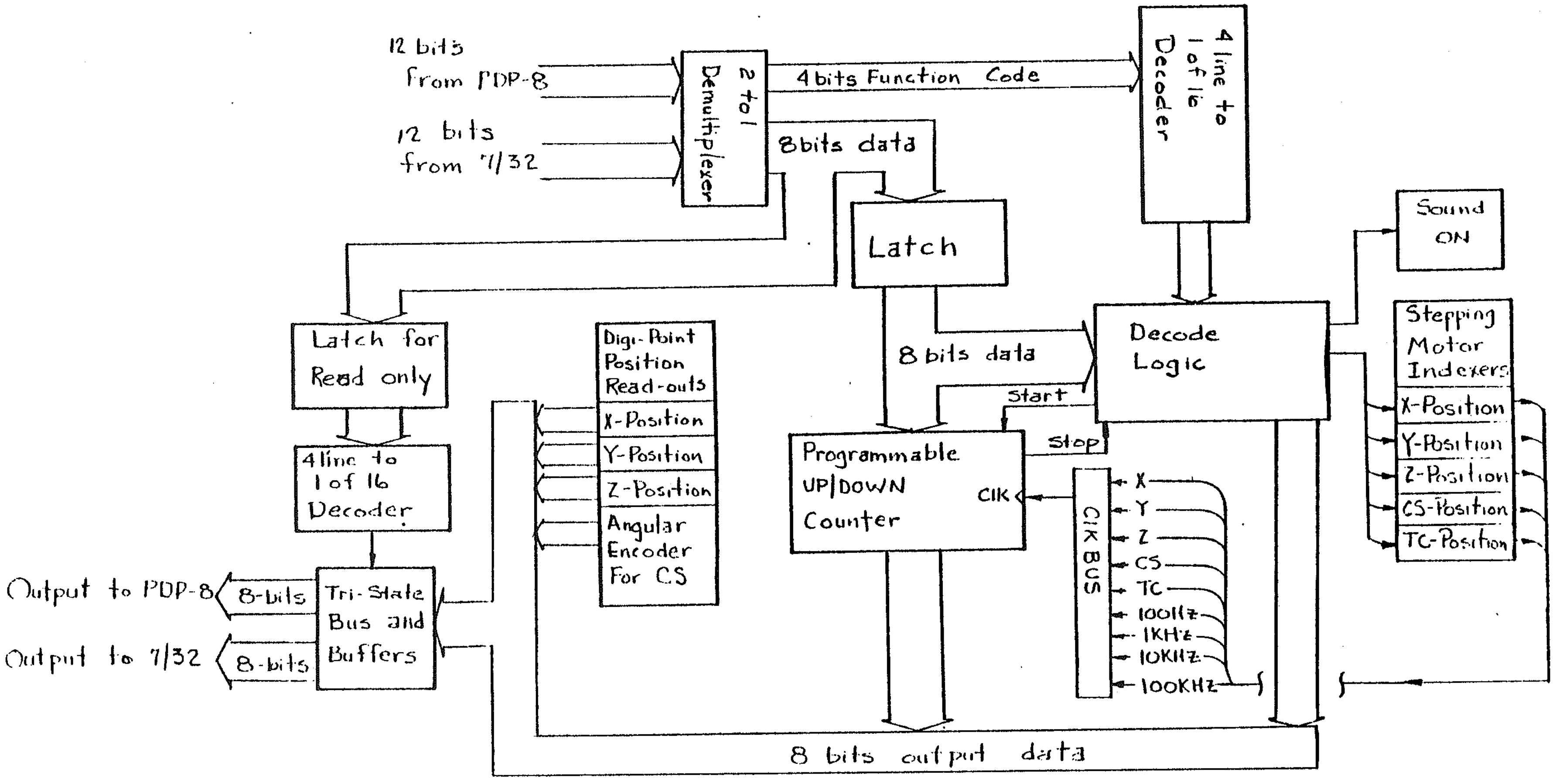


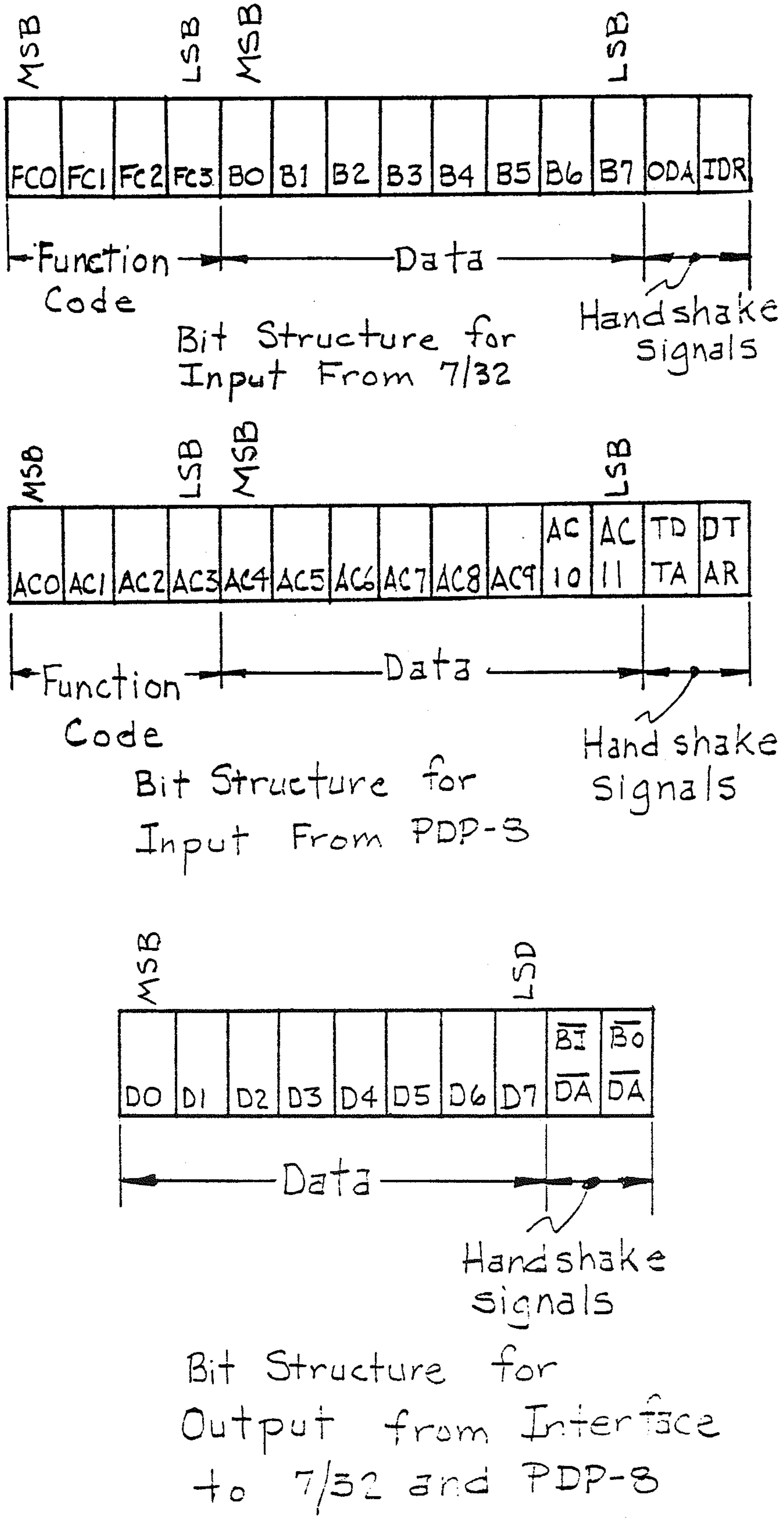
Figure 5-1. Mouse Room Interface--Block Diagram

A positioning operation can be implemented by loading the downcounter with the required number of steps for the stepping motor, enabling the steps to be counted by the downcounter and turning on the appropriate motor. The decode logic starts the downcounter and is signalled at the end of the count. A sound shot can be taken by a similar procedure.

There is also an 8 bit output bus that is routed to the 7/32 and PDP-8 for reading information from the interface to the computers. The downcounter and decode logic data output are tri-stated onto this bus and made available at both the 7/32 and PDP-8 at all times. To read the data into one machine a handshake signal is enabled through to that machine.

5.2 Bit Structure and Programming Considerations

This section will detail the bit structure used for the decode logic in the interface. Figure 5-2 illustrates the nomenclature associated with the 12 bit input and the 8 bit output data groups of the interface. The first four most significant input bits are designated the function code. These bits select the operation or "function" the interface will perform. The next 8 bits are the data bits containing information for the downconter or device selection. The bits are numbered from the most significant (B0) to the least



Bit Structure for Input From 7/32

Bit Structure for Input From PDP-8

Bit Structure for Output from Interface to 7/32 and PDP-8

Figure 5-2. Interface Input and Output Bit Structure

significant (B7). The handshake lines are ODA (output data available from the 7/32) and IDR (input data request from the 7/32). The output from the interface consists of 8 bits of data, numbered from the most significant (D0) to the least significant (D7). The two output handshake lines are BODA (output data received from 7/32) and BIDA (input data available from interface).

The bits from the PDP-8 are numbered consecutively with no differentiation made between the function code and the data bits. The first four bits (AC0-AC3) are used for the function code and the last 8 bits (AC4-AC11) are used for the data. The PDP-8 is a 12 bit machine and all bits are used for the interface. The 7/32 is a 32 bit machine with a 16 bit DIO (Digital Input/Output). The first 4 bits of this half-word digital output are used for device selection at the 7/32, (section 5.3) leaving the last 12 bits for interface use.

Figure 5-3 shows the function code assignments. The four bits are decoded into one of sixteen possible function codes. At present only 8 function codes are being used. Function codes 1, 2, and 3 load the appropriate stages of the downcounter with the data contained in bits B0-B7. Downcounter 1 (DC1) contains the least significant bits while

	← Data Bits →							
Function Code	B0	B1	B2	B3	B4	B5	B6	B7
0								
1 Load Downcouter 1 (Least significant bits)	← DATA →							
2 Load Downcounter 2	← DATA →							
3 Load Downcounter 3 (Most Significant bits)	← DATA →							
4								
5 Turn on Sound *		RF	ClK1	ClK2	ClK3	ClK4		
6 Turn on Motor	Dir.			TC	X	Y	Z	CS
7 Count Pulses from motor (upcounter)				TC	X	Y	Z	CS
8 Stop counter	S							
9 Read	(see figure 5.4)							
10								
11								
12								
13								
14								
15								

* ClK1 = 100Hz
ClK2 = 1KHz
ClK3 = 10KHz
ClK4 = 100KHz

Figure 5-3. Function Code Assignments

downcounter 3 (DC3) contains the most significant bits. Function code 5 will enable a clock signal through to the downcounter. If B1 is set to a logical one the ultrasound will be turned on. A one in B2, B3, B4, or B5 will select the clock frequency enabled through to the downcounter. In this manner a number (corresponding to a time duration) can be loaded into the downcounter and an appropriate clock signal input for counting.

Function code 6 will turn on a positioning motor. A one in B3, B4, B5, B6, or B7 selects the motor, while B0 selects the direction of movement. To move a required distance, a number is loaded into the downcounter and the positioning motor is turned on, also enabling the step pulses from the motor to be input into the counter. Function code 7 will count step pulses from the positioning motors. A one in B3, B4, B5, or B6 will determine which motor pulses are counted. Function code 8 will disable the downcounter and can be used to stop a move or turn the sound off. B0 must be set to a one for this operation.

Function code 9 allows for data to be read from the interface to the 7/32. The details of this function code are given in Figure 5-4. The last 4 bits of the data string (B4, B5, B6, and B7) are decoded into a 1 of 16 signal. This

Function code 9 (Interface Output)

Least Significant 4Bits in Hexadecimal	Bit Group Read
0	X-Position, High bits
1	X-Position, Low bits
2	Y-Position, High bits
3	Y-Position, Low bits
4	Z-Position, High bits
5	Z-Position, Low bits
6	CS-Position, High bits
7	CS-Position, Low bits, Sign for X, Y, Z
8	Immediately proceeding data
9	Check Bits
A	Downcounter 1
B	Downcounter 2
C	Downcounter 3
D	
E	
F	

Figure 5-4. Function Code 9 (Interface Output) Details

signal enables a certain output to be tri-stated onto the output bus. In this manner the interface can output information from the position encoders and the downcounters. It can also output a string of "check" bits that provide the status of the interface. The data contained in B0-B7 can also be output.

The x, y, and z position encoders each output 4 BCD numbers, totaling 16 bits of information apiece. Therefore, two read operations are required to transfer this data to the 7/32. Each encoder has High bits and Low bits. The High bits correspond to the BCD output of the ones and tenths place of the encoder while the Low bits are the hundredths and thousanths places. The CS encoder requires only 13 bits leaving the last 3 bits open to use for indicating the sign of x, y, and z. There are 8 bits of status information in the check bits and each downcounter can output 8 bits.

To provide an aid for software development refer to Chapter 6. Figure 6-1 summarizes all the functions of the interface with the appropriate hexadecimal half-word.

From this general discussion it can be seen that the interface is designed in a manner to facilitate future expansion. The interface can be expanded to include more function codes performing different operations in the mouse room or adjacent experimental rooms.

5.3 Interface Detailed Description

The following paragraphs provide a detailed description of the circuits constructed for the mouse room interface. The schematics for these circuits appear in Appendix B and reference is made to Appendix C for symbol definition.

The interface consists of six circuit boards arranged in a chassis mounted frame surrounded by a custom made enclosure. This frame is mounted in the mouse irradiation room approximately 75 feet from the Perkin Elmer 7/32 minicomputer. The circuit boards are wire wrap type with edge connectors provided for data transfers within the interface. TTL type logic is used because of its ability to fulfill any speed requirements imposed by the interface and its availability during the construction stage.

Power for the logic circuits is provided by a Power One modular D.C. power supply rated at 5 volts D.C. and 3 amps. Some of the existing mouse room equipment has a + or - 15 volt requirement, and to supply these levels the interface also has a Power One modular D.C. power supply rated at + or - 15 volts D.C. and 1 amp.

5.3.1 Perkin Elmer 7/32 DIO and Interfacing Circuits

A block diagram of the 7/32 DIO (Digital Input/Output) is shown in Figure 5-5 (7). The DIO has a 16 bit output and 16 bit input capability along with the ability to output and input 2 handshake signals. The handshake signals are used to synchronize data transfers to external devices. The handshake signals available are ODA (output data available), ODR (output data request), IDA (input data received) and IA (input available). The ODA signal is a 1 microsecond pulse that signals that the processor data is settled on the lines and is ready to be output to a peripheral device. The ODR signal is a 1 microsecond pulse originating from a peripheral device and signals the processor to output data. The IA signal is a 1 microsecond pulse signifying that a peripheral device is requesting to input data to the processor. The IDR signal is a 1 microsecond pulse that acknowledges data has been accepted by the processor. In the present configuration of the mouse room interface the ODA signal is used for data transfer operations. Future improvements of the system might include use of the IA signals (see Chapger 7).

Figure 5-5 also illustrates the circuit used for transmitting the data to the mouse room interface. The DIO output is TTL compatible and is fed directly into an open

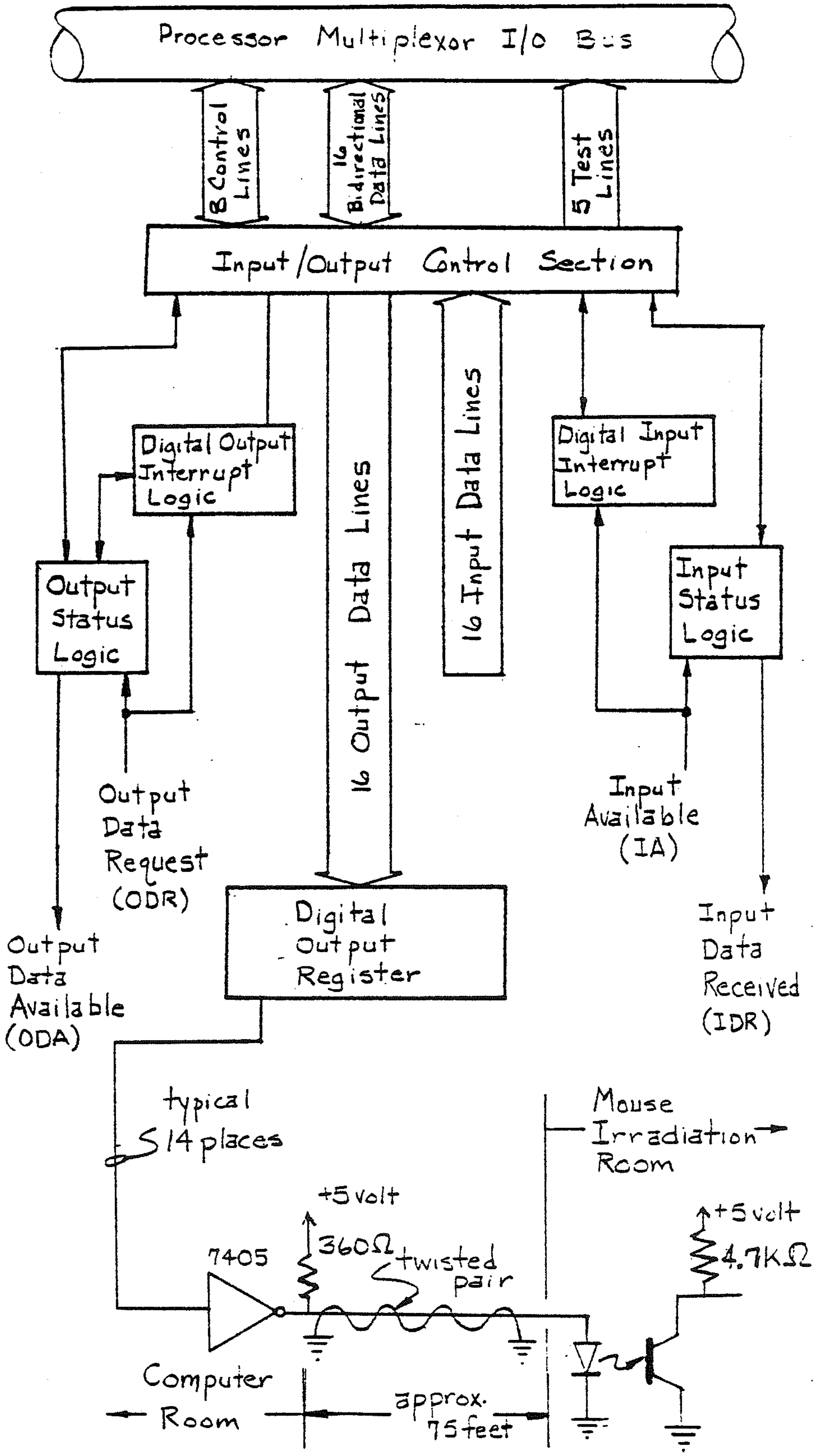


Figure 5-5. Perkin Elmer 7/32 DIO (Digital Input/Output) Configuration

collector hex inverter. The output of this inverter has a 360 ohm pull-up resistor and feeds directly into a twisted pair cable that carries the data bit and a ground signal 75 feet to the interface. At the interface the signal is received by an opto-isolator that provides isolation between the two systems (see section 5.3.2). For construction details of the circuit that accepts data from the 7/32 see Duback's thesis (2).

Future development of the DIO will include a system controller (7) that will decode the four most significant bits of the digital output and select the appropriate device number. The remaining 12 bits of digital output are routed through the corresponding device controller and onto the device. This system will enable different experiments requiring input and output capability to run concurrently on the 7/32. The mouse room interface will require one device number and has been assigned device number 13. For programming the device the inverse of this must be used which translates to the number 2. Therefore, to accomplish a write operation to the interface when the system controller is in operation the hexadecimal number loaded into the DIO must begin with the number 2. All the software developed for the interface has utilized this concept.

5.3.2 Data Receiving Circuit

The data receiving circuits are shown in Figures B-1 and B-2, Appendix B. In figure B-1 the data arrives from the 7/32 into a Scotchflex connector. The most significant function code bit corresponds to BIT4 from the 7/32, because BIT0 through BIT3 have been used to select a device at the 7/32 system controller. The data is routed from the connector to MCT2 opto-isolators, providing isolation between the 7/32 and the interface system. From the opto-isolator the signal is routed to 2 to 1 multiplexers. The other input to these multiplexers originates with the PDP-8 and is brought on-board via a Scotchflex connector. The strobe input (pin 15) of the multiplexers is kept low to enable operation. The select input (pin 1) is set by a front panel switch and determines which input is fed through to the interface. The select switch is shown in Figure B-7. To implement full system capability this switch would be replaced by appropriate hardware to enable software control of the selection process.

The multiplexer output is fed into the latches 4C1 and 4A1 shown in Figure B-2. The function code bits are always latched into 4C1 upon the receipt of a high pulse on MODA. The data bits (MB0 through MB7) are latched into 4A1 on the receipt of a high pulse on BODA when function code 9 is not present. Function code 9 would initiate a read operation and the data bits are latched into a separate circuit. The latch outputs are routed directly into a series of inverters that change the data into positive logic and act as buffers to increase the fan-out for the data and handshake signals. These inverted signals, FCO through FC3 and BO through B7, are then connected to the rear edge connectors.

Figure B-2 also illustrates a one-shot that accepts a 30 microsecond pulse on the ODA line and ouputs a delayed 30 microsecond pulse (BODA). Upon receipt of the time delayed BODA signal, assurance is made that the input lines are settled to a final value. Figure 5-6 is a timing diagram that shows the relationship between the handshake and the data signals. Figure B-2 also shows a one-shot for IDR. The IDR and time delayed BIDR signal would be used as handshake lines for communication back to the the 7/32 (see Chapter 7).

The four function code bits, FC0 through FC3, are routed from the inverters to a decoder on board 5 (5A22) illustrated in Figure B-11. The strobe input (pin 18) is connected to the BODA signal, assuring that only stable data is loaded into the decoder. The decoder brings one of its sixteen output lines low according to the binary input. This low

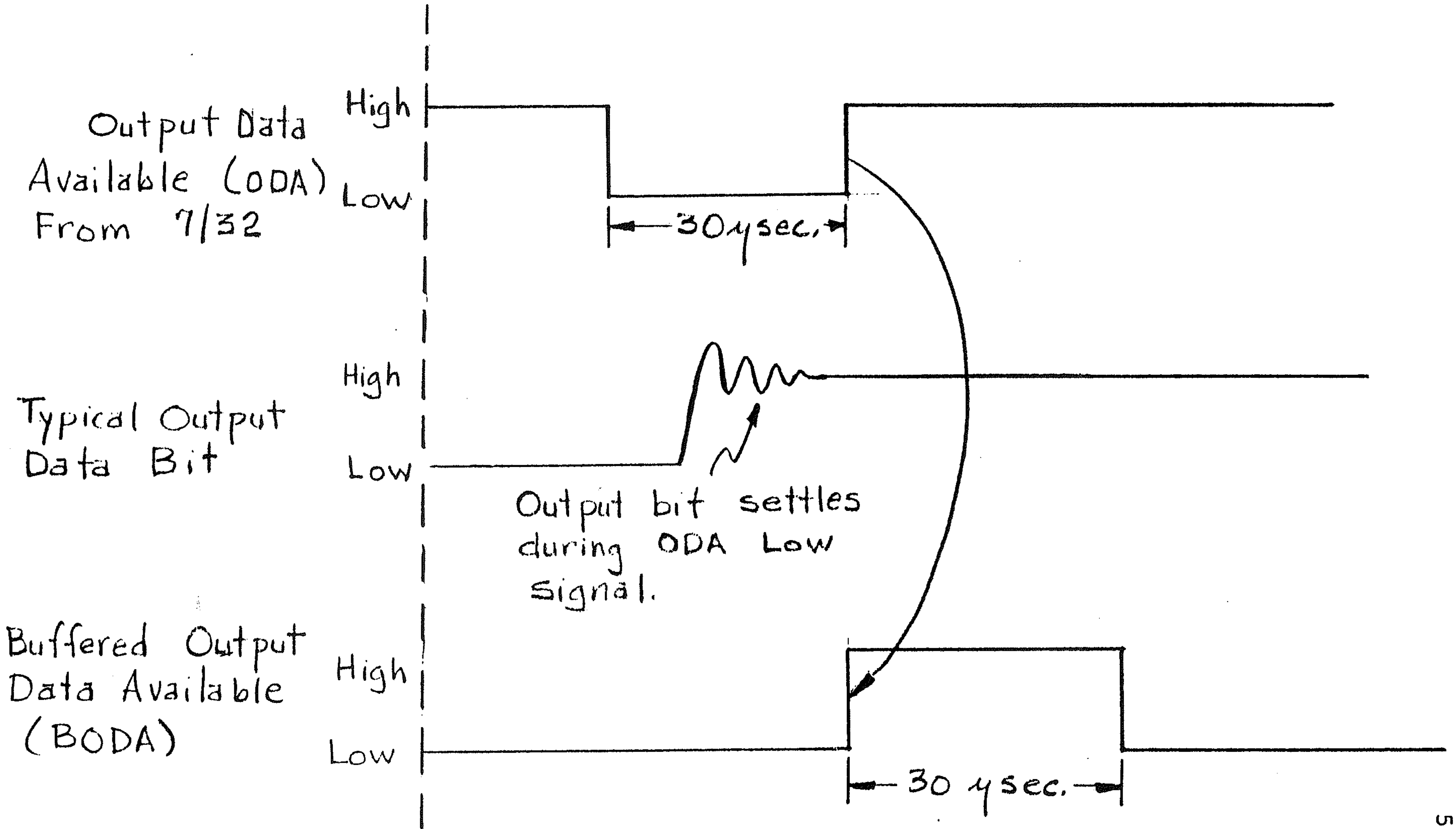

Output Data Available (ODA) From 7/32
High
Low
30 μsec.
Typical Output Data Bit
High
Low
Output bit settles during ODA Low signal.
Buffered Output Data Available (BODA)
High
Low
30 μsec.

Figure 5-6. Perkin Elmer 7/32 Output and Handshake Signals--Timing Diagram

output signal is used to further control the operation being performed.

5.3.3 Counter Circuit

The counter and its associated circuitry are shown in Figures B-3, B-4, and B-5 in Appendix B. The counter has a 24 bit capability and is made of 6 stages, each stage consisting of a synchronous 4 bit up/down binary counter (74LS169). The stages are connected in a parallel clocking and serial enabling configuration. This configuration allows for high speed operation by clocking all the stages at once with a count occurring only on the enabled stages. Each stage has two enable inputs (pins 7 and 10) that must both be low for counting and a ripple carry output (pin 15) that comes low once on each modulus of counting. The ripple carry outputs and the enable inputs are connected to provide a serial enable signal to each stage.

The data that can be loaded into the counter are limited to 8 bits at one time. Therefore, to load the entire 24 bits, three loading operations must occur. For this purpose the counter is subdivided into 3 groups, each containing 2 stages. Downcounter 1 (DC1) consists of the least significant bits, downcounter 2 (DC2) has the next 8 bits and downcounter 3 (DC3) contains the most significant 8 bits. To load 8 bits of data into one of these groups the information must appear on the 8 data lines (B0 through B7), the load input (pin 9) must be brought low and the counter must receive a clock pulse, assuring synchronous operation. The circuit used to generate this sequence is given in Figure B-5 with a timing diagram in Figure B-6. A signal to load DC1, DC2, or DC3 will come on FUNC1, FUNC2, or FUNC3, respectively. The two D-type flip-flops generate the LDEN signal that enables one cycle of the 1 kHz clock signal to be input into the counter via the clock bus (see Figure B-8). Simultaneously the load input is held low by the LDDC1, LDDC2, or LDDC3 signals.

To start a counting operation the DCEN signal must be brought low. The circuit that generates DCEN is shown in Figure B-4. The function code signals, FUNC5, FUNC6, or FUNC7 are used to clock a D-type flip-flop and this brings DCEN low. The counter is stopped upon the receipt of the DCOVER signal that is generated by the ripple carry of the last stage or upon the receipt of a FUNC8 signal aong with a high signal on B0. This last feature allows a software controlled stop of the counter.

The output of each group of counters, DC1, DC2, and DC3, can be read onto the output bus by supplying a low input signal on RDDC1, RDDC2, or RDDC3. The signals RDDC1, RDDC2, and RDDC3 are generated by the read circuit shown in Figure B-12 (see section 5.3.6). The counter can be used as an upcounter by supplying a high signal at the UD input (pin 1). This is accomplished by loading a function code 7 into the interface.

The clock input to the counter is supplied by the clock bus (CLKBUS). The various input signals tri-stated onto this bus are the four clock frequencies (see section 5.3.4) and the step pulses from the positioning indexers for x, y, z, CS, and TC (see Figure 5-1). After completion of appropriate decoding, one of these signals is routed through to the counter clock.

5.3.4 Timing Circuit

The circuits used for the generation of the four clock frequencies and the associated logic circuits are shown in Figures B-7 and B-8. A 1 MHz signal is generated by the crystal oscillator in Figure B-7. This signal is divided into 100 kHz, 10 kHz, 1 kHz, and 100 Hz by the four decade dividers (7490). These four clock signals, CLK4, CLK3, CLK2, and CLK1, respectively, are input to the tri-state gates that are attached to the CLKBUS in figure B-8.

There are two methods to determine which clock will be connected to the CLKBUS. In the manual mode the front panel toggle switch is turned to the "Manual Clock Control" setting and the frequency is selected by the 4-position switch. Upon receipt of a high signal on FC5EN the clock signal will be enabled through to the CLKBUS. In the computer mode, the front panel toggle switch is turned to "Computer Clock Control." The frequency is selected and enabled through to the CLKBUS by setting the appropriate data bit high (B2, B3, B4, or B5) and bringing FC5EN high.

Figure B-7 also shows the Power On Delay (POD) circuit, which provides a 1 second delayed signal when the power is applied to the interface. The POD signal is used to provide intitial set-up signals to flip-flops and other devices in the circuit.

5.3.5 Interface to Motor Indexing and RF Generation Circuits

Figure B-9 shows the circuits required to supply signals to the motor indexers and the sound generation circuit. Figure B-10 shows the circuit that accepts pulses from the indexers and routes them to the CLKBUS.

The Slo-Syn SP1800 Indexers used with the existing millbase positioning motors requires a -12 to -15 volt signal to turn on the motor, allowing the motor to run freely as long as this negative signal is applied. Figure B-9 shows the output to the SP1800 indexers. A high signal on FC6EN along with a high signal at position B4, B5, B6, or B7 will cause the MCT2 opto-isolator to turn off and a -15 volt level to appear on the appropriate output. The direction of x, y, z, or CS is determined by the output level of AND gate 3C13. Clockwise movement results when this output is at ground and counterclockwise movement results when this output is at +15 volts. The direction is determined by the levels of FC6EN and B0.

The thermocouple (TC) position indexer (Chapter 4) can also run freely when a low level is applied to the appropriate input. When FC6EN and B3 are both high signals the output of 3B13 is low and the TC motor will run freely. The direction of movement is determined by the output of 3A13

Figure B-10 illustrates the circuit required to accept the pulses from the indexers and connect them to the CLKBUS. The input from the SP1800 indexers arrive on board 3 via a Scotchflex connector and is fed into the comparator circuit (3D31). The comparator output goes into an opto-isolator whose output is used to trigger a one-shot. The one-shot output pulse is approximately 21 msec long to insure against false triggering due to the unstable nature of the SP1800 output. The one-shot output pulse is then tri-stated onto the CLKBUS if appropriate signals are present on FC6EN, FC7EN, and B3. The output of the TC indexer is TTL compatible and can be directly tri-stated onto the CLKBUS.

The remainder of Figure B-9 shows the circuit required to turn on the ultrasound generation circuits. The ultrasound generation circuits are being upgraded to include the new Amplifier Research model 1000L power amplifier, but during the interim the old amplifier may be used. A -15 volt level is required to turn on the old amp, while a +5 volt level is required for the new amp. Both of these signals are generated by the circuit shown in Figure B-9, and in this manner the interface can accommodate either amp.

5.3.6 Interface Output Circuits

The circuits used for outputting information back to the 7/32 or PDP-8 are shown in Figures B-11, B-12, B-13, and B-14. Bits B7, B6, B5, and B4 are latched when a function code 9 signal is present. This latched data is fed into a decoder that selects 1 of 16 lines to go low. This low signal is then routed to the appropriate tri-state gates connecting an 8-bit set of data onto the output bus. These data sets have been explained in the general description and only the check bits are expalined here. The check bits are shown in Figure B-14. These bits show the status of the counter, the motors and the sound circuits. A high signal on bit D7 will indicate that the counter is enabled. A high signal on bits D5 through D1 indicates the respective motor is on, while a high signal on D0 indicates the ultrasound is on. A high signal on D6 shows the counter is in the upcount mode, while a low signal indicates the downcount mode. The "check" bits can be useful for software implementation of control programs.

The output bus consists of bits D7 through D0 which are all routed to inverting buffers on board 5. The output of the first set of buffers are fed into a Scotchflex connector and on to the PDP-8. The second invertors are open collector with a 360 ohm pull up resistor used for sending the data to the 7/32. The output data is present at the PDP-8 and 7/32 at all times but is latched into a machine only under software control or the receipt of an appropriate handshake signal.

Chapter 6

Interface Software

6.1 Introduction

This chapter illustrates the software that has been developed to drive the mouse room interface. In presenting this software the author assumes the user has previous knowledge of the Perkin Elmer 7/32 operating system, the FORTRAN programming language and the CAL (common assembler language) programming language. The program listings are presented in Appendix D with the majority of comments and explanations appearing in these listings. A short introduction to each program is given in section 6.2.

These programs were developed to perform the initial tests of the interface. The overall philosophy used in developing this software was to divide the operations into subroutines and have one main calling program. In this manner the subroutines can be developed to perform basic interface operations and can be used by many different calling programs. The main program presented here (SVMAN.FTN) is used as an interface testing program, but could be expanded to the controlling program for the dosimetry experiment. Another approach would be to use this main program as a model and develop another program with a different command structure. In either case, the subroutines presented in section 6.2 are general purpose and can perform many useful functions with the interface.

To provide a quick reference the interface instruction set is given in Figure 6-1. This summary provides the hexadecimal number that must be written to the interface to perform the desired operation. The first number in each of these commands must be a "2" to select the correct device at the 7/32 (section 5.3.1).

6.2 Software

6.2.1 Main Testing Program

The main testing program (SVMAN.FTN). is presented in Figure D-1. This program uses an interactive command structure that prompts the user for the appropriate information. This program can initiate any of the interface operations such as moving the millbase, reading position encoders, and initiating an ultrasonic irradiation. It accomplishes this by an automatic sequencing of events, programmed by the user or by manual step by step operation, directed by the user.

	16 BIT HEX CODE	OPERATION
	21xx	load DC1
	22xx	load DC2
	23xx	load DC3
1376	2560	turn on sound with 100 Hz clock
	2550	turn on sound with 1 kHz clock
	2548	turn on sound with 10 kHz clock
	2544	turn on sound with 100 kHz clock
	2610	turn on TC motor
	2690	turn on TC motor opposite direction
1544	2608	turn on X motor
1672	2688	turn on X motor opposite direction
	2604	turn on Y motor
	2684	turn on Y motor opposite direction
	2602	turn on Z motor
	2682	turn on Z motor opposite direction
	2601	turn on CS motor
	2681	turn on CS motor opposite direction
	2710	count pulses from TC motor
	2708	count pulses from X motor
	2704	count pulses from Y motor
	2702	count pulses from Z motor
	2701	count pulses from CS motor
2176	2880	stop downcounter
	2900	read X position, HI bits
	2901	read X position, LO bits
	2902	read Y position, HI bits
	2903	read Y position, LO bits
	2904	read Z position, HI bits
	2905	read Z position, LO bits
	2906	read CS position, HI bits
	2907	read CS LO bits and X, Y, Z signs
	2908	read DATA bits from previous command
	2909	read the check bits
	290A	read DC1
	290B	read DC2
	290C	read DC3

Figure 6-1. Interface instruction set summary

6.2.2 Status Checking Subroutine

The subroutine "CHECK" is presented in Figure D-2. This subroutine checks the status of the interface counter. When the counter is enabled (performing a count operation), the subroutine loops back to check the status again. When the counter is disabled (not performing a count operation) the subroutine returns to the calling program.

6.2.3 Millbase and Thermocouple Moving Subroutine

The subroutine "MOVER" is presented in Figure D-3. This subroutine provides the commands required to move the millbase in the x, y, or z directions, move the thermocouple or move the capacitor (CS). This movement proceeds in an open-loop manner, without feedback from the position encoders and therefore the inaccuracy associated with the millbase gearing is not corrected.

6.2.4 Closed Loop Millbase Movement

The position encoders in the mouse room can be employed to yeild accurate millbase positioning. Figure D-4 presents the subroutine "BLASH" which provides closed loop control of the millbase. This subroutine reads the encoders and moves the millbase to the desired location within a preset accuracy.

6.2.5 Ultrasound Irradiation Subroutine

Figure D-5 presents the subroutine "SHOT" which can initiate an ultrasonic irradiation in the mouse room. The time duration of the shot can be set for any length.

6.2.6 Interface Output Subroutine

The interface has the capability of outputting information to the 7/32. Figure D-6 presents the subroutine "READR" which reads the 13 presently assigned bit groups from the interface into the 7/32. This subroutine stores this raw data into a buffer location but doesn't perform any interpretation of this data.

6.2.7 Position Encoder Interpretation Subroutine

The raw data returned from the subroutine "READR" needs further processing for correct interpretation. Figure D-7 presents the subroutine "BOUT" that converts the information contained in the buffer into useful information. The 6 data bit groups corresponding to the millbase position encoder read-outs are converted to signed integer numbers. These numbers correspond exactly with the Digi-point read-outs in the mouse room and can be used for closed loop control of the millbase.

Chapter 7

Recommendations for System Improvements

This chapter will present suggestions for improvement of the mouse stage, indexer and interface. Some of these improvements represent small circuit changes and can be easily implemented into the final construction.

7.1 Mouse Stage

Closed loop control of the thermocouple movement could be accomplished by adding a position encoder to the mouse stage. This encoder would give an accurate position read-out to the operator and to the computer. An inexpensive method of encoding this linear movement would be to use a linear potentiometer in conjunction with an accurate voltage source and A/D convertor. A method offering better accuracy would be to use a grated scale in combination with appropriate digital circuitry. With either of these methods digital read-outs could be made available for the operator and the data could be routed to the output bus of the interface enabling communication back to the 7/32.

7.2 Interface Circuits

Presently, the signals from the SP1800 indexers are all routed onto one line and brought into the interface. This signal pulses once for each step the corresponding motor takes. The three millbase motors run at approximately the same speed, but the CS motor can run at a much higher speed due to its decreased load. These signals are presently routed to a one-shot (Figure B-10) with an output pulse of 21 msec duration. To increase the CS motor speed this time duration would need to be decreased. This can be accomplished with the circuit shown in Figure 7-1. The shorter pulse duration is enabled through to the CLKBUS when the CS motor is running.

To fully implement the asynchronous data transfer capabilities of the 7/32, some handshake logic will have to be added to the interface. The input available (IA) handshake signal would need to be generated by the interface for transmission back to the DIO of the 7/32. One method for its generation would be to let the downcounter enable (DCEN) signal trigger a one-shot and use this pulse as the IA signal. By using this method the 7/32 DIO would be alerted everytime the downcounter finished an operation and could take appropriate action.

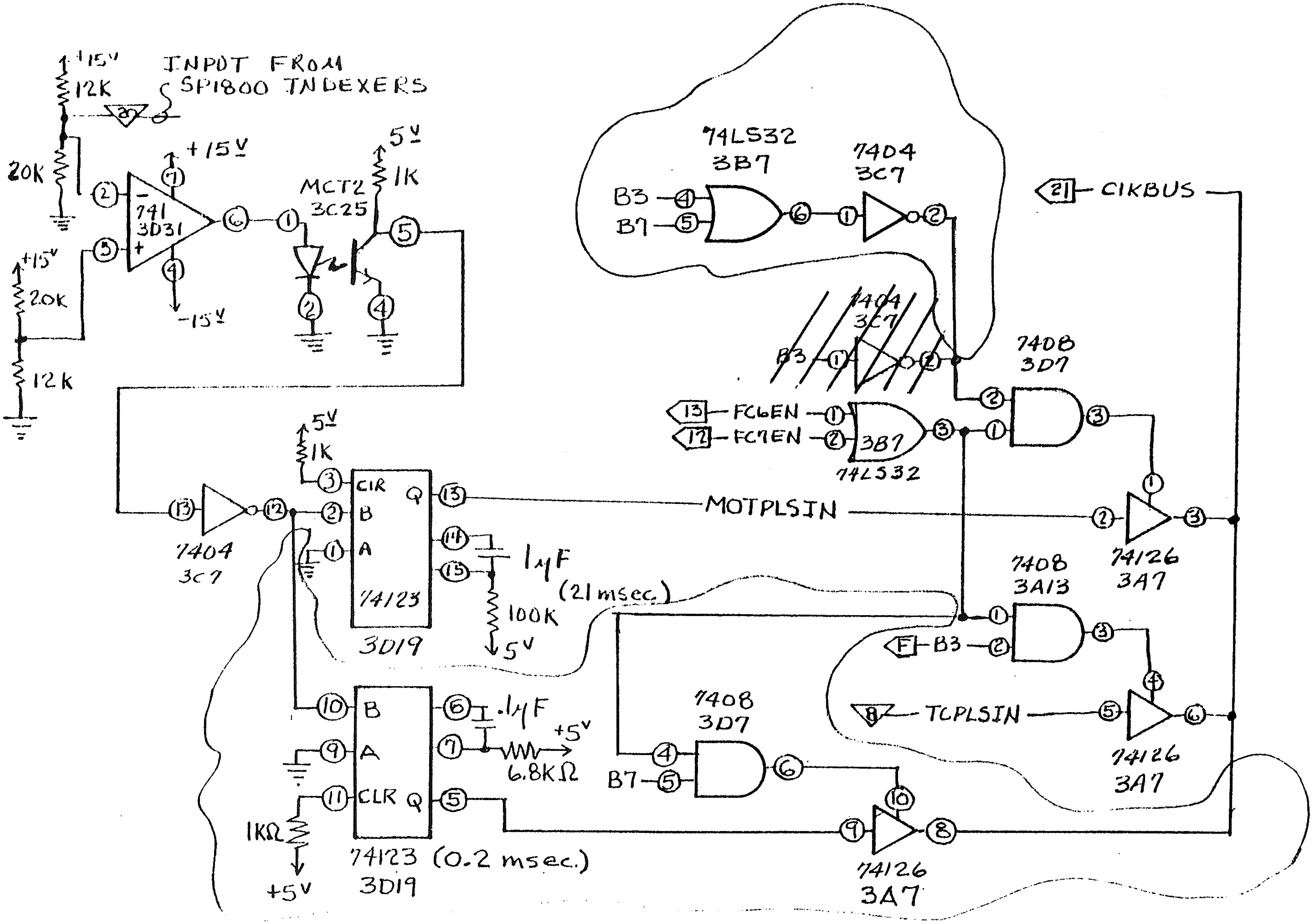


Figure 7-1. Improved Circuit for Routing of CS Pulse onto CLKBUS

REFERENCE

1. Chan, K. "A Data Acquisition System for Ultrasonic Tomography," M.S. Thesis, University of Illinois, 1980.

2. Duback, D. Master Thesis. Work in progress, University of Illinois, 1980.

3. Fry, R. B. and Fry, W. J. "Determination of Absolute Sound Levels and Acoustic Absorption Coefficients by Thermocouple Probes--Theory," J. Acoust. Soc. Am. 26, 294-310, 1954.

4. Goss, S. A., Cobb, J. W., and Frizzell. L. A. "Effect of Beam Width and Thermocouple Size on the Measurement of Ultrasonic Absorption Using the Thermoelectric Technique," Ultrasonics Symposium Proceedings IEEE cat. #77CH1264-1SU, 1977.

5. Grady, S. Master Thesis. Work in progress. University of Illinois, 1980.

6. O´Brien, W. D., Jr. "Ultrasound Dosimetry," in ULTRASOUND: ITS APPLICATIONS IN MEDICINE AND BIOLOGY, Vol. 3, ed. by F. J. Fry, 1973.

7. Perkin Elmer. "Digital Input/Output Controller Programming Manual," publication number 29-477, 1975.

8. Superior Electric Company. Manual on Slo-Syn Stepping Motors. Publication number SSM678.

9. Warner Electric Company. "Warner Miniature Ball Bearing Screws," publication number B-39.

Appendix A

Stepping Motor Indexer - Schematics

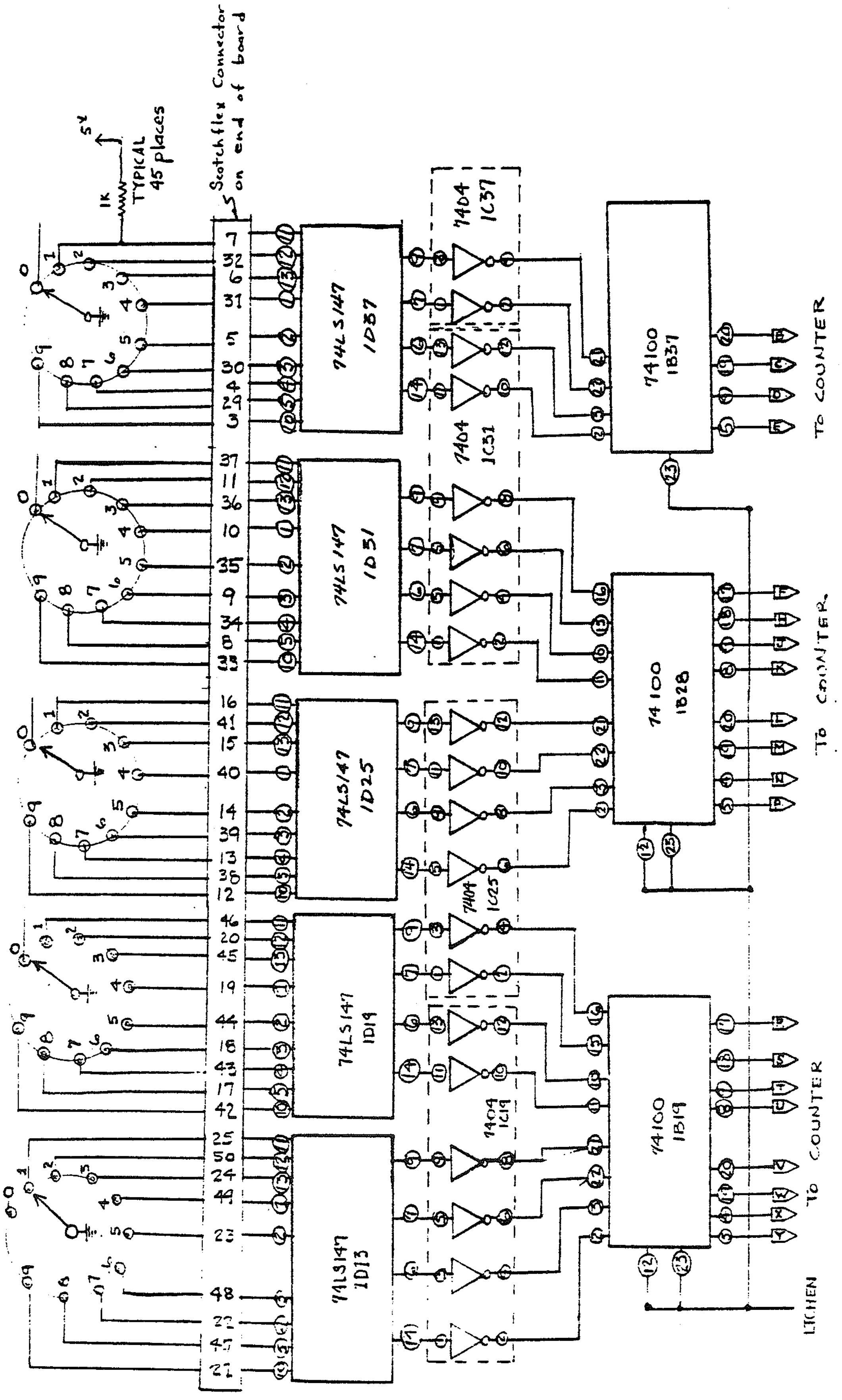


Figure A-1. Dial Circuits

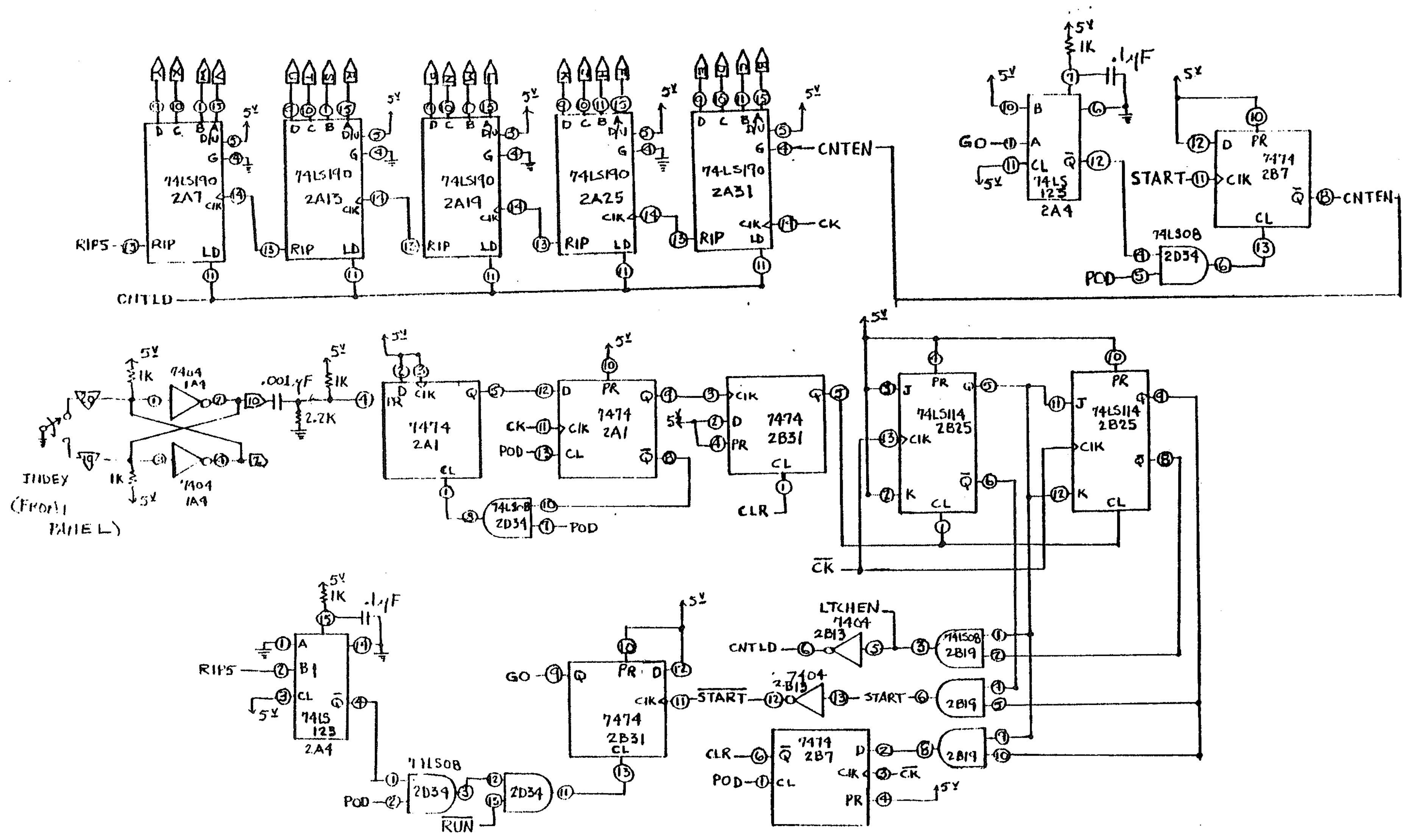


Figure A-2. Counter and Synchronizing Circuits

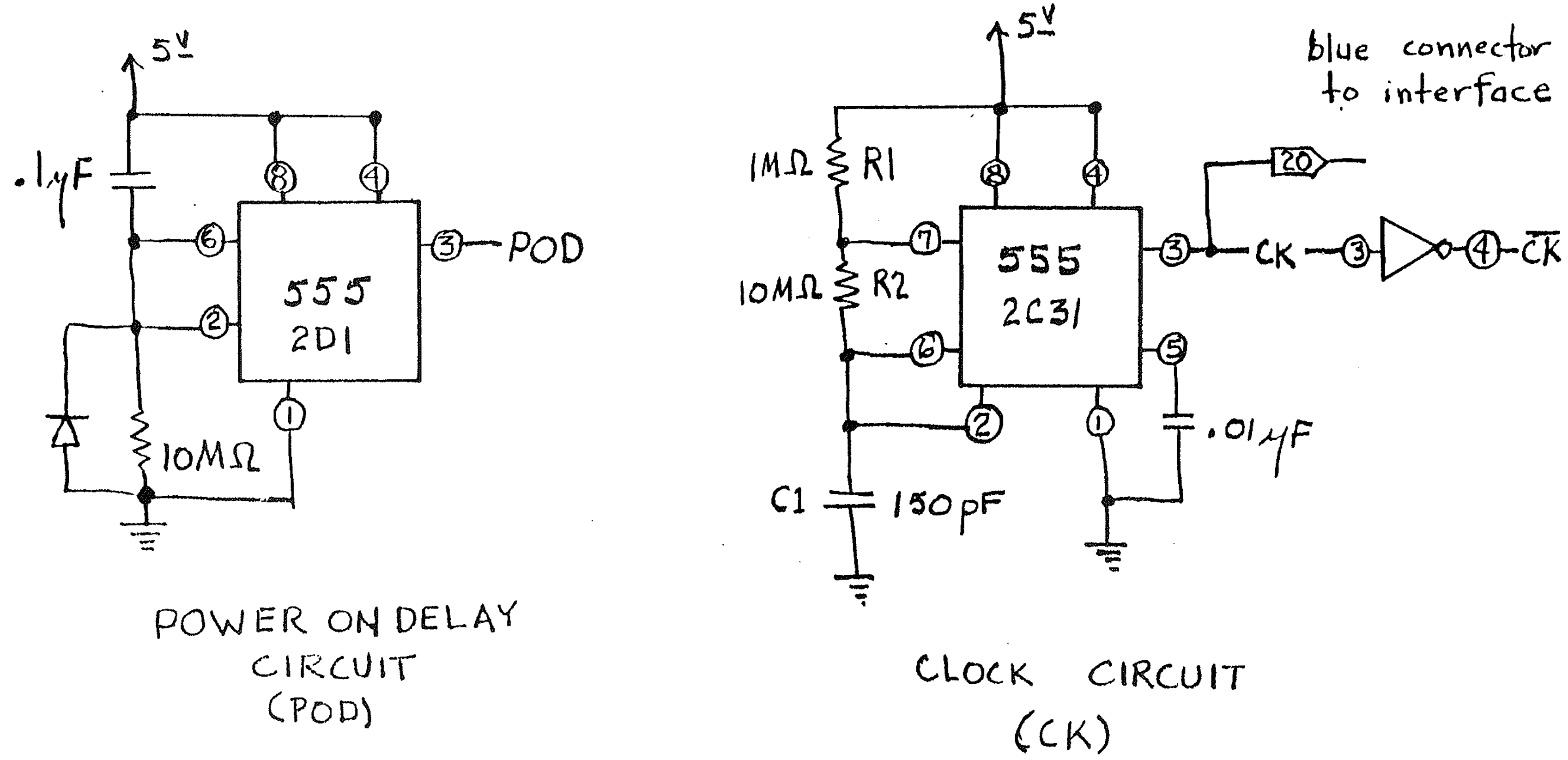
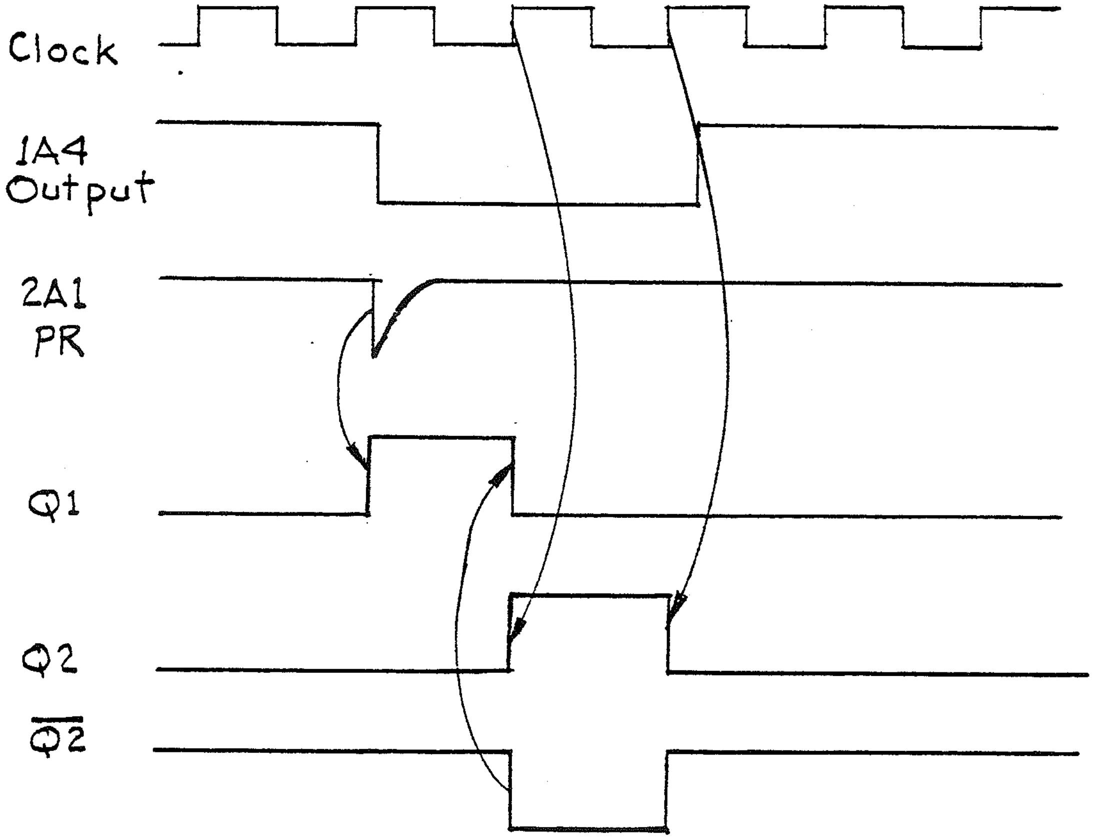


Figure A-3. Clock and POD Circuits



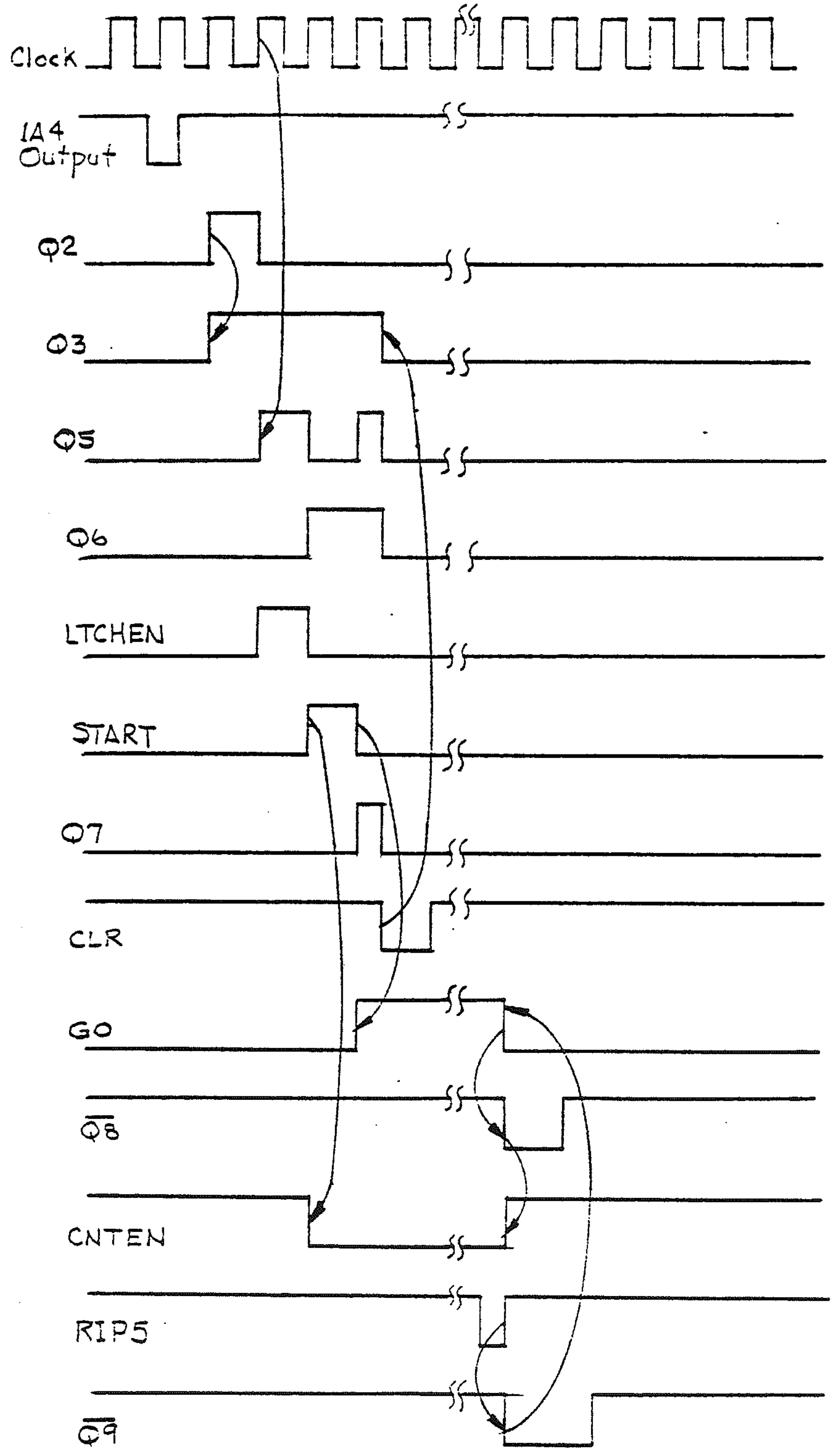
$Q2 = 2A1$ pin 9

$\overline{Q2} = 2A1$ pin 8

$Q1 = 2A1$ pin 5

$PR = 2A1$ pin 4

Figure A-4. "Index" Button Buffering--Timing Diagram



KEY: Q2 = 2A1 pin 9

Q3 = 2B31 pin 5

Q5 = 2B25 pin 5

Q6 = 2B25 pin 9

Q7 = 2B19 pin 5

$\overline{Q8}$ = 2A4 pin 12

$\overline{Q9}$ = 2A4 pin 4

Figure A-5. Synchronizing and Loading Circuit--Timing Diagram

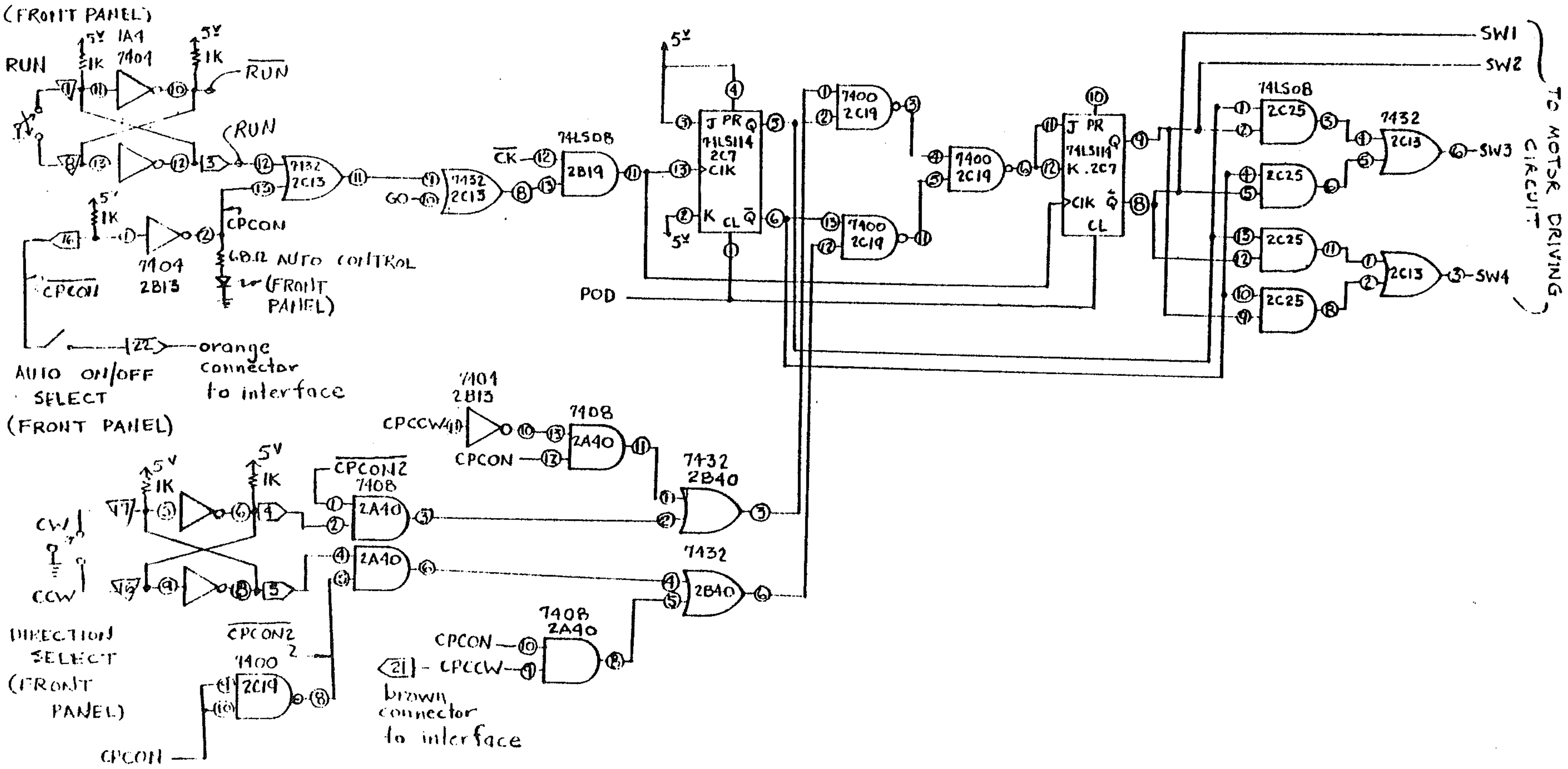
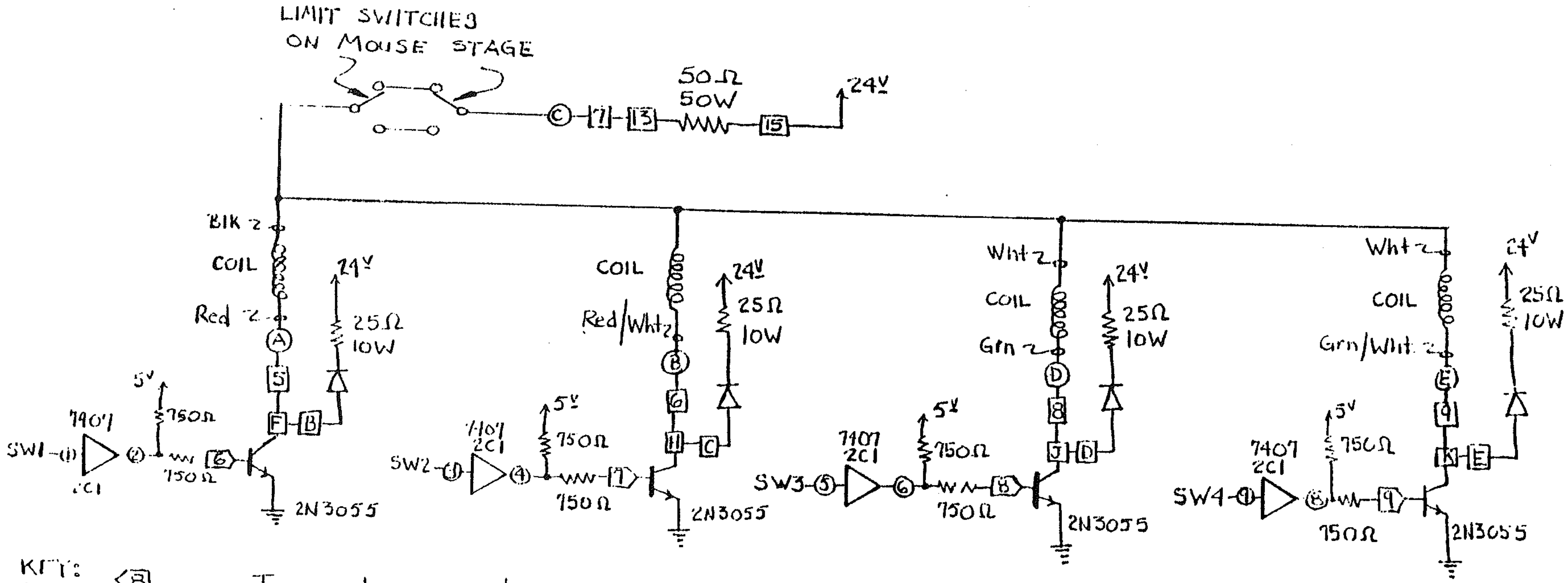


Figure A-6. Step Sequence Generator Circuit



KEY: <8 . . . Terminal connection on TRW edge connector in chassis box.

F, 5 } . . . Terminal connection on TRW edge connector in motor drive circuit.

Ⓐ . . . Terminal connection on Amphenol miniature rectangular connector on back of chassis.

Figure A-7. Motor Drive Circuits--Details

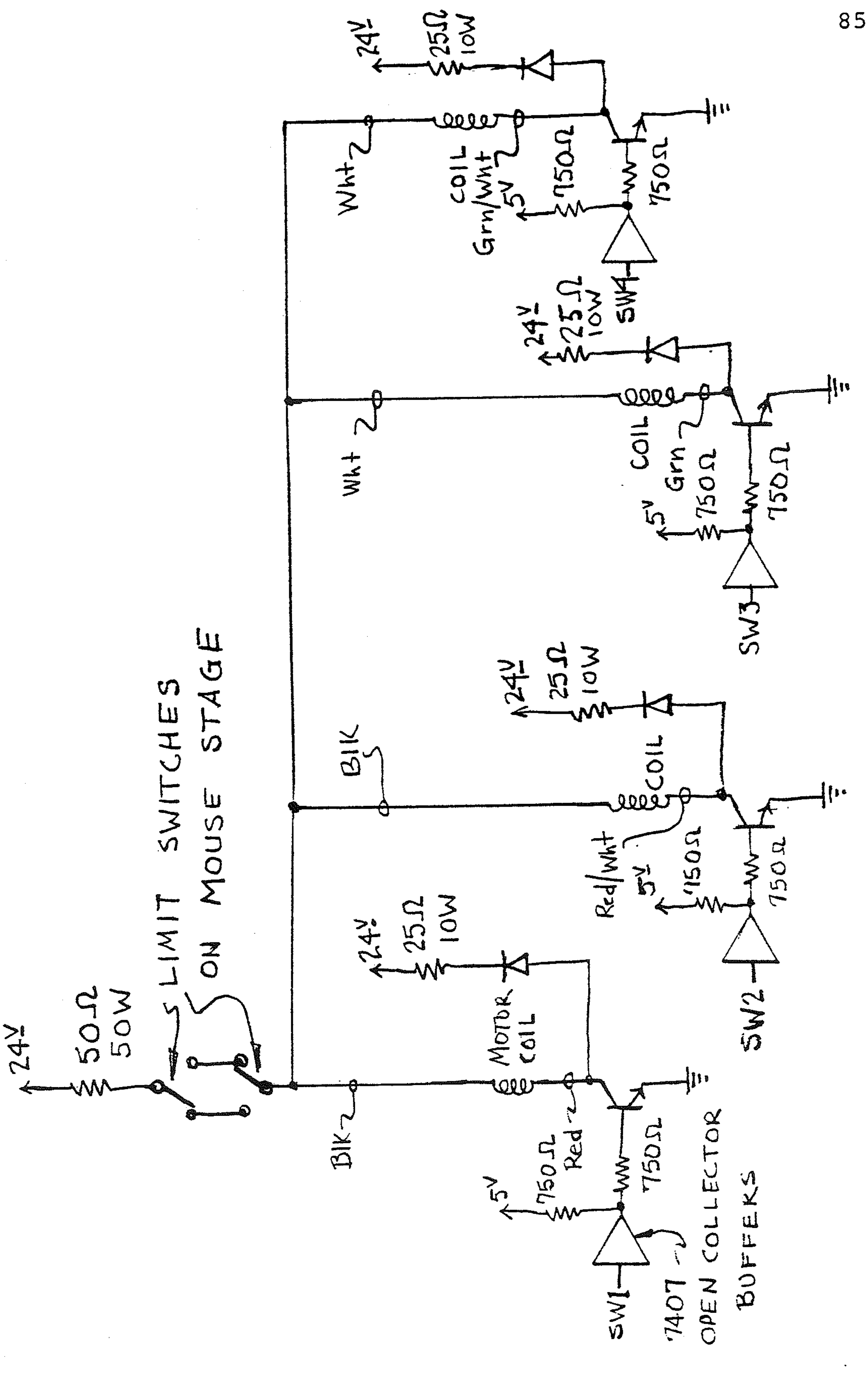


Figure A-8. Motor Drive Circuits--General

Appendix B

Computer Interface - Schematics

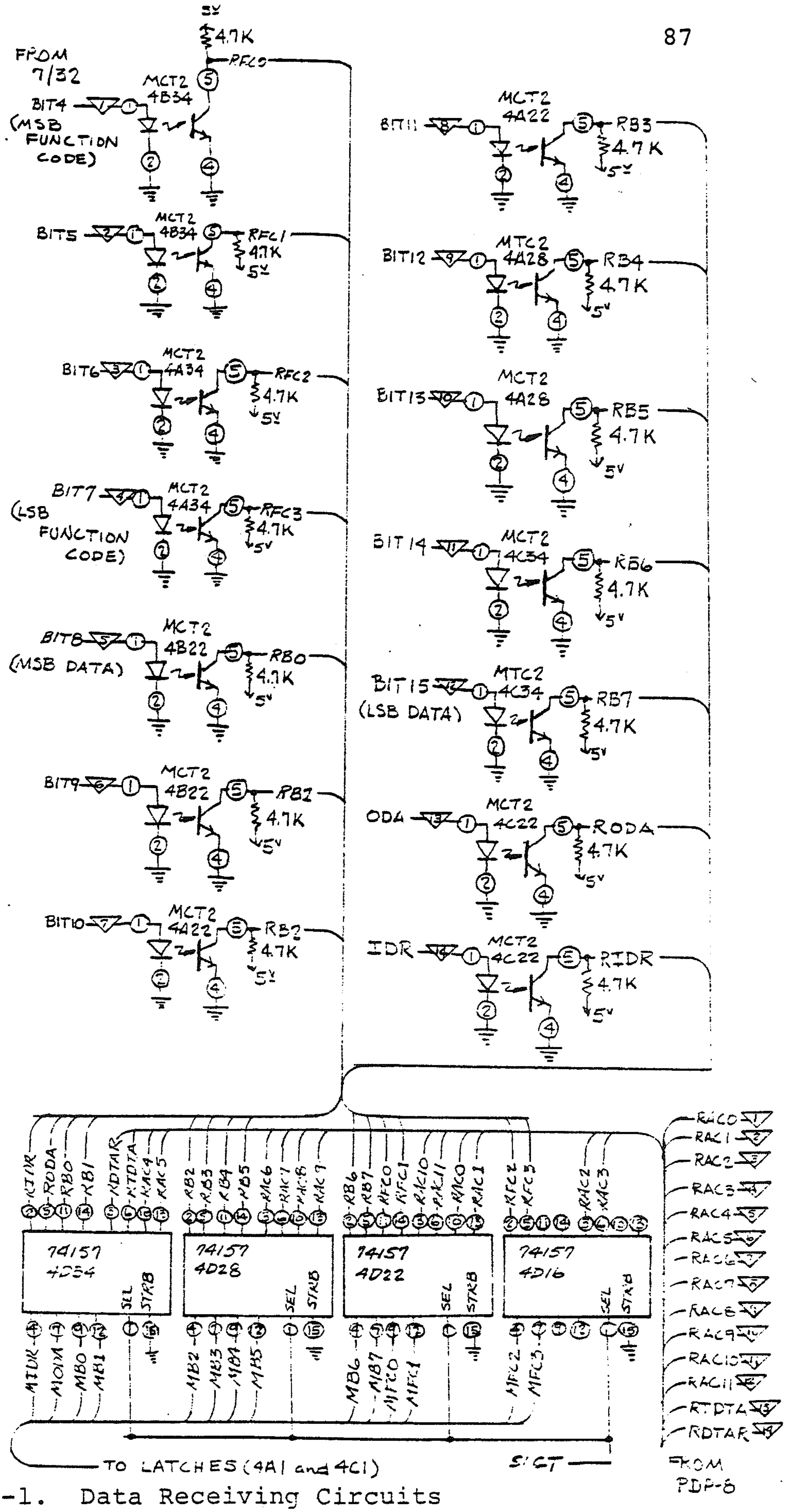


Figure B-1. Data Receiving Circuits

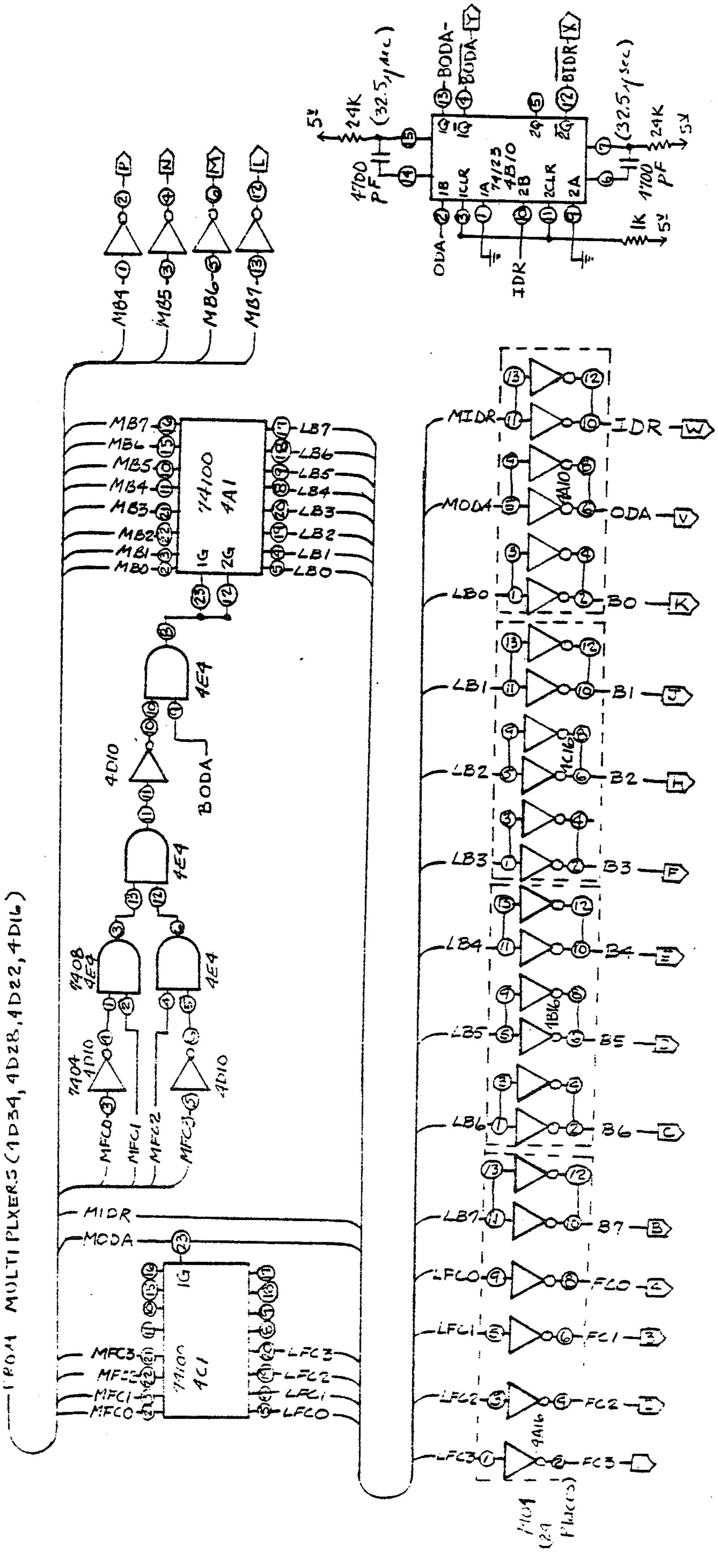


Figure B-2. Data Receiving Circuits -- Latches

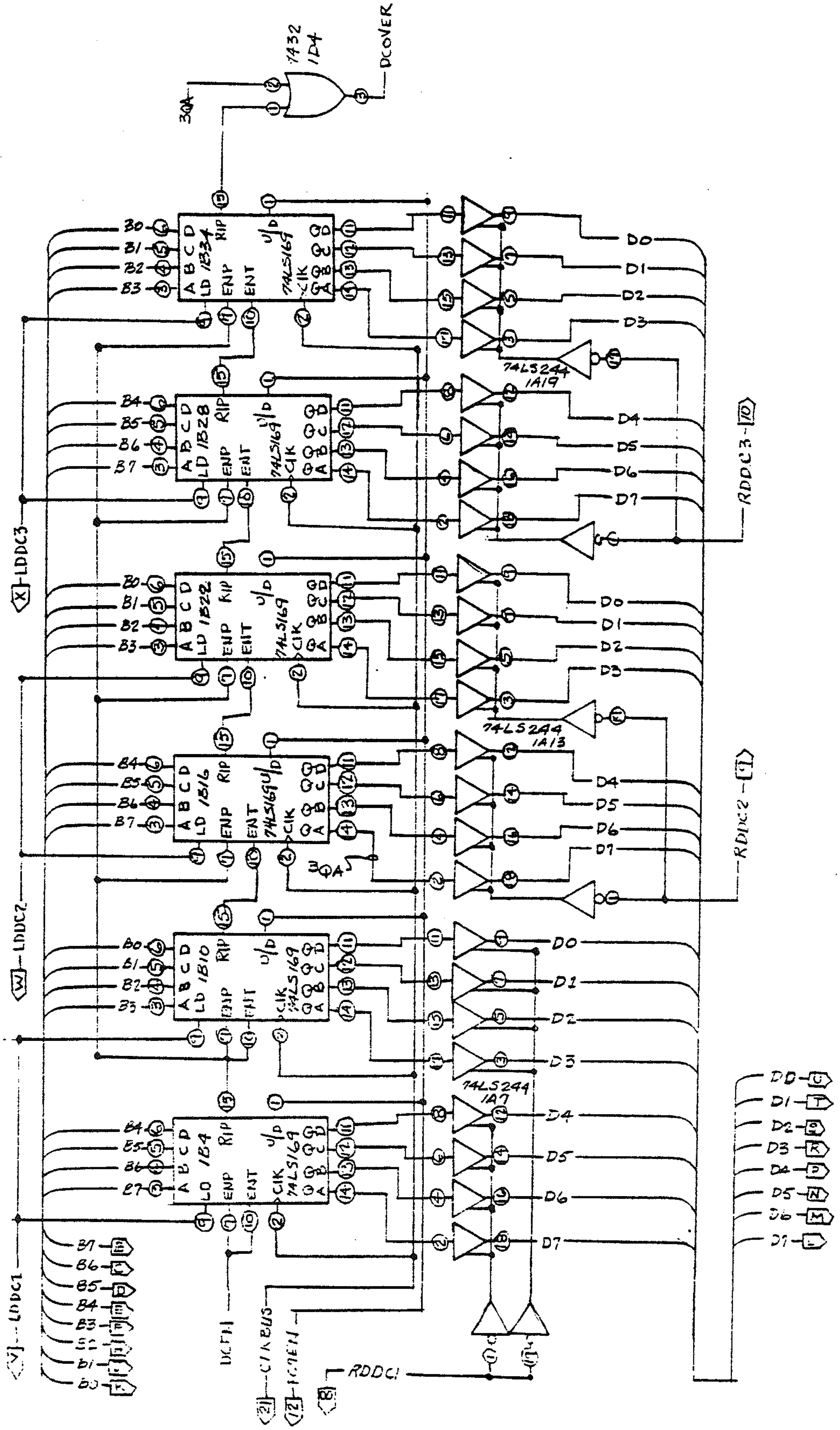


Figure B-3. Counter Circuit

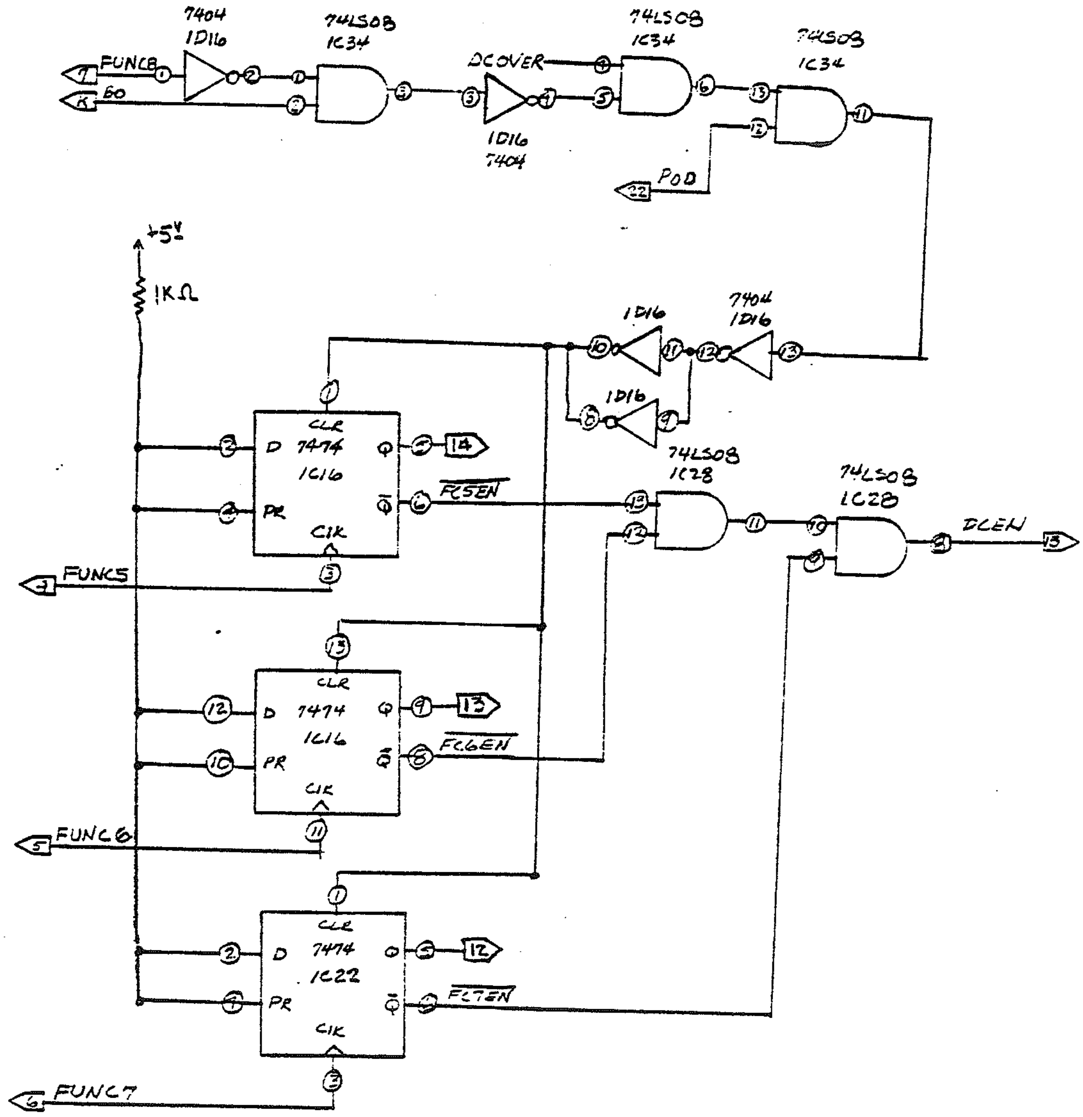


Figure B-4. Count Enable Circuit (DCEN)

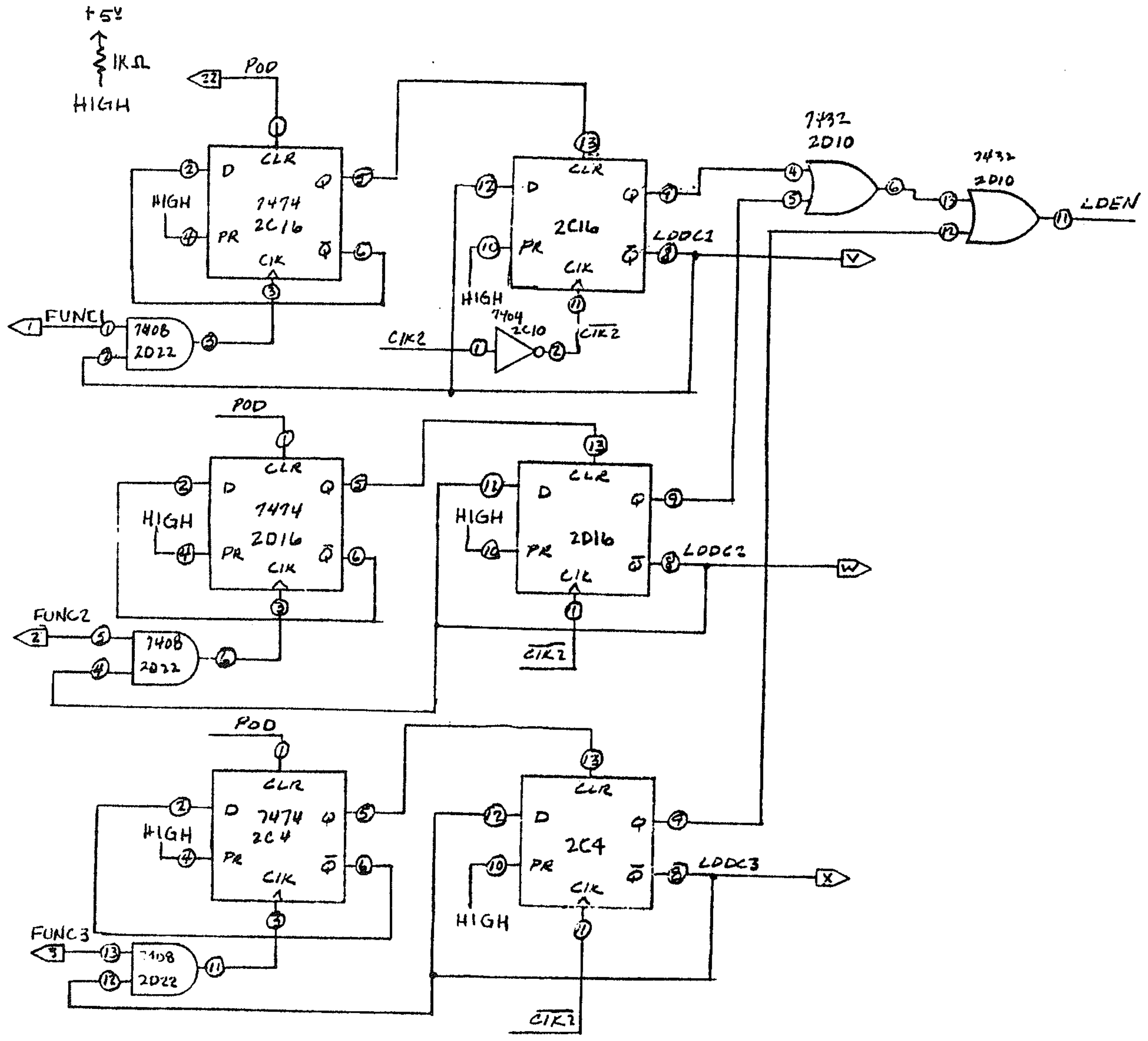
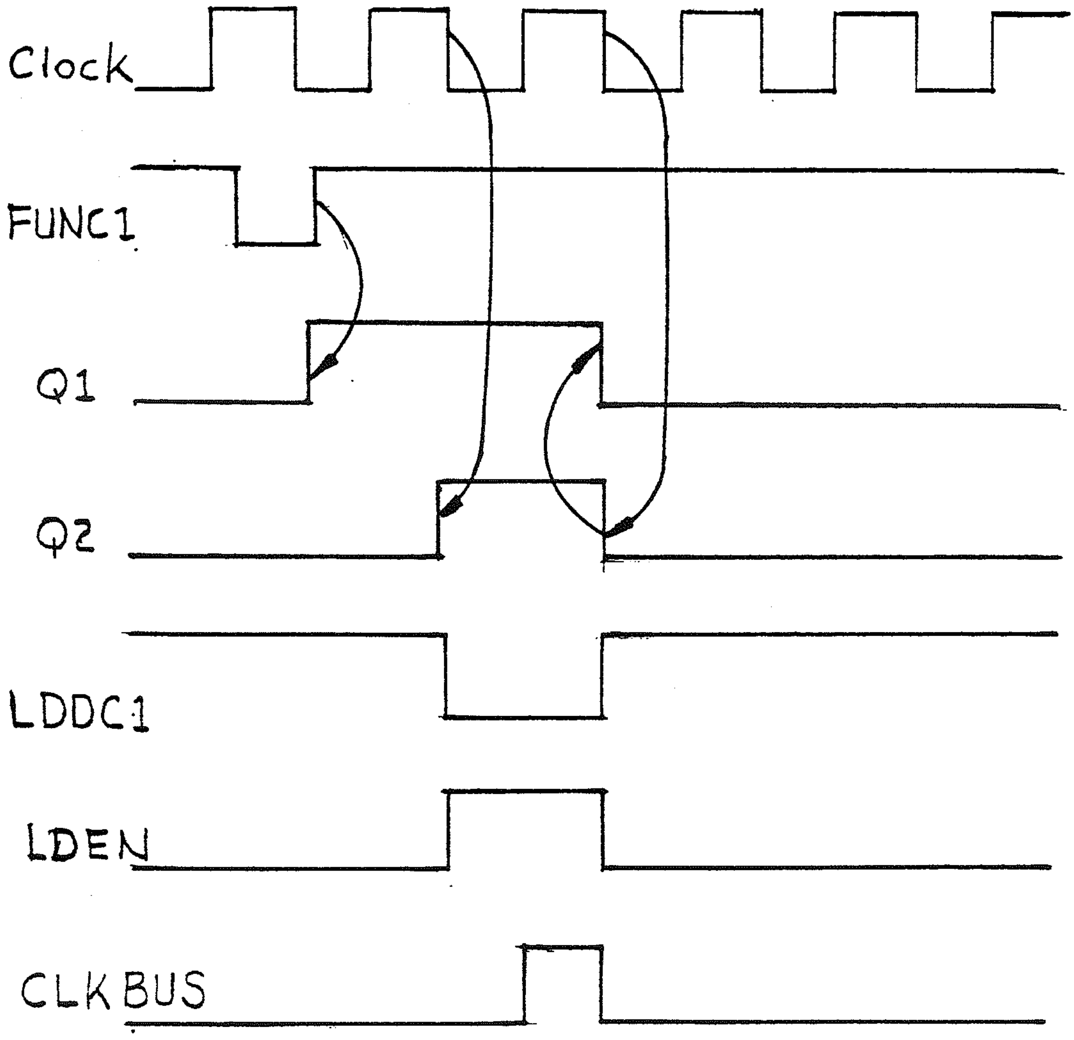
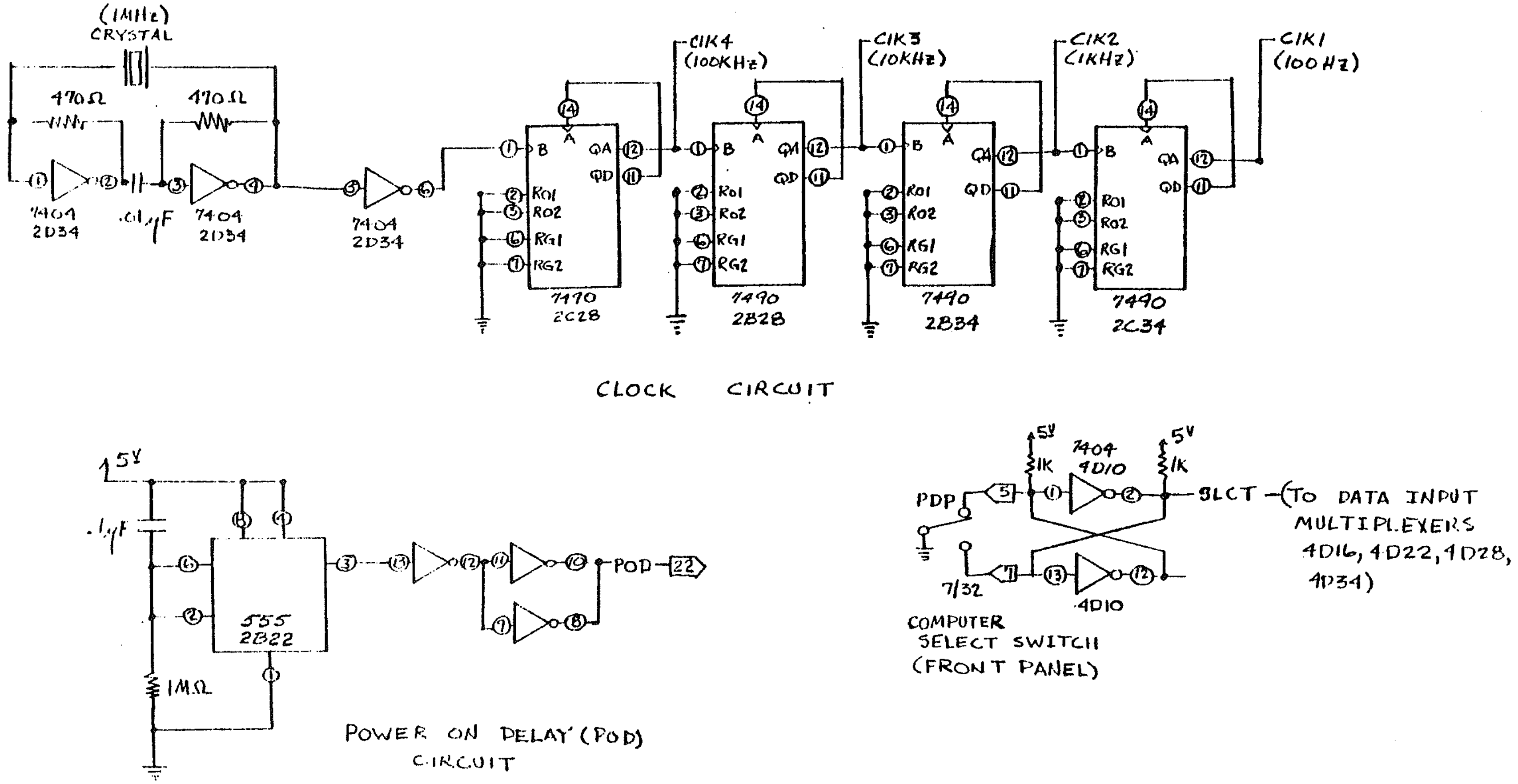


Figure B-5. Load Enable Circuit (LDEN)



KEY: Q1= 2C16 pin 5
Q2= 2C16 pin 9

Figure B-6. Load Enable--Timing Diagram



CLOCK CIRCUIT

POWER ON DELAY (POD) CIRCUIT

Figure B-7. Clock and POD Circuit

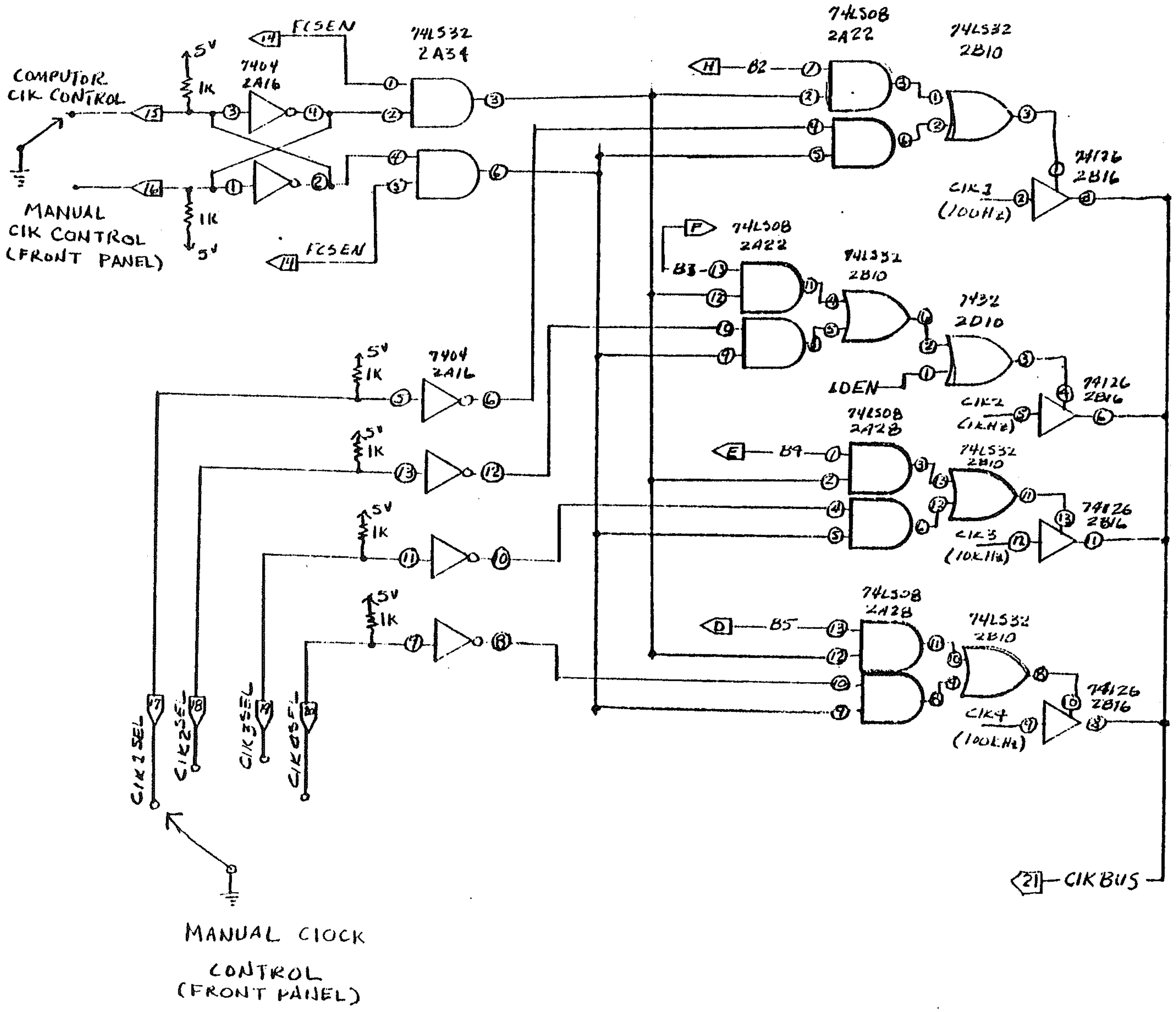


Figure B-8. CLKBUS Circuit

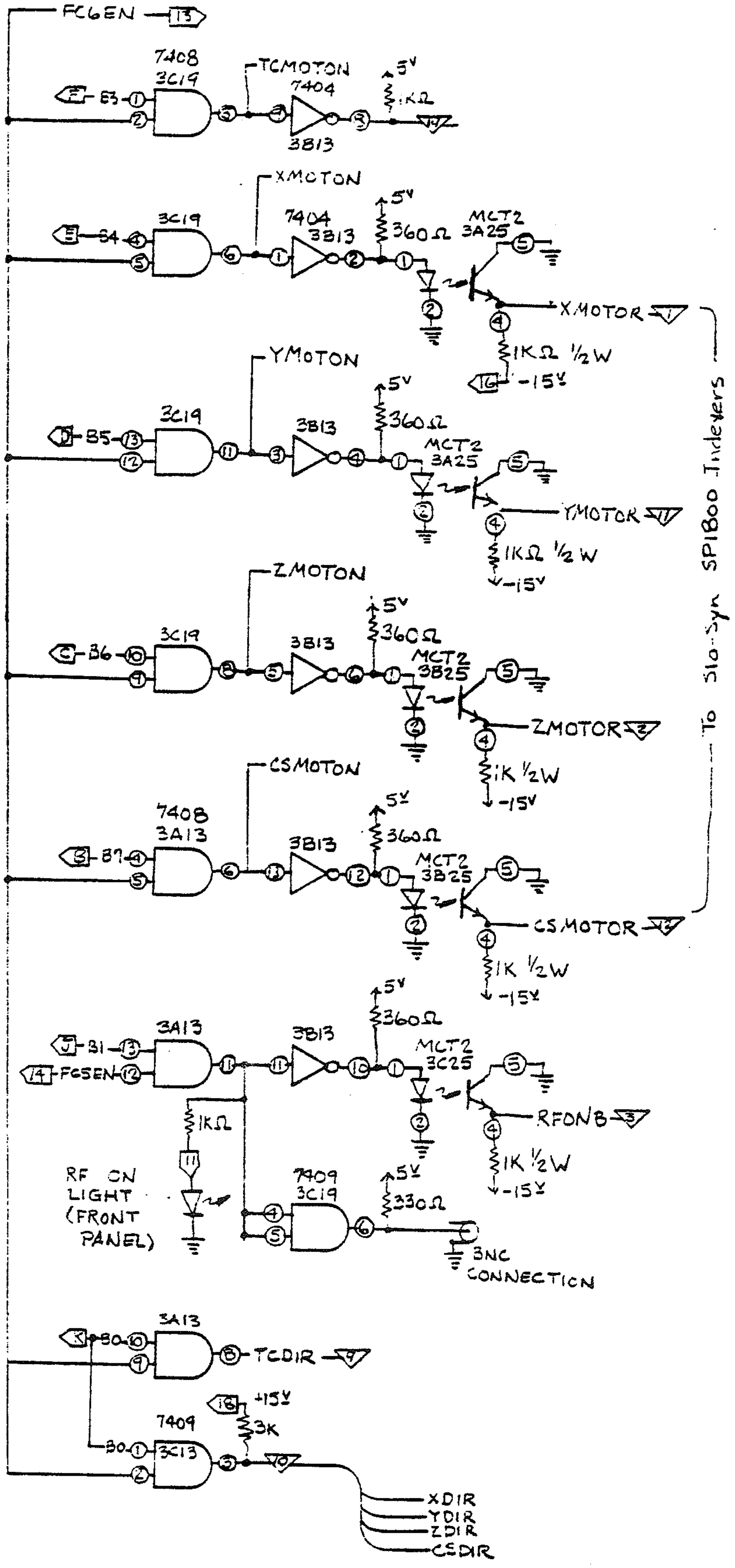


Figure B-9. Motor Indexer and RF Interface Circuits

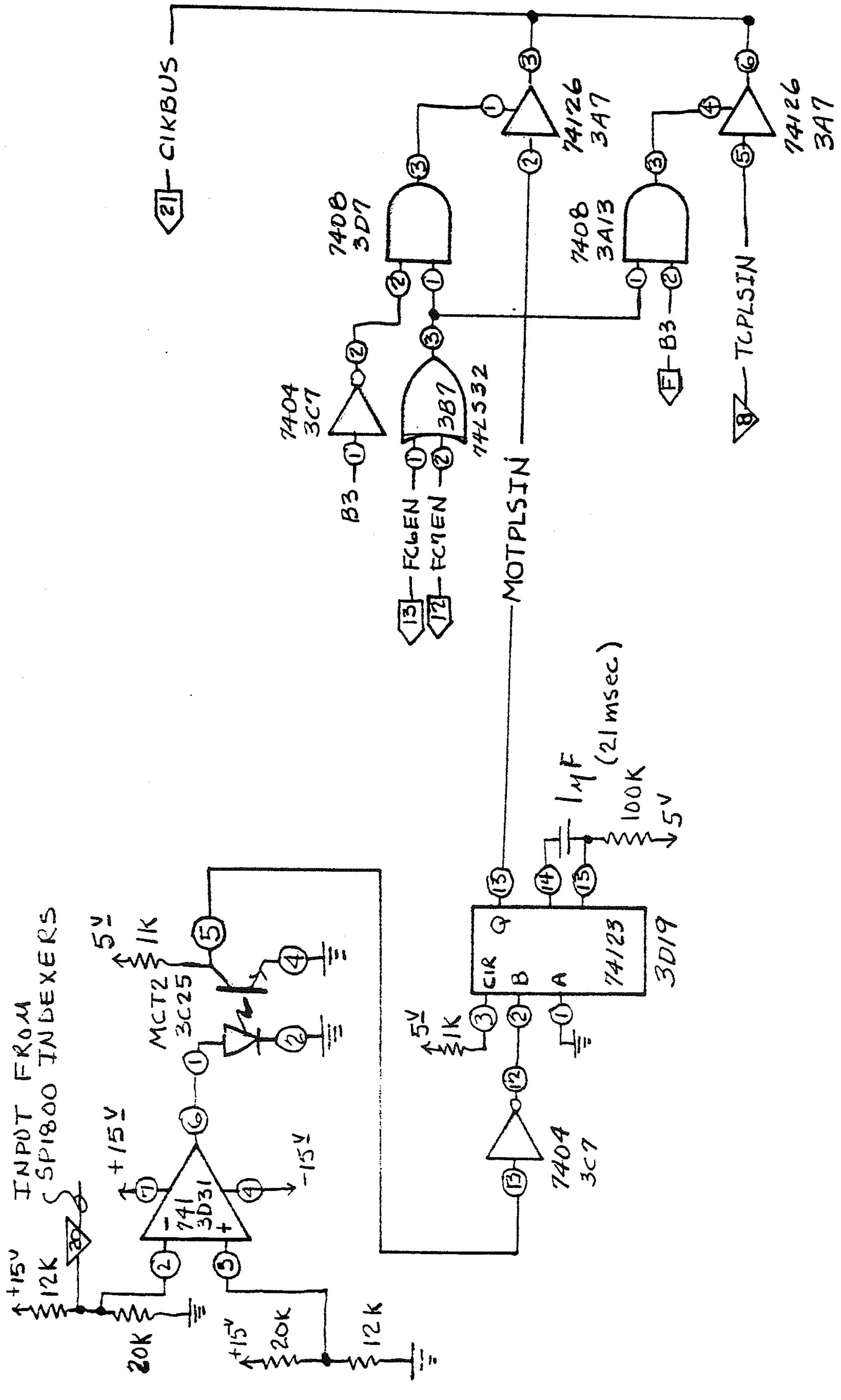
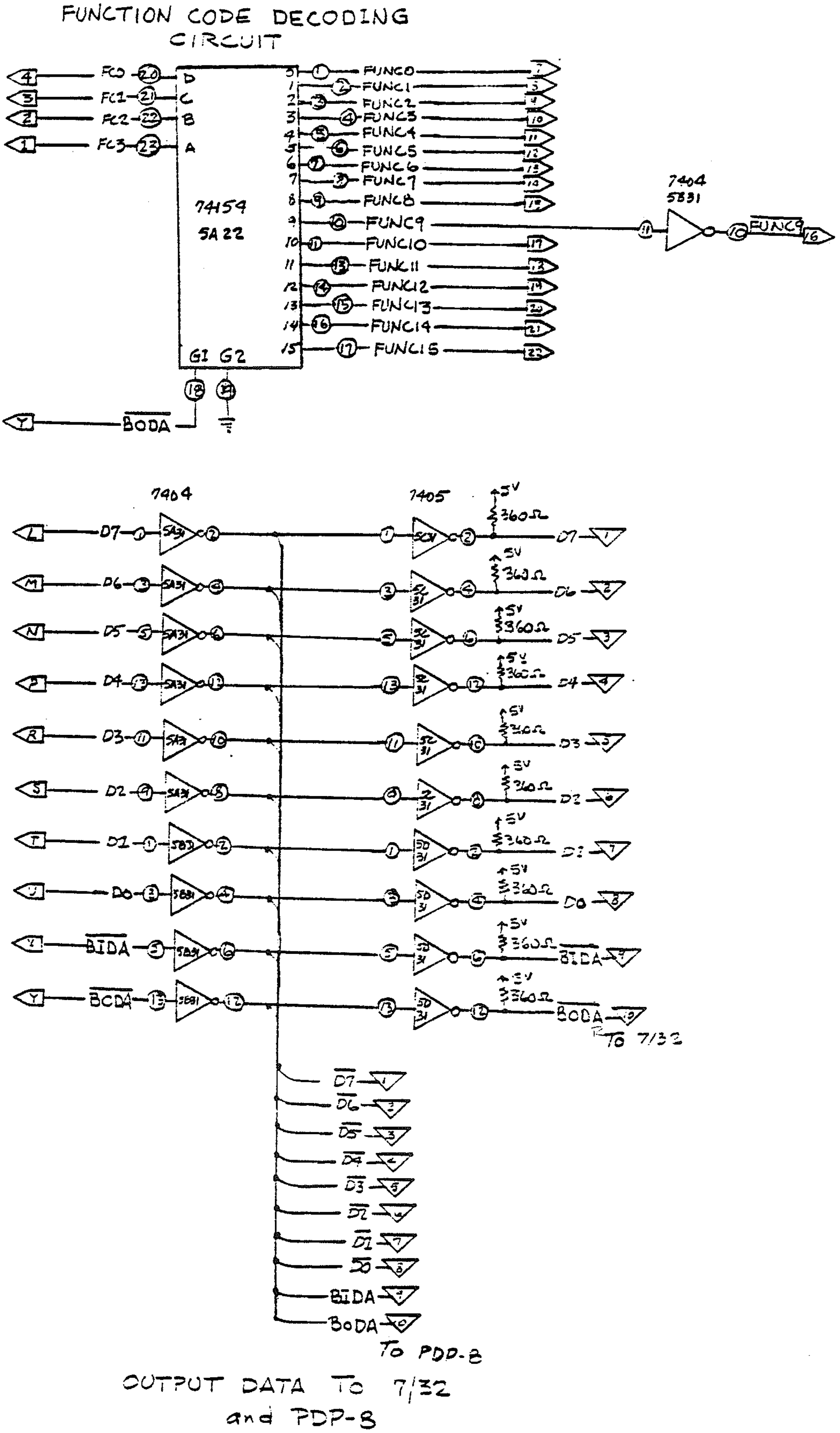


Figure B-10. Indexer Input Circuit


FUNCTION CODE DECODING CIRCUIT

OUTPUT DATA TO 7/32 and PDP-8

Figure B-11. Function Code Decoding and Output Buffers

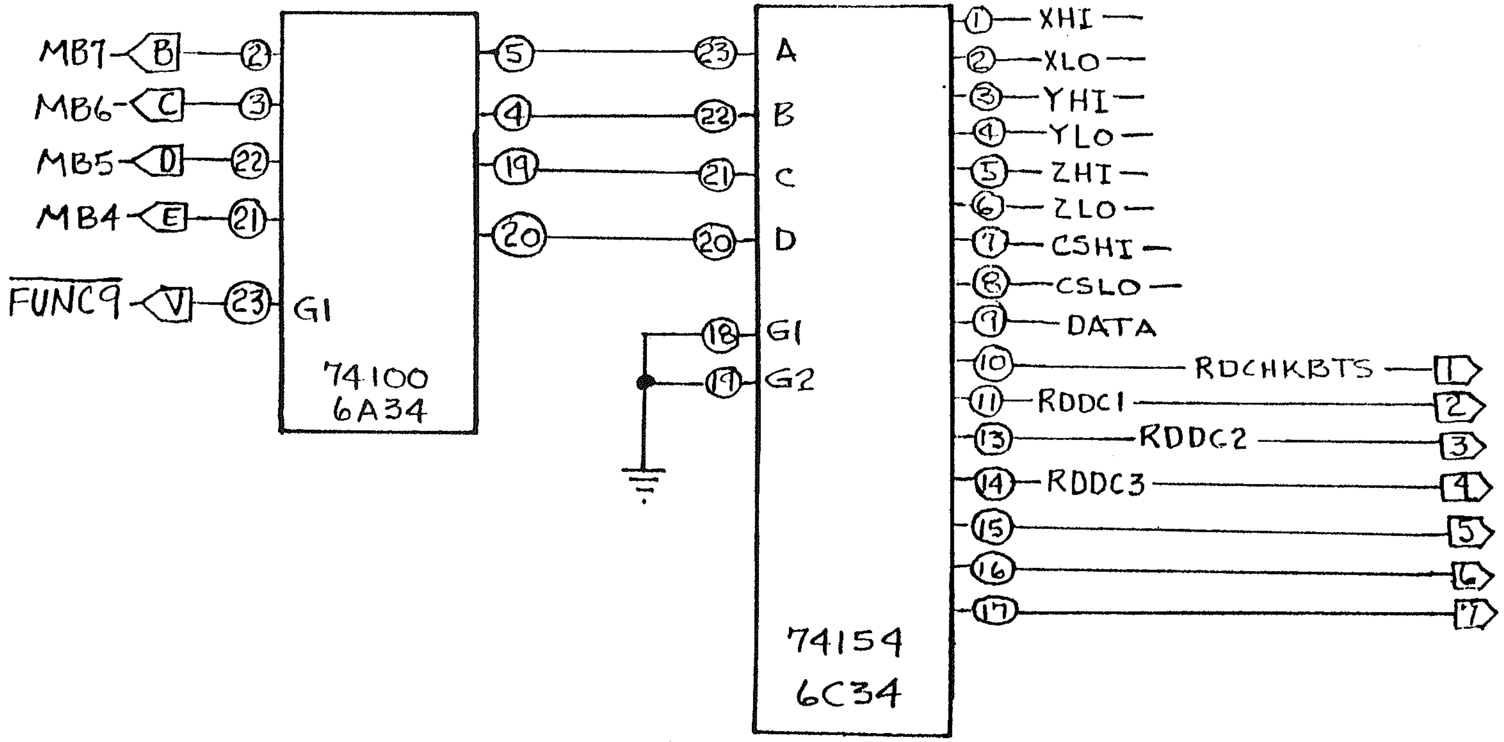


Figure B-12. Latch and Decoding Circuit for Interface Output

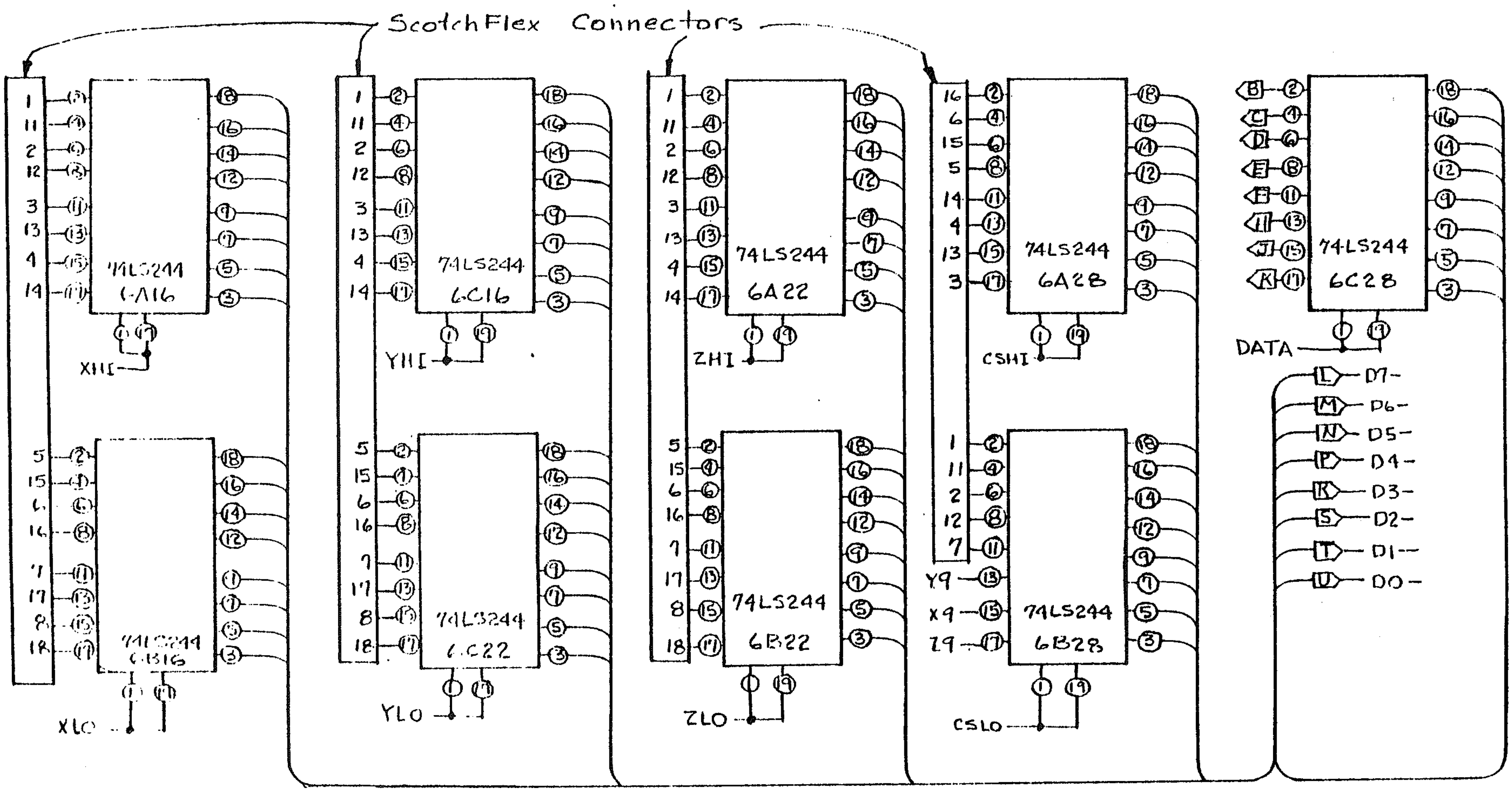


Figure B-13. Output Bus Circuits

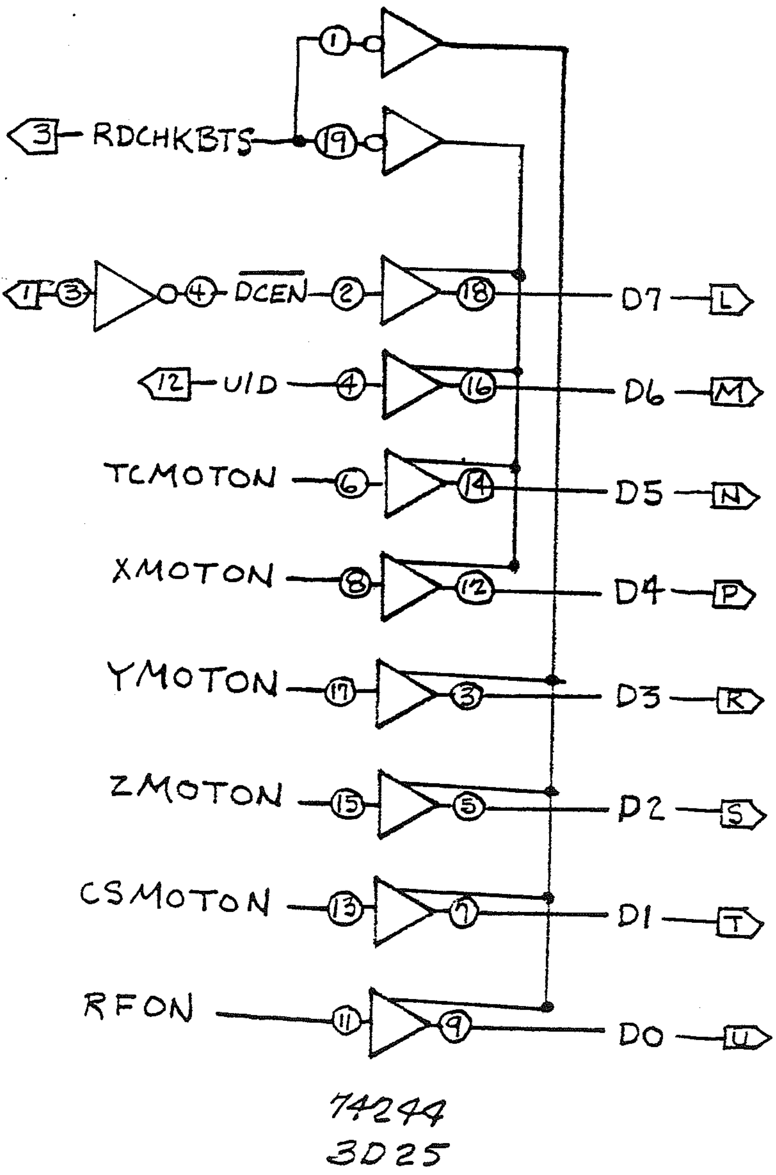


Figure B-14. Checkbits Circuit

Card 1 (Downcounter)				Card 2 (Timing)				Card 3 (Interfacing)			
1		A	+5 Volt	1	FUNC1	A	+5 Volt	1	DCEN	A	+5 Volt
2		B	B7 (LSB)	2	FUNC2	B		2		B	B7
3		C	B6	3	FUNC3	C		3	RDCHKBTS	C	B6
4	FUNC5	D	B5 Input	4		D	B5	4		D	B5
5	FUNC6	E	B4	5	SLCT	E	B4	5		E	B4
6	FUNC7	F	B3	6		F	B3	6		F	B3
7	FUNC8	H	B2	7	$\overline{\text{SLCT}}$	H	B2	7		H	
8	RDDC1	J	B1	8		J		8		J	B1
9	RDDC2	K	B0 (MSB)	9		K		9		K	B0
10	RDDC3	L	D7 (LSB)	10		L		10		L	D7
11		M	D6	11		M		11	RFON	M	D6
12	FC7EN	N	D5 Output	12		N		12	FC7EN	N	D5
13	FC6EN	P	D4	13		P		13	FC6EN	P	D4
14	FC5EN	R	D3	14	FC5EN	R		14	FC5EN	R	D3
15		S	D2	15	CCLKCONT	S		15		S	D2
16		T	D1	16	MCLKCONT	T		16	-15 Volt	T	D1
17		U	D0 (MSB)	17	CLK1SEL	U		17		U	D0
18		V	LDDC1	18	CLK2SEL	V	LDDC1	18	+15 Volt	V	
19		W	LDDC2	19	CLK3SEL	W	LDDC2	19		W	
20		X	LDDC3	20	CLK4SEL	X	LDDC3	20		X	
21	CLKBUS	Y		21	CLKBUS	Y		21	CLKBUS	Y	
22	POD	Z	GROUND	22	POD	Z	GROUND	22		Z	GROUND

Figure B-15(a). Edge Connector Connection Schedule

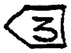
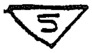
Card 4 (From 7/32 and PDP-8)				Card 5 (To 7/32 and PDP-8)				Card 6 (Read Function)			
1	FC3	A	+5 Volt	1	FC3	A	+5 Volt	1	RDCHKBTS	A	+5 Volt
2	FC2	B	B7	2	FC2	B		2	RDDC1	B	MB7
3	FC1	C	B6	3	FC1	C		3	RDDC2	C	MB6
4	FC0	D	B5	4	FC0	D		4	RDDC3	D	MB5
5	SLCT	E	B4	5		E		5		E	MB4
6		F	B3	6		F		6		F	
7	$\overline{\text{SLCT}}$	H	B2	7	FUNC0	H		7		H	
8		J	B1	8	FUNC1	J		8		J	
9		K	B0	9	FUNC2	K		9		K	
10		L	MB7	10	FUNC3	L	D7	10		L	
11		M	MB6	11	FUNC4	M	D6	11		M	
12		N	MB5	12	FUNC5	N	D5	12		N	
13		P	MB4	13	FUNC6	P	D4	13		P	
14		R		14	FUNC7	R	D3	14		R	
15		S		15	FUNC8	S	D2	15		S	
16		T		16	$\overline{\text{FUNC9}}$	T	D1	16		T	
17		U		17	FUNC10	U	D0	17		U	
18		V	ODA	18	FUNC11	V		18		V	$\overline{\text{FUNC9}}$
19		W	IDR	19	FUNC12	W	GROUND	19		W	
20		X	$\overline{\text{BIDR}}$	20	FUNC13	X	$\overline{\text{BIDA}}$	20		X	
21		Y	$\overline{\text{BODA}}$	21	FUNC14	Y	$\overline{\text{BODA}}$	21		Y	
22		Z	GROUND	22	FUNC15	Z		22		Z	

Figure B-15(b). Edge Connector Connection Schedule

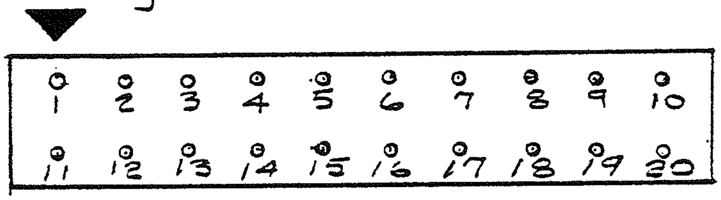
Appendix C

Key to Schematics

Key to Schematics:

Symbol	Definition
	Terminal connection on TRW edge connector at rear of chassis.
	Terminal connection on Scotchflex socket connector.
②	Pin connection on integrated circuit device.

Numbering for Scotchfex socket connectors



1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20

Integrated circuit location numbering

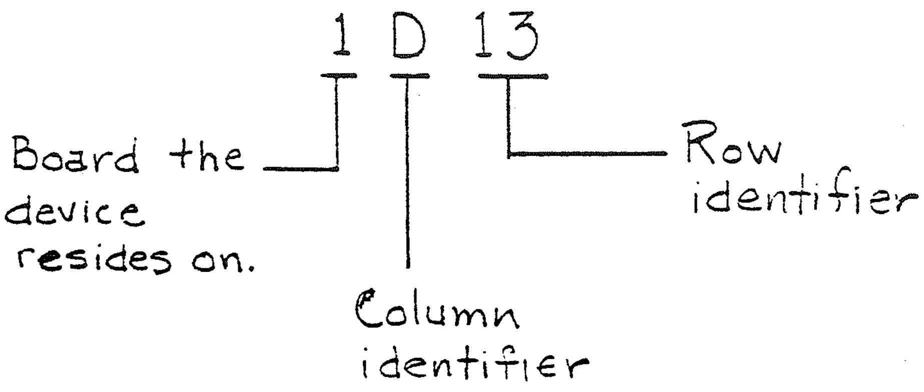


Figure C-1. Key to Schematic Drawings

Appendix D

Computer Program Listings

```
      INTEGER CMAND,A(26),SHTFLG,ALSEQ(50),SEQ,REP,RGSTG(16)
      INTEGER XC,YC,ZC,TC,CC,BUF(13),UNIT,DELAY,STATUS
      REAL DIST,TIME,NUMSEQ(50),FINAL
      INTEGER*2 LOAD,OUT5,BUFOUT(16),XSIGN,YSIGN,ZSIGN
      INTEGER NNPTS,ISHOT,CHAN,SHEND
      INTEGER*2 ADBUF(6000),AVBUF(6000)
      INTEGER DCHI,DCMID,DCLO
      REAL LENGTH,TLNGTH
      INTEGER M2
      COMMON/MOV/A(26),RGSTG(16)/COUNT/XC,YC,ZC,TC,CC
      COMMON/ATD/ADBUF(6000),AVBUF(6000)
      A(1)=Y'41202020'
      DO 2052 I=2,26
2052  A(I)=A(I-1)+Y'01000000'
      XC=0
      YC=0
      ZC=0
      TC=0
      CC=0
$ASSM
         STM    0,RGSTG
         LHI    1,X'A8'
         LHI    2,X'A9'
         LHI    3,X'C0'
         OCR    1,3
         OCR    2,3
         LH     3,K
         NHI    3,X'0FFF'
         OHI    3,X'2000'
         WHR    1,3
         RHR    2,3
         LM     0,RGSTG
$FORT
      NNPTS=6000
      CHAN=2
8     WRITE(1,3000)
3000  FORMAT(' ','AUTO OR MANUAL SEQUENCING?(A OR M)')
      READ(1,5003)CMAND
      IF(CMAND.EQ.A(1))GO TO 5014
      IF(CMAND.EQ.A(13))GO TO 2013
      IF(CMAND.EQ.A(5)) GO TO 5
      GO TO 8
5014  WRITE(1,5012)
5012  FORMAT(' ','SHOULD WE CALL UP A PREVIOUSLY STORED SEQUENCE?
     6(Y OR N)')
      READ(1,5003)CMAND
      IF(CMAND.EQ.A(25)) GO TO 4014
5010  WRITE(1,5000)
5000  FORMAT(' ','STEPS IN SEQUENCE?(I3)')
```

Figure D-1(a). Main Test Program -- SVMAN.FTN

```
      READ(1,5001)SEQ
5001  FORMAT(I3)
      DO 4000 I=1,SEQ
      WRITE(1,5002)I
5002  FORMAT(' ',2X,'STEP 'I3,2X,'DEVICE?(A1)')
      READ(1,5003)ALSEQ(I)
5003  FORMAT(A1)
      WRITE(1,5004)
5004  FORMAT(' ','DISTANCE/TIME?(F9.4)')
      READ(1,5005)NUMSEQ(I)
5005  FORMAT(F9.4)
4000  CONTINUE
      WRITE(1,5020)
5020  FORMAT(' ','NUMBER OF TIMES THIS SEQ. GETS REPEATED?(I3)')
      READ(1,5001)REP
4001  WRITE(1,5006)
5006  FORMAT('0','TYPE G TO PROCEED WITH SEQUENCING')
      READ(1,5003)CMAND
      IF(CMAND.NE.A(7)) GO TO 4001
      DO 4013 K=1,REP
      DO 4012 I=1,SEQ
      IF(ALSEQ(I).EQ.A(20)) GO TO 4002
      IF(ALSEQ(I).EQ.A(24)) GO TO 4003
      IF(ALSEQ(I).EQ.A(25)) GO TO 4004
      IF(ALSEQ(I).EQ.A(26)) GO TO 4005
      IF(ALSEQ(I).EQ.A(18)) GO TO 4006
      WRITE(1,5013)
5013  FORMAT(' ','ERROR IN SEQUENCE, START OVER')
      GO TO 5014
4002  KEY=A(20)
4009  DIST=NUMSEQ(I)
      CALL MOVER(KEY,DIST)
4401  CALL CHECK
      DELAY=1
      UNIT=2
      CALL WAIT(DELAY,UNIT,STATUS)
      WRITE(2,7000)STATUS,UNIT,DELAY
7000  FORMAT(' ',2X,'STATUS= ',I2,2X,'UNIT= ',I2,2X,'DELAY= ',I2)
      GO TO 4012
4400  DIST=NUMSEQ(I)
      CALL BLASH(KEY,DIST)
      GO TO 4401
4003  KEY=A(24)
      GO TO 4400
4004  KEY=A(25)
      GO TO 4400
4005  KEY=A(26)
      GO TO 4400
4006  KEY=A(18)
```

Figure D-1(b). Main Test Program -- SVMAN.FTN

```
      TIME=NUMSEQ(I)
      CALL SHOT(TIME)
      CALL CHECK
      DELAY=5
      UNIT=2
      CALL WAIT(DELAY,UNIT,STATUS)
      WRITE(2,7000)STATUS,UNIT,DELAY
4012  CONTINUE
4013  CONTINUE
      WRITE(1,4020)
4020  FORMAT(´ ´,´--------SEQUENCE COMPLETE----------´)
      GO TO 8
4014  READ(3,5001)SEQ
      DO 4015 I=1,SEQ
      READ(3,5003)ALSEQ(I)
      READ(3,5005)NUMSEQ(I)
4015  CONTINUE
      READ(3,5001)REP
      GO TO 4001
C
C
C
C-----------MANUAL MOVING LOADING----------
C
C
C
2013  WRITE(1,2000)
2000  FORMAT(´ ´,´SELECT X,Y,Z,C,T,R(SETS UP TIME OF SHOT)´/
     6´ ´,´M(MANUAL MOVE),A(TAKE SHOT),E(END),H(HELP)
     6,P(STOP MOVE),L(READ INTO BUFFER),I,J,K´)
2050  WRITE(1,2002)
2002  FORMAT(´ ´,´SELECT?´)
      READ(1,5003)CMAND
      DO 1 I=1,26
      IF(CMAND.EQ.A(I)) GO TO(2014,1,2005,1,5,1,1,8,9,9,9,12,2006,
     61,1,16,1,2004,19,2003,1,1,1,2003,2003,2003),I
1     CONTINUE
      GO TO 2050
C
C
C          COMMAND = X,Y,Z,OR T
C
C
2003  WRITE(1,2020)
2020  FORMAT(´ ´,´DISTANCE? ENTER +OR-XX.XX MM´)
      READ(1,2008)DIST
2008  FORMAT(F8.3)
      IF(CMAND.EQ.A(20)) GO TO 7010
      CALL BLASH(CMAND,DIST)
```

Figure D-1(c). Main Test Program -- SVMAN.FTN

```
      GO TO 7011
7010  CALL MOVER(CMAND,DIST)
7011  IF(SHTFLG.EQ.1) GO TO 2014
      GO TO 2013
C
C
C
C              COMMAND= R(SET DURATION OF SHOT)
C
C
C
2004  WRITE(1,2010)
2010  FORMAT(' ','DURATION OF SHOT? ENTER XX.XXXX SEC')
      READ(1,2011)TIME
2011  FORMAT(F8.4)
      WRITE(1,2012)
2012  FORMAT(' ','TAKE SHOT AFTER EVERY MOVE?(Y OR N)')
      READ(1,5003)CMAND
      IF(CMAND.EQ.A(25)) SHTFLG=1
      IF(CMAND.EQ.A(14)) SHTFLG=0
      GO TO 2013
C
C
C
C          COMMAND= C(CS)
C
C
C
2005  WRITE(1,2015)
2015  FORMAT(' ','FOR CS ENTER NO. OF STEPS +OR- XXX.')
      READ(1,2008)DIST
      CALL MOVER(CMAND,DIST)
      GO TO 2013
C
C
C
C       COMMAND = M(MANUAL)
C
C
C
2006  WRITE(1,2017)
2017  FORMAT(' ','FOR MANUAL LOAD, ENTER IN HEX; XXXX')
      READ(1,2018)LOAD
2018  FORMAT(Z4)
2021  WRITE(2,1001)LOAD
1001  FORMAT('0',10X,'LOAD= ',Z4)
$ASSM
        STM   0,RGSTG
        LHI   1,X'A8'
```

Figure D-1(d). Main Test Program -- SVMAN.FTN

```
          LHI   2,X'A9'
          LHI   3,X'C0'
          LHI   8,X'0800'
          OCR   1,3
          OCR   2,3
          LH    3,LOAD
          LR    4,3
          NHI   4,X'0FFF'
          OHI   4,X'2000'
          WHR   1,4
OSENS     SSR   1,10
          BTC   X'8',OAGAN
          B     READ
OAGAN     SIS   8,1
          BZ    READ
          B     OSENS
READ      RHR   2,5
          STH   5,OUT5
          LM    0,RGSTG
$FORT
      WRITE(2,108)OUT5
108   FORMAT('0',5X,Z8)
      GO TO 2013
C
C
C
C
C      COMMAND=A(TAKE A SHOT)
C
C
C
2014  CALL CHECK
      CALL SHOT(TIME)
      GO TO 2013
16    LOAD=X'2880'
      GO TO 2021
12    CALL READR(BUF)
      CALL BOUT(BUF,BUFOUT,XSIGN,YSIGN,ZSIGN)
      DO 7001 I=1,13
      JI=I-1
7001  WRITE(2,7002)JI,BUF(I),BUFOUT(I),BUFOUT(I)
7002  FORMAT(' ',2X,I2,2X,Z16,2X,Z16,2X,I8)
      GO TO 2013
9     WRITE(1,7012)
7012  FORMAT(' ','FINAL POSITION? ENTER +OR-X.XXXX')
      READ(1,7013)FINAL
7013  FORMAT(F8.5)
      CALL READR(BUF)
      XSIGN=0
```

Figure D-1(e). Main Test Program -- SVMAN.FTN

```
        YSIGN=0
        ZSIGN=0
        CALL BOUT(BUF,BUFOUT,XSIGN,YSIGN,ZSIGN)
        DO 7014 II=1,13
        JII=II-1
  7014  WRITE(2,7002)JII,BUF(II),BUFOUT(II),BUFOUT(II)
        IF(CMAND.EQ.A(9))I=1
        IF(CMAND.EQ.A(10))I=2
        IF(CMAND.EQ.A(11))I=3
        WRITE(2,7015)I
  7015  FORMAT(' ',2X,'I= ',I3)
        DIST=(FINAL-BUFOUT(I)/1000.)*25.4
        CALL BLASH(CMAND,DIST)
        GO TO 7011
  19    IF(TIME.LE. 0.0) GO TO 1900
        ISHOT=1
        CALL ATD(ADBUF,TIME,NNPTS,ISHOT,CHAN,SHEND)
        CALL ANORM(ADBUF,NNPTS)
        GO TO 2013
  1900  WRITE(1,1901)
  1901  FORMAT('0',2X,'NO TIME VALUE SPECIFIED FOR SHOT LENGTH!!!')
        GO TO 2013
  5     CONTINUE
        STOP
        END
```

Figure D-1(f). Main Test Program -- SVMAN.FTN

```
C
C
C--------SUBROUTINE CHECK
C
C
C          This subroutine checks the status of the
C     downcounter.  The subroutine enters a loop until
C     the downcounter is disabled.  No arguements are
C     passed into or out of the subroutine.
C
C
      SUBROUTINE CHECK
      COMMON/MOV/A(26),RGSTG(16)
      INTEGER RGSTG(16),A(26)
$ASSM
         STM   0,RGSTG
         LHI   1,X'A8'
         LHI   2,X'A9'
         LHI   3,X'C0'
*
*     Load the command to read the check bits.
*
         LHI   6,X'2909'
         OCR   1,3
         OCR   2,3
         WHR   1,6
*
*     Write the command "2909" to the interface and enter
*     a delay loop.
*
         LHI   11,X'0FFF'
ONE      SIS   11,1
         BNZS  ONE
*
*     Read the check bits into reg. 5 and shift the
*     least significant bit into the carry position.
*     Repeat this sequence until the carry bit is set.
*
LUP      RHR   2,5
         SRHLS 5,1
         BNCS  LUP
         LM    0,RGSTG
$FORT
      RETURN
      END
```

Figure D-2. Status Checking Subroutine -- CHECK

```
C
C
C--------SUBROUTINE MOVER-------
C
C
C          This subroutine will supply the commands to the
C     interface facilitating a movement of the millbase
C     in the X,Y, or Z directions.  It will also move the
C     thermocouple position and the capacitor(CS) value.
C     The arguements passed into this subroutine are
C     explained below:
C
C               KEY=INTEGER*4 (full word) variable containing
C                   the ASCII code for a command that determines
C                   which device will be moved.  The commands
C                   are:
C                        "X"= move in X direction
C                        "Y"= move in Y direction
C                        "Z"= move in Z direction
C                        "T"= move thermocouple
C                        "C"= move capacitor(CS)
C
C                DIST= REAL variable that contains the
C                      distance of number of steps to
C                      be moved.  The distance should be
C                      given in millimeters for the X,Y,
C                      and Z directions as well as the
C                      thermocouple movement.  The appropiate
C                      scaling is done by this subroutine.
C                      For capacitor movement DIST should
C                      contain the number of steps requested.
C
C
      SUBROUTINE MOVER(KEY,DIST)
      COMMON/MOV/A(26),RGSTG(16)/COUNT/XC,YC,ZC,TC,CC
      INTEGER*2 MOVR,XMOVE,YMOVE,ZMOVE,CMOVE,TMOVE,OUT1,OUT2,OUT3
     6,OUT4,OUT5
      INTEGER*2 PMFL,K
      INTEGER RGSTG(16),KEY,XC,ZC,YC,TC,CC,IDIST
     6,A(26)
      REAL RDIST,SCALE,DIST
      DATA XMOVE/X'2608'/,YMOVE/X'2604'/,ZMOVE/X'2602'/
      DATA TMOVE/X'2610'/,CMOVE/X'2601'/
      A(1)=Y'41202020'
C
C     Determine which command is contained in KEY and transfer
C     to the right location for scaling the distance.
C
C     The correct hexadecimal command is entered into the
```

Figure D-3(a). Millbase and Thermocouple Moving Subroutine — MOVER

```
C     variable MOVR for loading out to the interface.
C
      DO 2052 I=2,26
2052  A(I)=A(I-1)+Y'01000000'
      DO 210 I=1,26
200   IF(KEY.EQ.A(I))GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,
     616,17,18,19,20,21,22,23,24,25,26),I
210   CONTINUE
      GO TO 23
24    XC=XC+1
      MOVR=XMOVE
      IF(XC.EQ.3)GO TO 202
205   SCALE=39.0
      GO TO 201
202   XC=0
206   SCALE=40.0
      GO TO 201
25    YC=YC+1
      MOVR=YMOVE
      IF(YC.EQ.3)GO TO 203
      GO TO 205
203   YC=0
      GO TO 206
26    ZC=ZC+1
      MOVR=ZMOVE
      IF(ZC.EQ.3)GO TO 204
      GO TO 205
204   ZC=0
      GO TO 206
3     CC=CC+1
      MOVR=CMOVE
      SCALE=1.0
      GO TO 201
20    TC=TC+1
      MOVR=TMOVE
      IF(TC.EQ.2)GO TO 207
      SCALE=157.0
      GO TO 201
207   SCALE=158.0
      TC=0
201   WRITE(2,1000)DIST
1000  FORMAT('0',10X,'DIST=  ',F8.3)
7     RDIST=SCALE*DIST
      IF(RDIST.LT.0.0)PMFL=1
      RDIST=ABS(RDIST)
      IDIST=RDIST
      WRITE(2,1002)SCALE,RDIST,IDIST,IDIST,DIST
1002  FORMAT('0',2X,'SC=',F8.3,'RD=',F8.3,'ID=',I8,'IDH=',Z16,
     6'DIST=',F8.3)
```

Figure D-3(b). Millbase and Thermocouple Moving Subroutine — MOVER

```
$ASSM
         STM    0,RGSTG
         LHI    1,X'A8'
         LHI    3,X'C0'
         LHI    2,X'A9'
         OCR    1,3
         OCR    2,3
*
*    Load the distance into reg. 6, the command code into
*    reg.7, and the plus or minus indicator into reg.8.
*
         L      6,IDIST
         LH     7,MOVR
         LH     8,PMFL
         LR     3,6
         NI     3,Y'FFFFFF'
         LR     4,3
*
*    Mask out everything but the 8 least significant bits
*    and load them out to downcounter 1.
*
         NHI    4,X'FF'
         OHI    4,X'2100'
         WHR    1,4
         BAL    12,OWAIT
         RHR    2,5
         STH    5,OUT1
*
*    Shift the next 8 bits into the least significant
*    position and load them out to downcounter 2.
*
         SRLS   3,8
         LR     4,3
         NHI    4,X'FF'
         OHI    4,X'2200'
         WHR    1,4
         BAL    12,OWAIT
         RHR    2,5
         STH    5,OUT2
*
*    Shift the next 8 bits into the least significant
*    position and load them out to downcounter 3.
*
         SRLS   3,8
         LR     4,3
         NHI    4,X'FF'
         OHI    4,X'2300'
         WHR    1,4
         BAL    12,OWAIT
```

Figure D-3(c). Millbase and Thermocouple Moving Subroutine -- MOVER

```
         RHR    2,5
         STH    5,OUT3
*
*     Check to see if the move should proceed in the negative
*     direction. If so, add the appropiate bit to the command.
*
         CHI    8,1
         BNE    PM
         OHI    7,X'0080'
PM       NHI    7,X'0FFF'
         OHI    7,X'2000'
         WHR    1,7
         BAL    12,OWAIT
         RHR    2,5
         STH    5,OUT4
         LM     0,RGSTG
         B      $P211
OWAIT    LHI    11,X'0FFF'
LUP      SIS    11,1
         BZR    12
         B      LUP
$FORT
211    WRITE(2,106)IDIST,MOVR,PMFL,OUT1,OUT2,OUT3,OUT4,RDIST
106    FORMAT('0',5X,Z16,5X,Z8,5X,Z8,5X,Z8,5X,Z8,5X,Z8,5X,Z8,F16.8/)
       PMFL=0
1      CONTINUE
2      CONTINUE
4      CONTINUE
5      CONTINUE
6      CONTINUE
8      CONTINUE
9      CONTINUE
10     CONTINUE
11     CONTINUE
12     CONTINUE
13     CONTINUE
14     CONTINUE
15     CONTINUE
16     CONTINUE
17     CONTINUE
18     CONTINUE
19     CONTINUE
21     CONTINUE
22     CONTINUE
23     CONTINUE
       RETURN
       END
```

Figure D-3(d). Millbase and Thermocouple Moving Subroutine — MOVER

```
C
C
C-------SUBROUTINE BLASH(CMAND,DIST)
C
C
C            This subroutine moves the millbase in the
C       X,Y, or Z direction, taking into consideration the
C       backlash of the gearing.  It accomplishes this by
C       reading the beginning location of the  millbase,
C       computing the correct finishing location and moving
C       the millbase until this location is reached.  The
C       arguements passed into the subroutine are CMAND and
C       DIST.  CMAND is the ASCII code for X,Y, or Z and
C       DIST is the distance of movement in millimeters.
C
C
C
        SUBROUTINE BLASH(CMAND,DIST)
        INTEGER CMAND,BUF(13)
        INTEGER*2 BUFOUT(16),XSIGN,YSIGN,ZSIGN
        REAL DIST,AMT;FIN,BDIST,ABAMT
102     FORMAT(' ',2X,'OK',I2)
        CALL READR(BUF)
        XSIGN=0
        YSIGN=0
        ZSIGN=0
        CALL BOUT(BUF,BUFOUT,XSIGN,YSIGN,ZSIGN)
        WRITE(2,110)XSIGN,YSIGN,ZSIGN
110     FORMAT('0',5X,'X= ',I8,2X,'Y= ',I8,2X,'Z= ',I8)
        AMT=DIST*40.
        IF(CMAND.EQ.Y'58202020'.OR.CMAND.EQ.Y'49202020')I=1
        IF(CMAND.EQ.Y'59202020'.OR.CMAND.EQ.Y'4A202020')I=2
        IF(CMAND.EQ.Y'5A202020'.OR.CMAND.EQ.Y'4B202020')I=3
4       FIN=(BUFOUT(I)+AMT)/1000.0
        BDIST=DIST
104     FORMAT(' ',2X,'OK',I2,2X,I3)
101     CALL MOVER(CMAND,BDIST)
        CALL CHECK
        CALL READR(BUF)
        XSIGN=0
        YSIGN=0
        ZSIGN=0
        CALL BOUT(BUF,BUFOUT,XSIGN,YSIGN,ZSIGN)
        WRITE(2,110)XSIGN,YSIGN,ZSIGN
6       AMT=FIN-BUFOUT(I)/1000.0
        ABAMT=ABS(AMT)
        IF(ABAMT.LE..001) GO TO 100
        BDIST=AMT*25.4
        GO TO 101
```

Figure D-4(a). Closed Loop Millbase Movement -- BLASH

```
100    I=0
       RETURN
       END
```

Figure D-4(b). Closed Loop Millbase Movement — BLASH

```
C
C
C--------------SUBROUTINE SHOT(TIME)----------------
C
C
C          This subroutine commands the interface to
C     take a shot of a specified duration. The time
C     duration is passed to this subroutine by the
C     arguement TIME.  TIME is a real variable and
C     should correspond to the desired time duration
C     in seconds.  The interface must be under
C     computer clock control for this subroutine
C     to operate correctly.  This subroutine uses
C     the 10KHz. clock signal of the interface for
C     timing purposes.
C
C
      SUBROUTINE SHOT(TIME)
      INTEGER ITIME,A(26),RGSTG(16)
      INTEGER*2 OUT1,OUT2,OUT3,OUT4
      REAL TIME
      COMMON/MOV/A(26),RGSTG(16)
      A(1)=Y'41202020'
      DO 2052 I=2,26
2052  A(I)=A(I-1)+Y'01000000'
      TIME=TIME*10000.
      ITIME=TIME
$ASSM
         STM    0,RGSTG
         LHI    1,X'A8'
         LHI    2,X'A9'
         LHI    3,X'C0'
         OCR    1,3
         OCR    2,3
*
*     Load reg.6 with value of ITIME
*
         L      6,ITIME
         LR     3,6
*
*     Mask out first 8 bits and load into reg.4
*
         NI     3,Y'FFFFFF'
         LR     4,3
*
*     Mask out first 24 bits and load least significant
*     bits into downcounter 1 of interface.
*
         NHI    4,X'FF'
```

Figure D-5(a). Ultrasound Irradiation Subroutine --SHOT

```
         OHI   4,X´2100´
         WHR   1,4
         BAL   12,OWAIT
         RHR   2,5
         STH   5,OUT1
*
*     Shift next 8 bits into least significant position
*     and load these out to downcounter 2
*
         SRLS  3,8
         LR    4,3
         NHI   4,X´FF´
         OHI   4,X´2200´
         WHR   1,4
         BAL   12,OWAIT
         RHR   2,5
         STH   5,OUT2
*
*     Shift last 8 bits into least significant position
*     and load out to downcounter 3
*
         SRLS  3,8
         LR    4,3
         NHI   4,X´FF´
         OHI   4,X´2300´
         WHR   1,4
         BAL   12,OWAIT
         RHR   2,5
         STH   5,OUT3
*
*     Supply interface with command to initiate a shot.
*
         LHI   7,X´2548´
         WHR   1,7
         BAL   12,OWAIT
         RHR   2,5
         STH   5,OUT4
         LM    0,RGSTG
         B     $P211
OWAIT    LHI   11,X´0FFF´
LUP      SIS   11,1
         BZR   12
         B     LUP
$FORT
211   WRITE(2,106)ITIME,ITIME,OUT1,OUT2,OUT3,OUT4
106   FORMAT(´ ´,´SHOT DATA´,5X,I7,5X,Z16,5X,Z8,5X,Z8,5X,Z8,5X,Z8)
      TIME=TIME/10000.
      RETURN
      END
```

Figure D-5(b). Ultrasound Irradiation Subroutine -- SHOT

```
C
C---------SUBROUTINE READR(BUF)------
C
C
C            This subroutine reads the 13 bit groups
C       available from the interface into 13 full word
C       storage locations called BUF(13).
C       The data is not masked in any way.
C
C
        SUBROUTINE READR(BUF)
        INTEGER A(26),RGSTG(16),BUF(13)
        COMMON/MOV/A(26),RGSTG(16)
$ASSM
           STM    0,RGSTG
           LHI    1,X'A8'
           LHI    2,X'A9'
           LHI    3,X'C0'
           OCR    1,3
           OCR    2,3
           LHI    5,X'2900'
*
*      Load the interface command "2900" into reg.5
*      and hexadecimal "13" into reg.6.
*
           LHI    6,X'000E'
*
*      Load reg. 10 with address of BUF(1).
*
           L      10,BUF
*
*
*      Write to the DIO and branch to the delay routine.
*
WRITE      WHR    1,5
           BAL    12,WAITR
           RHR    2,7
*
*      Store contents of reg.7 into BUF
*
           ST     7,0(10)
           AIS    5,1
           AIS    10,4
*
*      Add 4 to reg.10 to set-up for next storage location.
*
           SIS    6,1
           BNZS   WRITE
           LM     0,RGSTG
```

Figure D-6(a). Interface Output Subroutine --READR

```
         B    $P100
WAITR    LHI  8,X´0F00´
SUB      SIS  8,1
         BZR  12
         B    SUB
$FORT
100   CONTINUE
      DO 10 I=1,13
      JI=I-1
10    WRITE(2,101)JI,BUF(I)
101   FORMAT(´0´,5X,I2,5X,Z16)
      RETURN
      END
```

Figure D-6(b). Interface Output Subroutine — READR

```
C
C
C--------SUBROUTINE BOUT(BUF,BUFOUT,XSIGN,YSIGN,ZSIGN)------
C
C
C          This subroutine converts the X, Y, Z, and CS
C     position readouts into integer numbers.  The raw
C     data must be stored in the locations of BUF which
C     can be accomplished by the subroutine READR(BUF).
C     The values of BUF are passed into this subroutine
C     and BUFOUT is passed out containing the 4 integer
C     values corresponding to the position readouts.
C     BUFOUT is organized in the following manner:
C          BUFOUT(1)= X position
C          BUFOUT(2)= Y position
C          BUFOUT(3)= Z position
C          BUFOUT(4)= CS position
C     BUFOUT is a half word integer and BUF is a full
C     word integer.
C
C          The positions are returned in BUFOUT with the
C     correct sign. Sign information is also returned
C     with the arguements XSIGN,YSIGN,and ZSIGN. The code
C     is given below:
C                     1=minus
C                     0=plus
C     XSIGN,YSIGN and ZSIGN should all have the value of
C     zero assigned to them previous to calling this
C     subroutine.
C
C
      SUBROUTINE BOUT(BUF,BUFOUT,XSIGN,YSIGN,ZSIGN)
      INTEGER BUF(13)
      INTEGER*2 BUFOUT(16),XSIGN,YSIGN,ZSIGN
      COMMON/MOV/A(26),RGSTG(16)
$ASSM
         STM   0,RGSTG
*
*     Load reg. 10 with the address of BUF(1) and reg. 11
*     with the address of BUFOUT(1).
*
         L     10,BUF
         L     11,BUFOUT
         LHI   8,X'0008'
*
*     Load the contents of BUF into reg. 6 and shift
*     right 4 places.
*
LOD      L     5,0(10)
```

Figure D-7(a). Position Encoder Interpretation Subroutine -- BOUT

```
         LR    6,5
         SRLS  6,4
         STH   6,0(11)
*
*     Mask out highest 28 bits
*
         NI    5,Y'0000000F'
*
*     Store these bits into the next BUFOUT location.
*
         AIS   11,2
         STH   5,0(11)
*
*     Set up reg. 10 to point to next BUF location.
*     and reg. 11  to point to next BUFOUT location.
*
         AIS   10,4
         AIS   11,2
         SIS   8,1
         BNZ   LOD
         LM    0,RGSTG
$FORT
C
C     Assign the correct weight to each BUFOUT location
C     and store the resulting integer number in the
C     first four locations of BUFOUT.
C
      I=1
      DO 10 J=1,4
      BUFOUT(J)=BUFOUT(I)*1000+BUFOUT(I+1)*100+BUFOUT(I+2)*10
     6+BUFOUT(I+3)
      I=I+4
10    CONTINUE
C
C     The following section extracts sign information from
C     location BUF(8) and adjusts the sign of BUFOUT(1),
C     BUFOUT(2) and BUFOUT(3) and assigns the correct
C     value to XSIGN,YSIGN and ZSIGN accordingly.
C
$ASSM
         STM   0,RGSTG
*
*     Load the address of BUF(1)
         L     10,BUF
*
*     Add 28 to reg. 10 to point to BUF(8)
*
         AHI   10,X'001C'
         L     11,XSIGN
```

Figure D-7(b). Position Encoder Interpretation Subroutine -- BOUT

```
        L      12,YSIGN
        L      13,ZSIGN
        LIS    6,1
*
*    Load the contents of BUF(8) into reg. 5
*
        L      5,0(10)
*
*    Check the least significant bit for a 1
*
        SRHLS 5,1
        BCS    NEXT
*
*    If this bit is a 1 (plus), skip to NEXT, if it is
*    a 0 (minus), store a 1 in reg.6.
*
        STH    6,0(13)
*
*    Repeat this checking for the next 2 bit locations.
*
NEXT    SRHLS 5,1
        BCS    NEX
        STH    6,0(11)
NEX     SRHLS 5,1
        BCS    NE
        STH    6,0(12)
NE      LM     0,RGSTG
$FORT
C
C      Adjust the signs of the BUFOUT locations accordingly.
C
      IF(XSIGN.EQ.1)BUFOUT(1)=-1*BUFOUT(1)
      IF(YSIGN.EQ.1)BUFOUT(2)=-1*BUFOUT(2)
      IF(ZSIGN.EQ.1)BUFOUT(3)=-1*BUFOUT(3)
      RETURN
      END
```

Figure D-7(c). Position Encoder Interpretation Subroutine -- BOUT