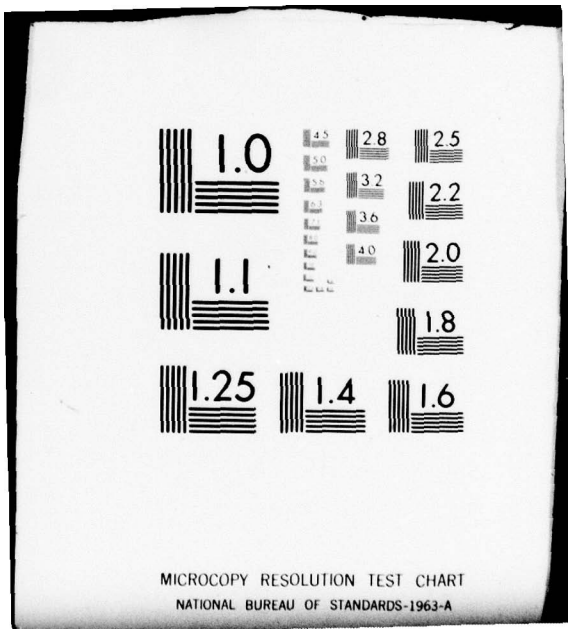
AD-A080 169 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 13/8
DESIGN OF A DIGITAL CONTROLLER FOR AN ELECTRON BEAM LITHOGRAPHY--ETC(U)
DEC 79 M L WEIDNER
UNCLASSIFIED AFIT/GE/EE/79-41 NL

1 OF 3
AD A080169


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

LEVEL II

UNITED STATES AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DDC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

Design of a Digital Controller for an Electron Beam Lithography System

Thesis

AFIT/GE/EE/79-41

Michael L. Weidner
Capt USAF

(14) AFIT/GE/EE/79-41

(6) Design of a Digital Controller for an Electron Beam Lithography System.

(9) Master's Thesis,

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

(10) Michael L. Weidner, B.S.E.
Capt USAF

Graduate Digital Engineering

(11) December 1979 (12) 246

Approved for public release; distribution unlimited

012225 LB

Acknowledgement

The extent and complexity of this thesis seemed to grow with each new day. Without the never ending faith of my wife Jean and the support of my daughters, this would probably never have come to pass.

I would like to express my thanks to the members of my thesis committee Maj. Alan Ross and Dr. Gary Lamont; and to my thesis advisor Capt. Mike Borky. I would especially like to thank those three, for their combined efforts to educate and advise me for the past year and a half resulted in this thesis. I hope it is a fitting tribute.

Not to go unmentioned, the Electrical Engineering lab technicians were responsible for wire wrapping the boards needed to prove the thesis. That is a long and labourious job and the dedicated efforts of Dick Wager and Dan Zambon saw the job to completion. A special thanks to Tom Kindle for keeping my lab room in one place, and to Bob Durham for breaking the red tape and getting the equipment and parts I needed.

Michael L. Weidner

Contents

Contents

List of Figures

List of Tables

Definition of Terms

Controller Period - The length of time between the A pulse outputs of the state generator.

Dwell Time - The length of time that the electron beam is turned on at any one point.

Micro-code - Software instructions that generate the register transfers of the controller. Not to be confussed with the higher level software of the Z-80A.

Micro-cycle - Time it takes to run the state generator one complete cycle.

Micro-memory - Memory used to hold the Micro-code.

Microprogram - Sequences of micro-code.

Micro-store - See Micro-memory.

Micro-store latch - A 40 bit register used to hold the current output of micro-store.

Primary buffer - Controller buffers loaded by the Z-80A.

Shape menu - A list of predefined shapes.

State Generator - Circuit that generates the timing signals to synchronize the controller.

Abstract

A digital controller for an electron beam lithography system was designed using microprocessor technology. Due to the high speed requirements of the system, the hardware controller was implemented using a microprogram sequencer. The controller was designed as a special purpose computer. The word length of the controller computer is 40 bits. Twenty-three of the bits are control bits. The other seventeen bits are used to determine which word to execute next.

A Z-80A microcomputer was used in the design as an input/output device. The Z-80A outputs the pattern data. The controller uses that data and sequences the electron beam to draw the requested pattern. The controller is double buffered for increased throughput. Since both the controller and the Z-80A are programmable, the capabilities of the system can be tailored to the needs of the user.

Chapter 1 System Requirements

Introduction

Integrated circuits are manufactured via a number of different processes. In all of these processes it is necessary at several steps to generate desired patterns on the substrate used for the integrated circuit. This thesis addresses controlling on electron microscope to write the patterns directly on the substrate using the focused electron beam. This technique is known as electron beam lithography. The scope of this study is limited to the design of the digital controller for a specific hardware configuration.

A scanning electron microscope with laser interferometer position feedback system is the main componement of the system. The electron beam column, the element of the microscope that generates, focuses, and positions the electron beam, has two sets of deflection lenses. The pre-lens is used by the scanning electron microscope for its video display. The post-lens, which provides a much finer beam focusing capability, will be used by the controller to position the beam. The post-lens requires high voltage levels on the order of ± 300 volts. These voltage levels are generated by high voltage amplifiers.

Requirements

There are two high voltage amplifiers (deflection amplifiers) driving the post-lens, one to define each axis (X and Y).

The inputs to the amplifiers are ±10 volts, full scale. Consequently, the output of a digital-to-analog converter can interface directly with the high voltage amplifiers. This technique will allow points on the substrate to be defined as X and Y coordinates which can then be converted to the analog signals that will position the beam to that point.

The electron beam column also has the capability to turn the beam on and off at very high speeds, referred to as beam blanking. In actuality the beam is not turned off; a set of parallel plates deflects the beam into a stopping aperture located approximately in the middle of the electron beam column. When beam blanking is not engaged, the electron beam is allowed to pass through to the lenses. The beam blanking signal will be provided by the digital controller. Beam blanking, combined with the ability to position the beam to specific points on the substrate, provides the capability to generate patterns.

However, if each point of a pattern must be defined, even the simplest patterns would be cumbersome. Therefore, the requirements for the beam deflection system are as follows:

1) Position the beam via X and Y coordinates.
2) Vector the beam between two points in the +X, -X, +Y, and -Y directions.
3) Programatically blank the electron beam.
4) Step (move) the beam at variable rates from 250 hertz to 250 kilohertz.

5) Provide variable bit density for X, Y coordinates from 12 to 16 bits.

6) Convert the digital values to analog signals of ±10 volts full scale.

These requirements are based on the exposure times of the various resists that are used for pattern generation. However, the ability to deflect the beam via the post-lens is not sufficient. The post-lens is only capable of deflecting the beam two millimeters on a side before the error due to warping the beam becomes excessive. Therefore, in order to generate patterns larger than four square millimeters, or to repeat the pattern several times across a wafer three inches in diameter, a stage is employed to reposition the wafer after pattern generation has been completed for the current deflection field.

The stage for this system is positioned by X and Y stepper motors. Pulses to the stepper motors must have at least 1 microsecond pulse width and a maximum of 40 kilohertz pulse frequency. Each pulse corresponds to 2.5 microns. This positioning is fairly gross compared to the submicron line widths desired. A laser interferometer monitors the X and Y stage position. As the stage moves, the laser interferometer generates a pulse for each .08 microns of movement. These pulses (UP, $\overline{\text{UP}}$, DOWN, and $\overline{\text{DOWN}}$) have a pulse width of 40 nanoseconds and a maximum frequency of 4.2 megahertz. Error and restart signals are provided to monitor the device since precise stage

movement is not possible without the laser interferometer operational.

Plan of Attack

The approach taken in this thesis was to first determine the requirements for the electron beam lithography system. Originally, construction of the hardware was going to be included in this thesis effort. However, in order to address a solvable portion of the problem during this thesis period, the scope of the problem addressed by this thesis did not include the construction of the hardware.

Results

The paper design of the digital controller and the digital interface boards was accomplished. Portions of the design were breadboarded to verify critical timing requirements. The three boards that comprise the digital controller were wire-wrapped. However, several minor wiring mistakes resulted in the inability to test the boards prior to the conclusion of the thesis.

Conclusion

The preceding paragraphs describe the requirements of the system to be designed by this thesis. These capabilities will allow the user to accomplish the tasks envisioned for the electron beam lithography system today. However, industry is moving toward finer resolution every day. In order to prevent the product of this thesis from being obsolete before the

control system is completed, the design effort was directed to exceed the requested capabilities and performance requirements. The controller was designed with a great deal of flexibility to accomodate future requirements.

Chapter 2 System Design

Introduction

With the advent of the microprocessor, very small inexpensive controllers became possible. For the most part these controllers are used in relatively inexpensive applications, while more expensive hardware systems seem to warrant larger and more expensive computer systems to control them. The goal of this design effort, therefore was to employ microprocessor technology in developing a controller with system capabilities similar to systems costing 100 times as much.

System Organization

Microcomputer instruction speeds are not sufficient to use a microcomputer directly as the controller. The requirements for the deflection system stipulate a 250 kilohertz beam step rate. This corresponds to a dwell time of four microseconds. The jump instruction on most microcomputers takes four microseconds. Since it would require many more instructions than just the jump to implement a simple control loop it becomes apparent that current microcomputers are not fast enough.

This instruction execution speed, however, is for the macro-instructions. The microcomputer may actually execute anywhere from 4 to 20 micro-instructions for each macro-in-

struction. If the micro-instructions were the individual controller instructions, then the microcomputer could be used directly. This was the general approach taken in the design.

A special purpose microprogrammable controller was designed. To increase performance and decrease the complexity of the microprogrammable portion of the system (henceforth called the Micro-controller), a dual-processor system was employed (See Figure 2-1). A Zilog eight bit micro-computer, the Z-80A, is used as the second processor. The Z-80A is responsible for all data handling while the Micro-controller is responsible for generating all control signals to sequence the pattern generation system. The functions necessary for pattern generation were grouped logically as follows:

1) X Deflection System
2) Y Deflection System
3) Stage Positioning System
4) Diagnostics System

The remainder of this chapter will address the design considerations and choices surrounding each block in Figure 2-1.

Z-80A

The function of the Z-80A microcomputer is data handling. This includes the generation of pattern data as well as directing pattern generation by the system using this data. The Z-80A is the executive of this control system. The menial tasks have

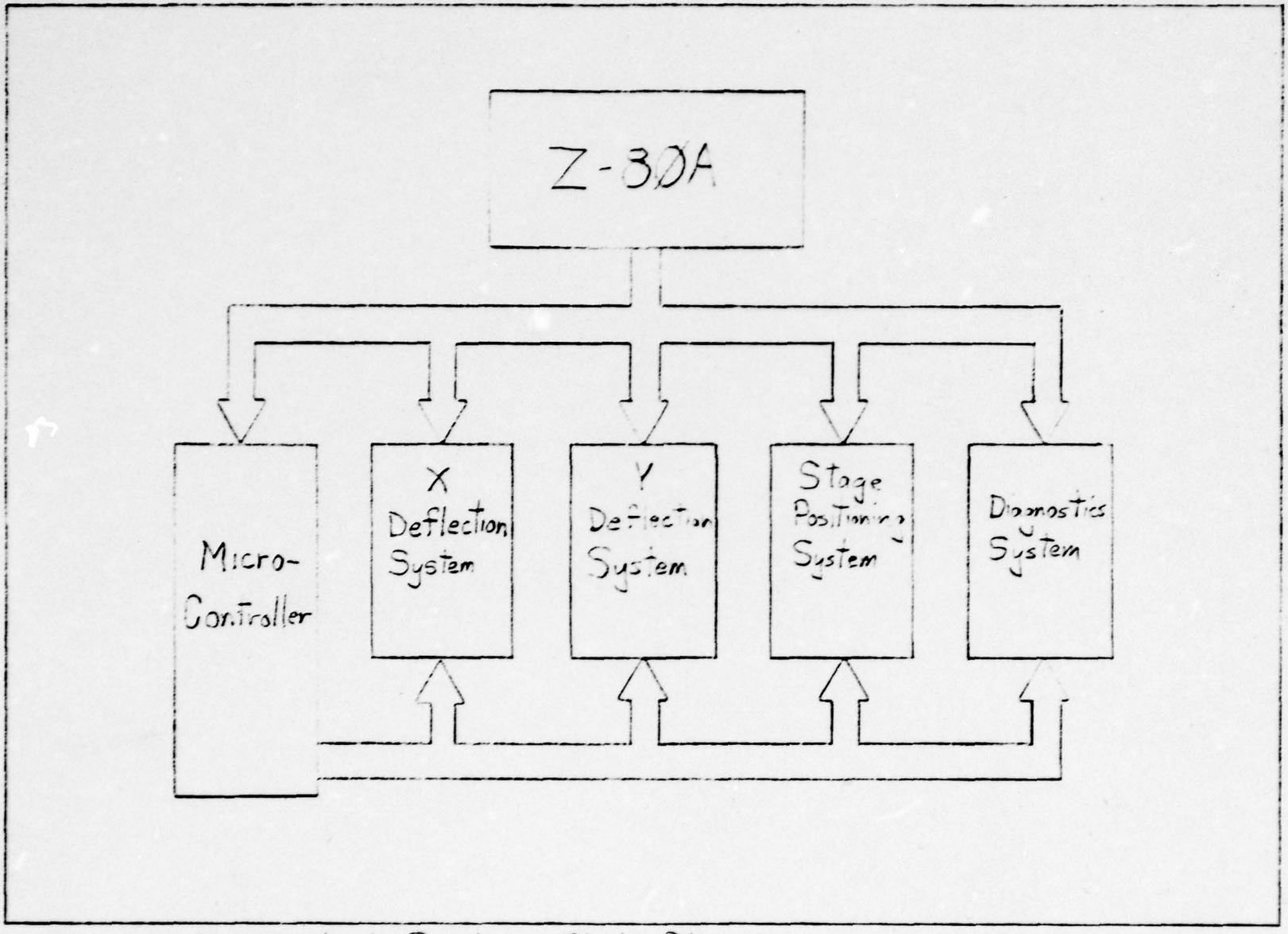


Fig 2-1 Control System Block Diagram

been delegated to the Micro-controller. The Micro-controller only performs when and what it is instructed to perform by the Z-80A.

Therefore, the Z-80A is responsible for supervising the work of the Micro-controller. The programs in the Z-80A use the pattern data to issue sequences of instructions to the Micro-controller. As one instruction is completed, the Z-80A issues a new instruction. When that instruction is completed, another is issued until the entire desired pattern has been drawn.

The various blocks of the control system are double buffered to increase throughput. The Z-80A fills the primary controller buffers with the data needed for the next instruction. When the necessary buffers are filled, the Z-80A instructs the Micro-controller to process the data via some specific microprogram. The primary buffers are then downloaded to the secondary buffers to be used by the Micro-controller. While the Micro-controller is processing the current instruction, the Z-80A can be reloading the primary buffers with data for the next instruction. When the Micro-controller has completed the current instruction, the Z-80A can issue the new instruction and the process repeats until the pattern is complete. For a more thorough treatment of the software systems required for the Z-80A see Chapter 8 Software Definition.

Micro-controller

The Micro-controller is the operational heart of the

control system. Its main function is to generate the control signals necessary to sequence the various elements of the control system (excluding the Z-80A). The Micro-controller is comprised of the following control modules:

1) Controller register decode logic
2) Bus interface
3) Variable Clock
4) State Generator
5) Micro-store
6) Memory Control Register
7) Microprogram sequencer
8) Z-80A Instruction Interface

Controller Register Decode Logic

For ease of maintenance all address decode logic for the control system is located on one board. The main controller registers are memory mapped as a block of 16 memory locations. A 12 position dip switch determines the high address of the block of registers anywhere within the address space of the Z-80A. Four input/output ports are also used to complete the addressing of the control system. Two ports are for the low and high bytes of the variable clock. One port is for the delay register, and the last port is for the memory control register. The port numbers are determined by their respective eight position dip switches.

Bus Interface

Since the Z-80A loads buffers in all modules of the control system, the Z-80A data bus must be available to all modules. Each board (except Micro-store) uses a pair of SN74LS367 bus drivers for the Z-80A data bus. Since these are low-power Schottky devices, the drain on the bus is minimal. The devices are enabled at all times so the controller can always read from the bus. The SN74LS367 is also used for several registers that output to the Z-80A data bus. The enable signal on these buffers is controlled by the Z-80A to insure the validity of the data on the bus and to eliminate timing problems.

Variable Clock

Two design criteria necessitated the design of a system clock. The first was the requirement for a variable step rate for the deflection system. Though there are a number of programmable clocks on the market, none could satisfy the second criteria, which was the range of clock rates required, from 200 hertz to 250 kilohertz. For these reasons a variable clock was designed.

The design was kept simple to decrease potential hardware problems. Like the rest of the control system the variable clock is double buffered. However, care must be taken with the timing of data transfer to the primary buffer by the Z-80A. This will become apparent as clock design and operation is discussed.

The variable clock is comprised of two 16 bit buffers and

a 16 bit presettable counter. The counter is configured as a down counter. The primary buffer is loaded by the Z-80A with a desired clock rate. Micro-code in the Micro-controller causes the clock rate to be loaded into the secondary buffer. This buffer is connected to the data inputs of the counter. When the counter reaches zero, the content of this register is loaded into the counter and the down count sequence starts again. If the value of the register is zero, the variable clock operates at the maximum frequency of eight megahertz.

Since this clock drives the whole control system, a means was needed to vary the clock rate within the same micro-program execution. Otherwise, certain data set-up tasks, which could be done at the maximum frequency, would have to be done at the step rate for that exposure. This could add several milliseconds to the time to draw each line as opposed to several microseconds. Since a zero in the register results in maximum frequency, the clear line for the secondary register can be set by micro-code in the Micro-controller.

In this way the Micro-controller can perform the various housekeeping tasks at the maximum frequency and only execute at the desired step rate during exposure times. It should be apparent that if the primary variable clock register is reloaded at the wrong time, for instance before the Micro-controller has downloaded it, the old data could be lost. For this reason, microprograms should only load the variable clock register once, and that should be at the start of the micro-

program.

This variable clock is used to sequence the control system. However, in order to avoid generating timing signals with awkward arrangement of one-shots triggered by the system clock, the variable clock is used to drive a state generator which provides precise timing signals which are synchronized to the variable clock.

State Generator

The state generator is used to synchronize the control system. Consequently, the first design decision to be addressed was the number of unique non-overlapping periods of the state generator that were required. Since the state generator provides the signals necessary to cycle micro-store, at least four periods are needed to generate those timing signals. Rather than attempt to overlap control signals and micro-store timing, it was necessary to add four more periods to the state generator for a total of eight periods. Since this design is a development effort, the exact nature of the state outputs are subject to change. Therefore, the state generator was designed so the state outputs could be easily altered.

The eight periods of the state generator are generated by a three-line-to-eight-line decoder fed by a three bit counter. These decoded outputs are used to set and reset flip-flops. Each flip-flop represents one of the state outputs of the state generator. All the user needs to know to

modify the state output is the period of the state generator at which the state output changes to a logical one and that at which the state output changes to a logical zero. Then the appropriate decoded state generator periods are tied to the preset or clear inputs of that flip-flop. The state outputs of the current state generator are shown in Figure 4-1. States B, C and D are used to generate the timing for micro-store.

Micro-store

Since this design was an initial development effort, the exact nature of the microprograms was not known. Therefore, micro-store was implemented as random access memory (RAM). In this way, the microprograms can be easily modified until the micro-code has been finalized. At that time electrically programmable read-only memories can be programmed with the micro-code to replace the volatile RAM memory. For ease in this transition, micro-store was implemented on a separate board. Therefore, in order to change micro-store, the user simply unplugs one board and plugs in the new one.

The contents of micro-store is the micro-code that drives the execution of the Micro-controller. In the development version of micro-store, the micro-code must be loaded by the Z-80A. The concept of shared memory by two or more processors is not unusual. However, the word length of memory which the Z-80A addresses is 8 bits, and micro-store is 40 bits. This requires that a 40 bit buffer be loaded by the Z-80A as five eight bit bytes and then written into micro-store. In order

not to confuse micro-store addresses with the Z-80A address space, an address buffer was added. The Z-80A loads the data buffer then loads the address buffer with the appropriate micro-store address and writes to micro-store.

The Z-80A does not directly cause the write-to or read-from micro-store. The Z-80A sets a flip-flop indicating the request for the operation. On the next full cycle of the state generator, the appropriate memory access is accomplished and the request flip-flop is reset. However, the Micro-controller also addresses micro-store directly. Therein lies a problem that had to be addressed. A means to arbitrate access to micro-store had to be developed so that the Z-80A and the Micro-controller did not both attempt to access micro-store simultaneously.

Memory Control Register

The arbitration problem was considerably simplified by the fact that the Z-80A only requires access to micro-store to load and edit it, while the Micro-controller only accesses micro-store after it has been loaded. Consequently, one bit in the control register can identify which processor has control of micro-store. This bit also doubles as the write protect for micro-store by making the read/write signal read-only while the Micro-controller has control of micro-store.

This procedure only determines what device has control of micro-store. If the Micro-controller has control, there is no further problem because micro-store is restricted to read-

only. However, when the Z-80A has control, micro-store must be both read and written. The next bit of the memory control register is used to signal a write. If the bit is set to a logical one, the next micro-store access by the Z-80A will be a write. Otherwise, the next micro-store access by the Z-80A will be a read.

The last bit of the memory control register is used to clear all register storage of the control system. This bit has no effect on micro-store. When this bit is set to logical zero, the system is cleared. It also disables the Micro-controller. This bit must be set to a logical one for the Micro-controller to function. When the Micro-controller has control of micro-store, the microprogram sequencer can execute the microprograms.

Microprogram Sequencer

The Fairchild 9408 Microprogram Sequencer is the driver chip for this module. This chip provides the following capabilities:

1) Ten bit branch address.
2) Unconditional branching.
3) Subroutine calls nested up to four deep.
4) Conditional branching.

The ten bit branch address allows the sequencer to directly address 1024 words of micro-store. Since the microprograms will likely be on the order of several words each, this amount of addressable memory was deemed sufficient. The

various means of addressing micro-store provide excellent flexibility in coding the microprograms. Conditional branching is accomplished in the same micro-cycle, based on one of four test inputs. This allows an instruction to loop on itself until a specific test input is either high or low. For instance, a single instruction to increment a counter could loop until the counter equaled zero. This feature will significantly decrease the amount of micro-code necessary since it will save one step on each conditional loop.

The capability to write microprograms as subroutines will facilitate easier program coding and debugging. It will also decrease the size of microprograms since routines that are repeated throughout the software need exist only once as a subroutine, and not many times as inline code. Restricting subroutine calls to four deep should not prove to be an unreasonable hindrance if it is a hindrance at all.

The 9408 has four unconditional branch instructions. Two signals on the output of the 9408 indicate which of the unconditional branch instructions is decoded. This feature is used to allow one instruction to reference an address register and the other instructions to reference micro-store. This capability greatly eased the problem of interfacing the Z-80A commands to the Micro-controller to execute a particular microprogram.

An instruction register is loaded by the Z-80A with the 9408 instruction to be executed. The instruction should be

either the conditional instruction that references the address register or a multi-way branch (See Chapter 8 Software Definition). The next instruction referenced should then cause a branch (conditional, unconditional, or subroutine) to the address in micro-store where the microprogram actually resides. This technique is known as indirect addressing. A design decision limited the address register to eight bits. Therefore, only the first 256 words of micro-store can be addressed directly. This poses only a slight inconvenience since all of micro-store can be addressed indirectly.

The branch address from micro-store and the address from the address register are both input to a two-line-to-one-line multiplexer. The decoded signal from the 9408 selects either micro-store or the address register for input. The instruction input selection to the 9408 required a similar solution.

Z-80A Instruction Interface

The system design requires that the Micro-controller remain in a halt state until instructed to execute a particular program. Once instructed by the Z-80A, it then begins executing the program. The Z-80A cannot issue a new instruction until the previous one has been completed. This scheme fits conveniently with instruction input selection. Address zero is the halt state for the Micro-controller. It is not really halted, but rather looping on word zero which does nothing. When the Z-80A wants the Micro-controller to execute a particular

microprogram, the Z-80A fills the address register with the address of the program (or an indirect address to the program) and loads the instruction register with the appropriate instruction.

When the instruction register is loaded, a flip-flop is set indicating the next micro-instruction is to come from the instruction register. While that instruction is being executed the flip-flop is cleared, signifying the next instruction comes from micro-store. Obviously loading the instruction register while the Micro-controller is executing could have disastrous results. To prevent this from happening, the instruction selection flip-flop can only be set when the Micro-controller is in the halt state.

Only 17 bits of micro-store are used for the functions discussed thus far. The remaining 23 bits combine with various state outputs to generate the signals that control the rest of the hardware.

X and Y Deflection Systems

These two elements of the control system will be discussed together since their functions and hence designs are identical.

The function of the deflection system is to define the coordinates of a point on the substrate in terms of X and Y binary values. The requirements further stipulated that the system must be capable of incrementing or decrementing these

coordinates at variable bit densities and at varying rates. variable bit density allows the user to vary the number of bits that define a deflection field and hence vary the step size. For instance, if the bit density was 16 bits, then 2^{16} bits define the deflection field, or approximately 65,000 unique points. However, if the bit density was 12 bits then 2^{12} bits define the deflection field as approximately 4,000 unique points. Since the deflection field remains the same, the different bit densities result in varying the increment size, or step size. The variable rate is accomplished by using an output of the state generator which is driven by the variable clock. The requirement for variable bit density necessitated the design of a special purpose counter.

The counter must be able to count from bit position zero, count from bit position one while holding bit position zero constant, count from bit position two while holding bits zero and one constant, count from bit position three while holding bits zero, one and two constant, or count from bit position four while holding bits zero, one, two, and three constant. It must also be capable of counting at a frequency of one megahertz. There is no counter commercially available that meets these requirements.

The approach taken was to design a single element of the counter which could be cascaded. After a bread-board version of the single element was operational, the deflection counter was designed. It consists of three SN74191 binary counters cas-

caded with a four bit version of the variable bit density counter. The outputs of the counter are input to the digital-to-analog converter for the appropriate deflection axis.

A multiplexer on the data inputs to the counter selects between a low address and a high address. When a load signal is present, the data selected will be loaded into the counter. The multiplexer is also used to select which address to compare against the counter for determining end-of-line status. Since the address registers must still be valid after the counter is loaded, both address registers are double buffered for increased throughput.

The X and Y deflection boards are completely independent except for bit density selection since the same bit density applies to both the X and Y axis. Consequently, any combination of increments or decrements of the X and Y deflection systems is possible. For instance, X can be incremented and Y decremented in the same micro-cycle, or Y can be incremented while X is held constant. The latter would result in a vertical line, the former would result in a diagonal line. By various combinations of X and Y increments lines can be drawn at any angle. Since the lines are being drawn as increments of X and Y, the lines are not straight but staircased. If the bit density is selected as 16 bits, the staircase effect is minimized because a least significant bit change in X or Y corresponds to about three one-hundredths of the line width.

The range of the deflection system is two millimeters on

a side. Since wafers can be as much as three or four inches in diameter, this would result in much wasted surface area and greatly limit the size of the devices that could be generated. To overcome that problem the wafer is mounted on a moveable stage. After the current deflection field has been exposed, the stage repositions the wafer for a new deflection field.

Stage Positioning System

The stage is moved by X and Y stepper motors. Each step corresponds to 2.5 microns of movement. However, this is not accurate enough. The stage must be capable of positioning a new deflection field accurately enough so that lines that connect between the fields do indeed connect. To accomplish this task, a laser interferometer monitors the position of the stage.

The laser interferometer is capable of measuring stage movements within 0.08 microns. However, the stage position can not be adjusted to that accuracy. These facts guided the design of the stage positioning system.

The stage positioning system is actually comprised of the X stage positioning system and the Y stage positioning system. However, their functions and designs are identical. Therefore any reference to the stage positioning system will refer to both the X and Y systems.

It takes 21 bits to represent a 4 inch stage movement in 0.08 micron increments; therefore, a 21 bit counter is used to

keep track of the actual stage position. Since a single step of the stage is not an even multiple of laser interferometer measurements, an eight bit counter is used to keep track of the overshoot or undershoot after it is moved to its new position. This corresponds to four stage steps of overshoot or undershoot. If the software which directs the stage movement also monitors the status of the stage movement, the error should not exceed one step. The error is converted to an analog signal and summed as a dc offset with the deflection analog signal to correct the relative position of the deflection field.

To determine when a stage movement has been completed, the stage positioning system has a register which contains the desired address. This address is compared with the actual address. When they are equal, the end stage movement flip-flop is triggered. At this point the stage position counter is still being updated, and the error counter begins receiving the count pulses. The updating is only stopped when the desired address register is loaded with an address different from the actual address. By this means, the error signal can correct for any minor pertubations during pattern generation.

When a different desired stage address is loaded, the error counter and stage complete flip-flop are cleared. Step pulses are sent to the stepper motors until the actual address equals the new desired stage position. Then, once again, the error counter is updated continuously to correct for error.

The desired address register is double buffered so the

stage movement can be set up while the last figure is being drawn. The time saving is not great considering the time required for a stage movement; however this does optimize the process with no appreciable overhead.

The design process just outlined deals with all the modules necessary for pattern generation. However, as an aid during development of the system, and as a check to determine if the system is operational, a diagnostic capability was included.

Diagnostics System

Diagnostics, as applied in this thesis, represents more a design philosophy than a separate module in the control system. The capability to check various elements of the control system was designed into the modules themselves rather than implemented in an additional module. To take advantage of these features, software routines must be written. These routines are referred to as the Diagnostic System.

A Zilog analog input board is included as an integral part of the system. All analog outputs of the system will be tied into the multiplexer of this board. Consequently, all elements of the deflection system can be checked by exercising them. For instance, a value can be loaded into the X deflection counter. The analog multiplexer can then select the X analog input and convert the signal to a digital value. The actual value can be compared against the expected value.

The variable bit density can also be checked in a similar

manner. First, the bit density is set to a desired value, then the values from the analog input board are compared to the expected results. This is then repeated with another bit density.

The diagnostics will not isolate a problem down to a specific chip, but will allow the user to determine within what module a problem exists. From there, the user should be able to isolate the problem fairly easily.

Conclusion

This chapter has provided a systems view of the design of this digital controller. The intent has been to furnish a general overview and to indicate the basis for fundamental design decisions. The details of the design are presented in the remaining chapters.

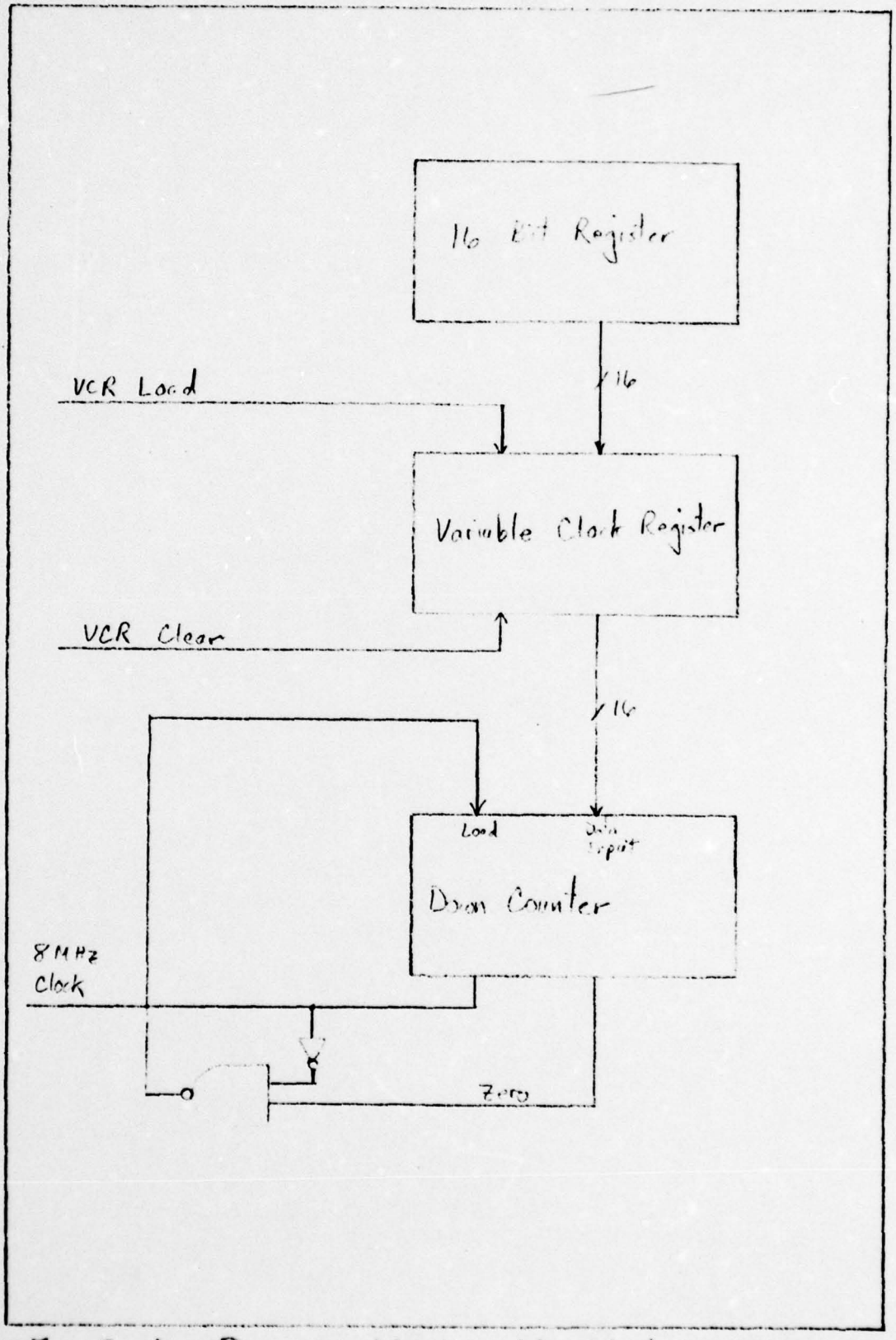


Fig 3-1 Programmable Variable Clock

Chapter 3 Programmable Variable Clock

Introduction

In the process of manufacturing integrated circuits, various patterns need to be layed out on the silicon wafer. The lines that comprise the patterns will be of varying line widths. Using electron beam exposure, the line width is a result of the beam current, type of resists, thickness of the resist and the beam latency period at each point along the line. The Micro-controller is not concerned with the type of resist and resist thickness. The software that generates the pattern data must take into account these factors affecting line width and generate the data appropriately. This will be discussed in greater depth in Chapter 8, Software Definition.

The line width can be varied by leaving the dwell time constant and varying the beam current. A larger current will result in greater line width (due to the splash of the electrons as they impact the surface). Conversely, a decreasing beam current will result in a finer line width. Beam current adjustment is not available to the controller; however, it can be adjusted manually. Due to this restriction an average beam current must be selected and the controller must vary the line width by varying the dwell time at each point along the line. In order to vary the dwell time, a programmable variable clock was designed (See Fig. 3-1). The purpose of this chapter is to address the details of the design and operation of the programmable variable clock.

Design

Dwell times from four microseconds to four milliseconds in increments of one microsecond were required for the controller. Ten bits of precision would be necessary for the range required. However, it takes no additional time for 16 bits as opposed to 10 bits. Since it provides a much greater range and extends future capabilities, the clock was designed with 16 bits of precision.

The variable clock drives the state generator. The state generator divides the clock by eight to generate the timing signals. Therefore, in order to furnish one microsecond increments to the controller, the variable clock frequency must be eight megahertz. This results in a beam dwell time of from 1 microsecond to 0.065936 seconds in increments of 1 microsecond.

In order to ease interfacing with the Z-80A microprocessor and to facilitate preprocessing by the Z-80A, the clock circuitry was doubly buffered. The primary buffer is loaded by the Z-80A. The Micro-controller can then load the variable clock register (VCR) from the primary buffer. The output of the VCR provides the input to 4 cascaded SN74191 binary counters. These counters are configured to count down. When the count reaches zero, the counters are reloaded from the VCR.

The variable clock was breadboarded using the elite-3 circuit design test system. In order for the circuit to operate properly, the 8 Megahertz clock input must have a pulse

width of 50 nanoseconds ±20 %. A shorter pulse width results in a race around condition and therefore generates spurious signals. A larger pulse width causes overlap in the signals and the output becomes fixed.

For stable operation, the maximum frequency of the clock input to the breadboarded variable clock was found to be 13.5 Megahertz. This is more than sufficient to meet the current requirements. If, in the future, a faster stepping rate is required, the variable clock will need to be redesigned.

Operation

When the system is powered-up, the reset signal from the Z-80A ($\overline{\text{Master Reset}}$) clears both the primary buffer and the VCR. Consequently, when the controller is not in the automatic mode, the variable clock operates at the maximum frequency of 8 Megahertz resulting in a controller bandwidth of 1 Megahertz. In order to vary the clock, the primary buffer must first be loaded by the Z-80A with the length of the clock period desired.

The following formula must be used to generate the correct value of the controller period:

Controller Period = (period desired) — 1 microsecond

If a microsecond period is desired then the controller period would equal one minus one, or zero. The number must be expressed in binary. If a 33 microsecond period were desired then the controller period would equal 33 minus 1, or 32. This

would be 40 in octal notation.

After the primary buffer is loaded, the VCR can be loaded and cleared by the microprogram in micro-store only when the Micro-controller is in the automatic mode. Bit six of micro-store is the load bit and bit five is the clear bit. With the controller in automatic mode, a logical one in bit six will cause the VCR to be loaded from the primary buffer at state generator pulse F (See Chapter 4). A one in bit five will cause the VCR to be cleared at state generator pulse E. If the programmer sets both bits five and six in the same micro-cycle, the VCR will be cleared during state E and loaded from the primary buffer during state F. This is the same as just loading the VCR. Loading and clearing the VCR should not be done with the beam on since the dwell time would be affected.

Various register transfers take place before the controller starts deflecting the beam. The same clock is used to accomplish all the register transfers and vector the beam accross the resist. Since the variable clock is doubly buffered, the register transfers take place at the maximum frequency. When these are complete, the VCR is loaded with the dwell time and the next state generator cycle will be at the new frequency. Provided the clearing and loading of the VCR is done only at the start of each microprogram, the Z-80A can set up for the next figure by loading the primary buffer with the next beam dwell time. However, if the microprogram

clears and loads the VCR anywhere else in the microprogram, the software in the Z-80A must take care not to update the primary buffer until the current figure is complete. Otherwise, the current beam dwell time could change as a result of the preprocessing.

Conclusion

The variable clock circuit designed exceeds the variable step rate requirements. It provides the capability to programmatically vary the step rate from 0.0659 seconds to 1 microsecond in intervals of 1 microsecond. However, this is only the variable clock for the Micro-controller, and cannot sequence the Micro-controller. The next step was to design a state generator, capable of being driven by the variable clock, that would provide all of the synchronization signals for the electron beam lithography system.

Chapter 4 State Generator

Introduction

A number of different signals are needed to synchronize all the tasks of the Micro-controller. These signals must occur in a specific sequence. For instance, a memory write to micro-store requires, the address inputs be valid for a specific time period; the chip enable be active beginning a number of microseconds after the address inputs become valid; the data must be valid on the inputs while the address and chip enable are valid; and finally a write signal must be generated to strobe the data into memory. This example only addresses timing for micro-store. There are also a number of other unique signals that must be generated to synchronize the rest of the Micro-controller. The purpose of this chapter is to address the problem of generating these signals.

Design

The initial effort was to utilize the 8224 two phase clock chip. Since the clock was only two phase the rest of the timing signals would still need to be generated. One-shots could be designed to provide pulses of fixed length with a fixed delay. This approach to generating the signals would have resulted in a kludge of resistors and timing capacitors. For that reason that design approach was abandoned. The next approach taken was to design a special digital circuit driven by the variable clock to generate the necessary timing signals.

The primary function of the state generator is to cycle micro-store and to synchronize all the elements of the controller. Non-availability of high speed memory chips resulted in the state generator being designed to cycle a slower memory chip, the 2102A. The memory read is one half of the total state generator cycle time. At maximum frequency, the read cycle time is 500 nanoseconds. This is more than sufficient for the slower memory devices.

An additional function of the state generator is to write into micro-store. The $\overline{\text{write}}$ signal must be active low for 350 nanoseconds. This is approximately three-fourths of the read cycle time. In order to generate a $\overline{\text{write}}$ signal for the right length of time, the state generator cycle must be broken up into eight periods. At maximum frequency each period would be 125 nanoseconds. Four periods are used to generate the memory access signals, the other four periods are used to generate various control signals for synchronizing the elements of the controller.

A three bit binary counter (74177) was used to generate the eight periods of the state generator (See Fig. 4-1). A three-to-eight line decoder (74138) was used to decode the current period. However, the outputs of the decoder were not clean signals. Spikes were seen in half of the decoded outputs that were large enough to trigger TTL logic. In order to generate clean decoded signals, the clock input was ANDed with the decoder outputs.

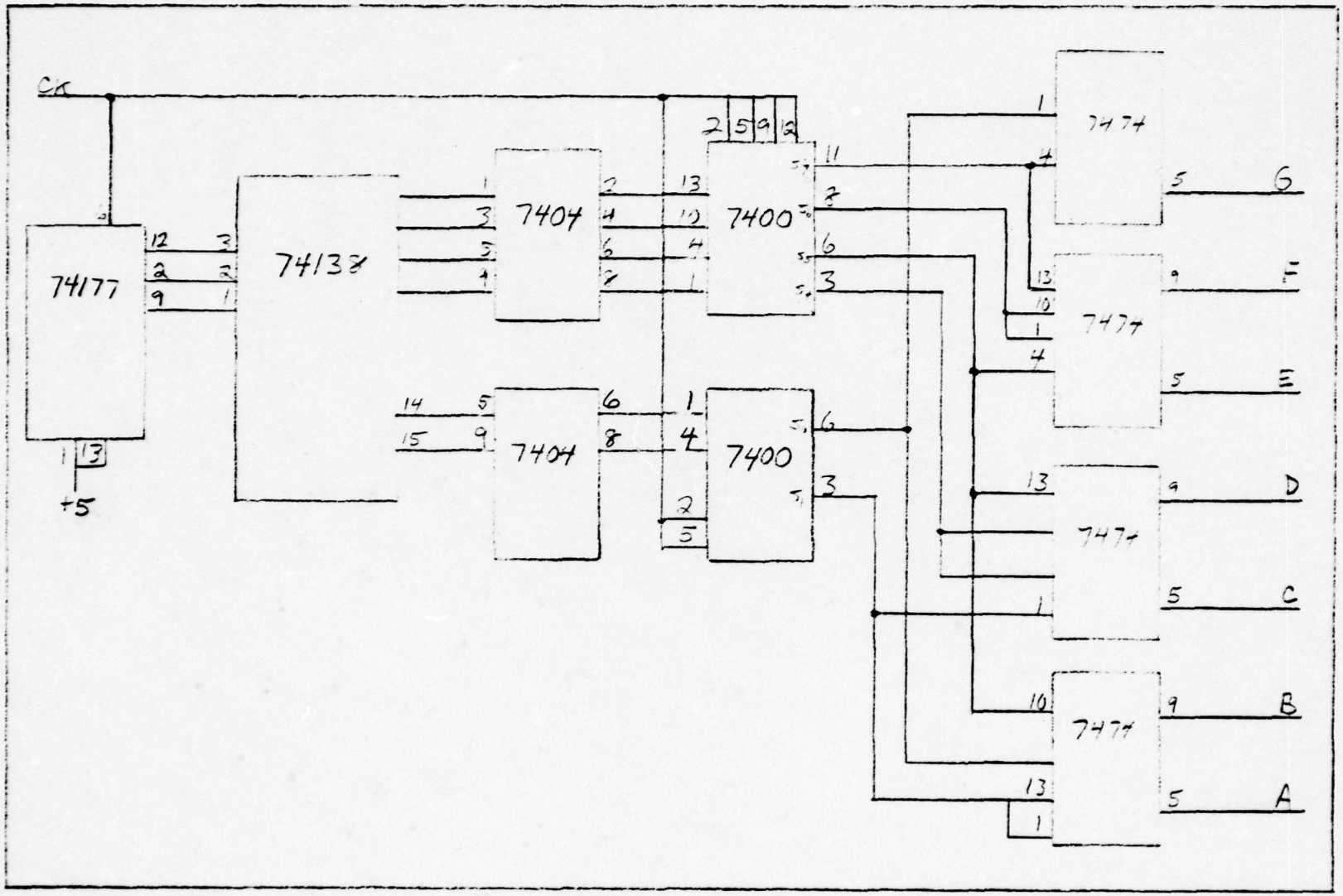


Fig 4-1 State Generator

The ANDed signals were inverted and tied to appropriate preset and clear inputs of flip-flops to generate the seven states (See Fig. 4-2). The state generator was breadboarded using an Elite 3 circuit design test system. Using the output of the variable clock breadboarded earlier, the state generator operated properly over the full range of the variable clock. Like the variable clock, the state generator is band limited at about 13 Megahertz. Consequently, any desired increase in operating frequency would necessitate a redesign of the state generator. In order to facilitate possible future changes in the state generator, the generation of state A will be discussed in detail.

State Generation

State A is defined as a signal which is high during period 0 and low at all other times. Thus the A flip-flop (IC 2-55) must be set by the period zero signal and reset by the period one signal. By connecting the clean period zero signal to the preset input of the flip-flop and the clean period one to the clear input of the same flip-flop, the Q output of that flip-flop will generate state A.

State Definitions

The various outputs of the state generator were designed to generate specific timing signals to synchronize the various elements of controller (See Table 4-1). The most important of these signals are the signals used to load and access memory

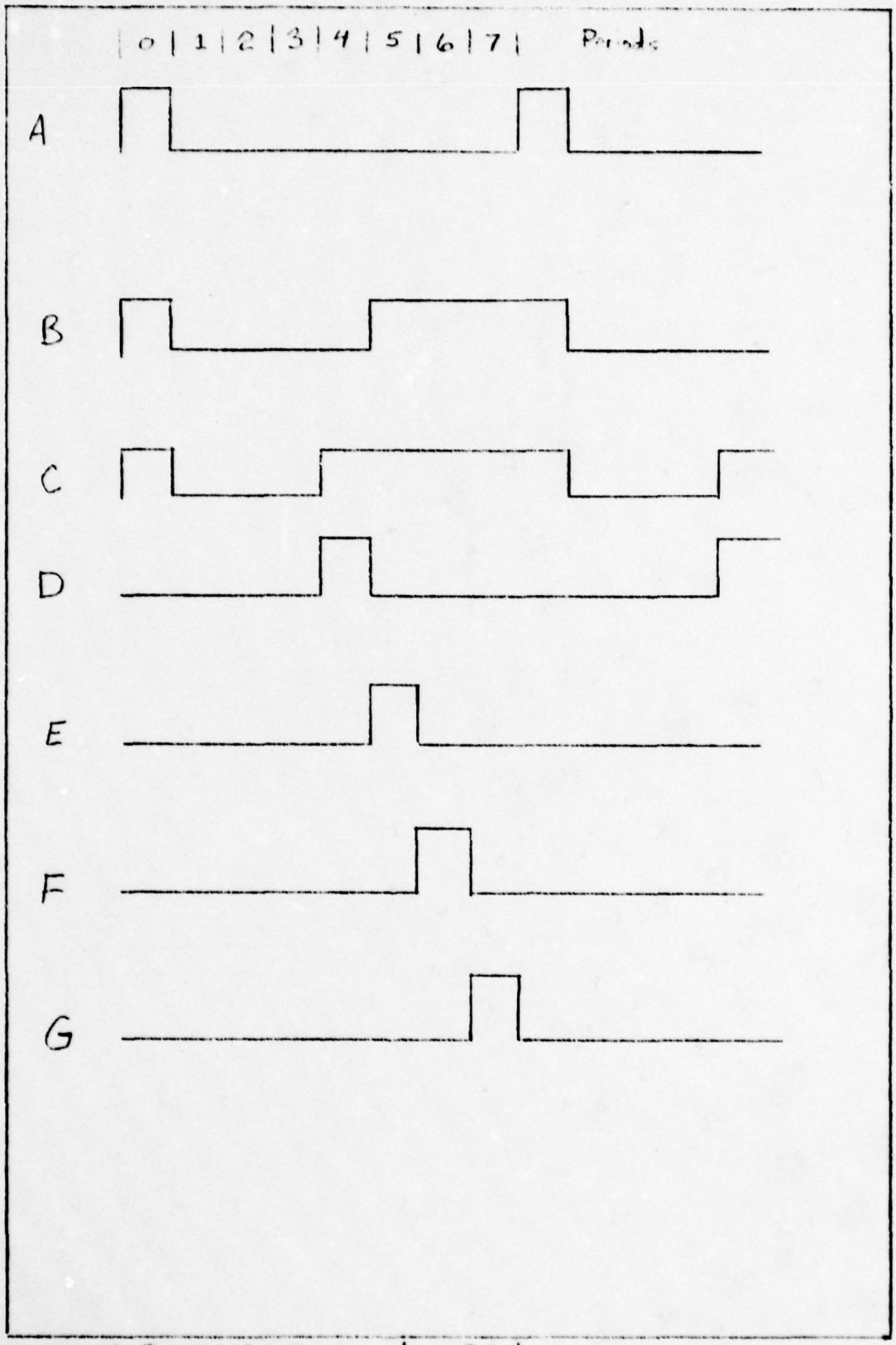


Fig 4-2 State Generator States

STATE	FUNCTION
A	Clock pulse
B	Chip Enable
C	Memory write
D	Load micro-store latch
E	Load primary buffers; count
F	Load Counters
G	9408 Strobe input; Beam ON; D/A Enable
$\overline{D}$	Stage Pulse

Table 4-1 State Generator States and Functions

(States B, C and D). The length and timing of the memory access signals were determined by the timing requirements of the 2102A memory device (see timing diagrams in Fig. 4-2).

State B is the chip enable ($\overline{CE}$) signal that enables memory to either read or write. The length of the chip enable signal was determined by the write cycle. During this cycle, the chip enable must be held low 20 nanoseconds beyond the write signal. One technique was to generate a delay for the chip enable signal to hold it low longer. However, this approach would not result in a clean simple design. It was decided to add one full period (125 nanoseconds) to the length of the chip enable signal. This resulted in $\overline{CE}$ being considerably longer than necessary. However, this approach was realized in a very simple manner.

State C is the write signal. This signal is only used to write to micromemory. The length of the write signal according to the timing diagram (Fig. 4-3) is a minimum of 350 nanoseconds. This requires 3 periods of the state generator for a total period of 375 nanoseconds.

State D is the data strobe. After the data is valid on the putputs of the memory chip, the data is loaded into the micro-store latch. With the outputs of micro-store latched, the next address can be set up, facilitating a pipeline scheme to decrease processing time.

The remainder of the signals are active only when the controller is in the Auto mode. In order to prevent signals

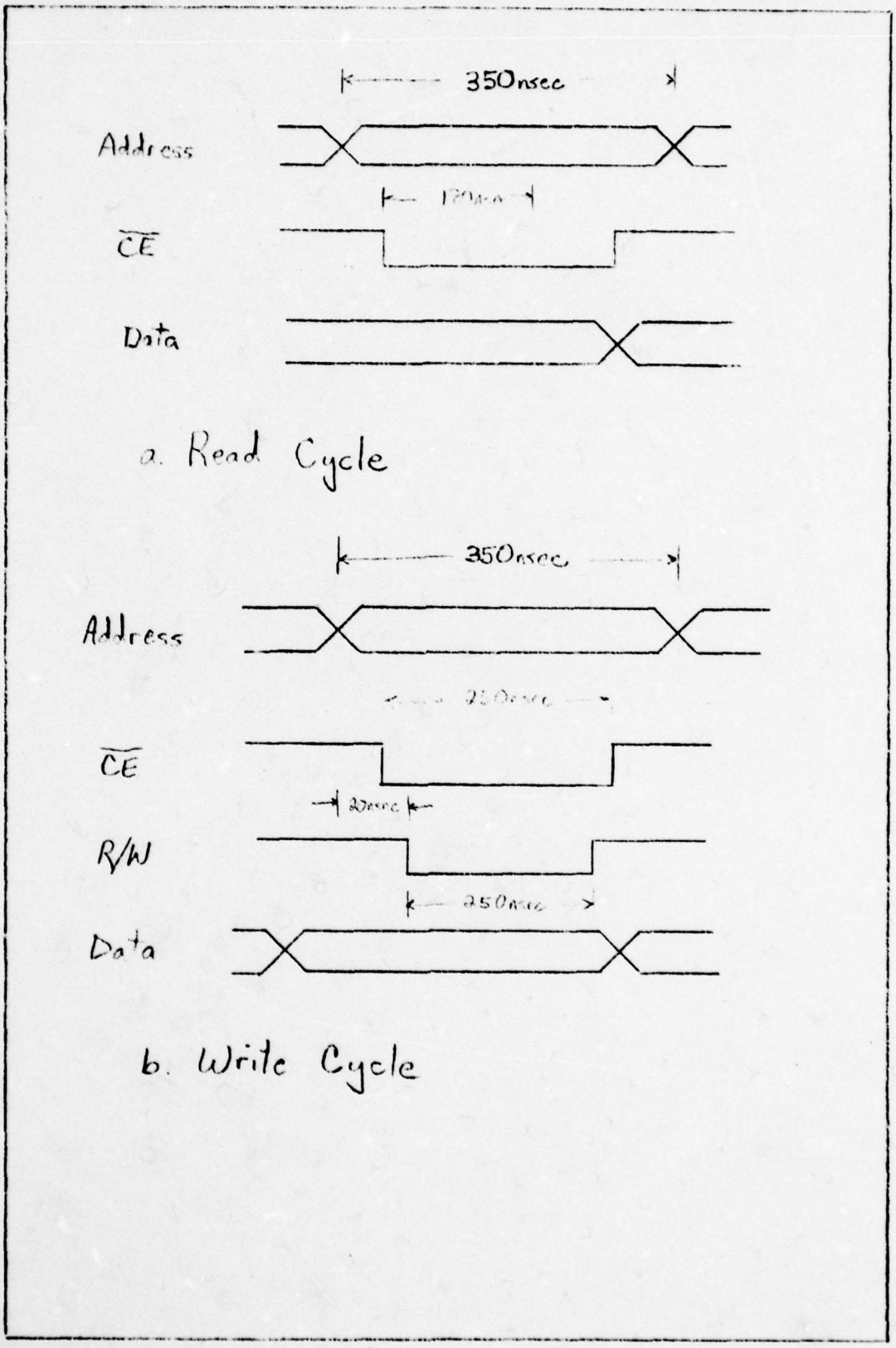


Fig. 4-3 Timing Requirements for 2102 Memory

from being generated during the loading and editing of micro-store, the remainder of the signals are suppressed when not in Auto mode.

State A is the clock to the Fairchild 9408 microprogram sequencer. The 9408 is configured in the non-pipeline mode, by connecting pin 5 of the 9408 to V_{cc}, so the next branch address will appear on the address outputs sooner. In order to accomodate branching in the same micro-cycle, the test inputs must be strobed into their flip-flops in the 9408 115 nanoseconds prior to the clock pulse. State G was selected for this function since the leading edge of this state occurs 125 nanoseconds before the leading edge of CP. State G is used to turn the electron beam on and off, as well as to enable the digital-to-analog (D/A) converters used in the X and Y deflection systems (See Chapter 6).

In order to load the X and Y deflection counters from the primary buffer in one micro-cycle, the controller uses states E and F. State E loads all secondary buffers from their primary buffers when bit 21 of micro-store is a logical 1. State F then loads the counters as specified by the micro-program in micro-store. Consequently, the controller can load the secondary buffer, load the counter, and enable the D/A converter in one micro-cycle. If there has been no change in the D/A input value, the electron beam could also be turned on during the same micro-cycle. However, if the D/A input value has changed, the electron beam should not be turned on until

the analog output has had sufficient time to settle. See Chapter 6 for a more detailed discussion of the electron beam deflection system.

The stepping motors which position the stage require a pulse width of one microsecond. In order to operate the micro-cycle as fast as possible and still output a step pulse, $\overline{D}$ is combined with the count commands for the X and Y stages to generate the appropriate step pulse. $\overline{D}$ is the inverted version of the signal used to load the micro-store latch. See Chapter 7 for more detail on the stage positioning system.

Conclusion

This chapter addressed the design of the state generator. The resultant design will generate the signals necessary to synchronize the elements of the Micro-controller, however, these signals are generated continuously. Therefore, the Micro-controller must be designed to selectively utilize these signals to generate the required signals.

Chapter 5 Micro-Controller

Introduction

The functions of the Micro-controller are to interface with the Z-80A and to generate the sequence of control signals necessary to synchronize the elements of the control system. Included in the interface tasks are 1) the decode logic for the controller registers, 2) the generation of the micro-store write sequence, and 3) special handshaking circuitry to handle the asynchronous communications with the Z-80A.

The main thrust of this chapter will be the detailed design of the Micro-controller. The system design of the Micro-controller will be addressed first, followed by a thorough coverage of the design of its components. The chapter will conclude with the lessons learned and recommendations concerning the Micro-controller.

System Design

The design of the Micro-controller could be accomplished either of two ways. The first approach was to design a purely hardware controller. The controller would be designed to accomplish a specific set of functions. The Z-80A could then use combinations of these functions to accomplish the tasks necessary for pattern generation.

The main advantage of this design approach is the increased bandwidth of the controller. This type of controller could be

designed, conceivable, to operate at frequencies of 10 to 15 megahertz. The disadvantages, however, are many. First, and foremost, the design is not flexible. The field of integrated circuit technology is fast changing, and a pure hardware design is not conducive to change. In many cases a complete redesign may be necessary to meet changing requirements.

The pure hardware controller is also much more complex to design. This is because all the functions of the controller must be implemented in the hardware. Plus, a complex control module must be designed to select and sequence the appropriate function modules. Consequently, interfacing of the many modules can become very complex. All of this could easily lead to a design that only the original designer could understand and change.

To alleviate this potential problem, a second design approach was employed. This approach requires only a slight change in design philosophy. The various function modules still need to be designed as hardware modules. However, the control module that decodes the commands and sequences the appropriate modules would be implemented in firmware. This means all the control signals would be generated by microprograms.

The advantages to this design approach are numerous. The most obvious advantage is the flexibility of the design. Since the microprograms sequence the controller, changing the way a function is accomplished can, in some cases, be

done just by changing the microprogram. Additionally the number and type of functions that can be accomplished is limited only by amount of addressable micro-store.

The design of the control module is considerably simplified since microprograms will generate the control signals. The design of the control module then becomes the design of a microprogram sequencer. This effort is partially accomplished by a number of manufacturers that fabricate microprogram sequencers. The function of these modules is to determine the address in micro-store of the next word to be executed in a sequence of instructions. That is only the first step in the design.

The Fairchild 9408 Microprogram Sequencer was chosen for the driver for several reasons. First, it provided a variety of conditional, unconditional and subroutine branching capabilities. It was capable of directly addressing 1024 words of micro-store. The 9408 had its own four level stack for nesting subroutine calls. The four unconditional branch instructions had decoded outputs that could be used to select any one of four branch address inputs. Finally, the circuit could be operated at frequencies above one megahertz.

With the 9408 chosen as the driver, the system level design of the Micro-controller was addressed. The design is predicated on the Z-80A interfacing indirectly with the Micro-controller via an instruction register and an address register. This will allow the Z-80A to request the execution of specific

microprograms to accomplish desired tasks. Note the Z-80A also has data inputs to micro-store.

Since this design will result in a development system, the microprograms were not yet determined. Therefore, the Z-80A must have a way of loading the microprograms into Micro-store. To insure that micro-store is written into only during loading, a memory control register is used. This register provides the write protect for micro-store. Micro-store was designed using random access memory to provide ease of program development. One board is dedicated to micro-store. Consequently, once the microprograms have been developed, they can be burned into non-volatile programmable read-only memory, and the new micro-store board would simply replace the previous one. In this way, the microprograms will not need to be reloaded prior to each pattern generation run.

If it was discovered that one set of microprograms did not provide all the capabilities desired, micro-store could be designed to allow the user, under software control, to select one of perhaps a number of programmable read-only memories. Thus, micro-store could be any number of 1,000 word memory blocks. This technique would support a shape menu approach in micro-store. The Z-80A would load a start address in the X and Y deflection buffers, and command the Micro-controller to draw the appropriate device at that location. This technique can be employed with the current design since micro-store can be loaded and reloaded at any time. However, depending on the

load routine, the micro-store load could require one or two seconds to accomplish. Consequently, if an excessive number of micro-store loads was necessary, the micro-store load time could exceed the pattern generation time. The controller would be spending more time loading microprograms than executing them.

Since the intent of this design was to maximize the capabilities of the controller, the capabilities discussed above must be included in the design of the Micro-controller. A number of circuits inherent in the Micro-controller, are necessary to accomplish interfacing between the Z-80A, The Micro-controller, and other elements of the control system. These circuits will also be addressed in the design of the Micro-controller. The functions of the Micro-controller were broken down into the following five modules.

1) Controller register decode logic
2) Bus interface
3) Micro-store
4) Z-80A Instruction interface
5) Microprogram sequencer

The design of each of the modules above will be addressed in detail in the remainder of this chapter. The variable clock and the state generator, though part of the Micro-controller, had significant enough impact on the control system to warrant separate chapters dedicated to their design and operation (See Chapters 3 and 4). They will not be addressed in this chapter.

Controller Register Decode Logic

For ease of maintenance all timing signals for the Microcontroller are generated on one board. The decode logic generates the select pulse that causes a particular controller register to be loaded from the Z-80A data bus. The method used to design the decode logic determines the addressing scheme that must be employed to address the controller registers. In order to take advantage of the multi-byte input/output instructions of the Z-80A, the main controller registers are memory mapped as a block of 16 memory locations. A 12 position dip switch determines the high address of the block of registers anywhere within the address space of the Z-80A.

Three SN7485 comparators, compare the data in the 12 position dip switch to the 12 most significant bits of the address on the Z-80A address bus. When the addresses are the same, the equal output of the comparator will be high. This signal is input to the positive enable line on 2 SN74138 3-line-to-8-line decoders. The inverted enable line is connected to the memory write signal from the Z-80A. Consequently, when the memory write line is low (signalling a write to memory) and when the equal input is high, a decoder will be selected based on the value of bit three of the address on the address bus.

This bit is routed directly to the third enable input of one of the decoders. The same bit is also inverted and routed to the third enable input of the other decoder. The

remaining three bits of address are connected to the select lines of both decoders. One decoder will generate controller register select signals 0 thru 7 and the other decoder will generate signals 8 thru 15.

The outputs of the decoders are normally high, except the selected line is low. These signals are used as the load input to SN74175 4-bit registers. The load input to the SN74175 is active high. Therefore, inverters were connected to the outputs of the decoders to generate the correct pulses. The decode logic discussed thus far was for memory write.

The decode logic for the memory read was designed in the same manner, with two exceptions. First, the memory read signal from the Z-80A is connected to one of the enable lines instead of the memory write signal. Second the memory read select signals are used to enable SN74LS367 tri-state buffers, and the enable signal is active low. Consequently, the outputs of the decoder do not require inverting.

The 16 decoded memory write signals were not sufficient. Four more select signals were necessary for the Z-80A to be able to load all the required registers in the Micro-controller. These control parameters change infrequently. Consequently, it was decided to use four input/output ports to complete the addressing of the control system. Two ports are for the low and high bytes of the variable clock. One port is for the memory control register, and the last port is for the delay register. Each of these decode circuits is designed essentially

the same.

An eight position dip switch is used to designate the port number. This number is compared to the eight least significant bits of the address bus. When they are equal, and when a port output is signaled, the select signal is generated. The select signal is generated by ANDing the equal signal with port output signal (both active high).

All of the decode circuits generate signals that cause Micro-controller registers to be loaded from (or output to) the Z-80A data bus. The Micro-controller must therefore provide bus interfacing.

Bus Interface

In order to prevent overloading the Z-80A address bus and data bus, the Micro-controller uses SN74LS367 tri-state buffers as bus drivers. Each board uses a pair of SN74LS367 tri-state buffers to input the Z-80A data bus. Since these are low-power Schottky devices, the drain on the bus is minimal. These buffers are enabled at all times so the controller always has the contents of the data bus available.

Board 2, which contains the decode logic, also uses the SN74LS367 to buffer the address bus. This is the only board that accesses the Z-80A address bus.

The tri-state buffer is also used as the input enabling device for memory reads from the Micro-controller. The enable lines are connected to the read select signals. Consequently,

the data is placed on the data bus, only when the Z-80A has specifically requested the data and is ready for it. The Micro-controller has no control over these buffers.

The SN74LS367 is used on board 1, micro-store, to buffer the address and control line inputs to the memory elements. Two levels of buffering were necessary in order to insure fan-out of the tri-state buffers was not exceeded.

Micro-store

The effort of this thesis was to design a development system. Since the hardware has not been built, the exact nature of the microprograms is not known. Therefore, micro-store was implemented as random access memory. In this way, the microprograms can be easily modified until the micro-code has been finalized. At some later date, the microprograms could be stored in one of the non-volative types of memory, to replace the volatile RAM memory. Micro-store was implemented on a separate board to facilitate readily changing micro-store. Consequently, in order to change micro-store, the user simply unplugs one board and plugs in the new one.

The word length of micro-store is 40 bits. The Z-80A word length is only 8 bits. This disparity arises whenever the Z-80A must load micro-store. Consequently, special circuitry needed to be designed to overcome this problem.

The first approach was to allow the Z-80A to directly address micro-store as five eight-bit arrays. This approach

had one major drawback. It required an excessive additional amount of hardware to decode the address and select the correct array to access. For this reason another approach was considered.

A 40 bit micro-store input buffer (addressable by the Z-80A, since it alone can write to micro-store) could be loaded as 5 bytes of data. In order not to confuse micro-store addresses with the Z-80A address space, an address buffer was also added. The Z-80A loads the data buffer then loads the address buffer with the appropriate micro-store address and writes to micro-store. Since the Z-80A does not directly address micro-store, the Z-80A cannot directly cause the write-to or read-from micro-store.

A special circuit was designed (See Fig. 5-1) to accept as asynchronous read or write request (depending on bit one of the memory control register) from the Z-80A, and generate the appropriate memory timing signals. The circuit uses the state generator outputs to cycle memory.

The Z-80A sets flip-flop 2 using the decoded write select 7, if the controller is not operational (bit 0 of the memory control register equals logical 0). The next state G signal will set flip-flop 1 to prepare for the next full cycle of the state generator. State B combines to generate the enable signal and state C combines with bit one of the memory control register to generate the $\overline{\text{write}}$ signal. The $\overline{\text{write}}$ signal can only be generated when the controller is not operational.

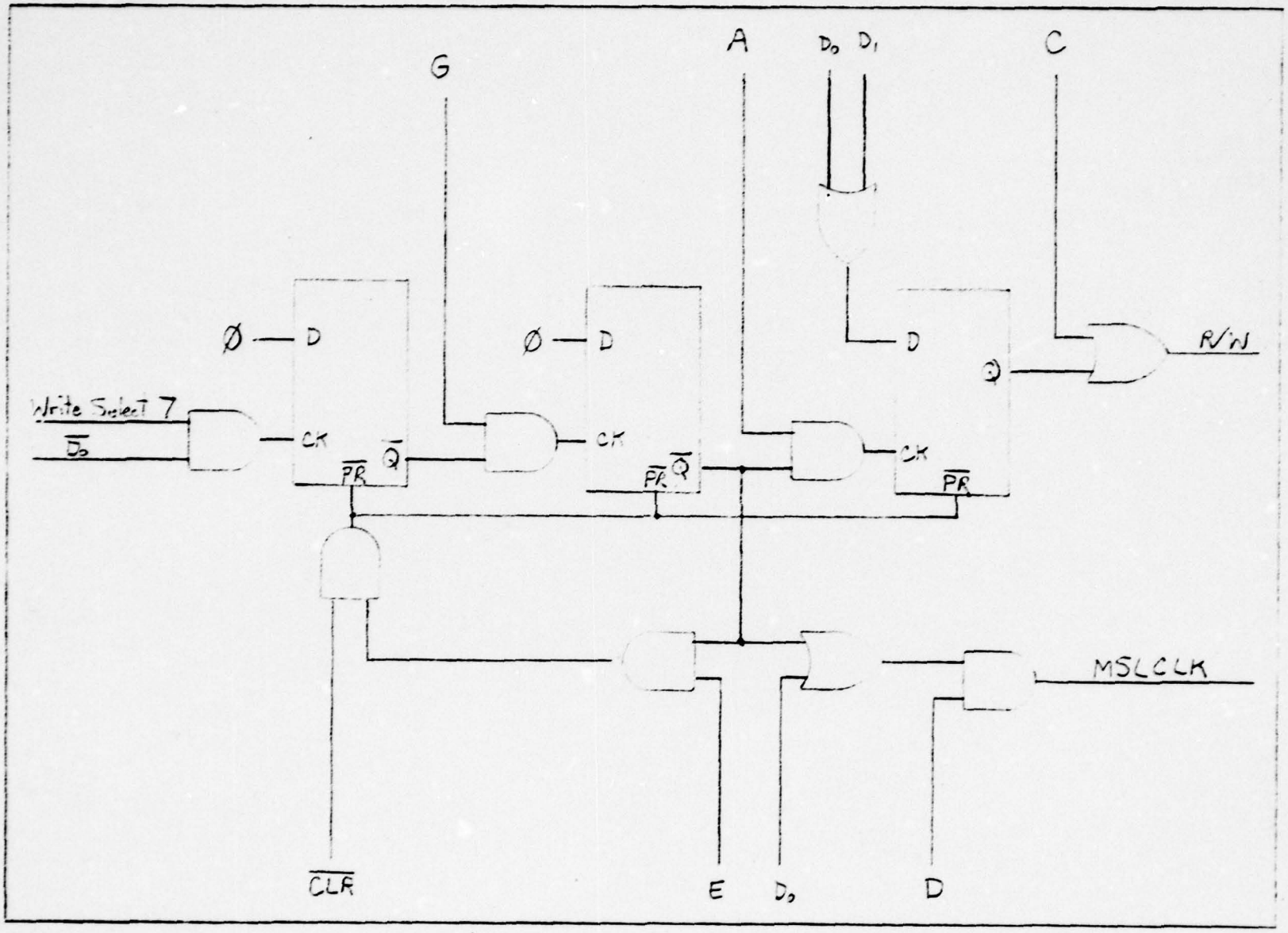


Fig 5-1 Micro-Store Read/Write Signal and Latch Clock Circuit

Therefore micro-store is write-protected when the controller is operational. State D is used to generate the load signal for the micro-store data latch.

One more asynchronous communications problem existed between the Z-80A and the Micro-controller. The Z-80A needed to be able to command the Micro-controller to begin executing a particular microprogram during the next full cycle of the state generator.

Z-80A Instruction Interface

The 9408 microprogram sequencer uses three inputs (excluding test inputs) to determine the next branch address; branch address, multiway branch data, and instruction. Therefore, the Z-80A must have a means of presenting this data to the 9408. However, for the Micro-controller to sequence through micro-store the same three inputs to the 9408 must come from micro-store. A circuit was needed to generate the appropriate select signal for a multiplexer (See Fig. 5-2).

The approach was to design a select circuit that would select the instruction input from an instruction register (filled by the Z-80A) only when the Micro-controller was halted and the instruction flip-flop was set. At all other times the circuit would select micro-store for the input.

The asynchronous instruction interfacing was addressed in a manner similar to the write to micro-store problem. Flip-flop one is set, asynchronously by the Z-80A (provided the Micro-controller is in the halt state) when the instruction

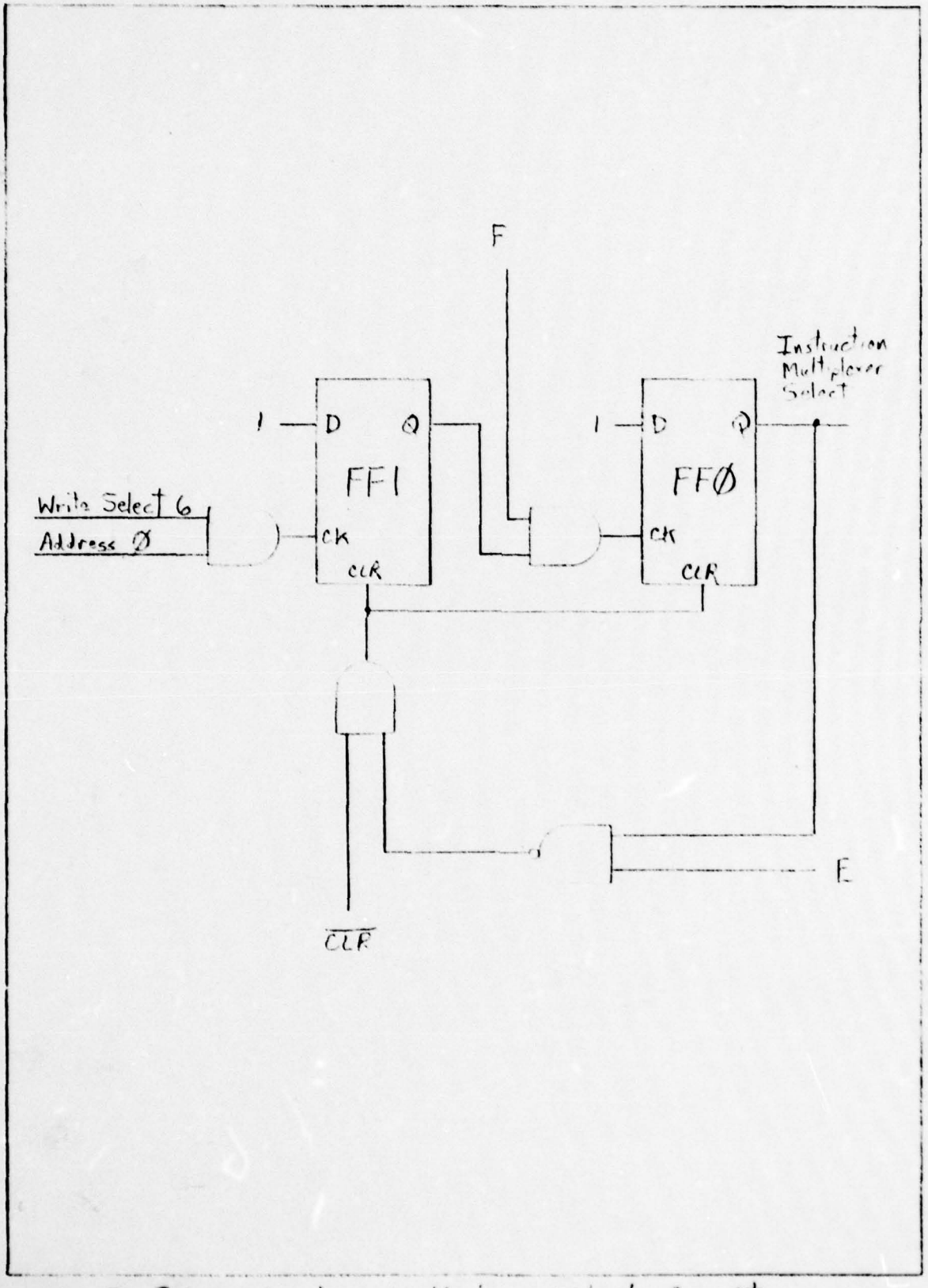


Fig 5-2 Instruction Multiplexer Select Circuit

is output to the instruction register. In order to insure the Micro-controller starts on a full-cycle of the state generator, flip-flop zero is set during state G. This causes the instruction register to be selected for input. State D clears both flip-flops, if flip-flop zero has been set. This results in selecting the instruction register only once, and then proceeding by selecting micro-store for further inputs.

Since the instruction flip-flop can only be set once the Micro-controller has halted, the Micro-controller will cycle through the instructions in micro-store until it reaches a halt. The halt instruction is actually an idle loop at address zero. The instruction merely loops on itself with all control lines disabled. This continues until the next instruction from the Z-80A is recognized. At which time, the Micro-controller processes that instruction. The process continues until the desired pattern is generated. This sequencing of instructions from micro-store is accomplished by the microprogram sequencer.

Microprogram Sequencer

The capabilities of the Fairchild 9408 Microprogram Sequencer have been discussed previously. Several sequencers had similar capabilities, however, the 9408 was the only one available during the time frame of this thesis. Consequently, this directed the early design decisions, though not adversely. When an instruction is executed by the 9408, decoded outputs can be used to generate the select signal for the address multiplexer.

Three decoded pinouts on the 9408 combine to select either the address register (loaded by the Z-80A) or micro-store. The BRV3 instruction is the only instruction that selects the address register for input to the 9408. Every other instruction causes micro-store to be selected. In order to limit hardware (to decrease input/output times) the address register was restricted to eight bits. This still allowed the Z-80A to directly address the first 255 words of micro-store, and indirectly address all the rest of micro-store. Since the indirect branch takes only one microsecond, this will amount to a savings of probably several microseconds for every address the Z-80A outputs.

Conclusions

Using the 9408 microprogram sequencer has provided a controller design that in every way exceeds specifications. The 9408 could run at twice the bandwidth of the current design, however, the strobe input to the 9408 that latches the test inputs would have to change since the test inputs must be clocked approximately 90 nanoseconds before the clock pulse (CP) input to the 9408. There is a faster version of the sequencer, the 9408A, which would solve even that problem if higher bandwidth were necessary.

Recommendations

If the size of micro-store proves inadequate, there are several possible solutions. Two have already been addressed.

The first was to have several ROMs that could be software selected by the Z-80A. Or similarly, several versions of micro-store that could be loaded and reloaded in the RAM version of micro-store. Another approach which has not been addressed yet, is to provide three bits in micro-store to control additional address bits. This could conceivably extend the range of micro-store indefinitely. These three bits would be used to increment, decrement, or clear a high address counter. The size of micro-store would be limited only by the size of the counter. Branches, especially conditional branches, would have to be programmed bery carefully. A careless branch instruction while the beam is deflected across the substrate could have disastrous results.

Chapter 6 Electron Beam-Deflection System

Introduction

This chapter will address the design of the electron beam deflection system. These boards are the interface between the Micro-controller and the post-lens of the electron beam column. The purpose of these boards is to identify the position of a point on the substrate. The position must be converted to an analog signal to interface with the post-lens. The analog signal directs the deflection of the electron beam across the substrate.

Patterns are generated by moving the beam across the substrate with the beam on for exposure and blanked for movement. Combinations of the two result in a pattern being exposed in the resist. A vector scan approach is used for beam movement. The beam position is defined by the X-Y coordinates, which identifies a specific point on the substrate.

The X-Y coordinates are generated by counters with the outputs feeding D/A converters to provide bipolar voltages to the deflection amplifiers. The voltages are amplified to $\pm$300 volts to provide the grid voltages for the electron beam column. Changing the grid voltages causes the magnetic field to shift which changes the direction of the electron beam.

Requirements

The system must be capable of moving the electron beam in the +X, -X, +Y, and -Y directions at variable rates. The

controller must be able to turn the electron beam on and off in conjunction with beam movements, under program control. The system must also provide variable bit density, from 12 to 16 bits, software selectable. The X-Y coordinates must be converted to a bipolar voltage output of $\pm$ 10 volts. Gain and offset adjustments on the D/A converters must be provided for calibration.

Design Considerations

There are two approaches that could be used for the X-Y deflection system. The first technique, called a shape menu, would be for the system to define each point to be exposed, load the D/A register with the point to be exposed, turn on the beam for a fixed period of time, then load the D/A register with the next point to be exposed, and so on. The use of a shape menu was used by IBM for the pattern generator in Vector Scan I (Ref 1). The number of points that are required to define a shape can grow dramatically with the complexity of the shape.

Using a microprocessor for the data transfer device greatly limits the capability to use a shape menu approach unless the beam dwell time at each point is 50 microseconds or more. That speed assumes that all the points to define the shape are already directly addressable and do not have to be read from a peripheral device. However, even though this method would be slow due to the data transfer rates of the

microprocessor, the shape menu is still a viable approach, and possibly the only reasonable approach for generating some patterns. Therefore, the design must include at least a limited shape menu capability.

The second approach is to define low and high X and Y addresses, load the X and Y counters with either address, and increment or decrement the addresses as appropriate to draw the desired line or rectangle. For instance, the X and Y counters might be initially loaded with their respective low addresses. A single point could be exposed by turning the electron beam on for some computed dwell time. If, while the electron beam was turned on, either the X or the Y address was incremented, the beam would expose a line across the resist. A rectangle could be exposed either by repeated X lines while incrementing Y until complete or by repeated Y lines while incrementing X until complete (See Fig. 6-1). Since most patterns that will be laid out will be composed mostly of lines and rectangles, this approach shows the most promise.

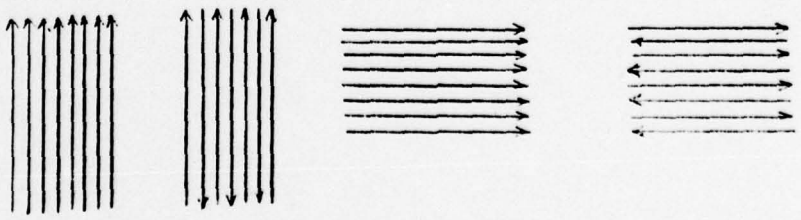


Fig. 6-1 Deflection Approach

Since the count commands are generated by software (the microprograms in micro-store), a 45 degree line can be drawn by incrementing or decrementing (as appropriate) X and Y at the same time. For instance, if the deflection counters were loaded with the low addresses, incrementing both X and Y counters would result in a 45 degree line. Lines of almost any angle can be drawn with the appropriate micro-code. However, as the angle of the line approaches the X or Y axis a large staircase effect will be noticed. This effect will increase, becoming worst at angles of 22.5 or 67.5 degrees, then decrease back to negligible as the line approaches the axes.

If a 16 bit density is selected, increment (or decrement) by 1 least significant bit represents 3 one-hundredths of the width of the line. Consequently, for angles close to 45 degrees, the staircase effect will appear as no more than a ripple on the edges of the line drawn. This ripple will increase with the ratio of X increments to Y increments necessary to generate the angle.

For instance, if it takes 15 X increments and 4 Y increments to draw the line at the desired angle then the ratio is 15:4, approximately 3:1. However, since each increment is only three percent of the total line width, the total worst deviation from a linear edge for this example is less than nine percent of the line width.

The worst case error can be reduced even further if the

micro-code increments X and Y every time a Y increment (in this example) is needed. Instead of two lines 90 degrees to each other being drawn, one line at 45 degrees will be drawn. This technique will result in a rounding of the edges, thus decreasing the total error. The design will include the capability to increment X and Y simultaneously or independently.

One further consideration for the design is the capability for the Z-80A microprocessor to set up for the next figure while the current figure is being drawn. The controller must have the current high and low address values available to it for the duration of the figure for repeated line drawing (rectangles) and to determine end-of-line and end-of-figure. Consequently a double buffering scheme must be employed.

Design

Since the functions and capabilities of the X and Y deflection boards are the same, the two boards are identical with one exception. The X deflection board (board 4) has the decode logic and the latch for the bit density register. The Y deflection board (board 5) does not require a separate bit density register since the bit density must be the same for both X and Y.

The deflection system was designed to accommodate all of the design considerations discussed previously. The block diagram in Fig. 6-2 shows the system level design. For ease of maintenance and modifiability, all control signals (with

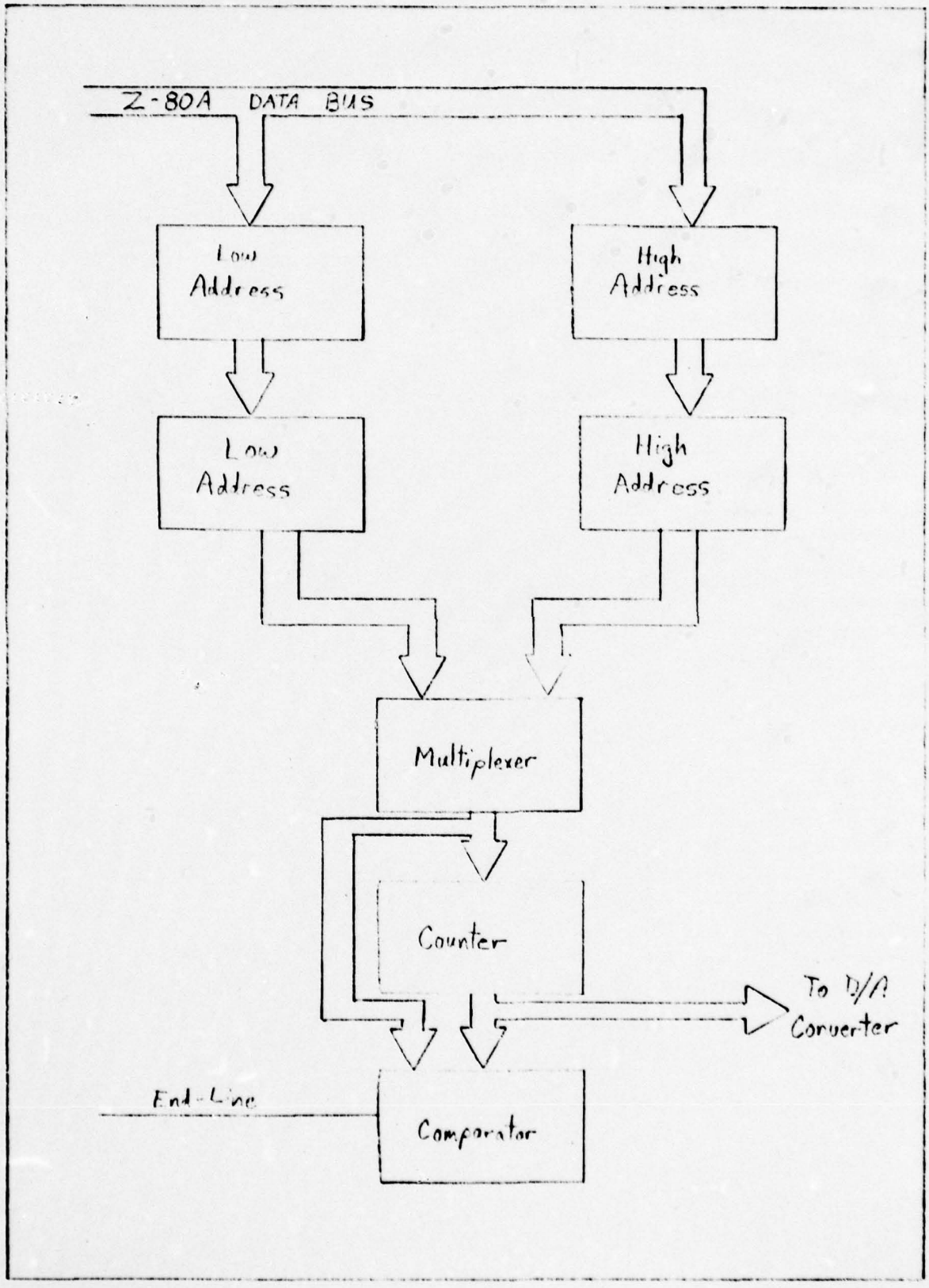


Fig 6-2 Deflection System Block Diagram (X or Y)

the exception of end-line which will be discussed later) were separated from the deflection system and generated on the controller boards. This results in the ability to test the boards independent of the rest of the system, since only the signal function need be checked and not the timing.

The first design consideration addressed was double buffering. The 16 bit low and high addresses are addressed by the Z-80A in pairs, to take advantage of the Z-80A multi-byte input output instructions. The lower 4 bits of the address identify the register, the upper 12 bits identify the bank of memory in which the controller is located. For a further discussion of the memory management technique employed by the controller see Memory Control Register (Chapter 5).

The primary registers are loaded by the Z-80A independent of the Micro-controller. The outputs of the primary register are immediataly available to the secondary registers. A load secondary signal from the Micro-controller gates the new information into the controller's secondary register. At this time the primary registers can be loaded with new information.

Since only one of the available X (or Y) addresses is needed to identify the X (or Y) position, the other address specifies the final X (or Y) position. The multiplexer selects one of the addresses to be loaded in a counter. The multiplexer is then switched to the other address. The output of the multiplexer and the counter are routed to a comparator. The counter is either incremented or decremented until the

counter value equals the other address value. By doing this the counter outputs have defined every point in a line in the $\pm$X direction.

The system requirements called for the ability to select a variable bit density to vary the step size. This is the ability to increment (or decrement) a counter from any of the 5 least significant bits. There was no counter commercially available that could meet these requirements.

A single bit cascadable, n-bit, variable bit density, up/down binary counter was designed. The counter must also be asynchronously loaded. A single element of the counter was designed (See Fig. 6-3), then four of the single elements were cascaded together to form the least significant bits of the 16 bit deflection counters.

J-K Master-Slave flip-flops with data lockout (SN74111) were used to eliminate race around conditions. The flip-flops were configured as toggle flip-flops. The preset and clear inputs were used for the asynchronous load function. Depending on the value of BUFF1, when the load signal was present each flip-flop was either preset or cleared.

In order to vary the bit density, a three bit register was designated to hold the value of the bit position from which to start counting. For instance, 0 means count from bit position 0 (16 bit density), and 3 means count from bit position 3 (13 bit density). When the bit position from which to count is other than zero, the bits lower than the count

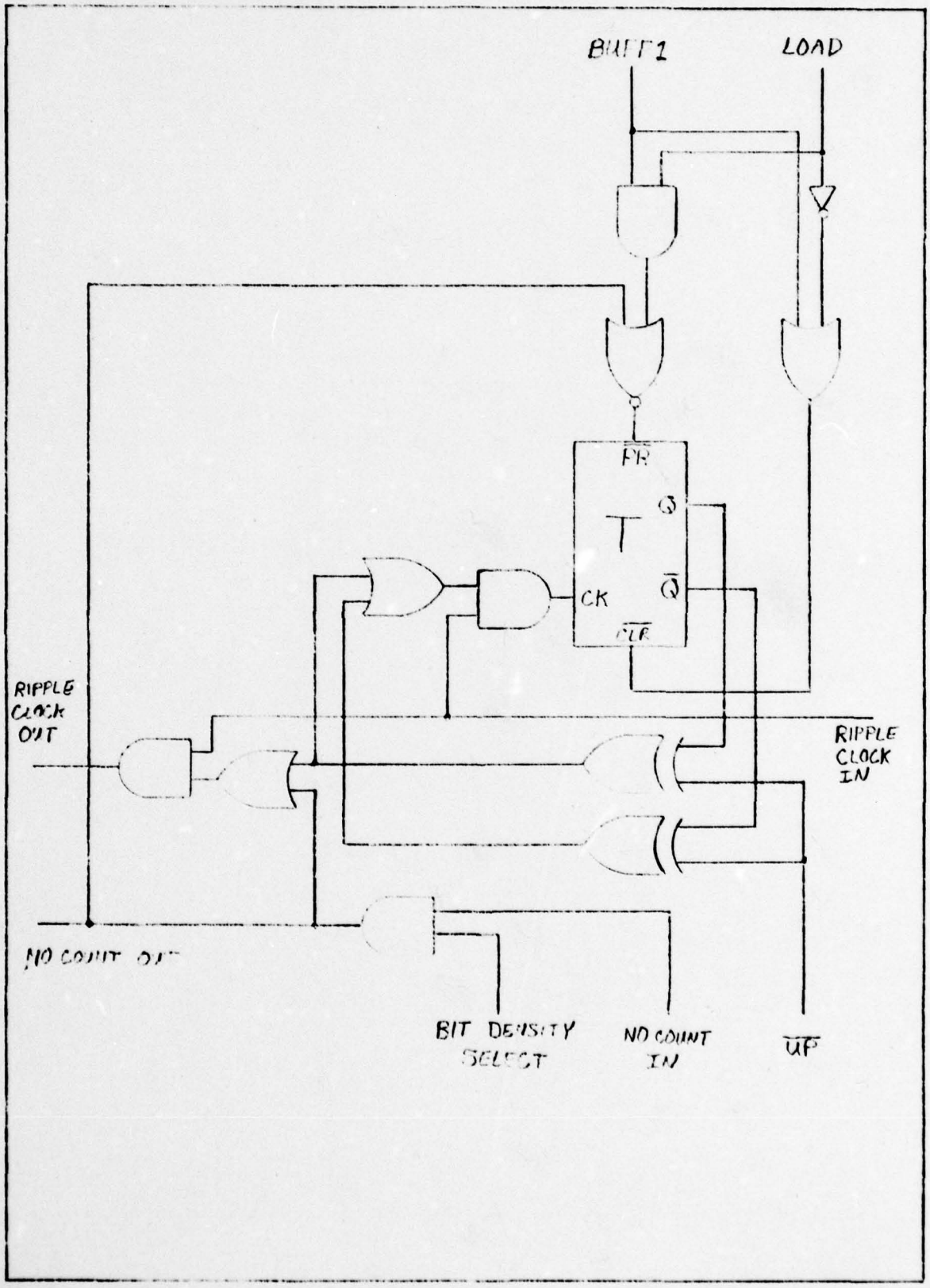


Fig. 6-3 Variable Bit Density Binary UP/Down Counter

bit position are held at logical one.

A three-line-to-eight-line decoder is used to decode the count bit position. The outputs of the decoder are logical ones except for the output line selected which is logical zero. The outputs of the decoder are the bit density select signals. If bit zero is selected, the Y0 output will be logical zero. For bit zero of the counter, the decoder output Y0 is the no-count-out signal. The no-count-out signal for bit zero is the no-count-in signal for bit one. The no-count-out signal bor bit one is the no-count-in signal for bit two, and so on. This signal is also used for two other functions.

When a count bit position greater than zero is selected, the no-count-out signal is used to hold the lower bits (bits lower than the count bit) to a logical one. For instance, if count bit position was 2 then Y2 output of the decoder would be logical zero and all others a logical one. The no-count-out signal for bit one is high. When NORed with the load signal it causes the preset line to flip-flop one to be held low. Consequently the output of flip-flop one is constant at one. When the no-count-out signal is high the clock signal is passed to the next stage. The no-count-in to bit position one is then high, and the bit density select for bit one is high; therefore, the no-count-out is high resulting in bit one being held constant at one and the clock being passed to the next stage. For bit two, the no-count-in is high, but the bit density select is low. This results in the no-count-out signal being

low, signaling that this bit is counting, and not held constant. The low no-count-out signal causes the next stage to have a low count-out signal. Consequently, had the count position been bit one, the no-count-out signals for bits one, two and three would have been low, allowing those bit positions to count.

When a bit position is counting, the clock is passed to the next stage depending on the count. When the flip-flop is zero and counting down the clock is passed. When the flip-flop is zero and counting up, or the flip-flop is one and counting down, the clock is not passed.

In order to make the choice of bit density independent of the data, the data input to the comparator must reflect the bit density selected. The no-count-out signals are ORed with the appropriate data bits from the multiplexer. Consequently, if the bit density is 13 bits, the lower three data bits are equal to the lower three bits of the deflection counter which is held constant at one. Care must be taken that the count position is not greater than the bit positions that vary between the low and high address, otherwise an end of line will be signalled immediately.

Operation

The electron beam column is capable of deflecting the electron beam varying distances, depending on the calibration selected, without moving the sample. The beam can be deflected from 2mm to .75mm on a side. With a bit density of 16 bits,

Bit Density	Increment Length
12	.49 micron
13	.24 micron
14	.12 micron
15	.06 micron
16	.03 micron

a. Increment length for maximum range of 2mm

Bit Density	Increment Length
12	.18 micron
13	.09 micron
14	.04 micron
15	.02 micron
16	.01 micron

b. Increment length for maximum range of .75mm

Table 6-1 Increment length for variable bit density

there are 2^{16}-1 (65,535) descrete points on a side. Table 6-1 demonstrates the step size given the bit density and the maximum range selected during calibration.

The tables demonstrate the range of a single step of the counter given various bit densities and different maximum range of the deflection beam. A device can be defined by points and the actual size of the device drawn can be determined during the calibration sequence by the maximum range which is set by adjusting the gain of the D/A converter.

The bit density for the system is initially set to 16 bits when the system is Reset. Selection of the bit density should not be arbitrary. The bit density combined with the variable clock rate determine the cumulative exposure effects of the beam dwell time. The electron column line width is currently one micron. If the increment length were one micron, the only overlap of the exposure time would be the time it takes to position the beam to the next position. However, if the beam increment length were 0.01 microns, the beam would have to move 100 increments before the beam would no longer have a cumulative exposure effect on the first point. Consequently, the cumulative exposure time would be 100 times the beam dwell time as determined by the variable clock rate.

The beam can be moved in the X or Y direction from 1 to 65,535 steps. A single line of 65,535 increments can take from 0.065535 seconds at the maximum step rate to 1.193 hours at the minimum step rate. The deflection system will also allow

the Z-80A to directly define each point in a figure to be exposed.

The Z-80A must write the X and Y coordinates into the primary registers. Then a program in the Micro-controller must be executed that loads the coordinates into the counters and exposes the resist at that point, then stops. The Z-80A would then enter the next point to be exposed and execute the microprogram again. This would continue until the figure had been completed. If for instance the figure was a line of 65,535 increments, the execution time would be considerably longer than discussed above. Now input/output delays between reading the coordinates from an external device and writing them to the controller registers must be added for each point. This delay could be as much as several milliseconds per point.

Obviously, the delays caused by defining each point can quickly become unreasonable. Where a complex microprogram would be necessary to position the beam, it may prove easier (though perhaps not more efficient) to allow the Z-80A to define each point.

The process of direct writing with the electron beam is a combination of movements, some with the beam on and some with the beam off. It may be necessary to reposition the beam for the next exposure at the opposite end of the field. In order to allow sufficient time for the beam to settle to its true position, a delay register with a down counter was included in the design of the Micro-controller. The delay counter is not

an inherent part of the deflection system but is necessitated by the deflection system. The delay counter is used by the Micro-controller to provide a wait timer for the controller. A delay of up to 40 microseconds could be necessary for a full swing of the electron beam, in order to allow transients to die out. An eight bit down counter, driven by the variable clock, is used to generate the delay. When the down counter reaches zero, an end-delay signal is sent to the Micro-controller. The Micro-controller can thus generate the gross beam movement, delay while the beam settles, and then draw the desired figure, all with a single instruction from the Z-80A.

Conclusion

The deflection system designed is capable of generating any pattern on the substrate. More complex figures, like a circle, would have to be generated one point at a time. However, simple figures, like lines and rectangles, can be defined as vectors and thus generated in one instruction to the Micro-controller. These capabilities are comparable to the commercial pattern generation systems.

Chapter 7 Stage Positioning System

Introduction

The wafers used in fabricating integrated circuits can be as large as four to six inches in diameter (and may eventually become larger). In order to maximize the amount of the wafer that is used, the system must be capable of exposing patterns over as much of the wafer as possible. The maximum total area over which the electron beam can be deflected for pattern writing is $4mm^2$. Obviously, very little of the wafer could be used if the deflection system was the only means available to position the beam on the wafer.

The stage positioning system extends the range over which the wafer can be written. This is accomplished by repositioning the wafer after the deflection system has drawn the $4mm^2$ figure in an individual device area. In order to position the wafer, the stage on which it is clamped is moved. Unless the figures to be drawn can be completely contained within the area the electron beam can be deflected, the stage movement must be as accurate as the beam deflection system. Otherwise lines that cross between areas might not meet.

This chapter will address the design of a stage positioning system. The requirements for the system will be established, then the design based on these requirements will be discussed. Finally, the operation of the hardware will be discussed.

Requirements

The stage for the system under discussion must be capable of positioning the wafer anywhere over a 100 cm^2 area. Additionally, the position must be accurate to within one micron, the approximate line width capability of the system.

The equipment configuration which this controller must drive has a stage that is positioned by direct current stepper motors. The stage can be moved in the $\pm X$ or $\pm Y$ directions. However, the stepper motors are only accurate to 2.5 microns. In order to measure the position of the stage to a sufficient degree of accuracy, a laser interferometer monitors stage position. The laser interferometer is capable of measuring movements of the table in 791 Angstrom ($\lambda/8$) steps. This corresponds to 0.08 microns per step. The additional pattern positioning adjustments employed to compensate for stepper motor step size are discussed below.

The stepper motors move the stage 2.5 microns each time a pulse is received from the controller. The pulse width must be at least one microsecond. The maximum pulse input frequency of the stepper motors is 40 kilohertz.

The laser interferometer system outputs several signals. For each 0.08 micron movement of the stage, either an Up or Down signal is generated (commensurate with the direction of movement of the stage). The pulse width of these pulses is 40 nanoseconds. The laser interferometer output pulses have a maximum frequency of 4.2 megahertz. Error and Restart lines

are also available to monitor status of the laser interferometer device. These are very important since precise stage movement is not possible without the laser interferometer operational.

Since there is a 100mm translational capability, each axis contains 1,250,000 individually addressed points (as defined by the laser interferometer). This corresponds to 1.56×10^{12} discrete points over the surface of the wafer. Any of these discrete address points can be defined by 2 21 bit words, representing the X and Y coordinates of the point.

One pulse to the stepping motors corresponds to a movement of 2.5 microns. Since the laser interferometer measures the position in 0.08mm increments, one stage pulse corresponds to 31.25 laser interferometer pulses. For this reason, the stage cannot be moved to precise (accurate to 1 micron) positions. Therefore, there must be a means of correcting for the error in the stage position. This is accomplished by providing an error counter. This error could be converted to an analog signal, and summed (as a DC offset) with the deflection system.

To facilitate preprocessing by the Z-80A, the stage positioning system must be double buffered. Additionally a stage motion complete signal must be provided so the Z-80A will know when the desired movement is complete.

Design

The functions of the stage positioning system were broken down into the two main functions:

1) stage movement.

2) stage position monitoring and correction.

The stage movement function involves generating pulses, of the appropriate width and frequency, to cause the stage stepper motors to move the stage to the desired location. The stage position monitoring system and correction function interfaces with the laser interferometer system to provide precise positioning of the stage via error correction.

Since the Micro-controller is driven by a variable clock, the Micro-controller could be used to generate the required pulses to stepper motors. However, there are only four test inputs to the microprogram sequencer, and all four are already used. Therefore, the Z-80A software must retain responsibility for determining when a stage movement is complete. The X Stage Complete and Y Stage Complete signals are bits five and six respectively of the controller status register (See Chapter 5). The Z-80A must define a desired stage position and then issue a command to the controller to generate a fixed number of stage pulses. When the pulses have been generated, the Z-80A must check to determine if the current stage position equals the desired stage position (stage movement complete). The process is repeated until stage movement is completed (see Chapter 8, Software Definition.)

The functions of the X stage position monitoring system and the Y stage position monitoring systems are identical. Therefore, only one half of the stage position monitoring system will be discussed, henceforth refered to as the Stage Position Monitoring System. Discussion will refer to both the X and Y systems.

The Stage Position Monitoring System was designed as shown in Figure 7-1. The primary buffers are 24 bit registers loaded as three consecutive bytes of data by the Z-80A. This information corresponds to the next desired stage address. The secondary level is loaded from the primary registers. prior to beginning a stage movement. The actual stage address is a 24 bit counter (6 cascaded SN74193 binary up/down counters) updated by the Up/Down pulses from the laser interferometer.

A comparator is used to compare the desired stage address with the actual address. The "equal" output of the comparator sets a flip-flop which represents stage movement complete. The signal is high when the desired stage address equals the actual stage address. The actual stage address is continuously updated. Consequently it always represents the actual stage position.

An eight bit counter is used to keep track of stage overshoot/undershoot. When the Stage Movement Complete signal is high, the error register must be loaded with Up/Down counts. The Stage Movement Complete signal is ANDed with the Up/Down pulses from the laser interferometer. This causes the error register to count the number of pulses received after the stage

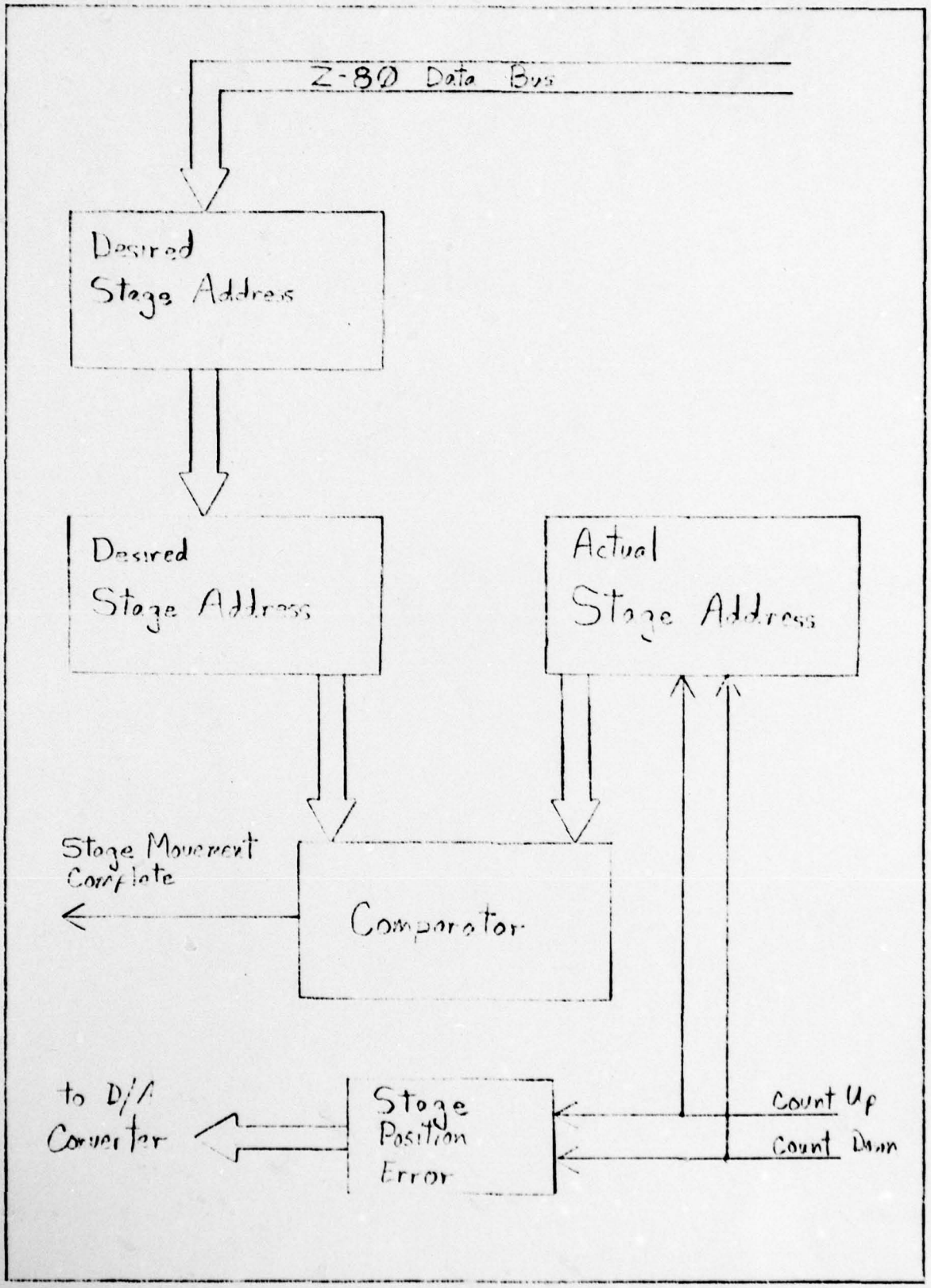


Fig 7-1 Stage Position Monitoring System (X or Y)

motion was complete. This eight bit register corresponds to ±10.24 microns (±128 laser interferometer pulses) of stage position error.

The Up and Down pulse trains are passed to the Down count and Up count inputs of the error register respectively. This automatically generates the correct error signal as follows. If the stage was moving in the +X direction, and the stage movement overshot the desired location by 7 pulses, then to correctly position the beam it must be adjusted back the equivalent of 7 laser interferometer pulses, or -7 pulses. Had motion been in the -X direction when the overshot occurred, the error would be the equivalent of +7 pulses. Switching the Up and Down pulse trains automatically results in the correct sign for the error.

Sometimes DC motors rock back and forth around the commanded shaft position until settled. This circuit design will handle that situation should it arise, since, once the stage movement complete signal is high, the error counter is updated. Also, the stage movement complete flip-flop can only change to not complete by loading the secondary desired stage address with an address different from the actual address.

The error counter is cleared each time a new desired stage address is loaded. Consequently, stage positioning errors should never exceed one imcrement of the stepper motors. The error is converted to a DC offset via a digital-to-analog converter and summed with the analog signal from the appropriate deflection

system. Thus, the error in the stage position is corrected by shifting the entire X and/or Y axis of the deflection system.

Operation

In order to accomplish a stage movement the following steps must be performed:

1) Z-80A clears Micro-controller and sets memory control register so Micro-controller is operational.
2) Z-80A reads command from pattern data.
3) Z-80A outputs desired stage address to primary buffer.
4) Z-80A issues command (instruction) to Micro-controller to execute the load secondary buffer routine.
5) Micro-controller routine generates a "load secondary" signal and halts.
6) Z-80A outputs appropriate variable clock value for stepper motor pulse width.
7) Z-80A issues command to Micro-controller to execute appropriate stepper motor pulse routine.
8) Micro-controller routine generates appropriate stage pulses and halts.
9) Z-80A checks status of stage movement.
   a) If stage movement not complete, sequence loops to step 7 above.
   b) If stage movement complete sequence returns to step 2.

All of the functions discussed above are accomplished by

either the software in the Z-80A or the software in the Micro-controller. See Chapter 8, Software Definition, for more detail on the software drivers required for the system. The operation of the stage position monitoring system will now be discussed relative to the sequence of steps above.

When the Z-80A is powered up and the Reset button is pushed, this results in a clear signal being generated and sent to the Micro-controller. The counters and registers of the stage position monitoring system are all cleared. The development system uses random access memory for micro-store. Consequently, the microprograms must be loaded before the Micro-controller can become operational. In order to insure that all the registers and counters of the system are initialized at zero, after loading micro-store, the Z-80A sets (and then resets) the clear bit in the memory control register. This clears all counters and registers. This step is not repeated once the pattern generation begins since all intermediate information will be lost. This is the only time the stage position error counter is cleared.

During steps two and three, the Z-80A reads a command from the pattern generation data. In this example the command is a move stage command. The desired-stage-position is then read and output to the primary desired-stage register of the stage position monitoring board. Since the secondary register has not yet been loaded, the comparator should still be indicating stage movement complete. Consequently, any pertubations

in the position of the stage will update the error register.

During steps four and five the secondary register is loaded. The Z-80A outputs an instruction to the Micro-controller to execute a load secondary microprogram. This microprogram generates a load signal for the secondary register and then halts. Provided the desired address just loaded in the secondary register is different from the actual register, pertubations in the position of the stage will update the actual stage position. If the secondary register is either not loaded with the new value or the new value equals the actual stage address, the comparator will still indicate stage movement complete. If the routine that causes the stepper motor pulses to be generated and monitors the stage complete signal checks for stage movement complete before requesting a pulse be generated, then stage movement errors can be avoided or at least minimized.

Steps six, seven and eight generate the appropriate stage pulses. An inverted version of state generator D output ($\overline{D}$) is used to generate the stepper motor pulses. If the state generator is operating at maximum speed then the $\overline{D}$ signal duration would be 875 nanoseconds. This is less than the necessary one microsecond. The value of the variable clock should be a 1 which results in a D pulse width of 1.75 microseconds. Therefore the Z-80A will output a one to the primary register of the variable clock. The Z-80A outputs an instruction to the Micro-controller to execute the appropriate stepper motor

pulse routine. This routine would first load the variable clock register to generate the needed pulse width and then generate the appropriate stage pulse(s). $\overline{D}$ is ANDed with the X and Y stage count bits to generate the stepper motor pulses. The X and Y stage motors can be pulsed independently or simultaneously under software control. After the pulse is generated the Micro-controller enters the halt state.

As the stage is moved, the laser interferometer transmits a pulse train to the stage position monitoring board. If the stage movement is not complete, this pulse train updates the actual stage address. Once the stage movement is complete, the pulse train also updates the stage position error counter.

In the last step, the Z-80A determines whether the stage movement is complete. If not complete the Z-80A issues the appropriate instruction to generate another stepper motor pulse then checks stage movement status again. This loop continues until stage movement is complete. At that time the Z-80A will process the next instruction.

Conclusion

The stage positioning system designed is capable of effectively positioning the stage to within one micron. This is accomplished by updating an error counter. The error is converted to a DC offset and used to shift the axis of the entire deflection field after the stage movement is complete. The key to all the elements of the controller is the software.

The development of the software was not within the scope of this thesis. However, the next chapter will address software definition so future efforts can result in software development.

Chapter 8 Software Requirements Definition

Introduction

The software for the Z-80A and the micro-code for the controller are beyond the scope of this thesis. However, no digital controller design is complete without defining the software that will be necessary to operate the equipment. This chapter will address in detail the software requirements definition for both the micro-sequencer and the Z-80A.

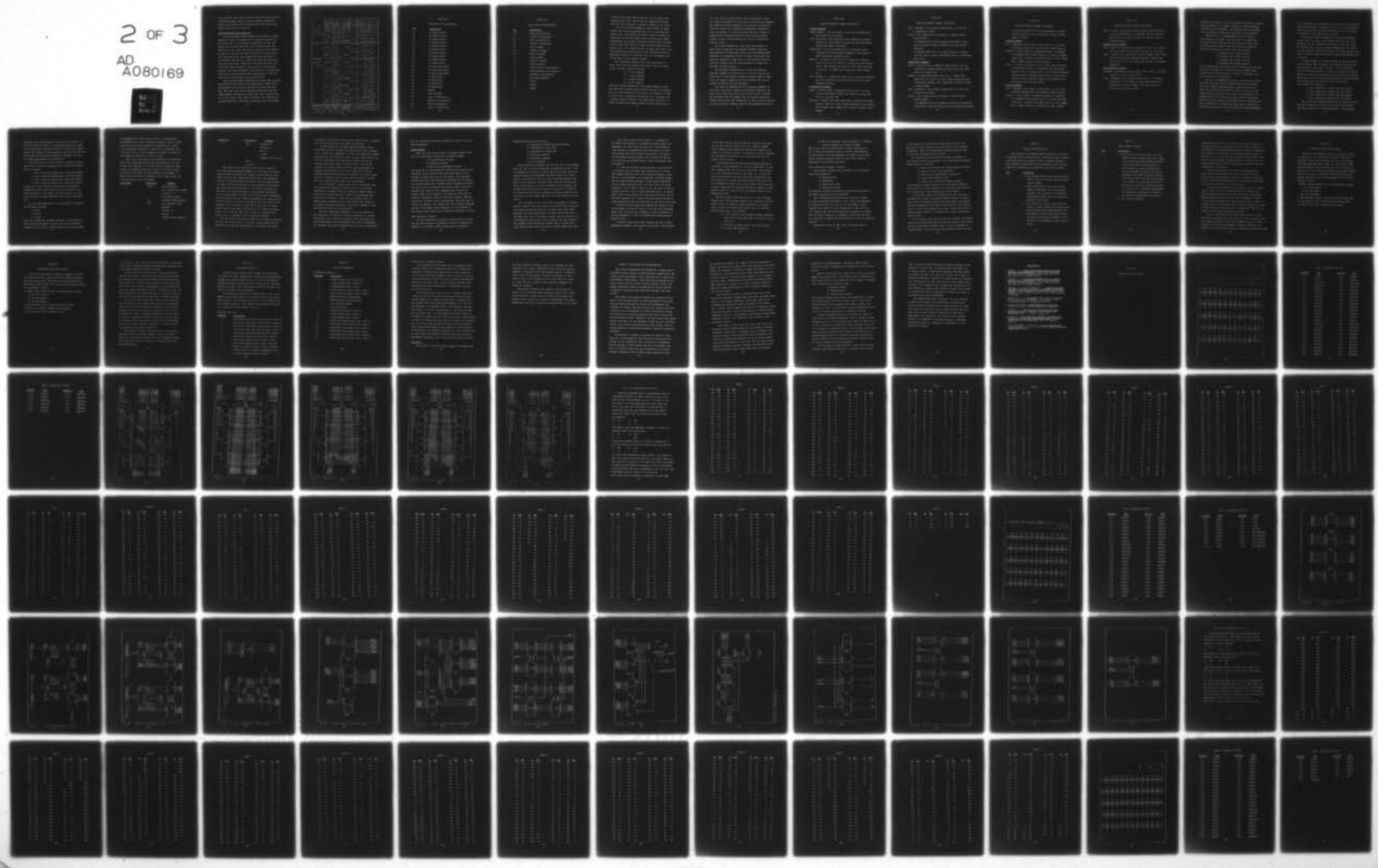
Micro-Sequencer Software

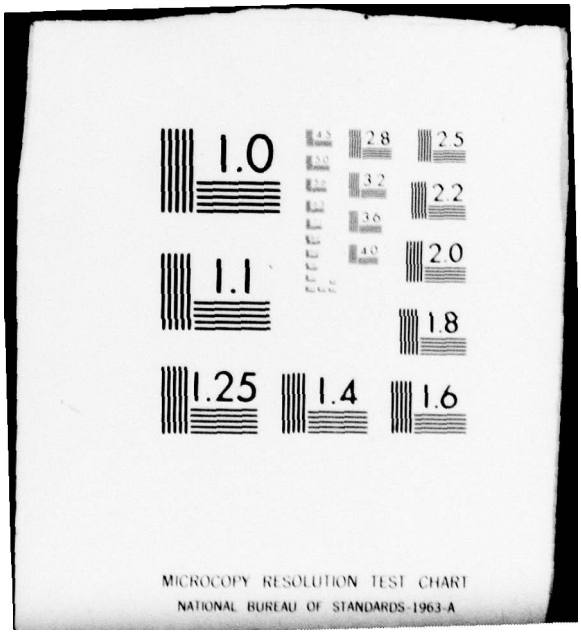
The microprograms of the Micro-controller sequencer are the means by which the register transfers within the controller take place. Table 8-1 lists the Micro-controller instruction set. These are the instructions to the Fairchild 9408 Microprogram/Sequencer. These instructions control sequencing of micro-store by defining conditional and unconditional branches and subroutine calls. Subroutine calls can only be nested four deep, due to the stack limitations of the 9408. However, this should pose no major restrictions on the microprogramming since the hardware is capable of conditional branches in the same micro-cycle.

The instructions, I_0 thru I_3, come from bits 23 thru 26 of micro-store and from the instruction register. As was discussed in Chapter 5, the 9408 receives no clock pulses until the instruction register has been loaded by the Z-80A. Consequently, the hardware cannot sequence without an instruc-

AD-A080 169 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 13/8
DESIGN OF A DIGITAL CONTROLLER FOR AN ELECTRON BEAM LITHOGRAPHY--ETC(U)
DEC 79 M L WEIDNER
UNCLASSIFIED AFIT/GE/EE/79-41 NL

2 OF 3
AD A080169




1.0
1.1
1.25
1.4
1.6
2.8
2.5
3.2
2.2
3.6
2.0
4.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

tion from the Z-80A. This instruction should be from the unconditional branch family to avoid erroneous conditional inputs. The instruction from micro-store can be any of the instructions found in Table 8-1.

Micro-controller Instruction Set

The unconditional branch instructions require a branch address be input to determine the location of the next instruction. For all instructions except BRV3 and BMW, the branch address comes from bits 30-39 of micro-store (See Table 8-2A). BRV3 cause the address register to be selected as input to the 9408. The address register is only 8 bits, and consequently can only address the first 256 words of micro-store. Therefore, to address the rest of micro-store, the next instruction (within the lower 256) that gets executed must cause a branch to the location in micro-store desired. The common approach to this problem is to make the first words in micro-store branch instructions to the routines that reside further down in micro-store.

This technique makes addressing the first seven routines very efficient by using the BMW instruction. Branch multiway replaces the low order three bits with MW0, MW1, and MW2. These three bits allow branching eight ways from the branch address specified in micro-store. Since address zero is the halt state, that leaves seven more words that can be addressed directly. Since this instruction uses the address

	Mnemonic	Definition	I_3-I_2-I_1-I_0	T_3 T_2 T_1 T_0	Description of Operation
Unconditional Branch Instructions	BRV0	Branch Via Micro-Store	0100	X X X X	A_0-A_9 → PC
	BRV1	Branch Via Micro-Store	0101	X X X X	A_0-A_9 → PC
	BRV2	Branch Via Micro-Store	0110	X X X X	A_0-A_9 → PC
	BRV3	Branch Via Address Register	0111	X X X X	AR_0-AR_9 → PC
	BMW	Branch Via Micro-Store Multi-Way	0011	X X X X	MW_0-MW_2 || A_3-A_9 → PC
	BSR	Branch to Subroutine	0001	X X X X	A_0-A_9 → PC & Push the Stack
Conditional Branch Instructions	BXH	Branch on X-Line Complete	1100	X X X H X X X L	IF End X-Line A_0-A_9 → PC IF Not End X-Line PC+1 → PC
	BYH	Branch on Y-Line Complete	1101	X X H X X X L X	IF End Y-Line A_0-A_9 → PC IF Not End Y-Line PC+1 → PC
	BFH	Branch on Figure Complete	1110	X H X X X L X X	IF End Figure A_0-A_9 → PC IF Not End Figure PC+1 → PC
	BDH	Branch on Delay Complete	1111	H X X X L X X X	IF End Delay A_0-A_9 → PC IF Not End Delay PC+1 → PC
	BXL	Branch on X-Line Not Complete	1000	X X X H X X X L	IF End X-Line PC+1 → PC IF Not End X-Line A_0-A_9 → PC
	BYL	Branch on Y-Line Not Complete	1001	X X H X X X L X	IF End Y-Line PC+1 → PC IF Not End Y-Line A_0-A_9 → PC
	BFL	Branch on Figure Not Complete	1010	X H X X X L X X	IF End Figure PC+1 → PC IF Not End Figure A_0-A_9 → PC
	BDL	Branch on Delay Not Complete	1011	H X X X L X X X	IF End Delay PC+1 → PC IF Not End Delay A_0-A_9 → PC
Miscellaneous Instructions	RTS	Return from Subroutine	0000	X X X X	Pop the Stack
	FTCH	Fetch	0010	X X X X	PC+1 → PC

Table 8-1 Micro-Controller Instruction Set

Table 8-2A

Micro-Store Bit Description

BIT	Description
39	A9 Branch Address
38	A8 Branch Address
37	A7 Branch Address
36	A6 Branch Address
35	A5 Branch Address
34	A4 Branch Address
33	A3 Branch Address
32	A2 Branch Address
31	A1 Branch Address
30	A0 Branch Address
29	MW2 Multi-way Branch
28	MW1 Multi-way Branch
27	MW0 Multi-way Branch
26	I3 Instruction
25	I2 Instruction
24	I1 Instruction
23	I0 Instruction
22	Beam On
21	Load Secondary
20	Count (X-Deflection)
19	Up/Down (X-Deflection)
18	Load (X-Deflection)
17	Source (X-Deflection)

Table 8-2A

Micro-Store Bit Description

Bit	Description
16	Count (Y-Deflection)
15	Up/Down (Y-Deflection)
14	Load (Y-Deflection)
13	Source (Y-Deflection)
12	Load (X-Stage)
11	Up/Down (X-Stage)
10	Count (X-Stage)
9	Load (Y-Stage)
8	Up/Down (Y-Stage)
7	Count (Y-Stage)
6	Load Variable Clock Register
5	Clear Variable Clock Register
4	Decrement Delay Register
3	Load Delay Register
2	Unused
1	Unused
0	Unused

in micro-store, the address register does not need to be loaded by the Z-80A, thus saving one memory access cycle each time the controller is issued a command by the Z-80A.

In order to take advantage of this savings in input/output time, the microprograms that will be executed most often by the Z-80A should have their respective branch instructions in the first seven words of micro-store. These instructions can be any of the branch instructions in micro-store, except care should be taken with the branch to subroutine call. If a subroutine call is made, upon return, the next instruction executed will be the instruction that immediately follows the subroutine call. The programmer must be sure that is the sequence desired.

The conditional branches allow the microprogram to query test inputs and branch accordingly. The test inputs to the 9408 are defined as follows:

T_0 - X-Line Complete

T_1 - Y-Line Complete

T_2 - Figure Complete

T_3 - Delay Complete

Since the branch address for a conditional branch is from the same word in micro-store as the instruction, an instruction can conditionally loop on itself. For instance, to draw a line in the X direction, one instruction to increment the X counter could be executed. The instruction would loop on itself by setting the branch address of that instruction to

its own location in micro-store, and executing BXL, branch on X-Line not complete. As long as the X-Line was not complete, the instruction would continue to be executed. As soon as X-Line was complete the instruction immediately following it would be executed. If the next instruction was a branch to address zero, the Micro-controller would set the program counter to zero and halt waiting for the next instruction from the Z-80A.

All of the discussion to this point has centered on micro-store addressing. The register transfers that have been discussed all take place within the 9408 Microprogram/ Sequencer. The remaining 23 bits of micro-store define the register transfers to take place in the controller and combine with signals from the state generator to create the necessary control signals.

The following discussion will address the register transfer commands from Table 8-2B. For the bit position in micro-store of the register transfer commands refer to Table 8-2A. The system commands have been addressed with the exception of Beam On and Load Secondary.

For these two commands and all following commands, except those that select between two different functions (e.g. count up or count down) and clear variable clock, a logical one indicates that the command will be executed during the current micro-cycle. Any combination of items could be executed in one micro-cycle. However, many of the combinations are not

Table 8-2B

Register Transfer Command Definitions

System Commands

Branch Address - Ten bit address of the next instruction in micro-store for execution.

Multi-Way Branch - This three bit field replaces the branch address low order three bits when the multi-way branch instruction (MBR) is executed.

Instruction - Four bit instruction to be executed by the Fairchild 9408 Micro-program Sequencer. (See Table 5-1 Micro-Controller Instruction Set)

Beam On - A logical one in this field causes the electron beam to turn on, or remain on, during the current micro-cycle, thus exposing the resist. A zero causes the beam to turn off, or remain off, during the current micro-cycle.

Load Secondary - A logical one loads the low and high addresses into the respective secondary registers of the X and Y deflection system.

X-Deflection Commands

Count - A single pulse is generated during state E for the X-deflection counter during each micro-cycle when this field is a logical one.

Up/Down - A logical one means count down. A logical zero means count up. This is a level signal supplied to the X-deflection counter. This command has no effect without a count command.

Table 8-2B

Register Transfer Command Definitions

Load - Generates a load signal during state F, to load the X-deflection counter.

Source - This command has two functions. (1=High Address, 0=Low Address)

1) Identifies the source register from which to load the X-deflection counter (either low or high address register).

2) Selects either the low or high address to compare with the output of the X-deflection counter to determine End X-Line.

Y-Deflection Commands

Count - A single pulse is generated during state E for the Y-deflection counter during each micro-cycle when this field is a logical one.

Up/Down - A logical one means count down. A logical zero means count up. This is a level signal supplied to the Y-deflection counter. This command has no effect without a count command.

Load - Generates a load signal, during state F, to load the Y-deflection counter.

Source - This command has two functions. (1=High Address, 0=Low Address)

1) Identifies the source register from which to load the Y-deflection counter (either low or high address register).

Table 8-2B

Register Transfer Command Definitions

2) Selects either the low or high address to compare with the output of the Y-deflection counter to determine End Y-line.

X-Stage Commands

Load - Generate a load signal, during state E, to load the secondary register with the desired X-stage location.

Up/Down - A level signal to determine whether to increment or decrement the X-stage position. A logical one means count down. A logical zero means count up. This command has no effect without a count command.

Count - A single pulse whose width is seven times the period of the state generator is generated each micro-cycle when this bit is a logical one. (Note: The stage stepper motors require a minimum one microsecond pulse width.) Therefore seven times the value of the variable clock must be at least one microsecond.

Y-Stage Commands

Load - Generate a load signal, during state E, to load the secondary register with the desired Y-Stage location.

Up/Down - A level signal to determine whether to increment or decrement the Y-stage position. A logical one means count down. A logical zero means count up. This command has no effect without a corresponding count command.

Table 8-2B

Register Transfer Command Definitions

Count - A single pulse, whose width is seven times the period of the state generator, is generated each micro-cycle when this bit is a logical one (see note for count of X-Stage Commands).

Variable Clock Commands

Load - Causes the Variable Clock Register (VCR), to be loaded from the primary clock buffer with the new clock rate.

Clear - Logical zero causes the VCR to be cleared. Once cleared the clock runs at the maximum frequency. Logical one allows the clock to be loaded, and operated at variable frequencies.

Delay Register Commands

Decrement - Generates a count signal during state E. The delay counter is wired to count down.

Load - Generates a load signal during state F. A decrement command and a load command in the same micro-cycle is equivalent to a load command. The delay constant is loaded into the delay counter.

reasonable together. In order to present everything in context, correct sequences of commands necessary to perform certain functions will be presented. Any anomalies that could arise will be presented within the discussion of the sequences.

The first step in pattern generation on a wafer is positioning of the wafer. Once the wafer is clamped onto the stage, this is accomplished by moving the stage. For a discussion of the stage positioning system see Chapter 7. The microprograms must be able to do the following:

1) Increment the X-Stage Position
2) Decrement the X-Stage Position
3) Increment the Y-Stage Position
4) Decrement the Y-Stage Position

As a minimum these four stage movement routines are necessary. The steps necessary to accomplish any one of these tasks differ only in the direction and the axis; therefore, only one routine will be presented.

In order to increment the X-Stage position, the X-Stage secondary register (desired stage address) must be loaded from the X-Stage primary register. During the same micro-cycle, the variable clock must be loaded with a one so the stepper motor pulse width is at least one microsecond. The instruction for the first micro-cycle of the microprogram should be a fetch during the next micro-cycle, the Up/Down for the X-Stage must be a logical zero and Count a logical one. This will result in a count up X-Stage pulse being generated.

The instruction for that micro-cycle would be an unconditional branch to the address specified in bits 30-39 of the current word in micro-store (provided the routine was not written as a subroutine). The address would be zero.

Hence, each time the routine was executed the desired address would be loaded, the variable clock set, the pulse generated, and the sequence would return to halt all within six microseconds. Since the maximum frequency of the pulses is four kilohertz, efforts to optimize the stage routine would serve no purpose unless micro-store storage requirements could be reduced.

A stage movement is usually followed by the positioning of the electron beam. The beam will often be moved from one field extreme to the other. In order to allow sufficient time for the beam movement to settle, the deflection counters should be loaded during execution of the stage pulse routine. There are a minimum of four load deflection routines differing in the deflection axis and the source for the load. The routines are as follows:

1) Load X-deflection counter from high address
2) Load X-deflection counter from low address
3) Load Y-deflection counter from high address
4) Load Y-deflection counter from low address

Any of the loads can be accomplished in one micro-cycle. The load secondary bit must be set to load the secondary registers of the X and Y deflection boards. The secondary

registers are loaded during the rising edge of state E. On the rising edge of state F, the selected address is loaded in the appropriate deflection counter. On the rising edge of state G the outputs of the deflection counters are gated into the buffers of the digital-to-analog converters. The load will take approximately one microsecond.

Since only one word of micro-store is needed for each type of load, the four combinations of X and Y loads should also be included.

1) Load X from high address and Y from high address
2) Load X from high address and Y from low address
3) Load X from low address and Y from high address
4) Load X from low address and Y from low address

During none of the previous routines should the Beam On command have been active. This would have resulted in erroneous exposure of the wafer. The beam should be on only while a pattern is being generated, and off for everything else.

As a minimum there should be microprograms to generate the following patterns:

1) point
2) X-Line
3) Y-Line

This is a minimum set of simple functions. Combinations of these can draw any figure. However, for more versatility routines to draw lines of various angles should be included.

For instance a 45 degree line is drawn by simultaneously incrementing the X and Y deflection counters. Lines of other angles would have to be made up of increments of X and Y. A microprogram to generate a X-line will be presented. The Y-line and lines of various angles are simply extensions of the techniques employed to draw an X-line.

Since every line drawn is not preceded by a stage movement, the routine may include the load sequence (or a call to the load sequence). If the beam position change is large enough, the delay register can be used to loop while doing nothing until the beam position settles. For instance, to draw a line from the X-low address to the X-high address the following steps would be necessary:

Micro-Cycle	Instruction	Commands
1	Fetch	Load Secondary
		Load X-deflection counter
		X Source = 0
		Load delay counter
2	BDL	Decrement delay register
3	BXH	Load variable clock register
		X Source = 1
		Beam On
		Variable clock clear = 1

Micro-Cycle	Instruction	Commands
4	BXL	X Source = 1
		Up/Down = 0
		Beam On
		X Count
		Variable clock clear = 1
5	BRVO	
	(or RTS)	

The routine listed above will draw either a point or an X-line. The micro-cycle number is to show only the ordered steps and not necessarily the number of micro-cycles it will take to draw a line. During micro-cycle one, the deflection counter is loaded from the low address, and the delay counter is loaded. The next instruction to be executed is micro-cycle two. This instruction branches to the address specified in that word as long as the delay is not complete. The address is set to the address of that word in micro-store. Consequently, the instruction loops on itself decrementing the delay register. When the delay is complete, micro-cycle three is executed. The variable clock register is loaded which means the new clock period begins with state G. The beam is also turned on during state G to expose the wafer. The X source is selecting the high address to compare to the X-deflection counter.

The BHX instruction says to branch to the address specified in this word if end X-line. To expose a single point, the low address would equal the high address and therefore the point

is complete. The address in that word would be zero. Consequently, the next micro-cycle would halt the machine.

If the counter did not equal the high address, micro-cycle four would be executed. The high address would continue to be selected for comparison. Up/Down would equal zero signalling count up. The beam would stay on, and X Count would be on. The BXL instruction would cause this command line to be executed every micro-cycle as long as the X line was not complete. The branch address for this word would equal the address in micro-store of this word. When the X line was complete micro-cycle five would be executed. This word should turn off X Count and Beam On, and cause a branch to address zero or a return if the routine was a subroutine.

It is important to notice that every microprogram eventually halts by returning to address zero in micro-store. Therefore, no commands should be enabled at address zero. A zero word at address zero will insure the appropriate halt state of the controller. Note that the variable clock is cleared when the machine halts. This enables the controller to operate at maximum frequency except when commanded to vary the beam dwell time in a microprogram.

The Micro-controller sequencer is the main element of the controller. It takes the pattern information or position information, and performs the assigned task. Obviously, the Micro-controller is not independent. The Z-80A is responsible for providing the pattern information and position information,

and for instructing the Micro-controller of what to do with this information.

Z-80A Software

There are three major groups of software routines that need to be developed to make this a complete system.

1) Data generation routines
2) Interface routines
3) Micro-code development routines

The discussion of data generation routines will address only the format of the various items since the physics of the problem is beyond the scope of this thesis. The interface routines and micro-code development routines will be discussed in more detail since certain ordered sequences of events must take place to accomplish the desired tasks. These discussions will center on the interface steps and not the actual coding of the routines. No attempt has been made to specify a language in which to write the routines. However, some driver routines would be considerably more efficient if written in assembly code. Neither the Z-80A assembly language nor any of the higher level languages will be addressed in this thesis since several good sources are available for each.

Data Generation Routines

The data generation routines are responsible for creating the data necessary to generate a pattern or sequence of patterns on the wafer. These routines must be capable of

accomplishing the following tasks:

1) Interactive interfacing with the user
2) Computation of dwell time
3) Computation of delay time
4) Defining a pattern
5) Repeating a pattern
6) Error checking

The interactive interface with the user must, at a minimum, allow the user to input coordinates identifying the point, line or rectangle to be drawn. It would obviously be very time consuming to input the coordinates for each figure to be drawn. Ideally, the user would sit at a high resolution graphics terminal with a light pen. The user would place the light on the screen cooresponding to the new location of the crosshair. The routines would then query the user to determine if a line was to be drawn between the old crosshair position and the new one. If a line was to be drawn, then the dwell time routine would be called.

The computation of the dwell time is dependent on several factors. First of all, the routine would have to have the exact dwell time equation for the type of resist used since these equations can vary dramatically. The beam current and resist thickness would have to be known in some cases, as well as, the line width and depth desired. The routine would then complete the exposure time required and convert that to a beam dwell time. The following formula is used to compute beam dwell time:

dwell time = (dwell time desired) - 1 microsecond. The dwell time desired is a sixteen bit binary value. Consequently, the dwell time can vary from 1 microsecond to 0.065 seconds in increments of one microsecond. If dwell times larger than 0.065 seconds are required, the figure can be redrawn again and again, since exposure of the resist does have an addative effect, until the exposure time is achieved.

If, when the crosshair had been moved, a line was not to be drawn, then the delay time routine would be called. The maximum settling time for the digital-to-analog converters is 35 microseconds. The routine must determine whether the last beam movement warrants a delay before generating the new figure from that point, and the size of the delay necessary. If the Z-80A is going to move the beam to the new location and set up for a figure from that point, the input/output time of the routine will likely be sufficient. Use of the delay papameter is to allow beam positioning followed by figure generation with one instruction to the Micro-controller.

The basic figure generation elements are defined by the microprograms, but should consist of at least the following; point, line, 45^{o}-line, and rectangle. For the discussion of pattern generation, it will be assumed that these microprograms do exist.

A routine should exist that can group the basic figure generation elements (with relative coordinates) into patterns.

In this way, various patterns that are repeated many times on the wafer need be defined only once. Another command would reference that pattern by some key and request it be drawn at a specific location. The addresses would be modified, by the relative addresses, to absolute addresses and the figure would be drawn.

This capability of defining and repeating patterns is conducive to the generation and use of a shape menu. In this way, commonly used cells could be predefined. For instance, 2, 3 and 4 input gates such as AND, NAND, OR, NOR and XOR could be predefined. The user could then simply select the cell desired and identify the connections to the cell. This is a simplified discussion of a very complex task. Firm conclusions concerning the feasability and desirability of this approach cannot be drawn at this preliminary stage but the capability exists.

The last routine that must exist is an error checking routine. This error checking routine must check not only for invalid data, for instance, alpha characters in a numeric field, but also for errors that might result in erroneous figures being drawn. For instance,

1) Is the current vector within the beam deflection field or will the stage have to be moved to complete it?
2) Does the new figure cross a previous figure?
3) Are all lines connected?

4) Have all parameters been initialized, or defined, prior to requesting a figure be drawn?

This list is not all inclusive but is meant rather as an aid to point out possible sources of error. The generation of valid data is only the beginning. The data is of no value unless interface routines exist for the Z-80A to get the information to the Micro-controller.

Micro-controller Interface Routines

The interface routines can be grouped in the following four major categories:

1) Initialization
2) Diagnostics
3) Micro-store Loader
4) Pattern Generation

The sequence of events necessary to perform the functions in the major categories will be discussed.

The initialization routine has one function, setting all Micro-controller registers to zero. This is accomplished by setting the Reset bit (bit two) in the Memory Control Register (See Table 8-3) to a logical zero. While this bit is a logical zero, all Micro-controller registers are cleared and the entire controller is disabled. Setting the bit back to a logical one enables the Micro-controller with all registers cleared.

This must be done at least twice. The first time is

during power-up, to insure the system starts cleared. The next time is after loading micro-store and prior to becoming operational. This is to insure the secondary registers have no erroneous information left in them.

The diagnostic routines should perform a thorough on-line check of the Micro-controller. As a minimum the following checks should be included:

1) Write to micro-store and verify what was written
2) Exercise X and Y stage movements
3) Verify laser interferometer operational
4) Exercise the variable clock
5) Verify X and Y deflection analog signals

The diagnostic routines should verify the correct operation of as much of the Micro-controller as possible. When this has been accomplished, the microprograms can be loaded.

In the final design of the Micro-controller, the micro-programs will reside in read-only memory. However, to facilitate program development during the design phase, micro-store has been implemented with random access memory. Since both the Z-80A and the Micro-controller must both access micro-store, a memory control register (See Table 8-3) is used to arbitrate memory access.

The Z-80A does not access micro-store directly. An address buffer and a micro-store data register are filled by the Z-80A. The Micro-controller variable clock is used to generate the timing signals. The micro-store data register is the only data

Table 8-3

Memory Control Register

The purpose of the Memory Control Register is two-fold. It provides first the micro-store write/write protect signals, and second the reset signal to clear the Micro-controller. This register is addressed as an input/output port. The port number is determined by an eight bit dip switch on Board 2.

Bit	Description
0	0 - Micro-store address comes from the address buffer. Used to load and edit micro-store (Read or Write).
	1 - Micro-store address comes from address outputs of Fairchild 9408 Microprogram Sequencer. The controller is considered "operational" when a one is in this bit position. (Read only)
1	0 - If bit zero is a logical zero, the next signal to access memory will cause a write from the micro-store data register into micro-store, at the address specified in the micro-store address register. If the controller is operational, this bit has no effect.

Table 8-3

Memory Control Register

Bit	Description
	1 - If bit zero is a logical zero, the next signal to access memory will cause a read from micro-store at the address specified in the micro-store address register.
2	0 - Reset signal. As long as this bit is a zero all register storage in the Micro-controller is cleared. The first step is to get the Z-80A to set this bit position to a logical one. After micro-store has been loaded and prior to setting bit zero to a logical one (controller operational) the controller must again be cleared so no erroneous information is processed.
	1 - The clear is disabled.

source for micro-store. However, the micro-store address can come from the address buffer or from the address outputs of the Fairchild 9408 Microprogram Sequencer. These two addresses are multiplexed by bit zero of the memory control register. When bit zero is a logical zero, the micro-store address comes from the address buffer. When bit zero is a logical one, the address comes from the 9408. A logical one indicates the Micro-controller is operational.

Once the Micro-controller becomes operational micro-store should not be altered. In order to insure the integrity of the microprograms, bit zero also serves as the write protect. When the Micro-controller is operational micro-store is read-only, otherwise, it is read-write.

Bit one of the memory control register is used to signal either a read or a write. This bit combined with bit zero provides the capability for the Z-80A to access micro-store. Tables 8-4A and 8-4B present the steps necessary to write to or read from micro-store respectively. Note that in order to verify a write to micro-store, the Z-80A must perform the steps in Table 8-4A and then simply read in registers 0 thru 4, comparing what was read with what was written.

Once the microprograms are loaded the last step is the generation of patterns from the data file. These programs should be a direct extension of the data generation routines. These routines must fill the appropriate controller registers (See Table 8-5) with the pattern data and command the Micro-controller

Table 8-4A

Z-80A Micro-Store Write Routine

The following sequence of steps is required to write to micro-store. When the register address is decoded it generates a memory access signal. In this case a memory write has been requested. On the next full cycle of the Micro-controller clock, the data word will be loaded in micro-store at the address specified in the address buffer. The data will also be latched in the micro-store output buffer. This buffer can be read directly by the Z-80A as registers 0 thru 4.

1) Memory Control Register must be configured as follows:
   i) bit zero equals 0
   ii) bit one equals 0
   iii) bit two equals 1
2) Load register 0 thru 4 with the micro-store data word.
3) Load register 5 and 6 with the micro-store address.
4) Output to register 7 (memory access signal).

Table 8-4B

Z-80A Micro-Store Read Routine

The following sequence of steps is required to read a word from micro-store. The memory access signal generates a read from micro-store, which latches the data word in the micro-store output buffer. The Z-80A reads the buffer directly as registers 0 thru 4.

1) Memory Control Register must be configured as follows:
   i) bit zero equals 0
   ii) bit one equals 1
   iii) bit two equals 1
2) Load registers 5 and 6 with micro-store address.
3) Output to register 7 (memory access signal).
4) Read data word from registers 0 thru 4.

to process the data. Since the Micro-controller is double buffered, these routines can begin loading buffers again as soon as the Micro-controller starts executing.

However, the Z-80A can only write to the instruction register (register 6) after the Micro-controller has halted. The instruction register must be the last register written into because the decoded write signal sets a flip-flop which starts the 9408 executing. Consequently, if the instruction was loaded before the pattern data was completely set up, the Micro-controller would begin with the old data and would not get the new data which was loaded after it began executing.

This routine must monitor the status of the Micro-controller (Address zero is halt state) in order to determine when to write to the instruction register of the Micro-controller. In most cases, the Micro-controller will be done executing the current instruction before the Z-80A is ready to load the instruction. However, there will be times when the current pattern could take several seconds to generate, in which case the Z-80A would just have to execute an idle loop waiting for the Micro-controller.

The routines discussed are all that will be necessary once the system is complete. However, during the design phase, the exact micro-code has not been established. Therefore tools must exist for the user to design and develope the microprograms.

Table 8-5

Controller Registers

The first eight register write signals are multiplexed to provide the eight signals needed during the non-operational loading and editing of micro-store. Bit zero of the control register determines whether the write signal is for the operational registers or the micro-store edit/load registers (See Table 5-4, Memory Control Register).

The controller registers are memory mapped on the Z-80A memory. The register number is the four low order bits of the memory address. The 12 high order bits are user definable via dip switches on board 2. This enables the controller registers to be configured at any address up to 65 k.

Edit/Load Registers

Registers	Description
0	Micro-store data input register bits 32 thru 39
1	Micro-store data input register bits 24 thru 31
2	Micro-store data input register bits 16 thru 23
3	Micro-store data input register bits 8 thru 15
4	Micro-store data input register bits 0 thru 7
5	Micro-store address register bits 8 thru 9
6	Micro-store address register bits 0 thru 7
7	Micro-store memory access signal. When this signal is decoded, either a read or a write of micro-store is generated depending on bit one of the memory control register.

Table 8-5

Controller Registers

Operational Registers

Registers	Description
0	X-Stage address bits 16 thru 23
1	X-Stage address bits 8 thru 15
2	X-Stage address bits 0 thru 7
3	X-Deflection low address bits 8 thru 15
4	X-Deflection low address bits 0 thru 7
5	Branch Address in Micro-memory
6	Instruction and Multiway branch D_0-D_2 Multiway branch D_4-D_7 Instruction
7	Y-Stage Address bits 16 thru 23
8	Y-Stage Address bits 8 thru 15
9	Y-Stage Address bits 0 thru 7
10	X-Deflection high address bits 8 thru 15
11	X-Deflection high address bits 0 thru 7
12	Y-Deflection low address bits 8 thru 15
13	Y-Deflection low address bits 0 thru 7
14	Y-Deflection high address bits 8 thru 15
15	Y-Deflection high address bits 0 thru 7

Micro-code Development Routines

At a minimum, editor routines must be available which will allow the user to input the micro-code and make modifications. Preferably the routine would either prompt the user for the individual element in the word and set the correct bit pattern, or allow the user to specify what elements must be set for that word. In this way the developer will not have to key in 40 binary bits of information for each word of micro-store.

The routines should also provide a display capability so the user can see the line of code and correct it if necessary before it is loaded in micro-store. A special listing routine should also provide the capability to list the microprograms in a readable format. For instance, line numbers should be provided corresponding to the address of the word in micro-store. The line number and address should be converted to decimal for the listing since that is much easier to read. The instruction code could be converted to the mnemonic so the user would not have to refer to the instruction table. Appropriate headings would suffice for the remainder of micro-store. The binary bit patterns for that section could be listed under the headings. These routines would provide reasonable development tools for the micro-program developer.

Conclusion

The software discussed in this chapter is necessary for

the total controller package. Much of the software has been addressed only lightly. Considerable time and effort will be required to design and code the software modules. A separate effort should be undertaken to design the modules for this control system. This chapter should be used only as a source for ideas, and as a reference where specific sequences of events are required.

This chapter completes the design of the digital controller for the electron beam lithography system. This design effort addressed a major portion of the electron beam lithography system. There are a number of improvements and design enhancements that could increase the capabilities of the system.

Chapter 9 Conclusions and Recommendations

This thesis demonstrates the capability to employ microprocessor control systems in very high speed applications. The intent during the design of this controller was to exceed all performance specifications. Since this pattern generation system is used by the Air Force to verify industry claims, the higher operational speeds and capabilities will be required to generate sub-micron line widths and spacing in the near future.

The design of the digital hardware was completed by this thesis. The variable clock, the state generator and the variable bit density deflection counter were breadboarded and verified the designs. The three boards that comprise the Micro-controller were wire-wrapped. The Zilog microcomputer system delivered did not contain sufficient card slots to mount the Micro-controller boards, therefore, the boards were never tested. Another card-cage will be necessary to mount all the controller boards. Since the Micro-controller was never tested, the following discussion of system performance is based on engineering estimates.

The hardware is capable of stepping the beam at a maximum rate of one megahertz. The beam can be moved from 0.01 microns to 0.49 microns per step, depending on calibration and bit density. At the maximum step rate, and maximum step size, a 20 micron square pad can be generated in 0.8 milliseconds. Depending on the software delays required to load

the controller registers and command the Micro-controller to process the data, device that can be drawn inside the 20 micron square (for instance, a transistor) might take from 0.6 to 1.5 milliseconds. Each deflection field ($4mm^2$ approximately 10,000 20 micron squares) could take approximately 6 to 15 seconds. A four inch wafer (approximately 2,000 $4mm^2$ deflection fields) could take from 3.8 hours to 8.8 hours. The last figure is based on a stage positioning time to an adjacent deflection field of one second.

These figures are based on maximum step rate. If the resist used required a longer beam dwell time, the above times would be even longer. Obviously, these times would not be reasonable for production runs. The effort of this thesis was to produce system capabilities and not a production system, since the Air Force is not in the business of manufacturing integrated circuits. With advanced technology producing faster electronic devices, the controller could be made to meet production speeds.

At the time of this design the factor that limited the bandwidth of this control system was the digital-to-analog converter. Sixteen bits of accuracy with a conversion rate (voltage output) of one microsecond was not available. Any efforts to increase the bandwidth of the controller should begin with the design of a very fast, precise, and accurate digital-to-analog converter. However, the bandwidth of the Micro-controller is four times faster than the bandwidth

specified in the requirements. Therefore, efforts should first be devoted to enhancing the capabilities of the current system.

There are three areas for future study that would greatly enhance the capabilities of the electron beam lithography system. Each of these three areas below would require a combination of hardware and software design.

1) Graphics Package

2) Auto-alignment

3) Variable Beam Current

The graphics package would provide the capability to display the pattern either while it is being drawn or take the raw data and generate the pattern on the screen. Ideally this package would have a sophisticated editing capability to allow the user to modify the data by modifying the display.

The auto-alignment package would enable the system to align itself for pattern generation. This is normally a time consuming task of 15 to 20 minutes each time the system is to be run. In order to accomplish the task, hardware would have to be designed to interface with the video signal of the scanning electron microscope. This hardware would also have to automatically adjust the offset and gain of the digital-to-analog converters for the deflection system. Software could be used to sequence the auto-alignment.

The last area of study would add a great deal of flexibility to the patterns that could be drawn by the electron

beam. Varying the beam current adds another dimension to the capability to draw lines of varying widths and depths, and at various speeds. In some cases increasing the beam current will enable the line to be drawn at a faster rate, thus increasing throughput. Decreasing the beam current will decrease the splash around the beam, thus lines of finer widths could be drawn. This capability will increase the complexity of the data generation routines since another variable will be added to the equations. The data generation software would have to be addressed with this enhancement.

The software described in Chapter 8 was not developed as part of this thesis. The enhancements discussed above assume that this software will have been designed and coded. Since the Micro-controller cannot operate without the software, these systems should be designed and coded before enhancements are considered. The design of this software package is recommended as a graduate project. A thorough, independent design will enhance the capabilities of this system even further.

Bibliography

1. Doherty, J. A. Electron Beam Fabrication of Surface Acoustic Wave Transducers. Griffiss Air Force Base, New York. 1977—. (Technical Report RADC-TR-77-164 for Rome Air Development Center).

2. Doherty, J. A. Electron Beam Fabrication of Surface Acoustic Wave Transducers. Griffiss Air Force Base, New York. 1976—. (Technical Report RADC-TR-76-289 for Rome Air Development Center).

3. Livesay, W. R., and Friedmann, E. B. Scanning Electron Beam Mask Generation System for Integrated Circuit Production. Wright Patterson Air Force Base, Ohio. 1977—. (Technical Report AFML-TR-76-162 for Air Force Materials Laboratory).

4. Ludwig, M. P., ed. IC Master. Santa Clara, California: United Technical Publications, Incorporated. 1978—. (A complete guide to integrated circuits).

5. Mostek Corporation. Microcomputer Z-80 Data Book. Carrollton, Texas: Mostek Corporation. August 1978.

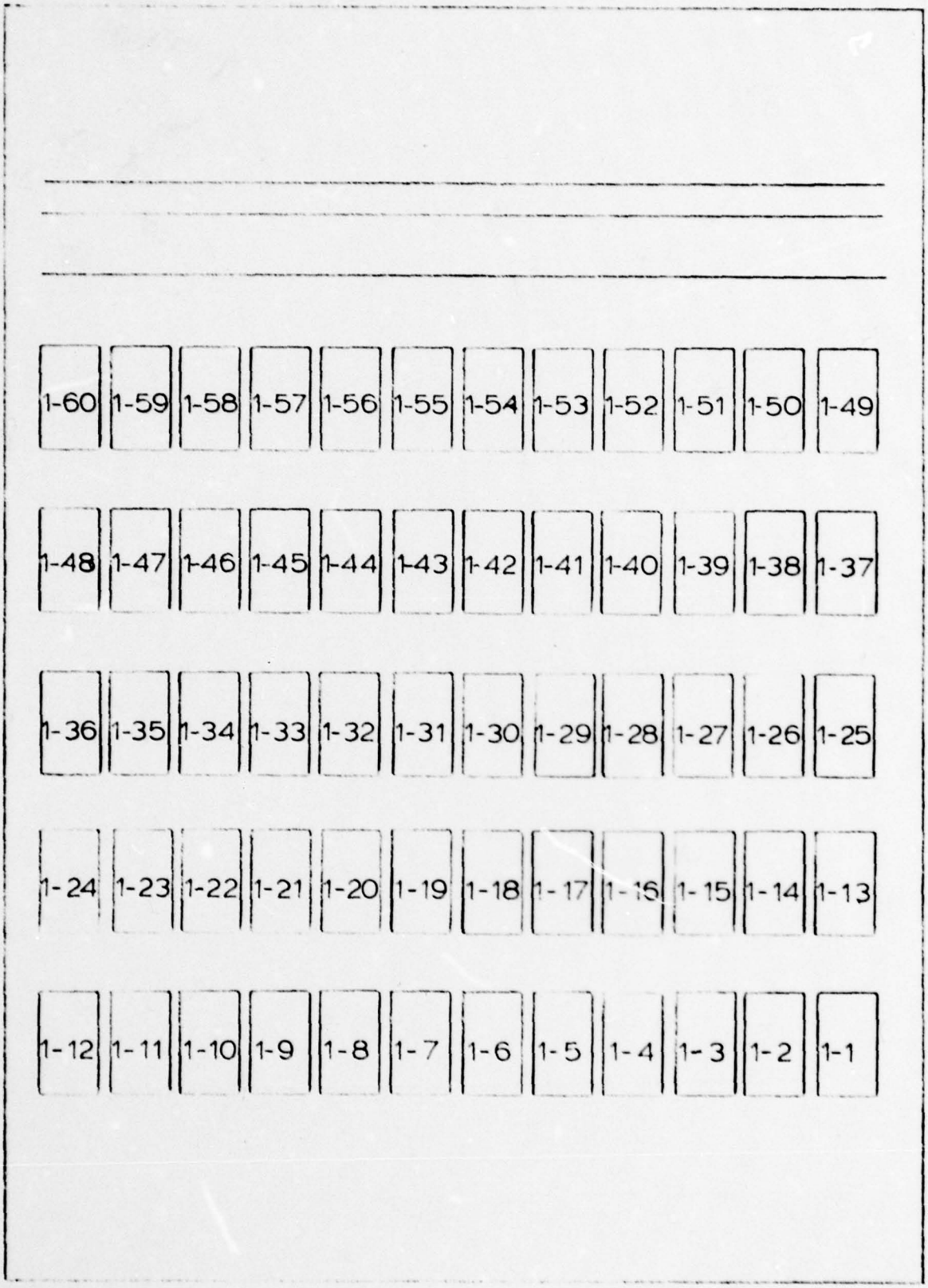
6. Pattach, A. M. Electron-beam Lithographic Pattern Generation System. Journal of Vacuum Science and Technology, Vol. 15, Number 3. May 1978.

7. Stephani, D. High-precision Automatic Alignment Procedure for Vector Scan E-beam Lithography. Journal of Vacuum Science and Technology, Vol. 15, Number 3. May 1978.

8. Texas Instruments Corporation. The TTL Data Book for Design Engineers. 2nd ed. Dallas, Texas: Texas Instruments Incorporated, 1976.

Appendix A

Controller Circuit Layouts


1-60 1-59 1-58 1-57 1-56 1-55 1-54 1-53 1-52 1-51 1-50 1-49
1-48 1-47 1-46 1-45 1-44 1-43 1-42 1-41 1-40 1-39 1-38 1-37
1-36 1-35 1-34 1-33 1-32 1-31 1-30 1-29 1-28 1-27 1-26 1-25
1-24 1-23 1-22 1-21 1-20 1-19 1-18 1-17 1-16 1-15 1-14 1-13
1-12 1-11 1-10 1-9 1-8 1-7 1-6 1-5 1-4 1-3 1-2 1-1

Board 1 Integrated Circuits

Position	Type	Position	Type
1-1	SN74175	1-25	SN74175
1-2	SN74175	1-26	SN74175
1-3	SN74LS367	1-27	SN74LS367
1-4	SN74LS367	1-28	SN74LS367
1-5	MM2102AJ	1-29	MM2102AJ
1-6	MM2102AJ	1-30	MM2102AJ
1-7	MM2102AJ	1-31	MM2102AJ
1-8	MM2102AJ	1-32	MM2102AJ
1-9	MM2102AJ	1-33	MM2102AJ
1-10	MM2102AJ	1-34	MM2102AJ
1-11	MM2102AJ	1-35	MM2102AJ
1-12	MM2102AJ	1-36	MM2102AJ
1-13	SN74175	1-37	SN74175
1-14	SN74175	1-38	SN74175
1-15	SN74LS367	1-39	SN74LS367
1-16	SN74LS367	1-40	SN74LS367
1-17	MM2102AJ	1-41	MM2102AJ
1-18	MM2102AJ	1-42	MM2102AJ
1-19	MM2102AJ	1-43	MM2102AJ
1-20	MM2102AJ	1-44	MM2102AJ
1-21	MM2102AJ	1-45	MM2102AJ
1-22	MM2102AJ	1-46	MM2102AJ
1-23	MM2102AJ	1-47	MM2102AJ
1-24	MM2102AJ	1-48	MM2102AJ

Board 1 Integrated Circuits

Position	Type	Position	Type
1-49	SN74175	1-55	MM2102AJ
1-50	SN74175	1-56	MM2102AJ
1-51	SN74LS367	1-57	MM2102AJ
1-52	SN74LS367	1-58	MM2102AJ
1-53	MM2102AJ	1-59	MM2102AJ
1-54	MM2102AJ	1-60	MM2102AJ

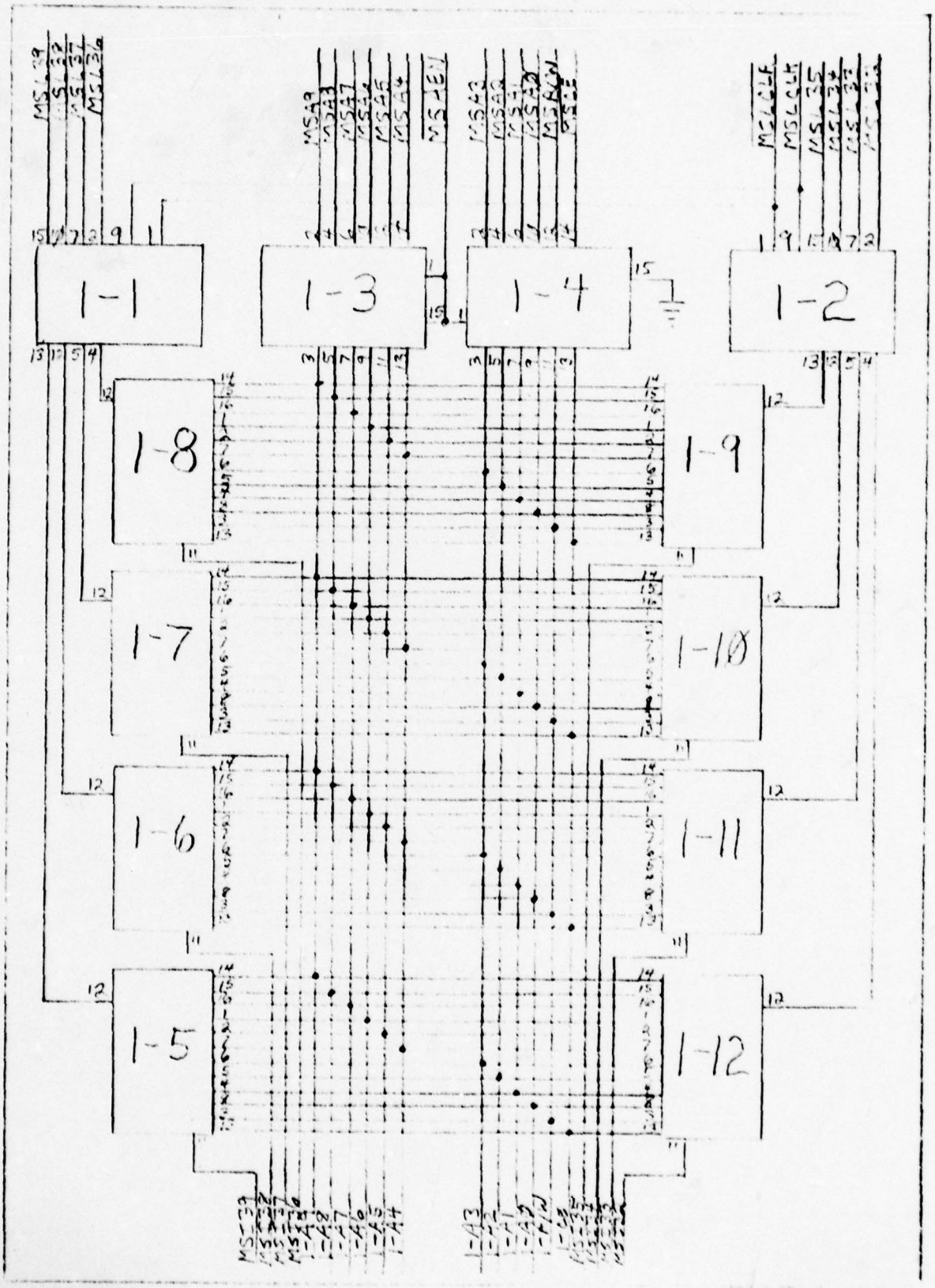


Fig 1 Micro-Stac Bits 32 thru 39

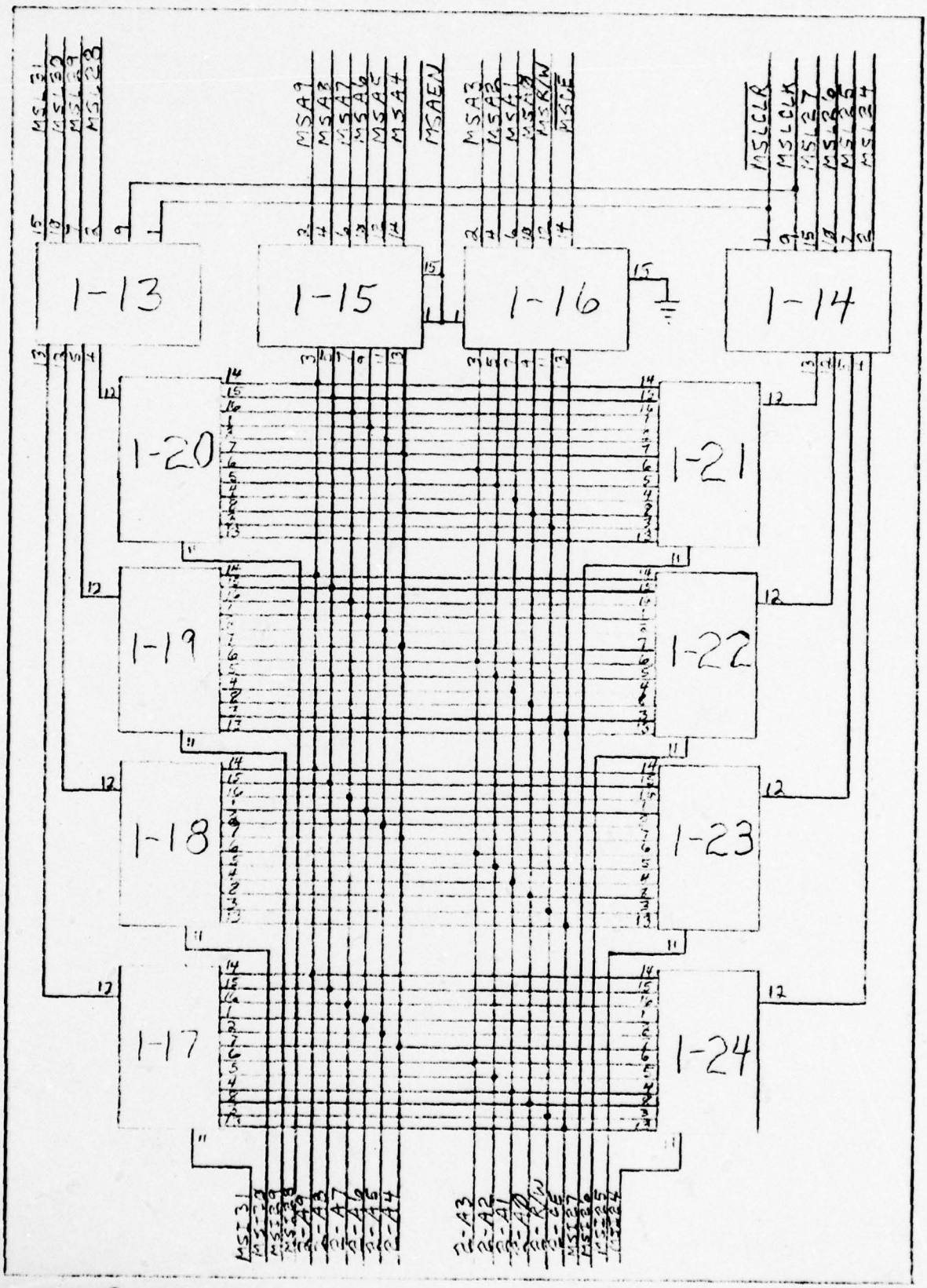


Fig 2 Micro-Store Bits 24 thru 31

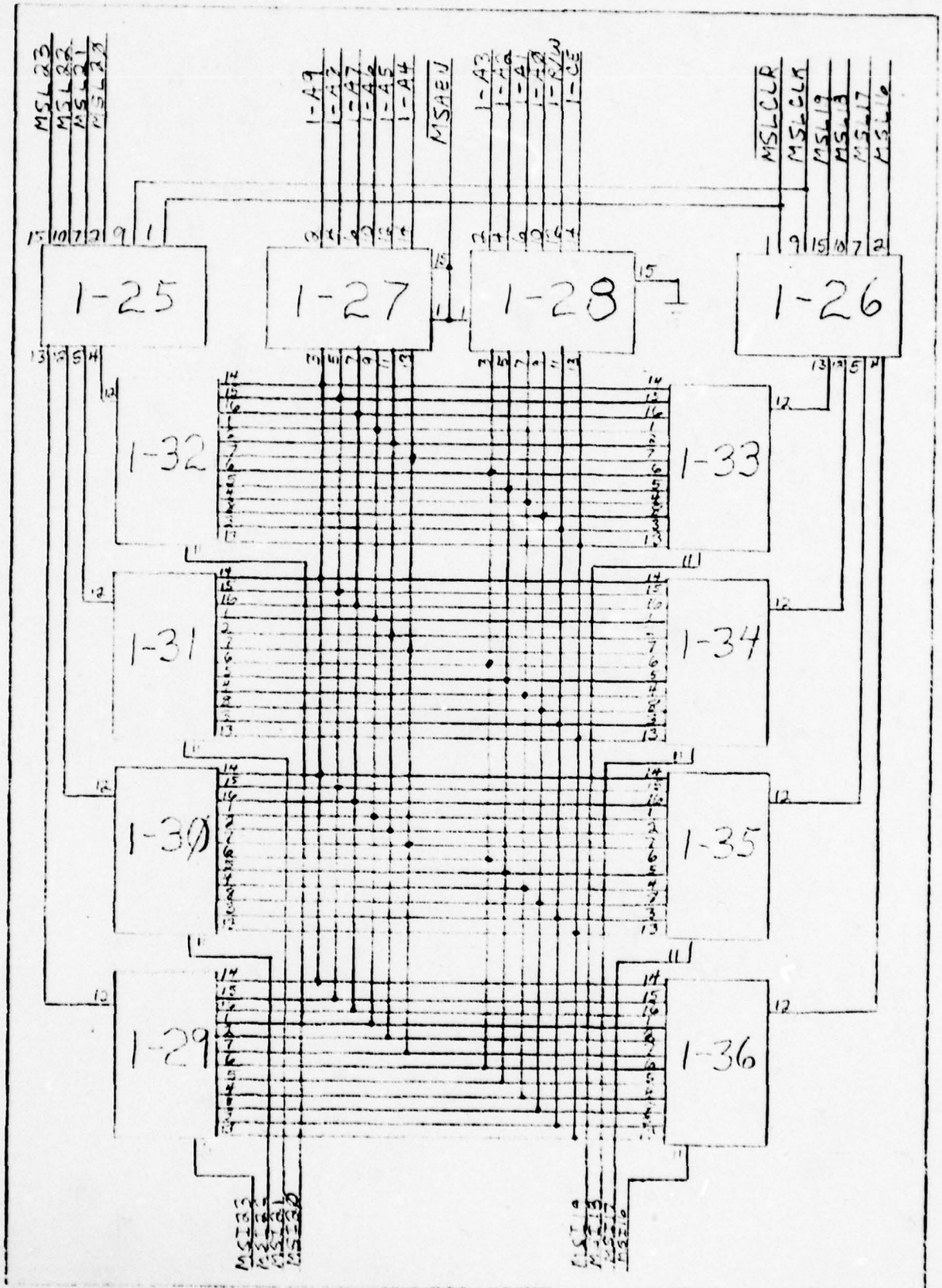


Fig 3 Micro-State Bits 16 thru 23

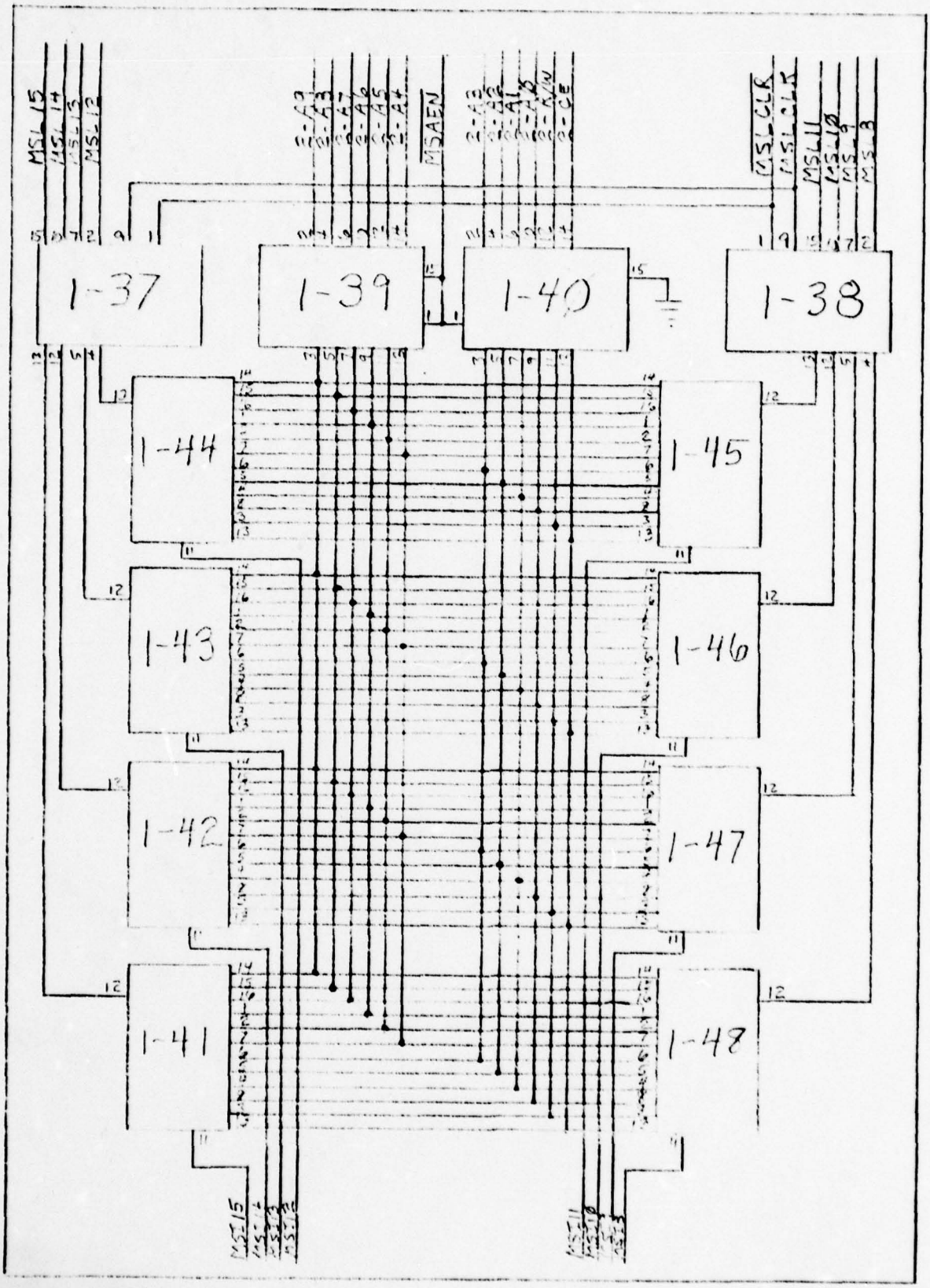


Fig 4 Micro-Store Bits 8 thru 15

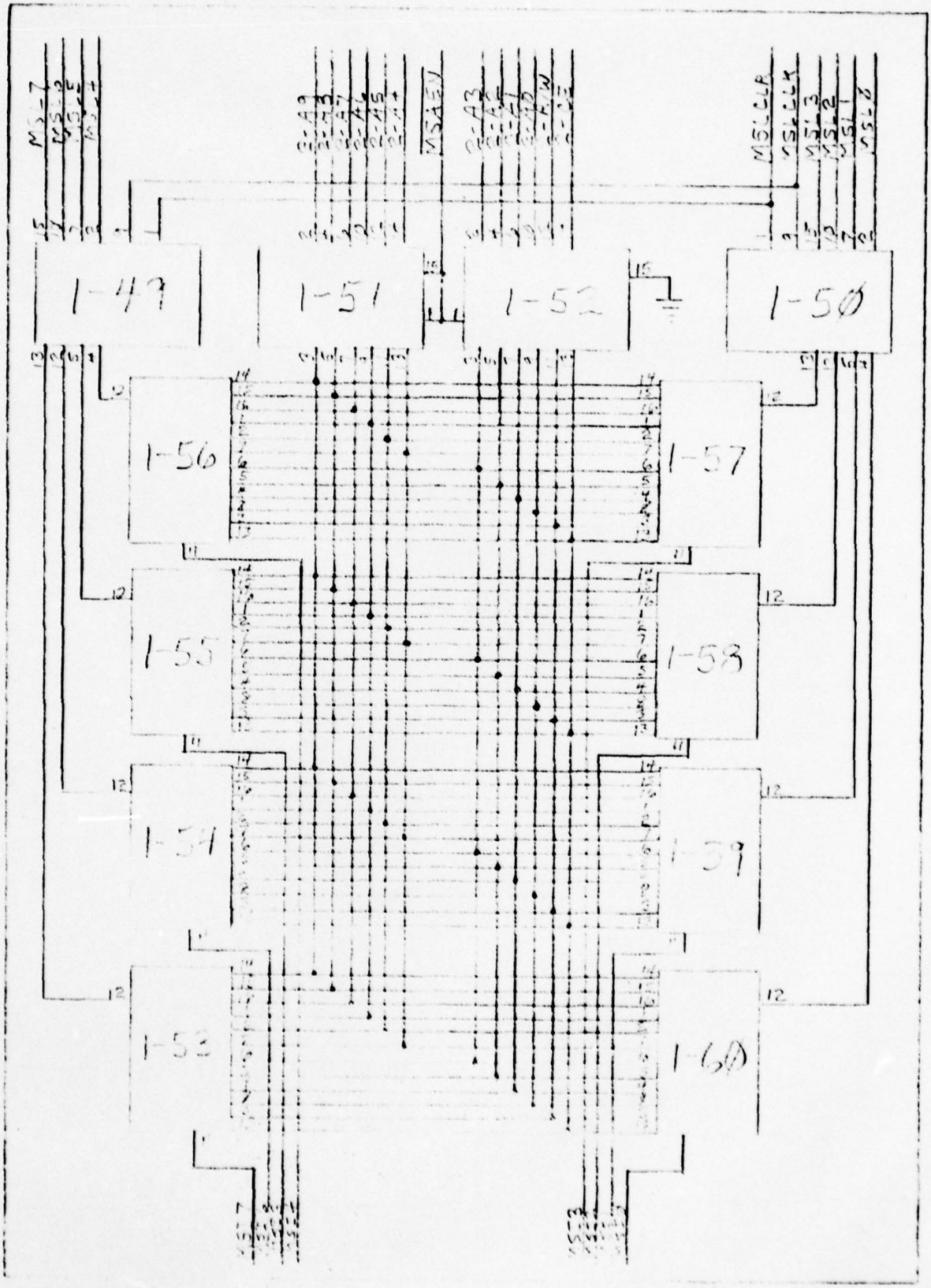


Fig 5 Micro Slice Bits Ø thru 7

Pin to Pin Connections for Board 1

The Zilog Wire-wrap Board is preconfigured with 16 pin sockets with the V_{cc} Plane connected to pin 16 and with the GND Plane connected to pin 8. In order to use the memory, the connecting plane was etched away from the V_{cc} and GND pins, electrically isolating them. A power and ground bus was connected to all the memory using wire wraps. Note the following examples to read the Appendix.

IC	PIN	IC	PIN
5	9	11	6

This means a wire was physically connected to these two uniquely defined wire wrap posts.

IC	PIN	IC	PIN
5	7		A28

A wire was connected from IC 5 at pin 7 to Column A of the strip on the top of the wire wrap board pin number 28.

IC	PIN	IC	PIN
5	14		68

IC 5 pin 14 was connected to Edge connector pin number 68. The 14 pin dips are mounted with pin 1 in socket number one. This requires 2 be added to the right side (8-14) pin number to arrive at the adjusted pin number for the 16 pin socket. This has already been done. Consequently, you will note some connections to pins 15 and 16 of 14 pin dips.

Note: There are two columns of connections on each page.

Board 1

IC	PIN	IC	PIN	IC	PIN	IC	PIN
35	14	36	14	35	15	36	15
35	16	36	16	35	1	36	1
35	2	36	2	35	7	36	7
35	6	36	6	35	5	36	5
35	4	36	4	35	8	36	8
35	3	36	3	35	13	36	13
33	14	34	14	33	15	34	15
33	16	34	16	33	1	34	1
33	2	34	2	33	7	34	7
33	6	34	6	33	5	34	5
33	4	34	4	33	8	34	8
33	3	34	3	33	13	34	13
31	14	32	14	31	15	32	15
31	16	32	16	31	1	32	1
31	2	32	2	31	7	32	7
31	6	32	6	31	5	32	5
31	4	32	4	31	8	32	8
31	3	32	3	31	13	32	13
29	14	30	14	29	15	30	15
29	16	30	16	29	1	30	1
29	2	30	2	29	7	30	7
29	6	30	6	29	5	30	5
29	4	30	4	29	8	30	8
29	3	30	3	29	13	30	13

Board 1

IC	PIN	IC	PIN	IC	PIN	IC	PIN
30	14	31	14	30	15	31	15
30	16	31	16	30	1	31	1
30	2	31	2	30	7	31	7
30	6	31	6	30	5	31	5
30	4	31	4	30	8	31	8
30	3	31	3	30	13	31	13
32	14	33	14	32	15	33	15
32	16	33	16	32	1	33	1
32	2	33	2	32	7	33	7
32	6	33	6	32	5	33	5
32	4	33	4	32	8	33	8
32	3	33	3	32	13	33	13
34	14	35	14	34	15	35	15
34	16	35	16	34	1	35	1
34	2	35	2	34	7	35	7
34	6	35	6	34	5	35	5
34	4	35	4	34	8	35	8
34	3	35	3	34	13	35	13
36	14	27	3	36	15	27	5
36	16	27	7	36	1	27	9
36	2	27	11	36	7	27	13
36	6	28	3	36	5	28	5
36	4	28	7	36	8	28	9
36	3	28	11	36	13	28	13

Board 1

IC	PIN	IC	PIN	IC	PIN	IC	PIN
11	14	12	14	11	15	12	15
11	16	12	16	11	1	12	1
11	2	12	2	11	7	12	7
11	6	12	6	11	5	12	5
11	4	12	4	11	8	12	8
11	3	12	3	11	13	12	13
9	14	10	14	9	15	10	15
9	16	10	16	9	1	10	1
9	2	10	2	9	7	10	7
9	6	10	6	9	5	10	5
9	4	10	4	9	8	10	8
9	3	10	3	9	13	10	13
7	14	8	14	7	15	8	15
7	16	8	16	7	1	8	1
7	2	8	2	7	7	8	7
7	6	8	6	7	5	8	5
7	4	8	4	7	8	8	8
7	3	8	3	7	13	8	13
5	14	6	14	5	15	6	15
5	16	6	16	5	1	6	1
5	2	6	2	5	7	6	7
5	6	6	6	5	5	6	5
5	4	6	4	5	8	6	8
5	3	6	3	5	13	6	13

Board 1

IC	PIN	IC	PIN	IC	PIN	IC	PIN
6	14	7	14	6	15	7	15
6	16	7	16	6	1	7	1
6	2	7	2	6	7	7	7
6	6	7	6	6	5	7	5
6	4	7	4	6	8	7	8
6	3	7	3	6	13	7	13
8	14	9	14	8	15	9	15
8	16	9	16	8	1	9	1
8	2	9	2	8	7	9	7
8	6	9	6	8	5	9	5
8	4	9	4	8	8	9	8
8.	3	9	3	8	13	9	13
10	14	11	14	10	15	11	15
10	16	11	16	10	1	11	1
10	2	11	2	10	7	11	7
10	6	11	6	10	5	11	5
10	4	11	4	10	8	11	8
10	3	11	3	10	13	11	13
12	14	3	3	12	15	3	5
12	16	3	7	12	1	3	9
12	2	3	11	12	7	3	13
12	6	4	3	12	5	4	5
12	4	4	7	12	8	4	9
12	3	4	11	12	13	4	13

Board 1

IC	PIN	IC	PIN	IC	PIN	IC	PIN
3	1	27	1	3	3	27	2
3	5	27	4	3	7	27	6
3	9	27	10	3	11	27	12
3	13	27	14	3	15	27	15
3	1	3	15	4	1	28	1
4	3	28	2	4	5	28	4
4	7	28	6	4	9	28	10
4	11	28	12	4	13	28	14
27	1	28	1	28	15	28	8
4	15	4	8	23	14	24	14
23	15	24	15	23	16	24	16
23	1	24	1	23	2	24	2
23	7	24	7	23	6	24	6
23	5	24	5	23	4	24	4
23	8	24	8	23	3	24	3
23	13	24	13	21	14	22	14
21	15	22	15	21	16	22	16
21	1	22	1	21	2	22	2
21	7	22	7	21	6	22	6
21	5	22	5	21	4	22	4
21	8	22	8	21	3	22	3
21	13	22	13	19	14	20	14
19	15	20	15	19	16	20	16
19	1	20	1	19	2	20	2

Board 1

IC	PIN	IC	PIN	IC	PIN	IC	PIN
19	7	20	7	19	6	20	6
19	5	20	5	19	4	20	4
19	8	20	8	19	3	20	3
19	13	20	13	17	14	18	14
17	15	18	15	17	16	18	16
17	1	18	1	17	2	18	2
17	7	18	7	17	6	18	6
17	5	18	5	17	4	18	4
17	8	18	8	17	3	18	3
17	13	18	13	18	14	19	14
18	15	19	15	18	16	19	16
18	1	19	1	18	2	19	2
18	7	19	7	18	6	19	6
18	5	19	5	18	4	19	4
18	8	19	8	18	3	19	3
18	13	19	13	20	14	21	14
20	15	21	15	20	16	21	16
20	1	21	1	20	2	21	2
20	7	21	7	20	6	21	6
20	5	21	5	20	4	21	4
20	8	21	8	20	3	21	3
20	13	21	13	22	14	23	14
22	15	23	15	22	16	23	16
22	1	23	1	22	2	23	2

Board 1

IC	PIN	IC	PIN	IC	PIN	IC	PIN
22	7	23	7	22	6	23	6
22	5	23	5	22	4	23	4
22	8	23	8	22	3	23	3
22	13	23	13	24	14	15	3
24	15	15	3	24	16	15	7
24	1	15	9	24	2	15	11
24	7	15	13	24	6	16	3
24	5	16	5	24	4	16	7
24	8	16	9	24	3	16	11
24	13	16	13	15	3	39	2
15	5	39	4	15	7	39	6
15	9	39	10	15	11	39	12
15	13	39	14	15	1	15	15
39	1	39	15	39	1	28	1
15	1	4	1	16	1	15	15
39	15	40	1	40	15	40	8
16	15	16	8	51	1	51	15
51	1	40	1	51	15	52	1
52	15	52	8	47	14	48	14
47	15	48	15	47	16	48	16
47	1	48	1	47	2	48	2
47	7	48	7	47	6	48	6
47	5	48	5	47	4	48	4
47	8	48	8	47	3	48	3

Board 1

IC	PIN	IC	PIN	IC	PIN	IC	PIN
47	13	48	13	45	14	46	14
45	15	46	15	45	16	46	16
45	1	46	1	45	2	46	2
45	7	46	7	45	6	46	6
45	5	46	5	45	4	46	4
45	8	46	8	45	3	46	3
45	13	46	13	43	14	44	14
43	15	44	15	43	16	44	16
43	1	44	1	43	2	44	2
43	7	44	7	43	6	44	6
43	5	44	5	43	4	44	4
43	8	44	8	43	3	44	3
43	13	44	13	41	14	42	14
41	15	42	15	41	16	42	16
41	1	42	1	41	2	42	2
41	7	42	7	41	6	42	6
41	5	42	5	41	4	42	4
41	8	42	8	41	3	42	3
41	13	42	13	42	14	43	14
42	15	43	15	42	16	43	16
42	1	43	1	42	2	43	2
42	7	43	7	42	6	43	6
42	5	43	5	42	4	43	4
42	8	43	8	42	3	43	3

Board 1

IC	PIN	IC	PIN	IC	PIN	IC	PIN
42	13	43	13	44	14	45	14
44	15	45	15	44	16	45	16
44	1	45	1	44	2	45	2
44	7	45	7	44	6	45	6
44	5	45	5	44	4	45	4
44	8	45	8	44	3	45	3
44	13	45	13	46	14	47	14
46	15	47	15	46	16	47	16
46	1	47	1	46	2	47	2
46	7	47	7	46	6	47	6
46	5	47	5	46	4	47	4
46	8	47	8	46	3	47	3
46	13	47	13	48	14	39	3
48	15	39	5	48	16	39	7
48	1	39	9	48	2	39	11
48	7	39	13	48	6	40	3
48	5	40	5	48	4	40	7
48	8	40	9	48	3	40	11
48	13	40	13	40	2	16	3
40	4	16	5	40	6	16	7
40	10	16	9	40	12	16	11
40	14	16	13	39	2	51	2
39	4	51	4	39	6	51	6
39	10	51	10	39	12	51	12

Board 1

IC	PIN	IC	PIN	IC	PIN	IC	PIN
39	14	51	14	40	2	52	2
40	4	52	4	40	6	52	6
40	10	52	10	40	12	52	12
40	14	52	14	59	14	60	14
59	15	60	15	59	16	60	16
59	1	60	1	59	2	60	2
59	7	60	7	59	6	60	6
59	5	60	5	59	4	60	4
59	8	60	8	59	3	60	3
57	14	58	14	57	15	58	15
57	16	58	16	57	1	58	1
57	2	58	2	57	7	58	7
57	6	58	6	57	5	58	5
57	4	58	4	57	8	58	8
57	3	58	3	55	14	56	14
55	15	56	15	55	16	56	16
55	1	56	1	55	2	56	2
55	7	56	7	55	6	56	6
55	5	56	5	55	4	56	4
55	8	56	8	55	3	56	3
55	13	56	13	53	14	54	14
53	15	54	15	53	16	54	16
53	1	54	1	53	2	54	2
53	7	54	7	53	6	54	6

Board 1

IC	PIN	IC	PIN	IC	PIN	IC	PIN
53	5	54	5	53	4	54	4
53	8	54	8	53	3	54	3
53	13	54	13	54	14	55	14
54	15	55	15	54	16	55	16
54	1	55	1	54	2	55	2
54	7	55	7	54	6	55	6
54	5	55	5	54	4	55	4
54	8	55	8	54	3	55	3
54	13	55	13	56	14	57	14
56	15	57	15	56	16	57	16
56	1	57	1	56	2	57	2
56	7	57	7	56	6	57	6
56	5	57	5	56	4	57	4
56	8	57	8	56	3	57	3
56	13	57	13	58	14	59	14
58	15	59	15	58	16	59	16
58	1	59	1	58	2	59	2
58	7	59	7	58	6	59	6
58	5	59	5	58	4	59	4
58	8	59	8	58	3	59	3
58	13	59	13	60	14	51	3
60	15	51	5	60	16	51	7
60	1	51	9	60	2	51	11
60	7	51	13	60	6	52	3

Board 1

IC	PIN	IC	PIN	IC	PIN	IC	PIN
60	5	52	5	60	4	52	7
60	8	52	9	60	3	52	11
60	13	52	13	49	1	50	1
49	9	50	9	49	13	53	12
49	12	54	12	49	5	55	12
49	4	56	12	50	13	57	12
50	12	58	12	50	5	59	12
50	4	60	12	13	1	14	1
13	9	14	9	1	1	2	1
1	9	2	9	25	1	26	1
25	9	26	9	37	1	38	1
37	9	38	9	1	1	13	1
1	9	13	9	14	1	26	1
14	9	26	9	25	1	37	1
25	9	37	9	38	1	50	1
38	9	50	9	13	13	17	12
13	12	18	12	13	5	19	12
13	4	20	12	14	13	21	12
14	12	22	12	14	5	23	12
14	4	24	12	1	13	5	12
1	12	6	12	1	5	7	12
1	4	8	12	2	13	9	12
2	12	10	12	2	5	11	12
2	4	12	12	25	13	29	12

Board 1

IC	PIN	IC	PIN	IC	PIN	IC	PIN
25	12	30	12	25	5	31	12
25	4	32	12	26	13	33	12
26	12	34	12	26	5	35	12
26	4	36	12	37	13	41	12
37	12	42	12	37	5	43	12
37	4	44	12	38	13	45	12
38	12	46	12	38	5	47	12
38	4	48	12	53	11		47
54	11		48	55	11		49
56	11		50	57	11		51
58	11		52	59	11		53
60	11		54	49	15		108
49	10		109	49	7		110
49	2		111	50	15		112
50	10		113	50	7		114
50	2		115	17	11		23
18	11		24	19	11		25
20	11		26	21	11		27
22	11		28	23	11		29
24	11		30	13	15		84
13	10		85	13	7		86
13	2		87	14	15		88
14	10		89	14	7		90
14	2		91	2	1		69

Board 1

IC	PIN	IC	PIN	IC	PIN	IC	PIN
2	9		70	16	1		66
3	2		5	3	4		6
3	6		7	3	10		8
3	12		9	3	14		10
4	2		11	4	4		12
4	6		13	4	10		14
4	12		67	4	14		68
5	11		15	6	11		16
7	11		17	8	11		18
9	11		19	10	11		20
11	11		21	12	11		22
1	15		76	1	10		77
1	7		78	1	2		79
2	15		80	2	10		81
2	7		82	2	2		83
25	15		92	25	10		93
25	7		94	25	2		95
26	15		96	26	10		97
26	7		98	26	2		99
37	15		100	37	10		101
37	7		102	37	2		103
38	15		104	38	10		105
38	7		106	38	2		107
29	11		31	30	11		32

Board 1

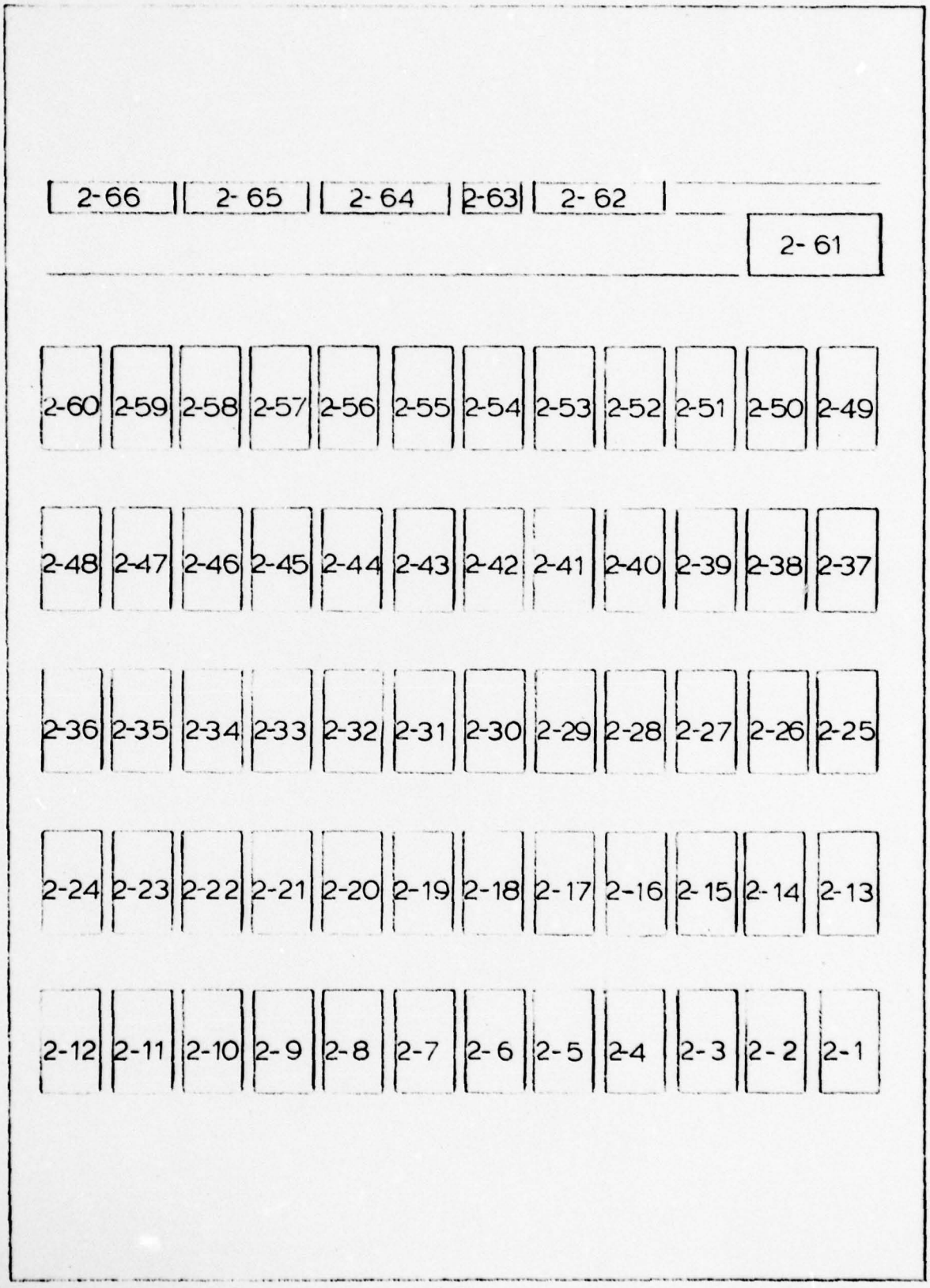
IC	PIN	IC	PIN	IC	PIN	IC	PIN
31	11		33	32	11		34
33	11		35	34	11		36
35	11		37	36	11		38
41	11		39	42	11		40
43	11		41	44	11		42
45	11		43	46	11		44
47	11		45	48	11		46
5	9	6	9	5	10	6	10
7	9	8	9	7	10	8	10
9	9	10	9	9	10	10	10
11	9	12	9	11	10	12	10
10	9	11	9	10	10	11	10
8	9	9	9	8	10	9	10
6	9	7	9	6	10	7	10
5	9	4	8	5	10	4	16
17	9	18	9	17	10	18	10
19	9	20	9	19	10	20	10
21	9	22	9	21	10	22	10
23	9	24	9	23	10	24	10
22	9	23	9	22	10	23	10
20	9	21	9	20	10	21	10
18	9	19	9	18	10	19	10
17	9	16	8	17	10	16	16
29	9	30	9	29	10	30	10

Board 1

IC	PIN	IC	PIN	IC	PIN	IC	PIN
31	9	32	9	31	10	32	10
33	9	34	9	33	10	34	10
35	9	36	9	35	10	36	10
34	9	35	9	34	10	35	10
32	9	33	9	32	10	33	10
30	9	31	9	30	10	31	10
29	9	28	8	29	10	28	16
41	9	42	9	41	10	42	10
43	9	44	9	43	10	44	10
45	9	46	9	45	10	46	10
47	9	48	9	47	10	48	10
46	9	47	9	46	10	47	10
44	9	45	9	44	10	45	10
42	9	43	9	42	10	43	10
41	9	40	8	41	10	40	16
53	9	54	9	53	10	54	10
55	9	56	9	55	10	56	10
57	9	58	9	57	10	58	10
59	9	60	9	59	10	60	10
58	9	59	9	58	10	59	10
56	9	57	9	56	10	57	10
54	9	55	9	54	10	55	10
53	9	52	8	53	10	52	16

Board 1

IC	PIN	IC	PIN	IC	PIN	IC	PIN
1	8		3	1	16		1
2	8		4	2	16		2
3	8		64	3	16		62
4	8		.65	4	16		63


2-66
2-65
2-64
2-63
2-62
2-61
2-60
2-59
2-58
2-57
2-56
2-55
2-54
2-53
2-52
2-51
2-50
2-49
2-48
2-47
2-46
2-45
2-44
2-43
2-42
2-41
2-40
2-39
2-38
2-37
2-36
2-35
2-34
2-33
2-32
2-31
2-30
2-29
2-28
2-27
2-26
2-25
2-24
2-23
2-22
2-21
2-20
2-19
2-18
2-17
2-16
2-15
2-14
2-13
2-12
2-11
2-10
2-9
2-8
2-7
2-6
2-5
2-4
2-3
2-2
2-1

Board 2 Integrated Circuits

Position	Type	Position	Type
2-1	SN74175	2-25	SN7404
2-2	SN74175	2-26	SN7421
2-3	SN74175	2-27	SN74191
2-4	SN74175	2-28	SN74191
2-5	SN74175	2-29	SN74191
2-6	SN74175	2-30	SN74191
2-7	SN74175	2-31	SN74175
2-8	SN74175	2-32	SN7404
2-9	SN74LS367	2-33	SN7404
2-10	SN74LS367	2-34	SN74138
2-11	SN74LS367	2-35	SN74138
2-12	SN74LS367	2-36	SN7408
2-13	SN74175	2-37	SN74177
2-14	SN74175	2-38	SN74138
2-15	SN74175	2-39	SN7404
2-16	SN74175	2-40	SN7485
2-17	SN74175	2-41	SN7485
2-18	SN74175	2-42	SN7485
2-19	SN74175	2-43	SN7485
2-20	SN7404	2-44	SN7485
2-21	SN7408	2-45	SN7485
2-22	SN74138	2-46	SN7485
2-23	SN74138	2-47	SN7485
2-24	SN7402	2-48	SN7485

Board 2 Integrated Circuits

Position	Type	Position	Type
2-49	SN74175	2-58	4114R
2-50	SN7400	2-59	4114R
2-51	SN7400	2-60	4114R
2-52	SN7474	2-61	SN74175
2-53	SN7474	2-62	Dip Switch(8)
2-54	SN7474	2-63	Dip Switch(8)
2-55	SN7474	2-64	Dip Switch(8)
2-56	4114R	2-65	Dip Switch(4)
2-57	4114R	2-66	Dip Switch(8)

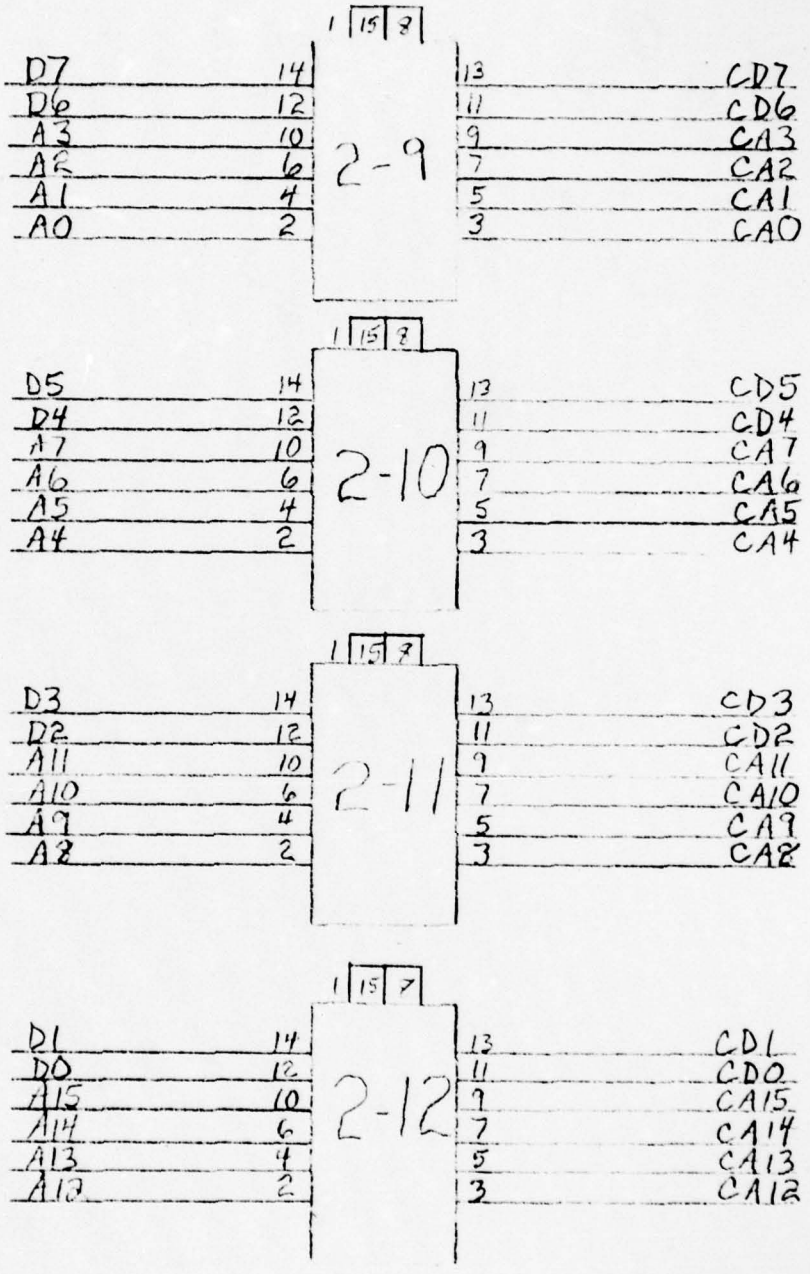


Fig 6 Board 2 Bus Interface

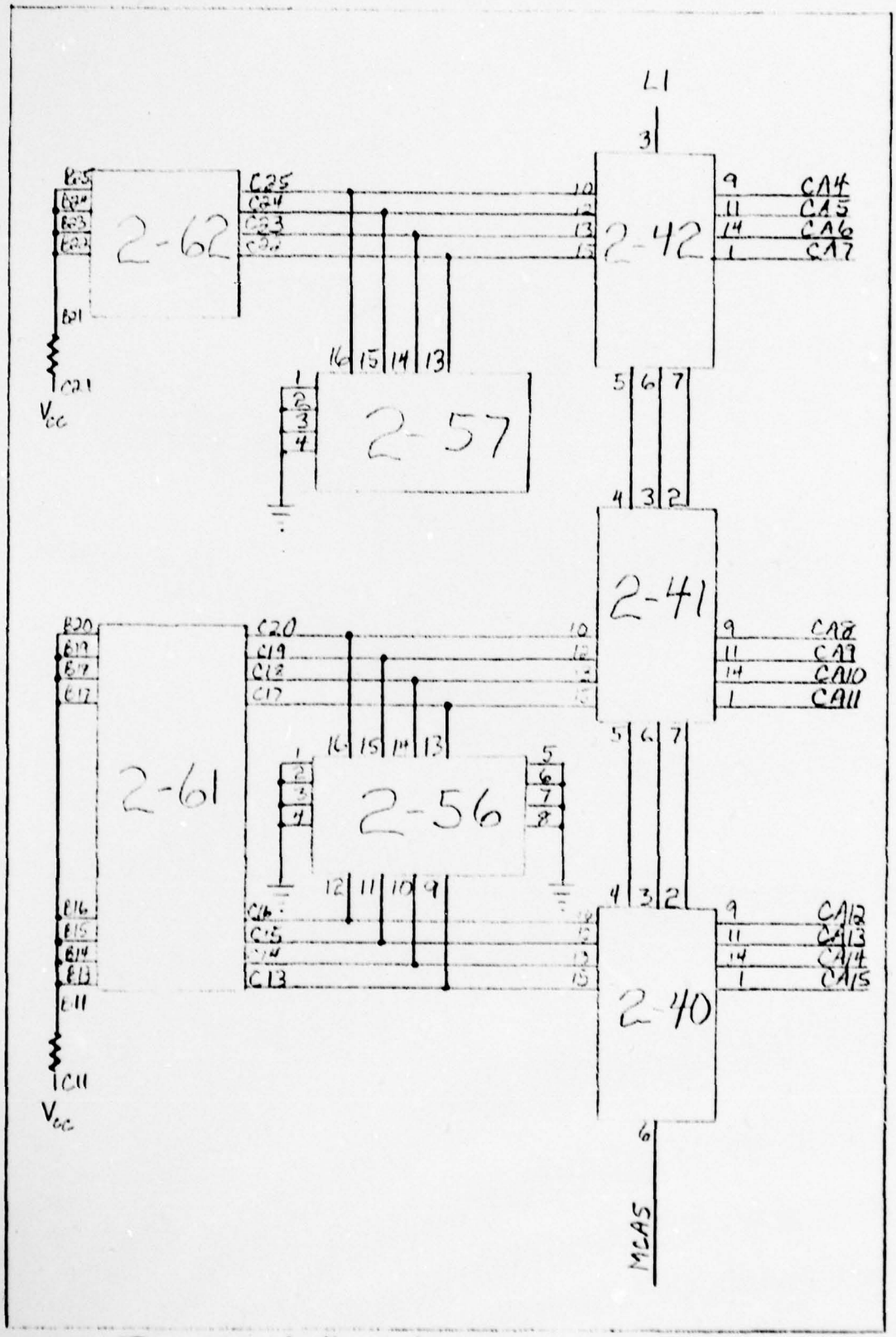


Fig 7 Controller Block Address Decoder

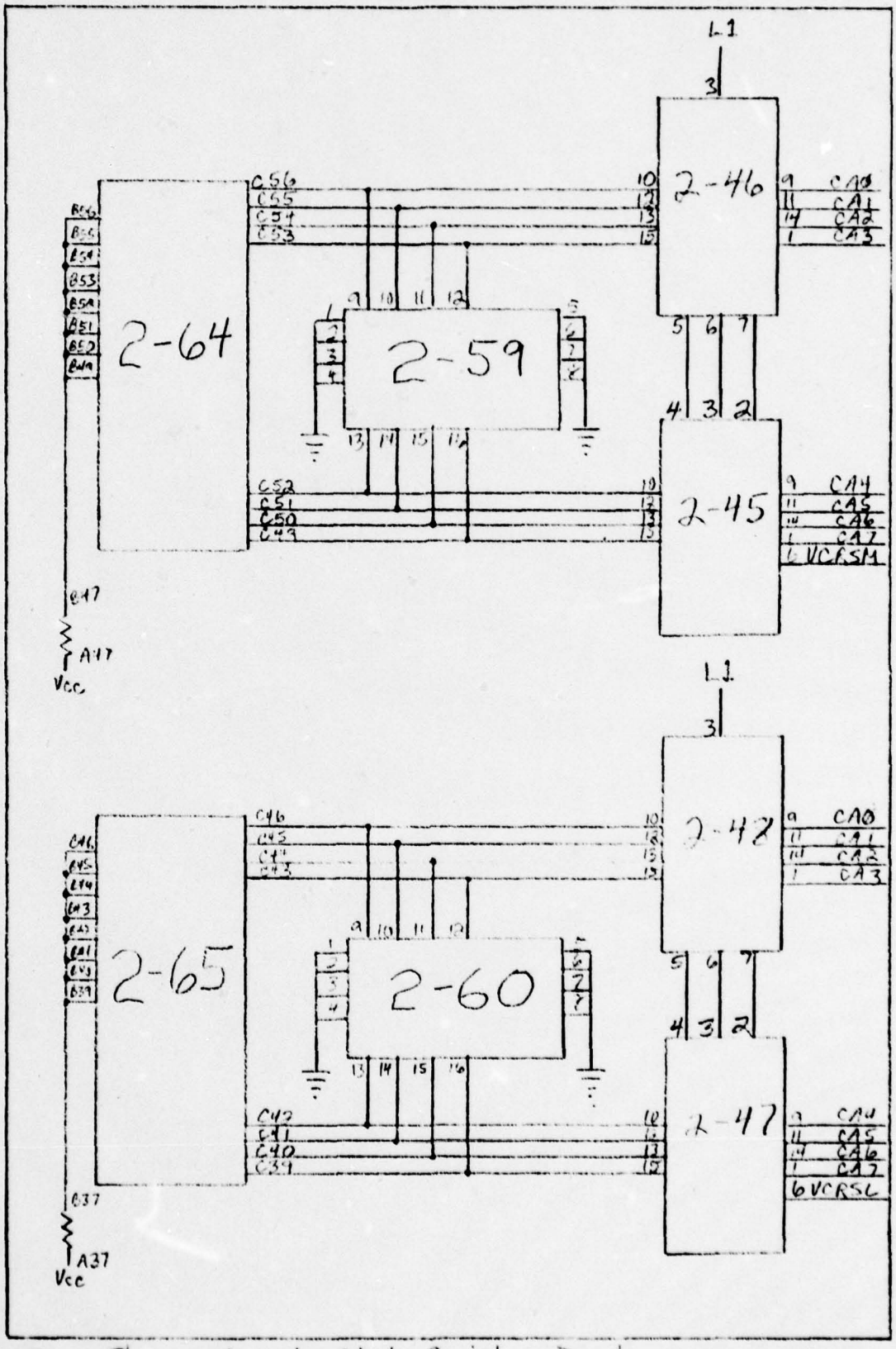


Fig 8 Variable Clock Register Decoder

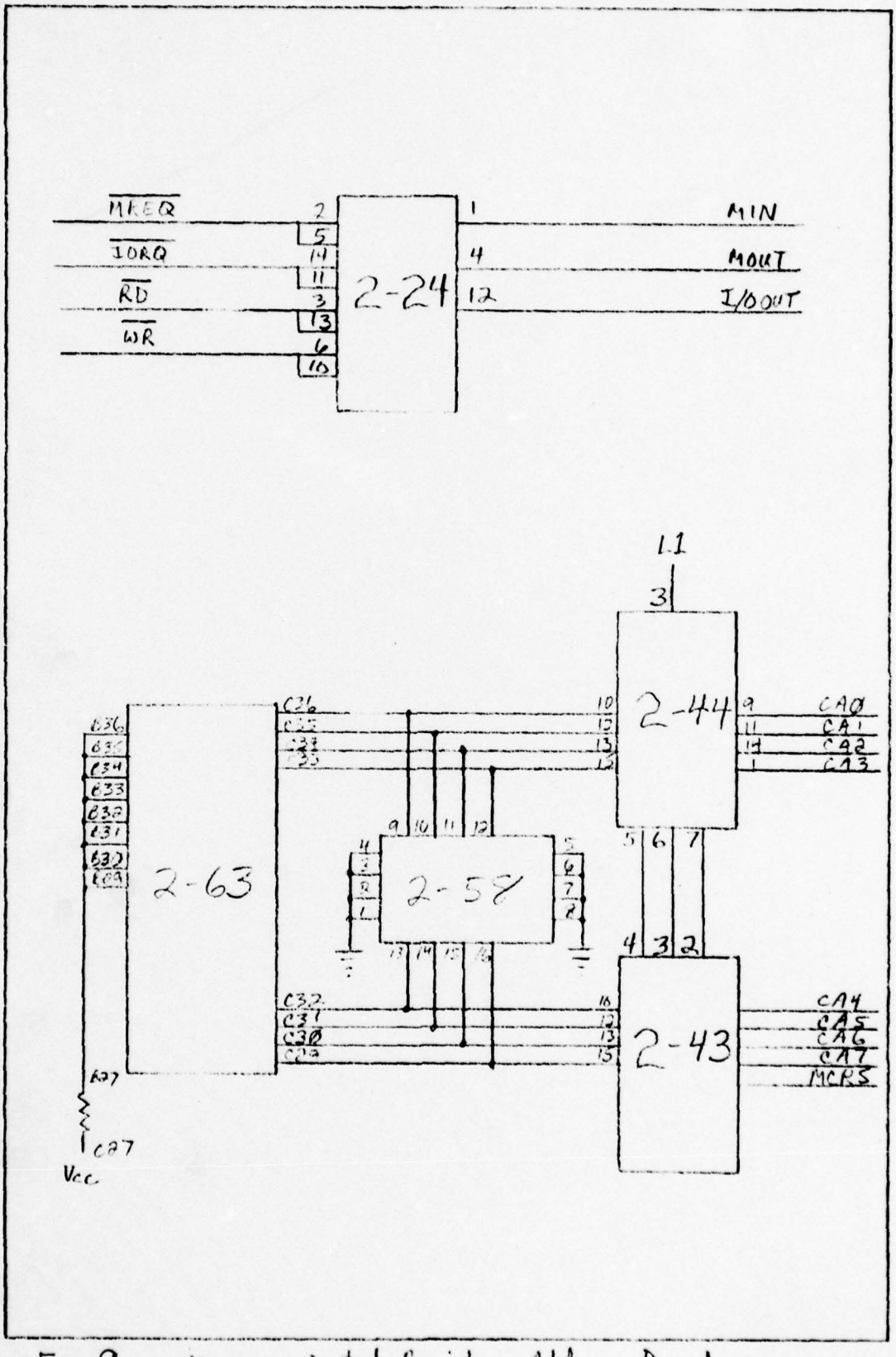


Fig 9 Memory Control Register Address Decoder

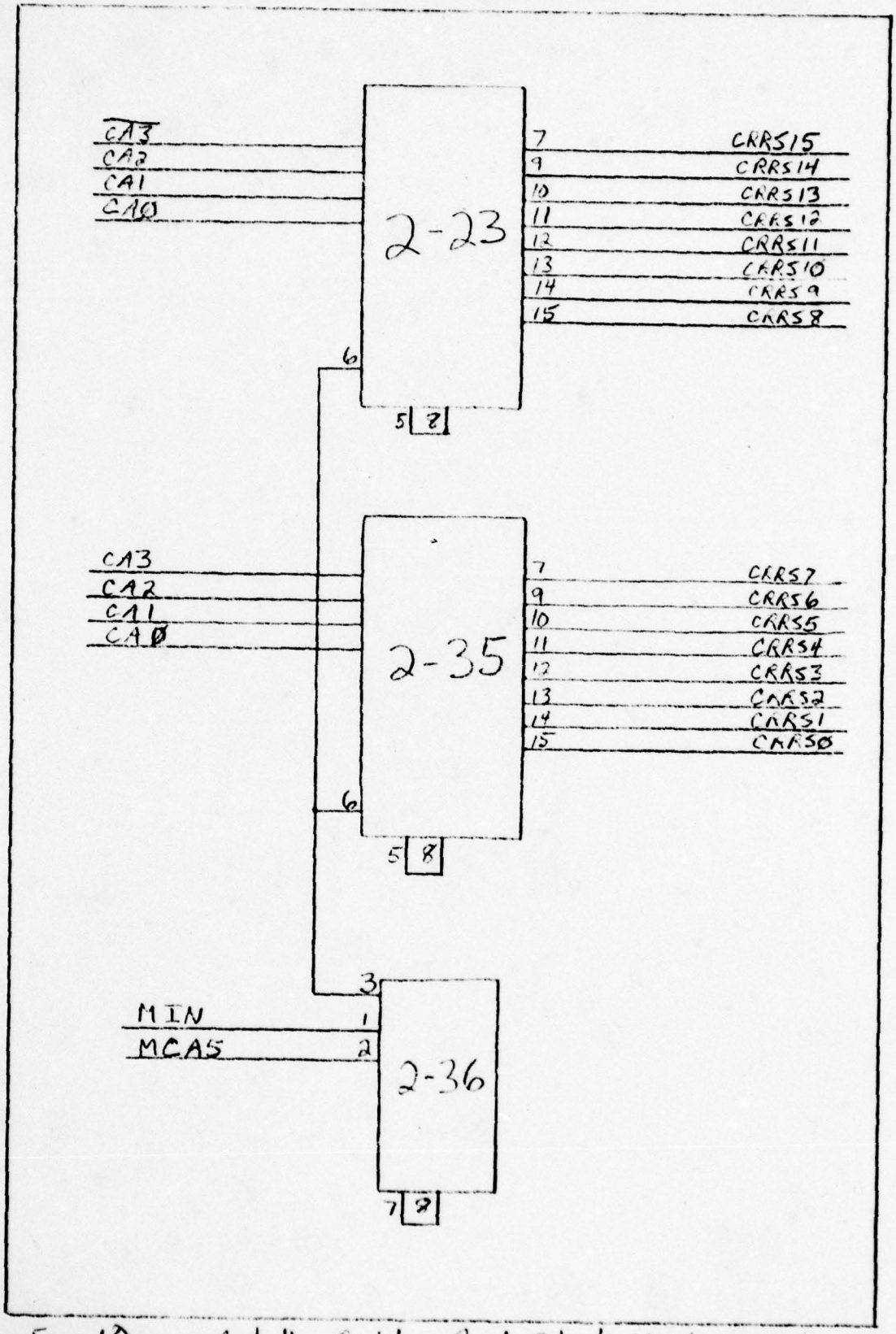


Fig 10 Controller Register Read Select Signals

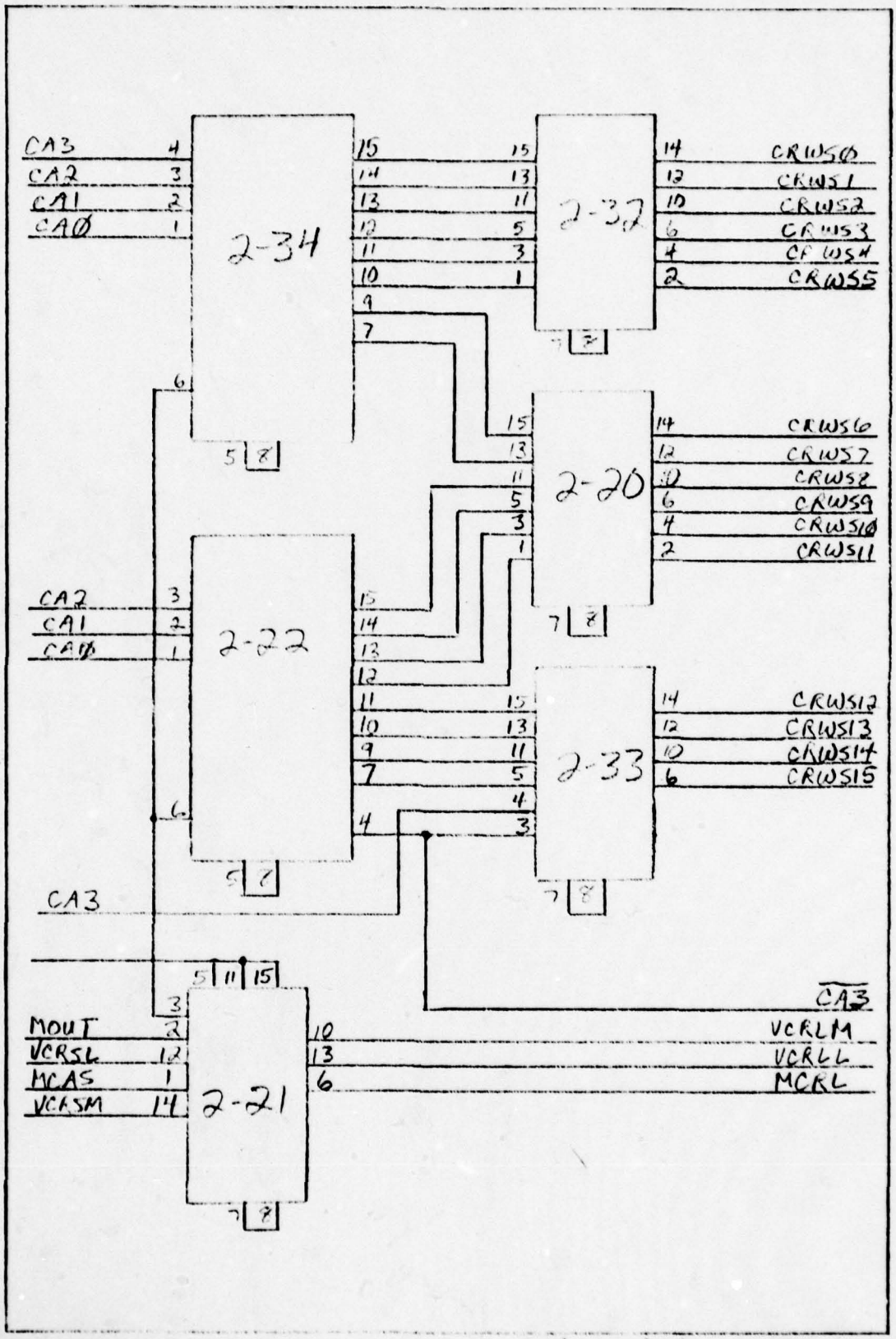


Fig 11 Controller Register Write Select Signals

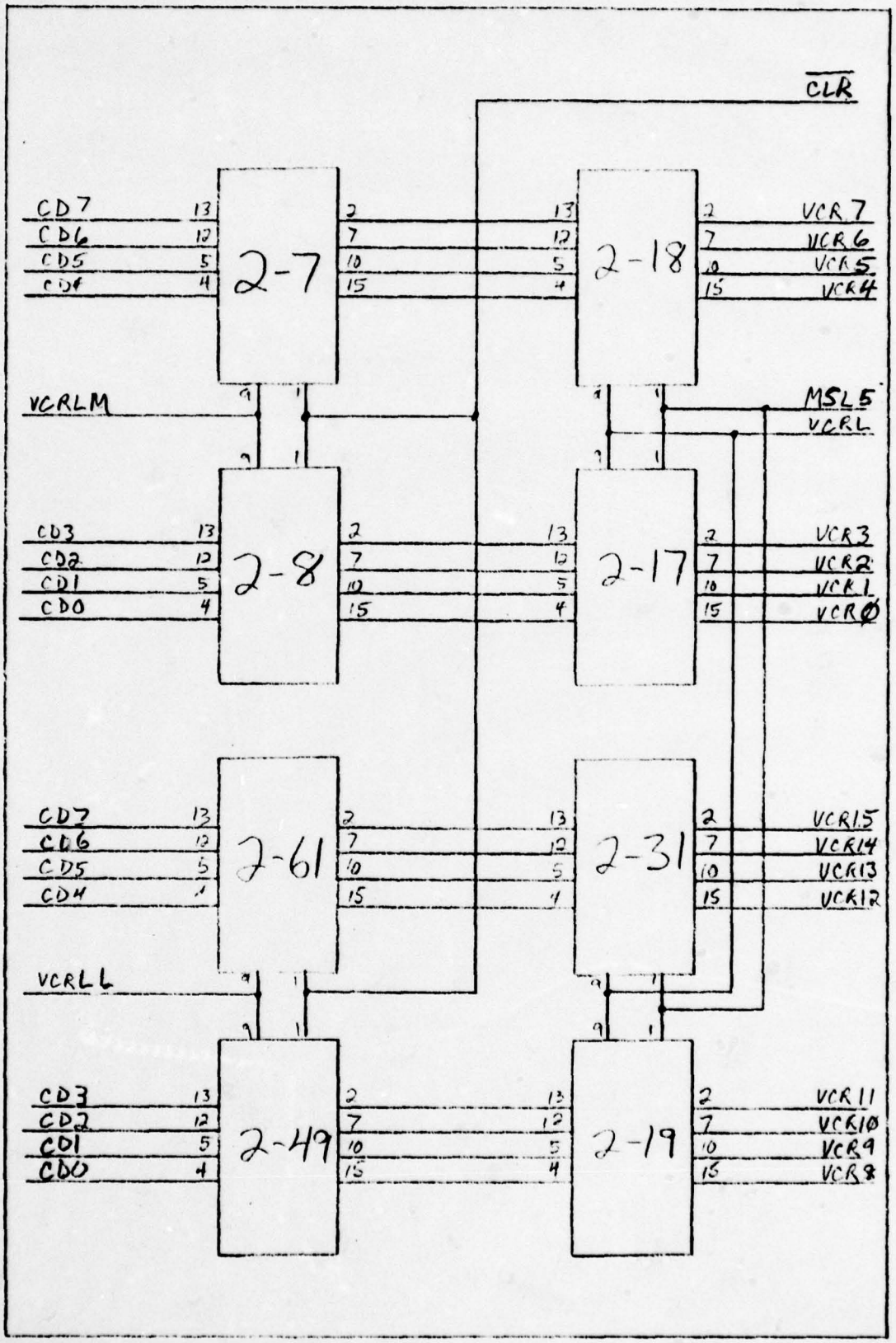


Fig 12 Variable Clock Registers

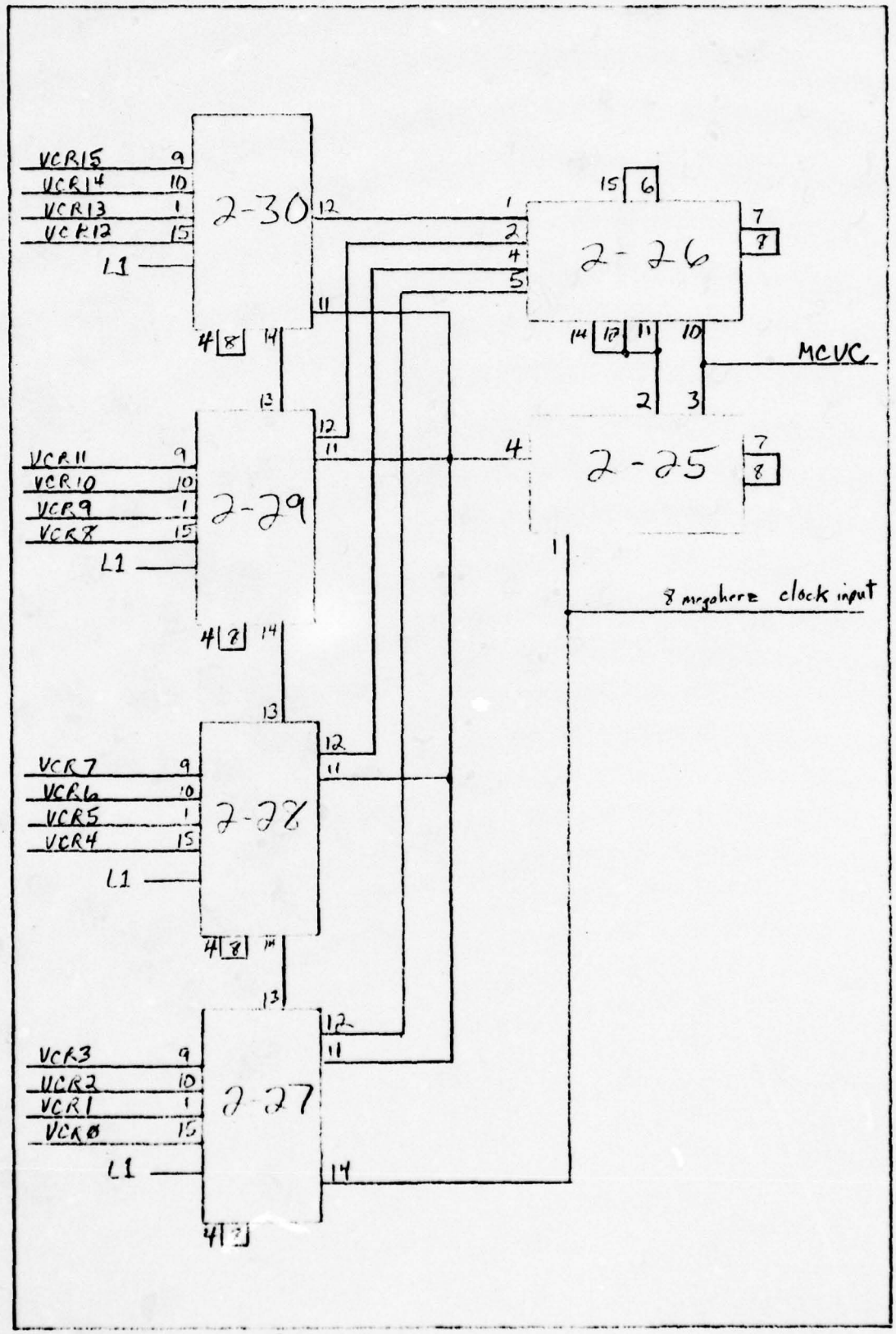


Fig 13 Variable Clock

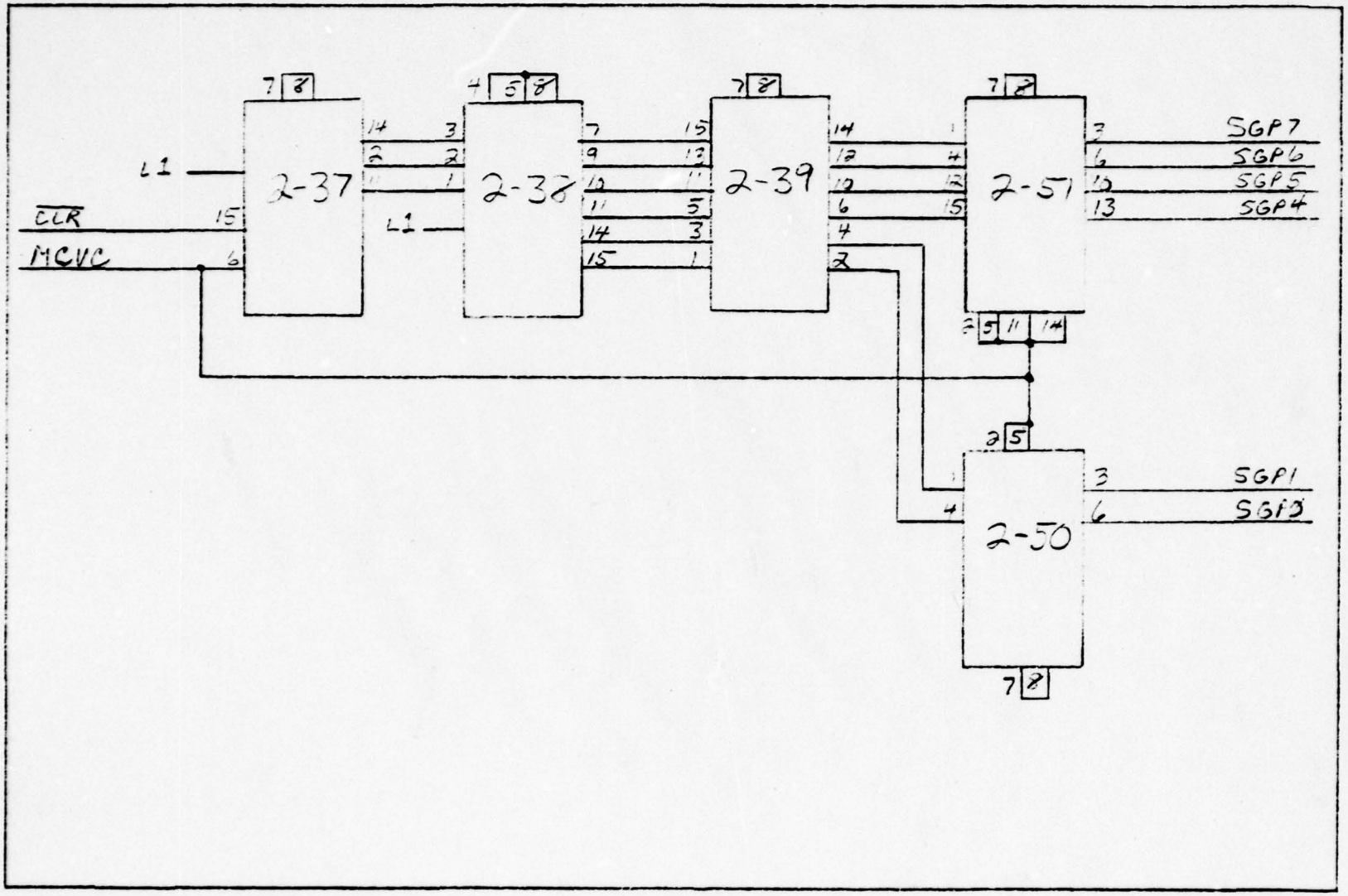


FIG 14 State Generator

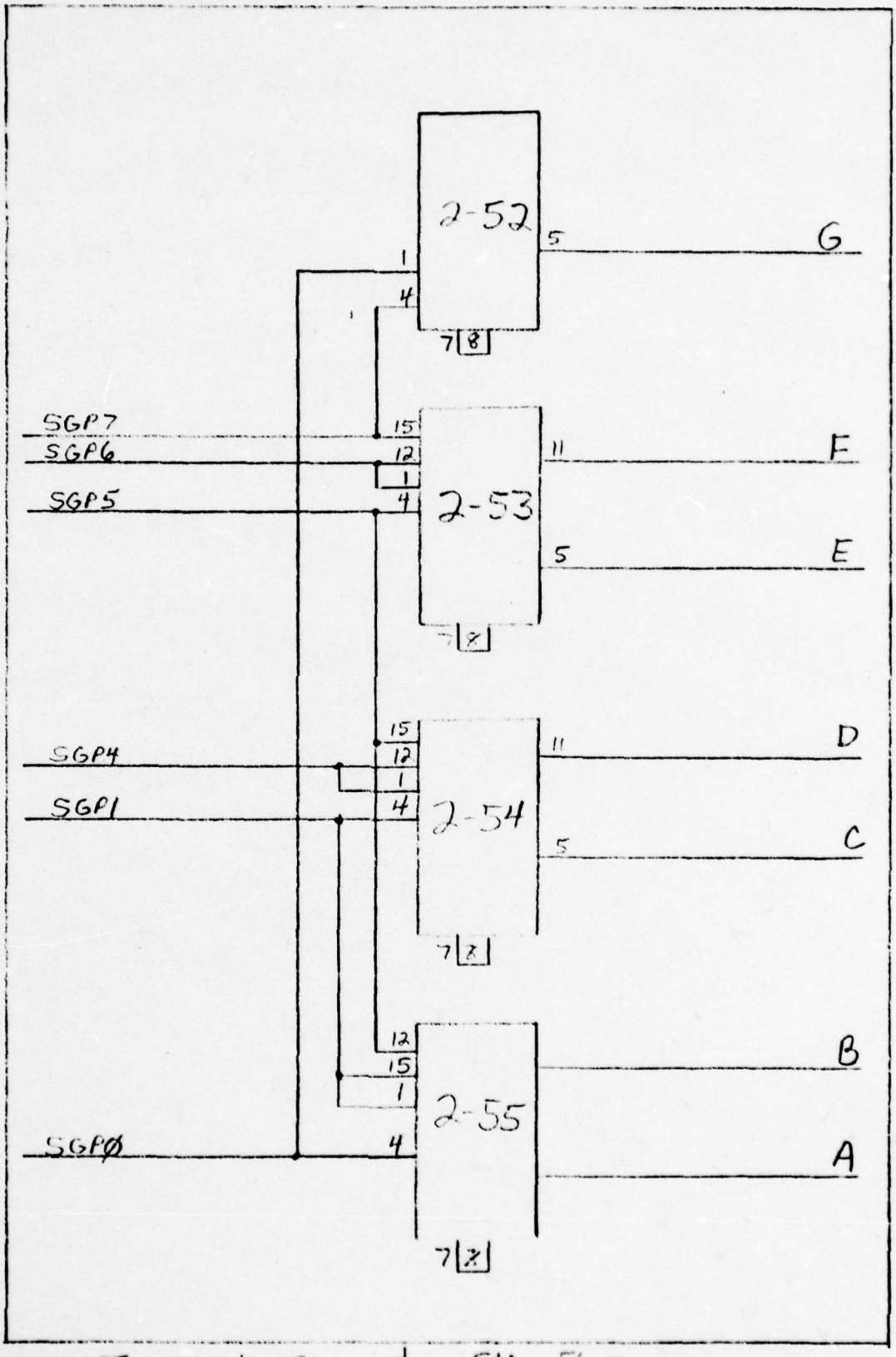


Fig 15 State Generator Flip-Flops

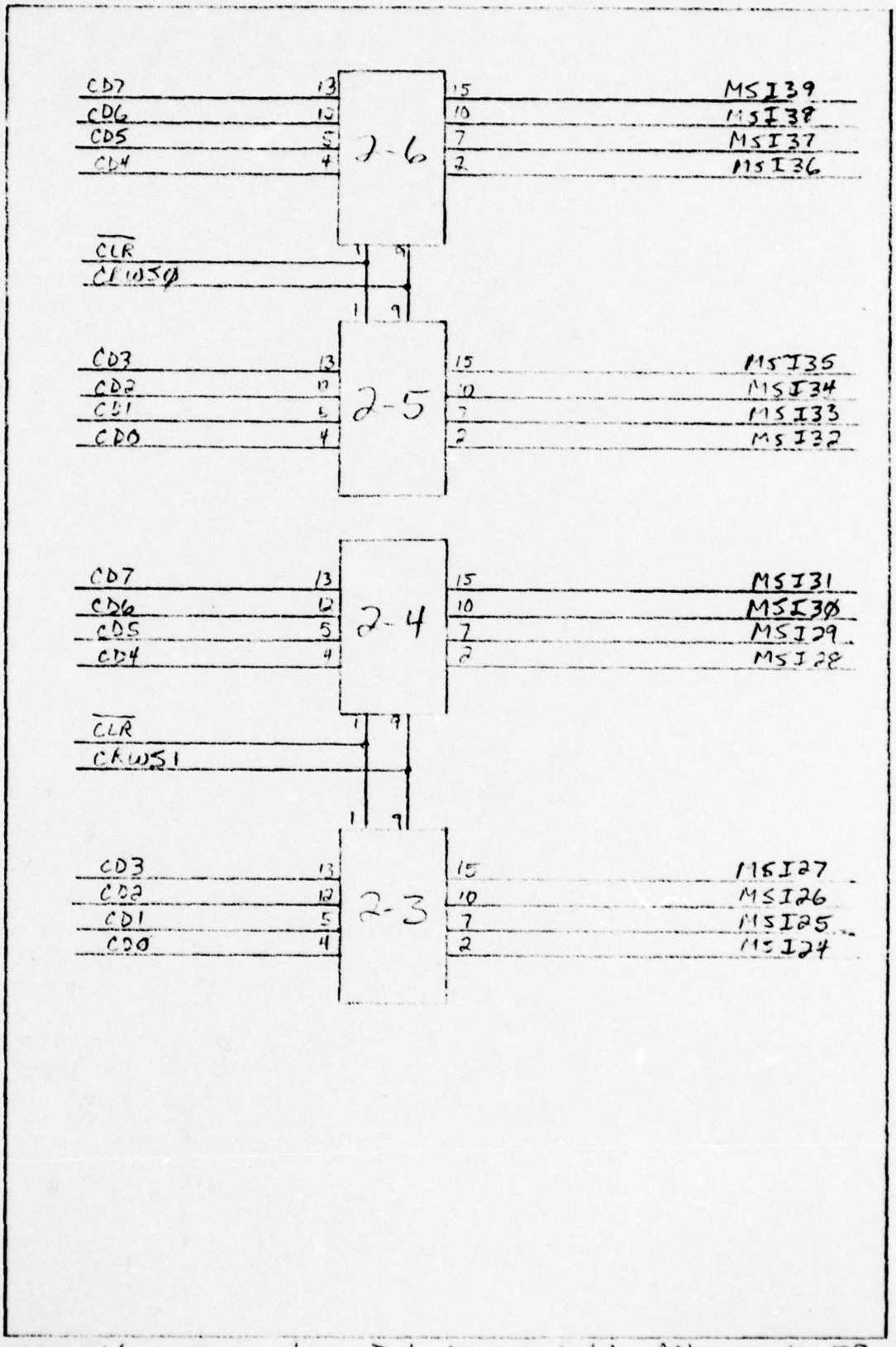


Fig 16 Micro-share Data Input Latch Bits 24 thru 39

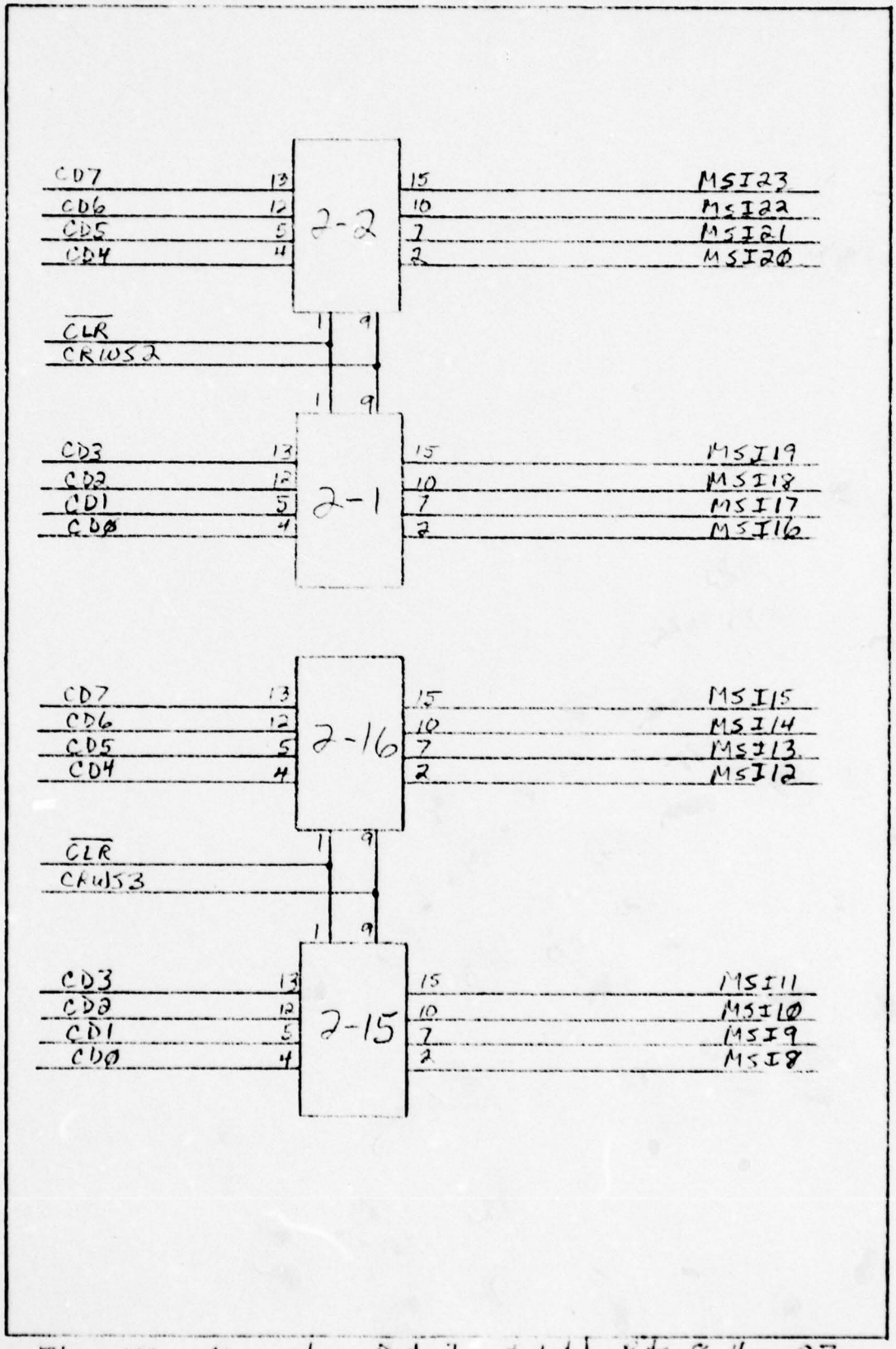


Fig 17 Micro-store Data Input Latch Bits 8 thru 23

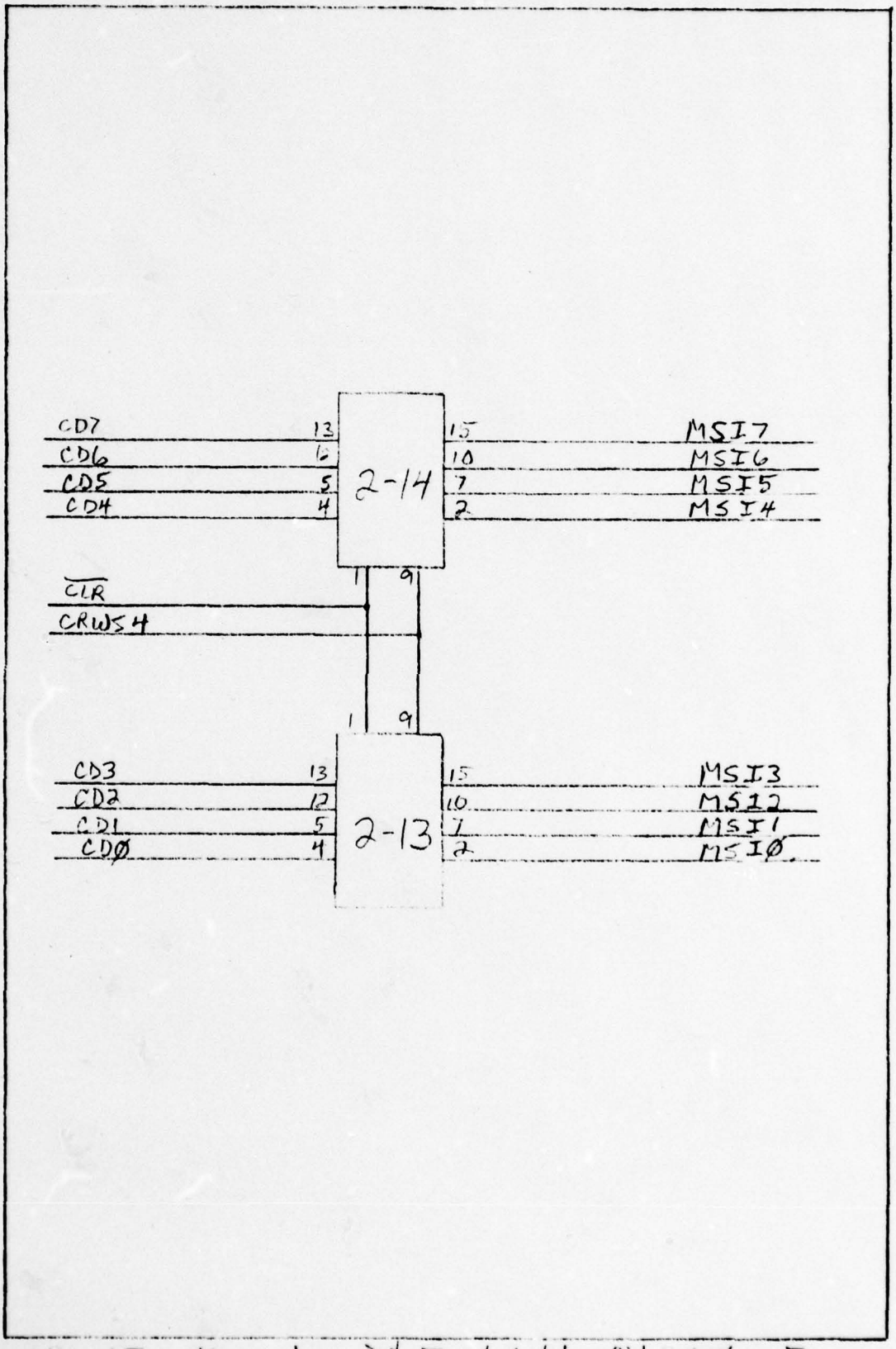


Fig 18 Micro-store Data Input Latch Bits Ø thru 7

Pin to Pin Connections for Board 2

The Zilog Wire-wrap Board is preconfigured with 16 pin sockets with the V_{cc} Plane connected to pin 16 and with the GND Plane connected to pin 8. Note the following examples to read the Appendix.

IC	PIN	IC	PIN
5	9	11	6

This means a wire was physically connected to these two uniquely defined wire wrap posts.

IC	PIN	IC	PIN
5	7		A28

A wire was connected from IC 5 at pin 7 to Column A of the strip on the top of the wire wrap board pin number 28.

IC	PIN	IC	PIN
5	14		68

IC 5 pin 14 was connected to Edge connector pin number 68. The 14 pin dips are mounted with pin 1 in socket number one. This requires 2 be added to the right side (8-14) pin number to arrive at the adjusted pin number for the 16 pin socket. This has already been done. Consequently, you will note some connections to pins 15 and 16 of 14 pin sips.

Note: There are two columns of connections on each page.

Board 2

IC	PIN	IC	PIN	IC	PIN	IC	PIN
	B11		B13		B13		B14
	B14		B15		B15		B16
	B16		B17		B17		B18
	B18		B19		B19		B20
	B21		B23		B23		B24
	B24		B25		B25		B26
	B27		B29		B29		B30
	B30		B31		B31		B32
	B32		B33		B33		B34
	B34		B35		B35		B36
	B37		B39		B39		B40
	B40		B41		B41		B42
	B42		B43		B43		B44
	B44		B45		B45		B46
	B47		B49		B49		B50
	B50		B51		B51		B52
	B52		B53		B53		B54
	B54		B55		B55		B56
	C11		V_{cc}		C21		V_{cc}
	C27		V_{cc}		C37		V_{cc}
	C47		V_{cc}	60	1	60	2
60	3	60	4	60	5	60	6
60	7	60	8	59	1	59	2
59	3	59	4	59	5	59	6

Board 2

IC	PIN	IC	PIN	IC	PIN	IC	PIN
59	7	59	8	58	1	58	2
58	3	58	4	58	5	58	6
58	7	58	8	57	1	57	2
57	3	57	4	57	5	57	6
57	7	57	8	56	1	56	2
56	3	56	4	56	5	56	6
56	7	56	8		C26		C28
	C32		Gnd		C22		V_{cc}
60	16		C49	60	15		C50
60	14		C51	60	13		C52
60	12		C53	60	11		C54
60	10		C55	60	9		C56
59	16		C39	59	15		C40
59	14		C41	59	13		C42
59	12		C43	59	11		C44
59	10		C45	59	9		C46
58	16		C29	58	15		C30
58	14		C31	58	13		C32
58	12		C33	58	11		C34
58	10		C35	58	9		C36
57	16		C23	57	15		C24
57	14		C25	57	13		C26
56	16		C13	56	15		C14
56	14		C15	56	13		C16

Board 2

IC	PIN	IC	PIN	IC	PIN	IC	PIN
56	12		C17	56	11		C18
56	10		C19	56	9		C20
48	3		C26	46	3		C26
44	3		C28	42	3		C28
37	1	38	6	38	6	42	3
60	16	47	15	60	15	47	13
60	14	47	12	60	13	47	10
60	12	48	15	60	11	48	13
60.	10	48	12	60	9	48	10
47	4	48	5	47	3	48	6
47	2	48	7	59	16	45	15
59	15	45	13	49	14	45	12
59	13	45	10	59	12	46	15
59	11	46	13	59	10	46	12
59	9	46	10	45	4	46	5
45	3	46	6	45	2	46	7
58	16	43	15	58	15	43	13
58	14	43	12	58	13	43	10
58	12	44	15	58	11	44	13
58	10	44	12	58	9	44	10
57	16	42	15	57	15	42	13
57	14	42	12	57	13	42	10
43	4	44	5	43	3	44	6
43	2	44	7	41	4	42	5

Board 2

IC	PIN	IC	PIN	IC	PIN	IC	PIN
41	3	42	6	40	4	41	5
40	3	41	6	40	2	41	7
56	16	40	15	56	15	40	13
56	14	40	12	56	13	40	10
56	12	41	15	56	11	41	13
56	10	41	12	56	9	41	10
12	9	40	1	12	7	40	14
12	5	40	11	12	3	40	9
11	9	41	1	11	7	41	14
11	5	41	11	11	3	41	9
12	1	12	8	12	15	12	8
11	1	11	8	11	15	11	8
10	1	10	8	10	15	10	8
9	1	9	8	9	15	9	8
47	9	45	9	47	11	45	11
47	14	45	14	47	1	45	1
43	9	42	9	43	11	42	11
43	14	42	14	43	1	42	1
10	9	47	1	10	7	47	14
10	5	47	11	10	3	47	9
45	9	43	9	45	11	43	11
45	14	43	14	45	1	43	1
48	9	46	9	48	11	46	11
48	14	46	14	48	1	46	1

Board 2

IC	PIN	IC	PIN	IC	PIN	IC	PIN
23	3	35	3	23	2	35	2
23	1	35	1	35	4	48	1
35	3	48	14	35	2	48	11
35	1	48	9	44	9	9	3
44	11	9	5	44	14	9	7
44	1	9	9	44	9	46	9
44	11	46	11	44	14	46	14
44	1	46	1	34	3	22	3
34	2	22	2	34	1	22	1
34	4	35	4	34	3	35	3
34	2	35	2	34	1	35	1
40	6	36	2	40	6	21	1
43	6	21	4	45	6	21	14
47	6	21	12	24	1	36	1
24	4	21	2	24	12	21	5
21	11	21	15	21	11	21	5
21	7	21	8	34	5	34	8
22	5	22	8	36	7	36	8
23	5	23	8	23	6	35	6
35	6	36	3	34	6	22	6
21	3	22	6	8	9	7	9
8	1	7	9	21	10	8	9
49	1	AB	B1	49	9	AB	A8
9	9	33	3	33	4	23	4

Board 2

IC	PIN	IC	PIN	IC	PIN	IC	PIN
32	7	32	8	20	7	20	8
33	7	33	8	34	15	32	15
34	14	32	13	34	13	32	11
34	12	32	5	34	11	32	3
34	10	32	1	34	9	20	15
34	7	20	13	22	15	20	11
22	14	20	5	22	13	20	3
22	12	20	1	22	11	33	15
22.	10	33	13	22	9	33	11
22	7	33	5	8	4	12	11
8	15	12	13	8	12	11	11
8	13	11	13	7	4	10	11
7	5	10	13	7	12	9	11
7	13	9	13	49	4	12	11
49	5	12	13	49	12	11	11
49	13	11	13	AB	A4	10	11
AB	A5	10	13	AB	B5	9	11
AB	B4	9	13	AB	A2	31	13
AB	A7	31	12	AB	B2	31	5
AB	B7	31	4	49	15	19	13
49	10	19	12	49	7	19	5
49	2	19	4	8	15	17	13
8	10	17	12	8	7	17	5
8	2	17	4	7	15	18	13

Board 2

IC	PIN	IC	PIN	IC	PIN	IC	PIN
7	10	18	12	7	7	18	5
7	2	18	4	31	1	19	1
31	9	19	9	17	1	18	1
17	9	18	9	31	1	18	1
31	9	18	9	49	1	37	15
37	1	1	1	6	1	5	1
4	1	3	1	2	1	1	1
16	1	15	1	14	1	13	1
7	1	6	1	5	1	4	1
3	1	2	1	15	1	14	1
16	1	4	1	6	9	5	9
4	9	3	9	2	9	1	9
16	9	15	9	14	9	13	9
31	15	30	9	31	10	30	10
31	7	30	1	31	2	30	15
19	15	29	9	19	10	29	10
19	7	29	1	19	2	29	15
18	15	28	9	18	10	28	10
18	7	28	1	18	2	28	15
17	15	27	9	17	10	27	10
17	7	27	1	17	2	27	15
30	14	29	13	30	5	29	5
28	14	27	13	28	5	27	5
29	14	28	13	29	5	28	5

Board 2

IC	PIN	IC	PIN	IC	PIN	IC	PIN
30	5	42	5	37	1	38	6
38	6	27	5	30	4	30	8
29	4	29	8	28	4	28	8
27	4	27	8	27	14	25	1
25	4	30	11	25	4	29	11
25	4	28	11	25	4	27	11
26	1	30	12	26	2	29	12
26	4	28	12	26	5	27	12
26	15	26	6	26	7	26	8
26	11	26	12	26	12	26	14
26	11	25	2	25	7	25	8
25	3	26	10	25	3	37	6
26	10	50	2	50	2	50	5
50	5	51	2	51	11	51	14
51	5	51	11	51	5	51	2
50	7	50	8	51	7	51	8
37	7	37	8	37	14	38	3
37	2	38	2	37	11	38	1
38	4	38	5	38	5	38	8
38	7	39	15	38	9	39	13
38	10	39	11	38	11	39	5
38	14	39	3	38	15	39	1
39	14	51	1	39	12	51	4
39	10	51	12	39	6	51	15

Board 2

IC	PIN	IC	PIN	IC	PIN	IC	PIN
39	4	50	1	39	2	50	4
52	4	53	15	53	12	53	1
53	15	51	3	53	12	51	6
53	4	51	10	54	15	55	12
54	15	53	4	54	12	54	4
51	13	54	12	55	15	55	1
50	3	54	1	54	1	55	1
50	6	52	1	52	1	55	4
32	14	6	9	32	12	4	9
32	10	2	9	32	6	16	9
32	4	14	9	6	13	7	13
6	12	7	12	6	5	7	5
6	4	7	4	5	13	8	13
5	12	8	12	5	5	8	5
5	4	8	4	4	13	2	13
4	12	2	12	4	5	2	5
4	4	2	4	3	13	1	13
3	12	1	12	3	5	1	5
3	4	1	4	16	13	14	13
16	12	14	12	16	5	14	5
16	4	14	4	15	13	13	13
15	12	13	12	15	5	13	5
15	4	13	4	6	13	4	13
6	12	4	12	6	5	4	5

Board 2

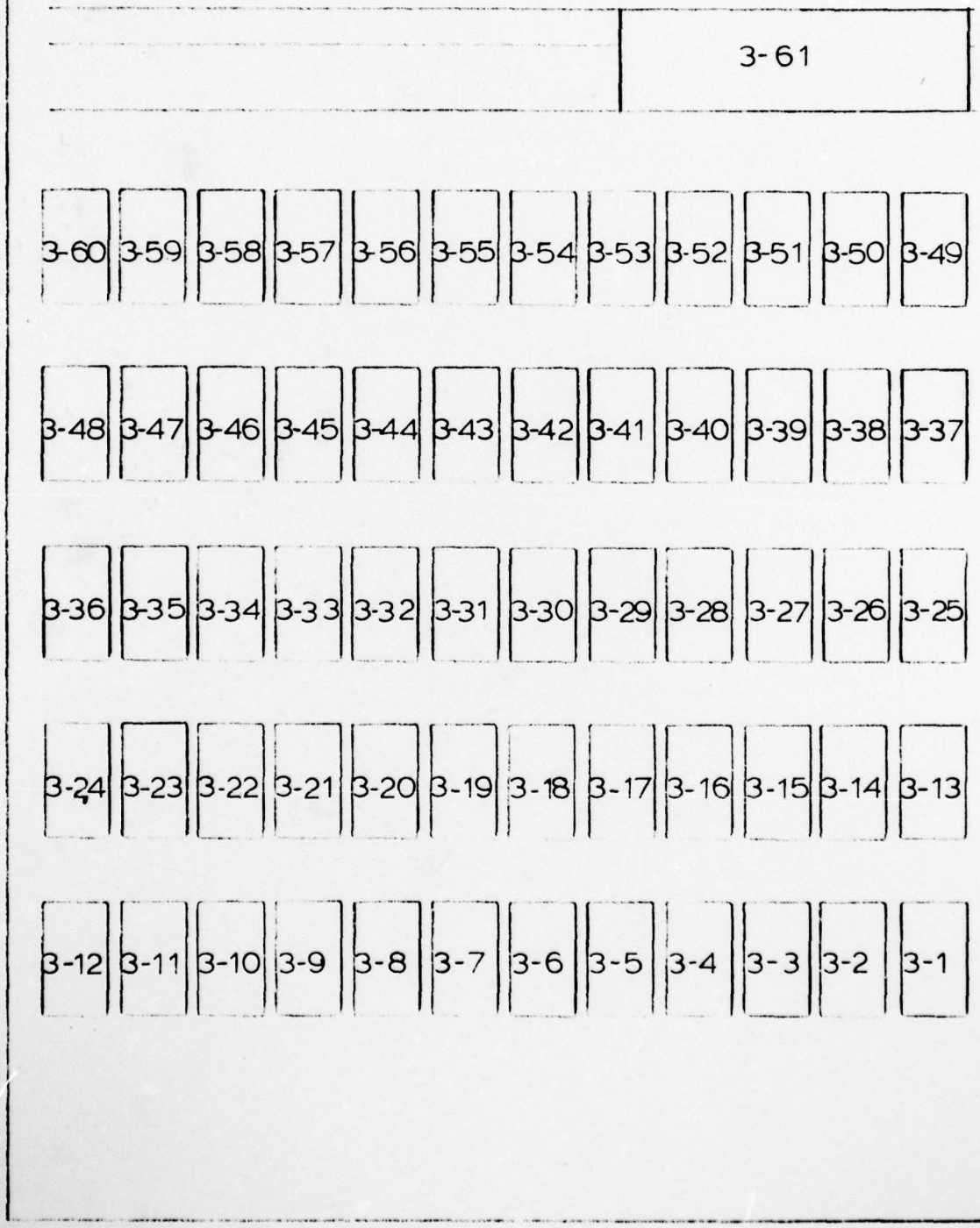
IC	PIN	IC	PIN	IC	PIN	IC	PIN
6	4	4	4	5	13	3	13
5	12	3	12	5	5	3	5
5	4	3	4	4	13	16	13
4	12	16	12	4	5	16	5
4	4	16	4	3	13	15	13
3	12	15	12	3	5	15	5
3	4	15	4	24	2	24	5
24	11	24	14	24	3	24	13
24	6	24	10	24	2		119
24	14		120	24	3		121
24	6		122	23	7		11
23	9		10	23	10		9
23	11		106	23	12		107
23	13		108	23	14		109
23	15		110	35	7		111
35	9		112	35	10		113
35	11		114	35	12		115
35	13		116	35	14		117
35	15		118	12	14		76
12	12		77	12	10		78
12	6		79	12	4		80
12	2		81	11	14		74
11	12		75	11	10		82
11	6		83	11	4		84

Board 2

IC	PIN	IC	PIN	IC	PIN	IC	PIN
11	2		85	10	14		72
10	12		73	10	10		86
10	6		87	10	4		88
10	2		89	9	14		70
9	12		71	9	10		90
9	6		91	9	4		92
9	2		93	21	6		6
32	14		105	32	12		104
32	10		103	32	6		102
32.	4		101	32	2		100
20	14		99	20	12		98
20	10		97	20	6		96
20	4		95	20	2		94
33	14		61	33	12		60
32	10		59	33	6		58
3	1		14	25	1		5
17	1		8	17	9		7
6	15		15	6	10		16
6	7		17	6	2		18
5	15		19	5	10		20
5	7		21	5	2		22
4	15		23	4	10		24
4	7		25	4	2		26
3	15		27	3	10		28

Board 2

IC	PIN	IC	PIN	IC	PIN	IC	PIN
3	7		29	3	2		30
2	15		31	2	10		32
2	7		33	2	2		34
1	15		35	1	10		36
1	7		37	1	2		38
16	15		39	16	10		40
16	7		41	16	2		42
15	15		43	15	10		44
15	7		45	15	2		46
14	15		47	14	10		48
14	7		49	14	2		50
13	15		51	13	10		52
13	7		53	13	2		54
52	5		57	53	11		56
53	5		55	54	11		69
54	5		68	55	11		67
55	5		66	1	16		1
2	16		2	3	16		62
4	16		63	1	8		3
2	8		4	3	8		64
4	8		65	52	8	53	8
52	16	53	16	54	8	55	8
54	16	55	16	53	8	54	8
53	16	54	16	51	8	52	8
51	16	52	16				


3-61
3-60 3-59 3-58 3-57 3-56 3-55 3-54 3-53 3-52 3-51 3-50 3-49
3-48 3-47 3-46 3-45 3-44 3-43 3-42 3-41 3-40 3-39 3-38 3-37
3-36 3-35 3-34 3-33 3-32 3-31 3-30 3-29 3-28 3-27 3-26 3-25
3-24 3-23 3-22 3-21 3-20 3-19 3-18 3-17 3-16 3-15 3-14 3-13
3-12 3-11 3-10 3-9 3-8 3-7 3-6 3-5 3-4 3-3 3-2 3-1

Board 3 Integrated Circuits

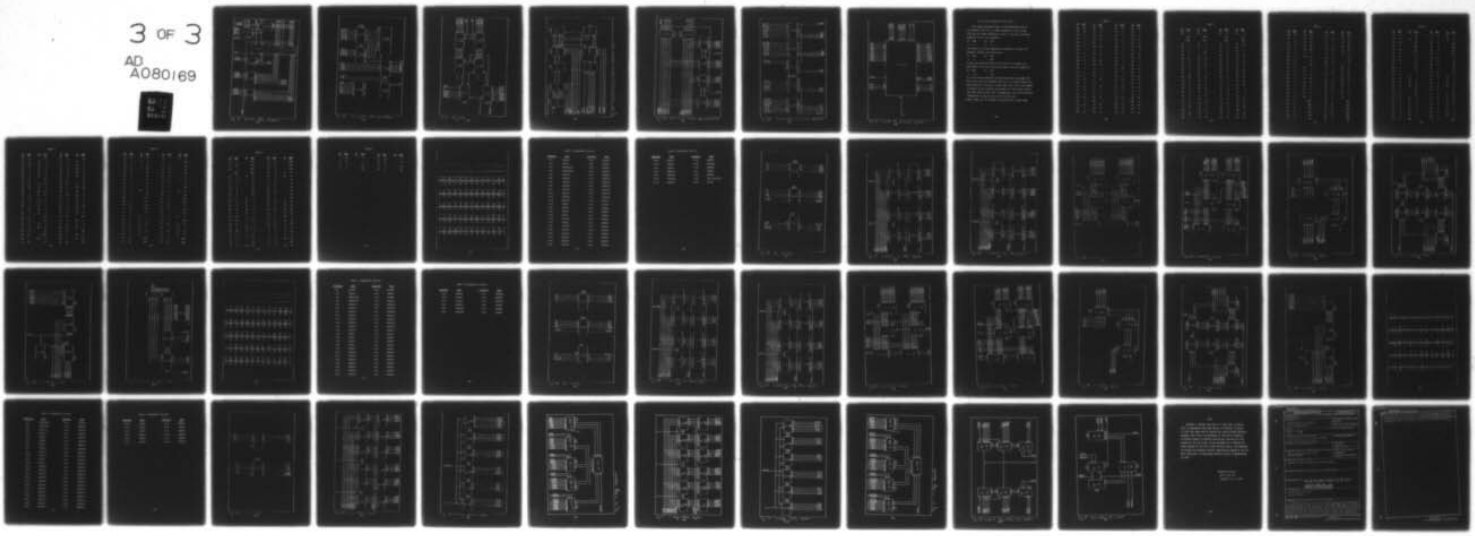
Position	Type	Position	Type
3-1	SN7408	3-25	SN74175
3-2	SN7408	3-26	SN74175
3-3	Unused	3-27	Unused
3-4	Unused	3-28	SN7408
3-5	Unused	3-29	SN7432
3-6	Unused	3-30	SN7474
3-7	Unused	3-31	SN7474
3-8	Unused	3-32	SN7408
3-9	Unused	3-33	SN7404
3-10	Unused	3-34	SN7408
3-11	Unused	3-35	SN7474
3-12	Unused	3-36	SN74LS367
3-13	SN74191	3-37	SN74175
3-14	SN74191	3-38	SN74175
3-15	Unused	3-39	SN74175
3-16	Unused	3-40	SN74LS367
3-17	Unused	3-41	SN74LS367
3-18	Unused	3-42	SN7408
3-19	Unused	3-43	SN74LS368
3-20	Unused	3-44	SN74175
3-21	Unused	3-45	SN74175
3-22	Unused	3-46	SN74175
3-23	Unused	3-47	SN74175
3-24	Unused	3-48	SN74LS367

Board 3 Integrated Circuits

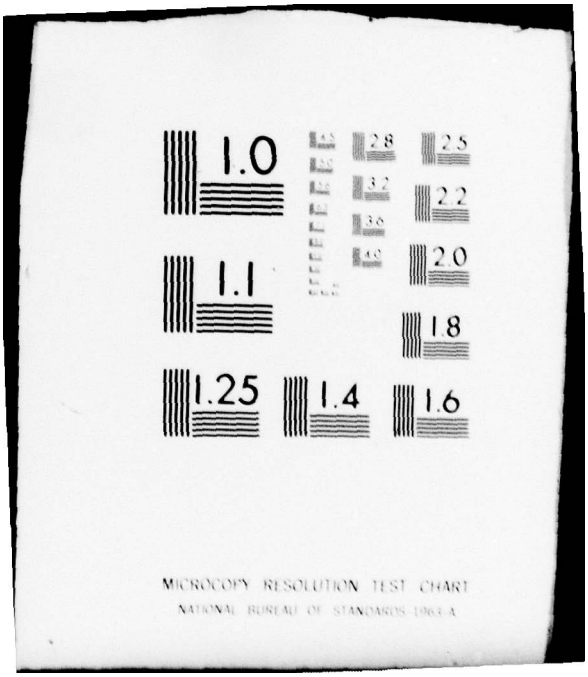
Position	Type	Position	Type
3-49	SN74157	3-56	Unused
3-50	SN74157	3-57	Unused
3-51	SN74157	3-58	Unused
3-52	SN74175	3-59	Unused
3-53	SN7408	3-60	Unused
3-54	SN74133	3-61	9408
3-55	SN74LS368		

AD-A080 169 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 13/8
DESIGN OF A DIGITAL CONTROLLER FOR AN ELECTRON BEAM LITHOGRAPHY--ETC(U)
DEC 79 M L WEIDNER
UNCLASSIFIED AFIT/GE/EE/79-41 NL

3 OF 3
AD A080169



END
DATE FILMED
3-80
DDC


1.0
1.1
1.25
1.4
1.6
2.8
3.2
3.6
2.5
2.2
2.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

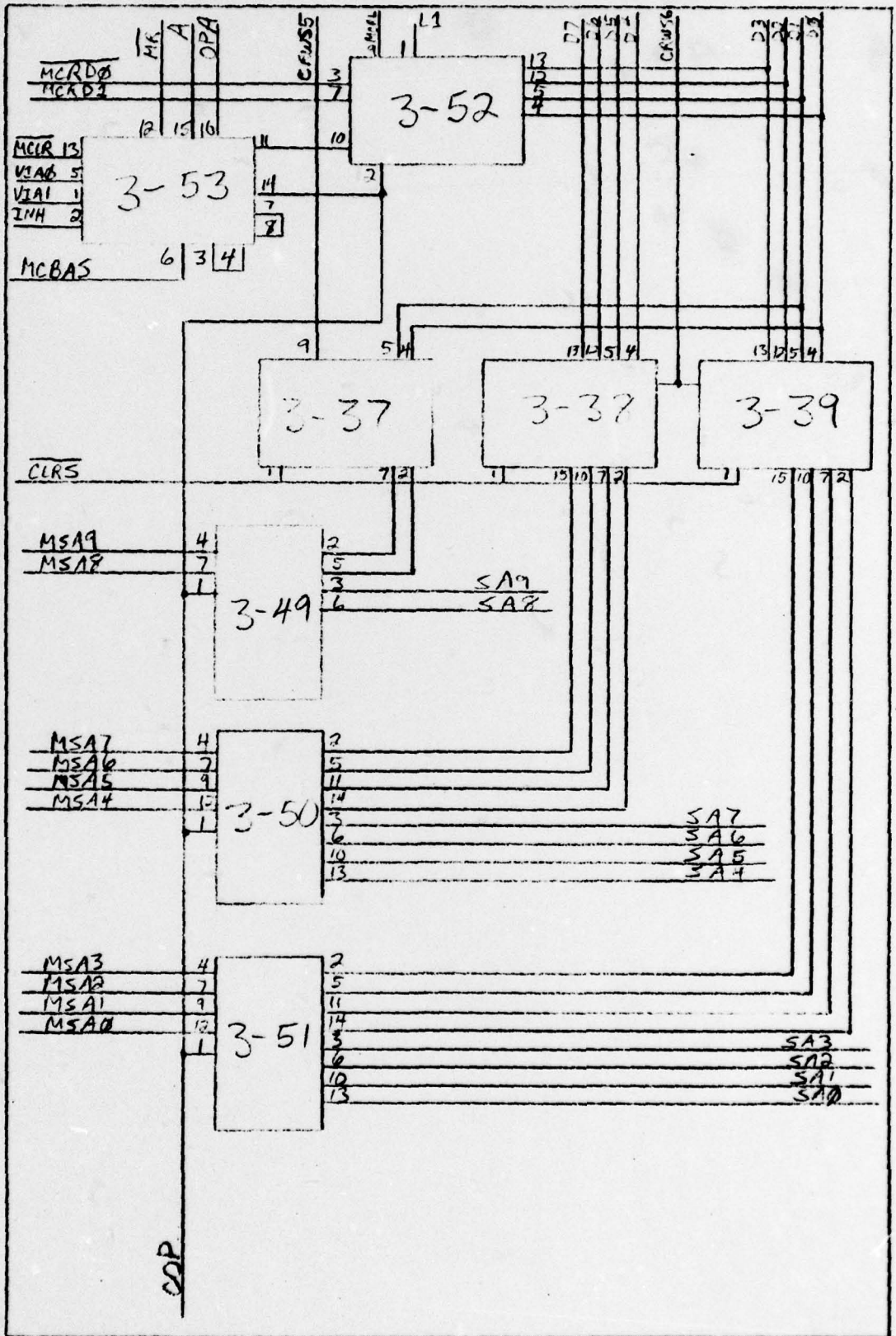


Fig 19 Micro-store Address Multiplexer

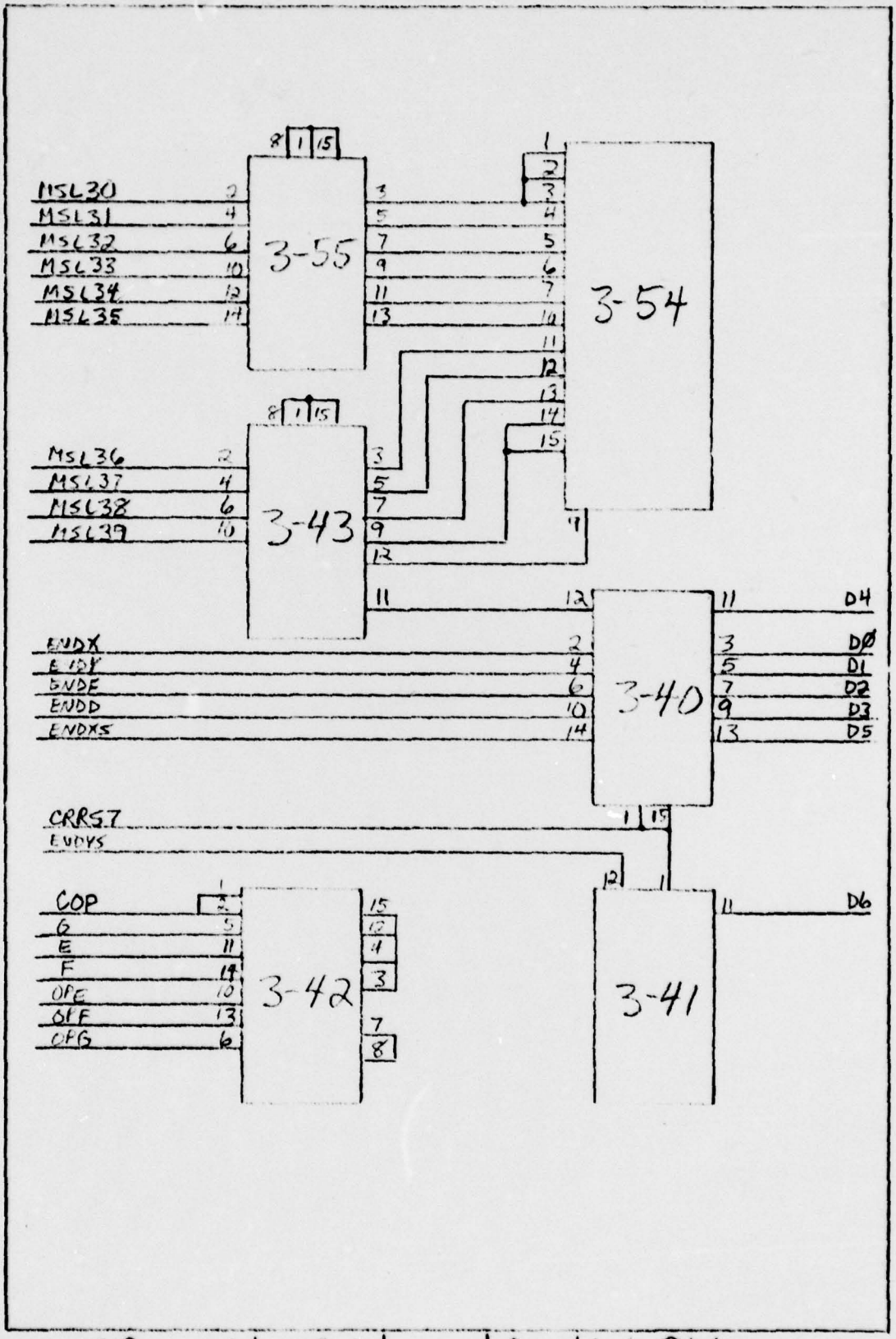


Fig 20 Status Register and Operational States

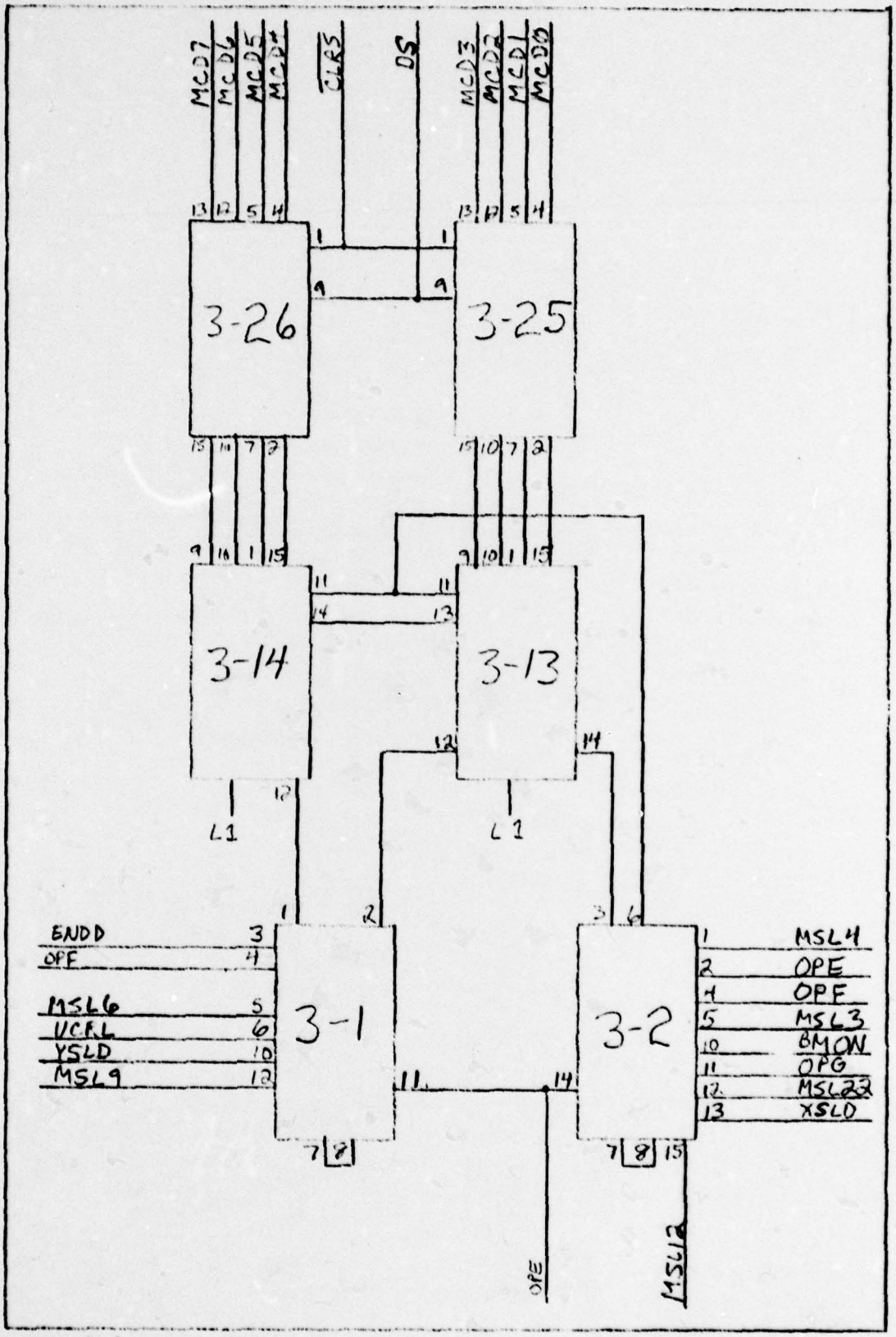


Fig 21 Delay Counter

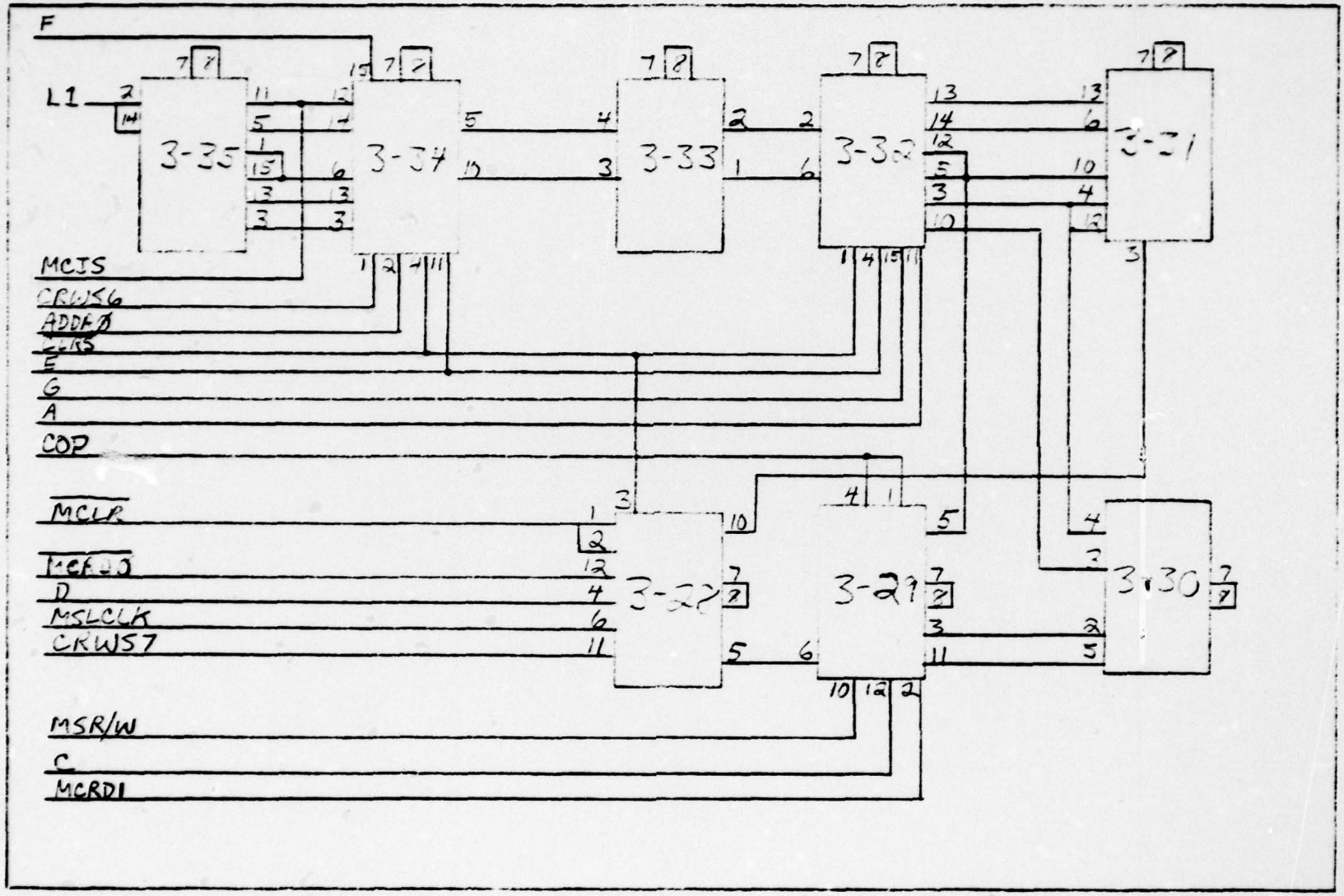


Fig 22 Instruction Select and Micro-store Read/Write Circuit

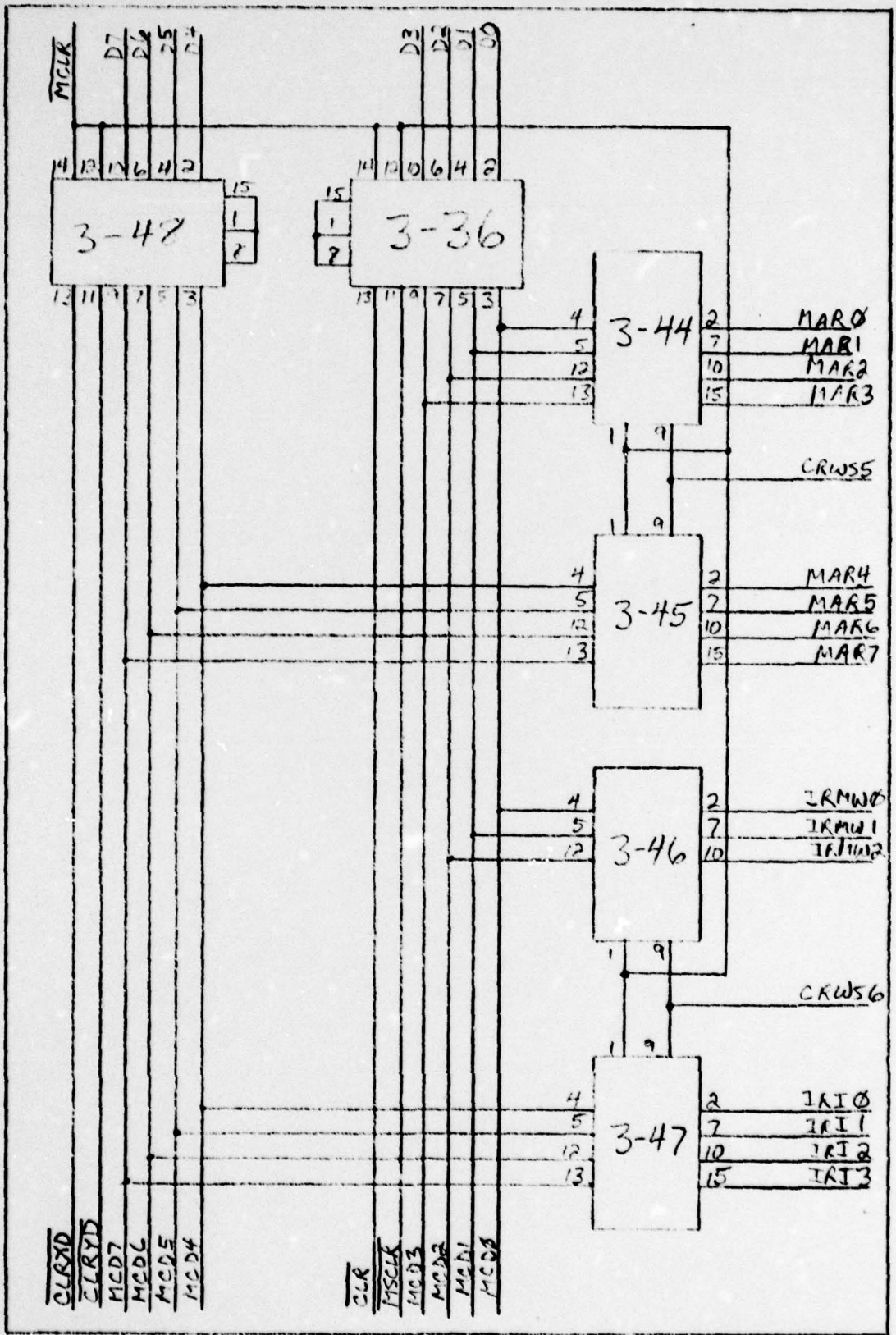


Fig 23 Board 3 Bus Interface and Instruction latch

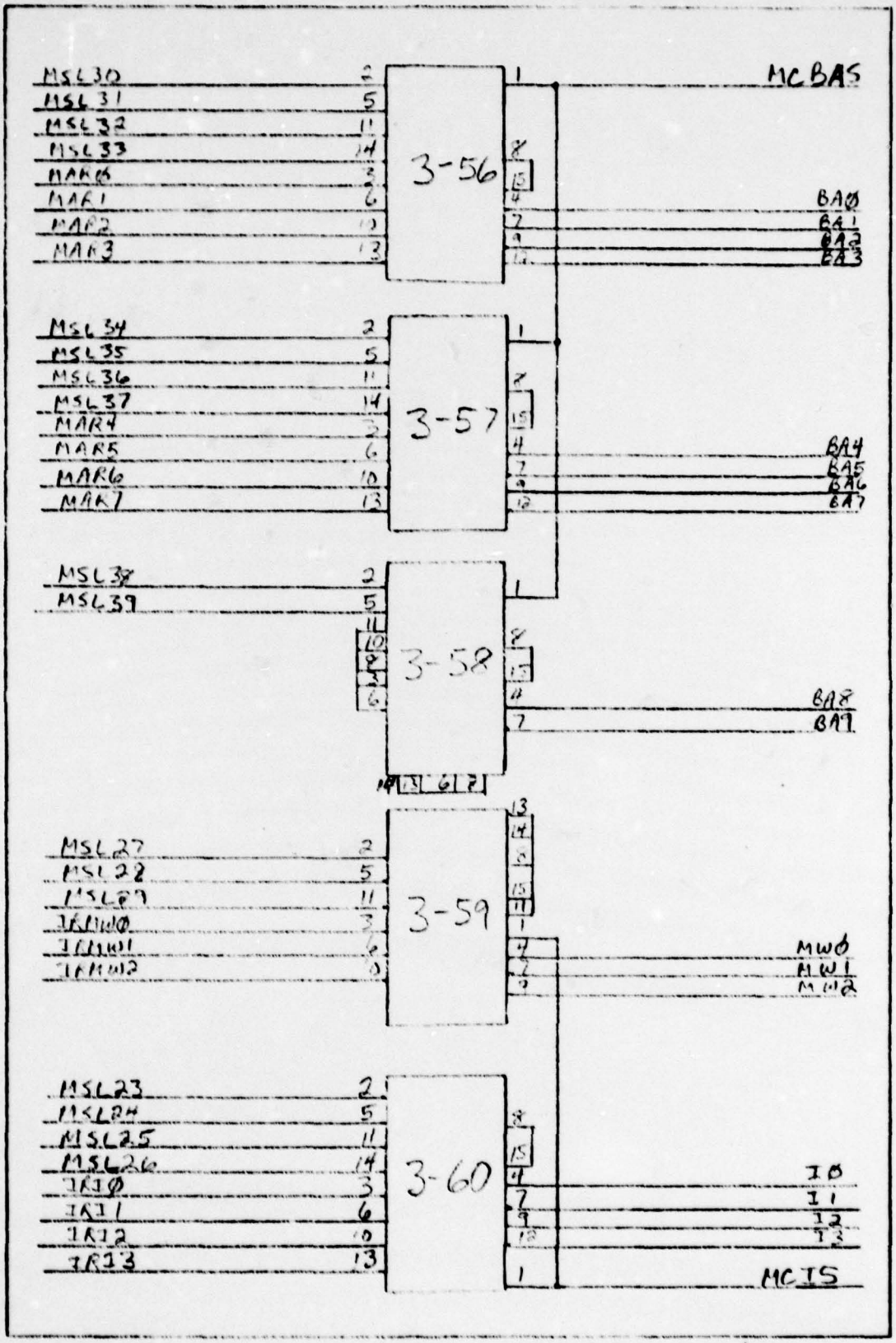


Fig 24 Branch Address and Instruction Multiplexers

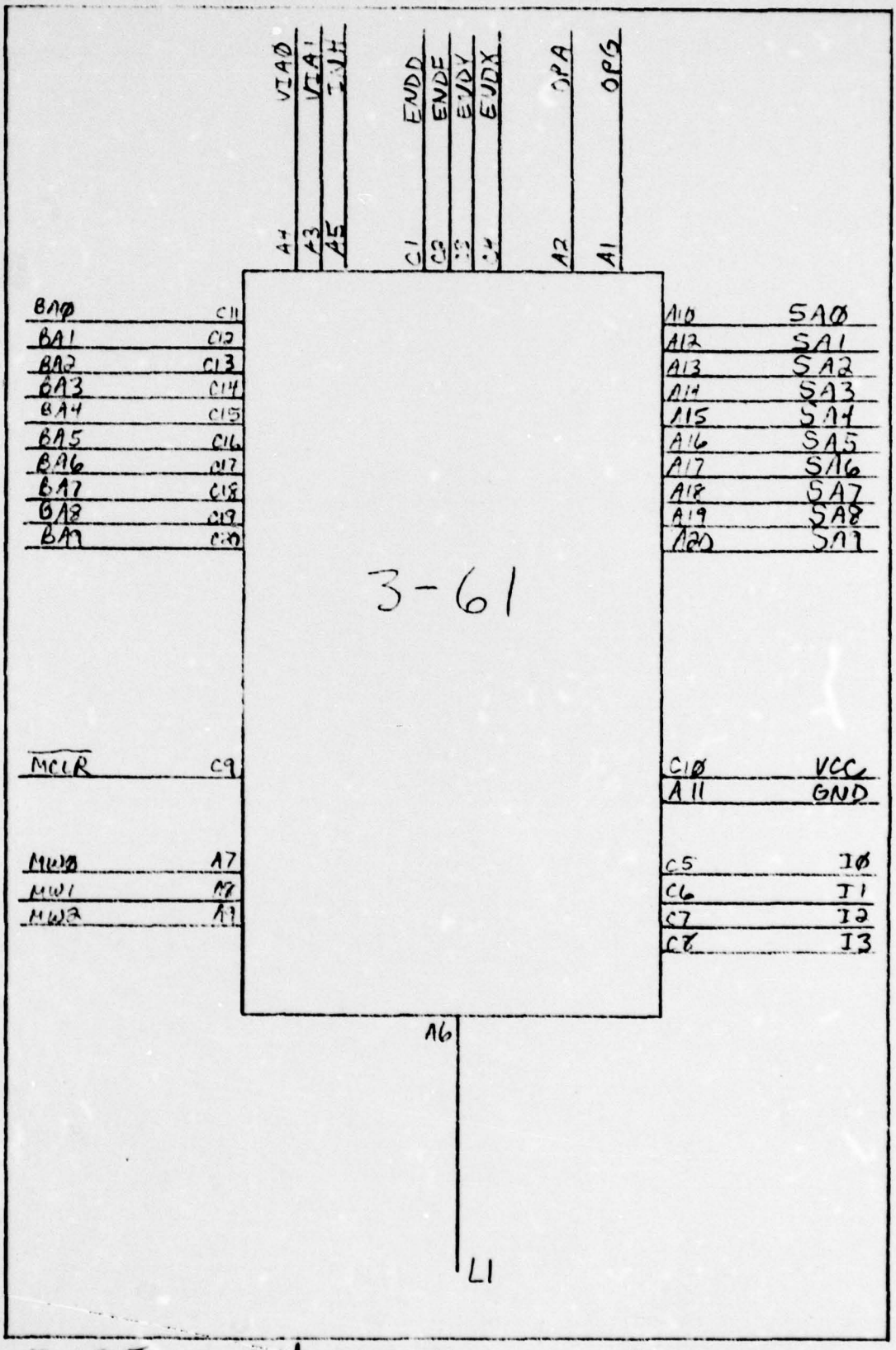


Fig 25 Fairchild 9407 Microprogram Sequencer

Pin to Pin Connections for Board 3

The Zilog Wire-wrap Board is preconfigured with 16 pin sockets with the V_{cc} Plane connected to pin 16 and with the GND Plane connected to pin 8. Note the following examples to read the Appendix.

IC	PIN	IC	PIN
5	9	11	6

This means a wire was physically connected to these two uniquely defined wire wrap posts.

IC	PIN	IC	PIN
5	7		A28

A wire was connected from IC 5 at pin 7 to Column A of the strip on the top of the wire wrap board pin number 28.

IC	PIN	IC	PIN
5	14		68

IC 5 pin 14 was connected to Edge connector pin number 68. The 14 pin dips are mounted with pin 1 in socket number one. This requires 2 be added to the right side (8-14) pin number to arrive at the adjusted pin number for the 16 pin socket. This has already been done. Consequently, you will note some connections to pins 15 and 16 of 14 pin dips.

Note: There are two columns of connections on each page.

Board 3

IC	PIN	IC	PIN	IC	PIN	IC	PIN
48	14	36	14	48	12	36	12
48	15	36	15	48	1	36	1
48	1	48	15	36	1	36	8
48	9	47	13	48	7	47	12
48	5	47	5	48	3	47	4
48	9	45	13	48	7	45	12
48	5	45	5	48	3	45	4
36	9	44	13	36	7	44	12
36	5	44	5	36	3	44	4
36	7	46	12	36	5	46	5
36	3	46	4	44	1	45	1
44	9	45	9	36	14	36	12
45	1	46	1	47	1	46	1
47	9	46	9	47	1	48	12
52	13	39	13	52	12	39	12
52	5	39	5	52	4	39	4
25	13	39	13	25	12	39	12
25	5	39	5	25	4	39	4
25	5	37	5	25	4	37	4
52	13	44	13	52	12	44	12
52	5	44	5	52	4	44	4
26	13	38	13	26	12	38	12
26	5	38	5	26	4	38	4
38	13	45	13	38	12	45	12

Board 3

IC	PIN	IC	PIN	IC	PIN	IC	PIN
38	5	45	5	38	4	45	4
	A21		GND		A28		V_{cc}
	A24		A25		A24		A6
13	5	14	5	35	2	35	14
52	1	13	5	35	2	14	5
52	1		A25	44	2	56	3
44	7	56	3	44	10	56	10
44	15	56	13	45	2	57	3
45	7	57	6	45	10	57	10
45	15	57	13	46	2	59	3
46	7	59	6	46	10	59	10
47	2	60	3	47	7	60	6
47	10	60	10	47	15	60	13
56	8	56	15	57	8	57	15
58	8	58	15	59	8	59	15
60	8	60	15	43	8	43	1
43	1	43	15	55	8	55	1
55	1	55	15	55	2	56	2
55	4	56	5	55	6	56	11
55	10	56	14	55	12	57	2
55..	14	57	5	43	2	57	11
43	4	57	14	43	6	58	2
43	10	58	5	56	2	58	37
56	5	58	38	56	11	58	39

Board 3

IC	PIN	IC	PIN	IC	PIN	IC	PIN
56	14	58	40	57	2	58	41
57	5	58	42	57	11	58	43
57	14	58	44	58	2	58	45
58	5	58	46	59	2	58	47
59	5	58	48	59	11	58	49
58	3	58	8	58	6	58	15
58	3	58	15	60	2		50
60	5		51	60	11		52
60	14		53	55	3	54	4
55	5	54	5	55	7	54	6
55	9	54	7	55	11	54	10
55	13	54	11	43	3	54	12
43	5	54	13	43	7	54	14
43	9	54	15	43	12	54	9
54	1	54	4	54	2	54	5
54	3	54	6	43	11	40	12
43	11	34	2	56	1	57	1
57	1	58	1	56	1	53	6
56	4		C11	56	7		C12
56	9		C13	56	12		C14
57	4		C15	57	7		C16
57	9		C17	57	12		C18
58	4		C19	58	7		C20
59	4		A7	59	7		A8

Board 3

IC	PIN	IC	PIN	IC	PIN	IC	PIN
59	9		A9	59	1	60	1
35	11	34	12	59	1	35	11
48	10		119	48	6		118
48	4		117	48	2		116
36	10		115	36	6		114
36	4		113	36	2		112
48	11		58	48	13		57
36	11		56	36	13		55
44	9		105	46	9		106
60	4		C5	60	7		C6
60	9		C7	60	12		C8
42	15	42	3	42	12	42	4
42	15	42	4	42	1	42	2
42	5	32	15	42	1	29	4
42	11	34	11	42	14	34	15
42	6		24	42	13		25
42	10		26	35	5	34	14
35	15	34	6	35	13	34	13
35	3	34	3	35	1	35	15
35	7	35	8	34	1	47	9
34	4	32	1	26	1	25	1
28	3	26	1	28	3	32	1
25	1	37	1	38	1	39	1
37	1	38	1	37	1		105

Board 3

IC	PIN	IC	PIN	IC	PIN	IC	PIN
34	11		30	34	15		29
34	11	32	4	34	5	33	4
34	10	33	3	34	7	34	8
33	7	33	8	32	7	32	8
29	7	29	8	28	7	28	8
33	1	32	6	33	2	32	2
32	15		28	32	11		33
32	13	31	13	32	14	31	6
32	10	30	3	32	5	32	12
32	5	31	10	32	3	31	4
31	4	31	12	31	12	30	4
31	10	29	5	30	2	29	3
30	5	29	11	29	4	29	1
29	1	42	1	29	4	53	14
53	14	52	2	49	1	50	1
51	1	52	2	50	1	51	1
29	4		23	29	10		81
29	12		32	29	6	28	5
31	7	31	2	31	2	31	8
31	3	28	10	28	12	52	3
28	4		31	28	6		80
28	11		107	28	1	28	2
28	1	44	1	28	1	53	13
53	19		C9	53	12		54

Board 3

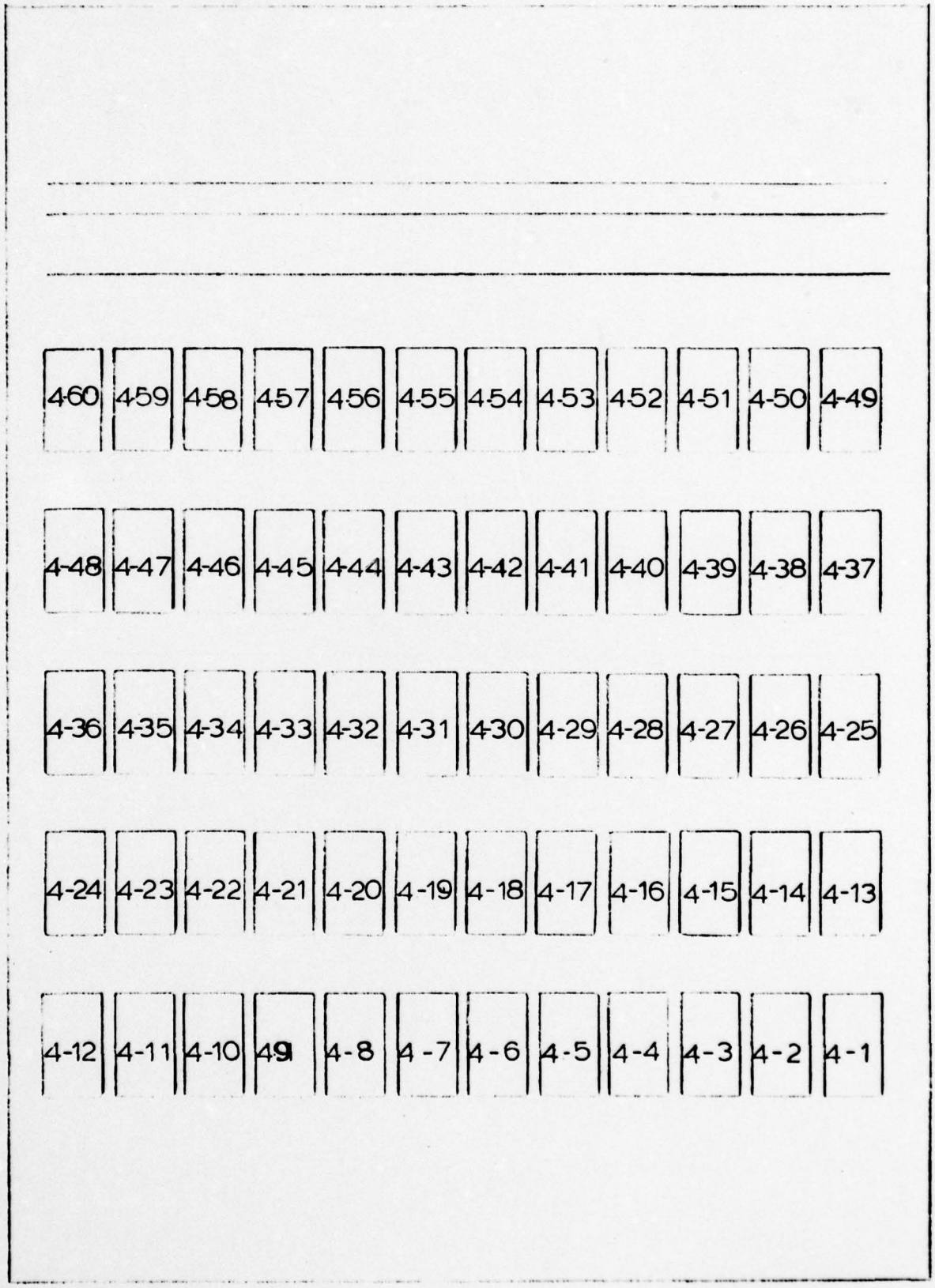
IC	PIN	IC	PIN	IC	PIN	IC	PIN
53	15	32	11	53	7	53	8
53	3	53	4	53	5		A4
53	1		A3	53	2		A5
53	11	52	10	53	10		A2
42	6		A1	40	10		C1
40	6		C2	40	4		C3
40	2		C4	40	10	1	10
40	6		36	40	4		35
40	2		34	40	14		78
40	1	40	15	40	15	41	15
40	1		77	40	3		112
40	5		113	40	7		114
40	9		115	40	11		116
40	13		117	41	12		79
41	11		118	41	13		119
37	1		59	37	9		105
37	7	49	5	37	2	49	2
38	9	39	9	38	9		106
38	15	50	14	38	10	50	11
38	7	50	5	38	2	50	2
39	15	51	14	39	10	51	11
39	7	51	5	39	2	51	2
51	3		A10	51	6		A12
51	10		A13	51	13		A14

Board 3

IC	PIN	IC	PIN	IC	PIN	IC	PIN
50	3		A15	50	6		A16
50	10		A17	50.	13		A18
49	3		A19	49	6		A20
	A11		GND	51	8	51	15
50	8	50	15	49	8	49	15
51	4		10	51	7		11
51	9		12	51	12		13
50	4		14	50	7		15
50	9		16	50	12		17
49	4		18	49	7		19
26	9	25	9	25	9		66
26	15	14	9	26	10	14	10
26	7	14	1	26	2	14	15
25	15	13	9	25	10	13	10
25	7	13	1	25	2	13	15
14	14	1	6	14	4	14	8
14	12	1	11	14	11	13	11
13	13	1	5	13	12	1	12
13	4	13	8	13	11	2	6
13	14	2	3	1	1	1	11
1	2	1	11	1	3	1	4
1	7	1	8	2	4	42	13
2	5		68	2	1		67
2	2	42	10	2	7	2	8

Board 3

IC	PIN	IC	PIN	IC	PIN	IC	PIN
1	16		1	13	16		2
2	16		3	14	16		4
1	8		62	13	8		63
2	8		64	14	8		65


460 459 458 457 456 455 454 453 452 451 450 449
448 447 446 445 444 443 442 441 440 439 438 437
436 435 434 433 432 431 430 429 428 427 426 425
424 423 422 421 420 419 418 417 416 415 414 413
412 411 410 49 48 47 46 45 44 43 42 41

Board 4 Integrated Circuits

Position	Type	Position	Type
4-1	Unused	4-25	Unused
4-2	Unused	4-26	Unused
4-3	SN74LS367	4-27	Unused
4-4	SN74LS367	4-28	Unused
4-5	SN74175	4-29	SN74157
4-6	SN74175	4-30	SN74191
4-7	SN74175	4-31	SN74157
4-8	SN74175	4-32	SN74191
4-9	SN74175	4-33	SN74157
4-10	SN74175	4-34	SN74191
4-11	SN74175	4-35	SN74157
4-12	SN74175	4-36	SN7432
4-13	SN7485	4-37	Unused
4-14	SN7485	4-38	Unused
4-15	SN7485	4-39	SN7408
4-16	SN7485	4-40	SN7432
4-17	SN74175	4-41	SN7432
4-18	SN74175	4-42	SN7408
4-19	SN74175	4-43	SN7432
4-20	SN74175	4-44	SN7408
4-21	SN74175	4-45	SN7402
4-22	SN74175	4-46	SN7404
4-23	SN74175	4-47	SN7408
4-24	SN74175	4-48	SN7432

Board 4 Integrated Circuits

Position	Type	Position	Type
4-49	Unused	4-56	SN7408
4-50	Unused	4-57	SN74138
4-51	Unused	4-58	SN74175
4-52	SN74111	4-59	SN7485
4-53	SN74136	4-60	SN7485
4-54	SN74136	4-61	Dip Switch(8)
4-55	SN74111	4-62	4114R

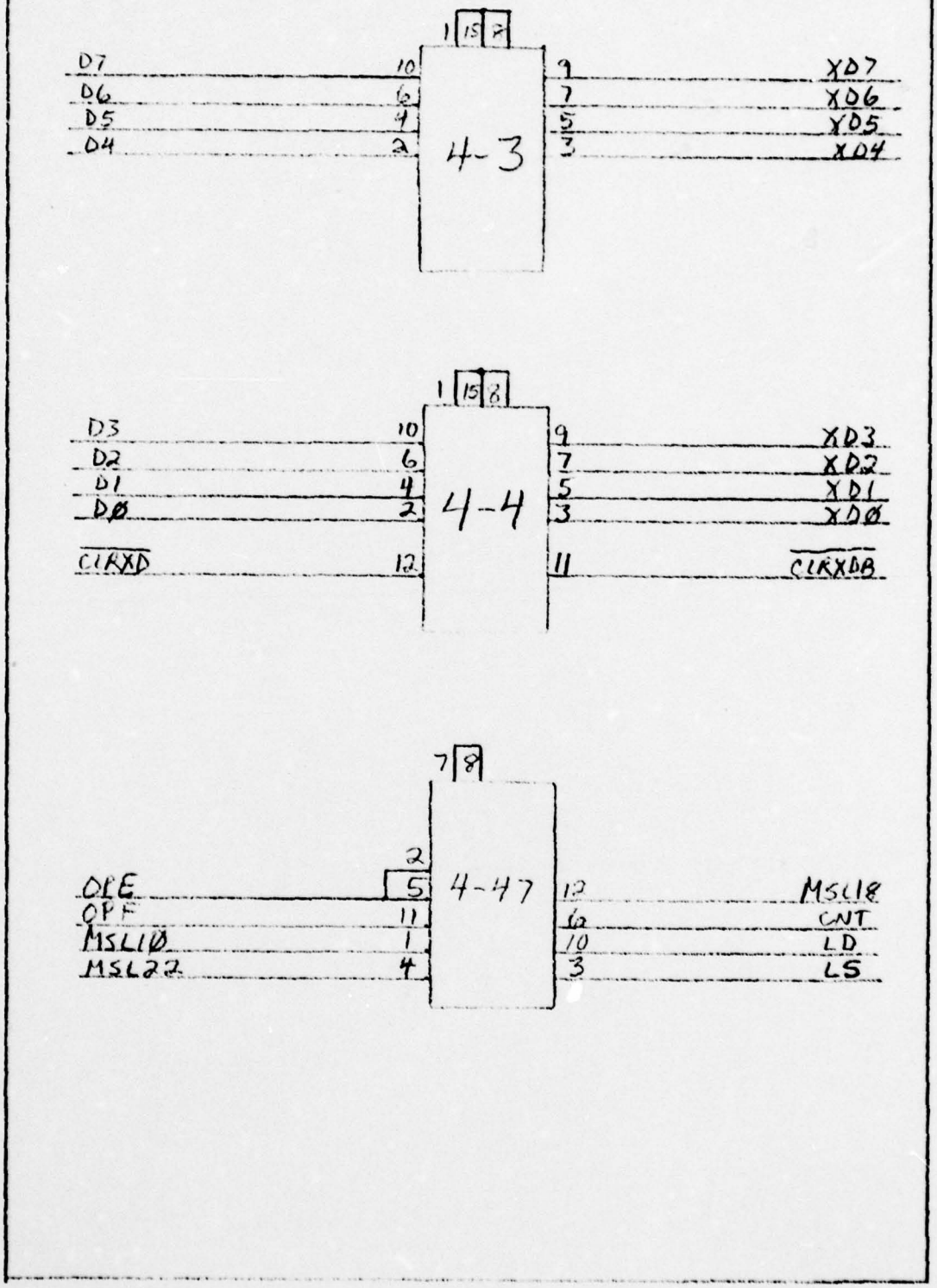


Fig 26 Bus Interface

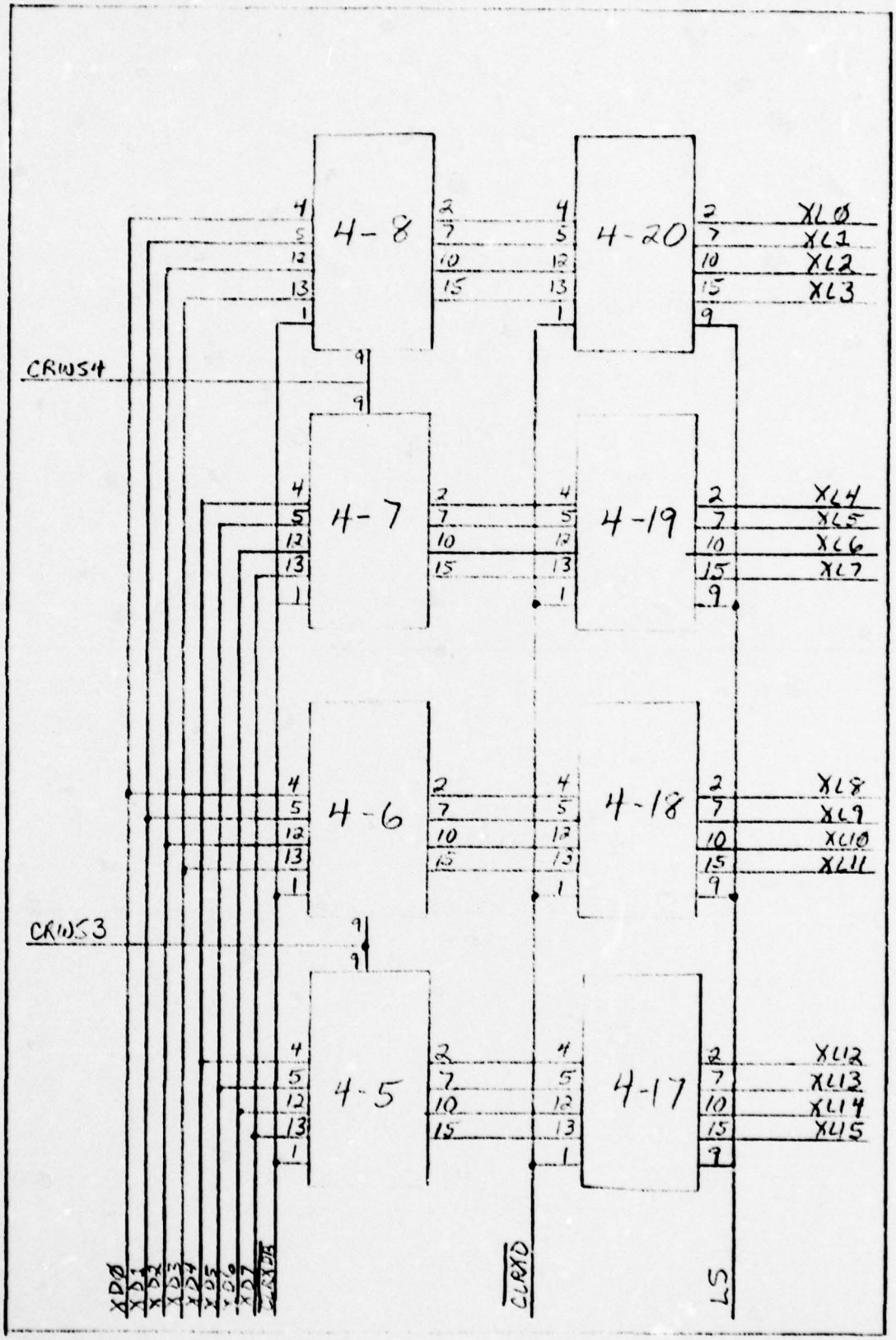


Fig. 27 X Deflection Low Address Register

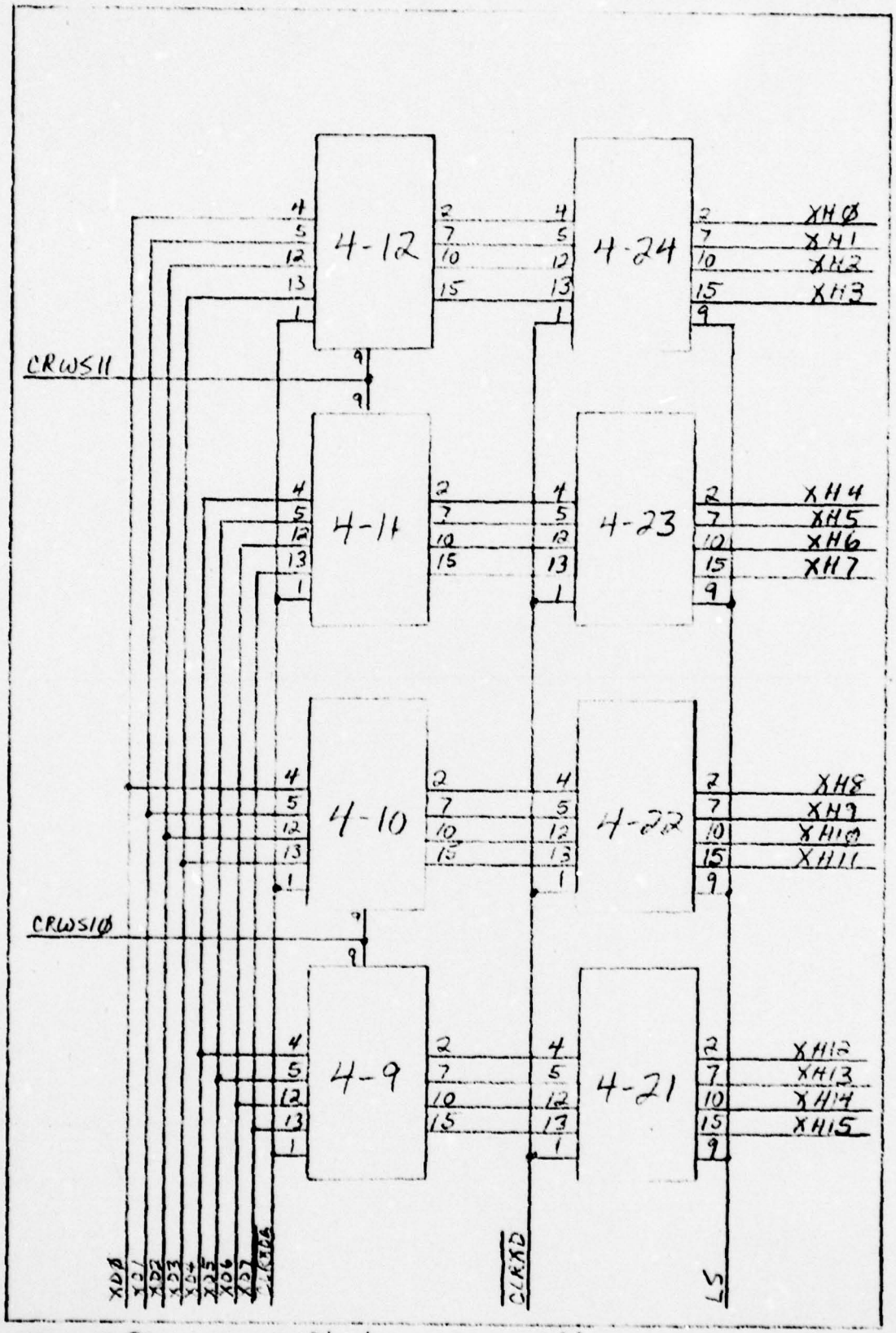


Fig. 28 X Deflection High Address Register

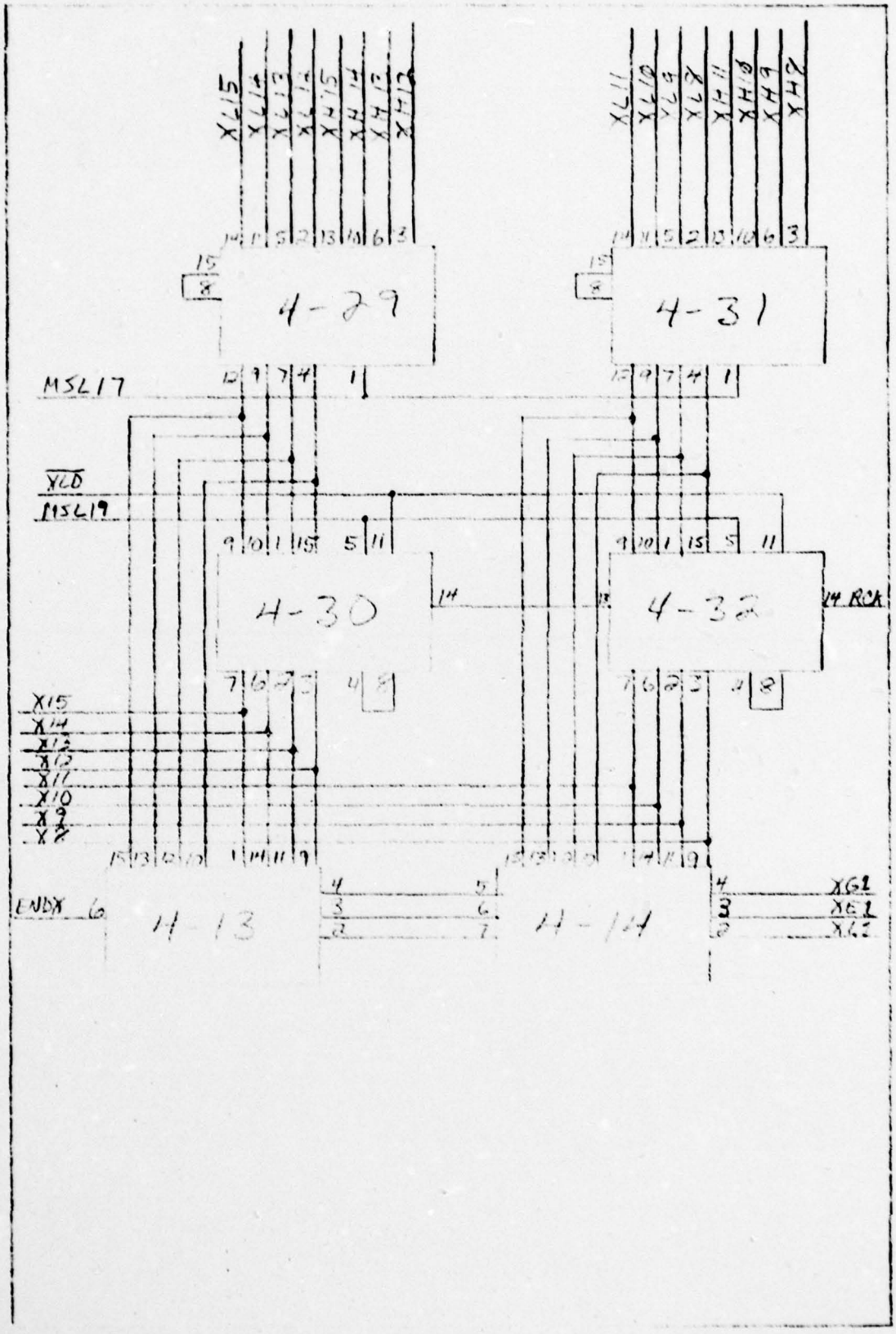


Fig. 29 X Deflection Counter

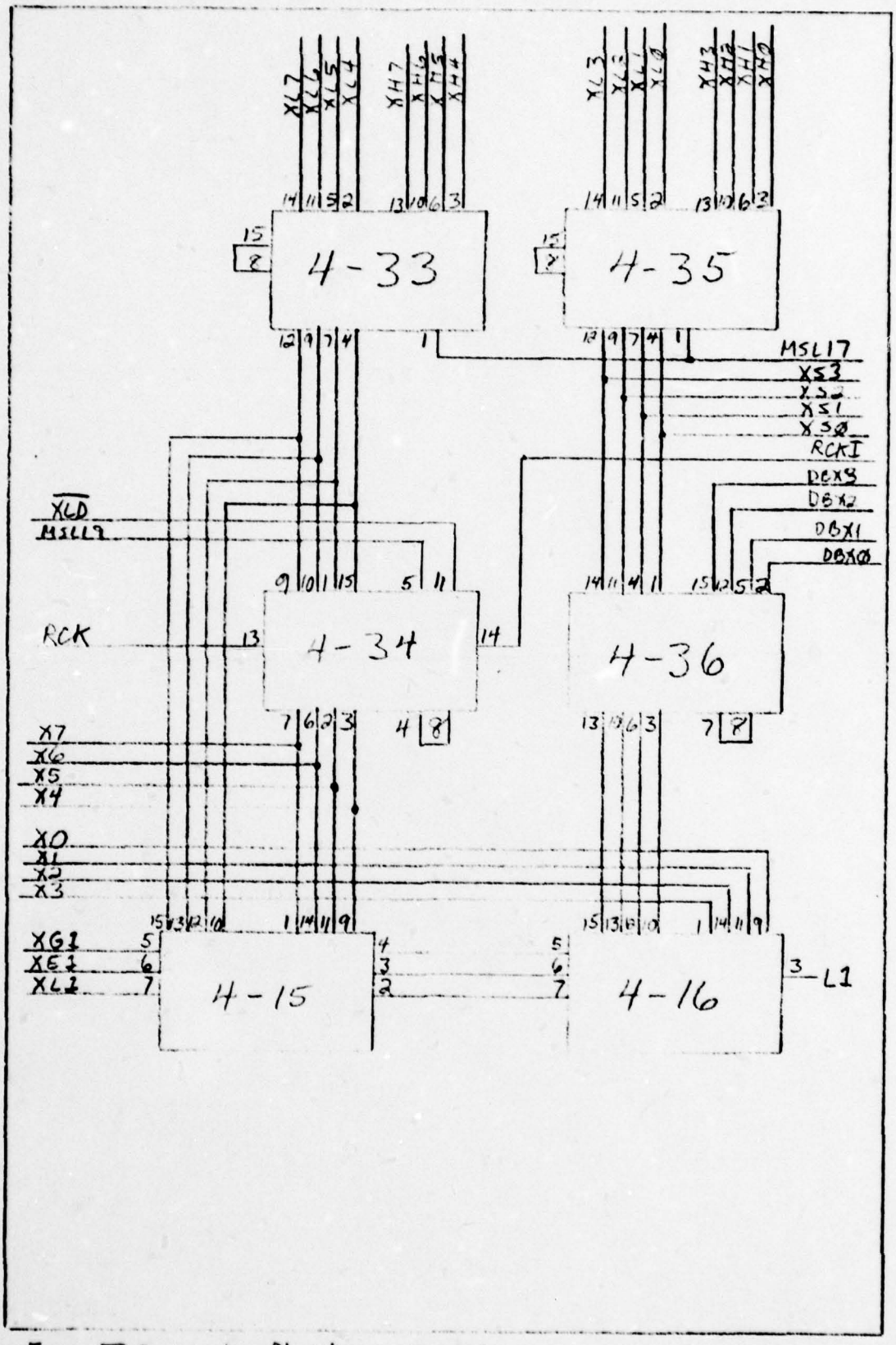


Fig. 30 X Deflection Counter

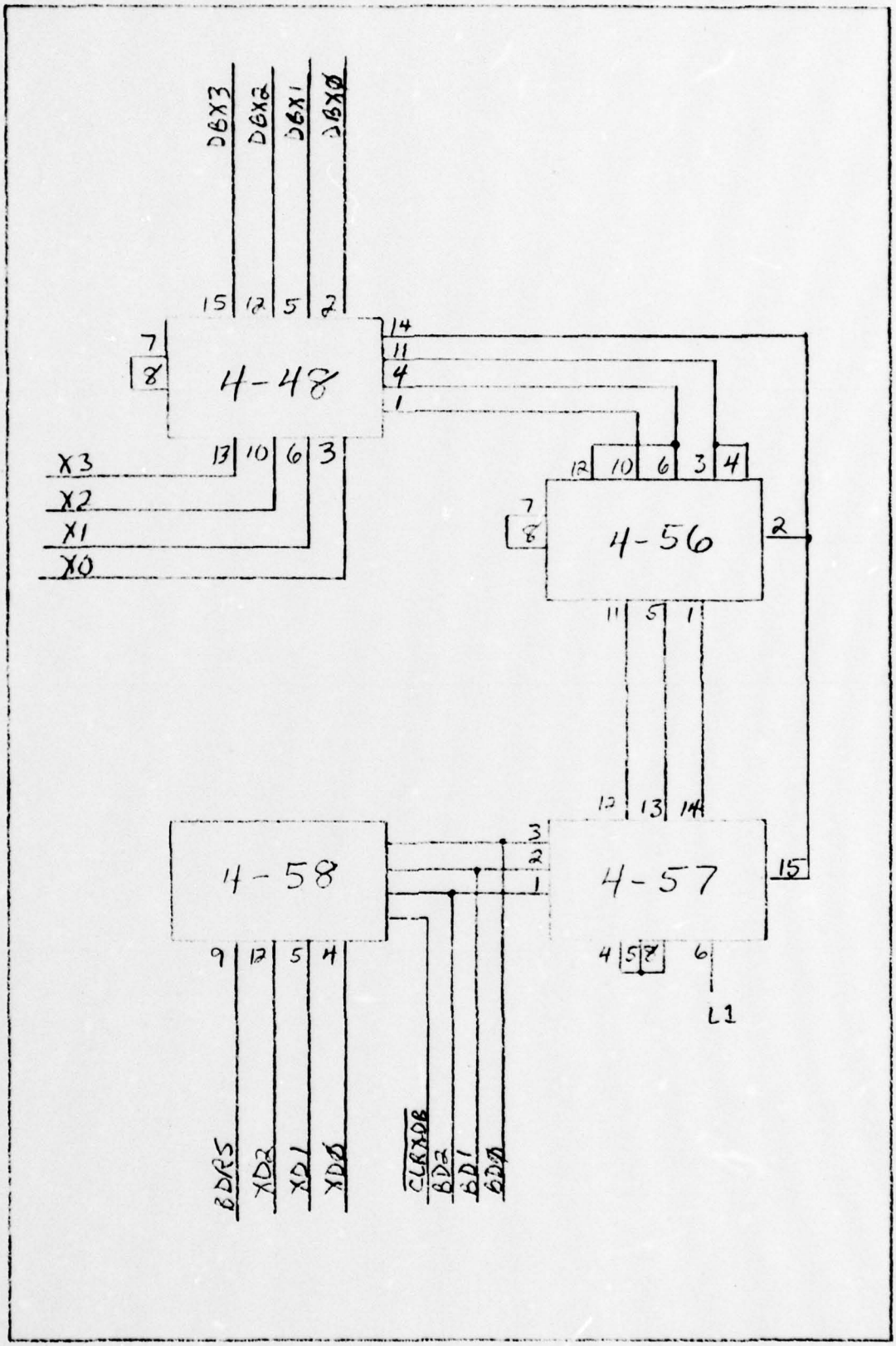


Fig. 31 X Deflection Counter

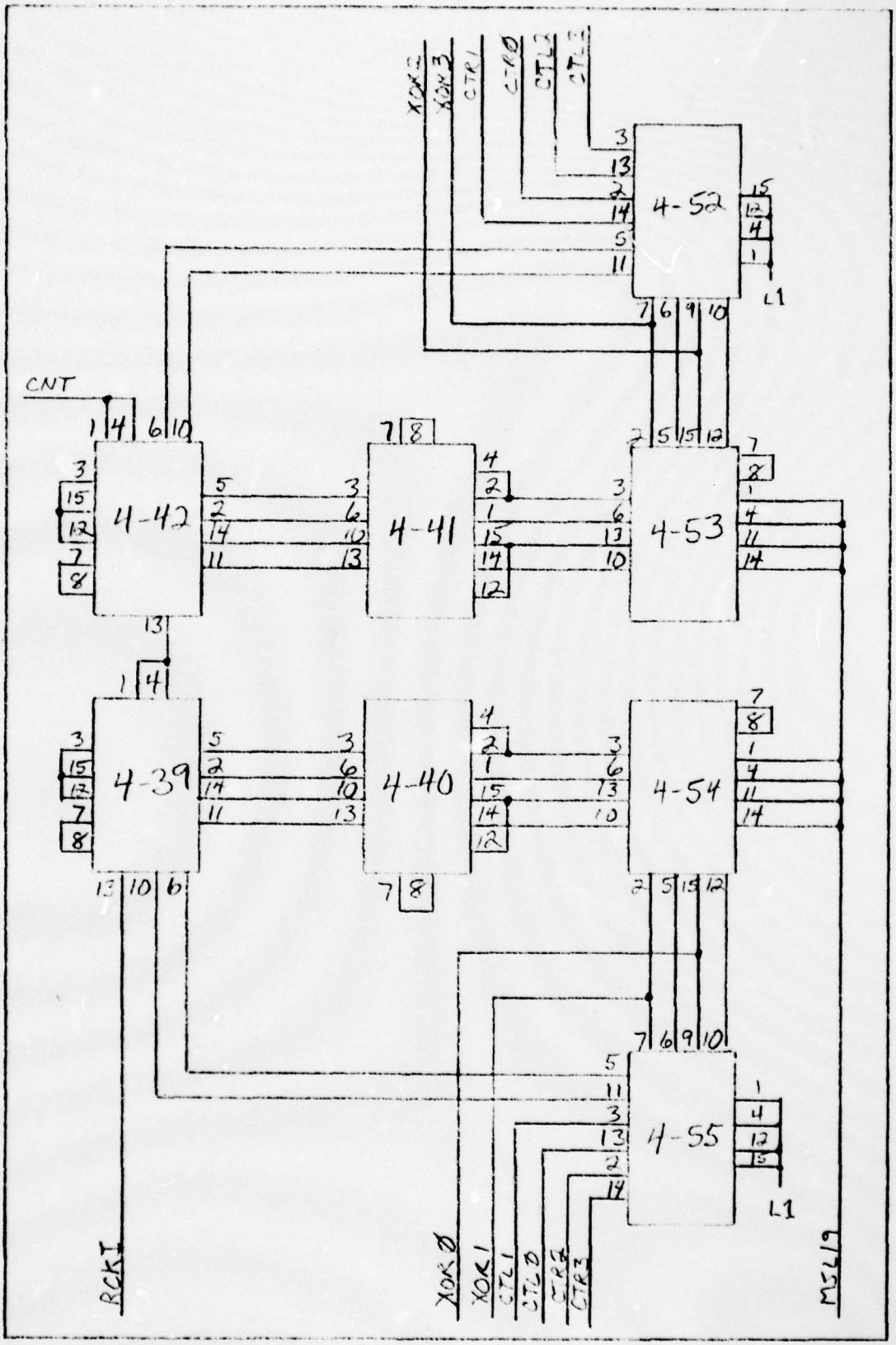


Fig. 32 X Deflection Counter

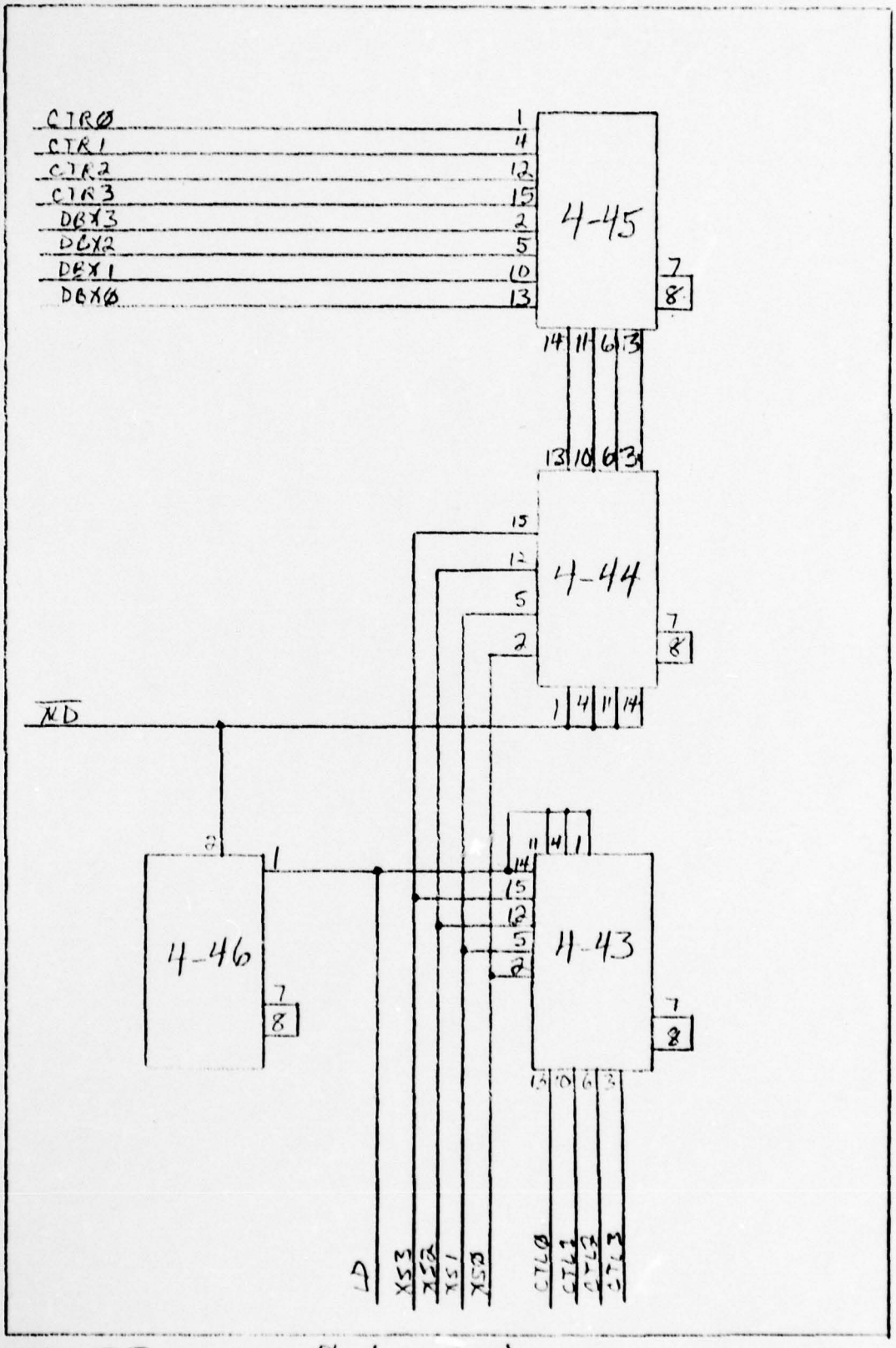


Fig 33 X Deflection Counter

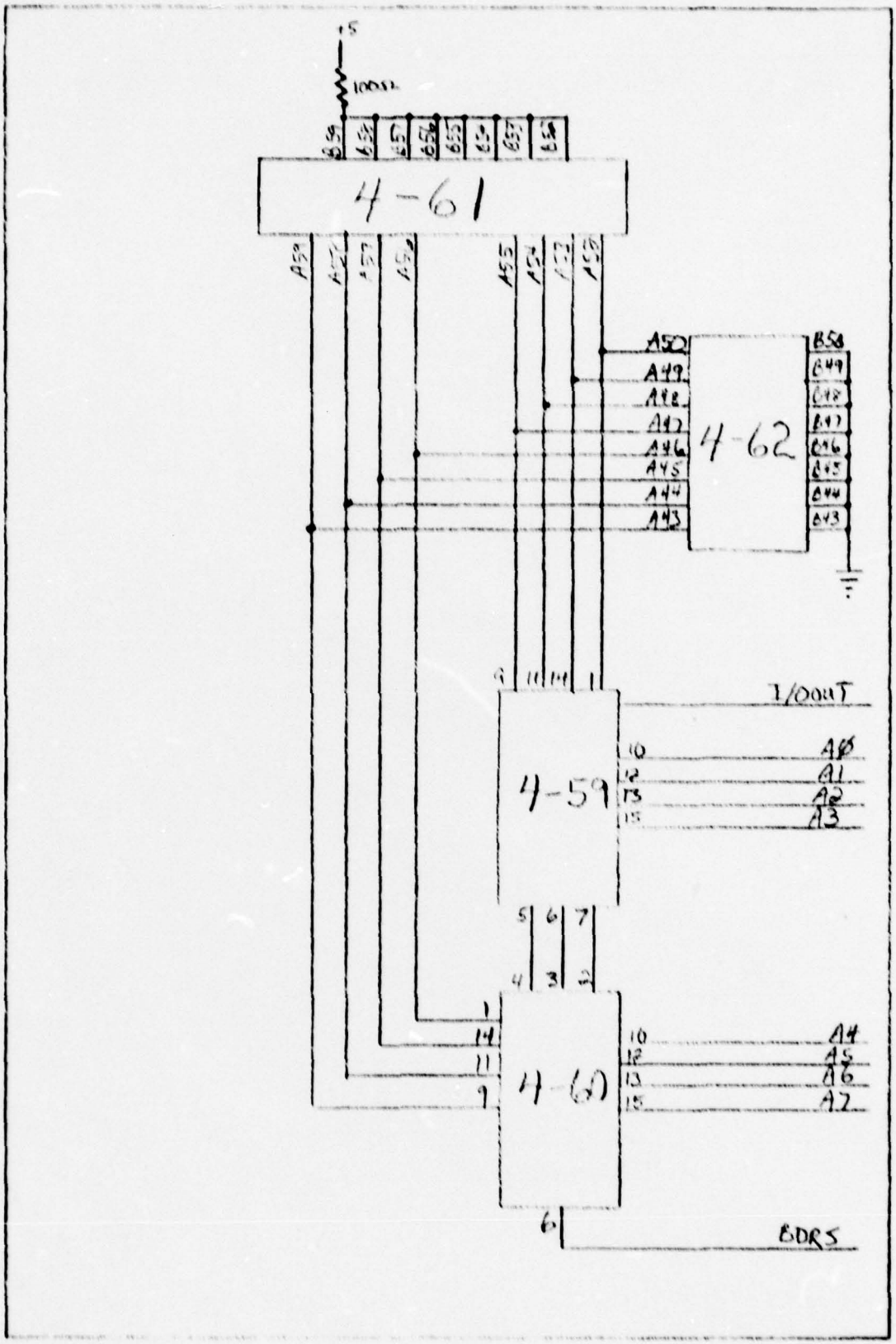
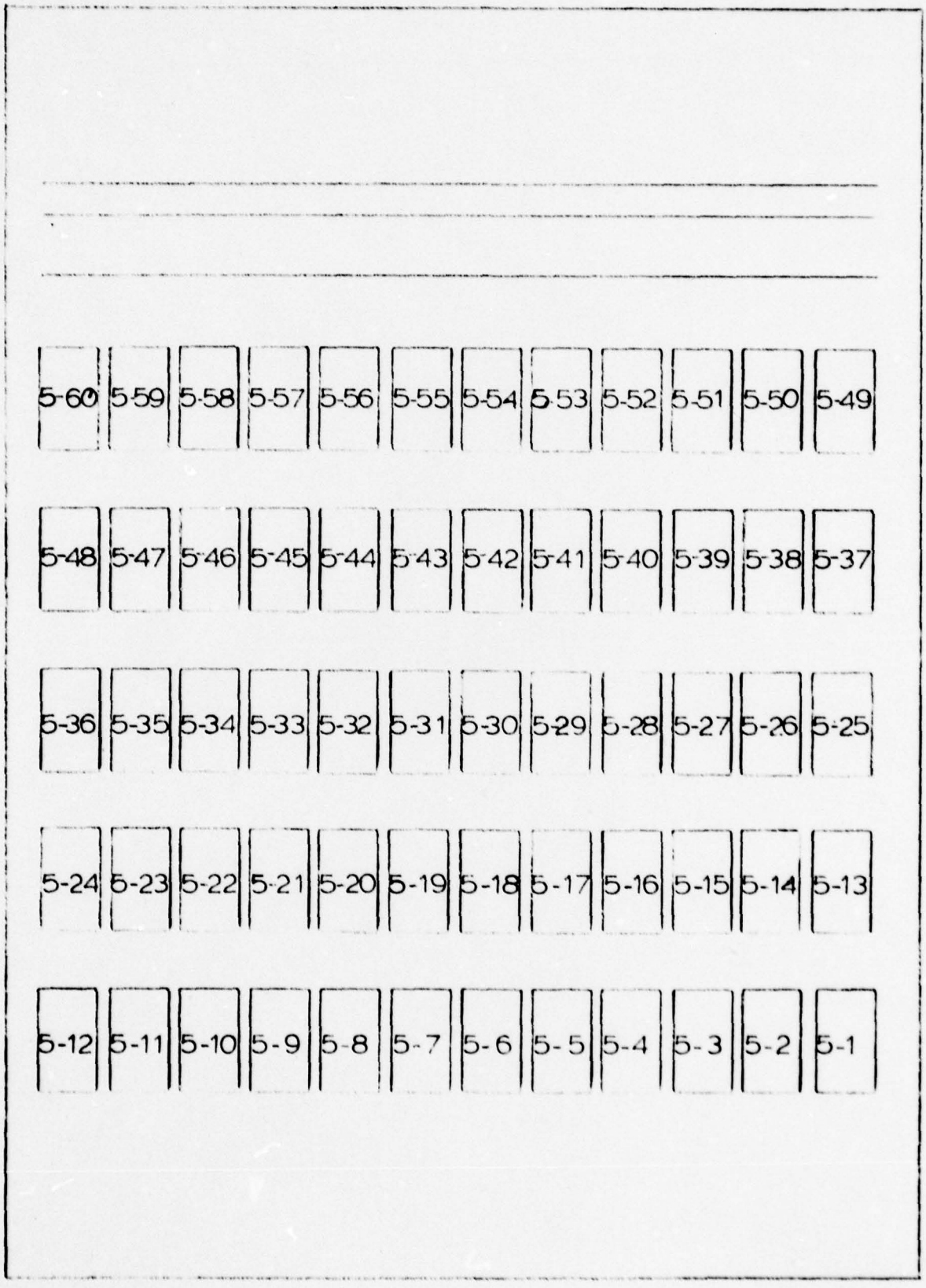


Fig. 34 Bit Density Register Select


5-60 5-59 5-58 5-57 5-56 5-55 5-54 5-53 5-52 5-51 5-50 5-49
5-48 5-47 5-46 5-45 5-44 5-43 5-42 5-41 5-40 5-39 5-38 5-37
5-36 5-35 5-34 5-33 5-32 5-31 5-30 5-29 5-28 5-27 5-26 5-25
5-24 5-23 5-22 5-21 5-20 5-19 5-18 5-17 5-16 5-15 5-14 5-13
5-12 5-11 5-10 5-9 5-8 5-7 5-6 5-5 5-4 5-3 5-2 5-1

Board 5 Integrated Circuits

Position	Type	Position	Type
5-1	Unused	5-25	Unused
5-2	Unused	5-26	Unused
5-3	SN74LS367	5-27	Unused
5-4	SN74LS367	5-28	Unused
5-5	SN74175	5-29	SN74157
5-6	SN74175	5-30	SN74191
5-7	SN74175	5-31	SN74157
5-8	SN74175	5-32	SN74191
5-9	SN74175	5-33	SN74157
5-10	SN74175	5-34	SN74191
5-11	SN74175	5-35	SN74157
5-12	SN74175	5-36	SN7432
5-13	SN7485	5-37	Unused
5-14	SN7485	5-38	Unused
5-15	SN7485	5-39	SN7408
5-16	SN7485	5-40	SN7432
5-17	SN74175	5-41	SN7432
5-18	SN74175	5-42	SN7408
5-19	SN74175	5-43	SN7432
5-20	SN74175	5-44	SN7408
5-21	SN74175	5-45	SN7402
5-22	SN74175	5-46	SN7404
5-23	SN74175	5-47	SN7408
5-24	SN74175	5-48	SN7432

Board 5 Integrated Circuits

Position	Type	Position	Type
5-49	Unused	5-55	SN74111
5-50	Unused	5-56	SN7408
5-51	Unused	5-57	SN74138
5-52	SN74111	5-58	Unused
5-53	SN74136	5-59	Unused
5-54	SN74136	5-60	Unused

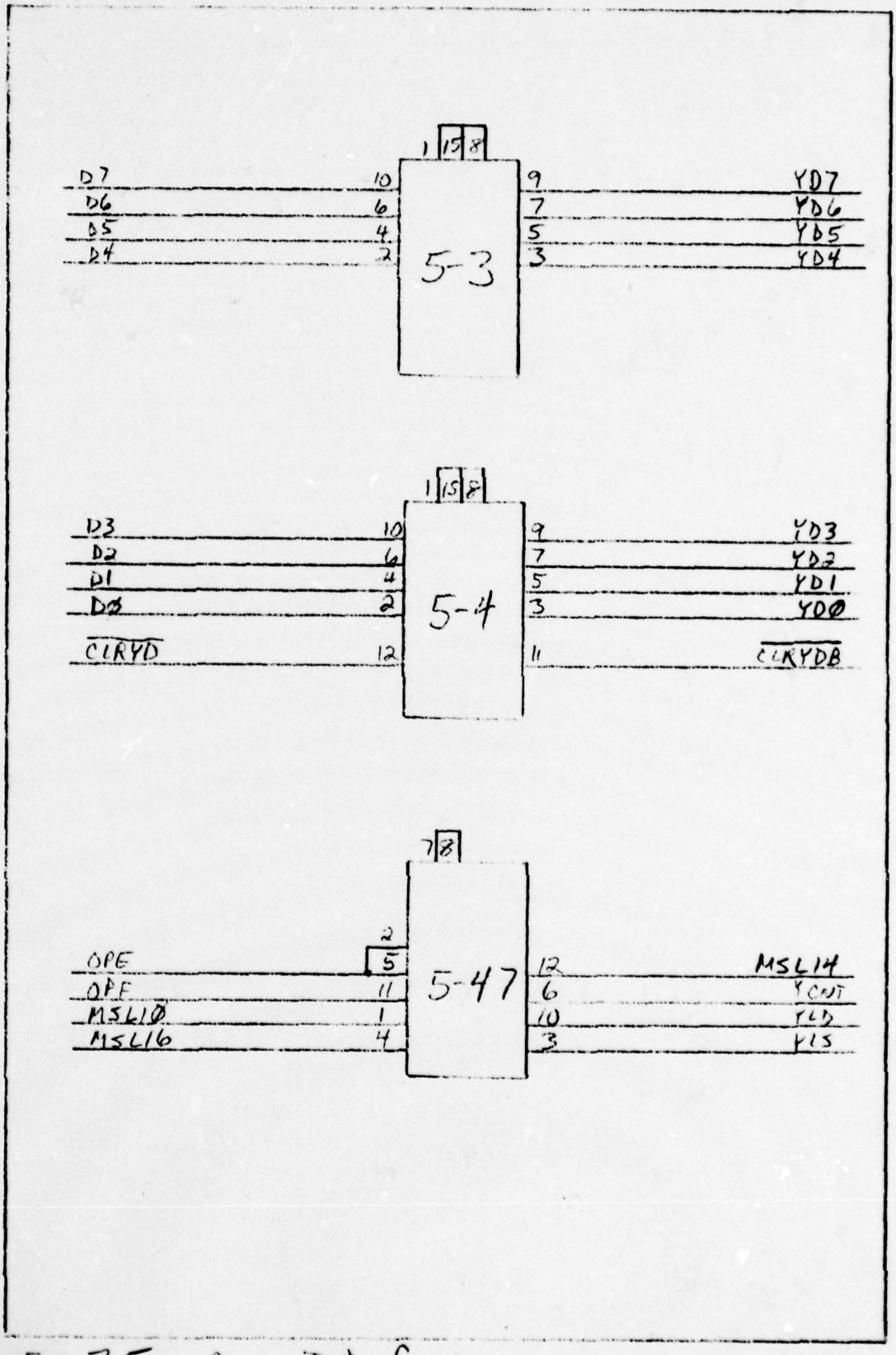


Fig 35 Bus Interface

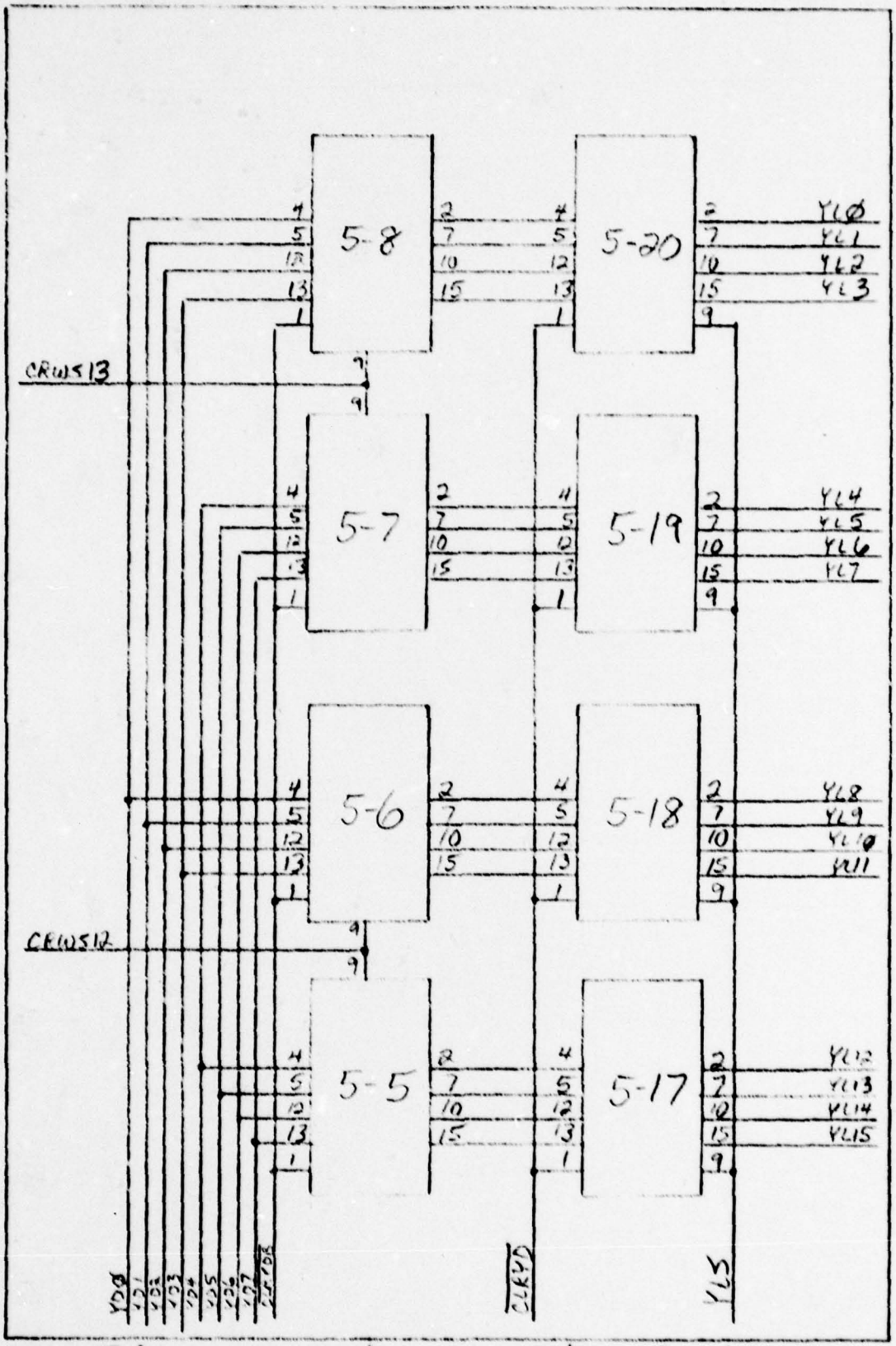


Fig 36 Y Deflection Low Address Register

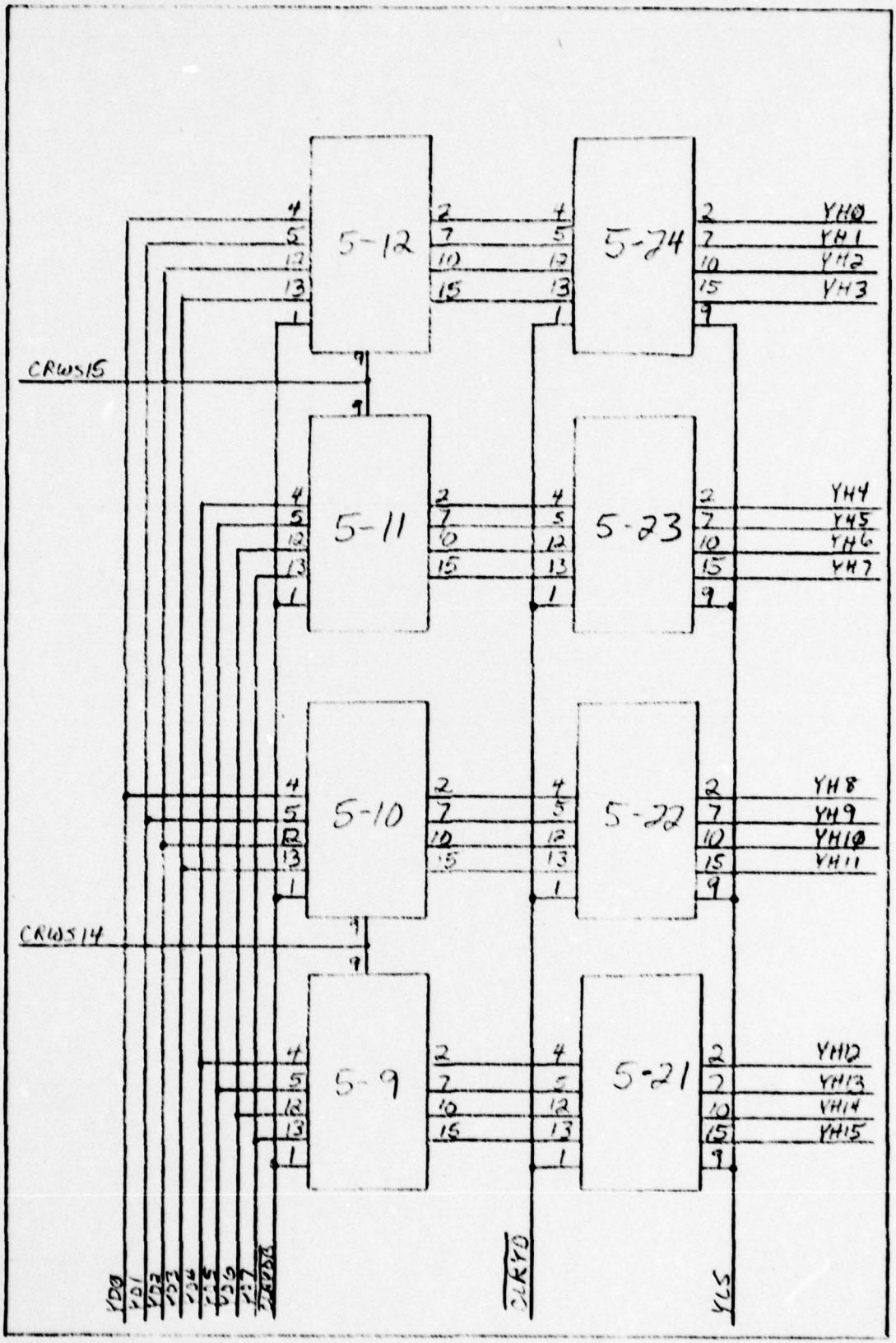


Fig. 37 Y Deflection High Address Register

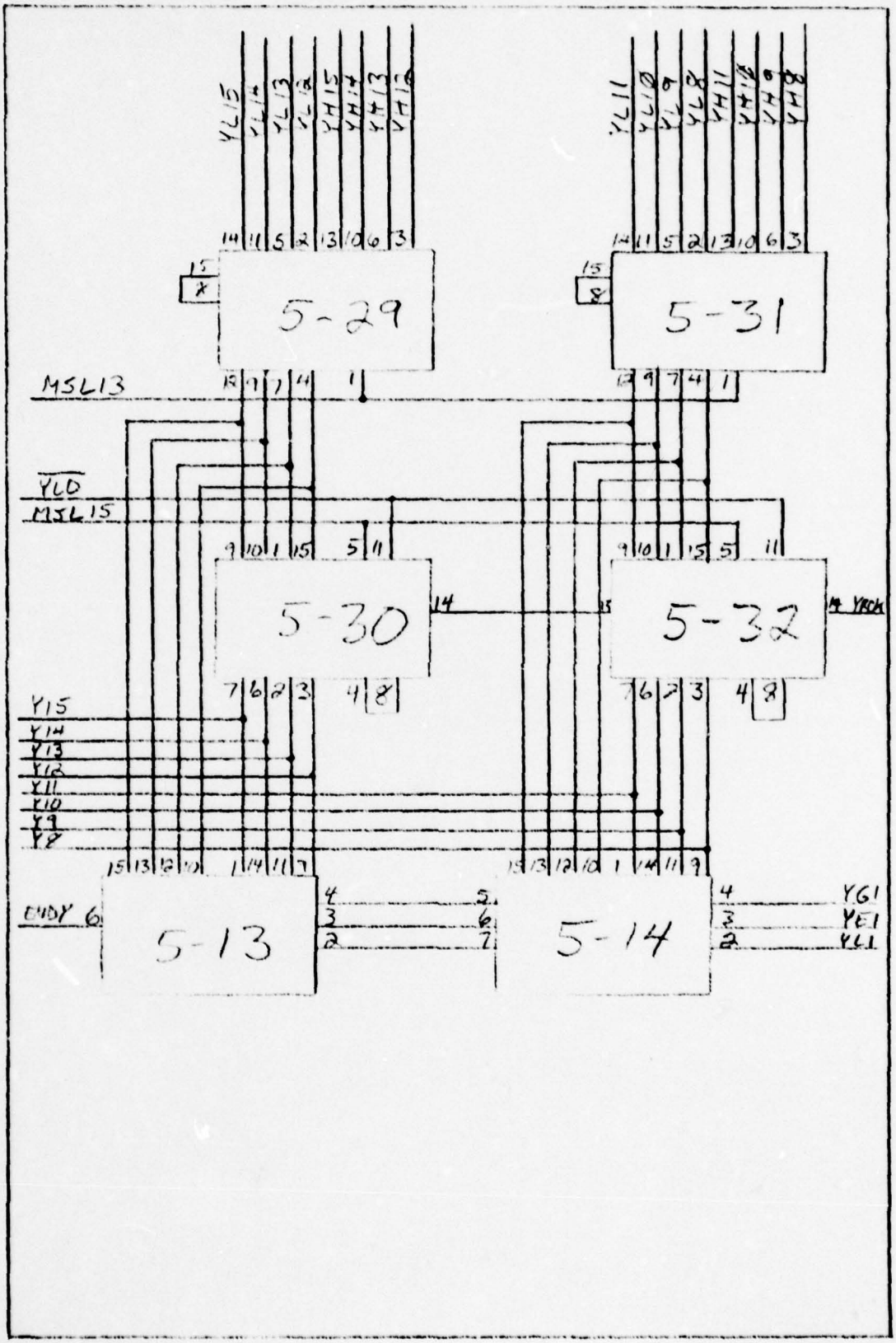


Fig 38 Y Deflection Counter

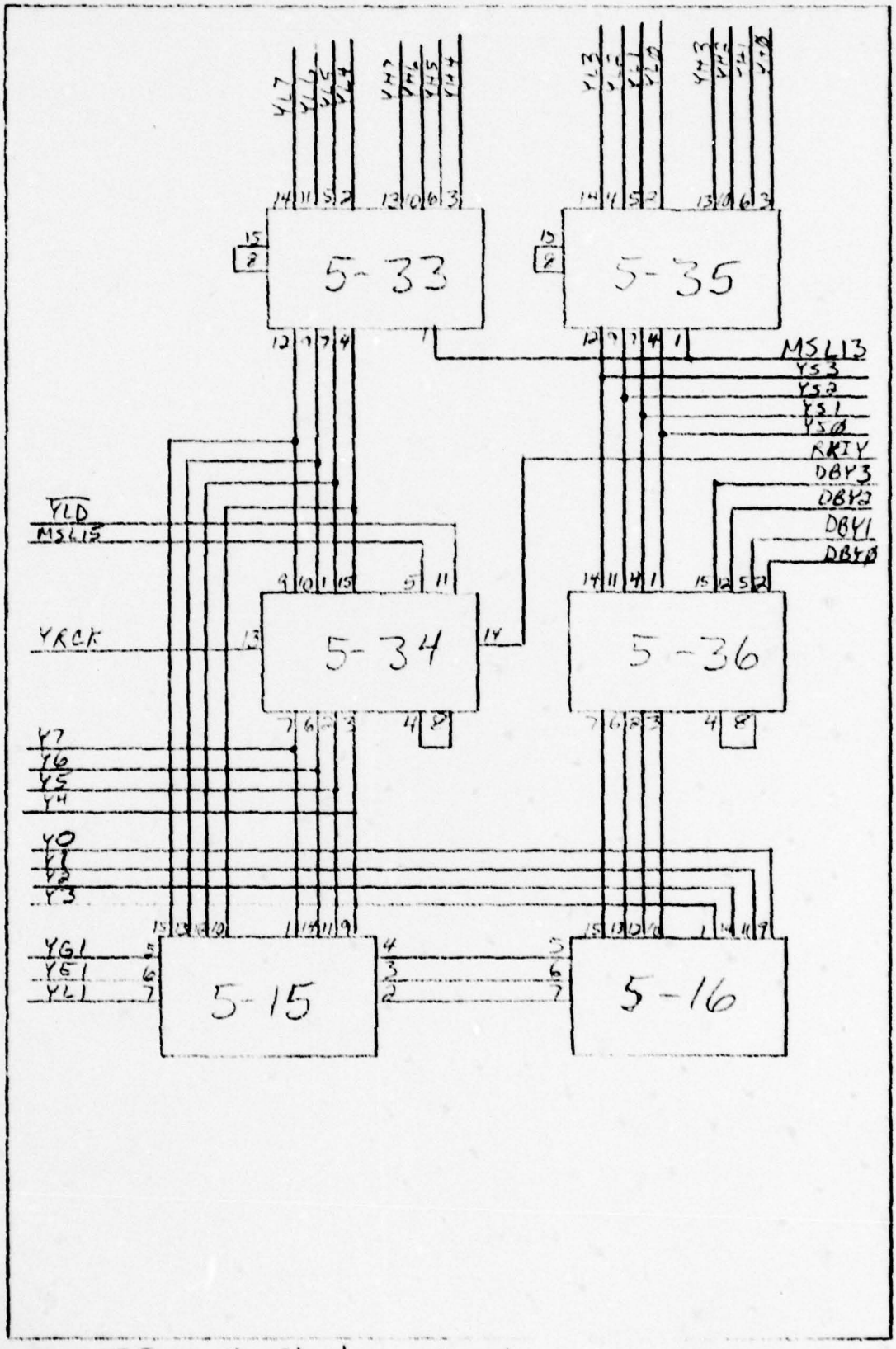


Fig. 39 Y Deflection Counter

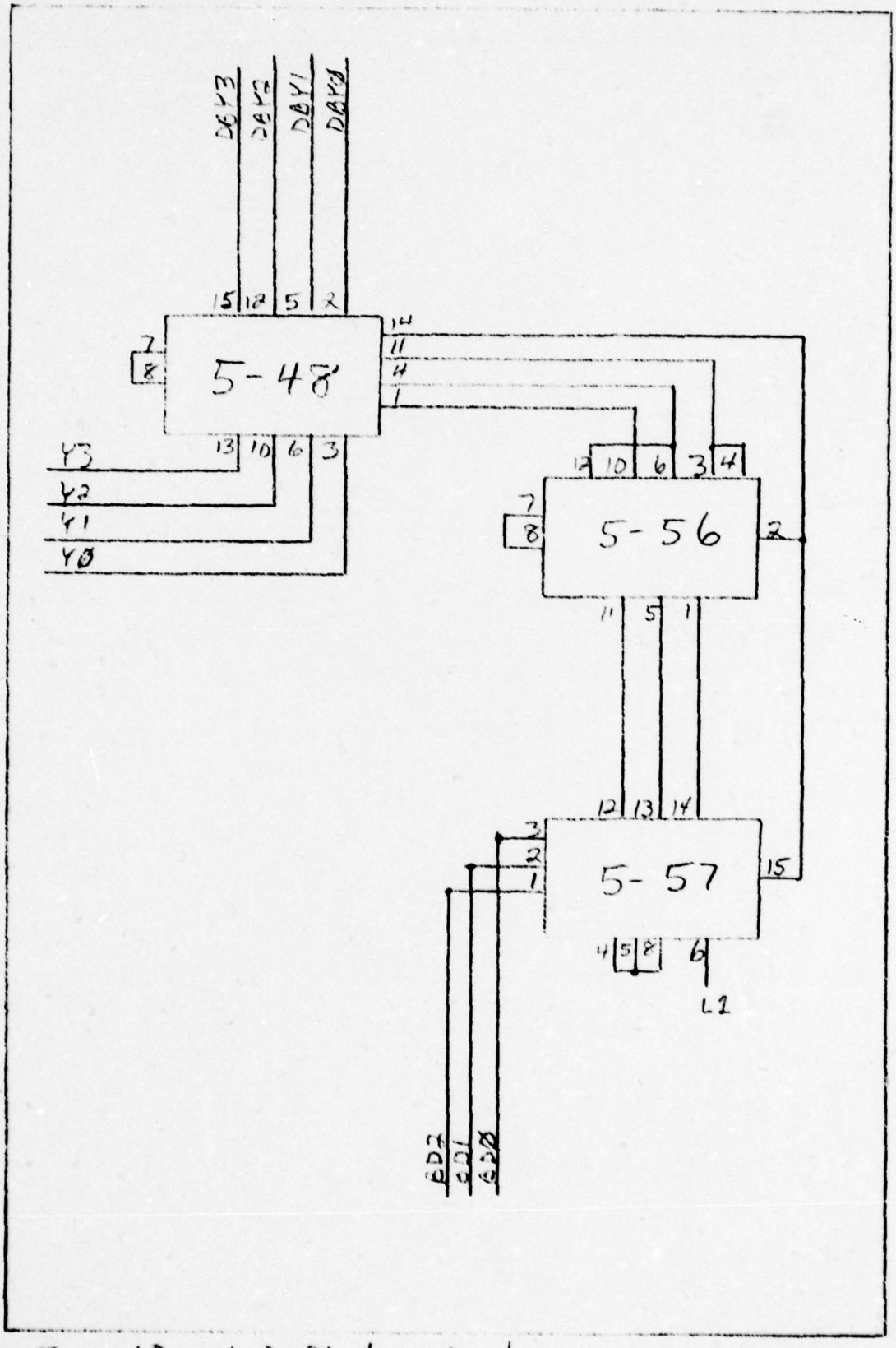


Fig 40 Y Deflection Counter

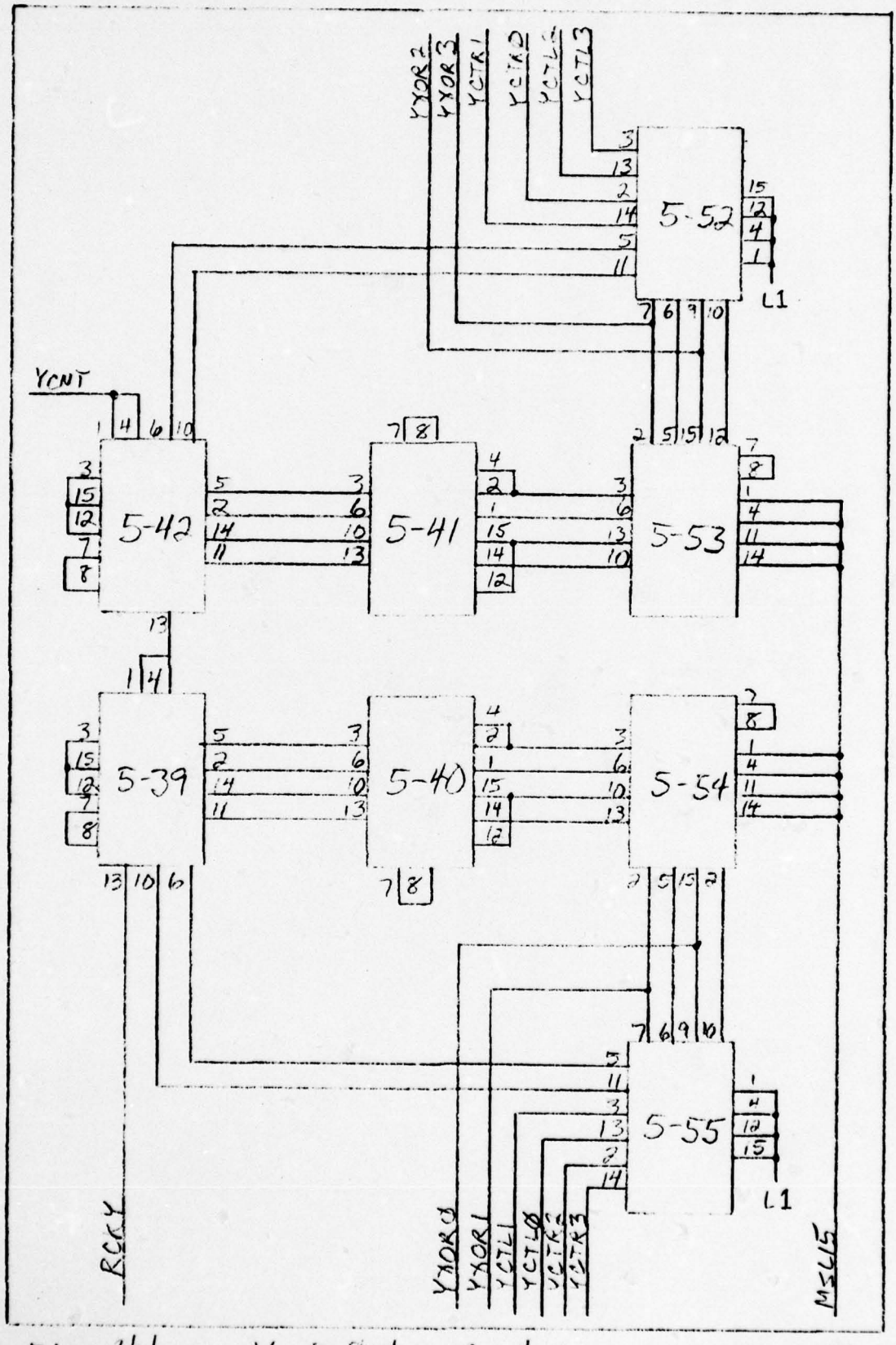


Fig. 41 Y Deflection Counter

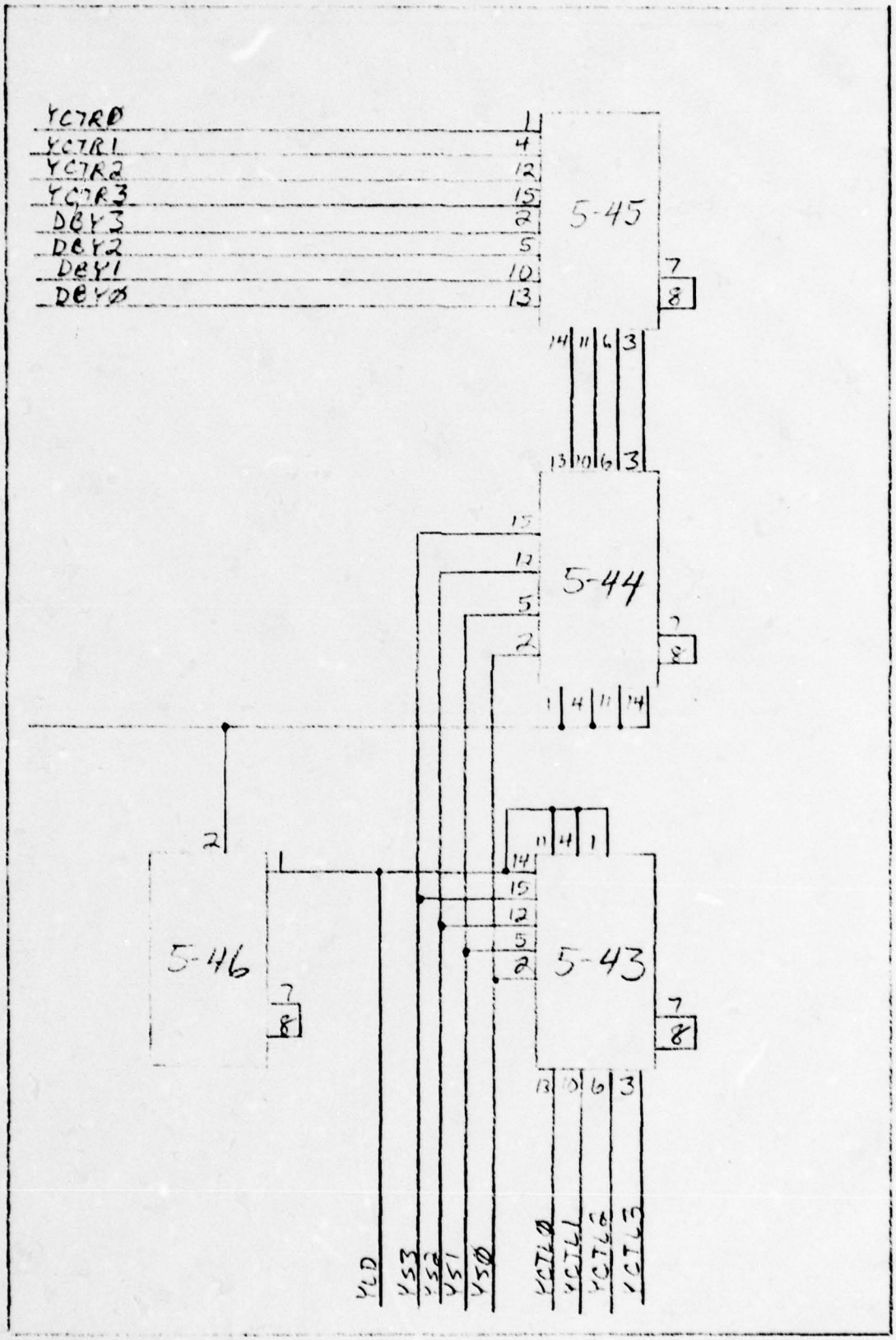
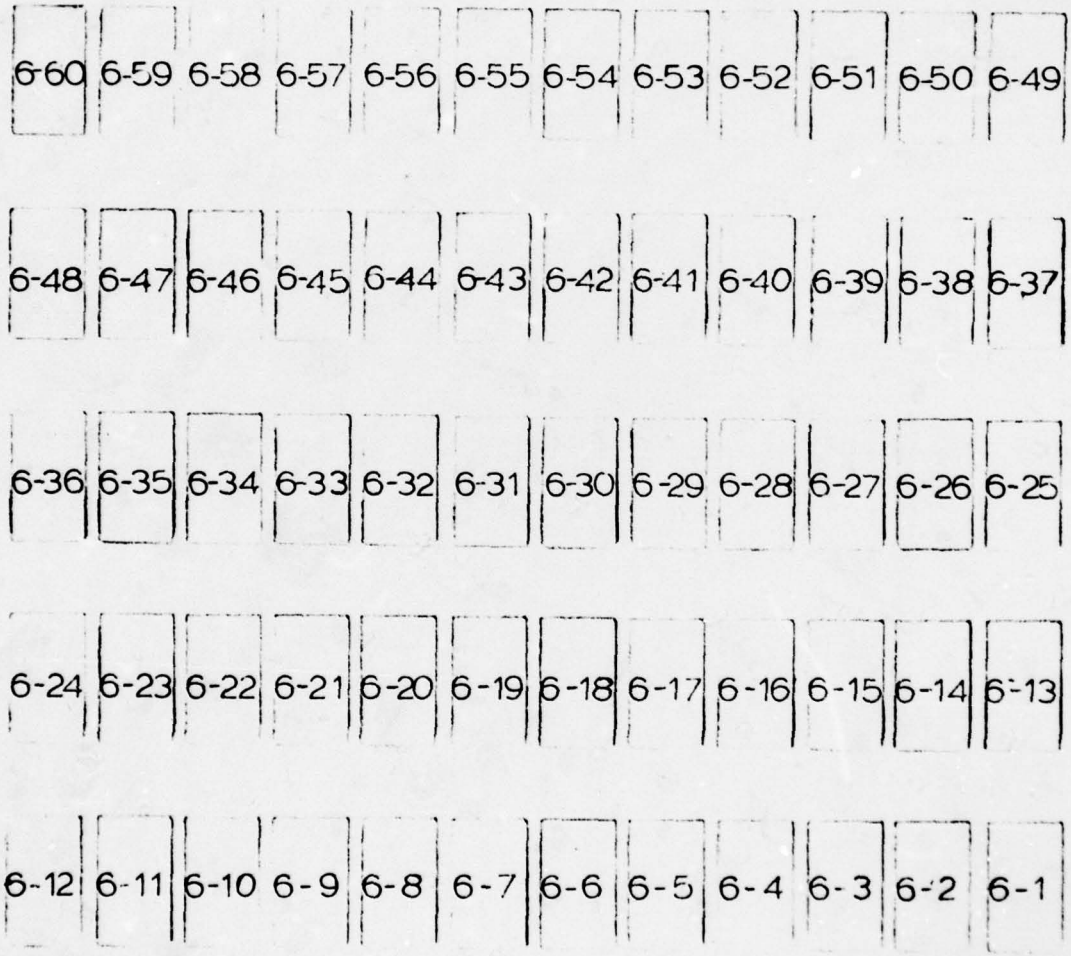


Fig. 42 Y Deflection Counter


6-60 6-59 6-58 6-57 6-56 6-55 6-54 6-53 6-52 6-51 6-50 6-49
6-48 6-47 6-46 6-45 6-44 6-43 6-42 6-41 6-40 6-39 6-38 6-37
6-36 6-35 6-34 6-33 6-32 6-31 6-30 6-29 6-28 6-27 6-26 6-25
6-24 6-23 6-22 6-21 6-20 6-19 6-18 6-17 6-16 6-15 6-14 6-13
6-12 6-11 6-10 6-9 6-8 6-7 6-6 6-5 6-4 6-3 6-2 6-1

Board 6 Integrated Circuits

Position	Type	Position	Type
6-1	SN74LS367	6-25	SN74175
6-2	SN74LS367	6-26	SN74175
6-3	SN74193	6-27	SN74175
6-4	SN74193	6-28	SN74175
6-5	SN7408	6-29	SN74175
6-6	SN7404	6-30	SN74175
6-7	SN7432	6-31	SN74175
6-8	SN7408	6-32	SN74175
6-9	SN74193	6-33	SN74175
6-10	SN74193	6-34	SN74175
6-11	Unused	6-35	SN74175
6-12	Unused	6-36	SN74175
6-13	SN74175	6-37	SN74193
6-14	SN74175	6-38	SN74193
6-15	SN74175	6-39	SN74193
6-16	SN74175	6-40	SN74193
6-17	SN74175	6-41	SN74193
6-18	SN74175	6-42	SN74193
6-19	SN74175	6-43	SN74193
6-20	SN74175	6-44	SN74193
6-21	SN74175	6-45	SN74193
6-22	SN74175	6-46	SN74193
6-23	SN74175	6-47	SN74193
6-24	SN74175	6-48	SN74193

Board 6 Integrated Circuits

Position	Type	Position	Type
6-49	SN7485	6-55	SN7485
6-50	SN7485	6-56	SN7485
6-51	SN7485	6-57	SN7485
6-52	SN7485	6-58	SN7485
6-53	SN7485	6-59	SN7485
6-54	SN7485	6-60	SN7485

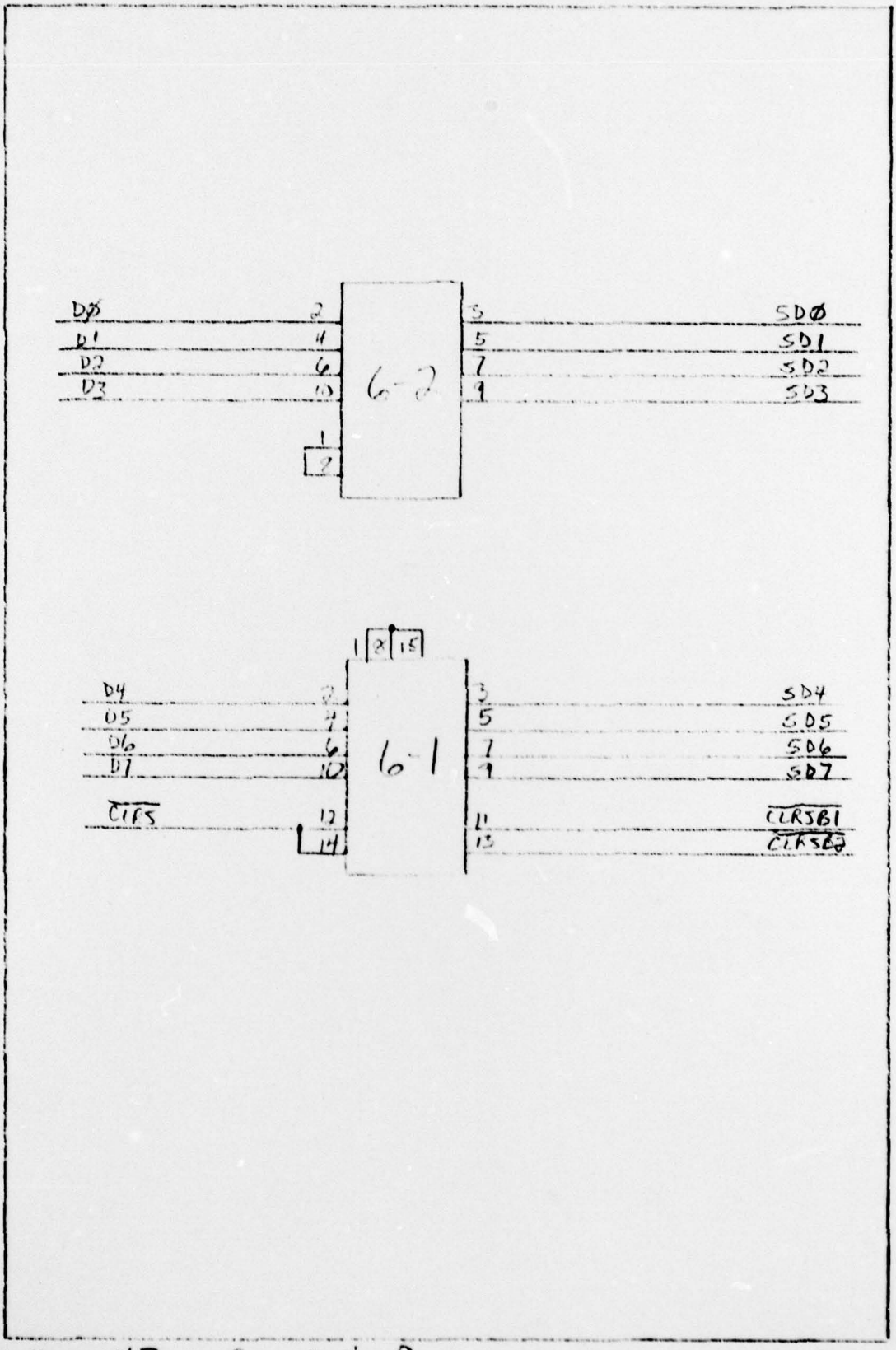


Fig 43 Bus Interface

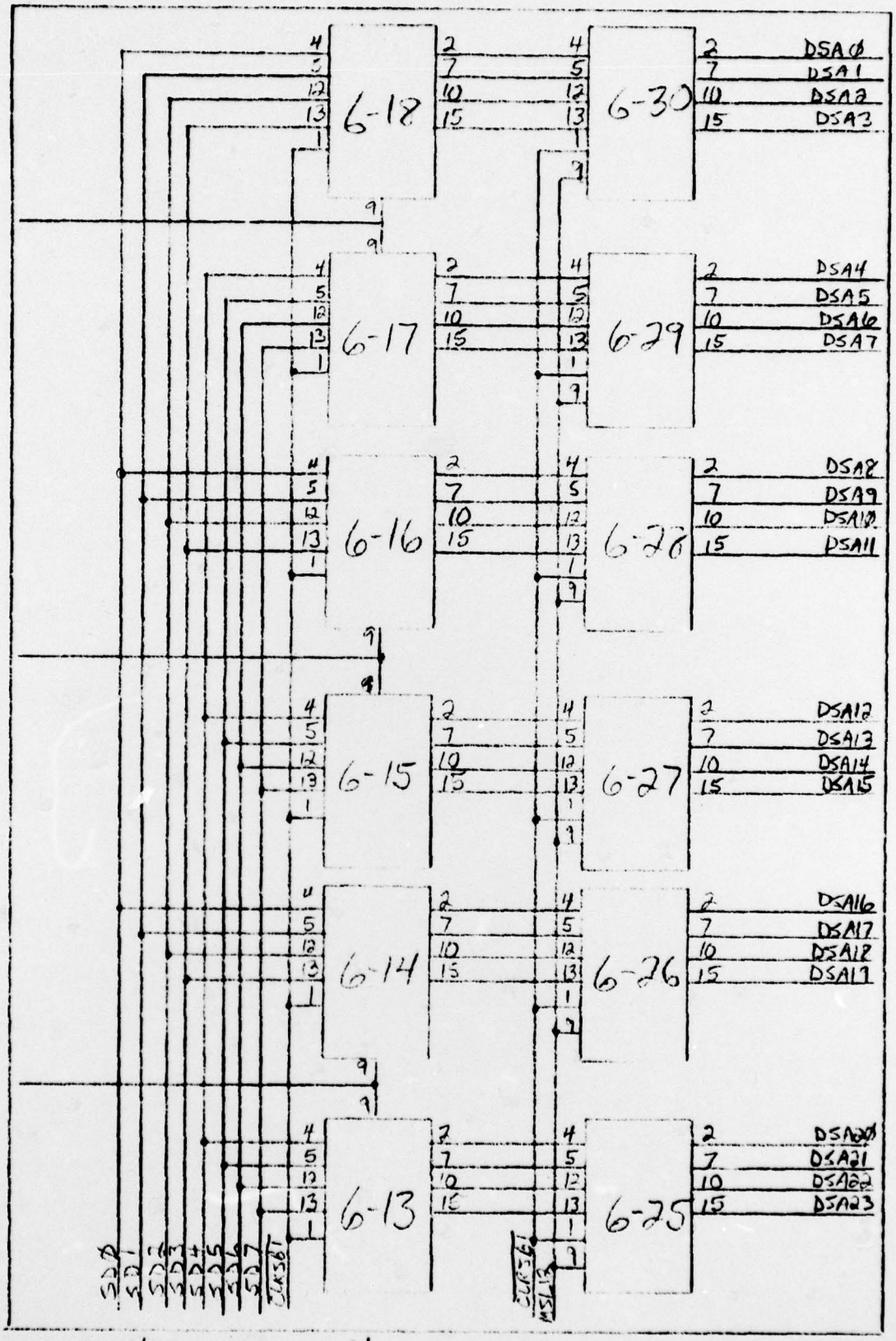


Fig 44 x stage Address Register

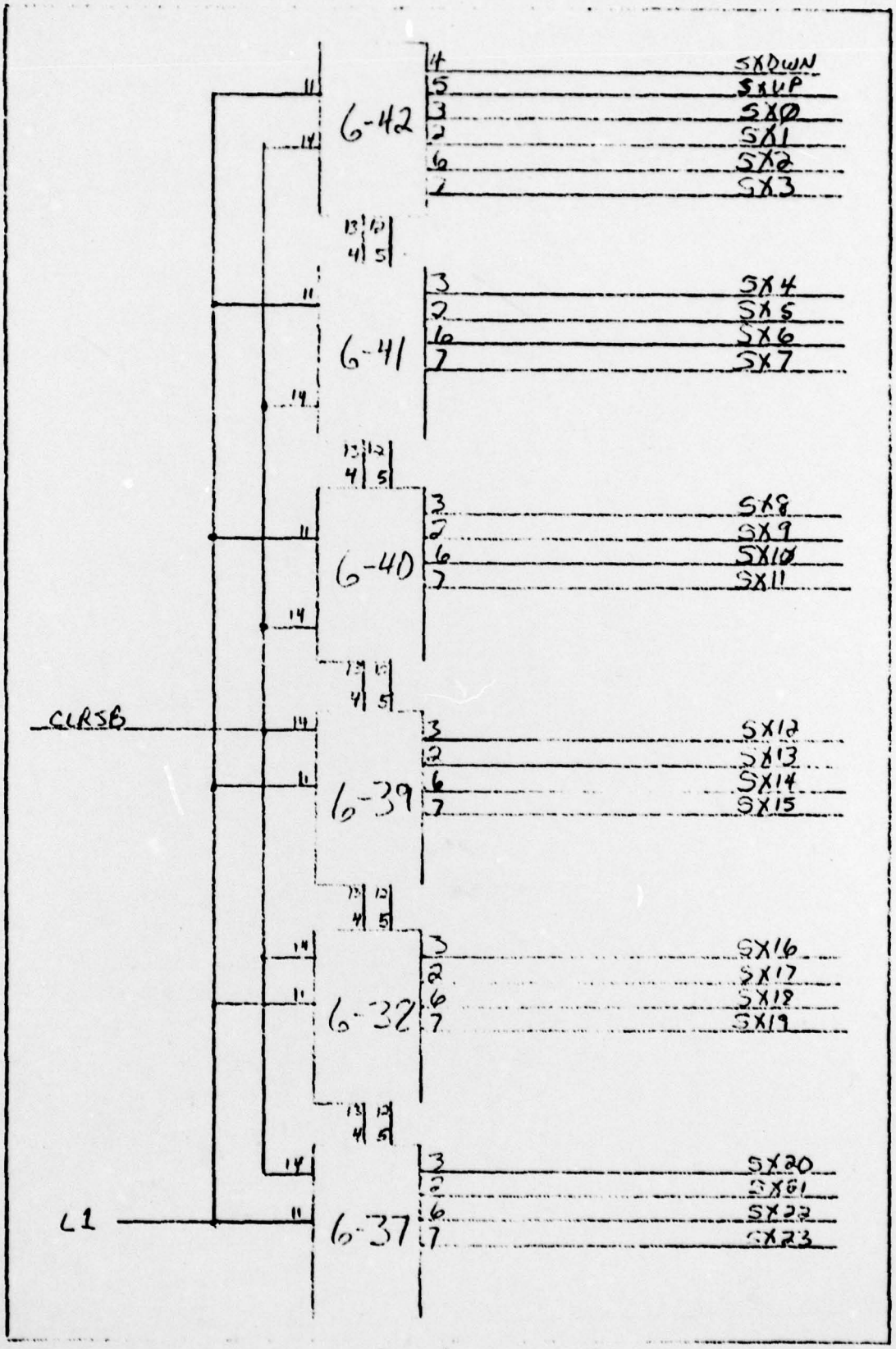


Fig. 45 X Stage Position Counter

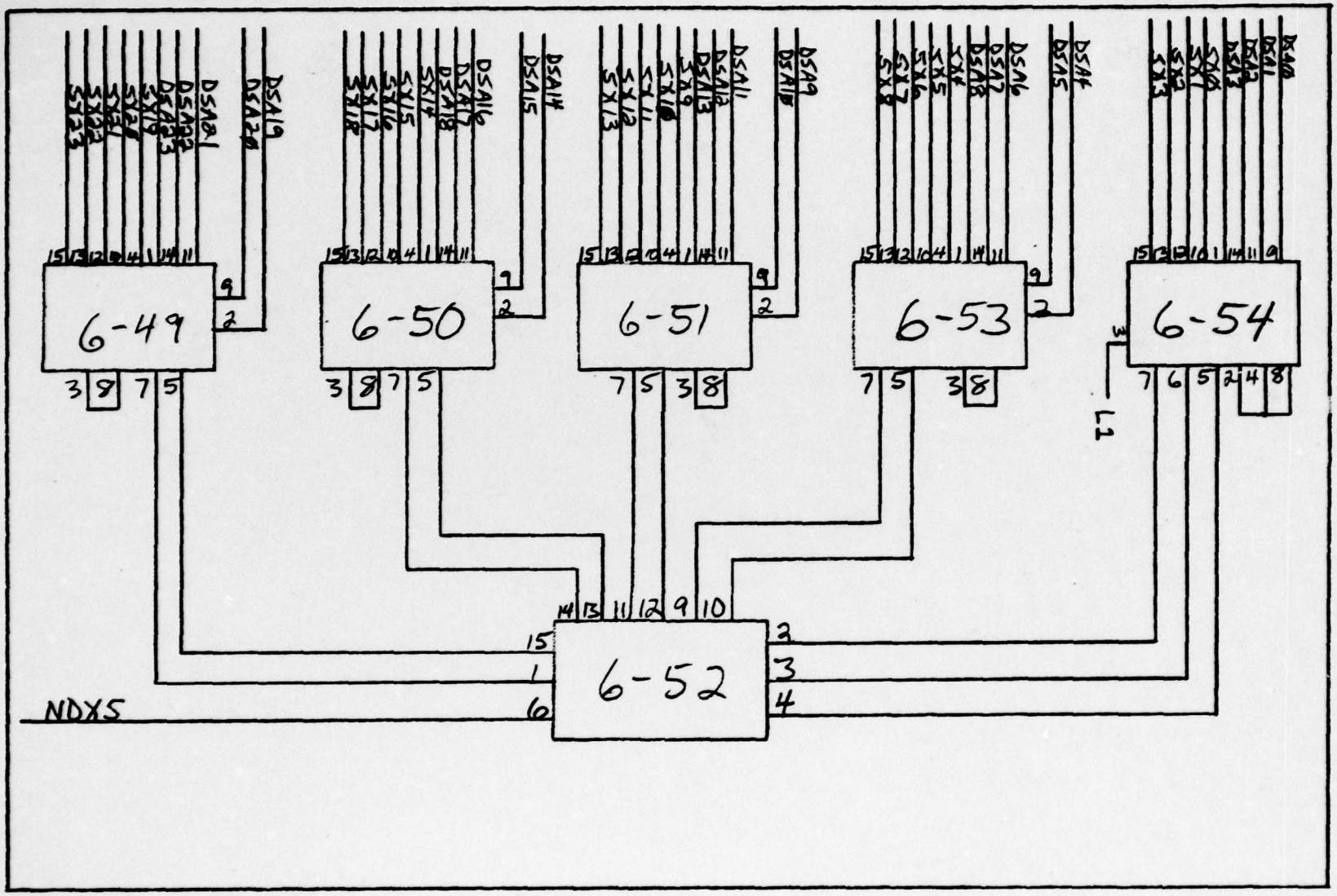


Fig 46 X Stage Comparator

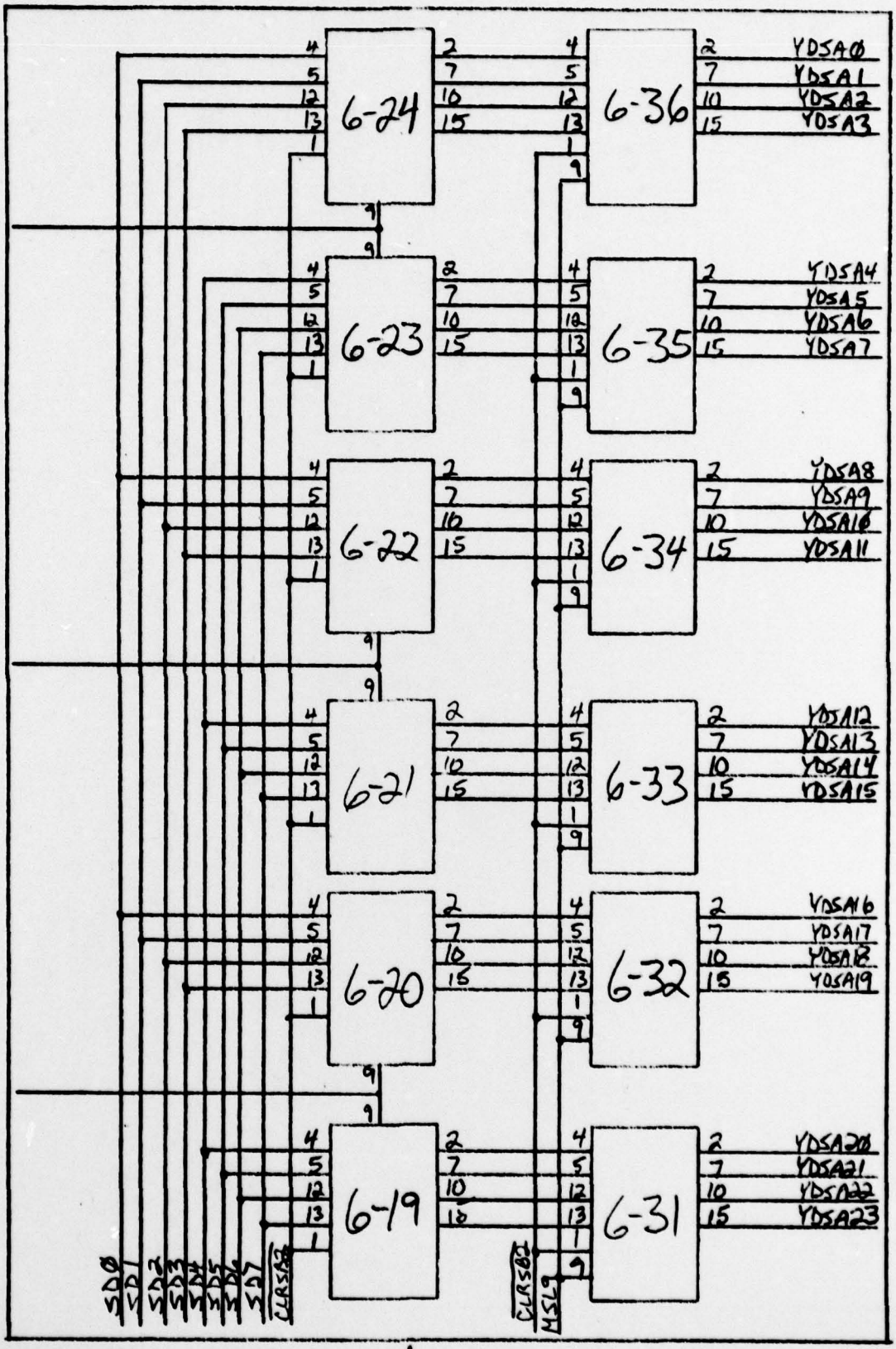


Fig 47 Y Stage Address Register

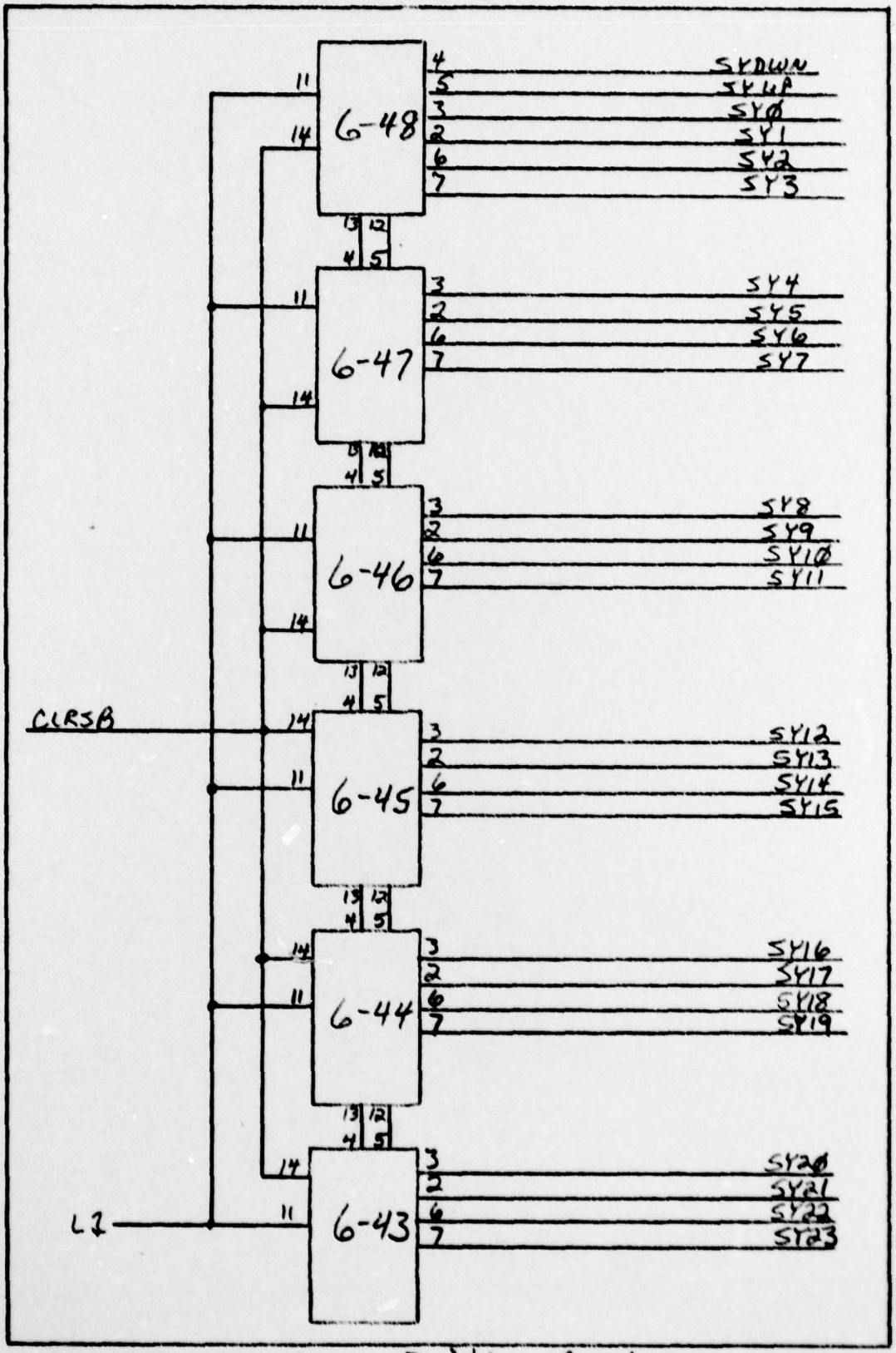


Fig 48 Y Stage Position Counter

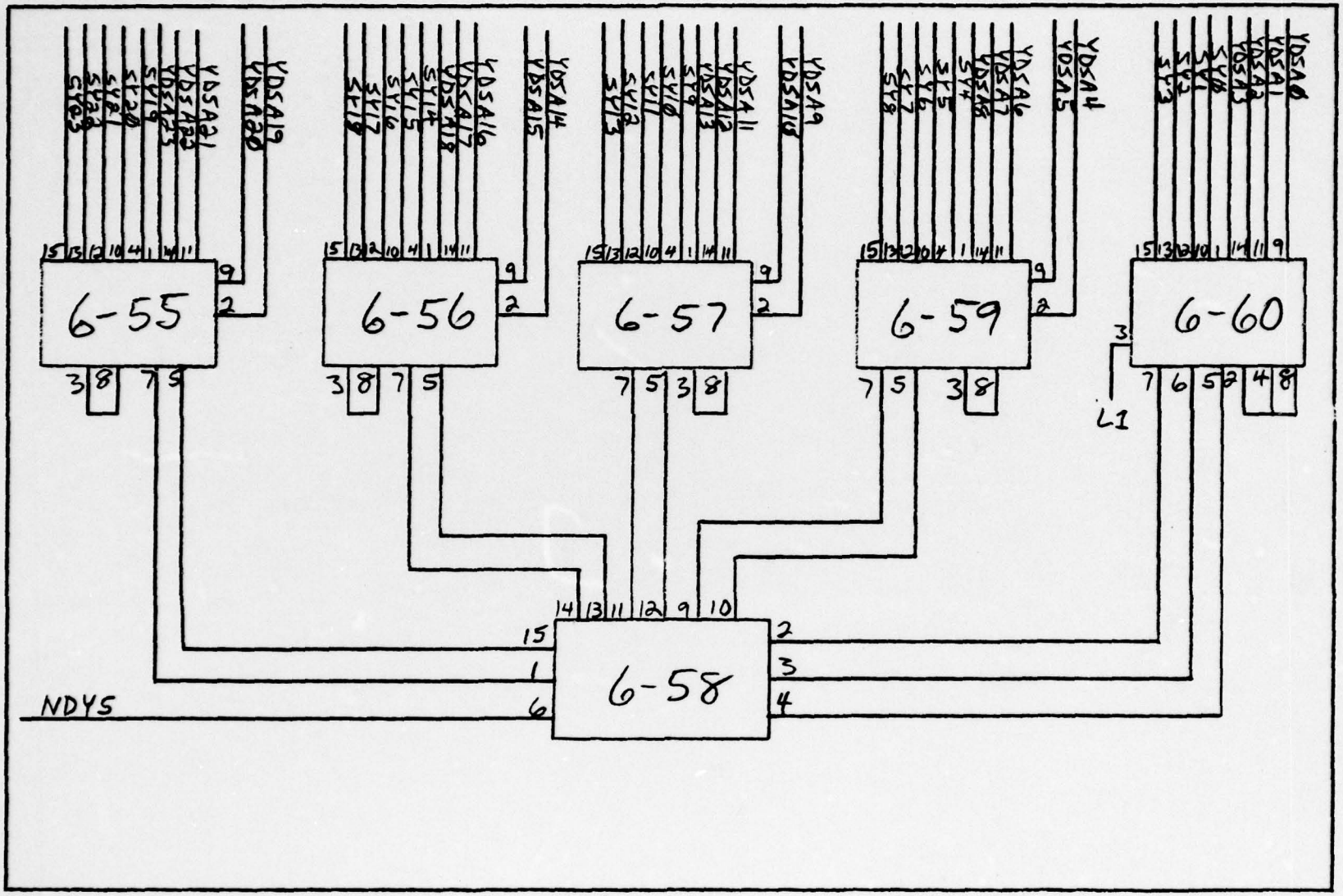


Fig 49 Y Stage Comparator

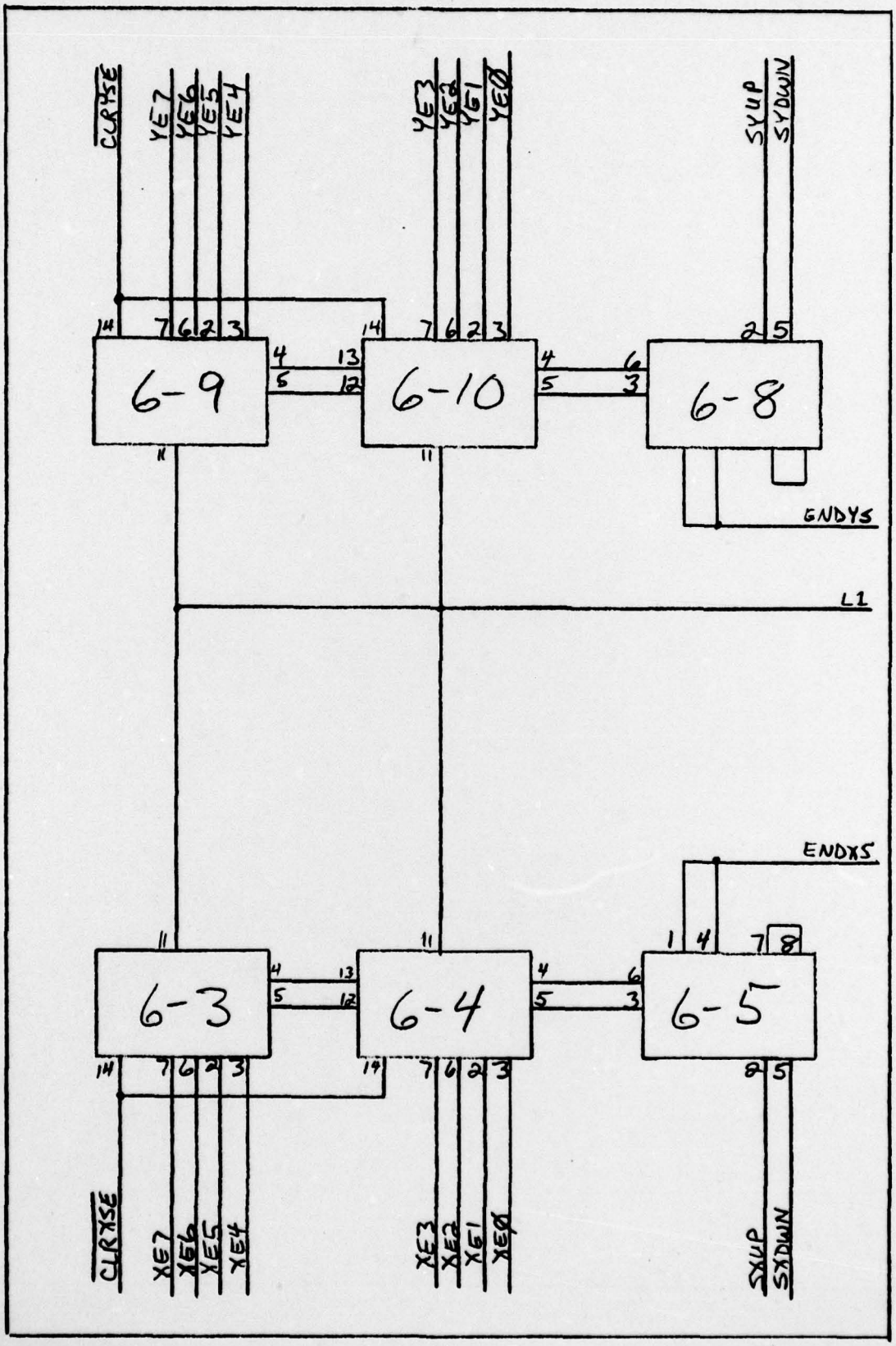


Fig 50 X and Y Stage Position Error Counters

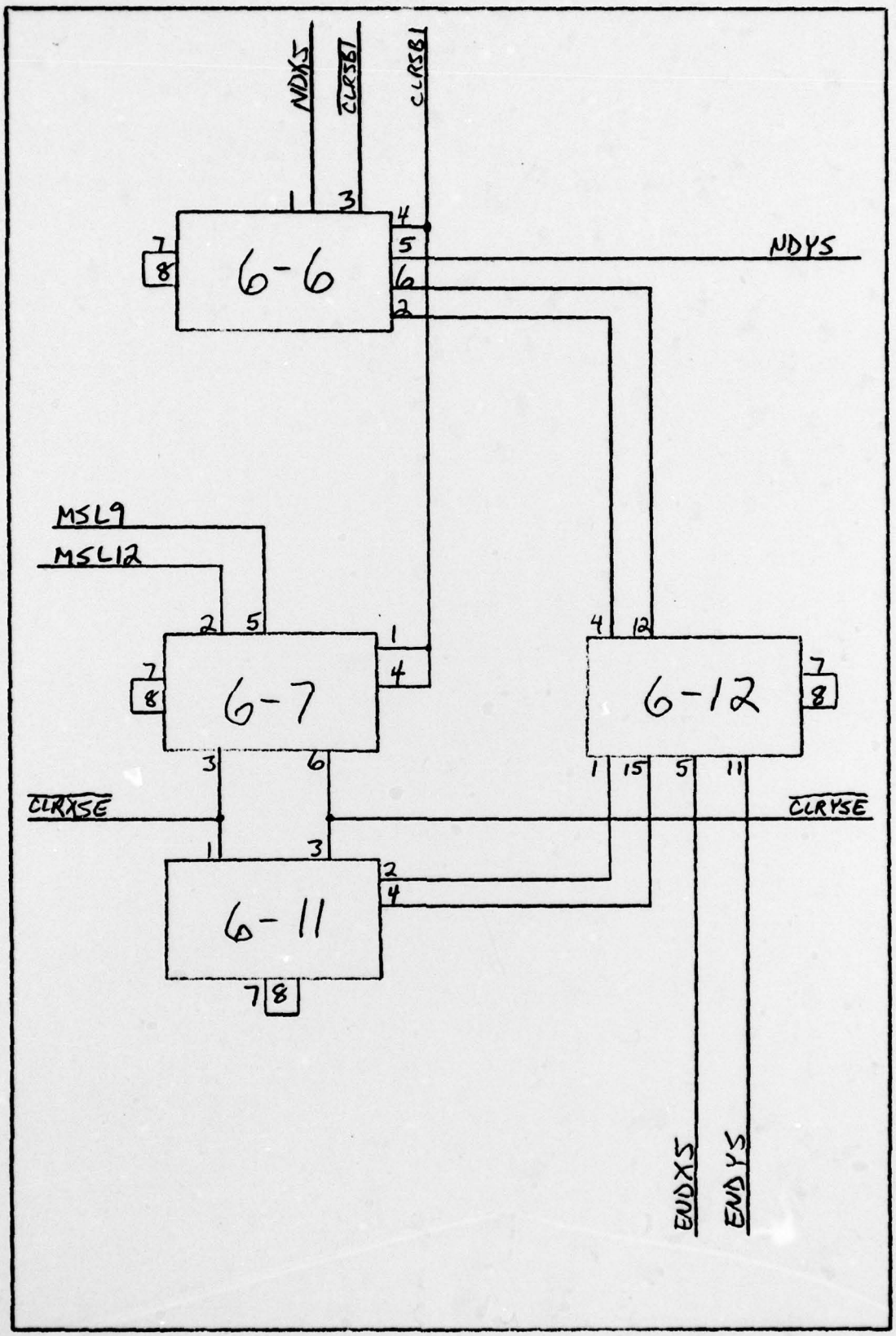


Fig 51 Error Counter Clear Circuit

Vita

Michael L. Weidner was born on 7 June 1952 in Dayton, Ohio. He graduated from high school in O'Fallon, Illinois in 1970. The same year he entered the United States Military Academy, West Point. He graduated in 1974 with a Bachelor of Science degree in Applied Science and received his commission in the Air Force. He was assigned as a computer systems analyst to Air Force Data Services Center, The Pentagon. He entered the Graduate Digital Engineering program at the Air Force Institute of Technology Resident School of Engineering in 1978.

Permanent Address:

RFD# 2 Box 280

Plymouth, N. H. 03264

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GE/EE/79-41	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Digital Controller for Electron Beam Lithography System		5. TYPE OF REPORT & PERIOD COVERED MS Thesis 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Michael L. Weidner Capt USAF		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT/EN) Wright Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE Dec 1979 13. NUMBER OF PAGES 241
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Rome Air Development Center (ET) Hanscom AFB, Mass		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for public release; IAW AFR 190-17

JOSEPH P. HIPPS, Maj, USAF
Director of Public Affairs

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Electron Beam Lithography

Pattern Generation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A digital controller for an electron beam lithography system was designed using microprocessor technology. Due to the high speed requirements of the system the hardware controller was implemented using a microprogram sequencer. The controller was designed as a special purpose computer. The word length of the controller computer is 40 bits. The first twenty-three bits are control bits. The other seventeen bits are used to determine which word to execute next. A Z-80A microcomputer was used in the design as an input/output device. The Z-80A outputs

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

the pattern data. The controller uses that data and sequences the electron beam to draw the requested pattern. The controller is double buffered for increased throughput. Since both the controller and the Z-80A are programmable, the capabilities of the system can be tailored to the needs of the user.