A 2900 MICROPROCESSOR DESIGN FOR THE GRAPHICS REAL TIME ANIMATION DISPLAY SYSTEM

by

© Parviz Shahriari, B.Eng.,
(UNI. of ARYAMEHR, 1972, IRAN)

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Engineering.

Department of Electrical Engineering,
McGill University,
Montreal, CANADA.

November, 1982.

To my mother

RESUME

Ce memoire decrit un module micro-ordinateur concu pour traiter efficacement les programmes de graphisme.

Ce module de 20 bits, concu en se basant sur la famille des MC2900. Ce micro-ordinateur est completement micro-programable afin de permettre a certaines traitements princitifs de graphiques d'etre implent's en microcodes.

Une description detaillee des circuits et des programmes utilises est suivie par une explication de la technique d'epuration service. Ensuite, l'evaluation de la performains du systeme est decrite a la fin du memoire.

ABSTRACT

This thesis describes a Micro-computer module that has been designed to provide efficient processing of graphics programs.

The design of this module is based on the MC2900 bipolar bit slice microprocessor family and has a word length of 20 bits. The Micro-Computer is completely micro-programmable to allow certain graphics premitives to be implemented in microcode.

The architecture, hardware implementation and software aspects of the microcomputer are described in detail in the various chapters. The debugging technique and performance evaluation of the system are also presented.

ACKNOWLEDGEMENT

I would like to express my sincer gratitude to all those who contributed to the presentation of this thesis.

My supervisor Dr. A.S. Malowany, with his deep knowledge and experience in digital hardware. was of immense help in the accomplishment of this thesis project. His guidance. dedication and patience made my work very enjoyable and a worthwhile experience. I deeply thank him for that.

Tanks are also due to my colleagues. in particular to R. Pancholy and G. Caravannis. for helpful suggestions during numerous discussions I had with them.

I also wish to deeply thank my mother for her encoura gement and the moral support.

I also greatly appreciate the understanding and the moral support given to me, during my work by my wife. Homa.

TABLE OF CONTENT

Page

CHAPTER I 1

INTRODUCTION 1

CHAPTER II 3

HISTORY AND FUTURE TREND 3
INTRODUCTION 3
COMPUTER GRAPHICS 4
INTERACTIVE GRAPHICS 5
CRT Displays 5
ALTERNATIVE DISPLAY SYSTEMS 6
ANIMATION 7
PARALLEL PROCESSORS 8
TWO AND THREE DIMENTIONAL PICTURE 9
TRUE THREE DIMENTIONAL PICTURE 9
FUTURE TREND OF COMPUTER GRAPHICS 10

CHAPTER III 12

THE SYSTEM ARCHITECTURE OF GRADS 12
INTRODUCTION 12
GRADS FEATURES 12
GRADS STRUCTURE AND OPERATION 13
HOST COMPUTER 15
DEC VAX 11/780 15
CROMEMCO Z80 15
GRADS SUBMODULES 16
HOST COMPUTER INTERFACE (HCI) 16
TRANSFERRING DATA 18
INTERRUPT ROUTING 18
CONTROL AND STATUS WORD TRANSFERS 18
MICROCOMPUTER MODULES 18
THE GRAPHICS CONTROLER 20
VIDEO MEMORY PLANES 22
THE T.V. SEQUENCER 22

CHAPTER IV 24

OVERVIEW OF THE 2900 MICRO-COMPUTER 24
INTRODUCTION 24
THE 2900 MICROCOMPUTER MAIN COMPONENTS 24
MC2901 - FOUR BIT TTL MICROPROCESSOR SLICE . . 26
MC 2909 MICROPROGRAM SEQUENCER 29
MC 2902 LOOK AHEAD CARRY GENERATOR 29

CHAPTER V 31

MICROCODE MEMORY 31
INTRODUCTION 31
THE CONTROL CIRCUIT 34
RECEIVER / DRIVERS 34
ADDRESS BUFFERS 37
RANDOM ACCESS MEMORY 39
MODES OF OPERATION 39

CHAPTER VI 43

LOCAL MEMORY 43
INTRODUCTION : 43
LOCAL MEMORY DESIGN 45
THE CONTROL CIRCUIT 47
DATA RECEIVERS/ DRIVERS 47
ADDRESS BUFFERS 48
SYNCHRONIZER 54

CHAPTER VII 58

THE M2900 CPU 58
INTRODUCTION 58
CENTRAL PROCESSOR 59
SEQUENCER AND BRANCH CONTROL 64
SEQUENCER 64
CONDITIONAL BRANCH CIRCUIT 68
TRACER 70
INTERNAL REGISTER, EXTERNAL REGISTER AND MULTIPLIER
GENERAL PURPOSE REGISTERS 75
MULTIPLIER 78
REGISTER CONTROLS 83
INTERNAL FUNCTION CONTROLS 83

EXTERNAL FUNCTION CONTROLS 86
STATUS 91
LOCAL MEMORY ARBITRATOR 93
PIPELINE 95
MICROWORD 96
CYCLE TIME OF CPU MODULE 101

CHAPTER VIII 104

HARDWARE REALIZATION, TEST AND DEBUGING 104
INTRODUCTION 104
HARDWARE REALIZATION 104
TEST TOOLS 105
TEST SET UP AND PROCEDURE 105
MICROCODE MEMORY TEST SETUP AND PROCEDURE 106
THE LOCAL MEMORY TEST SETUP AND PROCEDURE 108
CPU TEST SETUP AND PROCEDURE 108
TEST OF THE 2900 MICROCOMPUTER 114

CHAPTER IX 118

PERFORMANCE EVALUATION OF THE 2900 MICROCOMPUTER MODULE

DISCUSSION 121

CONCLUSION 123

REFFERENCES 126

FIGURES AND TABLES

	page
Figure 3-1 The GRADS organization	14
Figure 3-2 GRADS submodule	17
Figure 3-3 Point mode data block	21
Figure 4-1 Architecture of the 2900 Microcomputer	25
Figure 4-2 Architecture of the MC 2901	28
Figure 4-3 Architecture of MC 2909	30
Figure 5-1 Microcode Memory configuration	33
Figure 5-2 Control circuit of the Microcode Memory	35
Figure 5-3 Receiver/Driver of the Microcode Memory	36
Figure 5-4 Address buffers of the Microcode Memory	38
Figure 5-5 RAMs of the Microcode Memory	40
Figure 5-6 Microde Memory Read and Write timing diagram	
Figure 6-1 Local Memory Major Parts	44
Figure 6-2 Local Memory control circuit	49
Figure 6-3 Local Memory Data Receiver/Drivers	50
Figure 6-4 Local Memory Address Buffers	51
Figure 6-5 Local Memory Synchronizer	53
Figure 6-6 Local Memory RAMs	55
Figure 6-7 Local Memory Senchronizer timing diagram	56
Figure 7-1 2900 CPU subsections	60
Figure 7-2 Central Processor (CPU)	61
Figure 7-3NOP circuit	65

Figure 7-4 Sequencer 66
Figure 7-5 Conditional Branch circuit 69
Figure 7-6 Tracer circuit 71
Figure 7-7 Tracer timing diagram 72
Figure 7-8 Internal Reg. External Reg and Multiplier 76
Table 7-1 Control Register 80
Table 7-2 Status Register 81
Table 7-3 HCI Machine Register 82
Table 7-4 Internal function control selection 84
Figure 7-9 Internal Function Control circuit 85
Figure 7-10 External Function Control circuit 87
Table 7-5 External function control register 88
Figure 7-11 Intrrupt circuit 89
Table 7-6 Interrupt selection 90
Figure 7-12 A simple of Status Register 92
Figure 7-13 Local Memory Arbitrator 94
Table 7-7 Load constant 97
Table 7-8 Branch Format 98
Table 7-9 Status updat format 99
Table 7-10 ALU format 100
Figure 7-14 2900 Microcomputer timing diagram 102
Figure 8-1 Microcode Memory test setup 107
Figure 8-2 MICMEM flochart 109
Figure 8-3 Local Memory test setup 110
Figure 8-4 CPU test setup 112

Figure 8-5 CPUTST flowchart 115

Figure 8-6 2900 Microcomputer test setup 117

CHAPTER I

INTRODUCTION

Psychologists believe that 80% of our information is established via visual contact. It is not surprising, therefore, that over the last three decades, a lot of effort has been put in the field of computer graphics .

Currently an experiment, on color graphics animation is under way at the Electrical Engineering Department of McGill University. The main objective of this research is to construct a system capable of supporting color computer graphics animation in real time. This machine has been named GRADS (an acronym for Graphics Real time animation Display System).

The aim of this research is to design a microcomputer, based on the Motorola M2900 bit slice microprocessor chips, that will be operated in parallel with two other microprocessor on the GRADS. This microcomputer is the most powerful engine of the system as far as speed and arithmetic operations are concerned .

This report explains the components and functions of this microcomputer system .

Chapter II describes the history and the future trends in computers, displays, microcomputer and computer graphics.

Chapter III introduces the architecture of the GRADS and presents its essential block diagrams.

Chapter IV through VII contain the detailed information about the actual design of 2900 microcomputer hardware.

Chapter VIII presents the configuration, procedures for testing and software to debug the hardware .

Chapter IX presents the performance and results

CHAPTER II

HISTORY AND FUTURE TREND OF COMPUTER

INTRODUCTION

From as early as 500 (B.C) up to 1945, a variety of computational aids such as the Abacus and the slide rule were invented, which simplified but did not replace manual computation(32).

The ENIAC invented in 1945, was the first operational electronics computer which opened a new world of computation to the human beings(32). Following that, the invention of the transistor in late 1949, by Bell Telephone Laboratories accelerated the speed of research in this area.

The first TTL chips were introduced by Texas Tnstruments in 1961 and since then the development of IC's has progressed through small scale integration (SSI), medium scale integration (MSI) and large scale integration (LSI)(14).

Intel 4004 was the first commercially available microprocessor, introduced in 1971, and it proved to be the starting

point of the microprocessor revolution, the effect of which can be seen today . Systems such as the Motorola Mc2900, 68000, Zilog Z80 and the Intel 80 86 indicate the diversity of the microprocessor families of today(62).

COMPUTER GRAPHICS

In the early 50's, communication between man and the computing machine was one of the major problems. Professor Neuman pioneered in using a CRT for communicating with the machine , and later in 1951, the WHIRLWINDI project (20) at MIT became the first system which introduced the concept of computer graphics. From that time on, computer graphics systems have been intensely studied by a multitude of people for well over three decades. The goal of this ongoing research is to create an interactive machine which is flexible enough to calculate any complex image in three dimensional world and display it on the screen. An ideal graphics machine should be able to compute and display high resolution color images at a speed high enough to avoid 'flicker' and should allow for the participation of the user by allowing real time interaction.

INTERACTIVE GRAPHICS

'Sketchpad' was the first interactive graphics system which ntroducedwaby Sutheland at MIT(19). This terminal was based on a predetermined table of options which made possible it for the user to select them in order to communicate with the system.

Later on, the CRT displays, in conjuction with a key-board, became the most common interactive media between man and the machine(35).

CRT Displays

Cathode-Ray tube (CRT) has long been the only choice for graphics display because of color, brightness, reliability, and low cost. It, however, suffers from flicker, high volume, heavy weight and its cumbersome and fragile nature.

The CRT function by virtue of a focused electron beam or three focused electron beams (color CRT's) which stimulate phosphorecent coating on the screen to emit visible light(16).

Each stimulated area on the screen emits light for about 0.2 sec, so for having a continuous bright point with a constant

emissive intensity, this point must be stimulated repeatedly, higher than a certain rate, otherwise the viewer will see a blinking point. This effect is known as 'flicker' and to avoid that each set of points must be repeated at minimum rate of 25 times /sec.

The creation of a picture on the CRT can be achieved in two different ways :

1: Raster scanning

2: Random Positioning

In raster scanning, all the elements of the picture, known as pixels, are scanned periodicaly in horizontal lines and regardless of existing picture information on the CRT . A collection of such horizontal scan lines are needed to scan all the points on a CRT screen. This is known as a pair of frames. In random positioning the beam will travel randomly. Random Position presents finer images in black and white CRT's, as drawing straight lines or continuous curves are possible; Raster scanning is prefered for lower cost ,simplicity and for color applications (33).

ALTERNATIVE DISPLAY SYSTEMS

The idea of a flat Display, that can be hung on the wall or carried in the pocket,is an extremely attractive objective(42).

On the way toward this ultimate goal, active work has been done recently in developing flat panel displays such as :

1: Light Emitting Diods (LED) Displays(60).

2: Plasma Displays(44).

3: Liquid Crystal Displays(49).

4: Electroluminesent displays.

The invention of the blue LED (9) has made possible the use of LED's as the basic luminescent elements in an array form. Liquid Crystal and LED's are likly to be the material of choice for a new generation of flat displays positionned directly on top of memory arrays. This simplifies life and construction and it eliminates the need for the refresh cycles.

Although the above mentioned projects have improved their commercial feasibilities, the CRT remains the preferred display device currently and in the near future.

ANIMATION

In order to generate a feeling of continuous motion, the picture of each moving object should be displayed at least 25 times per second such that each frame projects its new position . If each

frame consists of an array of 512 by 512 pixels per color bit, the task of animation requires the updating of over six hundred thousand pixels every 1/30th of a second. This data flow is roughly equivalent to 10 million color words per second.

Such computational and data flow requirements generate the demand for an extremely fast and powerful computer to achieve animation . This has impelled researches to explore the possibility of using more superior processing concepts like parallel processing and pipelining for use in animation system (61).

PARALLEL PROCESSORS

Although, the very fast microcomputer is available today which could implement each instruction in less than 100 nsec, but the throughput of one microprocessor will not be sufficient for an interactive real time color graphics display. The best way of coming over this problem is using multiple microprocessor which is working in parallel, and all together will participate to performe each frame (4,8).

PIPELINE ARCHITECTURE

One of the several ways to have a higher throughput in a system is pipeline architecture. In this method, like in the assembly lines, several operators are used each after the other to manipulate the instructions. So in each stage the instruction will be decomposed to another level.

TWO AND THREE-DIMENSIONAL PICTURE

The display of three Dimensional pictures is much more difficult than that of two Dimensional images. This is due to frequent requirement, showing a perspective view of a 3D object on the 2D screen as well as the elimination of hidden surfaces . Till now, much effort has been spent on simplification of the algorithmes for these types of calculations (3).

TRUE THREE DIMENSIONAL PICTURE

Recently a new electrically light switching lenz LLZt (Lead Lanthanum Zirconate titandte) has been developed (10). Now if the odd and even frames are calculated from two slightly different reference points and if the screen is viewed through a pair

of LLZt glasses, and if Voltage pulses synchronized with the display of the odd and even fields darken the left and right lenses alternatively, as the result, the viewer sees a true 3-dimensional images.

COMPUTER GRAPHICS APPLICATIONS

By increasing the power of microprocessors and decreasing the cost of powerful systems, computer graphics has already found its way into different applications. In general, their applications can be categorized in six major areas (33).

Management information

Scientific Graphics

Command and Control

Image Processing

Design

Entertainment

Real time applications

FUTURE TREND OF COMPUTER GRAPHICS

The cost of real time two and three dimensional graphics

systems is still very high (14). Until recently no significant VLSI development in the special purpose domain has been made. Currentely,however, several major research teams are attempting to develop custom VLSI graphics systems. In the near future, therefore, highly compact computer graphics and animation system are likely to be widely available.

CHAPTER III

THE SYSTEM ARCHITECTURE OF GRADS

INTRODUCTION

As mentioned in the previous chapter, currently a computer graphics system is under construction at the Electrical Engineering Department of McGill university. This system has been named GRADS (an acronym for Graphics Real time Animation Display System). Since realization of such a system requires tremendous amounts of data flow and enormous amount of computational power, concepts of parallel processing and pipelining have been used in this design. the GRADS can perform parallel processing with the aid of a bank of independent microcomputers, and concurrent data flow by means of multiple busses and DMA machines. This chapter will briefly present the GRADS features, structure, and submodules.

GRADS FEATURES

The complete operation of the GRADS is programmable. for color or black and white operation, as well as for low resolution

(256 X 256) or high resolution (512 X 512).

In order to preserve a constant background, the programmer is able to change the mode of operation from " Opaque " to "transparent ". In Opaque mode, the old frame contents will be replaced by the new one, but in transparent mode the new information of each pixel will be logicaly "OR"ed with the old one.

GRADS STRUCTURE AND OPERATION

Figure 3-1 represents the organization of the GRADS A variety of user programmes can be executed on the host computer system by processing the user interaction and resulting the display of compounding animated images. These resulting images will appear at the output of the host computer in a high and compact format as a series of Macro-instructions such as polygons or lines. This information will then be broken down to N groups and will be submited to N paralleled microcomputer modules for translation into graphic controler instructions. The GRAPHIC CONTROLER unit will read the computed data buffers from each microcomputer module and process them to generate and store pixel information into the video memories. The resulting, intensity and color information of the pixel is thus stored at the required address in the frame buffer for display.

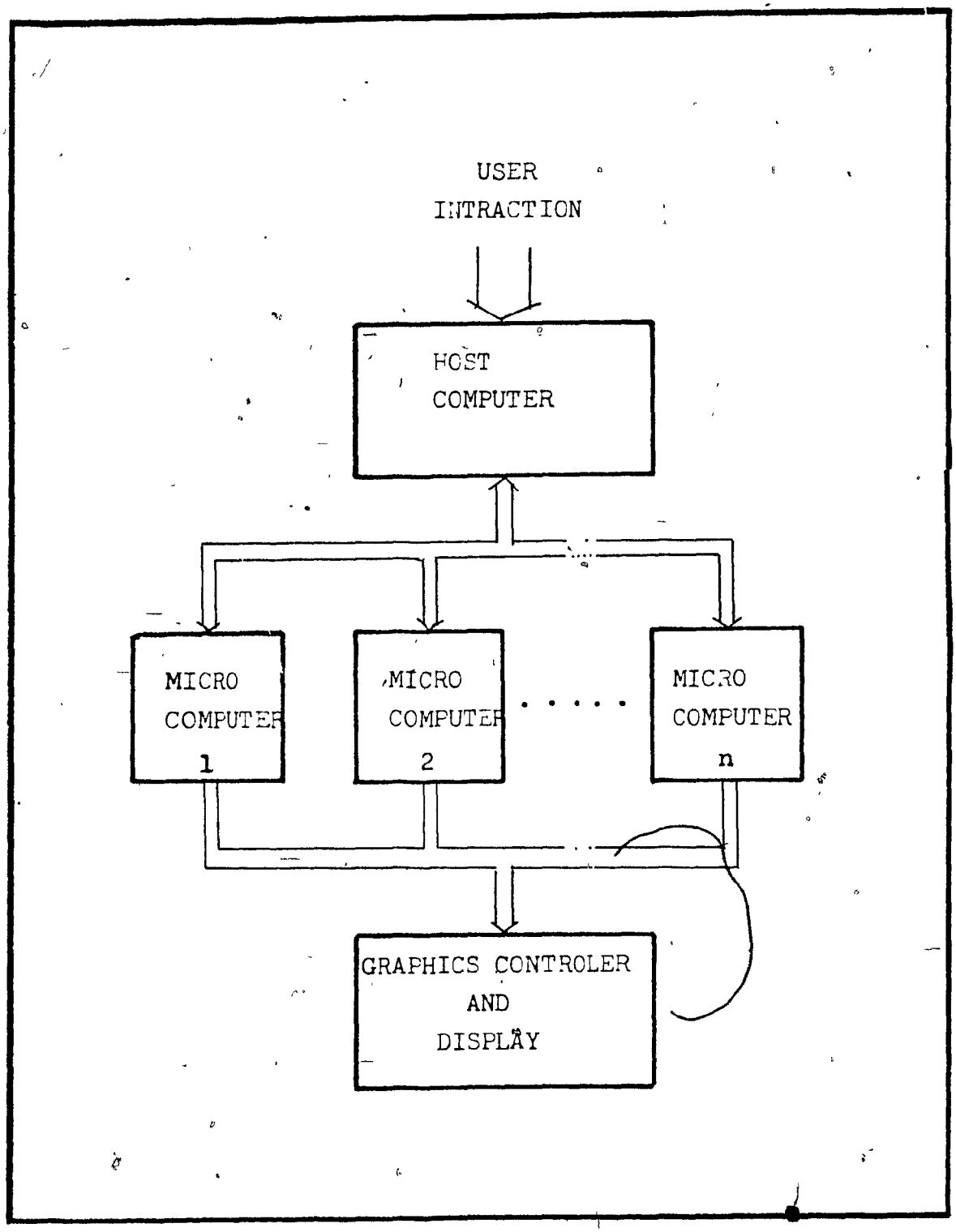


Figure 3-1 The GRADS organization

HOST COMPUTER

It has been decided to use a commercial computer to control the GRADS. This computer can have access to the GRADS through the Host Computer Interface. Because of its suitability for development of software applications and availability in the computer graphics laboratory, DEC VAX 11/780 and Cromemco Z80 have been chosen as the Host Computers. The first will be used to execute the application programmes while the second will be used for debugging and maintenance.

DEC VAX 11/780

The Dec VAX 11/780 is powerful enough to generate the macroinstructions at the required rate. Its architecture is based on a 32 bit word length, efficient memory management and v,irtual memory operations. It is also a priority-scheduled, event- driven system. The assigned priority and activities of the processes in the system determine the level of service they need. It is supported with 2 MK bytes of memory two 300 MB disc drive and one magnetic tape as secondary storages (18).

In the architecture of this machine two busses can be realized, the MASSBUS and the UNIBUS. The GRADS has access to it through the UNIBUS.

CROMEMCO Z80

This is a microcomputer consisting of a Z80 CPU based on the S100 BUS architecture and contalning 64K bytes of memory, two floppy disc drives and one CRT terminal. In addition, two Cromemco PIO8 boards each containing 8 parallel input output ports, are available (15).

As mentioned before this is not a powerful machine and can not generate complex animation at the required rate. It is only used to generate some static or very slow motion pictures for maintenance purposes.

GRADS SUBMODULES

Figure 3-2 presents a further break down of GRADS into its submodules which will be explained in the following sections.

HOST COMPUTER INTERFACE (HCI)

The HCI module links to three internal busses (CPU, MICRO GRAPHICS CONTROLER) and both external busses(UNIBUS, S100)(46). The services provided by the HCI include :

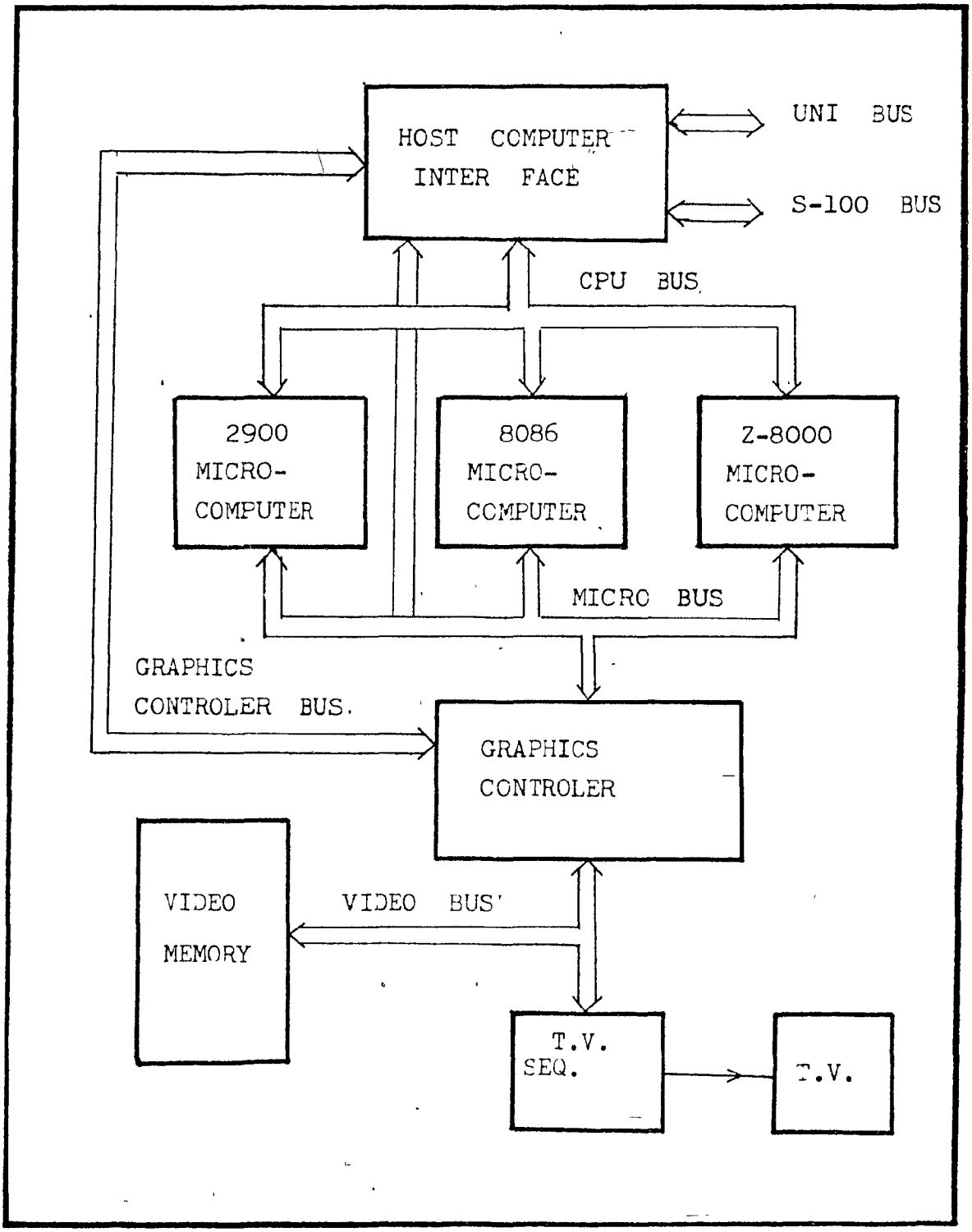


Figure 3-2 GRADS submodule

a: transferring data (by DMA).

b: Interrupt routing.

c: Control and status word transfers.

TRANSFERRING DATA

The HCI is equiped with a DMA (Direct Memory Access) machine and is able to provide a path from the Host Computers to the internal modules and vice versa. Data can also be transfered by the DMA machine from any microcomputer module to another. For the initialization, it allows downloading of microcode and programmes to the microcomputers and to Graphics Controler.

INTERRUPT ROUTING

The HCI provides interrupt servicing from either of the Host Computers to the GRADS modules or vice versa. Concurrent requests are taken care of through arbitration.

CONTROL AND STATUS WORD TRANSFERS

The HCI is also capable of providing a path from the Host Computer to the GRADS modules for reading/writing from or to their status registers. The CONTROL registers are used to regulate the operation of the microprocessor modules from the host computer while the STATUS registers are used for synchronization and scheduling.

MICROCOMPUTER MODULES

The array of microcomputer modules forms the second processing level in the GRADS system. These microcomputers are modular in nature and can be dynamicaly programmed by the Host computer to performe various processing functions. A total of four such units is envisaged for the full operation of the GRADS . Currently four microcomputer modules have been designed based on the Motorola Mc2900, 68000, Intel 8086 , Zilog Z8000 microprocessors. The same microinstructions can be processed by these different microcomputers for evaluation(5,55,61,27).

The first microcomputer module was designed around the Mc2900 bit-slice microprocessor and constitutes the most powerful microcomputer of the (GRADS). The design of this microcomputer is the topic of this thesis and will be discussed in greater detail in the subsecquent sections.

The second, which is based on the Intel 8086, is a 16 bit machine and is capable of providing a direct address of up to 64K, but, in segmentes it may address up to 1 M Bytes of memory. This unit with its powerful instruction set, 16 bit arithmetic capability, and 8 MHZ clock, can support low resolution (256*256) displays on the GRADS.

The third micro-computer is designed with zilog Z8000. It is also a 16 bit machine and provides a direct address up to 8 M Bytes. In conjunction with a memory management unit it can address up to 48M Bytes of memory. A builtin refresh mechanisme simplifies its use with memories.

THE GRAPHICS CONTROLER

The GRAPHICS CONTROLER performes the last level of processing in the GRADS. It is a special purpose DMA machine that provides for downloading of processed data from the microcomputers and for translating this processed data into actual pixel information. It then stores this pixel information in the frame buffers in the video memory planes of the frame buffer(12).

The GRAPHICS CONTROLER also arbitrates the use of the Video Bus by the TV sequencer.

Figure 3-3 shows an example of the information processed by the GRAPHICS CONTROLER. The first word of each block, which is command word, is saved in the command register to control the system during the block DMA. Four such command modes are available namely : Solid Area Mode, Shading Area Mode, Point Mode and Read Back Mode. The first is useful for painting any area on

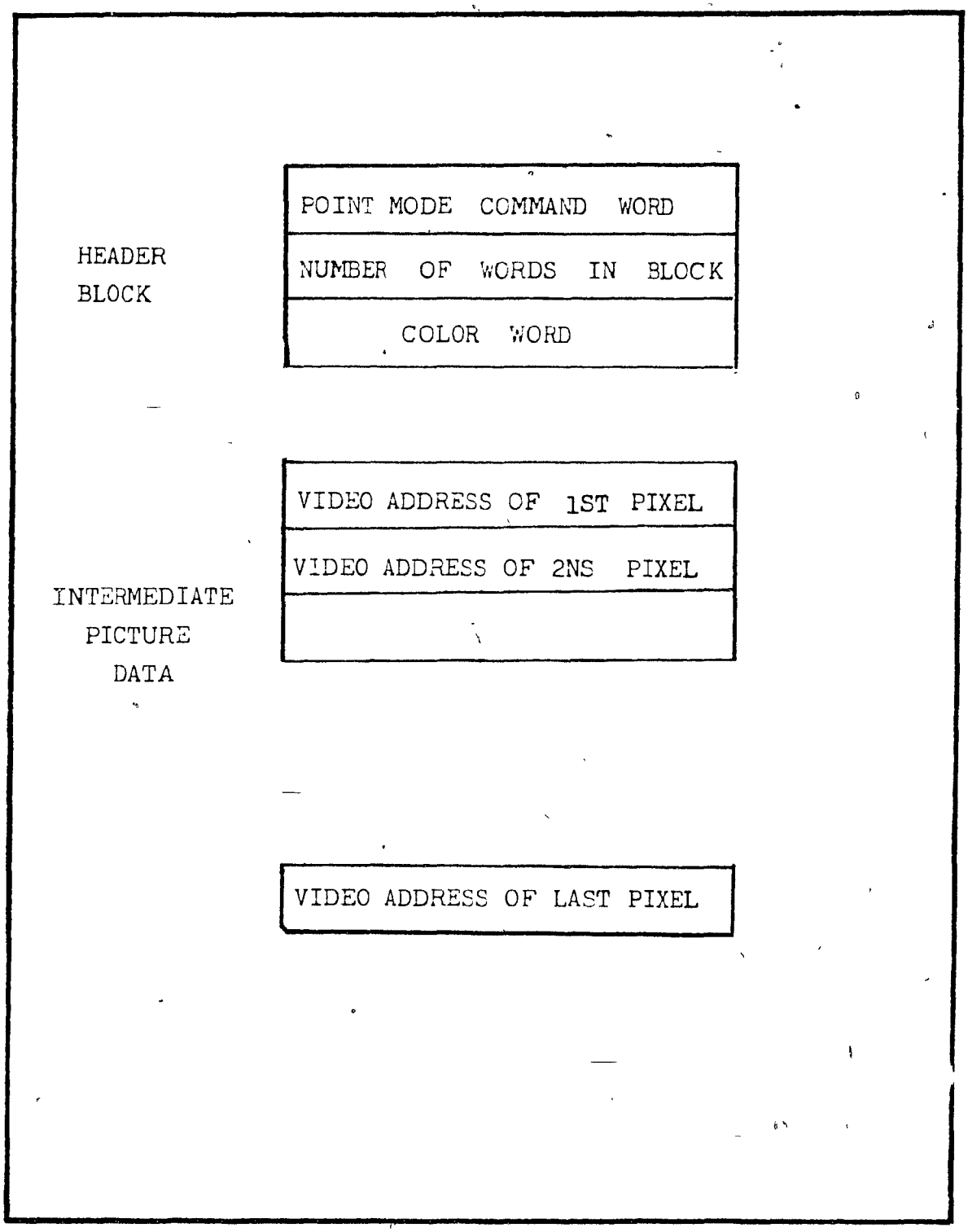


Figure 3-3 Point mode data block

the screan in a chosen uniform color, the second will serve to paint the chosen area with a shaded color, the third can create individual points with the same color and the last one is used for maintenance servicing.

In order to maintain a flicker free TV display during the graphics controler DMA transactions, a constante flow of information to the sequencer must be maintained by giving it the highest priority on the video bus.

VIDEO MEMORY PLANES

To support the constant refrashing rate required by the CRT, the pixel information must be saved in a buffer to be read out continuously to the CRT. Several 64K-bit video memory planes are assigned for this purpose. One 64K-bit memory plane can provide one bit information per pixel for 256 X 256 resolution. The 512 X 512 resolution requires four times as much memory.

Writing of information in each memory plane can be done by over writing (Opaque Mode) or by 'OR'ing the new information with the old one (Transparent Mode). Special hardware circuitry is used on the video memory planes to implement these two modes as speciefied in the "Mode bit" of the COMMAND WORD.

THE T.V. SEQUENCER

The T.V. sequencer is responsible for the synchronization and the continuous refreshing of the image on the RGB color TV monitor.

Shift registers are used to serialize words read on the frame buffer video memory planes. This serial data is fed to D/A convertors which generate the appropriate analog levels for the CRT.

A programmable clock can be operated at 12.6 or 6.3 MHZ to control the flow of the signals for high or low resolutions displays.

CHAPTER IV

OVERVIEW OF THE 2900 MICRO-COMPUTER

INTRODUCTION

As mentioned previously, this micro computer design is based on the motorala M2900 microprocessor families. It is the fastest and most powerful of the GRADS parallel microprocessores. The significant feature of this module is that it can perform a read-modify-write instruction in one clock cycle. Figure 4-1 presents the overall configuration of this module which will be detailed in the following paragraphs.

THE 2900 MICROCOMPUTER MAIN COMPONENTS

In order to obtain a 20 bit word size, the CPU plane has been constructed using five M2901 bit slice microprocessor chips. This module also incorporates three M2909 sequencers units for addressing 4K bytes microcode memory. More than a 100 bit long

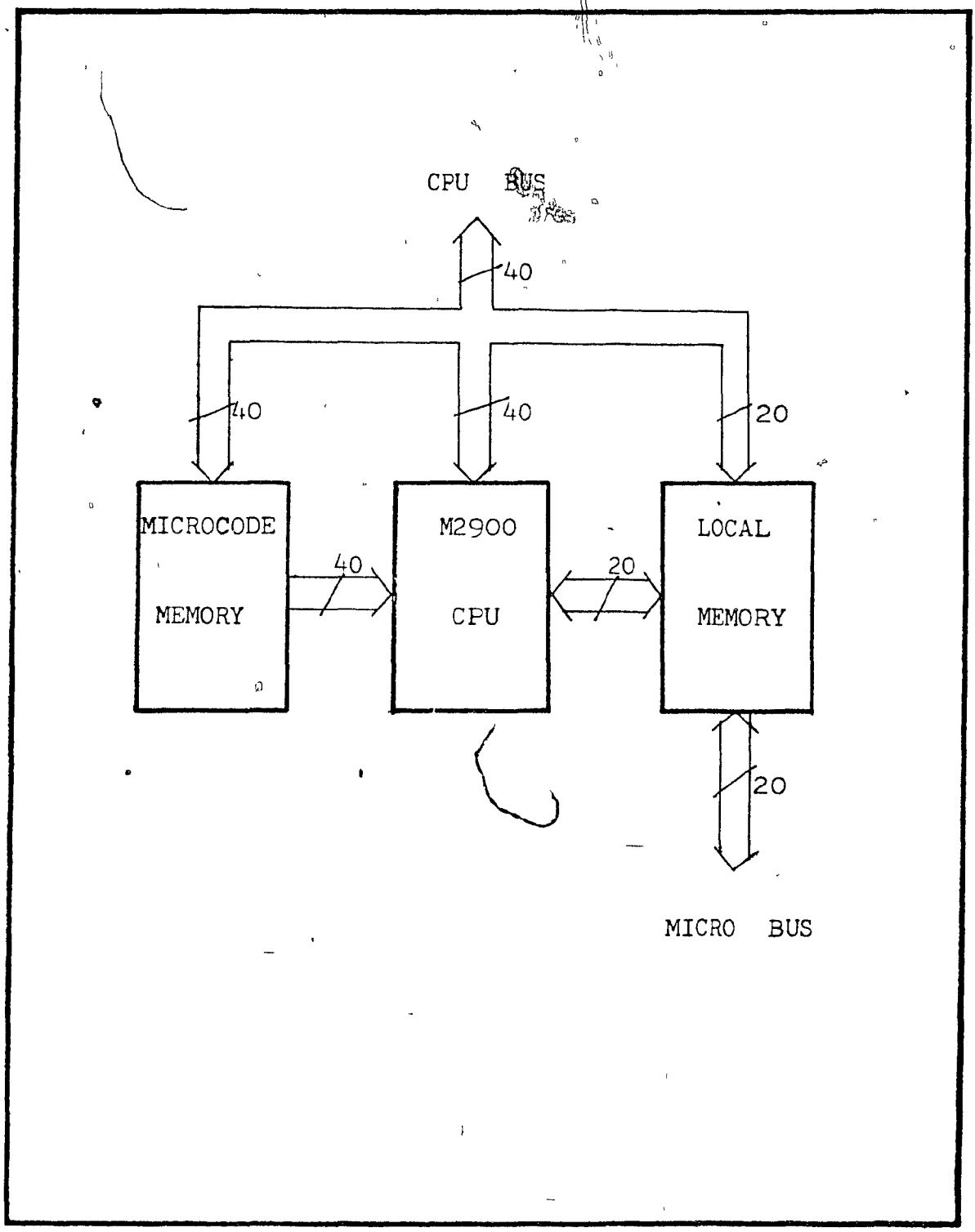


Figure 4-1 Architecture of the 2900 Microcomputer

microcode word would be required horizontally to microcode any of the designed functions in one instruction. The length of the word has actually been minimized to forty bits by dividing the instructions into for classes.

The microcode Memory is constructed, using very fast access time RAM chips (82S11). It is addressable by the CPU and the HCI. The HCI has read and write access to the memory while the CPU can only read from it.

The Local Memory consists of 1K by one bit schottky memory chips (21L02) to give a total of 4K words of 20 bits. Three bidirectional ports make it possible for the CPU, the HCI and the GRAPHICS CONTROLER to access to it(1).

In the following sections the specifications of the 2900 family chips used in this microcomputer are explained.

MC2901 - FOUR BIT TTL MICROPROCESSOR SLICE

The MC2901 bit-slice microprocessor is a four bit machine which can be cascaded to obtain a microprocessor with a longer word size. Since it is microprogrammable it can be made to emulate any digital computing machine by preparing the appropriate microinstructions.

Figure 4-2 presents the architecture of the MC2901 microprocessor. It can be seen that it has two address ports which can simultaneously access any two of the 16 RAM registers, eight arithmetic and logical functions of two operands selected from independent sources can be evaluated at very high speeds of approximately 110 nsec . These operands can be derived from five source ports .and 203 combinations of the source pairs can be obtained by the variation of nine control inputs. These nine control inputs are divided into three functional groups namely, the ALU source operands, the ALU function and the ALU destination register. The ALU results can be delivered at the output bus after each ALU operation or they may be saved in the internal registers through a shifting path. This shifting path can keep the result unchanged or can operate to shift it in either direction (X2 or :2).

Four status flags : carry, overflow, zero and MSB high are outputted after each ALU operation on the separate pins. The CN in conjunction of the carry generate (G) and carry propagate (P) can be fed to a MC2902 LOOK AHEAD CARRY GENERATOR chip when several microprocessors are in parallel.

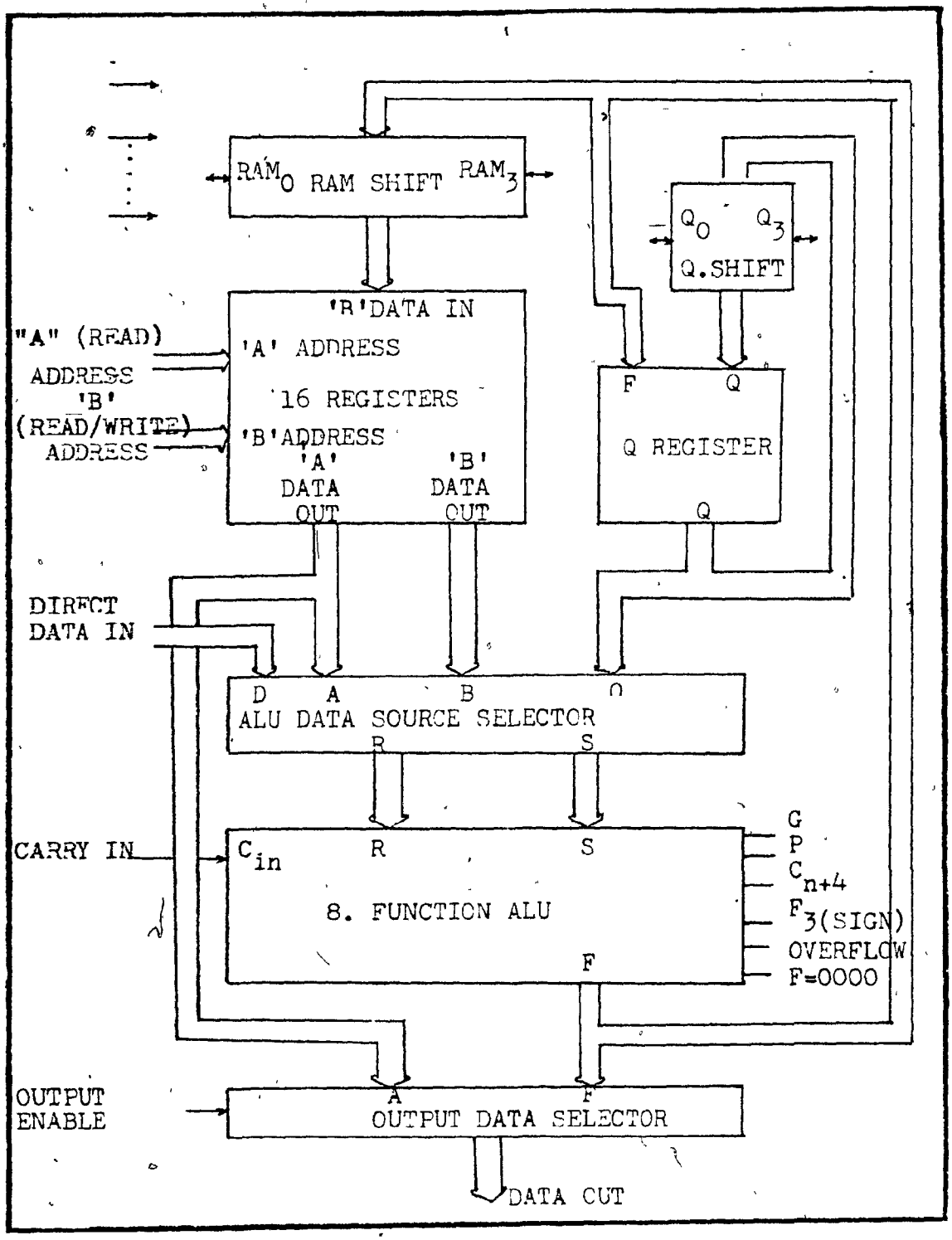


Figure 4-2 Architecture of the MC 2901

MC 2909 MICROPROGRAM SEQUENCER

The MC 2909 microprogramme sequencer is designed to address microinstruction memory sequencially or branch randomly as programmed at its inputs. Figure 4-3 presents its architecture and as can be seen, it contains one address registers and a four deep stack file which can be used for implementing subroutine calls. The output address is four bits long and several of these chips can be cascaded to address a longer microinstruction memory. The "D" input of the chip is used to implement an absolute jump to a chosen location while the zero input is used to initialize the microprogramme execution from address zero.

MC 2902 LOOK AHEAD CARRY GENERATOR

The MC2902 look ahead carry generator has four paires of generate and propagate inputs and is able to generate the carry output for up to four parallel microprocessors. The generate and propagate outputs of this chips are used to cascade several of them for a longer word machine.

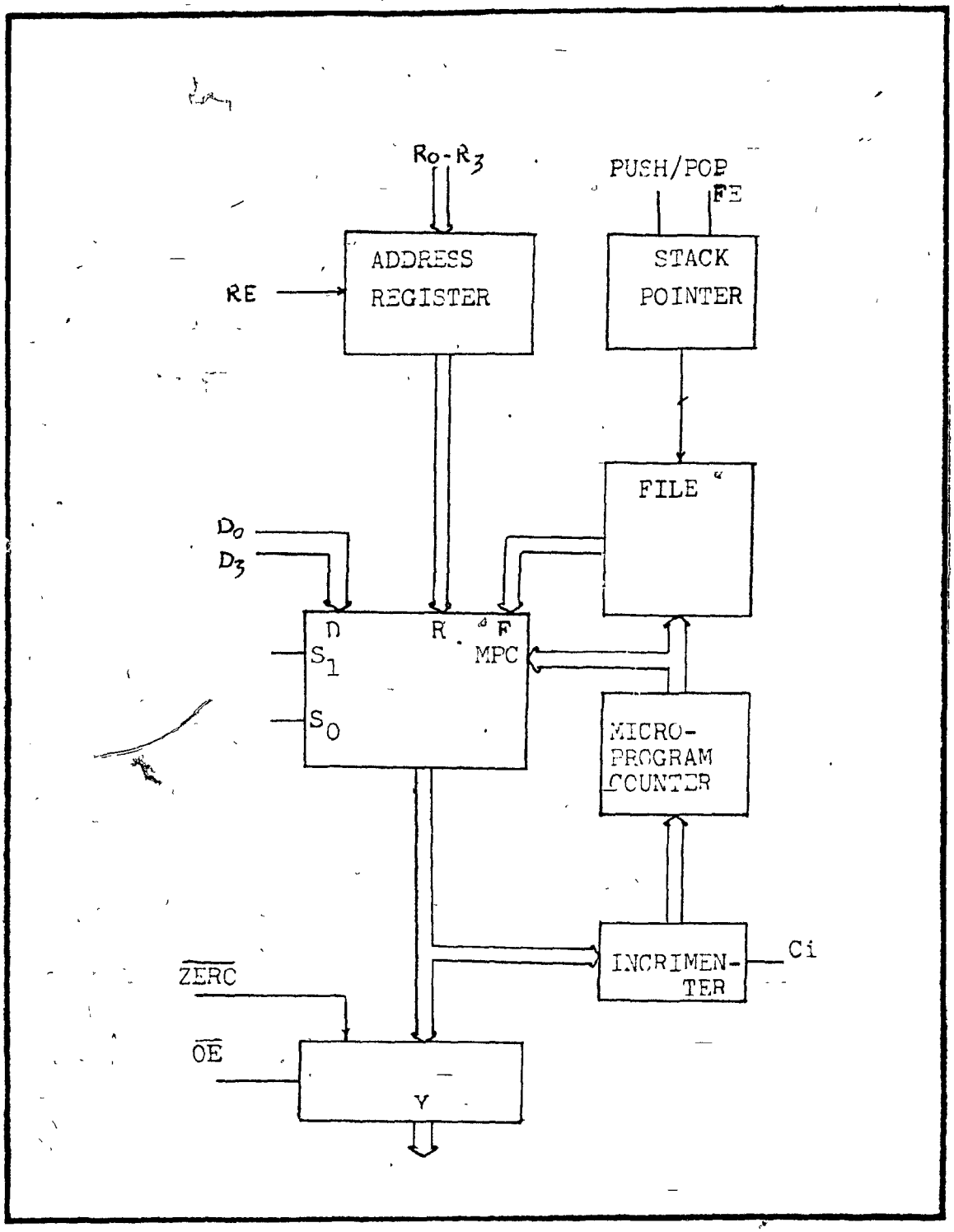


Figure 4-3 Architecture of MC 2909

CHAPTER V

MICROCODE MEMORY

INTRODUCTION

Most images are created from simple elements like point, line, polygon, etc.... The microprocessor module is used to expand such macroinstructions downloaded from the host computer into GRAPHICS CONTROLER instructions placed in an output buffer in the local memory. For highest efficiency, these are executed as microcoded subroutines executed using the parameters specified in the local memory. For example the drow line macro instruction calls a micro-routine where X1,Y1 and X2,Y2, the starting and ending addresses respectively are the transferred parameters.

As mentioned before the Microcode instruction lenght was to designed to be 40 bits and the microcode sequencer addressing de Memoryis awordde are40 be 4K. Thus the the Microco bits long, but the board presently constructed has only 1K words capacity. This capacity can be increased up to 4K if it is necessary.

The Microcode Memory has two individual ports, one bidirectional and the other unidirectional. The HCI uses the

while the bidirectional port for loading and verifying the microcode CPU is allowed "read only access" through the unidirectional port.

MICROCODE MEMORY DESIGN

Microcode Memory consist of four major parts shown in Figure 5-1.

Control Circuit

Receiver/ Drivers

Address Buffers

RAMs

In addition, several controls and busses are interconnected.

Four control signals come from the HCI. The function of these signals will now be explained.

DIR_HCI_TO_RAM H: A high level in this line, opens the data flow from the Microcode bus to the RAMs. A low level on this line reverses the data flow.

HCI_PORT_L: A low and high on this line selects between the HCI address and CPU address respectively.

WRITE_L: THIS low pulse sent by the HCI causes 40 bits

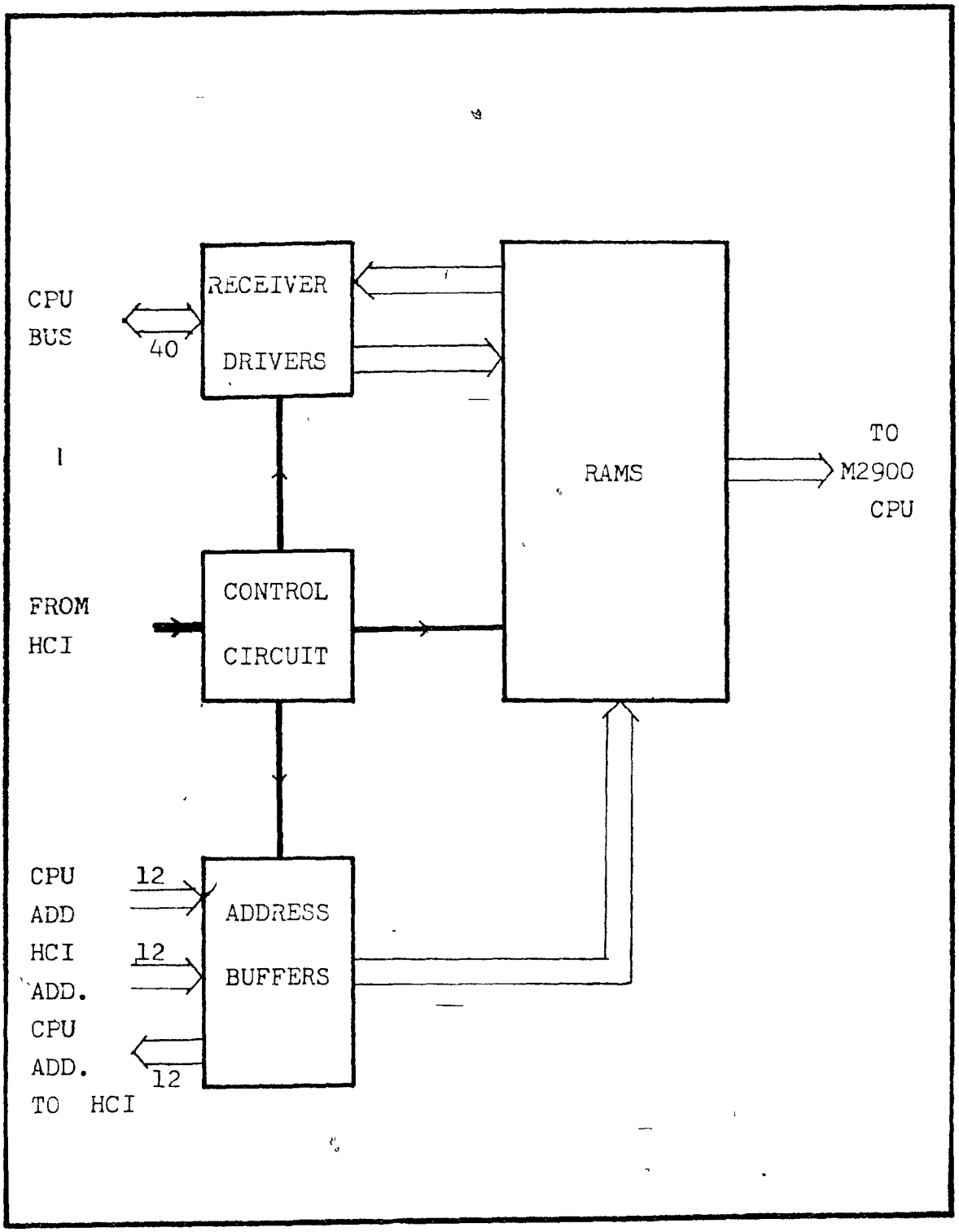


Figure 5-1 Microcode Memory configuration

of data presented at the HCI port to be saved in the RAMs at the set address. This line remains ineffective when " HCI_PORT_L" is high.

CPU_ADD_OUT_ENABLE_L: A low level on this line enables drivers to output the CPU address to the "CPU ADDRESS TO HCI" bus. This is useful for the tracing mode software which is explained in chapter 8.

THE CONTROL CIRCUIT

The control circuit is shown in Figure 5-2.

Two LEDs are assigned to show the flow direction. This helps in the hardware debugging and maintenance modes.

D1 : OFF, when the flow direction is from bus to RAM.

D2 : OFF, current "CPU ADD TO HCI" buffer is tristated off.

RECEIVER / DRIVERS

The receiver/driver unit consists of ten 8T26 chips each carrying 4 lines of data (Figure 5-3).The two lines governing the

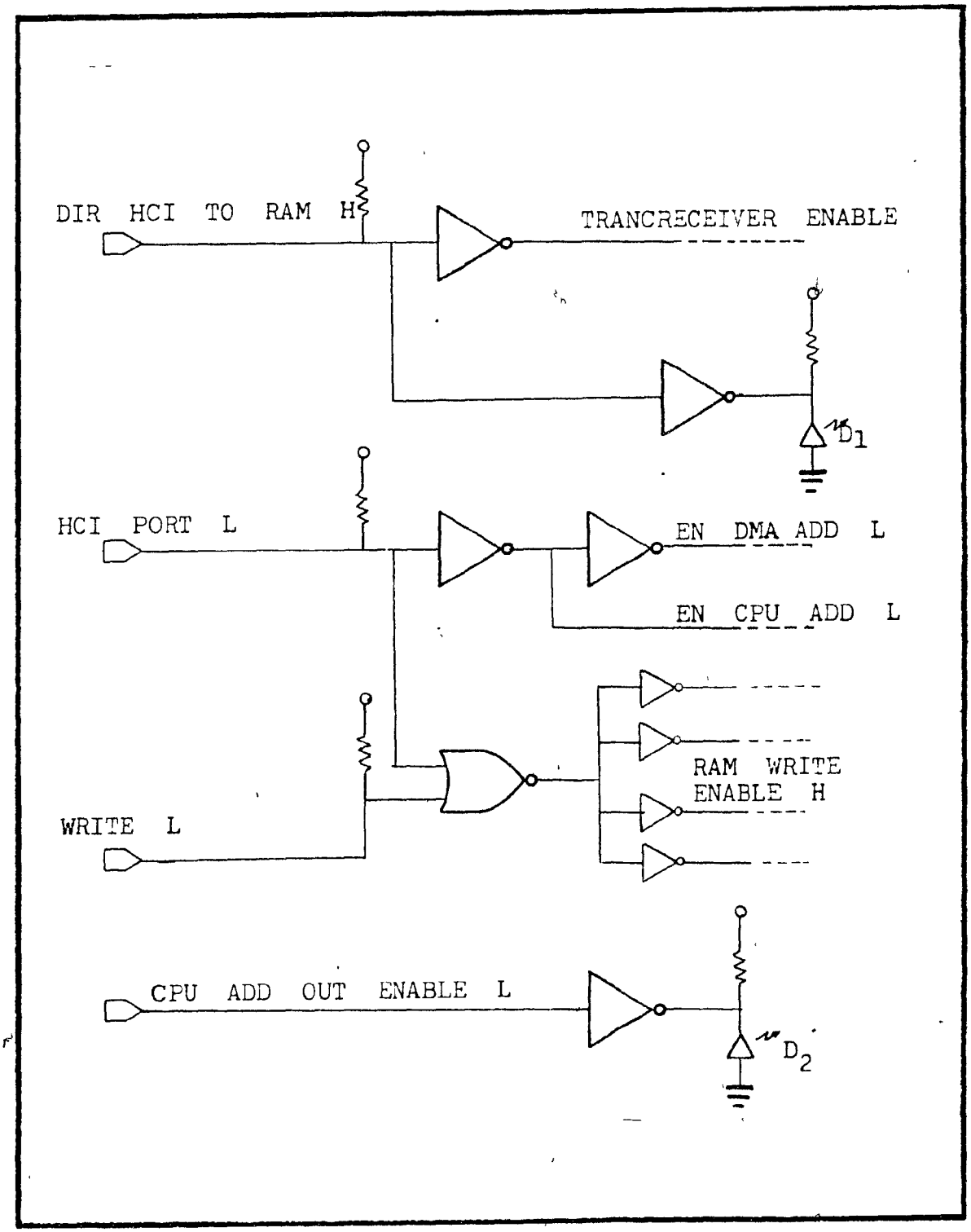


Figure 5-2 Control circuit of the Microcode Memory

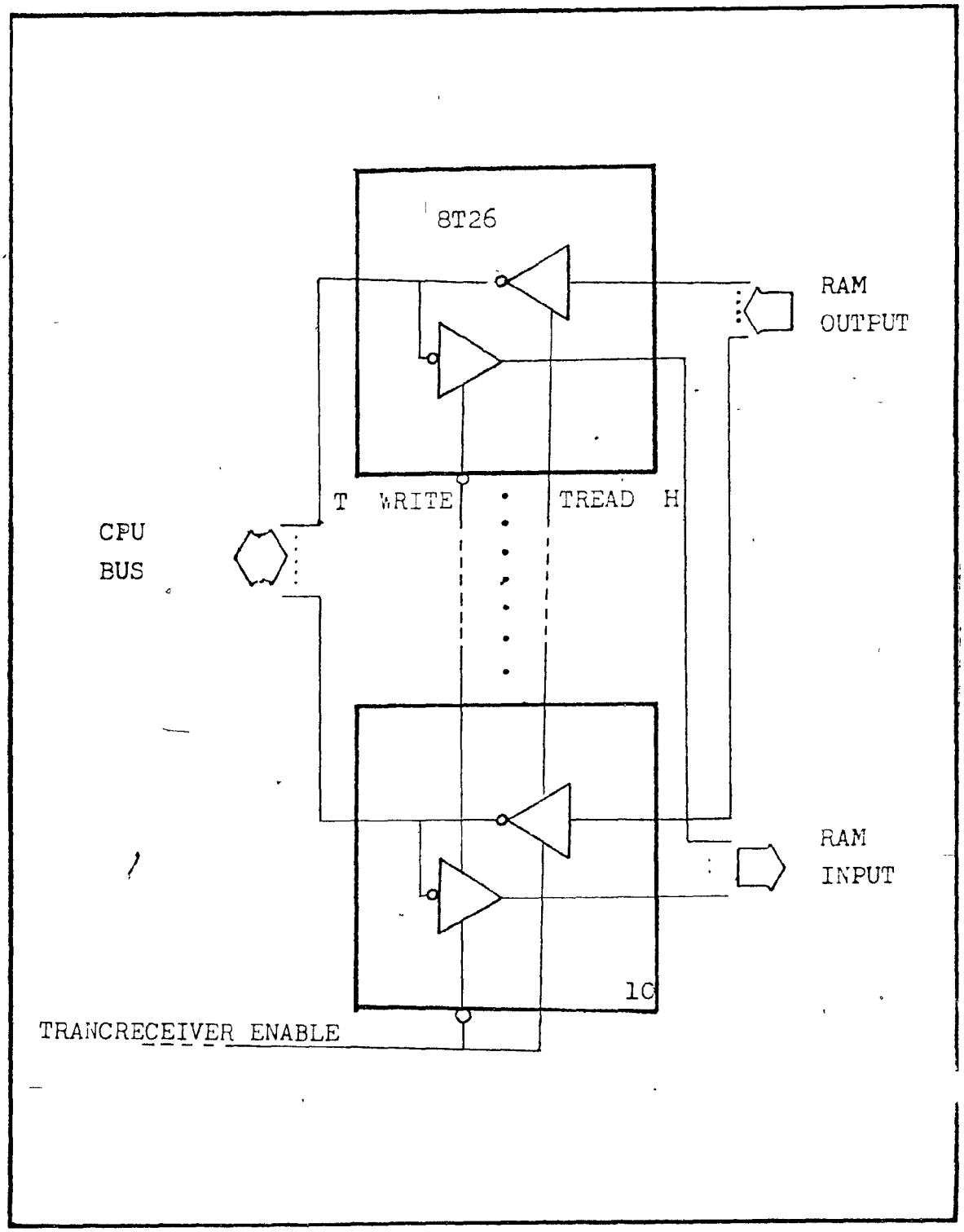


Figure 5-3 Receiver/Driver of the Microcode Memory

selection of information flow " TREAD H " and " T WRITE L " are driven in parallel by the control output, which is the inverted " DIR HCI TO RAM H " signal. A low at these lines dictates a flow from the CPU bus on to the inputs of the RAMs, and a high causes a flow from the outputs of the RAMs to the CPU bus.

These chips invert as they transmit or receive data because data presented by the HCI is inverted.

As shown in Figure 5-1, no drivers are assigned to the "2900 CPU" bus because this bus is private between CPU and the Microcode RAM and each output of the RAM is driving only one input on the CPU card.

ADDRESS BUFFERS

The address buffers consist of three groups of buffers, of which one group is inverting (8T96) and two groups noninverting (8T95) (Figure 5-4). In each group, two parallel drivers are used for each signal in order to meet the fan out requirements.

The inverting buffer lines carry 10 HCI address lines and buffer 4 write signals. The inversion corrects a prior inversion that occurs in the HCI.

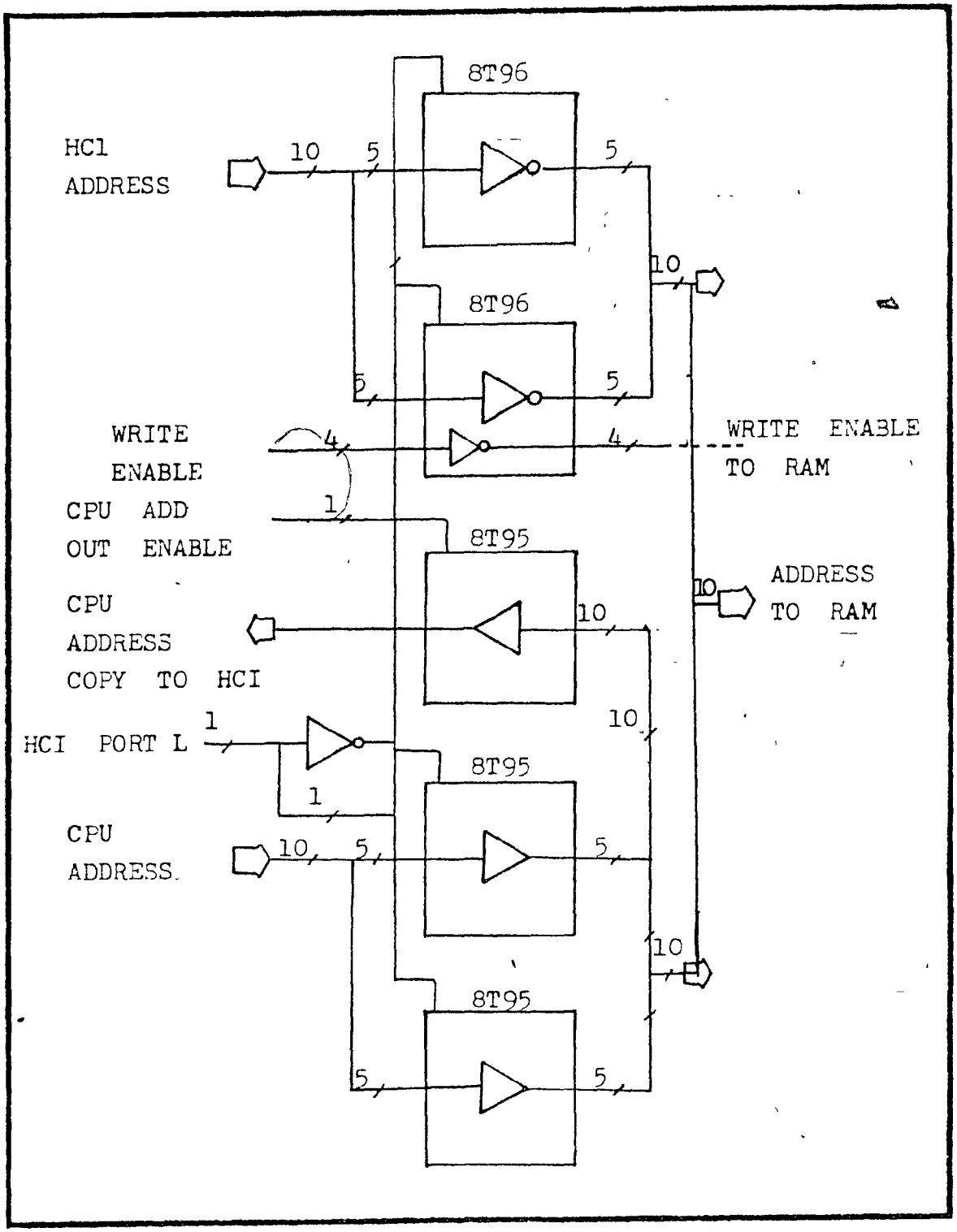


Figure 5-4 Address buffers of the Microcode Memory

One of the noninverting groups presents the copy of the current CPU addresses to HCI when needed. This request occurs during tracing and simulation tests. The last noninverting group carries the current CPU addresses to the Micro-code RAMs.

RANDOM ACCESS MEMORY

The memory portion of the circuit consists of 40 schottky RAM 1024X1 bit chips (82S11) shown in Figure 5-5. They are divided in two groups, each carrying the same address but driven by different drivers.

Each input to these forty RAM chips is fed from a driver coming from the HCI while each RAM output is connected to the "2900 CPU" bus as well as the input of the driver going to the HCI.

As shown in Figure 5-2, four buffers are used to meet the fanout requirements of te "Write Enable" inputs of these forty chips.

MODES OF OPERATION

Microcode Memory has three distinct modes of operations namely DMA write, DMA read and CPU read.

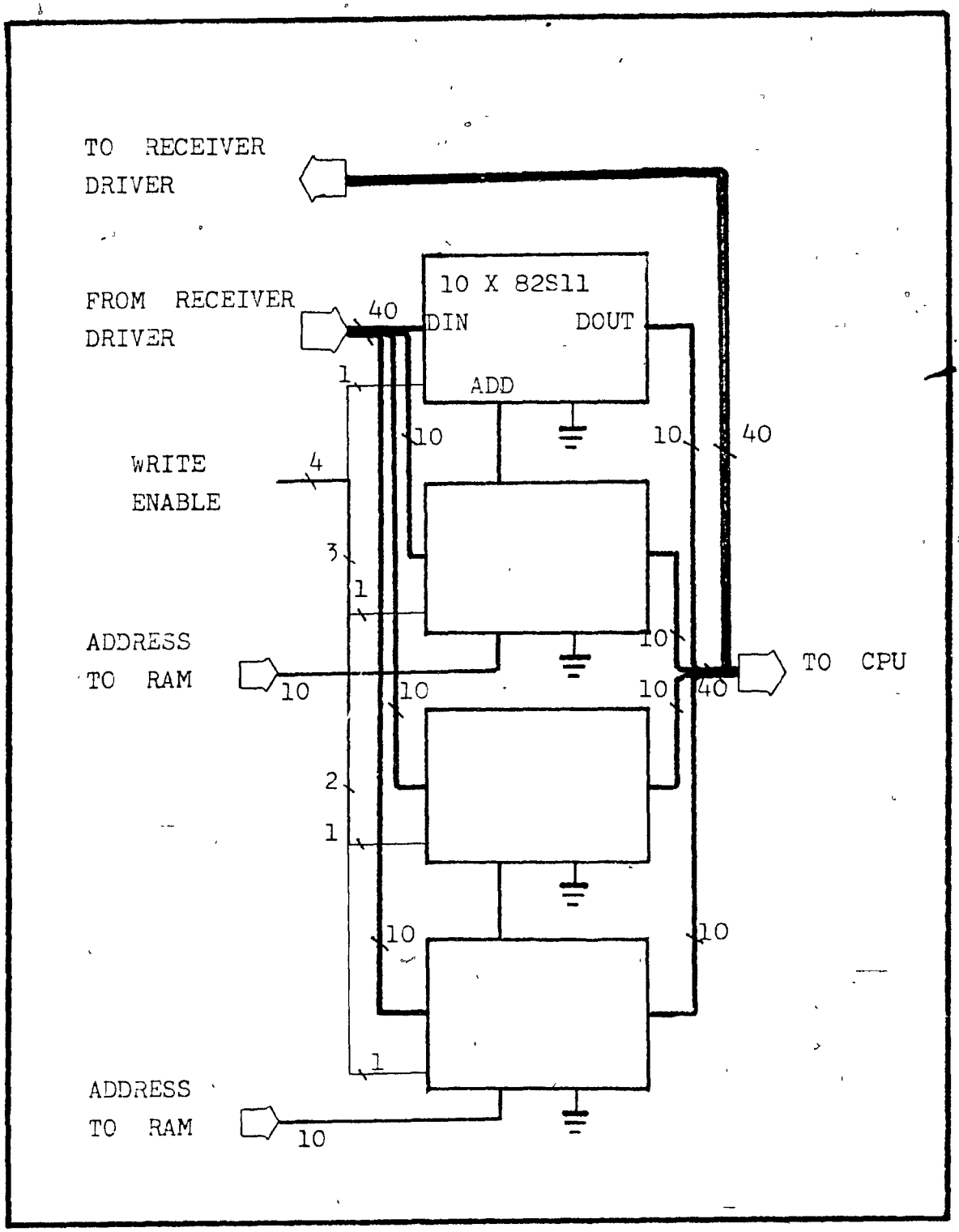


Figure 5-5 RAMs of the Microcode Memory

In the DMA write mode, the HCI sends 40 bits of data to the CPU bus and presents the microcode memory with 10 bits of address as well. Both the data and the address appear inverted. After initiating the write cycle, the "WRITE L" pulse is generated. Figure 5-6,a shows the timing diagram for this mode. On the trailling edge of this pulse, the data which is presented at the input of the RAM's is written into the RAM's at the address location.

In the read mode, the HCI receives 40 bits of data via the CPU bus. This data will correspond to the location whose address is presented by the HCI along the 10 address inputs of the RAMs. In this case the control signals shown in Figure 5-6,b are initiated by the HCI and are sent to the Microcode Memory. In this mode the same data is also presented to the CPU, but the CPU does not use it.

In the Read CPU mode, the CPU of the microcomputer presents the Microcode Memory with 10 bits of address and receives 40 bits of data. In this case all the controls, " DIR HCI TO RAM H ", " WRITE L " and " HCI PORT L" must be maintained high by the HCI.

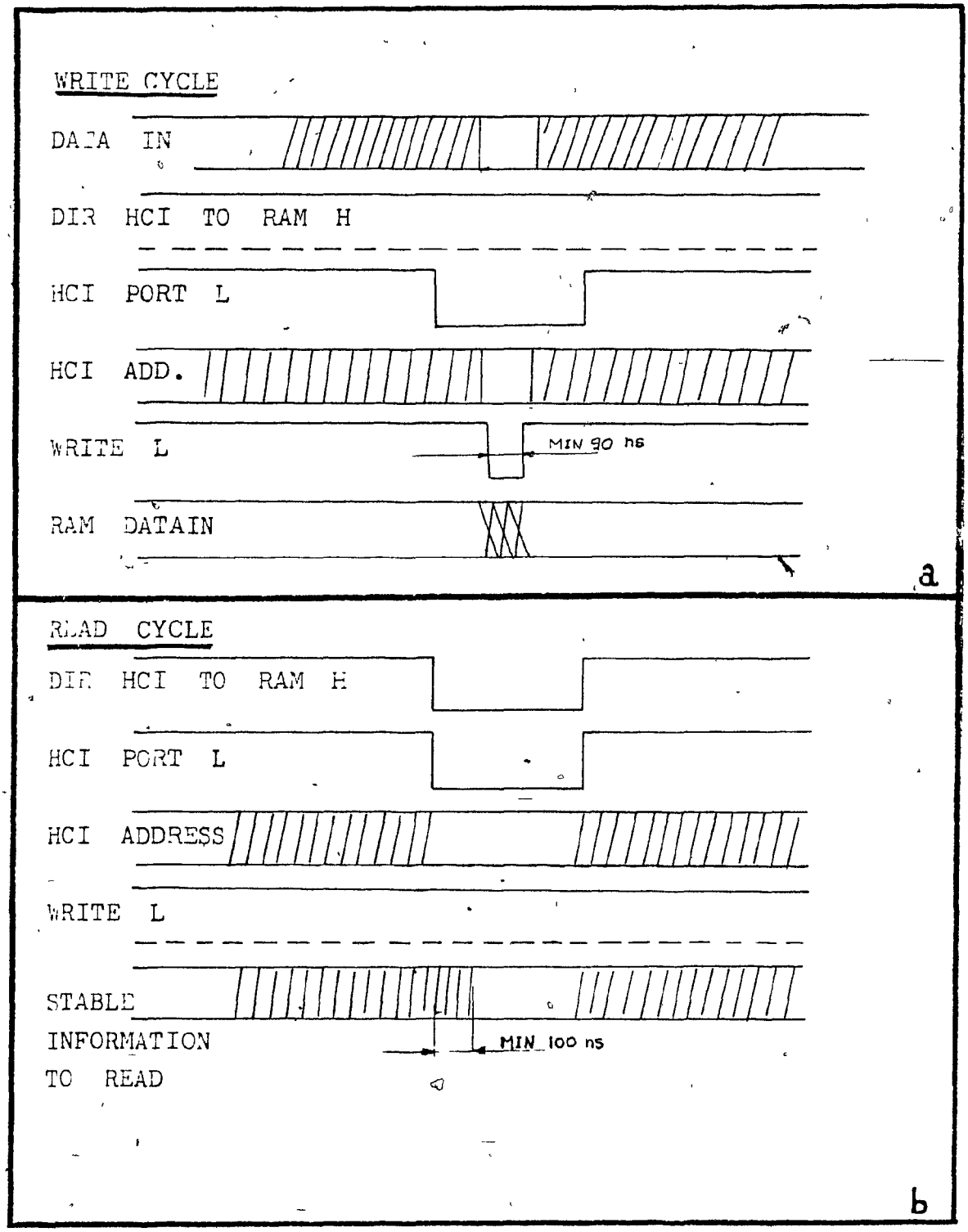


Figure 5-6 Microde Memory Read and Write timing diagram

CHAPTER VI

LOCAL MEMORY

INTRODUCTION :

The Local Memory is used to implement two buffers. The HCI fills the first one with the macroinstructions. The GRAPHICS CONTROLLER reads the second one once it is filled by the CPU with processed the data.

Basically the Local Memory design has three bidirectional ports and has relatively complicated control logic because of the arbitration requirement associated with Microbus, HCI and CPU ports, listed in decreasing priority.

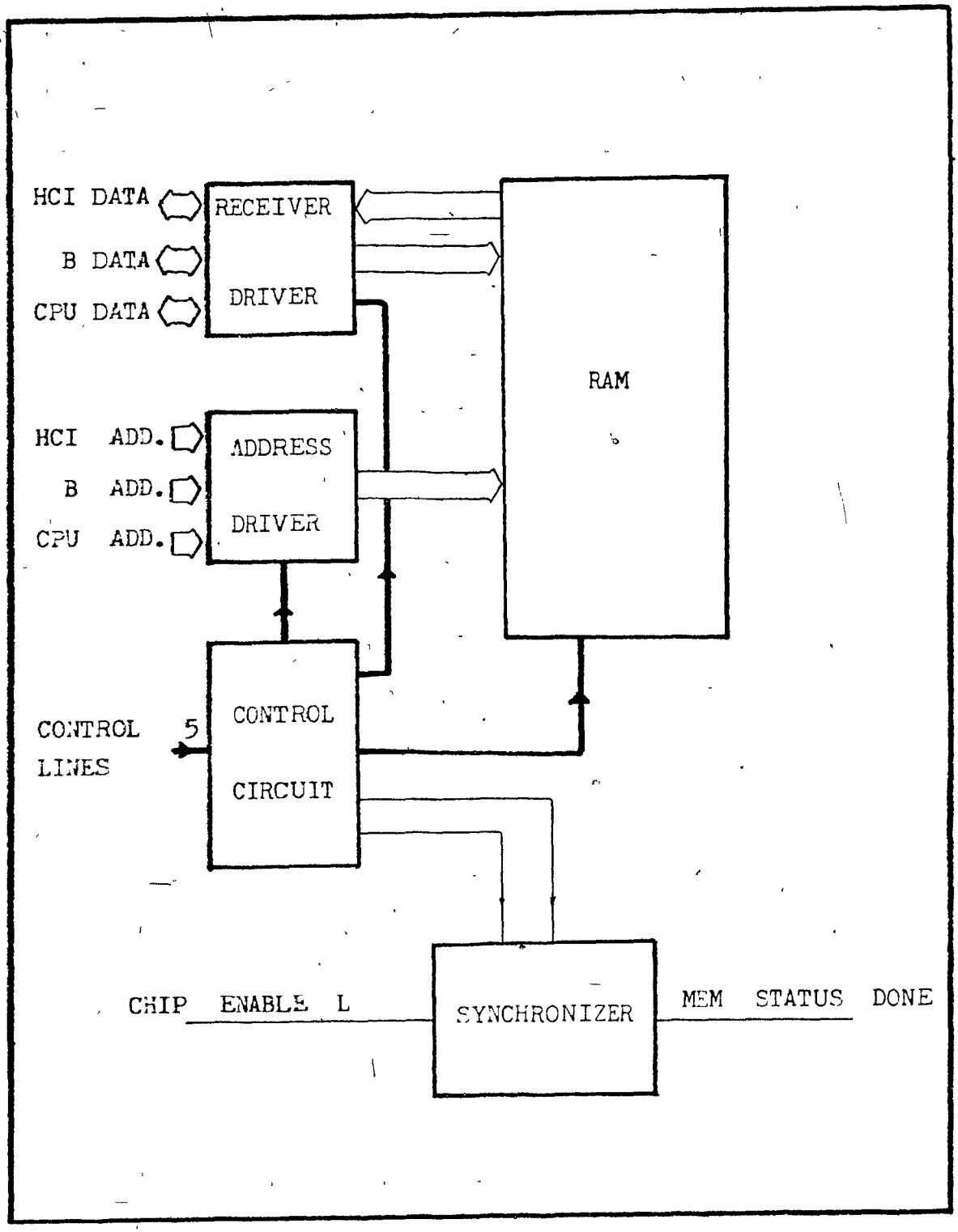


Figure 6-1 Local Memory Major Parts

In order to syncronize the Memory cycles, a logic "Hand Shaking" circuit is included.

Since the 'Memory Address Register' (MAR) of the CPU (will be explained in Chapter VII) is 12 bits long, the CPU is able to address a maximum of 4K words of local Memory, each word being 20 bits long.

LOCAL MEMORY DESIGN

The Local Memory Consists of five major parts:

Control Circuit

Data Receiver/ Drivers

Address Buffers

Synchronizer

RAMs

In addition to these parts, several other controls and busses are also shown in Figure 6-1

Five control signals are connected to the Local Memory. The function of each signal is explained as follows:

MICROBUS GRANT L: This signal is provided by the Local Memory arbitrator, located on the CPU board. It is active low and it results from a Micro bus request which has the highest priority.

HCI GRANT L: This signal is also provided by the Local Memory arbitrator. It is active low and it results from an HCI request, which has the second priority.

CPU GRANT H: This is the lowest priority signal provided by the Local Memory arbitrator. It results from a CPU request.

The above three signals enable the appropriate address drivers to Local Memory.

HCI READ L: This signal is provided by the HCI to regulate the HCI data port direction, Logic low will cause a read from Local Memory to HCI.

MICROBUS READ L : This signal is provided by the GRAPHICS CONTROLER to regulate the MICROBUS DATA port direction. Logic low will cause a read from Local Memory to the MICROBUS.

If all the five signals mentioned above are high then the

CPU will be able to read from the local Memory .

THE CONTROL CIRCUIT

The major chips in this circuit are the multiplexer and the two decoders. These three units, in conjunction with some inverters and drivers, take care of switching the three data and address busses. The output of the multiplexer is also used to enable the write pulse provided by the synchronizer (Figure 6-2).

The two decoders are fed by the same signal but only one of them is enabled by the multiplexer output at a time. The first decoder enables the buffers to the RAM input. The second, enables buffers to the DATA busses. This circuit guarantees that only one buffer is enabled at a time.

Three diodes D1,D2 and D3 are provided to display buffer transmission to the HCI,MICROBUS and CPUBUS respectively.

DATA RECEIVERS/ DRIVERS

The Local Memory has three identical bidirectional ports. Except for the CPU DATA port,

the rest are inverted in order to correct the polarity of the data. Each port consists of 40 drivers which are coupled in pairs in order to have 20 bit bidirectional ports (Figure 6-3).

ADDRESS BUFFERS

One of the three groups of addresses must be presented at the RAM's address inputs according to the result of the arbitration (Figure 6-4). The nand gate enables the CPU ADD lines

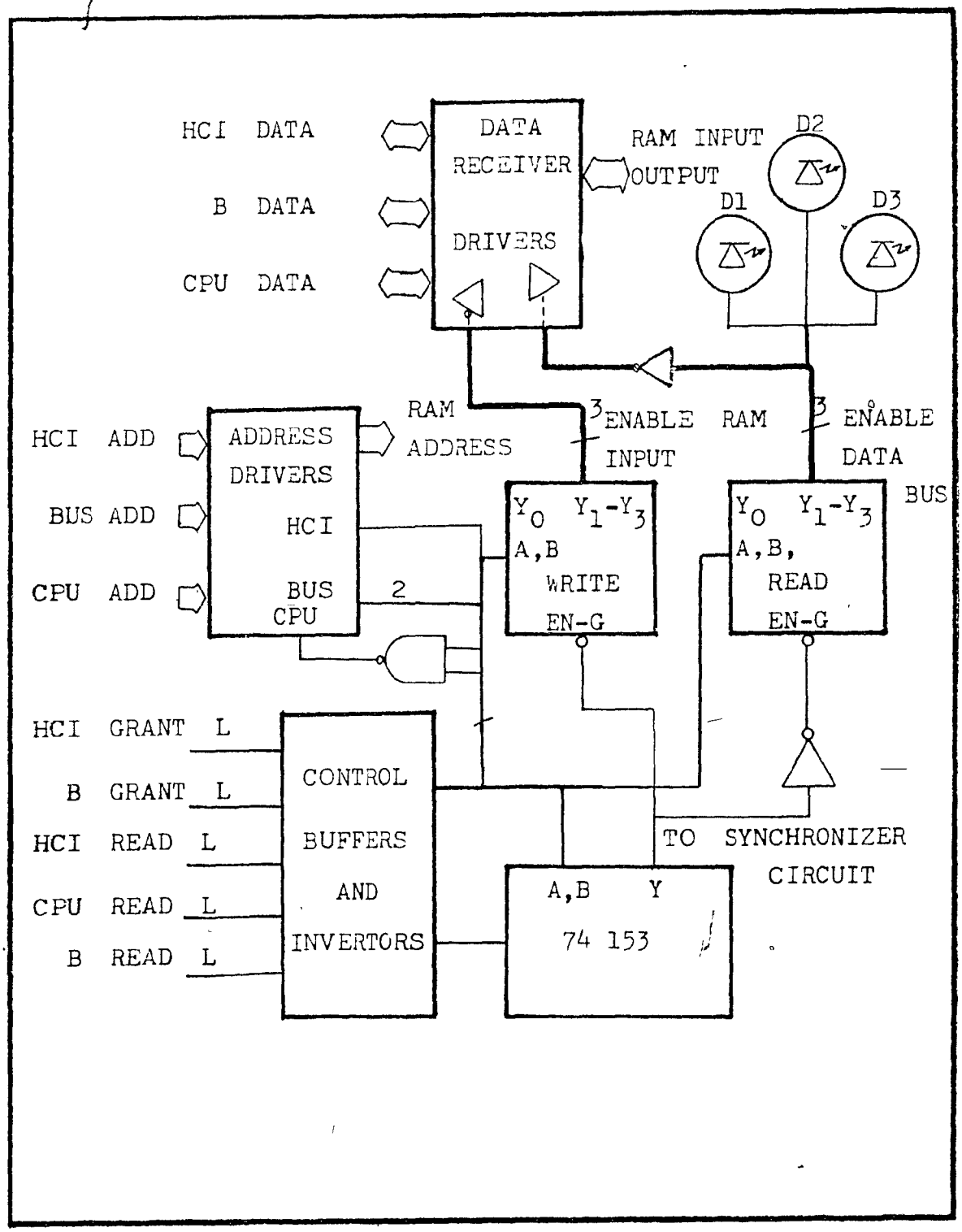


Figure 6-2 Local Memory control circuit

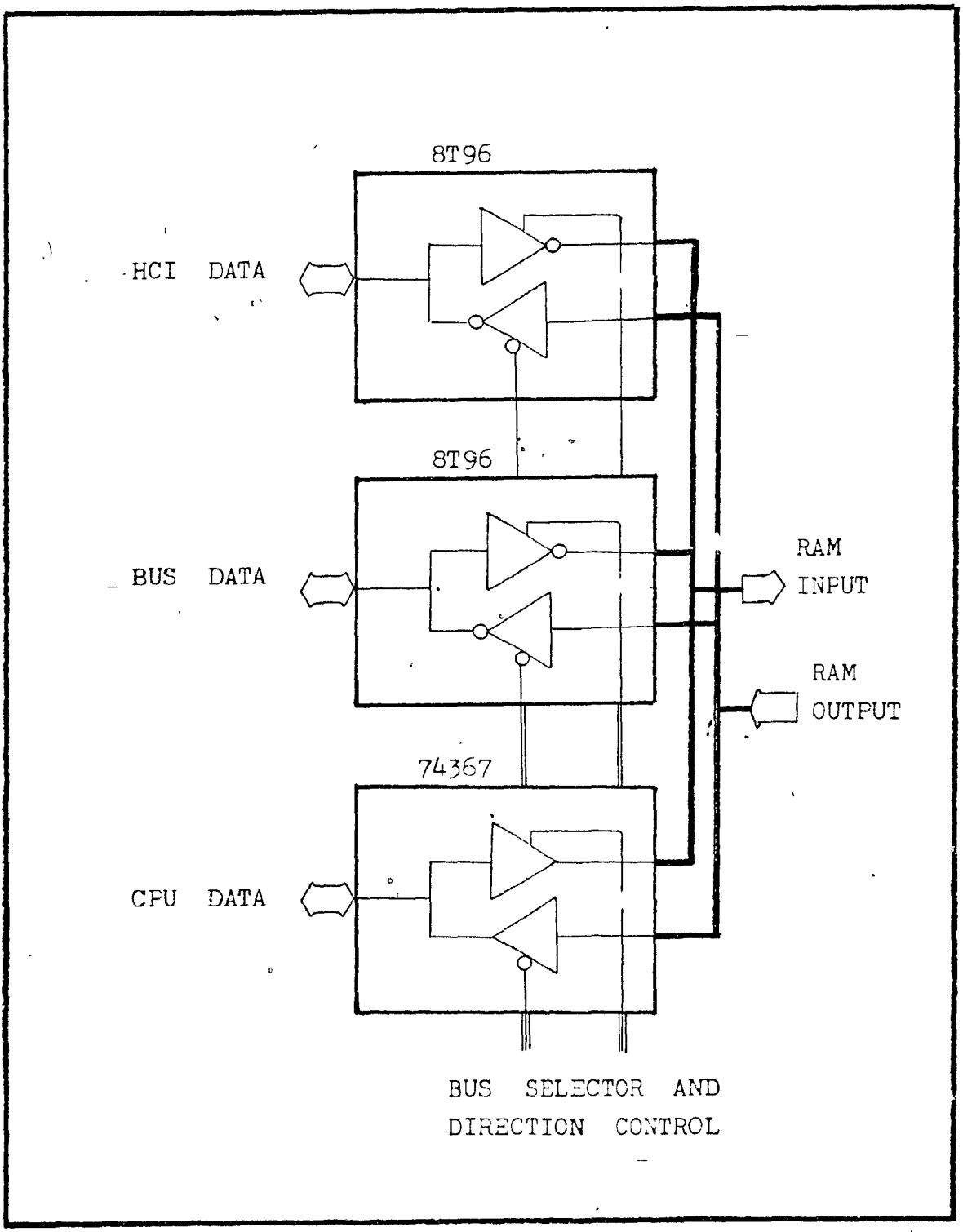


Figure 6-3 Local Memory Data Receiver/Drivers

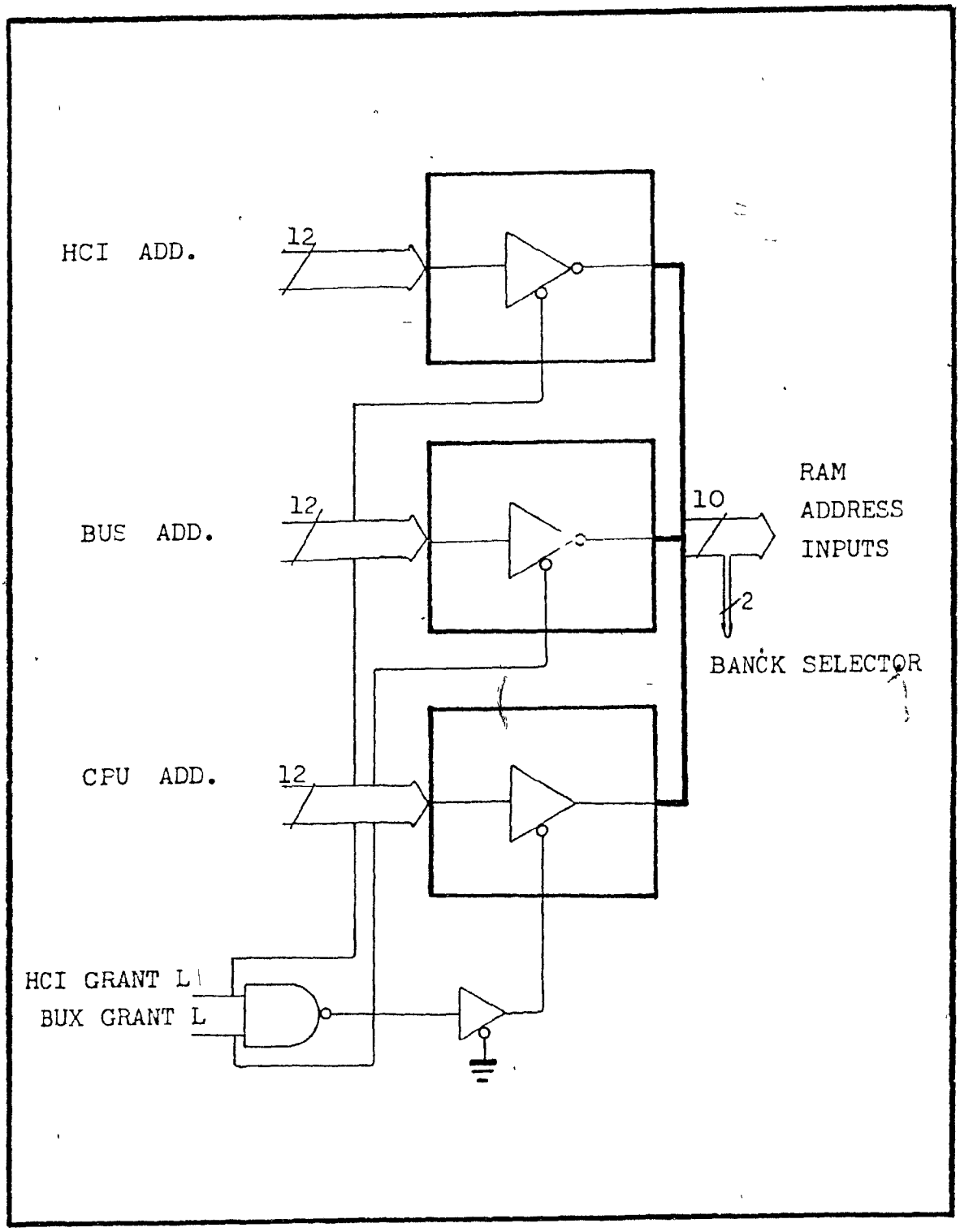


Figure 6-4 Local Memory Address Buffers

when none of the other address buffers are selected.

RAM's

The Local Memory was designd, using 80 chips of 1K bit static RAMs (21L02). These RAMs are grouped in four rows, in order to realize 4K words of memory. Each group has 20 chips to create a 20 bit long (Figure 6-5).

Each RAM chip has only 10 address inputs. In order to be able to address 4K bytes, the two most significant bits of the address buffers are connected to one selector. With the four outputs of this selector, one of the RAM groups can be chosen by enabling their "CE" input.

The 'write enable' (WE) of all the chips can be shorted together for the "WE" will be effective only in the selected memory bank. In practice, however, it was found desirable to use one more selector to direct separate write pulses to each bank selected. This divides the load by four and helps to improve the noise and fanout problems.

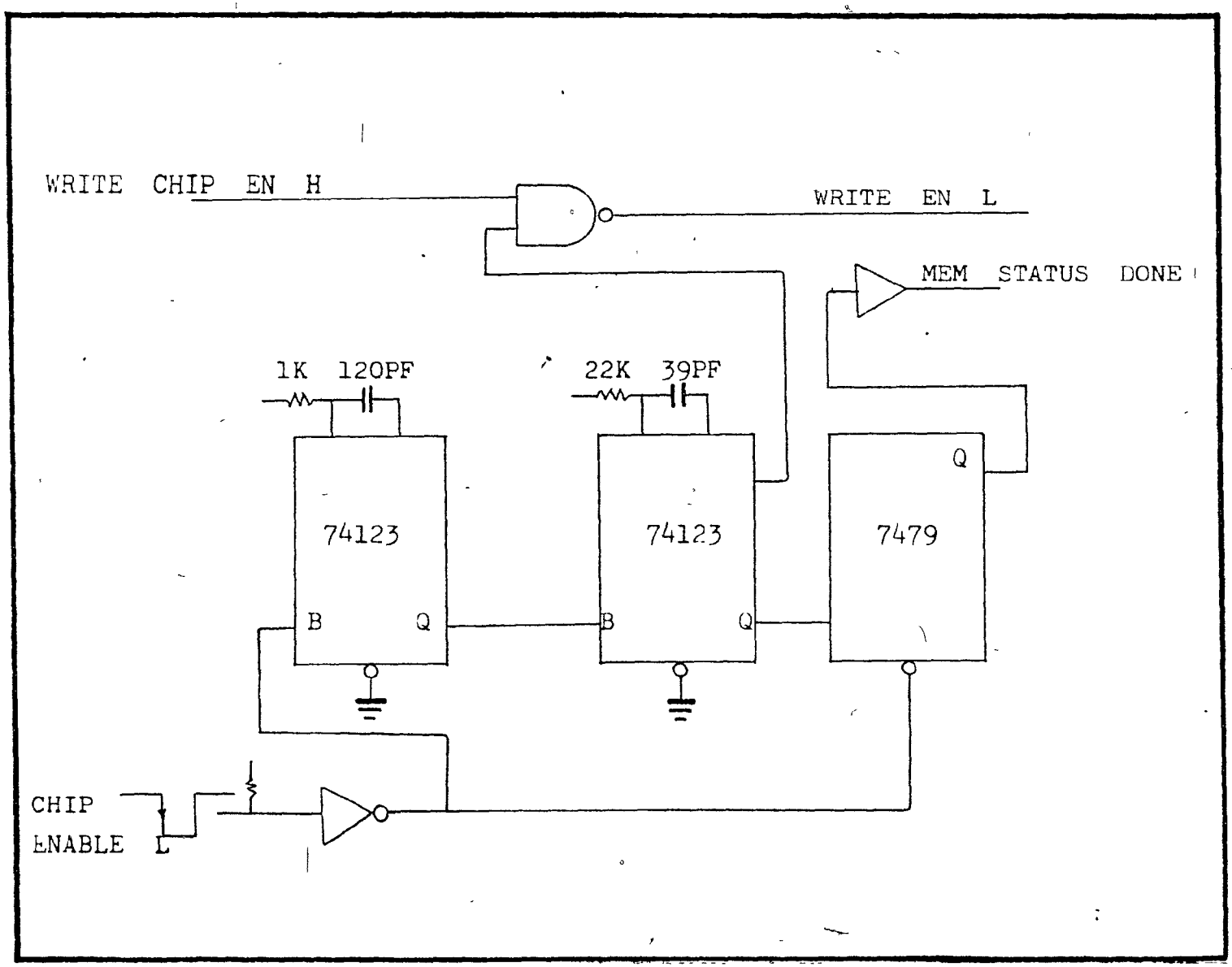


Figure 6-5 Local Memory Synchronizer

SYNCHRONIZER

The Local Memory can be accessed by three systems with different speed. In order to handle this operation, the Synchronize has been included in the Local Memory (Figure 6-6). This circuit consists of two timers and one flip flop to regulate a "hand-shaking" protocol between the Local Memory and the external devices. The first timer is used to adjust the address setup time and the second one is used to adjust the writing or reading time. The flip flop is used for generating the done status pulse.

In the next few paragraphes operation of this circuit will be explained. The Local Memory arbitrator, by granting any request, pulls down the "CHIP ENABLE L" line. This signal triggers the first timer and releases the flip flop. This first timer interval is used for address decoding, bank selection and memory address setup time (Figure 6-7). The trailing edge of the first timer pulse triggers a second timer. The second timer pulse is used to generate a memory write pulse during memory write transactions only. The trailing edge of the second pulse clocks the flip flop to set the "memory status done" pulse which is sent to the arbitrator. This pulse resets the arbitrator and therefore the "CHIP ENABLE L" is removed after a certain gating delay. For the read mode, valid data can be read at the leading edge of the "memory done status" pulse. The protocol requires that the request to the Local Memory

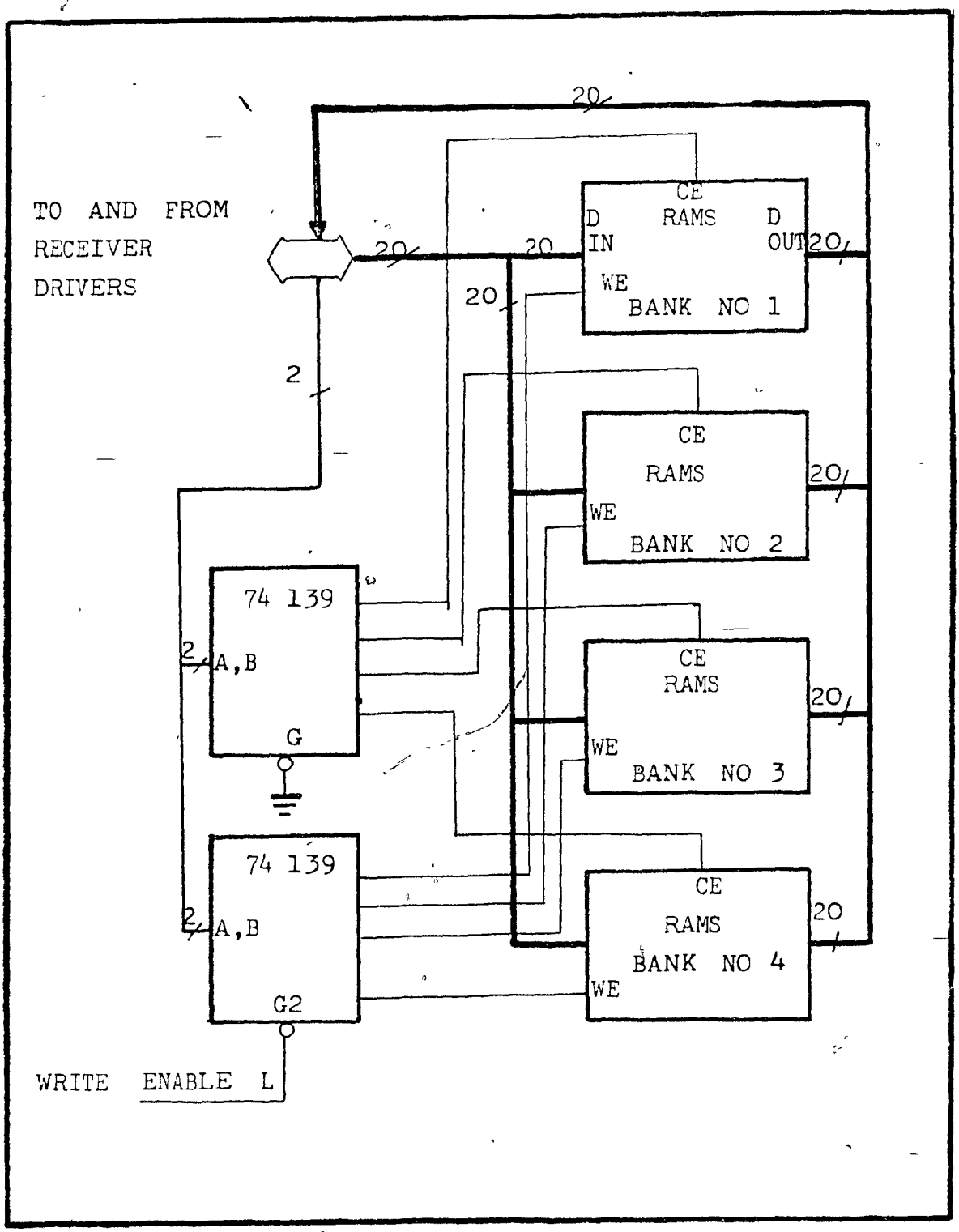


Figure 6-6 Local Memory RAMs

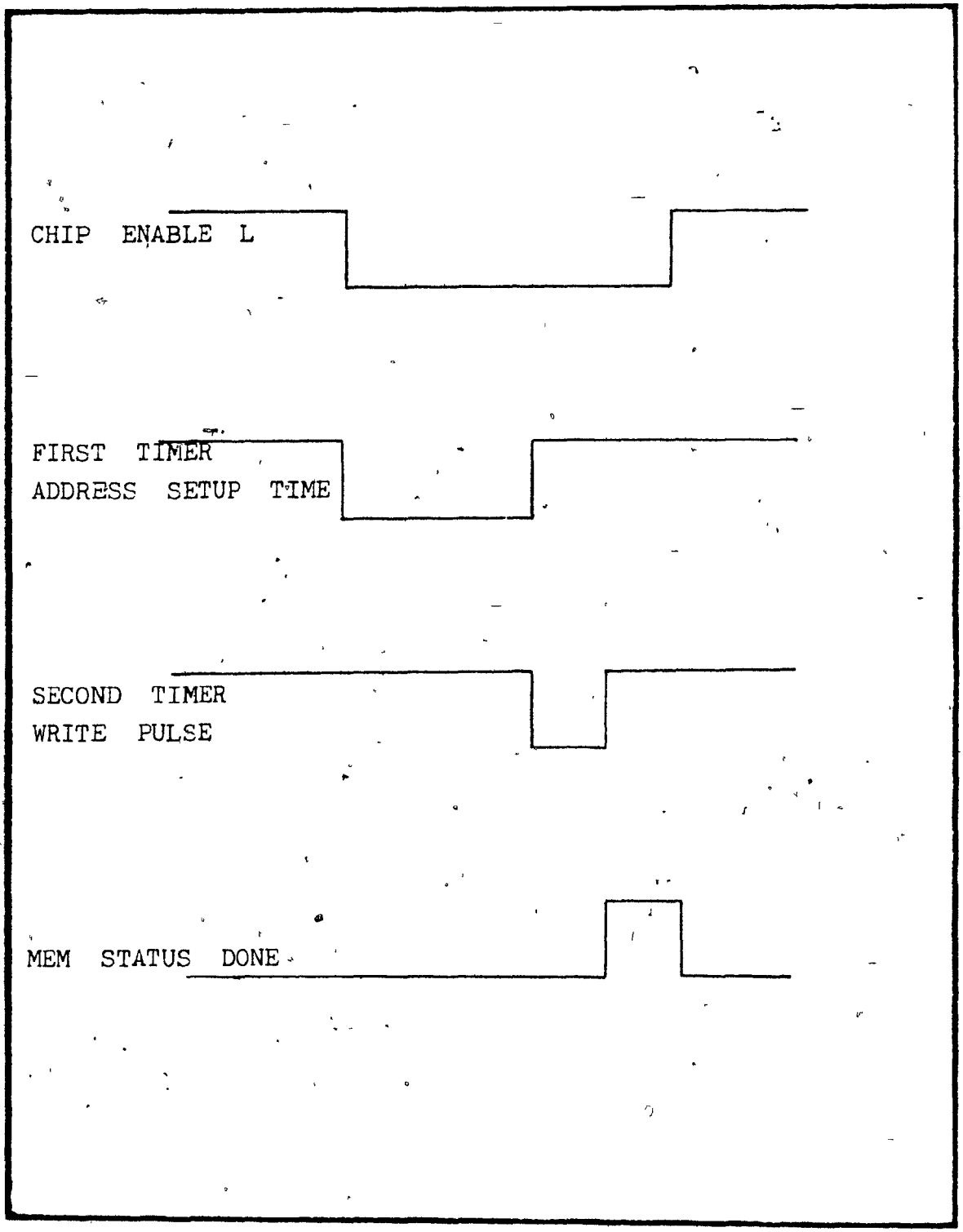


Figure 6-7 Local Memory Senchronizer timing diagram

arbitrator be removed to disassert chip enable L. This results the "Memory Done Status" before the next cycle.

CHAPTER VII

THE M2900 CPU

INTRODUCTION

As mentioned before, several microcomputers will participate in breaking down the macroinstructions into lower level information. The M2900 is one of these microcomputers. The CPU of this unit is based on the M2900 microprocessor family. To make the unit as powerful as possible, several other circuits have been added to this unit. A 16 bit by 16 bit multiplier and the external registers are the examples of the extra hardware features.

The CPU can communicate with the local Memory and the Microcode Memory, through individual and private buses, and with the other CPU's and the HCI through the CPU BUS. Thus, the CPU has three ports, one unidirectional to read from the Microcode Memory and two bidirectional ports to communicate with the HCI and the Local Memory. A group of 14 wires are assigned to transfer the status of the CPU to the HCI.

The Local Memory Arbitrator is included on this board to arbitrate the access requests of the GRAPHICS CONTROLLER, HCI

and the CPU with the Local memory in case of simultaneous requests.

In the rest of this chapter a detailed description of the hardware making up the M2900 CPU board is presented. This design is broken up into seven distinct sections:

1- Central processor
2- Sequencer and branch controls
3- Internal, external registers and the multiplier
4- Register controls and interrupts
5- Status
6- Local arbitrator
7- Microword register or Pipeline

Figure 7-1 shows these sections and their internal and external connections.

CENTRAL PROCESSOR

The Central Processing unit (CPU) consists of five MC 2901, 4 bit-slice microprocessor chips of the 2900 family, connected in parallel in order to obtain a twenty bit processor (Figure 7-2).

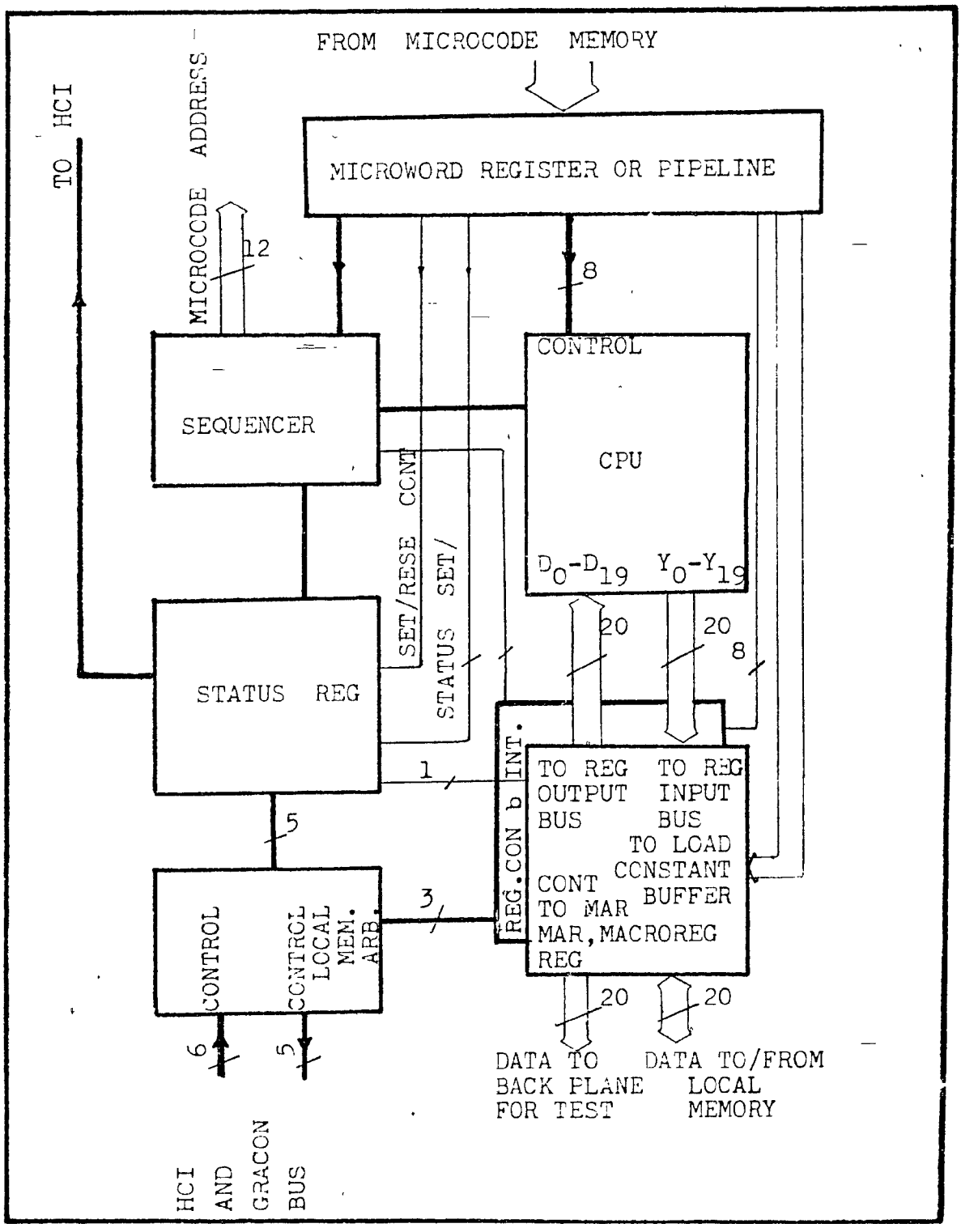


Figure 7-1 2900 CPU subsections

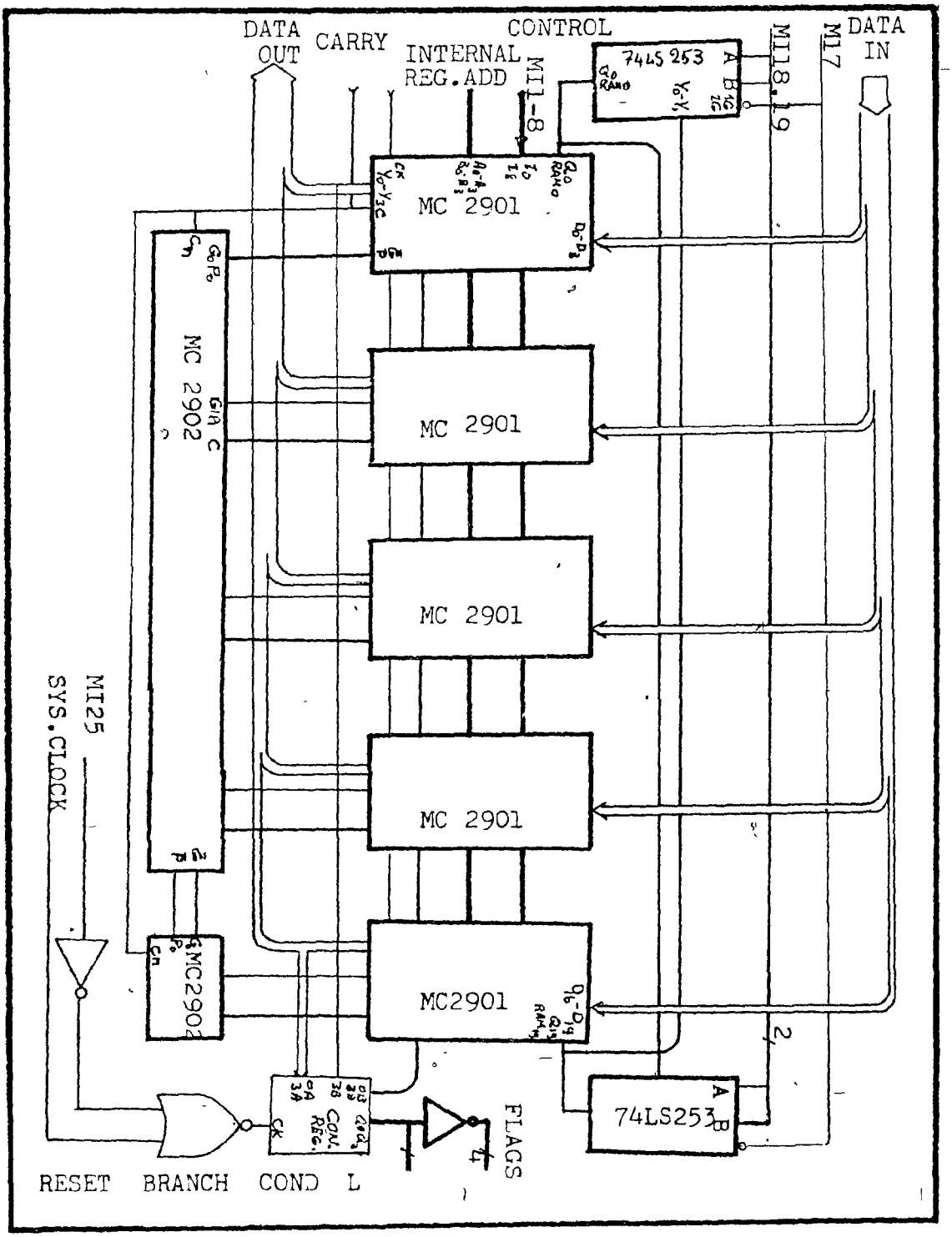


Figure 7-2 Central Processor (CPU)

Two MC2902, carry look ahead, chips are used to provide the carry generation and propagation network between the processors. Each 2902 gets its inputs from the 'G' (carry generator), the 'P' (Carry propagator) and the 'CN' carry outputs of the 2901 and outputs its signal from 'CN+X'. All five processor chips are clocked synchronously. The carry presented to the next chip is not corrected immediately after the clock pulse, but setteles after a delay and the correction ripples through to the last slice.

The control inputs of these five chips as well as the registers addresses 'A' and 'B' are all paralleled. For this reason all five chips react as a single 20 bit processor.

The 2901 contains 17 internal registers which are similar in function and purpose. The 2901 microprocessor has the capability of shifting or rotating the data which is to be stored in any of its registers. These special functions can be accomplished by presenting different combination of signals at the pins labelled 'Q0', 'Q3', 'RAM0' and 'RAM3' of the 2901. In order to have a flexible system, two multiplexors (74253) are used to select different paths between these pins. The selection of the multiplexor paths is done under software control through microcode instruction bits 18 and 19 (MI 18, MI 19).

The status register (25LS09) is used to save the flags

generated by the processor. The clocking of this register can be enabled or disabled by bit 25 of the microinstruction. The outputs of this status register can be read into and subsequently restored from the 2901 microprocessor.

The other microinstruction bits which are relevant to the ALU section of the microprocessor module are:

- Bits 0-2 are used to select the origin of the ALU source operands.

- Bits 3-5 are used to select the ALU function to be performed.

- Bits 6-7 are used to select the destination of the result.

- Bit 8 is used to shift the stored result of an operation.

- Bits 9-12 are used to select the 'A' address to the 2901.

- Bits 13-16 are used to select the 'B' address to the 2901.

Bits 6,7 and 8 are controlled by bits 17 or 25 of the microinstruction (Figure 7-3). If one of these bits is low, then 'NOP' becomes high, generating a high, low and low in lines I6, I7 and I8 respectively. This code means "do nothing" for the microprocessor. This is useful during the conditional branch (bit 25 is low) status update (bit 17 L) which will be explained later.

SEQUENCER AND BRANCH CONTROL

A very important part of the microcomputer is the sequencer and branch control. By carefully designing this part, the execution time of the system has been decreased. This part can be divided into three main sub-modules, each module assigned to a specific function. These modules are : Sequencer, conditional branch controller and tracer.

SEQUENCER

The sequencer consists of three 2909 chips, paralleled in order to obtain a microcode addressing space of 4K words (Figure 7-4). This means that most, 12 bits are needed to address any microinstruction in the Microcode Memory. Since only 1K words of Microcode Memory are available at present, only 10 of the 12 sequencer bits are used.

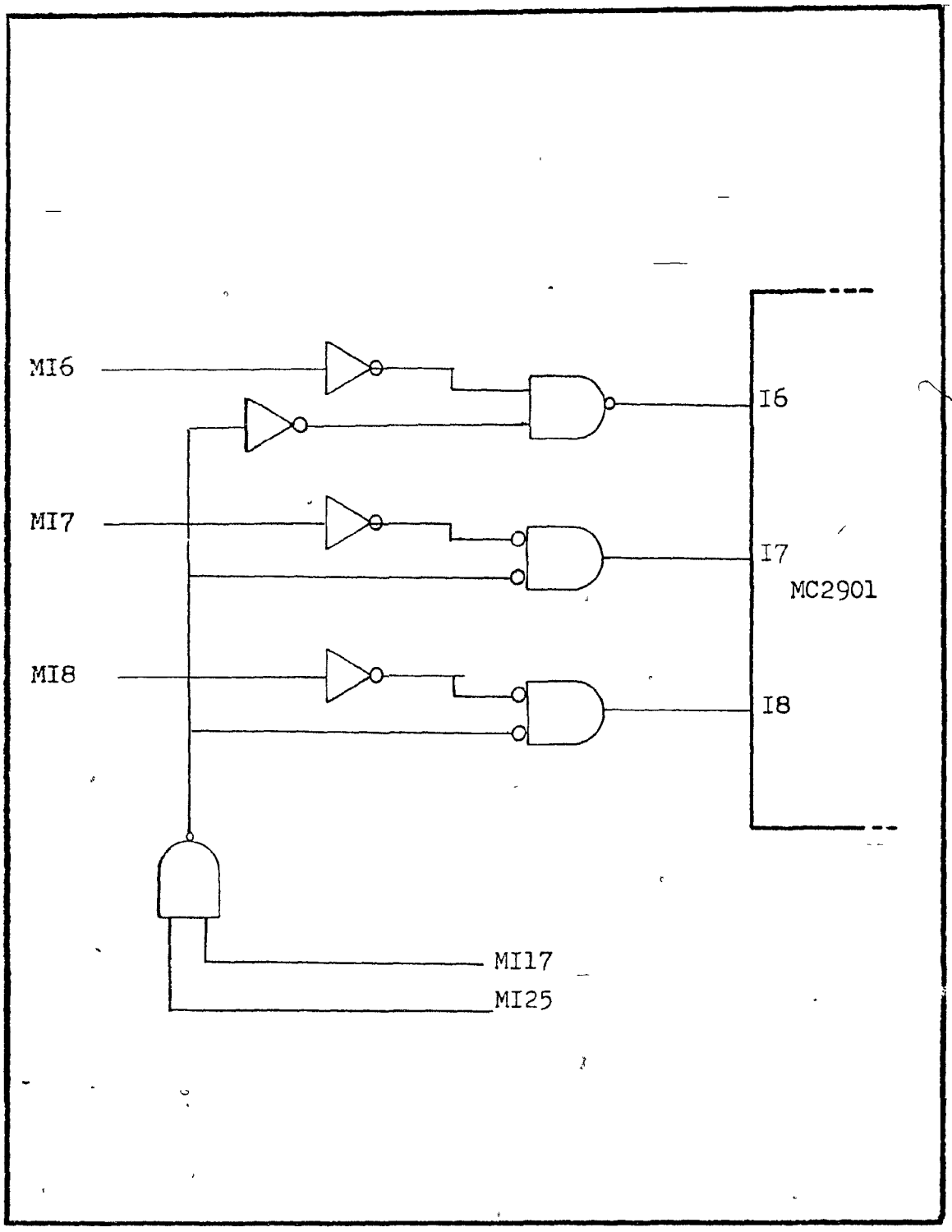


Figure 7-3 NOP circuit

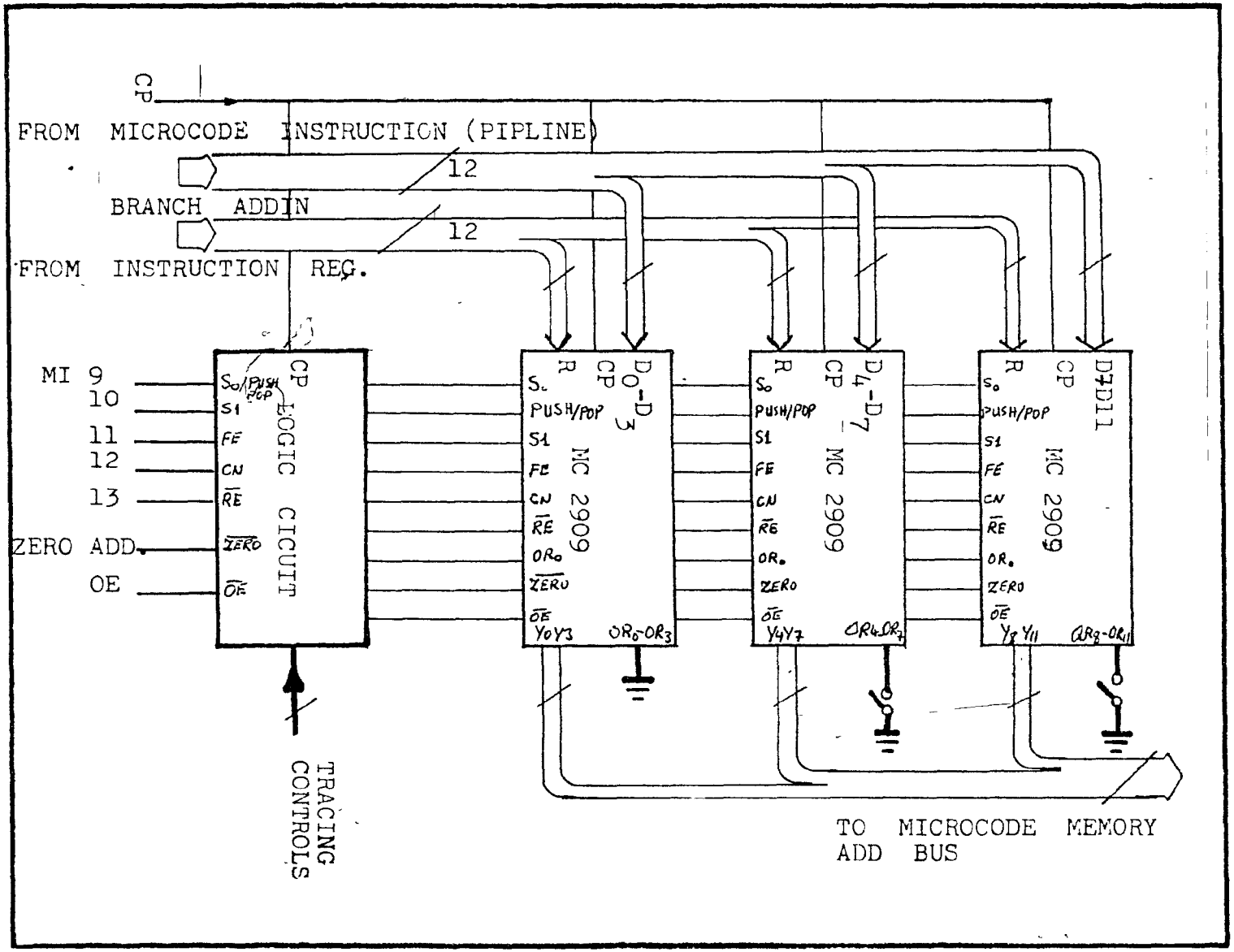


Figure 7-4 Sequencer

The sequencer achieves

- 12 bits of direct address input(D0-D11).
- 12 bits of internal register address input(R0-R11)
- stack of four 12 bit words.

The sequencer has the following capabilities for branching:

- The microprogramme can branch to any address by presenting that address on the 'D' inputs. - The twelve microcode address bits are located at bits MI 0-MI 8,MI 14, MI 15 and MI 16 in the microinstruction word.

- During execution of graphics programmes, the macroinstructions are loaded from the Local Memory into the Macroinstruction Register (MIR) which in turn feeds the 'R' inputs of the sequencer, for branching to the appropriate microcoded routines. The address register is loaded when the 'RE' input is enabled. The address register can be selected using bits MI 9 - MI 13 of the microinstruction.

- The sequencer is able to stack up to four microinstruction adresses on its internal stack, so that the programme can have subroutine capabilities (care must be taken since no more than four addresses are pushed into the stack at any one time).

A combination of bits MI 9, MI 10 and MI 11 are used to push and pop addresses.

The rest of the sequence operation is explained as follows:

- Setting microinstruction bit 13 increments the microcode programme address counter for the programme to continue sequentially. On the other hand, if this bit is lowa test and branch instruction can be repeated untill the test satisfied.

- The 'zero' input provided on the 2909 chip clears the microprogramme counter to zero. This input can be controlled from the host copmputer through the control register or manually using a push button.

CONDITIONAL BRANCH CIRCUIT

This consists of a 16 input selector (74150) and five two input nor gates (Figure 7-5). Bits MI 9-13 are used to programme the branches, are passed individually through these five nor gates. The second inputs of these five gates are driven by the output of the selector. Bits MI 21-24 are connected to the A,B,C and D select codes by which the branch condition can be chosen. Anytime the branch condition is met, the output of the selector goes low which unblocks the nor gates and allows the appropriate branch to occur.

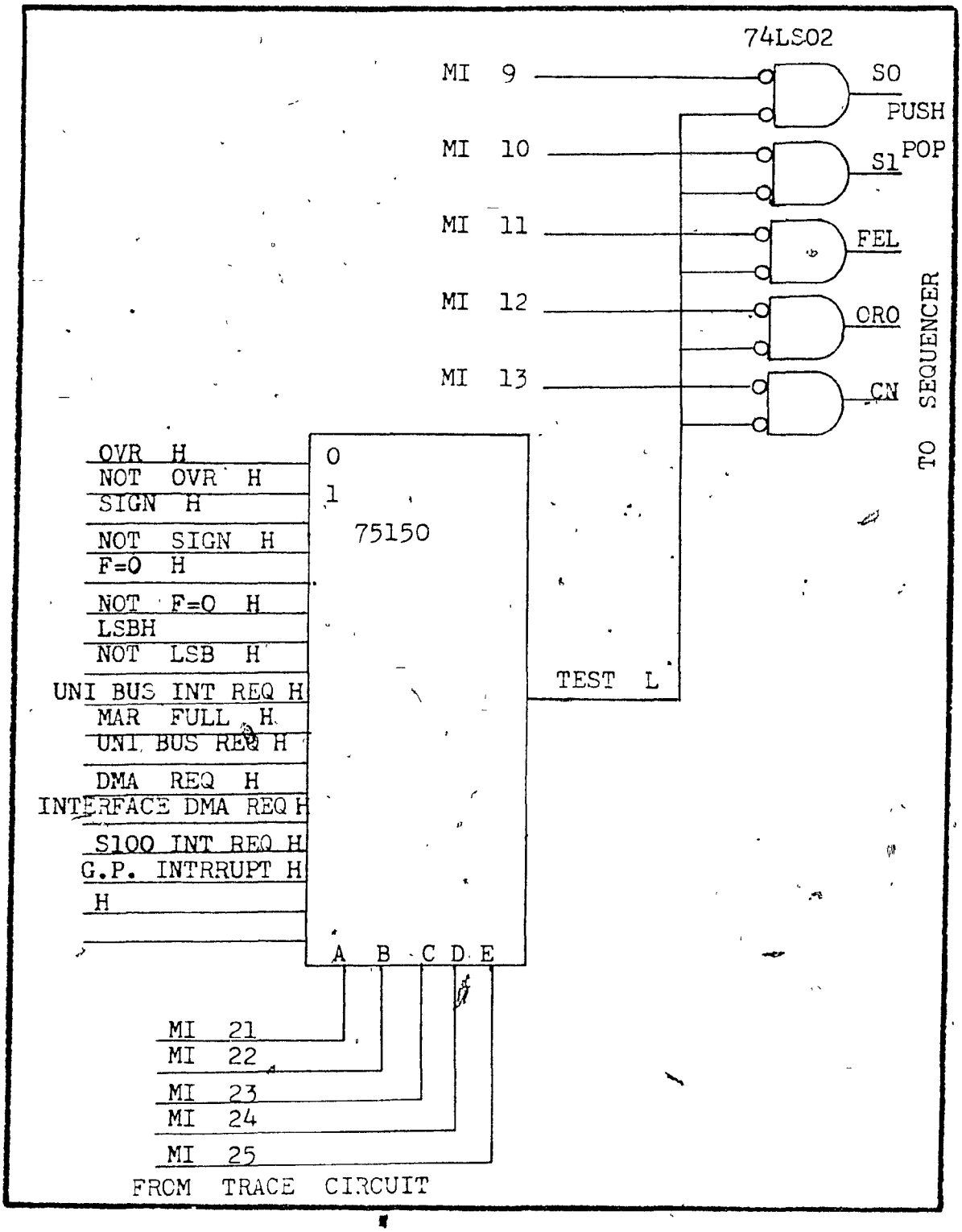


Figure 7-5 Conditional Branch circuit

This selector is disabled by hardware if the trace circuit is in operation. The Trace mode is explained in the next section.

TRACER

While debugging a microprogramme it may be helpful to dump the 2900 CPU registers after executing each microinstruction. This tracing facility was incorporated to help testing the combined performance of both the microcode software and the 2900 hardware system. It is implemented using microcoded subroutines for copying the 2900 system register contents into a local memory buffer after the execution of each microcode instruction. It also uses some hardware circuits to force the 2900 CPU sequencer to implement subroutine calls to the above mentioned routines. This trace function may be triggered manually or through a control signal sent from the host computer system.

The tracer consists mainly of one counter (74161), two clock synchronizers (74120), three 'D' flip flops, two dip switches and several gates (Figure 7-6).

Figure 7-7 presents the complete timing diagram of this circuit. The tracer's function can be illustrated as follow: By setting the 'TRACE L' to low, the first flip flop (TRFF) captures it and 'Q' output of this flip flop resets the 'CN' input to the sequencer which force the sequencer to repeat the same address.

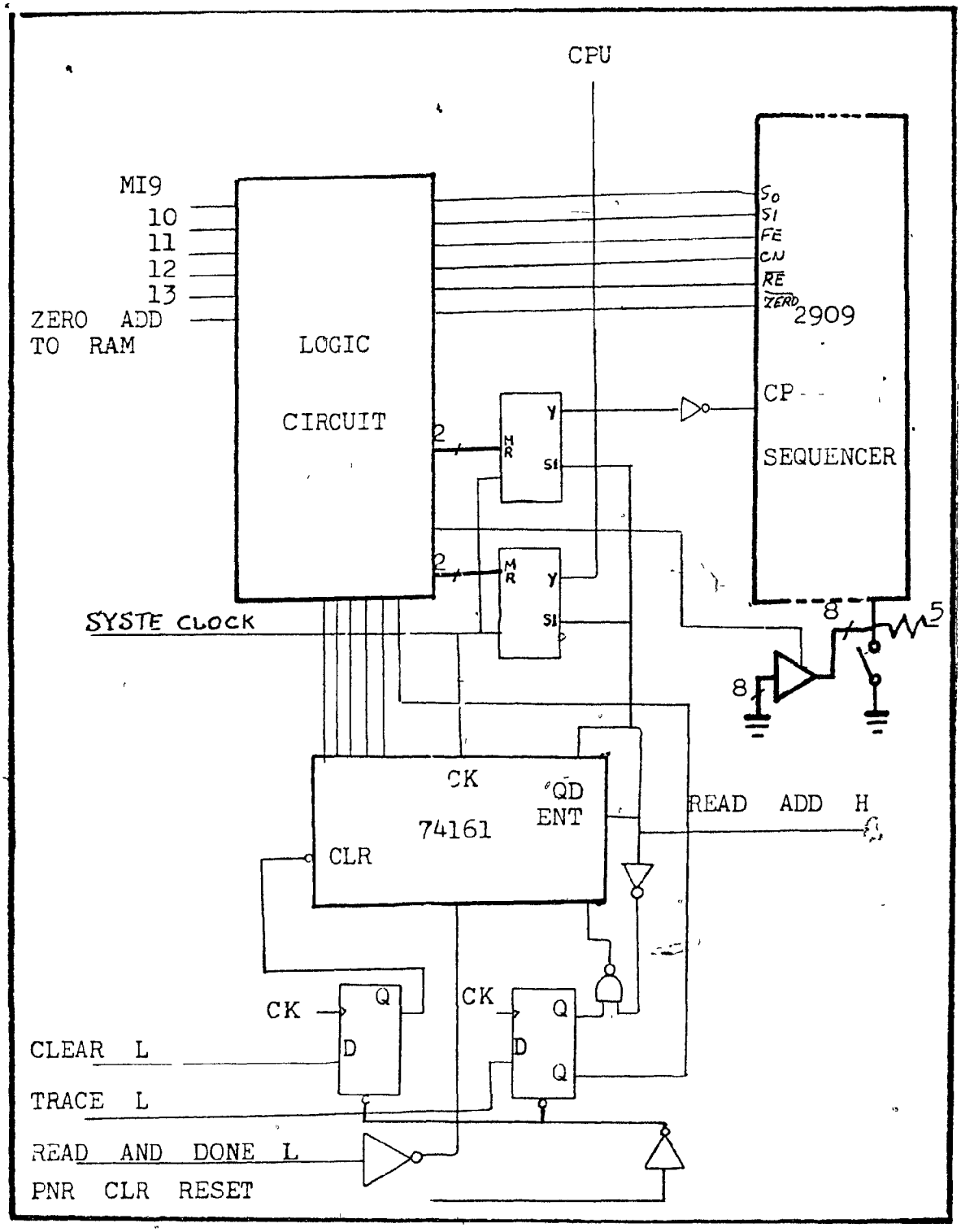


Figure 7-6 Tracer circuit

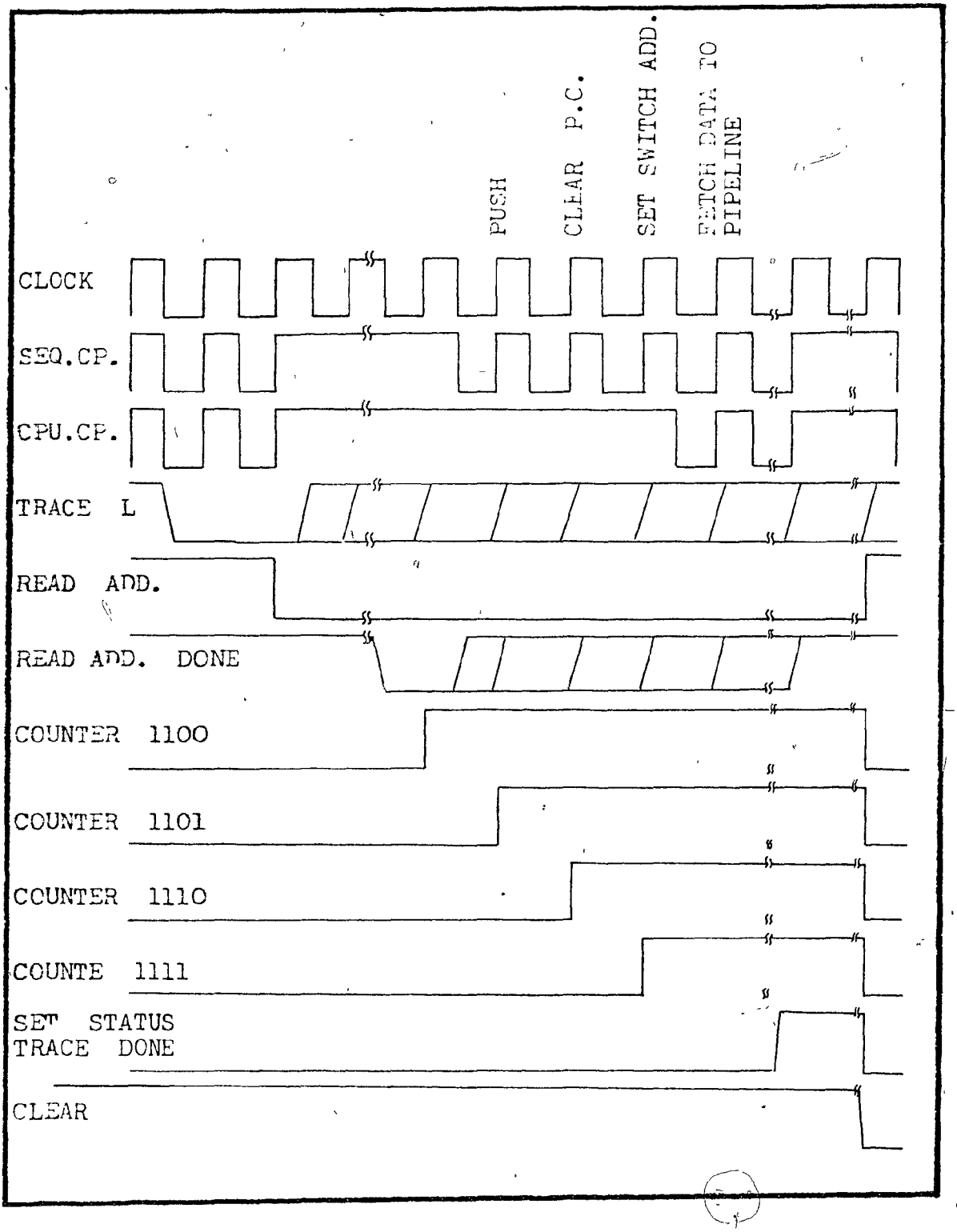


Figure 7-7 Tracer timing diagram

With the next clock pulse, the current instruction, which was loaded into the pipeline, is executed and at the same time, the most significant bit of the counter is set synchronously by the output of the first flip flop. The 'Qd' output of the counter freezes the clock pulses of the microprocessors and of the sequencer. At the same time the condition for the push is composed by the logic circuit on the sequencer control inputs. Once everything is frozen on the microcomputer, the trace requester is able to read the current address of the sequencer which is communicated to the requester by setting " READ ADD H ". Then the requester, after reading the address, may issue the " READ ADD DONE L ", which sets the 'Qc' of the counter. 'Qc' releases the sequencer clock through the clock synchronizer, and sets the 'zero' input of the sequencer to clear the microprogramme counter. By the next clock pulse the counter jumps to the counting mode and the sequencer output address becomes zero. Since the control inputs are in the push state the old address is pushed into the stack. From this state until the time the counter arrives at the "1111" state, the CPU receives only one clock pulse which causes it to execute the instruction corresponding to pushed address. In the mean time the sequencer receives three additional pulses which cause the zero initialy saved in the programme counter and the address set by the dip switches to enter from the "OR" input of the sequencer. The sequencer is set by the hardware to continue. Obviously this address is "OR"ed with the programme counter address

which was cleared, and is saved in the sequencer programme counter. In the last clock pulse, the instruction corresponding to this switched address, and it is forced by the tracer circuit into the sequencer, is fetched into the pipeline. After this, everything returns to normal except that the system executes the trace subroutine which starts at the address set by the dip switches. At the end of this subroutine, one statement sets the "status 3" high. The Host Computer scans looking for "Status 3" high which means that all the register's contents have been transferred to the local memory. The CPU's and the sequencer's clocks, are frozen by "status 3" and the 2900 CPU waits. The Host Computer then reads all registers from the Local Memory and at the end issues a " CLEAR L "which resets all the flip flops and the counter of the tracer circuit. However, the programme counter still contains the end address of the trace subroutine. At this stage a return statement causes the stack address to be popped which restores the return address to the main programme. Therefore, care must be taken to include the " set status 3 " and " Return " statement at the end of the trace subroutine.

The 8 most significants of the starting address for the trace subroutine can be chosen using the eight DIP switches, the four least significant bits are fixed to zero.

Keeping " TRACE L " low, all the time, would cause

execution of one instruction followed by the execution of the trace routine. This repeats as long as the " TRAGE L " is not removed.

INTERNAL REGISTER, EXTERNAL REGISTER AND MULTIPLIER

There are 17 internal registers available on the 2901 ALU. One of these is referenced as the 'Q' register. The others, are accessed through the 'A' and 'B' address ports, and are called "RAM" registers. Our microcomputer containes, in addition to the above internal registers, a number of external registers, some of them used as general purpose registers while others are dedicated to a particular task. The external registers and their functions are explained in the following sections (Figure 7-8).

GENERAL PURPOSE REGISTERS

A set of sixteen 20 bit word registers, built up from five AM27S03 chips are used as general purpose registers. A careful study of the MC2901 tables shows that one of the two ALU operands has to come either from an internal or the 'Q' register. For this reason, only one operand in an ALU Transaction can come from an external register, and since there is only one set of direct data inputs on the MC2901, Therefore the 'A' and 'b' address fields are sufficient to address the internal and external registers. In the processor design, address field 'B' has been used to select one of the

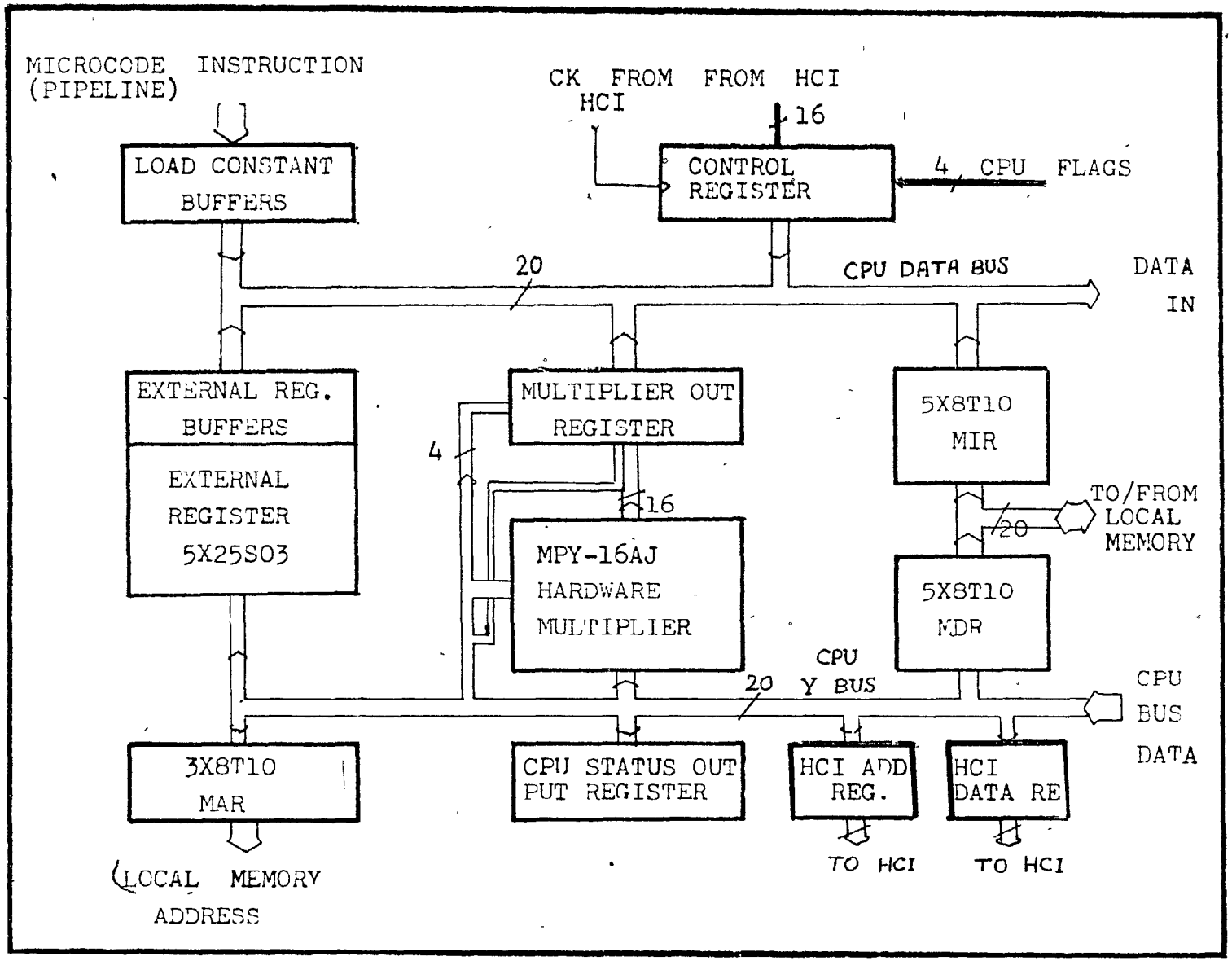


Figure 7-8 Internal Reg. External Reg and Multiplier

external registers. Bits MI 13-16 define this 'B' field while bit MI 26 is used to " write enable " the external registers and bit MI 27 is used for the chip selection of the external registers. The reasons favouring use of the 'B' addres are :

- The availablity of ALU functions associated with the direct inputs require the 'A' field to specify internal registers.

- The only way that the external registers can participate in any ALU function is through the direct inputs. So if the 'B' group of address lines are used to select the external register then any combination of one internal and one external register can be processed.

The input of these external registers are connected directely to the 2901 output Y BUS but their outputs are passed through the tristate buffers and connected to the " DATA IN " (Microprocessor input), in order to tristate them off when ever they are not required.

SPECIAL PURPOSE REGISTERS

Three other registers called MAR (Memory Address Register), MDR (Memory Data Register) and MIR (Memory Instruction Register) are used to directly access the Local Memory.

Since the MAR is only twelve bits long, it is connected to the twelve least significant lines of the microprocessor " Y BUS ". It is used to address any location in Local Memory.

The MDR also gets its inputs from the " Y BUS ". It contains the data that is to be stored in the location addressed by the MAR.

The MIR is used to hold the instructions read from the Local Memory location addressed by the MAR and presented to the DATA IN lines of the 2901 CPU, for executing the corresponding microcoded routines.

MULTIPLIER

Since the MC290.1 can not perform hardware multiplication, a multiplier, the MPY-16A, is included in the system. To perform multiplication, the two operands are first stored in the 'X' and 'Y' registers, which are included in the multiplier. The 16 LSB bits and 4 of the 16 MSB of the product are then stored in the MUltiplier Register (MUR).

- The CONTROL REGISTER, The function of this register is

to store the different controls coming from the inside as well as outside the microprocessor. Sixteen least significant bits are received from the host computer and the four most significant bits are received in order to save the four flags of the 2901 microprocessor (overflow, sign, F0, F19). Table 7-1 shows the bit assignment of the CONTROL REGISTER.

- CPU STATUS OUTPUT REGISTER, this register is used by the host computer system in synchronizing the operation of the microprocessor array. The function of these bits are presented in Table 7-2

- HCI DATA AND ADDRESS REGISTERS, The DMA operation to be performed by the HCI systems can also be engaged by the 2900 microprocessor system. This involves loading the five " HCI MACHINE REGISTERs" shown in Table 7-3 . Each register is loaded by specifying the "HCI MACHINE REGISTER" to be loaded in the "HCI ADDRESS REGISTER" and writing the desired content into the "HCI DATA REGISTER".

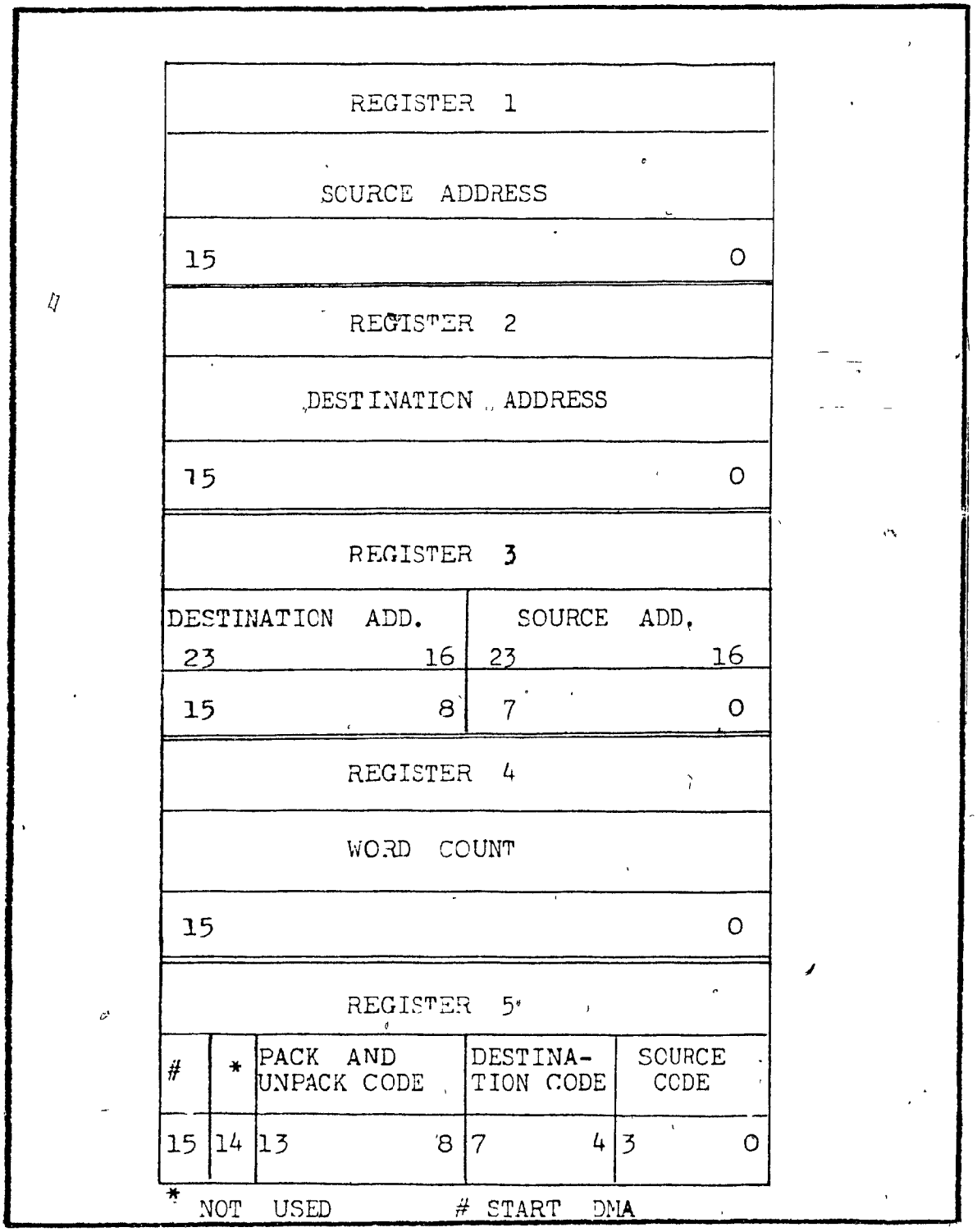
- THE LOAD CONSTANT BUFFER, this is used to allow microprogrammer to load arbitrary constant into 2901 cpu through the "DATA IN" lines as defined in the microcode instruction field.

BIT	NAME OF THE SIGNAL
0	Microbus Error Intr.(GRAPH. CONT.)
1	Microbus DMA Done INTR.(GRAPH. CONT.)
2	HCI DMA Done (HCI)
3	Unibus Intr. (HCI)
4	S100 INTR. (HCI)
5	HCI Machine Grant INTR. (HCI)
6	Reset System
7	Reset
8	NMI (HCI)
9	
10	
11	
12	
13	
14	Input Buffer 1 Full (HCI)
15	Input Buffer 2 Full (HCI)

Table 7-1 · CONTROL REGISTER

BIT	NAME OF THE SIGNAL
0	Not used
1	Interrupt to Unibus
2	HCI Machine Request
3	Not used
4	Interrupt to S100
5	
6	
7	
8	
9	
10	
11	
12	
13	Microbus Error
14	Input Buffer 1 Acknowledge
15	Input Buffer 2 Acknowledge

Table 7-2CPU Status Register



REGISTER 1	
SOURCE ADDRESS	
15	0

REGISTER 2	
DESTINATION ADDRESS	
15	0

REGISTER 3	
DESTINATION ADD.	SOURCE ADD.
23 16	23 16
15 8	7 0

REGISTER 4	
WORD COUNT	
15	0

REGISTER 5				
#	*	PACK AND UNPACK CODE	DESTINATION CODE	SOURCE CODE
15	14	13 8	7 4	3 0

* NOT USED # START DMA

Table 7-3 HCI Machine register

The commmunication of data through the above registers and busses is controlled through microcode software and "REGISTER CONTROL " hardware which will be discussed in the next section.

REGISTER CONTROLS

This section describes most of the controls applied to each individual element on the CPU card. The design of such controls is difficult, since it should not lead to any hardware conflicts for all possible combinations of the microwords.

These controls are presented in two major parts according to the engaged functions being internal or external to the 2900 microprocessor module.

INTERNAL FUNCTION CONTROLS

All these control registers are accessible by software through the bits MI 13-16,MI 21-25 and MI 28 of the Microword. Except for bits MI 13-16, which are used to address the 16 external general purpose registers, the rest are assigned to control their inputs, outputs and write enables. Table 7-4 shows the selection and figure 7-9 gives the synthesis of the "INTERNAL FUNCTION CONTROLS".

MI 24=Low MI 25=High MI 28=Low			
MI 21	MI 22	MI 23	Action
0	0	0	MDR ENABLE L
0	0	1	MAR ENABLE L
0	1	0	CLOCK FLAG REG L
0	1	1	READ FLAG REG L
1	0	0	CLK Y REG OF MULT
1	0	1	CLK X REG OF MULT
1	1	0	CLK MULT OUT REG L
1	1	1	ENABLE MULTI OUT REG L

TABLE 7-4 INTERNAL FUNCTION CONTROL SELECTION

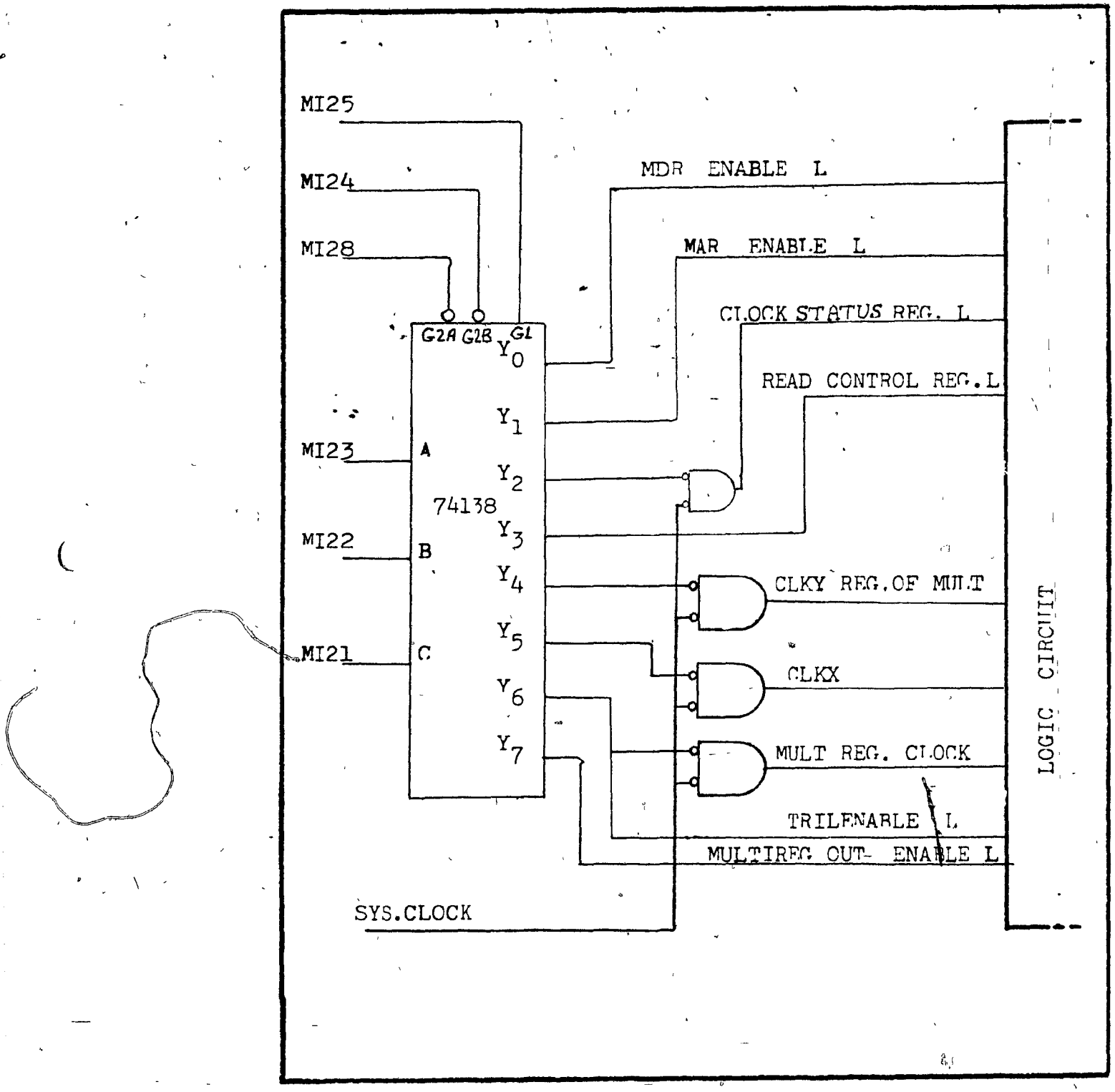


Figure 7-9 Internal Function Control circuit

The three-to-eight line decoder (74S138) used in this circuit, guarantees an absence of bus contention between these registers.

EXTERNAL FUNCTION CONTROLS

One 74S138 decoder is used to control the registers associated with the HCI. These controls are programmable through bits MI 21-23,MI 25,MI 28 and MI 29. Table 7-5 shows the selection and Figure 7-10 presents the logic circuit. "HCI DATA AND ADDRESS REGISTER" are enabled upon receiving the HCI DMA grant. This permits to define DMA transactions parameters which to be carried out by the HCI.

The MC2901 microprocessor has no interrupt inputs. So, as such, it can not be interrupted. However, the programme can search for the status and interrupt the main programme in case of such a request. This is done by the 16 input selector which has been already discussed in the previous sections. The input "E14" of this selector represent the general purpose interrupt input. This interrupt scheme is shown in greater details in figure 7-11. As it is shown, several status bits are "OR"ed together to compose the general purpose inter . One three input multiplexor is also used to programme a part of the status. Table 7-6 gives the selections. Three JK flip flops are assigned to latch overflow, Micro Bus

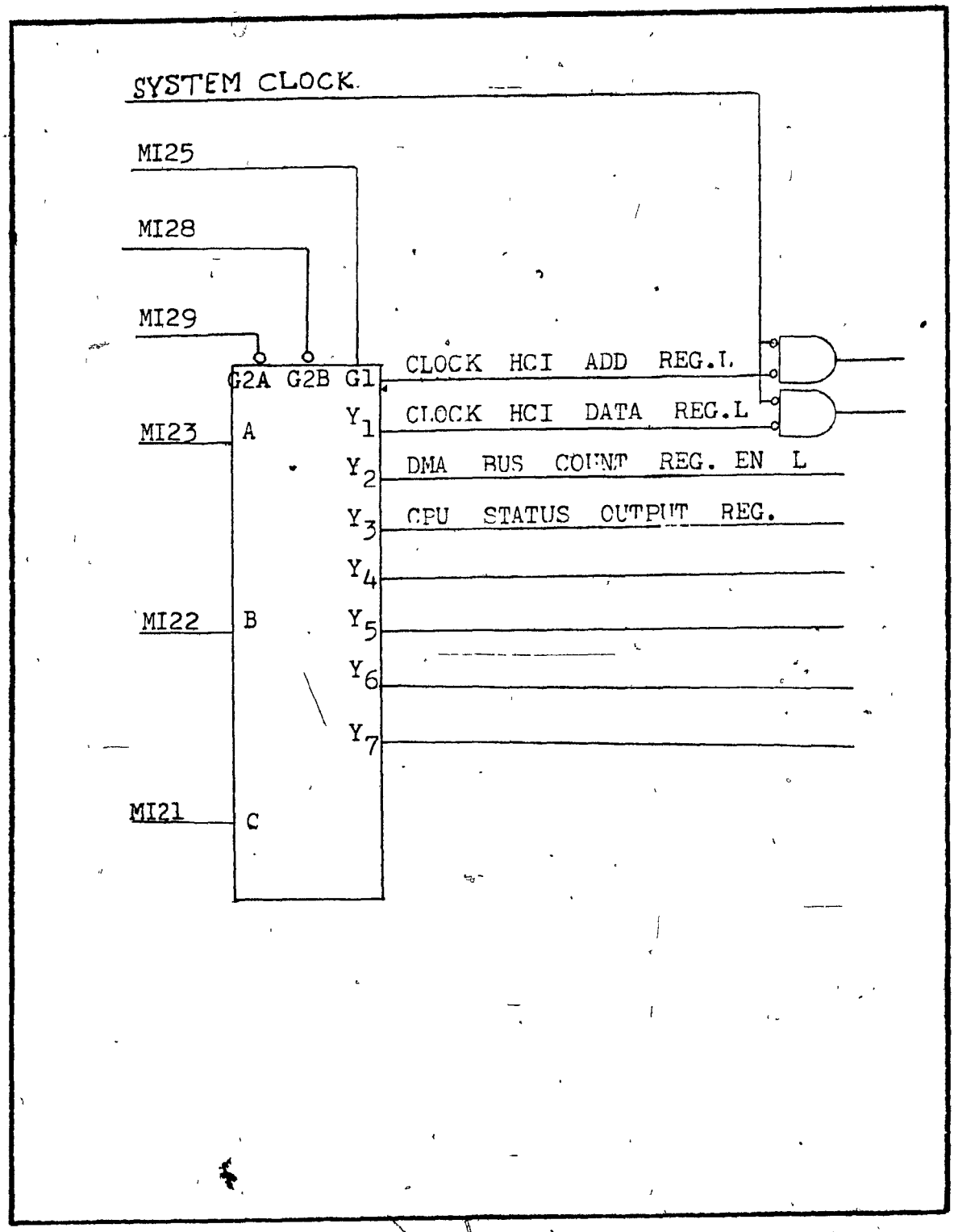


Figure 7-10 External Function Control circuit

MI 25=High MI 28=Low MI 29=Low			
MI 21	MI 22	MI 23	Action
0	0	0	DMA BUS SOURCE ADD REG EN L
0	0	1	DMA BUS DEST ADD REG EN L
0	1	0	DMA BUS COUNT REG ENABLE L
0	1	1	DMA BUS CONT WORD REG EN L
1	0	0	not used
1	0	1	not used
1	1	0	not used
1	1	1	not used

TABLE 7-5 EXTERNAL FUNCTION CONTROLS SELECTION

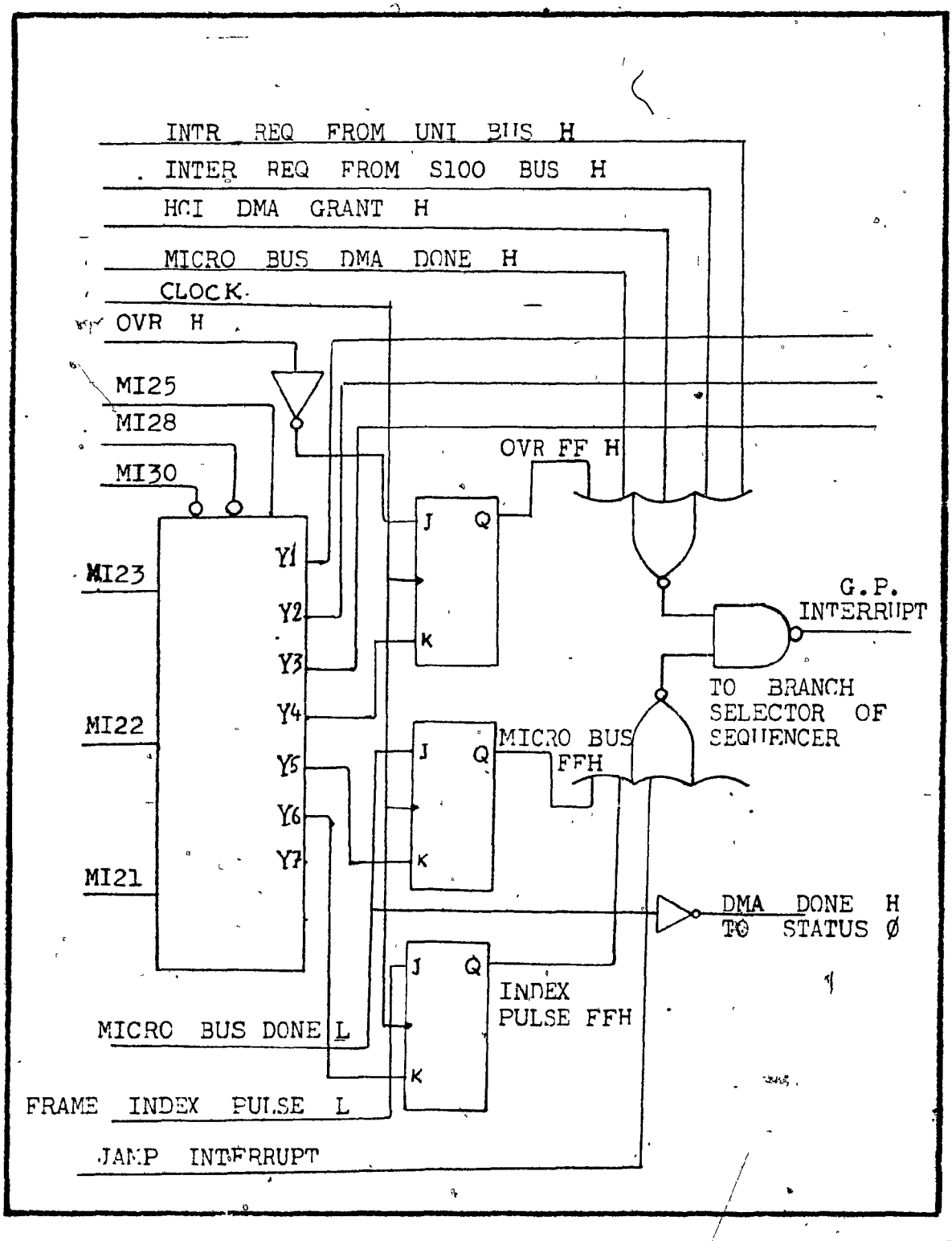


Figure 7-11 Intrrupt circuit

MI 25=HIGH MI 28=LOW MI 30=Low			
MI 21	MI 22	MI 23	Action
0	0	0	not used
0	0	1	S100 ALU INTR
0	1	0	SEQUENCER REG ENABLE L
0	1	1	CLEAR FRAME INDEX PULSE FF L
1	0	0	CLEAR MICRO BUS DONE FF L
1	0	1	CLEAR OVER FF L

TABLE 7-6 INTERRUPT SELECTION

Done and frame Index pulse signals. An inverted copy of the Microbus Done is also used to clear the Micro Bus DMA Request.

STATUS

Two hardware status registers are reserved for synchronizing operation with Local Memory and another is used for hardware tracing facility.

STATUS 0 Local Memory address register full

STATUS 1 Direction to Local Memory

Status 2 Tracing cycle done

Figure 7-12 shows the implementation of one sample register. The clock pulse to them can be enabled through bit MI 17 of the microword. Bit MI 16 of the microword is used, through some logic, to set or reset these registers. These registers can also be cleared through their asynchronous clear inputs. The "LOCAL MEMORY REQUEST STATUS" can be set by the microcode instruction loading the memory address register and be reset directly by the hardware memory done pulse. The "TRACE STATUS" will be set by the microcoded software after all the register contents have been copied out to the Local Memory. After retrieving and displaying the register content, the external tracing programme will

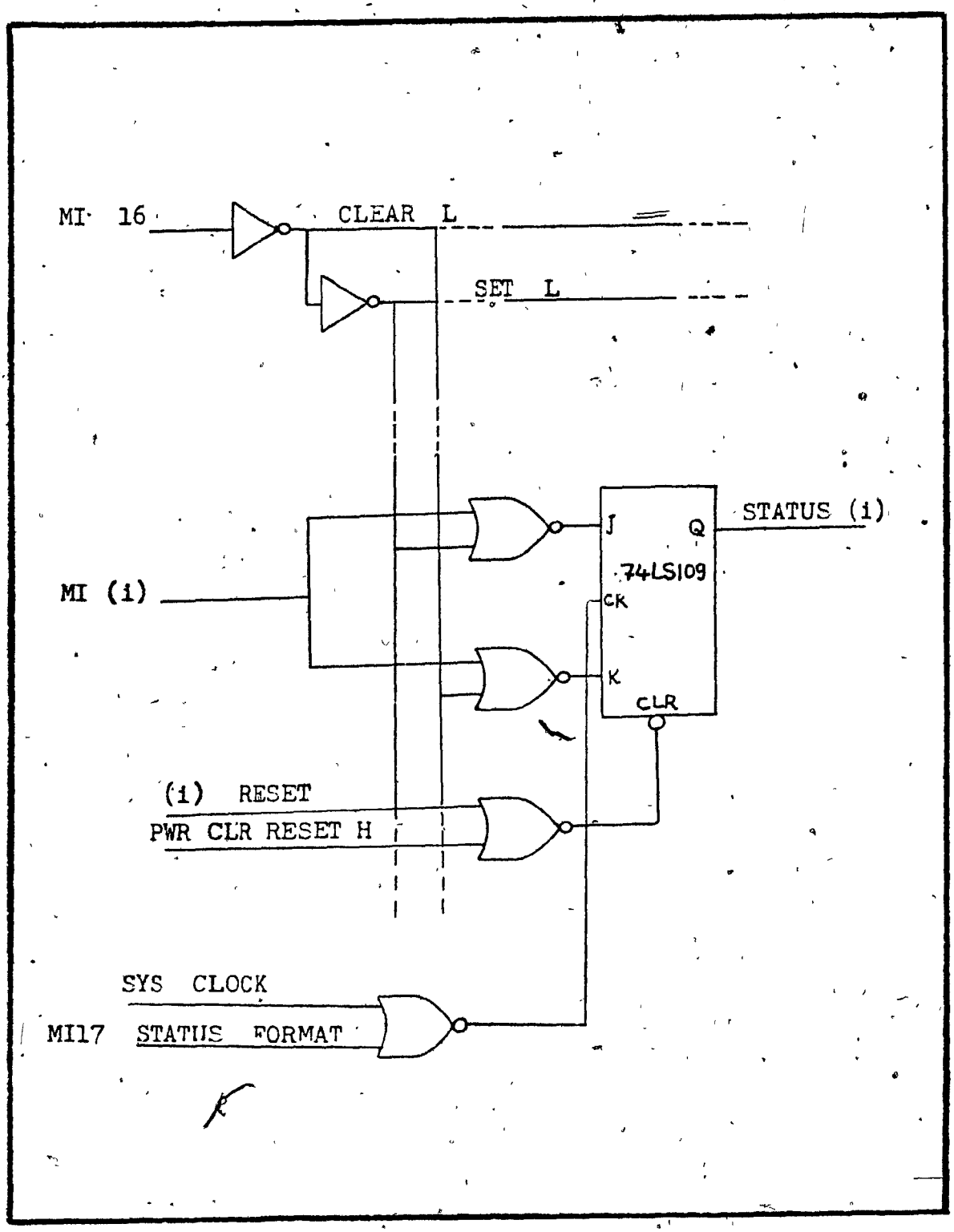


Figure 7-12 A simple of Status Register

clear the "TRACE STATUS REGISTER" to end the trace cycle and resume the execution of the microcode programme.

LOCAL MEMORY ARBITRATOR

The Local Memory of the microprocessor is connected to three different parts of the GRADS system:

- The Graphics Controller
- The Host Computer Interface
- The M2900 Microprocessor

Since the Local Memory is a shared resource, an arbitrator is needed to grant access to the Local Memory to one of the three possible parts of the GRADS. The Arbitrator design is based on a 74148 priority encoder and gives highest priority of usage to the Graphics Controler followed by the Host Computer and the CPU (Figure 7-13).

Inputs 4,2 and 1 of the 74148 are used for priority selection- with input 4 having the highest priority and input 1 the lowest priority. After granting any requests the inputs of this chip are blocked by using the 7410 nand gates. Each granted request causes a low level at the 'GS' output which starts the transmission

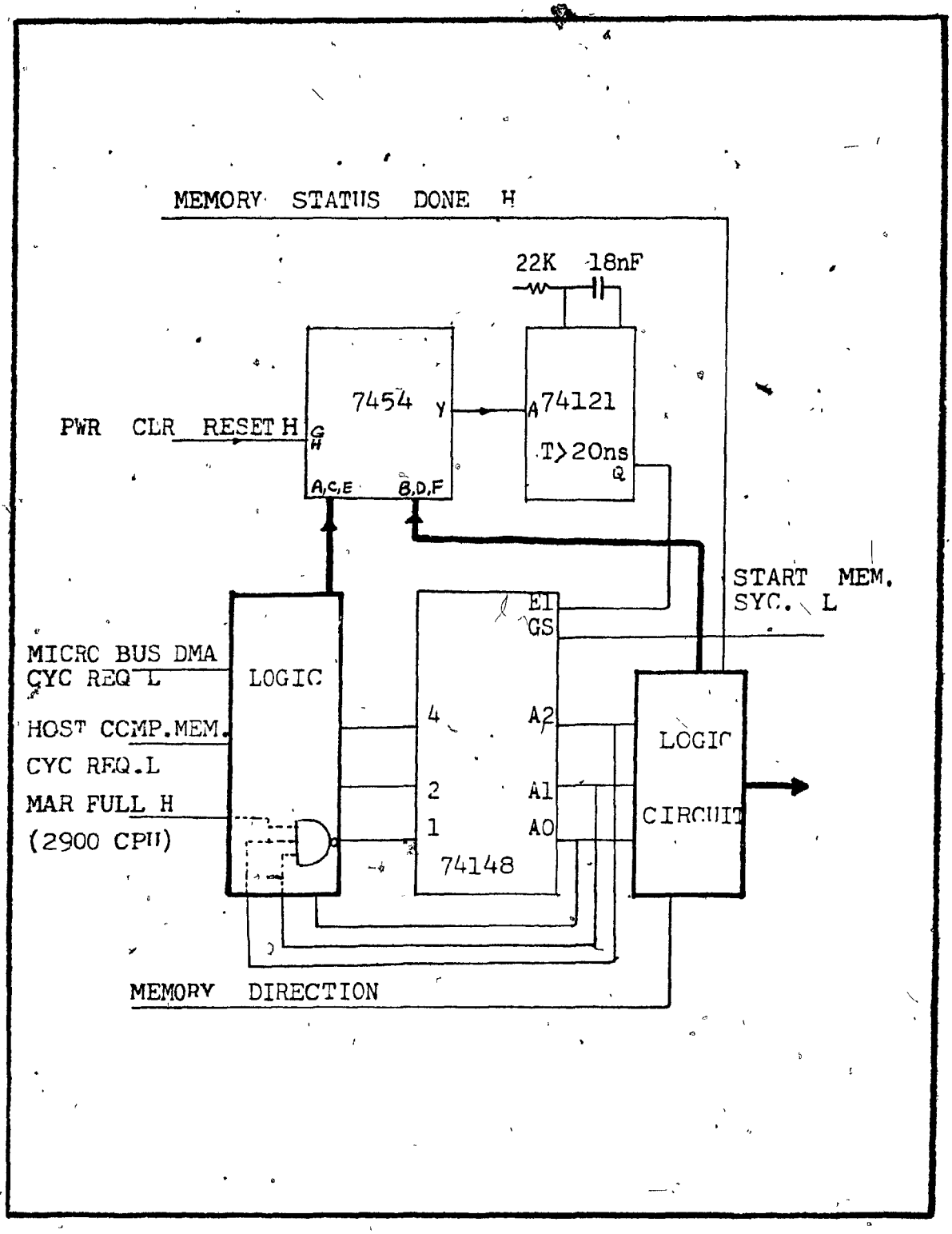


Figure 7-13 Local Memory Arbitrator

cycle in the Local Memory. At the end of this cycle the request is reset by the memory status pulse generated by the Local Memory. The removal of a request at 7454 chip triggers the input of a 74121 monostable, to apply a narrow pulse (40 ns) to the "EI" input of the 74148. During this 40ns all outputs of the 74148 go high. At the end of the pulse any pending requests at the inputs of the 74148 will generate a new grant cycle (If no requests are pending, all outputs will stay in the idle position).

The application of a manual reset to the system at any time causes, the arbitrator to reset.

PIPELINE

As mentioned earlier, one of the ways to minimize the execution of a programme time is to pipeline the programme instruction; in other words prepare the instruction ahead of time. In this design, a one word deep pipeline has been chosen. This pipeline is basically a 40 bits long register which works as follows : at every clock pulse a new address is presented to the microcode memory and after a certain delay (memory access time) the appropriate microinstruction stabilizes at the inputs of the pipeline. On the next clock pulse this instruction is executed when it is presented to at the inputs of the processor. During that same clock

period, the next microinstruction is prepared for loading into the pipeline. It can be seen that, with the pipeline, the instruction fetching time can be omitted from the instruction execution time allowing the processor to run at a higher speed. The pipeline is constructed from 7 units 74S174 chips.

MICROWORD

In this section, the Microinstruction and its bit assignments are explained briefly.

The microcode instructions can be subdivided into four formats:

- Load Constant Format
- Branch Format
- Status Call Format
- ALU Format

Any instruction is implemented using one 40 bit microcode word. However, the bits of the Microword are assigned to different functions in each of the four different formats. The format is specified using MI 17,25 and 28. Tables 7-7,8,9,10 show the detailes of the bit assignment of each format.

BIT		NAME
0	]	
1	|	ALU source operand
2	]	
3	]	
4	|	ALU function
5	]	
6	]	
7	|	ALU destination
8	]	
9	]	
10	|	
11	|	Constant field 5
12	]	
13	]	
14	|	Destination reg address. (B field)
15	|	
16	]	
17	1	
18	]	
19	]	Constant field 4
20	X	
21	]	
22	|	Constant field 3
23	]	
24	X	
25	1	
26	X	
27		Constant field 2
28	1	
29	]	
30	|	
31	|	
32	|	
33	|	
34	|	Constant field 1
35	|	
36	|	
37	|	
38	]	
39	X	

Table 7-7 LOAD CONSTANT FORMAT

BIT		NAME
0	]	
1		
2		
3		Microcode Memory
4		Page offset
5		
6		
7	]	
8	]	
9		
10		Branch address sources
11	]	
12		'OR11' input to sequencer
13		To enable micro-programme counter incrimenting (SEQ Cin)
14		Microcode Memory page address
15		(8 page of 512 words)
16		
17	1	
18	X	
19	X	
20	X	
21	]	
22		
23		16 branch condition selection
24	]	
25	0	
26	X	
27	X	
28	0	
29	X	
30	X	
31	X	
32	X	
33	X	
34	X	
35	X	
36	X	
37	X	
38	X	
39	X	

Table 7-8 BRANCH FORMAT

BIT		NAME
0		Local Memory ADD REG full
1		Direction to Local Memory
2		Tracing cycle done
3		11 status flags (not used)
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15	X	
16		Status set (Low) and clear (High)
17	0	
18	X	
19	X	
20	X	
21	X	
22	X	
23	X	
24	X	
25	1	
26	X	
27	X	
28	0	
29	X	
30	X	
31	X	
32	X	
33	X	
34	X	
35	X	
36	X	
37	X	
38	X	
39	X	

Table 7-9 STATUS-UPDATE FORMAT

BIT		NAME
0	]	
1	|	ALU source operand
2	]	
3	]	
4	|	ALU function
5	]	
6	]	
7	|	ALU destination
8	]	
9	]	
10	|	
11	|	A field address (internal reg. only)
12	]	
13	]	
14	|	
15	|	B field address internal and external reg.
16	]	
17	1	
18		
19		Shift multiplexer control
20		ALU Carry in (High)
21		
22		Selection of multiplexers controled External Registers
23		
24		To enable the Ext. Reg. Multiplexers L
25	1	
26		External reg. read (High) and write(Low).
27		To enable external reg (High) or MIR (Low)
28	0	
29		To enable the HCI Reg. multiplexer
30		To enable the Status Reg. multiplexer
31	X	
32	X	
33	X	
34	X	(x= DONT CARE)
35	X	
36	X	
37	X	
38	X	
39	X	

Table 7- 10 ALU FORMAT

CYCLE TIME OF CPU MODULE

The most important parameter in any microcomputer is the cycle time requirement of the CPU. This cycle time limits the clock rate of the system. As mentioned earlier, the GRADS system requires very high performance so it is important that all cycle should be made as short as possible. Figure 7-14 shows the timing diagram for the worst case on the M2900 Microcomputer system. the worst case cycle time is calculated to be:

(40 ns pipeline setting)+(80 ns G,P propagation)+
(38 ns carry lookaheads)+(55 ns ALU setting)+
(22 ns external shift multiplexer setting)+
(30 ns clock low pulse)= 258 ns

Therefore maximum system clock frequency is estimated to be 3.87 MHZ.

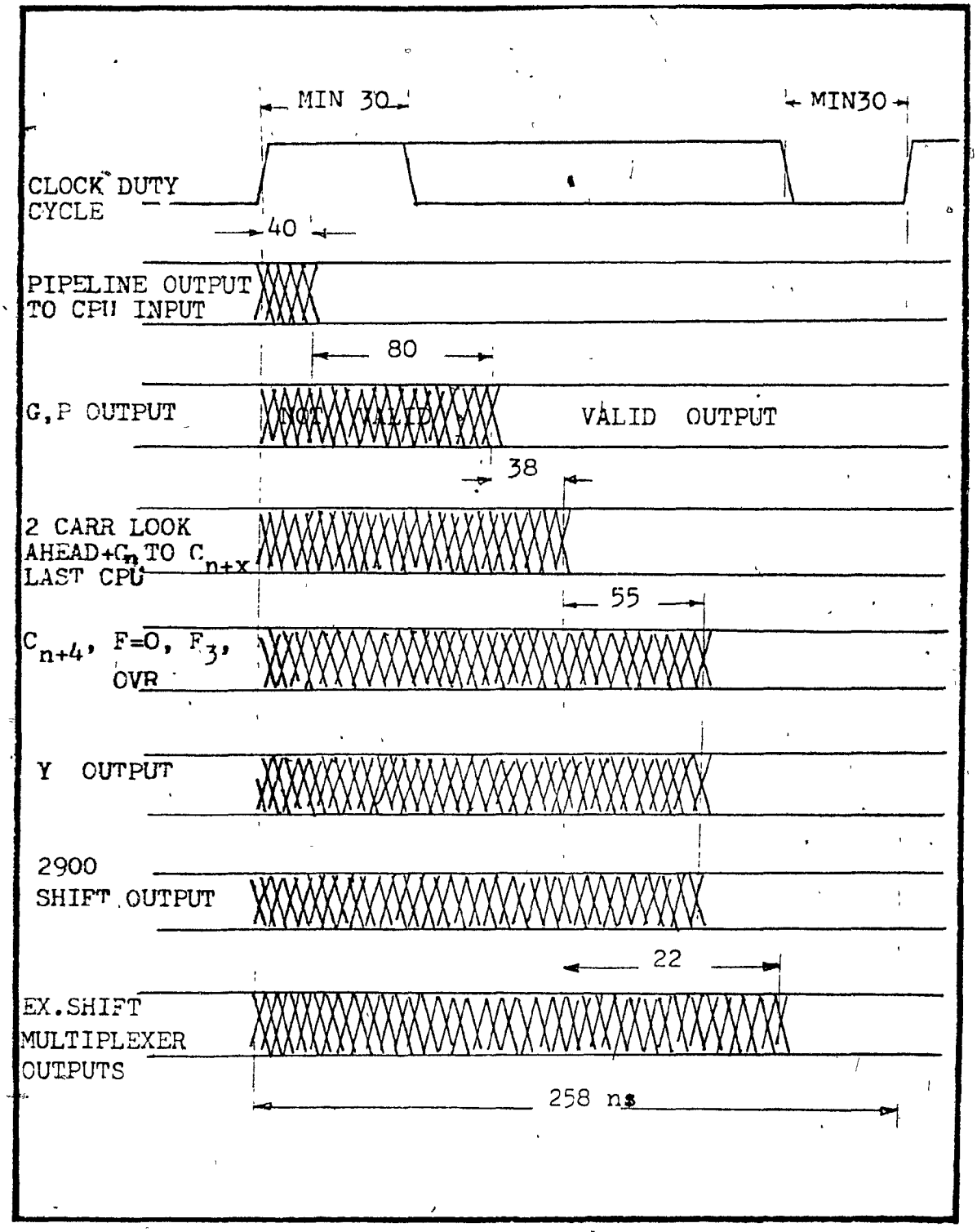


Figure 7-14 2900 Microcomputer timing diagram

This chapter has briefly reviewed the logic designe of the main subsystems of the 2900 microprocessor module. The original schematics related to this chapter are on file at the Department of Electrical Engineering, McGill University.

CHAPTER VIII

HARDWARE REALIZATION, TEST AND DEBUGING

INTRODUCTION

Traditional approaches to testing of digital systems are becoming increasingly inadequate, specially when applied to more complex circuits which consist of several LSI and VLSI components.

General purpose, Micro-computers, due to their low price and availability, are becomming an attractive alternative for the testing, debugging and validation of complex digital systems.

A Cromemco Z80 micro-computer has been used for debugging of the hardware described in this thesis. The following section discribes the test set up and the software which was used for the validation.

HARDWARE REALIZATION

The MC2900 micro-computer module has been constructed

from a total of 320 IC chips mounted on three separate boards. These three boards are, the Microcode Memory, the Local Memory, and the CPU. A total of seven card slot spaces in the GRADS rack assembly have been reserved for these three boards. Three slots are for the permanent operation and three slots are reserved for testing. The seventh one has been allocated for placing a special test board which is connected to the programmable I/O ports of the Z80 test Micro-computer.

TEST TOOLS

The following tools have been used to debug the hardware:

- Cromemco micro-computer with Z80 CPU
- 32 K bytes of RAM type memory
- Two PI08 boards
- Hazeltine 1500 CRT terminal
- Two 8" floppy disc drivers "
- 100 MHZ oscillscope
- Digital logic analyzer
- Cromemco Z80 system software including CDOS operating system, Editor, assembler, debugger,...

TEST SET UP AND PROCEDURE

Testing of complex digital circuits is not an easy task. It becomes much easier if each unit can be tested separately and then integrated into the final system. For this reason a total of four test setups have been implemented. These setups are as follows:

- Microcode Memory test setup
- Local Memory test set up
- CPU test set up
- micro computer module test set up

MICROCODE MEMORY TEST SETUP AND PROCEDURE

The Microcode Memory has been debugged with the help of the Cromemco Z80 microcomputer (Figure 8-1). By inserting the board into the test slot and plugging in the appropriate PIO8 cables, the Cromemco system can simulate reading and writing from or to the Microcode Memory, with the help of a special programme called " MICMEM ". Two buffer in the microcomputer's memory are assigned for this test. By typing "W" on the terminal a copy of the first buffer is written into the Microcode Memory. By typing the "R", the Microcode Memory contents are read back into

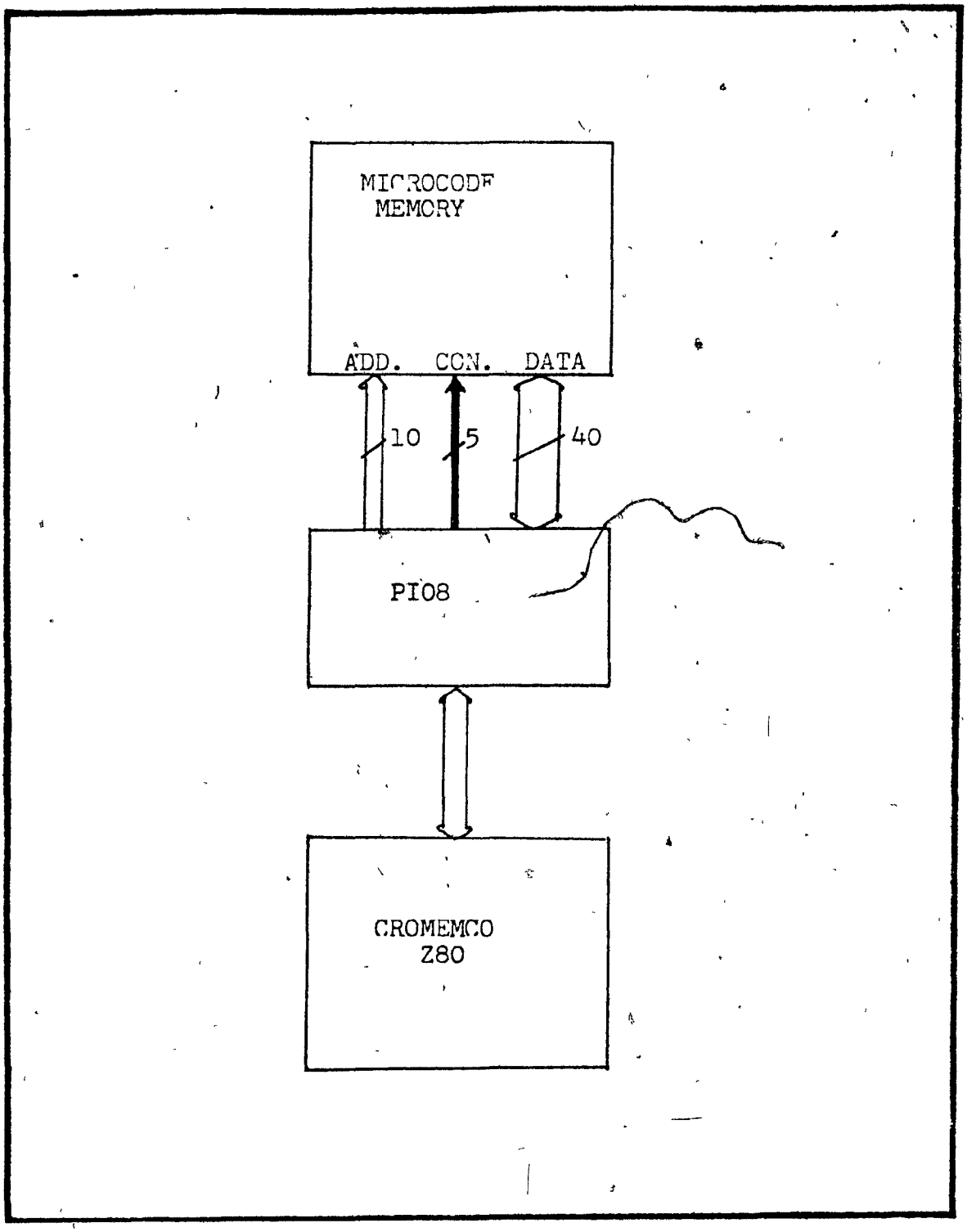


Figure 8-1 Microcode Memory test setup

the second buffer. By verifying these two buffers, (i.e. if no diferences are detected), the Microcode Memory can be assumed to function properly. In case of errors a special repeat mode of the " MICMEM " programme can be used. This can then be exercised by looping the 'W' and 'R' functions continuously repeat the memory operation at one location to permit tracing the circuit waveforms using a oscilloscope. Figure 8-2 shows the " MICMEM " programme flowchart.

THE LOCAL MEMORY TEST SETUP AND PROCEDURE

An approach similar to the testing of the Microcode Memory has been used to debug the Local Memory. All three bidirectional ports were shorted together and driven from the same lines (Figure 8-3). The corresponding test programme is called "LOCTST ". This programme ressembles the " MICMEM " programme in function and implementation. One slot in the GRADS rack is reserved for this test. This board may be tested anytime by plugging in the Local Memory at the test slot and executing the " LOCTST " programme.

CPU TEST SETUP AND PROCEDURE

To test the CPU is a relatively more complex task. For this test, the Cromemco microcomputer, two PI08 boards, an

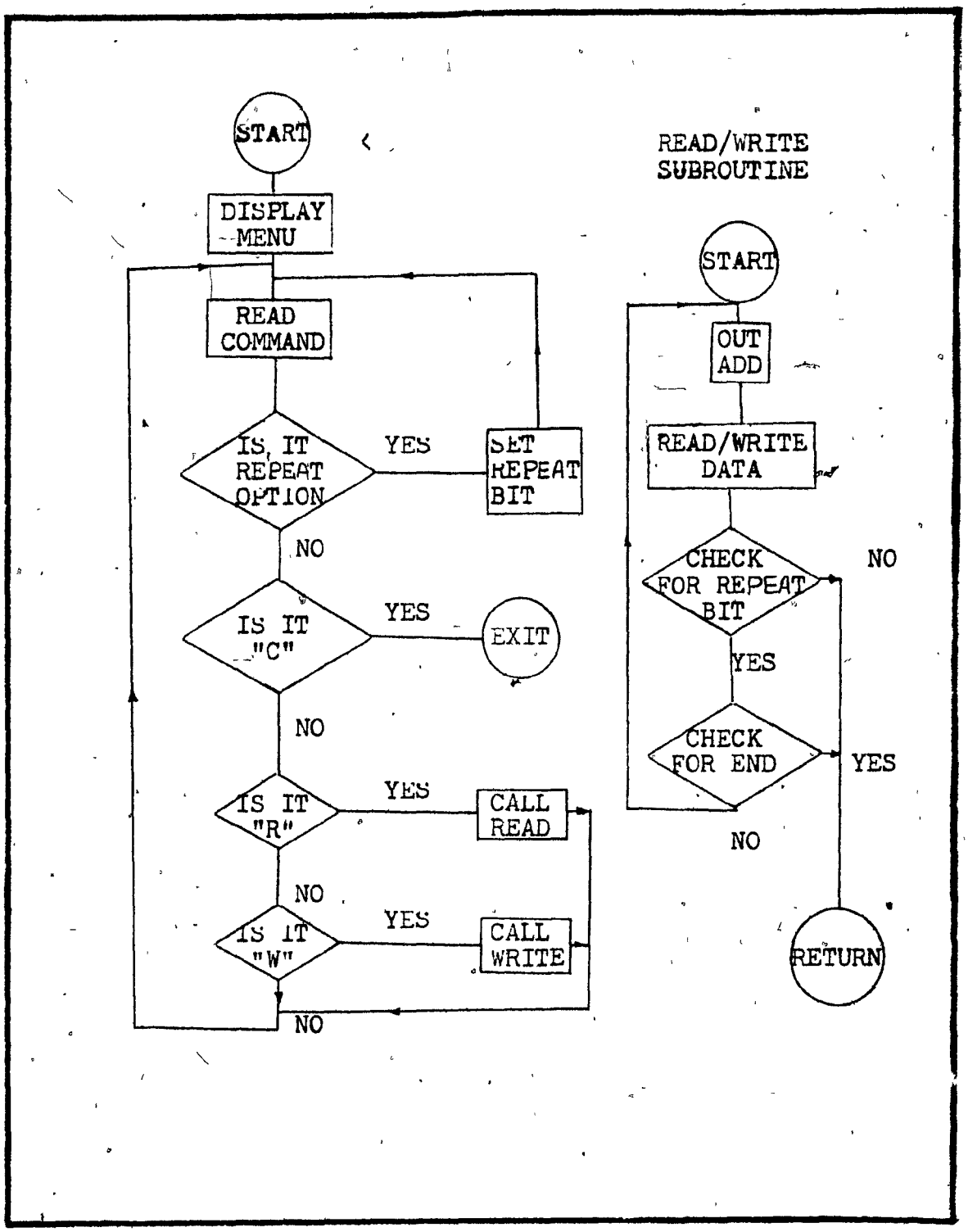


Figure 8-2 MICMEM flowchart

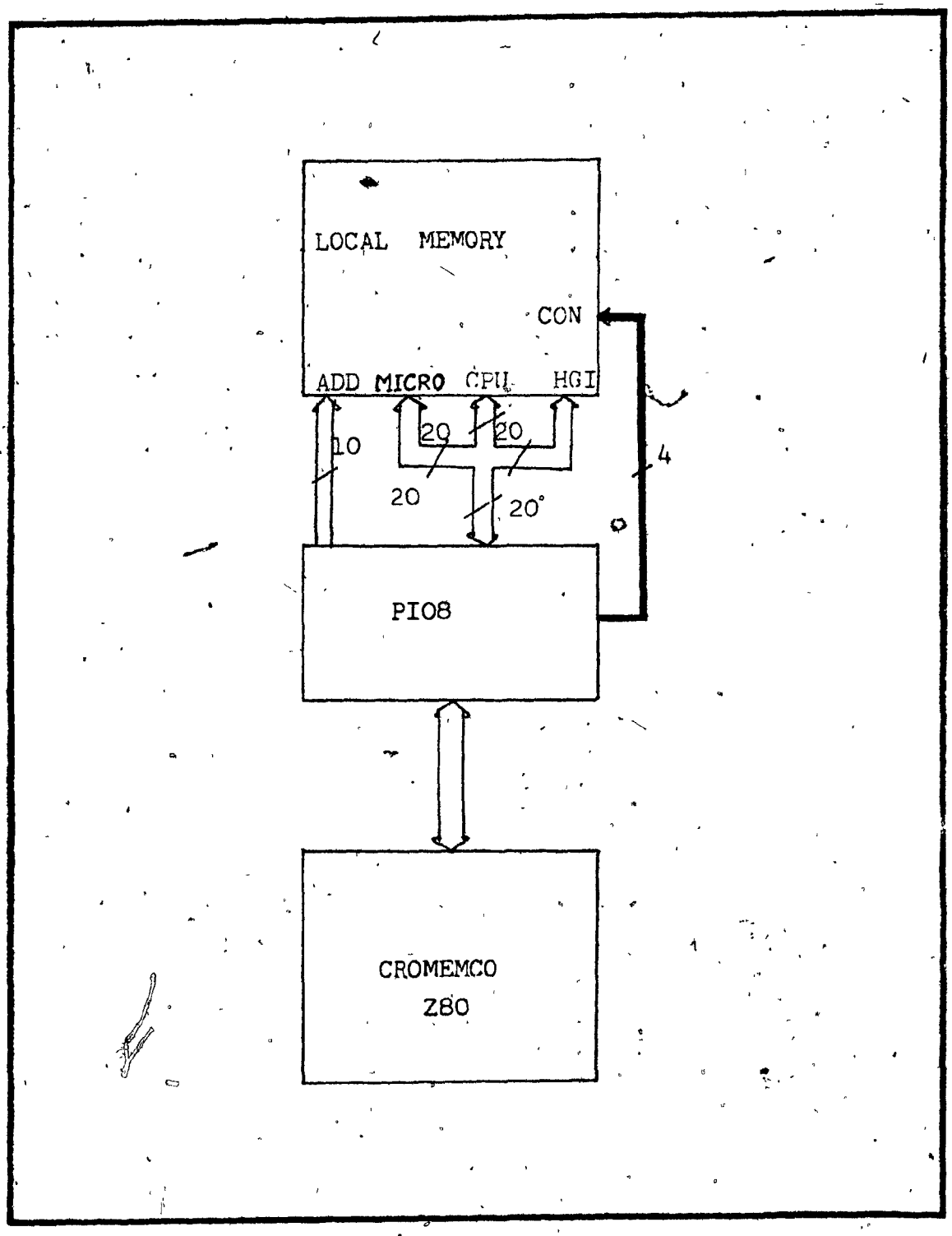


Figure 8-3 Local Memory test setup

osciloscope and the digital analyzer have to be used. The test configuration is given in figure 8-2. The programme written to test the CPU, simulates the Microcode Memory using a buffer in the Cromemco's memory. With the help of this programme, called " CPUTEST", any microwords can be saved in the simulated microcode buffer. Through interactive commands, the CPU sequencer can be set to zero or any other arbitrary address and then started to execute the microwords in a ' single step ' or 'Run' mode. The options available in 'CPUTEST' are:

M = To display a menu of commands on the CRT terminal.

L = To load microcode into buffer

R = To reset the 2900 CPU (Zero add to Ram).

G (CR) = to run the programme

Gs (CR) = to single step a programme

p = To print

T = trace

C = Return to CDOS

These commands can be explained as follows :

M: By hitting 'M' the above menu will be displayed.

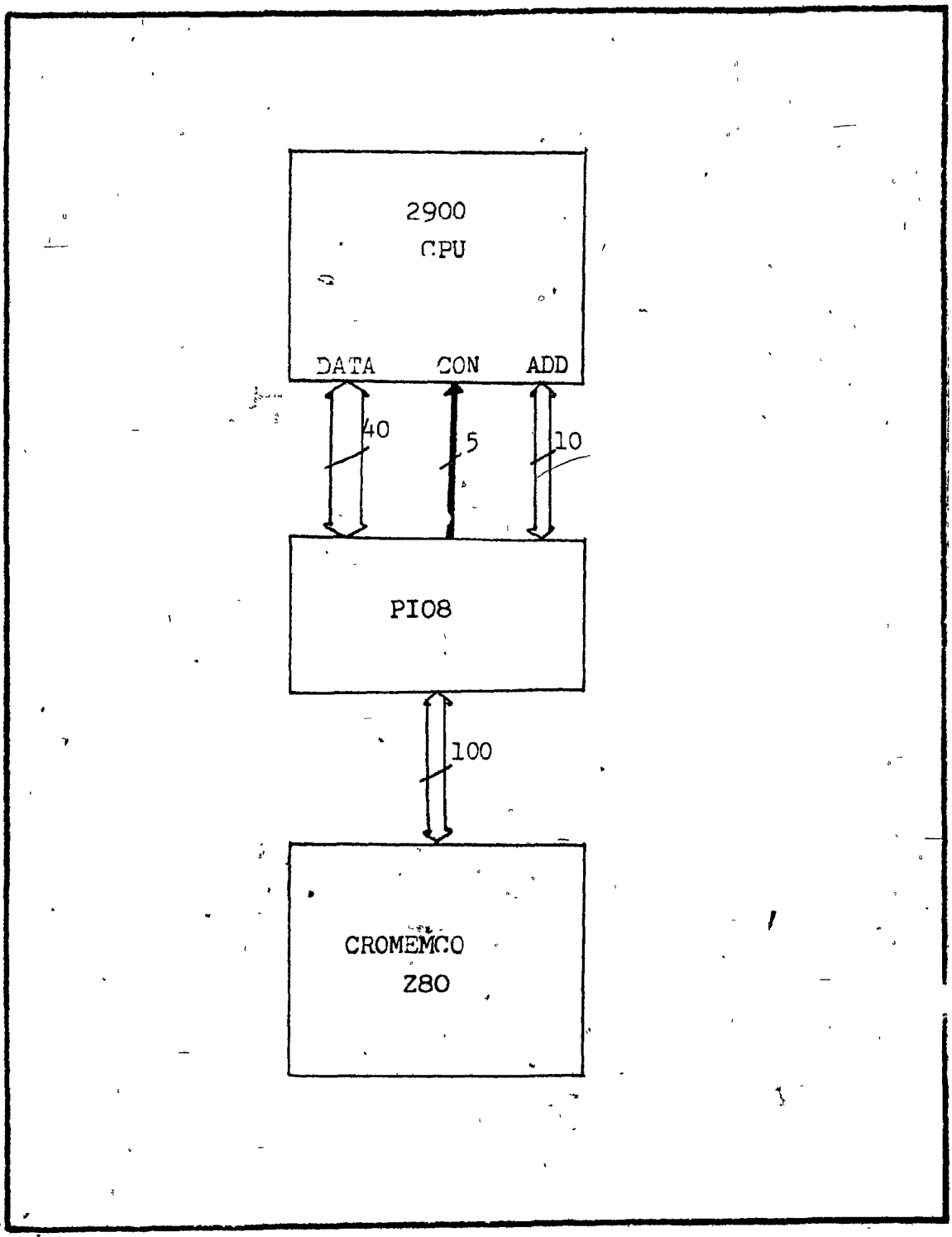


Figure 8-4 CPU test setup

L: Hitting 'L' will invoke the loader and "L>" will be displayed. A four digit address and ten digits of data can be typed in. These two groups are separated automatically and if the digits are not hexadecimal, the programme displays the " ILLEGAL HEX " message. By typing a 'space' the next buffer address is displayed and now typing ten digit of the data would suffice. By hitting the " Return" key, the system goes back to ready state to accept any commands listed in the menu.

R: Typing character "R" will force the CPU sequencer to zero address.

G: Typing character "G" and then "Return" will cause the programme to run in the following manner:

a: Reading the sequencer address

b: Supplying the microword corresponding to this address at the input of the pipeline

c: sending one clock pulse to the CPU

d: Displaying the address followed by the micro-instruction.

and the CPU output at the CRT

f: Repeating the above for each new address offered by the sequencer

Gs: Typing "GS" and then "Return" will cause execution of steps a,b,c,and d. The programme will then look for character "S" and "Return" in order to proceed. In case of "S" the above cycle will be repeated for the new address and in case of a "Return", the programme will be ready to accept any commands of the menu.

P : Typing 'P' will cause a print out on the CRT of the address and microinstruction for each being executed.

T : Hitting character "T" will generate the sequence of pulses which are needed to test the hardware execution of the "TRACE CALL" (mentioned in Chapter VII). The register content are not displayed by this test programme.

C : This character will stop the execution of "CPUTST" programme and the control will return to the CDOS monitor programme.

This programme uses the interrupt facility of the Cromemco Z80. Figure 8-5 shows the flowchart of the "CPUTST" programme.

TEST OF THE 2900 MICROCOMPUTER

With the help of the above mentioned programmes, all three

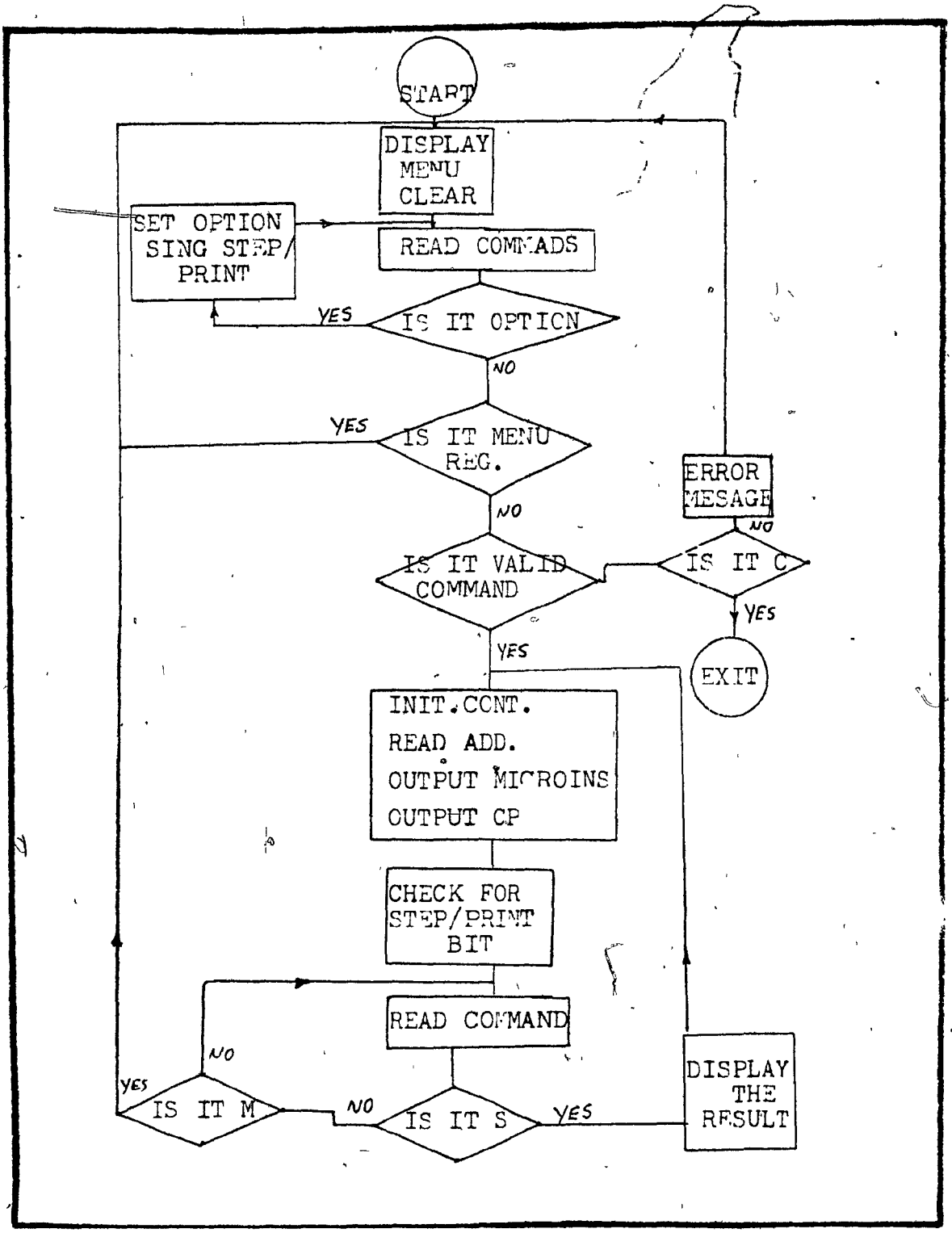


Figure 8-5 CPUTST flochart

boards were successfuly debugged, but individualy. They have now been grouped together in their permanent slots. For testing the microcomputer module, as a whole, a very sophisticated programme(36) has been written. This programme, with the help of Cromemco microcomputer and through two PIO8 boards, has been used to test the combined operation of the CPU, Microcode and Local Memory modules. This test programme supports the testing and debugging of the M2900 microcomputer module including the microcoding software.

Figure 8-6 gives the test setup configuration for the testing of the integrated module.

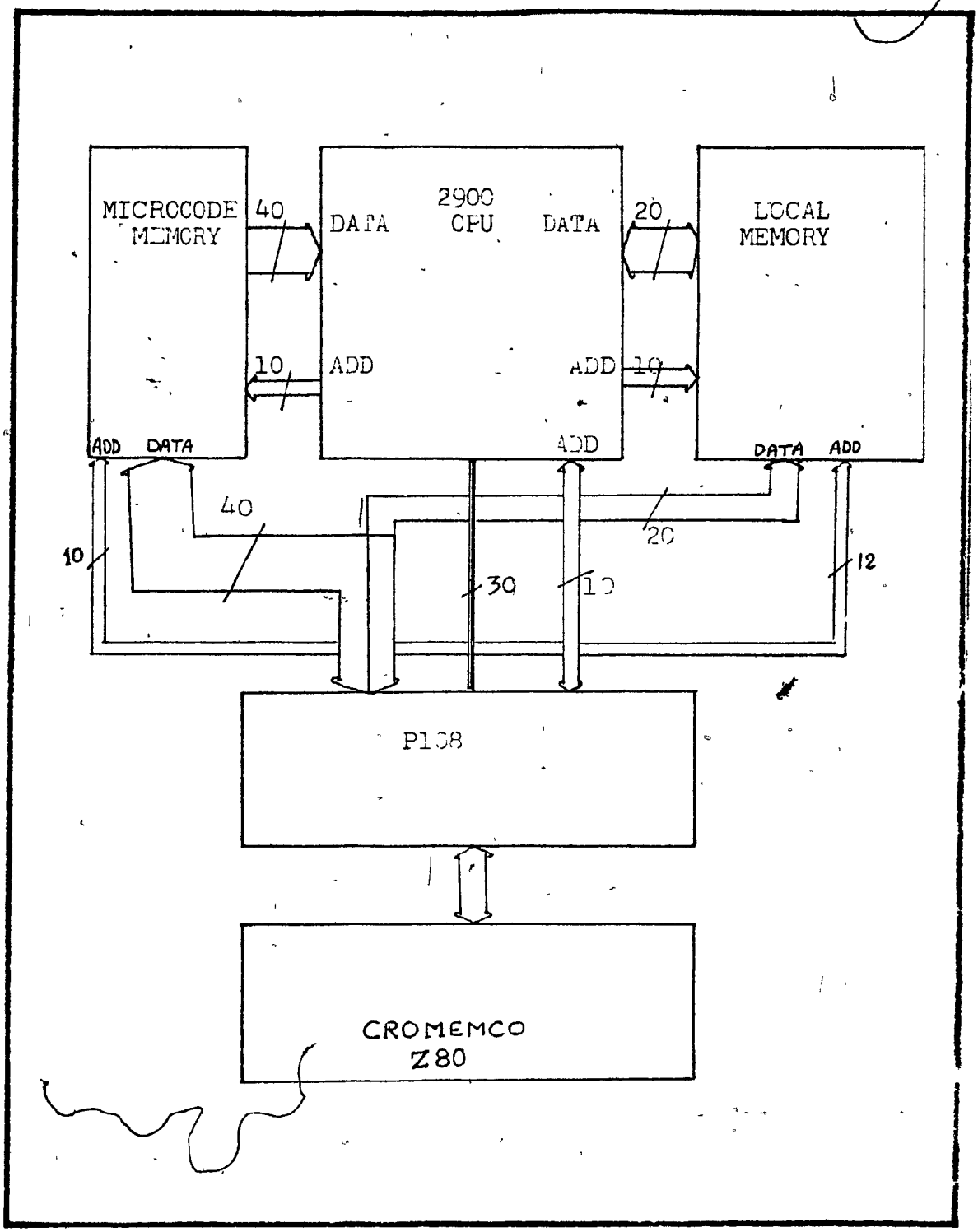


Figure 8-6 2900 Microcomputer test setup

CHAPTER IX

PERFORMANCE EVALUATION OF THE 2900 MICROCOMPUTER MODULE

The 2900 microcomputer module design will be reviewed in this chapter. Comparison between the projected performance and the actual figures are also presented.

The most important parameter, which is of the utmost importance in all the GRADS submodules, as well as the microcomputer modules, is the " Cycle time ". This time should be decreased in order to have a higher rate at which polygons and/or vectors can be animated.

The initial objective of the 2900 microcomputer module was a cycle time of 110 ns, but experience has shown that this rate can not be reached, although, the microprocessor chip alone has a minimum cycle time of 110 ns.

The worst case cycle time for the 2900 microcomputer module has been calculated for the case where the CPU reads from the external register and writes the result in the same set. This time was estimated to be 258 nsec(chap. VII)

Reading and writing from or to the Local Memory is not considered in the above cycle time since its operation should be pipelined by properly microcoded programmes.

The concept of using four different fields for the Microwords and thus diminishing its length has been successful. Except for some restrictions on the programming, this concept has not had any degrading effect on the operating speed of the system.

The external registers have increased the working area for the programmer and have considerably reduced, the time for saving and retrieving intermediate results.

The external multiplier has increased the arithmetic power of the system. After loading the arguments, the product can be read out on the next instruction.

The "tracer" has proved to be a very good tool for the programmer of this micro-computer module. With the help of the tracer, the Microcode can be single stepped and the execution of each instruction observed.

Since the debugging of all other submodules in the GRADS is not yet completed, performance of this module has not been evaluated in the integrated environment. However, the module itself

has been tested with the help of a microcode tracing programme, (36) and some simple routines have been executed on it. The proper execution of each routine has been observed by the "TRACER".

The microcode programming of the operating system and graphics instructions for the 2900 microcomputer module are being completed as a separate thesis project.

DISCUSSION

Currently the evolution in the electronics and computer fields is very rapid. The availability of new LSI components dramatically influences a hardware design making it obsolete in few years. This can be seen in the present implementation which could benefit from the following improvments.

Faster and larger static RAM chips such as the Intel 2148, are now available for the Microcode Memory.

The MC2901 microprocessor can now be replaced by an enhanced version, the MC 2903. Using this microprocessor, the General Purpose External Register implementation can offer the same capability as its Internal Registers with the addition of external two port RAM (AM29705). This RAM can be addressed with the same address fields as the Internal registers. This new configuration can substantially reduce the Cycle Time.

The average execution time of the microcomputer can be improved using a programmable clock generator such as the Am2925. Since the execution cycle of all the instructions are not equal, they can be divided into a number of categories, each category running with a maximum different clocking frequency and optimized duty

cycle.

The complicated logic circuits for the control input lines of the sequencer could be realized by high speed ROMs with greater simplificity a faster execution times. In addition, the microcode instruction format fields could be easily reorganized to correct the present fragmentation, which resulted as this prototype was redesigned and debugged.

CONCLUSION

After a brief history of the computer Graphics, the GRADS (an acronym for the Graphics Real-time Animation Display system) has been discribed. This system is currentely under the design and construction at the Electrical Engineering Department of McGill University.

The architecture of the GRADS presented with specific reference to its use of pipelining and parallel processing.

Two host computer are chosen to serve the system. The DEC VAX 11/780 is intended for the execution of specific applications, while the Cromemco Z80, a smaller dedicated Microcomputer is used for maintenance or test purposes.

The main objective of this research was the design and prototypping of the 2900 Microcomputer module. The 2900 Microcomputer module is constructed on three separate planes; the Microcode, The Local Memory and the CPU plane.

The Microcode Memory is used to efficiently programme graphics subroutines required in the animated display of colored images .

The Local Memory implements two separate buffers , one for receiving high level instructions provided by the host computer and the other for storing the Graphics Controler instructions generated by the 2900 CPU module. This Local Memory design incorporates three ports to accomodate the GRADS architecture.

The hardware function and design of the 2900 Microcomputer has been explained in detail.

A collection of test programmes were developed for the individual debugging of the Microcode, Local Memory and CPU cards. The testing procedure and the programme have been outlined.

The 2900 Microcomputer module has been successfully integrated and tested using a microcode tracing programme.

The performance of this 2900 Microcomputer design was analyzed and its maximum clocking rate established. Suggestion were made for improving the overall design.

This module will soon be operated in the integrated GRADS environment as most of the systems are now operational.

REFFERENCES

(1) ADVANCED MICRO DEVICE CATALOG,

AM 2900 Bipolar Microprocessor Family, 1980.

(2) ADVANCED MICRO DIVICES INC.,

AM 2900 evaluation and learning KIT

(3) AGKLAND, Brayan D. ; WESTE, Neil H.

The edge flag algorithm - A fill method for raster scan display, IEEE Trans. on Computers, Jan., 1981.

(4) AMELING, Walter

Parallelism in computer architecture, World congress system simulation and scientific computation, 1982.

(5) AMPUDIA, R.

A Microprocessor Design using the INTEL 8086 for a Color Graphics Animation System, M. Eng. Thesis, McGill University, 1982

(6) ARTWICH, Bruce A.

Microcomputer Interfacing, Prentice-Hall, Inc., New Jersey, 1980

(7) AYON, M. ; HIN-WAH, Tam

Twelve Plane RGB Video Memory Board, Internal Report, McGill University, 1981

(8) BAER, Jean Loup Multiprocessing systems, IEEE Trans. on Computers, Dec., 1976.

(9) BERESFORD, R.

Integration raires the intellegence resolution... Electronics, Oct., 1981.

(10) BRIDGEMAN, R. C.

An Interface for a Color Display Animation System using Microprocessors, M. Eng. Thesis, McGill University, 1980

(11) BROWN, B. E.

The future of computer graphics, Byte, Nov., 1981.

(12) CARAYANIS, Gregory

A color display system for real time animation, M. Eng. Thesis, McGill Uni., Aug., 1981.

(13) CHEN, T. C.

Introduction to ComputerArchitecture, H. Stone ed. SRA, Chicago, 1975

(14) CLARK, James,

A VLSI geometry processor for graphics, IEEE Computer, July 1980.

(15) CROMEMCO INC.

User Manual for RDOS , CDOS and Z80 Hardware Manuals, 1978.

(16) CROW, F. C.

A comparison of antiatiasing techniques, IEEE Computer Graphics and Applications, Jan., 1981.

(17) DICK, A.

A planar single substrate AC plasma display with capacitive bias, IEEE trans. on Elec. Devices. Aug.,1979.

(18) DIGITAL EQUIPMENT CORPORATION,

VAX 11 Architectur and Hardware Handbooks, 1980.

(19) DINH_TUAN, P.

State inversion of plasma panel cell, IEEE Trans. on Elec. devices, Aug.,1979.

(20) EVERETT, R. R.

The WHIRLWIND I computer review of electronic design,

AIEE-IRE conference, 1952.

(21) GEMBALLA, Renate ; LINDNER, Rolf

The multiple write bus technique, IEEE computer, Sep., 1982.

(22) GHARADJEDAGHI, F.

Anovel type of inverted dichromic display ...., IEEE Trans. on Elec. Devices , Nov.,1980

(23) HAYES, John P.

Computer Architecturee and Organization, McGraw-Hill, 1978

(24) HAYES, John P.

Testability consideration in microprocessor-based design, IEEE Computer, March, 1980

(25) HOBBS, L. C.

Computer Graphics display hardware, IEEE Computer Graphics & Applications, Jan.,1981

(26) HWANG, K.

Cmputer Arithmetic. Principles, Architecture and Design. Wiley, New York, 1979

(27) INTEL CORPORATION

Component data books and 8086 Familly User's Manual, 1979-1980.

(28) KARPLUS, W. J.

Parallelism and pipelining in high speed digital simulators, World congress on system simulation and scientific computation, 1982.

(29) KOYAMA, K, ; SASAKI, S. ; TOKITA, M.

Advanced pattern data processing technique for a RASTER scan electron beam exposure systems, J. VAC. SCI. & TECHNOL, DEC 1981

(30) LOCEFF, Michel

A new appoach to high speed computer graphics, IEEE computer, June, 1980.

(31) LYPACZEWSKI, EPATA

GRADS Microcomputer Local Memory, Internal report, McGill Uni., 1978.

(32) MACHOVER, C.

A brief personal history of computer graphics, IEEE Computers, 1978.

(33) MACHOVER, C. ;NEIGHBORN, M. ; STUART, C.

Graphic display, IEEE Spectrum, Aug., 1977.

(34) MAYER Ware

Computer graphics : The human interface, IEEE Computer, June, 1980.

(35) MICK, JOHN ; BRICK, JAMES

Bit-Slice Microprocessor Design, McGraw-Hill, 1982.

(36) MIGNOT, Alain

Un logiciel de development et d'exploitation de Mico programmes pour un Micro-ordinateur 2900,M. Eng.Thesis McGill Uni. Nov., 1980.

(37) MIKOSHIBA, SH.

A positivecolumn discharge memory panel for color TV display, IEEE Trans. on Electronics Devices, Aug, 1979.

(38) MILLER, W.

Display requirement for future Man-Machine systems, IEEE Trans. on Elec. Div., 1971.

(39) MORCE, S. P. ; RAVENEL, B. W. ; MASOR S. ; POLMAN, W. B.

Intel microprocessor 8008 8086, Computer,Oct., 1980.

(40) MOTOROLA SEMICONDUCTOR Component Data books 1976-1981.

(41) NEGREPONTE, N.

Raster scan approach to computer graphics, Computer &Graphics Feb. 1977.

(42) NEWMAN, W. M. ; SPROULL, R. F.

Principles of Interactive Computer Graphics, McGraw-Hill, 1979.

(43) NIINA, T.

A high brightness of green LED flat panel device for character and TV display,Aug., 1979.

(44) OKAMOTO, Y.

A positive column discharge memory panel without current limiting resistors for color TV display, IEEE Trans. on Elec. Dev. Sept., 1980.

(45) PANCHOLY, R.

A Networking Interface for a Multiprocessor Color Graphics Animation System, M. Eng. Thesis, McGill Uni.(to be submitted).

(46) PANCHOLY, R.

Host Computer Interface (HCI), Internal report, McGill Uni. 1980.

(47) PAPAPETROS, A.

The design of a Color Display System for Real Time Animation using Microprocessors, M. Eng. Thesis, McGill Uni., 1977.

(48) PRITSKER, A. Alan B.

Introduction to simulation and SLAM, John Wiley & Son, 1979

(49) ROBERT, J.

TV image with liquid cristal display, IEEE Trans. on Electronics, Aug., 1979.

(50) SATYANARAYANAN, M.

Commercial multiprocessing systems, IEEE Computer, May, 1980.

(51) NOT USED

(52) SHOTLIFF, J. R. ; HALL, K. L.

The nodal parallel system, Conference paper, Sorento,

Italy 1980

(53) SIEWIORK, D. P. ; GORDON BELL, C. ; NEWELL, Allen,
Computer structures: Principles and Examples, McGraw-Hill, 1982.

(54) SRIDHAR, THIRUMALAI ; HAYES, JOHN P.
A functional approach to testing Bit-Sliced Microprocessores, IEEE Trans. on Computers, august 1981.

(55) STRITTER, E. ; GUNTER, T.
Microprocessor Architecture for a changing world , The Motorola 68000, IEEE Computer, Feb. 1979

(56) TEIXEIRA, J. ; DEMIR, B.
A RGB Video Memory for GRADS, Internal Report, McGill university , 1981.

(57) TEXAS INSTRUMENTS INC,
The TTL data book and Bipolar Microcomputer components, 1977-1980.

(58) THURBER, K. J.
Parallel processor architectures , Computer Design, Jan.

and Feb., 1979.

(59) THONG, W. K. ; CHUNG, M. ; HIAM, C.

A Color TV Sequencer, Internal report, McGill university, 1981.

(60) YAMAGUCHI, T.

Ahigh brightness gap multicolor LED, IEEE Trans. on Elec. Dec. May 1981.

(61) VANAKEN, Jerry R. ; ZICK, Gergory L.

The expression processor: Apipelined, multiple processor architecture, IEEE Trans. on Computers, Aug., 1981.

(62) ZILOG,

Z 8000 CPU Technical Manual, 1980.