A COLOR GRAPHICS DISPLAY SYSTEM

FOR REAL-TIME ANIMATION USING MICROPROCESSORS

by

© Salim G.A. Ramji, B.Eng.

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.

April, 1979.

ABSTRACT

This thesis describes the hardware implementation of a graphics display system dedicated to perform real time animation of computer generated images on a raster scanned color television monitor. The design uses a frame buffer concept and an array of paralleled microprocessor units. The design, realization and testing of the display section are emphasized. A projected performance evaluation is also presented.

RESUME

Cette thèse décrit la réalization d'un système graphique capable de générer des images à partir d'un ordinateur, et ceci en temps réel sur un écran de télévision en couleur. On applique un nombre de microprocesseurs exécutant en parallèle et fournissant une mémoire vidéo.

Le dessin, la réalization, et la vérification de la section d'affichage graphique sont détaillés. Le rendement projecté pour ce système est analyzé.

ACKNOWLEDGEMENT

I would like to express my sincere 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.

I also greatly appreciate the understanding, the motivation and the moral support given to me, during my work, by my lovable wife, Al-Shamsha.

I also wish to deeply thank my parents for giving me support and love at all times; and to the Agakhan Foundation for financing me during my M.Eng. program. I am also extremely grateful to the Faculty of Graduate Studies and Research at McGill for granting me a Summer Research Fellowship.

Sincere thanks are also due to Mr. John Mohammed and Jean-Yves Lamarre for their valuable suggestions, and to Mrs. P. Hyland for the excellent typing of this manuscript.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

Computer graphics has in recent years been a field of great interest within the industrial and academic communities. Computer aided graphics-display systems fulfill the old saying that a single picture is worth a thousand words [31] when wanting to describe the material world.

After the development of Whirlwind [15] - the first computer that had a graphical output - researchers and designers became interested in developing better graphical computer systems. A further milestone was achieved when Sutherland [57], utilizing Licklider's [32] concept on interactive computing, published his report on the "Sketchpad" - the first interactive computer graphics system. Since then, interactive computer graphics has had a revolutionary impact on the design of both the hardware and its dual, the software, of later computer systems.

At present there are three main areas in computer graphics where most of the efforts are directed. The first is to improve the capability of graphical displays [46] to represent things and processes. This capability extends into dimensionality, versimilitude, complexity and motion.

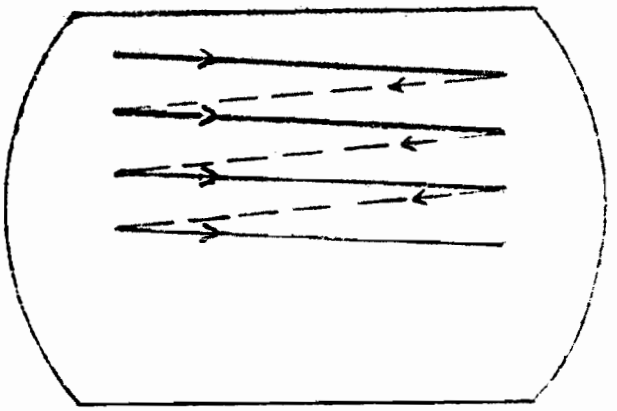
The second main effort is to improve the interaction between men and computers by using graphics as the communication medium.

The third main effort is to develop applications of computer graphics.

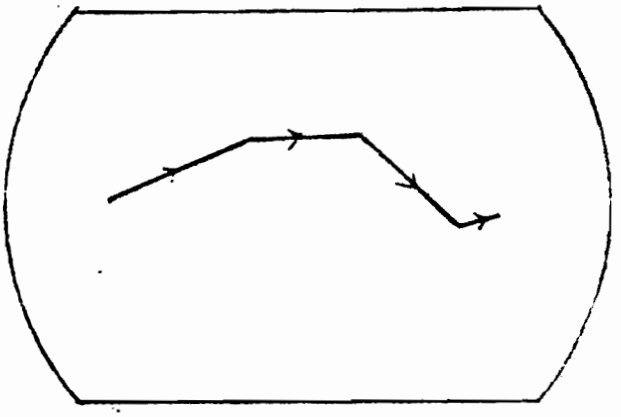
We shall briefly discuss the efforts directed towards the development of these main areas.

1.1 Efforts in Improving Graphical Display Technology

Computer graphics during its active history [36] has always been heavily influenced by graphic display technology [46]. Although computer-graphics displays incorporate many diverse technologies, two techniques are fundamental to all graphics display devices. They are the random position (also known as line-drawing or calligraphic displays), and the raster scan [37]. Figure 1.1 illustrates these. Random positioning offers high resolution, as would typically be required in engineering and scientific applications. Its drawbacks include limited color capability and low flicker free information content. The major issue in the design of random positioning displays is the refresh process : the light-emitting phosphor of the CRT must be excited repeatedly if the picture is to be displayed without flicker. The Direct View Storage Tube (DVST) [10], [45] solves the refresh process issue by retaining the picture on a storage mesh inside the tube. However, if only parts of the picture are to be deleted, it cannot be done with the DVST unless the whole of the screen is cleared and the picture redrawn. The Plasma Panel [4] merges the storage capabilities of the DVST and the selective erasibility of the CRT . Other devices of matrix construction have appeared more recently and they include the liquid crystal display [29] and the electroluminescent panel [20] . However the CRT has established well over its 100 years of history and there are various hurdles in displacing it [60] .



(a) RASTER SCAN



(b) RANDOM POSITION

F I G U R E 1.1

FUNDAMENTAL TECHNIQUES IN GRAPHIC DISPLAYS

As there is no inherent memory when using the CRT, the picture must be stored as a display file in a refresh memory, and passed repeatedly to the CRT via some form of channel or display processor. In order to conserve memory and speed up interaction, early graphics systems used to have the picture stored in a highly structural form. This resulted in a complex display processor that passes the information to the CRT. Later on it was shown [44] by Sproul and Newman that by avoiding the hierarchic structures and by placing the display file after the transformation process it is possible to get a simpler and a more generally useful graphics system design. Hence the most convenient way of designing a graphics system was to utilize a display processor that performs the transformations and then stores the transformed picture in a refresh memory as shown in Figure 1.2. This method can be used in highly interactive applications without burdening the transformation processor, which no longer has to process lines rapidly enough to prevent flicker. Using microprocessors these systems have a higher performance / cost ratio compared to the previous non-buffered displays.

The raster scanning technique offers excellent color presentation and high flicker-free information content. A further point in favour of this technique is that it uses a T.V. monitor which has a highly developed and relatively inexpensive technology. However its limitation is resolution - typically one half to one fourth that of random position. Moreover the display file for a raster-scanned CRT must normally be arranged as a set of intensity values, rather than as a list of line segments. This demands a

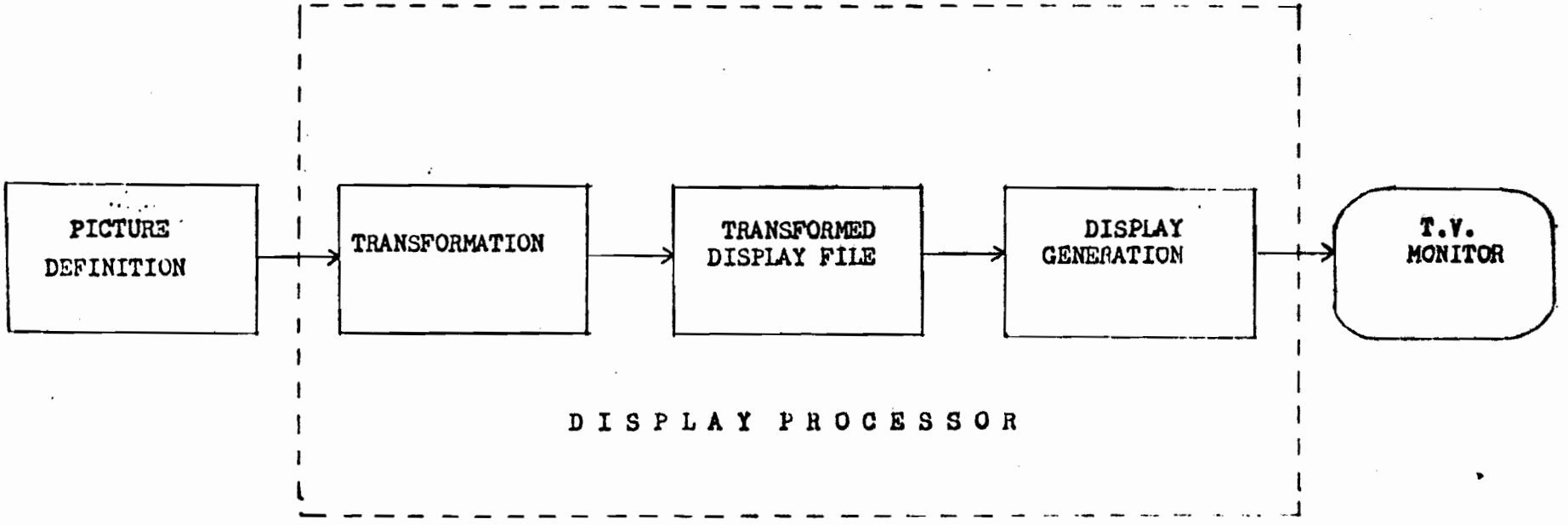


FIGURE 1.2

BUFFERED TRANSFORMATION PROCESSOR

large frame buffer [24] for the display file. It also means that each line segment is represented by many intensity values stored throughout the buffer. Speed of interaction suffers because of the slow scan conversion process that computes these intensity values and because of the difficulty of modifying individual lines in the displayed picture. One other problem with the raster-scan CRT is the way in which lines drawn at a slant (with a non-integer slope) tend to show ugly staircase-like quantization effects.

The information in a frame buffer is basically an unencoded video image, hence the bulk. There are several ways of encoding the image so as to save space : they all require construction of a processor capable of decoding the information and passing it to the CRT. In many ways the most attractive method is to encode the display file such that it is identical in form to that used with line-drawing CRT so that identical software can be employed. The problem with this approach lies in designing the processor to perform what is called, real-time scan conversion to pass over the display file, 30 times a second, converting it into a video signal. Useful real-time scan converters can be built if some means of using parallel processors is implemented.

However the encoding schemes developed so far hamper interactive modifications to the picture. The frame buffer avoids this problem. Moreover with the frame buffer, pictures may contain solid areas of uniform or continuously varying gray or colored shading. Also the background

shading can be in various solid colors. Frame buffers have been used very effectively for computer art. Systems in market such as the 'Visual Data Processor' and the 'Micro Video Processor' [47] by Norpak and the ID 5000 series [11] of display system by Deanza Systems Inc. utilize the Frame Buffer concept.

It has been suggested by Newman [46] that real progress in this area can be achieved by merging the frame buffer and the processors for real-time scan conversion, in such a way that the frame-buffer receives the output of the scan-converter. The advantages of both schemes are retained in this approach. The GCT-3011 developed by Genisco [19] uses this scheme.

The display processor and the real-time scan converters form a part of satellite graphics system in which either one of them (depending on which graphic technique is used) is a satellite and is dedicated to perform special tasks. Satellite systems fall between the class of dedicated stand-alone systems and large multiprogrammed computer systems that support a number of activities [17] . They exploit the advantages of both the system types. The advantages are in many ways the same as those when using distributed computing networks - like resource conservation, accessibility, response, compatibility , etc., etc. References [18], [53] , and [66] describe these and more advantages in detail. However the extensive software required for the satellite graphics system is very complex [46] .

1.2 Efforts in Improving Man-Computer Interaction

For computer graphics to be effective, it has to be a very good medium of communication. One of the most important problems in computer graphics is that of establishing excellent two-way man-computer communication. The language between the two should be able to recognize not only figures but such abstract ideas as force, field, cause, effect, etc. Only at a few places have such computer programming interactions made man-machine interactions play a very important role in computer aided design of complex systems and working with various real-time systems. In the animation[1] industry, man and machine work together resulting in a process called "interactive computer-mediated animation" [2] . Three aspects of the role of direct graphical interaction in computer graphics are particularly relevant to computer animation:

(i) The availability of immediate visual feedback of results, final or intermediate;

(ii) the ability to factor picture construction into stages, and to view the results after each stage;

(iii) The ability to sketch pictures directly into the computer.

---

1. Animation is the graphic art which occurs in time. It conveys complex information through a sequence of images seen in time. The source of information for the viewer of animation is in the changes in picture : change in relative position, shape and dynamics.

Immediate visual feedback in animation can bring about revolutionary results. In the next section, it will be shown how important it is for application purposes.

At present work is going on in the development of 3D input and output devices [50] .for better man-machine communication.

1.3 Applications of Computer Graphics

Computer graphics applications are classified into six general areas [37] :

(i) Management Information

Management information systems offer decision makers faced with a large amount of statistics, a timely and convenient method to determine trends accurately [52] . The computer displays bar charts, graphs and other pictorial data and interaction is possible. IBM's "Trend Analysis 370" and General Electric's "Genigraphics" are two of the most innovative systems in the field. Large screen conference-room displays for tasks that require team efforts are becoming very popular.

(ii) Scientific Graphics

For scientific graphics two- and three-dimensional plotting tasks are quite common. Typical ones are waveform analysis, function

visualization, curve fitting and interactive design and analysis. Interaction is very useful when designing filters etc., for optimal results. Both static and dynamic displays are used, though the former is more popular due to its lower cost.

(iii) Command and Control

In the operation of very large systems, a control centre is very essential in order to monitor and control selected subsystems. At the centre, computer graphics is often used to present information in a useful format. Electric utilities are the largest group of commercial users of this application. The military uses command-and-control graphics to a vast extent also, especially for tactical simulations of battle conditions.

(iv) Mechanical and Electrical Design

Computer-aided drafting and design is always more cost and time effective compared to manual techniques [38] . This type of designing is very useful for designers of automobiles [51] and aerospace vehicles. By interacting with the computer, the design on the display can be modified and altered to achieve optimization.

Computer aided designing has also been very useful to design printed-circuits integrated circuits and hybrid circuits.

(v) Image Processing

Image processing is very useful in applications like surveillance and non-destructive examination of materials [26]. In recent years, the analysis of data collected remotely by earth satellites is getting very useful. Users of such data include weather-related services, agriculture forecasters, land developers, geological surveyors, [21] , etc., etc. Image processing is also becoming very useful in medical application [12] , [22] , [56] .

(vi) Real-time Image Generation

Real-time computer generated imagery is a very rapidly developing area, due to its usefulness. One of its first applications was in the development of simulators for air and space crafts. Animated images simulating the landing strip etc., are observed by the trainee pilot directly viewing a CRT. By proper interaction with the real-time computer, he can be trained to perform difficult tasks. This type of "man-in-the-loop" equipment is much safer and cost effective than actual training in the aeroplanes, ships, etc. Another very interesting application of real-time Shaded Computer Graphics[1] is in the entertainment industry [8]. 2-D and 3-D animation of pictures and models are aided by computers to produce motion pictures and cartoons. American International's "Future-

---

1. This refers to computer-generated images in which intensity is calculated for all the resolvable picture-elements (called pixels) on the screen.

world", a film released in August 1976 uses computer graphics to produce its special effects. Also a few scenes in the now-famous movie "Star Wars" were computer-generated by Larry Cuba and his associates [41] .

T.V. games, which also use interactive computer graphics, are also getting very popular lately.

The use of real-time computer-graphics technology to produce low-cost films for educational purposer promises a lot of benefits [23] .

It is in the area of real-time animation computer-graphics systems, that our efforts have been directed. So we shall outline this area in greater detail.

Two major problems are encountered in implementing real-time animation:

(a) The rate at which the display data is to be transferred to the display device has to be fast enough to satisfy the animation process and also to avoid flicker;

(b) the computations involved in generating the animated picture information are enormous.

When a raster-scan CRT is used, in order to maintain a flicker-free animated image, each pixel (or picture element) information has to be changed 30 times per second. In the high resolution mode,

the display screen is made up of 512 x 512 matrix of pixels. Also if the display is colored with each of the primary colors, Red, Blue and Green having 32 (2^5) different shades (or levels), each pixel has a fifteen bit word containing the color information. Hence for flicker-free display, (512 x 512 x 30) ≈ 7.86 M fifteen bit computer words must be delivered to the CRT every second.

One of the best solutions to this problem of high computation and high throughput [64] is to apply parallel computing techniques [55] . With the advent of cheaper hardware and hence cheaper processing units, the trend towards this type of solution is a natural one [33] .

Of the four categories of computer systems as classified in Flynn's widely referenced paper [16] , the Multiple-Instruction Multiple Data [MIMD] organization will be outlined here as it is the most general and very promising in future applications.

The MIMD structure consists of a number of independent processors, each one operating on a different data stream and executing a different instruction set. There is some interaction between the processors to provide for communication during execution. Two features are of interest to differentiate among designs: the coupling or switching of processor units and memories and the homogeneity of the processing units [3] . In tightly coupled multiprocessors, the number of processing units is fixed, and they operate under the supervision of hardware controller. The trend towards tight coupling is apparent only in supercomputers.

When the system is going to consist of numerous small modules and when it is intended for general-purpose applications, the connection are necessarily of the loose type. A looser and more modular type of coupling is made in recent homogeneous multiprocessing systems and distributed function architecture. For switching between processors and memory, one of the best approaches is to use time-shared buses. The simplest way of organization is to have all processors, memories, and input / output units connected to a single bus. However to attain more parallelism, at the price of more complexity one can have several buses, either uni- or multi-directional. Priorities can be given to specific units by adding an arbitrator to resolve them. If one wants to connect a large number of processors (e.g., microprocessors) and small memory modules, the time-shared multibus techniques seem to be the best. The bus arbitrator scheme has been chosen for the Minerva multimicroprocessor [67] with a single bus for transmission between devices, but with each device having a direct connection to the arbitrator.

The objective of this thesis is to present a display subsystem for a graphics system for real-time animation that is being implemented at McGill. Particular emphasis will be made on the presentation of an arbitrator cum interpreter (called the Graphics Controller) since the author was involved in its design and realization. The author was also responsible for testing the other modules of the display system and integrating them with the Graphics Controller.

The design uses a raster-scan type of display and the MIMD architecture to cope with the high computation rates involved. The coupling of the processors and the memories is loose and modular. The switching is of the time-shared multibus type to accommodate an efficient operation of the system with many processors. The memory is of the frame-buffer type. Chapter II will present the system architecture.

The Graphics Controller is responsible for the arbitration of the buses to the processors and memories. Generally it acts as one of the satellites of the system which helps in accomplishing tasks distributed to it.

Chapter III describes the functions of the Graphics Controller in greater detail. It describes the Graphic Modes that the Graphics Controller implements. In Chapter IV the design and implementation of the Graphics Controller is discussed. The T.V. Sequencer and the Video Memory plane is briefly discussed here also. Chapter V evaluates the performance of the Graphics system as a whole and illustrates the role of the Graphics Controller in the performance. In this chapter the performance of the four graphic modes that the system uses are also evaluated.

Chapter VI discusses the assembly and debugging of the Graphic Controller and its integration with the T.V. Sequencer and the Video Memory.

Chapter VII presents some conclusions.

CHAPTER II

THE SYSTEM ARCHITECTURE OF GRADS

As mentioned in the previous chapter, enormous computations and high transfer rates are necessary to maintain flicker-free, colored and animated image on a raster scan CRT. Most general purpose computers presently available cannot fulfill the above requirements necessary to display such an image.

However work is presently underway in the Department of Electrical Engineering at McGill to develop a satellite system that is used by a general purpose host computer to perform the task mentioned above. This system, named GRADS (an acronym for Graphics Real-time Animation Display System) is dedicated to generate, store, manipulate and display animated and colored images on a standard raster-scan T.V. monitor. The GRADS display system uses the frame buffer concept to obtain high density displays without flicker and uses the full color range of the raster T.V. color monitor. The GRADS design uses a satellite system, offering the advantages of a dedicated stand alone system as well as exploiting the power of a large multiprogrammed computer. The host computer is responsible for supervising the whole system.

By using micro-computers in parallel, the throughput capability of the host computer is increased since the high level information is expanded into the detailed pixel information which is fed to the display

[48]. This allows the host computer to process the image at the high level only and let the microcomputers perform the basic operations. The performance to cost ratio of the system increases also as the micro-computers used are available at a fairly cheap price. A multi-bus structure has been adopted to accommodate the large throughput rates, that are required for the task at hand.

The whole system accommodates flexibility also. Although it has been designed for real-time animation purposes, it can be made to perform other tasks which need parallel processing. In an academic environment this is definitely useful. GRADS is also versatile in terms of modularity and expandability. The time-shared bus structures enables us to use the system with a variable number of modules. The system throughput can be enhanced by adding more micro-computer modules or more system memory modules. Also the throughput can be increased by using faster memories (as illustrated in Chapter V).

This can be achieved by simply replacing the present chips with faster versions. Entire modules designs can be improved by respecting bus compatability. Also a modular system is generally easier to design, debug and maintain than a large integrated system.

One very versatile feature that GRADS has is that, both the number of frame buffer planes selected and the display resolution are software programmable. Either square or 4/3 aspect ratios can be obtained by the user. The system is also software programmable for choosing be-

tween colored displays and black and white displays. A choice between "opaque" writing mode which overwrites previous frame content and "transparent" mode which mixes the incoming information to the previous frame buffer content, is available.

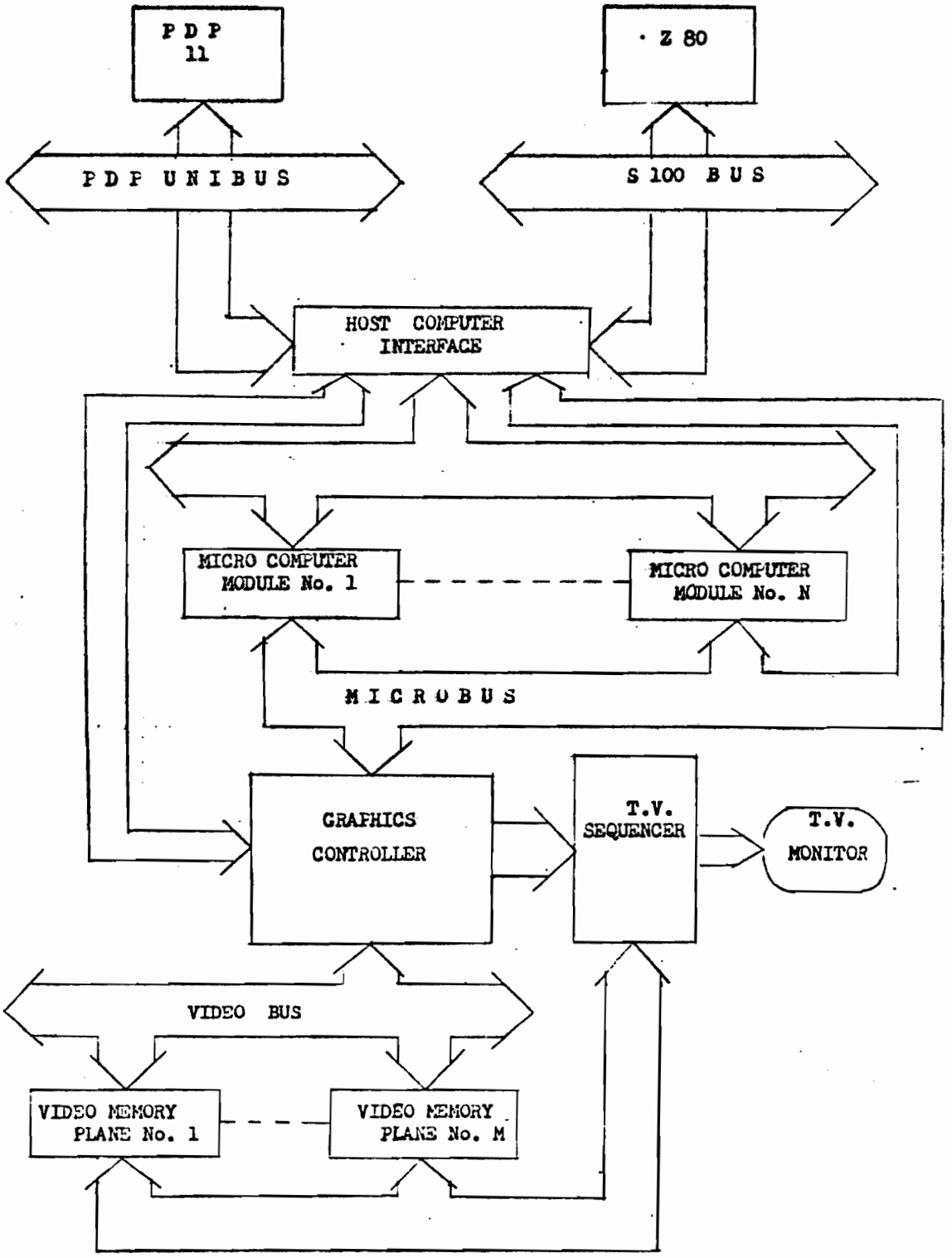
2.1 The Implementation of GRADS

The architecture used to implement GRADS is illustrated in Figure 2.1. It exhibits a pyramid structure with the host computer at the top and the system memory at the bottom. Highly compacted information at relatively low rates are associated at the top and the microcomputer and the Graphics Controller expand and process this to large volumes of pixel information travelling at high rates to the Video memory planes at the bottom from which the T.V. Sequencer shifts data serially to display a flicker free (30 frames per second) colored display on the T.V. monitor.

In the following sections, each of the modules shown in Figure 2.1 will be discussed.

2.2 The Host Computers

The role of the host computers is to control the activity of the system and to generate the high level commands to be used by the microprocessors. To be able to handle complex animations, a powerful computer



F I G U R E 2.1

G.R.A.D.S. ARCHITECTURE

having a large memory is required. These include functions such as 3D rotation, translation, clipping and color shading. However if simpler tasks are to be performed, such as debugging and system maintenance, the use of a powerful computer is not necessary. A S100 based 8 bit microprocessor system with associated peripheral facilities is adequate for this purpose. It can be dedicated all the time to the system for debugging and maintenance purposes, as opposed to the bigger computer which is less readily available. Hence the system was made to accommodate two types of host computers. For high throughput rates, an interface to the Unibus of the VAX 11/780 computer in the Image Processing laboratory of the Electrical Engineering Department, is included. The VAX [14] is a 32 bit computer with 512K bytes of memory, a 67M Bytes disc and a tape drive. The VAX VMS operating system is a virtual memory system with multiusers and multitasking.

For low throughput rates the S100 micro-computer bus is used. This supports a Cromemco Z2 microprocessor system (using the Z80 microprocessor) with console terminal and secondary storage on floppy disc.

2.3 The Host Computer Interface

The Host Computer Interface (HCI) provides a fast communication channel between different parts of the system [27] . It basically accomplishes four tasks:

(i) It can perform DMA between the memories of any of the host computers and any micro-computer module. It also has a direct path to the micro-bus, thus bypassing the parallel microcomputers. This is useful for maintenance or trouble-shooting purposes.

(ii) It provides the host computer with a path to access the status of the different microprocessors.

(iii) It also provides a path to transfer one word of data from a microprocessor to the UNIBUS host computer, without performing a full DMA transaction, thus eliminating DMA overhead.

(iv) The DMA Controller of the Graphics Controller uses a ROM Sequencer concept [34] to implement its two Mealy machines - the Header machine and the Transaction machine. It is the responsibility of the Host Computer Interface to load the microcode into the RAMs (used instead of ROMS since updating its contents is easier than those of ROMs).

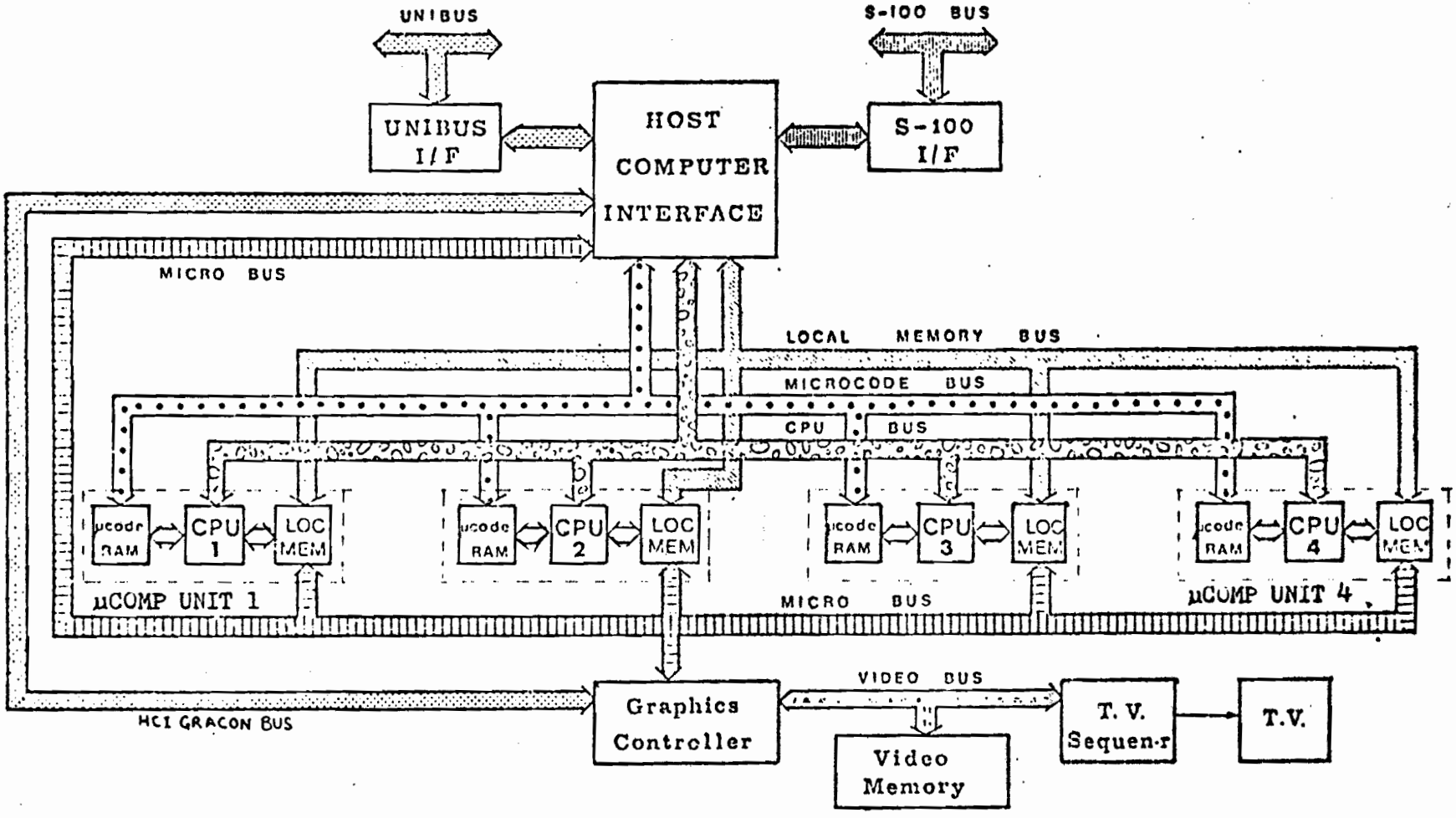
To perform these tasks, a multibus arrangement has been selected. This organization allows service to several microcomputers with the minimum number of interconnections and it also permits high data transfer rates to be obtained. Figure 2.2 illustrates the buses with which the Host Computer Interface communicates with the rest of the system.

2.4 The Micro-Computer Modules

The host computer sends data and commands to the microcomputers, each of which consists of the CPU, a Local Memory and a Microcode Memory (see Figure 2.2).

The micro-program is written into the fast RAM [59]. This memory is a 1K x 40 bit arrangement and can be expanded to 4K. The Local Memory [28] is a slower RAM memory with 4K x 20 bit arrangement. The macro-instructions from the host computer and the data computed by the microprocessor are stored in the Local Memory. For each microcomputer a local arbitrator arbitrates simultaneous requests to the Local Memory so that the Graphics Controller, the HCI and the CPU have bidirectional access to it.

The CPU used in the microcomputer is based on the AMD 2900 [1] family of bipolar bit slice microprocessors. It features a 40



F I G U R E 2.2

THE MULTIBUS STRUCTURE USED BY GRADS

bits micro instruction word, 20 bits data words, and has a read only access to the Microcode Memory, a read/write access to the Local Memory, and has a direct path to the HCI as well as control signals to the Microbus. It has a fast cycle time of approximately 200 nsec.

2.5 The Graphics Controller

The Graphics Controller is responsible for merging partial image buffers from the parallel microcomputers into the Video Memory planes. It is also responsible for the arbitration of the two main buses of the system [40] (the Microbus and the Video bus).

It also interprets the data from the Local Memory of the Microcomputer before storing the corresponding data into the Video memory. The interpretation depends on what graphic mode the system is using. These modes are the Solid Area Mode, the Point or the Line Mode, the Shading Area Mode and the Read Back Mode. When certain areas of an image have to be shaded in one uniform color, the Solid Area mode is the best type of mode to use. When an area has to be shaded with uniformly varying color intensities, the Shading Area mode is preferred.

The Point Mode is useful for displaying lines and vectors. The Read-back Mode is essentially used by the system for maintenance purposes. The details of these modes are explained in Chapter III.

2.6 The Video Memory

The video memory of the GRADS system uses the frame buffer concept. Here a number of similar planes are parallelled to augment the number of bits of color encoding associated with each of the three primary color channels, red, green and blue. Each such plane stores a sequential array of bits corresponding to the adjacent picture elements (pixels) of the image displayed on the television monitor. The basic video memory plane module contains 64K bits of static RAM arranged as 4K words of 16 bits. Low resolution displays of 256 by 256 pixels require one such plane per bit of color encoding for the display frame. The high resolution mode of 512 by 512 requires four such planes per color bit. Larger word sizes of 20, 24, 32 bits per word are required to support the 4/3 aspect ratio as well as higher display resolutions. The current design supports a maximum of 15 planes giving up to 32 intensity levels for each of the three primary colors. Chapter IV gives further details of the Video Memory Plane.

2.7 The T.V. Sequencer

The T.V. Sequencer [25], [65] is responsible for continuously reading digital data from the Video Memory planes and generating the analog signals required to drive a color television monitor. By using a double buffer arrangement it can refresh the picture 30 times a second. This it does under the supervision of the Graphics Controller giving the T.V.

Sequencer the highest priority in arbitrations against any other DMA transfers in progress.

The T.V. Sequencer is responsible for generating and supplying the T.V. Monitor, with a standard composite synchronization signal to keep it synchronized with the data being supplied via the red, green and blue channels. It is also responsible for generating the blanking signals and the even/odd field information for keeping in step with the synchronizing information being supplied to the T.V. Monitor. Chapter IV gives further details of this module also.

CHAPTER III

THE FUNCTIONAL DESCRIPTION OF THE GRAPHICS DISPLAY SYSTEM OF GRADS

3.1 The Graphics Controller

The main function of the Graphics Controller module involves servicing the flow of image information to the video frame buffer. To reduce the workload on the microcomputer modules, the Graphics Controller uses a DMA process to accomplish the block moves of data from the microcomputer buffers to the Video memory. It also maximizes the throughput data rates. The volume of data calculated and buffered at the microcomputer modules is minimized by using a header format preceding the block of data. The header format defines four packing modes which are described later in this chapter.

The Graphics Controller is responsible for reading and interpreting the data from the Local Memory of the microcomputer with the help of the header, before storing the corresponding data in the Video Memory planes. Hence it is the responsibility of the Graphics Controller to develop all the control signals and address information necessary to execute the DMA reading and writing between the parallel microcomputers and the Video Memory Planes.

As the design of the display system uses a time-shared multibus structure (the Microbus and the Videobus), it is necessary to arbitrate these buses between the various devices on a priority basis. This responsi-

bility is undertaken by the Graphics Controller. On the Microbus, it arbitrates the bus between the parallel micro-computers. The arbitration scheme gives the highest priority to the microcomputer which is closest to the Graphics Controller. On the Videobus, it arbitrates the bus between the Video Memory planes and the T.V. Sequencer, giving the latter the highest priority. While the DMA transfer is proceeding between the buses, the T.V. Sequencer, at regular intervals posts requests to the Graphics Controller for gaining access to the Video Memory. By using a double buffer arrangement in the Video Memory, the Graphics Controller services the T.V. Sequencer's request anytime before the next request arrives. This service is done on a cycle stealing basis by suspending the DMA activity for one cycle.

From an economics point of view, the Graphics Controller was designed to minimize control hardware in the micro-computers and in the frame buffer. As a result, the hardware complexity of the individual modules is decreased and replicable (or bus compatible) modules (of microcomputers and Video Memory) can be produced at a much lesser cost.

In the next section, the graphic modes that the system uses, will be described.

3.2 The Graphic Modes

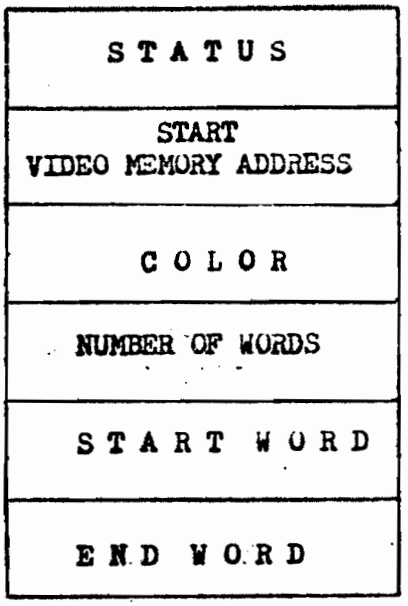
As has been indicated previously, the micro-computer executes the Macro-instructions under the control of appropriate Micro-programs to "expand" the information from the host computer into properly formatted detailed data which is stored into its Local memory. Part of this data from the microcomputer's Local memory consists of a header block which contains information about the type of graphic mode the system is using. The Graphics Controller interprets this header and generates the data which is to be stored in the Video Memory plane.

In this section, the formats of these graphic modes are presented.

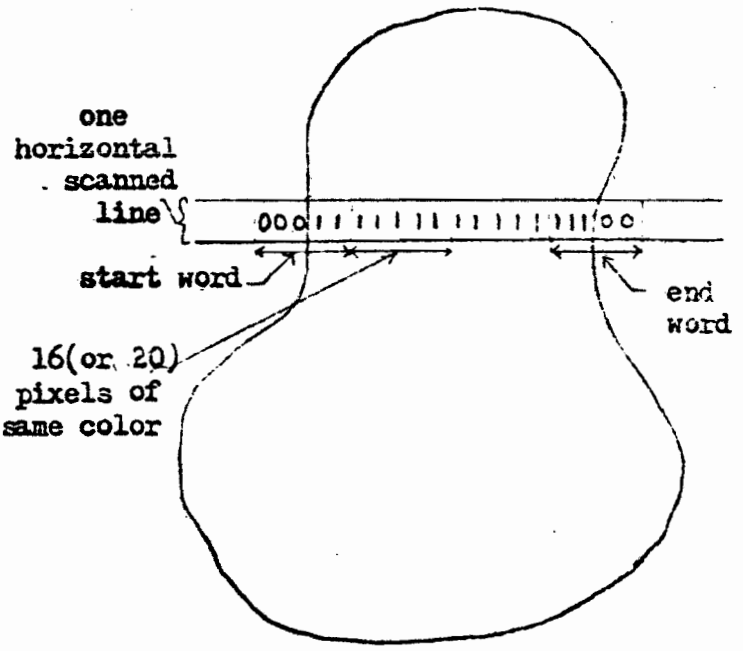
3.2.1 The Solid Area Mode

This mode is useful for painting areas in a chosen uniform color. This color may be one of the possible colors formed by mixing any of the intensities of the three primary colors, red, green and blue. Figure 3.1(a) shows the header block with its content and Figure 3.1(b) shows an (arbitrary) area to be scanned using this type of mode. For each horizontal scanned line in the bounded area shown, the micro-processor lists the header block containing the information shown.

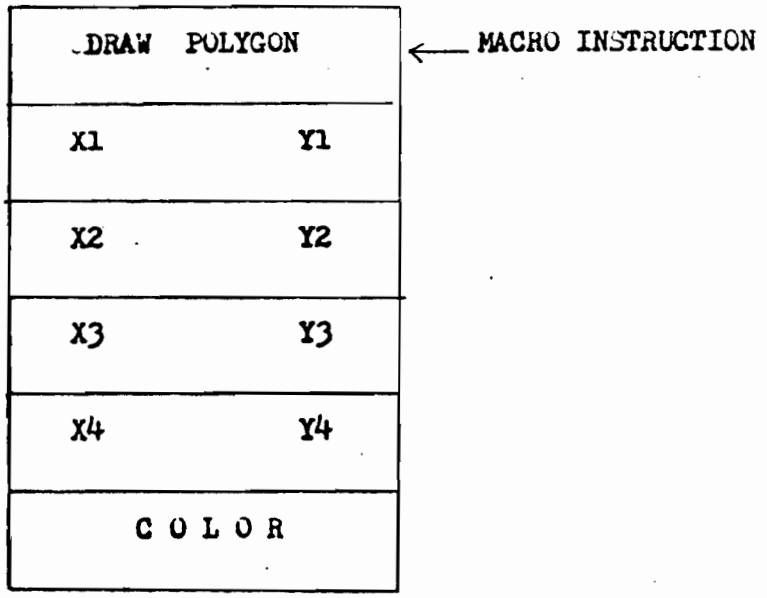
The header status word carries most of the control information to the Graphics Controller. Figure 3.2(a) shows the header status word



(a)
HEADER BLOCK



(b)
AREA TO BE SCANNED



(c)
MACRO - PROGRAM FORMAT

FIGURE 3.1
SOLID AREA MODE

HS19 — HS12 HS11-HS6 HS5 HS4 HS3 HS2 HS1 HS0

FOR FUTURE ASSIGNMENT	NO. OF PLANES INVOLVED	COLOR 1 B/W 0	RESOLUTION	OPAQ 1 TRANSP 0	MODE

(a)

FORMAT OF THE HEADER STATUS WORD (BITS HS0 - HS19)

HS0	HS1	GRAPHIC MODE
0	0	READ BACK
0	1	POINT
1	0	SHADING
1	1	SOLID AREA

(b)

HS2	WRITING MODE
0	TRANSPARENT
1	OPAQUE

(c)

HS3	HS4	RESOLUTION
0	0	LOW (256x256)
0	1	HIGH (512x512)
1	0	LOW (256x342)
1	1	HIGH (512x683)

(d)

HS5	DISPLAY MODE
0	BLACK & WHITE
1	COLOR

(e)

(b),(c),(d),(e) DETAILS OF THE STATUS WORD BITS

FIGURE 3.2

THE HEADER STATUS WORD

format. This status word encodes the type of Graphic mode to be implemented (HS0 and HS1) (see Figure 3.2(b)) as well as the selected options (HS2 - HS19). Bit HS2 controls the two writing modes of the display as shown in Figure 3.2(c). In the opaque mode the color of the current drawing overwrites that of the previous drawing while in the transparent mode, the color of the drawing mixes (in an OR operation) with that of the previous drawing to give a new color. Bits HS3 and HS4 provide information on the type of resolution used to display the image. Figure 3.2(d) shows the codes used for the various resolutions. Bit HS5 indicates whether the display is colored or uses grey scales (see Figure 3.2(e). The six bits HS6 through HS11 show the number of planes used in the frame buffer. The last eight bits of the status word (HS12 - HS19) are left unassigned at present, but can be used to implement more control functions in the future. The selection of the graphic option used is derived from suitable macro-instructions.

The second word in the header indicates the starting address in the frame buffer of the location into which the first data word is to be written. The third word specifies the color for the scanned line. The color fields specify which of the five intensity bits of each of the red, green and blue colors are selected. More details of this are presented in Chapter IV. The next word in the header gives the number of words between the start word and the end word (of the scanned line) which have to be filled with '1's'. Due to hardware considerations this count must actually be expressed as the 1's complement of the desired number.

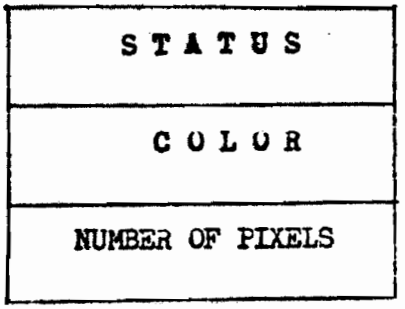
The last two words in the header are the start word and the end word. As seen in Figure 3.1(b), the start word is at the beginning of the scanned line of the bounded area and the end word is at the end of the same scanned line. Within the area, both the words are filled with '1's' and outside it with '0's'.

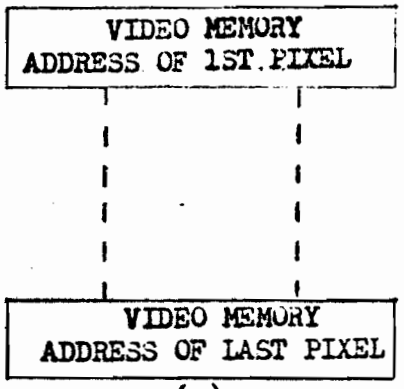
At the macro level, it is assumed that appropriate Micro-routines exist so that various Macro-instructions are available to the programmers. For example, with the appropriate Micro-routines, a Macro-instruction like "Draw Polygon" with appropriate parameters (as shown in Figure 3.1(c)) is sufficient, as far as the host computer is concerned, to specify one uniformly colored polygon.

3.2.2 The Point Mode

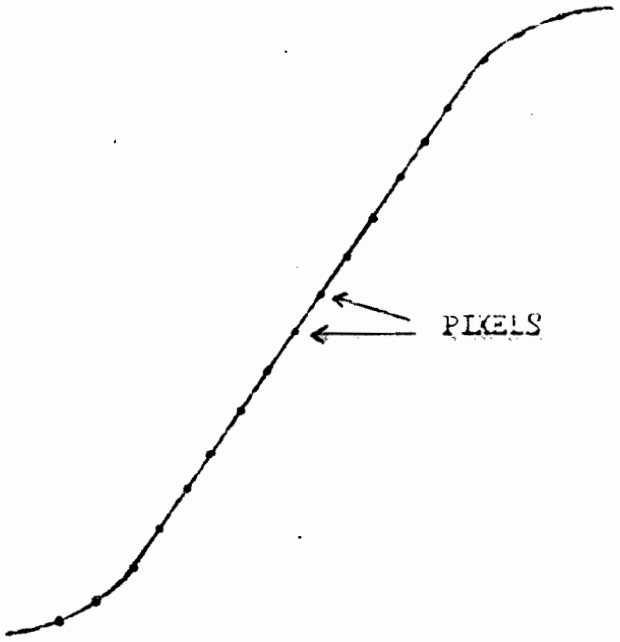
This mode draws lines by specifying the pixel location of each point on the vector or line as shown in Figure 3.3(b). Each point on the line has the same color information.

Figure 3.3(a) shows the header and the video addresses of each of the pixels on the line, as computed by the micro-processor. The status word and the color word formats have already been presented in Section 3.2.1. For the word count, the 2's complement of the total number of pixels on the screen is required by the hardware implementation. The list of video addresses following the header can randomly address points on

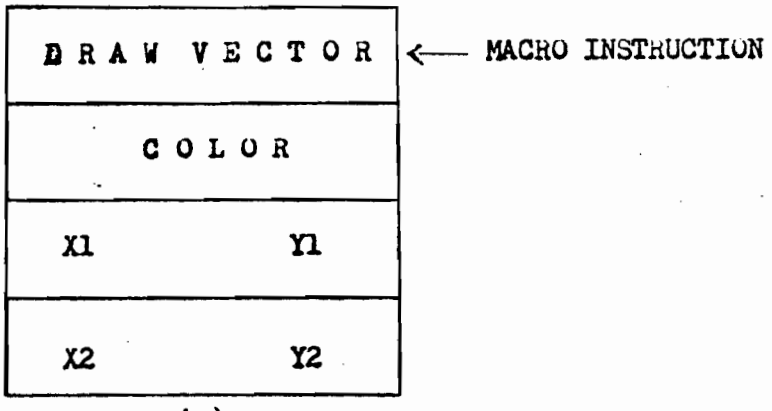




(a)
HEADER AND RELATED DATA



(b)
LINE TO BE SCANNED



(c)
MACRO-PROGRAM FORMAT

FIGURE 3.3

POINT MODE

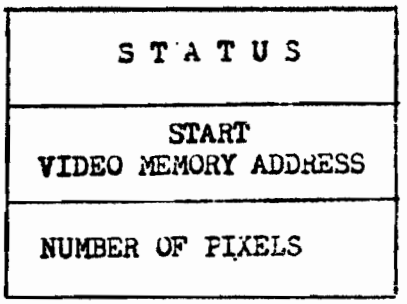
the screen. Hence the results of any number of vectors or curves of the same color can be sent to the Video memory by using this Point mode format (Figure 3.3(a)).

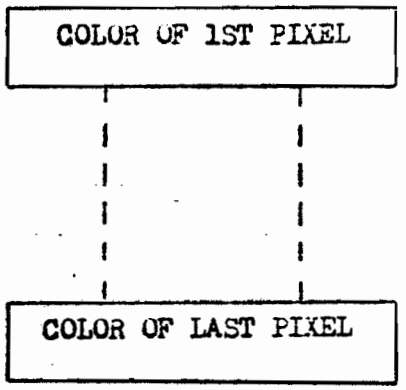
Figure 3.3 (c) presents a possible macro-instruction format for drawing a vector in a chosen color. The starting and ending coordinates of the line are given by X1, Y1 and X2, Y2 respectively.

3.2.3 The Shading Area Mode

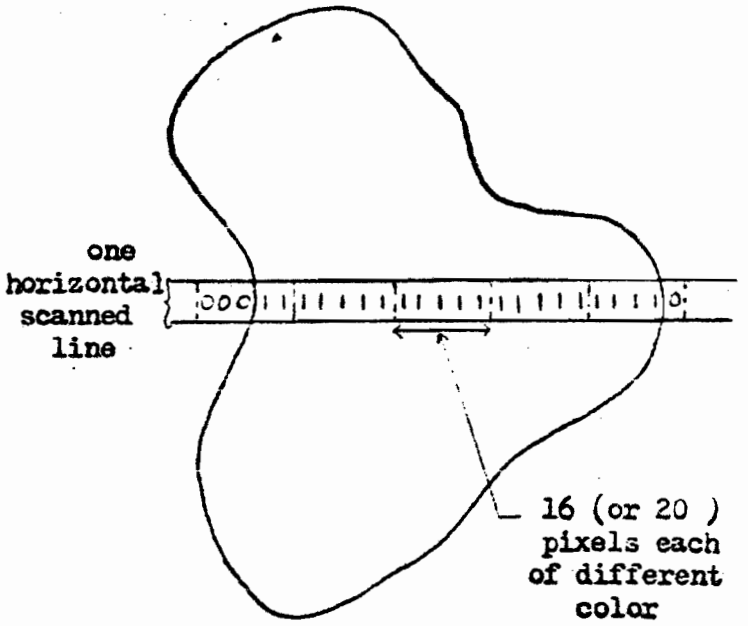
This mode is used for efficiently displaying areas of non-uniform or shaded color. As is seen in Figure 3.4(b), it horizontally scans bounded areas, one line at a time. Each pixel on the horizontal scan line may have a different color. The header and the color of each consecutive internal pixel as computed by the microprocessor are shown in Figure 3.4(a). The details of the words are similar to those explained previously for the Point mode.

Figure 3.4(c) presents a possible macro-instruction format (as furnished by the host computer) for a horizontal scan line in the Shading Area mode. Here the host computer specifies the start and end points, a starting color shade for the first (left-hand) pixel, as well as a color increment value for the pixels towards the right.

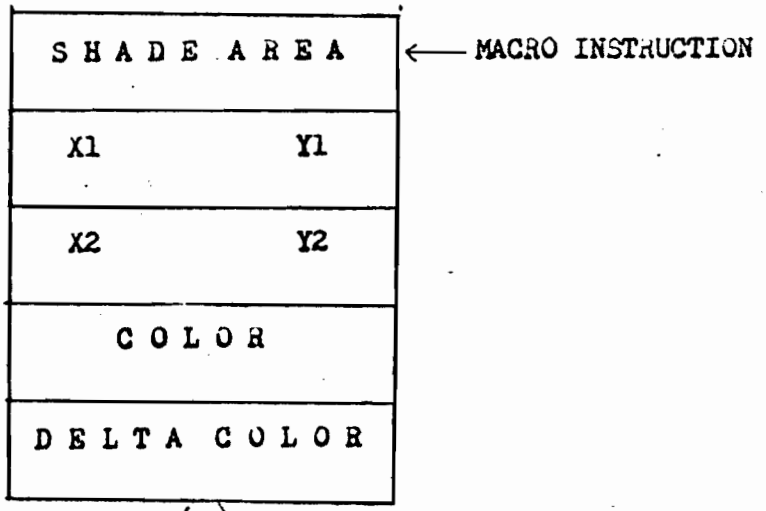




(a)
HEADER AND RELATED DATA



(b)
AREA TO BE SCANNED



(c)
MACRO-PROGRAM FORMAT

FIGURE 3.4

THE SHADING AREA MODE

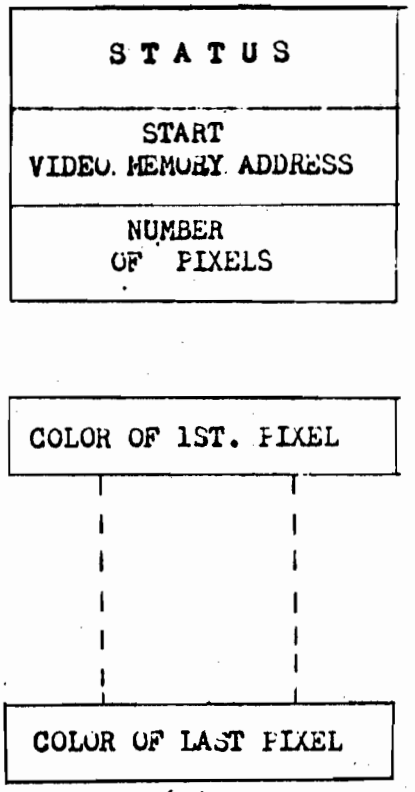
A higher level macro-instruction like "Shade Polygon" could also have been microcoded in the Microcode RAM of the 2900 series microcomputer resulting in the buffered data as shown in Figure 3.4(a).

3.2.4 The Read-Back Mode

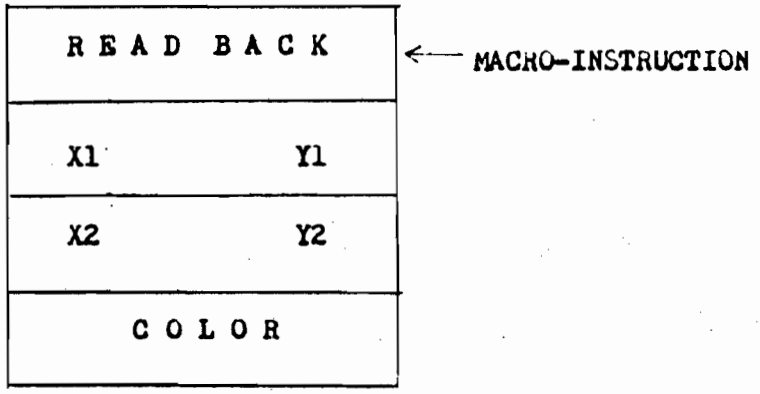
For the three modes mentioned in the previous three sub-sections, the Graphics Controller transfers data from the Local Memory of the microcomputer to the Video Memory planes. In the Read-Back mode on the other hand, the Graphics Controller makes DMA transfers from the Video memory planes to the Local memory of the microcomputer. The header block and the related data as seen in Figure 3.5(a) are essentially the same as those for the Shading Area mode except for the coding of the two least significant bits of the status word. A possible macro-instruction format for this type of mode is shown in Figure 3.5(b).

This mode serves basically two purposes:

(a) It is used for maintenance purposes. The microprocessor can use this mode to verify whether the Video memory planes and the rest of the system are functioning well or not.



(a) HEADER BLOCK AND RELATED AREA



(b) MACRO-PROGRAM FORMAT

FIGURE 3.5

READ-BACK MODE

(b) It is intended for future studies applying the paralleled microprocessors for computer image analysis. Here T.V. images will be stored into the Video memory from a T.V. camera in real-time and be read from it into the Local Memory of a microcomputer unit. The ID5000 [11] series of display system by Deanza Systems Inc. uses a similar feature for applications where enhancement and analysis are required.

CHAPTER IV

THE DISPLAY SYSTEM DESIGN

In this chapter the design of the three modules of the display system will be discussed in detail. Particular emphasis will be paid to the design of the Graphics Controller. The design of the Graphics Controller is considered in two main parts. The first part considers a variety of registers and circuits used to facilitate a proper data flow between the Micro and the Video buses. The second part considers the control circuitry which maintains the proper data flow and communication with the rest of the system.

However before explaining the design considerations of the display system, the bus structure of this system will be defined.

4.1 The Bus Structure of the Display System

Figure 4.1 displays the bus signals used by the display system modules to communicate with each other and with the rest of the GRADS system.

The following sub-sections discuss the functions of each of the buses.

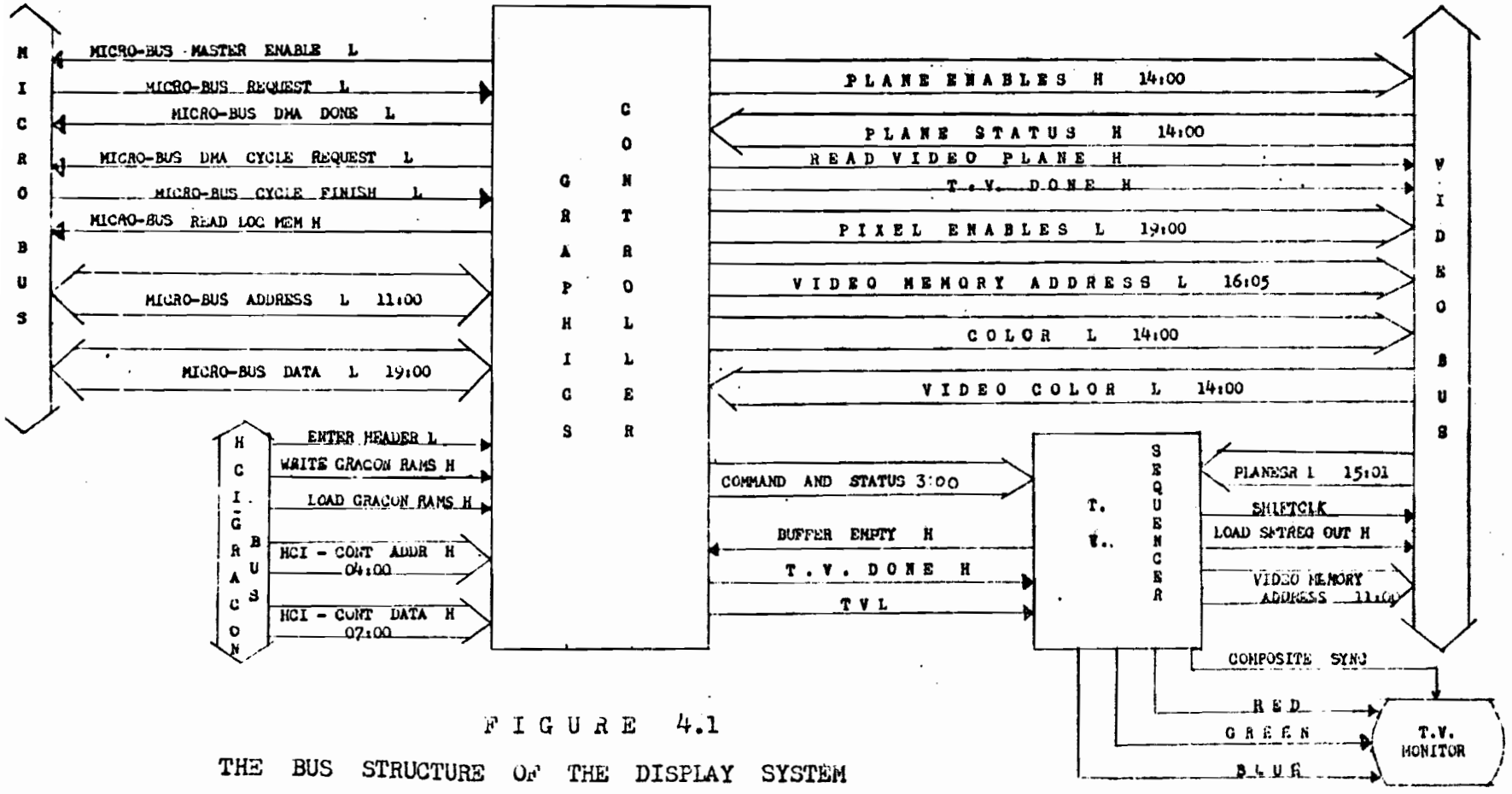


FIGURE 4.1

THE BUS STRUCTURE OF THE DISPLAY SYSTEM

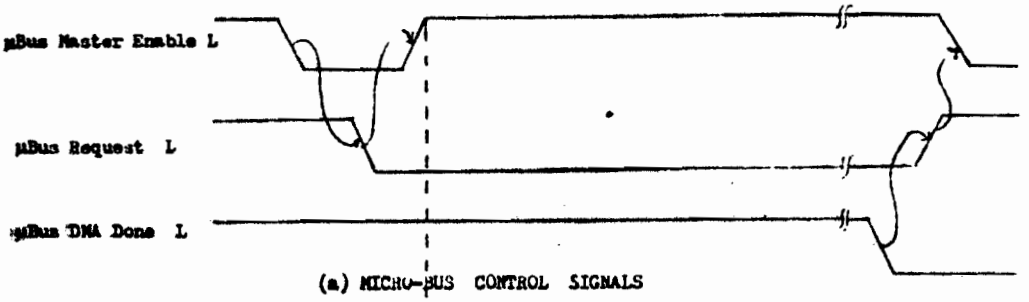
4.1.1 The Micro-bus - Graphics Controller Signals

The first three signals shown in Figure 4.1 are the Micro-bus Control Signals. They control the selection of a new microcomputer for DMA activity on the basis of a priority chain and also ensure that no other microcomputer is latched on while the involved microcomputer is busy doing a DMA transaction. Figure 4.2(a) shows the timing diagram for these control signals.

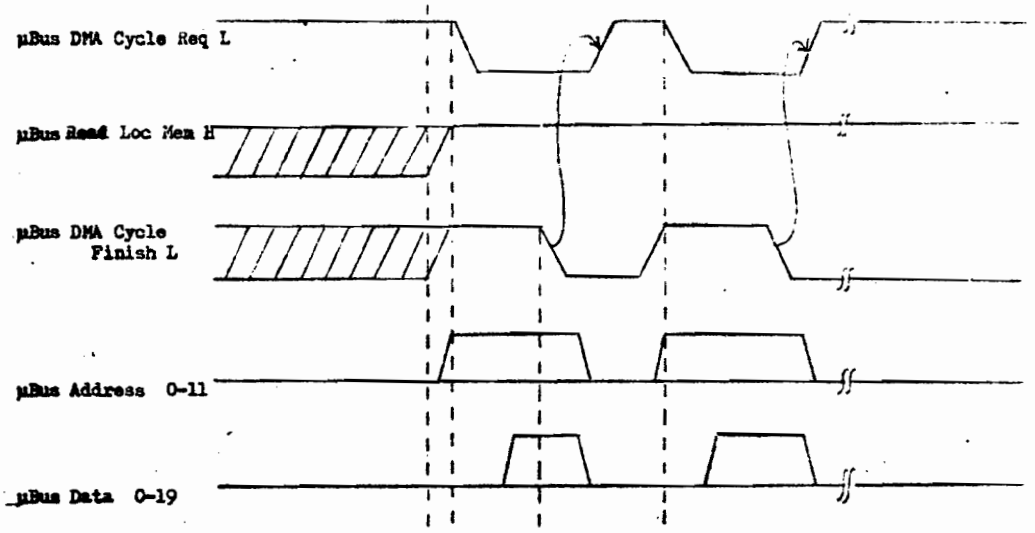
When the Graphics Controller is ready to service a new micro-computer request for DMA transfer, the MICROBUS MASTER ENABLE L signal goes low. As soon as a micro-computer is latched onto the MICROBUS MASTER ENABLE L, the MICROBUS REQUEST L signal for the latched micro-computer goes low. The latter signal requests the Graphics Controller to start the DMA transfer and also prevents other microcomputers from latching on by making the MICROBUS MASTER ENABLE L go high.

As soon as the Graphics Controller has completed the DMA transfer from the latched microcomputer, MICROBUS DMA DONE L goes low. This makes the MICROBUS REQUEST signal go high which again by hand-shaking operation makes MICROBUS MASTER ENABLE L go low, thus making the Graphics Controller ready for the next DMA transfer.

The next three signals of the Microbus (Figure 4.1), the MICROBUS DMA CYCLE REQUEST L, the READ LOC MEM H and the MICROBUS DMA CYCLE FINISH L, control the memory references made by the Graphics Con-

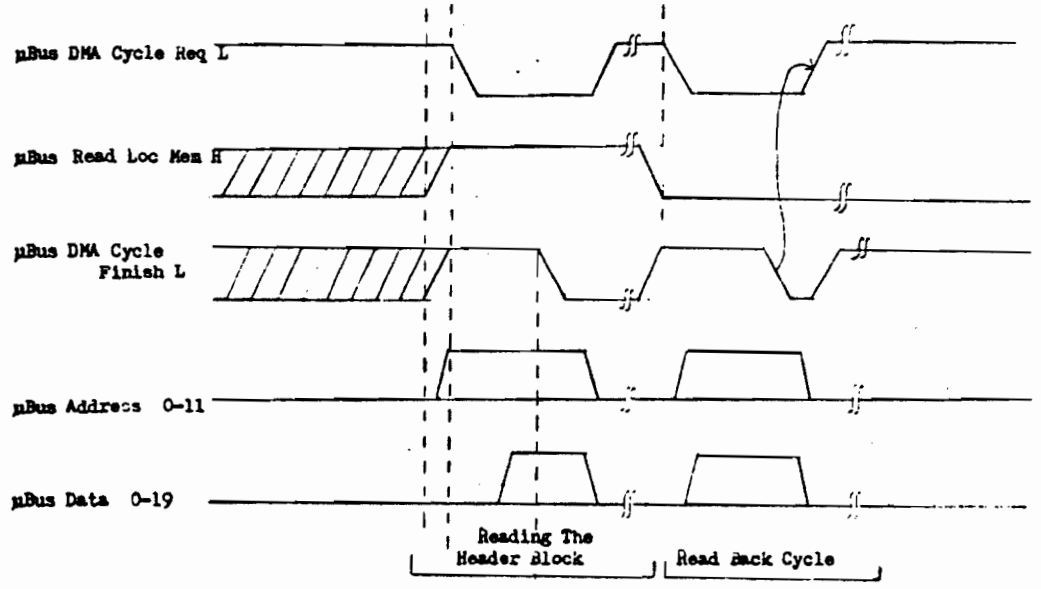


(a) MICRO-BUS CONTROL SIGNALS



(b)

(i) MICRO-COMPUTER'S MEMORY ACCESS SIGNALS DURING THE READ CYCLE FOR 3 OF THE MODES



(b)

(ii) MICRO-COMPUTER'S MEMORY ACCESS SIGNALS DURING THE WRITE CYCLE FOR THE READ-BACK MODE

FIGURE 4.2

MICRO-BUS TIMING DIAGRAM

troller to the Local memory of the latched microcomputer. Whenever the Graphics Controller wants to access the Local memory it puts the address of the word on the bus and enables MICROBUS DMA CYCLE REQUEST L signal (see Figure 4.2 (b (i))) to initiate a transaction. As soon as the data is put on the bus, the microcomputer enables the MICROBUS DMA CYCLE FINISH L signal to indicate to the Graphics Controller that the memory cycle is completed. The Graphics Controller in turn uses this signal to disable the MICROBUS DMA CYCLE REQUEST L signal.

The 20 bit data word from the Local Memory is latched when on the Microbus so that if the microprocessor makes access to its own Local Memory during this time, the data for the Graphics Controller on the bus is not lost.

The READ LOC MEM H signal indicates the direction of the DMA transaction. When high, DMA transfers occur from the Local memory to the Video memory and when low the tranfers occur in the other direction (for Read-Back mode). Figure 4.2 (b (ii)) shows the later case. For the first three cycles the header is read from the Local Memory and on interpreting the header to be a Read-back, the Graphics Controller disables the READ LOC MEM H signal.

4.1.2 The Host Computer Interface - Graphics Controller Bus (HCI - GRACON Bus)

Since the control circuitry of the Graphics Controller uses a ROM Sequencer philosophy, it is necessary to initially program the RAMs of this DMA control circuitry which is divided into two - the Header machine and the Transaction machine. The Host Computer Interface is responsible for initially programming the control sequences into these RAMs. The ENTER HEADER L signal is responsible for first writing (when low) into the Header machine and then (when high) into the Transaction machine of the Graphics Controller. As the RAMs of both the machines can be addressed consecutively, this signal can be the most significant bit of the address lines.

The WRITE GRACON RAMs signal is nanded with the ENTER HEADER L signal to either write (or read) into the RAMs of the Header machine or those of the Transaction machine.

The LOAD GRACON RAMs H signal enables the loading of the RAMs from the Host Computer Interface, when high. When the Graphics Controller is in its normal operating mode, this signal has to be low.

Figure 4.3 gives the timing diagram for these signals.

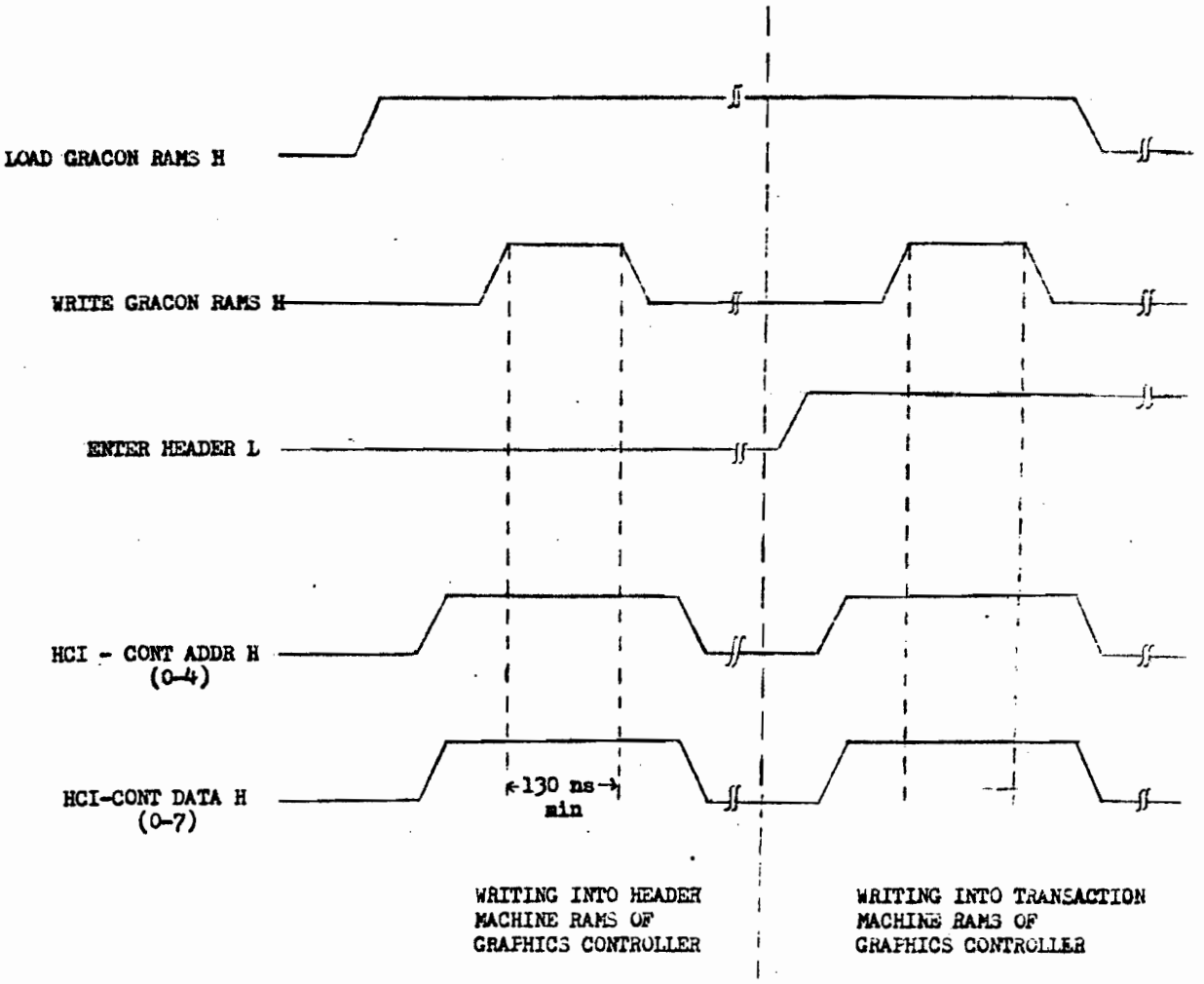


FIGURE 4.3

THE HCI - GRACON BUS SIGNALS

4.1.3 The Video Bus - Graphics Controller Signals

The PLANE ENABLE signals initiate the memory transactions between the Graphics Controller and the Video Memory planes. The PLANE ENABLES are derived from the color information by enabling the writing into the corresponding planes.

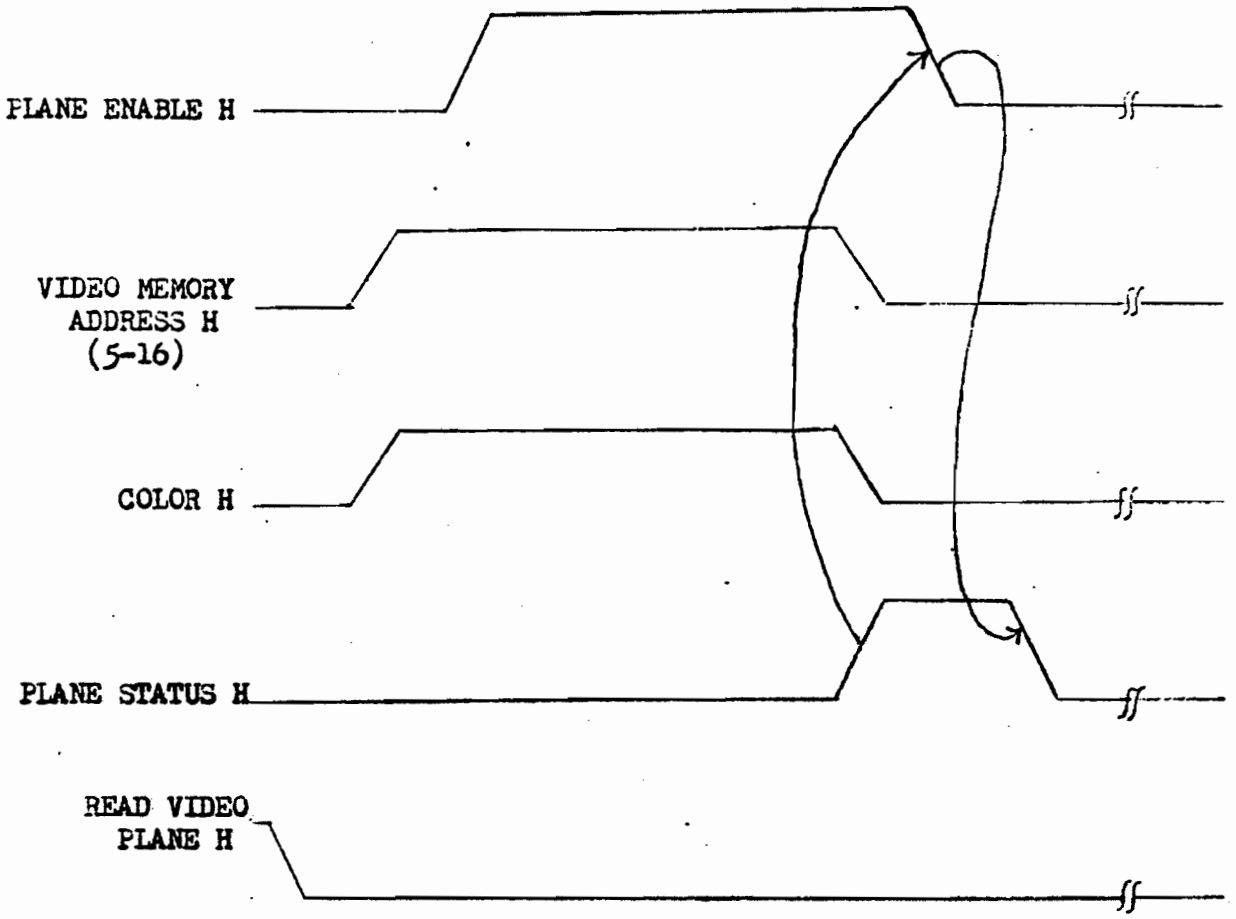
The Video memory plane indicates the cycle completion by asserting PLANE STATUS H. This disables the PLANE ENABLES which in turn disables the PLANE STATUS (see Figure 4.4).

The READ VIDEO PLANE H signal goes to the Write Enable line for all the chips on the plane. A low indicates a write operation and a high a read operation.

The T.V. DONE signal, generated by the Graphics Controller, is used by the Video memory planes to clock data into the T.V. Sequencer's buffer situated at the Video memory planes.

The Video Memory Address lines address one word on each plane and the PIXEL ENABLES, address a required bit in the word (by connecting to the chip select inputs of the memory chips on the plane).

The COLOR lines constitute the pixel data to be stored in the Video memory plane. When the system is in Read-Back mode, the pixel information is read from the Video memory planes and passed to the Graphics Controller over the Video Color lines.



F I G U R E 4.4

THE VIDEO MEMORY WRITE CYCLE

Figure 4.4 illustrates the Video memory access signal timing diagram.

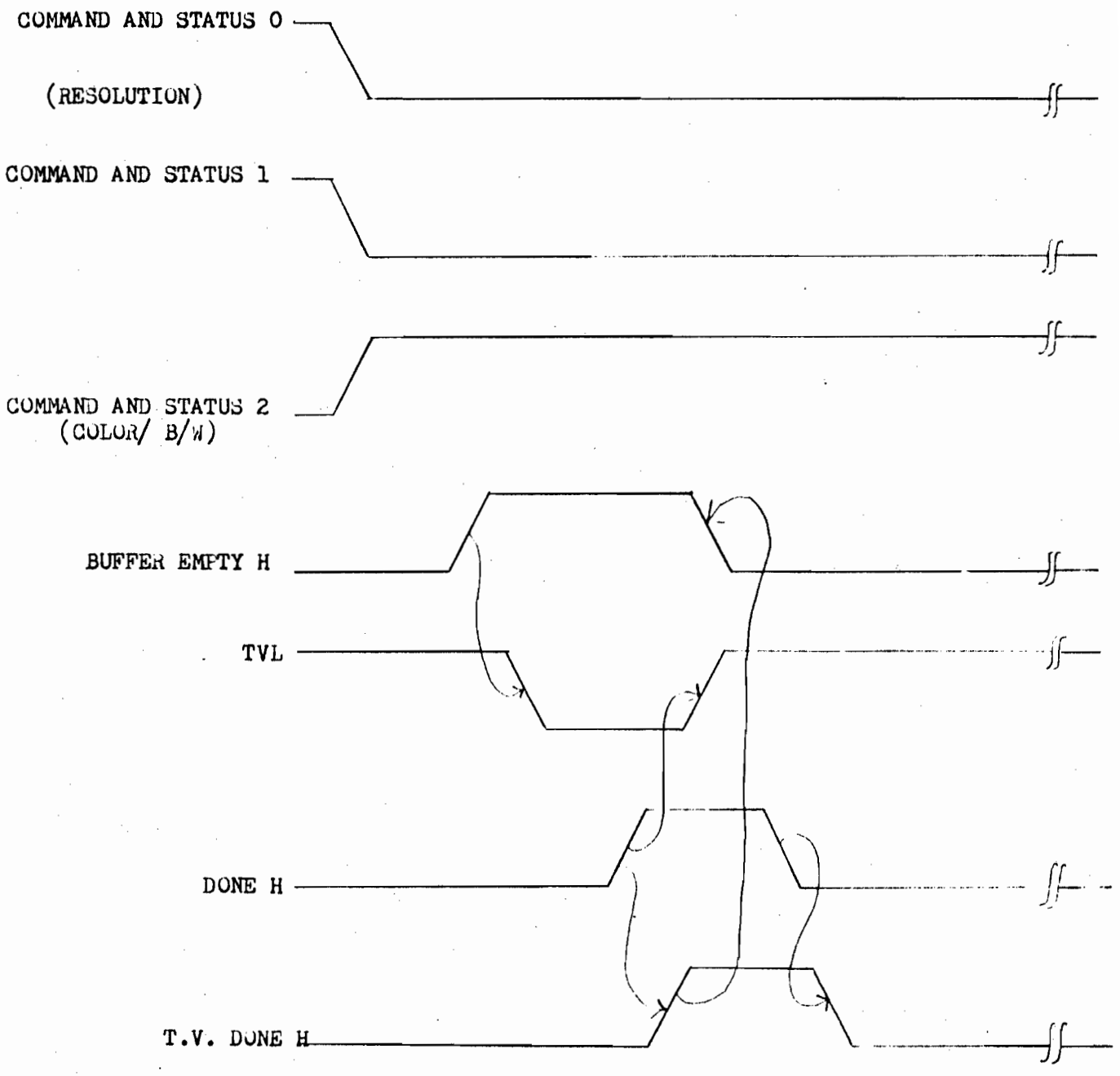
4.1.4 The Graphics Controller - T.V. Sequencer Signals

Three signals which inform the T.V. Sequencer, the display resolution and whether the image is in color or shades of gray, are sent by the Graphics Controller to the T.V. Sequencer over the Command and Status lines.

As soon as the T.V. Sequencer has loaded the information in its buffer into its shift registers (at the Video memory), it asks the Graphics Controller to fill the buffer with new data by asserting BUFFER EMPTY H signal. The Graphics Controller on receiving this signal, allows the T.V. Sequencer to address the Video Memory by asserting the T V L signal.

After the Graphics Controller has granted the T.V. Sequencer's request, it asserts the signal T.V. DONE H which is used by the T.V. Sequencer to increment its address register.

Figure 4.5 shows the timing diagram of these signals for low resolution color mode.



F I G U R E 4.5

THE GRAPHICS CONTROLLER - T.V. SEQUENCER SIGNALS
FOR LOW RESOLUTION COLOR MODE

4.1.5 The T.V. Sequencer - Video Bus Signals

The lines labelled PLANESRL (in Figure 4.1) provide the serial data from the shift registers on the Video memory planes to the T.V. Sequencer. This digital data is converted to the RED, GREEN and BLUE analog signals by the T.V. Sequencer. Before being converted to analog signals, the PLANESRL signals are grouped by a selector network depending on the mode of the display system. The three digital to analog converters (used in the T.V. Sequencer), one per color, accept five parallel bits of digital data, each allowing up to 32 different intensity levels.

The SHIFTCLK line is used by the T.V. Sequencer to clock the shift registers on the Video memory planes. The LOAD SFTREG OUT H line is asserted by the T.V. Sequencer to parallel load the new word into the shift register of the double buffer arrangement.

4.2 Data Flow

In this section, the various registers and circuits used by the Graphics Controller, to facilitate a proper data flow between the Micro and the Video buses, will be considered. Figure 4.6 shows the registers and circuits used and that will be considered in the following sub-sections. References [54], [62], [63] (which are data handbooks) were extensively used.

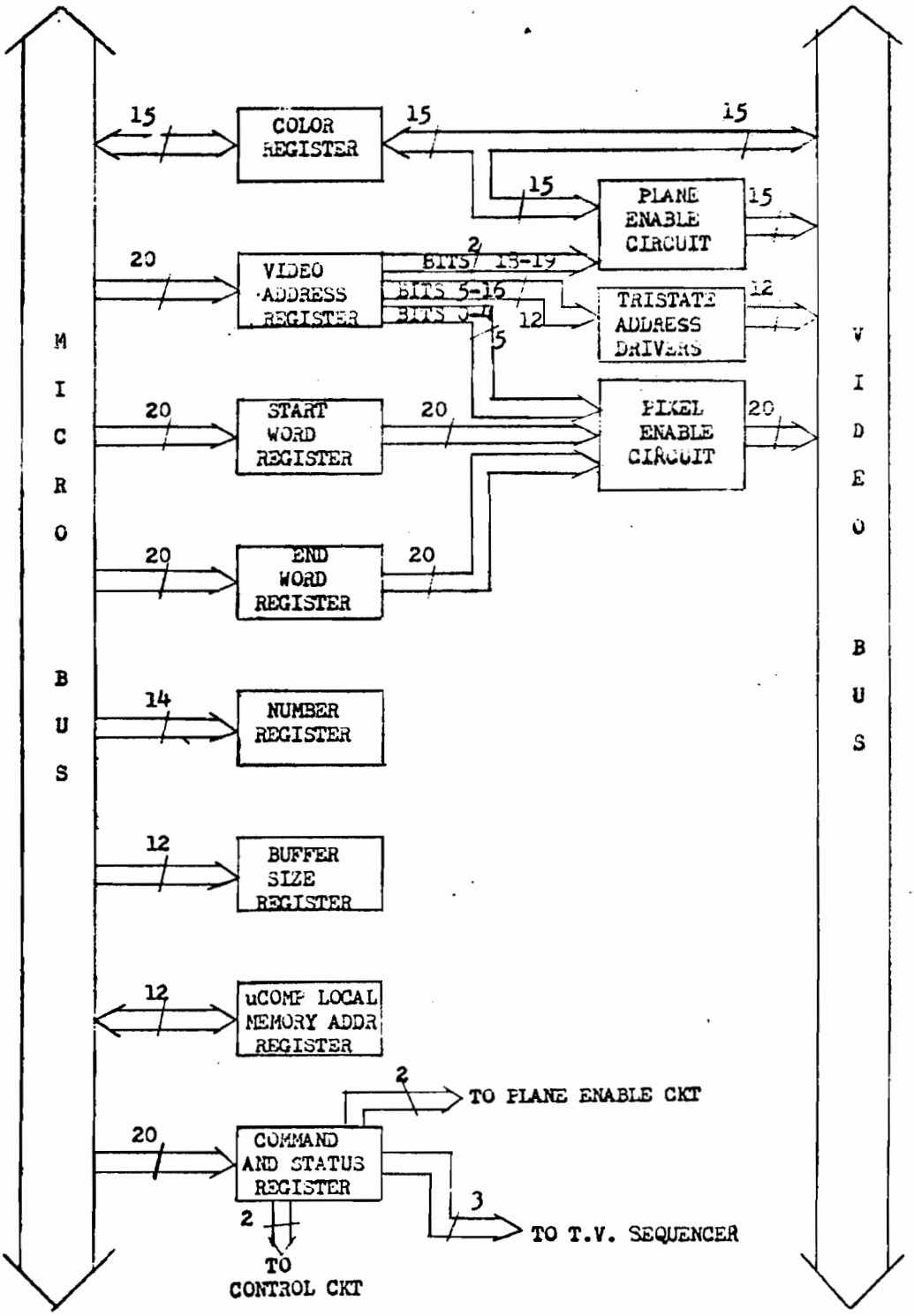


FIGURE 4.6

DATA FLOW HANDLED BY THE GRAPHICS CONTROLLER

4.2.1 The Color Register

This bi-directional register transfers 15 bit color words to and from the Video Memory and the microcomputer's Local Memory. It also transfers 15 bit color word information to the Plane Enable circuit as can be seen in Figure 4.6. The Plane Enable circuit is responsible for properly enabling the required Video Memory planes as indicated in Table 4.1.

In the low resolution color mode, there is one plane associated with each color bit. Hence with only 3 memory planes only one bit can be obtained per primary color and with the full 15 memory planes, 5 bits (giving 32 intensity levels) can be achieved for each of the three primary colors. The sequence in which the planes are selected is shown in Table 4.1. In the low resolution black and white mode, each plane gives the same intensity information for each of the three primary colors, to give one shade of grey level. Hence if the display has 32 intensities of grey, 5 memory planes are sufficient (as illustrated in Table 4.1).

Both of the high resolution modes require 4 planes for each of the five bits of the three primary colors. Each plane therefore contributes for only a quarter of the picture. The sequence in which the planes are selected can again be seen in the table. The high resolution color mode requires 12 memory planes to give one bit of each of the three primary colors. Hence in a 15 plane configuration, planes 13

SHADE / MODE	BIT 0 RED 0	RED 1	RED 2	RED 3	RED 4	GREEN 0	GREEN 1	GREEN 2	GREEN 3	GREEN 4	BLUE 0	BLUE 1	BLUE 2	BLUE 3	BIT14 BLUE 4
LOW RESOLUTION COLOR MODE	P1	P4	P7	P10	P13	P2	P5	P8	P11	P14	P3	P6	P9	P12	P15
LOW RESOLUTION BLACK & WHITE MODE	P1	P2	P3	P4	P5	P1	P2	P3	P4	P5	P1	P2	P3	P4	P5
HIGH *RESOLUTION COLOR MODE	P1	P13	P25	P37	P49	P5	P17	P29	P41	P53	P9	P21	P33	P45	P57
	P2	P14	P26	P38	P50	P6	P18	P30	P42	P54	P10	P22	P34	P46	P58
	P3	P15	P27	P39	P51	P7	P19	P31	P43	P55	P11	P23	P35	P47	P59
	P4	P16	P28	P40	P52	P8	P20	P32	P44	P56	P12	P24	P36	P48	P60
HIGH *RESOLUTION BLACK & WHITE MODE	P1	P5	P9	P13	P17	P1	P5	P9	P13	P17	P1	P5	P9	P13	P17
	P2	P6	P10	P14	P18	P2	P6	P10	P14	P18	P2	P6	P10	P14	P18
	P3	P7	P11	P15	P19	P3	P7	P11	P15	P19	P3	P7	P11	P15	P19
	P4	P8	P12	P16	P20	P4	P8	P12	P16	P20	P4	P8	P12	P16	P20

*High resolution requires 4 planes per color bit

T A B L E 4.1

TABLE ILLUSTRATING THE MEMORY PLANES ENABLED DURING THE DIFFERENT MODES

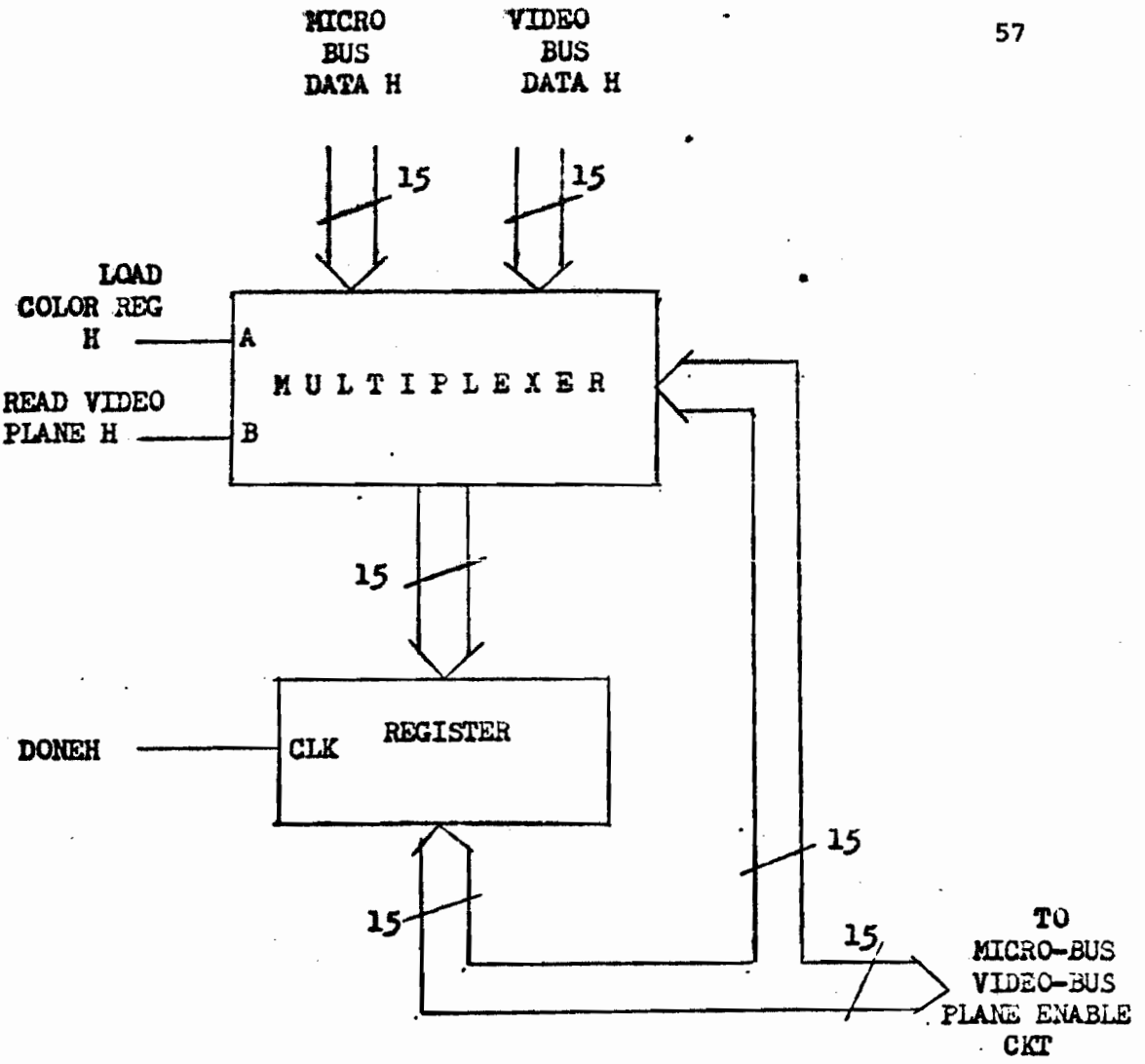
through 15 are ignored. If 32 levels of intensities are required for each of the three primary colors, the system would need 60 Video memory planes as shown in Table 4.1. For the high resolution black and white mode, the same 4 planes are enabled for each intensity bit of the three colors. Hence with 12 planes, 3 intensity bits give 8 grey levels. For providing the 32 grey levels, 20 planes would be needed as is shown in Table 4.1.

The circuit used to implement the Color Register is illustrated in Figure 4.7.

For the Solid Area mode and the Point mode, the color word is first loaded into the color register from the Microbus and subsequently maintained (by recycling) throughout the block transaction. The recycling is not done for the Shading Area mode since the color information for each pixel is different. For the Read-back mode, the pixel color from the Video bus is latched in the color register for the Micro bus write cycle. This also results in an improved access time for the T.V. Sequencer reading of the Video frame memory.

4.2.2 The Video Address Register

The function of this register is to specify the location in the Video memory plane where the data from the Color register is to be stored. As can be seen in Figure 4.6, the output lines of the register form three



(a) COLOR REGISTER IMPLEMENTATION

LOAD COLOR REG H	READ VIDEO PLANE H	DATA MULTIPLEXED
0	0	PREVIOUS STORED COLOR
0	1	PREVIOUS STORED COLOR
1	0	MICRO-BUS DATA H
1	1	VIDEO-BUS DATA H

(b) TRUTH TABLE FOR THE COLOR MULTIPLEXER

FIGURE 4.7

COLOR REGISTER

groups. The two most significant lines carry the bits responsible for identifying one of the 4 quadrants on the screen (in the high resolution mode) and are routed to the Plane Enable circuit. The next 12 lines are responsible for providing the word addresses for the color words to be stored in the Video Memory. The last 5 lines go to the Pixel Enable circuit where they permit the circuit to enable the required bits of the addressed word.

The hardware implementation of this register is illustrated in Figure 4.8. In the Point, Shading Area and the Readback modes, bits 0-4 of the Video Address register are used to select a single bit of the memory plane word being addressed by the Plane Enables and the Video Memory Address bits 5 - 19. A word size selection logic network accommodates Video memory word sizes of 8, 12, 16 or 20 bits per word.

For the Solid Area mode, the Video memory plane processes whole words and the bits 0-4 of the register which go to the Pixel Enable circuit, are not used.

4.2.3 Start and End Word Registers

The Start and End word registers are used only for the Solid Area mode. These two registers are used to buffer the start word and end word of the header sequence in the Solid Area mode. This permits the writing operation based on a word format rather than pixels, with an obvious

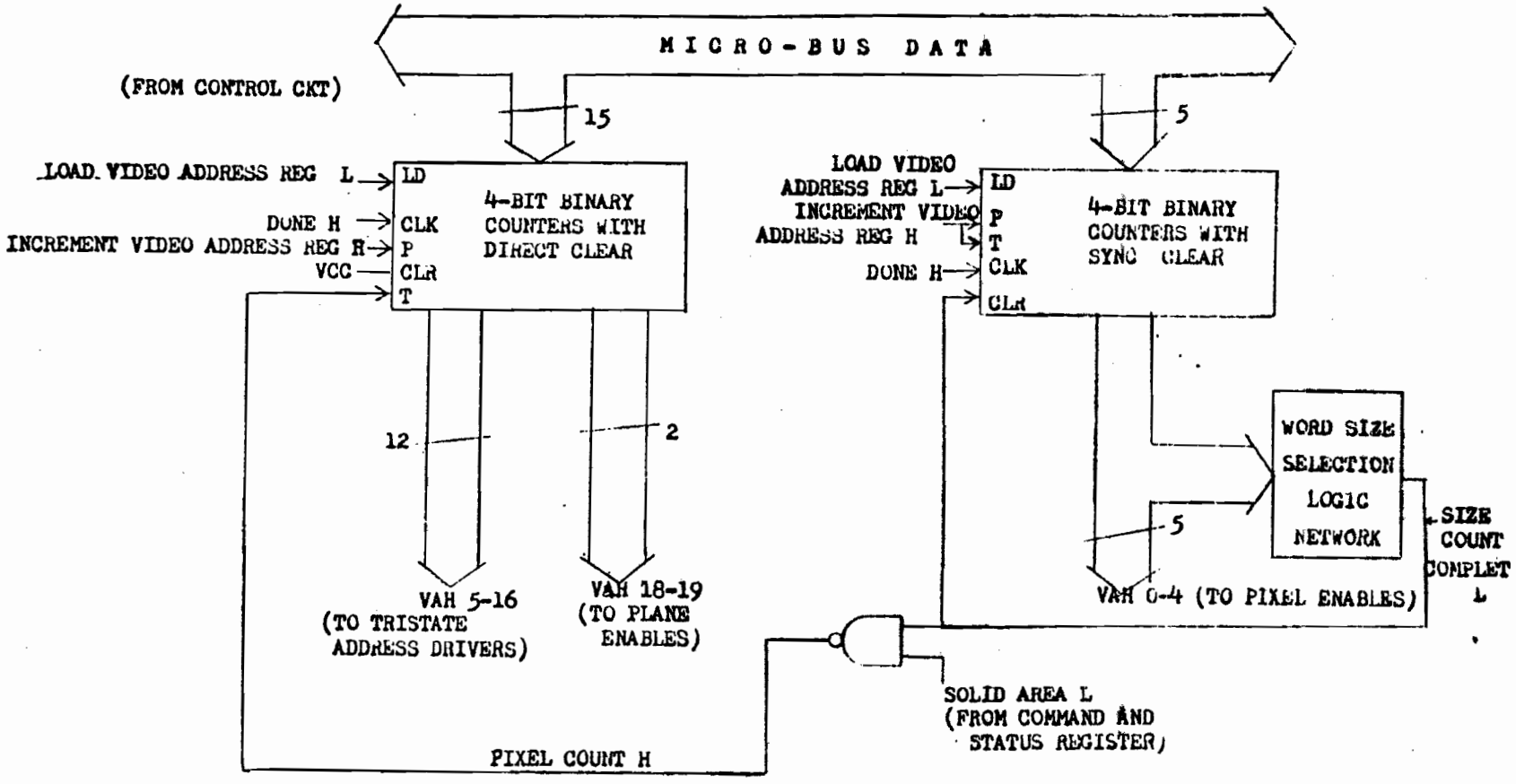


FIGURE 4.8

THE VIDEO ADDRESS REGISTER

increase in performance. The realization of these registers is shown in Figure 4.9.

4.2.4 The Plane Enable Circuit

This circuit generates the Plane Enable signals fed to the Video bus to select the planes involved in a transaction. Each plane represents one level of intensity or chroma information. For T.V. Sequencer reads, all the planes must be enabled so as to read information from every plane in the Video frame memory. In low resolution opaque mode, all the planes must be enabled so as to access the information in every plane. For the transparent mode it is desirable to write only into those chips for which the color bit is enabled. Further in the high resolution mode, 4 planes are required for each bit of intensity of color. Here each plane represents 1/4 of the T.V. screen. The quarter which should be enabled is specified by the two highest order bits of the Video Address register.

Figure 4.10 illustrates the hardware implementation of the Plane Enable circuit as well as the tabulations of the relevant control signals.

The CHIP ENABLE signal is generated by the timing circuitry of the Graphics Controller, as a strobe pulse to achieve correct timing of the Plane Enable waveforms.

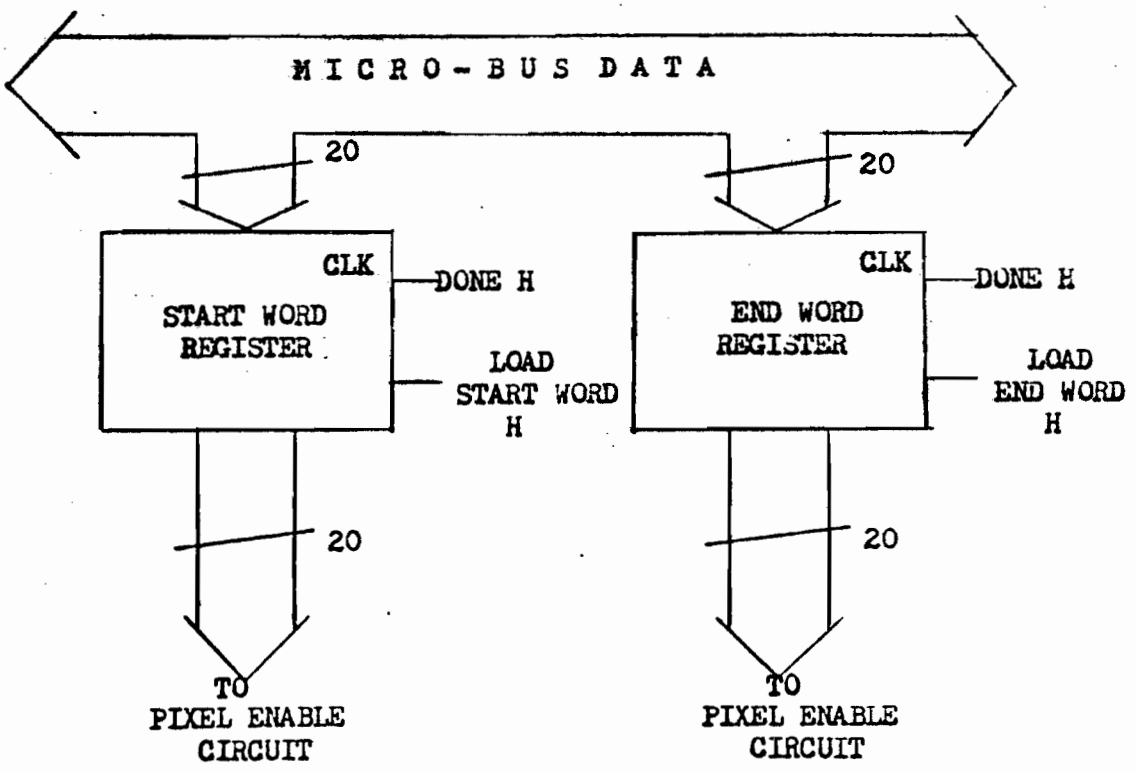
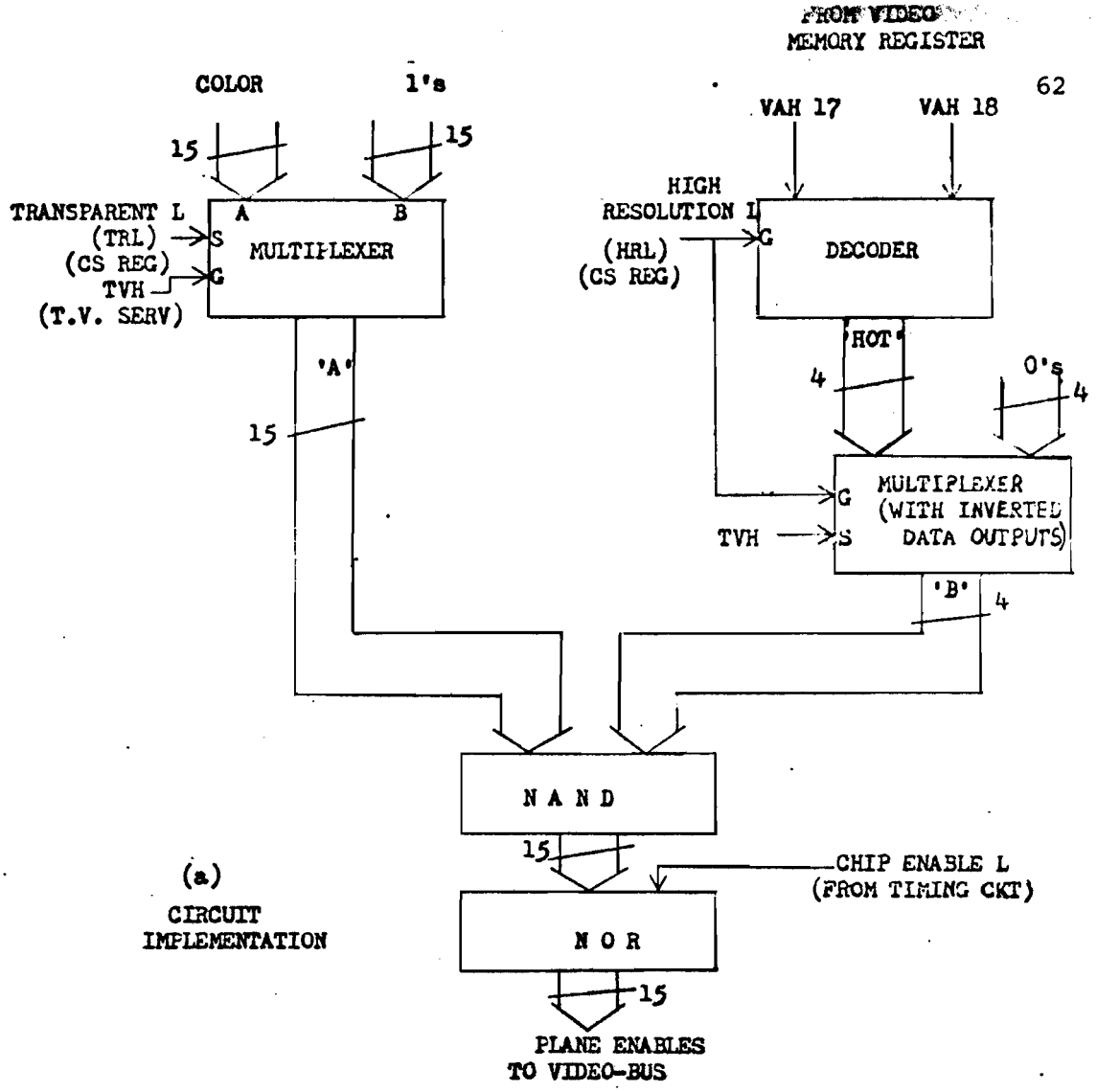


FIGURE 4.9

START AND END WORD
REGISTERS



(a) CIRCUIT IMPLEMENTATION

TVH	TRL	OUTPUT 'A'
0	0	COLOR
0	1	1's
1	0	0's
1	1	0's

(b) MULTIPLEXER

TVH	HRL	OUTPUT 'B'
0	0	'HOT'
0	1	1's
1	0	1's
1	1	1's

(c) MULTIPLEXER (WITH INVERTED OUTPUTS)

			OUTPUTS 'HOT'			
VAH 17	VAH 18	HRL	Y0	Y1	Y2	Y3
0	0	0	0	1	1	1
0	0	1	1	1	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	1
1	0	0	1	0	1	1
1	0	1	1	1	1	1
1	1	0	1	1	1	0
1	1	1	1	1	1	1

(d) DECODER

(b),(c),(d) THE TRUTH TABLES

FIGURE 4.10

PLANE ENABLE CIRCUIT

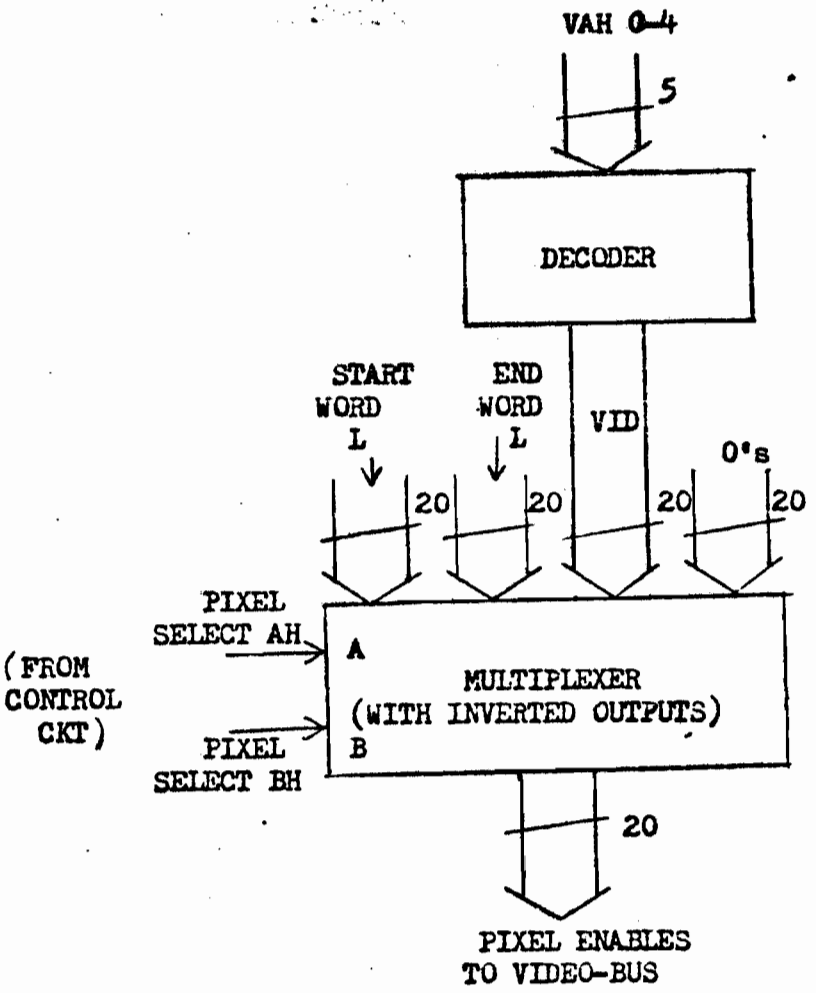
4.2.5 The Pixel Enable Circuit

The Pixel Enable signals are used to select a particular bit of a given Video memory word for the Point, Shading Area and Read-Back modes. In the Solid Area mode, all the bits of a word are accessed simultaneously. As seen in Figure 4.11, for the Solid Area mode, first the Start word is put onto the Pixel Enable lines to enable all the pixels which are part of the area and to disable those which are not. For every intermediate word, all the Pixel Enables are asserted. Finally the End word is multiplexed onto the Pixel Enable lines to define the end boundary of the area.

For the other modes . the decoded five lowest order bits from the Video Address Register specify the bits to be enabled in the Video memory plane.

4.2.6 The Number Register

This register is a counter that keeps a tally for the Graphics Controller on all transactions made with the Video memory planes for DMA purposes (T.V. Sequencer reads are not counted). As was mentioned in Chapter III, each header specifies the number of memory references required. For the Line, Shading Area and Read-back modes, the two's complement of the number of pixels involved is the third and final word in the header. For the Solid Area mode, the one's complement of the number of interior words



(a) CIRCUIT IMPLEMENTATION

PIXEL SELECT BH	PIXEL SELECT AH	OUTPUT
0	0	VID
0	1	1's
1	0	START WORD L
1	1	END WORD L

(b) THE TRUTH TABLE FOR THE MULTIPLEXER

F I G U R E 4.11

T H E P I X E L E N A B L E C I R C U I T

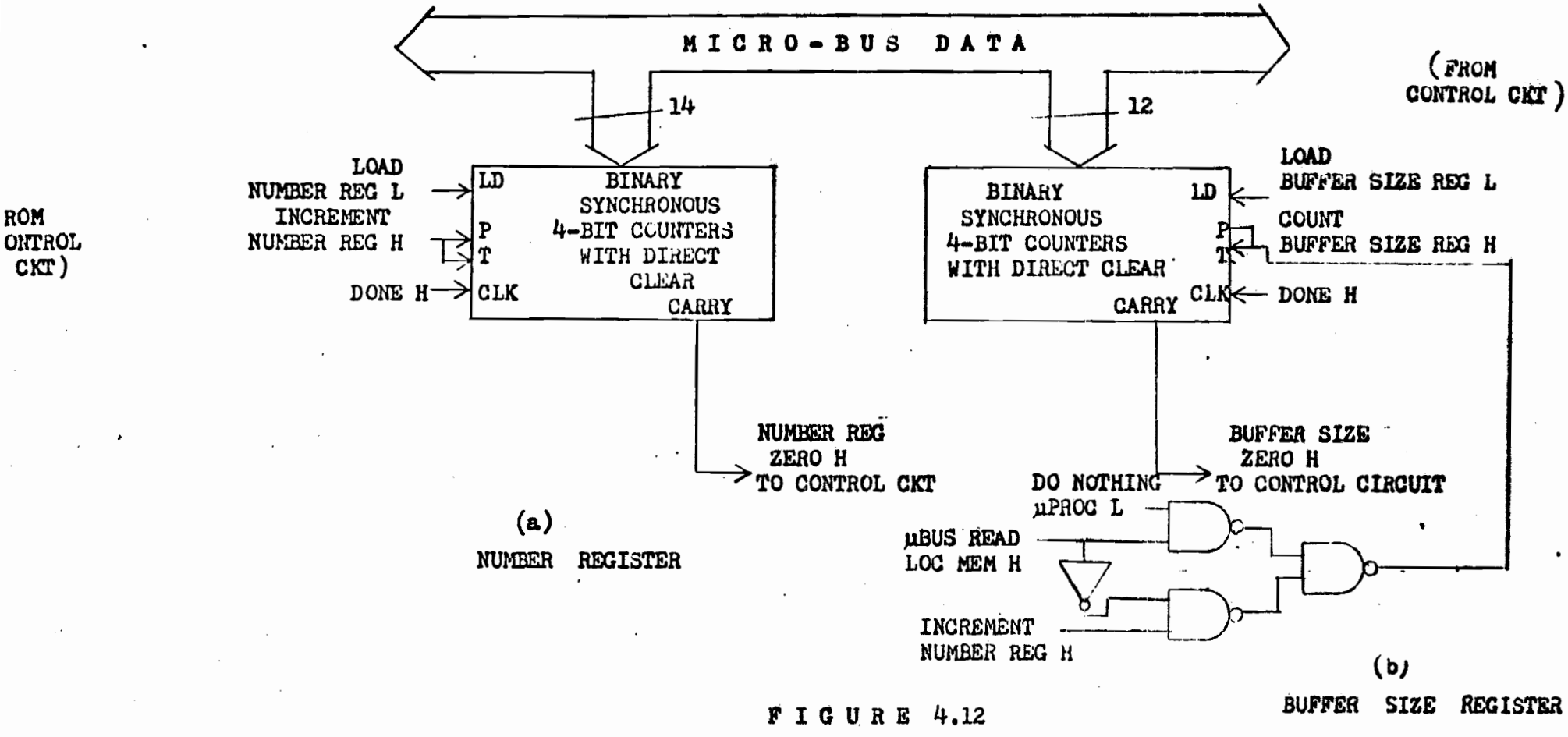


FIGURE 4.12

THE NUMBER AND BUFFER SIZE REGISTERS

is the fourth word in the header. Figure 4.12 illustrates the Number Register and the control signals to it.

4.2.7 The Buffer Size Register

The Graphics Controller determines when it has completed the processing of a microcomputer's buffer by using the Buffer Size register to count the number of memory references made by it to the Local memory of the microcomputer.

The first location of every buffer in the Local memory contains the two's complement of the number of locations in the buffer (including the first one). This number is loaded into the Buffer Size register in the right state of the Header machine of the Graphics Controller. Figure 4.12 illustrates the Buffer Size register with its Control signals.

4.2.8 The Microcomputer Local Memory Address Register

A feature of the GRADS is that the microprocessor may place the buffer with the header and related data information anywhere within the Local memory provided it stores a pointer to it in location 0000_8 of the Local memory.

Initially the Microcomputer Local Memory Address register is cleared and hence location 0000_8 is accessed. The pointer stored in this location is then loaded into the register which then acts as a program counter for the display system until the DMA block transfer is over. Figure 4.13(b) shows this register with its control signals.

4.2.9 The Command and Status Register

The microprocessor's buffer may contain more than one set of header and data streams, and each set may require a different mode of DMA transfer. To identify the type of transfer mode required, each header has a status word (whose details were given in Chapter III) which specifies the mode of operation for the Graphics Controller and the T.V. Sequencer. The Header machine loads this status word into the Command and Status register when required. Figure 4.13(a) illustrates the hardware implementation of this register.

4.2.10 The Tri-state Address Drivers

These drivers, drive the Video Address lines for the Video memory under the control of the TVL signal. Similar address drivers exist on the T.V. Sequencer controlled by the same signal. The T.V. Sequencer gets a priority to drive the address lines on the Video bus, over those generated by the Graphics Controller for DMA transfer.

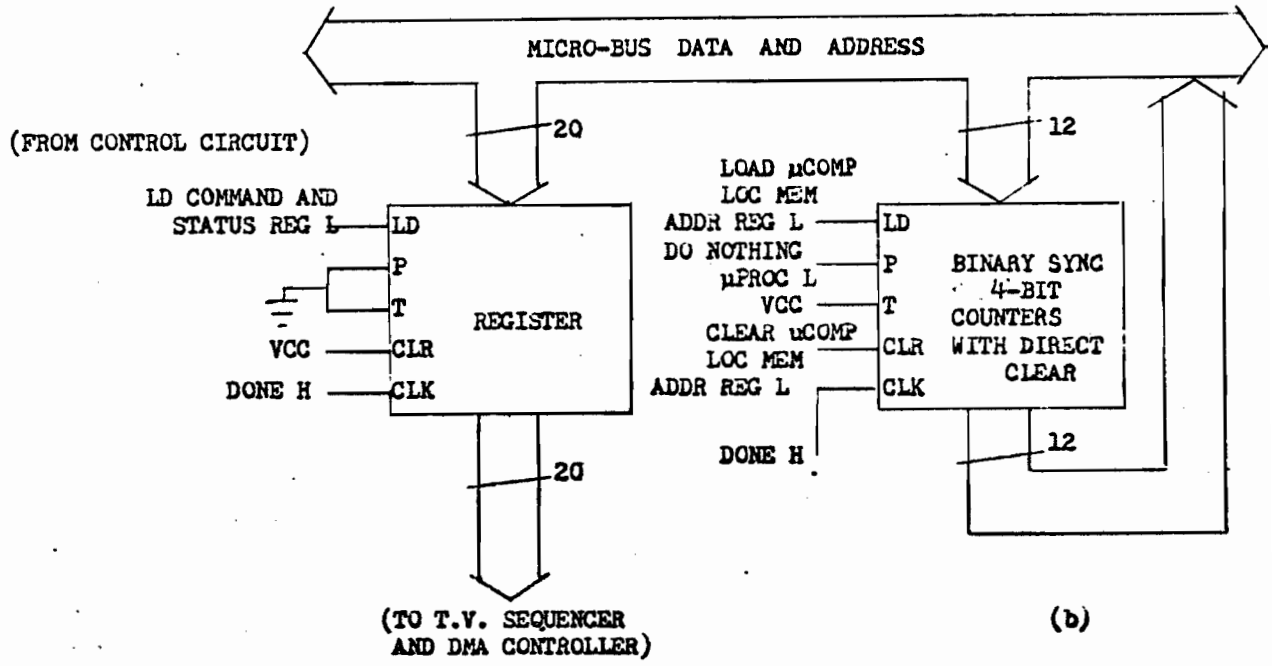


FIGURE 4.13

(a) THE COMMAND AND STATUS REGISTER

(b) THE MICRO-COMPUTER LOCAL MEMORY ADDRESS REGISTER

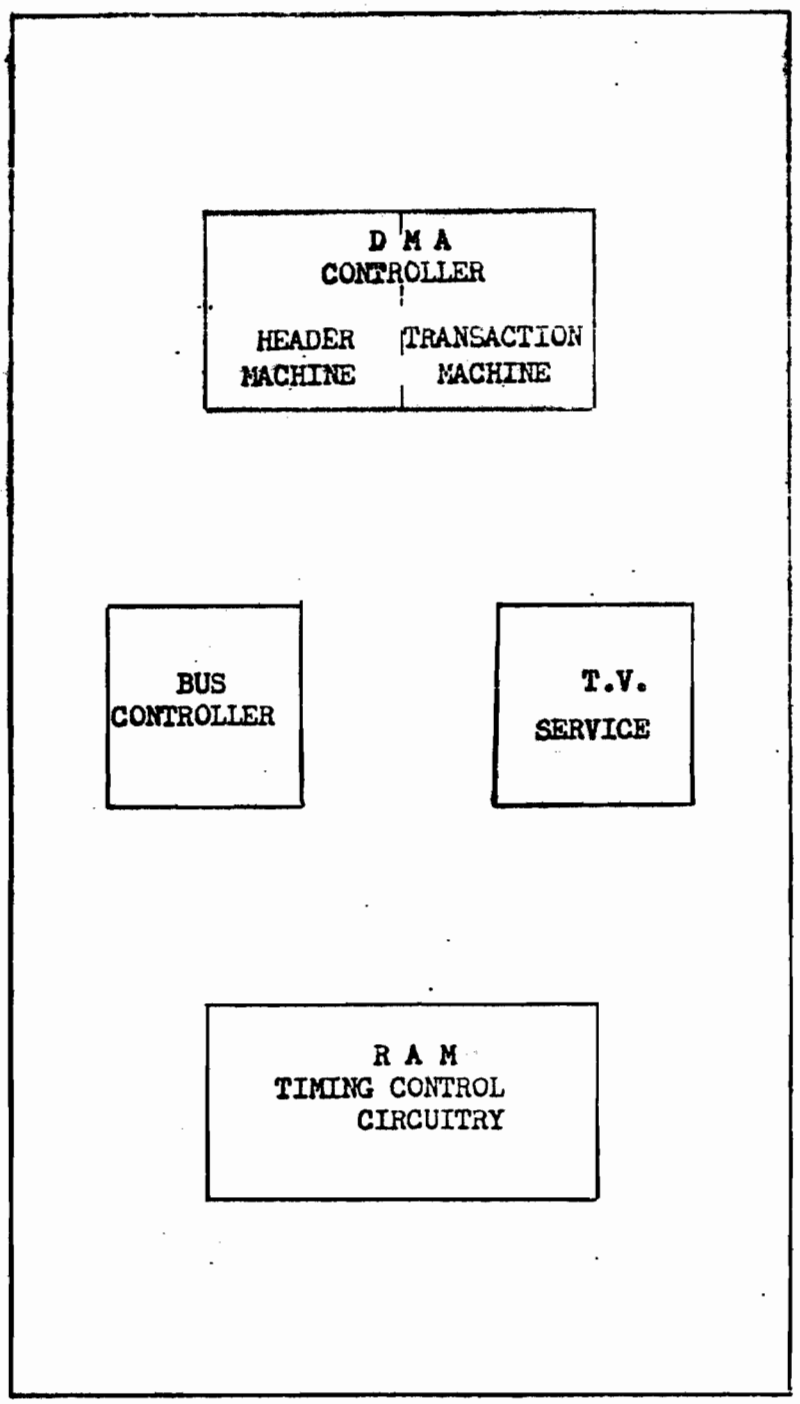
4.3 The Bus Arbitration and Information Interpreting Control Circuitry

In this section the circuitry that generates the control signals necessary to achieve the required communication and data-flow described previously, will be described in detail.

For convenience in analysis, the control circuits of the Graphics Controller can be segmented into different functional blocks. Figure 4.14 illustrates this segmentation. In the following sub-sections, the analytic description and the hardware realization of each of the control blocks illustrated, will be discussed.

4.3.1 The Direct Memory Access Controller (DMAC)

The DMAC is responsible for the control of DMA block transfers between the Local memory of the microcomputer and the Video bus. The transfer is made transparent to the microprocessor by a cycle stealing mechanism. Also, it obtains direct access to the Video frame memory by a process which is transparent to the T.V. Sequencer. Further, as in many DMAC's, it interprets the data stream from the microcomputer's Local memory and generates a new stream for the Video memory. It can accomplish this by using the instructions in a header block, which are stored in special purpose registers at the beginning of the DMA transfer.



F I G U R E 4.14

THE BLOCK DIAGRAM ILLUSTRATION OF
THE CONTROL CIRCUITS OF
THE GRAPHICS CONTROLLER

However this DMAC has a subtle difference from most other DMA controllers in the way the header block is obtained. In this system the DMAC obtains the header information in the same cycle-stealing fashion that it uses to control the rest of the data stream.

The DMAC of this system has a further capability of performing several different block transfers sequentially without being interfered by the microprocessor which requested them.

Several methods of implementing state sequential machines can be used by a designer. The traditional approach has been to use SSI and MSI technology and the design stages were well defined [13]. With the advent of LSI, new approaches became possible - using microprocessors, programmable logic arrays and ROMs [34]. These methods are more flexible and changes in the state diagram made during the testing phase can be implemented by changing the content of the memory element. This is easier and much faster than designing the combinational circuits used in the traditional approach. To realize the above capabilities, the DMAC of the Graphics Controller uses the ROM Sequencer concept to implement two Mealy machines: the Header machine which reads the header block and initializes the control registers; and the transaction machine which controls the DMA transfer. They will be discussed separately in the following sub-sections.

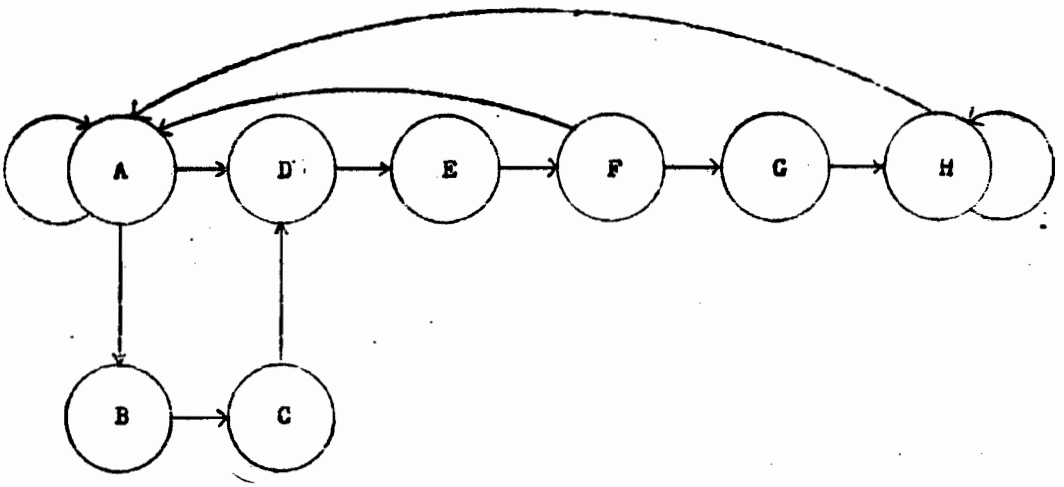
4.3.1.1 The Header Machine

The Header machine is a synchronous sequential machine which is dedicated to performing DMA transfers between the micro-bus and the Video-bus in one of the four graphics modes described previously. The sequence of events required to accomplish these four types of DMA transfers is programmed into the wiring and firmware of the header machine.

The state transition diagram of the Header machine is illustrated in Figure 4.15(a) and Figure 4.15(b) shows the table of the activity performed by the machine for each of the states.

In state A, the Header machine is in an idle state. In this state, the RAM TIMING CIRCUITRY is told not to initiate any memory reference to the microcomputers Local memory.

In state B of the Header machine, location 0000_8 of the Local memory of the microcomputer involved, is accessed. As mentioned previously in location 0000_8, the microprocessor stores a pointer to the beginning of the buffer area in which the header and data information for DMA transfer are stored. The pointer is then stored in the Micro-computer Local Memory Address Register which then addresses the buffer area by incrementing until the whole buffer has been accessed. In state C of the Header machine, the contents of the first location of the buffer, pointed to by the Microcomputer Local Memory Register is loaded into the Buffer Size Register. As has been mentioned previously, this first location con-



(a) THE STATE TRANSITION DIAGRAM FOR THE HEADER MACHINE

STATE CODE	MODE / STATE	SOLID AREA	POINT	SHADING AREA	READ BACK
0 0 0	A	IDLE (DO NOTHING)			
0 0 1	B	LOAD MICROCOMPUTER LOCAL MEMORY ADDRESS REGISTER			
0 1 0	C	LOAD BUFFER SIZE REGISTER			
0 1 1	D	LOAD COMMAND & STATUS REGISTER			
1 0 0	E	LOAD VIDEO ADDRESS REGISTER	LOAD COLOR REGISTER	LOAD VIDEO ADDRESS REGISTER	
1 0 1	F	LOAD COLOR REGISTER	LOAD NUMBER REGISTER START TRANSACTION MACHINE		
1 1 0	G	LOAD NUMBER REGISTER	(DON'T CARE)		
1 1 1	H	LOAD START WORD OR END WORD REGISTER START TRANS-ACTION MACHINE	(DON'T CARE)		

(b) THE TABLE OF ACTIVITY PERFORMED BY THE HEADER MACHINE FOR EACH OF THE STATES

FIGURE 4.15

THE HEADER MACHINE STATES

tains the 2's complement of the buffer size. In state D, the header status word, which specifies the mode of operation of the Header machine and the T.V. Sequencer, is loaded into the Command and Status register.

When the Controller first grants a microcomputer's request for a DMA activity, the Header machine goes through states B and C to determine the position and size of the buffer before processing the first header. Nevertheless, for subsequent headers within the same buffer, the Header machine goes from the idle state directly to state D. To realize this a definite format must be used for the buffer area. Two constraints are hence placed on the format of the buffer:

(i) The data area for every header must immediately follow the header;

(ii) Subsequent header and data sets must be contiguous with the preceding header and data sets.

In state E, for the Solid Area mode, Shading Area mode and Read Back mode, the Header machine loads the Video Address Register with the initial video address required for the transaction. In the Point Mode, the address of each pixel on the line(s) is a datum and so the transaction machine is responsible for loading the Video Address Register at this stage. For the Point Mode, the Header machine loads the Color word for each pixel into the Color Register in state E.

In state F, the Header machine loads the color word into the color register for the Solid Area mode. For the Point, Shading Area, and Read-Back modes, in state F, the two's complement of the number of pixels involved in the third and final word in the header block. This is loaded into the Number Register which counts the number of memory references made to the Video Memory for DMA purposes only. For the Solid Area mode, the one's complement of the number of interior words for each horizontal line section is the fourth word in the header block, and the Header machine obtains this value in state G.

In state H, the Header machine loads the start word register with the Starting word of the horizontal line section in the Solid Area mode. At the same time, the Header machine does not change state and in the following cycle the End Word is loaded into the End Word Register.

When sufficient header information has been obtained the Header machine asserts a signal Start Transaction H (STRNSH) to start the Transaction machine. On the following cycle the Transaction machine may start. As is seen in Figure 4.15(b), the STRNSH signal is asserted during the same state of the header machine in which the final header word is being fetched for Line, Shading and Read-Back modes. Since the data being fetched is clocked into the appropriate register by the same clocking pulse which transfers the Transaction machine from its idle state to its first operative state the above implementation is justified since all header information will have been obtained before the transaction machine tries to use it.

For the Solid Area mode, the signal STRNSH is asserted while the START WORD is being fetched. Hence the Transaction machine writes the Start word into the Video memory while the END WORD is being fetched. Since the RAM TIMING CIRCUITRY synchronizes the references to the two memories, it can be guaranteed that the END WORD will be clocked into the End Word Register before it is needed by the Transaction machine even if no interior words are written.

4.3.1.1.1 The Header Machine Realization

The state counter for the header machine is realized by the use of a four bit synchronous counter with synchronous load and asynchronous clear (74LS161 chip). The state transition diagram is realized by manipulating the clear, enable, load, and the input pins of the counter by using a dual two wide, two input AND-OR-INVERT package, a four input NAND gate and a three input NOR gate. Figure 4.16 illustrates this realization.

As shown in Figure 4.15(b), the states A through H are encoded with the binary numbers for 0 through 7 respectively. The NOR gate is so wired that in state A, the counter is always set up for a parallel load. If the INTERLOCK latch is set, the inputs to the counter are zero, and hence the counter remains in state A. If the latch is reset, the inputs to the counter are the binary number 3 and hence the state of the header machine

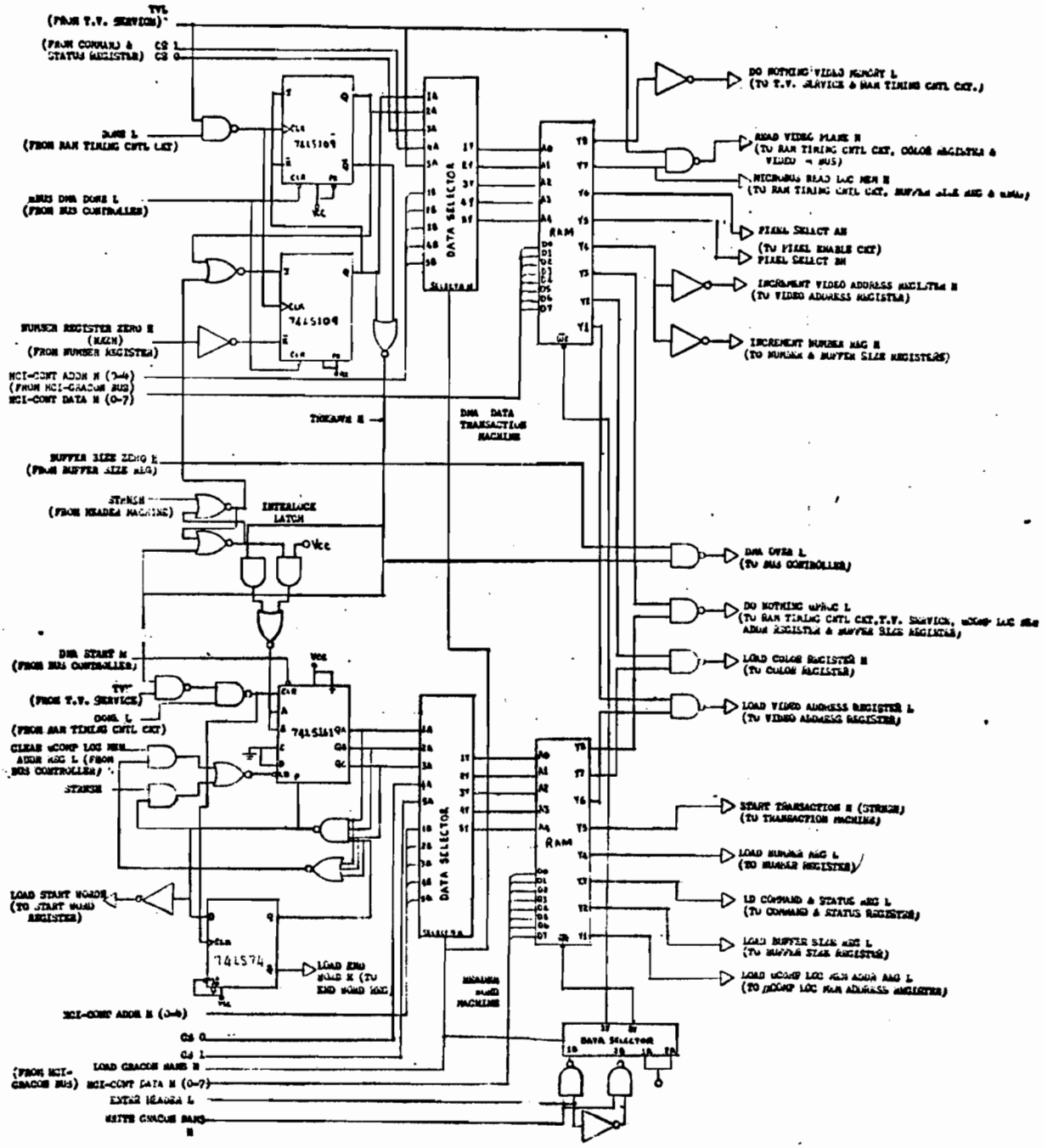


FIGURE 4.16

THE HEADER AND TRANSACTION MACHINES OF THE DMA CONTROLLER

advances to state D. However if the signal BUFFER SIZE ZERO H is asserted, the inputs to the counter remain at ZERO, keeping the header machine in the A state which is the idle state.

When the header machine first commences processing a buffer, the CLEAR MICRO-COMP LOCAL MEMORY ADDRESS REGISTER L signal from the Bus Controller disables the load input so that the header machine advances to state B when the counter increments. When the header machine asserts STRNSH, the load input of the counter is enabled. Since the INTERLOCK latch is set by the same signal, the counter loads a zero, returning the Header state to A. However if LOAD START WORD H is also asserted, the counter is disabled from loading or counting until LOAD START WORD H is cleared, which always occurs after the Start Word is loaded. The four input NAND gate is used to assert LOAD START WORD H by detecting the simultaneous condition that the header machine is in state H and the start word-end word latch is set.

The TVH signal and the TRNSOVR signal are nanded and the result is further nanded with the DONE pulse which clocks the state counter. This is done to prevent the Header machine from starting to process a new header whenever the transaction machine is interrupted while in the FINAL state.

The output signals from the states of the machine are decoded by the firmware of the ROM Sequencer RAMs (32 x 8 RAM). In order to load

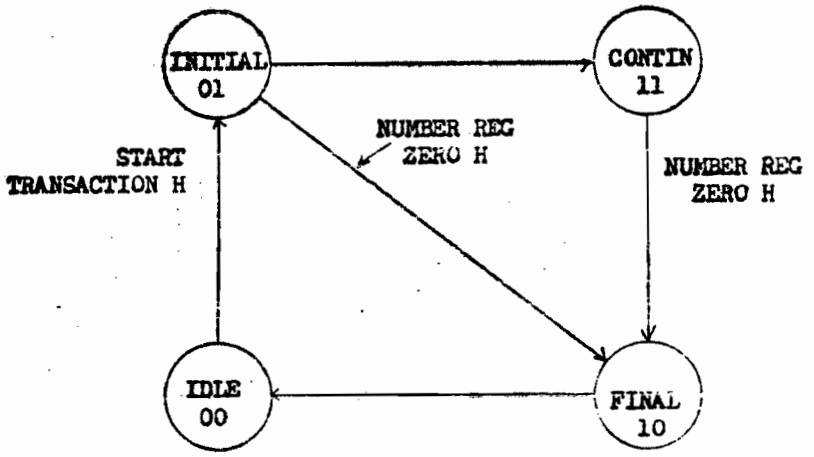
the RAMS with the sequence control data so that they can decode the states, an arrangement that uses three 2-to-1 data selectors (LS157s) is used. These LS157s are controlled by signals, generated by the Host Computer Interface in such a way that in one mode the micro-code data is fed to the RAMS and in the other mode the RAMS decode the states of the Header machine. Appendix I gives information on the micro-code used for the sequence control.

4.3.1.2 The Transaction Machine

The Transaction machine is also a synchronous sequential machine clocked by the DONE pulse. Figure 4.17(a) shows the state transition diagram and Figure 4.17(b) is a table of the activity of the Controller for each state of the Transaction machine depending on the mode being implemented.

The transaction required for every mode is characterized by an initial activity (INITIAL) a continuing activity (CONTIN) and a final activity (FINAL) aside from the Idle state (IDLE).

As has been mentioned previously, the Transaction machine is driven into the INITIAL state by the STRNSH signal from the header machine. From there it normally goes to the CONTIN state, where it stays until the Number register develops the signal Number Register Zero (NRZH), which tells the machine that the correct number of video memory references have been made. From the Continuing state, NRZH drives the Transaction machine into



(a)

THE STATE TRANSACTION DIAGRAM

STATE CODE	MODE / STATE	SOLID AREA	LINE	SHADING AREA	READ BACK
00	IDLE	WRITE START WORD INTO VIDEO MEMORY	READ FIRST PIXEL ADDRESS FROM µCOMP LOCAL MEMORY	READ FIRST PIXEL COLOR FROM uCOMP LOCAL MEMORY	READ FIRST PIXEL COLOR FROM VIDEO MEMORY
01	INITIAL	GENERATE AND WRITE INTERIOR WORD INTO VIDEO MEMORY	WRITE PIXEL INTO VIDEO MEMORY READ NEXT ADDRESS	WRITE PIXEL COLOR INTO VIDEO MEMORY READ NEXT COLOR	WRITE COLOR INTO uCOMP LOCAL MEMORY READ NEXT COLOR FROM VIDEO MEM.
11	CONTIN	WRITE END WORD INTO VIDEO MEMORY	WRITE LAST PIXEL INTO VIDEO MEMORY	WRITE LAST PIXEL COLOR INTO VIDEO MEMORY	WRITE LAST COLOR INTO uCOMP LOCAL MEMORY
10	FINAL	D O N O T H I N G			

(b)

THE TABLE OF ACTIVITY PERFORMED FOR EACH OF THE STATES

F I G U R E 4.17

T H E T R A N S A C T I O N M A C H I N E S T A T E S

the FINAL state. From the FINAL state the Transaction machine returns to the IDLE state. If the Transaction requires only one video memory reference, or, in the case of Solid Area mode no interior words are required, the NUMBER register develops the signal NRZH immediately, and the Transaction machine goes from the INITIAL state directly to the FINAL state.

4.3.1.2.1 The Transaction Machine Realization

Figure 4.16 also shows how the hardware realization of the Transaction Machine was realized.

A J-K edge-triggered flip-flop package is used to count the states of the Transaction machine. By encoding the states as shown in Figure 4.17(a), the state counter is realized with only one external NOR gate. The DONE pulse which clocks the machine is NANDED with the TVL signal, so that when TVL is asserted the state of the Transaction machine does not advance. When the T.V. read is completed, the Transaction machine continues from the same point at which it was interrupted. Also TVL forces READ VIDEO PLANE H signal to go high without causing READ LOC MEM H to go low. Thus, if the Header machine is doing a read while a T.V. Sequencer read is being performed, it is not disturbed.

When the Transaction machine is in the FINAL state a NOR gate generates the Transaction Over (TRNSOVRH) signal which resets the INTER-

LOCK latch. Also, if the BUFFER SIZE register has asserted the BUFFER SIZE ZERO H signal, then assertion of TRNSOVRH causes the signal DMA OVER L to be generated, which causes the Graphics Controller to look for a new buffer to process.

Once the INTERLOCK latch is set, the Header machine can change state if it is not in the idle state. When the latch is reset by TRNSOVRH the Header machine is permitted to advance to state D to commence processing a new header. However, if BFZH is asserted, the header machine is held in state A.

Appendix II gives the micro-coding used for the state sequencer RAMs of the Transaction machine.

4.3.2 The RAM Timing Control Circuit

The RAM Timing Control circuit is responsible for initiating all memory references to both the Microcomputer's Local Memory and the Video frame memory planes. It also controls the timing and synchronization of these memory references, and generate the DONE pulse. The DONE pulse serves as a completion signal, a clock, and a timing pulse.

Figure 4.18 illustrates the RAM Timing Control circuit realization.

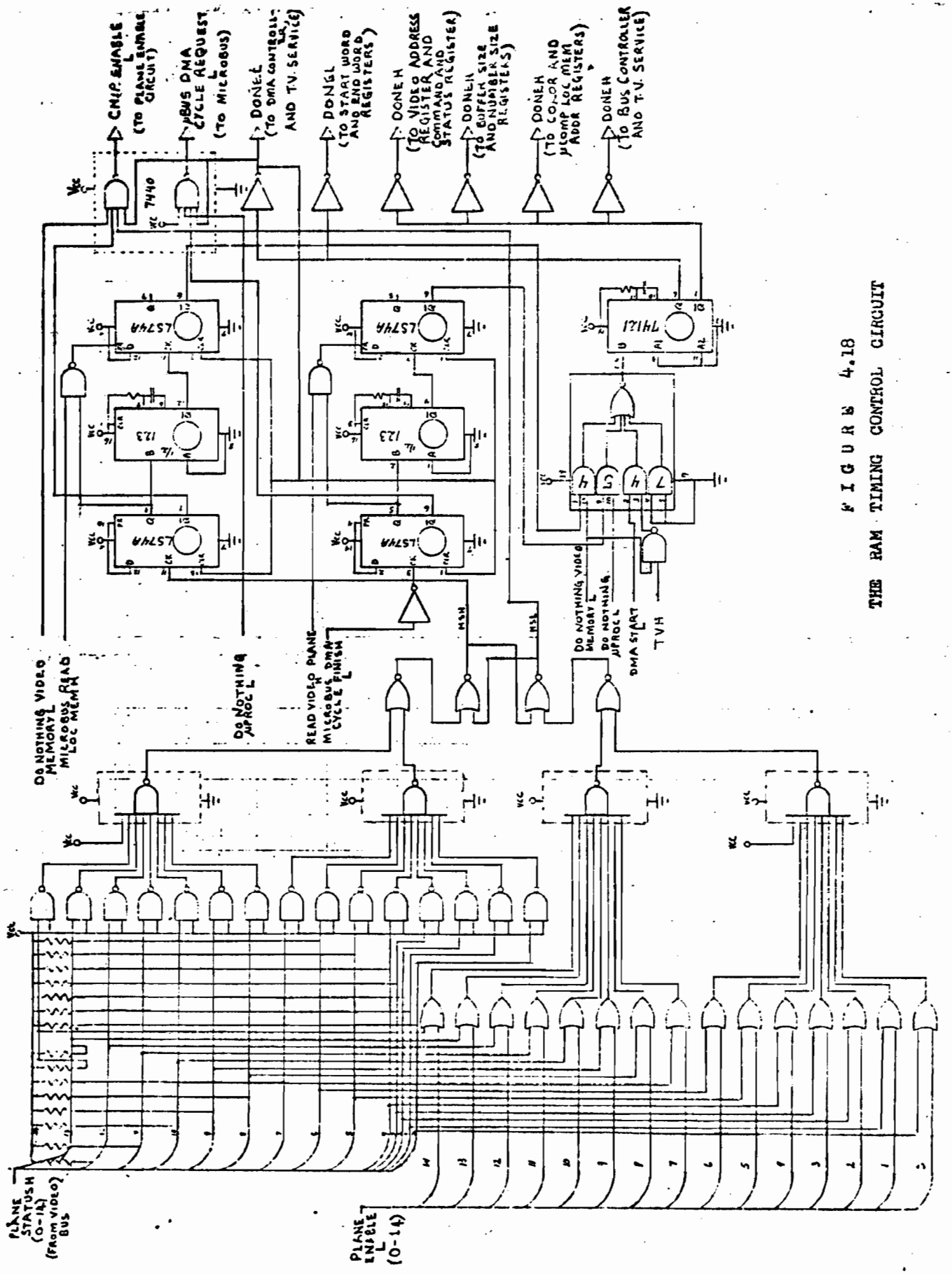


FIGURE 4.18

THE RAM TIMING CONTROL CIRCUIT

The MEMORY STATUS signal is asserted only when all Video Memory planes have completed their memory cycles, and the signal is removed only when all the Video Memory planes are ready for the next cycle. The Video Memory planes indicate cycle completion by asserting PLANE STATUS and readiness by removing PLANE STATUS H. If a plane is not enabled, its PLANE STATUS H signal will never be asserted. Hence, the other input to each OR gate is tied to PLANE ENABLE L so that planes not asserted are considered to have already asserted PLANE STATUS H. Also if a plane does not physically exist, its PLANE STATUS H will never be negatively asserted Hence the NAND gate associated with a non-existent plane has its second input tied to ground (instead to V_{CC} as shown in the figure) thus indicating that the PLANE STATUS H signal is already negatively asserted.

The removal of the MEMORY STATUS H signal is controlled by the Video Memory planes as specified above. Also the removal of DMA CYCLE FINISH is performed by the microcomputer as soon as DMA cycle REQUEST L is removed. Hence, there is no guarantee that either of these two signals will persist to inhibit CHIP ENABLE and DMA CYCLE REQUEST L until the DONE pulse has been generated. Therefore a D-type flip-flop is used to latch onto the leading edge of each of these completion signals, and the outputs of these latches are used to inhibit CHIP ENABLE L and DMA CYCLE REQUEST L until the leading edge of the DONE pulse. The DONE pulse itself then inhibits both these signals until its trailing edge indicates that the de-skew delay is over.

The DONE pulse is responsible for clocking the DMA Control Circuitry and all the registers of the Graphics Controller at the end of the memory cycles requested by the DMA Controller. If only one memory cycle is requested, then the DONE pulse is generated at the end of the memory cycle of the memory being accessed. However if both, the microcomputer's Local memory and the Video memory planes are being accessed at the same time, the DONE pulse is generated when the slowest memory has completed its cycle.

When reading from a memory, it is necessary to delay the DONE pulse to allow the data from the memory to de-skew on the bus. This is accomplished for each of the two memories by a small delay circuit.

For the Video memory planes, when the MEMORY STATUS H is asserted it is immediately latched and it inhibits CHIP ENABLE L. The output of the D flip-flop which latches MEMORY STATUS H is also used to fire a monostable. The monostable produces a negative pulse whose trailing edge will clock a second D flip-flop. Connected to the preset of the second flip-flop is a NAND gate which NANDs the latched MEMORY STATUS H and READ LOC MEM H. If READ LOC MEM H is asserted, then a write operation is indicated for the Video Memory planes. For this condition the preset of the flip-flop is asserted and the flip-flop sets immediately without a delay. However if READ LOC MEM H is not asserted, a read operation is indicated for the Video Memory Plane.

The flip-flop is not preset, and hence does not set until the trailing edge of the monostable pulse. The pulse-width of the monostable determines the length of the de-skew delay. The delay circuit for the microcomputer bulk memory operates in precisely the same way.

A third monostable determines the pulse-width of the DONE pulse also. This is used to overcome skew and propagation delay from the Graphics Controller to the two memories.

Also notice that the DMA START L (DMASTL) signal from the Bus Controller also generates a DONE pulse to start the DMA Controller by incrementing the state counter of the header machine.

4.3.3 The Bus Controller

This circuitry is responsible to arbitrate the selection of a microcomputer ready for DMA activity.

Figure 4.19 illustrates the realization of this circuitry.

While the DMA Controller is operating, the three flip-flops are set, thus asserting DMA START H. When the final memory cycle of the last transaction is done, the DONE pulse resets the first flip-flop, which in turn clears the two other flip-flops. This action asserts Micro-Bus

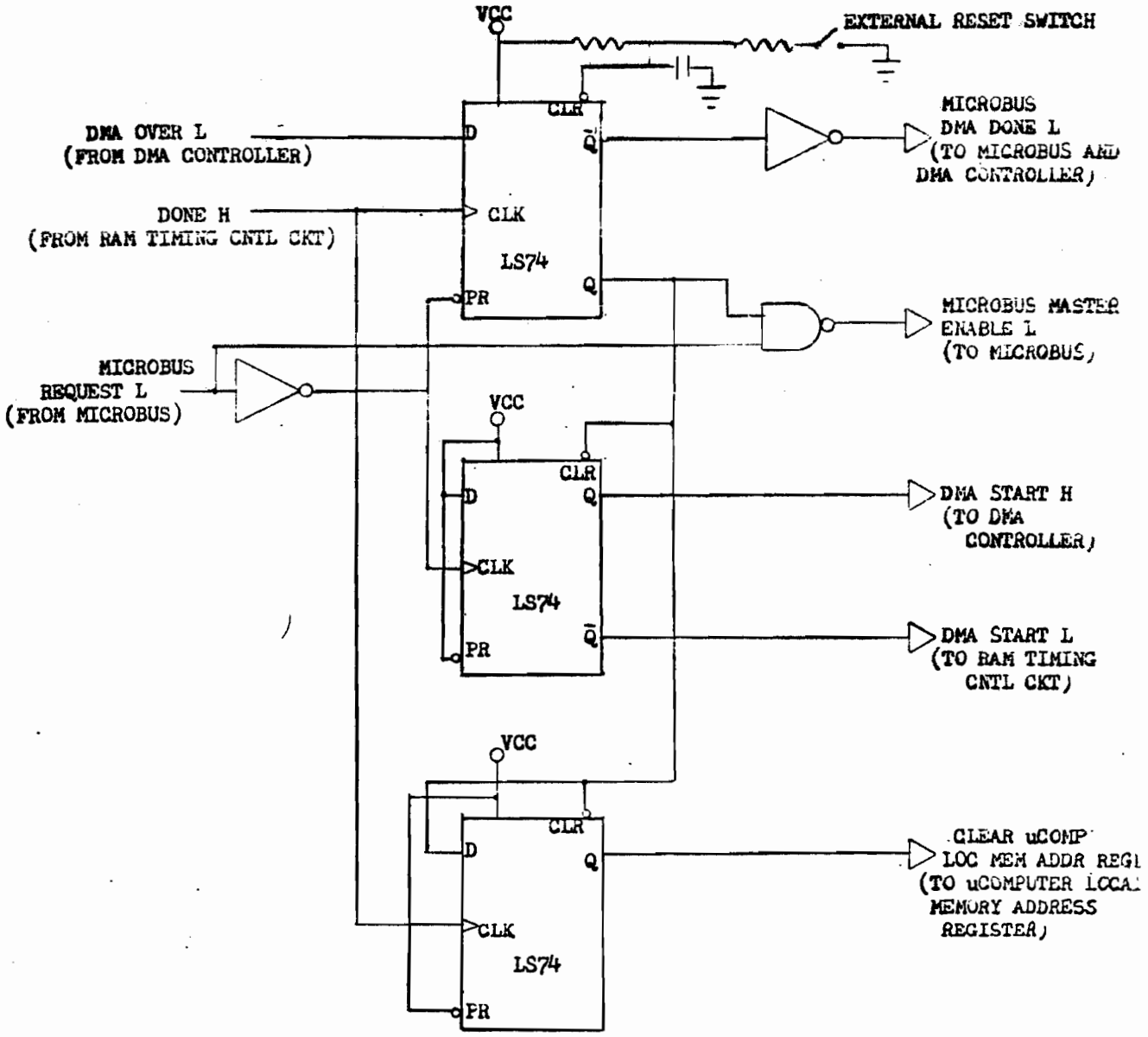


FIGURE 4.19

THE BUS CONTROLLER

DMA DONE which clears the DMA Grant flip-flop on the microcomputer involved. Also CLEAR MICROCOMP ADD REG L is asserted clearing the Microcomputer Local Memory Address register to location 0000_8. Moreover, the negative assertion of DMA START H and the assertion of DMA DONE L keep the Header and the Transaction machines in their idle states.

When the DMA Grant flip-flop of the microcomputer has been successfully cleared, MICROBUS REQUEST L is no longer asserted. Hence MICROBUS DMA DONE L is removed and MICROBUS MASTER ENABLE L is asserted.

This state persists until one of the microcomputers sets its DMA Grant flip-flop, thus asserting MICROBUS REQUEST L. This removes MICROBUS MASTER ENABLE L making sure no other microcomputer will be latched while one of them is busy.

Notice that there is a circuit on the Bus Controller for initializing the Graphics Controller when power is first turned on. The RC circuit on the clear input of the first flip-flop ensures that this flip-flop is initially reset. With this condition the Bus Controller can initialize all the circuits of the Graphics Controller which require initialization.

4.3.4 The T.V. Service

The T.V. Sequencer transfers data to the T.V. monitor by shifting it serially out from shift registers on the Video memory planes. The shift registers must be periodically reloaded from a buffer register. Once the contents of the buffer register have been transformed to the shift register on each plane, new information must be loaded into the buffer register by doing a video memory read. As soon as this is done, the T.V. Sequencer asserts the signal BUFFER EMPTY H .

Figure 4.20 illustrates how the T.V. Service circuitry deals with the BUFFER EMPTY H signal. The signals TVL and TVH are responsible for accomplishing a number of tasks. TVL interrupts the Transaction machine. It is also used as an address bit for the RAMs of the Transaction machine. By this method, the outputs of the Transaction machine are changed to those required for a T.V. read. In addition TVL is responsible for selecting the addresses provided by either the Video Address Register or the T.V. Sequencer.

Since the Transaction machine will negatively assert DO NOTHING VIDEO MEMORY L causing a video memory cycle to be initiated, a DONE pulse will be generated at the end of the cycle. This pulse will toggle the J-K-flip-flop thus removing TVH and TVL. The pulse also clocks TVH into a second J-K-flip-flop (before the signal is removed) thus asserting TV DONE H. This signal is sent to the Video memory planes to clock the data accessed by the read into the buffer registers and to the T.V. Sequencer to increment its Video Memory Address Register.

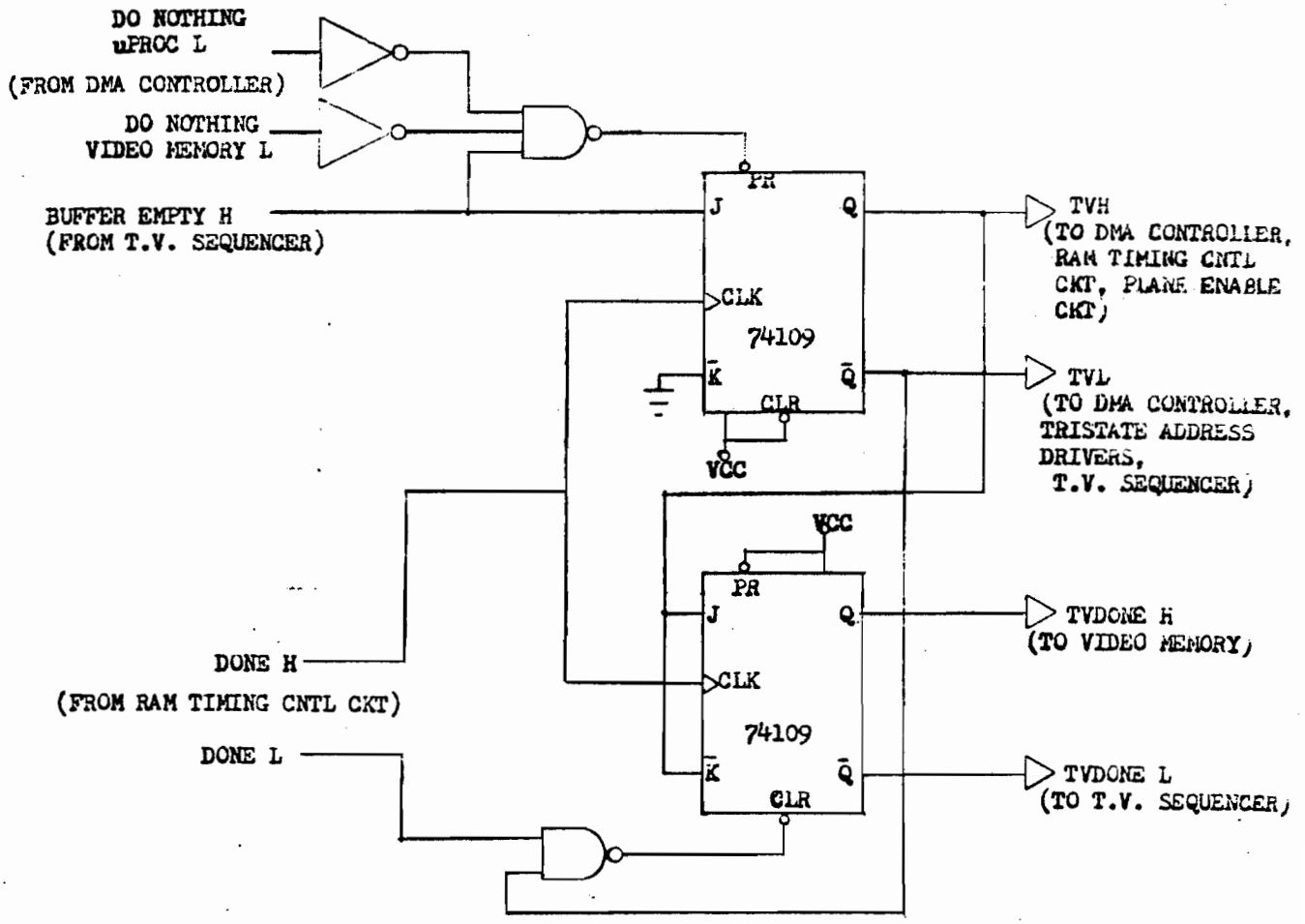


FIGURE 4.20
THE T.V. SERVICE

The J-K flip-flop is cleared by the trailing edge of DONE so that the pulse-width of TV DONE H is approximately the same as that of DONE pulse.

4.4 The T.V. Sequencer Design

The T.V. Sequencer can be divided into six sub-systems. The interconnections of these sub-systems are shown in Figure 4.21. These sub-systems will be briefly discussed in this section. Also on the T.V. Sequencer a hardware test facility has been installed to test it independently [65].

4.4.1 The System Clock

The system clock generates all the clocking signals required by the other Sequencer sub-systems. The master clock for the Sequencer is switch selectable as 12.6 MHz or 20 MHz. By using a proper dividing circuitry the system clock generates two output signals that drive the entire Sequencer. The 'SYNCLK' which is switch selectable as 1.26 MHz or 2.0 MHz is used to clock the Synchronous generator sub-system. The 'SHIFTCLK' controls the shifting of the individual pixels onto the T.V. monitor. For the high resolution mode, the 'SHIFTCLK' is either 12.6 MHz (with the 12.6 MHz clock) or 10 MHz (with the 20 MHz clock). For the low resolution

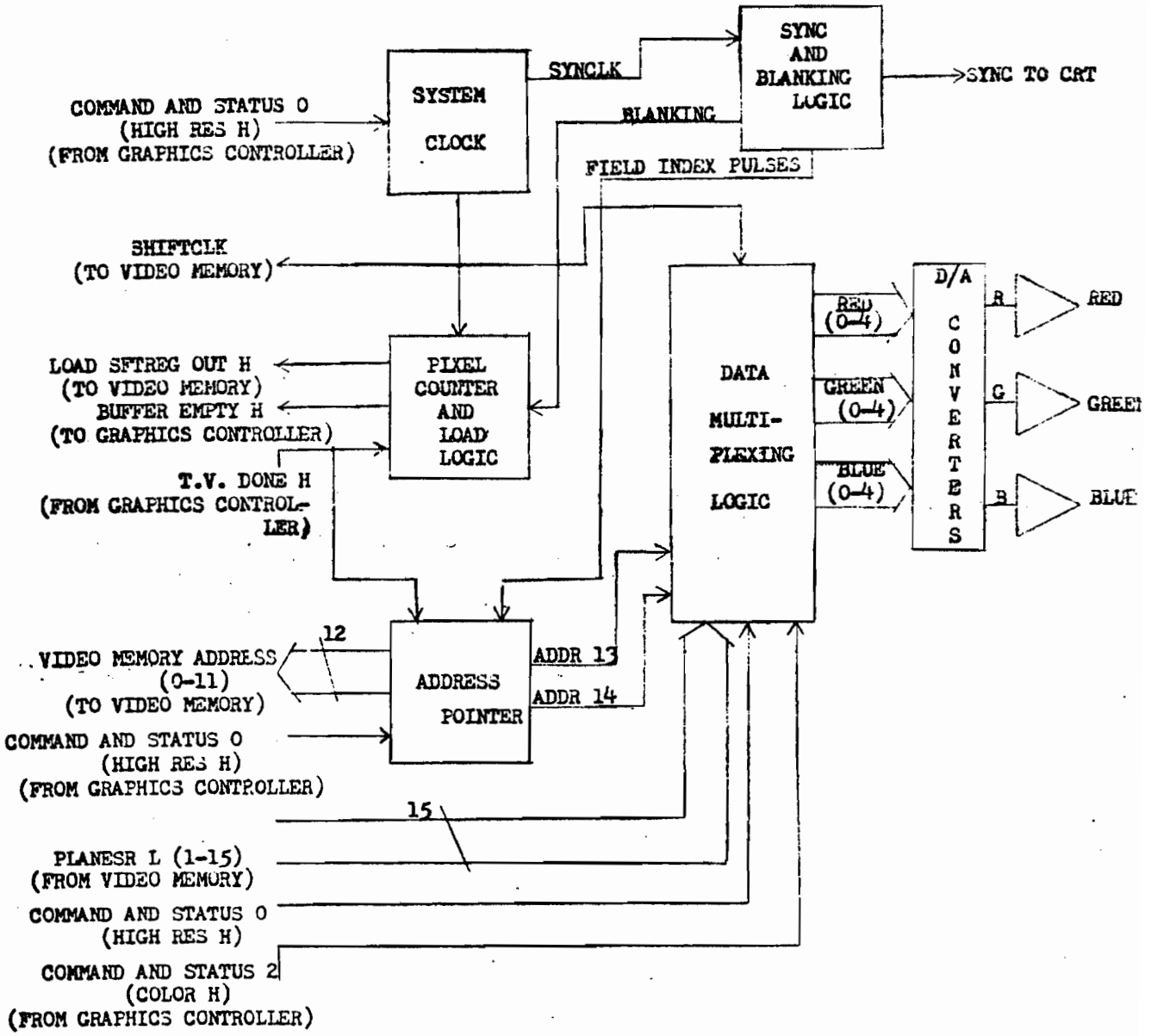


FIGURE 4.21

THE T.V. SEQUENCER SUB-SYSTEMS

mode, it is divided by two to give either 6.3 MHz or 5 MHz.

4.4.2 The Synchronous Generator

The Synchronous generator is built around the MM5320 MOS-LSI [43] Synchronous generator chip. This chip requires a clock (SYNCLK) with a 50% duty cycle, of either 1.26 MHz or 2.0 MHz selectable by switch. From this input clock, the chip generates four signals required to run the rest of the Synchronous generator circuitry.

The Composite Synchronous signal is a standard signal that is used to synchronize the T.V. monitor with the data being supplied by the T.V. Sequencer via the RED GREEN and BLUE signals.

The Blanking signal is a standard signal which is used to blank the T.V. monitor screen during retrace.

The Field Index pulses are used to differentiate between the even and the odd fields of the frame.

4.4.3 The Pixel Counter and Load Logic

The Pixel Counter keeps track of the number of pixels shifted out of the memory planes. When enough pixels have been shifted to empty

the shift registers on the Video Memory plane, the pixel counter issues commands to load the shift registers with new data words. The Blanking signal from the Synchronous generator is used to inhibit the fetching of new data from memory during horizontal or vertical retrace, when the screen is forced to go black. This feature saves memory, as data need not be supplied to the Sequencer when signal conventions require the screen to go black during retrace.

4.4.4 The Address Pointer

The Address Pointer supplies the memory address for the read requests, to the Video Bus under the supervision of the Graphics Controller.

This pointer is incremented after each memory request, causing data to be fetched from sequential memory locations. Only 12 address lines are supplied to the Video-bus as there are only 4096 addressable locations on the memory planes. When the Sequencer is in the high resolution mode, these 12 bits are inadequate, as the high resolution mode requires four times the amount of data that the low resolution mode requires. Therefore two bits, more significant than the most significant bit sent to the Video-bus, called ADDR 13 and ADDR 14 are sent to the Data Multiplexing Logic so that it may select data from the appropriate memory planes when in the high resolution mode. The Field Index information from the Synchronous generator, is used to reset the Address Pointer at the beginning of each frame, to

force the data to be fetched from the beginning of the picture. In the low resolution mode, the same data is provided for both the fields. This is accomplished by resetting the Address Pointer before each field whenever the low resolution mode is selected.

4.4.5 The Data Multiplexing Logic

This sub-system accepts data from the Video Memory plane serial output lines (PLANESRL) and routes it to the RED, GREEN and BLUE digital output lines according to the mode selected by the HRESH and COLOR H control lines. The Data Multiplexing Logic also contains a set of D-type flip-flops at the output lines, clocked by SHIFTCLK. These flip-flops are used to de-skew the data arriving at the output of the multiplexing logic as some data paths have more gate delays than others.

4.4.6 The Digital to Analog Converters

The digital outputs of the multiplexing logic drive three 5 bit D/A converters, one per primary color, to obtain the analog signals necessary to drive the T.V. monitor. The DAC 90 BG [6] chip is used for this purpose. The three analog signals are buffered by current drivers [42] (the LH0002) before being sent to the RED, GREEN and BLUE inputs of the T.V. monitor.

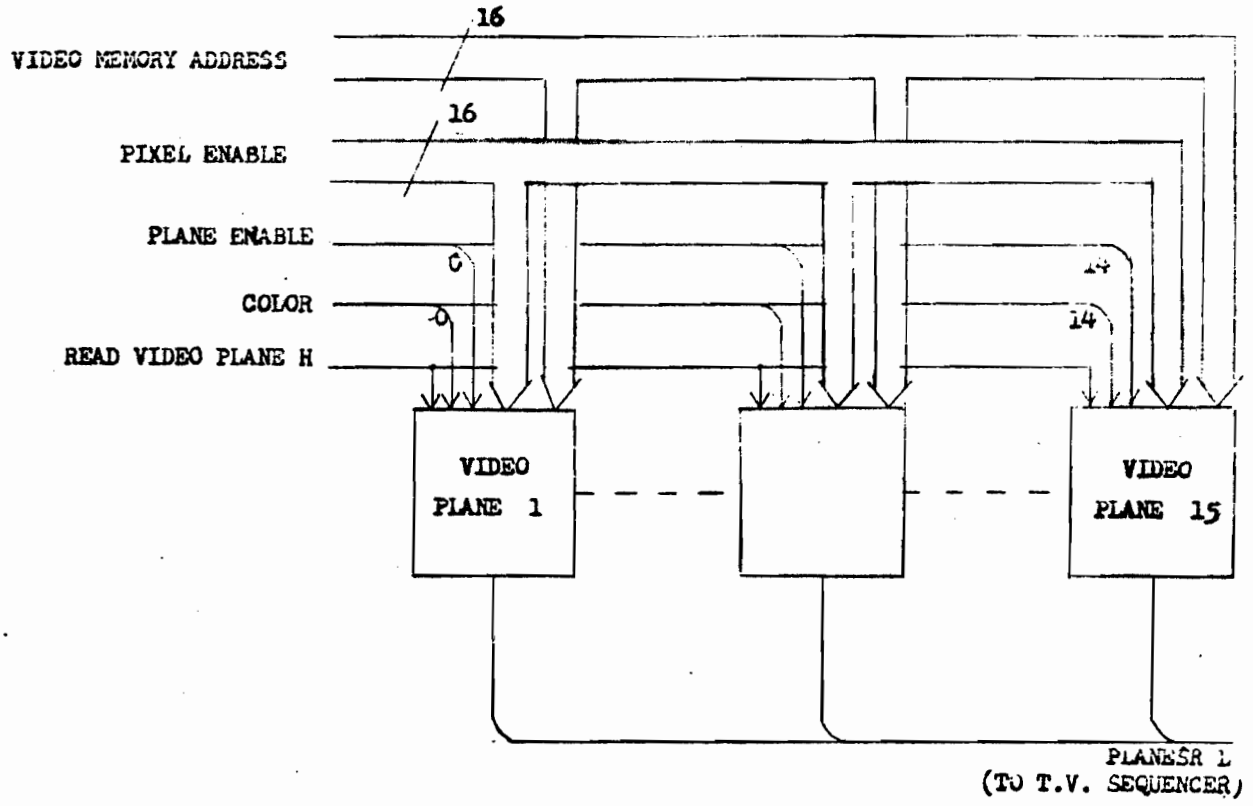
4.5 The Video Memory Planes Design Consideration

As has been mentioned previously, the system's Video memory has a modular construction. The basic module, the Video Memory plane, contains 64K bits of static RAM arranged in 4K words by 16 bits [5],[39]. This packaging accommodates the (256 x 256) resolution display. Hence one such plane would store one bit of intensity information per pixel for the complete display.

In the high resolution mode of (512 x 512) pixels, four Video memory planes are necessary to provide the one intensity bit per pixel storage for the display. In such a case, the T.V. display is divided into four horizontal strips and each of the four Video memory planes contains information for one strip.

The layout of the Video Memory system is shown in Figure 4.22. One Plane Enable and one color line go to each plane while the Video Memory Address lines, the Pixel Enable lines and the Read Video Mem line are common to all the planes. The Video Memory Address specify the location of the word on the memory while the Pixel Enables locate the bits in the word by enabling the required chips in a plane. Each plane contributes a single line to the T.V. Sequencer giving only one bit of information for each displayed pixel.

The implementation of one Video memory plane is shown in Figure 4.23. The configuration uses four (1K x 1) chips tied together to give the 4K x 1 block in a column. 16 such blocks give 4K words each 16



F I G U R E 4.22

THE LAYOUT OF THE VIDEO MEMORY SYSTEM

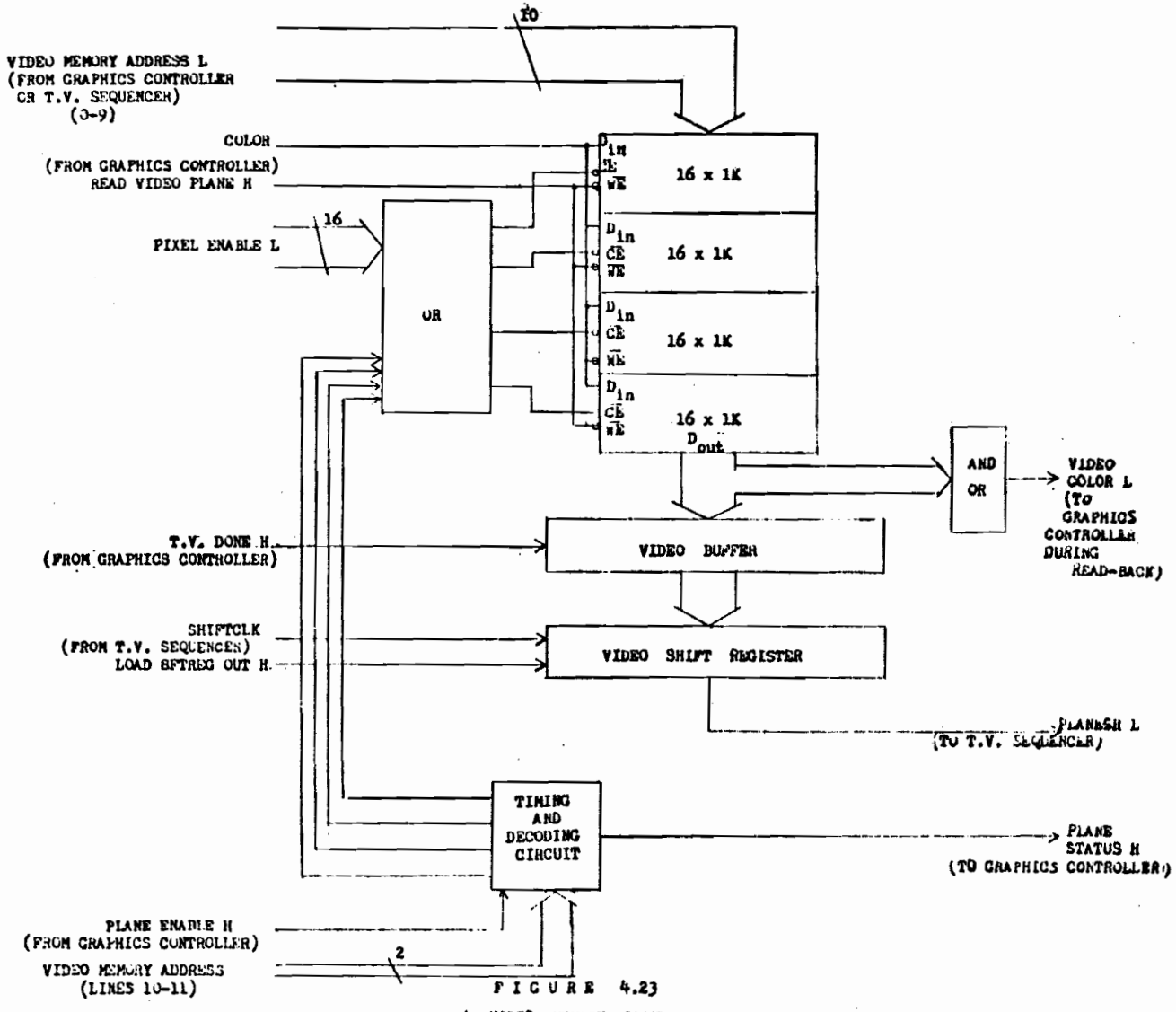


FIGURE 4.23

A VIDEO MEMORY PLANE

bits wide. The Color data, the Video Memory address and the Read Video Mem H (the write enable signal) are common to all memory chips in the plane. One Pixel Enable line is sent to each chip in a column to enable them together. Color information which is placed on the data line would be stored in the memory location pointed to by the Pixel Enable and the Video Address coordinates. The Plane Enable line, after passing through a timing circuit, is used to strobe a decoder generating a"one hot"code derived from the two most significant bits of the address lines. The 4 lines from the decoder are ORed with each Pixel Enable line to generate the Chip Enables for each of the 4 blocks of the memory.

The Plane Enables from the Graphics Controller are responsible to implement opaque or transparent displays. In the opaque mode, the color information overwrites the previous image. The Graphics Controller does this by enabling all the planes in the low resolution mode. In the transparent mode, the color information mixes with the previous image color to give a new color. In this case, the Graphics Controller enables only those planes that give the new color information while the previous color information remains the same from the planes not enabled.

After the required access delay time, the Plane Status signal is asserted. This in turn by a hand-shaking operation disables Plane Enable H. This clears the flip-flop that generates Plane Status and hence disables it on receiving the disabled Plane Enable.

For the Read-Back mode all the outputs from the tied memory chips are ORed together to give information for only one bit of the word (i.e., one column of the 16).

For servicing the T.V. Sequencer requests, a double buffer arrangement is used. This allows the Graphics Controller to service a T.V. Sequencer request any time before the next word arrives.

The data to the T.V. Sequencer is shifted out serially, one bit at a time from the Video Shift Register into the plane's serial output line. Each memory plane supplies a serial output line to the T.V. Sequencer providing a maximum of 15 bits in parallel when the system is fully configured for the low resolution mode (15 memory planes). Upon receipt of a shift command (SHIFTCLK) from the T.V. Sequencer all the memory planes simultaneously place the next shift register bit onto the serial output lines. The T.V. Sequencer counts the bits shifted out of the Video Shift register and when the shift register has been emptied, it issues a parallel load command (LOADSFTREGOUT) to all the memory planes, forcing the next data word to be loaded into the shift registers from the Video Buffer. Simultaneously, the T.V. Sequencer issues a command to the Graphics Controller (BUFFER EMPTY H) to read a new data word into the Video Buffer from the memory. The Graphics Controller uses the T.V. DONE H signal to load the word into the Video Buffer. This memory read operation need not complete until the shift registers are once again empty. This is an elegant solution to the problem of the high refresh data rates involved.

CHAPTER V

THE PERFORMANCE EVALUATION OF G. R. A. D. S. AND ITS GRAPHIC MODES

5.1 The Purpose of Evaluation

The purpose of evaluation of G.R.A.D.S., which is still under design and building stages, is performance projection. In all systems that are under design, it is necessary to estimate the feasibility and performance of the design prior to the actual development.

The approach that will be used in the evaluation of this system is an analytic one. In an analytic approach the performance of the system is evaluated with respect to various system parameters and system work load. An analytic model has always been useful as an additional point of reference in hardware analysis for all of the major evaluation purposes [35]. Usually when a system has been established for a while, the most accurate way to obtain a performance model is by measuring the system under its real workload. However at this stage of the development of the system, it is sufficient to develop its functional model. The values of the performance measures are then obtained by analytic means. Measurement and modelling are however complementary processes [58] since:

(a) A model provides a frame work for measurement;

(b) Measurement provides data for validating the model;

(c) The model aids in testing hypothesis and finding solutions to performance problems.

(d) Correctness of model predictions is finally verified by measurement.

5.2 The Performance of the Display System Using the Graphic Modes

The rate at which polygons and vectors can be animated or changed and the number of such polygons and vectors involved are inter-related parameters. Both of these parameters depend on the picture resolution and the number of microcomputer modules used. The user can trade off one for another so that in any given system implementation more polygons can be animated if resolution and/or animation rates are reduced [48].

As has been previously mentioned during normal operation, the host computer via the interface DMA machine, provides the start and end coordinates and color information by loading the appropriate Macro-instruction/data words in each of the microcomputer's Local memory. The microprocessor then reads this data from the Local memory and proceeds with the "Bit expansion" operation whereby the X, Y, coordinates and the color for each of the pixels that would approximate the picture is produced. This pixel information is then stored in the Local memory. The time spent by the Host Computer Interface to write the Macro-instruction data into the Local memory and the time spent by the microprocessor to read the Macro-instruction/data from the Local memory is considered as overhead, because

it is added to the time spent by the microprocessor to compute the actual pixel data. It is referred to as the Macro-overhead. The time spent in producing the header is also considered as an overhead. It is referred to as the Header-overhead.

The Host Computer Interface has been designed to operate at 2 MHz to transfer the 20 bit words to the Local memory. The Local memory itself has a cycle time of about 500 nsec., i.e., it operates at about 2 MHz also. Hence the time spent by the Interface DMA machine to write and the microprocessor to read each Macro-instruction/data word totals to 1 μsec.

Now the actual limiting factors in the data throughput rate involve the microprocessors, the Local Memory, the Video Memory and the Graphics Controller systems. As will be derived in Section 5.5, the throughput rate of the current design based on the 500 nsec. cycle times for the Local and Video memories, is analyzed to be 807 nsec. write transaction with the Video memory. The microprocessor used in this system takes about 200 nsec. to execute one micro-instruction word. Now let us assume, that for vector drawing routines written in assembly language for the VAX approximately n micro-instructions should be executed, on the average, to compute the coordinates or color data for each pixel of an arbitrary vector. In the analysis of the four Graphic modes, it is assumed that the relevant microcode routines are similar in complexity and involve the same number of steps n. Hence the time taken by the microprocessor

to produce the necessary information for one pixel is (200 x n) nsec. = 0.2 n μsec.

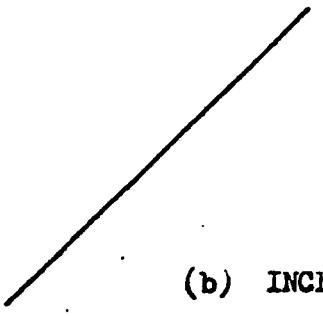
These timing characteristics are used for comparing the various Graphic modes performance. The workload used by the Graphic modes for the analysis are standard test patterns [49] as illustrated in Figure 5.1. The relative analysis applies over a reasonable range of values for n. A typical value of 10 appears reasonable and will be used in the analysis.

5.2.1 The Performance with Solid Area Mode

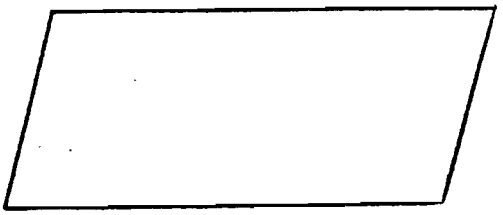
The Solid Area mode is used by the system to implement horizontal line segments in a uniformly colored area. In this section, the capability of this mode to implement the horizontal line segments of uniform color will be demonstrated by illustrating a few derived performance figures. The analysis is based on the microprocessor taking (0.2 x 10) 2 μs to process one word of the header and taking 1 μs to move and read a macro-instruction word as well as a Video memory write transaction time of 807 nsec for one word as explained on page 103.

The workload to be used to show the capability of the Solid Area mode is a horizontal uniformly colored line about 15 cm long. This means that in the low resolution mode of 256 x 256 pixels and with a

(a) HORIZONTAL LINE



(b) INCLINED LINE



(c) PARALLELOGRAM

FIGURE 5.1

STANDARD TEST PATTERNS

48.3 cm (19") T.V. tube, with an aspect ratio of 4/3, the line has about 100 pixels. In the high resolution mode of 512 x 512 pixels, each such line has twice as many pixels (i.e., about 200).

To establish the best possible performance it will be assumed that a buffer of image information is always available for the Graphics Controller to access.

Now to compute the six Header words of the Solid Area mode, the microprocessor uses (2 x 6 μsec. (at 2 μsec/word) = 12 μsec. A macro overhead of 4 words (at 1 μs/word) needs a further 4 μsec. for this mode. Also it takes the Graphics Controller (100 pixels ÷ 16 pixels per word) x 807 nsec. write cycle time) 5.04 μsec. to store the 16 pixel words into the Video memory. Hence a total of (12 + 4 + 5.04) = 21.04 μsec. are needed to implement the given line in this mode. Therefore the number of such lines that can be drawn in $\frac{1}{30}$ sec. (the T.V. frame refresh rate) is $\frac{1}{30} \times \frac{1}{21.04 \times 10^{-6}} \simeq 1584$. Similar figures can be derived using the Point, Shading Area and the Readback modes, taking into consideration that single pixel write operations are involved. Table 5.1 shows the performance of all the modes to implement a 15 cm, uniformly colored horizontal line in both the low and the high resolutions. It is clear that the Solid Area mode is the most efficient for displays which have large uniformly colored images that can be broken down into such horizontal line segments.

GRAPHIC MODE	NO. OF 15cm HORIZONTAL UNIFORMLY COLOURED LINES DRAWN IN 1/30 SEC IN LOW RES.	NO. OF 15cm HORIZONTAL UNIFORMLY COLOURED LINES DRAWN IN 1/30 SEC IN HIGH RES.
SOLID AREA	1584	1277
POINT	114	58
SHADING AREA	114	58
READ-BACK	114	58

TABLE 5.1

THE CAPABILITY OF SOLID AREA MODE FOR THE PRESENT DISPLAY SYSTEM

GRAPHIC MODE	NO. OF 15cm HORIZONTAL LINES WITH VARYING SHADING DRAWN IN 1/30 SEC IN LOW RES.	NO. OF 15cm HORIZONTAL LINES WITH VARYING SHADING DRAWN IN 1/30 SEC IN HIGH RES.
SOLID AREA	22	11
POINT	114	58
SHADING AREA	33	17
READ-BACK	33	17

TABLE 5.2

THE CAPABILITY OF POINT MODE FOR THE PRESENT DISPLAY SYSTEM

5.2.2 The Performance with the Point Mode

A 15cm line having about 100 pixels of the same color and inclined at about 37° (to the horizontal) is considered to show the capability of the Point mode. After analyzing in a similar way as described in the previous sub-section (and considering that one word carries information for one pixel) Table 5.2 was generated. It should be noted that for the Solid Area mode, the Shading Area mode and Read Back mode, a header is generated for horizontal line segments only and hence for every pixel on the inclined line a header is generated for these modes. It is clear from Table 5.2 that the Point mode is the best mode to use when uniformly colored lines with arbitrary orientation are to be animated and displayed. This mode can also very effectively implement uniformly colored randomly generated points.

5.2.3 The Performance with the Shading Area Mode

A 15 cm horizontal line consisting of about 100 pixels all with varying shades of color will be considered as the workload. A similar analysis as described previously was followed in the generation of Table 5.3. It should be noticed that for the Solid Area and Point modes, a header will have to be generated for every pixel on the horizontal line since it is of a different color. It is clear that the Shading-Area Mode is the best mode that the system uses to display shaded images.

GRAPHIC MODE	NO. OF 15cm HORIZONTAL LINES WITH VARYING SHADING DRAWN IN 1/30 SEC IN LOW RES.	NO. OF 15cm HORIZONTAL LINES WITH VARYING SHADING DRAWN IN 1/30 SEC IN HIGH RES
SOLID AREA	22	11
POINT	30	15
SHADING AREA	114	58
READ-BACK	114	58

TABLE 5.3

SHADED LINE CAPABILITY OF PRESENT DISPLAY SYSTEM

LENGTH OF EACH VECTOR	THE MICROPROCESSOR COMPUTED NO. OF UNIFORMLY COLORED VECTORS THAT CAN BE ANIMATED AT A 30Hz RATE
30 cm	82
15 cm	164
7.5cm	325

TABLE 5.4

RELATIONSHIP BETWEEN VECTOR SIZES AND ANIMATION RATES AS COMPUTED BY A MICROPROCESSOR

5.3 The Relationship between Vector Sizes and Animation Rates as Computed by a Microprocessor

With the use of Point mode, it will be illustrated in this section that when the size of vectors stored in a fixed buffer is varied, the animation rates also vary. It is assumed that the microprocessor generates N uniformly colored vectors each with an average length of about L cm in low resolution. The buffer size used in the Local memory is assumed to accommodate information for 1000 pixels. For a given buffer size, the number of vectors stored is inversely proportional to the vector length. When L = 30 cm, 200 pixels constitute one line and the buffer would be able to accommodate 5 such lines (i.e., N = 5). The total time needed to compute this buffer = Macro-overhead + Header Overhead + data time. This is found to be 12 + (2 x 3) + (2 x 1000) = 2018 μs. This means that $\frac{10^6}{2018 \times 30} = 16.52$ buffers or 82, 30 cm lines can be computed in 1/30 sec. Similar calculations were made with two other sizes. Table 5.4 illustrates the result. It is seen that as the length of the lines decrease, the number of vectors successfully animated increases. However the macro-overhead for the same buffer size increases as the length is reduced.

Tables 5.2 and 5.4 show that the Graphic Controller throughput rate of 114, 15 cm lines is less than the microprocessors computing capability of 164, 15 cm lines. Therefore a dual (parallel) processor operation where one buffer is being emptied while the second one is being calculated seems desirable for the Point mode of operation.

5.4 Implementation of Colored Polygons by the Microprocessors

In this section, analysis will first be made on the microprocessor computation to generate single-colored polygons followed by a similar analysis on shaded polygons. It is assumed that on the average each polygon (with 4 sides) has 10 horizontal lines each line consisting of 100 pixels.

5.4.1 Analysis of Microprocessor Computation for Uniformly Colored Polygons

When using the Solid Area mode, the microprocessor reads the 6 word macro-instruction with its parameters to implement the single colored four sided polygon in 6 μs. Six header words are computed for each line requiring 6 words x 2 μs/word = 12 μsec. Hence for the ten lines of the Polygon, (12 x 10) = 120 μsec. are required. Hence 264 such polygons can be computed by the microprocessor using the Solid Area Mode in 33 milliseconds. This corresponds to 2640 scan lines. Comparing this with the 1584 horizontal line capability of the graphics system given in Table 5.1, it seems that again two microcomputers are adequate for displaying about 264 average sized polygons in low resolution. Similarly for the high resolution mode the number of polygons (of same dimensions as in low resolution) computed by the microprocessor in 33 milliseconds is calculated to be 135. This corresponds to (135 x 20) = 2700 scan lines. Again comparing

this with the 1277 horizontal line capability of the graphic system, given in Table 5.1, it seems that again two microcomputers are sufficient for displaying the polygons in high resolution.

Table 5.5 illustrates the superiority of the Solid Area mode over the other modes in implementing animated single-colored typical polygons each containing 10 horizontal lines with 100 pixels per line. Hence this mode can very efficiently be used to generate uniform colors like those of the sky or the sea or for blanking the entire screen with a solid color. In about 1 msec., the entire screen can be painted in one color using the Solid Area mode.

5.4.2 Analysis of Microprocessor Computation for Shaded Polygons

To enhance the realism of computer generated displays, shading is often desirable. In general this requires the display of different colors at each of the picture elements. The Shading Area mode was designed to efficiently implement such shaded displays.

The hardware design favours the direction of shading to be along the raster scan line. For the typical polygon with 10 horizontal line segments, each with 100 pixels of different color shades, in the Shading Area mode, the microprocessor needs a macro-overhead of ((1+(4x10))words x 1 μs/word) 41 μs, a header overhead of (30 words x 2 μs/word) = 60 μs, and a data computation time

GRAPHIC MODE	NO. OF SINGLE COLORED (1000 PIXEL) POLYGONS THAT CAN BE COMPUTED IN 1/30 SEC. IN LOW RESOLUTION	NO. OF SINGLE COLORED (4000 PIXEL) POLYGONS THAT CAN BE COMPUTED IN 1/30 SEC. IN HIGH RESOLUTION
SOLID AREA	264	135
POINT	16	4
SHADING AREA	15	4
READ-BACK	15	4

TABLE 5.5

TABLE ILLUSTRATING THE SUPERIORITY OF SOLID AREA MODE TO IMPLEMENT UNIFORMLY COLORED POLYGONS

GRAPHIC MODE	NO. OF SHADED (1000 PIXEL) POLYGONS THAT CAN BE COMPUTED IN 1/30 SEC. IN LOW RESOLUTION	NO. OF SHADED (4000 PIXEL) POLYGONS THAT CAN BE COMPUTED IN 1/30 SEC. IN HIGH RESOLUTION
SOLID AREA	2	0
POINT	8	2
SHADING AREA	15	4
READ-BACK	15	4

TABLE 5.6

SHADING AREA MODE CAPABILITY TO IMPLEMENT SHADED POLYGONS

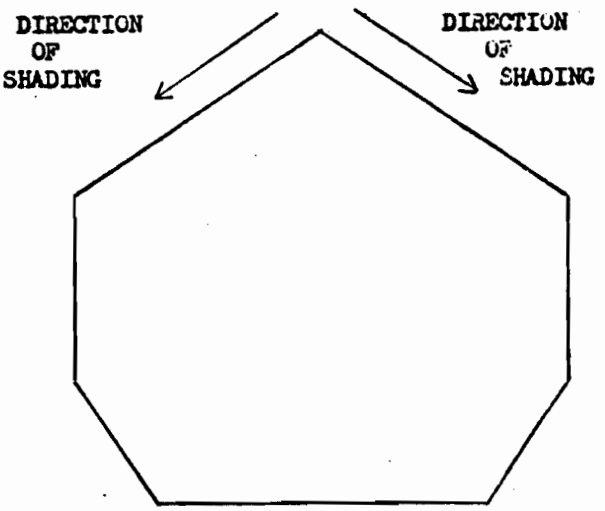
of (2 μs/pixel 10 lines of 100 pixels each) = 2000 μs all totalling to 2101 μs for computation, for the 10 lines associated with the polygons. Hence 15 such polygons can be calculated in 1/30 of a second. This corresponds to 150 scan lines (of 100 pixels each) being computed in 1/30 of a second. Comparing this against the 127 shaded lines of Table 5.3, it can be seen that two microcomputers would be suitable for this task in the low resolution mode. One microprocessor would compute information while the second is being serviced by the Graphics Controller.

Table 5.6 illustrates the superiority of Shading Area mode for computing shaded polygons. Notice that Solid Area mode is useless to implement polygons with every pixel of different color shade.

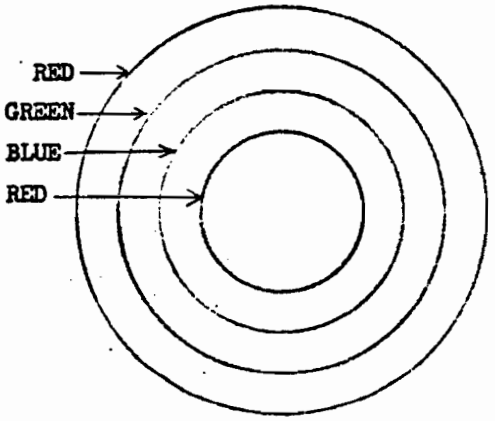
Next consider a polygon with 100 horizontal line segments each with 164 pixels of different colors. The microprocessor needs a macro-overhead of ((1 + (4 x 100)) words x 1 μs/word) 401 μs, a header overhead of (300 words x 2 μs/word) = 600 μs and a data computation time of (2 μs/pixel x 100 lines of 164 pixels each) all totalling to 33.8 ms for computations. Hence only one such polygon can be calculated in 1/30 of a second. Now notice that this fully shaded polygon occupies 25% of the screen and therefore it means that GRADS could support up to 25% shading of the complete raster screen in low resolution. Because of this relatively limited capability for shaded graphics, it is recommended to use this mode for about 10% of the display composition. Such displays do give the impression of being "shaded" and hence look pleasant or "realistic".

Similar analysis for the high resolution Shading Area Mode implementing shaded polygons indicates that only about 6.25% of the screen could be shaded. Also by comparing the value of (4 x 20) = 80 horizontal lines computed in $\frac{1}{30}$ sec. (in Table 5.6) to the value of 58 such lines computed in the same time (in Table 5.3), it is again suggested that the use of two microprocessor units is sufficient. Table 5.6 illustrates the superiority Shading Area mode relative to the Solid Area and Point modes when shaded polygons have to be computed. Notice again that the Solid Area mode is useless for shading polygons, with every pixel of a different shade in the high resolution.

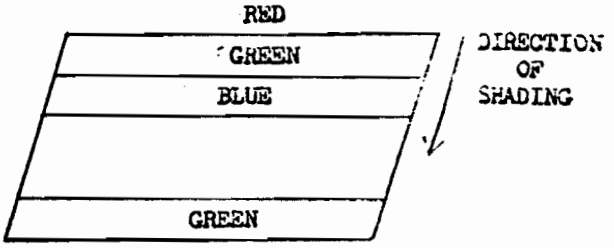
In general the Shading Area mode is the best whenever every pixel in any polygon is of a different color (Figure 5.2(a)). However if the shading algorithm of the polygon can generate vectors or curves (oriented in any arbitrary direction) with points of equal color, (like Figure 5.2(b)),the Point mode would be the most efficient. Finally the Solid Area mode is the most effective mode if the shading is such that each of the horizontal line segments of the polygon is of a different solid color as shown in Figure 5.2(c).



(a) POLYGON WITH EVERY PIXEL SHADED DIFFERENTLY



(b) SHADING OF CURVES WITH EQUAL COLORS



(c) POLYGONS WITH SOLID COLORED HORIZONTAL LINE SEGMENTS

FIGURE 5.2

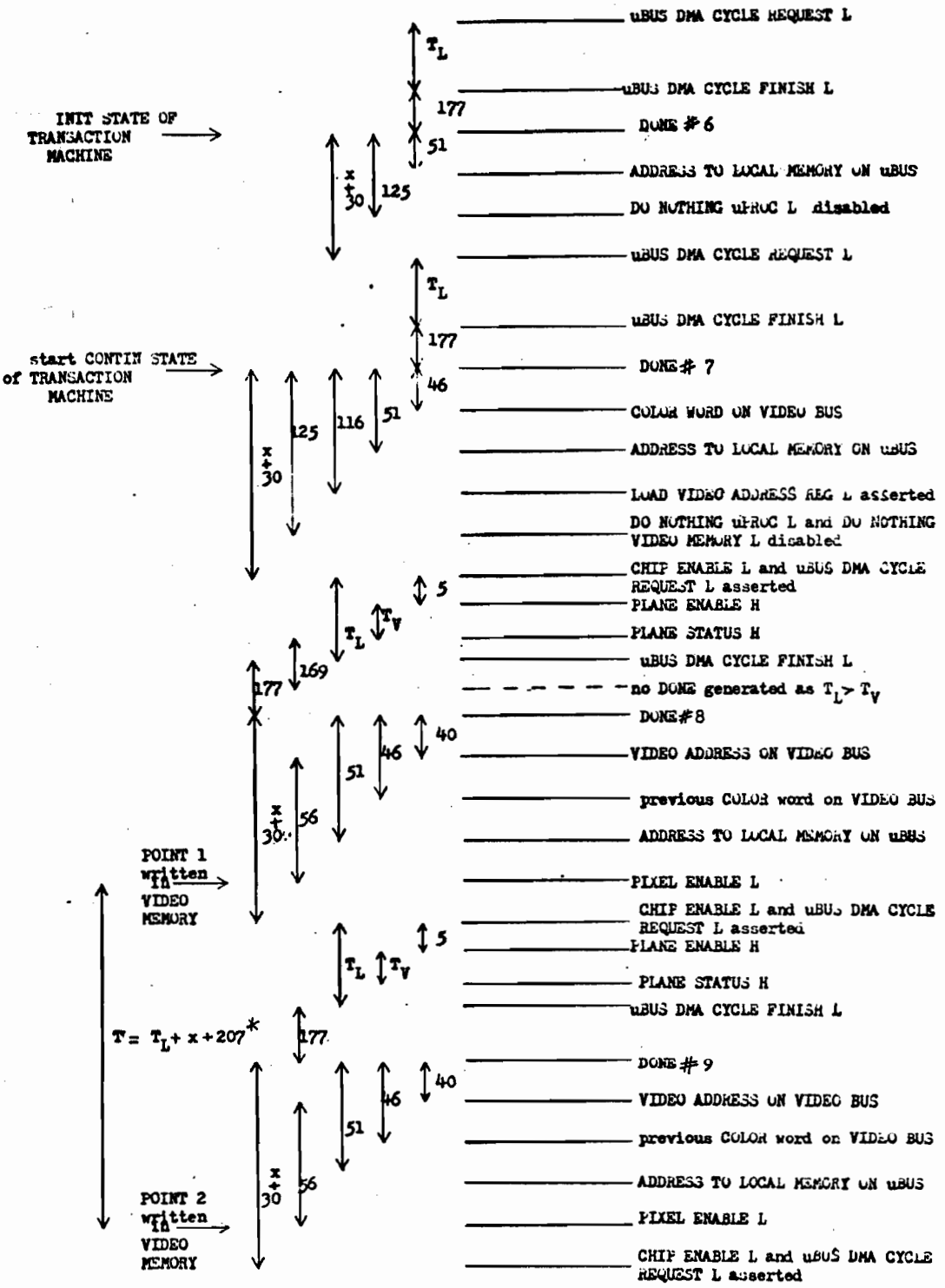
SHADED POLYGONS

5.5 The Performance Evaluation of an Enhanced GRADS System

It is obvious that the potential computing power of the system increases if more microcomputer modules are added in parallel to the first one. The Local memory and the Video memory in the system also determine the throughput of the system. If the two memories have different bandwidths, the Graphics Controller synchronizes the faster one to the slower one, the system throughput being then determined by the slower memory. At present, due to the integrated circuits available, both the Local memory and the Video memory systems have cycle times of 500 nsec.

Figure 5.3 illustrates the timing diagrams (for the worst case) and the signal sequences in the Graphics Controller used to transfer pixel information from the Micro-bus to the Video bus for the Point mode. It illustrates the sequences and timing when the Cycle time of the Local Memory (T_M) is greater than that of the Video memory (T_V). If $T_V > T_L$, then Graphics Controller waits for the Plane Status to generate the DONE pulse. Hence in Figure 5.3 for example, the generation of DONE #8 is determined by the Plane Status, and not by the shorter cycle time of the Local memory. Figure 5.3 will now be used to illustrate the effects of the Local memory and Video memory bandwidths on the system throughput and display quality.

The T.V. Sequencer needs to scan the 16 words of one complete horizontal line of the raster screen in about 50 μsec. Hence it needs one word every $\frac{50,000}{16}$ = 3125 nsec. to display an image. To fulfill this



* all time intervals shown are in nanoseconds

FIGURE 5.3

GRAPHIC CONTROLLER SEQUENCES FOR THE POINT MODE OF OPERATION WITH $T_L > T_V$

it requests for one word of pixel information every 3125 nsec. However as seen from Figure 5.3, the Graphics Controller fills the Video Memory with pixel information about every $(T_L + X + 207)$ nsec. when $T_L > T_V$ where T_L is the cycle time of the Local Memory, T_V is the cycle time of the Video Memory and X is the Video bus settling time (deskew time). At present T_L (and also T_V) is about 500 nsec. and X is adjusted to be 100 nsec. Hence the Video Memory is being accessed every 807 nsec. by the Graphics Controller. Therefore, on the average for 3.87 words of pixel information (for the write transaction) that are stored in the Video Memory one word is read by the T.V. Sequencer. By reducing T_L or T_V, similar calculations were made to derive the graph of Figure 5.4. It can be seen from this Figure 5.4 that if T_L or T_V or both are reduced by using faster memories, the number of words of pixel information written into the Video memory before being accessed by the T.V. Sequencer, is increased. This increased performance would reflect in the estimated capabilities derived in Sections 5.2 to 5.4.

It should also be noticed that the Graphics Controller performance is based on gating delays of 207 nsec. This figure is derived using worst case specifications for chip delay times. However actual figures are expected to be less since the Controller will execute at the actual chip speeds due to it asynchronous design.

Similar conclusions were drawn by analyzing the other Graphic Modes.

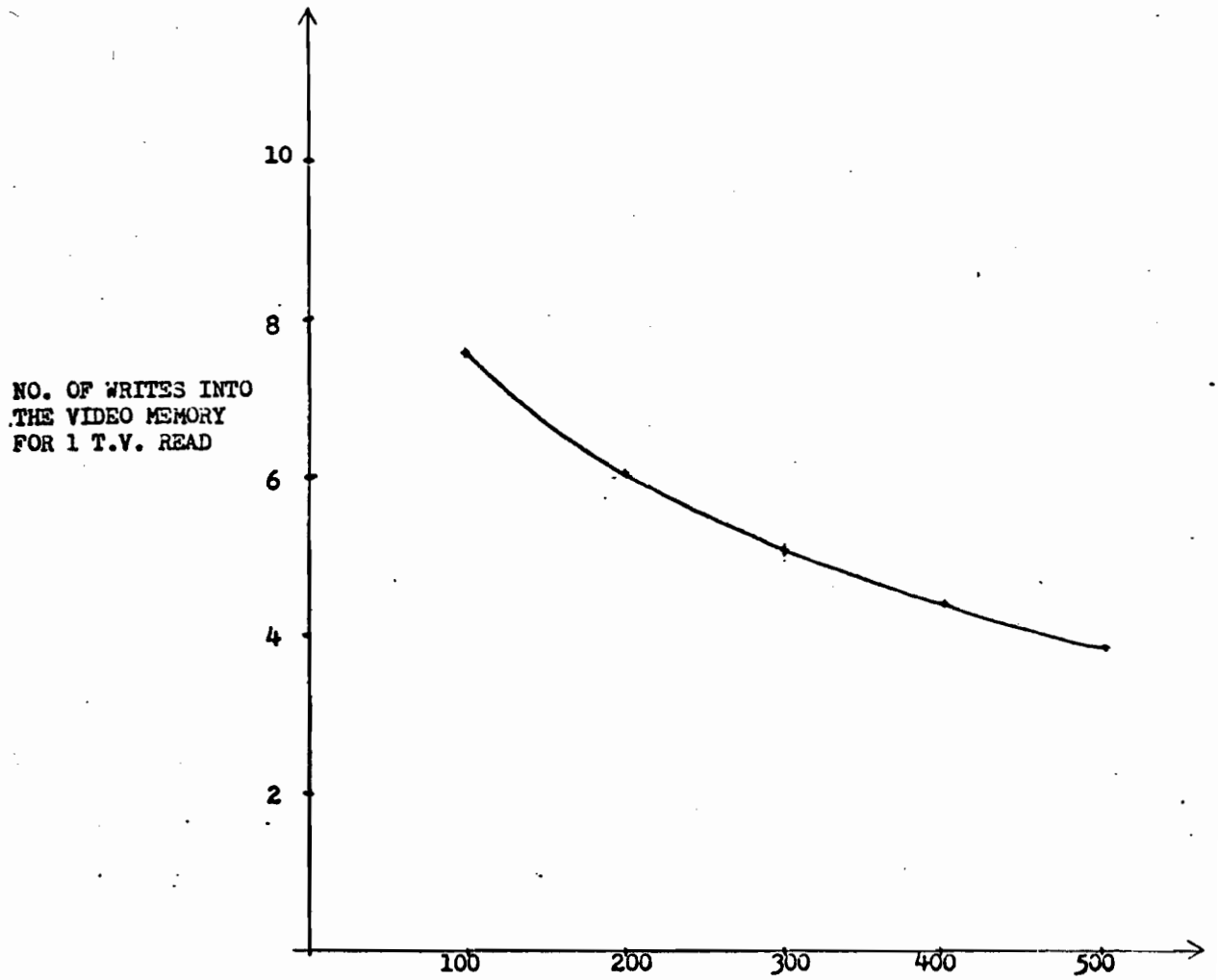


FIGURE 5.4

EFFECT OF INCREASING BANDWIDTHS OF THE LOCAL MEMORY OR THE VIDEO MEMORY

CHAPTER VI

THE SYSTEM INTEGRATION

The author was responsible for realizing the Graphics Controller and integrating it with the T.V. Sequencer and one Video memory plane.

In Computer Engineering conceiving an idea and putting it on paper is an art; but realizing the idea needs a very meticulous scientific approach. The debugging of the Graphics Controller and parts of the T.V. Sequencer and the Video memory plane was a test of patience and endurance. As for any system, Murphy's Law was unequivocally proved in the testing process - if anything can go wrong it will.

There were a number of available techniques used by the author to identify and correct the types of problems involved in trouble-shooting a system with about 375 Integrated Circuits. This chapter describes the types of trouble sources encountered and the experimental set-up used to identify and correct the problems when integrating the display system of GRADS.

6.1 The Trouble Sources

When working with digital systems, most failures encountered can be attributed to four essential factors: wiring faults, component failure, software bugs and interference of noise.

6.1.1 Wiring Faults

Wiring faults are the most common and troublesome problems. The conventional technique to detect them is by making a resistance check from point to point in the system.

The type of wire used and the way it is wrapped are two factors a designer must consider at this stage. Some types of wire which do not require stripping will scrape off very easily on the slightest mishandling of the socket boards. Such wires should not be tightly attached to the DIP socket pins around corners. One learns by experience to choose the right type of wire. The wrapping around the pin should be 3 to 5 turns deep with the bottom-most wrap being insulated to avoid a short with the ground bus.

6.1.2 Component Failure

No component is perfect in its performance or mechanical strength. Sooner or later they all experience failures. Most components exhibit a lifetime characteristics as shown in Figure 6.1 [30] . As is seen from this figure, even new components are liable to fail. Hence components that work well for a while can occasionally fail and cause frustration to the tester who assumes that they are functioned when they are not. It should also be noted that high precision components need not be highly reliable.

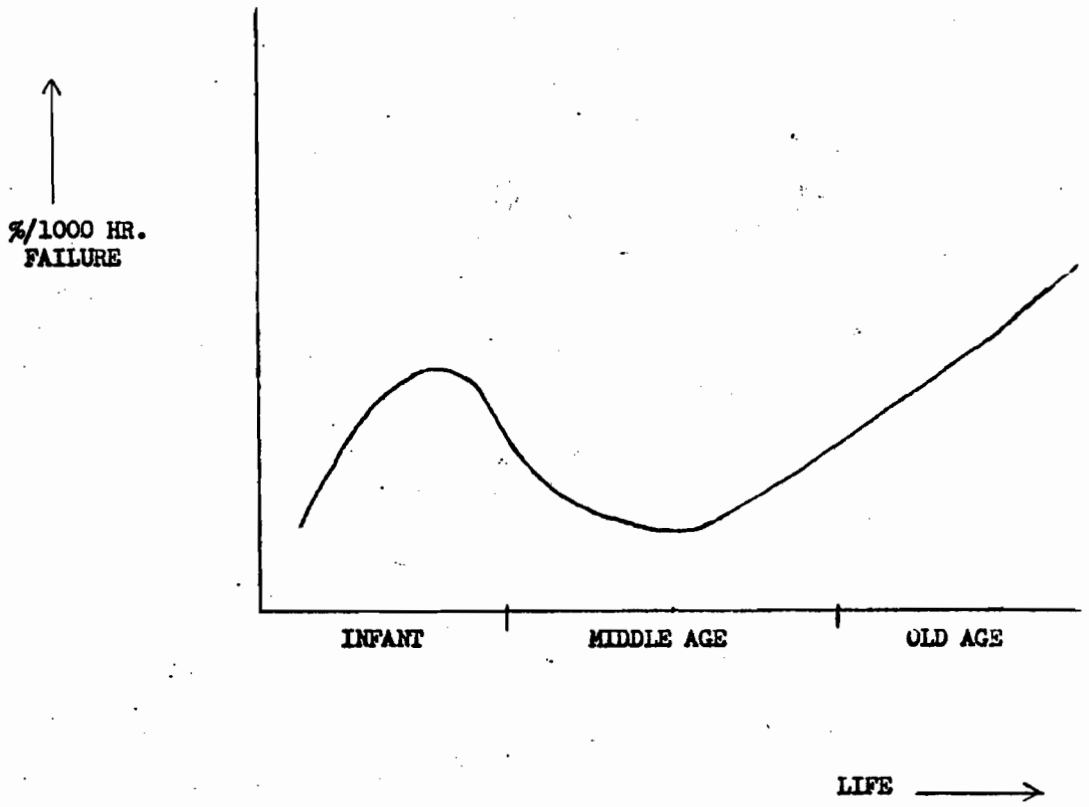


FIGURE 6.1

COMPONENT FAILURE RATE VERSUS AGE

Most components are given a figure of merit, known as its mean-time-between-failures (MTBF). This is a statistical prediction, in hours, of the time the part remains functional in a given environment. The failure-rate of a system is defined as $\frac{1}{MTBF}$. It can be found by adding the failure rates of all the components in the system. A table of percent per 1000 hrs. failure-rate is shown in Table 6.1 for application in military avionics. The failure-rates of the display system modules can be calculated by using this table.

Table 6.2 illustrates how the failure rate for the Video memory plane was calculated. Table 6.3 illustrates the failure rates and the MTBF for the three modules of the display system. These values were calculated using the same technique as illustrated in Table 6.2. As seen in Table 6.3, the failure rate of the display system is about 17% per 1000 hrs. usage. Hence the mean-time-between failures of the display system is 5900 hrs.

6.1.3 Software Problems

The third type of trouble source is the software failure. Software problems, or bugs, are often the hardest to identify since the designer often overlooks the fact that failure of his system may be due to faulty software. Since much of the Graphics Controller design is based on a ROM Sequencer concept, firmware errors are also included in this software category.

COMPONENT	%/1000 HR. FAILURE RATE
INTEGRATED CIRCUITS SSI, MSI, LSI	0.015
RESISTOR	0.002
CAPACITOR	0.02
QUARTZ CRYSTAL	0.05
SOLDERED JOINT	0.0002
WIRE-WRAPPED JOINT	0.00002
CONNECTOR CONTACT	0.005
DIODE	0.013
TRANSISTOR	0.04
VARIABLE RESISTOR	0.01
TRANSFORMER	0.5

TABLE 6.1

COMPONENT FAILURE RATE *

* Lesea A, Zaks R: Microprocessor Interfacing Techniques, Sybex, 1977, p.347

COMPONENT	QUANTITY	TOTAL %/1000 HR. FAILURE RATE	
I.C.s	134	$134 \times 0.015 = 2.01$	
RESISTORS	17	$17 \times 0.002 = 0.034$	
CAPACITORS	148	$148 \times 0.02 = 2.96$	
SOLDERED JOINTS	474	$474 \times 2 \times 10^{-4} = 0.095$	
WIRE-WRAPPED JNTS	1472	$1472 \times 2 \times 10^{-5} = 0.029$	MTBF HRS.
CONNECTOR CONTACTS	70	$70 \times 5 \times 10^{-3} = 0.35$	
TOTAL FAILURE RATE		5.478	$\frac{1000}{\frac{5.478}{100}} \approx 18300$

TABLE 6.2

CALCULATION OF THE FAILURE RATE OF THE VIDEO MEMORY PLANE

MODULE	FAILURE RATE % PER 1000 HR.	MEAN TIME BETWEEN FAILURES HRS.
GRAPHICS CONTROLLER	7.96	12,600
T.V. SEQUENCER	3.05	32,800
VIDEO MEMORY PLANE	5.48	18,300
TRANSFORMER	0.5	200,000

TABLE 6.3

APPROXIMATE SYSTEM FAILURE RATE

Generally no program is ever perfect. A program is limited in precision, speed and flexibility.

6.1.4 Noise

Noise or interference can be wherever there is an electromagnetic field. This can be within the system or from external sources. When integrated circuits switch, they generate small current changes in their power requirements because of internal circuit characteristics. If too many circuits switch at once, the power-supply voltage may change enough to affect other parts of the circuit. Usually bypass-capacitors are installed near each I.C. to prevent this type of noise. If two wires are close together, induction may create noise between them. The power supply should be well designed to avoid serious ripple on the output. Noise spikes can pass to the circuit from the switching effects of nearby peripherals like the teletype or disk drives.

6.2 The Methodology to Trouble-Shooting

In this section, the tools and methods used to find the faults will be discussed.

A systematic approach is required to test a circuit in as little time as is possible and without causing much damage. The method used to debug the display system is summarized in Figure 6.2.

The tools available for testing definitely determine the speed and efficiency of the debugging process. Table 6.4 illustrates the tools currently available for solving specific problems while trouble-shooting a digital system.

The actual set-up used for debugging the system is illustrated in Figure 6.3. For testing purposes, the Volt-Ohmeter (VOM), a buzzer, an oscilloscope and a logic analyzer (digital domain analyzer) were available. The VOM and the buzzer were mainly used for locating shorts in the circuit. For monitoring and solving timing problems, an oscilloscope with a bandwidth of 20 MHz was extensively used. The logic analyzer [9] was useful for observing the operation of the state machines of the Graphics Controller.

For implementing the necessary software a S100 computer with the Cromemco Disk Operating System [7] and the Zapple monitor [61] were widely used. Various interfaces to I/O peripherals were also available.

The software mostly emulated the Host Computer Interface bus to the Graphics Controller, and the Microbus. Programs were written to load and read-back the micro-code into and from the RAMs of the DMA Controller machine of the Graphics Controller. The reading back of the micro-

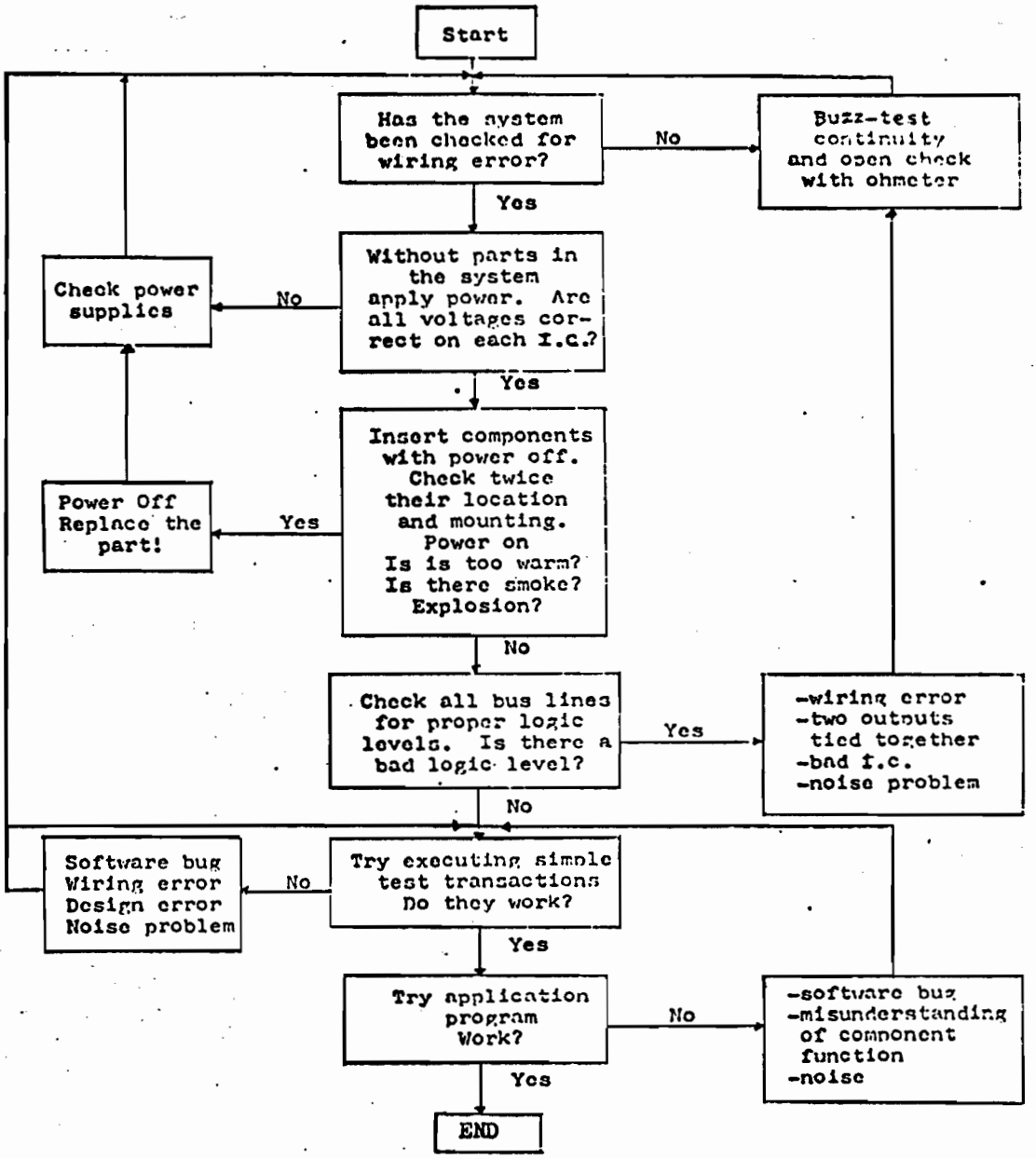


FIGURE 6.1

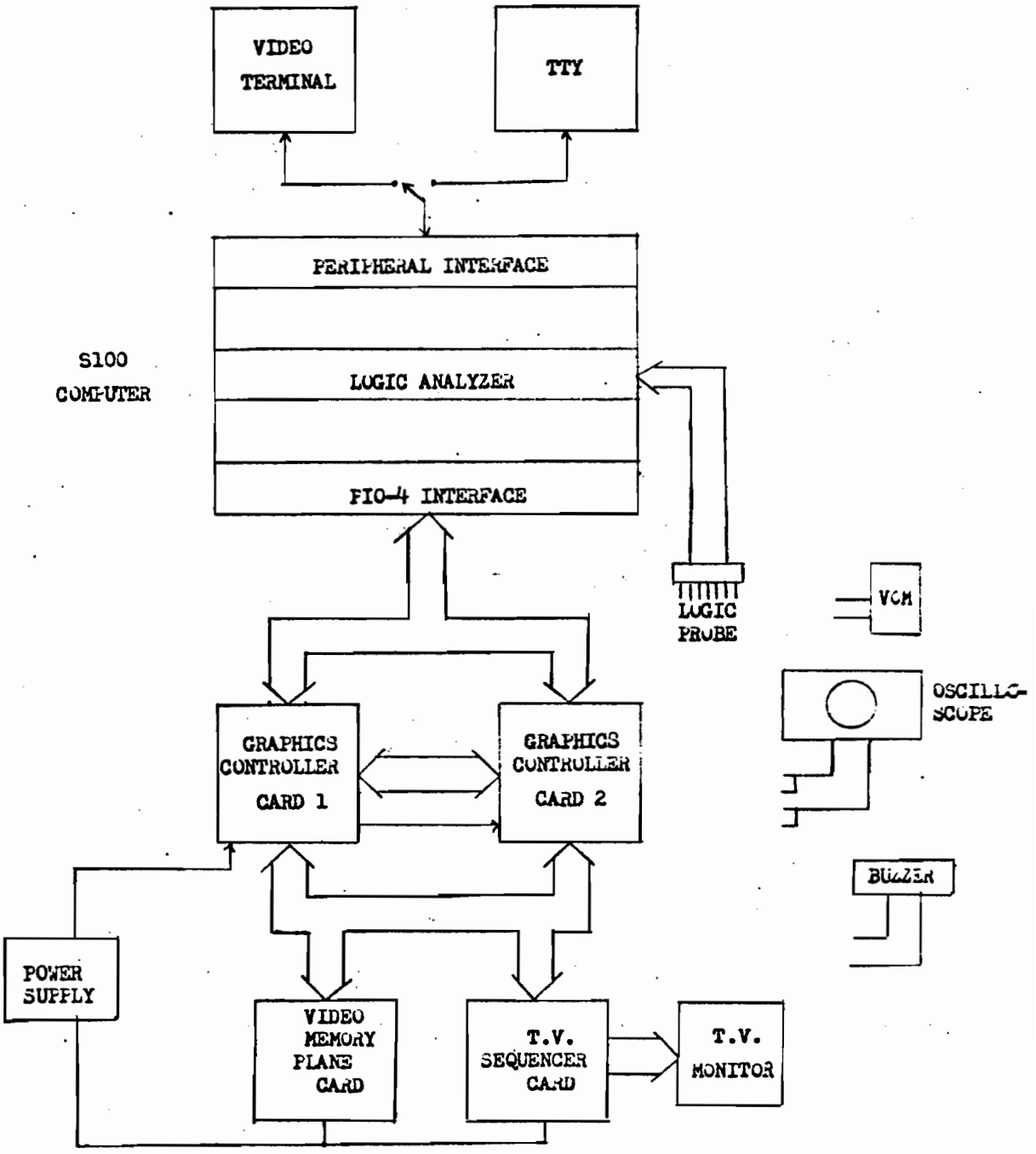
DEBUGGING FLOW CHART

Problems \ Equipment	VOM	Probes	Sign. Ana	Osc	D.D.A.	ICE	EMU
- Shorts, open wrong voltages	Yes	Maybe	No	Yes	Maybe	Maybe	No
- Bad resistors, capacitors	Yes	No	No	Yes	No	No	No
- Unknown logic signals	Yes	Yes	Maybe	Yes	Yes	Maybe	No
- Software faults	No	No	No	Maybe	Yes	Yes	Yes

VOM - Volt, ohm, milliampmeter
Probe - logic probes
Sign. Ana. - Signature analyzer
D.D.A. - Digital domain analyzer
ICE - in circuit emulator
EMU - software emulator or simulator

TABLE 6.4

PROBLEMS AND TOOLS FOR CIRCUIT TESTING



F I G U R E 6.3

THE TEST SET-UP FOR THE DISPLAY SYSTEM OF GRADS

code provided a test to check the RAMs and the associated wiring. The PIO-4 was used as the interface between the S100 and the display system hardware. Other programs were written to emulate the microcomputer activity on the Microbus after a software generated reset was used to initialize the necessary circuits on all the three modules of the display system.

Using the available tools and techniques, the T.V. Sequencer was first debugged and integrated with a Video memory plane and a black and white T.V. monitor. The Header and Transaction control machines of the Graphics Controller were then tested together with the Video memory and the T.V. Sequencer modules. Here the S100 computer facility was used to emulate the microcomputer activities to the Microbus. DMA transactions to the Video memory system were successfully achieved but at less than the rated bus speeds due to the execution time requirements of these emulation programs. Due to the concurrent debugging of the micro-computer modules as a separate project, they were not available for integration with the display system and hence they are not included in the present study.

CHAPTER VII

CONCLUSION

A real-time color animation display system for the Graphics and Image Processing Laboratory in the Electrical Engineering Department at McGill has been presented. The system uses a DEC, VAX-11/780 host computer for handling the complex data structures associated with computer generated images and for supervising several high speed microprocessor satellites. These microprocessor satellites are based on the AMD 2900 bit slice processor family. They work in parallel to achieve the high throughput rates required for the flicker-free animation of color raster displays on a T.V. monitor. A Host Computer Interface module interconnects the VAX 11/780 with these micro-computers to yield a general purpose multiprocessor system. A Graphics Controller links the microprocessor memories with a video frame buffer. By using a double buffer arrangement, this video frame buffer feeds a Sequencer module that is responsible for driving the CRT display at a refresh rate of 30 frames per second.

In this presentation, the display system of the whole real-time animation system was examined in detail. Particular emphasis was made on the Graphics Controller, an asynchronous machine, which is capable of arbitrating the multi-time-shared buses of the display system and also of interpreting the information from the microcomputer's local memory. The interpretation of the data depends on the type of graphic

mode the system is using. The four graphic modes that the system uses are based on painting of an area in a solid uniform color; displaying randomly positioned points of a selected color for drawing vectors or curves; shading an area with uniform color variation in the horizontal direction and the reading back of color information for picture analysis and maintenance. The status word of the header information preceding data to be put on the bus, is software programmable from the Host Computer. Hence the designer (or user) can, by programming, determine the graphic mode the system should use, and other features that determine the type of resolution, whether the picture is opaque or transparent, whether it is in color or black and white and the number of memory planes used.

The implementation of the interpreting circuitry of the Graphics Controller uses the synchronous ROM Sequencer concept.

The performance of the four graphic modes and that of the display system was evaluated. It was seen that when polygons of a single color shade have to be implemented, the Solid Area mode was the best. It can be used to paint backgrounds like the sea or the sky at a speed which is unmatched by the other modes. The Point mode was found to be very efficient for displaying vectors or curves of a single color. For displaying polygons with varying colors or shading the Shading Area mode was found to be the most efficient.

It was also seen that the display system performance can be improved by increasing the bandwidths of the Video memory and the Local

memory systems. The flexibility and modularity of the system architecture adapts well to such changes.

The system performance analysis also investigated the microcomputer array requirements for the present graphics system. Two such units were recommended to support the displays in both the low resolution and the high resolution.

Finally the Graphics Controller was tested and successfully integrated with the T.V. Sequencer and the Video memory plane.

The future prospects for fully implementing GRADS look very bright. With the completion of the debugging of the micro-computer and its support software, both of which are in progress, complete system integration is expected during the summer of 1979.

APPENDIX I

The RAM output bits for the Header Machine are assigned as follows:

OUTPUT	SIGNAL NAME	ASSERTED WHEN
Y1	LOAD μCOMP LOC MEM ADDR REG	LOW
Y2	LOAD BUFFER SIZE REG	LOW
Y3	LD COMMAND & STATUS REG	LOW
Y4	LOAD NUMBER REG	LOW
Y5	START TRANSACTION	HIGH
Y6	LOAD VIDEO ADDRESS REGISTER	HIGH
Y7	LOAD COLOR REGISTER	LOW
Y8	DO NOTHING μPROC	HIGH

Table A1 gives the microcode stored in the Header machine RAMS at the given addresses.

TABLE A.1. MICRODE FOR THE HEADER MACHINE

ADDRESS					DATA							
A4	A3	A2	A1	A0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8
0	0	0	0	0	1	1	1	1	0	0	1	1
0	0	0	0	1	0	1	1	1	0	0	1	0
0	0	0	1	0	1	0	1	1	0	0	1	0
0	0	0	1	1	1	1	0	1	0	0	1	0
0	0	1	0	0	1	1	1	1	0	1	1	0
0	0	1	0	1	1	1	1	0	1	0	1	0
0	0	1	1	0	x	x	x	x	x	x	x	x
0	0	1	1	1	x	x	x	x	x	x	x	x
0	1	0	0	0	1	1	1	1	0	0	1	1
0	1	0	0	1	0	1	1	1	0	0	1	0
0	1	0	1	0	1	0	1	1	0	0	1	0
0	1	0	1	1	1	1	0	1	0	0	1	0
0	1	1	0	0	1	1	1	1	0	1	1	0
0	1	1	0	1	1	1	1	0	1	0	1	0
0	1	1	1	0	x	x	x	x	x	x	x	x
0	1	1	1	1	x	x	x	x	x	x	x	x
1	0	0	0	0	1	1	1	1	0	0	1	1
1	0	0	0	1	0	1	1	1	0	0	1	0
1	0	0	1	0	1	0	1	1	0	0	1	0
1	0	0	1	1	1	1	0	1	0	0	1	0
1	0	1	0	0	1	1	1	1	0	0	0	0
1	0	1	0	1	1	1	1	0	1	0	1	0
1	0	1	1	0	x	x	x	x	x	x	x	x
1	0	1	1	1	x	x	x	x	x	x	x	x
1	1	0	0	0	1	1	1	1	0	0	1	1
1	1	0	0	1	0	1	1	1	0	0	1	0
1	1	0	1	0	1	0	1	1	0	0	1	0
1	1	0	1	1	1	1	0	1	0	0	1	0
1	1	1	0	0	1	1	1	1	0	1	1	0
1	1	1	0	1	1	1	1	1	0	0	0	0
1	1	1	1	0	1	1	1	0	0	0	1	0
1	1	1	1	0	1	1	1	0	0	0	1	0

APPENDIX II

The RAM output bits for the Transaction Machine are assigned as follows:

OUTPUT	SIGNAL NAME	ASSERTED WHEN
Y1	LOAD VIDEO ADDRESS REGISTER	HIGH
Y2	LOAD COLOR REGISTER	LOW
Y3	DO NOTHING μPROC	HIGH
Y4	INCREMENT VIDEO ADDRESS REGISTER & INCREMENT NUMBER REG	LOW
Y5	PIXEL SELECT B	HIGH
Y6	PIXEL SELECT A	HIGH
Y7	MICROBUS READ LOC MEM	HIGH
Y8	DO NOTHING VIDEO MEMORY	HIGH

Table B1 gives the microcode stored in the Transaction Machine RAMS at the given addresses.

TABLE B.1. MICROCODE FOR THE TRANSACTION MACHINE

ADDRESS					DATA							
A4	A3	A2	A1	A0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8
0	0	0	0	0	0	1	1	1	0	1	1	0
0	0	0	0	1	0	1	1	1	0	1	1	0
0	0	0	1	0	0	1	1	1	0	1	1	0
0	0	0	1	1	0	1	1	1	0	1	1	0
0	0	1	0	0	0	1	1	1	0	1	1	0
0	0	1	0	1	0	1	1	1	0	1	1	0
0	0	1	1	0	0	1	1	1	0	1	1	0
0	0	1	1	1	0	1	1	1	0	1	1	0
0	1	0	0	0	0	1	1	1	0	1	1	0
0	1	0	0	1	0	1	1	1	0	1	1	0
0	1	0	1	0	0	1	1	1	0	1	1	0
0	1	0	1	1	0	1	1	1	0	1	1	0
0	1	1	0	0	0	1	1	1	0	1	1	0
0	1	1	0	1	0	1	1	1	0	1	1	0
0	1	1	1	0	0	1	1	1	0	1	1	0
0	1	1	1	1	0	1	1	1	0	1	1	0
1	0	0	0	0	0	1	1	1	0	0	0	1
1	0	0	0	1	0	0	1	0	0	0	0	0
1	0	0	1	0	0	1	0	1	0	0	0	1
1	0	0	1	1	0	0	0	0	0	0	0	0
1	0	1	0	0	0	1	1	1	0	0	1	1
1	0	1	0	1	0	0	0	1	0	0	1	1
1	0	1	1	0	0	1	1	0	0	0	1	0
1	0	1	1	1	0	0	0	0	0	0	1	0
1	1	0	0	0	1	1	1	1	0	0	1	1
1	1	0	0	1	1	1	0	1	0	0	1	1
1	1	0	1	0	0	1	1	0	0	0	1	0
1	1	0	1	1	1	1	0	0	0	0	1	0
1	1	1	0	0	0	1	1	1	1	0	1	1
1	1	1	0	1	0	1	1	0	1	0	1	0
1	1	1	1	0	0	1	1	0	1	1	1	0
1	1	1	1	1	0	1	1	0	0	1	1	0

REFERENCES

[1] Advanced Micro Device Catalogue, "AM 2900 Bipolar Microprocessor Family", 1976.

[2] Baecker, R.M., "Picture-Driven Animation", AFIPS SJCC, 1969.

[3] Baer, J.L., "Multiprocessing Systems", IEEE Trans. on Comp., Vol. C-25, No. T2, December 1976, pp. 1271-12.

[4] Bitzer, D.L. and Slottow, H.G., "The Plasma Panel - A Digitally Addressable Panel with Inherent Memory", 1966 Fall Joint Comput. Conf., ARIPS Conf. Proc., Vol. 29, Washington, D.C., Spartan Press, 1966, p. 541.

[5] Botulynsky, G., and So, W., "Interleaving Color Video Memory Plane", Internal Report., Electrical Engineering Department, McGill University, June 1977.

[6] Burr-Brown Research Corp., Data Sheet on "Monolithic Microcircuit Digital-To-Analog Converter",1976.

[7] Cromemco Inc., User Manual for RDOS and CDOS, 1978.

[8] Crow, F.C., "Shaded Computer Graphics in the Entertainment Industry", IEEE Comp. Magaz., Vol. 11, No. 3, March 1978.

[9] Databyte Inc., "Datalyzer Users' Manual and Assembly Instructions", January 1978.

[10] Davis, S., "Computer Data Displays", Englewood Cliffs, N.J., Prentice Hall, Chapter 9, 1969.

[11] De Anza Systems Incorporated, Briefs on ID 5000 Series of Image Display System.

[12] Deekshatulu, B.L., "Digital Processing of Biomedical Pictures", J. Inst. Electron. and Telecommun. Eng. (India), Vol. 22, No. 5, May 1976, pp. 276-280.

[13] Dietmeyer, D., "Logic Design of Digital Systems", Second Edition, Allyn and Bacon Inc., 1978.

[14] Digital Equipment Corp., "VAX 11/780 Architecture Handbook", 1977.

[15] Everett, R.R., "The Whirlwind I Computer", Review of Electronic Digital Computers, Joint AIEE - IRE Conference, February 1952, p. 70.

[16] Flynn, M.J., "Very High-Speed Computing Systems, IEEE Proceedings, Vol. 54, No. 12, December 1966, p. 1901.

[17] Foley, J.D., "A Tutorial on Satellite Graphics Systems", IEEE Computer Magazine, Vol. 9, No. 8.

[18] Foley, J.D., "Software for Satellite Graphics Systems", Proc. of the ACM 1973 Annual Conference, pp. 76-80.

[19] Genisco Computers, Inc., Irvine, Ca., "GCT-3011 Programmable Graphics Processor".

[20] Graff, H. and Martell, R., "A Display Screen with Controlled Electroluminescence", Information Display, Vol. 2, September 1965, pp. 53-57.

[21] Guruswamy, V., Kristof, S.J., Baumgardner, M.F., "Digital Analysis of Landsat Data for Geological Studies in Sangagiri - Tiruchengode - Namakkal Area in Tamilnadu, India", IEEE 4th Annual Symposium on Machine Processing of Remotely Sensed Data, U.S.A., 1977.

[22] Herman, G., and Liu, H., "Three-Dimensional Display of Human Organs from Computed Tomograms", Computer Graphics and Image Processing International Journal, Vol. 9, No. 1, January 1979.

[23] Huggins, W.H. and Entwisle, D.R., "Computer Animation for the Academic Community", AFIPS, SJJC, 1969.

[24] Kajiya, J.T., Sutherland, I.E. and Cheadle, E.C., "A Random-Access Video Frame Buffer", Proc. Conf. Comput. Graphics, Pattern Recognition Data Structures, IEEE Comp. Soc., May 1975, p.1.

[25] Kashtan, D., "A Colour Television Sequencer", Internal Report, Electrical Engineering Department, McGill University, June 1977.

[26] Kristof, S.J., et al, "Comparing Soil Boundaries Delineated by Digital Analysis of Multi-Spectral Scanner Data from High and Low Spatial Resolution Systems", IEEE 4th Annual Symposium on Machine Processing of Remotely Sensed Data, U.S.A., 1977.

[27] Lamarre, J., "A Multiprocessor Interface for a Graphics Display System", M.Eng. Thesis, McGill University, 1979.

[28] Lamarre J-Y., Jankowski, R., "The Local Memory", Internal Report, McGill University, May 1978.

[29] Lechner, B.J., "Liquid Crystal Displays", in Pertinent Concepts in Computer Graphics, M. Faiman and J. Nievergelt, eds., Urbana, Il. : University of Illinois Press, 1969, p. 1.

[30] Lesea, A., and Zaks, R., "Microprocessor Interfacing Techniques", Sybex, 1977.

[31] Licklider, J.C.R., "A Picture is Worth a Thousand Words - and It Costs ....", AFIPS, SJCC, 1969.

[32] Licklider, J.C.R., "Man Computer Symbiosis", I.R.E. Trans. Human Factors Electron., Vol. HFF-1, March 1960.

[33] Liebowitz, B.H., "Multiple Processor Minicomputer Systems - Part 1 : Design Concepts", Computer Design, October 1978.

[34] Lin, W., "Microprocessor Based Digital System Design Fundamentals and the Development Laboratory for Hardware Designers and Engineering Executives", Proceedings of IEEE, Vol. 65, No.8, August 1977.

[35] Lucas, H.C., "Performance Evaluation and Monitoring", Computing Surveys, Vol. 3, No. 3, September 1971, pp. 79-90.

[36] Machover, C., "A Brief Personal History of Computer Graphics", Computer, November 1978.

[37] Machover, C., Neighbors, M., Stuart, C., "Graphics Displays", IEEE Spectrum, August 19-7, pp. 24-32.

[38] Machover, C., "State of the Art, Computer Aided Drafting", Reprographics, United Business Publications, New York, N.Y., February 1976, p. 16.

[39] Mignot, A., Payrebrune, M., "Design of a Video Memory Board", Internal Report, Electrical Engineering Department, McGill University, December 1978.

[40] Mohamed, J., Ramji, S., Solyom, J.P., "A Bus Arbitrator for the Color Video Frame Memory System", Internal Report, Electrical Engineering Department, McGill University, April 1977.

[41] Myers, W., "Interactive Computer Graphics: Poised for Takeoff", IEEE Comp. Magaz., Vol. 11, No. 1, January 1978.

[42] National Semiconductor Corp. Handbook on Linear Integrated Circuits, "Current Amplifier".

[43] National Semiconductor Corp., Handbook on MOS Integrated Circuits, 1974.

[44] Newman, W.M., and Sproul, R.F., "An Approach to Graphics System Design", IEEE Proc., Vol. 62, April 1974, pp. 471-483.

[45] Newman, W.M., and Sproul, R.F., "Principles of Interactive Computer Graphics", McGraw-Hill, Chapters 1 and 17.

[46] Newman, W.M., "Trends in Graphic Display Design", IEEE Transactions on Computers, Vol. C-25, No. 12, December 1976.

[47] Norpak Product, Briefs on "Visual Data Processor", and "Micro Video Processor".

[48] Papapetros, A., "The Design of a Color Display System for Real-time Animation using Microprocessors", M.Eng. Thesis, McGill University, 1977.

[49] Ramji, S., "The Performance Evaluation of a T.V. Display System Designed for Real-Time Animation, and of the Graphic Modes that it Uses", Internal Report, Electrical Engineering Department, McGill University.

[50] Riesenfeld, R., "Current Trends in Computer Graphics", Comput. & Graphics, Vol. 3, No. 4, May 1978, pp. 115-122.

[51] Roberts, D.I., "3-D Interactive Graphics Assisted Diesel Engine Design", Mech. Eng., Vol. 100, No. 3, March 1978, pp. 10-45.

[52] Rogers, B., "The Fundamental Beauty of Computer Graphics", Tekscope, Vol. 9, No. 2, 1977, pp. 8 - 12.

[53] Rundle, A.R., "Software for Satellite Graphics", Proc. Computer Graphics, 1970.

[54] Signetics, Data Manual on Logic Memories, Interface, Analog, Microprocessor, Military, 1976.

[55] Special Issue on Parallel Processors and Processing, IEEE Trans. on Computers, Vol. C-26, No. 2, February 1977.

[56] Strand, E.M., et al, "Computer Applications in Clinical Cardiology", Proceedings of Canadian Information Processing Society Canada, 1975.

[57] Sutherland, I.E., "Sketchpad : A Man-Machine Graphical Communication System", AFIPS Proceedings, SJCC, 1963, p. 343.

[58] Svobodova, L., Computer Performance Measurement and Evaluation Methods : Analysis and Applications, Elsevier North-Holland, Inc., New York, 1976.

[59] Tabrizi, Moshtag, V., "The Microcode Memory", Internal Report, McGill University, May 1978.

[60] Tannas, L.E., and Goede, W.F., "Flat-Panel Displays : A Critique", IEEE Spectrum, July 1978.

[61] Technical Design Labs., Inc., Zapple Monitor Manual.

[62] Texas Instruments, Inc., "Bipolar Microcomputer Components Data Book for Design Engineers, January 1977.

[63] Texas Instruments, Inc., "The TTL Data Book for Design Engineers", Second Edition, 1976.

[64] Thurber, K., Parallel Processor Architectures - Part 1 : General Purpose Systems, Comput. Design, January 1979, pp. 89 - 97.

[65] To, R., Lee, A., "A Color T.V. Sequencer", Internal Report, Electrical Engineering Department, McGill University, April 1978.

[66] Van Dam, A., and Stabler, G.M., "Intelligent Satellites for Interactive Graphics", Proc. AFIPS, 1973 NCC., Vol. 42, pp. 229-238.

[67] Widdoes, L.C., "The Minerva Multimicroprocessor", in Proc. 3rd Symp. on Computer Architecture, 1976, pp. 34-39.