A Video Image Interface

by

Ricardo Eugenio Velez

B.S. Universidad Iberoamericana

(1977)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS OF THE
DEGREE OF

MASTER OF SCIENCE IN

ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1982

[i.e. June 1982]

Signature of Author [illegible]
Department of Electrical Engineering and Computer Science.

Certified by [illegible]
William F. Schreiber
Thesis Supervisor

Accepted by [illegible]
Chairman, Departmental Graduate Committee

TABLE OF CONTENTS

TABLE OF ILLUSTRATIONS

A VIDEO IMAGE INTERFACE

by

Ricardo Eugenio Velez

Submitted to the Department of Electrical Engineering
and Computer Science on May 7, 1982
in partial fulfillment
for the Degree of Master of Science in
Electrical Engineering and Computer Science

ABSTRACT

A technique has been designed and developed for storing high resolution color images at high speed rates. This interface contributes to the total construction of the Color System whose main goal is to print graphic art quality color pictures. This system has been developed in CIPG.

The video image interface is capable of "grabbing" images from a color monitor and storing them onto disk. These images are the result of previously processed images by the Color Translation Module which is capable of modifying selectively the hue and saturation of colors in images encoded digitally in a Luminance/Chrominance color space.

Due to real time constraints the video image interface uses a Direct Memory Access board to transfer data to the CPU memory, and achieve the most efficient method of grabbing and storing a color picture.

A prototype was built and required software developed as a standard IPS process.

Thesis Supervisor: Dr. William Schreiber

Title: Professor of Electrical Engineering.

ACKNOWLEDGEMENTS

First of all, I wish to thank Profesor William Schreiber for supervising this thesis. Special thanks also go to Profesor D.E. Troxel and Russell Walker who helped me throughout the project, and last, but not least, to Mike McIlrath who contributed with a great deal of software.

INTRODUCTION.

The color system described in this thesis is a multiuser, multiprocessing image processing system capable of input and output of a number of pictures simultaneously in real time. Its final goal is the reduction of cost and quality improvement of color pictures, printed by engraved cylinders.

Real time operating constraints are raised by the nature of picture scanners and the desire to provide quick responses to various parameters commanded by an operator, using a console or knobs such as the control panel of the Color Translation Module.

Obtaining desired colors is a characteristic of this system. This is a very complex problem, since the original copy is not perfect and exact reproduction of colors is almost impossible. Most of the color reproduction technologies introduce several color distortions due to physical properties of photographic materials. An example is Kodachrome film that darkens blue sky and brightens most other colors thus beginning a chain of distortions. Picture scanning, which converts colors and shapes into electrical signals also introduces distortions. No perfect image can be made without distortion when a transfer is made from one medium to another.

Engraving cylinders and plating of the copper also introduces distortions , and finally, light absorptions and interactions of dyes further distort reproduction.

The central element of the system is a high resolution color monitor that accurately displays the appearance of colors in either input or output images. Data fed to the color monitor is a set of Red, Green and Blue signals (RGB). These are converted to L, C1 and C2 values, and operated on in this form all through the system.

Once the desirable output colors is decided, the ink densities are calculated and compensated with the help of look-up tables. In order to engrave cylinders, the processed data has to be stored again onto disk, which is the purpose of this thesis.

L, C1 and C2 will reproduce the same image of the monitor that already has been processed and colors selected. These three values per pel (picture element), will be saved in three different files, one for each value type, so each can be used independently.

The use of a DR11-B DMA (Direct Memory Access), permits attaining the speed requirements for data transfers in this project.

A final data conversion is made to calculate ink densities. This data derives appropriate signals to engrave copper cylinders making possible color printouts using the Helio-Klischograph.

THE COLOR SYSTEM.

Description.

IPS, the brain of the system, is a multiuser computer for image processing, which has been developed to serve a variety of needs. The facility can be used to perform image manipulations such as enlarging and cropping, as well as data conversions to correct or modify colors and obtain excellent color pictures. Such applications requires quite a bit of special purpose hardware.

The primary goal is to provide computer power for image processing. Automated printout of color pages reduces costs and improve quality in graphic art color reproductions.

Since it is a multiple image processing system, and considerable buffer memory is required, the PDP 11/34 used for this purpose has to be modified, adding an Interbus Link which allows another 512 K words of memory to be addressed, independent of the 128 K word maximum original accessed by the PDP 11/34.

The nature of image processing requires parallel computation. Many operations must be performed in parallel for high speed image enlarging. The multiprocessing capability of the PDP 11/34 permits processing more than one picture at the same time. Engraving several copper cylinders at the same time is achievable and desirable.

A high resolution TV color monitor is required to display the input or final output picture, and is essential to apply the "Company Translation", which assures that output colors match with those seen on the screen.

A fast large capacity disk drive is mandatory to store several pictures at the same time. This is accomplished by a CDC 300 Megabyte disk drive. In addition the system has two extra 2.5 Megabyte disk drives.

The console is the main operator's interface with the system. This console is divided by software into three independent sections.

The top section is primarily for the user's command operations; the monitor process takes the commands and issues prompts. The center section is used to examine commands, reports results and also to enter instructions for the system editors. The bottom section is used to issue warnings, error messages, and changes in the system status.

Fig 1a. and 1b. show a simple overall block diagram of the system. The scanner outputs uncorrected color head data. These uncorrected values are converted by a look-up table into CIE values, and stored into disk. A TV resolution image is buffered in the PCTV memory, and is displayed directly on the monitor. Selecting data between the PCTV memory and the Color Translation Module is accomplished by the Color Interface Module multiplexer. If an input image is to be directly displayed, data stays in RGB form throughout the data path. No L,C1, C2 conversion is required so the conversion module is bypassed.

In the remote case that no changes have to been made to the input image, the Color Translation Module is bypassed and there is no need to save unchanged images on the disk.

The Color Translation Module facilitates color control printing and permits an operator to alter precisely the colors in an image via a control panel. Results can be observed immediately on the color monitor. Fig. 2 shows the block diagram of the Color Translation Module.

The main applications of the CTM are:

a) Compensates color errors in the head values of source images.

b) Compensates alteration of values due to noise or any other cause inherent in the transfer of images from a physical material to electric signals and vice-versa.

c) Permits aesthetic changes in the image.

The Color Translation Module compensates systematic errors by means of the "Company Translation", that calculates ink densities required to produce a selected color. The system permits the user to observe his modifications on the TV monitor in real time. If the output picture is not satisfactory he can adjust it with the translation controls.

The graduation module allows the user to make changes in tone separation (shadow contrast), and gives values to the darkest and lightest part of the picture.

The neutral color balance controls the overall color cast of a picture, affecting the highlight, midtone and shadows of the image separately.

The selective color correction allows the user to shift colors towards neighboring hues. This module modifies selectively the pels which fall within one of the seven hue domains (Blue, Magenta, Red, Orange, Yellow, Green and Cyan).

The special color correction enables an operator to make changes to very specific colors in an image, defining a narrower hue domain.

The cartesian to polar coordinate converter is required to process data through the CTM. The polar to cartesian converter returns signals to their original form.

When the full resolution image is processed and satisfactory results are obtained, then it is stored on disk via the Television Acquisition Module (TVAM).

The TVAM is the subject of this thesis, and consists of the video image interface that "grabs" the picture, and a Direct Memory Access controller (DR11-B), to make possible data transfers at high speeds. Data can be stored in either RGB or L, C1, C2 form. The second is better because it saves disk area. Pictures stored in L,C1, C2 occupy 1/3 less space than RGB files. The TVAM will be described in more detail.

The Color Interface Module comprises all the hardware required to make possible the display on the TV monitor of images in either form (RGB or L, C1, C2). Selectors and converters make proper changes in the data path.

A very wide gamut of colors is obtained with highly saturated quantities on Red, Green and Blue phosphors. Using this gamut we can simulate color printing. RGB values are fed to a Tone Scale memory and accurately converted into analog signals required by the monitor.

The color monitor is not an optional peripheral to the system. On the contrary it is an absolute necessity since color printing depend on the reliability of the appearance of colors on the TV screen. Even if the adjustments in the high resolution TV are not exactly as specifications recommend, the translation input to output will still be the correct.

When all pages have been successfully processed and stored into disk by the TVAM, engraving can be done (see Fig. 1b). The images are retrieved form disk, decoded and converted into ink densities. This process is done basically by look-up tables.

Color printing and photography use a totally different approach to synthesize colors from the one used by the TV monitor. It is not the object to replicate the same wavelength distribution of intensities to reproduce colors, but to perceive identical appearance using various amounts of three primaries.

The additive method to obtain colors is the sum of light energies of each component wavelength. The subtractive method can be thought as a filter that subtracts out certain colors, although this method is better called multiplicative because the product of the transmittances of colors is given by the transmittance of each wavelength. Fig. 3 shows both methods of reproducing colors using for the additive Red, Green and Blue, and for the subtractive Magenta, Yellow and Cyan.

Ink mixing is more nearly subtractive than additive, but other problems occur, such as:

a).- Inks are not completely transparent. Inks absorb wavelength intensities that they should not.

b).- The need of a good K-algorithm to find the amount of Black ink needed and the reduced densities D'r, D'g, D'b when K is removed from a three color mixture specified by Dr, Dg, Db.

c).- The gamut of colors is restricted somewhat. We can reach all possible colors if we begin with a set of 512 correct three-color entries which go from zero to maximum density of each ink. But with the addition of Black ink to these colors, we do not get all possible four color combinations.

d).- The dynamic range is also diminished. Adding black to these colors decreases luminance, extending the gamut into higher densities.

The three primary colors used in additive methods usually are RGB; these three produce a wide gamut of colors. The same gamut could be achieved by controlling reflected light subtractively if each of the inks absorbed in only one of the primary wavelengths. Yellow, Cyan and Magenta are selected for this purpose, although typical inks are not perfect.

The addition of a fourth ink in color printing presents an extra problem, but has several advantages. It makes color balance less critical, tends to minimize the effect of misregistration and reduces the quantity of expensive inks to achieve a desired color. The amount of black inks is calculated from the original YMC signals.

The process of replacing colored ink by Black is called "Undercolor Removal" (UCR). It is possible to have more black in dark areas where color is not so critical and save more colored ink but in light areas the saving is minimal.

The UCR algorithm is:

D'r = (Dr-K)/(1-asubrK), asubr = .77 *

D'g = (Dg-K)/(1-asubgK), asubg = .87 *

D'b = (Db-K)/(1-asubbK), asubb = .65 *

(* This factor should be scaled if the ink densities are scaled).

This algorithm can be implemented by the construction of two LUT's without interpolation. The algorithm and B matrix are invertible, so Black (K) can be calculated from Drgb. However the resulting K algorithm is not very simple and has to be implemented by another LUT.

Several printing processing techniques have been proposed and deeply studied. One that is workable and presently been developed is shown in fig. 4.

Drgb signals are the logarithm of the originals RGB and are the densities of ideal subtractive inks. They are applied to an Under Color Removal algorithm to calculate colored ink removal due to Black quantity which is calculated by the K algorithm. The densities of Drgb have to be in the range of 0 to 1.9 to represent all the printable colors, using 8 bit values.

The linear B matrix defines the theoretical dyes in terms of their absorption density in each of the three wavelengths bands. These ideal dyes absorb some light in all three wavelengths although they are "perfectly subtractive dyes", and have densities very near to those actually used.

Due to the fact that inks are not perfectly subtractive, (transparent), lookup table T2 is required to calculate the small errors given by the B matrix.

At last a Tone Scale memory is designed to output the desired ink densities. This compensates for the distortions caused by the Helio, obtaining the desired densities.

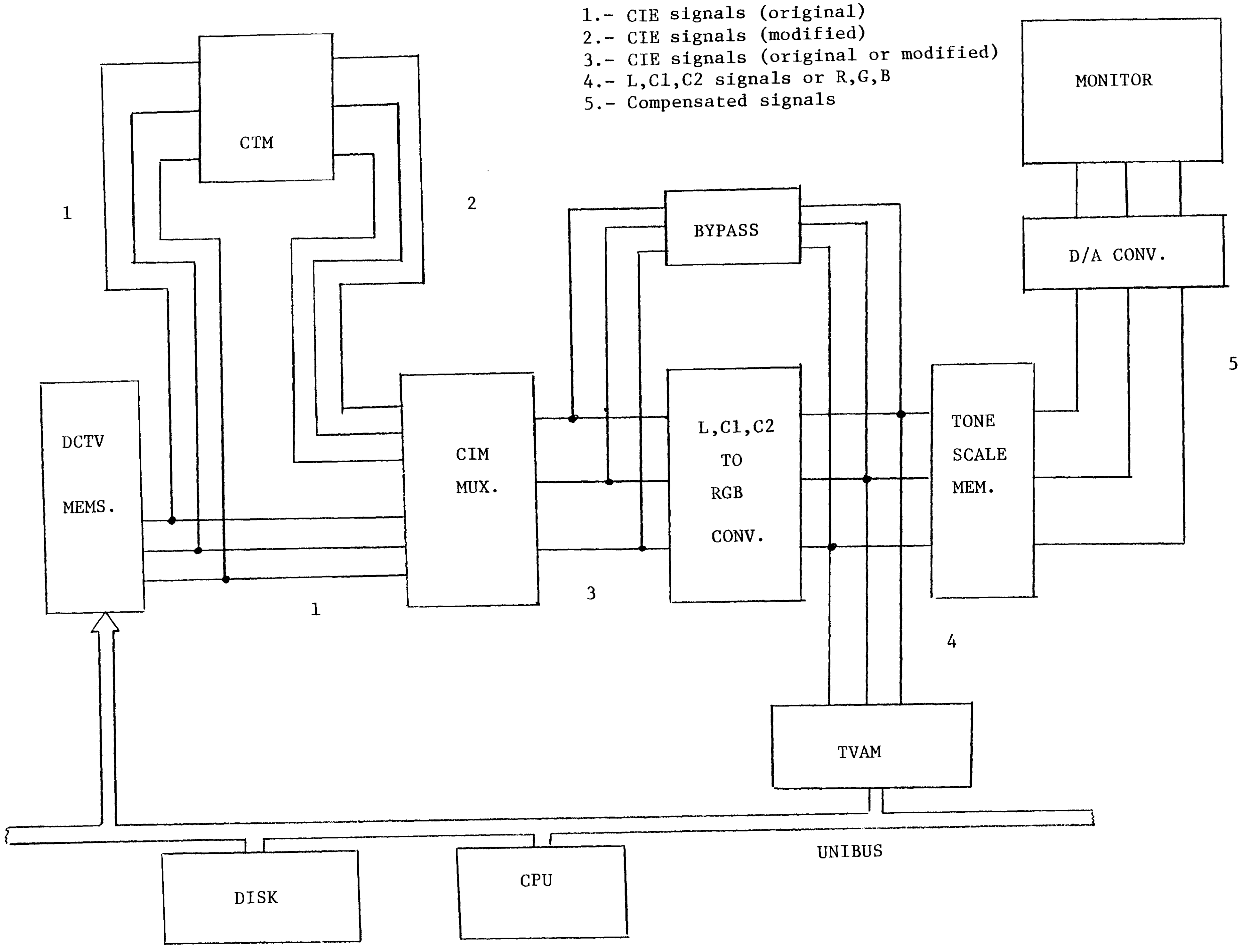
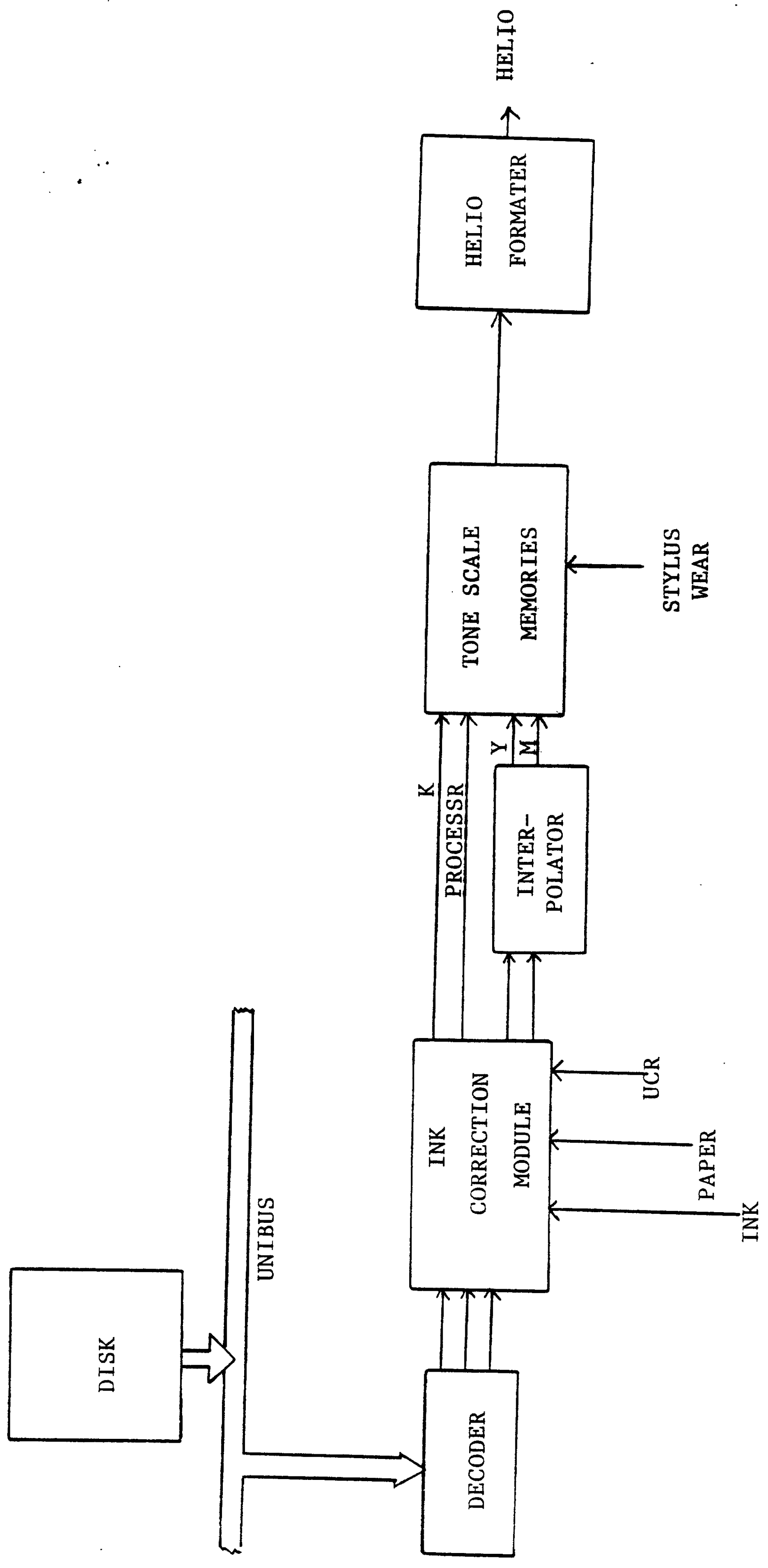
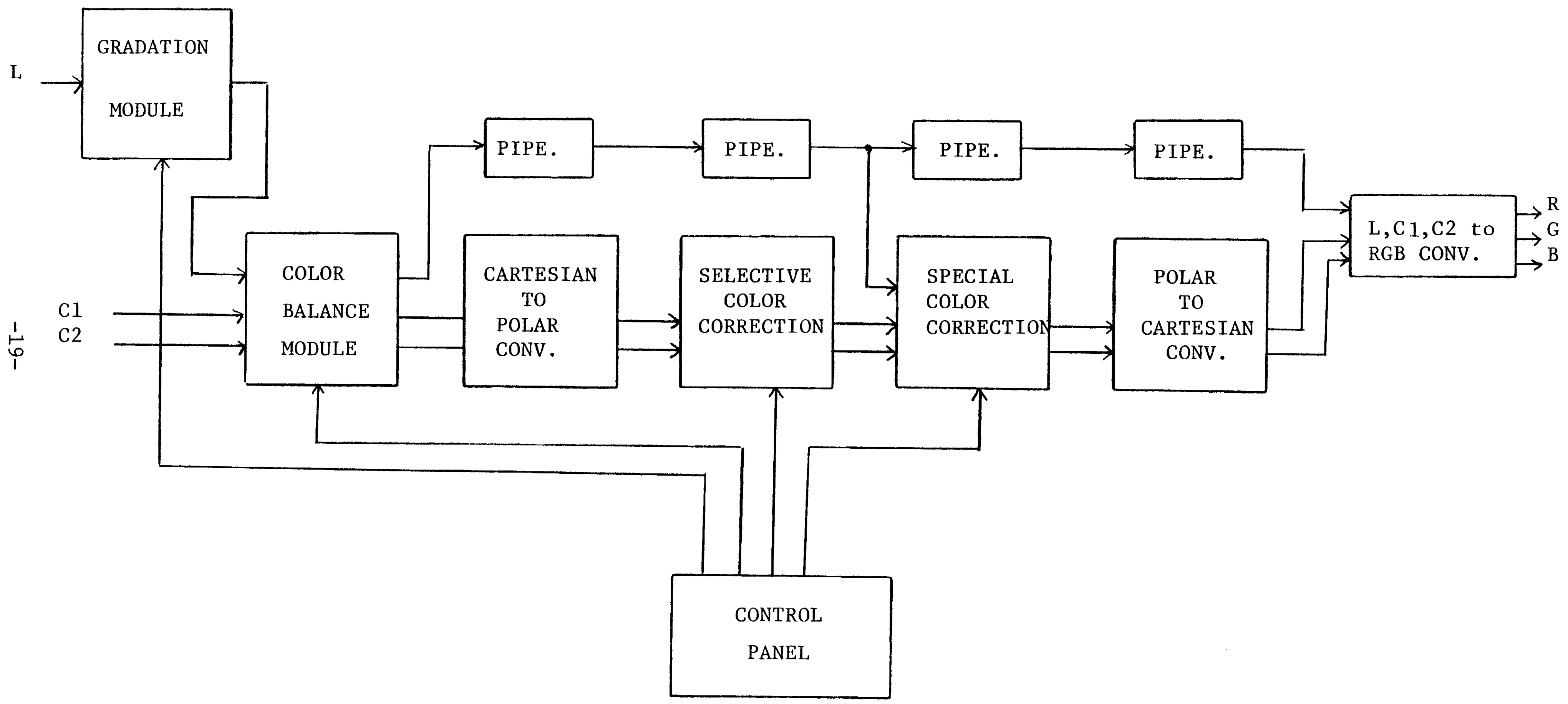


Fig. 1a. Color System Block Diagram



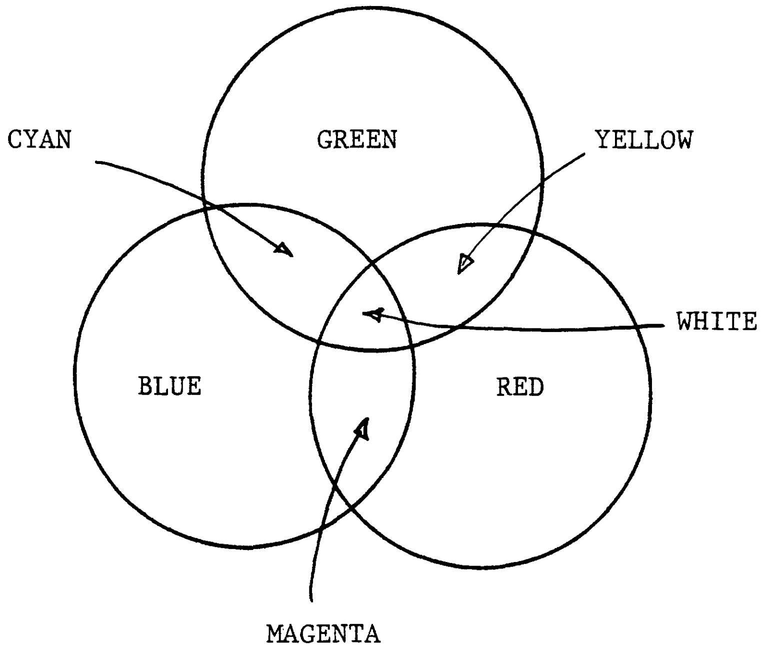
Engraving Process Block Diagram

Fig. 1b.

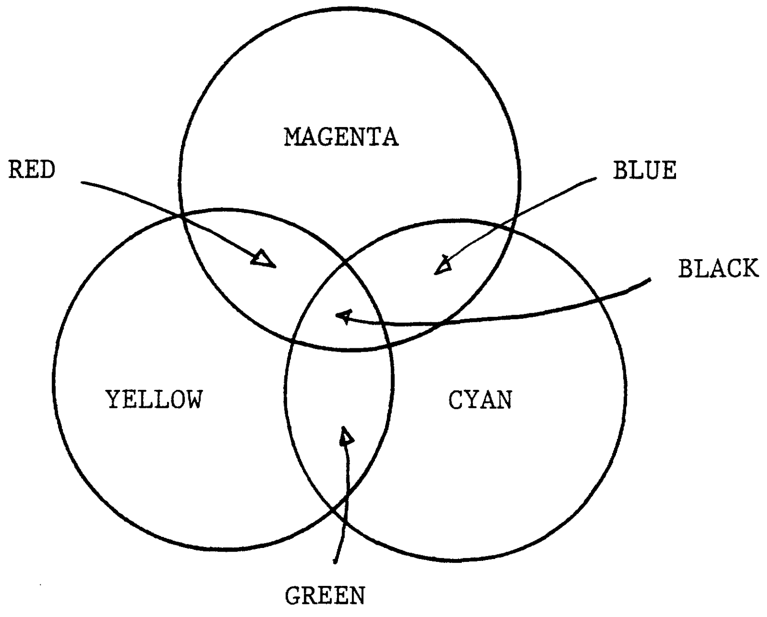


Color Translation Module Block Diagram.

Fig. 2

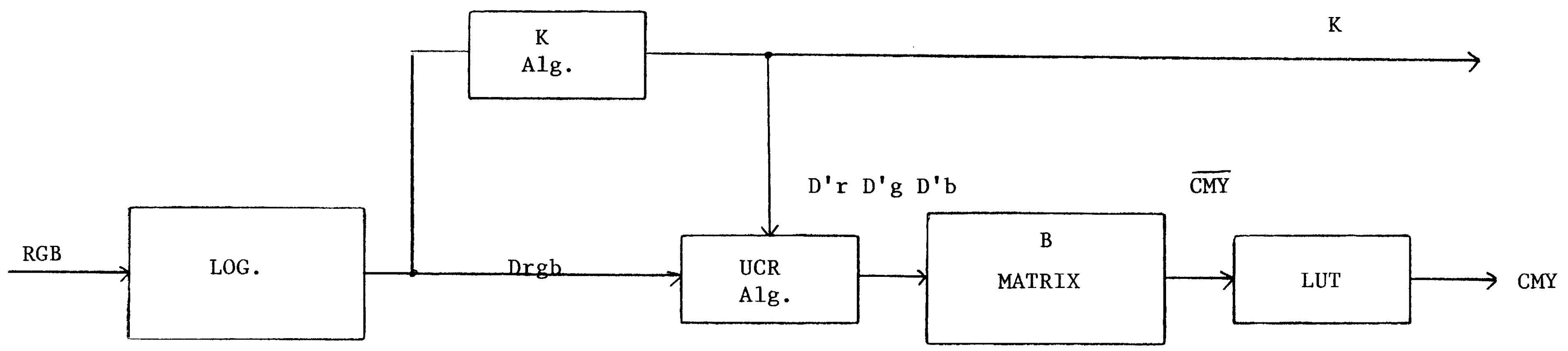


Rules of Additive Color Mixing



Rules of Subtractive Color Mixing

Fig. 3



Four Color System Block Diagram.

Fig. 4

DESIGN GOALS.

L,C1,C2 are linear transformation values of R,G and B signals. This triplet is represented by three eight bit binary numbers. The CTM requires this form of representation to be transformed into polar coordinates to facilitate data manipulation, and then to be retransformed to its original mode.

Cartesian to polar coordinate conversion gives us two values S and H, where S represents the magnitude and has a one-to-one correspondence with the saturation of a pel multiplied by its luminance, and H represents the angle and has a direct correspondence with the hue of that particular pel. Luminance can be defined as the brightness of each pel.

The object is to store L, C1 and C2 signals, where L has a bandwith of 10 megahertz and a digital range of values between 0 and 255. C1 and C2 each has a bandwith of 5 megahertz and ranges from -128 to 127.

To achieve good quality color images, 512 X 512 pels are scanned in the color monitor, requiring the previously mentioned rates. The original full resolution pictures supplied by the scanner are stored on disk. The TV monitor displays a part of the high resolution picture; the picture buffered into the monitor's memory is called a TV resolution picture and it is the data to be grabbed and stored into disk.

Storing several TV resolution pictures will form a complete full resolution picture that has been processed and modified by the Color Translation Module.

The method chosen for this purpose is to grab continuous data blocks in several frames. The number of frames used depends on memory size and data transfers from the video interface to the CPU memory.

Each image component is stored in a separate and independent file, one for L, one for C1 and one for C2. All three combined will reproduce the processed image. C1 and C2 are half the size of the L file. If the picture is stored in RGB form, the three resultant files will have the same size.

Displaying pictures in real time requires very fast data transfer. A Direct Memory Access board such as the DR11-B is required to make possible fast picture storage using almost no CPU time.

HARDWARE.

A desirable way to grab an image is to store in one frame several blocks of successive data. This is possible using a pipeline scheme. Data is coming out at a 10 Megahertz rate, which can not be directly stored into memory without having an overrun causing data losses. A latch and holding technique is used for this purpose. This technique and a small buffer allows fast transfers that comprise a fast picture grabbing method.

The complete block diagram of the video interface is shown in fig. 5.

The image video interface is selected via the Device Selector.

The address of the interface is set by the 18 address switches of the video interface board. Whenever that address is valid in the UNIBUS a SSYNL is generated to inform the CPU that the interface exists and it is ready to perform an operation. Data is written in the Input Status register by the program to select L, C1 or C2 and the resolution of the picture to store. A GO signal is sent through the DR11-B and an operation begins.

Selecting the required function by software gives the ability to grab one or two of the three picture components (L, C1 or C2) of a particular image. The resolution is also software selectable. Sometimes the signals are half of the original frequency, in particular those proceeding from the CTM (C1 and C2). This half resolution is not very desirable and much effort has been made to correct this problem. The Video Image Interface foresees this difficulty making the resolution an option.

It is absolutely necessary to use static RAMs such as 2148, as no refresh time is available due to the real time constraints. The 2148 memory ICs have a capability of storing 1024 X 4 bits.

The addressing technique employed by this method plays an important role in determining the capacity and speed of the system. Using the same address we can store two values of L, C1 or C2 (one 16 bit word). Fig. 6 shows a schematic of how 2 bytes are buffered.

First data is latched into two sample-and-hold blocks. Each block latches different and subsequent sets of data having a total time of 200 nsec. for full resolution (100 nsec. for each set of data latched) or 400 nsec. for half resolution. This time is sufficient to address and access the buffer. Memory access time for the 2148s is 70 nanoseconds, and propagation delay time for the 10 bit counter is 80 nsec. giving a worst case total time of 150 nsec. This time is enough to settle all addresses and required signals to store one word of data.

Using four 2148s we have a total capacity of 2 K bytes, that permits storage of 4 consecutive horizontal lines (each horizontal line has a resolution of 512 pels).

After four lines have been grabbed and stored into the buffer, transfer to the CPU memory has to be made through the Direct Memory Access board (DR11-B). For this purpose a programmable number of horizontal lines of the monitor are not grabbed. This selection is made by switches on the video interface. An optimal selection has to be made to obtain maximum transfer rates without having overruns. Two switches make this selection, the possibilities are:

SW1	SW2	
OFF	OFF	28 lines.
ON	OFF	60 lines.
OFF	ON	124 lines.
ON	ON	252 lines.

There is no data transfer to the computer until the entire buffer is loaded to maximize transfers. When memory is completely full, a flag is transmitted to the computer's interface and a block transfer is performed. This maximizes the advantages of the direct memory access mode,(DMA is an interface that takes data from an external device and access it directly to the UNIBUS using almost no CPU time).

When a transfer is carried out, a second block is grabbed and so on, until the entire image is stored. Using 2148 RAMs, four horizontal lines are grabbed, the next 28 lines (if the line selection is 32), are used to have the required time to make the transfer to memory. This constitutes a grab-cycle, 16 of which are performed in one frame. Each frame has 512 X 512 pels. To store a complete image sixteen frames are used.

The data transfers made from the video interface to main memory through the DR11-B DMA board can be interrupted by any other task of the computer that has a higher priority level, making the transfer time not periodic. This has to be taken into consideration by the video interface. If the data transfer cannot be finished in the selected number of lines, an overrun occurs but no data is missed. The video interface sets an error flag warning that an overrun has occurred. When the transfer has been completed, the rest of that frame is released in order to get into synchronization with the next frame and grab the four lines that caused the overrun, recovering the normal grab-transfer technique.

It is obvious that the selection of the switches determines the incidence of overruns. The switches have to be selected to obtain best data transfers and avoid overruns, if possible. The overall time to grab an image is greatly affected by the overruns since a whole frame is needed to resynchronize.

Fig. 7a shows the timing signals of a two pel grab cycle and the corresponding chip-select (CS) strobe. The W/R signal together with the CS controls the writing or reading of the 2148s. The W/R signal does not change for the entire writing sequence, the CS providing adequate time. In fig. 7b we can see the timing signals on a read-transfer word from the video interface to the DR11-B controlled by the BUSY H strobe.

Good synchronization with the video data is essential. This is done by the horizontal and vertical retrace signals. These signals are used to precisely control the number of lines which are to be grabbed and transferred to the computer. Fig. 8 shows a sequence of blocks transferred which correspond to the pels contained in the shaded areas of the monitor.

During a storage cycle, it is very important that parameters stay constant until all data is grabbed and writing interface activity ends. This means that no variation of the Color Translation Module knobs is permissible until a picture already accepted is completely stored.

All timing signals are generated by the clock block. It generates the 5 Mhz. clock for the half resolution when requested, the latching clocks for the 74LS395 latches, the CS for the memory and the clock for the addressing counters.

The Data Selector chooses which of the three sets of data is to be grabbed (L,C1 or C2). This selection is controlled by software via the input status register. Two of the four registers are clocked and latch a pel. After 100 nsec. the other two are enabled and a 16 bit word (two pels) is ready to be written in the buffer. The 74LS395 IC was chosen for this purpose due to its tri-state capability required by the 2148 RAMs that shares input and output pins.

The selection of which register is enabled is done by the Control Selector Register.

Three 74LS161 counters are used for addressing the memory due to their look-ahead connection configuration that enables very high speeds counts. The 10 bit address has to be settled in less than 100 nanoseconds.

The most important register is the Transfer Control. This register participates in a variety of functions. It commands all activities in the video Image Interface. It has two main functions (grabbing or transferring) and behaves very differently in the two states. When it is writing to the buffer (grabbing data), the Transfer Control register inhibits data to be transferred to the DMA, until the buffer is completely loaded. When this is done, the Transfer Control Register changes to the reading cycle (transfers from buffer),and begins the transfer to the CPU. The DR11-B (DMA) becomes master and control the subsequent data request to the video interface. During this period no new data is grabbed until this activity ends.

If the transfer is interrupted by any other device with higher priority level, all activity stops to avoid any loss of data. When that interrupt is over, the DR11-B finishes the previous transfer and asks for another transfer to the Video Image Interface. If the interrupt was long enough (28, 60, 124, 252 lines switch selectable), an overrun is detected and latched to the Overrun Register.

The writing cycle, the Line Counter gives the actual count to the comparator deciding which are the lines to grab, matching with the number of transfers which already have been made. The Line Counter never stops even if the interface is reading and transferring data to the CPU. If a strobe indicates that another set of data has to be grabbed and the video interface is still in the transfer cycle, an overrun is generated and the rest of the frame is left free to resynchronize again with the line and resync. counters at the beginning of the next frame, so no data is lost. There is only a time increment penalty.

The line counter is programmed with the two switches previously described. This count is compared to the 6 more significant bits of the Transfer counter to properly obtain the lines to be grabbed. Fig. 9 describes how this is done.

The Request Control Register generates a Cycle Request to the DR11-B. A transfer is ready to be performed, 16 bits of data are transferred and the Busy H signal is set. No other cycle request can be made until the previous one ends and Busy H is cleared. After this is done another CR is generated until all the buffer is transferred.

The number of words to be transferred is loaded by software into the Word Counter Register of the DR11-B. This count must coincide with the total capacity of the Video Image Interface.

The picture to be stored has to be loaded to the PCTV memory in a very special form. This format is determined by the lines to be "grabbed" and to the even and odd scanning line technique used on the screen. This scrambled source image when captured and passed throughout the TVAM obtains the adequate sequence and the image stored on disk is then appropriate.

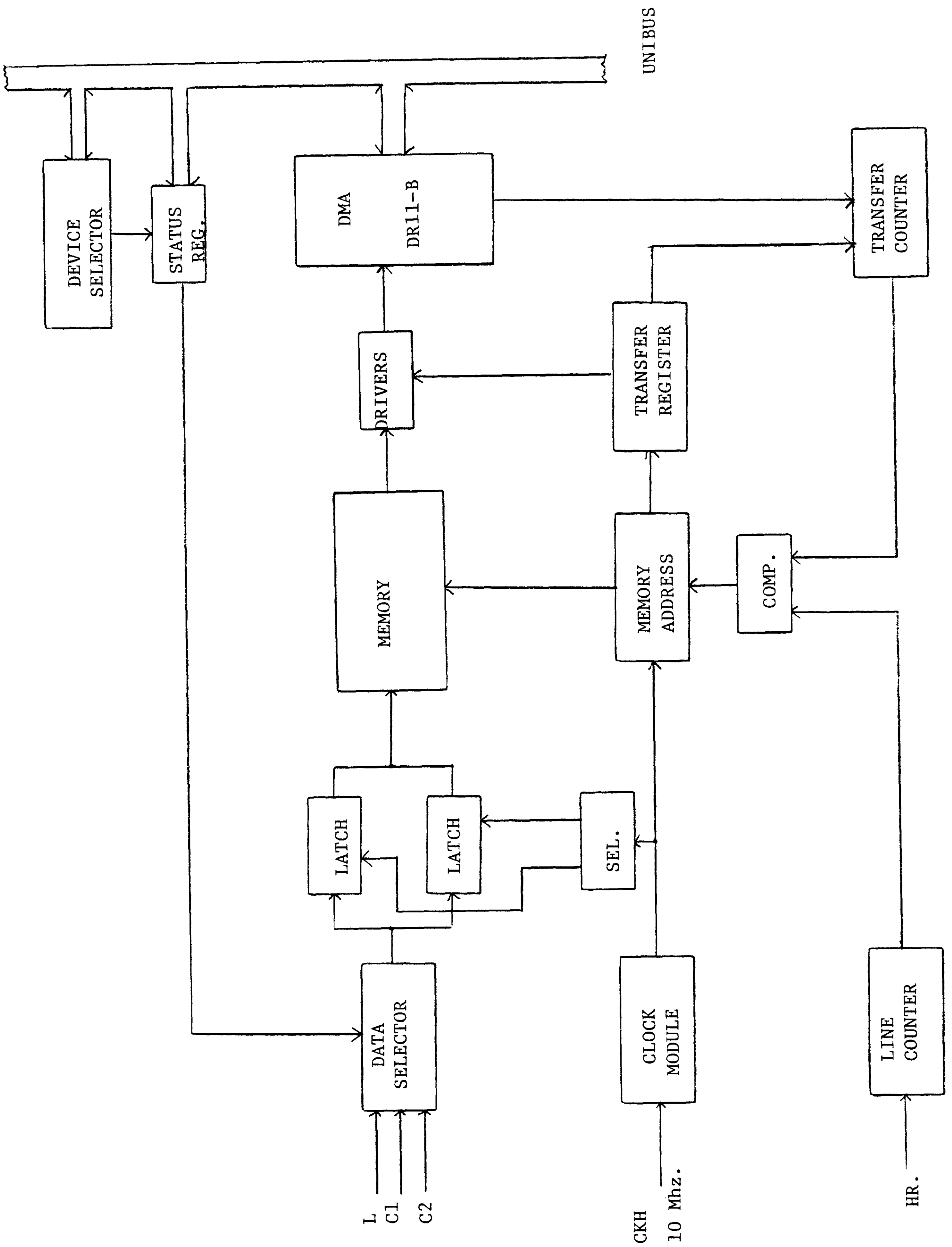
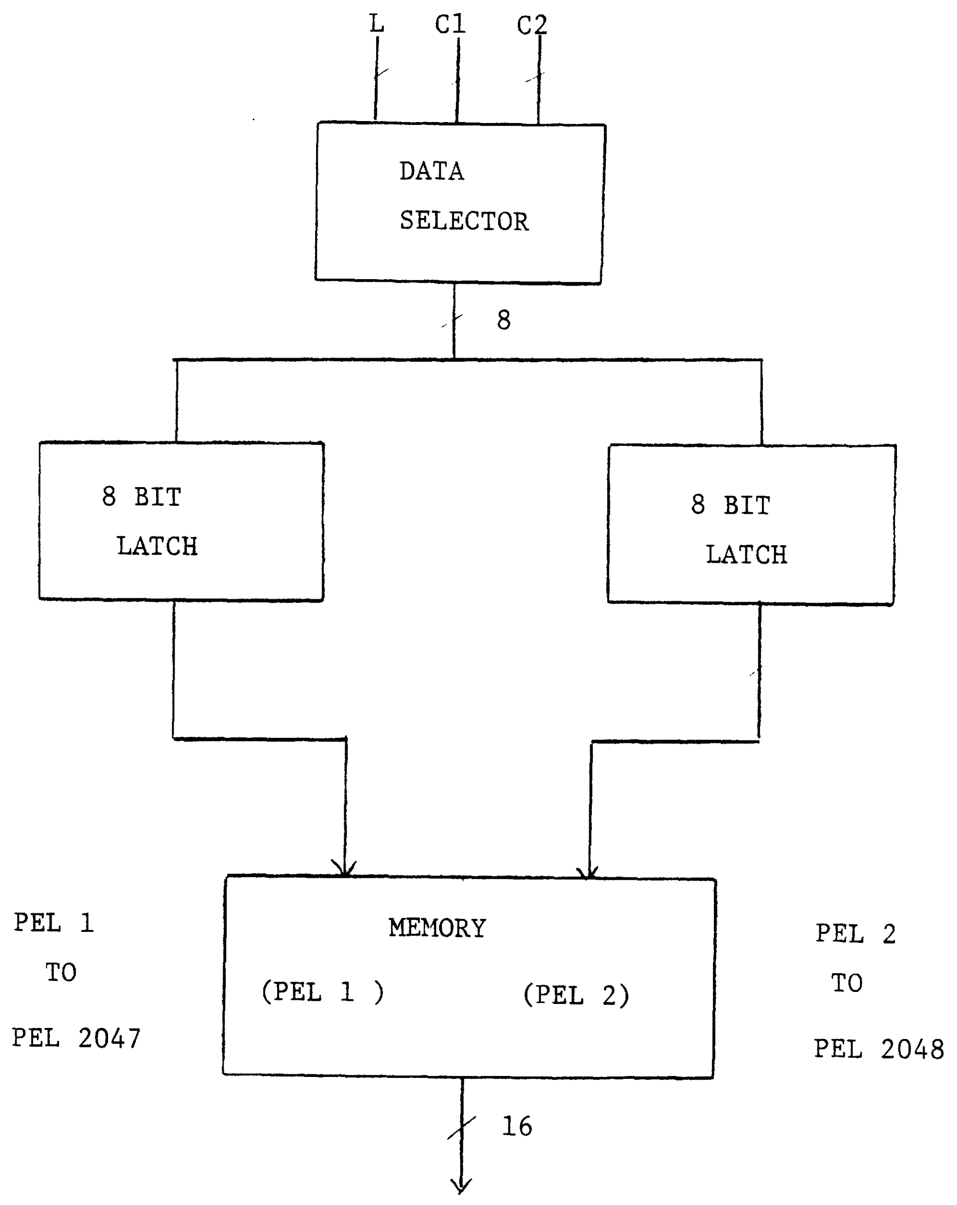


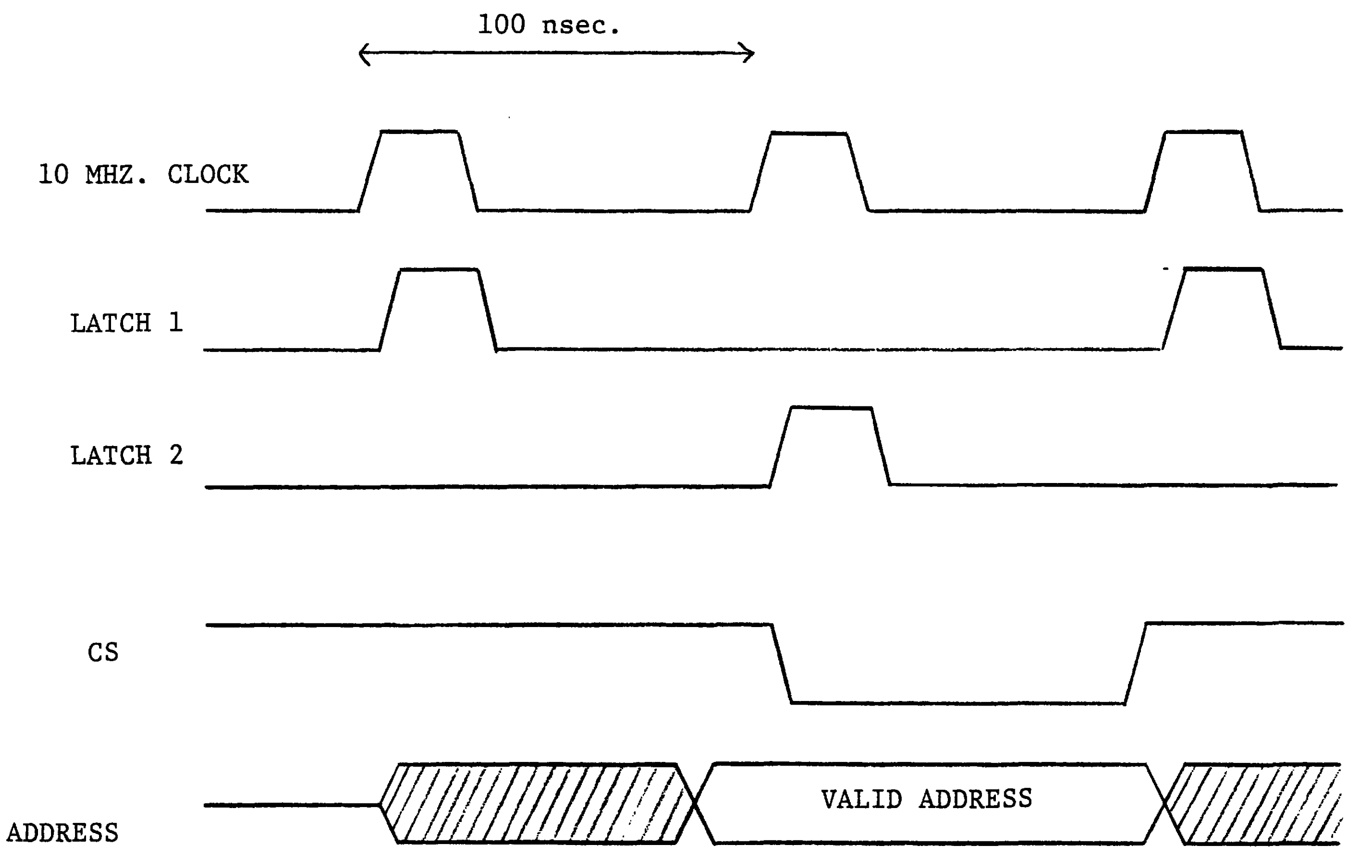
Image Video Interface Block Diagram.

Fig. 5



Latch and Buffer Technique.

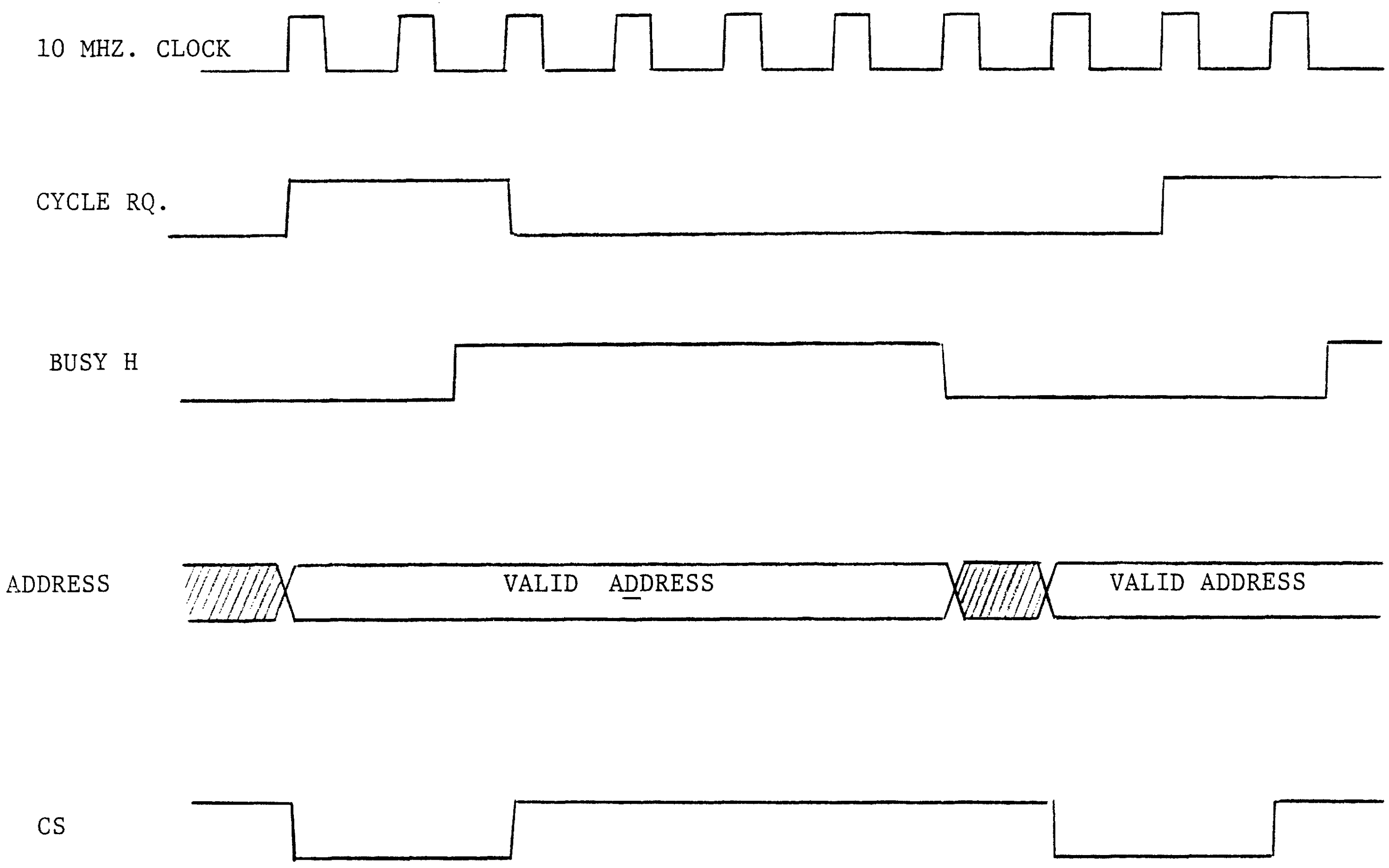
Fig. 6



LATCH-WRITE SIGNALS

TIMING DIAGRAM

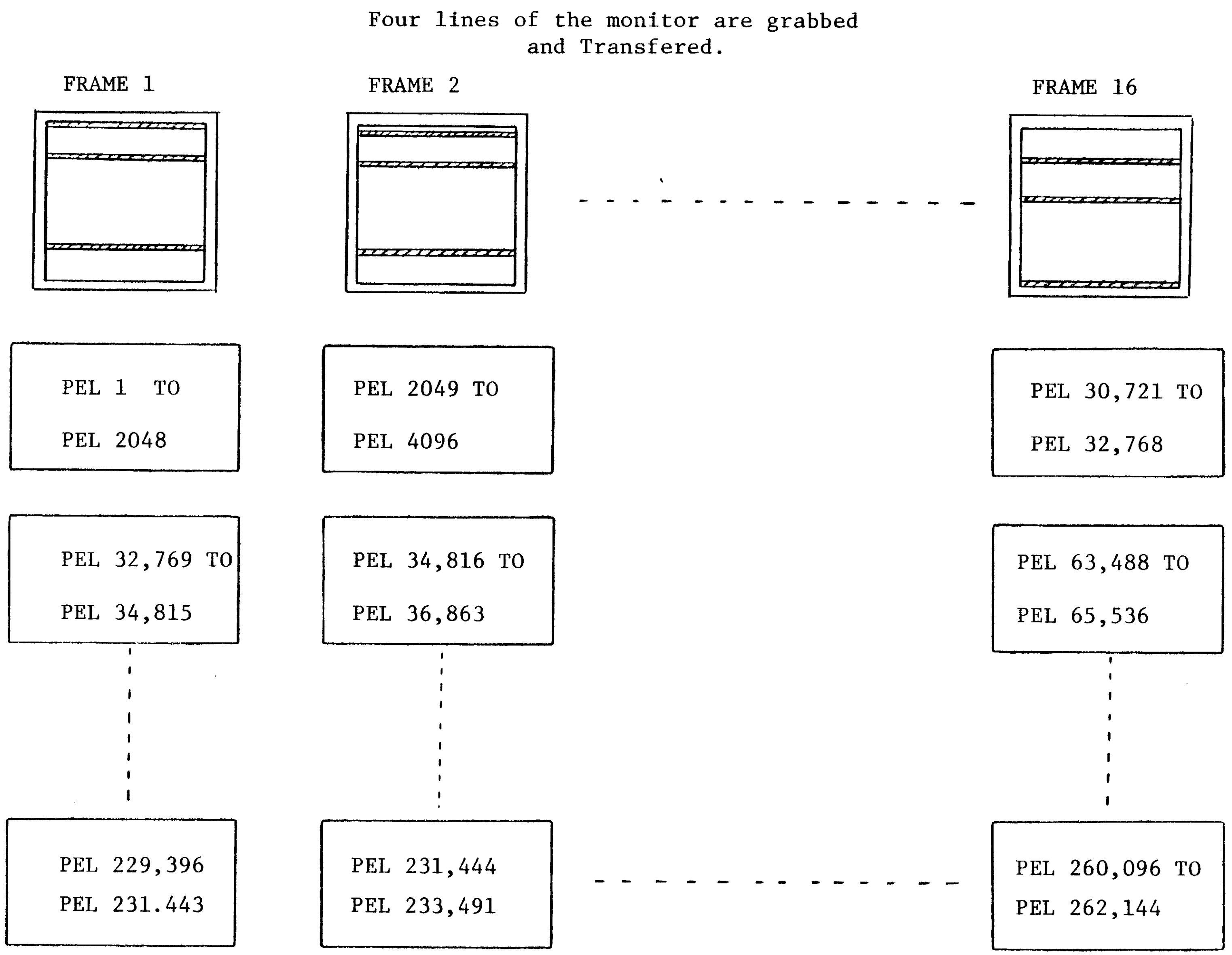
Fig. 7a.



READ-TRANSFER SIGNALS.

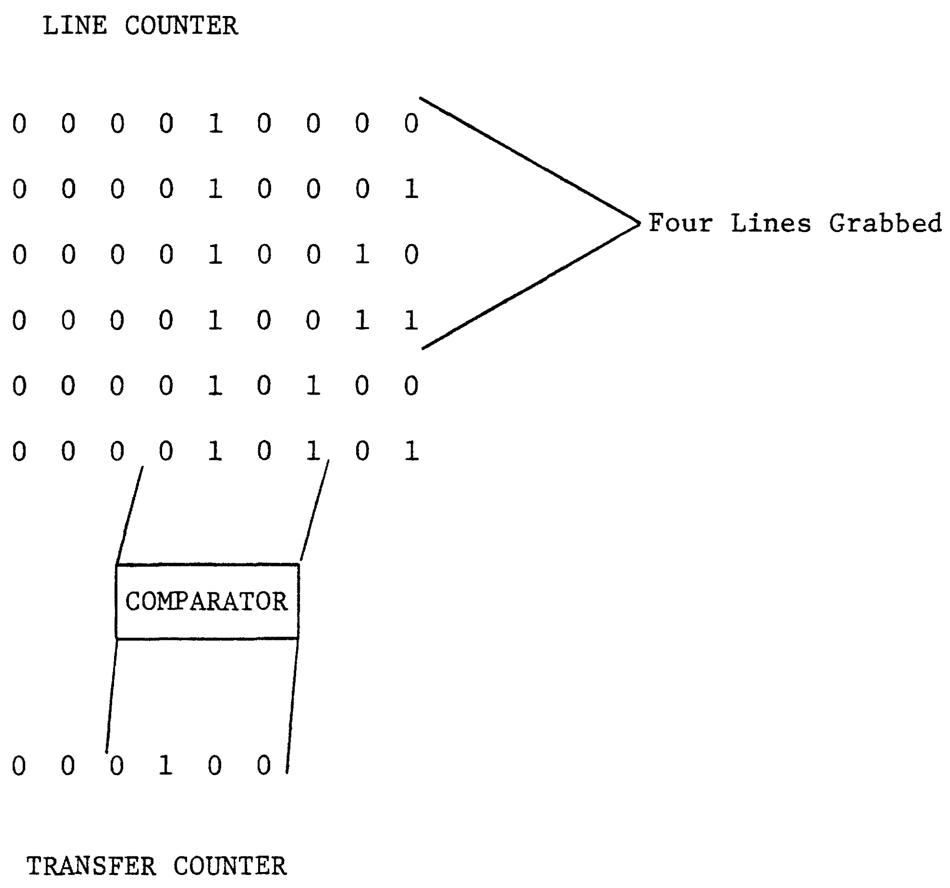
TIMING DIAGRAM

Fig. 7b.



Lines-Pels Grabbed Diagram

Fig. 8



Example Of How Four Lines Are Grabbed

Fig. 9

Test Procedures

The hardware can be separated in two basic functions. One is the control section and the other is the data path. The data path is constituted by the data selectors (LS153), the latches (LS395), the memory (2148) and the output drivers (LS367). As shown in Appendix C circuit diagrams.

If the control hardware is working properly, the data path is easy to debug. Grounding all inputs *, we are able to detect a bad bit by running the main program, changing only the "for" instruction making only one pass through the transfer task, and printing the results of the sequence of pels transferred by the DR11-B. This procedure is implemented in the software as a comment not interfering with the normal operation.

The complexity of the board resides in the control section. The first action that the program does is select the Video Image Interface by comparing the address bits to a set of switches in the board. This comparison is done by the 8136s, in Fig. 10.

When an address is matched, a positive going pulse is generated and sends back a SSYN pulse to the CPU to inform that this interface received an instruction and is ready to

* Make sure to disconnect all input signals. A short can damage other boards.

begin an operation. At the same time when the address is matched, the data bus loads the Input Status Register to select which of the components (L,C1,C2) is to be "grabbed", and the resolution of the image. This process is done by the bus trans-receivers 8T38s and a four bit latch (LS75) to store the values throughout the whole operation. The first two bits of the Input Status Register are used to select the components:

SW2	SW1	
0	0	No selection.
0	1	C1
1	0	C2
1	1	Luminance

The third bit is the resolution desired; asserted (high) is half resolution. The fourth bit is used to initialize all the registers.

Setting the switches has a pecularity. Due to the physical proximity of the switches in the board, careful attention has to be taken to assure a correct setting. Two 10 switch banks are placed at the middle top section of the Video Image Interface board. The first two bottom switches perform the selection of the number of lines that takes the DR11-B to make a 2 K. block data transfer. These two switches were previously explained. The other 18 switches perform the address selection. Fig. 11 shows the two switch banks and an example of the selection using the base address

764240.

When the selection has been made, the software sends a DMA operation command and the DR11-B propagates a GO pulse to the Video Image Interface. This signal is latched by the GO register (LS74) and remains set throughout the picture transfer.

Good timing signals are essential in this board. Two synchronizations are needed, one with the CIM and the other with the DR11-B. Timing is complicated due to the fact that the signals are not always present. These signals depend on many conditions, but principally are related to the horizontal blanking (HR), vertical blanking (VR), and, most important, the transfer register signal WE. If WE is high, a complete grab cycle has been performed and it is ready to make transfers. Naturally the signals stop when this is high. The DR11-B provides the timing signals for the transfers, so the signals generated by the video image interface are hard to debug. If a problem exists with these signals, disconnect the two connectors from the DR11-B and ground pins 2 and 3 of B4, forcing a clear state to the transfer control register providing indefinite time signals. This makes it easy to watch and debug. There are periodic signal gaps due to the VR and HR signals.

The 5 Megahertz clock should be always present. This signal does not depend on any other and is obtained by dividing the 10 Mhz. input signal with a flip-flop (LS74).

The image resolution bit gates the proper clock. Full resolution corresponds to false (low), and gates the 10 Mhz. clock. If true (high), the 5 Mhz. clock is selected.

CK signal is the resulting clock signal and is divided by two (CK/2) that goes to the memory address counters (LS161). These chips count up once every two pels "grabbed". The CS signal enables the memory to write or read. When the system is "grabbing" an image, the memory is writing and the CS depends on CK. When reading, transfers are made to the DMA board; this signals depend on the DR11-B using the BUSY signals to enable the memory.

If an overrun occurs, the ATTN signal is set by the Overrun register and no other pel is "grabbed" until the DR11-B finishes the transfer in progress. No clock signals are generated until the next frame is synchronized and the lines which caused the overrun are about to be "grabbed". This ATTN signal is cleared by a comparator (7585) which compares the number of block lines already "grabbed" in a frame with the number of transfers in the next frame needed to get to the required line where the overrun occurred last frame. The number of block lines that have been "grabbed" are counted in the Transfer Counter, as well as the number of transfers to get in place by the Sync. Counter. When a match exists, the ATTN signal is cleared and the lines are "grabbed".

The Memory Address Counter (LS161s), is a 10 bit counter to address the full capacity of the 2148 memories. The circuit is connected as a look ahead carry configuration giving the speed required. When the maximum count is attained, a pulse is sent to the transfer counter, the transfer register and the counters are loaded with 0s except the first bit that is loaded with a 1. The pulse that generates the loading gives the 0 count for the entire counter.

The transfer register is clocked by the previous pulse and makes WE high. A REQUEST signal is sent to the DR11-B by the Cycle Request Register which has a configuration to assure adequate settling time for all the signals after each transfer to the DR11-B. The absence of the REQUEST signal is easy to detect, the processor hangs up waiting for it, and no answer is given back to the DR11-B by the Video Image Interface.

The Select register gives the required clock signals to latch the LS395s. Two LS3395s latch a pel, and the next is latched by the other two, giving a 16 bit word containing two pels.

The sequence of lines "grabbed" is controlled by the line counter and the frame counter. The TVAM can be tested using vertical bar images that essentially contain the same information on each line, avoiding line considerations. If no problems occur when storing vertical bar images, then the loading of the special format can be done. To fully check out the image stored, it should be in the correct order.

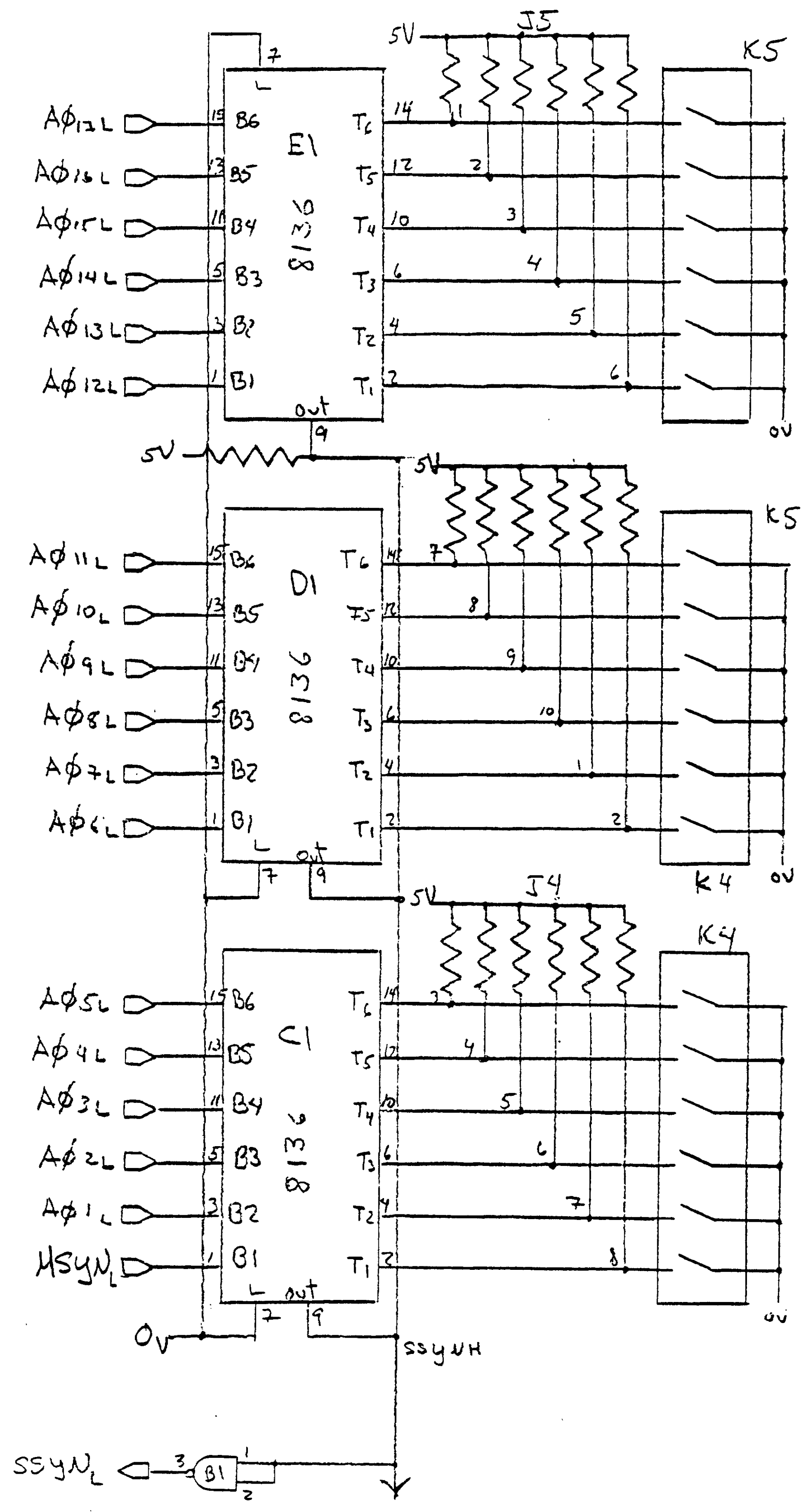
The line counter never stops, using the horizontal blanking signal ORed with the vertical blanking signal

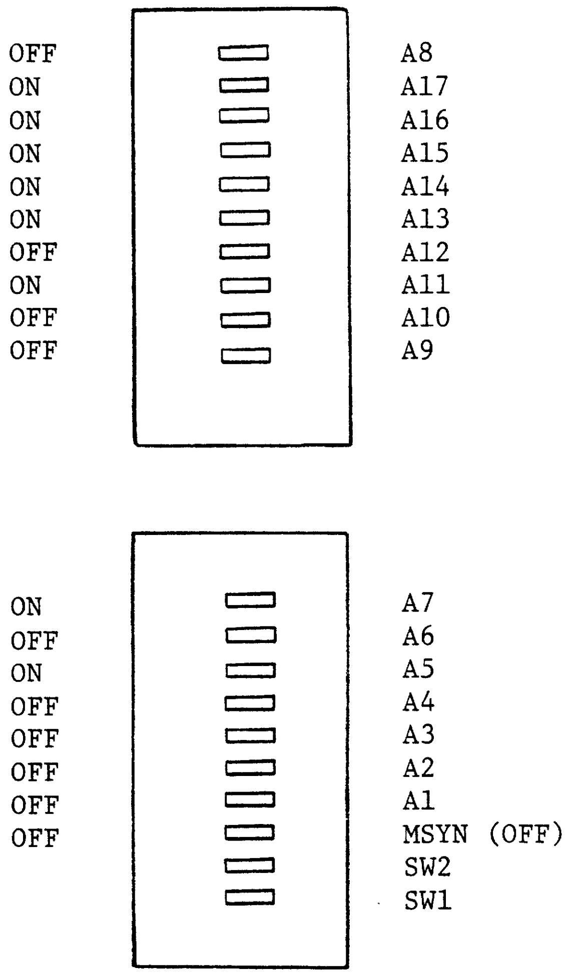
(HVBH) to clock it. The six more significant bits are compared by two 7485 to the frame counter, determining which are the lines to be "grabbed". The resulting signal CE, enables the memory address counters to address the memory. If the DR11-B is still in a transfer cycle, and one CE signal is generated, an overrun occurs and the sequence described above is performed.

The counts of the frame counter and the transfer counter depend on the programmed number of lines needed for the DR11-B to make a block transfer. A table of the possibilities is shown in Fig. 12. The frame counter only counts frames that have been successfully "grabbed", changing the set of four lines to be chosen on the next frame.

It is obvious that before trouble shooting the Video Image Interface, the DR11-B has to be tested. This task is done by connecting the input connector to the output connector (P1 to P2) of the DR11-B, looping back all the signals, and using the DRBTST.TCI program, ran with the EXEC command. If the DR11-B returns the values entered by the operator, the DMA board is in operating conditions.

Fig. 10 · Board Selector Circuit





Address 764240

SWITCH BANKS.

Fig. 11

SW1	SW2	L. C.	F.C.	T.C.
0	0	256	64	2
1	0	128	32	4
0	1	64	16	8
1	1	32	8	16

LINE, FRAME AND TRANSFER COUNTERS

POSSIBILITIES.

Fig. 12

DIRECT MEMORY ACCESS.

The DR11-B is a general purpose, UNIBUS, direct memory access device. The DR11-B is designed for installation in a single quad small peripheral controller slot, and is connected to the video image interface.

The DR11-B uses four programmable registers. The Data Register (DR), is divided into Input and Output Registers (IDR and ODR), being at the same physical address. Reading or writing to the Control Status Register also divides this register in two CSR and Error and Information Register EIR which are grouped under a common main heading.

Prior to a data transfer, the Word Counter Register (WCR) is loaded with the two's complement of the total number of words to be transferred. During subsequent transfers, the WCR is incremented by one for each word transferred. Upon transfer of the last word, the WCR overflows and causes the READY signal to set, an action that tells the Video Image Interface that its transfer is complete.

The Bus Address Register (BAR), like the WCR, is word-addressable only, and can be read or written by the CPU. The BAR is normally incremented by two after an NPR data transfer, so that succeeding transfers are made to consecutive words; i.e., the bus address is advanced by two byte-address increments after each transfer.

The input and output data registers share the same address. During writes to the CPU memory, the IDR buffers data received from the Video Image Interface. In the programmed I/O mode, the program can obtain data by reading the IDR. When the CPU writes to the Data Register address, the ODR is loaded, and in this application the ODR is never used (no data is transferred from the UNIBUS to the Video Image Interface).

The CSR and EIR also share the same address. Writing to this address always writes to the CSR; the EIR is a read-only register. Reading the EIR tell us the status of the DR11-B with respect to any particular operation. Appendix tables 1-1 and 1-2 describes the bit functions in the CSR and EIR.

In response to software commands, the DR11-B is capable of crossing 32K boundaries to transfer maximum blocks of 64K 16 bit words. For this application 4 K blocks are transferred.

The DR11-B can be operated in either a programmed I/O or DMA mode. In programmed I/O, data is moved to or from the picture grabber (video interface) under CPU program control. When operated in DMA mode, the DR11-B becomes master via a Non Processor Request (NPR), and operates directly on the memory to satisfy requests originated at the video interface.

The DR11-B has three operating modes:

1.- Programmed data transfers, in which the CPU reads or writes to the DR11-B registers.

2.- DR11-B interrupt of the CPU via conventional bus request/bus grant sequence.

3.- DMA data transfers to or from the memory with the DR11-B functioning as bus master after a nonprocessor request and bus grant sequence.

Programmed Data Transfers.

Programmed data transfers are basically program controlled. In this operating mode, the DR11-B functions as a slave to the CPU. This mode of operation is the same as using a DR11-K controller, which is a general purpose programmable controller. Although this interface is less expensive than the DR11-B, it was not selected for this project because it can not handle our high speed transfers requirements.

Interrupt Operation.

The interrupt level is used-selected. Four priority-select levels are available.

At the end of a transfer, the word count register overflows and generates a flag which causes an interrupt. The outputs of the six flip-flops used for indicating errors are ORed together, then ANDed with the output of the interrupt enable flip-flop to initiate an interrupt request.

When an error is detected, the interrupt flip-flop causes an interrupt request to be generated. If the CPU is operating at priority level 4 or less, this signal is asserted on completion of the current instruction. Then if no higher-priority device is requesting service, the signal BUS GRANT propagates to the DR11-B.

DMA Operation.

The DR11-B becomes bus master (by generating a NPR) to effect the transfer of data between the video image interface and the UNIBUS. The video image interface controls the type of data transfer by suitably coding control signals.

There are three DMA operating modes: block mode, 2-cycle burst mode, and N-cycle burst mode. The most suitable for this application is block mode.

In block mode. the DR11-B must obtain and then release the bus for each word transferred, which enables other devices on the bus to interleave their data transfers with those of the DR11-B. Transfers normally continue at a constant rate until the specified number of words have been transferred.

During DMA transfers, the DR11-B control logic sends control signals to the video image interface. The video interface responds with sixteen bits of data. This data reaches the DR11-B data multiplexer via the input control lines to the output port and I/O receivers. The video image interface also has to generate several signals that are applied to the DR11-B control multiplexer.

This signals are:

1.- CO CNTL H.

2.- C1 CNTL H.

3.- CYCLE RQ A.

4.- WC INC ENB H.

5.- BA INC ENB H, and

6.- BURST RQ L.

The video image interface generated data is transmitted to the UNIBUS through the input data register.

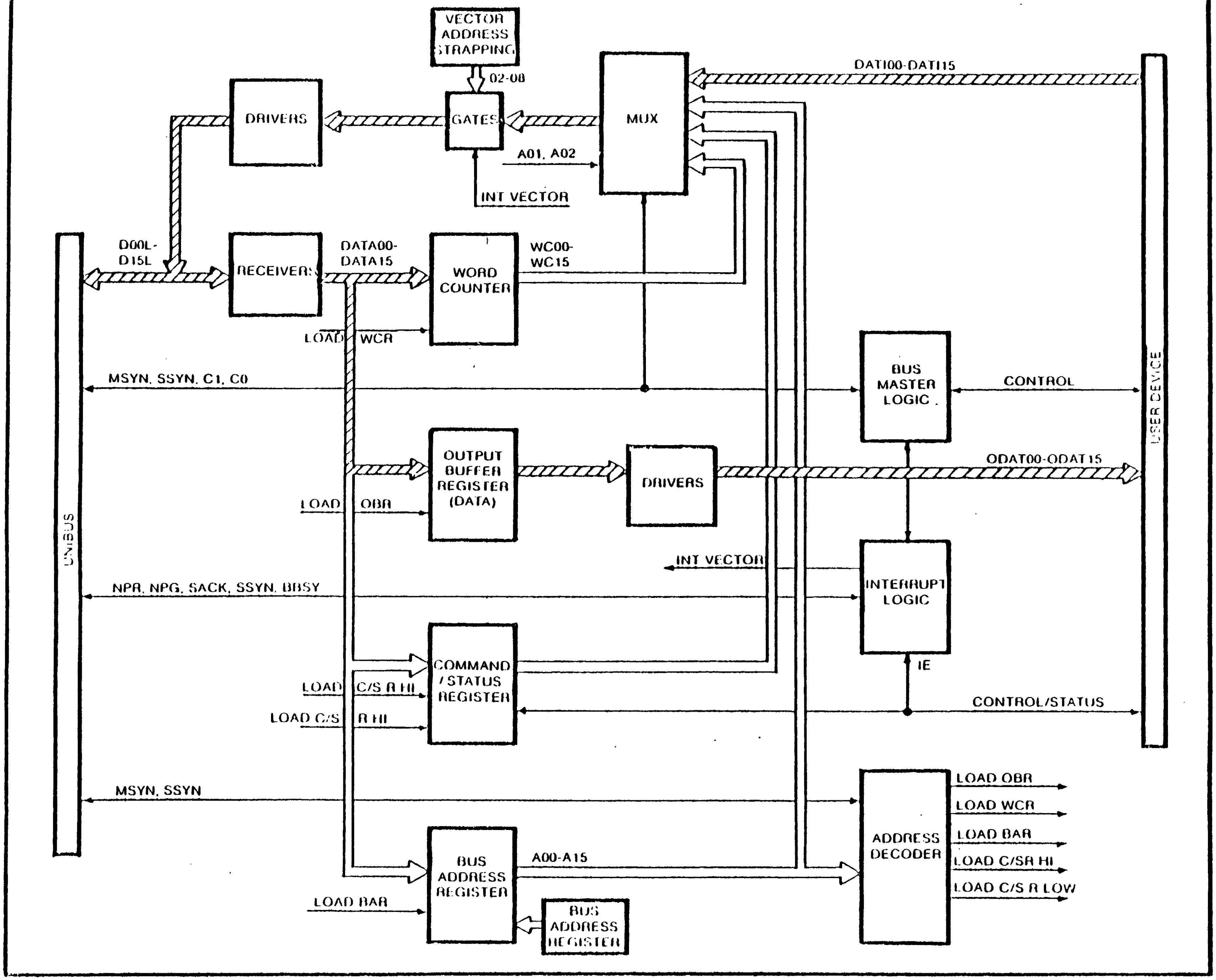
To initiate a DMA operation the video interface checks BUSY L to determine whether or not the DR11-B is in a busy cycle. If BUSY L is not asserted, the video image interface can initiate a transfer.

A block diagram of the DR11-B is shown in Fig. 13.

DMA operation begins when the DR11-B sends GO and READY to the Video Image Interface. These signals are generated in the control logic, within which the READY flip-flop is cleared by a GO pulse if no ERROR condition exists. The busy flip-flop in the control logic is cleared by the CPU initialization (BUS INIT) or by completion of the previous NPR operation.

To initiate a DMA operation, the Video Interface checks BUSY and send a CYCLE REQUEST, when the BUSY is cleared again generates an other CR until all data is transmitted. The Video Interface can not generate a second CYCLE RQ when BUSY is asserted to guarantee that no ERROR is set by a MULTICYCLE RQ.

The number of total block transfers depend on the selection of the Line Counter switches in the Video Image Interface. The transfers are controlled completely by software, so the number of required transfers has to match with the number of transfers in the program.



MDB-DR11B, Logic Organization

Fig. 13

SUGGESTIONS FOR FURTHER WORK

The Television Acquisition Module described here is intended to form part of the first Color System prototype. The experiences in designing model circuit boards will result in a highly refined system including all the improvements resulting from the use and development of this prototype. Making these original ideas become a reality, helps us to develop more and better characteristics for future generations of color printing systems.

Loading the video memory with image lines in the order required to get a completely sequential image is indeed a requirement for the present system. The image loaded in this form looks like "garbage" instead of the proper picture already processed. This problem can be solved by making the buffer larger (32 K), so only one transfer is made per frame, thereby eliminating the need to preload in a special order. The memory chips required for this purpose were not used in the prototyping stage because their pin configuration is not well suited to wirewrap prototyping on DEC standard boards. If the hardware were rebuilt and printed circuit used for this purpose, the amount of buffer recommended can be incorporated.

Another further recommendation will be the use of a software controllable Line Counter instead of the independent Line Counter of the prototype. More applications to the TVAM could be made if the grabbing sequence could be programmable. Not only would it be possible to store the image on the disk but one could re-load the data interleaving tasks by any other device, making the system more versatile.

Not many hardware changes are required to achieve this capability. A one word register would be enough to control the count loading, since the counters are 74LS161 and can load an address in parallel, making changes fairly easy. Some software changes certainly will be required.

BIBLIOGRAPHY

GILL, A. Machine and Assembly Language Programming of the PDP-11, J. Wiley, New York, 1979.

LEE, E., A Color Translation System , S.M. Thesis, MIT, October 1980.

MACEWEN, G., Introduction to Computer System Using The PDP-11 and Pascal , McCgraw-Hill, 1980.

PEYNADO-SANCHEZ, E., A Digital Proccessor for Color Images , S.M. Thesis, MIT, February 1981.

SCHREIBER, W.F., Color Processing For The Helio-Klischograph Cognitive Information Processing Group.
Internal Memorandum, MIT, July 1978.

SCHREIBER, W.F., Four Color Printing
Internal Memorandum, PROV-108, MIT, February 1982.

TROXEL, D.E., An Interactive Image Processing System , IEEE Transactions, Vol. PAMI-3, No. 1, January 1981.

SHAPHIRO, S., A Digital Color Correction Module , S.B. Thesis, MIT, January 1980.

VACHON, G., Selective Luminance Alteration and Correction, S.M. Thesis, MIT, May 1981.

APPENDIX A

SOFTWARE

The following C programs were used to test and debug the hardware. I am including also the macro programs used to set and run the tasks on IPS, the more important "include" tables which the main program refers to, and finally a C program to test the DR11-B.

```
                                        /*
This is the main program to grab a picture using the
TVAM.
*/

#define VERSION         1
#define REVISION        0
#define CSRINIT         GO|FNCT1|READY|CYCLE

#include        "gb.h"          /* define register base */
                                /* for TVAM and CSR bits */
/*
#include <ctype.h>
#include <stdio.h>
#include <signal.h>
*/
#include <ips/pic.h>
#include <ips/core.h>
#include <ips/map.h>
#include <ips/fdb.h>
#include <ips/tcb.h>
#include <ips/drb.h>
#include <ips/ioop.h>

unsigned        csrld = CSRINIT;
unsigned        *regbase = (unsigned*)GBREG;
unsigned        verbose = TRUE;


struct pic      inpic,outpic;


main(argc,argv)
int argc;
char *argv[];
{

        int     args[10];
        int     *buf;
        int     chap = C_UB0;
        int     i;

        printf("Television Acquisition V%u %u0,VERSION,
                                                REVISION);
                printf(" (default all) store =");

                tvamreg();

        outpic.p_pic = argv[1];
        outpic.p_cbsw = 1024;
        outpic.p_mbpl = 256;

        buf=getbuf(1024, C_UB0);
        outpic.p_buf = buf;

        for(i=0; i<64; i++){
        args[0] = C_GET|ZROCOR;
        args[1] = 40;
        args[2] = chap;
        ztrap(M_CORE,args);

        args[0] = MMZ2U3;        /* maps the buffer */
        args[1] = args[2];
        args[2] = RWACES|(2*0400);
        ztrap(M_MAPU,args);
        buf = args[2];

        args[0] = ZIOTRN|ZIOKBF;/* DR11-B transfer begins */
        args[1] = _tcb->t_FDB0;
        args[2] = 0177400;
        args[3] = 0;
        args[4] = buf;
        args[5] = 01000    /* number of word transferred */
        args[6] = ZIORED;
        args[7] = 0;
        args[8] = (WTF<<8)|2;
        args[9] = 0;
        ztrap(M_IOOP,args);
/*      printf("transfers %d %d %d %d %d %d %d %d ......
                                                    %d %d0,
                  buf[0],buf[1],buf[2],buf[3],buf[4],
                  buf[5],buf[6],buf[8],buf[076],buf[077]);
*/
                if(pwrit(&outpic))/*write the buffer to disk*/
                {
                        error("write error %s",NULL);
                }
        }
        pclos(&outpic);     /* close the file */

        args[0] = buf;
        args[1] = 2;
        printf("number of words %o 0,1024/040);
        ztrap(M_FRET,args);

}

tvamreg()       /* sets the Video image Interface CSR */

{
register unsigned       *addr;


        int args[3];
        args[0] = MMZ2U3;
        args[1] = 07600;
        args[2] = RWACES|(0400 * 0200);
        ztrap(M_MAPU, args);
        addr = 04240 + args[2];
        *addr = CONE|CTWO|HALFRES|GOB;/*set status register*/
        printf("reg contens =====> %o %o 0, addr, *addr);
        printf("ya %o0,addr);
}


/* function to get a memory buffer of size words */

getbuf(words,chap)
{
        int args[3];
        args[0] = C_GET|ZROCOR;
        args[1] = words/040;
        args[2] = chap;
        ztrap(M_CORE,args);
        return(args[2]);
}

/* This function frees the buffer */
freebuf(block,words)
{
        int args[3];
        args[0] = C_RET;
        args[1] = words/040;
        args[2] = block;
        ztrap(M_CORE,args);
}

/* error routine */

error(s1, s2)
char *s1, *s2;
{
        printf(s1,s2);
        printf("0);
        exit(1);
}
```

```
                         /*   gb.h
        register definitions for Televition Acquitition Module
        This table must be included by the main program
*/

#define GBREG (struct gb *) 040

struct  gb {
        unsigned wcr;            /* word counter register */
        unsigned bar;            /* bus address register */
        unsigned csr;            /* control status register */
        unsigned idr;            /* input data register */
        unsigned selr;           /* select register */
        };

/* define csr bits  */

#define MAINT   02000         /* allows diagnostic testings */
#define CYCLE   00200         /* initiate one NPR operation */
#define READY   0100          /* set DMA to make new operation */
#define IE      040               /* interrupt enable */
#define FNCT2   04
#define FNCT1   02
#define GO      01                /* begin transfer */

#define BSIZE   04000

#define BKSIZE  4096
#define FALSE   0
#define TRUE    1
#define BARA    2
#define CSRA    3
#define IDRA    4
#define SELRA   5

/* define selr bits */

#define CTWO    02            /* CONE|CTWO selects Luminance */
#define CONE    01            /* CONE -> C1 */
                              /* CTWO -> C2 */
#define HALFRES 04            /* select image resolution */
#define GOB     010           /* initiates grab of image */
```

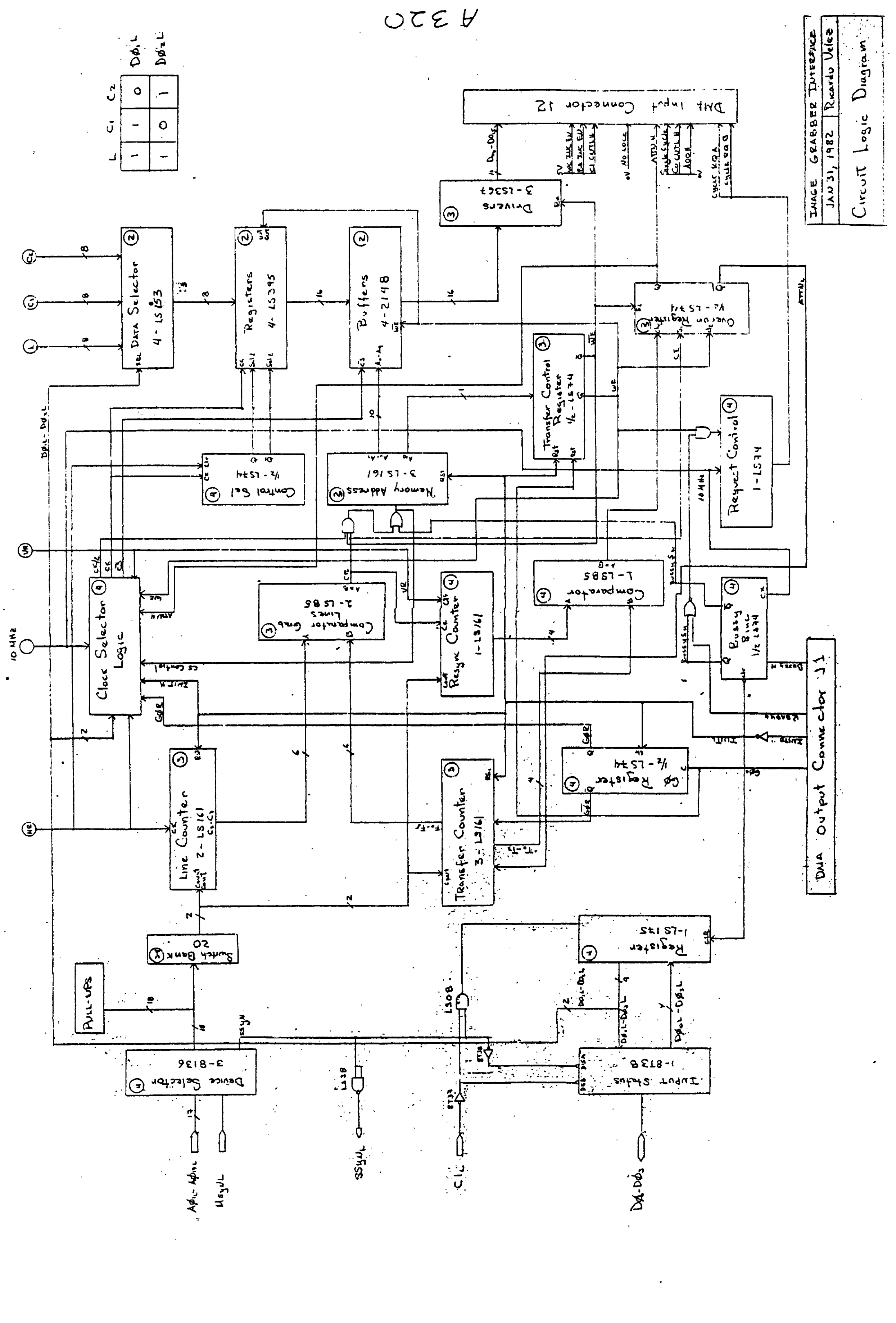
```
                                        /*
Pic.h include file
*/


struct pic
{
        char*   p_pdb;
        unsigned p linc;
        int     p_mbpl;
        int     p_pic;
        int     p_crem;
        char*   p_mbuf;
        int     p_cbsd;
        int     p_cbsw;
        int     p_ebod;
        int     p_dir;
        int     p_wds;
        int     p_buf;
        int     p_cblk;
        int     p_cuni;
        int     p_nwt;
        int     p_extn;
        int     p_mode;
        int     p_stat;
};
```

```
        /* drb.h -- defines for IPS dr11b driver */

struct spec
{
        unsigned func;          /* func code & wtf, see below */
        unsigned dat;
};
#define WTF ((sizeof (struct spec))>>1)

/* function codes: all cleverly contain the "words to follow"
 * in the hi byte, and only use the lo byte as a specific code ;
 * thus, the spec block is the same size for all functions
 */

#define DRGET    (1 | (WTF<<8))
#define DRRDB    (2 | (WTF<<8))
#define DRWDB    (3 | (WTF<<8))
```

```
; PGB           Picture Grabber

        ; Macro routine to set grab.tci program on IPS
        ;
                T.INSTM=T.RTRN+2
                T.OUTSTM=T.RTRN+4
                T.ERRSTM=T.RTRN+6

                .MCALL  .SKD,S.MU, SYSDEF, .TCB, .TCBSYM, TCBE
                .MCALL  .SET,.SCHED, .SETN
                SYSDEF
                S.MU
        BASE=.

                TCBE    TCB.PC,$TS,$TE-$TS+77/100

        TASK:   TCBE    TCB.PG!TCB.NO,TSK+T.PROG,,,GRA,,,,1
                TCBE    TCB.MB!TCB.PD,TSK+T.PAR2,,,<      PICsys>,
                                                1,ZIOXSF,MESO,1
                TCBE    TCB.NO!TCB.DW,TSK+T.NARG,2,,,,,,1
                TCBE    TCB.OP!TCB.FN,TSK+T.FDB0,-1,,,,
                                      ZIOXSD!ZIOPOF,MES1,1
                TCBE    TCB.OP!TCB.SW!TCB.AS,TSK+T.INSTM,2,,
                                      <EXMSTM]>,,,MES2,1,ASC
                TCBE    TCB.OP!TCB.SW!TCB.AS,TSK+T.OUTSTM,2,,
                                      <EXMSTM]>,,,MES3,1,ASC
                TCBE    TCB.OP!TCB.SW!TCB.AS,TST+T.ERRSTM,3,,
                                      <ERRSTM]>,,,MES4,1,ASC
                TCBE    TCB.DN


        MESO:   .ASCII  /OUTPUT PICTURE <(.PIC;sys1)>/<EOD>
        MES1:   .ASCII  /File specifier )/<EOD>
        MES2:   .ASCII  /INPUT STREAM <STREAM NUMBER (EXMSTM)>/<EOD>
        MES3:   .ASCII  /OUTPUT STREAM <STREAM NUMBER (EXMSTM)>/<EOD>
        MES4:   .ASCII  /ERROR OUTPUT STREAM <STREAM NUMBER
                                                (ERRSTM)>/<EOD>

        $TS:    .SKD    SCD
                .TCB    TSK
        $TE:    .WORD   0

                .SETN   SCD,S.NAME,PGB
                .SETN   TSK,T.NAME,GRA
                .SET    SCD,X.L40W,$TE-$TS+77/100
                .SCHED  SCD,<TSK>

                .END
                /* drb driver test program */
#include <ips/drb.h>
#include <ips/ioop.h>
#include <ips/core.h>
#include <ips/map.h>
#include <ips/tcb.h>

main(argc,argv)
char **argv;
{
        int args[10];
        int *p;
        int chap = C_UB0;

top:    args[0] = C_GET|ZROCOR; /* get unibus core */
        args[1] = 2;
        args[2] = chap;
        ztrap(M_CORE,args);
        args[0] = MMZ2U3;       /* map into it */
        args[1] = args[2];
        args[2] = RWACES|(2*0400);
        ztrap(M_MAPU,args);
        p = args[2];
        args[0] = ZIOSPC;       /* load dr11b data buffer */
        args[1] = _tcb->t_FDB0;
        args[2] = DRWDB;
        args[3] = atoi(argv[1]);
        ztrap(M_IOOP,args);
        args[0] = ZIOTRN|ZIOKBF;        /* xfr from dr11b */
        args[1] = _tcb->t_FDB0;
        args[2] = 0177400;
        args[3] = 0;
        args[4] = p;
        args[5] = 0100;
        args[6] = ZIORED;
        args[7] = 0;
        args[8] = (WTF<<8)|2;
        args[9] = 0;    /* no data */
        ztrap(M_IOOP,args);
        printf("drbtst: got %d %d %d %d ... %d0,
                        p[0],p[1],p[2],p[3],p[077]);
        args[0] = p;    /* return core */
        args[1] = 2;
        ztrap(M_FRET,args);
        if (chap == C_UB0)
        {
                chap = C_STM;
                goto top;
        }
}
```

APPENDIX B

A detailed block diagram of the Video Image Interface is shown, including all the modules and DR11-B input and output connections.



A 320
IMAGE GRABBER INTERFACE
JAN 31, 1982
Ricardo Velez
Circuit Logic Diagram
DMA Input Connector J2
DMA Output Connector J1
Drivers 3-LS367
Data Selector 4-LS153
Registers 4-LS395
Buffers 4-2148
Overrun Register 1/2-LS74
Transfer Control Register 1/2-LS74
Request Control 1-LS74
Control Sel 1/2-LS74
Memory Address 3-LS161
Clock Selector Logic
Comparator Grab Lines 2-LS85
Resync Counter 1-LS161
Comparator 1-LS85
Bussy Bit 1/2 LS74
Gφ Register 1/2-LS74
Line Counter 2-LS161
Transfer Counter 3-LS161
Register 1-LS175
Switch Bank 20
PULL-UPS
Device Selector 3-8136
Input Status 1-8T38
10 MHz
LS08
LS38
8T38
8T37
Aφ1L-Aφ16L
MsyL
SSyL
CIL
Dφφ-Dφ3
L C1 C2 Dφ1L Dφ2L
1 1 0
1 0 1

APPENDIX C

VIDEO IMAGE INTERFACE CIRCUIT DIAGRAMS

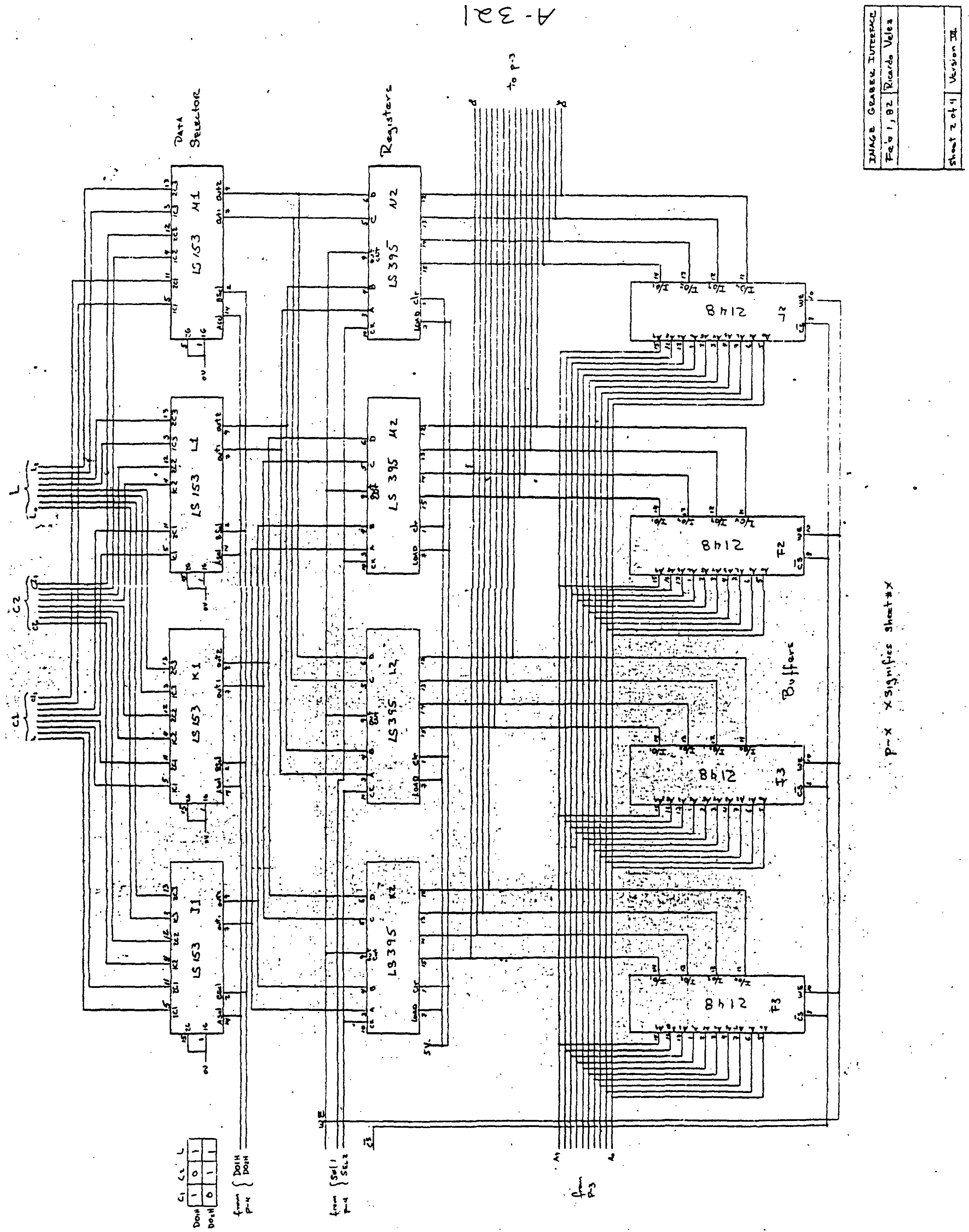
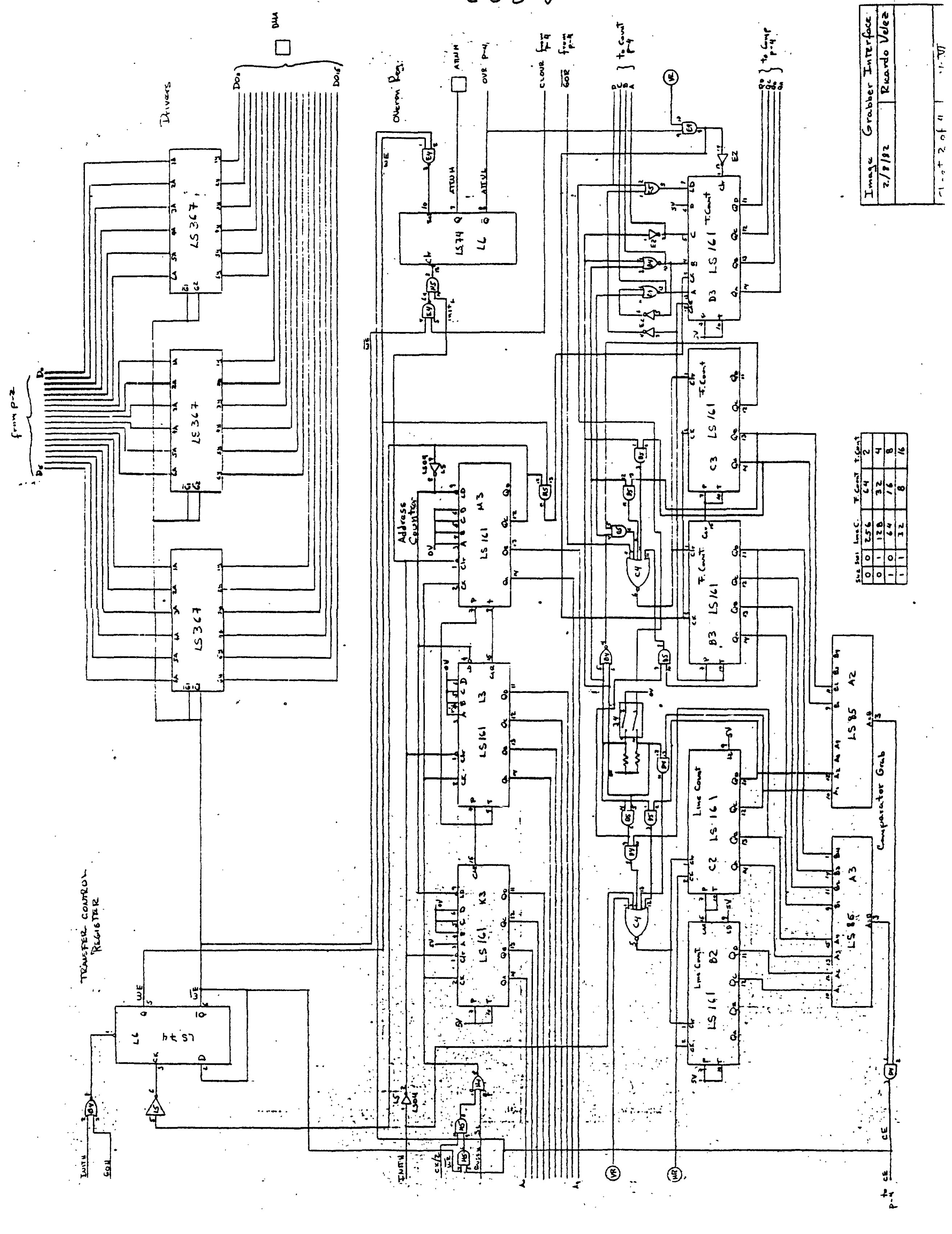
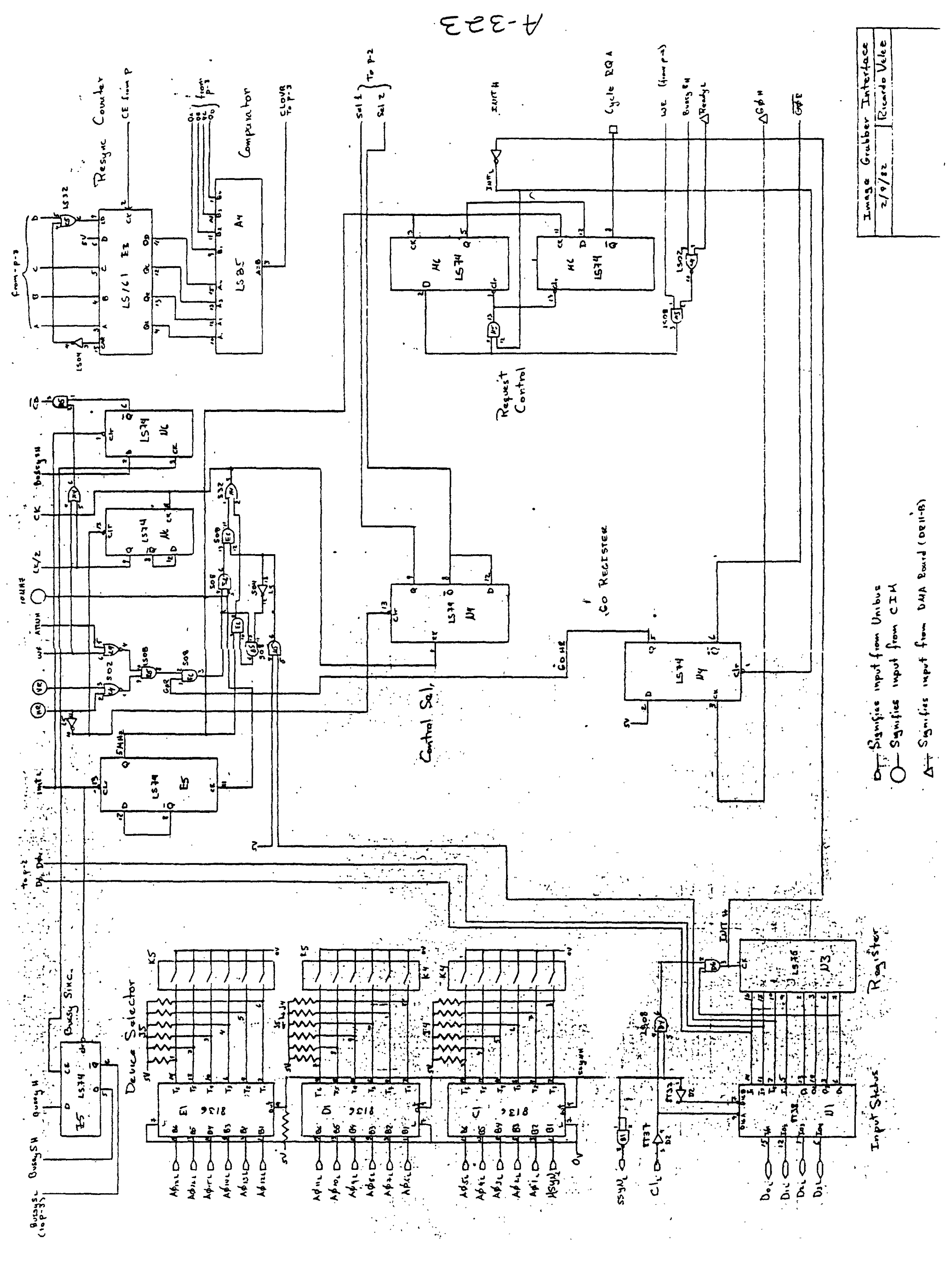


IMAGE GRABBER INTERFACE
Feb 1, 82
Ricardo Velez
Sheet 2 of 4
Version III
A-321
Data Selector
Registers
Buffers
to P-3
from P-3
from P-4 Sel 1 Sel 2
from P-4 DO1H DO2H
LS 153 M1
LS 153 L1
LS 153 K1
LS 153 J1
LS 395 N2
LS 395 M2
LS 395 L2
LS 395 K2
2148 J2
2148 F2
2148 F3
L
C1
C2
WE
CS
5V
0V
P-X X signifies sheet # X

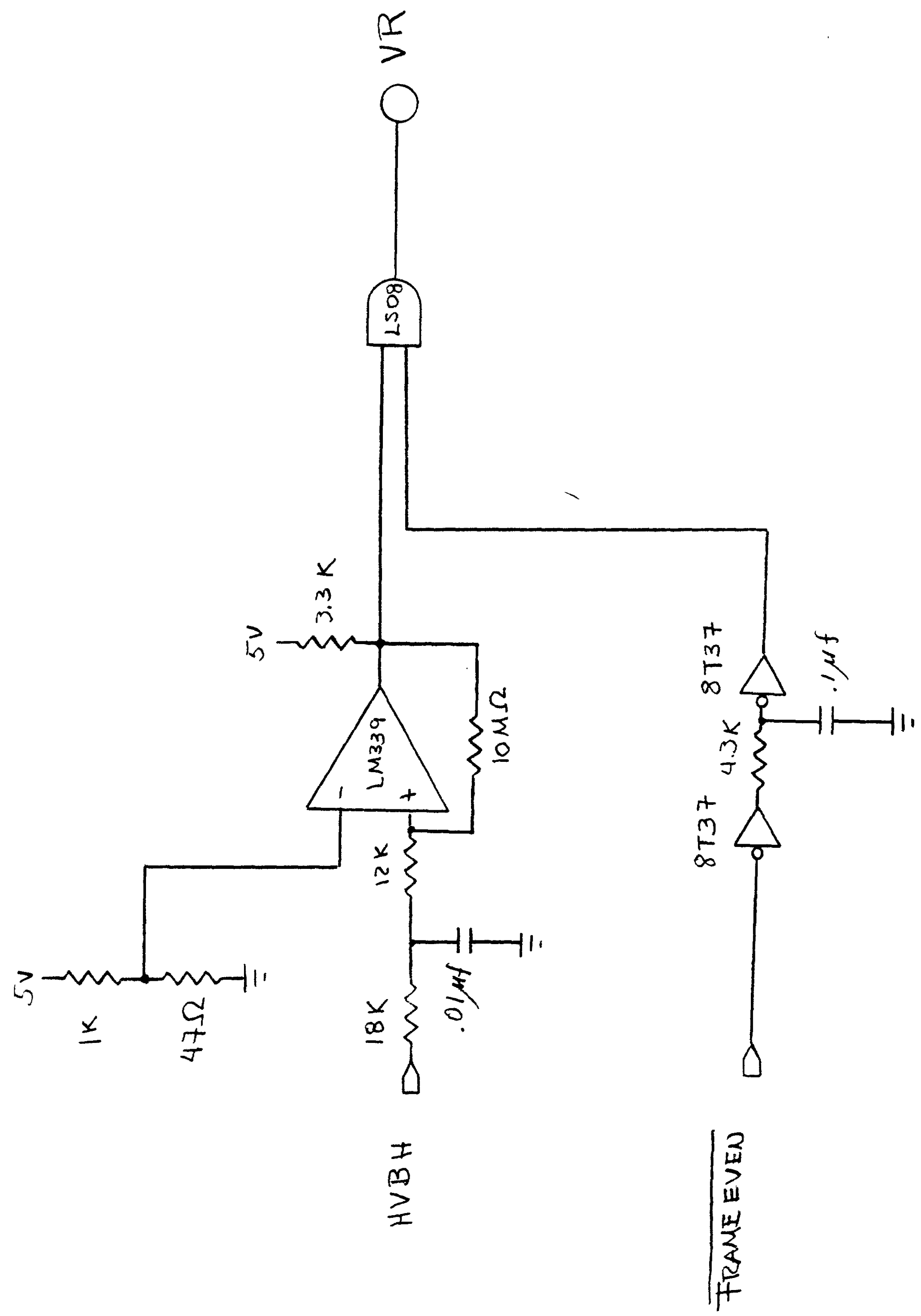


Sw2	Sw1	Line C.	F. Count	T. Count
0	0	256	64	2
0	1	128	32	4
1	0	64	16	8
1	1	32	8	16


A-323
Image Grabber Interface
2/9/82
Ricardo Velez
Resync Counter
Comparator
Request Control
Control Sel.
Go Register
Busy Sinc.
Device Selector
Input Status
Register
LS161
LS85
LS74
LS32
LS02
LS08
LS04
8136
LS175
8T38
Signifies input from Unibus
Signifies input from CIH
Signifies input from DMA Board (DR11-B)
Cycle RQ A
Sel 1
Sel 2
To P-2
from P-3
CLOVR To P-3
INIT H
WE
Busy SH
Ready L
GO H
G/E
CK
CK/2
10 MHZ
ATTN
Busy SL
Busy H
SSYN L
C1 L
D0 L
D1 L
D2 L
D3 L

APPENDIX D

Due to the fact that VR signal was no provided by the CIM, the following decoding circuit was used.



Vertical Retrace Signal Decoding Circuit

APPENDIX TABLES

DR11-B CONTROL STATUS REGISTER AND ERROR AND INFORMATION REGISTER BIT DESCRIPTION

Table 2. Command/Status Bits

Bit	Name	Description and Effect
15	ERROR (read only)	a. Indicates error as follows: 1. NEX (bit 14), or 2. ATTN (bit 13), or 3. Interlock error (module/connector discontinuity), or No Lock signal (P2-28) not grounded, or 4. Bus address overflow (BAOF) as bus address changes from all-"1's" to all-"0's". b. Sets READY (bit 7) and causes interrupt if IE (bit 6) has been set. c. ERROR is cleared by clearing all error conditions, as follows: 1. Module is seated in connector. 2. Bus address is cleared or reloaded. 3. Bit 14 is loaded with a "0". 4. Bit 13 is cleared by the user device.
14	NEX (read)	a. Non-existent Memory. Indicates that the module, acting as bus master, failed to receive a SSYN response within 20 micro-seconds after asserting MSYN. b. NEX sets ERROR bit. c. Cleared by INIT or by loading "0".
13	ATTN (read only)	a. Attention. Shows state of user device ATTN signal. b. Sets ERROR for device-initiated interrupt. c. Set and cleared only by user device.
12	MAINT (read/write)	a. Maintenance. Used to enable execution of diagnostic programs. b. Cleared by INIT.

Table 2. Command/Status Bits (cont'd)

Bit	Name	Description and Effect
11 10 09	DSTATA DSTATB (read only) DSTATC	a. **Device Status.** Indicate state of user-designated DSTATA, DSTATB, and DSTATC signals. b. Set and cleared only by user device.
08	CYCLE (read/write)	a. If set when GO is issued, enables an immediate bus cycle. b. Cleared by INITI, or start of bus cycle.
07	READY (read only)	a. Indicates the MDB-DR11B is able to accept a new command. b. Set by INIT or ERROR, or by word count overflow. c. Cleared by GO. d. If bit 6 is set, READY causes an interrupt, forcing module to release the Unibus.
06	IE (read/write)	a. **Interrupt Enable.** Enables either ERROR or READY to set an interrupt. b. Cleared by INIT.
05 04	XBA17 XBA16 (read/write)	a. **Extended Bus Address.** Along with contents of Bus Address Register, specify address for **indirect** memory transfers. b. Cleared by INIT. c. Bits XBA17 and XBA16 are not incremented when Bus Address Register overflows, but ERROR is set.
03 02 01	FNCT3 FNCT2 (read/write) FNCT1	a. **Function.** Bits available to user device for assignment by user. b. Cleared by INIT.
00	GO (write only)	a. Causes MDB-DR11B to signal user device that a command has been issued. b. Clears READY.