CCD IMAGING WITH A TMS34010 GRAPHICS SYSTEM PROCESSOR

by

CURTIS WAYNE MUELLER, B.S. in E.E.

A THESIS

IN

ELECTRICAL ENGINEERING

Submitted to the Graduate Faculty
of Texas Tech University in
Partial Fulfillment of
the Requirements for
the Degree of

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

Approved

Accepted

May 1990

ACKNOWLEDGEMENTS

I would like to extend thanks to Dr. Larry Masten, and Dr. Sunanda Mitra for serving as members on my thesis committee. For Dr. Parten, I could not possibly express thanks enough for his continued support, encouragement, and dedication to my scholastic career. Were it not for his guidance and persistent belief in my abilities, I would surely not be writing these words today.

Thanks also to my parents for their guidance and assistance, and for providing me with the ability to achieve any of the goals set before me. And to my wife, Candyce, who fulfills my life and provides me with an escape from the occasional doldrums of day to day living.

Finally I would like to express my sincere gratitude to Kevin Kelly and John Van Meter for their assistance in the implementation of the imaging system described herein, and to Steve and Judy Patterson for their contributions to the success of the project.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I
INTRODUCTION

Modern integrated circuit (IC) technology presents system design engineers with many alternatives for implementing complex hardware systems. Complex instruction set computers (CISC), reduced instruction set computers (RISC), and application specific ICs (ASIC) provide designers with specialized hardware to attack a wide variety of electronic system implementations.

RISC microprocessors, while having a limited number of instructions, provide very fast--typically single cycle--execution of the entire instruction set. This is accomplished by hardwiring the instruction set rather than using a microprogrammable core for instruction execution as is done with CISC technology. RISC microprocessors are typically provided with a specialized instruction set for rapid execution of programs designed for a particular application, such as digital signal processing (DSP). A large number of RISC microprocessors targeted at DSP applications are now available, with one example being the Texas Instruments TMS320 family of DSP devices.

CISC microprocessors possess a large instruction set, thus providing system designers with a more versatile general purpose IC for use in virtually unlimited applications. An example of a CISC based system would be the personal computer or PC, based on the Intel 8086 family of microprocessors.

Another example of a CISC processor, but one which is targeted at a specific application, is the TMS34010 graphics system processor (GSP) from Texas Instruments (TI). The GSP provides, on a single IC, virtually all of the hardware necessary to implement a graphic display system. Although program and image memory reside external to the GSP, refresh timing for dynamic memory devices, and timing signals for cathode ray tube (CRT) control are provided by the GSP. The GSP also contains instructions which allow graphic operations to be carried out with minimum hardware and software overhead. The TMS34010 is merely one member in a series of products from TI which support a number of graphic system implementations.

TI's GSP and DSP families are both examples of a trend in conventional IC technology towards a more structured, building block approach to IC design. Electronic systems which, years ago, may have occupied many circuit boards in an electronic chassis are now being implemented on a single integrated circuit. This is made possible through the use of computer-aided design (CAD) tools which allow systems to be built from a number of smaller (primitive) devices or systems. The ideas from an early generation device can be expanded and included in more complex implementations, which include more functionality, and provide higher performance solutions. Such is the methodology behind ASIC or application specific IC technology, where not only semiconductor manufacturers, but system designers as well, can specify and implement complex electronic systems on a single integrated circuit.

This thesis describes the implementation of an image capture and display system based on the TMS34010 GSP. The GSP provides all timing and control signals for reading the image stored in an image sensor, and for displaying the image on a display device. The system is capable of capturing an image from an image sensor (such as a charge coupled device (CCD), a vidicon tube, or any other frame oriented image sensor), storing the acquired image in a frame buffer, and then displaying the image on a display device such as a CRT or liquid crystal display. Additionally, the system is capable of transmitting the image over a serial interface through an on-board universal synchronous-asynchronous receiver transmitter or USART.

What makes the imaging system described herein unique is a considerable reduction in the hardware necessary to implement a complete imaging system. Historically, imaging systems have utilized a substantial amount of discrete hardware to perform tasks such as providing the timing signals required to move the image stored in an image sensor into a frame buffer, and displaying the acquired image on a display device. In addition to this discrete hardware, some sort of a processor is then required to perform image processing operations. An imaging system that utilizes the TMS34010 for image capture and display can provide a dramatic reduction in imaging hardware. This is done by providing the logic to acquire an image, display the image, and perform image processing tasks, on a single IC.

In Chapter II, an overview of the image capture process is provided. An imaging system based on a traditional discrete image capture and display architecture is introduced.

Chapter III presents the TMS34010 graphics system processor. The internal architecture, and the external interfaces of the GSP are discussed. The chapter concludes with a presentation of several systems which utilize the GSP.

Chapter IV is used to introduce the reader to the hardware platform which serves as the basis for the imaging system discussed in Chapter V. The implementation utilizes a TMS34010 software development board available from Texas Instruments. This implementation was chosen because it serves both as a hardware prototype for future GSP-based imaging systems, with applications which might include low cost video-telephone systems or perhaps remote digital surveillance systems, and as a software development system for those same applications. Use of the software development board allows the application software to be run under SDB debugger control, thus providing a more productive environment for implementing GSP based software. This chapter should provide the reader with the necessary background to understand the SDB and its usefulness as a general purpose imaging system.

Chapter V provides the reader with details of the hardware interface between the image sensor and the SDB. Here, the camera interface board--a board which provides the interface circuitry between the SDB and the computer camera--and its associated hardware are described in detail. Schematics and relevant timing diagrams are provided for the camera interface board. Results ob tained with the imaging system are presented in Chapter VI. Finally, Chapter VII provides a conclusion, along suggestions for future work.

CHAPTER II
IMAGING SYSTEMS

An imaging system, as defined for the context of this thesis, is a system capable of capturing, digitizing, storing, processing, and displaying graphic images. A block diagram of one such system is shown in Figure2.1. The system consists of a device for capturing still or moving images, such as a video camera with a CCD image sensor or a vidicon tube. The analog output from the camera is digitized and stored in the video memory. After image capture, the image is displayed on a display device by the display controller. Image processing tasks are handled by the central processing unit (CPU).

A problem with the system shown in Figure 2.1 is due to the large amount of hardware required to implement such a system. Unfortunately, because there are so many different types of input devices, it is typically up to the end user to develop the timing circuitry to produce the clock signals which move an image from the image sensor to the video memory system. Solutions typically require an array of counters which drive state decoders and are implemented with either standard TTL or CMOS logic, with programmable logic devices (PLDs), or with ASICs.[1,2]

In addition to providing timing circuitry for the image sensor, additional devices must be present to provide refresh cycles for the dynamic video RAMs, and to control screen refresh operations for the video display device. The purpose for each of these subsystems will be explained in the last portion of this chapter.

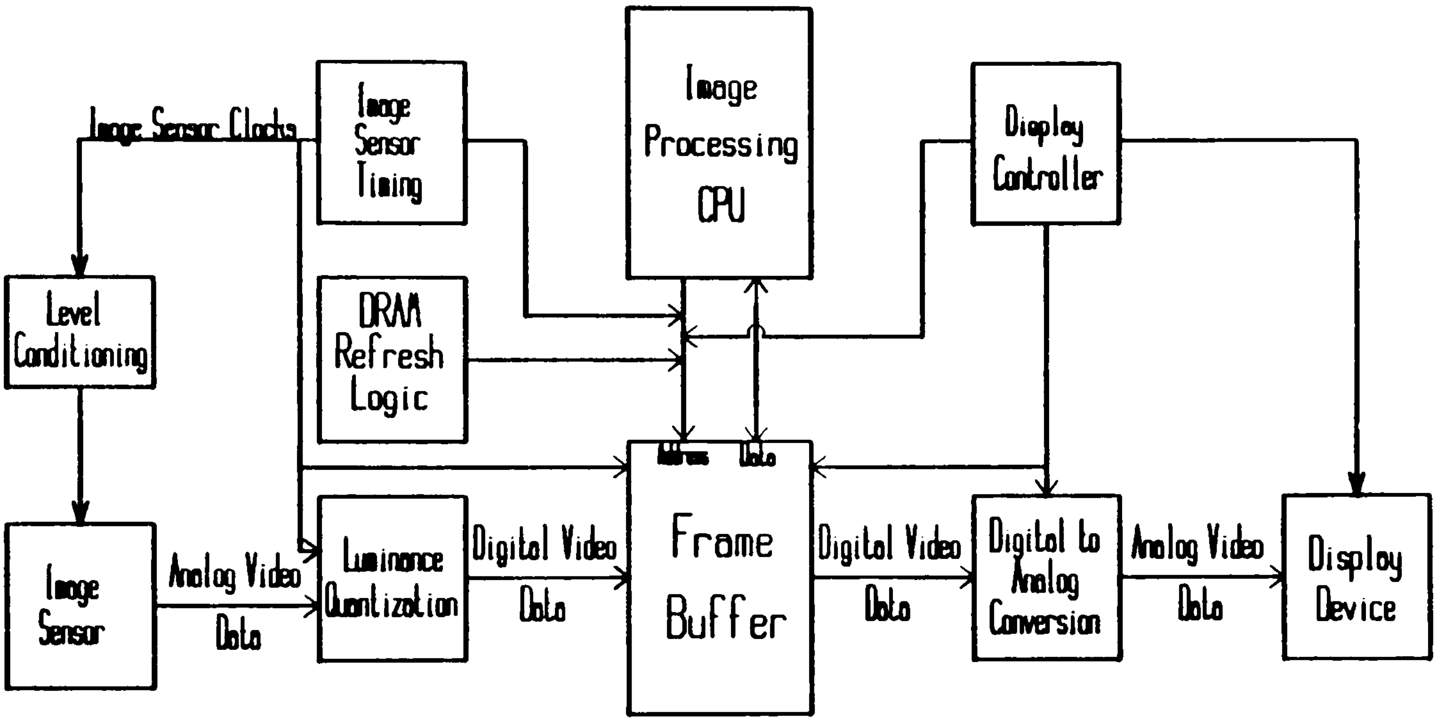


Figure 2.1 Block Diagram for an Imaging System

Temporal, Spatial, and Luminance Quantization

The representation of an image in the digital domain requires the ability to store and process enormous amounts of data. If the image is output from a moving picture device such as a video camera, the image first undergoes temporal sampling. Here the image is captured, a frame at a time, as fast as is necessary to achieve what appears to be continuous motion. Next the image is spatially sampled, resulting in an array of picture elements commonly referred to as pixels or pels. The quality of the acquired image is determined in part by the density of the sampling in both the vertical and horizontal directions. This quality is referred to as the resolution of the image. An imaging system might have a resolution ranging from 128x128 (128 pixels in the vertical direction by 128 pixels in the horizontal direction) to perhaps as high as 2048x2048. Figure 2.2 illustrates the effect of varying the spatial resolution of an image.[3]

Once the image has been spatially sampled, it is necessary to quantize each pixel, thus representing each pixel by some finite number of bits. The number of bits used to represent each pixel is determined by the application for the imaging system. For example, a common imaging system fitting the definition given above would be a facsimile machine which transmits a copy of a black and white image to a remote location. In this case, each pixel is represented by one binary digit; a pixel can be either off or on, black or white. In a more complex imaging system, however, one for which it is not satisfactory to display merely two colors or gray shades, more than one bit per pixel would be required to adequately quantize the image.

For an imaging system used as part of a CAD workstation where animation, or perhaps processing and enhancement of photographic images takes place,

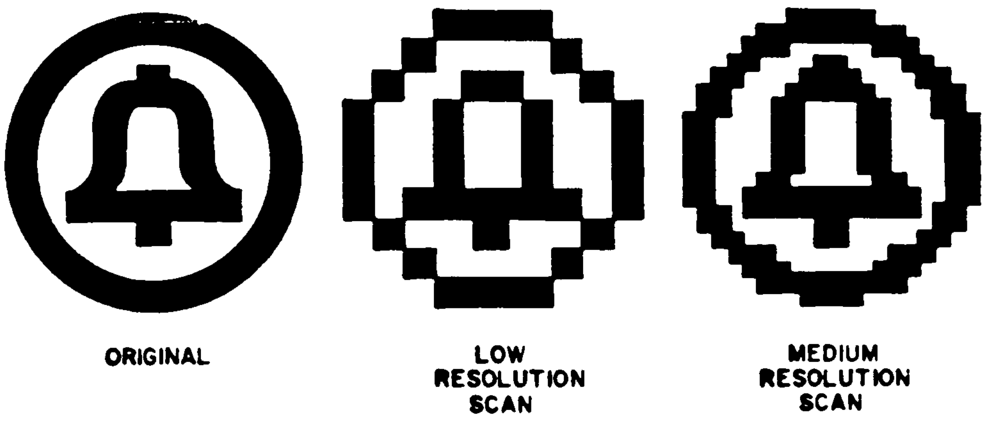


Figure 2.2 Effect of Varying Spatial Resolution

the amount of quantization might be 8 bits per pixel, thus resulting in 2^8 or 256 gray levels. This number would adequately represent an image, since the human eye can only perceive a limited number of changes in the luminance of an image. Research shows that the number of changes perceivable is limited to around 500 over a large image.[4] Figure 2.3 illustrates the process of temporal and spatial sampling of an image followed by the quantization of spatial information.[5]

After the image has been spatially sampled and each sample has been quantized, the resulting digital image must be stored and then processed. The amount of data in a typical image with a spatial resolution of 256x256 pixels quantized at 8 bits/pixel would require 64 Kbytes (256^2) of memory for image storage.

CCD Image Sensor

Although it is possible to use any of a number of image sensors to capture an image, CCD sensors have become quite popular in recent years due to their small size, ruggedness, and dependability. For this reason, the CCD image sensor, rather than some other sensor such as a vidicon tube, will be discussed exclusively in the remainder of this thesis.

A CCD sensor operates by utilizing a lens to focus an image onto the surface of a solid state sensor. Light incident on the sensor causes charge to accumulate in an array of photosensor elements in direct proportion to the intensity of the incident light. The stored charge is then available for conversion to a voltage which is provided at the output terminal of the image sensor. The voltage output by the sensor thus provides spatial luminance information which is directly related to the illumination on the surface of the device.

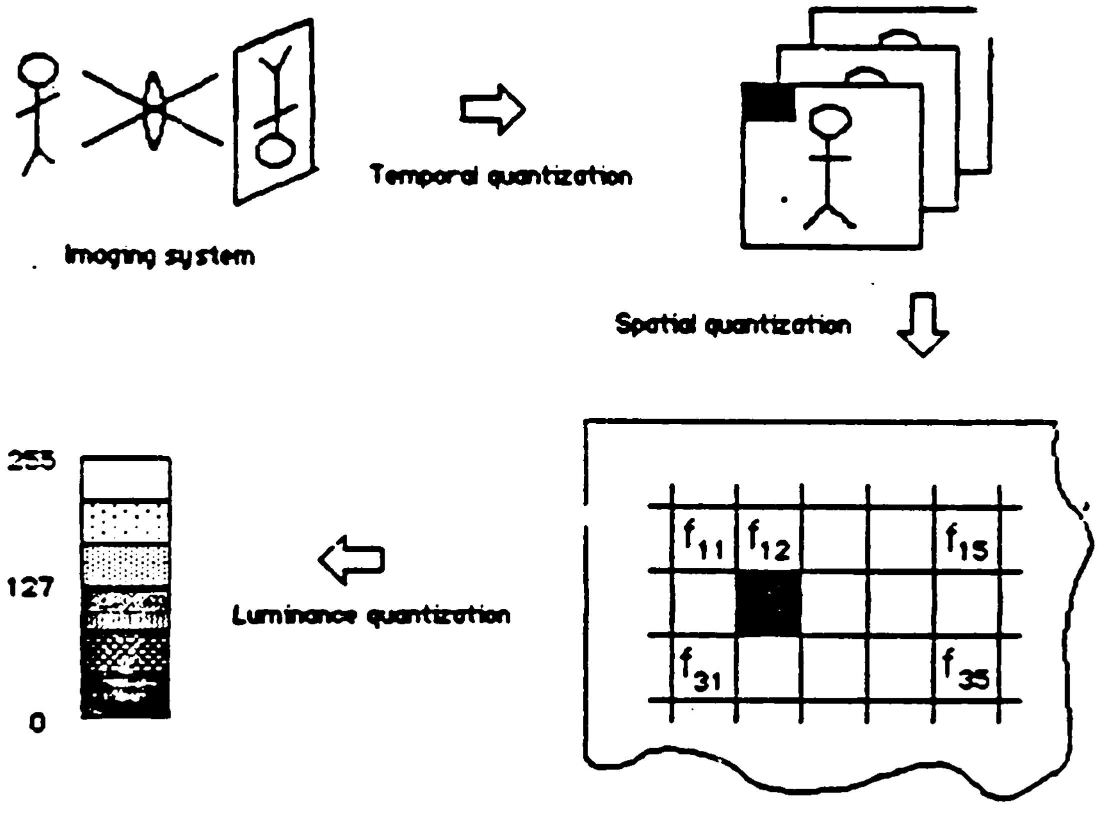


Figure 2.3

An Illustration of Temporal, Spatial, and Luminance Quantization

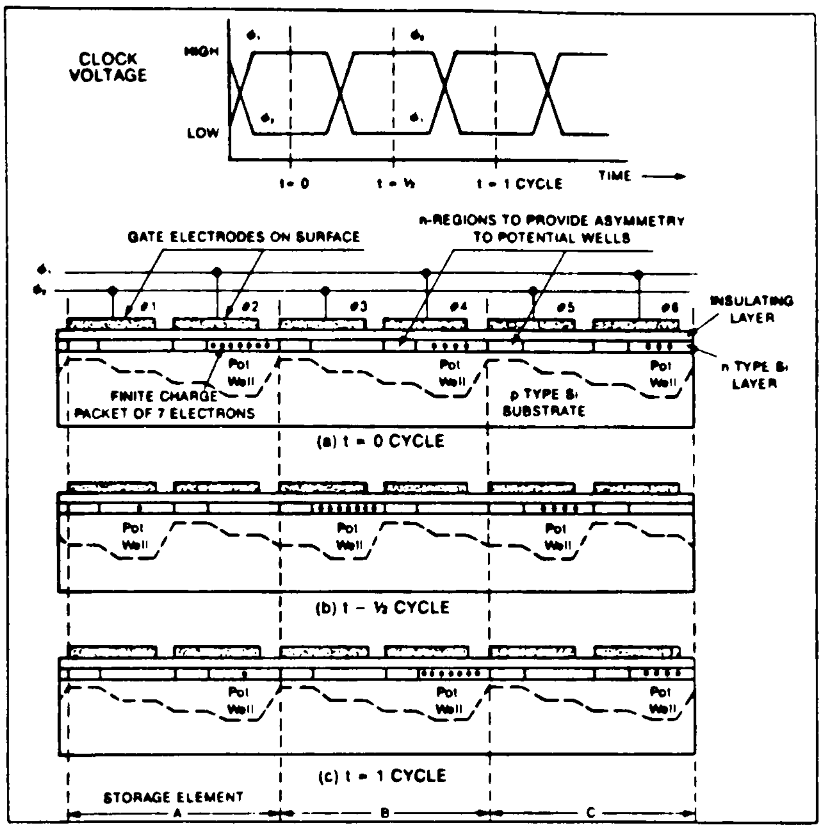
CCD Operation

The principle behind the operation of a CCD is known as charge coupling. The Einstein photoelectric effect dictates that a finite amount of charge is created in a silicon semiconductor material as a result of illuminating the silicon surface. A CCD sensor utilizes this principle by dividing the silicon surface into a number of specific locations. Each of these locations is known as a storage element or potential well, and is created by the field from a pair of gate electrodes very near the surface of the silicon. When the storage elements are placed adjacent to one another, the voltage on the gates can be alternately raised and lowered, thus causing movement of the stored charge from one storage element to the next. In this way, a simple analog shift register has been created, and the charge which occurs upon illumination of the register can be converted to a voltage at the output of the shift register. Figure 2.4 illustrates the process of charge transfer with a two-phase clocking system.[6]

Image Transfer

An example CCD which can be used to describe characteristics typical of a CCD sensor is the TC211 CCD image sensor from TI. The TC211 is configured into 165 lines of 210 sensing elements each. Twelve of the 210 sensing elements are shielded from incoming light and can thus be used to provide a dark reference restoration of the black level of the video image. Figure 2.5 is a functional block diagram.[7]

The inputs to the sensor are comprised of three clock signals (image area gate (IAG), serial register gate (SRG), and antiblooming gate (ABG)). The supply voltage, and the ground reference provide the remaining inputs.



A two-phase CCD shift register. The two complementary clock voltage waveforms ϕ_1 and ϕ_2 are connected to alternate closely-spaced gate electrodes on the surface of the thin insulating layer on the silicon. A deep potential well which attracts electrons is created under the electrode clock voltage HIGH and disappears under the electrodes at clock voltage LOW At t = 0, ϕ_2 voltage is HIGH and the finite charge packet of seven electrons is in the potential well under gate electrode #2 in storage element "A". At t = ½ cycle later, the potential well under gate #2 has collapsed due to ϕ_1 having gone LOW, and, since at the same time the adjacent electrode #3 connected to ϕ_2 has gone HIGH, the seven electron charge packet has been attracted to the new potential well under electrode #3 Another half cycle later, at t = 1 cycle, the potential well under electrode #3 has collapsed with ϕ_2 going LOW and the electron packet moves to the new well under electrode #4 which has gone HIGH with clock voltage ϕ_1.

Figure 2.4

An Illustration of Charge Transfer
Utilizing a Two-Phase Clock

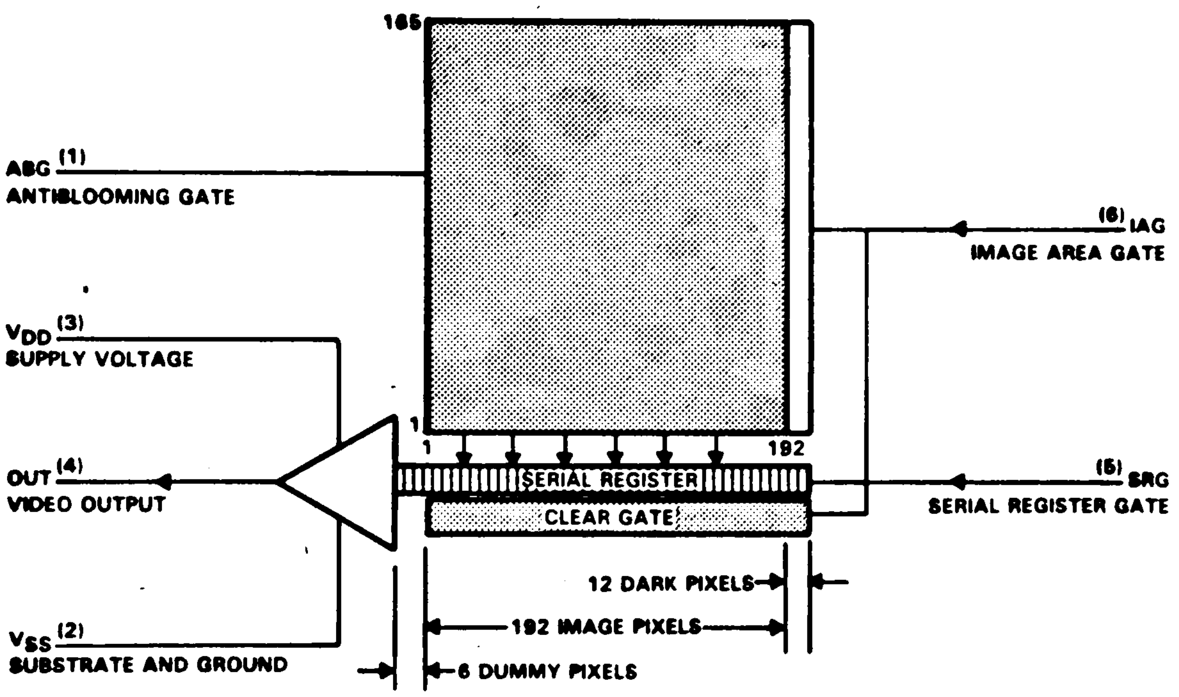


Figure 2.5 Functional Block Diagram for the TC211

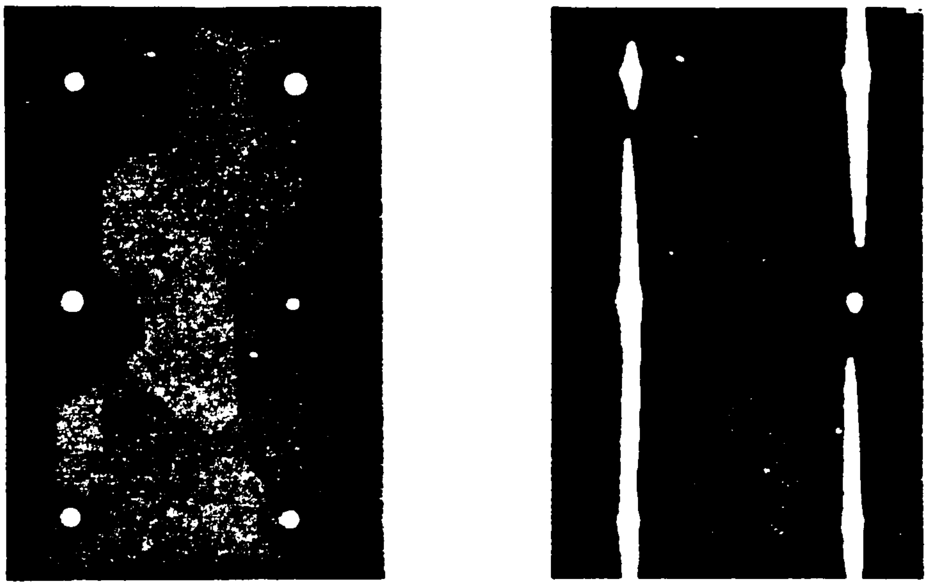
A buffered analog output supplies the video information stored in the photosensor elements.

The IAG is used to transfer charges, a row at a time, into the serial output register. Additionally, each pulse applied to IAG causes an automatic fast clear of the serial register before the new charges are transferred in. This feature permits a dumping cycle in which the entire image area may be initialized before an image is captured, thus clearing any unwanted charge present in the storage elements.

SRG is the serial register clock which causes the packets of electrons present in the serial output shift register to be transported to the 2 stage source-follower output amplifier. Each pulse applied to the SRG input results in the conversion of a charge packet to a voltage level that appears at the video output pin. Charge is converted to signal voltage at 4 uV per electron.

ABG is the antiblooming clock, which prevents blooming from occurring. Blooming is the result of a potential well becoming saturated with electrons, and is caused when local areas of an image sensor are illuminated with an intensity several orders of magnitude greater than the surrounding regions. An image that might cause blooming would be the illumination of a highly reflective object. Once a storage element becomes saturated, the charge stored in the element begins to spill over into adjacent cells and causes streaks to appear in the stored image above and below the region of high illumination, as illustrated in Figure 2.6.

Because it is not possible to control the content of an image that is input to the CCD sensor, some other means of controlling blooming is necessary. A solution is to provide an additional gate, which is an integral part of each sensing element. The surface of CCD sensor is exposed to the incident light during the exposure or integration period. It is during this time that clock pulses are applied



Anti-blooming control provided by accumulation around photosites. The light spot at right center just saturation the CCD; the light spot at the upper left is at 10 times saturation and the remaining spots are at 100 times saturation. The photograph on the left shows the result with the anti-blooming control active and the photograph on the right shows the result with the anti-blooming control inactive. (Courtesy of RCA.)

Figure 2.6 An Example of Blooming

to the antiblooming gate. These clock pulses drain the excess unwanted charge from each storage element to ground.

After exposure to the image, the charge stored in the CCD array is read out, pixel-by-pixel, one row at a time, until all 165 rows have been output from the device. Once this has occurred, a new dumping cycle may begin followed by another integration cycle and readout cycle. A timing diagram illustrating the integration and readout cycles is given in Figure 2.7.

Traditional Imaging System Hardware

The block diagram in Figure 2.1 is intended to demonstrate the functional blocks that are present in typical imaging systems. Figure 2.8 illustrates the hard ware present in an imaging system which utilizes a CCD sensor for image capture, dynamic VRAMs for image storage, and a CRT for image display. This implementation demonstrates the hardware required to implement an imaging system utilizing discrete logic.

Image Sensor Timing Generation

The image sensor timing generation logic provides the clock signals necessary to move an image stored in the CCD sensor into the video memory. As discussed in the previous section, the signals required to drive the TC211 image sensor are the serial register gate (SRG), the image area gate (IAG), and the antiblooming gate (ABG). In order to provide these signals at the appropriate times, as dictated by the timing diagram in Figure 2.7, a number of counters are required. The counters would be driven by a master clock signal which establishes

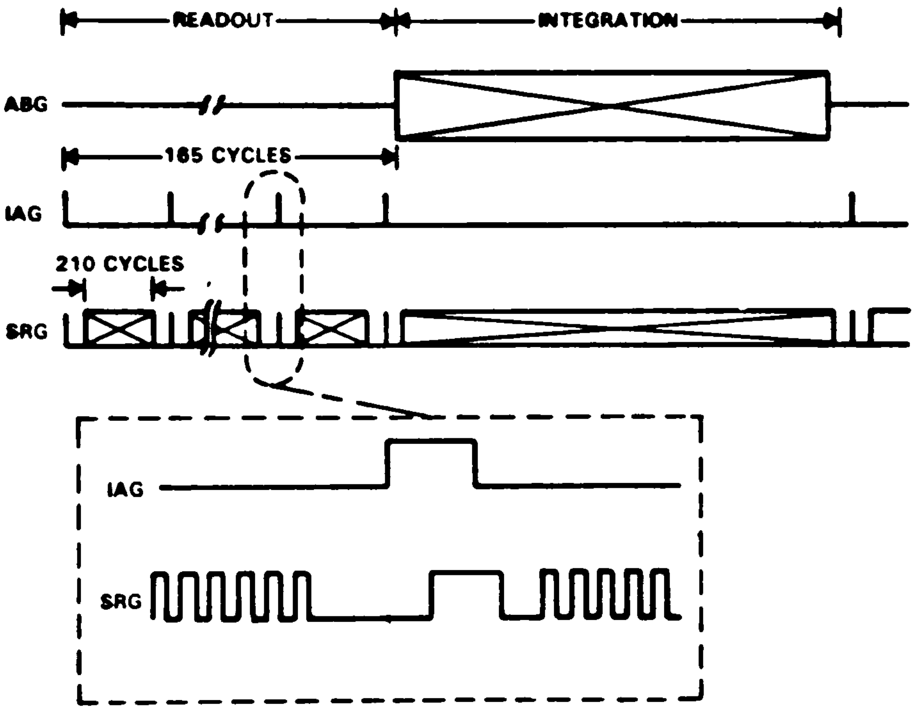


Figure 2.7 CCD Timing Diagram

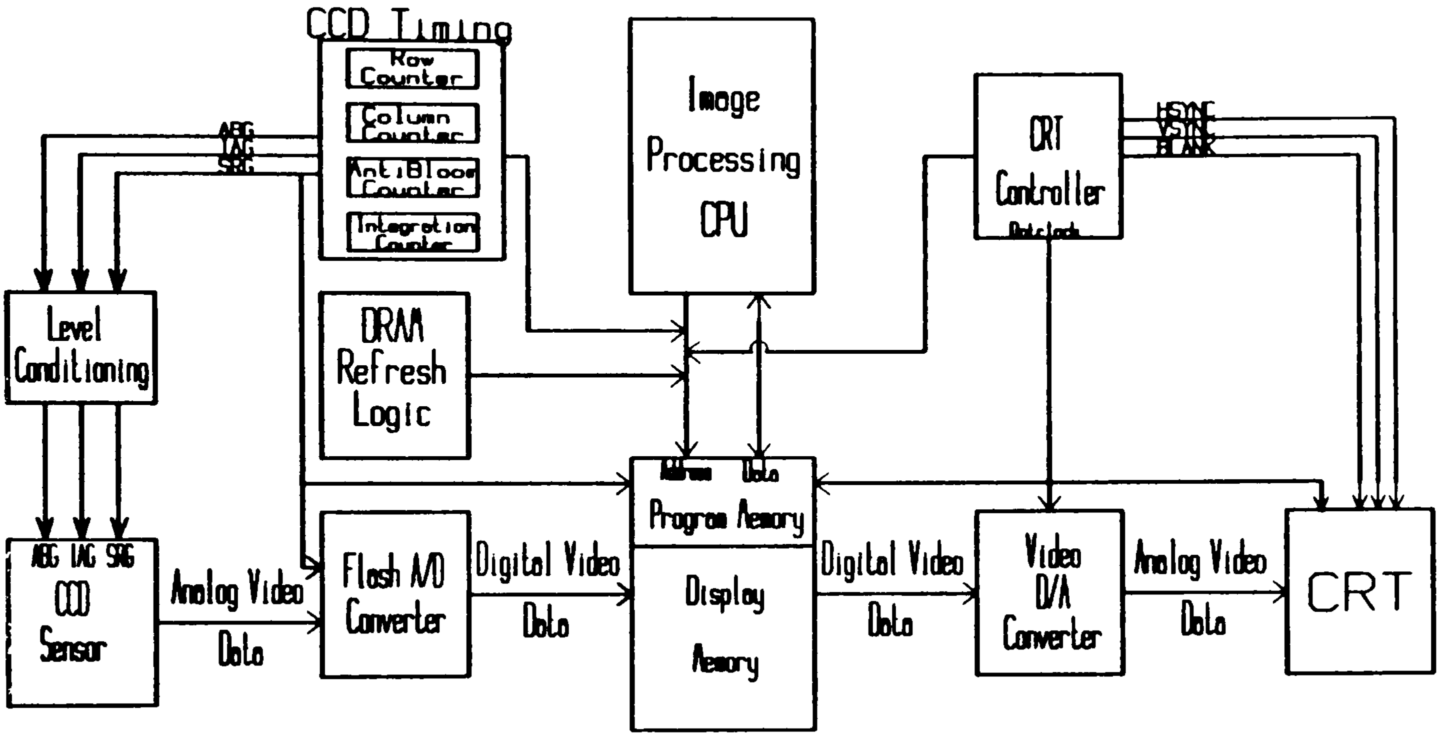


Figure 2.8

A CCD Based Imaging System Utilizing Discrete, Rather than Highly Integrated, Functional Blocks the rate at which pixels are output from the image sensor. A column counter is necessary to keep track of the number of pixels output for a line, thus producing the SRG signal. A row counter is necessary to keep track of the number of lines in each frame of image data, thus producing the IAG signal. Two more counters are then required for the acquisition of an image--one is used to regulate the exposure or integration time, and the other is used to provide antiblooming pulses during integration. Furthermore, if it is an additional requirement that any of these parameters be varied, such as the integration time, then a loadable counter must be provided for timing that particular parameter. A block diagram for a camera driver utilizing these counters is shown in Figure 2.9.

It should be clear that, if standard TTL or CMOS logic is utilized, the implementation of this camera driver would be quite costly in terms of board real-estate, power consumption, and parts cost. The preceding discussion names only the number of counters which would be required, and makes no mention of the hardware required to decode the states of the counters and generate control signals.

Level Conditioning Hardware

Because the TC211 is designed to operate with voltage levels different from standard TTL or CMOS logic, it is necessary to provide level conditioning hardware to translate standard logic signals into voltage levels which will properly drive the image sensor. This hardware will be present on any imaging system which uses standard logic to drive an image sensor.

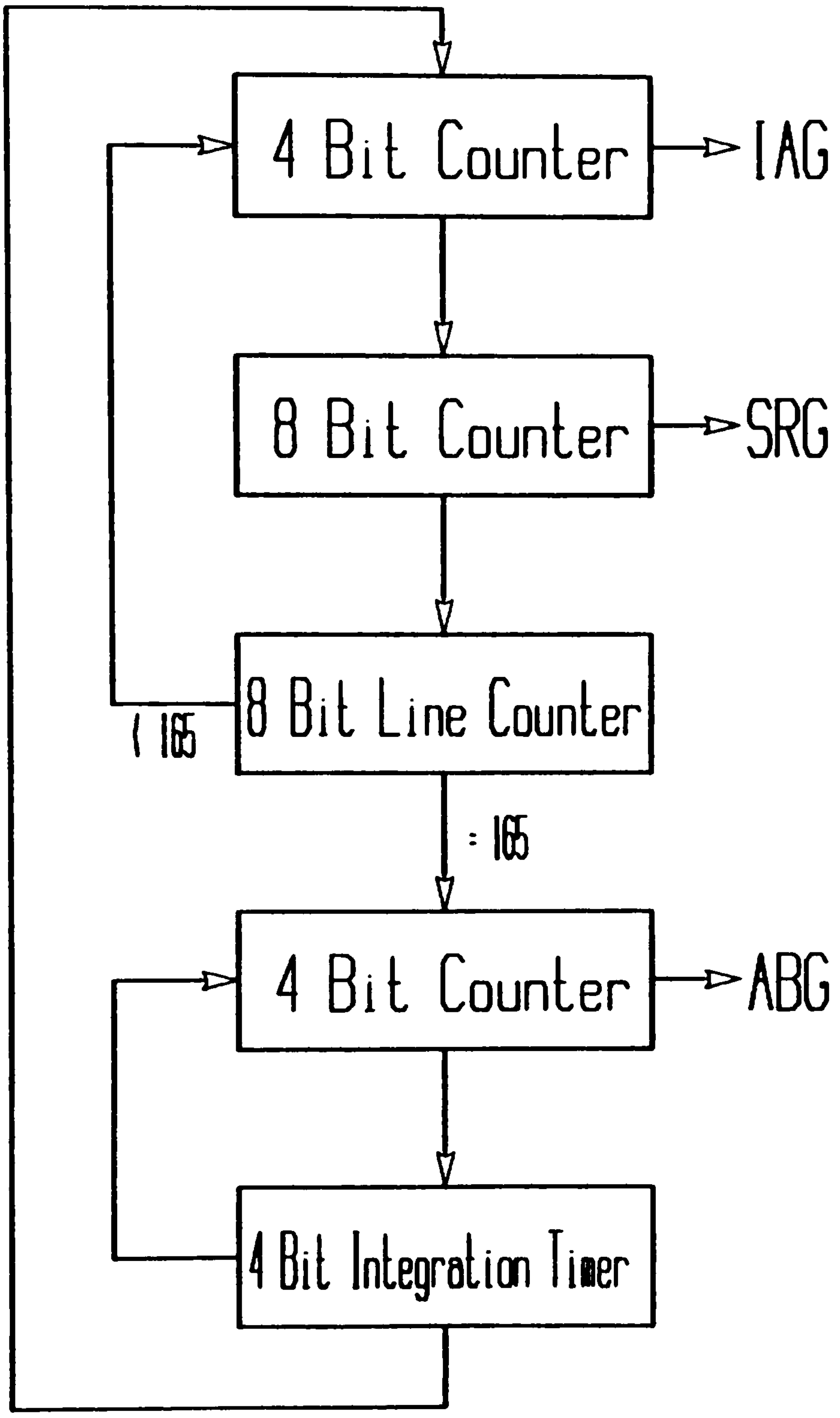


Figure 2.9 Block Diagram for a CCD Camera Timing Generator

A/D Converter

The A/D converter is used to digitize the analog data output by the image sensor. Video data is usually digitized at very high speed because of the large amount of data present in an image. Conversion rates of 10 MHz and higher are commonplace in a video capture environment. For this reason, successive approximation A/D converters, with maximum conversion rates of approximately 100 KHz, would not be suitable. Flash A/D converters, however, can provide conversion rates greater than 100 MHz and are therefore well suited for video applications.[8]

Refresh Logic

The large amount of memory required for digital image processing and storage dictates that dynamic, rather than static memory, be used for image storage. Dynamic memories, although more difficult to use, provide an inexpensive and compact storage medium when compared to static memories. The difficulty presented by dynamic memory devices is caused by the requirement that support hardware be provided to DRAM arrays. This requirement is due to the physics which govern the operation of a DRAM storage cell. Information is stored in a DRAM, by the presence or absence of charge on a capacitor. Because capacitors are not ideal storage devices, and will lose charge over time, the charge on the capacitor must be periodically refreshed in order to retain information; thus, refresh logic must be present on any system utilizing dynamic memory devices. A refresh controller may be built from discrete logic devices, or one may be selected from a number of integrated versions available.

Video Memory

The video memory is used to store the digitized data output by the A/D converter. Once the image data is stored in the video memory, it can be processed by an imaging CPU or directly output to a CRT. The video memory utilizes dynamic video random access memory (VRAM) devices which provide a simple and efficient means for loading an image into the memory array, and subsequently displaying that image on a display device.

An illustration of the difference between DRAM and VRAM devices is provided in Figure 2.10 and Figure 2.11, respectively. Figure 2.10 is the block diagram for a TMS4164 DRAM, and Figure 2.11 is the block diagram for a TMS416 VRAM.[9] It can be seen that the two devices are nearly identical, except for the presence of a shift register on the VRAM. The purpose of the shift register is to provide the memory with two access ports; one a random access port for use by a CPU, and the other a serial access port specifically targeted at applications, such as refreshing a CRT display system. The two ports enable a processor to place data which is to be displayed on a CRT into the memory through the random access port. That data may then be shifted out of the serial port at a rate fast enough to provide pixel data for a CRT in a raster scan display system. In this way, processor access to the display memory does not impede screen refresh operations, and the two processes can thus take place simultaneously.

Not only can the VRAM serial port be used to move data out of the RAM array, it can also be used to load data into the array. Data from an external source, such as a CCD sensor, can be digitized and then shifted into the register at high speed with virtually no interaction. All that is required of the control logic is to produce the control signals and VRAM addresses which cause transfers to

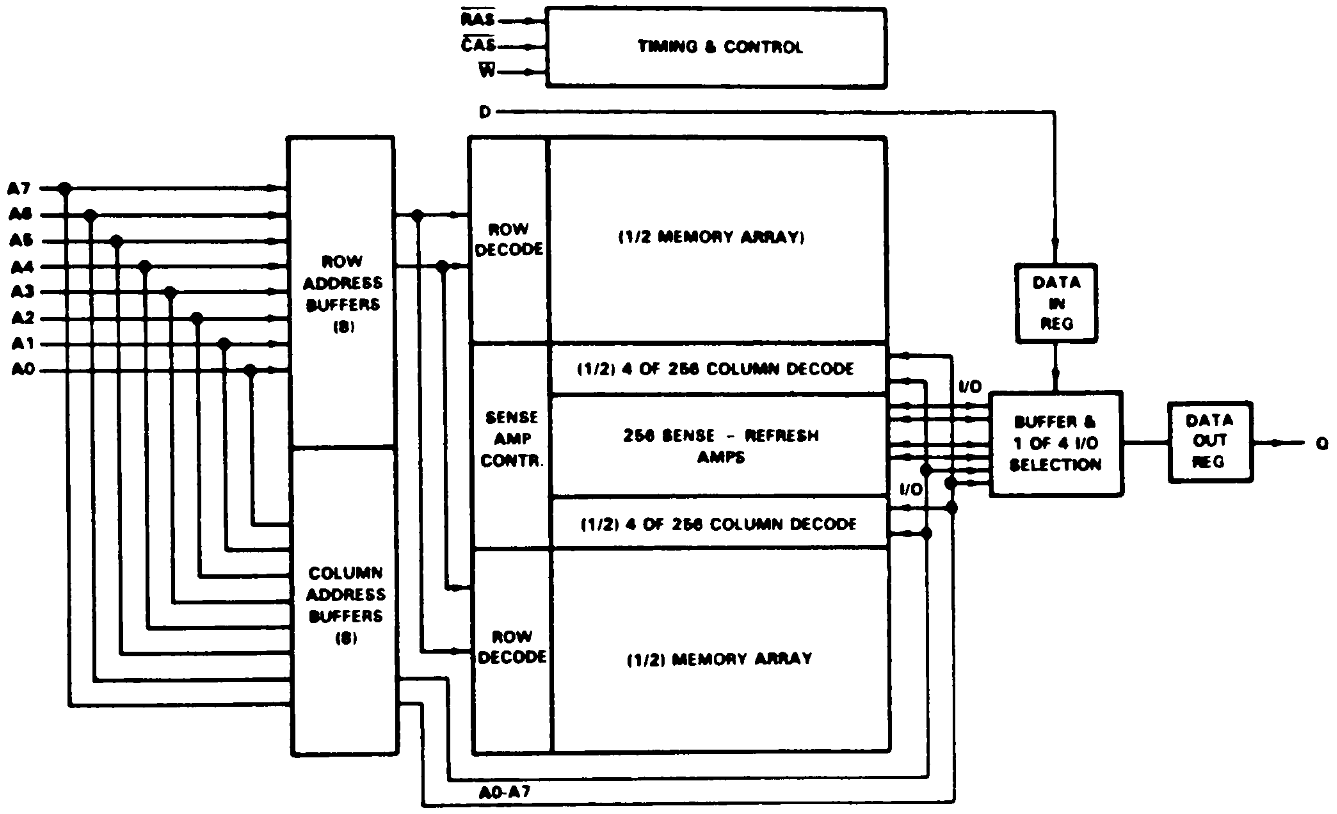


Figure 2.10 Block Diagram for a DRAM

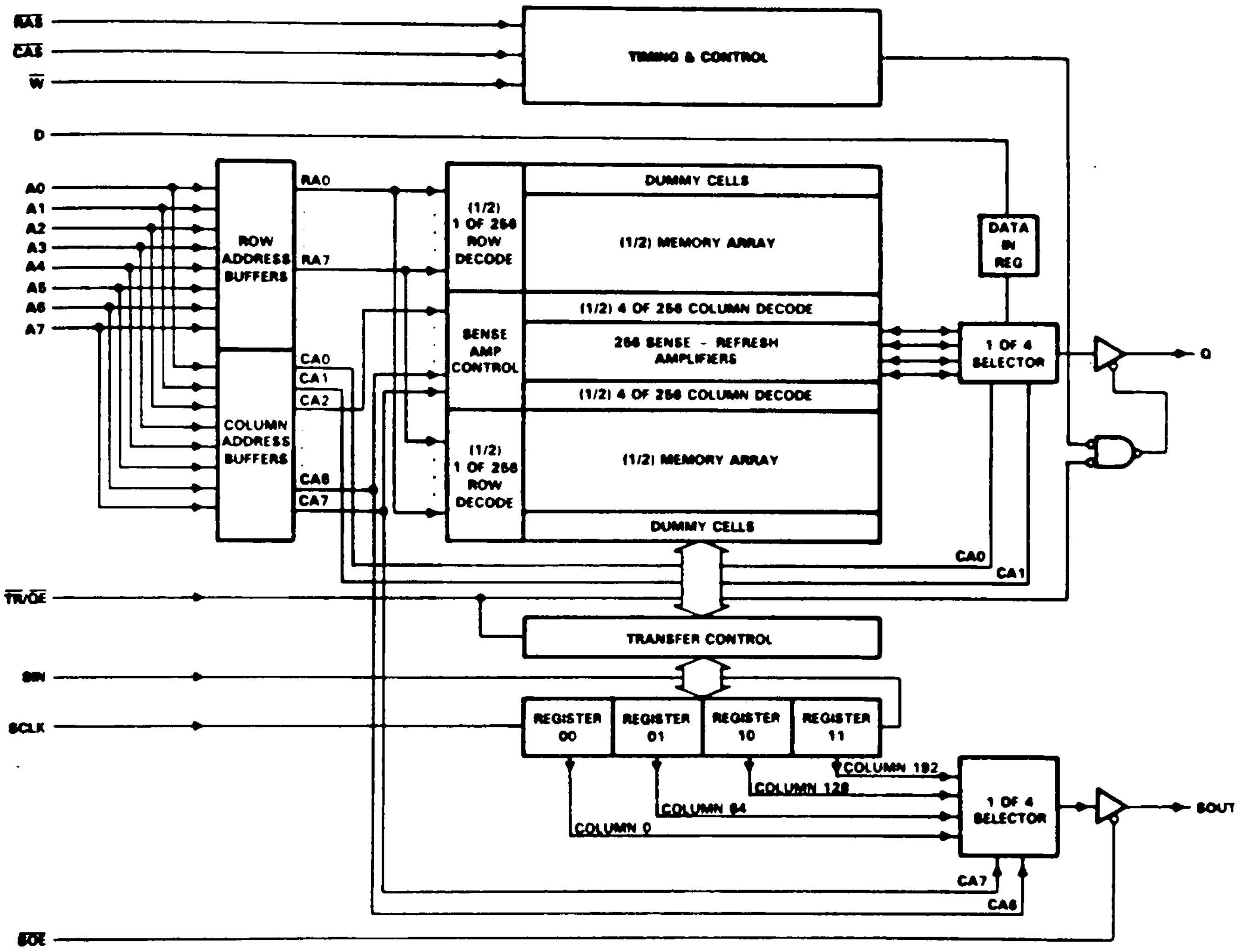


Figure 2.11 Block Diagram for a VRAM

occur from the shift register to the memory array whenever the shift register becomes full.

CRT Control

Before discussing timing signals that control a CRT, a brief discussion of CRT operation is in order. The image displayed on a graphic display device, such as the CRT of a video monitor, is not actually one image, but a series of images painted on the screen anywhere from 30 to 70 times per second. This is what enables the CRT to display what appears to be the continuous motion of an object or group of objects. Each time an image is drawn on the CRT is referred to as a screen refresh cycle.

The image which appears on the screen during a refresh cycle is made up of a group of closely spaced horizontal lines called the raster. An electron beam successively scans each line, beginning at the upper left corner of the screen. An illustration of the path of the electron beam is shown in Figure 2.12. The beam sweeps across the screen, from left to right, displaying one pixel at a time. The rate at which the beam scans across the screen is determined by the dot clock or pixel clock. Typical dot clock rates are on the order of 25MHz or more for high resolution graphics monitors.

When the beam reaches the end of a line, it is deflected downward and to the left, to the beginning of the next line. The next line is then scanned and the process repeats until the last line is displayed, at which time a new screen refresh cycle begins. A number of carefully timed events occur during a screen refresh cycle in order to produce the image as described above. The timing signals which

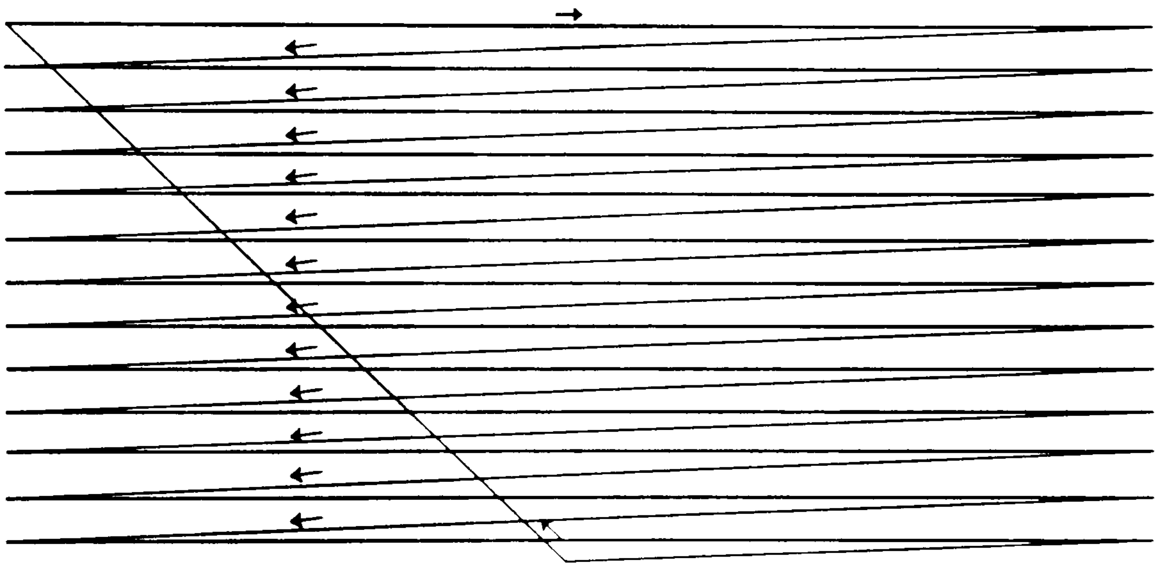


Figure 2.12

The Path Followed by an Electron Beam
During a Screen Refresh Cycle.

control the operation are the horizontal sync pulse, the vertical sync pulse, the blanking signal and the dot clock. [10]

The video timing signals which need to be provided to a video monitor are:

Horizontal Sync	(HSYNC)
Vertical Sync	(VSYNC)
Blank	(BLANK)

The purpose of the HSYNC pulse is to indicate when the beginning of a new scan line occurs, while the VSYNC pulse indicates the beginning of a new video frame. The BLANK signal indicates the times for which the CRT's electron beam should be turned off, such as occurs when the beam is deflected to the beginning of a new scan line, or when the beam reaches the end of the frame and must be sent to the upper left corner of the screen. The three timing signals are produced by counting the dot clock signals which cause the electron beam to move across the display. Although it is possible to construct a CRT controller from discrete hardware, there are integrated CRT controllers available.

D/A Converter

The purpose of the D/A converter is to convert digital data output from the VRAM shift register into an analog waveform which drives the video monitor. The D/A converter must be capable of conversion rates in excess of 25 MHz.

Although any D/A converter that is capable of running at this speed would be acceptable, external hardware would likely be necessary to drive a video monitor. For this reason, special video D/A converters are available and are capable of directly driving 75 ohm inputs on standard monitors. Additionally, some of these devices may also have blanking and sync inputs, thus reducing grated Device Technology, is shown in Figure 2.13.[11]

As can be seen, the implementation of a digital imaging system can require a significant amount of hardware. This is true because traditionally, imaging systems utilize discrete hardware to provide the timing signals for image capture and display devices, and to provide refresh addresses for dynamic memory devices.

The hardware required to implement an imaging system can be significantly reduced by utilizing specialized hardware. A specific device which provides many of the functions required by an imaging system, the TMS34010 graphics system processor, will be discussed in the following chapter.

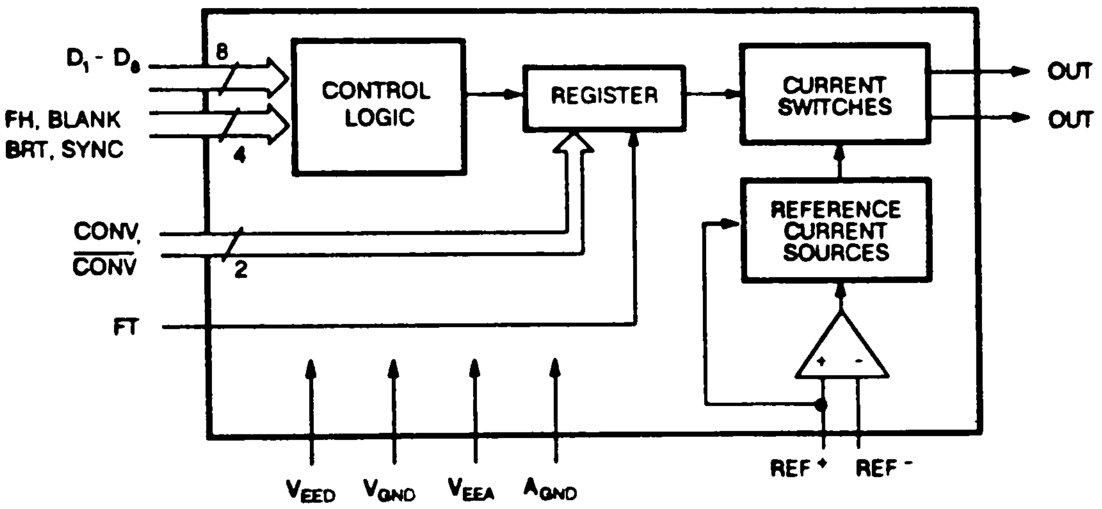


Figure 2.13 Block Diagram of a Video D/A Converter

CHAPTER III
THE TMS34010 GRAPHICS SYSTEM PROCESSOR

The TMS34010 graphics system processor combines the features of general purpose microprocessors with hardware typically present in special purpose graphics controllers to provide a microprocessor ideally suited for use in a graphics environment. Features provided by the GSP include: refresh circuitry for dynamic memory devices, direct interface to DRAMs and VRAMs with no external address multiplexing, control of the VRAM serial port, screen refresh circuitry for automatic update of a display device, and a microprocessor with an instruction set suited for both general purpose applications and specialized graphics processing and control applications.

Architectural Overview

The TMS34010 graphics system processor is an advanced 32-bit microprocessor, optimized for graphics operations. The 34010 can operate in a stand-alone configuration, or it can serve as an embedded graphics controller for a host computer. Figure 3.1 illustrates the use of a GSP in a simple graphics system.

The GSP provides most of the functions necessary to implement a graphics system. Figure 3.1 shows the major GSP internal functions enclosed by a dashed line. Internal to the GSP is the graphics processor, the host interface, DRAM refresh control, screen refresh control, and CRT timing control. The DRAMs

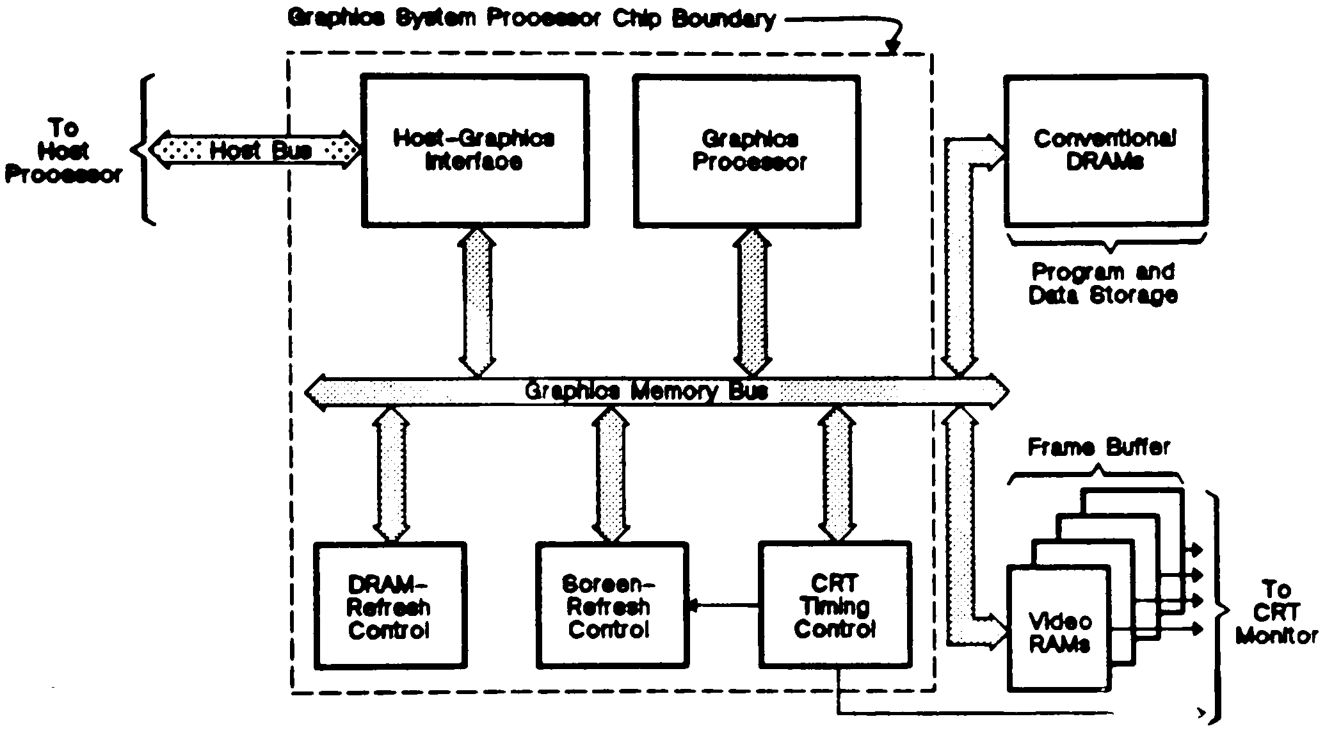


Figure 3.1 A Simple GSP Based Graphics System

used for program and data storage, and VRAMs used for image storage, reside external to the GSP and are connected via the graphics memory bus.

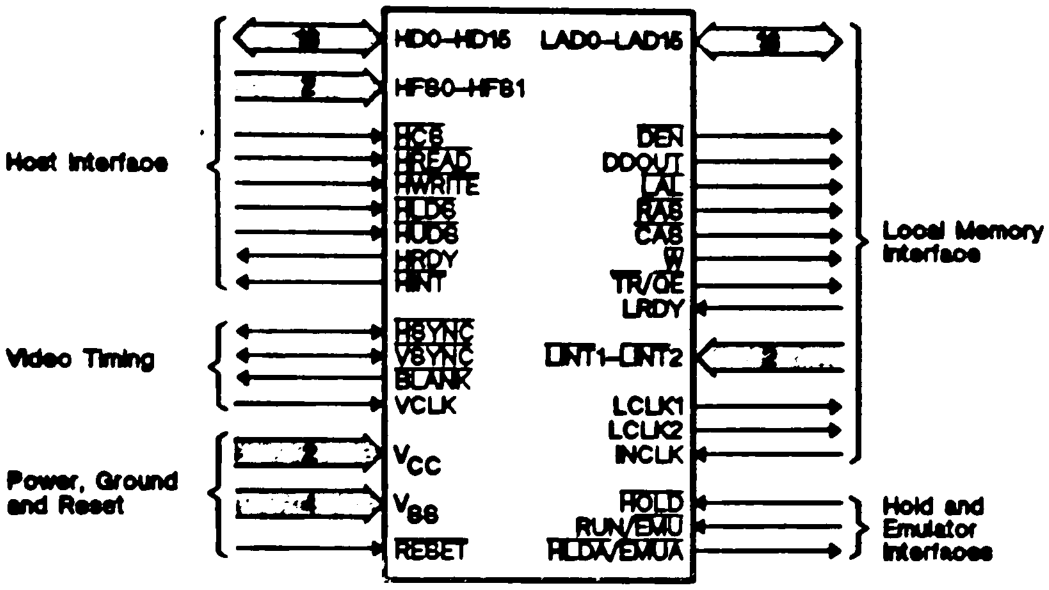
The graphics processor executes both graphics instructions and general purpose instructions. A number of general purpose instructions are provided for performing tasks such as: move and store operations, subroutine calls and returns, conditional and unconditional jumps, and shifting operations. Graphic instructions include: 6 arithmetic and 16 Boolean pixel processing operations, line drawing, filling of pixel blocks, moving of single pixels and two dimensional blocks of pixels. Additionally, the bit addressable graphics processor supports a number of pixel sizes, provides automatic bit-alignment, and is capable of direct, indirect, and X-Y addressing of an external memory as large as 128 megabytes.[12]

Internal Architecture

An illustration of the major external interfaces to the TMS34010 is provided in Figure 3.2. Figure 3.3 provides a glimpse into the internal architecture of the GSP, and how the architecture is associated with the external interface.

Host Interface

The host interface on the GSP allows a host processor to access data and control registers internal to the GSP, and also provides the means for addressing the local memory space. It is through this interface that the host may configure video timing registers, download data and or programs to the GSP local memory, and perform many other tasks necessary in a host environment. Efficient transfer of data to the local memory is supported via an auto-incrementing address register



Host interface	25 pins
Local memory interface	29 pins
Video timing interface	4 pins
Hold and emulator interfaces	3 pins
Power and reset	7 pins

Figure 3.2 Major External GSP Interfaces

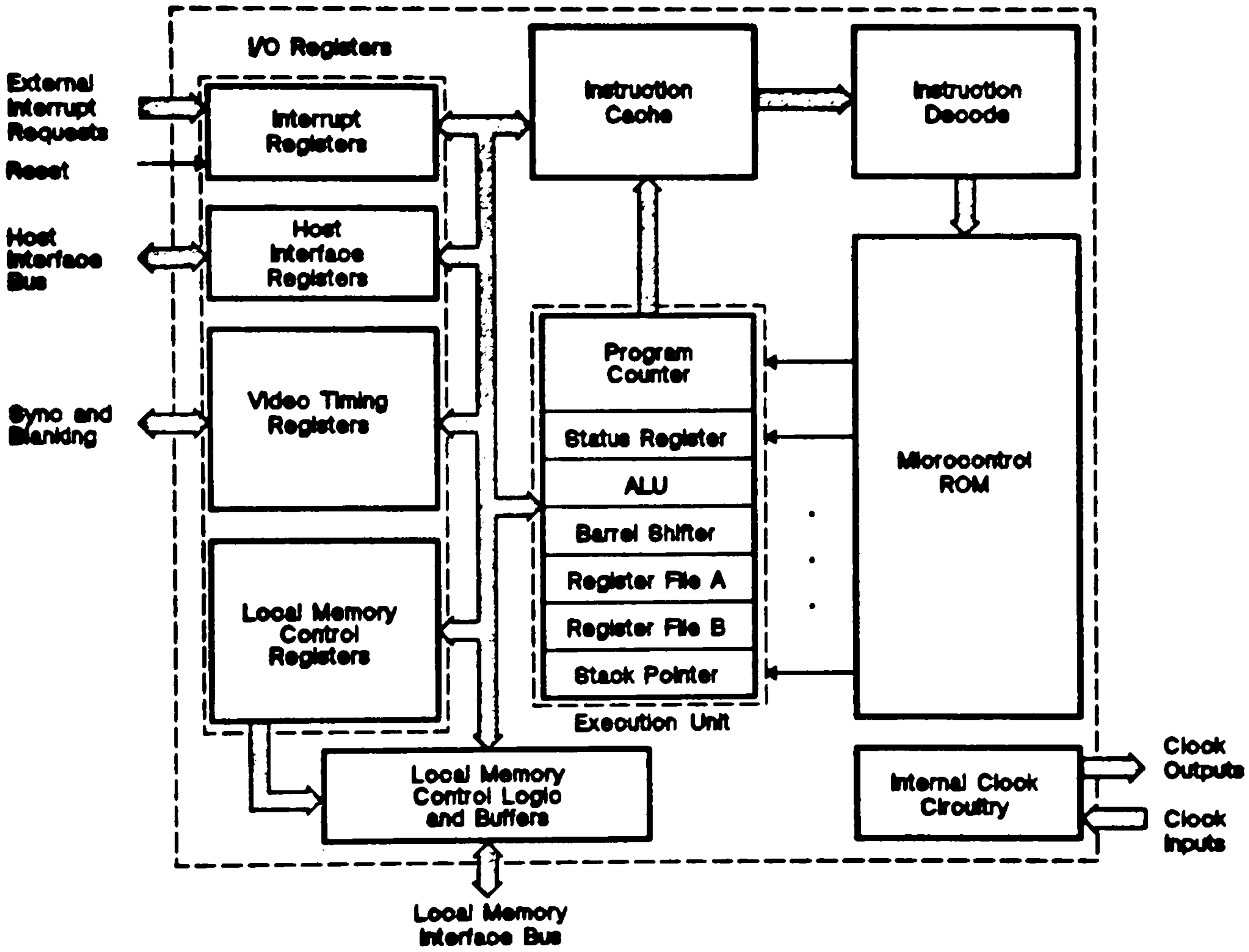


Figure 3.3 GSP Internal Architecture

and data port. Communication between the host and the GSP is established through an interrupt/interrupt-acknowledge handshaking sequence.

Although a host interface is provided on the GSP, its use is not mandatory. The GSP is designed as a general purpose microprocessor, and as such, can be configured to boot from, and run programs out of, a read only memory (ROM). This enables the GSP to serve as the sole processor in a stand-alone application.

Local Memory Interface

A TMS34010 graphics system typically utilizes two different types of main memory. Dynamic random access memory (DRAM) devices are used for program and data storage, while video random access memory (VRAM) devices are used for storing images. Additionally, ROMs may be present for non-volatile program storage for applications such as bootstrap routines.

The local memory interface of the GSP consists of a triple-multiplexed address/data bus and the associated control signals. Row addresses, column addresses, and data are all transmitted over the same bus. Row and column addresses are presented in a manner which permits direct interface to DRAM and VRAM arrays.

Control signals provided at the local memory interface include: row and column address strobes (RAS' and CAS'), output enable, latching, and bus direction control signals (DEN', LAL', and DDOUT), a write strobe (W'), and a VRAM shift register control signal (TR'/QE'). The VRAM shift register control pin allows direct control of the VRAM shift register by the GSP.

Additionally, the local memory interface provides the means for allowing other hardware to directly control the local memory interface through a hold/hold-acknowledge handshaking sequence. This interface permits other processors to request use of local memory space to perform any task which may be desired.

DRAM refresh cycles can be programmed to occur automatically if so desired. The time interval between refresh cycles is also programmable. The types of refresh cycles allowed include both RAS'-only, and CAS'-before-RAS'; thus, the GSP is capable of supporting the latest generations of available DRAMs as well as earlier, less sophisticated DRAM devices.

Video Timing Interface

The video timing interface allows the GSP to directly control an output display device such as a CRT or a liquid crystal display (LCD). The TMS34010 produces the synchronization and blanking signals necessary to drive a video screen in a graphics system. The GSP can be programmed to support a wide variety of screen resolutions, and can provide interlaced or non-interlaced video output. The GSP also provides direct support of VRAM devices by producing the memory-to-shift register transfer cycles necessary for updating a display.
This is all accomplished by providing a number of programmable registers which determine when synchronization, blanking and transfer signals should occur.

Figure 3.4 illustrates the relationship between the blanking (BLANK), horizontal sync (HSYNC') and vertical sync (VSYNC') signals. Also included in this figure are the names of programmable registers which enable the timing signals to be produced. For example, suppose it is desired to program the waveform for the HSYNC' signal. By programming the horizontal end sync register (HESYNC) and the horizontal total register (HTOTAL), the host processor

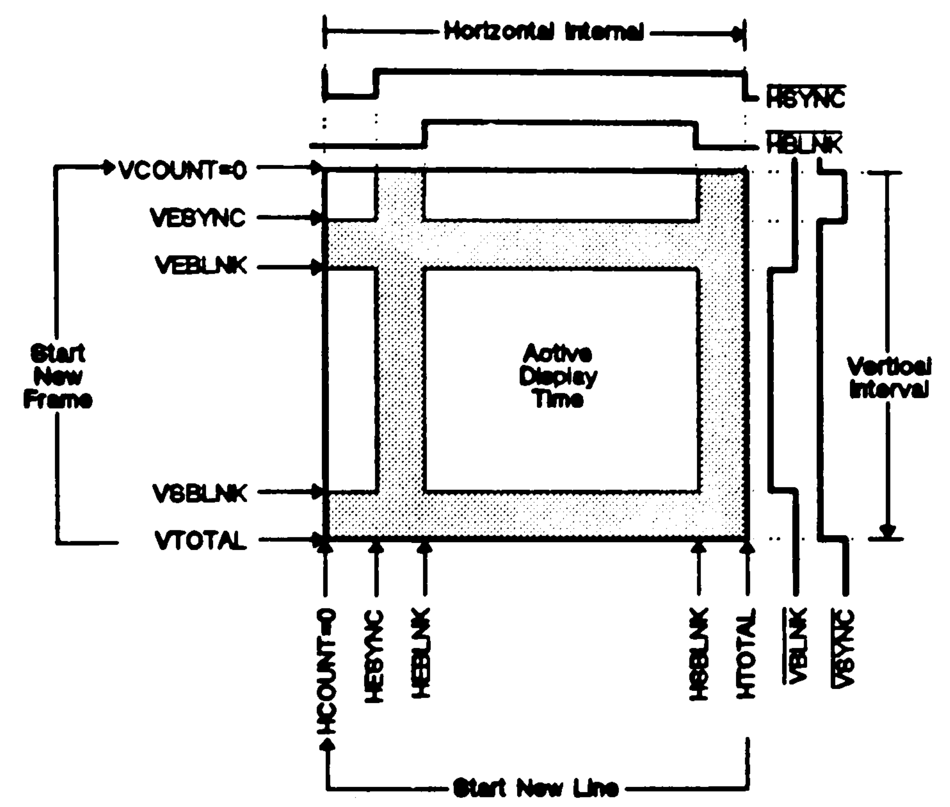


Figure 3.4 Horizontal and Vertical Timing Relationship

informs the GSP how many VCLK cycles occur in one horizontal line. A comparator then monitors the HCOUNT register, which keeps a running total of the number of VCLKs for the current line, and when the HCOUNT register reaches the number stored in HTOTAL the GSP has reached the end of the current scan line and the horizontal sync pulse is asserted. The horizontal end sync (HESYNC) register then provides the number of VCLK periods for which the HSYNC' signal should be asserted.

Other timing signals are produced in a similar manner, but it should be noted that internally the GSP produces horizontal blank (HBLANK') and vertical blank (VBLANK') signals. These two signals are logically ORed to produce the BLANK' signal which is output by the GSP.

What makes the GSP particularly useful for performing screen refresh operations is its ability to control the shift register on a VRAM. Screen refresh operations, when enabled, take place automatically, and pixel data stored in the VRAM array is shifted from the RAM array to the VRAM shift register under GSP control. Conversely, by providing a small amount of external hardware, it is possible to have the GSP provide timing signals to an image sensor, and cause shift register-to-VRAM transfer cycles to occur automatically. In this way, the very same signals that are used to drive the HSYNC', VSYNC', BLANK', and dot-clock signals on a CRT can be used to drive the IAG, SRG, and ABG inputs on a CCD sensor.

Examples of TMS34010 Systems

Since the introduction of the GSP in the spring of 1987, the device has been utilized in a large number of applications with widely varying characteristics. The

GSP has been applied to graphic display systems, laser printer controllers, medical imaging equipment, and control system applications, just to name a few examples. This is possible because of the flexibility provided by the GSP not only as a graphics controller, but also as a general purpose microprocessor. An example of GSP based systems which are currently available can be seen in the TMS34010 Third Party User's Guide available from TI. This reference provides information regarding hardware and software implementations for GSP systems including: display and printer control systems, imaging systems, process monitoring and control applications, and numerous software applications.[13]

Use of the GSP in an imaging system not only provides integration of the hardware required to implement such a system, but also enables use of a wealth of software already written for GSP based systems. An example of software which can be readily applied to GSP based imaging systems is an image processing software package written at the University of Washington. This public domain software, available free of charge from TI via the graphics bulletin board service, provides basic image processing functions such as histogramming, filtering, convolutions, and edge detection, along with many others. By utilizing these routines, and the software development board to be discussed in the following chapter, the user can develop image processing software targeted at specific applications in a timely manner.

CHAPTER IV
THE TMS34010 SOFTWARE DEVELOPMENT BOARD

The TMS34010 software development board is a GSP based circuit board which interfaces to the expansion bus of an IBM PC, or IBM PC compatible computer. The SDB provides an environment for developing software for GSP based systems, and also serves as an example of a GSP based graphics system. Provided with the SDB is a debugger which allows the user to download application programs, single step programs, set breakpoints, and modify and observe the contents of registers internal to the GSP. By utilizing the SDB and the debugger, the user is able to develop application software and evaluate GSP performance before a user specific GSP hardware implementation has been realized. A Block diagram for the SDB is shown in Figure 4.1, and schematics are reproduced in Appendix A.[14,15]

Host Bus Interface

The host interface port provides a means for the PC to communicate with the software development board. The interface allows the host access to registers internal to the GSP, and also to the entire range of program and data memory space. Through this port, the host is able to control operation of the SDB, and download programs and data to the GSP's local memory. Figure 4.2 provides a schematic diagram of the PC bus to GSP interface.

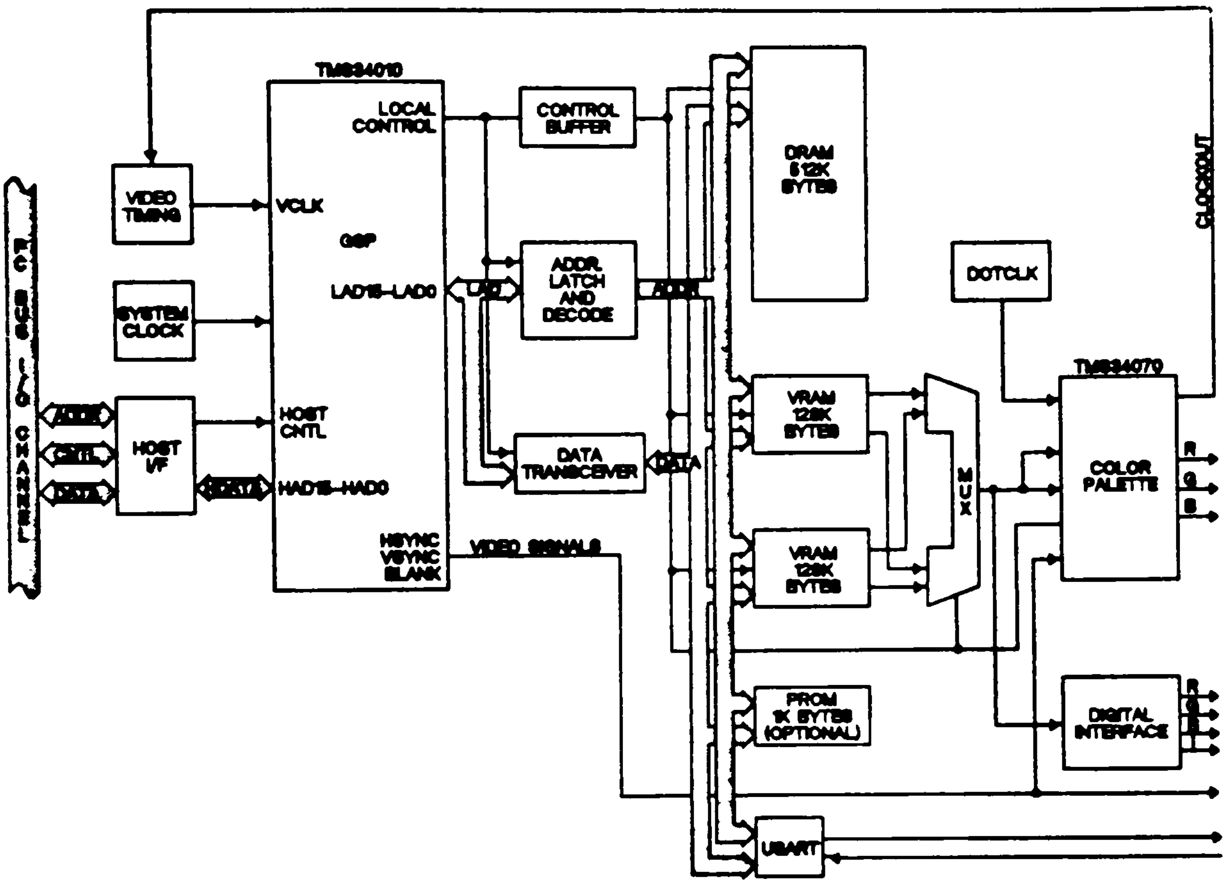


Figure 4.1 Software Development Board Block Diagram

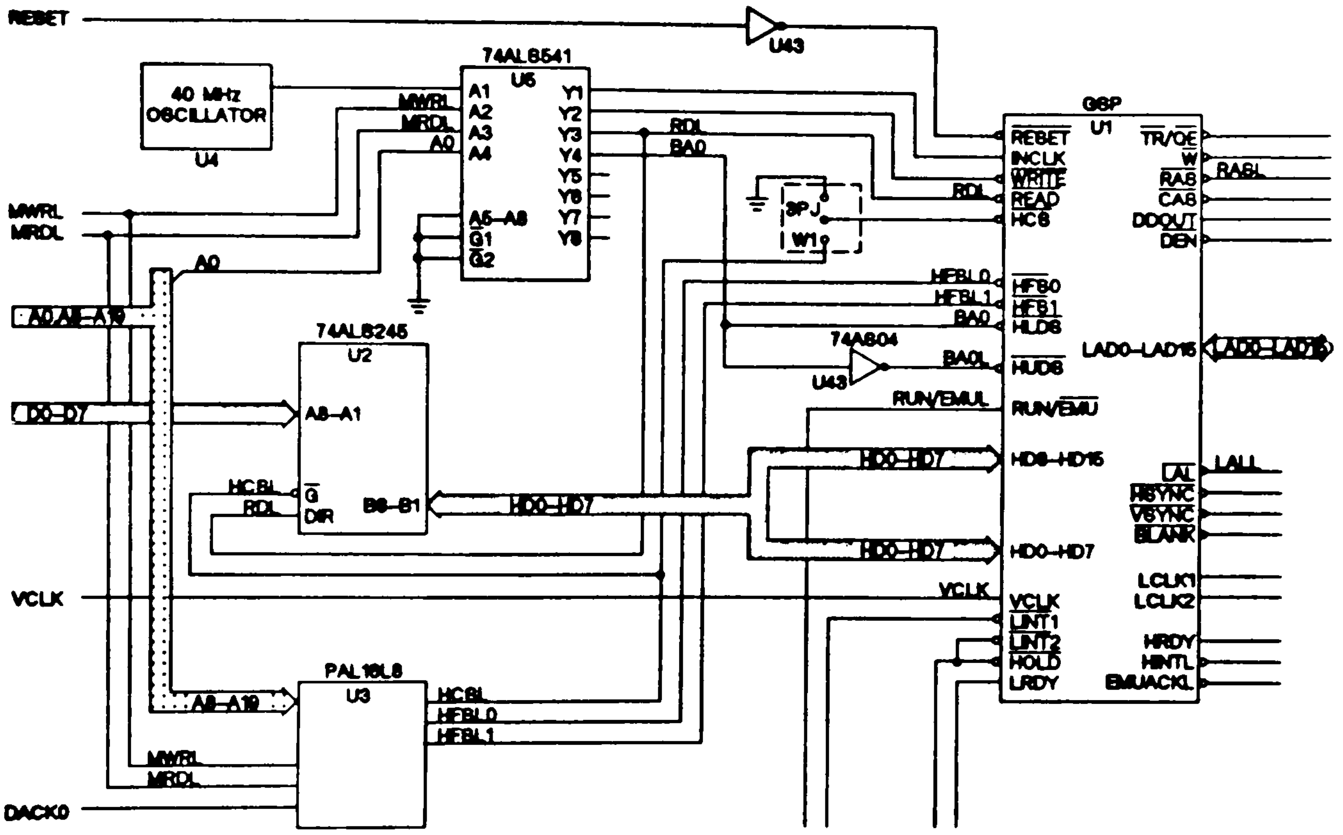


Figure 4.2 PC to GSP Interface Schematic

Local Bus Interface

The local bus provides the GSP with a path to transfer data to and from: a 512 K byte DRAM array used for program and data storage, a 256 K byte frame buffer divided into two 128 K byte banks, and a USART. A memory map for the SDB is shown in Figure 4.3.

Program Memory

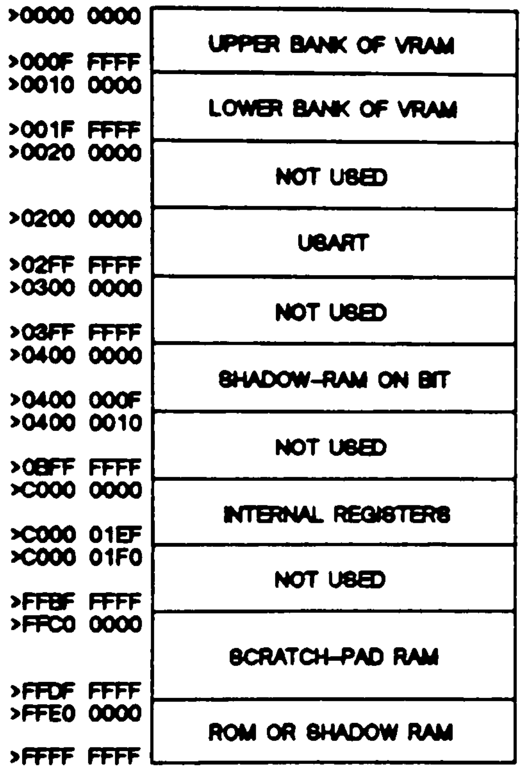
The SDB provides 512 K bytes of onboard DRAM which can be used by the GSP to execute drawing functions, application programs, and displays. The DRAM array is implemented with four TM4256EC4 single inline packages.

Additional program storage can be provided with the installation of TBP28S42 PROMs. These PROMs are optional, and are not installed on the SDB as shipped from TI.

Display Memory

The software development board has 256 K bytes of display memory consisting of 8 TM4161EV4 single inline packages. The memory is divided into two banks, with 64 K words in each bank. This organization enables the SDB to conserve power by having only one bank of memory active at a time.

Each pixel is represented by four bits of data, thus permitting a maximum screen resolution 1Kx512 pixels. Pixels are arranged in a packed pixel format; therefore, each 16 bit word contains data for four pixels. Figure 4.4 provides an illustration of packed-pixel versus bit-plane memory organization.



NOTE: BECAUSE SOME LEAST-SIGNIFICANT MEMORY-ADDRESS BITS ARE NOT DECODED ON ADDRESS LINES, MEMORY AREAS APPEAR LARGER THAN ACTUAL ON-CHIP MEMORY.

Figure 4.3 Software Development Board Local Memory Map

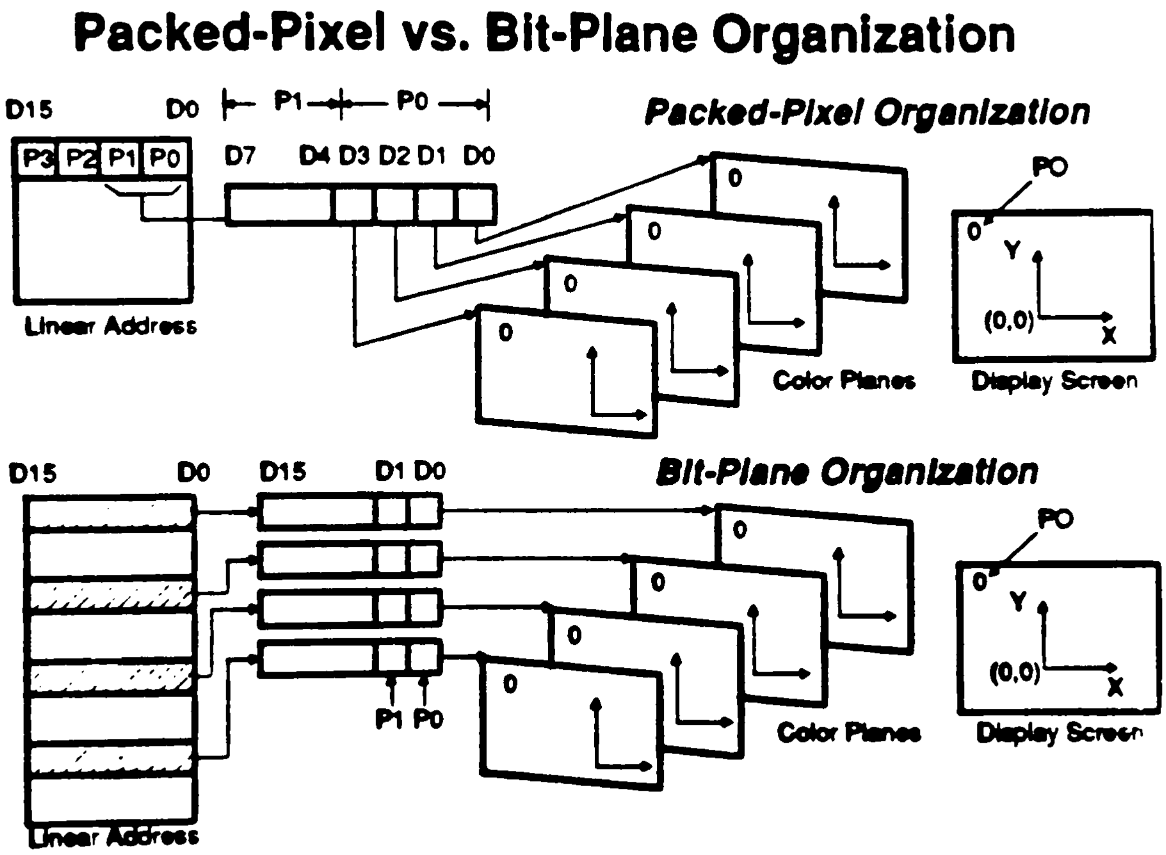


Figure 4.4 Packed Pixel Memory Organization

USART

A universal synchronous-asynchronous receiver transmitter is present onboard the SDB for serial communication ability. The USART is memory mapped into the local memory space as indicated in Figure 4.3. Figure 4.5 illustrates the specific register addresses allocated to the USART.

Frame Buffer to CRT Interface

An illustration of the interface between the CRT and the frame buffer is provided in Figure 4.6. The major feature of this interface is provided by two layers of multiplexing which reduce the speed of the VRAM shift register clock. One layer of multiplexing is provided by the TMS34070 color palette, and the other is provided by a pair of 74LS257 quad 2-to-1 multiplexers.

The TMS34070 color palette is used to convert a four bit input value to display 16 of 4096 colors simultaneously. This is accomplished by providing 16 registers on board the color palette, which select the particular colors to be displayed. The registers can be programmed prior to the start of each horizontal scan line, or at the beginning of each video frame. The outputs of the color palette are capable of directly driving 75 ohm monitor inputs.

The TMS34070 also incorporates a 2-to-1 multiplexer which enables the VRAM shift register to be clocked at a rate one half the speed of the actual dot clock to the CRT. For each half cycle of the dot clock, alternating nibbles of input data are converted to analog signals, and appear at the outputs of the palette.

An additional layer of multiplexing circuitry resides between the serial outputs of the VRAM array and the color palette, and is provided by the

Address	Function
0200 0000	Receive Holding Register
0200 1000	Read Status Register
0200 2000	Read Mode Registers 1 and 2
0200 3000	Read Command Register
0220 0000	Write to Transmit Holding Register
0220 1000	Write to SYN1/SYN2/DLE Registers
0220 2000	Write to Mode Registers 1 and 2
0220 3000	Write to Command Register

Fig. 4.5 USART Register Addresses

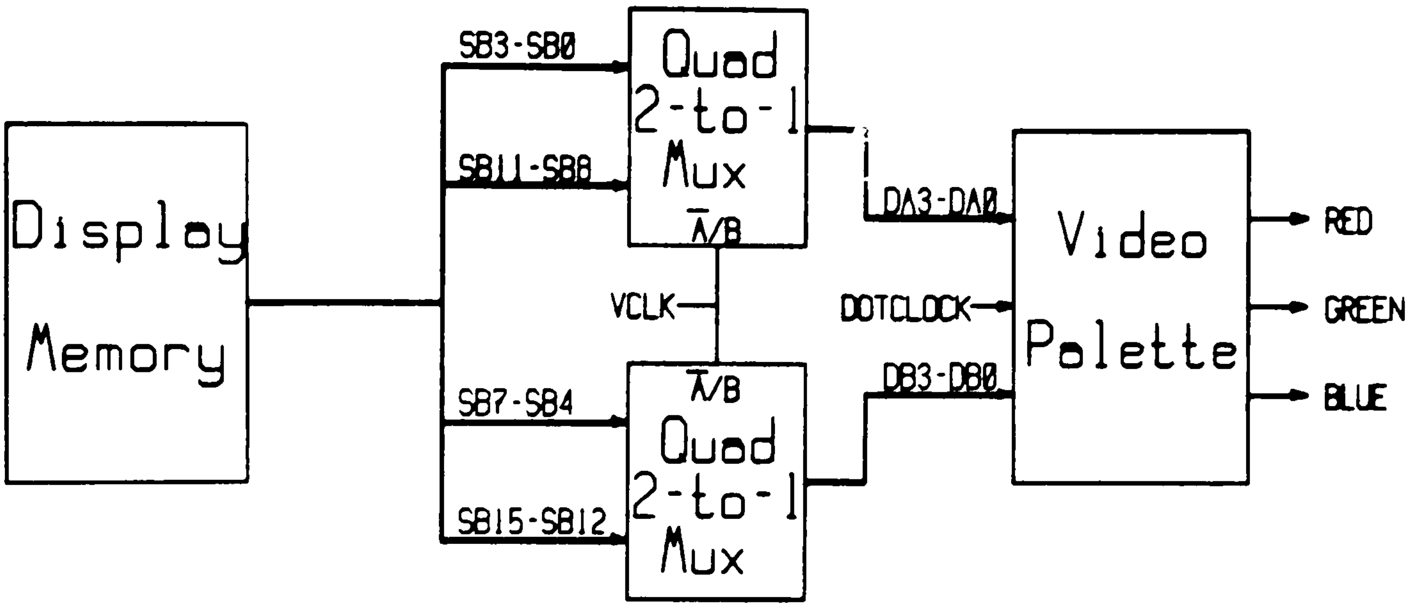


Figure 4.6 Frame Buffer to CRT Interface

74LS257s. This additional layer of multiplexing allows the VRAM shift register to be clocked at a rate 4 times slower than the dot clock of the CRT. For the SDB, driving a Multisync monitor with a dot clock of 25MHz, the VRAM shift registers are thus shifted at 6.25MHz.

CHAPTER V

A TMS34010 SDB IMAGING SYSTEM

The features of the TMS34010 graphics system processor enable the construction of a small, relatively low cost imaging system, that requires limited circuit board real estate. This is possible because many of the features required for construction of an imaging system are provided by the GSP. These features include: refresh circuitry for dynamic memory devices, direct interface to DRAMs and VRAMs with no external address multiplexing, control of the VRAM serial port, screen refresh circuitry for automatic update of a CRT display and acquisition of CCD images, and a general purpose microprocessor for image processing and control applications.

A block diagram of the imaging system to be discussed in the remainder of this chapter is shown in Figure 5.1. Here, the GSP is used to provide the timing signals to drive an image sensor as well as a display device, thus creating a system capable of image capture and display, without the need for specialized image capture hardware.

The imaging system shown in Figure 5.1 represents a unique application of the GSP to an imaging system. Rather than using the GSP only as a display controller and graphics processor, the device is used to provide the timing signals to capture digital images from a video source. Typical GSP based imaging systems do not utilize the GSP to perform image capture, but instead use specialized

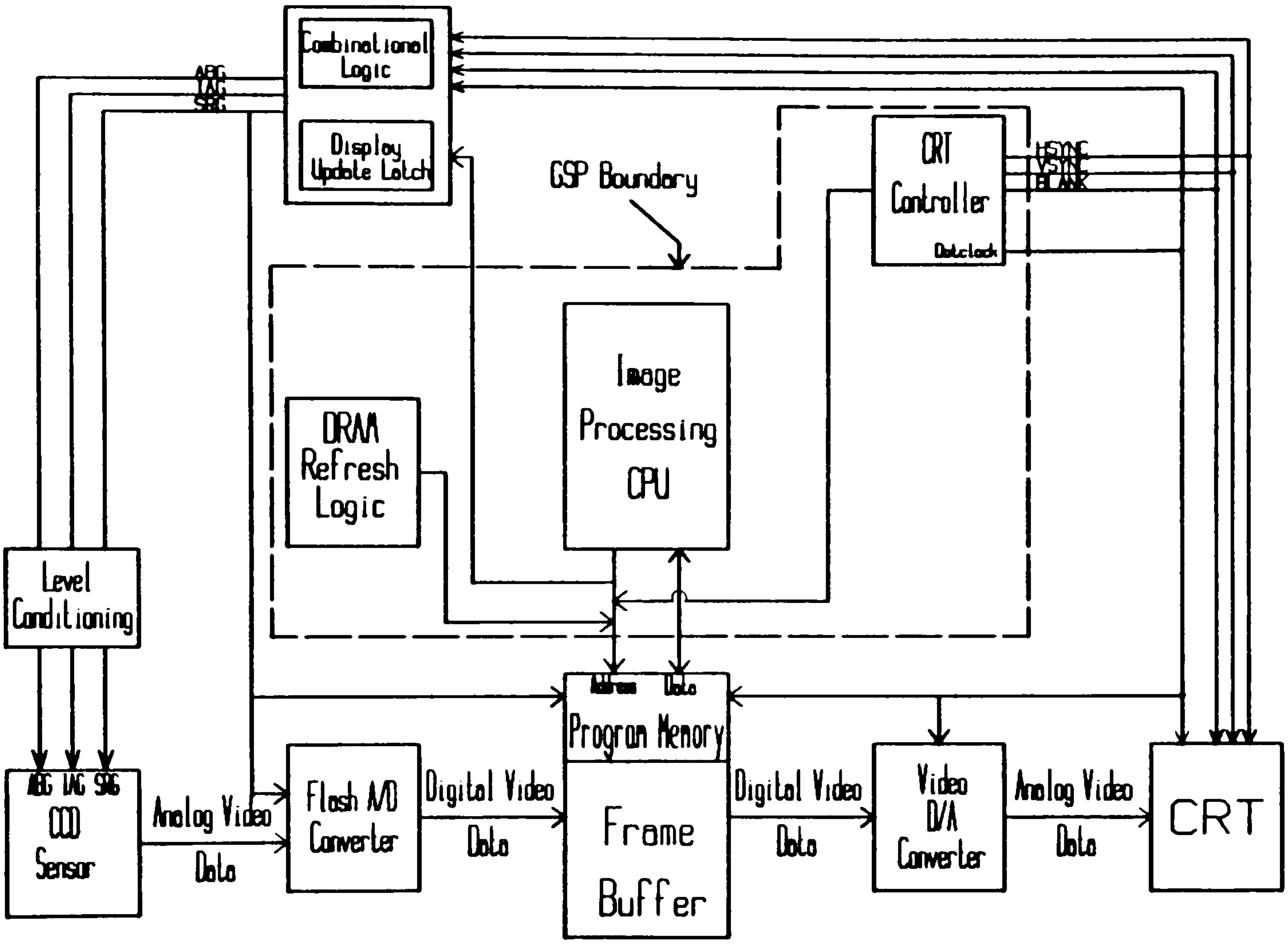


Figure 5.1 Imaging System Block Diagram

circuitry to provide the timing signals which drive the image sensor. Although this may result in an imaging system with a faster image capture rate, it does so at a considerable cost increase. Use of the GSP provides adequate performance for systems which do not require high resolution image capture at high speed. The GSP would thus be suitable to provide image capture and display for applications such as stand alone video-telephone systems.

Imaging System Overview

The TMS34010 software development board was chosen as the vehicle for implementing a prototype imaging system based on the GSP. Use of the SDB not only provides a ready made, PC based hardware platform, which eliminates the need to produce an entire GSP prototype system, but also enables the use of a wealth of software already written for the SDB. An overall block diagram illustrating the use of the SDB for an imaging system is shown in Figure 5.2.

The hardware required to interface the SDB to an image sensor is provided by the camera interface board described in this chapter. The main goal in developing the interface was to keep SDB modifications minimal; however, a few modifications to the SDB itself were necessary and are documented in SDB schematics found in Appendix A.

Image Sensor

The choice of input device for the imaging system was governed by the need for a small and rugged, low cost camera. Use of the TC211 solid state image sensor along with limited support circuitry enables the construction of such a camera. A camera which meets these requirements is part of the EDC-1000

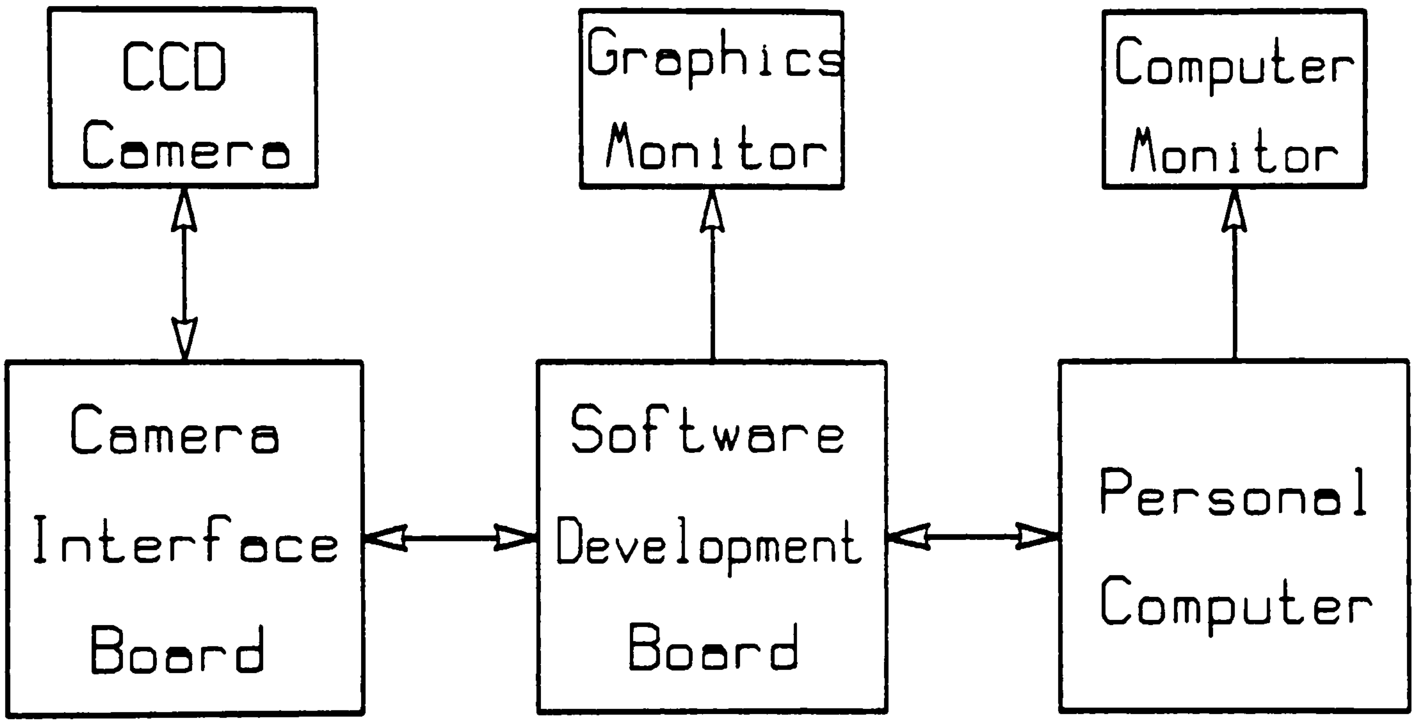


Figure 5.2 Overall System Block Diagram

computer camera system from Electrim Corporation, and was chosen as the input device for the imaging system. Although the EDC-1000 is a complete image capture system designed to connect to a PC, only the image sensor from the computer camera system was utilized for the imaging system described in this chapter. The camera portion of the system contains only a CCD sensor followed by a unity gain amplifier, and level conditioning circuitry to convert TTL level signals into CCD compatible clock signals. The camera contains no timing circuitry. A block diagram of the camera is provided in Figure 5.3.

Camera Interface Board

The camera interface board (CIB) is used to interface the computer camera to the TMS34010 software development board. A block diagram for the camera interface board is shown in Figure 5.4. Functions provided by the board include: digitizing the analog output from the computer camera, formatting of the digitized data for input to the SDB, and counting and multiplexing operations to provide the appropriate timing signals to the TMS4161EV4 VRAMs on the SDB.

Camera Interface Board Hardware Description

Part hardware present on the camera interface board enables data from the A/D converter to be properly routed into the frame buffer on the SDB. This hardware includes, an assortment of flip flops to provide counting functions, and a group of latches for storing A/D output data. One counter is present to slow down the SCLK signal from the SDB. Another two bit counter provides the clock signals for a group of latches which route nibbles of A/D output data into the frame buffer, so that data will be correctly displayed on the CRT.

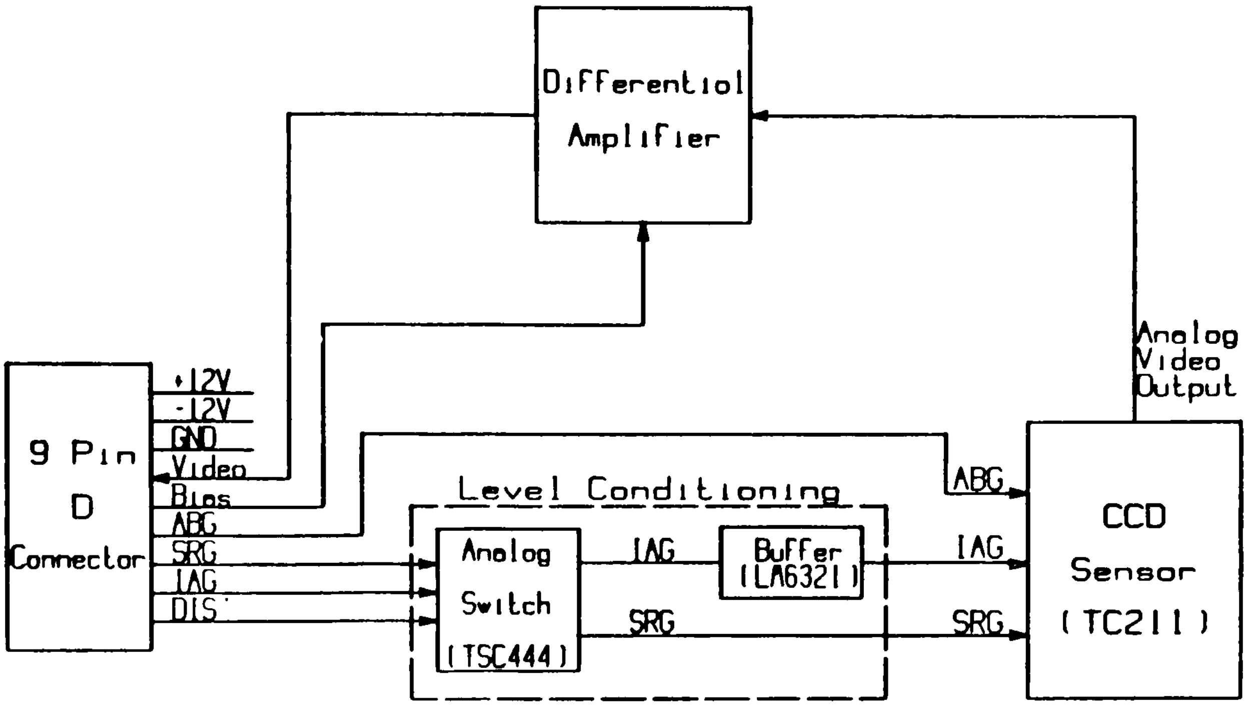


Figure 5.3 Computer Camera Block Diagram

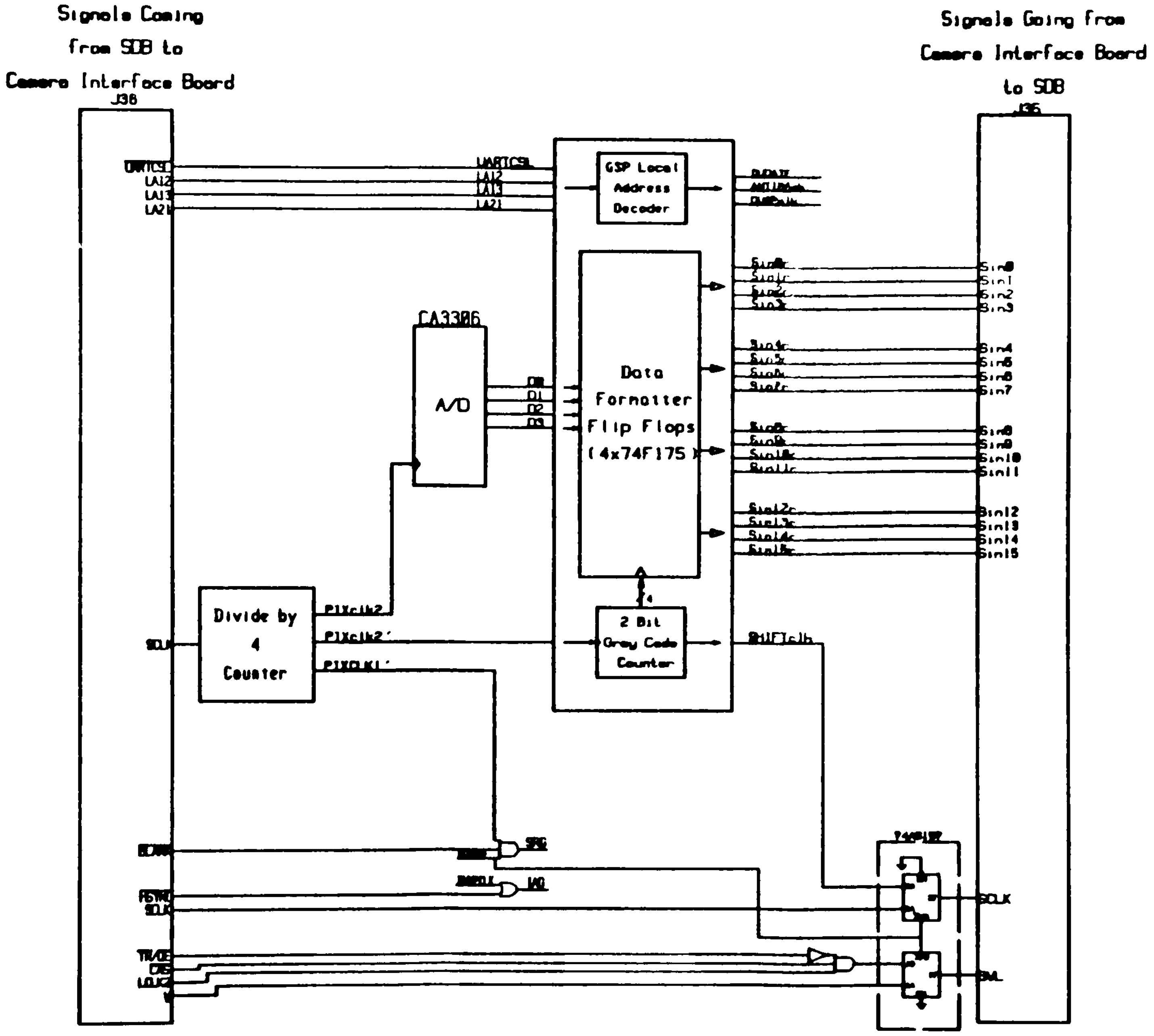


Figure 5.4 Camera Interface Board Block Diagram

Additional hardware present on the CIB is used to overcome the inability of the GSP to cause transfers from the VRAM internal shift register into the memory array. The typical mode of operation for the GSP is to produce memory-to-shift register transfers for screen refresh operations. External hardware is necessary to allow the GSP to transfer data from the shift register to the memory array, as is necessary when loading the memory from an external source such as a CCD. The external hardware consists of a display update direction latch which is used to distinguish between screen refresh and image acquisition cycles; a write signal generator which produces the write signal in place of the GSP during image acquisition cycles; and multiplexers which correctly route signals to the VRAMs depending on the type of cycle being executed.

Shift-Register-Transfer Conditioning Hardware

One of the biggest drawbacks of using the 34010 for image acquisition is the inability to change the direction of the VRAM shift register transfer during screen refresh operations. When generating the timing signals to drive an image sensor and load a VRAM with the digitized output from the sensor, it is necessary to have the ability to perform a VRAM shift-register to memory transfer cycle. If the 34010 is used to provide the timing signals to the image sensor through the use of the automatic screen refresh mode of operation, the user is confronted with the inability to change the direction of the display update. In other words, the 34010 forces a transfer from memory-to-shift register whenever a screen refresh operation is taking place.

In order to overcome this inherent limitation of the 34010, external hardware is necessary to allow either memory-to-shift register, or shift register-to-memory transfer. The external hardware required consists of a display update direction latch which indicates whether a screen refresh, or an image capture cycle is taking place; a write signal generator, which provides the appropriate write signal depending on the state of the display update direction latch; and multiplexers which allow the correct signals to be sent to the VRAMs based on the state of the display update direction latch.

Display Update Direction Latch

The schematic diagram for the display update direction latch is shown in Figure 5.5. The address decoding for the latch is provided in part by U11 on the SDB. U11 provides the local address bus decoding for the GSP. By just glancing at the SDB schematics shown in Appendix A, one might draw the conclusion that there are two available outputs left on PAL U11, which might be used for additional address decoding; however, examination of the PAL programming file shown in Figure 5.6 reveals that those two outputs are used for internal feedback registers. For this reason it is necessary to either have another device perform the address decoding, or use one of the select lines from U11. Because of the way the addresses are mapped on the SDB it is not possible to map the display update direction latch into one of the unused areas in the SDB memory map. For this reason the select line for the USART was used as to provide a portion of the address decoding for the display update direction latch.

The display update direction latch is mapped into the space from 0210 0000h to 0210 3000h on the SDB. An access to location 0210 2000h causes the display update latch to be reset, so that screen refresh operations take place; an access to location 0220 3000h sets the latch for an image acquisition cycle.

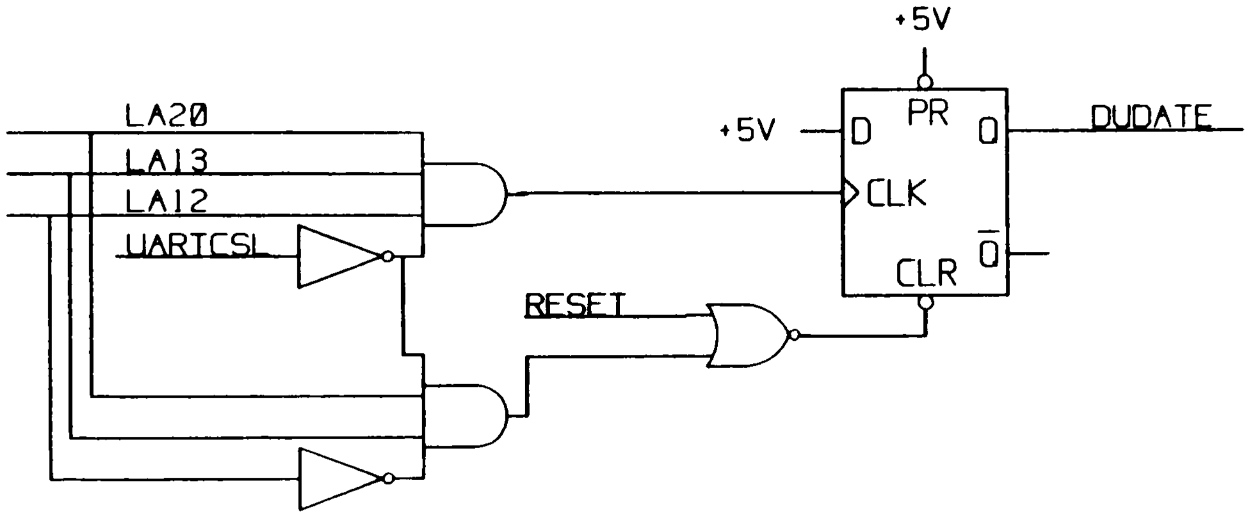
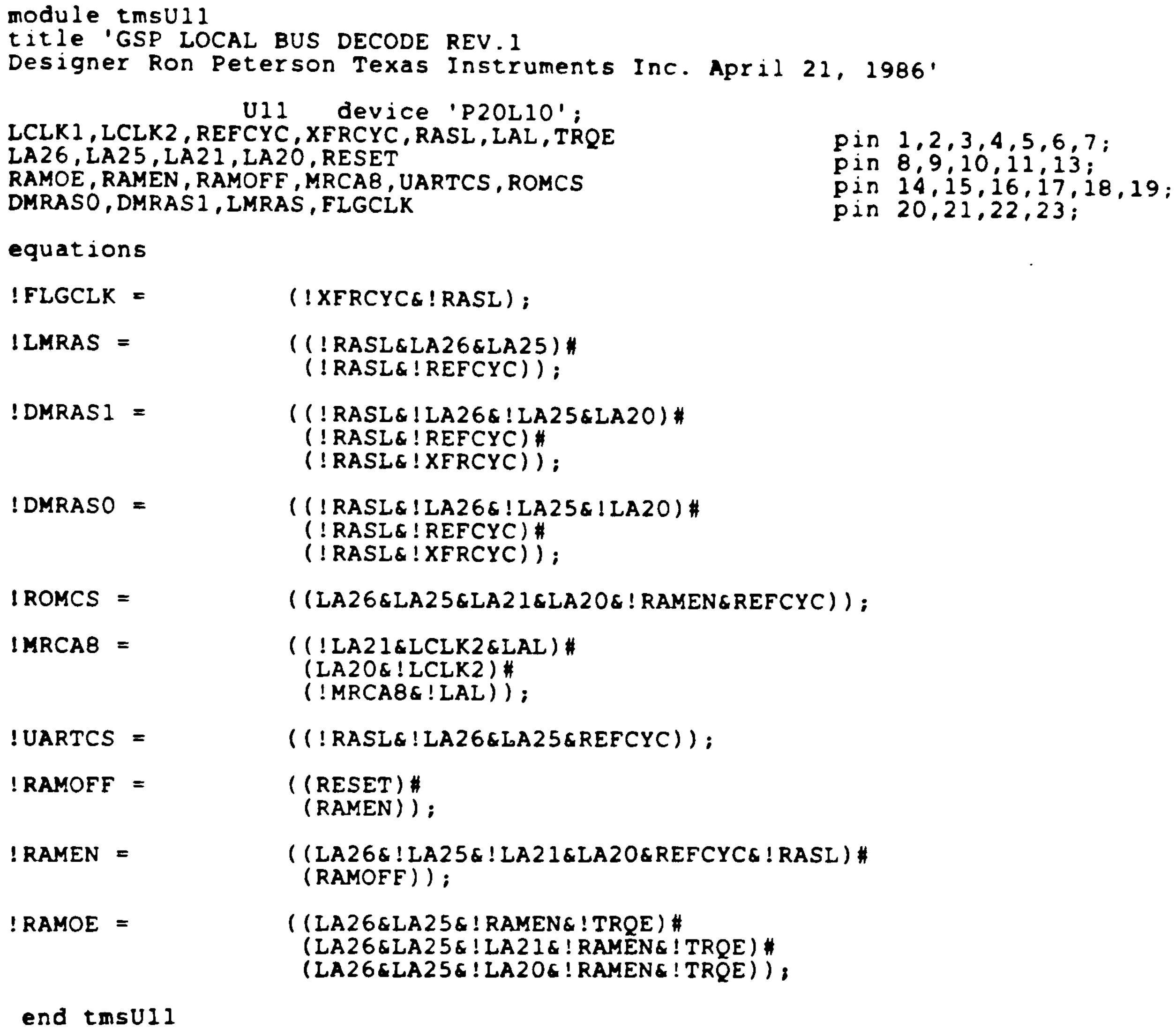


Figure 5.5 Display Update Direction Latch Schematic



```
module tmsU11
title 'GSP LOCAL BUS DECODE REV.1
Designer Ron Peterson Texas Instruments Inc. April 21, 1986'

                U11   device 'P20L10';
LCLK1,LCLK2,REFCYC,XFRCYC,RASL,LAL,TRQE              pin 1,2,3,4,5,6,7;
LA26,LA25,LA21,LA20,RESET                            pin 8,9,10,11,13;
RAMOE,RAMEN,RAMOFF,MRCA8,UARTCS,ROMCS                pin 14,15,16,17,18,19;
DMRAS0,DMRAS1,LMRAS,FLGCLK                           pin 20,21,22,23;

equations

!FLGCLK =         (!XFRCYC&!RASL);

!LMRAS =          ((!RASL&LA26&LA25)#
                   (!RASL&!REFCYC));

!DMRAS1 =         ((!RASL&!LA26&!LA25&LA20)#
                   (!RASL&!REFCYC)#
                   (!RASL&!XFRCYC));

!DMRAS0 =         ((!RASL&!LA26&!LA25&!LA20)#
                   (!RASL&!REFCYC)#
                   (!RASL&!XFRCYC));

!ROMCS =          ((LA26&LA25&LA21&LA20&!RAMEN&REFCYC));

!MRCA8 =          ((!LA21&LCLK2&LAL)#
                   (LA20&!LCLK2)#
                   (!MRCA8&!LAL));

!UARTCS =         ((!RASL&!LA26&LA25&REFCYC));

!RAMOFF =         ((RESET)#
                   (RAMEN));

!RAMEN =          ((LA26&!LA25&!LA21&LA20&REFCYC&!RASL)#
                   (RAMOFF));

!RAMOE =          ((LA26&LA25&!RAMEN&!TRQE)#
                   (LA26&LA25&!LA21&!RAMEN&!TRQE)#
                   (LA26&LA25&!LA20&!RAMEN&!TRQE));

 end tmsU11
```

Figure 5.6 Local Bus Decode PAL Programming File

An access to these locations corresponds to a write access to the USART. An access to location 0210 2000h serves as a write to mode registers 1 and 2 of the USART, while an access to 0210 3000h serves as a write to the USART command register. As long as the USART is not being used, there will be no adverse affect from writing to these two registers, and one can set or reset the display update latch without harm to the USART or the SDB.

Write Signal Generator

The display update direction latch is used to affect the direction of VRAM shift register transfers. The means by which this is accomplished is through the addition of a write signal generator which produces the write signal for the SDB whenever the display update direction latch is set for an image acquisition cycle. When an image is captured from the image sensor, data must be digitized, transferred into the VRAM's internal shift register and then transferred, one row at a time, into the VRAM memory array. The timing signals generated by the GSP during memory-to-shift register and shift register-to-memory transfer cycles are shown in figures 5.7 and 5.8, respectively. Figure 5.9 shows the relevant set-up and hold times which must be adhered to.

The signal which distinguishes between the two types of transfer cycles is the write signal. If the write signal is not asserted then a memory-to-shift register transfer cycle takes place, as shown in Figure 5.7. If write is asserted before RAS goes low, then a shift register-to-memory transfer cycle will occur, as shown in Figure 5.8.

memory-to-register cycle timing

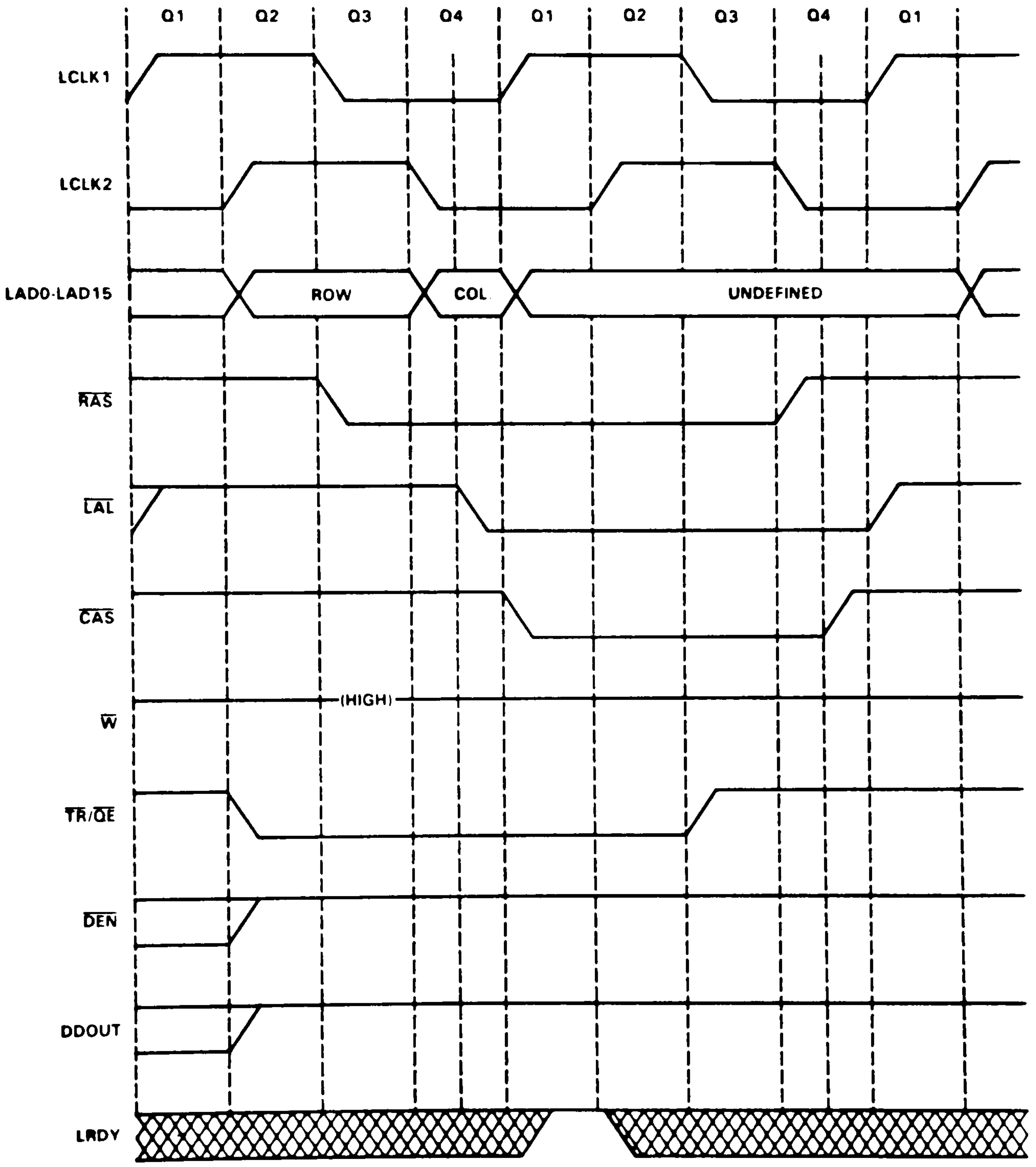


Fig 5.7 Memory-to-Shift Register Transfer

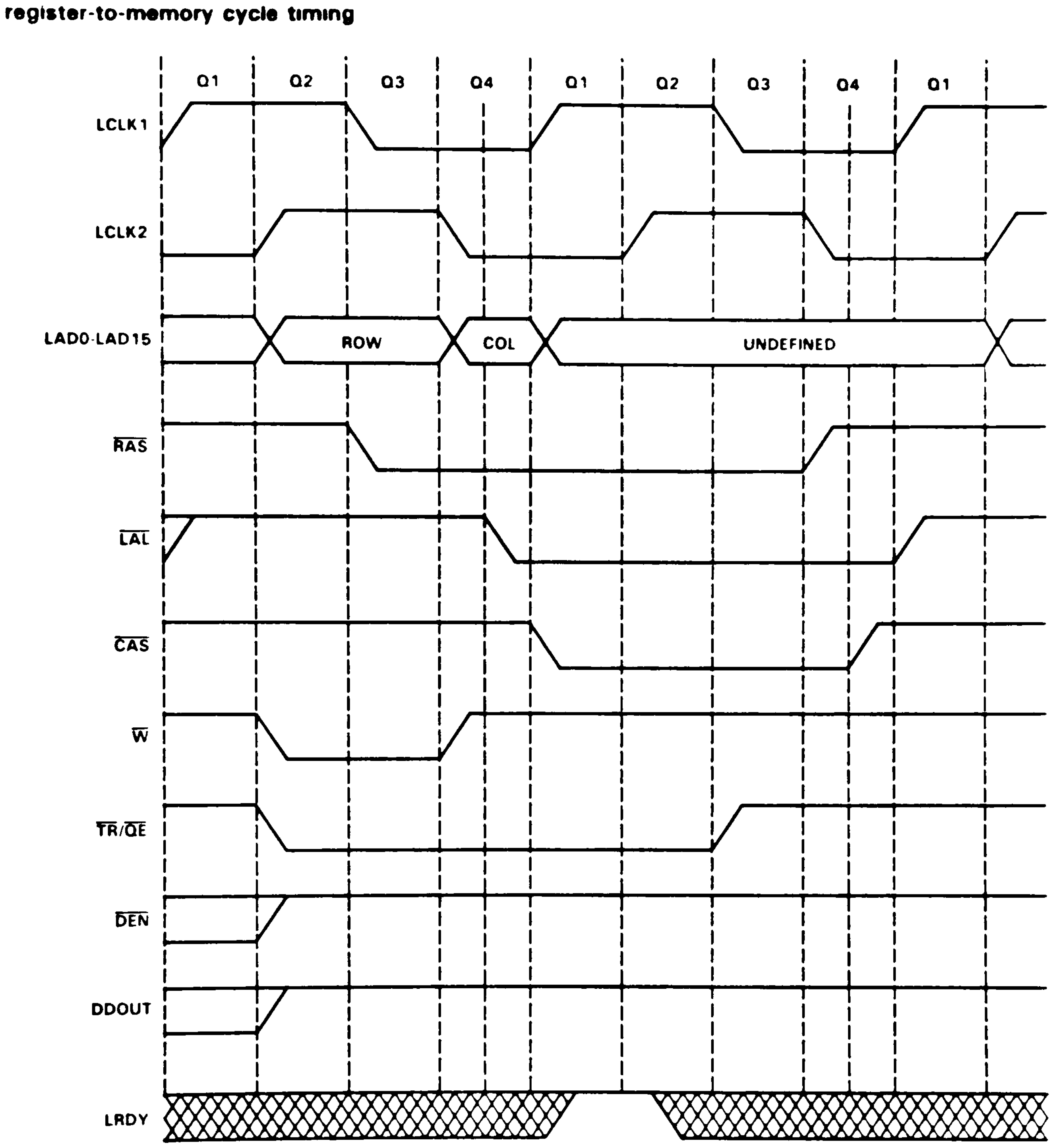


Figure 5.8 Shift Register-to-memory Transfer

Quarter clock time t_Q, which appears in the following table, is one quarter of a local output clock period or $2t_{c(ICK)}$.

NO	PARAMETER		TMS34010-40		TMS34010-50 TMS34010-60		UNIT
			MIN	MAX	MIN	MAX	
74	$t_{su(WL-RL)}$	Setup time of $\overline{W}$ low to $\overline{RAS}\downarrow$, shift register transfer cycle	$t_Q - 20$		$t_Q - 10$		ns
75	$t_{h(RL-WL)}$	Hold time of $\overline{W}$ low after $\overline{RAS}$ low, shift register transfer cycle	$t_Q - 20$		$t_Q - 10$		ns
76	$t_{su(TRL-RL)}$	Setup time of $\overline{TR}/\overline{OE}$ low to $\overline{RAS}\downarrow$, shift register transfer cycle	$t_Q - 20$		$t_Q - 10$		ns
77	$t_{h(RL-TRL)}$	Hold time of $\overline{TR}/\overline{OE}$ low after $\overline{RAS}$ low, shift register transfer cycle	$4t_Q - 20$		$4t_Q - 10$		ns
78	$t_{h(CL-TRL)}$	Hold time of $\overline{TR}/\overline{OE}$ low after $\overline{CAS}$ low, shift register transfer cycle	$2t_Q - 20$		$2t_Q - 10$		ns
79	$t_{su(TRH-RH)}$	Setup time of $\overline{TR}/\overline{OE}$ high to $\overline{RAS}\uparrow$, shift register transfer cycle	$t_Q - 20$		$t_Q - 10$		ns
80	$t_{su(TRH-CH)}$	Setup time of $\overline{TR}/\overline{OE}$ high to $\overline{CAS}\uparrow$, shift register transfer cycle	$1.5t_Q - 25$		$1.5t_Q - 10$		ns

NOTES 1. Advance information notices apply only to the TMS34010-60.
2. Parameters 81 and 82 intentionally omitted

local bus timing parameters: shift register transfer cycle

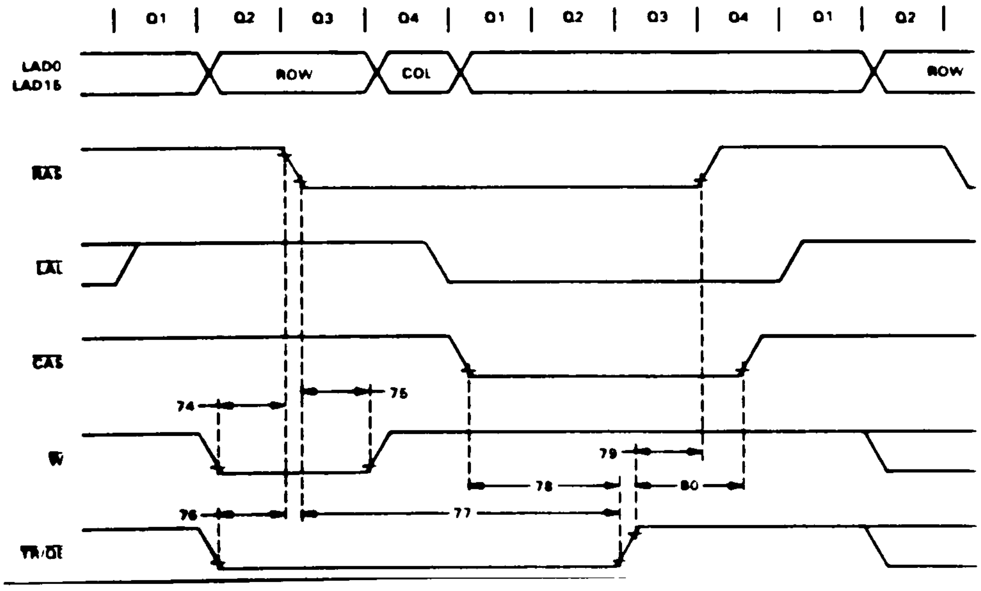


Figure 5.9 Shift Register Timing

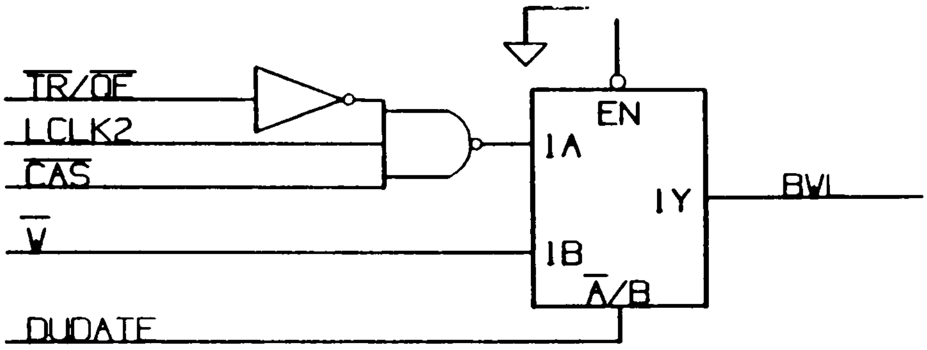
The WRITE'signal output by the write cycle generator is formed by NANDing CAS', !TR'/QE', and LCLK2 together. The result is shown in Figure 5.10, along with relevant propagation delays. CAS' will always be high when a shift register transfer (indicated by the high to low transition of TR'/QE') is taking place. The WRITE' signal should be asserted for one half of a local clock cycle (1/2 of LCLK2), and must meet the set-up and hold times shown in Figure 5.9.

Write Cycle Multiplexer

The output from the write cycle generator, and the W/ signal output from the 34010, serve as the inputs to a quad two input multiplexer which resides on the camera interface board. The output of the multiplexer depends on the state of the display update direction latch. If the latch is in the reset state, the output of the mux is the W/ output from the 34010, which enables memory-to-shift register transfers for updating the CRT. If the latch is set, the output of the mux is the WRITE/ signal from the write cycle generator; thus enabling shift register-to-memory transfer cycles.

CCD to VRAM Interface

A group of latches attached to the output of the A/D converter provide the majority of the interface circuitry between the SDB and the EDC-1000 computer camera. The interface includes four quad D flip-flops, a 2-bit gray code counter, and the A/D converter. Four bits of the 6-bit flash A/D converter are tied directly to the inputs of four quad latches. The gray code counter is used to count groups of four pixels, so that the four nibbles of video data can be loaded--in packed pixel format--into a 16-bit wide VRAM bank.



Propogation Delays:
74AS04 tplh=5ns
tphl=4ns
74AS10 tplh=4.5ns
tphl=4.5ns
74AS157 (A or B to Y)
tplh=7.5ns
tphl=6.5ns

Worst Case Delay = 17ns

Figure 5.10 Write Cycle Generator

A/D Converter

A 6-bit flash A/D converter from RCA is used to digitize the analog CCD output data. The converter chosen is the CA3306. Constructed using CMOS technology, the 3306 features: a single supply voltage (5V), 6-bit latched 3-state output with overflow, built in internal voltage reference with an optional external reference, and a 66 ns conversion time. Although only four bits of digital output are utilized in the current design, the part was chosen because of availability, cost and a limited selection of 4-bit devices.

Data Formatter

The purpose of the data formatter is to properly format the pixel data before it is sent to the serial inputs of the VRAMs. This is necessary due to the way in which the frame buffer is constructed on the SDB. Referring to the SDB schematics, one can see that there are two layers of multiplexing circuitry between the VRAM array and the CRT. The first multiplexer resides within the 34070 video palette. This device incorporates a 2-to-1 multiplexer which enables the VRAM shift register to be clocked at a rate one half the speed of the actual dot clock to the CRT. The second multiplexer is used to feed the inputs to the 34070. This second layer of multiplexing circuitry allows the VRAM shift register to be clocked at a rate four times slower than the dot clock of the CRT. For the SDB, driving a Multisync monitor with a dot clock of 25MHz, the VRAM shift registers are thus shifted at 6.25MHz.

During a screen refresh operation, pixel data from a bank of VRAMs is provided to the monitor one nibble at a time from each of the four VRAMs in that bank; data must therefore be loaded into the bank of VRAM in a like manner.

This is the function of the data formatter. When clocked by the outputs of the gray code counter, the flip-flops enable the output of the A/D converter to be input to the least significant nibble of VRAM array for the first pixel; the second pixel is placed into the next nibble, and so on, with the fourth pixel placed into the most significant nibble of the sixteen bit wide VRAM bank. All of the pixel data is shifted into the VRAM array in packed pixel format until all pixels in the CCD have been output by the A/D converter.

VRAM Shift-clock Multiplexer

A multiplexer is used to provide the appropriate clock signal to the VRAM shift register based on the state of the display update direction latch. If the latch is in the reset state, the shift clock signal is provided by the SDB (the SCLK signal). If the latch is in the set state, the output of the multiplexer is provided by the SHIFTclk output from the gray code counter. Use of the multiplexer ensures that the appropriate shift clock will be provided to the VRAMs depending on the mode of operation for the SDB.

CCD Interface Timing

A number of clocks are produced on the camera interface board, and are used to drive clock inputs on the camera and on the SDB. These clocks are used for shifting data out of the CCD and into the VRAM internal shift registers.

Data Formatter Timing

A timing diagram for the data formatter is shown in Figure 5.11. VCLK is the signal from the SDB which, after being combined with the blanking signal to

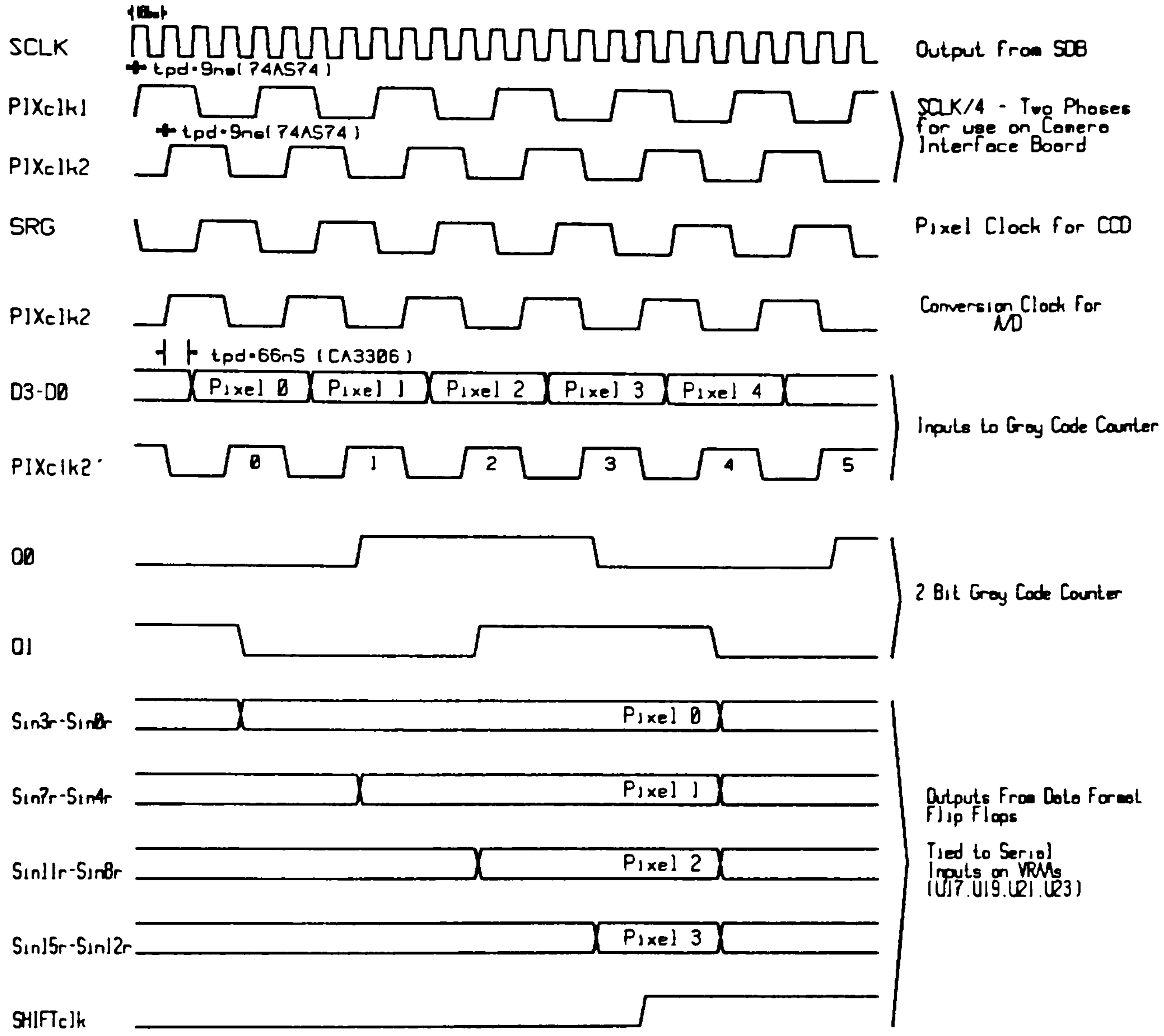


Figure 5.11 Data Formatter Timing Diagram

produce SCLK, is used to drive the shift clock on the VRAM shift register during screen refresh operations. SCLK is divided by four on the camera interface board to produce PIXclk1 and PIXclk2. PIXclk1' is ANDed with DUDATE and BLANK' to produce the SRG signal that drives the CCD camera. Pixels are shifted out of the CCD sensor on the falling edge of the SRG pulse. PIXclk2 is used to drive the conversion clock on the A/D converter. PIXclk2' drives a 2-bit gray code counter which counts the four nibbles that make up the sixteen bit wide VRAM bank.

A pixel is digitized with each rising edge of PIXclk2, and that pixel is then transferred to the outputs of the D flip-flops with the rising edge of PIXclk2'. On the first rising edge of PIXclk2', the outputs of the A/D are transferred to flip-flop outputs Sin3r-Sin0r. On the second rising edge of PIXclk2', the outputs of the A/D are transferred to flip-flop outputs Sin7r-Sin4r. On the third rising edge of PIXclk2' the outputs of the A/D are transferred to flip-flop outputs Sin11r-Sin8r. The fourth rising edge of PIXclk2' transfers the outputs of the A/D to flip-flop outputs Sin15r-Sin12r, and the falling edge of PIXclk2' causes the data to be strobed into the serial inputs on the VRAMs. After a 16-bit word has been strobed into the VRAM bank, the process begins again, and continues until an entire row of CCD data has been shifted out of the TC211. Once an entire row has been completed, the 34010 causes a shift register-to-memory transfer cycle to occur; thus shifting a row of CCD data into the VRAM bank.

CCD Timing Signals

Besides the display update direction latch, two additional signals are produced by decoding the GSP local address bus. These two signals--DUMPclk, and ANTIBMck--serve as clock signals for the TC211. DUMPclk provides a clock to the IAG pin on the image sensor just prior to image acquisition. This signal is used to clear the sensor of unwanted charge leftover from the previous image. ANTIBMck provides antiblooming pulses to the sensor while an image is being acquired.

DUMPclk is produced by decoding a GSP access to location 0210 0000h. Whenever an access to this location occurs, a pulse is produced by the local address decoder; this pulse is then provided to the IAG input on the TC211. Each pulse to the IAG input causes an automatic fast clear of the serial register before the next row is transferred in; therefore, in order to clear the entire array, the IAG pin is pulsed once for every line in the sensor, or a total of 165 times. In order to clear the sensor, a GSP program is used to loop on address location 0210 0000h, until the array has been cleared. Notice that the camera interface board schematics show the DUMPclk signal ORed with the HSYNC' signal to produce the output to the image sensor, so that DUMPclk produces the IAG pulse before an image is acquired, and HSYNC' produces the IAG pulse while an image is being acquired.

ANTIBMck is produced by decoding a GSP access to location 0210 0001h. Whenever an access to this location occurs, a pulse is produced by the local address decoder; this pulse is then provided to the antiblooming input on the CCD sensor. The antiblooming signal prevents the charge of overexposed elements from spilling into neighboring elements. The antiblooming signal is pulsed at a rate of approximately 1 MHz during the time that light is incident on the sensor. While testing the imaging system, the antiblooming circuit, although present on the

CIB, was not used. The antiblooming input to the image sensor was instead tied to ground.

In addition to the VCLK signal from the SDB, the HSYNC', VSYNC' and BLANK' signals from the 34010 are monitored by the camera interface board. These signals are used to produce the IAG, and SRG signals for the TC211 image sensor. The rising edge of the HSYNC' is used to provide the IAG clock to the TC211 while the display update latch is set. SRG is output to the CCD, and serves as the clock to shift pixel data out of the CCD, as long as the BLANK' signal is inactive, and the display update latch is set for image acquisition.

Software Description

The software required to run the camera interface board is quite straightforward. A flowchart for the program is shown in Figure 5.12, and the program listing is provided in Figure 5.13. The first task to be accomplished by the program is that of providing the proper addresses for the GSP and CIB registers to be addressed. This is the first portion of the program and is accomplished using the ".set" assembler directive.

The next step in image acquisition is to disable screen refresh so that the video timing registers can be reprogrammed with the values to be used during image capture. Screen refresh is disabled by masking a bit in a GSP control register. After the timing registers have been programmed, a frame dump is necessary to clear the image sensor before the image is actually captured. This is accomplished by repeatedly writing to the address for DUMPclk on the CIB. The frequency of the DUMPclk signal is determined by the length of the DUMP subroutine.

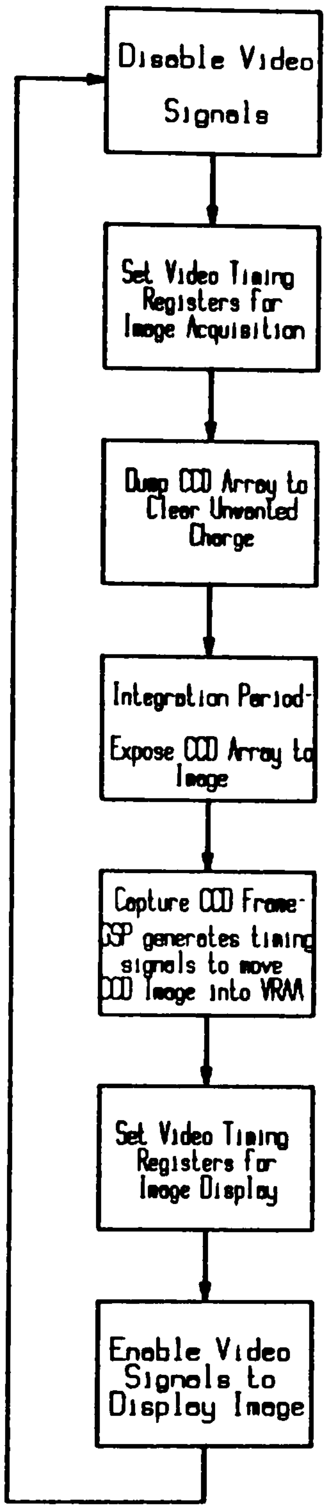

Disable Video Signals
Set Video Timing Registers for Image Acquisition
Dump CCD Array to Clear Unwanted Charge
Integration Period- Expose CCD Array to Image
Capture CCD Frame- GSP generates timing signals to move CCD Image into VRAM
Set Video Timing Registers for Image Display
Enable Video Signals to Display Image

Figure 5.12 Flowchart for the Image Capture Program

```
            .file   "camctl.asm"
;       EE 4333 FALL 89. REMOTE IMAGING SYSTEM
;       Kelly, Van Meter,
;       MAIN CONTROL PROGRAM
;       copyright John Van Meter 1989
CAMERA    nop
          .even
DISPC     .set     0C0000080h    ;CONTROL REG. ADDRESS
MASKA     .set     0FFBFFFFFh    ;DISABLE EX. INTERRUPTS
MASKB     .set     00200000h     ;ENABLE EXTERNAL INTERRUPTS
MASKC     .set     07410h        ;DISABLE VIDEO SIGNALS
MASKD     .set     0F410h        ;ENABLE VIDEO SIGNALS
HESYNC    .set     0C0000000h    ;ADDRESS FOR HOR.SYNC
HEBLNK    .set     0C0000010h    ;ADDRESS FOR END HOR.BLANK
HSBLNK    .set     0C0000020h    ;ADDRESS FOR START HORZ.BLANK
HTOTAL    .set     0C0000030h    ;ADDRESS FOR TOTAL HORZ.COUNTS
VESYNC    .set     0C0000040h    ;ADDRESS FOR END VERT SYNC
VEBLNK    .set     0C0000050h    ;ADDRESS FOR END BLANK
VSBLNK    .set     0C0000060h    ;ADDRESS FOR VERT END BLANK
VTOTAL    .set     0C0000070h    ;ADDRESS FOR VERT TOTAL COUNTS
DUDATES   .set     02103000h     ;ADDRESS FOR DUDATE SET
DUDATERS  .set     02102000h     ;ADDRESS FOR DUDATE RESET
          .text
          MOVI     00000004h,A9  ;SET PIXEL SIZE
          MOVE     A9,@0C0000150h;  "  "    "
          MOVI     014h, A14     ;VALUE FOR INTEGRATION TIME LOOP
          MOVI     0FFFFFh,A13   ;VALUE FOR DISPLAY INTERRUPT LIMIT
;-------DISABLE VIDEO SIGNALS---------------------------------
        CALLA DISV
;-------SET VIDEO TIMING REGISTERS FRAME----------------------
DOT       CALLA SVTRF
;-------DO A FRAME DUMP---------------------------------------
          CALLA FRDUMP
;-------INTEGRATION PERIOD------------------------------------
          CALLA SHOT
;-------GET FRAME INTO VRAMS----------------------------------
          CALLA FRAME
;-------SET VIDEO TIMING FOR DISPLAY
          CALLA SVTRD
;-------ENABLE VIDEO------------------------------------------
          CALLA ENV
;-------LOOP BACK TO START------------------------------------
          JAUC DOT
;
;
;
```

Figure 5.13 Program Listing for the Image Capture Program

```
 ----DO A FRAME DUMP SUBROUTINE---------------------

;
FRDUMP    MOVI    0A6h,A1         ;NUMBER OF IAG PULSES IS 166
DUMP
          MOVE    @02100000h,A0 ;PULSE DUMP CLOCK
              NOP               ;DELAY
              NOP               ;DELAY
              NOP               ;DELAY
              DSJ  A1,DUMP      ;END OF DELAY LOOP
          RETS
 ;
;
;-----------------INTEGRATION PERIOD SUBROUTINE------
SHOT      MOVE  A14,A0     ;DELAY PERIOD
INTEGR        NOP            ;ADJUST THE # OF NOPS TO SET TIME
              DSJ  A0,INTEGR ;END OF EXPOSURE LOOP
          RETS
;
 ;
;
;------SET TIMING REGISTERS FOR DISPLAY SUBROUTINE--------
SVTRD     MOVI    001Ch,A0
          MOVE    A0,@HESYNC
          MOVI    001Ah,A0
          MOVE    A0,@HEBLNK
          MOVI    00CAh,A0
          MOVE    A0,@HSBLNK
          MOVI    00CEh,A0
          MOVE    A0,@HTOTAL
          MOVI    0003H, A0
          MOVE    A0,@VESYNC
          MOVI    001BH,A0
          MOVE    A0,@VEBLNK
          MOVI    01FBH,A0
          MOVE    A0,@VSBLNK
          MOVI    01FDH,A0
          MOVE    A0,@VTOTAL
          MOVI   00000000h,A0
          MOVE    A0,@0C0000090h ;SET DISPLAY START ADDRESS
;set video pallett
        MOVI 0010h,A4         ;NUMBER OF LOOPS
        MOVI 0000h,A5         ;A5 HAS PALLETT VALUES
        MOVI 00000000h,A6     ;A6 HAS THE ADDRESS
LOOP    MOVE A5,*A6           ;MOVE ONE REGISTER TO MEMORY AT A6
        ADDI 01110h,A5    ;INCREMENT EACH NIBBLE BY ONE
        ADDI 0010h,A6         ;INCREMENT ADDRESS BY 16
        DSJ A4,LOOP
        RETS
 ;
;
```

Figure 5.13 Continued

```
;-----SET TIMING REG. FRAME SUBROUTINE-------
SVTRF     MOVI    0001h,A0      ; HESYNC = 1
          MOVE    A0,@HESYNC
          MOVI    0002h,A0      ; HEBLNK = 2
          MOVE    A0,@HEBLNK
          MOVI    0348h,A0      ; HSBLNK = 840
          MOVE    A0,@HSBLNK
          MOVI    0350h,A0      ; HTOTAL = 842
          MOVE    A0,@HTOTAL
          MOVI    0001h,A0      ; VESYNC = 1
          MOVE    A0,@VESYNC
          MOVI    0002h,A0      ; VEBLNK = 2
          MOVE    A0,@VEBLNK
          MOVI    00A8h,A0      ; VSBLNK = 168
          MOVE    A0,@VSBLNK
          MOVI    00A9h,A0      ; VTOTAL = 169
          MOVE    A0,@VTOTAL
          MOVI    00000100h,A0
          MOVE    A0,@0C0000090h ;SET DISPLAY START ADDRESS
          RETS
 ;
;
;-----------------DISABLE VIDEO SUBROUTINE----------
DISV      MOVI MASKC,A1
          MOVE A1,@DISPC
          RETS
 ;
;
; ----------------ENABLE VIDEO SUBROUTINE------------
ENV       MOVI    MASKD,A1
          MOVE    A1,@DISPC
          RETS
 ;
;
;----------------------FRAME SUBROUTINE--------------
FRAME     CLR A10
          CLR A12
          MOVE  @DUDATES,A0     ;SET DUDATE
          MOVE A13,A0        ;TIME LIMIT FOR DISPLAY INTERRUPT
          CALLA ENV             ;ENABLE VIDEO
WAIT     NOP
             DSJ A0,WAIT        ;END WAIT LOOP UNCONDITIONAL
OVER      MOVE @DUDATERS,A11
          CALLA DISV            ;DISAVBLE VIDEO
          RETS                  ;RETURN FROM SUBROUTINE
;
;
ENDOF     nop
          NOP
          TRAP 10
          .end
```

Figure 5.13 Continued

Having cleared the image sensor of unwanted charge, the integration period can begin. The length of the integration period is determined by the length of the SHOT subroutine. Once the SHOT subroutine has completed, it is necessary to move the image out of the image sensor and into the VRAM array. This is accomplished by the FRAME subroutine. Here the DUDATE latch is set for image capture, so that timing signals from the GSP are routed to the image sensor instead of the multisync monitor. When all 165 lines have been read out of the image sensor, the DUDATE latch is reset, and the GSP timing registers are restored to the correct values for driving the multisync monitor.

The final step in the program is to re-enable the video timing signals and display the acquired image on the CRT. With the process of image acquisition complete, a new frame of image data may be read out of the image sensor.

The result of executing the program is demonstrated in the following chapter. Several test images were utilized to demonstrate the ability of the imaging system to capture and display images.

CHAPTER VI
RESULTS

Evaluation of the results for a hardware oriented project, such as that discussed in Chapter 5, is not particularly difficult; the project either works, or it does not work. Much to the delight of the author, the proposal to develop an image capture and display system based on the GSP, was successfully implemented and tested, as evidenced by the photographs in the remainder of this chapter.

Figures 6.1 through 6.7 demonstrate the results obtained utilizing the hardware described in Chapter 5. Figure 6.1 is a photograph of the entire hardware setup, with the CCD camera attached to the camera interface board, and focused on a test image. The monitor in the upper left corner of the photograph displays the image captured by the CCD camera. Figure 6.2 is a photograph illustrating the CCD camera's view of the test image which provides information on two aspects of the performance of the image capture system, resolution and the ability to distinguish between different gray levels. Figure 6.3 is a photograph of the monitor while displaying the image captured by the CCD camera. Note that the image is reproduced several times on the monitor, demonstrating the ability to show several images at once because the spatial sampling provided by the TC211 image sensor is considerably less than the available resolution provided by the graphic display system on the SDB. Figure 6.4 is a photograph of the monitor, while displaying a reflection density guide as captured by the CCD camera.

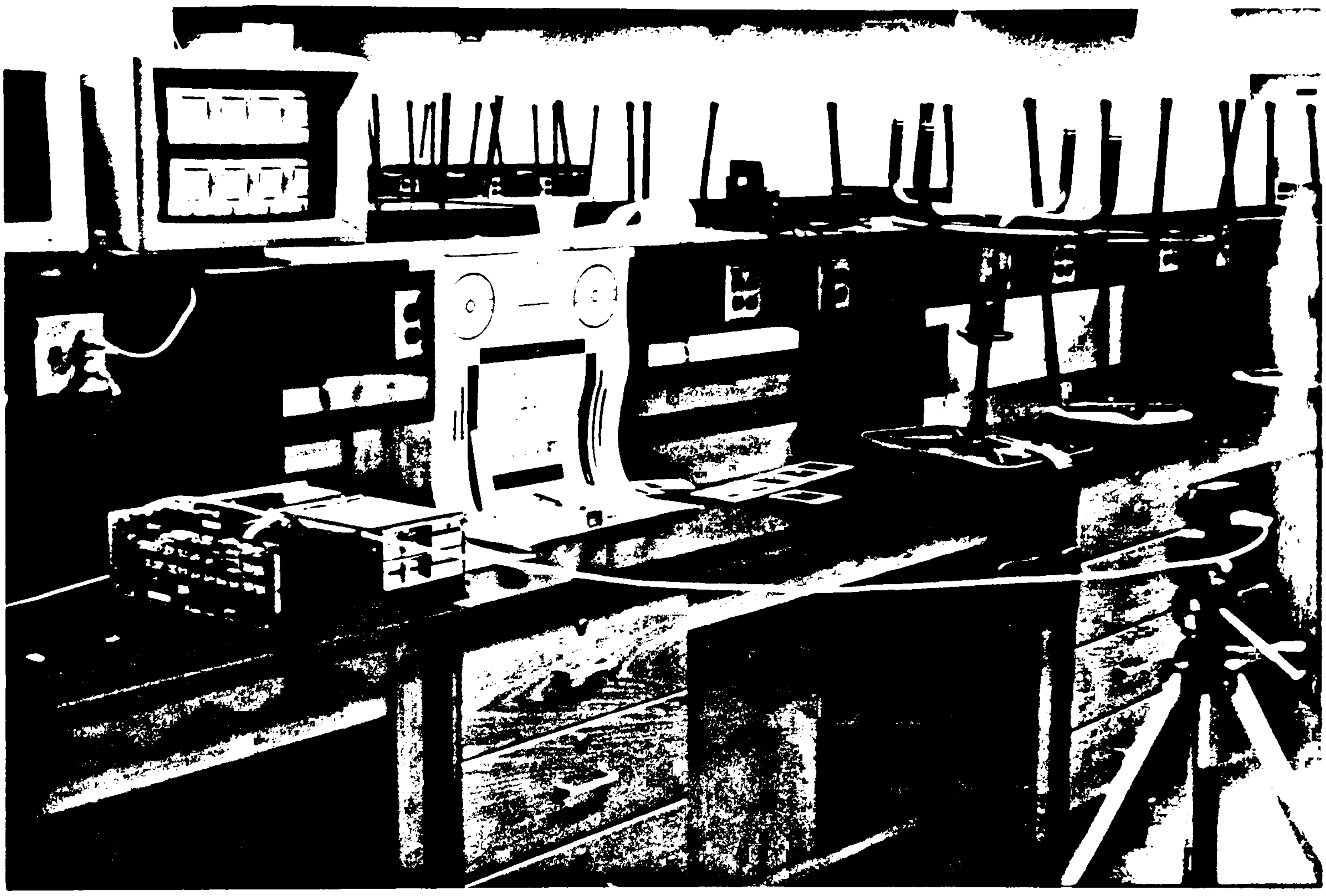


Figure 6.1

Photograph of the Overall Hardware Setup

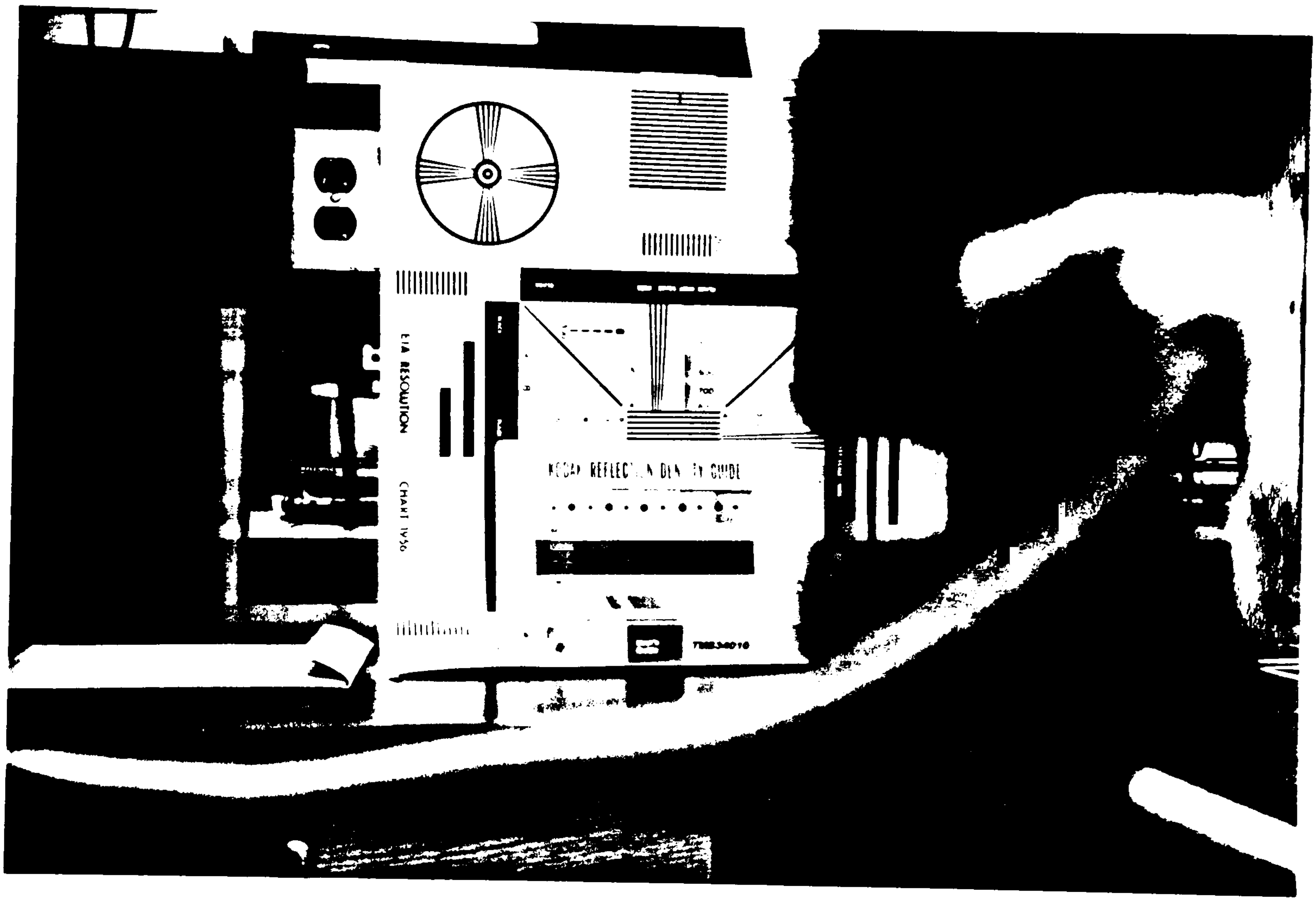


Figure 6.2

CCD Camera's View of the Test Image

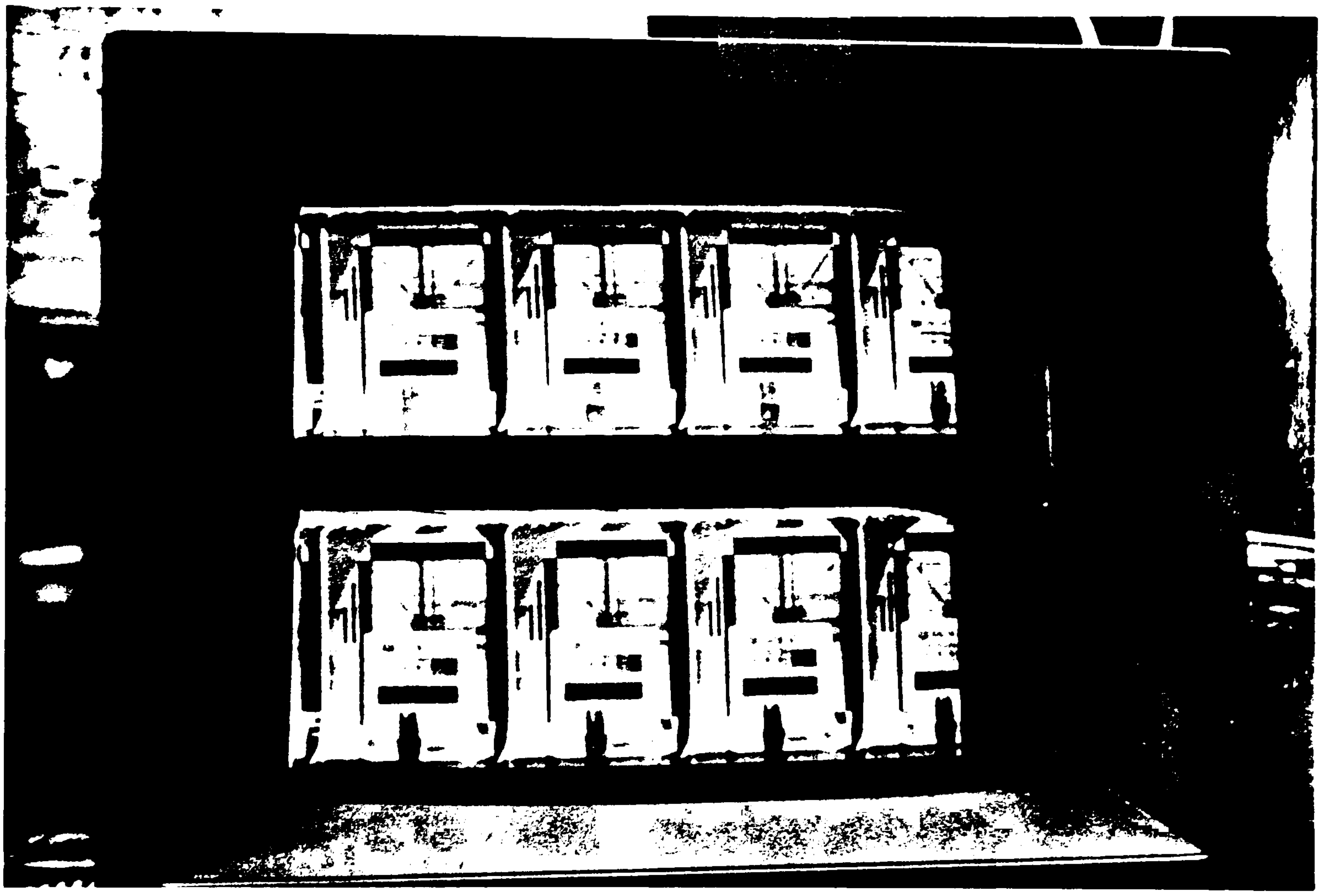


Figure 6.3

Test Image Captured with the CCD Camera

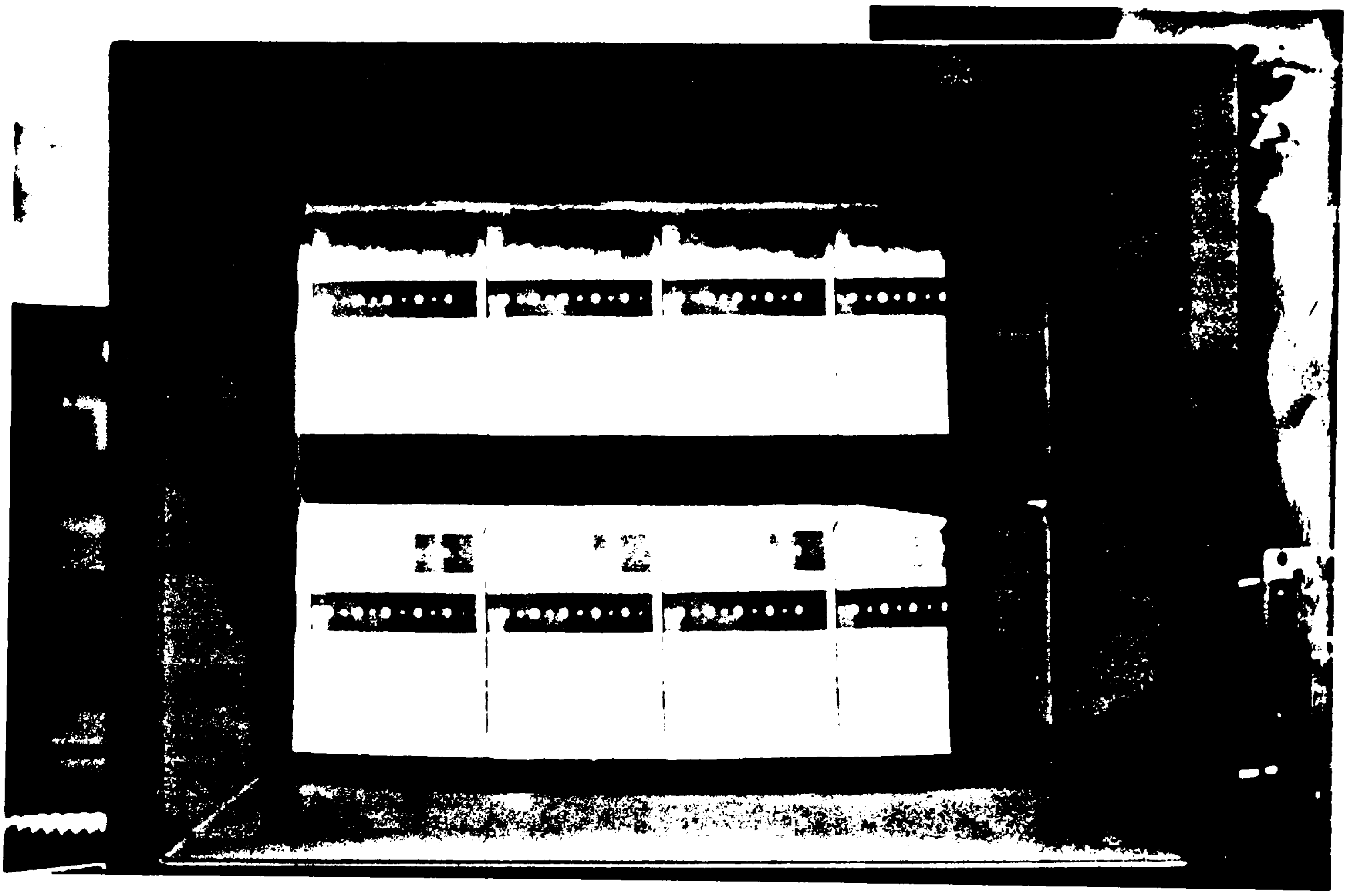


Figure 6.4

Reflection Density Guide Test Image Captured with the CCD Camera

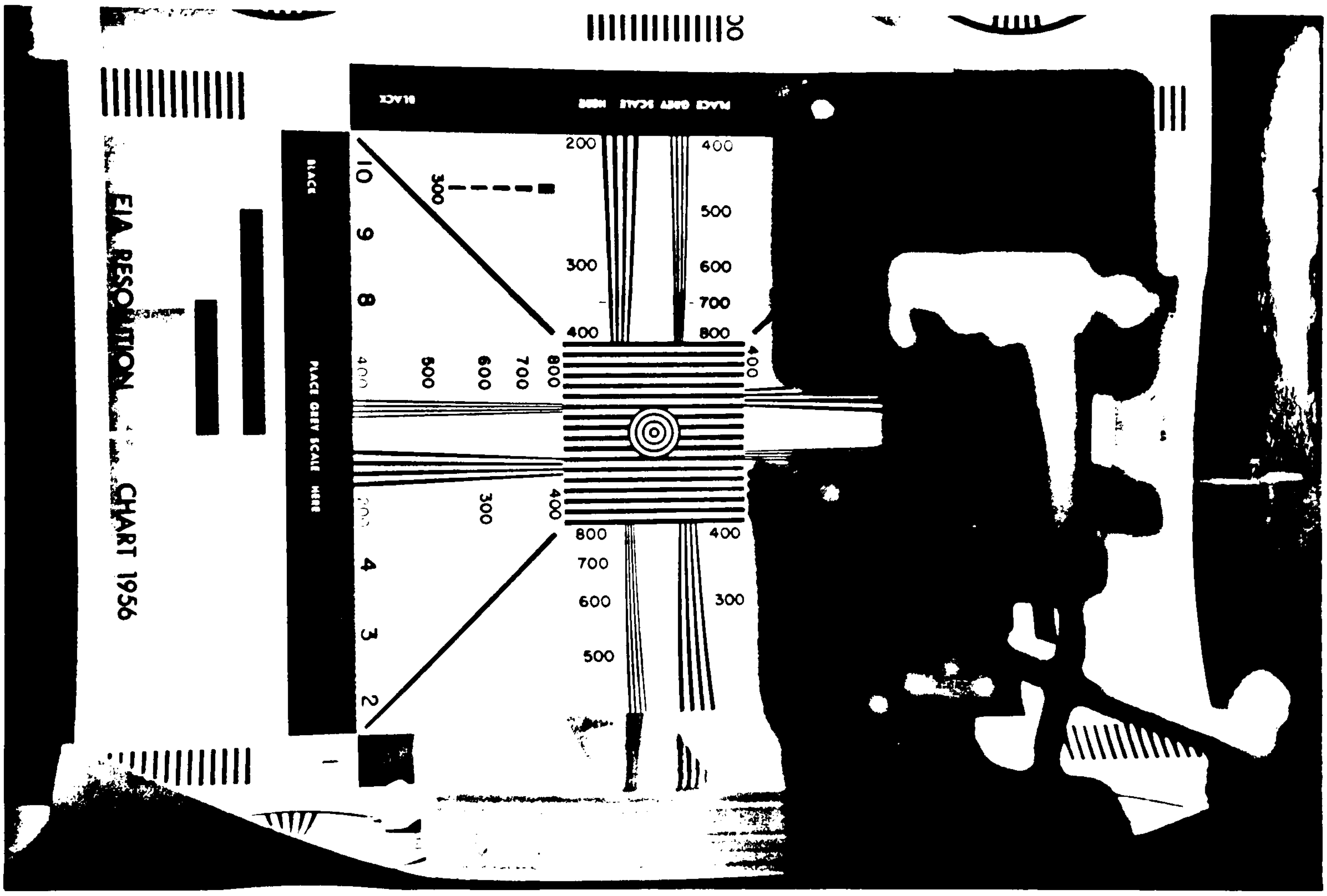


Figure 6.5

CCD Camera's View of the Resolution Test Image

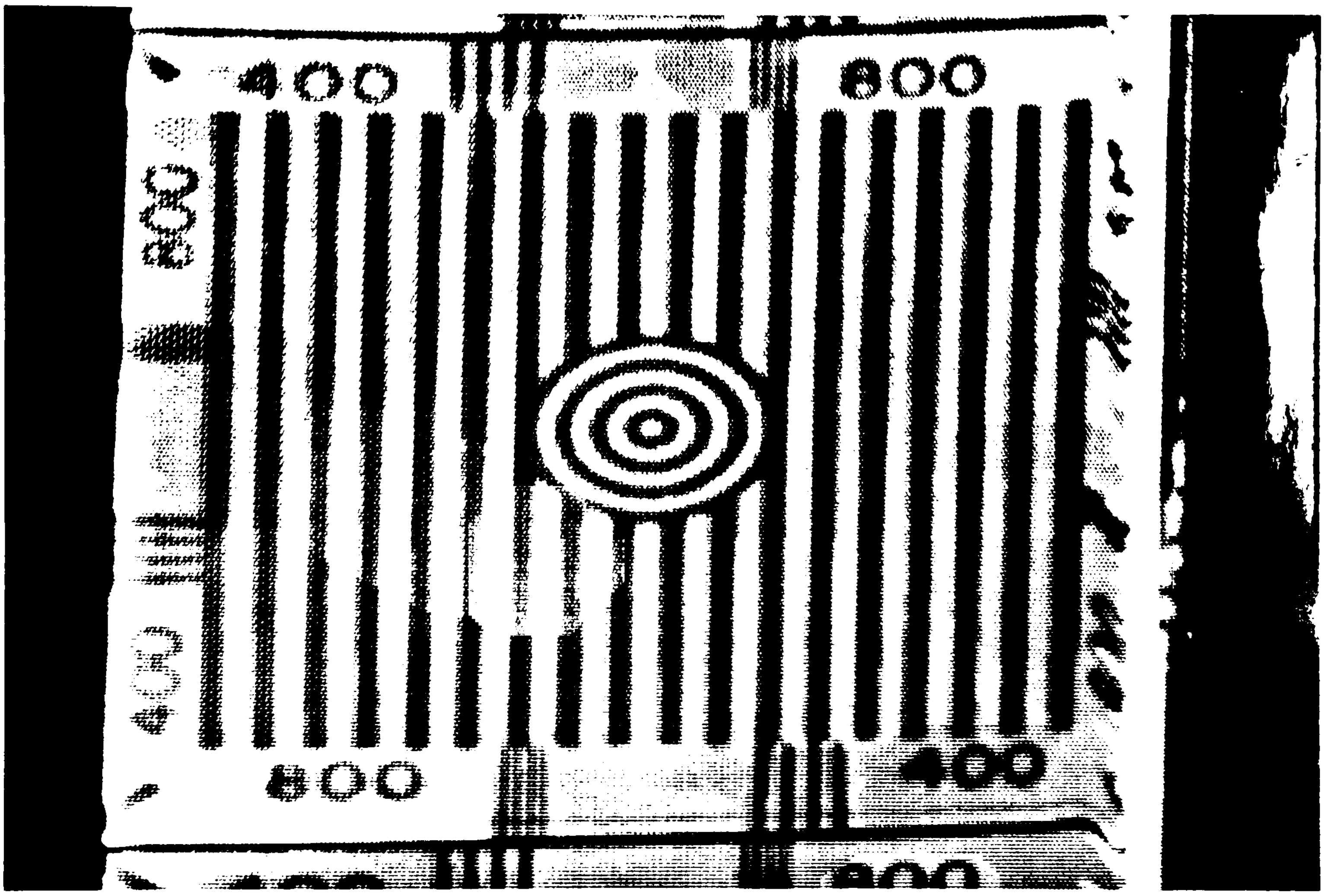
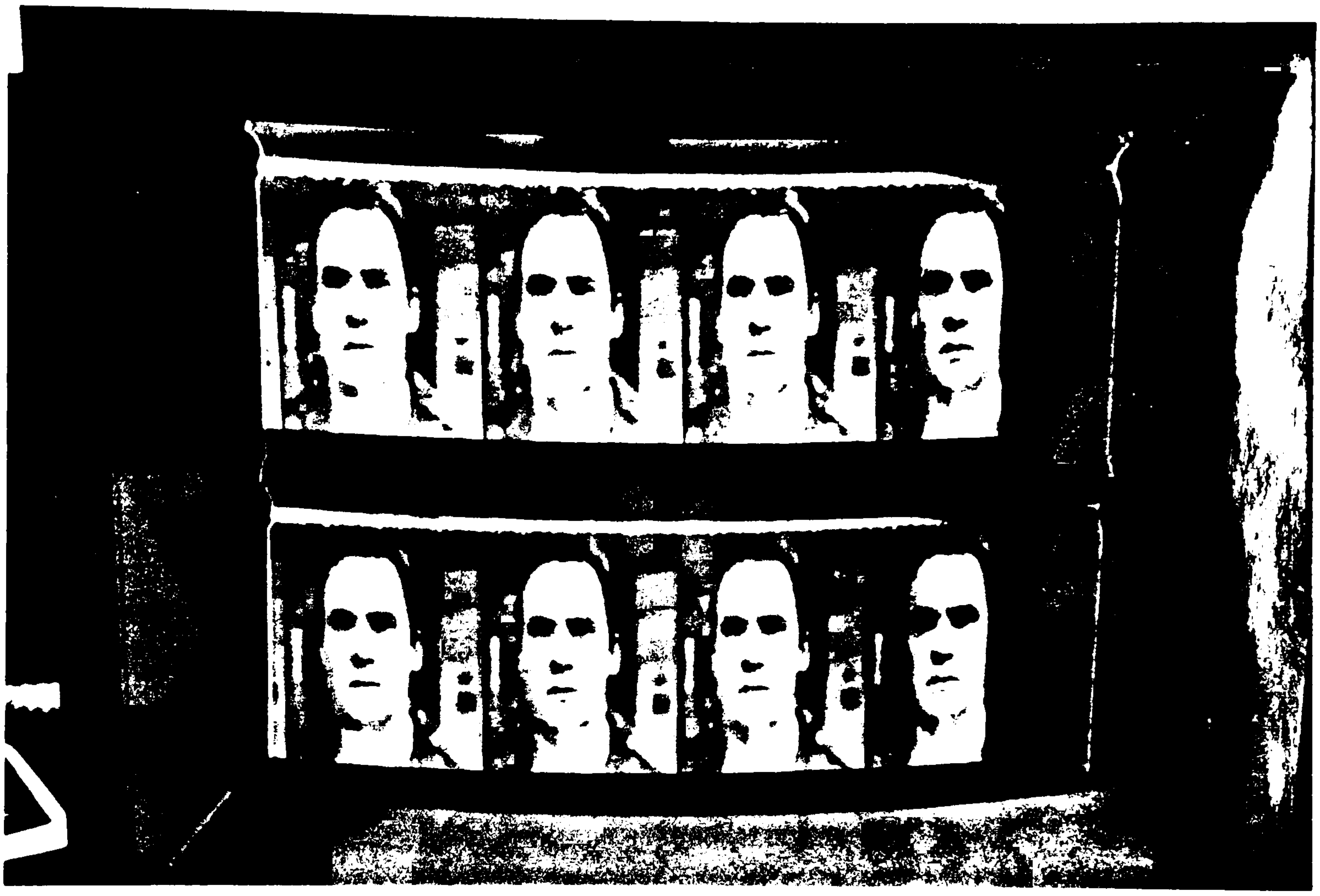


Figure 6.6

Closeup View of the Resolution Test Image Captured with the CCD Camera



. Figure 6.7

Image of Mr. John Van Meter Captured with the CCD Camera

Figure 6.5 is a photograph illustrating the CCD camera's view of a slightly different test image which tests the resolution of the image capture system. Figure 6.6 is a closeup view of the monitor while displaying the image captured by the system.

Finally, Figure 6.7 illustrates an additional image captured by the image capture system; this is a photograph of Mr. John Van Meter, author of the image capture software provided in Figure 5.13.

CHAPTER VII

CONCLUSION AND SUGGESTIONS FOR FUTURE WORK

The objective to develop an prototype imaging system based on the TMS34010 graphics system processor has been met. The implementation, as described in Chapter V, results in a dramatic reduction of imaging system hardware, by utilizing a special purpose graphics processor to provide the timing signals for image capture and display, and for dynamic memory refresh. Use of the GSP enabled the elimination of sixteen integrated circuits required by a prior PC based imaging system.

Given that the prototype image capture and display system operates correctly, it is prudent to suggest possible enhancements to the system, and directions for future work. Perhaps the most obvious improvement to the system implemented in Chapter 5 would be the consolidation of all discrete logic devices into a programmable logic device. Such an exercise would result in a considerable reduction of power consumption, and circuit board real estate, and would most likely realize cost benefits.

Further reduction in hardware would result from targeting the prototype system at a specific stand-alone GSP implementation. Implementation of a stand-alone design, such as a video telephone system, would allow elimination of much of the hardware currently provided on the software development board, and on the camera interface board, resulting in a cost-effective, minimum part solution.

REFERENCES

(1) Kim-You Mau, A CCD Image Sensor Frame Grabber and Conditioner, Master's Thesis, Texas Tech University, December, 1987.

(2) Ming Young, A Vision System for a Small Image CCD Array, Master's Thesis, Texas Tech University, August, 1988.

(3) Arun N. Netravali and Barry G. Haskell, Digital Pictures, Plenum Press, 1988.

(4) Ernest L. Hall, Computer Image Processing and Recognition, Academic Press, 1979.

(5) Richard Louis Baker, Vector Quantization of Digital Images, Dissertation, Stanford University, 1984.

(6) Frank H. Bower, "CCD Fundamentals," Military Electronics/Countermeasures, February 1978.

(7) Optoelctronics and Image-Sensor Data Book, Texas Instruments, 1987.

(8) Peter D. Lawrence and Konrad Mauch, Real Time Microcomputer System Design: An Introduction, McGraw Hill, 1987.

(9) MOS Memory Databook, Texas Instruments, 1986.

(10) Richard Wilton, Programmer's Guide to PC and PS/2 Video Systems, Microsoft Press, 1987.

(11) High Performance CMOS Databook, Integrated Device Technology, 1988.

(12) TMS34010 User's Guide, Texas Instruments, 1988.

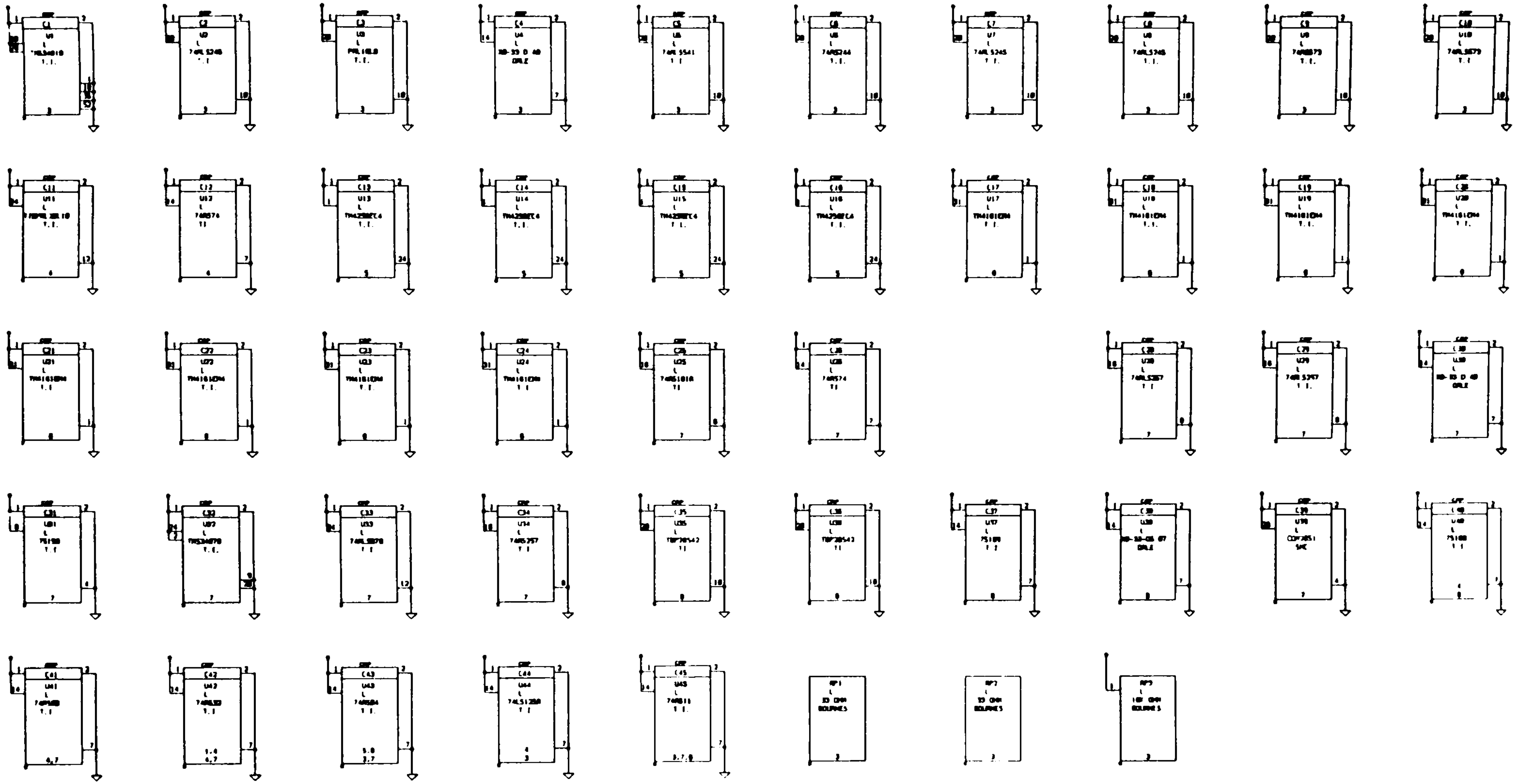
(13) TMS34010 Third Party Guide, Texas Instruments, January 1989.

(14) TMS Software Development Board User's Guide, Texas Instruments, 1987.

(15) TMS34010 Software Development Board Schematics, Texas Instruments, 1986.

APPENDIX A

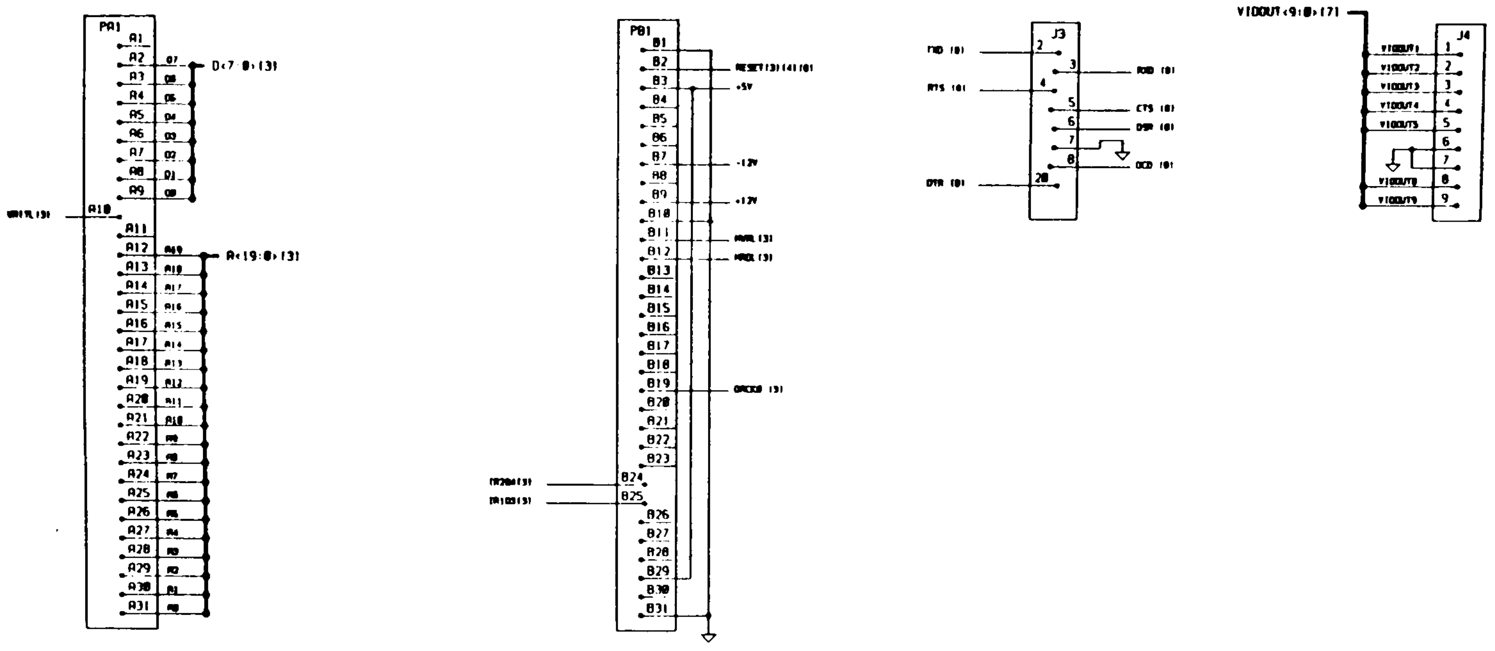
SDB SCHEMATICS



NOTE:
1 ALL CAPACITORS ARE .22μF 25V CERAMIC DECOUPLING TYPE.
UNLESS OTHERWISE SPECIFIED.

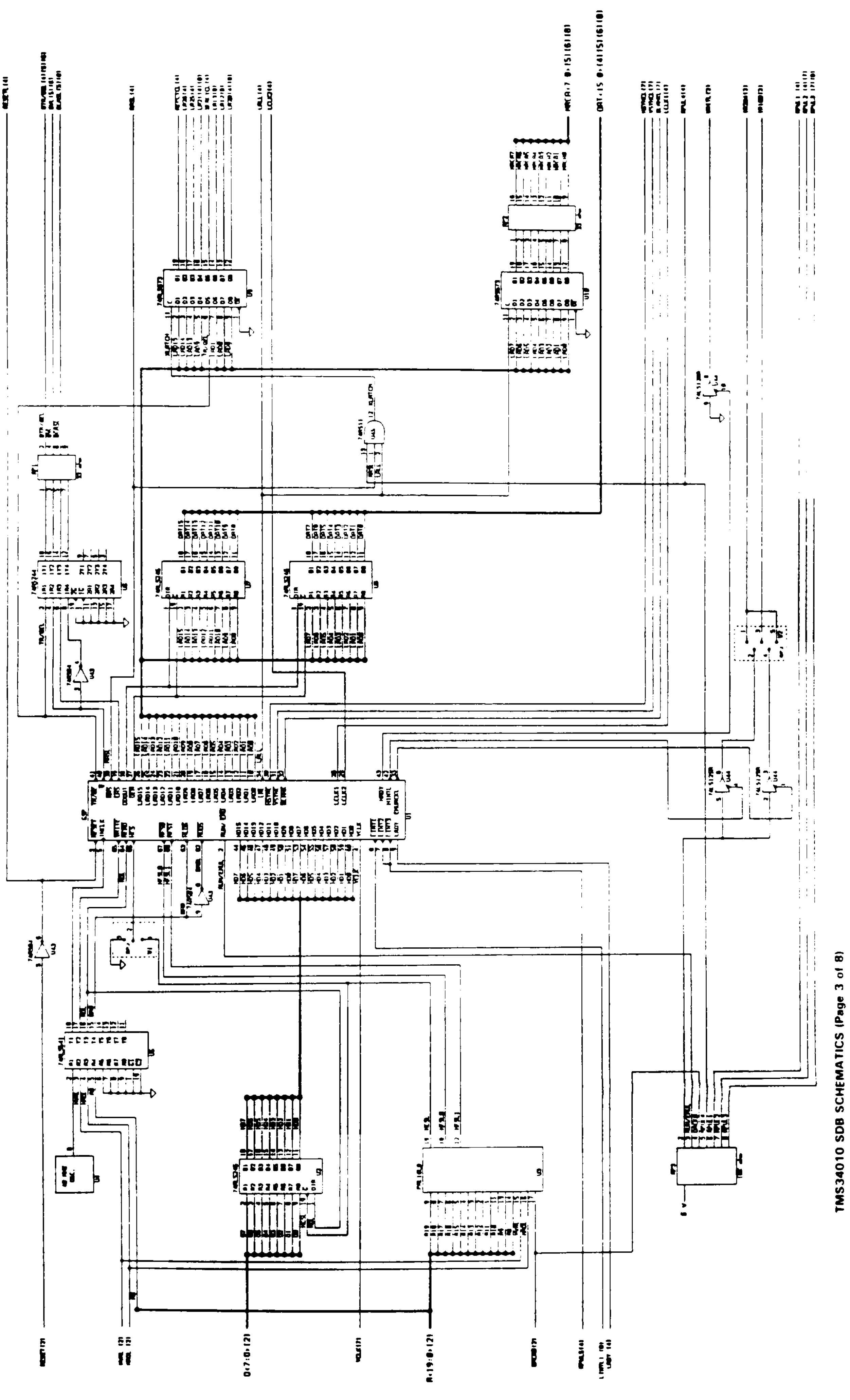
TMS34010 SDB SCHEMATICS (Page 1 of 8)
COMPONENTS DEFINITION

Figure A.1 SDB Schematic Diagram



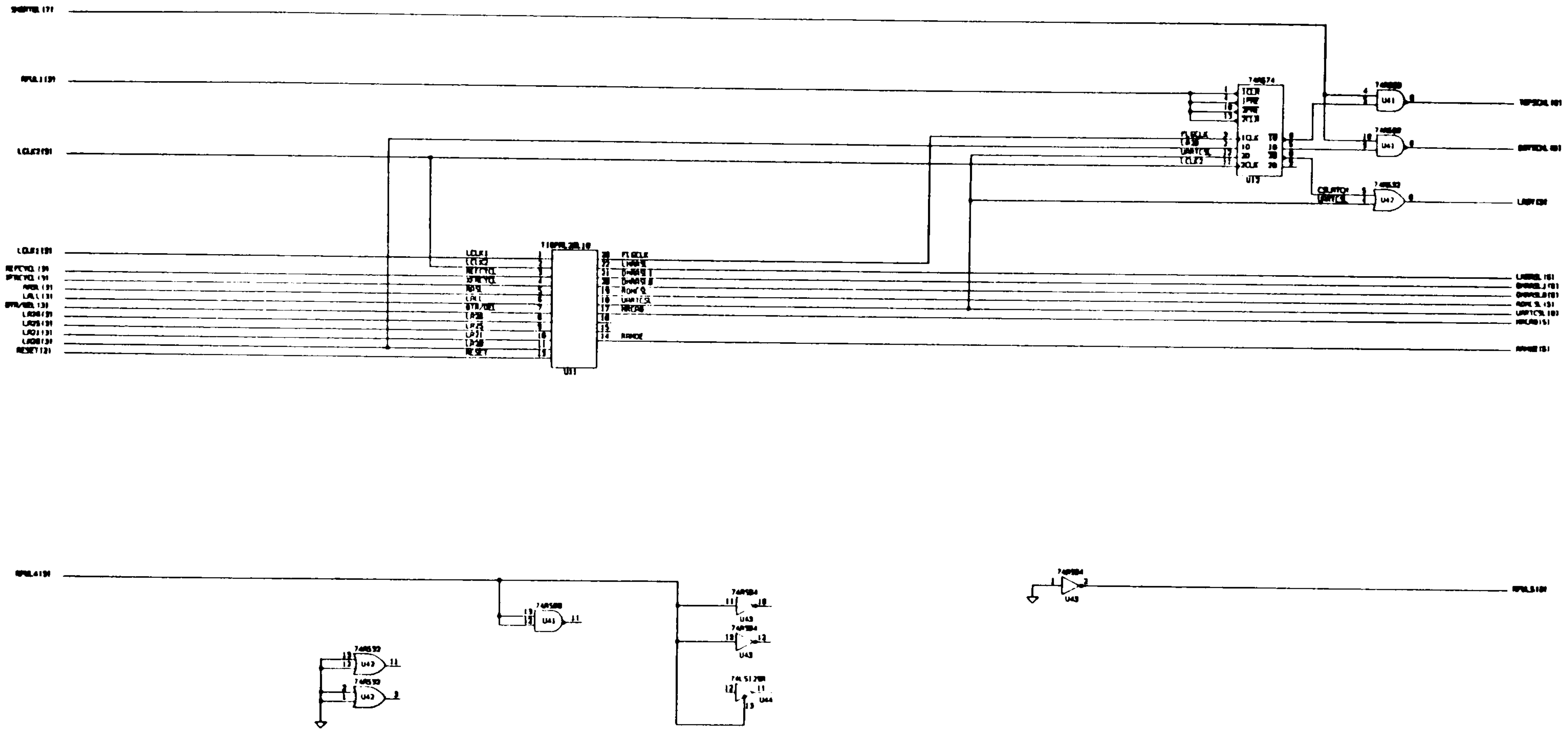
TMS34010 SDB SCHEMATICS (Page 2 of 8)
EXTERNAL CONNECTIONS

Figure A.1 Continued



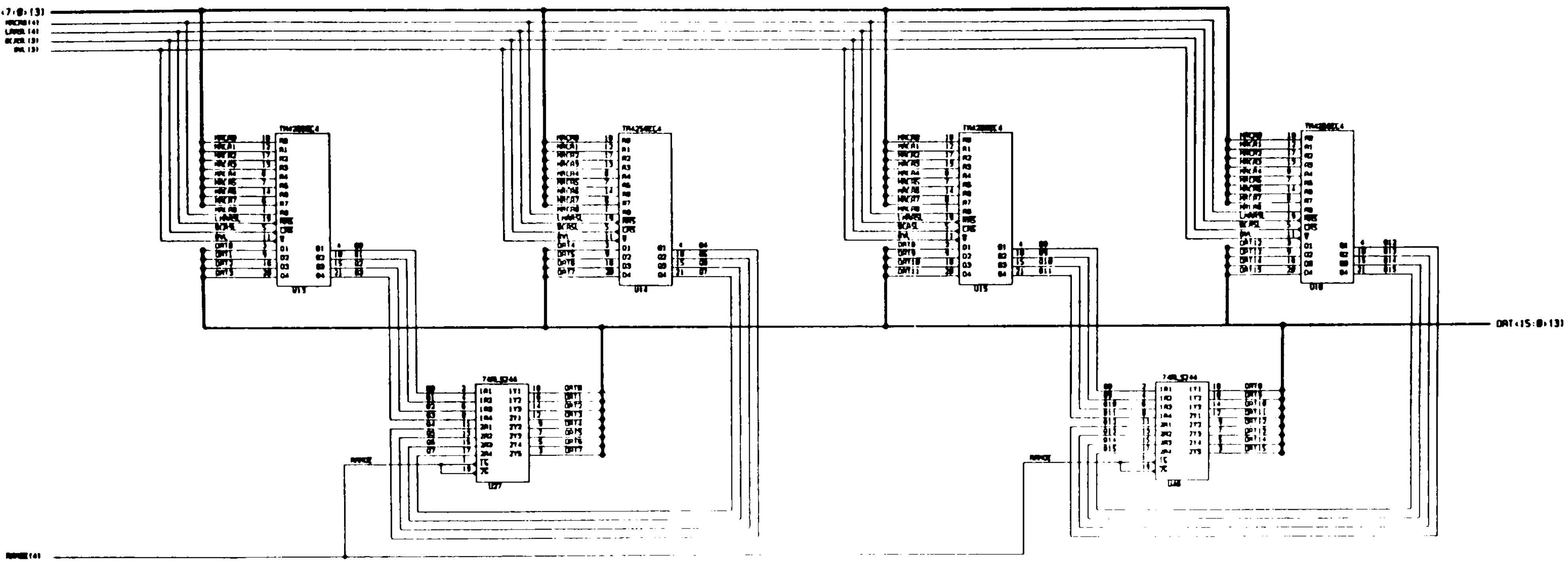
TMS34010 SDB SCHEMATICS (Page 3 of 8)
CPU SECTION

Figure A.1 Continued



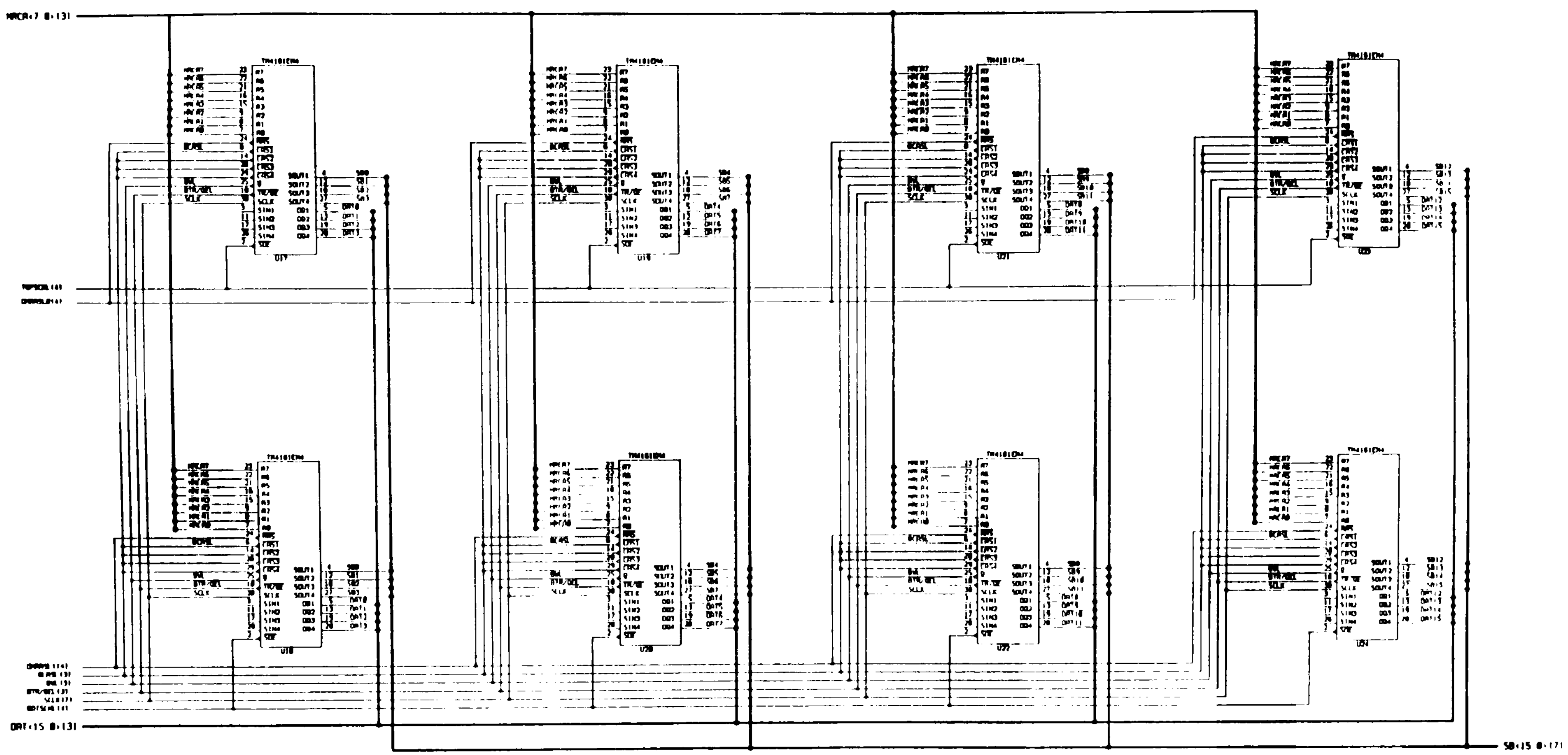
TMS34010 SDB SCHEMATICS (Page 4 of 8)
ADDRESS DECODER AND RAS CONTROLLER

Figure A.1 Continued



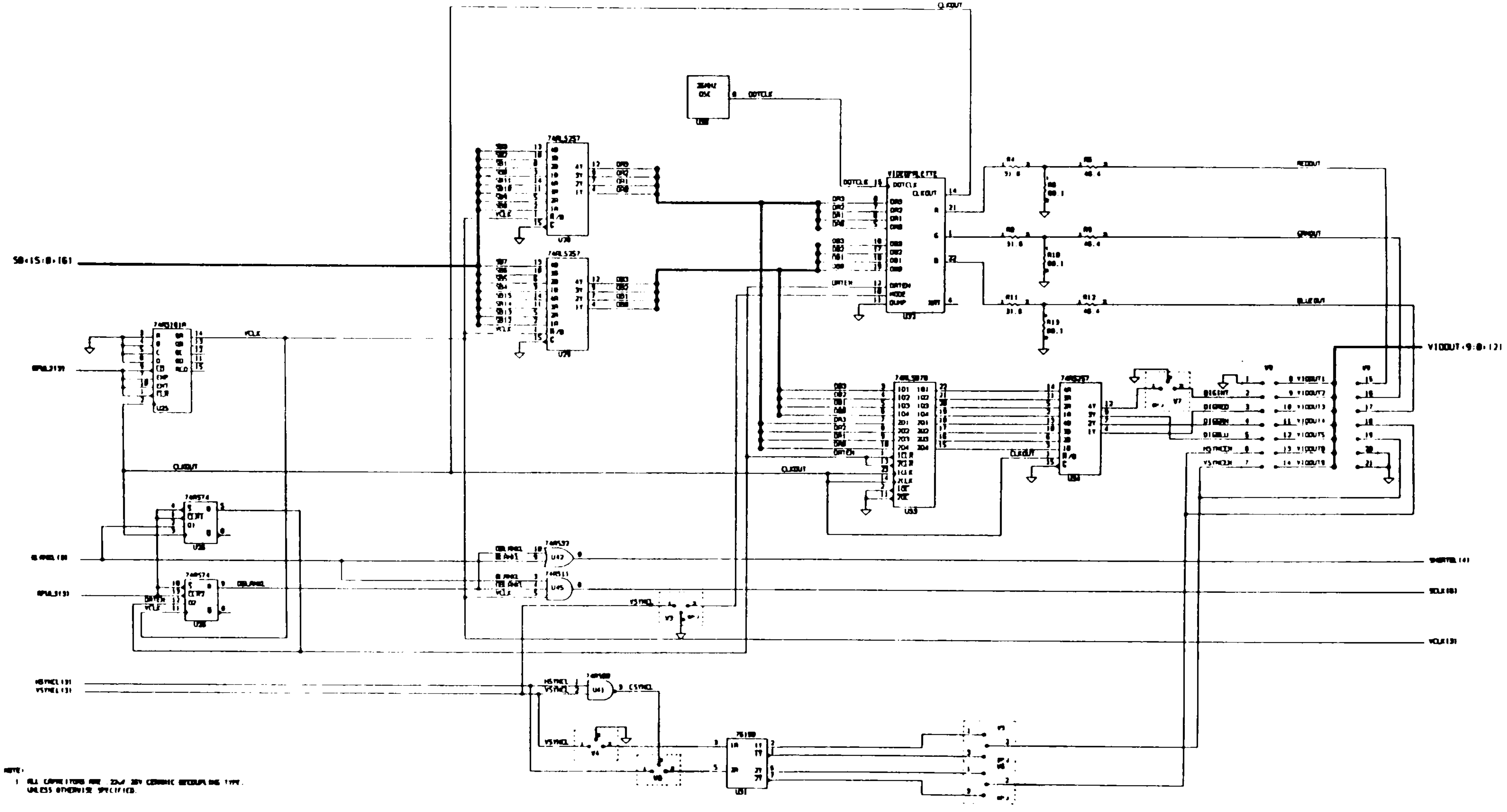
TMS34010 SDB SCHEMATICS (Page 5 of 8)
LOCAL RAM

Figure A.1 Continued



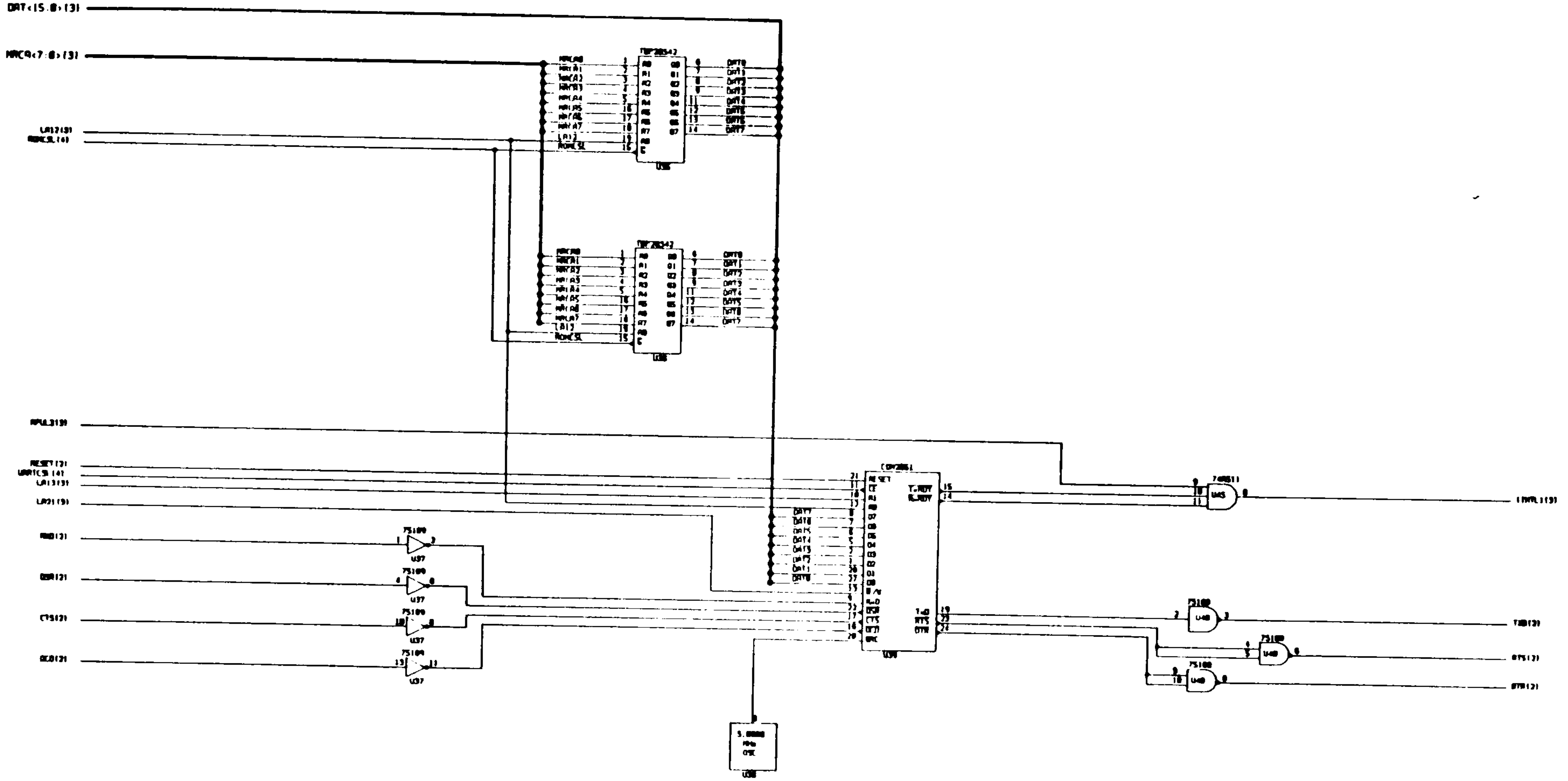
TMS34010 SDB SCHEMATICS (Page 6 of 8)
DISPLAY MEMORY

Figure A.1 Continued



TMS34010 SDB SCHEMATICS (Page 7 of 8)
JUMPER INTERCONNECTIONS

Figure A.1 Continued



TMS34010 SDB SCHEMATICS (Page 8 of 8)
SERIAL COMMUNICATIONS

Figure A.1 Continued

APPENDIX B

CAMERA INTERFACE BOARD SCHEMATICS

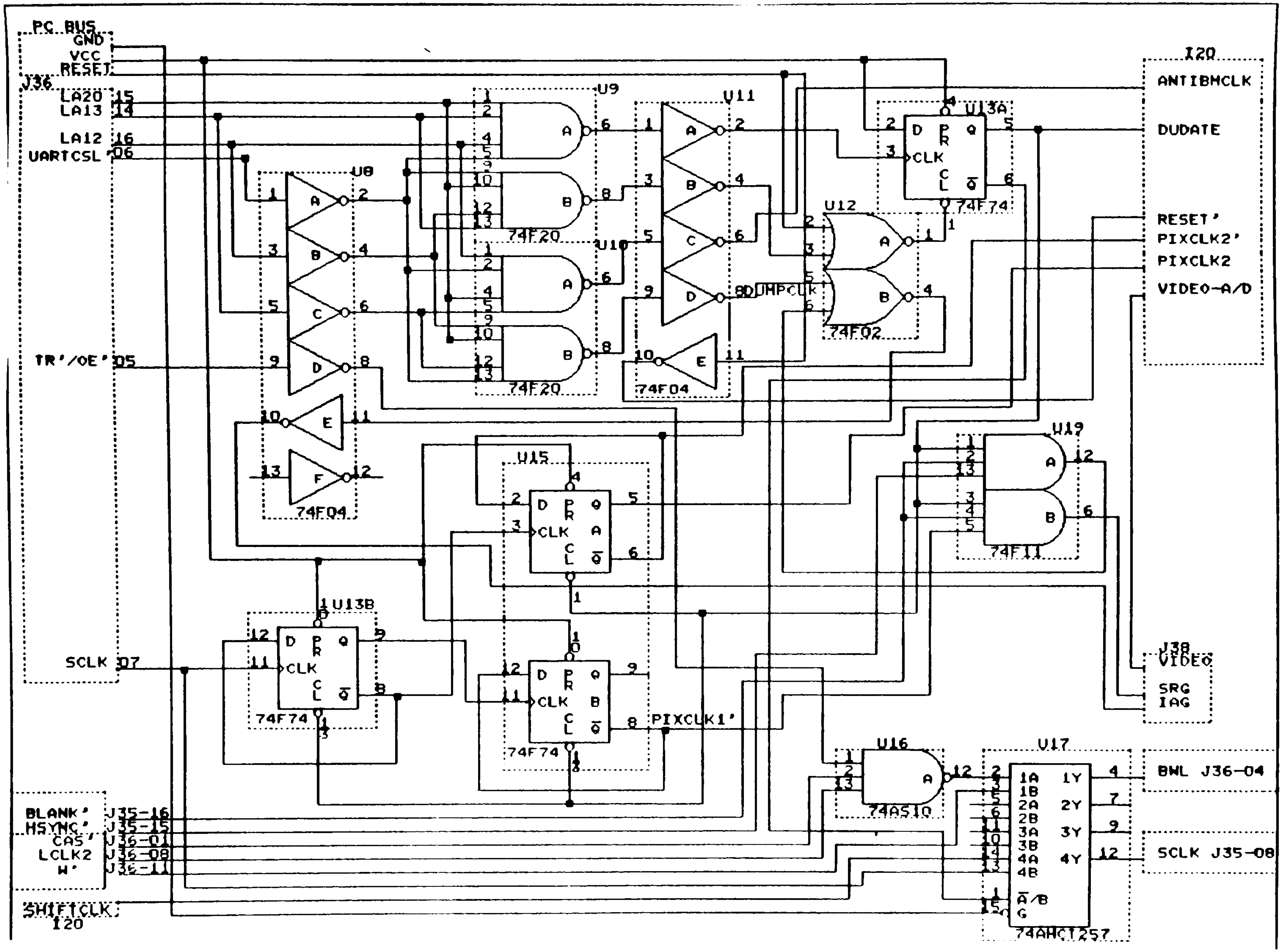


Figure B.1 Camera Interface Board Schematics

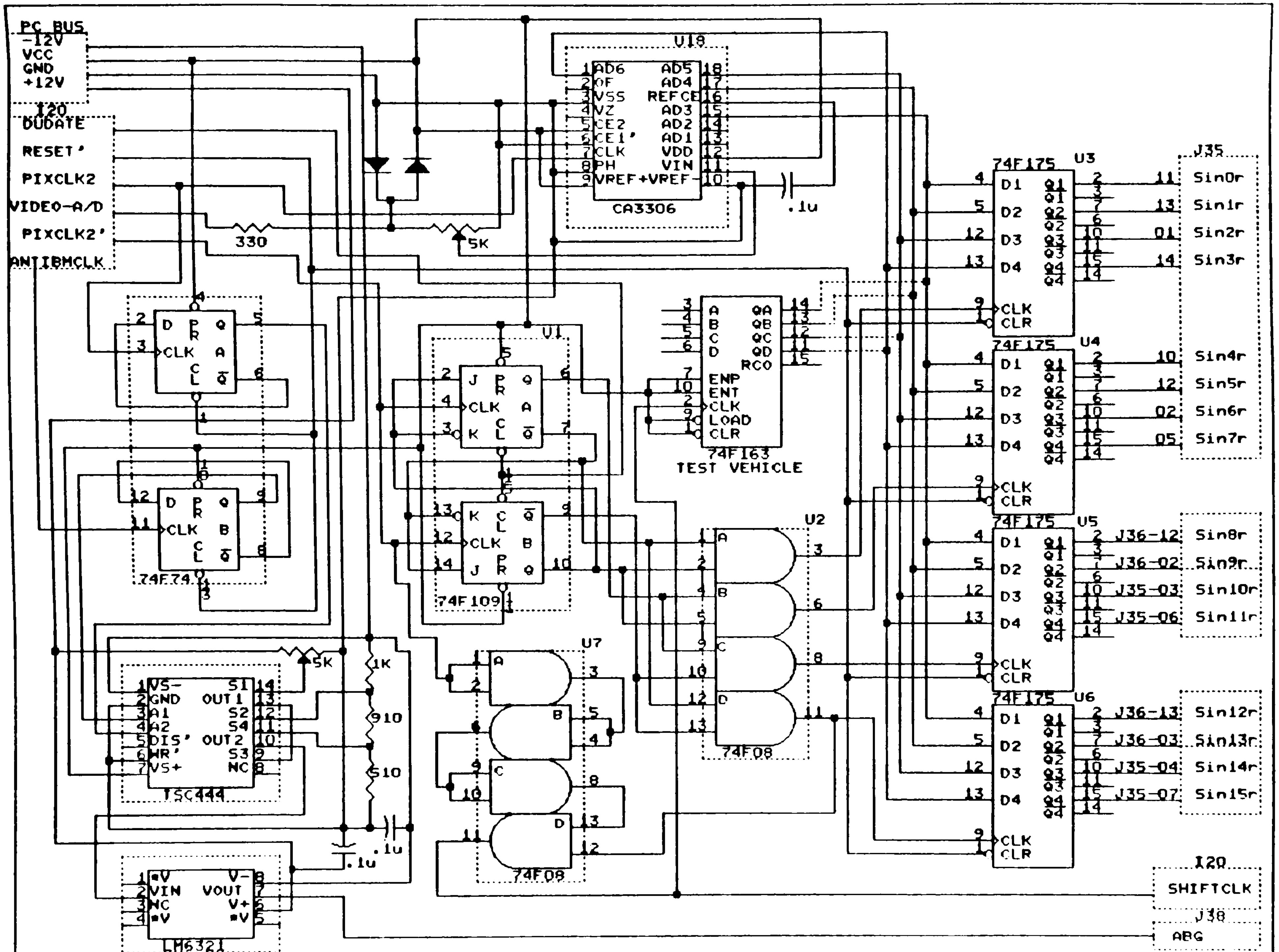


Figure B.1 Continued

PERMISSION TO COPY

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Texas Tech University, I agree that the Library and my major department shall make it freely available for research purposes. Permission to copy this thesis for scholarly purposes may be granted by the Director of the Library or my major professor. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my further written permission and that any user may be liable for copyright infringement.

Disagree (Permission not granted)

Student's signature

Date

Agree (Permission granted)

Student's signature

4/16/90

Date