On the Design and Optimization of a Smart Microprocessor-Based CRT Terminal to Serve as an Operator Interface to a Frequency Hopping Radio

Juan F. Salas

A Thesis

in

the Department

of

Electrical Engineering

Presented in Partial Fulfillment of the Requirements
for the degree of Master of Engineering at
Concordia University
Montréal, Québec, Canada

April 1984

ABSTRACT

This report presents the study, design and optimization of a smart microprocessor-based CRT terminal, which must communicate via an RS-232C port with a 68000- based board which plays the role of the Frequency Hop Generator in a communication's system.

The research and development presented results from the state-of-the-art Frequency Hopping Radio presently being developed at the Defence Communications Department of Canadian Marconi Company, which must function under the direct intervention of an operator who must enter adequate system operating parameters. Such coded parameters force the system to run under specific conditions. The need for the operator to directly interface with the system calls for a method to enter parameters which must be fast, efficient, reliable, low costing and must permit ease in updating such parameters at any desirable time. The simplest, cheapest, and most versatile method of programming such a system is via a CRT terminal.

The terminal will permit the operator to enter and visualize the system parameters and by means of a monitor system, residing on the 68000-based board, he will be able to:

- edit parameter tables depending on the system's needs;
- keep track of the system's real time clock; and
- place the system in any of 3 possible modes.

The report outlines the basic terminal concepts and CRT fundamentals necessary to develop a knowledgeable foundation in order that later on adequate decisions during the design stages can be made. By the same token, it carries out a detailed examination and comparison of three major LSI CRT controller chips available on the market in order to sort out the best functional component fitted for the job. Finally, it culminates by describing in detail the optimized design of a 32-chip smart terminal, where chip functionality has been blended in order to achieve the best performance out of every device.

Throughout the investigations carried out, and the design stages, it was always kept in mind that the main characteristics desired out of the terminal implemented were cost efficiency, low power, low component count, reliability, speed and intelligence. Needless to say, such were to be obtained without sacrificing operator flexibility to enter system parameters and yet incorporate sufficient intelligence to ease the task.

To Juan, Maruja, Mercedes and Suzanne;
Four special people who give life...and
love, true meaning.

ACKNOWLEDGEMENTS

As with the preparation of any lengthy report, I have become indebted to several people and would like to take this opportunity to thank them. My thanks go to Dr. N. Dimopoulos, my advisor and to Canadian Marconi Company for the practical experience I had a chance to acquire during the last three years in their employment as a design engineer. And last but not least, thanks to my girlfriend, Suzanne, for her patience and understanding, for the typing and proofreading of the entire report, as well as her suggestions that helped me improve the writing style and structure of the report.

TABLE OF CONTENTS

TABLE OF CONTENTS
(cont'd)

APPENDICES

TABLE OF CONTENTS
(cont'd)

SECTION 3 - DESIGN

TABLE OF CONTENTS
(cont'd)

APPENDICES

LIST OF FIGURES

Figure Page

CHAPTER 1

CHAPTER 2

CHAPTER 3

CHAPTER 4

LIST OF FIGURES
(cont'd)

CHAPTER 5

LIST OF FIGURES
(cont'd)

Figure Page

CHAPTER 6

APPENDICES

LIST OF TABLES

INTRODUCTION

Since the introduction of the microprocessor, video terminals have been designed with fewer chips, more features, and greater end-user flexibility. Although the microprocessor is the driving force behind these changes, reduction of chip count and consequently lower cost cannot be achieved with a microprocessor alone. Programmable I/O ports, programmable event counters, keyboard scanners and encoders, UARTS, Baud rate generators, video timers and controllers, character generators, and memory are the added functions needed to implement video terminals. In order to reduce the chip count, these individual functions in turn must be implemented in LSI. Fortunately, these complex functions are available in single or multiple chip MOS/LSI sets and it is now possible to build a versatile video terminal with 20 to 40 standard chips, depending on terminal features.

This dissertation will examine the various hardware and software decisions that lead to the design of a smart and compact, low power, low chip count, fast video terminal. The idea to develop a terminal with such characteristics arose from a study conducted while trying to purchase a low cost, intelligent terminal at the Defence Communications Department of Canadian Marconi Company (CMC).

The dissertation's breakdown is as follows:

Chapter 1 focuses on the various generations the terminal industry has passed through since the first teletypes. It points out as well the main differences in IQ between dumb, smart and intelligent terminals, while establishing general CRT terminal features to be considered by the designer in the early stages of the system. Thus, pointing out that the sum total of the features to be implemented will directly affect the hardware/software complexity of the final product.

Chapter 2 establishes the video basics necessary to develop a knowledgeable foundation to allow adequate signal interfacing during the design stage.

The next step, Chapter 3, is to consider the basic building blocks making up a CRT terminal and realize the logic which is most apt to be integrated into an LSI chip, the CRT controller. Once this is established, the interfacing format between the CRT controller and the character generator, and the various methods available to address and interface with the display memory are pointed out to further build upon the necessary knowledge required to choose the appropriate CRT controller for the job.

The background built upon during the first three chapters is utilized in Chapter 4 to carry out a detailed analysis of three CRT controllers, the potential candidates to be used in the terminal. The devices in question were chosen due to their availability, low cost, functional capabilities and characteristics from a wide range of CRT controllers available on the market today. For each CRT controller a terminal block diagram is presented pointing out the necessary number of components to implement it. To aid in the task of evaluating the best device, the choice is simplified and more obvious by presenting in tabular format a comparison of the CRT controllers' signals available, their characteristics and the necessary system components (by function). The reasons for the final choice are pointed out as well.

Chapters 5 and 6 describe the detailed steps and analyses undertaken to arrive at the final hardware and software design, respectively, making up the CRT terminal implementation.

Following is a comprehensive report, including the hardware (refer to **Figure 5.2**), software (refer to Appendix G - Software Listing) and all pertinent information regarding the design of my 32-chip "smart" terminal.

SECTION 1

PRINCIPLES

CHAPTER 1

TERMINAL FEATURES, CLASSIFICATION & TERMINOLOGY

The individual undertaking the task of designing a terminal, must first of all become familiar with the jargon pertinent to this aspect of the industry. By the same token, to arrive at an optimum design, he must consider the minimal terminal features, which to this day have become industry standards, that must be incorporated into the design in order to classify the terminal under the appropriate category and improve upon it.

This chapter outlines the general (but necessary) features and functions that must make up a present day state-of-the-art terminal, the basic differences that classify a terminal according to its degree of intelligence as well as other factors that must be taken into account to make it human interfaceable - ergonomic. The progress of terminal technology, through its first four generations is herein pointed out as well.

1.1 The Terminal's Genealogical Tree

The Cathode Ray Tube (CRT) terminal is one of the most (if not the most) popular computer-related products on the market today. It has become the keystone of interactive, decentralized data processing, and the focal point of almost all small business system development. While its utilization is becoming more and more common as the computer infiltrates all job markets, terminal manufacturers are becoming ever more convinced of the need for a compact, inexpensive, highly functional terminal with a variety of features and durability; a kind of universal tool. Such degree of sophistication, as portrayed by today's terminals, has not been accomplished overnight: it is the result of innovative breakthroughs over the past fifteen years.

In order to be able to understand why todays state- of-the-art CRT terminal industry is where it is, we must look back to the development and advancements in semi- conductor technology and the direct impact it had on terminal design by revolutionizing the hardware implementation. Such technology imposed restrictions on the design engineer, limiting him, as it developed, to use only the possible logic functions available on the standard SSI and MSI packages. The engineer's creative talent did the rest. Thus, functions not available as integrated circuits had to be implemented discretely; the price to pay was bulkiness and power. However, the achievements accomplished and the awareness obtained throughout the industry during every CRT terminal generation, dictated what the next logical step of integration should be in order to bring about improvements, versatility and compactness.

To better appreciate its present state, let's look at the evolution of the CRT terminal through its first four generations.

<u>First Generation</u>. In the first generation of <u>low-cost</u> CRT terminals (prior to 1974), both character decoding logic and CRT control were built from state-of-the-art TTL. This was a significant advance, and it resulted in a total system IC count of about 150. This was, of course,

a substantial amount in comparison to today's terminals. **Figure 1.1** shows a general block diagram of the discrete logic circuitry making up such a terminal.

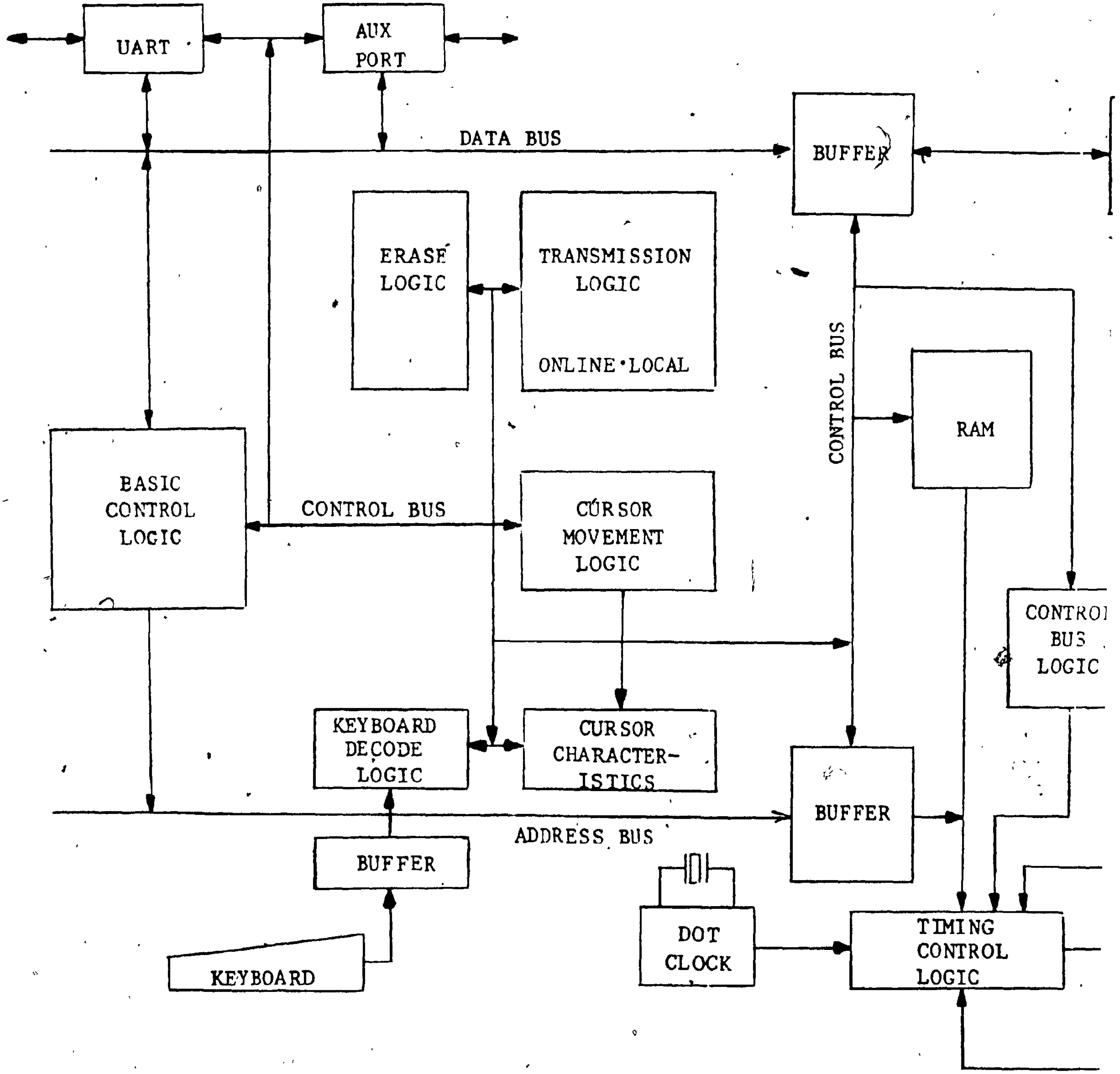
Every function, such as the generation of horizontal and vertical sync timing, was performed by discrete logic. The execution of line field increments, character-line address counting, or carriage returns required numerous gates to accomplish each function. Each command was treated independently by the hardware; it was necessary to have separate circuitry to detect each control character and cause the appropriate function to occur. Simply decoding input characters to cause normal displayable data to get written into memory needed decisions affecting numerous counters in the circuitry.

Everything was hard-wired in the first generation CRT. Options or changes were difficult to provide and labour accounted for the major portion of the terminal cost. An inexpensive early first generation CRT terminal was priced in the range of $1600 to $2000 and was simply an ASR33 compatible device - more or less a "glass-teletype".[1]

Second Generation. With the availability of LSI technology in the mid-seventies, low-cost terminals were introduced. However, the first real simplifying breakthrough came with the microprocessor (uP). The availability of the inexpensive uP permitted not only a simplification of the character decoding, but also allowed the designers to implement features that had been prohibitively expensive and to freely add new

FIGURE 1.1

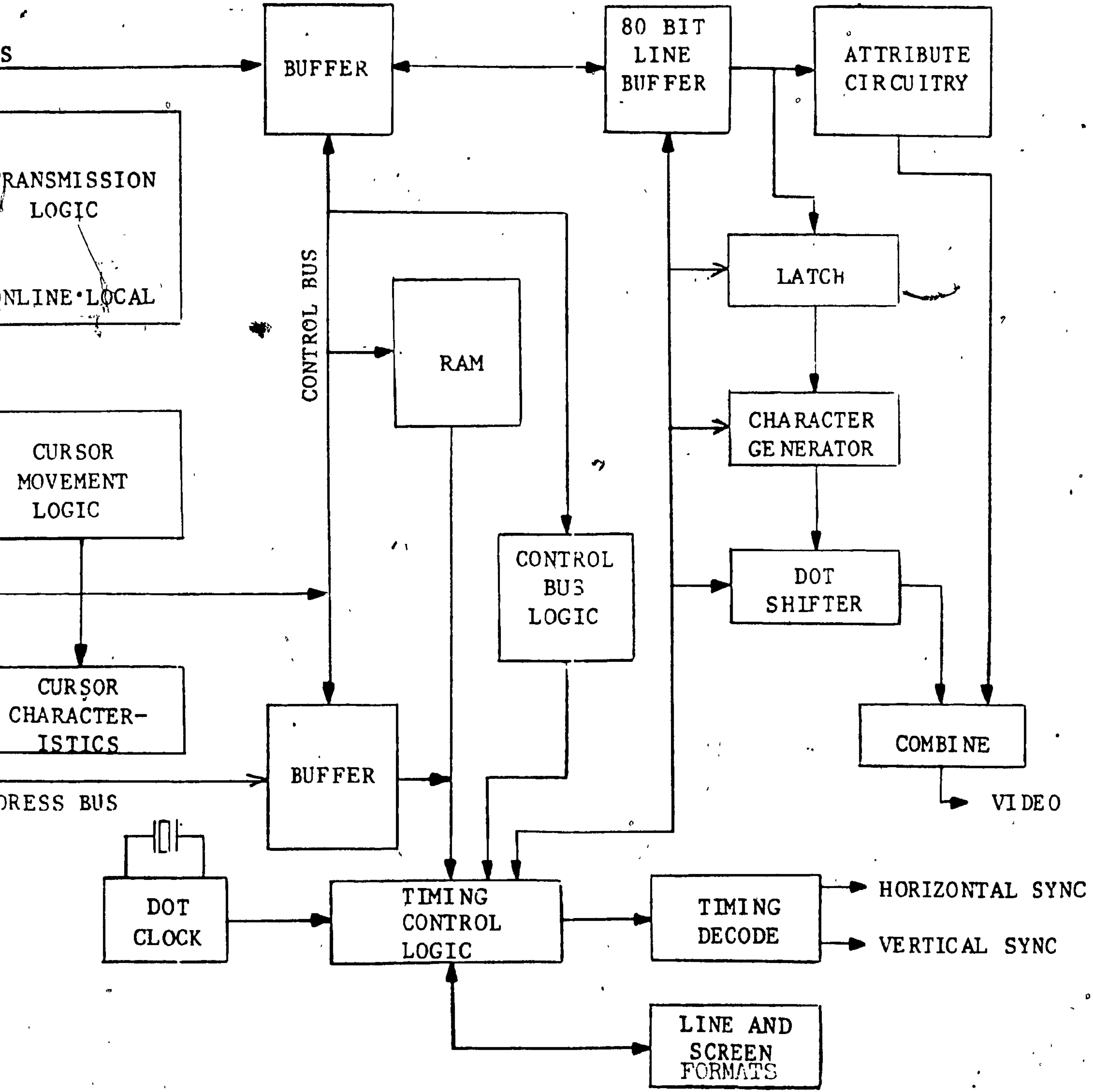
CRT Terminal Discrete Logic Implementation



1 OF 1

FIGURE 1.1

nal Discrete Logic Implementation



2of2

functions. It was then possible to build a second generation terminal with an IC count of about 100.

Adding the uP was really intended to give more function, rather than simply to emulate TTL circuitry. However, it wasn't just a question of adding a $15 to $20 processor to the system; other requirements such as ROM, RAM and clock circuitry became necessities as well. With the uP, editing functions could easily be added to the terminal. By adding extra memory and logic to allow the screen memory to be interrogated, a full editing terminal could be built having an IC count of about 125.

By using uP technology, a good deal of TTL circuitry was eliminated, but most noteworthy of the second generation was the ability to easily add many more standard features and options at a considerable reduced cost.

Previously, optional functions such as cursor addressing, the ability to set tab stops at every position, tab and backtab keys, automatic repeat keys, separate print keys, full upper and lower case characters, or a fully buffered print port, could be offered as standard features in second generation terminals because the additional labour charges occurred only once - in programming the uP. More important were the other implications - the promises for future enhancements - such as the ease with which options like function keys might be provided.

Inexpensive, dumb, second generation terminals sold for $900 to $1600 in 1976.[2] As uPs became more advanced, terminals incorporating the latest silicon intelligence could no longer be called "dumb". The terminal trade coined a new word, the "smart" terminal.

Somewhat suprisingly, the first uP-based intelligent terminals were no less complicated inside than the dumb variety - about 150 ICs for the first generation versus 125 ICs for the second. Computer circuitry had replaced much of the discrete logic, but the expanded functions had also necessitated increased complexity in the display-driver circuitry. An integrated solution to discrete video circuitry was needed.[3]

Third Generation. The market trend continued toward the development of smarter terminals. This was evident by taking a further step towards a reduction of the IC count. It was at this point that the second technological achievement took place. The development of the integrated CRT Controller (CRTC) or video controller chip resulted in lower circuitry complexity. Semiconductor companies such as Intel, Motorola, National Semiconductors and Standard Micro Systems (SMC) provided the next major opportunity for change with their introduction of such LSI chips as the 8275, DP8350, 6845 and 5027, respectively.

In the third generation terminals, the CRT control logic was replaced by an LSI circuit while still retaining the uP for the character decoding function. The programmable CRT controllers incorporated many of the discrete counters, registers and character-attribute circuits needed in a modern terminal. As a result, a third generation terminal could be built with about 60 ICs.

The new controller chips made it easy to do tricks with character attributes: blinking, blanked or underline characters; half-intensity or reverse video; and expansion to double height, double width or both. Adding the CRT controller gave terminal manufacturers some flexibility in the system. They could generate a number of different screen formats by reprogramming the controller (CRTC); make a whole family of terminals by changing the control firmware - from a simple "glass teletype" to a sophisticated editing terminal - with only minor hardware differences.

Actually, the CRTC simply interrogates memory and produces a display; that is its sole function: it generates and displays whatever is in the screen memory. **Figure 1.2** shows the general block diagram of a typical third generation CRT terminal.

But simply eliminating some logic on board, by using the LSI controller is overkill - it isn't cheap! The question that needed an answer was this: Can enough savings be realized with the general purpose, third generation controller to justify their use in low-cost terminals?

<u>Fourth Generation</u>. As terminal popularity and applications grew, and as the demand for such devices increased, the manufacturers that dominated this industry realized that the character decoding logic and the CRT control logic could be combined and incorporated into one custom LSI

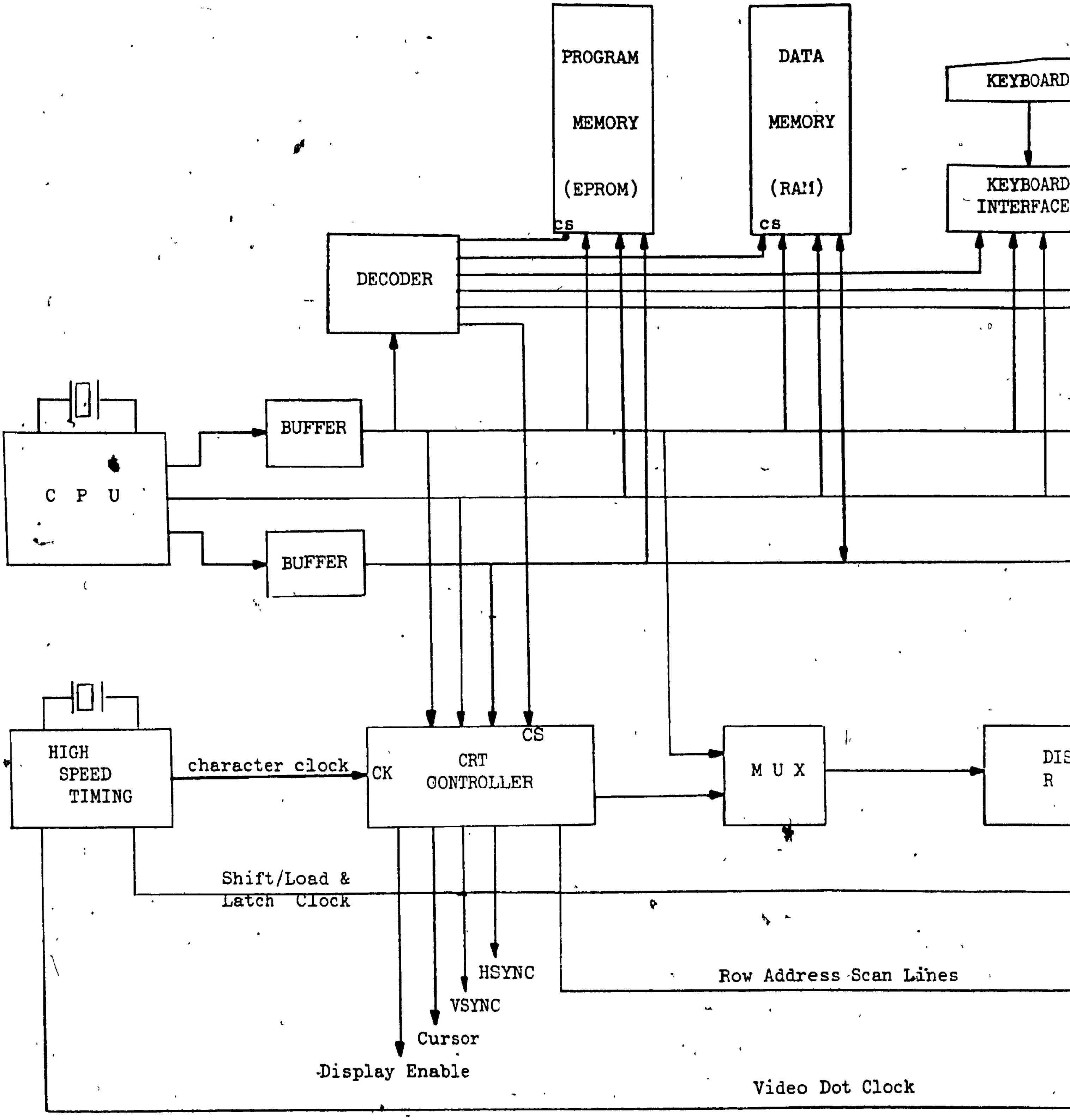
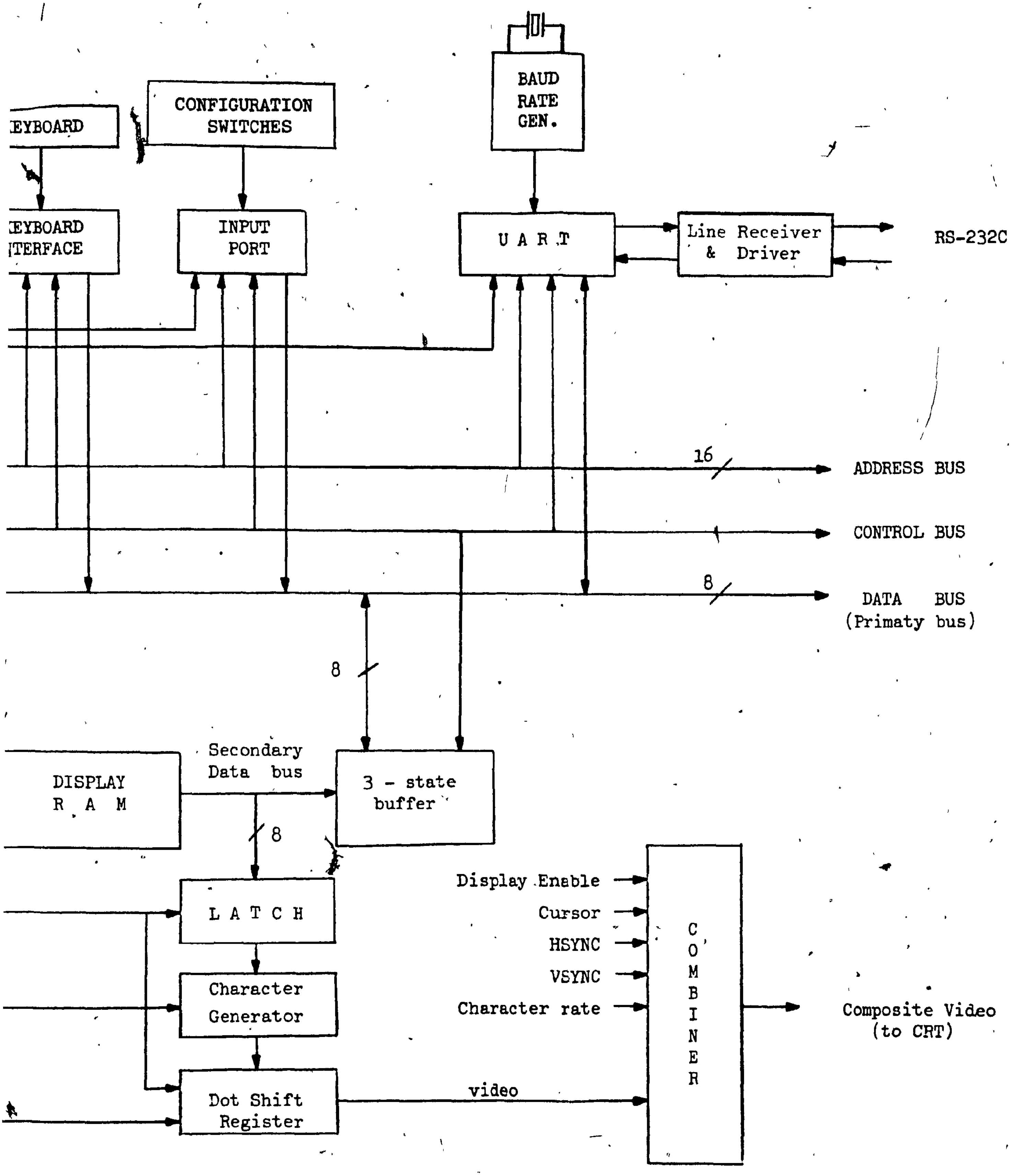


FIGURE 1.2

Third Generation CRT Terminal Bloc



al Block Diagram

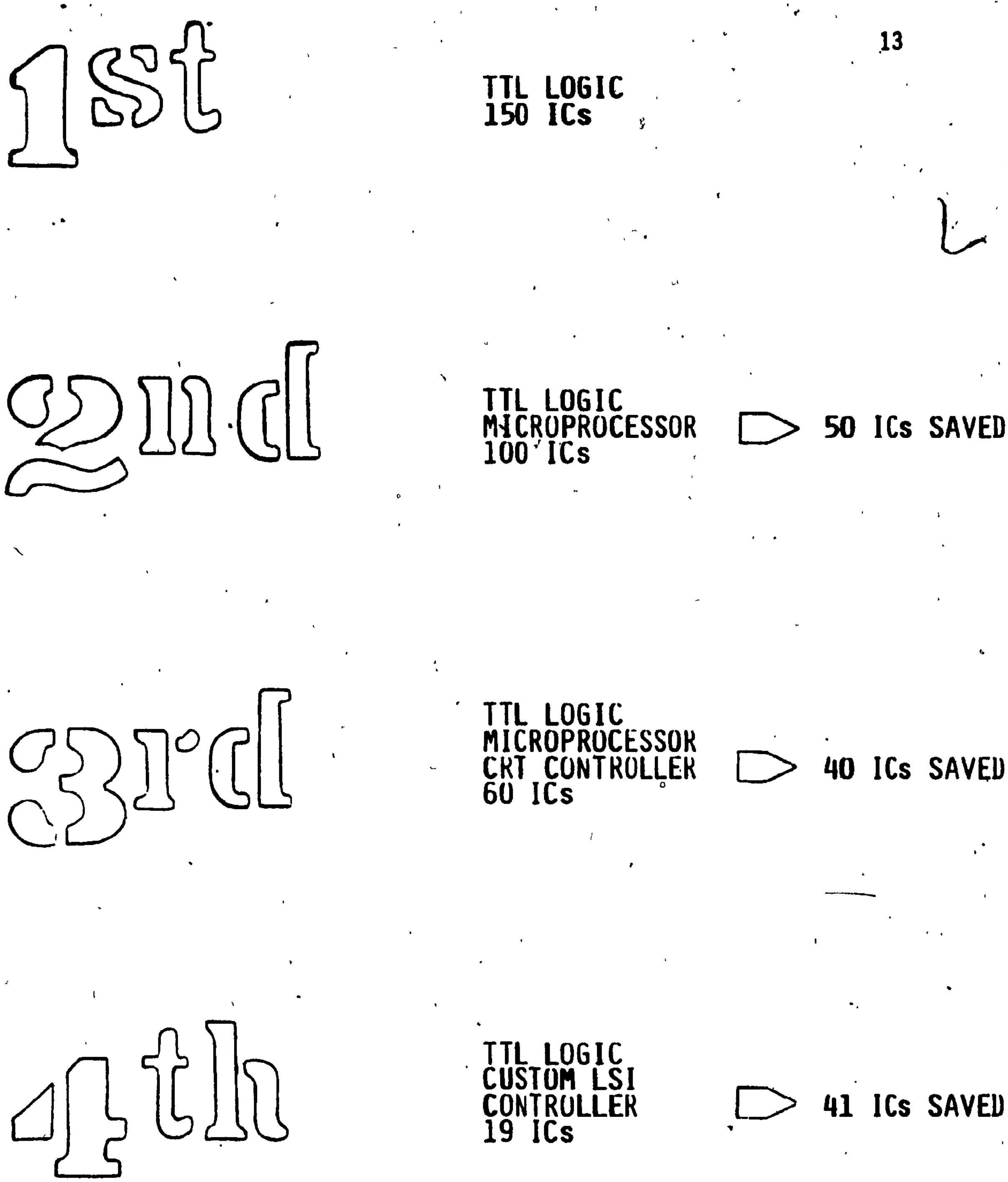
2 of 2

chip. The design of the custom controller chip, when produced in very large quantities, had the advantage of allowing a very cost-effective CRT implementation.[4] However, such an option was feasible only to manufacturers that had their own LSI design facility. Only then was cost reduction possible.

To design the custom LSI, the following factors were considered throughout the evaluation:

- the entire CRT circuitry had to be re-examined with an eye to redesign it;
- the major goals sought were significant cost reduction and performance improvement wherever possible; and
- the available general purpose LSI controller chips had additional features which were unnecessary for low-cost CRTs, while adding to manufacturing costs; thus, the custom chip had to incorporate only the necessary functions to be used for the specific needs and eliminate the many unnecessary options.

The research and development costs involved in such an operation were possible only to well-established giants such as Perkin Elmer, Lear Siegler and others. Nonetheless, their research proved fruitful, allowing fourth generation low-cost CRT terminals to be built with only 19 ICs. The less resourceful companies, however, had no choice but to remain with the third generation approach based upon the uP and CRT controller. **Figure 1.3** summarizes the evolution of CRT terminal technology.



1st

TTL LOGIC
150 ICs

2nd

TTL LOGIC
MICROPROCESSOR
100 ICs

▷ 50 ICs SAVED

3rd

TTL LOGIC
MICROPROCESSOR
CRT CONTROLLER
60 ICs

▷ 40 ICs SAVED

4th

TTL LOGIC
CUSTOM LSI
CONTROLLER
19 ICs

▷ 41 ICs SAVED

FIGURE 1.3

Compares the four CRT terminal generations using the low price CRT terminal for the comparison. It should be noted that the actual number of ICs to implement such a generation will vary with the options and cost. What is more important, are the numbers of ICs saved with each new design generation.

What can be expected from the next generation? Categorically, we can say that we have recently entered the fifth generation. With the advent of 16 and 32-bit uPs and the introduction of advanced CRT controllers by some of the major semiconductor companies, such devices are finding their way into the latest terminals being introduced. However, as terminal complexity increases and additional features get incorporated, the number of ICs necessary to implement them increases accordingly. Therefore, the trend towards ever-increasing intelligent terminals or towards the bi-functional terminal/microcomputer device directly affects the complexity of the software and in turn the hardware sophistication to implement it. This results in a higher chip count, which is a trend reversal from the previous four generations of lowering chip count while increasing functionality.

1.2 Terminal IQ[35,36,37,38,39,40]

Literally, a computer's window to the world, the CRT terminal, is the most widely used operator-controlled peripheral. This is so because it combines input, output, control and communications-interface functions all in one unit. The degree to which each of these functions is present determines a terminal's intelligence, which in turn governs its ability to manipulate data in a system environment. About 10 years ago or so, virtually all terminals were "nonintelligent" because they required an external host computer to control their operation. But nowadays, uPs and LSIs have changed all that.

CRT terminals are generally classified with respect to their features and internal capabilities in terms of relative intelligence. Such intelligence relates directly to the user's ability to program it. Most manufacturers classify terminals as follows:

Dumb: A dumb terminal is essentially a "glass teletype". They are merely conversational devices. They consist of:

1. a keyboard,
2. a CRT, and
3. an interface to a communications line.

The keyboard and CRT screen provide the basic I/O. Once data are entered via the keyboard, the terminal is limited to displaying and transmitting codes to the host computer which are then interpreted by the host for subsequent transmission and display. Any intelligence involved is in the device interfacing with the terminal. Dumb terminals offer limited control features such as carriage return and line feed. They are low-priced and easily adaptable to most computer systems.

Smart: A smart terminal has enough intelligence to place the data directly on the display screen from the keyboard without help from the host. By the same token, it provides data formatting, highlighting and text editing, allowing the data to be modified before block transmission.

Additionally, a smart terminal supports peripheral devices such as printers or mass-storage devices and transfers displayed information to the off-line storage device. It also buffers the information being sent to a printer or another peripheral device connected to the terminal. Many independently controlled and optional features are usually provided by smart terminals, each requiring additions and/or alterations to standard terminals. These include memory extension for display storage, function-keys, personality modifications, optional character-generator sets, and modifications of the standard program.

In general, most manufacturers cite the following as being part of the features making up a smart terminal:

- use of one or two uPs
- vendor (rather than user) programmability
- provides editing and screen functions supported by function keys
- provides block transmission capabilities
- provides minimal two page local memory

Intelligent: They usually feature:

- a wider variety of processing power, ranging from 8-bit uPs to 32-bit bit-slice CPUs
- user programmability
- sophisticated operating systems and high-level language capability
- larger local memory: 16K bytes and more
- support peripherals without reliance on the host such as floppy-discs and tape storage

Like the other terminals, intelligent units require support of a host system, particularly for downloading programs or data. However, they can stand alone when their hosts crash. These terminals resemble more and more computers. The emphasis in this segment of the terminal market is software. Their cost reflects their degree of intelligence.

Stand-alone: These terminals constitute a higher level of capability; they provide the basic features of intelligent ones, but add built-in computing power and thus don't require a host. They not only provide local intelligence and storage, but also offer capabilities similar to large host systems.

Terminal classification is becoming ever more difficult to make out, especially at the rate this industry is progressing. Although this doesn't imply that there is no distinction between some smart and intelligent or intelligent and stand-alone terminals, their capabilities overlap and the differences can be very subtle.

Despite differences over matters of definition, all terminal manufacturers are attempting to provide greater capabilities without seriously affecting price and performance. To enhance their products they are concentrating their efforts on three key design factors:

1. processing power;
2. display capabilities; ~~and~~
3. software.

A system's end-user often determines the type of processor its terminal requires, thus, whereas graphic systems may require bit slice uPs or 32-bit-wide data paths; on the other hand, alphanumeric applications are usually implemented using 8- and 16-bit uPs. However, many intelligent-terminal manufacturers agree that 8-bit uPs are sufficiently powerful for most applications.

1.3 Terminal Features

In order to qualify a terminal under a specific category, one must consider the entire spectrum of features it has to offer. Thus, the terminal manufacturer must decide from the conceptual stages whether he wishes to enter the market to compete with others either in the dumb, smart or intelligent terminal category. Depending upon the kind of terminal he manufactures, he may encounter tighter or more relaxed competition; he must carry out adequate market research. However, nowadays due to the larger scale of integration incorporated into chips and their intelligence, we can say that the introduction of new dumb terminals is practically nonexistent; there is a trend away from them. The lowest scale terminals that most manufacturers introduce into the market today are usually the "smart" ones. The manufacturers can accomplish such a degree of intelligence with a relatively small number of LSI ICs and at practically no additional cost when compared to the implementation of an equivalent dumb terminal. There is a large variety of ICs, off-the-shelf, to accomplish such a feat and more are being introduced every other day, incorporating more functions, greater intelligence and lower power consumption. The consequences of such a technological revolution are evident, they allow manufacturers to pack more computing power into a smaller space and at a lower cost, to the extent that more equipment is becoming multi-functional. That is, it may operate as a terminal in one mode, as a word processor in another mode and yet be a personal or business computer.

One of the main forces behind the innovation and variety of terminal products available today is the constant competitive war among the industrial giants to grasp a larger share of the existing market. After all, necessity may be the mother of invention, but it is competition that fathers price-slashing, which in turn forces manufacturers to greater extents of innovation in order to remain competitive, thus being caught in a vicious circle. The end result is to expand the frontiers of science, since it is such industrial giants whom through their R & D laboratories are able to improve upon older methods, discover new techniques and/or invent new processes which further adds to the fast-paced technological revolution we are all part of today. However, industrial giants are not the only contributors; often enough, Universities through their R & D programs, as well as inventors, provide major technological breakthroughs.

A terminal's IQ depends upon its programmability and the features it offers the end-user. **Table 1.1** summarizes the necessary capabilities and characteristics accepted as standard in this industry and supported by different manufacturers and OEMs, to a larger or lesser extent. For any particular terminal, its intelligence may be determined from this table by considering the number of features it supports.

I. SYSTEM CAPABILITY	II. SOFTWARE	III. AUXILIARY DEVICE DRIVE CAPABILITY	IV. EDITING CAPABILITY
A. User programmable 1. Firmware controlled (with PROM-burn capability) 2. Data entry controlled (tape, disk, etc.) B. Off-line processing C. Ability to control additional terminals D. Multiple CPUs E. Over 64K RAM, with expansion capability	A. High-level languages 1. Pascal 2. Basic 3. Other B. Assembly language C. Diagnostics 1. Off-line 2. On-line D. Graphics,color E. Graphics,monochrome F. Formatted forms control G. ANSI control	A. Disk B. Platter C. Printer D. Digitizer E. Magnetic Tape	A. Character Insert and Delete B. Line Insert and Delete C. Erase 1. Character 2. Line message 3. Field 4. Screen D. Character repeat E. Block move F. Word wrap G. Search & replace H. Margin adjust

TABLE 1.1

Possible Terminal Features; They Determine Termir

1 OF

NG
Y

Insert
!
't and

ter
essage

repeat

replace
just

V. KEYBOARD

A. Programmable Function Keys
B. Numeric pad

VI. DISPLAY CAPABILITY

A. Multiple scrolling fields
B. Programmable character generation
C. Protected format
   1. Alpha only
   2. Numeric only
D. Terminal command and status data
E. Tabulation
F. Alphanumeric Error message
G. Cursor functions
   1. Position
   2. Blink
   3. Home
   4. Tab
   5. Address read
H. Manual scroll
I. Paging
J. Blinking
   1. Character
   2. Field
K. Video
   1. Reverse
   2. High-low (fixed levels) or highlights
L. Underline character
M. Double-width characters
N. Double-height characters

VII. TRANSMISSION CAPABILITY

A. Buffered transmit
B. Node
C. Auto Call
D. Partial Screen transmit
E. Protocol Emulation
F. Transmission-interface format
   1. RS-232
   2. RS-449

.1

ne Terminal IQ[35,36,37,38,39,40,41]

2 of 2

1.4 The Man-Machine Interface

From the operator's point of view, the **keyboard** and the **CRT screen** provide the basic I/O. Thus, the legibility of the display and the feel and usability of the keyboard are often the most important criteria by which the individual operator judges the **quality** of a terminal and the merits of working with a terminal-based system. The terminal designer and manufacturer must consider the two carefully; they are vital to the success or failure of the product.

The Visual Display: The visual display provides the operator with a means - and often the only means - for checking the content and accuracy of the information which is entered via his/her own keyboard or which is communicated to the terminal from other parts of the system. In most terminal applications, therefore, the visual display serves a vitally important **control function** in allowing the information to be searched in order to locate specific items of information, errors, etc.

The effectiveness and ease with which this can be done depends on both the **legibility** and **readibility** of the display.

Poor legibility can have serious consequences on the ability of the individual operator to successfully and reliably carry out the work for which the terminal is intended. Thus, display legibility is, therefore, one of the most important criteria by which the merits and quality of a terminal-based system are judged.

To ensure the good readability of a visual display, a number of requirements should be satisfied with regards to the design, formation and stability of the character images. Coding, display capacity and formatting also play a major role in determining the suitability and ease of use of a terminal for specific applications. Among the key factors to consider which affect the legibility and readability of the display, there is:

Character Formation

Scanning resolution
Character height and width
Intercharacter spacing
Line spacing
Upper & lower case characters
Character color

Display Capacity

Screen size
Scrolling
- single page
- multiple page
(by character or smooth)

Display Coding

Alphanumeric codes
Graphics
Enhancement codes
- character brightness
- reverse video
- blink
- cursor

Image Stability

Flicker
Swim or drift

The Keyboard: Keyboarding is a complex process in which the movement of the hands and fingers is activated and controlled by signals from the brain. These signals are generated in response to three basic types of feedback:

1. kinesthetic feedback, i.e., by the sensation of touch, position and movement;
2. auditory feedback; and
3. visual feedback.

Feedback, therefore, is important for accurate and rapid keying and must play an important part in the design and characteristics of the keyboard. The parameters of the keyboard and individual keys which affect its feel and usability and have a direct effect on the operator's keying efficiency, and frequency of errors, are:

- the shape and profile of the key tops
- keyboard profile and layout
- keyboard thickness
- the dimension of the keys
- size and coding of the key legends
- key force and travel
- tactile and/or audible feedback
- key-roll characteristics (2-key, n-key)
- color and reflection characteristics of the keys and keyboard surface
- provision of numeric pad and function keys

Thus, overall, the keyboard being the only operator data-entry device, must be designed to provide as much user friendliness as possible.

1.5 Terminal Problems and Human-Engineering Factors to be considered when Selecting a Terminal[35,36,37,38,39,40,41]

Although terminal features may be very appealing to the potential customer of a particular model, there are several problems he must be aware of which currently plague alphanumeric CRT display terminals in general. These include the following:

- tube too bulky
- limited display size
- lack of human factor design considerations
- lack of built in diagnostics to aid the customer in pinpointing failure problems

For the terminal manufactuher, it is advantageous to consider human-engineering factors and design them in, making the terminal more appealing due to its improved man-machine interface. These factors facilitate the operator's use and interaction with the terminal, thus improving his/her efficiency and preventing or delaying fatigue and errors. The major human-engineering factors to be considered which determine the ergonomics, or working merits, of the terminal are:

- provision of tactile and audible feedback;
- detachable keyboard designed for minimum finger extension and comfort using color-coded or lighted special function keys;
- large 9 x 14 display font with generous spacing between lines and characters;
- lower case characters with descenders below the line;
- specially darkened and etched glass to diffuse the tube's surface reflections and increase contrast with the data being displayed on the screen;
- tube tilt features and swivel action to allow operator adjustment of viewing angle depending upon ambient lighting conditions, thus reducing glare;
- quiet displays (no fans incorporated);
- compact, lightweight displays to allow ease of relocation by operator;
- audible/visible alarms and controls for instantaneous operator feedback.

Other non-ergonomic factors which deserve special consideration are:

- modular design to simplify maintenance;
- built-in diagnostics to pinpoint failure problems:

The more features and ergonomics that are incorporated into the terminal, the better the product. However, the terminal's price tag will reflect the added features.

As a final note, it must be stressed that the terminal industry is about to face a turning point in the near future. The result of such an event is due to the commercial availability, in large quantities, of new processes and technologies which are presently reaching the maturing stage and are due for release within the next year or two. In particular, the following will influence the packaging, compactness, intelligence, and general features of future terminals:

- the solid-state flat panel display replacement of the CRT tube - its weight, ruggedness and size will prove an unbeatable advantage over CRTs;
- multi-line dot-matrix LCDs - their low power will prove crucial for portable compact flat terminals;
- **voice recognition** and **voice synthesis** built-in circuitry to speed up and ease the operator's interaction with the machine (leading to the keyboard's obsolesence?);
- larger memories: - 128 K x 8 EPROMs, ROMs, RAMs
  - 256 K x 1 dynamic RAMs
  - Megabit RAMs;
- CMOS versions of existing 16- and 32-bit uPs.

CHAPTER 2

VIDEO BASICS

In order to effectively use the LSI CRT controller circuits available today, some background knowledge of TV and video electronics is a necessity. In this chapter we will describe the general principles of operation of the CRT and the basic design techniques of video terminals and TV. Since the theory of operation of the CRT is described in great detail in numerous texts, our description here will concern the general aspects one should know in order to understand the interface between a CRT and its controller.

2.1 The NTSC Standard

The television industry has been alive and well for over 30 years, and yet the method of transmitting and receiving video picture information has remained essentially unchanged. In fact, the only real controversy in all that time occurred around 1953 when the Federal Communication Commission (FCC) was assigned the task of deciding upon an industry standard, based on the recommendations of the National Television System Committee (NTSC), for color TV operation that would be compatible with the existing black and white standard.[8] The NTSC standard for black and white TV existing at the time, and the one decided upon then for color TV are still intact today and used quite successfully. Such a standard specified the horizontal and vertical sweeping frequencies, among other parameters being used in North America. Understanding such parameters is becoming ever-more important as increasing numbers of home TVs are being used as low-cost video terminals.

2.2 Elements and Terminology

In all X-Y display applications, the purpose is to draw an image on the screen for human interaction. However, before discussing the methods of drawing these images, it is important to discuss display-related terminology and some of the important elements of a picture. Once this common ground is established, the methods of forming a picture can be analyzed in detail.

The important aspects of any picture are resolution, brightness and data density.

Resolution Two types of resolution, spot size and addressable resolution, are important in discussing a display device. Spot size resolution is simply the diameter of the electron beam spot divided into the screen dimensions. Since the light intensity of a spot is not uniform across its diameter, measurement of spot size is not an exact science.[7]

Addressable resolution, on the other hand, has nothing to do with spot size, but simply is the incremental accuracy in which the beam can be positioned. The smallest positional movement of the electron beam divided into the CRT dimensions then gives the addressable resolution. Unlike spot resolution, addressable resolution is an exact computable number. It is important to realize that addressable resolution holds no fixed relationship to spot resolution. Addressable resolution can be either greater or less than spot resolution. The relationship of addressable resolution to spot resolution does play a significant role in the perceived "quality" of a display or picture.

Brightness — Brightness is simply the light output of the picture being displayed. The relative light output of a displayed picture is significantly different from the maximum light output capability of the CRT.

Data Density — The amount of data displayed (data density) is simply the area of the CRT screen that is lit versus the entire screen size (or the amount of lit area versus dark area). The lit portion of the screen consists of all the alphanumeric characters that comprise the picture.

Raster — Raster is a predetermined pattern of scanning lines that provides substantially uniform coverage of an area. In other words the raster is the rectangular pattern of light which appears on the CRT screen when the screen is scanned by the electron beam.

Z-Axis — This refers to that part of the display that controls CRT intensity. It includes the input circuit, an amplifier (called the Z-axis amplifier), and the circuit that drives the CRT control grid. The control grid controls the amount of beam current allowed to fall on the phosphor at the front of the tube which varies the intensity of the picture.

2.3 CRT Deflection Fundamentals

The primary objective in any video display system is to present information to the viewer, whether it be in a picture format - as with standard broadcast TV - or in a text or graphic format - as with a computer. The device used for this purpose is the Cathode Ray Tube (CRT),

Figure 2.1a, a vacuum tube with a large, flat face coated with a special phosphor material that emits light when struck by an electron beam. A high voltage (15,000-25,000 volts) is used to accelerate the beam from the rear of the tube to the phosphor-coated face; the amount of light depends upon the beam current. The horizontal and vertical deflection circuitry determine which area on the screen is illuminated, with additional circuitry used to control the brightness, contrast, and focus of the display.

Two types of horizontal and vertical deflection methods - to deflect the beam inside the CRT - are in wide use today:

Electrostatic Deflection

Used in lab oscilloscopes and high speed graphic displays, provides the highest picture resolution and speed, but is also expensive to implement. Electrostatic deflection systems consist of complex electron gun structures containing two sets of deflection plates; one set for horizontal deflection and one set for vertical deflection. Electrostatic deflection, involving voltage charging of capacitive plates to deflect the electron beam, is capable of speeds several orders of magnitude higher than electromagnetic for a comparable amplifier cost (but not CRT cost). Since even inexpensive industrial oscilloscopes generally are capable of displaying 500 kHz or more on the vertical axis (and often on the horizontal axis as well), it is not suprising that most oscilloscopes and X-Y displays use the electrostatic deflection scheme.

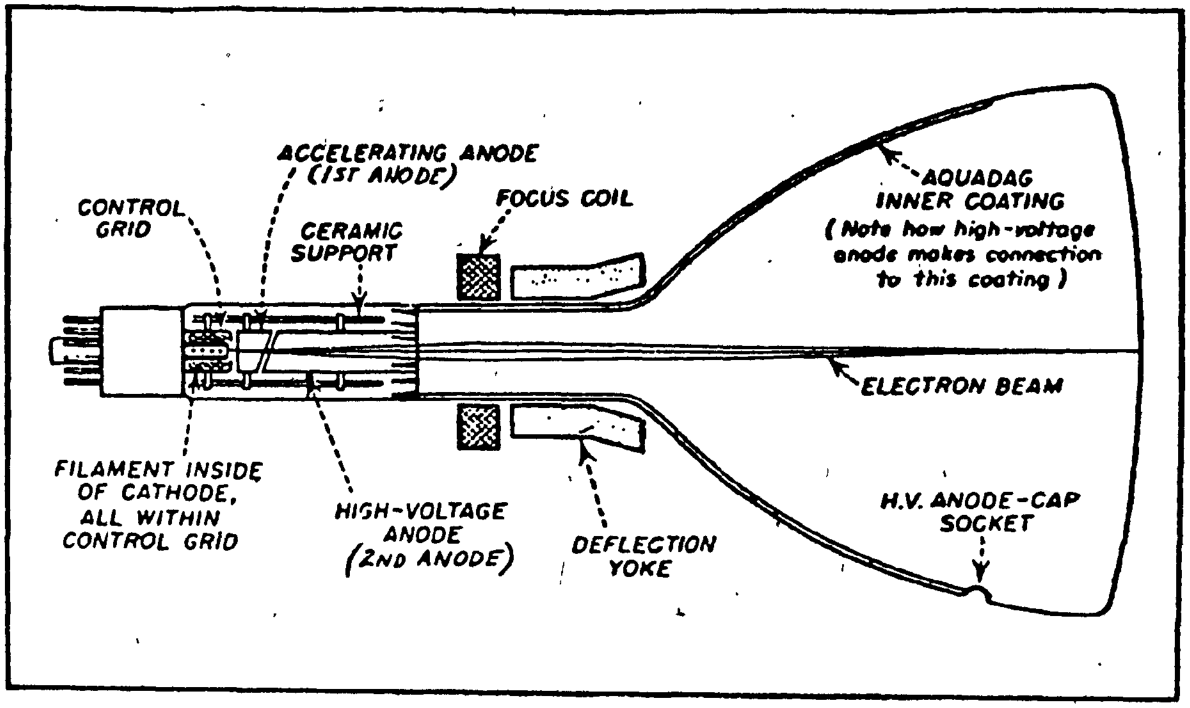

ACCELERATING ANODE (1ST ANODE)
CONTROL GRID
CERAMIC SUPPORT
FOCUS COIL
AQUADAG INNER COATING (Note how high-voltage anode makes connection to this coating)
ELECTRON BEAM
FILAMENT INSIDE OF CATHODE, ALL WITHIN CONTROL GRID
HIGH-VOLTAGE ANODE (2ND ANODE)
DEFLECTION YOKE
H.V. ANODE-CAP SOCKET

FIGURE 2.1a

Electromagnetic Deflection-Type CRT. Commonly used in television, computer display terminals and some inexpensive low frequency (less than 20 kHz) oscilloscopes.[7]

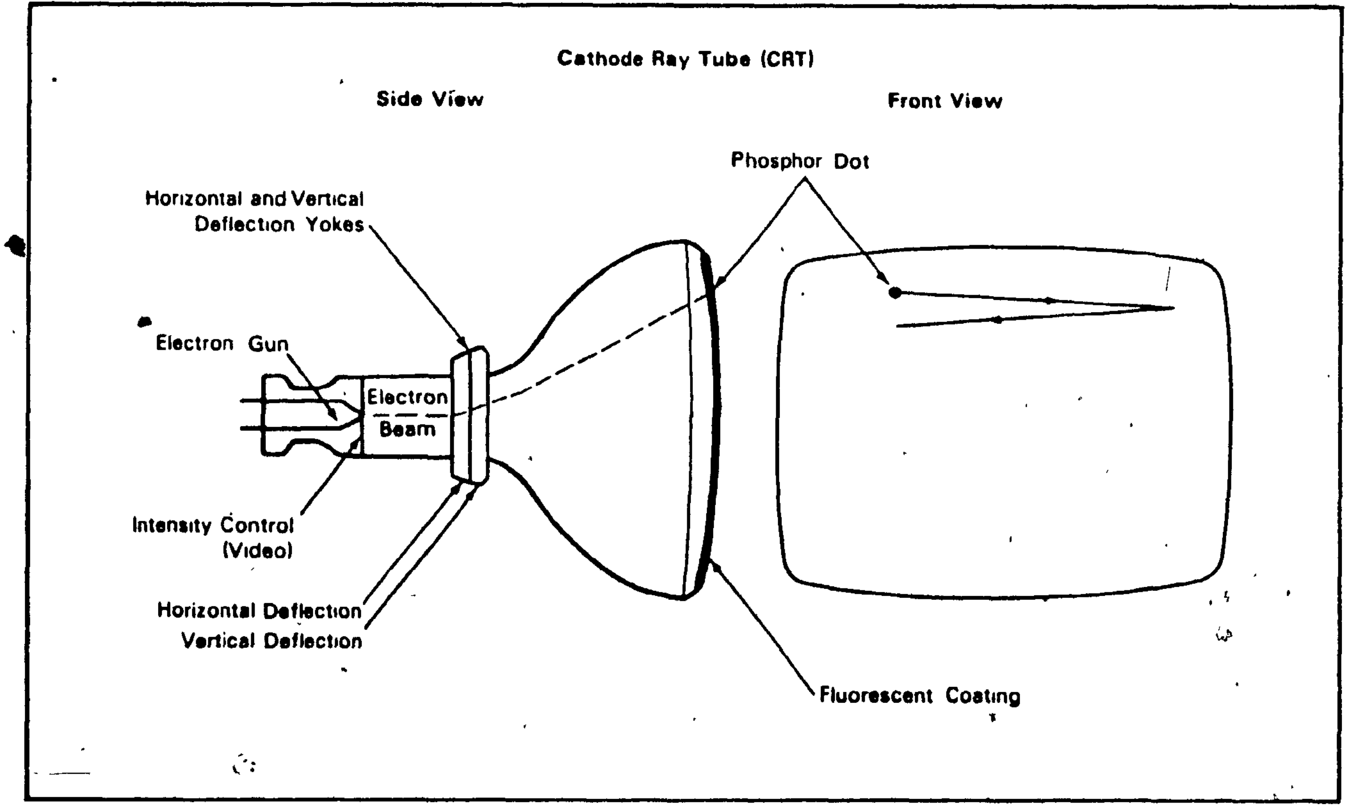


FIGURE 2.1b

CRT Deflection System - How the electron beam's movement is controlled.[5]

Electromagnetic Deflection

Electromagnetic deflection systems utilize an inductive coil (called a yoke) surrounding the neck of the CRT through which a current signal is passed to generate a magnetic field to deflect the electron beam. At high frequencies, inductors with few turns are necessary to obtain fast current changes, and larger currents are required to obtain the required field strength. Consequently, above repetition rates of 20 kHz, large power dissipations are required to obtain full-scan displays. Most **television sets and computer display terminals** use the magnetic deflection scheme.

Electromagnetic vs. Electrostatic

The major difference between the two systems are light output and frequency response. The magnetic system generally offers more light output while the electrostatic system offers high-speed response at low power consumption. In addition, magnetic deflection allows a wider beam-deflection angle than does electrostatic deflection. Moreover, when the full-screen defletion bandwidth desired is less than 20 kHz, the electromagnetic deflection system (amplifier and CRT) has a substantial cost advantage over an electrostatic system. This is one of the reasons television sets, many medical monitors, and some oscilloscopes rely upon CRTs with electromagnetic deflection.[6][7]

When talking about methods of beam deflection, two more factors must be considered:

Light Output

Brightness, which is a function of beam current, is governed by the internal construction of the CRT. Given the same spot size, with all other things equal, an electromagnetic CRT will be brighter than a comparable electrostatic CRT. This is mainly due to the fact that the electromagnetic CRT gun structure is very simple, allowing most of the beam current to pass through to the screen. The electrostatic gun, on the other hand, has a large number of internal elements, and the front-most element of the focus lense has a relatively small aperture, which strips away 70-90% of the beam current just before the beam enters the deflection plate region thus, affecting the brightness output.

Deflection Speed

In either type of CRT, it is desirable to deflect the beam at as large a mechanical distance from the screen as feasible. This optimizes deflection sensitivity by providing the greatest deflection distance at the screen per volt, or gauss, of applied deflection field. In the electrostatic CRT, the plates must also be reasonably close together to achieve sufficient deflection field strength with an applied voltage in the range of 300 volts. A greater swing would create significant X and Y amplifier design problems.

Electrostatic displays have faster deflection systems and use less current than electromagnetic displays. This is the result of physics; the load of the deflection plates is only the stray capacitance of a few picofarads. On the other hand, the inductance of the yoke is the load to the driving amplifiers. The higher the inductance (i.e., more turns and stronger magnetic field), the easier it is to deflect the beam; however, higher inductance requires high power. To lower the inductance reduces power requirements but also reduces deflection sensitivity. This then requires reduction in accelerating potentials in the CRT so that the beam can be deflected full screen, which also reduces the light output.[7]

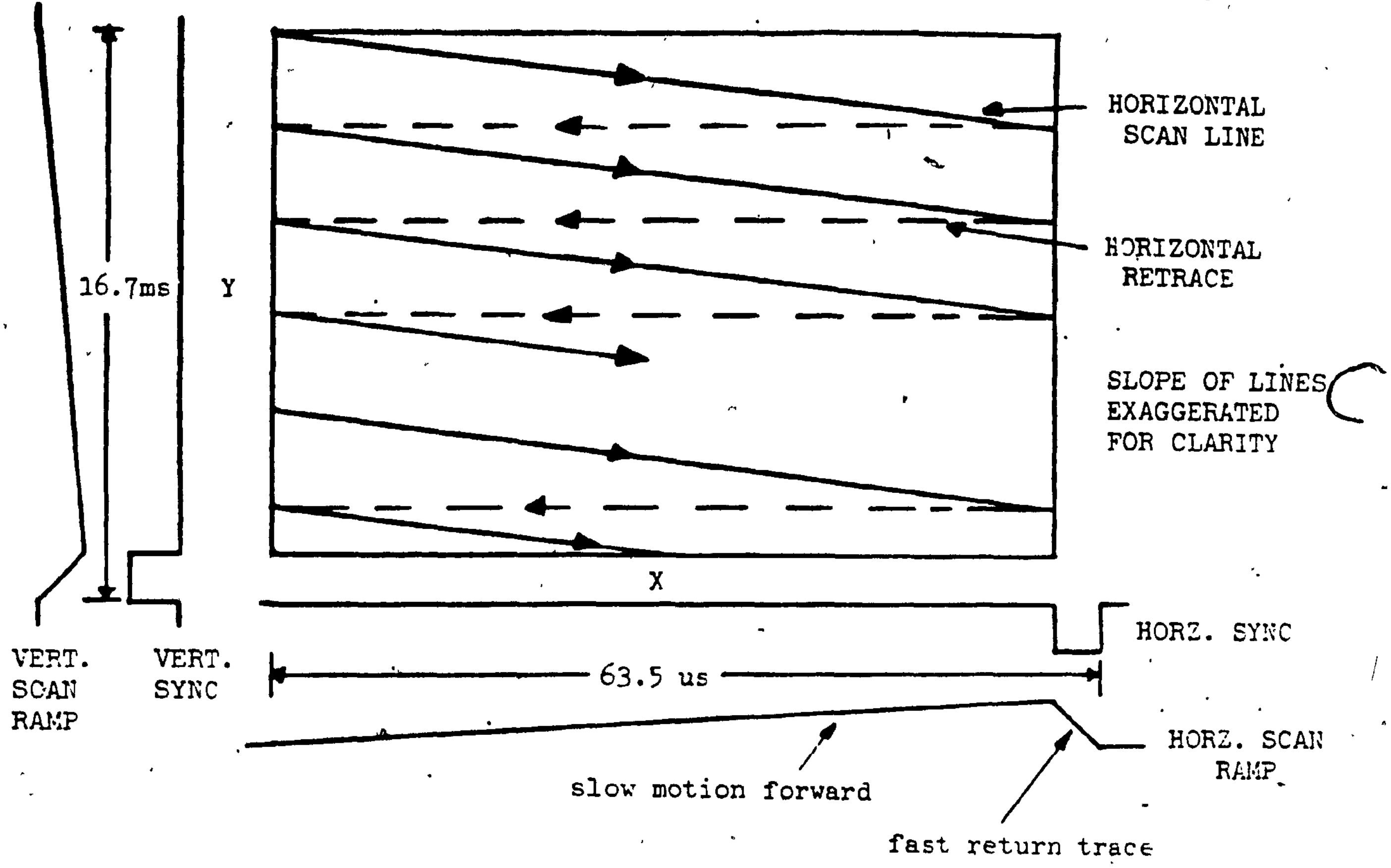
Control of Deflection Direction

Whichever method is used, the deflection system may be thought of as an X-Y coordinate plot, with the horizontal being the X direction, and the vertical being the Y direction (see **Figure 2.1b**). The two variables (X and Y) may be controlled independently by two methods - plotting and sweeping - yielding three distinctly different display systems:

Plot X - Plot Y In this method, known as stroke writing, a plot of X and Y is performed to produce a vector display. The beam is moved and illuminated only for the screen positions necessary to form the desired display. The CRT used is a storage-type CRT, in which the phosphor material has a long persistence allowing the beam trace to remain visible after the deflection operation.

Sweep X - Plot Y In this method, used extensively in lab oscilloscopes, a variable time-base generator is used to repetitively move the beam from left to right across the screen (Sweep X). The beam is moved and illuminated in the Y direction by the vertical amplifier circuitry, which responds to external voltages (Plot Y), to produce a time-varying voltage waveform display. Both non-storage and storage-type CRT's are utilized.

Sweep X-Sweep Y This method, known as **raster scan** display, is used exclusively in broadcast TV and in most video



A sawtooth current, as illustrated here, when passed through a set of horizontal deflecting coils, will cause the electron beam to move from left to right and to the left again.

FIGURE 2.2
Raster Scan Display

terminals. The beam is moved at a constant rate from left to right by the horizontal sweep circuitry and from top to bottom by the vertical sweep circuitry, Figure 2.2. Since the raster scan rate is fixed, the display is rewritten (refreshed) at a periodic rate equal to the vertical scan frequency, eliminating the need for a storage-type CRT. The horizontal and vertical scan rates are selected to provide refreshing often enough to prevent any visually annoying flicker in the display.

2.4 Principles of Scanning

In the NTSC raster scan video display, the electron beam is started in the upper lefthand corner of the screen and deflected horizontally to the right (assuming the observer is facing the screen of the CRT) at a frequency of 15,750 Hz (63.5 us).[8] When the beam reaches the right side of the screen, a horizontal sync pulse occurs and the beam is returned (retraced) very quickly (approx. 5 us) to the left side of the screen (see **Figure 2.2**). During its motion from right to left, no picture information is transmitted; this is called **blanking** and its purpose is to prevent the electron beam from reaching the fluorescent screen. At the same time, the beam is also being deflected vertically from the top of the screen to the bottom at a frequency of 60 Hz (16.7 ms). When the beam reaches the bottom righthand corner of the screen, a vertical sync pulse occurs, and the beam is retraced very quickly (approx. 1 ms) to the top of the screen. Since the beam is being deflected in two directions simultaneously, each horizontal scan line is actually moving at an unnoticeable downward slope to the right (see **Figure 2.2**). The horizontal deflection occurs much more rapidly than the vertical deflection, allowing 262.5 (15,750 / 60) horizontal lines to be scanned within each vertical field. As the electron beam is moved across the screen, its intensity is varied in proportion to the picture information seen by the TV camera.

Flicker

The scanning frequencies used in NTSC TV were selected to present a flicker-free image with the appearance of smooth, continuous motion, within the allotted transmitting bandwidth.[5] In order to provide motion, the entire screen must be scanned within 1/20 sec (50 ms) or faster. This is the minimum time required for effective persistence of vision, which is the eye's ability to retain an image after it has been removed from direct view. The TV frame rate of 1/30 sec (33.4 ms) easily meets this requirement. However, to prevent flicker, the screen must be illuminated at least twice the 1/20 sec rate. This could be done by simply transmitting all 525 lines (the amount of lines necessary for good picture resolution) at twice the frame rate of 1/60 sec (16.7 ms), but this would require a transmitting bandwidth much higher than the allotted 6 MHz. The method used to overcome this problem is called **interlaced scanning**, in which two fields (even-line field and odd-line field) of video information are transmitted at 1/30 sec, with every other field started 1/2 horizontal line later than the one before it. The two fields are interlaced at a 2 to 1 rate to produce one complete picture frame. Since the picture content changes at a rate slower than the field time (16.7 ms), the two fields appear to be seen simultaneously. The net effect is a frame every 33.4 ms, which is a composite of two 262.5-line fields and which can be transmitted within the 6 MHz channel bandwidth, resulting in a total screen resolution of 525 lines (see **Figures 2.3a and 2.3b**). Note that 525 times 30 results in the horizontal scanning frequency 15,750 Hz used in North America.

The Video Signal

To work effectively, the horizontal and vertical deflection system must be precisely controlled by the signal sent from the broadcast TV transmitter. This is done by adding synchronization information to the picture information. The horizontal and vertical sync pulses which are transmitted keep the TV set locked to the exact scene that the camera sees. **Figure 2.4** shows the horizontal and vertical sync pulse timing

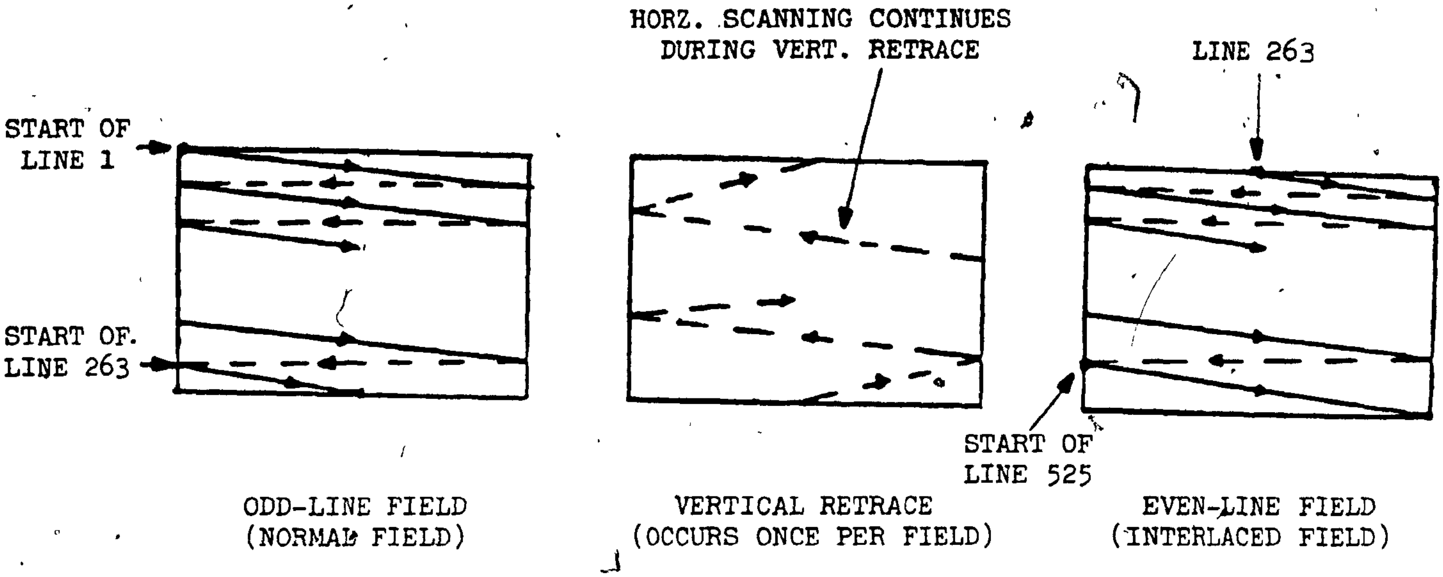

HORZ. SCANNING CONTINUES DURING VERT. RETRACE
LINE 263
START OF LINE 1
START OF. LINE 263
START OF LINE 525
ODD-LINE FIELD (NORMAL FIELD)
VERTICAL RETRACE (OCCURS ONCE PER FIELD)
EVEN-LINE FIELD (INTERLACED FIELD)

FIGURE 2.3a

2 to 1 Odd Line Interlace Scanning

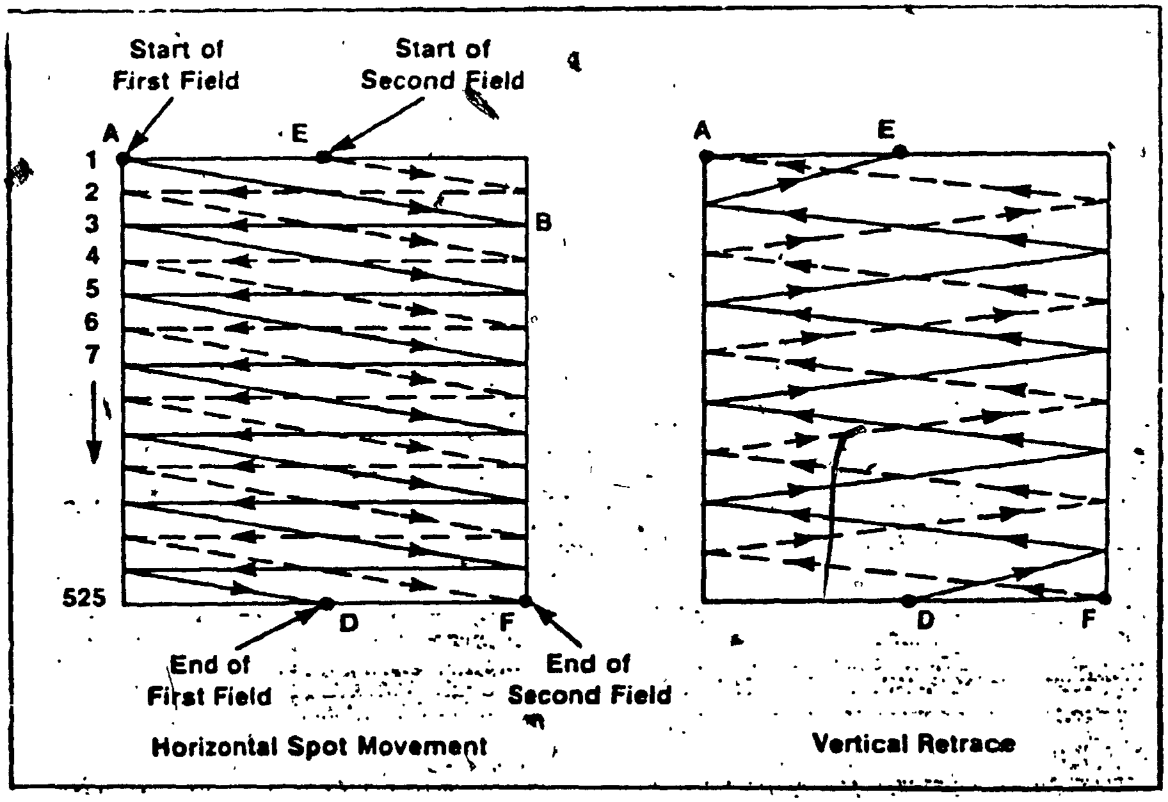
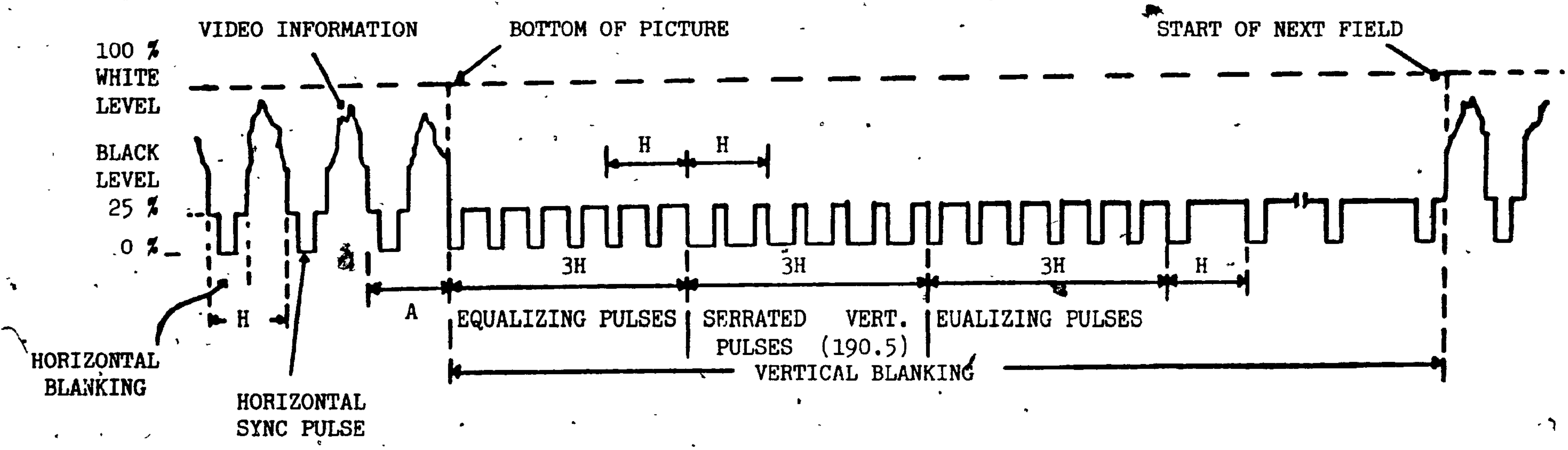


FIGURE 2.3b

Typical TV interlaced raster scanning pattern taking place during one frame.[7]



NOTE: A = H FOR FULL SCAN LINE
A = 0.5H FOR HALF SCAN LINE
H = HORIZONTAL SCAN LINE, INCLUDING RETRACE (63.5 us)

FIGURE 2.4

TV Sync Timing [5]

relationship. The horizontal pulses occur at the end of each line and the vertical pulses occur at the end of each field. Although the vertical sync that is produced during the vertical blanking period is composed of shorter pulses at twice the horizontal rate, it is actually converted in the TV chassis to a single vertical sync pulse with a duration of 190.5 us and a repetition rate of 16.7 ms. The equalizing pulses and serrated vertical pulses are used to maintain horizontal sync during the vertical retrace time. Also, during both the horizontal and vertical blanking time, the video information is held at the black level to prevent interference with normal picture quality. Although **Figure 2.4** shows the two-to-one interlace relationship (A), it is not absolutely necessary for video terminal applications and is often omitted.

Since the horizontal and vertical sync are transmitted together, some facility must be provided in the TV set to distinguish between the two. This operation is performed by the sync separator section of the T.V. The two basic circuits used are the differentiator and the integrator. The differentiator responds to sync clock-edges at the horizontal timing frequency and triggers an oscillator that is free-running near the 15,750 Hz rate. The integrator responds to the sync pulse width at the vertical timing frequency and triggers an oscillator that is free-running near the 60 Hz rate. The normal horizontal sync pulses are ignored by the vertical integrator circuit because their comparatively low duty cycle cannot charge the integrator to the vertical oscillator trigger point. During vertical retrace, however, the serrated vertical pulses, which are essentially double-frequency inverted horizontal pulses, provide a much higher duty cycle that allows the vertical integrator to charge to the necessary trigger level. Because the equalizing and serrated pulses are at a multiple (2H) of the horizontal scanning frequency, the horizontal differentiator continues to respond to these pulses, maintaining horizontal sync during vertical retrace. After the horizontal and vertical sync pulses are separated they are sent to their respective drive circuits to control the CRT deflection yokes.

The video information is also separated at this time and sent to the video amplifiers and CRT drive circuits. The scene, as it appears to the TV camera, can now be displayed in a synchronized time relation to the viewer. **Figure 2.5** shows a simplified block diagram of a typical black and white TV chassis. The information presented thus far has followed a route from the CRT to point B in the diagram. The remaining circuitry is used to amplify the RF antenna signal, select the desired channel, and recover the video, sync, and sound information from the modulated RF carrier frequency.

It is at point B that many CRT controller circuits used in video terminal applications are connected. The composite video and sync signal produced by the CRT controller is designed to interface at a standard video input level as shown in **Figure 2.6**. However, some CRT controllers interface earlier or later in the block diagram signal path. In the earlier path, ~~the~~ composite video and sync signal from the CRT controller is used to drive an RF modulator circuit, which produces a standard output that contains a modulated Radio Frequency signal with a carrier frequency

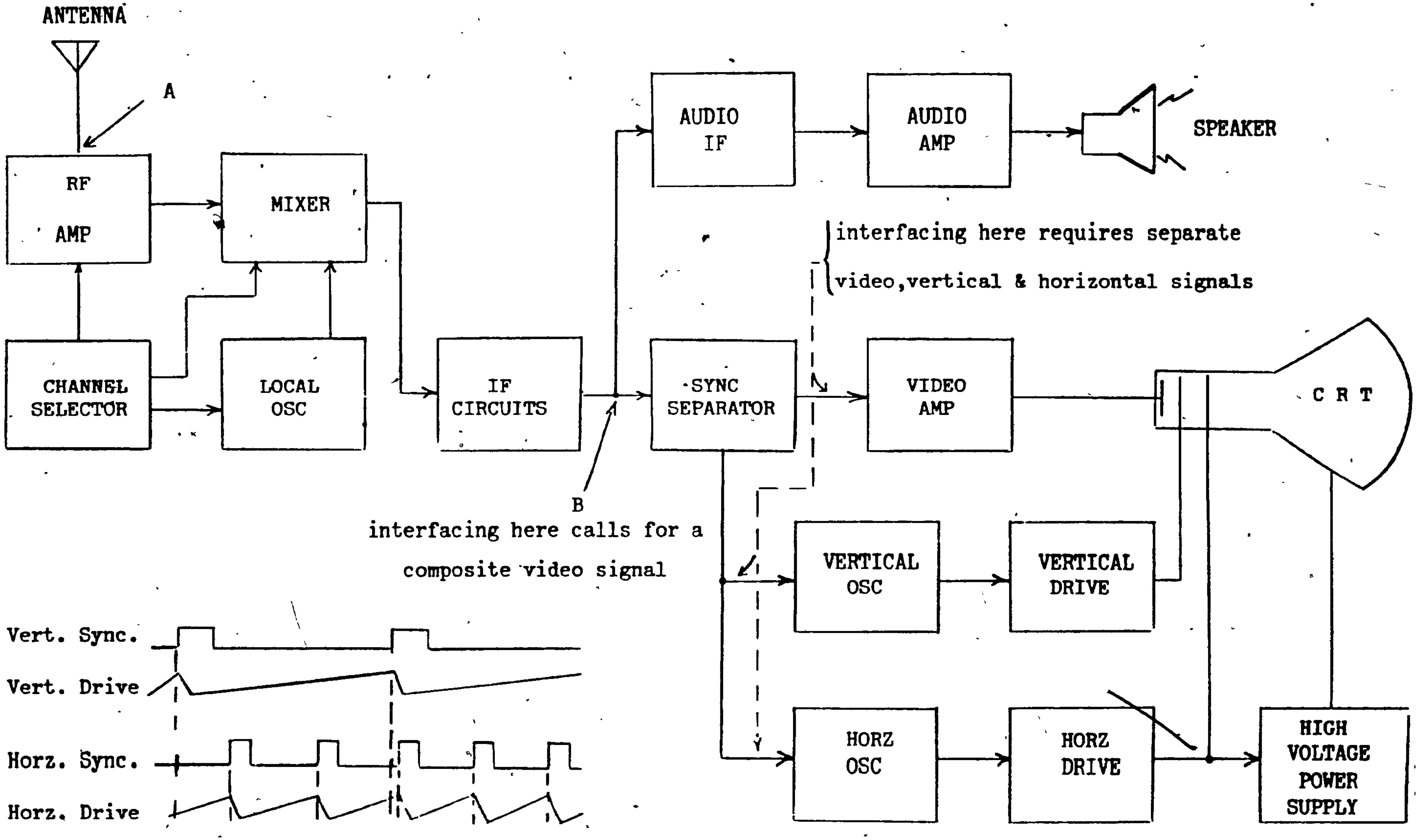


FIGURE 2.5

Simplified TV Receiver Block Diagram [5]

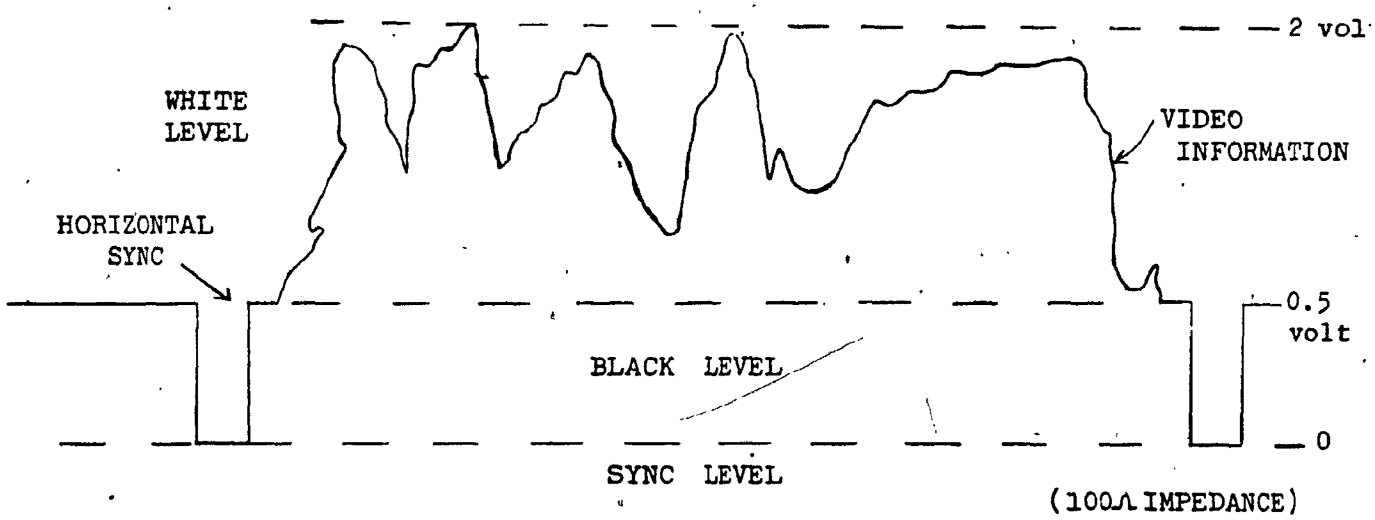


FIGURE 2.6

Standard Video Signal Levels

equal to the TV channel picture carrier (i.e., Channel 2 = 55.25 MHz). This signal is essentially the same as the signal produced by the broadcast TV transmitter and may be connected directly to a standard TV at the antenna terminal inputs. This method provides a simple interface, but requires the use of the TV IF and RF sections shown in **Figure 2.5.** The signal bandwidth is also limited by the RF section and the 4.5 MHz sound section, preventing more than approximately 40 characters from being displayed horizontally on the screen. Since these signal limiting sections are by-passed using the composite video method, which is connected at point B, higher resolution displays are possible. This method may require minor modifications to a standard TV chassis to interface at point B, although some commercial models are supplied with an external video input for use in monitor applications. Other commercial models are available strictly for monitor use, with the RF and IF sections omitted, offering higher resolution display capabilities.

Standard TV chassis circuitry may be further reduced by interfacing still later in the signal path by using separate horizontal, vertical, and video signals connected directly to the appropriate TV drive circuits. This method is most often used for high resolution color monitors and the drive circuits required often use non-standard signals, making interfacing more difficult, thereby limiting the CRT controller to a particular TV chassis type.

2.5 Summary

Obviously, many options are available to both the CRT controller designer and user. Of the many controller circuits on the market, few are directly interchangeable. Some are intended for low resolution and simple interfacing, while others are quite complex, in both circuitry and display capability. The user will find it necessary to decide on a CRT controller circuit, based on such factors as display density, ease of interfacing, additional IC circuitry required, cost, flexibility, and second-sourcing. However, since the broadcast TV standards, and subsequently the video terminal requirements, have long been established, new application design pains are somewhat eased. In some cases, the low-cost, production volume and time-tested equipment produced by the broadcast TV industry is used to advantage by the video terminal industry, to the point, in fact, of using standard TV chassis as video display units. However, in the case of most CRT terminal manufacturers, their video display units are designed to conform to the specifications of each particular model.

CHAPTER 3

CRT CONTROLLER FUNDAMENTALS

The widespread and ever-increasing use of CRTs in computer terminals has led semiconductor manufacturers to design LSI devices to simplify and reduce the costs of control functions for CRT-based terminals. In this chapter we will take a look at the basic building blocks making up a CRT controller (CRTC), as well as the terminology and definitions which apply in general to all of the CRTCs we will consider later on. We will also consider the circuitry necessary to interface directly with any CRTC, regardless of the approach it uses to communicate with the central processor and the screen memory.

3.1 Introduction

The CRTC is an LSI controller which is designed to provide an interface for microprocessors to raster scan type CRT displays. Most CRT controllers available off-the-shelf today belong to a particular family of devices, depending upon the IC semiconductor manufacturer who makes them, and are thus designed to have direct signal compatability with the family's microprocessor(s) and/or microcomputer(s) to provide ease of interfacing. Some of the CRTCs may be easily interfaced to other processors which don't belong to their family, at the expense of using extra hardware in order to obtain the correct interface signals.

The CRTC's primary function is to generate timing signals which are necessary for raster scan type CRT displays according to the specifications programmed in the CPU's ROM. In order to incorporate versatility into such devices, most CRTCs are designed as programmable controllers. This allows applicability to a wide-range of CRT display formats; from small, low-function character displays to raster type full graphic displays, as well as large high-function graphic displays.

3.2 Basic Building Blocks Making Up a CRTC

Most CRTCs nowadays, replace the equivalent of from 30 to 40 or more MSI discrete ICs making up the CRT controller functional block of the first and second generation terminals. **Figure 3.1** shows a general block diagram of the circuitry making up such a discrete CRT controller. The basic building blocks are as follows:

1. The Master Dot oscillator and timing counters.
2. Valid position decoders and address counters.
3. Cursor control and scrolling.
4. Video memory and character dot decoder.
5. Video generation circuitry.

As will be shown later, most of the circuitry in **Figure 3.1** has been replaced by an LSI chip, the CRTC. Mainly, the circuitry making up sections 1, 2 and 3, nowadays resides within most CRTCs with minor modifications from IC semiconductor manufacturer to manufacturer. Not only that, but programmability has been incorporated as well in such devices, giving rise to highly versatile, low power, low cost and highly reliable controllers.

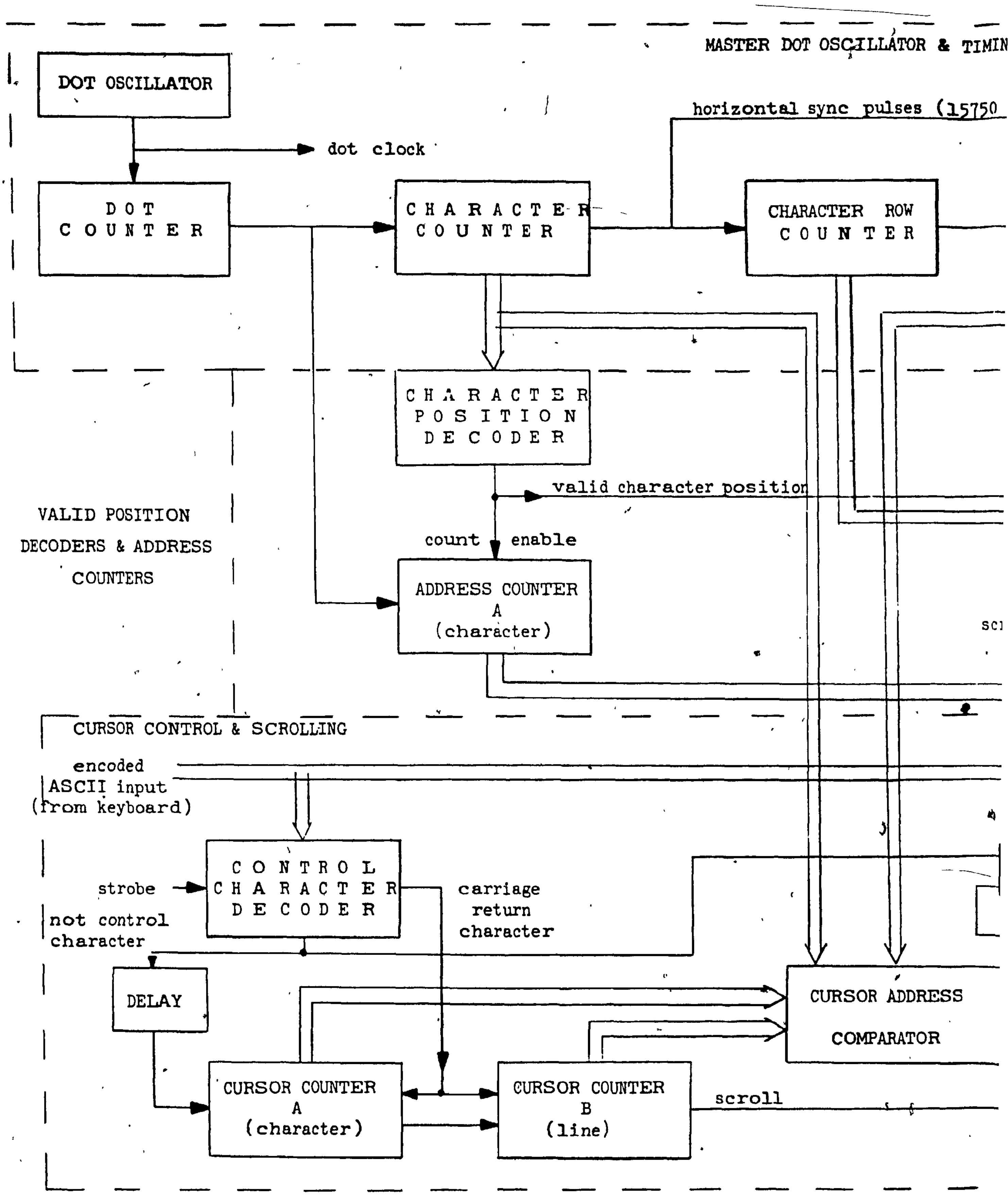
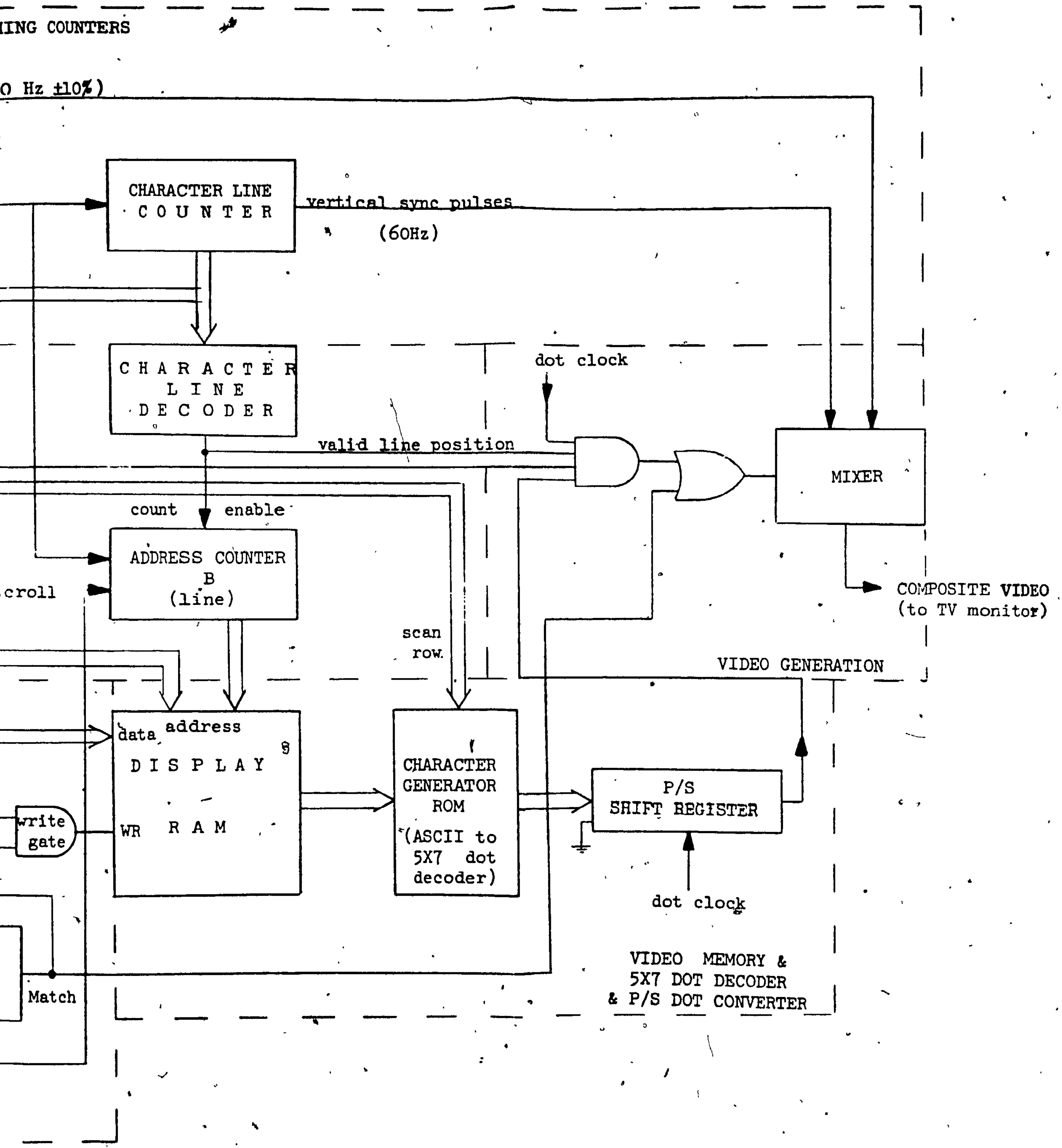

MASTER DOT OSCILLATOR & TIMIN
DOT OSCILLATOR
dot clock
horizontal sync pulses (15750
DOT COUNTER
CHARACTER COUNTER
CHARACTER ROW COUNTER
CHARACTER POSITION DECODER
valid character position
VALID POSITION DECODERS & ADDRESS COUNTERS
count enable
ADDRESS COUNTER A (character)
sc
CURSOR CONTROL & SCROLLING
encoded ASCII input (from keyboard)
strobe
CONTROL CHARACTER DECODER
carriage return character
not control character
DELAY
CURSOR ADDRESS COMPARATOR
CURSOR COUNTER A (character)
CURSOR COUNTER B (line)
scroll

FIGURE 3.1
General block diagram of a CRT controller made u

1 OF



up of discrete logic [9]

2 of 2

3.3 CRT Interface Signals - Clarification

Although chapter 2 clearly pointed out the CRT signal standards adopted by the TV industry before we go any further, when it comes to interfacing to a TV or monitor several points require further consideration and clarification. Keeping them in mind will allow the designer to compromise whenever difficulties arise, as well as recommend appropriate equipment purchase. These points are as follows:

1. Horizontal Sync — Although the standard horizontal sweep frequency is 15.75 KHz (63.5 usec/line), this can vary by as much as ±10% for most display systems without degrading picture quality and synchronization.

2. Vertical Sync — Vertical sweep can vary between 45 to 65 Hz, however, in many CRT display systems, the power supplies are poorly filtered and shielded, which causes some line-frequency modulation of the raster. Ideally, the sweep rate should be equal to the power-line frequency for the best display. If the vertical sweep isn't synchronized, the raster will appear to move at the difference of the two frequencies, i.e., the **beat frequency.** However, if the two frequencies are equal, no beat frequency results, preventing modulation of the raster.

3. Interlace/ Non-Interlace (Higher Horizontal Sync) — For 60 Hz line frequencies, there are 262.5 lines per complete raster scan. However, in many applications such a line count does not provide enough resolution. A possible solution is to use

interlacing - inserting a second set of lines between the first set. Unfortunately, the line sets are not generated simultaneously. As a result, several disadvantages take place when using this technique:

(i) the circuitry necessary for scanning is more complex than in non-interlaced display circuits;

(ii) the overall vertical refresh rate drops to half that of non-interlaced units; as a result, the display can flicker when you use a CRT monitor with standard P4 phosphor (a fast reacting phosphor - low persistence). For best viewing, in this case, a P31, P33 or P39 high-persistence phosphor CRT should be used.

Another way to get good resolution, without interlacing, is to use a CRT system that operates at a **higher horizontal sweep frequency** to obtain more scan lines per field. However, these systems are more expensive than TV monitors and use non-standard horizontal scan components.

4. Composite Video

It is preferrable to use a single, **composite video** signal and let the circuits inside the monitor separate it into its component parts. The major advantage of a composite video signal is that it can be **sent over long distances** on a single 75Ω coaxial cable.

3.4 Description of CRT Controller Operation

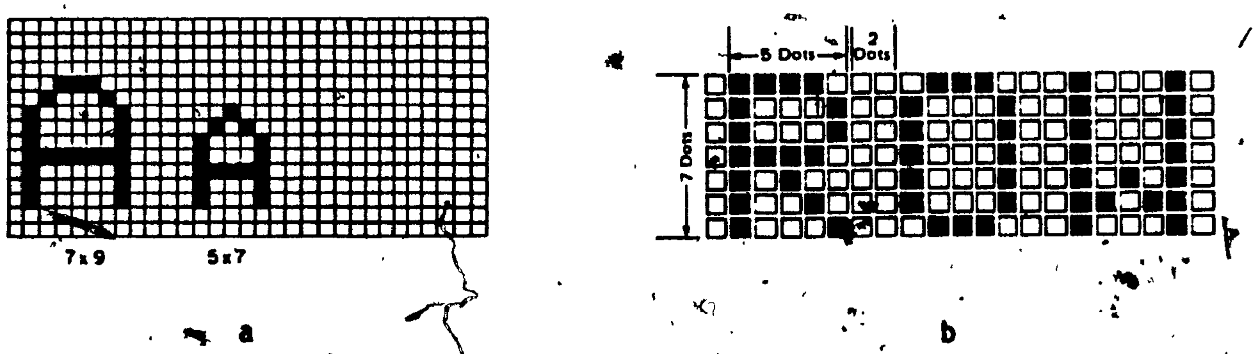
In order to be able to understand how the CRT LSI controller operates, we must first of all comprehend the interaction of the different blocks making up the discrete logic CRT controller, as depicted in **Figure 3.1.** The operation applies to most CRT controllers, whether discrete or integrated, or may vary slightly; however, the principle of the matter is standard.

The main part of any CRT-monitor control system is its **sync generator**, which provides all the sync and timing signals necessary to control the display. To design a sync generator in the most straightforward manner, first the CRT screen is divided into small cells. Then the beam's position is controlled by keeping track of the cell number where it is positioned.

Character Generator Logic

The cells are also handy for generating characters, since each character can be defined as requiring one cell. For **alphanumeric** displays, each cell usually corresponds to **one character position.** In **graphic** displays, a cell can be a **single** dot or a **group** of dots.[10]

In the case of alphanumeric displays, characters are generated by lighting up the proper combination of dots in a 5 x 7 or a 7 x 9 **dot matrix** that makes up the cell, **Figure 3.2a.** However, many other dot matrix combinations are available or may be designed into the system, depending upon system parameters. The patterns for each character are often stored in a character-generator ROM (or EPROM).[14,15,16] For the CRT display, the ROM should be row addressable, not column addressable, as will be shown later. The cell (or each character dot matrix) on the CRT screen must also allow for horizontal and vertical row spacing - usually one (or two) extra row(s) or columns(s) of unlit dots, **Figure 3.2b** - to provide separation between characters, then the cell becomes either 6 x 8 (7 x 9) or 8 x 10 (9 x 11) dots. Some of the ROMs also include the blank lines of dots needed for separation, so the external circuits can be simplified even further, **Figure 3.3**.



Standard 5 x 7 and 7 x 9 dot matrix displays can be generated from ROM (EPROM)-based data.[10]

5 x 7 dot matrix showing separation between characters.[11]

FIGURE 3.2

ROW SELECT TRUTH TABLE

RS3	RS2	RS1	RS0	OUTPUT
0	0	0	0	R0
0	0	0	1	R1
0	0	1	0	R2
0	0	1	1	R3
0	1	0	0	R4
0	1	0	1	R5
0	1	1	0	R6
0	1	1	1	R7
1	0	0	0	R8
1	0	0	1	R9
1	0	1	0	R10
1	0	1	1	R11
1	1	0	0	R12
1	1	0	1	R13
1	1	1	0	R14
1	1	1	1	R15

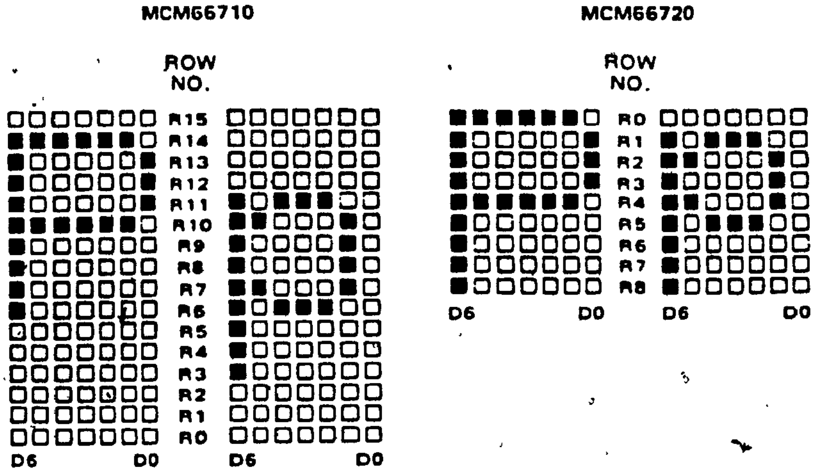


FIGURE 3.3

Motorla Semiconductor MCM66700 series character generator uses 7-bit address code to select 1 of 128 characters in 7 x 9 matrix. 4-bit row select code causes one row of the addressed character to appear at seven output lines. Most ROMs in the series provide shift capability for characters extending below the line, as is the case with MCM66710. Others, such as MCM66720, do not have that capability. Preprogrammed ROMs provide various character sets. The user has the option of obtaining custom programming of the ROM.[13]

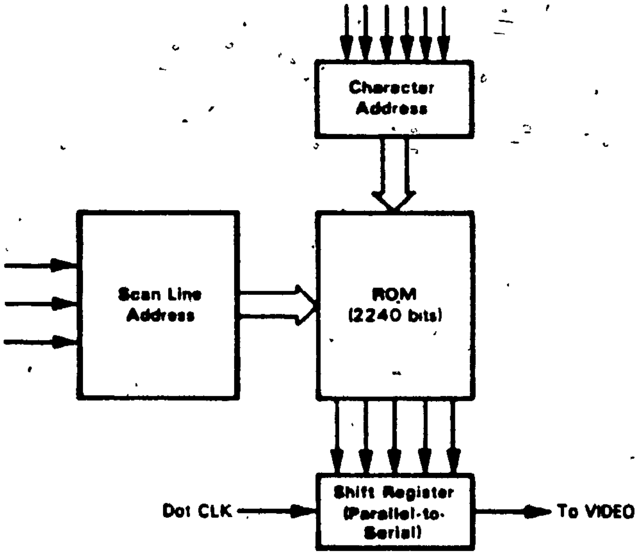


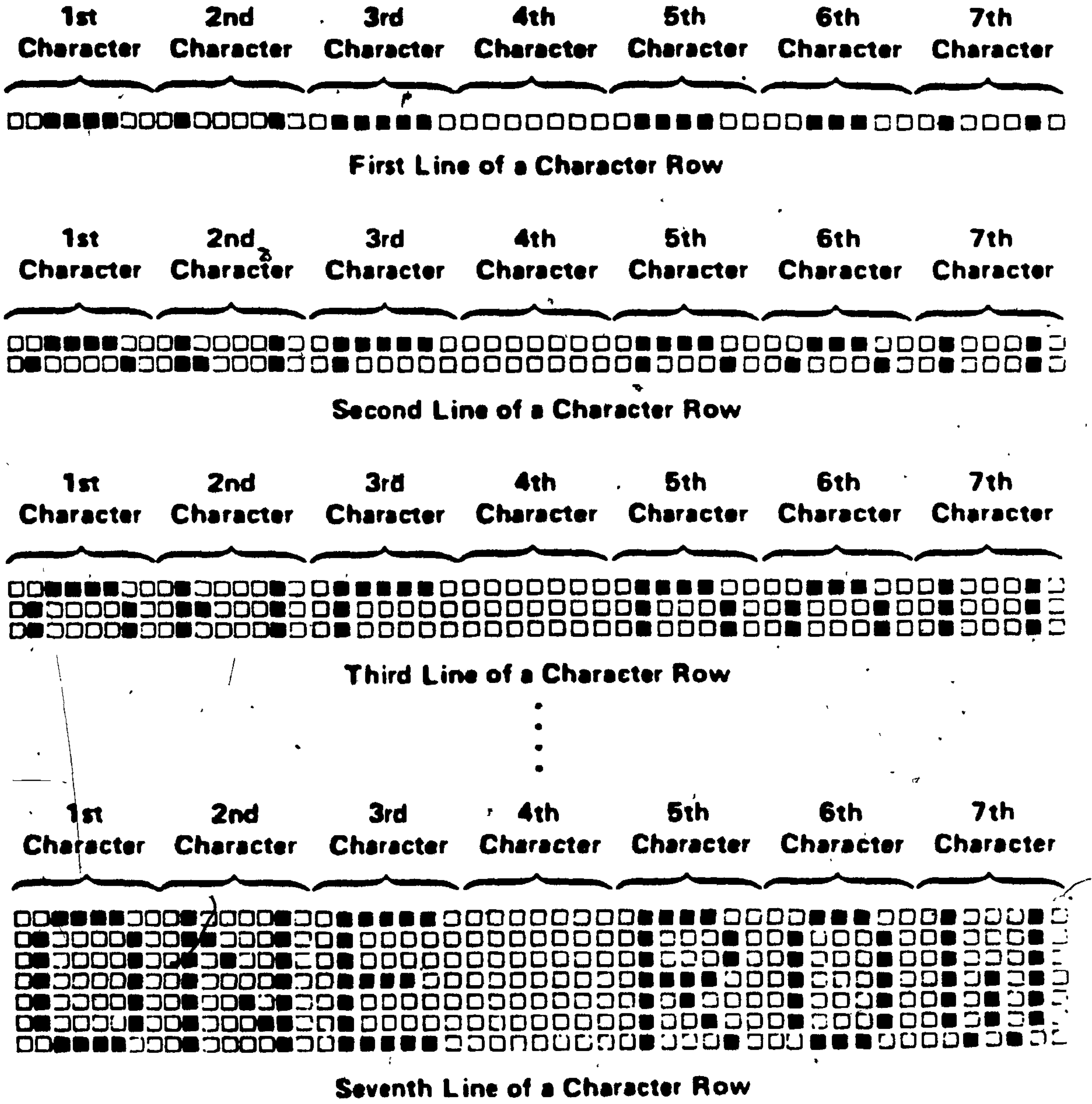
FIGURE 3.4

Typical Character Generator ROM configuration shown with parallel-to-serial shift register to convert the ROM's character data into dot information.[12]

The ROMs require two sets of input addresses - one for the character to be addressed and one for either the row or column of the character to be displayed, **Figure 3.4**. The ROM inputs are usually ASCII-encoded so that a keyboard's output can be fed directly to the ROM.

To visualize how the characters are displayed on the CRT, once again we must consider the scanning process taking place in the monitor. Inside the video display, the character generator receives stable row and character addresses to determine which character is to be displayed. Then the output of the generator is loaded into a shift register and clocked out to form the video signal, **Figure 3.4**. Hence, a master clock is needed to synchronize all the timing, see **Figure 3.1**. It is at this VIDEO input that the dot information making up each character is provided. The purpose of the shift register is to convert data that is normally handled in 8-bit parallel format by the ROM into a serial bit stream needed to create a corresponding data character on the CRT screen. The ROM merely holds the desired character matrix to be displayed on the CRT, and since each character on the screen is represented by many dots, it takes more than 8 bits of information to represent an alphanumeric character - usually it takes 1 byte per scan line making up the character's dot matrix.

Figure 3.5 shows the sequence that takes place when displaying the first seven characters of a single character row of alphanumeric data on a CRT screen. Since the CRT electron beam provides slices of the characters, one row at a time, as it scans a line, first, the dot



First Line of a Character Row

Second Line of a Character Row

Third Line of a Character Row

Seventh Line of a Character Row

FIGURE 3.5

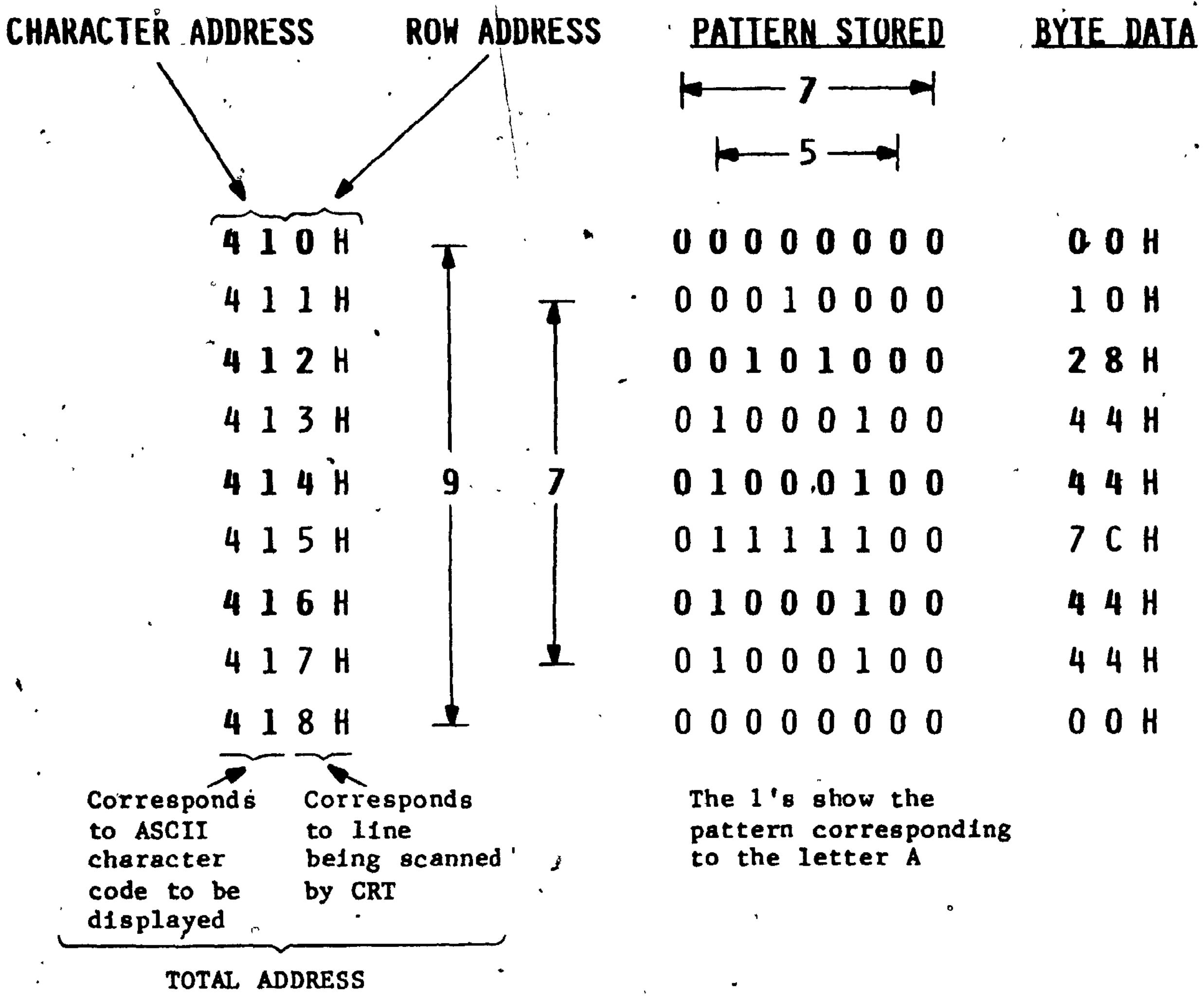
Display of the first seven characters making up a character row of data. The character dot matrix is 5 x 7.[11]

information for scan line 1 is applied via the video input to the CRT. Then the dot pattern for all the characters on scan line 1 are presented consecutively. At the end of the first scan line, the dot information for scan line 2 is presented to the CRT. Such a sequence is repeated until all the scan lines for that character row (depending on the size of the character's dot matrix) have been completed. In our example, **Figure 3.5**, since the character matrix is 5 x 7, the dot information for each character making up this character-line is presented to the CRT a total of seven times. As shown, the dot pattern presented for each character differs for every scan line. Such a phenomena is clearly seen if one examines closely **Figure 3.6**. For every row making up the character matrix of the letter A, there corresponds a particular byte of information pertinent to the row being scanned by the CRT. For every character of every row the outputs of the character generator ROM are transferred to the parallel-to-serial register, and then shifted into the video-generation circuits via the shift register's serial output to provide the relevant character dot pattern to be displayed. Of course, everything happens in synchronism.

Character Generator

If we want to be able to display the entire ASCII code on the CRT, that is, 128 characters, and that we are using a 5 x 7 character dot matrix, we will need a 4480-bit ROM (i.e., 128 x 5 x 7). There are two approaches open to obtain such a device, either:

(a) programming an EPROM with the character format needed;



CHARACTER ADDRESS / ROW ADDRESS	PATTERN STORED	BYTE DATA
4 1 0 H	0 0 0 0 0 0 0 0	0 0 H
4 1 1 H	0 0 0 1 0 0 0 0	1 0 H
4 1 2 H	0 0 1 0 1 0 0 0	2 8 H
4 1 3 H	0 1 0 0 0 1 0 0	4 4 H
4 1 4 H	0 1 0 0 0 1 0 0	4 4 H
4 1 5 H	0 1 1 1 1 1 0 0	7 C H
4 1 6 H	0 1 0 0 0 1 0 0	4 4 H
4 1 7 H	0 1 0 0 0 1 0 0	4 4 H
4 1 8 H	0 0 0 0 0 0 0 0	0 0 H

Corresponds to ASCII character code to be displayed

Corresponds to line being scanned by CRT

TOTAL ADDRESS

The 1's show the pattern corresponding to the letter A

RESULT: "A" ASCII 41H

NOTE 1: The first and last byte of the cell is 00H to provide vertical spacing between characters.

NOTE 2: The cell matrix is actually 7 x 9 dots on a 5 x 7 only visible character matrix.

FIGURE 3.6

Typical information stored on character generator ROM for the letter A (or ASCII 41H).

(b) buying a pre-programmed ROM character generator available from different companies.

There are off-the-shelf character generator devices available which provide the required dot patterns for full ASCII character sets with a 5 x 7, 7 x 9 and other dot matrix formats. One of the specifications worth noting about such devices is that some of them offer descenders for such lower case letters as j, g, q, and y, while others don't. Those that don't are restricted to a smaller dot matrix size such as 5 x 7, hence when such letters are displayed on the CRT they may lead to confusion on the part of the operator. Instead, character generators with larger cell sizes allow such letters to be extended below the baseline (making use of descenders) thus making the text perfectly readable and unambiguous.

The amount of logic included on such devices varies from device to device, some include character address latches, scan line counters and parallel-to-serial shift registers, while others consist only of the ROM with the necessary dot patterns stored in the device.[12]

The Sync Generator

In most CRT terminals designed for the display of alphanumeric data, the relevant CRT display used is of the raster scan type. Such a display has the particularity of utilizing a master clock from which all other timing is derived. To generate such timing, the logic must **decode the appropriate counter states,** see **Figure 3.1,** allowing the derivation of all signals such as the scanning, horizontal and vertical TV synchronization and blanking.

The clock that drives the sync generator and video shift register operates at what is called the **dot frequency**, which can be calculated by:

f_{dot} = f_{line} x number of lines x number of character cells per line x number of horizontal dots per character cell

where: number of lines = number of character lines x number of rows per character cell

The **dot frequency** is fed into a counter that divides by the number of dots per character, **Figure 3.7.** Logic circuits driven by this counter control the flow of data to the shift register delivering the serial video data. The final output of the dot counter drives another counter that keeps track of the **number of characters** on the current line, the **character cell counter.**

Most TV sets and monitors are purposely adjusted to overscan. that is, the picture on the CRT screen is intentionally adjusted to be larger than the screen. The reason for this is to allow the horizontal and vertical sync pulses to take place while the beam is in the overscanned region of the display. In general, such an overscanning is about 20 to 25% more than the total number of displayed characters during the horizontal scanning and the sync pulse is placed in the middle of the retrace interval. Hence, making allowances for the number of displayed characters along with the necessary retrace time, thus an 80 character line should be designed to have about 100 characters. That is, an extra 25% more characters for retrace.

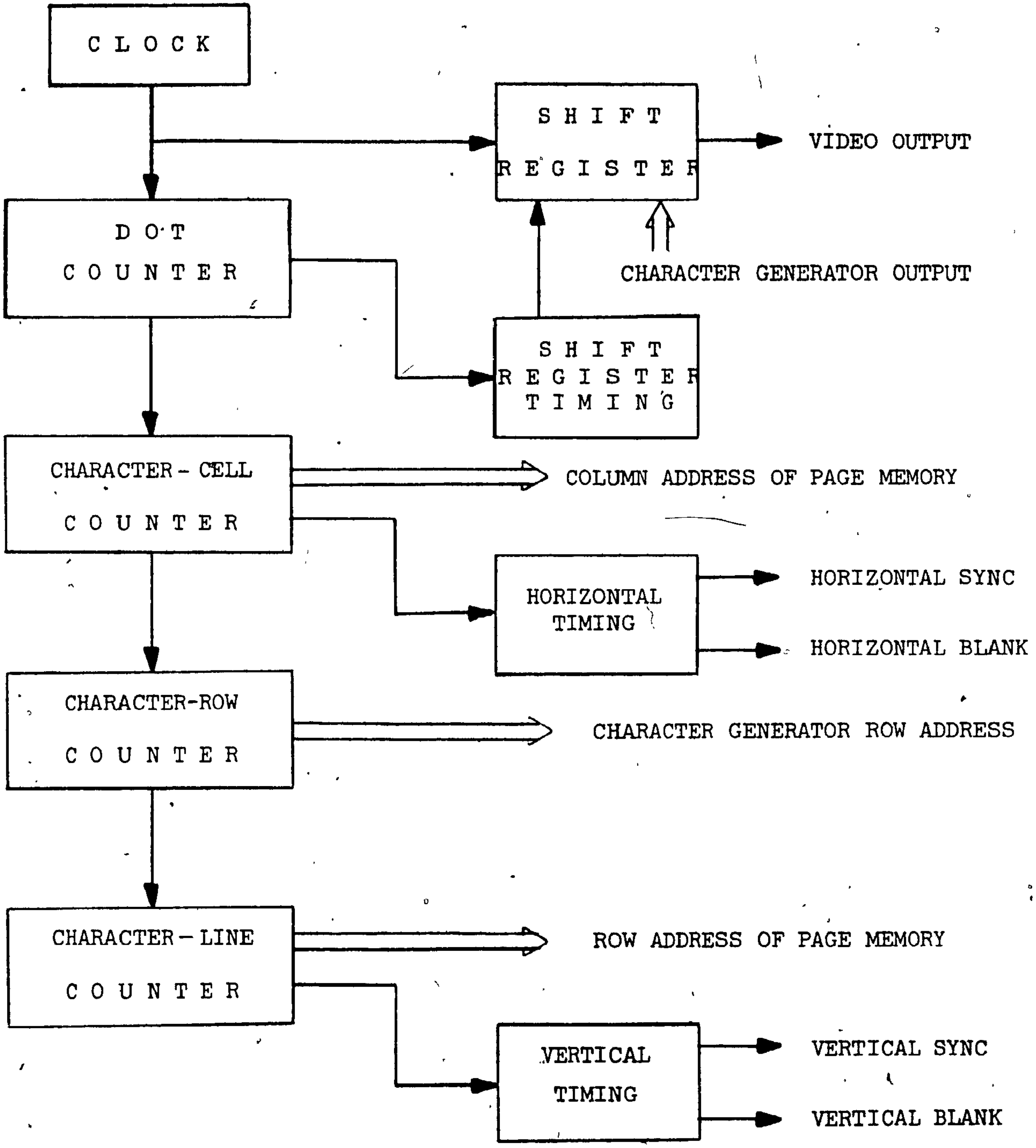


FIGURE 3.7

Inside the sync generator many counters are used to generate all the various timing signals and provide the addresses for the RAM where the displayed data is temporarily stored.

Logic circuits connected to the character-cell counter can generate the necessary sync and blanking signals during the non-display portion of the line, see **Figure 3.7**, and the outputs of the character-cell counter can also be used to drive some of the address lines of the display memory.

The output of the character-cell counter also feeds to another counter that keeps track of the number of lines or rows (i.e., scan lines) displayed for a particular character-cell. This **character-row** counter feeds its output directly to the row-address lines of the character generator.

One more major counter is used in the system - a **character-line counter**. Fed by the final output of the character-row counter, it keeps track of how many character lines have been displayed. To permit enough time for the vertical retrace of the beam, the count value must be about 10% more than actually needed. The logic that generates the vertical sync and blanking signals is controlled by this counter. Additionally, some of the outputs can be used to drive the remaining address lines of the character memory.

Since a CRT doesn't hold data on the screen indefinitely, all video data must be refreshed or rewritten, every 1/30 of a second at least. To hold all the data that must be displayed, a block of fairly fast RAM must be used for temporary storage. Each full screen of data is often referred to as a page, and the minimum storage usually included for a display is one page. However, some terminals that are designed to handle large amounts of data, can often store several pages of data.

Video (Screen or Display) Memory

At the system level, specific regions of the memory map are allocated exclusively to the display. The memory area used for such a purpose is often referred to as "**video RAM**", versus the rest of the RAM known as "system RAM". Although any kind of memory can be used in page storage - the type of memory page used is governed by the type of memory circuit and the method of interface - either static or dynamic RAMs are utilized since they are the easiest memory to use. In the case of dynamic RAMs, the sync generator can carry out the memory refreshing. Thus, in display systems requiring a great deal of memory, the lower cost of dynamic RAMs outweighs their disadvantages. Static RAM interfacing is simple and requires no refreshing.

The sync generator circuit, besides supplying the timing necessary to provide horizontal and vertical sync, also scans the screen's memory page. As the scanning takes place, the data in the screen memory is presented to the character generator, which then sends the appropriate dot stream to the CRT.

On a more detailed basis, let's consider how everything is interrelated. The video RAM organization for a typical interface arrangement between a sync generator and the video page memory is shown in **Figure 3.8**. Considering we wish to display on the CRT the message "VIDEO RAM", as in **Figure 3.9**, the circuitry will operate as follows:

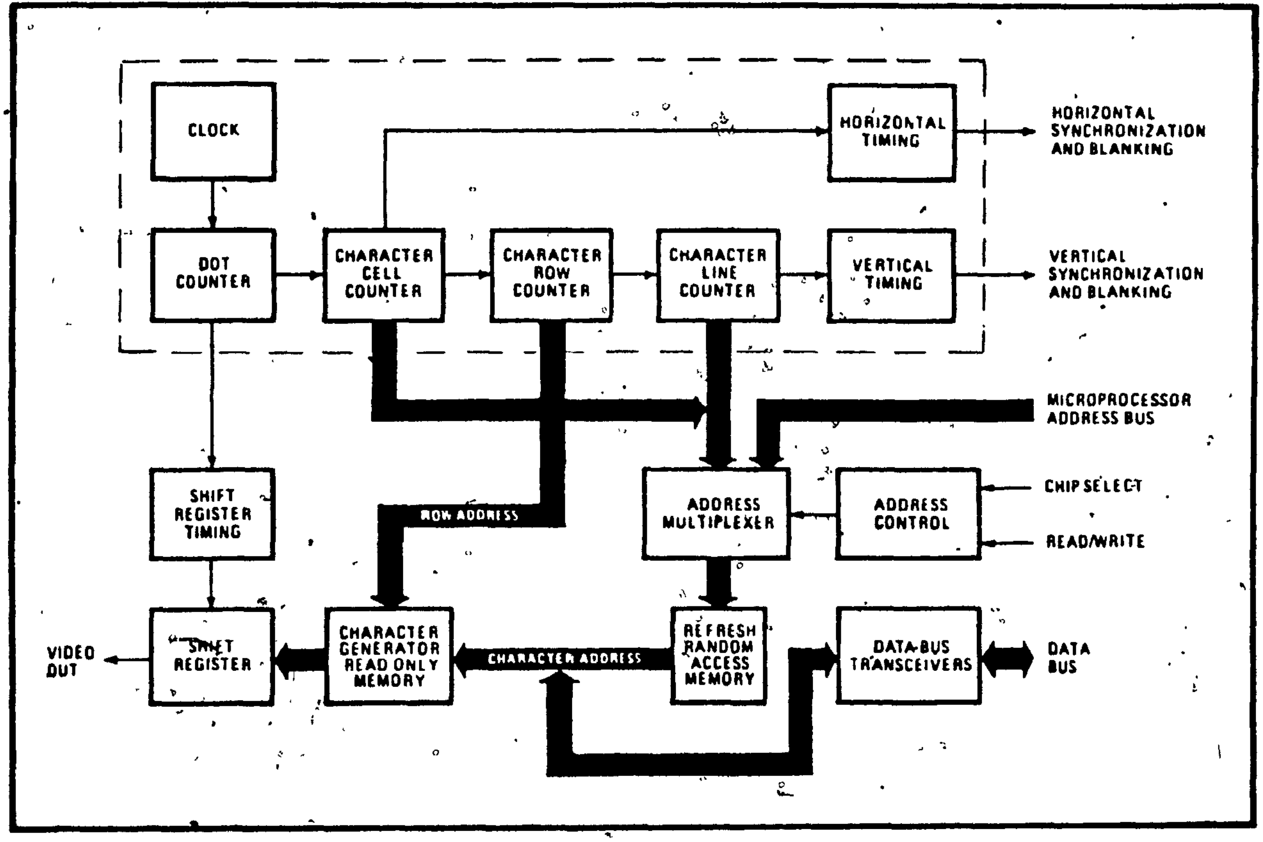


FIGURE 3.8

A video RAM circuit combines the sync generator with all the counting circuits, the character generator and the necessary memory page(s) to form an almost complete video display system. The refresh memory is shared between the sync generator and the uP. Unfortunately, accesses by the uP interrupt refreshing and streaks can be seen on the screen. The sync generator, shown dotted, creates all timing signals. The handshaking address control logic is used to make sure the uP does not contend with the sync generator while the latter is trying to access the video RAM.[17]

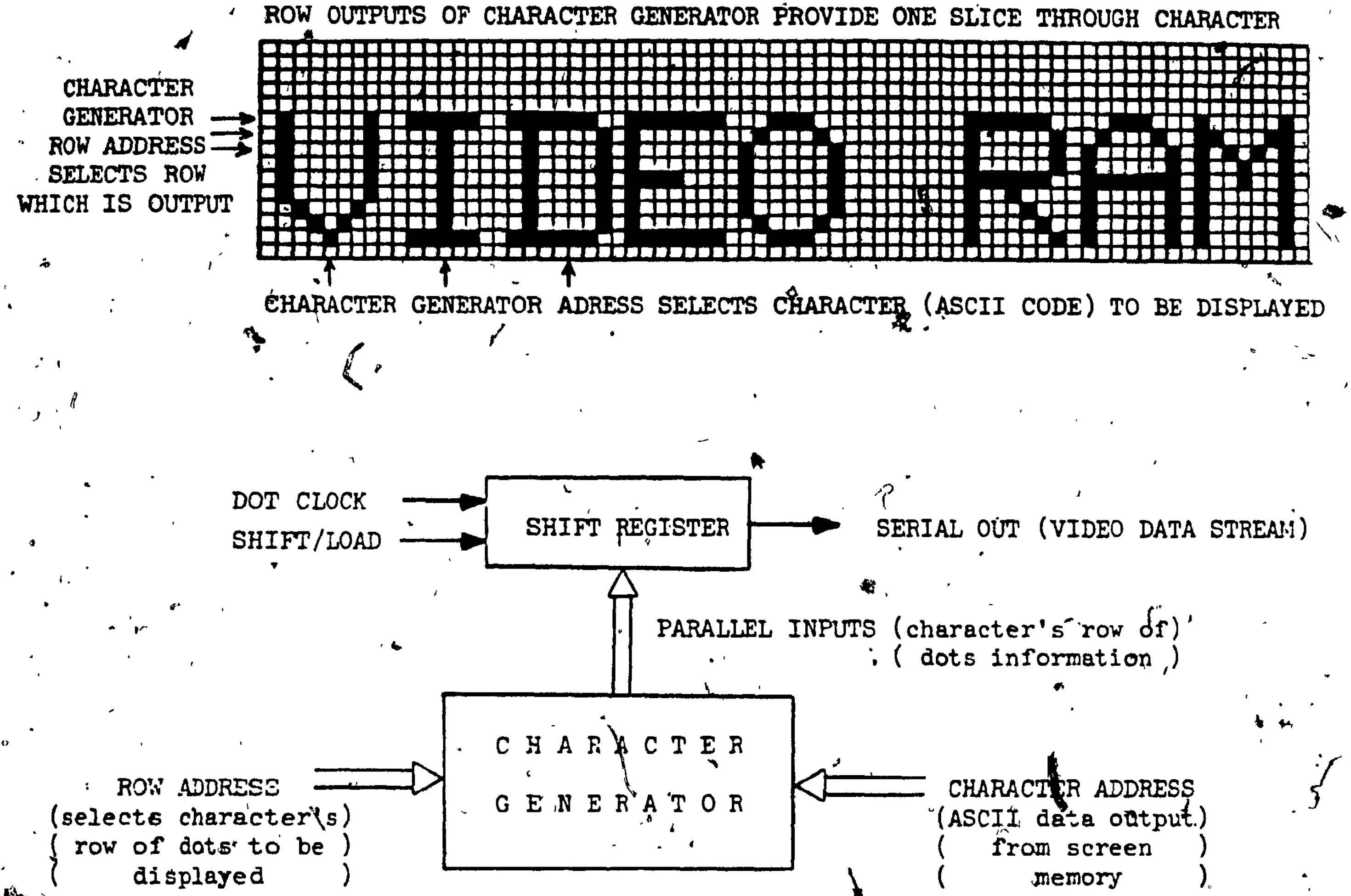

ROW OUTPUTS OF CHARACTER GENERATOR PROVIDE ONE SLICE THROUGH CHARACTER
CHARACTER GENERATOR ROW ADDRESS SELECTS ROW WHICH IS OUTPUT
VIDEO RAM
CHARACTER GENERATOR ADRESS SELECTS CHARACTER (ASCII CODE) TO BE DISPLAYED
DOT CLOCK
SHIFT/LOAD
SHIFT REGISTER
SERIAL OUT (VIDEO DATA STREAM)
PARALLEL INPUTS (character's row of dots information)
CHARACTER GENERATOR
ROW ADDRESS (selects character's row of dots to be displayed)
CHARACTER ADDRESS (ASCII data output from screen memory)

FIGURE 3.9

Standard 5 x 7 and 7 x 9 dot matrix displays can be generated from ROM (EPROM)-based character generators. To actually form the characters, a line-by-line slice is read out from the row-addressable character generator.

Shown above is a 7 x 9 character matrix (with 1 dot separation between characters) on an 8 x 10 dot matrix.

The character's screen image begins with the electron beam, at the top left hand corner of the display, writing a slice through the series of characters making up the "VIDEO RAM" message, as it scans a line. The **column address** of the memory is controlled by the character-cell counter. On the next line, a different slice of the same characters is scanned through since only the address to the **scan (or row) line** inputs of the character-generator ROM has changed. This is due to the **character-cell** counter incrementing each time at the end of the count corresponding to the total number of characters that are to appear on a line. The **character-row** counter incrementation continues until this counter resets and increments the **character-line counter.** That is, since we have a 7 x 9 character matrix within an 8 x 10 dot matrix (**Figure 3.9**), the extra row and column serving for character separation, after the cell line counter reaches a count of 9 (or the tenth count), it resets. Note that the **character-line** counter is interpreted as the **row address** of page memory. Thus, the beam then starts to slice through another set of characters on the new (or next) line. **Table 3.1** shows the sequence of events taking place synchronously while writing the display memory contents on the CRT.

Addressing the Display Memory

Presently, two methods exist for addressing the display memory; either **linearly** or on a **row/column** basis. Each has its advantages and disadvantages.

CHARACTER-LINE COUNTER OUTPUT	CHARACTER COUNTER OUTPUT (DISPLAY RAM COLUMN) ADDRESS	DISPLAY RAM DATA (ASCII CODE OUTPUT)	LINE OF CHARACTER COUNTER OUTPUT (CHARACTER GENERATOR ROW) ADDRESS	CHARACTER GENERATOR OUTPUT	COMMENTS
top → 0	0	V (56 H)*	0 (first scan	10000010	V
line of 0	1	I (49 H)	0 line of	01111100	I
display 0	2	D (44 H)	0 character)	11111100	D
0	3	E (45 H)	0	11111110	← top row E
0	4	O (4F H)	0	00111000	of O
0	5	space (00 H)	0	00000000	letter
0	6	R (52 H)	0	11111100	R
0	7	A (41 H)	0	00111000	A
0	8	M (4D H)	0	10000010	M
0	0	V	1 (second scan	10000010	
0	1	I	1 line of	00010000	
0	2	D	1 character)	01000010	
0	3	E	1	10000000	
0	4	O	1	01000100	
0	5	space	1	00000000	
0	6	R	1	10000010	
0	7	A	1	01000100	
0	8	M	1	11000110	
0	0	V	2	0	
.	.	.	.	.	
.	.	.	.	.	
.	.	.	. .	.	

↑ separation between characters

*ASCII code corresponding to character

TABLE 3.1

Sequence followed by the Sync Generator Logic to display "VIDEO RAM" on the CRT.

Linear Addressing In this case the addressing is done as if all locations on the display were stored in a continuous string of memory locations. To the uP, this is the way the memory is configured. Hence, as the data appears on the CRT, the first character of each row is stored in the memory location immediately following the last character of the previous row, Figure 3.10.[12]

Row/Column Addressing This methods is used when the number of characters on a row equals some power of two. Otherwise, when going from the last character of one row to the first character of the next row, some memory locations will not be accessed and remain useless, Figure 3.10. There are ways around this problem, such as using external logic or ROM to map the addresses generated by the uP (which are continuous) and those generated by the CRT controller for the display memory into the same address space.[12]

Thus, when having to implement text editing functions such as insert/delete line and/or character, it is often most convenient and efficient if the data can be handled on a display line basis. This is where the row/column addressing approach excels, and helps in simplifying the software. However, the price paid is memory inefficiency since some of the display locations are not used. The tradeoffs the designer must make when considering the use of any one of these two techniques are:

1. more complicated software for text manipulation (and thus, more system RAM needed) when using linear addressing; or

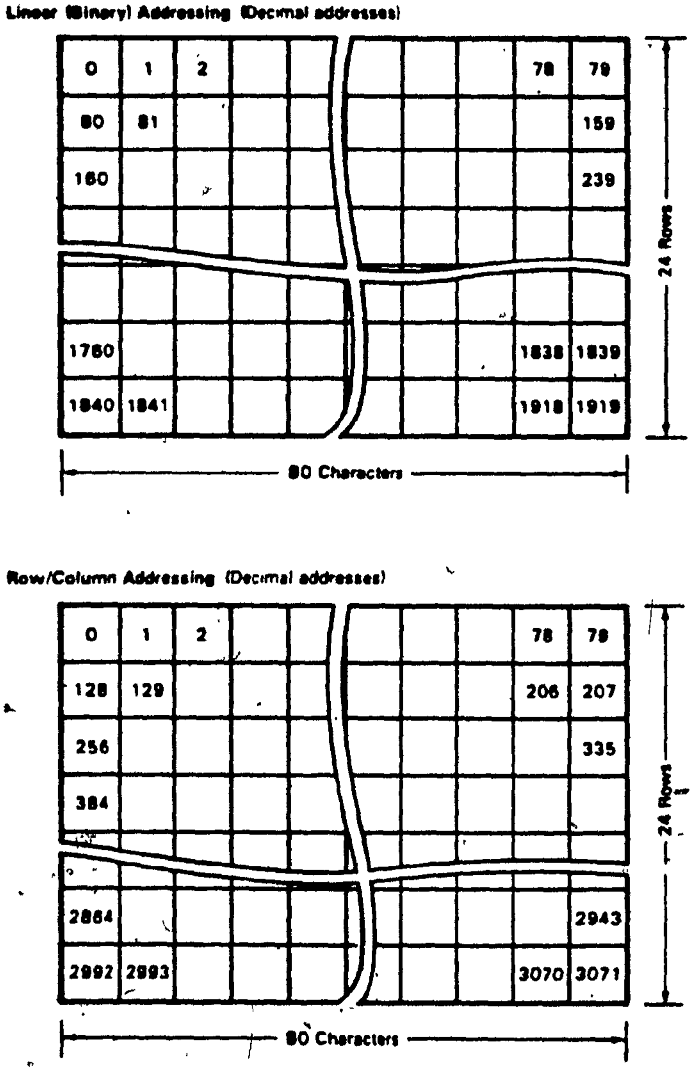


FIGURE 3.10

Linear versus Row/Column Memory Addressing[12]

2. inefficient use of memory space and additional address mapping logic when using row/column addressing.

Interfacing the Memory to the Display

Ideally, the display memory of any CRT terminal system should be accessible to two sources: the sync generator (or CRT controller) and the uP. However, there exists a classic **memory contention** problem common to all CRT displays. The problem arises when the display (refresh) memory must be accessed by both the sync generator (or CRT controller) to refresh the screen and by the CPU to update the screen's information. Which of the two should be given priority? The result is that:

(a) either the CPU's throughput is restricted; or
(b) the screen flickers annoyingly (exhibiting "snow") when the refresh memory is accessed.[10,12,17]

The three most commonly used techniques to solve such a problem are:

1. a video RAM interface;
2. a direct memory access (DMA) interface; or
3. a transparent (or interlace) memory interface.

Let's start by describing the easiest method.

The Video RAM (VRAM) Interface

This approach uses a memory mapped addressing technique to access the video (or refresh) RAM. The CPU and the sync generator (or CRT controller) share the video RAM via a two-input multiplexing circuitry, **Figure 3.8.**

Normally, the sync generator's address outputs control the display memory. However, due to the multiplexer, the address lines of the memory

can thus be switched between two sets of address data, the generator's and the uP's. The switching is controlled by a single address line from the uP that is used much like a chip select line.

The memory input/output bus is connected to the system's data bus via three-state transceivers. They are controlled by the uP's read/write and chip select lines. Thus, by activating the chip select line, the external system can take control of the video RAM. This organization creates a display interface that looks like a RAM to the CPU, hence the name video RAM. Each character position on the screen corresponds to a particular memory location.

The disadvantage: using such an arrangement, the CPU (who has the priority over the sync generator) access to the VRAM, for either a read or write operation, interrupts the normal scanning of the display memory. This results in visible streaks or glitches (normally referred to as "snow") on the CRT screen.[12,17] This problem usually doesn't matter if memory accesses occur in infrequent burst for updates. However, such is not the case usually, since display refreshing is carried out quite frequently on a regular basis. This interference can be eliminated by confining VRAM access to the horizontal and vertical retrace intervals, but this severely restricts the frequency of screen updates.

The DMA Interface

This approach can eliminate altogether the problem of CRT memory scanning since the refresh memory is part of the CPU's system memory,

Figure 3.11. When the display has to access the memory for refreshing the data on the screen, it must stop the uP in order for it to release control of the address and data buses to an external device, called the DMA controller. This device is in turn controlled by the sync generator (or the CRT controller).[12,17]

The sync generator requests data from the system RAM in bursts via the DMA controller. The controller transfers information from the RAM to the CRT display's buffer memory, which can then be read out by the sync generator and put on the screen.

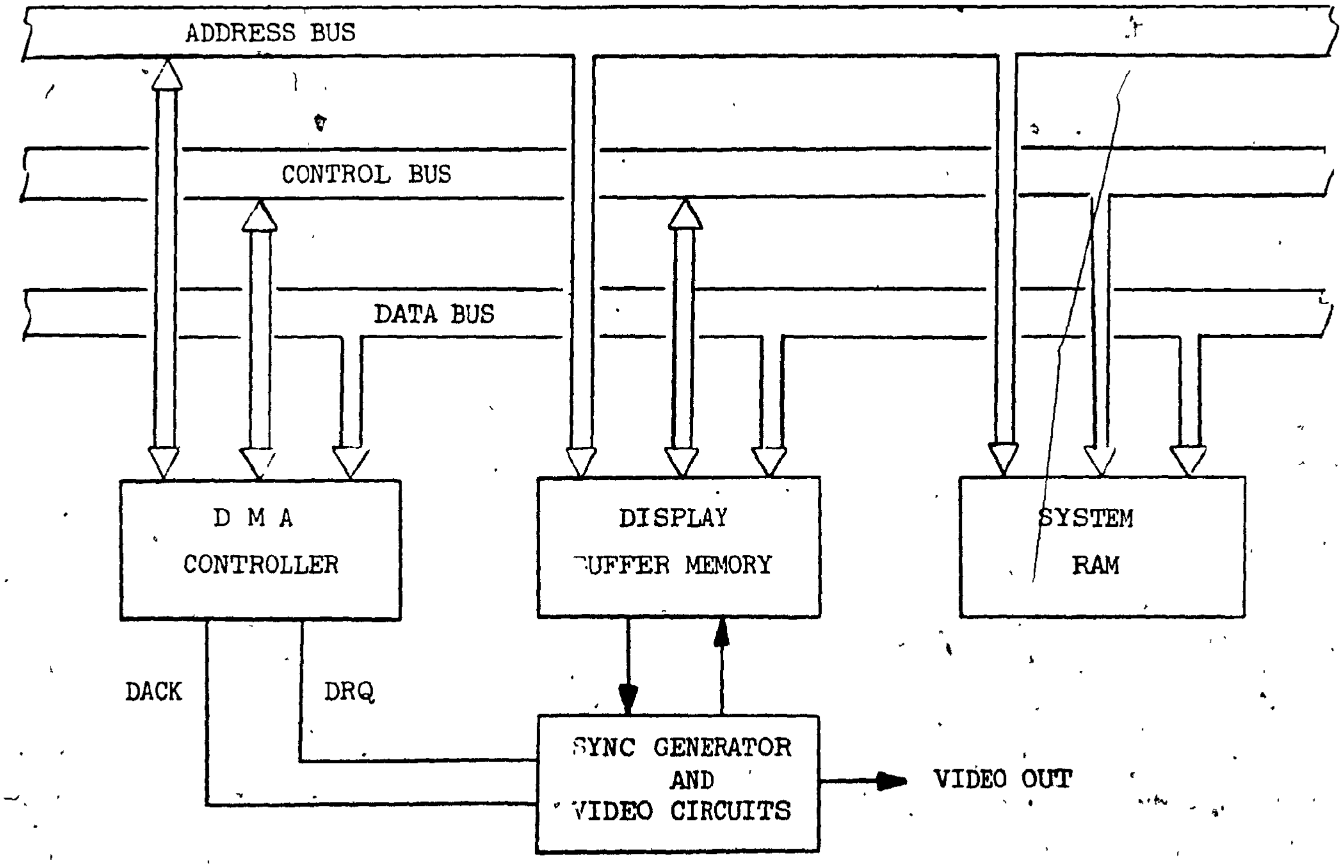


FIGURE 3.11

Direct memory access. The refresh memory is directly addressable by the micrprocessor in the DMA approach. However, the microprocessor is now responsible for refreshing the screen and is slowed down by the frequency with which this must be done.[12,17]

The disadvantage: while this method avoids the interference problem of the video RAM previously mentioned, it is more complex and more expensive to put into use. Also, since screen refreshing occurs frequently, when a DMA occurs, the CPU is forced to stop, thus slowing and cutting down system speed.

The Transparent Memory Interface[12,17]

This approach is similar to the video RAM interface, except that the memory is regularly and systematically made available to both the uP and sync generator, **Figure 3.12**. Access to the memory is still controlled by a multiplexer. However, the multiplexer is alternated between the uP's address bus and the sync generator outputs no matter what. Thus, the memory appears to be transparent to both. The signal controlling the address-multiplexer switching is connected to a square-wave (which acts as a single phase clock) derived from the sync generator dot counter, **Figure 3.13**. Each square-wave period equals one character-cell time, and during the second half of each cycle the RAM is connected to the sync generator address lines. Considering **Figure 3.13**, it can be clearly seen that the RAM speed must be faster than half a cell time. During this same half cell time, the next character to be displayed is transferred from the RAM and stored in the **pipeline latch**. Note that a handshake control block (**Figure 3.13**) helps the processor coordinate its access to the RAM so that it doesn't interfere with the sync generator.

The RAM is connected to the uP address bus during the first half of the character cell. Thus, data transfers from the uP can occur only

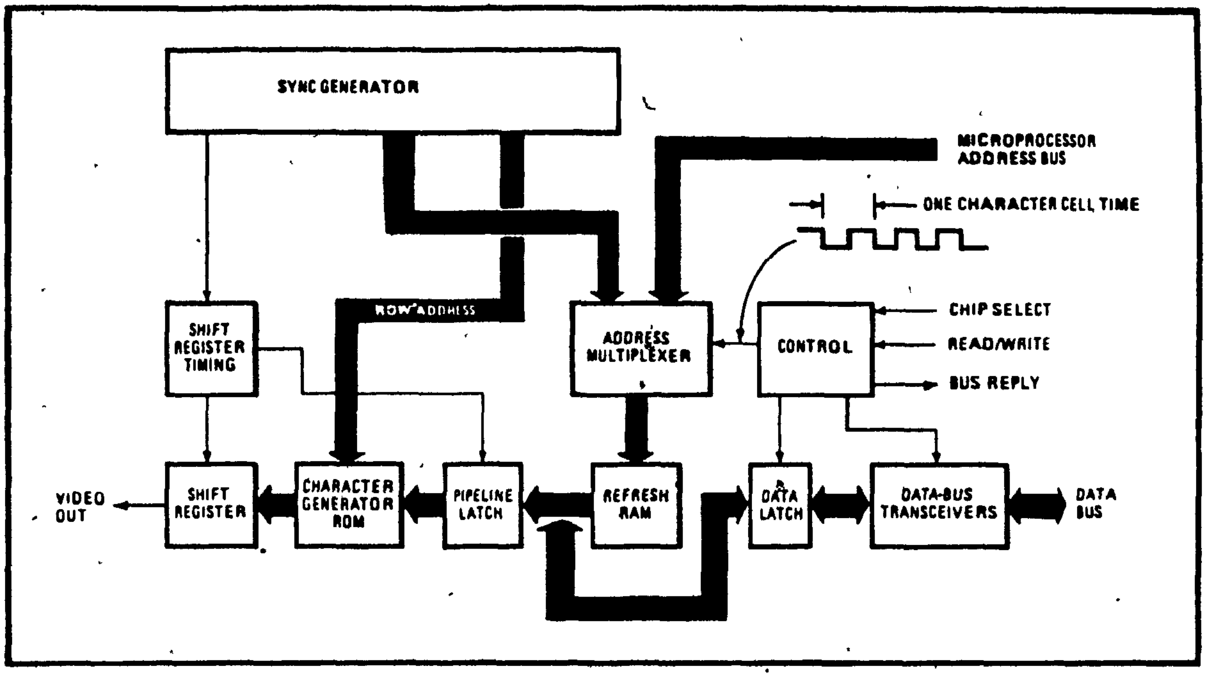
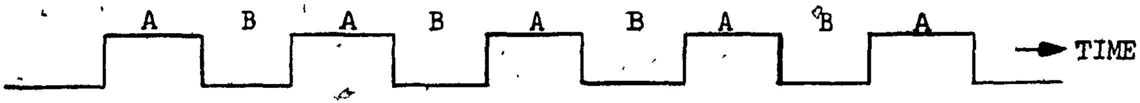


FIGURE 3.12

Transparent memory. The problems inherent in both the video-RAM and the DMA approach are neatly solved with the transparent-memory concept. The screen is never garbled and the CPU can run at top speed. The penalty is cost: a little extra, faster RAM is required.[17]



A – DISPLAY MEMORY AVAILABLE TO THE MICROPROCESSOR

B – DISPLAY MEMORY AVAILABLE TO THE SYNC GENERATOR

FIGURE 3.13

To control the timing all that's needed is a single-phase clock. When High it lets the processor feed into the multiplexer. When Low, it lets the sync generator control the display memory through the multiplexer.

during this time frame. Write operations are straightforward, provided that the uP holds the data stable on the data bus throughout the required half cell time. For read operations, a data latch (see **Figure 3.13**) must be added to retain the memory data longer than the half cell time during which it is available. The only requirement for the uP is that it must have some form of wait or bus reply control line so that data transfers can be synchronized to the next available character-cell time.

Due to its more complex multiplexing, transparent memory requires two to four more MSI TTL packages than does a standard video RAM design. However, a single LSI chip, Motorola's MC6883, Synchronous Address Multiplexer can do the same job. Comparing **Figures 3.8 and 3.12** we can see that, the video RAM interface approach is the basis for the transparent method technique with two extra blocks, the pipeline and data latches. Also, the memory must be twice as fast as that used in the video RAM. In return for these two penalties, this method solves the memory-contention problem.

As will be shown in Chapter 4, among the CRT controllers available off-the-shelf, some lend themselves more easily to one technique of interfacing than another.

Memory Speed Considerations When Interfacing

Memory speed is important especially when a large number of characters are to be displayed per line, since the memory's access time can affect the display performance. The greater the number of characters,

As a direct result, the time available to look up the ASCII-character code in the refresh memory and the character slice in the character generator ROM becomes smaller.

The time available for such functions is known as the **character-cell time**. This is fixed by the time, in microseconds, of a line scan divided by the total number of characters per line. However, regardless of the interfacing technique used, this timing is critical and must be respected to allow proper system synchronism and functioning.

Controlling the Screen Data

All CRT terminals, depending upoh their degree of intelligence, provide the operator with more or less a variety of functions to control the screen data. Among others, the necessary ones are:

Cursor Control: The cursor indicates to the operator where the next character to be entered will be positioned. On the screen, the cursor may appear in one of two forms:

1. as a block-type symbol that may either blink on and off or remain on; or
2. as an underline-type symbol that may blink on and off or remain on.

By direct intervention from the keyboard, the cursor movement may be controlled in at least 4 directions: up, down, right and left. Part of the circuitry making up the sync generator controls the cursor; refer back to **Figure 3.1**.

Scroll Control: Since the CRT screen displays one page of memory at a time, once the bottom line of the screen is reached and filled, the next carriage return will move the text on the screen up, forcing the top row of the page to disappear from view and the new bottom row will be shown empty. This is called **scrolling** and it is carried out on a character-line-to-character-line basis. However, scrolling may take place "smoothly" if it is carried out on a "scan-line-to-scan-line" basis.

The simplest method used to scroll, in order not to burden the CPU, is to keep the address of the screen memory location associated with the first character of the top line in a register. Hence, every time a line is to be scrolled, the contents of this register is incremented by the number of characters per line, thus, pointing to the next line's first character as being the top of the page. Using such a method, no **data movement** actually takes place, but merely the pointer is different.

When scrolling, whether it be for terminals having a single or several page screen memory capability, there is a phenomena known as **wraparound** that takes place. This wraparound concerns the screen memory, and is such that when the last line of the last page is being displayed on the last row and a scroll is performed, the next line to be displayed will correspond to the first line of the screen memory. One can picture this as if the first line and last line of the screen memory were adjacent to each other in a circular fashion, thus, wrapping around.

Incorporated as part of the scroll control is another function [illegible] can scroll on a page basis at the depression of a single key. In this case, to the register containing the address of the top of the page being displayed is added the total number of characters held in one page thus forcing it to point to the next page's top.

CRT Timing Chain Example - Putting It All Together

Before designing an interface between a uP and a sync generator (or CRT controller), the designer must decide upon several factors depending upon system requirements. That is, the type of display format and the drive technique - how many lines, how many characters per line, how many dots per character position and whether interlaced or non-interlaced raster scanning will be used.

In general, a CRT display system is the most flexible display one can interface with a uP. Under program control, the CRT's electron beam can be made to form any character or pattern desired, provided the timing, character generator, CRT monitor, memory speed and interfacing have been carefully chosen. In order to show the adequate procedure to follow when deriving the CRT timing chain parameters, let's consider an example.

Design Parameters: the actual design must start by selecting a screen format and a character generator. Since the 80-character x 24-line format is an industry standard, let's use it in our example. With regards to the character generator, the character-cell will consist of a 6 x 8 character formed in an 8 x 10 dot matrix.

Horizontal Parameters: The HORIZONTAL PERIOD contains both the **displayed** portion of the scan and the **retrace** time. The horizontal synchronization (HSYNC) scan frequency used in North America is 15.75 KHz ±10% (i.e., 63.49 ±6 usec), however, we will use a frequency of 15625 Hz, which is within the allowed tolerance, in order to obtain an even 64 usec duration for each horizontal scan line. Before proceeding any further, two more things must be kept in mind when considering the HSYNC pulse to be generated:

1. it must take place approximately in the middle of the horizontal blanking interval during each horizontal scan line.
2. the horizontal blanking should be approximately 20 to 25% of the horizontal scanning period.

To further illustrate the importance that timing plays, let's have a closer look at the HSYNC signal. The horizontal retrace time is often subdivided into three intervals as shown in **Figure 3.14**:

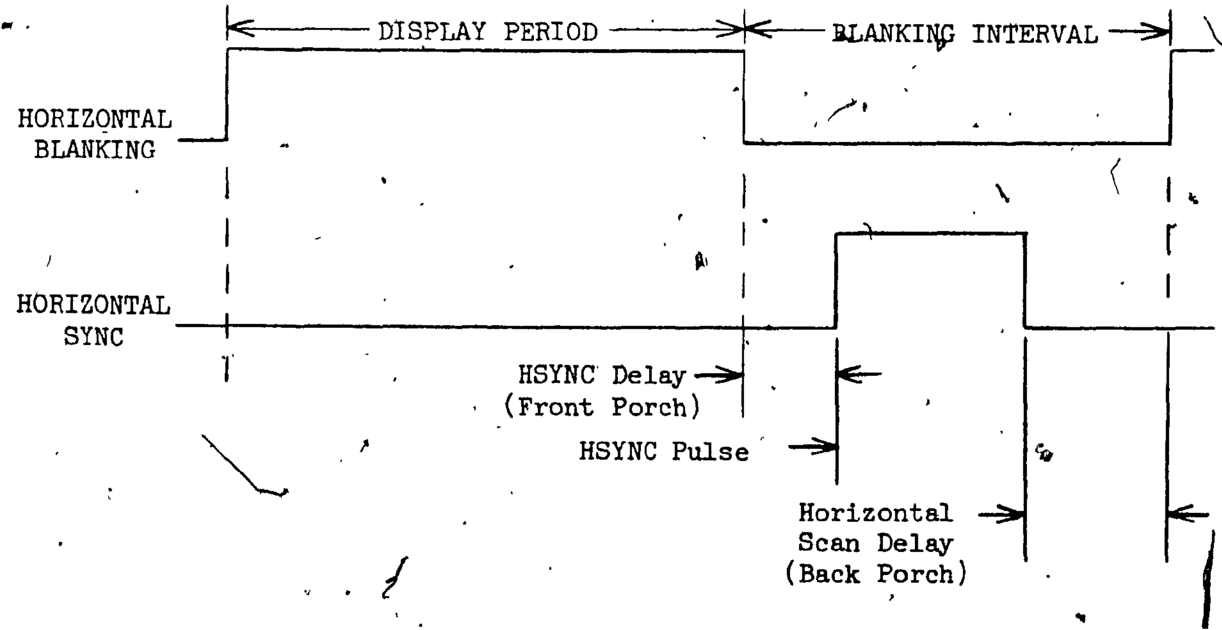


FIGURE 3.14
Horizontal Timing

1. **HSYNC Delay** (or **front porch**) - the number of character times delay after the display period until the HSYNC pulse;
2. **HSYNC Pulse** - the width of the horizontal sync pulse (defined in character times);
3. **Horizontal Scan Delay** (**back porch**) - the character times delay after the HSYNC pulse and prior to the active scan.

The sum of these three intervals is the **Horizontal Blanking Interval.** This interval is required as a window in the horizontal scan period to allow the beam to return or **RETRACE** to the left side of the screen. The retrace time is internal to the CRT monitor and is a function of horizontal scan components. The retrace time is always (must be) less than the horizontal blanking interval. The designer must determine the values of these three intervals from the operating characteristics of the CRT monitor being used. However, typical values compatible with a wide range of commercial CRT monitors dictate a ratio of 1:2:2 to be kept among them. By the same token, the horizontal blanking range should be between 10 to 12 usec.

Since we want to display 80 characters per line, to keep calculations simple, let's consider a **total** of 100 characters on a line, thus resulting in a 25% allowable retrace time. The character rate, the rate at which characters are displayed on the screen, thus would be:

character rate (clock) = horizontal scan frequency x number of total characters per line

= 15625 Hz x 100

= 1.5625 MHz (i.e., 640 nsec)

Hence, the horizontal scan time for each character along any line would be 640 nanoseconds. (See **Figure 3.16**).

Since there are 8 dots (horizontally) per character, the dot clock needed may be calculated as follows:

dot clock = character clock x number of total horizontal dots per character

= 1.5625 MHz x 8

= 12.5 MHz

This is the clock rate applied to the shift register generating the VIDEO signal interfacing directly with the CRT monitor. At this point, all the calculations required for the horizontal parameters are complete.

Vertical Parameters: The VERTICAL PERIOD contains both the **displayed** portion of the screen and the retrace time. Like the horizontal period, the vertical period can be broken down into small units; in this case such units are called **character rows.** One character row is the number of scan lines required to display one character row plus the number of blank scan lines between two rows. In our example we have chosen an 8 x 10 matrix, hence each row of characters will need a total of 10 scan lines. Note that the units for vertical events can be either **scan lines** or **character rows.** To determine the character row clock rate we have:

character row clock = horizontal scan frequency / number of total scan lines per character

= 15625 Hz / 10

= 1562.5 Hz

To determine the vertical sync (VSYNC) pulse width we must consider that as in the case of the horizontal retrace, the vertical retrace is subdivided into three intervals, **Figure 3.15**:

1. **VSYNC Delay** - the number of character rows delay after the vertical display period and before the VSYNC pulse;
2. **VSYNC Pulse** - the width of the vertical sync pulse (measured in scan lines);
3. **Vertical Scan Delay** - the delay after the vertical sync and the next displayed information in scan lines.

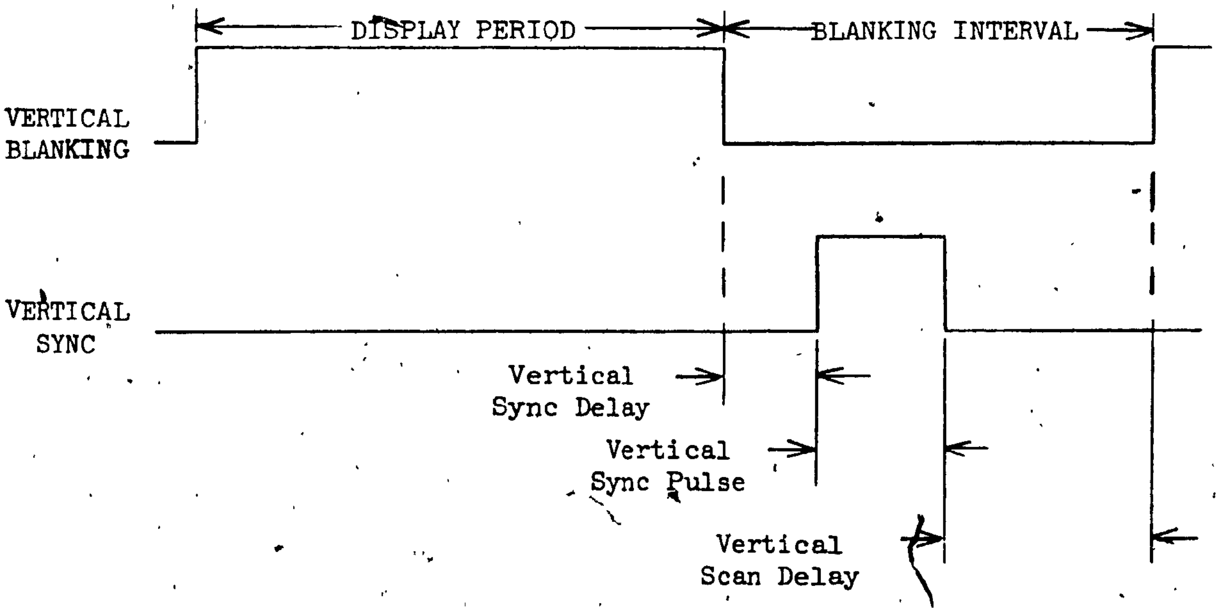


FIGURE 3.15
Vertical Timing

The sum of these three intervals is the **Vertical Blanking Interval**.

For every frame there are 262 lines. Since we are only displaying 24 character rows of 10 lines each (240 scan lines), there remains 22 scan lines for the vertical blanking interval. Considering that 15625/60 = 260.4, we only need 20 extra lines to obtain the adequate vertical sync frequency (15625 / 260 = 60.096), or two more character rows for a total of 26. Considering that a typical value for the vertical retrace time is **1 msec**, 20 horizontal scan lines would provide enough time (20 x 64 usec = 1.28 msec) for the vertical retrace.

Figure 3.16 summarizes the entire timing chain used in this example.

Taking into consideration the contents of this chapter, we are now ready to compare the circuitry making up the Sync Generator (**Figure 3.1**) and that of an LSI CRT controller (**Figure 3.17**). It can be already seen that most of the discrete logic in **Figure 3.1**, except the display RAM, character generator and shift register, has been integrated into Hitachi's HD 6845 CRT controller, **Figure 3.17**, disregarding the internal registers. Other CRTC's available off-the-shelf offer the same if not more circuitry within their controllers. Thus, to the designer and manufacturer of CRT terminals, the advantages of replacing 30 to 40 or more MSI packages by an LSI CRTC chip are obvious, making it an ideal candidate for the job.

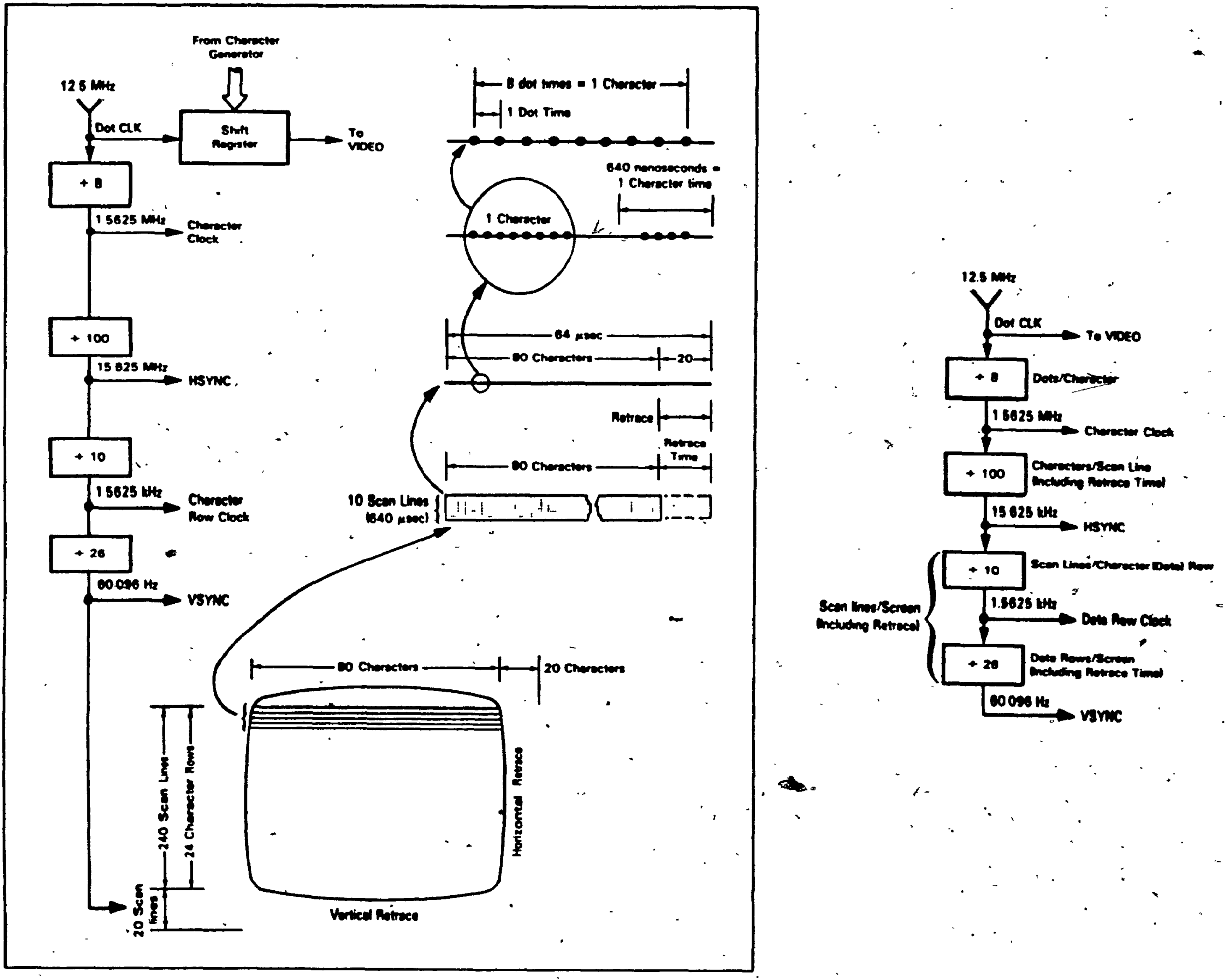


FIGURE 3.16

Summary of CRT Timing Chain Example[12]

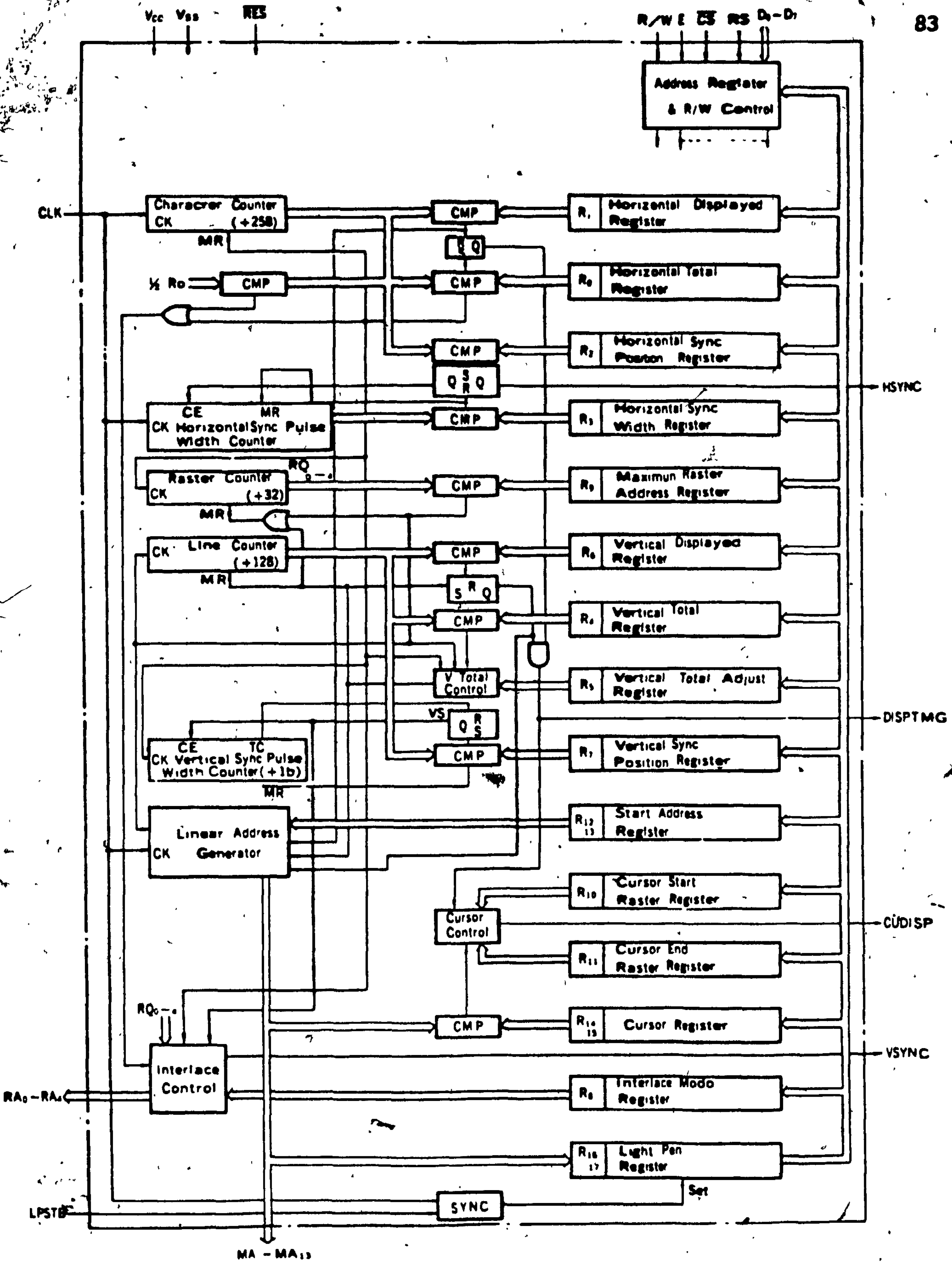

Vcc
Vss
RES
R/W
E
CS
RS
D0-D7
Address Register & R/W Control
CLK
Character Counter CK (+258)
MR
CMP
R1 Horizontal Displayed Register
R0 Horizontal Total Register
½ R0
R2 Horizontal Sync Position Register
Q S R Q
CE MR CK Horizontal Sync Pulse Width Counter
R3 Horizontal Sync Width Register
HSYNC
RQ0-4
Raster Counter CK (+32)
R9 Maximun Raster Address Register
Line Counter CK (+128)
R6 Vertical Displayed Register
S R Q
R4 Vertical Total Register
V Total Control
R5 Vertical Total Adjust Register
VS
DISPTMG
CE TC CK Vertical Sync Pulse Width Counter (+16)
R7 Vertical Sync Position Register
Linear Address CK Generator
R12 13 Start Address Register
R10 Cursor Start Raster Register
Cursor Control
CUDISP
R11 Cursor End Raster Register
R14 15 Cursor Register
Interlace Control
VSYNC
RA0-RA4
R8 Interlace Modo Register
R16 17 Light Pen Register
Set
LPSTB
SYNC
MA - MA13

FIGURE 3.17
Hitachi's HD6845 CRT Controller Block Diagram[18]

The added premium of incorporating programmability into the device, giving it the necessary flexibility and versatility to generate various CRT signal interface timing conditions under different parameter constraints, makes the CRTC even more appealing.

SECTION 2

EVALUATION

CHAPTER 4

COMPARISON OF CRT CONTROLLERS

Although there are significant differences between the functions, capabilities and characteristics of each of the CRT controllers available off-the-shelf, all of them have been shaped by the requirements of the cathode ray tube itself and by prevailing interface standards to this device. In this chapter we carry out a detail analysis of each of three commonly available, low-cost CRT controllers (CRTC) in order to decide upon the best device to be incorporated into the design of the terminal. For every CRTC, a terminal's system block diagram is generated as well, in order to evaluate the minimum chip count necessary to achieve the low-cost, low-count, terminal's implementation.

4.1 The Basis for CRT Controller Comparison

Considering the contents of Chapter 3, we are now able to generate a general block diagram identifying the typical logic functions that must be provided to create the interface between a uP and a CRT monitor, **Figure 4.1**. This will serve us as a common basis to point out and compare the different functions each CRT controller device has built-in. The CRT controllers we will consider here, will not include **all** of the functions shown in **Figure 4.1**, the reasons being as follows:

1. The **dot timing circuitry**, for example, operates in the 10's of megahertzs range in most applications, and conventional NMOS circuitry (the process from which these devices are made) will

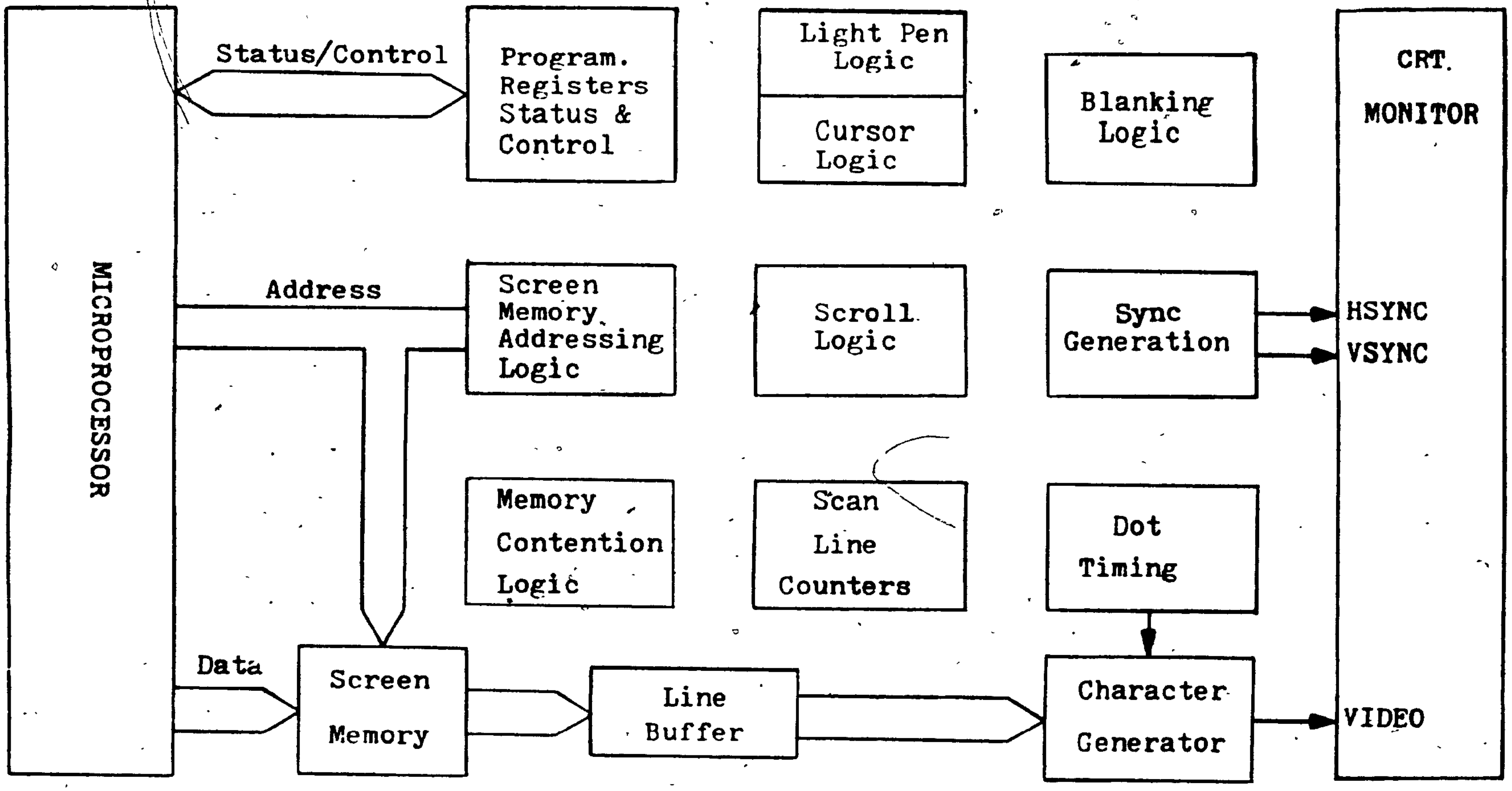


Figure 4.1 General Block Diagram of a CRT Controller

either not operate in this range or degrade the signal characteristics due to its behaviour at such high frequencies. As a result, MOS CRTCs can't include an on-chip, dot-rate crystal oscillator or a dot counter and associated logic due to frequency limitations. However, CRTCs built with bipolar circuitry (I^2L or TTL) don't lend themselves to such limitations.

2. Implementing all possible functions on a chip will force the manufacturer to sacrifice flexability, thus forcing the user to utilize the chip in a dedicated manner. However, the more versatile the chip, the larger the market for its use. Thus, most CRTCs contain a limited number of built-in functions.
3. By allowing the screen memory to reside externally to the CRTC, the user can decide upon its size to best suit its system requirements while staying with the limits provided by the CRTC architecture.
4. Character generation logic, as well, is external to the CRTC to allow the user flexibility depending upon its particular application.

A most important point to note at this stage, which will impose serious limitations, if not considered carefully at an early phase of the design, is that:

> "the approach that the chip manufacturer has taken when implementing the CRT controller functions will dictate, to a large extent, the approach that the designer must take when incorporating the device in the system."[12]

Carelessness on the part of the designer will affect the final hardware and software as well, since they depend heavily on the CRT controller chosen for the job.

4.2 Type of CRT Controller

Presently, several forms of displays are in existence; graphic and alphanumeric.

Alphanumeric: In this kind of display, the data making up the character line is repetitively presented to the character generator logic until all the scan lines comprising that character row have been displayed - as described in Chapter 3. Such an arrangement allows memory storage requirements and memory accesses to be minimized and the character generator logic assumes responsibility for producing each individual dot required along a scan line. This results in a limitation on the display capability - which can only be as good as the complexity of the character generator logic. In most alphanumeric displays, each portion of the scan line making up each character on the line is stored in one byte. Alphanumeric displays have limited graphic capabilities.

Graphic: When graphic information needs to be displayed, then the system must have the capability of manipulating individually every dot on the CRT screen. Such an approach typically requires more memory and resources beyond those usually provided by a CRT controller. In this case, in order to independently control every dot on the CRT screen, one must map every dot to a specific memory data bit. The only way to do so is by providing memory storage for each dot of each scan line versus the provision of memory storage for every 7 or more scan lines (depending on the character size) for an alphanumeric display. Thus, graphic displays tend to be more complex and memory-hungry than alphanumeric.

The CRT controllers that will be desribed herein are all oriented towards alphanumeric displays.

4.3 Description and Analysis of Individual CRT Controllers

In order to be able to choose the right CRTC we must study the features of the available controllers. We will now proceed to describe the characteristics and capabilities of three low-cost, off-the-shelf, versatile CRT controllers.

4.3.1 The 5037 CRT Controller

Among the CRTCs that will be described, no other device is available from such a large member of manufacturers:

Mostek Corporation (device no. MK3807)[19]
Solid State Scientific (SSS) (device no. SND5027/37)[21]
Standard Microsystems Corporation (device no. CRT5027/37/47/57)[20]
Texas Instruments (device no. TMS9927)[22]

This is mainly due to the length of time this device has been in existence and to its acceptance and use. However, SMC and SSS are the only manufacturers providing different versions of the same device.

The circuitry built into the 5037 is shown in **Figure 4.2.** From its block diagram, we can derive the logic functions provided by this CRTC, as shown in **Figure 4.3** (shaded areas).

General Description

The 5037 is a CRT video timer and controller (VTAC) chip that is user programmable. It is a 40-pin N-channel MOS/LSI device containing the logic functions required to generate all the timing signals for the presentation and formatting of interlaced and non-interlaced video data on a standard or non-standard CRT monitor.

With the exception of the dot counter, all frame formatting such as:

horizontal sync
vertical sync
composite sync

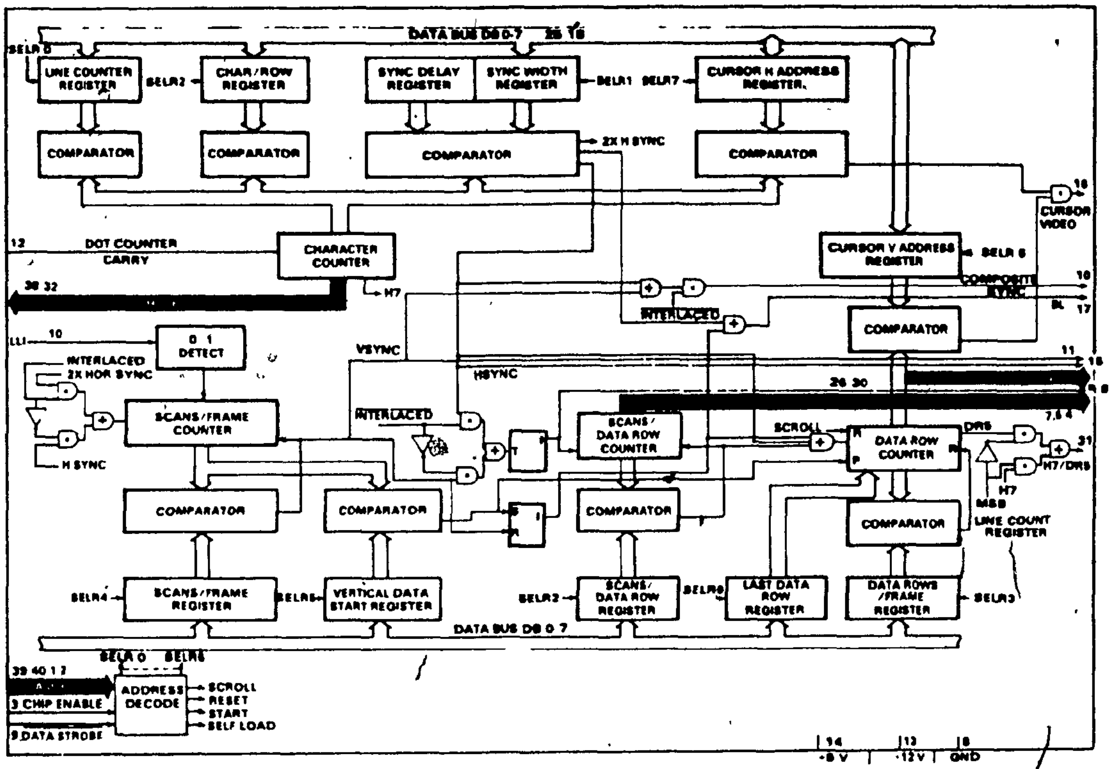


FIGURE 4.2
Block Diagram of the 5037 CRT Controller[19,20,21]

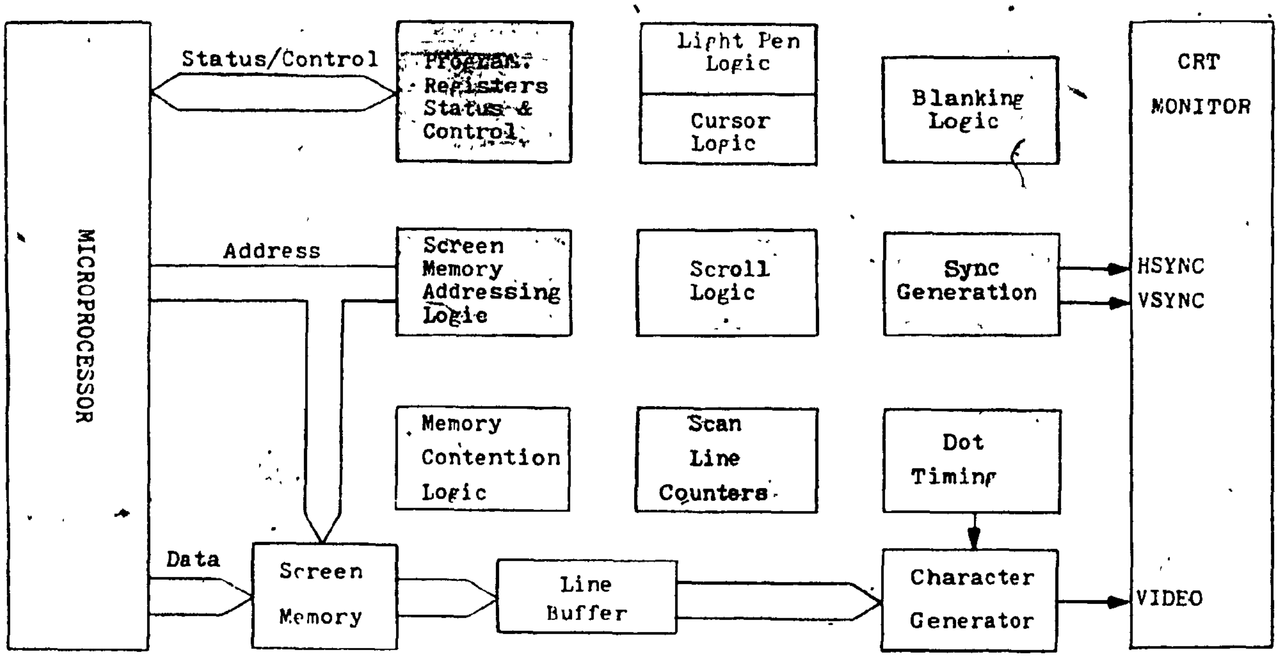


FIGURE 4.3
Functional logic built into the 5037 CRT Controller (shown shaded)

characters per data row
data rows per frame
raster scans per data row
raster scans per frame

is totally user programmable. The data row counter it contains facilitates scrolling as well.

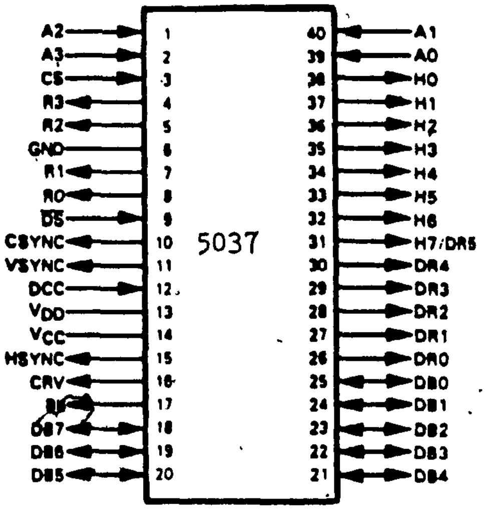
Programming takes place by means of loading **seven** 8-bit **control registers** directly off an 8-bit bidirectional data bus. **Four register address lines** and **a chip select** line provide complete uP compatability for program controlled set up. The device can also be "self-loaded" via an external PROM tied on the data bus.

In addition to the 7 control registers, 2 **additional registers** are provided to store the cursor character and data row addresses for the generation of the cursor video signal. The contents of these two registers can also be read out onto the bus for update by the program.

Built-in Signals

The 5037 was one of the first LSI controllers to be introduced. Although the functions it provides may appear to be more elementary than those available with devices introduced more recently, such as Motorola's MC6845 and Intel's 8275, it still provides several interesting functions not available on any of the other devices we will consider.

Figure 4.4 shows the pinout and signal names as well as the functional description of the 5037. These signals may be divided into **four categories:**



Description of Pin Functions

Pin No.	Symbol	Name	Input/ Output	Function
25-18	DBØ-7	Data Bus	I/O	Data bus. Input bus for control words from microprocessor or PROM. Bidirectional bus for cursor address.
3	CS	Chip Select	I	Signals chip that it is being addressed
39,40,1,2	AØ-3	Register Address	I	Register address bits for selecting one of seven control registers or either of the cursor address registers
9	DS	Data Strobe	I	Strobes DBØ-7 into the appropriate register or outputs the cursor character address or cursor line address onto the data bus
12	DCC	DOT Counter Carry	I	Carry from off chip dot counter establishing basic character clock rate. Character clock.
38-32	HØ-6	Character Counter Outputs	O	Character counter outputs.
7, 5, 4	R1-3	Scan Counter Outputs	O	Three most significant bits of the Scan Counter, row select inputs to character generator
31	H7/DR5	H7/DR5	O	Pin definition is user programmable. Output is MSB of Character Counter if horizontal line count (REG.Ø) is ≥128; otherwise output is MSB of Data Row Counter.
8	RØ	Scan Counter LSB	O	Least significant bit of the scan counter. In the interlaced mode with an even number of scans per data row, RØ will toggle at the field rate; for an odd number of scans per data row in the interlaced mode, RØ will toggle at the data row rate.
26-30	DRØ-4	Data Row Counter Outputs	O	Data Row counter outputs.
17	BL	Blank	O	Defines non active portion of horizontal and vertical scans
15	HSYN	Horizontal Sync	O	Initiates horizontal retrace.
11	VSYN	Vertical Sync	O	Initiates vertical retrace.
10	CSYN/ LLI	Composite Sync Output/ Line Lock Input	O/I	Composite sync is provided on the CRT 5027 and CRT 5037. This output is active in non-interlaced mode only. Provides a true RS-170 composite sync wave form. For the CRT 5057, this pin is the Line Lock Input. The line frequency waveform, processed to conform to the VTAC's® specified logic levels, is applied to this pin.
16	CRV	Cursor Video	O	Defines cursor location in data field.
14	Vcc	Power Supply	PS	+5 volt Power Supply
13	VDD	Power Supply	PS	+12 volt Power Supply

FIGURE 4.4

5037 CRT Controller Pinout & Signal Description[12,19,20,21]

1. signals used to interface the CRTC to the uP and system buses (timing input signals are included in this group);
2. signals used to interface the CRTC to the video moemory and character generator logic;
3. signals used to interface directly with the CRT monitor; and
4. power supply signals.

The 5037 signals that belong to each category are shown in **Table 4.1** below.

uP System Interface Signals	**Video Memory & Character Generator Signals**	**CRT Monitor Interface Signals**	**Power Signals**
AO-A3	DRO-DR4	BL	VCC (+5V)
CS	HO-H6	CRV	VDD(+12V)
DBO-DB7	H7/DR5	CSYNC	GND
DCC	RO-R3	HSYNC	
DS		VSYNC	

TABLE 4.1*

5037 CRTC Signal Classification

*This tabular format will be used to classify the signals of the other CRTCs that will be evaluated in order to ease the task of comparing them later on.

Most signals are standard to all of the CRTCs we will consider, and are well defined in **Figure 4.4**. However, several unusual signals not found on any other CRTC need further elaboration:

Data Strobe ($\overline{DS}$) signal: the 5037 has no read and write control signals; however, the $\overline{DS}$ is the functional combination of both of these signals. Hence, the uP's READ and WRITE signals must be combined to be able to strobe data into and out of the device via the data bus. Thus, $\overline{DS}$ must be pulsed low to allow a register access or command initiation.

HO-H7 and DRO-DR5 signals: these are respectively the **horizontal character or column** address outputs and the **vertical or data row** address outputs. The 5037 generates such signals to address the memory on a **row/column basis.** As was already pointed out, although such a method gives some flexibility in utilizing the memory address space, it results in an inefficient use of memory space in most cases. However, when used adequately, it can prove advantageous.

CSYNC signal: this is the only CRTC we will encounter with such a signal. It is the **composite synchronization** signal. Its format is a pulse stream which includes both the HSYNC and VSYNC signals. This signal can only be used if the 5037 is operating in the **non-interlaced mode.** The advantage of providing such a signal is that it can be externally mixed (using less hardware) with the video signal to produce a composite video output to the CRT monitor.

Programmable Registers

The 5037 contains a total of 16 registers, **Figure 4.5a.** These may be classified under three categories:

1. **Control Registers**: A total of 7 **registers** (RO to R6); they are programmable and are used to define timing parameters and basic screen format (**Figure 4.5a,b**). **Figures 4.6 and 4.7** show the bit assignment and programming charts for these registers. These are **write-only** registers, and are usually loaded when a system is first turned on (power-up) and are not accessed thereafter.

REGISTER SELECTS/COMMAND CODES

A3	A2	A1	A0	Select/Command	Description
0	0	0	0	Load Control Register 0	see Figure 4.5b
0	0	0	1	Load Control Register 1	
0	0	1	0	Load Control Register 2	
0	0	1	1	Load Control Register 3	
0	1	0	0	Load Control Register 4	
0	1	0	1	Load Control Register 5	
0	1	1	0	Load Control Register 6	
0	1	1	1	Processor Initiated Self Load	Command from processor instructing MK3807 VCU to enter Self Load Mode (via external PROM)
1	0	0	0	Read Cursor Line Address	
1	0	0	1	Read Cursor Character Address	
1	0	1	0	Reset	Resets timing chain to top left of page Reset is latched on chip by $\overline{DS}$ and counters are held until released by start command.
1	0	1	1	Up Scroll	Increments address of first displayed data row on page, i e , prior to receipt of scroll command—top line = 0, bottom line = 23 After receipt of Scroll Command—top line = 1, bottom line = 0 .
1	1	0	0	Load Cursor Character Address[1]	
1	1	0	1	Load Cursor Line Address[1]	
1	1	1	0	Start Timing Chain	Receipt of this command after a Reset or Processor Self Load command will release the timing chain approximately one scan line later In applications requiring synchronous operation of more than one VCU the dot counter carry should be held low during the $\overline{DS}$ for this command
1	1	1	1	Non-Processor Self Load	Device will begin self load via PROM when $\overline{DS}$ goes low The 1111 command should be maintained on A3-0 long enough to guarantee self load (Scan counter should cycle through at least once) Self load is automatically terminated and timing chain initiated when the all "1's" condition is removed, independent of $\overline{DS}$ For synchronous operation of more than one VCU the Dot Counter Carry should be held low when the command is removed

NOTE 1 During Self Load the Cursor Character Address Register (REG 7) and the Cursor Row Address Register (REG 8) are enabled during states 0111 and 1000 of the R3-R0 Scan Counter outputs respectively. Therefore Cursor data in the PROM should be stored at these addresses

FIGURE 4.5a[12,19,20,21]

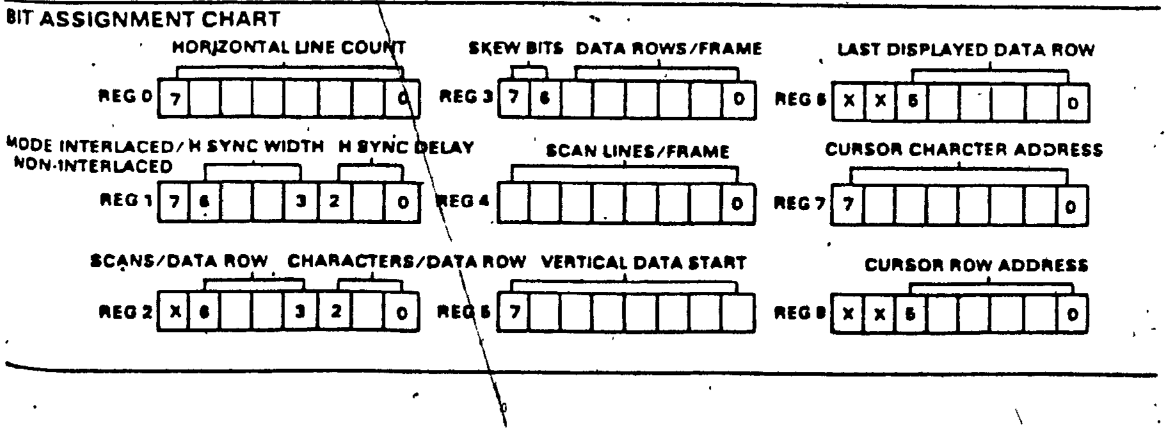


FIGURE 4.5b[12,19,20,21]

Horizontal Formatting:	
Characters/Data Row	A 3 bit code providing 8 mask programmable character lengths from 20 to 132 The standard device will be masked for the following character lengths, 20, 32, 40, 64, 72, 80, 96, and 132
Horizontal Sync Delay	3 bits assigned providing up to 8 character times for generation of "front porch".
Horizontal Sync Width	4 bits assigned providing up to 16 character times for generation of horizontal sync width
Horizontal Line Count	8 bits assigned providing up to 256 character times for total horizontal formatting.
Skew Bits	A 2 bit code providing from a 0 to 2 character skew (delay) between the horizontal address counter and the blank and sync (horizontal vertical, composite) signals to allow for retiming of video data prior to generation of composite video signal The Cursor Video signal is also skewed as a function of this code
Vertical Formatting	
Interlaced/Non-interlaced	This bit provides for data presentation with odd/even field formatting for interlaced systems It modifies the vertical timing counters as described below A logic 1 establishes the interlace mode
Scans/Frame	8 bits assigned, defined according to the following equations Let X = value of 8 assigned bits. 1) in interlaced mode—scans/frame = 2X + 513 Therefore for 525 scans, program X = 6 (00000110) Vertical sync will occur precisely every 262 5 scans, thereby producing two interlaced fields. Range = 513 to 1023 scans/frame, odd counts only 2) in non-interlaced mode—scans/frame = 2X + 256 Therefore for 262 scans, program X = 3 (00000011) Range = 256 to 766 scans/frame, even counts only In either mode, vertical sync width is fixed at three horizontal scans (≡3H)
Vertical Data Start	8 bits defining the number of raster scans from the leading edge of vertical sync until the start of display data At this raster scan the data row counter is set to the data row address at the top of the page
Data Rows/Frame	6 bits assigned providing up to 64 data rows per frame
Last Data Row	6 bits to allow up or down scrolling via a preload defining the count of the last displayed data row
Scans/Data Row	4 bits assigned providing up to 16 scan lines per data row

FIGURE 4.6

Control Register Bit Assignment[19]

CONTROL REGISTERS PROGRAMMING CHART

Horizontal Line Count: Total Characters/Line = N + 1, N = 0 to 255 (DB0 = LSB)

Characters/Data Row

DB2	DB1	DB0		
0	0	0	= 20	Active Characters/Data Row
0	0	1	= 32	
0	1	0	= 40	
0	1	1	= 64	
1	0	0	= 72	
1	0	1	= 80	
1	1	0	= 96	
1	1	1	= 132	

Horizontal Sync Delay: = N from 1 to 7 character times (DB0 = LSB N = 0 Disallowed)

Horizontal Sync Width: = N, from 1 to 15 character times (DB3 = LSB N = 0 Disallowed)

Skew Bits

DB7	DB6	Sync/Blank Delay	Cursor Delay (Character Times)
0	0	0	0
1	0	1	0
0	1	2	1
1	1	2	2

Scans. Frame: 8 bits assigned, defined according to the following equations.
Let X = value of 8 assigned bits DB0 = LSB)
1) in interlaced mode— scans/frame = 2X + 513 Therefore for 525 scans, program X = 6 (0000110) Vertical sync will occur precisely every 262 5 scans, thereby producing two interlaced fields.
Range = 513 to 1023 scans/frame odd counts only.
2) in non-interlaced mode—scans/frame = 2X + 256 Therefore for 262 scans, program X = 3 (00000011)
Range = 256 to 766 scans/frame, even counts only
In either mode vertical sync width is fixed at three horizontal scans (= 3H)

Vertical Data Start: N = number of raster lines delay after leading edge of vertical sync of vertical start position (DB0 = LSB)

Data Rows/Frame: Number of data rows = N + 1, N = 0 to 63 (DB0 = LSB)

Last Data Row: N = Address of last displayed data row, N = 0 to 63, ie, for 24 data rows, program N = 23 (DB0 = LSB)

Mode, Register 1, DB7 = 1 established interlace

Scans/Data Row

Interlace Mode

Scans per data Row = N + 2 N = 0 to 14, odd or even counts,

Non-Interlace Mode

Scans per Data Row = N + 1, odd or even count, N = 0 to 15

FIGURE 4.7[19]

2. **Cursor Positioning Registers:** These 2 **read** and 2 **write** registers (4 addresses) allow the programmer to write data into the registers to position the cursor on the screen and read it as well to ascertain the current cursor position. Since the 5037 has no READ/WRITE signals, then two separate addresses must be used to access each of these registers; one for the write operation and another for the read. To position the cursor, one must specify the **row address** as well as the **character** (or **column**) address. These 4 registers are classified as being part of the 9 different commands available on the 5037, **Figure 4.8**.
3. **Command Registers:** The remaining 5 registers, when selected, force specific control functions (called commands) to take place (**Figure 4.8**). The possible commands that may be issued are:

   - **Processor-Initiated Self-Load:** used when a uP loads the control registers and cursor position registers.
   - **Reset:** used to cause the timing chain (the internal counters) of the 5037 to be reset.
   - **Up Scroll:** used to scroll upwards by incrementing register R6 by 1 (**Figure 4.5a,b**).
   - **Start Timing Chain:** must be used after a reset or processor-initiated self-load to allow the 5037 to resume operation.
   - **Non-Processor Self-Load:** used when an external PROM loads the control registers and cursor position registers. In this mode, the 5037 itself provides the address inputs to the PROM. This loading procedure is rarely used nowadays since most systems are under the control of a uP.

Note that when using a uP to initialize the 5037, the commands to be issued must follow the following sequence:

Address				Command Code (Hex)	Read (R) Write(W)	Command
A3	A2	A1	A0			
0	1	1	1	7	X	Processor-Initiated Self-Load
1	0	0	0	8	R	Read Cursor Row Address
1	0	0	1	9	R	Read Cursor Character (Column) Address
1	0	1	0	A	X	Reset
1	0	1	1	B	X	Up Scroll
1	1	0	0	C	W	Load Cursor Character (Column) Address
1	1	0	1	D	W	Load Cursor Row Address
1	1	1	0	E	X	Start Timing Chain
1	1	1	1	F	X	Non-Processor Self-Load

FIGURE 4.8

CRT 5037 Possible Controller Commands[12]

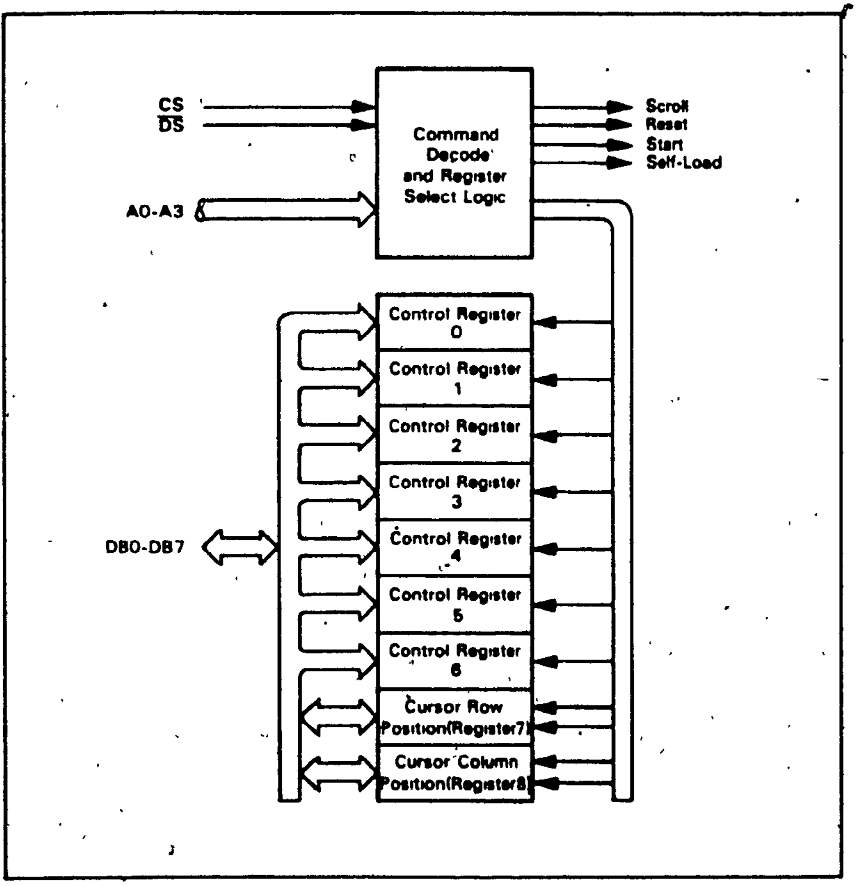


FIGURE 4.9

CRT 5037 Control (write), Cursor Positioning (write/read) and Command Registers[12]

Start Timing Chain
Reset
Load Register 0
Load Register 1
Load Register 2
Load Register 3
Load Register 4
Load Register 5
Load Register 6
Start Timing Chain

Access to any of these 16 registers takes place when the register address inputs (A0-A3) are applied to the 5037 followed by the CS and $\overline{DS}$ signals. The 5037 then decodes the address inputs to generate the appropriate register select signal (**Figure 4.9**).

An important point that deserves caution on the part of the programmer is that since the 5037 addresses memory on a row/column basis, the software must always keep track of where the end of a row is when it moves the cursor.

Interfacing with the uP

The interface between the 5037 and the uP is simple. The signals involved are shown on **Figure 4.10**. The active high **chip select (CS)** signal and the **register address (A0,A3)** are derived from the uP's system address bus. Since the 5037 has 16 registers, it would occupy 16 memory or I/O locations in the system's addressing space. Hence, each register/command of the 5037 would be addressed as a separate memory or I/O device. READ/WRITE operations take place depending upon the register/command to be addressed; of course, under the control of the uP.

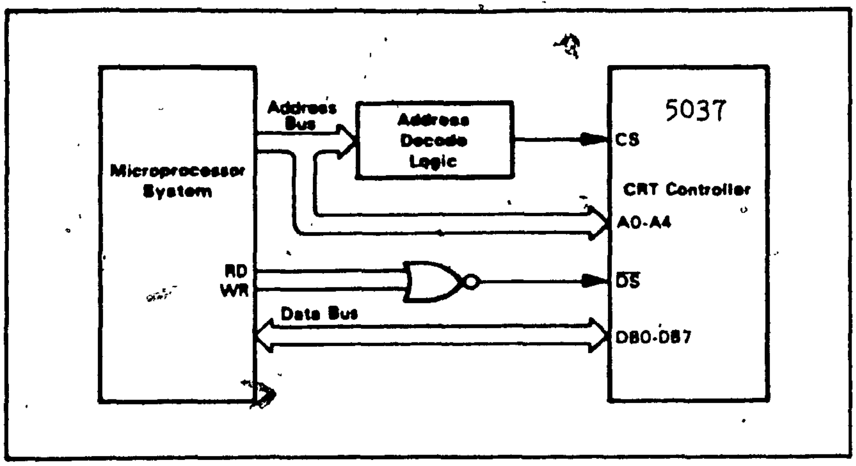
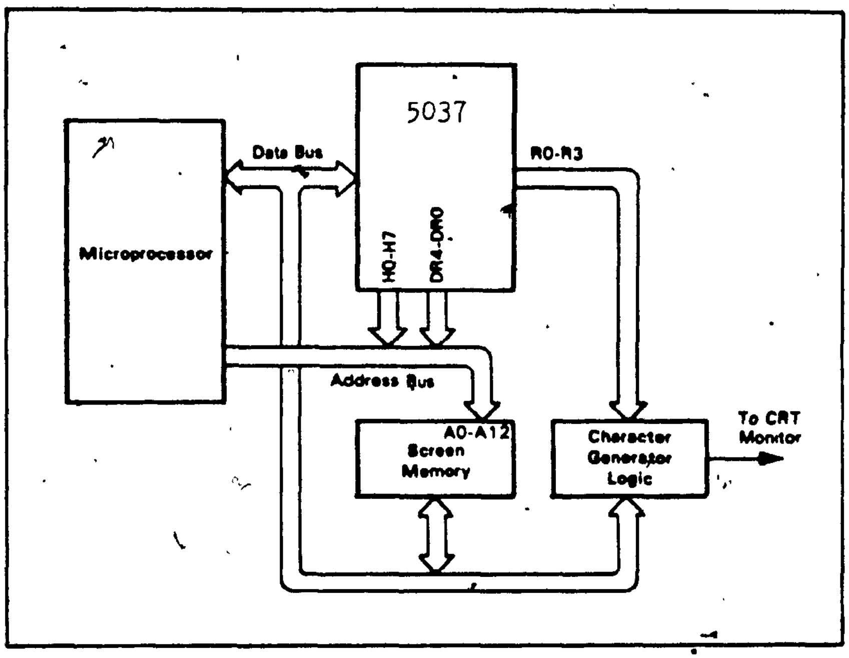


FIGURE 4.10

5037 to uP System Interface[12]

FIGURE 4.11

5037 Interface to Character Generator and Video Memory[12]



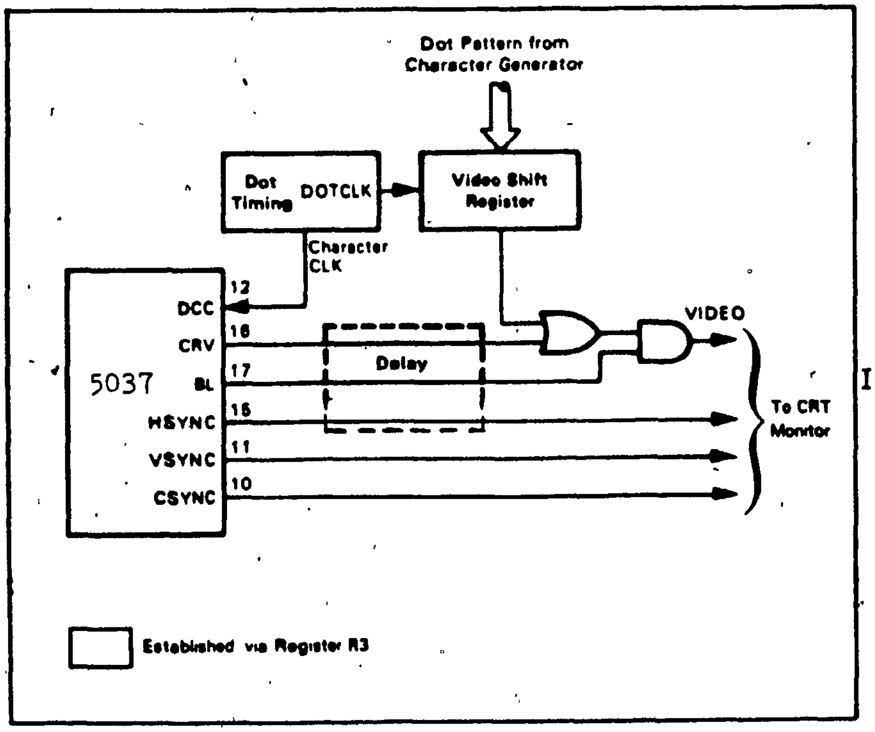


FIGURE 4.12

5037 CRT Monitor Interface Signals[12]

Unlike other CRTCs, the 5037 **does not** have READ and WRITE inputs, or a combination of the two, READ/$\overline{\text{WRITE}}$. However, it utilizes a **Data Strobe ($\overline{\text{DS}}$)** input when reading from or writing to any of its registers. **Figure 4.10** shows the need of a NOR gate to generate the active low $\overline{\text{DS}}$ signal, assuming READ and WRITE are active high. However, for uPs that generate active low READ and WRITE signals, an AND gate would be needed instead. Data transfers take place via the bidirectional **data lines (DBO-DB7).**

Interfacing with the Character Generator and the Video Memory

Since the 5037 addresses the vido memory on a **row/column** basis instead of a linear fashion, it generates **8 horizontal** character count outputs (H0-H7) and **5 vertical** or data row counter outputs (DR0-DR4), **Figure 4.11.** A total of 13 outputs allows to address a memory space of 8K. However, due to the use of the row/column addressing method, memory is utilized inefficiently, preventing the entire 8K space from being used. Nevertheless, this method is well suited for manipulation of video data.

With respect to the character generator, the 5037 only provides 4 raster addresses (R0-R3) to indicate the line being scanned within each character-cell of each character-row. This output combined with the video memory character code output, provide the necessary inputs to the character generator logic, **Figure 4.11.**

Interfacing with the CRT Monitor

The 5037 provides altogether 5 different signals to interface with a CRT monitor, **Figure 4.12.** They are:

Horizontal Sync (HSYNC): programmable via register R1;

Vertical Sync (VSYNC): fixed to 3 scan lines (it meets the requirements of EIA RS-170 video specification[23]);

Composite Sync (CSYNC): a combination of HSYNC and VSYNC signals;

Cursor Video (CRV): fixed block cursor format, offers no programmable options;

Blanking (BL): active high during horizontal and vertical retrace times.

In order to compensate for the delay encountered while accessing a character from the video memory, generate the dot pattern via the character generator and shifted out the video shift register, the 5037 by means of register 3 (**Figure 4.5b**) allows a programmable delay (skew or displacement) to be introduced before generating the CRV, BL and HSYNC signals. This is shown in **Figure 4.12** by means of the dotted block denoted as "Delay". To summarize, the relationship between the 5037 programmable registers and the generation of the timing chain signals to interface to the CRT monitor is shown in **Figure 4.13.**

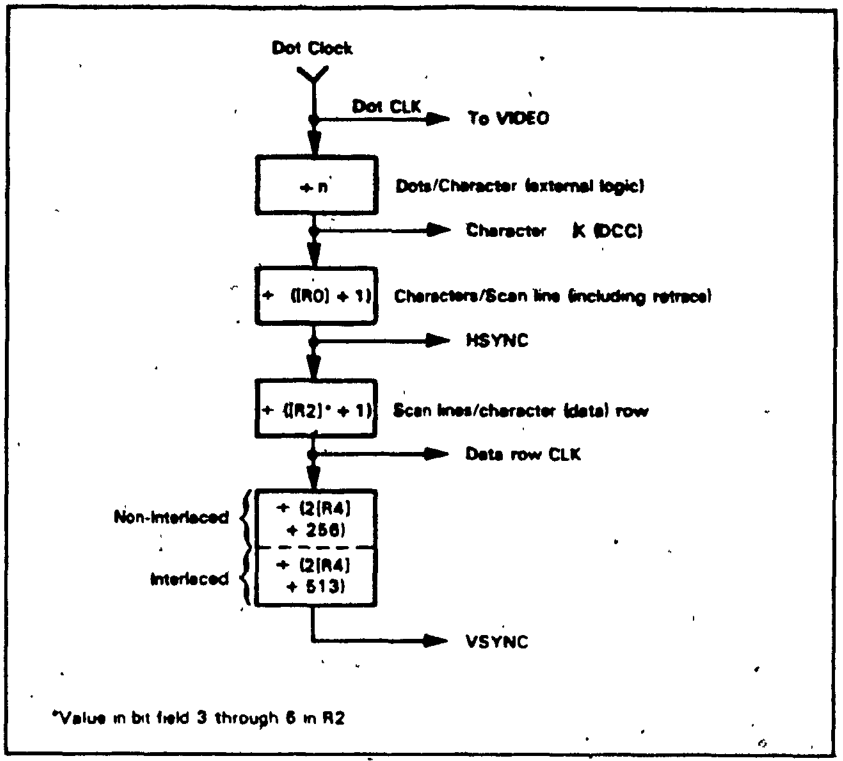
Missing Signals, Logic or Registers

The 5037 does not contain or provide any of the following:

- memory contention logic signals to simplify access to the video memory by both CRTC and uP; it must be provided externally;

- generation of linear addressing to the video memory;

- options to provide cursor blinking nor different cursor shapes via a programmable register; the cursor displayed is always of the **block** form and reverse video, thus any desired changes must be implemented by means of external hardware;



*Value in bit field 3 through 6 in R2

FIGURE 4.13

Relationship between 5037 Programmable Register and the CRT Timing Chain[12]

- light pen register/signal.

Terminal Design Simplification via the 5037

To ease the designer's task of implementing a terminal with a relatively small number of components, the manufacturers of the 5037 have introduced a companion chip to this device, the 8002. Known as "the CRT Video Generator and Attributes Controller" or VDAC for short, it incorporates all of the following functions:

- character generator (7 x 11 on a 9 x 12 dot matrix)
- video shift register (up to 20 MHz)
- wide and thin graphics modes of operation
- attribute controller
- different cursor modes and blink rates (programmable)
- programmable character blink rate
- allows generation of subscripts

The result is obvious. First of all, the character generator, the video shift register and associated circuitry are all part of the same chip. Thus, lowering the system's component count, its power consumption and reducing the size and complexity of the terminal's PCB. The other factor, is the ease with which changes can be brought about. The device's versatility is due to its programmability.

Terminal Implementation Block Diagram

Considering our objective, the design of a low power, low chip count, reliable, fast, cost effective and smart terminal, the necessary circuitry required to achieve it was identified and is shown on the block diagram of **Figure 4.14.** The system is built around the 8085 uP, the 5037 CRTC and the 8002 VDAC, the major LSI components providing system and video control. The total chip count is 28 ICs.

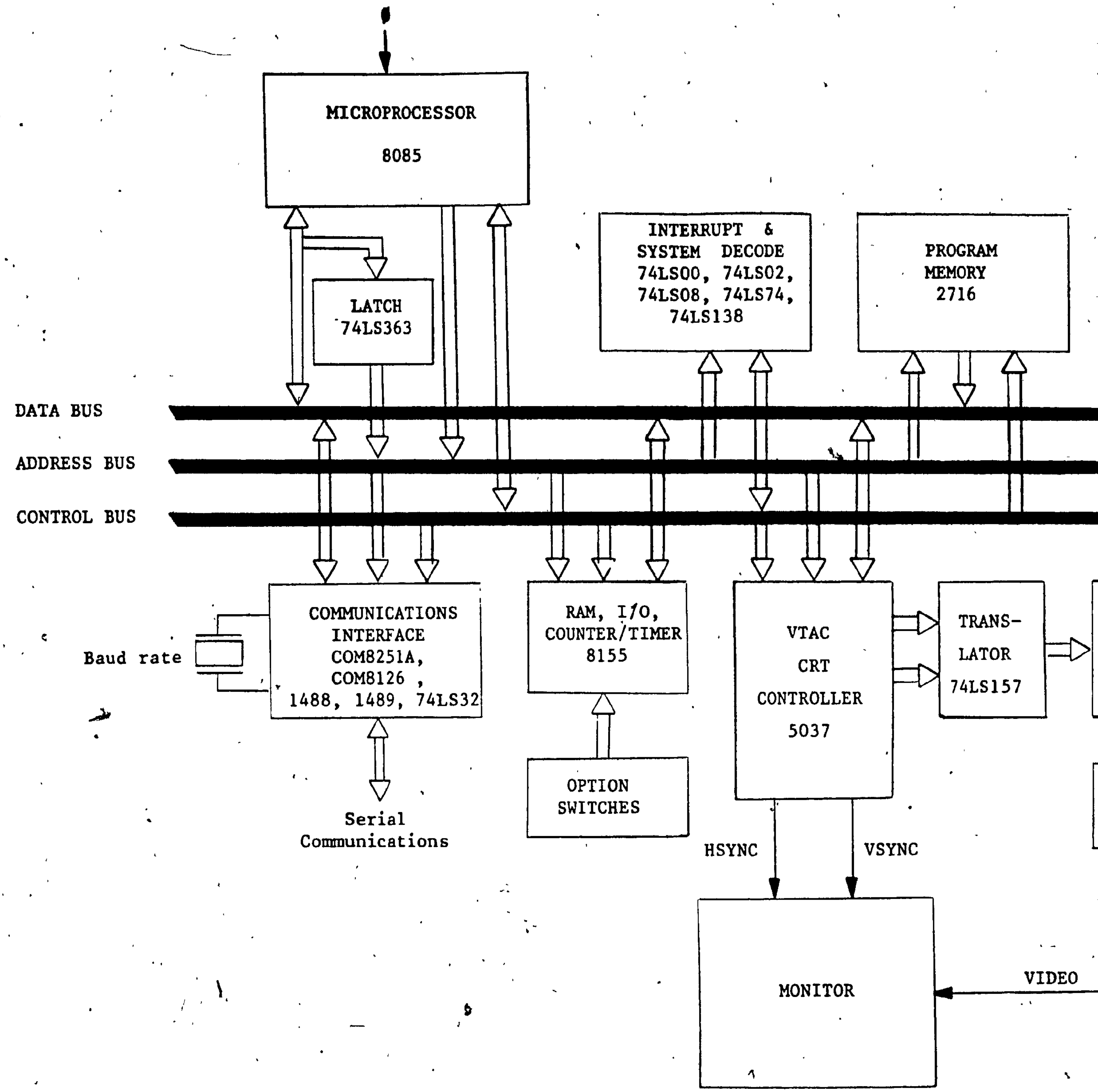


Figure 4.14

Terminal Block Diagram using the 5037 CRT contro

(total component count is 28 ICs)

1 OF

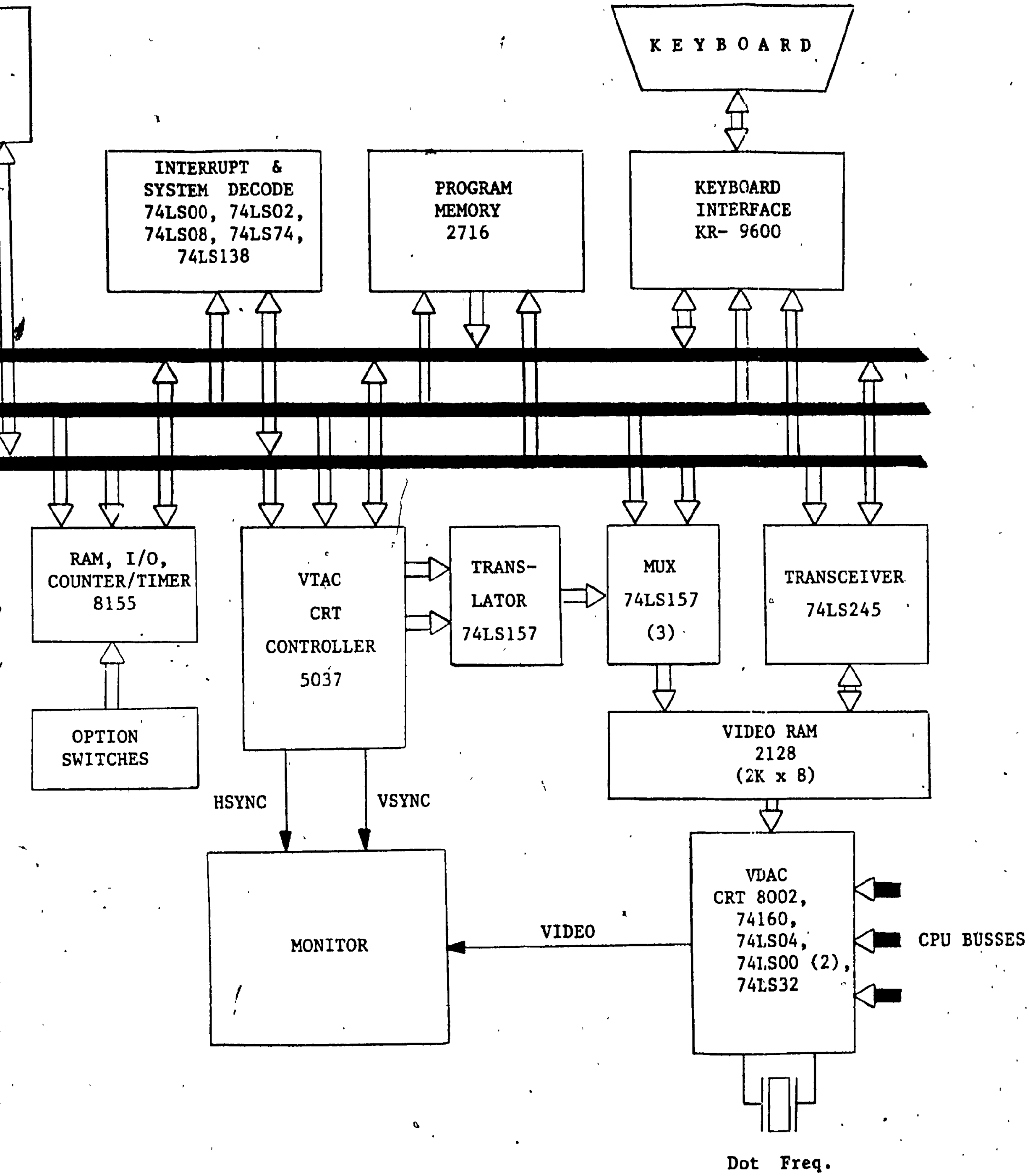


Figure 4.14

Terminal Block Diagram using the 5037 CRT controller

(total component count is 28 ICs)

4.3.2 The 6845 CRT Controller

The 6845 is one of the more recently introduced CRT controllers. However, its popularity and use is gaining recognition at an accelerated pace. Although it was introduced by Motorola, several second sources exist; however, most of these supplies offer either improved or simplified versions of the original device. Among the different suppliers there is:

Identical Versions

Motorola (device no. MC6845)[23]
Hitachi (device no. HD6845)[18,24]

Different Versions

AMI (device no. S68045)[25]: does not support the light pen input and associated register; registers R0 to R11 are mask programmed, hence not controllable by the user.

Synertek (device no. SY6545-1)[26]: supplies the best version of the improved 6845. It contains memory contention logic built-in, thus preventing the need for external hardware; permits memory addressing in a linear fashion or either by row/column; has improved light pen register circuitry and it contains a status register which can be read by the uP.

Rockwell (device no. R6545-1)[27]: same as Synertek's version.

To compete with the suppliers offering cheaper versions, Motorola has recently made available, as well, a low-cost, pin compatible version of the 6845, the MC6835.[23] It is identical except that:

- it is mask programmable, supporting two selectable screen formats using a program select input, thus registers R0 to R11 are <u>not</u> accessible by the user via the data bus.
- does not support the light pen option.
- its die size is smaller, since the light-pen registers have been removed, thus offering a price advantage.

Throughout this analysis, we will consider only the original device introduced by Motorola, the MC6845.

The circuitry built into the 6845 appears in **Figure 4.15**. From the general block diagram of a CRTC, **Figure 4.1**, the logic functions built into the 6845 are shown in **Figure 4.16** (shaded areas).

General Description

The 6845 CRT controller performs the interface between a uP and a raster-scan CRT display. It is user programmable and it is housed in a 40-pin N-channel MOS LSI integrated circuit. This CRTC has been optimized for the hardware/software balance required for maximum flexibility. As such it provides video timing and screen memory addressing but **no memory contention logic**. More particularly, the 6845 contains the necessary logic to support the following functions:

scroll control (by page, line or character)
cursor format and blink rate control, and
a light pen

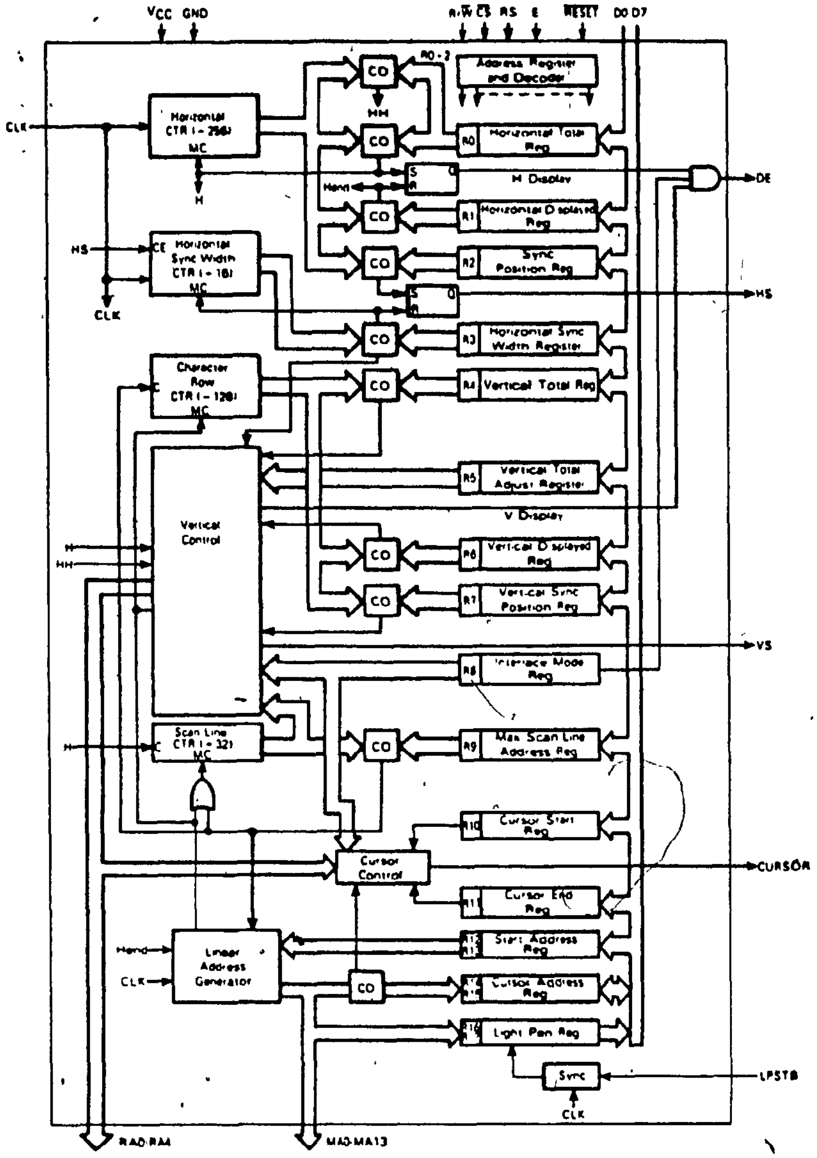
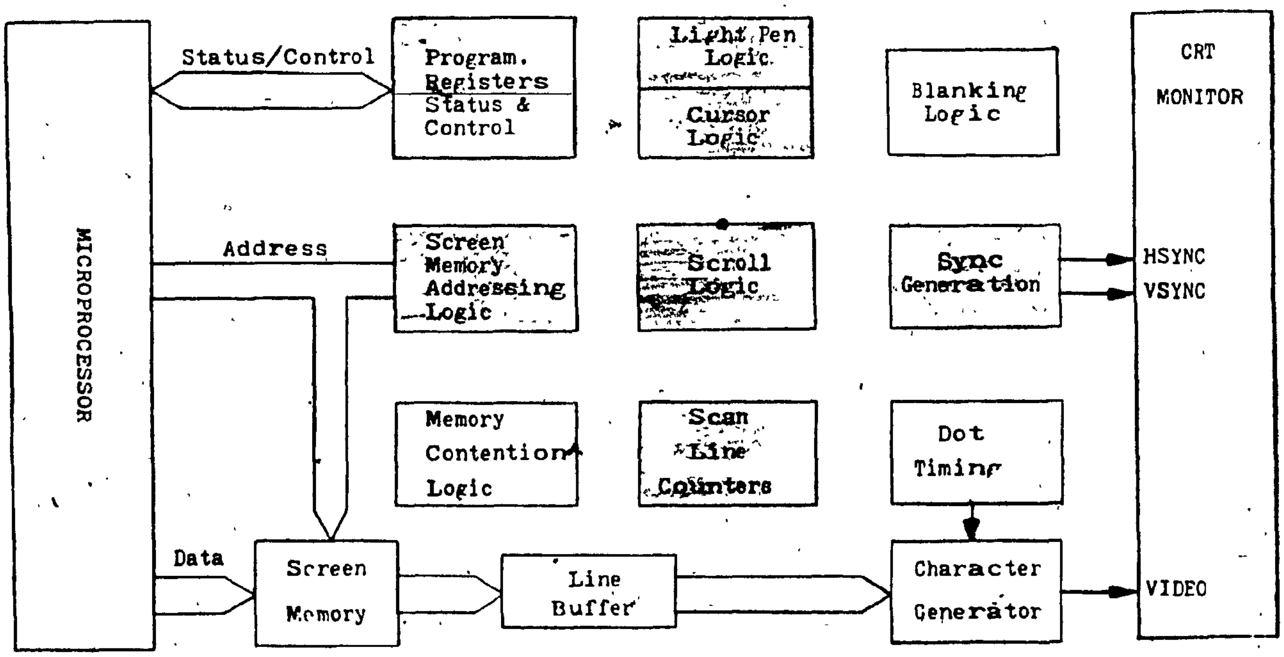


FIGURE 4.15

Motorola's MC6845 Block Diagram[23]

FIGURE 4.16

Functional logic built into the 6845 CRT Controller (shown shaded)



Although no dot timing loyic is provided on the chip, since it is an N-MOS device, all frame formatting such as:

horizontal sync
vertical sync
characters per data row
scan lines per data row
raster scans per frame
blanking and scanning mode
(interlace/non-interlace)

is under user control by accessing the appropriate programmable register. The cursor and blanking (or Display Enable (DE)) signals may be delayed by means of a programmable skew.

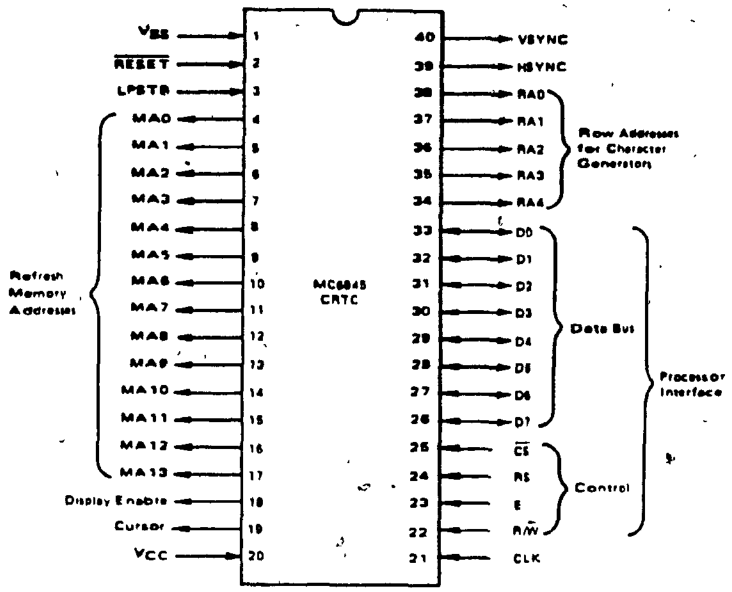
All of the registers making up the 6845, a total of 19, except one, are programmable. It contains no status or control registers.

Built-in Signals

Figure 4.17 shows the pinout and signal names of the 6845. Similar to the 5037, these signals may be divided into four categories, depending upon the function they perform:

uP System Interface Signals	Video Memory & Character Generator Signals	CRT Monitor Interface Signals	Power Signals
CLK	MAO-MA13	CURSOR	VCC
CS	RAO-RA4	DISPEN	VDD
DO-D7		HSYNC	
E		LPSTB	
RS		VSYNC	
R/W			
RESET			

TABLE 4.2
6845 CRTC Signal Classification



Symbol	Name	Description
CLK	Clock	Clock input to synchronize all CRT functions except the processor interface, it corresponds to the character rate clock.
$\overline{\text{CS}}$	Chip Select	Signals chip that is being addressed.
E	Enable	Enables data bus I/O buffers and clocks data to and from CRTC.
$\overline{\text{RESET}}$	Reset	Signal used to initialize the 6845.
RS	Register Select	Selects either the address register (RS=0) or one of the data registers (RS=1) or the internal register file.
R/$\overline{\text{W}}$	Read/Write	Determines whether data is to be written to or read from the 6845.
MA0-MA13	Memory Address	Address lines used to refresh the screen memory.
RA0-RA4	Raster Address	Scan line counter outputs to indicate to character generator which scan line of a character row is being scanned.
HSYNC	Horizontal Sync	Determines the horizontal position of the displayed text.
VSYNC	Vertical Sync	Determines the vertical position of the displayed text.
DISPEN	Display Enable	Indicates the CRTC is providing addressing in the active display area.
CURSOR	Cursor Enable	Indicates a valid cursor address.
LPSTP	Light Pen Strobe	Signal used to latch the current refresh address on the light pen register.
VCC	Power Supply	+5 volt.
VDD	Power Supply	Ground.

FIGURE 4.17

Most signals generated by the 6845 are standard, however, the following deserve further elaboration:

MAO-MA13: The 6845 addresses the screen memory on a linear (or binary) basis. These 14 outputs allow this CRTC to access up to 16K bytes of screen memory or 8 pages.

E: Since the 6845 belongs to the 6800 uP family, it requires this signal to stay in synchronism with the uP. This signal corresponds to Ø2 of a two phase clock, within a 6800-based system.

DISPEN: This is equivalent to the blanking signal of other CRTCs; it is high while the raster scan is within the display area of the CRT.

LPSTB: Since the 6845 supports a light pen input, this signal is used to detect, by means of external hardware, where the light pen has been placed within the display area of the CRT. The screen memory address corresponding to the pen's position on the CRT is latched on the light pen register upon detection.

Internal Circuit Behaviour (Refer to **Figure 4.15**)

This CRTC consists of programmable horizontal and vertical timing generators, programmable linear address register, programmable cursor logic, light pen capture register, and control circuitry to interface to a processor bus.

All CRTC timing is derived from CLK, usually the output of an external dot rate counter. Coincidence (CO) circuits continuously compare counter contents to the contents of the programmable register file, RO-R17. For horizontal timing generation, comparisons result in:

1. Horizontal sync pulse (HS) of a frequency, position, and width determined by the registers;
2. Horizontal Display Signal of a frequency, position, and duration determined by the registers.

The Horizontal counter produces the H clock which drives the Scan Line Counter and Vertical Control. The contents of the Raster Counter are continuously compared to the Max Scan Line Address Register. A coincidence resets the Raster Counter and clocks the Vertical Counter.

Comparisons of Vertical Counter contents and Vertical Registers result in:

1. Vertical sync pulse (VS) of a frequency and position determined by the registers - the width is fixed at 16 raster lines in the vertical control section and is not programmable;
2. Vertical Display of a frequency and position determined by the registers.

The Vertical Control Logic has other functions:

1. Generate row selects, RA0-RA4, from the Raster Count for the corresponding interlace or non-interlace modes.
2. Extend the number of scan lines in the vertical total by the amount programmed in the Vertical Total Adjust Register.

The Linear Address Generator is driven by CLK and locates the relative positions of characters in memory with their positions on the screen. Fourteen lines, MA0-MA13, are available for addressing up to four pages of 4K characters, 8 pages of 2K characters, etc. Using the Start Address Register, hardware scrolling through 16K characters is possible.

The Linear Address Generator repeats the same sequence of addresses for each scan line of a character row.

The cursor logic determines the cursor location, size, and blinking rate on the screen. All are programmable.

The light pen strobe going high causes the current contents of the Address Counter to be latched in the Light Pen Register. The contents of the Light Pen Register are subsequently read by the Processor.

Internal CRTC registers are programmed by the processor through the data bus, D0-D7, and the control signals - R/$\overline{W}$, $\overline{CS}$, RS and E.[23]

Programmable Registers

Of the 19 registers making up the 6845, eighteen are programmable, with the remaining one being an **address register**. Only **two** memory or I/O addresses are required to enter data into the programmable registers. To access any of these registers, **Figure 4.18**, a two-step sequence is followed:

1. the 5-bit write-only address register is loaded with the number of the programmable register that is to be accessed, it serves as **pointer**; and
2. a read or write operation to the addressed register is performed.

The RS (register select) signal is always connected to the system's least significant address bit (A0). Thus, when $\overline{CS}$ and A0 are low, and the data bus contains the number of the register one wishes to access, the address

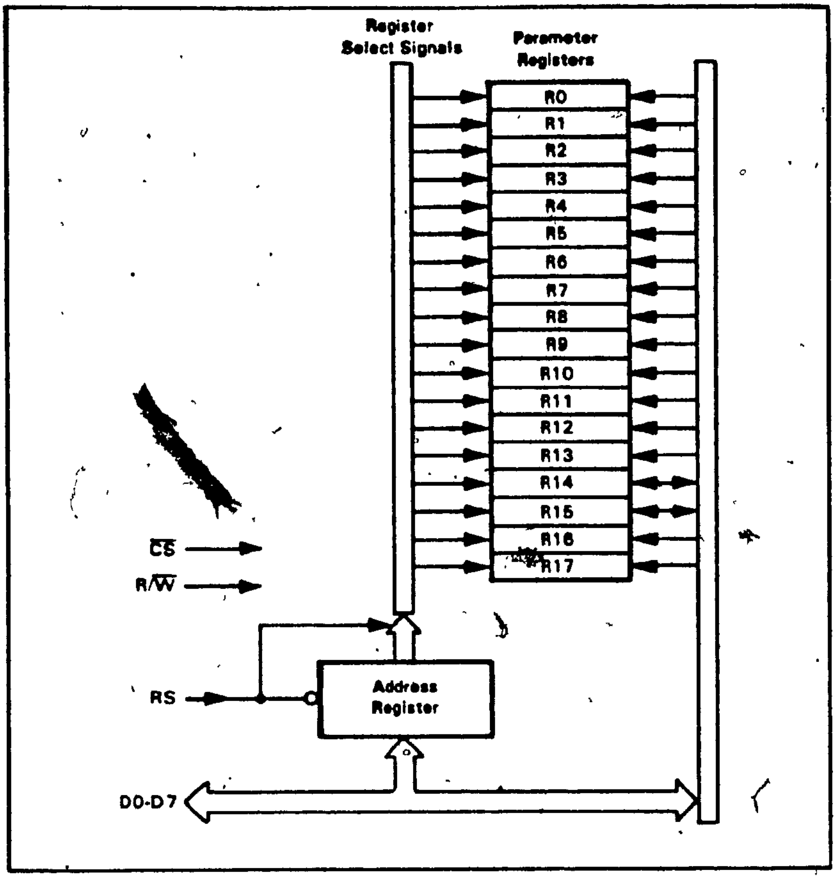


FIGURE 4.18

How to Access the 6845 Registers[12]

	Register			Read (R) Write (W)	Bits	Range — Units
	No.	Name/Function				
Horizontal Format and Timing	0 (00$_{16}$)	Horizontal Total		W	8	1 – 256 (0-FF$_{16}$) CLKs
	1(01$_{16}$)	Characters/Row		W	8	1 – 256 (0-FF$_{16}$) CLKs
	2(02$_{16}$)	HSYNC Position		W	8	1 – 256 (0-FF$_{16}$) CLKs
	3(03$_{16}$)	HSYNC Width		W	4	1 – 16 (0-F$_{16}$) CLKs
Vertical Format and Timing	4(04$_{16}$)	Vertical Total		W	7	1–128 (0-7F$_{16}$) Character Rows
	5(05$_{16}$)	VSYNC Adjust		W	5	1 – 32 (0-1F$_{16}$) Scan Lines
	6(06$_{16}$)	Character Rows/Frame		W	7	1–128 (0-7F$_{16}$) Character Rows
	7(07$_{16}$)	VSYNC Position		W	7	1–128 (0-7F$_{16}$) Character Rows
	8(08$_{16}$)	Interlace Mode		W	2	0-3
	9(09$_{16}$)	Scan Lines/Row		W	5	1 – 32 (0-1F$_{16}$) Scan Lines
Primary Operating Registers	10(0A$_{16}$)	Cursor Start Scan Line		W	7*	1 – 32 (0-1F$_{16}$) Scan Lines
	11(0B$_{16}$)	Cursor Stop Scan Line		W	5	1 – 32 (0-1F$_{16}$) CLKs
	12(0C$_{16}$)	(MSB)	Start Address (Top of Page)	W	6	1 – 16,384 (0000-4FFF$_{16}$)
	13(0D$_{16}$)	(LSB)		W	8	
	14(0E$_{16}$)	(MSB)	Cursor Position	R/W	6	0 – 16,384 (0000-4FFF$_{16}$)
	15(0F$_{16}$)	(LSB)		R/W	8	
	16(10$_{16}$)	(MSB)	Light Pen Position	R	6	0 – 16,384 (0000-4FFF$_{16}$)
	17(11$_{16}$)	(LSB)		R	8	

* Two bits used to specify cursor blink characteristics

FIGURE 4.19
6845 Programmable Registers[12]

register is loaded with such a register number. The next access will either retrieve (read) or store (write) information depending upon the parameter the register pointed to.

The individual parameter controlled by each of the 18 registers is shown in **Figure 4.19.** The registers have been subdivided into groups pertaining to the general function which they perform:

R0 to R3	Horizontal format and timing
R4 to R9	Vertical format and timing
R10 to R17	Cursor characteristics, screen memory addressing and light pen interface

Typically, registers R0 through R11 are loaded upon system power up and are never accessed again. However, R12 to R17 will be accessed on a regular basis during system operation to perform scrolling (R12, R13), to establish the cursor position (R14, R15) and to determine the light pen position (R16, R17).

Interfacing with the uP

Interfacing the 6845 with the uP is relatively simple, especially if the CPU belongs to Motorola's 6800 family of processors, as shown in **Figure 4.20.** However, if another processor controls the system, like the 8085 or Z80, relevant signals must be derived from the control bus to simulate the timing of the E signal (more on this later) and the combination of R/$\overline{W}$ to access this CRTC.

The $\overline{CS}$ signal must be derived from the address bus by means of the address decoder logic. The uP can then transfer parameter information to

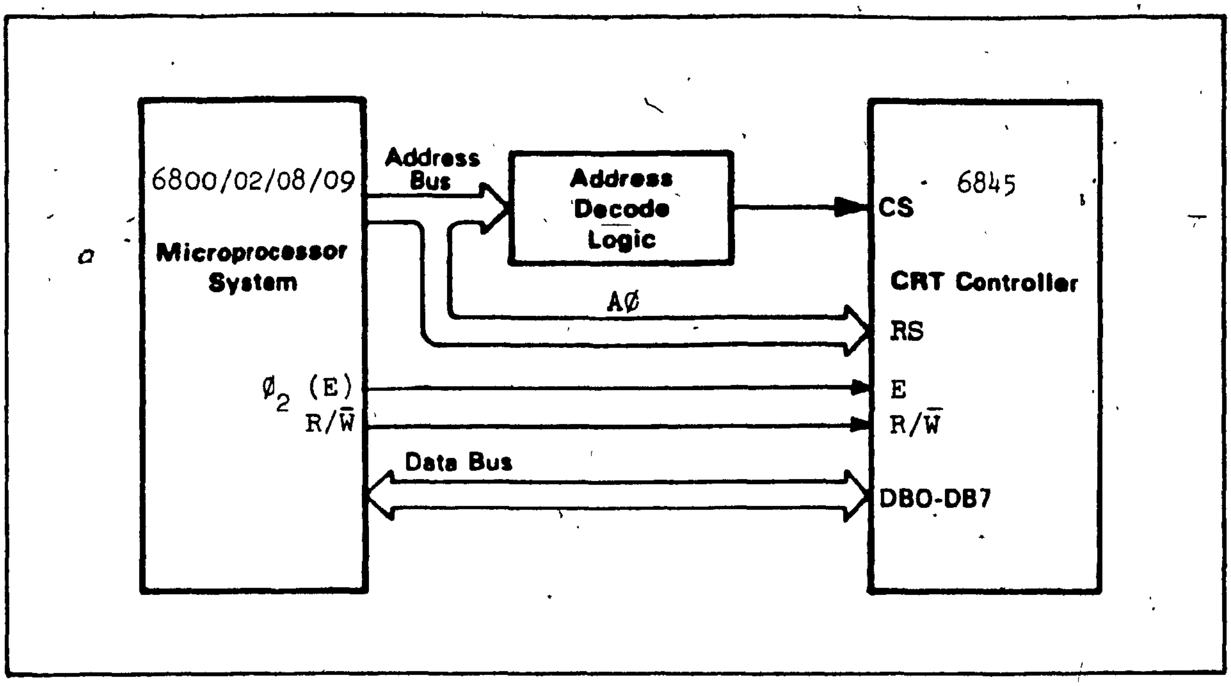
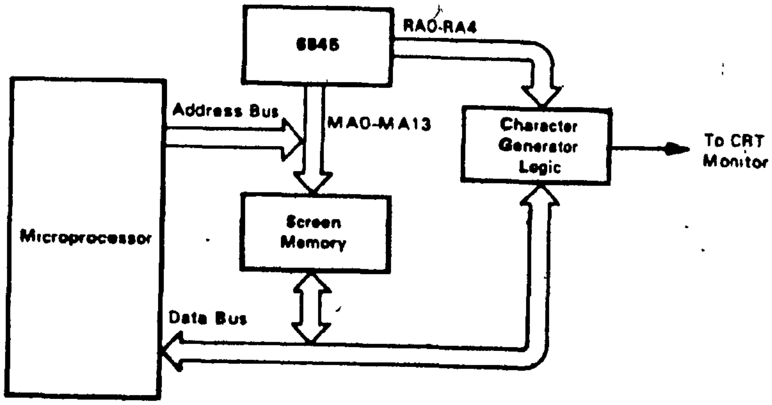


FIGURE 4.20
6845 to uP Interface

FIGURE 4.21

6845 Interface to Character Generator and Video Memory[12]



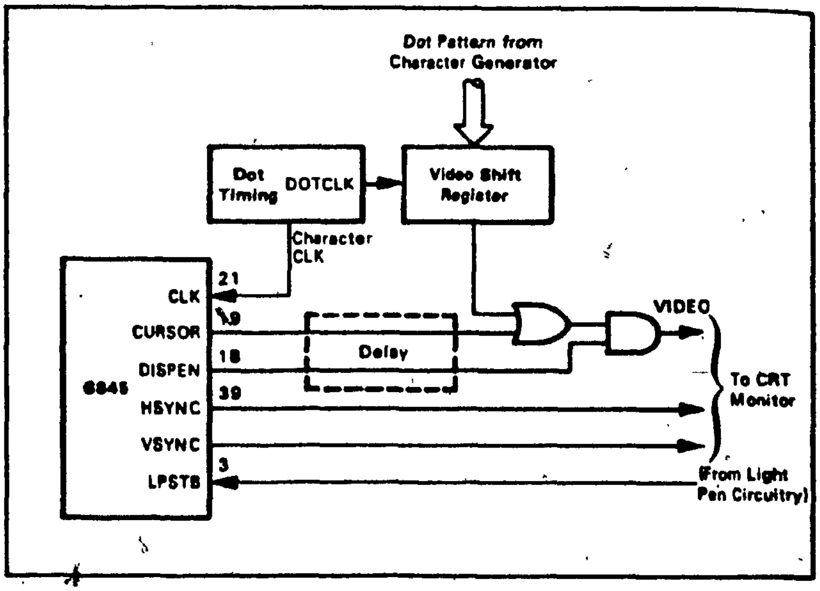


FIGURE 4.22

6845 CRT Monitor Interface Signals[12]

and from the 6845 simply by issuing memory write or read commands to the device. Due to the role played by the **address register**, the 6845 occupies only **two addresses** on the memory map. Hence, the reason for conneting AO to the RS input, as was previously pointed out. Data transfers then take place by means of the data bus. The timing of the above-mentioned signals is straightforward, with the exception of the E signal. Since this input needs a **continuous clock signal** to drive it - as required by all 6800 peripherals - in order to allow adequate data transfers between the CPU and its supporting devices, non-6800 processors may require a complicated interface to generate this signal as per its timing constraints.

Unlike other CRTCs, the 6845 provides no **memory contention**, **DMA**, **interrupt** or **status** signals. Another point to notice is that this CRTC does not have a **data register** either; hence, the data to be displayed goes directly from the video memory to the character generator logic, without intervention on the part of the 6845. **Status** and **command** registers are lacking also. Thus, the uP interacts with the 6845 only on power up or during a reset operation, when the programmable registers are loaded with the pertinent parameter values to generate adequate timing characteristics.

Interfacing with the Character Generator and the Video Memory

Since the 6845 addresses the video memory on a **linear** (or binary) basis, it interfaces with such memory by providing 14 address lines, MAO-MA13, **Figure 4.21.** Registers R12 and R13 contain the **Start Address** (or **Top of Page**), **Figure 4.19,** of the page being displayed, thus, at the beginning of each frame, the 6845's internal address counter is set to the value of such a start address. Thereafter, the contents of these two registers are incremented at the Character Clock (CLK) rate during each scan line. At the end of each scan line, these registers will be once again set to the start address and the cycle is repeated until all scan lines making up a character row have been completed. Once a character row is completed, the address counter is loaded with the address of the first character on the next row and the cycle repeats itself until the last character of the last row and back again to the first character on the top row.

Although the timing to address the video memory is quite straightforward, the 6845 **does not** contain adequate logic to solve the memory contention problem to determine when the CPU or the CRTC may access the video memory without interfering with each other. Such a contention must be resolved by means of external circuitry.

Chapter 3 pointed out three possible methods to deal with memory contention. However, since the 6845 does not generate any DMA signals, it does not lend itself very easily to solve the problem by means of the DMA appoach. However, either the "**video RAM**" or "**transparent**" method may be used. Since the "video RAM" approach causes visible streaks to appear on the CRT screen, we must discard this technique. However, the "transparent" method proves to be the most adequate to implement since the 6845 is driven by a two phase clock and being a 6800 family peripheral, it only accesses the memory while Ø2 is low. Thus, allowing the CPU to carry out video memory updates while Ø2 is high.

With respect to the character generator logic, the 6845 generates 5 raster addresses or scan line counter outputs, **Figure 4.21**. Thus, a character addressed by this CRTC should not contain more than 32 scan lines. Register R9, **Figure 4.19**, controls the number of scan lines per row. Note that the scan line counter is incremented at the Horizontal Sync (HSYNC) rate. Register R8 may be programmed to allow the 6845 to operate on an interlace or non-interlace mode.

Interfacing with the CRT Monitor

The 6845 provides Horizontal Sync (HSYNC) and Vertical Sync (VSYNC) signals as all other CRTCs, **Figure 4.22**. All horizontal and vertical parameters are programmable by means of registers R0 through R3, and R4 through R9, respectively. The 6845 however, does not provide a composite video signal. Nevertheless, HSYNC and VSYNC along with the video signal may be combined by means of external circuitry in order to generate it.

The 6845 not only generates a **cursor** signal whenever the video memory is equal to the address contained in R14, R15, the Cursor Position register, **Figure 4.22**, but it also allows the user to program the cursor shape desired as well as two possible blinking rates. The cursor signal is synchronized to the screen memory address outputs.

The **display enable (DISPEN)** signal is equivalent to the **blanking** signal on other CRTCs. It is set high during the horizontal and vertical retrace periods and it may be used to **turn off** the video signal during these intervals. Generation of the DISPEN signal is independent from

HSYNC and VSYNC signals. However, it does depend on the specified number of displayable characters on the screen. This signal is also synchronized to the video memory address outputs. Since the CURSOR and DISPEN signals may be activated before the corresponding character is sent to the screen, due to video memory and character logic access delays, external circuitry may be required to introduce delays, as shown in **Figure 4.22**, in order to synchronize them appropriately.

The LPSTB (Light Pen Strobe) input signal forces the 6845 to store the current value of video memory address into R16 and R17, the Light Pen registers. Such a storage is synchronized with the Character Clock (CLK) input to the 6845.

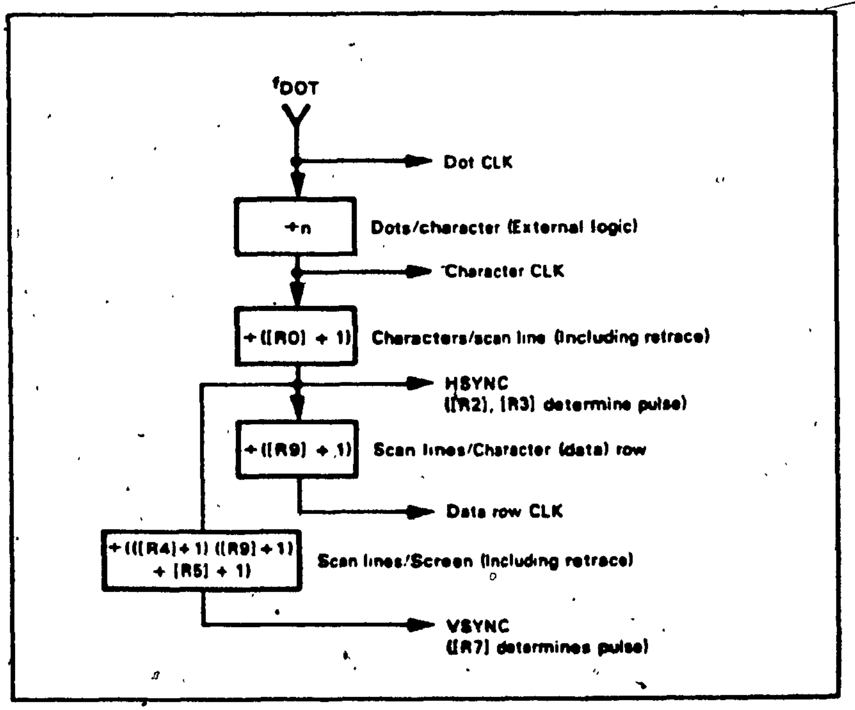
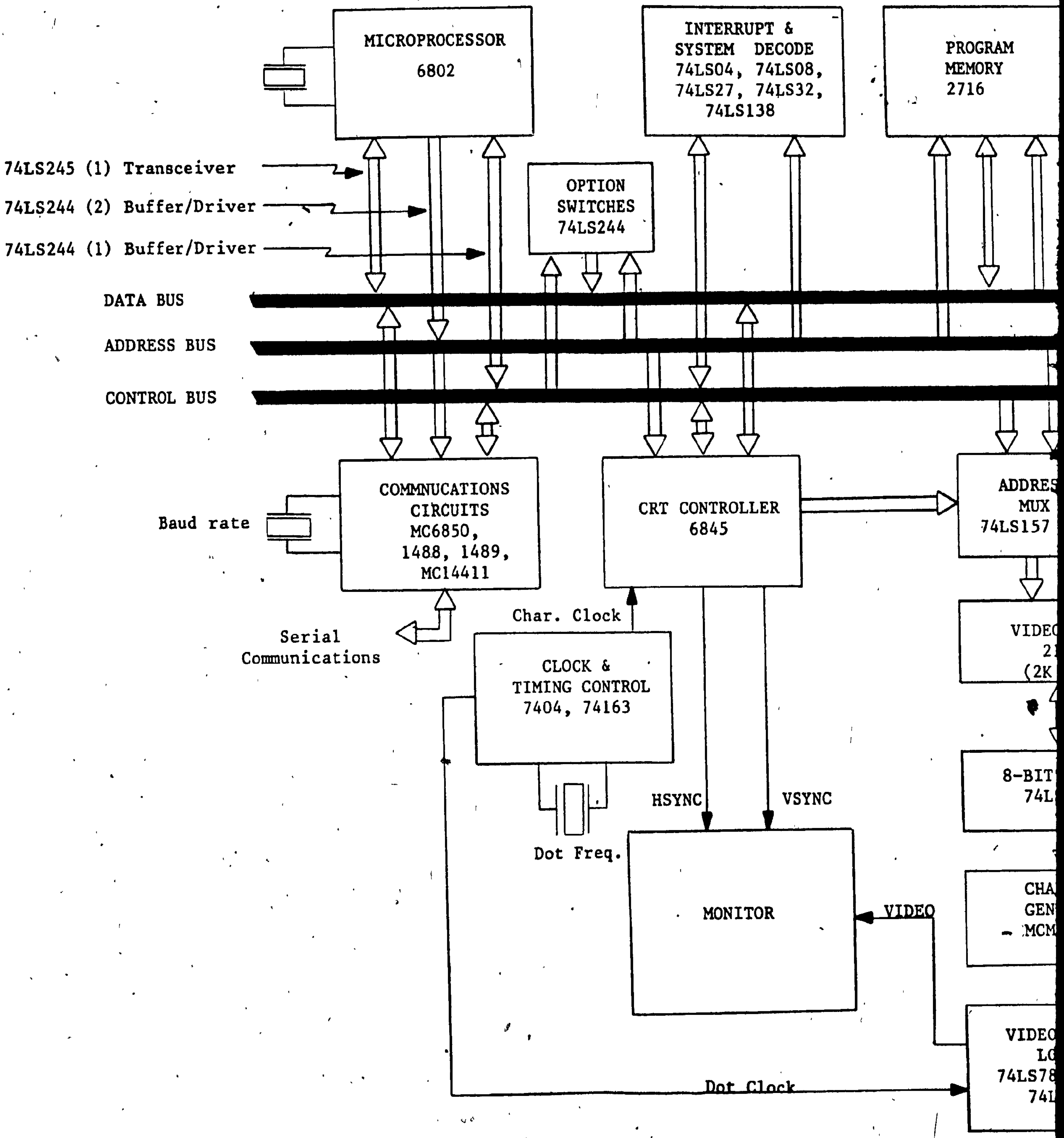

fDOT
Dot CLK
÷n
Dots/character (External logic)
Character CLK
÷([R0] + 1)
Characters/scan line (including retrace)
HSYNC
([R2], [R3] determine pulse)
÷([R9] + 1)
Scan lines/Character (data) row
Data row CLK
÷(([R4]+1) ([R9]+1)
+ [R5] + 1)
Scan lines/Screen (including retrace)
VSYNC
([R7] determines pulse)

FIGURE 4.23
6845 Programmable Registers to CRT Timing Chain Relationship[12]

The relationship between the relevant 6845 programmable registers and the generation of the timing chain signals to interface to the CRT monitor is shown in **Figure 4.23.**

Terminal Implementation Block Diagram

In order to implement the optimized terminal with the characteristics already mentioned, the necessary circuitry was identified and is shown in block diagram form in **Figure 4.24.** The total chip count arrived at was 32.



1 OF

Figure 4.24

Terminal Block Diagram using the 6845 CRT controller

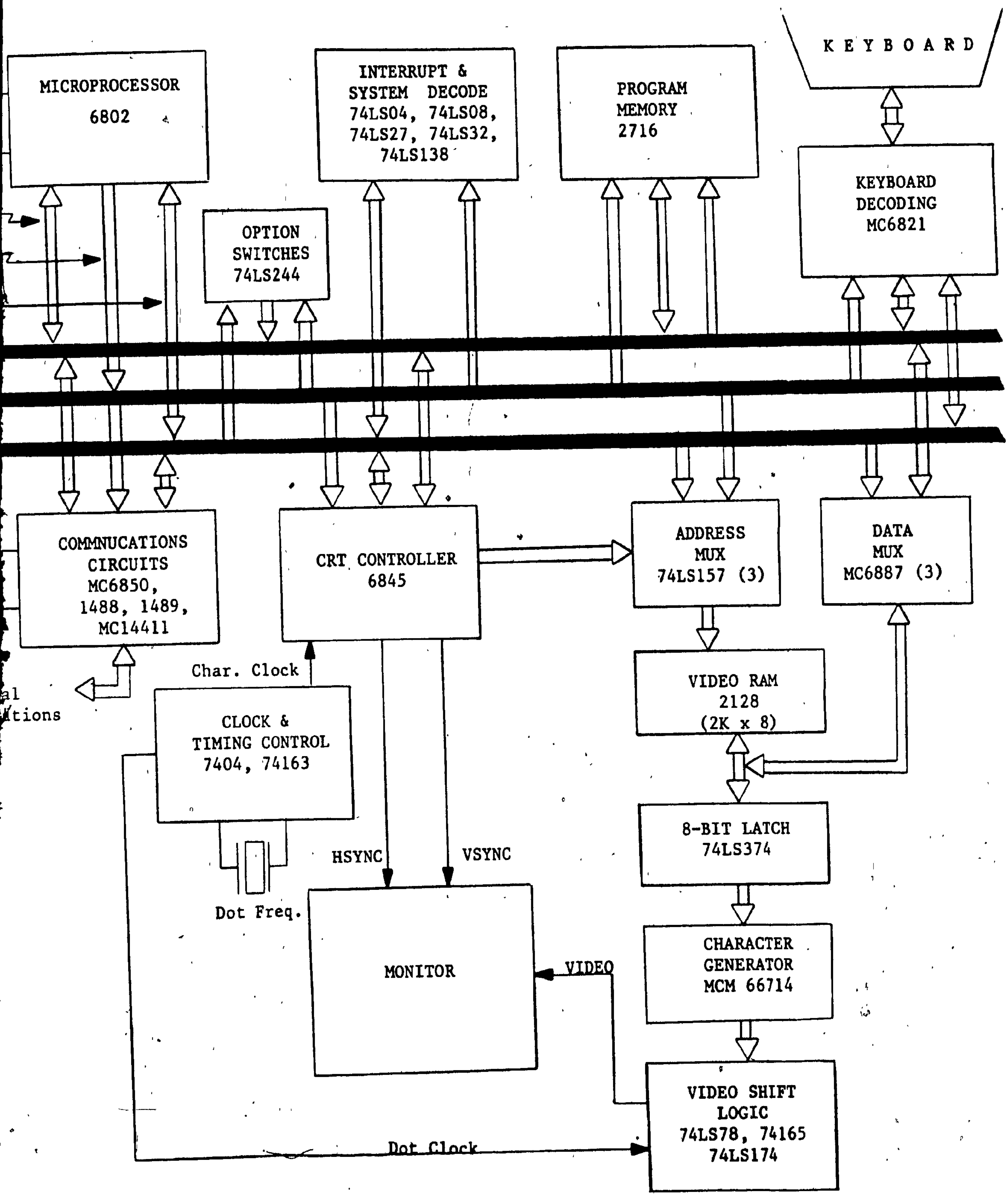


Figure 4.24

Terminal Block Diagram using the 6845 CRT controller

4.3.3 The 8275 CRT Controller

The 8275, like the 6845 is one of the more recently available controllers. Although it was introduced by Intel[28] several years ago, it remained its sole source until only recently when it became available through Western Digital.[29]

The 8275 is quite a sophisticated LSI device intended to interface CRT raster scan displays with Intel's microcomputer systems. It is a 40-pin NMOS device. The circuitry built into the 8275 is shown in **Figure 4.25.** Comparing it to the general block diagram of a CRTC, **Figure 4.1**, we can identify the logic functions built into it, as shown in **Figure 4.26** (shaded areas).

Intel recently introduced an offspring of the 8275, the 8276. It is a CRT controller as well, however, it is less sophisticated and thus has functional limitations. We will not consider this device in the analysis that follows, although most does apply to it being a smaller version of the 8275.[30,31]

General Description

The 8275 is an intelligent controller. In order to operate adequately, it requires a uP and a **direct memory access device**, since the technique utilized by the 8275 to interface with video memory is by means of DMA. The uP initializes this CRTC during power up and also shares system memory with it, instead of needing **separate** video memory like the other CRTCs previously described.

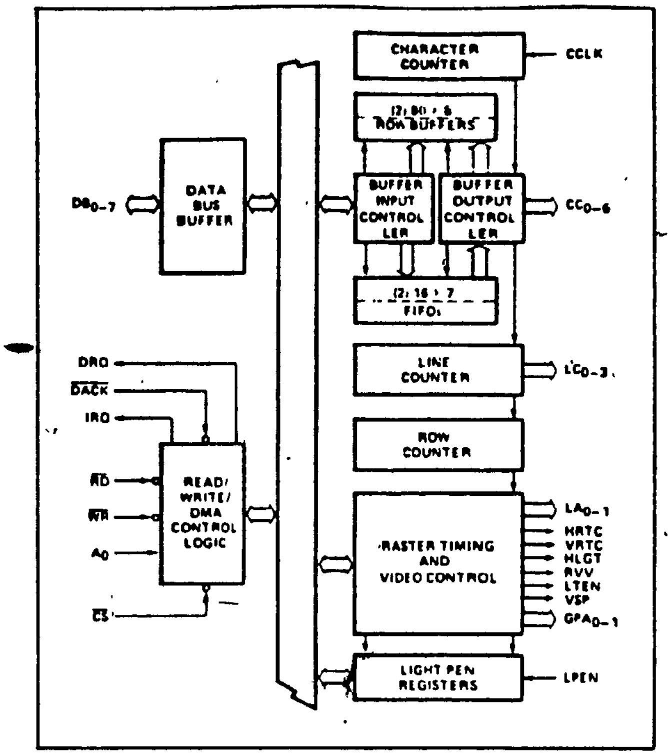


FIGURE 4.25

Intel's 8275 Block Diagram

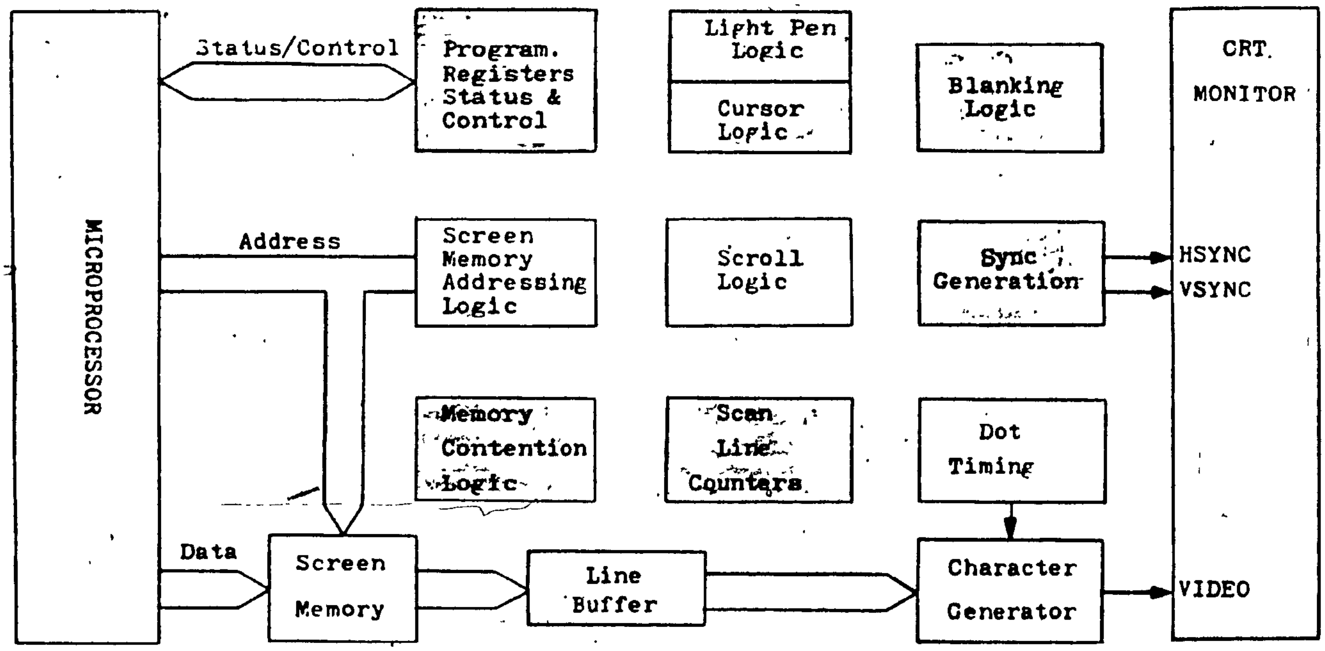


FIGURE 4.26

Functional Logic Built into the 8275 CRT Controller (shown shaded)

The 8275 differs from other CRTCs in that it contains two 80-character buffers. The presence of such buffers and the logic it provides to interface with the uP force the system to use a specific hardware configuration, DMA. Considering **Figure 4.26**, another aspect of this device quite different from other CRTCs, is the absence of any refresh memory addressing logic. This is a direct result imposed upon the system in order to utilize DMA. As such, the DMA controller generates the video memory addresses and loads the data that is to be displayed into the 8275's row buffers.

Another feature provided by this device, which is absent from the other CRTCs so far considered, is the presence of memory contention logic. It consists of those signals used to interface to the DMA controller, the 8257. Thus, all the 8275 has to do is to wait for external logic to provide it with data it has requested.

The 8275 does not contain any scrolling logic. However, the scrolling function is performed via external logic. It is performed transparently as far as this device is concerned.

Although no dot timing logic is provided on-chip, the following signals are generated by this device:

horizontal sync
vertical sync
character codes
scan lines

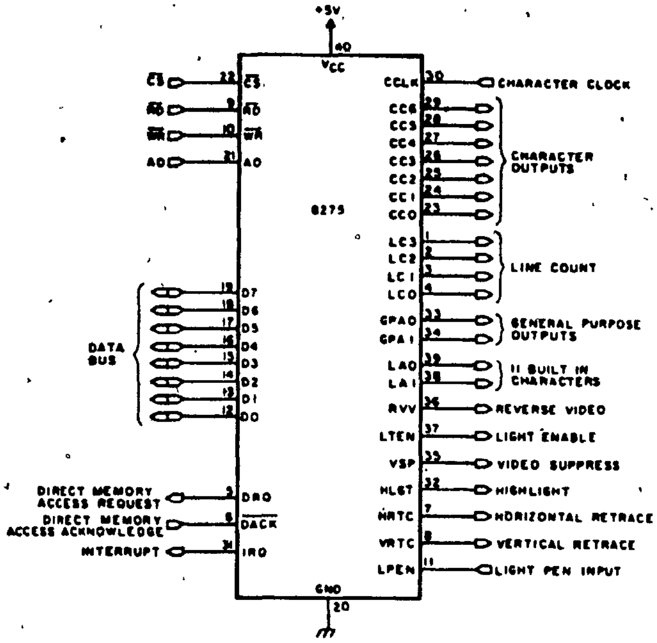
blanking
cursor
reverse video
highlight indicator
line attribute codes
general purpose attribute codes
light pen detect

They are all programmer definable by accessing relevant internal registers.

A distinguishable feature of this device, uncommon to the other CRTCs, is that it allows characters to be displayed either **single** or **double** spaced plus it supports **visual attributes.** Other CRT controllers require additional external logic or a dedicated LSI device to provide such features. Worthy of note, is that the 8275 contains Status, Command as well as Parameter registers.[12,28,32]

Built-in Signals

Figure 4.27 shows the pinout and signal names of the 8275. Similar to the other CRTCs considered, its signals may be divided into four categories, depending upon their function:



Pin Descriptions

Symbol	Pin No	Type	Name and Function
LC$_3$ LC$_2$ LC$_1$ LC$_0$	1 2 3 4	O	**Line Count** Output from the line counter which is used to address the character generator for the line positions on the screen
DRQ	5	O	**DMA Request** Output signal to the 8257 DMA controller requesting a DMA cycle
$\overline{\text{DACK}}$	6	I	**DMA Acknowledge** Input signal from the 8257 DMA controller acknowledging that the requested DMA cycle has been granted
HRTC	7	O	**Horizontal Retrace** Output signal which is active during the programmed horizontal retrace interval During this period the VSP output is high and the LTEN output is low
VRTC	8	O	**Vertical Retrace** Output signal which is active during the programmed vertical retrace interval During this period the VSP output is high and the LTEN output is low
$\overline{\text{RD}}$	9	I	**Read Input** A control signal to read registers
$\overline{\text{WR}}$	10	I	**Write Input** A control signal to write commands into the control registers or write data into the row buffers during a DMA cycle
LPEN	11	I	**Light Pen** Input signal from the CRT system signifying that a light pen signal has been detected
DB$_0$ DB$_1$ DB$_2$ DB$_3$ DB$_4$ DB$_5$ DB$_6$ DB$_7$	12 13 14 15 16 17 18 19	I/O	**Bi-Directional Three-State Data Bus Lines** The outputs are enabled during a read of the C or P ports
Ground	20		Ground

Symbol	Pin No	Type	Name and Function
V$_{CC}$	40		+5V Power Supply
LA$_0$ LA$_1$	39 38	O	**Line Attribute Codes** These attribute codes have to be decoded externally by the dot timing logic to generate the horizontal and vertical line combinations for the graphic displays specified by the character attribute codes
LTEN	37	O	**Light Enable** Output signal used to enable the video signal to the CRT This output is active at the programmed underline cursor position and at positions specified by attribute codes
RVV	36	O	**Reverse Video** Output signal used to indicate the CRT circuitry to reverse the video signal This output is active at the cursor position if a reverse video block cursor is programmed or at the positions specified by the field attribute codes
VSP	35	O	**Video Suppression.** Output signal used to blank the video signal to the CRT This output is active —during the horizontal and vertical retrace intervals. —at the top and bottom lines of rows if underline is programmed to be number 8 or greater —when an end of row or end of screen code is detected —when a DMA underrun occurs —at regular intervals (1/16 frame frequency for cursor 1/32 frame frequency for character and field attributes —to create blinking displays as specified by cursor character attribute or field attribute programming
GPA$_0$ GPA$_1$	34 33	O	**General Purpose Attribute Codes** Outputs which are enabled by the general purpose field attribute codes
HLGT	32	O	**Highlight** Output signal used to intensify the display at particular positions on the screen as specified by the character attribute codes or field attribute codes
IRQ	31	O	**Interrupt Request**
CCLK	30	I	**Character Clock (from dot timing logic)**
CC$_6$ CC$_5$ CC$_4$ CC$_3$ CC$_2$ CC$_1$ CC$_0$	29 28 27 26 25 24 23	O	**Character Codes** Output from the row buffers used for character selection in the character generator
$\overline{\text{CS}}$	22	I	**Chip Select** The read and write are enabled by CS
A$_0$	21	I	**Port Address** A high input on A$_0$ selects the C port or command registers and a low input selects the P port or parameter registers

FIGURE 4.27

8275 CRT Controller Pinout & Signal Description[28,32]

uP System Interface Signals	Video Memory & Character Generator Signals	CRT Monitor Interface Signals	Power Signals
AO	CCO-CC6	GPAO, GPA1	VCC
CCLK	LCO-LC3	HGLT	GND
$\overline{\text{CS}}$		HRTC	
$\overline{\text{DACK}}$		LAO, LA1	
DO-D7		LTEN	
DRQ		RVV	
IRQ		VRTC	
$\overline{\text{RD}}$		VSP	
$\overline{\text{WR}}$			

TABLE 4.3

8275 CRTC Signal Classifications

The majority of these signals generated by the 8275 are standard, however, the following are only common to this device:

DRQ & $\overline{\text{DACK}}$: used as handshake signals with the 8257 DMA controller.

GPAO, GPA1: enable general purpose attributes.

HGLT: highlight signal to intensify a particular character(s).

INT: interrupt request to the CPU.

LAO, LA1: line attribute codes.

LPEN: light pen - equivalent to the same function on other CRTCs.

LTEN: light enable - used for the cursor and for positions specified by attribute codes.

VSP: video suppression - equivalent to the blanking signal on other CRTCs and used as well under special conditions in the case of the 8275.

Available Registers

Before describing the programmable registers, it must be pointed out that the 8275 contains a **Status**, as well as a **Command** register. **Figure 4.28** shows all the available registers. Accessing any of these registers depends upon the following signal levels:

$\overline{\text{CS}}$	**A0**	**$\overline{\text{RD}}/\overline{\text{WR}}$**	**Register Accessed**
0	1	$\overline{\text{RD}}$	Status
0	1	$\overline{\text{WR}}$	Command
0	0	$\overline{\text{RD}}$	Parameter Reg.
0	0	$\overline{\text{WR}}$	Parameter Reg.
1	X	X	**None**

While the Status and Command registers are read and write-only, and may be accessed by the uP at any time, some of the Parameter registers are read-only or write-only and they must be accessed as part of a command sequence. To access the latter, a byte must be written into the Command register, followed by a read or write command to the relevant register.

The Status Register

The contents of this register reflects the predominant conditions taking place within the 8275, **Figure 4.29.** The uP can read its contents at any time.

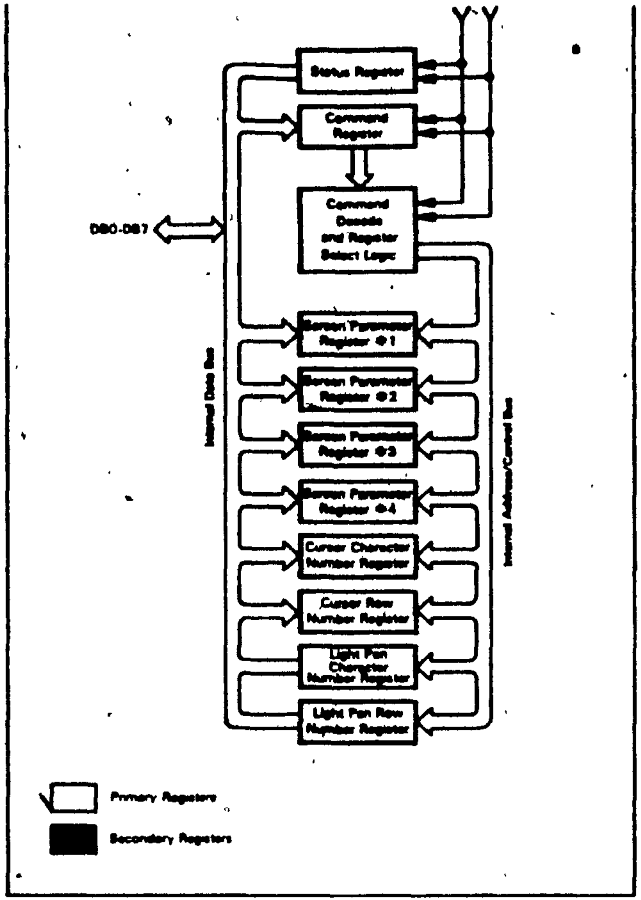

Status Register
Command Register
Command Decode and Register Select Logic
DB0-DB7
Internal Data Bus
Internal Address/Control Bus
Screen Parameter Register #1
Screen Parameter Register #2
Screen Parameter Register #3
Screen Parameter Register #4
Cursor Character Number Register
Cursor Row Number Register
Light Pen Character Number Register
Light Pen Row Number Register
Primary Registers
Secondary Registers

FIGURE 4.28

8275 Available Registers[12]

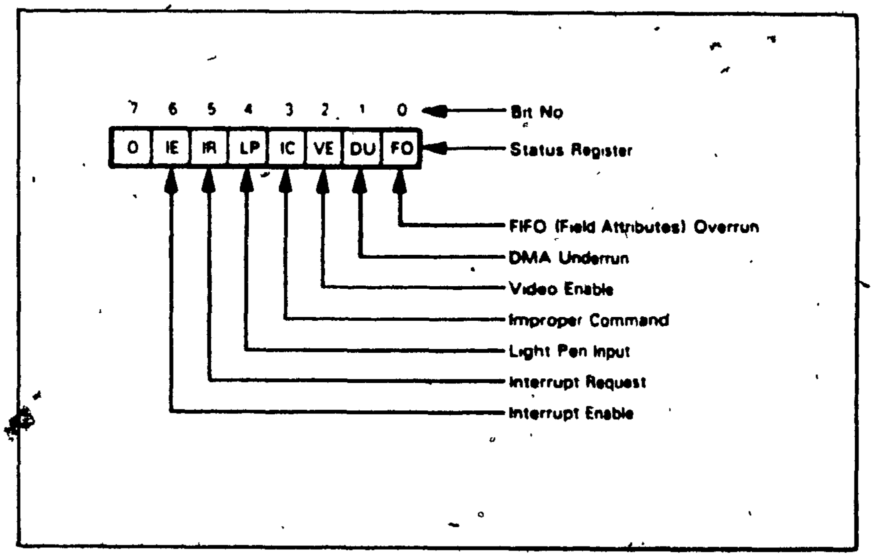


FIGURE 4.29

8275 Status Register Bit Assignments[12]

The Command Register

A total of 8 commands may be issued to the 8275, as shown in the table of **Figure 4.30**. Any command may be issued at any time, as long as the following signals hold true:

$$A0 = 1, \overline{CS} = 0, \overline{WR} = 0$$

The contents of the data bus (D0-D7) containing the command byte will then be input to the 8275.

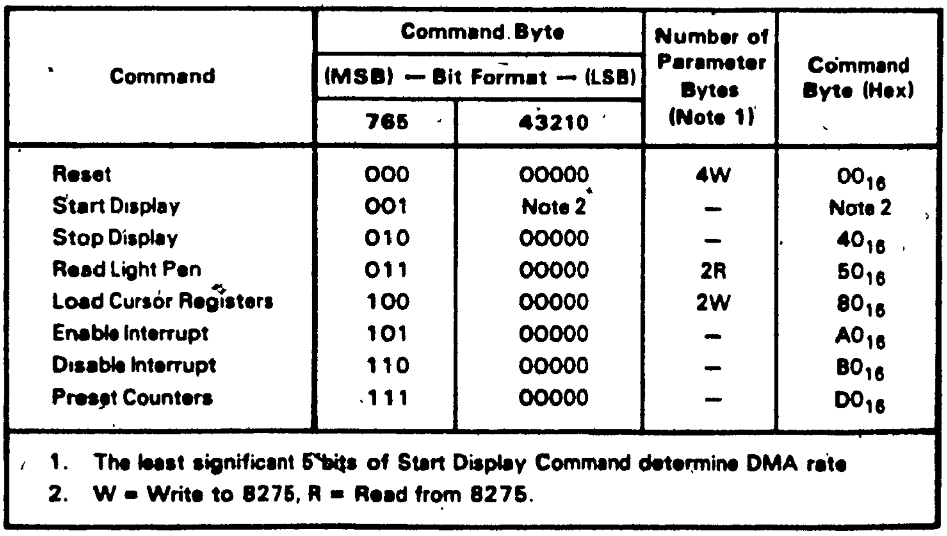
Each command byte is sub-divided in two (**Figure 4.30**):

1. the 3 MSBs determine the command to be executed; and
2. the 5 LSBs are set to 0 except for the **Start Display** command.

Three of the commands have parameter bytes associated with them:

(a) Reset and Load Cursor require the CPU to write additional parameter bytes to the 8275 after the command byte.

(b) The Read Light pen command requires two byte readings after issuing the command.



Command	Command Byte (MSB) — Bit Format — (LSB) 765	 43210	Number of Parameter Bytes (Note 1)	Command Byte (Hex)
Reset	000	00000	4W	00_{16}
Start Display	001	Note 2	—	Note 2
Stop Display	010	00000	—	40_{16}
Read Light Pen	011	00000	2R	50_{16}
Load Cursor Registers	100	00000	2W	80_{16}
Enable Interrupt	101	00000	—	$A0_{16}$
Disable Interrupt	110	00000	—	$B0_{16}$
Preset Counters	111	00000	—	$D0_{16}$

1. The least significant 5 bits of Start Display Command determine DMA rate
2. W = Write to 8275, R = Read from 8275.

FIGURE 4.30

8275 Command Register Summary[12]

Programmable Registers

Upon power up, the CPU accesses the 8275 by issuing a RESET command in order to allow device initialization. As shown in **Figure 4.30**, the RESET command must be followed by 4 write operations. The contents of these 4 bytes establishes the basic operating characteristics of the 8275. Such an operation loads the screen parameter registers shown in **Figure 4.28**, with the desired screen format parameters. **Figure 4.31** shows the detailed and tabular format of these 4 parameter registers. The available programmable parameters are:

Single or double spaced rows
Horizontal characters per row*
Vertical Retrace row count*
Vertical rows per frame*
Underline placement
Number of lines per character row*
Line counter mode*
Field attribute mode
Cursor format
Horizontal retrace count*

The characteristics denoted with an asterisk (*) are entered during power up and never change afterwards. The others are altered depending on the data that is to be displayed, at the programmer's discretion. Note that each byte allows more than one parameter to be entered, thus reducing the number of required registers. This is contrary, and more efficient, to the format used by other CRTCs that assign one parameter to each register, thus increasing the number of registers needed and the time to program the registers.

Reset Command:

	OPERATION	A_0	DESCRIPTION	DATA BUS MSB							LSB
Command	Write	1	Reset Command	0	0	0	0	0	0	0	0
Parameters	Write	0	Screen Comp Byte 1	S	H	H	H	H	H	H	H
	Write	0	Screen Comp Byte 2	V	V	R	R	R	R	R	R
	Write	0	Screen Comp Byte 3	U	U	U	U	L	L	L	L
	Write	0	Screen Comp Byte 4	M	F	C	C	Z	Z	Z	Z

Action – After the reset command is written, DMA requests stop, 8275 interrupts are disabled, and the VSP output is used to blank the screen. HRTC and VRTC continue to run. HRTC and VRTC timing are random on power-up.

As parameters are written, the screen composition is defined.

Parameter – S Spaced Rows

S	FUNCTIONS
0	Normal Rows
1	Spaced Rows

Parameter – HHHHHHH Horizontal Characters/Row

H	H	H	H	H	H	H	NO. OF CHARACTERS PER ROW
0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	2
0	0	0	0	0	1	0	3
1	0	0	1	1	1	1	80
1	0	1	0	0	0	0	Undefined
1	1	1	1	1	1	1	Undefined

Parameter – VV Vertical Retrace Row Count

V	V	NO. OF ROW COUNTS PER VRTC
0	0	1
0	1	2
1	0	3
1	1	4

Parameter – RRRRRR Vertical Rows/Frame

R	R	R	R	R	R	NO. OF ROWS/FRAME
0	0	0	0	0	0	1
0	0	0	0	0	1	2
0	0	0	0	1	0	3
1	1	1	1	1	1	64

Parameter – UUUU Underline Placement

U	U	U	U	LINE NUMBER OF UNDERLINE
0	0	0	0	1
0	0	0	1	2
0	0	1	0	3
1	1	1	1	16

Parameter – LLLL Number of Lines per Character Row

L	L	L	L	NO. OF LINES/ROW
0	0	0	0	1
0	0	0	1	2
0	0	1	0	3
1	1	1	1	16

Parameter – M Line Counter Mode

M	LINE COUNTER MODE
0	Mode 0 (Non-Offset)
1	Mode 1 (Offset by 1 Count)

Parameter – F Field Attribute Mode

F	FIELD ATTRIBUTE MODE
0	Transparent
1	Non-Transparent

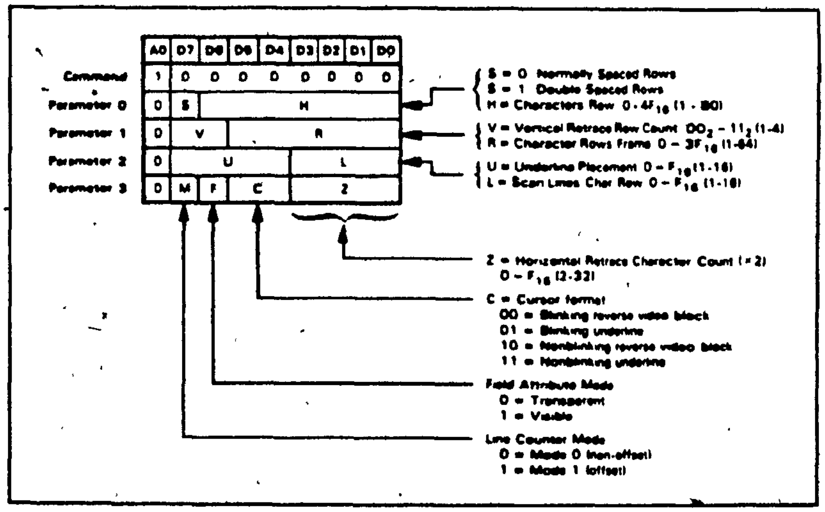
Parameter – CC Cursor Format

C	C	CURSOR FORMAT
0	0	Blinking reverse video block
0	1	Blinking underline
1	0	Nonblinking reverse video block
1	1	Nonblinking underline

Parameter – ZZZZ Horizontal Retrace Count

Z	Z	Z	Z	NO. OF CHARACTER COUNTS PER HRTC
0	0	0	0	2
0	0	0	1	4
0	0	1	0	6
1	1	1	1	32

Note: uuuu MSB determines blanking of top and bottom lines (1 = blanked, 0 = not blanked)



	A0	D7	D6	D5	D4	D3	D2	D1	D0
Command	1	0	0	0	0	0	0	0	0
Parameter 0	0	S	H						
Parameter 1	0	V		R					
Parameter 2	0	U				L			
Parameter 3	0	M	F	C		Z			

S = 0 Normally Spaced Rows
S = 1 Double Spaced Rows
H = Characters Row $0 - 4F_{16}$ (1 - 80)

V = Vertical Retrace Row Count $00_2 - 11_2$ (1-4)
R = Character Rows Frame $0 - 3F_{16}$ (1-64)

U = Underline Placement $0 - F_{16}$ (1-16)
L = Scan Lines Char Row $0 - F_{16}$ (1-16)

Z = Horizontal Retrace Character Count (+2)
$0 - F_{16}$ (2-32)

C = Cursor format
00 = Blinking reverse video block
01 = Blinking underline
10 = Nonblinking reverse video block
11 = Nonblinking underline

Field Attribute Mode
0 = Transparent
1 = Visible

Line Counter Mode
0 = Mode 0 (non-offset)
1 = Mode 1 (offset)

FIGURE 4.31
8275 Detailed and Tabular Format of Programmable
Parameter Registers Accessed when Issuing a RESET Command[12,28]

The following limitations about the 8275 deserve special notice:

1. The maximum number of displayable characters per row is 80.
2. The horizontal and vertical retrace pulses generated by this device do not allow either front or back porch delays to be specified. **(Figures 3.14 and 3.15)**; external logic must be incorporated to generate such delays and position these pulses adequately.
3. The maximum number of scan lines per character row is 16.

The cursor control register allows one of four programmable options to be displayed:

1. underline;
2. reverse video block;
3. blinking;
4. non-blinking.

Interfacing with the uP[12,28,29,31,32,33]

Although the structure which the 8275 is built upon is to make use of the DMA technique, interfacing may be accomplished without utilizing it.[33] Regardless, the same handshake signals and timing sequences must take place in order to use the 8275 adequately.

The transfer of data to be displayed on the CRT must go from the video memory (which in this case is part of the **system's memory**) to the 8275 under the control of the DMA controller, **Figure 4.32**. However, when the CPU has to issue commands and parameters to the 8275 or to read the contents of its Status register, it accesses the device by issuing either an I/O read or I/O write operation. The 8275 occupies 2 I/O port locations, thus the need for the A0 signal, **Figure 4.27**.

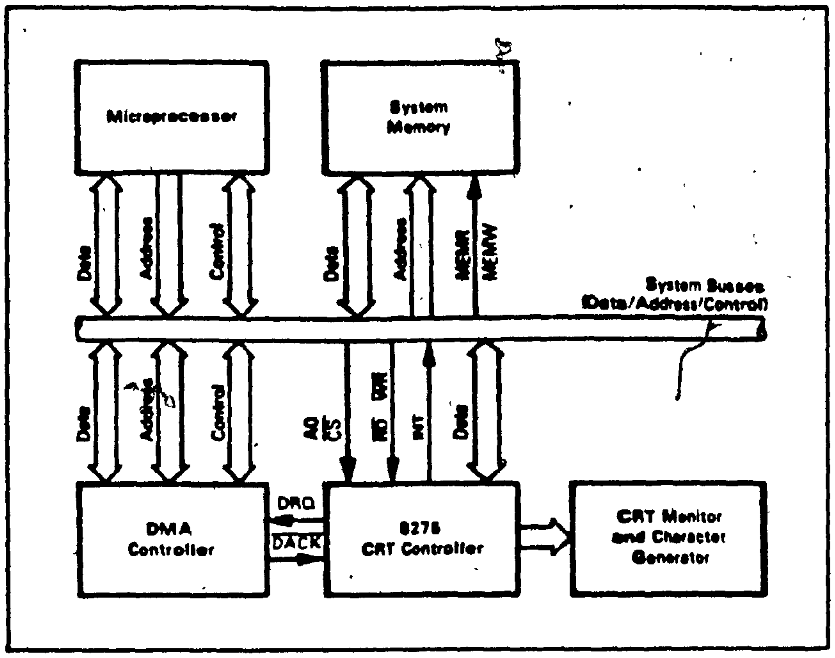


FIGURE 4.32

Relationship between the 8275, the DMA controller, the uP and memory within the system[12]

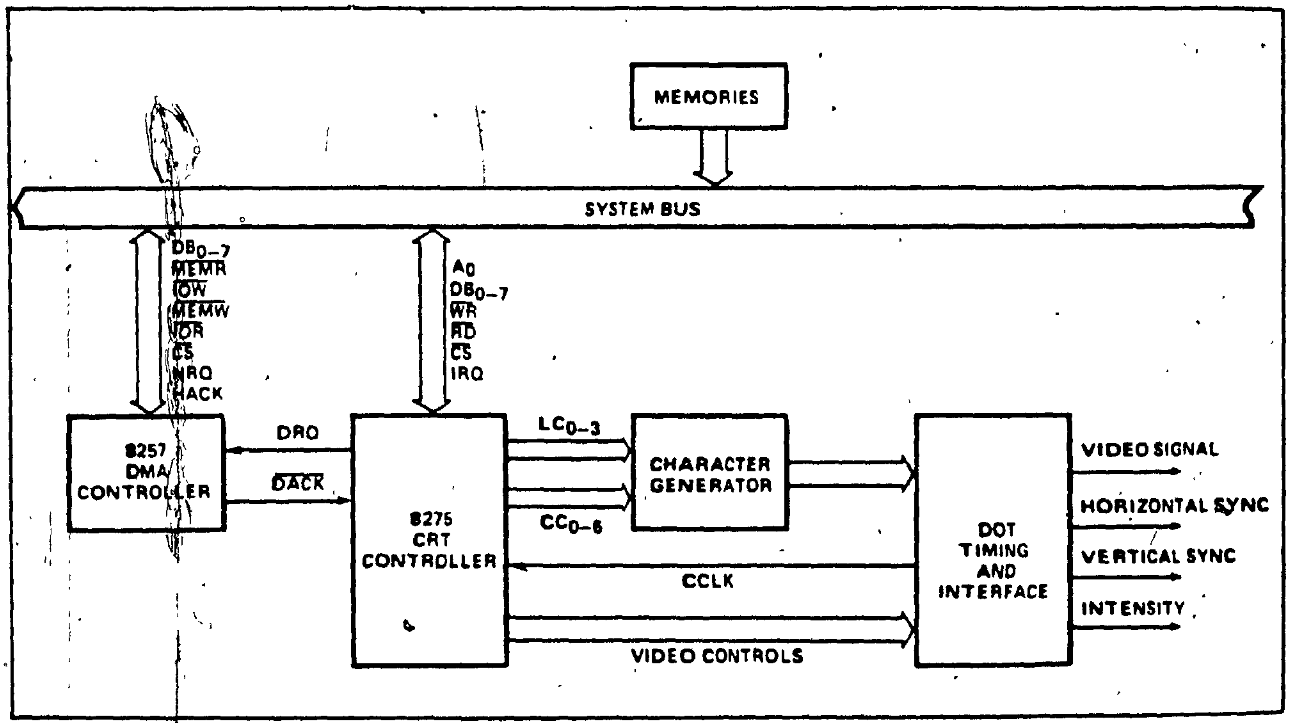


FIGURE 4.33

System Block Diagram showing System Operation during data transfers between the memory and the 8275[28]

The protocol between the uP and the 8275 takes place mostly upon system power up, when device initialization is established. However, the CPU must also access this CRTC to move the cursor and to initiate and terminate the display of information. Hence, transferring of data between the 8275 and the video (system) memory takes place with almost no uP involvement at all. It is conducted by the interaction between the DMA controller and the 8275. **Figure 4.33** shows the components involved in such a transfer. Note the uP has not been shown due to its non-interaction during such a transfer.

As the 8275 requires data for its character row buffers, it generates DMA requests by setting its DRQ signal high. The DMA controller responds by gaining control of the system buses, reading a byte of data from system memory (using $\overline{\text{MEMR}}$), and simultaneously writing that byte into the 8275's character row buffer (using $\overline{\text{IOW}}$).

The 8275 DRQ signal is reset when the $\overline{\text{DACK}}$ and $\overline{\text{WR}}$ signals are received back from the DMA controller. Such a signal combination loads the byte of data being presented to the 8275 on its data bus lines into the character row buffer. During DMA operations, the $\overline{\text{CS}}$ signal remains high and the $\overline{\text{DACK}}$ signal from the DMA controller performs the chip select. This is logical since the address information of the system buses contains video memory addresses during these DMA cycles.

The previous discussion pointed out the transfer of a single byte of data, however, data transfers may take place by one of two modes:

1. on a **byte-by-byte** basis with each DMA cycle being individually initiated; or
2. on a **burst mode** where several bytes of data are transferred to the 8275 in response to a single request.

The 8275 may be programmed to operate in either of these modes.

In order to access the proper system memory space and obtain the video data, the DMA controller receives the video memory starting address from the uP upon system initialization. Upon each DMA request from the 8275, the DMA controller increments its address counter for each byte of memory transferred so that it may be able to access the next character in the video memory upon the next DMA request. At the beginning of the last row of characters of the video memory for each frame, the 8275 generates an interrupt to the CPU in order to reload the DMA controller address register with the starting address of the video memory of the present page being displayed, in preparation for the next frame of data.

The 8275 contains two 80-character (bytes) row buffers in order to allow loading of one buffer from the video memory under the control of the DMA controller, while the contents of the other buffer are being presented to the character generator logic as every scan line of each character row is sent to the CRT monitor. The switching back and forth between the two row buffers is controlled internally by the 8275, hence, the operation is completely transparent to the uP system.

Interfacing with the Character Generator

Like other CRTCs, the 8275 generates 7 output lines, the character codes (CC0-CC6), corresponding to the character being displayed on the CRT at the Character Clock (CCLK) rate. CC0-CC6 are the outputs from the 8275's internal character row buffer. Since each character row is made up of several scan lines, for every scan line the row buffer is recirculated (or shifted) at the Character Clock rate in order for the code of each character to be output to the character generator.

The 8275 generates as well, four scan line outputs (LC0-LC3) from its scan line counter, thus allowing a maximum of 16 scan lines per character row. These lines are held steady during each horizontal scan line while the character codes are being output.[1] After every scan line is completed, the scan line counter is incremented to address the next line of dots for the character codes that will be output.

Although limited, the 8275 provides the user the option of generating certain graphic characters. This is accomplished via the **Line Attribute** (LA0-LA1) signals. Since video data is stored in 8-bit bytes, and since an ASCII character only uses 7 of these bits, with bit 8 equal to 0, when the 8275 receives a byte of data from the video memory with the MSB set to 1, it recognizes this as a control code. Control codes may be of three types:

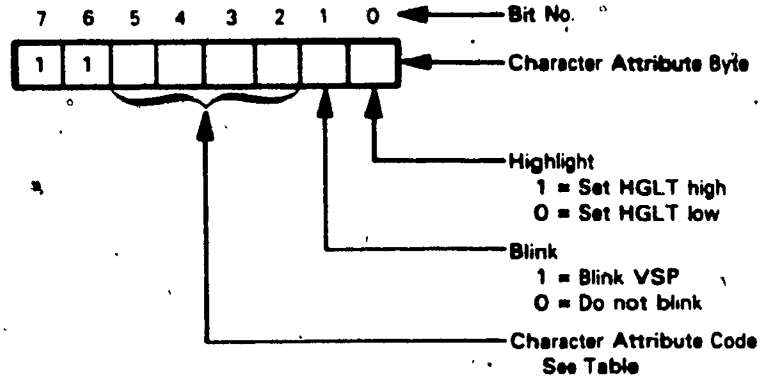
1. **Character Attribute Codes:** they cause line attribute outputs to be manipulated.
2. **Field Attribute Codes:** they affect the visual characteristics of a group of characters that are to be displayed.
3. **Special Codes:** these affect DMA and screen blanking.

However, the graphic capabilities that the 8275 provides may be implemented at the expense of additional external hardware.

Character Attributes: This code is defined by the character attribute byte. These codes are used to generate graphic symbols without the use of a character generator. As such, they may control the display output and generate 11 symbols as shown in Figure 4.34. Blinking and highlighting of any character is controlled by this code as well. Figure 4.35 shows the typical (external) character attribute logic necessary to generate such symbols.

Field Attributes: The Field Attribute byte code affects all data bytes which follow the code until another attribute control byte is read from the video memory. Hence, all the characters between these two control bytes, the "field", will have the characteristics defined by the Field Attribute Control byte. Figure 4.36 shows the byte's format. Besides controlling the highlight, blinking, reverse video (RVV), and underlining, this byte is also used to manipulate the General Purpose Attribute signals, a total of 4. These signals have interesting applications since they may be used to control the color in color CRT monitors or to select alternate character generator logic to implement special characters within a field.

Special Codes: Four additional control codes are recognized by the 8275; they affect DMA operation and screen blanking. They are shown in Figure 4.37.



Character Attribute Code		Outputs				Symbol	Description
Bits 6432	For Scan Lines	LA1	LA0	VSP	LTEN		
0000	Above Underline Underline Below Underline	0 1 0	0 0 1	1 0 0	0 0 0	┌	Top Left Corner
0001	Above Underline Underline Below Underline	0 1 0	0 1 1	1 0 0	0 0 0	┐	Top Right Corner
0010	Above Underline Underline Below Underline	0 1 0	1 0 0	0 0 1	0 0 0	└	Bottom Left Corner
0011	Above Underline Underline Below Underline	0 1 0	1 1 0	0 0 1	0 0 0	┘	Bottom Right Corner
0100	Above Underline Underline Below Underline	0 0 0	0 0 1	1 0 0	0 1 0	┬	Top Intersect
0101	Above Underline Underline Below Underline	0 1 0	1 1 1	0 0 0	0 0 0	┤	Right Intersect
0110	Above Underline Underline Below Underline	0 1 0	1 0 1	0 0 0	0 0 0	├	Left Intersect
0111	Above Underline Underline Below Underline	0 0 0	1 0 0	0 0 1	0 1 0	┴	Bottom Intersect
1000	Above Underline Underline Below Underline	0 0 0	0 0 0	1 0 1	0 1 [illegible]	─	Horizontal Line
1001	Above Underline Underline Below Underline	[illegible] 0 0	1 1 1	0 0 0	[illegible]	│	Vertical Line
1010	Above Underline Underline Below Underline	0 0 0	1 0 1	[illegible] 0 0	[illegible] 1 [illegible]	┼	Crossed Lines
1011	Above Underline Underline Below Underline	0 0 0	[illegible]	0 [illegible]	0 [illegible]		Not Recommended*
[illegible]	Above Underline Underline Below Underline	0 0 0	[illegible]	1 1 1	[illegible]		Special Codes
	Above Underline Underline Below Underline	Undefined					Illegal
	Above Underline Underline Below Underline	Undefined					Illegal
[illegible]	Above Underline Underline Below Underline	Undefined					Illegal

*Character Attribute Code 1011 is not recommended for normal operation. Since none of the attribute outputs are active, the character generator will not be disabled and an indeterminate character will be generated.
Character Attribute Codes 1101, 1110 and 1111 are illegal.

FIGURE 4.34

Character Attribute Byte Format, Codes and Resultant Symbols[12,28,31,32,33]

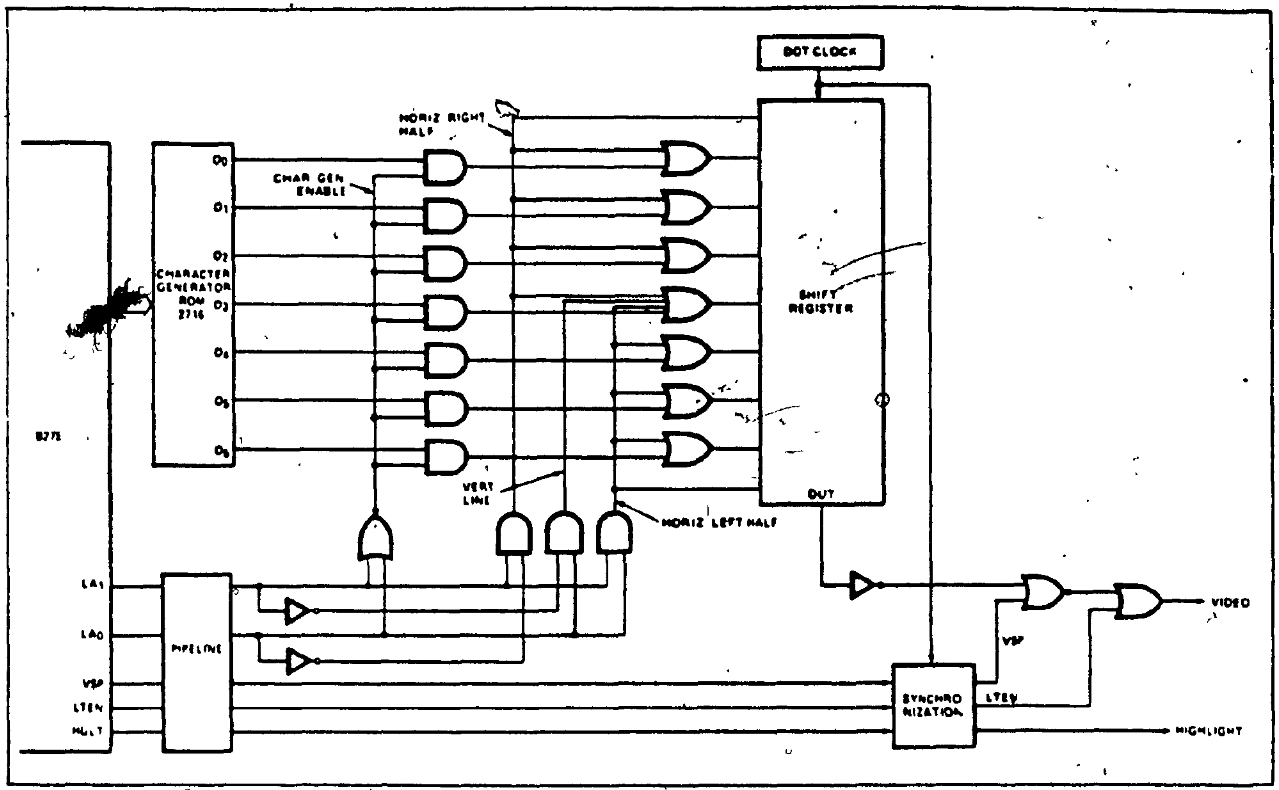


FIGURE 4.35

Typical Character Attribute Logic[28]

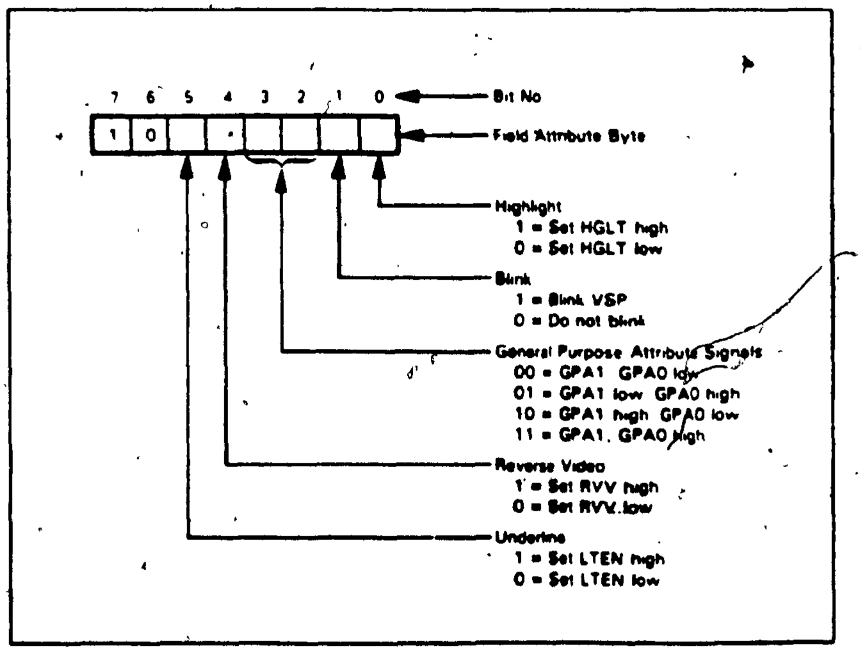


FIGURE 4.36

Field Attribute Control Byte Format[12]

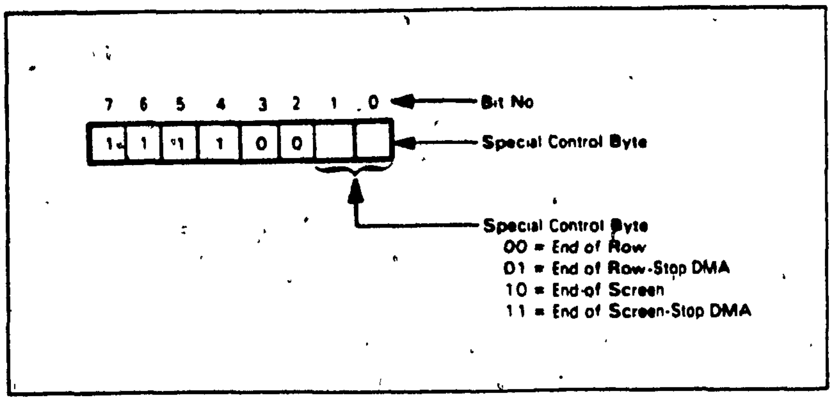


FIGURE 4.37

Special Control Codes Byte Format[12]

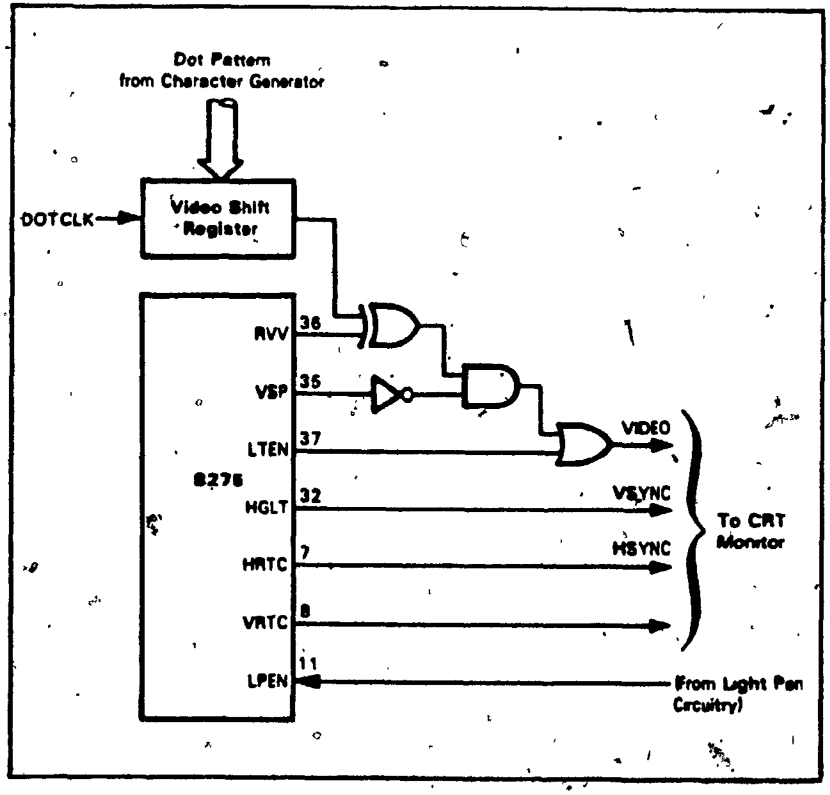


FIGURE 4.38

8275 CRT Monitor Interface Signals[12]

Interfacing with the CRT Monitor

Figure 4.38 depicts the signals involved to implement this interface. The 8275 provides HRTC (Horizontal Retrace) and VRTC (Vertical Retrace) signals, the equivalent of HSYNC and VSYNC on other CRTCs. The parameter registers accessed during a RESET command, upon power up (see Programmable Registers), allow the programmer to enter the frequency and pulse intervals of these two signals as per system requirements. However, the 8275 provides somewhat limited control over these two signals since front and back porch delays are not programmable and must be implemented by means of external hardware in order to position these pulses adequately.

Other signals making up the CRT monitor interface are:

Highlight (HGLT)
Light Enable (LTEN)
Light Pen (LPEN)
Reverse Video (RVV)
Video Supression (VSP)

The LTEN, RVV and VSP can all be combined with the output from the video shift register, **Figure 4.38**, to produce different visual effects on the screen. The HGLT signal as well can add to the visual effects by displaying characters at different intensity levels. Finally, the LTEN input allows external circuitry to trigger the controller to store the contents of the character and row counter values in the 8275 light pen register, **Figure 4.28**.

Terminal Implementation Block Diagram

Designing a smart terminal built around the 8275 CRTC using the DMA technique, consisted of evaluating the minimum circuitry involved to implement it while making sure that the circuitry in question could provide enough flexibility and versatility to give the terminal enough intelligence. The circuitry was identified, its block diagram is shown in **Figure 4.39.** A total of 42 chips were used to implement the terminal.

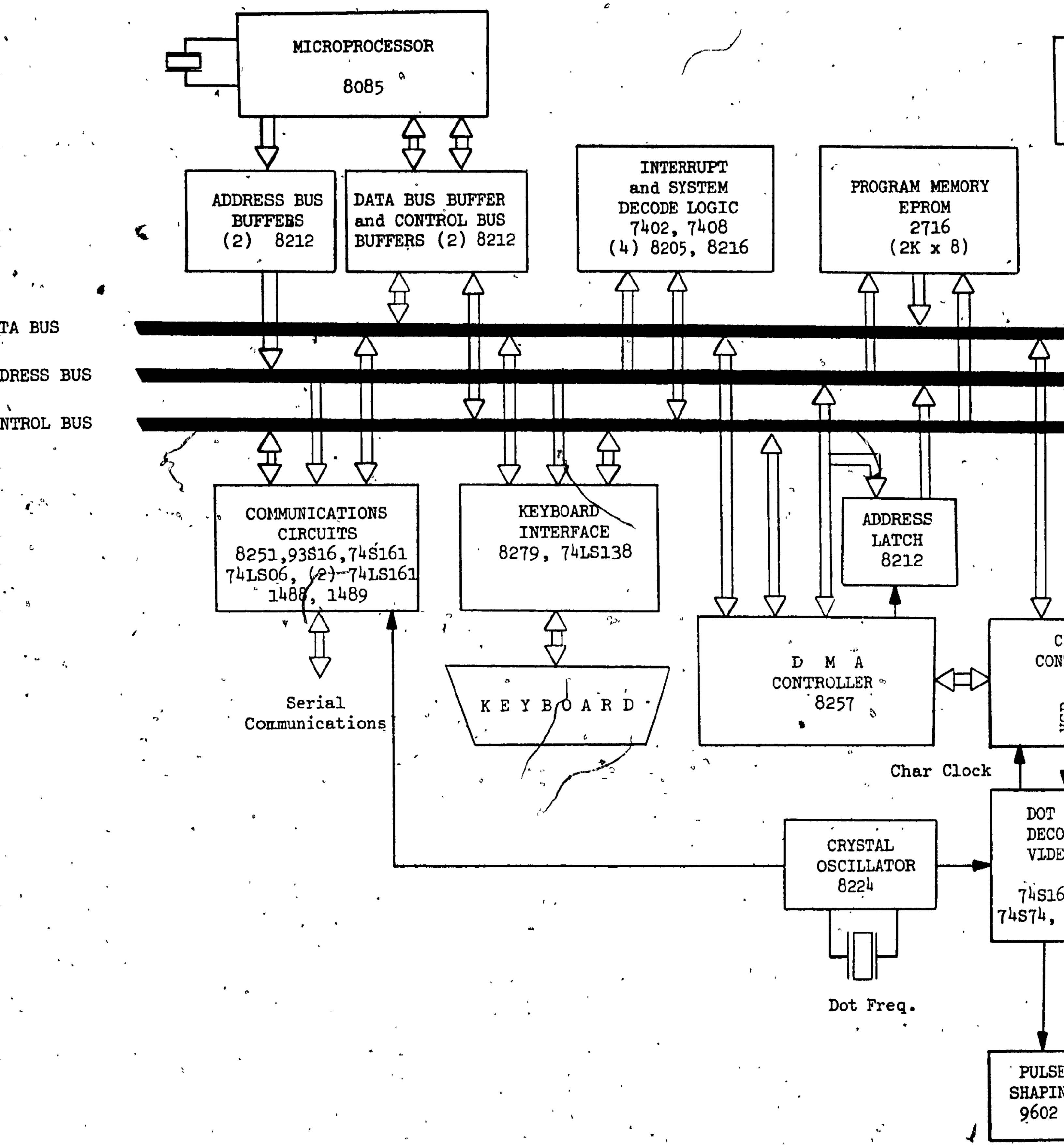
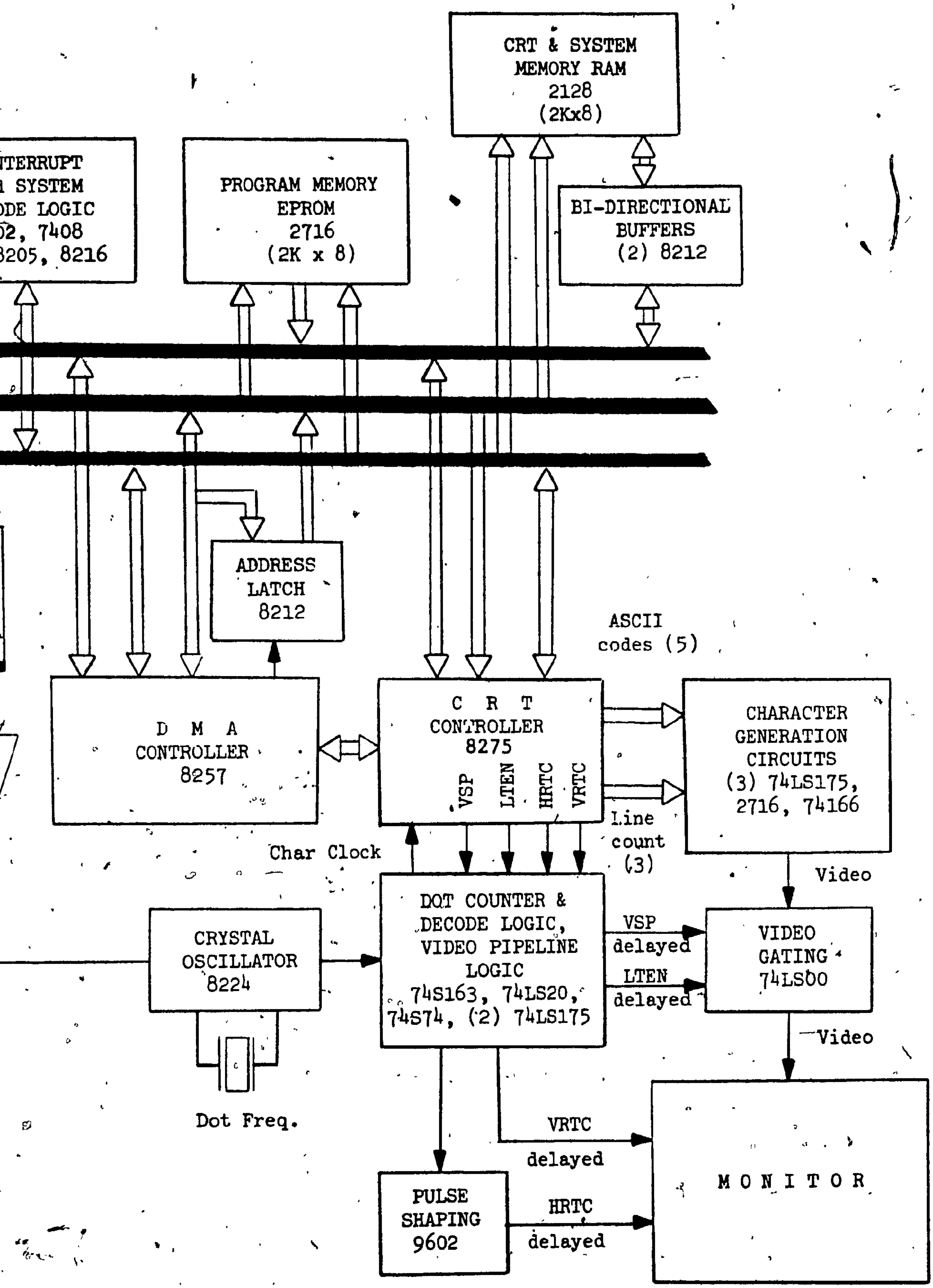

MICROPROCESSOR
8085
ADDRESS BUS BUFFERS (2) 8212
DATA BUS BUFFER and CONTROL BUS BUFFERS (2) 8212
INTERRUPT and SYSTEM DECODE LOGIC 7402, 7408 (4) 8205, 8216
PROGRAM MEMORY EPROM 2716 (2K x 8)
TA BUS
DRESS BUS
NTROL BUS
COMMUNICATIONS CIRCUITS 8251,93S16,74S161 74LS06, (2) 74LS161 1488, 1489
KEYBOARD INTERFACE 8279, 74LS138
ADDRESS LATCH 8212
D M A CONTROLLER 8257
Serial Communications
K E Y B O A R D
C
CON
Char Clock
DOT
DECO
VIDE
74S16
74S74,
CRYSTAL OSCILLATOR 8224
Dot Freq.
PULSE
SHAPIN
9602

Figure 4.39

Terminal Block Diagram using the 8275 CRTC [28,32,33,34]

(total component count is 42 ICs)



4.39

the 8275 CRTC [28,32,33,34]

count is 42 ICs)

4.4 Controller Comparison

In the course of our analysis we have considered 3 CRTCs that operated under 3 completely different approaches in order to implement the controller function. Memory interfacing, depending on the device's internal structural design, may have been carried out by one of the following techniques:

1. a video RAM interface;
2. a DMA interface; or
3. a transparent memory interface.

By the same token, the device's internal structure determined if the memory addressing method utilized was either:

1. linear; or
2. by row/column.

However, while each device allows the circuit designer to specify many different options, under program control , he must exercise extreme care when evaulating the different controllers and before reaching a final decision. He must be careful to check not only the CRTC but also the associated circuitry required and the total number of external components needed to build a complete system. Some controllers require more than others, as illustrated by **Figures 4.14, 4.24 and 4.39.** Another important point to be taken into consideration, is the effect the controller's internal structure has on the software. Some CRTCs raise the software overhead of the uP.

Up to this point, we have had a chance to carry out an in-depth study of the different capabilities and characteristics of three off-the-shelf CRTCs individually. Now, the tables summarizing the controllers signals, **Table 4.4**, their characteristics, **Table 4.5**, and the necessary components, by function, required to build a smart terminal system, **Table 4.6**, will allow us to establish a common basis for a comparison in order to facilitate the task of deciding upon which CRT controller is best suited for the final design.

From the three tables and the detailed analysis of each CRTC we can discard the 8275 right away, for the following reasons:

> Although it is quite a versatile device, its main disadvantage lies in the need for the use of DMA; a must since the 8275 has been conceived to operate under such a memory contention method without alternatives. As a result, the hardware overhead required to implement a low cost smart terminal becomes considerable. The implications arising from the extra hardware are obvious:
>
> 1. higher power consumption - thus larger power supply and more weight;
> 2. larger printed circuit area needed to house the circuitry;
> 3. lower reliability; and
> 4. higher costs, as a result of all the above.
>
> The end result will be a more expensive and bulkier terminal. Undesirable these days, especially if complexity and money can be saved.
>
> Other points against the 8275 CRTC are:
>
> - it is 8080/8085 bus-oriented and is hard to use (needs additional hardware) with non-Intel processors;

- difficult to program due to its complexity, thus resulting in software overhead;
- the use of Field Attribute Control bytes increases the amount of video memory per displayed page thus complicating the system software in order to keep track of the pointer locations and the size of the screen memory;
- additional hardware required to generate front and back porch delays for the horizontal and vertical sync signals.

Although the 8275 has many pros going for it, other cons exist as well (besides those just mentioned), however, the above more than justify the decision against its use.

The choice between the 5037 and the 6845 is not simple to make, especially considering the versatility and ease of use of both CRTCs. However:

1. the chip count is lower in the case of the system implemented with the 5037;
2. the use of the CRT-8002 chip, a companion to the 5037, further reduces the chip count, the PCB complexity and power supply requirements, since it contains on-chip:
   - the video shift register (up to 20 MHz dot clock);
   - the character generator logic; and
   - the attribute logic;

   in the case of the 6845 all of the above require additional hardware;
3. The 5037 is easily interfaced with 8085, Z80 or 6800 series processors, whereas the 6845 is a 6800 family device and requires additional hardware to interface to other processors which may be more powerful than Motorola's.

All of the above give the 5037 a considerable edge over the 6845. Thus, the CRT controller used to implement the smart terminal will be the 5037. Other important factors for such a choice will become more evident as we describe the terminal's design in the next chapter.

Signal Subdivision	Description	Mostek, SMC, SSS, TI CRT 5037 VTAC	
Micrprocessor System Interface Signals	Data Bus	DB0-DB7	
	Register Address (or Select)	A0-A3	
	Chip Select	CS	
	Read and/or Write	$\overline{DS}$ --	
	Character Clock Rate	DCC	
	Enable Sychronization Signal	--	
	Reset	--	
	Interrupt Request	--	
	DMA Request	--	
	DMA Acknowledge	--	
Character Generator Signals	Scan Line	R0-R3 (4)	
	Character Code	--	
Screen (Video) Memory Signals	Memory Address	H0-H7 (8) horizontal DR0-DR4 (5) data row (row/column addressing)	(li
CRT Monitor Signals	Horizpntal Sync	HSYNC	
	Vertical Sync	VSYNC	
	Composite Sync	CSYNC	
	Blanking	BL (blanking)	DISPEN (
	Cursor	CRV (cursor video)	CURSOR (
	Light Pen	--	LPSTB (L
	Reverse Video	--	
	Highlight Indicator	--	
	Attributes	--	
	General Attributes	--	
Power Signals	+5V, GND, (V_{DD})	V_{CC}, GND, (+12V)	

TABLE 4.4
CRT Controller Signal Compari

1OF

tek, SMC, SSS, TI CRT 5037 VTAC	Motorola MC 6845 CRTC	Intel 8275 CRTC	Type
DBO-DB7	DO-D7	DBO-DB7	Bidirectional (8)
AO-A3	RS	AO	Input (1)
CS	$\overline{CS}$	$\overline{CS}$	Input (1)
$\overline{DS}$	R/$\overline{W}$	$\overline{RD}$	Input (1)
--	--	$\overline{WR}$	Input (1)
DCC	CLK	CCLK	Input (1)
--	E	--	Input (1)
--	$\overline{RESET}$	--	Input (1)
--	--	IRQ	Input (1)
--	--	DRQ	Input (1)
--	--	$\overline{DACK}$	Input (1)
RO-R3 (4)	RAO-RA4 (5)	LCO-LC3 (4)	Output
--	--	CCO-CC6 (7)	Output
-H7 (8) horizontal	MAO-MA13 (14)	--	Output
O-DR4 (5) data row	(linear addressing)	--	Output
/column addressing)		(linear addr. using DMA technique)	
HSYNC	HSYNC	HRTC	Output (1)
VSYNC	VSYNC	VRTC	Output (1)
CSYNC	--	--	Output (1)
BL (blanking)	DISPEN (Display Enable)	VSP (Video Suppression)	Output (1)
RV (cursor video)	CURSOR (Cursor Enable)	LTEN (Light Enable)	Output (1)
--	LPSTB (Light Pen Strobe)	LPEN (Light Pen Detect)	Input (1)
--	--	RVV	Output (1)
--	--	HGLT	Output (1)
--	--	LAO,LA1 (Line Attribute Code)	Output (2)
--	--	GPAO,GPA1 (Gen. Purpose Attr.)	Output (2)
V_{CC}, GND, (+12V)	V_{CC}, V_{SS}, (-)	V_{CC}, GND, (-)	Input (2 or 3)

TABLE 4.4
CRT Controller Signal Comparison

2 of 2

FUNCTION	DESCRIPTION	SMC CRT 5037	MOTORLA MC 6845	INTEL 8275
Power	CRT Power Supplies Required	+5V, +12V	+5V	+5V
Availability	Second Sources	Mostek Solid State Scientific Texas Instruments	AMI+ Hitachi Rockwell+ Synertek+ (+ modified version of MC6845)	Western Digital
Cursor	Readable	Yes	Yes	No
	Writeable	Yes	Yes	Yes
	Incrementable	No	No	No
	Size	Programmable	Programmable	Location Only
	Mode Controls	Block	Blink, block, underline, reverse video	Blink, block, underline, reverse video
CRT /nchronization Signals	**Horizontal Timing Control**			
	Front Porch	Yes	Yes	only total retrace time
	Width	Yes	Yes	
	Back Porch	Yes	Yes	
	Vertical Timing Control			
	Front Porch	Yes	Yes	only total retrace time
	Width	3 scan lines	16 scan lines	
	Back Porch	Yes	Yes	
	Composite Sync Available	Yes (non-interlace only)	No*	No*
	Interlaced scan	Yes	Yes (2 modes)	No
Video (Refresh) Memory	Addressing Method	Row/Column 8-column(or horiz. counts) 5- row 13 bits	Linear 14 bits	Linear 16 bits
	Addressing Limits	0 -> 8 K	0 -> 16 K	0 -> 64 K
Character Generation	Dots per character (max.)	9	10	10
	Characters per row	20-132	1-256	1-80
	Scan Lines per row	1-16	1-32	1-16
	Rows per frame	1-64	1-128	1-64
	Character rate	4.0 MHz (typ.)	2.5 MHz	3.1 MHz
	Number of Displayable Characters	128 ASCII Graphics with	128 ASCII plus	128 ASCII plus

1OF

Character Generation	Dots per character (max.)	9	10	10
	Characters per row	20-132	1-256	1-80
	Scan Lines per row	1-16	1-32	1-16
	Rows per frame	1-64	1-128	1-64
	Character rate	4.0 MHz (typ.)	2.5 MHz	3.1 MHz
	Number of Displayable Characters	128 ASCII Graphics with external circuits	128 ASCII plus 37 graphics	128 ASCII plus 11 graphics*
	Thin Graphics	Yes	No	Yes
Attributes	Blink	Yes	No*	Yes
	Reverse Video	Yes	No*	Yes
	Highlight	Yes	No*	Yes
	Blank	Yes	No*	No
	Underline	Yes	No*	Yes
	Graphics	No*	No*	See Thin Graphics
Split Screen	Capabilities	No	No	Software
Line Frequency	50 Hz/60 Hz operation	Software	Software	Software
Memory-Contention	DMA required	No	No	Yes
	Memory-Contention Logic	No	No	DMA
	Memory-Contention circuit required	Yes	Yes	No
	On-chip buffer	No	No	Two, 80-bytes each
Miscellaneous	Additional components needed to be able to achieve system implementation	CRT 8002 (attribute controller)	None	8257 (DMA controller)
	Total number of programmable registers	9 + command register (cursor, scroll, top-of-page)	18 + address register (screen format & timing registers, cursor, top-of-page)	8 + status + command registers (cursor, parameters, screen format & timing)
	Light pen register	No	Yes	Yes
	On-chip dot timing	No	No	No
	Microprocessor family	8080/8085 with interface	6800	8080/8085
	Technology	NMOS	NMOS	NMOS
	Pins	40	40	40

xternal hardware required to implement

TABLE 4.5
Summary of CRT Controller Functional Characteristics

MANUFACTURER	CRT CONTROLLER	CHARACTER GENERATOR	VIDEO & ATTRIBUTE CONTROL	KEYBOARD
STANDARD MICROSYSTEMS CORPORATION (SMC)	CRT 5037 74LS157 (4)	CRT 8002*	CRT 8002* 74160	KR-9600
MOTOROLA	MC6845 CRTC 74LS157 (3)	MCM66714 MC6887 (3) 74LS374	7404, 74LS78 74163, 74165, 74LS174	MC6821 (PIA
INTEL (USING DMA)	8275 CRTC 8257 DMA 74S163	2716 (EPROM) 74LS175 (3) 74LS00	74LS20, 74S74, 74LS175(2), 74166 9602	8279 74LS138

*PERFORMS A DUAL FUNCTION.

TABLE 4.6

SYSTEM ORGANIZATION AND I

KEYBOARD	COMMUNICATIONS	RS 232C INTERFACE	SYSTEM SUPPLY REQUIREMENTS	MINIMAL SYSTEM PARTS COUNT
KR-9600	COM 8216 (BAUD RATE GEN.) COM 8251A (USART)	1488 1489	+5, ±12V	28
MC6821 (PIA)	MC6850 (ACIA) MC14411 (BAUD RATE GEN.)	1488 1489	+5, ±12V	32
8279 74LS138	8251 (USART) 74LS06, 93S16 74LS161(2) (BAUD RATE GEN.)	1488 1489	+5, ±12V	42

TABLE 4.6

IZATION AND IMPLEMENTATION

2 of 2

SECTION 3

DESIGN

CHAPTER 5

THE TERMINAL'S HARDWARE

As shown so far, terminal design is an intricate subject requiring carefully "thought of" decisions in order to provide adequate operator ease-of-use and system flexibility. This chapter will be devoted to the detailed description of the hardware making up the 32-chip "smart" video terminal design, along with the design decisions and reasons for implementing it as it was. The terminal has been named "SAVE" (SmArt compact VidEo terminal) reflecting its characteristics of low power, smartness, speed, compactness and low cost.

5.1 Design Specifications

The design of the system in question, as well as any other system, must start with a design specification. Above all, the designer must know what to design for, where to begin and up to what extent, i.e., when to stop, in order not to make the final product bulky, expensive and a never-ending project where the design gets redesigned over and over, trying to come up with an ideal product. This is where the design specifications play a major role in providing the designer with a global picture of the final product while indicating the limitations as well as outlining its features.

The design of the subject terminal proceeded from the following (preliminary) specifications:

- low power consumption;
- minimal component count possible;
- low cost;
- clean display exhibiting no "snow" and no flicker;
- 24 lines of 80 characters (with provision to implement other formats easily);
- upper and lower case alphanumeric character set with descendens incorporated for such letters as g, j, p, q and y;
  (the descender requirements will have a direct effect on the number of scanned lines per character, thus affecting the maximum dot clock frequency from which other relevant frequencies are derived.)
- easily readable high resolution character matrix;
- video attributes: underline, reverse video, character blinking;
- editing features: character insert/delete, line insert/delete, tab operations, full/partial screen erase;
- one page of text available (with ease of upgrading to 2 or more pages);
- several data communications ports (RS-232C) with selectable baud rates to accommodate a host and a printer among others;
  - full or half-duplex;
  - line or local mode select;
- composite video output (TTL level) to allow direct interfacing to a monochrome monitor;
- the heart of the system must be resident in software to allow ease of upgradeability, expandability and flexibility.

5.2 Operational Characteristics

Figure 5.1 illustrates the block diagram of the SAVE terminal. **Figure 5.2** shows the detailed circuit schematic which the operational description that follows will **make** reference to.

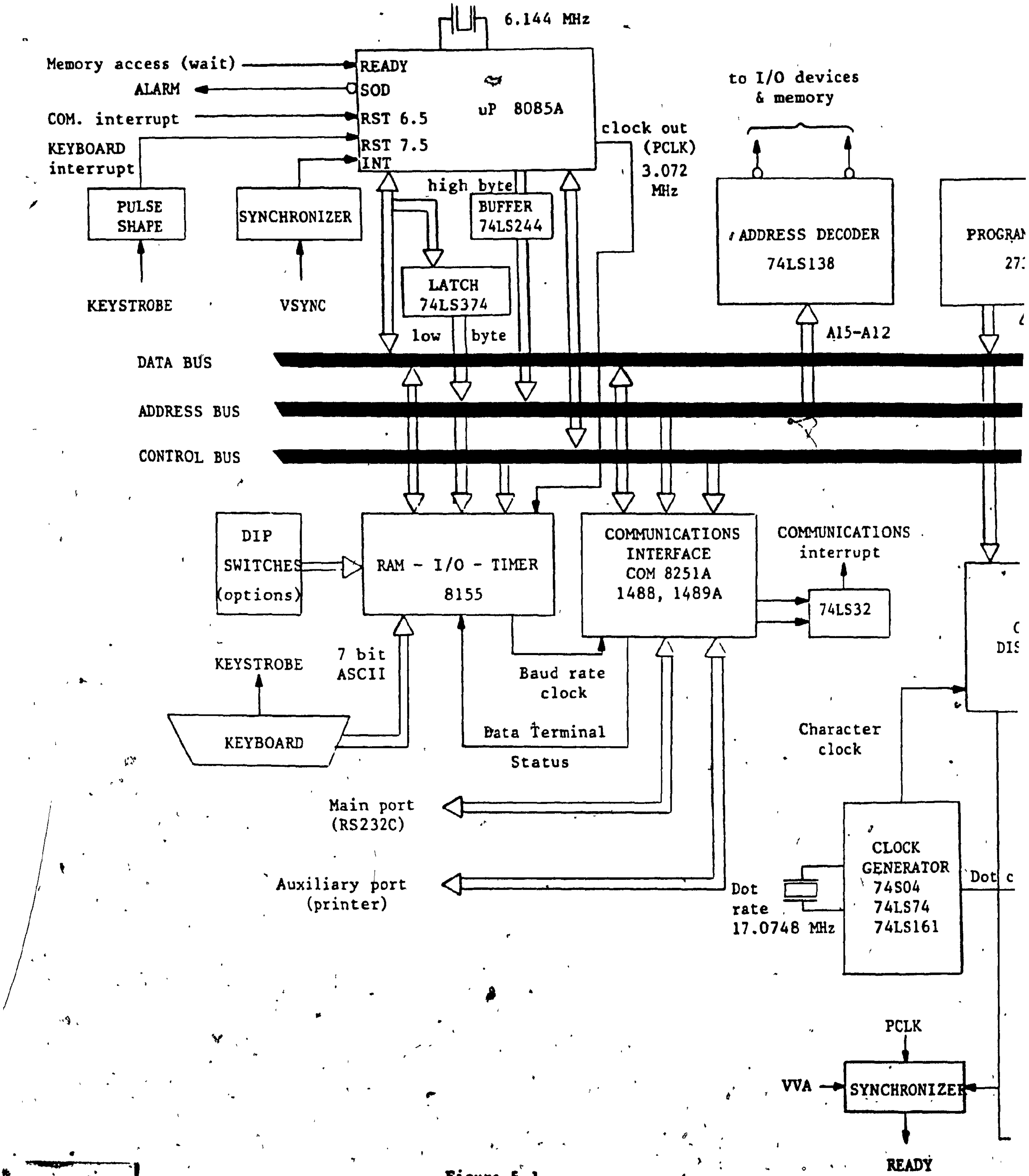
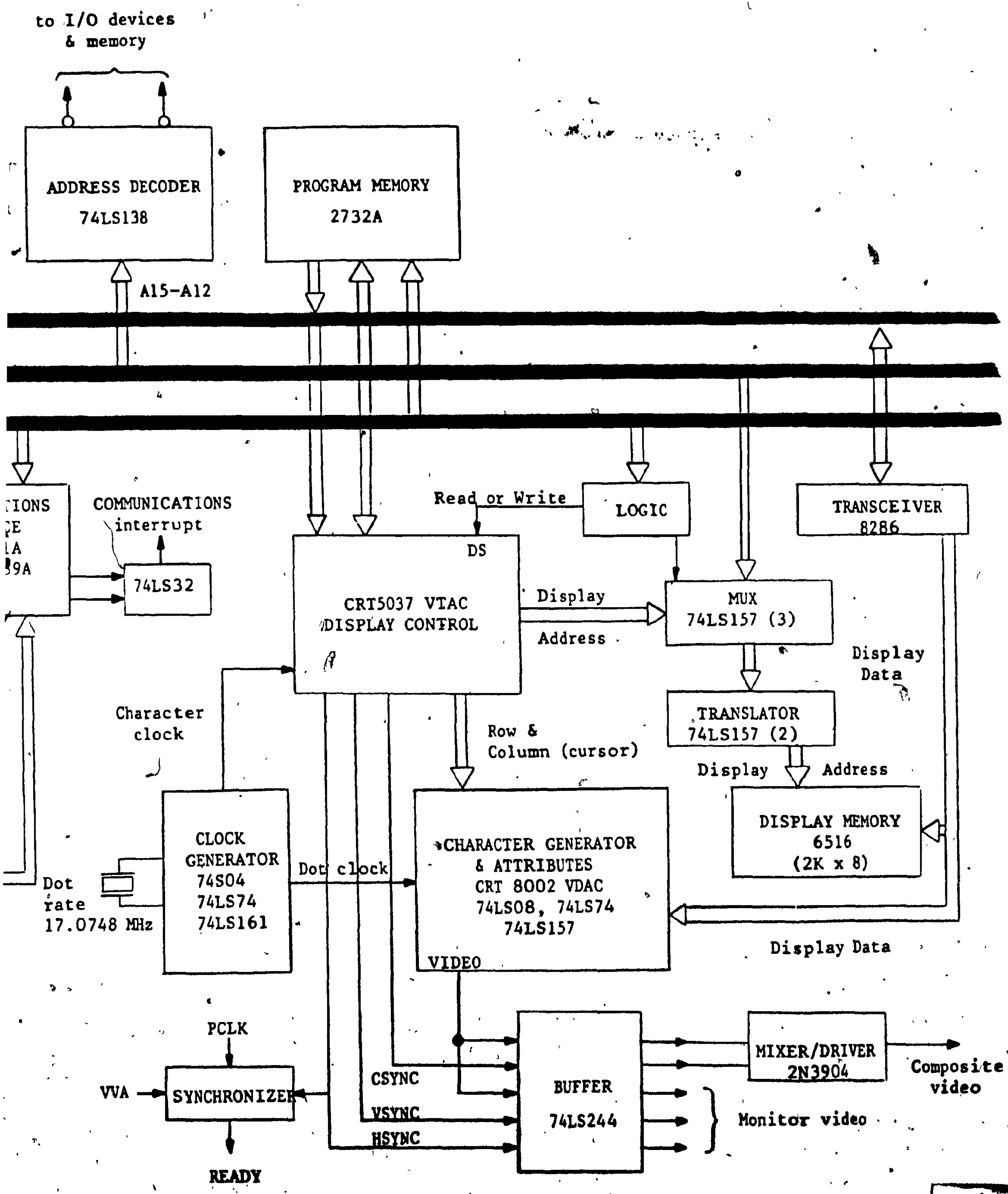


Figure 5.1
Terminal's Block Diagram

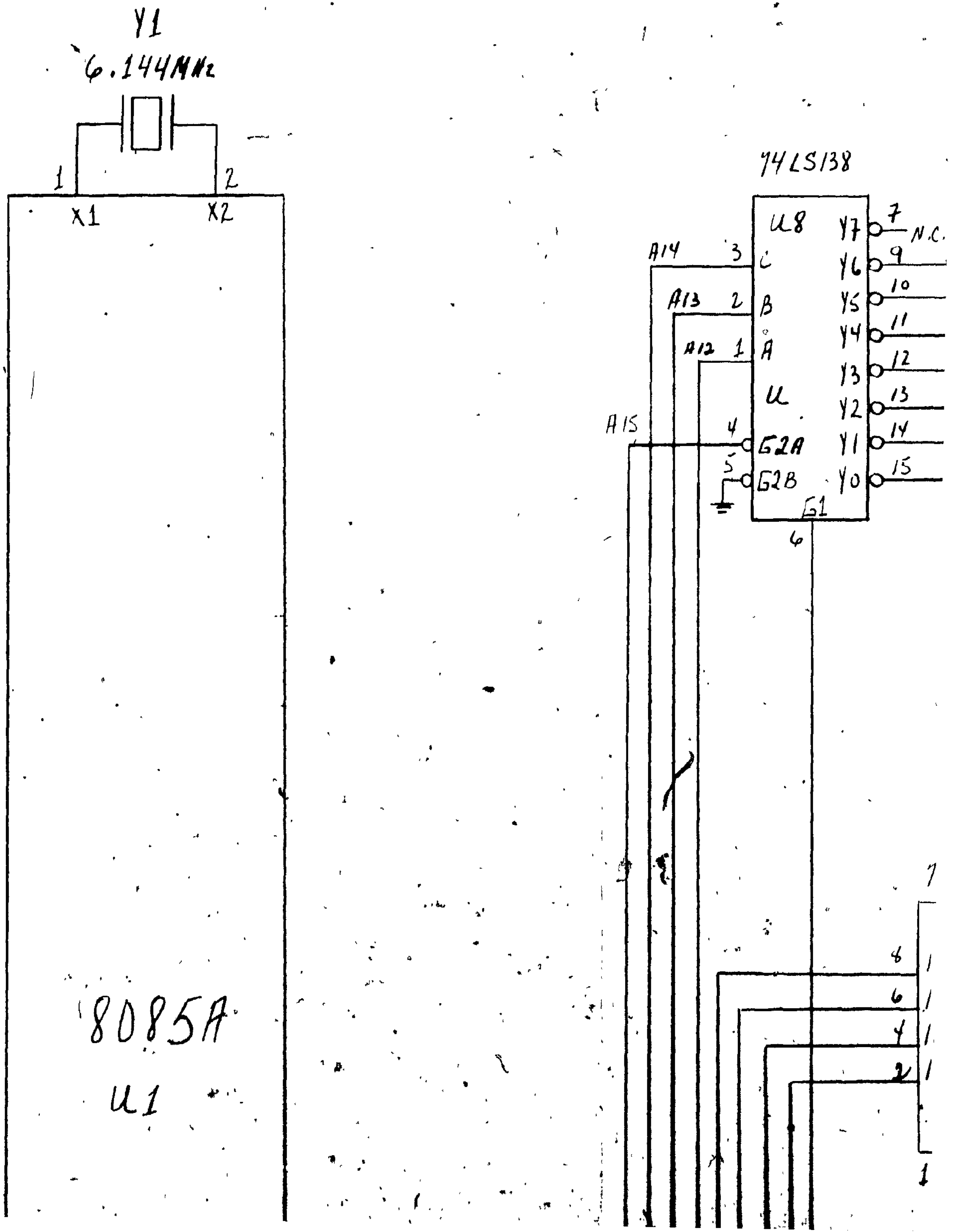

to I/O devices
& memory
ADDRESS DECODER
74LS138
PROGRAM MEMORY
2732A
A15-A12
:IONS
;E
LA
39A
COMMUNICATIONS
interrupt
74LS32
Read or Write
LOGIC
TRANSCEIVER
8286
DS
CRT5037 VTAC
DISPLAY CONTROL
Display
Address
MUX
74LS157 (3)
Display
Data
TRANSLATOR
74LS157 (2)
Character
clock
Row &
Column (cursor)
Display
Address
DISPLAY MEMORY
6516
(2K x 8)
CLOCK
GENERATOR
74S04
74LS74
74LS161
Dot
rate
17.0748 MHz
Dot clock
CHARACTER GENERATOR
& ATTRIBUTES
CRT 8002 VDAC
74LS08, 74LS74
74LS157
VIDEO
Display Data
PCLK
VVA
SYNCHRONIZER
READY
CSYNC
VSYNC
HSYNC
BUFFER
74LS244
MIXER/DRIVER
2N3904
Composite
video
Monitor video

2 of 2

FIGURE 5.2

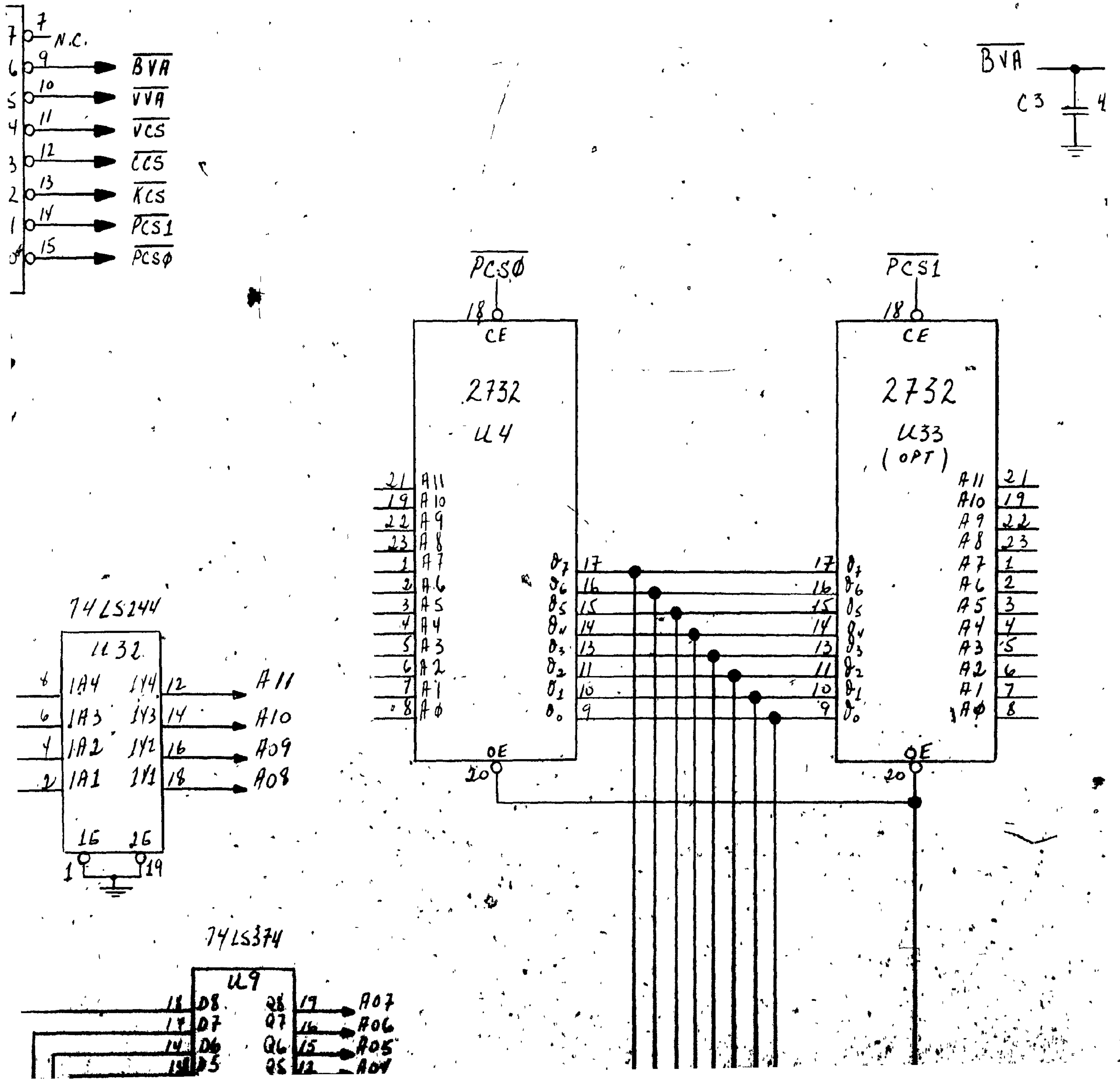
Terminal Circuit Diagram

14



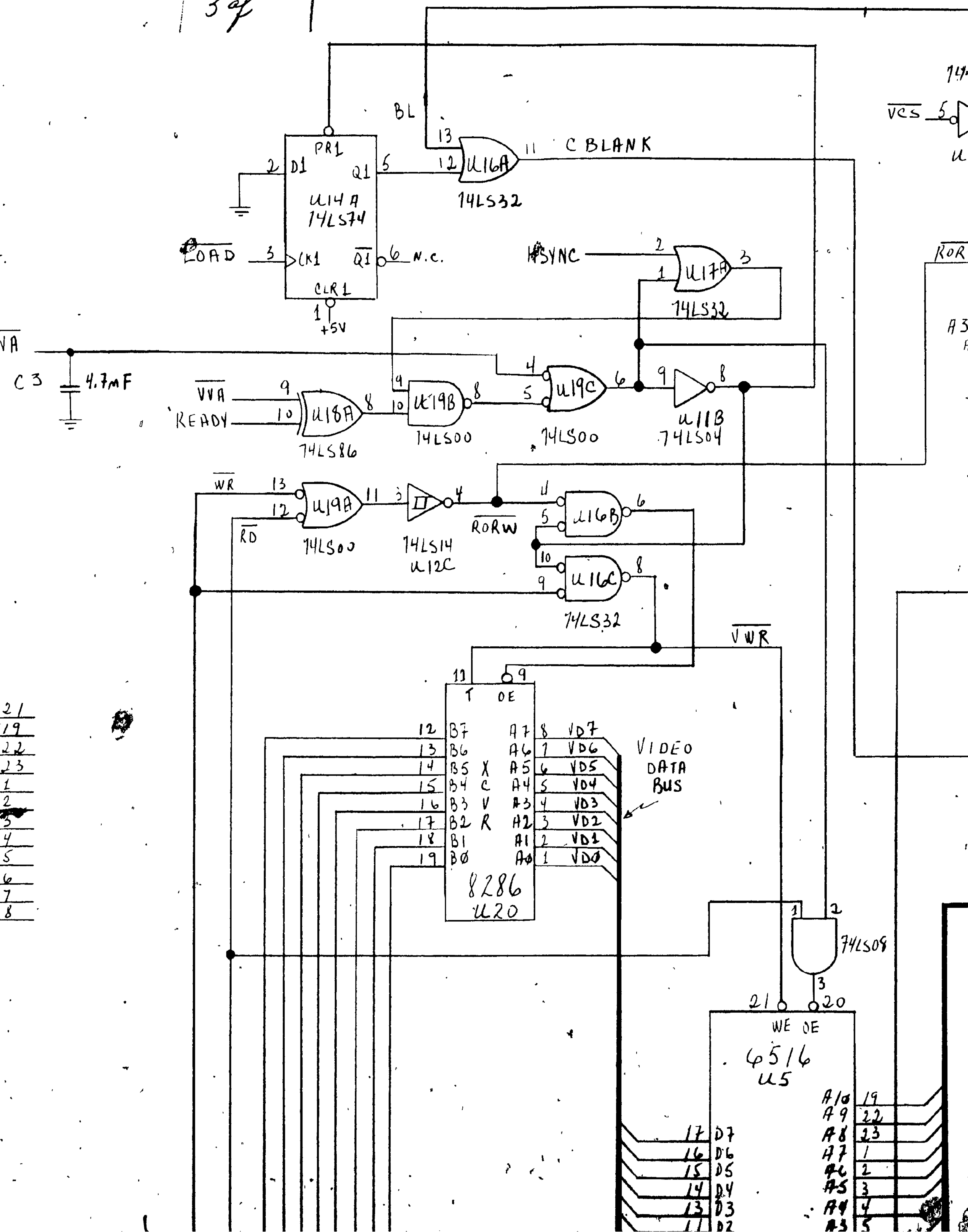
Y1
6.144MHz
1
2
X1
X2
8085A
U1
74LS138
U8
A14
3
C
A13
2
B
A12
1
A
U
A15
4
G2A
5
G2B
G1
6
Y7
7
N.C.
Y6
9
Y5
10
Y4
11
Y3
12
Y2
13
Y1
14
Y0
15
7
8
6
4
2
1

2 of

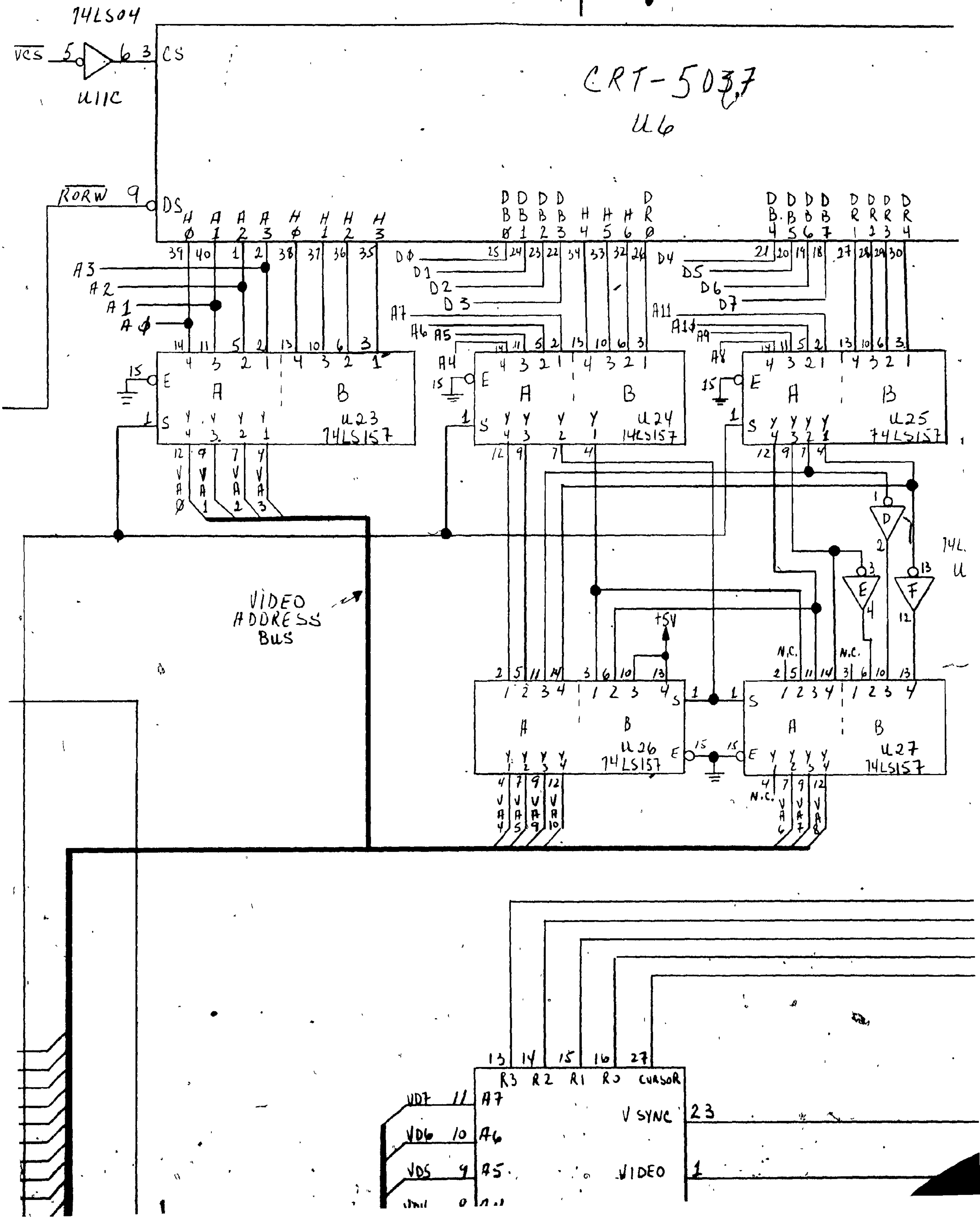


7 N.C.
6 9 BVA
5 10 VVA
4 11 VCS
3 12 CCS
2 13 KCS
1 14 PCS1
Ø 15 PCSØ
BVA
C3
4
PCSØ
18
CE
2732
U4
PCS1
18
CE
2732
U33
(OPT)
21 A11
19 A10
22 A9
23 A8
1 A7
2 A6
3 A5
4 A4
5 A3
6 A2
7 A1
8 AØ
D7 17
D6 16
D5 15
D4 14
D3 13
D2 11
D1 10
D0 9
OE
20
74LS244
U32
8 1A4
6 1A3
4 1A2
2 1A1
1Y4 12 A11
1Y3 14 A10
1Y2 16 A09
1Y1 18 A08
1G 2G
1 19
74LS374
U9
18 D8
17 D7
14 D6
13 D5
Q8 19 A07
Q7 16 A06
Q6 15 A05
Q5 12 A04

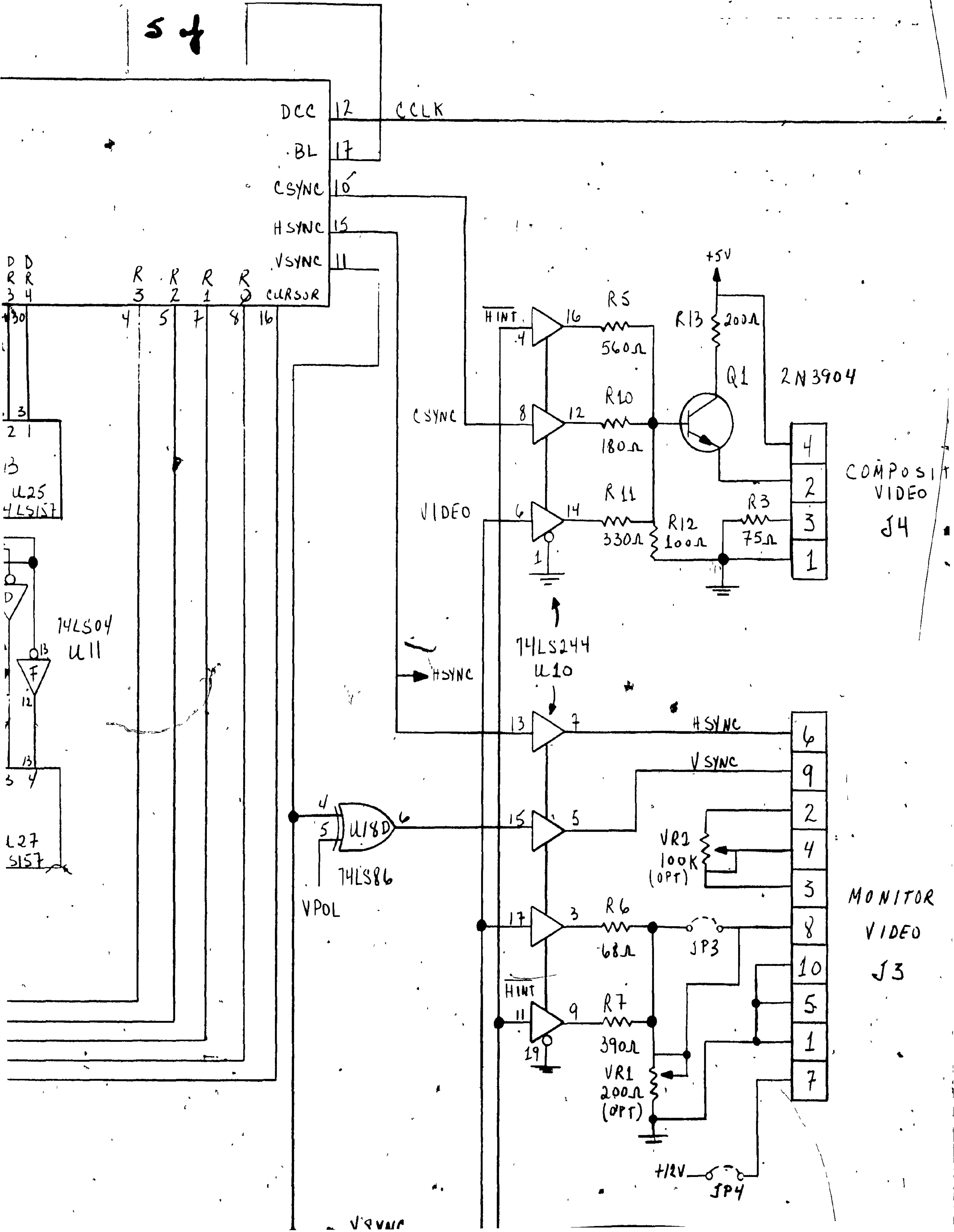
3 of


BL
PR1
2 D1
Q1 5
U14A
74LS74
LOAD
3 CK1
Q1 6 N.C.
CLR1
1
+5V
13
12 U16A
74LS32
11 CBLANK
HSYNC
2
1 U17A 3
74LS32
VA
C3
4.7mF
VVA
READY
9
10 U18A 8
74LS86
9
10 U19B 8
74LS00
4
5 U19C 6
74LS00
9
U11B 8
74LS04
WR
13
12 U19A 11
RD
74LS00
3 4
74LS14
U12C
RORW
4
5 U16B 6
10
9 U16C 8
74LS32
VWR
11 T
9 OE
12 B7
13 B6
14 B5
15 B4
16 B3
17 B2
18 B1
19 BØ
X
C
V
R
A7 8 VD7
A6 7 VD6
A5 6 VD5
A4 5 VD4
A3 4 VD3
A2 3 VD2
A1 2 VD1
AØ 1 VDØ
8286
U20
VIDEO
DATA
BUS
1
2
74LS08
3
21 WE
20 OE
6516
U5
17 D7
16 D6
15 D5
14 D4
13 D3
11 D2
A10 19
A9 22
A8 23
A7 1
A6 2
A5 3
A4 4
A3 5
21
19
22
23
1
2
3
4
5
6
7
8
74
VCS 5
U
ROR
A3

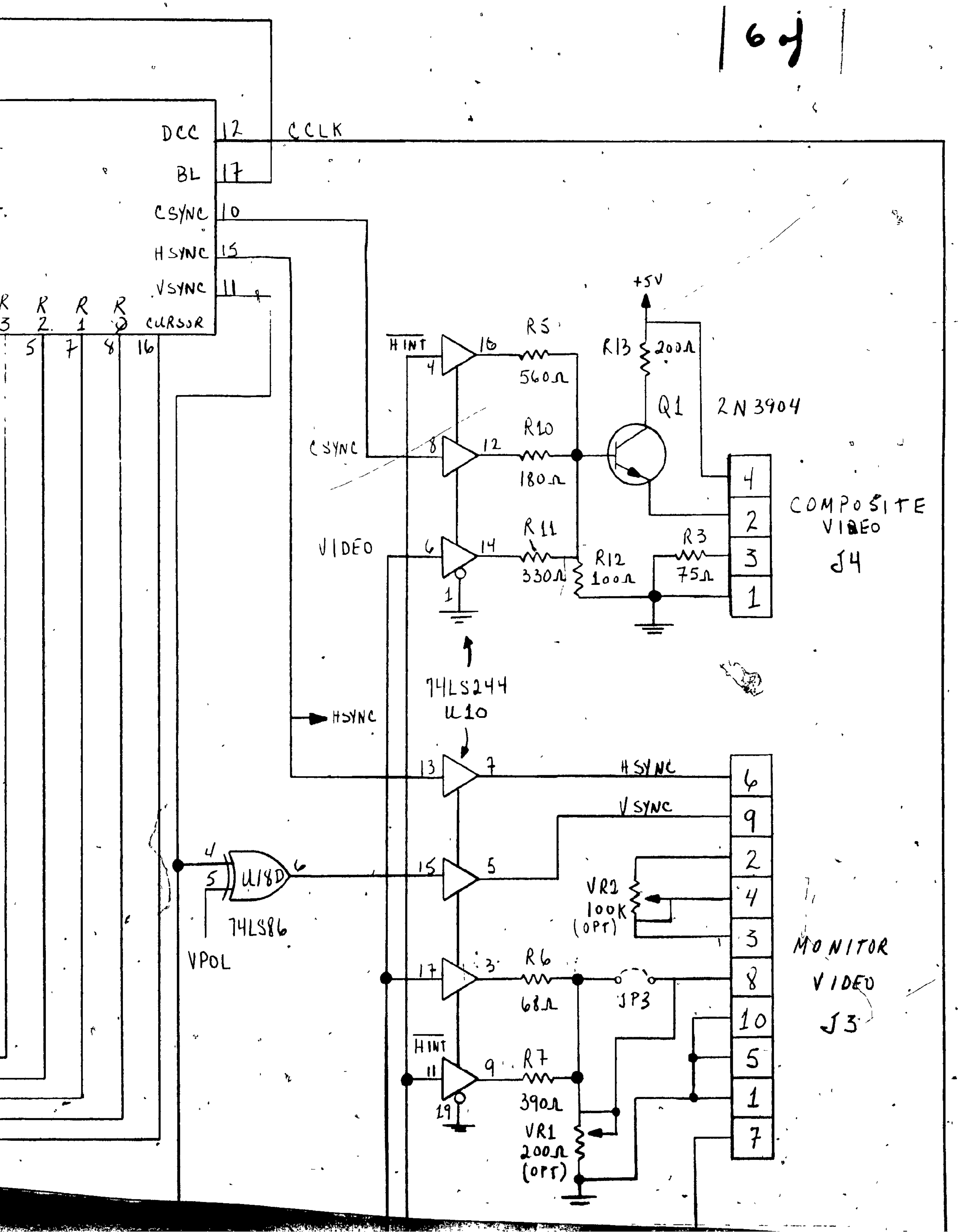
4 of



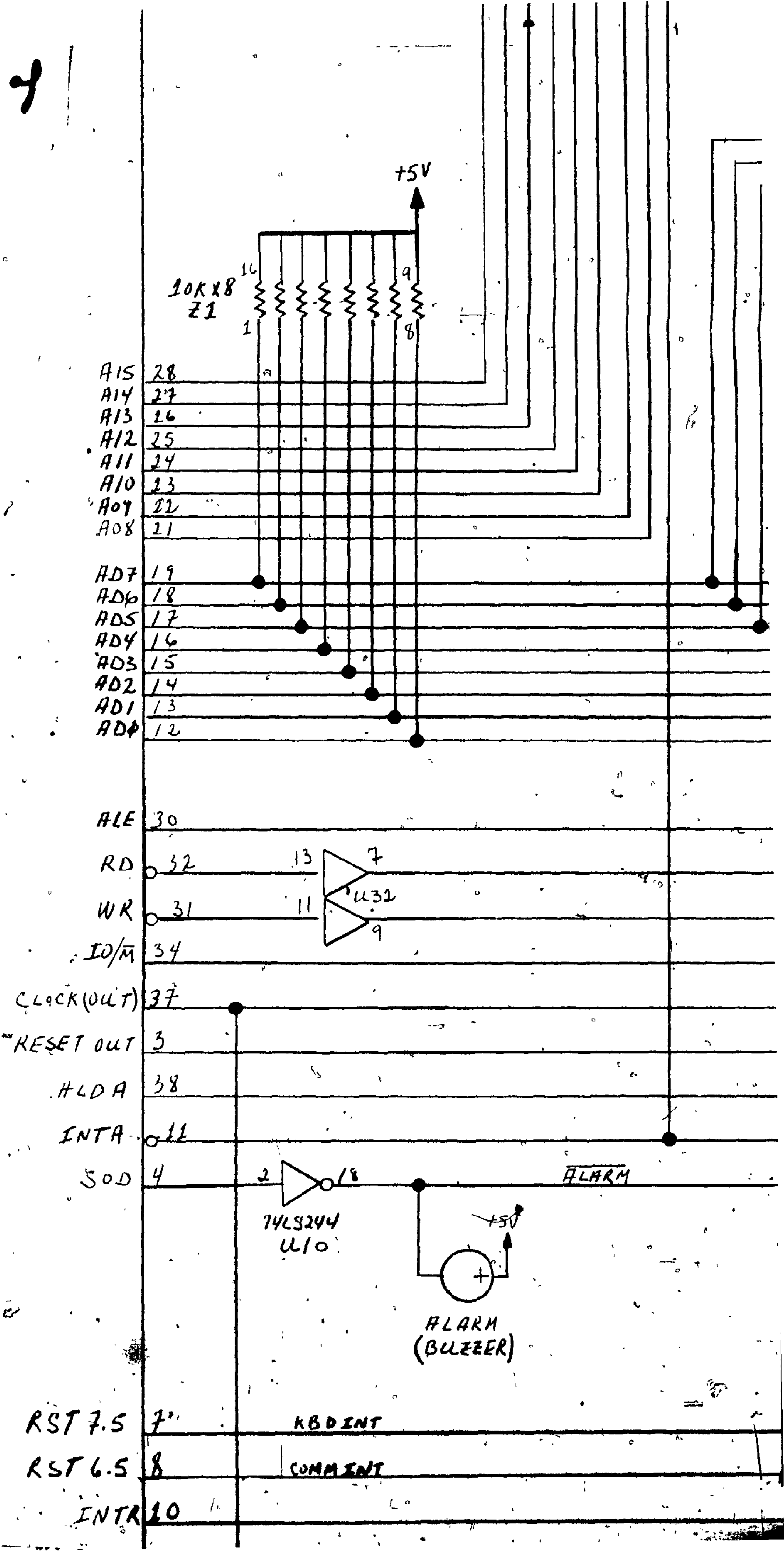
74LS04
VCS
U11C
CRT-5037
U6
RORW
CS
DS
H0 A1 A2 A3 H0 H1 H2 H3
DB0 DB1 DB2 DB3 H4 H5 H6 DR0
DB4 DB5 DB6 DB7 DR1 DR2 DR3 DR4
A3 A2 A1 A0
D0 D1 D2 D3
A7 A6 A5 A4
D4 D5 D6 D7
A11 A10 A9 A8
U23 74LS157
U24 74LS157
U25 74LS157
U26 74LS157
U27 74LS157
VA0 VA1 VA2 VA3
VA4 VA5 VA9 VA10
VA6 VA7 VA8
N.C.
+5V
74L
U
D
E
F
VIDEO ADDRESS BUS
R3 R2 R1 R0 CURSOR
VD7 A7
VD6 A6
VD5 A5
V SYNC
VIDEO

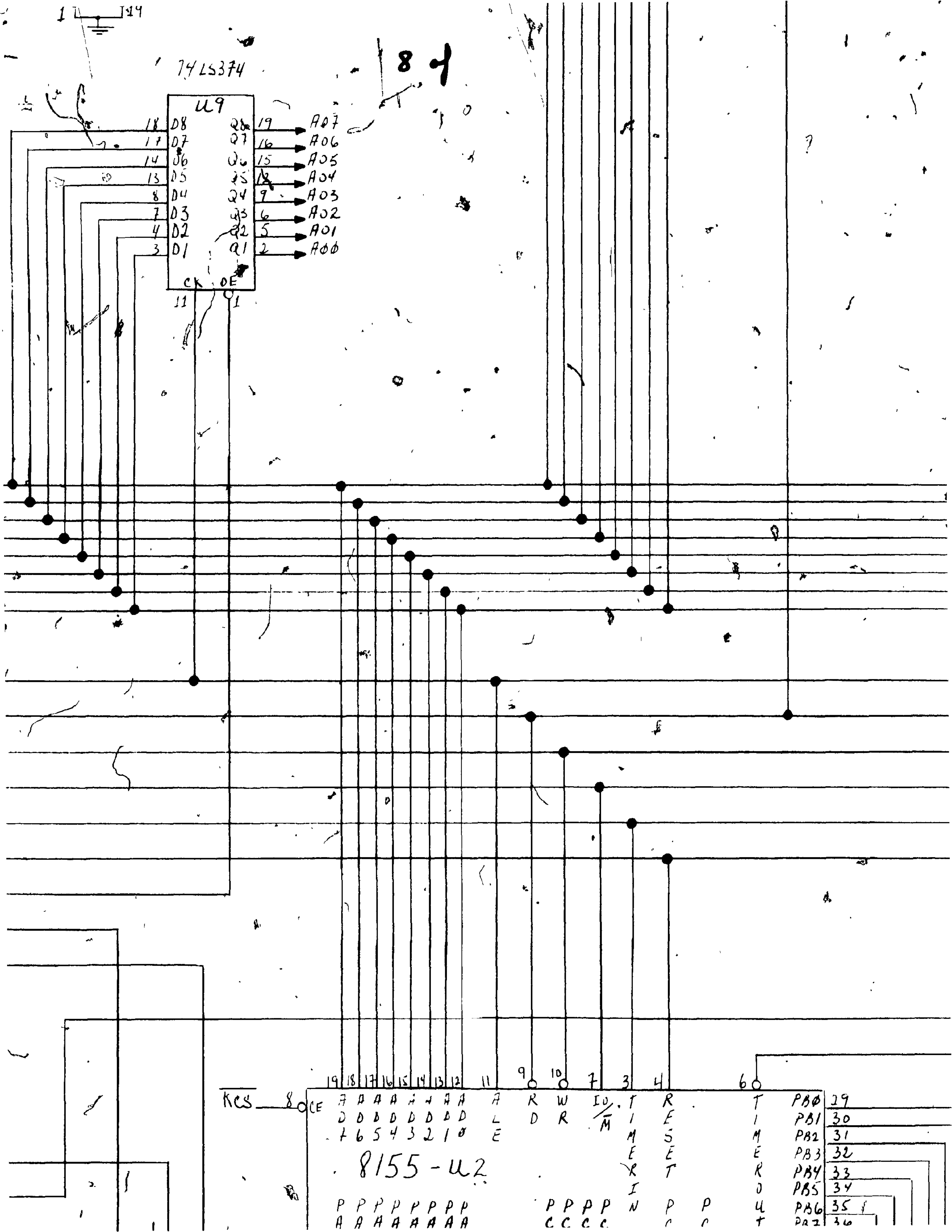

5 4
DCC
12
CCLK
BL
17
CSYNC
10
HSYNC
15
VSYNC
11
CURSOR
D R 3
D R 4
R 3
R 2
R 1
R 0
30
4
5
7
8
16
3
2 1
13
U25
74LS157
74LS04
U11
F
D
13
12
13
3 4
L27
S157
HINT
4
16
R5
560Ω
+5V
R13
200Ω
Q1
2N3904
CSYNC
8
12
R10
180Ω
VIDEO
6
14
R11
330Ω
R12
100Ω
R3
75Ω
1
4
2
3
1
COMPOSIT VIDEO
J4
74LS244
U10
HSYNC
13
7
HSYNC
6
V SYNC
9
4
5
U18D
6
74LS86
VPOL
15
5
2
VR2
100K
(OPT)
4
3
MONITOR
VIDEO
J3
17
3
R6
68Ω
JP3
8
10
5
HINT
11
9
R7
390Ω
1
19
VR1
200Ω
(OPT)
7
+12V
JP4
VSYNC

6 of

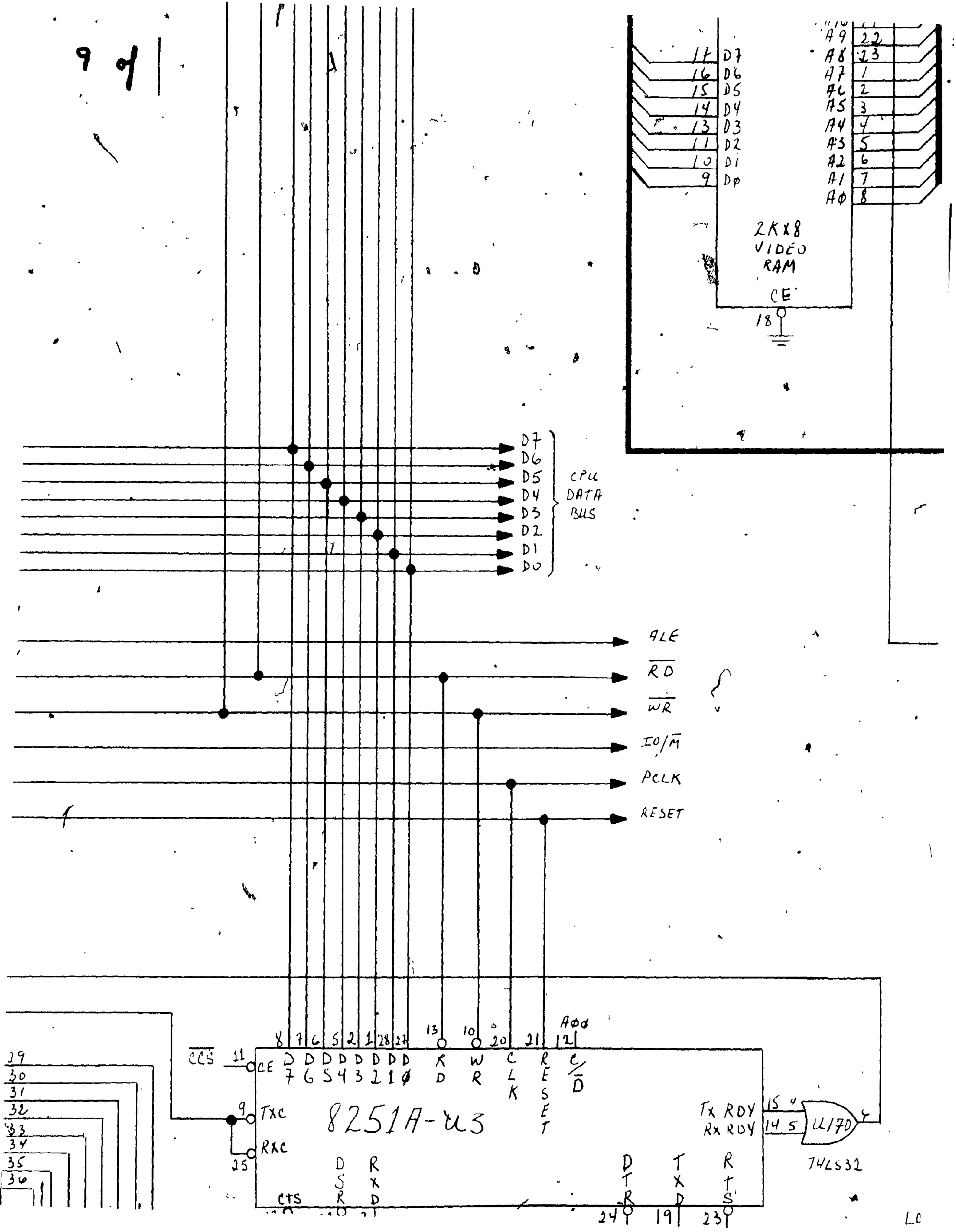


DCC 12
BL 17
CSYNC 10
HSYNC 15
VSYNC 11
CURSOR
R 3
R 2
R 1
R Ø
5
7
8
16
CCLK
+5V
R13 200Ω
Q1 2N3904
HINT
4
16
R5 560Ω
CSYNC
8
12
R10 180Ω
VIDEO
6
14
R11 330Ω
R12 100Ω
R3 75Ω
1
4
2
3
1
COMPOSITE VIDEO J4
74LS244 U10
HSYNC
13
7
HSYNC
VSYNC
6
9
2
4
3
8
10
5
1
7
4
5
U18D
6
74LS86
VPOL
15
5
VR2 100K (OPT)
17
3
R6 68Ω
JP3
HINT
11
9
R7 390Ω
19
VR1 200Ω (OPT)
MONITOR VIDEO J3


+5V
10KX8
Z1
16
1
9
8
A15 28
A14 27
A13 26
A12 25
A11 24
A10 23
A09 22
A08 21
AD7 19
AD6 18
AD5 17
AD4 16
AD3 15
AD2 14
AD1 13
ADϕ 12
ALE 30
RD 32
WR 31
13
7
11
9
U32
IO/M̄ 34
CLOCK(OUT) 37
RESET OUT 3
HLDA 38
INTA 11
SOD 4
2
18
74LS244
U10
ALARM
+5V
ALARM
(BUZZER)
RST 7.5 7
RST 6.5 8
INTR 10
KBDINT
COMMINT

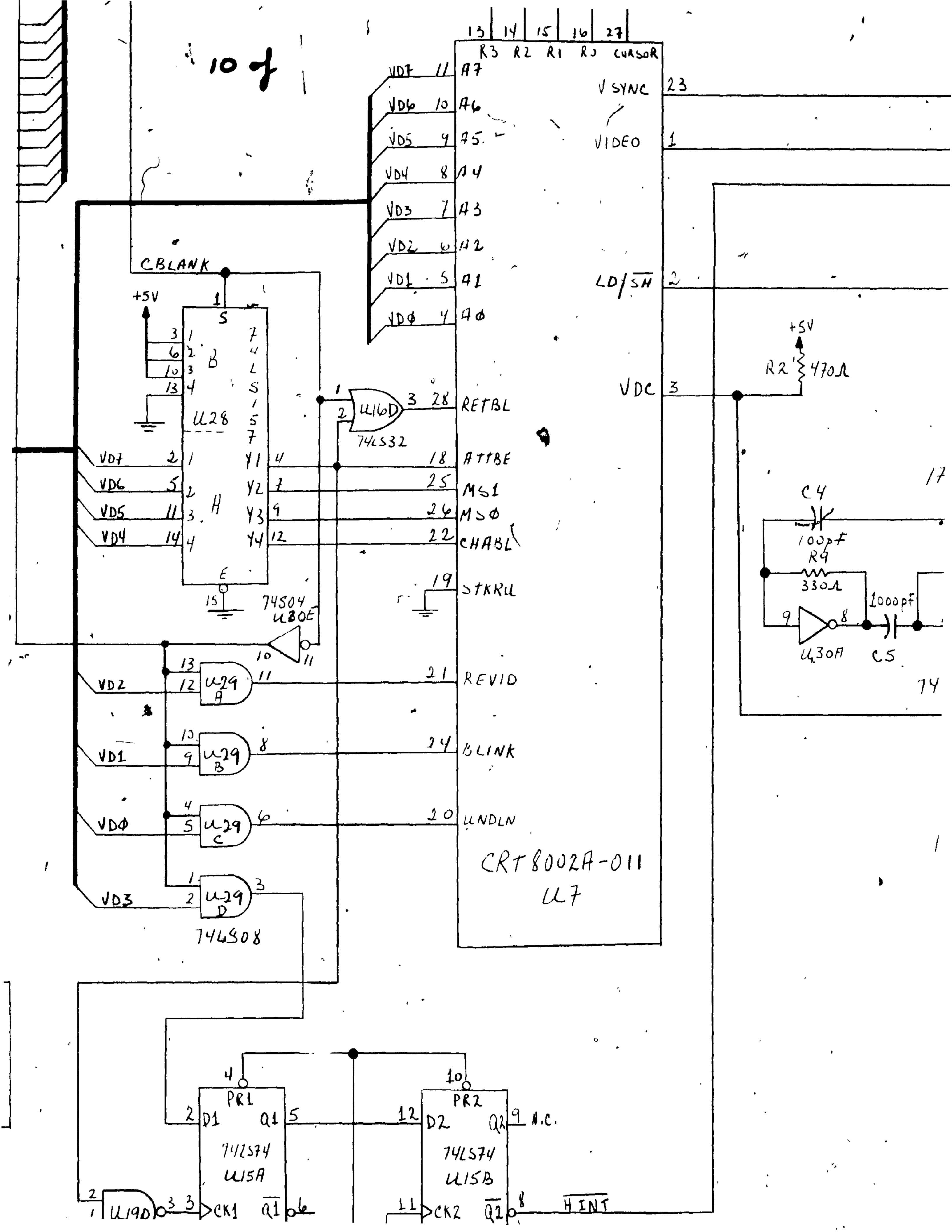

1
14
84
74LS374
U9
18 D8
17 D7
14 D6
13 D5
8 D4
7 D3
4 D2
3 D1
Q8 19 A07
Q7 16 A06
Q6 15 A05
Q5 12 A04
Q4 9 A03
Q3 6 A02
Q2 5 A01
Q1 2 A00
CK 11
OE 1
KCS
8 CE
19 AD7
18 AD6
17 AD5
16 AD4
15 AD3
14 AD2
13 AD1
12 AD0
11 ALE
9 RD
10 WR
7 IO/M
3 TIMER IN
4 RESET
6 TIMER OUT
PB0 29
PB1 30
PB2 31
PB3 32
PB4 33
PB5 34
PB6 35
PB7 36
8155 - U2
P P P P P P P P
A A A A A A A A
P P P P
C C C C
P
C
P
C

9 of

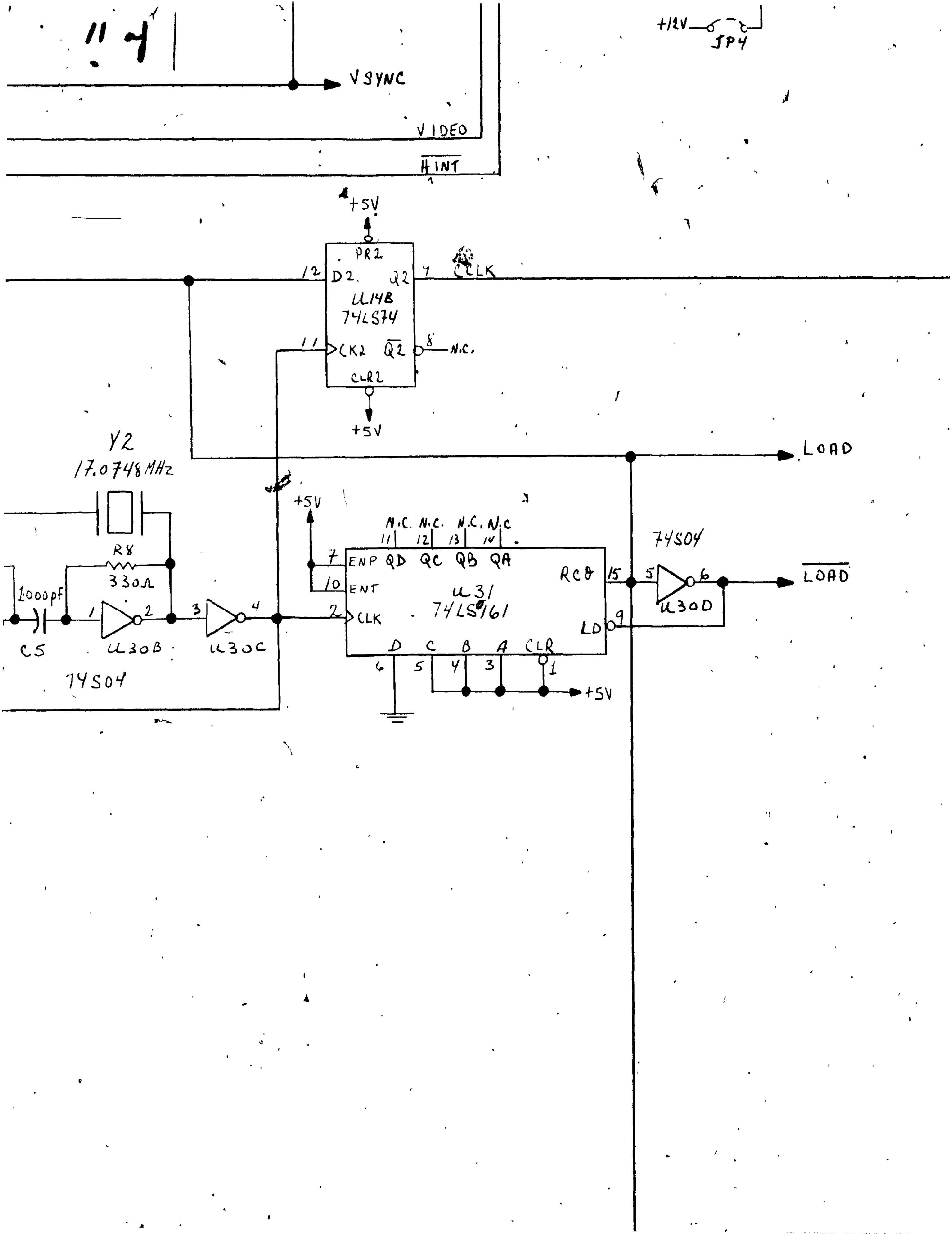


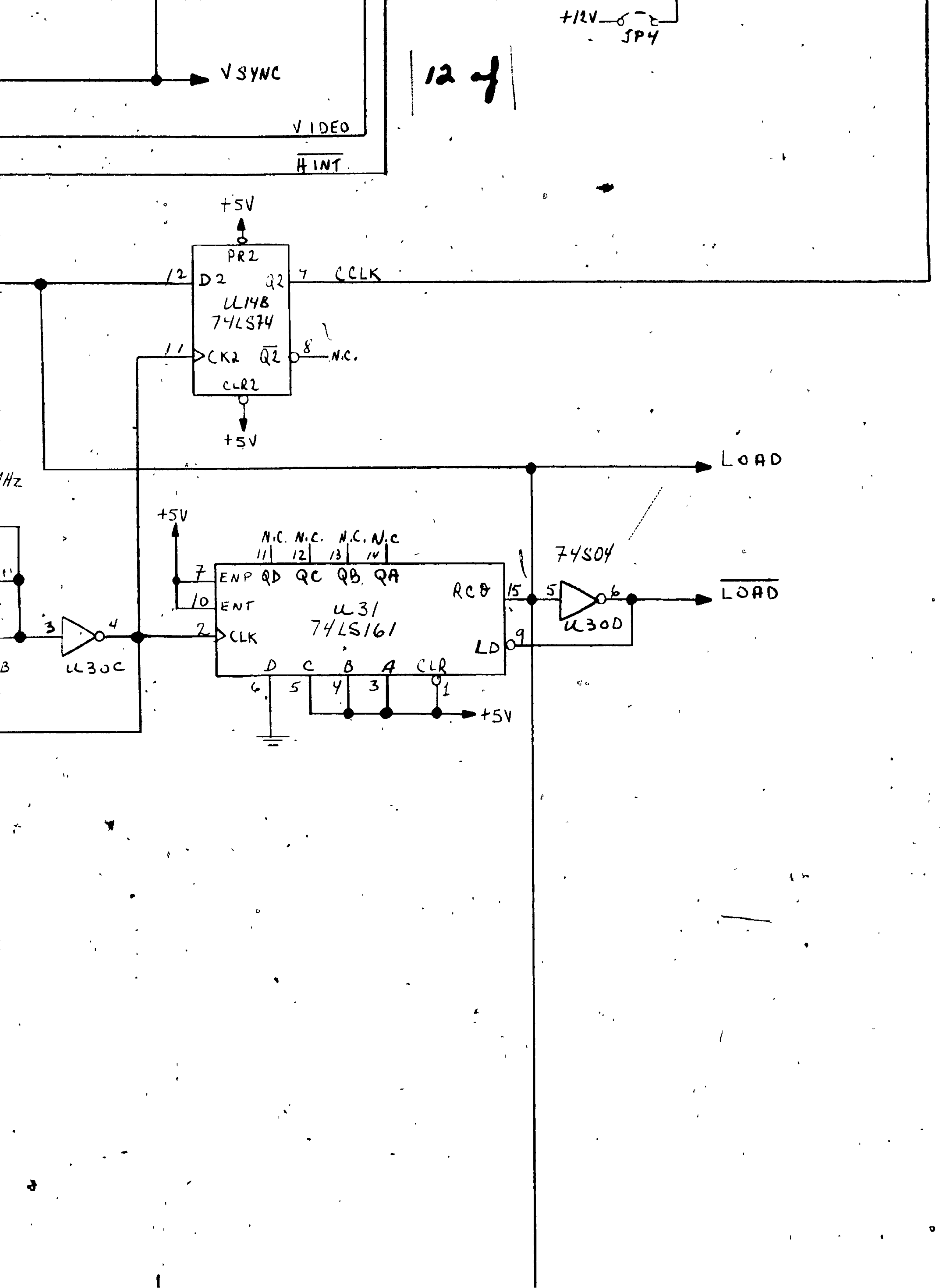
A9 22
A8 23
A7 1
A6 2
A5 3
A4 4
A3 5
A2 6
A1 7
AΦ 8
17 D7
16 D6
15 D5
14 D4
13 D3
11 D2
10 D1
9 DΦ
2KX8
VIDEO
RAM
CE
18
D7
D6
D5
D4
D3
D2
D1
D0
CPU
DATA
BUS
ALE
RD
WR
IO/M
PCLK
RESET
AΦΦ
8 7 6 5 2 1 28 27
13 10 20 21 12
CCS
11
CE
D7 D6 D5 D4 D3 D2 D1 DΦ
RD
WR
CLK
RESET
C/D
9
TxC
RxC
25
8251A-U3
TxRDY
RxRDY
15 4
14 5
U17D
6
74LS32
DSR
RXD
CTS
DTR
TXD
RTS
24
19
23
29
30
31
32
33
34
35
36
LC

10 of

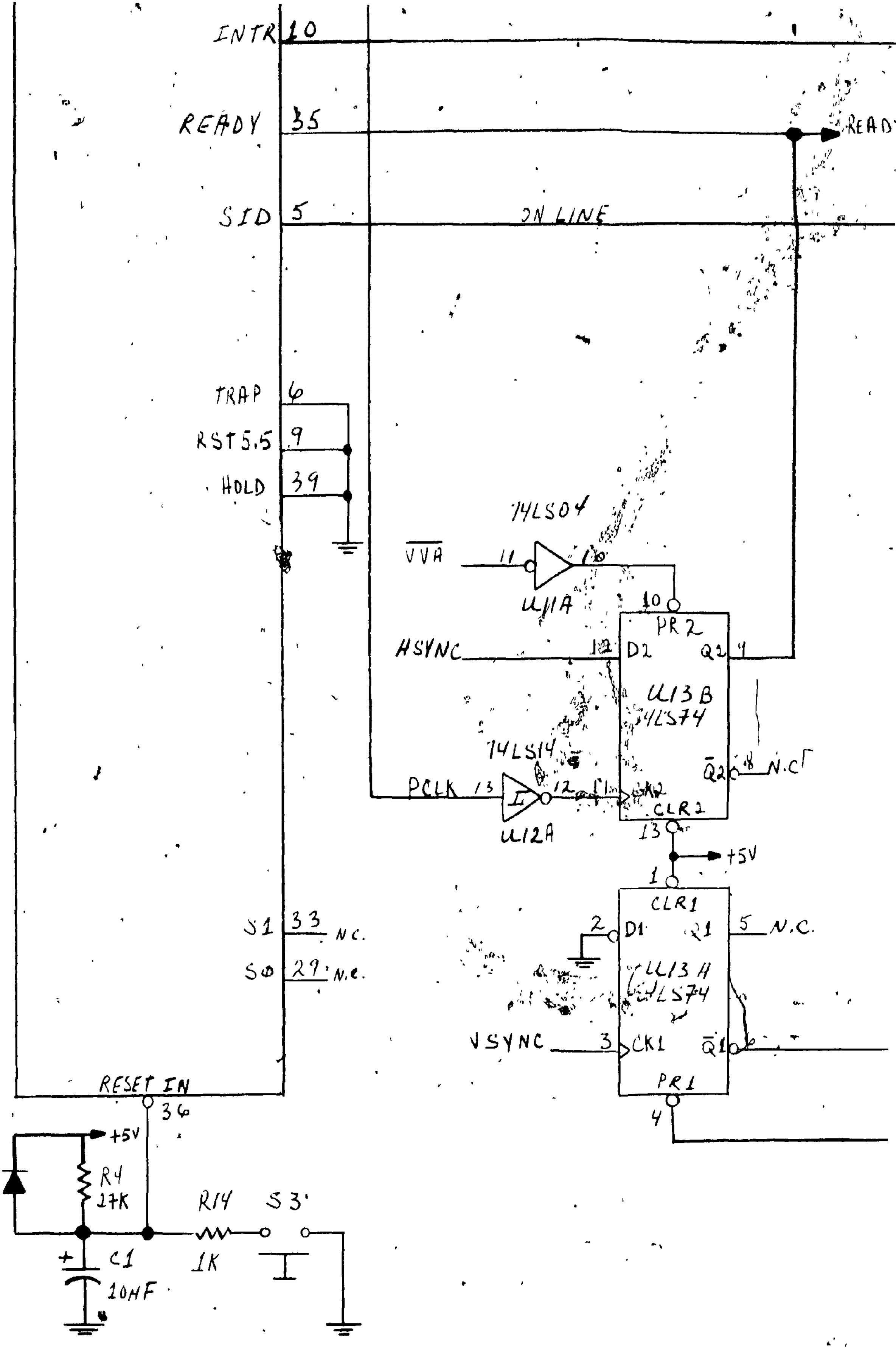


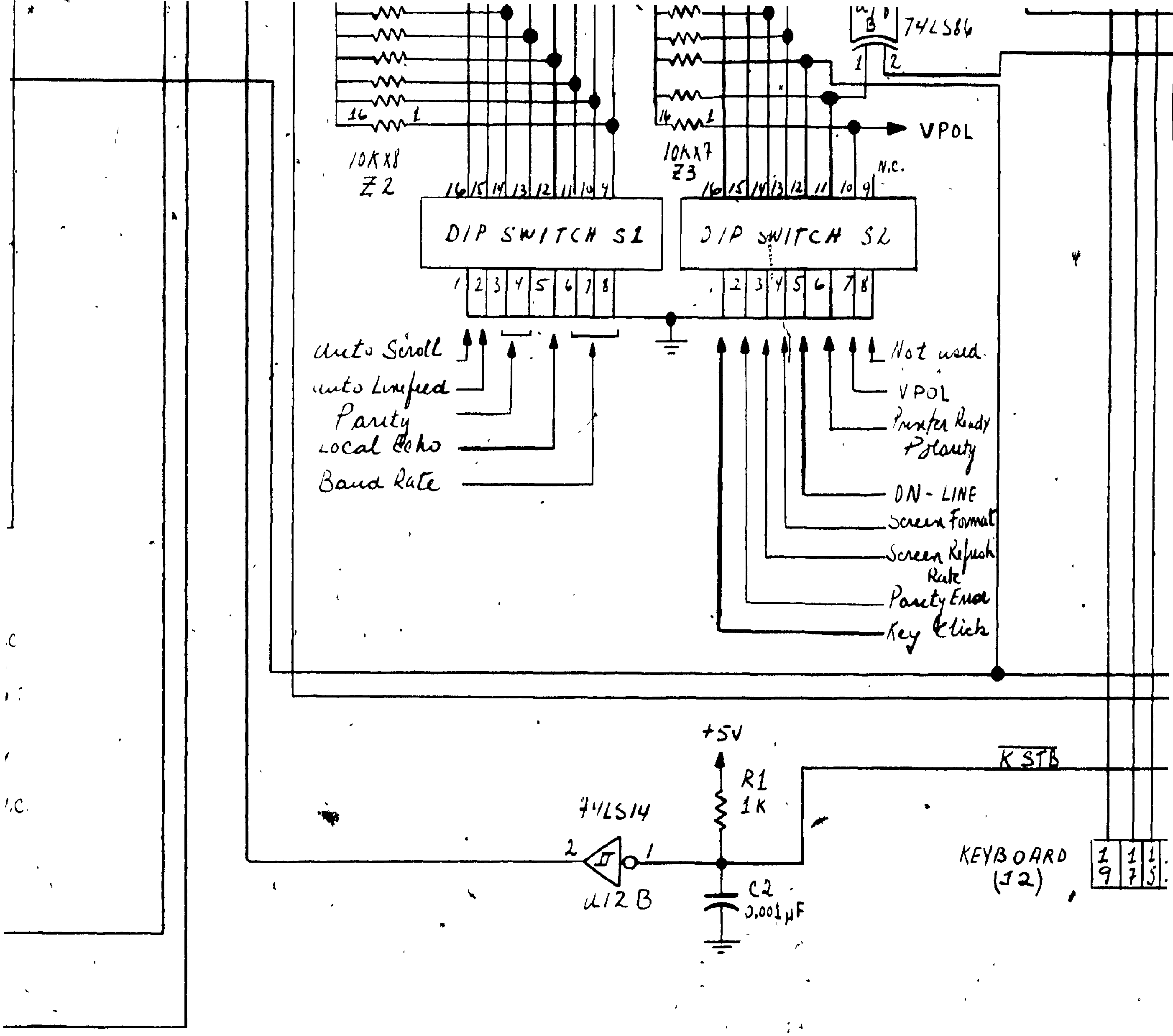
13
14
15
16
27
R3
R2
R1
R0
CURSOR
VD7 11 A7
VD6 10 A6
VD5 9 A5
VD4 8 A4
VD3 7 A3
VD2 6 A2
VD1 5 A1
VDØ 4 AØ
V SYNC 23
VIDEO 1
LD/SH 2
VDC 3
+5V
R2 470Ω
CBLANK
+5V
S
B
U28
A
E
15
1
74LS32
U16D
28 RETBL
18 ATTBE
25 MS1
26 MSØ
22 CHABL
19 STKRU
74S04
U30E
21 REVID
24 BLINK
20 UNDLN
CRT 8002A-011
U7
U29 A
U29 B
U29 C
U29 D
74LS08
VD7
VD6
VD5
VD4
VD2
VD1
VDØ
VD3
C4
100pF
R9
330Ω
1000pF
U30A
C5
17
14
PR1
D1
Q1
74LS74
U15A
CK1
Q1
PR2
D2
Q2
N.C.
74LS74
U15B
CK2
Q2
HINT
U19D


+12V
JP4
VSYNC
VIDEO
$\overline{\text{HINT}}$
+5V
PR2
12
D2
Q2
9
CCLK
U14B
74LS74
11
CK2
$\overline{Q2}$
8
N.C.
CLR2
+5V
LOAD
Y2
17.0748MHz
R8
330Ω
1000pF
C5
1
2
U30B
3
4
U30C
74S04
+5V
N.C. N.C. N.C. N.C
11 12 13 14
7
ENP
QD QC QB QA
10
ENT
2
CLK
U31
74LS161
RCO
15
LD
9
D C B A CLR
6 5 4 3 1
+5V
74S04
5
6
U30D
$\overline{\text{LOAD}}$

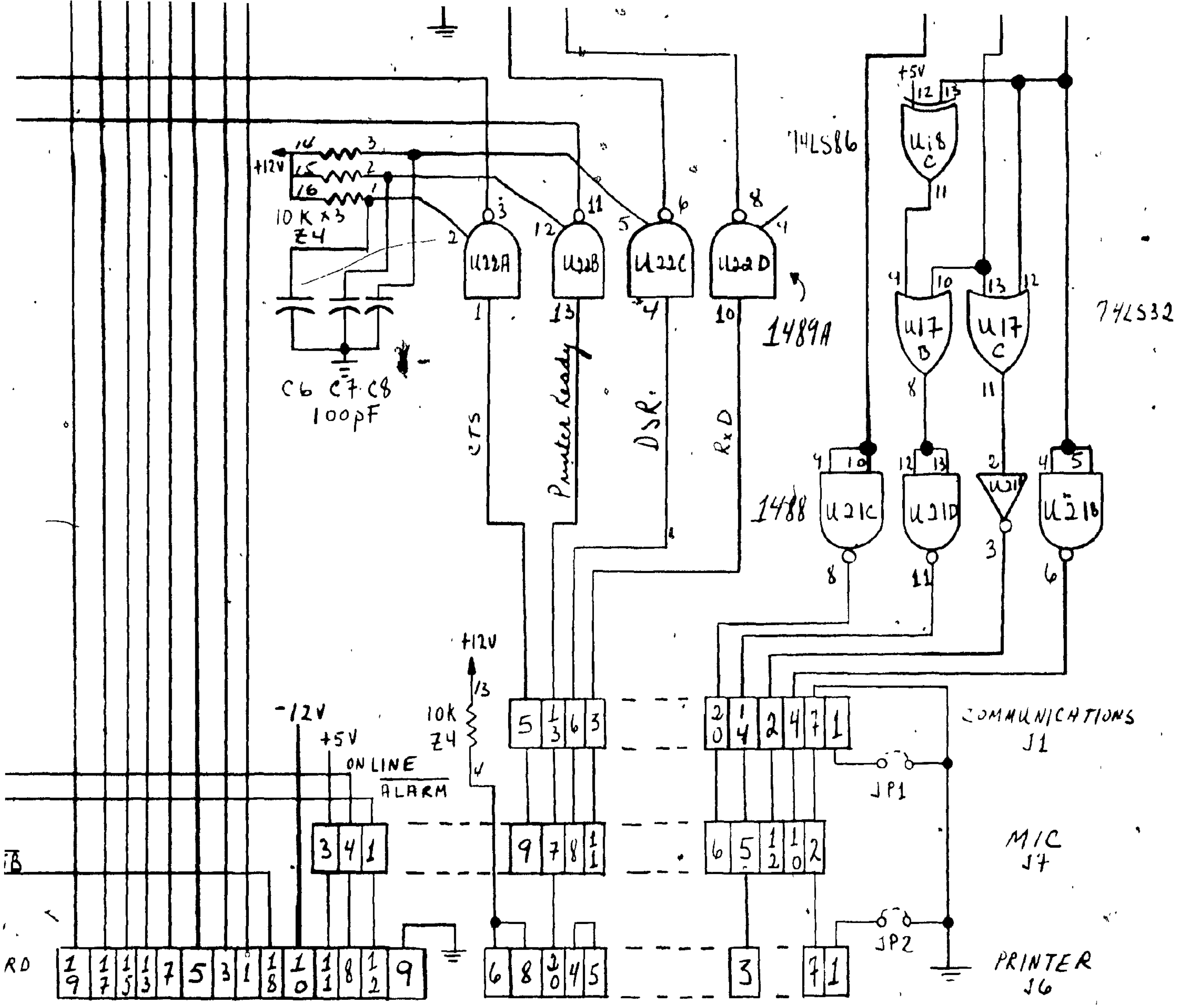

+12V
JP4
12 of
VSYNC
VIDEO
H INT
+5V
PR2
12
D2
Q2
7
CCLK
U14B
74LS74
11
CK2
Q2
8
N.C.
CLR2
+5V
LOAD
Hz
+5V
N.C. N.C. N.C. N.C
11 12 13 14
7
ENP
QD
QC
QB
QA
10
ENT
2
CLK
U31
74LS161
RCO
15
LD
9
D C B A CLR
6 5 4 3 1
+5V
3
4
U30C
B
74S04
5
6
U30D
LOAD

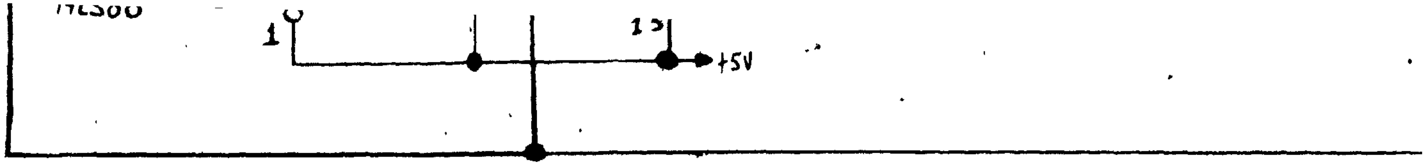
13 of

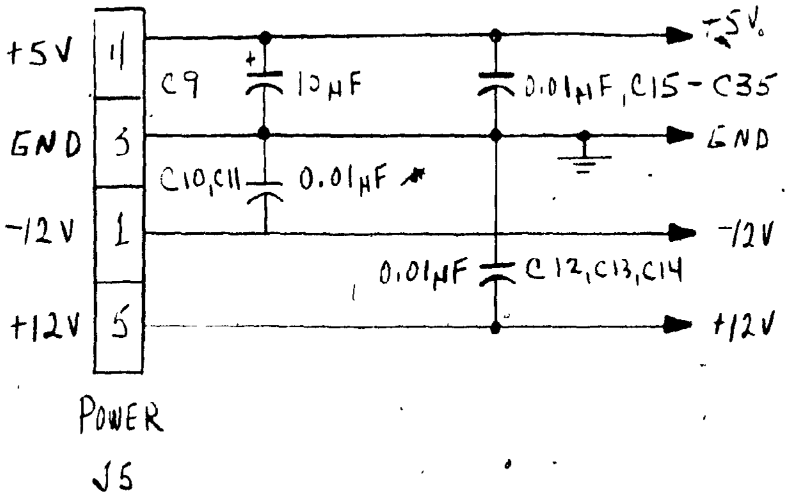

INTR 10
READY 35
READY
SID 5
ON LINE
TRAP 6
RST5.5 9
HOLD 39
74LS04
VVA
11
10
U11A
10
PR2
HSYNC
12
D2
Q2
9
U13B
74LS74
74LS14
Q2
8
N.C.
PCLK 13
12
11
CK2
U12A
CLR2
13
+5V
1
CLR1
2
D1
Q1
5
N.C.
U13A
74LS74
VSYNC
3
CK1
Q1
PR1
4
S1 33 N.C.
S0 29 N.C.
RESET IN
36
+5V
D1
1N4148
R4
27K
R14
1K
S3
C1
10MF


74LS86
1
2
16
1
VPOL
10KX8
Z2
14
1
10KX7
Z3
N.C.
16 15 14 13 12 11 10 9
16 15 14 13 12 11 10 9
DIP SWITCH S1
DIP SWITCH S2
1 2 3 4 5 6 7 8
2 3 4 5 6 7 8
auto Scroll
auto Linefeed
Parity
Local Echo
Baud Rate
Not used
VPOL
Printer Ready Polarity
ON-LINE
Screen Format
Screen Refresh Rate
Parity Error
Key Click
+5V
R1
1K
KSTB
74LS14
2
1
U12B
C2
0.001μF
KEYBOARD
(J2)
19
17
15
.C
N.C.

14 oF


74LS86
+5V
U18 C
74LS32
U17 B
U17 C
+12V
10K x3
Z4
U22A
U22B
U22C
U22D
1489A
C6 C7 C8
100pF
CTS
Printer Ready
DSR
RxD
1488
U21C
U21D
U21
U21B
+12V
10K
Z4
-12V
+5V
ONLINE
ALARM
COMMUNICATIONS
J1
JP1
MIC
J7
JP2
PRINTER
J6
RD
B





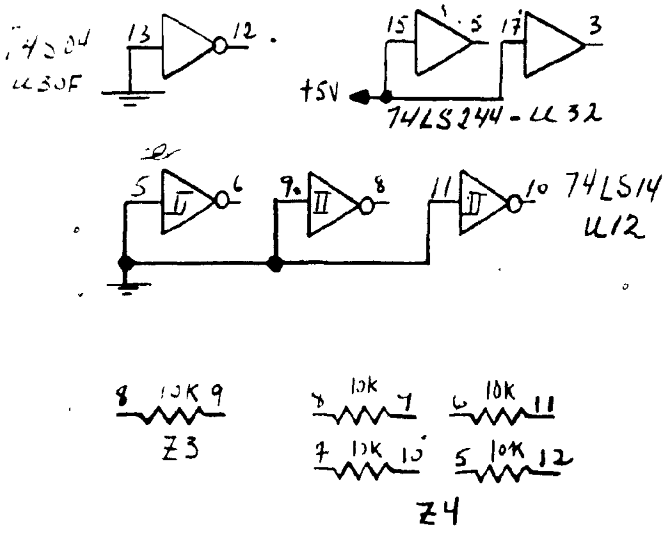
16 of

TITLE: SAVE TERM

DESIGN: JUAN (

17 of

SPARES



POWER DISTRIBUTI				
CHIPS	GND	+5V	+12V	-12
8085A 8155	20	40		
CRT 5037	6	14	13	
CRT 8002	17	12		
COM 8251A	4	26		
EPROM 1 (EPROM 2) } 2732	12	24		
74LS244 74LS374 8286	10	20		
RAM → 6516	12	24		
74LS138 74LS157 74LS161	8	16		
1488	7		14	1
ALL OTHER	7	14		

TERMINAL

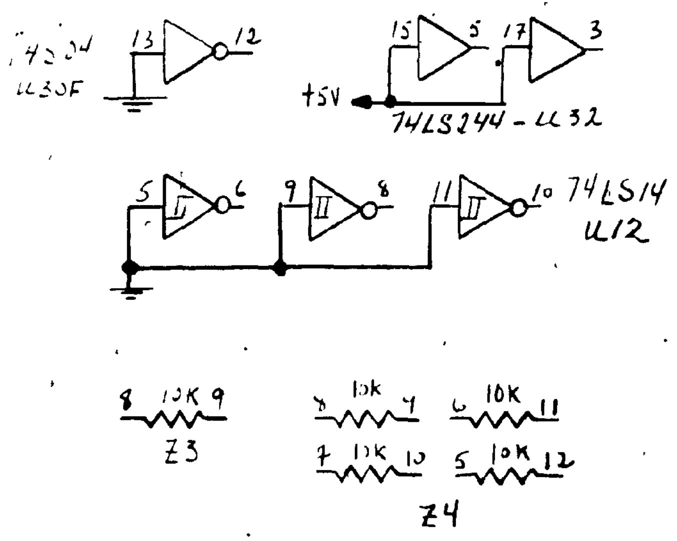
N (SEPT 83)

18 of 18

POWER DISTRIBUTION

CHIPS	GND	+5V	+12V	-12V	# PINS
8085A 8155	20	40			40
CRT 5037	6	14	13		40
CRT 8002	17	12			28
COM 8251A	4	26			28
EPROM1 (EPROM2) } 2732	12	24			24
74LS244 74LS374 8286	10	20			20
RAM→ 6516	12	24			24
74LS138 74LS157 74LS161	8	16			16
1488	7		14	1	14
ALL OTHER	7	14			14

SPARES



5.2.1 The Video Controller and Character Generator Section

The **heart** of the terminal is the CRT 5037 Video Timer and Controller (VTAC) chip (U6, **Figure 5.2**) and the CRT 8002 Video Display Attribute Controller (VDAC) (U7). As stated out in Section 4.3.1, the CRT 5037 is a flexible, programmable CRT controller chip that generates all the timing signals, e.g. syncs and blanking, necessary to run the CRT video monitor. The CRT 8002[[illegible],43], a companion chip to the CRT 5037, is a highly versatile video generator providing character generation, character attributes, graphic modes and cursor generation, while incorporating an on chip video shift register that can be clocked up to 20 MHz. The degree of circuit integration aids in keeping chip count to a minimum. **Figure 5.3** shows the CRT 8002 block diagram and timing specifications necessary to interface it. The character generator ROM built into the CRT 8002, **Figure 5.4**, provides for 128 characters, each in a 7 x 11 character matrix, for high resolution character formation. Video RAM (U5) is configured as a two port memory in which addresses can be sourced from either the CRT controller (U6) or the 8085 uP (U1). It consists of a single 2K x 8 static RAM, allowing the screen to be formatted as 24 data rows of 80 characters each.

Video timing is provided by the dot-clock crystal, Y2 (**Figure 5.2**) and associated circuitry 74S04 (U30A,B), C4, C5, R8 and R9, making up the dot-clock oscillator. The 74S04 (U30C) buffer's the video dot clock (VDC) input of the CRT 8002. The pull-up resistor, R2, is used at the output of U30C to guarantee the logic 1 requirement of the VDAC input, see **Figure 5.3** "AC timing diagram". The video dot clock, 17.0748 MHz, which

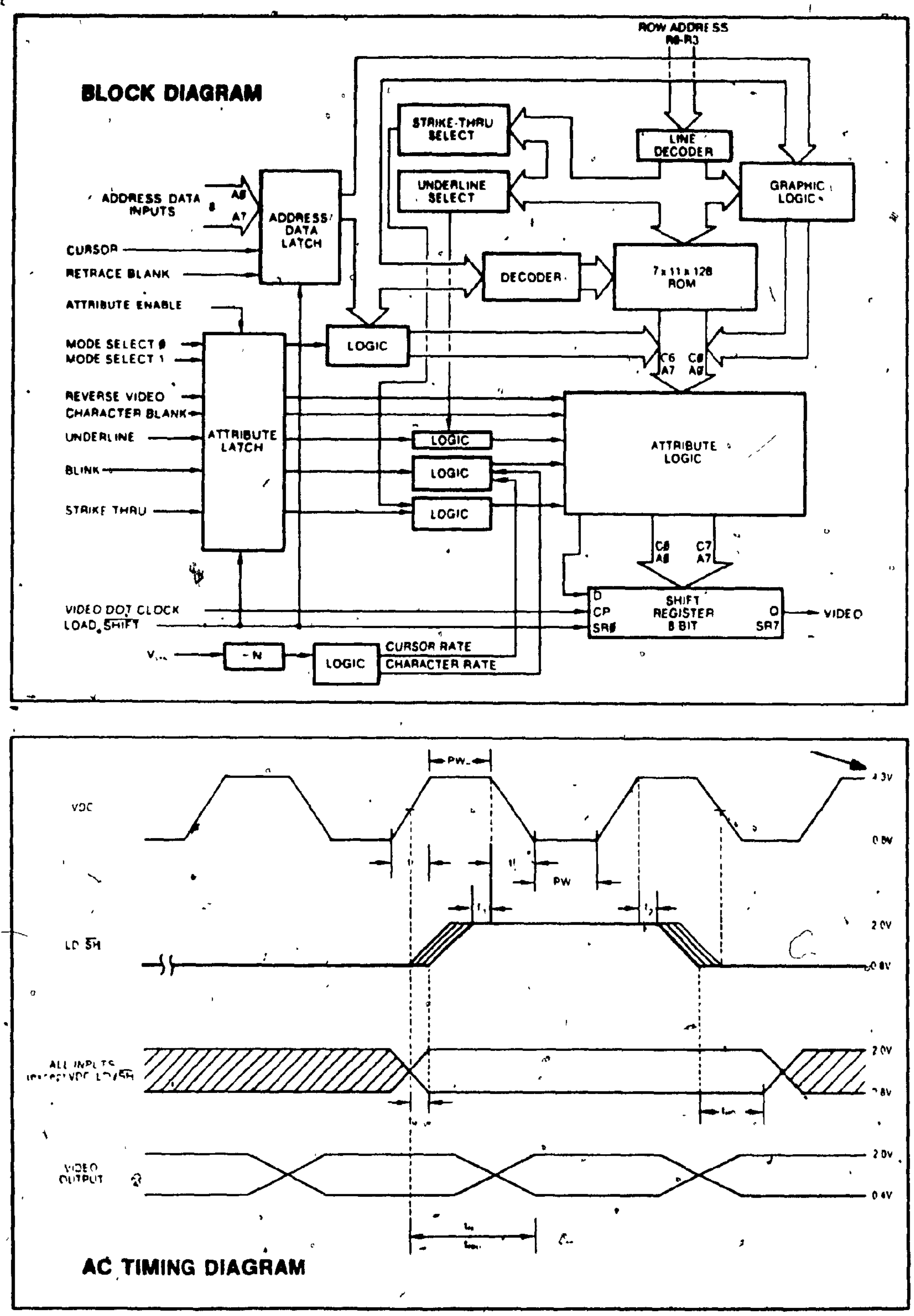
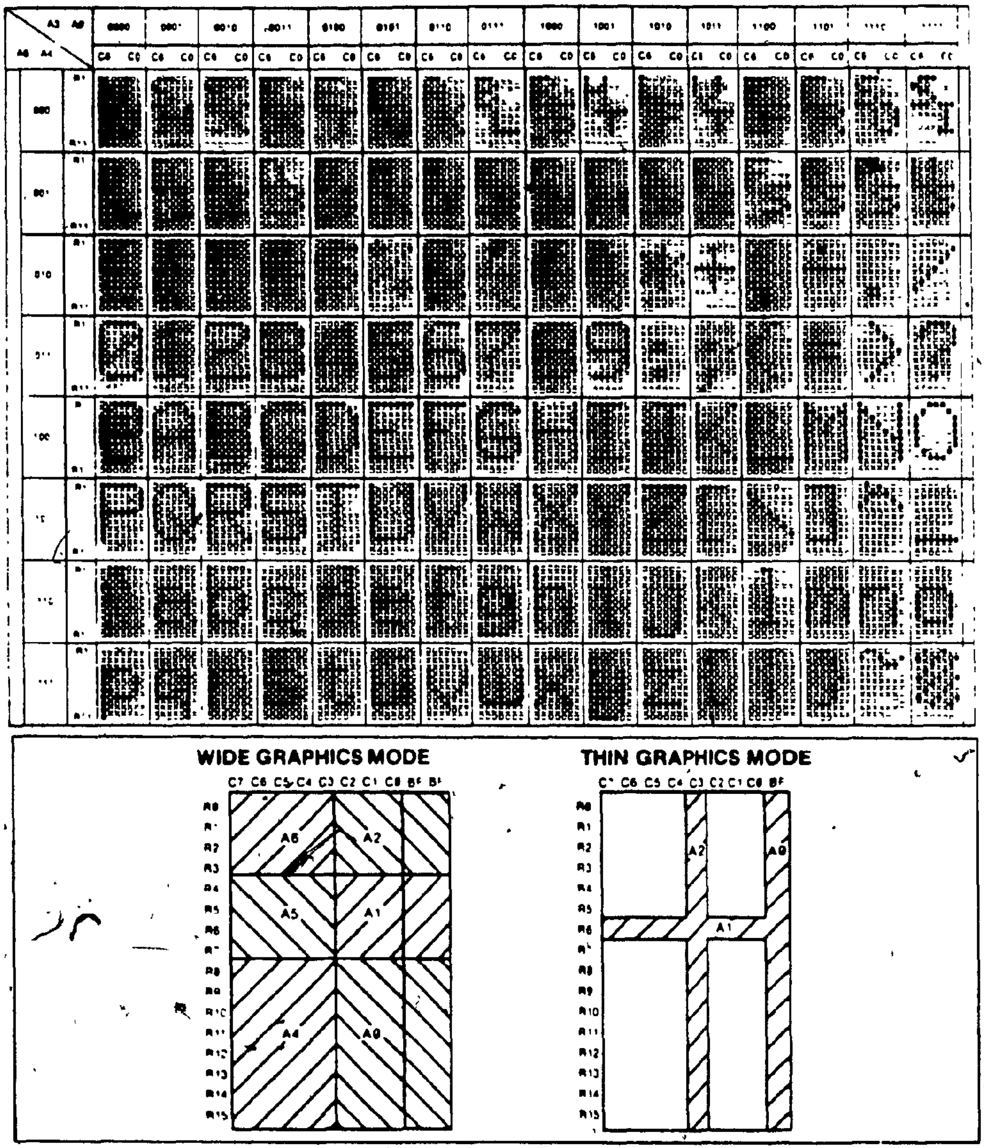


FIGURE 5.3

CRT 8002 (VDAC) Block Diagram and Timing Specifications[42,43]

CRT 8002-011
(ASCII)
CODING INFORMATION

CRT Video Display-Controller Video Generator VDAC™



ATTRIBUTES

Underline
Underline will be a single horizontal line at R11

Cursor
Cursor will be a blinking reverse video block blinking at 3.75 Hz

Blink Rate
The character blink rate is 1.875 Hz

Strike-Thru
The strike-thru will be a double line at rows R5 and R6

FIGURE 5.4

CRT 8002 (VDAC) Built-in Character Generator ROM, Graphics Mode and Attribute Specifications[42,43] determines the actual video data rate, is divided down by 9, by means of the 74LS161 (U31) synchronous 4-bit binary counter, to provide the LOAD (LD) signal to the video shift register within the CRT 8002. The same LOAD signal is delayed by a dot time by 74LS74 (U14B) to provide the character clock (CCLK) to the CRT 5037 display controller. The character clock determines the speed each character is addressed.

The CRT 8002 requires a mimimum 8 x 12 character block matrix to form its basic 7 x 11 character, in order to provide one line and one dot spacing between characters; however, to allow fully framing a character for a reverse video presentation, the horizontal character block must be increased to 9 or 10 dots, i.e., 2 or 3 dot spacing between characters. For the same reason, allocating 13 lines per character allows top and bottom framing as well. In this application, the character block designed-in was 9 x 12, (binary counter (U31) divides by 9), in order not to increase the horizontal scan frequency above 20 KHz (resulting in a price increase for the video monitor), and keep the dot clock well below the maximum limit of the CRT 8002.

The 5037 VTAC provides display addresses to the video RAM (U5) which sends ASCII character data to the VDAC. The VDAC (U7) in turn converts the ASCII data to serial video and shifts it out to the driving circuitry. The VTAC provides horizontal, vertical, blanking and composite sync signals as well, which together with the VDAC's video signal, are buffered via the 74LS244 (U10). To provide interfacing flexibility, the terminal was designed with two types of video monitor signal outputs.

Connector J4 (see Appendix C) provides a composite video interface (TTL level) by means of transistor Q1 (2N3904), used in a mixer-driver configuration. Connector J3 (Appendix C) provides separate horizontal sync, vertical sync and video signals (TTL levels) for monitors with such input requirements. Variable resistors VR1 and VR2 are optional and may be incorporated to provide contrast and brightness control, depending on the monitor's built-in circuitry. Note as well, that the polarity of the vertical drive signal (VPOL) is controlled by the setting on DIP switch S2 pin 10 (Appendix E) and the 74LS86 (U18D). Depending upon the monitor's requirements, the signal polarity may be configured as either active high or low.

The video RAM's address generated by the VTAC is gated through selectors 74LS157 (U23 to U25) to allow address mapping to take place from row-column format to binary format by the 74LS157s (U26, U27), in order to address the 2K-byte display memory (U5). During the display time, data is fetched from video memory, by means of the addresses generated by the CRT 5037, and passed to the CRT 8002 VDAC data and attribute lines; during blanking, the 74LS157 (U28) and 74LS08 (U29) force a "Normal Video" attribute into the VDAC. The half-intensity attribute is pipelined through 74LS74 (U15) as $\overline{\text{HINT}}$; this is combined with the VIDEO output signal of the VDAC and sent to both J3 and J4 monitor connectors.

The VDAC handles a variety of character attributes, besides "half-intensity" which has been described above. The others, supported by this particular design are:

- underline, blink, reverse video, blank, block graphics and line graphics.

The VDAC separate attribute inputs, along with the attribute enable strobe (ATTBE) input, allow system operation with either **embedded** or **invisible** attributes. They are defined as follows:

Embedded Attributes: is a method of generating attributes by inserting, or embedding, attribute characters within the displayable character stream. All subsequently displayable characters will exhibit the attributes defined by the last attribute character encountered. A displayable character position is lost each time an "embedded attribute" is used.

Invisible Attributes: is a method of generating attributes using a wider memory.. Each character will carry attribute bits appended to the character allowing for attributes to change as often as every character, with no loss of display position.

For 8-bit systems, embedded rather than invisible attributes are used, since the latter require more than 8 bits. An elaborate discussion will be carried out later under "Attribute Formatting".

Calculation of Screen Format Parameters to Program the CRT 5037 Registers. The VTAC is flexible enough to allow one to design this terminal with a variety of register programmed screen formats. However, it should be noted that the screen formats are limited only by the video RAM memory size. Screen formats with greater total character counts than 24 x 80 are certainly possible. Generally, a different initialization of the VTAC, a faster crystal and a faster horizontal scan rate of the monitor are all that is required. One also has the ability to increase the number of characters on the screen by running the monitor in an interlace format which doubles the number of data rows on the screen. In this case, the VTAC will provide the odd field/even field timing signals necessary for interlace operation when desired.

Programming the 5037 is a relatively easy task. However, in order to pick the correct video dot clock frequency and to program the registers in the VTAC it is first necessary to determine several key parameters. Among these parameters we need:

the vertical refresh rate = 60 Hz
the number of displayed characters = 80
the number of data rows = 24
the character block matrix format = 9 x 12
the monitor's horizontal sweep frequency = 18.6 KHz

The remaining information can now be calculated by the designer if it is carried out in an orderly fashion so as to determine the other relevant frequencies and thus derive the register contents from the above parameters. To do so, **Table 5.1** was used; the known parameters are entered in the table to simplify calculations as they are calculated.

Since the horizontal sweep frequency is known, the time per scan is calculated:

t_{scan} = 1/18.6 KHz = 53.76 us (Step 12 in **Table 5.1**)

The value for step 11 in **Table 5.1** becomes:

$$\underset{\text{(Step 11)}}{\text{total no. of scan lines}} = \frac{\text{Horiz. sweep freq.(Step 12)}}{\text{Vert. frame rate (Step 5)}} = \frac{18600}{60} = 310$$

1. H CHARACTER MATRIX (No. of Dots):........................... 7 } VDAC's
2. V CHARACTER MATRIX (No. of Horiz. Scan Lines):............. 11 } ROM
3. H CHARACTER BLOCK (Step 1 + Desired Horiz. Spacing (2)
   = No. in Dots):.. 9
4. V CHARACTER BLOCK (Step 2 = Desired Vertical Spacing (1)
   = No. in Horiz. Scan Lines):............................... 12
5. VERTICAL FRAME (REFRESH) RATE (Freq. in Hz):............... 60
6. DESIRED NO. OF DISPLAYED DATA ROWS:........................ 24
7. TOTAL NO. OF ACTIVE "VIDEO DISPLAY" SCAN LINES
   (Step 4 x Step 6 = No. in Horiz. Scan Lines):..............288
8. VERT. SYNC DELAY (No. in Horiz. Scan Lines):............... 0
9. VERT. SYNC (No. in Horiz. Scan Lines; T = 161.28 us*):...... 3
10. VERT. SCAN DELAY (No. in Horiz. Scan Lines;
    T = 1.021 ms*):... 19
11. TOTAL VERTICAL FRAME (Add Steps 7 thru 10 = No. in Horiz.
    Scan Lines):..310
12. HORIZONTAL SCAN LINE RATE (Step 5 x Step 11 = Freq.
    in KHz):..18.6
    (53.76 us)
13. DESIRED NO. OF CHARACTERS PER HORIZ. ROW:.................. 80
14. HORIZ. SYNC DELAY (No. in Character Time Units;
    T = 2.10 us**):... 4
15. HORIZ. SYNC (No. in Character Time Units; T = 4.74 us**):.. 9
16. HORIZ. SCAN DELAY (No. in Character Time Units;
    T = 4.74 us**):... 9
17. TOTAL CHARACTER TIME UNITS IN 1 HORIZ. SCAN LINE
    (Add Steps 13 thru 16):...................................102
18. CHARACTER RATE (Step 12 x Step 17 = Freq. in MHz):.........1.8972
    (0.52709us)
19. CLOCK (DOT) RATE (Step 3 x Step 18 = Freq. in MHz):........17.0748

*Vertical Interval

**Horizontal Interval

TABLE 5.1

CRT 5037 Worksheet for the 80 x 24 Non-Interlace Screen Format

Prior to computing the value of Step 10, the vertical scan delay time (**Figure 3.15**) must be determined. **Figure 5.5** reflects the timing to be taken into consideration during horizontal and vertical intervals. The vertical scan delay information is obtained from the monitor's specifications. However, in general most monitors require a minimum vertical sync delay of 1 ms. Hence the value for Step 10 becomes:

$$\text{Step 10} = \frac{\text{vert. scan delay}}{t_{scan}} = \frac{1\text{ ms}}{53.76\text{us}} = 18.6 \approx 19$$

The vertical sync pulse width generated by the CRT 5037 is equal to **three** horizontal scans; therefore, Step 9 equals 3. The vertical sync time is calculated as follows:

$$t_{vsync} = 3 \times t_{scan} = 3 \times 53.76\text{ us} = \mathbf{161.28\ us}$$

This value is also entered in Step 9 and should be verified to be within the monitor specifications.

The value for Step 7 is now calculated:

Step 7 = Step 4 x Step 6 = 12 x 24 = **288 scan lines**

From the above we can obtain Step 8; the vertical sync delay:

Step 8 = Step 11 - (Step 7 + Step 9 + Step 10)

= 310 - (288 + 3 + 19) = **0**

Hence, the vertical sync delay will be 0. As a result, the total blanking interval, which is the sum of the "Front Porch, Vertical Sync Pulse and Back Porch" (**Figure 3.15**), is **22 scan lines** long or **1.18 ms**. Some flexibility exists in the choice of these parameters.

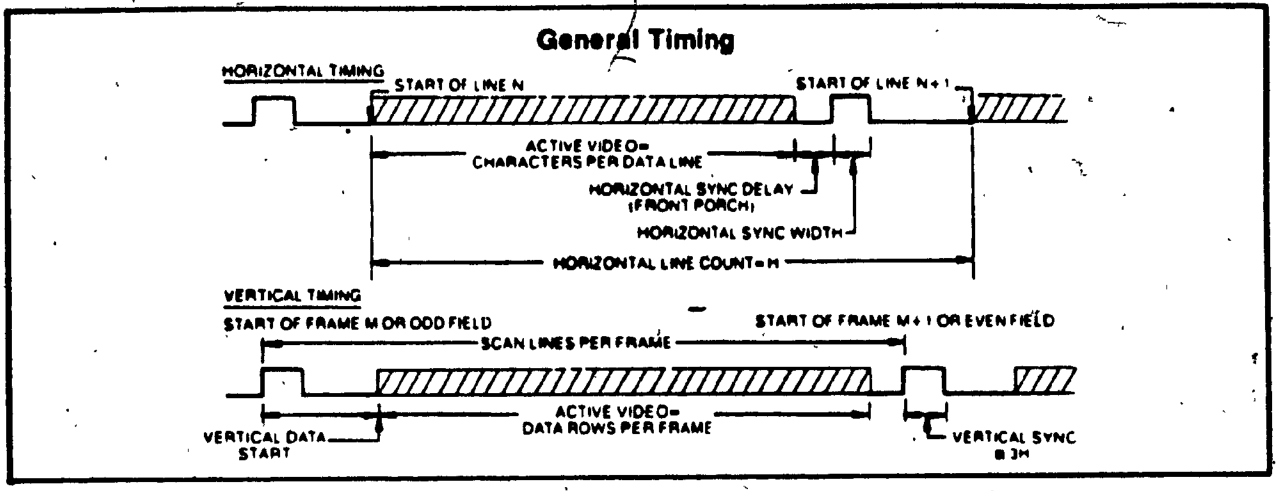


FIGURE 5.5

Horizontal and Vertical Timing Considerations
When Calculating the Screen Format Parameters[19,20,21]

To calculate the horizontal blanking interval parameters (Figure 3.14), Steps 14 through 16, we must consider that the horizontal retrace time is less than the horizontal blanking interval. In fact, this interval is normally 25% of the total horizontal scanning period. Thus, in an 80 character per row data system, this would give 20 extra characters, for a total of 100. However, we will use 102 characters to provide for some degree of flexibility, thus, a total of 22 character time units is available for the horizontal blanking interval.

In general, the horizontal sync parameters are monitor dependent. However, most monitors responding to 18.6 KHz horizontal scanning frequencies have minimum horizontal sync delays of 2 us and horizontal sync pulse widths and scan delays of 5 us each. In general, the ratio of such parameters is 1:2:2. Thus, for this application, the time for one character is given by:

$$t_{char} = \frac{1}{\text{characters x horizontal scan freq.}} = \frac{1}{102 \times 18600}$$

t_{char} = **0.52709 us** (or **1.8972 MHz** (Step 18))

We can now obtain the values for Steps 14, 15 and 16 with the value obtained above:

$$\text{Step 14} = \frac{2 \text{ us}}{0.52709 \text{ us}} = 3.79 \approx \mathbf{4}$$

$$\text{Step 15} = \text{Step 16} = \frac{5 \text{ us}}{0.52709 \text{ us}} = 9.48 \approx \mathbf{9}$$

Step 17 should match against the sum of Step 13 through Step 16, i.e., 80 + 4 + 9 + 9 = 102. Therefore, it checks out.

The video dot clock frequency may now be obtained:

Step 19 = Step 3 x Step 18 = 1.8972 MHz x 9

= **17.0748 MHz**

Having chosen the display format and calculated all the relevant parameters, the actual content for the VTAC registers can now be established, as shown in **Table 5.2.** For a detailed description of the VTACs programmable registers, refer to Section 4.3.1 and **Figures 4.5b, 4.6 and 4.7.** Note that registers 7 and 8, the Cursor Character Address and Cursor Row Address, should both be initialized to "ØØ$_{16}$" to place the cursor in the "home" position upon power-up.

REG. #	ADDRESS A3-A0	FUNCTION	BIT ASSIGNMENT	HEX.	DEC.
0	0000	TOTAL HORIZ. CHARACTER COUNT: 102 (ENTER TOTAL -1)	0 1 1 0 0 1 0 1	65	101
1	0001	INTERLACE: 0 (NONE) H SYNC WIDTH: 9 H SYNC DELAY: 4	0 1 0 0 1 1 0 0	4C	76
2	0010	SCANS/DATA ROW: 12 (ENTER TOTAL -1) CHARACTERS/ROW: 80 (CODE = 101)	X 1 0 1 1 1 0 1	5D	93
3	0011	SKEW CHARACTERS: 00 (NONE) DATA ROWS: 24 (ENTER TOTAL -1)	0 0 0 1 0 1 1 1	17	23
4	0100	SCAN LINES TOTAL/FRAME: 310 (FOR NON-INTERLACE X = (TOTAL - 256)/2) X = 27	0 0 0 1 1 0 1 1	1B	27
5	0101	VERTICAL DATA START = 3 + VERTICAL SCAN DELAY: SCAN DELAY: 19 DATA START: 22	0 0 0 1 0 0 1 1	13	22
6	0110	LAST DISPLAYED DATA ROW (TOTAL DISPLAYED DATA ROWS -1)	X X 0 1 0 1 1 1	17	23

TABLE 5.2

CRT 5037 VTAC REGISTER LOADING WORKSHEET

5.2.2 Microprocessor and I/O Section

The brain of this terminal is an 8085A (U1) uP; all operations are under its control. It runs at a 6.144 MHz input clock (Y1), which drives its internal clock generator, resulting in a 3.072 MHz cycle clock, since the input frequency is divided by two to give the processor's internal operating frequency. The crystal that drives the uP was particularly selected to allow direct derivation of the 8 necessary baud rates, in order to achieve asynchronous communication without the need for additional circuitry. The 8085A has a clock output which may be used for this purpose.

The four MSBs of the address bus (A15 to A12) are decoded by the 74LS138 (U8) to provide the various chip select signals, as configured in the system's memory map. Since the 8085A multiplexes its lower address byte and data lines, the 74LS374 (U9) latches the 8 address lines by means of the ALE signal going high during the first T state of the machine cycle, in order to keep the address stable throughout it.[44] Program instructions and constants are fetched from U4, the 2732A (4K x 8) EPROM. Provision for future system expansion has been incorporated into the design by means of U32, another 4K x 8 EPROM; address decoding for such an option was implemented in the design as well.

To further decrease component count, the 8155 (U2) RAM-I/O-TIMER device was designed in. It contains 256 bytes of RAM, sufficient for certain software scratch pad variables, the communications buffer and stack requirements. This device provides the entire system's RAM needs excluding the video RAM. The 8155 is initialized via software to have three input ports.[45] The ports correspond to:

- **Port A (DIP switch S1):** the options supported by this port control the baud rate, parity, local echo, auto linefeed and auto scroll (refer to Appendix E).
- **Port B:** interfaces directly with the keyboard's ASCII code input (refer to Appendix D).
- **Port C (DIP switch S2):** the options supported by this port control the key click, parity error indicator, 50 Hz/60 Hz operation, screen format, ON-line or OFF-line option, vertical sync polarity and printer ready polarity (refer to Appendix E).

The status of the DIP switches is checked upon power-up or system reset and is used to set up the various system configurations. The keyboard's input on the other hand is read upon a keystrobe (KSTB) occurrence. Such a strobe is filtered by R1 and C2, cleaned up by 74LS14 (U12B) and becomes a rising-edge interrupt RST7.5 at the uP. This interrupt service routine reads the new 8-bit keyboard code and acts accordingly. The different control codes (Escape sequences) recognized by the terminal are outlined in Appendix B. The programmable timer in the 8155 running from the 3.072 MHz processor clock provides a 16X baud clock to the 8251A USART through its CLOCK output. The baud rate is set depending upon the set up on DIP switch S1, thus providing an easy means for changes.

5.2.3 Memory Addressing

The VTAC addresses the display memory on a row/column basis. There are obvious advantages when using this technique (refer to Section 3.4, "Addressing the Display Memory"), such as generating oversize characters, page scrolling and software addressing. The latter is the most important since most programmers use X-Y (row-column) addressing when writing software for CRT terminals in order to facilitate the task. This was the reason for implementing in silicon such a technique as part of the CRT 5037 addressing scheme. However, the software benefits obtained when using row/column addressing generates hardware problems when the number of characters per row is non-binary, for example 80, as in this application. Addressing the display RAM in this case is wasteful of memory.

To the uP, the display memory occupies a section of its memory map, 2K in this case. When it wants to either read or write from/to this memory, the CPU does so in a linear fashion, i.e., the addresses it generates are continuous. However, the VTAC addresses this same memory on a row/column basis. Thus, a problem arises since each generates different address formats to access the same byte at a particular location. To solve this problem and still retain the advantages of row/column addressing, a hardware **address mapping** is performed, as shown in **Figure 5.6**. Thus, the CPU's linear addresses are converted to row/column addresses. Without this feature, a software algorithm would have to convert a binary address to row/column every time the uP wants to access the display memory. This algorithm would create significant overhead and program execution time. Thus, the reason for implementing it in hardware.

Considering the VTAC and **Figure 5.6**, the character column and character row outputs combine to form the character address bus. This bus, along with the uP address bus, is connected to 2-to-1 selectors, 74LS157s (U23 to U25 in **Figure 5.2**). The selector's output, (Z) in **Figure**

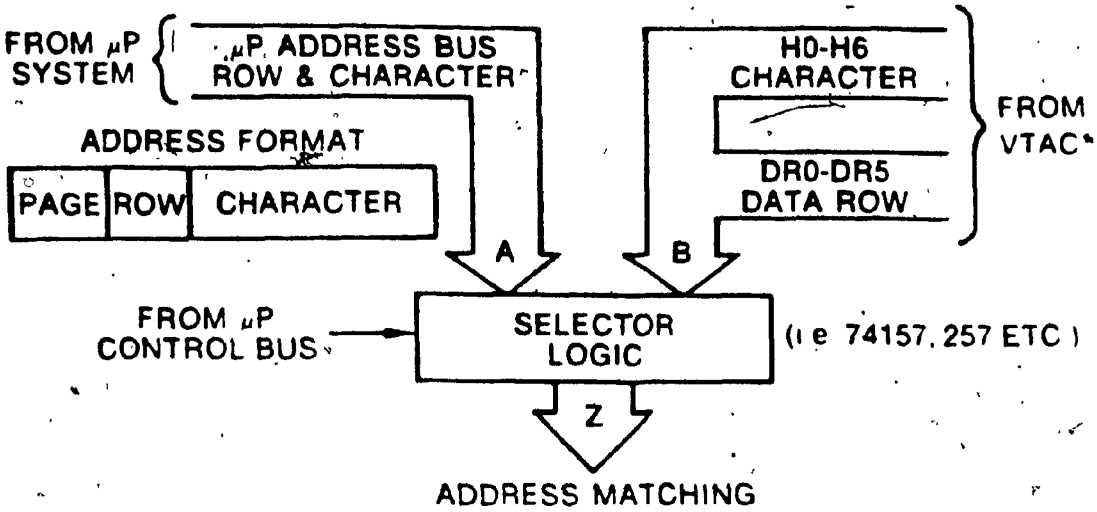


FIGURE 5.6
uP and VTAC Address Bus Mapping[50]

5.6, is decomposed into two fields, row (Y) and column or character (X) by the address mapping (or translator) devices, also 74LS157s (U26, U27 in **Figure** 5.2). The address mapping is shown in **Table 5.3** below (inputs (A) and (B), and output (Z) are with reference to **Figure 5.6**).

	SELECTOR												
uP ADDRESS BUS	INPUT (A)	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
FUNCTIONS		ROW					CHARACTER (Column)						
VTAC OUTPUTS	INPUT (B)	DR4	DR3	DR2	DR1	DR0	H6	H5	H4	H3	H2	H1	H0
SELECTOR OUTPUTS	OUTPUT (Z)	Y4	Y3	Y2	Y1	Y0	X6	X5	X4	X3	X2	X1	X0

TABLE 5.3

Address Bus Mapping for an 80 x 24 Display Format

The mapping technique is better illustrated in **Figure 5.7**. The first 64 characters are mapped directly and the next 16 characters (H6 = 1) are mapped in a higher part of the display RAM. The uP address, being linear, is overlayed onto the VTAC's address bus (row/column) via the selectors and translators. Thus, every character is addressed by its row and column from both the uP and the VTAC. The same memory location will be accessed whether the same identical address originates from the uP (in a linear format) or the VTAC address bus (in a row/column format). Note that the memory section shown as "unused" in **Figure 5.7** results from the 1920 bytes necessary for an 80 x 24 display format versus the 2048 bytes available in the 2K RAM and not from an inefficient use of memory due to the row/column addressing format.

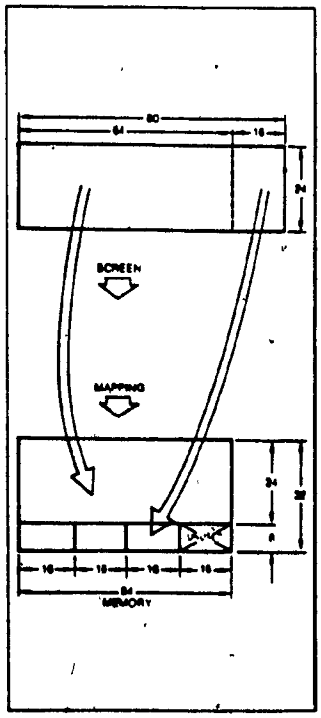


FIGURE 5.7

Address Mapping Scheme for an 80 Characters/Line
by 24 Lines Display Format[50]

5.2.4 Memory Contention

The power of a video terminal run by a uP is that the screen characters can be manipulated by simply instructing the CPU to move data from one memory location to another. The instructions that carry out such information movement are extremely fast. It is useful to be able to manipulate the information in the video RAM at the full uP speed in order to provide the throughput necessary for such tasks as communications processing and the various options and features of the video terminal. A monitor, however, has to be refreshed constantly with information to produce the dot patterns necessary to give a stable video presentation. It is therefore necessary for the VTAC to continually address the video RAM so that screen refreshing can occur. When the uP, which shares the same RAM, wishes to address it, memory contention occurs between the CPU and the CRT controller. (Section 3.4 "Interfacing the Memory to the Display"). If the uP takes over the display RAM addressing during the actual display period, thereby interrupting the display, flashing of the display occurs. Such a phenomena must be prevented.

Several memory contention solutions are possible; however, the choice of one memory contention scheme over another can greatly affect the flexibility and expandability of the terminal by affecting the uP's throughput and efficiency. The terminal described herein gives the VTAC priority in its addressing of the video RAM whenever the VTAC is reading and sending visible displayable information to the monitor. When the uP tries to address an area of video RAM, defined by the Valid Video Access (VVA) signal from the address decoder 74LS138 (U8, **Figure 5.2**), it is put into a wait state, until the horizontal drive interval (HSYNC), by lowering the READY line of the 8085A (U1) and the memory cycle is extended until the next horizontal retrace period. At this time, since no visible video information is being sent to the screen, the uP's memory cycle is allowed to conclude. Using this technique the uP has limited access to the video RAM whereas the VTAC has unlimited access to it. This contention scheme is implemented in hardware by means of the $\overline{\text{VVA}}$ and HSYNC signals and flip flop U13B to synchronize the READY line, on the CPU, using the uPs clock out (PCLK) signal. Thus, when signal HSYNC does take place, the logic circuitry made up of U11B, U12C, U16B, U16C, U17A, U18A, U19A, U19B and U19C will switch address multiplexers U23-25 to the uP bus and enable the bus transceiver 8286 for a transfer to/from the display RAM. Hence the CRT display will not flash since there can be no interference to valid video information during this interval. The uP's throughput will be reduced in this case since memory cycles can be extended for as long as 42 us, the time during each horizontal scan when information is being displayed on the CRT. If moving or manipulating all 1920 characters (80 x 24) in the video RAM is required, each access to this video RAM will take about 42 us (one access maximum per scan line) and the speed of the video operation is reduced. However, note that access to the main system memory (on the 8155 (U2)) and the program memory (U4) is maintained at full uP speed. For operations which can't wait, access via the Blanked Video Access ($\overline{\text{BVA}}$) address range will force access without waiting, and will blank the display to avoid "flashing" on it. Since one of the main design criterions for this terminal was minimum chip count, this scheme provides a fine solution to the memory contention problem. During high speed data communications, however, the incoming information that must be displayed on the monitor may not be able to be entered into the video RAM at a rate fast enough to keep up with the high speed communications channel. To allow operation at high baud rates, a communication buffer using 128 bytes of the 8155's RAM is employed.

A real-time clock, at 60 Hz is provided by 74LS74 (U13A) clocked by the vertical sync (VSYNC) signal causing an 8085A interrupt (INTR); pull-up resistors Z1 and the $\overline{\text{INTA}}$ signal from the uP disabling address decoding (U8) forces "RST7" interrupt, and $\overline{\text{INTA}}$ resets 74LS74 (U13A) display RAM access request.

5.2.5 Communications Interface

Serial communications is performed by the 8251A (U3) USART[46] through level converters MC1488 (U21 - the driver) and MC1489A (U22 - the receiver), serial out gating network 74LS32 (U17B, U17C) and 74LS86 (U18C), with a 16X baud clock provided by the programmable timer in the 8155 running from the 3.072 MHz processor clock. The terminal can generate up to 8 different baud rates determined by a 3-bit code from S1, one of the two banks of 8 position DIP switches. (Refer to Appendix E). The baud rates implemented are the most popular ones; from 110 to 19,200. Controlled by the RTS output of the 8251A, serial data can be directed to the communications port ($\overline{\text{RTS}}$ = low) or the printer port ($\overline{\text{RTS}}$ = high). Transmit and receive ready signals, TxRDY, RxRDY respectively, are ORed together (U17D) and become RST6.5 interrupt COMMINT. Audible Alarm ($\overline{\text{ALARM}}$) is buffered by 74LS244 (U10) and controlled by the SOD line from the 8085A. It drives a miniature buzzer and may serve as well as a remote alarm through connector J7 (Appendix C).

5.2.6 CRT Interface

The VTAC produces horizontal sync, vertical sync and video blanking signals to allow operation with monitors having separate, direct drive, TTL compatible outputs. This design supports such an option via connector J3 (refer to Appendix C). In addition, the composite sync output of the VTAC is combined with the video signal from the VDAC to produce a composite video signal (TTL levels), connector J4 (refer to Appendix C). The design supports such interfaces to allow the use of an 18.6 KHz horizontal sweep monitor and is compatible with the Ball Brothers TV 120 and other similar monitors.

5.2.7 Attribute Formatting

When ASCII data is loaded into video RAM, the seven ASCII bits define the particular characters to be displayed from the VDAC's character generator ROM. To take full advantage of its features, however, the VDAC requires specification of the attributes associated with each character. The four basic attributes are underline, blink, blank and reverse video. Thus, at any time a character might be displayed as blinking, and/or underlined, and/or in a reverse video field (black characters on a white background).

There are two ways to handle attributes: **Invisible** attributes require one to use a video RAM wider than 8 bits where each extra bit individually enables or disables its corresponding attribute. Although the CRT 8002 VDAC can easily handle video memory greater than 8 bits and generate the appropriate attributes, this alternative was not practical for a terminal designed for minimum cost and chip count. In addition, since the 8085A is an 8-bit uP it will easily handle an 8-bit wide video RAM. In order to keep the video memory at 8 bits, embedded attributes are used. Instead of every character carrying an attribute field, a particular attribute or combination of attributes can be initiated inserting a specifically designated attribute byte into video RAM. All characters from that point on will exhibit the attribute(s) until another attribute byte is encountered that changes the attribute byte and hence the subsequent display.

By using the attribute enable pin (ATTBE), the VDAC (U7) can maintain attributes internally and can be wired to update its internal attribute latch only during an attribute byte. Since each attribute occupies a location in video RAM, its associated display position on the screen must be blanked to prevent an erroneous display. This character position can be blanked easily by making use of the retrace blank (RETBL) pin on the VDAC. Embedded attributes, under most conditions, are neatly displayed, since they often occur when the information presented to the screen would normally be blanked. For example, the underline attribute will usually occur on a word basis. The space code normally preceding the word to be underlined is replaced by an attribute byte that starts the field and the space code following the word is replaced by an attribute byte ending the underline field. An attribute byte is recognized and distinguished from a displayable character by its most significant bit (D7) being equal to a logic one. Thus, the seven remaining bits can each individually define up to seven character attributes.

The hardware required to generate the relevant attributes is all built into the VDAC. However, some additional gating is needed externally. From **Figure 5.2**, when an attribute character is presented to the CRT 8002, blanking is forced for that character since the MSB of the video data bus, VD7, is ORed (U16D), appearing at the retrace blank (RETBL) input of the VDAC. Thus, the remaining bits, VD6 to VDØ, are latched into the VDAC as attributes. **Half-intensity** (bit VD3) is latched in 74LS74 (U15A) and delayed (U15B) in order to line up with the attributes in the VDAC at the proper character position. During retrace (active blank BL signal from the CRT 5037), or after a forced RAM access ($\overline{\text{BVA}}$ signal active), blanking by means of 74LS32 (U16A), signal CBLANK, assures a "neutral" attribute (normal video); CBLANK switches the 74LS157 (U28) and 74LS08 (U29) from the video data bus to a forced "normal video" constant.

5.2.8 Graphics Mode

The VDAC allows the user to enter 4 different display modes. An **alphanumeric** mode displays normal ASCII characters from the on-chip ROM. An **external** mode enables display of special characters not found in the on-chip ROM bypassing the internal character generator. The two other modes are **thin graphics** and **wide graphics**. All of the modes are defined via the MS1 and MSØ inputs on the VDAC. They are encoded as follows:

MS1	MSØ	MODE
0	0	wide graphics
0	1	external mode
1	0	thin graphics
1	1	alphanumeric

These 4 modes can be intermixed on a per character basis. Although several versions of the CRT 8002 are available, the wide and thin graphics mode format implemented in the VDAC utilized in this design (CRT 8002A-011) is shown in **Figure 5.4**.

The mode desired, is entered by means of an attribute character, i.e., bit 7 equal to 1. In this mode, ASCII characters are interpreted and video is generated in the VDAC in an entirely different way than in the alphanumeric mode. In the wide graphics mode, the character block is divided into 6 smaller boxes as shown in **Figure 5.4**. Depending on the content of each byte, during this mode of operation, the status of the six boxes may be defined ON or OFF thus allowing up to 64 (2^6) graphic characters to be generated. This mode is useful to generate bar charts and histograms. In the thin graphics mode of operation, bits Ø, 1 and 2 are loaded into the thin graphic logic (bits 3 to 7 are don't cares) along with the row address. This logic will define the segments of a graphic entity as defined in **Figure 5.4**.

This design supports the alphanumeric, wide and thin graphic modes. Bits VD5 and VD6 on the video data bus control the mode of operation by accessing the MS1 and MSØ inputs on the VDAC via the 74LS157 (U28).

5.2.9 Terminal Variations

Although the terminal has been designed with an 80 x 24 character format, other configurations may be easily accomplished, such as 64 x 16, by merely changing the dot rate crystal to conform to the desired format, the loading sequence of the 4-bit counter, the 74LS161 (U31) and reprogramming the initialization of the VTAC. It is that simple.

5.3 Conclusion

This chapter has described a 32-chip video terminal design with features required by today's competitive marketplace and found in terminals containing significantly more chips. With the exception of the memory contention circuitry, the hardware design is fairly straightforward due to the large number of standard MOS/LSI chips employed (see Appendix G). The "smartness" of the terminal is primarily determined by the software design. However, many of the functions implemented are due to the 5037 CRT controller and the CRT 8002 VDAC combination. Together they provide flexibility, intelligence and minimum chip count implementation. The terminal supports rates up to 19,200 Baud.

The terminal's overall features are summarized below:

Operational Features

- ASCII Code Format: supports the full upper and lowercase alphanumeric character set with descenders.
- Character Format: high resolution 7 x 11 character matrix in a 9 x 12 dot character field.
- Graphic Characters: 64 available.

- Display Format: 24 lines x 80 characters (requires 18.6 KHz video monitor with 60 Hz screen frame rate).
- Video Attributes: supports character blinking, character blanking, half-intensity, reverse video and underline.
- Allows Direct Cursor Addressing.
- Editing Features: supports character insert/delete; line insert/delete; full/partial screen erase; and tab operations.
- Keyboard Interface: Parallel ASCII encoded (negative-edge strobe).
- Video Monitor Interface: supports both direct drive and composite video outputs (TTL levels).
  contains on-board CRT brightness and contrast controls.
  compatible with Ball TV120 and other similar monitors.
- Contains on-board alarm and remote data connector.
- Power Requirements: +5V @ 1 Amp (max)
  +12V @ 100 mA (max)
  -12V @ 50 mA (max)

Communication Features

- Baud Rate: supports 8 switch selectable baud rates from 110 to 19,200 Baud.
- Parity: selectable odd, even or none.
- Contains complete RS-232C asynchronous main I/O port.
- Supports Full Duplex Communication with Local Echo option (switch selectable).
- Provides auxiliary RS-232C Printer output port.
- Character Transmit/Receive mode.
- Page Transmit/Receive mode.

CHAPTER 6

THE TERMINAL'S SOFTWARE

Having described the terminal's hardware, the software will be briefly outlined now. Presenting it in the following way provides a better perspective in order to fully appreciate the inter-relationship between the hardware and software.

6.1 Software Considerations

The memory address space has been partitioned in such a way so as to allow for minimum chip count in the decoding circuitry. Only one chip is required for such a task, U8 in **Figure 5.2**. **Table 6.1** shows the various address spaces and their assigned memory locations. Software for this terminal was implemented in modular format.

6.2 General Software Description

Upon power-on or manual reset, the CPU carries out system initialization. It consists of accessing the VTAC, the 8155 and the 8251A respectively, in order to configure their registers appropriately. By the same token, the CRT display is cleared and the DIP switches are read to configure the rest of the system as per their indicated set-up.

Software for the terminal centers around three interrupts; in order of priority they are:

1. Keyboard interrupt (RST 7.5).
2. Communications interrupt (RST 6.5).
3. Real-time clock interrupt (INT).

MEMORY ADDRESS (Hex)	MEMORY SPACE	FUNCTION
8000 - FFFF	32K	Not used.
7000 - 7FFF	4K	Not used.
6000 - 6FFF	4K	Blanking Video Access ($\overline{\text{BVA}}$) (Video Memory 2K)
5000 - 5FFF	4K	Valid Video Access ($\overline{\text{VVA}}$) (Video Memory 2K)
4000 - 4FFF	4K	VTAC Addresses ($\overline{\text{VCS}}$)
4000		Horizontal line count (register 0)
4001		Interlace bit, HSYNC front porch & width (register 1)
4002		Scans/data row, characters/data row (register 2)
4003		Skew bits, data rows/frame (register 3)
4004		Scan lines/frame (register 4)
4005		Vertical data start (register 5)
4006		Last data row displayed (register 6)
4008		Vertical cursor read
4009		Horizontal cursor read
400A		Reset command address
400B		Up scroll command address
400C		Horizontal cursor write address
400D		Vertical cursor write address
400E		Start command
3000 - 3FFF	4K	8251A USART addresses ($\overline{\text{CCS}}$)
3000		Transmit/receive registers (write/read)
3001		Control/status registers (write/read)
2000 - 2FFF		8155 RAM-I/O-Timer Addresses ($\overline{\text{KCS}}$)
2000 - 20FF		256 bytes of general purpose RAM (IO/$\overline{\text{M}}$=0)
2000		Command register (write) (IO/$\overline{\text{M}}$=1)
2000		Status register (read) (IO/$\overline{\text{M}}$=1)
2001		Port A register (DIP switch S1)_(IO/$\overline{\text{M}}$=1)
2002		Port B register (keyboard) (IO/M=1)
2003		Port C register (DIP switch S2) (IO/$\overline{\text{M}}$=1)
1000 - 1FFF	4K	Future program memory space - EPROM U32 - optional ($\overline{\text{PCS1}}$)
0000 - 0FFF	4K	Present program memory space - EPROM U4 ($\overline{\text{PCS0}}$)

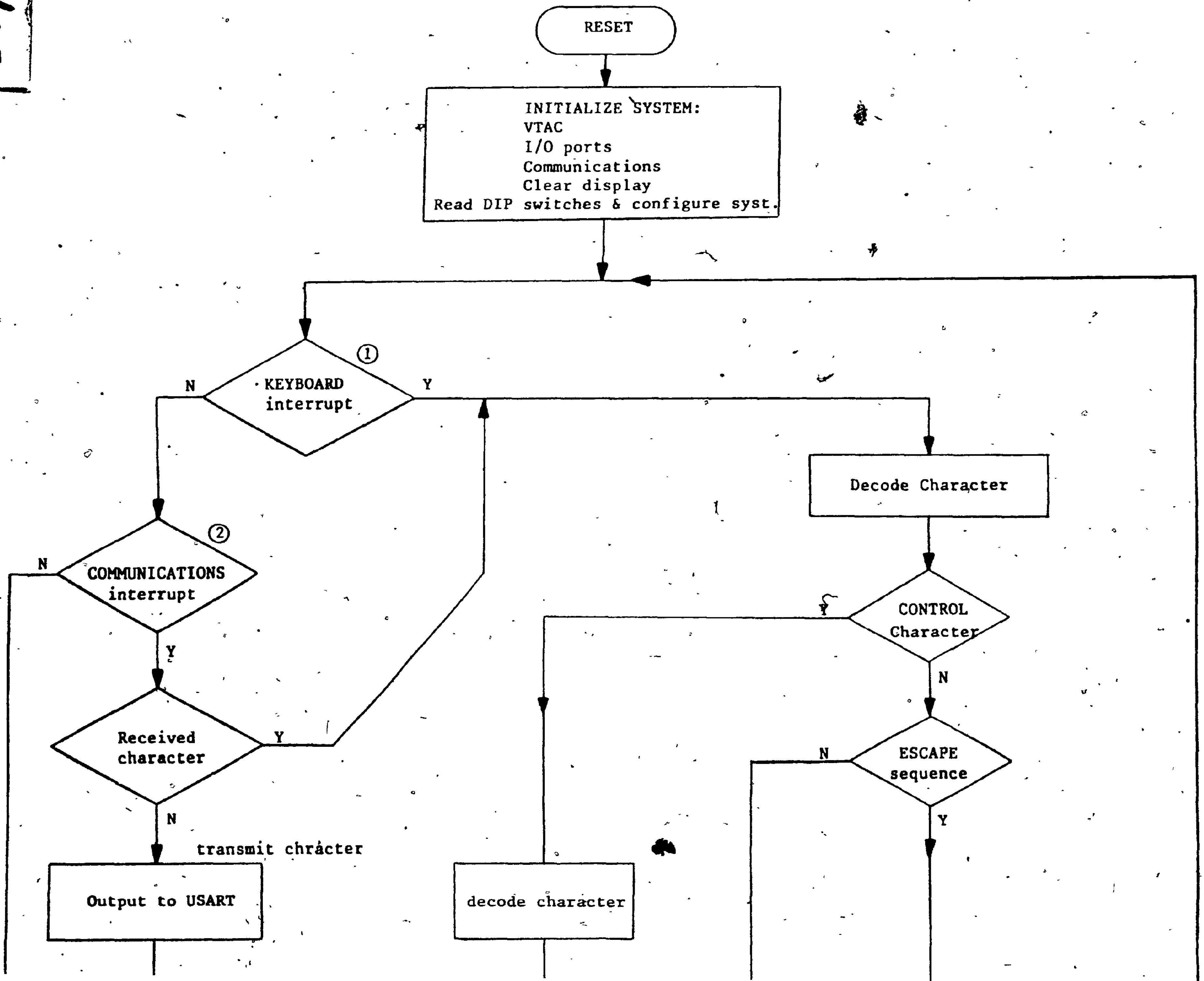
TABLE 6.1
Terminal's Memory Address Space

All other interrupts are not used.

The keyboard interrupt occurs when new keyboard data is strobed into the 8155 (U2). The key data is decoded in software and the appropriate action is taken (see Flowchart, **Figure 6.1**). The simplest routine occurs when an alphanumeric key is depressed and normal entry to the screen is required. The keyboard interrupt routine will read the character. Then the character entry routine will take the alphanumeric character, after decoding it, and write it to the cursor's present location within video RAM during the valid video access time. The VTAC has two registers that keep track of the cursor and based on the contents of these two registers the program calculates the memory location into which the ASCII data must be entered. Once character entry has occurred, the cursor's position is updated. Functional key inputs are two-key operations in which the ESC key places the terminal into an escape sequence mode (refer to Appendix B). When this mode is entered, the following ASCII inputs will be interpreted not as displayable characters but as a control function. In this way an ESC E depression can be distinguished from a normal letter E depression. The flexibility of the software is illustrated here since one can decode and initiate appropriate routines based on the type of key input sequence defined. Control codes play a similar role, however, only one-key operations are necessary. In either case, the escape or control command, if it must operate on the video memory, will wait until a valid video access takes place.

The communications interrupt allows reception and transmission of characters. If a character is received, it must be decoded, as in the

1 OF


RESET
INITIALIZE SYSTEM:
VTAC
I/O ports
Communications
Clear display
Read DIP switches & configure syst.
1
KEYBOARD interrupt
N
Y
Decode Character
2
COMMUNICATIONS interrupt
N
Y
CONTROL Character
Y
N
Received character
Y
N
ESCAPE sequence
N
Y
transmit chrácter
Output to USART
decode character

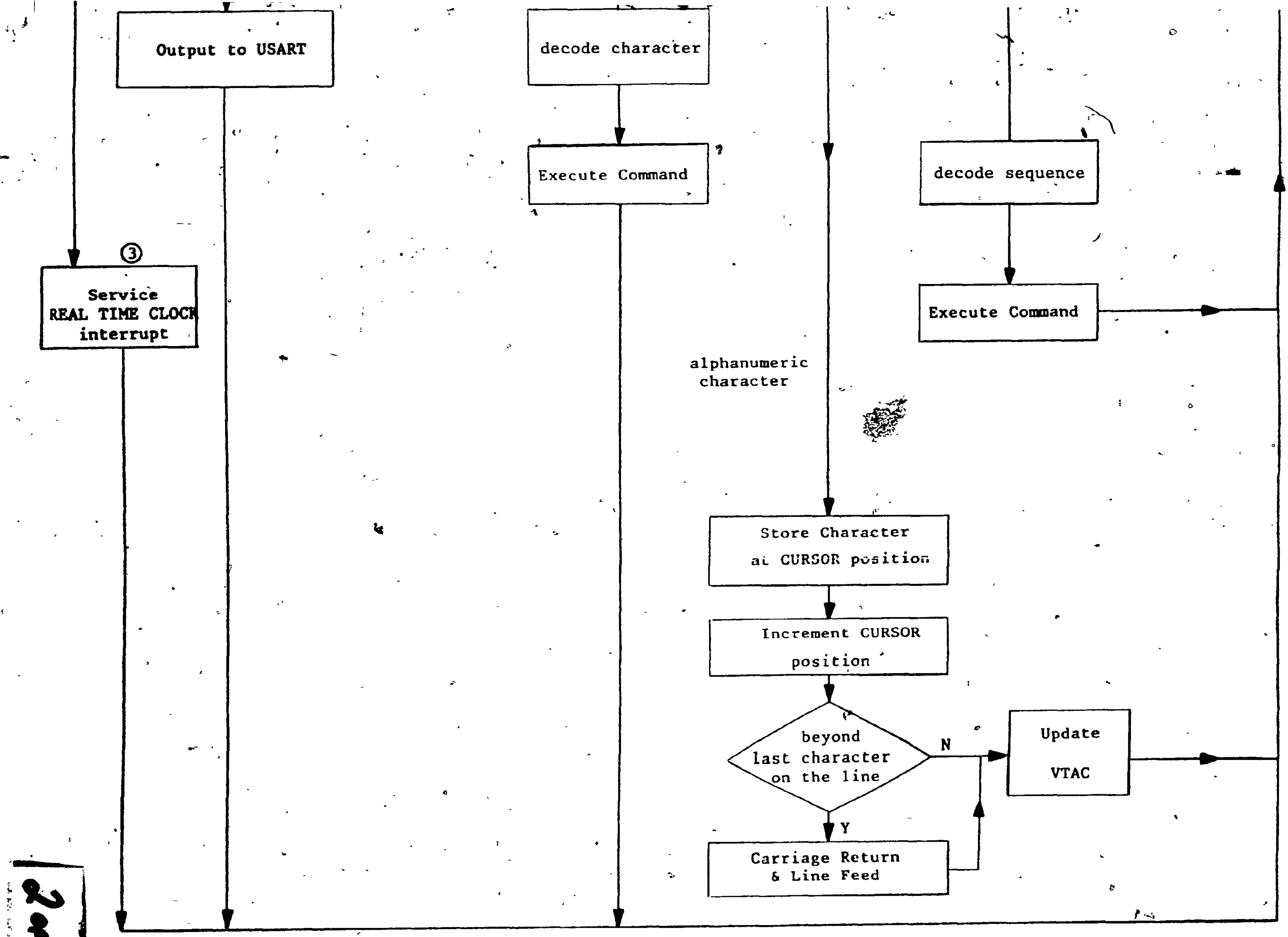


Figure 6.1
Terminal's System Software

2 of 2

case of a keyboard entry, in order for the software to determine whether it is an alphanumeric character, a control character or an escape sequence and then take appropriate action. Entire messages can also be received since every reception will raise the communications interrupt line forcing the CPU to decode the character and enter it within the video RAM. The 8251A USART serves as the communications handler over the RS232C asynchronous I/O port. Note that if the terminal is in the local echo mode (switch S1-5 open), all characters transmitted are also sent to the display.

As well, the software supports word processing features such as line insert/delete, character insert/delete and partial line/page erase. An auxiliary RS232C printer output port is also available (Appendix C - Connector J6), allowing the display's data to be routed to a printer. Appendix B describes all the control codes and escape sequences supported by the terminal via relevant software routines.

6.3 Detailed Software Description

6.3.1 Keyboard Service Routine:

The keyboard interrupt (RST 7.5) holds the highest priority among the interrupts used in this terminal's implementation. The reason for assigning the keyboard such a priority being, that the major interaction with the terminal will take place via the operator entering the radio system's parameters.

Figure 6.2 shows the keyboard's interrupt service routine. Every time the operator depresses a key, an interrupt is generated, forcing the

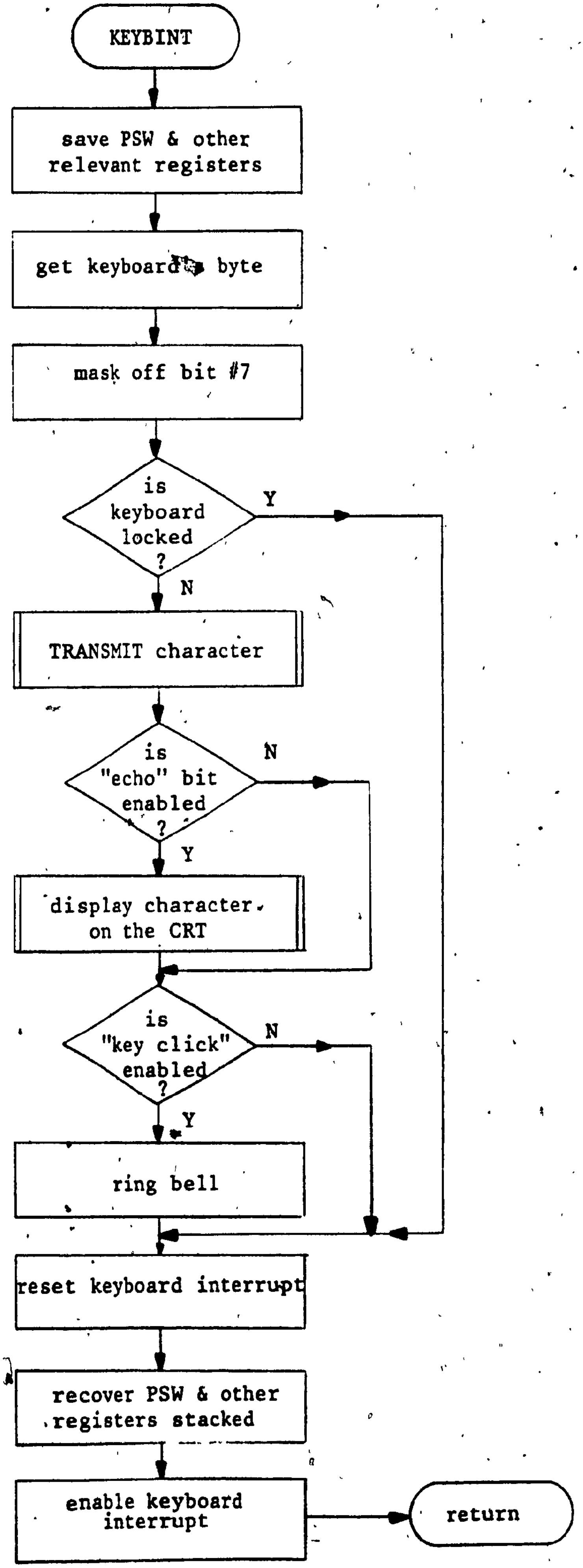


Figure 6.2
Keyboard Interrupt
Service Routine Flowchart

CPU to vector to address ØØ3CH upon completion of the instruction presently taking place and after having saved the program counter. This service routine begins by saving the present state of the flags (the Program Status Word (PSW)) and the other CPU resiters in pairs. Since the keyboard data is in ASCII format, bit 7 of the incoming byte is set to Ø and the keyboard's input character is checked for the presence of a "keyboard lock" code. On the occurrence of such a code, the CPU takes no action. However, if such is not the case, the CPU sends the relevant character to the RS232C port via the 8251A USART by calling upon the TRANSMIT subroutine. The CPU then proceeds to check if bit 3 of switchbank no. 1, port A, is set. If such is the case the character entered by the operator is displayed on the screen. Otherwise, the character will not be echoed on the display. Since the hardware implementation supports an interface to a buzzer, the CPU's next step is to check whether the "key click" option, switchbank no. 2, port C, bit 3 has been enabled. If true, the buzzer is turned ON to provide an audible key depression. The CPU then proceeds to reset the RST 7.5 flip flop by means of a SIM instruction, to recover the registers and the flags stacked, and to enable the interrupts before returning from this interrupt service routine. Page 4 of the terminal's Assembly Language Driver Program Listing (Appendix G) describes this routine.

6.3.2 Receive and Transmit Service Routines

Serial Communications, either on a data link or with a peripheral, occurs in one of two basic formats; asynchronous or synchronous. These formats are similar in that they both require framing information to be added to the data to enable proper detection of the character at the receiving end. The major difference is that the asynchronous format requires framing information to be added to each character, while the synchronous format adds framing information to blocks of data, or messages. Since the synchronous format is more efficient than the asynchronous format but requires more complex decoding, it is typically found on high-speed data links, while the asynchronous format is used on lower speed lines.

This terminal supports the **asynchronous** format since its maximum data rate is only 19200 baud. The communications handler is the 8251A USART (U3). It appends START, STOP and PARITY bits to the ASCII data sent over the RS232C link. By the same token, it extracts such bits when receiving a character, in order to place the raw ASCII data on the bus when the CPU demands it. The rate of transmission is set by the CPU upon power ON or reset by reading the baud rate setup on 3 swtiches of bank swithc no. 1 and the internal timer on the 8155 (U2). The terminal, via the 8251A flag bits, has the capability of checking for frame, overrun and parity errors taking place on the information received. The 8251A USART allows transmission to take place in either half or full duplex mode and is double-buffered, i.e., the software has a complete character time to respond to a service request.

Before reception of a character can take place, the receiver must be enabled by the RxE (Receiver enable) bit (D2) of the command instructions. If this bit is not set, the receiver will not assert the RxRDY (Receiver Ready) bit. Upon reception of a character, the RxRDY signal is asserted (active high) to indicate that a character is available. The hardware is configured such that when either the RxRDY or the TxRDY (Transmitter Ready) signal goes high it generates an RST 6.5 interrupt via the U17D OR gate (see **Figure 5.2**). The CPU services such an interrupt by vectoring to address ØØ34H.

In order to receive a character, the CPU must first enable the transmitter by setting the TxEN (Transmitter Enable) bit in the command instruction register high and the CTS input must be low. Note that the CTS input on the USART is hardwired to ground in **Figure 5.2**. Whenever the USART is ready to accept a character, the TxRDY output signal goes high. However, such a course of action on the part of the USART takes place only if transmit Data buffer is empty and the USART has been enabled to transmit.

Since both RxRDY and TxRDY generate an interrupt, how does the CPU know which of the two, the transmitter or the Reciever, has generated the interrupt? It does so by polling. When the interrupt is recognized, the CPU chekcs the condition of the RxRDY and TxRDY flags in the USART's status register by means of a read operation. Depending upon which of the two ready flags is active, the program branches accordingly to service the flag that has taken place, **Figure 6.3.**

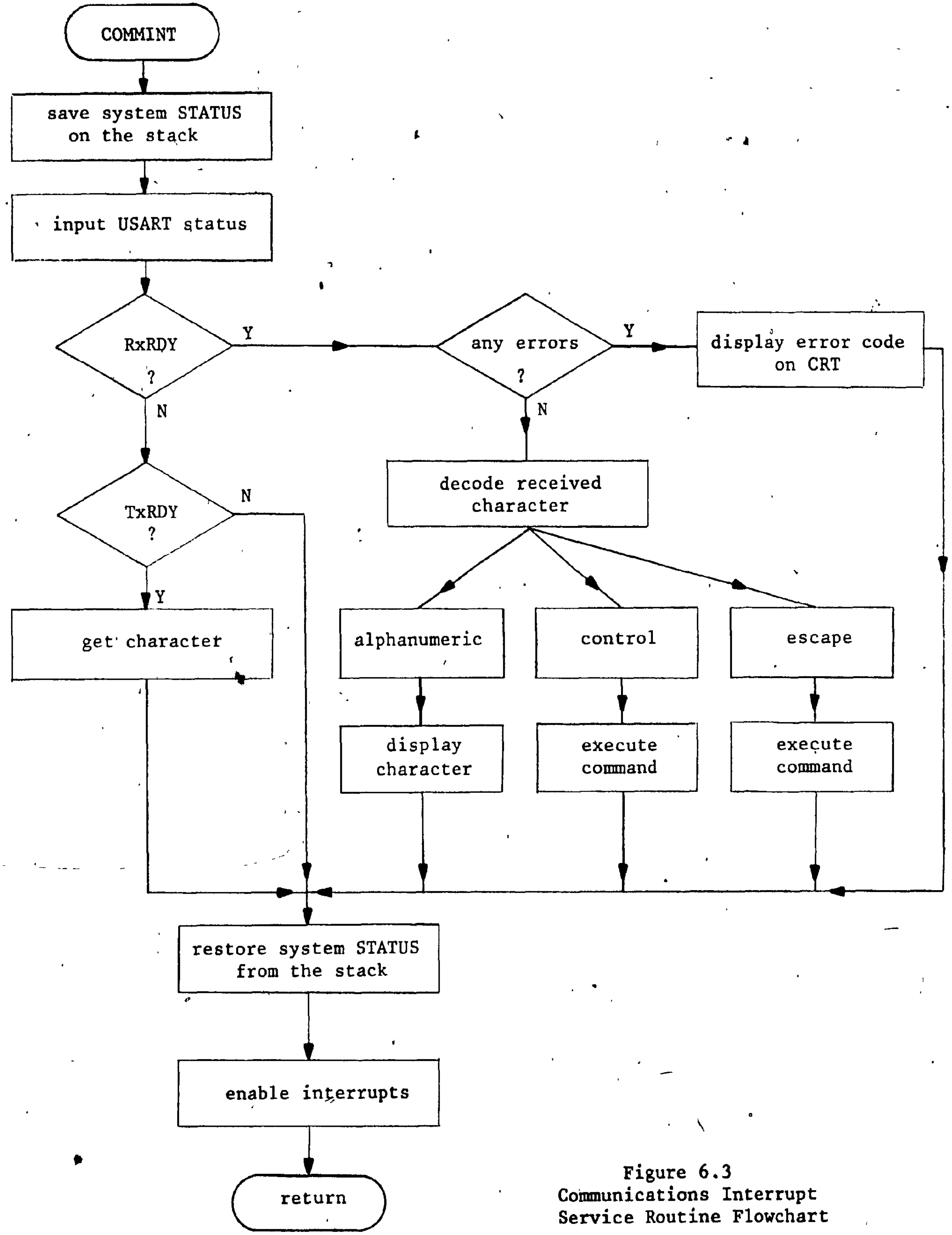


Figure 6.3
Communications Interrupt
Service Routine Flowchart

In the case of a received character, the program checks the 8251A status register flags for any presence of either frame, overrun or parity errors. If such is the case, it displays on the CRT a DELETE character in the place of the erroneous character in order to indicate to the operator the error. If no errors are present, the character is decoded. Depending upon the character, one of three things can take place:

1. if **alphanumeric** - the character is displayed right away;
2. if a **control character** - the control command is identified and the relevant action is carried out;
3. if an **escape sequence** - the CPU waits to identify the next character(s) to fully identify the escape sequence and executes the pertinent function associated with it.

If a character is to be transmitted, the CPU transfers the character to the USART via the data bus and the USART takes care of appending the START, PARITY and necesary STOP bits.

Note that upon entering the communications interrupt routine the CPU saves the state of the system; by the same token, before exiting the routine the status of the running program is restored, the interrupts are enabled, followed by a return.

Figures 6.4 and 6.5 show the different decoding functions taking place at a deeper programming level when either control characters or escape sequences are recognized.

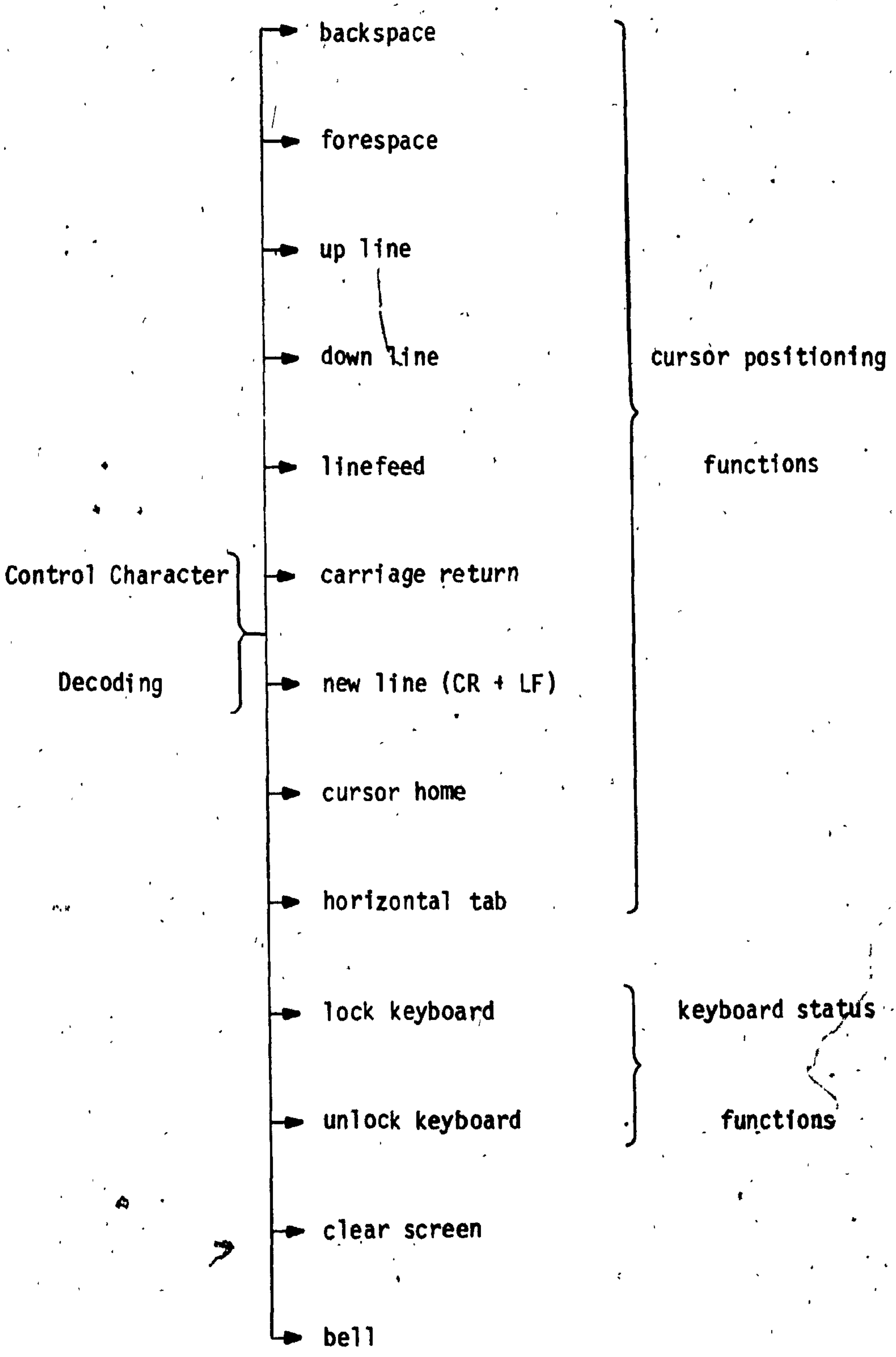


FIGURE 6.4

Control Character Decoding Chart

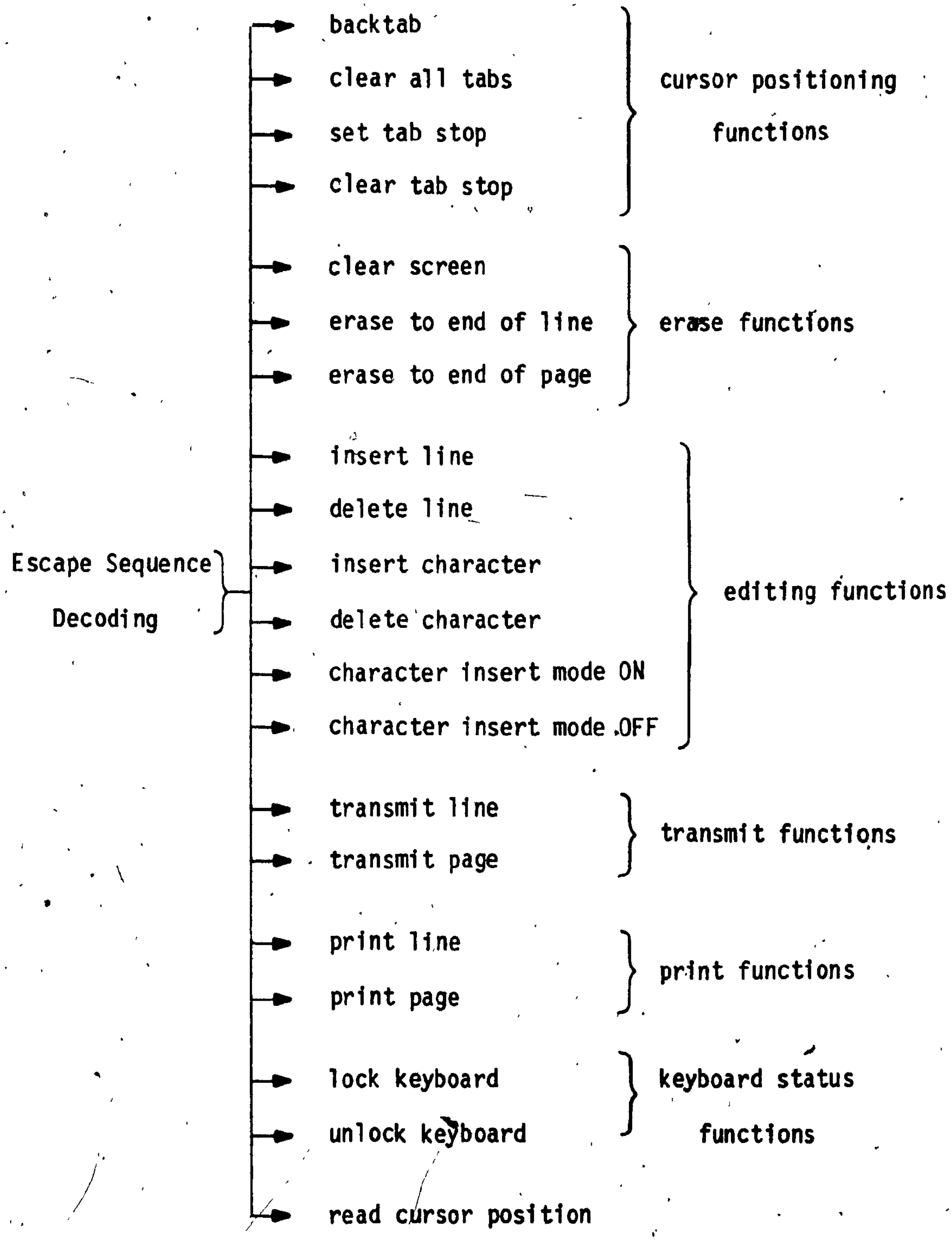


FIGURE 6.5

Escape Sequence Decoding Chart

6.3.3 Video Access and Interrupt Handling

Since the VTAC is given priority to access the video memory and the CPU can only do so during the horizontal retracing, for every scan line (total duration of 54 usecs), 42 usecs are allocated to the VTAC and 12 usecs to the CPU. Considering that the terminal has been set up to run at 19200 baud maximum, and that each character is made up of 10 bits, 1 START bit, 7 DATA bits, plus 1 PARITY bit and 1 STOP bit, then a character will be received every 521 usecs [$(19200)^{-1}$ x 10]. If the CPU tries to access the video memory during the 42 usecs allocated to the VTAC, flip flop U13B will bring the CPU's READY line low, forcing the CPU to enter a wait state, until a horizontal sync takes place. Thus, if an interrupt takes place while the READY line is low, it will not be serviced until the READY input returns to a high, but only once the CPU has finished the instruction that was presently taking place. Considering that the CPU clock has a period of 0.325 usec, that the longest 8085A instruction is 18 T-states (i.e., 5.85 usecs) and the interrupt took place just at the beginning of the display enable area, then in order to service the interrupt, the CPU must wait:

$$42 \text{ usecs} + 5.85 \text{ usecs} = 47.85 \text{ usecs}$$

Since a character is being received every 512 usecs, this leaves the CPU with 464.75 usecs to get the received character and process it. This is enough time considering the CPU's clock rate. To further prove the point, at such a clock rate and considering an average instruction takes 9 T-states, then within 464.75 usecs the CPU will have enough time to carry out 159 instructions. Sufficient processing to get the character and display it, or decode it and carry out the pertinent function associated with it. However, since the longest software routine is made up of 94 instructions, and considering an average of 9 T-states per instruction, this loop will take a maximum of 274.95 usecs. This is well within the 464.75 usecs available to the CPU to decode and process the incoming character. Thus, operation of the terminal at 19200 baud is guaranteed.

6.4 Conclusion

The terminal herein described provides a level of intelligence to classify it under the "smart" category. It was implemented by optimizing the hardware requirements to a minimum of only 32 chips, thus requiring low power, and providing greater reliability - all at a low cost! Centered around an 8-bit uP, the 8085, and several VLSI peripheral chips, along with an LSI CRT controller and a companion attribute controller, the only other system requirements are a keyboard and a CRT. The compactness as well as the speed and the intelligence were obtained by blending chip functionality in order to achieve the best performance out of every device. The "smartness" of the terminal was primarily determined by the software design. It provides the operator with sufficient flexibility to edit, print, transmit and receive information, among other functions, and yet incorporates sufficient intelligence to ease the task.

In closing, the hardware design (**Figure 5.2**), the software listing (Appendix G), and all the information provided herein should compliment the final product; my personal 32-chip "smart terminal" - as opposed to a bulky and costly off-the-shelf CRT terminal.

APPENDICES

APPENDIX A

ASCII CODE CHART

				b6	0	0	0	0	1	1	1	1
				b5	0	0	1	1	0	0	1	1
Bits				b4	0	1	0	1	0	1	0	1
b 3	b 2	b 1	b Ø	Column Row	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	´	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10(A)	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11(B)	VT	ESC	+	;	K	[	k	{
1	1	0	0	12(C)	FF	FS	,	<	L	\	l	¦
1	1	0	1	13(D)	CR	GS	—	=	M	]	m	}
1	1	1	0	14(E)	SO	RS	.	>	N	^	n	~
1	1	1	1	15(F)	SI	US	/	?	O	—	o	DEL

NUL	Null
SOH	Start of Heading
STX	Start of Text
ETX	End of Text
EOT	End of Transmission
ENQ	Enquiry
ACK	Acknowledge
BEL	Bell (audible or attention signal)
BS	Backspace
HT	Horizontal Tabulation (punched card skip)
LF	Line Feed
VT	Vertical Tabulation
FF	Form Feed
CR	Carriage Return
SO	Shift Out
SI	Shift In
DLE	Data Link Escape
DC1	Device Control 1
DC2	Device Control 2
DC3	Device Control 3
DC4	Device Control 4 (Stop)
NAK	Negative Acknowledge
SYN	Synchronous Idle
ETB	End of Transmission Block
CAN	Cancel
EM	End of Medium
SUB	Substitute
ESC	Escape
FS	File Separator
GS	Group Separator
RS	Record Separator
US	Unit Separator
DEL	Delete

APPENDIX B

CONTROL CODES, ESCAPE SEQUENCES AND DISPLAY OPERATION

NOTE: Blank code is b; zero is Ø.
Control code or special character is indicated by <> brackets.

CURSOR POSITIONING

CODE — FUNCTION

<BS> Move cursor LEFT one space; if currently at Left margin, position to Right Margin, Previous Line; if at Top of Screen, position to Right Margin, last row. (Top of screen is first row, left Margin). **(backspace)**

<FF> Move cursor RIGHT one space; if currently at Right margin perform <CR>. **(forespace)**

<VT> Move cursor UP 1 row; if currently in first row, move to last row. **(up line)**

<LF> Move cursor DOWN one row; if currently in last row and Auto Scroll enabled, scroll; if Auto Scroll disabled, move to first row. **(linefeed)**

<CR> Move cursor to Right margin; if Auto LF enabled, perform <LF>. **(carriage return)**

<ESC = <row> <col> Position cursor to row, column as specified by characters <row> <col>. (See "Absolute Cursor Position chart, Appendix F".) If a value is out-of-range, it is limited to its maximum value.

<RS> Position cursor to Top of Screen. **(cursor HOME)**

<US> Perform <CR><LF>. (Ignore Auto LF for <CR>). **(new line)**

<HT> or <ESC> i Advance cursor to next tab stop on row; if no more, perform <US> and advance to first tab stop on next row; if no tabs set, perform <FF>. **(horizontal tab)**

CURSOR POSITIONING (cont'd)

CODE	FUNCTION
<ESC> I	Move cursor backwards to previous tab stop on line; if none, go to previous row, right margin and move backwards to last stop on row; if no tabs set, perform <BS>. **(backtab)**
<ESC> Ø	Clear all tab stops (power-up sets tab stops at cols. 8, 16, 24, ...).
<ESC> 1	Set tab stop at current cursor column.
<ESC> 2	Clear tab stop at current cursor column.

Erases

<SUB> or <ESC> * or <ESC> +	Clear screen, position cursor to top of screen. **(clear all to null)**
<ESC> T or <ESC> t	Erase from cursor (inclusive) to end of current line. **(line erase null)**
<ESC> Y or <ESC> y	Erase from cursor to end of page. **(erase page to null)**

Editing Functions

<ESC> E	Insert Line: move cursor to left margin, move (current row) to (last row-1); DOWN one row, deleting last row; erase current row.
<ESC> R	Delete Line: Move cursor to left margin; move (current row +1) to (last row); Up one row; erase last row.
<ESC> Q	Insert character; move (current character) to (last character-1) on current row; RIGHT one column, deleting last character; blank current character.
<ESC> W	Delete character: move (current character +1) to (last character) on current row; LEFT one column, deleting current character; blank last character.
<ESC> q	Character Insert Mode ON. **(set Insert Mode)**

CURSOR POSITIONING (cont'd)

CODE	FUNCTION
<ESC> r	Character Insert Mode OFF (power-up default). If this mode is ON, an (<ESC> Q) is performed before ANY character is stored in the screen. **(clear Insert Mode)**
<ESC> 2 <char>	Store <char> in screen regardless of value. For storing control characters and <DEL> in screen.
<ESC> G <char>	Store Attribute character in screen. <char> is bit-encoded.

Other Functions

<ESC> 4	Transmit from start of current row to cursor (inclusive), normal. **(send line)**
<ESC> 5	Transmit from start of page to cursor (inclusive), normal. **(send page)**
<ESC> 6	Transmit from start of current row to cursor (inclusive), literal. **(send line all)**
<ESC> 7	Transmit from start of page to cursor (inclusive), literal. **(send page all)**
<ESC> P	Print from cursor (inclusive) to end of page, literal. **(print page all)**
<ESC> p	Print from cursor (inclusive) to end of page, normal.

Transmit will send out via main communications port; Print will send out via Printer Port. "Literal" translates attribute characters to blanks; "Normal" translates attribute characters to <ESC> G <attr> sequences. In all cases, end-of-line is indicated by <CR> code if Auto LF enabled, <CR><LF> codes if Auto LF disabled.

<ESC> ?	Read Cursor Position: terminal will reply over communications port with sequence <ESC> Y <row> <col> , where row col as specified in Absolute Cursor addressing.
<SO> or <ESC> "	Unlock keyboard. Re-enables key strobes. **(shift out)**

CURSOR POSITIONING (cont'd)

CODE	FUNCTION
<SI> or <ESC> #	Lock keyboard. Prevents key strobes from entering data. **(shift in)**
<BEL>	Sound alarm for 0.5 seconds.

NOTE: The control codes and escape sequences described above are compatible with Lear Siegler's ADM-42 and ADM-3A terminals.[48,49]

APPENDIX C

CONNECTOR (INTERFACE) CONFIGURATIONS

Communications Port Interface (J1)

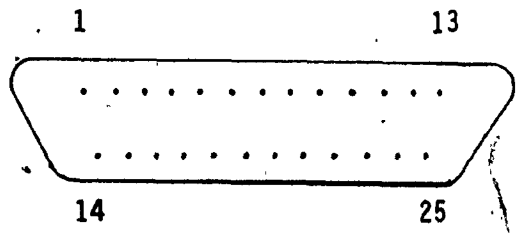
Connector J1 is provided for a communications interface. This interface conforms to EIA standard RS232C/CCITT standard V.24 and appears as DTE (Data Terminal Equipment).

Connector J1 pin	Signal Name RS232C*	V.24*	Terminal	Source
1	AA	101	Frame Ground 1/(opt)	Terminal
2	BA	103	Transmitted data (TxD)	Terminal
3	BB	104	Received data (RxD)	Comm. Line
4	CA	105	Request to Send (RTS)	Terminal
5	CB	106	Clear to Send (CTS)	Comm. Line
6	CC	107	Data Set Ready (DSR)	Comm. Line
7	AB	102	Signal Ground	Comm. Line
13	SCB	121	Printer Ready	Printer
14	SBA	118	Printer Data	Terminal
20	CD	108.2	Data Terminal Ready (DTR)	Terminal

*Reference [47]

NOTE: Printer port connections also appear on J1; these occur in positions reserved for reverse channel operation, and conform (per voltages and source) to the standards, but are not for use with a half duplex modem!

J1 mates with Cinch DB25S connector or equivalent.



J1 (face edge view)

Voltage Levels:

Inputs: "ON", "SPACE": $3V \leq Vin \leq 15V$
"OFF", "MARK": $-15V \leq Vin \leq -3V$

NOTE: If not connected, the following inputs appear "ON" or "SPACE":

Clear to Send (CTS), Printer Ready

If not connected, the following inputs appear "OFF" or "MARK":

Received Data (RxD), Data Set Ready (DSR)

Outputs: "ON", "SPACE": Vo = 12V ± 4V
"OFF", "MARK": Vo = -12V ± 4V

Descriptions:

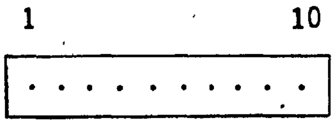
- Frame Ground 1 - connects to signal ground via jumper JP1 (normally connected).
- Signal Ground - common reference for all other signals.
- Transmitted Data (TxD) - serial data from terminal to communications line.
- Received Data (RxD) - serial data to terminal from communications line.
- Request to Send (RTS) - ON means terminal may transmit to communications line at any time. This signal goes on after terminal's power-on, and only goes OFF while performing a local print function.
- Clear to Send (CTS) - ON permits transmission via Transmitted Data, OFF holds off transmission. This is sampled when terminal has to send a character, and does not have be synchronized to character boundaries.
- Data Set Ready (DSR) - ignored by terminal.
- Data Terminal Ready (DTR) - goes ON after terminal's power-on and stays ON.

Keyboard Interface (J2) - See Appendix D.

Monitor Interface (J3)

Connector J3 is meant for monitors requiring separate video signals. It is a 10-pin (0.156") connector header which mates with Molex 09-07-5105 or equivalent.

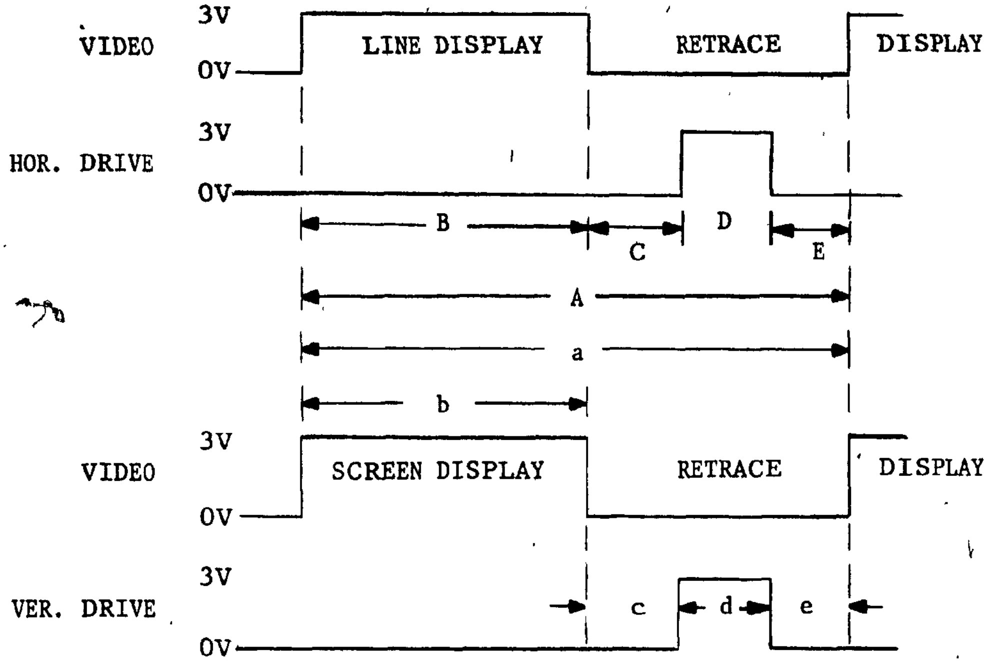
Connector J3 pin	Signal Name	
1, 5, 10	Ground	
2	CW	Brightness Control
3	CCW	
4	Wiper	
6	Horizontal Sync	
7	+12V supply	
8	Video	
9	Vertical Sync	



J3 (top view)

Signal Description:

- Ground - Signal common.
- +12V - Supply for monitor via jumper JP4.
- Brightness control - 100K Ω potentiometer on terminal for monitors with remote brightness control option.
- Horizontal Sync - positive logic pulse (see chart and timing diagram), TTL signal.
- Vertical Sync - pulse (see chart and timing diagram), TTL signal, polarity selected by switch S2-7 (refer to Appendix E).
- Video - video analog signal (see timing diagram).



Horizontal Parameter*	24x80 Screen Format
A	53.76uS
B	42.16uS
C	2.10uS
D	4.74uS
E	4.74uS
Horizontal Frequency	18.6KHz

Vertical Parameter*	24x80 Screen Format
a	16.67mS
b	15.49mS
c	0
d	161.28uS
e	1.021mS
Vertical Frequency	60Hz

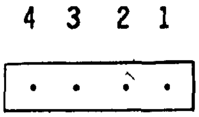
*Refer to Table 5.1 and Section 5.2 for parameter description and calculations.

Composite Video Interface (J4)

Connector J4 provides a composite video output for either a 75 cable drive or an external RF modulator (if a 64 x 16 screen format is implemented).

Connector J4 pin	Signal
1	Ground
2	Composite video, unterminated
3	75⎍ terminated to ground
4	+5V

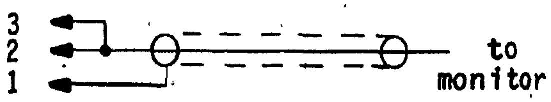
J4 is a 4 pin (0.100") connector capable of mating with a Panduit CE100F 26-4 or equivalent.



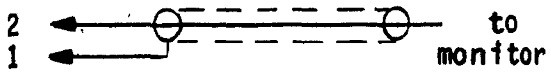
(top view)

Connection for 75⎍ cable:

(short run, unterminated monitor)



Connection for 75⎍ cable:
(to terminated monitor)



Power Supply Connector (J5)

Connector J5 provides power to the terminal. It is a 5-pin (0.156 ") header which mates with Molex 09-07-5055 or equivalent.

Connector J5 pin	Signal Name	Current (max)
1	-12V ± 10%	50 mA
2	Ground	-----
3	+5V ± 5%	1 Amp*
4	+12V ± 10%	100 mA

*Assuming the video monitor is not driven from this supply and the keyboard requirements are 250 mA maximum.

Typical and maximum terminal current consumption (excluding keyboard and monitor) is 500 mA and 750 mA respectively.

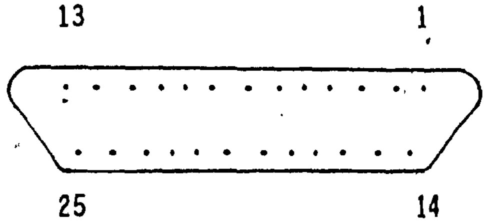
Printer Port Interface (J6)

Connector J6 is provided for a local printer. This interface conforms to EIA standard RS232C/CCITT standard V.24 and appears as DCE (Data Communications Equipment).

	Signal	Name		
Connector J6 pin	**RS232C***	**V.24***	**Terminal**	**Source**
1	AA	101	Frame Ground 2	Terminal
3	BB	104	Printer Data	Terminal
4	CA	105	Request to Send (RTS)	Internally Coupled
5	CB	106	Clear to Send (CTS)	
6	CC	107	Data Set Ready (DSR)	Terminal
7	AB	102	Signal Ground	Terminal
8	CF	109	Carrier Detect	Terminal
20	CD	108.2	Printer Ready	Printer

*Reference [47]

J6 mates with a Cinch DB25P or equivalent.



J6 (edge face view)

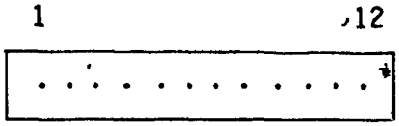
Descriptions:

- Frame Ground 2 - connects to signal ground via jumper JP2 (normally connected).
- Signal Ground - common signal reference.
- Printer Data - serial data to printer from terminal.
- Data Set Ready (DSR), Carrier Detect - always ON.
- Printer Ready - used to hold off transmission of data to printer if printer is not available. Polarity selected by Switch S2-6 (refer to Appendix E).

Multiple Interface Connector (MIC) (J7)

Connector J7, a 12-pin (0.100") header is provided for installations where a remote-connect harness is needed in place of the normal communications connections J1 and J6; it mates with a Panduit CE100F26-12 or equivalent.

Connector J7 pin	Signal Name	Reference Connection
1	Alarm	J2-12
2	Ground	J1-7
3	+5V	J2-11
4	On line	J2-8
5	Printer Data	J6-3
6	Data Terminal Ready (DTR)	J1-20
7	Printer Ready	J6-20
8	Data Set Ready (DSR)	J1-6
9	Clear to Send (CTS)	J1-5
10	Request to Send (RTS)	J1-4
11	Received Data (RxD)	J1-3
12	Transmitted Data (TxD)	J1-2



J7 (top view)

APPENDIX D

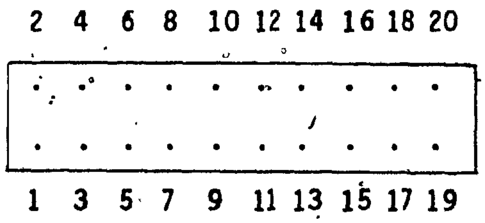
Keyboard Interface (J2)

Connector J2 is provided to interface directly with a parallel keyboard supplying seven (7) bits of ASCII encoded data and a negative-logic strobe at TTL levels. The connector's signal breakdown is as follows:

Connector J2 pin	Signal	Source
1	Key data Ø	Keyboard
3	Key data 1	Keyboard
5	Key data 2	Keyboard
7	Key data 3	Keyboard
8	On line	Keyboard
9	Signal Ground	Keyboard
10	-12V	Terminal
11	+5V	Terminal
12	Alarm	Terminal
13	Keydata 4	Keyboard
15	Keydata 5	Keyboard
17	Keydata 6	Keyboard
18	Key strobe	Keyboard
19	Key data 7	Keyboard

Unlisted pins are not used.

J2 is a 2 x 10 pin, 0.100" header; mating connector is 3M 3421-5000, Molex A-4700-20A551 or equivalent.



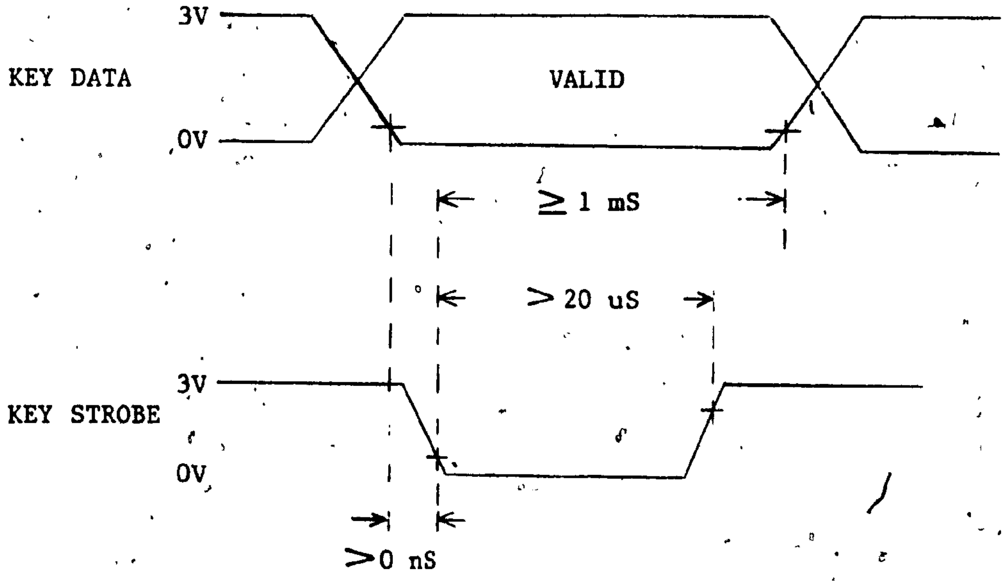
J2 (top view)

Signal Description:

- Key data 7 - not used.
- Key data 0-6 - ASCII code of struck key - true logic.
- Key strobe - negative strobe for every key struck.
- On-line - HIGH or not connected means terminal is ON-LINE;
  LOW means terminal is OFF-LINE.
- Alarm - High means alarm OFF,
  Low means alarm ON.
  (For optional keyboard-mounted alarm).
- +5, -12V, signal ground - supply and common connections.

Voltage Levels:

Inputs: Vil ≤ 0.8V @ -400uA
Vih ≥ 2.0V @ 100uA



Keyboard Signals

APPENDIX E

DIP Switches Control Functions

Certain operating features of the display terminal are controlled by the banks of switches S1 and S2 (see "circuit diagram", Figure 5.2). Each bank has eight (8) switches labelled 1 to 8; in the following discussion, ON, OFF, OPEN and CLOSED refer to the switch positions as denoted on the switch bank, and

"ON" = "CLOSED" = "0" or low

"OFF" = "OPEN" = "1" or high

DIP SWITCH S1

Auto Scroll:

S1-1 OPEN - means that an attempt to advance the cursor DOWN past the last displayed line will call the entire display up 1 line, deleting the first displayed line and bringing a blank line into the last displayed line. Note that only incremental cursor positioning will trigger this; absolute positioning will block any effort to go beyond the last line.

S1-1 CLOSED - means that any attempt to advance the cursor DOWN past the last line will put the cursor into the first line.

Auto Linefeed:

S1-2 OPEN - means that:

1. the display will interpret <CR> as <CR><LF> ; and

2. print screen and transmit functions will terminate lines with <CR> .

S1-2 CLOSED - means that:

1. the display will interpret <CR> normally; and
2. print screen and transmit functions will terminate lines with <CR><LF> .

Pantry:

S1-3,4 -

PARITY	S1-3	S1-4
ODD	CLOSED	OPEN
EVEN	OPEN	OPEN
MARKING	OPEN	CLOSED
SPACING	CLOSED	CLOSED

Local Echo:

S1-5 OPEN - means that the terminal will cause all keyed characters to also go to the display. Note that these characters will be interleaved with any communications characters arriving simultaneously, e.g.,

```
COMMUNICATIONS: H   O   ER .
KEYBOARD:        ELL  TH  E
DISPLAY RESULT: HELLO THERE.
```

S1-5 CLOSED - means that all keyed characters will be transmitted over the communications line and will not be echoed to the display.

Baud Rate:

S1-6,7,8 -

RATE	S1-6	S1-7	S1-8
110	CLOSED	CLOSED	CLOSED
150	CLOSED	CLOSED	OPEN
300	CLOSED	OPEN	CLOSED
1200	CLOSED	OPEN	OPEN
2400	OPEN	CLOSED	CLOSED
4800	OPEN	CLOSED	OPEN
9600	OPEN	OPEN	CLOSED
19200	OPEN	OPEN	OPEN

NOTE: 110 Baud has 2 stop bits, all others have 1 stop bit.

- -

DIP SWITCH S2

Key Click:

S2-1 OPEN - means that all key strokes will produce a short tone burst ($\overline{\text{ALARM}}$ signal).

S2-1 CLOSED - means that no such burst will occur.

Parity Error Indicate:

S2-2 OPEN - means that a parity error on received data that will cause the display of a <DEL> (█) character in place of the erroneous character.

S2-2 CLOSED - means that the character itself is displayed regardless of error.

Screen Refresh Rate:

S2-3 OPEN - 50 Hz operation.

S2-3 CLOSED - 60 Hz operation.

This switch is only checked on power-up.

Screen Format:

S2-4 OPEN - means 16 lines of 64 characters.

S2-4 CLOSED - means 24 lines of 80 characters.

This switch is only checked at power-up in order to initialize the CRT 5037 VTAC accordingly. This option was implemented to allow terminal flexibility at a future date. However, note that each format requires a different crystal to generate the corresponding video dot clock.

On-Line:

S2-5 OPEN - means that the terminal is ON-LINE.

S2-5 CLOSED - means that the terminal is OFF-LINE.

This switch is wired in parallel to the ON-LINE signal from the keyboard and is ordinarily left open to allow the keyboard to dictate ON- or OFF-LINE status.

Printer Ready Polarity:

S2-6 OPEN - means that Printer Ready (J6-20 or J1-13) must be MARKING (-) to allow data to go to the printer.

S2-6 CLOSED - means that it must be SPACING (+) to allow data to go to the printer.

This switch is set to conform to printer requirements.

Vertical Sync Polarity:

S2-7 OPEN - means that the VSYNC signal to the video is a negative-going pulse.

S2-7 CLOSED - gives a positive-going pulse.

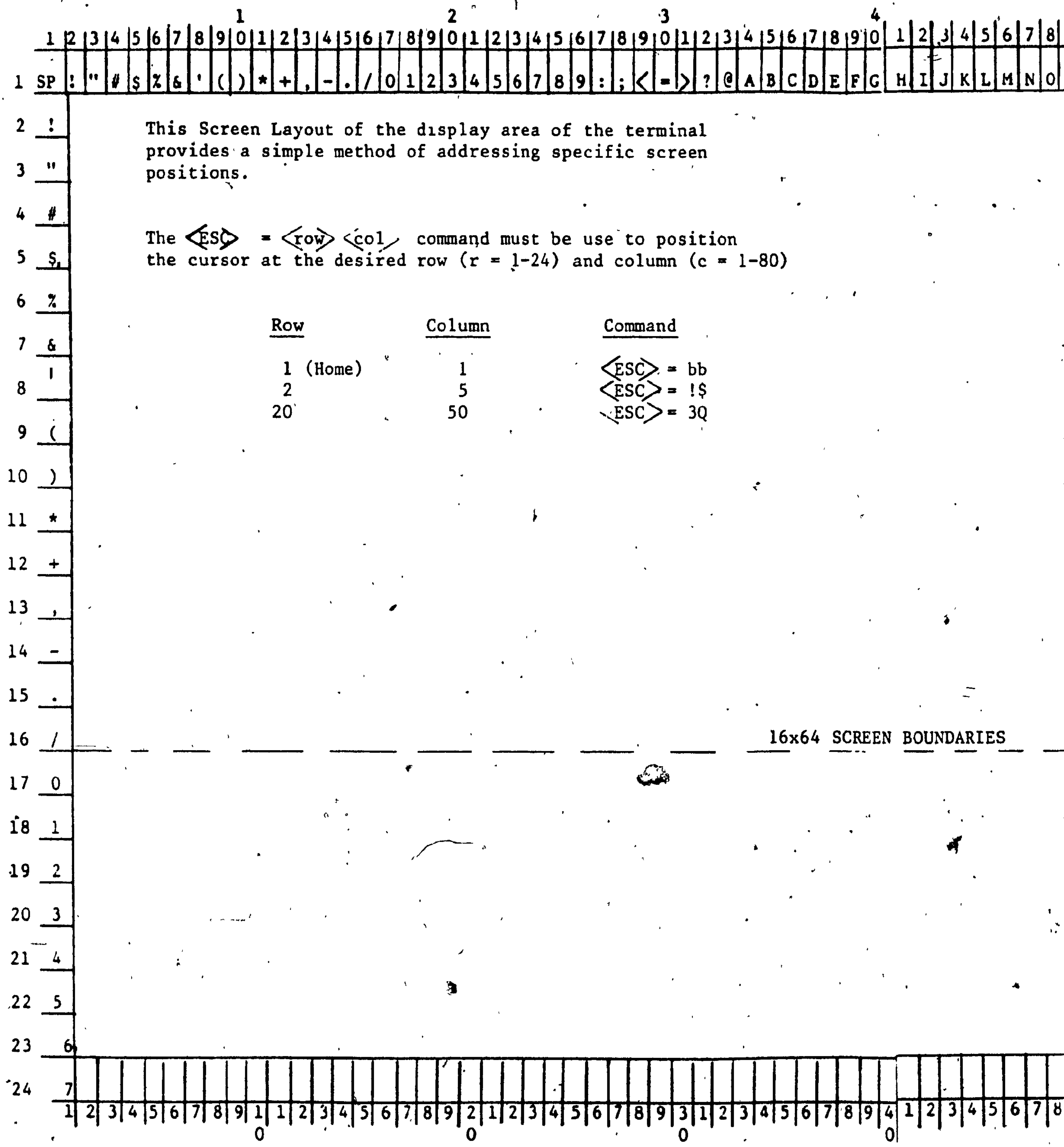
S2-8 - this switch is not used.

APPENDIX F

REFERENCE TABLE TO ABSOLUTE CURSOR POSITIONING SEQUENCE

X or Y POSITION	MODULO NUMBER	ASCII CODE	HEX CODE	X or Y POSITION	MODULO NUMBER	ASCII CODE	HEX CODE
1	0	SPACE	20	41	40	H	48
2	1	!	21	42	41	I	49
3	2	"	22	43	42	J	4A
4	3	#	23	44	43	K	4B
5	4	$	24	45	44	L	4C
6	5	%	25	46	45	M	4D
7	6	&	26	47	46	N	4E
8	7	'	27	48	47	O	4F
9	8	(	28	49	48	P	50
10	9	)	29	50	49	Q	51
11	10	*	2A	51	50	R	52
12	11	+	2B	52	51	S	53
13	12	,	2C	53	52	T	54
14	13	-	2D	54	53	U	55
15	14	.	2E	55	54	V	56
16	15	/	2F	56	55	W	57
17	16	0	30	57	56	X	58
18	17	1	31	58	57	Y	59
19	18	2	32	59	58	Z	5A
20	19	3	33	60	59	[	5B
21	20	4	34	61	60	\	5C
22	21	5	35	62	61	]	5D
23	22	6	36	63	62	^	5E
24	23	7	37	64	63	_	5F
25	24	8	38	65	64	`	60
26	25	9	39	66	65	a	61
27	26	:	3A	67	66	b	62
28	27	;	3B	68	67	c	63
29	28	<	3C	69	68	d	64
30	29	=	3D	70	69	e	65
31	30	>	3E	71	70	f	66
32	31	?	3F	72	71	g	67
33	32	@	40	73	72	h	68
34	33	A	41	74	73	i	69
35	34	B	42	75	74	j	6A
36	35	C	43	76	75	k	6B
37	36	D	44	77	76	l	6C
38	37	E	45	78	77	m	6D
39	38	F	46	79	78	n	6E
40	39	G	47	80	79	o	6F

SCREEN LAYOUT



This Screen Layout of the display area of the terminal provides a simple method of addressing specific screen positions.

The <ESC> = <row> <col> command must be use to position the cursor at the desired row (r = 1-24) and column (c = 1-80)

Row	Column	Command
1 (Home)	1	<ESC> = bb
2	5	<ESC> = !$
20	50	<ESC> = 3Q

Figure F.1 ABSOLUTE CURSOR ADDRESSING CHART

1OF

SCREEN LAYOUT

7	8	9	40	1	2	3	4	5	6	7	8	9	50	1	2	3	4	5	6	7	8	9	60	1	2	3	4	5	6	7	8	9	70	1	2	3	4	5	6	7	8	9	80	
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	–	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	1

2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

:64 SCREEN BOUNDARIES

6 7 8 9 40 1 2 3 4 5 6 7 8 9 50 1 2 3 4 5 6 7 8 9 60 1 2 3 4 5 6 7 8 9 70 1 2 3 4 5 6 7 8 9 80

2 of 2

APPENDIX G

Terminal's Assembly Language Driver Program

```
FILE: TERMINAL:pJUAN     HEWLETT-PACKARD: 8085 Assembler                       Sat, 15 Oct 1983,  3:2

LOCATION OBJECT CODE LINE     SOURCE LINE

                     1 "8085"
                     2
                     3 *               *******************************************************
                     4 *               *                                                     *
                     5 *               * CONTROL PROGRAM  FOR THE "SAVE" - SINGLE BOARD - TERMINAL *
                     6 *               *                                                     *
                     7 *               *         written by: JUAN SALAS                      *
                     8 *               *            date:    15 October,1983 (latest revision)   *
                     9 *               *          revision:  4                               *
                    10 *               *******************************************************
                    11
                    12
                    13 *               --------------------LIST OF EQUATES --------------------
                    14
                    15
                    16 *               8155 - RAM/IO/TIMER
                    17 *               ===================
                    18
    (0020)          19 IOCS8155        EQU       20H                 ;8155 command/status
    (00C0)          20 IOCSINIT        EQU       0C0H                ;initialize to all inputs, start TIMER
                    21
                    22 *               SWITCH BANK CONTROL PORT (input)
                    23 *               --------------------------------
                    24
    (0021)          25 SWITCH1         EQU       21H                 ;PORT A, switch bank S1
    (0080)          26 SW1BASCR        EQU       80H                 ;AUTO SCROLL
    (0040)          27 SW1BALF         EQU       40H                 ;AUTO LINE FEED
    (0030)          28 SW1BPAR         EQU       30H                 ;PARITY (0-space, 1-odd, 2-mark, 3-even)
    (0008)          29 SW1BECHO        EQU       08H                 ;ECHOPLEX mode
    (0007)          30 SW1BBDR         EQU       07H                 ;BAUD RATE
                    31
                    32 *               KEYBOARD PORT (input)
                    33 *               ---------------------
                    34
    (0022)          35 KEYBOARD        EQU       22H                 ;PORT B, KEYBOARD data
                    36
                    37 *               SWITCH BANK CONTROL PORT (input)
                    38 *               --------------------------------
                    39
    (0023)          40 SWITCH2         EQU       23H                 ;PORT C, switch bank 2
    (0020)          41 SW2BPRDY        EQU       20H                 ;PRINTER READY #
    (0010)          42 SW2BCTS         EQU       10H                 ;CLEAR TO SEND #
    (0008)          43 SW2BKCE         EQU       08H                 ;KEY CLICK ENABLE
    (0004)          44 SW2BPERI        EQU       04H                 ;PARITY ERROR INDICATE
    (0002)          45 SW2B50HZ        EQU       02H                 ;50 Hz REFRESH RATE
    (0001)          46 SW2B16L         EQU       01H                 ;16 X 64 FORMAT
    (0024)          47 TIMRLOW         EQU       24H                  LOW order TIMER
    (0025)          48 TIMRHI          EQU       25H                  HIGH order TIMER
                    49
                    50 *               8251A - COMMUNICATIONS CONTROL
                    51 *               ==============================
                    52
    (0030)          53 DR8251A         EQU       30H                 ;communications DATA Register of 8251A
    (0031)          54 CR8251A         EQU       31H                 ;communications mode/COMMAND/STATUS Register of 8
    (0080)          55 CRBSBS          EQU       80H                 ;2 Stop Bit Select
    (0020)          56 CRBEPS          EQU       20H                 ;Even Parity Select
    (0010)          57 CRBPEN          EQU       10H                 ;Parity ENable
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

       <0004>      58 CRBLEN       EQU      04H             ;8 bit data select
       <004A>      59 CRBCON       EQU      4AH             ;constant or mask for mode instr.
       <0040>      60 CRBRST       EQU      40H             ;ReSeT
       <0020>      61 CRBRTS       EQU      20H             ;Request To Send OUT/ serial datapath selector
       <0010>      62 CRBERST      EQU      10H             ;receive Error ReSeT
       <0008>      63 CRBFBRK      EQU      08H             ;Force BReaK
       <0004>      64 CRBRCVEN     EQU      04H             ;ENable ReCeiVe/interrupts
       <0002>      65 CRBDTR       EQU      02H             ;Data Transmit Ready OUT
       <0001>      66 CRBTXEN      EQU      01H             ;ENable Transmit/interrupts
       <0080>      67 CRBDSR       EQU      80H             ;Data Set Ready IN
       <0020>      68 CRBFE        EQU      20H             ;Framing Error/break detect
       <0010>      69 CRBOE        EQU      10H             ;Overrun Error
       <0008>      70 CRBPE        EQU      08H             ;Parity Error
       <0004>      71 CRBTXEMT     EQU      04H             ;Transmit EMpTy flag
       <0002>      72 CRBRXRDY     EQU      02H             ;Receive  ReaDY flag
       <0001>      73 CRBTXRDY     EQU      01H             ;Transmit ReaDY flag
                   74
                   75 *            CRT 5037 VTAC CONTROLLER REGISTERS
                   76 *            ================================
                   77
       <0046>      78 VTACCR6      EQU      46H             ;VTAC Control Register #6
       <0048>      79 VTACCROW     EQU      48H             ;VTAC Cursor ROW (READ) - register #8
       <0049>      80 VTACCCOL     EQU      49H             ;VTAC Cursor COLumn (READ) - register #9
       <004A>      81 VTACRSET     EQU      4AH             ;VTAC RESET command - register #10
       <004C>      82 VTACWCOL     EQU      4CH             ;VTAC Cursor COLumn (WRITE) - register #12
       <004D>      83 VTACWROW     EQU      4DH             ;VATC Cursor ROW (WRITE) - register #13
       <004E>      84 VTACSTRT     EQU      4EH             ;VTAC START command - register #15
       <4000>      85 VTACMEM      EQU      4000H           ;MEMory address of VTAC controller
       <5000>      86 VIDEORAM     EQU      5000H           ;VIDEO RAM DISPLAY BUFFER - wait state access
       <5C90>      87 FREEDRAM     EQU      VIDEORAM+0C90H  ;unused RAM from 0C90H to 0CBFH
       <6000>      88 FASTVRAM     EQU      6000H           ;DISPLAY BUFFER - blanking access
       <0006>      89 XMTQLEN      EQU      6               ;LENgth of TRANSMIT buffer
   <FFFFFFE2>      90 CONBL6       EQU      -30
   <FFFFFFE7>      91 CONBL5       EQU      -25
       <001E>      92 STKSIZE      EQU      30              ;STACK SIZE
                   93
                   94 *            CONTROL CHARACTERS RECOGNIZED BY THE TERMINAL
                   95 *            ==========================================
                   96 *                     ASCII  code           Corresponding control description
                   97 *                     ----------           ------------------------------
       <0007>      98 CONBEL       EQU      07H             ;(BEL) - bell
       <0008>      99 CONBS        EQU      08H             ;(BS)  - back space
       <0009>     100 CONHT        EQU      09H             ;(HT)  - horizontal tab
       <000A>     101 CONLF        EQU      0AH             ;(LF)  - line feed
       <000B>     102 CONVT        EQU      0BH             ;(VT)  - vertical tab
       <000C>     103 CONFF        EQU      0CH             ;(FF)  - feed forward
       <000D>     104 CONCR        EQU      0DH             ;(CR)  - carriage return
       <000E>     105 CONSO        EQU      0EH             ;(SO)  -
       <000F>     106 CONSI        EQU      0FH             ;(SI)  -
       <001A>     107 CONSUB       EQU      1AH             ;(SUB) -
       <001B>     108 CONESC       EQU      1BH             ;(ESC) - escape
       <001E>     109 CONRS        EQU      1EH             ;(RS)  -
       <001F>     110 CONUS        EQU      1FH             ;(US)  -
       <0022>     111 CONQUO       EQU      22H             ;  "   - (quotation)
       <005B>     112 CONLBR       EQU      5BH             ;  [   -
       <0069>     113 CONLI        EQU      69H             ;  i   -
       <0070>     114 CONLP        EQU      70H             ;  p   -
LOCATION OBJECT CODE LINE     SOURCE LINE

              (0071)  115 CONLQ      EQU      71H              ; q  -
              (0072)  116 CONLR      EQU      72H              ; r  -
              (0074)  117 CONLT      EQU      74H              ; t  -
              (0079)  118 CONLY      EQU      79H              ; y  -
              (007C)  119 CONORBAR   EQU      7CH              ; |  -
              (007E)  120 CONTILDE   EQU      7EH              ; )  -
              (007F)  121 CONDEL     EQU      7FH              ;(DEL) - delete
                      122
                      123
                      124 *-----------------------------------------------------------------
                      125
                      126            ORG      0000H
                      127
                      128 *          RESET COMES HERE (MASTER CLEAR)
                      129 *          ============================
                      130
    0000 3121FF       131 SAVE100    LXI      SP,SCRATCH+1FFH
    0003 AF           132            XRA      A
    0004 47           133            MOV      B,A
    0005 48           134            MOV      C,B
    0006 C5           135 INIT100    PUSH     B                ;clear SCRATCHPAD
    0007 3C           136            INR      A
    0008 C20006       137            JNZ      INIT100
    000B 312031       138            LXI      SP,STACKEND-2    ;set up STACK
    000E 3EC0         139            MVI      A,IOCSINIT       ;set up 8155
    0010 D320         140            OUT      IOCS8155
    0012 DB21         141            IN       SWITCH1          ;force BAUDRATE & PARITY first time
    0014 2F           142            CMA
    0015 322004       143            STA      SSWITCH1
    0018 210299       144            LXI      H,INITABLE       ;initialize nonzero SCRATCHPAD values
    001B 112039       145            LXI      D,DISQSPTR
    001E 06FA         146            MVI      B,-INITLEN
    0020 CD0280       147            CALL     MOVER
    0023 DB23         148            IN       SWITCH2          ;initialize VTAC
    0025 E603         149            ANI      SW2B50HZ+SW2B16L
    0027 87           150            ADD      A
    0028 87           151            ADD      A
    0029 87           152            ADD      A
    002A D1           153            POP      D
    002B 5F           154            MOV      E,A
    002C 19           155            DAD      D
    002D D34E         156            OUT      VTACSTRT
    002F D34A         157            OUT      VTACRSET
    0031 C302AF       158            JMP      INIT120          ;continue at 2nd part
                      159
                      160
                      161 *-----------------------------------------------------------------
                      162
                      163 *          8251A TXRDY + RXRDY INTERRUPTS HERE (RST 6.5)
                      164 *          =========================================
                      165
    0034 F5           166 COMMINT    PUSH     PSW
    0035 C30073       167            JMP      COMM100          ;continue at 2nd part
                      168
                      169
                      170 *-----------------------------------------------------------------
                      171
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                      172 *            VERTICAL RETRACE INTERRUPTS HERE (RST 7)
                      173 *            ========================================
                      174
  0038 F5             175 VERTINT      PUSH       PSW
  0039 C301DB         176              JMP        VERT100           ;continue at 2nd part
                      177
                      178
                      179 *---------------------------------------------------------------------------
                      180
                      181 *            KEYBOARD STROBE INTERRUPTS HERE (RST 7.5)
                      182 *            ==========================================
                      183
  003C F5             184 KEYBDINT     PUSH       PSW
  003D E5             185              PUSH       H
  003E D5             186              PUSH       D
  003F C5             187              PUSH       B
  0040 DB22           188              IN         KEYBOARD
  0042 E67F           189              ANI        7FH
  0044 47             190              MOV        B,A
  0045 20             191              RIM
  0046 B7             192              ORA        A
  0047 F2005C         193              JP         KEYBD100          ;OFF-LINE
  004A 3A2007         194              LDA        STATUS1
  004D E680           195              ANI        ST1BKBDL
  004F C2006A         196              JNZ        KEYBD120          :LOCKED - do nothing
  0052 CD024B         197              CALL       TRANSMIT
  0055 DB21           198              IN         SWITCH1
  0057 E608           199              ANI        SW1BECHO
  0059 CA005F         200              JZ         KEYBD110          ;not ECHOPLEX
  005C CD0261         201 KEYBD100     CALL       DISPBUFF          ;send to SCREEN
  005F DB23           202 KEYBD110     IN         SWITCH2
  0061 E608           203              ANI        SW2BKCE
  0063 CA006A         204              JZ         KEYBD120          ;no CLICK
  0066 21203D         205              LXI        H,VERTBELL
  0069 35             206              DCR        M                 ;ding BELL
  006A 3E10           207 KEYBD120     MVI        A,10H
  006C 30             208              SIM                          ;clear KEY interrupt
  006D C1             209 INTREXIT     POP        B
  006E D1             210              POP        D
  006F E1             211              POP        H
  0070 F1             212              POP        PSW
  0071 FB             213              EI
  0072 C9             214              RET
                      215
                      216
                      217 *---------------------------------------------------------------------------
                      218
                      219 *            SECOND PART COMMUNICATIONS INTERRUPT HANDLER
                      220 *            ============================================
                      221
  0073 E5             222 COMM100      PUSH       H
  0074 D5             223              PUSH       D
  0075 C5             224              PUSH       B
  0076 210073         225              LXI        H,COMM100
  0079 E5             226              PUSH       H
  007A DB31           227              IN         CR8251A
  007C E602           228              ANI        CRBRXRDY
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

  007E C8            229                RZ                                ;not received character
  007F DB31          230                IN        CR8251A
  0081 E638          231                ANI       CRBFE+CRBPE+CRBOE
  0083 CA00A0        232                JZ        COMM110                 ;no errors
  0086 47            233                MOV       B,A
  0087 3A2000        234                LDA       SCR8251A
  008A F610          235                ORI       CRBERST
  008C D331          236                OUT       CR8251A
  008E 78            237                MOV       A,B
  008F E620          238                ANI       CRBFE
  0091 C0            239                RNZ       IGNORE_IF_OPEN_LINE
  0092 DB23          240                IN        SWITCH2
  0094 E604          241                ANI       SW2BPERI
  0096 CA00A0        242                JZ        COMM110                 ;don't indicate error
  0099 DB30          243                IN        DR8251A
  009B 067F          244                MVI       B,CONDEL
  009D C300D0        245                JMP       COMM120
                     246
  00A0 DB30          247 COMM110        IN        DR8251A
  00A2 E67F          248                ANI       7FH
  00A4 47            249                MOV       B,A
  00A5 212008        250                LXI       H,STATUS2
  00A8 7E            251                MOV       A,M
  00A9 E603          252                ANI       ST2BFETM
  00AB CA00C2        253                JZ        COMM112
  00AE 35            254                DCR       M
  00AF CA00D0        255                JZ        COMM120                 ;store control character regardless
  00B2 35            256                DCR       M
  00B3 78            257                MOV       A,B
  00B4 FE5A          258                CPI       'Z'
  00B6 CA00BE        259                JZ        COMM111
  00B9 FE47          260                CPI       'G'
  00BB C200CA        261                JNZ       COMM118
  00BE 34            262 COMM111        INR       M
  00BF C300D0        263                JMP       COMM120
  00C2 78            264 COMM112        MOV       A,B
  00C3 FE1B          265                CPI       CONESC
  00C5 C200CA        266                JNZ       COMM118
  00C8 34            267                INR       M
  00C9 34            268                INR       M
  00CA 78            269 COMM118        MOV       A,B
  00CB B7            270                ORA       A
  00CC C8            271                RZ                                ;discard (NUL)
  00CD FE7F          272                CPI       7FH
  00CF C8            273                RZ                                ;discard (DEL)
                     274
  00D0 20            275 COMM120        RIM
  00D1 B7            276                ORA       A
  00D2 FA0261        277                JM        DISPBUFF                ;displat if ON-LINE
  00D5 E1            278                POP       H
                     279
  00D6 DB31          280 COMM130        IN        CR8251A
  00D8 E601          281                ANI       CRBTXRDY
  00DA CA006D        282                JZ        INTREXIT                ;exit if not transmit
  00DD 210169        283                LXI       H,COMM140
  00E0 E5            284                PUSH      H
  00E1 3A2007        285                LDA       STATUS1
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    00E4 E610         286              ANI     ST1BLPTA          ;local PRINT active?
    00E6 CA0175       287              JZ      COMM139           ;no...do regular schtick with X-BUFF
    00E9 DB23         288              IN      SWITCH2
    00EB E620         289              ANI     SW2BPRDY          ;PRINTER ready?
    00ED C0           290              RNZ                       ;no...clear INTR and wait
    00EE 212005       291 COMM130A     LXI     H,LPTCOLMN
    00F1 3A2003       292 COMM130B     LDA     MAXCOL
    00F4 3C           293              INR     A
    00F5 BE           294              CMP     M                 ;END-OF-LINE?
    00F6 FA0142       295              JM      COMM134           ;past it
    00F9 CA013C       296              JZ      COMM132           ;at it
    00FC E5           297 COMM131      PUSH    H
    00FD 2C           298              INR     L
    00FE 46           299              MOV     B,M
    00FF CD05F5       300              CALL    BKGD492
    0102 210195       301              LXI     H,COMM180
    0105 E3           302              XTHL
    0106 7E           303              MOV     A,M
    0107 3A2005       304              LDA     LPTCOLMN
    010A B3           305              ORA     E
    010B 5F           306              MOV     E,A
    010C 1A           307              LDAX    D
    010D 47           308              MOV     B,A
    010E 34           309              INR     M
    010F B7           310              ORA     A
    0110 F0           311              RP
    0111 EB           312              XCHG
    0112 212007       313              LXI     H,STATUS1
    0115 7E           314              MOV     A,M
    0116 E604         315              ANI     ST1B4MAT
    0118 CA0139       316              JZ      COMM131J
    011B 7E           317              MOV     A,M
    011C E603         318              ANI     ST1B4MXX
    011E C20126       319              JNZ     COMM131B
    0121 061B         320              MVI     B,CONESC
    0123 C3012C       321              JMP     COMM131D
    0126 3D           322 COMM131B     DCR     A
    0127 C20130       323              JNZ     COMM131F
    012A 0647         324              MVI     B,'G'
    012C 34           325 COMM131D     INR     M
    012D EB           326              XCHG
    012E 35           327              DCR     M
    012F C9           328              RET
    0130 7E           329 COMM131F     MOV     A,M
    0131 E6FC         330              ANI     .NT.ST1B4MXX
    0133 77           331              MOV     M,A
    0134 78           332              MOV     A,B
    0135 EEA0         333              XRI     0A0H
    0137 47           334              MOV     B,A
    0138 C9           335              RET
    0139 0620         336 COMM131J     MVI     B,' '             ;ATTRIBUTES become blanks
    013B C9           337              RET
    013C 060D         338 COMM132      MVI     B,CONCR
    013E 34           339              INR     M
    013F C30195       340              JMP     COMM180
    0142 3C           341 COMM134      INR     A
    0143 BE           342              CMP     M
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    0144 C20154      343             JNZ       COMM135
    0147 34          344             INR       M
    0148 DB21        345             IN        SWITCH1
    014A E640        346             ANI       SW1BALF
    014C C20154      347             JNZ       COMM135
    014F 060A        348 COMM134B    MVI       B,CONLF
    0151 C30195      349             JMP       COMM180
    0154 3A2007      350 COMM135     LDA       STATUS1
    0157 E608        351             ANI       ST1BXMIT
    0159 C201D0      352             JNZ       COMM192
    015C 3A2002      353             LDA       MAXROW
    015F 2C          354             INR       L
    0160 BE          355             CMP       M
    0161 C8          356             RZ
    0162 34          357             INR       M
    0163 2D          358             DCR       L
    0164 3600        359             MVI       M,0
    0166 C300FC      360             JMP       COMM131
                     361
    0169 212000      362 COMM140     LXI       H,SCR8251A
    016C 7E          363             MOV       A,M
    016D E6FE        364             ANI       .NT.CRBTXEN        ;TRANSMITTER OFF
                     365
    016F 77          366 COMM150     MOV       M,A
    0170 D331        367             OUT       CR8251A
    0172 C3006D      368             JMP       INTREXIT
                     369
    0175 DB23        370 COMM139     IN        SWITCH2
    0177 E610        371             ANI       SW2BCTS
    0179 C0          372             RNZ
                     373
    017A 3A2007      374 COMM160     LDA       STATUS1
    017D E608        375             ANI       ST1BXMIT
    017F C201A0      376             JNZ       COMM190
    0182 21203C      377 COMM165     LXI       H,XMTQSPTR
    0185 4E          378             MOV       C,M
    0186 2D          379             DCR       L
    0187 7E          380             MOV       A,M
    0188 3C          381             INR       A
    0189 FE39        382             CPI       XMTQEND
    018B C20190      383             JNZ       COMM170
    018E 3E33        384             MVI       A,XMTQUEUE         ;wrap around of queue
                     385
    0190 B9          386 COMM170     CMP       C
    0191 C8          387             RZ                           ;0 EMPTY-CLEAR interrupt
    0192 77          388             MOV       M,A
    0193 6F          389             MOV       L,A
    0194 46          390             MOV       B,M
                     391
    0195 3A2004      392 COMM180     LDA       SSWITCH1
    0198 E620        393             ANI       20H
    019A 87          394             ADD       A
    019B 87          395             ADD       A
    019C B0          396             ORA       B
    019D D330        397             OUT       DR8251A
    019F C1          398             POP       B
    01A0 212000      399             LXI       H,SCR8251A
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

   01A3 7E           400                MOV     A,M
   01A4 F601         401                ORI     CR0TXEN
   01A6 C3016F       402                JMP     COMM150
                     403
   01A9 212005       404 COMM190        LXI     H,LPTCOLMN
   01AC DB49         405                IN      VTACCCOL
   01AE BE           406                CMP     M
   01AF F200F1       407                JP      COMM130D
   01B2 2C           408                INR     L
   01B3 3A2014       409                LDA     CURSROW
   01B6 3D           410                DCR     A
   01B7 BE           411                CMP     M
   01B8 F200EE       412                JP      COMM130A
   01BB 2C           413                INR     L
   01BC 7E           414                MOV     A,M
   01BD E603         415                ANI     ST1B4MXX
   01BF CA013C       416                JZ      COMM132
   01C2 7E           417                MOV     A,M
   01C3 E6F7         418                ANI     .NOT.ST1BXMIT
   01C5 77           419                MOV     M,A
   01C6 DB21         420                IN      SWITCH1
   01C8 E640         421                ANI     SW1BALF
   01CA CA014F       422                JZ      COMM134B
   01CD C30182       423                JMP     COMM165
   01D0 3600         424 COMM192        MVI     M,0
   01D2 2C           425                INR     L
   01D3 34           426                INR     M
   01D4 2D           427                DCR     L
   01D5 C300FC       428                JMP     COMM131
                     429
                     430
                     431 *----------------------------------------------------------------------
                     432
                     433 *              SECOND PART VERTICAL RETRACE INTERRUPT HANDLER
                     434 *              ==============================================
                     435
   01D8 E5           436 VERT100        PUSH    H
   01D9 D5           437                PUSH    D
   01DA C5           438                PUSH    B
   01DB 2100D6       439                LXI     H,COMM130
   01DE E5           440                PUSH    H
   01DF 2101FE       441                LXI     H,VERT110
   01E2 E5           442                PUSH    H
   01E3 21203D       443                LXI     H,VERTBELL          ;count BELL timer
   01E6 34           444                INR     M
   01E7 3E00         445                MVI     A,0
   01E9 47           446                MOV     B,A
   01EA C8           447                RZ
   01EB F8           448                RM
   01EC 70           449                MOV     M,B
   01ED 0680         450                MVI     B,80H
   01EF 20           451                RIM
   01F0 E604         452                ANI     04H
   01F2 C8           453                RZ
   01F3 2E00         454                MVI     L,SCR8251A
   01F5 7E           455                MOV     A,M
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

  01F8 D331          457              OUT     CRB251A
  01FA 77            458              MOV     M,A
  01FB 3E19          459              MVI     A,19H
  01FD C9            460              RET
  01FE F640          461 VERT110      ORI     40H
  0200 B0            462              ORA     B
  0201 30            463              SIM
  0202 DB31          464              IN      CRB251A
  0204 2F            465              CMA
  0205 E605          466              ANI     CRBTXRDY+CRBTXEMT
  0207 C0            467              RNZ
  0208 2E00          468              MVI     L,SCRB251A
  020A EB            469              XCHG
  020B 212007        470              LXI     H,STATUS1
  020E 7E            471              MOV     A,M
  020F E620          472              ANI     ST1BLPTR
  0211 7E            473              MOV     A,M
  0212 CA0225        474              JZ      VERT120
  0215 E6DF          475              ANI     .NT.ST1BLPTR
  0217 F610          476              ORI     ST1BLPTA
  0219 77            477              MOV     M,A
  021A EB            478              XCHG
  021B 7E            479              MOV     A,M
  021C E6DF          480              ANI     .NT.CRBRTS
  021E F601          481              ORI     CRBTXEN
  0220 77            482              MOV     M,A
  0221 D331          483              OUT     CRB251A
  0223 EB            484              XCHG
  0224 7E            485              MOV     A,M
  0225 E610          486 VERT120      ANI     ST1BLPTA
  0227 C8            487              RZ
  0228 2E02          488              MVI     L,MAXROW
  022A 3A2006        489              LDA     LPTROW
  022D BE            490              CMP     M
  022E C0            491              RNZ
  022F 2C            492              INR     L
  0230 3A2005        493              LDA     LPTCOLMN
  0233 D602          494              SUI     2
  0235 BE            495              CMP     M
  0236 F8            496              RM
  0237 2E07          497              MVI     L,STATUS1
  0239 7E            498              MOV     A,M
  023A E6EF          499              ANI     .NT.ST1BLPTA
  023C 77            500              MOV     M,A
  023D EB            501              XCHG
  023E 7E            502              MOV     A,M
  023F F620          503              ORI     CRBRTS
  0241 77            504              MOV     M,A
  0242 D331          505              OUT     CRB251A
  0244 C9            506              RET
                     507
                     508
                     509 *-------------------------------------------------------------
                     510
                     511 *            QUEUE CHARACTER TO TRANSMIT
                     512 *            ===========================
                     513
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

   0245 F3              514 TRANSMT3    DI
   0246 E63F            515 TRANSMT0    ANI        3FH
   0248 C620            516 TRANSMT1    ADI        32
   024A 47              517 TRANSMT2    MOV        B,A
   024B 21203B          518 TRANSMIT    LXI        H,XMTQFPTR
   024E 7E              519             MOV        A,M
   024F 2C              520             INR        L
   0250 BE              521             CMP        M
   0251 C8              522             RZ
   0252 6E              523             MOV        L,M
   0253 70              524             MOV        M,B
   0254 7D              525             MOV        A,L
   0255 3C              526             INR        A
   0256 FE39            527             CPI        XMTQEND
   0258 C2025D          528             JNZ        TRANS100
   025B 3E33            529             MVI        A,XMTQUEUE          ;wrap around end of queue
   025D 32203C          530 TRANS100    STA        XMTQSPTR
   0260 C9              531             RET
                        532
                        533
                        534 *---------------------------------------------------------------------
                        535
                        536 *           QUEUE CHARACTER TO DISPLAY
                        537 *           ========================
                        538
   0261 21203A          539 DISPBUFF    LXI        H,DISQFPTR
   0264 7E              540             MOV        A,M
   0265 2D              541             DCR        L
   0266 BE              542             CMP        M
   0267 C8              543             RZ                             ;no room in queue
   0268 6E              544             MOV        L,M
   0269 70              545             MOV        M,B
   026A 7D              546             MOV        A,L
   026B 3C              547             INR        A
   026C C20271          548             JNZ        DISPB100
   026F 3E3E            549             MVI        A,DISQUEUE          ;wrap around end of queue
                        550
   0271 322039          551 DISPB100    STA        DISQSPTR
   0274 C9              552             RET
                        553
                        554
   0275 3E80            555 TABSINIT    MVI        A,80H
   0277 21200A          556 TABSCLR     LXI        H,TABTABLE
   027A 11200B          557             LXI        D,TABTABLE+1
   027D 06F7            558             MVI        B,-9
   027F 77              559             MOV        M,A
                        560
                        561 *           MOVE DATA
                        562 *           =========
                        563
   0280 7E              564 MOVER       MOV        A,M
   0281 12              565 MOVER1      STAX       D
   0282 23              566             INX        H
   0283 13              567             INX        D
   0284 04              568             INR        B
   0285 C20280          569             JNZ        MOVER
   0288 C9              570             RET
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                      571
                      572
                      573 *-----------------------------------------------------------------
                      574
                      575 *              BAUD RATE DIVISER TABLE (ASSUMES 6.144 MHz 8085A CRYSTAL)
                      576 *              ==================================================
                      577
 0289 46D1            578 BDRTABL        DW      46D1H           ;  110 BAUD
 028B 4500            579                DW      4500H           ;  150 BAUD
 028D 4200            580                DW      4200H           ;  300 BAUD
 028F 40A0            581                DW      40A0H           ; 1200 BAUD
 0291 4050            582                DW      4050H           ; 2400 BAUD
 0293 4028            583                DW      4028H           ; 4800 BAUD
 0295 4014            584                DW      4014H           ; 9600 BAUD
 0297 400A            585                DW      400AH           ;19200 BAUD
                      586
                      587
                      588 *              SCRATCHPAD INITIALIZATION TABLE
                      589 *              ===============================
                      590
 0299 40              591 INITABLE       DB      DISQUEUE+2
 029A 3D              592                DB      DISQUEUE-1
 029B 32              593                DB      XMTQUEUE-1
 029C 33              594                DB      XMTQUEUE
 029D E2              595                DB      CONBL6
 029E 1A              596                DB      CONSUB
                      597
          (0006)      598 INITLEN        EQU     $-INITABLE
                      599
                      600
                      601 *              CRT 5037 CONTROLLER DATA TABLE
                      602 *              ==============================
                      603
 029F 65              604 VTACTABL       DB      102-1           ;24 X 80, 60Hz, 102 characterS total
 02A0 4A              605                DB      4AH             ;noninterlace scan, 2 delay/9 drive horizontal
 02A1 5D              606                DB      5DH             ;12 scans/datarow, 80 characters/datarow
 02A2 17              607                DB      24-1            ;no skew, 24 datarows/frame
 02A3 1B              608                DB      (310-256)/2     ;310 scan lines/frame, horiz. freq.=18.6 KHz
 02A4 13              609                DB      19              ;vert. start delay in scan lines
 02A5 17              610                DB      24-1            ;last data row displayed initial
 02A6 00              611                DB      0
                      612
 02A7 51              613                DB      82-1            ;16 X 64, 60Hz, 82 characters total
 02A8 37              614                DB      37H             ;noninterlace scan, 6 delay/6 drive horizontal
 02A9 6B              615                DB      6BH             ;14 scans/datarow, 64 characters/datarow
 02AA 0F              616                DB      16-1            ;no skew, 16 datarows/frame
 02AB 02              617                DB      (260-256)/2     ;260 scan lines/frame, horiz. freq.=15.6 KHz
 02AC 19              618                DB      25              ;vertical start delay in scan lines
 02AD 0F              619                DB      16-1            ;last data row displayed initial
 02AE 00              620                DB      0
                      621
                      622
                      623 *-----------------------------------------------------------------
                      624
                      625 *              SECOND PART INITIALIZATION
                      626 *              ==========================
                      627
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    02AF 114000       628 INIT120     LXI      D,VTACMEM
    02B2 06F9         629             MVI      B,-7
    02B4 CD0280       630             CALL     MOVER
    02B7 D34E         631             OUT      VTACSTRT
    02B9 CD0275       632             CALL     TABSINIT            ;preset TABS to colomns 0,8,16...
    02BC DB23         633             IN       SWITCH2
    02BE 213F0F       634             LXI      H,3F0FH             ;16 X 64 format
    02C1 E601         635             ANI      SW2B16L
    02C3 C202C9       636             JNZ      INIT130
    02C6 214F17       637             LXI      H,4F17H             ;24 X 80 format
                      638
    02C9 222002       639 INIT130     SHLD     MAXROW              ;preset row/colomn maxima
    02CC 7D           640             MOV      A,L
    02CD 322001       641             STA      LDROW
    02D0 3E92         642             MVI      A,92H
    02D2 D331         643             OUT      CR8251A             ;guarantee command mode
                      644
                      645 *-----------------------------------------------------------------------------------
                      646
                      647 *           BACKGORUND PROCESS... INTERPRET CHARACTERS AND ACT UPON
                      648 *           ========================================================
                      649
                      650 *           ALL SEQUENCES INVOLVING MEMORY OR I/O SHARED WITH INTE-
                      651 *           RRUPT (FOREGROUND) TASKS MUST INTERLOCK.
                      652
    02D4 3A2009       653 BKGD100     LDA      BKGMODE
    02D7 FE07         654             CPI      7                   ;don't look at switches in maintenace mode
    02D9 CA0320       655             JZ       BKGD150
    02DC DB21         656             IN       SWITCH1
    02DE 212004       657             LXI      H,SSWITCH1
    02E1 AE           658             XRA      M
    02E2 E637         659             ANI      SW1BBDR+SW1BPAR
    02E4 CA0320       660             JZ       BKGD150
    02E7 DB21         661             IN       SWITCH1             ;BAUD RATE or PARITY change
    02E9 77           662 BKGD105     MOV      M,A
    02EA E607         663             ANI      SW1BBDR
    02EC 17           664             RAL
    02ED 4F           665             MOV      C,A
    02EE 7E           666             MOV      A,M
    02EF E630         667             ANI      SW1BPAR
    02F1 47           668             MOV      B,A
    02F2 0F           669             RRC
    02F3 0F           670             RRC
    02F4 2F           671             CMA
    02F5 E604         672             ANI      04H
    02F7 B0           673             ORA      B
    02F8 47           674             MOV      B,A
    02F9 212000       675             LXI      H,SCR8251A
    02FC F3           676             DI
    02FD 7E           677             MOV      A,M
    02FE F640         678             ORI      CRDRST
    0300 D331         679             OUT      CR8251A             ;access mode register
    0302 79           680             MOV      A,C
    0303 B7           681             ORA      A
    0304 3E4A         682             MVI      A,4AH
    0306 [illegible]  683             JNZ      BKGD120
    0309 [illegible]  684             ORI      [illegible]         ;110 BAUD
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

   030B B0          685 BKGD120     ORA      B
   030C D331        686             OUT      CRB251A           ;new command mode
   030E FB          687             EI
   030F 210289      688             LXI      H,BDRTABL
   0312 0600        689             MVI      B,0
   0314 09          690             DAD      B
   0315 7E          691             MOV      A,M
   0316 D324        692             OUT      TIMRLOW
   0318 23          693             INX      H
   0319 7E          694             MOV      A,M
   031A D325        695             OUT      TIMRHI
   031C 3EC0        696             MVI      A,IOCSINIT
   031E D320        697             OUT      IOCS8155
   0320 F3          698 BKGD150     DI                         ;get a character
   0321 212039      699             LXI      H,DISQSPTR
   0324 46          700             MOV      B,M
   0325 23          701             INX      H
   0326 7E          702             MOV      A,M
   0327 3C          703             INR      A
   0328 C2032D      704             JNZ      BKGD160
   032B 3E3E        705             MVI      A,DISQUEUE        ;wrap around end of queue
   032D B8          706 BKGD160     CMP      B
   032E CA0371      707             JZ       BKGD165           ;no characters...idle
   0331 77          708             MOV      M,A
   0332 6F          709             MOV      L,A
   0333 46          710             MOV      B,M
   0334 FB          711             EI
   0335 110320      712             LXI      D,BKGD150
   0338 D5          713             PUSH     D
   0339 2E19        714             MVI      L,BKGMODE
   033B 7E          715             MOV      A,M
   033C 4F          716             MOV      C,A
   033D 3600        717             MVI      M,0
   033F D608        718             SUI      8
   0341 CA0462      719             JZ       BKGD167
   0344 FEFC        720             CPI      4-8
   0346 78          721             MOV      A,B
   0347 CA0460      722             JZ       BKGD166
   034A FE20        723             CPI      ' '
   034C 21065C      724             LXI      H,BRTBLM0-2
   034F FA037A      725             JM       BKGD210
   0352 79          726             MOV      A,C
   0353 210357      727             LXI      H,BRTBLX0-2
   0356 C3037C      728             JMP      BKGD211
                    729
   0359 00          730 BRTBLX0     DB       0
   035A 0464        731             DW       BKGD170
   035C 01          732             DB       1
   035D 0376        733             DW       BKGD200
   035F 00          734             DB       0
   0360 039D        735             DW       BKGD240
   0362 00          736             DB       0
   0363 038D        737             DW       BKGD230
   0365 04          738             DB       4
   0366 0460        739             DW       BKGD166
   0368 05          740             DB       5
   0369 06A0        741             DW       BKGD560
```

```
LOCATION OBJECT CODE LINE      SOURCE LINE

    036B 06             742                  DB      6
    036C 06E7           743                  DW      BKGD565
    036E 07             744                  DB      7
    036F 06F1           745                  DW      BKGD570.
                        746
    0371 FB             747 BKGD165          EI
    0372 76             748                  HLT
    0373 C302D4         749                  JMP     BKGD100
                        750
    0376 78             751 BKGD200          MOV     A,B             ;MODE 1... (ESC) prefixed decode
    0377 210602         752                  LXI     H,BRTBLM1-2
    037A 0600           753 BKGD210          MVI     B,0
    037C 23             754 BKGD211          INX     H
    037D 23             755                  INX     H
    037E BE             756                  CMP     M
    037F F8             757                  RM                      ; no match found
    0380 23             758                  INX     H
    0381 C2037C         759                  JNZ     BKGD211
    0384 5E             760                  MOV     E,M
    0385 23             761                  INX     H
    0386 66             762                  MOV     H,M
    0387 6B             763                  MOV     L,E
    0388 AF             764                  XRA     A
    0389 5F             765                  MOV     E,A
    038A 57             766                  MOV     D,A
    038B 4F             767                  MOV     C,A
    038C E9             768                  PCHL                    ;vectro off to selected routine
                        769
    038D 78             770 BKGD230          MOV     A,B             ;MODE 3... (ESC) = COLUMN adress
    038E D620           771                  SUI     20H
    0390 47             772                  MOV     B,A
    0391 3A2003         773                  LDA     MAXCOL
    0394 B8             774                  CMP     B
    0395 DA0399         775                  JC      BKGD235
    0398 78             776                  MOV     A,B
    0399 D34C           777 BKGD235          OUT     VTACWCOL
    039B AF             778                  XRA     A
    039C C9             779                  RET
                        780
    039D 3E03           781 BKGD240          MVI     A,3             ;MODE 2... (ESC) = ROW address
    039F 322009         782                  STA     BKGMODE         ;chain to COLUMN address
    03A2 78             783                  MOV     A,B
    03A3 D620           784                  SUI     20H
    03A5 47             785 BKGD242          MOV     B,A
    03A6 3A2002         786                  LDA     MAXROW
    03A9 4F             787                  MOV     C,A
    03AA 78             788                  MOV     A,B
    03AB B9             789                  CMP     C
    03AC FA03B0         790                  JM      BKGD244
    03AF 41             791                  MOV     B,C             ;overrange...force bottom of screen
    03B0 CD05E7         792 BKGD244          CALL    BKGD490
    03B3 D34D           793 BKGD245          OUT     VTACWROW
    03B5 78             794                  MOV     A,B
    03B6 322014         795                  STA     CURSROW
    03B9 C9             796                  RET
                        797
    03BA CD03C7         798 BKGD250          CALL    BKGD270         ;SET tabstop
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    03BD B6            799              ORA       M
    03BE 77            800              MOV       M,A
    03BF C9            801              RET
                       802
    03C0 CD03C7        803 BKGD260      CALL      BKGD270            ;CLEAR tabstop
    03C3 2F            804              CMA
    03C4 A6            805              ANA       M
    03C5 77            806              MOV       M,A
    03C6 C9            807              RET
                       808
    03C7 21200A        809 BKGD270      LXI       H,TABTABLE         ;find TABSTOP
    03CA DB49          810              IN        VTACCCOL
    03CC 0F            811              RRC
    03CD 0F            812              RRC
    03CE 0F            813              RRC
    03CF E60F          814              ANI       0FH
    03D1 5F            815              MOV       E,A
    03D2 19            816              DAD       D
    03D3 DB49          817              IN        VTACCCOL
    03D5 E607          818              ANI       07H
    03D7 47            819              MOV       B,A
    03D8 3E80          820              MVI       A,80H
    03DA C8            821 BKGD275      RZ
    03DB 0F            822              RRC
    03DC 05            823              DCR       B
    03DD C303DA        824              JMP       BKGD275
                       825
    03E0 CD03F2        826 BKGD280      CALL      BKGD285            ;DELETE character
    03E3 DB49          827              IN        VTACCCOL
    03E5 B3            828              ORA       E
    03E6 5F            829              MOV       E,A
    03E7 3C            830              INR       A
    03E8 6F            831              MOV       L,A
    03E9 78            832              MOV       A,B
    03EA B7            833              ORA       A
    03EB C40280        834              CNZ       MOVER
    03EE 3E20          835              MVI       A,20H
    03F0 12            836              STAX      D
    03F1 C9            837              RET
                       838
    03F2 DB49          839 BKGD285      IN        VTACCCOL
    03F4 212103        840 BKGD286      LXI       H,MAXCOL
    03F7 96            841              SUB       M
    03F8 47            842              MOV       B,A
    03F9 DB48          843              IN        VTACCROW
    03FB CD05F8        844              CALL      BKGD495
    03FE 62            845              MOV       H,D
    03FF C9            846              RET
                       847
    0400 CD03F2        848 BKGD290      CALL      BKGD285            ;INSERT character
    0403 3A2103        849              LDA       MAXCOL
    0406 B3            850              ORA       E
    0407 5F            851              MOV       E,A
    0408 3D            852              DCR       A
    0409 6F            853              MOV       L,A
    040A 78            854              MOV       A,B
    040B B7            855              ORA       A
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

  040C CA0417      856            JZ        BKGD297
  040F 7E          857 BKGD295    MOV       A,M
  0410 12          858            STAX      D
  0411 2B          859            DCX       H
  0412 1B          860            DCX       D
  0413 04          861            INR       B
  0414 C2040F      862            JNZ       BKGD295
  0417 23          863 BKGD297    INX       H
  0418 3620        864            MVI       M,20H
  041A C9          865            RET
                   866
  041B 0640        867 BKGD300    MVI       B,ST1BINSM        ;INSERT mode ON
  041D 212007      868 BKGD305    LXI       H,STATUS1         ;INSERT mode OFF
  0420 7E          869            MOV       A,M
  0421 E6BF        870            ANI       .NT.ST1BINSM
  0423 B0          871            ORA       B
  0424 77          872            MOV       M,A
  0425 C9          873            RET
                   874
  0426 CD03F2      875 BKGD310    CALL      BKGD285           ;ERASE FROM CURSOR TO END OF LINE
  0429 3A2003      876 BKGD311    LDA       MAXCOL
  042C B3          877            ORA       E
  042D 6F          878            MOV       L,A
  042E 3620        879            MVI       M,20H
  0430 3D          880            DCR       A
  0431 5F          881            MOV       E,A
  0432 C3040F      882            JMP       BKGD295
                   883
  0435 D34C        884 BKGD320    OUT       VTACWCOL          ;HOME CURSOR to top left
  0437 322014      885            STA       CURSROW
  043A C303A5      886            JMP       BKGD242
  043D CD0435      887 BKGD330    CALL      BKGD320           ;CLEAR SCREEN
  0440 CD0426      888 BKGD335    CALL      BKGD310           ;ERASE FROM CURSOR TO END OF SCREEN
  0443 DB23        889            IN        SWITCH2
  0445 E601        890            ANI       SW2B16L
  0447 0E05        891            MVI       C,5
  0449 CA044E      892            JZ        BKGD336
  044C 0E04        893            MVI       C,4
  044E 3A2014      894 BKGD336    LDA       CURSROW
  0451 47          895            MOV       B,A
  0452 04          896 BKGD337    INR       B
  0453 3A2002      897            LDA       MAXROW
  0456 B8          898            CMP       B
  0457 F8          899            RM
  0458 C5          900            PUSH      B
  0459 CD04D5      901            CALL      BKGD380
  045C C1          902            POP       B
  045D C30452      903            JMP       BKGD337
                   904
  0460 3EA0        905 BKGD166    MVI       A,0A0H
  0462 A8          906 BKGD167    XRA       B
  0463 47          907 BKGD180    MOV       B,A
  0464 C5          908 BKGD170    PUSH      B
  0465 3A2007      909            LDA       STATUS1
  0468 E640        910            ANI       ST1BINSM
  046A C40400      911            CNZ       BKGD290
  046D C1          912            POP       B
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

   046E DB48        913           IN      VTACCROW
   0470 CD05F8      914           CALL    BKGD495
   0473 DB49        915           IN      VTACCCOL
   0475 B3          916           ORA     E
   0476 5F          917           MOV     E,A
   0477 78          918           MOV     A,B
   0478 12          919           STAX    D
   0479 DB49        920 BKGD340   IN      VTACCCOL        ;CURSOR RIGHT
   047B 3C          921           INR     A
   047C 47          922           MOV     B,A
   047D 3A2003      923           LDA     MAXCOL
   0480 B8          924           CMP     B
   0481 78          925           MOV     A,B
   0482 D34C        926           OUT     VTACWCOL
   0484 F0          927           RP
   0485 AF          928 BKGD343   XRA     A               ;CURSOR TO LEFT MARGIN & DOWN 1 LINE
   0486 D34C        929 BKGD344   OUT     VTACWCOL
   0488 212014      930 BKGD345   LXI     H,CURSROW       ;CURSOR DOWN
   048B 34          931           INR     M
   048C 3A2002      932           LDA     MAXROW
   048F 47          933           MOV     B,A
   0490 BE          934           CMP     M
   0491 F2049C      935           JP      BKGD350
   0494 DB21        936           IN      SWITCH1
   0496 E680        937           ANI     SW1BASCR        ;AUTO SCROLL?
   0498 C204A0      938           JNZ     BKGD355
   049B 77          939           MOV     M,A             ;NO...go to TOP LINE
   049C 46          940 BKGD350   MOV     B,M
   049D C303B0      941           JMP     BKGD244
                    942
   04A0 70          943 BKGD355   MOV     M,B             ;SCROLL
   04A1 2E01        944           MVI     L,LDROW
   04A3 34          945           INR     M
   04A4 78          946           MOV     A,B
   04A5 BE          947           CMP     M
   04A6 F204AB      948           JP      BKGD360
   04A9 3600        949           MVI     M,0
   04AB 7E          950 BKGD360   MOV     A,M
   04AC CD03B3      951           CALL    BKGD245
   04AF AF          952           XRA     A
   04B0 CD03F4      953           CALL    BKGD286
   04B3 CD0429      954           CALL    BKGD311
   04B6 3A2001      955           LDA     LDROW
   04B9 D346        956           OUT     VTACCR6
   04BB C9          957           RET
                    958
   04BC DB49        959 BKGD365   IN      VTACCCOL        ;CURSOR LEFT
   04BE 3D          960           DCR     A
   04BF D34C        961 BKGD370   OUT     VTACWCOL
   04C1 F0          962           RP
   04C2 3A2003      963           LDA     MAXCOL
   04C5 D34C        964 BKGD373   OUT     VTACWCOL
   04C7 212014      965 BKGD375   LXI     H,CURSROW       ;CURSOR UP
   04CA 35          966           DCR     M
   04CB F2049C      967           JP      BKGD350
   04CE 3A2002      968           LDA     MAXROW
   04D1 77          969           MOV     M,A
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

   04D2 C3049C        970                JMP       BKGD350
                      971
   04D5 3A2002        972 BKGD380        LDA       MAXROW                ;ERASE LINE ROUTINE: enter with
   04D8 3C            973                INR       A                     ;virtual ROW address in B register
   04D9 57            974                MOV       D,A                   ;destroys all other registers
   04DA 3A2001        975                LDA       LDROW
   04DD 3C            976                INR       A
   04DE 80            977                ADD       B
   04DF BA            978                CMP       D
   04E0 FA04E4        979                JM        BKGD382
   04E3 92            980                SUB       D
   04E4 0F            981 BKGD382        RRC
   04E5 5F            982                MOV       E,A
   04E6 E60F          983                ANI       0FH
   04E8 F600          984                ORI       FASTVRAM
   04EA 57            985                MOV       D,A
   04EB 7B            986                MOV       A,E
   04EC E680          987                ANI       080H
   04EE 5F            988                MOV       E,A
   04EF 3A2003        989                LDA       MAXCOL
   04F2 3C            990                INR       A
   04F3 B3            991                ORA       E
   04F4 5F            992                MOV       E,A
   04F5 210000        993                LXI       H,0                   ;ERASE A LINE, FAST!
   04F8 F3            994                DI
   04F9 39            995                DAD       SP
   04FA EB            996                XCHG
   04FB F9            997                SPHL
   04FC 210020        998                LXI       H,' '
   04FF E5            999 BKGD385        PUSH      H
   0500 E5           1000                PUSH      H
   0501 E5           1001                PUSH      H
   0502 E5           1002                PUSH      H
   0503 E5           1003                PUSH      H
   0504 E5           1004                PUSH      H
   0505 E5           1005                PUSH      H
   0506 E5           1006                PUSH      H
   0507 0D           1007                DCR       C
   0508 C204FF       1008                JNZ       BKGD385
   050B 210000       1009                LXI       H,0
   050E 39           1010                DAD       SP
   050F EB           1011                XCHG
   0510 F9           1012                SPHL
   0511 FB           1013                EI
   0512 C9           1014                RET
                     1015
   0513 D34C         1016 BKGD390        OUT       VTACWCOL              ;CURSOR TO LEFT MARGIN
   0515 DB21         1017                IN        SWITCH1
   0517 E640         1018                ANI       SW1BALF
   0519 C8           1019                RZ
   051A C30488       1020                JMP       BKGD345
                     1021
   051D D34C         1022 BKGD400        OUT       VTACWCOL              ;INSERT LINE
   051F 3A2014       1023                LDA       CURSROW
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

  0527 91           1027                SUB      C
  0528 4F           1028                MOV      C,A
  0529 0C           1029                INR      C
  052A 0D           1030 BKGD405        DCR      C
  052B CA0426       1031                JZ       BKGD310
  052E C5           1032                PUSH     B
  052F CD05F5       1033                CALL     BKGD492
  0532 C1           1034                POP      B
  0533 05           1035                DCR      B
  0534 C5           1036                PUSH     B
  0535 CD053C       1037                CALL     BKGD410
  0538 C1           1038                POP      B
  0539 C3052A       1039                JMP      BKGD405
                    1040
  053C 6B           1041 BKGD410        MOV      L,E
  053D 62           1042                MOV      H,D
  053E CD05F5       1043                CALL     BKGD492
  0541 3A2003       1044                LDA      MAXCOL
  0544 3C           1045                INR      A
  0545 4F           1046                MOV      C,A
  0546 1A           1047 BKGD415        LDAX     D
  0547 77           1048                MOV      M,A
  0548 23           1049                INX      H
  0549 13           1050                INX      D
  054A 0D           1051                DCR      C
  054B C20546       1052                JNZ      BKGD415
  054E C9           1053                RET
                    1054
  054F D34C         1055 BKGD420        OUT      VTACWCOL          ;DELETE LINE
  0551 3A2014       1056                LDA      CURSROW
  0554 47           1057                MOV      B,A
  0555 4F           1058                MOV      C,A
  0556 3A2002       1059                LDA      MAXROW
  0559 90           1060                SUB      B
  055A 47           1061                MOV      B,A
  055B 04           1062                INR      B
  055C 05           1063 BKGD425        DCR      B
  055D CA0570       1064                JZ       BKGD435
  0560 C5           1065                PUSH     B
  0561 41           1066                MOV      B,C
  0562 CD05F5       1067                CALL     BKGD492
  0565 C1           1068                POP      B
  0566 0C           1069                INR      C
  0567 C5           1070                PUSH     B
  0568 41           1071                MOV      B,C
  0569 CD053C       1072                CALL     BKGD410
  056C C1           1073                POP      B
  056D C3055C       1074                JMP      BKGD425
                    1075
  0570 DB48         1076 BKGD435        IN       VTACCROW
  0572 F5           1077                PUSH     PSW
  0573 3A2001       1078                LDA      LDROW
  0576 D34D         1079                OUT      VTACWROW
  0578 CD0426       1080                CALL     BKGD310
  057B F1           1081                POP      PSW
  057C D34B         1082                OUT      VTACWROW
  057E C9           1083                RET
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                     1084
    057F 210486      1085 BKGD440     LXI      H,BKGD344       ;TAB FORWARD
    0582 E5          1086             PUSH     H
    0583 210479      1087             LXI      H,BKGD340
    0586 E5          1088             PUSH     H
    0587 CD05DA      1089             CALL     BKGD462
    058A D2059F      1090 BKGD442     JNC      BKGD446
    058D 7B          1091             MOV      A,E
    058E B7          1092             ORA      A
    058F F20593      1093             JP       BKGD444
    0592 5A          1094             MOV      E,D
    0593 DB49        1095 BKGD444     IN       VTACCCOL
    0595 BA          1096             CMP      D
    0596 F2059F      1097             JP       BKGD446
    0599 7A          1098             MOV      A,D
    059A D34C        1099 BKGD445     OUT      VTACWCOL        ;TAB TO NEXT STOP
    059C E1          1100             POP      H
    059D E1          1101             POP      H
    059E C9          1102             RET
    059F 14          1103 BKGD446     INR      D
    05A0 CD05C7      1104             CALL     BKGD460
    05A3 C3058A      1105             JMP      BKGD442
                     1106
    05A6 2104C5      1107 BKGD450     LXI      H,BKGD373       ;BACK TAB
    05A9 E5          1108             PUSH     H
    05AA 2104BC      1109             LXI      H,BKGD365
    05AD E5          1110             PUSH     H
    05AE CD05DA      1111             CALL     BKGD462
    05B1 D205B5      1112 BKGD452     JNC      BKGD454
    05B4 5A          1113             MOV      E,D
    05B5 14          1114 BKGD454     INR      D
    05B6 DB49        1115             IN       VTACCCOL
    05B8 BA          1116             CMP      D
    05B9 C205C1      1117             JNZ      BKGD456
    05BC 7B          1118             MOV      A,E
    05BD B7          1119             ORA      A
    05BE F2059A      1120             JP       BKGD445
    05C1 CD05C7      1121 BKGD456     CALL     BKGD460
    05C4 C305B1      1122             JMP      BKGD452
                     1123
    05C7 04          1124 BKGD460     INR      B
    05C8 C205E3      1125             JNZ      BKGD466
    05CB 23          1126             INX      H
    05CC 3A2003      1127             LDA      MAXCOL
    05CF BA          1128             CMP      D
    05D0 F205E0      1129             JP       BKGD464
    05D3 E1          1130             POP      H
    05D4 E1          1131             POP      H
    05D5 7B          1132             MOV      A,E
    05D6 B7          1133             ORA      A
    05D7 F0          1134             RP
    05D8 E3          1135             XTHL
    05D9 C9          1136             RET
    05DA 21200A      1137 BKGD462     LXI      H,TABTABLE
    05DD 1100FF      1138             LXI      D,00FFH
    05E0 06F8        1139 BKGD464     MVI      B,-8
    05E2 4E          1140             MOV      C,M
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    05E3 79           1141 BKGD466    MOV     A,C
    05E4 07           1142            RLC
    05E5 4F           1143            MOV     C,A
    05E6 C9           1144            RET
                      1145
    05E7 3A2002       1146 BKGD490    LDA     MAXROW          ;calculate physical row
    05EA 3C           1147            INR     A
    05EB 4F           1148            MOV     C,A             ;enter with virtualrow in B
    05EC 3A2001       1149            LDA     LDROW           ;exit with physical row in A
    05EF 3C           1150            INR     A
    05F0 80           1151            ADD     B
    05F1 B9           1152            CMP     C
    05F2 F8           1153            RM
    05F3 91           1154            SUB     C
    05F4 C9           1155            RET
                      1156
    05F5 CD05E7       1157 BKGD492    CALL    BKGD490
    05F8 0F           1158 BKGD495    RRC
    05F9 4F           1159            MOV     C,A
    05FA E680         1160            ANI     80H
    05FC 5F           1161            MOV     E,A
    05FD 79           1162            MOV     A,C
    05FE E60F         1163            ANI     0FH
    0600 F600         1164            ORI     VIDEORAM
    0602 57           1165            MOV     D,A
    0603 C9           1166            RET
                      1167
                      1168
                      1169 *------------------------------------------------------------
                      1170 *------------------------------------------------------------
                      1171
                      1172
                      1173 *          MODE 1 DECODE/BRANCH TABLE (interprets (ESC) prefixes)
                      1174 *          ================================================
                      1175
    0604 22           1176 BRTBLM1    DB      CONQUO          ;UNLOCK KEYBOARD
    0605 0697         1177            DW      BKGD515
    0607 23           1178            DB      '#'             ;LOCK KEYBOARD
    0608 0695         1179            DW      BKGD510
    060A 2A           1180            DB      '*'             ;CLEAR SCREEN
    060B 043D         1181            DW      BKGD330
    060D 2B           1182            DB      '+'             ;CLEAR SCREEN
    060E 043D         1183            DW      BKGD330
    0610 30           1184            DB      '0'             ;CLEAR ALL TABS
    0611 0277         1185            DW      TABSCLR
    0613 31           1186            DB      '1'             ;SET TABSTOP
    0614 03BA         1187            DW      BKGD250
    0616 32           1188            DB      '2'             ;CLEAR TABSTOP
    0617 03C0         1189            DW      BKGD260
    0619 34           1190            DB      '4'             ;XMIT LINE FORMATED
    061A 077D         1191            DW      BKGD588
    061C 35           1192            DB      '5'             ;XMIT PAGE FORMATTED
    061D 0778         1193            DW      BKGD587
    061F 36           1194            DB      '6'             ;XMIT LINE TRANSPARENT
    0620 077F         1195            DW      BKGD590
    0622 37           1196            DB      '7'             ;XMIT PAGE TRANSPARENT
    0623 0783         1197            DW      BKGD595
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

  0625 3D          1198              DB        '='           ;ENTER MODE 2 ( (ESC) = ROW position)
  0626 06A8        1199              DW        BKGD525
  0628 3F          1200              DB        '?'           ;READ CURSOR POSITION
  0629 06AE        1201              DW        BKGD540
  062B 45          1202              DB        'E'           ;INSERT LINE
  062C 051D        1203              DW        BKGD400
  062E 47          1204              DB        'G'           ;ENTER MODE 4 ( (ESC) G - visual ATTRIBUTE set)
  062F 06A6        1205              DW        BKGD530
  0631 49          1206              DB        'I'           ;BACK TAB
  0632 05A6        1207              DW        BKGD450
  0634 50          1208              DB        'P'           ;PRINT SCREEN TRANSPARENT
  0635 06C8        1209              DW        BKGD550
  0637 51          1210              DB        'Q'           ;INSERT CHARACTER
  0638 0400        1211              DW        BKGD290
  063A 52          1212              DB        'R'           ;DELETE LINE
  063B 054F        1213              DW        BKGD420
  063D 54          1214              DB        'T'           ;ERASE LINE (from CURSOR to end of line)
  063E 0426        1215              DW        BKGD310
  0640 57          1216              DB        'W'           ;DELETE CHARACTER
  0641 03E0        1217              DW        BKGD280
  0643 59          1218              DB        'Y'           ;ERASE PAGE (from CURSOR to end of screen)
  0644 0440        1219              DW        BKGD335
  0646 5A          1220              DB        'Z'           ;DISPLAY CONTROL
  0647 06EC        1221              DW        BKGD585
  0649 5B          1222              DB        CONLBR        ;ENTER MODE 5 ( (ESC) [ prefix interpreted)
  064A 06A5        1223              DW        BKGD535
  064C 69          1224              DB        CONLI         ;TAB FORWARD ( (ESC) i)
  064D 057F        1225              DW        BKGD440
  064F 70          1226              DB        CONLP         ;PRINT SCREEN FORMATTED ( (ESC) p)
  0650 06C6        1227              DW        BKGD545
  0652 71          1228              DB        CONLQ         ;INSERT MODE ON ( (ESC) q)
  0653 041B        1229              DW        BKGD300
  0655 72          1230              DB        CONLR         ;INSERT MODE OFF ( (ESC) r)
  0656 041D        1231              DW        BKGD305
  0658 74          1232              DB        CONLT         ;ERASE LINE - from CURSOR to end of line ( (ESC) t)
  0659 0426        1233              DW        BKGD310
  065B 79          1234              DB        CONLY         ;ERASE PAGE - from CURSOR to end of screen ( (ESC) y)
  065C 0440        1235              DW        BKGD335
                   1236
                   1237
                   1238 *          MODE 0 DECODE/BRANCH TABLE (normal interpret & display)
                   1239 *          =================================================
                   1240
  065E 7F          1241 BRTBLM0      DB        CONDEL        ;SOUND BELL (ASCII "BEL" control character)
  065F 0686        1242              DW        BKGD500
  0661 08          1243              DB        CONBS         ;CURSOR LEFT (ASCII "BS",Back Space)
  0662 04BC        1244              DW        BKGD365
  0664 09          1245              DB        CONHT         ;TAB FORWARD (ASCII "HT",Horizontal Tab - advance cur
  0665 057F        1246              DW        BKGD440
  0667 0A          1247              DB        CONLF         ;CURSOR DOWN (ASCII "LF",Line Feed)
  0668 0488        1248              DW        BKGD345
  066A 0B          1249              DB        CONVT         ;CURSOR UP (ASCII "VT",Vertical Tab - up line)
  066B 04C7        1250              DW        BKGD375
  066D 0C          1251              DB        CONFF         ;CURSOR RIGHT (ASCII "FF", Feed Forward)
  066E 0479        1252              DW        BKGD340
  0670 0D          1253              DB        CONCR         ;CURSOR TO LEFT MARGIN (ASCII "CR", Carriage Return)
  0671 0513        1254              DW        BKGD390
```

```
LOCATION OBJECT CODE LINE    SOURCE LINE

   0673 0E          1255             DB        CONSO          ;UNLOCK KEYBOARD (ASCII "SO", Shift Out)
   0674 0697        1256             DW        BKGD515
   0676 0F          1257             DB        CONSI          ;LOCK KEYBOARD (ASCII "SI", Shift In)
   0677 0695        1258             DW        BKGD510
   0679 1A          1259             DB        CONSUB         ;CLEAR SCREEN (ASCII "SUB", clear all to null)
   067A 043D        1260             DW        BKGD330
   067C 1B          1261             DB        CONESC         ;ENTER MODE 1 ( (ESC) prefix interpret)
   067D 06A9        1262             DW        BKGD520
   067F 1E          1263             DB        CONRS          ;CURSOR TO TOP LEFT (ASCII "RS", cursor home)
   0680 0435        1264             DW        BKGD320
   0682 1F          1265             DB        CONUS          ;CURSOR TO LEFT MARGIN AND DOWN 1 (ASCII "US", new line)
   0683 04B5        1266             DW        BKGD343
   0685 00          1267             DB        0              ;END THIS TABLE
                    1268
                    1269
                    1270 *---------------------------------------------------------------------------------------------
                    1271 *---------------------------------------------------------------------------------------------
                    1272
   0686 DB23        1273 BKGD500     IN        SWITCH2        ;BELL
   0688 E602        1274             ANI       SW2B50HZ
   068A 3EE7        1275             MVI       A,CONBL5
   068C C20691      1276             JNZ       BKGD501
   068F 3EE2        1277             MVI       A,CONBL6
   0691 32203D      1278 BKGD501     STA       VERTBELL
   0694 C9          1279             RET
                    1280
   0695 0680        1281 BKGD510     MVI       B,ST1BKBDL     ;KEYBOARD LOCK
   0697 212007      1282 BKGD515     LXI       H,STATUS1      ;KEYBOARD UNLOCK
   069A 7E          1283             MOV       A,M
   069B E67F        1284             ANI       .NT.ST1BKBDL
   069D B0          1285             ORA       B
   069E 77          1286             MOV       M,A
   069F C9          1287             RET
                    1288
   06A0 78          1289 BKGD560     MOV       A,B
   06A1 D630        1290             SUI       '0'
   06A3 C0          1291             RNZ
   06A4 3C          1292             INR       A              ;(ESC) [ 0 PREFIX
   06A5 3C          1293 BKGD535     INR       A              ;(ESC) [ PREFIX INTERPRET (enter modes- MODE 5)
   06A6 3C          1294 BKGD530     INR       A              ;(ESC) G PREFIX (visual attribute set - enter MODE 4)
   06A7 3C          1295 BKGD527     INR       A              ;(ESC) = ROW ADDRESS PREFIX
   06A8 3C          1296 BKGD525     INR       A              ;(ESC) = PREFIX (ROW position - enter MODE 2)
   06A9 3C          1297 BKGD520     INR       A              ;(ESC) PREFIX INTERPRET (enter MODE 1)
   06AA 322009      1298 BKGD567     STA       BKGMODE
   06AD C9          1299             RET
                    1300
   06AE F3          1301 BKGD540     DI                       ;SEND CURSOR
   06AF 061B        1302             MVI       B,CONESC
   06B1 CD024B      1303             CALL      TRANSMIT       ;(ESC)
   06B4 0659        1304             MVI       B,'Y'
   06B6 CD024B      1305             CALL      TRANSMIT       ;Y
   06B9 3A2014      1306             LDA       CURSROW
   06BC CD024B      1307             CALL      TRANSMIT       ;ROW 32
   06BF DB49        1308             IN        VTACCCOL
   06C1 CD0248      1309             CALL      TRANSMT1       ;COLUMN+32
   06C4 FB          1310             EI
   06C5 C9          1311             RET
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                     1312
    06C6 0E04        1313 BKGD545      MVI      C,ST1B4MAT          ;PRINT SCREEN FORMATTED
    06C8 3A2014      1314 BKGD550      LDA      CURSROW
    06CB 47          1315              MOV      B,A
    06CC 1620        1316              MVI      D,ST1BLPTR
    06CE DB49        1317              IN       VTACCCOL            ;PRINT SCREEN TRANSPARENT
    06D0 212005      1318 BKGD552      LXI      H,LPTCOLMN
    06D3 77          1319              MOV      M,A
    06D4 2C          1320              INR      L
    06D5 70          1321              MOV      M,B
    06D6 2C          1322              INR      L
    06D7 F3          1323              DI
    06D8 7E          1324              MOV      A,M
    06D9 E6F8        1325              ANI      .NT.(ST1B4MAT+ST1B4MXX)
    06DB B2          1326              ORA      D
    06DC B1          1327              ORA      C
    06DD 77          1328              MOV      M,A
    06DE FB          1329              EI
    06DF 76          1330 BKGD555      HLT                          ;WAIT
    06E0 7E          1331              MOV      A,M
    06E1 E638        1332              ANI      ST1BLPTR+ST1BLPTA+ST1BXMIT
    06E3 C206DF      1333              JNZ      BKGD555
    06E6 C9          1334              RET                          ;DONE
                     1335
    06E7 78          1336 BKGD565      MOV      A,B
    06E8 D64D        1337              SUI      'M'
    06EA C0          1338              RNZ
    06EB 3D          1339              DCR      A                   ;(ESC) [ 0 M PREFIX (maintenance - MODE 7)
    06EC C608        1340 BKGD585      ADI      8                   ;(ESC) Z PREFIX (store control character - MODE E
    06EE C306AA      1341              JMP      BKGD567             ;                         display control)
                     1342
    06F1 212009      1343 BKGD570      LXI      H,BKGMODE
    06F4 3607        1344              MVI      M,7
    06F6 78          1345              MOV      A,B
    06F7 E640        1346              ANI      40H                 ;OPERATING IN MAINTENANCE MODE
    06F9 D1          1347              POP      D
    06FA 2E04        1348              MVI      L,SSWITCH1
    06FC 78          1349              MOV      A,B
    06FD C202E9      1350              JNZ      BKGD105             ;SIMULATE BAUD RATE, PARITY
    0700 D5          1351              PUSH     D
    0701 E607        1352              ANI      07H
    0703 CA06AA      1353              JZ       BKGD567             ;0: EXIT MAINTENANCE MODE
    0706 3D          1354              DCR      A
    0707 C20767      1355              JNZ      BKGD580
    070A 110000      1356              LXI      D,0                 ;1: CALCULATE ROM AND REPORT IT
    070D 210000      1357              LXI      H,0
    0710 CD0732      1358 BKGD575      CALL     CRCGEN
    0713 7C          1359              MOV      A,H
    0714 FE08        1360              CPI      08H
    0716 FA0710      1361              JM       BKGD575
    0719 7B          1362              MOV      A,E
    071A CD0245      1363              CALL     TRANSMT3
    071D 7A          1364              MOV      A,D
    071E CD0246      1365              CALL     TRANSMT0
    0721 7B          1366              MOV      A,E
    0722 07          1367              RLC
    0723 07          1368              RLC
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

  0724 E603        1369                ANI     03H
  0726 5F          1370 BKGD579        MOV     E,A
  0727 7A          1371                MOV     A,D
  0728 0F          1372                RRC
  0729 0F          1373                RRC
  072A E630        1374                ANI     30H
  072C B3          1375                ORA     E
  072D CD0248      1376 BKGD578        CALL    TRANSMT1
  0730 FB          1377                EI
  0731 C9          1378                RET
                   1379
  0732 7E          1380 CRCGEN         MOV     A,M
  0733 0F          1381                RRC
  0734 E680        1382                ANI     80H
  0736 AE          1383                XRA     M
  0737 47          1384                MOV     B,A
  0738 7A          1385                MOV     A,D
  0739 17          1386                RAL
  073A A8          1387                XRA     B
  073B 47          1388                MOV     B,A
  073C 7A          1389                MOV     A,D
  073D 0F          1390                RRC
  073E 0F          1391                RRC
  073F 4F          1392                MOV     C,A
  0740 0F          1393                RRC
  0741 0F          1394                RRC
  0742 E6F0        1395                ANI     0F0H
  0744 A8          1396                XRA     B
  0745 47          1397                MOV     B,A
  0746 79          1398                MOV     A,C
  0747 E680        1399                ANI     80H
  0749 A8          1400                XRA     B
  074A 47          1401                MOV     B,A
  074B 5F          1402                MOV     E,A
  074C E67F        1403                ANI     7FH
  074E A8          1404                XRA     B
  074F 47          1405                MOV     B,A
  0750 7B          1406                MOV     A,E
  0751 07          1407                RLC
  0752 E601        1408                ANI     01H
  0754 A8          1409                XRA     B
  0755 47          1410                MOV     B,A
  0756 7B          1411                MOV     A,E
  0757 0F          1412                RRC
  0758 0F          1413                RRC
  0759 0F          1414                RRC
  075A 0F          1415                RRC
  075B 4F          1416                MOV     C,A
  075C A8          1417                XRA     B
  075D 47          1418                MOV     B,A
  075E 79          1419                MOV     A,C
  075F 0F          1420                RRC
  0760 E680        1421                ANI     80H
  0762 A8          1422                XRA     B
  0763 5A          1423                MOV     E,D
  0764 57          1424                MOV     D,A
  0765 23          1425                INX     H
```

```
LOCATION OBJECT CODE LINE    SOURCE LINE

    0766 C9          1426              RET
                     1427
    0767 3D          1428 BKGD580      DCR      A
    0768 C20798      1429              JNZ      BKGD600           ;other
    076B DB21        1430              IN       SWITCH1           ;2: read switches
    076D 57          1431              MOV      D,A
    076E CD0245      1432              CALL     TRANSMT3
    0771 DB23        1433              IN       SWITCH2
    0773 E60F        1434              ANI      0FH
    0775 C30726      1435              JMP      BKGD579
                     1436
    0778 0E04        1437 BKGD587      MVI      C,ST1B4MAT        ;TRANSMIT PAGE FORMATTED
    077A C30783      1438              JMP      BKGD595
    077D 0E04        1439 BKGD588      MVI      C,ST1B4MAT        ;TRANSMIT LINE FORMATTED
    077F 3A2014      1440 BKGD590      LDA      CURSROW           ;TRANSMIT LINE TRANSPARENT
    0782 47          1441              MOV      B,A
    0783 1608        1442 BKGD595      MVI      D,ST1BXMIT
    0785 AF          1443              XRA      A                 ;TRANSMIT PAGE TRANSPARENT
    0786 C306D0      1444              JMP      BKGD552
                     1445
    0789 212000      1446 SCRCRC       LXI      H,SCRATCH
    078C 110000      1447              LXI      D,0
    078F CD0732      1448 SCRCRC1      CALL     CRCGEN
    0792 7D          1449              MOV      A,L
    0793 B7          1450              ORA      A
    0794 C2078F      1451              JNZ      SCRCRC1
    0797 C9          1452              RET
                     1453
    0798 3D          1454 BKGD600      DCR      A
    0799 C206AA      1455              JNZ      BKGD567           ;any other exits
    079C F3          1456              DI                         ;3: perform SCRATCHPAD tests
    079D 315CC0      1457              LXI      SP,FREEDRAM+48
    07A0 CD0789      1458              CALL     SCRCRC
    07A3 25          1459              DCR      H
    07A4 D5          1460              PUSH     D
    07A5 55          1461              MOV      D,L
    07A6 7D          1462 BKGD605      MOV      A,L
    07A7 0F          1463              RRC
    07A8 0F          1464              RRC
    07A9 0F          1465              RRC
    07AA 0F          1466              RRC
    07AB AE          1467              XRA      M
    07AC 77          1468              MOV      M,A
    07AD 2C          1469              INR      L
    07AE C207A6      1470              JNZ      BKGD605
    07B1 7E          1471 BKGD610      MOV      A,M
    07B2 2F          1472              CMA
    07B3 77          1473              MOV      M,A
    07B4 BE          1474              CMP      M
    07B5 CA07BA      1475              JZ       BKGD615
    07B8 1601        1476              MVI      D,1
    07BA 2C          1477 BKGD615      INR      L
    07BB C207B1      1478              JNZ      BKGD610
    07BE 7D          1479 BKGD620      MOV      A,L
    07BF 0F          1480              RRC
    07C0 0F          1481              RRC
    07C1 0F          1482              RRC
```

```
LOCATION OBJECT CODE LINE    SOURCE LINE

   07C2 0F          1483             RRC
   07C3 2F          1484             CMA
   07C4 AE          1485             XRA     M
   07C5 77          1486             MOV     M,A
   07C6 2C          1487             INR     L
   07C7 C207DE      1488             JNZ     DKGD620
   07CA 7A          1489             MOV     A,D
   07CB B7          1490             ORA     A
   07CC C207DA      1491             JNZ     DKGD625
   07CF CD0789      1492             CALL    SCRCRC
   07D2 E1          1493             POP     H
   07D3 7B          1494             MOV     A,E
   07D4 BD          1495             CMP     L
   07D5 C207DA      1496             JNZ     DKGD625
   07D8 7A          1497             MOV     A,D
   07D9 BC          1498             CMP     H
   07DA 3E11        1499 DKGD625     MVI     A,11H
   07DC 312031      1500             LXI     SP,STACKEND-2
   07DF C2072D      1501             JNZ     DKGD578
   07E2 3D          1502             DCR     A
   07E3 C3072D      1503             JMP     DKGD578
                    1504
                    1505
                    1506 *----------------------------------------------------------------------------
                    1507 *----------------------------------------------------------------------------
                    1508
                    1509
                    1510             ORG     2000H
                    1511
          (2000)    1512 SCRATCH     EQU     $                 ;8155 RAM, SCRATCHPAD
   2000             1513 SCR8251A    DS      1                 ;8251A COMMAN REGISTER IMAGE
   2001             1514 LDROW       DS      1                 ;VTACCR6 IMAGE (LAST DISPLAYED DATA ROW)
   2002             1515 MAXROW      DS      1                 ;MAXIMUM ROW COUNT
   2003             1516 MAXCOL      DS      1                 ;MAXIMUM COLUMN COUNT
   2004             1517 SSWITCH1    DS      1                 ;SWITCH1 IMAGE
   2005             1518 LPTCOLMN    DS      1                 ;LOCAL PRINT COLUMN POINTER
   2006             1519 LPTROW      DS      1                 ;LOCAL PRINT ROW POINTER
   2007             1520 STATUS1     DS      1                 ;GENERAL STATUS
          (0080)    1521 ST1BKBDL    EQU     80H               ;KEYBOARD LOCK
          (0040)    1522 ST1BINSM    EQU     40H               ;INSERT CHARACTER MODE
          (0020)    1523 ST1BLPTR    EQU     20H               ;LOCAL PRINT TRIGGER
          (0010)    1524 ST1BLPTA    EQU     10H               ;LOCAL PRINT ACTIVE
          (0008)    1525 ST1BXMIT    EQU     08H               ;BLOCK TRANSMIT FLAG
          (0004)    1526 ST1B4MAT    EQU     04H               ;FORMATTED OUTPUT FLAG
          (0003)    1527 ST1B4MXX    EQU     03H               ;FORMAT SEND STAE COUNTER
   2008             1528 STATUS2     DS      1                 ;GENERAL STATUS 2
          (0003)    1529 ST2BFETM    EQU     03H               ;FRONT-END TRANSPARENCY MODE
   2009             1530 DKGMODE     DS      1                 ;DISPLAY DECODE MODE
                    1531
                    1532 *           MODE    0                 ;NORMAL INTERPRET AND DISPLAY
                    1533 *           MODE    1                 ;<ESC> PREFIXED INTERPRET
                    1534 *           MODE    2                 ;<ESC> = ROW POSITION
                    1535 *           MODE    3                 ;<ESC> = COLUMN POSITION
                    1536 *           MODE    4                 ;<ESC> G VISUAL ATTRIBUTE SET
                    1537 *           MODE    5                 ;<ESC> I PREFIXED INTERPRET
                    1538 *           MODE    6                 ;<ESC> I 0 PREFIXED INTERPRET
                    1539 *           MODE    7                 ;<ESC> I 0 M MAINTENANCE MODE
```

```
LOCATION OBJECT CODE LINE       SOURCE LINE

                     1540 *              MODE      0              ;(ESC) Z STORE CONTROL CHARACTER
                     1541
    200A             1542 TABTABLE       DS        10             ;TABSTOP  BITMASK TABLE
    2014             1543 CURSROW        DS        1              ;CURSOR ROW-VIRTUAL
    2015             1544                DS        STKSIZE        ;STACK
           (2033)    1545 STACKEND       EQU       $
    2033             1546 XMTQUEUE       DS        XMTQLEN        ;TRANSMIT Q
           (2039)    1547 XMTQEND        EQU       $
    2039             1548 DISQSPTR       DS        1
    203A             1549 DISQFPTR       DS        1
    203B             1550 XMTQFPTR       DS        1
    203C             1551 XMTQSPTR       DS        1
    203D             1552 VERTBELL       DS        1              ;BELL TIMER
           (203E)    1553 DISQUEUE       EQU       $              ;DISPLAY Q IS REST OF SCRATCHPAD
                     1554                END

Errors=   0
```

```
LINE#   SYMBOL         TYPE    REFERENCES

 570   BDRTABL          A  608
 653   BKGD100          A  749
 662   BKGD105          A  1350
 685   BKGD120          A  683
 690   BKGD150          A  655,660,712
 706   BKGD160          A  704
 747   BKGD165          A  707
 905   BKGD166          A  722,739
 906   BKGD167          A  719
 908   BKGD170          A  731
 907   BKGD180          A
 751   BKGD200          A  733
 753   BKGD210          A  725
 754   BKGD211          A  728,759
 770   BKGD230          A  737
 777   BKGD235          A  775
 781   BKGD240          A  735
 785   BKGD242          A  806
 792   BKGD244          A  790,941
 793   BKGD245          A  951
 798   BKGD250          A  1187
 803   BKGD260          A  1189
 809   BKGD270          A  798,803
 821   BKGD275          A  824
 826   BKGD280          A  1217
 839   BKGD285          A  826,848,875
 840   BKGD286          A  953
 848   BKGD290          A  911,1211
 857   BKGD295          A  862,882
 863   BKGD297          A  856
 867   BKGD300          A  1229
 868   BKGD305          A  1231
 875   BKGD310          A  888,1031,1080,1215,1233
 876   BKGD311          A  954
 884   BKGD320          A  887,1264
 887   BKGD330          A  1181,1183,1260
 888   BKGD335          A  1219,1235
 894   BKGD336          A  892
 896   BKGD337          A  903
 920   BKGD340          A  1087,1252
 928   BKGD343          A  1266
 929   BKGD344          A  1085
 930   BKGD345          A  1020,1248
 940   BKGD350          A  935,967,970
 943   BKGD355          A  938
 950   BKGD360          A  948
 959   BKGD365          A  1109,1244
 961   BKGD370          A
 964   BKGD373          A  1107
 965   BKGD375          A  1250
 972   BKGD380          A  901
 981   BKGD382          A  979
 999   BKGD385          A  1008
1016   BKGD390          A  1254
1022   BKGD400          A  1203
1030   BKGD405          A  1039
1041   BKGD410          A  1037,1072
```

```
FILE: TERMINAL:pJUAN    CROSS REFERENCE TABLE          PAGE  30

LINE#   SYMBOL        TYPE     REFERENCES

 1047  BKGD415         A  1052
 1055  BKGD420         A  1213
 1063  BKGD425         A  1074
 1076  BKGD435         A  1064
 1085  BKGD440         A  1225,1246
 1090  BKGD442         A  1105
 1095  BKGD444         A  1093
 1099  BKGD445         A  1120
 1103  BKGD446         A  1090,1097
 1107  BKGD450         A  1207
 1112  BKGD452         A  1122
 1114  BKGD454         A  1112
 1121  BKGD456         A  1117
 1124  BKGD460         A  1104,1121
 1137  BKGD462         A  1089,1111
 1139  BKGD464         A  1129
 1141  BKGD466         A  1125
 1146  BKGD490         A  792;1157
 1157  BKGD492         A  300,1033,1043,1067
 1158  BKGD495         A  844,914
 1273  BKGD500         A  1242
 1278  BKGD501         A  1276
 1281  BKGD510         A  1179,1258
 1282  BKGD515         A  1177,1256
 1297  BKGD520         A  1262
 1296  BKGD525         A  1199
 1295  BKGD527         A
 1294  BKGD530         A  1205
 1293  BKGD535         A  1223
 1301  BKGD540         A  1201
 1313  BKGD545         A  1227
 1314  BKGD550         A  1209
 1318  BKGD552         A  1444
 1330  BKGD555         A  1333
 1289  BKGD560         A  741
 1336  BKGD565         A  743
 1298  BKGD567         A  1341,1353,1485
 1343  BKGD570         A  745
 1358  BKGD575         A  1361
 1376  BKGD578         A  1501,1503
 1370  BKGD579         A  1435
 1428  BKGD580         A  1355
 1340  BKGD585         A  1221
 1437  BKGD587         A  1193
 1439  BKGD588         A  1191
 1440  BKGD590         A  1195
 1442  BKGD595         A  1197,1438
 1454  BKGD600         A  1429
 1462  BKGD605         A  1470
 1471  BKGD610         A  1478
 1477  BKGD615         A  1475
 1479  BKGD620         A  1488
 1499  BKGD625         A  1491,1496
 1530  BKGMODE         A  653,714,782,1298,1343
 1241  BRTBLM0         A  724
 1176  BRTBLM1         A  752
  730  BRTBLX0         A  727
```

```
FILE: TERMINAL:pJUAN     CROSS REFERENCE TABLE          PAGE  31

LINE#   SYMBOL        TYPE     REFERENCES

 222    COMM100         A   167,225
 247    COMM110         A   232,242
 262    COMM111         A   259
 264    COMM112         A   253
 269    COMM118         A   261,266
 275    COMM120         A   245,255,263
 280    COMM130         A   439
 291    COMM130A        A   412
 292    COMM130B        A   407
 297    COMM131         A   360,428
 322    COMM131B        A   319
 325    COMM131D        A   321
 329    COMM131F        A   323
 336    COMM131J        A   316
 338    COMM132         A   296,416
 341    COMM134         A   295
 348    COMM134B        A   422
 350    COMM135         A   343,347
 370    COMM139         A   287
 362    COMM140         A   283
 366    COMM150         A   402
 374    COMM160         A
 377    COMM165         A   423
 386    COMM170         A   383
 392    COMM180         A   301,340,349
 404    COMM190         A   376
 424    COMM192         A   352
 166    COMMINT         A
  98    CONBEL          A
  91    CONBL5          A   1275
  90    CONBL6          A   595,1277
  99    CONBS           A   1243
 104    CONCR           A   338,1253
 121    CONDEL          A   244,1241
 108    CONESC          A   265,320,1261,1302
 103    CONFF           A   1251
 100    CONHT           A   1245
 112    CONLBR          A   1222
 101    CONLF           A   348,1247
 113    CONLI           A   1224
 114    CONLP           A   1226
 115    CONLQ           A   1228
 116    CONLR           A   1230
 117    CONLT           A   1232
 118    CONLY           A   1234
 119    CONORBAR        A
 111    CONQUO          A   1176
 109    CONRS           A   1263
 106    CONSI           A   1257
 105    CONSO           A   1255
 107    CONSUB          A   596,1259
 120    CONTILDE        A
 110    CONUS           A   1265
 102    CONVT           A   1249
  54    CRB251A         A   227,230,236,280,367,457,464,483,505,643,679,686
  59    CRBCON          A
  67    CRBDSR          A
```

```
FILE: TERMINAL:pJUAN     CROSS REFERENCE TABLE          PAGE  32

LINE#   SYMBOL        TYPE     REFERENCES

  65  CRBDTR           A  456
  56  CRBEPS           A
  62  CRBERST          A  235
  63  CRBFBRK          A
  68  CRBFE            A  231,238
  58  CRBLEN           A
  69  CRBOE            A  231
  70  CRBPE            A  231
  57  CRBPEN           A
  64  CRBRCVEN         A  456
  60  CRBRST           A  678
  61  CRBRTS           A  456,480,503
  72  CRBRXRDY         A  228
  55  CRBSBS           A  684
  71  CRBTXEMT         A  466
  66  CRBTXEN          A  364,401,481
  73  CRBTXRDY         A  281,466
1380  CRCGEN           A  1358,1448
1543  CURSROW          A  409,795,885,894,930,965,1023;1056,1306,1314,1440
 551  DISPB100         A  548
 539  DISPBUFF         A  201,277
1549  DISQFPTR         A  539
1548  DISQSPTR         A  145,551,699
1553  DISQUEUE         A  549,591,592,705
  53  DR8251A          A  243,247,397
  88  FASTVRAM         A  984
  87  FREEDRAM         A  1457
 135  INIT100          A  137
 628  INIT120          A  158
 639  INIT130          A  636
 591  INITABLE         A  144,598
 598  INITLEN          A  146
 209  INTREXIT         A  282,368
  19  IOCS8155         A  140,697
  20  IOCSINIT         A  139,696
 201  KEYBD100         A  193
 202  KEYBD110         A  200
 207  KEYBD120         A  196,204
 184  KEYBDINT         A
  35  KEYBOARD         A  188
1514  LDROW            A  641,944,955,975,1078,1149
1518  LPTCOLMN         A  291,304,404,493,1318
1519  LPTROW           A  489
1516  MAXCOL           A  292,773,840,849,876,923,963,989,1044,1127
1515  MAXROW           A  353,488,639,786,897,932,968,972,1025,1059,1146
 564  MOVER            A  147,569,630,834
 565  MOVER1           A
 131  SAVE100          A
1513  SCR8251A         A  234,362,399,454,468,675
1512  SCRATCH          A  131,1446
1446  SCRCRC           A  1458,1492
1448  SCRCRC1          A  1451
1517  SSWITCH1         A  143,392,657,1348
1526  ST1B4MAT         A  315,1313,1325,1437,1439
1527  ST1B4MXX         A  318,330,415,1325
1522  ST1BINSM         A  867,870,910
1521  ST1BKBDL         A  195,1281,1284
```

```
FILE: TERMINAL:pJUAN   CROSS REFERENCE TABLE        PAGE 33

LINE#  SYMBOL       TYPE   REFERENCES

 1524  ST1BLPTA      A  286,476,486,499,1332
 1523  ST1BLPTR      A  472,475,1316,1332
 1525  ST1BXMIT      A  351,375,418,1332,1442
 1529  ST2BFETM      A  252
 1545  STACKEND      A  138,1500
 1520  STATUS1       A  194,205,313,350,374,470,497,868,909,1282
 1528  STATUS2       A  250
   92  STKSIZE       A  1544
   27  SW1BALF       A  346,421,1018
   26  SW1BASCR      A  937
   30  SW1BBDR       A  659,663
   29  SW1BECHO      A  199
   28  SW1BPAR       A  659,667
   46  SW2B16L.      A  149,635,890
   45  SW2B50HZ      A  149,1274
   42  SW2BCTS       A  371
   43  SW2BKCE       A  203
   44  SW2BPERI      A  241
   41  SW2BPRDY      A  289
   25  SWITCH1       A  141,198,345,420,656,661,936,1017,1430
   40  SWITCH2       A  148,202,240,288,370,633,889,1273,1433
  556  TABSCLR       A  1185
  555  TABSINIT      A  632
 1542  TABTABLE      A  556,557,809,1137
   48  TIMRHI        A  695
   47  TIMRLOW       A  692
  530  TRANS100      A  528
  518  TRANSMIT      A  197,1303,1305,1307
  515  TRANSMT0      A  1365
  516  TRANSMT1      A  1309,1376
  517  TRANSMT2      A
  514  TRANSMT3      A  1363,1432
  436  VERT100       A  176
  461  VERT110       A  441
  486  VERT120       A  474
 1552  VERTBELL      A  205,443,1278
  175  VERTINT       A
   86  VIDEORAM      A  87,1164
   80  VTACCCOL      A  405,810,817,827,839,915,920,959,1095,1115,1308,1317
   78  VTACCR6.      A  956
   79  VTACCROW      A  843,913,1076
   85  VTACMEM       A  628
   81  VTACRSET      A  157
   84  VTACSTRT      A  156,631
  604  VTACTABL      A
   82  VTACWCOL      A  777,884,926,929,961,964,1016,1022,1055,1099
   83  VTACWROW      A  793,1079,1082
 1547  XMTQEND       A  382,527
 1550  XMTQFPTR      A  518
   89  XMTQLEN       A  1546
 1551  XMTQSPTR      A  377,530
 1546  XMTQUEUE      A  384,529,593,594
```

APPENDIX H

PARTS LIST

IC #	DESCRIPTION
U1	8085A (uP - 4 MHz)
U2	8155 (RAM-I/O-TIMER)
U3	8251A (USART)
U4	2732A (EPROM - 200 ns) (4K x 8)
U5	HM-6516 (RAM - 200 ns) (2K x 8) [Harris]
U6	CRT-5037 (VTAC)
U7	CRT-8002A-011 (VDAC) 20 MHz
U8	74LS138 (3:8 decoder)
U9	74LS374 (octal latch TS)
U10	74LS244 (octal buffer TS)
U11	74LS04 (hex inverter)
U12	74LS14 (hex Schmitt inverter)
U13, U14, U15	74LS74 (dual D flip flop)
U16, U17	74LS32 (quad 2 I/P OR)
U18	74LS86 (quad 2 I/P XOR)
U19	74LS00 (quad 2 I/P NAND)
U20	8286 (octal bus transceiver)
U21	1488 (quad RS232C driver)
U22	1489A (quad RS232C receiver)
U23 to U28	74LS157 (quad 2:1 Mux)
U29	74LS08 (quad 2 I/P AND)
U30	74S04 (hex inverter)
U31	74LS161 (4 bit counter, synchronous)
U32	74LS244 (octal buffer/driver)
U33 (optional)	2732A (EPROM - 200 ns) (4K x 8)

RESISTORS

(all in ohms, ±5%, ¼W, carbon film)

R1 -	1K
R2 -	470
R3 -	75
R4 -	27K
R5 -	560
R6 -	68
R7 -	390
R8 -	330
R9 -	330
R10 -	180
R11 -	330
R12 -	100
R13 -	200
R14 -	1K

POTENTIOMETERS

VR1 - 100K Ω pot (optional)
VR2 - 200 Ω pot (optional)

DIP RESISTORS

Z1, Z2 - 10K Ω, 8 per package Beckman
Z3, Z4 - 10K Ω, 8 per package Beckman

CAPACITORS

C1, C9 - 10uF, 35V, radial, electrolytic
C2, C5 - 1000pF, 50V, ceramic
C3 - 4700pF, 50V, ceramic
C4, C6, C7, C8 - 100pF, 100V, ceramic
C10-C35 - 0.01uF, 50V, ceramic

DIODE

D1 IN4148

TRANSISTOR

Q1 2N3904

SWITCHES

S1, S2 - 8 position, DIP
S3 - normally open push button

CRYSTALS

Y1 - 6.144 MHz, 0.01% for 0°C to 70°C, HC18 w/hold down pin
Y2 - 17.0748 MHz, 0.01% for 0°C to 70°C, HC18 w/hold down pin

BUZZER

ALARM - miniature buzzer

CONNECTORS

J1
J2
J3
J4
J5
J6
J7

REFERENCES

1. R.D. Taylor, "CRT Terminal Evaluation", Digital Design, Vol. 4, No. 3, pp. 48-53, March 1974.

2. , "CRT Terminals", Digital Design, Vol. 6, No. 1, pp. 44-58, January 1976.

3. L. Solomon, "Interactive Terminals Suited to a CRT", Digital Design, Vol. 6, No. 5, pp. 54-62, May 1976.

4. D. Ratcliffe, "Key to the 4th Generation CRT; Custom LSI Circuitry", Digital Design, Vol. 8, No. 11, pp. 66-71, November 1978.

5. M.S. Kiver & M. Faufman; Television Simplified, 7th edition, Delmar Publishers, New York, 1973.

6. Sol Sherr, Electronic Displays, John Wiley & Sons, New York, 1979, pp. 77-79, Chapters 2 and 4.

7. Jim Bechtold, "All About CRTs", Bench Briefs, Hewlitt Packard Journal, pp. 1-7, March-May 1981.

8. NTSC standard for TV and color TV.

9. Pete Stark, "Television Interfaces for Your Home Computer", Modern Electronics Magazine, pp. 27-32, September 1978.

10. B. Matic and L. Trottier, "Graphics CRT Displays Are Easy to Design with Raster-Scanning Imaging", ED, 1978.

11. 8275 CRT Controller Data Sheet, Microprocessor and Peripherals Handbook, Intel, Santa Clara, CA., 1983, pp. 6-306, 6-309.

12. Gerry Kane, The CRT Controller Handbook, Osborne/McGraw Hill; Berkeley, 1980, Chapters 1 to 5.

13. 66700 Series Data Sheet, Microcomputer Data Library Databook, Motorola, Phoenix, Arizona, 1978, pp. 5-158, 5-171.

14. Don Lancaster, TV Typewriter Cookbook, Howard W. Sams Co., Indianapolis, 1978, Chapter 3.

15. Don Lancaster, The Cheap Video Cookbook, Howard W. Sams Co., Indianapolis, 1979, Chapter 2.

16. Don Lancaster, Son of Cheap Video, Howard W. Sams Co., Indianapolis, 1980, Chapter 3.

17. L. Trottier and B. Matic, "Transparent Memory Ends Conflicts Over CRT Control", Electronics, pp. 136-139, July 5, 1979.

18. Hitachi, HD6845 CRT Controller Users Manual, 1980, p. 7.

19. Mostek, MK 3807 Programmable CRT Video Control Unit (VCU) Data Sheet, Microelectronic Data Book; Carrollton, Texas, 1982, pp. IX-11, IX-21.

20. Standard Microsystems Corporation, CRT 5027/37/57, CRT Video Timer and Controller Data Sheet, SMC Data Catalog, Hauppauge, New York, 1983, pp. 239-246.

21. Solid State Scientific Inc., SND 5027/37 CRT Video Timer-Controller, Datacom Databook, Montgomerville, Pennsylvania, 1982, pp. 4-3, 4-10.

22. Texas Instruments, TMS 9927 Video Timer/Controller, 9900 Family Systems Design Handbook, Houston, Texas, 1982, pp. 8-297, 8-301.

22. Electrical Performance Standard - Monochrome Television Studio Facilities, EIA Standard RS-170, New York, 1957.

23. Motorla, MC6845 CRT Controller Data Sheet, Microprocessors Data Manual, Austin, Texas, 1981, pp. 4-457, 4-479.

24. Hitachi, HD6845 CRT Controller Data Sheet, Microcomputer Data Book, Tokio, 1981, pp. 546-584.

25. AMI, S68045 CRT Controller Data Sheet.

26. Synertek, SY6545-1 CRT Controller Data Sheet, 1981-1982 Data Catalog, Santa Clara, California, 1981, pp. 3-155, 3-168.

27. Rockwell, R6545-1 CRT Controller Data Sheet, Data Catalog; Anaheim, California, 1981.

28. Intel, 8275 Programmable CRT Controller, Component Data Catalog, Santa Clara, California, 1978, pp. 12-165, 12-187.

29. Western Digital, WD8275 Programmable CRT Controller, Data Communications Products Handbook, Irvine, California, 1983, pp. 11, 34.

30. Intel, 8276 Small System CRT Controller, Microprocessor and Peripheral Handbook, Santa Clara, California, 1983, pp. 6-330, 6-346.

31. Thomas Rossi, "Low Cost CRT Control Does More With Less", Electronic Design, April 30, 1981.

32. Chris Tennant, "The Intel 8275 CRT Controller", Byte, May 1979, pp. 130-148.

33. John Katausky, "A Low Cost CRT Terminal Using the 8275", Intel Application Note No. 62, Santa Clara, California, 1979.

34. J. Murray & G. Alexy, "CRT Terminal Design Using the Intel 8275 and 8279", Intel Application Note, Peripheral Design Handbook, Santa Clara, California, 1978, p. 2-190, 2-247.

35. P. Snigier, "Special Report - Computer Terminals: Dumb, Smart or Intelligent, They Form the Key Man-Computer Interface", EDN, May 5, 1977, pp. 42-54.

36. S. Davis, "CRT Terminals - Programmability's the Key to IQ", EDN, June 5, 1978, pp. 74-85.

37. K.M. Straith, "Many Inexpensive Options Let You Customize CRT Terminals", EDN, January 20, 1980, pp. 129-134.

38. P. Snigier, "Alphanumeric Terminals: Some Selection Criteria", Digital Design, June 1980, pp. 28-33.

39. C. Warren, "Intelligent Terminals", EDN, November 20, 1980, pp. 128-141.

40. R.H. Edelson, "Terminals - They Do More and Cost Less", Interface Age, November 1980, pp. 72-73.

41. A. Caker, D.J. Hart, T.F.M. Stewart, Visual Display Terminals, John Wiley & Sons, New York, 1980, Chapters 3, 4 and 5.

42. Standard Microsystems Corporation, CRT 8002, Video Display Attributes Controller Video Generator (VDAC) Data Sheet, SMC Data Catalog, Hauppauge, New York, 1983, pp. 279-291.

43. Solid State Scientific, SND 8002 CRT Video Generator and Attributes Controller Data Sheet, Montgomerville, Pennsylvania, 1983.

44. Intel, 8085A/8085A-2 Single Chip 8-bit N-channel Microprocessor, Microprocessor and Peripheral Handbook, Santa Clara, California, 1983, pp. 2-10, 2-29.

45. Intel, 8155/8156 2048-Bit Static NMOS RAM with I/O Ports and Timer, Microprocessor and Peripheral Handbook, Santa Clara, California, 1983, pp. 2-30, 2-44.

46. Intel, 8251A, Programmable Communications Inteface, Microprocessor and Peripheral Handbook, Santa Clara, California, 1983, pp. 6-122, 6-138.

47. EJA RS232C Standard, Washington, August 1969, (revised June 1981).

48. Lear Siegler Inc., ADM-42 Data Display Terminal Operators Manual, Anaheim, California, September 1982.

49. Lear Siegler Inc., ADM-42 Ergonomic Terminal Video Display Unit Maintenance Manual, Anaheim, California, September 1982.

50. Standard Microsystems Inc., "The CRT 5027/37/57 VTAC", Application Note 1-1, Hauppauge, New York, August 1981.