A SINGLE - CHIP GRAPHICS DISPLAY CONTROLLER FOR SOPHISTICATED DISPLAY TERMINALS

Jeff Wise and Henryk Szejnwald

PRESENTED TO THE

ISSC Conference
February 19, 1981

SESSION: THPM 13.4

NEC MICROCOMPUTERS, INC.
173 Worcester Street
Wellesley, MA 02181

SLIDE 1
"NEED TO
UNDERSTAND"

In this age of computers and communications, collecting data seems easier than understanding it. With the amounts of information available today, it is impossible to thoroughly study raw data to uncover its message. Presenting this data in a graphical form, on the other hand, can make possible a quick and accurate evaluation of the information and its subtleties. Computer-generated graphics is a natural way to make this presentation.

SLIDE 2
"GDC STAR"

An increasingly popular technique for computer graphics is the raster-scan terminal. Its capabilities for bright colors, filled areas, selective erase, and interactive operation appeal to its users, while its rapidly improving price/performance ratio appeals to its buyers. Indeed, plummeting memory and CRT display costs are encouraging rapid growth in the computer graphics industry, but compared with the other components of a graphics terminal, advances in display controller technology have lagged behind. A more technologically advanced display controller has been needed for some time.

SLIDE 3
"RASTER-SCAN
GRAPHICS"

In what follows, I will describe a new display controller for raster-scan computer graphics application which satisfies this need. Using advanced LSI technology and a sophisticated architecture, this graphics display controller, (abbreivated - GDC), handles the high speed, repetitive tasks of a computer graphics system. When combined with a microprocessor, a bit-mapped display memory, and a CRT display unit, this controller is the key to a low-IC count, high performance system.

II. DESIGN CONSIDERATIONS

The design of the GDC encompasses many considerations. First among these is the overall functionality requirement. A sophisticated graphics system requires activities at several levels to generate a graphics display.

SLIDE 4 "ACTIVITIES"

First, a high level description of the object to be displayed needs to be generated and maintained using a vector list, for example. Second, this representation must be reduced to drawable figures which result from application dependent manipulations such as 3D rotation and hidden line removal. Third, drawing parameters must be calculated as required by the figure drawing algorithm. Fourth, the addresses of each figures' pixels must be determined. Fifth, the memory at these addresses must be modified to draw the figures into display memory. And, finally, the video raster-scanning process must proceed continuously.

Let me point out that the first three activities share the requirement for involved and application dependent manipulations. Activities 4, 5, and 6, on the other hand, are repetitive in nature and use comparatively straightforward and application independent algorithms. This makes them highly suitable for a special purpose, integrated circuit implementation. Designing this section of the system into an IC yields high performance and a low package count.

while maintaining application generality.

Another major design consideration is the effect of the system architecture on throughput. Although architectures which use task partitioning promise great improvements in performance, the divisions must be chosen carefully to realize these potentialities. The figure drawing process, which includes activities 3, 4, and 5, is the most important task to partition due to its high frequency of occurrence.

SLIDE 5
"DESIGN
CONSIDERATIONS"

To this end, display memory should be separated from system memory so that the microprocessor can calculate the drawing parameters for the next figure, while the current figure is being drawn by the GDC. The GDC needs to control both the display memory address and data to be able to draw figures while the system microprocessor is calculating the next figure's parameters. In addition, the interface between the microprocessor and the GDC must allow an efficient flow of drawing information to keep system throughput high.

The display memory architecture also must respond to design considerations. An architecture which can use 64K RAMs will have an advantage if, for example, 1024 by 1024 pixel displays can be generated with 64K RAMs when a 16-bit data word is used. In addition, the most flexible display memory is linearly addressed so that there is one continuous range of addresses used for the display; but, generating a two dimensional "X-Y" display from this linear memory structure requires more complex circuitry in the GDC. For additional flexibility, the memory should be able to support both bit-mapped graphics and coded characters within the same memory structure. Finally, read-modify-write operations should be used to speed up display memory manipulations.

Display generation should emphasize flexibility in the video timing, and should have powerful capabilities, such as partitioned display areas, mixed graphics and characters, zoom, pan, scroll, and multiple GDC synchronization. Finally, the device must be housed in a package no larger than 40 pins.

III. DESIGN CONCEPT

SLIDE 6
"SYSTEM BLOCK
DIAGRAM"

From these considerations, a design concept has been developed. Its basic architecture is illustrated in this system block diagram. The graphics display controller is seen situated between the microprocessor and the display memory. Notice that the display memory is completely isolated from the microprocessor system. The GDC forms the only bridge between the two memories.

SLIDE 7
"GDC BLOCK
DIAGRAM"

Looking more closely at the display controller, we see an 8-bit data interface to the microprocessor. A 16 byte FIFO is used to pass commands to the GDC's command processor over the internal data path, and the command's parameters are stored in the RAM.

The display memory interface uses a time multiplexed address and data bus to control the display memory. Dedicated hardware is included for read-modify-write operations which can replace, complement, reset, or set the display memory contents. Also included is figure drawing hardware for drawing lines, arcs, circles, rectangles, and 8 by 8 pixel, graphics characters.

To drive a CRT display unit, a video sync timing generator provides highly flexible raster-scanning, partitioned display areas, zoomed display, panning, and scrolling. Also, a light pen input is included.

IV. IMPLEMENTATION HIGHLIGHTS

SLIDE 8
"BUS
REGISTERS"

Several details of the GDC's implementation deserve highlighting.

The microprocessor interface presents two read/write addresses to the system bus. Through these locations, the status register and the FIFO can be accessed. The read only status register is accessed at address Ø. GDC responses can be read from the FIFO at address 1. Both commands or parameters bytes are written into the FIFO, but must be differentiated by the address used for the write operation. Note that DMA data to and from display memory also passes through this interface but bypasses the FIFO.

SLIDE 9
"COMMAND
CATEGORIES"

Commands to the GDC can be categorized into five groups. Both the video timing and display formats can be controlled flexibly, while the figure drawing hardware has its own group of commands. Data from the display memory and the light pen and cursor addresses can be queried. DMA data transfers to and from display memory can be initiated with the DMA control commands. Any rectangular area in display memory can then be accessed via DMA.

SLIDE 10

"RMW CYCLE"

The display memory interface uses a time multiplexed address and data bus. A RMW cycle is shown here in-this slide. Based on the clock, the address is first output to the display memory address latches. The memory responds with the data read from that address. The modify operation is performed within the GDC, and the resulting data is output for writing back into memory. This process uses four clock cycles, which, with a 5 MHz clock, takes 800 ns. Each RMW cycle can draw one pixel of a figure or modify any specified bits of the 16 bit memory word. Note that during figure drawing, the address of the next pixel of the figure is being determined, while the current pixel is written into display memory. This pipelining capability allows the GDC to execute back-to-back RMW cycles while drawing.

SLIDE 11

"RMW OPERATION OPERATIONS"

Four operations are selectable for the RMW cycle. In this slide, notice on the left, the drawing pattern to be moved into display memory. Below this is shown the previous contents of memory in the area of the RMW operation. If the REPLACE operation is selected, a common write operation is performed. The other operations, COMPLEMENT, RESET, and SET, modify display memory only where the corresponding pattern bit is a "1".

SLIDE 12

"DISPLAY AREAS"

The display on the CRT can be divided into as many as four independent areas. Each separate area in display memory is scanned, in sequence, to generate the display. Each area is described to the GDC by a starting address and a length, which are stored in the on-chip RAM for access during the display scanning process. Independent scrolling and panning of each area are easily accomplished by changing the respective starting address.

V. APPLICATION EXAMPLE

To illustrate the use of the GDC, I will take as an example, a 1024 pixels by 1024 lines, color graphics system.

A good starting point for any CRT system design is the video field timing. Since the CRT monitor speed and the clock rate of the GDC are intimately related to the video timing, all must be considered together. Although the GDC can control a non-interlaced display of this size, available color CRT monitors will limit us to an interlaced 30 Hz display.

SLIDE 13
"VIDEO
TIMING"

Within these constraints, numerous parameters can be specified to the GDC to establish the video timing. The programmable active display, blanking and sync signal parameters will allow us to generate the appropriate video timing for our application.

SLIDE 14
"DISPLAY
MEMORY"

Another interesting point to highlight is the structure of the multiple plane, display memory which has four bits for each displayed pixel. In this example, each color output is controlled by one bit so that black, six colors, and white can be displayed. The fourth plane can be used to generate a white overlay. Each plane uses 16, 64K RAMs and a 16-bit video output shift register. During a RMW cycle, the 16 read data bits from the addressed plane are gated onto the address and data bus, and back to the GDC. Note, the address latch and write data buffer serve to de-multiplex the outgoing information from the GDC.

VI. FEATURES SUMMARY

SLIDE 15
"FLEXIBILITY"

Although the GDC fits this configuration well, many other system designs are equally realizable. Since the overall design goals were flexibility, performance, and simplicity of use, the scope of applications for the GDC is very broad.

Let me summarize some of the features which make this possible. Of the greatest importance to flexibility is the display memory architecture, which can be configured in almost any format up to a total of 256K, 16-bit words. Simultaneous support of bit-mapped graphics and coded characters in split field displays also extend the range of applications. These partitioned areas are easily and independently panned and scrolled. In addition, the display may be zoom magnified from one to 16 times, or the displayed image may act as a roving window in a larger display memory. The video pixel rate can exceed 80 MHz.

In character display applications, up to 256 characters per row or 100 rows per screen can be displayed while 64K characters and attributes can be stored in the 16-bit wide display memory.

For bit-mapped graphics, 4096 pixels per line or 2048 lines per screen can be displayed for a total of up to 4 megapixels. Hardware figure drawing needs only 800 ns per pixel for any of the GDC's figures, drawn in any direction. The DMA capability is ideal for text editing, hard-copy applications, and for saving and restoring the video memory's contents.

The light pen input is valuable for interactive applications, and multiple GDC's can be video synchronized for expanded systems.

VII. THE DIE

SLIDE 16
"PHOTO-
MICROGRAPH"

The embodiment of the GDC design is seen in this photomicrograph of the pre-production die. The 13,000 N-MOS transistor design was developed using 4 micron rules. This version is expected to run with a 5 MHz clock and uses a single 5 volt supply.

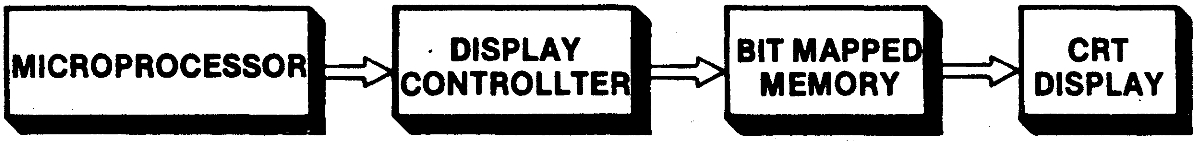
Taking a closer look at the die itself, we can see its functional elements. This block in the upper right corner contains the 16 byte Data RAM and the 16-level FIFO. The area directly below is the command processor control ROM holding 128, 14-bit instructions. Just to the left and splitting the die horizontally is the command processor logic. The microprocessor interface and the DMA control logic is bounded below by the light colored channel. Across the bottom of the die is the video sync generator. In the center of the die, above the sync generator, is the figure drawing hardware. The entire left side is the display memory interface, including the RMW data path circuitry.

VIII. CONCLUSION

As you can see, with this level of integration, the GDC represents a new generation of display controller technology. It makes practical graphics equipment suitable for a wide range of uses, from personal computers, through CAD systems, to sophisticated stand-alone graphics terminals.

THE NEED TO UNDERSTAND

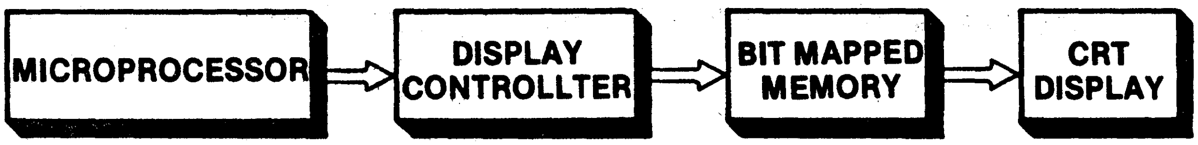
RASTER-SCAN GRAPHICS



GRAPHIC ACTIVITIES

1. **VECTOR LIST GENERATION**
2. **GRAPHICS PRIMITIVES DERIVATION**
3. **DRAWING PARAMETERS DETERMINATION**
4. **FIGURE DRAWING**
5. **DISPLAY MEMORY MANIPULATION**
6. **DISPLAY RASTER SCANNING**

DESIGN CONSIDERATIONS



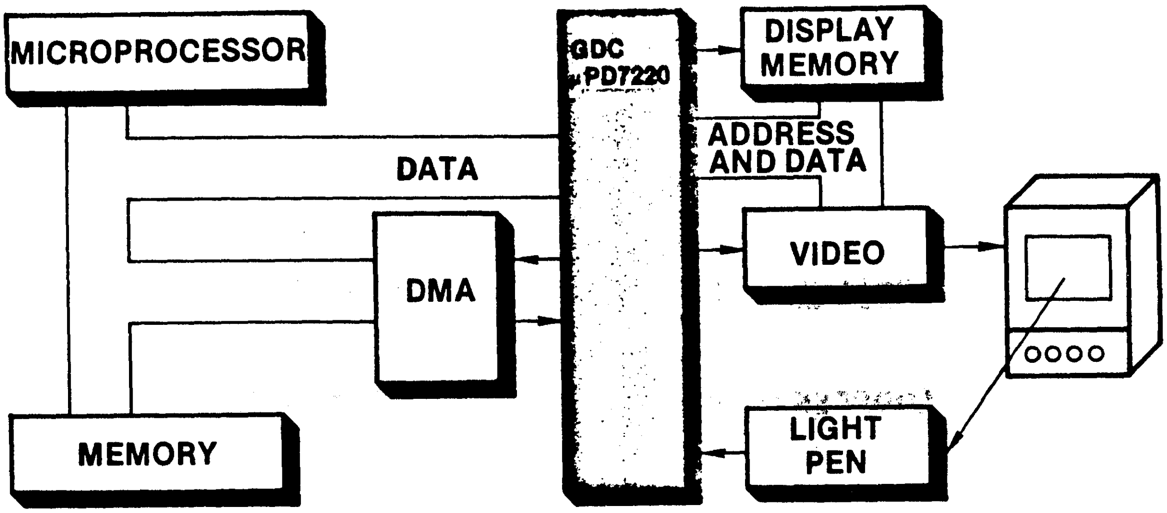
MICROPROCESSOR

DISPLAY CONTROLLTER

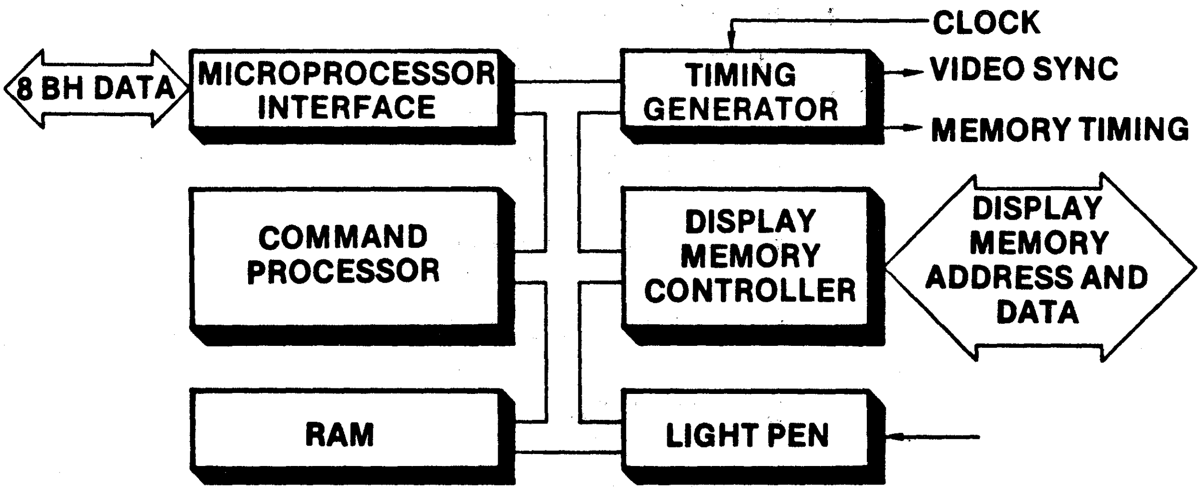
BIT MAPPED MEMORY

CRT DISPLAY

SYSTEM BLOCK DIAGRAM



GDC BLOCK DIAGRAM



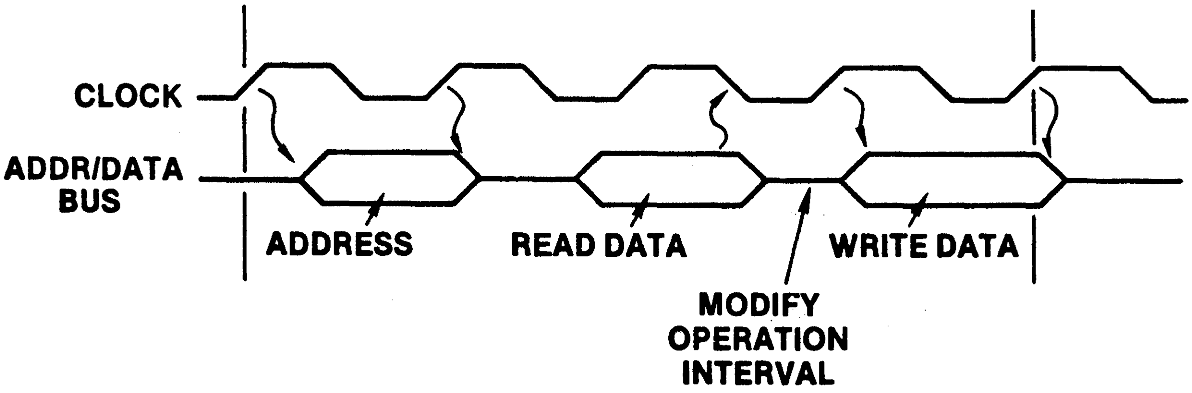
MICROPROCESSOR INTERFACE REGISTERS:

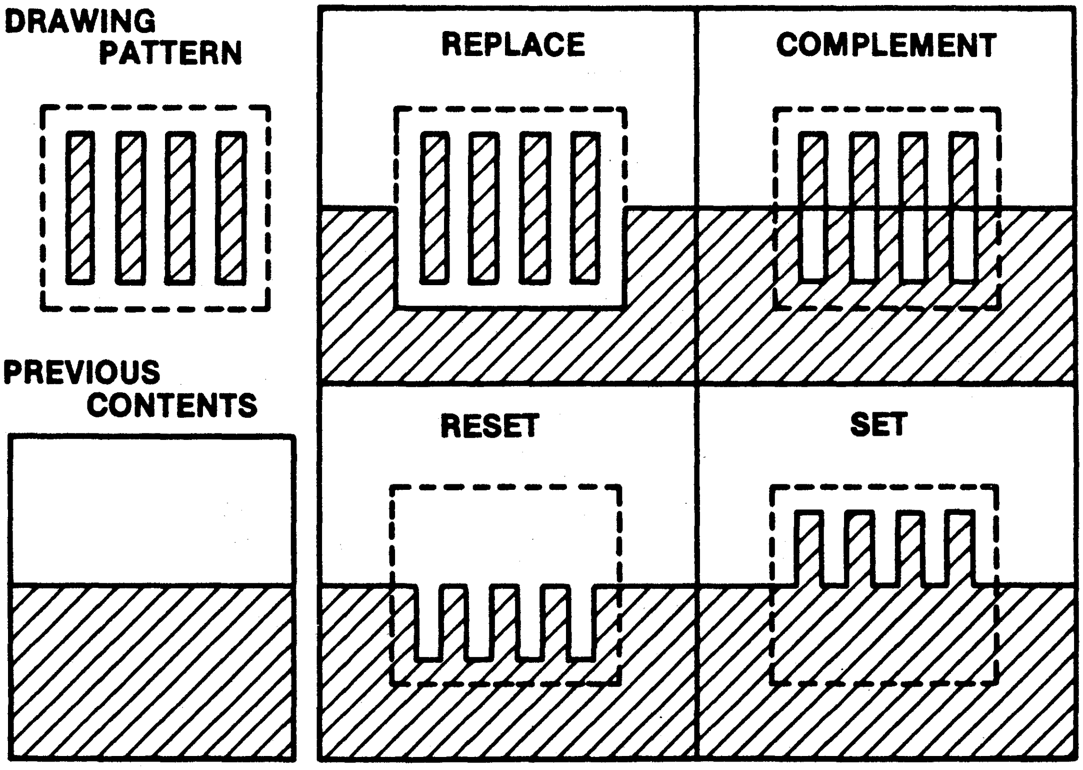
	READ	WRITE
ADDRESS 0:	STATUS REG	COMMAND
ADDRESS 1:	FIFO READ	PARAMETERS

COMMAND CATEGORIES:

1. VIDEO CONTROL
2. DISPLAY CONTROL
3. DRAWING CONTROL
4. DATA READ
5. DMA CONTROL

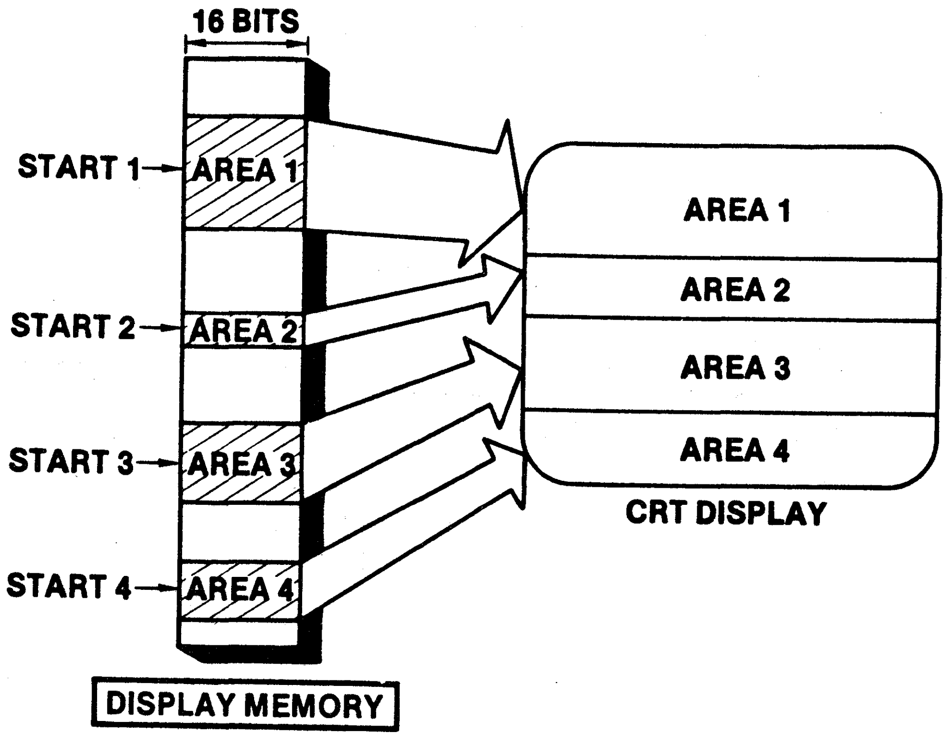
READ MODIFY WRITE CYCLE



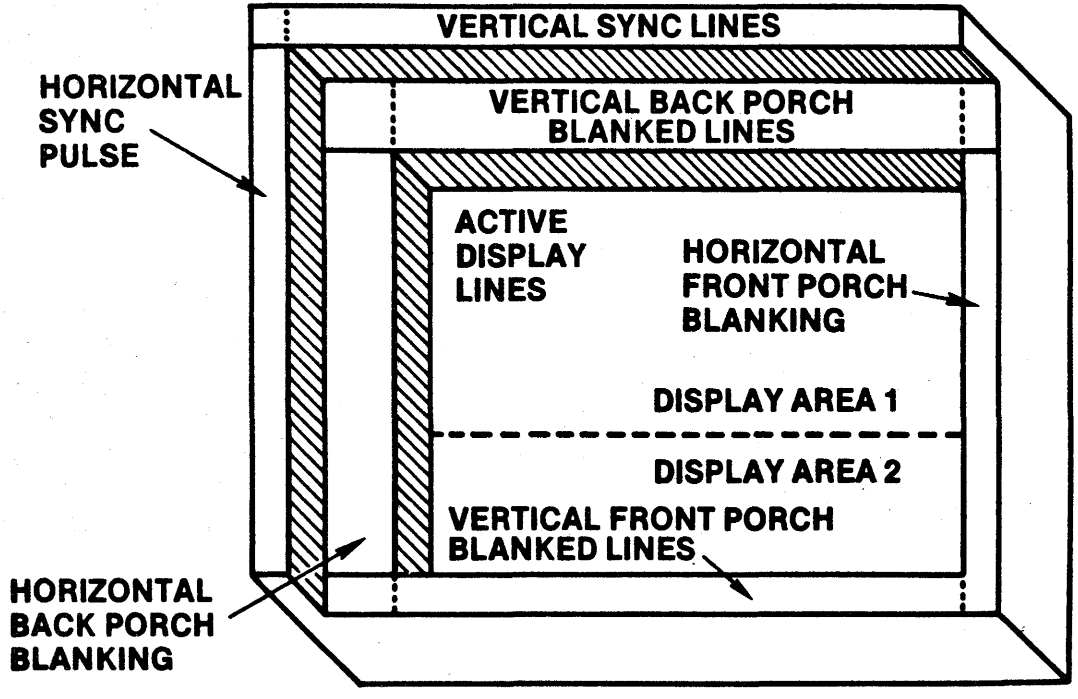


RMW OPERATIONS

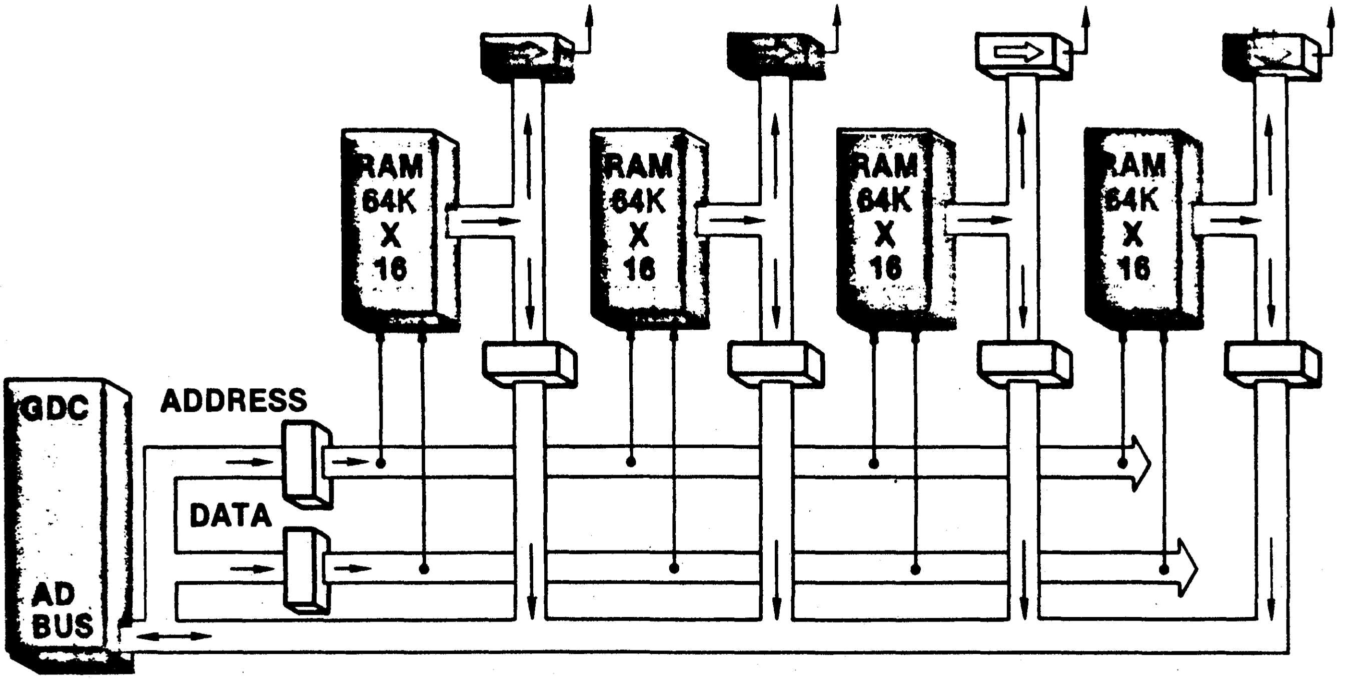
DISPLAY AREAS PARTITIONING



VIDEO FIELD TIMING


VERTICAL SYNC LINES
HORIZONTAL SYNC PULSE
VERTICAL BACK PORCH BLANKED LINES
ACTIVE DISPLAY LINES
HORIZONTAL FRONT PORCH BLANKING
DISPLAY AREA 1
DISPLAY AREA 2
VERTICAL FRONT PORCH BLANKED LINES
HORIZONTAL BACK PORCH BLANKING

DISPLAY MEMORY ORGANIZATION



FLEXIBILITY

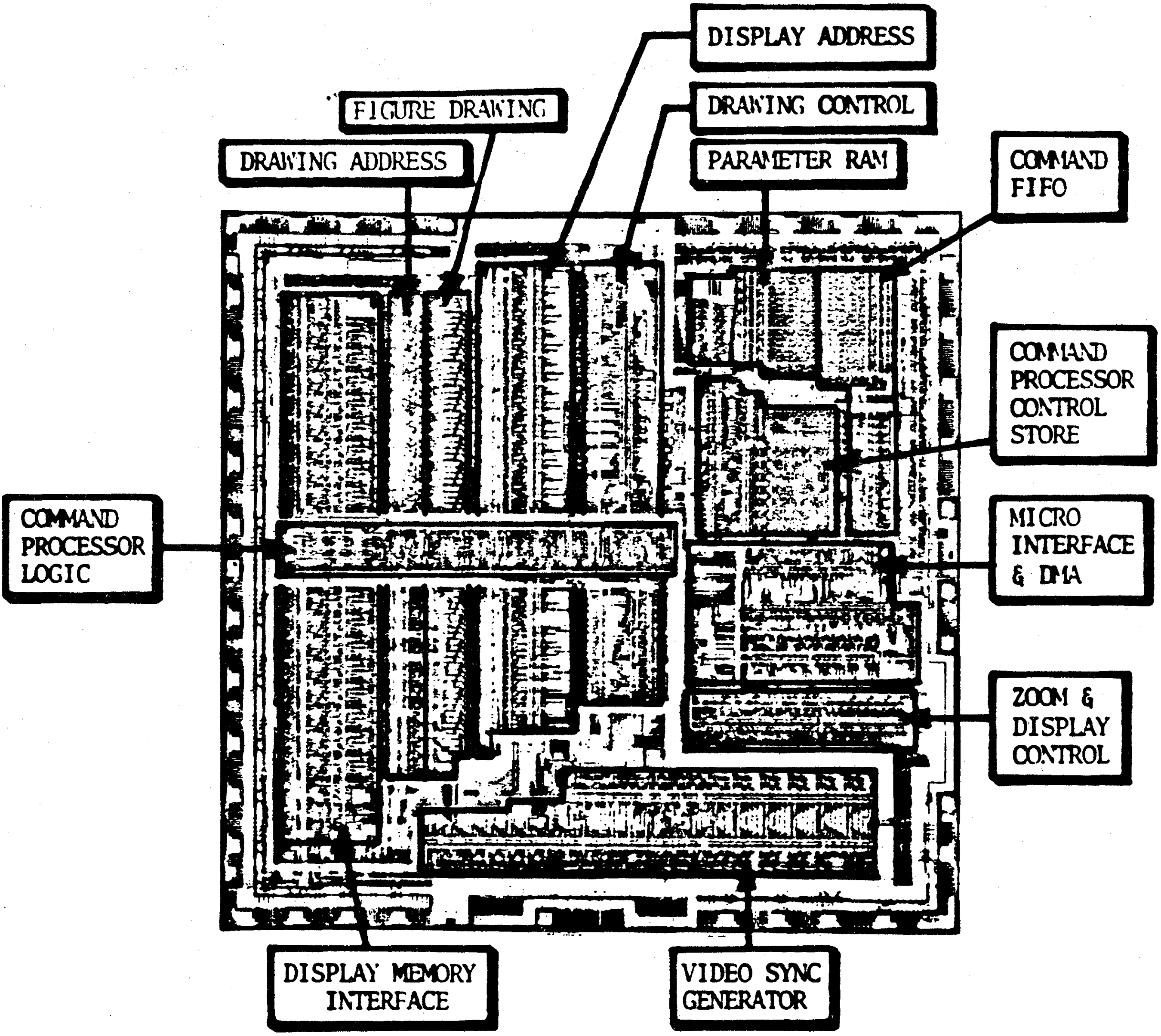
PERFORMANCE

SIMPLICITY

NEC uPD7220 "GDC"

Graphics Display Controller

Photomicrograph



NEC
NEC Microcomputers
A Division of NEC Electronics U.S.A. Inc.

μPD7220/GDC GRAPHICS DISPLAY CONTROLLER PRELIMINARY

Description

The μPD7220 Graphics Display Controller (GDC) is an intelligent microprocessor peripheral designed to be the heart of a high-performance raster-scan computer graphics and character display system. Positioned between the video display memory and the microprocessor bus, the GDC performs the tasks needed to generate the raster display and manage the display memory. Processor software overhead is minimized by the GDC's sophisticated instruction set, graphics figure drawing, and DMA transfer capabilities. The display memory supported by the GDC can be configured in any number of formats and sizes up to 256K 16-bit words. The display can be zoomed and panned, while partitioned screen areas can be independently scrolled. With its light pen input and multiple controller capability, the GDC is ideal for advanced computer graphics applications.

Features

□ Microprocessor Interface
DMA transfers with 8257 or 8237-type controllers
FIFO Command Buffering

□ Display Memory Interface
Up to 256K words of 16 bits
Read-Modify-Write (RMW) Display Memory cycles in under 800ns
Dynamic RAM refresh cycles for non-accessed memory

□ Light Pen Input

□ External video synchronization mode

□ Graphics Mode:
Four megabit, bit-mapped display memory

□ Character Mode:
8K character code and attributes display memory

□ Mixed Graphics and Characters Mode
64K if all characters
1 megapixel if all graphics

□ Graphics Capabilities:
Figure drawing of lines, arc/circles, rectangles, and graphics character in 800ns per pixel
Display 1024 by 1024 pixels with 4 planes of color or grayscale
Two independently scrollable areas

□ Character Capabilities:
Auto cursor advance
Four independently scrollable areas
Programmable cursor height
Characters per row: up to 256
Character rows per screen: up to 100

□ Video Display Format
Zoom magnification factors of 1 to 16
Panning
Command-settable video raster parameters

□ Technology
Single +5 volt, NMOS, 40-pin DIP

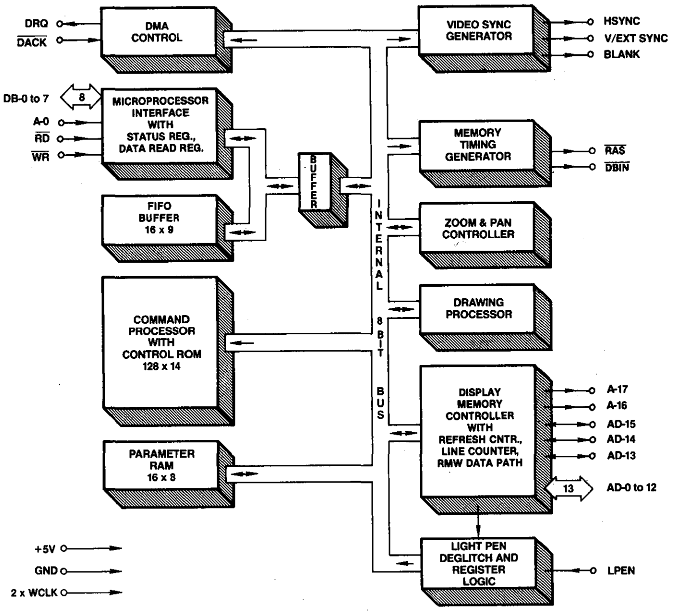
□ DMA Capability:
Bytes or word transfers
4 clock periods per byte transferred

System Considerations

The GDC is designed to work with a general purpose microprocessor to implement a high-performance computer graphics system. Through the division of labor established by the GDC's design, each of the system components is used to the maximum extent through six-level hierarchy of simultaneous tasks. At the lowest level, the GDC generates the basic video raster timing, including sync and blanking signals. Partitioned areas on the screen and zooming are also accomplished at this level. At the next level, video display memory is modified during the figure drawing operations and data moves. Third, display memory addresses are calculated pixel by pixel as drawing progresses. Outside the GDC at the next level, preliminary calculations are done to prepare drawing parameters. At the fifth level, the picture must be represented as a list of graphics figures drawable by the GDC. Finally, this representation must be manipulated, stored, and communicated. By handling the first three levels, the GDC takes care of the high-speed and repetitive tasks required to implement a graphics system.

GDC Components

The GDC block diagram illustrates how these tasks are accomplished.



Microprocessor Bus Interface

Control of the GDC by the system microprocessor is achieved through an 8-bit bi-directional interface. The status register is readable at any time. Access to the FIFO buffer is coordinated through flags in the status register and operates independently of the various internal GDC operations, due to the separate data bus connecting the interface and the FIFO buffer.

Command Processor

The contents of the FIFO are interpreted by the command processor. The command bytes are decoded, and the succeeding parameters are distributed to their proper destinations within the GDC. The command processor yields to the bus interface when both access the FIFO simultaneously.

DMA Control

The DMA control circuitry in the GDC coordinates transfers over the microprocessor interface when using an external DMA controller. The DMA Request and Acknowledge handshake lines directly interface with a μPD8257 or μPD8237 DMA controller, so that display data can be moved between the microprocessor memory and the display memory.

Parameter RAM

The 16-byte RAM stores parameters that are used repetitively during the display and drawing processes. In character mode, this RAM holds four sets of partitioned display area parameters; in graphics mode, the drawing pattern and graphics character take the place of two of the sets of parameters.

Video Sync Generator

Based on the clock input, the sync logic generates the raster timing signals for almost any interlaced, non-interlaced, or "repeat field" interlaced video format. The generator is programmed during the idle period following a reset. In video sync slave mode, it coordinates timing between multiple GDCs.

Memory Timing Generator

The memory timing circuitry provides two memory cycle types: a two-clock period refresh cycle and the read-modify-write (RMW) cycle which takes four clock periods. The memory control signals needed to drive the display memory devices are easily generated from the GDC's RAS and DBIN outputs.

Zoom & Pan Controller

Based on the programmable zoom display factor and the display area entries in the parameter RAM, the zoom and pan controller determines when to advance to the next memory address for display refresh and when to go on to the next display area. A horizontal zoom is produced by slowing down the display refresh rate while maintaining the video sync rates. Vertical zoom is accomplished by repeatedly accessing each line a number of times equal to the horizontal repeat. Once the line count for a display area is exhausted, the controller accesses the starting address and line count of the next display area from the parameter RAM. The system microprocessor, by modifying a display area starting address, can pan in any direction, independent of the other display areas.

Drawing Processor

The drawing processor contains the logic necessary to calculate the addresses and positions of the pixels of the various graphics figures. Given a starting point and the appropriate drawing parameters, the drawing processor needs no further assistance to complete the figure drawing.

Display Memory Controller

The display memory controller's tasks are numerous. Its primary purpose is to multiplex the address and data information in and out of the display memory. It also contains the 16-bit logic unit used to modify the display memory contents during RMW cycles, the character mode line counter, and the refresh counter for dynamic RAMs. The memory controller apportions the video field time between the various types of cycles.

Light Pen Deglitcher

Only if two rising edges on the light pen input occur at the same point during successive video fields are the pulses accepted as a valid light pen detection. A status bit indicates to the system microprocessor that the light pen register contains a valid address.

Programmer's View of GDC

The GDC occupies two addresses on the system microprocessor bus through which the GDC's status register and FIFO are accessed. Commands and parameters are written into the GDC's FIFO and are differentiated based on address bit A0. The status register or the FIFO can be read as selected by the address line.

A0	READ	WRITE
0	STATUS REGISTER	PARAMETER INTO FIFO
1	FIFO READ	COMMAND INTO FIFO

GDC Microprocessor Bus Interface Registers

Commands to the GDC take the form of a command byte followed by a series of parameter bytes as needed for specifying the details of the command. The command processor decodes the commands, unpacts the parameters, loads them into the appropriate registers within the GDC, and initiates the required operations.

The commands available in the GDC can be organized into five categories as described in the following section.

GDC Command Summary

Video Control Commands

1. RESET: Resets the GDC to its idle state and specifies the video display format.
2. VSYNC: Selects master or slave video synchronization mode.
3. CCHAR: Specifies the cursor and character row heights.

Display Control Commands

1. START: Starts the display scanning process.
2. ZOOM: Specifies zoom factors for the display and graphics characters writing.
3. CURS: Sets the position of the cursor in display memory.
4. PRAM: Defines starting addresses and lengths of the display areas and specifies the eight bytes for the graphics character.
5. PITCH: Specifies the width of the X dimension of display memory.

Drawing Control Commands

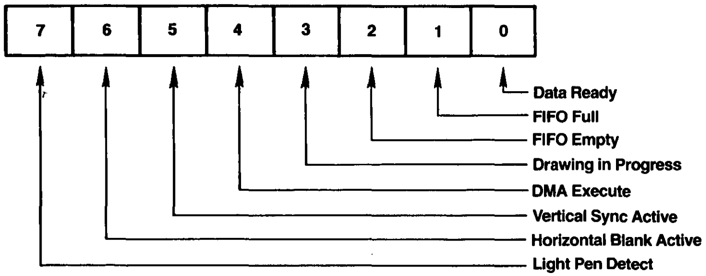
1. WDAT: Writes data words or bytes into display memory.
2. MASK: Sets the mask register contents.
3. FIGS: Specifies the parameters for the drawing processor.
4. FIGD: Draws the figure as specified above.
5. GCHRD: Draws the graphics character into display memory.

Data Read Commands

1. RDAT: Reads data words or bytes from display memory.
2. CURD: Reads the cursor position.
3. LPRD: Reads the light pen address.

DMA Control Commands

1. DMAR: Requests a DMA read transfer.
2. DMAW: Requests a DMA write transfer.



Status Register (SR)

Status Register Flags

SR-7: Light Pen Detect

When this bit is set to 1, the light pen address (LAD) register contains a deglitched value that the system microprocessor may read. This flag is reset after the 3-byte LAD is moved into the FIFO in response to the light pen read command.

SR-6: Horizontal Blanking Active

A 1 value for this flag signifies that horizontal retrace blanking is currently underway.

SR-5: Vertical Sync

Vertical retrace sync occurs while this flag is a 1. The vertical sync flag coordinates display format modifying commands to the blanked interval surrounding vertical sync. This eliminates display disturbances.

SR-4: DMA Execute

This bit is a 1 during DMA data transfers.

SR-3: Drawing in Progress

While the GDC is drawing a graphics figure, this status bit is a 1.

SR-2: FIFO Empty

This bit and the FIFO Full flag coordinate system microprocessor accesses with the GDC FIFO. When it is 1, the Empty flag ensures that all the commands and parameters previously sent to the GDC have been processed.

SR-1: FIFO Full

A 1 at this flag indicates a full FIFO in the GDC. A 0 ensures that there is room for at least one byte. This flag needs to be checked before each write into the GDC.

SR-0: Data Ready

When this flag is a 1, it indicates that a byte is available to be read by the system microprocessor. This bit must be tested before each read operation. It drops to a 0 while the data is transferred from the FIFO into the microprocessor interface data register.

FIFO Operation & Command Protocol

The first-in, first-out buffer (FIFO) in the GDC handles the command dialogue with the system microprocessor. This flow of information uses a half-duplex technique, in which the single 16-location FIFO is used for both directions of data movement, one direction at a time. The FIFO's direction is controlled by the system microprocessor through the GDC's command set. The microprocessor coordinates these transfers by checking the appropriate status register bits.

The command protocol used by the GDC requires the differentiation of the first byte of a command sequence from the succeeding bytes. This first byte contains the operation code and the remaining bytes carry parameters. Writing into the GDC causes the FIFO to store a flag value alongside the data byte to signify whether the byte was written into the command or the parameter address. The command processor in the GDC tests this bit as it interprets the entries in the FIFO.

The receipt of a command byte by the command processor marks the end of any previous operation. The number of parameter bytes supplied with a command is cut short by the receipt of the next command byte. A read operation from the GDC to the microprocessor can be terminated at any time by the next command.

The FIFO changes direction under the control of the system microprocessor. Commands written into the GDC always put the FIFO into write mode if it wasn't in it already. If it was in read mode, any read data in the FIFO at the time of the turnaround is lost. Commands which require a GDC response, such as RDAT CURD and LPRD, put the FIFO into read mode after the command is interpreted by the GDC's command processor. Any commands and parameters behind the read-evoking command are discarded when the FIFO direction is reversed.

Read-Modify-Write Cycle

Data transfers between the GDC and the display memory are accomplished using a read-modify-write (RMW) memory cycle. The four clock period timing of the RMW cycle is used to: 1) output the address, 2) read data from the memory, 3) modify the data, and 4) write the modified data back into the initially selected memory address. This type of memory cycle is used for all interactions with display memory including DMA transfers, except for the two clock period display and RAM refresh cycles.

The operations performed during the modify portion of the RMW cycle merit additional explanation. The circuitry in the GDC uses three main elements: the Pattern register, the Mask register, and the 16-bit Logic Unit. The Pattern register holds the data pattern to be moved into memory. It is loaded by the WDAT command or, during drawing, from the parameter RAM. The Mask register contents determine which bits of the read data will be modified. Based on the contents of these registers, the Logic Unit performs the selected operations of REPLACE, COMPLEMENT, SET, or CLEAR on the data read from display memory.

The Pattern register contents are ANDed with the Mask register contents to enable the actual modification of the memory read data, on a bit-by-bit basis. For graphics drawing, one bit at a time from the Pattern register is combined with the Mask. When ANDed with the bit set to a 1 in the Mask register, the proper single pixel is modified by the Logic Unit. For the next pixel in the figure, the next bit in the Pattern register is selected and the Mask register bit is moved to identify the pixel's location within the word. The Execution word address pointer register, EAD, is also adjusted as required to address the word containing the next pixel.

In character mode, all of the bits in the Pattern register are used in parallel to form the respective bits of the modify data word. Since the bits of the character code word are used in parallel, unlike the one-bit-at-a-time graphics drawing process, this facility allows any or all of the bits in a memory word to be modified in one RMW memory cycle. The Mask register must be loaded with 1s in the positions where modification is to be permitted.

The Mask register can be loaded in either of two ways. In graphics mode, the CURS command contains a four-bit dAD field to specify the dot address. The command processor converts this parameter into the one-of-16 format used in the Mask register for figure drawing. A full 16 bits can be loaded into the Mask register using the MASK command. In addition to the character mode use mentioned above, the 16-bit MASK load is convenient in graphics mode when all of the pixels of a word are to be set to the same value.

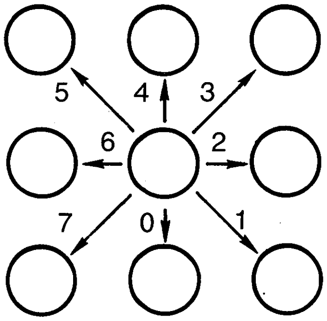
The Logic Unit combines the data read from display memory, the Pattern Register, and the Mask register to generate the data to be written back into display memory. Any one of four operations can be selected: REPLACE, COMPLEMENT, CLEAR or SET. In each case, if the respective Mask bit is 0, that particular bit of the read data is returned to memory unmodified. If the Mask bit is 1, the modification is enabled. With the REPLACE operation, the modify data simply takes the place of the read data for modification enabled bits. For the other three operations, a 0 in the modify data allows the read data bit to be returned to memory. A 1 value causes the specified operation to be performed in the bit positions with set Mask bits.

Figure Drawing

The GDC draws graphics figures at the rate of one pixel per read-modify-write (RMW) display memory cycle. These cycles take four clock periods to complete. At a clock frequency of 5MHZ, this is equal to 800ns. During the RMW cycle the GDC simultaneously calculates the address and position of the next pixel to be drawn.

The graphics figure drawing process depends on the display memory addressing structure. Groups of 16 horizontally adjacent pixels form the 16-bit words which are handled by the GDC. Display memory is organized as a linearly addressed space of these words. Addressing of individual pixels is handled by the GDC's internal RMW logic.

During the drawing process, the GDC finds the next pixel of the figure which is one of the eight nearest neighbors of the last pixel drawn. The GDC assigns each of these eight directions a number from 0 to 7, starting with straight down and proceeding counterclockwise.



Drawing Directions

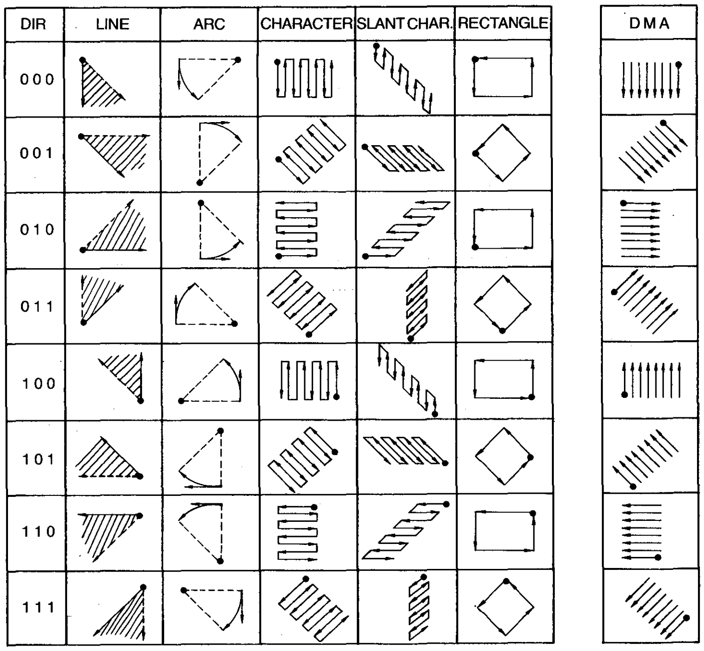
Figure drawing requires the proper manipulation of the address and the pixel bit position according to the drawing direction to determine the next pixel of the figure. To move to the word above or below the current one, it is necessary to subtract or add the number of words per line in display memory. This parameter is called the pitch. To move to the word to either side, the Execute word address cursor, EAD, must be incremented or decremented as the dot address pointer bit reaches the LSB or the MSB of the Mask register. To move to a pixel within the same word, it is necessary to rotate the dot address pointer register to the right or left. The table below summarizes these operations for each direction.

DIR	OPERATIONS TO ADDRESS THE NEXT PIXEL
000	EAD + P → EAD
001	EAD + P → EAD dAD (MSB) = 1 : EAD + 1 → EAD dAD → LR
010	dAD (MSB) = 1 : EAD + 1 → EAD dAD → LR
011	EAD − P → EAD dAD (MSB) = 1 : EAD + 1 → EAD dAD → LR
100	EAD − P → EAD
101	EAD − P → EAD dAD (LSB) = 1 : EAD − 1 → EAD dAD → RR
110	dAD (LSB) = 1 : EAD + 1 → EAD dAD → RR
111	EAD + P → EAD dAD (LSB) = 1 : EAD + 1 → EAD dAD → RR

Where P = Pitch, LR = Left Rotate, RR = Right Rotate
EAD = Execute Word Address
dAD = Dot Address stored in the Mask Register

Whole word drawing is useful for filling areas in memory with a single value. By setting the Mask register to all 1s with the MASK command, both the LSB and MSB of the dAD will always be 1, so that the EAD value will be incremented or decremented for each cycle regardless of direction. One RMW cycle will be able to effect all 16 bits of the word for any drawing type. One bit in the Pattern register is used per RMW cycle to write all the bits of the word to the same value. The next Pattern bit is used for the word, etc.

For the various figures, the effect of the initial direction upon the resulting drawing is shown below:



Drawing Parameters

In preparation for graphics figure drawing, the GDC's Drawing Processor needs the figure type, direction and drawing parameters, the starting pixel address, and the pattern from the microprocessor. Once these are in place within the GDC, the Figure Draw command, FIGD, initiates the drawing operation. From that point on, the system microprocessor is not involved in the drawing process. The GDC Drawing Processor coordinates the RMW circuitry and address registers to draw the specified figure pixel by pixel.

The algorithms used by the processor for figure drawing are designed to optimize its drawing speed. To this end, the specific details about the figure to be drawn are reduced by the microprocessor to a form conducive to high-speed address calculations within the GDC. In this way the repetitive, pixel-by-pixel calculations can be done quickly, thereby minimizing the overall figure drawing time. The table below summarizes the parameters.

	DC	D	D2	D1	DM
Initial Value	0	8	8	−1	−1
Line	ΔX	2|ΔY|−|ΔX|	2|ΔY|−2|ΔX|	2|ΔY|	-
Circle	r/√2↑	r-1	2(r-1)	−1	0
Arc	r/√2↑	r-1	2(r-1)	−1	r sin Θ ↓
Rectangle	3	A*	B*	−1	A*
Character	7	-	-	-	-
Area Fill	B**	A**	-	-	-
DMA	L	I	-	-	-
R/W DAT	W	-	-	-	-

Graphics Figure Drawing Parameters

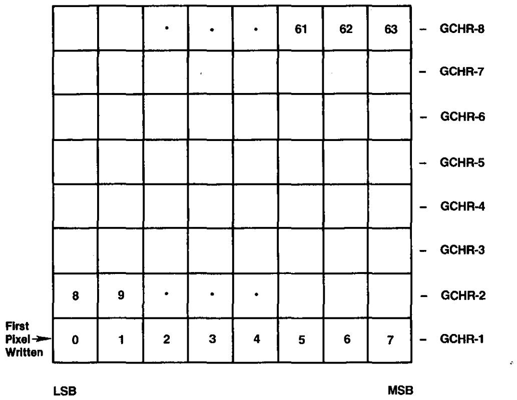
WHERE:

ΔX = # of pixel positions in the line horizontally.
ΔY = # of pixel positions in the line vertically.
r = Radius of the curve, in pixels.
↑ = Rounded up to the next higher integer.
Θ = The angle subtended from the octant axis to the arc.
A* = The length − 1, in pixels, in the direction initially specified.
B* = The length − 1, in pixels, in the direction at right angles to the initial direction.
A** = # of pixels in the initial direction.
B** = # of pixels in the right angle direction − 1.
L = # of word address in the perpendicular direction for DMA.
I = # of DMA transfers in the initially specified direction.
W = # of word addresses to be written into.
DC = Drawing Count.
DM = Dots Masked Count.

Graphics Character Drawing

Graphics characters can be drawn into display memory pixel-by-pixel. The 8-by-8 character is loaded into the GDC's parameter RAM by the system microprocessor. Consequently, there are no limitations on the character set used. By varying the drawing parameters and drawing direction, numerous drawing options are available. In area fill applications, a character can be written into display memory as many times as desired without reloading the parameter RAM.

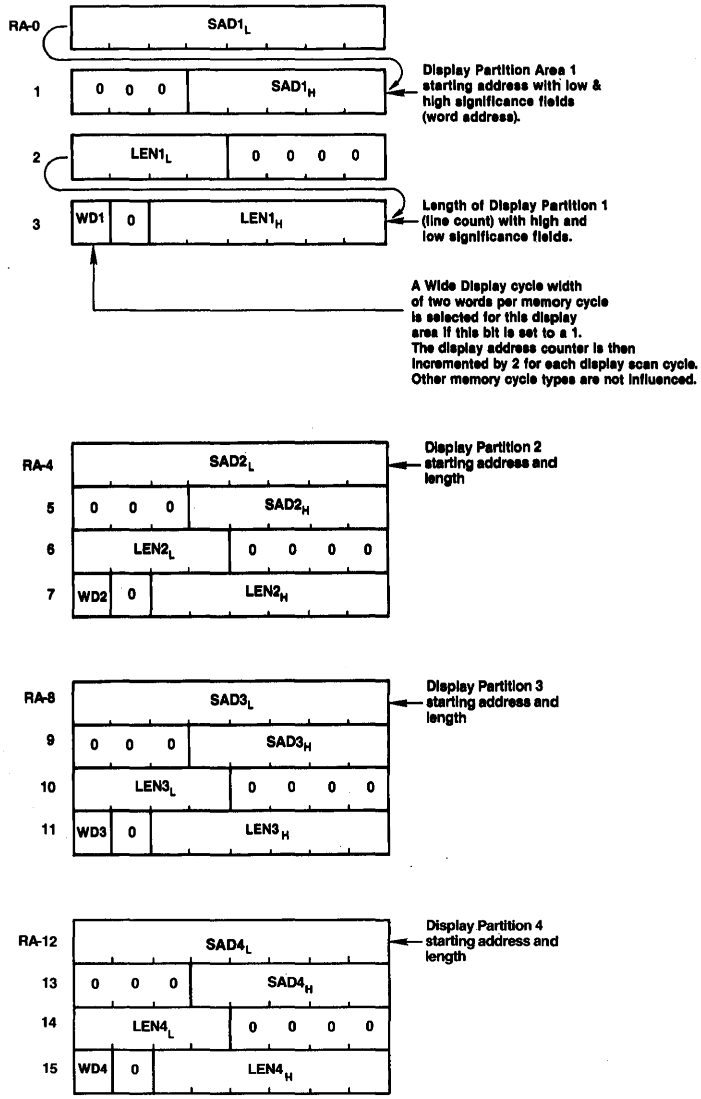
Once the parameter RAM has been loaded with the eight graphics character bytes by the appropriate PRAM command, the GCHRD command can be used to draw the eight bytes into display memory starting at the cursor. The zoom magnification factor for writing, set by the zoom command, controls the size of the character written into the display memory in integer multiples of 1 through 16. The bit values in the PRAM are repeated horizontally and vertically the number of times specified by the zoom factor. As seen in display memory, the writing is started in the lower left corner of the eventual 8-by-8 block and proceeds right to left and bottom to top. The LSBs of the stored character bytes are written first, and parameter RAM location 15 is the first byte to go out.



Display Memory

Parameter Ram Contents

Character Mode: RAM Address RA 0 to 16



Display Partition Area 1 starting address with low & high significance fields (word address).

Length of Display Partition 1 (line count) with high and low significance fields.

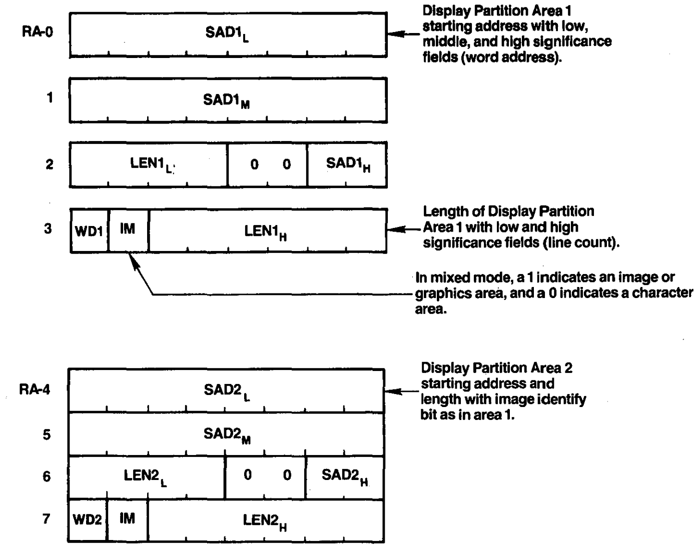
A Wide Display cycle width of two words per memory cycle is selected for this display area if this bit is set to a 1. The display address counter is then incremented by 2 for each display scan cycle. Other memory cycle types are not influenced.

Display Partition 2 starting address and length

Display Partition 3 starting address and length

Display Partition 4 starting address and length

Graphics and Mixed Graphics and Characters Mode

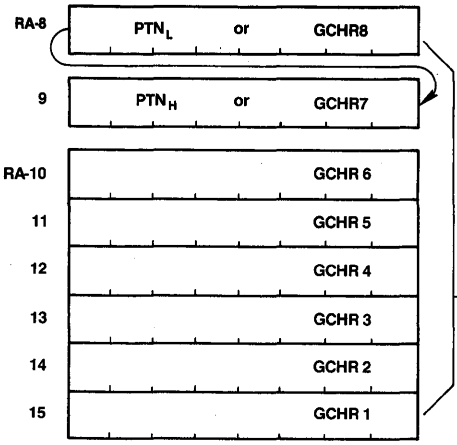


Display Partition Area 1 starting address with low, middle, and high significance fields (word address).

Length of Display Partition Area 1 with low and high significance fields (line count).

In mixed mode, a 1 indicates an image or graphics area, and a 0 indicates a character area.

Display Partition Area 2 starting address and length with image identify bit as in area 1.



Pattern of 16 bits used for figure drawing to pattern dotted, dashed, etc. lines.

Graphics character bytes to be moved into display memory with graphics character drawing

Command Bytes Summary

RESET:	0	0	0	0	0	0	0	0
VSYNC:	0	1	1	0	1	1	1	M
CCHAR:	0	1	0	0	1	0	1	1
START:	0	1	1	0	1	0	1	1
ZOOM:	0	1	0	0	0	1	1	0
CURS:	0	1	0	0	1	0	0	1
PRAM:	0	1	1	1	SA			
PITCH:	0	1	0	0	0	1	1	1
WDAT:	0	0	1	TYPE		0	MOD	
MASK:	0	1	0	0	1	0	1	0
FIGS:	0	1	0	0	1	1	0	0
FIGD:	0	1	1	0	1	1	0	0
GCHRD:	0	1	1	0	1	0	0	0
RDAT:	1	0	1	TYPE		0	0	0
CURD:	1	1	1	0	0	0	0	0
LPRD:	1	1	0	0	0	0	0	0
DMAR:	1	0	1	TYPE		1	0	0
DMAW:	0	0	1	TYPE		1	MOD	

Video Control Commands

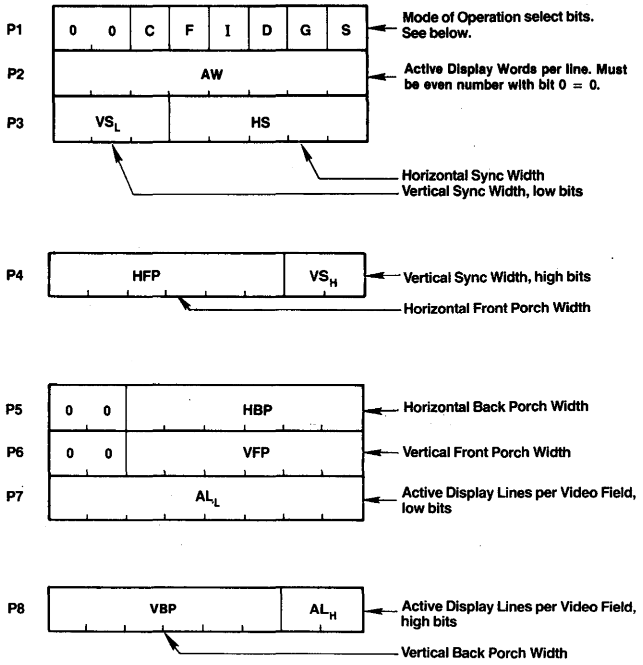
Reset

RESET:	0	0	0	0	0	0	0	0

Blank the display, enter idle mode, and initialize within the GDC:
- FIFO
- Command Processor
- Internal Counters

This command can be executed at any time and does not modify any of the parameters already loaded into the GDC.

If followed by parameter bytes, this command also sets the sync generator parameters as described below.
Idle mode is exited with the START command.


P1
0 0 C F I D G S
Mode of Operation select bits. See below.
P2
AW
Active Display Words per line. Must be even number with bit 0 = 0.
P3
VS_L
HS
Horizontal Sync Width
Vertical Sync Width, low bits
P4
HFP
VS_H
Vertical Sync Width, high bits
Horizontal Front Porch Width
P5
0 0
HBP
Horizontal Back Porch Width
P6
0 0
VFP
Vertical Front Porch Width
P7
AL_L
Active Display Lines per Video Field, low bits
P8
VBP
AL_H
Active Display Lines per Video Field, high bits
Vertical Back Porch Width

In graphics mode, a word is a group of 16 pixels. In character mode, a word is one character code and its attributes, if any.
The number of active words per line must be an even number from 2 to 256.
An all-zero parameter value selects a count equal to 2^n where n=number of bits in the parameter field.
All horizontal widths are counted in display words.
All vertical widths are counted in lines.

Modes of Operation Bits

C	G	Display Mode
0	0	Mixed Graphics & Character
0	1	Graphics Mode
1	0	Character Mode
1	1	Invalid

I	S	Video Framing
0	0	Non-Interlaced
0	1	Invalid
1	0	Interlaced Repeat Field for Character Displays
1	1	Interlaced

Repeat Field Framing: 2 Field Sequence with ½ line offset between otherwise identical fields.
Interlaced Framing: 2 Field Sequence with ½ line offset. Each field displays alternate lines.
Non-interlaced Framing: 1 field brings all of the information to the screen.

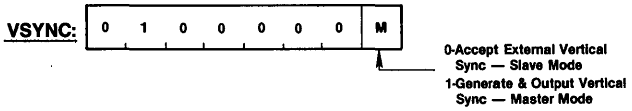
D	Dynamic RAM Refresh Cycles Enable
0	No Refresh — STATIC RAM
1	Refresh — Dynamic RAM

Dynamic RAM refresh is important when high display zoom factors or DMA are used in such a way that not all of the rows in the RAMs are regularly accessed during display raster generation and for otherwise inactive display memory.

F	Drawing Time Window
0	Drawing during active display time and retrace blanking
1	Drawing only during retrace blanking

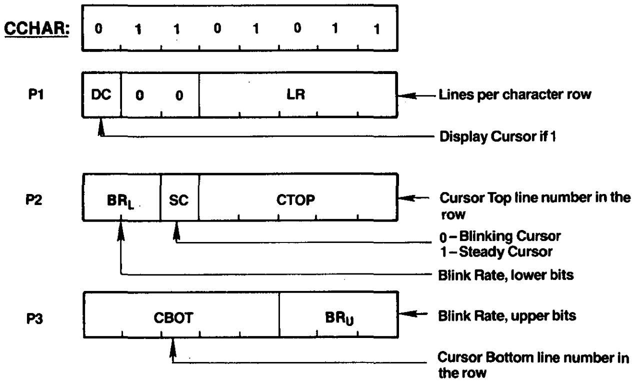
Access to display memory can be limited to retrace blanking intervals only, so that no disruptions of the image are seen on the screen.

Vertical Sync Mode


VSYNC:
0 1 0 0 0 0 0 M
0-Accept External Vertical Sync — Slave Mode
1-Generate & Output Vertical Sync — Master Mode

When using two or more GDCs to contribute to one image, one GDC is defined as the master sync generator, and the others operate as its slaves. The VSYNC pins of all GDCs are connected together.

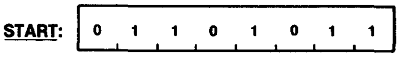
Cursor & Character Characteristics



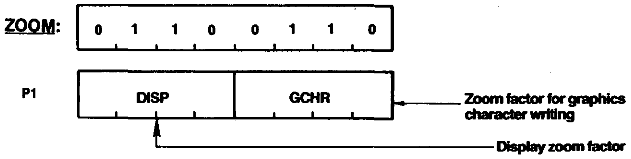
In graphics mode, LR should be set to 1.
The blink rate parameter controls both the cursor and attribute blink rates. The cursor blink-on time = blink-off time = 2 x BR (video frames). The attribute blink rate is always ½ the cursor rate but with a ¾ on-¼ off duty cycle.

Display Control Commands

Start Display & End Idle Mode

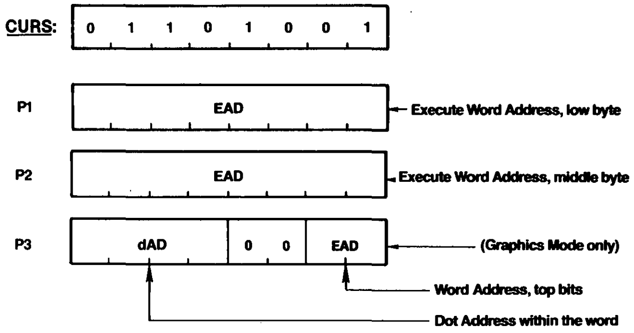


Zoom Factors Specify



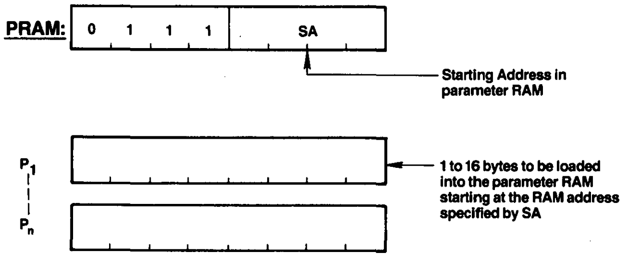
Zoom magnification factors of 1 through 16 are available using codes 0 through 15, respectively.

Cursor Position Specify


CURS:
0 1 1 0 1 0 0 1
P1
EAD
Execute Word Address, low byte
P2
EAD
Execute Word Address, middle byte
P3
dAD 0 0 EAD
(Graphics Mode only)
Word Address, top bits
Dot Address within the word

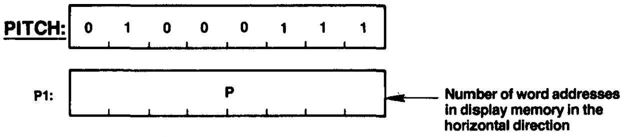
In character mode, the third parameter byte is not needed. The cursor is displayed for the word time in which the display scan address (DAD) equals the cursor address.
In graphics mode, the cursor word address specifies the word containing the starting pixel of the drawing; the dot address value specifies the pixel within that word.

Parameter RAM Load


PRAM:
0 1 1 1 SA
Starting Address in parameter RAM
P1
Pn
1 to 16 bytes to be loaded into the parameter RAM starting at the RAM address specified by SA

From the starting address, SA, any number of bytes may be loaded into the parameter RAM at incrementing addresses, up to location 15. The sequence of parameter bytes is terminated by the next command byte entered into the FIFO. The parameter RAM stores 16 bytes of information in predefined locations which differ for graphics and character modes. See the parameter RAM discussion for bit assignments.

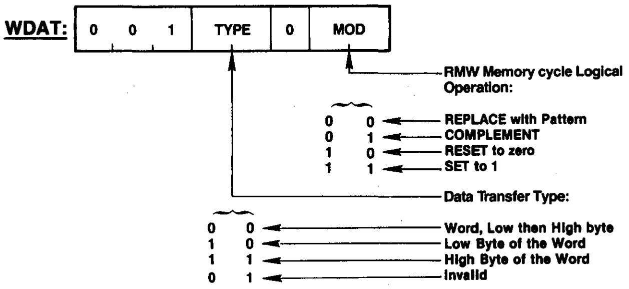
Pitch Specification

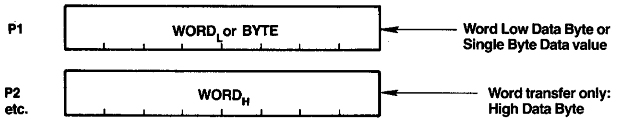

PITCH:
1 0 0 0 1 1 1 (0 1 0 0 0 1 1 1)
P1:
P
Number of word addresses in display memory in the horizontal direction

This value is used during drawing by the drawing processor to find the word directly above or below the current word, and during display to find the start of the next line.

Drawing Control Commands

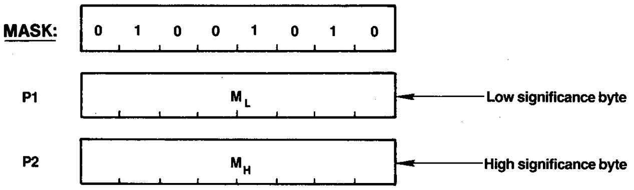
Write Data into Display Memory




P1
WORD$_L$ or BYTE
Word Low Data Byte or Single Byte Data value
P2 etc.
WORD$_H$
Word transfer only: High Data Byte

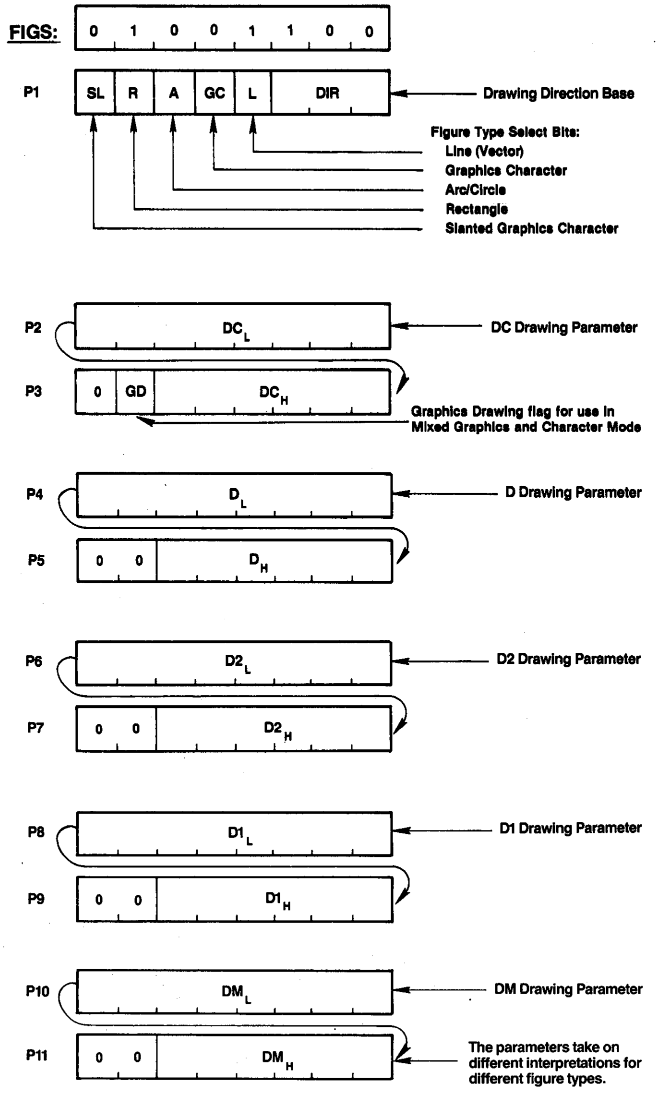
Upon receiving a set of parameters (two bytes for a Word transfer, one for a byte transfer), one RMW cycle into Video Memory is done at the address pointed to by the cursor EAD. The EAD pointer is advanced to the next word, according to the previously specified direction. More parameters can then be accepted.
For byte writes, the unspecified byte is treated as all zeros during the RMW memory cycle.

Mask Register Load



This command sets the value of the 16-bit Mask register of the figure drawing processor. The Mask register controls which bits can be modified in the display memory during a read-modify-write cycle.

Figure Drawing Parameters Specify

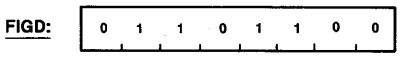


Valid Figure Type Select Combinations

SL	R	A	GC	L	Operation
0	0	0	0	0	Character Display Mode Drawing and Individual Dot Drawing
0	0	0	0	1	Straight Line Drawing
0	0	0	1	0	Graphics Character Drawing and Area filling with graphics character pattern
0	0	1	0	0	Arc and Circle Drawing
0	1	0	0	0	Rectangle Drawing
1	0	0	1	0	Slanted graphics character drawing and slanted area filling

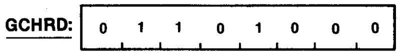
Only these bit combinations assure correct drawing operation.

Figure Draw Start



On execution of this instruction, the GDC loads the parameters from the parameter RAM into the drawing processor and starts the drawing process at the pixel pointed to by the cursor, EAD, and the dot address, dAD.

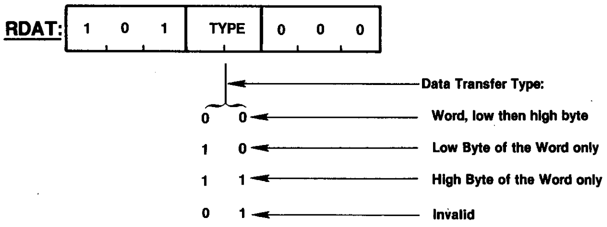
Graphics Character Draw Start



Based on parameters loaded with the FIGS command, this command initiates the drawing of the graphics character or area filling pattern stored in Parameter RAM. Drawing begins at the address in display memory pointed to by the EAD and dAD values.

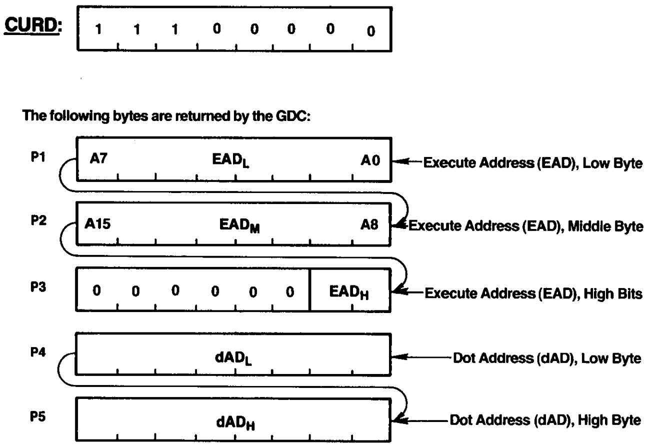
Data Read Commands

Read Data from Display Memory



Using the DIR and DC parameters of the FIGS command to establish direction and transfer count, multiple RMW cycles can be executed without specification of the cursor address after the initial load (DC = number of words or bytes + 1).
As this instruction begins to execute, the FIFO buffer direction is reversed so that the data read from display memory can pass to the microprocessor. Any commands or parameters in the FIFO at this time will be lost. A command byte sent to the GDC will immediately reverse the buffer direction back to write mode, and all RDAT information not yet read from the FIFO will be lost.

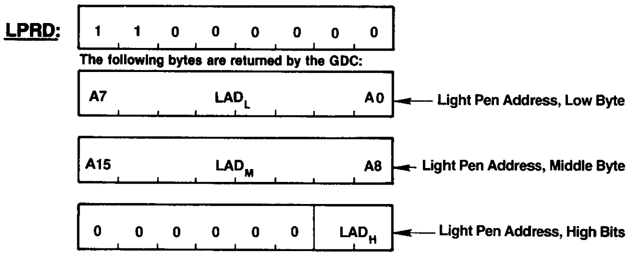
Cursor Address Read


CURD:
1 1 1 0 0 0 0 0
The following bytes are returned by the GDC:
P1
A7
EAD_L
A0
Execute Address (EAD), Low Byte
P2
A15
EAD_M
A8
Execute Address (EAD), Middle Byte
P3
0 0 0 0 0 0
EAD_H
Execute Address (EAD), High Bits
P4
dAD_L
Dot Address (dAD), Low Byte
P5
dAD_H
Dot Address (dAD), High Byte

The Execute Address, EAD, points to the display memory word containing the pixel to be addressed.

The Dot Address, dAD, within the word is represented as a 1-of-16 code for graphics drawing operations.

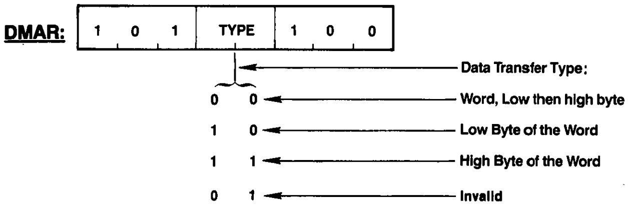
Light Pen Address Read


LPRD:
1 1 0 0 0 0 0 0
The following bytes are returned by the GDC:
A7
LAD_L
A0
Light Pen Address, Low Byte
A15
LAD_M
A8
Light Pen Address, Middle Byte
0 0 0 0 0 0
LAD_H
Light Pen Address, High Bits

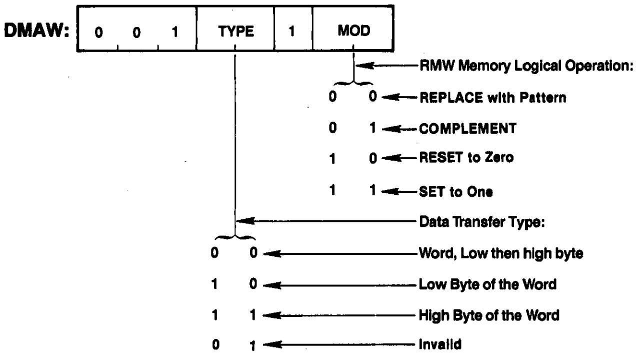
The light pen address, LAD, corresponds to the display word address, DAD, at which the light pen input signal is detected and deglitched.

The light pen may be used in graphics, character, or mixed modes but only indicates the word address of light pen position.

DMA Read Request



DMA Write Request



Absolute Maximum Ratings* (Tentative)

Ambient Temperature under Bias	0°C to 70°C
Storage Temperature	−65°C to 150°C
Voltage on any Pin with respect to Ground	−0.5V to +7V
Power Dissipation	1.5 Watt

*COMMENT: Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

T_A=0°C to 70°C; Vcc=5V ± 10%

PARAMETER	SYMBOL	MIN	MAX	UNITS	TEST CONDITIONS
Input Low Voltage	V_{IL}	−0.5	0.8	V	
Input High Voltage	V_{IH}	2.0	Vcc + 0.5	V	
Output Low Voltage	V_{OL}		0.45	V	I_{OL} = 2.2 mA
Output High Voltage	V_{OH}	2.4		V	I_{OH} = −400 uA
Input Low Leak Current	I_{IL}		−10	uA	V_I = 0V
Input High Leak Current	I_{IH}		+10	uA	V_I = Vcc
Output Low Leak Current	I_{OL}		−10	uA	V_O = 0V
Output High Leak Current	I_{OH}		+10	uA	V_O = Vcc
Clock Input Low Voltage	V_{CL}	−0.5	0.6	V	
Clock Input High Voltage	V_{CH}	3.9	Vcc + 1.0	V	
Vcc Supply Current	I_{CC}		270	mA	

Capacitance

T_A=25°C; Vcc=GND=OV

PARAMETER	SYMBOL	MIN	MAX	UNITS	TEST CONDITIONS
Input Capacitance	C_{IN}		10	pF	fc = 1 MHz
I/O Capacitance	$C_{I/O}$		20	pF	V_I (unmeasured) = 0V

AC Characteristics

TA=0°C to 70°C; Vcc=5V ± 10%

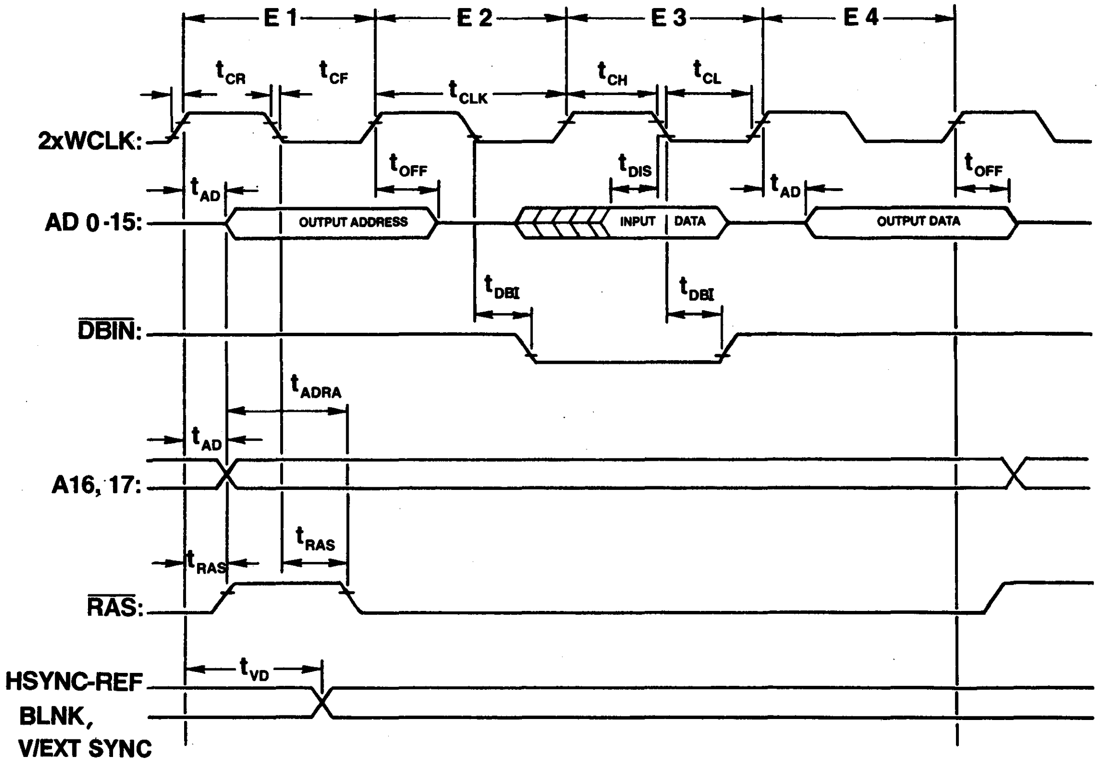
PARAMETER	SYMBOL	MIN	MAX	UNITS	TEST CONDITIONS
Clock Cycle Period	t_{CLK}	167	2000	nS	C_L = 100 pF
Clock High Time	t_{CH}	70		nS	
Clock Low Time	t_{CL}	70		nS	
Clock Rise Time	t_{CR}		10	nS	
Clock Fall Time	t_{CF}		10	nS	
DMA Cycle Time	t_E	$4xt_{CLK}$		nS	
Data In Set Up Time	t_{DIS}	40		nS	
Data Set Up Time for $\overline{WR}$	t_{DW}	150		nS	
Data Hold Time for $\overline{WR}$	t_{WD}	0		nS	
$\overline{DACK}$ High Level Width	t_{DK}	t_{CLK}		nS	

Timing Requirements

PARAMETER	SYMBOL	MIN	MAX	UNITS	TEST CONDITIONS
Address Output Delay	t_{AD}	20	80	nS	C_L = 100 pF
Data Bus In Delay	t_{DBI}		80	nS	
Address Active to $\overline{RAS}$ Low	t_{ADRA}	60		nS	
Video Signal Delay	t_{VD}		120	nS	
$\overline{RD}$ Low to DB Delay	t_{RD}		120	nS	
$\overline{RD}$ High to DB Floating	t_{DF}		120	nS	
$\overline{DACK}$ Low to DRD Delay	t_{DR}		120	nS	
$\overline{RAS}$ Output Delay	t_{RAS}		80	nS	
AD Floating Delay	t_{OFF}		80	nS	

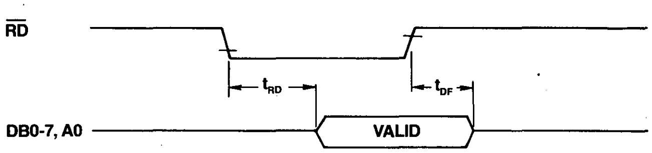
Timing Responses

Display Memory Bus Timing

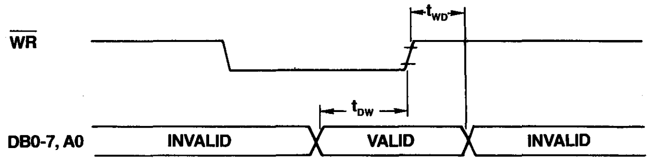


Read-Modify-Write Cycle

Microprocessor Bus Timing

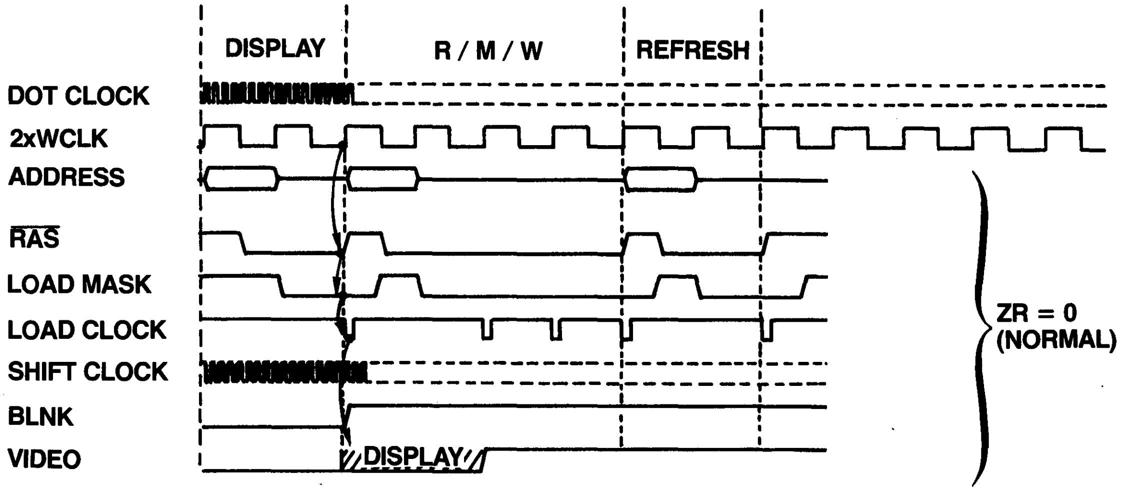


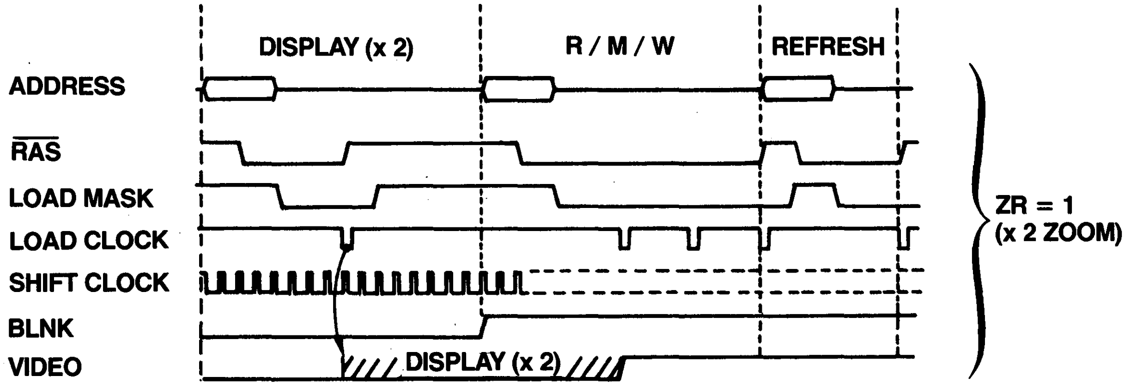
Read Timing

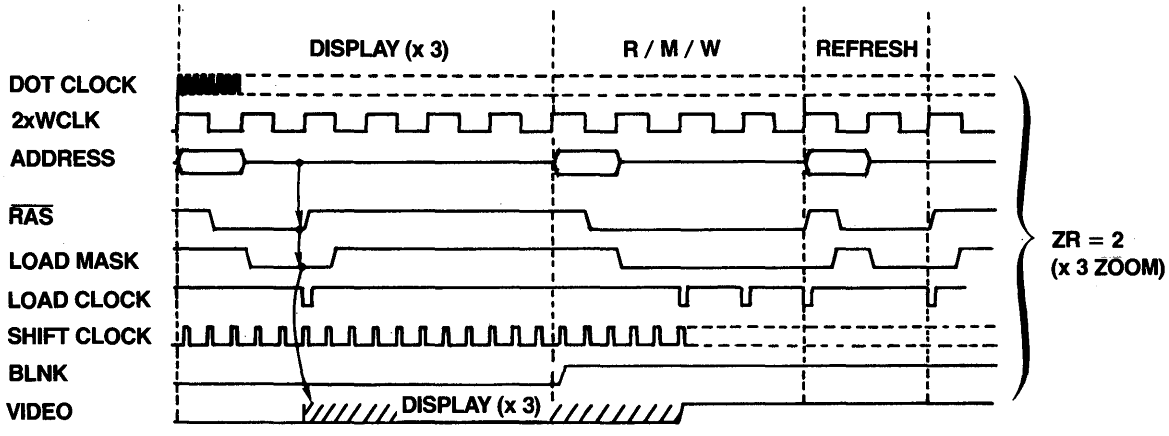


Write Timing

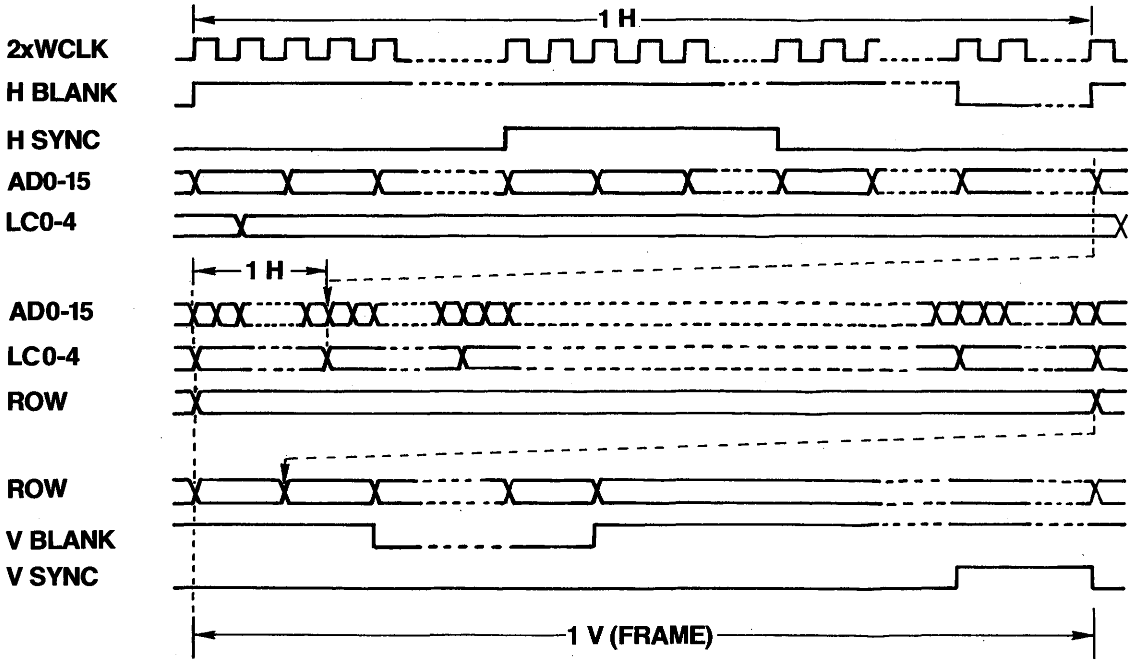
Zoomed Display Timing

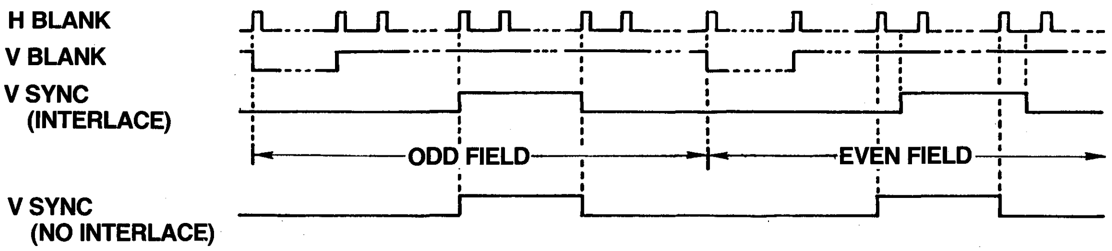




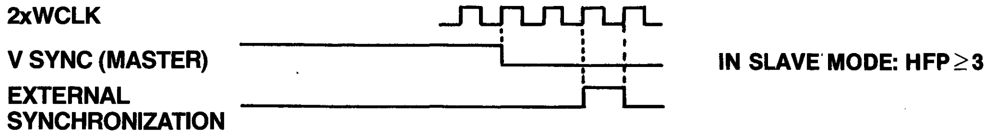


Video Sync Signals Timing

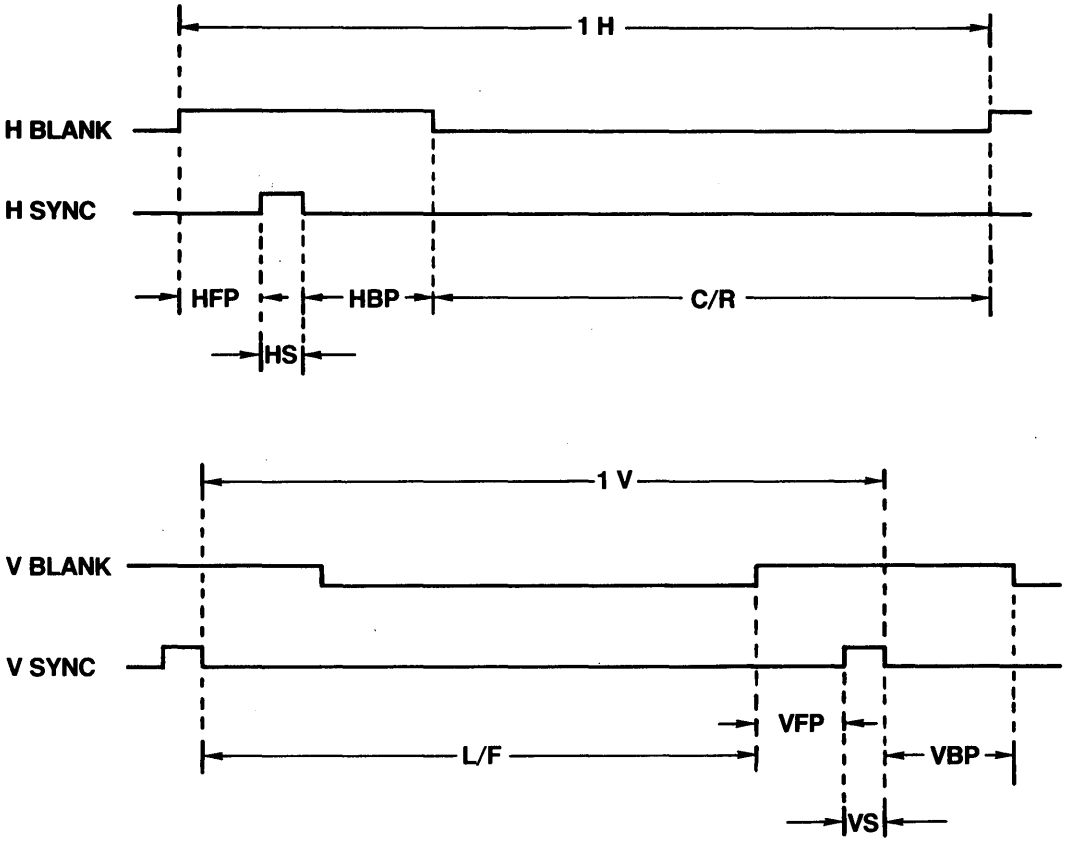




Interlaced Video Timing

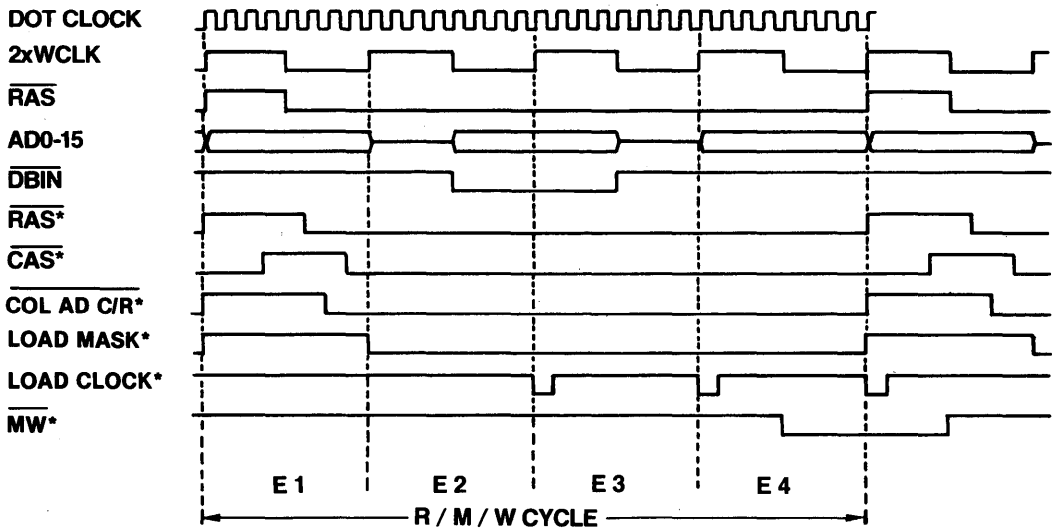


External Synchronization



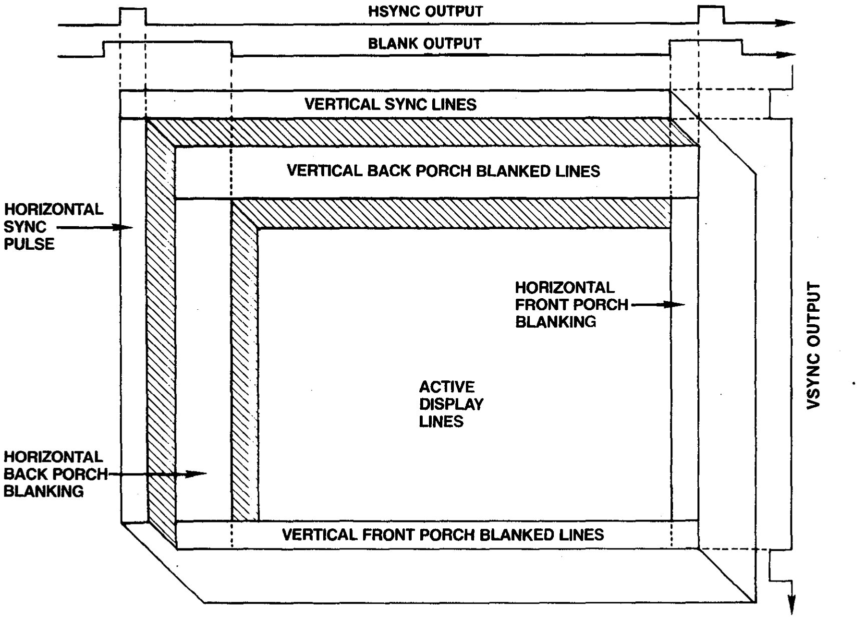
Video Sync Generator Parameters

Read-Modify-Write Cycle Timing

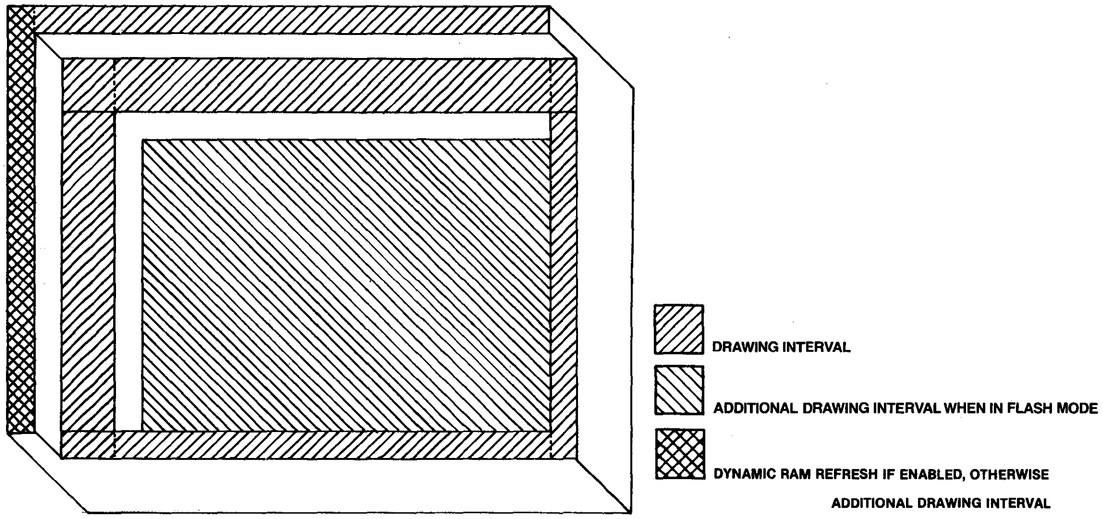


*externally generated signals

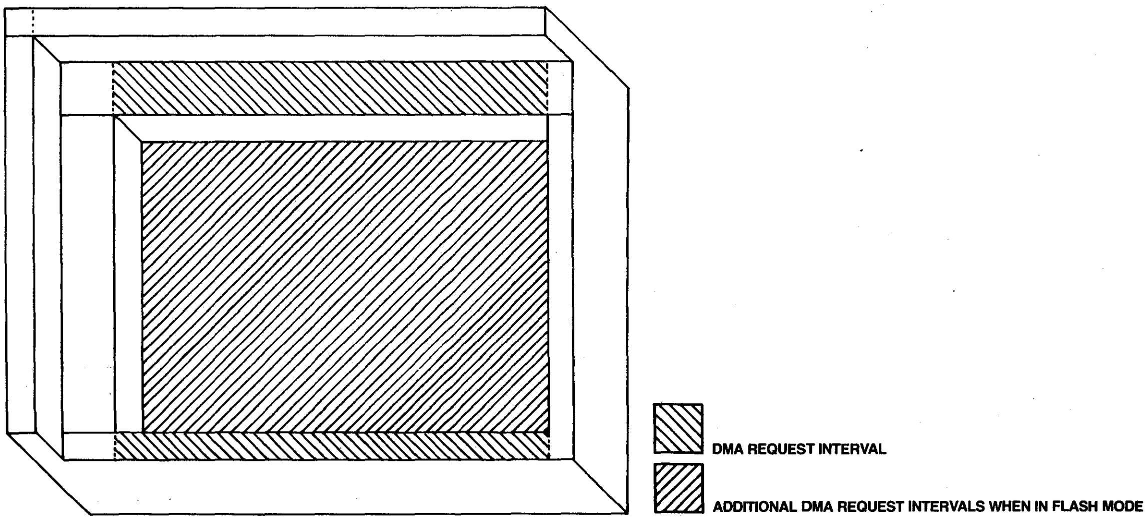
Video Field Timing


HSYNC OUTPUT
BLANK OUTPUT
VERTICAL SYNC LINES
VERTICAL BACK PORCH BLANKED LINES
HORIZONTAL SYNC PULSE
HORIZONTAL FRONT PORCH BLANKING
ACTIVE DISPLAY LINES
HORIZONTAL BACK PORCH BLANKING
VERTICAL FRONT PORCH BLANKED LINES
VSYNC OUTPUT

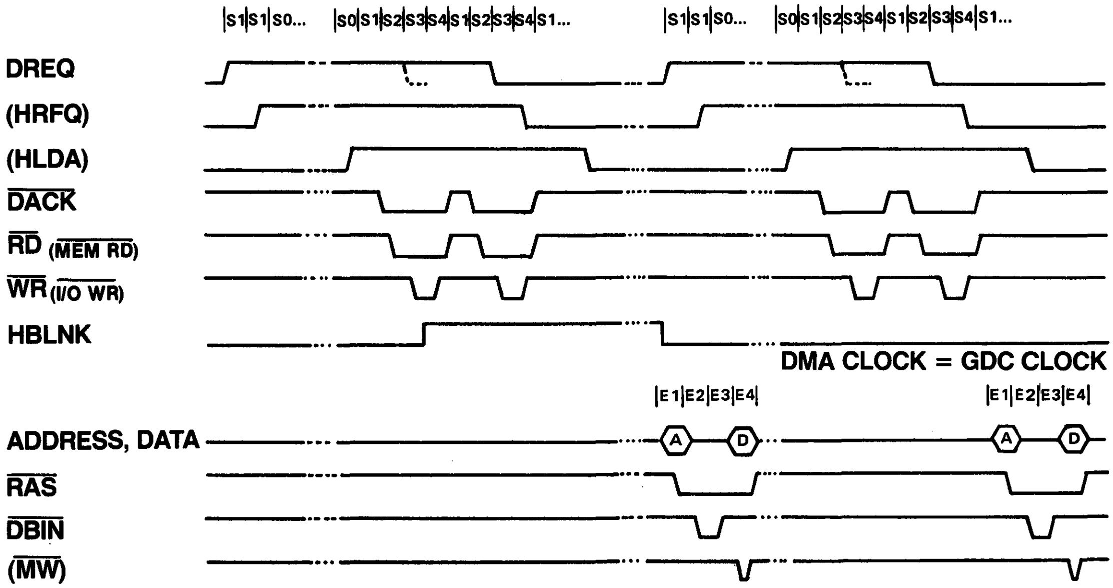
Drawing Intervals


DRAWING INTERVAL
ADDITIONAL DRAWING INTERVAL WHEN IN FLASH MODE
DYNAMIC RAM REFRESH IF ENABLED, OTHERWISE ADDITIONAL DRAWING INTERVAL

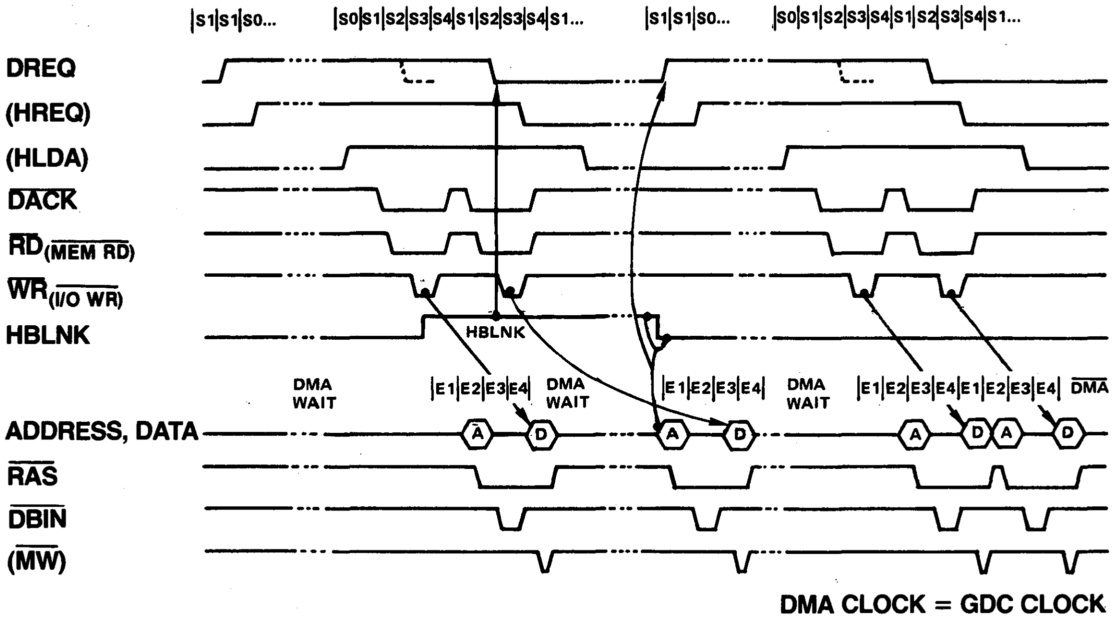
DMA Request Intervals



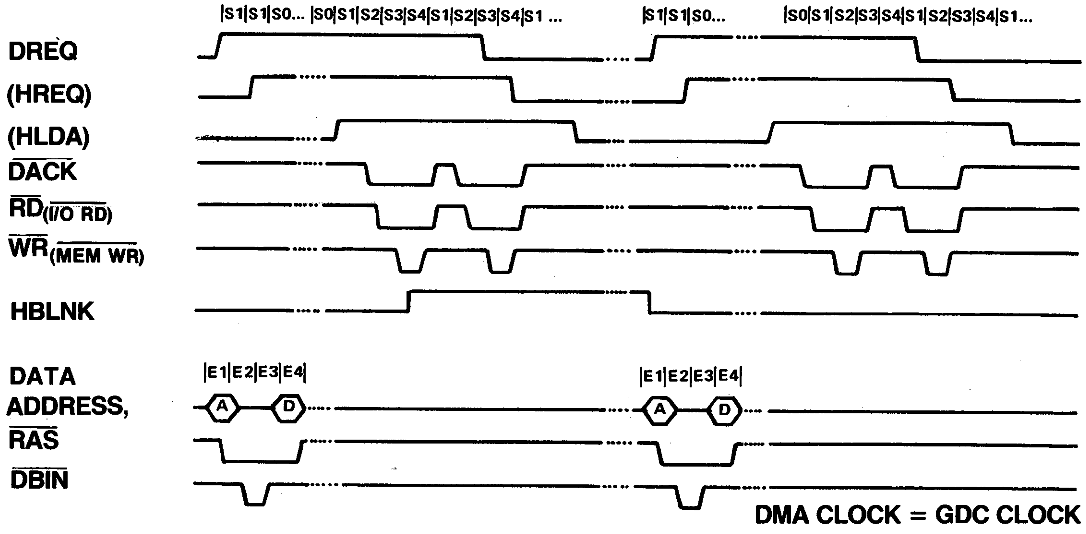
DMA Write (Word) Timing



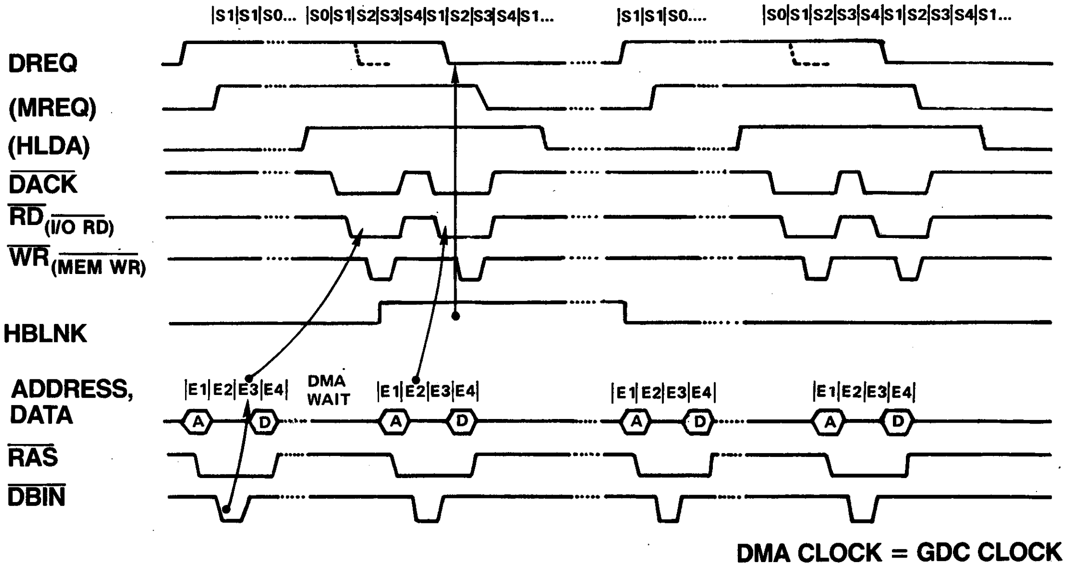
DMA Write (Low Byte, High Byte) Timing



DMA Read (Word) Timing

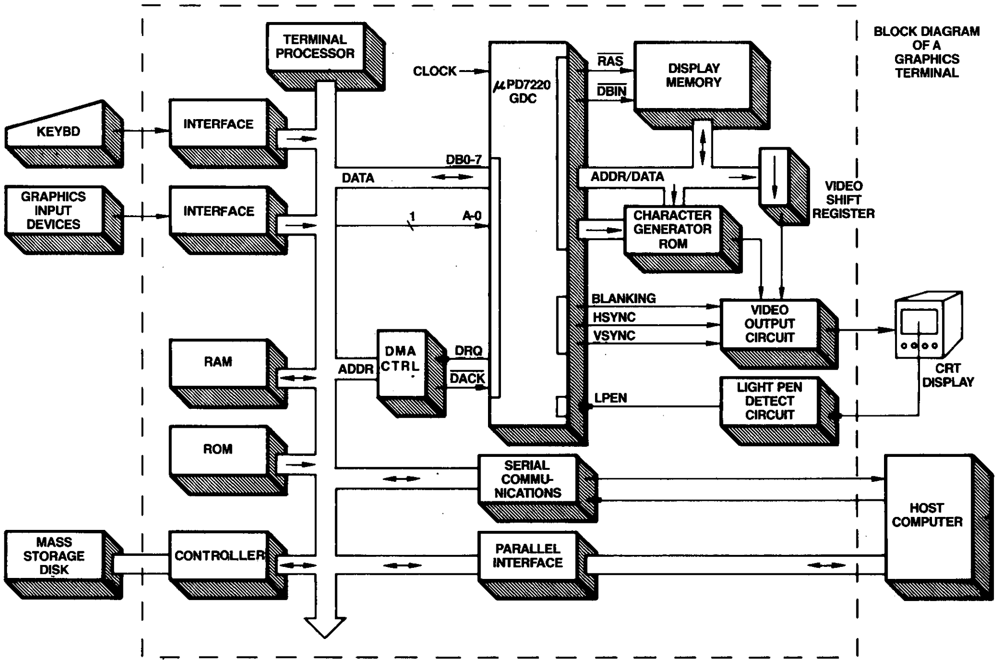


DMA Read (Low Byte, High Byte) Timing

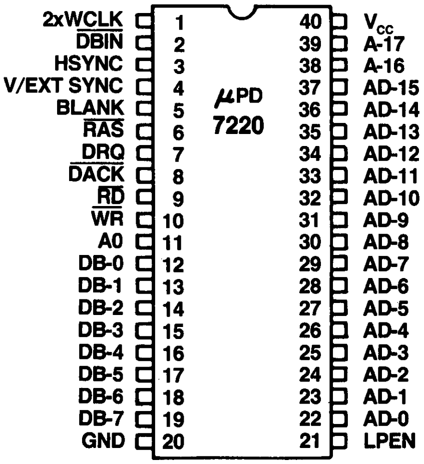


PIN #	PIN NAME	DIRECTION	MODES OF OPERATION		
			GRAPHICS	CHARACTER	MIXED
7	DRQ	OUT	DMA Request Output		
8	$\overline{\text{DACK}}$	IN	DMA Acknowledge Input		
12 to 19	DB-0 to 7	IN/OUT	Data Bus to Microprocessor		
11	A0	IN	Address Select Input for Microprocessor Interface		
9	$\overline{\text{RD}}$	IN	Read Strobe Input for Microprocessor Interface		
10	$\overline{\text{WR}}$	IN	Write Strobe Input for Microprocessor Interface		
3	HSYNC	OUT	Horizontal Video Sync Output		
4	V/EXT SYNC	IN/OUT	Vertical Video Sync Output or External Sync Input		
5	BLANK	OUT	CRT Beam Blanking Output		
1	2xWCLK	IN	Clock Input		
6	$\overline{\text{RAS}}$ (ALE)	OUT	Row Address Strobe and Address Latch Enable		
2	$\overline{\text{DBIN}}$	OUT	Display Memory Data Bus "Input" Read Cycle Output		
39	A-17	OUT	Address Bit 17 Output	Cursor Output	Cursor & Image Mode Flag
38	A-16	OUT	Address Bit 16 Output	Line Counter Bit 3	Attribute Blink & Clear Line Counter
35 to 37	AD-13 to 15	IN/OUT	Address Bits 13 to 15	Line Counter Bits 0 to 2	Address Bits 13 to 15
22 to 34	AD-0 to 12	IN/OUT	Address and Data Lines to Video Display Memory		
21	LPEN	IN	Light Pen Detect Input		
40	V_{CC}	IN	+5V		
20	GND	IN	Ground		

GDC Pin Names



Block Diagram of a Graphics Terminal



Pin Configuration

NEC
NEC Microcomputers
A Division of NEC Electronics U.S.A. Inc.

173 Worcester Street
Wellesley, Massachusetts 02181
Tel: 617-237-1910 TWX 710-383-1745

The information presented in this document is believed to be accurate and reliable. It is subject to change without notice.

PRINTED IN U.S.A.
7220DS-6-81-6K