Overview

Video RAMs (VRAMs) and Synchronous Graphics RAMs (SGRAMs) are DRAMs designed with extra features that make them especially well suited to graphics applications. VRAM architecture and operation are based on those of the standard DRAM. SGRAM architecture and operation are based on those of the Synchronous DRAM. In order to understand how VRAMs and SGRAMs work, you must first understand DRAM and SDRAM operation.

This document briefly describes VRAMs and SGRAMs and how they work. Additional detailed information about the various VRAM and SGRAM features can be found in the product datasheets and in the Applications Notes. Basic information on DRAM operation can be found in "Understanding DRAM Operation."

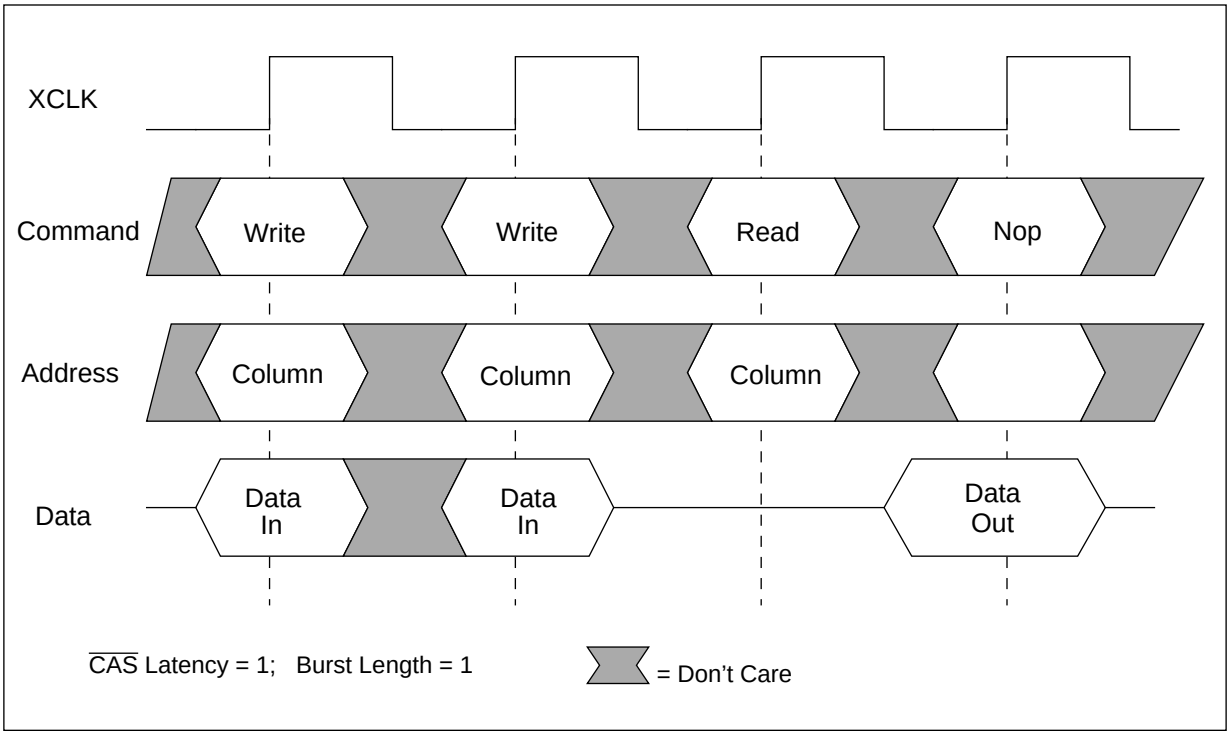
Synchronous Graphics RAM

Graphics applications such as games, multimedia, frame buffers for 2D and 3D applications, and high speed buffers for laser printers, RAID systems and disk drives require large amounts of memory in a wide I/O format. Synchronous Graphics RAMs (or SGRAMs) are designed to address the high speed, wide I/O requirements of this type of application.

SGRAM Operation

SGRAM operation is very similar to that of an SDRAM. Its primary performance features are those found on the SDRAM. This section describes features that have been added to enhance performance and flexibility in graphics applications.

SGRAM Cycle Format



In General...

As with all DRAMs, the main functions of the SGRAM are storing data in the memory array and reading the data out of the array. And, because the SGRAM's memory array is a DRAM array, it must be refreshed periodically.

To make reading and writing data fast and efficient, Block Write and Write-Per-Bit functions have been added to specifically address graphics applications requirements. These features are selected via a

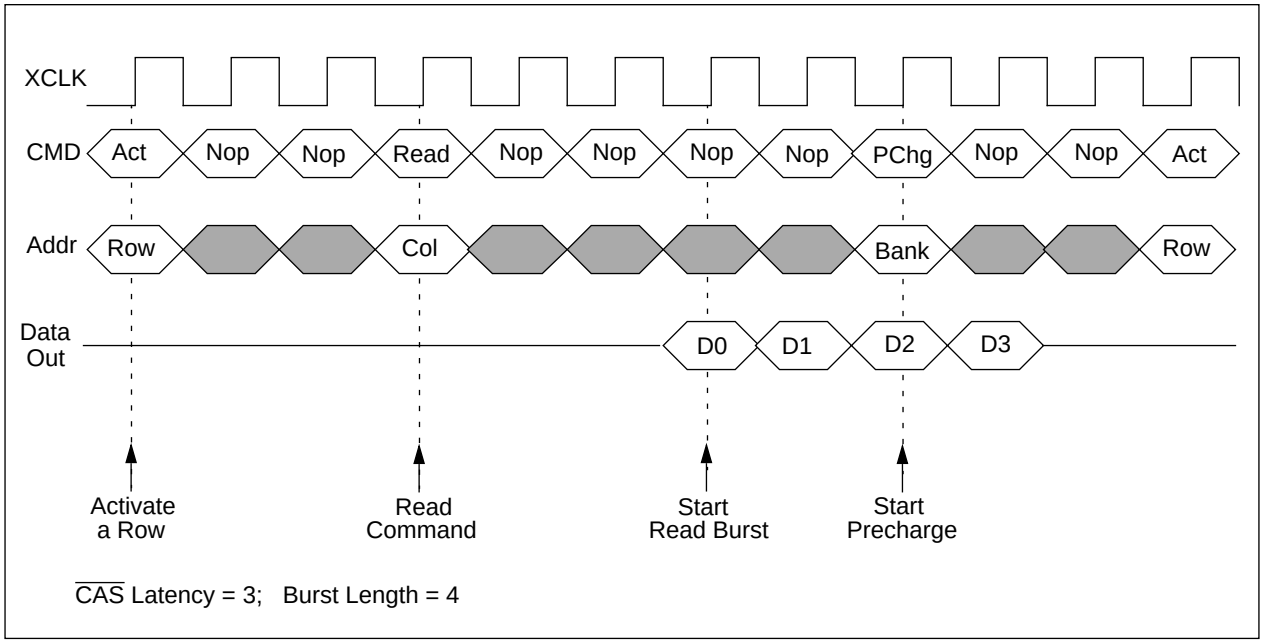
special mode of registers and additional command pins that can be loaded with the appropriate information.

As in the SDRAM, all input signals are registered on the positive edge of the clock signal. IBM's SGRAM is designed for reading and writing in bursts of 1, 2 , 4, 8 bits or a full page. Once a row has been activated, only the starting column address for each burst is required. An internal counter increments the column address for each memory location after the first in a burst. When the read or write parameters are set up, the memory continues through the entire burst until completion or interruption with new data being presented or accepted on each cycle of the burst. To move to a new row, the current row must be precharged and then the new row may be activated.

Reading Data from the SGRAM

The SGRAM specification defines each operation as a series of commands executed on the positive going edge of the clock. A good example for understanding the command structure on an SGRAM is the read operation. To read the data from one or more memory locations, you must do the following:

SGRAM Simplified Read Cycle



Any mode options must be set up before a read can begin, usually on power up. For example, you must choose the length of the burst (how many bits to read once the read operation begins) and type of burst (sequential bits of data and $\overline{CAS}$ latency or interleaved data). Each of these options is selected by loading the Mode Register. You will find more information about the Mode Register, the operations related to it, and diagrams of the required timing in the SGRAM datasheet.

Set up and execute the ROW ACTIVE command. To do so, the row address must be present at the address pins, and $\overline{RAS}$ must be low when the positive transition of the clock occurs.

Following an adequate row activate period, set up and execute the READ command. The beginning column address must be present at the address pins, $\overline{CAS}$ must be low, and $\overline{WE}$ must be high when the positive edge of the clock occurs.

Once the READ operation begins, it continues for the number of column addresses specified in the burst length unless it is interrupted by another burst or termination command.

If the READ operation is to be terminated before the

burst sequence is complete, a TERMINATE BURST, PRECHARGE, WRITE, or a new READ command can be issued.

Once the READ operation is complete and the user is done with the current row, a PRECHARGE command must be issued in order to move to a new row. If automatic precharge was selected as part of the READ command, this step is done for you.

Although this sounds more complicated than standard asynchronous DRAM operation, it isn't. The primary difference is in the clocking requirement. As with all electronics components, it is important to study the timing requirements carefully and understand the implementation of the various features.

SGRAM Features

Once you understand the synchronous timing requirements, implementing the various SGRAM features becomes quite easy. Using the Operative Command Tables, you choose the sequence of commands that lets you perform the operation you require. Some of the choices you can make are:

Programmable burst lengths. You choose how many bits to read or write in a single burst. Once you select the burst length, you can choose the burst operation, and the SGRAM performs the operation for the number of bits specified by the burst length. When you choose the burst length, you can also choose the burst type -- sequential column addresses or interleaved addresses, as described in the Mode Register Definition.

Programmable $\overline{\text{CAS}}$ latency. Because different systems have different timing requirements for accepting and processing data, the SGRAM permits you to choose the timing for the data out. You can choose a delay of one, two, or three clock cycles. Each of these assumes an inherent delay to data out as defined in the SGRAM datasheet.

8-column Block Write and Write-per-Bit modes. Block write permits you to write the data stored in the Color Register to eight consecutive memory locations or columns in a single cycle. Mask data stored in the Mask Register coupled with Write-per-Bit (WPB) mask data present at the inputs (DQs) are used to mask specific bits and prevent them from being written. This feature is especially useful when filling large areas (such as a polygon) on a screen with a single color. The memory locations that contain the color information for the pixels that make up the polygon receive the color data while the rest of the memory locations remain unchanged. Both the Color and WPB Mask Registers can be loaded using a SPECIAL MODE REGISTER command.

Color Register. The 32 bit Color Register is used to store the data for one or more colors. This information is then written to the appropriate memory locations using the 8-column Block Write and Write-per-Bit options. The mask data that appears at the data inputs coupled with the data stored in the Mask Register determine which memory locations receive which data. This option is particularly useful for quickly filling shapes (polygons) on the screen with a particular color.

Video RAMs

Video RAMs were first developed to provide continuous streams of serial data for refreshing video screens. Each pixel or dot on a monitor requires one or more bytes of data that define the color. To maintain a high quality image on the screen, screen data must be refreshed on a fixed interval determined by the design of the monitor itself.

To provide this continuous stream of data, a serial port was added to the standard DRAM. Serial Access Memory (SAM) registers were added to allow for high speed data to the screen while permitting access to the DRAM for data updates, and so on. The SAM registers store a portion of a page of data, freeing up the DRAM for other uses.

VRAM Architecture

The Video RAM is a very specialized form of DRAM. To provide high speed serial streams of data to a video monitor, the VRAM design includes a series of registers called SAM registers, tied to a serial port. These registers access and store a portion of a page of data from the DRAM array, then the data is clocked out of the serial port to a D/A connector at a very high speed. Because the primary use for this type of RAM is video screen refresh, the VRAM is designed to allow serial data to be accessed from this port continuously.

Video RAMs also have a DRAM interface that is completely separate from the serial registers and

serial port. While data is being read from the serial port, other data may be stored in or read from the DRAM array via the DRAM port. For the most part, the DRAM functions on the VRAM operate as they do on a standard DRAM device.

Additional circuitry is included to implement special features such as Block Write, Flash Write, data masks, and color registers.

Reading and Writing Serial Data

Data accessed via the serial port on a VRAM is transferred from the DRAM array to a group of SAM registers. The data from the registers is then transferred via the SAM input/output buffers to the serial port pins.

A SAM is a Serial Access Memory. It is designed to hold part of a page of data from the DRAM array. The objective is to provide continuous data to the serial port to allow video screen updates without tying up the DRAM port. By transferring video data to the SAM registers, the DRAM is available for use by the memory controller, whether for updating the image data stored in the DRAM or for performing other functions.

At its simplest (there are several ways to do this), reading data from the serial port requires the following:

The row address for the data to be read must be present at the address input pins. DSF, $\overline{W}$, and $\overline{TRG}$ must be low.

$\overline{RAS}$ switches from high to low.

The starting column address for the data to be read must be present at the address input pins. $\overline{CAS}$ switches from high to low.

After the appropriate delay, $\overline{TRG}$ goes high, and $\overline{SE}$ goes low.

Serial data appears at the serial outputs, one bit per output for each serial clock cycle (SC).

When the transfer of data from the DRAM to the SAM registers is complete, $\overline{RAS}$ and $\overline{CAS}$ switch from low to high. This portion of the $\overline{RAS}$ and $\overline{CAS}$ cycles is independent of the data read on the serial port.

Other operations on the serial port work in a similar fashion. Data at the serial I/Os is either read from the SAM registers or written to the SAM registers. There is no direct link between the Serial I/Os (SDQs) and the DRAM. Except for the instances in which data is being transferred between the DRAM and the SAM registers, the DRAM and SAM operate independently of each other.

Reading and Writing DRAM Data

The DRAM portion of the Video RAM operates exactly as one would expect a standard DRAM to work. Reading and writing functions, including Fast Page and EDO, are designed and implemented exactly as they are in the standard DRAM. To understand the operation of a Video RAM, one must first understand how a DRAM operates.

Other VRAM Features

There are a number of VRAM features specifically designed to enhance performance and flexibility in graphics applications. Among these are 8-Column Block Write, Write-per-Bit, Flash Write, Mask Registers, and Color Registers. All of these options work with the DRAM portion of the VRAM and are used to efficiently update screen data stored in the DRAM. A brief description of each of these features follows.

8-Column Block Write

8-Column Block Write is used to write the contents of the color registers into eight consecutive column locations in the DRAM in one operation. Using the Masking features, you can select exactly which memory locations get the color data. This option is useful for filling large areas such as polygons with a single color quickly.

Write-per-Bit

Write-per-Bit is a temporary masking option used to mask specific inputs during Write operations. Used in conjunction with the data in the Mask Register, Write-per-Bit is used to select which memory locations get written.

Flash Write

Flash Write clears large portions of the DRAM quickly. Each time the Flash Write option is selected, an entire row of data in the DRAM is cleared.

Mask Register

The Mask Register stores mask data that can be used to prevent certain memory locations from being written. This feature is generally used with the Block Write option and can be used during noraml writes. Bits that are masked (mask data = 0) retain their old data, while unmasked bits are overwritten with new data.

Color Register

The Color Register stores the color data for one or more screen colors. This data is then written to memory locations in the DRAM corresponding to the portions of the screen that will use the stored color. Color Register data is primarily used to rapidly store the color data associated with large areas of a single color, such as a filled polygon.

For example, to fill the polygon, the fill color is stored in the Color Register. Then, using Block Write, all of the memory locations that make up the polygon's area are written with the fill color data.

For more information contact your IBM Microelectronics sales representative or
visit us on World Wide Web at http://www.chips.ibm.com

IBM Microelectronics manufacturing is ISO 9000 compliant.

FormID