

## Overview

Dynamic Random Access Memory (DRAM) devices are used in a wide range of electronics applications. Although they are produced in many sizes and sold in a variety of packages, their overall operation is essentially the same.

DRAMs are designed for the sole purpose of storing data. The only valid operations on a memory device are reading the data stored in the device, writing (or storing) data in the device, and refreshing the data periodically. To improve efficiency and speed, a number of methods for reading and writing the memory have been developed.

This document describes basic asynchronous DRAM operation, including some of the most commonly used features for improving DRAM performance. While many aspects of a synchronous DRAM are similar to an asynchronous DRAM, synchronous operation differs because it uses a clocked interface and multiple bank architecture. Additional information regarding specific features and design issues may be found in the Applications Notes.

## DRAM Architecture

DRAM chips are large, rectangular arrays of memory cells with support logic that is used for reading and writing data in the arrays, and refresh circuitry to maintain the integrity of stored data.

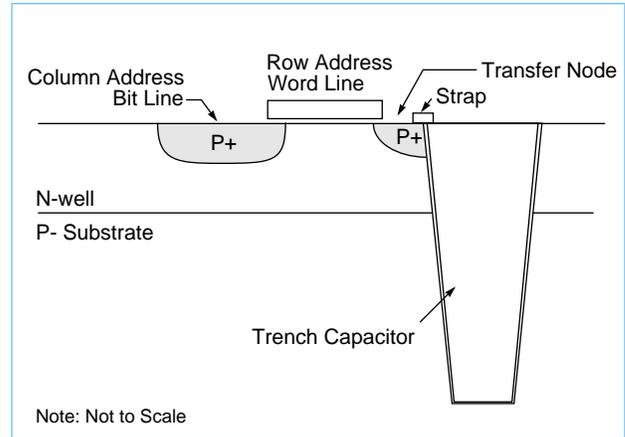
## Memory Arrays

Memory arrays are arranged in rows and columns of memory cells called wordlines and bitlines, respectively. Each memory cell has a unique location or address defined by the intersection of a row and a column.

## Memory Cells

A DRAM memory cell is a capacitor that is charged to produce a 1 or a 0. Over the years, several different structures have been used to create the memory cells on a chip. In today's technologies, trenches filled with dielectric material are used to create the capacitive storage element of the memory cell.

**Figure 1: IBM Trench Capacitor Memory Cell**



## Support Circuitry

The memory chip's support circuitry allows the user to read the data stored in the memory's cells, write to the memory cells, and refresh memory cells. This circuitry generally includes:

- Sense amplifiers to amplify the signal or charge detected on a memory cell.
- Address logic to select rows and columns.
- Row Address Select ( $\overline{RAS}$ ) and Column Address Select ( $\overline{CAS}$ ) logic to latch and resolve the row and column addresses and to initiate and terminate read and write operations.
- Read and write circuitry to store information in the memory's cells or read that which is stored there.
- Internal counters or registers to keep track of the refresh sequence, or to initiate refresh cycles as needed.
- Output Enable logic to prevent data from appearing at the outputs unless specifically desired.

## Understanding the DRAM Timing Diagram

The most difficult aspect of working with DRAM devices is resolving the timing requirements. DRAMs are generally asynchronous, responding to input signals whenever they occur. As long as the signals are applied in the proper sequence, with signal durations and delays between signals that meet the specified limits, the DRAM will work properly.

The datasheets for our products contain timing diagrams for the particular devices. You may want to view a relevant timing diagram while reading the information below.

There are only a few signals that control the operation of a DRAM.

**Row Address Select (Strobe) ( $\overline{\text{RAS}}$ )** The  $\overline{\text{RAS}}$  circuitry is used to latch the row address and to initiate the memory cycle. It is required at the beginning of every operation.  $\overline{\text{RAS}}$  is active low; that is, to enable  $\overline{\text{RAS}}$ , a transition from a high voltage to a low voltage level is required. The voltage must remain low until  $\overline{\text{RAS}}$  is no longer needed. During a complete memory cycle, there is a minimum amount of time that  $\overline{\text{RAS}}$  must be active ( $t_{\text{RAS}}$ ), and a minimum amount of time that  $\overline{\text{RAS}}$  must be inactive, called the  $\overline{\text{RAS}}$  precharge time ( $t_{\text{RP}}$ ).  $\overline{\text{RAS}}$  may also be used to trigger a refresh cycle ( $\overline{\text{RAS}}$  Only Refresh, or ROR).

**Column Address Select (Strobe) ( $\overline{\text{CAS}}$ )**  $\overline{\text{CAS}}$  is used to latch the column address and to initiate the read or write operation.  $\overline{\text{CAS}}$  may also be used to trigger a  $\overline{\text{CAS}}$  before  $\overline{\text{RAS}}$  refresh cycle. This refresh cycle requires  $\overline{\text{CAS}}$  to be active prior to  $\overline{\text{RAS}}$  and to remain active for a specified time. It is active low. The memory specification lists the minimum amount of time  $\overline{\text{CAS}}$  must remain active ( $t_{\text{CAS}}$ ) to initiate a read or write operation. For most memory operations, there is also a minimum amount of time that  $\overline{\text{CAS}}$  must be inactive, called the  $\overline{\text{CAS}}$  precharge time ( $t_{\text{CP}}$ ). (An ROR cycle does not require  $\overline{\text{CAS}}$  to be active.)

**Address** The addresses are used to select a memory location on the chip. The address pins on a memory device are used for both row and column address selection (multiplexing). The number of addresses depends on the memory's size and organization. The voltage level present at each address at the time that  $\overline{\text{RAS}}$  or  $\overline{\text{CAS}}$  goes active determines the row or column address, respectively, that is

selected. To ensure that the row or column address selected is the one that was intended, set up and hold times with respect to the  $\overline{\text{RAS}}$  and  $\overline{\text{CAS}}$  transitions to a low level are specified in the DRAM timing specification.

**Write Enable ( $\overline{\text{WE}}$ )** The write enable signal is used to choose a read operation or a write operation. A low voltage level signifies that a write operation is desired; a high voltage level is used to choose a read operation. The operation to be performed is usually determined by the voltage level on  $\overline{\text{WE}}$  when  $\overline{\text{CAS}}$  goes low (Delayed Write is an exception). To ensure that the correct operation is selected, set up and hold times with respect to  $\overline{\text{CAS}}$  are specified in the DRAM timing specification.

**Output Enable ( $\overline{\text{OE}}$ )** During a read operation, this control signal is used to prevent data from appearing at the output until needed. When  $\overline{\text{OE}}$  is low, data appears at the data outputs as soon as it is available.  $\overline{\text{OE}}$  is ignored during a write operation. In many applications, the  $\overline{\text{OE}}$  pin is grounded and is not used to control the DRAM timing.

**Data In or Out (DQs)** The DQ pins (also called Input/Output pins or I/Os) on the memory device are used for input and output. During a write operation, a voltage (high=1, low=0) is applied to the DQ. This voltage is translated into the appropriate signal and stored in the selected memory cell. During a read operation, data read from the selected memory cell appears at the DQ once access is complete and the output is enabled ( $\overline{\text{OE}}$  low). At most other times, the DQs are in a high impedance state; they do not source or sink any current, and do not present a signal to the system. This also prevents DQ contention when two or more devices share the data bus.



## Using Fast Page Mode and EDO

A number of read options have been developed to improve system performance when reading data. These include Fast Page Mode (FPM), Extended Data Out (EDO) Mode (formerly called Hyper Page Mode or HPM), and burst data using Synchronous DRAMs. All three of these methods rely on the ability to access all of the data stored in a page without having to initiate a completely new memory cycle. Synchronous DRAMs are described in the Applications Note "Synchronous DRAMs: The DRAM of the Future."

A page is defined as all of the memory cells that have a common row address and are wired to a DQ. Each page is as many bits long as the number of column addresses being accessed, ranging from a minimum of two to a maximum of whatever number of columns are in the device's memory array (for example, IBM's 1Mx4 chip can have a page as long as 1024 bits.) The number of full pages (all column addresses on a row) on a memory chip is equal to the number of rows times the number of data DQs (or I/Os). The block diagrams in the datasheets show the number of rows, columns, and DQs (I/Os) for each DRAM configuration.

## Fast Page Mode (FPM)

To read data using Fast Page Mode, start as you would a normal read operation (steps 1 through 7 in the Reading Data From Memory sequence listed above.) Once the data is valid, switch  $\overline{\text{CAS}}$  and  $\overline{\text{OE}}$  high, but leave  $\overline{\text{RAS}}$  low. When  $\overline{\text{CAS}}$  has been inactive (high) for the required amount of time ( $t_{\text{CP}}$ ), repeat steps 3 through 7 of the read operation. You can continue in this way until a new row address is required or the chip needs to be refreshed.  $\overline{\text{OE}}$  does not have to be cycled. If it remains low, the data outputs will be disabled when  $\overline{\text{CAS}}$  goes high.

Figure 3 is a complete timing diagram that illustrates a Fast Page Mode read cycle.

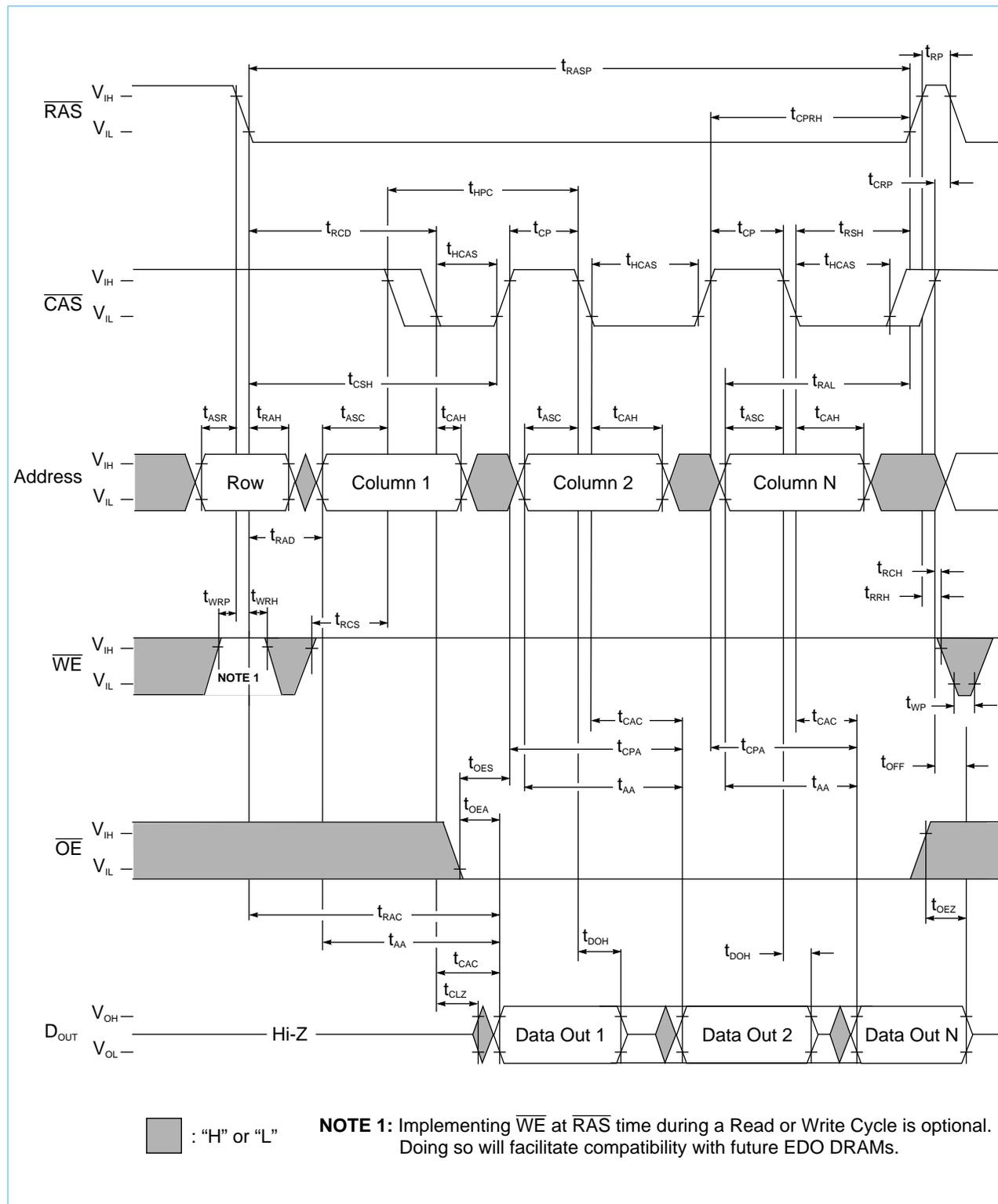
## Extended Data Out (EDO) or Hyper Page Mode

EDO is very similar to FPM. The main difference is that the data output drivers are not disabled when  $\overline{\text{CAS}}$  goes high on the EDO DRAM, allowing the data from the current read cycle to be present at the outputs while the next read cycle begins, and resulting in a faster cycle time. Figure 4 is a complete timing diagram that illustrates an EDO mode read cycle.

The Applications Note "EDO (Hyper Page Mode)" explains Fast Page Mode and EDO operation in more detail.



Figure 4. EDO (Hyper Page) Mode Read Cycle



## Writing Data To Memory

Figure 3 is the timing diagram of a simplified Write cycle that illustrates described below.

To write to a memory cell, the row and column address for the cell must be selected and data must be presented at the data input pins. The chip's on-board logic either charges the memory cell's capacitor or discharges it, depending on whether a 1 or 0 is to be stored. In terms of timing, the following steps must occur:

1. The row address must be applied to the address input pins on the memory device for the prescribed amount of time before  $\overline{\text{RAS}}$  goes low and be held for a period of time.
2.  $\overline{\text{RAS}}$  must go from high to low.
3. A column address must be applied to the address input pins on the memory device for the prescribed amount of time after  $\overline{\text{RAS}}$  goes low and before  $\overline{\text{CAS}}$  goes low and held for the prescribed time.
4.  $\overline{\text{WE}}$  must be set low for a certain time for a write operation to occur ( $t_{\text{WP}}$ ). The timing of the transitions are determined by  $\overline{\text{CAS}}$  going low ( $t_{\text{WCS}}$ ,  $t_{\text{WCH}}$ ).
5. Data must be applied to the data input pins the

prescribed amount of time before  $\overline{\text{CAS}}$  goes low ( $t_{\text{DS}}$ ) and held ( $t_{\text{DH}}$ ).

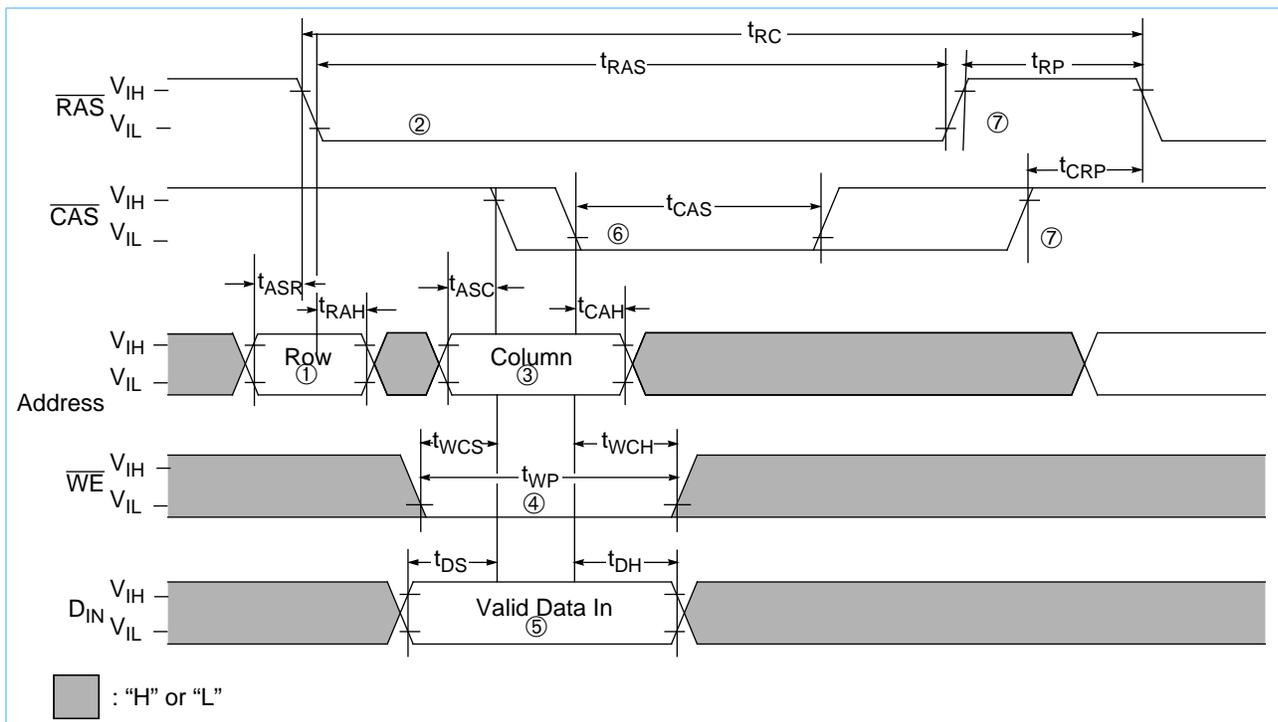
6.  $\overline{\text{CAS}}$  must switch from high to low.
7. Before the write cycle can be considered complete,  $\overline{\text{CAS}}$  and  $\overline{\text{RAS}}$  must return to their inactive states.

Note: There is considerable latitude within the memory chip's timings with respect to wOEhen data is actually written. The memory specifications show how to set up chip timings for early and delayed write options.

## Using Fast Page Mode and EDO

The improved performance gained by using FPM or EDO mode to read multiple bits on a page can also be seen when writing data. After the first memory location has been selected and the data stored,  $\overline{\text{CAS}}$  must go high. Subsequent memory locations can then be written to using steps 3 through 7 in the Writing Data To Memory sequence listed above. In step 7,  $\overline{\text{RAS}}$  needs to remain low. You can continue in this way until you need to change the row address or the chip needs to be refreshed.

## Simplified Write Cycle



## Using Read-Modify-Write

Figure 5 shows the timing parameters of a simplified Read-Modify-Write cycle as described below.

Read-Modify-Write is a special case in which a memory location is first read and then written to. Many DRAMs have designed their chips to allow a single memory location to be selected for both a read and a write operation without incurring the time penalty that results when a new address must be selected. Note that the overall cycle time is extended ( $t_{RWC}$ ) to allow both operations to occur.

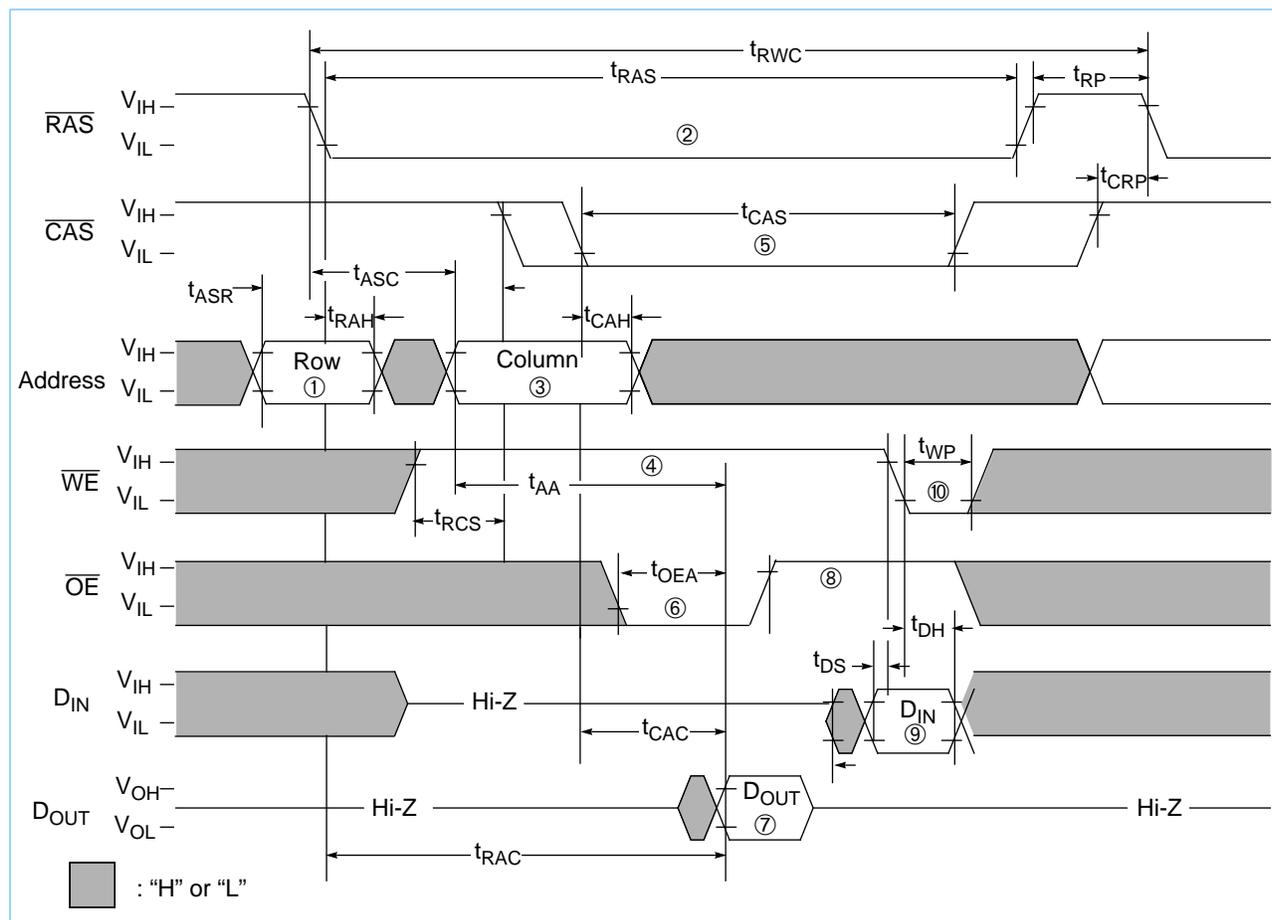
To execute a Read-Modify-Write cycle, the following steps must occur:

1. The row address must be applied to the address input pins on the memory device for the prescribed amount of time before  $\overline{RAS}$  goes low.
2.  $\overline{RAS}$  must go from high to low.
3. A column address must be applied to the

address input pins on the memory device for the prescribed amount of time after  $\overline{RAS}$  goes low and before  $\overline{CAS}$  goes low.

4.  $\overline{WE}$  must be set high for the read operation to occur.
5.  $\overline{CAS}$  must switch from high to low.
6.  $\overline{OE}$  goes low within the prescribed window of time.
7. Data appears at the data output pins of the memory device.
8.  $\overline{OE}$  goes high and disables the data output.
9. Data must be applied to the data input pins for the prescribed amount of time before  $\overline{WE}$  goes low.
10.  $\overline{WE}$  must switch from high to low for the write operation to occur.
11. Before the Read-Modify-Write cycle can be considered complete,  $\overline{CAS}$  and  $\overline{RAS}$  must return to their inactive states.

## Simplified Read-Modify-Write Cycle



## Refreshing the Memory

Since DRAM memory cells are capacitors, the charge they contain can leak away over time. If the charge is lost, so is the data. To prevent this from happening, DRAMs must be refreshed -- that is, the charge on the individual memory cells must be restored periodically. The frequency with which refresh must occur depends on the silicon technology used to manufacture the memory chip and the design of the memory cell itself.

Reading or writing a memory cell has the effect of refreshing the selected cell. Unfortunately, not all cells are read or written within the time limitations. Thus each cell in the array must be accessed and restored during the refresh interval. In most cases, refresh cycles involve restoring the charge along an entire row. Over the course of the entire interval, every row is accessed and restored. At the end of the interval, the process begins again.

System designers have a lot of latitude in designing and implementing memory refresh. They may choose to fit refresh cycles between normal read and write cycles, or they may decide to run refresh cycles on a fixed schedule, forcing the system to queue read and write operations when they conflict with refresh requirements.

There are several different ways to refresh a memory array. The method you use will depend on the memory product you choose and the requirements of the system you are designing. Three common refresh options are briefly described below. Another refresh option is hidden refresh, in which a read or write operation and a refresh cycle are performed during a single  $\overline{\text{CAS}}$  active period.

### Using $\overline{\text{RAS}}$ Only Refresh (ROR)

Normally, DRAMs are refreshed one row at a time. The refresh cycles are distributed across the entire refresh interval in such a way that all rows are refreshed within the required interval. To refresh one row of the memory array using  $\overline{\text{RAS}}$  Only Refresh, the following steps must occur.

1. The row address of the row to be refreshed must be applied at the address input pins.
2.  $\overline{\text{RAS}}$  must switch from high to low.  $\overline{\text{CAS}}$  must remain high.

3. At the end of the required amount of time,  $\overline{\text{RAS}}$  must return high.

### Using $\overline{\text{CAS}}$ before $\overline{\text{RAS}}$ Refresh (CBR)

To refresh one row of the memory array using  $\overline{\text{CAS}}$  before  $\overline{\text{RAS}}$  Refresh, the following steps must occur.

1.  $\overline{\text{CAS}}$  must switch from high to low.
2.  $\overline{\text{WE}}$  must switch to a high state (Read).
3. After the prescribed delay,  $\overline{\text{RAS}}$  must switch from high to low.
4. An internal counter determines which row is to be refreshed.
5. After the required delay,  $\overline{\text{CAS}}$  returns to a high level.
6. After the required delay,  $\overline{\text{RAS}}$  returns to a high level.

The main difference between ROR and CBR is the method for keeping track of the row address to be refreshed. With ROR, the system must provide the row address to be refreshed. With CBR, the chip keeps track of the addresses using an internal counter.

### Self Refresh (SR)

Self Refresh, also referred to as Sleep Mode or Auto Refresh, is unique because it uses an on-chip oscillator to determine the refresh frequency and a counter to keep track of addressing. SR is most often used for battery-powered mobile applications and applications that use a battery for backup power. While in sleep mode, the device uses extremely low current.

The timing required to initiate SR is a CBR cycle with  $\overline{\text{RAS}}$  active for a minimum of 100 microseconds. The length of time that a device can be left in sleep mode is limited by the power source used. To exit,  $\overline{\text{RAS}}$  and  $\overline{\text{CAS}}$  are brought high.



© International Business Machines Corp.1996

Printed in the United States of America  
All rights reserved

IBM and the IBM logo are registered trademarks of the IBM Corporation.

This document may contain preliminary information and is subject to change by IBM without notice. IBM assumes no responsibility or liability for any use of the information contained herein. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of IBM or third parties. The products described in this document are not intended for use in implantation or other direct life support applications where malfunction may result in direct physical harm or injury to persons. **NO WARRANTIES OF ANY KIND, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, ARE OFFERED IN THIS DOCUMENT.**

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.