İstanbul
Bilgi Üniversitesi

CMPE321 - Computer Architecture

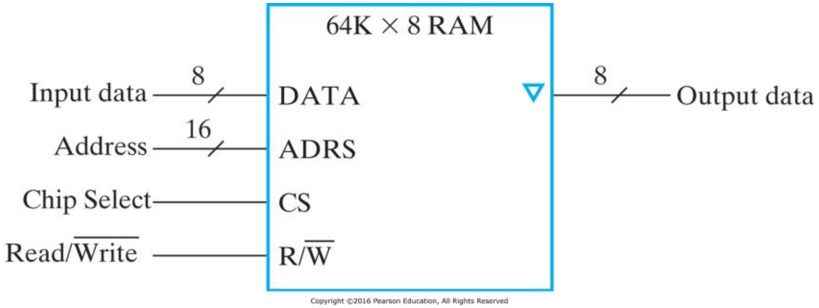
Lecture 6
Array of SRAM ICs & DRAM

Hakan Ayral, PhD.

Array of SRAM ICs

- Integrated-circuit RAM chips are available in a variety of sizes.
- If the memory unit needed for an application is larger than the capacity of one chip, it is necessary to combine a number of chips in an array to form the required size of memory.
- The capacity of the memory depends on two parameters: the **number of words** and the **number of bits per word**.
- An increase in the number of words requires that we increase the address length.
- Every bit added to the length of the address doubles the number of words in memory.
- An increase in the number of bits per word requires that we increase the number of data input and output lines, but the address length remains the same.

Array of SRAM ICs

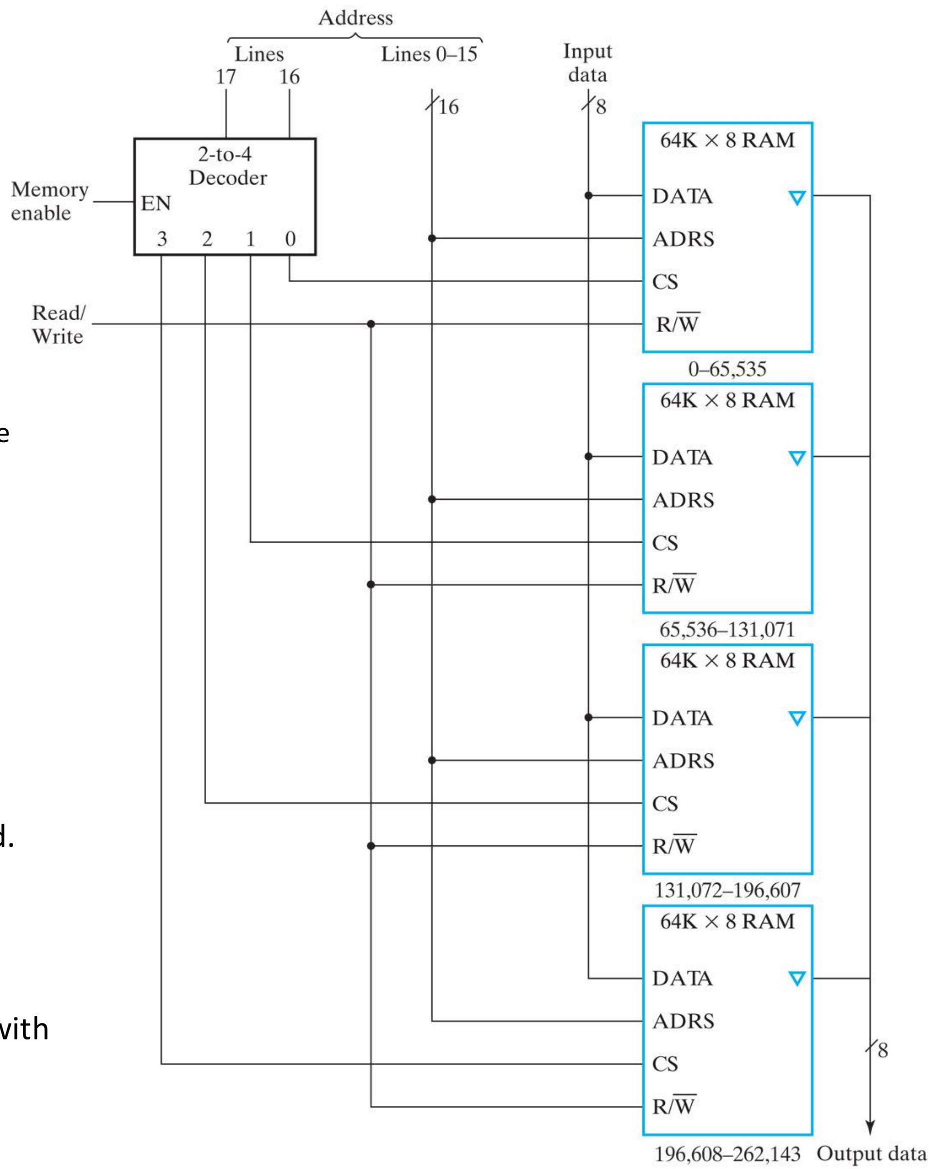


Copyright ©2016 Pearson Education, All Rights Reserved

- To illustrate an array of RAM ICs, let us first introduce a RAM chip using the condensed representation for inputs and outputs shown in Figure on right.
- The capacity of this chip is 64K words of 8 bits each.
- The chip requires a 16-bit address and 8 input and output lines.
- Instead of 16 lines for the address and 8 lines each for data input and data output, each is shown in the block diagram by a single line.
- Each line has a slash across it with a number indicating the number of lines represented.
- The *CS* (Chip Select) input selects the particular RAM chip, and the R/$\overline{\text{W}}$ input specifies the read or write operation when the chip is selected.
- The small triangle shown at the outputs is the standard graphics symbol for three-state outputs.
- The *CS* input of the RAM controls the behavior of the data output lines.
- When *CS* = 0, the chip is not selected, and all its data outputs are in the high-impedance state.
- With *CS* = 1, the data output lines carry the eight bits of the selected word.
- Suppose that we want to increase the number of words in the memory by using two or more RAM chips.
- Since every bit added to the address doubles the binary number that can be formed, it is natural to increase the number of words in factors of two.
- For example, two RAM chips will double the number of words and add one bit to the composite address. Four RAM chips multiply the number of words by four and add two bits to the composite address.

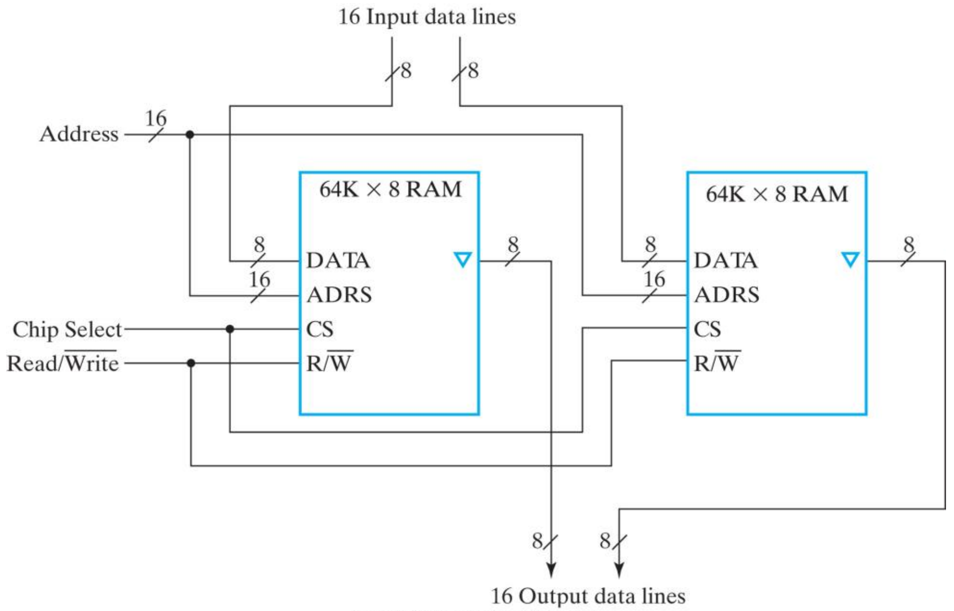
Array of SRAM ICs

- Consider constructing a 256K × 8 RAM with four 64K × 8 RAM chips, as shown in Figure.
- The eight data input lines go to all the chips.
- The three-state outputs can be connected together to form the eight common data output lines.
    - This type of output connection is possible only with three-state outputs.
    - Just one Chip Select input will be active at any time, while the other three chips will be disabled.
    - The eight outputs of the selected chip will contain 1s and 0s, and the other three will be in a high-impedance state, presenting only open circuits to the binary output signals of the selected chip.
- The 256K-word memory requires an 18-bit address.
    - The 16 least significant bits of the address are applied to the address inputs of all four chips.
    - The two most significant bits are applied to a 2 × 4 decoder.
- The four outputs of the decoder are applied to the *CS* inputs of the four chips.
    - The memory is disabled when the *EN* input of the decoder, Memory Enable, is equal to 0.
    - All four outputs of the decoder are then 0, and none of the chips is selected.
- When the decoder is enabled, address bits 17 and 16 determine the particular chip that is selected.
- If these bits are equal to 00, the first RAM chip is selected.
- The remaining 16 address bits then select a word within the chip in the range from 0 to 65,535.
- The next 65,536 words are selected from the second RAM chip with an 18-bit address that starts with 01 followed by the 16 bits from the common address lines.
- The address range for each chip is listed in decimal under its symbol in the figure.



Array of SRAM ICs

- It is also possible to combine two chips to form a composite memory containing the same number of words, but with twice as many bits in each word.
- Figure on right shows the interconnection of two 64K × 8 chips to form a 64K × 16 memory.
- The 16 data input and data output lines are split between the two chips.
- Both receive the same 16-bit address and the common *CS* and *R*/W control inputs.
- The two techniques just described may be combined to assemble an array of identical chips into a large-capacity memory.
- The composite memory will have a number of bits per word that is a multiple of that for one chip.
- The total number of words will increase in factors of two times the word capacity of one chip.
- An external decoder is needed to select the individual chips based on the additional address bits of the composite memory.
- To reduce the number of pins on the chip package, many RAM ICs provide common terminals for the data input and data output.
- The common terminals are said to be ***bidirectional***, which means that for the read operation
- they act as outputs, and for the write operation they act as inputs.
- Bidirectional lines are constructed with three-state buffers.
- The use of bidirectional signals requires control of the three-state buffers by both Chip Select and Read/Write.

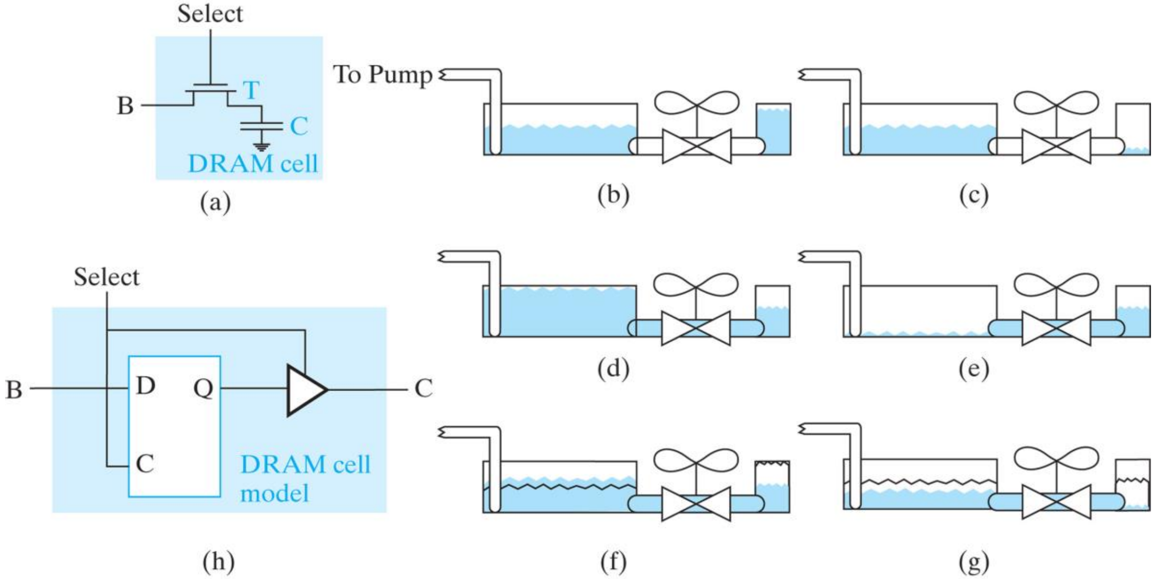


DRAM ICs

- Because of its ability to provide high storage capacity at low cost, dynamic RAM (DRAM) dominates the high-capacity memory applications, including the primary RAM in computers.
- Logically, DRAM in many ways is similar to SRAM.
- However, because of the electronic circuit used to implement the storage cell, its electronic design is considerably more challenging.
- Further, as the name **dynamic** implies, the storage of information is inherently only temporary.
- As a consequence, the information must be periodically **refreshed** to mimic the behavior of static storage.
- This need for refresh is the primary logical difference in the behavior of DRAM compared to SRAM.
- We explore this logical difference by examining the dynamic RAM cell, the logic required to perform the refresh operation, and the impact of the need for refresh on memory system operation.

DRAM Cell

- The dynamic RAM cell circuit is shown in Figure below. It consists of a capacitor C and a transistor T.
- The capacitor is used to store electrical charge.
  - If sufficient charge is stored on the capacitor, it can be viewed as storing a logical 1.
  - If insufficient charge is stored on the capacitor, it can be viewed as storing a logical 0.
- The transistor acts much like a switch.
  - When the switch is “open,” the charge on the capacitor roughly remains fixed—in other words, is stored.
  - But when the switch is “closed,” charge can flow into and out of the capacitor from the external Bit (B) line.
- This charge flow allows the cell to be written with a 1 or 0 and to be read.
- In order to understand the read and write operations for the cell, we will use a hydraulic analogy with charge replaced by water, the capacitor by a small storage tank, and the transistor by a valve.
- Since the bit line has a large capacitance, it is represented by a large tank and pumps which can fill and empty this tank rapidly.
- This analogy is given in Figure (b) and (c) with the valve closed.
- Note that in one case the small storage tank is full, representing a stored 1, and in the other case it is empty, representing a stored 0.
- Suppose that a 1 is to be written into the cell.
  - The valve is opened and the pumps fill up the large tank.
  - Water flows through the valve, filling the small storage tank, as shown in Figure (d).
  - Then the valve is closed, leaving the small tank full, which represents a 1.
- A 0 can be written using the same sort of operations, except that the pumps empty the large tank as shown in (e).



(a) (b) (c) (d) (e) (f) (g) (h)

DRAM Cell

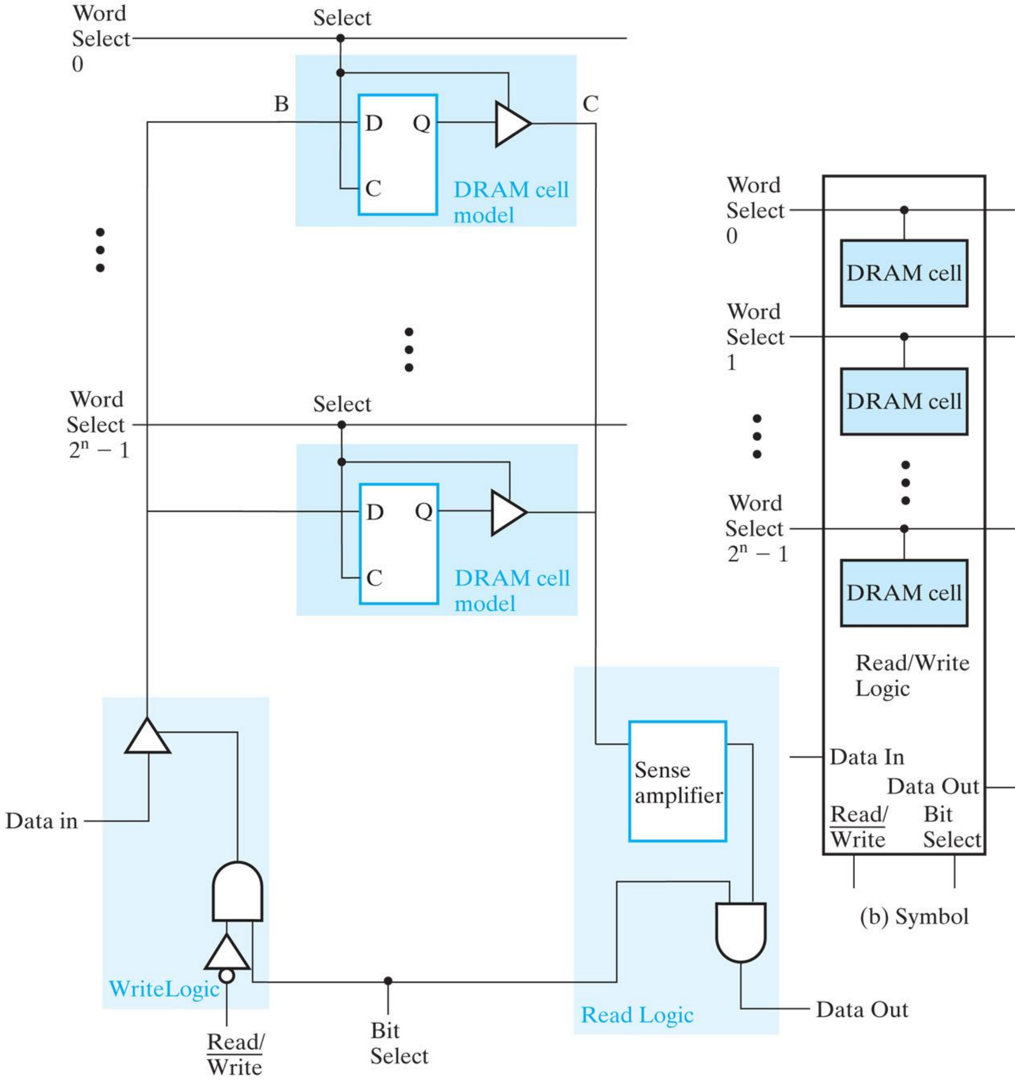
- Now, suppose we want to read a stored value and that the value is a 1 corresponding to a full storage tank. With the large tank at a known intermediate level, the valve is opened.
- Since the small storage tank is full, water flows from the small tank to the large tank, increasing the level of the water surface in the large tank slightly as shown in previous Figure (f).
- This increase in level is observed as the reading of 1 from the storage tank.
- Correspondingly, if the storage tank is initially empty, there will be a slight decrease in the level in the large tank in Figure (g), which is observed as the reading of a 0 from the storage tank.
- In the read operation just described, Figure (f) and (g) show that, regardless of the initial stored value in the storage tank, it now contains an intermediate value which will not cause enough change in the level of the external tank to permit a 0 or 1 to be observed.
- So the read operation has destroyed the stored value; this is referred to as a ***destructive read***. To be able to read the original stored value in the future, we must ***restore*** it (i.e., return the storage tank to its original level).
  - To perform the <u>restore for a stored 1</u> observed, the large tank is filled by the pumps and the small tank fills through the open valve.
  - To perform the <u>restore for a stored 0</u> observed, the large tank is emptied by the pumps and the small tank drains through the open valve.

DRAM Cell

- In the actual storage cell, there are other paths present for charge flow.
- These paths are analogous to small leaks in the storage tank.
- Due to these leaks, a full small storage tank will eventually drain to a point at which the increase in the level of the large tank on a read cannot be observed as an increase.
- In fact, if the small tank is less than half full when read, it is possible that a decrease in the level of the large tank may be observed.
- To compensate for these leaks, the small storage tank storing a 1 must be periodically refilled.
- This is referred to as a refresh of the cell contents.
- Every storage cell must be refreshed before its level has declined to a point at which the stored value can no longer be properly observed.
- Through the hydraulic analogy, the DRAM operation has been explained.
- Just as for the SRAM, we employ a logic model for the cell.
- The model shown in Figure (h) is a D latch.
- The ***C* input** to the D latch is **Select** and the ***D* input** to the D latch is ***B***.
- In order to model the output of the DRAM cell, we use a three-state buffer with Select as its control input and *C* as its output.
- In the original electronic circuit for the DRAM cell in Figure (a), *B* and *C* are the same signal, but in the logical model they are separate.
- This is necessary in the modeling process to avoid connecting gate outputs together.

DRAM Bit Slice

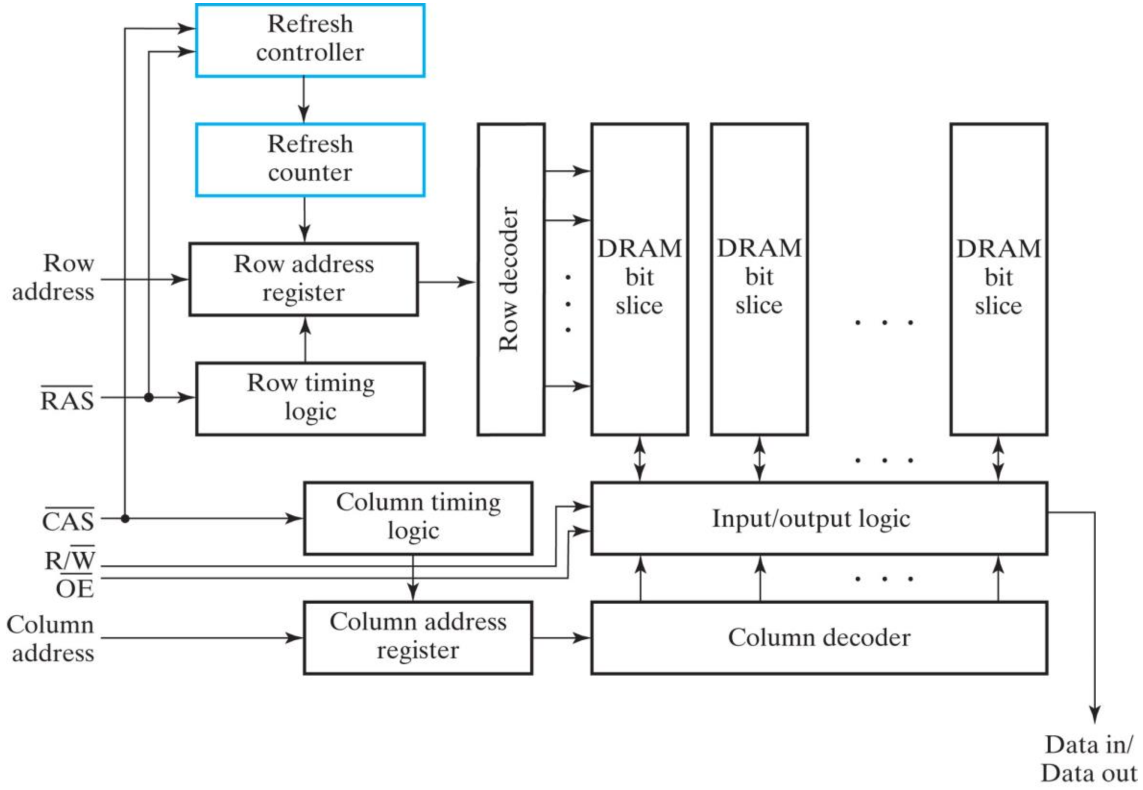
- Using the logic model for the DRAM cell, we can construct the DRAM bit-slice model shown in Figure. This model is similar to that for the SRAM bit slice.
- Aside from the cell structure, the two RAM bit slices are logically similar.
- However, from the standpoint of cost per bit, they are quite different.
- The DRAM cell consists of a capacitor plus one transistor.
- The SRAM cell typically contains six transistors, giving a cell complexity roughly three times that of the DRAM.
- Therefore, the number of SRAM cells in a chip of a given size is less than one-third of those in the DRAM.
- The DRAM cost per bit is less than one-third the SRAM cost per bit, which justifies the use of DRAM in large memories.
- Refresh of the DRAM contents remains to be discussed. But first, we need to develop the typical structure used to handle addressing in DRAMs.
- Since many DRAM chips are used in a DRAM, we want to reduce the physical size of the DRAM chips.
- Large DRAMs require 20 or more address bits, which would require 20 address pins on each DRAM chip.



(a) Logic diagram

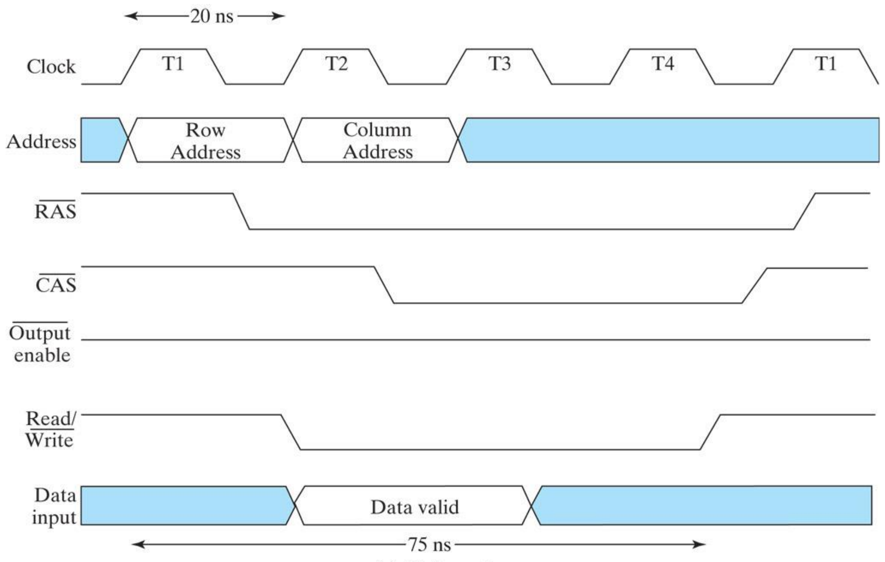
DRAM Bit Slice

- To reduce the number of pins, the DRAM address is applied serially in two parts with the row address first and the column address second.
- This can be done since the row address, which performs the row selection, is actually needed before the column address, which reads out the data from the row selected.
- In order to hold the row address throughout the read or write cycle, it is stored in a register, as shown in Figure to the right.
- The column address is also stored in a register.
- The load signal for the row address register is $\overline{\text{RAS}}$ (Row Address Strobe) and for the column addresses is $\overline{\text{CAS}}$ (Column Address Strobe).
- Note that in addition to $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ control signals for the DRAM chip include R/W and OE (Output Enable).
- Also note that this design uses signals active at the LOW (0) level.

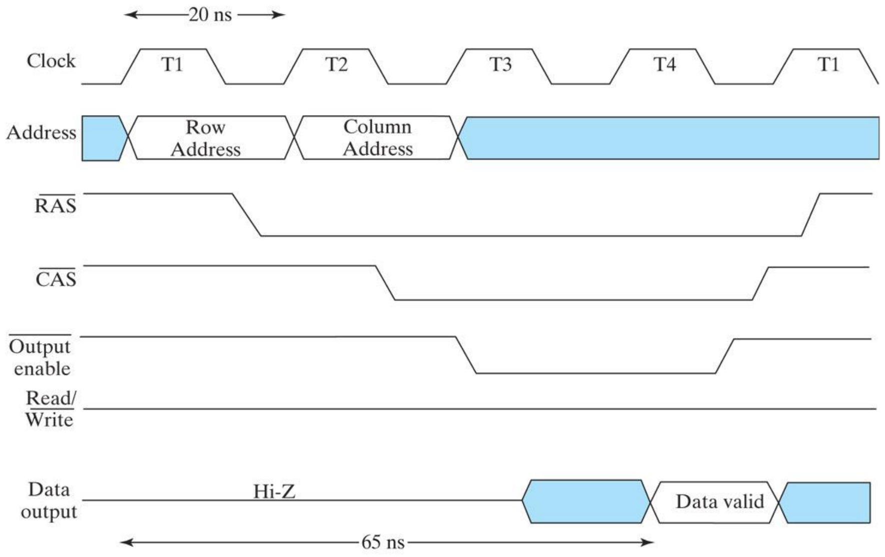


DRAM Bit Slice

- The timing for DRAM write and read operation appears in Figure 15(a).
- The row address is applied to the address inputs, and then RAS changes from 1 to 0, loading the row address into the row address register.
- This address is applied to the row address decoder and selects a row of DRAM cells.
- Meanwhile, the column address is applied, and then CAS changes from 1 to 0, loading the column address into the column address register.
- This address is applied to the column address decoder, which selects a set of columns of the RAM array of size equal to the number of RAM data bits.
- The input data with Read/$\overline{\text{Write}}$ = 0 is applied over a time interval similar to that for the column address.
- The data bits are applied to the set of bit lines selected by the column address decoder, which in turn apply the values to the DRAM cells in the selected row, writing the new data into the cells.
- When CAS and RAS return to 1, the write cycle is complete and the DRAM cells store the newly written data.
- Note that the stored data in all of the other cells in the addressed row has been restored.



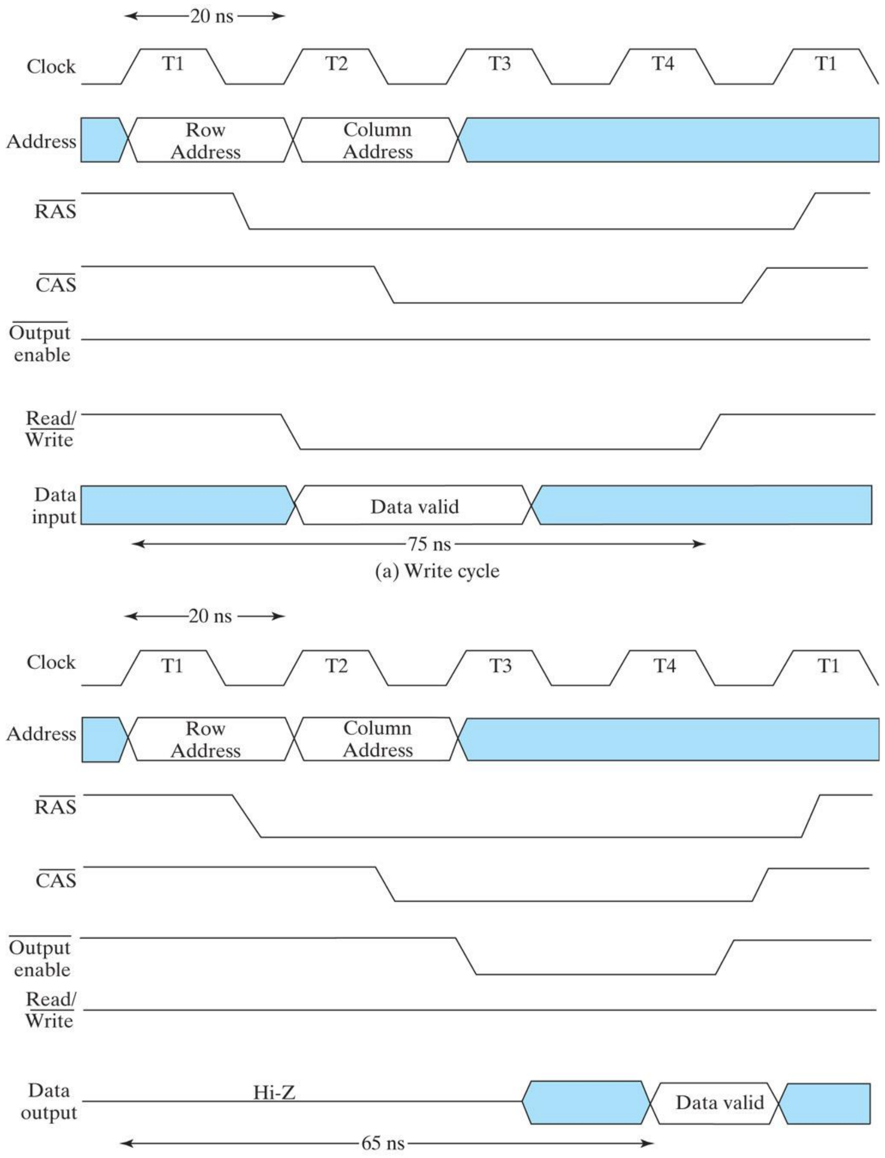
(a) Write cycle



(b) Read cycle

DRAM Bit Slice

- The read operation timing shown in Figure (b) is similar.
- Timing of the address operations is the same.
- However, no data is applied and Read/$\overline{\text{Write}}$ is 1 instead of 0.
- Data values in the DRAM cells in the selected row are applied to the bit lines and sensed by the sense amplifiers.
- The column address decoder selects the values to be sent to the Data output, which is enabled by $\overline{\text{Output Enable}}$.
- During the read operation, all values in the addressed row are restored.

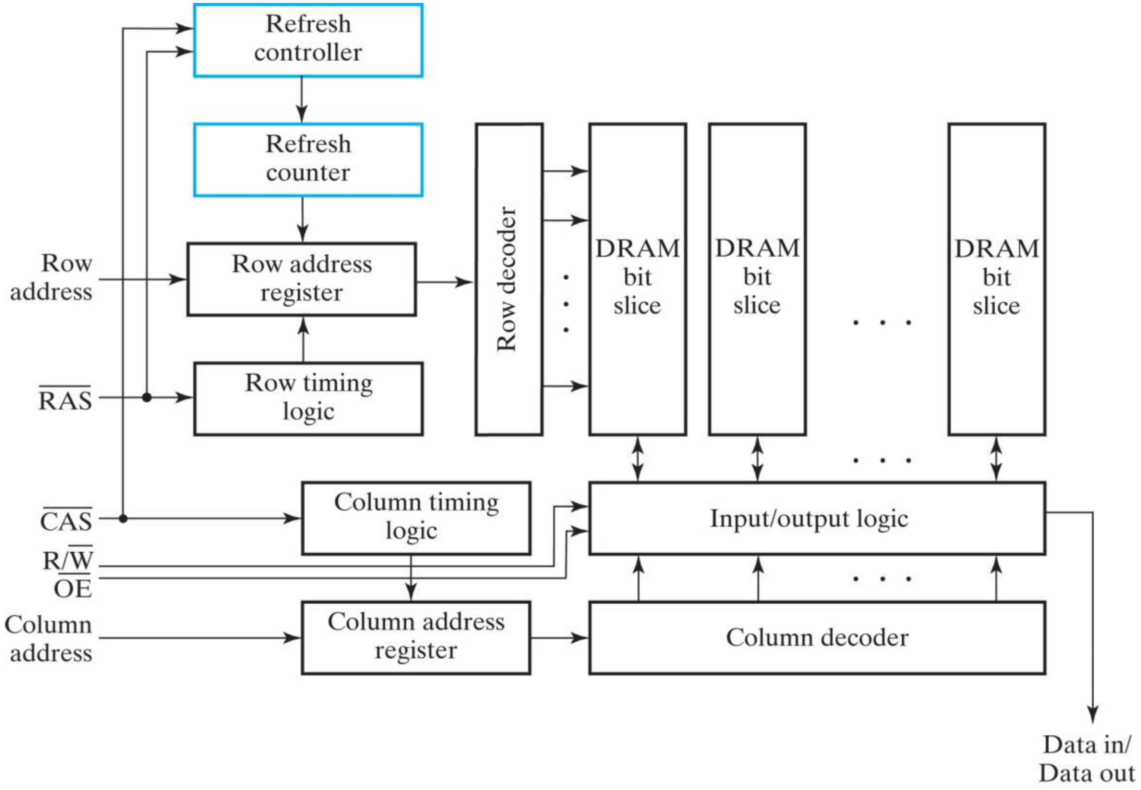


(a) Write cycle

(b) Read cycle

DRAM Refresh

- Additional logic for refresh is shown in blue on block diagram to the right.
- There is a Refresh counter and a Refresh controller.
- The Refresh counter is used to provide the address of the row of DRAM cells to be refreshed.
- It is essential for the refresh modes that require the address to be provided from within the DRAM chip.
- The refresh counter advances on each refresh cycle.
- Due to the number of bits in the counter, when it reaches 2*n* – 1, where *n* is the number of rows in the DRAM array, it advances to 0 on the next refresh.
- The standard ways in which a refresh cycle can be triggered and the corresponding refresh types are as follows:



DRAM Refresh

- The standard ways in which a refresh cycle can be triggered and the corresponding refresh types are as follows:

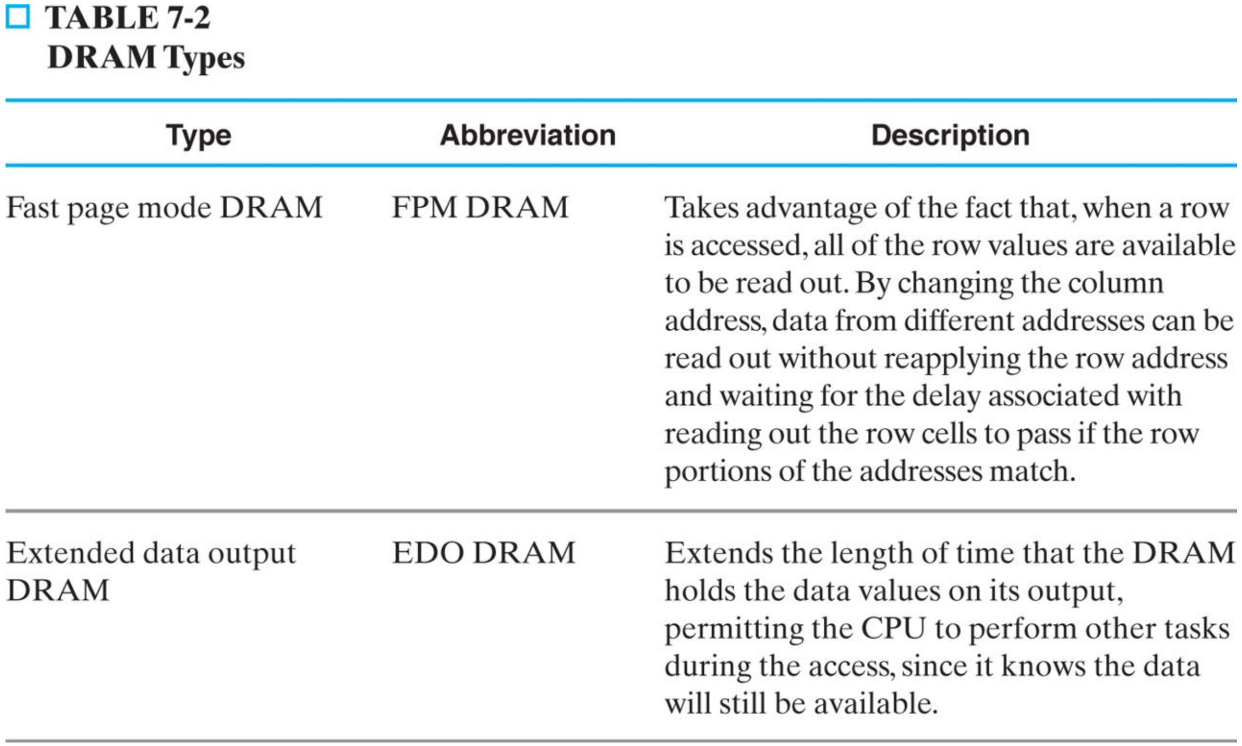
1. **RAS-only refresh:** A row address is placed on the address lines and RAS is changed to 0. In this case, the refresh addresses must be applied from outside the DRAM chip, typically by an IC called a DRAM controller.
2. **CAS-before-RAS refresh:** The CAS is changed from 1 to 0 followed by a change from 1 to 0 on RAS. Additional refresh cycles can be performed by changing RAS without changing CAS. The refresh addresses for this case come from the refresh counter, which is incremented after the refresh for each cycle.
3. **Hidden refresh:** Following a normal read or write, CAS is left at 0 and RAS is cycled, effectively performing a CAS-before-RAS refresh. During a hidden refresh, the output data from the prior read remains valid. Thus, the refresh is hidden. Unfortunately, the time taken by the hidden refresh is significant, so a subsequent read or write operation is delayed.

DRAM Refresh

- In all cases, note that the initiation of a refresh is controlled externally by using the and signals.
- Each row of a DRAM chip requires refreshing within a specified maximum refresh time, typically ranging from 16 to 64 milliseconds (ms).
- Refreshes may be performed at evenly spaced points in the refresh time, an approach called distributed refresh.
- Alternatively, all refreshes may be performed one after the other, an approach called burst refresh.
- For example, a 4M × 4 DRAM has a refresh time of 64 ms and has 4096 rows to be refreshed.
- The length of time to perform a single refresh is 60 ns, and the refresh interval for distributed refresh is 64 ms/4096 = 15.6 microseconds (μs).
- A total time out for refresh of 0.25 ms is used out of the 64 ms refresh interval.
- For the same DRAM, a burst refresh also takes 0.25 ms.
- The DRAM controller must initiate a refresh every 15.6 μs for distributed refresh and must initiate 4096 refreshes sequentially every 64 ms for burst refresh.
- During any refresh cycle, no DRAM reads or writes can occur.
- Since use of burst refresh would halt computer operation for a fairly long period, distributed refresh is more commonly used.

DRAM Types

- Over the last two decades, both the capacity and speed of DRAM have increased significantly.
- The quest for speed has resulted in the evolution of many types of DRAM.
- Several are listed with brief descriptions in following Tables.
- Of the memory types listed, the first two have largely been replaced in the marketplace by the more advanced **SDRAM** and **RDRAM** approaches.
- Our discussion of memory types here will omit the **ECC** feature and focus on **synchronous DRAM**, **double-data-rate synchronous DRAM**, and **Rambus DRAM**.

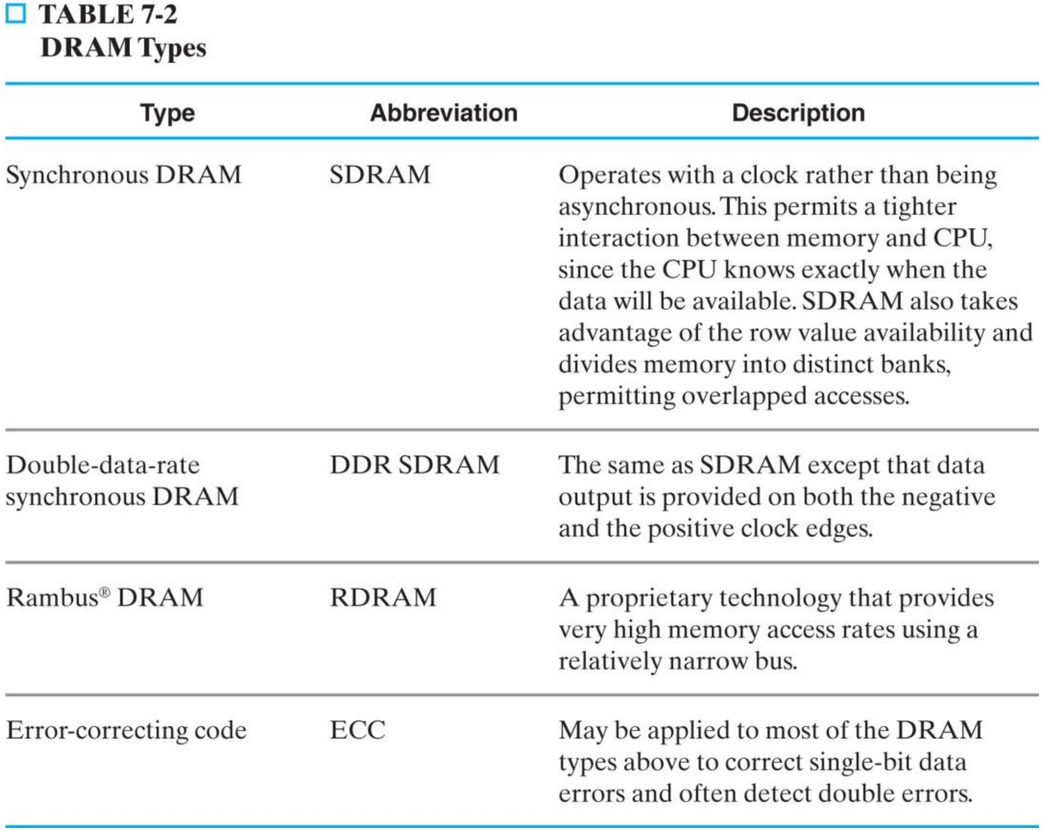


□ **TABLE 7-2**
DRAM Types

Type	Abbreviation	Description
Fast page mode DRAM	FPM DRAM	Takes advantage of the fact that, when a row is accessed, all of the row values are available to be read out. By changing the column address, data from different addresses can be read out without reapplying the row address and waiting for the delay associated with reading out the row cells to pass if the row portions of the addresses match.
Extended data output DRAM	EDO DRAM	Extends the length of time that the DRAM holds the data values on its output, permitting the CPU to perform other tasks during the access, since it knows the data will still be available.

DRAM Types

- First, all three of these DRAM types work well because of the particular environment in which they operate.
- In modern high-speed computer systems, the processor interacts with the DRAM within a memory hierarchy.
- Most of the instructions and data for the processor are fetched from two lower levels of the hierarchy, the L1 and L2 caches.
- These are comparatively smaller SRAM-based memory structures.
- For our purposes, the key issue is that most of the reads from the DRAM are not directly from the CPU, but instead are initiated to bring data and instructions into these caches.
- The reads are in the form of a ***line*** (i.e., some number of bytes in contiguous addresses in memory) that is brought into the cache.
- For example, in a given read, the 16 bytes in hexadecimal addresses 000000 through 00000F would be read.
- This is referred to as a ***burst read***.
- For burst reads, the effective *rate* of reading bytes, which is dependent upon reading bursts from contiguous addresses, rather than the access time is the important measure.
- With this measure, the three DRAM types we are discussing provide very fast performance.
- Second, the effectiveness of these three DRAM types depends upon a very fundamental principle involved in DRAM operation, the reading out of all of the bits in a row for each read operation.
- The implication of this principle is that all of the bits in a row are available after a read using that row if only they can be accessed.
- With these two concepts in mind, the **synchronous DRAM** can be introduced.



□ **TABLE 7-2**
DRAM Types

Type	Abbreviation	Description
Synchronous DRAM	SDRAM	Operates with a clock rather than being asynchronous. This permits a tighter interaction between memory and CPU, since the CPU knows exactly when the data will be available. SDRAM also takes advantage of the row value availability and divides memory into distinct banks, permitting overlapped accesses.
Double-data-rate synchronous DRAM	DDR SDRAM	The same as SDRAM except that data output is provided on both the negative and the positive clock edges.
Rambus® DRAM	RDRAM	A proprietary technology that provides very high memory access rates using a relatively narrow bus.
Error-correcting code	ECC	May be applied to most of the DRAM types above to correct single-bit data errors and often detect double errors.