# Lecture 4

## DRAM TYPES

# Introduction

- 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.
- 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.

# Cont.,

- 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.

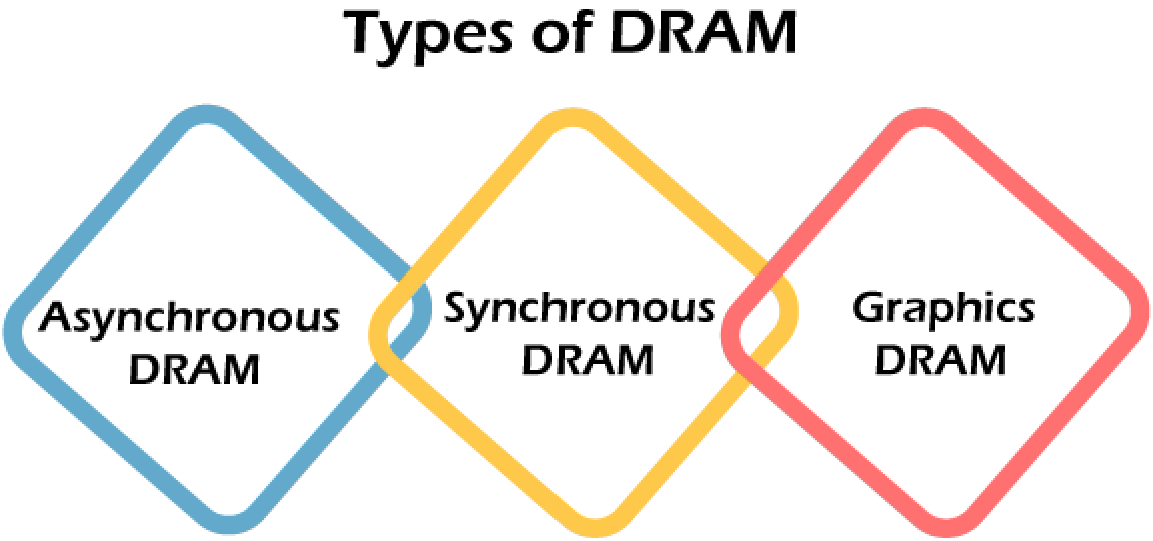
# Cont.,

- our discussion of memory types here will focus on synchronous DRAM, double-data-rate synchronous DRAM, and Rambus® DRAM.
- The effectiveness of these 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”.**
- With these concepts in mind, the synchronous DRAM can be introduced.

# Burst Mode

- It refers to a device is transmitting data repeatedly without going through all the steps required to transmit each piece of data in a separate transaction.
- 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.

# Types of DRAM



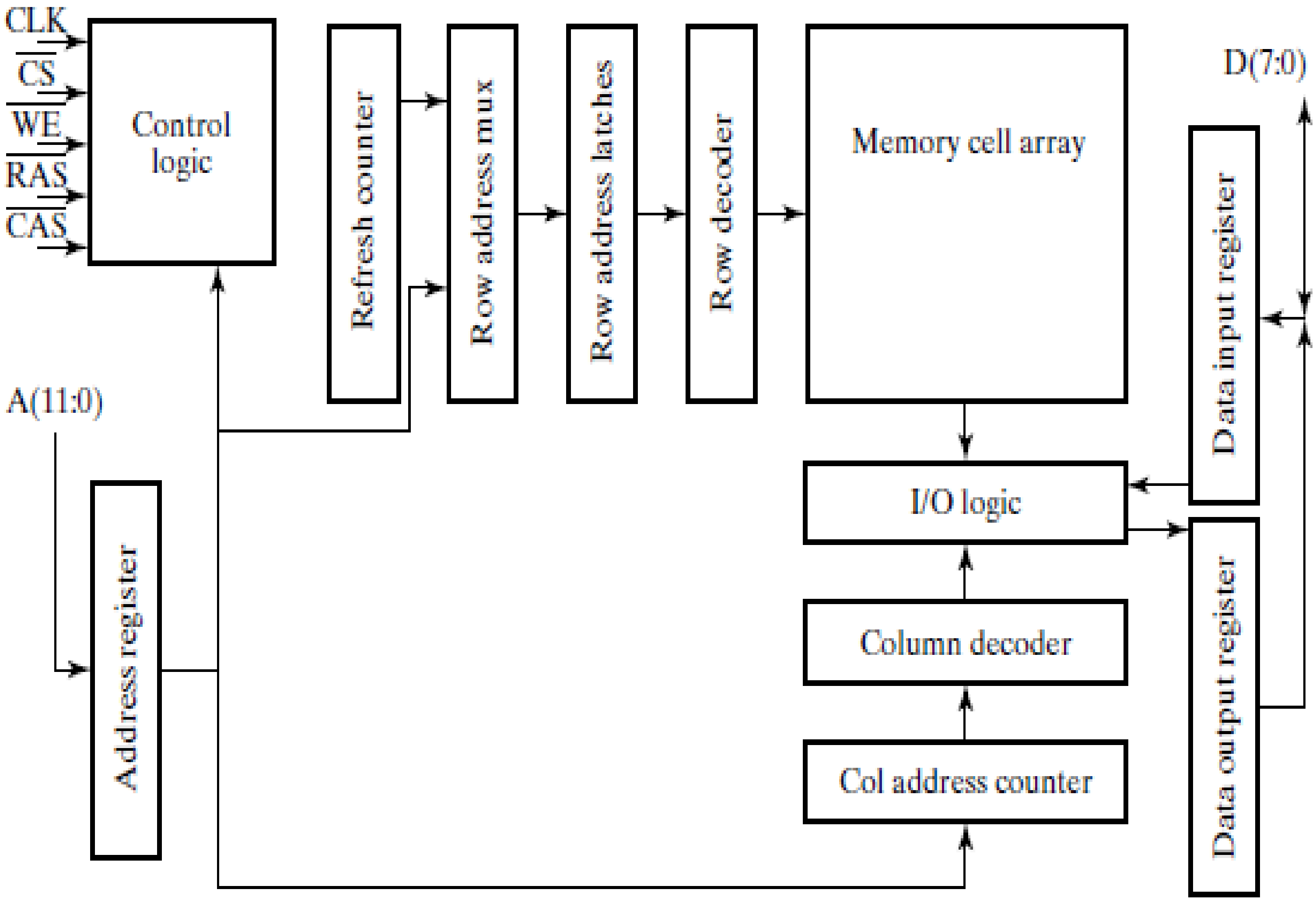
# Cont.,

- **Asynchronous DRAM:** The memory access was not synchronized with the system clock. That's why it's called asynchronous. It was the first type of DRAM in use but was gradually replaced by synchronous DRAM.
- **Synchronous DRAM:** In synchronous DRAM (SDRAM), the clock is synchronized with the memory interface. Synchronous DRAM syncs memory speeds with CPU clock speeds, letting the memory controller know the CPU clock cycle.
- **Graphics DRAM:** Graphics DRAMs are asynchronous and synchronous DRAMs designed for graphics-related tasks such as texture memory and frame buffers found on video cards.

# 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 (Interleaving Memory).

# Block Diagram of a 16 MB SDRAM



# The differences in the Internal Structure with DRAM

- Synchronous registers on the address inputs and the data inputs and outputs.
- In addition, a column address counter has been added, which is key to the operation of the SDRAM (specify the burst length).
- While the control logic may appear to be similar, the control in this case is much more complex, since the SDRAM has a mode control word that can be loaded from the address bus.

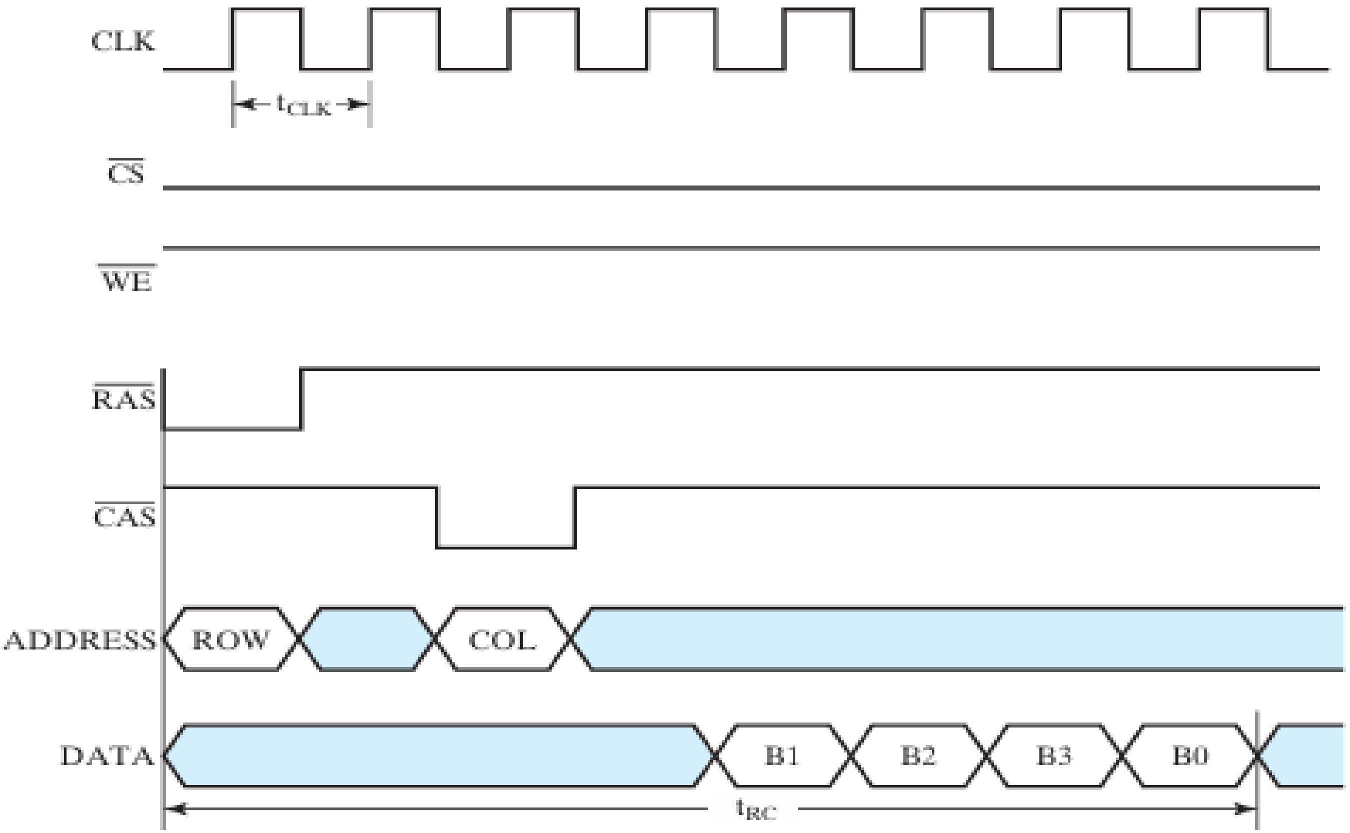
# SDRAM Operation

- Consider 16 MB SDRAM, with no of rows = 13 lines, and no. of columns = 11 lines. It means that we have $2^{13}$ rows, each with $2^{11}$ bytes.
- As with the regular DRAM, the SDRAM applies the row address first, followed by the column address.
- The timing, however, is somewhat different, and some new terminology is used.
- Before performing an actual read operation from a specified column address, the entire row of 2048 bytes specified by the applied row address is read out internally and stored in the I/O logic.
- Internally, this step takes a few clock cycles.

# Cont.,

- Next, the actual read step is performed with the column address applied.
- After an additional delay of a few clock cycles, the data bytes begin appearing on the output, one per clock period.
- The number of bytes that appear, the burst length, has been set by loading a mode control word into the control logic from the address input.

# Timing Diagram for an SDRAM



□ **FIGURE 7-17**
Timing Diagram for an SDRAM

# Where:

- The timing of a burst read cycle with burst length equal to four is shown in Figure.
- The read begins with the application of the row address and the row address strobe (RAS), which causes the row address to be captured in the address register and the reading of the row to be initiated.
- During the first two clock periods, the reading of the row is taking place.

# Cont.,

- During the third clock period, the column address and the column address strobe are applied,
- The column address captured in the address register and the reading of the first data byte initiated.
- The data byte is then available to be read from the SDRAM at the positive clock edge of the fifth clock cycle.
- The second, third, and fourth bytes are available for reading on subsequent clock edges.

# Example

- Compare the byte rate for reading bytes from SDRAM to that of the basic DRAM. Assume that the read cycle time $t_{RC}$ for the basic DRAM is 60 ns and that the clock period $t_{CLK}$ for the SDRAM is 7.5 ns.

**Solution:**

- The byte rate for the basic DRAM is one byte per 60 ns, or 16.67 MB/sec.

- For the SDRAM: 60 ns = 8 clock cycles, it means that SDRAM can read **four** bytes, giving a byte rate of 66.67 MB/sec.

# Cont.,

- If the burst is eight instead of four bytes,

  a read cycle time of 90 ns is required, giving a byte rate of 88.89 MB/sec.
- Finally, if the burst is the entire 2048-byte row of the SDRAM, the read cycle time becomes 60 + (2048 − 4) × 7.5 = 15,390 ns,
- Giving a byte rate of 133.07 MB, which approaches the limit of one byte per 7.5 ns clock period.

# Example

- Memory data path width: 1 word = 4 bytes
  Burst size: 8 words = 32 bytes
  Memory clock frequency: 5 ns
  Latency time (from application of row address until first word available): 4 clock cycles
  Read cycle time: (4 + 8) x 5 ns = 60 ns
  Memory Bandwidth: 32/(60 x $10^{-9}$) = 533 Mbytes/sec

# 2- Double Data Rate Synchronous DRAM

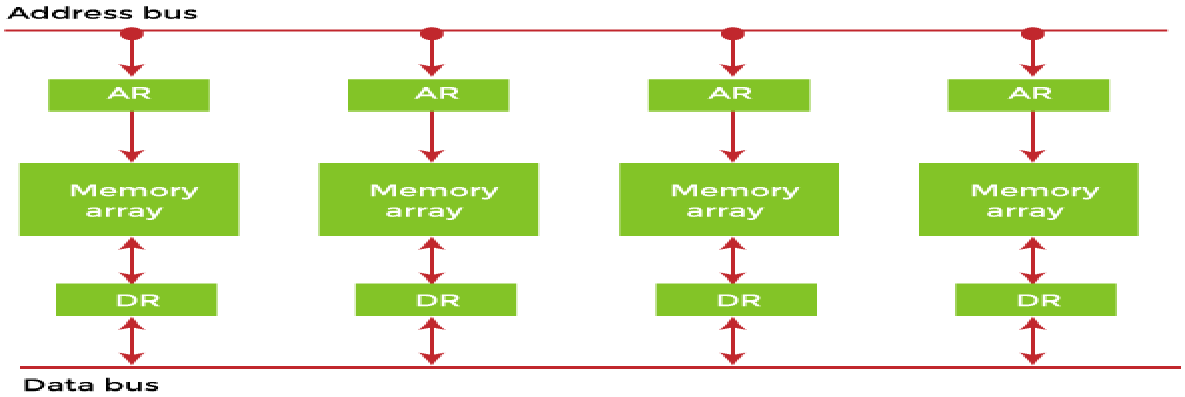
- Transfers data on both edges of the clock
- Provides a transfer rate of 2 data words per clock cycle
- Example: Same as for synchronous DRAM
  - Read cycle time = 60 ns
  - Memory Bandwidth: $(2 \times 32)/(60 \times 10^{-9}) = 1066$ Mbytes/sec

# How to Improve Bandwidth in DDRAM

- If the actual accesses needed are to different rows of the RAM, the delay from the application of the RAS pulse to read out the first byte of data is significant and leads to performance well below the limit.
- This can be partially offset by breaking up the memory into multiple banks, where each bank performs the row read independently.
- Provided that the row and bank addresses are available early enough, row reads can be performed on one or more banks while data is still being transferred from the currently active row.
- When the column reads from the currently active row are complete, data can potentially be available immediately from other banks, permitting an uninterrupted flow of data from the memory.

# Interleaved Memory

- Interleaved memory is designed to compensate for the relatively slow speed of dynamic random-access memory (DRAM) by spreading memory addresses evenly across memory banks.
- In this way, contiguous memory reads and writes use each memory bank, resulting in higher memory throughput due to reduced waiting for memory banks to become ready for the operations.
- It is different from multi-channel memory architectures, primarily as interleaved memory does not add more channels between the main memory and the memory controller.



- In Interleaved memory, virtual address 0 will be with the first bank, 1 with the second memory bank, 2 with the third bank and 3 with the fourth, and then 4 with the first memory bank again.
- Hence, the CPU can access alternate sections immediately without waiting for memory to be cached.
- There are multiple memory banks that take turns for the supply of data.
- Example: If the CPU requests an 8-word burst starting at memory address X, interleaving might access the memory in the following way: Bank 0: Words X, X+2, X+4, X+6 Bank 1: Words X+1, X+3, X+5, X+7.
- These two memory banks can then simultaneously transfer data to the CPU, reducing the overall access time.

# RAMBUS® DRAM (RDRAM)

- As microprocessors get faster, designers are using wider, high-speed buses to transfer data quickly enough to prevent the CPU from stalling.
- It's not that new, but it seems new because Intel started to use it with its Pentium 4 processors and 800-series chipset.
- Rambus memory is integrated onto Rambus Inline Memory Modules (RIMMs).
- RDRAM is known as a *narrow channel* system because data is transferred only 2 bytes (16 bits) at a time.
- This might seem small, but those 2 bytes move extremely fast!

# Cont.,

- The Rambus data bus is 16 bits wide, as opposed to the more typical 32 or 64 bits wide.
- Additionally, Rambus memory sends data more frequently.
- It reads data on both the rising and falling edges of the clock signal.
- RDRAM chips use the processor's memory bus timing frequency, not the motherboard clock.
- RDRAM ICs are designed to be integrated into a memory system that uses a packet-based bus for the interaction between the RDRAM ICs and the memory bus to the processor.
-

# Cont.,

- Each bus of which operates independently and supports serial packet communications.
- The primary components of the bus are a 3-bit path for the row address, a 5-bit path for the column address, and a 16-bit or 18-bit path for data.
- Pipeline Operation: Rambus memory operates in a pipelined manner, where multiple requests for data can be processed simultaneously without waiting for each individual piece of data to be fully transferred before the next request starts.

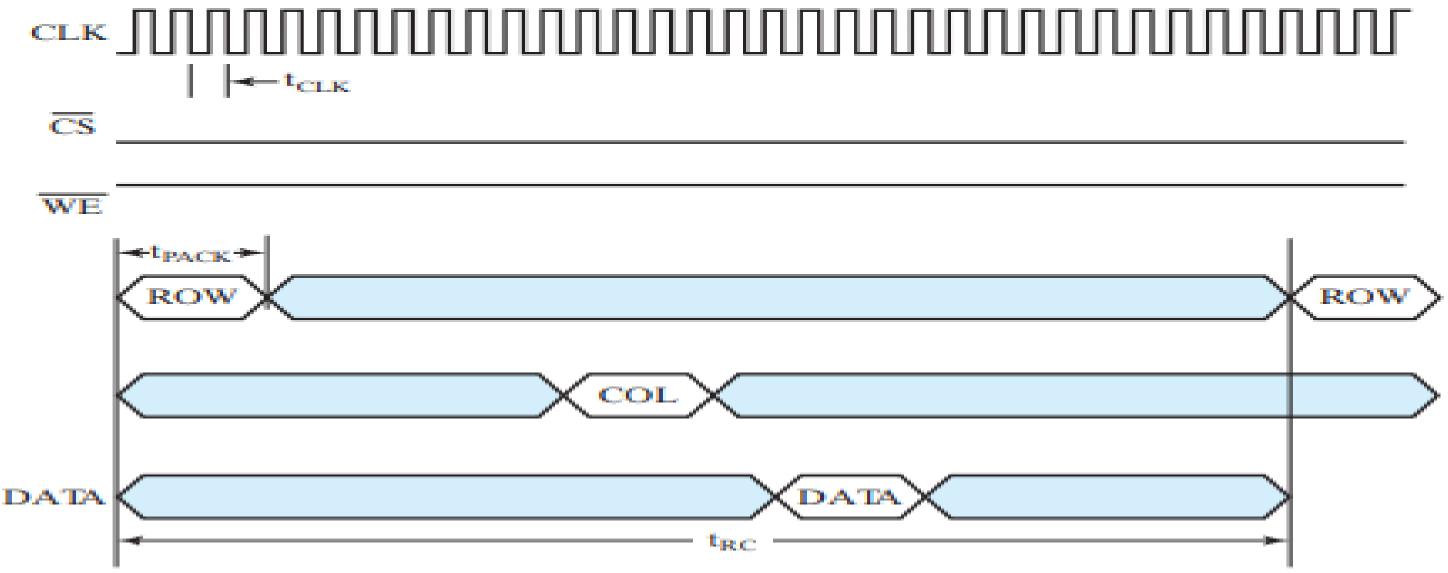


FIGURE 7-18
Timing of a 16 MB RDRAM

- Due to the sophisticated electronic design of the RAMBUS system, we can consider a clock period of 1.875 ns.
- Thus, the time for transmission of a packet is:

  $t_{PACK}$ = 4 * 1.875 = 7.5 ns.
- The cycle time for accessing a single data packet of 8 byte pairs or 16 bytes is 32 clock cycles or 60 ns, as shown in Figure 7-18.
- The corresponding byte rate is 266.67 MB/s.
- If four of the byte packets are accessed from the same row, the rate increases to 1.067 GB/s.

# RDRAM Vs. DDRAM

**Architecture:**

- **Rambus (RDRAM):**
  - Developed by Rambus Inc.
  - Uses a high-speed, point-to-point connection.
  - Operates at higher clock speeds, allowing for higher bandwidth.
  - Typically features a narrow data bus (e.g., 16 bits).
- **DDR (DDR SDRAM):**
  - Developed as a standard by JEDEC.
  - Uses a more traditional parallel architecture with a wider data bus (e.g., 64 bits).
  - Transfers data on both the rising and falling edges of the clock cycle, effectively doubling the data rate.

# Performance:

- **Rambus:**
  - Initially offered higher performance due to its bandwidth advantages.
  - Achieved speeds around 800 MT/s or more in early implementations.
  - Limited by its higher cost and complexity of implementation.
- **DDR:**
  - Became the standard for mainstream memory due to its balance of performance and cost.
  - DDR, DDR2, DDR3, DDR4, and now DDR5 each brought improvements in speed and efficiency, with current DDR5 capable of over 8400 MT/s.
  - More widely adopted, resulting in better economies of scale.

# Cost and Adoption:

- **Rambus:**
  - Initially expensive and required specialized chipsets.
  - Limited adoption primarily in certain high-performance applications (e.g., some gaming consoles and high-end workstations).
  - Eventually overshadowed by DDR due to cost and compatibility issues.
- **DDR:**
  - Became the de facto standard for most consumer and enterprise applications.
  - More cost-effective due to widespread manufacturing and integration into motherboards.

# Compatibility:

- **Rambus:**
  - Requires specific motherboards and chipsets designed for RDRAM, leading to compatibility issues.
  - Less flexible in terms of upgrades and interoperability.
- **DDR:**
  - Supported by a wide range of motherboards and chipsets, making upgrades easier.
  - Different generations of DDR are usually backward compatible to some extent, though often with reduced speeds.

# Future:

- **Rambus:**
  - While it has been phased out in consumer markets, Rambus technology is still used in some niche applications.
  - The company now focuses on licensing its technology rather than manufacturing memory chips.
- **DDR:**
  - Continues to evolve with ongoing improvements in speed, efficiency, and power consumption.
  - DDR5 is currently in widespread use, with research into future technologies like DDR6.

# Conclusion:

- While Rambus offered high performance in its time, the widespread adoption, cost-effectiveness, and evolving capabilities of DDR memory have made it the dominant technology in the market.
- RDRAM is largely a historical footnote in the evolution of computer memory, while DDR continues to develop and thrive in various applications.

# Summary:

- DRAM often used as a part of a memory hierarchy.
- Reads from DRAM bring data into lower levels of the hierarchy.
- Transfers from DRAM involve multiple consecutively addressed words.
- Many words are internally read within the DRAM ICs using a single row address and captured within the memory.
- This read involves a fairly long delay.
- These words are then transferred out over the memory data bus using a series of clocked transfers.
- These transfers have a low delay, so several can be done in a short time.
- The column address is captured and used by a synchronous counter within the DRAM to provide consecutive column addresses for the transfers

### □ 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. |
| 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. |

# *Course Project:*

***Design a DRAM controller with the following properties:***

- Separation of the address into row address and column address and timing their application
- Providing RAS and CAS and timing their application
- Performing refresh operations at required intervals
- Providing status signals to the rest of the system (e.g., indicating whether or not the memory is active or is busy performing refresh)

# Assignments

- Explain in details:

    - Interleaved Memory