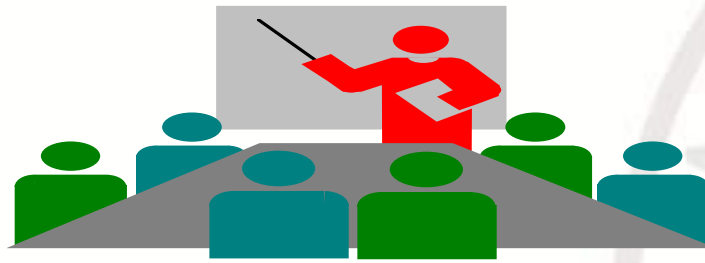




浙江大学  
ZHEJIANG UNIVERSITY



数字逻辑设计

# LOGIC and Computer Design Fundamentals

## CHAPTER 7 Memory Basics

施青松

Asso. Prof. Shi Qingsong

College of Computer Science and Technology, Zhejiang University

[zjsqs@zju.edu.cn](mailto:zjsqs@zju.edu.cn)

---

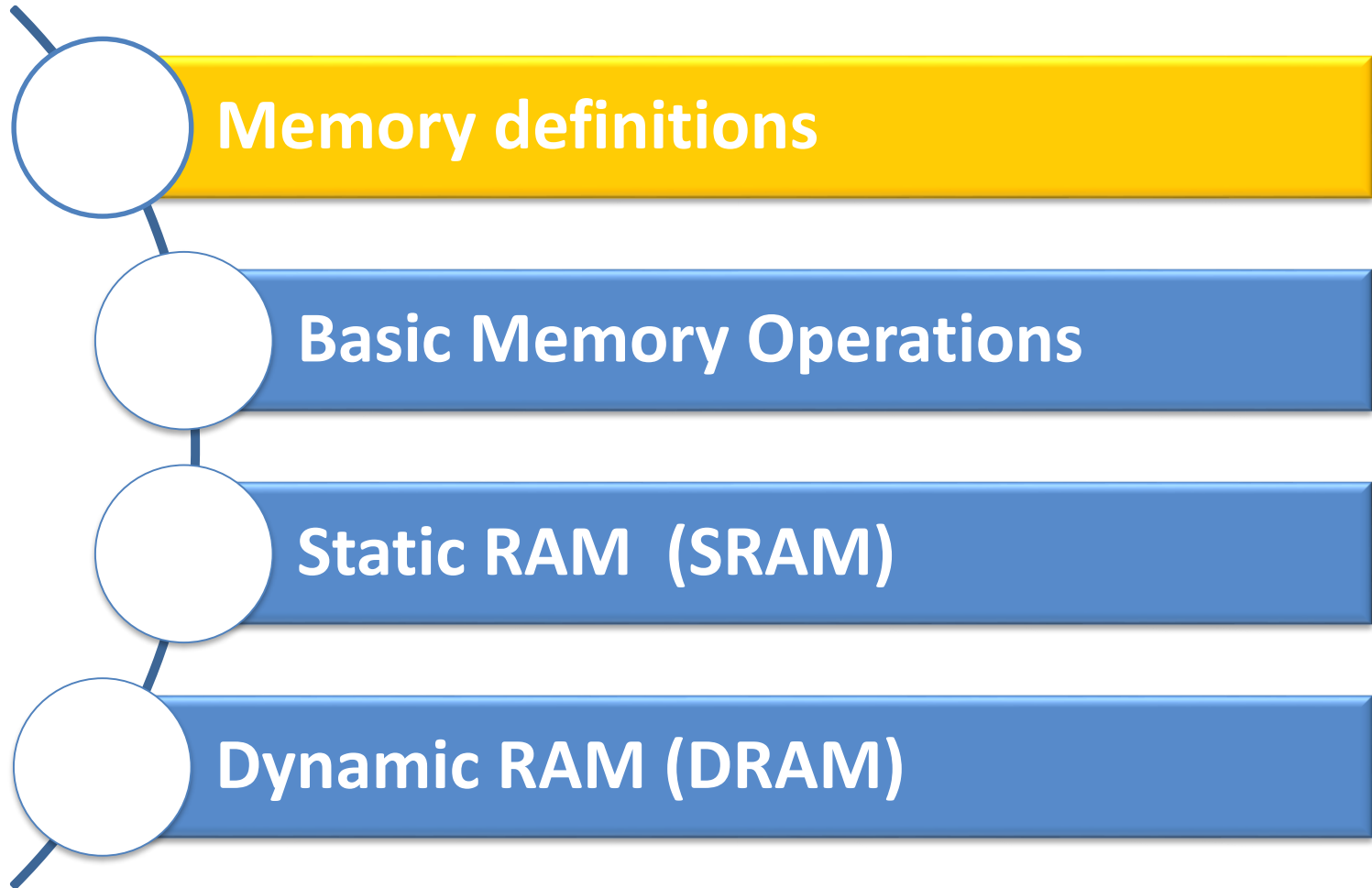
# Overview

---

- ❑ **Memory definitions**
- ❑ **Random Access Memory (RAM)**
- ❑ **Static RAM (SRAM) integrated circuits**
  - Cells and slices
  - Cell arrays and coincident selection
  - Arrays of SRAM integrated circuits
- ❑ **Dynamic RAM (DRAM) integrated circuits**
- ❑ **DRAM Types**
  - Synchronous (SDRAM)
  - Double-Data Rate (DDR SDRAM)
  - RAMBUS DRAM (RDRAM)
  - Arrays of DRAM integrated circuits



# Course Outline





# Memory Definitions

- ❑ **Memory** — A collection of storage cells together with the necessary circuits to transfer information to and from them.
- ❑ **Memory Organization** — the basic architectural structure of a memory in terms of how data is accessed.
- ❑ **Random Access Memory (RAM)** — a memory organized such that data can be transferred to or from any cell (or collection of cells) in a time that is not dependent upon the particular cell selected.
- ❑ **Memory Address** — A vector of bits that identifies a particular memory element (or collection of elements).

# Data Elements and Operations

## □ Typical data elements are:

- *bit* — a single binary digit
- *byte* — a collection of eight bits accessed together
- *word* — a collection of binary bits whose size is a typical unit of access for the memory. It is typically a power of two multiple of bytes (e.g., 1 byte, 2 bytes, 4 bytes, 8 bytes, etc.)

## □ *Memory Data* — a bit or a collection of bits to be stored into or accessed from memory cells.

## □ *Memory Operations* — operations on memory data supported by the memory unit. Typically, *read* and *write* operations over some data element (bit, byte, word, etc.).

# Memory Organization

- ❑ Organized as an indexed *array of words*. Value of the **index** for each word is the *memory address*.
- ❑ Often organized to fit the needs of a particular computer architecture. Some historically significant computer architectures and their associated memory organization:
  - Digital Equipment Corporation PDP-8 – used a 12-bit address to address 4096 12-bit words.
  - IBM 360 – used a 24-bit address to address 16,777,216 8-bit bytes, or 4,194,304 32-bit words.
  - Intel 8080 – (8-bit predecessor to the 8086 and the current Intel processors) used a 16-bit address to address 65,536 8-bit bytes.

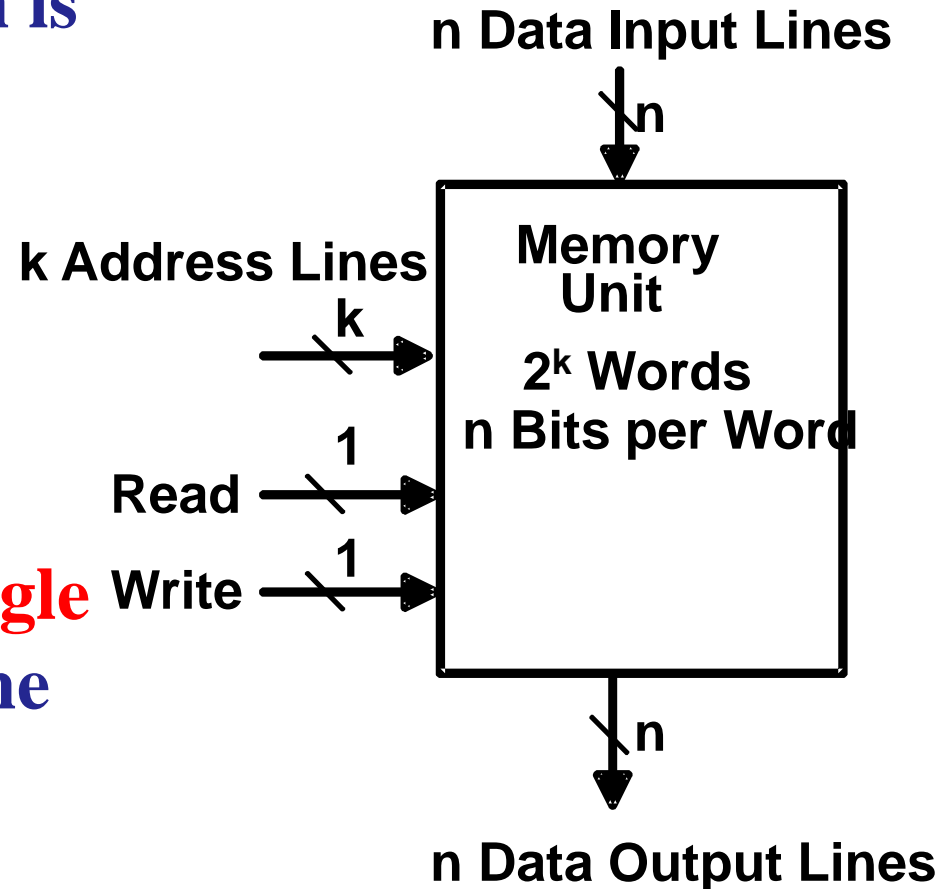
# Memory Block Diagram

□ A basic memory system is shown here:

□  $k$  *address lines* are decoded to address  $2^k$  words of memory.

□ Each word is  $n$  bits.

□ **Read** and **Write** are **single** control lines defining the simplest of memory operations.



# Memory Organization Example

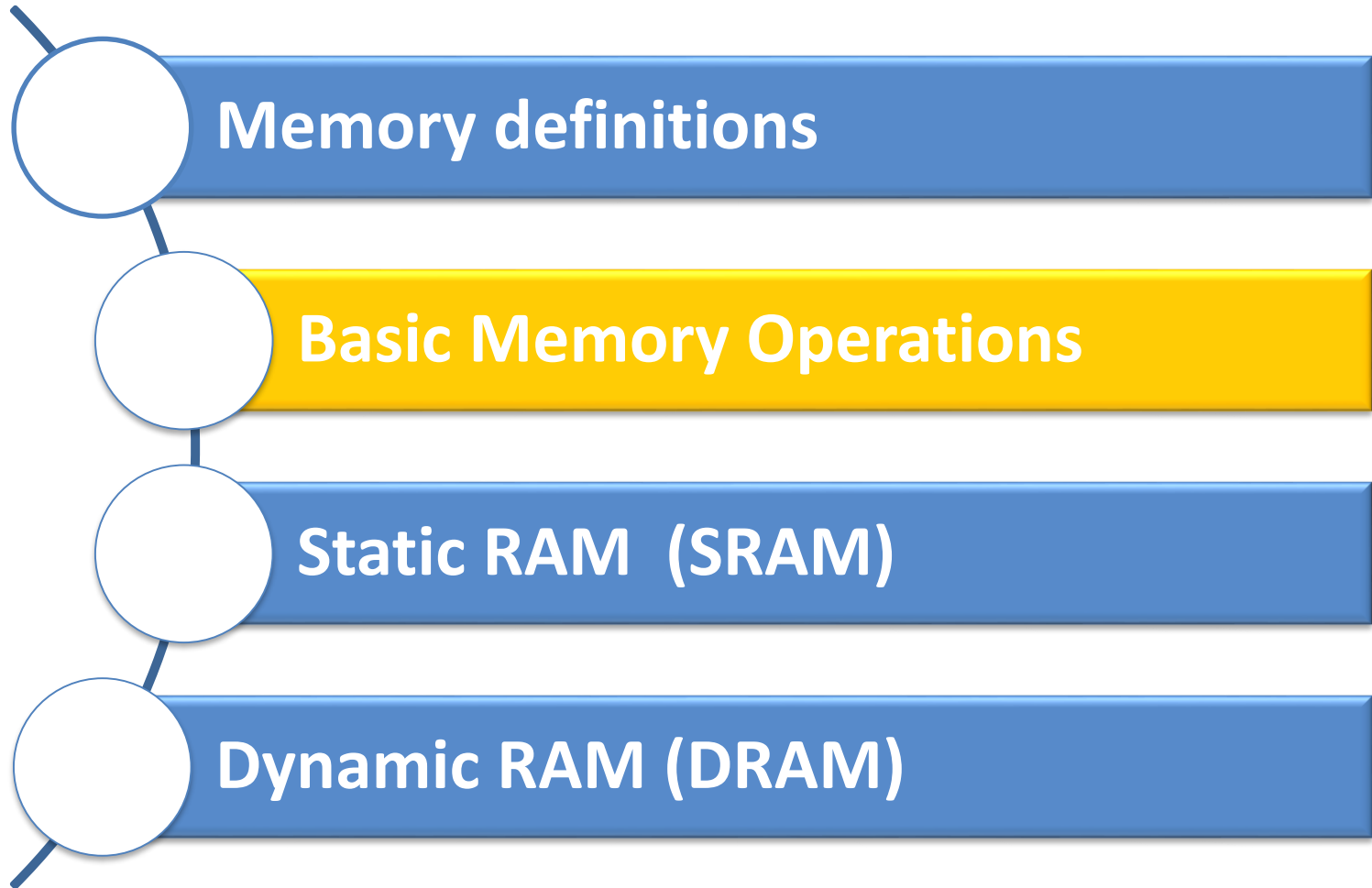
## □ Example memory contents:

- A memory with 3 address bits & 8 data bits has:
- $k = 3$  and  $n = 8$  so  $2^3 = 8$  **addresses** labeled 0 to 7.
- $2^3 = 8$  **words** of 8-bit data

Memory Address		Memory Content
Binary	Decimal	
0 0 0	0	1 0 0 0 1 1 1 1
0 0 1	1	1 1 1 1 1 1 1 1
0 1 0	2	1 0 1 1 0 0 0 1
0 1 1	3	0 0 0 0 0 0 0 0
1 0 0	4	1 0 1 1 1 0 0 1
1 0 1	5	1 0 0 0 0 1 1 0
1 1 0	6	0 0 1 1 0 0 1 1
1 1 1	7	1 1 0 0 1 1 0 0



# Course Outline



# Basic Memory Operations

## □ Memory operations require the following:

- **Data** — data written to, or read from, memory as required by the operation.
- **Address** — specifies the memory location to operate on. The address lines carry this information into the memory. Typically:  $n$  bits specify locations of  $2^n$  words.
- **An operation** — Information sent to the memory and interpreted as control information which specifies the type of operation to be performed.
  - Typical operations are **READ** and **WRITE**.
  - Others are READ followed by WRITE and a variety of operations associated with delivering blocks of data.
  - **Operation signals may also specify timing info.**

# Basic Memory Operations : R/W

## □ **Read Memory** — an operation that reads a data value stored in memory:

- Place a valid address on the address lines.
- Wait for the read data to become stable.

Chip select CS	Read/ $\overline{\text{Write}}$ R/ $\overline{\text{W}}$	Memory operation
0		None
1	0	Write to selected word
1	1	Read from selected word

## □ **Write Memory** — an operation that writes a data value to memory:

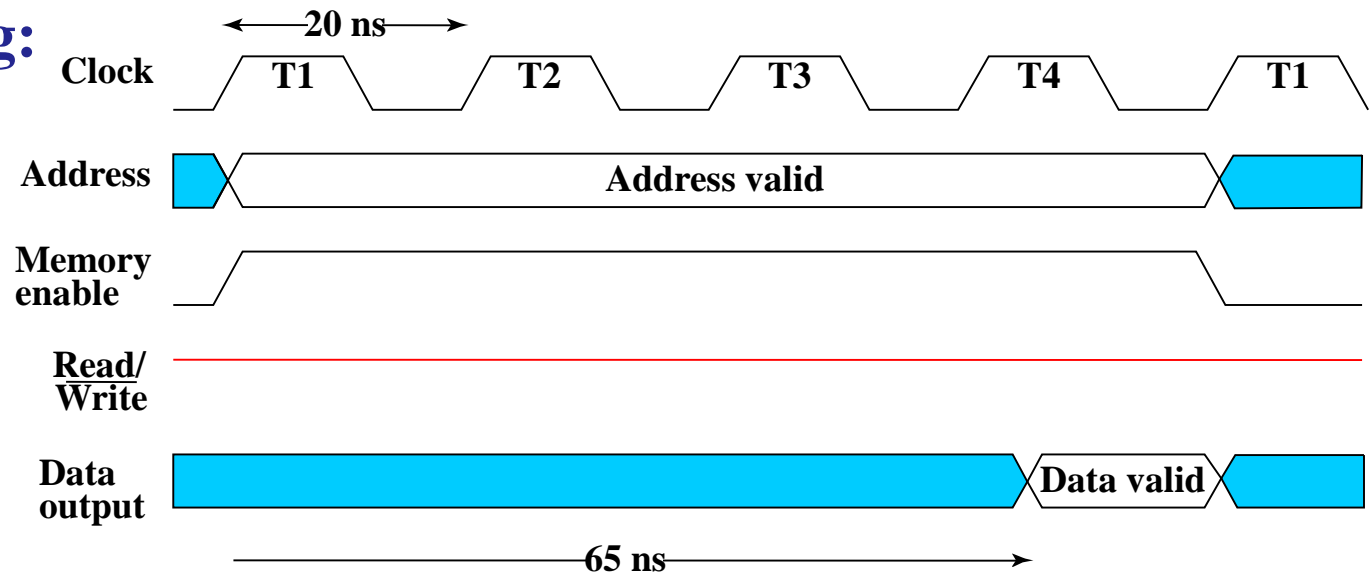
- Place a valid address on the address lines and valid data on the data lines.
- Toggle the memory write control line

## □ Sometimes the read or write **enable line** is defined as a clock with precise timing information (e.g. Read Clock, Write Strobe).

- Otherwise, it is just an interface signal.
- Sometimes memory must acknowledge that it has completed the operation.

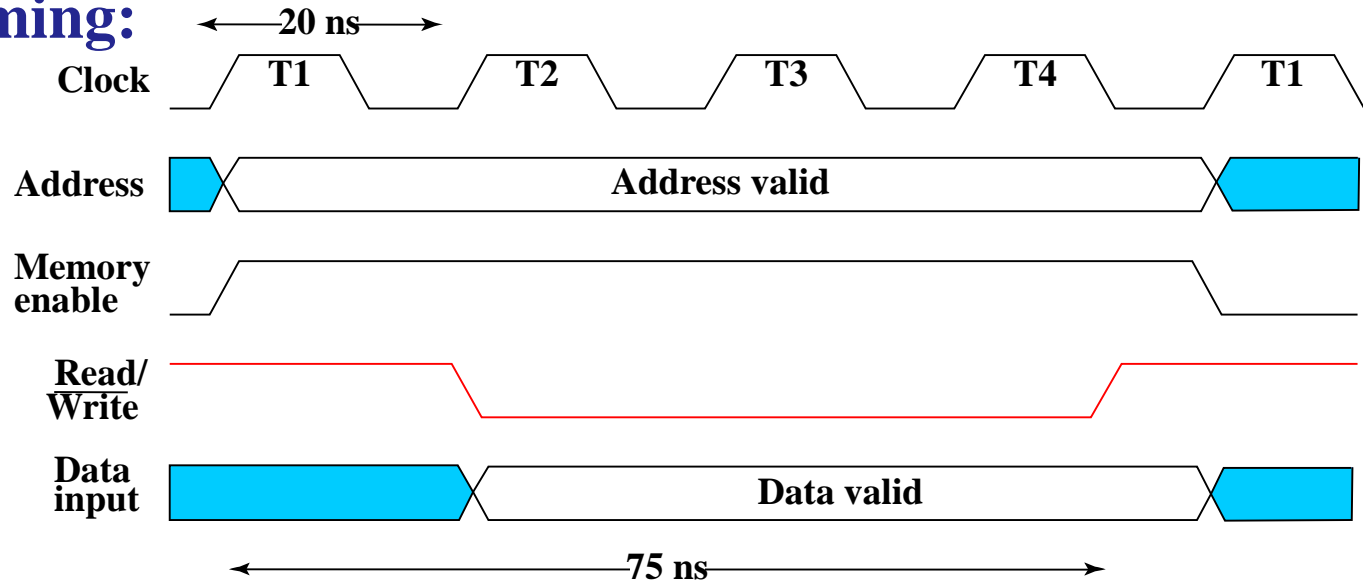
# Memory Operation Timing: Read

- ❑ Most basic memories are *asynchronous*
  - Storage in latches or storage of electrical charge
  - No clock
- ❑ Controlled by control inputs and address
- ❑ Timing of signal changes and data observation is critical to the operation
- ❑ Read timing:



# Memory Operation Timing : Write

## □ Write timing:



## □ Critical times measured with respect to edges of write pulse (1-0-1):

- **Address** must be established at least a specified time before 1-0 and held for at least a specified time after 0-1 to avoid disturbing stored contents of other addresses
- **Data** must be established at least a specified time before 0-1 and held for at least a specified time after 0-1 to write correctly

# RAM Integrated Circuits

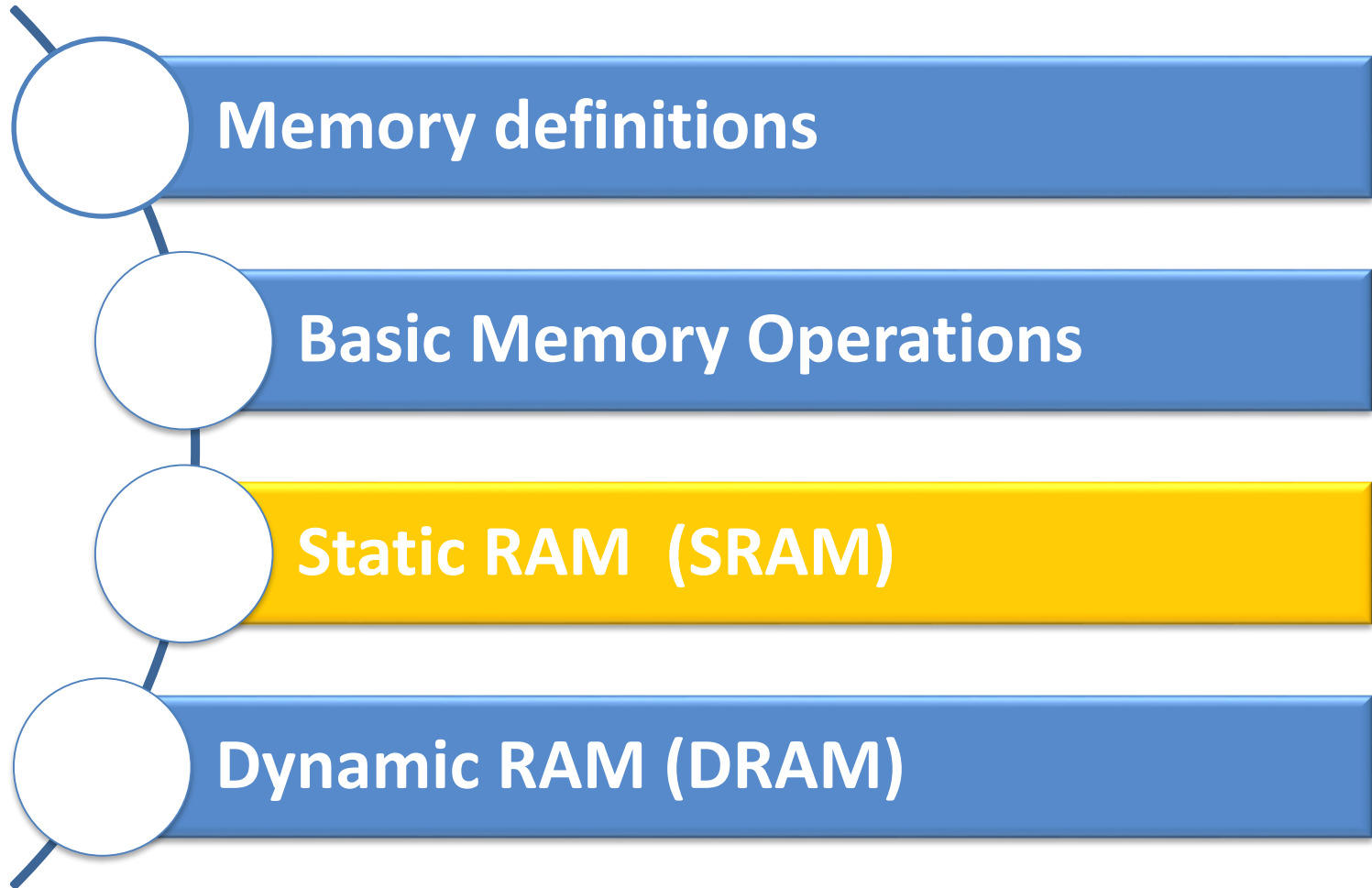
## □ Types of random access memory

- *Static* – information stored in **latches**
- *Dynamic* – information stored as **electrical charges** on capacitors
  - Charge “leaks” off
  - Periodic *refresh* of charge required

## □ Dependence on Power Supply

- *Volatile* – loses stored information when power turned off
- *Non-volatile* – retains information when power turned off

# Course Outline

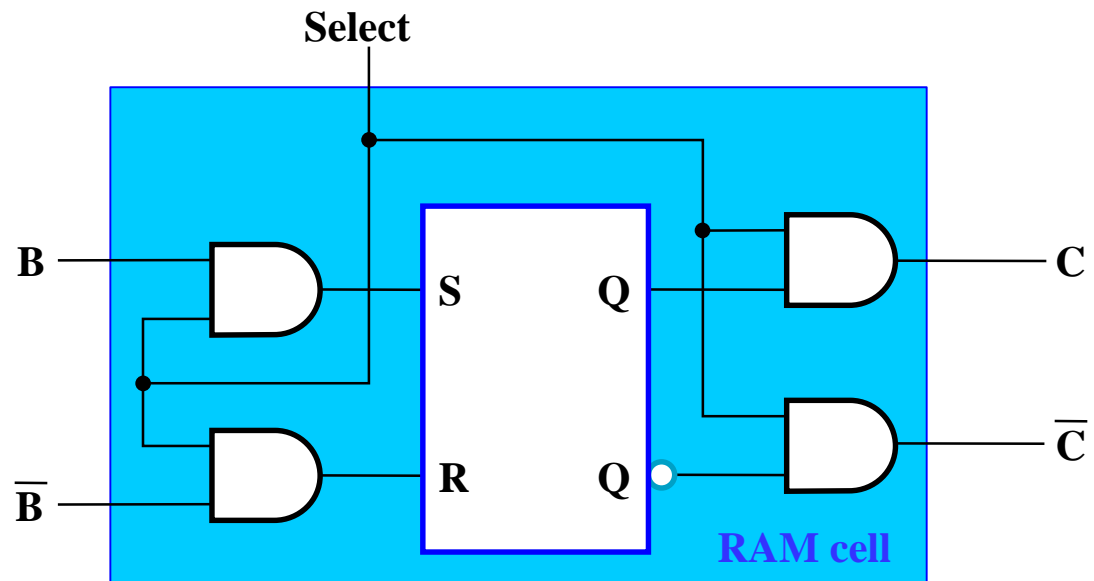


# Static RAM Cell

□ Array of storage cells used to implement static RAM

□ Storage Cell

- SR Latch
- Select input for control
- Dual Rail Data Inputs  $B$  and  $\overline{B}$
- Dual Rail Data Outputs  $C$  and  $\overline{C}$

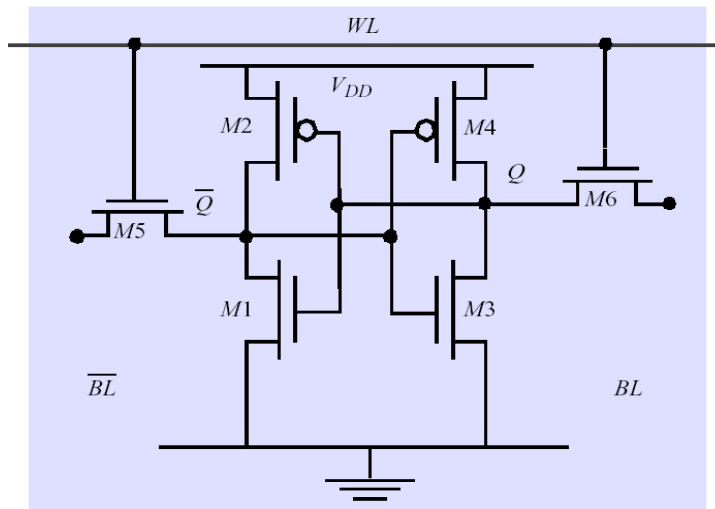




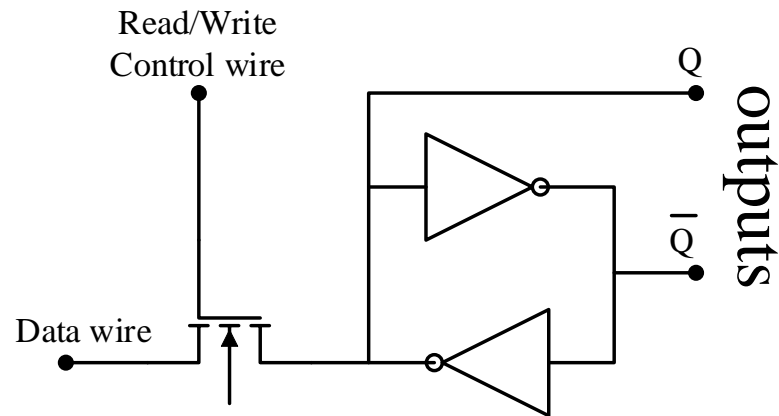
# Really SRAM cell

## □ SRAM:

- value is stored on a pair of inverting gates
- very fast but takes up more space than DRAM (4 to 6 transistors)



(a) the basic structure of SRAM



(b) SRAM equivalent structure

# Static RAM Bit Slice

□ Represents all circuitry that is required for  $2^n$  1-bit words

■ Multiple RAM cells

■ Control Lines:

□ Word select  $i$   
— one for each word

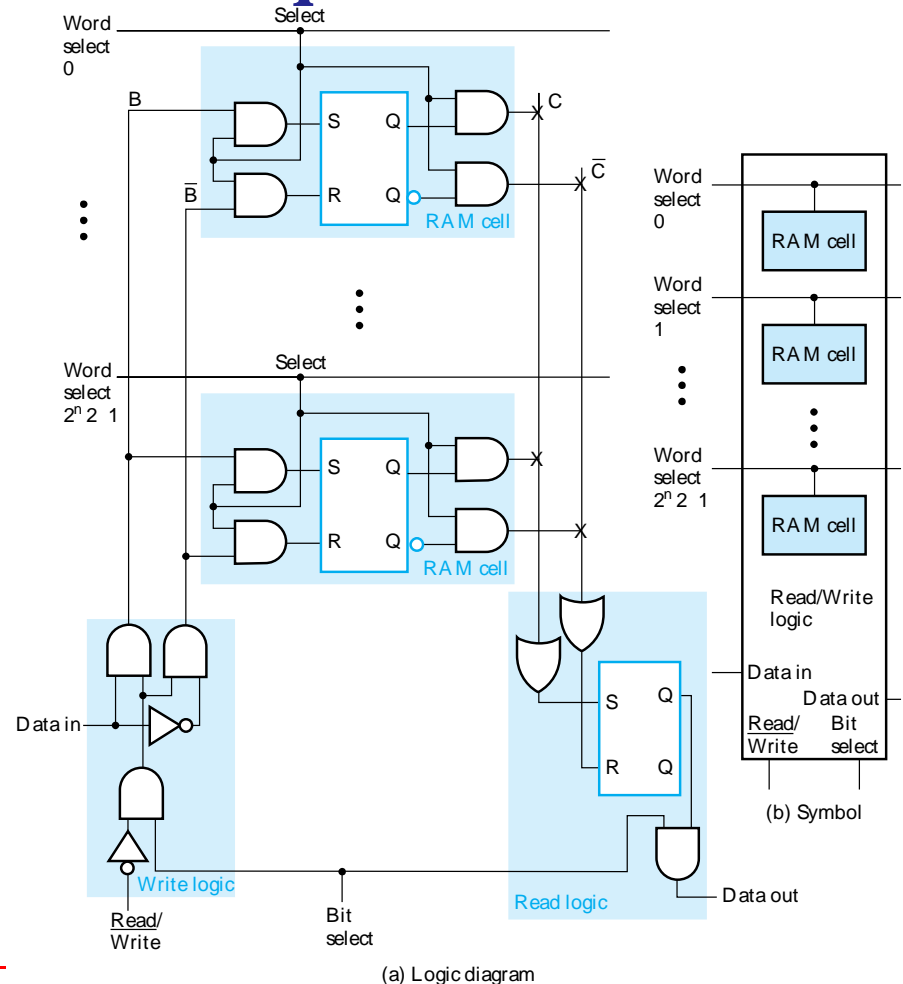
□ **Read / Write**

□ Bit Select

■ Data Lines:

□ Data in

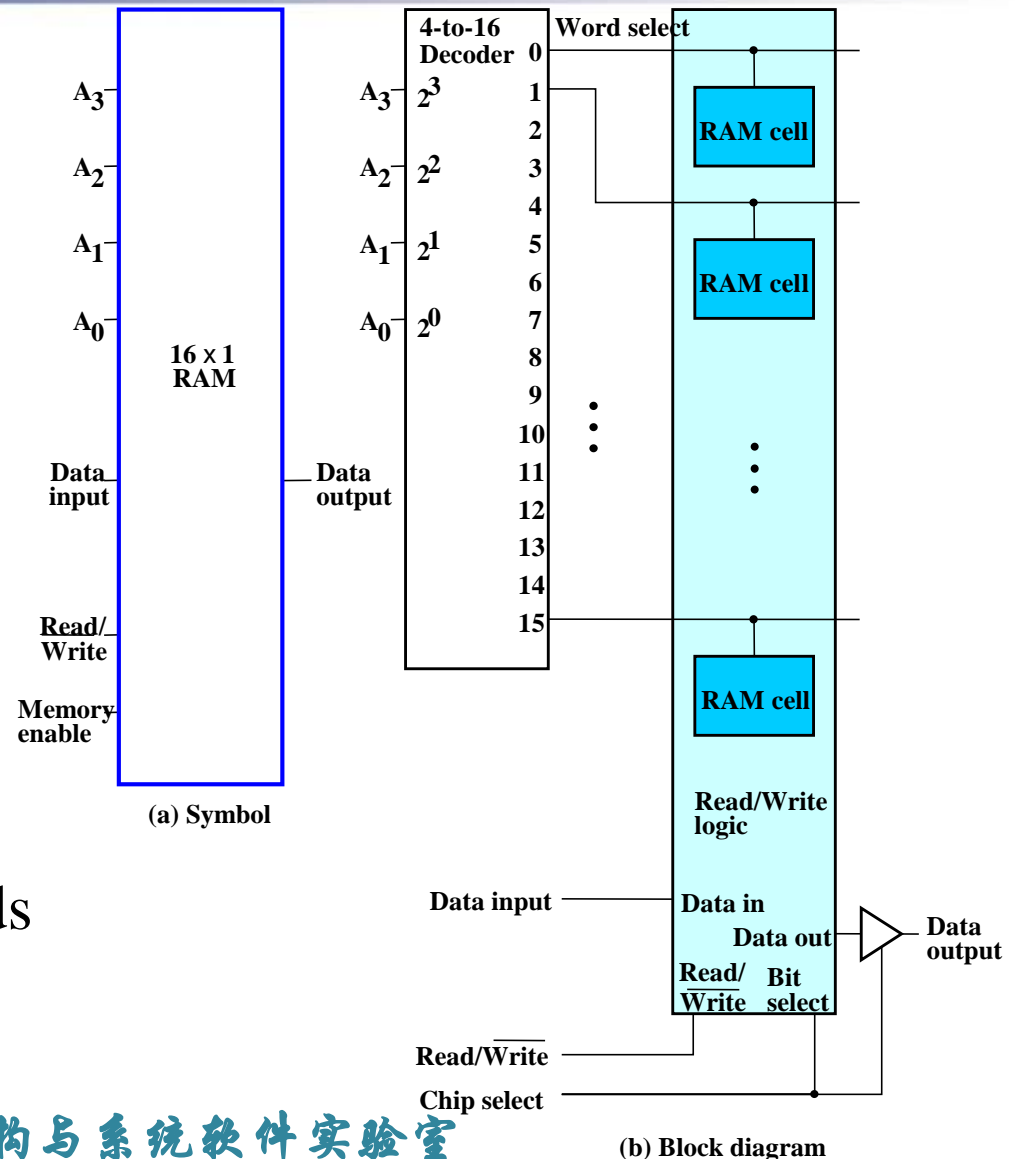
□ Data out



Word selected control line too much\_

# 2<sup>n</sup>-Word × 1-Bit RAM IC

- To build a RAM IC from a RAM slice, we need:
  - **Decoder** 译 decodes the n address lines to 2<sup>n</sup> word select lines
  - A **3-state buffer**
  - on the data output permits RAM ICs to be combined into a RAM with  $c \times 2^n$  words

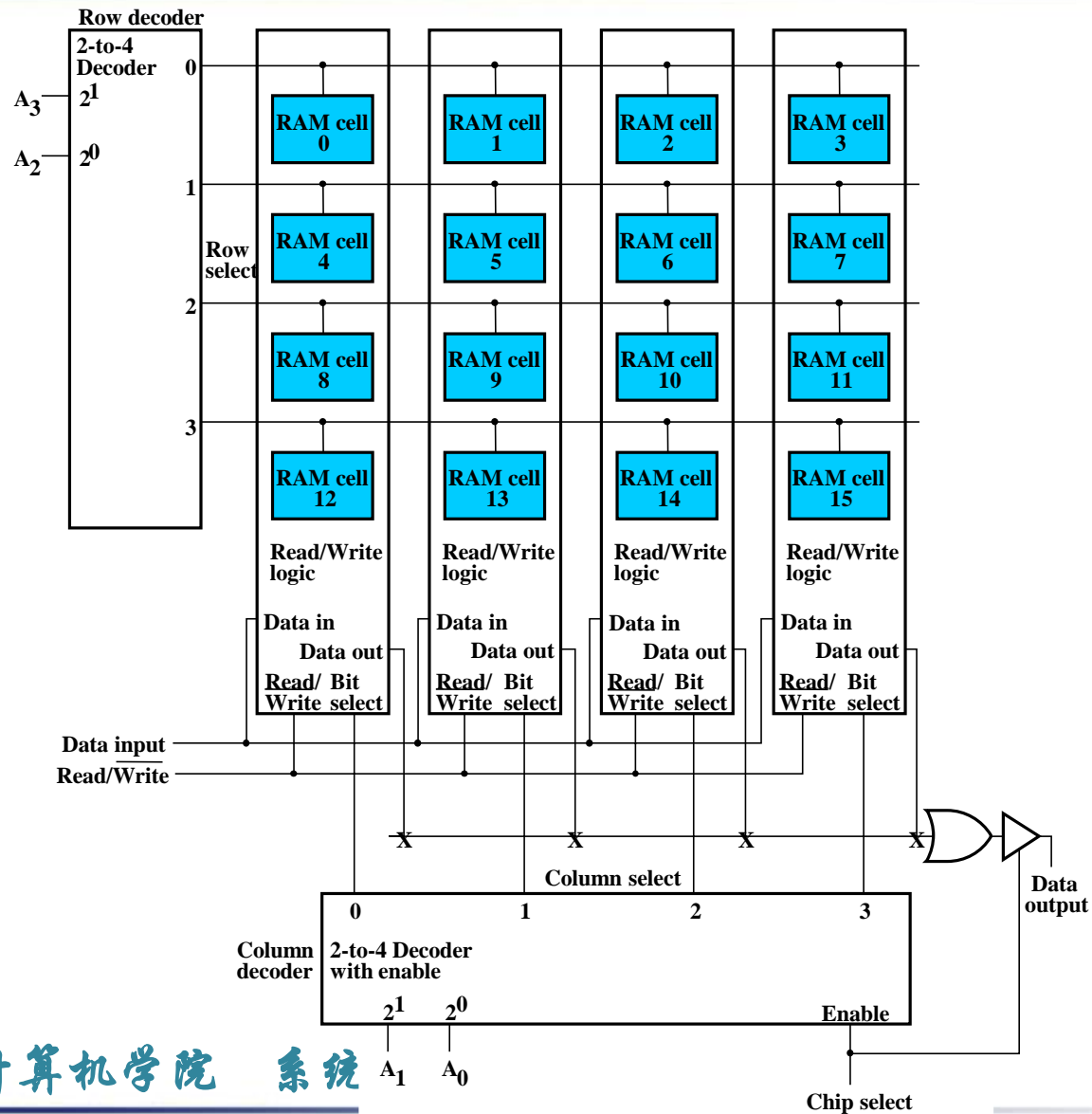


# Cell Arrays and Coincident Selection

- **Memory arrays can be very large =>**
  - Large decoders
  - Large fanouts for the bit lines
  - The decoder size and fanouts can be reduced by approximately  $\sqrt{n}$  by using a **coincident selection** in a **2-dimensional** array
  - Uses two decoders, one for words and one for bits
  - **Word select** becomes **Row select**
  - **Bit select** becomes **Column select**
- **See next slide for example**
  - $A_3$  and  $A_2$  used for Row select
  - $A_1$  and  $A_0$  for Column select

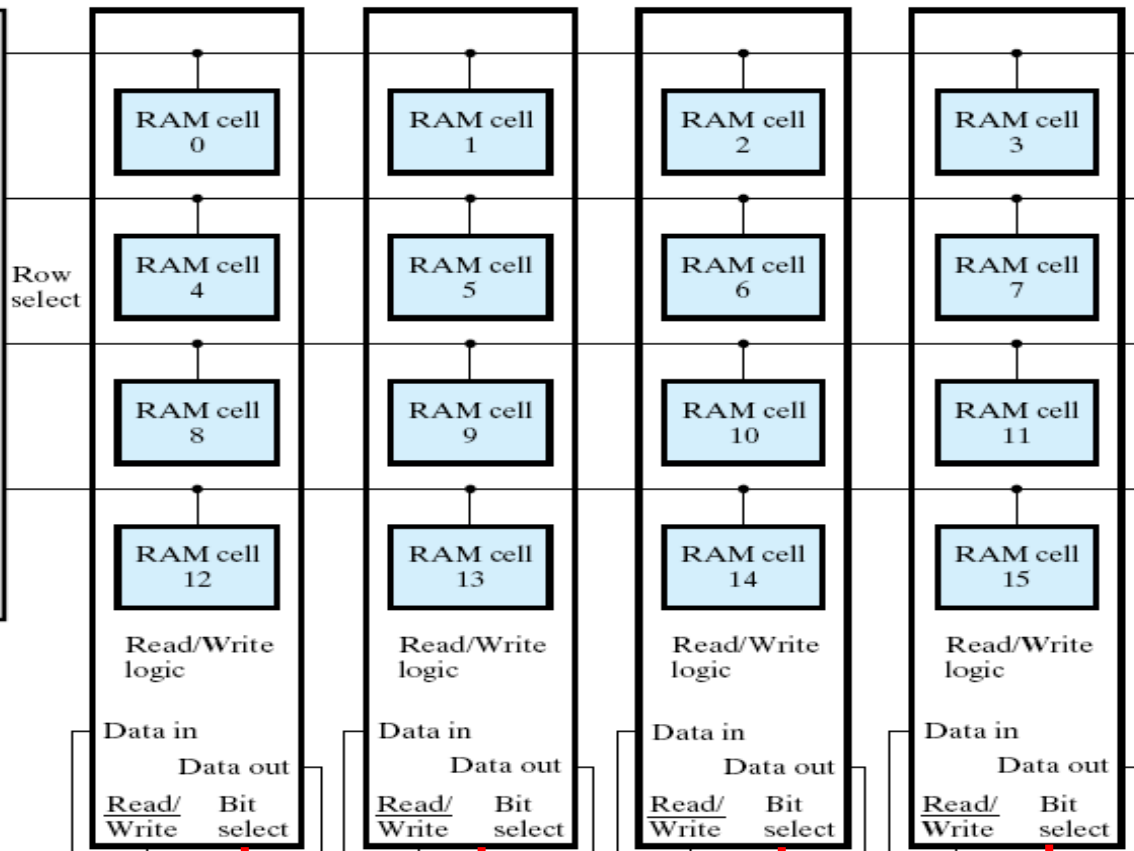
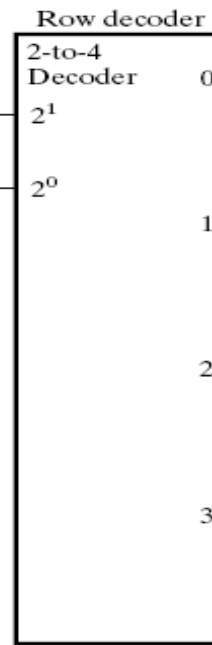
# Cell Arrays and Coincident Selection

## 2-dimensional



# RAM ICs with $> 1$ Bit/Word

- ❑ Word length can be quite high.
- ❑ To better balance the number of words and word length, use ICs with  $> 1$  bit/word
- ❑ See Figure 8-8 for example
  - 2 Data input bits
  - 2 Data output bits
  - Row select selects 4 rows
  - Column select selects 2 pairs of columns

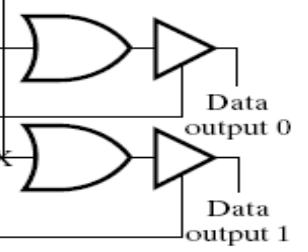
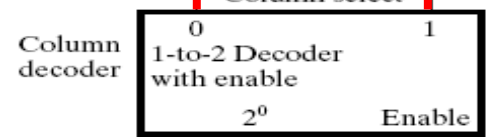


**8×2RAM**

Data input 0

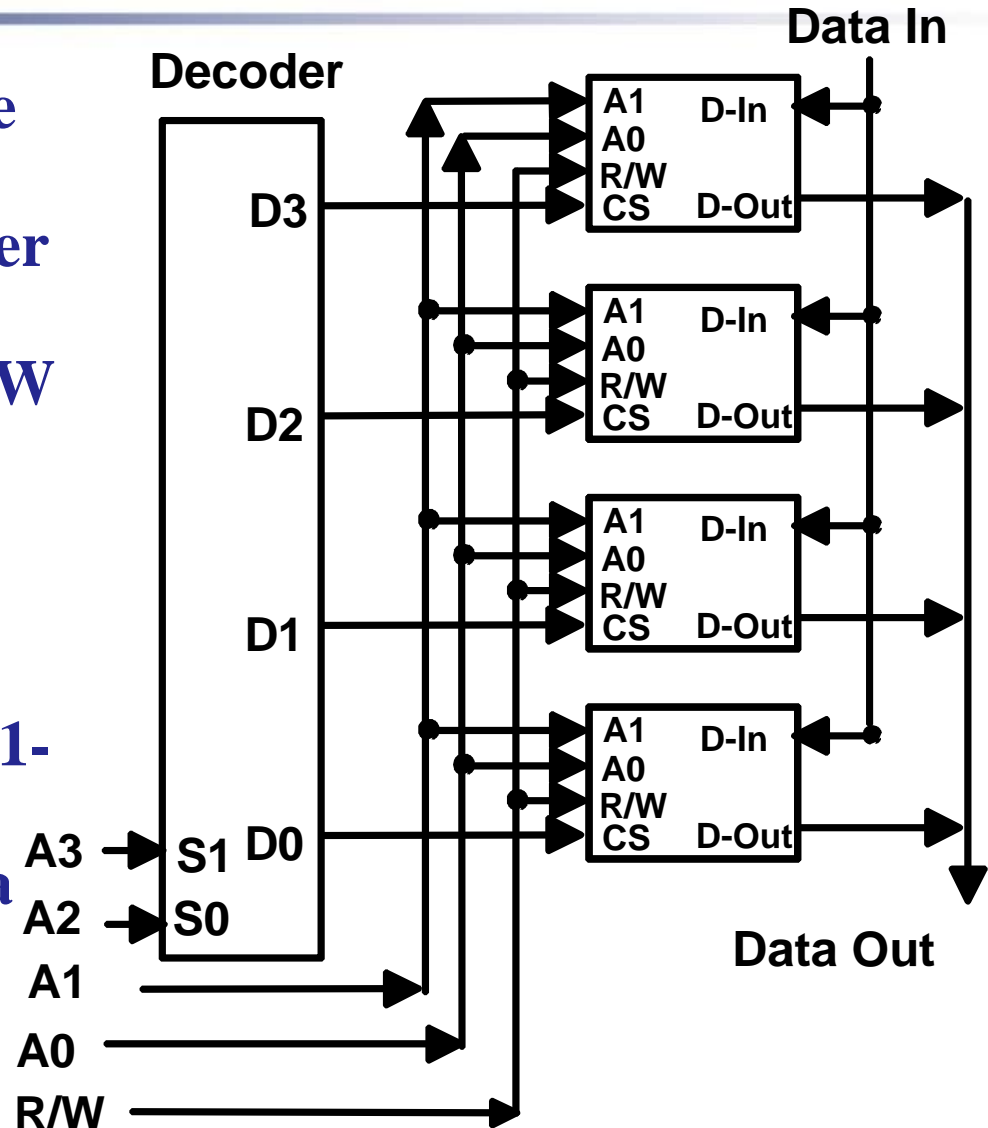
Data input 1

Read/Write



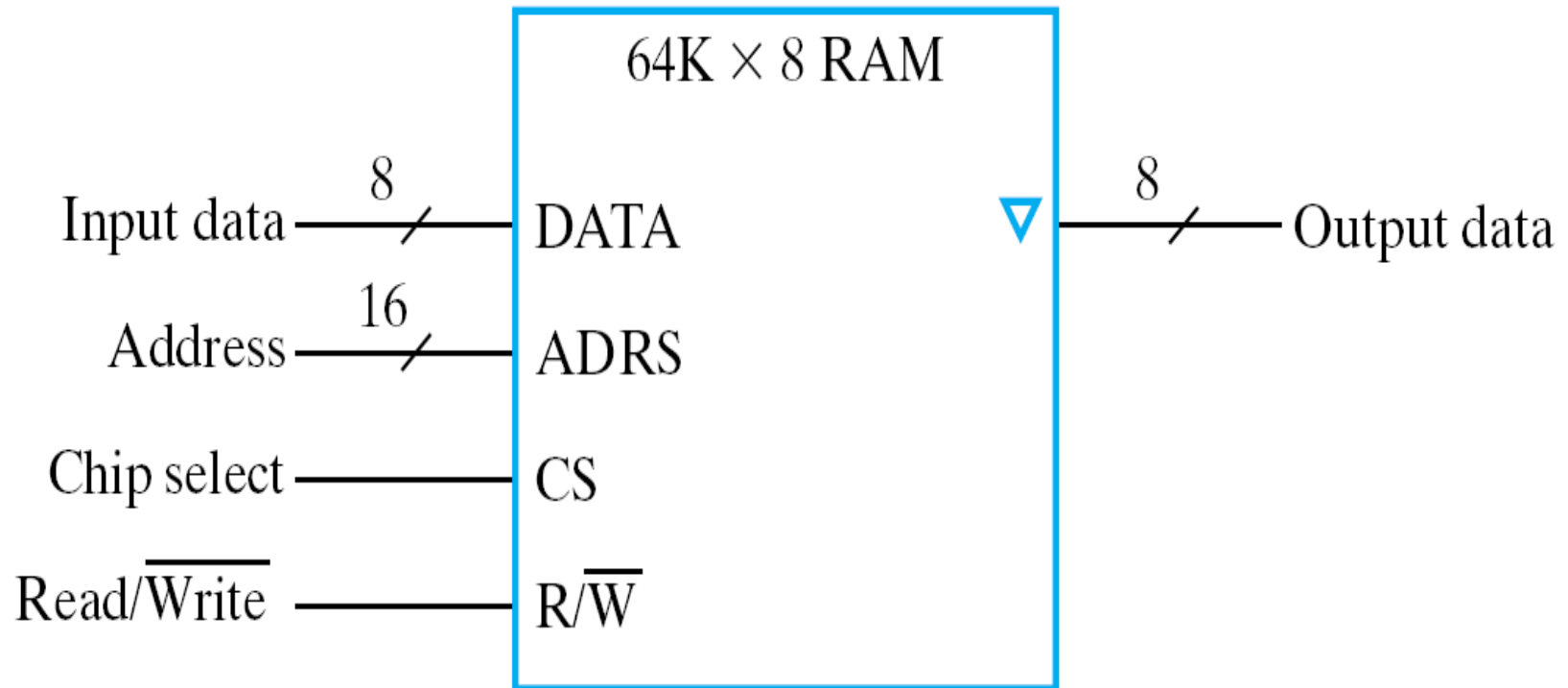
# Making **Larger** Memories: **Word extension**

- Using the CS lines, we can make larger memories from smaller ones by tying all address, data, and R/W lines in parallel, and using the decoded higher order address bits to control CS.
- Using the 4-Word by 1-Bit memory from before, we construct a 16-Word by 1-Bit memory.  $\Rightarrow$





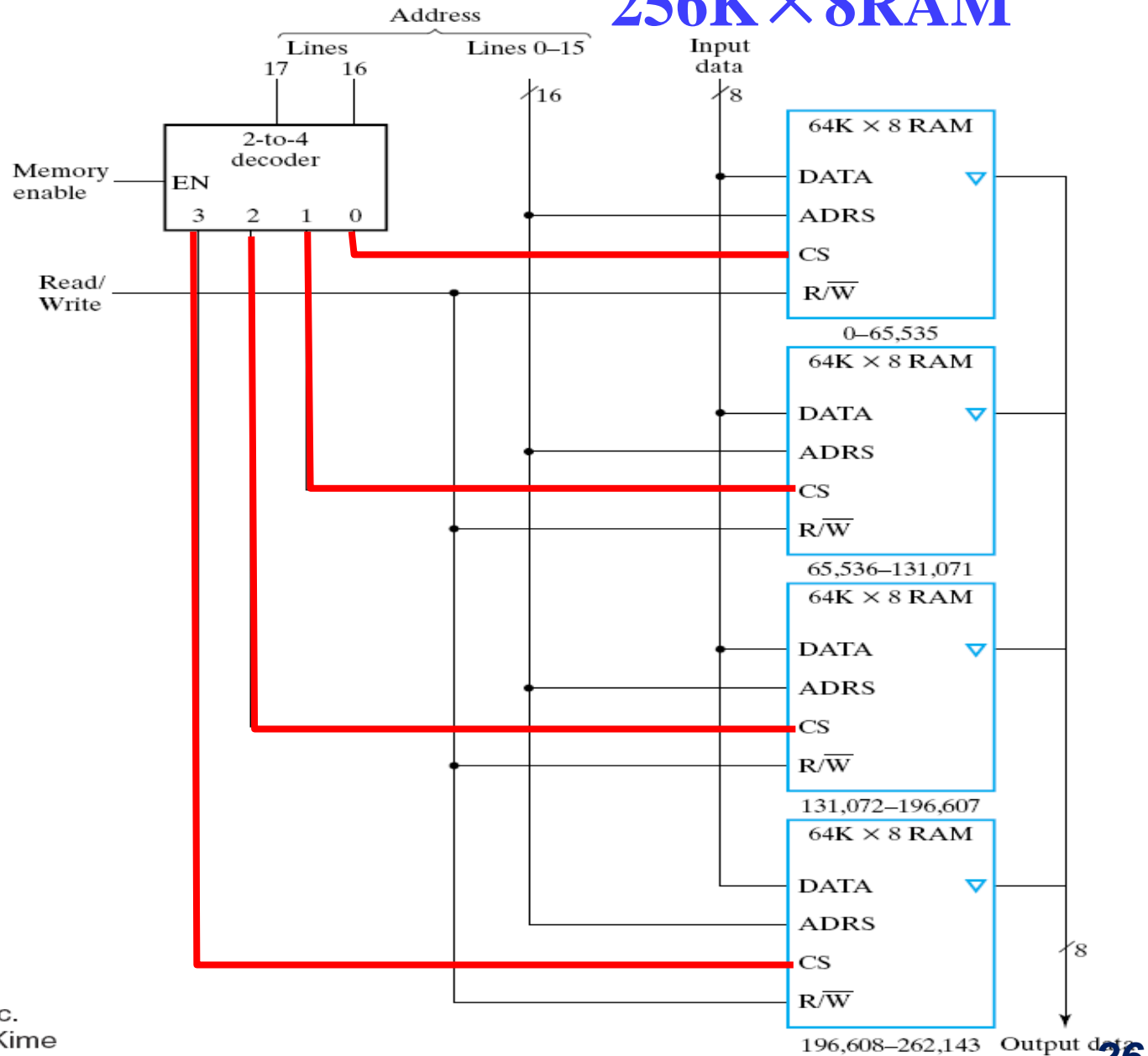
# 64K × 8 RAM



# Using 64K × 8 RAM construct

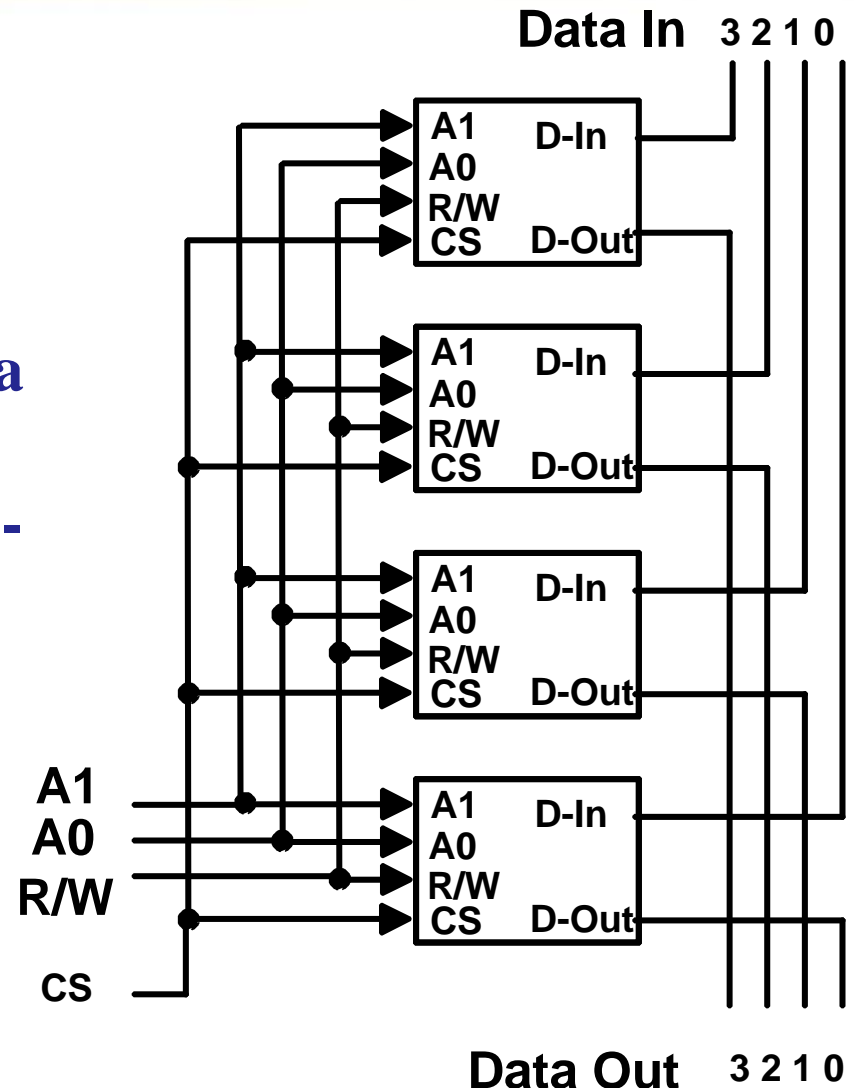


## 256K × 8 RAM



# Making **Wider** Memories: **Bit extension**

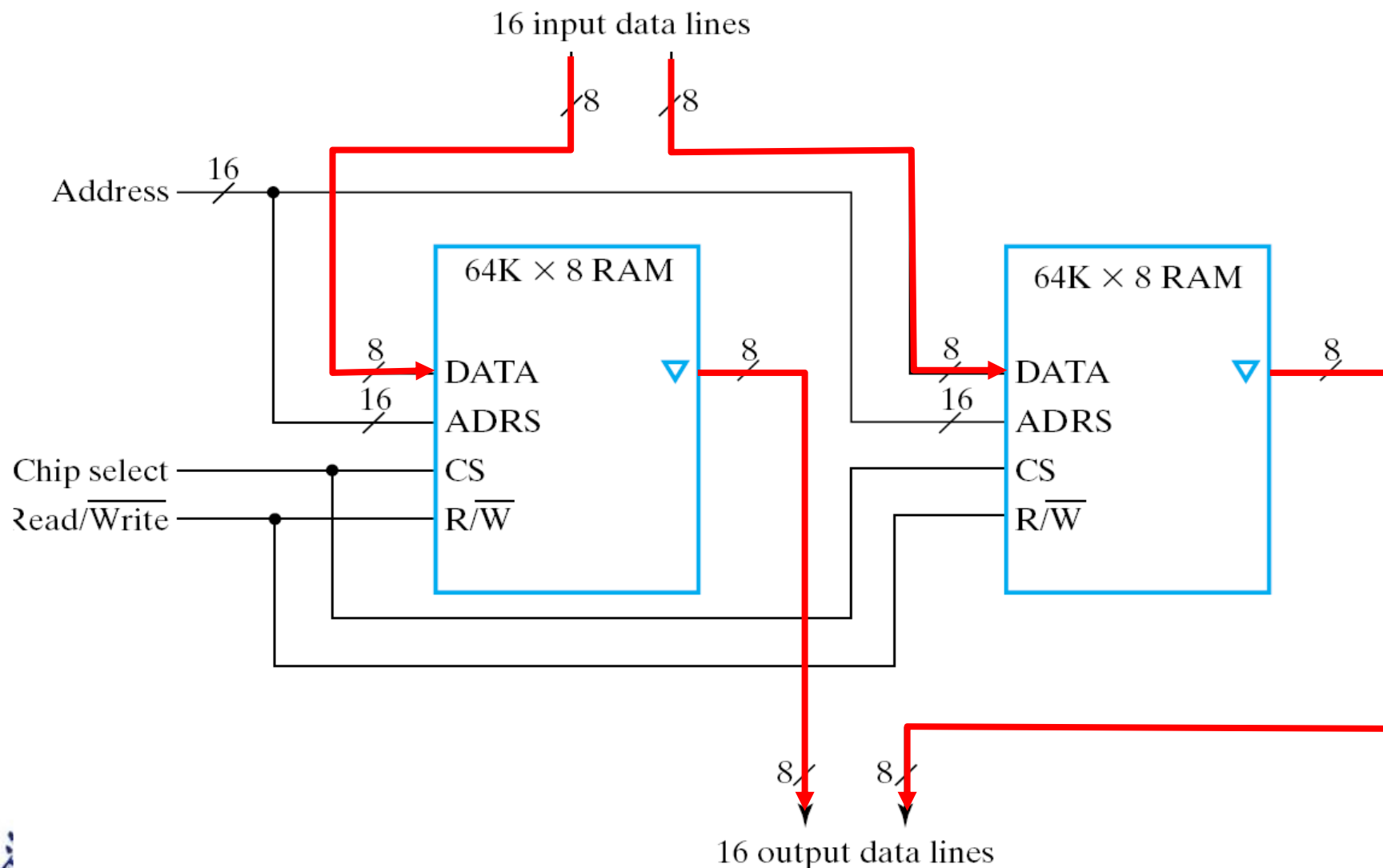
- To construct wider memories from narrow ones, we tie the address and control lines in parallel and keep the data lines separate.
- For example, to make a 4-word by 4-bit memory from 4, 4-word by 1-bit memories  $\Rightarrow$
- Note: Both 16x1 and 4x4 memories take 4-chips and hold 16 bits of data.



# Using $64\text{K} \times 8\text{RAM}$ construct

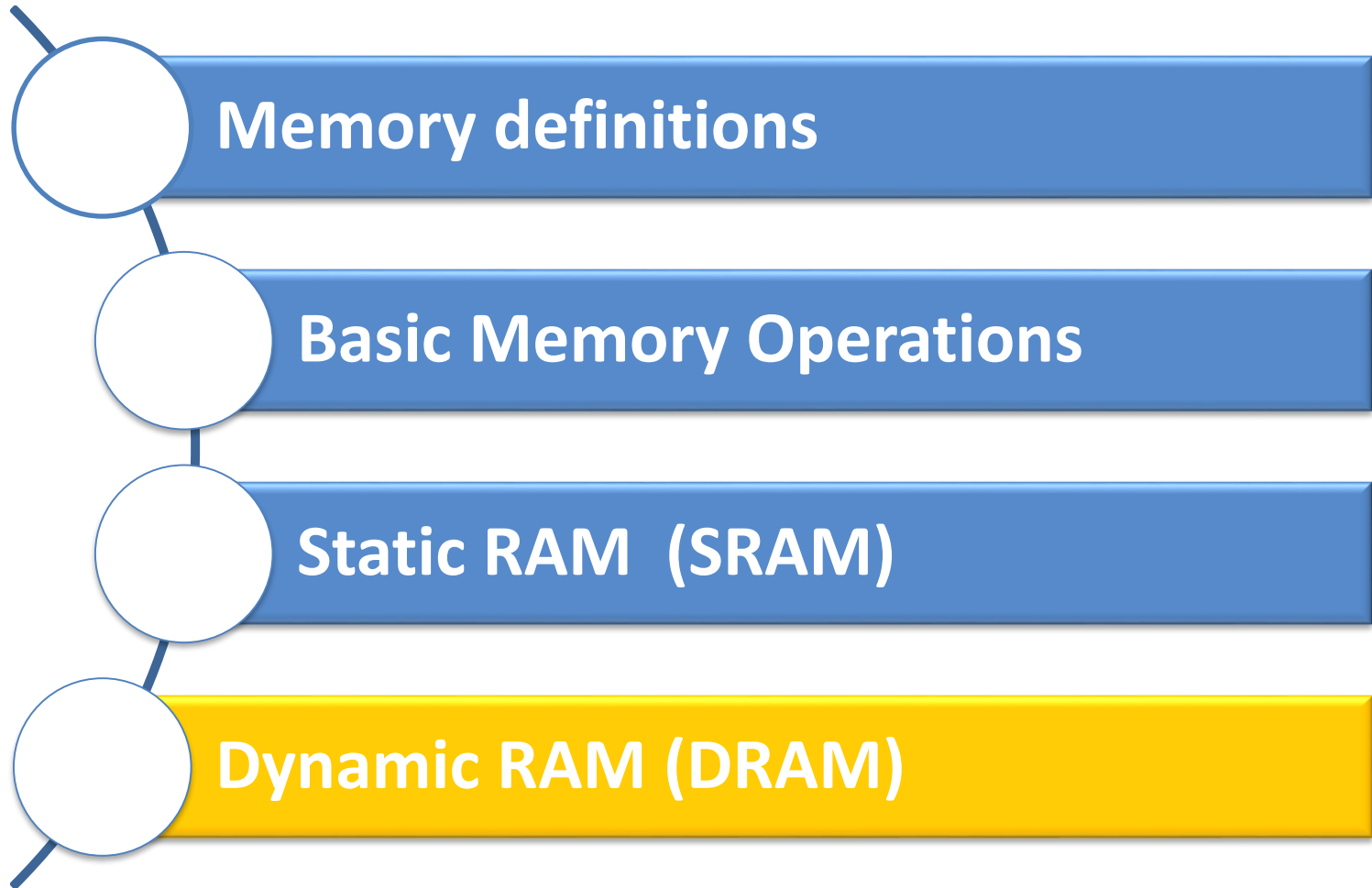


## $64\text{K} \times 16\text{RAM}$



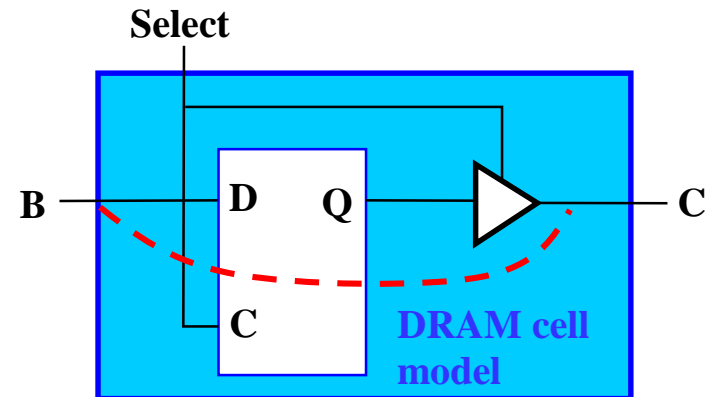
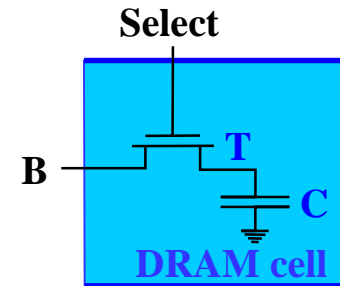


# Course Outline

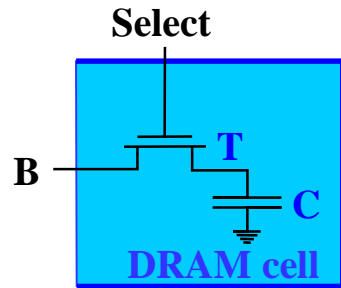


# Dynamic RAM (DRAM)

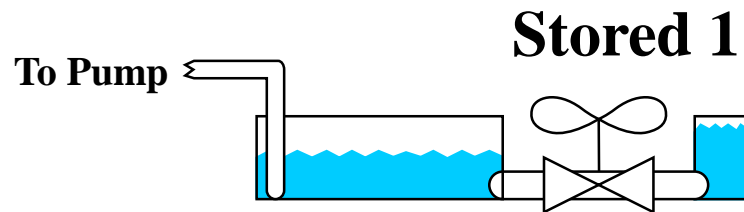
- ❑ **Basic Principle: Storage of information on capacitors.**
- ❑ **Charge and discharge of capacitor to change stored value**
- ❑ **Use of transistor as “switch” to:**
  - Store charge
  - Charge or discharge
- ❑ **Refresh**
  - 集中式刷新: 16~64ms
  - 分散式刷新: 15.6 $\mu$ s



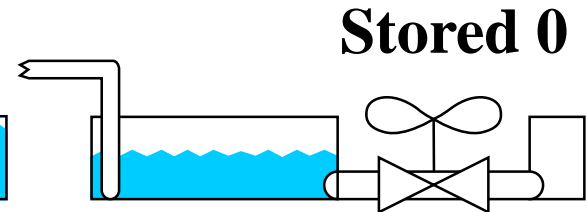
# Dynamic RAM Write & Read



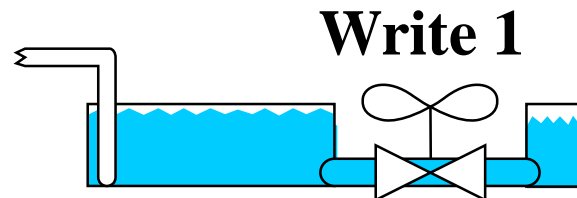
(a)



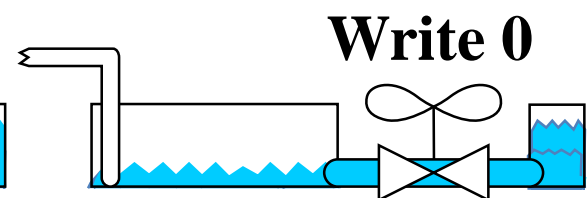
(b)



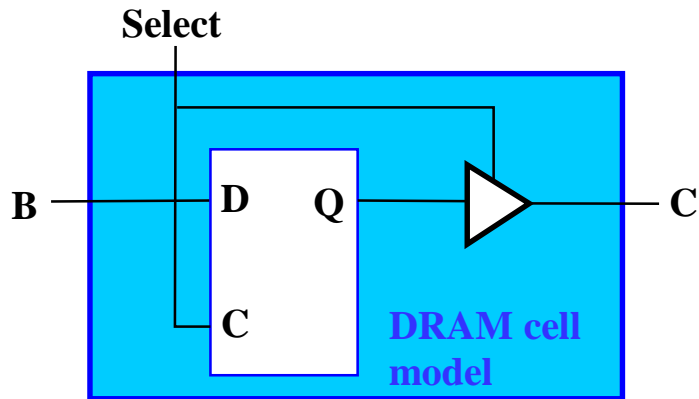
(c)



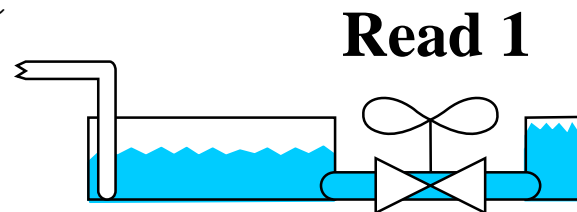
(d)



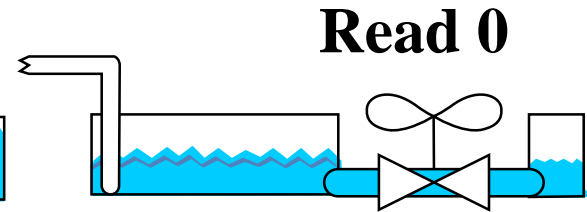
(e)



(h)



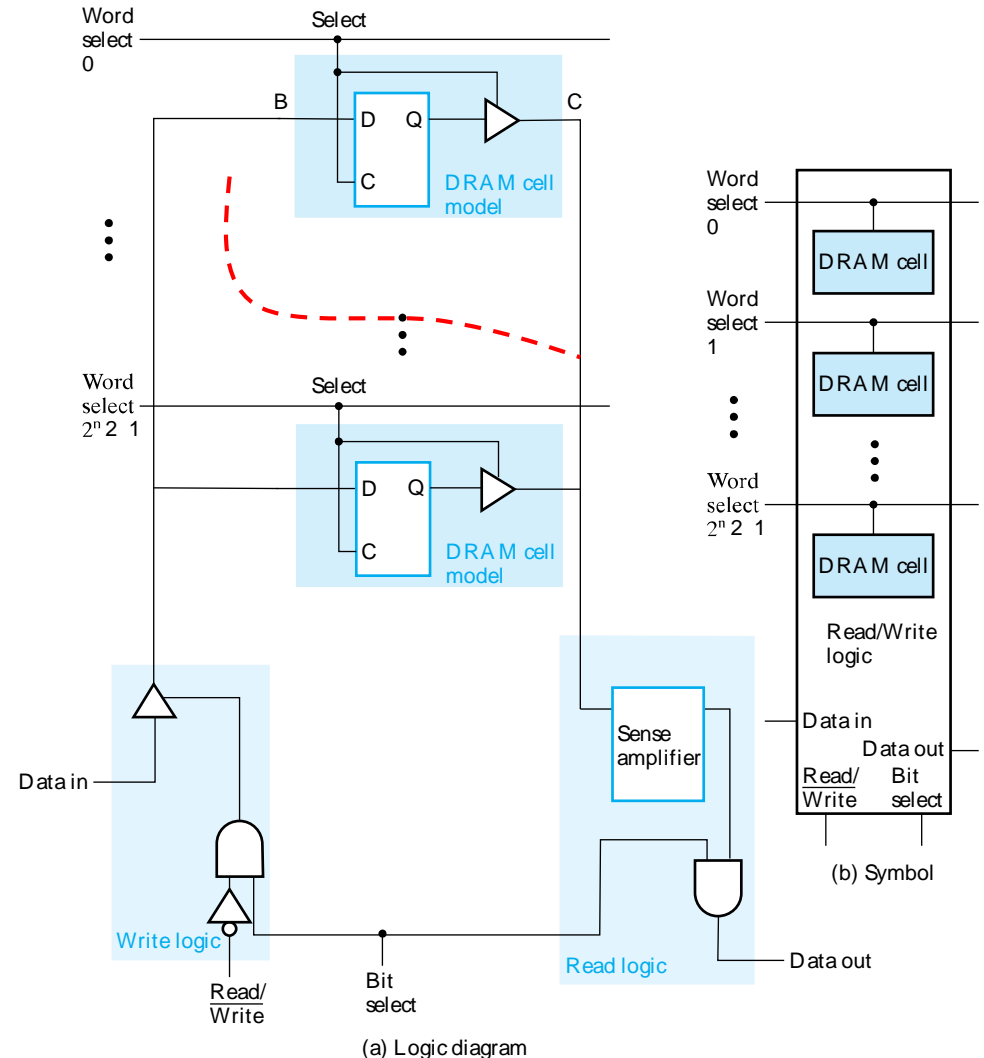
(f)



(g)

# Dynamic RAM - Bit Slice

- C is driven by 3-state drivers
- Sense amplifier is used to change the small voltage change on C into H or L
- In the electronics, B, C, and the sense amplifier output are connected to make destructive read into **non-destructive read**

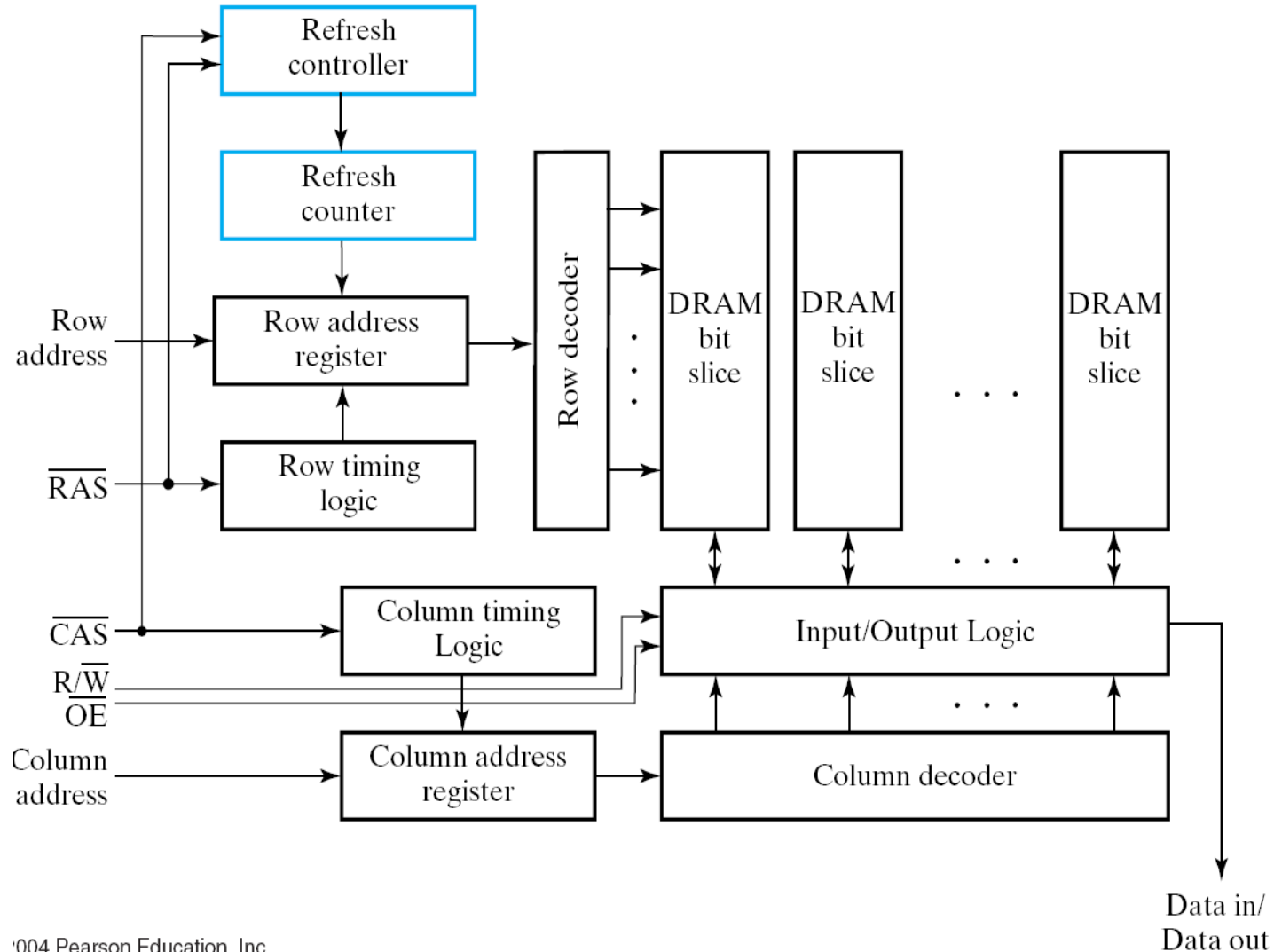




# Dynamic RAM - Block Diagram

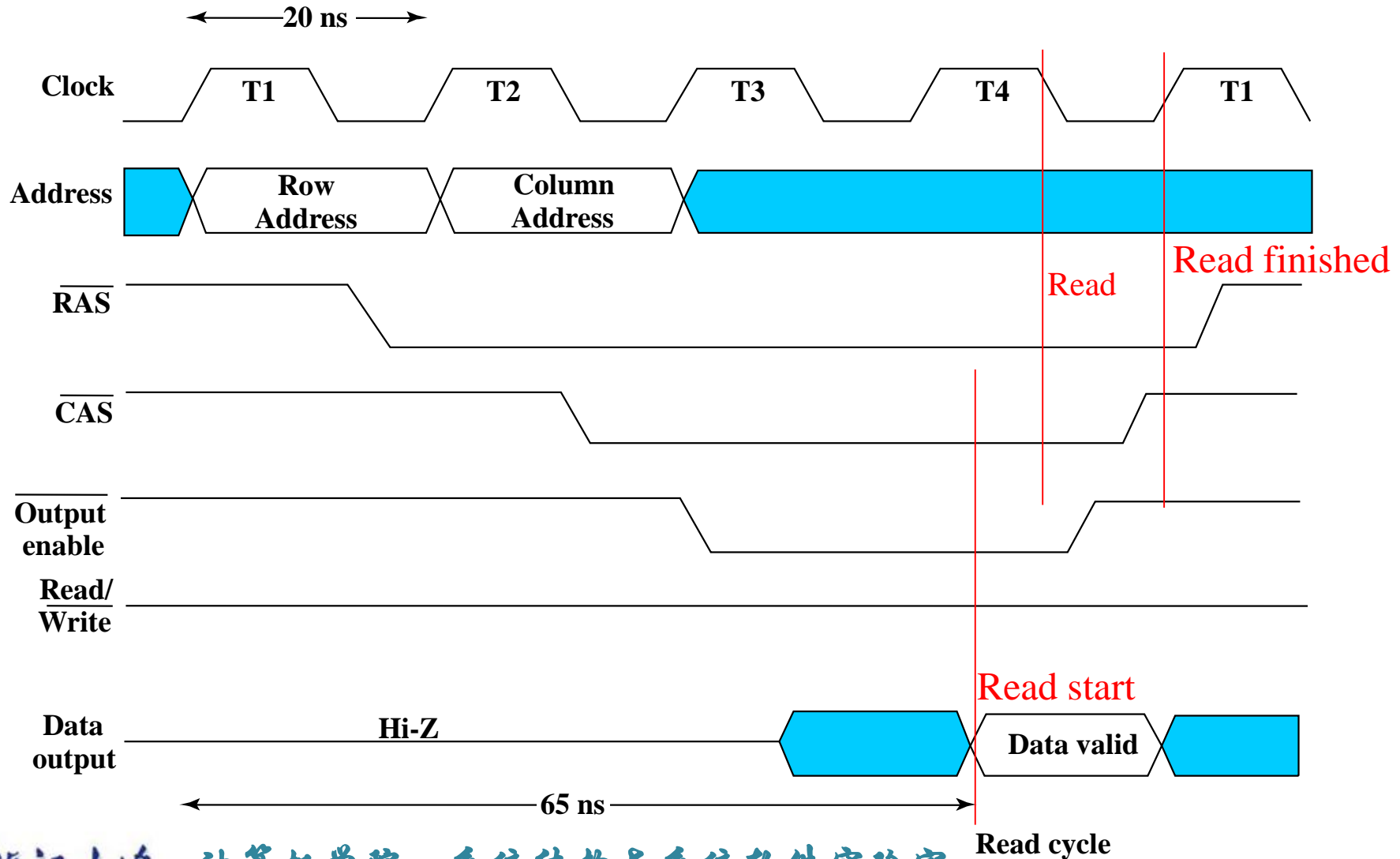
- ❑ Block Diagram – See Figure 8-14 in text
- ❑ Refresh Controller and Refresh Counter
- ❑ Read and Write Operations
  - Application of row address:  $\overline{RAS}$
  - Application of column address:  $\overline{CAS}$
  - Why is the address split?
  - Why is the row address applied first?

# Block Diagram with Refresh Controller

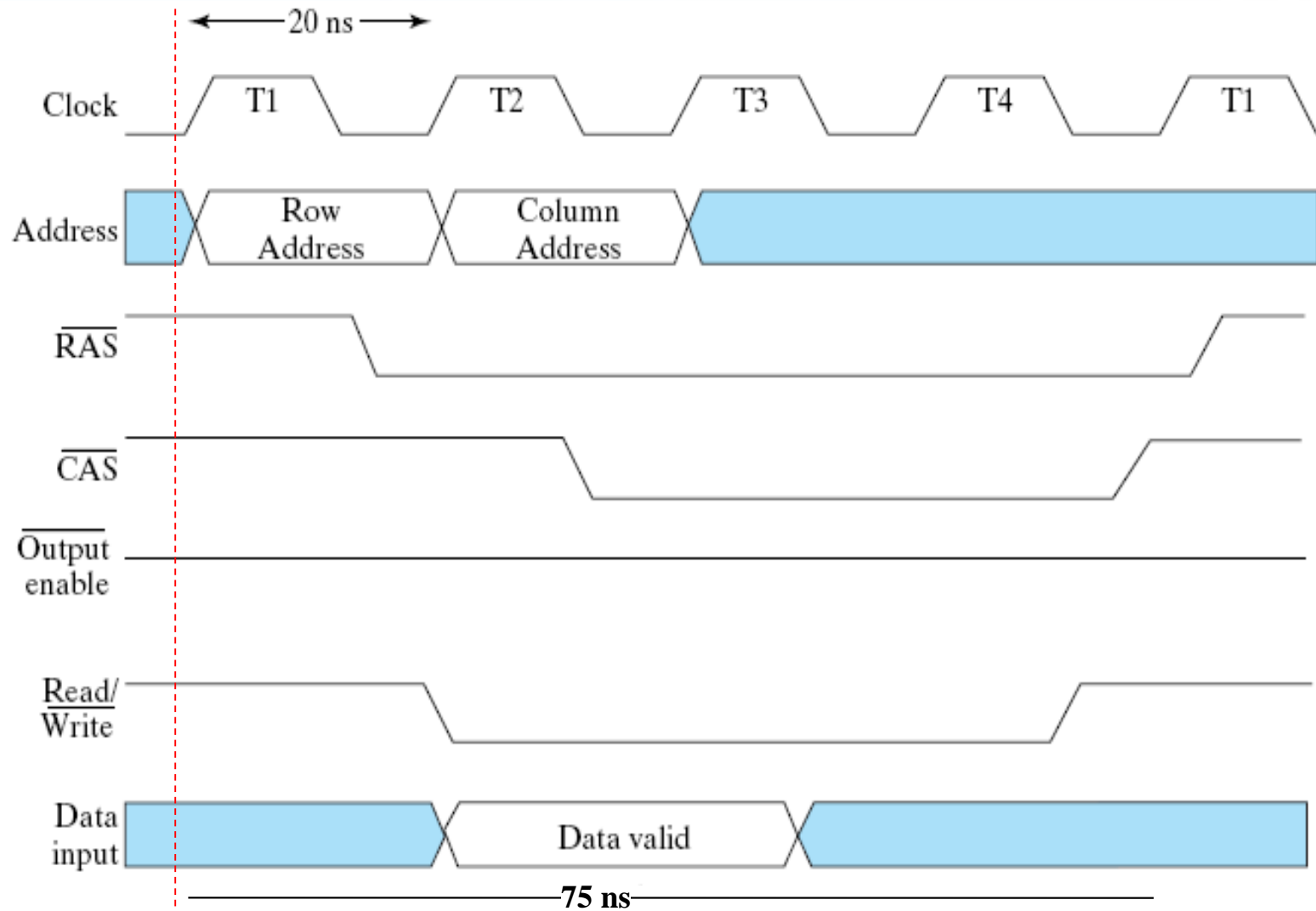


©2004 Pearson Education, Inc.

# Dynamic RAM Read Timing



# Dynamic RAM Write Timing



# DRAM Types

## □ Types to be discussed

- Synchronous DRAM (SDRAM)
- Double Data Rate SDRAM (DDR SDRAM)
- RAMBUS® DRAM (RDRAM)

## □ Justification for effectiveness of these types

- DRAM often used as a part of a memory hierarchy (See details in chapter 14)
- 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

# DRAM Types (continued)

## □ Justification for effectiveness of these types (continued)

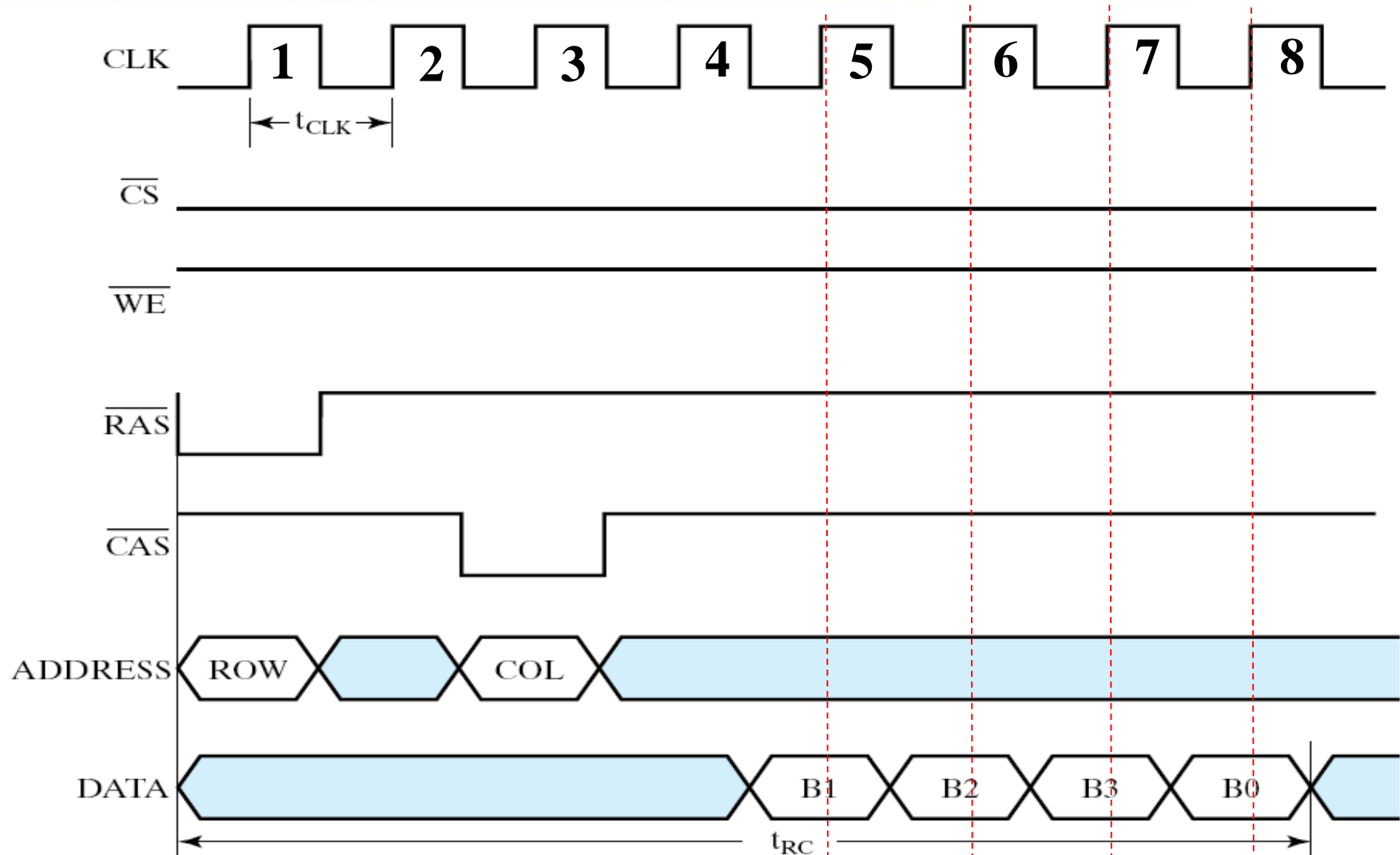
- These words are then transferred out over the memory data bus using a **series** of clocked transfers
- These transfers have a low delay, so several can be done in a short time
- The column address is captured and used by a synchronous counter within the DRAM to provide consecutive column addresses for the transfers

## □ ***burst read*** – the resulting multiple word read from consecutive addresses

# Synchronous DRAM

- ❑ Transfers to and from the DRAM are synchronize with a clock
- ❑ Synchronous registers appear on:
  - Address input
  - Data input
  - Data output
- ❑ Column address counter
  - for addressing internal data to be transferred on each clock cycle
  - beginning with the column address counts up to column address + burst size – 1
- ❑ 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:  $(4 + 8) \times 5 \text{ ns} = 60 \text{ ns}$
  - Memory Bandwidth:  $32 / (60 \times 10^{-9}) = 533 \text{ Mbytes/sec}$

# SDRAM *burst* time-- burst size=4





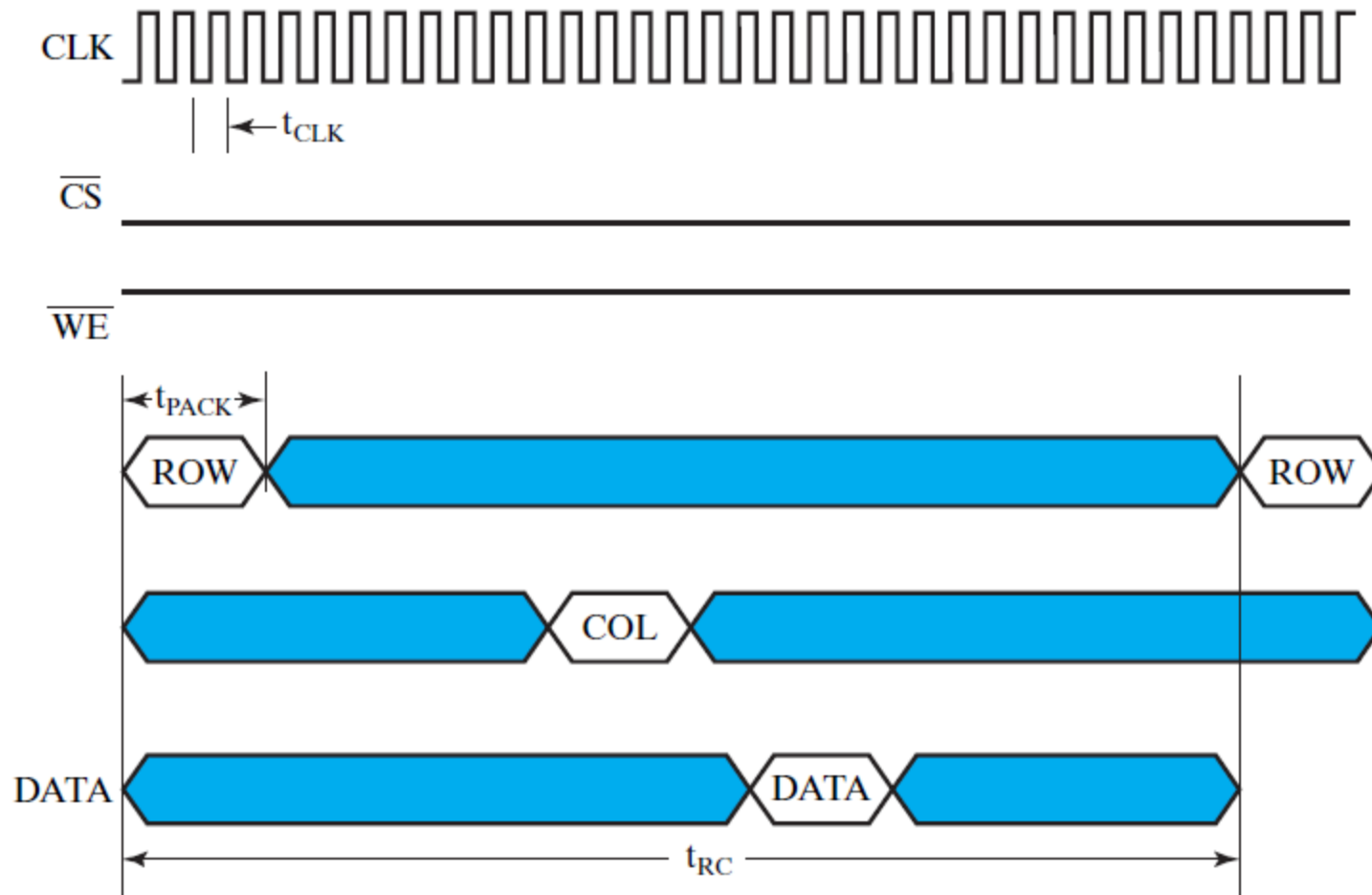
# Double Data Rate Synchronous DRAM

- ❑ 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:  $(2 \times 32)/(60 \times 10^{-9}) = 1.066$  Mbytes/sec

# RAMBUS DRAM (RDRAM)

- ❑ Uses a packet-based bus for interaction between the RDRAM ICs and the memory bus to the processor
- ❑ The bus consists of:
  - A 3-bit row address bus
  - A 5-bit column address bus
  - A 16 or 18-bit (for error correction) data bus
- ❑ The bus is synchronous and transfers on both edges of the clock
- ❑ Packets are 4-clock cycles long giving 8 transfers per packet representing:
  - A 12-bit row address packet
  - A 20-bit column address packet
  - A 128 or 144-bit data packet
- ❑ Multiple memory banks are used to permit concurrent memory accesses with different row addresses
- ❑ The electronic design is sophisticated permitting very fast clock speeds

# Timng of RDRAM



# Arrays of DRAM Integrated Circuits

- ❑ Similar to arrays of SRAM ICs, but there are differences typically handled by an IC called a *DRAM controller*:
  - Separation of the address into row address and column address and timing their application
  - Providing  $\overline{\text{RAS}}$  and  $\overline{\text{CAS}}$  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)



# 第七章作业

## Ch 7

page431-432:

7-1, 7-4, 7-5, 7-8

*From Registers To Registers, On this matter!*

Thank you!



# Quiz

---

1. **A serial 2's complementer is to be designed. A binary integer of arbitrary length is presented to the serial 2's complementer, least significant bit first, on input X. When a given bit is presented on input X, the corresponding output bit is to appear during the same clock cycle on output Z. To indicate that a sequence is complete and that the circuit is to be initialized to receive another sequence, input Y becomes 1 for one clock cycle. Otherwise, Y is 0.**
  - a) **Find the state diagram for the serial 2's complementer.**
  - b) **Find the state table for the serial 2's complementer.**
  - c) **Find the output equations and next-state equations.**
  - d) **Draw the circuit diagram.**