Memory – Part B

Types of memories, memory management

Gloria Beraldo (gloria.beraldo@unipd.it)
Department of Information Engineering, University of Padova

Topics:

- Types of memories
- RAM, SRAM, DRAM e SDRAM
- Array of memories SRAM
- Addressing, timing, refresh of the memories DRAM

Book Reference:

- Chapter 7

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

SRAM: Static Random Access Memory

- SR Latch as model of a cell SRAM

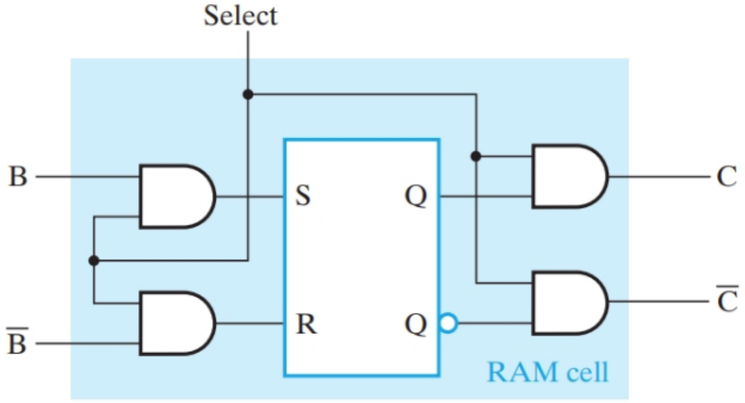
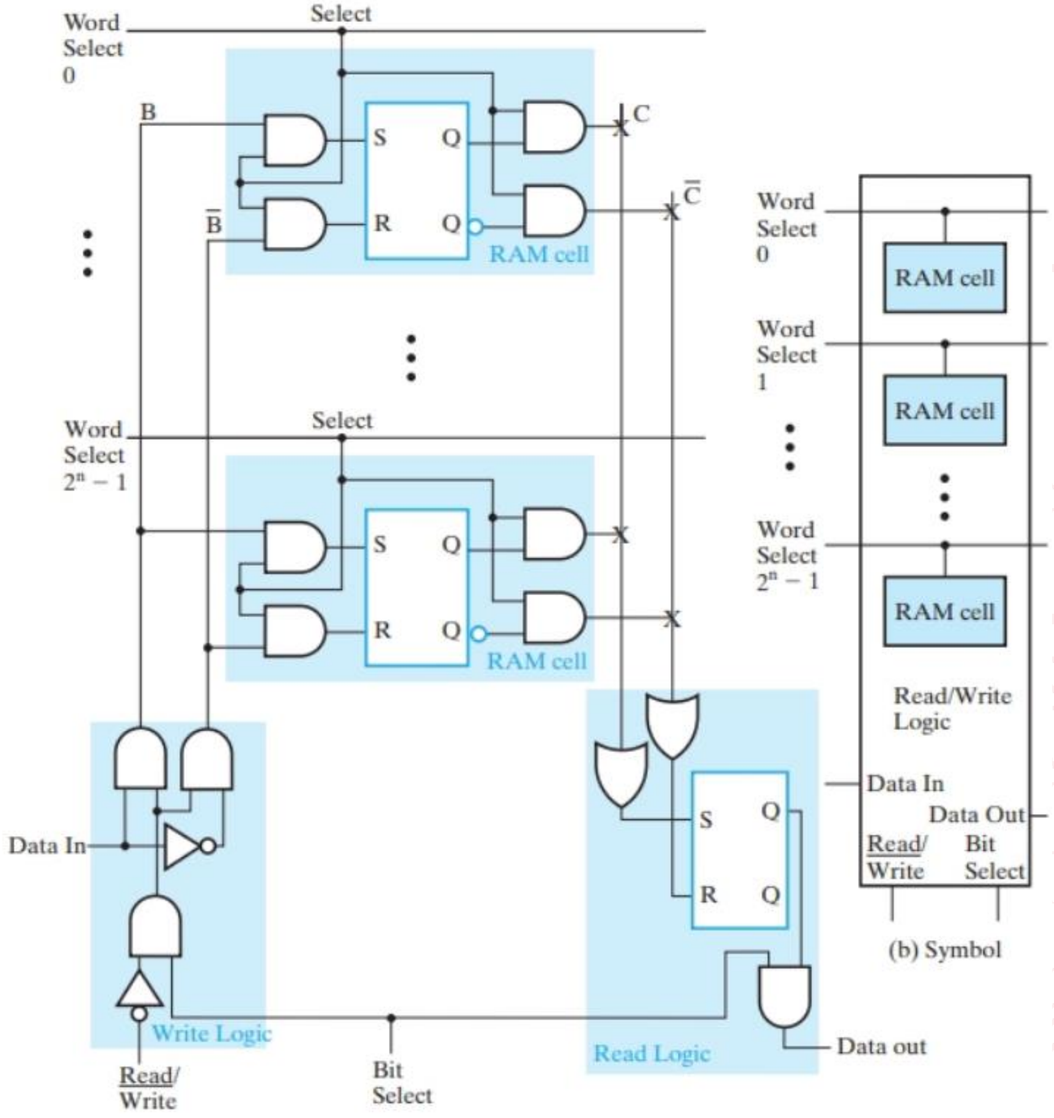


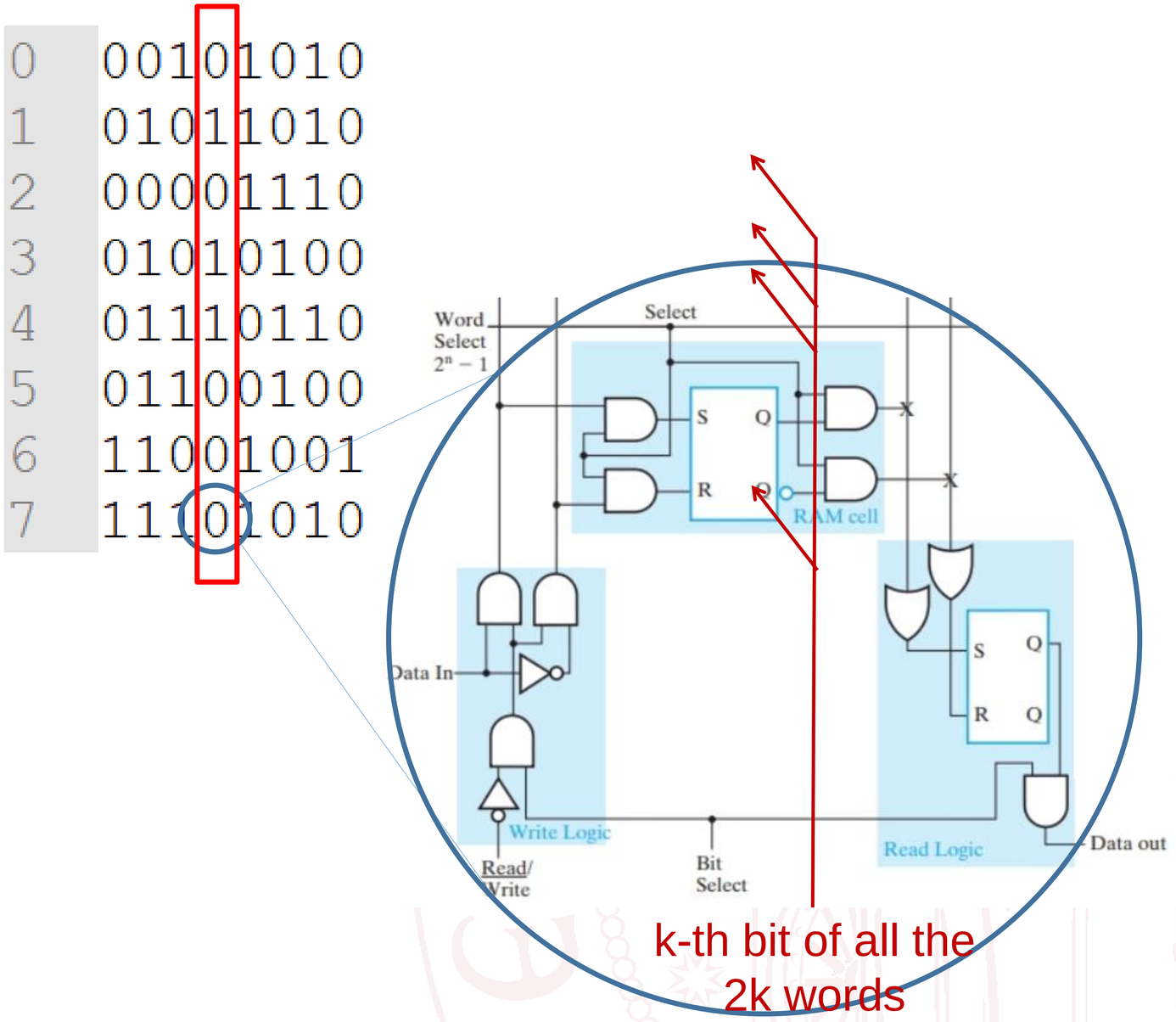
FIGURE 7-4
Static RAM Cell

- If $Select = 1$, the stored content is determined by the values $B, \overline{B}$
- If $Select = 0$, both $C, \overline{C} = 0$
- $Select$ enables/disables the access to the cell
- RAM bit slice → contains all of the circuitry associated with a single bit position of a set of RAM words



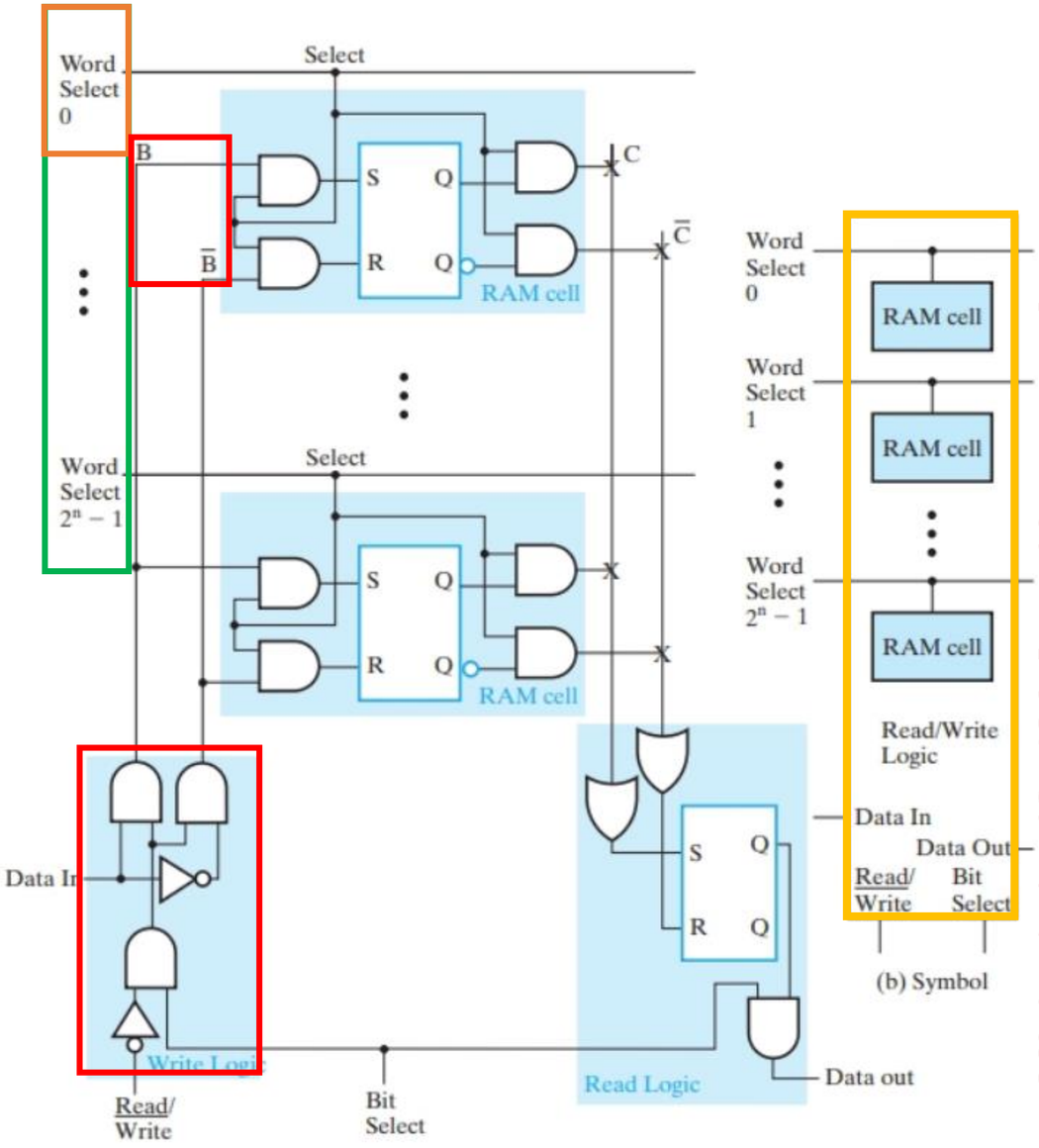
SRAM Bit Slice

- Every bit of our memory needs all this logic
- Can we design the circuit to be cheaper?
- In memory we can access only ONE location (=row) at a time
- We can decide to share the select, read, write logic of a single bit with all the bits of the same column!
- We call this component BIT SLICE



SRAM

- The loading of a cell latch is controlled by a signal Word Select
  - *You can require the access to the k bit of the word at the address* $0\ or\ 1\ or\ 2\ or\ \ldots\ 2^n - 1$
- For «*Word Select 0*» = *0* The k bit of the word 0 remain unchanged
- For «*Word Select 0*» = *1* the stored content is determined by the values on $B, \overline{B}$ that are determined in turn by the Write Logic
- This applies to one bit of the word
- Each RAM bit slice is interconnected with other to control all the bits of a word



16-Word by 1-Bit RAM Chip

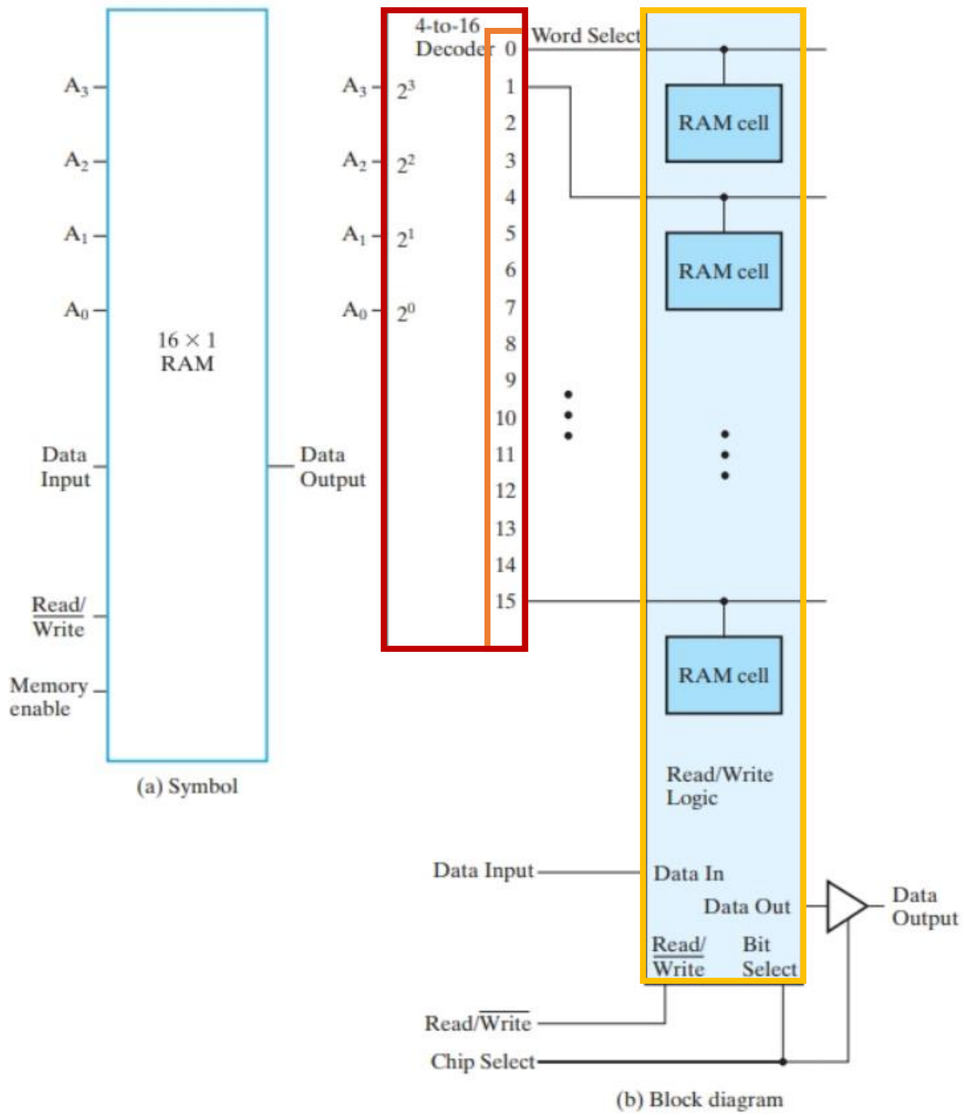


FIGURE 7-6
16-Word by 1-Bit RAM Chip

- For simplicity, we consider a SRAM chip with 16 one-bit words
- The internal structure of the RAM chip consists of a RAM bit slice having 16 RAM cells
- To select a word, we need 16 addresses
- To select 16 addresses, we need a 4-to-16 line decoder
- In this case, the address is composed of 4 bits ($16 = 2^4$)

A_3	A_2	A_1	A_0

16x1 SRAM chip with 4x4 RAM Array

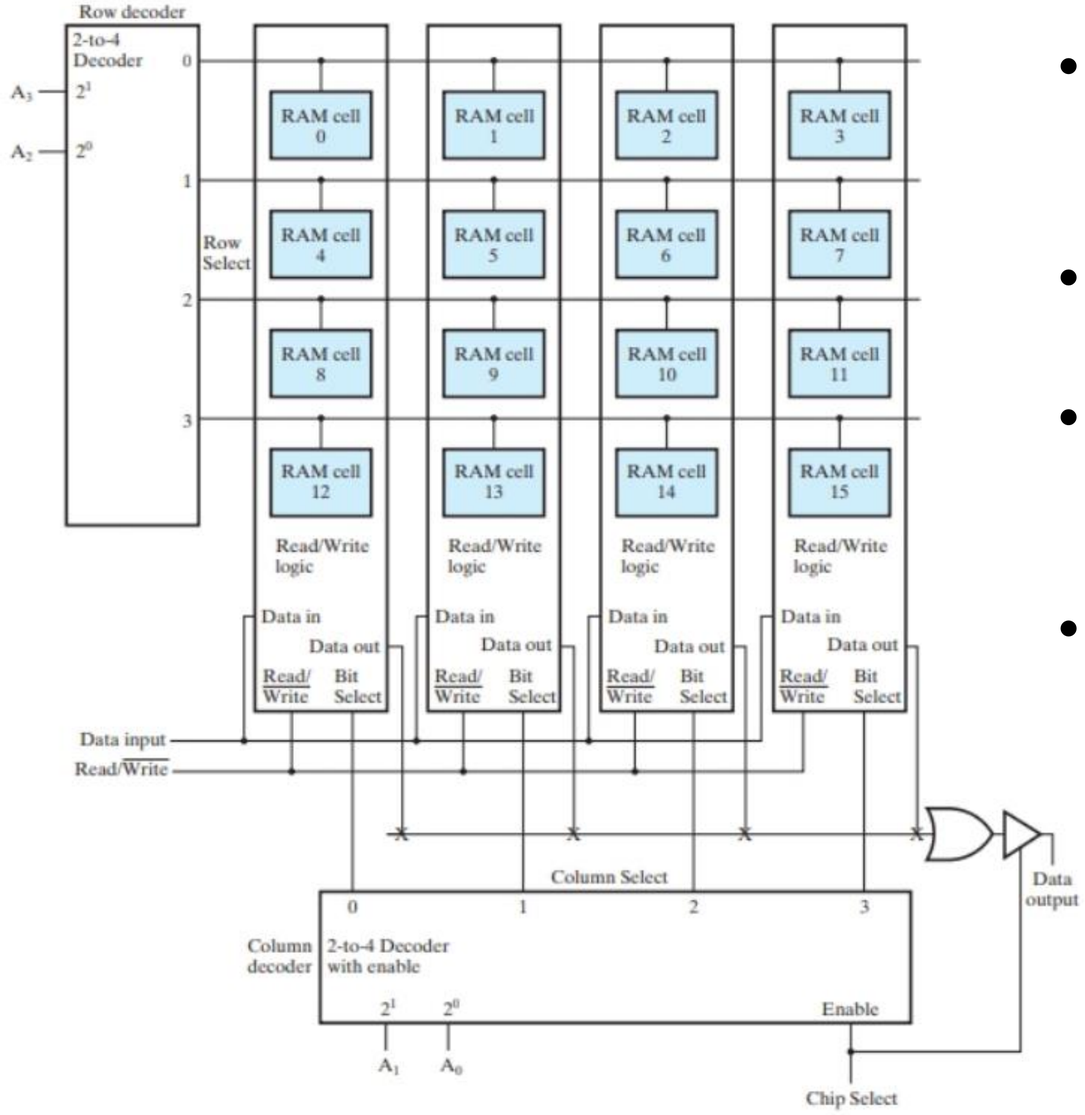
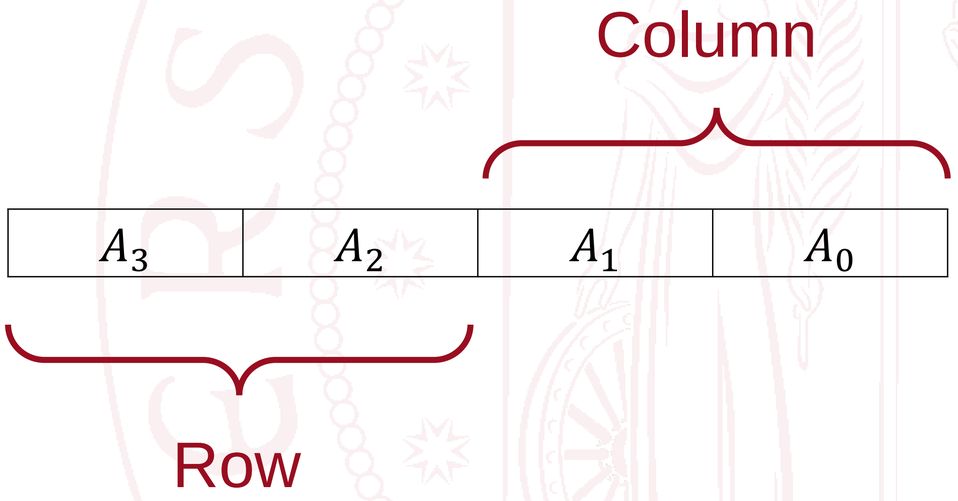


FIGURE 7-7
Diagram of a 16 × 1 RAM Using a 4 × 4 RAM Cell Array

- Different structure (to optimize space and components)
- 2D matrix of 16 RAM cells
- Selection through a 2-to-4-line row decoder to select the row and the column
- The address is composed of 4 bits



8x2 SRAM chip using 4x4 RAM Cells Array

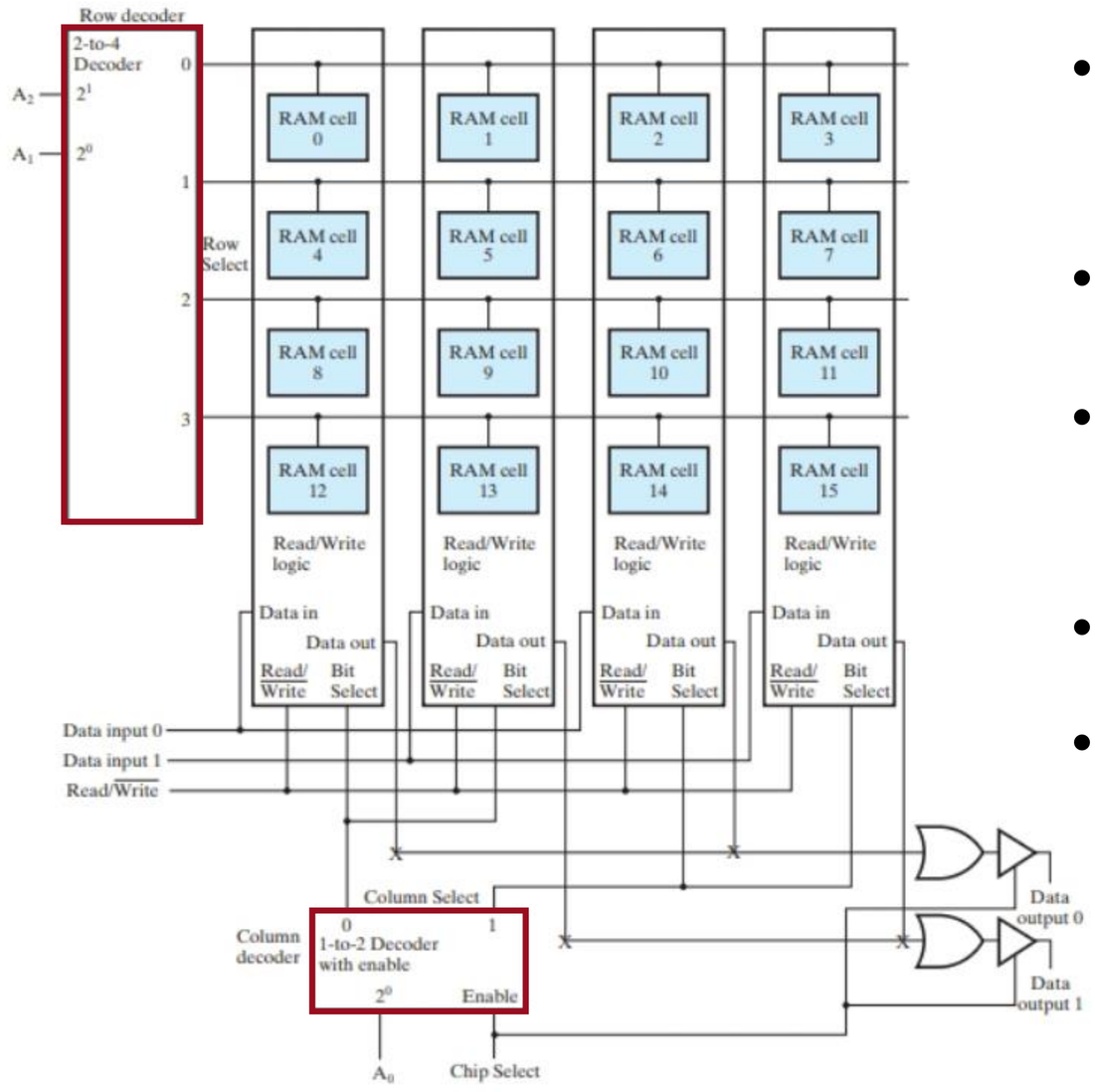
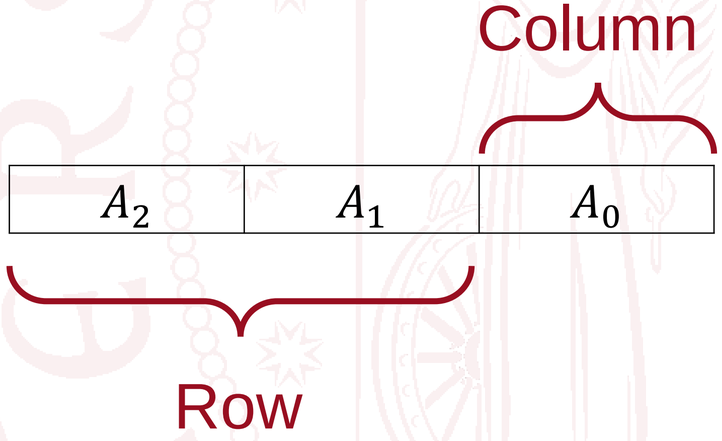


FIGURE 7-8
Block Diagram of an 8 × 2 RAM Using a 4 × 4 RAM Cell Array

- Chip SRAM 8 x 2 (8 words of two bits each)
- Same approach of the previous example
- The number of bit in the address to identify a word: 3 ($8 = 2^3$)
- 2 bits in the address for the row decoder
- 1 bit in the address for the column decoder



8x2 SRAM chip using 4x4 RAM Cells Array

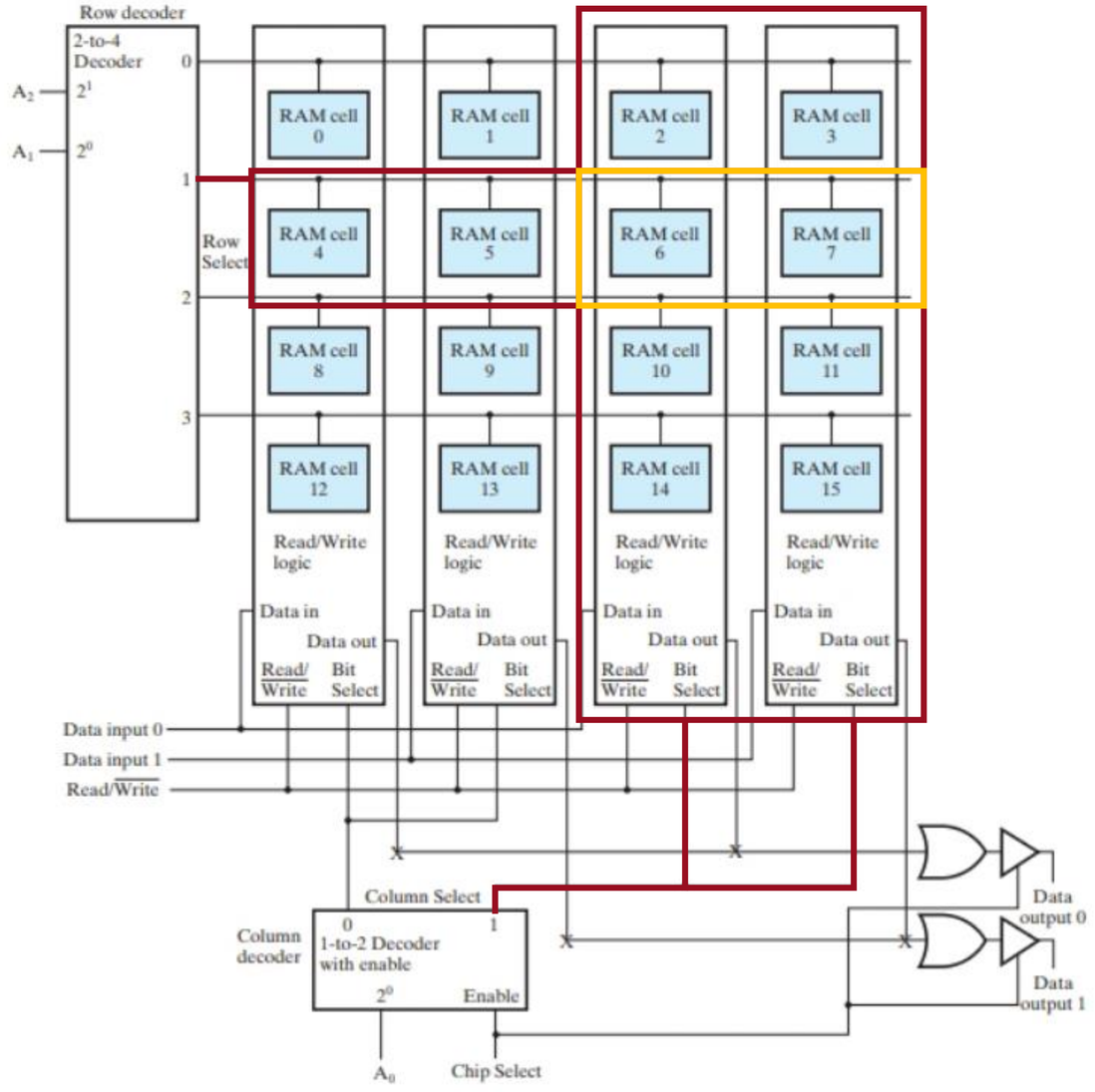


FIGURE 7-8
Block Diagram of an 8 × 2 RAM Using a 4 × 4 RAM Cell Array

Example: A = 011_2

A_2	A_1	A_0
0	1	1

- Row decoder:
  - Selected cells: 4, 5, 6, 7
- Column decoder:
  - Selected cells: 2, 6, 10, 14, 3, 7, 11, 15
- Selected word: cells 6 and 7 (2 bits)

Example SRAM chip

- We want to design a SRAM 32K x 8
  - Namely: with 32K words of 8 bits

  (Cypress SRAM CT62256.pdf)

- This SRAM chip contains a total of 256K bits (32K x 8)
- The minimum number of addresses from 32K words is k such that: $2^k \geq words$ is true
- $2^k \geq 32 \cdot 1024 \Rightarrow k \geq \log_2 32768 \Rightarrow k \geq 15$
- To make the number of rows and columns in the array equal:
  - $nrows\ ncolumns = \sqrt{256K} = \sqrt{(256 \cdot 1024)} = 512 = 2^9$

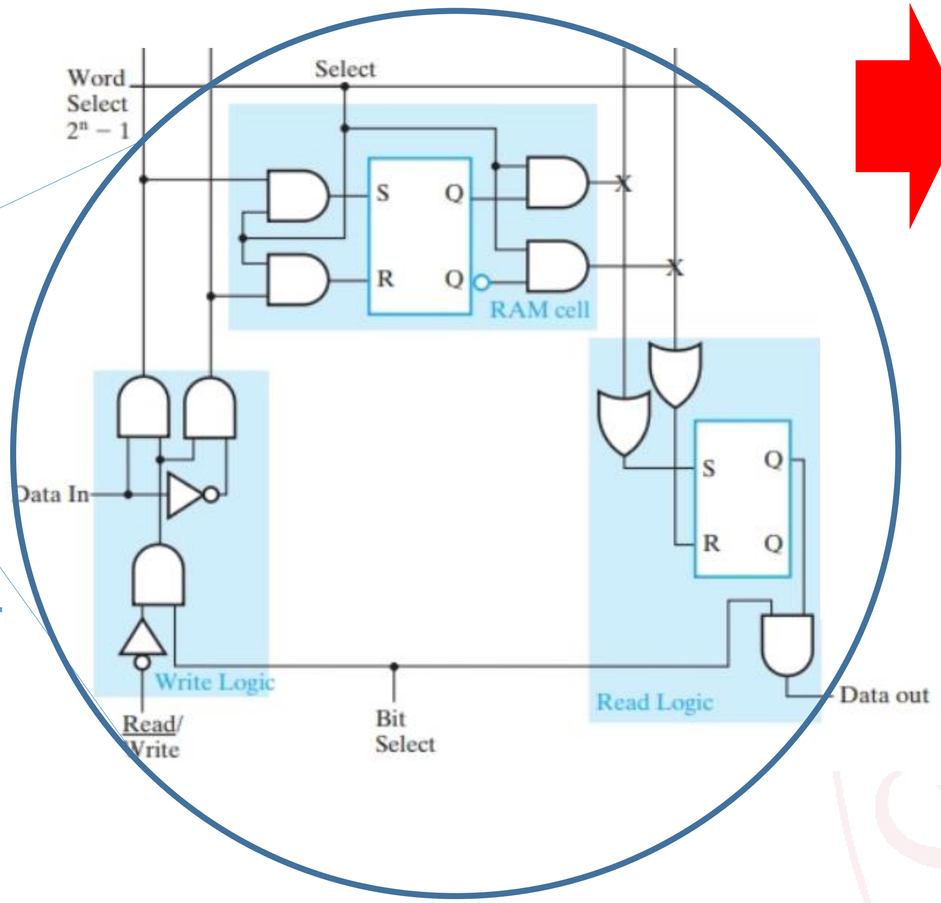
since we address the words (groups of 8 bits) and not the individual bits, we just need 9-3=6 bits

A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
Row									Column					

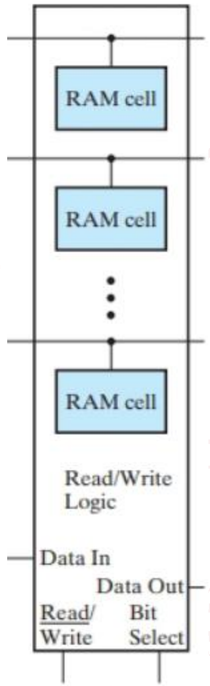
Recap

0	00101010
1	01011010
2	00001110
3	01010100
4	01110110
5	01100100
6	11001001
7	11101010

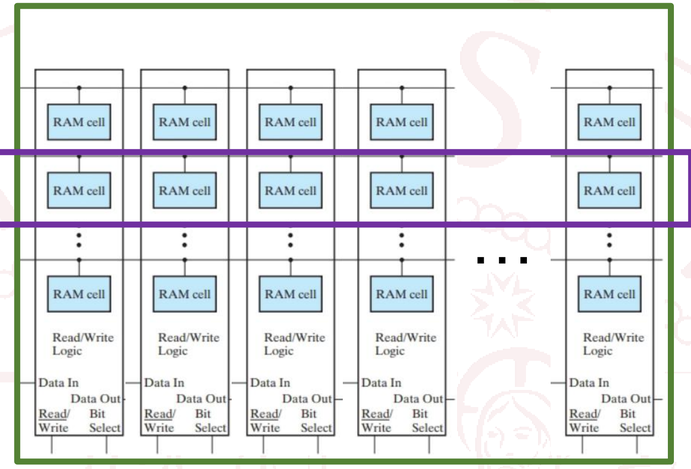
ONE BIT LOGIC



BIT COLUMN LOGIC (RAM slices)



LOTS OF RAM SLICES = ALL RAM (organized as an array)



The bits of a WORD (memory location) are stored on multiple RAM slices

DRAM: Dynamic Random Access Memory

- Low cost and high storage capacity
- Very common in the memory in our current computers
- Similar implementation to SRAM, but with **significant differences**:
  - The electronic design is considerably more challenging
  - The storage of information is inherently only temporary (dynamic in the name)
  - The information must be periodically refreshed

DRAM Cell

- From the electronic point of view, a DRAM consists of a transitor (T) and a capacitor (C)
- If sufficient charge is stored on the capacitor, the stored value is 1, otherwise is 0
- The transistor acts much like a switch:
  - When the switch is open, the charge in C remains fixed (the information is stored)
  - When the switch is closed, the charge can flow into and out of the C from the external Bit Line (B)
- These operations refer to the cell readings/writings

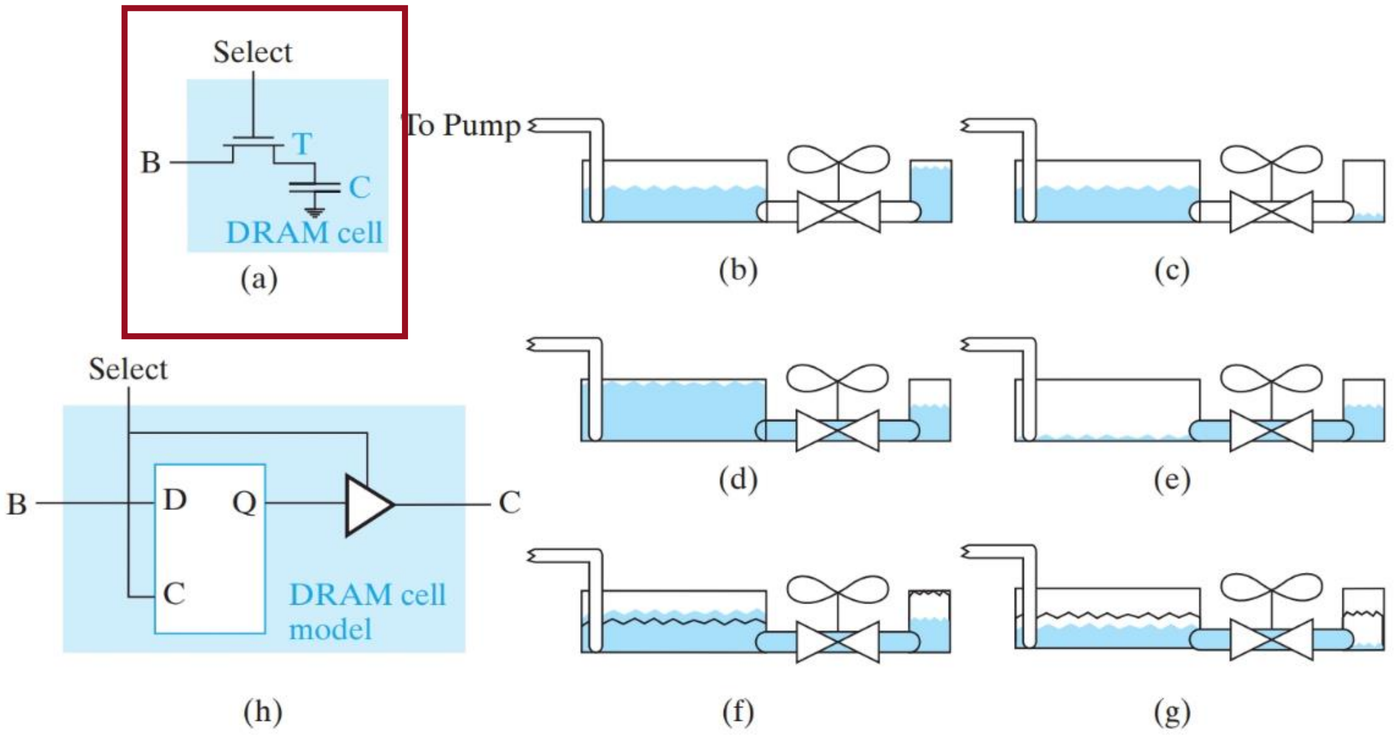


FIGURE 7-12
Dynamic RAM cell, hydraulic analogy of cell operation, and cell model

DRAM Cell: hydraulic analogy

- With closed valve (T opened):
  - C = 1
  - C = 0

storage

- With opened valve (T closed):
  - B = 0 →1; C = 0 → 1
  - B = 1 → 0; C = 1 → 0

write

- With opened valve (T closed):
  - C = 1; B = + dC = 1
  - C = 0; B = - dC = 0

read

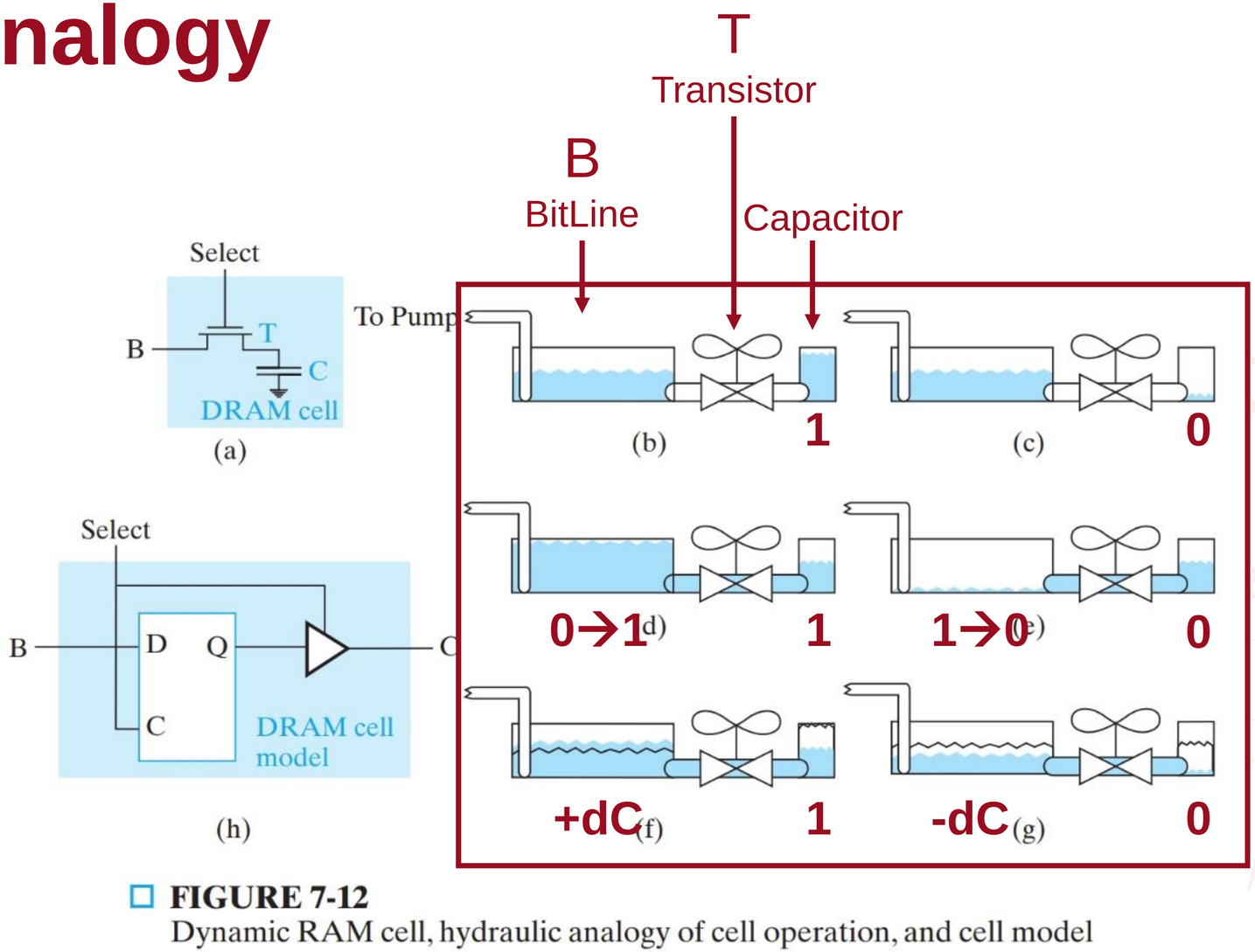
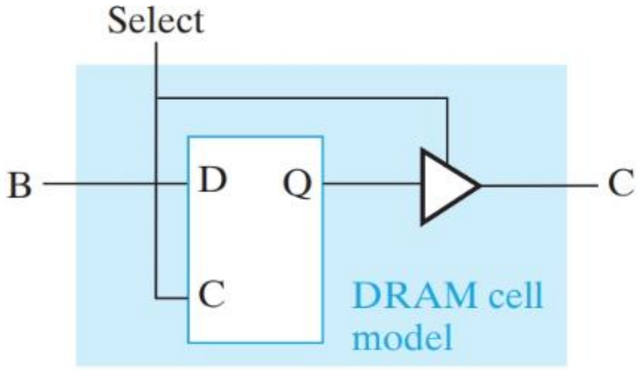


FIGURE 7-12
Dynamic RAM cell, hydraulic analogy of cell operation, and cell model

- In the read operation, the content in C is destroyed (destructive read)
- To be able to read the original value in the future, a restore operation is needed
- Also with the closed valve (T opened), there are some leaks of charge→ refresh

DRAM bit slice

- *Latch D as model for a* DRAM cell



- From the standpoint of cost per bit, the two RAM bit slices are different:
  - DRAM →Transistor + capacitor (2)
  - SRAM → 6 transistors (6)
- From the cell structure, the two RAM bit slices are logically similar, but the number of components is definetely lower.
- Given a fixed size, the number of cells in SRAM < number of cells in DRAM
- DRAM capacity is higher → used in large memories

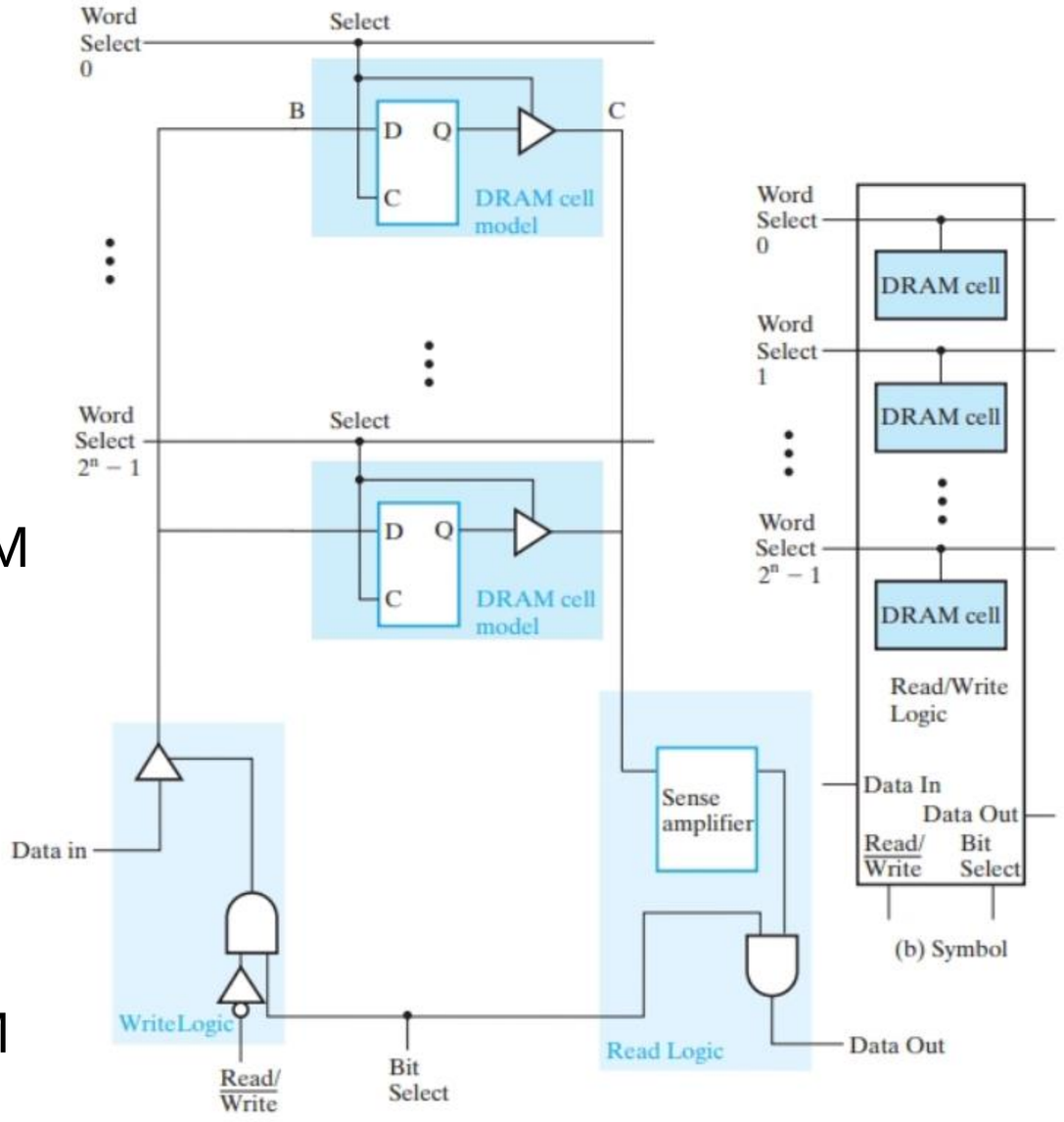


FIGURE 7-13
DRAM Bit-Slice Model

Addressing in DRAM memories

- Large DRAM requires 20 or more address bits (e.g., 20 address pins in each DRAM chip)
- To reduce the number of pins, the structure is slighty different than SRAM
- The DRAM address is applied serially in two parts:
  - The **row address** first
  - The **column address** second
- The row and column addresses are stored in registers.

  $\overline{RAS}$ (Row Address Strobe) and $\overline{CAS}$ (Column Address Strobe) are exploited to enable the time access to registers

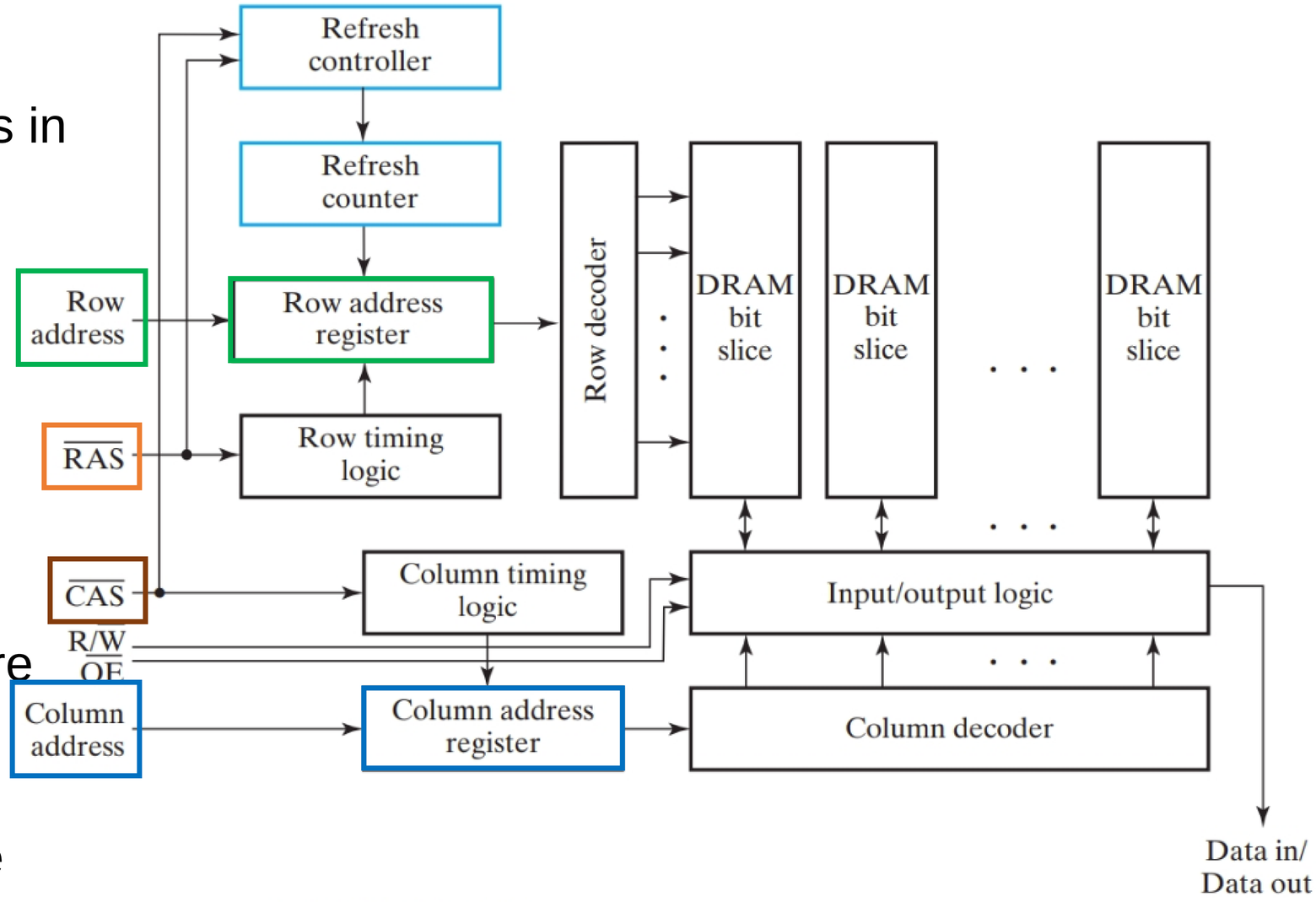
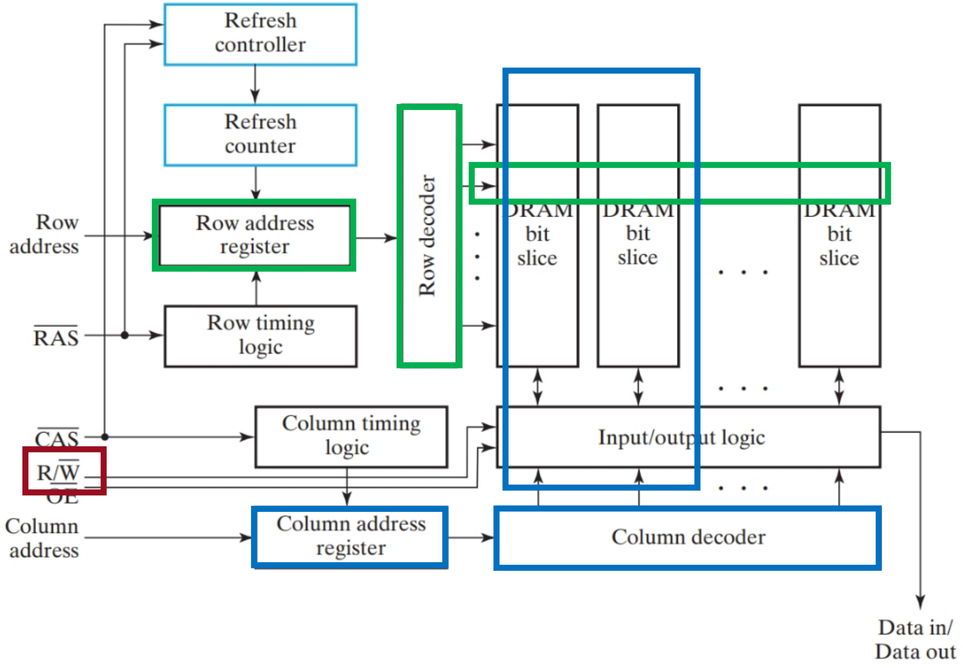
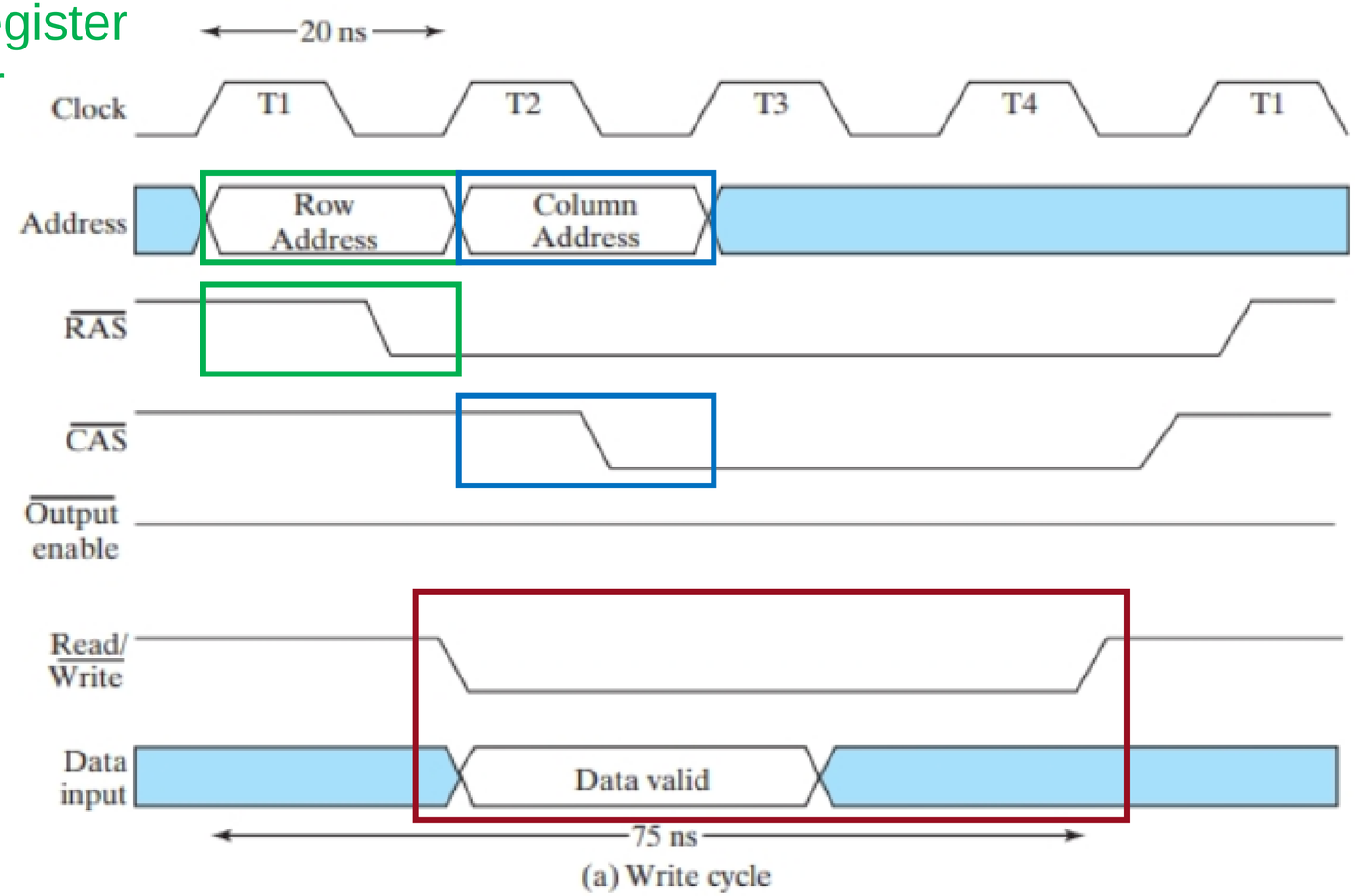


FIGURE 7-14
Block Diagram of a DRAM Including Refresh Logic

Timing for DRAM memories: Write

1. Row adress is applied to the address input
2. RAS →0, row address is loaded into the register
3. Row address is applied to the row decoder
4. Row is selected
5. Column address
6. CAS →0, column address is loaded
7. Column address is applied to the column decoder
8. Column is selected



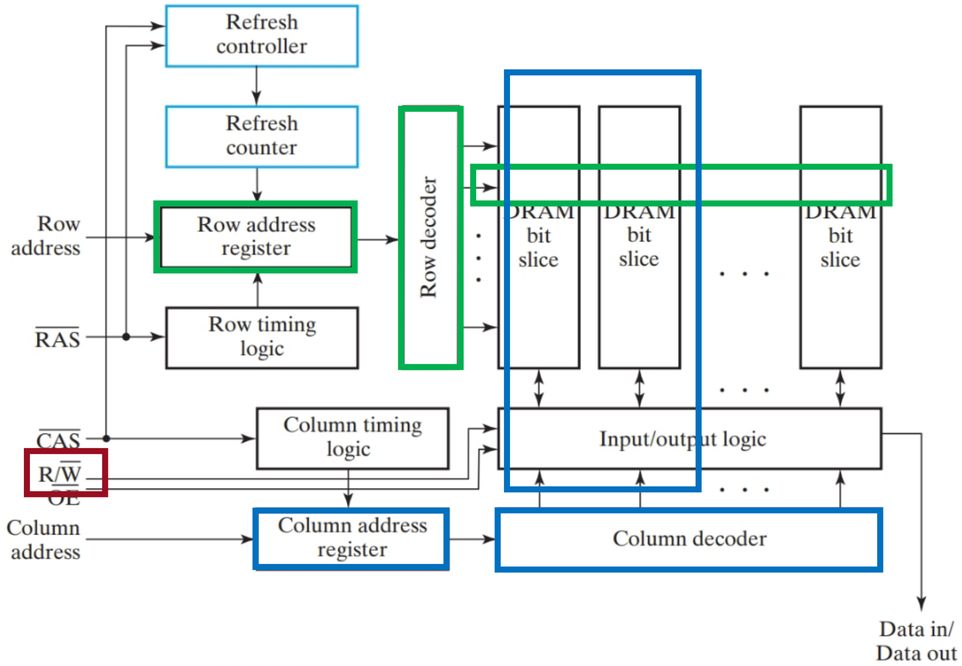


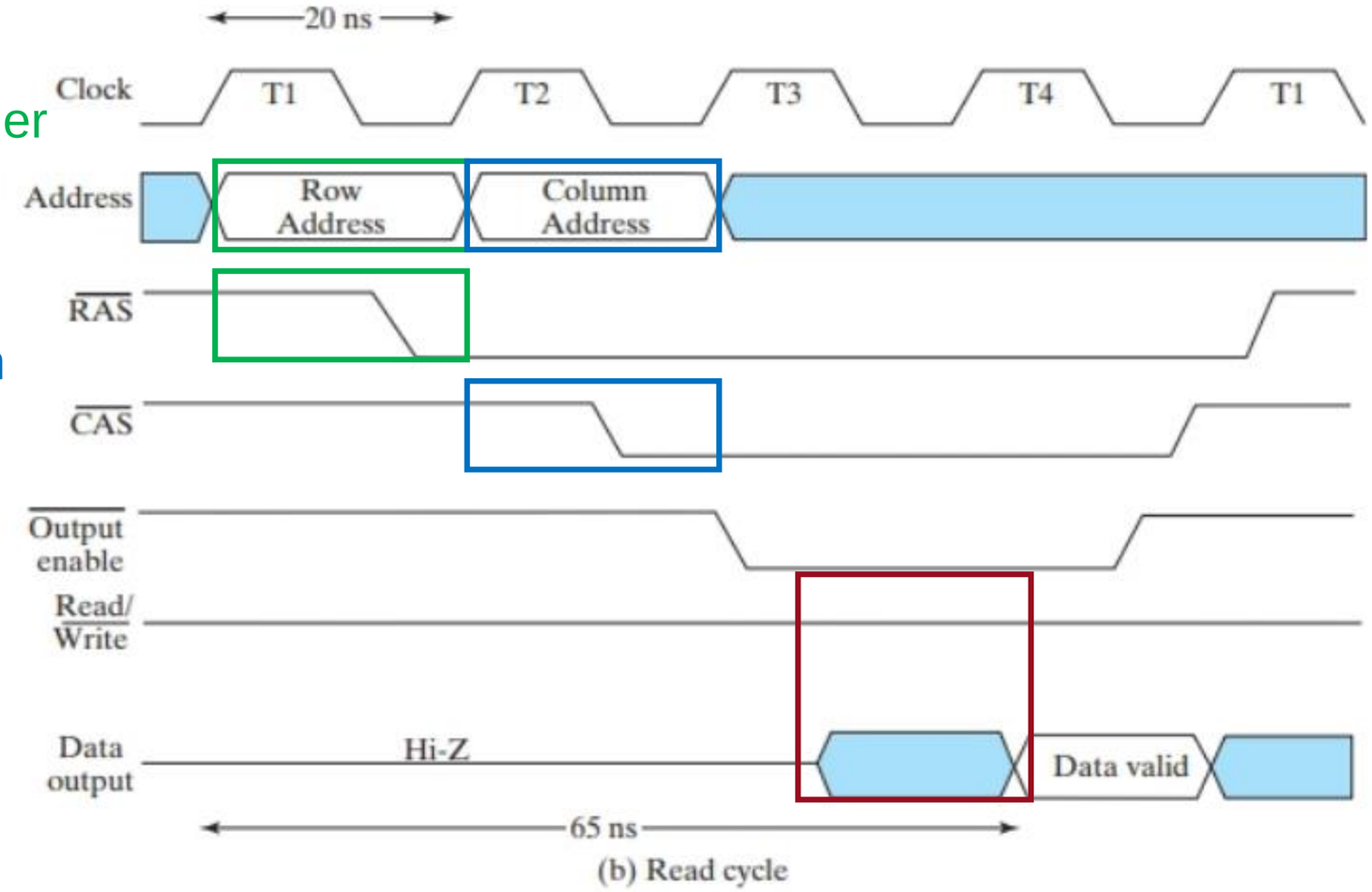
(a) Write cycle

- RW → 0, Input data are written in the selected cells
- The stored data in all of the other cells are restored

Timing for DRAM memories: read

1. Row adress is applied to the address input RAS →0, row address is loaded into the register
2. Row address is applied to the row decoder
3. Row is selected
4. Column address
5. CAS →0, column address is loaded
6. Column address is applied to the column decoder
7. Column is selected





(b) Read cycle

- RW → 1, The selected cells are read
- The stored data in all of the other cells are restored

Refresh in DRAM memories

- **Refresh counter**: the address of the row of DRAM cells to be refreshed
- **Refresh controller**: enables the refresh
- Three ways to trigger a refresh:
  1. **RAS-only refresh**
     - Refresh of a specific row
     - It is must be applied from outside
  2. **CAS-before-RAS refresh**
     - CAS →0 and RAS →0
     - It is controlled by the refresh counter
  3. **Hidden refresh**
     - Following a normal read or write
     - The time take to read/write data is significant

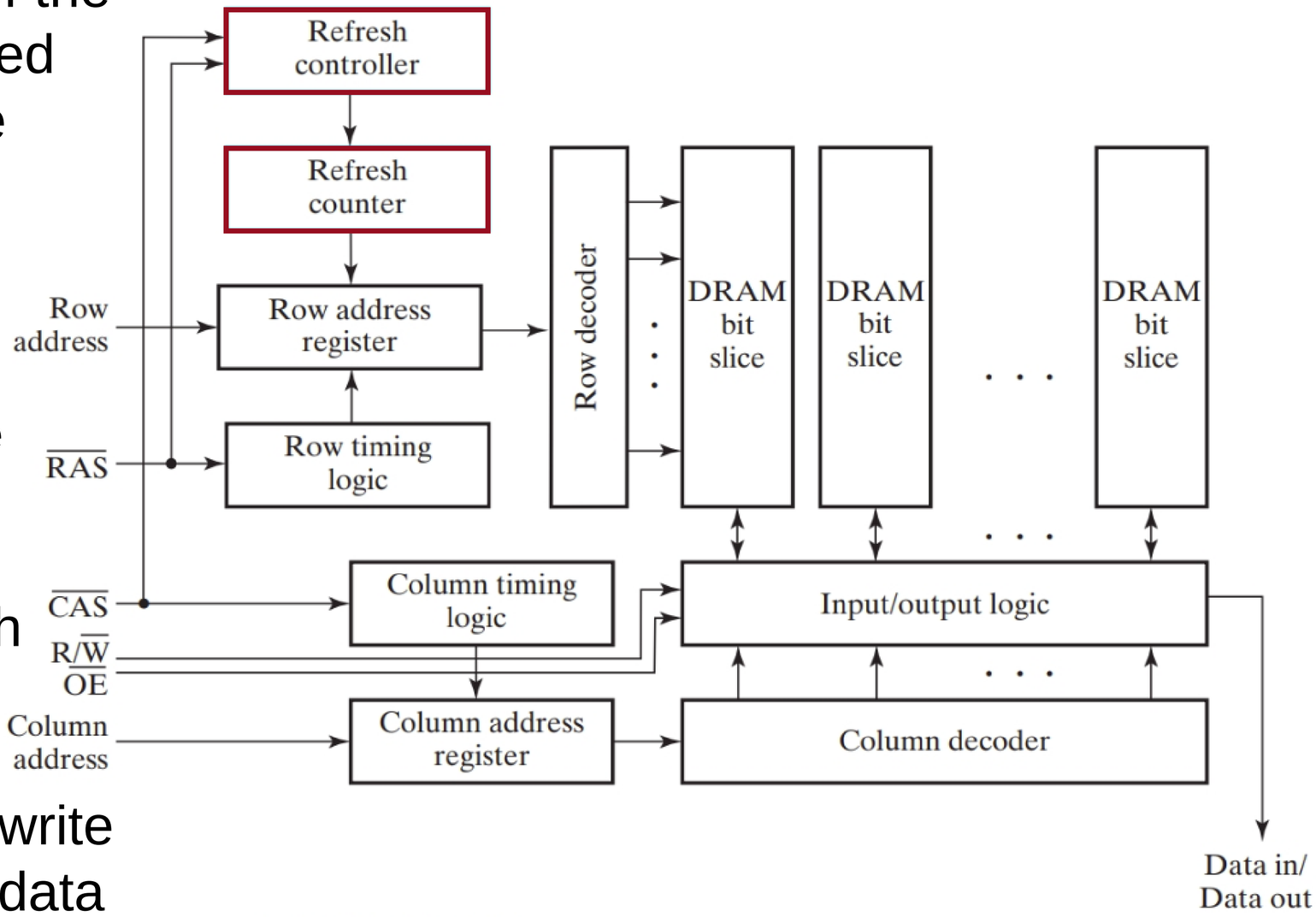
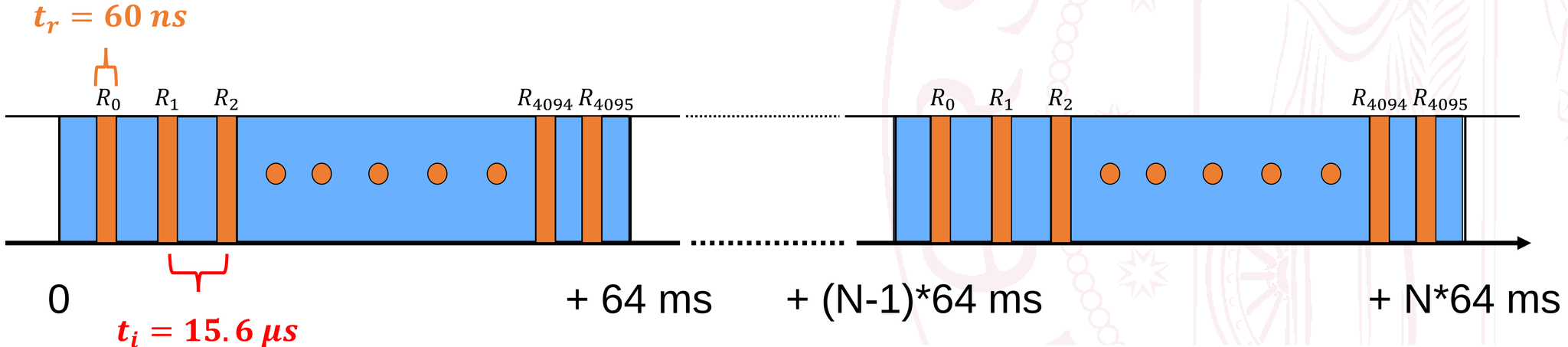


FIGURE 7-14
Block Diagram of a DRAM Including Refresh Logic

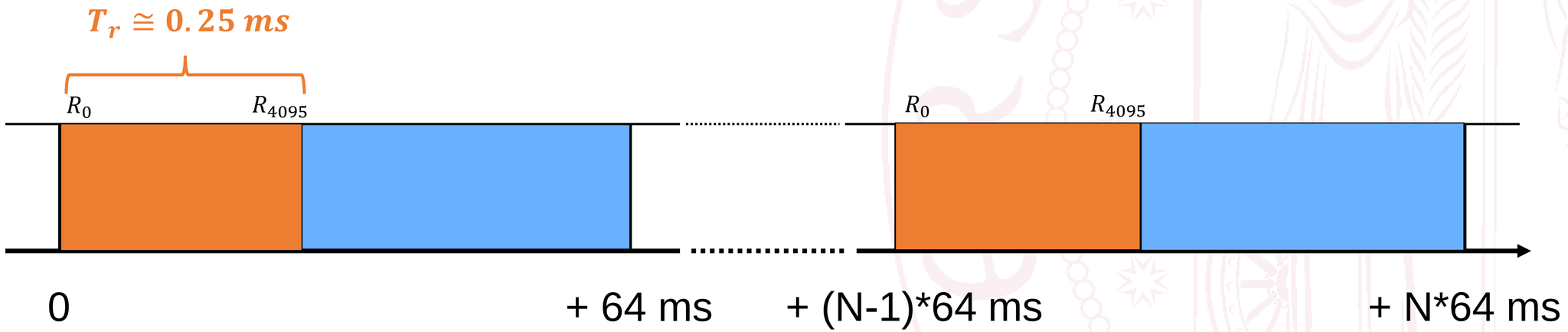
Refresh in DRAM memories

- Each row of a DRAM chip requires refreshing within a specified maximum refresh time, typically ranging from 16 to 64 ms.
- **Distributed Refresh**: refreshes may be performed at evently spaced points in the **refresh time**
  - **Example:** 4M x 4 DRAM has a refresh time of 64 ms
  - $N_r = 4 \cdot 1024 = 4096$ rows to be refreshed
  - **A single refresh** is executed in $t_r = 60\ ns$
  - **The refresh interval for distributed refresh** is: $t_i = \frac{64\ ms}{4096} = 15.6\ \mu s$
  - A total time out for refresh is $T_R = t_r \cdot N_R = 60\ ns\ \cdot 4096 = 0.00024576 \cong 0.25\ ms$
  - During any refresh cycle, no DRAM reads or writes can occur



Refresh in DRAM memories

- Each row of a DRAM chip requires refreshing within a specified maximum refresh time, typically ranging from 16 to 64 ms.
- **Burst refresh**: all the refreshes may be performed one after the other (**refresh time**)
  - **Example:** 4M x 4 DRAM with refresh time of 64 ms
  - $N_r = 4 \cdot 1024 = 4096$ rows to be refreshed
  - A **single refresh** is executed in $t_r = 60\ ns$
  - The **4096 refreshes** are performed one after the other
  - A total time out for refresh is $T_R = t_r \cdot N_R = 60\ ns\ \cdot 4096 = 0.00024576 \cong 0.25\ ms$
  - The memory is busy for the next 0.25 ms



Synchronous DRAM (SDRAM)

Basic concepts:

- In the modern computer, the CPU **does not** directly interact with DRAM
- There is a memory hierarchy: CPU ←→ Cache (L1, L2) ←→Memory
- The reads are in the form of a **line (bytes in contiguous addresses in memory)**
- For effectiveness, it is preferable to **read out of all the bits in a row for each read operation**

16 MB SDRAM

- Similar to the structure of a DRAM
- The addition of a **clock** for synchronous operation
- Control logic is much more complex

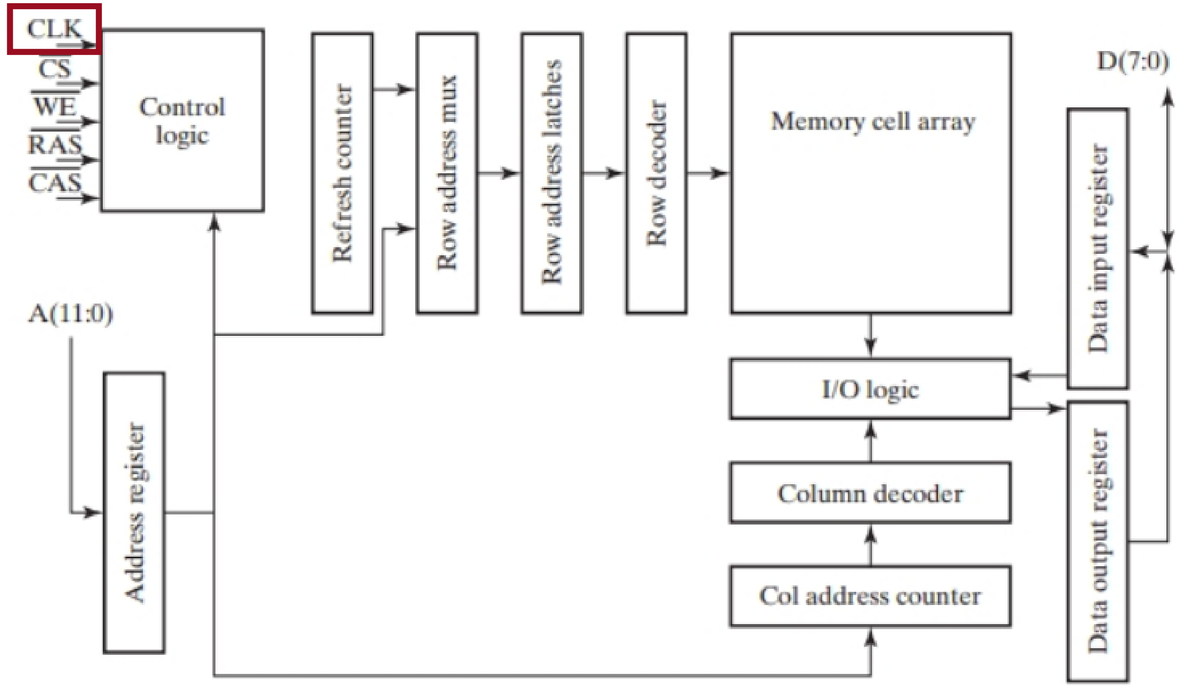


FIGURE 7-16
Block Diagram of a 16 MB SDRAM

Synchronous DRAM (SDRAM)

Esempio 16 MB SDRAM:

- $N_{bits} = 16 \cdot 2^{20} \cdot 8 = 134217728\ bits$
- $N_{rows} = 8192,\ N_{cols} = 16384 \Rightarrow N_{rows} \cdot N_{cols} = N_{bits}$
- $N_{ARbit} = \log_2 8192 = 13\ bits$
- $N_{ACbit} = \log_2 \frac{16384}{8\ bit} = 11\ bits$
- $N_{ARbit} + N_{ACbit} = 24 = \log_2(16 \cdot 2^{20})$

- **Row address** is applied
- The whole role is loaded into the **I/O logic**
- **Column address** is applied
- Data in the **output register**, 1 bit per clock

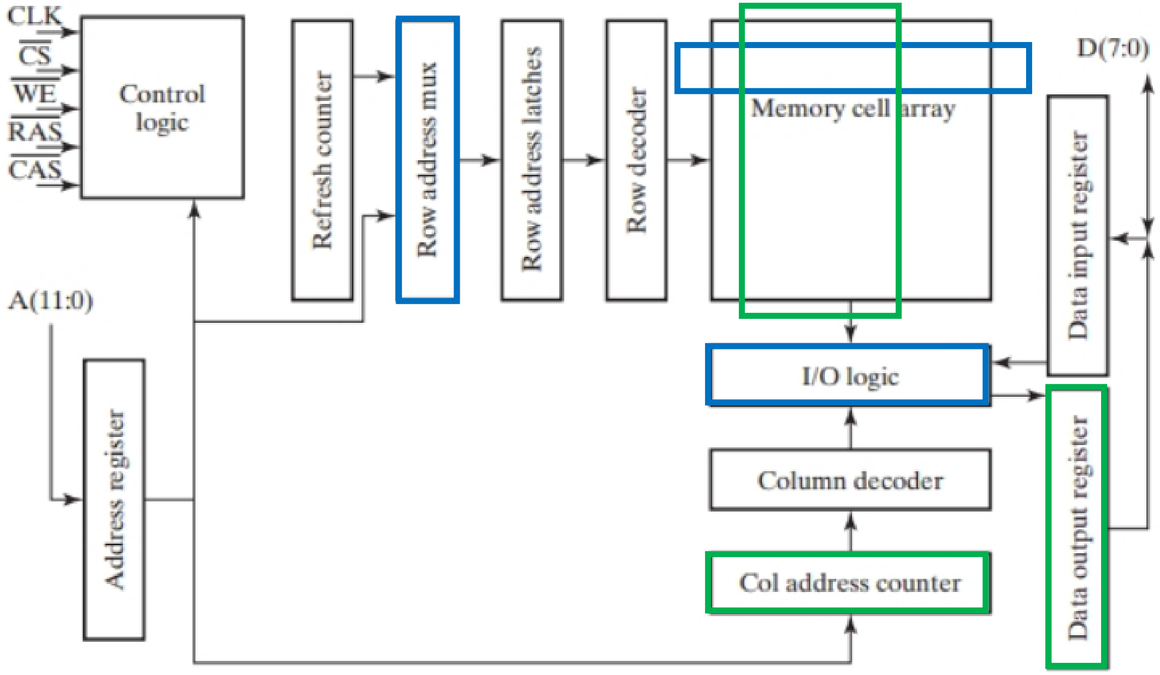


FIGURE 7-16
Block Diagram of a 16 MB SDRAM

Synchronous DRAM (SDRAM)

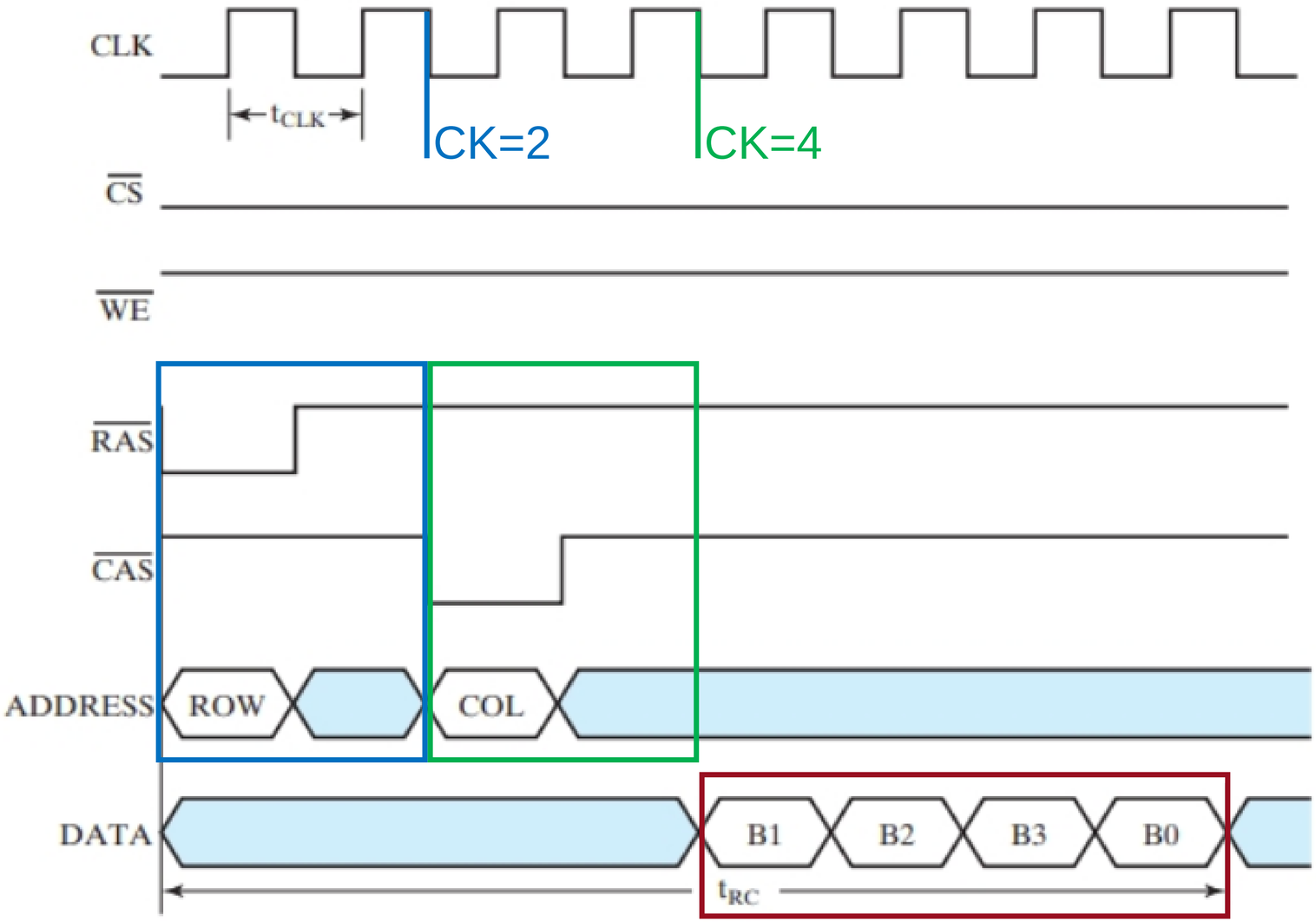


FIGURE 7-17
Timing Diagram for an SDRAM

- **Row address** into the register
- **Reading of the row**
- **Column address** into the register
- **Reading of the bytes**
- **Output of 4 bytes**, one per clock cycle

SDRAM vs. DRAM

DRAM

- Read cycle time: $t_{RC} = 60\ ns$
- 1 byte per read cycle time
- $BitRate = \frac{1}{t_{rc}} = 16.67\ MB/s$

It approximates a byte per clock cycle

SDRAM

To read 4 bytes:

- Clock: $t_{CLK} = 7.5\ ns$
- $t_{RC} = 7.5\ ns \cdot (4\ fixed\ clock\ cycles + 4\ clock\ cycles) = 60\ ns$
- $BitRate = \frac{1}{t_{rc}} = 66.67\ MB/s$

To read 8 bytes:

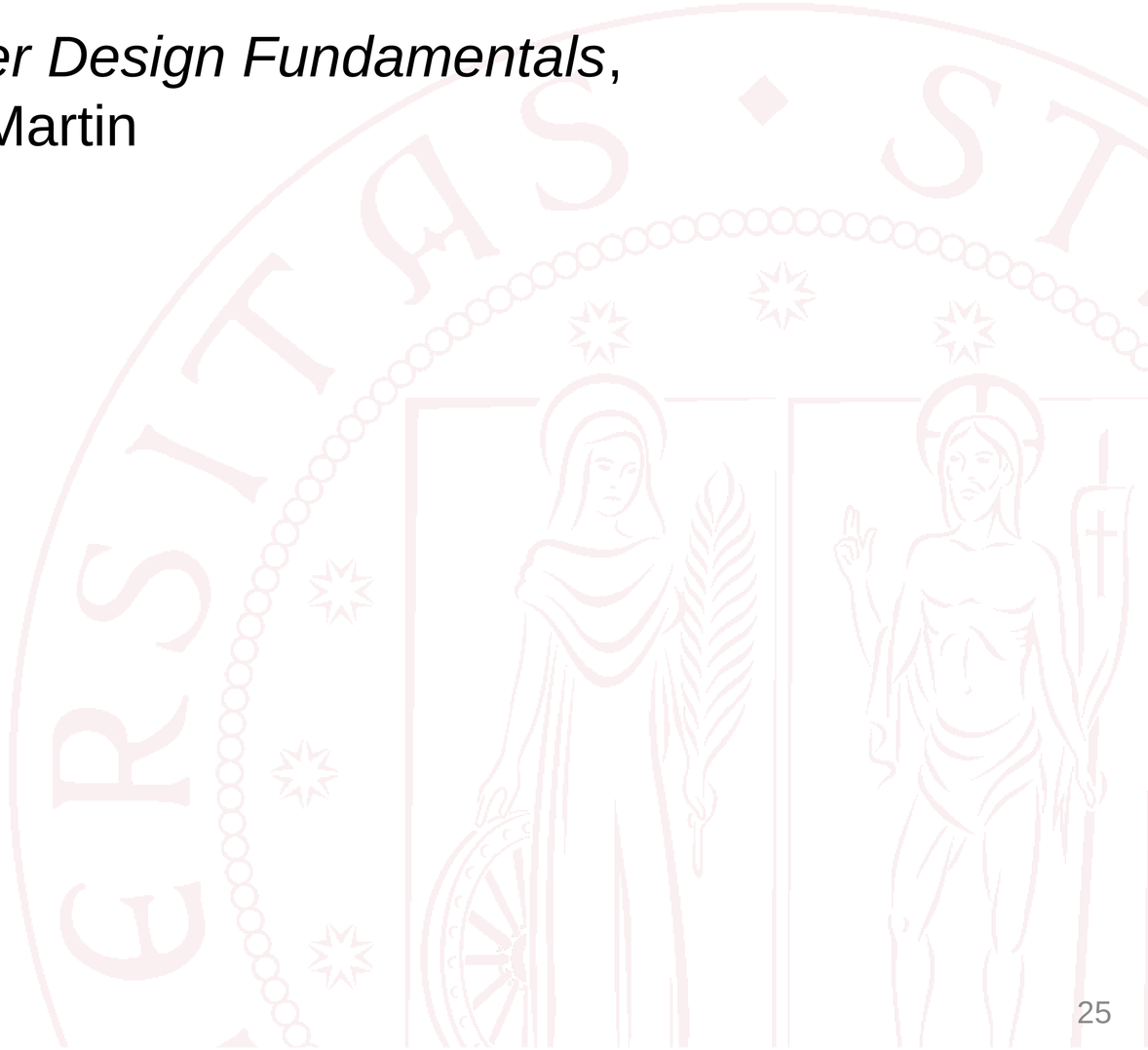
- $t_{RC} = 7.5\ ns \cdot (4\ fixed\ clock\ cycles + 8\ clock\ cycles) = 90\ ns$
- $BitRate = \frac{8}{t_{RC}} = 88.89\ MB/s$

To read the entire row (2048 bytes):

- $t_{RC} = 7.5\ ns \cdot (4\ fixed\ clock\ cyles + 2048\ clock\ cycles) = 15390\ ns$
- $BitRate = \frac{2048}{t_{RC}} = 133.07\ MB/s$

Disclaimer

Figures from *Logic and Computer Design Fundamentals*,
Fifth Edition, GE Mano | Kime | Martin



Questions