Memory – Part A

Types of memories, memory management

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

Topics:

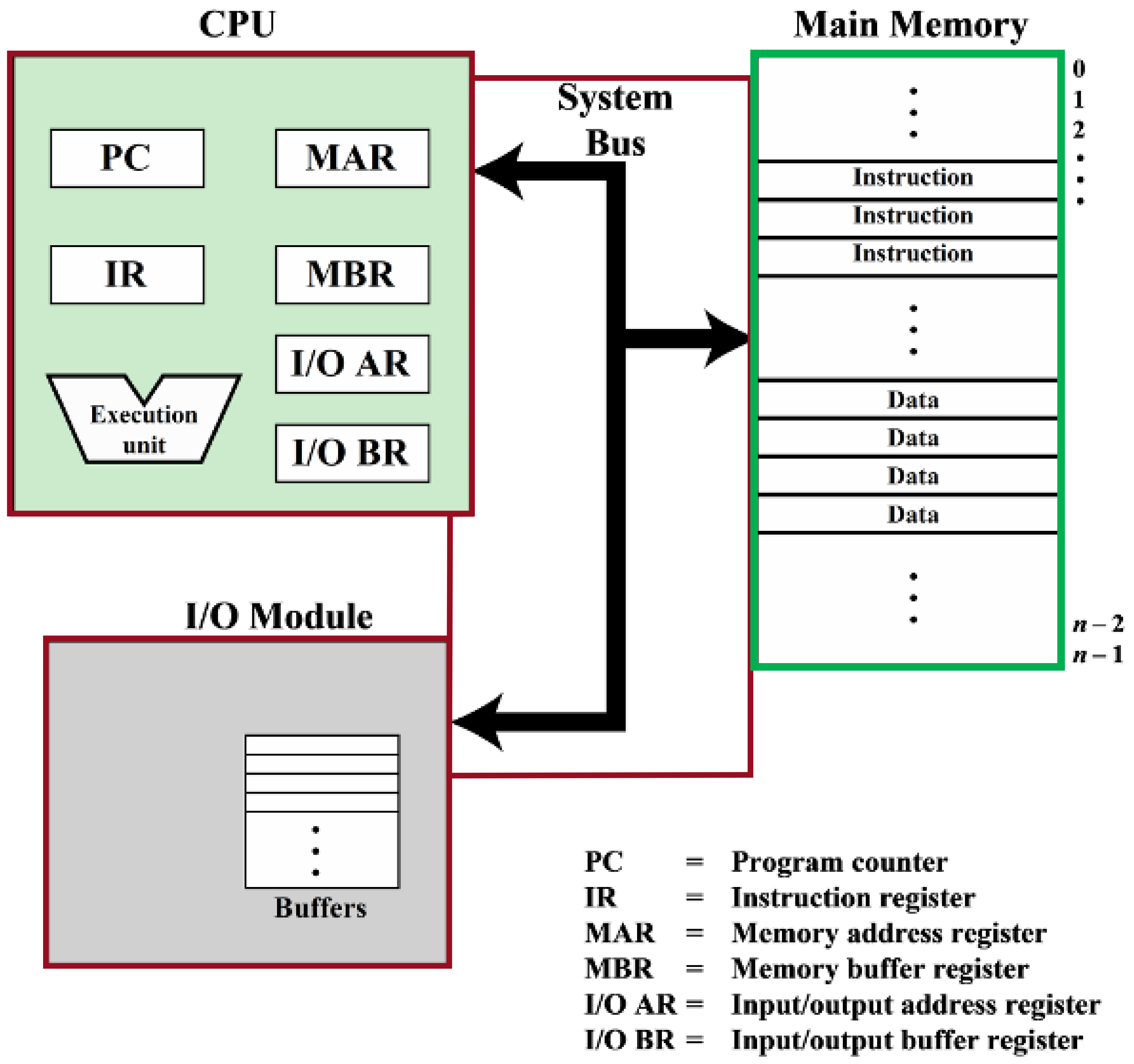
- Types of memories
- RAM, SRAM, DRAM e SDRAM
- Array of memories SRAM
- Addressing, timing, refresh of the memories DRAM

Book Reference:

- Chapter 7

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Computer architecture



PC	=	Program counter
IR	=	Instruction register
MAR	=	Memory address register
MBR	=	Memory buffer register
I/O AR	=	Input/output address register
I/O BR	=	Input/output buffer register

A simple computer is composed of:

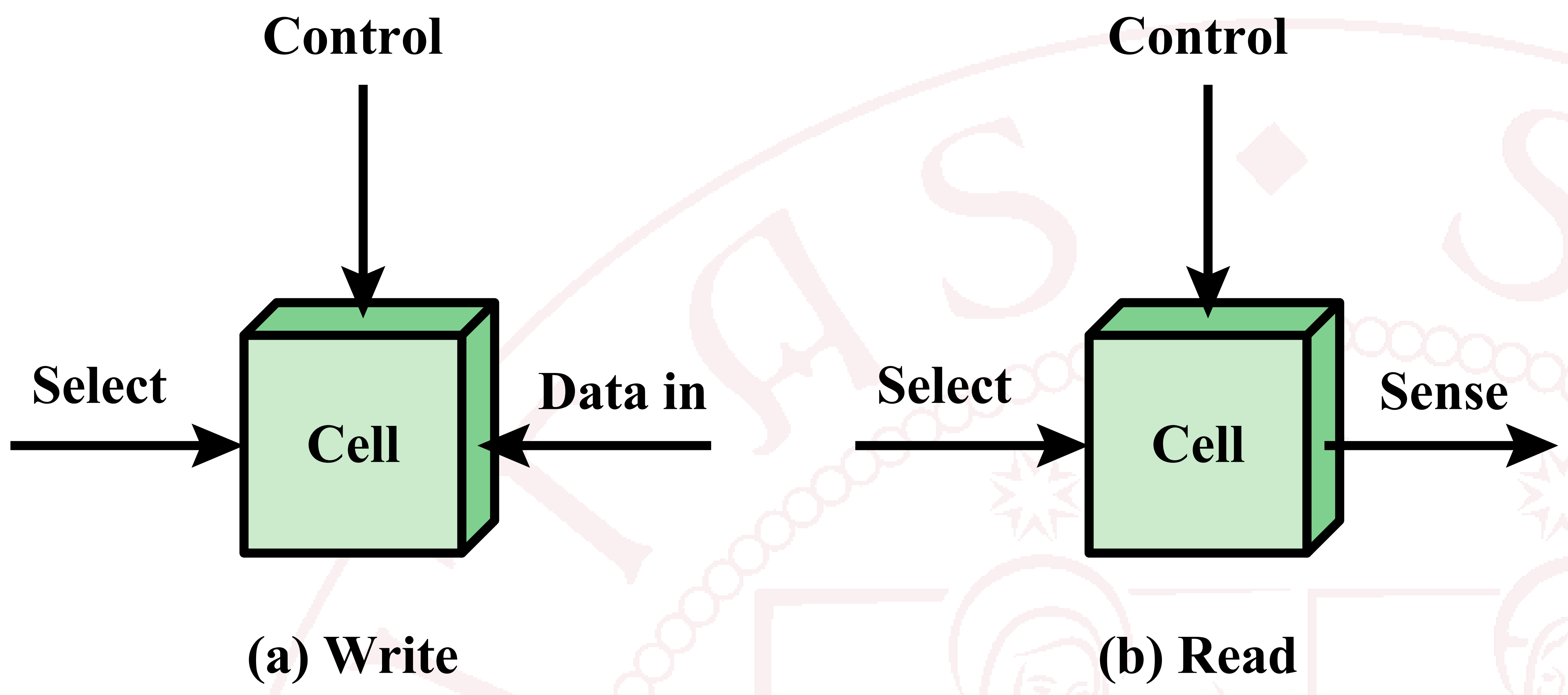
- CPU
- Input/Output module
- Bus of the system
- Central memory

Memories are used by several components in the computer, for example:

- CPU (MAR, MBR)
- I/O Module (buffers)
- Central memory

Memory definitions

- A memory is a collection of cells capable of storing binary information
- Each cell can have two values: 1 or 0
- Data are retrieved from memory, processed from ALU and the output is re-stored in the memory (in the same or different location)
- In the I/O devices, data from/to the device are temporaly stored in memories (buffer)



Kind of memories

We can classify the memories into:

SEQUENTIAL MEMORIES

Time to access depends on the position

RANDOM-ACCESS MEMORIES

Time to access is constant

Types of memories

- Two types of memories are used in the computer:
  - Random-Access Memory (RAM)
  - Read-Only Memory (ROM)

ROM

- Persistent memory
- They can be read severeal times

RAM

- Volatile memory
- They can be read and written several times

Types of memories

Memory Type	Category	Erasure	Write Mechanism	Volatility
Random-access memory (RAM)	Read-write memory	Electrically, byte-level	Electrically	Volatile
Read-only memory (ROM)	Read-only memory	Not possible	Masks	Nonvolatile
Programmable ROM (PROM)			Electrically	
Erasable PROM (EPROM)	Read-mostly memory	UV light, chip-level		
Electrically Erasable PROM (EEPROM)		Electrically, byte-level		
Flash memory		Electrically, block-level		

Read-Only Memory (ROM)

- The content of the memory is **permanent** and cannot be modified
- Power supply is not nedeed to store the value in memory
- Data are cabled in the chip during the fabrication process
- Cons:
  - The errors cannot be made because they cannot be corrected
  - Data transfer is expensive

Programmable ROM (PROM)

- The memory content is **permanent** and cannot be changed
- The writing can be done **after** the fabrication process thorugh a specific electric equipment
- Power supply is not nedeed to store the value in memory
- Higher flexibility and less cost than ROM

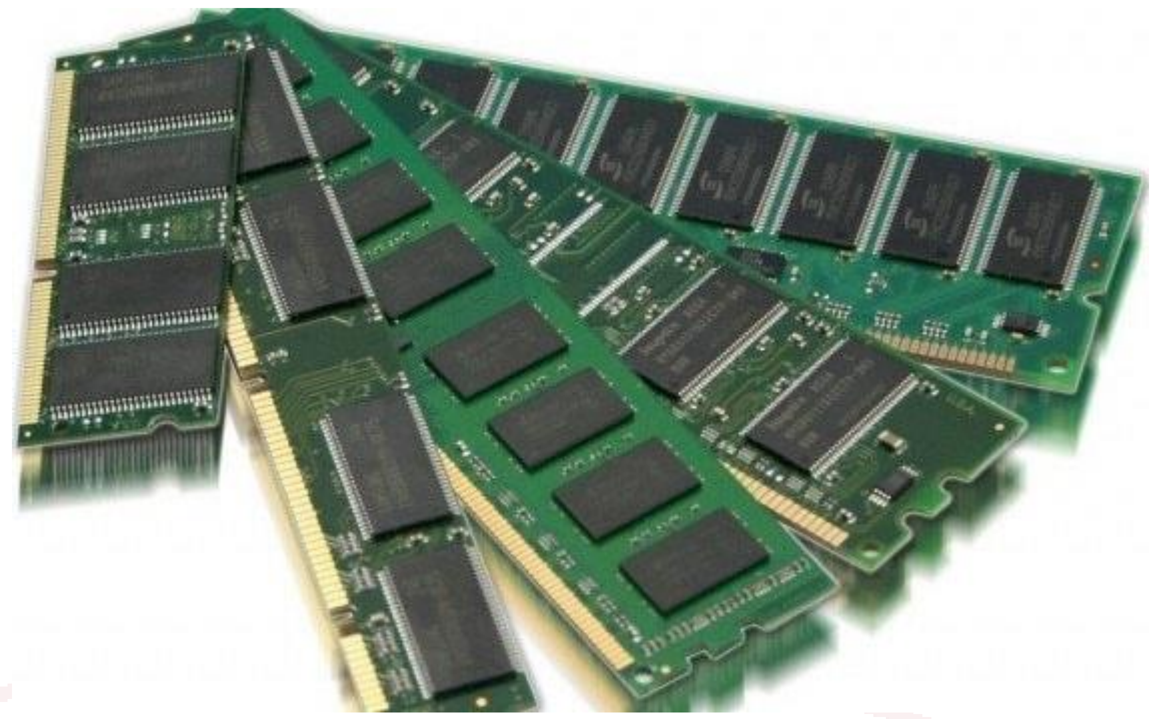
Programmable persistent memories

- Persistent memories allow to modify their content
- **Erasable programmable read-only memory (EPROM)**
  - EPROM can be erased by exposing it to strong ultraviolet light source
- **Electrically erasable programmable read-only memory (EEPROM)**
  - It is possible to erase only single bytes via electrical signals
  - It is more expensive and lower density (less bits per chip) than EPROM: each bit requires two transistors
- **Flash Memory**
  - They are placed in an intermediate position between EPROM and EEPROM considering costs and functionalities.
  - Flash memories are erased per sectors
  - Their density is comparable to the one in EPROM: each bit requires a transistor

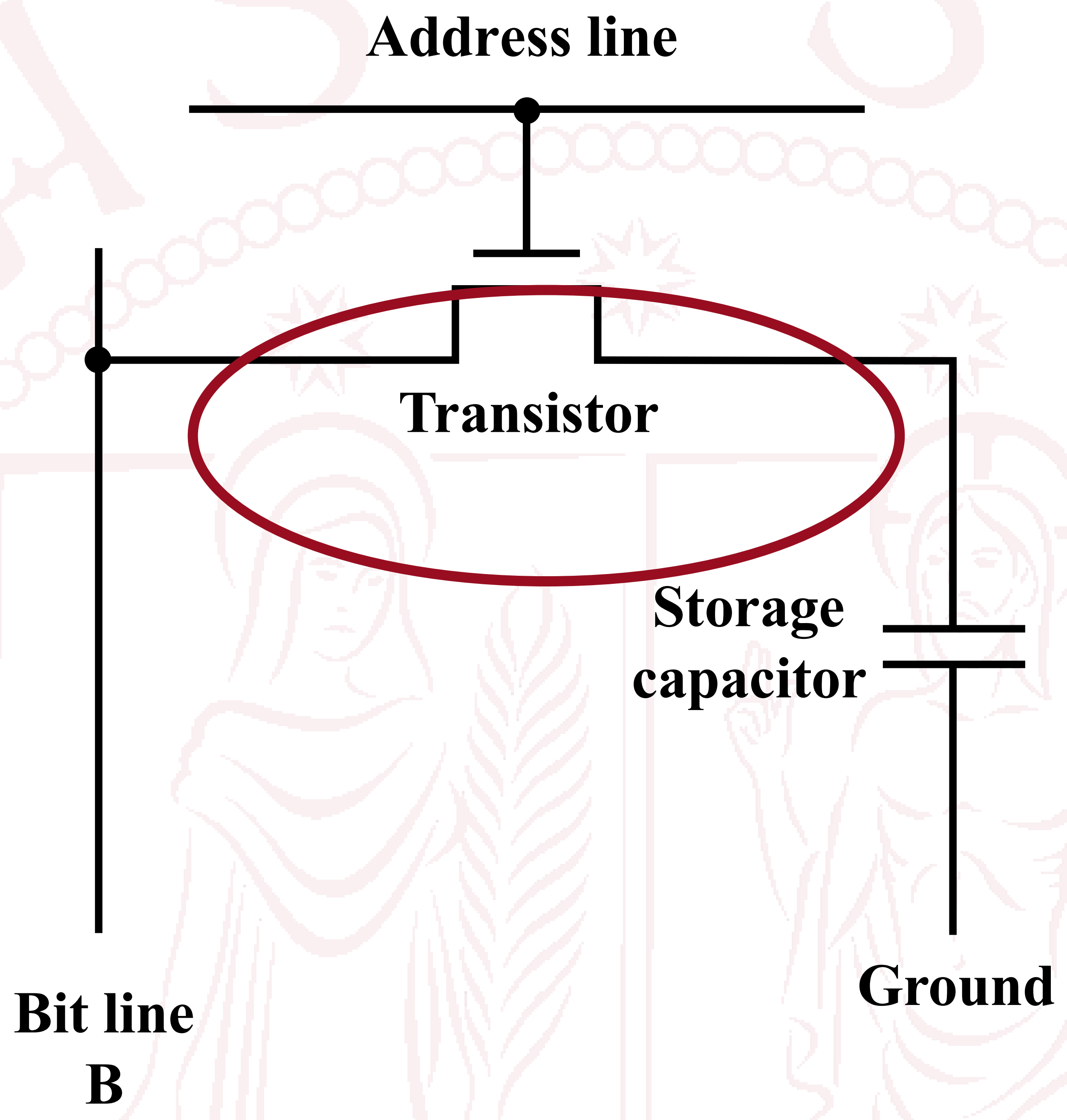
Random-Access Memory

- Memory cells can be accessed to transfer information to or from any desired location (**same time regardless of the location**)
  - In contrast, serial memory takes different times to access information depending on where the target location is (e.g., magnetic-tape data storage)
- RAM is a **volatile memory** that allows **read** and **writing** efficiently
- The main types of RAM memories:
  - **Dynamic RAM (DRAM)**
  - **Static RAM (SRAM)**

Dynamic RAM (DRAM)



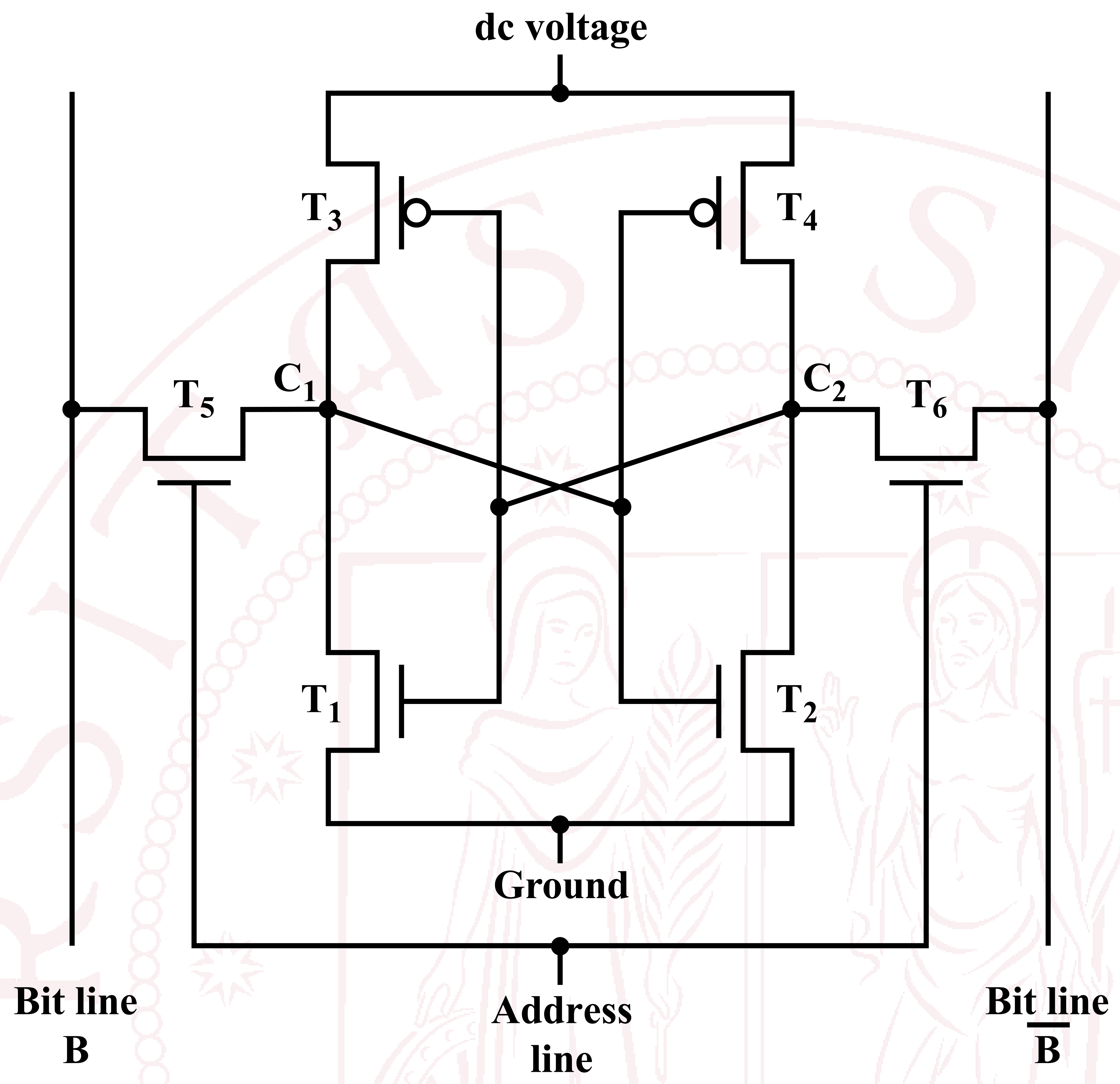
- DRAM stores the binary information in the form of electric charges on capacitors
- The capacitors must be **periodically recharged by refreshing the DRAM** to keep the value over time
  - The stored charge on the capacity tends to discharge with time
  - Refresh: the data are read and then re-written
- Higher density (1 transistor, 1 capacitors), limited access velocity (charge capacitors)



(a) Dynamic RAM (DRAM) cell

Static RAM (SRAM)

- SRAM consists of internal latches that store the binary information
- The stored information remains valid as long as power is applied to the SRAM
- Lower density (6 transistors), high access velocity



(b) Static RAM (SRAM) cell

DRAM vs. SRAM

DRAM		SRAM
• Volatile memory	⟷	• Volatile memory
• Easy to make, low cost	⟷	• High cost
• High density: 2 components per bit	⟷	• Low density: 6 components per bit
• Limited access velocity	⟷	• High access velocity
• Refresh is needed	⟷	• Refresh is not needed
• It is used as main memory	⟷	• It is used as cache

DRAM vs. SRAM: access times

- Static RAM (SRAM):
    - 1.4 ÷ 15 ns (high speed)
    - 35 ÷ 100 ns (low power)

- Dynamic RAM (DRAM):
    - 50 ÷ 70 ns (Asynchronous DRAM)
    - 7 ÷ 12 ns (Synchronous SDRAM)

(+ latency: the first data takes a 4-5 times higher time than the next ones

Properties of Memory (1)

- Binary information is stored in memory in groups of bits
- Each group of which is called a **word**
- A group of **8 bits** is called a **byte**
- Most computer memories use words that are multiples of 8 bits (1 byte)
- The term **word** is used to indicate the entity of bits/bytes that represents the unit information in a computer architecture
- The size of a word is variable
  - word can contain 16 bit (2 bytes), 32 bit (4 bytes), 64 bit (8 bytes), …

Properties of Memory (2)

- The memory unit is specified by the
  - **Number of words** it contains
  - **Numer of bits** in each word
- Each word in memory is assigned an **address (identification number)**
- Addresses range from 0 to $2^k - 1$ where k is the number of address lines
- The selection of a specific word is done by applying the **k-bit binary address to the address lines**
- A decoder select the corresponding word
- Note: $2^{10} = 1024 = 1K$, $2^{20} = 1M = 1024 * 1024$, $2^{30} = 1G = 1024* 1024* 1024$

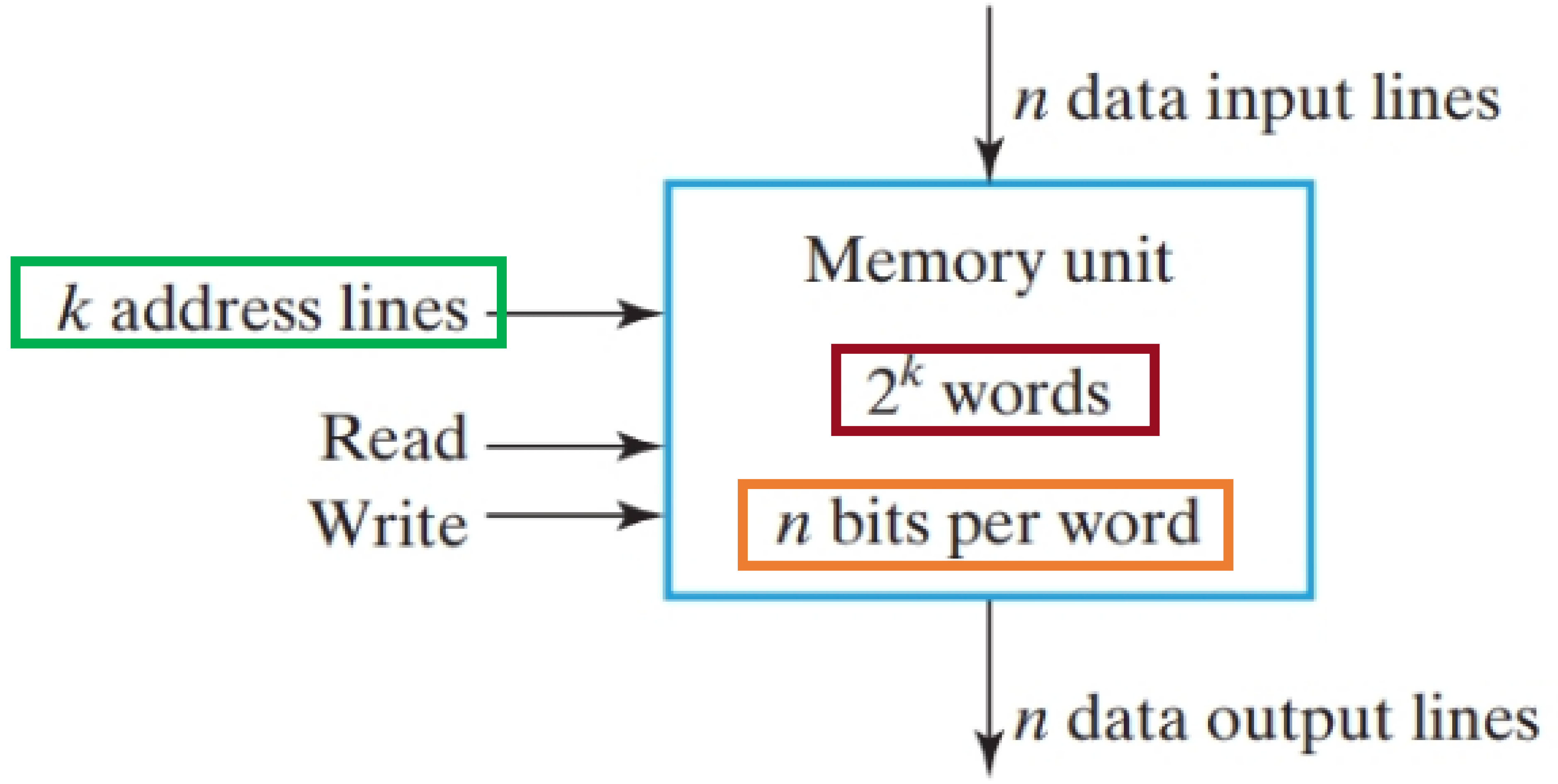


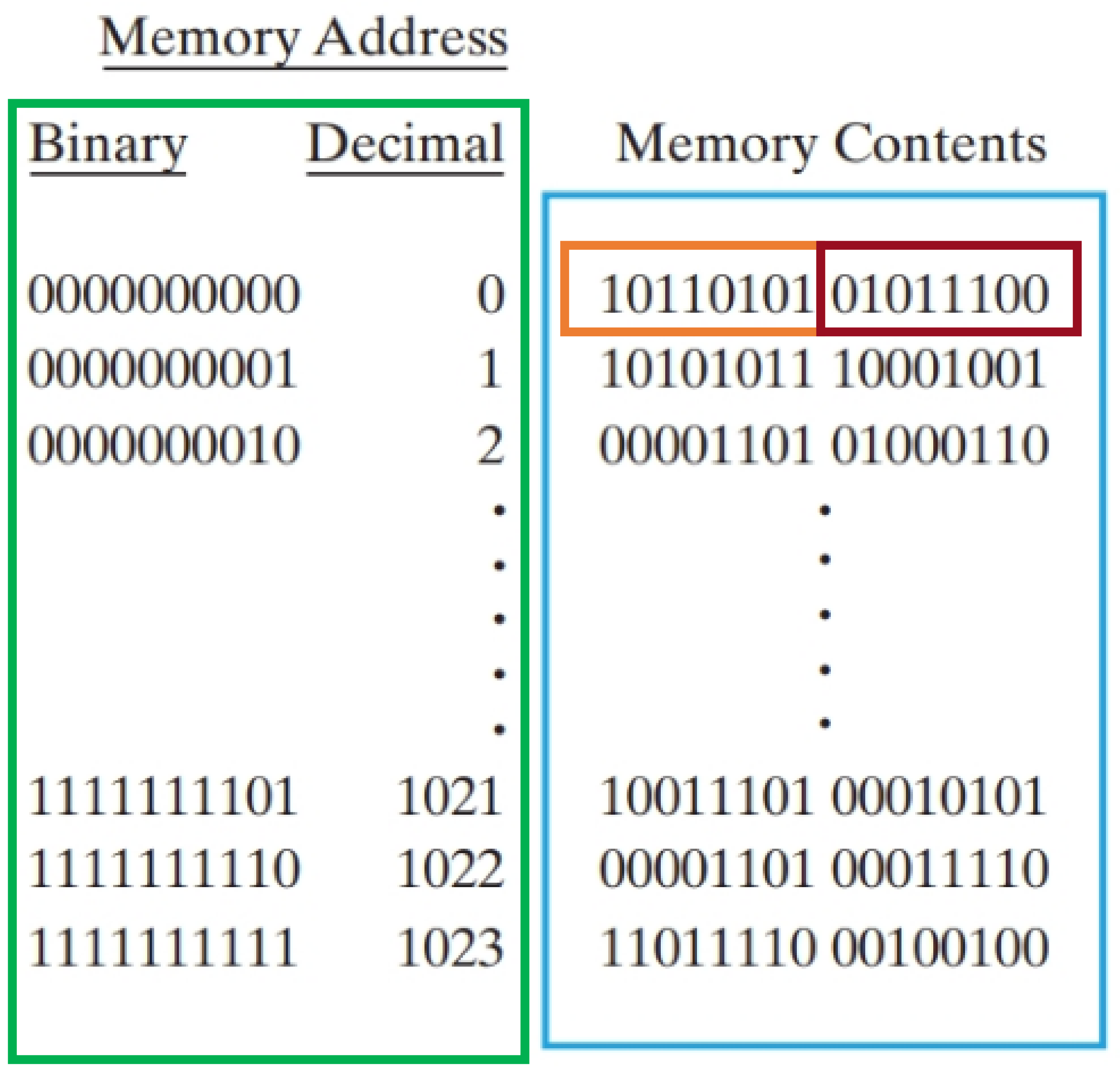
FIGURE 7-1
Block Diagram of Memory

Properties of Memory (3)

- The capacity of a memory is usually stated as the total number of bytes

- For instance, consider a memory with a capacity of 1K words of 16 bits each
  - $1K = 1024 = 2^{10}$
  - 16 bits = 2 bytes
  - The memory contains $2048 = 2K = 2 \cdot 2^{10}$ bytes

- The 1024 addresses are identified by number in the range: 0 a 1023 → 0000000000 a 1111111111
- Each address selects a word of 16 bit, namely 2 bytes
- When a word is read or written, the memory operates on all the 16 bits as a single unit
- The number of address bits needed in memory is dependent on the total number of words



Memory Address		
Binary	Decimal	Memory Contents
0000000000	0	10110101 01011100
0000000001	1	10101011 10001001
0000000010	2	00001101 01000110
⋮	⋮	⋮
1111111101	1021	10011101 00010101
1111111110	1022	00001101 00011110
1111111111	1023	11011110 00100100

FIGURE 7-2
Contents of a 1024 × 16 Memory

Endianess: Ordering of bytes in memory

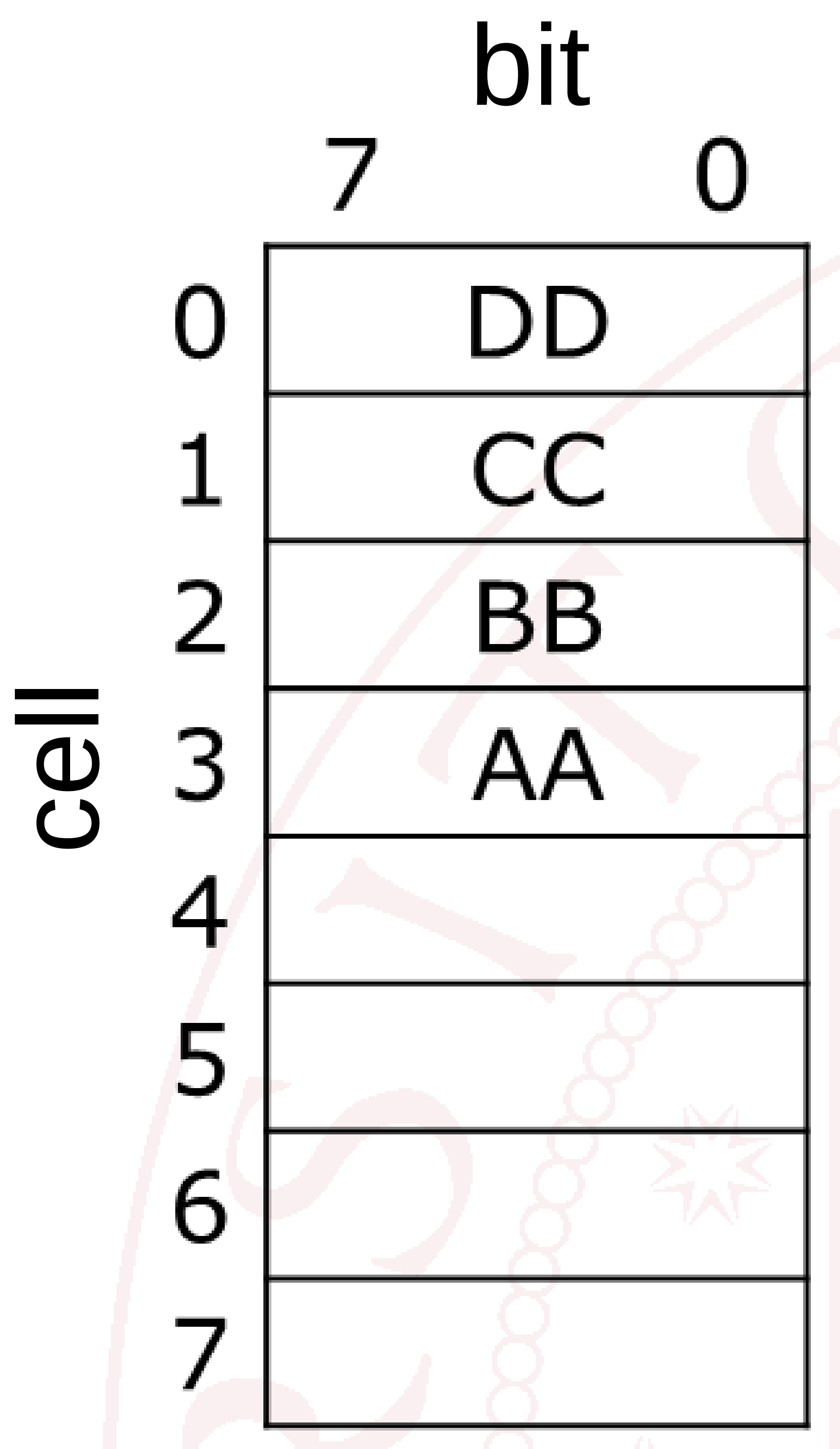
Problem:

Consider a memory organized in byte (1 cell = 1 byte), how is it possible to store a word (4 bytes) in memory?

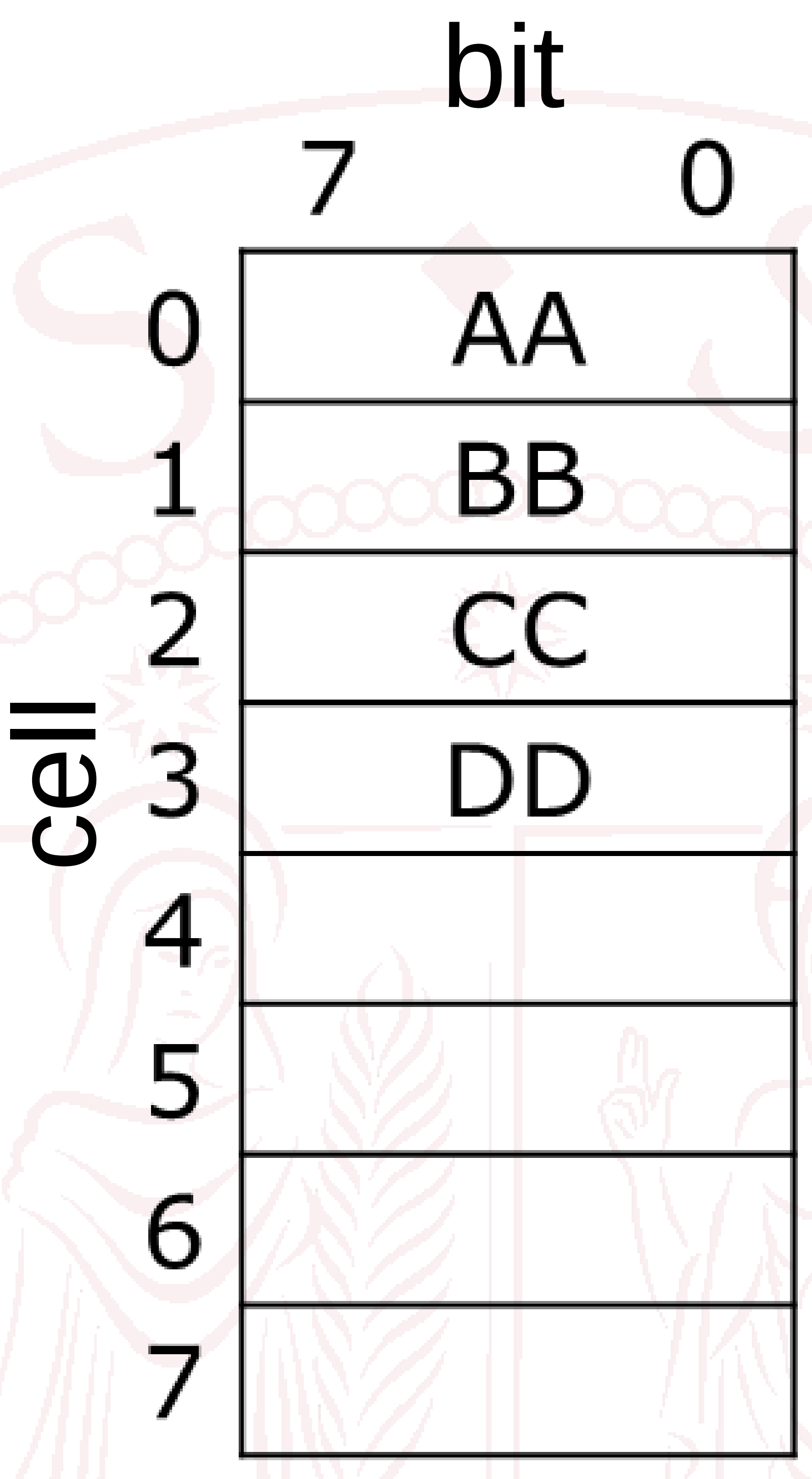
word: 0xAABBCCDD

Two possibilities:

- Big-endian
- Little-endian



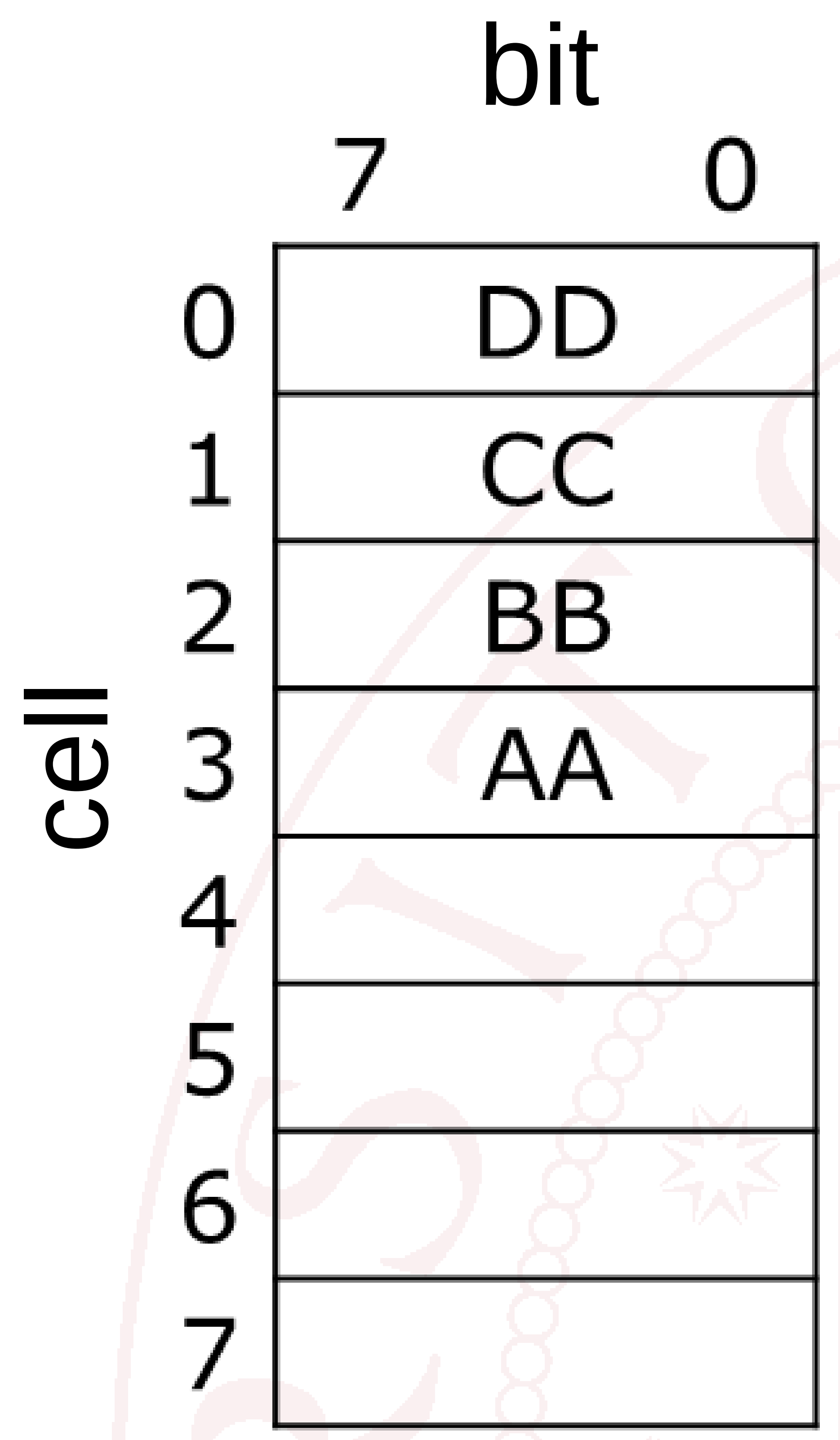
Little-endian



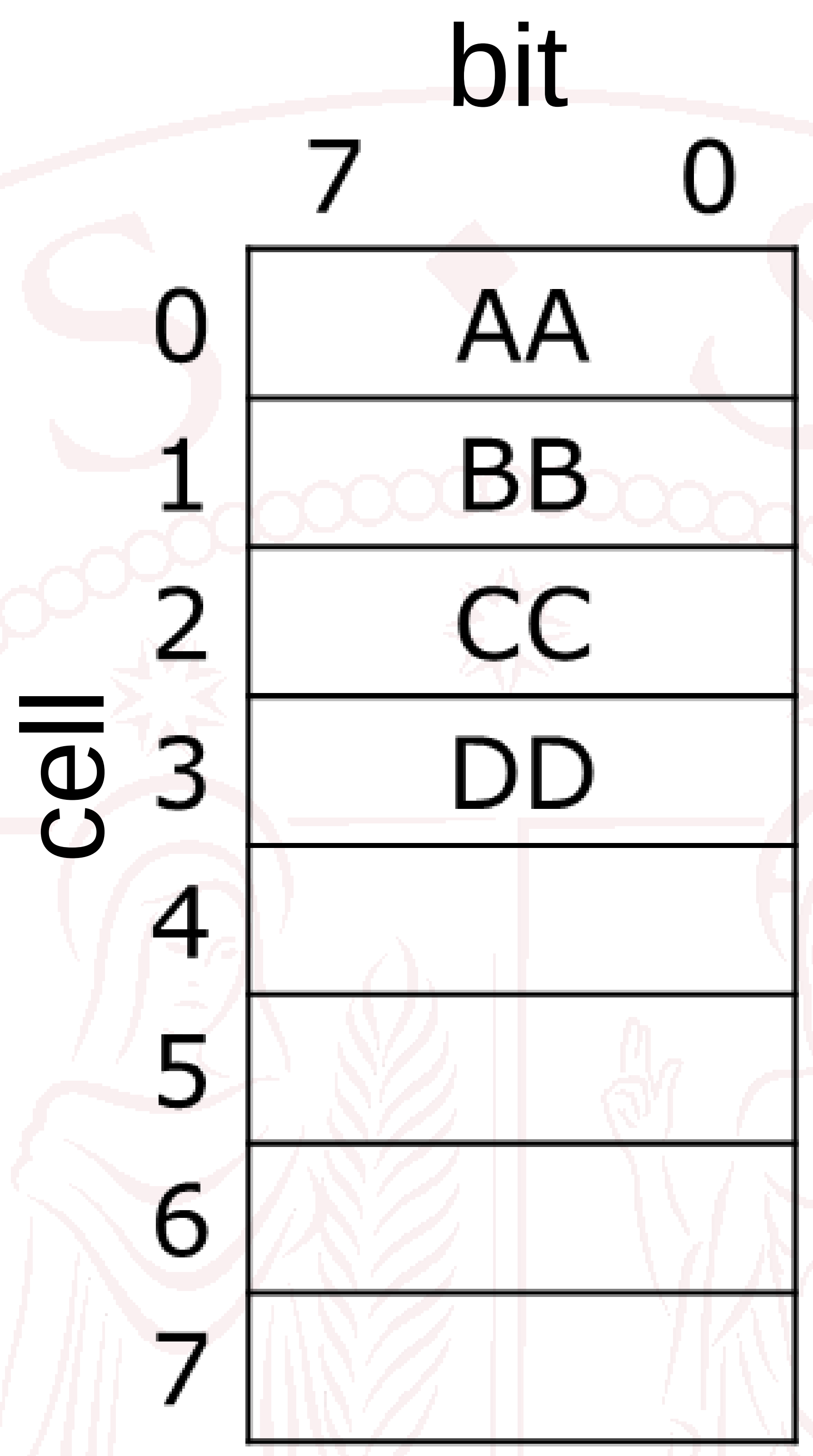
Big-endian

Big-endian vs. little-endian (1)

- **Big-endian** is an order in which the most significant value in the sequence (BIG END) is stored first (the lowest memory address).
- **Little-endian** is an order in which the least significant value in the sequence (LITTLE END) is stored first.
- The two alternatives are both usable and used.



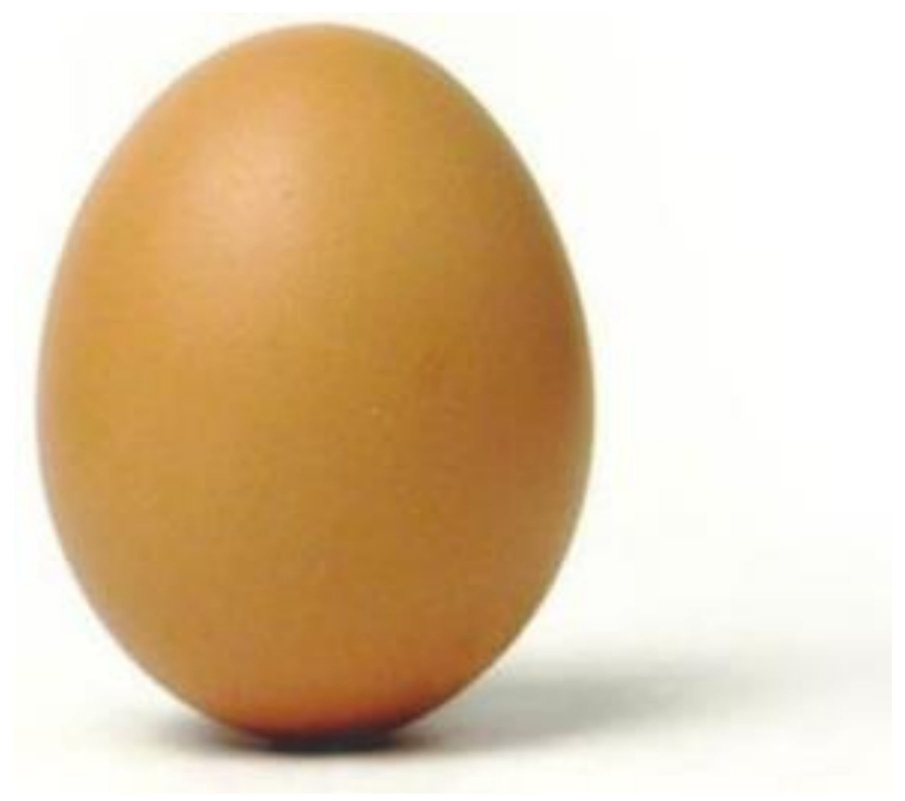
Little-endian

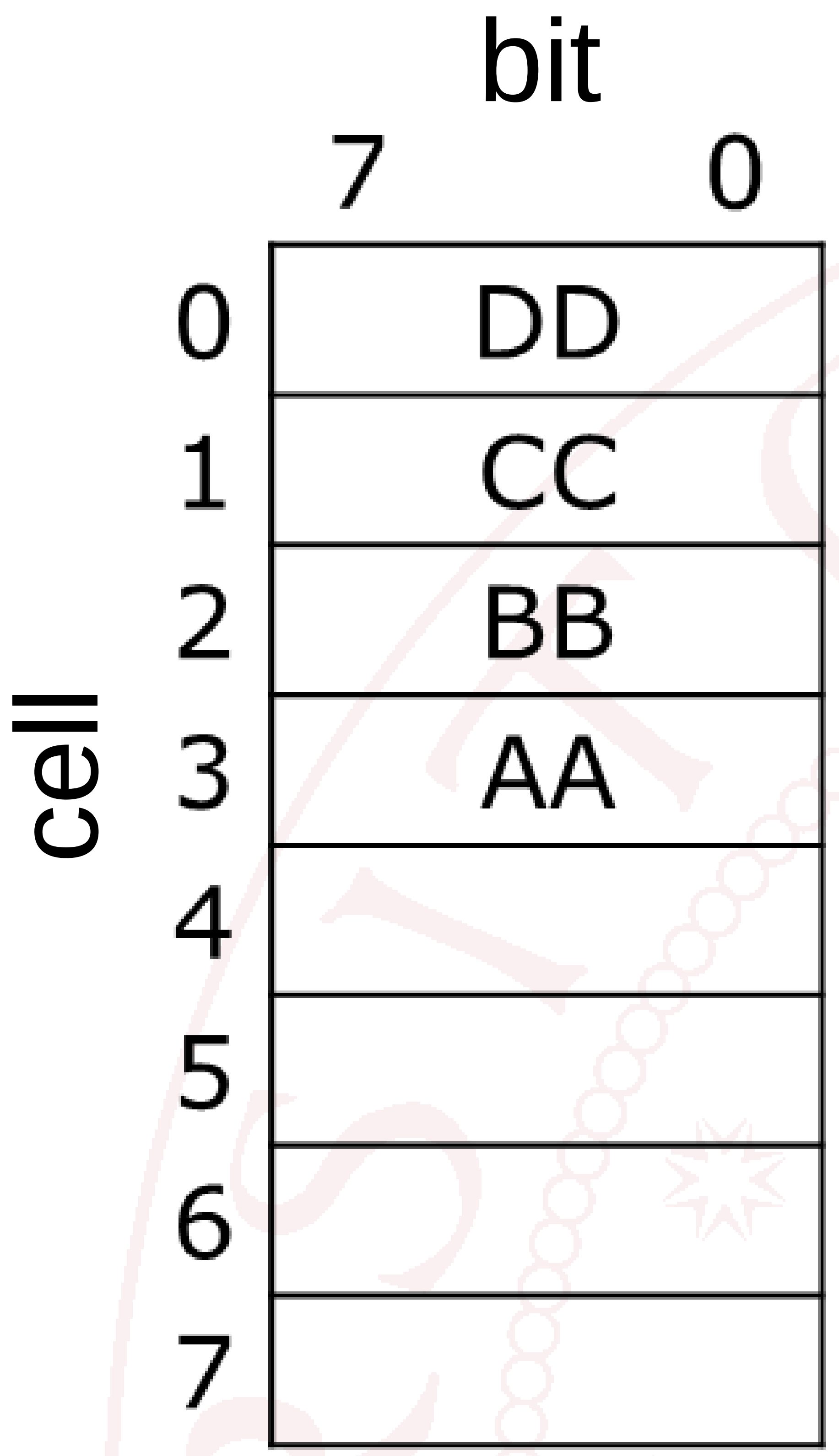


Big-endian

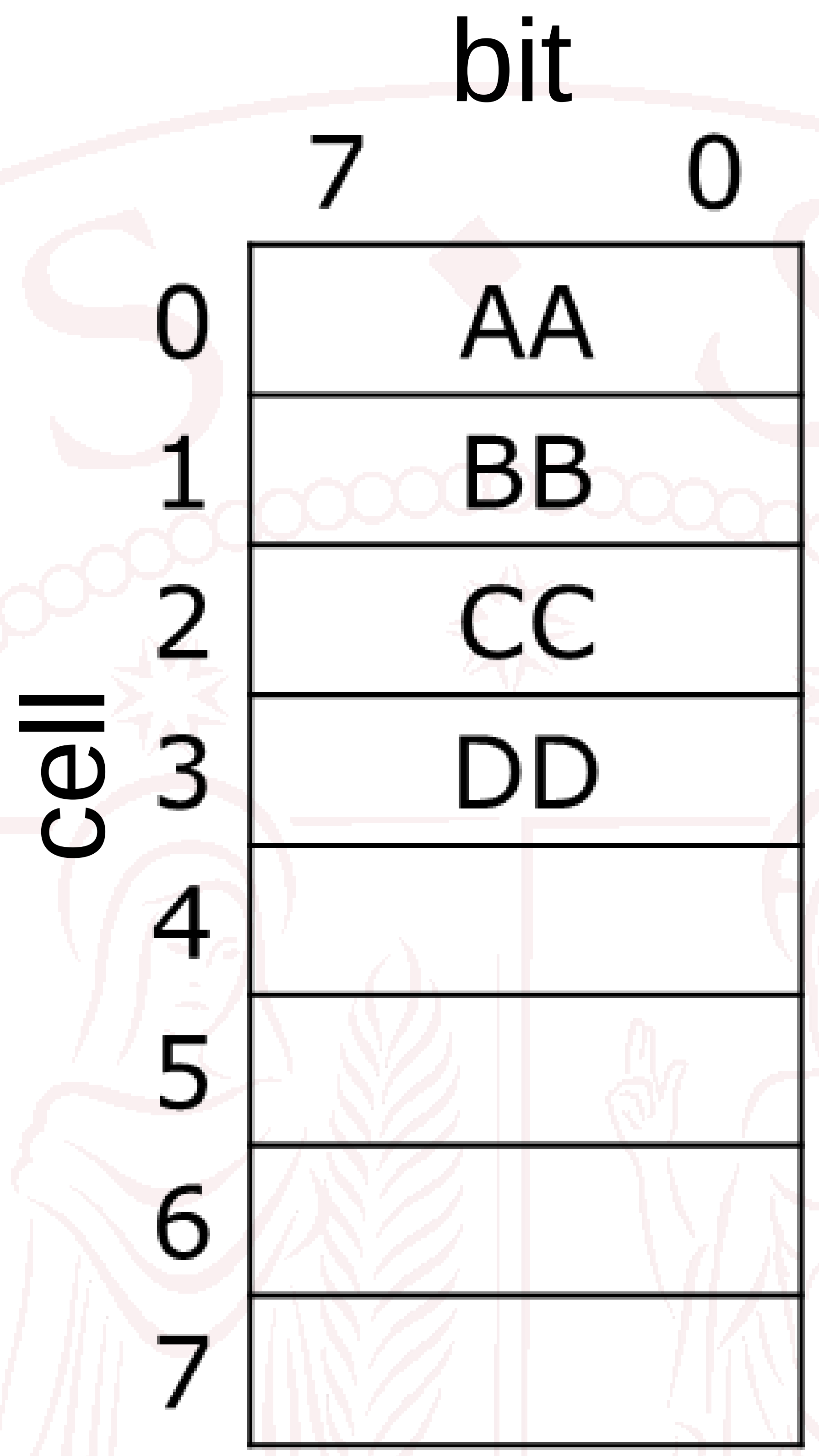
Big-endian vs. little-endian (2)

- The names are inspired by the book “Gulliver's Travels” (Jonathan Swift)
- The Big Endians are a group of people in Lilliput who believe that boiled eggs should be broken at the big rather than at the little end (at the tip), as commanded by the Emperor of Lilliput (the Little Endians).





Little-endian



Big-endian

Ordering of bytes in memory: big-endian or little-endian?

- The historical computer (IBM370) and the Motorola processors use the big-endian method
- INTEL processors use the little-endian ordering, because they are considered the most efficient method in the data transmission when the least significant value is provided

- The PowerPC and ARM support both the modalities

- Similarly, the graphical formats support different ordering:
  - GIF → Little Endian
  - JPEG → Big Endian

SRAM: Static Random Access Memory

- SR Latch as model of a cell SRAM

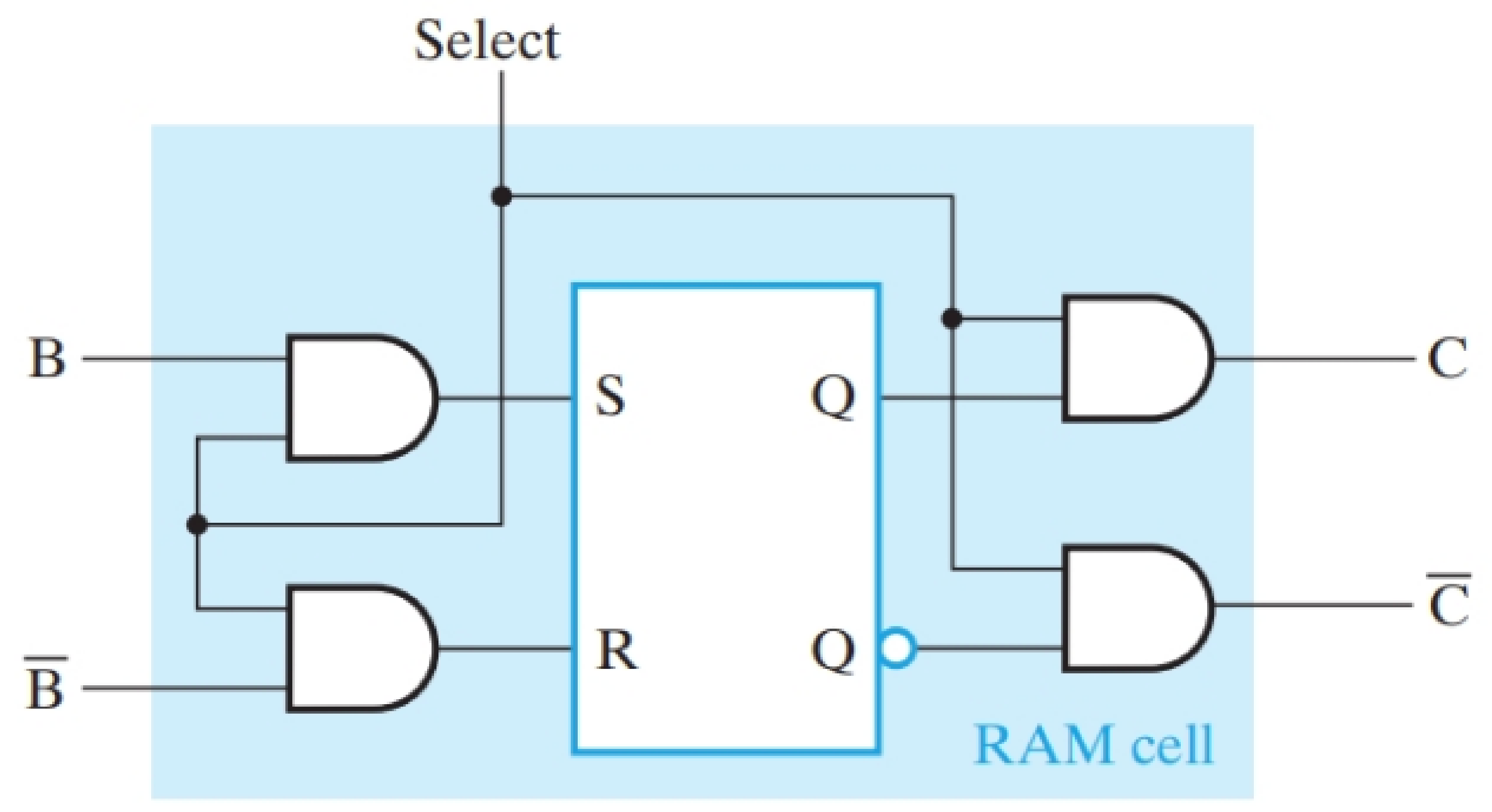
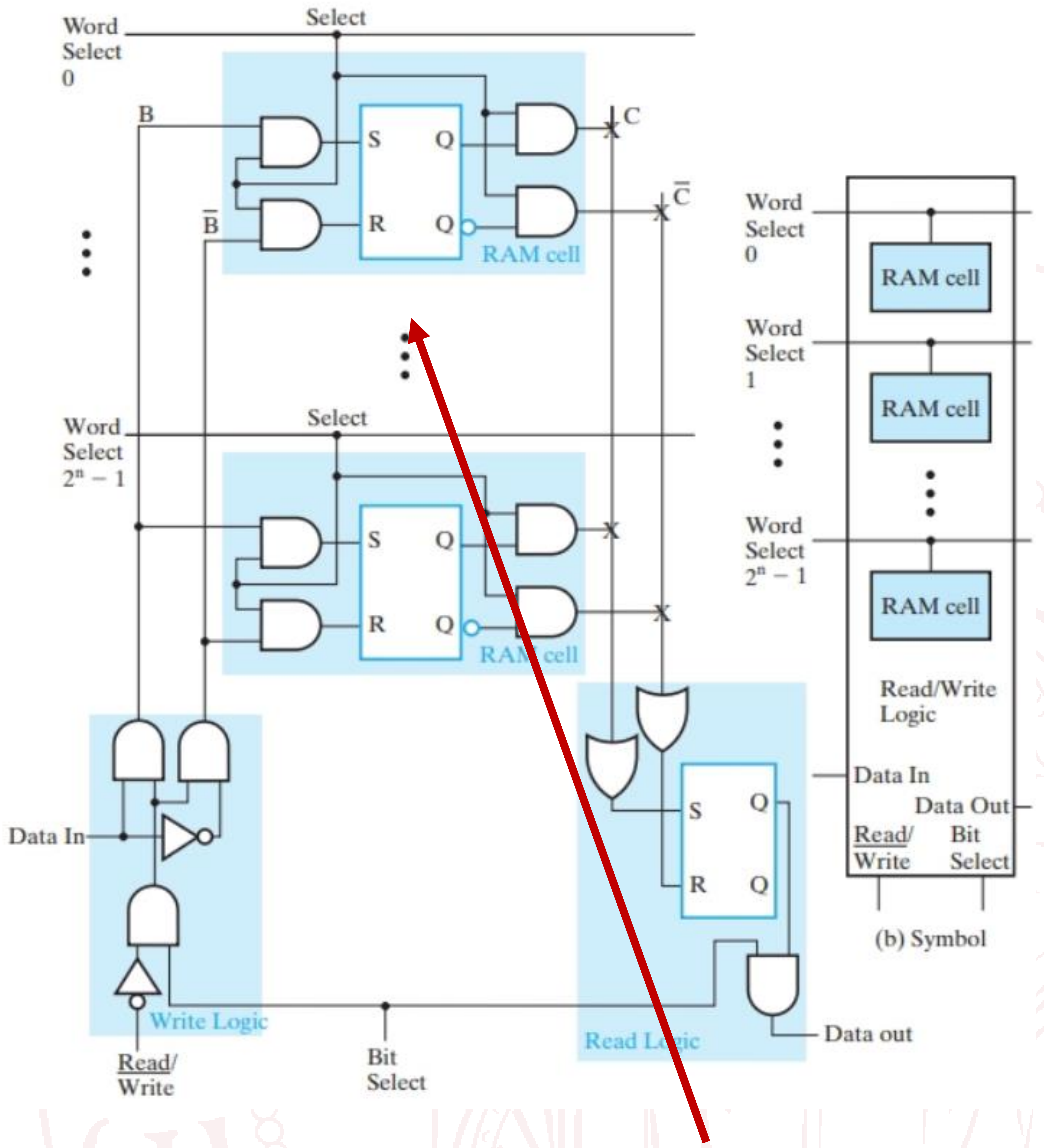


FIGURE 7-4
Static RAM Cell

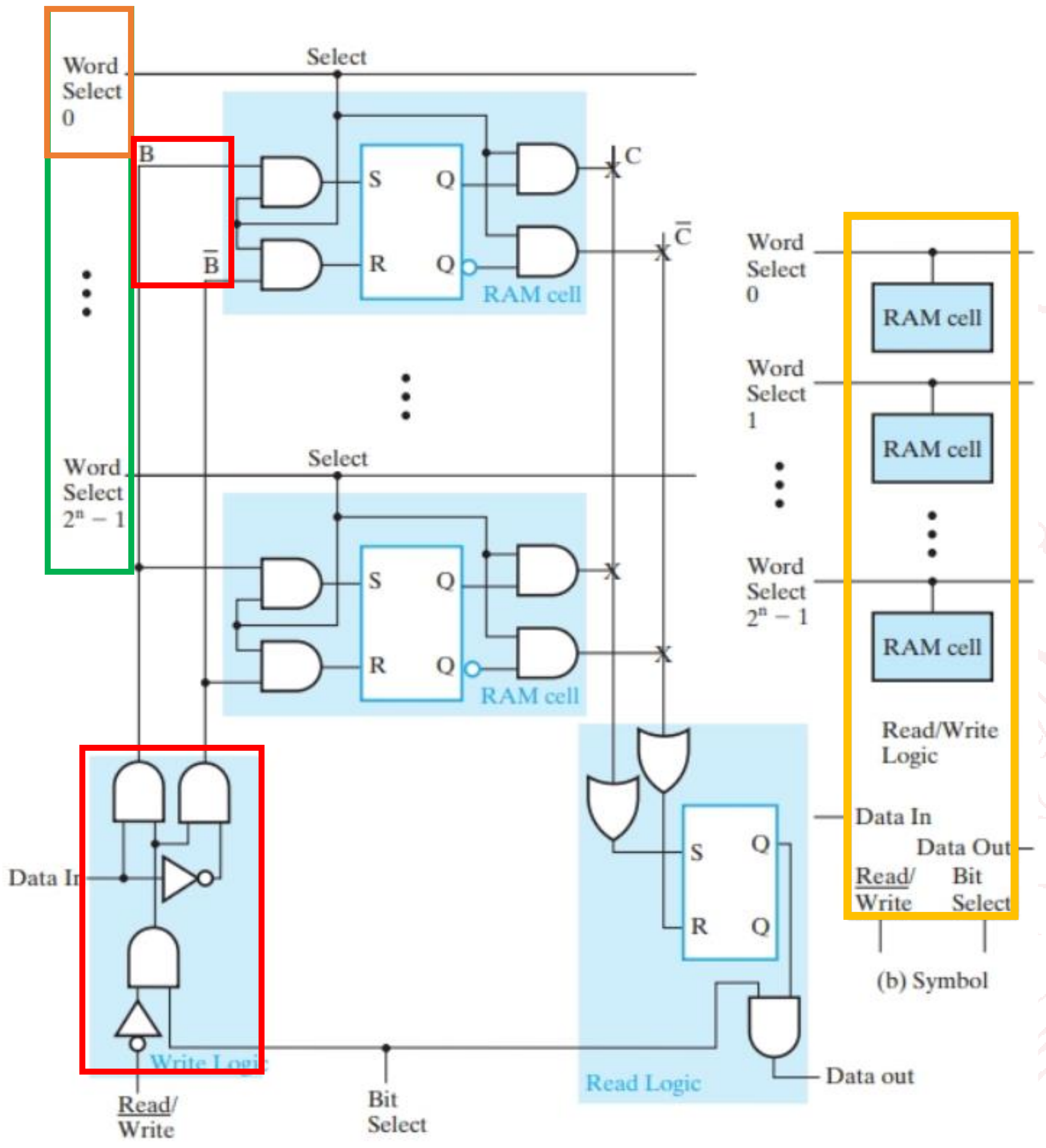
- If $Select = 1$, the stored content is determined by the values $B, \overline{B}$
- If $Select\ =\ 0$, both $C, \overline{C} = 0$
- $Select$ enables/disables the access to the cell
- RAM bit slice → contains all of the circuitry associated with a single bit position of a set of RAM words



k^th bit of all the words

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* ... $2^n - 1$
- 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

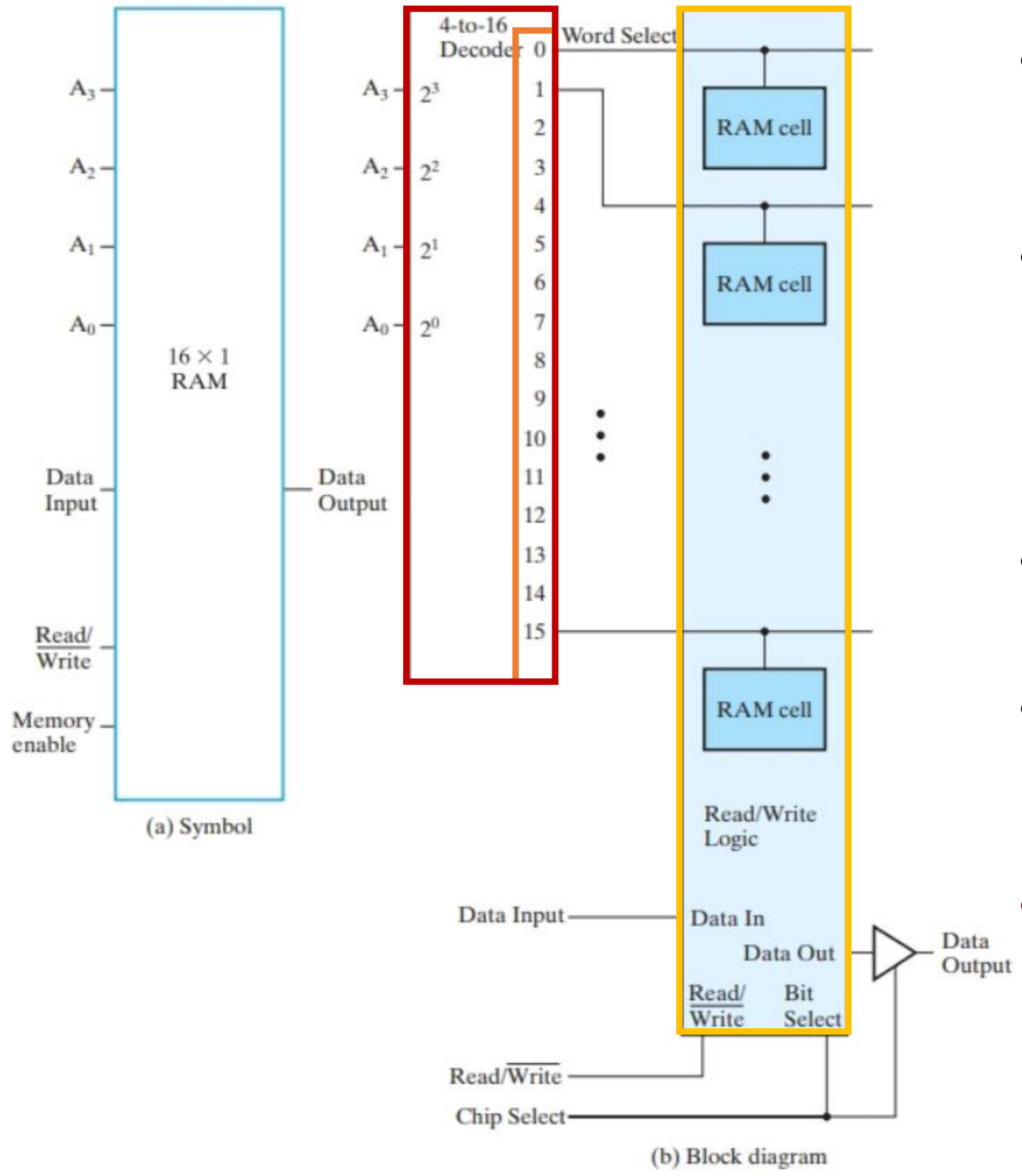


FIGURE 7-6
16-Word by 1-Bit RAM Chip

- 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

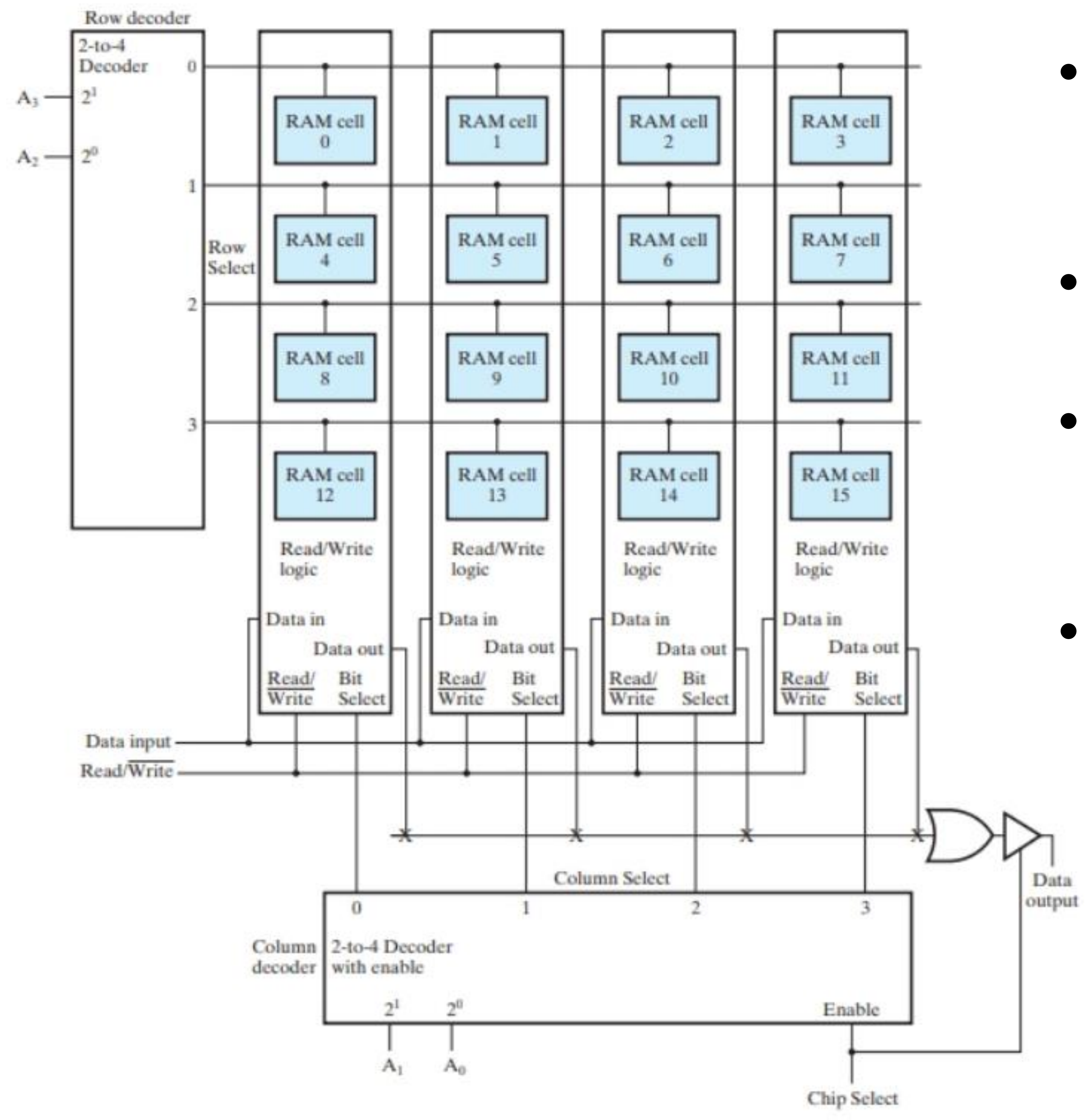


FIGURE 7-7
Diagram of a 16 × 1 RAM Using a 4 × 4 RAM Cell Array

- 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

$$\underbrace{\boxed{A_3}\boxed{A_2}}_{\text{Row}}\overbrace{\boxed{A_1}\boxed{A_0}}^{\text{Column}}$$

8x2 SRAM chip using 4x4 RAM Cells Array

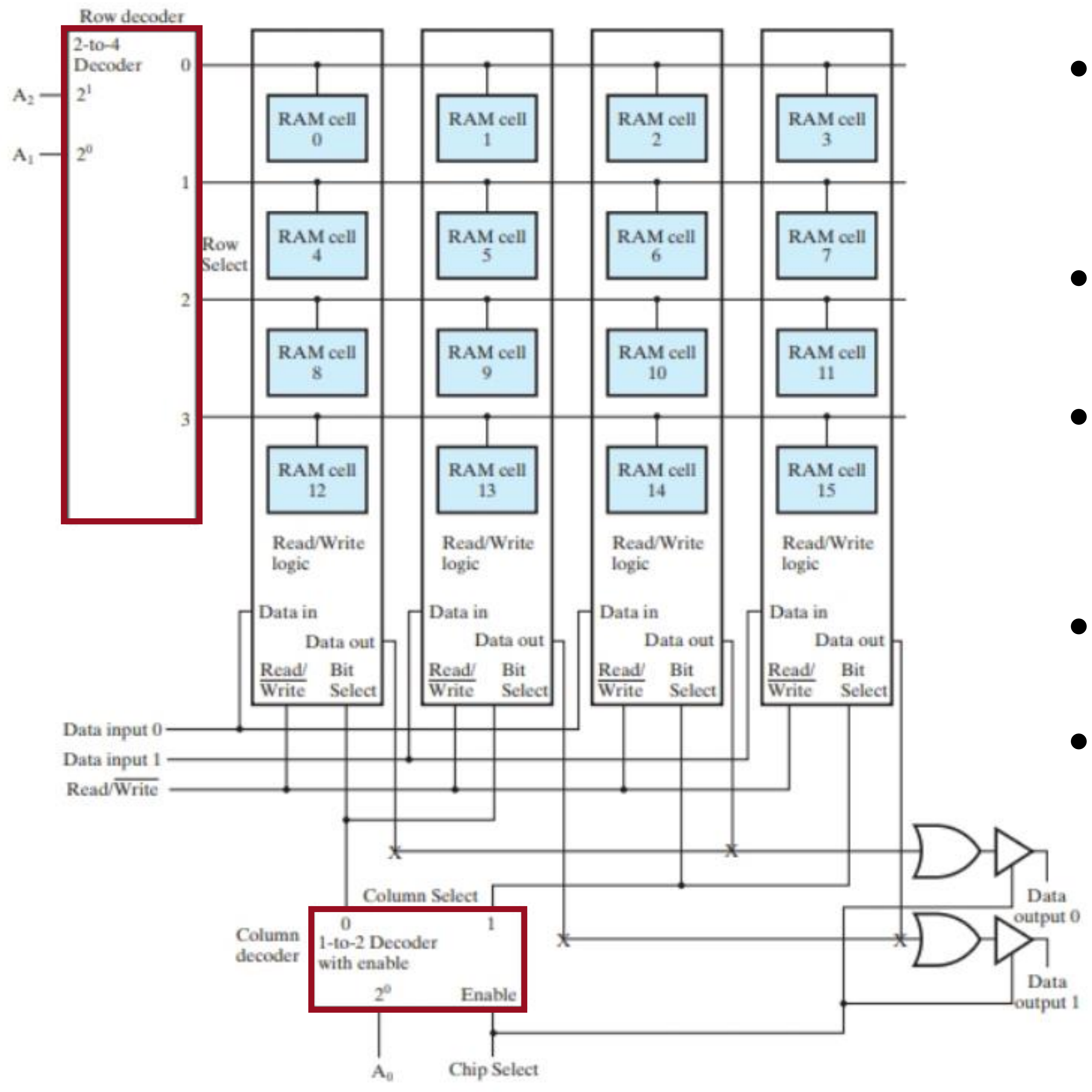


FIGURE 7-8
Block Diagram of an 8 × 2 RAM Using a 4 × 4 RAM Cell Array

- 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

$$\underbrace{A_2 \quad A_1}_{\text{Row}} \quad \overbrace{A_0}^{\text{Column}}$$

8x2 SRAM chip using 4x4 RAM Cells Array

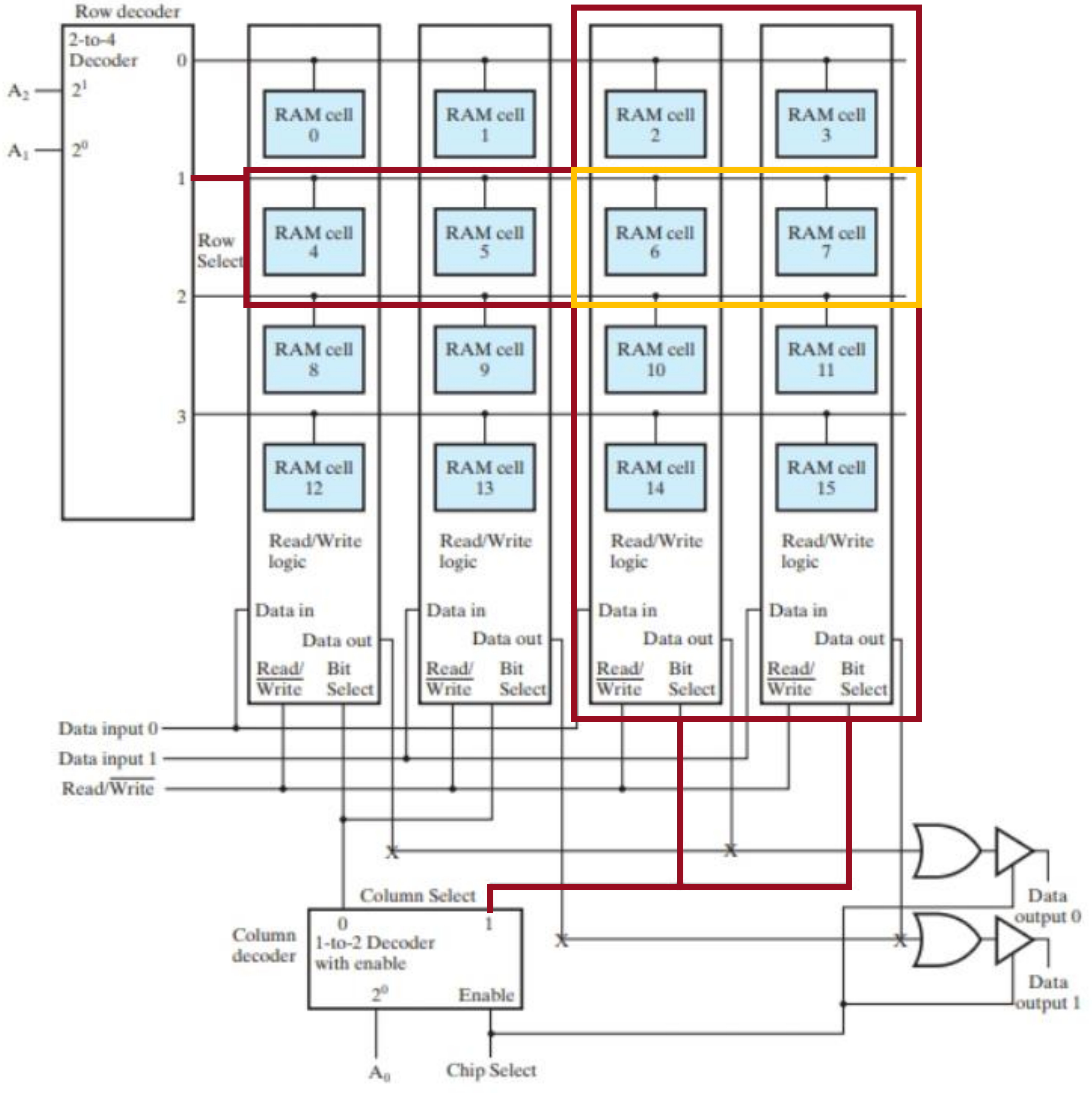


FIGURE 7-8
Block Diagram of an 8 × 2 RAM Using a 4 × 4 RAM Cell Array

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:
  - $nrows\ ncolumns = \sqrt{256K} = \sqrt{(256 \cdot 1024)} = 512 = 2^{\boxed{9}}$

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

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

Disclaimer

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

© 2016 Pearson Education, Ltd

Questions