



Computer Organization and Software Systems

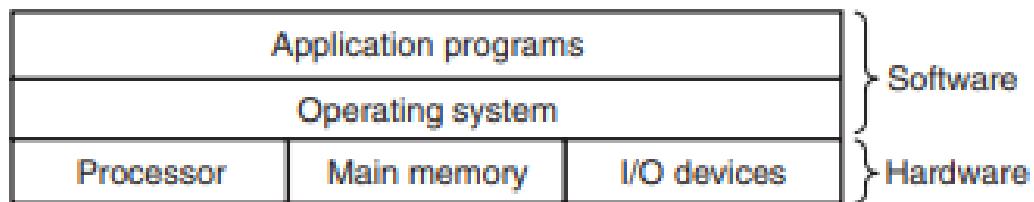
Contact Session 2

Dr. Lucy J. Gudino



Operating System

- collection of software/ Program that acts as an intermediary between an user of a computer and the computer hardware.
- is a program that helps to run all the other programs
- Three main functions:
 - Resource management
 - Establish an user interface
 - Execute and provide services for application software



Layered view of a computer system.



```

main()
{
    int x;
    scanf("./d", &x);
    printf("x = ./d", x);
}
  
```



Main objectives

- Convenience
- Efficiency
- Ability to evolve and offer new services
- Maximize System performance
- Protection and access control
- Footprint of OS should be small



Important Note

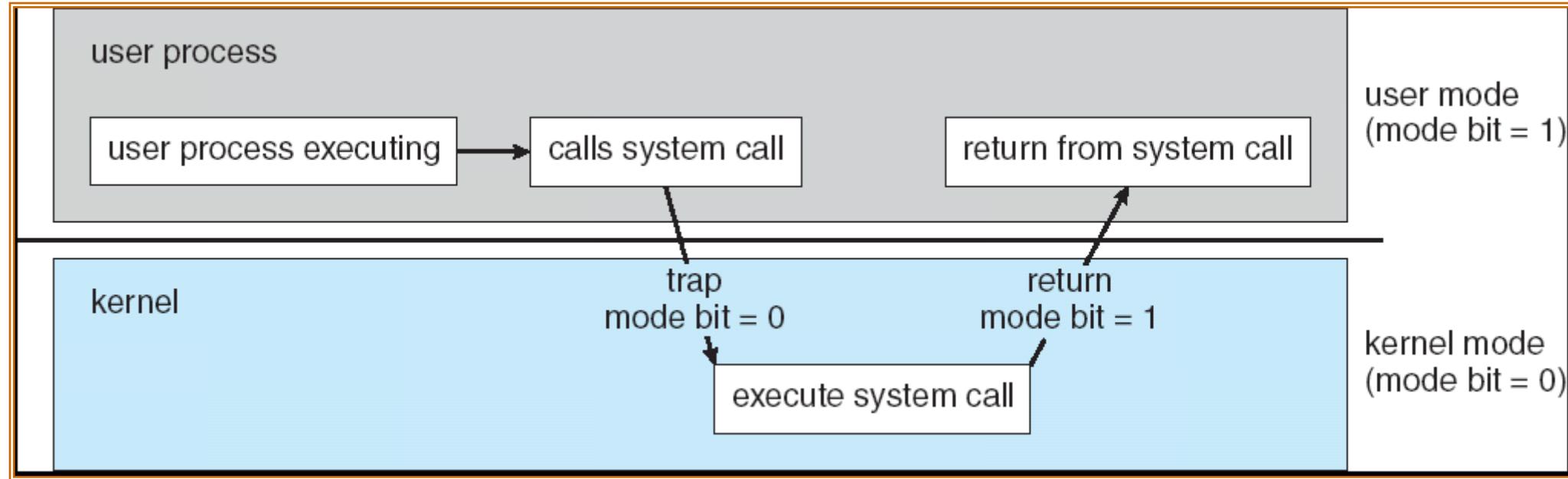
- **bootstrap program** is loaded at power-up or reboot
 - Typically stored in ROM or EPROM, generally known as **firmware**
 - Initializes all aspects of system
 - Loads operating system kernel and starts execution
- “The one program running at all times on the computer” is the **kernel**. Everything else is either a system program (ships with the operating system) or an application program



Operating System Operations

- Dual-mode operation
 - User mode
 - Kernel mode (also known as System Mode / Supervisor mode/ privileged mode)
- User mode(1):
 - user program executes in user mode
 - certain areas of memory are protected from user access
 - certain privileged instructions may not be executed
- Kernel Mode (0)
 - privileged instructions may be executed
 - protected areas of memory may be accessed

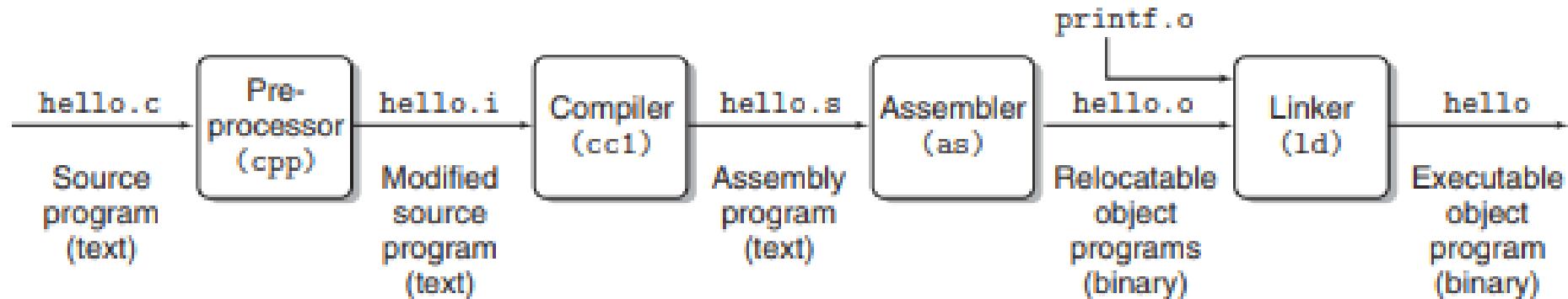
Transition from user to kernel mode



Running a Hello.c Program

```
#include <stdio.h>

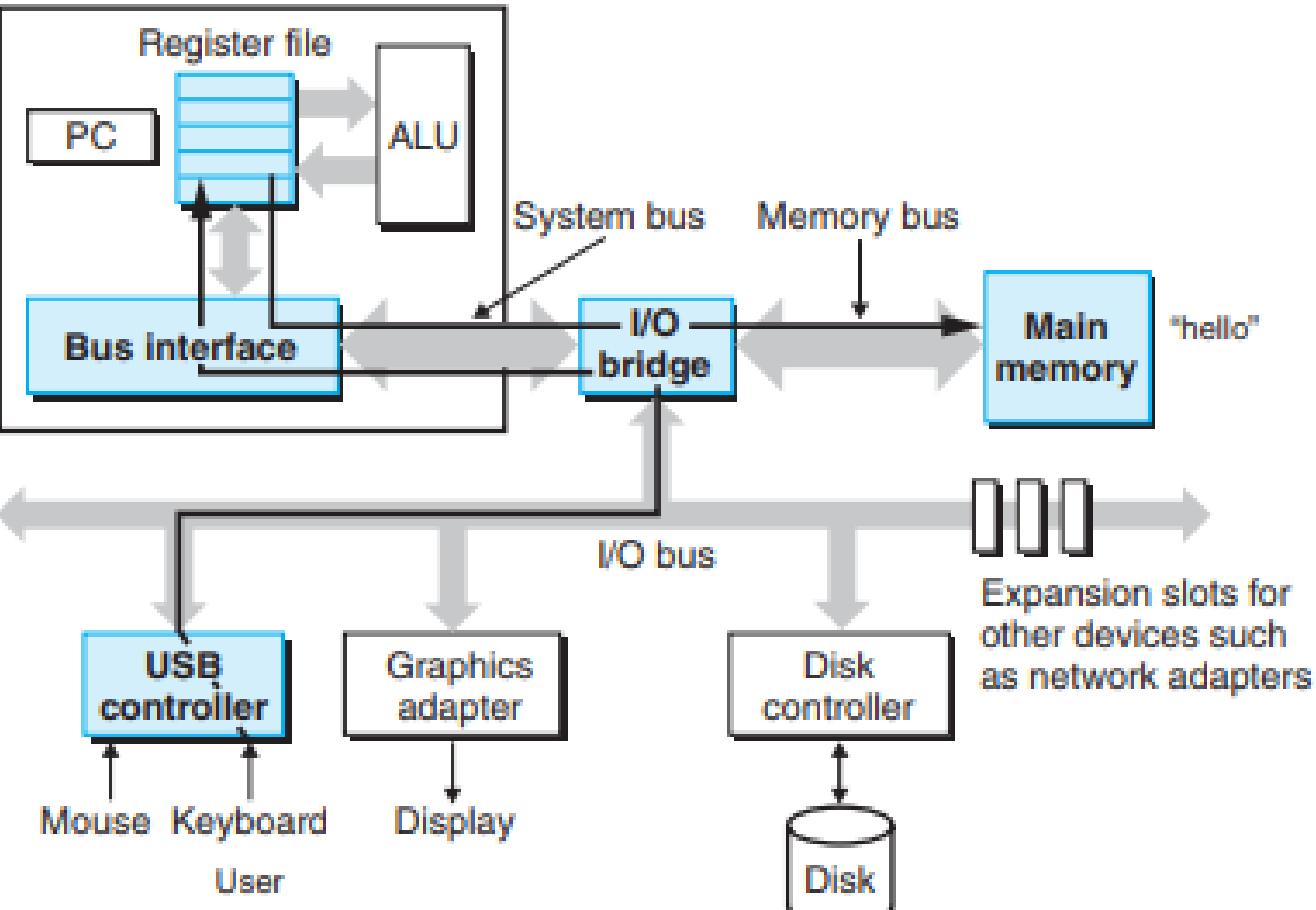
int main()
{
    printf("hello, world\n");
}
```



The compilation system.

Reading ./hello command from Keyboard

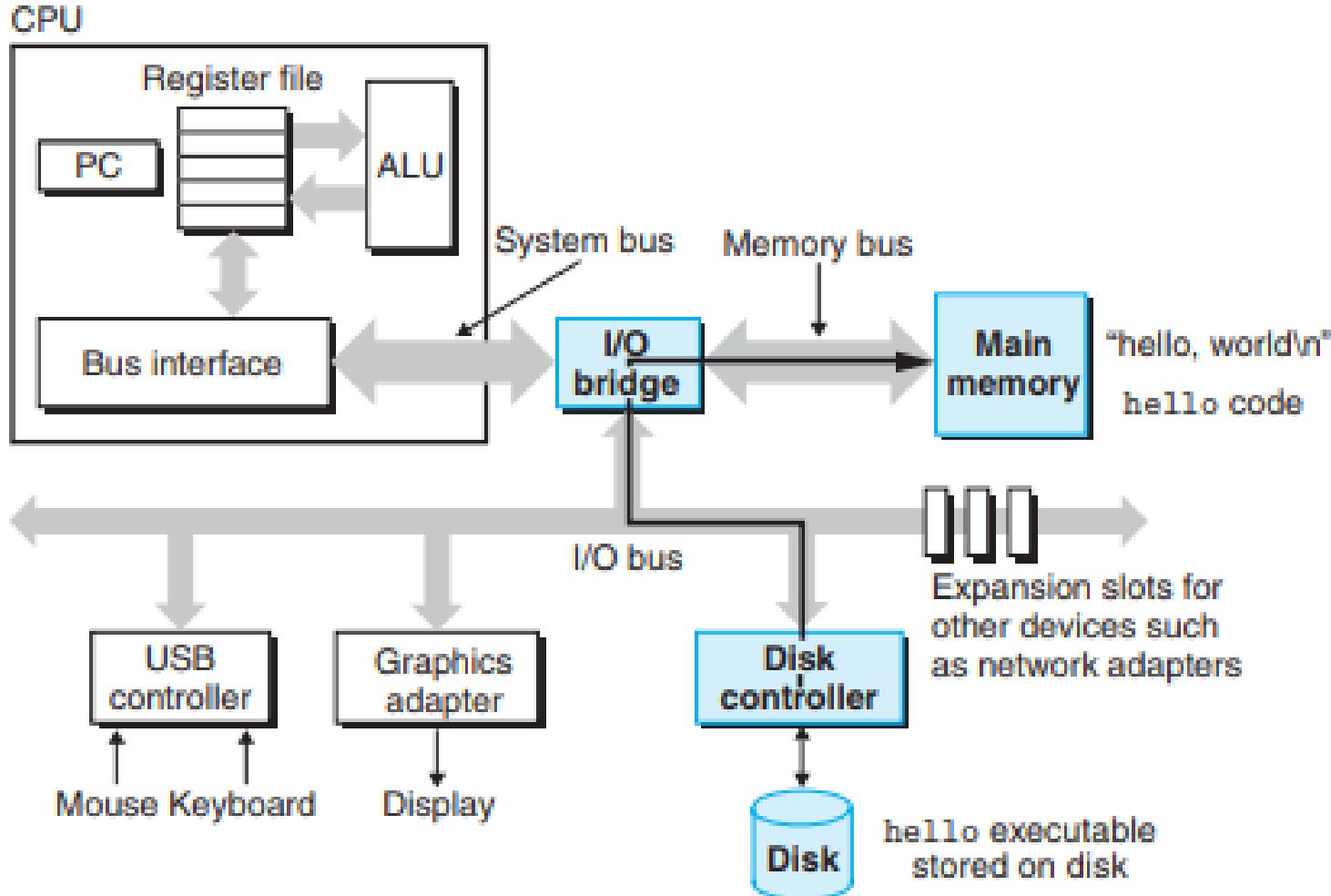
CPU



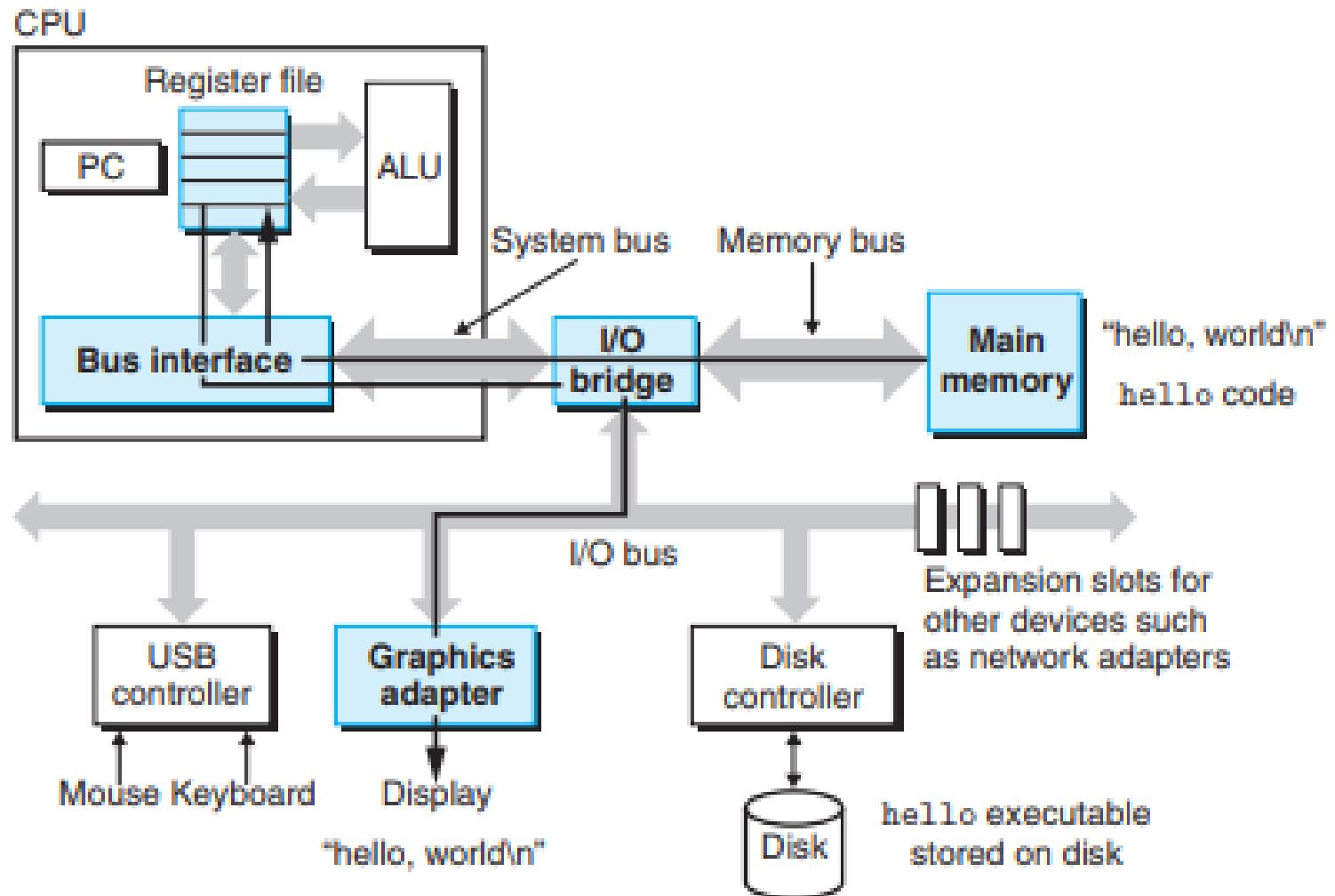
```
#include <stdio.h>
```

```
int main()
{
    printf("Hello, world\n");
}
```

Loading the executable from disk into main memory



Writing the output string from memory to the display





Why do we need to know how compilation works?

- Optimizing program performance.
- Understanding link-time errors
- Avoiding security holes.

Today's Class



Contact Hour	List of Topic Title	Text/Ref Book/external resource
3-4	Memory Organization - Internal Memory - External Memory	T1, R2

Memory Organization



Internal Memory Organization

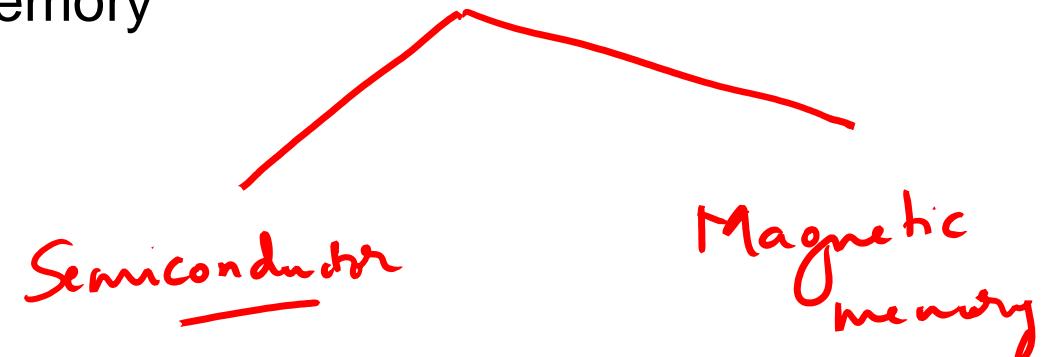


- Internal Memory

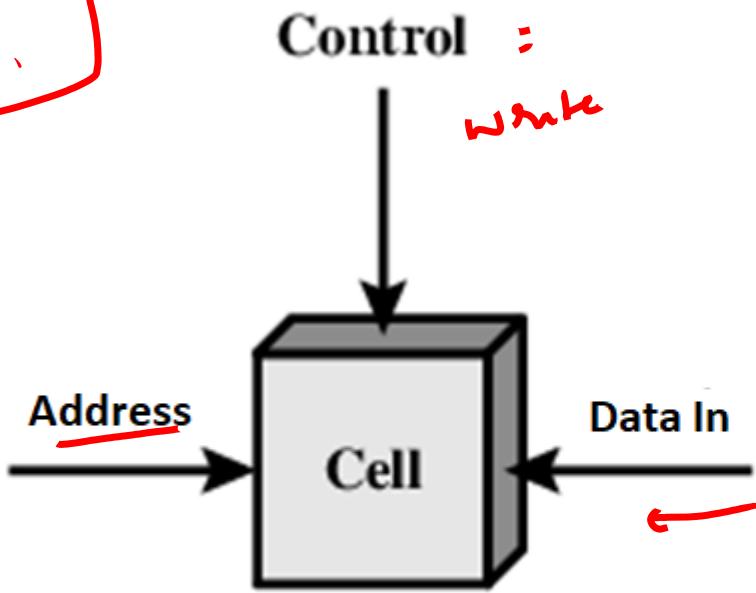
- Also known as main memory or Primary Memory
- Small data storage but quick access.
- Examples : RAM, ROM

- External Memory

- Also Known as secondary Memory
- Huge data stored persistently
- Examples: hard disk, solid state drives, USB flash drives etc.

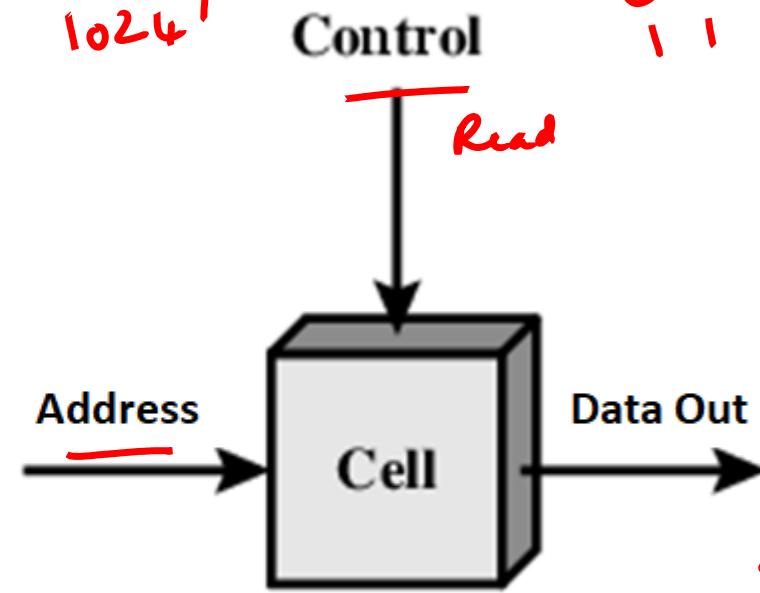


Semiconductor Memory



(a) Write

1000
address space
 $1K \times 8\text{ bits}$
1024
 $1000 \times$

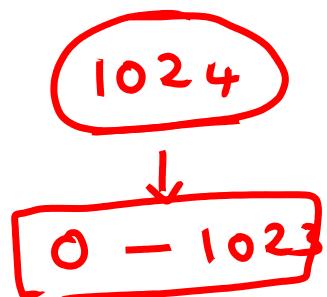


(b) Read

Two Stable
E ON \Rightarrow True - 1
OFF \Rightarrow False - 0

m - Location

n =
0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1

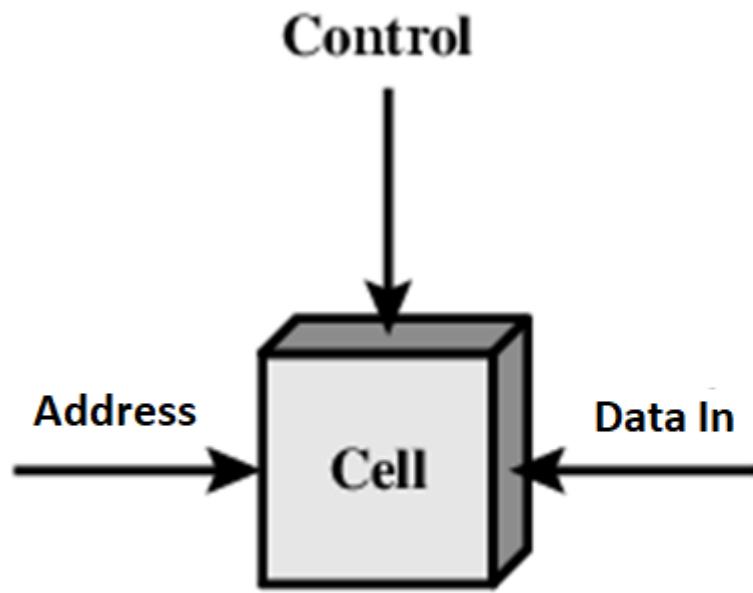


2^x = address space

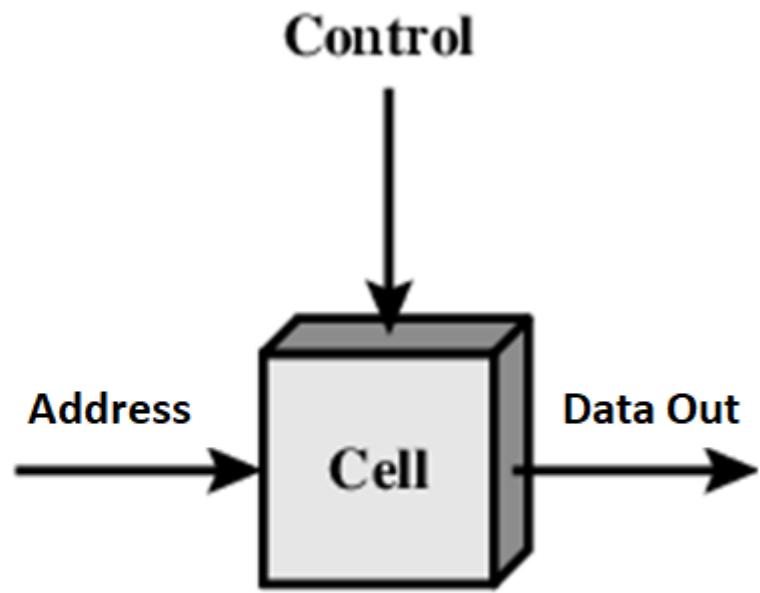
$$2^x = 1024$$

$x = 10$

Semiconductor Memory



(a) Write



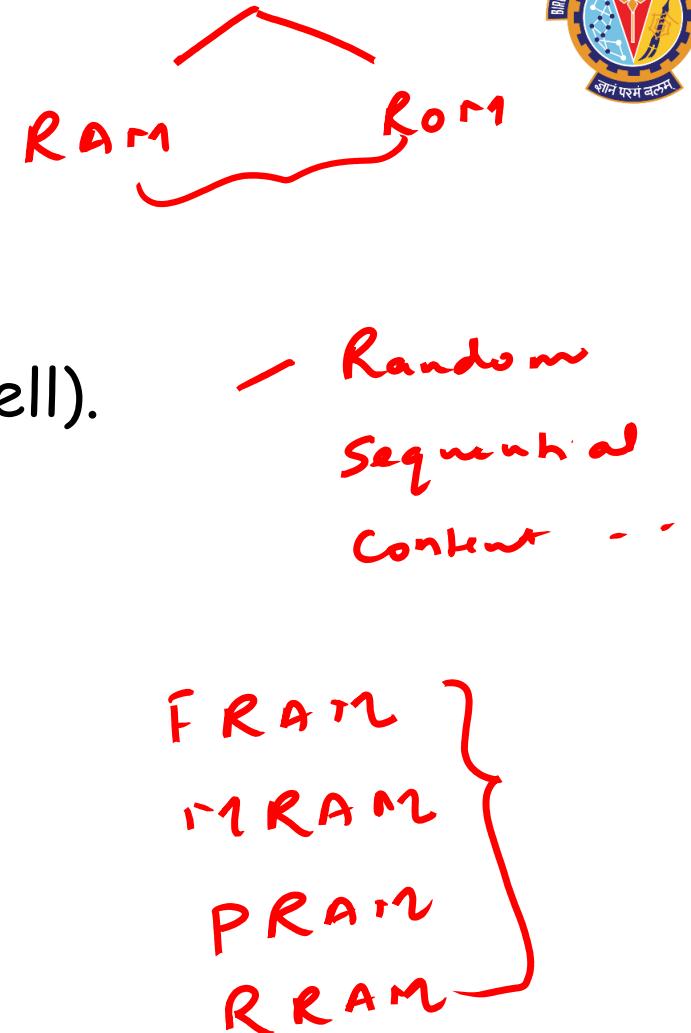
(b) Read

Random-Access Memory (RAM)



- Key features

- RAM is traditionally packaged as a chip.
- Basic storage unit is normally a **cell** (one bit per cell).
- Multiple RAM chips form a memory.
- RAM comes in two varieties:
 - SRAM (Static RAM)
 - DRAM (Dynamic RAM)
- SRAM and DRAM are volatile memories
 - Loose information if powered off.



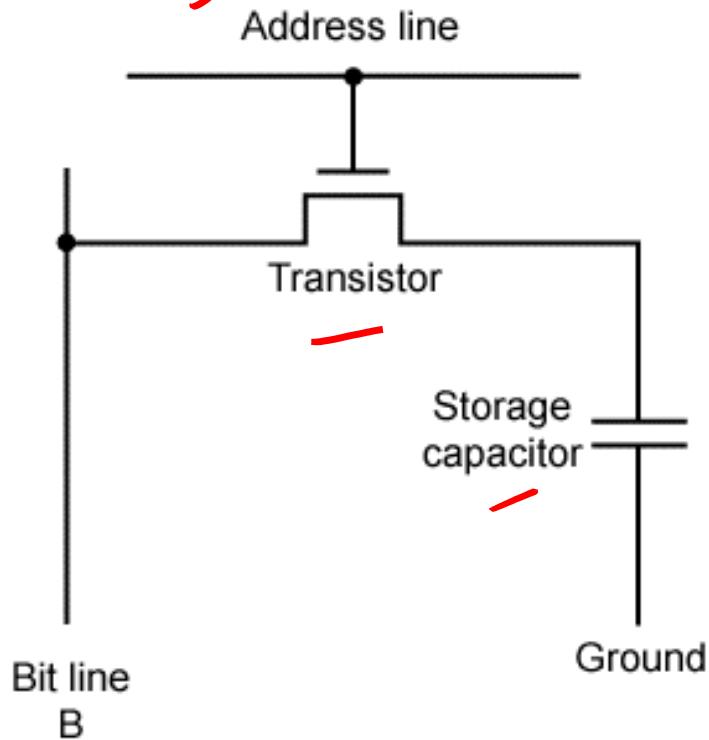
SN 74 LS207

↓ 1024 × 1 bit

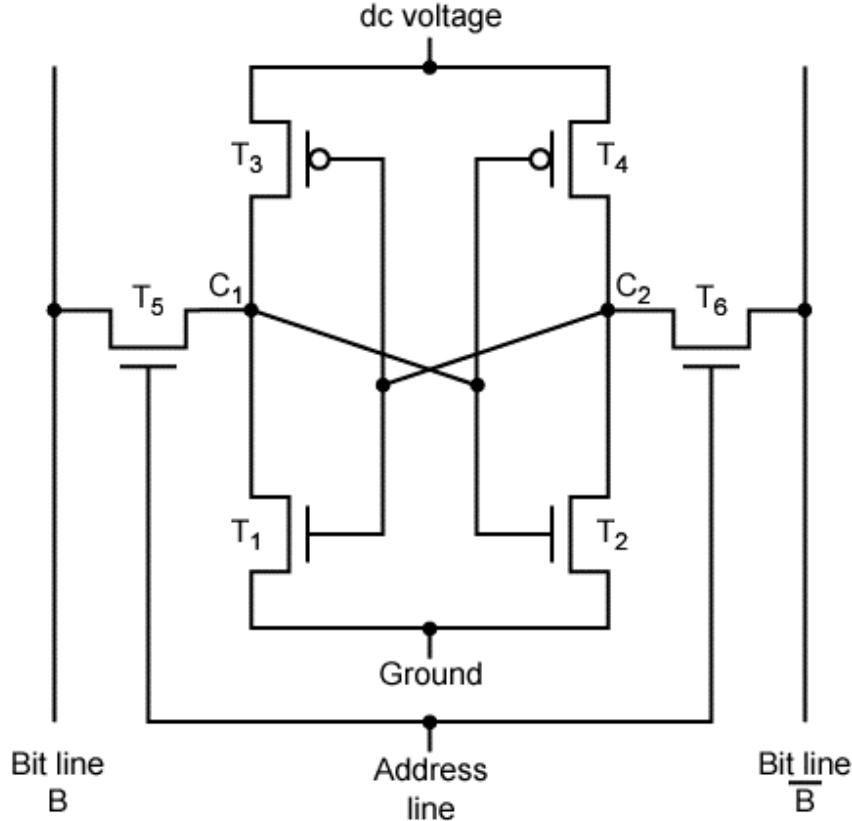
↓ 1024 × 8 bit

DRAM vs SRAM

DRAM



S-Ram



Flip-flop

	Trans. per bit	Access time	Needs refresh?	Needs EDC?	ECC	Cost	Applications
SRAM	4 to 6	1X	No	Maybe		100x	Cache ✓
DRAM	1	10X	Yes	Yes		1X	Main memories, frame buffers

Read Only Memory

- Permanent Storage and Nonvolatile Memories
- Read Only Memory Variants:
 - Read-only memory (**ROM**): programmed during production
 - ✓ Programmable ROM (**PROM**): can be programmed once
 - ✓ Erasable PROM (**EPROM**): can be bulk erased (UV, X-Ray)
 - ✓ Electrically erasable PROM (**EEPROM**): electronic erase capability
 - Flash memory: EEPROMs. with partial (block-level) erase capability
 - Wears out after about 100,000 erasing
- Firmware

PROM
programm

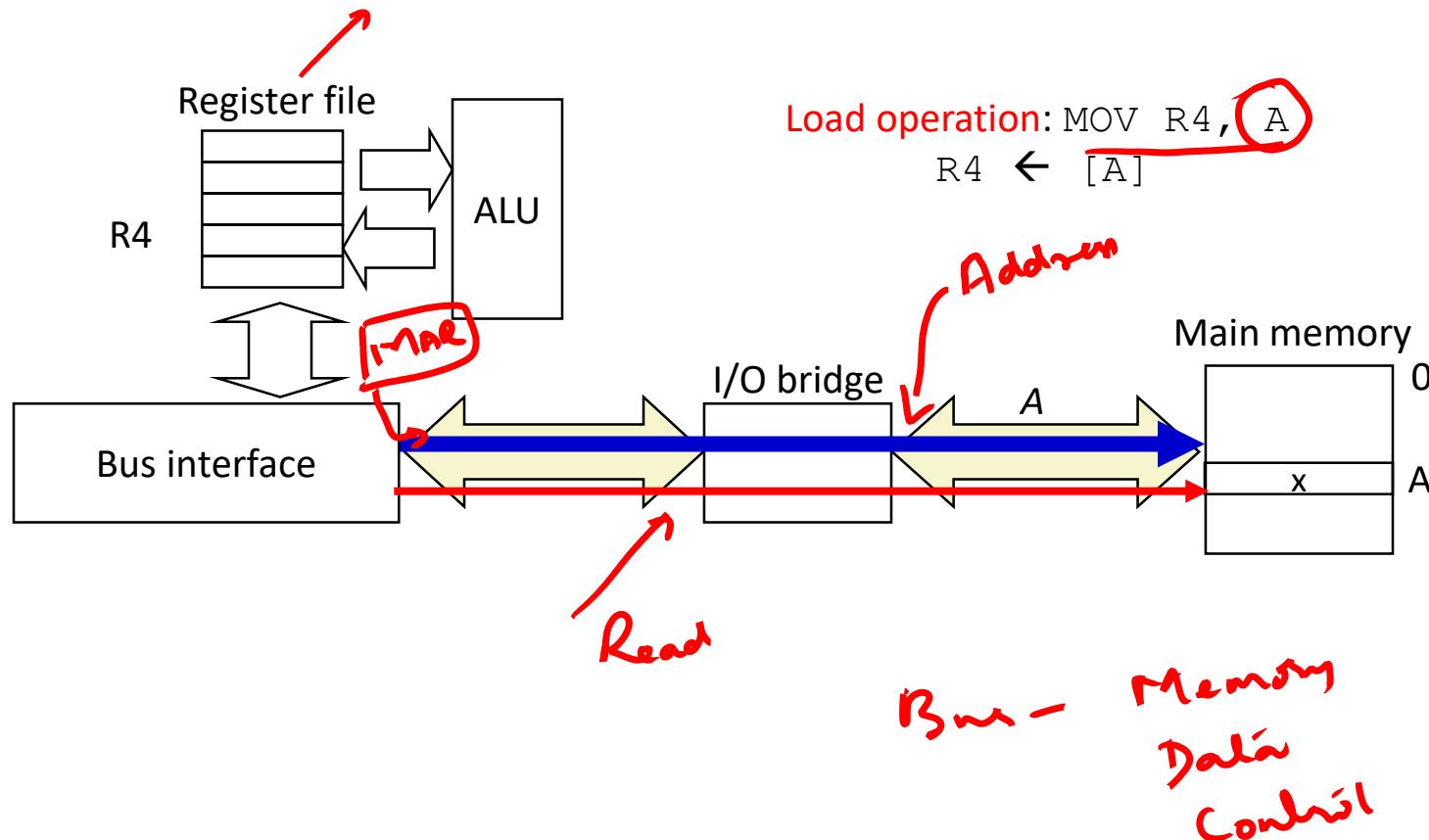
Applications



- Storing fonts for printers
- Storing sound data in musical instruments
- Video game consoles
- Implantable Medical devices.
- High definition Multimedia Interfaces(HDMI)
- BIOS chip in computer
- Program storage chip in modem, video card and many electronic gadgets, controllers for disks, network cards,

Memory Read Operation (1)

- CPU places address A and then read control signal on the memory bus



$R4 \leftarrow [A]$

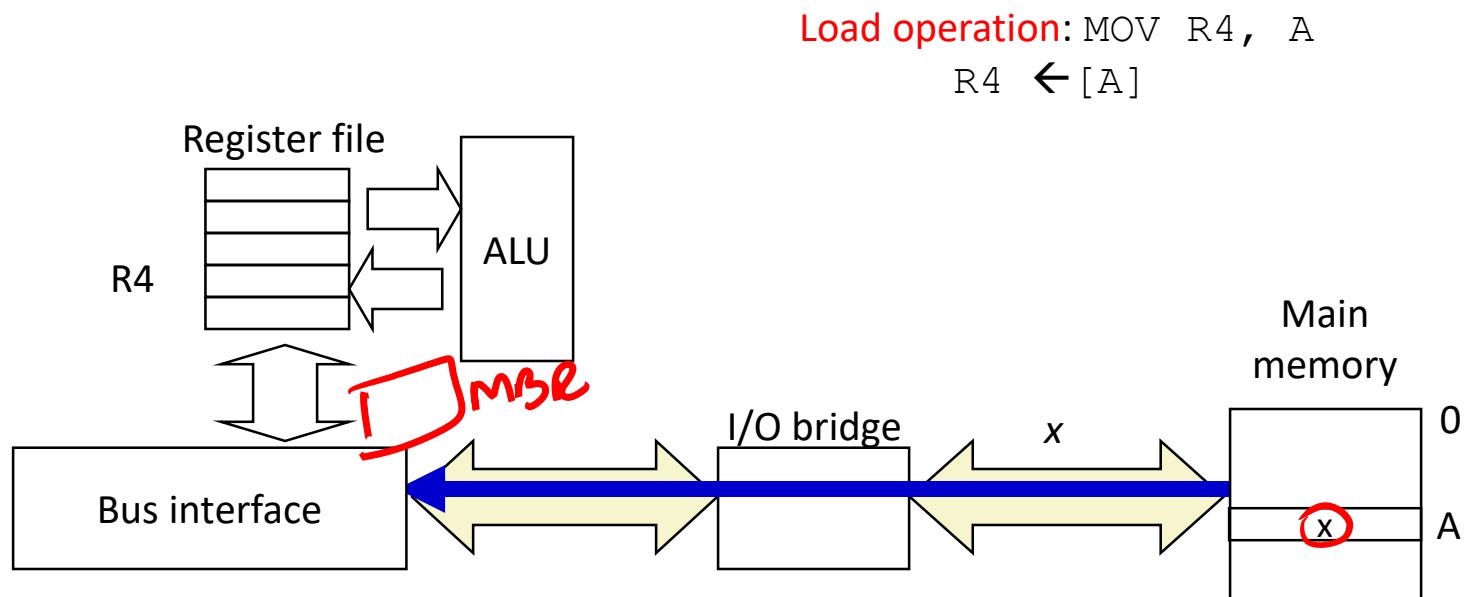
— MBR / MDR

— MAR

Address

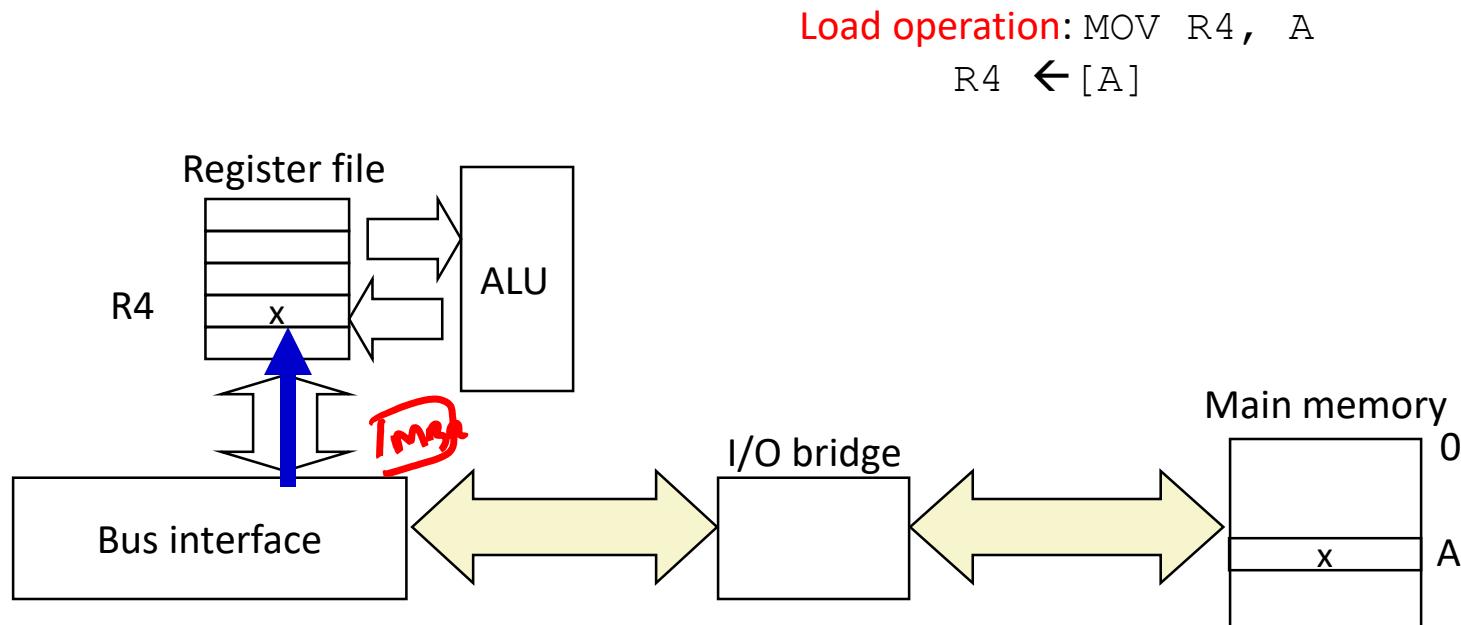
Memory Read Operation (2)

- Main memory reads A from the memory bus, retrieves word x , and places it on the bus



Memory Read Operation (3)

- CPU read word x from the bus and copies it into register R4.

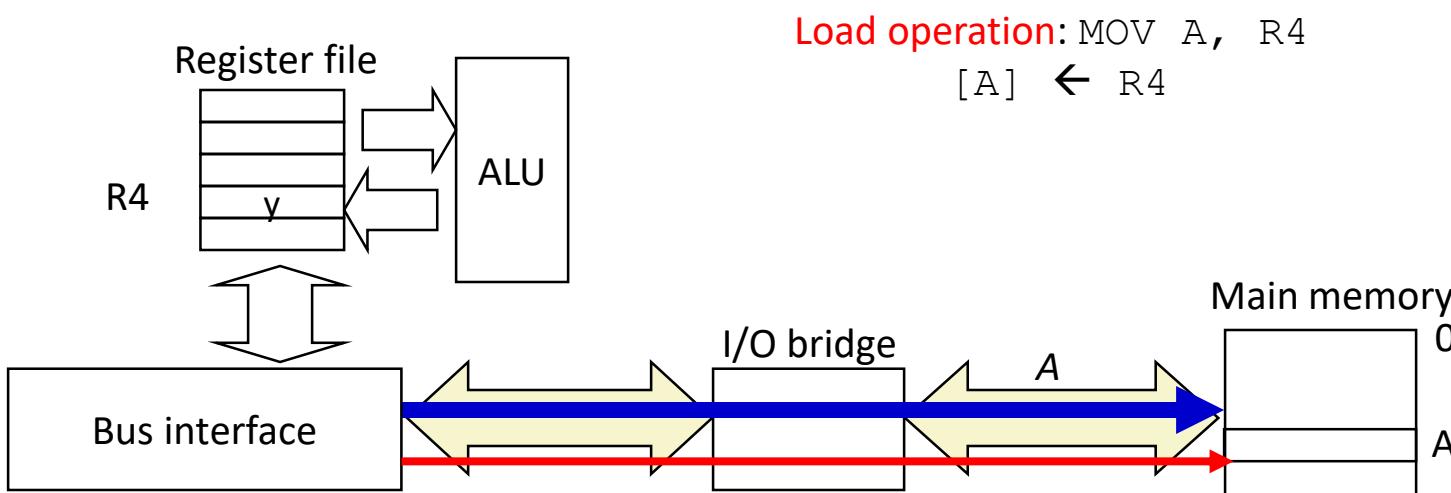


- ① $A \rightarrow \text{MAR} \rightarrow \text{Address bus}$
- ② Read - Control bus
- ③ Memory reads data from the memory location \Rightarrow ~~Block~~ bus
- ④ Data \Rightarrow MBR
- ⑤ MBR $\Rightarrow R4$

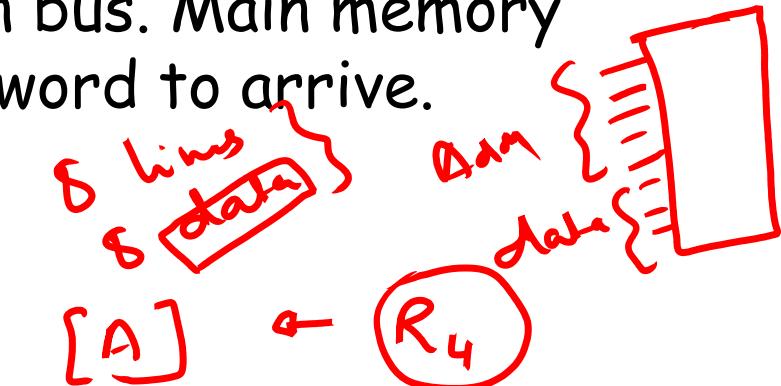
Memory Write Operation (1)



- CPU places address A and WRITE control signal on bus. Main memory reads them and waits for the corresponding data word to arrive.



16 bits
8 bits
8 bits
8 bits
24 bits

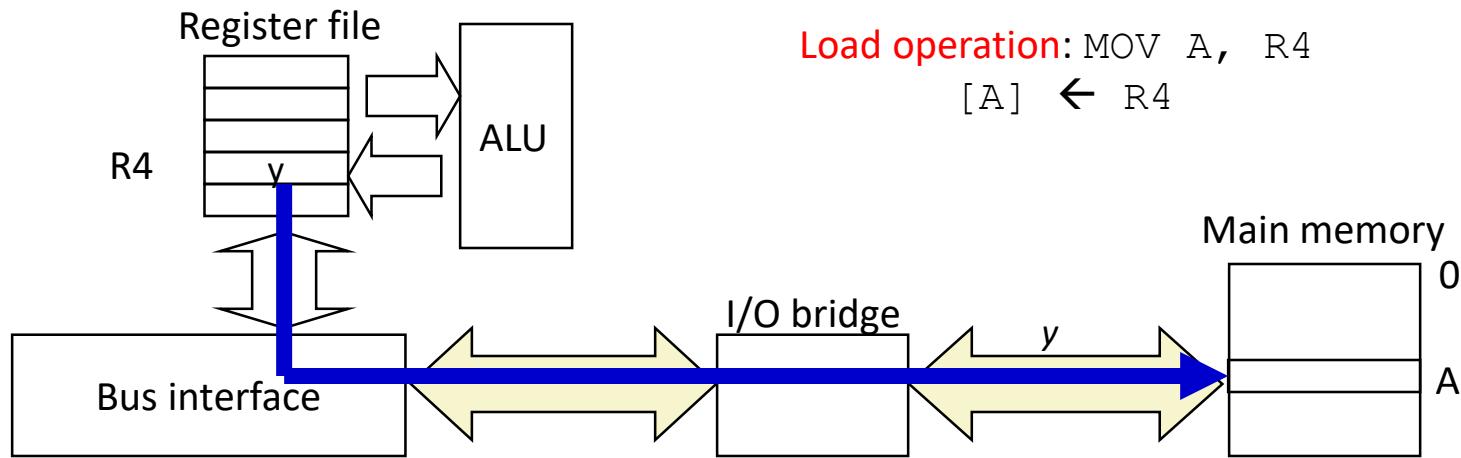


$[A] \leftarrow R_4$

- ① Address \Rightarrow MAR \Rightarrow Address Bus
- ② Write Signal \Rightarrow Control bus
- ③ $R_4 \Rightarrow$ MBR
- ④ MBR \Rightarrow Data bus
- ⑤ Data bus \Rightarrow Memory

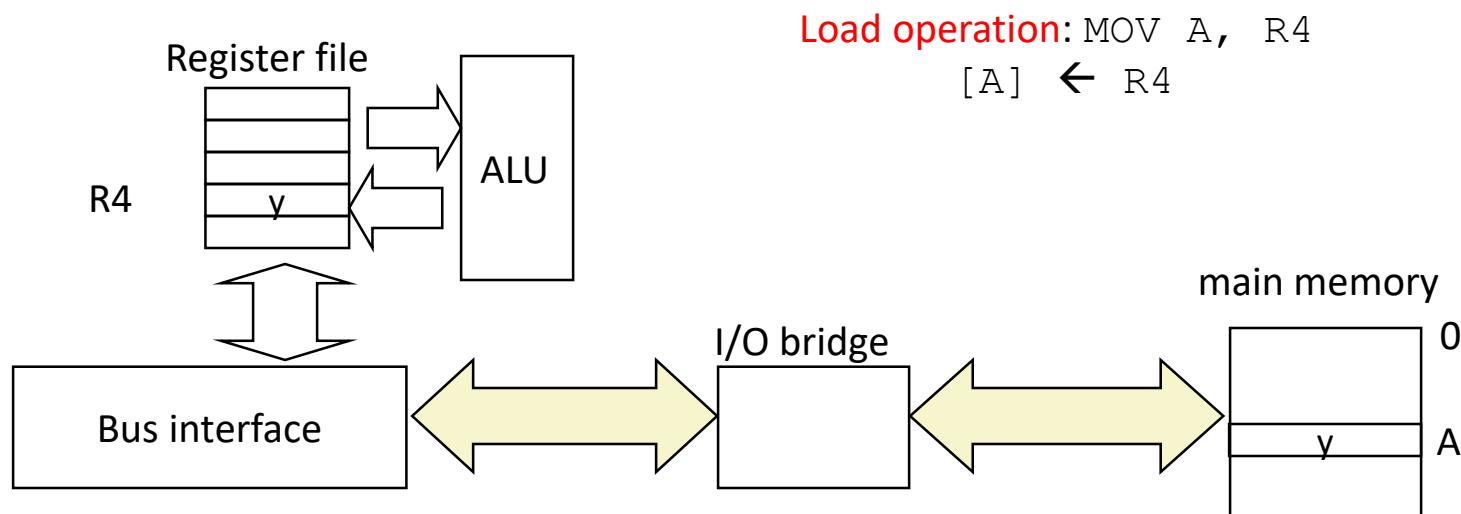
Memory Write Operation (2)

- CPU places data word y on the bus



Memory Write Operation (3)

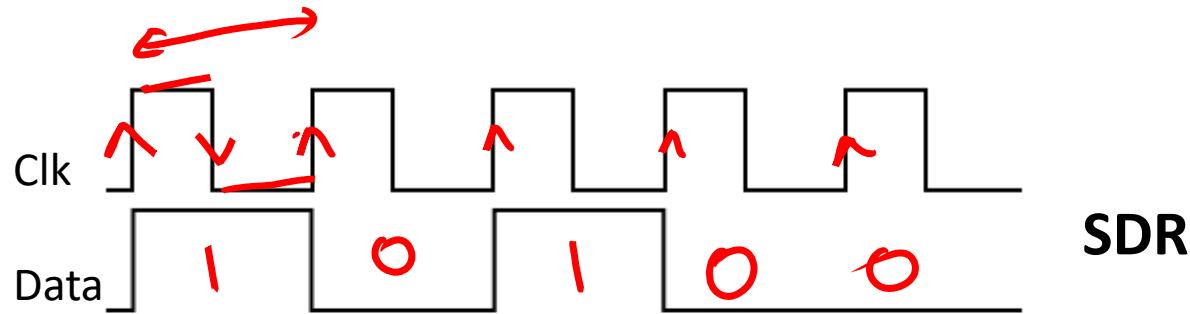
- Main memory reads data word y from the bus and stores it at address A .



SDR Vs DDR

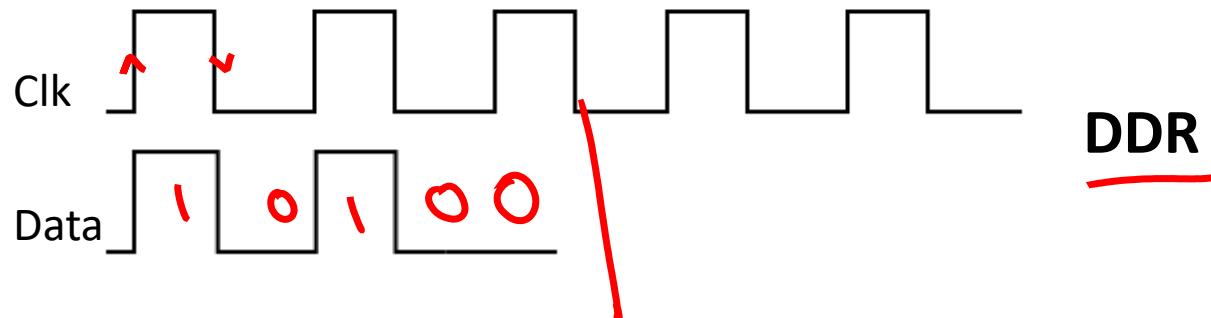


- SDR → Single Data Rate ✓
- DDR → Double Data Rate ✓



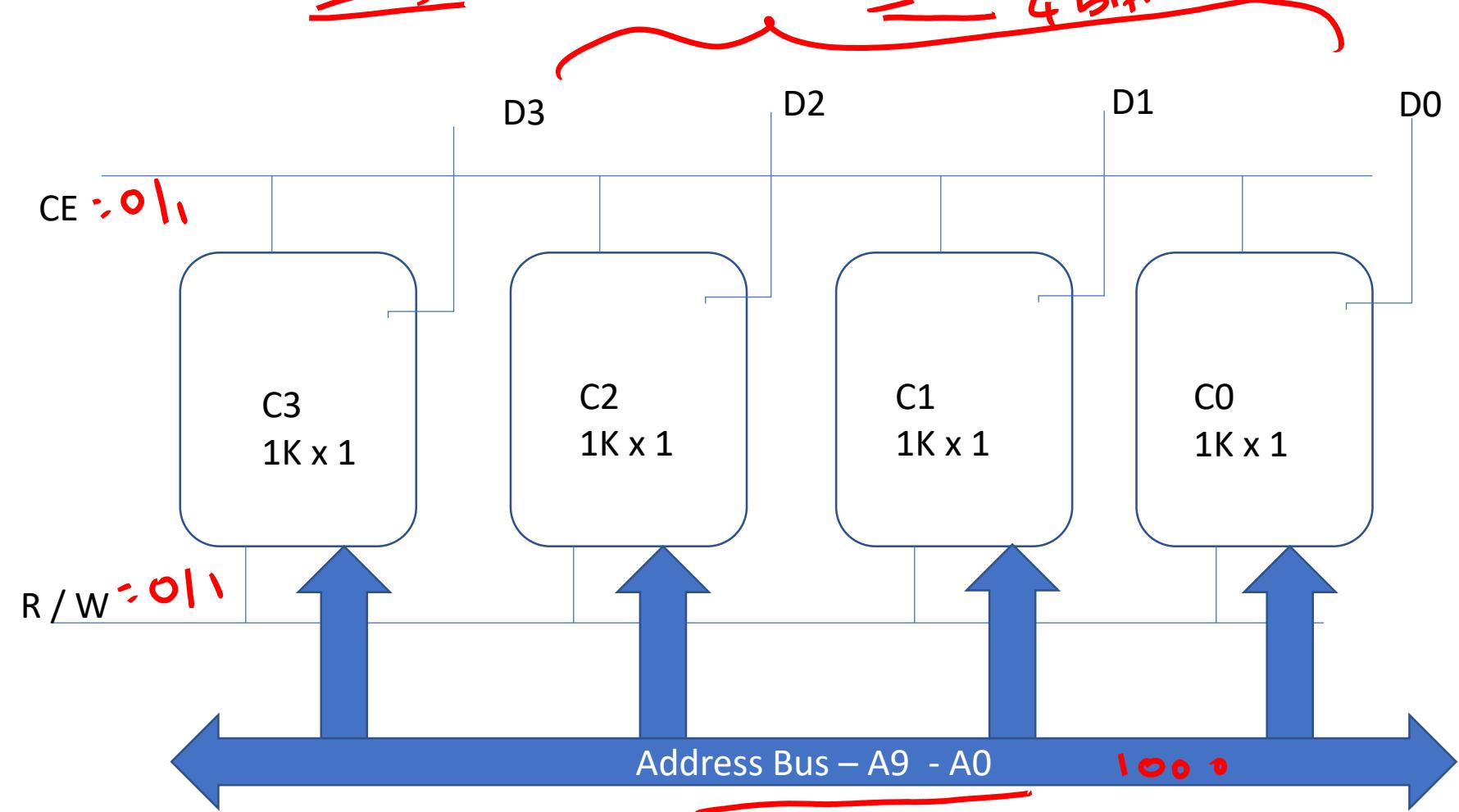
SDRAM
Synchronous
time
SDR SDRAM - 200
DDR SDRAM - 200

triggering



Typical Memory Connection Examples

- Construct $1K \times 4$ bit memory using $1K \times 1$ bit chip



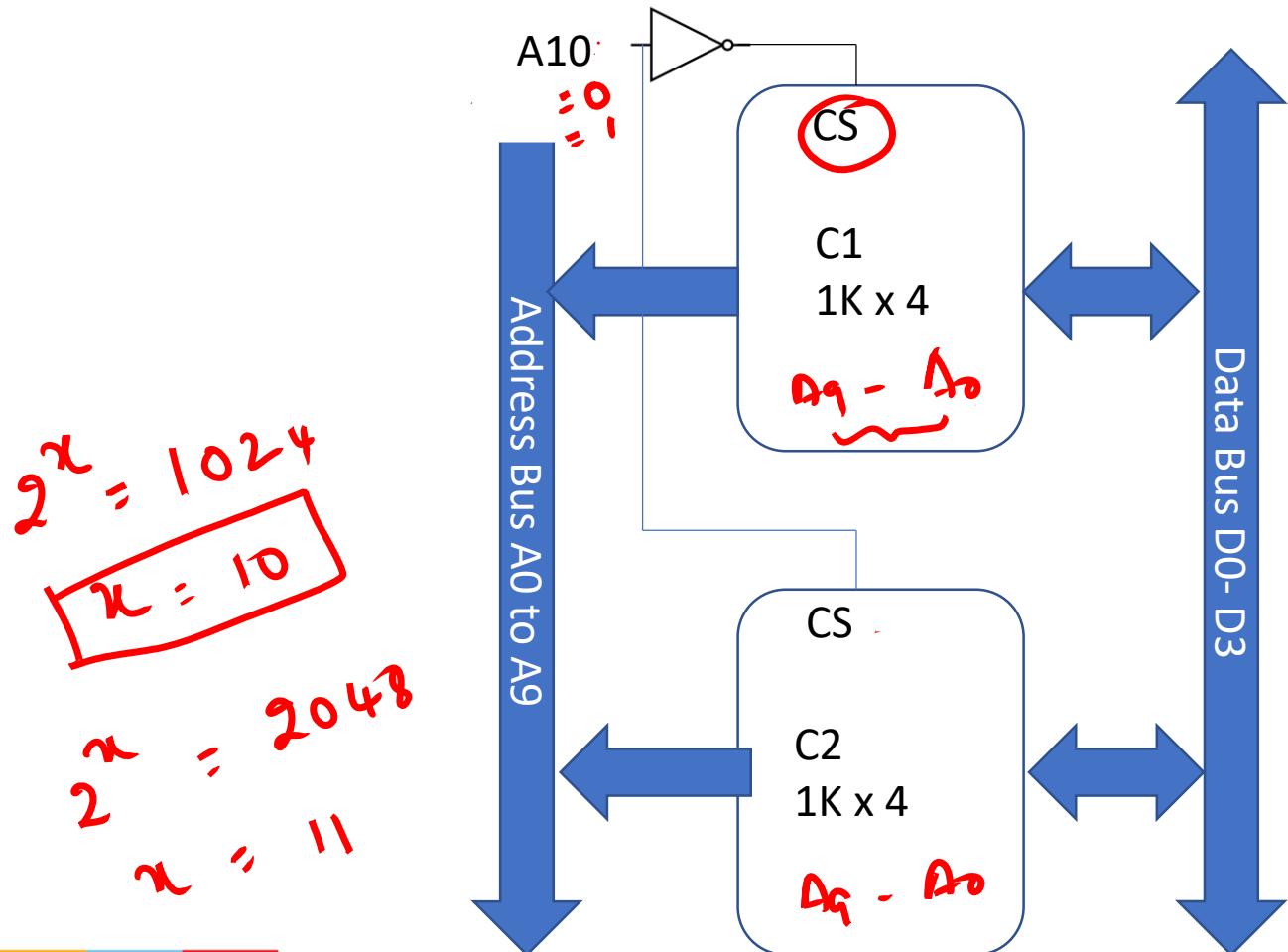
- ① How many
- ② How many data 4
- ③ How many lines

$1K \times 4$ bit
 address space
 1×1024
 \Downarrow
 1024
 $2^{\underline{x}} = 1024$
 $x = 10$ lines
 $A_9 \quad A_0$

Typical Memory Connection Examples



- Construct 2K X4 bit memory using ~~1Kx4~~ bit chip



2×4 bit

$1K \times 4\text{bit}$

- ① How many chip - 2^{chip}
 - ② Dataline \Rightarrow 4bit

③ Address line

Diagram illustrating a 10-bit address bus (Address bus) and its connection to two 5-bit output lines (labeled 0 and 1).

The address bus consists of 10 bits, labeled A₉ through A₀. The bits are grouped into two sets of 5 bits each, labeled C₁ and C₂.

Output line 0 is connected to the lower 5 bits of the address bus (A₄ to A₀). Output line 1 is connected to the upper 5 bits of the address bus (A₉ to A₅).

Both output lines have a range of 0 to 1023.

$$2K = 2 \times 1024 = \boxed{2048}$$

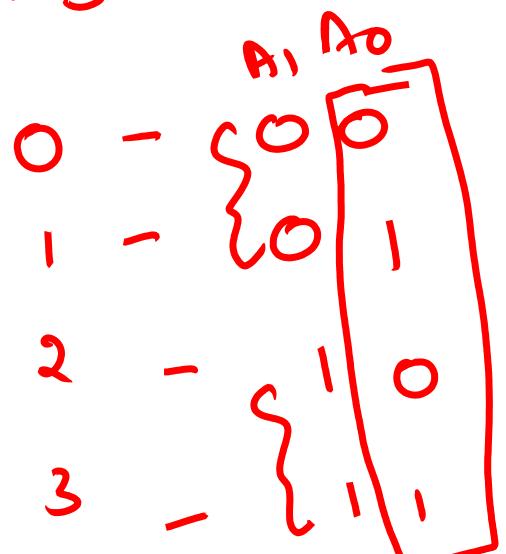
O - 2047

Addressability

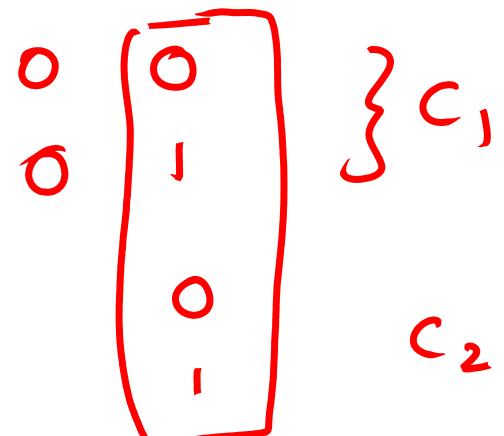
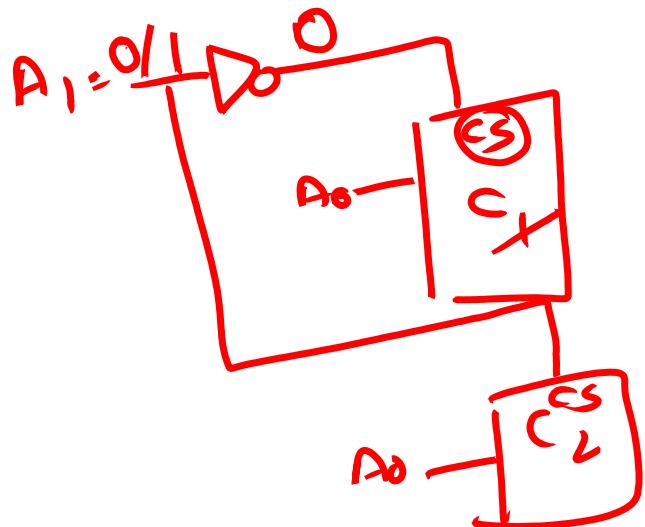
Address span

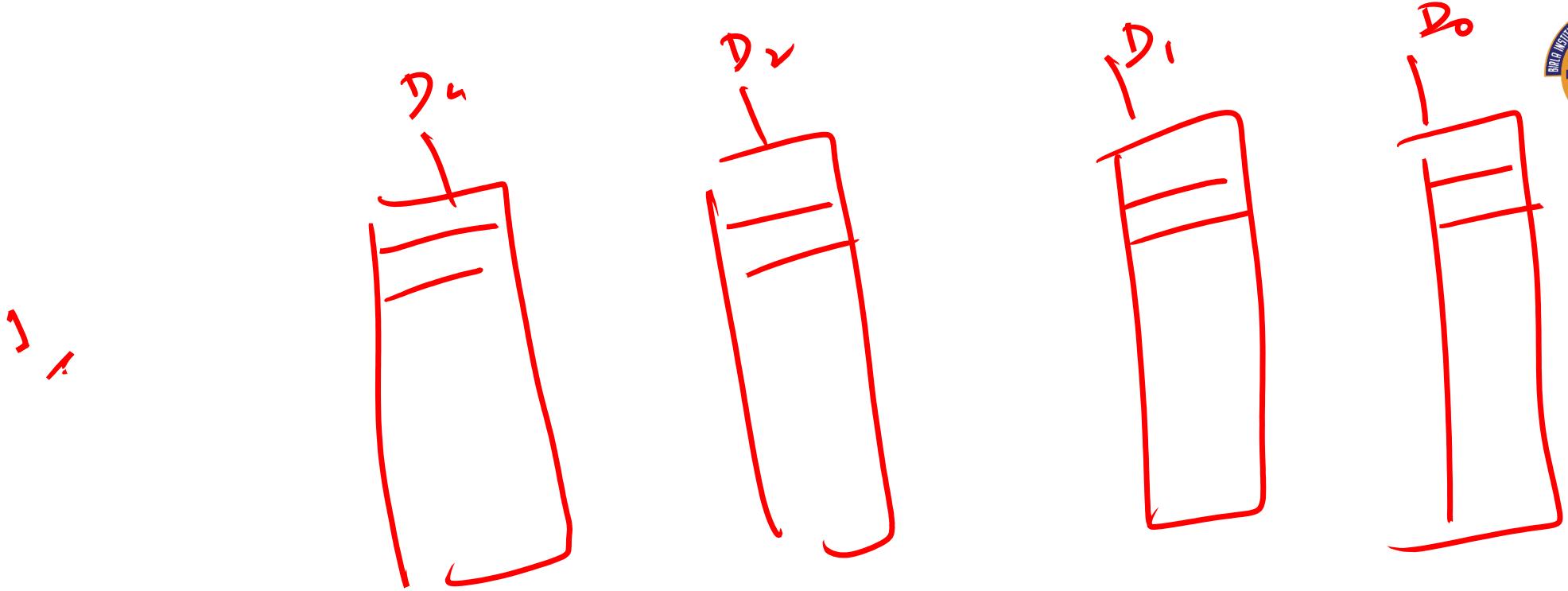
4×1 bit
1 bit

4
0 - 3



2×1 bit
1 bit
2
0 - 1





Error Correction

✓ Hard Failure

- Caused by harsh environmental abuse or manufacturing defects or wear
- Memory cell is permanently stuck at 0 or 1
- Permanent defect

✓ Soft Error

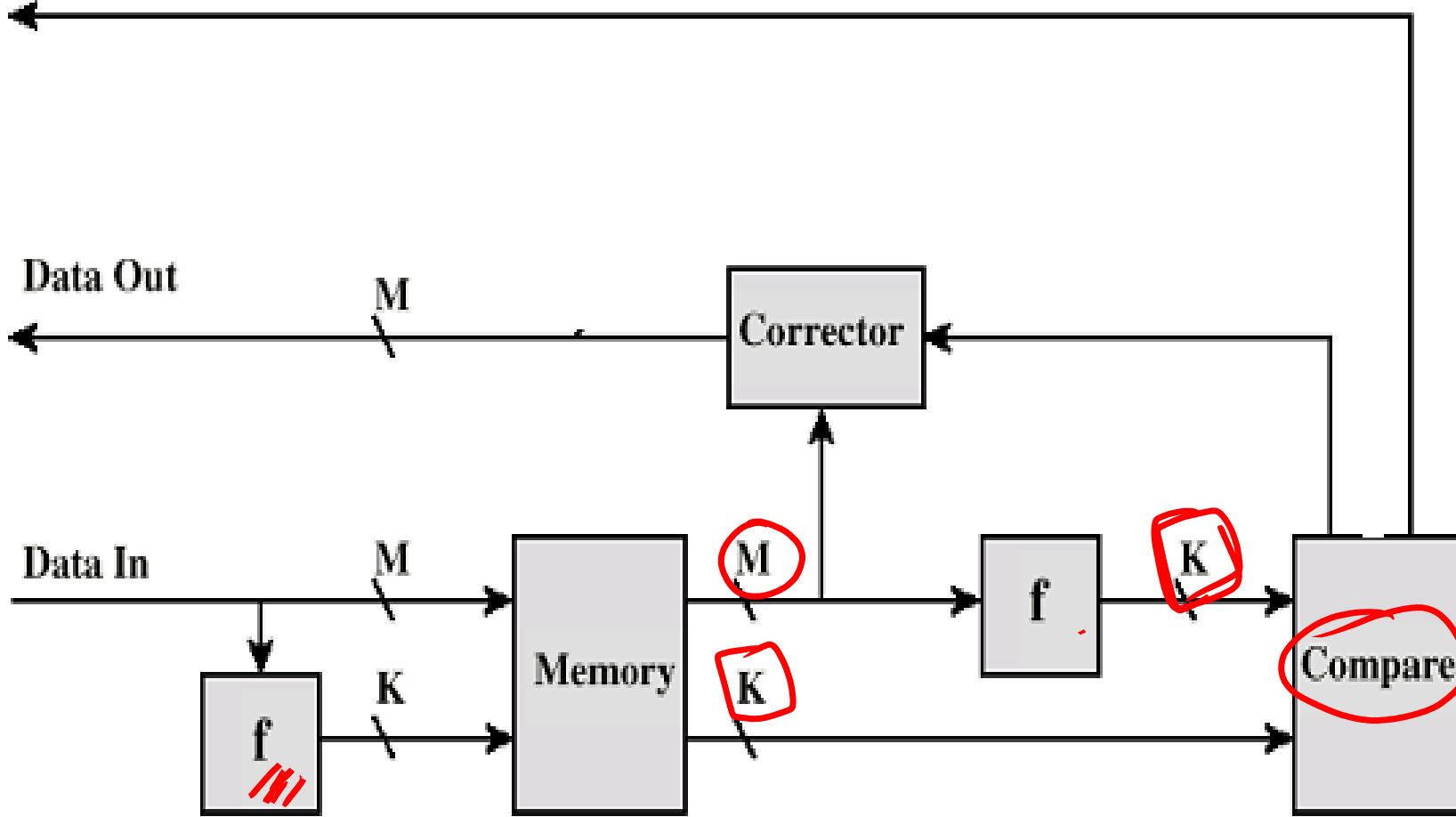
- Random, non-destructive
- alters the contents of one or more memory cells without damaging the memory.
- No permanent damage to memory
- Caused by power supply problems
- Detected using Hamming error correcting code

Error Correcting Code Function



Data $\rightarrow M$ bits

Error Signal



Error Correcting Code Function

- The comparison logic receives as input two K-bit values. A bit-by-bit comparison is done by taking the exclusive-OR of the two inputs. The result is called the syndrome word.
- The comparison yields one of three results
 - No errors are detected.
 - An error is detected and it is possible to correct the error
 - An error is detected but it is not possible to correct it.

x y z

0	0	0
0	1	-
0	0	-

Hamming Code....

- What should be the length of the code K ?
- Result of comparison is known as syndrome word
- length of the syndrome word is K bits, Length of Data is “M” bits
- Length of K should satisfy
- $2^k - 1 \geq M + K$

$$M = 8 \text{ bits}$$

$$K = 4 \text{ bits}$$

-

$$K = 1 \quad 2^1 - 1 \geq 8 + 1 \Rightarrow F$$

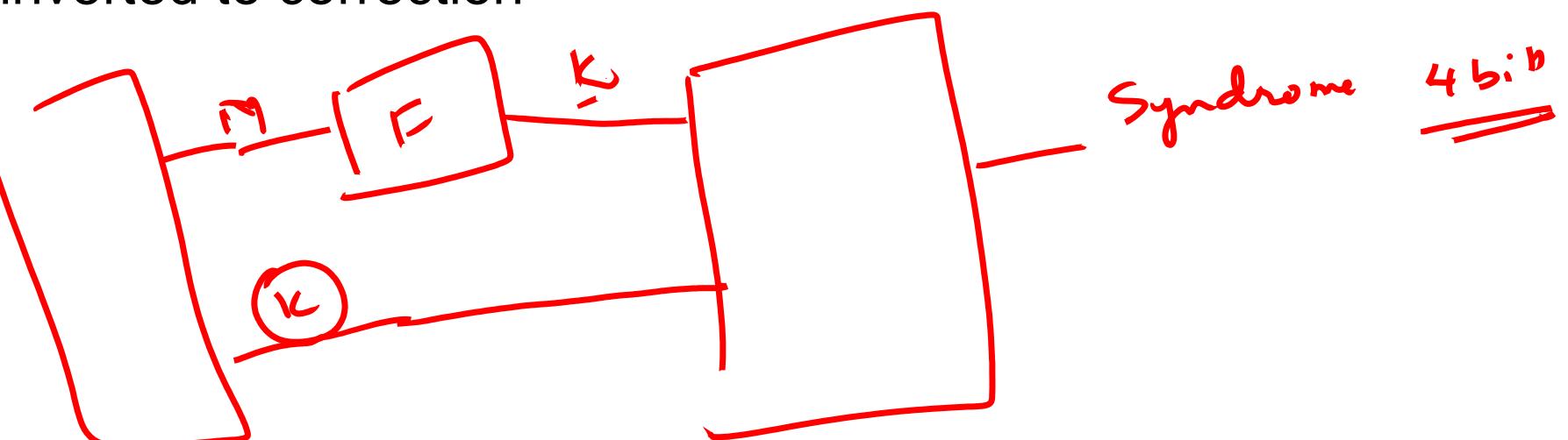
$$K = 2 \quad 2^2 - 1 \geq 8 + 2 \Rightarrow F$$

$$K = 3 \quad 2^3 - 1 \geq 8 + 3 \Rightarrow F$$

$$K = 4 \quad 2^4 - 1 \geq 8 + 4 \Rightarrow T$$

Hamming code....

- Generate 4-bit syndrome for an 8 bit data word with following characteristics
 - If the syndrome contains all 0's, no error has been detected.
 - If the syndrome contains one and only one bit set to one, then an error has occurred in one of the 4 check bits. No correction is needed
 - If the syndrome contains more than one bit set to 1, then the numerical value of the syndrome indicates the position of the data bit in error. This data bit is inverted to correction



Layout of Data and Check bits



$$\begin{array}{r} r_1 = 8 \\ K = 4 \\ \hline 12 \end{array}$$

$D_8 \dots D_2 D_1$
 $C_4 C_3 C_2 C_1$

Bit position	12	11	10	9	8	7	6	5	4	3	2	1
Position Number	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data bit	D8	D7	D6	D5		D4	D3	D2		D1		
Check Bit					C4				C3		C2	C1

8421

Layout of Data and Check bits

Bit position	12	11	10	9	8	7	6	5	4	3	2	1
Position Number	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data bit	D8	D7	D6	D5		D4	D3	D2		D1		
Check Bit					C4				C3		C2	C1

- $C_1 = D_1 \oplus D_2 \oplus D_4 \oplus D_5 \oplus D_7$ *(Exor)*
- $C_2 = D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_7$ *(Exor)*
- $C_3 = D_2 \oplus D_3 \oplus D_4 \oplus D_8$
- $C_4 = D_5 \oplus D_6 \oplus D_7 \oplus D_8$

Layout of Data and Check bits

Bit position	12	11	10	9	8	7	6	5	4	3	2	1
Position Number	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data bit	D8	D7	D6	D5		D4	D3	D2		D1		
Check Bit					C4				C3		C2	C1

- $C_1 = D_1 \oplus D_2 \oplus D_4 \oplus D_5 \oplus D_7$
- $C_2 = D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_7$
- $C_3 = D_2 \oplus D_3 \oplus D_4 \oplus D_8$
- $C_4 = D_5 \oplus D_6 \oplus D_7 \oplus D_8$

CS



Problem 1

Consider the data + code k is as follows:

110101011101

Find out if there is an error. If so which bit is having error?

Problem 1 - Solution



Consider the data + check bit is as follows:

110101011101 \Rightarrow 12

Find out if there is an error. If so which bit is having ?

Bit position	12	11	10	9	8	7	6	5	4	3	2	1
Position Number	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data bit	D8	D7	D6	D5		D4	D3	D2	D1			
Check Bit				C4				C3	C2	C1		
	1	1	0	1	0	1	0	1	1	1	0	1

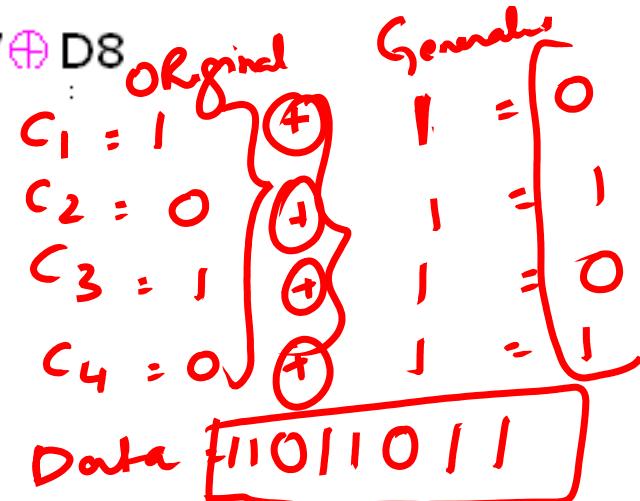
$$C_1 = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \Rightarrow 1$$

$$C_2 = 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \Rightarrow 1$$

$$C_3 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \Rightarrow 1$$

$$C_4 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \Rightarrow 1$$

- $C_1 = D_1 \oplus D_2 \oplus D_4 \oplus D_5 \oplus D_7$
- ✓ $C_2 = D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_7$ ✓
- $C_3 = D_2 \oplus D_3 \oplus D_4 \oplus D_8$
- $C_4 = D_5 \oplus D_6 \oplus D_7 \oplus D_8$



$$\begin{matrix} x & y & z \\ 0 & 0 & 0 \\ a & 1 & 1 \end{matrix}$$

$$\begin{matrix} & & \\ 1 & 0 & 1 \end{matrix}$$

$$\begin{matrix} & & \\ 1 & 0 & 1 \end{matrix}$$

Syndrome:

$$\begin{matrix} C_4 & C_3 & C_2 & C_1 \\ 1 & 0 & 1 & 0 \end{matrix} = 10$$

- $C_1 = D_1 \oplus D_2 \oplus D_4 \oplus D_5 \oplus D_7$
- $C_2 = D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_7$
- $C_3 = D_2 \oplus D_3 \oplus D_4 \oplus D_8$
- $C_4 = D_5 \oplus D_6 \oplus D_7 \oplus D_8$



Problem 2

Data : 10101100 ($M = 8$)

Compute Check Bits

Bit position	12	11	10	9	8	7	6	5	4	3	2	1
Position Number	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data bit	D8	D7	D6	D5		D4	D3	D2		D1		
Check Bit					C4				C3		C2	C1

$$\begin{array}{r}
 0101 \\
 1101 \\
 \hline
 1000
 \end{array}
 \quad \text{Syndrom}$$

M + K

① Position item on the layout
findout $D_8 - D_4$

$C_4 C_3 C_2 C_1 \Rightarrow$ Original check bits

② recalculate $C_1 C_2 C_3 C_4$

$C_4 C_3 C_2 C_1$

③ Perform XOR original & calculated check bits

0010

0110

↓
6

\Downarrow
Syndrom

④ Examine Syndrom

- if all bits of quo \Rightarrow no error
- if syndrome contains single 1 \Rightarrow error in the check bit
- if more than one bit in syndrome is one \Rightarrow There is a error