بسم الله الرحمن الرحيم

# Benha University

*Prof . Mohamed Ahmed Ebrahim*

**Benha University**
**Faculty of Engineering at Shoubra**
**Electrical Engineering Dept.**

# Industrial Controls (1)

*By*

*Associate Prof. / Mohamed Ahmed Ebrahim Mohamed*

E-mail: mohamedahmed_en@yahoo.com

mohamed.mohamed@feng.bu.edu.eg

Web site: http://bu.edu.eg/staff/mohamedmohamed033

# Lecture (5)

# 02 – 05 - 2021

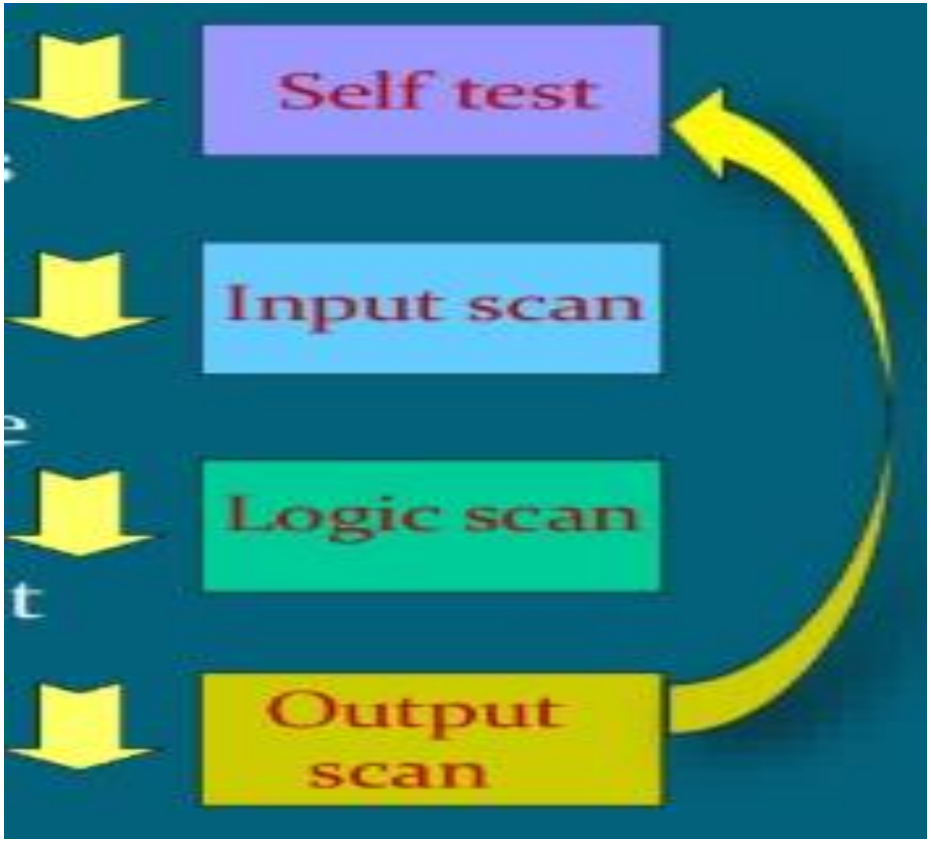
# PLC operation sequence



1. Self test
- Testing of its own hardware and software for faults.
2. Input scan
- If there are no problems, PLC will copy all the inputs and copy their values into memory.
3. Logic solve/scan
- Using inputs, the ladder logic program is solved once and outputs are updated.

4. Output scan

- While solving logic the output values are updated only in memory when ladder scan is done, the outputs will be updated using temporary values in memory.



# Programming languages of PLC

- Most common languages encountered in PLC programming are:

1. Ladder logic.
2. Functional Block Diagram.
3. Sequential Function Chart.
4. Boolean Mnemonics.

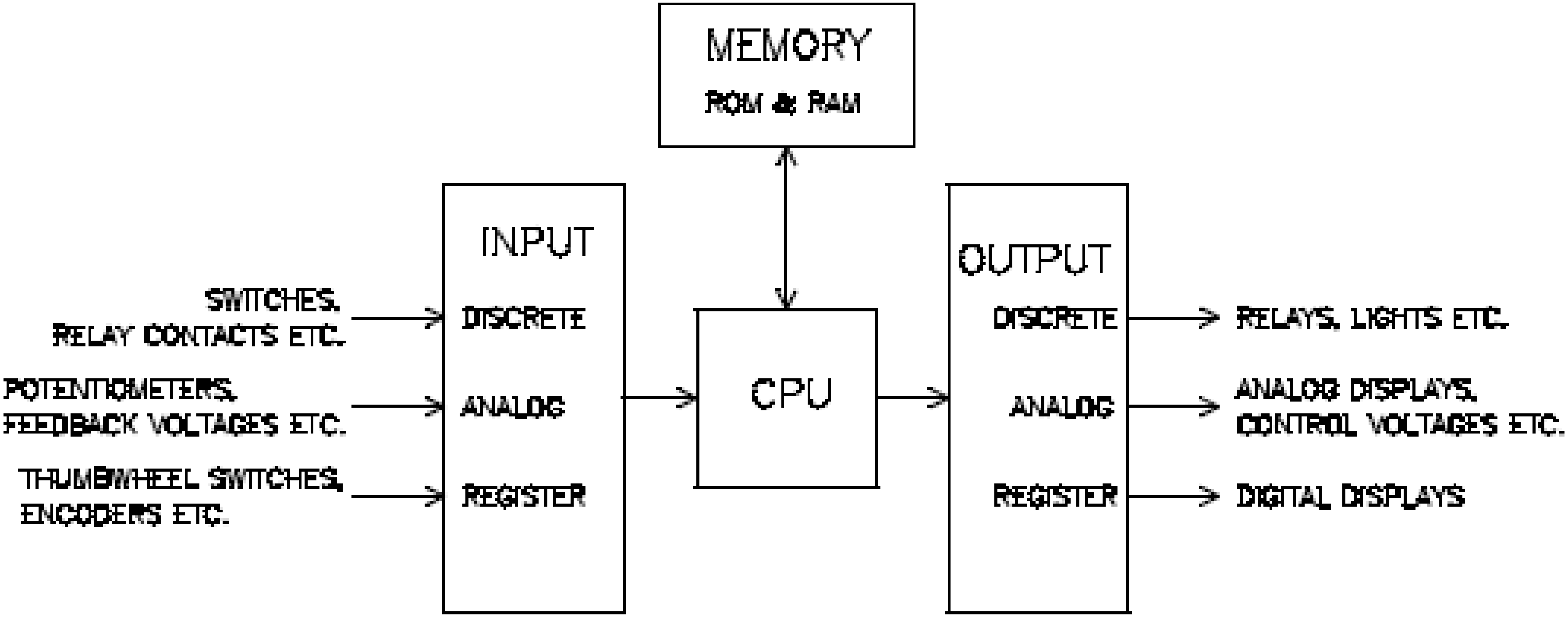
# Introduction to Ladder Programming

*Prof . Mohamed Ahmed Ebrahim*

# Outline

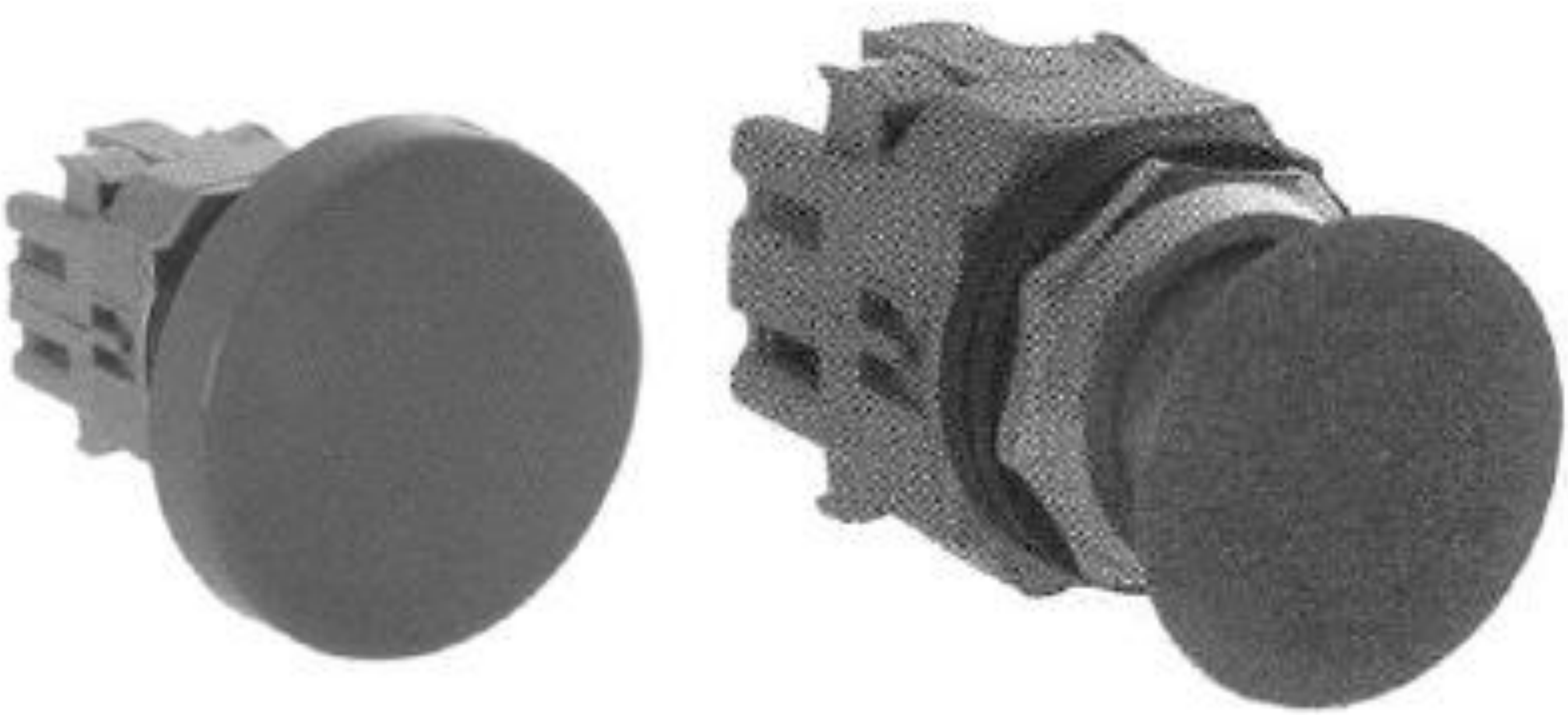
1. System Block Diagram
2. Basic Components and Their Symbols
3. Ladder Diagram Fundamentals
4. Applications

# PLC Block Diagram



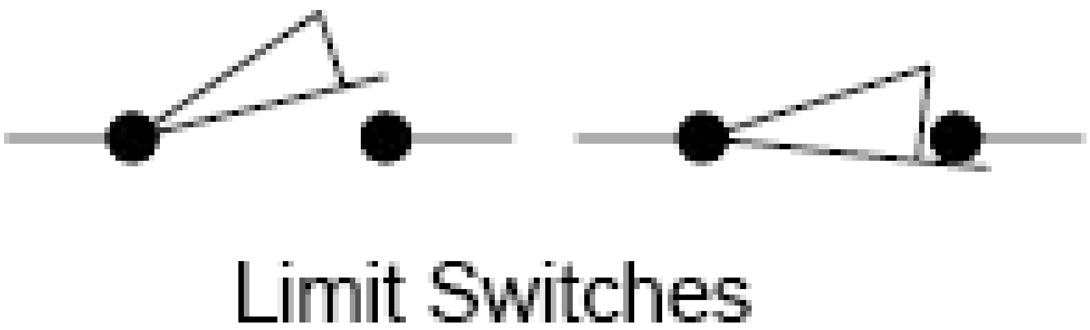
# Basic Components and Their Symbols

## Mushroom Head Push Button Switches



# Basic Components (Cont'd)

## Limit Switches (LS)



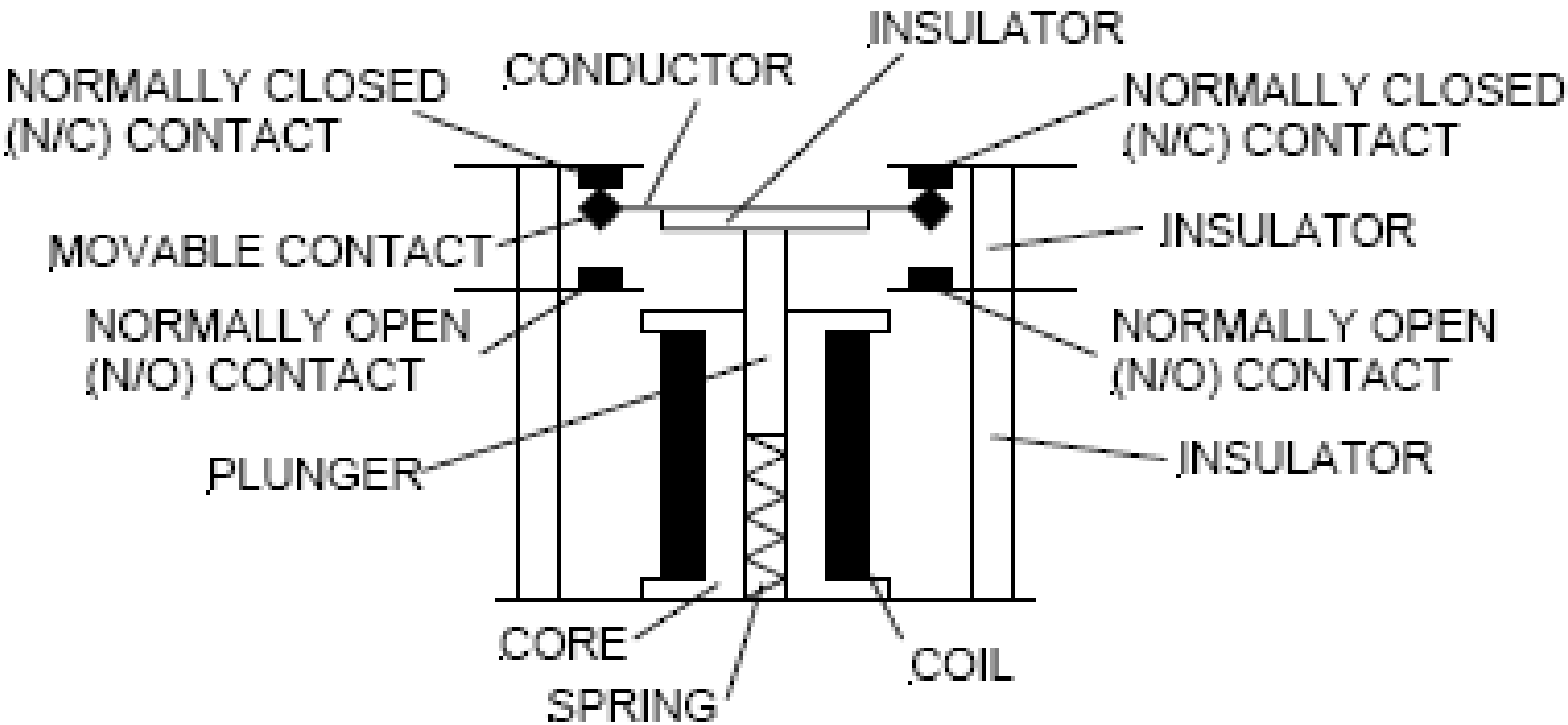

**Limit switches can be mechanical or light activated switches**

**Examples: limit switches on the refrigerator door that turns ON the inside or to open doors in supermarkets**

# Basic Components (Cont'd)

## Relays or Contactors

Electromagnetic devices

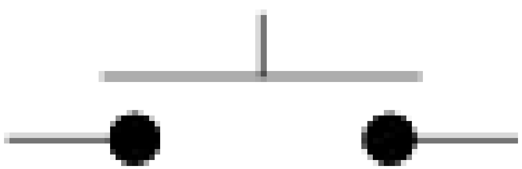



Relay or Contactor

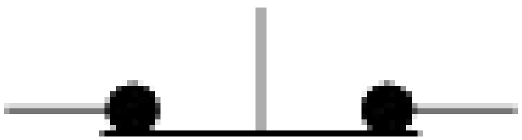
*Prof. Mohamed Ahmed Ebrahim*

# Basic Components (Cont’d)

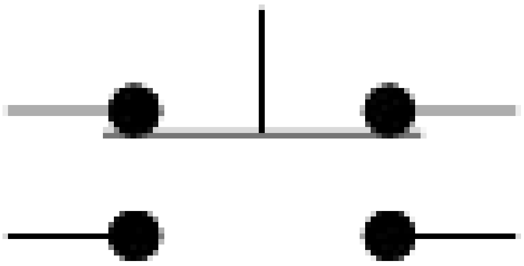
## Push Button (PB) Switches



Normally Open
(NO or N/O)



Normally Closed
(NC or N/C)

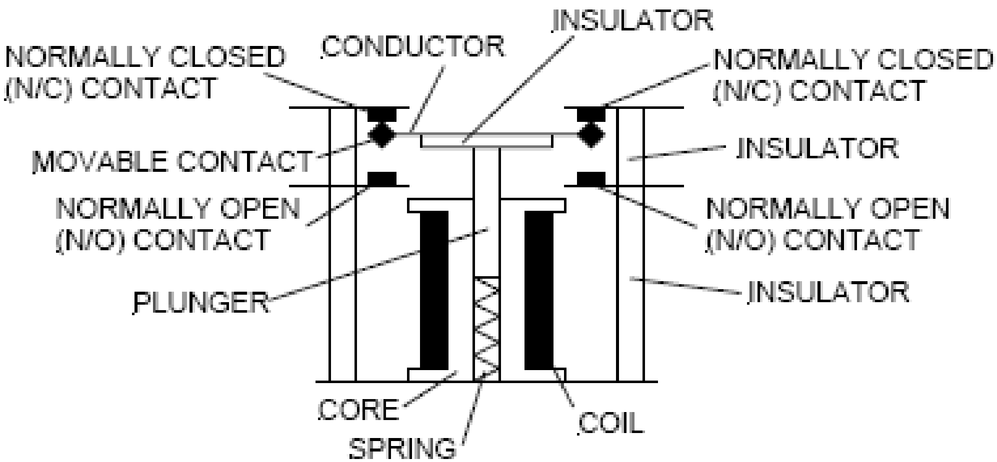


One PB with both
NO and NC contacts

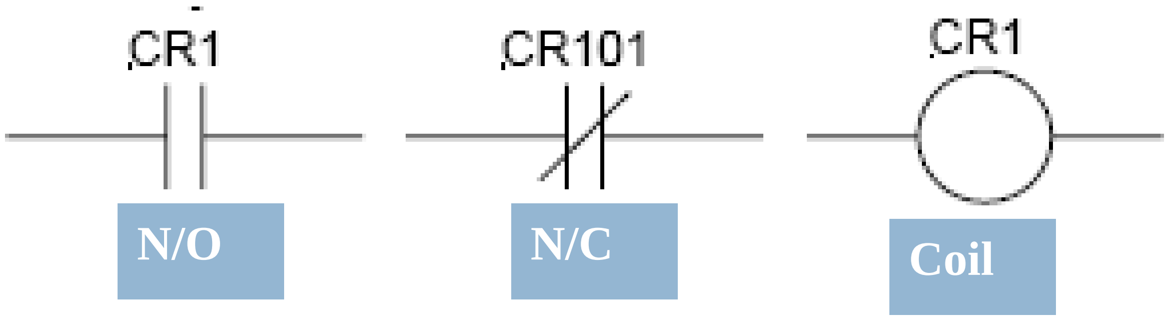
Momentary Pushbutton Switches

# Basic Components (Cont'd)

## Relays Symbols




Relay or Contactor

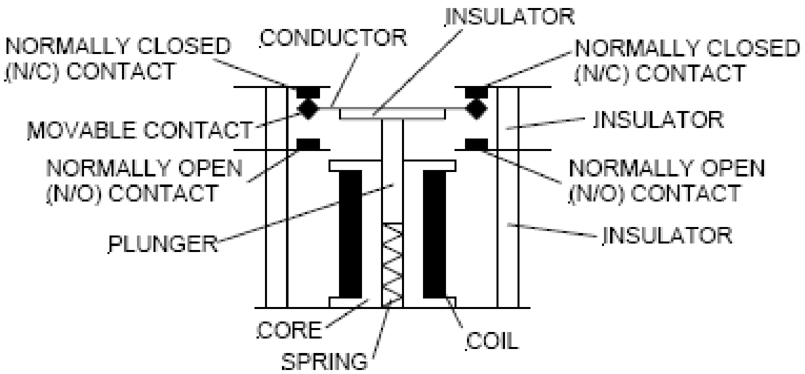



Relay Symbols

# Basic Components (Cont'd)

CR

control relay
(internal relay or software relay)



Relay or Contactor

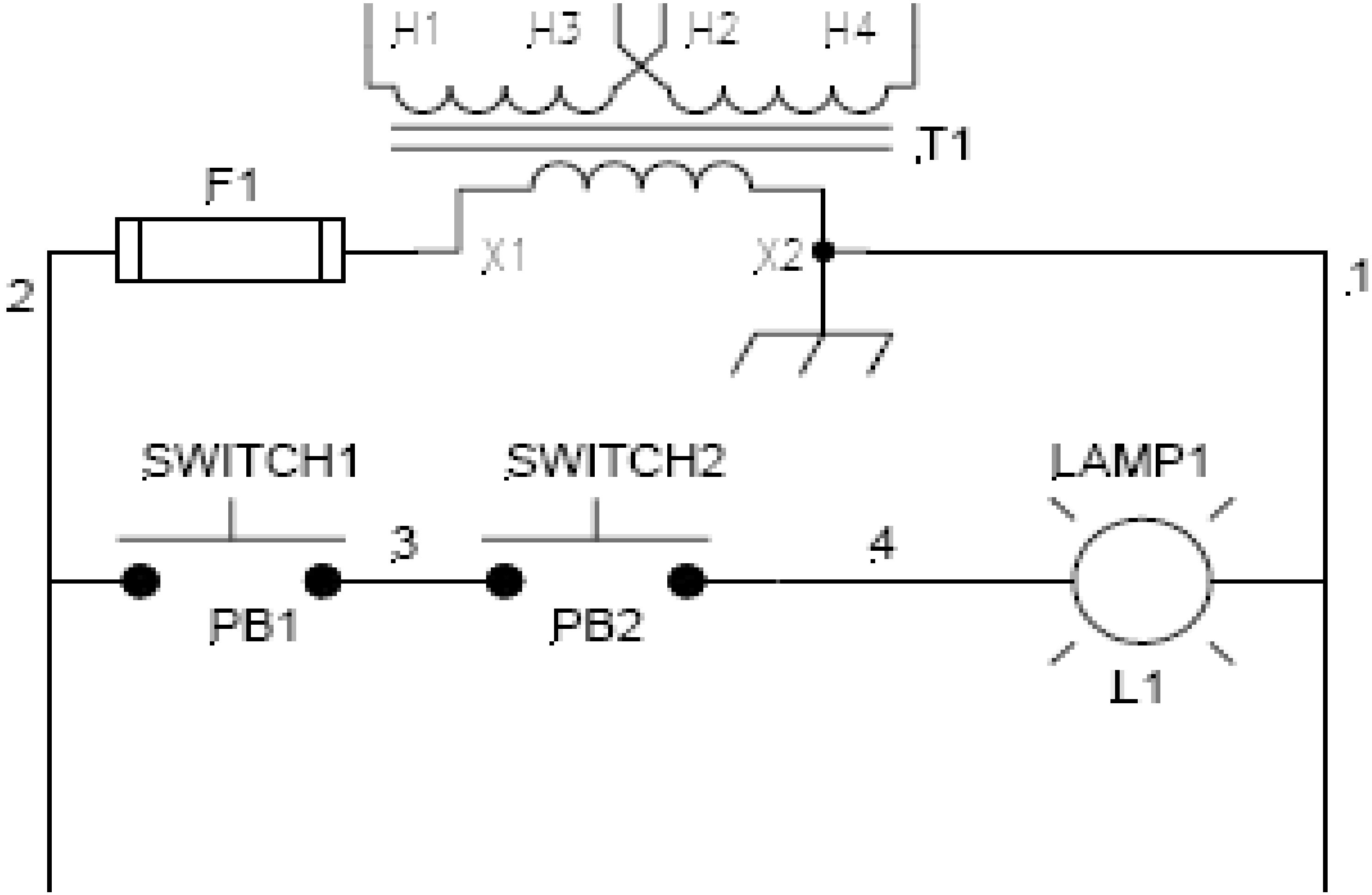
When coil CR1 is energized, all the N/O CR1 contacts will be closed and all the N/C CR1 contacts will be open.

Likewise, if coil CR1 is de-energized, all the N/O CR1 contacts will be open and all the N/C CR1 contacts will be closed.

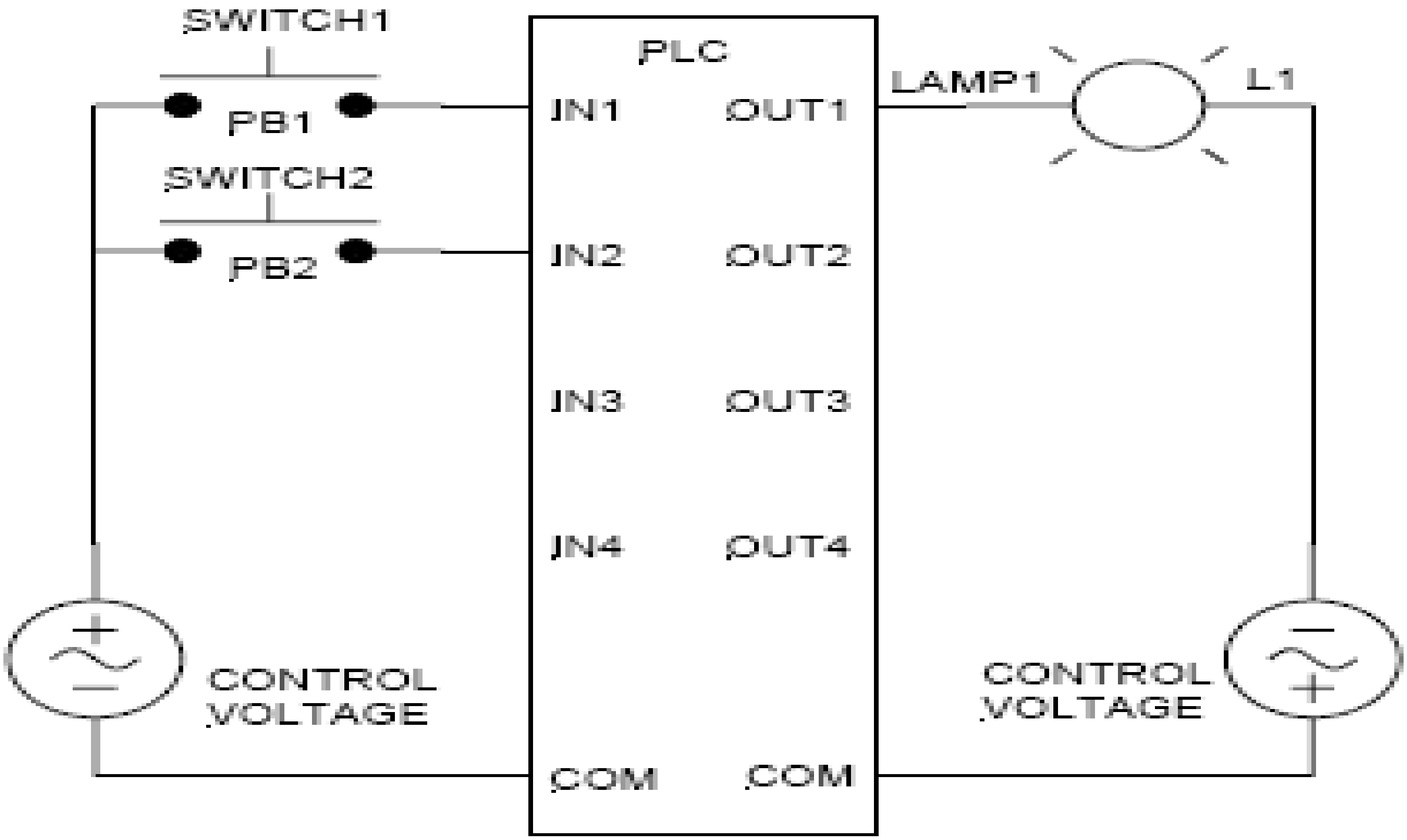
A contact labeled CR indicates that it is associated with a relay coil.

Each relay will have a specific number associated with it. The range of numbers used will depend upon the number of relays in the system.

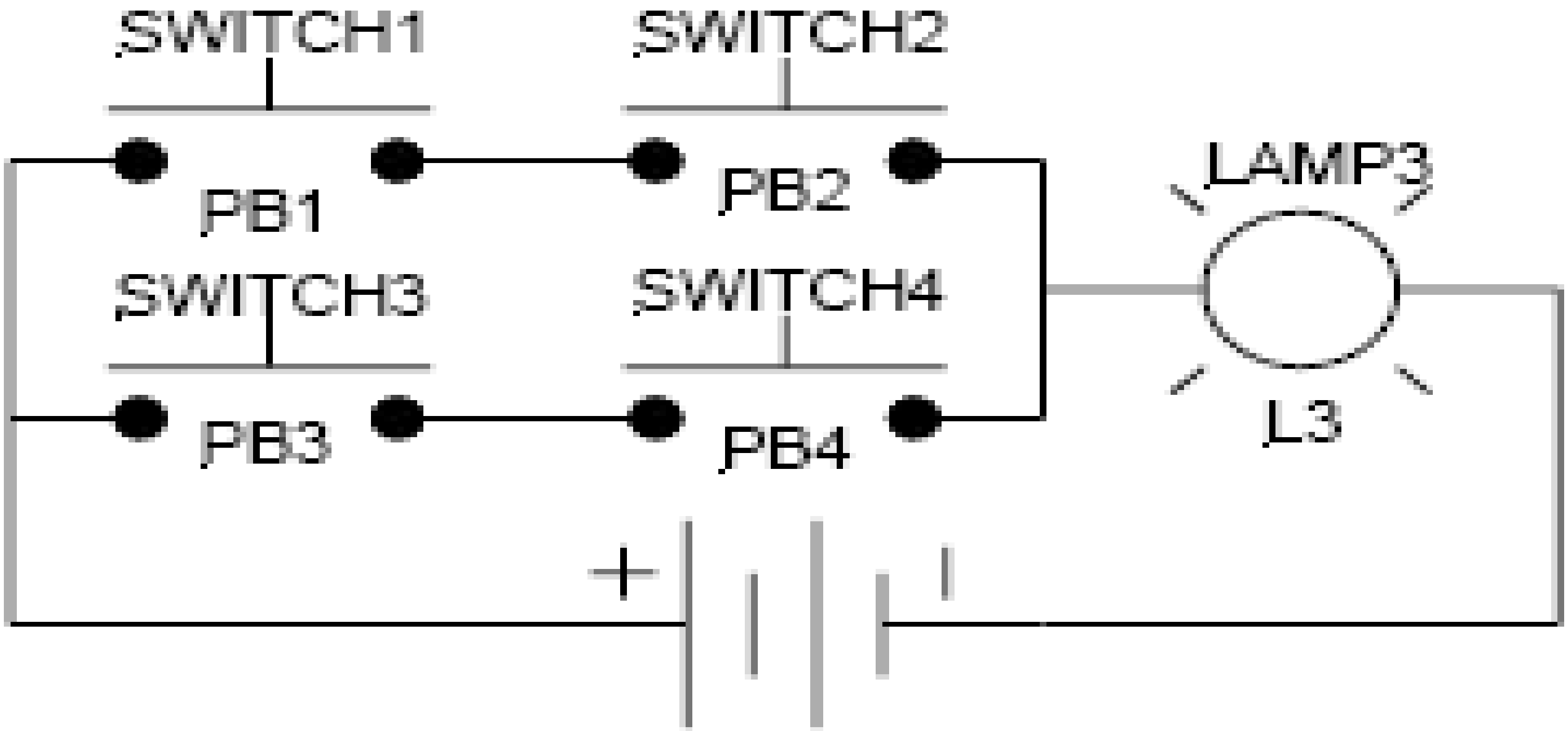
# Example: AND Circuit



# Example: AND Circuit (Cont'd)



# Example: AND/OR Circuit




AND-OR Lamp Circuit

# Numbering Systems

# &

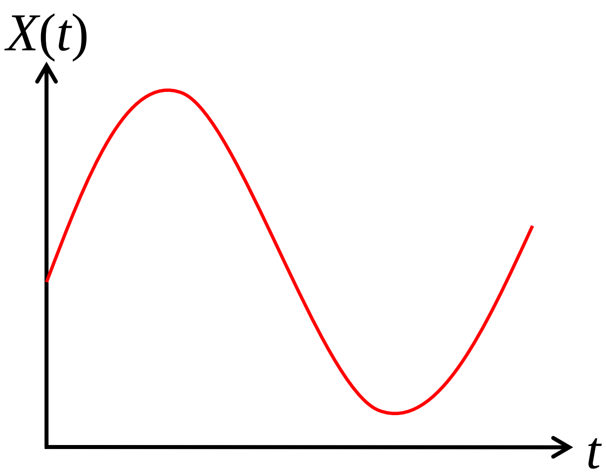
# Codes

*Prof . Mohamed Ahmed Ebrahim*

# Analog and Digital Signal

## Analog system

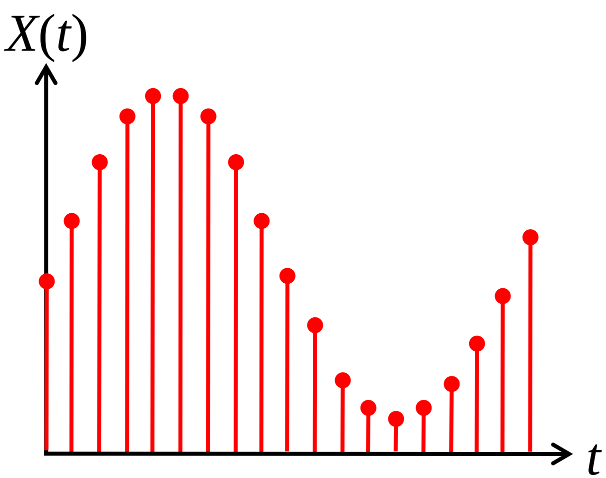
- The physical quantities or signals may vary continuously over a specified range.




Analog signal

## Digital system

- The physical quantities or signals can assume only discrete values.
- Greater accuracy

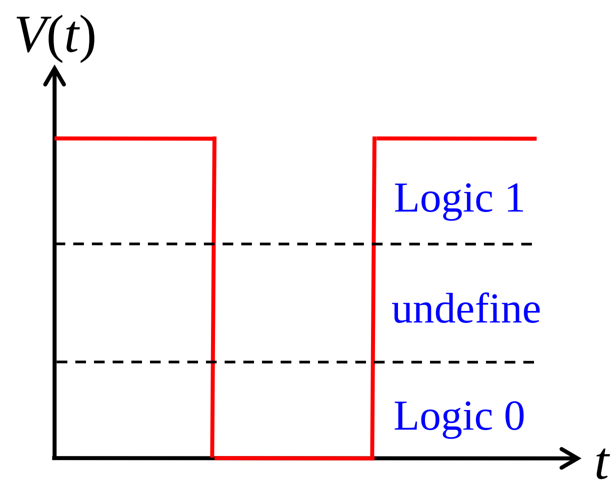



Digital signal

*Dr : Mohamed Ahmed Ebrahim*

# Binary Digital Signal

- An information variable represented by physical quantity.
- For digital systems, the variable takes on discrete values.
  - Two level, or binary values are the most prevalent values.
- Binary values are represented abstractly by:
  - Digits 0 and 1
  - Words (symbols) False (F) and True (T)
  - Words (symbols) Low (L) and High (H)
  - And words On and Off
- Binary values are represented by values or ranges of values of physical quantities.



Binary digital signal

# Numbering Systems

- A familiarity with number systems is quite useful when working with programmable controllers.

- In general, programmable controllers use binary numbers in one form or another to represent various codes and quantities.

# Cont.

- **The following statements apply to any number system:**

1. Every number system has a base or radix.
2. Every system can be used for counting.
3. Every system can be used to represent quantities or codes.
4. Every system has a set of symbols.

# Cont.

- The number systems usually encountered while using programmable controllers are base 2, base 8, base10, and base 16. These systems are called binary, octal, decimal, and hexadecimal, respectively.

| Numbering Systems | | |
|---|---|---|
| System | Base | Digits |
| Binary | 2 | 0 1 |
| Octal | 8 | 0 1 2 3 4 5 6 7 |
| Decimal | 10 | 0 1 2 3 4 5 6 7 8 9 |
| Hexadecimal | 16 | 0 1 2 3 4 5 6 7 8 9 A B C D E F |

# Decimal Numbering System

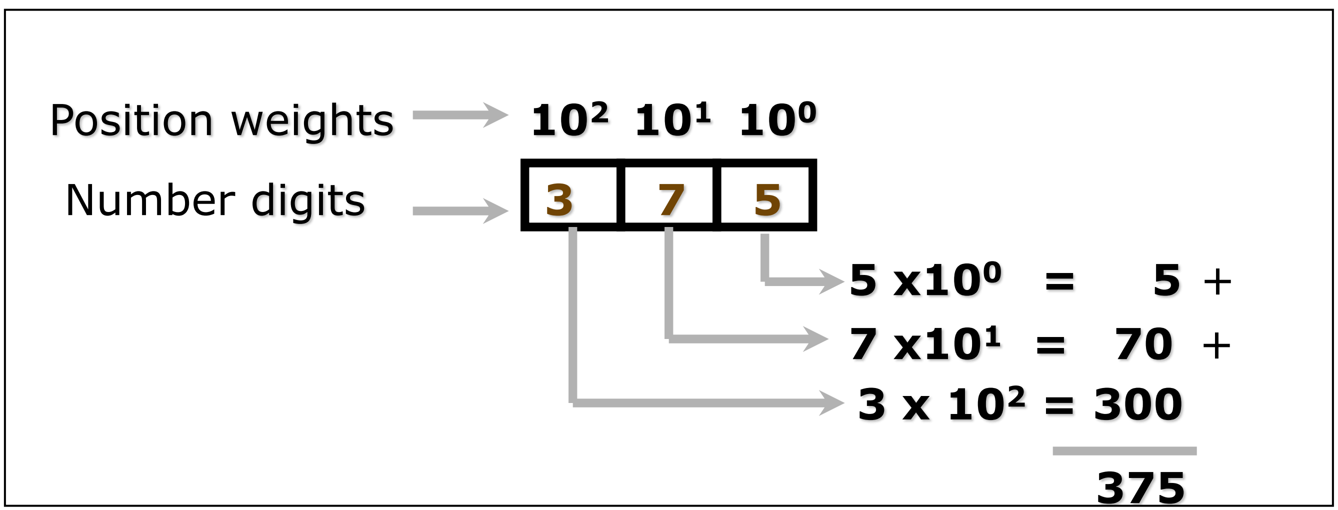
***Prof . Mohamed Ahmed Ebrahim***

# 1. Decimal Number System

- How is a **positive integer** represented in decimal?
- Let's analyze the decimal number **375**:

$$\mathbf{375} = (\mathbf{3} \times 100) + (\mathbf{7} \times 10) + (\mathbf{5} \times 1)$$

$$= (\mathbf{3} \times 10^2) + (\mathbf{7} \times 10^1) + (\mathbf{5} \times 10^0)$$



# Decimal System Principles

- A decimal number is a sequence of **digits**
- Decimal digits must be in the set:
  {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} (Base 10)
- Each digit contributes to the value the number represents
- The value contributed by a digit equals the product of the digit times the weight of the position of the digit in the number

# Cont.

- Position weights are powers of 10
- The weight of the rightmost (least significant digit) is $10^0$ (i.e.**1**)
- The weight of any position is $10^x$, where x is the number of positions to the right of the least significant digit

| | | | | | |
|---|---|---|---|---|---|
| Position weights | $10^4$ | $10^3$ | $10^2$ | $10^1$ | $10^0$ |
| digits | | | 3 | 7 | 5 |

# Bits

- In a computer, information is stored using digital signals that translate to binary numbers

- A single binary digit (0 or 1) is called a Bit.

  - A single bit can represent two possible states, on (1) or off (0)

- Combinations of bits are used to store values.

# Data Representation

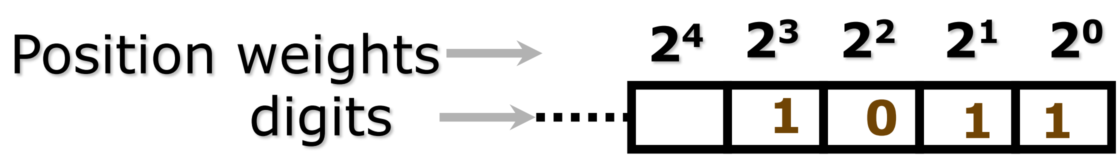
- Data representation means encoding **data** into **bits.**
  - Typically, **multiple bits** are used to represent the '**code**' of each **value** being represented

- Values being represented may be characters, numbers, images, audio signals, and video signals.

- Although a different scheme is used to encode each type of data, in the end the code is always a string of **zeros** and **ones.**

# Decimal to Binary

- So in a computer, the only possible digits we can use to encode data are **{0,1}**
  - The numbering system that uses this set of digits is the base 2 system (also called the Binary Numbering System)
- We can apply all the principles of the base 10 system to the base 2 system



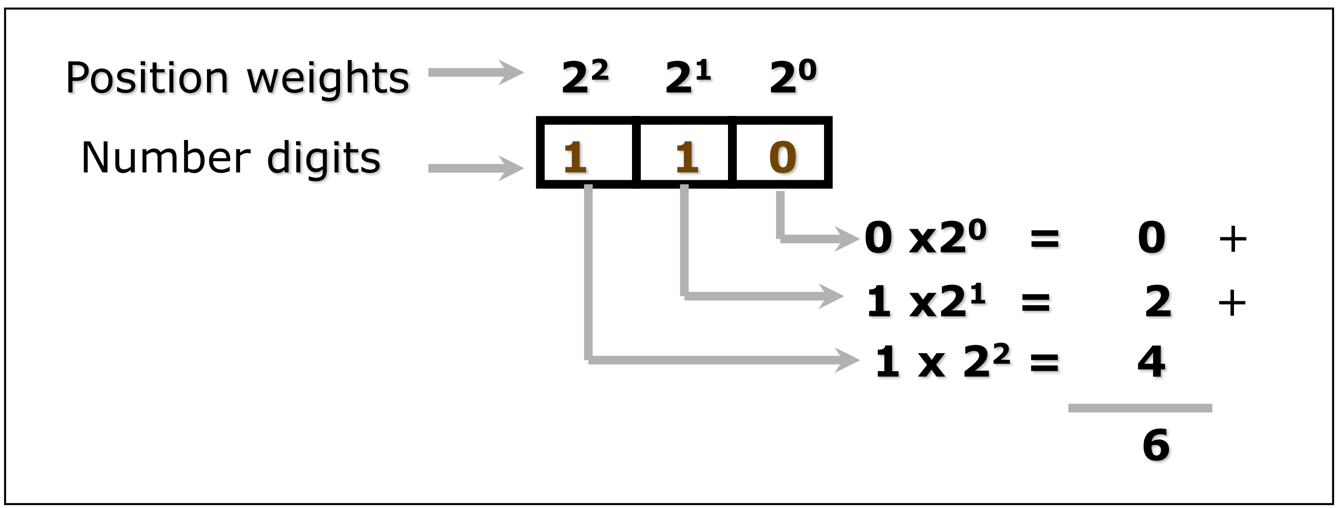
# Binary Numbering System

Prof . Mohamed Ahmed Ebrahim

# 2. Binary Numbering System

- How is a **positive integer** represented in **binary**?
- Let's analyze the binary number **110**:

$$\mathbf{110} = (\mathbf{1} \times 2^2) + (\mathbf{1} \times 2^1) + (\mathbf{0} \times 2^0)$$
$$= (\mathbf{1} \times 4) + (\mathbf{1} \times 2) + (\mathbf{0} \times 1)$$




- So a count of **SIX** is represented in binary as **110**

# Binary to Decimal Conversion

- To convert a base 2 (binary) number to base 10 (decimal):
  - Add all the values (positional weights) where a **one** digit occurs
  - Positions where a **zero** digit occurs do NOT add to the value, and can be ignored

# Cont.

**Example (1): Convert binary 100101 to decimal**

(written $1\ 0\ 0\ 1\ 0\ 1_2$ ) =

$1*2^0 +$ → $1 +$

$0*2^1 +$

$1*2^2 +$ → $4 +$

$0*2^3 +$

$0*2^4 +$

$1*2^5$ → $32$

$37_{10}$

# Cont.

**Example (2):** $10111_2$

| | | | | | |
|---|---|---|---|---|---|
| positional powers of 2: | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
| decimal positional value: | **16** | **8** | **4** | **2** | **1** |
| binary number: | 1 | 0 | 1 | 1 | 1 |

$16 + 4 + 2 + 1 = 23_{10}$

# Cont.

**Example (3):** $110010_2$

| | | | | | | |
|---|---|---|---|---|---|---|
| positional powers of 2: | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
| decimal positional value: | 32 | 16 | 8 | 4 | 2 | 1 |
| binary number: | 1 | 1 | 0 | 0 | 1 | 0 |

$$32 + 16 + 2 = 50_{10}$$

# Decimal to Binary Conversion

## The Division Method

1) Start with your number (call it N) in base 10
2) Divide N by 2 and record the remainder
3) If (quotient = 0) then stop

   else make the quotient your new N, and go back to step 2

The **remainders** comprise your answer, starting with the last remainder as your first (leftmost) digit.

In other words, divide the decimal number by 2 until you reach zero, and then collect the remainders in reverse.

# Cont.

Using the **Division** Method:

Divide decimal number by 2 until you reach zero, and then collect the **remainders** in reverse.

**Example(1):** $22_{10} = 10110_2$

| | Rem: |
|---|---|
| 2 ) 22 | |
| 2 ) 11 | 0 |
| 2 ) 5 | 1 |
| 2 ) 2 | 1 |
| 2 ) 1 | 0 |
| 0 | 1 |

# Cont.

Using the **Division** Method

**Example 2:** $56_{10} = 111000_2$

| | Rem: |
|---|---|
| 2 ) 56 | |
| 2 ) 28 | 0 |
| 2 ) 14 | 0 |
| 2 ) 7 | 0 |
| 2 ) 3 | 1 |
| 2 ) 1 | 1 |
| 0 | 1 |

Prof. Mohamed Ahmed Ebrahim

# Cont.

## The Subtraction Method

- Subtract out largest power of 2 possible (without going below zero), repeating until you reach 0.
  - Place a 1 in each position where you COULD subtract the value
  - Place a 0 in each position that you could NOT subtract out the value without going below zero.

# Cont.

**Example 1:** $21_{10}$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 21 | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
| | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| - 16 | | | | | | | |
| 5 | | | 1 | 0 | 1 | 0 | 1 |
| - 4 | | | | | | | |
| 1 | | | | | | | |
| - 1 | | | | | | | |
| 0 | | | | | | | |

Answer: $21_{10} = 10101_2$

*Prof. Mohamed Ahmed Ebrahim*

# Cont.

**Example 2:** $56_{10}$

$$
\begin{array}{r}
56 \\
- \underline{\mathbf{32}} \\
24 \\
\underline{-\ \mathbf{16}} \\
8 \\
\underline{-\ \mathbf{8}} \\
0
\end{array}
$$

| $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
|---|---|---|---|---|---|---|
| 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| | 1 | 1 | 1 | 0 | 0 | 0 |

Answer: $56_{10} = 111000_2$

*Prof. Mohamed Ahmed Ebrahim*

# Octal Numbering System

Prof . Mohamed Ahmed Ebrahim

# 3. Octal Numbering System

- **Base:** 8
- **Digits:** 0, 1, 2, 3, 4, 5, 6, 7

➢ Octal number: $357_8$

$$= (3 \times 8^2) + (5 \times 8^1) + (7 \times 8^0)$$

To convert to base 10, beginning with the **rightmost** digit, multiply each **n**th digit by $8^{(n-1)}$, and add all of the results together.

# Octal to Decimal Conversion

- **Example 1:** $357_8$

| | | | |
|---|---|---|---|
| positional powers of 8: | $8^2$ | $8^1$ | $8^0$ |
| decimal positional value: | 64 | 8 | 1 |
| Octal number: | 3 | 5 | 7 |

$$(3 \times 64) + (5 \times 8) + (7 \times 1)$$

$$= \quad 192 + 40 + 7 = 239_{10}$$

*Prof. Mohamed Ahmed Ebrahim*

# Cont.

- **Example 2:** $1246_8$

| | | | | |
|---|---|---|---|---|
| positional powers of 8: | $8^3$ | $8^2$ | $8^1$ | $8^0$ |
| decimal positional value: | 512 | 64 | 8 | 1 |
| Octal number: | 1 | 2 | 4 | 6 |

$$(1 \times 512) + (2 \times 64) + (4 \times 8) + (6 \times 1)$$

$$= 512 + 128 + 32 + 6 = 678_{10}$$

# Decimal to Octal Conversion

## The **Division** Method

1) Start with your number (call it N) in base 10
2) Divide N by 8 and record the remainder
3) If (quotient = 0) then stop

   else make the quotient your new N, and go back to step 2

The **remainders** comprise your answer, starting with the last remainder as your first (leftmost) digit.

In other words, divide the decimal number by 8 until you reach zero, and then collect the remainders in reverse.

# Cont.

Using the **Division** Method:

Example 1: $\mathbf{214_{10}} = \mathbf{326_8}$

| | | |
|---|---|---|
| 8 ) 214 | | Rem: |
| 8 ) 26 | | 6 |
| 8 ) 3 | | 2 |
| 0 | | 3 |

*Prof. Mohamed Ahmed Ebrahim*

# Cont.

Example 2: $4330_{10} = 10352_8$

| | | |
|---|---|---|
| 8 ) 4330 | | Rem: |
| 8 ) 541 | | 2 |
| 8 ) 67 | | 5 |
| 8 ) 8 | | 3 |
| 8 ) 1 | | 0 |
| | 0 | 1 |

*Prof. Mohamed Ahmed Ebrahim*

# Cont.

## The Subtraction Method

- Subtract out multiples of the largest power of 8 possible (without going below zero) each time until you reach 0.
  - Place the multiple value in each position where you COULD subtract the value.
  - Place a 0 in each position that you could NOT subtract out the value without going below zero.

# Cont.

**Example 1:** $315_{10}$

$$
\begin{array}{rl}
 & 315 \\
- & \underline{256} \quad (4 \times 64) \\
 & 59 \\
- & \underline{56} \quad (7 \times 8) \\
 & 3 \\
- & \underline{3} \quad (3 \times 1) \\
 & 0
\end{array}
$$

| $8^2$ | $8^1$ | $8^0$ |
|---|---|---|
| 64 | 8 | 1 |
| 4 | 7 | 3 |

Answer: $315_{10} = 473_8$

*Prof. Mohamed Ahmed Ebrahim*

# Cont.

Example 2: $2018_{10}$

$$
\begin{array}{rl}
 & 2018 \\
- & \underline{1536} \quad (3 \times 512) \\
 & 482 \\
- & \underline{448} \quad (7 \times 64) \\
 & 34 \\
- & \underline{32} \quad (4 \times 8) \\
 & 2 \\
- & \underline{2} \quad (2 \times 1) \\
 & 0
\end{array}
$$

| $8^4$ | $8^3$ | $8^2$ | $8^1$ | $8^0$ |
|---|---|---|---|---|
| 4096 | 512 | 64 | 8 | 1 |
| | 3 | 7 | 4 | 2 |

Answer: $2018_{10} = \mathbf{3742_8}$

*Prof. Mohamed Ahmed Ebrahim*

# Hexadecimal (Hex) Numbering System

Prof . Mohamed Ahmed Ebrahim

# 4. Hexadecimal (Hex)Numbering System

- Base: 16
- Digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

➢ Hexadecimal number: $1F4_{16}$

$$= (1 \times 16^2) + (F \times 16^1) + (4 \times 16^0)$$

# HEX Extra Digits

| Decimal Value | Hexadecimal Digit |
|---|---|
| 10 | A |
| 11 | B |
| 12 | C |
| 13 | D |
| 14 | E |
| 15 | F |

# Hex to Decimal Conversion

- **To convert to base 10:**

  A. Begin with the rightmost digit

  B. Multiply each **n**th digit by $16^{(n-1)}$

  C. Add all of the results together

# Cont.

- Example 1: $1F4_{16}$

positional powers of 16: $16^3$ $16^2$ $16^1$ $16^0$
decimal positional value: 4096 256 16 1

Hexadecimal number: 1 F 4

$$(1 \times 256) + (F \times 16) + (4 \times 1)$$
$$= (1 \times 256) + (15 \times 16) + (4 \times 1)$$

Answer $= 256 + 240 + 4 = 500_{10}$

# Cont.

- Example 2: $25AC_{16}$

positional powers of 16: $16^3$ $16^2$ $16^1$ $16^0$
decimal positional value: 4096 256 16 1

Hexadecimal number: 2 5 A C

$(2 \times 4096) + (5 \times 256) + (A \times 16) + (C \times 1)$

$= (2 \times 4096) + (5 \times 256) + (10 \times 16) + (12 \times 1)$

Answer $= 8192 + 1280 + 160 + 12 = 9644_{10}$

# Decimal to Hex Conversion

## The **Division** Method

1) Start with your number (call it N) in base 10
2) Divide N by 16 and record the remainder
3) If (quotient = 0) then stop

   else make the quotient your new N, and go back to step 2

The **remainders** comprise your answer, starting with the last remainder as your first (leftmost) digit.

In other words, divide the decimal number by 16 until you reach zero, and then collect the remainders in reverse.

# Cont.

Using The **Division** Method:

Example 1: **$126_{10}$ =**

| | | |
|---|---|---|
| 16) 126 | | Rem: |
| 16) 7 | | 14=E |
| | 0 | 7 |

**Answer= $7E_{16}$**

# Cont.

Example 2: $603_{10}$ =

| | |
|---|---|
| 16) 603 | Rem: |
| 16) 37 | 11=B |
| 16) 2 | 5 |
| 0 | 2 |

**Answer= $25B_{16}$**

# Cont.

## The Subtraction Method

- Subtract out multiples of the largest power of 16 possible (without going below zero) each time until you reach 0.
  - Place the multiple value in each position where you COULD to subtract the value.
  - Place a 0 in each position that you could NOT subtract out the value without going below zero.

# Cont.

Example 1: $810_{10}$

$$
\begin{array}{rl}
810 & \\
\underline{-\ 768} & (3 \times 256) \\
42 & \\
\underline{-\ \ \ 32} & (2 \times 16) \\
10 & \\
\underline{-\ \ \ 10} & (10 \times 1) \\
0 &
\end{array}
$$

| $16^2$ | $16^1$ | $16^0$ |
|---|---|---|
| 256 | 16 | 1 |
| 3 | 2 | A |

Answer: $810_{10} = 32A_{16}$

# Cont.

Example 2: $\mathbf{156_{10}}$

$$
\begin{array}{rl}
156 & \\
\underline{-\ 144} & (9 \times 16) \\
12 & \\
\underline{-\quad 12} & (12 \times 1) \\
0 &
\end{array}
$$

| $16^2$ | $16^1$ | $16^0$ |
|---|---|---|
| 256 | 16 | 1 |
| | 9 | C |

Answer: $156_{\mathbf{10}} = \mathbf{9C_{16}}$

# Numbering Conversion

*Prof . Mohamed Ahmed Ebrahim*

# Binary to Octal Conversion

- The maximum value represented in 3 bit is: $2^3 - 1 = 7$

- So using 3 bits we can represent values from 0 to 7
which are the digits of the Octal numbering system.

- Thus, three binary digits can be converted to one octal digit.

# Cont.

| Three-bit Group | Decimal Digit | Octal Digit |
|---|---|---|
| 000 | 0 | 0 |
| 001 | 1 | 1 |
| 010 | 2 | 2 |
| 011 | 3 | 3 |
| 100 | 4 | 4 |
| 101 | 5 | 5 |
| 110 | 6 | 6 |
| 111 | 7 | 7 |

# Cont.

**Ex:** Convert $10100110_2$ to octal

Starting at the right end, split into groups of 3:

10 100 110 →

110 = 6

100 = 4

010 = 2 (pad empty digits with 0)

Answer: $10100110_2 = 246_8$

# Octal to Binary Conversion

Ex: Convert $742_8$ to binary

Convert each octal digit to 3 bits:

7 = 111

4 = 100

2 = 010

111 100 010

$742_8 = 111100010_2$

# Binary to Hex Conversion

- The maximum value represented in 4 bit is:

  $$24 - 1 = 15$$

- So using 4 bits we can represent values from 0 to 15 which are the digits of the Hexadecimal numbering system.

- Thus, four binary digits can be converted to one hexadecimal digit.

| Four Bit Group | Decimal Digit | HEX Digit |
|---|---|---|
| 0000 | 0 | 0 |
| 0001 | 1 | 1 |
| 0010 | 2 | 2 |
| 0011 | 3 | 3 |
| 0100 | 4 | 4 |
| 0101 | 5 | 5 |
| 0110 | 6 | 6 |
| 0111 | 7 | 7 |
| 1000 | 8 | 8 |
| 1001 | 9 | 9 |
| 1010 | 10 | A |
| 1011 | 11 | B |
| 1100 | 12 | C |
| 1101 | 13 | D |
| 1110 | 14 | E |
| 1111 | 15 | F |

# Cont.

**Ex:** Convert $110100110_2$ to hex

Starting at the right end, split into groups of 4:

11010 0110 →

0110 = 6

1010 = A

0001 = 1 (pad empty digits with 0)

**Answer:** $110100110_2$ = $1A6_{16}$

# Hex to Binary Conversion

**Ex:** Convert $3D9_{16}$ to binary

Convert each hex digit to 4 bits:

3 = 0011

D = 1101

9 = 1001

0011 1101 1001 →

**Answer:** $3D9_{16} = 1111011001_2$ (can remove leading zeros)

# Octal to Hex Conversion

- To convert between the Octal and Hexadecimal numbering systems:
  - Convert from one system to binary first
  - Then convert from binary to the new numbering system

# Cont.

**Ex:** Convert $752_8$ to hex

1. First convert the octal to binary:

   $111\ 101\ 010_2$

   $0001\ \ 1110\ \ 1010$ re-group by 4 bits (add leading zeros)

2. Then convert the binary to hex:

   1 E A

**So** $752_8 = 1EA_{16}$

Prof. Mohamed Ahmed Ebrahim

# Hex to Octal Conversion

**Ex:** Convert $E8A_{16}$ to octal

1. First convert the hex to binary:

   $1110\ 1000\ 1010_2$

   111 010 001 010 and re-group by 3 bits (starting on the right)

2. Then convert the binary to octal:

   7 2 1 2

**So** $E8A_{16} = 7212_8$

*Prof. Mohamed Ahmed Ebrahim*

# Activity

- **Ex:** Convert the following numbers:

1. $1010111101_2$ to Hex
2. $82F_{16}$ to Binary

# Answers

1. $1010111101_2$ → 10 1011 1101 = $2BD_{16}$

2. $82F_{16}$ = 0100 0010 1111 → $1000010111 1_2$

# Thank You For Your Attention

Mohamed Ahmed Ebrahim