Boolean Algebra

Instructor Name: Engr. Rimsha

Register Transfer

- A register transfer operation involves the transfer of binary information from one set of binary registers into other set of binary registers.
- The capture and storage of information requires:
  - An input register to store the key inputs from the keyboard
  - A processor register to store the data when processed by the CPU
  - A memory register in the memory unit to store the values

Register Transfer

- Data input at keyboard
- Shifted into place
- Stored in memory

  - NOTE: Data input in ASCII

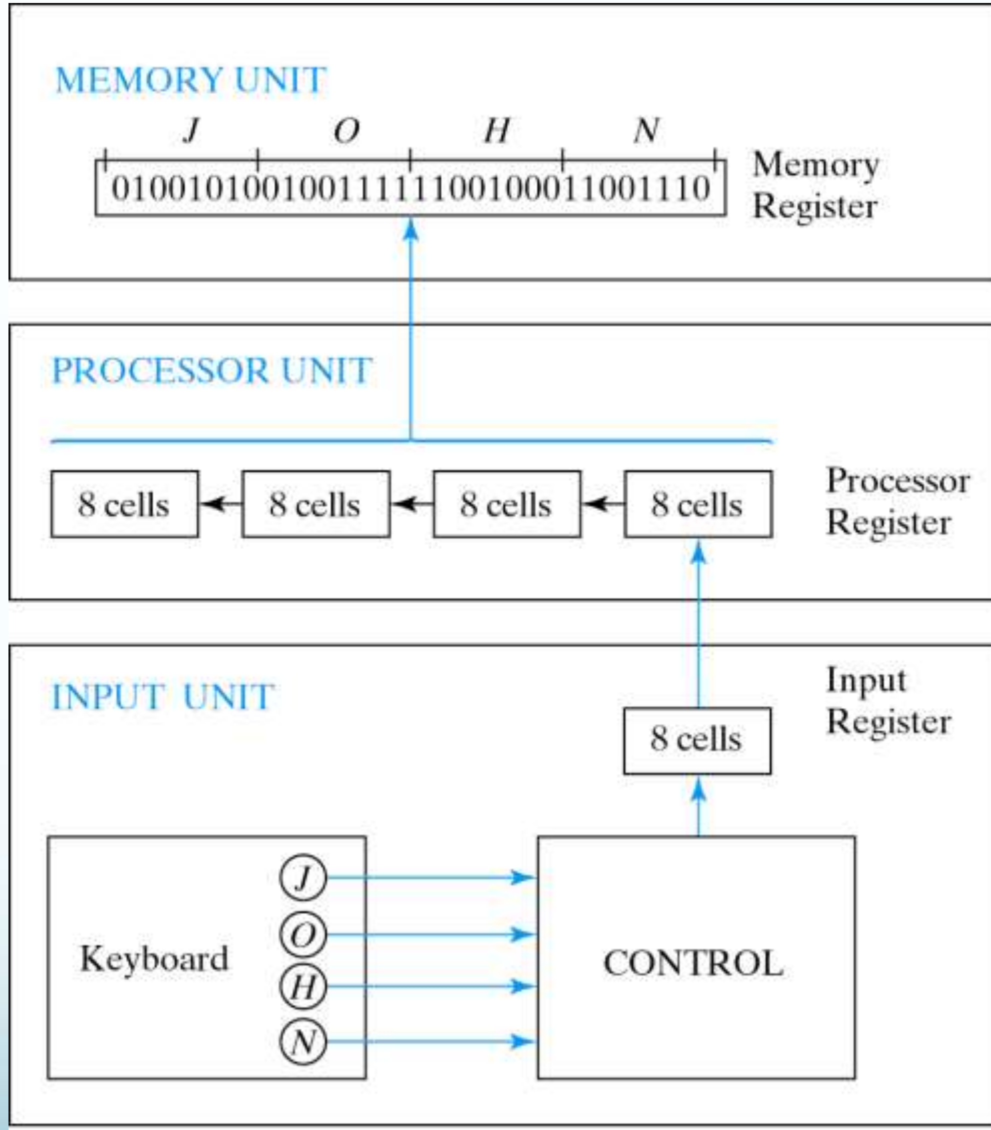


Fig. 1-1 Transfer of information with registers

Binary Information Processing

- The actual processing of binary information in a computer is completed by digital logic circuits which have been implemented to serve a specific purpose (i.e. addition).
  - The registers are accessed (read and write) when they are needed to complete an operation. For example we need two register sets to store two values to be added and a register set to store the result of the sum.
    - Furthermore, we need three registers in both the memory unit and in the processor.

Example Binary Information Processing

- We need processing
- We need storage
- We need communication

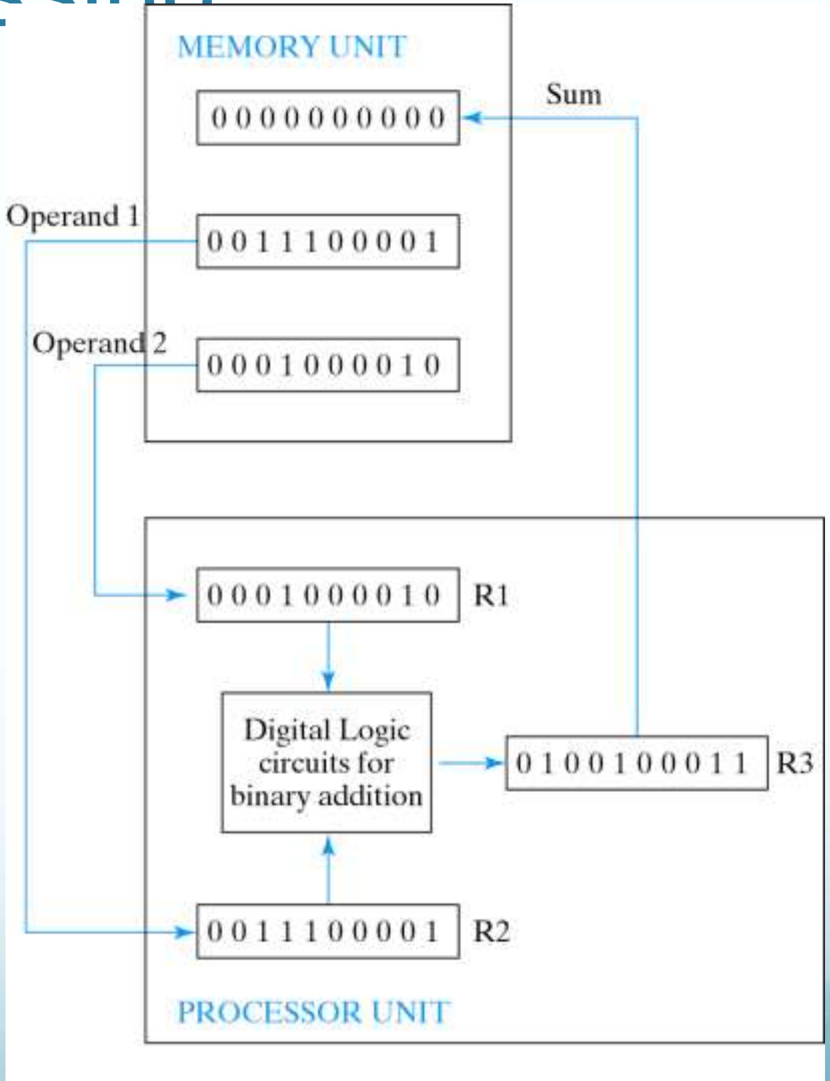


Fig. 1-2 Example of binary information processing

Binary Logic

- Binary logic consists of binary variables and logical operations.
  - The variables are designated by letters of the alphabet (A, B, C, x, y, z, etc.).
  - There are three basic logical operations:
    - AND
    - OR
    - NOT

Logical Operations

- AND is represented by a dot or the absence of an operator.
  - x·y = z or xy = z
  - Read as “x and y is equal to z”
  - Means that z=1 if and only if x=1 and y=1
- OR is represented by a plus sign.
  - x+y = z
  - Read as “x or y is equal to z”
  - Means that z=1 if x=1 or y=1 or both x=1 and y=1
- NOT is represented by a prime or an overbar.
  - x’=z or x = z
  - Read as not x is equal to z
  - Means that if x=1 then z=0 or if x=0 then z=1

Truth Tables

- Since each binary variable consists of value of 0 or 1, each combination of values for the variables involved in a binary operation has a specific result value.
- A truth table is a method of visualizing all possible combinations of the input values and the respective output values that occur due to the operation on the specified combination.

AND Truth Table

x	y	x·y
0	0	0
0	1	0
1	0	0
1	1	1

AND

A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

OR Truth Table

x	y	x+y
0	0	0
0	1	1
1	0	1
1	1	1

OR

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

NOT Truth Table

x	x'
0	1
1	0

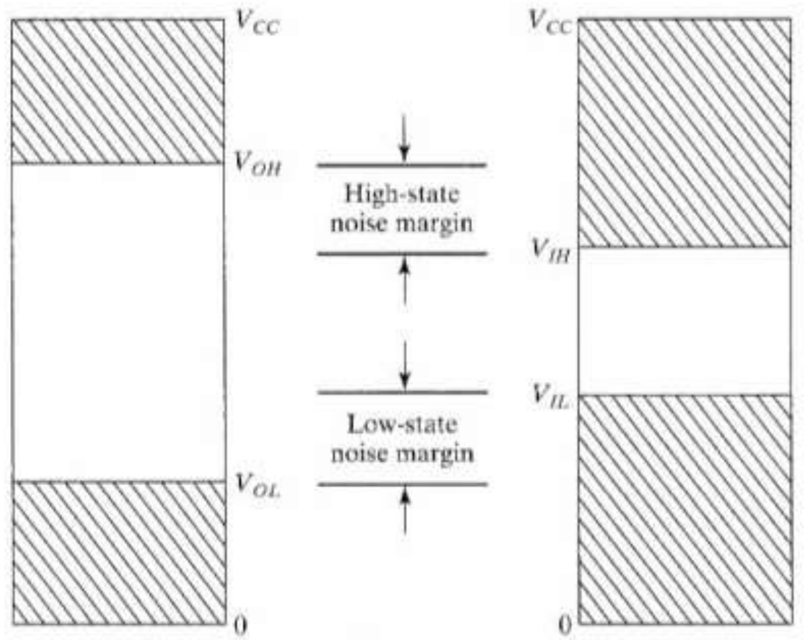
NOT

A	A'
0	1
1	0

Binary Signal

- Two separated voltage levels represents a binary variable equal to logic 1 or logic 0

- **noise margin in a standard TTL NAND gate are *VOH*= 2.4 V,** VoL = 0.4**V,** ***VIH***= 2**V,**and ***VIL***= **0.8V.**
- **The high-state noise margin is 2.4 - 2** = **0.4V, and the low-state noise margin is 0.8 - 0.4** = 0.**4V**

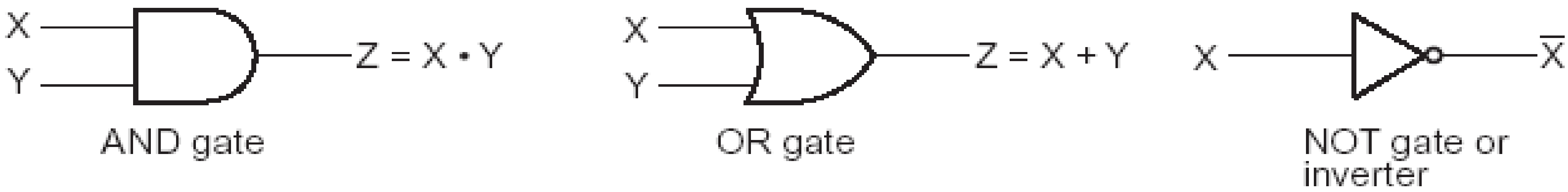


(a) Output voltage range
(b) Input voltage range

Logic Gates

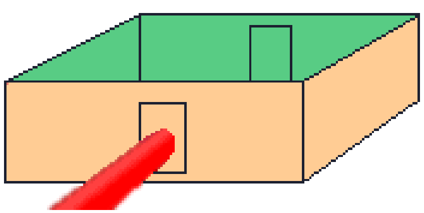
- Logic gates are electronic circuits that operate on one or more input signals to produce an output signal.
  - The state (high-low, on-off) of electricity on a line represents each of the two states for binary representation (1 or 0).

- Logic Gate Notation



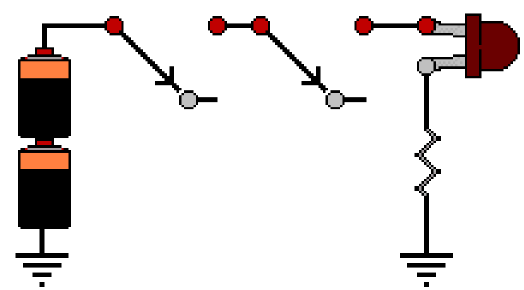
AND Logic Function

- Using door
  - Both doors are opened to pass the light



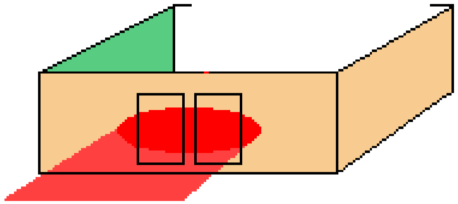
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

- **Using Switches**
  - **Switches are input and LED is output**
  - **Both switches closed (ON) to give output**



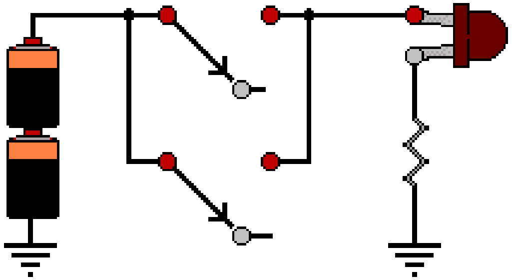
OR Logic Function

- Using door
  - Any one or both doors are opened to pass the light



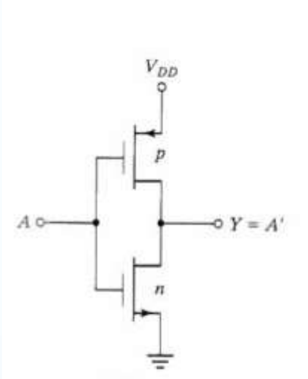
A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

- **Using Switches**
  - **Switches are input and LED is output**
  - **Any one switch or both closed to give output**



IC Digital Logic Families

- **RTL** Resistor-transistor Logic
- **DTL** Diode Transistor Logic
- **TTL** Transistor-Transistor logic
- **ECL** Emitter-Coupled Logic
- **MOS** Metal-Oxide Semiconductor
- **CMOS** Complementary Metal-Oxide Semiconductor



Note : More information see Chapter 10

Mathematical Systems

- Mathematical systems **can be defined with:**
  - **A set of** elements **containing a set of objects with common properties.**
  - **A set of** operators **that can be performed on any subset of the elements.**
  - **A set of** axioms **or** postulates **forming a basis from which we can deduce rules, theorems and properties of the system.**

Set Notations

- **The following notations are being used in this class:**
  - $x \in S$ **indicates that x is an element of the set S.**
  - $y \notin S$ **indicates that y is not an element of the set S.**
  - **A = {1, 2, 3, 4} indicates that set A exists with a finite number of elements (1, 2, 3, 4).**

Basic Postulates

- **The basic postulates of a mathematical system are:**
  - Closure. **A set S is closed w.r.t a binary operator if this operation only produces results that are within the set of elements defined by the system.**
  - Associative Law. **A binary operator is said to be associative when:**
    - » **(x * y) * z = x * (y * z)**
  - Commutative Law. **A binary operator is said to be commutative when:**
    - » **x * y = y * x**
  - Identity Element. **A set is said to have an identity element with respect to a binary operation if there exists an element, e, that is a member of the set with the property:**
    - » **e * x = x * e = x for every element of the set**
      - **Additive identity is 0 and multiplicative identity is 1**
  - **Note: * + and . are binary operators**

Basic Postulates

- Inverse. For a set with an identity element with respect to a binary operation, the set is said to have an inverse if for every element of the set the following property holds:
  - » x * y = e
    - The additive inverse of element a is –a and it defines subtraction, since a + (–a) = 0. Multiplicative inverse of a is 1/a and defines division, since a.1/a = 1
- Distributive Law. * is said to be distributive over . when
  - » x * (y · z) = (x * y) · (x * z)
- Note: * + and . are binary operators. Binary operator + defines addition and binary operator . defines multiplication

Two-Valued Boolean Algebra

- In 1854,George Boole developed an algebraic system now called *Boolean algebra*
- *In 1938* C. E,Shannon introduced a two-valued Boolean algebra called *switching algebra*
- Two-value Boolean algebra is defined by the:
  - The set of two elements B={0, 1}
  - The operators of AND (·) and OR (+)
  - Huntington Postulates are satisfied

Huntington Postulates

- **Boolean algebra has the following postulates:**
  1. Closure.
     - a) with respect to the binary operation OR (+)
     - b) with respect to the binary operation AND (·)
  2. Identity.
     - a) with respect to OR (+) is 0:
       - ❖ $x + 0 = 0 + x = x$, for $x = 1$ or $x = 0$
     - b) with respect to AND (·) is 1:
       - ❖ $x \cdot 1 = 1 \cdot x = x$, for $x = 1$ or $x = 0$
  3. Commutative Law.
     - a) With respect to OR (+):
       - ❖ $x + y = y + x$
     - b) With respect to AND (·):
       - ❖ $x \cdot y = y \cdot x$

Huntington Postulates Continued…

- **Boolean algebra has the following postulates:**
  4. Distributive Law.
     - **a) with respect to the binary operation OR (+):**
       - ❖ $x + (y \cdot z) = (x + y) \cdot (x + z)$ **+ is distributive over .**
     - **b) with respect to the binary operation AND (·):**
       - ❖ $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$ **. is distributive over +**
  5. Complement.
     - **a)** $x + x' = 1$, **for** $x = 1$ **or** $x = 0$
     - **b)** $x \cdot x' = 0$, **for** $x = 1$ **or** $x = 0$
  6. Membership.
     - ❖ **There exists at least two elements, x and y, of the set such that** $x \neq y$.
       - ❖ $0 \neq 1$

Notes on Huntington Postulates

- The associative law is not listed but it can be derived from the existing postulates for both operations.
- The distributive law of + over . i.e.,

  $$x+(y \, . \, z) = (x + y) \, . \, (x + z)$$

  is valid for Boolean algebra but not for ordinary algebra.
- Boolean algebra doesn’t have inverses (additive or multiplicative) therefore there are no operations related to subtraction or division.
- Boolean algebra deals with only two elements, 0 and 1

End of Lecture