# CSC 370 - Computer Organization and Architecture

Lecture 1 - Review of Boolean Algebra

# Digital Logic

- Digital logic hides the pitfalls of the analog world by mapping all physical values as sequences of 0s and 1s.
- Regardless of what type of digital circuit we use, 0 and 1 are represented by two ranges separated by an undefined range in between. These two ranges are called low and high respectively.

## Physical states representing 0 and 1

| | State Representing Bit | |
|---|---|---|
| **Technology** | **0** | **1** |
| Pneumatic logic | Fluid at low pressure | Fluid at high pressure |
| Relay logic | Circuit open | Circuit closed |
| CMOS logic | 0-1.5V | 3.5-5.0 V |
| TTL logic | 0-0.8V | 2.0-5.0V |
| Fiber optics | Light off | Light on |
| Dynamic Memory | Capacitor discharged | Capacitor charged |
| Magnetic tape or disk | Flux direction “north” | Flux direction “south” |
| CD-ROM | No pit | Pit |

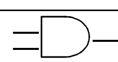
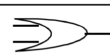
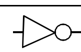
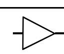
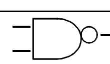
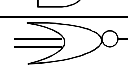
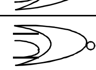
## Combinational Circuits

- A logic circuit whose output depends only on its current inputs is called a ***combinational circuit***. Its operation is fully described by a truth table that lists all possible combinations of inputs and the output values produced by each input set.

## Combinational Circuit Truth Table – An Example

| X | Y | Z | F |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

## Logic Gates

| Name | Graphic Symbol | Algebraic Function |
|---|---|---|
| AND |  | $x = A \circ B$ |
| OR |  | $x = A+B$ |
| Inverter |  | $x = A'$ |
| Buffer |  | $x = A$ |
| NAND |  | $x = (AB)'$ |
| NOR |  | $x = (A+B)'$ |
| XOR |  | $x = A \oplus B$ |
| Exclusive NOR |  | $x = (A \oplus B)'$ |

## Logic Gates and Their Truth Tables

| A | B | AB |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

| A | B | A+B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

## Logic Gates and Their Truth Tables (continued)

| A | A’ |
|---|---|
| 0 | 1 |
| 1 | 0 |

| A | x = A |
|---|---|
| 0 | 0 |
| 1 | 1 |

## Logic Gates and Their Truth Tables (continued)

| A | B | (AB)’ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

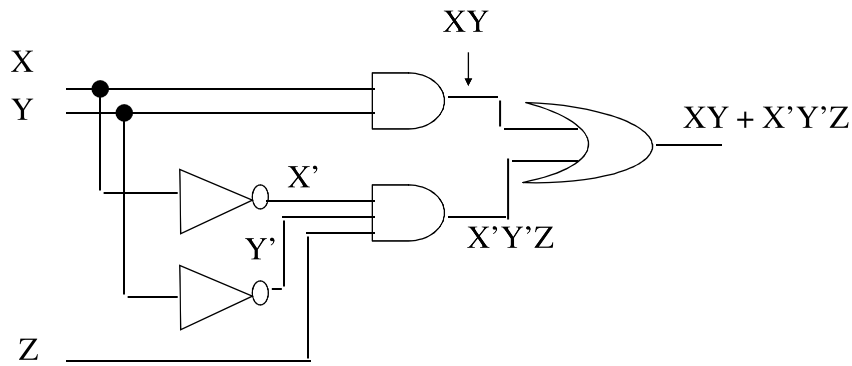
| A | B | (A+B)’ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

## Logic Gates and Their Truth Tables (continued)

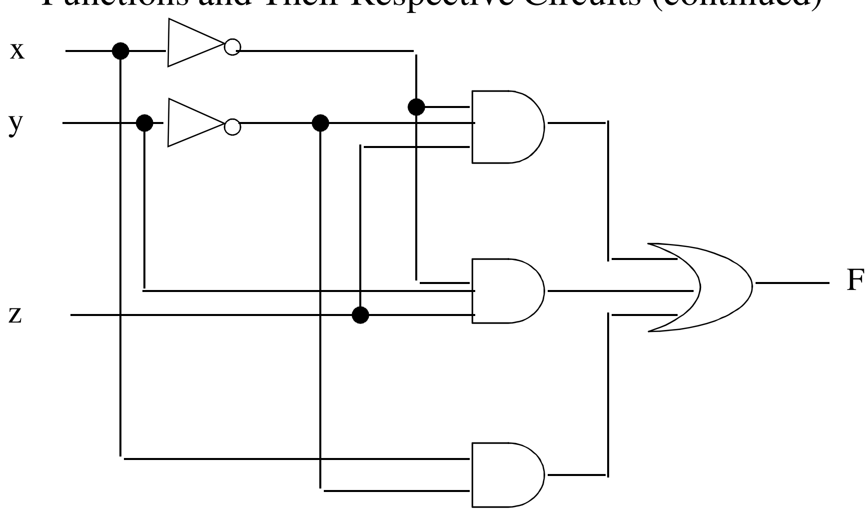
| A | B | A⊕B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

| A | B | (A⊕B)’ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

## Functions and Their Respective Circuits

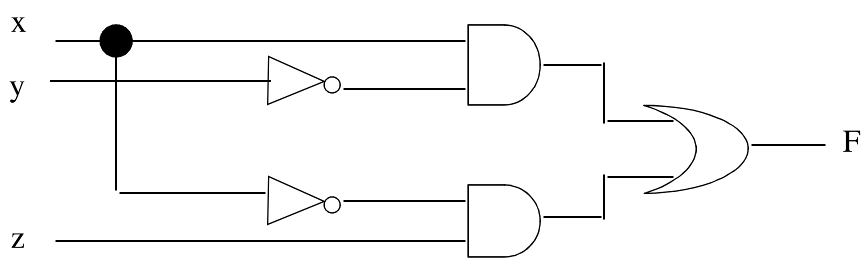



## Functions and Their Respective Circuits (continued)




F = x’y’z + x’yz + xy’

Functions and Their Respective Circuits (continued)




F = xy’ + x’z

# Types of Circuits

There are many ways to design an electronic logic circuit. These are among the most significant:

- TTL Logic
- MOS Logic
- CMOS Logic

# TTL Logic

- TTL (***T***ransistor-***t***ransistor ***l***ogic) is the most successful family of bipolar logic circuit designs.
  - Bipolar logic circuits have junctions where positively “doped” semiconductors meet negatively “doped” semiconductors.
- First introduced in the 1960s, TTL is now a family of logic families that are compatible with each other but differ in speed, power consumption and cost.
- TTL was largely replaced by CMOS in the 1990s.

# MOS Logic

- In MOS (***M***etal ***O***xide ***S***emiconductor) logic, increasing the voltage decreases the effective resistance of the transistor.
- It was not until the 1960s that fabrication methods were practical enough for manufacturing.
- MOS was significantly slower than TTL but used much less power.

## CMOS Logic

- CMOS (Complementary Metal Oxide Semiconductor) Logic is an improved variation on MOS logic and is commonly used now in large-scale integrated circuits.
- CMOS logic is the most capable and the easiest to understand commercial digital logic technology.

## Boolean Algebra

- ***Boolean algebra*** (named for British mathematician George Boole) is the algebra of logical values (***true*** and ***false***).
- Boolean algebra gives us postulates and theorems that provides ways for us to simplify logic expressions and therefore come up with simpler circuits that perform the same function as the ones with which we started.

## Fundamental Concepts of Boolean Algebra

- Boolean algebra uses the + sign to indicate the logical OR operation.

  0 + 0 = 0 0 + 1 = 1

  1 + 0 = 1 1 + 1 = 1

- Boolean algebra uses the • indicate the logical AND operation.

  0 • 0 = 0 0 • 1 = 0

  1 • 0 = 0 1 • 1 = 1

- Complementation is taken as:

  1’ = 0 0’ = 1

## Boolean Algebra and Truth Tables

- Truth tables are used in the evaluation of logical functions:

| X | Y | Z | Z’ | YZ’ | X+YZ’ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 |

## Minterms and Maxterms

- A truth table must account for every combination of independent logical variables.
  - These combination are called ***minterms***.
  - If there are n independent variables, there will be $2^n$ minterms.
- Minterms are written as the product of independent variables or their complements.
  - We can also write them as the sum of the independent variables or their complements. These are called ***maxterms***.
  - For every minterm, there is a corresponding maxterm.

## Minterms and Maxterms For 3 Variables

| **x** | **y** | **z** | **Term** | **Designation** | **Term** | **Designation** |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | x'y'z' | $m_0$ | x+y+z | $M_0$ |
| 0 | 0 | 1 | x'y'z | $m_1$ | x+y+z' | $M_1$ |
| 0 | 1 | 0 | x'yz' | $m_2$ | x+y'+z | $M_2$ |
| 0 | 1 | 1 | x'yz | $m_3$ | x+y'+z' | $M_3$ |
| 1 | 0 | 0 | xy'z' | $m_4$ | x'+y+z | $M_4$ |
| 1 | 0 | 1 | xy'z | $m_5$ | x'+y+z' | $M_5$ |
| 1 | 1 | 0 | xyz' | $m_6$ | x'+y'+z | $M_6$ |
| 1 | 1 | 1 | xyz | $m_7$ | x'+y'+z' | $M_7$ |

## Truth Tables for 16 2-Variable Function

| **x** | **y** | $F_0$ | $F_1$ | $F_2$ | $F_3$ | $F_4$ | $F_5$ | $F_6$ | $F_7$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

## Truth Tables for 16 2-Variable Function (continued)

| **x** | **y** | $F_8$ | $F_9$ | $F_{10}$ | $F_{11}$ | $F_{12}$ | $F_{13}$ | $F_{14}$ | $F_{15}$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

## Sixteen Logic Microoperations

| **Boolean Function** | **Name** |
|---|---|
| $F_0 = 0$ | Clear |
| $F_1 = xy$ | AND |
| $F_2 = xy'$ | |
| $F_3 = x$ | Transfer A |
| $F_4 = x'y$ | |
| $F_5 = y$ | Transfer B |
| $F_6 = x \oplus y$ | Exclusive-OR |
| $F_7 = x + y$ | OR |

## Sixteen Logic Microoperations (continued)

| Boolean Function | Name |
|---|---|
| $F_8 = (x+y)'$ | NOR |
| $F_9 = (x \oplus y)'$ | Exclusive-NOR |
| $F_{10} = y'$ | Complement B |
| $F_{11} = x + y'$ | |
| $F_{12} = x'$ | Complement A |
| $F_{13} = x' + y$ | |
| $F_{14} = (xy)'$ | NAND |
| $F_{15} = 1$ | Set |

## The Axioms of Boolean Algebra

| | |
|---|---|
| x + 0 = x | x • 1 = x |
| x + x' = 1 | x • x' = 0 |
| x + x = x | x • x = x |
| x + 1 = 1 | x • 0 = 0 |
| (x')' = x | |
| x + y = y + x | xy = yx |
| x + (y+z) = (x+y) + z | x(yz) = (xy)z |
| x(y + z) = xy + xz | x + yz = (x+y)(x+z) |
| (x+y)' = x'y' | (xy)' = x' + y' |
| x + xy = x | x(x+y) = x |
| x + x'y = x + y | x(x'+y) = xy |

## Simplifying Logical Expressions With Boolean Algebra

$$
\begin{aligned}
xyz + x'y + xyz' &= xyz + xyz' + x'y \\
&= xy(z + z') + x'y \\
&= xy \bullet 1 + x'y \\
&= xy + x'y \\
&= (x + x') \bullet y \\
&= 1 \bullet y \\
&= y
\end{aligned}
$$

## Simplifying Logical Expressions With Boolean Algebra (continued)

$y(wz' + wz) + xy \quad = yw(z' + z) + xy$

$= yw + xy$

$= wy + xy$

$= (w + x)\,y$

## Simplifying Logical Expressions With Boolean Algebra (continued)

$xy(x'yz' + xy'z' + x'y'z')$

$= xy\,(x'yz' + x'y'z' + xy'z')$

$= xy\,(x'z'(y+y') + xy'z')$

$= xy(x'z' + xy'z') \quad = \; xy(x'+xy')z'$

$= xy(x' + y')z' \quad = xx'yz' + xyy'z'$

$= \mathbf{0}$

## Simplifying Logical Expressions With Boolean Algebra (continued)

$AB + AB' + A'C + A'C'$

$= A\,(B+B') + A'(C+C')$

$= A + A'$

$= \mathbf{1}$

## Simplifying Logical Expressions With Boolean Algebra (continued)

$(x'+y)(x+z)(y+z)$

$= (x' + y)(xy + z)$

$= x'xy + x'z + yxy + yz$

$= 0 + x'z + yxy + yz$

$= x'z + xyy + yz$

$= x'z + xy + yz$

$= xy + x'z + yz$