NAAC ACCREDITED "B++" (CGPA 2.89)

Yogoda Satsanga Mahavidyalaya

Jagannathpur, Dhurwa, Ranchi-834004

www.ysmranchi.net

ysmranchi4@gmail.com

Course :Computer System Architecture

Class : Sem-1

Lesson :Logic Simplification

By :Goutam Sanyal

This Video is an Intellectual Property of Yogoda Satsanga Mahavidyalaya, Dhurwa, Ranchi, Jharkhand

Digital Logic (Review)

Binary Logic

- Deals with binary variables that take 2 discrete values (0 and 1), and with logic operations
- Three basic logic operations:
  - AND, OR, NOT
- Binary/logic variables are typically represented as letters: A,B,C,…,X,Y,Z

Binary Logic Function

F(vars) = *expression*

set of binary variables

- Operators (+, •, ‘)
- Variables
- Constants (0, 1)
- Groupings (parenthesis)

Example: F(a,b) = a’•b + b’

G(x,y,z) = x•(y+z’)

Basic Logic Operators

- AND
- OR

Binary

- NOT

Unary

- F(a,b) = a•b, F is 1 if and only if a=b=1
- G(a,b) = a+b, G is 1 if either a=1 or b=1
- H(a) = a’, H is 1 if a=0

Basic Logic Operators (cont.)

- 1-bit logic AND resembles binary multiplication:

  $0 \bullet 0 = 0, \quad 0 \bullet 1 = 0,$

  $1 \bullet 0 = 0, \quad 1 \bullet 1 = 1$

- 1-bit logic OR resembles binary addition, except for one operation:

  $0 + 0 = 0, \quad 0 + 1 = 1,$

  $1 + 0 = 1, \quad 1 + 1 = 1 \ (\neq 10_2)$

Truth Tables for logic operators

Truth table: tabular form that uniquely represents the relationship between the input variables of a function and its output

2-Input AND

A	B	F=A•B
0	0	0
0	1	0
1	0	0
1	1	1

2-Input OR

A	B	F=A+B
0	0	0
0	1	1
1	0	1
1	1	1

NOT

A	F=A’
0	1
1	0

Truth Tables (cont.)

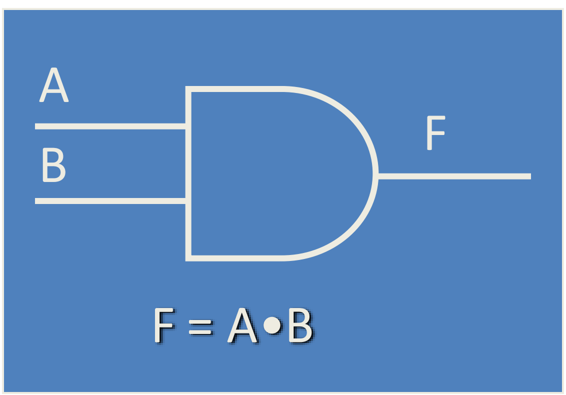
- Q: Let a function F() depend on *n* variables. How many rows are there in the truth table of F() ?

- **A:** 2^n rows, since there are 2^n possible binary patterns/combinations for the *n* variables

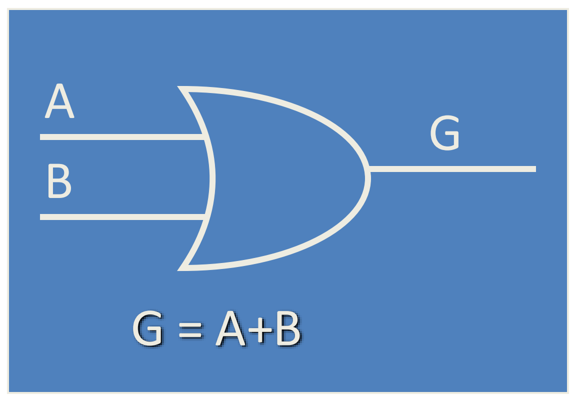
Logic Gates

- Logic gates are abstractions of electronic circuit components that operate on one or more input signals to produce an output signal.

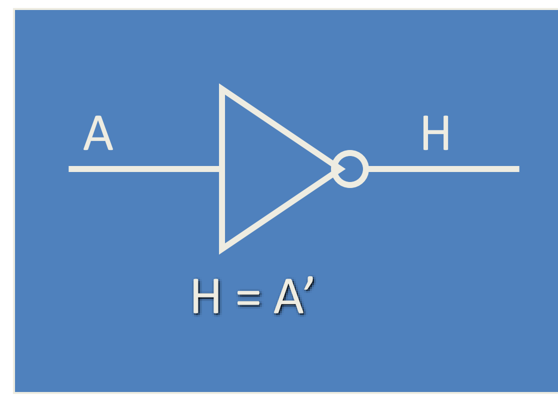
2-Input AND



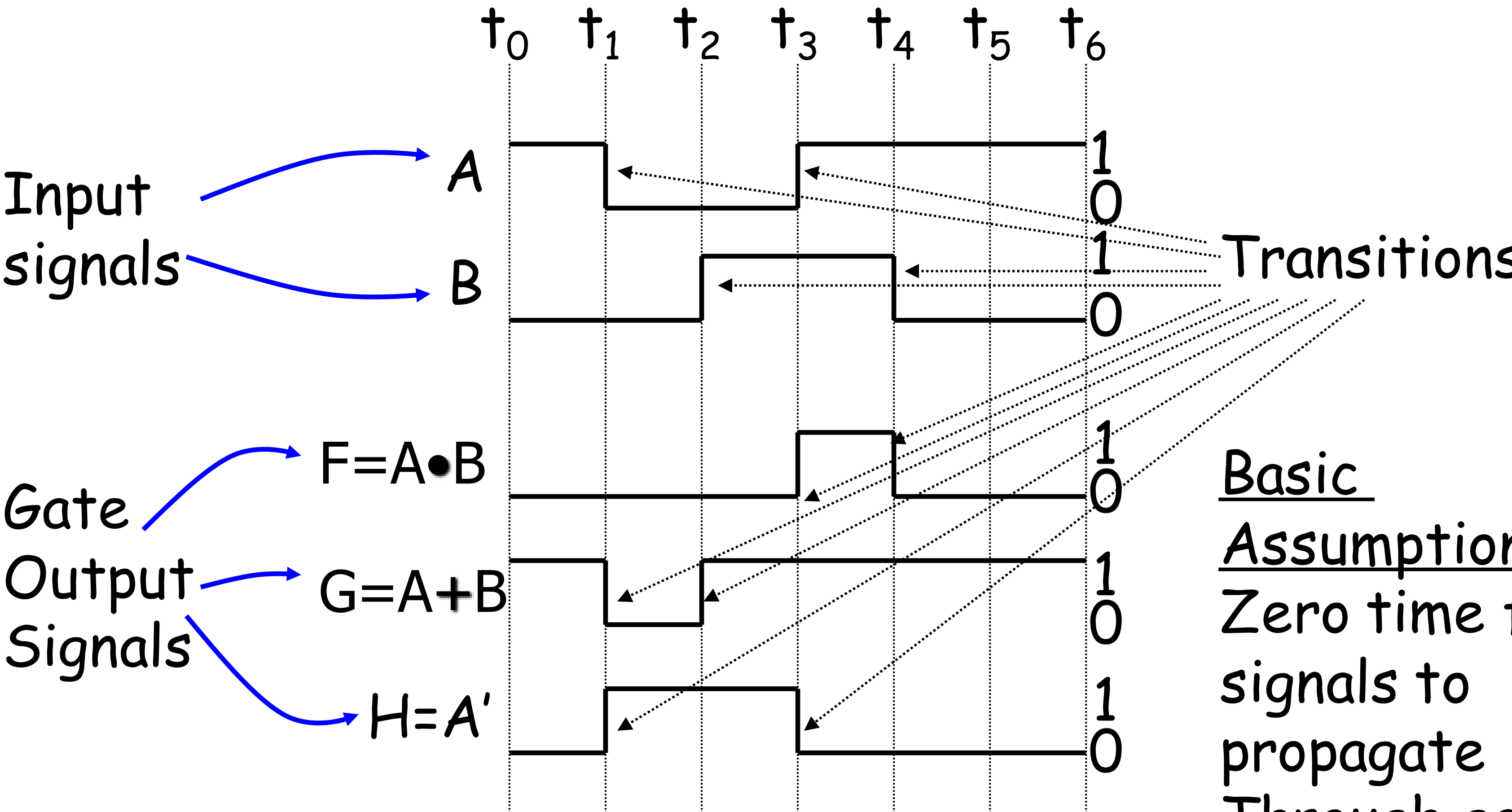
2-Input OR



NOT (Inverter)



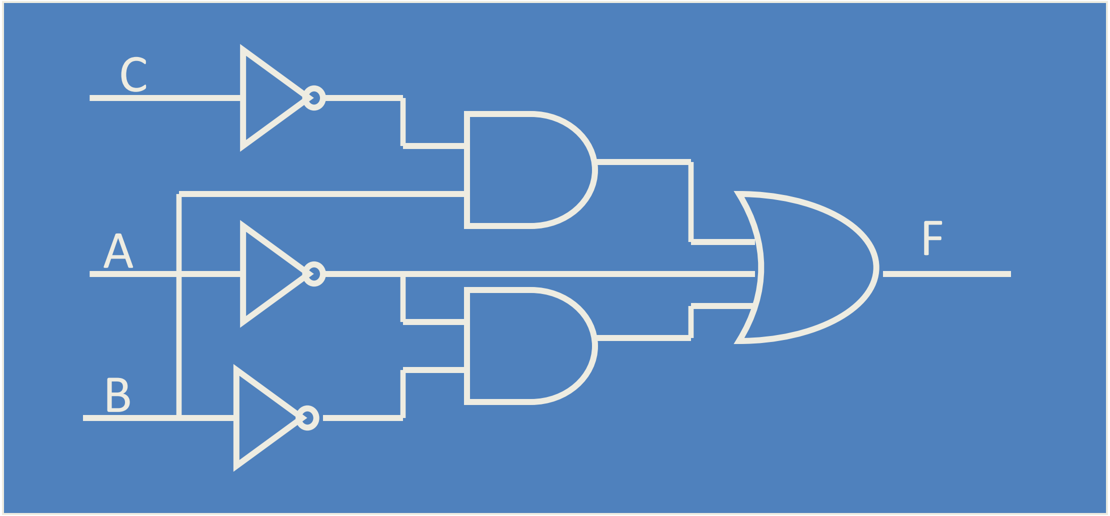
Timing Diagram



Basic ***Assumption:***
Zero time for signals to propagate Through gates

Combinational Logic Circuit from Logic Function

- Consider function F = A’ + B•C’ + A’•B’
- A combinational logic circuit can be constructed to implement F, by appropriately connecting input signals and logic gates:
  - Circuit input signals → from function variables (A, B, C)
  - Circuit output signal → function output (F)
  - Logic gates → from logic operations

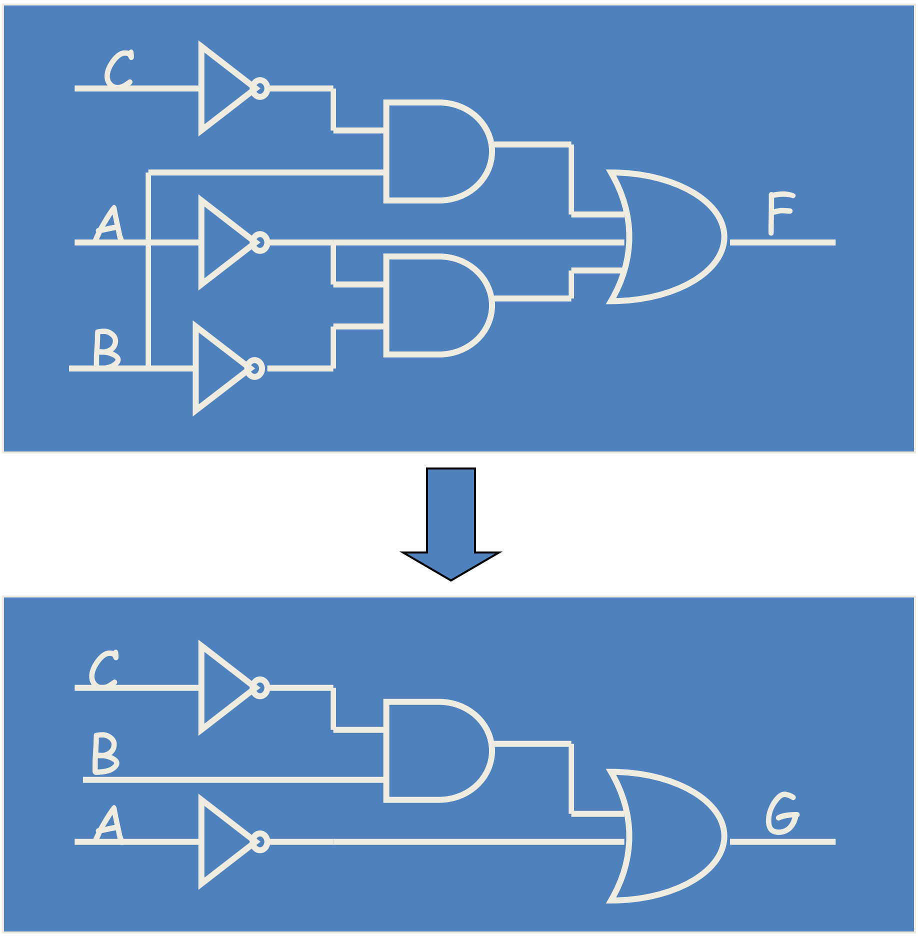


Combinational Logic Circuit from Logic Function (cont.)

- In order to design a cost-effective and efficient circuit, we must minimize the circuit's size (area) and propagation delay (time required for an input signal change to be observed at the output line)
- Observe the truth table of F=A' + B•C' + A'•B' and G=A' + B•C'
- Truth tables for F and G are identical → same function
- Use G to implement the logic circuit (less components)

A	B	C	F	G
0	0	0	1	1
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	0	0	0
1	0	1	0	0
1	1	0	1	1
1	1	1	0	0

Combinational Logic Circuit from Logic Function (cont.)



Boolean Algebra

- VERY nice machinery used to manipulate (simplify) Boolean functions
- George Boole (1815-1864): “An investigation of the laws of thought”
- Terminology:
  - *Literal:* A variable or its complement
  - *Product term:* literals connected by •
  - *Sum term:* literals connected by +

Boolean Algebra Properties

Let X: boolean variable, 0,1: constants

1. $X + 0 = X$ -- Zero Axiom
2. $X \bullet 1 = X$ -- Unit Axiom
3. $X + 1 = 1$ -- Unit Property
4. $X \bullet 0 = 0$ -- Zero Property

Boolean Algebra Properties (cont.)

Let X: boolean variable, 0,1: constants

5. X + X = X -- Idempotence
6. X • X = X -- Idempotence
7. X + X’ = 1 -- Complement
8. X • X’ = 0 -- Complement
9. (X’)’ = X -- Involution

Duality

- The dual of an expression is obtained by exchanging (• and +), and (1 and 0) in it, provided that the precedence of operations is not changed.
- Cannot exchange x with x’
- Example:
  - Find H(x,y,z), the dual of F(x,y,z) = x’yz’ + x’y’z
  - H = (x’+y+z’) (x’+y’+ z)

Duality (cont'd)

With respect to duality, Identities 1 – 8 have the following relationship:

1. X + 0 = X	2. X • 1 = X (dual of 1)
3. X + 1 = 1	4. X • 0 = 0 (dual of 3)
5. X + X = X	6. X • X = X (dual of 5)
7. X + X' = 1	8. X • X' = 0 (dual of 8)

More Boolean Algebra Properties

Let X,Y, and Z: boolean variables

10. $X + Y = Y + X$ 11. $X \bullet Y = Y \bullet X$ -- Commutative

12. $X + (Y+Z) = (X+Y) + Z$ 13. $X \bullet (Y \bullet Z) = (X \bullet Y) \bullet Z$ -- Associative

14. $X \bullet (Y+Z) = X \bullet Y + X \bullet Z$ 15. $X+(Y \bullet Z) = (X+Y) \bullet (X+Z)$ -- Distributive

16. $(X + Y)' = X' \bullet Y'$ 17. $(X \bullet Y)' = X' + Y'$ -- DeMorgan's

In general,

$(X_1 + X_2 + \ldots + X_n)' = X_1' \bullet X_2' \bullet \ldots \bullet X_n'$, and

$(X_1 \bullet X_2 \bullet \ldots \bullet X_n)' = X_1' + X_2' + \ldots + X_n'$

Absorption Property

1. $x + x \bullet y = x$
2. $x \bullet (x+y) = x$ (dual)

- **Proof:**

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

QED (2 true by duality, why?)

Power of Duality

1. $x + x \bullet y = x$ is true, so $(x + x \bullet y)' = x'$
2. $(x + x \bullet y)' = x' \bullet (x' + y')$
3. $x' \bullet (x' + y') = x'$
4. Let $X = x'$, $Y = y'$
5. $X \bullet (X + Y) = X$, which is the dual of $x + x \bullet y = x$.
6. The above process can be applied to any formula. So if a formula is valid, then its dual must also be valid.
7. Proving one formula also proves its dual.

Consensus Theorem

1. $xy + x'z + yz = xy + x'z$
2. $(x+y) \bullet (x'+z) \bullet (y+z) = (x+y) \bullet (x'+z)$ -- (dual)

- **Proof:**

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

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

  $= (xy + xyz) + (x'z + x'zy)$

  $= xy + x'z$

  QED (2 true by duality).

Truth Tables (revisited)

- Enumerates all possible combinations of variable values and the corresponding function value
- Truth tables for some arbitrary functions $F_1(x,y,z)$, $F_2(x,y,z)$, and $F_3(x,y,z)$ are shown to the right.

x	y	z		F_1	F_2	F_3
0	0	0		0	1	1
0	0	1		0	0	1
0	1	0		0	0	1
0	1	1		0	1	1
1	0	0		0	1	0
1	0	1		0	1	0
1	1	0		0	0	0
1	1	1		1	0	1

Truth Tables (cont.)

- Truth table: a unique representation of a Boolean function
- If two functions have identical truth tables, the functions are equivalent (and vice-versa).
- Truth tables can be used to prove equality theorems.
- However, the size of a truth table grows exponentially with the number of variables involved, hence unwieldy. This motivates the use of Boolean Algebra.

Boolean expressions-NOT unique

- Unlike truth tables, expressions representing a Boolean function are NOT unique.
- Example:
  - $F(x,y,z) = x' \bullet y' \bullet z' + x' \bullet y \bullet z' + x \bullet y \bullet z'$
  - $G(x,y,z) = x' \bullet y' \bullet z' + y \bullet z'$
- The corresponding truth tables for F() and G() are to the right. They are identical.
- Thus, F() = G()

x	y	z		F	G
0	0	0		1	1
0	0	1		0	0
0	1	0		1	1
0	1	1		0	0
1	0	0		0	0
1	0	1		0	0
1	1	0		1	1
1	1	1		0	0

Algebraic Manipulation

- Boolean algebra is a useful tool for simplifying digital circuits.
- Why do it? Simpler can mean cheaper, smaller, faster.
- Example: Simplify F = x'yz + x'yz' + xz.

$$
\begin{aligned}
F &= x'yz + x'yz' + xz \\
&= x'y(z+z') + xz \\
&= x'y\bullet 1 + xz \\
&= x'y + xz
\end{aligned}
$$

Algebraic Manipulation (cont.)

- Example: Prove

  $x'y'z' + x'yz' + xyz' = x'z' + yz'$

- **Proof:**

  $x'y'z' + x'yz' + xyz'$

  $= x'y'z' + x'yz' + x'yz' + xyz'$

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

  $= x'z' \bullet 1 + yz' \bullet 1$

  $= x'z' + yz'$

  QED.

Complement of a Function

- The complement of a function is derived by interchanging (• and +), and (1 and 0), and complementing each variable.
- Otherwise, interchange 1s to 0s in the truth table column showing F.
- The *complement* of a function IS NOT THE SAME as the *dual* of a function.

Complementation: Example

- Find G(x,y,z), the complement of
  F(x,y,z) = xy'z' + x'yz

- G = F' = (xy'z' + x'yz)'
  = (xy'z')' • (x'yz)' *DeMorgan*
  = (x'+y+z) • (x+y'+z') *DeMorgan* again

- Note: The complement of a function can also be derived by finding the function's *dual,* and then complementing all of the literals

Summary