CSE1003-Digital Logic Design

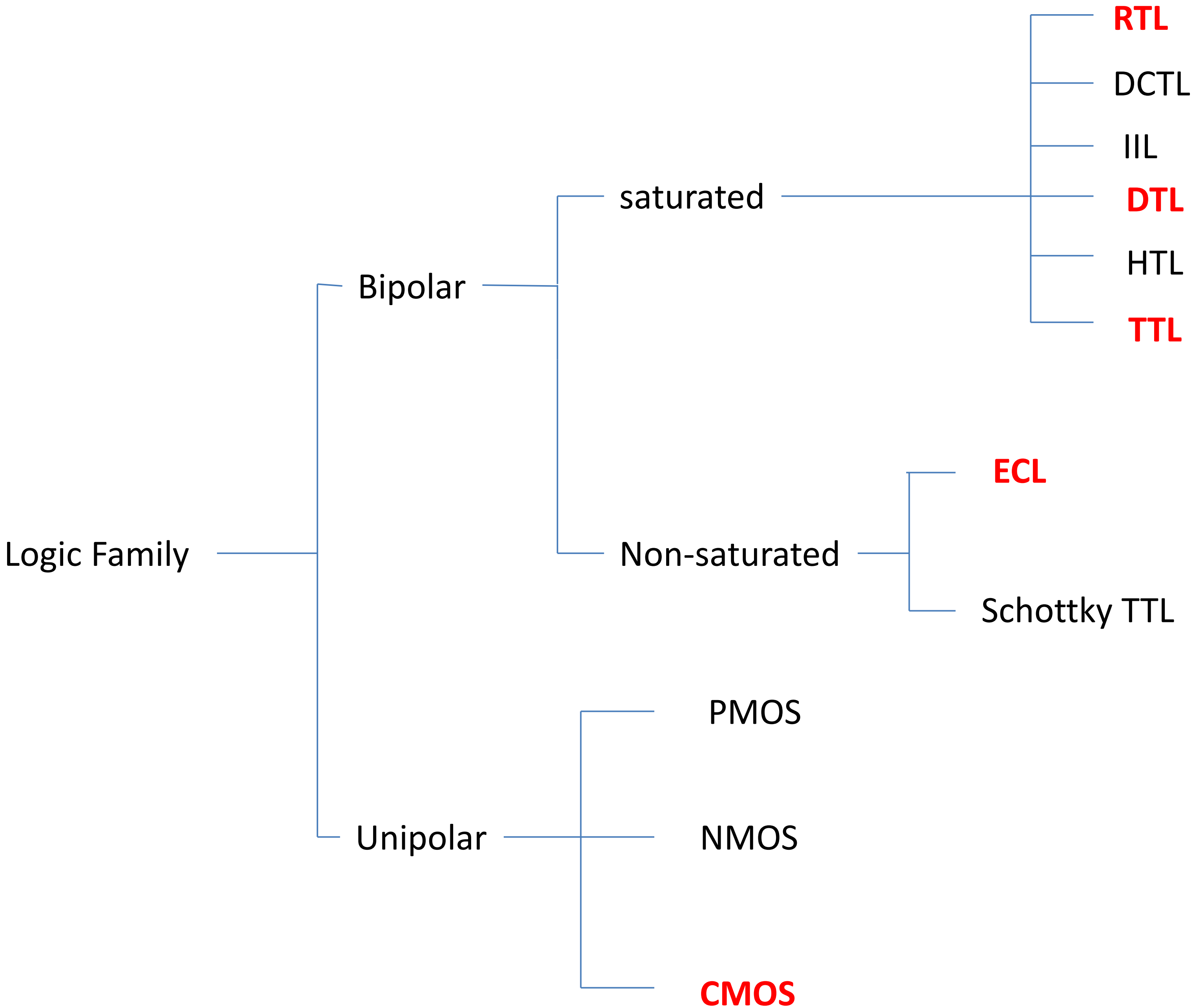
Dr.Penchalaiah Palla

Dept. of Micro and Nanoelectronics

School of Electronics Engineering,VIT, Vellore

Module:2	BOOLEAN ALGEBRA	8 hours
Boolean algebra - Properties of Boolean algebra - Boolean functions - Canonical and Standard forms - Logic gates - Universal gates – Karnaugh map - Don"t care conditions - Tabulation Method		

Logic Families

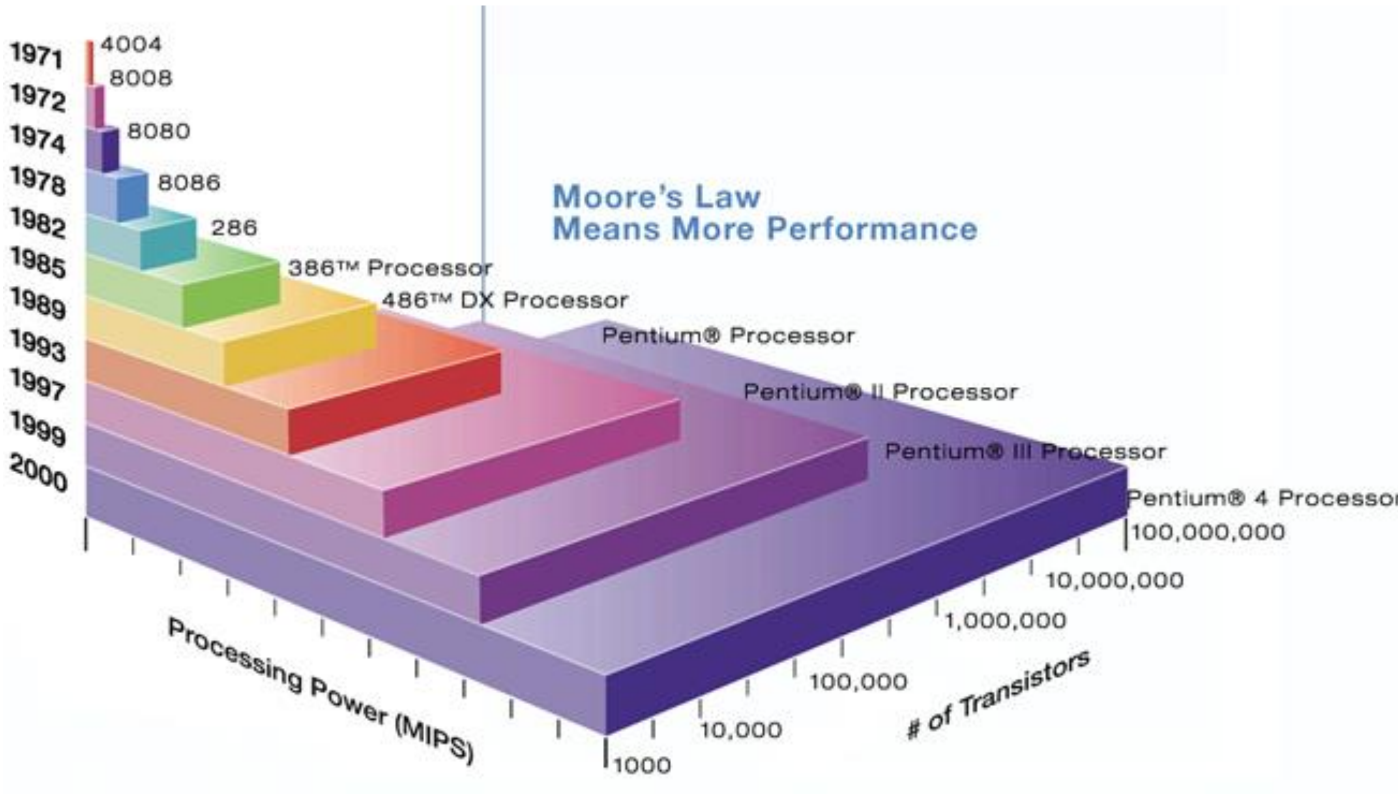


Integration Levels

- Gate/transistor ratio is roughly 1/10
  - SSI < 12 gates/chip
  - MSI < 100 gates/chip
  - LSI …1K gates/chip
  - VLSI …10K gates/chip
  - ULSI …100K gates/chip
  - GSI …1Meg gates/chip

Moore's law

- A prediction made by Moore (a co-founder of Intel) in 1965: "… a number of transistors to double every 2 years."

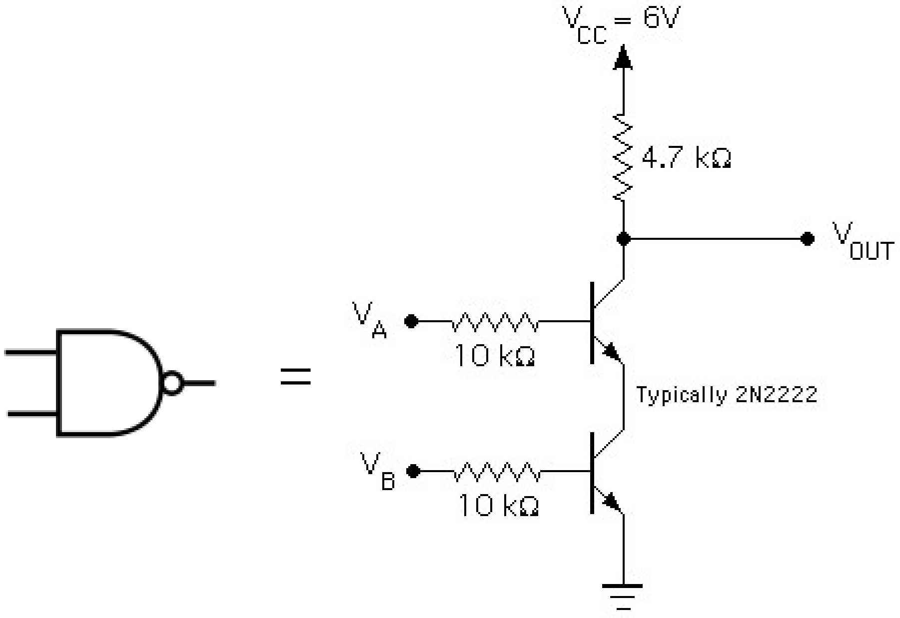


Characteristics of Logic Families

1. Speed
2. Fan-out
3. Fan-in
4. Power dissipation
5. Propagation delay
6. Noise Margin
7. Figure of Merit = propagation delay X Power dissipation
8. Logic Swing ($V_{OH} – V_{OL}$)
9. Breadth : No. of functions the we can take from the circuit

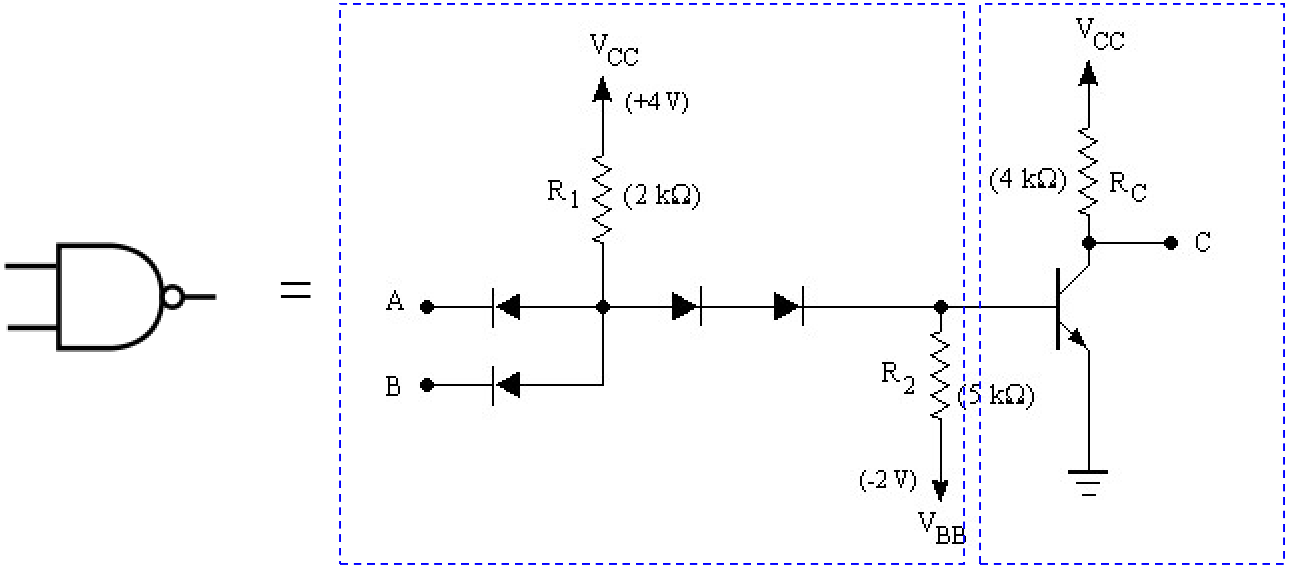
Resistor-Transistor Logic (RTL)

• replace diode switch with a transistor switch
• can be cascaded
• large power draw



Diode-Transistor Logic (DTL)

- essentially diode logic with transistor amplification
- reduced power consumption
- faster than RTL



DL AND gate

Saturating inverter

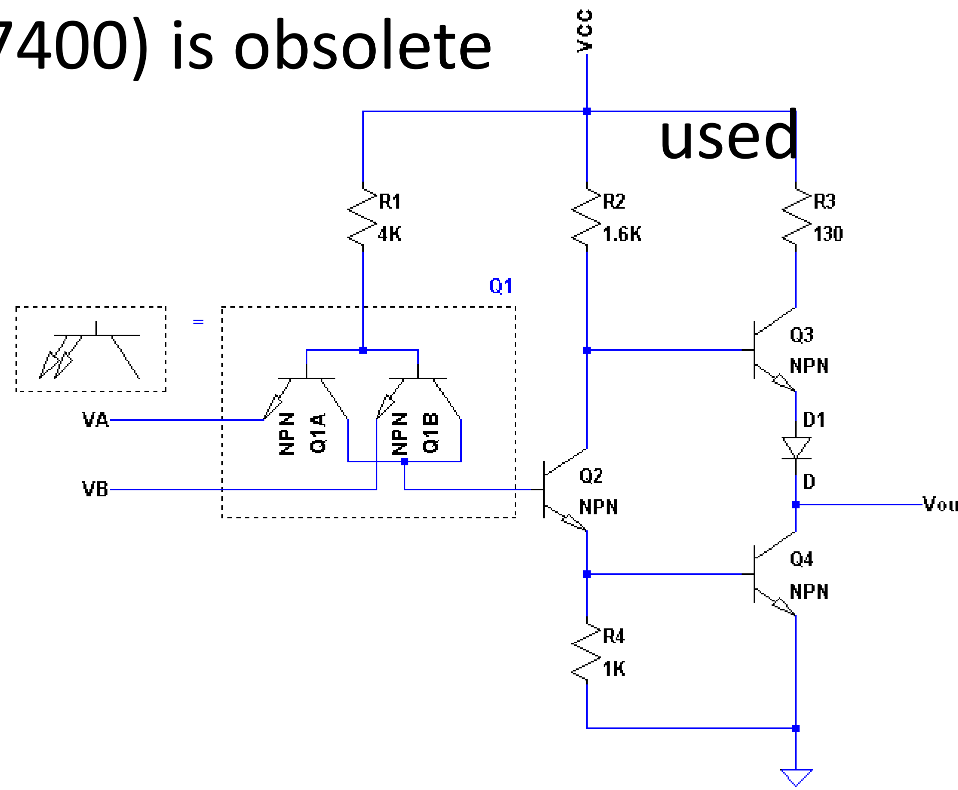
TTL

Bipolar Transistor-Transistor Logic (TTL)

- first introduced by in 1964 (Texas Instruments)
- TTL has shaped digital technology in many ways
- Standard TTL family (e.g. 7400) is obsolete
- Newer TTL families still used (e.g. 74ALS00)

Distinct features

- Multi-emitter transistors
- Totem-pole transistor arrangement



2-input NAND

Emitter-Coupled Logic (ECL)

- <u>PROS</u>: Fastest logic family available (~1ns)

- <u>CONS</u>: low noise margin and high power dissipation

- Operated in emitter coupled geometry (recall differential amplifier or emitter-follower), transistors are biased and operate near their Q-point (never near saturation!)
- Logic levels. “0”: –1.7V. “1”: –0.8V
- Such strange logic levels require extra effort when interfacing to TTL/CMOS logic families.

Boolean Algebra

In ordinary algebra, the letter symbols take any number of values. In Boolean algebra, they take two values, i.e. 0 and 1. ... The values assigned to a variable have a numerical significance in ordinary algebra, whereas in Boolean algebra they have a logical significance.

The “WHY” slide

- Boolean Algebra
  - When we learned numbers like 1, 2, 3, we also then learned how to add, multiply, etc. with them. Boolean Algebra covers operations that we can do with 0’s and 1’s. Computers do these operations ALL THE TIME and they are basic building blocks of computation inside your computer program.
- Axioms, laws, theorems
  - We need to know some rules about how those 0’s and 1’s can be operated on together. There are similar axioms to decimal number algebra, and there are some laws and theorems that are good for you to use to simplify your operation.

a statement or proposition which is regarded as being established, accepted, or self-evidently true."the **axiom that** sport builds character"

How does Boolean Algebra fit into the big picture?

- It is part of the Combinational Logic topics (memoryless)
  - Different from the Sequential logic topics (can store information)
- Learning Axioms and theorems of Boolean algebra
    - Allows you to design logic functions
    - Allows you to know how to combine different logic gates
    - Allows you to simplify or optimize on the complex operations

Basic Definitions

- Binary Operators
  - AND

    $z = x \bullet y = x\,y$ $\quad$ $z=1$ *if* $x=1$ *AND* $y=1$
  - OR

    $z = x + y$ $\quad$ $z=1$ *if* $x=1$ *OR* $y=1$
  - NOT

    $z = \overline{x} = x'$ $\quad$ $z=1$ *if* $x=0$
- Boolean Algebra
  - Binary Variables: only ‘0’ and ‘1’ values
  - Algebraic Manipulation

Boolean Algebra Postulates

- Commutative Law

  $x \bullet y = y \bullet x \qquad x + y = y + x$

- Identity Element

  $x \bullet 1 = x \qquad x + 0 = x$

- Complement

  $x \bullet x' = 0 \qquad x + x' = 1$

Boolean Algebra Theorems

- Duality
  - The *dual* of a Boolean algebraic expression is obtained by interchanging the AND & OR operators and replacing the 1's by 0's and the 0's by 1's.
  - $x \bullet (y + z) = (x \bullet y) + (x \bullet z)$
  - $x + (y \bullet z) = (x + y) \bullet (x + z)$

Applied to a valid equation produces a valid equation

- Theorem 1
  - $x \bullet x = x$ $\quad\quad$ $x + x = x$
- Theorem 2
  - $x \bullet 0 = 0$ $\quad\quad$ $x + 1 = 1$

Boolean Algebra Theorems

- Theorem 3: *Involution*
  - $(x')' = x$ $\qquad (\overline{\overline{x}}) = x$
- Theorem 4: *Associative & Distributive*
  - $(x \bullet y) \bullet z = x \bullet (y \bullet z)$ $\qquad (x + y) + z = x + (y + z)$
  - $x \bullet (y + z) = (x \bullet y) + (x \bullet z)$

    $x + (y \bullet z) = (x + y) \bullet (x + z)$
- Theorem 5: *DeMorgan*
  - $(x \bullet y)' = x' + y'$ $\qquad (x + y)' = x' \bullet y'$
  - $(\overline{x \bullet y}) = \overline{x} + \overline{y}$ $\qquad (\overline{x + y}) = \overline{x} \bullet \overline{y}$
- Theorem 6: *Absorption*
  - $x \bullet (x + y) = x$ $\qquad x + (x \bullet y) = x$

Operator Precedence

- Parentheses

  $(\ldots) \bullet (\ldots)$

- NOT

  $x' + y$

- AND

  $x + x \bullet y$

- OR

$$x\left[y + z\,\overline{(w + x)}\right]$$

$$(w + x)$$

$$\overline{(w + x)}$$

$$z\,\overline{(w + x)}$$

$$y + z\,\overline{(w + x)}$$

$$x\left[y + z\,\overline{(w + x)}\right]$$

DeMorgan's Theorem

$$\overline{a\,[b + c\,(d + \overline{e}\,)]}$$

$$\overline{a} + \overline{[b + c\,(d + \overline{e}\,)]}$$

$$\overline{a} + \overline{b}\,(\overline{c\,(d + \overline{e}\,)})$$

$$\overline{a} + \overline{b}\,(\overline{c} + (\overline{d + \overline{e}}\,))$$

$$\overline{a} + \overline{b}\,(\overline{c} + (\overline{d}\ \overline{\overline{e}}))$$

$$\overline{a} + \overline{b}\,(\overline{c} + \overline{d}\ e)$$

Useful laws and theorems

Identity:	$X + 0 = X$	Dual: $X \bullet 1 = X$
Null:	$X + 1 = 1$	Dual: $X \bullet 0 = 0$
Idempotent:	$X + X = X$	Dual: $X \bullet X = X$
Involution:	$(X')' = X$	
Complementarity:	$X + X' = 1$	Dual: $X \bullet X' = 0$
Commutative:	$X + Y = Y + X$	Dual: $X \bullet Y = Y \bullet X$
Associative:	$(X+Y)+Z=X+(Y+Z)$	Dual: $(X\bullet Y)\bullet Z=X\bullet(Y\bullet Z)$
Distributive:	$X\bullet(Y+Z)=(X\bullet Y)+(X\bullet Z)$	Dual: $X+(Y\bullet Z)=(X+Y)\bullet(X+Z)$
Uniting:	$X\bullet Y+X\bullet Y'=X$	Dual: $(X+Y)\bullet(X+Y')=X$

Useful laws and theorems (con’t)

Absorption: $X+X\bullet Y=X$ Dual: $X\bullet(X+Y)=X$

Absorption (#2): $(X+Y')\bullet Y=X\bullet Y$ Dual: $(X\bullet Y')+Y=X+Y$

de Morgan's: $(X+Y+...)'=X'\bullet Y'\bullet ...$ Dual: $(X\bullet Y\bullet ...)'=X'+Y'+...$

Duality: $(X+Y+...)^D=X\bullet Y\bullet ...$ Dual: $(X\bullet Y\bullet ...)^D=X+Y+...$

Multiplying & factoring: $(X+Y)\bullet(X'+Z)=X\bullet Z+X'\bullet Y$

Dual: $X\bullet Y+X'\bullet Z=(X+Z)\bullet(X'+Y)$

Consensus: $(X\bullet Y)+(Y\bullet Z)+(X'\bullet Z)= X\bullet Y+X'\bullet Z$

Dual:

$$(X+Y)\bullet(Y+Z)\bullet(X'+Z)=(X+Y)\bullet(X'+Z)$$

Proving theorems

- Example 1: Prove the uniting theorem-- $X \bullet Y + X \bullet Y' = X$

Distributive	$X \bullet Y + X \bullet Y' = X \bullet (Y + Y')$
Complementarity	$= X \bullet (1)$
Identity	$= X$

- Example 2: Prove the absorption theorem-- $X + X \bullet Y = X$

Identity	$X + X \bullet Y = (X \bullet 1) + (X \bullet Y)$
Distributive	$= X \bullet (1 + Y)$
Null	$= X \bullet (1)$
Identity	$= X$

Proving theorems

- Example 3: Prove the consensus theorem-- $(XY)+(YZ)+(X'Z)= XY+X'Z$

Complementarity $XY+YZ+X'Z = XY+(X+X')YZ + X'Z$

Distributive $= XYZ+XY+X'YZ+X'Z$

- *Use absorption {AB+A=A} with A=XY and B=Z*

$= XY+X'YZ+X'Z$

Rearrange terms $= XY+X'ZY+X'Z$

- *Use absorption {AB+A=A} with A=X'Z and B=Y*

$XY+YZ+X'Z = XY+X'Z$

Boolean vs. Ordinary

- Boolean Algebra
  - Associate law not included (but still valid)
  - Distributive law is valid
  - No additive or multiplicative inverses
  - Define complement
  - No. of elements is not clearly defined
    - 2 for two-valued Boolean algebra
- Ordinary Algebra
  - Associate law included
  - Distributive law may not valid
  - Have additive and multiplicative inverses
  - No complement
  - Operator Deal with real numbers
    - Infinite set of elements

Boolean Functions

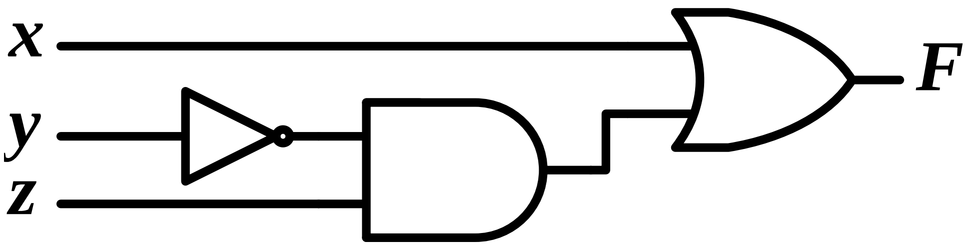
- Boolean Expression

  Example: $F = x + y' z$

- Truth Table

  All possible combinations of input variables

- Logic Circuit



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

Algebraic Manipulation

- *Literal*:

  A single variable within a term that may be complemented or not.

- Use Boolean Algebra to simplify Boolean functions to produce simpler circuits

  Example: Simplify to a minimum number of literals

  $F = x + x' y$ (3 *Literals*)

  $= x + (x' y)$

  Distributive law (+ over •)

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

  $= (1) (x + y) = x + y$ (2 *Literals*)

Complement of a Function

- DeMorgan's Theorm

$$F = A + B + C$$

$$\overline{F} = \overline{A + B + C}$$

$$\overline{F} = \overline{A} \bullet \overline{B} \bullet \overline{C}$$

- Duality & Literal Complement

$$F = A + B + C$$

$$F \quad A \bullet B \bullet C$$

$$\overline{F} = \overline{A} \bullet \overline{B} \bullet \overline{C}$$

Canonical Forms

- Minterm
  - Product (*AND* function)
  - Contains all variables
  - Evaluates to '1' for a specific combination

Example

$$\left.\begin{aligned} A &= 0 \\ B &= 0 \\ C &= 0 \end{aligned}\right\} \quad \begin{matrix} \overline{A} & & \overline{B} & & \overline{C} \\ (\overline{0}) & \bullet & (\overline{0}) & \bullet & (\overline{0}) \\ \downarrow & & \downarrow & & \downarrow \\ 1 & \bullet & 1 & \bullet & 1 = 1 \end{matrix}$$

	A	B	C	Minterm	
0	0	0	0	m_0	$\overline{A}\,\overline{B}\,\overline{C}$
1	0	0	1	m_1	$\overline{A}\,\overline{B}\,C$
2	0	1	0	m_2	$\overline{A}\,B\,\overline{C}$
3	0	1	1	m_3	$\overline{A}\,B\,C$
4	1	0	0	m_4	$A\,\overline{B}\,\overline{C}$
5	1	0	1	m_5	$A\,\overline{B}\,C$
6	1	1	0	m_6	$A\,B\,\overline{C}$
7	1	1	1	m_7	$A\,B\,C$

Canonical Forms

- Maxterm
  - Sum (*OR* function)
  - Contains all variables
  - Evaluates to '0' for a specific combination

Example

$$\left.\begin{array}{l} A = 1 \\ B = 1 \\ C = 1 \end{array}\right\} \begin{array}{ccccc} \bar{A} & & \bar{B} & & \bar{C} \\ (\bar{1}) & + & (\bar{1}) & + & (\bar{1}) \\ \downarrow & & \downarrow & & \downarrow \\ 0 & + & 0 & + & 0 = 0 \end{array}$$

	A B C	Maxterm	
0	0 0 0	M_0	$A+B+C$
1	0 0 1	M_1	$A+B+\bar{C}$
2	0 1 0	M_2	$A+\bar{B}+C$
3	0 1 1	M_3	$A+\bar{B}+\bar{C}$
4	1 0 0	M_4	$\bar{A}+B+C$
5	1 0 1	M_5	$\bar{A}+B+\bar{C}$
6	1 1 0	M_6	$\bar{A}+\bar{B}+C$
7	1 1 1	M_7	$\bar{A}+\bar{B}+\bar{C}$

Canonical Forms

- Truth Table *to* Boolean Function

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

$$F = \overline{A}\,\overline{B}C + A\overline{B}\,\overline{C} + A\overline{B}C + ABC$$

Standard Forms

- Sum of Products (SOP)

$$F = \overline{A}\,\overline{B}C + A\overline{B}\,\overline{C} + A\overline{B}C + ABC$$

$$A\overline{B}(\overline{C} + C)$$

$$= A\overline{B}(1)$$

$$= A\overline{B}$$

$$AC(\overline{B} + B)$$

$$= AC$$

$$\overline{B}C(\overline{A} + A)$$

$$= \overline{B}C$$

$$F = \overline{B}C\cancel{(\overline{A} + A)} + A\overline{B}\cancel{(\overline{C} + C)} + AC\cancel{(\overline{B} + B)}$$

$$F = \overline{B}C + A\overline{B} + AC$$

Standard Forms

- Product of Sums (POS)

$$\overline{A}B(\overline{C}+C)$$

$$\overline{F} = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\overline{C} + \overline{A}BC + AB\overline{C}$$

$$B\overline{C}(\overline{A}+A)$$

$$\overline{A}\,\overline{C}(\overline{B}+B)$$

$$\overline{F} = \overline{A}\,\overline{C}\cancel{(\overline{B}+B)} + \overline{A}B\cancel{(\overline{C}+C)} + B\overline{C}\cancel{(\overline{A}+A)}$$

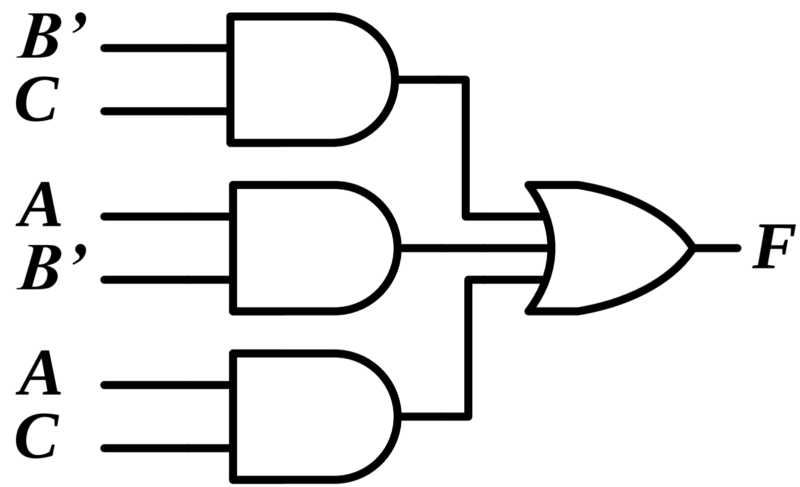
$$\overline{\overline{F}} = \overline{\overline{A}\,\overline{C} + \overline{A}B + B\overline{C}}$$

$$F = (A+C)(A+\overline{B})(\overline{B}+C)$$

Two-Level Implementations

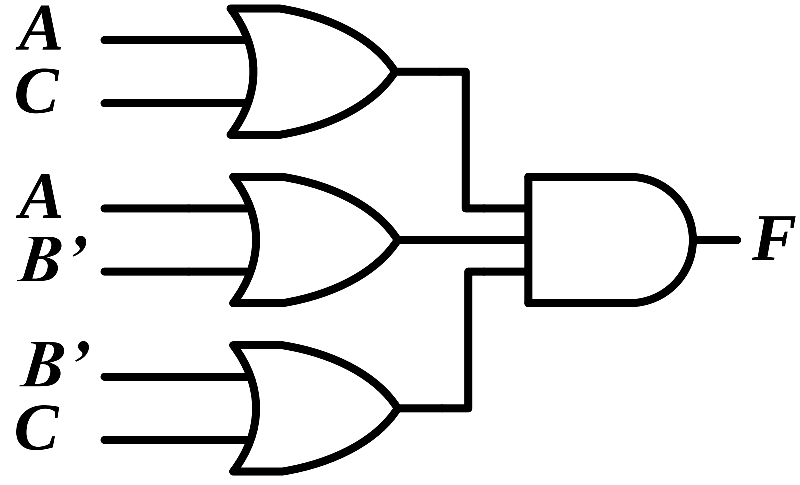
- Sum of Products (SOP)

$$F = \overline{B}C + A\overline{B} + AC$$



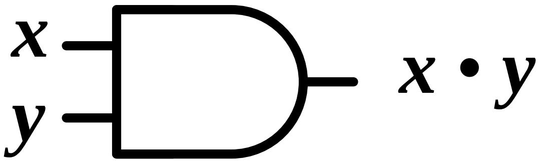
- Product of Sums (POS)

$$F = (A+C)(A+\overline{B})(\overline{B}+C)$$



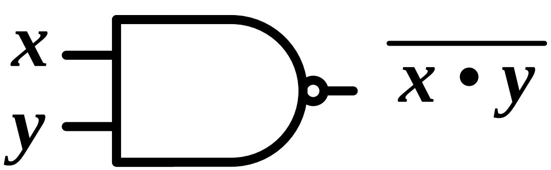
Logic Operators

- AND



x	y	AND
0	0	0
0	1	0
1	0	0
1	1	1

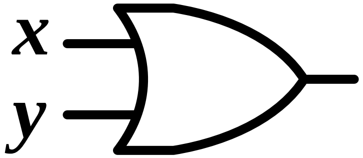
- NAND (Not AND)



x	y	NAND
0	0	1
0	1	1
1	0	1
1	1	0

Logic Operators

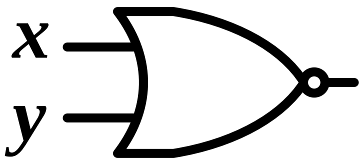
- OR



$x + y$

x	y	OR
0	0	0
0	1	1
1	0	1
1	1	1

- NOR (Not OR)

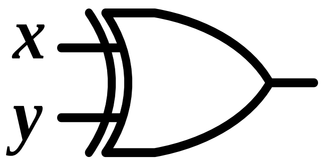


$\overline{x + y}$

x	y	NOR
0	0	1
0	1	0
1	0	0
1	1	0

Logic Operators

- XOR (Exclusive-OR)

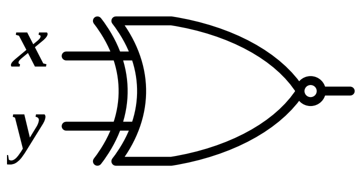


$x \oplus y$

$\bar{x} y + x \bar{y}$

x	y	XOR
0	0	0
0	1	1
1	0	1
1	1	0

- XNOR (Exclusive-NOR)
  (Equivalence)



$\overline{x \oplus y}$

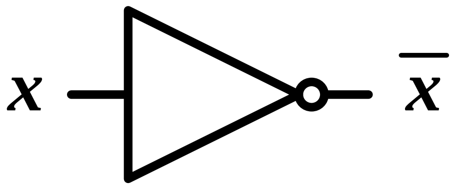
$x \odot y$

$\bar{x}\, \bar{y} + x\, y$

x	y	XNOR
0	0	1
0	1	0
1	0	0
1	1	1

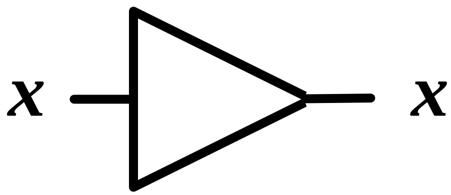
Logic Operators

- NOT (Inverter)



x	*NOT*
0	1
1	0

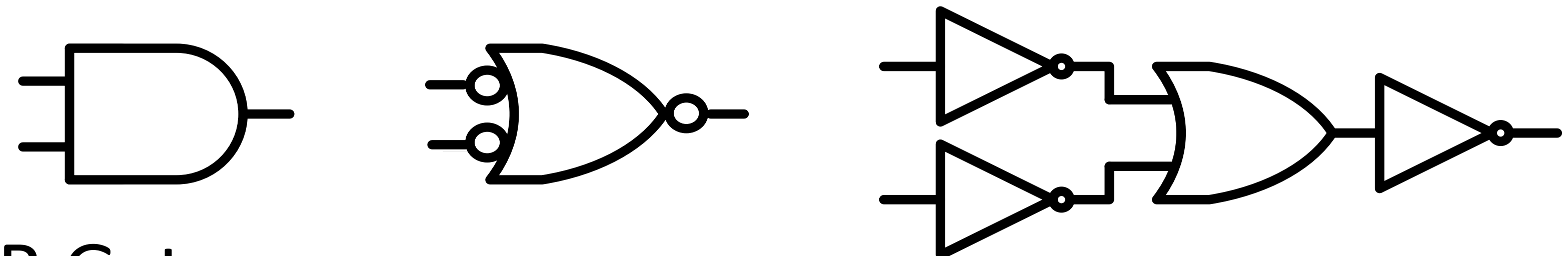
- Buffer



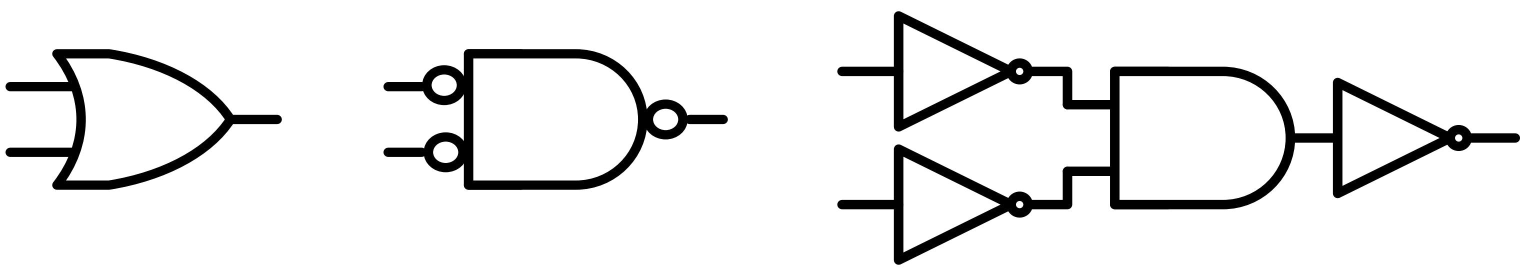
x	*Buffer*
0	0
1	1

DeMorgan’s Theorem on Gates

- AND Gate
  - $F = x \bullet y$ $\overline{F} = (\overline{x \bullet y})$ $\overline{F} = \overline{x} + \overline{y}$



- OR Gate
  - $F = x + y$ $\overline{F} = (\overline{x + y})$ $\overline{F} = \overline{x} \bullet \overline{y}$



➔ **Change the “Shape” and “bubble” all lines**

Homework

- Mano

2-4 **Reduce the following Boolean expressions to the indicated number of literals:**

(a) $A'C' + ABC + AC'$ **to three literals**

(b) $(x'y' + z)' + z + xy + wz$ **to three literals**

(c) $A'B(D' + C'D) + B(A + A'CD)$ **to one literal**

(d) $(A' + C)(A' + C')(A + B + C'D)$ **to four literals**

2-5 **Find the complement of** $F = x + yz$**; then show that** $FF' = 0$ **and** $F + F' = 1$

Homework

2-6 Find the complement of the following expressions:

(a) $xy' + x'y$ **(b)** $(AB' + C)D' + E$

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

2-8 List the truth table of the function:

$$F = xy + xy' + y'z$$

2-9 Logical operations can be performed on strings of bits by considering each pair of corresponding bits separately (this is called bitwise operation). Given two 8-bit strings A = 10101101 and B = 10001110, evaluate the 8-bit result after the following logical operations: (a) AND, (b) OR, (c) XOR, (d) NOT A, (e) NOT B.

Homework

2-10 **Draw the logic diagrams for the following Boolean expressions:**

(a) $Y = A'B' + B(A + C)$ **(b)** $Y = BC + AC'$

(c) $Y = A + CD$ **(d)** $Y = (A + B)(C' + D)$

2-12 **Simplify the Boolean function T_1 and T_2 to a minimum number of literals.**

A	B	C	T_1	T_2
0	0	0	1	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

Homework

2-15 Given the Boolean function

$$F = xy'z + x'y'z + w'xy + wx'y + wxy$$

(a) Obtain the truth table of the function.

(b) Draw the logic diagram using the original Boolean expression.

(c) Simplify the function to a minimum number of literals using Boolean algebra.

(d) Obtain the truth table of the function from the simplified expression and show that it is the same as the one in part (a)

(e) Draw the logic diagram from the simplified expression and compare the total number of gates with the diagram of part (b).

Homework

2-18 Convert the following to the other canonical form:

(a) $F(x, y, z) = \sum(1, 3, 7)$

(b) $F(A, B, C, D) = \prod(0, 1, 2, 3, 4, 6, 12)$

2-19 Convert the following expressions into sum of products and product of sums:

(a) $(AB + C)(B + C'D)$

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