Modern Digital System Design

ECE 2372 / Fall 2018 / Lecture 02

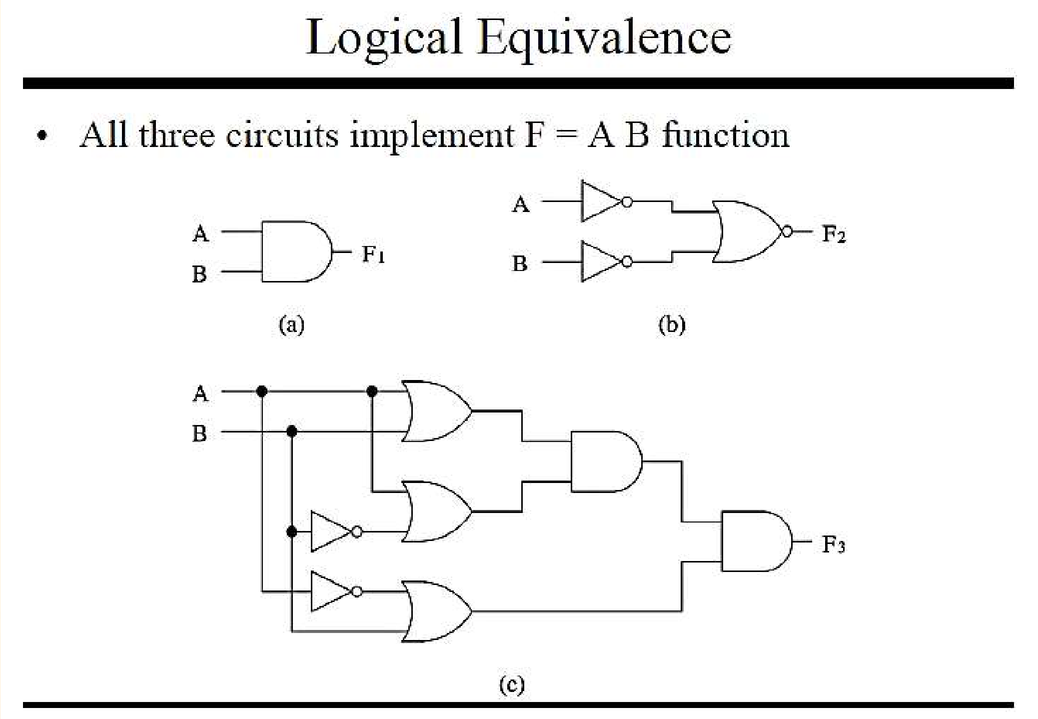
Texas Tech University
Dr. Tooraj Nikoubin

Logic Gates and Boolean Algebra

Logic Functions

- **Logical functions can be expressed in several ways:**

  - ∗ Truth table
  - ∗ Logical expressions
  - ∗ Graphical form



Boolean Algebra

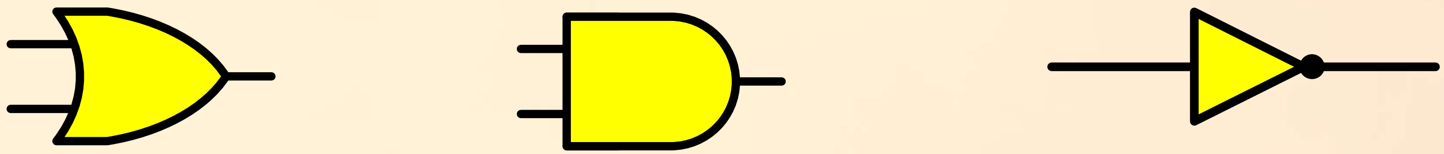
- Computer hardware using binary circuit greatly simply design

- Binary circuits: To have a conceptual framework to manipulate the circuits algebraically

- George Boole (1813-1864): developed a mathematical structure
–To deal with binary operations with just two values.

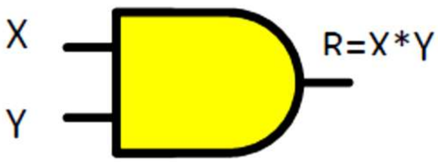
Basic Gates in Binary Circuit

- Element 0 : “FALSE”. Element 1 : “TRUE”.
- ‘+’ operation “OR”,‘*’ operation “AND” and ’ operation “NOT”.



AND Gate

- '*' operation "AND"



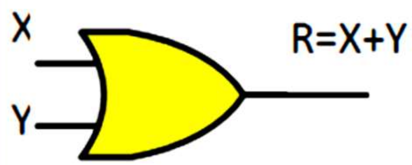
X	Y	R=X AND Y R= X * Y
0	0	0
0	1	0
1	0	0
1	1	1

0 * Y = 0

One "0" value as a input is enough to set the output "0"

OR Gate

- '+' operation "OR"



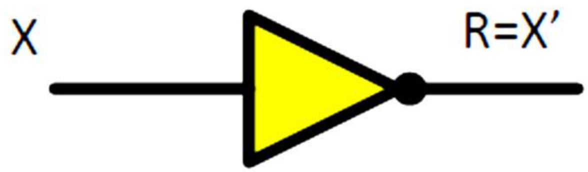
X	Y	R=X OR Y R= X + Y
0	0	0
0	1	1
1	0	1
1	1	1

1 + Y = 1

One "1" value as a input is enough to set the output "1"

NOT Gate (Inverter)

- ' operation "NOT"

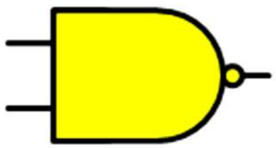


X	R=X' R= NOT X
0	1
1	0

$$Y = \overline{A} = A'$$

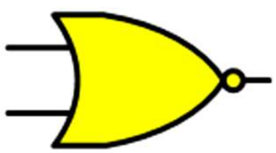
Logic Gates: NAND, NOR

- NAND



A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

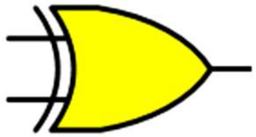
- NOR



A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

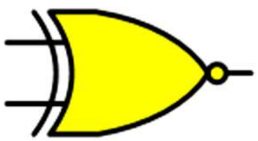
Logic Gates: XOR, XNOR

- XOR



A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

- XNOR



A	B	A XNOR B
0	0	1
0	1	0
1	0	0
1	1	1

XOR: Exclusive OR **XNOR: Exclusive NOR**

Boolean Algebra Defined

Boolean Algebra B : 5-tuple

{ +, *, ', 0, 1}

+ and ***** are *binary* operators,

' is a *unary* operator.

$$Y = \overline{A} = A'$$

Boolean Algebra Defined

- ***Axiom #1: Closure***

If a and b are Boolean
(a + b) and (a * b) are Boolean.

- ***Axiom #2: Cardinality***

if a is Boolean then a’ is Boolean

- ***Axiom #3: Commutative***

$$(a + b) = (b + a)$$
$$(a * b) = (b * a)$$

Boolean Algebra Defined

• ***Axiom #4: Associative :*** **If a and b are Boolean**

(a + b) + c = a + (b + c)

(a * b) * c = a * (b * c)

• ***Axiom #5: Distributive***

a * (b + c) = (a * b) + (a * c)

a + (b * c) = (a + b) * (a + c)

2nd one is Not True for Decimal numbers System

Boolean Algebra Defined

- ***Axiom #6: Identity Element :***

  B has identity to + and *

  0 is identity element for + : a + 0 = a

  1 is identity element for * : a * 1 = a

- ***Axiom #7: Complement Element***

  a + a' = 1

  a * a' = 0

Terminology

- **Juxtaposition implies * operation:**

$$ab = a * b = a.b$$

- **Operator order of precedence is:**

$$(\ ) > ' > * > +$$

$$a+bc = a+(b*c) \neq (a+b)*c$$
$$ab' = a(b') \neq (a*b)'$$

Named Theorems

Idempotent	a + a = a	a * a = a
Boundedness	a + 1 = 1	a * 0 = 0
Absorption	**a + (a*b) = a**	**a*(a+b) = a**
Associative	(a+b)+c= a+(b+c)	(a*b)*c= a*(b*c)

Involution	(a')' = a	
DeMorgan's	(a+b)' = a' * b'	(a*b)'=a' + b'

Logical Expression Simplification

- Three basic methods
  - ∗ **Algebraic manipulation**
    - » Use Boolean laws to simplify the expression
  - – Difficult to use
  - – Don't know if you have the simplified form
  - ∗ **Karnaugh map method**
    - » Graphical method
    - » Easy to use
  - – Can be used to simplify logical expressions with a few variables
  - ∗ **Quine-McCluskey method**
    - » Tabular method
    - » Can be automated

Simplification Theorem

- Uniting :

$XY + XY' = X$ $\quad X(Y+Y')=X.1=X$

$(X + Y)(X + Y') = X$ $\quad XX+XY'+YX+YY'=X+X(\mathbf{Y+Y'})+0=X$

- Absorption:

$X + XY = X$ $\quad X(1+Y)=X.1=X$

$X(X + Y) = X$ $\quad XX+XY=X+XY=X$

- Adsorption

$(X + Y')Y = XY,\ XY' + Y = X + Y$ $\quad XY+YY'=XY+0=XY$

N-bit Boolean Algebra

Single bit to *n-bit* Boolean Algebra

Let a = 1101010, b = 1011011

a + b = 1101010 + 1011011

$$\begin{array}{r} a + b = 1101010\ + \\ 1011011 \\ \hline 1111011 \end{array}$$

a * b = 1101010 * 1011011

$$\begin{array}{r} a * b = 1101010\ * \\ 1011011 \\ \hline 1001010 \end{array}$$

a’ = 1101010’ = 0010101

$$\begin{array}{r} a' = 1101010' \\ \hline 0010101 \end{array}$$

Proof by Truth Table

- Consider the distributive theorem:

a + (b * c) = (a + b)*(a + c)

Is it true for a two bit Boolean Algebra?

- Can prove using a truth table

–How many possible combinations of *a*, *b*, and *c* are there?

- Three variables, each with two values

–2*2*2 = 2^3 = 8

Proof by Truth Table

a	b	c	b*c	a+(b*c)	a+b	a+c	(a+b)*(a+c)
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

$$a + (b * c) = (a + b)*(a + c)$$

Proof using Theorems

Use the properties of Boolean Algebra to proof

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

Warning, make sure you use the laws precisely

$(x + y)(x + x)$	**Given**
$(x + y)x$	**Idempotent**
$x(x + y)$	**Commutative**
x	**Absorption**

Converting to Boolean Equations

Convert the following English statements to a Boolean *a* equation

– Q1. a is 1 and b is 1.

Answer: F = a AND b = ab

– Q2. either of a or b is 1.

Answer: F = a OR b = a+b

– Q3. both a and b are not 0.

Answer:

– (a) Option 1: F = NOT(a) AND NOT(b) = a’b’

– (b) Option 2: F = a OR b = a+b

– Q4. a is 1 and b is 0.

Answer: F = a AND NOT(b) = ab’

Complete sets

- A set of gates is complete
  - » if we can implement any logical function using only the type of gates in the set
    - – You can uses as many gates as you want

- Some example complete sets
  - » {AND, OR, NOT} ← Not a minimal complete set
  - » {AND, NOT}
  - » {OR, NOT}
  - » {NAND}
  - » {NOR}

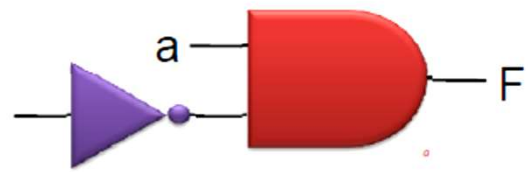
- Minimal complete set
  - – A complete set with no redundant elements.

Example: Converting a Boolean

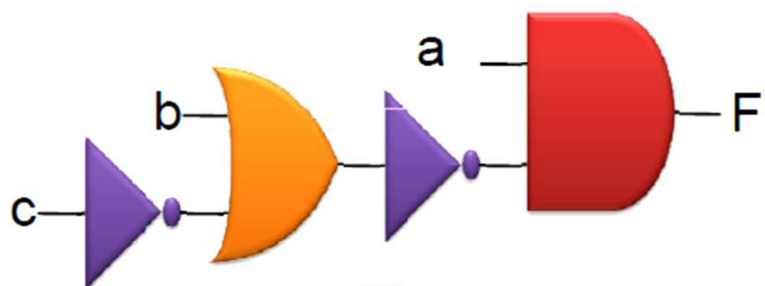
Equation to a Circuit of Logic Gates

Q: Convert the following equation to logic gates:

F = a AND NOT(b OR NOT(c))

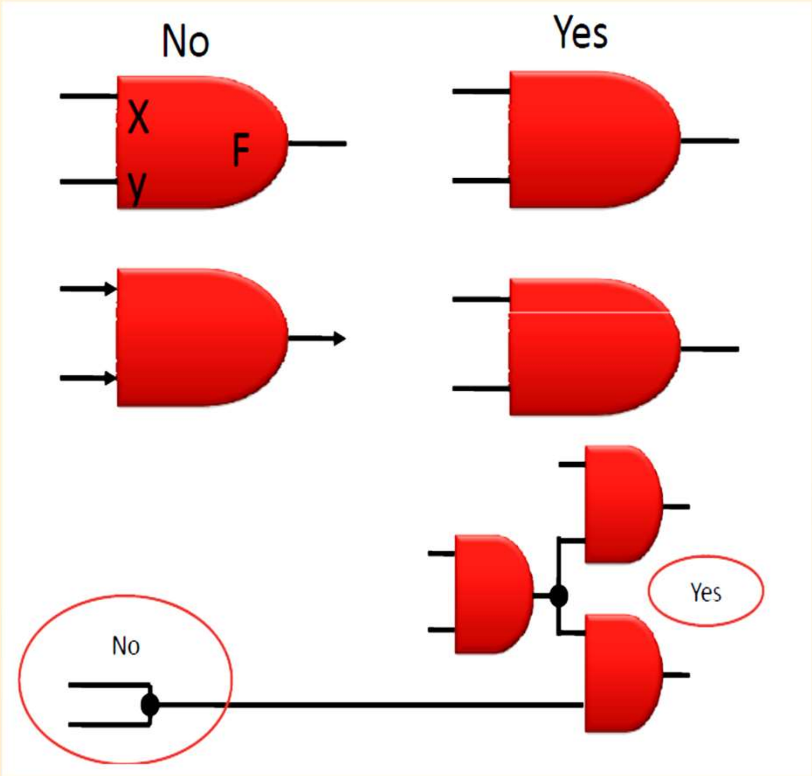


(a)



(b)

Some Circuit Drawing Conventions



Duality examples

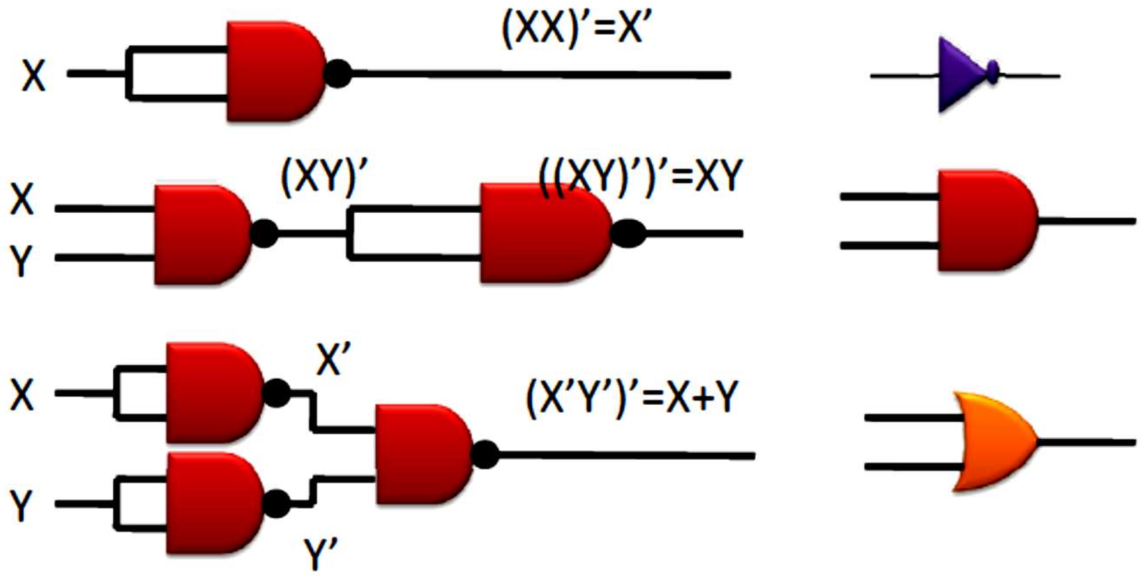
$x + 0 = x$	$x.1=x$
$x + x'=1$	$x \,.\, x'=0$
$A + B'C$	$A.\,(B'+C)$
$A'B' + AB$	$(A'+B').(A+B)$

NAND & NOR are universal

$(xx)' = x'$

$((xy)')' = xy$

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

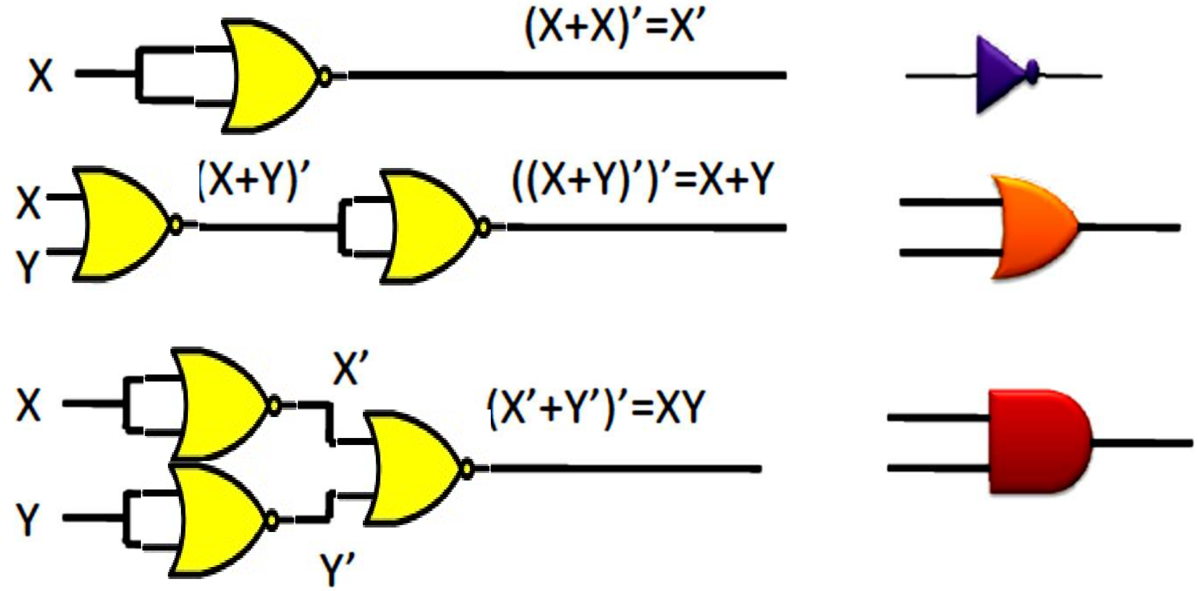


NAND & NOR are universal

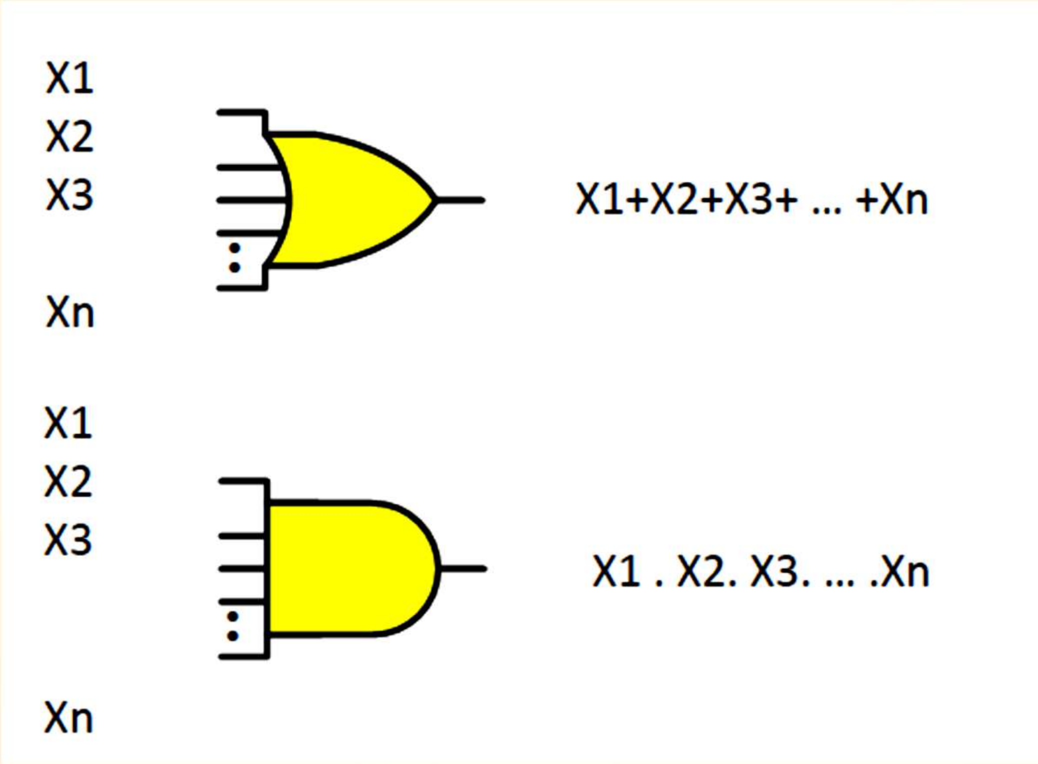
$(x+x)' = x'$

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

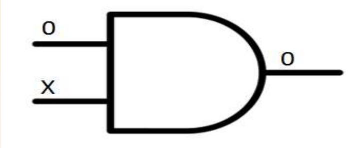
$(x'+y')' = x.y$

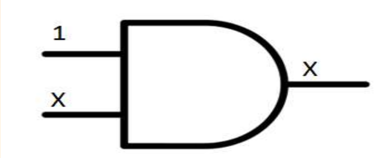


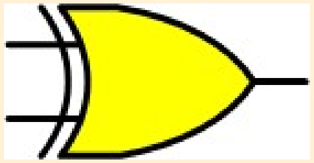
Multi-input gate

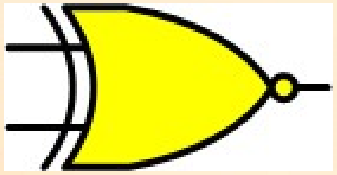


Effect of 0 & 1 on logic gates









0 & 1
for OR Logic Gates
for AND Logic Gates
for XOR Logic Gates
for XNOR Logic Gates

Canonical form or Standard Form

Example : 3 –Input Majority Function

- Canonical forms

–Sum of minterms (SOM)

–Product of maxterms (POM)

- Standard forms (may use less gates)

–Sum of products (SOP)

–Product of sums (POS)

F = ab+a' (already sum of products : SOP)

F = ab + a'(b+b') (expanding term)

F = ab + a'b + a'b' (it is canonical form : SOM)

Shannon Expansion

- $F(X, Y, Z) = X \, . \, F(1,Y,Z) + X' . \, F(0, Y, Z)$

Example:

$XY+X'Z+YZ$

$= X. \, (1.Y+0.Z+YZ) + X' \, (0.Y+1.Z+YZ)$

$= X.(Y+YZ)+X'(Z+YZ)$

$= X.(Y(1+Z))+X'(Z(1+Y))$

$= X(Y.1)+X'(Z.1)$

$= XY+X'Z$

Consensus Theorem

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

$$
\begin{aligned}
& XY + X'Z + YZ \\
&= xy + x'z + (x + x')yz \\
&= xy + x'z + xyz + x'yz \\
&= xy + xyz + x'z + x'yz \\
&= xy(1 + z) + x'z(1 + y) \\
&= xy + x'z
\end{aligned}
$$

Duality

Consensus (collective opinion) of X.Y and X'.Z is Y.Z

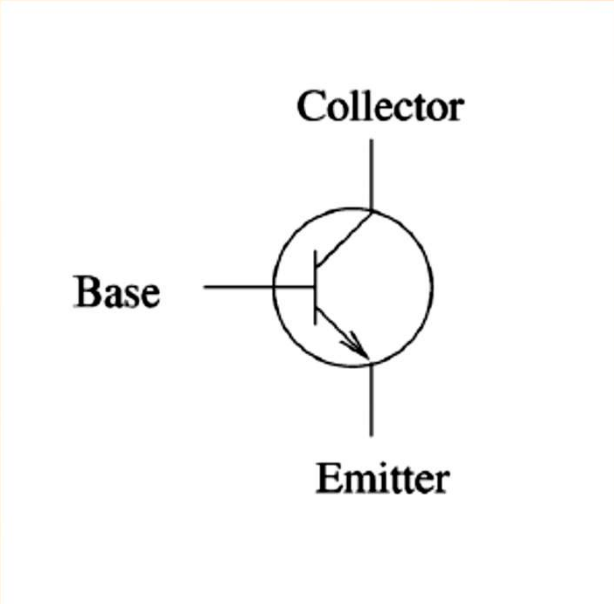
- $(X + Y)(X' + Z)(Y + Z) = (X + Y)(X' + Z)$

Principle of Duality

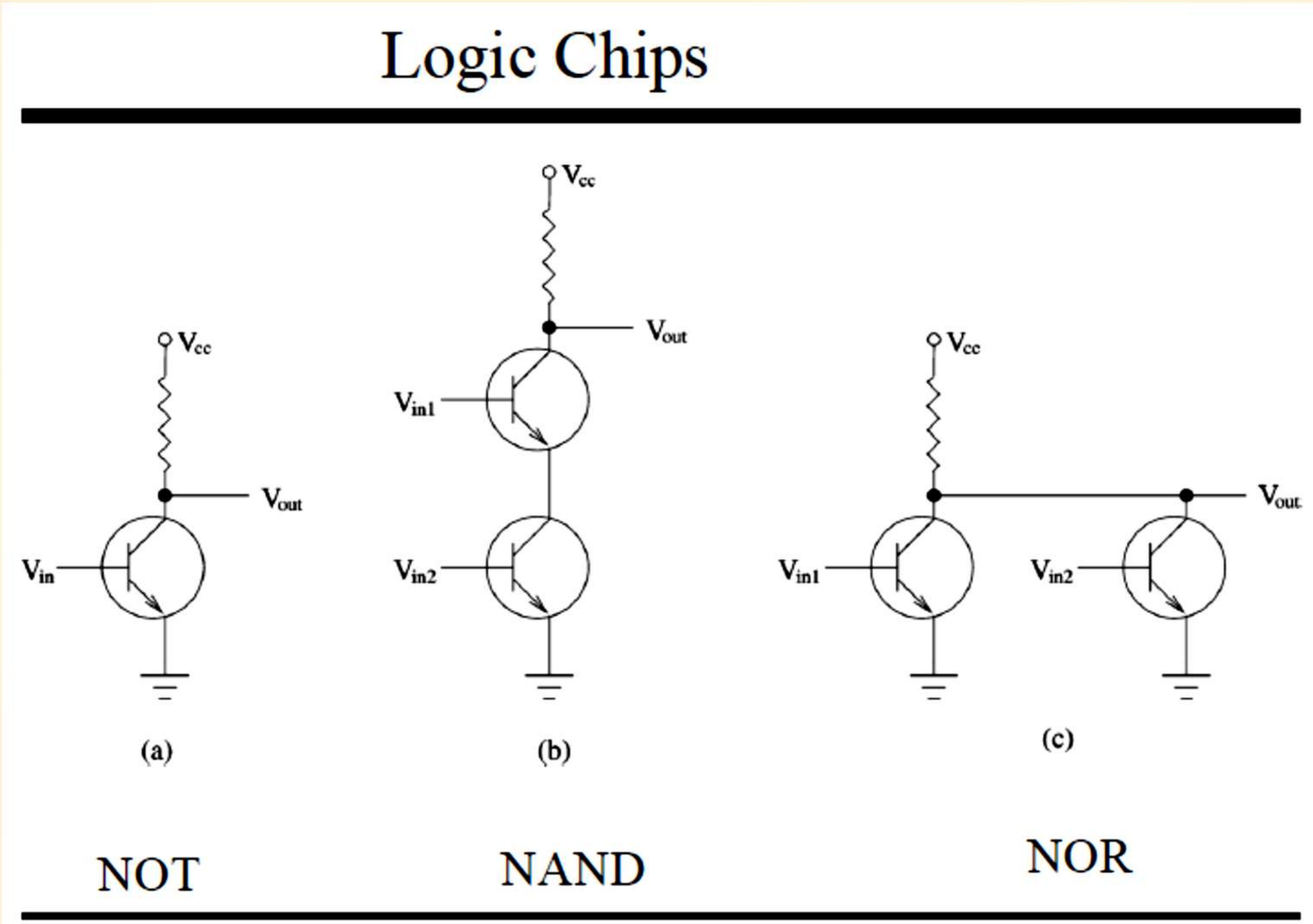
- Dual of a statement S is obtained
  - By interchanging * and +
  - By interchanging 0 and 1
- Dual of $(a*1)*(0+a') = 0$ is $(a+0)+(1*a') = 1$

Basic building block:

- » Transistor
- Three connection points
  - ∗ Base
  - ∗ Emitter
  - ∗ Collector
- Transistor can operate
  - ∗ Linear mode
    - » Used in amplifiers
  - ∗ Switching mode
    - » Used to implement digital circuits

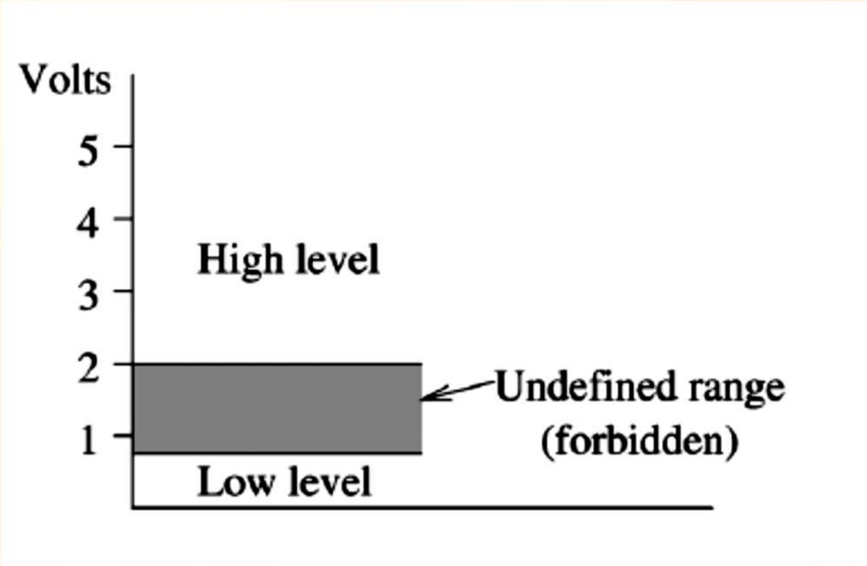


Basic building block:



Level of voltages for Low and High

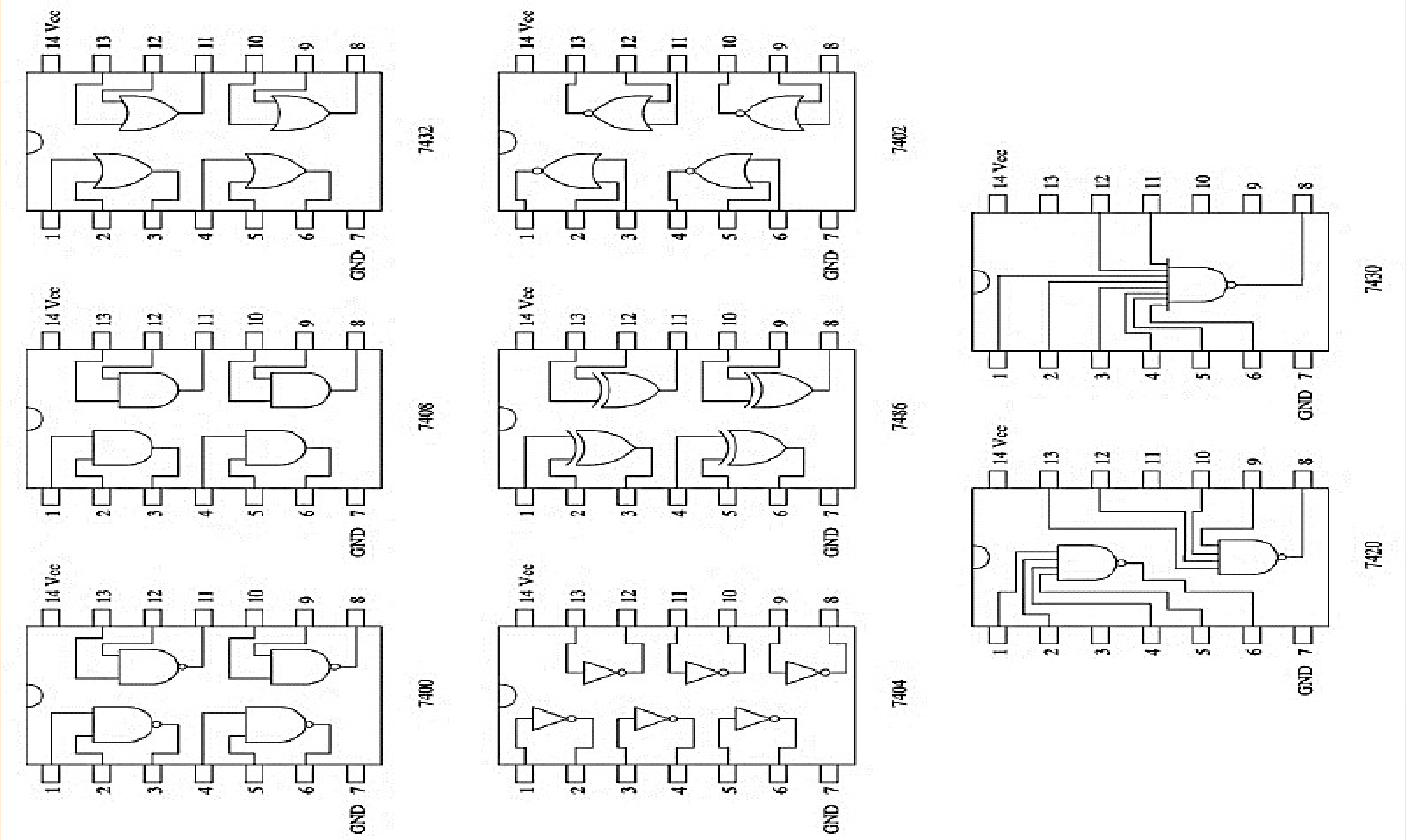
- Low voltage level: < 0.4V
- High voltage level: > 2.4V
- Positive logic:
  - Low voltage represents 0
  - High voltage represents 1
- Negative logic:
  - High voltage represents 0
  - Low voltage represents 1
- Propagation delay
  - Delay from input to output
  - Typical value: 5-10 ns



Integration levels

- SSI (small scale integration)
  - » Introduced in late 1960s
  - » 1-10 gates (previous examples)
- MSI (medium scale integration)
  - » Introduced in late 1960s
  - » 10-100 gates
- LSI (large scale integration)
  - » Introduced in early 1970s
  - » 100-10,000 gates
- VLSI (very large scale integration)
  - » Introduced in late 1970s
  - » More than 10,000 gates

Logic Chips (SSI)



SSI: small scale integration

Review

1. $A_1 . (A_1 + A_2) = A_1$

2. $A_1 . (A_1' + A_2) = A_1 . A_2$

3. $A_1 + (A_1' . \; A_2) = A_1 + A_2$

4. $A_1 . \; A_2 + (A_1 . \; A_2') = A_1$

5. $(A_1 + A_2) . (A_1 + A_2') = A_1$

Review

0 . 0 = 0	X . X = X
0 . 1 = 0	X . X´ = 0
1 . 0 = 0	X´ . X = 0
1 . 1 = 1	X´ . X´ = X´

0 + 0 = 0	X + X = X
0 + 1 = 1	X + X´ = 1
1 + 0 = 1	X´ + X = 1
1 + 1 = 1	X´ + X´ = X´

Review

1. $\overline{A + B} = \overline{A} \, . \, \overline{B}$

2. $\overline{\overline{Y}} = Y$

3. $\overline{A \, . \, B} = \overline{A} + \overline{B}$

$\overline{AC + \overline{\overline{B}}D} = \overline{AC} \, . \, (\overline{B}D)$

0 . X = 0	**0 + X = X**
0 . X´ = 0	**0 +X´ =X’**
1 . X = X	**1 + X = 1**
1 . X´ = X´	**1+ X´ = 1**

Review

Boolean Algebra

Boolean identities

Name	AND version	OR version
Identity	$x \cdot 1 = x$	$x + 0 = x$
Complement	$x \cdot \overline{x} = 0$	$x + \overline{x} = 1$
Commutative	$x \cdot y = y \cdot x$	$x + y = y + x$
Distribution	$x \cdot (y+z) = xy+xz$	$x + (y \cdot z) = (x+y)(x+z)$
Idempotent	$x \cdot x = x$	$x + x = x$
Null	$x \cdot 0 = 0$	$x + 1 = 1$

Review

Boolean Algebra (cont'd)

- Boolean identities (cont'd)

Name	AND version	OR version
Involution	$\overline{\overline{x}} = x$	---
Absorption	$x \cdot (x+y) = x$	$x + (x \cdot y) = x$
Associative	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$	$x + (y + z) = (x + y) + z$
de Morgan	$\overline{x \cdot y} = \overline{x} + \overline{y}$	$\overline{x + y} = \overline{x} \cdot \overline{y}$

Laws and Theorems of Boolean Algebra

Operations with 0 and 1:

1. $X + 0 = X$ 1D. $X \cdot 1 = X$
2. $X + 1 = 1$ 2D. $X \cdot 0 = 0$

Idempotent laws:

3. $X + X = X$ 3D. $X \cdot X = X$

Involution law:

4. $(X')' = X$

Laws of complementarity:

5. $X + X' = 1$ 5D. $X \cdot X' = 0$

Commutative laws:

6. $X + Y = Y + X$ 6D. $XY = YX$

Associative laws:

7. $(X + Y) + Z = X + (Y + Z) = X + Y + Z$ 7D. $(XY)Z = X(YZ) = XYZ$

Distributive laws:

8. $X(Y + Z) = XY + XZ$ 8D. $X + YZ = (X + Y)(X + Z)$

Simplification theorems:

9. $XY + XY' = X$ 9D. $(X + Y)(X + Y') = X$
10. $X + XY = X$ 10D. $X(X + Y) = X$
11. $(X + Y')Y = XY$ 11D. $XY' + Y = X + Y$

DeMorgan's laws:

12. $(X + Y + Z + \ldots)' = X'Y'Z' \ldots$ 12D. $(XYZ \ldots)' = X' + Y' + Z' + \ldots$

Duality:

13. $(X + Y + Z + \ldots)^D = XYZ \ldots$ 13D. $(XYZ \ldots)^D = X + Y + Z + \ldots$

Theorem for multiplying out and factoring:

14. $(X + Y)(X' + Z) = XZ + X'Y$ 14D. $XY + X'Z = (X + Z)(X' + Y)$

Consensus theorem:

15. $XY + YZ + X'Z = XY + X'Z$ 15D. $(X + Y)(Y + Z)(X' + Z) = (X + Y)(X' + Z)$

Thank You