Lect 05

Boolean Algebra & Functions
Canonical Form

CS221: Digital Design

Dr. A. Sahu
Dept of Comp. Sc. & Engg.
Indian Institute of Technology Guwahati

1

Outline

- Gates in Digital System
  - Basic Gates (AND, OR & NOT)
  - Universal Gates (NAND & NOR)
  - Others : XOR, XNOR
- Boolean Algebra
  - Axioms
- Boolean Functions
- Canonical form of Function
  - SOP and POS

2

Boolean Algebra

Boolean Algebra Axioms and Rules

Axioms/Rules	Equations	
Closure	If a and b are BLN **a+b** and **a*b** are BLN.	
Cardinality	If a is BLN **a'** is BLN.	
Commutative	(a + b) = (b + a)	(a * b) = (b * a)
Associative	(a+b)+c = a+(b+c)	(a*b)*c = a*(b*c)
Distributive	a*(b+c)=(a*b)+(a*c)	a+(b*c) =(a+b)*(a+c)
Identity	a + 0 = a	a *1 =a
Complement	a + a' = 1	a *a' =0
Op Precedence	( ) > ' > * > +	
Involution	(a')'=a	

Boolean Algebra Axioms and Rules

Axioms/Rules	Equations	
Idempotent	a + a = a	a * a = a
Boundness	a + 1 = 1	a * 0 = 0
Absorption	a +(a*b) = a	a * (a+b) = a
Adorption	(X + Y')Y = XY,	XY' + Y = X + Y
uniting	XY + XY' = X	(X + Y)(X + Y') = X
duality	**Dual (S) by interchanging * & +, 0 &1**	
shannon	F(A,B…Z)= A' . F(0,B..Z) + A. F(1,B…Z)	
Consensus	XY + X'Z + YZ = XY + X'Z	
DeMorgans	(a+b)' = a' * b'	(a*b)'=a' + b'

N-bit Boolean Algebra

- Single bit to *n-bit* Boolean Algebra
- Let a = 1101010, b = 1011011
  - a + b = 1101010

        + 1011011

          1111011
  -

N-bit Boolean Algebra

- Single bit to *n-bit* Boolean Algebra
- Let a = 1101010, b = 1011011
  - a * b = 1101010
  -       * 1011011

  -      = 1001010

N-bit Boolean Algebra

- Single bit to *n-bit* Boolean Algebra
- Let a = 1101010, b = 1011011
  - a' = 1101010'
  - = 0010101

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

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** **x(x+y)=xx+xy=x(1+y)=x**

How to prove 2+2=5?

We know 2+2=4

$$2+2 = 4 - \frac{9}{2} + \frac{9}{2} = \boxed{\sqrt{(4 - \frac{9}{2})^2}} + \frac{9}{2}$$

$$= \sqrt{16 - 2.4.\frac{9}{2} + (\frac{9}{2})^2} + \frac{9}{2}$$

$$= \sqrt{-20 + (\frac{9}{2})^2} + \frac{9}{2} = \sqrt{25 - 45 + (\frac{9}{2})^2} + \frac{9}{2}$$

$$= \sqrt{5^2 - 2.4.\frac{9}{2} + (\frac{9}{2})^2} + \frac{9}{2} = \sqrt{(5 - \frac{9}{2})^2} + \frac{9}{2}$$

$$= 5 - \frac{9}{2} + \frac{9}{2} = 5$$

Where is the mistake?

$\sqrt{x^2}=x$ is true only when $x \geq 0$

12

Converting to Boolean Equations

- Convert the following English statements to a Boolean 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

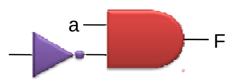
Converting to Boolean Equations

- Convert the following English statements to a Boolean equation
  - –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'

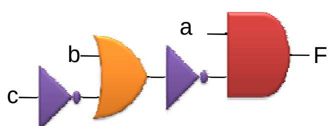
Example: Converting a Boolean Equation to a Circuit of Logic Gates

- Q: Convert the following equation to logic gates:

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

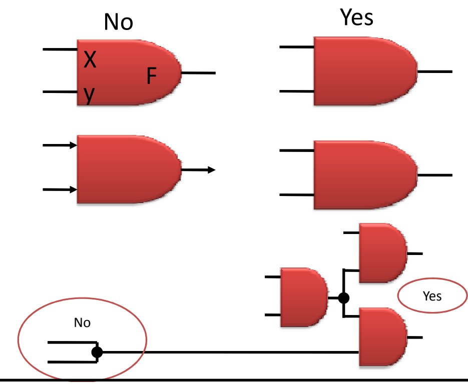


(a)



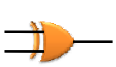
(b)

Some Circuit Drawing Conventions



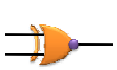
Logic Gates: XOR, XNOR

- XOR



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

- XNOR

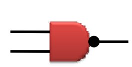


A	B	A XNOR B
0	**0**	**1**
0	1	0
1	0	0
1	**1**	**1**

17

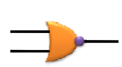
Logic Gates: NAND, NOR

- NAND



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

- NOR

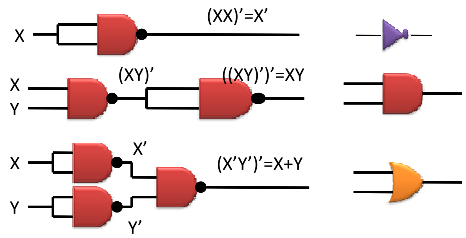


A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

18

NAND & NOR are universal

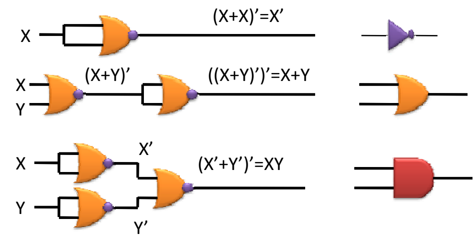
- (xx)’ = x’
- ((xy)’)’= xy
- (x’y’)’= x+y



19

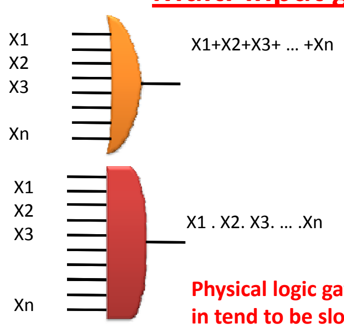
NAND & NOR are universal

- (x+x)’ = x’
- ((x+y)’)’= x+y
- (x’+y’)’= x.y



20

Multi-input gate



Physical logic gates with a large fan-in tend to be slower than those with a small fan-in

21

Boolean Functions

Boolean Functions: Terminology

F(a,b,c) = a’bc + abc’ + ab + c

- **Variable**
  - Represents a value (0 or 1), Three variables: **a**, **b**, and **c**
- **Literal**
  - Appearance of a variable, in true or complemented form
  - Nine literals: **a’**, **b, c, a, b, c’, a, b,** and **c**

Boolean Functions: Terminology

F(a,b,c) = a’bc + abc’ + ab + c

- **Product term**
  - Product of literals, Four product terms: **a’bc, abc’, ab, c**
- **Sum-of-products (SOP)**
  - Above equation is in sum-of-products form.
  - “F = (a+b)c + d” is not.

Representations of Boolean Functions

- A function can be represented in different ways
  - **English, Equation**, Circuit, and Truth Table

English 1: F outputs 1 when a is 0 and b is 0, or when a is 0 and b is 1.

Equation 1: F(a,b) = a'b' + a'b

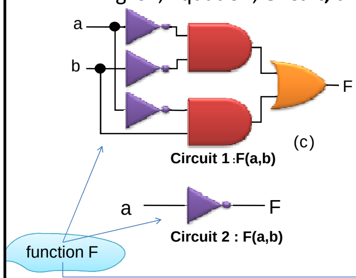
English 2: F outputs 1 when a is 0, regardless of b's value

Equation 2: F(a,b) = a'

function F

Representations of Boolean Functions

- A function can be represented in different ways
  - English, Equation, **Circuit, and Truth Table**



a	b	F
0	0	**1**
0	1	**1**
1	0	**0**
1	1	**0**

Truth table
(d)

Representations of Boolean Functions

English 1: F outputs 1 when a is 0 and b is 0, or when a is 0 and b is 1.

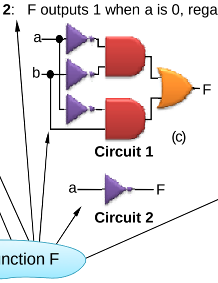
English 2: F outputs 1 when a is 0, regardless of b's value

(a)

Equation 1: F(a,b) = a'b' + a'b

Equation 2: F(a,b) = a'

(b)



a	b	F
0	0	**1**
0	1	**1**
1	0	**0**
1	1	**0**

Truth table
(d)

Above shows seven representations of the same functions F(a,b), using four different methods: English, Equation, Circuit, and Truth Table

Truth Table Representation of Functions

- Define value of F for each possible combination of input values
  - 2-input function: 4 rows
  - 3-input function: 8 rows
  - 4-input function: 16 rows

a	b	F
0	0	
0	1	
1	0	
1	1	

a	b	c	F
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

a	b	c	d	F
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Truth Table Representation of Functions

- Q: Use truth table to define function F(a,b,c) that is 1 when abc is 5 or greater in binary

a	b	c	F
0	0	0	**0**
0	0	1	**0**
0	1	0	**0**
0	1	1	**0**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

Converting among Representations

- Can convert from any representation to any other
- Common conversions
  - Equation to circuit
  - Truth table to equation
  - Equation to truth table
    - Easy -- just evaluate equation for each input combination (row)
    - Creating intermediate columns helps

Converting among Representations

Q: Convert to equation

a	b	F	Term
0	0	**1**	**a'b'**
0	1	**1**	**a'b**
1	0	**0**	
1	1	**0**	

F = a'b' + a'b

Converting among Representations

Q: Convert to equation

a	b	c	F	Term
0	0	0	**0**	
0	0	1	**0**	
0	1	0	**0**	
0	1	1	**0**	
1	0	0	**0**	
1	0	1	**1**	**ab'c**
1	1	0	**1**	**abc'**
1	1	1	**1**	**abc**

F = ab'c + abc' + abc

Converting among Representations

Q: Convert to truth table: F = a'b' + a'b

Inputs				Output
a	b	a'b'	a'b	F
0	0	1	0	1
0	1	0	1	1
1	0	0	0	0
1	1	0	0	0

Standard Representation

- How to determine two functions are the same?
  - Use algebraic methods
  - But if we failed, does that prove *not* equal? No.
- Solution: Convert to truth tables
  - Only ONE **truth table** representation of given same functions: Standard representation

Standard Representation: Truth Table

Only ONE truth table representation of given same functions: Standard representation

F=ab+a'		
a	b	F
0	0	**1**
0	1	**1**
1	0	**0**
1	1	**1**

F=a'b'+a'b+ab		
a	b	F
0	0	**1**
0	1	**1**
1	0	**0**
1	1	**1**

Same