WEEK 2-2018

Combinational circuit I

ELEC2141: Digital Circuit Design

UNSW | ENGINEERING

THE UNIVERSITY OF NEW SOUTH WALES

Binary logic and Gates

- **Transistors interconnected each other to form logic gates**

- **Each gate has inputs and an output. It performs a specific logical operation on its binary inputs and provides a binary value at the output.**

- **Inputs and output of a logic gate are designated by alphabetical variables .**

- **These variables can assume only 1 or 0 values and are known as Binary variables.**

Basic logical operations

- **There are three basic logical operations : AND, OR, and NOT.**

AND

- represented by a dot or an absence of operation

$$Z = X \cdot Y = XY = X \wedge Y$$

- Z is 1 if and only if X=1 and Y=1; otherwise Z = 0

$X \cdot Y$
$0 \cdot 0 = 0$
$0 \cdot 1 = 0$
$1 \cdot 0 = 0$
$1 \cdot 1 = 1$

X	Y	$Z = X \cdot Y$
0	0	0
0	1	0
1	0	0
1	1	1

Truth Table for AND gate

AND logical operation can be extended to more than two input binary variables.

Basic logical operations

OR

– **represented by a plus symbol or "∨"**

$$Z = X + Y = X \vee Y$$

– **Z is 1 if X=1 or Y=1; Z is 0 if X = 0 and Y= 0**

X + Y
0+0 = 0
0+1 = 1
1+0 = 1
1+1 = 1

X	Y	X + Y
0	0	0
0	1	1
1	0	1
1	1	1

Truth Table for OR gate

OR logical operation can be extended to more than two input binary variables.

Basic logical operations

NOT

- **represented by a bar over the variable**

$$Z = \overline{X}$$

- **Z is 1 if X=0 and Z is 0 if X = 1**

X
$\overline{0} = 1$
$\overline{1} = 0$

X	$Z = \overline{X}$
0	1
1	0

Truth Table
for NOT gate

Logic gates

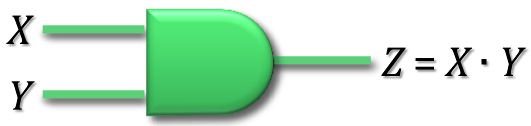
- Logical Gates - Electrical circuits that implement logical operations.

- The input terminals accept voltage signals within allowable range (as a binary signals) and gives out at the output terminal a binary signals that also falls with in the allowable range.

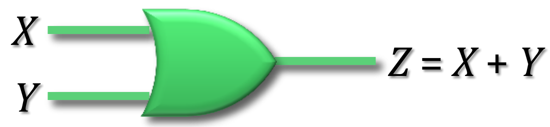
- Intermediate values are crossed only during transitions from 0 to 1 or otherwise.

- Graphical (symbolical) representations of the three types of gates: AND, OR, NOT

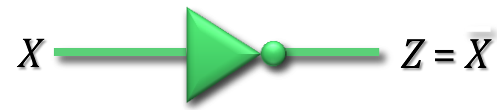
Logic gates



AND gate

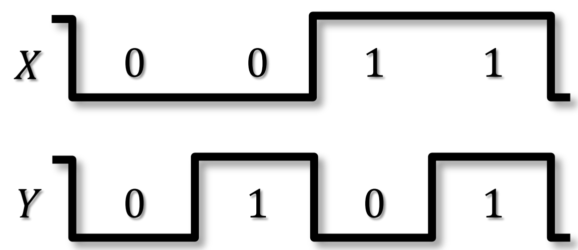


OR gate



NOT gate (inverter)

- **Time diagram (waveform behavior)**



(AND) $X \cdot Y$

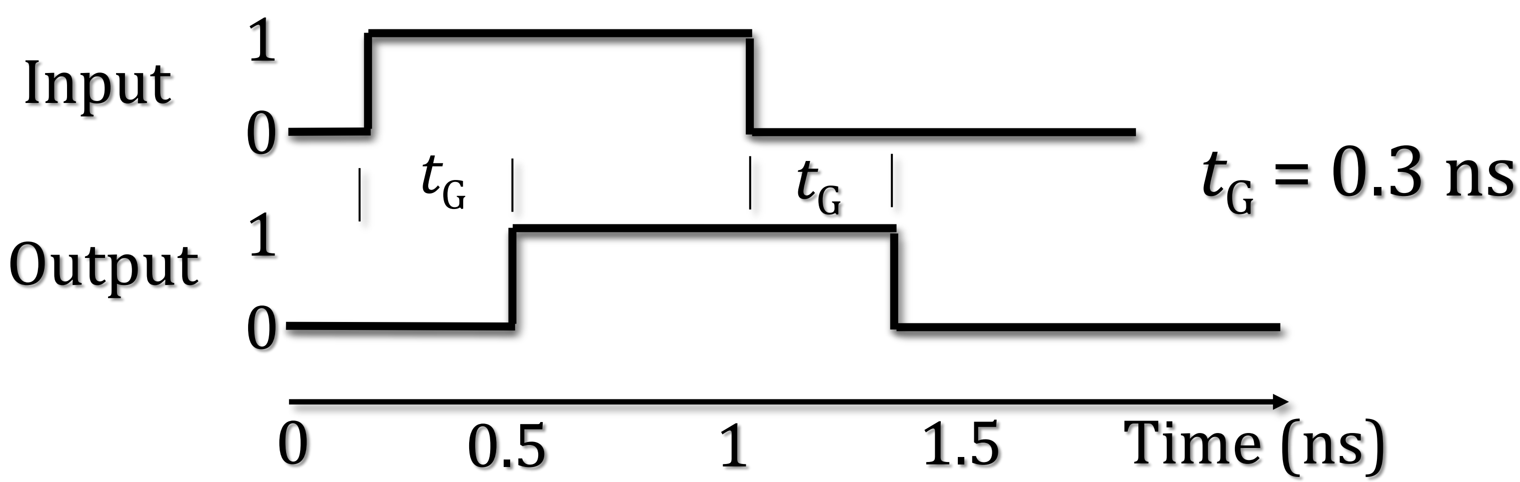
(OR) $X + Y$

(NOT) $\overline{X}$

Gate delay

- In reality, there is a gate delay

- Gate delay – the length of time it takes for an input changes to result in the corresponding output change.



- Gate delay is a function of gate type, number of inputs, underlying technology, and circuit design of the gate.

Boolean Algebra

- **A *Boolean Algebra* is an algebra dealing with binary variables and logic operations.**

- **A *Boolean expression* is an algebraic expression formed by using binary variables, the constants 0 and 1, the logic operation symbols, and parentheses**

- **The order of evaluation in a Boolean expression:**

( ), NOT, AND, OR

Boolean function

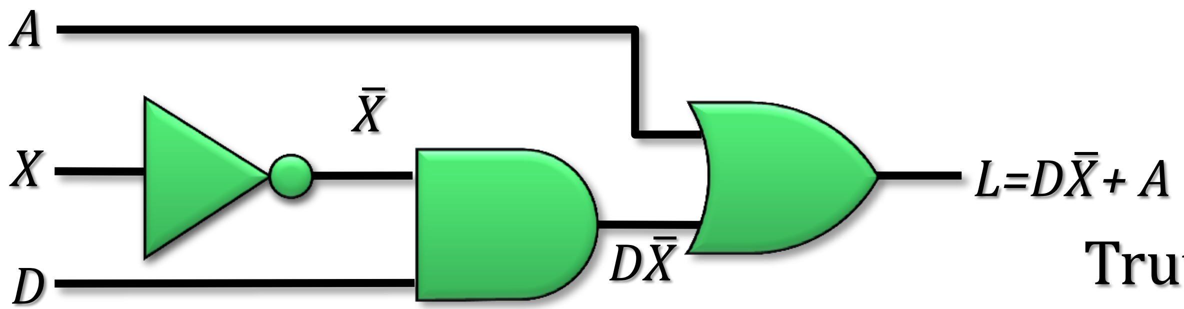
- **A *Boolean function*** **is a boolean equation consisting of a binary variable identifying the function followed by an equal sign and a boolean expression.**

$$\underbrace{L(D,X,A) = \overbrace{D\bar{X} + A}^{\text{Boolean expression}}}_{\text{Boolean function or Boolean equation}}$$

- **A boolean function can be transformed into a circuit diagram (logic diagram) composed of logic gates and interconnected by wires.**

Boolean function

Logic Diagram



Truth Table

D	X	A	L = D$\bar{X}$ + A
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 boolean function can be represented by a truth table**

Boolean function

- There is only one way that a boolean function can be represented by a truth table

- The Boolean function, however, can be expressed by various boolean equations, which are not the same but equivalent. Eg. L and F have the same function.

$$L(D,X,A) = D\overline{X} + A$$

$$F(D,X,A) = D\overline{X} + AD + A\overline{D}$$

- The boolean equation dictates the interconnection of gates in the logic circuit diagram.

- Simpler expression reduces the number of gates and the number of inputs into the gates

Basic identities of boolean algebra

1. $X + 0 = X$	2. $X \cdot 1 = X$	**Identity**
3. $X + 1 = 1$	4. $X \cdot 0 = 0$	**Null**
5. $X + X = X$	6. $X \cdot X = X$	**Idempotence**
7. $X + \overline{X} = 1$	8. $X \cdot \overline{X} = 0$	**Complementarity**
9. $\overline{\overline{X}} = X$		**Involution**

Basic identities of boolean algebra

10. $Y+X = X+Y$	11. $Y \cdot X = X \cdot Y$	Commutative
12. $X + (Y + Z) = (X+Y) + Z$	13. $(XY)Z = X(YZ)$	Associative
14. $X(Y + Z) = XY + XZ$	15. $X + YZ = (X + Y)(X + Z)$	Distributive
16. $\overline{X + Y} = \overline{X}\,\overline{Y}$	17. $\overline{XY} = \overline{X} + \overline{Y}$	De morgan's law

Basic identities of boolean algebra

- Any expression can replace the variable X in all boolean identity.

Eg. $X + 1 = 1$

with $X = AB + C$

$AB + C + 1 = 1$

- Identity 10 -14 are similar to ordinary algebra. However, Identity 15 does not hold in ordinary algebra.
- Identity 16 and 17 are De' morgan's rule and they are very important rules

X	Y	$\overline{X} + \overline{Y}$	$\overline{XY}$	$\overline{X + Y}$	$\overline{X}\,\overline{Y}$
0	0	1		1	
1	0	1		0	
0	1	1		0	
1	1	0		0	

Basic identities of boolean algebra

- **De'morgan's law can be extended to three or more variables. The general De'morgan's theorem can be expressed as**

$$\overline{X_1 + X_2 + \cdots + X_n} = \overline{X_1}\,\overline{X_2} \cdots \overline{X_n}$$

$$\overline{X_1 X_2 \ldots X_n} = \overline{X_1} + \overline{X_2} + \cdots + \overline{X_n}$$

Duality

- **The *dual* of an algebraic expression is obtained by interchanging + and · and interchanging 0's and 1's**
- **Any Boolean theorem that can be proven is thus also proven for its dual!**
- **Note that previous identities appear in dual pairs**
- **Example:**

$$XY + \bar{X}Z + YZ = XY + \bar{X}Z$$

Dual $(X+Y)(\bar{X} + Z)(Y+Z) = (X+Y)(\bar{X} + Z)$

Useful theorems of Boolean algebra

$X+XY = X$	$X(X+Y) = X$	Absorption
$XY + X\overline{Y} = X$	$(X+Y)(\overline{X}+Y)=X$	Minimization
$X+\overline{X}Y = X+Y$	$X(\overline{X}+Y) = XY$	Simplification
$XY + \overline{X}Z + YZ = XY + \overline{X}Z$	$(X+Y)(\overline{X}+Z)(Y+Z) = (X+Y)(\overline{X}+Z)$	Consensus

Proof of consensus theorem

- **Example: Prove the Consensus Theorem**

$$
\begin{aligned}
XY + \bar{X}Z + YZ &= XY + \bar{X}Z + YZ\cdot 1 && \textbf{(2)}\\
&= XY + \bar{X}Z + YZ(X + \bar{X}) && \textbf{(7)}\\
&= XY + \bar{X}Z + XYZ + \bar{X}YZ && \textbf{(14)}\\
&= XY + XYZ + \bar{X}Z + \bar{X}YZ && \textbf{(10)}\\
&= XY(1 + Z) + \bar{X}Z(1 + Y) && \textbf{(14)}\\
&= XY + \bar{X}Z && \textbf{(3)}
\end{aligned}
$$

Expression simplification

- **A *literal*** **is a complemented or uncomplemented variable in a term**
- **Simplify the following expression to contain the smallest number of literals:**

$$AB + ACD + \overline{A}BD + AC\overline{D} + \overline{A}BCD$$

$$= AB + \overline{A}BCD + ACD + AC\overline{D} + \overline{A}BD$$

$$= AB + \overline{A}BCD + AC(D + \overline{D}) + \overline{A}BD$$

$$= AB + AC + \overline{A}BD = B(A + \overline{A}D) + AC$$

$$= B(A + D) + AC$$

- **Only 5 literals!**

Complement of a function

- The complement of a function F , $\overline{F,}$ can be obtained in two ways:
  - By applying DeMorgan's theorem
  - By taking the dual of the function and complementing each literal

Example: $F = \bar{X}Y\bar{Z} + \bar{X}\bar{Y}Z$

Standard Forms

- *Standard forms* are standard ways to express Boolean functions
- Contain product terms (Minterms) and sum terms (Maxterms)
- Result in more desirable expression for circuit implementation

Minterms:

- Minterms are AND terms with every variable present in either true or complement form
- Every variable combination in a truth table has a corresponding minterm

Minterms

- **For a function with *n* variables, there will be 2^n minterms**
- **The literals are listed in the same order for all minterms (usually alphabetically)**

X	Y	Z	Product Term	symbols	m_o	m_1	m_2	m_3	m_4	m_5	m_6	m_7
0	0	0	$\bar{X}\bar{Y}\bar{Z}$	m_o	1	0	0	0	0	0	0	0
0	0	1	$\bar{X}\bar{Y}Z$	m_1	0	1	0	0	0	0	0	0
0	1	0	$\bar{X}Y\bar{Z}$	m_2	0	0	1	0	0	0	0	0
0	1	1	$\bar{X}YZ$	m_3	0	0	0	1	0	0	0	0
1	0	0	$X\bar{Y}\bar{Z}$	m_4	0	0	0	0	1	0	0	0
1	0	1	$X\bar{Y}Z$	m_5	0	0	0	0	0	1	0	0
1	1	0	$XY\bar{Z}$	m_6	0	0	0	0	0	0	1	0
1	1	1	XYZ	m_7	0	0	0	0	0	0	0	1

Minterms

X	Y	Z	Product Term	symbols	m_o	m_1	m_2	m_3	m_4	m_5	m_6	m_7
0	0	0	$\bar{X}\bar{Y}\bar{Z}$	m_o	1	0	0	0	0	0	0	0
0	0	1	$\bar{X}\bar{Y}Z$	m_1	0	1	0	0	0	0	0	0
0	1	0	$\bar{X}Y\bar{Z}$	m_2	0	0	1	0	0	0	0	0
0	1	1	$\bar{X}YZ$	m_3	0	0	0	1	0	0	0	0
1	0	0	$X\bar{Y}\bar{Z}$	m_4	0	0	0	0	1	0	0	0
1	0	1	$X\bar{Y}Z$	m_5	0	0	0	0	0	1	0	0
1	1	0	$XY\bar{Z}$	m_6	0	0	0	0	0	0	1	0
1	1	1	XYZ	m_7	0	0	0	0	0	0	0	1

- **A variable in a minterm *is complemented* for 0 and is *not complemented* for 1**
- **The symbol index corresponds to the binary combination of the variables**

Sum-of-Minterms Canonical Form

- Express a function as a sum of minterms
- Logical sum (*OR*) of all minterms where the function value is 1

- Example:

X	Y	Z	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Maxterms

- Similarly, maxterms are OR terms with every variable present
- In maxterms, a variable *is complemented* for a 1 and *not complemented* for a 0

X	Y	Z	Sum Term	symbols	M_0	M_1	M_2	M_3	M_4	M_5	M_6	M_7
0	0	0	$X+Y+Z$	M_0	0	1	1	1	1	1	1	1
0	0	1	$X+Y+\bar{Z}$	M_1	1	0	1	1	1	1	1	1
0	1	0	$X+\bar{Y}+Z$	M_2	1	1	0	1	1	1	1	1
0	1	1	$X+\bar{Y}+\bar{Z}$	M_3	1	1	1	0	1	1	1	1
1	0	0	$\bar{X}+Y+Z$	M_4	1	1	1	1	0	1	1	1
1	0	1	$\bar{X}+Y+\bar{Z}$	M_5	1	1	1	1	1	0	1	1
1	1	0	$\bar{X}+\bar{Y}+Z$	M_6	1	1	1	1	1	1	0	1
1	1	1	$\bar{X}+\bar{Y}+\bar{Z}$	M_7	1	1	1	1	1	1	1	0

Maxterms

- The symbol index corresponds to the binary combination of the variables
- For a function with *n* variables, there will be 2^n maxterms

X	Y	Z	Sum Term	symbols	M_0	M_1	M_2	M_3	M_4	M_5	M_6	M_7
0	0	0	$X+Y+Z$	M_0	0	1	1	1	1	1	1	1
0	0	1	$X+Y+\bar{Z}$	M_1	1	0	1	1	1	1	1	1
0	1	0	$X+\bar{Y}+Z$	M_2	1	1	0	1	1	1	1	1
0	1	1	$X+\bar{Y}+\bar{Z}$	M_3	1	1	1	0	1	1	1	1
1	0	0	$\bar{X}+Y+Z$	M_4	1	1	1	1	0	1	1	1
1	0	1	$\bar{X}+Y+\bar{Z}$	M_5	1	1	1	1	1	0	1	1
1	1	0	$\bar{X}+\bar{Y}+Z$	M_6	1	1	1	1	1	1	0	1
1	1	1	$\bar{X}+\bar{Y}+\bar{Z}$	M_7	1	1	1	1	1	1	1	0

Maxterms

- The literals are listed in the same order for all minterms (usually alphabetically)
- Every variables combination in a truth table has a corresponding minterm

X	Y	Z	Sum Term	symbols	M_o	M_1	M_2	M_3	M_4	M_5	M_6	M_7
0	0	0	$X+Y+Z$	M_o	0	1	1	1	1	1	1	1
0	0	1	$X+Y+\bar{Z}$	M_1	1	0	1	1	1	1	1	1
0	1	0	$X+\bar{Y}+Z$	M_2	1	1	0	1	1	1	1	1
0	1	1	$X+\bar{Y}+\bar{Z}$	M_3	1	1	1	0	1	1	1	1
1	0	0	$\bar{X}+Y+Z$	M_4	1	1	1	1	0	1	1	1
1	0	1	$\bar{X}+Y+\bar{Z}$	M_5	1	1	1	1	1	0	1	1
1	1	0	$\bar{X}+\bar{Y}+Z$	M_6	1	1	1	1	1	1	0	1
1	1	1	$\bar{X}+\bar{Y}+\bar{Z}$	M_7	1	1	1	1	1	1	1	0

Product-of-Maxterms Canonical Form

- **Express a function as a product of maxterms**
- **Logical product (*AND*) of all maxterms where the function value is 0**
- **Example:**

X	Y	Z	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Minterms and Maxterms Relationship

- **Review: DeMorgan's Theorem**

- **Two-variable example:**

Function Complements

- The complement of a function expressed as a sum of minterms is constructed by selecting the minterms missing in the expression

- Alternatively, the complement of a function expressed as a sum of minterms is simply the product of maxterms with the same indices

Function Complement Example

- **Find complement expressions for the function:**

$$G(X, Y, Z) = \sum m(1, 3, 5, 7)$$

X	Y	Z	G
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

- **As a sum of minterms:**

- **As a product of maxterms:**

Sum-of-Products

- ***Sum of Products*** **form: equations are written as an OR of AND terms**

- **Similar to sum of minterms but does not need to contain all variables in every term**

- **Can be directly implemented as a *two-level circuit***

Sum-of-Products Implementation

- **Example - the function:**

$$F = \overline{Y} + \overline{X}Y\overline{Z} + XY$$

Product-of-Sums

- ***Product of Sums*** **form: equations are written as an AND of OR terms**
- **Can also be implemented as a two-level circuit**
- **Example – the function:**

$$F = X(\overline{Y} + Z)(X + Y + \overline{Z})$$

is expressed as a POS

Cost Criteria

- **Need a method to measure the complexity of a logic circuit**
- **Define the *Gate-Input Cost* as the number of inputs to the gates in the implementation**
- **Can be calculated directly from the equation corresponding to the circuit**
- **Only count distinct terms and complemented literals**

Gate-Input Cost

- **Example:**

$$F = AB + \bar{C}(D + E)$$

- **The corresponding logic diagram:**