Chapter 2 – Combinational Digital Circuits

Part 1 – Gate Circuits and Boolean Equations

Overview

- Part 1 – Gate Circuits and Boolean Equations
  - Binary Logic and Gates
  - Boolean Algebra
  - Standard Forms
- Part 2 – Circuit Optimization
  - Two-Level Optimization
  - Map Manipulation
  - Practical Optimization (Espresso)
  - Multi-Level Circuit Optimization
- Part 3 – Additional Gates and Circuits
  - Other Gate Types
  - Exclusive-OR Operator and Gates
  - High-Impedance Outputs

Binary Logic and Gates

- Binary variables take on one of two values.
- Logical operators operate on binary values and binary variables.
- Basic logical operators are the logic functions AND, OR and NOT.
- Logic gates implement logic functions.
- Boolean Algebra: a useful mathematical system for specifying and transforming logic functions.
- We study Boolean algebra as a foundation for designing and analyzing digital systems!

Binary Variables

- Recall that the two binary values have different names:
  - True/False
  - On/Off
  - Yes/No
  - 1/0
- We use 1 and 0 to denote the two values.
- Variable identifier examples:
  - A, B, y, z, or X_1 for now
  - RESET, START_IT, or ADD1 later

Logical Operations

- The three basic logical operations are:
  - AND
  - OR
  - NOT
- AND is denoted by a dot (·).
- OR is denoted by a plus (+).
- NOT is denoted by an overbar (¯), a single quote mark (') after, or (~) before the variable.

Notation Examples

- Examples:
  - $\mathbf{Y = A \cdot B}$ is read "Y is equal to A AND B."
  - $\mathbf{z = x + y}$ is read "z is equal to x OR y."
  - $\mathbf{X = \overline{A}}$ is read "X is equal to NOT A."
- Note: The statement:

  1 + 1 = 2 (read "one <u>plus</u> one equals two")

  is not the same as

  1 + 1 = 1 (read "1 <u>or</u> 1 equals 1").

Operator Definitions

- Operations are defined on the values "0" and "1" for each operator:

AND	OR	NOT
$0 \cdot 0 = 0$	$0 + 0 = 0$	$\overline{0} = 1$
$0 \cdot 1 = 0$	$0 + 1 = 1$	$\overline{1} = 0$
$1 \cdot 0 = 0$	$1 + 0 = 1$	
$1 \cdot 1 = 1$	$1 + 1 = 1$	

Truth Tables

- ***Truth table -*** a tabular listing of the values of a function for all possible combinations of values on its arguments
- **Example:** Truth tables for the basic logic operations:

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

OR		
X	Y	$Z = X+Y$
0	0	0
0	1	1
1	0	1
1	1	1

NOT	
X	$Z = \overline{X}$
0	1
1	0

Boolean Algebra

- B={0,1} kümesi üzerinde tanımlı
- İkili İşlemler : VE, VEYA (·, +)
- Birli İşlem: TÜMLEME (¯)

Axioms

Let a, b, c $\in$ B

1. Closure:	$a + b = c$	$a \cdot b=c$
2. Commutative:	$a + b = b + a$	$a \cdot b=b \cdot a$
3. Distributive:	$a+(b \cdot c)=(a+b) \cdot (a+c)$	$a \cdot (b+c)=a \cdot b+a \cdot c$
4. Associative:	$a+(b+c)=(a+b)+c$	$a \cdot (b \cdot c)=(a \cdot b) \cdot c$
5. Neutral Element:	$a+0=a$	$a \cdot 1=a$
6. Inverse:	$a+a'=1$	$a \cdot a'=0$

Boolean Operator Precedence

- **The order of evaluation in a Boolean expression is:**
  1. Parentheses
  2. NOT
  3. AND
  4. OR
- **Consequence: Parentheses appear around OR expressions**
- **Example:** $F = A(B + C)(C + \overline{D})$

Properties and Theorems

- These properties and theorems can be proved by using the axioms of Boole algebra.

1. Identity element: $a+1=1$ $\quad a \cdot 0=0$
2. Transformation: $(a')'=a$
3. Constant power: $a+a+\ldots+a=a$ $\quad a \cdot a \cdot \ldots \cdot a=a$
4. Absoption: $a+a \cdot b=a$ $\quad a \cdot (a+b)=a$
5. De Morgan's Theorem:
   $(a+b)'=a' \cdot b'$ $\quad (a \cdot b)'=a'+b'$
6. General De Morgan's Theorem:
   $f'(X1,X2,\ldots,Xn,0,1,+,\cdot) \Leftrightarrow f(X1',X2',\ldots,Xn',1,0,\cdot,+)$

Example 1: Boolean Algebraic Proof

- **$A + A \cdot B = A$ (Absorption Theorem)**

Proof Steps	**Axiom or Theorem**
$A + A \cdot B$	
$= A \cdot 1 + A \cdot B$	$X = X \cdot 1$
$= A \cdot (1 + B)$	$X \cdot Y + X \cdot Z = X \cdot (Y + Z)$ (Distributive Law)
$= A \cdot 1$	$1 + X = 1$
$= A$	$X \cdot 1 = X$

- Our primary reason for doing proofs is to learn:
  - Careful and efficient use of the identities and theorems of Boolean algebra, and
  - How to choose the appropriate identity or theorem to apply to make forward progress, irrespective of the application.

Example 2: Boolean Algebraic Proofs

- $AB + A'C + BC = AB + A'C$ (Consensus Theorem)

Proof Steps	Axiom or Theorem
$AB + A'C + BC$	
$= AB + A'C + 1 \cdot BC$	$1 . X = X$
$= AB + A'C + (A + A') \cdot BC$	$X + X' = 1$
$= AB + A'C + ABC + A'BC$	$X(Y + Z) = XY + XZ$
$= AB + ABC + A'C + A'BC$	$X + Y = Y + X$
$= AB + A'(C + BC)$	$X(Y + Z) = XY + XZ$
$= AB + A'C$	$X + X \cdot Y = X$

Example 3: Boolean Algebraic Proofs

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

Proof Steps	Axiom or Theorem
$(\overline{X+Y})Z + X\overline{Y}$	
$= X'Y'Z+XY'$	$(X+Y)'=X'Y'$
$= (X'Z+X)Y'$	$(X+Y)Z=XZ+YZ$
$= (X'+X)\ (Z+X)Y'$	$XY+Z=(X+Z)(Y+Z)$
$= (Z+X)Y'$	$X'+X=1$

Boolean Function Evaluation

$F1 = xy\bar{z}$
$F2 = x + \bar{y}z$
$F3 = \bar{x}\bar{y}\bar{z} + \bar{x}\,y\,z + x\bar{y}$
$F4 = x\bar{y} + \bar{x}\,z$
$F4' = (xy' + x'z)' = (x'+y)(x+z')$

- If the the input number is = *n*
- There is 2^n different input combinations

- Hence, 2^{2^n} different Boolean functions can be defined

x	y	z	F1	F2	F3	F4	F4'
0	0	0	0	0	1	0	1
0	0	1	0	1	0	1	0
0	1	0	0	0	0	0	1
0	2^{2^n} 1	1	0	0	1	1	0
1	0	0	0	1	1	1	0
1	0	1	0	1	1	1	0
1	1	0	1	1	0	0	1
1	1	1	0	1	0	0	1

Expression Simplification

- Simplify to contain the smallest number of literals:

$$AB + \overline{A}CD + \overline{A}BD + \overline{A}C\overline{D} + ABCD$$

$= AB + ABCD + A'CD + A'CD' + A'BD$

$= AB + AB(CD) + A'C(D + D') + A'BD$

$= AB + A'C + A'BD$

$= B(A + A'D) + A'C$

$= B(A + A')(A + D) + A'C$

$= B(A + D) + AC$

5 literals

Complementing Functions

- Use DeMorgan's Theorem to complement a function:
  1. Interchange AND and OR operators
  2. Complement each constant value and literal
- Example: Complement F = $\overline{x}y\overline{z} + x\overline{y}\overline{z}$

  $F \underline{=} (x + y' + z)(x + y' + z')$
- Example: Complement G = (a + bc)d′ + e′

  G =

Overview – Canonical Forms

- What are Canonical Forms?
- Minterms and Maxterms
- Index Representation of Minterms and Maxterms
- Sum-of-Products (SOP) Representations
- Product-of-Sums (POS) Representations
- Representation of Complements of Functions
- Conversions between Representations

Canonical Forms

- It is useful to specify Boolean functions in a form that:
  - Allows comparison for equality.
  - Has a correspondence to the truth tables
- Canonical Forms in common usage:
  - Sum of Products (SOP)
  - Product of Sums (POS)

Minterms

- Minterms are AND terms with every variable present in either true or complemented form.
- Given that each binary variable may appear normal (e.g., x) or complemented (e.g., $\overline{x}$), there are 2^n minterms for n variables.
- Example: Two variables (X and Y) produce 2 x 2 = 4 combinations:
  - XY (both normal)
  - $X\overline{Y}$ (X normal, Y complemented)
  - $\overline{X}Y$ (X complemented, Y normal)
  - $\overline{X}\,\overline{Y}$ (both complemented)
- Thus there are four minterms of two variables.

$x'y'z'$
$x'y'z$
$x'yz'$
$xy'z'$
xyz'
$xy'z$
$x'yz$
xyz

Minterms (AND)

$x+y+z$
$x+y+z'$
$x+y'+z$
$x'+y+z$
$x+y'+z'$
$x'+y+z'$
$x'+y'+z$
$x'+y'+z'$

Maxterms (OR)

Sıralarını yanlış yazmışım

Index 5: $(101)_2 = 5_{10}$

$xy'z$ — Minterm

$x'+y+z'$ — Maxterm

To transform minterm to maxterm (or otherwise) apply De Morgans Theorm

Maxterms

- Maxterms are OR terms with every variable in true or complemented form.
- Given that each binary variable may appear normal (e.g., x) or complemented (e.g., $\overline{x}$), there are 2^n maxterms for n variables.
- Example: Two variables (X and Y) produce 2 x 2 = 4 combinations:

  $X + Y$ (both normal)

  $X + \overline{Y}$ (x normal, y complemented)

  $\overline{X} + Y$ (x complemented, y normal)

  $\overline{X} + \overline{Y}$ (both complemented)

Maxterms and Minterms

- Examples: Two variable minterms and maxterms.

Index	Minterm	Maxterm
0	$\overline{x}\ \overline{y}$	$x + y$
1	$\overline{x}\ y$	$x + \overline{y}$
2	$x\ \overline{y}$	$\overline{x} + y$
3	$x\ y$	$\overline{x} + \overline{y}$

- The index above is important for describing which variables in the terms are true and which are complemented.

Standard Order

- Minterms and maxterms are designated with a subscript
- The subscript is a number, corresponding to a binary pattern
- The bits in the pattern represent the complemented or normal state of each variable listed in a standard order.
- All variables will be present in a minterm or maxterm and will be listed in the same order (usually alphabetically)
- Example: For variables a, b, c:
  - Maxterms: $(a + b + \bar{c})$, $(a + b + c)$
  - Terms: $(b + a + c)$, $a\ \bar{c}\ b$, and $(c + b + a)$ are NOT in standard order.
  - Minterms: $a\ \bar{b}\ c$, $a\ b\ c$, $\bar{a}\ \bar{b}\ c$
  - Terms: $(a + c)$, $\bar{b}\ c$, and $(\bar{a} + b)$ do not contain all variables

Purpose of the Index

- The index for the minterm or maxterm, expressed as a binary number, is used to determine whether the variable is shown in the true form or complemented form.
- For Minterms:
  - “1” means the variable is “Not Complemented” and
  - “0” means the variable is “Complemented”.
- For Maxterms:
  - “0” means the variable is “Not Complemented” and
  - “1” means the variable is “Complemented”.

Index Example in Three Variables

- Assume the variables are called X, Y, and Z.
- The standard order is X, then Y, then Z.
- The Index 0 (base 10) = 000 (base 2) for three variables).
- All three variables are complemented for minterm 0 ($\overline{X},\overline{Y},\overline{Z}$) and no variables are complemented for Maxterm 0 (X,Y,Z).
  - Minterm 0, called m_0 is $\overline{X}\overline{Y}\overline{Z}$.
  - Maxterm 0, called M_0 is $(X + Y + Z)$.
  - Minterm 6 ?
  - Maxterm 6 ?

Index Examples – Four Variables

Index	Binary	Minterm	Maxterm
i	Pattern	m_i	M_i
0	0000	$\overline{a}\,\overline{b}\,\overline{c}\,\overline{d}$	$a + b + c + d$
1	0001	$\overline{a}\,\overline{b}\,\overline{c}\,d$	?
3	0011	?	$a + b + \overline{c} + \overline{d}$
5	0101	$\overline{a}\,b\,\overline{c}\,d$	$a + \overline{b} + c + \overline{d}$
7	0111	?	$a + \overline{b} + \overline{c} + \overline{d}$
10	1010	$a\,\overline{b}\,c\,\overline{d}$	$\overline{a} + b + \overline{c} + d$
13	1101	$a\,b\,\overline{c}\,d$	?
15	1111	$a\,b\,c\,d$	$\overline{a} + \overline{b} + \overline{c} + \overline{d}$

Minterm and Maxterm Relationship

- Review: DeMorgan's Theorem
  $\overline{x \cdot y} = \overline{x} + \overline{y}$ and $\overline{x + y} = \overline{x}\ \overline{y}$
- Two-variable example:
  $M_2 = \overline{x} + y$ and $m_2 = x \cdot \overline{y}$
  Thus M_2 is the complement of m_2 and vice-versa.
- Since DeMorgan's Theorem holds for *n* variables, the above holds for terms of *n* variables
- giving: $M_i = \overline{m_i}$ $m_i = \overline{M_i}$

Thus M_i is the complement of m_i.

Minterm Function Example

- Find the truth table of $F_1 = m_1 + m_4 + m_7$
- $F_1 = \overline{x}\ \overline{y}\ z + x\ \overline{y}\ \overline{z} + x\ y\ z$

x y z	index	m_1	+	m_4	+	m_7	$= F_1$
0 0 0	0	0	+	0	+	0	= 0
0 0 1	1	1	+	0	+	0	= 1
0 1 0	2	0	+	0	+	0	= 0
0 1 1	3	0	+	0	+	0	= 0
1 0 0	4	0	+	1	+	0	= 1
1 0 1	5	0	+	0	+	0	= 0
1 1 0	6	0	+	0	+	0	= 0
1 1 1	7	0	+	0	+	1	= 1

$f(x, y, z) = m_1 + m_4 + m_7$

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

001 100 111

$f(x, y, z) = xy + z'$ this can not be written in form of minterm

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

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

$= m_7 + m_6 + m_4 + m_2 + m_0$

You can convert EVERY fon to minterm form

Minterm Function Example

00010 01001 10001 10111

- $F(A, B, C, D, E) = m_2 + m_9 + m_{17} + m_{23}$ 4 2 1 0 16+4+2+1
- $F(A, B, C, D, E) = A'B'C'DE' + A'BC'D'E + AB'C'D'E + AB'CDE$

$A'B'C'DE' + A'BC'D'E + AB'C'D'E + AB'CDE$

Maxterm Function Example

- Implement F_1 in maxterms:

$$F_1 = M_0 \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6$$

$$F_1 = (x + y + z) \cdot (x + \overline{y} + z) \cdot (x + \overline{y} + \overline{z}) \cdot (\overline{x} + y + \overline{z}) \cdot (\overline{x} + \overline{y} + z)$$

x y z	i	$M_0 \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6 = F_1$
0 0 0	0	$0 \cdot 1 \cdot 1 \cdot 1 \cdot 1 = 0$
0 0 1	1	$1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 = 1$
0 1 0	2	$1 \cdot 0 \cdot 1 \cdot 1 \cdot 1 = 0$
0 1 1	3	$1 \cdot 1 \cdot 0 \cdot 1 \cdot 1 = 0$
1 0 0	4	$1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 = 1$
1 0 1	5	$1 \cdot 1 \cdot 1 \cdot 0 \cdot 1 = 0$
1 1 0	6	$1 \cdot 1 \cdot 1 \cdot 1 \cdot 0 = 0$
1 1 1	7	$1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 = 1$

Ex: Find $F(x, y, z) = M_0 M_2 M_3 M_5 M_6$

$M_0 : 000$
$M_2 : 010$
$M_3 : 011$
$M_5 : 101$
$M_6 : 110$

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

x	y	z	index	M_0	M_2	M_3	M_5	M_6	=	$F(x,y,z)$
0	0	0	0	0	1	1	1	1	=	0
0	0	1	1	1	1	1	1	1	=	1
0	1	0	2	1	0				=	0
0	1	1	3	1	1	0			=	0
1	0	0	4							

Maxterm Function Example

- $F(A,B,C,D) = \overset{0011}{M_3} \times \overset{1000}{M_8} \times \overset{1011}{M_{11}} \times \overset{1110}{M_{14}}$
- $F(A, B,C,D) = (A+B+C'+D')(A'+B+C+D)(A'+B+C'+D')(A'+B'+C'+D)$

$(A+B+C'+D')(A'+B+C+D)(A'+B+C'+D')(A'+B'+C'+D)$

$(A+B+C'+D')(A'+B+C+D)(A'+B+C'+D')(A'+B'+C'+D)$

Canonical Sum of Products

- Any Boolean function can be expressed as a Sum of Minterms.
  - For the function table, the minterms used are the terms corresponding to the 1's
  - For expressions, expand all terms first to explicitly list all minterms. Do this by "ANDing" any term missing a variable v with a term ($v + \overline{v}$).
- Example: Implement $f = x + \overline{x}\ \overline{y}$ as a sum of minterms.

  First expand terms: $f = x(y + \overline{y}) + \overline{x}\ \overline{y}$

  Then distribute terms: $f = xy + x\overline{y} + \overline{x}\,\overline{y}$

  Express as sum of minterms: $f = m_3 + m_2 + m_0$

Another SOP Example

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

- There are three variables, A, B, and C which we take to be the standard order.

- Expanding the terms with missing variables:

$F = A(B + B')(C + C') + (A + A')\,B'C$

- Distributing the literals over parenthesis

$= ABC + ABC' + AB'C + AB'C' + AB'C + A'B'C$

- Collect terms

$= ABC + ABC' + AB'C + AB'C' + A'B'C$

- Express as SOM:

$= m_7 + m_6 + m_5 + m_4 + m_1 = m_1 + m_4 + m_5 + m_6 + m_7$

Shorthand SOP Form

- From the previous example, we started with:

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

- We ended up with:

$F = m_1 + m_4 + m_5 + m_6 + m_7$

- This can be denoted in the formal shorthand:

  $$\mathbf{F(A, B, C) = \Sigma_m(1,4,5,6,7)}$$

- Note that we explicitly show the standard variables in order and drop the "m" designators.

Canonical Product of Sums

- Any Boolean Function can be expressed as a Product of Sums (POS).
  - For the function table, the maxterms used are the terms corresponding to the 0's.
  - For an expression, expand all terms first to explicitly list all maxterms. Do this by first applying the second distributive law , "ORing" terms missing variable v with a term equal to $v \cdot \overline{v}$ and then applying the distributive law again.
- Example: Convert to product of maxterms:

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

Apply the distributive law:

$$x + \overline{x}\,\overline{y} = (x + \overline{x})(x + \overline{y}) = 1 \cdot (x + \overline{y}) = x + \overline{y}$$

Add missing variable z:

$$x + \overline{y} + z \cdot \overline{z} = (x + \overline{y} + z)(x + \overline{y} + \overline{z})$$

Express as POS: $f = M_2 \cdot M_3$

Another POS Example

- Find Product of Sums representation of f :

$f=AC'+BC+A'B'$

- $f=(AC'+BC+A')(AC'+BC+B')$

$f=((AC'+B)(AC'+C)+A')((AC'+B)(AC'+C)+B')$

$f=((A+B)(C'+B)(A+C)(C'+C)+A')((A+B)(C'+B)(A+C)(C'+C)+B')$

$f=((A+B)(C'+B)(A+C)+A')((A+B)(C'+B)(A+C)+B')$

$f=(A+B+A')(C'+B+A')(A+C+A')(A+B+B')(C'+B+B')(A+C+B')$

$f=(A'+B+C')(A+B'+C)$

$f = M_5 \cdot M_2$

$$F(A,B,C) = A(B'+C)$$

$$= (A+BB'+CC')(AA'+B'+C)$$

$$= \Big(\underline{(A+B)(A+B')+CC'}\Big)\,(A+B'+C)(A'+B'+C)$$

Disturbute

$$= (A+B+C)(A+B'+C)(A+B+C')(A+B'+C')(A+B'+C)$$

Function Complements

- The complement of a function expressed as a SOP is constructed by selecting the minterms missing in the SOP canonical forms.
- Alternatively, the complement of a function expressed by a SOP form is simply the POS with the same indices.

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

- Example: Given

$$F(x, y, z) = \Sigma_m(1,3,5,7)$$

001 011 101 111

$$\bar{F}(x, y, z) = \Sigma_m(0,2,4,6)$$

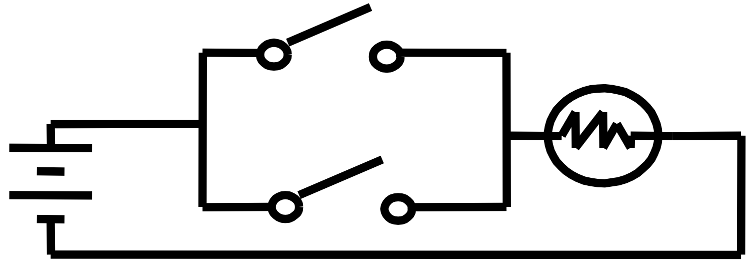
$$\bar{F}(x, y, z) = \Pi_M(1,3,5,7)$$

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

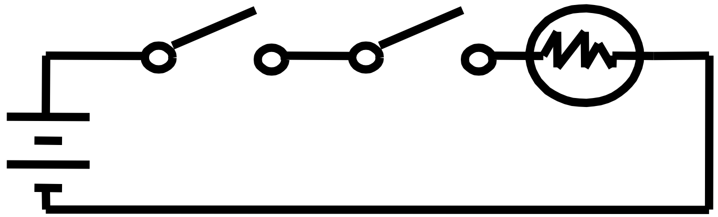
Implementation of Boolean Functions

- Using Switches
  - For inputs:
    - logic 1 is switch closed
    - logic 0 is switch open
  - For outputs:
    - logic 1 is light on
    - logic 0 is light off.
  - NOT uses a switch such that:
    - logic 1 is switch open
    - logic 0 is switch closed

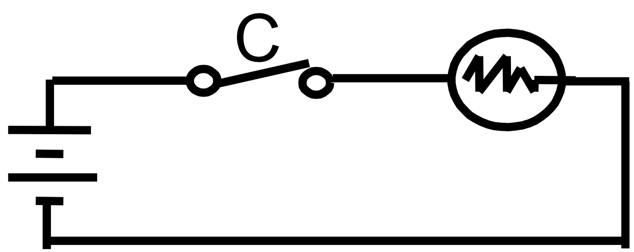
Switches in parallel => OR



Switches in series => AND

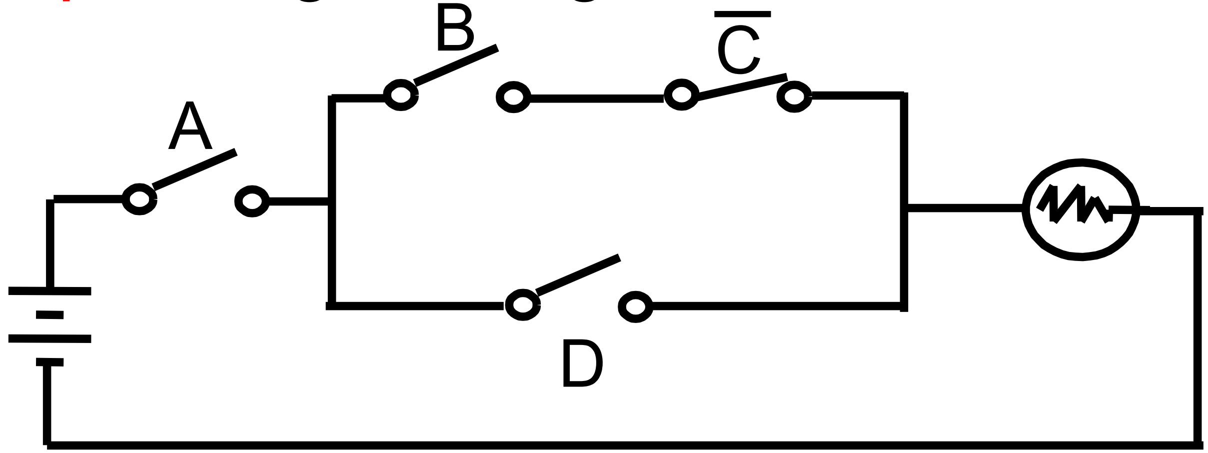


Normally-closed switch => NOT



Implementation of Boolean Functions (Continued)

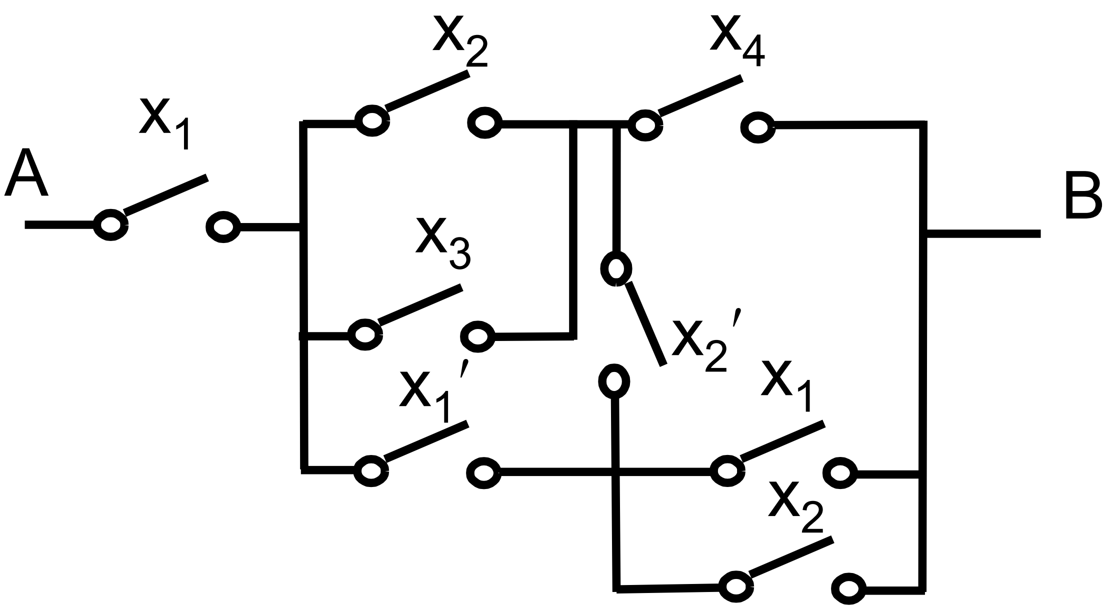
- Example: Logic Using Switches



- Light is on ($L = 1$) and off ($L = 0$), otherwise.
  - Sum of path functions:
    - $L(A, B, C, D) = ABC' + AD$
  - Product of cut functions:
    - $f(A, B, C, D) = A\,(B+D)\,(C'+D)$ $BC' + D = (B+D)(C'+D)$

f_{AB}

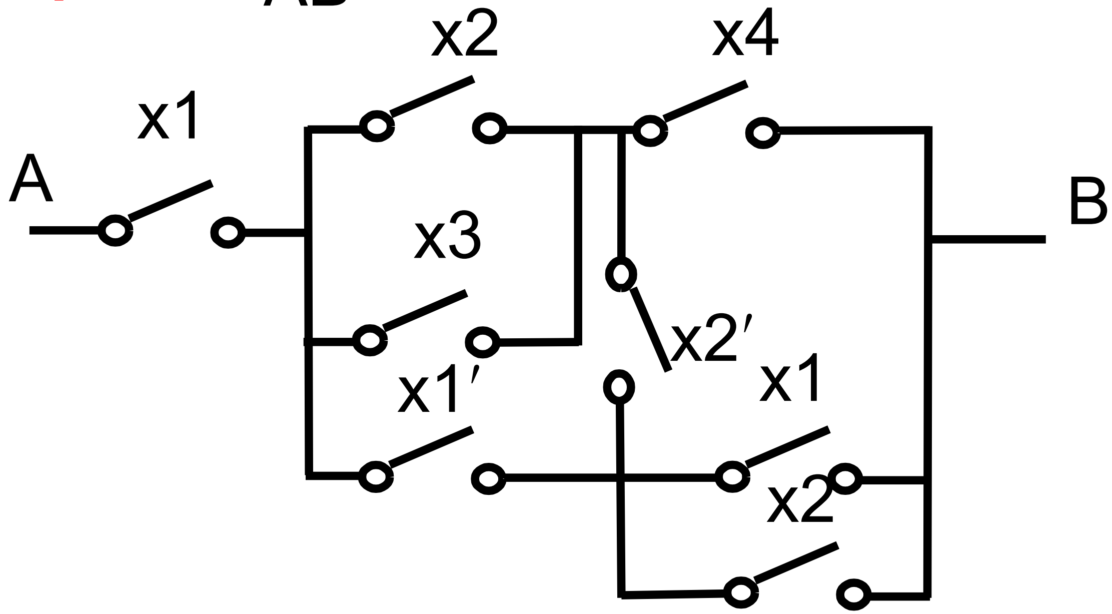
Example: f_{AB}=?



	x_1	x_2	x_3	x_4	f_{AB}
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
✓10	1	0	1	0	1
✓11	1	0	1	1	1
12	1	1	0	0	0
✓13	1	1	0	1	1
14	1	1	1	0	0
✓15	1	1	1	1	1

- $f_{AB}=\Sigma_m(10,11,13,15)$
- $f_{AB}=\Pi_M(0,1,2,3,4,5,6,7,8,9,12,14)$

Example: f_{AB}=?



- Sum of path functions:

f_{AB}
$= x_1x_2x_4+x_1x_2x_2'x_1+x_1x_2x_2'x_2+x_1x_3x_4+x_1x_3x_2'x_1+x_1x_3x_2'x_2+x_1x_1'x_2'x_4+x_1x_1'x_1+x_1x_1'x_2$
$= x_1x_2x_4+x_1x_3x_4+x_1x_3x_2'$

- Product of cut functions:

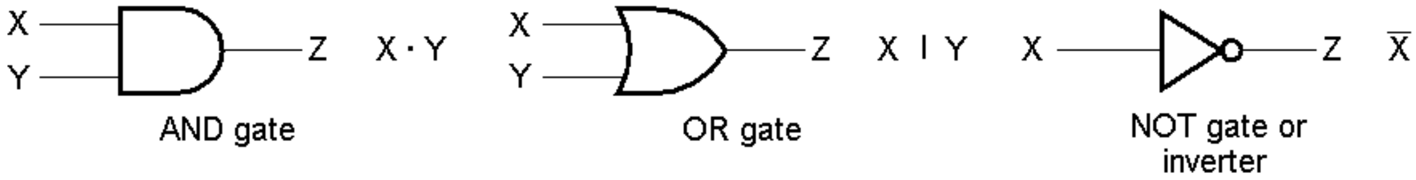
f_{AB}
$= x_1(x_2+x_3+x_1')(x_2+x_3+x_2'+x_1+x_2)(x_4+x_2'+x_1')(x_4+x_1+x_2)$
$= x_1(x_2+x_3+x_1')(x_4+x_2'+x_1')(x_4+x_1+x_2)$

Logic Gates

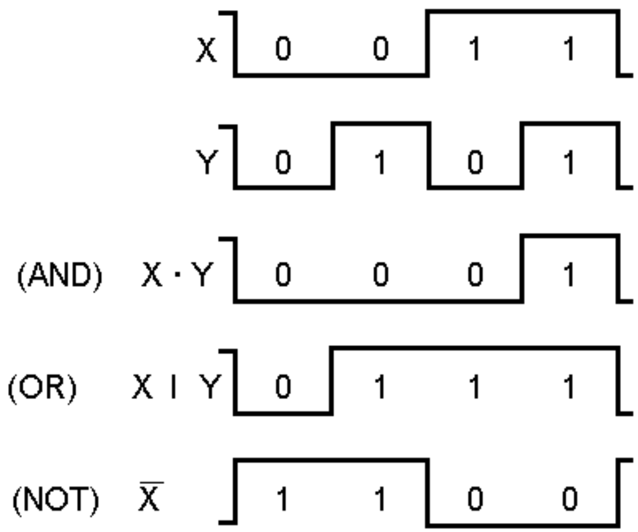
- In the earliest computers, switches were opened and closed by magnetic fields produced by energizing coils in *relays*. The switches in turn opened and closed the current paths.
- Later, *vacuum tubes* that open and close current paths electronically replaced relays.
- Today, *transistors* are used as electronic switches that open and close current paths.

Logic Gate Symbols and Behavior

- Logic gates have special symbols.
- And waveform behavior in time follows:



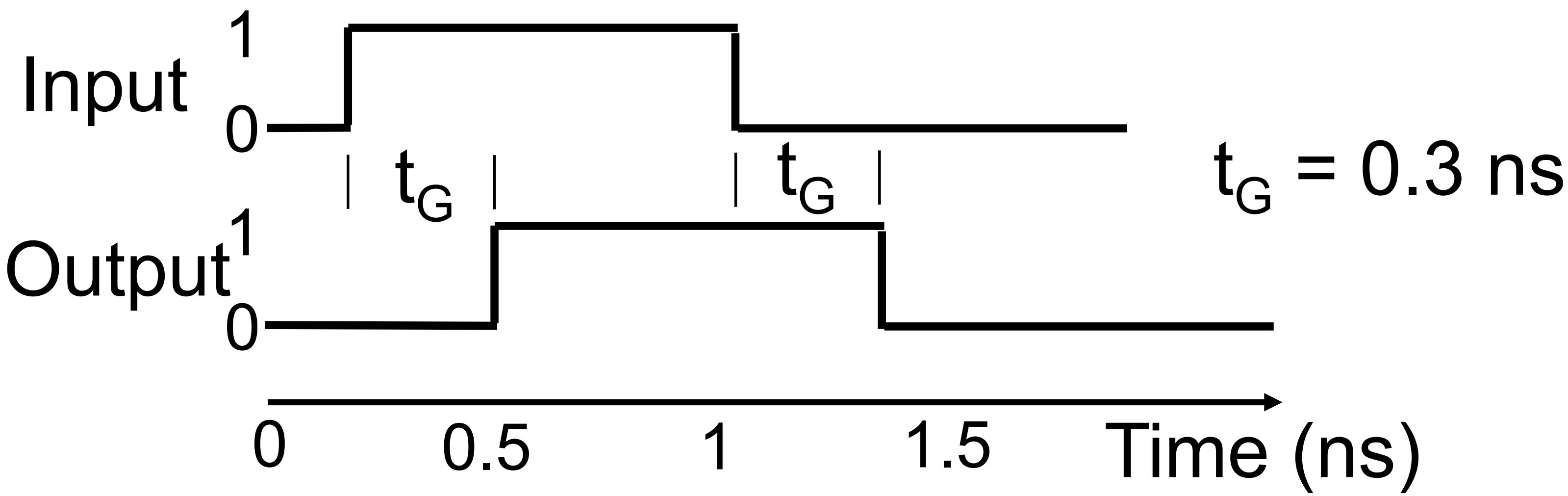
(a) Graphic symbols



(b) Timing diagram

Gate Delay

- In actual physical gates, if one or more input changes causes the output to change, the output change does not occur instantaneously.
- The delay between an input change(s) and the resulting output change is the *gate delay* denoted by t_G:



Logic Diagrams and Expressions

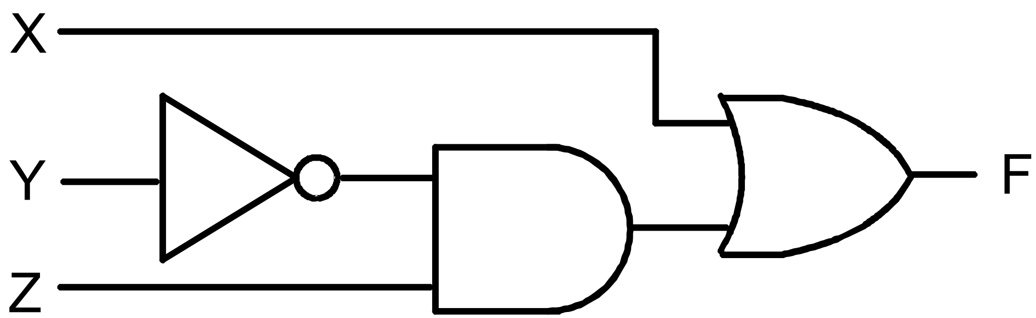
Truth Table

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

Equation

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

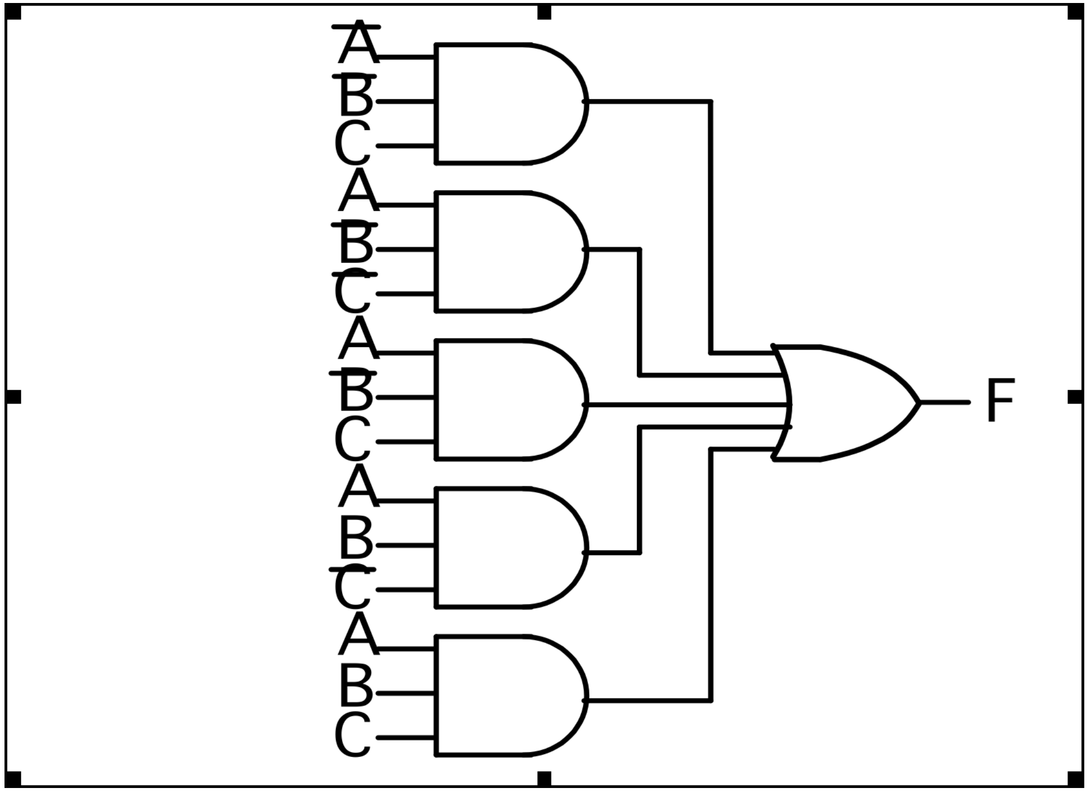
Logic Diagram

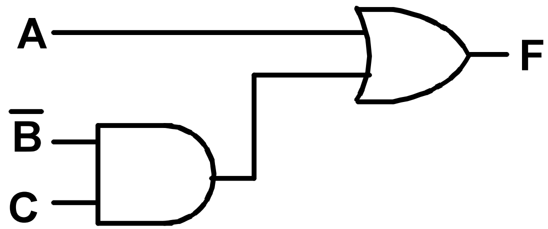


- Boolean equations, truth tables and logic diagrams describe the same function!
- Truth tables are unique; expressions and logic diagrams are not. This gives flexibility in implementing functions.

AND/OR Two-level Implementation of SOP Expression

- The two implementations for F are shown below – it is quite apparent which is simpler!





SOP and POS Observations

- The previous examples show that:
  - Canonical Forms (Sum-of-minterms, Product-of-Maxterms), or other standard forms (SOP, POS) differ in complexity
  - Boolean algebra can be used to manipulate equations into simpler forms.
  - Simpler equations lead to simpler two-level implementations
- Questions:
  - How can we attain a “simplest” expression?
  - Is there only one minimum cost circuit?
  - The next part will deal with these issues.