CMPE 212
Principles of Digital Design

Lecture 7

Switching Functions

February 15, 2016

www.csee.umbc.edu/~younis/CMPE212/CMPE212.htm

Lecture's Overview

- ***Previous Lecture:***
  - ➔ Logic Gates
    (famous gates, gate symbols)
  - ➔ Circuit implementation of logic gates
    (TTL and CMOS transistors, logic equivalent voltage level)

- ***This Lecture:***
  - ➔ Truth table and derivation of logic functions
  - ➔ Minterms and Maxterms
  - ➔ Sum of products and product of sums
  - ➔ Canonical form of switching functions

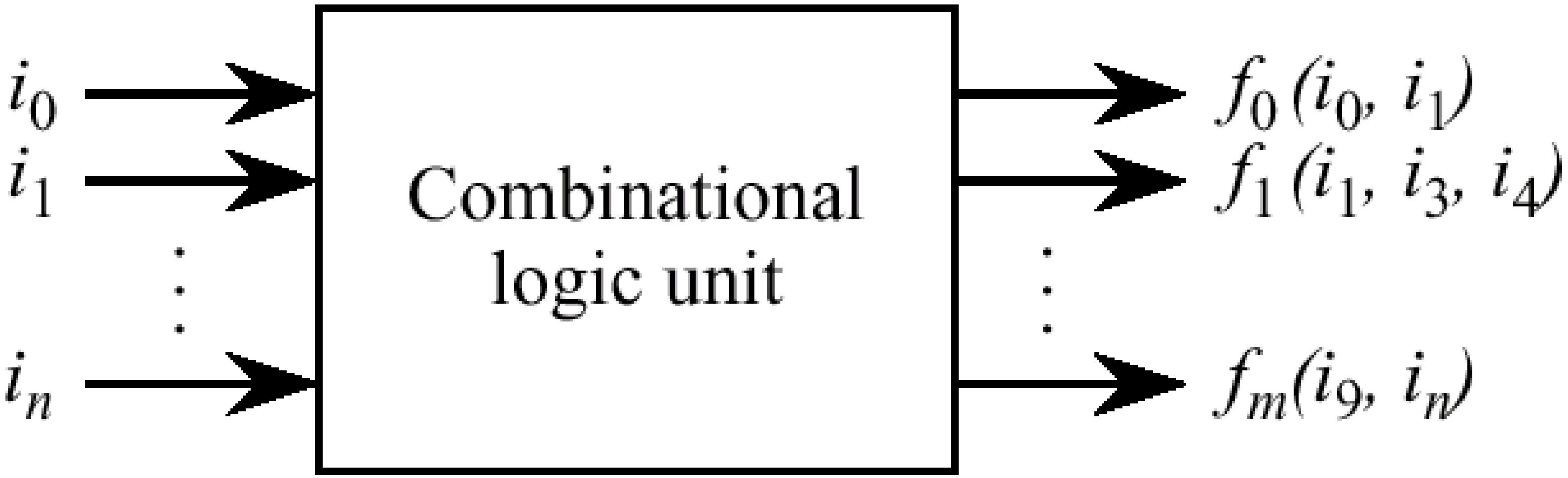
Some Definitions

- *Combinational logic*: a digital logic circuit in which logical decisions are made based only on combinations of the inputs, e.g. an adder.
- *Sequential logic*: a circuit in which decisions are made based on combinations of the current inputs as well as the past history of inputs. e.g. a memory unit.
- *Finite state machine*: a circuit which has an internal state, and whose outputs are functions of both current inputs and its internal state. e.g. a vending machine controller.

* Slide is courtesy of M. Murdocca and V. Heuring

The Combinational Logic Unit

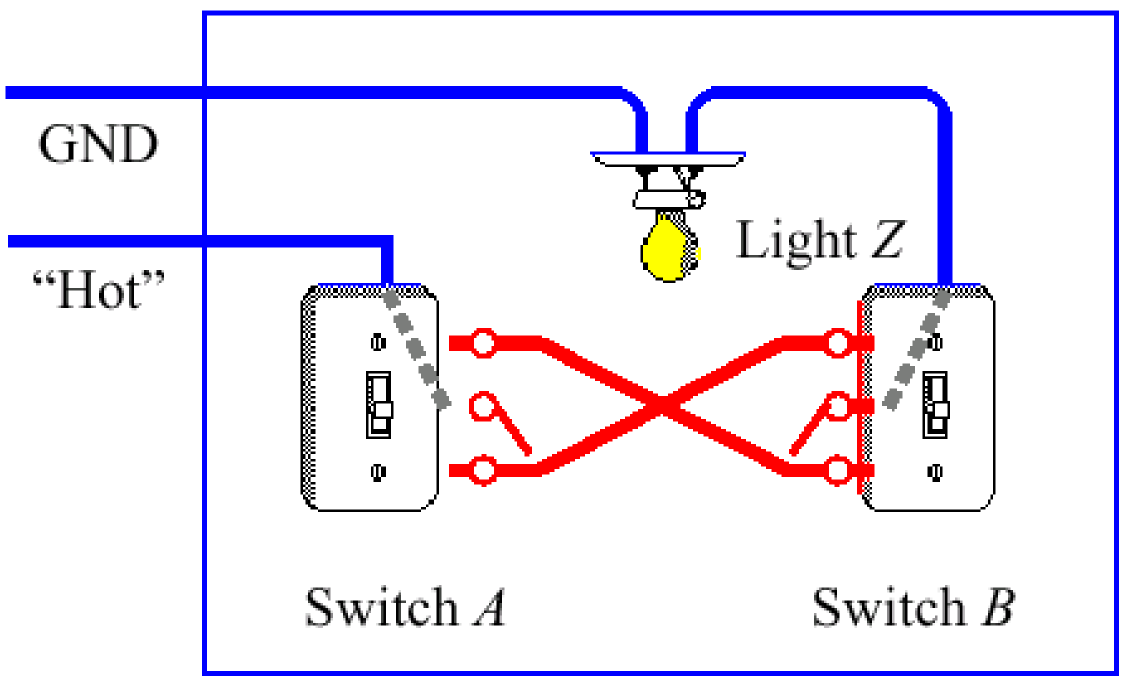
- Translates a set of inputs into a set of outputs according to one or more mapping functions.
- Inputs and outputs for a combination logic unit normally have two distinct (binary) values: high and low, 1 and 0, or 5 volt and 0 volt.
- The outputs of a CLU are strictly functions of the inputs, and the outputs are updated immediately after the inputs change. A set of inputs $i_0 – i_n$ are presented to the CLU, which produces a set of outputs according to mapping functions $f_0 – f_m$



* Slide is courtesy of M. Murdocca and V. Heuring

Truth Tables

- Developed in 1854 by George Boole.
- Further developed by Claude Shannon (Bell Labs).
- Outputs are computed for all possible input combinations (how many input combinations are there?)
- Consider a room with two light switches. How must they work?



Inputs		Output
A	*B*	*Z*
0	0	0
0	1	1
1	0	1
1	1	0

* Slide is courtesy of M. Murdocca and V. Heuring

Truth Table

- Truth table is an exhaustive description of a switching function. Contains 2^n input combinations for n variables.
- Example: $f(A,B,C) = A\,B + \bar{A}\,C + A\,\bar{C}$

2^n rows

n Input variables			Output
A	B	C	f(A,B,C)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Alternate Assignment of Outputs to Switch Settings

We can make the assignment of output values to input combinations any way that we want to achieve the desired input-output behavior.

Inputs		Output
A	B	Z
0	0	1
0	1	0
1	0	0
1	1	1

How Many Switching Functions?

- Output column of truth table has length 2^n for *n* input variables.
- It can be arranged in 2^{2^n} ways
- Example: n = 1,
  e.g., single variable.

		ut functions		
		(A)	F3(A)	F4(A)
		)	1	1
1	0	1	0	1

Slide is courtesy of M. Murdocca, V. Heuring and Vishwani D. Agrawal

Truth Tables Showing All Possible Functions of Two Binary Variables

The more frequently used functions have names: AND, XOR, OR, NOR, NXOR, and NAND. (Always use upper case spelling.)

Inputs		Outputs							
A	B	*False*	*AND*	$A\overline{B}$	A	$\overline{A}B$	B	*XOR*	*OR*
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1

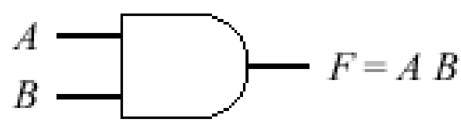
Inputs		Outputs							
A	B	*NOR*	*XNOR*	$\overline{B}$	$A+\overline{B}$	$\overline{A}$	$\overline{A}+B$	*NAND*	*True*
0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1

* Slide is courtesy of M. Murdocca and V. Heuring

Logic Gates and Their Symbols

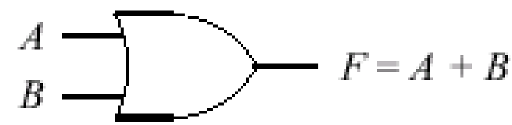
- Logic symbols for AND, OR, buffer, and NOT Boolean functions
- Note the use of the "inversion bubble."
- Be careful about the "nose" of the gate when drawing AND vs. OR.

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



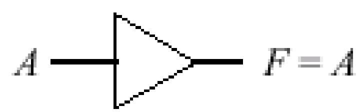
AND

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



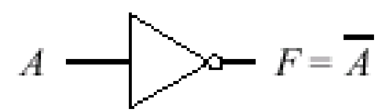
OR

A	F
0	0
1	1



Buffer

A	F
0	1
1	0



NOT (Inverter)

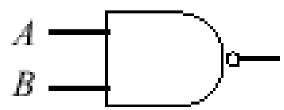
* Slide is courtesy of M. Murdocca and V. Heuring

Logic Gates and Their Symbols (Cont.)

NAND ≡ NOT AND

NOR ≡ NOT OR

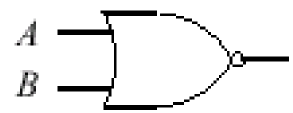
A	B	F
0	0	1
0	1	1
1	0	1
1	1	0



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

NAND

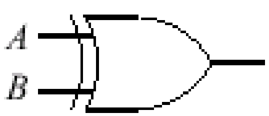
A	B	F
0	0	1
0	1	0
1	0	0
1	1	0



$F = \overline{A + B}$

NOR

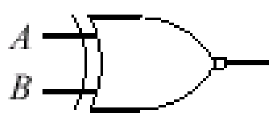
A	B	F
0	0	0
0	1	1
1	0	1
1	1	0



$F = A \oplus B$

Exclusive-OR (XOR)

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



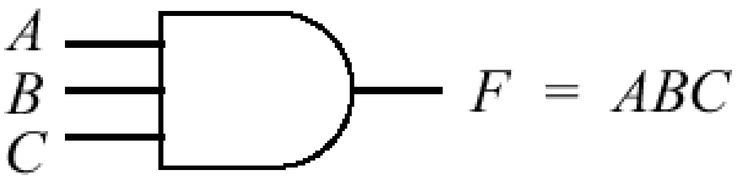
$F = A \odot B$

Exclusive-NOR (XNOR)

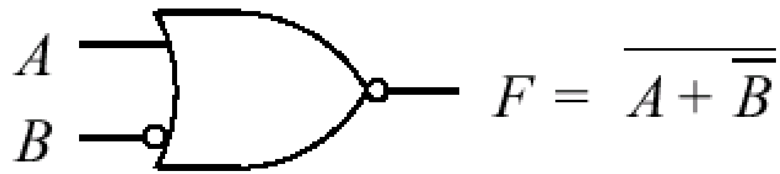
Sometimes called equivalence function

* Slide is courtesy of M. Murdocca and V. Heuring

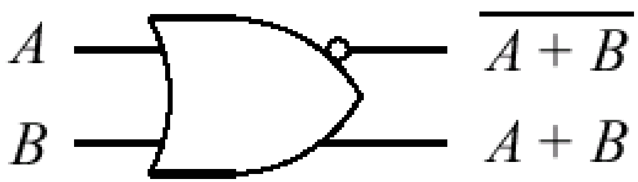
Variations of Logic Gate Symbols



(a)



(b)



(c)

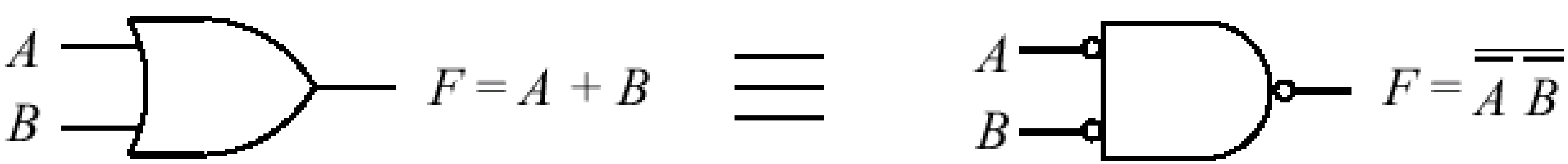
(a) 3 inputs **(b) A Negated input** **(c) Complementary outputs**

* Slide is courtesy of M. Murdocca and V. Heuring

DeMorgan's Theorem

$A \quad B$	$\overline{A\,B} = \overline{A} + \overline{B}$		$\overline{A + B} = \overline{A}\,\overline{B}$	
0 0	1	1	1	1
0 1	1	1	0	0
1 0	1	1	0	0
1 1	0	0	0	0

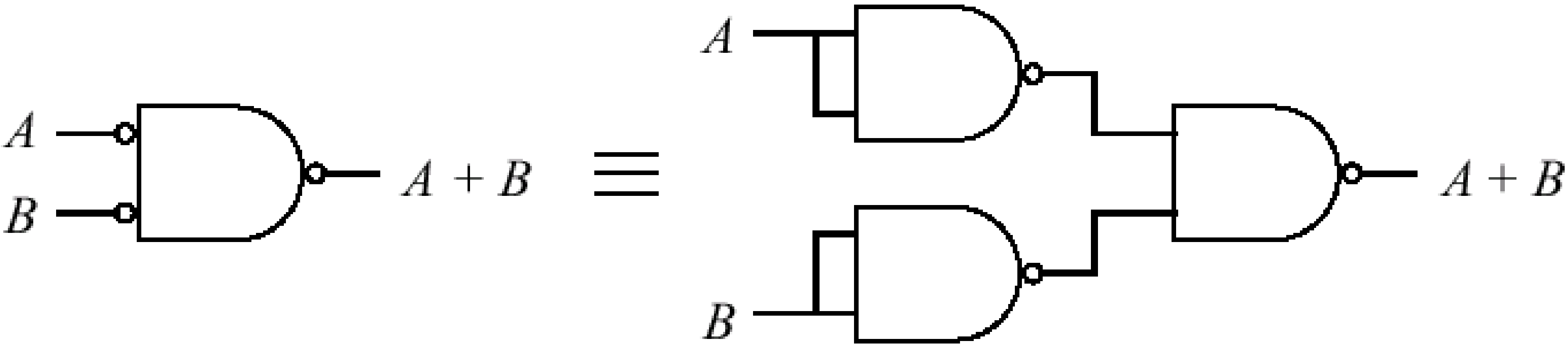
DeMorgan's theorem: $A + B = \overline{\overline{A + B}} = \overline{\overline{A}\,\overline{B}}$



* Slide is courtesy of M. Murdocca and V. Heuring

All-NAND Implementation of OR

NAND alone implements all other Boolean logic gates.

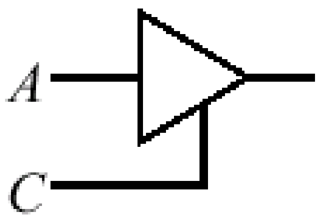


* Slide is courtesy of M. Murdocca and V. Heuring

Tri-State Buffers

- **Outputs can be 0, 1, or "electrically disconnected."**

C	A	F
0	0	$\varnothing$
0	1	$\varnothing$
1	0	0
1	1	1



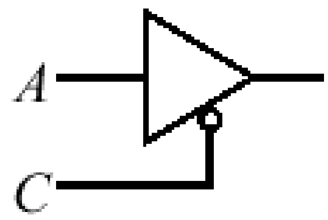
$F = A\,C$

or

$F = \varnothing$

Tri-state buffer

C	A	F
0	0	0
0	1	1
1	0	$\varnothing$
1	1	$\varnothing$



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

or

$F = \varnothing$

Tri-state buffer, inverted control

* Slide is courtesy of M. Murdocca and V. Heuring

Algebraic Form of Switching Functions

- Boolean variable: A variable denoted by a symbol; can assume a value 0 or 1.
- Literal: Symbol for a variable or its complement.
- Product or product term: A set of literals, ANDed together, Example, $a\ b\ \bar{c}$.
- Sum: A set of literals, Ored together; Example, $a + b + \bar{c}$.
- SOP (sum of products): A Boolean function expressed as a sum of products.

  Example: $f(A,B,C) = A\,B + \bar{A}\,C + A\,\bar{C}$
- POS (product of sums): A Boolean function expressed as a product of sums.

  Example: $f(A,B,C) = (\bar{A} + \bar{B} + \bar{C})(\bar{A} + B + \bar{C})(A + \bar{B} + C)$

Slide is courtesy of Vishwani D. Agrawal

Minterm

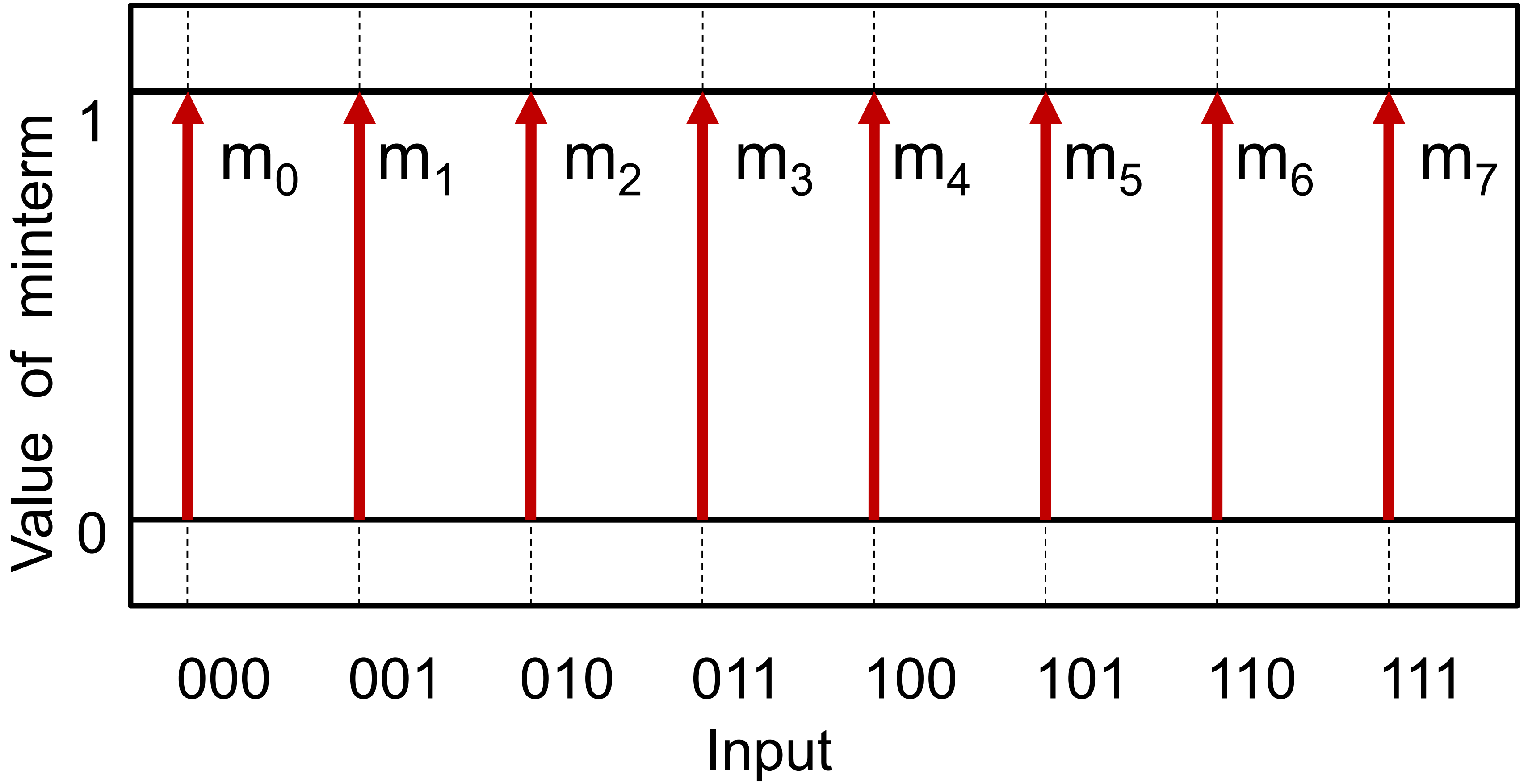
- A product term in which each variable is present either in true or in complement form
- For *n* variables, there are 2^n unique minterms.

	Minterm	Product
000	m_0	$\bar{A}\ \bar{B}\ \bar{C}$
001	m_1	$\bar{A}\ \bar{B}\ C$
010	m_2	$\bar{A}\ B\ \bar{C}$
011	m_3	$\bar{A}\ B\ C$
100	m_4	$A\ \bar{B}\ \bar{C}$
101	m_5	$A\ \bar{B}\ C$
110	m_6	$A\ B\ \bar{C}$
111	m_7	$A\ B\ C$

Slide is courtesy of Vishwani D. Agrawal

Minterms are Canonical Functions

A switching function that is represented as a sum of ONLY minterms is called canonical SOP function



Slide is courtesy of Vishwani D. Agrawal

Canonical SOP Form

- A Boolean function expressed as a sum of minterms.
- Example: $f(A,B,C) = A\,B + \bar{A}\,C + A\,\bar{C}$

$$= \bar{A}\,\bar{B}C + \bar{A}BC + A\,\bar{B}\,\bar{C} + AB\,\bar{C} + ABC$$

$$= m_1+m_3+m_4+m_6+m_7 = \sum m(1, 3, 4, 6, 7)$$

Truth table with row numbers

Row No.	A	B	C	f(A,B,C)
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Slide is courtesy of Vishwani D. Agrawal

Maxterm

- A summation term in which each variable is present either in true or in complement form.
- For *n* variables, there are 2^n unique maxterms.

	Maxterm	Sum
000	M_0	$A + B + C$
001	M_1	$A + B + \overline{C}$
010	M_2	$A + \overline{B} + C$
011	M_3	$A + \overline{B} + \overline{C}$
100	M_4	$\overline{A} + B + C$
101	M_5	$\overline{A} + B + \overline{C}$
110	M_6	$\overline{A} + \overline{B} + C$
111	M_7	$\overline{A} + \overline{B} + \overline{C}$

Slide is courtesy of Vishwani D. Agrawal

Canonical POS Form

- A Boolean function expressed as a product of maxterms.
- Example: $f(A,B,C) = A\,B + \bar{A}\,C + A\,\bar{C}$

$$= (A + B + C)(A + \bar{B} + C)(\bar{A} + B + \bar{C})$$

$$= M_0\, M_2\, M_5 = \Pi\, M(0, 2, 5)$$

Truth table with row numbers

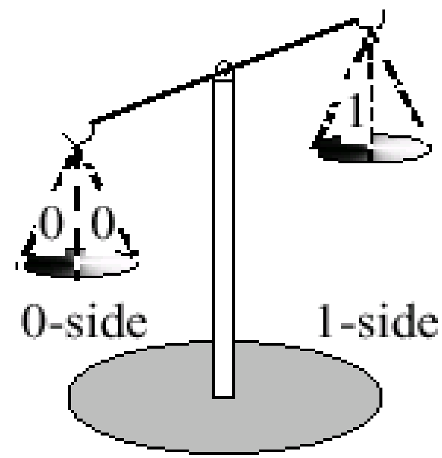
Row No.	A	B	C	f(A,B,C)
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Sum-of-Products Form

- The SOP form for the 3-input majority function is:

  $M = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC = m3 + m5 + m6 + m7 = \Sigma\ (3, 5, 6, 7).$
- Each of the 2^n terms are called *minterms*, ranging from 0 to 2^n - 1.
- Note relationship between minterm number and boolean value.

Minterm Index	*A*	*B*	*C*	*F*
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

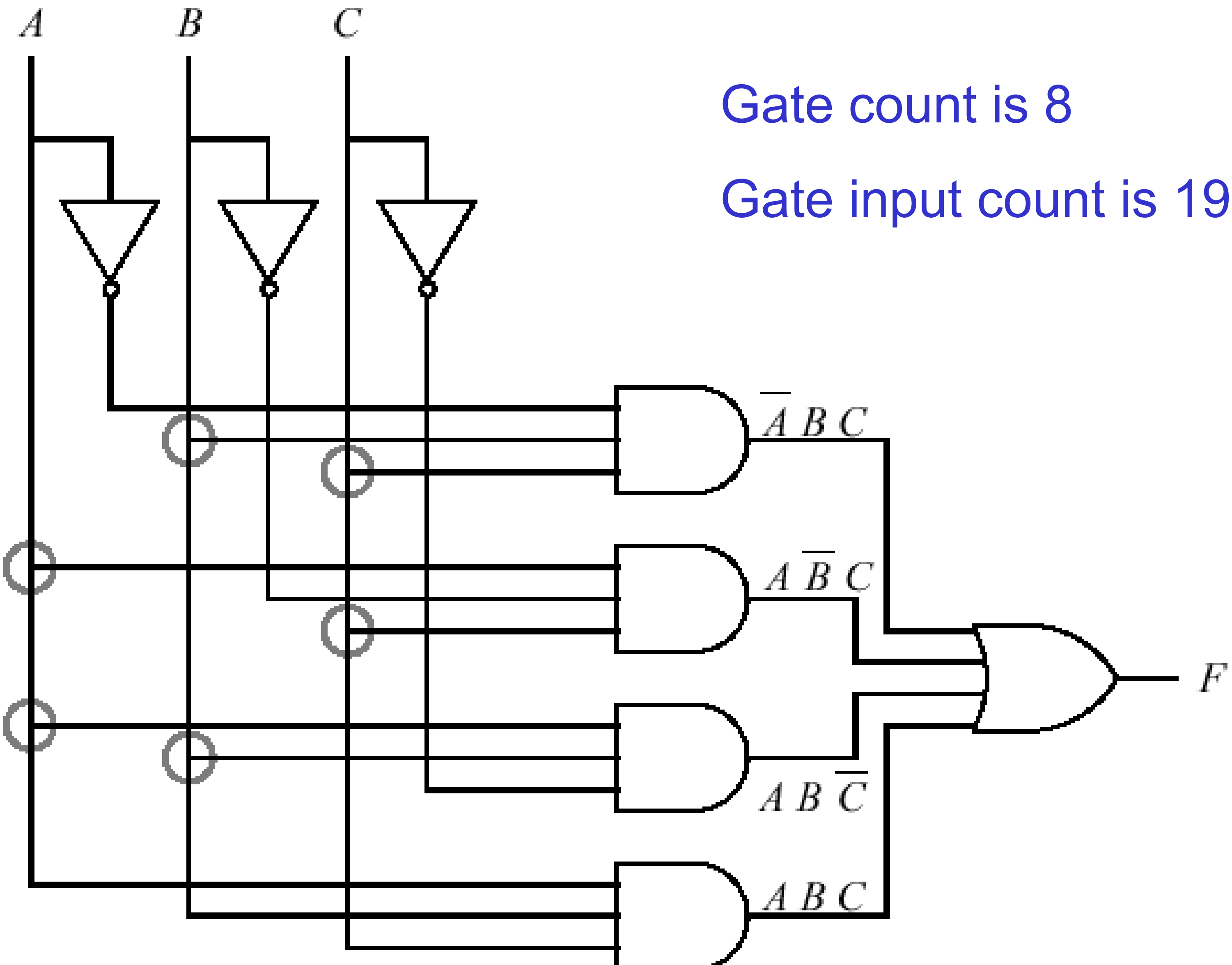


A balance tips to the left or right depending on whether there are more 0's or 1's.

For functions with large number of minterms, simplification becomes tedious. The complement function can be used instead

* Slide is courtesy of M. Murdocca and V. Heuring

AND-OR Implementation of Majority

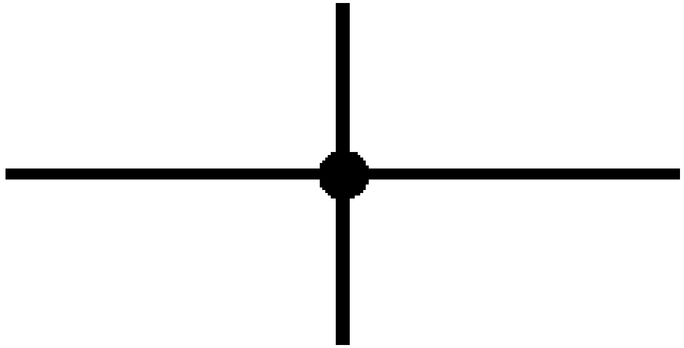


Gate count is 8

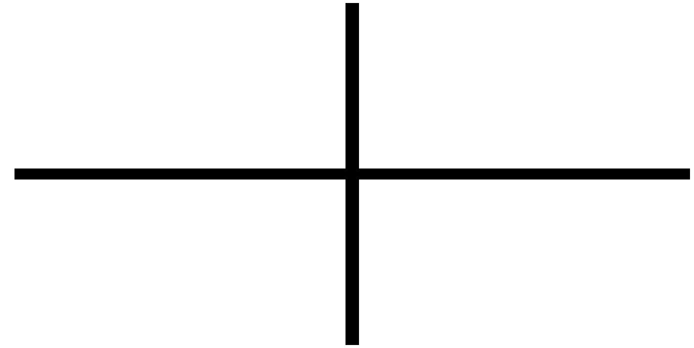
Gate input count is 19

* Slide is courtesy of M. Murdocca and V. Heuring

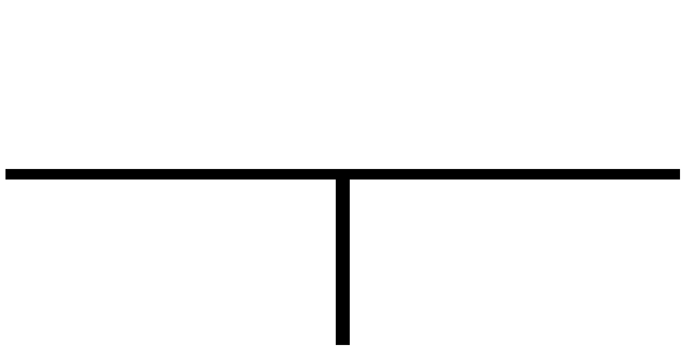
Notation Used at Circuit Intersections



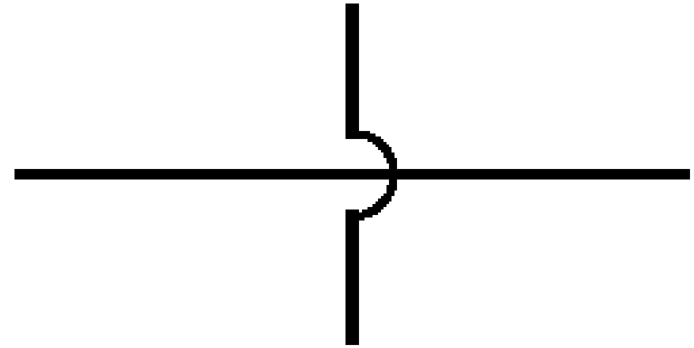
Connection



No connection



Connection

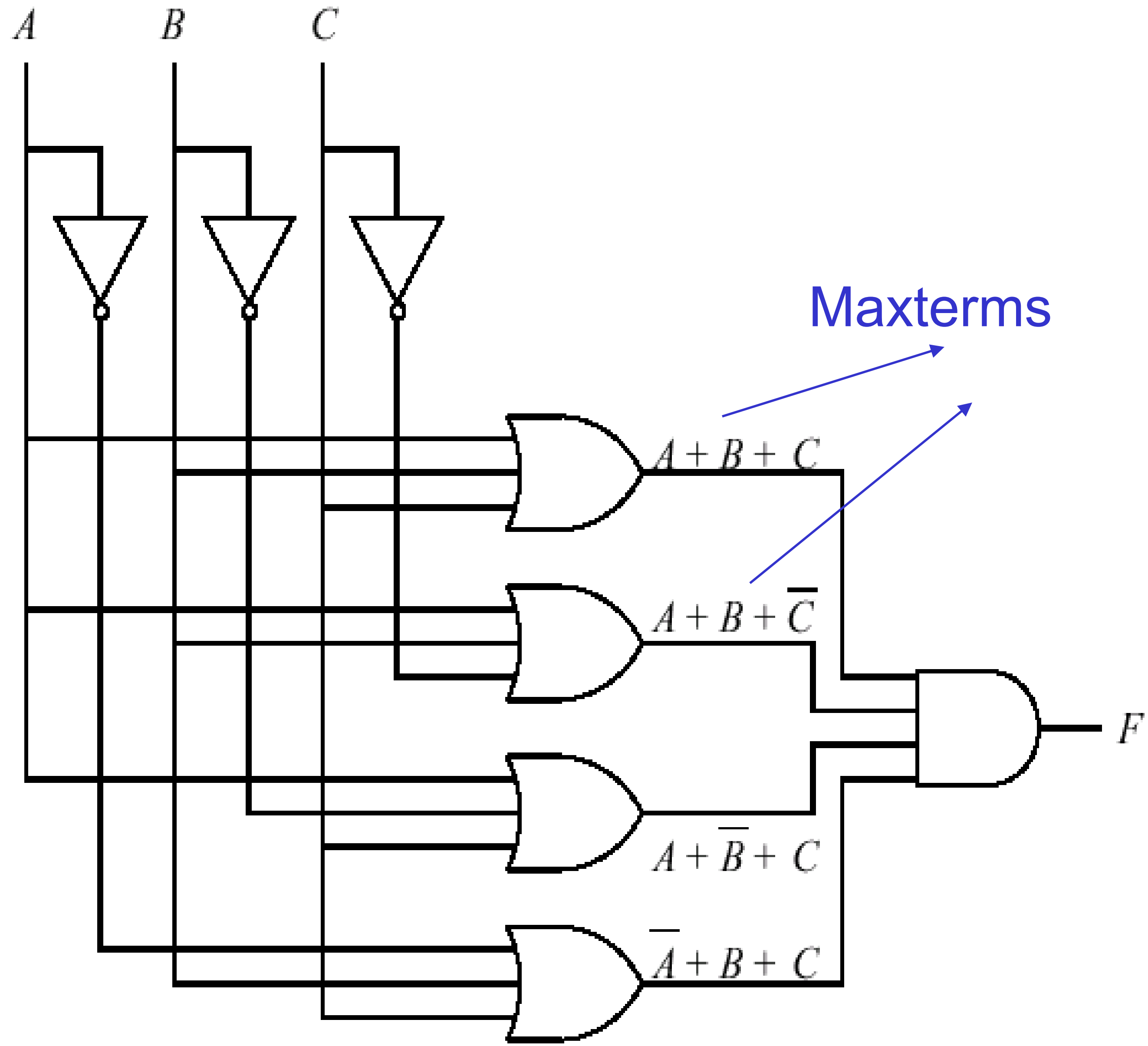


No connection

* Slide is courtesy of M. Murdocca and V. Heuring

OR-AND Implementation of Majority

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



* Slide is courtesy of M. Murdocca and V. Heuring

Converting to Canonical Form

Example: $f(x) = AB + A\bar{C} + \bar{A}C$

$$
\begin{aligned}
&= AB(C+\bar{C}) + A\bar{C}(B+\bar{B}) + \bar{A}C(B+\bar{B})\\
&= ABC + AB\bar{C} + A\bar{C}B + A\bar{C}\bar{B} + \bar{A}CB + \bar{A}C\bar{B}\\
&= ABC + AB\bar{C} + AB\bar{C} + A\bar{B}\bar{C} + \bar{A}BC + \bar{A}\bar{B}C\\
&= m_7 + m_6 + m_6 + m_4 + m_3 + m_1\\
&= \sum m(1,3,4,6,7)
\end{aligned}
$$

Example: $f(A,B,C) = A(A+\bar{C})$

$$
\begin{aligned}
A &= (A+\bar{B})(A+B)\\
&= (A+\bar{B}+\bar{C})(A+\bar{B}+C)(A+B+\bar{C})\,(A+B+C)\\
&= M_3M_2M_1M_0
\end{aligned}
$$

$$
\begin{aligned}
(A+\bar{C}) &= (A+\bar{C}+\bar{B})\,(A+\bar{C}+B)\\
&= (A+\bar{B}+\bar{C})\,(A+B+\bar{C}) = M_3M_1
\end{aligned}
$$

$$f(A,B,C) = (M_3M_2M_1M_0)(M_3M_1) = \prod M(0,1,2,3)$$

Conclusion

- ***Summary***
  - Introduction to combinational circuits
    (Truth table and Derivation of logic function)
  - Minterms and Maxterms
  - Sum of products and product of sums
  - Canonical form of switching functions
    (conversion from simplified to canonical form)
- ***Next Lecture***
  - Analyzing switching circuits using algebraic methods
  - Analysis of timing diagram
  - Synthesis o combinational logic circuits

Reading assignment: Section 2.2 in the textbook