

Introduction to Digital Design

Prof. Dr. Sıddıka Berna Örs Yalçın

Istanbul Technical University, Department of Electronics and Communication
Engineering

October 13, 2023

Contact Info

Room Number: 2318

email: siddika.ors@itu.edu.tr

web page: <http://web.itu.edu.tr/~orssi/>

Goals and objectives

- ▶ The goal of this course is to
 - ▶ provide a good understanding of the digital systems
 - ▶ introduce the basic building blocks of digital design including combinational logic circuits, combinational logic design, arithmetic functions and circuits and sequential circuits
 - ▶ Showing how these building blocks are employed in larger scale digital systems
- ▶ Having successfully completed this course, the student will:
 - ▶ acknowledge the importance of digital systems.
 - ▶ Design a digital circuit given a Boolean function.
 - ▶ Get familiar with typical combinatorial (adders, decoders, multiplexers, encoders) and sequential (D flip-flops, counters, registers, shift registers) components.
 - ▶ Understand how larger systems are organized.

Content

1. Digital Systems, Analog/Digital conversion, Number Systems, Number-Base Conversions, Binary Numbers
2. Axiomatic Definition, Basic Theorems and Properties of Boolean Algebra
3. Boolean Functions: Canonical Forms. Digital Logic Gates
4. Minimization: Quinn McCluskey Method
5. Karnaugh Diagrams, Don't-Care Conditions, Universal Gates, NAND and NOR Implementation
6. Analysis and Design of Combinational Circuits
7. Middle Scale Integrated (MSI) Elements (Decoders, Encoders, Multiplexers, Demultiplexers). Realization of Boolean Function using MSI elements.
8. Storage Elements: latches, Flip-Flops, HDL Model of SR Latch, SR latch with control input, D Latch, D Flip-flop
9. Analysis of Synchronous Sequential Circuits, Meally and Moore Finite State Machines
10. Design of Synchronous Sequential Circuits

Grading

1st Homework	2nd week	2.5 %
2nd Homework	4th week	2.5 %
1st Midterm	7th week	25 %
3rd Homework	8th week	2.5 %
4th Homework	10th week	2.5 %
2nd Midterm	12th week	25 %
Final		40 %

VF requirement: average of midterms $> 0,4 \times$ average of midterms of all the students in the class

References

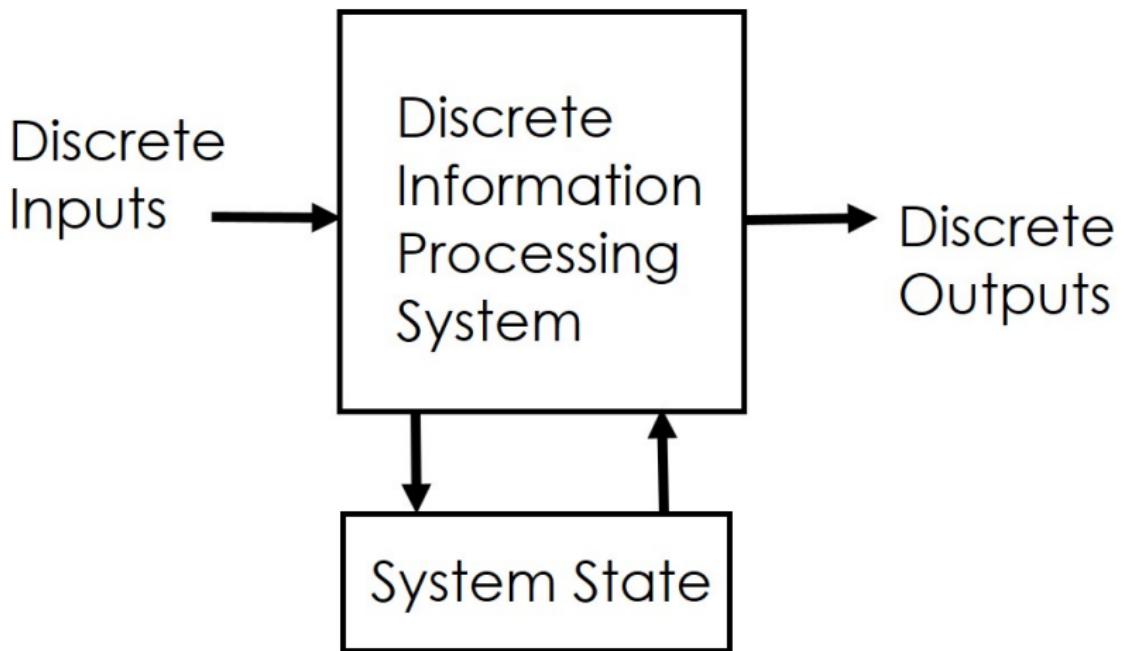
- ▶ Text Books:
 - ▶ Digital Design, M. Morris Mano, Michael D. Ciletti
 - ▶ Logic and Computer Design Fundamentals, 4/E, M. Morris Mano and Charles Kime , Prentice Hall, 2008.
- ▶ Slides and all announcements : Ninova

What is a Digital System?

- ▶ One characteristic:
 - ▶ Ability of manipulating **discrete elements of information**
- ▶ A **set** that has a finite number of elements contains discrete information
- ▶ Examples for discrete sets
 - ▶ Decimal digits $\{0, 1, \dots, 9\}$
 - ▶ Alphabet $\{A, B, \dots, Y, Z\}$
 - ▶ Binary digits $\{0, 1\}$
- ▶ One important problem:
 - ▶ how to represent the elements of discrete sets in physical systems?

Digital System

Takes a set of discrete information **inputs** and discrete internal information (**system state**) and generates a set of discrete information **outputs**.

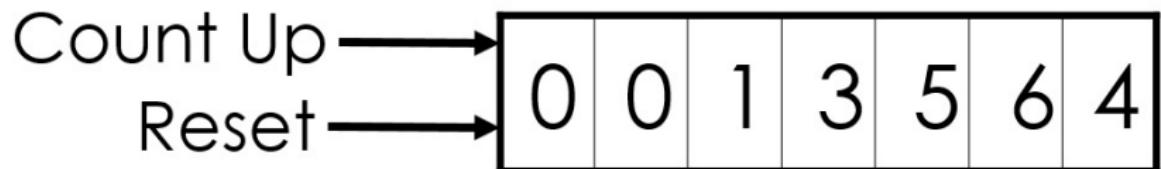


Types of Digital Systems

- ▶ No state present
 - ▶ Combinational Logic System
 - ▶ $\text{Output} = \text{Function}(\text{Input})$
- ▶ State present
 - ▶ State updated at discrete times \Rightarrow Synchronous Sequential System
 - ▶ State updated at any time \Rightarrow Asynchronous Sequential System
 - ▶ $\text{State} = \text{Function}(\text{State}, \text{Input})$
 - ▶ $\text{Output} = \text{Function}(\text{State}) \text{ or } \text{Function}(\text{State}, \text{Input})$

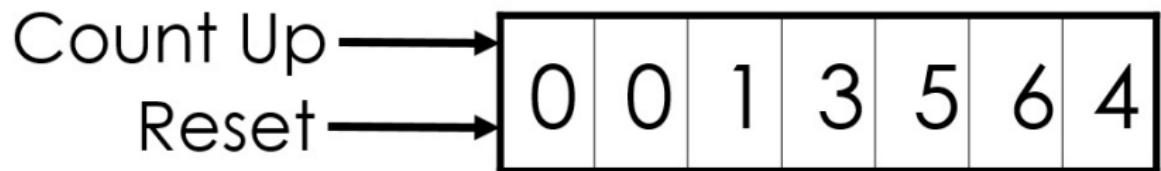
Digital System Example:

A Digital Counter (e. g., odometer):



Digital System Example:

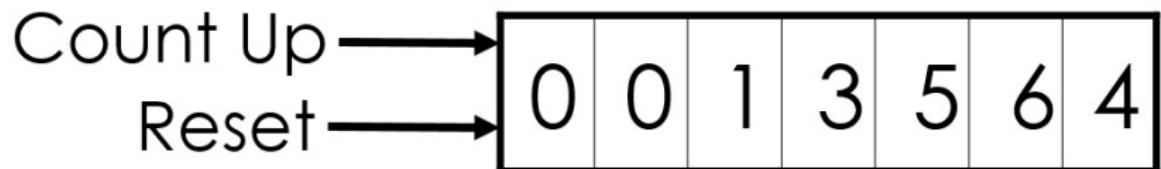
A Digital Counter (e. g., odometer):



- ▶ Input: Count Up, Reset

Digital System Example:

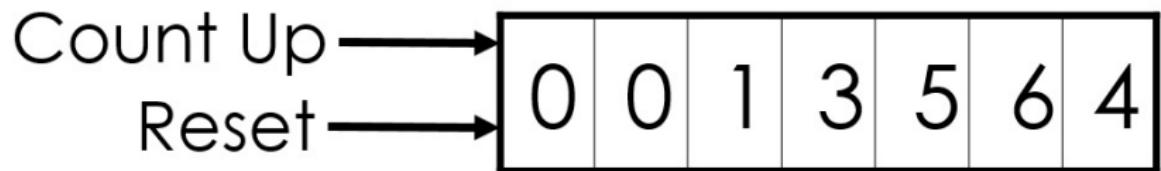
A Digital Counter (e. g., odometer):



- ▶ Input: Count Up, Reset
- ▶ Output: Visual Display

Digital System Example:

A Digital Counter (e. g., odometer):

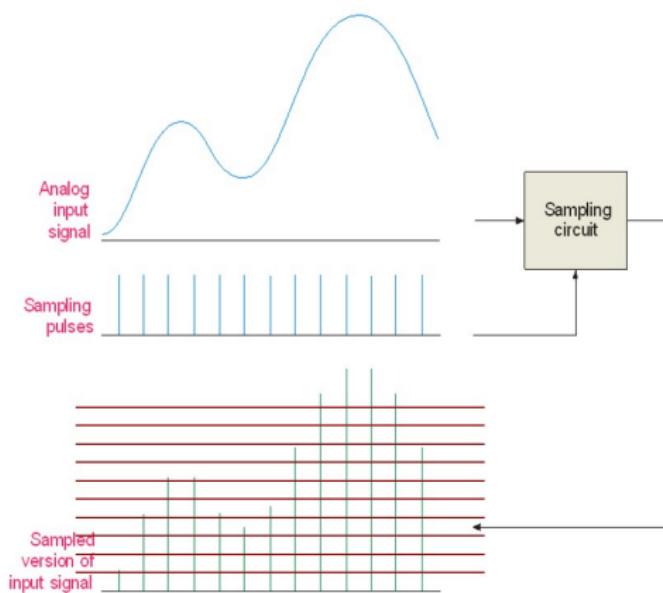


- ▶ Input: Count Up, Reset
- ▶ Output: Visual Display
- ▶ State: “Value” of stored digits

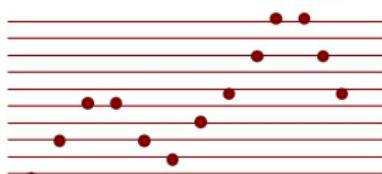
Analogue – Digital Signals

- ▶ The physical quantities in real world like current, voltage, temperature values change in a continuous range.
- ▶ The signals that can take any value between the boundaries are called **analogue** signals.
- ▶ Information take discrete values in digital systems.
- ▶ Binary **digital** signals can take one of the two possible values: 0-1, high-low, open-closed.

Conversion of Analogue Signals to Digital Signals



▪ Quantized Signal



How to Represent?

- ▶ In electronics circuits, we have electrical signals
 - ▶ voltage
 - ▶ current
- ▶ Different strengths of a physical signal can be used to represent elements of the discrete set.
- ▶ Which discrete set?
- ▶ Binary set is the easiest
 - ▶ two elements $\{0, 1\}$
 - ▶ Just two signal levels: 0 V and 4 V
- ▶ This is why we use binary system to represent the information in our digital system.

Binary System

- ▶ Binary set $\{0, 1\}$
 - ▶ The elements of binary set, 0 and 1 are called **binary digits** or shortly **bits**.
- ▶ How to represent the elements of other discrete sets
 - ▶ Decimal digits $\{0, 1, \dots, 9\}$
 - ▶ Alphabet $\{A, B, \dots, Y, Z\}$
- ▶ Elements of any discrete sets can be represented using **groups of bits**.
 - ▶ $9 \rightarrow 1001$
 - ▶ $A \rightarrow 1000001$

Number Systems – Representation

- ▶ Positive radix, positional number systems
- ▶ A number with radix r is represented by a string of digits:
 $Number_r = A_{n-1}A_{n-2}\dots A_1A_0.A_{-1}A_{-2}\dots A_{-m+1}A_{-m}$
in which $0 \leq A_i < r$ and $.$ is the radix point.
- ▶ The string of digits represents the power series:

$$Number_{10} = \sum_{i=-m}^{n-1} A_i r^i$$

Representation of positive integers

► Example:

$$215_{10} = (11010111)_2 =$$

$$1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

- The largest positive number that can be represented by 8 bits is: $(11111111)_2 = 255_{10}$
- The smallest positive number that can be represented by 8 bits is: $(00000000)_2 = 0_{10}$

Representation of positive integers

- ▶ Example:

$$215_{10} = (11010111)_2 =$$

$$1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

Most Significant Bit

- ▶ The largest positive number that can be represented by 8 bits is: $(11111111)_2 = 255_{10}$
- ▶ The smallest positive number that can be represented by 8 bits is: $(00000000)_2 = 0_{10}$

Representation of positive integers

- ▶ Example:

$$215_{10} = (11010111)_2 =$$

$$1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

Most Significant Bit

Least Significant Bit

Representation of positive integers

► Example:

$$215_{10} = (11010111)_2 =$$

$$1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

Most Significant Bit

Least Significant Bit

- The largest positive number that can be represented by 8 bits is: $(11111111)_2 = 255_{10}$
- The smallest positive number that can be represented by 8 bits is: $(00000000)_2 = 0_{10}$

Some Bases

Name	Base	Digits
Decimal	10	0, 1, ..., 9
Binary	2	0, 1
Octal	8	0, 1, ..., 7
Hexadecimal	16	0, 1, ..., 9, A, B, C, D, E, F

Some Numbers in Different Bases

Decimal (Base 10)	Binary (Base 2)	Octal (Base 8)	Hexadecimal (Base 16)
0	00000	00	00
1	00001	01	01
2	00010	02	02
3	00011	03	03
4	00100	04	04
5	00101	05	05
6	00110	06	06
7	00111	07	07
8	01000	10	08
9	01001	11	09
10	01010	12	0A
11	01011	13	0B
12	01100	14	0C
13	01101	15	0D
14	01110	16	0E
15	01111	17	0F
16	10000	20	10

Base Conversions

- ▶ From base- r to decimal is easy
 - ▶ expand the number in power series and add all the terms
- ▶ Reverse operation is somewhat more difficult
 - ▶ divide the decimal number successively by r
 - ▶ accumulate the remainders.
- ▶ If there is a fraction, then integer part and fraction part are handled separately.

Conversion from base r to base decimal

- ▶ Convert 101110.10110_2 to base 10

$$\begin{aligned}101110.10110_2 &= 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} \\&\quad + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4} \\&= 32 + 8 + 4 + 2 + \frac{1}{2} + \frac{1}{8} + \frac{1}{16} \\&= 46.6875\end{aligned}$$

Example: 46.6875_{10} to Binary (base 2)

Convert 46 to binary

$$\begin{array}{r} 46 \\ \hline 46 & 2 \\ \hline 0 & 23 & 2 \\ & \hline 1 & 22 & 11 & 2 \\ & & 1 & 10 & 5 & 2 \\ & & & 1 & 4 & 2 & 2 \\ & & & & 1 & 2 & 1 \\ & & & & & 0 \\ & & & & & & \leftarrow \end{array}$$

$$46 = 101110_2$$

Example: 46.6875_{10} to Binary (base 2)

- ▶ Convert 0.6875 to binary
 - ▶ $0.6875 \times 2 = 1.375$

Example: 46.6875_{10} to Binary (base 2)

- ▶ Convert 0.6875 to binary
 - ▶ $0.6875 \times 2 = 1.375$
 - ▶ $0.375 \times 2 = 0.75$

Example: 46.6875_{10} to Binary (base 2)

- ▶ Convert 0.6875 to binary
 - ▶ $0.6875 \times 2 = 1.375$
 - ▶ $0.375 \times 2 = 0.75$
 - ▶ $0.75 \times 2 = 1.5$

Example: 46.6875_{10} to Binary (base 2)

► Convert 0.6875 to binary

- $0.6875 \times 2 = 1.375$
- $0.375 \times 2 = 0.75$
- $0.75 \times 2 = 1.5$
- $0.5 \times 2 = 1$

Example: 46.6875_{10} to Binary (base 2)

► Convert 0.6875 to binary

- $0.6875 \times 2 = 1.375$
- $0.375 \times 2 = 0.75$
- $0.75 \times 2 = 1.5$
- $0.5 \times 2 = 1$
- $0 \times 2 = 0$

Example: 46.6875_{10} to Binary (base 2)

- ▶ Convert 0.6875 to binary
 - ▶ $0.6875 \times 2 = 1.375$
 - ▶ $0.375 \times 2 = 0.75$
 - ▶ $0.75 \times 2 = 1.5$
 - ▶ $0.5 \times 2 = 1$
 - ▶ $0 \times 2 = 0$
- ▶ Put two results together with a radix point

101110.10110_2

Example: 46.6875_{10} to Hexadecimal (base 16)

- ▶ Convert 46 to hexadecimal

- ▶ $46 = 2 \times 16 + 14$

Example: 46.6875_{10} to Hexadecimal (base 16)

- ▶ Convert 46 to hexadecimal

- ▶ $46 = 2 \times 16 + 14$

- ▶ $2 = 0 \times 16 + 2$

Example: 46.6875_{10} to Hexadecimal (base 16)

- ▶ Convert 46 to hexadecimal
 - ▶ $46 = 2 \times 16 + 14$
 - ▶ $2 = 0 \times 16 + 2$
- ▶ Convert 0.6875 to hexadecimal
 - ▶ $0.6875 \times 16 = 11.0$

Example: 46.6875_{10} to Hexadecimal (base 16)

- ▶ Convert 46 to hexadecimal
 - ▶ $46 = 2 \times 16 + 14$
 - ▶ $2 = 0 \times 16 + 2$
- ▶ Convert 0.6875 to hexadecimal
 - ▶ $0.6875 \times 16 = 11.0$
 - ▶ $0 \times 16 = 0$

Example: 46.6875_{10} to Hexadecimal (base 16)

- ▶ Convert 46 to hexadecimal
 - ▶ $46 = 2 \times 16 + 14$
 - ▶ $2 = 0 \times 16 + 2$
- ▶ Convert 0.6875 to hexadecimal
 - ▶ $0.6875 \times 16 = 11.0$
 - ▶ $0 \times 16 = 0$
- ▶ Put two results together with a radix point

$2E.B_{16}$

Conversions between Binary, Octal and Hexadecimal

- ▶ Octal to Binary

- ▶ $743.056_8 = 111\ 100\ 011.000\ 101\ 110_2$

Conversions between Binary, Octal and Hexadecimal

- ▶ Octal to Binary
 - ▶ $743.056_8 = 111\ 100\ 011.000\ 101\ 110_2$
- ▶ Hexadecimal to Binary
 - ▶ $A49.0C6_{16} = 1010\ 0100\ 1001.0000\ 1100\ 0110_2$

Conversions between Binary, Octal and Hexadecimal

- ▶ Octal to Binary
 - ▶ $743.056_8 = 111\ 100\ 011.000\ 101\ 110_2$
- ▶ Hexadecimal to Binary
 - ▶ $A49.0C6_{16} = 1010\ 0100\ 1001.0000\ 1100\ 0110_2$
- ▶ Binary to Octal
 - ▶ $1\ ||\ 011\ ||\ 100\ ||\ 011.000\ ||\ 101\ ||\ 110\ ||\ 1_2 = 1343.0564_8$

Conversions between Binary, Octal and Hexadecimal

- ▶ Octal to Binary

- ▶ $743.056_8 = 111\ 100\ 011.000\ 101\ 110_2$

- ▶ Hexadecimal to Binary

- ▶ $A49.0C6_{16} = 1010\ 0100\ 1001.0000\ 1100\ 0110_2$

- ▶ Binary to Octal

- ▶ $1\ ||\ 011\ ||\ 100\ ||\ 011.000\ ||\ 101\ ||\ 110\ ||\ 1_2 = 1343.0564_8$

- ▶ Binary to Hexadecimal

- ▶ $1\ ||\ 1010\ ||\ 0100\ ||\ 1001.0010\ ||\ 1100\ ||\ 0110\ ||\ 1_2 = 1A49.2C68_{16}$

Binary Logic and Gates

- ▶ **Binary variables** take one of the 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 X1 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 (\cdot).
- ▶ OR is denoted by a plus ($+$).
- ▶ NOT is denoted by an overbar (\bar{a}), a single quote mark (a') after, or ($\sim a$) before the variable.

Notation Examples

- ▶ Examples:
 - ▶ $Y = A \cdot B$ is read “Y is equal to A AND B.”
 - ▶ $z = x + y$ is read “z is equal to x OR y.”
 - ▶ $X = \bar{A}$ is read “X is equal to NOT A.”
- ▶ Note: The statement:
 - ▶ $1 + 1 = 2$ (read “one plus one equals two”) is not the same as
 - ▶ $1 + 1 = 1$ (read “1 or 1 equals 1”).

Operator Definitions

Operations are defined on the values “0” and “1” for each operator:

AND

$$0 \cdot 0 = 0$$

$$0 \cdot 1 = 0$$

$$1 \cdot 0 = 0$$

$$1 \cdot 1 = 1$$

Operator Definitions

Operations are defined on the values “0” and “1” for each operator:

AND

0	.	0	=	0	0	+	0	=	0
0	.	1	=	0	0	+	1	=	1
1	.	0	=	0	1	+	0	=	1
1	.	1	=	1	1	+	1	=	1

OR

Operator Definitions

Operations are defined on the values “0” and “1” for each operator:

AND

$$\begin{array}{l} 0 \\ \cdot \\ 0 \end{array} = 0$$

$$\begin{array}{l} 0 \\ \cdot \\ 1 \end{array} = 0$$

$$\begin{array}{l} 1 \\ \cdot \\ 0 \end{array} = 0$$

$$\begin{array}{l} 1 \\ \cdot \\ 1 \end{array} = 1$$

OR

$$\begin{array}{l} 0 \\ + \\ 0 \end{array} = 0$$

$$\begin{array}{l} 0 \\ + \\ 1 \end{array} = 1$$

$$\begin{array}{l} 1 \\ + \\ 0 \end{array} = 1$$

$$\begin{array}{l} 1 \\ + \\ 1 \end{array} = 1$$

NOT

$$\bar{0} = 1$$

$$\bar{1} = 0$$

Truth Tables

Truth table: - a tabular listing of the values of a function for all possible combinations of values on its arguments

AND

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

AND

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

OR

X	Y	Z
0	0	0
0	1	1
1	0	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

AND

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

OR

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

NOT

X	Z
0	1
1	0

Boolean Algebra

- ▶ Operators are defined over $B=\{0, 1\}$ set
- ▶ Double variable operators: AND (.), OR (+)
- ▶ Single variable operator: NOT (')

Axioms

Let $a, b, c \in B$

1. Closure:

$$a+b=c$$

$$a.b=c$$

Boolean Algebra

- ▶ Operators are defined over $B=\{0, 1\}$ set
- ▶ Double variable operators: AND (.), OR (+)
- ▶ Single variable operator: NOT (')

Axioms

Let $a, b, c \in B$

1. Closure: $a+b=c$ $a.b=c$
2. Commutative: $a+b=b+a$ $a.b=b.a$

Boolean Algebra

- ▶ Operators are defined over $B=\{0, 1\}$ set
- ▶ Double variable operators: AND (.), OR (+)
- ▶ Single variable operator: NOT (')

Axioms

Let $a, b, c \in B$

1. Closure: $a+b=c$ $a.b=c$
2. Commutative: $a+b=b+a$ $a.b=b.a$
3. Distributive: $a+(b.c)=(a+b).(a+c)$ $a.(b+c)=a.b+a.c$

Boolean Algebra

- ▶ Operators are defined over $B=\{0, 1\}$ set
- ▶ Double variable operators: AND (.), OR (+)
- ▶ Single variable operator: NOT (')

Axioms

Let $a, b, c \in B$

1. Closure: $a+b=c$ $a.b=c$
2. Commutative: $a+b=b+a$ $a.b=b.a$
3. Distributive: $a+(b.c)=(a+b).(a+c)$ $a.(b+c)=a.b+a.c$
4. Associative: $a+(b+c)=(a+b)+c$ $a.(b.c)=(a.b).c$

Boolean Algebra

- ▶ Operators are defined over $B=\{0, 1\}$ set
- ▶ Double variable operators: AND (.), OR (+)
- ▶ Single variable operator: NOT (')

Axioms

Let $a, b, c \in B$

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

Boolean Algebra

- ▶ Operators are defined over $B=\{0, 1\}$ set
- ▶ Double variable operators: AND (.), OR (+)
- ▶ Single variable operator: NOT (')

Axioms

Let $a, b, c \in B$

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

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: $F = A(B + C)\overline{(C + D)}$

Properties and Theorems

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

1. Identity element:

$$a+1=1$$

$$a.0=0$$

Properties and Theorems

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

1. Identity element: $a+1=1$ $a.0=0$
2. Transformation: $(a')'=a$

Properties and Theorems

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

1. Identity element: $a+1=1$ $a.0=0$
2. Transformation: $(a')'=a$
3. Constant power: $a+a+\dots+a=a$ $a.a.\dots.a=a$

Properties and Theorems

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

- | | | |
|----------------------|-----------------|-----------------|
| 1. Identity element: | $a+1=1$ | $a.0=0$ |
| 2. Transformation: | $(a')'=a$ | |
| 3. Constant power: | $a+a+\dots+a=a$ | $a.a.\dots.a=a$ |
| 4. Absorption: | $a+a.b=a$ | $a.(a+b)=a$ |

Properties and Theorems

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

- | | | |
|-------------------------|-----------------|-----------------|
| 1. Identity element: | $a+1=1$ | $a.0=0$ |
| 2. Transformation: | $(a')'=a$ | |
| 3. Constant power: | $a+a+\dots+a=a$ | $a.a.\dots.a=a$ |
| 4. Absorption: | $a+a.b=a$ | $a.(a+b)=a$ |
| 5. De Morgan's Theorem: | $(a+b)'=a'.b'$ | $(a.b)'=a'+b'$ |

Properties and Theorems

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

1. Identity element: $a+1=1$ $a.0=0$
2. Transformation: $(a')'=a$
3. Constant power: $a+a+\dots+a=a$ $a.a.\dots.a=a$
4. Absorption: $a+a.b=a$ $a.(a+b)=a$
5. De Morgan's Theorem: $(a+b)'=a'.b'$ $(a.b)'=a'+b'$
6. General De Morgan's Theorem:

$$f'(X_1, X_2, \dots, X_n, 0, 1, +, .) = f(X_1', X_2', \dots, X_n', 1, 0, ., +)$$

Example 1: Boolean Algebraic Proof

$$A + A \cdot B = A \quad \text{Absorption Theorem}$$

Example 1: Boolean Algebraic Proof

$A + A \cdot B = A$ Absorption Theorem

Proof Steps Axiom or Theorem used

Example 1: Boolean Algebraic Proof

$A + A \cdot B = A$ Absorption Theorem

Proof Steps **Axiom or Theorem used**

$A + A \cdot B$

Example 1: Boolean Algebraic Proof

$A + A \cdot B = A$ Absorption Theorem

Proof Steps **Axiom or Theorem used**

$$A + A \cdot B$$

$$= A \cdot 1 + A \cdot B$$

$$X \cdot 1 = X$$

Neutral Element

Example 1: Boolean Algebraic Proof

$$A + A \cdot B = A \quad \text{Absorption Theorem}$$

Proof Steps **Axiom or Theorem used**

$$A + A \cdot B$$

$$= A \cdot 1 + A \cdot B \quad X \cdot 1 = X \quad \text{Neutral Element}$$

$$= A \cdot (1 + B) \quad X \cdot Y + X \cdot Z = X \cdot (Y + Z) \quad \text{Distributive}$$

Example 1: Boolean Algebraic Proof

$$A + A \cdot B = A \quad \text{Absorption Theorem}$$

Proof Steps **Axiom or Theorem used**

$$A + A \cdot B$$

$$= A \cdot 1 + A \cdot B \quad X \cdot 1 = X \quad \text{Neutral Element}$$

$$= A \cdot (1 + B) \quad X \cdot Y + X \cdot Z = X \cdot (Y + Z) \quad \text{Distributive}$$

$$= A \cdot 1 \quad 1 + X = 1 \quad \text{Identity Element}$$

Example 1: Boolean Algebraic Proof

$$A + A \cdot B = A \quad \text{Absorption Theorem}$$

Proof Steps **Axiom or Theorem used**

$$A + A \cdot B$$

$$= A \cdot 1 + A \cdot B \quad X \cdot 1 = X \quad \text{Neutral Element}$$

$$= A \cdot (1 + B) \quad X \cdot Y + X \cdot Z = X \cdot (Y + Z) \quad \text{Distributive}$$

$$= A \cdot 1 \quad 1 + X = 1 \quad \text{Identity Element}$$

$$= A \quad X \cdot 1 = X \quad \text{Neutral Element}$$

Our primary reason for doing proofs is to learn:

- ▶ Careful and efficient use of the identities and theorems of Boolean algebra
- ▶ How to choose the appropriate identity or theorem to apply to make forward progress, irrespective of the application

Example 2: Boolean Algebraic Proofs

$$AB + A'C + BC = AB + A'C \quad \text{Consensus Theorem}$$

Example 2: Boolean Algebraic Proofs

$$AB + A'C + BC = AB + A'C \quad \text{Consensus Theorem}$$

Proof Steps Axiom or Theorem used

Example 2: Boolean Algebraic Proofs

$$AB + A'C + BC = AB + A'C \quad \text{Consensus Theorem}$$

Proof Steps

Axiom or Theorem used

$$AB + A'C + BC$$

Example 2: Boolean Algebraic Proofs

$$AB + A'C + BC = AB + A'C \quad \text{Consensus Theorem}$$

Proof Steps

$$AB + A'C + BC$$

$$= AB + A'C + 1 \cdot BC$$

Axiom or Theorem used

$$X \cdot 1 = X$$

Neutral Element

Example 2: Boolean Algebraic Proofs

$$AB + A'C + BC = AB + A'C \quad \text{Consensus Theorem}$$

Proof Steps

$$AB + A'C + BC$$

$$= AB + A'C + 1 \cdot BC$$

$$= AB + A'C + (A + A') \cdot BC$$

Axiom or Theorem used

$$X \cdot 1 = X$$

$$X + X' = 1$$

Neutral Element

Inverse

Example 2: Boolean Algebraic Proofs

$$AB + A'C + BC = AB + A'C \quad \text{Consensus Theorem}$$

Proof Steps

Axiom or Theorem used

$$AB + A'C + BC$$

$$= AB + A'C + 1 \cdot BC \quad X \cdot 1 = X$$

Neutral Element

$$= AB + A'C + (A + A') \cdot BC \quad X + X' = 1$$

Inverse

$$= AB + A'C + ABC + A'BC \quad X \cdot Y + X \cdot Z = X \cdot (Y + Z)$$

Distributive

Example 2: Boolean Algebraic Proofs

$$AB + A'C + BC = AB + A'C \quad \text{Consensus Theorem}$$

Proof Steps

Axiom or Theorem used

$$AB + A'C + BC$$

$$= AB + A'C + 1 \cdot BC$$

$$X \cdot 1 = X$$

Neutral Element

$$= AB + A'C + (A + A') \cdot BC$$

$$X + X' = 1$$

Inverse

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

$$X \cdot Y + X \cdot Z = X \cdot (Y + Z)$$

Distributive

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

$$X + Y = Y + X$$

Commutative

Example 2: Boolean Algebraic Proofs

$AB + A'C + BC = AB + A'C$	Consensus Theorem	
Proof Steps	Axiom or Theorem used	
$AB + A'C + BC$		
$= AB + A'C + 1 \cdot BC$	$X \cdot 1 = X$	Neutral Element
$= AB + A'C + (A + A') \cdot BC$	$X + X' = 1$	Inverse
$= AB + A'C + ABC + A'C$	$X \cdot Y + X \cdot Z = X \cdot (Y + Z)$	Distributive
$= AB + ABC + A'C + A'C$	$X + Y = Y + X$	Commutative
$= AB + A'C + A'C$	$X + XY = X$	Absorption

Example 2: Boolean Algebraic Proofs

Proof Steps	Axiom or Theorem used	
$AB + A'C + BC = AB + A'C$	Consensus Theorem	
$AB + A'C + BC$		
$= AB + A'C + 1 \cdot BC$	$X \cdot 1 = X$	Neutral Element
$= AB + A'C + (A + A') \cdot BC$	$X + X' = 1$	Inverse
$= AB + A'C + ABC + A'C$	$X \cdot Y + X \cdot Z = X \cdot (Y + Z)$	Distributive
$= AB + ABC + A'C + A'C$	$X + Y = Y + X$	Commutative
$= AB + A'C + A'C$	$X + XY = X$	Absorption
$= AB + A'(C + BC)$	$X \cdot Y + X \cdot Z = X \cdot (Y + Z)$	Distributive

Example 2: Boolean Algebraic Proofs

Proof Steps	Axiom or Theorem used	
$AB + A'C + BC = AB + A'C$	Consensus Theorem	
$AB + A'C + BC$		
$= AB + A'C + 1 \cdot BC$	$X \cdot 1 = X$	Neutral Element
$= AB + A'C + (A + A') \cdot BC$	$X + X' = 1$	Inverse
$= AB + A'C + ABC + A'C$	$X \cdot Y + X \cdot Z = X \cdot (Y + Z)$	Distributive
$= AB + ABC + A'C + A'C$	$X + Y = Y + X$	Commutative
$= AB + A'C + A'C$	$X + XY = X$	Absorption
$= AB + A'(C + BC)$	$X \cdot Y + X \cdot Z = X \cdot (Y + Z)$	Distributive
$= AB + A'C$	$X + XY = X$	Absorption

Example 3: Boolean Algebraic Proofs

$$(X+Y)Z + XYI = YI(X+Z)$$

Example 3: Boolean Algebraic Proofs

$$(X+Y)Z + XYI = YI(X+Z)$$

Proof Steps

Axiom or Theorem used

Example 3: Boolean Algebraic Proofs

$$(X+Y)Z + XYI = YI(X+Z)$$

Proof Steps Axiom or Theorem used

$$(X+Y)Z + XYI$$

Example 3: Boolean Algebraic Proofs

$$(X+Y)Z + XYI = YI(X+Z)$$

Proof Steps

$$(X+Y)Z + XYI$$

$$= XIZ + XYI$$

Axiom or Theorem used

$$(X+Y)I = XI \cdot YI$$

De Morgan

Example 3: Boolean Algebraic Proofs

$$(X+Y)Z + XYI = YI(X+Z)$$

Proof Steps Axiom or Theorem used

$$(X+Y)Z + XYI$$

$$= XIZ + XYI$$

$$= YI(XIZ + X)$$

$(X+Y)I = XI \cdot YI$

$X \cdot Y + X \cdot Z = X \cdot (Y + Z)$

De Morgan

Distributive

Example 3: Boolean Algebraic Proofs

$$(X+Y)Z + XYI = YI(X+Z)$$

Proof Steps Axiom or Theorem used

$$(X+Y)Z + XYI$$

$$= XIZ + XYI \quad (X+Y)I = XI \cdot YI \quad \text{De Morgan}$$

$$= YI(XIZ + X) \quad X \cdot Y + X \cdot Z = X \cdot (Y + Z) \quad \text{Distributive}$$

$$= YI(XI + X)(Z + X) \quad (X+Y) \cdot (X+Z) = X + (Y \cdot Z) \quad \text{Distributive}$$

Example 3: Boolean Algebraic Proofs

$$(X+Y)Z + XYI = YI(X+Z)$$

Proof Steps Axiom or Theorem used

$$(X+Y)Z + XYI$$

$$= XIZ + XYI \quad (X+Y)I = XI \cdot YI$$

$$= YI(XIZ + X) \quad X \cdot Y + X \cdot Z = X \cdot (Y + Z) \quad \text{De Morgan}$$

$$= YI(XI + X)(Z + X) \quad (X+Y) \cdot (X+Z) = X + (Y \cdot Z) \quad \text{Distributive}$$

$$= YI1(Z + X) \quad X + XI = 1 \quad \text{Inverse}$$

Example 3: Boolean Algebraic Proofs

$$(X+Y)Z + XYI = YI(X+Z)$$

Proof Steps Axiom or Theorem used

$$(X+Y)Z + XYI$$

$$= XIZ + XYI \quad (X+Y)I = XI \cdot YI$$

$$= YI(XIZ + X) \quad X \cdot Y + X \cdot Z = X \cdot (Y + Z)$$

$$= YI(XI + X)(Z + X) \quad (X+Y) \cdot (X+Z) = X + (Y \cdot Z)$$

$$= YI1(Z + X) \quad X + XI = 1$$

$$= YI(Z + X) \quad X \cdot 1 = X$$

De Morgan

Distributive

Distributive

Inverse

Neutral Element

Boolean Function Evaluation

$$F_1 = xyz'$$

$$F_2 = x + y/z$$

$$F_3 = x'y'z' + x'yz + xy'$$

$$F_4 = xy' + x'z$$

$$F_4' = (xy' + x'z)' = (x' + y)(x + z')$$

Boolean Function Evaluation

$$F1 = xyz'$$

$$F2 = x + y/z$$

$$F3 = x'y'z' + x'yz + xy'$$

$$F4 = xy' + x'z$$

$$F4' = (xy' + x'z)' = (x' + y)(x + z')$$

x	y	z	F1	F2
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	0	0
1	0	0	0	1
1	0	1	0	1
1	1	0	1	1
1	1	1	0	1

Boolean Function Evaluation

$$F1 = xyz'$$

$$F2 = x + y/z$$

$$F3 = x'y'z' + x'yz + xy'$$

$$F4 = xy' + x'z$$

$$F4' = (xy' + x'z)' = (x' + y)(x + z')$$

x	y	z	F1	F2	F3
0	0	0	0	0	
0	0	1	0	1	
0	1	0	0	0	
0	1	1	0	0	
1	0	0	0	1	
1	0	1	0	1	
1	1	0	1	1	
1	1	1	0	1	

Boolean Function Evaluation

$$F1 = xyz'$$

$$F2 = x + y/z$$

$$F3 = x'y'z' + x'yz + xy'$$

$$F4 = xy' + x'z$$

$$F4' = (xy' + x'z)' = (x' + y)(x + z')$$

x	y	z	F1	F2	F3
0	0	0	0	0	1
0	0	1	0	1	
0	1	0	0	0	
0	1	1	0	0	1
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	1	1	
1	1	1	0	1	

Boolean Function Evaluation

$$F1 = xyz'$$

$$F2 = x + y/z$$

$$F3 = x'y'z' + x'yz + xy'$$

$$F4 = xy' + x'z$$

$$F4' = (xy' + x'z)' = (x' + y)(x + z')$$

x	y	z	F1	F2	F3
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	0	0	1
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	1	1	0
1	1	1	0	1	0

Boolean Function Evaluation

$$F1 = xyz'$$

$$F2 = x + y/z$$

$$F3 = x'y'z' + x'yz + xy'$$

$$F4 = xy' + x'z$$

$$F4' = (xy' + x'z)' = (x' + y)(x + z')$$

x	y	z	F1	F2	F3	F4
0	0	0	0	0	1	
0	0	1	0	1	0	
0	1	0	0	0	0	
0	1	1	0	0	1	
1	0	0	0	1	1	
1	0	1	0	1	1	
1	1	0	1	1	0	
1	1	1	0	1	0	

Boolean Function Evaluation

$$F1 = xyz'$$

$$F2 = x + y/z$$

$$F3 = x'y'z' + x'yz + xy'$$

$$F4 = xy' + x'z$$

$$F4' = (xy' + x'z)' = (x' + y)(x + z')$$

x	y	z	F1	F2	F3	F4
0	0	0	0	0	1	1
0	0	1	0	1	0	1
0	1	0	0	0	0	0
0	1	1	0	0	1	1
1	0	0	0	1	1	1
1	0	1	0	1	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	0

Boolean Function Evaluation

$$F1 = xyz'$$

$$F2 = x + y/z$$

$$F3 = x'y'z' + x'yz + xy'$$

$$F4 = xy' + x'z$$

$$F4' = (xy' + x'z)' = (x' + y)(x + z')$$

x	y	z	F1	F2	F3	F4
0	0	0	0	0	1	0
0	0	1	0	1	0	1
0	1	0	0	0	0	0
0	1	1	0	0	1	1
1	0	0	0	1	1	1
1	0	1	0	1	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	0

Boolean Function Evaluation

$$F1 = xyz'$$

$$F2 = x + y/z$$

$$F3 = x'y'z' + x'yz + xy'$$

$$F4 = xy' + x'z$$

$$F4' = (xy' + x'z)' = (x' + y)(x + z')$$

x	y	z	F1	F2	F3	F4	F4'
0	0	0	0	0	1	0	
0	0	1	0	1	0	1	
0	1	0	0	0	0	0	
0	1	1	0	0	1	1	
1	0	0	0	1	1	1	
1	0	1	0	1	1	1	
1	1	0	1	1	0	0	
1	1	1	0	1	0	0	

Boolean Function Evaluation

$$F1 = xyz'$$

$$F2 = x + y/z$$

$$F3 = x'y'z' + x'yz + xy'$$

$$F4 = xy' + x'z$$

$$F4' = (xy' + x'z)' = (x' + y)(x + z')$$

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	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

Boolean Function Evaluation

$$F_1 = xyz'$$

$$F_2 = x + y/z$$

$$F_3 = x'y'z' + x'yz + xy'$$

$$F_4 = xy' + x'z$$

$$F_{4'} = (xy' + x'z)' = (x' + y)(x + z')$$

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	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

- ▶ If the input number is $= n$
- ▶ There are 2^n different input combinations
- ▶ Hence, 2^{2^n} different Boolean functions can be defined

Expression Simplification

Simplify to contain the smallest number of literals:

$$A B + A' C D + A' B D + A' C D' + A B C D$$

Expression Simplification

Simplify to contain the smallest number of literals:

$$\begin{aligned} & A B + A' C D + A' B D + A' C D' + A B C D \\ & = A B + A B C D + A' C D + A' B D + A' C D' \end{aligned}$$

Expression Simplification

Simplify to contain the smallest number of literals:

$$\begin{aligned} & A B + A' C D + A' B D + A' C D' + A B C D \\ & = A B + A B C D + A' C D + A' B D + A' C D' \\ & = A B + A' C D + A' C D' + A' B D \end{aligned}$$

Expression Simplification

Simplify to contain the smallest number of literals:

$$\begin{aligned} & A B + A' C D + A' B D + A' C D' + A B C D \\ & = A B + A B C D + A' C D + A' B D + A' C D' \\ & = A B + A' C D + A' C D' + A' B D \\ & = A B + A' C (D + D') + A' B D \end{aligned}$$

Expression Simplification

Simplify to contain the smallest number of literals:

$$\begin{aligned} & A B + A' C D + A' B D + A' C D' + A B C D \\ & = A B + A B C D + A' C D + A' B D + A' C D' \\ & = A B + A' C D + A' C D' + A' B D \\ & = A B + A' C (D + D') + A' B D \\ & = A B + A' B D + A' C \end{aligned}$$

Expression Simplification

Simplify to contain the smallest number of literals:

$$\begin{aligned} & A B + A' C D + A' B D + A' C D' + A B C D \\ & = A B + A B C D + A' C D + A' B D + A' C D' \\ & = A B + A' C D + A' C D' + A' B D \\ & = A B + A' C (D + D') + A' B D \\ & = A B + A' B D + A' C \\ & = B (A + A' D) + A' C \end{aligned}$$

Expression Simplification

Simplify to contain the smallest number of literals:

$$\begin{aligned} & A B + A' C D + A' B D + A' C D' + A B C D \\ &= A B + A B C D + A' C D + A' B D + A' C D' \\ &= A B + A' C D + A' C D' + A' B D \\ &= A B + A' C (D + D') + A' B D \\ &= A B + A' B D + A' C \\ &= B (A + A' D) + A' C \\ &= B (A + A')(A + D) + A' C \end{aligned}$$

Expression Simplification

Simplify to contain the smallest number of literals:

$$\begin{aligned} & A B + A' C D + A' B D + A' C D' + A B C D \\ &= A B + A B C D + A' C D + A' B D + A' C D' \\ &= A B + A' C D + A' C D' + A' B D \\ &= A B + A' C (D + D') + A' B D \\ &= A B + A' B D + A' C \\ &= B (A + A' D) + A' C \\ &= B (A + A')(A + D) + A' C \\ &= B (A + D) + A' C \end{aligned}$$

5 literals

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., x'), there are 2^n minterms for n variables.
- ▶ Example: Two variables (X and Y) produce $2 \times 2 = 4$ combinations:
 - ▶ XY (both normal)
 - ▶ XY' (X normal, Y complemented)
 - ▶ X/Y (X complemented, Y normal)
 - ▶ X/Y' (both complemented)
- ▶ Thus there are four minterms of two variables.

Maxterms

- ▶ Maxterms are OR 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., x'), there are 2^n maxterms for n variables.
- ▶ Example: Two variables (X and Y) produce $2 \times 2 = 4$ combinations:
 - ▶ $X+Y$ (both normal)
 - ▶ $X+Y'$ (X normal, Y complemented)
 - ▶ $X'+Y$ (X complemented, Y normal)
 - ▶ $X'+Y'$ (both complemented)
- ▶ Thus there are four maxterms of two variables.

Maxterms and Minterms

- ▶ **Examples:** Two variable minterms and maxterms.

Index	Minterm	Maxterm
0	$x'y'$	$x+y$
1	$x'y$	$x+y'$
2	xy'	$x'+y$
3	xy	$x'+y'$

- ▶ 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 an index
- ▶ The index 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: $(a + b + c')$, $(a + b + c)$
 - ▶ Terms: $(b + a + c)$, $a'c'$, b , and $(c + b + a)$ are NOT in standard order.
 - ▶ Minterms: $a'b'c$, $a'b'c$, $a'b'c$
 - ▶ Terms: $(a + c)$, $b'c$, and $(a' + b)$ 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 normal form or complemented form.
- ▶ For Minterms:
 - ▶ “1” means the variable is “Not Complemented” and
 - ▶ “0” means the variable is “Complemented”.
- ▶ For Maxterms:
 - ▶ “1” means the variable is “Complemented” and
 - ▶ “0” means the variable is “Not 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 ($X\bar{I}$, $Y\bar{I}$, $Z\bar{I}$) and no variables are complemented for Maxterm 0 (X, Y, Z).
 - ▶ Minterm 0, called m_0 is $X\bar{Y}\bar{Z}$.
 - ▶ Maxterm 0, called M_0 is $(X+Y+Z)$.
 - ▶ Minterm 6 ?
 - ▶ Maxterm 6 ?

Index Examples – Four Variables

Index	Binary Pattern	Minterm	Maxterm
i		m_i	M_i
0	0000	$a'b'c'd'$	$a+b+c+d$
1	0001	$a'b'c'd$?
3	0011	?	$a+b+c'+d'$
5	0101	$a'bc'd$	$a+b'+c+d'$
7	0111	?	$a+b'+c'+d'$
10	1010	$ab'cd'$	$a'+b+c'+d$
13	1101	$abc'd$?
15	1111	?	$a'+b'+c'+d'$

Minterm and Maxterm Relationship

- ▶ Review: DeMorgan's Theorem $(x \cdot y)' = x' + y'$ and $(x + y)' = x'y'$
- ▶ Two-variable example: $M_2 = x' + y$ and $m_2 = x \cdot y'$
- ▶ 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: $M_i = m_i'$ and $m_i = M_i'$

Minterm Function Example

- ▶ Find the truth table of $F(x, y, z) = m_1 + m_4 + m_7$

Minterm Function Example

- ▶ Find the truth table of $F(x, y, z) = m_1 + m_4 + m_7$
- ▶ $F(x, y, z) = x'y'z + xy'z' + xyz$

Minterm Function Example

- ▶ Find the truth table of $F(x, y, z) = m_1 + m_4 + m_7$
- ▶ $F(x, y, z) = x'y'z + xy'z' + xyz$

x	y	z	index	m_1	+	m_4	+	m_7	=	$F(x, y, z)$
---	---	---	-------	-------	---	-------	---	-------	---	--------------

Minterm Function Example

- ▶ Find the truth table of $F(x, y, z) = m_1 + m_4 + m_7$
- ▶ $F(x, y, z) = x'y'z + xy'z' + xyz$

x	y	z	index	m_1	+	m_4	+	m_7	=	$F(x, y, z)$
0	0	0	0	0	+	0	+	0	=	0

Minterm Function Example

- ▶ Find the truth table of $F(x, y, z) = m_1 + m_4 + m_7$
- ▶ $F(x, y, z) = x'y'z + xy'z' + xyz$

x	y	z	index	m_1	+	m_4	+	m_7	=	$F(x, y, z)$
0	0	0	0	0	+	0	+	0	=	0
0	0	1	1	1	+	0	+	0	=	1

Minterm Function Example

- ▶ Find the truth table of $F(x, y, z) = m_1 + m_4 + m_7$
- ▶ $F(x, y, z) = x'y'z + xy'z' + xyz$

x	y	z	index	m_1	+	m_4	+	m_7	=	$F(x, y, z)$
0	0	0	0	0	+	0	+	0	=	0
0	0	1	1	1	+	0	+	0	=	1
0	1	0	2	0	+	0	+	0	=	0

Minterm Function Example

- ▶ Find the truth table of $F(x, y, z) = m_1 + m_4 + m_7$
- ▶ $F(x, y, z) = x'y'z + xy'z' + xyz$

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

Minterm Function Example

- ▶ Find the truth table of $F(x, y, z) = m_1 + m_4 + m_7$
- ▶ $F(x, y, z) = \overline{x}\overline{y}z + \overline{x}y\overline{z} + xyz$

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

Minterm Function Example

- ▶ Find the truth table of $F(x, y, z) = m_1 + m_4 + m_7$
- ▶ $F(x, y, z) = x'y'z + xy'z' + xyz$

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

Minterm Function Example

- ▶ Find the truth table of $F(x, y, z) = m_1 + m_4 + m_7$
- ▶ $F(x, y, z) = \overline{x}\overline{y}z + \overline{x}y\overline{z} + xyz$

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

Minterm Function Example

- ▶ Find the truth table of $F(x, y, z) = m_1 + m_4 + m_7$
- ▶ $F(x, y, z) = x'y'z + xy'z' + xyz$

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

Minterm Function Example

- ▶ $F(A, B, C, D, E) = m_2 + m_9 + m_{17} + m_{23}$

Minterm Function Example

- ▶ $F(A, B, C, D, E) = m_2 + m_9 + m_{17} + m_{23}$
- ▶ $F(A, B, C, D, E) =$
 $A'B'C'D'E' + A'BC'D'E + AB'C'D'E + AB'CDE$

Maxterm Function Example

- ▶ Find the truth table of $F(x, y, z) = M_0M_2M_3M_5M_6$

Maxterm Function Example

- ▶ Find the truth table of $F(x, y, z) = M_0M_2M_3M_5M_6$
- ▶ $F(x, y, z) =$
 $(x + y + z)(x + y' + z)(x + y' + z')(x' + y + z')(x' + y' + z)$

Maxterm Function Example

- ▶ Find the truth table of $F(x, y, z) = M_0 M_2 M_3 M_5 M_6$
 - ▶ $F(x, y, z) =$
 $(x + y + z)(x + y' + z)(x + y' + z')(x' + y + z')(x' + y' + z)$
-
- | x | y | z | index | $M_0.$ | $M_2.$ | $M_3.$ | $M_5.$ | M_6 | = | $F(x, y, z)$ |
|---|---|---|-------|--------|--------|--------|--------|-------|---|--------------|
|---|---|---|-------|--------|--------|--------|--------|-------|---|--------------|

Maxterm Function Example

- ▶ Find the truth table of $F(x, y, z) = M_0 M_2 M_3 M_5 M_6$
- ▶ $F(x, y, z) =$
 $(x + y + z)(x + y' + z)(x + y' + z')(x' + y + z')(x' + y' + z)$

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

Maxterm Function Example

- ▶ Find the truth table of $F(x, y, z) = M_0 M_2 M_3 M_5 M_6$
- ▶ $F(x, y, z) =$
 $(x + y + z)(x + y' + z)(x + y' + z')(x' + y + z')(x' + y' + z)$

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

Maxterm Function Example

- ▶ Find the truth table of $F(x, y, z) = M_0 M_2 M_3 M_5 M_6$
- ▶ $F(x, y, z) =$
 $(x + y + z)(x + y' + z)(x + y' + z')(x' + y + z')(x' + y' + z)$

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.	1.	1.	1	=	0

Maxterm Function Example

- ▶ Find the truth table of $F(x, y, z) = M_0 M_2 M_3 M_5 M_6$
- ▶ $F(x, y, z) =$
 $(x + y + z)(x + y' + z)(x + y' + z')(x' + y + z')(x' + y' + z)$

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.	1.	1.	1	=	0
0	1	1	3	1.	1.	0.	1.	1	=	0

Maxterm Function Example

- ▶ Find the truth table of $F(x, y, z) = M_0 M_2 M_3 M_5 M_6$
- ▶ $F(x, y, z) =$
 $(x + y + z)(x + y' + z)(x + y' + z')(x' + y + z')(x' + y' + z)$

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.	1.	1.	1	=	0
0	1	1	3	1.	1.	0.	1.	1	=	0
1	0	0	4	1.	1.	1.	1.	1	=	1

Maxterm Function Example

- ▶ Find the truth table of $F(x, y, z) = M_0 M_2 M_3 M_5 M_6$
- ▶ $F(x, y, z) =$
 $(x + y + z)(x + y' + z)(x + y' + z')(x' + y + z')(x' + y' + z)$

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.	1.	1.	1	=	0
0	1	1	3	1.	1.	0.	1.	1	=	0
1	0	0	4	1.	1.	1.	1.	1	=	1
1	0	1	5	1.	1.	1.	0.	1	=	0

Maxterm Function Example

- ▶ Find the truth table of $F(x, y, z) = M_0 M_2 M_3 M_5 M_6$
- ▶ $F(x, y, z) =$
 $(x + y + z)(x + y' + z)(x + y' + z')(x' + y + z')(x' + y' + z)$

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.	1.	1.	1	=	0
0	1	1	3	1.	1.	0.	1.	1	=	0
1	0	0	4	1.	1.	1.	1.	1	=	1
1	0	1	5	1.	1.	1.	0.	1	=	0
1	1	0	6	1.	1.	1.	1.	0	=	0

Maxterm Function Example

- ▶ Find the truth table of $F(x, y, z) = M_0 M_2 M_3 M_5 M_6$
- ▶ $F(x, y, z) =$
 $(x + y + z)(x + y' + z)(x + y' + z')(x' + y + z')(x' + y' + z)$

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.	1.	1.	1	=	0
0	1	1	3	1.	1.	0.	1.	1	=	0
1	0	0	4	1.	1.	1.	1.	1	=	1
1	0	1	5	1.	1.	1.	0.	1	=	0
1	1	0	6	1.	1.	1.	1.	0	=	0
1	1	1	7	1.	1.	1.	1.	1	=	1

Maxterm Function Example

- ▶ $F(A, B, C, D) = M_3.M_8.M_{11}.M_{14}$

Maxterm Function Example

- ▶ $F(A, B, C, D) = M_3 \cdot M_8 \cdot M_{11} \cdot M_{14}$
- ▶ $F(A, B, C, D) = (A + B + C' + D')(A' + B + C + D)(A' + B + C' + D')(A' + B' + C' + D)$

Canonical Sum of Products

- ▶ Any Boolean function can be expressed as a OR of Minterms
- ▶ For the function table, the minterms used are the terms corresponding to the 1's

Canonical Sum of Products

- ▶ Any Boolean function can be expressed as a OR 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 variables

Canonical Sum of Products

- ▶ Any Boolean function can be expressed as a OR 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 variables
- ▶ **Example:** $F(A, B, C) = A + B'C$

Canonical Sum of Products

- ▶ Any Boolean function can be expressed as a OR 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 variables
- ▶ **Example:** $F(A, B, C) = A + B'C$
- ▶ There are three variables, A, B, and C which we take to be the standard order.

Canonical Sum of Products

- ▶ Any Boolean function can be expressed as a OR 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 variables
- ▶ Example: $F(A, B, C) = A + B'C$
- ▶ There are three variables, A, B, and C which we take to be the standard order.
- ▶ Expanding the terms with missing variables:

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

Canonical Sum of Products

- ▶ Any Boolean function can be expressed as a OR 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 variables
- ▶ Example: $F(A, B, C) = A + B'C$
- ▶ There are three variables, A, B, and C which we take to be the standard order.
- ▶ Expanding the terms with missing variables:
$$F(A, B, C) = A(B + B')\left(C + C'\right) + \left(A + A'\right)B'C$$
- ▶ Distributing the literals over parenthesis:
$$F(A, B, C) = ABC + ABC' + AB'C + AB'C' + AB'C + A'B'C$$

Canonical Sum of Products

- ▶ Any Boolean function can be expressed as a OR 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 variables
- ▶ Example: $F(A, B, C) = A + B'C$
- ▶ There are three variables, A, B, and C which we take to be the standard order.
- ▶ Expanding the terms with missing variables:

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

- ▶ Distributing the literals over parenthesis:

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

- ▶ Collect terms:

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

Canonical Sum of Products

- ▶ Any Boolean function can be expressed as a OR 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 variables
- ▶ Example: $F(A, B, C) = A + B'C$
- ▶ There are three variables, A, B, and C which we take to be the standard order.
- ▶ Expanding the terms with missing variables:
$$F(A, B, C) = A(B + B')\left(C + C'\right) + \left(A + A'\right)B'C$$
- ▶ Distributing the literals over parenthesis:
$$F(A, B, C) = ABC + ABC' + AB'C + AB'C' + AB'C + A'B'C$$
- ▶ Collect terms:
$$F(A, B, C) = ABC + ABC' + AB'C + AB'C' + A'B'C$$
- ▶ Express as SOP: $F(A, B, C) = m_7 + m_6 + m_5 + m_4 + m_1$

Shorthand SOP Form

- ▶ From the previous example, we started with:
 $F(A, B, C) = A + B'C$
- ▶ We ended up with: $F(A, B, C) = m_1 + m_4 + m_5 + m_6 + m_7$
- ▶ This can be denoted in the formal shorthand:
 $F(A, B, C) = \Sigma_m(1, 4, 5, 6, 7)$
- ▶ 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 an AND of Maxterms
- ▶ For the function table, the maxterms used are the terms corresponding to the 0's

Canonical Product of Sums

- ▶ Any Boolean function can be expressed as an AND of Maxterms
- ▶ For the function table, the maxterms used are the terms corresponding to the 0's
- ▶ For expressions, expand all terms first to explicitly list all variables

Canonical Product of Sums

- ▶ Any Boolean function can be expressed as an AND of Maxterms
- ▶ For the function table, the maxterms used are the terms corresponding to the 0's
- ▶ For expressions, expand all terms first to explicitly list all variables
- ▶ **Example:** $F(A, B, C) = A(B' + C)$

Canonical Product of Sums

- ▶ Any Boolean function can be expressed as an AND of Maxterms
- ▶ For the function table, the maxterms used are the terms corresponding to the 0's
- ▶ For expressions, expand all terms first to explicitly list all variables
- ▶ Example: $F(A, B, C) = A(B' + C)$
- ▶ Expanding the terms with missing variables:

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

Canonical Product of Sums

- ▶ Any Boolean function can be expressed as an AND of Maxterms
- ▶ For the function table, the maxterms used are the terms corresponding to the 0's
- ▶ For expressions, expand all terms first to explicitly list all variables
- ▶ Example: $F(A, B, C) = A(B' + C)$
- ▶ Expanding the terms with missing variables:
$$F(A, B, C) = (A + BB' + CC')(AA' + B' + C)$$
- ▶ Distributing the literals over parenthesis:
 - ▶
$$F(A, B, C) = ((A+B)(A+B') + CC')(A+B'+C)(A'+B'+C)$$

Canonical Product of Sums

- ▶ Any Boolean function can be expressed as an AND of Maxterms
- ▶ For the function table, the maxterms used are the terms corresponding to the 0's
- ▶ For expressions, expand all terms first to explicitly list all variables
- ▶ Example: $F(A, B, C) = A(B' + C)$
- ▶ Expanding the terms with missing variables:
$$F(A, B, C) = (A + BB' + CC')(AA' + B' + C)$$
- ▶ Distributing the literals over parenthesis:
 - ▶
$$\begin{aligned} F(A, B, C) &= ((A+B)(A+B') + CC')(A+B'+C)(A'+B'+C) \\ &= ((A+B)(A+B') + C)((A+B)(A+B') + C')(A+B'+C)(A'+B'+C) \end{aligned}$$

Canonical Product of Sums

- ▶ Any Boolean function can be expressed as an AND of Maxterms
- ▶ For the function table, the maxterms used are the terms corresponding to the 0's
- ▶ For expressions, expand all terms first to explicitly list all variables
- ▶ Example: $F(A, B, C) = A(B' + C)$
- ▶ Expanding the terms with missing variables:
$$F(A, B, C) = (A + BB' + CC')(AA' + B' + C)$$
- ▶ Distributing the literals over parenthesis:
 - ▶
$$\begin{aligned} F(A, B, C) &= ((A+B)(A+B') + CC')(A+B'+C)(A'+B'+C) \\ &= ((A+B)(A+B') + C)((A+B)(A+B') + C')(A+B'+C)(A'+B'+C) \end{aligned}$$
 - ▶
$$(A+B+C)(A+B'+C)(A+B+C')(A+B'+C')(A+B'+C)(A'+B'+C)$$

Canonical Product of Sums

- ▶ Any Boolean function can be expressed as an AND of Maxterms
- ▶ For the function table, the maxterms used are the terms corresponding to the 0's
- ▶ For expressions, expand all terms first to explicitly list all variables
- ▶ Example: $F(A, B, C) = A(B' + C)$
- ▶ Expanding the terms with missing variables:
$$F(A, B, C) = (A + BB' + CC')(AA' + B' + C)$$
- ▶ Distributing the literals over parenthesis:
 - ▶
$$\begin{aligned} F(A, B, C) &= ((A+B)(A+B') + CC')(A+B'+C)(A'+B'+C) \\ &= ((A+B)(A+B') + C)((A+B)(A+B') + C')(A+B'+C)(A'+B'+C) \end{aligned}$$
 - ▶
$$(A+B+C)(A+B'+C)(A+B+C')(A+B'+C')(A+B'+C)(A'+B'+C)$$
- ▶ Collect terms: $F(A, B, C) = (A+B+C)(A+B'+C)(A+B+C')(A+B'+C')(A'+B'+C)$

Canonical Product of Sums

- ▶ Any Boolean function can be expressed as an AND of Maxterms
- ▶ For the function table, the maxterms used are the terms corresponding to the 0's
- ▶ For expressions, expand all terms first to explicitly list all variables
- ▶ Example: $F(A, B, C) = A(B' + C)$
- ▶ Expanding the terms with missing variables:
$$F(A, B, C) = (A + BB' + CC')(AA' + B' + C)$$
- ▶ Distributing the literals over parenthesis:
 - ▶
$$\begin{aligned} F(A, B, C) &= ((A+B)(A+B') + CC')(A+B'+C)(A'+B'+C) \\ &= ((A+B)(A+B') + C)((A+B)(A+B') + C')(A+B'+C)(A'+B'+C) \\ &= (A+B+C)(A+B'+C)(A+B+C')(A+B'+C')(A+B'+C)(A'+B'+C) \end{aligned}$$
- ▶ Collect terms: $F(A, B, C) = (A+B+C)(A+B'+C)(A+B+C')(A+B'+C')(A'+B'+C)$
- ▶ Express as SOP: $F(A, B, C) = M_0 M_2 M_1 M_3 M_6$

Shorthand POS Form

- ▶ From the previous example, we started with:
 $F(A, B, C) = A(B' + C)$
- ▶ We ended up with: $F(A, B, C) = M_0M_2M_1M_3M_6$
- ▶ This can be denoted in the formal shorthand:
 $F(A, B, C) = \Pi_M(0, 1, 2, 3, 6)$
- ▶ Note that we explicitly show the standard variables in order and drop the “M” designators.

Function Complements

- ▶ The complement of a function expressed in SOP is constructed by selecting the minterms missing in the SOP canonical forms.
- ▶ The complement of a function expressed in POS is constructed by selecting the maxterms missing in the POS canonical forms.

Function Complements

- ▶ The complement of a function expressed in SOP is constructed by selecting the minterms missing in the SOP canonical forms.
- ▶ The complement of a function expressed in POS is constructed by selecting the maxterms missing in the POS canonical forms.
- ▶ Example:
 - ▶ $F(x, y, z) = \Sigma_m(1, 3, 5, 7)$

Function Complements

- ▶ The complement of a function expressed in SOP is constructed by selecting the minterms missing in the SOP canonical forms.
- ▶ The complement of a function expressed in POS is constructed by selecting the maxterms missing in the POS canonical forms.
- ▶ Example:
 - ▶ $F(x, y, z) = \Sigma_m(1, 3, 5, 7)$
 - ▶ $F(x, y, z) = \Pi_M(0, 2, 4, 6)$

Function Complements

- ▶ The complement of a function expressed in SOP is constructed by selecting the minterms missing in the SOP canonical forms.
- ▶ The complement of a function expressed in POS is constructed by selecting the maxterms missing in the POS canonical forms.
- ▶ Example:
 - ▶ $F(x, y, z) = \Sigma_m(1, 3, 5, 7)$
 - ▶ $F(x, y, z) = \Pi_M(0, 2, 4, 6)$
 - ▶ $F'(x, y, z) = \Sigma_m(0, 2, 4, 6)$

Function Complements

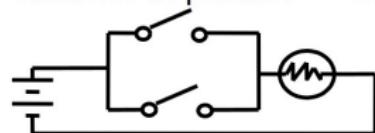
- ▶ The complement of a function expressed in SOP is constructed by selecting the minterms missing in the SOP canonical forms.
- ▶ The complement of a function expressed in POS is constructed by selecting the maxterms missing in the POS canonical forms.
- ▶ Example:
 - ▶ $F(x, y, z) = \Sigma_m(1, 3, 5, 7)$
 - ▶ $F(x, y, z) = \Pi_M(0, 2, 4, 6)$
 - ▶ $F'(x, y, z) = \Sigma_m(0, 2, 4, 6)$
 - ▶ $F'(x, y, z) = \Pi_M(1, 3, 5, 7)$

Implementation of Boolean Functions

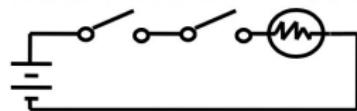
Using Switches

- ▶ 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

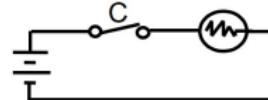
Switches in parallel => OR



Switches in series => AND

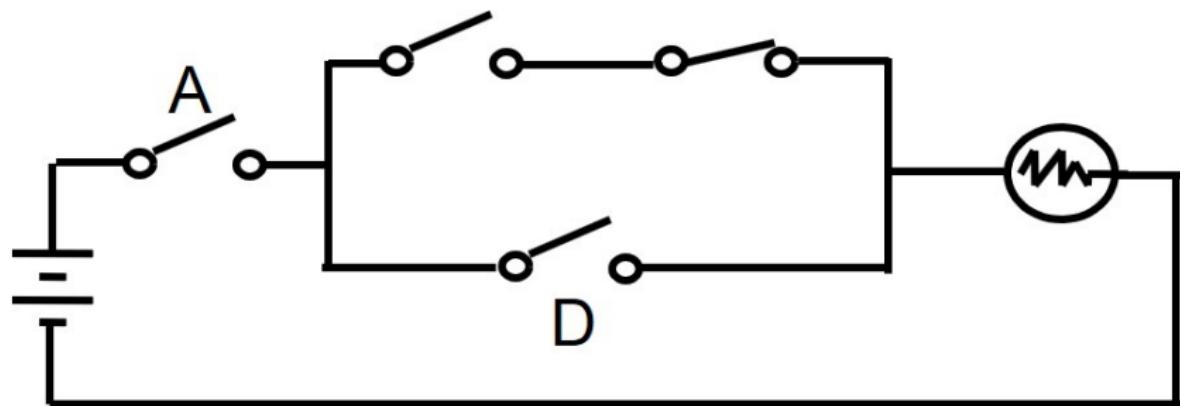


Normally-closed switch => NOT



Implementation of Boolean Functions

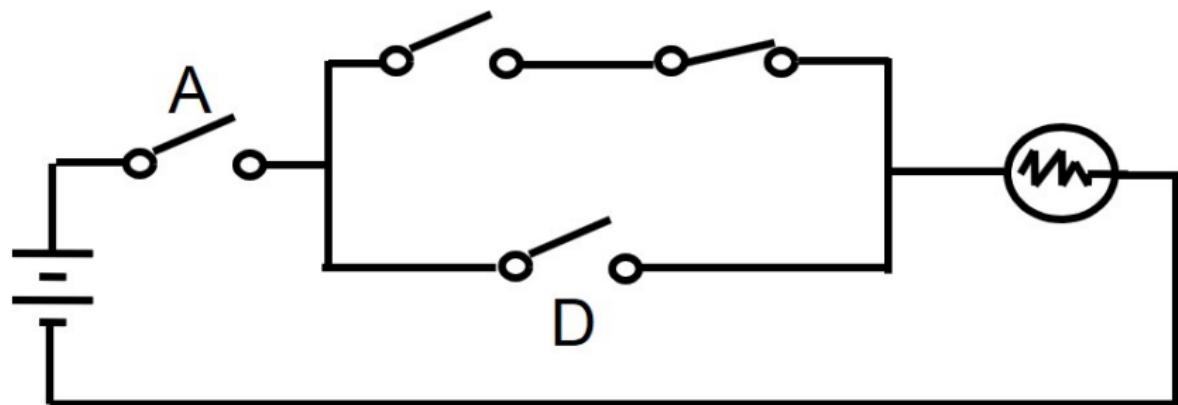
Example: Logic Using Switches



If $L = 1$, light is on and if $L = 0$ light is off.

Implementation of Boolean Functions

Example: Logic Using Switches

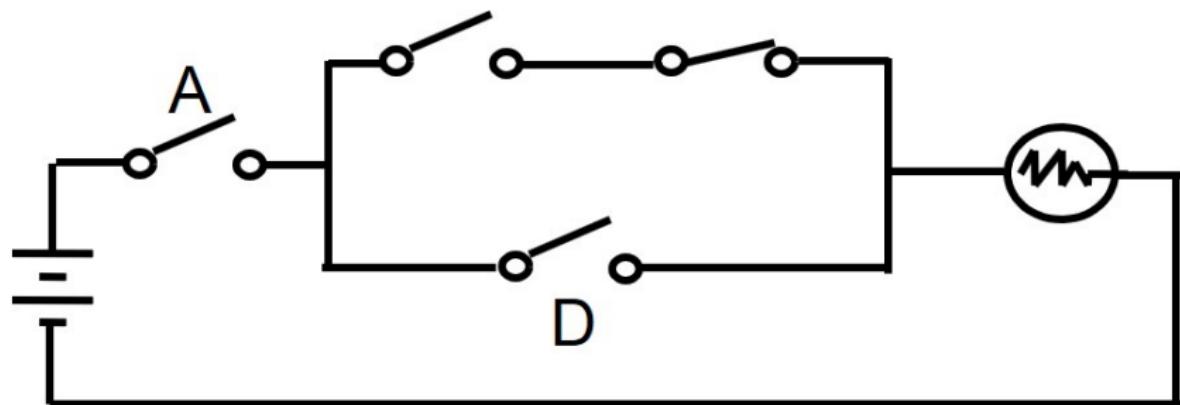


If $L = 1$, light is on and if $L = 0$ light is off.

- ▶ Sum of path functions:
 - ▶ $L(A, B, C, D) = ABCI + AD$

Implementation of Boolean Functions

Example: Logic Using Switches



If $L = 1$, light is on and if $L = 0$ light is off.

- ▶ Sum of path functions:
 - ▶ $L(A, B, C, D) = ABCI + AD$
- ▶ Product of cut functions:
 - ▶ $f(A, B, C, D) = A(B + D)(CI + D)$

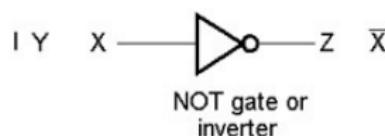
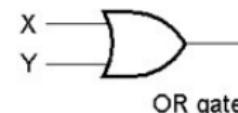
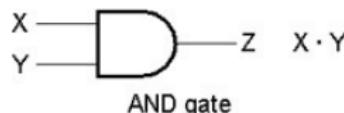
Logic Gates

- ▶ 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.

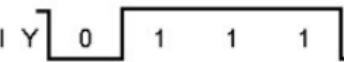
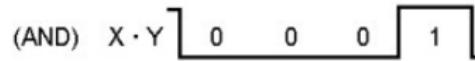
Logic Gate Symbols and Behavior

Logic gates have

- ▶ special symbols
- ▶ waveform behavior in time



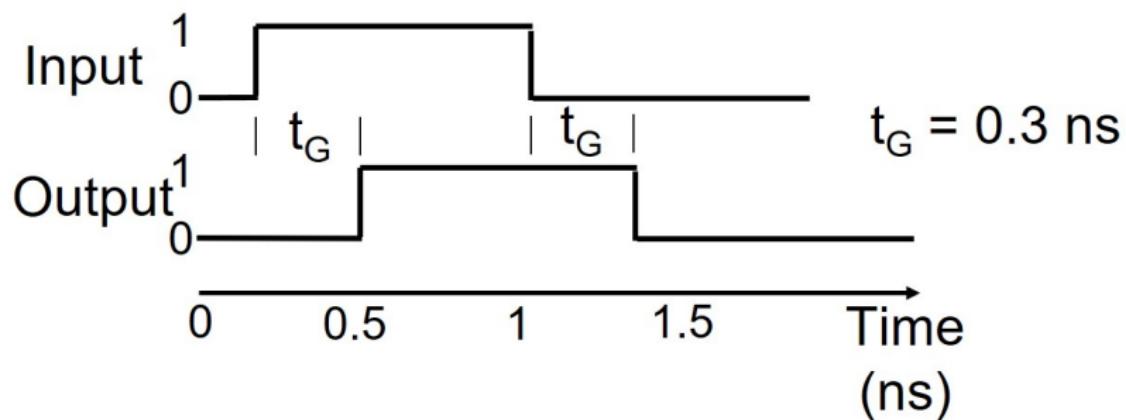
(a) Graphic symbols



(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

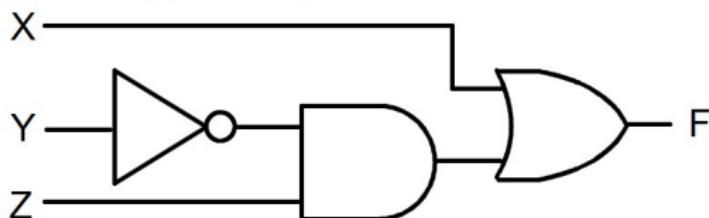
Truth Table

x	y	z	F
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

$$F(x, y, z) = x + y/z$$

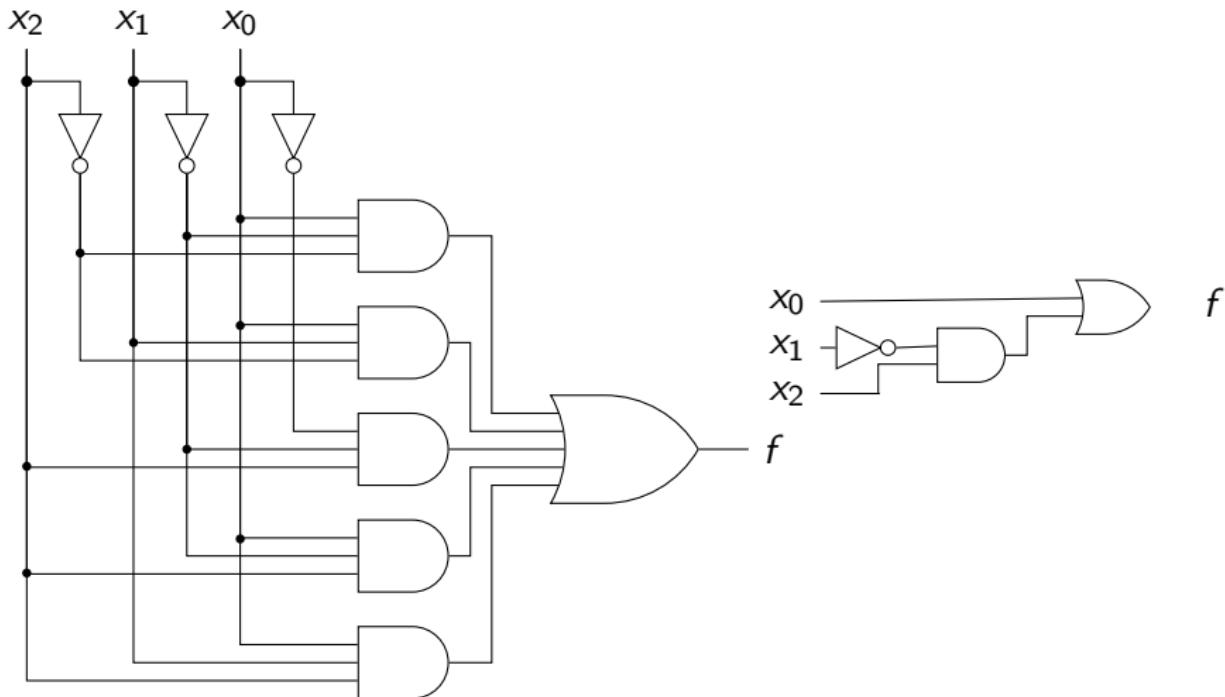
Logic Diagram



- ▶ 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!



Circuit Optimization

- ▶ **Goal:** To obtain the simplest implementation for a given function
- ▶ Optimization is a more formal approach to simplification that is performed using a specific procedure or algorithm
- ▶ Optimization requires a cost criterion to measure the simplicity of a circuit
- ▶ Distinct cost criteria we will use:
 - ▶ Literal cost (L)
 - ▶ Gate input cost (G)
 - ▶ Gate input cost with NOTs (GN)

Literal Cost

- ▶ **Literal:** a variable or its complement
- ▶ **Literal cost:** The number of literal appearances in a Boolean expression corresponding to the logic circuit diagram
- ▶ **Examples:**
 - ▶ $F = BD + AB'C + AC'DI$, L = 8

Literal Cost

- ▶ **Literal:** a variable or its complement
- ▶ **Literal cost:** The number of literal appearances in a Boolean expression corresponding to the logic circuit diagram
- ▶ **Examples:**
 - ▶ $F = BD + AB'C + AC'DI$, L = 8
 - ▶ $F = BD + AB'C + AB'DI + ABCI$, L = ?

Literal Cost

- ▶ **Literal:** a variable or its complement
- ▶ **Literal cost:** The number of literal appearances in a Boolean expression corresponding to the logic circuit diagram
- ▶ **Examples:**
 - ▶ $F = BD + AB'C + AC'D'$, L = 8
 - ▶ $F = BD + AB'C + AB'D' + ABCI$, L = ?
 - ▶ $F = (A + B)(A + D)(B + C + D')$, L = ?

Literal Cost

- ▶ **Literal:** a variable or its complement
- ▶ **Literal cost:** The number of literal appearances in a Boolean expression corresponding to the logic circuit diagram
- ▶ **Examples:**
 - ▶ $F = BD + AB'C + AC'D'$, L = 8
 - ▶ $F = BD + AB'C + AB'D' + ABCI$, L = ?
 - ▶ $F = (A + B)(A + D)(B + C + D')$, L = ?
 - ▶ Which solution is the best?

Gate Input Cost

- ▶ **Gate input costs:** The number of inputs to the gates in the implementation corresponding exactly to the given equation or equations. (G - inverters not counted, GN - inverters counted)
- ▶ For SOP and POS equations, it can be found from the equation(s) by finding the sum of:
 - ▶ all literal appearances
 - ▶ the number of terms excluding single literal terms,(G) and
 - ▶ optionally, the number of distinct complemented single literals (GN).
- ▶ **Examples:**
 - ▶ $F = BD + AB'C + AC'D'$, G = 11, GN = 14

Gate Input Cost

- ▶ **Gate input costs:** The number of inputs to the gates in the implementation corresponding exactly to the given equation or equations. (G - inverters not counted, GN - inverters counted)
- ▶ For SOP and POS equations, it can be found from the equation(s) by finding the sum of:
 - ▶ all literal appearances
 - ▶ the number of terms excluding single literal terms,(G) and
 - ▶ optionally, the number of distinct complemented single literals (GN).
- ▶ **Examples:**
 - ▶ $F = BD + AB'C + AC'D'$, G = 11, GN = 14
 - ▶ $F = BD + AB'C + AB'D' + ABCI$, G = ?, GN = ?

Gate Input Cost

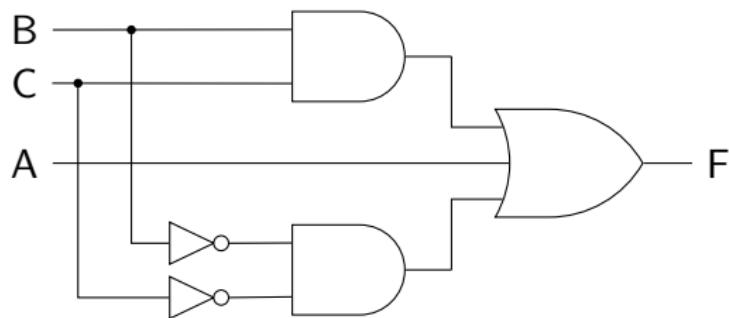
- ▶ **Gate input costs:** The number of inputs to the gates in the implementation corresponding exactly to the given equation or equations. (G - inverters not counted, GN - inverters counted)
- ▶ For SOP and POS equations, it can be found from the equation(s) by finding the sum of:
 - ▶ all literal appearances
 - ▶ the number of terms excluding single literal terms,(G) and
 - ▶ optionally, the number of distinct complemented single literals (GN).
- ▶ **Examples:**
 - ▶ $F = BD + AB'C + AC'D'$, G = 11, GN = 14
 - ▶ $F = BD + AB'C + AB'D' + ABC'$, G = ?, GN = ?
 - ▶ $F = (A+B)(A+D)(B+C+D')(B'+C'+D)$, G = ?, GN = ?

Gate Input Cost

- ▶ **Gate input costs:** The number of inputs to the gates in the implementation corresponding exactly to the given equation or equations. (G - inverters not counted, GN - inverters counted)
- ▶ For SOP and POS equations, it can be found from the equation(s) by finding the sum of:
 - ▶ all literal appearances
 - ▶ the number of terms excluding single literal terms,(G) and
 - ▶ optionally, the number of distinct complemented single literals (GN).
- ▶ **Examples:**
 - ▶ $F = BD + AB'C + AC'D'$, G = 11, GN = 14
 - ▶ $F = BD + AB'C + AB'D' + ABCI$, G = ?, GN = ?
 - ▶ $F = (A+B)(A+D)(B+C+D')$, G = ?, GN = ?
 - ▶ Which solution is the best?

Cost Criteria

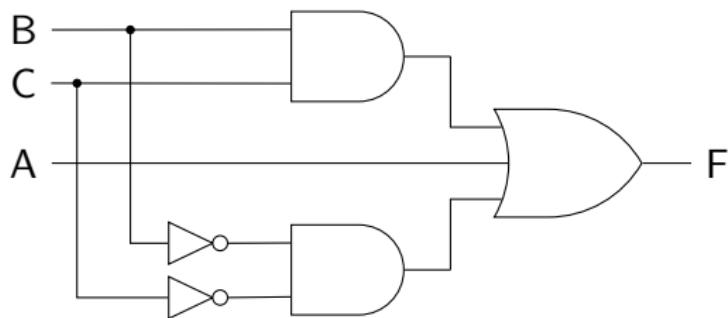
► Example:



- L (literal count) counts the AND inputs and the single literal OR input.
- G (gate input count) adds the remaining OR gate inputs
- GN (gate input count with NOTs) adds the inverter input

Cost Criteria

- ▶ Example:

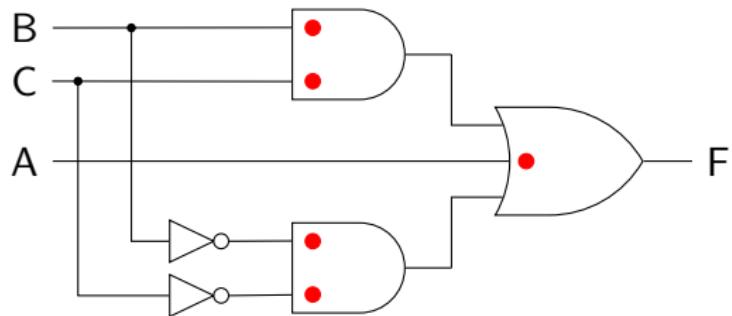


$$F = BC + A + B'C'$$

- ▶ L (literal count) counts the AND inputs and the single literal OR input.
- ▶ G (gate input count) adds the remaining OR gate inputs
- ▶ GN (gate input count with NOTs) adds the inverter input

Cost Criteria

- ▶ Example:



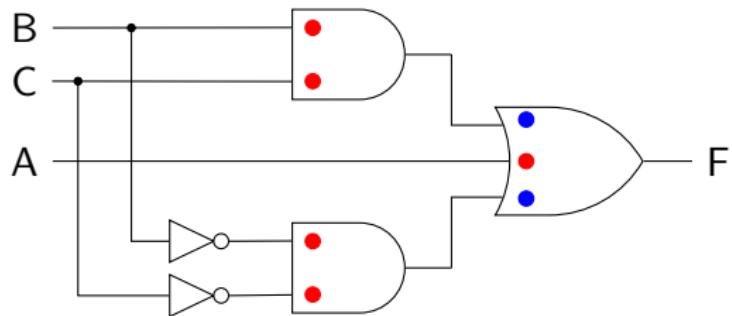
$$F = BC + A + B'C'$$

$$L = 5$$

- ▶ L (literal count) counts the AND inputs and the single literal OR input.
- ▶ G (gate input count) adds the remaining OR gate inputs
- ▶ GN (gate input count with NOTs) adds the inverter input

Cost Criteria

- ▶ Example:



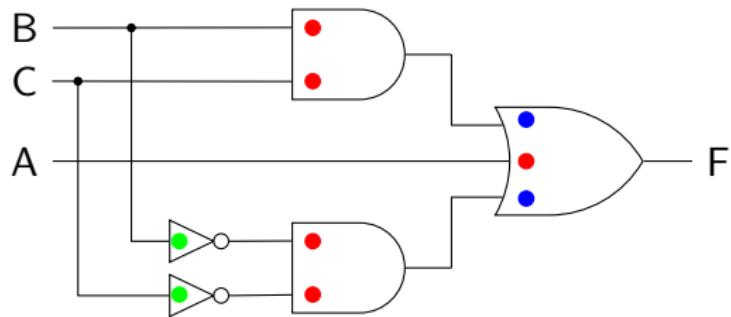
$$F = BC + A + B'C'$$

$$L = 5 \quad G = L + 2 = 7$$

- ▶ L (literal count) counts the AND inputs and the single literal OR input.
- ▶ G (gate input count) adds the remaining OR gate inputs
- ▶ GN (gate input count with NOTs) adds the inverter input

Cost Criteria

- ▶ Example:



$$F = BC + A + B'C'$$

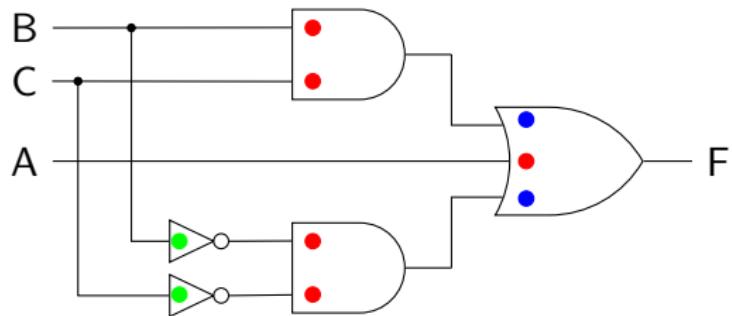
$$L = 5$$

$$G = L + 2 = 7$$

$$GN = G + 2 = 9$$

Cost Criteria

- ▶ Example:



$$F = BC + A + B'C'$$

$$L = 5$$

$$G = L + 2 = 7$$

$$GN = G + 2 = 9$$

- ▶ L (literal count) counts the AND inputs and the single literal OR input.
- ▶ G (gate input count) adds the remaining OR gate inputs
- ▶ GN (gate input count with NOTs) adds the inverter input