

Introduction to Digital Systems

Part I (4 lectures)

2022/2023

Introduction

Number Systems and Codes

Combinational Logic Design Principles

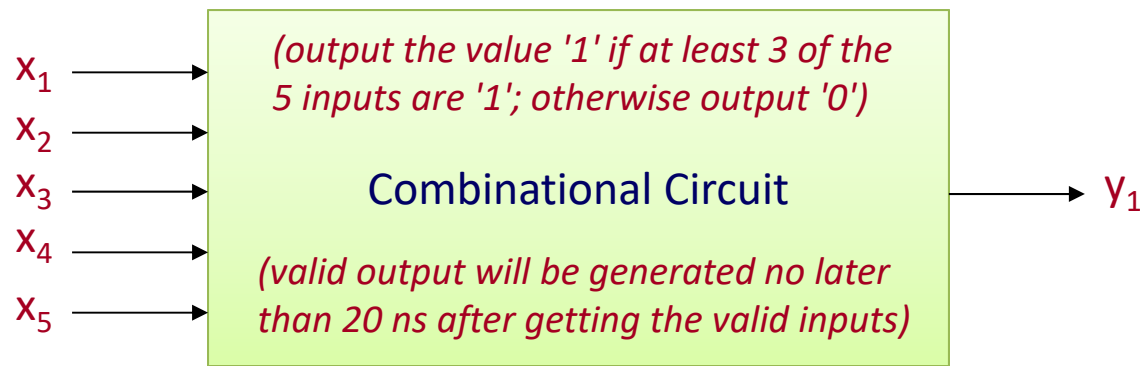
Arnaldo Oliveira, Augusto Silva, Iouliia Skliarova

Lecture 3 contents

- Combinational circuits
- Boolean algebra
 - axioms
 - theorems
 - duality
 - algebraic simplification of logic functions
 - canonical forms
- Standard representations of logic functions

Combinational Circuits

- A logic circuit whose outputs depend only on its current inputs is called a **combinational circuit**.
- A combinational circuit is characterized by
 - one or more inputs
 - one or more outputs
 - a functional specification describing each output as a function of the inputs
 - a time specification that includes at least the maximum time it will take the circuit to produce valid output values for an arbitrary set of input values -> **propagation delay**.



Boolean Algebra

- Formal analysis techniques for digital circuits have their roots in the work of an English mathematician, George Boole.
- In 1854, he invented a two-valued algebraic system, now called **Boolean algebra**.
- In 1938, Bell Labs researcher Claude E. Shannon showed how to adapt Boolean algebra to analyze and describe the behavior of circuits.
- In switching algebra we use a symbolic variable, such as x , to represent the condition of a logic signal.
- A logic signal is in one of two possible conditions: low or high, off or on, and so on, depending on the technology.
- We say that x has the value "0" for one of these conditions and "1" for the other.

Axioms

- The **axioms** (or **postulates**) of a mathematical system are a minimal set of basic definitions that we assume to be true, from which all other information about the system can be derived.
- A variable x can take on only one of two values:
 - $x = 0$ if $x \neq 1$
 - $x = 1$ if $x \neq 0$
- Inversion (definition of NOT):
 - if $x = 0$ then $\bar{x} = 1$
 - if $x = 1$ then $\bar{x} = 0$
- Definition of the AND and OR operations:
 - $0 \cdot 0 = 0$, $1 \cdot 1 = 1$, $0 \cdot 1 = 1 \cdot 0 = 0$
 - $1 + 1 = 1$, $0 + 0 = 0$ $1 + 0 = 0 + 1 = 1$
- These pairs of axioms completely define switching algebra.
- All other facts about the system can be proved using these axioms as a starting point.

Operator Precedence

- **Operator precedence** is an ordering of logical operators designed to allow the dropping of parentheses in logical expressions.
- The following list gives a hierarchy of precedences for the Boolean operators (from highest to lowest):
 - NOT
 - AND
 - OR

Example:

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

Single-Variable Theorems

- Switching algebra **theorems** are statements, known to be always true, that permit us to manipulate algebraic expressions to allow simpler analysis or more efficient synthesis of the corresponding circuits.
- Identities:
 - $x + 0 = x$
 - $x \cdot 1 = x$
- Null elements:
 - $x + 1 = 1$
 - $x \cdot 0 = 0$
- Idempotency:
 - $x + x = x$
 - $x \cdot x = x$
- Involution:
 - $\bar{\bar{x}} = x$
- Complements:
 - $x + \bar{x} = 1$
 - $x \cdot \bar{x} = 0$

Two- and Three-Variable Theorems

- Commutativity:

- $x + y = y + x$
- $x \cdot y = y \cdot x$

- Associativity:

- $(x + y) + z = x + (y + z)$
- $(x \cdot y) \cdot z = x \cdot (y \cdot z)$

- Distributivity:

- $x \cdot y + x \cdot z = x \cdot (y + z)$
- $(x + y) \cdot (x + z) = x + y \cdot z$

- Covering:

- $x + x \cdot y = x$
- $x \cdot (x + y) = x$

- Combining:

- $x \cdot y + x \cdot \bar{y} = x$
- $(x + y) \cdot (x + \bar{y}) = x$

- Simplification:

- $x + \bar{x} \cdot y = x + y$
- $x \cdot (\bar{x} + y) = x \cdot y$

- Consensus:

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



Resume of Theorems

- Most theorems in switching algebra are exceedingly simple to prove using a technique called **perfect induction**:
 - prove a theorem by proving that it is true for all possible values (“0” and “1”) of all the variables
- In all of the theorems, **it is possible to replace each variable with an arbitrary logic expression**:
 - $x + x \cdot (a + b \cdot c \cdot \bar{d} \cdot e) = x$
- When realizing the AND(OR) operation, we can connect gate inputs in any order: either one n -input gate or $(n - 1)$ 2-input gates interchangeably, though propagation delay and cost are likely to be higher with multiple 2-input gates:
 - $w \cdot x \cdot y \cdot z = (w \cdot x) \cdot (y \cdot z) = (w \cdot (x \cdot (y \cdot z))) \dots$
- In Boolean algebra, logical addition distributes over logical multiplication:
 - $(x + y) \cdot (x + z) = x + y \cdot z$

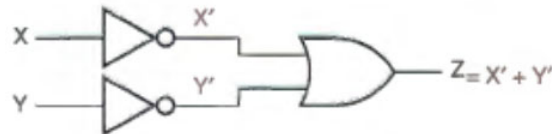
DeMorgan's Theorems

- An n -input AND gate whose output is complemented is equivalent to an n -input OR gate whose inputs are complemented:

- $\overline{x \cdot y} = \bar{x} + \bar{y}$



- $\overline{\prod_{i=0}^{n-1} x_i} = \sum_{i=0}^{n-1} \bar{x}_i$

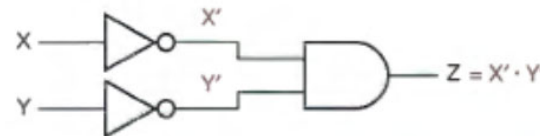


- An n -input OR gate whose output is complemented is equivalent to an n -input AND gate whose inputs are complemented:

- $\overline{x + y} = \bar{x} \cdot \bar{y}$



- $\overline{\sum_{i=0}^{n-1} x_i} = \prod_{i=0}^{n-1} \bar{x}_i$



Generalized DeMorgan's Theorem

- Given any n -variable fully parenthesized logic expression, its complement can be obtained by swapping $+$ and \cdot and complementing all variables:

$$\overline{F(x_0, x_1, \dots, x_{n-1}, +, \cdot)} = F(\overline{x_0}, \overline{x_1}, \dots, \overline{x_{n-1}}, \cdot, +)$$

Example:

$$F(w, x, y, z) = \bar{w} \cdot x + x \cdot y + w \cdot (\bar{x} + \bar{z})$$

$$\overline{F(w, x, y, z)} = (w + \bar{x}) \cdot (\bar{x} + \bar{y}) \cdot (\bar{w} + x \cdot z)$$

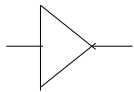


Principle of Duality

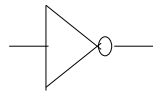
- The **principle of duality** states that any theorem or identity in switching algebra remains true if 0 and 1 are swapped and \cdot and $+$ are swapped throughout.
 - Duals of all the axioms are true.
 - Duals of all switching-algebra theorems are true.
- If $F(x_0, x_1, \dots, x_{n-1}, +, \cdot)$ is a fully parenthesized logic expression involving the variables x_0, x_1, \dots, x_{n-1} , and the operators $+$ and \cdot , then the dual of F , written F^D is the same expression with $+$ and \cdot swapped:
 - $F^D(x_0, x_1, \dots, x_{n-1}, +, \cdot) = F(x_0, x_1, \dots, x_{n-1}, \cdot, +)$
- The generalized DeMorgan's theorem may now be restated as follows:
 - $\overline{F(x_0, x_1, \dots, x_{n-1})} = F^D(\overline{x_0}, \overline{x_1}, \dots, \overline{x_{n-1}})$

NAND and NOR Gates

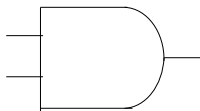
buffer



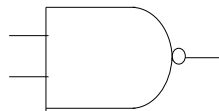
NOT



AND



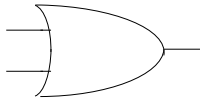
NAND



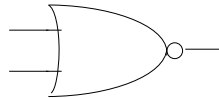
$$\overline{x \cdot y}$$

| x | y | x NAND y |
|---|---|-----------------|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

OR



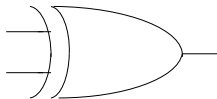
NOR



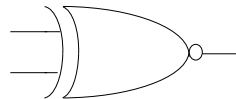
$$\overline{x + y}$$

| x | y | x NOR y |
|---|---|----------------|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

XOR



XNOR



Functional Completeness

- A **functionally complete set** of Boolean operators is one which can be used to describe the behavior of any digital circuit.
- Examples:
 - {AND, OR, NOT}
 - {AND, NOT}
 - {OR, NOT}
 - {NAND}
 - {NOR}

NAND and NOR Gates

- To write a Boolean expression only with the operators **NAND**, first put the expression in the **sum-of-products** form and then apply the **involution** theorem ($\bar{\bar{x}} = x$) followed by the **DeMorgan's** theorem ($\sum_{i=0}^{n-1} x_i = \prod_{i=0}^{n-1} \bar{x}_i$).
- To write a Boolean expression only with the operators **NOR**, first put the expression in the **product-of-sums** form and then apply the **involution** theorem ($\bar{\bar{x}} = x$) followed by the **DeMorgan's** theorem ($\prod_{i=0}^{n-1} x_i = \sum_{i=0}^{n-1} \bar{x}_i$).
- Always assume that complemented versions of input variables are available.

Examples:

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

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

Boolean Functions

- A Boolean function $f(x_0, x_1, \dots, x_{n-1})$ is a match that associates an element of the set $\{0,1\}$ with each of the 2^n possible combinations that variables can assume.
- There are $2^{m \times 2^n}$ different Boolean functions that can be implemented in a digital system with n inputs and m outputs.



Examples:

For $n=1, m=1$: $2^{1 \times 2^1} = 4$

| x | constant '0' | x | \bar{x} | constant '1' |
|---|--------------|---|-----------|--------------|
| 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |

For $n=4, m=3$: $2^{3 \times 2^4} = 2^{48} = 281\,474\,976\,710\,656$

Truth Table

- The most basic representation of a logic function is the **truth table**.
- A truth table simply lists the output of the circuit for every possible input combination.
- Traditionally, the input combinations are arranged in rows in ascending binary counting order, and the corresponding output values are written in a column next to the rows.
- The truth table for an n -variable logic function has 2^n rows.

Example ($n=3$):

| | x | y | z | f(x,y,z) |
|---|---|---|---|----------|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 0 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | 1 | 0 | 1 | 1 |
| 6 | 1 | 1 | 0 | 1 |
| 7 | 1 | 1 | 1 | 1 |

Minterms and Maxterms

- A **literal** is a variable or the complement of a variable. Examples: x, y, \bar{x} .
- A **product term** is a single literal or a logical product of two or more literals. Examples: $\bar{z}, x \cdot y, x \cdot \bar{y} \cdot z$.
- A **sum term** is a single literal or a logical sum of two or more literals. Examples: $\bar{z}, x + y, x + \bar{y} + z$.
- A **normal term** is a product or sum term in which no variable appears more than once.
- An n -variable **minterm** is a normal product term with n literals. There are 2^n such product terms.
 - A minterm m_i corresponds to row i of the truth table.
 - In minterm m_i , a particular variable appears complemented if the corresponding bit in the binary representation of i is 0; otherwise, it is uncomplemented.
- An n -variable **maxterm** is a normal sum term with n literals. There are 2^n such sum terms.
 - A maxterm M_i corresponds to row i of the truth table.
 - In maxterm M_i , a particular variable appears complemented if the corresponding bit in the binary representation of i is 1; otherwise, it is uncomplemented.

Example:

| | x | y | z | f(x,y,z) |
|---|---|---|---|----------|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 0 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | 1 | 0 | 1 | 1 |
| 6 | 1 | 1 | 0 | 1 |
| 7 | 1 | 1 | 1 | 1 |

$$m_0 = \bar{x} \cdot \bar{y} \cdot \bar{z}$$

$$M_0 = x + y + z$$

$$m_5 = x \cdot \bar{y} \cdot z$$

$$M_5 = \bar{x} + y + \bar{z}$$

$$m_i = \overline{M_i} \quad i = 0, 1, \dots, 2^n - 1$$

Algebraic Representations

- Any Boolean function can be presented as:
 - a sum of the minterms corresponding to truth-table rows (input combinations) for which the function produces a 1 -> **canonical sum**
 - a product of the maxterms corresponding to truth-table rows (input combinations) for which the function produces a 0 -> **canonical product**

Example:

| | x | y | z | f(x,y,z) |
|---|---|---|---|----------|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 0 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | 1 | 0 | 1 | 1 |
| 6 | 1 | 1 | 0 | 1 |
| 7 | 1 | 1 | 1 | 1 |

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

$$= \sum_{x,y,z} m(1,4,5,6,7)$$

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

$$= \prod_{x,y,z} M(0,2,3)$$

Shannon's Expansion Theorems

- Any Boolean function $f(x_0, x_1, \dots, x_{n-1})$ can be presented in the following forms:
 - $\overline{x_0} \cdot f(0, x_1, \dots, x_{n-1}) + x_0 \cdot f(1, x_1, \dots, x_{n-1})$
 - $(\overline{x_0} + f(1, x_1, \dots, x_{n-1})) \cdot (x_0 + f(0, x_1, \dots, x_{n-1}))$

Perfect induction:

If $x_0 = 0$ then: $f(0, x_1, \dots, x_{n-1}) = 1 \cdot f(0, x_1, \dots, x_{n-1}) + 0 \cdot f(1, x_1, \dots, x_{n-1})$

If $x_0 = 1$ then: $f(1, x_1, \dots, x_{n-1}) = 0 \cdot f(0, x_1, \dots, x_{n-1}) + 1 \cdot f(1, x_1, \dots, x_{n-1})$

Canonical Sum

- Extending to 2 variables:

$$\begin{aligned} f(x_0, x_1, \dots, x_{n-1}) &= \bar{x}_0 \cdot f(0, x_1, \dots, x_{n-1}) + x_0 \cdot f(1, x_1, \dots, x_{n-1}) = \\ &= \bar{x}_0 \cdot \bar{x}_1 \cdot f(0, 0, x_2, \dots, x_{n-1}) + \bar{x}_0 \cdot x_1 \cdot f(0, 1, x_2, \dots, x_{n-1}) + \\ &+ x_0 \cdot \bar{x}_1 \cdot f(1, 0, x_2, \dots, x_{n-1}) + x_0 \cdot x_1 \cdot f(1, 1, x_2, \dots, x_{n-1}) \end{aligned}$$

- Extending to n variables:

$$f(x_0, x_1, \dots, x_{n-1}) = \sum_{i=0}^{2^n-1} m_i \cdot f_i \quad f_i = f((x_0, x_1, \dots, x_{n-1}) = i)$$

Canonical Product

- Extend the Shannon expansion theorem $f(x_0, x_1, \dots, x_{n-1}) = (\overline{x_0} + f(1, x_1, \dots, x_{n-1})) \cdot (x_0 + f(0, x_1, \dots, x_{n-1}))$ to n variables:
 - $f(x_0, x_1, \dots, x_{n-1}) = \prod_{i=0}^{2^n-1} (f_i + M_i)$



3rd and 4th Canonical Forms

- 3rd canonical form:

$$f(x_0, x_1, \dots, x_{n-1}) = \overline{\overline{f(x_0, x_1, \dots, x_{n-1})}} = \overline{\sum_{i=0}^{2^n-1} f_i \cdot m_i} = \overline{\prod_{i=0}^{2^n-1} f_i \cdot m_i}$$

- 4th canonical form:

$$f(x_0, x_1, \dots, x_{n-1}) = \overline{\overline{f(x_0, x_1, \dots, x_{n-1})}} = \overline{\prod_{i=0}^{2^n-1} f_i + M_i} = \overline{\sum_{i=0}^{2^n-1} f_i + M_i}$$

Canonical Forms

canonical sum
of products:

$$f(x_0, x_1, \dots, x_{n-1}) = \sum_{i=0}^{2^n-1} m_i \cdot f_i$$

AND-OR

canonical product
of sums:

$$f(x_0, x_1, \dots, x_{n-1}) = \prod_{i=0}^{2^n-1} (f_i + M_i)$$

OR-AND

3rd canonical form:

$$f(x_0, x_1, \dots, x_{n-1}) = \overline{\prod_{i=0}^{2^n-1} f_i \cdot m_i}$$

NAND-NAND

4th canonical form :

$$f(x_0, x_1, \dots, x_{n-1}) = \overline{\sum_{i=0}^{2^n-1} f_i + M_i}$$

NOR-NOR

Canonical Forms (cont.)

Example: Derive the canonical forms of function $f(x,y,z)$:

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

| | x | y | z | $f(x,y,z)$ |
|---|---|---|---|------------|
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 1 |
| 3 | 0 | 1 | 1 | 0 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | 1 | 0 | 1 | 0 |
| 6 | 1 | 1 | 0 | 1 |
| 7 | 1 | 1 | 1 | 1 |

1st: $f(x, y, z) = \sum m(0,2,4,6,7)$

2nd: $f(x, y, z) = \prod M(1,3,5)$

3rd: $f(x, y, z) = \overline{\prod m(0,2,4,6,7)}$

4th: $f(x, y, z) = \overline{\sum M(1,3,5)}$

Standard Representations of Logic Functions

- A truth table
- Algebraic
- Logic circuit

An algebraic representation frequently includes redundant terms:

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

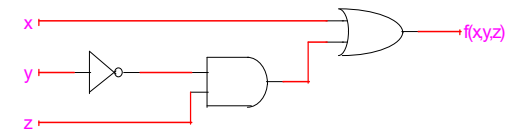
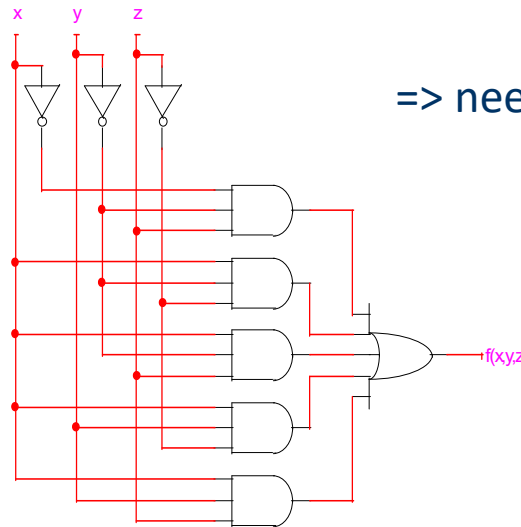
A truth table representation is unique:

| x | y | z | f(x,y,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 |

Logic circuit:

=> need for simplification

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



Exercises

- Are the following expressions correct?

$$[x + y \cdot z]^D = x \cdot y + z$$

$$[x + y \cdot z]^D = \bar{x} \cdot (\bar{y} + \bar{z})$$

- A self-dual logic function is a function f such that $f = f^D$. Which of the following functions are self-dual?

$$f_1(x, y, z) = \bar{x} \cdot y + \bar{x} \cdot z + y \cdot z$$

$$f_2(x, y) = \bar{x} \cdot y + x \cdot \bar{y}$$

Exercises (cont.)

- Express the function y in the simplest form using only the operator NAND.

$$y = x_1 \cdot (x_2 + \bar{x}_3 \cdot x_4) + x_2$$

$$y = x_1 \cdot x_2 + x_1 \cdot \bar{x}_3 \cdot x_4 + x_2 = x_2 + x_1 \cdot \bar{x}_3 \cdot x_4$$

$$y = \overline{\overline{x_2 + x_1 \cdot \bar{x}_3 \cdot x_4}} = \overline{\bar{x}_2 \cdot x_1 \cdot \bar{x}_3 \cdot x_4}$$

- Express the function y in the simplest form using only the operator NOR.

$$y = x_1 \cdot (x_2 + \bar{x}_3 \cdot x_4) + x_2$$

$$y = (x_1 + x_2) \cdot (x_2 + \bar{x}_3 \cdot x_4 + x_2) = (x_1 + x_2) \cdot (x_2 + \bar{x}_3 \cdot x_4)$$

$$y = (x_1 + x_2) \cdot (x_2 + \bar{x}_3) \cdot (x_2 + x_4)$$

$$y = \overline{\overline{(x_1 + x_2) \cdot (x_2 + \bar{x}_3) \cdot (x_2 + x_4)}} = \overline{\overline{(x_1 + x_2)} + \overline{\overline{(x_2 + \bar{x}_3)}} + \overline{\overline{(x_2 + x_4)}}}$$

Exercises (cont.)

- Determine all the canonical forms of the function f :

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

$$f(x, y, z) = \sum m(0, 2, 3, 6, 7) = \bar{x} \cdot \bar{y} \cdot \bar{z} + \bar{x} \cdot y \cdot \bar{z} + \bar{x} \cdot y \cdot z + x \cdot y \cdot \bar{z} + x \cdot y \cdot z$$

$$f(x, y, z) = \prod M(1, 4, 5) = (x + y + \bar{z}) \cdot (\bar{x} + y + z) \cdot (\bar{x} + y + \bar{z})$$

$$f(x, y, z) = \overline{\prod m(0, 2, 3, 6, 7)} = \overline{(\bar{x} \cdot \bar{y} \cdot \bar{z}) \cdot (\bar{x} \cdot y \cdot \bar{z}) \cdot (\bar{x} \cdot y \cdot z) \cdot (x \cdot y \cdot \bar{z}) \cdot (x \cdot y \cdot z)}$$

$$f(x, y, z) = \overline{\sum M(1, 4, 5)} = \overline{(x + y + \bar{z}) + (\bar{x} + y + z) + (\bar{x} + y + \bar{z})}$$

- Minimize this function.
- Minimize the following functions:

$$f(a, b, c) = \bar{a} \cdot b + \bar{a} \cdot \bar{c} + a \cdot c + a \cdot \bar{b} + b + c$$

$$f(a, b, c) = \bar{a} \cdot \bar{b} \cdot \bar{c} + \bar{a} \cdot b \cdot c + a \cdot b \cdot \bar{c} + a \cdot \bar{b} \cdot c$$