IF120 Discrete Mathematics

14 Combinatorial Circuits & Boolean Algebra

Angga Aditya Permana, Januar Wahjudi, Yaya Suryana, Meriske, Muhammad Fahrury Romdendine

REVIEW

- Tree Traversals
- Decision Trees
- Trees Isomorphism
- Game Trees

OUTLINE

- Combinatorial Circuits
- Boolean Algebra

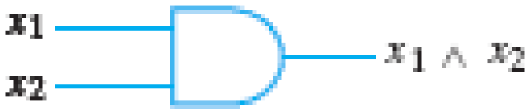
Combinatorial Circuits

- In a digital computer, there are only two possibilities, written **0** and **1**, for the smallest, indivisible object.
- All programs and data are ultimately reducible to combinations of **bits**.
- A variety of devices have been used throughout the years in digital computers to store bits.
- **Electronic circuits** allow these storage devices to **communicate** with each other.
- A bit in one part of a circuit is transmitted to another part of the circuit as a **voltage**.
- Thus two voltage levels are needed—for example, a high voltage can communicate 1 and a low voltage can communicate 0.
- The output of a combinatorial circuit is uniquely defined for every combination of inputs.
- A combinatorial circuit has **no memory**; previous inputs and the state of the system do not affect the output of a combinatorial circuit.
- Combinatorial circuits can be constructed using solid-state devices, called **gates,** which are capable of switching voltage levels (bits).

Combinatorial Circuits

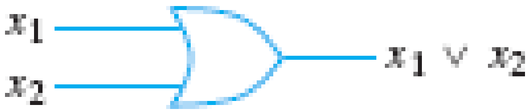
- **Definition 14.1**: An **AND** *gate* receives inputs x_1 and x_2, where x_1 and x_2 are bits, and produces output denoted $x_1 \wedge x_2$, where

$$x_1 \wedge x_2 = \begin{cases} 1 & if\ x_1 = 1\ and\ x_2 = 1 \\ 0 & otherwise. \end{cases}$$



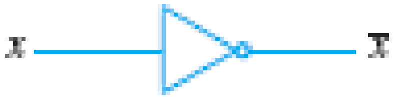
- **Definition 14.2**: An **OR** *gate* receives inputs x_1 and x_2, where x_1 and x_2 are bits, and produces output denoted $x_1 \vee x_2$, where

$$x_1 \vee x_2 = \begin{cases} 1 & if\ x_1 = 1\ or\ x_2 = 1 \\ 0 & otherwise. \end{cases}$$



- **Definition 14.3**: A **NOT** *gate* receives inputs x, where x is a bit, and produces output denoted $\bar{x}$, where

$$\bar{x} = \begin{cases} 1 & if\ x = 0 \\ 0 & if\ x = 1. \end{cases}$$



Combinatorial Circuits

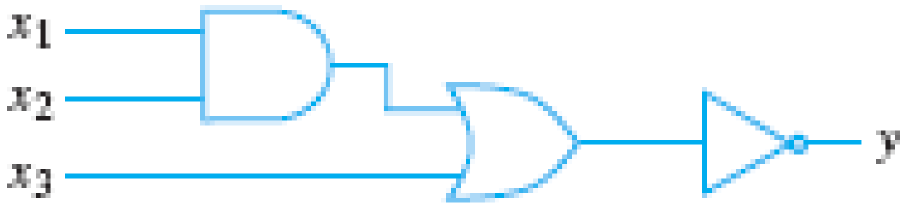
- The **logic table** of a combinatorial circuit lists all possible inputs together with the resulting outputs.
- Following are the logic tables for the basic AND, OR, and NOT circuits.
- We note that performing the operation AND (OR) is the same as taking the minimum (maximum) of the two bits x_1 and x_2.

x_1	x_2	$x_1 \wedge x_2$
1	1	1
1	0	0
0	1	0
0	0	0

x_1	x_2	$x_1 \vee x_2$
1	1	1
1	0	1
0	1	1
0	0	0

x	$\overline{x}$
1	0
0	1

The circuit of Figure below is an example of a combinatorial circuit since the output y is uniquely defined for each combination of inputs x_1, x_2, and x_3.

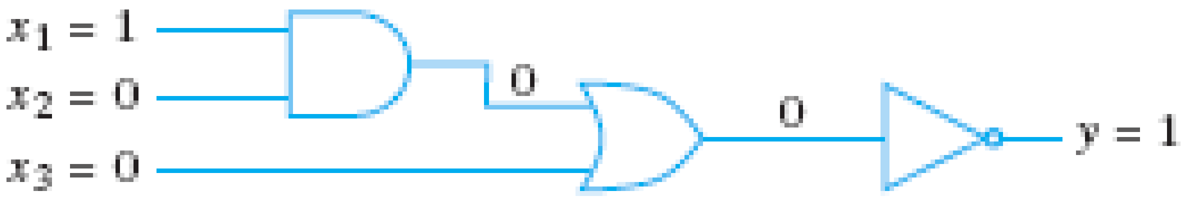


x_1	x_2	x_3	y
1	1	1	0
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	1
0	0	1	0
0	0	0	1

The logic table for this combinatorial circuit follows.

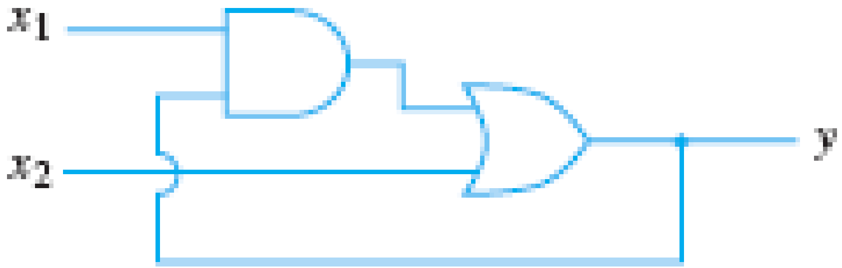
Notice that all possible combinations of values for the inputs x_1, x_2, and x_3 are listed. For a given set of inputs, we can compute the value of the output *y* by tracing the flow through the circuit. For example, the fourth line of the table gives the value of the output *y* for the input values

$$x_1 = 1, x_2 = 0, x_3 = 0$$

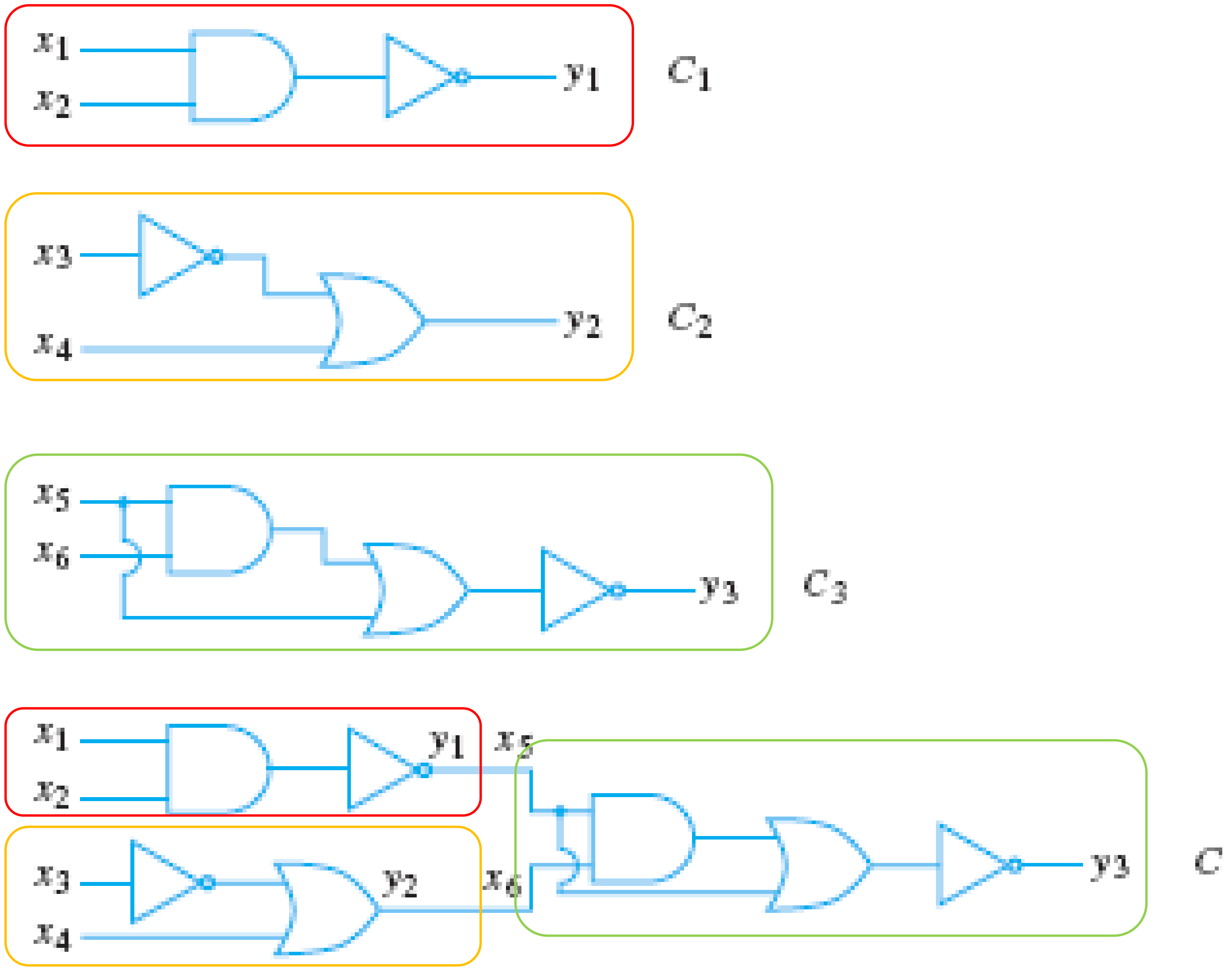


Combinatorial Circuits

- The circuit of Figure below is not a combinatorial circuit, because the output y is not uniquely defined for each combination of inputs x_1 and x_2. For example, suppose that $x_1 = 1$ and $x_2 = 0$. If the output of the AND gate is 0, then $y = 0$. On the other hand, if the output of the AND gate is 1, then $y = 1$. Such a circuit might be used to store one bit.



❑ Individual combinatorial circuits may be interconnected. The combinatorial circuits C_1, C_2, and C_3 of Figure below may be combined, as shown, to obtain the combinatorial circuit C.

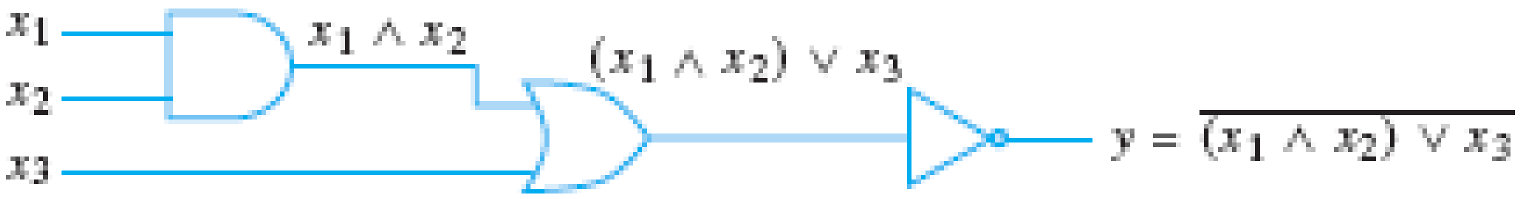


Combinatorial Circuits

- A combinatorial circuit with one output, such as that in Figure below, can be represented by an expression using the symbols $\wedge$, $\vee$, and $\overline{}$. We follow the flow of the circuit symbolically. First, x_1 and x_2 are ANDed, which produces output $x_1 \wedge x_2$. This output is then ORed with x_3 to produce output $(x_1 \wedge x_2) \vee x_3$. This output is then NOTed. Thus the output y may be

$$y = \overline{(x_1 \wedge x_2) \vee x_3}$$

  Expressions such as shown above are called **Boolean expressions.**



Boolean Expressions

- **Definition 14.4**: **Boolean expressions** in the symbols $x_1, \ldots, x_n$ are defined recursively as follows.

$$0, 1, x_1, \ldots, x_n$$

are Boolean expressions. If X_1 and X_2 are Boolean expressions, then

$$\text{(a) } (X_1), \quad \text{(b) } \overline{X_1}, \quad \text{(c) } X_1 \vee X_2, \quad \text{(d) } X_1 \wedge X_2$$

are Boolean expressions.

- If X is a Boolean expression in the symbols $x_1, \ldots, x_n$, we sometimes write

$$X = X(x_1, \ldots, x_n)$$

- Either symbol x or $\bar{x}$ is called a *literal*.

Boolean Expressions

- If $X = X(x_1, \dots, x_n)$ is a Boolean expression and $x_1, \dots, x_n$ are assigned values $a_1, \dots, a_n$ in $\{0,1\}$, we may use Definitions 14.1–14.3 to compute a value for X. We denote this value $X(a_1, \dots, a_n)$ or $X(x_i = a_i)$.

❑For $x_1 = 1, x_2 = 0$, and $x_3 = 0$, the Boolean expression $X(x_1, x_2, x_3) = \overline{(x_1 \wedge x_2) \vee x_3}$ becomes

$$X(1,0,0) = \overline{(1 \wedge 0) \vee 0} = \overline{0 \vee 0} = \bar{0} = 1.$$

We have again computed the fourth row of the table in the first Example of this section.

- In a Boolean expression in which parentheses are not used to specify the order of operations, we assume that ∧ is evaluated before ∨.

❑For $x_1 = 0, x_2 = 0$, and $x_3 = 1$, the value of the Boolean expression $x_1 \wedge x_2 \vee x_3$ is

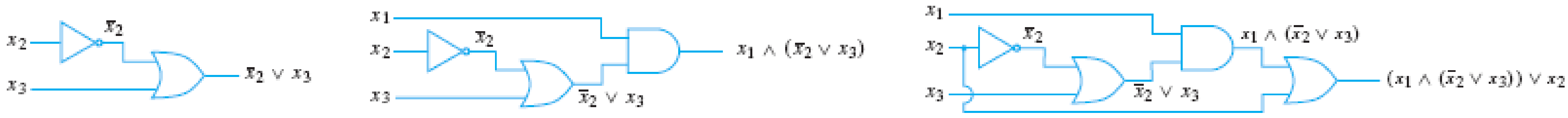
$$x_1 \wedge x_2 \vee x_3 = 0 \wedge 0 \vee 1 = 0 \vee 1 = 1.$$

- Find the combinatorial circuit corresponding to the Boolean expression

$$\left(x_1 \wedge (\overline{x_2} \vee x_3)\right) \vee x_2$$

and write the logic table for the circuit obtained.

We begin with the expression $\overline{x_2} \vee x_3$ in the innermost parentheses. This expression is converted to a combinatorial circuit, as shown in Figure below (left). The output of this circuit is ANDed with x_1 to produce the circuit drawn in Figure below (middle). Finally, the output of this circuit is ORed with x_2 to give the desired circuit drawn in Figure below (right).



The logic table follows

x_1	x_2	x_3	$(x_1 \wedge (\overline{x}_2 \vee x_3)) \vee x_2$
1	1	1	1
1	1	0	1
1	0	1	1
1	0	0	1
0	1	1	1
0	1	0	1
0	0	1	0
0	0	0	0

Properties of Combinatorial Circuits

- **Theorem 14.1**: *If* $\wedge, \vee,$ *and* $\overline{\ }$ *are as in Definitions 14.1–14.3, then the following properties hold for all* $a, b, c \in Z_2$.

a) *Associative laws:*

$$(a \vee b) \vee c = a \vee (b \vee c)$$
$$(a \wedge b) \wedge c = a \wedge (b \wedge c)$$

b) *Commutative laws:*

$$a \vee b = b \vee a, \qquad a \wedge b = b \wedge a$$

c) *Distributive laws:*

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$$
$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$$

d) *Identity laws:*

$$a \vee 0 = a, \qquad a \wedge 1 = a$$

e) *Complement laws:*

$$a \vee \bar{a} = 1, \qquad a \wedge \bar{a} = 0$$

Properties of Combinatorial Circuits

- By using Theorem 14.1, show that the combinatorial circuits of Figure below have identical outputs for given identical inputs.

The Boolean expressions representing the circuits are, respectively,

$$x_1 \vee (x_2 \wedge x_3), \qquad (x_1 \vee x_2) \wedge (x_1 \vee x_3).$$

By Theorem 14.1(c),

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c) \; for\, all\; a, b, c \in Z_2.$$

But the above expression says that the combinatorial circuits of the figure have identical outputs for given identical inputs.

Properties of Combinatorial Circuits

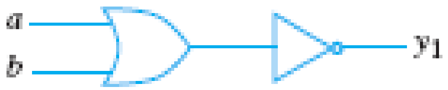
- **Definition 14.5**: Let $X_1 = X_1(x_1, \dots, x_n)$ and $X_2 = X_2(x_1, \dots, x_n)$ be Boolean expressions. We define X_1 to be **equal** to X_2 and write
$$X_1 = X_2$$
if $X_1(a_1, \dots, a_n) = X_2(a_1, \dots, a_n)$ for all $a_i \in Z_2$.

- Show that
$$(\overline{x \vee y}) = \bar{x} \wedge \bar{y}.$$
According to Definition 14.5, the expression holds if the equation is true for all choices of *x* and *y* in Z_2. Thus we may simply construct a table listing all possibilities to verify it.

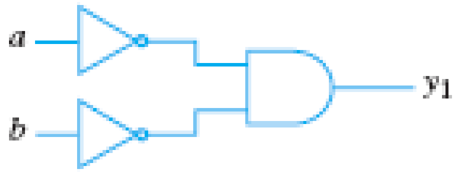
x	y	$(\overline{x \vee y})$	$\bar{x} \wedge \bar{y}$
1	1	0	0
1	0	0	0
0	1	0	0
0	0	1	1

- **<u>Definition 14.6</u>**: We say that two combinatorial circuits, each having inputs $x_1, \dots, x_n$ and a single output, are **equivalent** if, whenever the circuits receive the same inputs, they produce the same outputs.

❑ The combinatorial circuits of Figures below are equivalent since, as shown, they have identical logic tables.



a	b	y_1
1	1	0
1	0	0
0	1	0
0	0	1



a	b	y_1
1	1	0
1	0	0
0	1	0
0	0	1

- **<u>Theorem 14.2</u>**: Let C_1 and C_2 be combinatorial circuits represented, respectively, by the Boolean expressions $X_1 = X_1(x_1, \dots, x_n)$ and $X_2 = X_2(x_1, \dots, x_n)$. Then C_1 and C_2 are equivalent if and only if $X_1 = X_2$.

Boolean Algebra

- **Definition 14.7**: A **Boolean algebra** B consists of a set S containing distinct elements 0 and 1, binary operators $+$ and $\cdot$ on S, and a unary operator $'$ on S satisfying the following laws.

a) Associative laws:

$$(x + y) + z = x + (y + z)$$
$$(x \cdot y) \cdot z = x \cdot (y \cdot z) \quad for\, all\, x, y, z \in S.$$

b) Commutative laws:

$$x + y = y + x, \qquad x \cdot y = y \cdot x \quad for\, all\, x, y \in S.$$

c) Distributive laws:

$$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$$
$$x + (y \cdot z) = (x + y) \cdot (x + z) \quad for\, all\, x, y, z \in S.$$

d) Identity laws:

$$x + 0 = x, \qquad x \cdot 1 = x \quad for\, all\, x \in S.$$

e) Complement laws:

$$x + x' = 1, \qquad x \cdot x' = 0 \quad for\, all\, x \in S.$$

- If B is a Boolean algebra, we write $B = (S, +, \cdot, ', 0, 1)$.

Boolean Algebra

- By Theorem 14.1, $(Z_2, \vee, \wedge, \overline{\ \ }, 0, 1)$ is a Boolean algebra. (We are letting Z_2 denote the set {0, 1}.) The operators $+, \cdot, '$ in Definition 14.7 are $\vee, \wedge, \overline{\ \ }$ respectively.
  - As is the standard custom, we will usually abbreviate $a \cdot b$ as ab. We also assume that $\cdot$ is evaluated before $+$. This allows us to eliminate some parentheses. For example, we can write $(xy) + z$ more simply as $xy + z$.
  - **Theorem 14.3**: In a Boolean algebra, the element x' of Definition 14.7(e) is unique. Specifically, if $x + y = 1$ and $x\,y = 0$, then $y = x'$.
  - In a Boolean algebra, we call the element x' the **complement** of x.
  - We can now derive several additional properties of Boolean algebras.

- **Theorem 14.4**: Let $B = (S, +, \cdot, ', 0, 1)$ be a Boolean algebra. The following properties hold.

a) Idempotent laws:

$$x + x = x, \qquad xx = x \qquad for\ all\ x \in S.$$

b) Bound laws:

$$x + 1 = 1, \qquad x0 = 0 \qquad for\ all\ x \in S.$$

c) Absorption laws:

$$x + xy = x, \qquad x(x + y) = x \qquad for\ all\ x, y \in S.$$

d) Involution law:

$$(x')' = x \qquad for\ all\ x \in S.$$

e) 0 and 1 laws:

$$0' = 1, \qquad 1' = 0.$$

f) De Morgan's laws for Boolean algebras:

$$(x + y)' = x'\,y', \qquad (xy)' = x' + y' \qquad for\ all\ x, y \in S.$$

- **<u>Definition 14.8</u>**: The **dual** of a statement involving Boolean expressions is obtained by replacing 0 by 1, 1 by 0, + by ·, and · by +.

❑ The dual of

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

is

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

- **<u>Theorem 14.5:</u>** *The dual of a theorem about Boolean algebras is also a theorem.*

❑ The dual of

$$x + x = x$$

is

$$xx = x$$

We proved the first statement as shown on the left. If we write the dual of each statement in the proof of first statement, we obtain the following proof for second statement (right).

$$\begin{aligned} x &= x + 0 \\ &= x + (xx') \\ &= (x + x)(x + x') \\ &= (x + x)1 \\ &= x + x \end{aligned}$$

$$\begin{aligned} x &= x\,1 \\ &= x\,(x + x') \\ &= x\,x + x\,x' \\ &= x\,x + 0 \\ &= x\,x \end{aligned}$$

Boolean Functions

- **Definition 14.9**: The **exclusive-OR** of x_1 and x_2 written $x_1 \oplus x_2$ is defined by Table below.

x_1	x_2	$x_1 \oplus x_2$
1	1	0
1	0	1
0	1	1
0	0	0

- A logic table, with one output, is a function.
- The domain is the set of inputs and the range is the set of outputs.
- For the exclusive-OR function above, the domain is the set

$$\{(1,1),(1,0),(0,1),(0,0)\}$$

and the range is the set

$$Z_2 = \{0,1\}.$$

Boolean Functions

- **Definition 14.10**: Let $X(x_1, \ldots, x_n)$ be a Boolean expression. A function f of the form
$$f\,(x_1, \ldots, x_n) \;=\; X(x_1, \ldots, x_n)$$
is called a **Boolean function**.

❑ The function $f\colon\, Z_2^3 \;\rightarrow Z_2$ defined by
$$f\,(x_1, x_2, x_3) \;= x_1 \wedge\, (\overline{x_2} \vee x_3)$$
is a Boolean function. The inputs and outputs are given in the following table.

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
1	1	1	1
1	1	0	0
1	0	1	1
1	0	0	1
0	1	1	0
0	1	0	0
0	0	1	0
0	0	0	0

❑Show that the function f given by the following table is a Boolean function.

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	1
0	0	1	0
0	0	0	0

Consider the first row of the table and the combination

$$x_1 \wedge x_2 \wedge x_3 \qquad \textbf{(14.1)}$$

Notice that if $x_1 = x_2 = x_3 = 1$, as indicated in the first row of the table, then (14.1) is 1. The values of x_i given by any other row of the table give (14.1) the value 0. Similarly, for the fourth row of the table we may construct the combination

$$x_1 \wedge \overline{x_2} \wedge \overline{x_3} \qquad \textbf{(14.2)}$$

Expression (14.2) has the value 1 for the values of x_i given by the fourth row of the table, whereas the values of x_i given by any other row of the table give (14.2) the value 0.

The procedure is clear. We consider a row *R* of the table where the output is 1. We then form the combination $x_1 \wedge x_2 \wedge x_3$ and place a bar over each x_i whose value is 0 in row *R*. The combination formed is 1 if and only if the x_i have the values given in row *R*. Thus, for row 6, we obtain the combination

$$\overline{x_1} \wedge x_2 \wedge \overline{x_3} \qquad \textbf{(14.3)}$$

Next, we OR the terms (14.1)–(14.3) to obtain the Boolean expression

$$(x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x_2} \wedge \overline{x_3}) \vee (\overline{x_1} \wedge x_2 \wedge \overline{x_3}) \qquad \textbf{(14.4)}$$

We claim that $f(x_1, x_2, x_3)$ and (14.4) are equal. To verify this, first suppose that $x_1, x_2,$ and x_3 have values given by a row of the table for which $f(x_1, x_2, x_3) = 1$. Then one of (14.1)–(14.3) is 1, so the value of (14.4) is 1. On the other hand, if $x_1, x_2,$ and x_3 have values given by a row of the table for which $f(x_1, x_2, x_3) = 0$, all of (14.1)–(14.3) are 0, so the value of (14.4) is 0. Thus f and the Boolean expression (14.4) agree on Z_2^3; therefore,

$$f(x_1, x_2, x_3) = (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x_2} \wedge \overline{x_3}) \vee (\overline{x_1} \wedge x_2 \wedge \overline{x_3})$$

as claimed.

- <u>**Definition 14.11**</u>: A **minterm** in the symbols $x_1, \dots, x_n$ is a Boolean expression of the form

$$y_1 \wedge y_2 \wedge \cdots \wedge y_n,$$

where each y_i is either x_i or $\overline{x_i}$.

- **Theorem 14.6**: *If* $f: Z_2^n \to Z_2$*, then* f *is a Boolean function. If* f *is not identically zero, let* $A_1, \dots, A_k$ *denote the elements* A_i *of* Z_2^n *for which* $f(A_i) = 1$*. For each* $A_i = (a_1, \dots, a_n)$*, set*

$$m_i = y_1 \wedge \cdots \wedge y_n,$$

  where

$$y_j = \begin{cases} x_j & if \;\; a_j = 1 \\ \overline{x_j} & if \;\; a_j = 0 \end{cases}$$

  Then

$$f\,(x_1, \dots, x_n) = m_1 \vee m_2 \vee \cdots \vee m_k. \qquad \textbf{(14.5)}$$

- **Definition 14.12**: The representation (14.5) of a Boolean function $f: Z_2^n \to Z_2$ is called the **disjunctive normal form** of the function f.
- Theorem 14.6 has a dual. In this case the function f is expressed as

$$f\,(x_1, \dots, x_n) = M_1 \wedge M_2 \wedge \cdots \wedge M_k. \qquad \textbf{(14.6)}$$

  Each M_i is of the form

$$y_1 \vee \cdots \vee y_n, \qquad \textbf{(14.7)}$$

  where y_j is either x_j or $\overline{x_j}$. A term of the form (14.7) is called a **maxterm** and the representation of f (14.6) is called the **conjunctive normal form.**

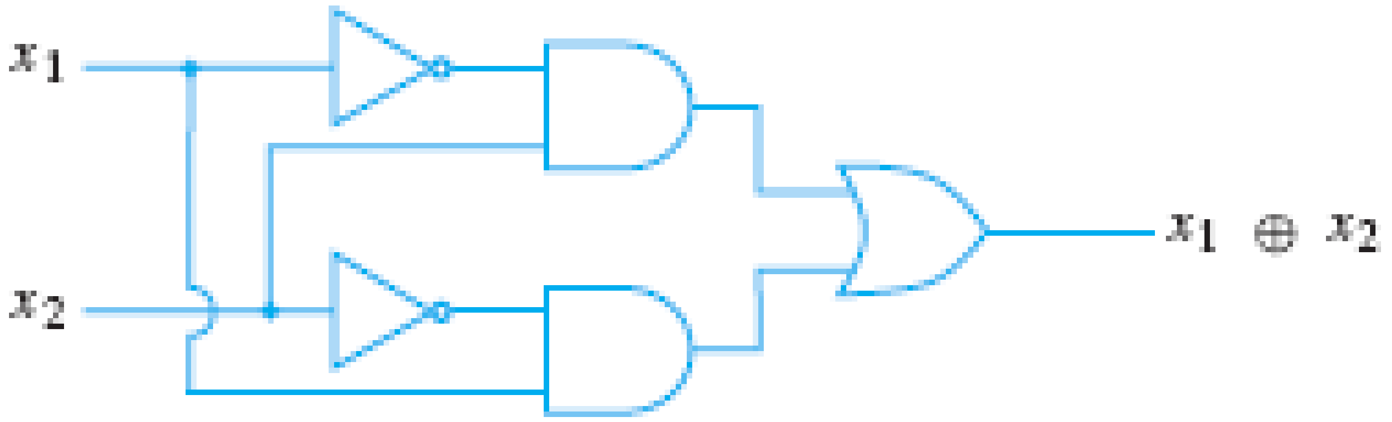
Boolean Functions

- Design a combinatorial circuit that computes the exclusive-OR of x_1 and x_2.
  The logic table for the exclusive-OR function $x_1 \oplus x_2$ is given in Table below.
  The disjunctive normal form of this function is

$$x_1 \oplus x_2 = (x_1 \wedge \overline{x_2}) \vee (\overline{x_1} \wedge x_2) \qquad \textbf{(14.8)}$$

  The combinatorial circuit corresponding to (14.8) is given in Figure below.

x_1	x_2	$x_1 \oplus x_2$
1	1	0
1	0	1
0	1	1
0	0	0



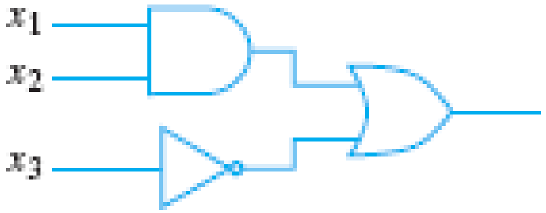
PRACTICE

PRACTICE 1

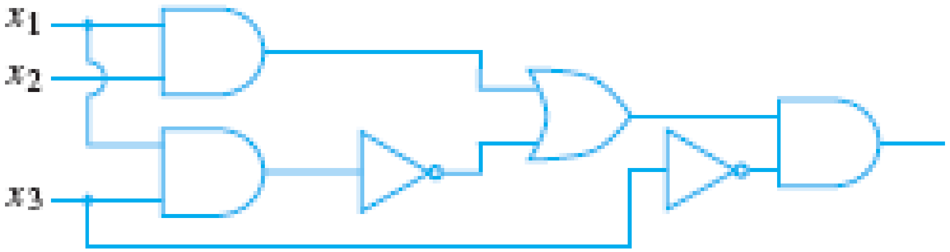
- In Exercises below, write the Boolean expression that represents the combinatorial circuit, write the logic table, and write the output of each gate symbolically!

1.

2.



3.



PRACTICE 2

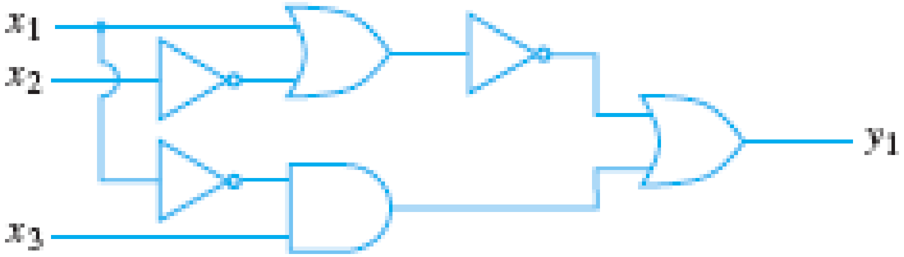
- Find the combinatorial circuit corresponding to each Boolean expression in Exercises below and write the logic table.

1. $x_1 \vee (\overline{x_2} \wedge x_3)$
2. $(x_1 \wedge \overline{x_2}) \vee (x_1 \vee \overline{x_3})$
3. $\left(x_1 \wedge \left(x_2 \vee (x_1 \wedge \overline{x_2})\right)\right) \vee \left((x_1 \wedge \overline{x_2}) \vee \overline{(x_1 \wedge \overline{x_3})}\right)$

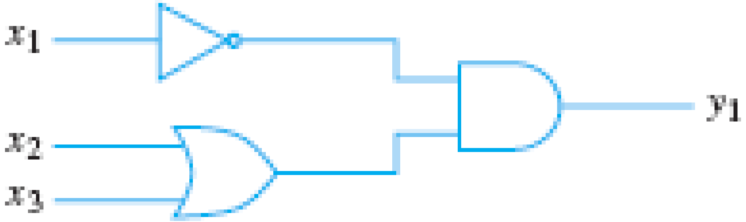
PRACTICE 3

- Show that the combinatorial circuits of Exercises below are equivalent.

1.

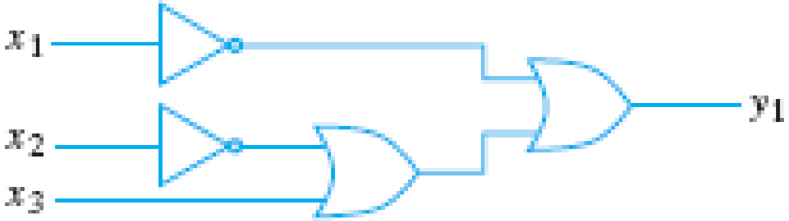


(a)

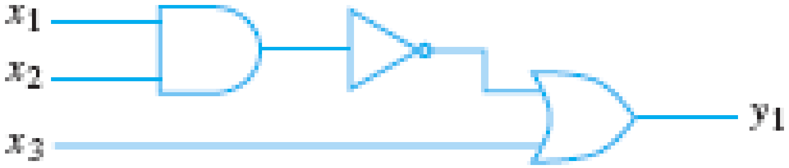


(b)

2.



(a)



(b)

PRACTICE 4

- Write the dual of each statement in Exercises below.

1. $(x + y)(x + 1) = x + xy + y$
2. $(x' + y')' = xy$
3. $If\ x + y = x + z\ and\ x' + y = x' + z, then\ y = z.$

PRACTICE 5

- In Exercises below, find the disjunctive normal form of each function and draw the combinatorial circuit corresponding to the disjunctive normal form.

1.

x	y	z	$f(x, y, z)$
1	1	1	1
1	1	0	1
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	0
0	0	1	1
0	0	0	1

2.

x	y	z	$f(x, y, z)$
1	1	1	0
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	1
0	1	0	1
0	0	1	1
0	0	0	0

REFERENCES

- Johnsonbaugh, R., 2005, *Discrete Mathematics*, New Jersey: Pearson Education, Inc.
- Rosen, Kenneth H., 2005, *Discrete Mathematics and Its Applications*, 6^{th} edition, McGraw-Hill.
- Hansun, S., 2021, *Matematika Diskret Teknik*, Deepublish.
- Lipschutz, Seymour, Lipson, Marc Lars, *Schaum's Outline of Theory and Problems of Discrete Mathematics*, McGraw-Hill.
- Liu, C.L., 1995, *Dasar-Dasar Matematika Diskret*, Jakarta: Gramedia Pustaka Utama.
- Other offline and online resources.

Visi

Menjadi Program Studi Strata Satu Informatika **unggulan** yang menghasilkan lulusan **berwawasan internasional** yang **kompeten** di bidang Ilmu Komputer (*Computer Science*), **berjiwa wirausaha** dan **berbudi pekerti luhur**.

INFORMATIKA UMN UNIVERSITAS MULTIMEDIA NUSANTARA

Misi

1. Menyelenggarakan pembelajaran dengan teknologi dan kurikulum terbaik serta didukung tenaga pengajar profesional.
2. Melaksanakan kegiatan penelitian di bidang Informatika untuk memajukan ilmu dan teknologi Informatika.
3. Melaksanakan kegiatan pengabdian kepada masyarakat berbasis ilmu dan teknologi Informatika dalam rangka mengamalkan ilmu dan teknologi Informatika.