

COMBINATIONAL LOGIC CIRCUITS

4.1 INTRODUCTION

The digital system consists of two types of circuits, namely:

- (i) Combinational circuits and
- (ii) Sequential circuits

A combinational circuit consists of logic gates, where outputs are at any instant and are determined only by the present combination of inputs without regard to previous inputs or previous state of outputs. A combinational circuit performs a specific information-processing operation assigned logically by a set of Boolean functions. Sequential circuits contain logic gates as well as memory cells. Their outputs depend on the present inputs and also on the states of memory elements. Since the outputs of sequential circuits depend not only on the present inputs but also on past inputs, the circuit behavior must be specified by a time sequence of inputs and memory states.

A combinational circuit consists of input variables, logic gates, and output variables. The logic gates accept signals from inputs and output signals are generated according to the logic circuits employed in it. Binary information from the given data transforms to desired output data in this process. Both input and output are obviously the binary signals, *i.e.*, both the input and output signals are of two possible states, logic 1 and logic 0. Figure 4.1 shows a block diagram of a combinational logic circuit. There are n number of input variables coming from an electric source and m number of output signals go to an external destination. The source and/or destination may consist of memory elements or sequential logic circuit or shift registers, located either in the vicinity of the combinational logic circuit or in a remote external location. But the external circuit does not interfere in the behavior of the combinational circuit.

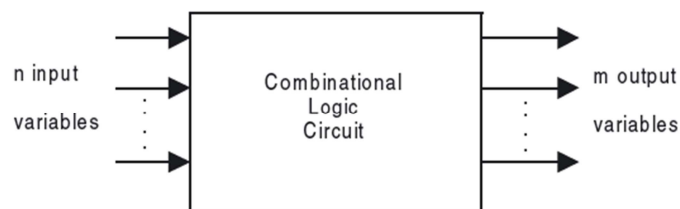


Figure (4-1)

4.2 DESIGN PROCEDURE

Any combinational circuit can be designed by the following steps of design procedure.

1. The problem is stated.
2. Identify the input variables and output functions.
3. The input and output variables are assigned letter symbols.
4. The truth table is prepared that completely defines the relationship between the input variables and output functions.
5. The simplified Boolean expression is obtained by any method of minimization: algebraic method, Karnaugh map method, or tabulation method.
6. A logic diagram is realized from the simplified expression using logic gates.

4.3 ADDERS

Various information-processing jobs are carried out by digital computers. Arithmetic operations are among the basic functions of a digital computer. Addition of two binary digits is the most basic arithmetic operation. The simple addition consists of four possible elementary operations, which are $0+0 = 0$, $0+1 = 1$, $1+0 = 1$, and $1+1 = 10$. The first three operations produce a sum of one digit, but the fourth operation produces a sum consisting of two digits. The higher significant bit of this result is called the *carry*. A combinational circuit that performs the addition of two bits as described above is called a *half-adder*. When the augend and addend numbers contain more significant digits, the carry obtained from the addition of two bits is added to the next higher-order pair of significant bits. Here the addition operation involves three bits—the augend bit, addend bit, and the carry bit and produces a sum result as well as carry. The combinational circuit performing this type of addition operation is called a *full-adder*. In circuit development two half-adders can be employed to form a full-adder.

4.3.1 Design of Half-adders

As described above, a half-adder has two inputs and two outputs. Let the input variables augend and addend be designated as A and B, and output functions be designated as S for sum and C for carry. The truth table for the functions is below.

<i>Input variables</i>		<i>Output variables</i>	
<i>A</i>	<i>B</i>	<i>S</i>	<i>C</i>
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Figure (4-2)

From the truth table in Figure 4.2, it can be seen that the outputs S and C functions are similar to Exclusive-OR and AND functions respectively, as shown in Figure 4.3 in Chapter 3. The Boolean expressions are

$$S = A'B + AB'$$

$$C = AB.$$

Figure 4.3 shows the logic diagram to implement the half-adder circuit.

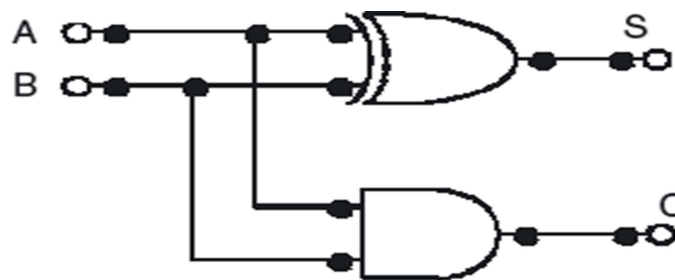


Figure (4-3)

4.3.2 Design of Full-adders

A combinational circuit of full-adder performs the operation of addition of three bits—the augend, addend, and previous carry, and produces the outputs sum and carry. Let us designate the input variables augend as A, addend as B, and previous carry as X, and outputs sum as S and carry as C. As there are three input variables, eight different input combinations are possible. The truth table is shown in Figure 4.4 according to its functions.

Input variables			Outputs	
X	A	B	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figure (4-4)

To derive the simplified Boolean expression from the truth table, the Karnaugh map method is adopted as in Figures 4.5(a)-(b).

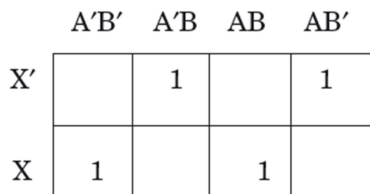


Figure 4.5(a) Map for function S.

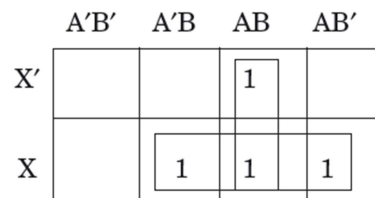


Figure 4.5(b) Map for function C.

The simplified Boolean expressions of the outputs are

$$S = X'A'B + X'AB' + XA'B' + XAB \quad \text{and}$$

$$C = AB + BX + AX.$$

The logic diagram for the above functions is shown in Figure 4.6. It is assumed complements of X, A, and B are available at the input source.

Note that one type of configuration of the combinational circuit diagram for full-adder is realized in Figure 4.6, with two-input and three-input AND gates, and three input and four-input OR gates. Other configurations can also be developed where number and type of gates are reduced. For this, the Boolean expressions of S and C are modified as follow.

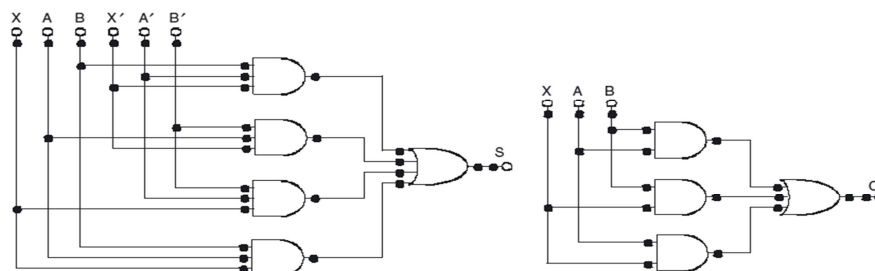


Figure (4-6)

$$\begin{aligned}
S &= X'A'B + X'AB' + XA'B' + XAB \\
&= X'(A'B + AB') + X(A'B' + AB) \\
&= X'(A \oplus B) + X(A \oplus B)' \\
&= X \oplus A \oplus B \\
C &= AB + BX + AX = AB + X(A + B) \\
&= AB + X(AB + AB' + AB + A'B) \\
&= AB + X(AB + AB' + A'B) \\
&= AB + XAB + X(AB' + A'B) \\
&= AB + X(A \oplus B)
\end{aligned}$$

Logic diagram according to the modified expression is shown Figure 4.7.

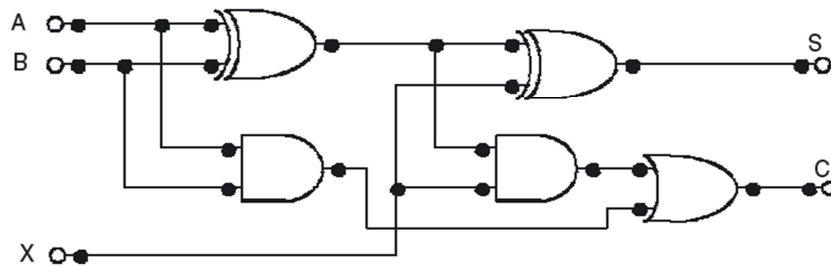


Figure (4-7)

You may notice that the full-adder developed in Figure 4.7 consists of two 2-input AND gates, two 2-input XOR (Exclusive-OR) gates and one 2-input OR gate. This contains a reduced number of gates as well as type of gates as compared to Figure 4.6. Also, if compared with a half-adder circuit, the full-adder circuit can be formed with two half-adders and one OR gate.

4.4 SUBTRACTORS

Subtraction is the other basic function of arithmetic operations of information-processing tasks of digital computers. Similar to the addition function, subtraction of two binary digits consists of four possible elementary operations, which are $0-0 = 0$, $0-1 = 1$ with borrow of 1, $1-0 = 1$, and $1-1 = 0$. The first, third, and fourth operations produce a subtraction of one digit, but the second operation produces a difference bit as well as a *borrow* bit. The borrow bit is used for subtraction of the next higher significant bit. A combinational circuit that performs the subtraction of two bits as described above is called a *half-subtractor*. The digit from which another digit is subtracted is called the *minuend* and the digit which is to be subtracted is called the *subtrahend*. When the minuend and subtrahend numbers contain more significant

digits, the borrow obtained from the subtraction of two bits is subtracted from the next higher-order pair of significant bits. Here the subtraction operation involves three bits—the minuend bit, subtrahend bit, and the borrow bit, and produces a different result as well as a borrow. The combinational circuit that performs this type of addition operation is called a *full-subtractor*. Similar to an adder circuit, a full-subtractor combinational circuit can be developed by using two half-subtractors.

4.4.1 Design of Half-subtractors

A half-subtractor has two inputs and two outputs. Let the input variables minuend and subtrahend be designated as X and Y respectively, and output functions be designated as D for difference and B for borrow. The truth table of the functions is as follows.

<i>Input variables</i>		<i>Output variables</i>	
<i>X</i>	<i>Y</i>	<i>D</i>	<i>B</i>
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Figure (4-8)

By considering the minterms of the truth table in Figure 4.8, the Boolean expressions of the outputs D and B functions can be written as

$$D = X'Y + XY'$$

and

$$B = X'Y.$$

Figure 4.9 shows the logic diagram to realize the half-subtractor circuit.

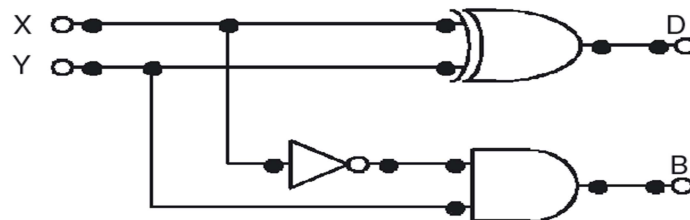


Figure (4-9)

4.4.2 Design of Full-subtractors

A combinational circuit of full-subtractor performs the operation of subtraction of three bits—the minuend, subtrahend, and borrow generated from the subtraction operation of previous significant digits and produces the outputs difference and borrow. Let us designate the input variables minuend as X, subtrahend as Y, and previous borrow as Z, and outputs difference as D and borrow as B. Eight different input combinations are possible for three input variables. The truth table is shown in Figure 4.10(a) according to its functions.

Input variables			Outputs	
X	Y	Z	D	B
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Figure 4-10 (a)

	Y'Z'	Y'Z	YZ	YZ'
X'		1		1
X	1		1	

Figure 4.10(b) Map for function D.

	Y'Z'	Y'Z	YZ	YZ'
X'		1	1	1
X			1	

Figure 4.10(b) Map for function B.

Karnaugh maps are prepared to derive simplified Boolean expressions of D and B as in Figures 4.10(b) and 4.10(c), respectively.

The simplified Boolean expressions of the outputs are

$$D = X'Y'Z + X'YZ' + XY'Z' + XYZ \text{ and}$$

$$B = X'Z + X'Y + YZ.$$

The logic diagram for the above functions is shown in Figure 4.11.

Similar to a full-adder circuit, it should be noticed that the configuration of the combinational circuit diagram for full-subtractor as shown in Figure 4.11 contains two-input and three-input AND gates, and three-input and four-input OR gates. Other

configurations can also be developed where number and type of gates are reduced. For this, the Boolean expressions of D and B are modified as follows.

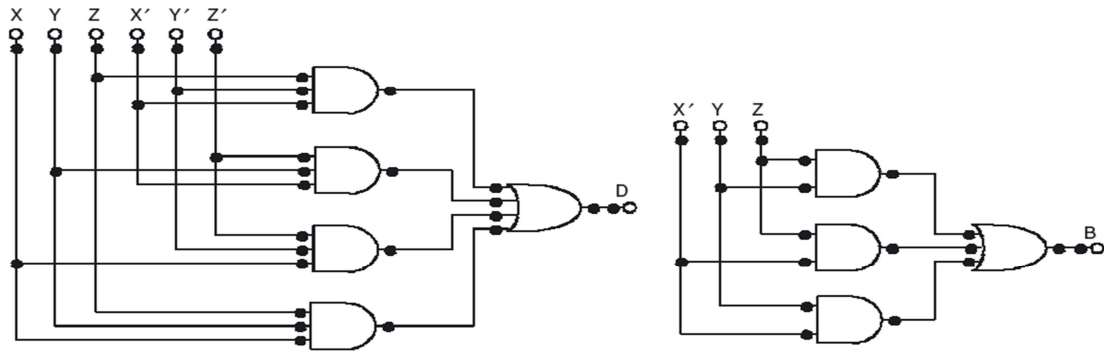


Figure 4-11

$$\begin{aligned}
 D &= X'Y'Z + X'YZ' + XY'Z' + XYZ \\
 &= X' (Y'Z + YZ') + X (Y'Z' + YZ) \\
 &= X' (Y \oplus Z) + X (Y \oplus Z)' \\
 &= X \oplus Y \oplus Z
 \end{aligned}$$

$$\begin{aligned}
 B &= X'Z + X'Y + YZ = X'Y + Z(X' + Y) \\
 &= X'Y + Z(X'Y + X'Y' + XY + X'Y) \\
 &= X'Y + Z(X'Y + X'Y' + XY) \\
 &= X'Y + X'YZ + Z(X'Y' + XY) \\
 &= X'Y + Z(X \oplus Y)'
 \end{aligned}$$

Logic diagram according to the modified expression is shown in Figure 4.12.

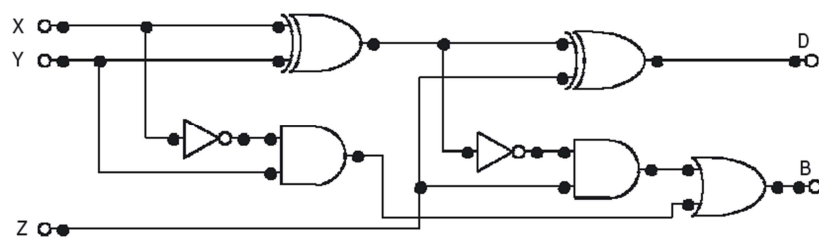


Figure 4-12

Note that the full-subtractor developed in Figure 4.12 consists of two 2-input AND gates, two 2-input XOR (Exclusive-OR) gates, two INVERTER gates, and one 2-input OR gate. This contains a reduced number of gates as well as type of gates as compared to Figure 4.12. Also, it may be observed, if compared with a half-subtractor circuit, the full-subtractor circuit can be developed with two half-subtractors and one OR gate.