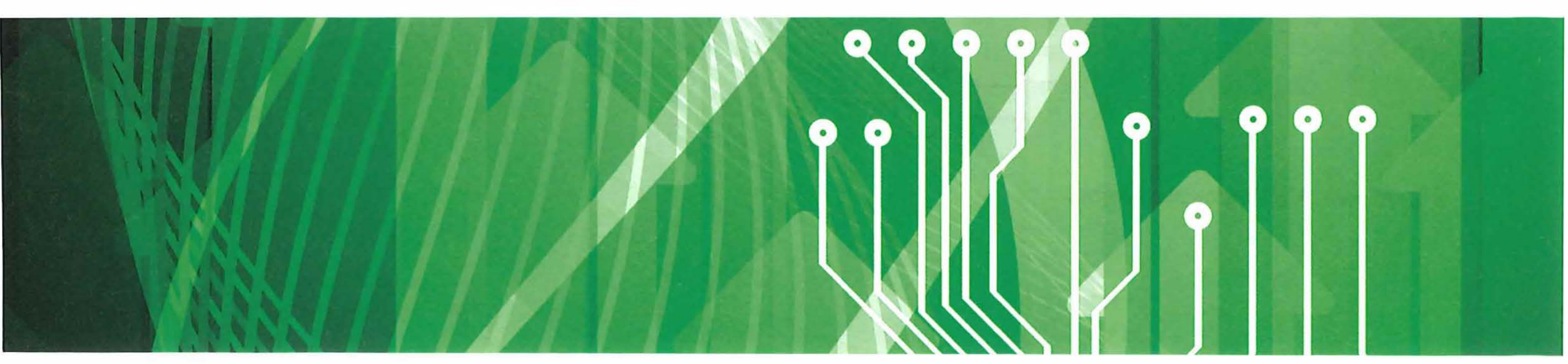
Chapter 4: Logic Gates

Learning objectives

By the end of this chapter you will:

- understand and be able to define the functions of AND, OR, NOT, NAND, NOR and XOR logic gates
- understand the binary output produced by the logic gates when two inputs are given (only one input will be given in a NOT gate)
- be able to produce a truth table from a logic gate
- be able to determine a logic gate from a truth table
- be able to produce a logic circuit to solve a given problem
- be able to produce a logic circuit for a given logic statement.



4.01 Types of logic gate

A **logic gate** is a basic foundation of a **digital circuit**. Data and instructions are both stored and transmitted by digital computers as the binary digits 0 and 1. (To learn more about binary digits see Chapter 1.)

Various tasks are performed in a computer by switching on a high **voltage** or a low voltage through a digital circuit called a logic gate. High voltage is represented by the binary digit 1 and low voltage is represented by the binary digit 0. Some circuits may only have a few logic gates, others, such as a **microprocessor**, could have thousands. Each logic gate controls the flow of electronic signals in a predetermined way.

Each logic gate that we will look at, except a NOT logic gate, will have two inputs and one output. A NOT logic gate will only have one input and one output. We can work out what the output of a logic gate will be from different inputs, using the rules of each different logic gate.

KEY TERM

Logic gate – the basic foundation of a digital circuit that controls the flow of electronic signals.

Digital circuit – a circuit where electronic signals are one of two values, high voltage (1) or low voltage (0).

Voltage – the potential difference across an electrical component needed to make electricity flow through it.

Microprocessor – an integrated circuit that provides the same functions of a CPU.

Truth table – a way of showing every outcome of a logic gate.

SYLLABUS CHECK

Understand and define the functions of NOT, AND, OR, NAND, NOR and XOR (EOR) gates, including the binary output produced from all the possible binary inputs (all gates, except the NOT gate, will have 2 inputs only).

Recognise and use standard symbols to represent logic gates: NOT, AND, OR, NAND, NOR and XOR.

The AND gate

When using an AND logic gate the output (X) will only be high voltage if both inputs (A and B) are high voltage. This means that X can only be 1 if A and B are 1.

We can write logic statements to represent a logic gate or logic circuit. Logic statements are also known as 'Boolean statements' or 'Boolean logic'. A logic statement for an AND gate is:

X = A AND B   or   X = A.B

A dot (.) can be used to represent the word AND in a logic statement.

The AND logic gate is represented by the symbol shown in Figure 4.01.

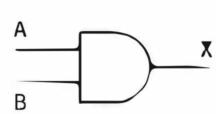


Figure 4.01 AND logic gate symbol

This logic gate has the labels A and B for the inputs and X as the output. The logic of this circuit is that if A AND B are 1 then X will be 1. If either A or B are 0 then X will be 0.

Truth tables

SYLLABUS CHECK

Draw truth tables and recognise a logic gate from its truth table.

A **truth table** is a way of showing every outcome of a logic gate. The value of each input is shown along with the value that will be output as a result of these inputs. We can represent the logic of the AND gate in a truth table as shown in Figure 4.02.

Input		Output
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

Reading a truth table

→ If A = 0 and B = 0 then X = 0

→ If A = 0 and B = 1 then X = 0

→ If A = 1 and B = 0 then X = 0

→ If A = 1 and B = 1 then X = 1

Figure 4.02 Logic of the AND gate in a truth table

With two inputs to the AND gate, there are four combinations and two possible outcomes.

The OR gate

When using an OR logic gate the output X will be high voltage if input A or input B is high voltage. This means that X will be 1 if A or B is 1.

As a logic statement we could write this as:

$$X = A \text{ OR } B \qquad \text{or} \qquad X = A+B$$

A cross sign (+) can be used to represent the word OR in a logic statement.

The logic gate OR is represented by the symbol shown in Figure 4.03.

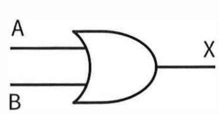


Figure 4.03 OR logic gate symbol

The logic of this circuit is that if A or B are 1 then X will be 1. If A and B are 0 then X will be 0.

We can represent the logic of this gate in a truth table.

Table 4.01

Input		Output
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

With two inputs to the OR gate, there are four possible outcomes.

The NOT gate

When using a NOT logic gate there is only one input and one output. The output X will be high voltage only if the input is low voltage. This means that X will only be 1 if A is 0.

As a logic statement we could write this as:

$$X = \text{NOT } A \qquad \text{or} \qquad X = \bar{A}$$

A line, called a vinculum, over an input label can be used to represent the word NOT in a logic statement.

The logic gate NOT is represented by the symbol shown in Figure 4.04.

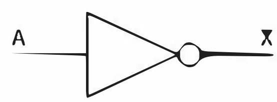


Figure 4.04 NOT logic gate symbol

We can represent the logic of the NOT gate in a truth table:

Table 4.02

Input	Output
A	X
0	1
1	0

With one input to the NOT gate, there are two possible outcomes. In a NOT gate, the output is always the opposite of the input.

As well as producing a truth table for a logic gate it is important that we can recognise a logic gate from a truth table. To do this we need to look at the output of the logic circuit and at the inputs that go into it. We need to recall the rules of each logic gate and see which set of rules are being applied to the inputs to achieve each output. For example, if we can see that the output is only 1 when both inputs are 1, from the rules we know this is an AND gate.

Logic gates can be combined. When certain types of logic gates are combined they create another type of logic gate.

The NAND gate

The NAND logic gate is a combination of an AND gate followed by a NOT gate. The output X will be high voltage unless both inputs are high voltage. If both inputs are high voltage the output will be low voltage. This means that X will always be 1 unless A and B are 1.

As a logic statement we could write this as:

$$X = \text{NOT (A AND B)} \qquad \text{or} \qquad X = \overline{A.B}$$

The logic gate NAND is represented by the symbol shown in Figure 4.05.

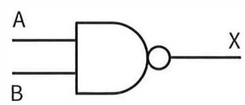


Figure 4.05 NAND logic gate symbol

Broken down into the two original logic gates before they are combined, it would be as shown in Figure 4.06.

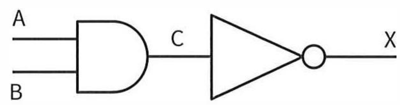


Figure 4.06 Logic gates NOT and NAND before they are combined

We can represent the logic of the NAND gate in a truth table. To understand this logic gate more easily we will refer to the interim output/input in Table 4.3 as C, as shown in Figure 4.06:

Table 4.03

Input		Interim	Output
A	B	C	X
0	0	0	1
1	0	0	1
0	1	0	1
1	1	1	0

We would complete a truth table like this by setting each of the starting inputs to either 1 or 0, writing down what that will give as an interim value, then taking that interim value and writing down what that would give as a final output.

For example, the first line of the truth table states that both inputs A and B are 0. We know that if either input A or B in an AND gate are 0, then the output will always be 0. Therefore, we write down 0 as the interim value.

We then use this interim value as the input for the next logic gate. We know that if the input to a NOT gate is 0, the output will always be the opposite, so 1. Therefore we write down 1 as the output for X. With two inputs to the NAND gate there are four possible outcomes.

We need to make sure that for the input values we have covered every combination of 0 and 1 that could occur. As we can see with the AND gate above, we could have 0 and 0, 1 and 0, 0 and 1, 1 and 1 as combinations.

The NOR gate

The NOR logic gate is a combination of an OR gate followed by a NOT gate. The output X will be high voltage unless both inputs are high voltage. If either input is high voltage the output will be low voltage. This means that X will always be 0 unless A and B are 0.

As a logic statement we could write this as:

$$X = \text{NOT (A OR B)} \quad \text{or} \quad X = \overline{A+B}$$

The logic gate NOR is represented by the symbol shown in Figure 4.07.

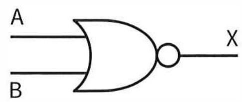


Figure 4.07 NOR logic gate symbol

Broken down into the two original logic gates before they are combined, it would be as shown in Figure 4.08.

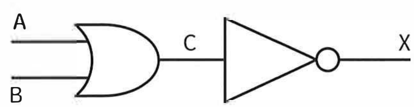


Figure 4.08 Logic gates NOT and OR before they are combined

To understand this logic gate more easily we will refer to the interim output/input as C in the truth table:

Table 4.04

Input		Interim	Output
A	B	C	X
0	0	0	1
1	0	1	0
0	1	1	0
1	1	1	0

With two inputs to the NOR gate, there are four possible outcomes.

The XOR (EOR) gate

An XOR or Exclusive OR gate has a slightly changed logic from an OR gate. Similar to an OR gate, the output X will be high voltage if either of the inputs are high voltage. However in an XOR gate, if both inputs are high voltage, the output will be low voltage. This means that X will always be 1 if inputs A and B are different.

As a logic statement we could write this as:

$$X = \text{(NOT A AND B)} + \text{(A AND NOT B)} \quad \text{or} \quad X = A \oplus B$$

The logic gate XOR is represented by the symbol shown in Figure 4.09.

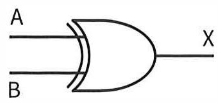


Figure 4.09 XOR logic gate symbol

We can represent the logic of the XOR gate in a truth table:

Table 4.05

Input		Output
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

With two inputs to the XOR gate, there are four possible outcomes.

When we start to combine a series of logic gates, working out the output value can be very challenging.

4.02 Logic circuits

SYLLABUS CHECK

Use logic gates to create electronic circuits.

Produce truth tables for given logic circuits.

A logic circuit is an electronic circuit that combines a number of logic gates. The gates in the logic circuit will control the flow of electricity through the circuit.

Creating a truth table for a logic circuit

A logic circuit using an AND, an OR and an XOR logic gate is shown in Figure 4.10.

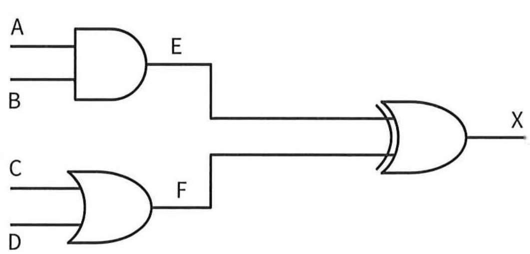


Figure 4.10 Logic circuit using an AND, OR and XOR logic gate

We have added interim values to this logic circuit to help us work out the output. In the truth table we will refer to these as work space as they are needed to work out the output. A truth table for this would look like Table 4.06.

Table 4.06

Input				Work space		Output
A	B	C	D	E	F	X
0	0	0	0	0	0	0
1	0	0	0	0	0	0
0	1	0	0	0	0	0
1	1	0	0	1	0	1
0	0	1	0	0	1	1
1	0	1	0	0	1	1
0	1	1	0	0	1	1
1	1	1	0	1	1	0
0	0	0	1	0	1	1
1	0	0	1	0	1	1
0	1	0	1	0	1	1
1	1	0	1	1	1	0
0	0	1	1	0	1	1
1	0	1	1	0	1	1
0	1	1	1	0	1	1
1	1	1	1	1	1	0

We would complete this truth table by starting at the top left of the circuit. It is often a good place to start by setting all the input values in the first gates to 0. Therefore, we would write down that input A and B are both 0.

We know that if either input into an AND gate is 0, the output will be 0. Therefore, we would write down the output from this gate, which we have labelled as E, is 0.

We would then look at the logic gate below the AND gate, which is an OR gate. We again can start by setting both input values to 0. We know that if both input values are 0 in an OR gate, the output will be 0. Therefore we would write down the output from this gate, which we have labelled as F, as 0.

We then move to the next logic gate that outputs E and F go to, which is an XOR gate. E and F now become the inputs for this gate. We know that if both inputs for an XOR gate are 0, the output will be 0. Therefore we can write down that output X is 0.

As there are more inputs into this logic circuit, there are more combinations that can occur. We need to make sure that we have covered all combination that could occur across the four initial inputs, for example 0000, 1000, 0100, etc. When we have four inputs into a logic circuit it creates 16 possible combinations.

Figure 4.11 shows a logic circuit using a NAND, a NOR and an OR gate:

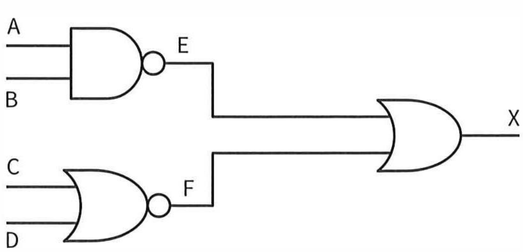


Figure 4.11 Logic circuit using a NAND, NOR and OR gate

We again have added extra labels to this logic circuit to help us work out the output. In the truth table we will refer to these as work space. A truth table for this would look like Table 4.07:

Table 4.07

Input				Work space		Output
A	B	C	A	B	C	A
0	0	0	0	1	1	1
1	0	0	0	1	1	1
0	1	0	0	1	1	1
1	1	0	0	0	1	1
0	0	1	0	1	0	1
1	0	1	0	1	0	1
0	1	1	0	1	0	1
1	1	1	0	0	0	0
0	0	0	1	1	0	1
1	0	0	1	1	0	1
0	1	0	1	1	0	1
1	1	0	1	0	0	0
0	0	1	1	1	0	1
1	0	1	1	1	0	1
0	1	1	1	1	0	1
1	1	1	1	0	0	0

Creating a circuit diagram from a logic statement

SYLLABUS CHECK

Produce a logic circuit to solve a given problem or to implement a given written logic statement.

A logic circuit can also be written in the form of a logic statement. From this logic statement we can produce a logic circuit diagram.

Let us look at the following logic statement:

X = NOT(A OR B)

From looking at this statement we can see that we will need to use the logic gates NOT and OR. We can see the output X will come from a NOT gate.

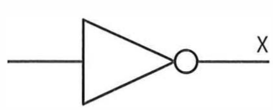


Figure 4.12 NOT logic gate

The interim input into this NOT gate comes from the output of an OR gate. This OR gate has two inputs, A and B. Figure 4.13 shows the logic circuit for the statement X = NOT(A OR B).

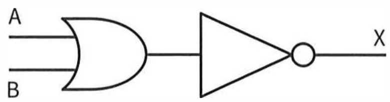


Figure 4.13 Logic circuit for the statement X = NOT (A OR B)

TEST YOURSELF

1 What name is given to the combination of the two gates in Figure 4.13?
2 Draw the logic circuit for the logic statement X = A AND (B AND C).

Let us look at another logic statement:

X = (NOT C) AND (A OR B)

From looking at this statement we can see that we will need to use the logic gates AND, NOT and OR. We can see the output X will be determined by NOT C and A OR B. Therefore, it is the AND gate that output X will come from. The AND gate will link the NOT and the OR gates.

We have noted that the output X will come from an AND gate:

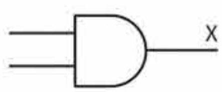


Figure 4.14 AND gate

Into the AND gate will be a NOT logic gate with an input of C:

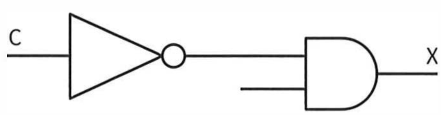


Figure 4.15 NOT logic gate with an input of C into the AND gate

Also into the AND gate will be an OR gate with the inputs A and B:

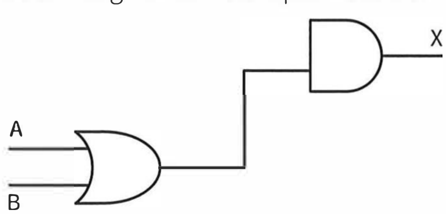


Figure 4.16 OR gate with inputs A and B into the AND gate

Figure 4.17 shows the full logic circuit for the logic statement X = (NOT C) AND (A OR B).

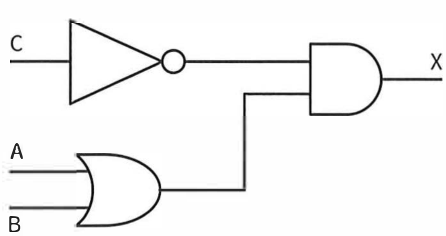


Figure 4.17 Full logic circuit for the statement X = (NOT C) AND (A OR B)

TEST YOURSELF

Produce a truth table for the logic circuit in Figure 4.17?

Creating a logic circuit to solve a problem

We can also create a logic circuit to solve a problem.

In a system there are three buttons (A, B and C) and a lamp (X). The lamp only comes on if B or C are pressed, but not if they are pressed together. If A is pressed the lamp goes off even if B or C are pressed.

If we start to think about the logic of this problem we can see that if NOT(A is pressed) AND (B is pressed or C is pressed, but not together) then X will light up.

Therefore, as a logic statement this would be X = (NOT A) AND (B XOR C).

Now we have a logic statement for our problem we can create our logic circuit. We can see from our logic statement that the output will need to come from an AND gate. Going into this AND gate will need to be a NOT gate with an input of A and an XOR gate with inputs B and C.

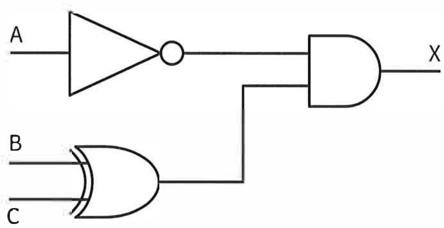


Figure 4.18 Full logic circuit using AND, NOT and XOR gates

TEST YOURSELF

Produce a truth table for the logic circuit in Figure 4.18?

Summary

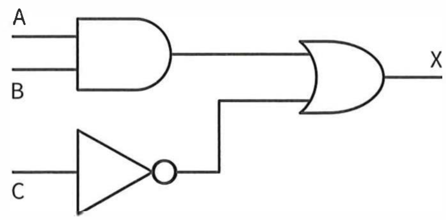
- Various tasks are performed in a computer by switching on a high voltage or a low voltage through a digital circuit called a logic gate. Each logic gate controls the flow of electronic signals in a predetermined way.
- When using an AND logic gate the output (X) will only be high voltage if both inputs (A and B) are high voltage.
- When using an OR logic gate the output X will be high voltage if input A or input B is high voltage.
- When using a NOT logic gate there is only one input and one output. The output X will be high voltage only if the input is low voltage.
- The NAND logic gate is a combination of an AND gate followed by a NOT gate. The output X will be high voltage unless both inputs are high voltage. If both inputs are high voltage the output will be low voltage.
- The NOR logic gate is a combination of an OR gate followed by a NOT gate. The output X will be high voltage unless both inputs are high voltage. If either input is high voltage the output will be low voltage.

- An XOR or Exclusive OR logic gate has a slightly changed logic from an OR gate. Similar to an OR gate, the output X will be high voltage if either of the inputs are high voltage. However, in an XOR gate, if both inputs are high voltage, the output will be low voltage.
- A truth table is a way of showing every outcome of a logic gate.
- A logic circuit is an electronic circuit that combines a number of logic gates. The gates in the logic circuit will control the flow of electricity through the circuit.

Exam-style questions

1 Describe the role of a logic gate in a logic circuit. (1 mark)

2 Consider the following logic circuit:



a Write the logic statement to represent this logic gate. (3 marks)

b When A = 0, B = 0 and C = 1, output X = ... (1 mark)

c When A = 1, B = 0 and C = 1, output X = ... (1 mark)

3 Draw the logic circuit for the logic statement X = (A OR B) AND C. (2 marks)

4 The following truth table is for the logic statement X = A XOR (B AND C) (3 marks)

Fill in the missing values to complete the truth table.

Input			Output
A	B	C	X
0	0	0	
1	0	0	1
0		0	0
1	1		
	0	1	0
1	0	1	1
0	1		1
1	1	1	

5 In a system there are three switches and an alarm. The alarm will sound only if switches A and B are on. If switch C is turned on then the alarm will stop.

a Write the logic statement for this problem. (2 marks)

b Draw the logic circuit for this problem. (2 marks)

c Create a truth table for this problem. (2 marks)