Cambridge

OL- IGCSE

Computer science

CODE: (0478)

Chapter 10

Boolean logic

10.1 Standard logic gate symbols

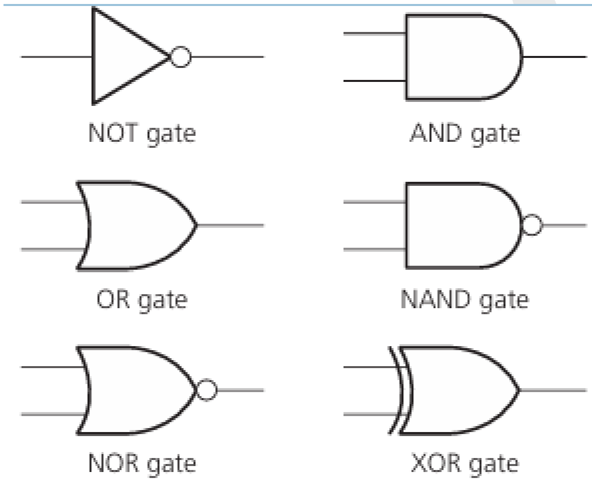
Electronic circuits in computers, solid state drives and controlling devices are made up of thousands of **logic gates**. Logic gates take binary inputs and produce a binary output. Several logic gates combined together form a **logic circuit** and these circuits are designed to carry out a specific function.

The checking of the output from a logic gate or logic circuit is done using a **truth table**.

This chapter will consider the function and role of logic gates, logic circuits and truth tables. Also a number of possible applications of logic circuits will be considered. A reference to **Boolean algebra** will be made throughout the chapter; but this is really outside the scope of this text book. However, Boolean algebra will be seen on many logic gate websites and is included here for completeness, since many students may prefer this notation to logic statements.

10.1.1 Logic gate symbols

Six different logic gates will be considered in this chapter:



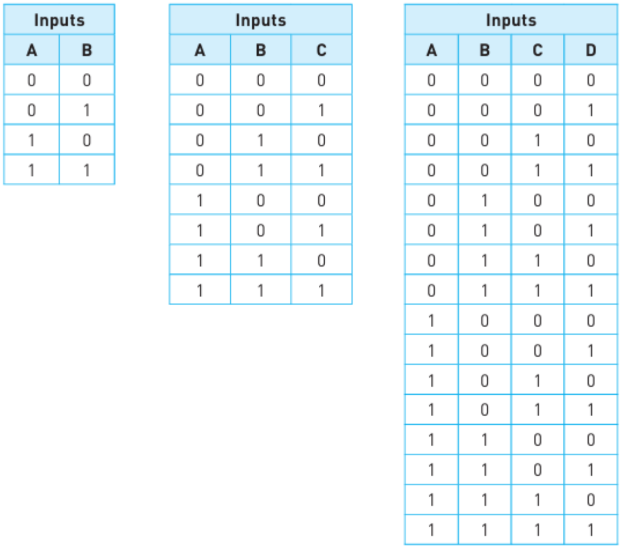
▲ **Figure 10.1** Logic gate symbols

Truth tables

Truth tables are used to trace the output from a logic gate or logic circuit. The NOT gate is the only logic gate with one input; the other five gates have two inputs (see Figure 10.1).

Although each logic gate can only have one or two inputs, the number of inputs to a logic circuit can be more than

▼ **Table 10.1** All possible inputs for truth tables with two, three and four inputs



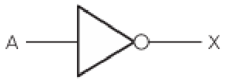
Inputs	
A	B
0	0
0	1
1	0
1	1

Inputs		
A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Inputs			
A	B	C	D
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

10.2 The function of the six logic gates

10.2.1 NOT gate



▲ **Figure 10.2**

<table>
<tr><th>Description:</th><th>Truth table:</th><th>How to write this:</th></tr>
<tr>
<td>The output, X, is 1 if:
the input, A, is 0</td>
<td>▼ Table 10.2
<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><th>A</th><th>X</th></tr>
<tr><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td></tr>
</table>
</td>
<td>X = NOT A (logic notation)
X = $\overline{A}$ (Boolean algebra)</td>
</tr>
</table>

Note the use of Boolean algebra to represent logic gates. This is optional at IGCSE but many students may prefer to use this notation (see NOTE later).

10.2.2 AND gate

▲ **Figure 10.3**

<table>
<tr><th>Description:</th><th>Truth table:</th><th>How to write this:</th></tr>
<tr>
<td>The output, X, is 1 if:
both inputs, A and B, are 1</td>
<td>▼ Table 10.3
<table>
<tr><th colspan="2">Inputs</th><th>Outputs</th></tr>
<tr><th>A</th><th>B</th><th>X</th></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>
</td>
<td>X = A AND B (logic notation)
X = A . B (Boolean algebra)</td>
</tr>
</table>

10.2.3 OR gate

▲ **Figure 10.4**

<table>
<tr><th>Description:</th><th>Truth table:</th><th>How to write this:</th></tr>
<tr>
<td>The output, X, is 1 if:
either input, A or B, or both, are 1</td>
<td>▼ Table 10.4
<table>
<tr><th colspan="2">Inputs</th><th>Output</th></tr>
<tr><th>A</th><th>B</th><th>X</th></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>
</td>
<td>X = A OR B (logic notation)
X = A + B (Boolean algebra)</td>
</tr>
</table>

10.2.4 NAND gate (NOT AND)

▲ **Figure 10.5**

<table>
<tr><th>Description:</th><th>Truth table:</th><th>How to write this:</th></tr>
<tr><td>The output, X, is 1 if:
input A AND input B are NOT both 1</td><td>▼ Table 10.5
<table><tr><th colspan="2">Inputs</th><th>Output</th></tr><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td><td>X = A NAND B (logic notation)
$X = \overline{A . B}$ (Boolean algebra)</td></tr>
</table>

10.2.5 NOR gate (NOT OR)

▲ **Figure 10.6**

<table>
<tr><th>Description:</th><th>Truth table:</th><th>How to write this:</th></tr>
<tr><td>The output, X, is 1 if:
neither input A nor input B is 1</td><td>▼ Table 10.6
<table><tr><th colspan="2">Inputs</th><th>Output</th></tr><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td><td>X = A NOR B (logic notation)
$X = \overline{A + B}$ (Boolean algebra)</td></tr>
</table>

10.2.6 XOR gate

▲ **Figure 10.7**

<table>
<tr><th>Description:</th><th>Truth table:</th><th>How to write this:</th></tr>
<tr><td>The output, X, is 1 if:
(input A is 1 AND input B is 0)
or
(input A is 0 AND input B is 1)</td><td>▼ Table 10.7
<table><tr><th colspan="2">Inputs</th><th>Output</th></tr><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td><td>X = A XOR B (logic notation)
$X = (A . \overline{B}) + (\overline{A} . B)$ (Boolean algebra)
NOTE: this is sometimes written as: $(A + B) . (\overline{A . B})$</td></tr>
</table>

You will notice in the Boolean algebra, three new symbols; these have the following meaning:
» . represents the AND operation
» + represents the OR operation
» a bar (above the letter or letters, e.g. a) represents the NOT operation.

10.3 Logic circuits, logic expressions, truth tables and problem statements

When logic gates are combined together to carry out a particular function, such as controlling a robot, they form a logic circuit. The following eight examples show how to carry out the following tasks:

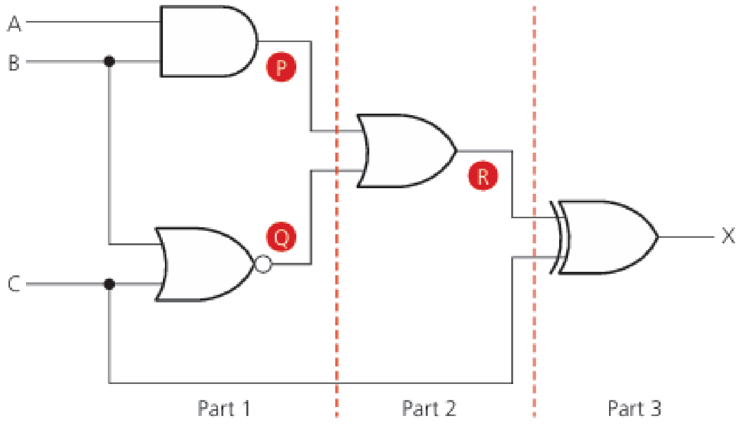
» Create a logic circuit from a:
– problem statement (examples 6 and 7)
– logic or Boolean expression (examples 3 and 8)
– truth table (examples 4 and 5

» Complete a truth table from a:
– problem statement (examples 6 and 7)
– logic or Boolean expression (examples 3 and 8)
– logic circuit (example 1)

» Write a logic or Boolean expression from a:
– problem statement (examples 6 and 7)
– logic circuit (example 2)
– truth table (examples 4 and 5

Example 1

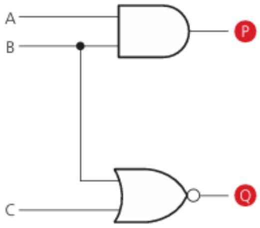
Produce a truth table for the following logic circuit (note the use of black circles at the junctions between wires):



▲ **Figure 10.8**

There are three inputs to this logic circuit, therefore, there will be eight possible binary values that can be input.

To show stepwise how the truth table is produced, the logic circuit has been split up into three parts, as shown by the dotted lines, and intermediate values are shown as P, Q and R.



▲ **Figure 10.9**

Part 1

This is the first part of the logic circuit; the first task is to find the intermediate values P and Q.

The value of P is found from the AND gate where the inputs are A and B. The value of Q is found from the NOR gate where the inputs are B and C. An intermediate truth table is produced using the logic function descriptions in Section 10.2.

▼ **Table 10.8**

input values			**Output values**	
A	**B**	**C**	**P**	**Q**
0	0	0	**0**	**1**
0	0	1	**0**	**0**
0	1	0	**0**	**0**
0	1	1	**0**	**0**
1	0	0	**0**	**1**
1	0	1	**0**	**0**
1	1	0	**1**	**0**
1	1	1	**1**	**0**

▲ **Figure 10.10**

Part 2

The second part of the logic circuit has P and Q as inputs and the intermediate output, R:

This produces the following intermediate truth table. (Note: even though there are only two inputs to the logic gate, we have generated eight binary values in part 1 and these must all be used in this second truth table).

▼ **Table 10.9**

Inputs		Output
P	Q	R
0	1	**1**
0	0	**0**
0	0	**0**
0	0	**0**
0	1	**1**
0	0	**0**
1	0	**1**
1	0	**1**

▲ **Figure 10.11**

Part 3

The final part of the logic circuit has R and C as inputs and the final output, X:

This gives the third intermediate truth table:

▼ **Table 10.10**

Inputs		Output
R	C	X
1	0	**1**
0	1	**1**
0	0	**0**
0	1	**1**
1	0	**1**
0	1	**1**
1	0	**1**
1	1	**0**

Putting all three intermediate truth tables together produces the final truth table, which represents the original logic circuit:

▼ **Table 10.11**

Input values			Intermediate values			Output
A	B	C	P	Q	R	X
0	0	0	*0*	*1*	*1*	1
0	0	1	*0*	*0*	*0*	1
0	1	0	*0*	*0*	*0*	0
0	1	1	*0*	*0*	*0*	1
1	0	0	*0*	*1*	*1*	1
1	0	1	*0*	*0*	*0*	1
1	1	0	*1*	*0*	*1*	1
1	1	1	*1*	*0*	*1*	0

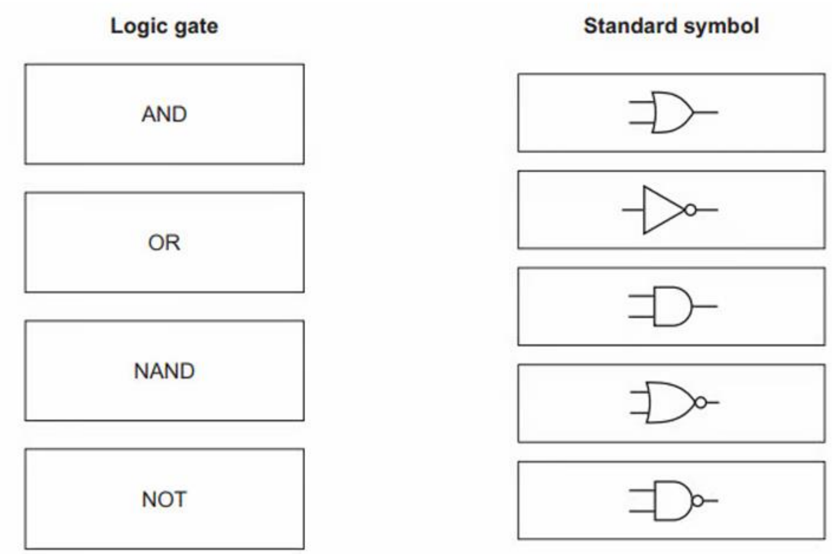
The intermediate values can be left out of the final truth table, but it is good practice to leave them in until you become confident about producing the truth tables. The final truth table would then look like this:

▼ **Table 10.12**

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

Revision questions

1. Four logic gates and five standard symbols for logic gates are shown Draw one line to link each logic gate to its standard symbol. Not all standard symbols will be used.



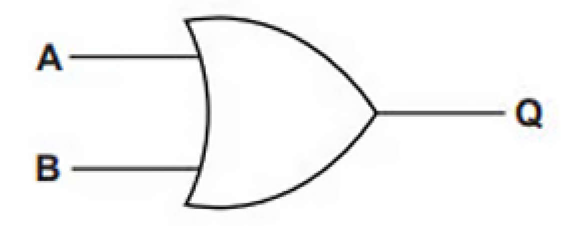
2 (a) Complete the truth table in Fig. 1 for the Boolean statement P = NOT (A AND B).

A	B	P
0	0	1
0	1	
1	0	
1	1	0

Fig. 1

(b) Complete the truth table for the following logic gate.

A	B	Q
0	0	0
0	1	1
	0	
1		



2. (a) Consider this logic expression.

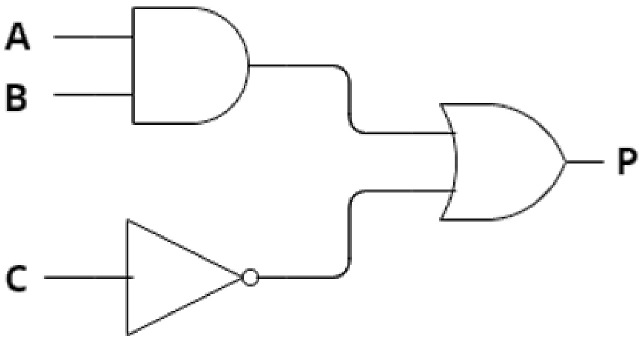
X = (**A** AND **B**) OR (**B** AND NOT **C**)

Draw a logic circuit for this logic expression.
Each logic gate must have a maximum of two inputs.
Do not simplify this logic expression.

(b) Complete the truth table from the given logic expression.

A	B	C	Working space	X
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	1		

3. Write the Boolean expression represented by the logic diagram below:



4. (a) Consider the truth table:

Draw a logic circuit to represent the given truth table.

Each logic gate should have maximum of two inputs.

Do not simplify the logic circuit

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

b) Write a logic expression for the given truth table. Do not simplify the logic expression

5. (a) Consider this logic expression.

Z = (NOT **A** OR **B**) AND (**B** XOR **C**)

Draw a logic circuit for this logic expression.
Each logic gate must have a maximum of two inputs.
Do not simplify this logic expression.

(b) Complete the truth table from the given logic expression.

A	B	C	Working space	X
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

6. Consider this logic expression

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

Complete the truth table for this logic expression.

A	B	C	Working space	X
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

7. Consider the following logic statement

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

Draw a logic circuit to represent the given logic statement.

Do not attempt to simplify the logic statement. All logic gates must have a maximum of two inputs.

(b) Complete the truth table for the given logic statement

A	B	C	Working space	X
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

8. (a) Consider the following logic statement:

X = (((**A** AND NOT **B**) OR (NOT (**B** NOR **C**))) AND **C**)

Draw a logic circuit to represent the given logic statement.
Do not attempt to simplify the logic statement. All logic gates must have a maximum of two inputs

(b) Complete the truth table for the given logic statement

A	B	C	Working space	X
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		