Module 2 – Switching circuits and logic elements

Module 2 overview

Switching circuits and logic elements are the building blocks of logic systems. This can be seen in the following concept map.

Combinational logic design (concept map)

Objectives

At the completion of this module you will be able to:

- describe switches and their operation in electronic circuits
- show how switching circuit theory may be implemented by logic gates
- describe AND, OR, NOT, NAND, and NOR gates and define their operation with truth tables
- define switch bounce and show its effects on logic systems.

2.1 Switches

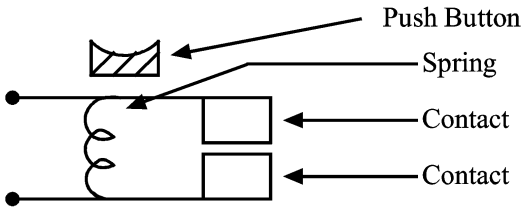
All computer systems are based on logic. Logic may be defined in several ways but in this context we mean logic as the 'ordered interpretation of a series of switched signals'. The signals are usually a fixed level DC voltage and the switching may be achieved manually by depressing a key (switch) on a computer keyboard, or automatically by an electronic circuit (logic gate) inside the computer itself.

The switching may be slow, for example one keystroke per second such as a student typing in an assignment, or fast, for example twenty million per second such as a logic gate processing a signal.

Let us examine some of these switches and see how they are related to logic systems. It is important for you to realise that a computer, no matter what type or complexity, is composed of millions of switches arranged in various combinations, to achieve many different functions.

2.2 Switch types

The most basic of switches is a mechanical device consisting of two contacts either held together or apart by a spring.

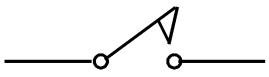


Switch Assembly

Pressure applied to the push button overcomes the mechanical force of the spring and the contacts move together (close) and if the switch was connected in a circuit, current could flow from one contact to the other.

Several terms are used to describe the operation of switches e.g. the switch shown here is normally 'OFF' and is only 'ON' when manually pressed. Other terms for ON and OFF are 'CLOSED' and 'OPEN', 'ENABLED' and 'DISABLED' or even a 'TRUE' or 'FALSE' condition may be said to exist.

The electrical symbol for a switch usually only depicts the contact set in its relaxed or 'rest' state. Here are some typical switch symbols.

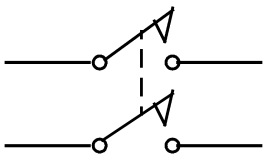


One set of contacts.

Single Pole Switch

Operation of a switch is historically called '**throw**'. For example this is called a Single Pole, Single Throw switch and is abbreviated in circuit diagrams to a SPST switch.

Here is another type of switch:

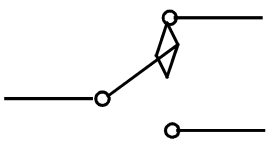


Two sets of contacts, both operate together.

Double Pole Switch

This is a Double Pole Single Throw switch or DPST.

Yet another type is:



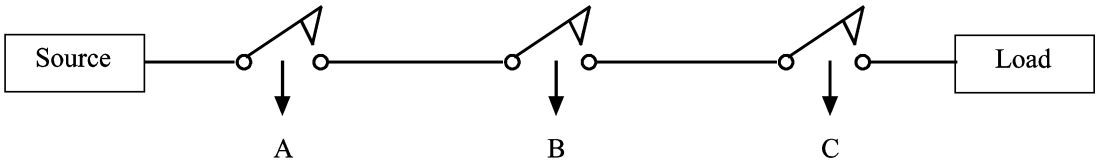
The switch contact can be either up as shown, or down.

This is a Single Pole Double Throw or SPDT switch.

2.3 Switching circuits

2.3.1 Series switches

In this circuit we have a ‘source’ of energy connected to a load via three SPST switches in series.



Statement: The load is ‘enabled’ (i.e. connected to the source) only if we have switches A and B and C all ‘activated’ (i.e. switched on).

2.3.2 Logic representation

If we consider the operation of the first switch as: A = ON and NOT A = OFF = $\overline{A}$

and so on for the other switches, we can see the load is connected to the source only during the conditions:

A AND B AND C

In logic terminology this is written

A · B · C Where the symbol ‘.’ equals the ‘AND’ function.

For 3 switches, each with 2 possible states (i.e. **on** or **off**) we will have 8 possible combinations, usually presented in tabular form called a ‘truth table’.

2.3.3 Truth tables

In logic work it is usual to indicate the **on** or ‘True’ condition as **logic one** (1) and the **off** or ‘False’ condition as **logic zero** (0).

We can then write a table of all possible combinations or conditions for a system. It is called a **truth table**. It is a statement defining all possible **inputs** and all possible **outputs** for a logic system.

In this case a truth table would be:

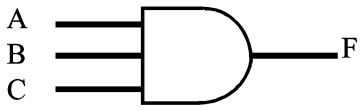
(Inputs) Switch Conditions A	B	C	(Outputs) Load
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

only condition for load to be enabled

2.3.4 Logic gates

This situation is used many times in logic circuits so a special electronic device is manufactured to do the task. Its called an **AND gate**.

Symbol:



3 Input AND Gate

Usually 3 or 4 of these gates are packaged in one integrated circuit (IC). It is normal to have 2 or 3 or more inputs to any one AND gate.

We can write a logic equation for this gate:

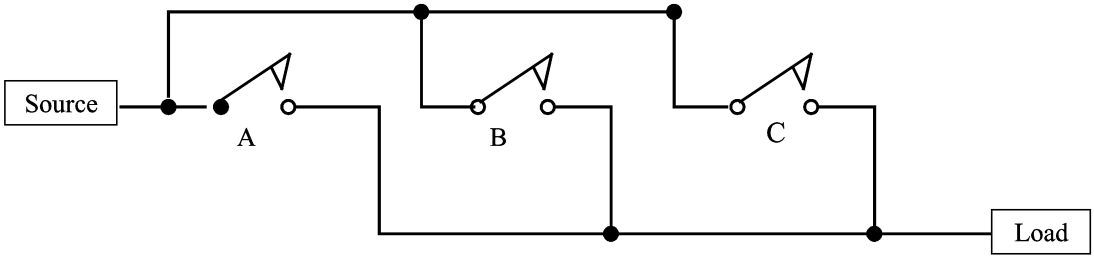
$$F = A \cdot B \cdot C$$

Often for convenience the '**.**' symbol is omitted and we write:

$$F = ABC$$

2.3.5 Parallel switches

Now consider these switches:



In this case, the load is energised if we activate:

A OR B OR C

In logic terminology this is written:

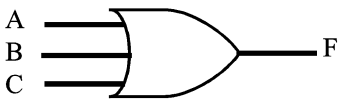
A + B + C where + = the 'OR' function.

Again we have 8 possible conditions hence the truth table is:

Switches			Output
A	B	C	
0	0	0	0
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	1

A gate has also been manufactured for this facility called an OR gate.

Symbol:

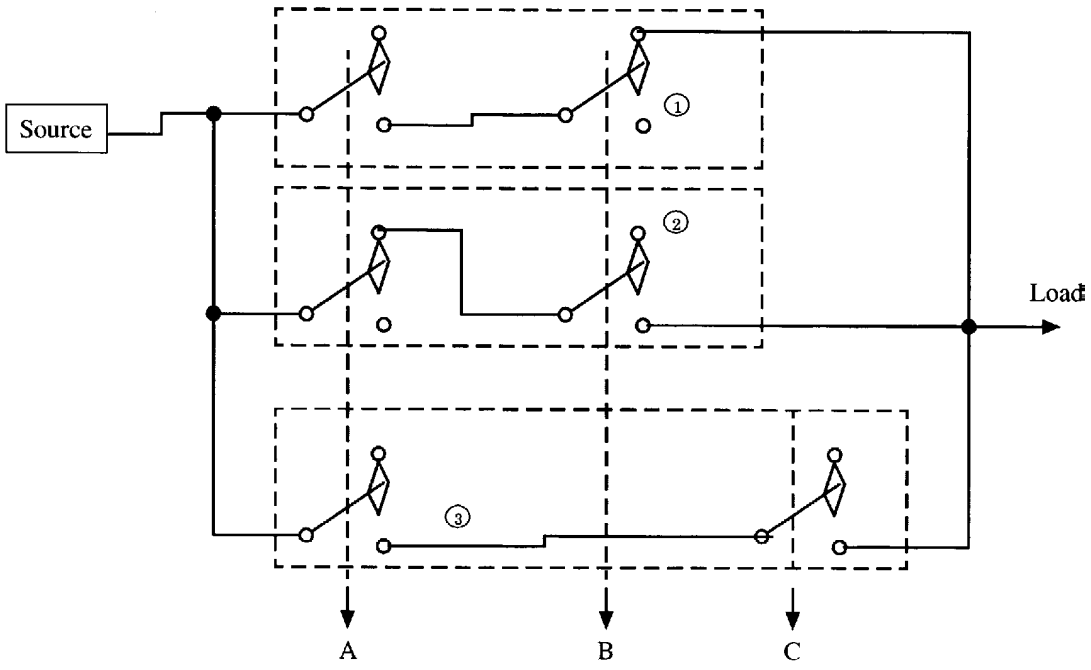


The equation is:

$$F = A + B + C$$

2.3.6 Combination switches

Suppose we had a combination:



You can see the system is a combination OR, AND circuit for the source to energise the load.

We wish to produce a truth table for this arrangement.

It's best to break it down to 3 areas as shown in the dotted boxes.

True conditions for ① are A $\overline{B}$ (A and not B)

and for ② are $\overline{A}$ B (not A and B)

and for ③ are A C (A and C)

Now circuits ① and ② can be combined as an OR circuit:

$$A\ \overline{B} + \overline{A}\ B$$

We usually drop the '**.**' unless its necessary for some other reason.

$$A\overline{B} + \overline{A}B$$

This combined expression is now OR'ed with ③ to give the final result:

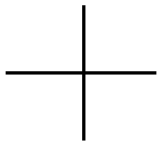
$$A\overline{B} + \overline{A}B + AC$$

The truth table could then be constructed in two stages. Firstly the areas enclosed by the dotted lines, AND functions, and then each of these could be considered as the OR functions.

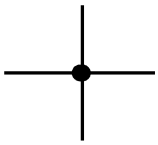
Inputs			'AND' Facility			'OR' Facility
A	B	C	$A\overline{B}$	$\overline{A}B$	AC	AB + AB + AC
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	1	0	1
0	1	1	0	1	0	1
1	0	0	1	0	0	1
1	0	1	1	0	1	1
1	1	0	0	0	0	0
1	1	1	0	0	1	1

The right hand column is really all that is required as it is the circuit output. The middle column is an intermediate stage included for the sake of clarity.

Note: When drawing circuit diagrams the following conventions are always used.



Wire crossing and not joining

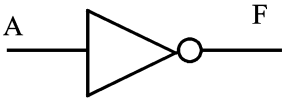


Wire crossing and joining

2.4 Other logic gates

2.4.1 The inverter

Many logic systems may require the **inverse** of a function. An integrated circuit known as a logic inverter is available for this purpose. The symbol for an inverter is:



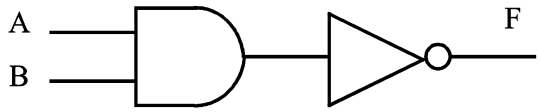
where

$$F = \overline{A}$$

F is the inverse of A, or any other logic signal which may be applied to the inverter's input. Manufacturers usually package 6 of these inverters into the one integrated circuit. It is then known as a hex-inverter.

2.4.2 The NAND gate

An inverter may be placed immediately following an AND gate as shown in the diagram:



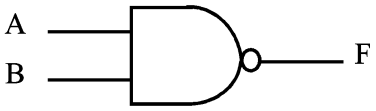
The equation is:

$$F = \overline{A \cdot B}$$

F is equal to the inverse of A and B. It is the NOT of the AND function

i.e. NOT-AND or NAND for short

The NAND function is often required in logic design, so manufacturers provide logic designers with a NAND gate. Its symbol is:



where $F = \overline{AB}$

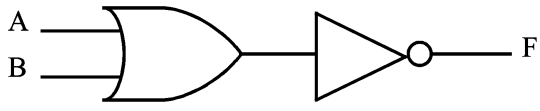
The truth table is:

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

You can verify for yourself that this is the inverse of an AND function.

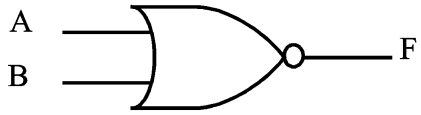
2.4.3 The NOR gate

A similar logic may be applied to the OR gate resulting in the NOT-OR or NOR gate.



where $F = \overline{A + B}$

A commercial NOR gate is available which has the symbol:



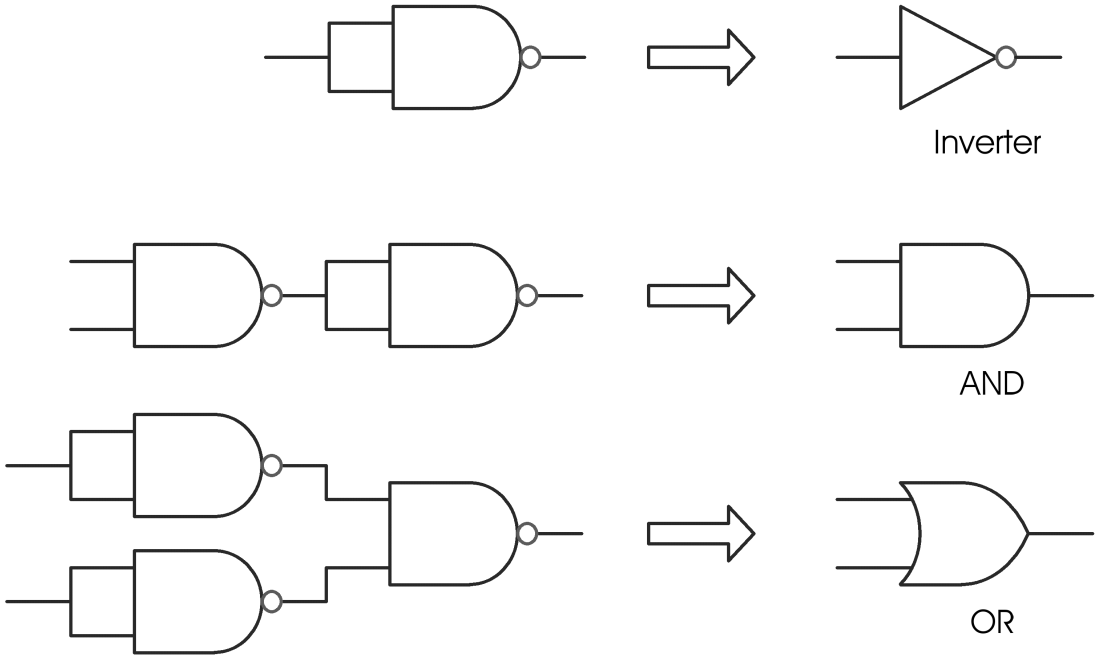
where $F = \overline{A + B}$ as before.

The truth table is:

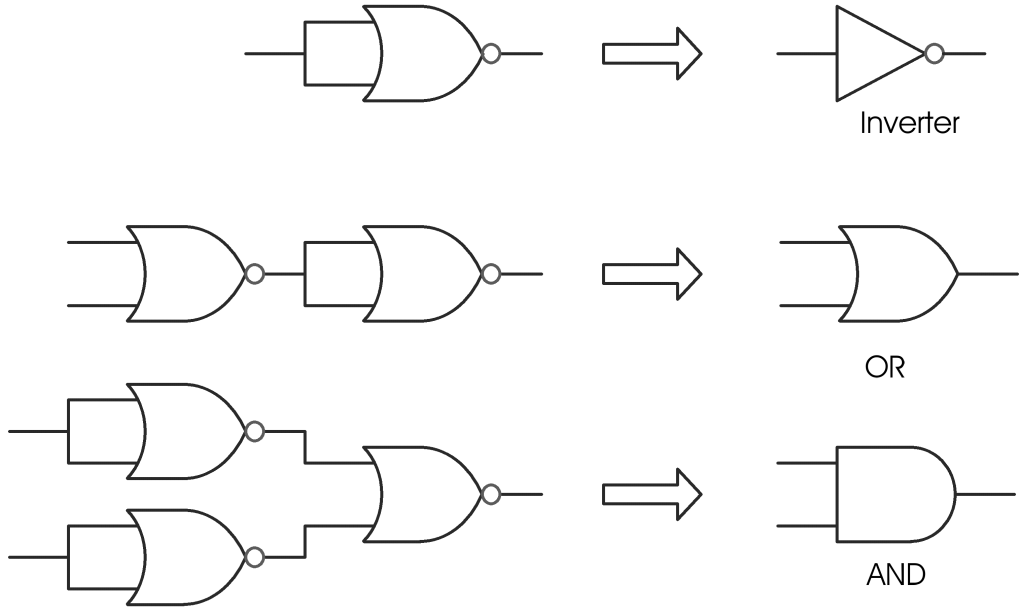
A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

2.4.4 Gate equivalents

The basic logic operations of AND, OR and INVERT can be implemented using only NAND gates if required. Using NAND gates only it is possible to reduce the chip (integrated circuit) count of a logic system. This is often done to reduce the overall cost of a logic design. While the following diagrams illustrate the NAND equivalent of basic logic operation, this method of conversion is not often used in practice. In a later module you will discover a more efficient method to convert a logic circuit to its NAND equivalent.



In a similar way NOR gates can also be used to implement AND, OR and INVERT operations.



2.4.5 Integrated circuit packages

Logic gates are manufactured as integrated circuit (IC) packages, with typically 14 pins. The 74 series is the cheapest and most popular today, and exist in many forms. A 14-pin dual-in-line (DIL) package may contain four, two-input, single-output gates, together with a ground and a positive power supply pin, as shown below.

Note, that as the number of gate inputs increase, the number of gates within the IC package must decrease. For example, a 3-input NAND gate requires 4 pins per gate (3 input and 1 output), therefore the 14-pin DIL package only has enough pins for 3 of this type of gate.

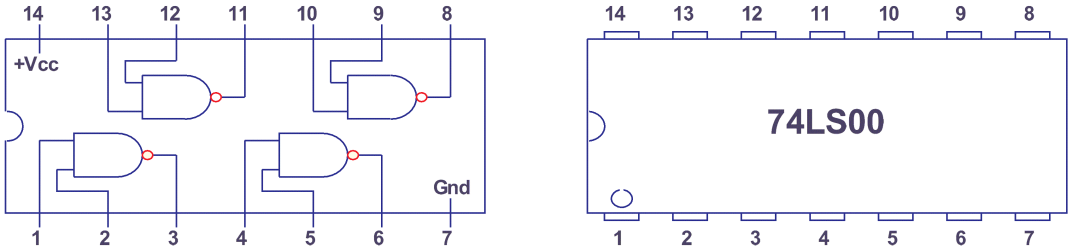
When designing logic circuits, as will be done in module 4, the designer must be aware of the number and type of gates per IC package in order to make good use of the available gates. To reduce the IC count it may be necessary to substitute logic gates with an equivalent in order to implement a specific logic function. In this way fewer gates are wasted and fewer IC packages are required.

ICs must be orientated correctly when placed in a circuit. Pin 1 is at the bottom left of the DIL package, marked with a circle or dot on the top of the device, or with an indentation on the left-hand side viewed from the top. Physically, all 14-pin DIL packages look the same. It is the device number that differentiates the internal logic functions.

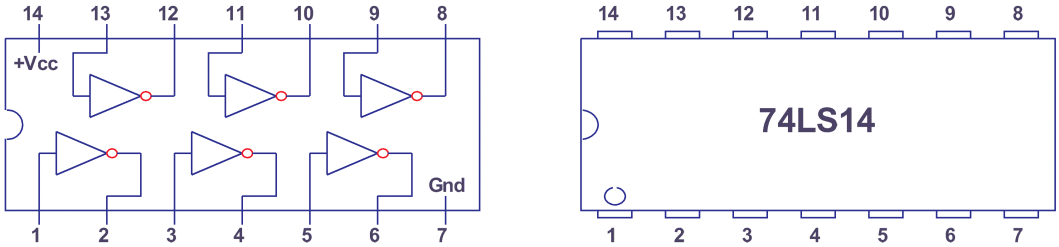
Pin-outs for a 74LS00, quad 2-input NAND integrated circuit

a. Location of logic circuits.

b. top view of the actual IC package.



Pin-outs for a 74LS14, hex inverter integrated circuit.

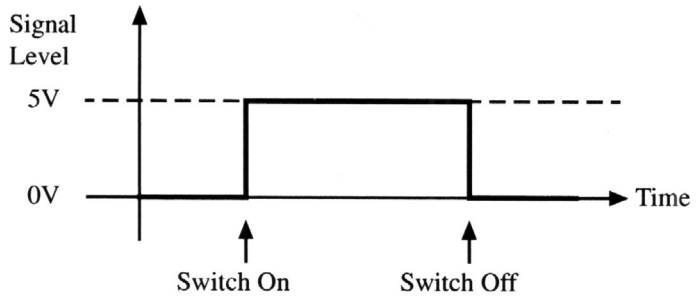


2.5 Switch bounce

It is important to consider here the mechanical operation of a switch and the effects it may have on high speed digital electronic circuits, such as logic gates.

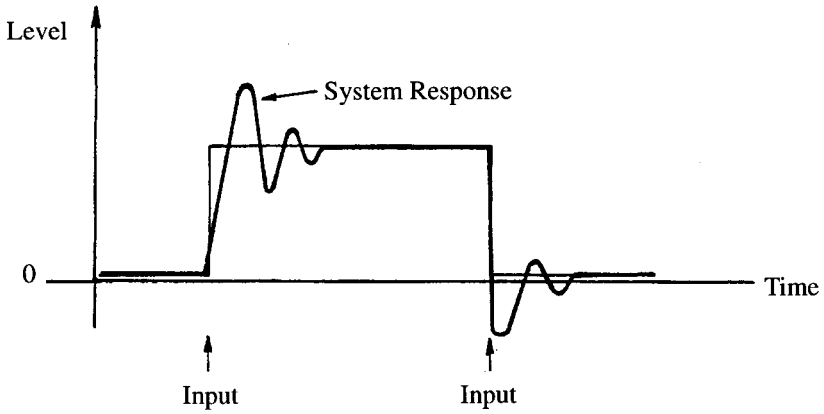
As we saw earlier a switch has contacts operated by a spring mechanism. From your knowledge of basic mechanics this mechanism consists of a mass (contacts) and spring assembly. When a mass connected to a spring has a force applied and removed (i.e. step input), it vibrates (i.e. oscillates) at a certain frequency until it comes to rest (i.e. damped oscillation). So a simple switch when operated 'on' or 'off', behaves as a **second order system**.

Graphically we can draw the ideal characteristics of a perfect switch as it is operated 'ON' and 'OFF' controlling a DC. electrical signal.

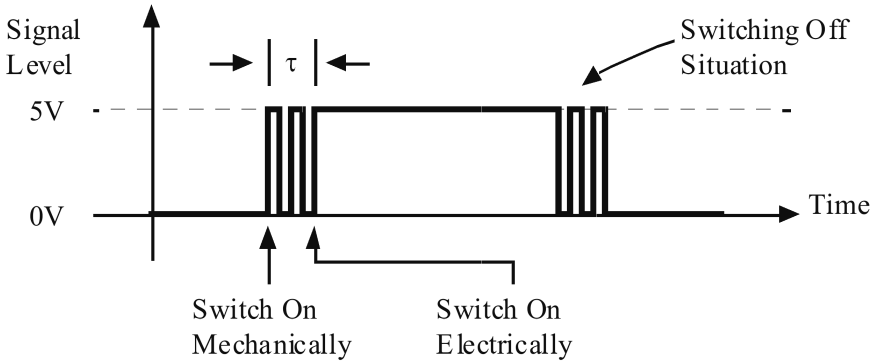


Perfect Switch Characteristics

If we now look graphically at the response of a second order system to a step input we have:



This oscillatory response actually causes the switch contacts to vibrate, or open and close rapidly for a short period each time the switch is operated. The resulting electrical signal is then:



It is clear there is a period of time (t), from when the switch is first activated until the mechanical oscillations (vibration) have been damped, when the switch output is uncertain. Practically the period of time (t) for most switches is very short, typically 15 to 20 ms. Note that time is relative to the situation being considered. 15 ms to a human is a very short period of time, however 15 ms to an electronic logic circuit (gate) is a very long time. A typical logic gate may be capable of switching at 15 MHz (15 million times a second). So you see if such a switch was connected to such a logic gate as just described, the logic gate would say to the switch 'make up your mind – are you 'on' or 'off'?

This problem has to be remembered whenever switches or other mechanical devices involving contacts are connected to computers. There are several ways to overcome this problem and we will consider some of them in a later module.

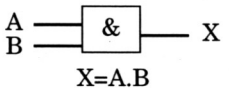
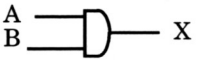
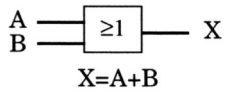
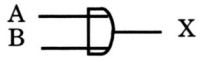
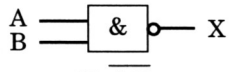
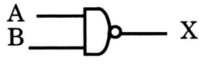
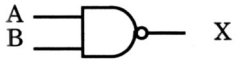
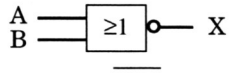
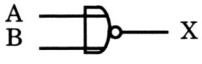
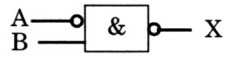
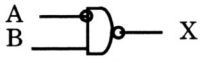
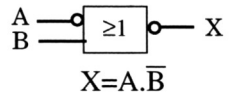
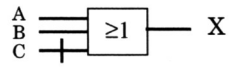
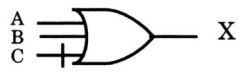
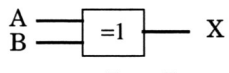
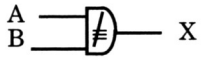
Activity 2.1

1. Explain the operation of AND and OR, NAND and NOR gates using switch analogies and truth tables.
2. Using diagrams describe the phenomenon known as switch bounce.
3. Refer to the ELE1301 course page on 'Study Desk' and complete the following experiments:
   a. Home experiment 2-1 – Switches, lamps and inverters
   b. Home experiment 2-2 – AND, OR, NAND and NOR gates.

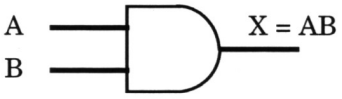
2.6 Logic diagram variations

Table 2.1 shows the variations between American and European standards for logic symbols.

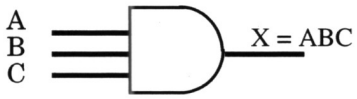
Table 2.1: Various logic symbols

Circuit	I.E.C. Proposal	DIN norm 40700	American Standard
AND	 X=A.B		
OR	 X=A+B		
NAND	 $X=\overline{A.B}$		
NOR	 $X=\overline{A+B}$		
NAND with one inverting input	 $X=A+\overline{B}$		
NOR with one inverting input	 $X=A.\overline{B}$		
INHIBIT GATE			
EXCLUSIVE OR	 $X=A\overline{B} + \overline{A}B$		

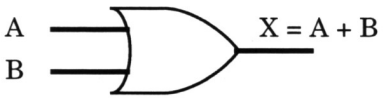
2.7 Common logic devices



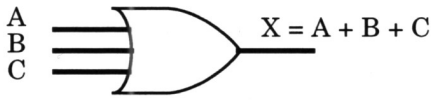
(A) Two-input AND gate



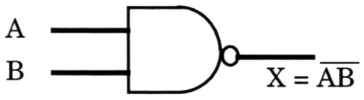
(B) Three-input AND gate



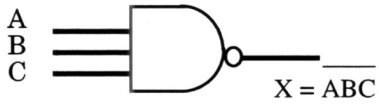
(C) Two-input OR gate



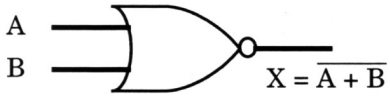
(D) Three-input OR gate



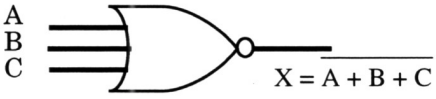
(E) Two-input NAND gate



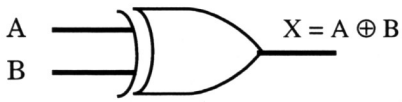
(F) Three-input NAND gate



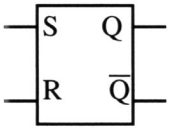
(G) Two-input NOR gate



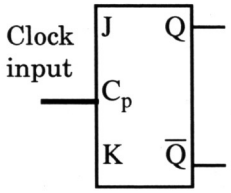
(H) Three-input NOR gate



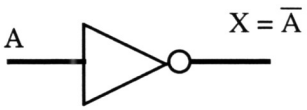
Schematic symbol for the exclusive OR function



SR Flip Flop



J-K Flip Flop



Inverter