LJ Technical Systems

An Introduction to Microprocessor Technology

DT100 Text Book

MP626/A

DiGiAC

Issue Number: MP626/A

Written by: LJ Technical Publications Dept.

LJ Technical Systems Ltd.
Francis Way
Bowthorpe Industrial Estate
Norwich. NR5 9JA. England
Telephone: (01603) 748001
Fax: (01603) 746340

LJ Technical Systems Inc.
85, Corporate Drive, Holtsville,
11742-2007, New York, USA.
Telephone: (631) 758 1616
Fax: (631) 758 1788

About this Manual

For tomorrows' engineers and technicians, training in the use of microprocessor systems and the design of control tasks will be very important.

We see microprocessors used in almost every area of modern life. They control domestic appliances, automated Teller machines, VCRs, automobile engine management and braking systems and so on - the applications are endless. In addition to these less obvious uses, microprocessors dominate todays' working environment in the shape of the personal computer.

This teaching manual is intended for those students who are encountering microprocessors for the first time.

By following this manual, you will gain a basic understanding of Microprocessor Technology, which will enable you to continue your microprocessor studies using any of the following teaching manuals in the LJ range:

An Introduction to 6502 Microprocessor Applications.
An Introduction to 6502 Microprocessor Troubleshooting.
An Introduction to Z80 Microprocessor Applications.
An Introduction to Z80 Microprocessor Troubleshooting.
An Introduction to 68000 Microprocessor Concepts and Applications.
An Introduction to 68000 Microprocessor Applications.

As you work through each chapter you will be guided by a series of student objectives and your progress will be assessed by questions in the Student Assessments.

Your instructor has a copy of the Solutions book for this manual. It contains all the solutions to the assessment questions.

Computerized Assessment of Student Performance

If your laboratory is equipped with the ***DIGIAC 3000*** Computer Based Training System, then the system may be used to automatically monitor your progress as you work through this manual.

If your instructor has asked you to use this facility, then you should key in your responses to the questions in this manual at your computer managed workstation.

To remind you to do this, a  symbol is printed alongside questions which require a keyed-in response.

The following D3000 Lesson Module is available for use with this manual:

D3000 Lesson Module 8.10

Contents

Chapter 1 Basic Systems

Objectives of this Chapter

Having studied this chapter you will be able to:

- Define a System.
- Explain the difference between an Input signal and an Output signal.
- State the Inputs, Outputs and Processes involved in three examples of electronic systems.

Introduction

A system can be defined as an orderly grouping of physical or abstract objects which perform a definite function.

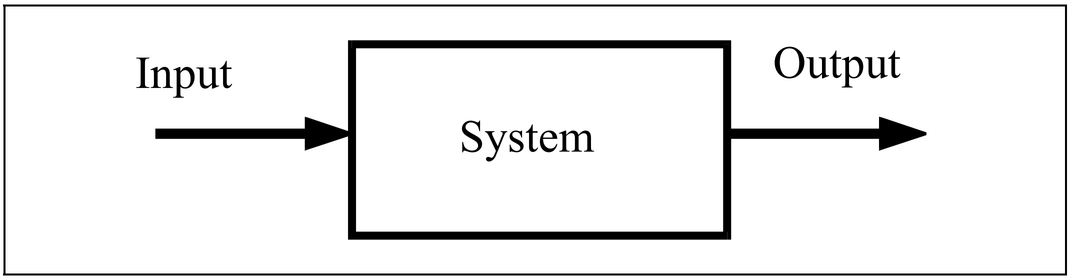
Examples of Physical Systems:

- Electrical Systems (eg: Television Receiver)
- Mechanical Systems (eg: Automobile)
- Biological Systems (eg: Human Body)

Examples of Abstract Systems:

- Economic Systems (eg: Monetary System)
- Political Systems (eg: Democracy)

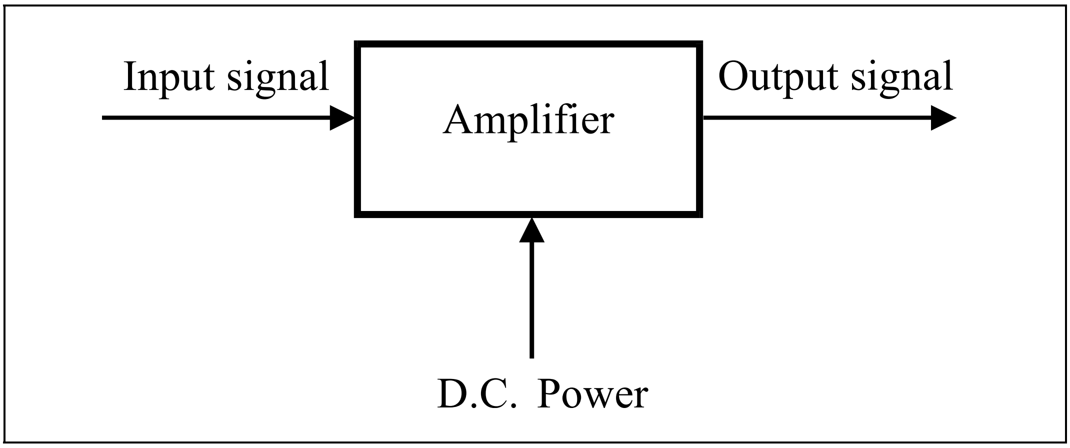
All types of Processing Systems will have both Input and Output signals or quantities.



In general input signals are passed **into** a system from the external environment. Output signals are passed **from** a system into the external environment.

The system processes the input or inputs to modify the output or outputs if necessary. Systems may also process their inputs to actually modify the type of process performed by the system.

Consider a simple Voltage Amplifier System:



The inputs for this system are:

- The input signal voltage.
- The dc power input.

Note: In many systems the power available at the input is too small to allow the system to function as required. Thus many systems require a **power** input, in addition to the other inputs.

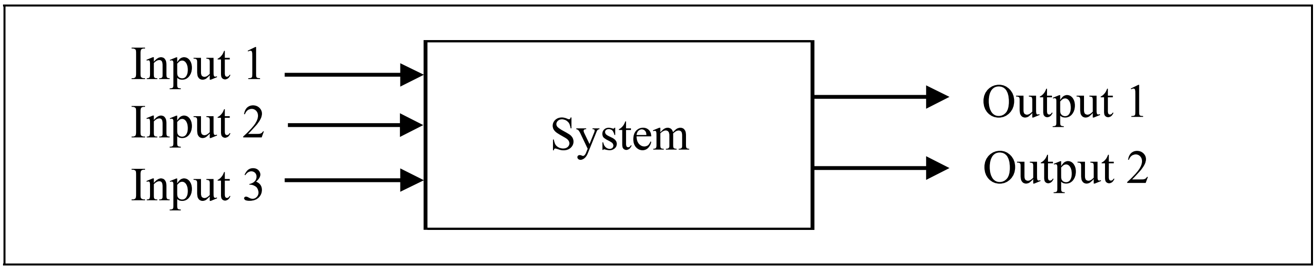
The outputs for this system are:

- The output signal voltage.

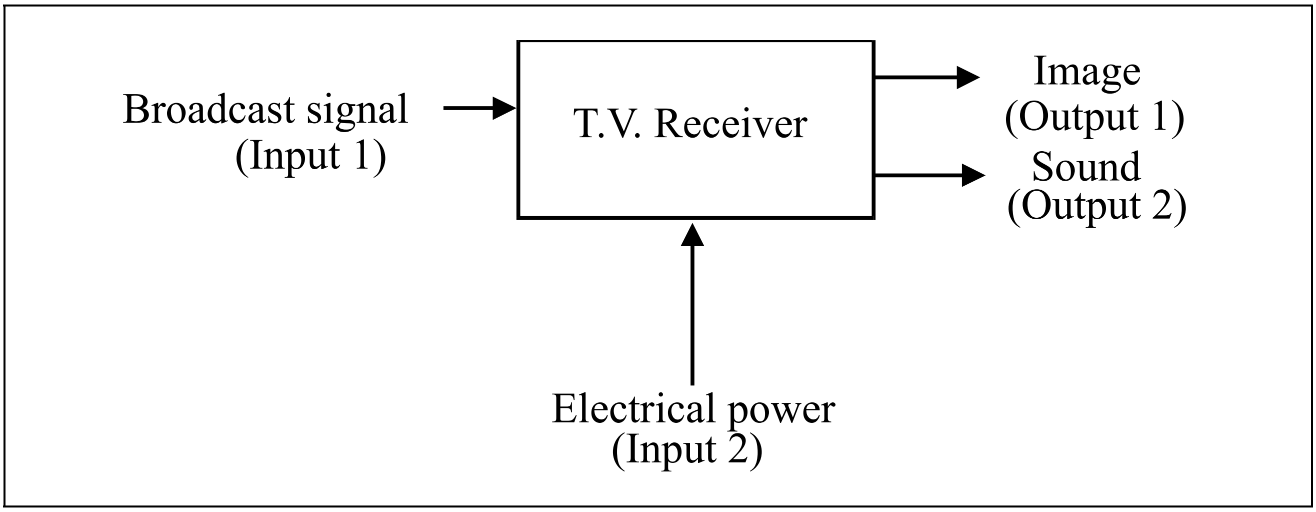
The process which this system performs is:

- The input signal controls the dc power to produce a magnified version of the input signal at the output.

Sometimes systems may have **more than one** Input or Output:



A television receiver is an example of such a system:



The LJ Microcomputers are also examples of systems. They will process inputs and change outputs as necessary. All LJ microcomputers will require electrical power, in the form of a +5V dc supply input. Keypads can be used to input data, whilst 7-segment LED displays or video monitors are used as data outputs. There are several other inputs and outputs which will be examined later.

Student Assessment 1

1. **An orderly grouping of physical or abstract objects which perform a definite function is called:**
   - a a process
   - b an input
   - c a system
   - d an output

2. **Signals which enter a system from the external environment are called:**
   - a process signals
   - b outputs
   - c amplified signals
   - d inputs

3. **Signals which leave a system for the external environment are called:**
   - a inputs
   - b power signals
   - c outputs
   - d switching signals

4. **In addition to the input signal voltage, the other input to a simple voltage amplifier system is the:**
   - a output signal voltage
   - b DC power input
   - c amplifier control signal
   - d image signal

Continued ...

Student Assessment 1 Continued ...

5. A television receiver is a system which has:

- a two inputs and two outputs
- b two inputs and one output
- c one input and one output
- d one input and two outputs

6. Which one of the following devices can <u>not</u> receive output data from a microcomputer:

- a video monitor
- b printer
- c LED display
- d keypad

Chapter 2 Transducers

Objectives of this Chapter

Having studied this chapter you will be able to:

- Explain what is meant by a Transducer.
- Explain the operation of different types of Transducer.
- Identify Transducers from their operation.

Introduction

Electrical systems need not necessarily have an electrical power input. For example: The power input to an electrical generator is mechanical energy (generated by a steam turbine, for instance).

Similarly the output of an electrical system need not be electrical. For example: The output from an electric motor is mechanical energy.

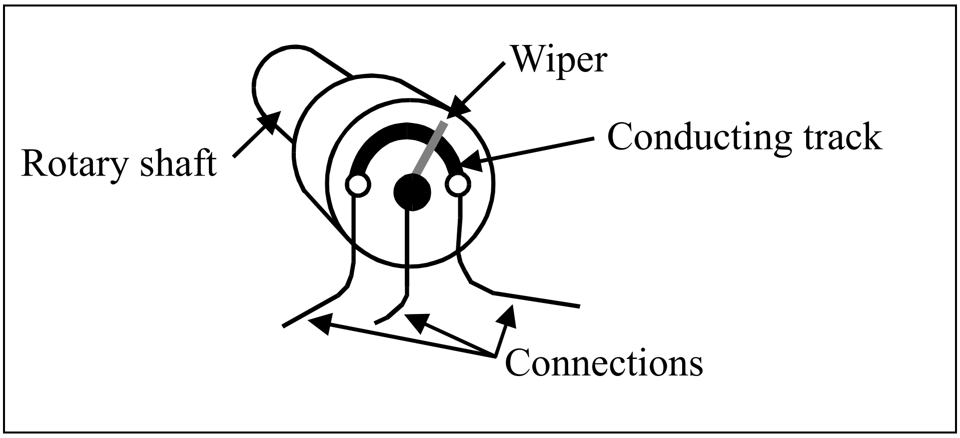
Many systems include devices which convert energy from one form to another. These are called **transducers**. There are a number of transducers on LJ Applications Modules.

Input transducers are often called **sensors**. Output transducers are usually called **output devices**.

There are many types of transducer. Some examples of transducers are given on the following pages.

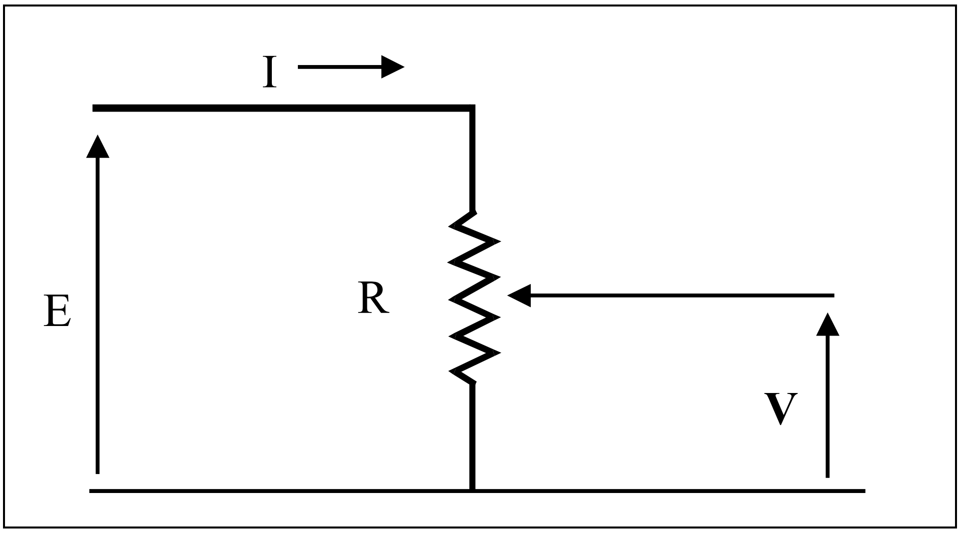
Potentiometer

These devices have a linear resistive track with which a movable wiper makes contact. There are potentiometers to be found in most LJ equipment.



The circuit symbol for a potentiometer is shown below. The arrow identifies the wiper.

If a supply voltage is applied to the track then the voltage at the wiper is directly proportional to the position of the wiper.

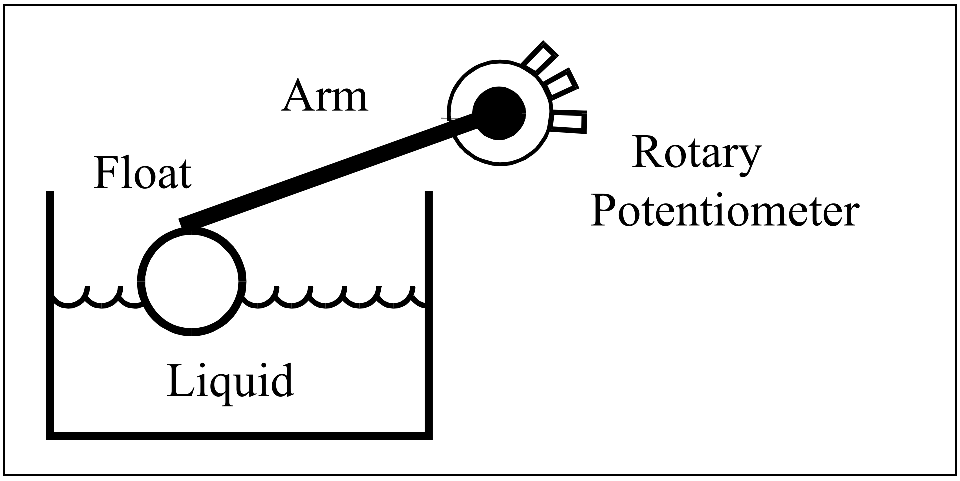


So a potentiometer can be used to measure a **shift** in **position** (or **displacement**).

Rotary potentiometers give an output voltage in proportion to **angular** displacement. **Linear** potentiometers give an output voltage in proportion to **linear** displacement.

The output voltage can be measured by a suitable voltmeter. This can be calibrated in terms of displacement rather than voltage. Alternatively, the output voltage can be used to control some other system or sub-system. Potentiometers can also be used to measure other physical variables **indirectly**.

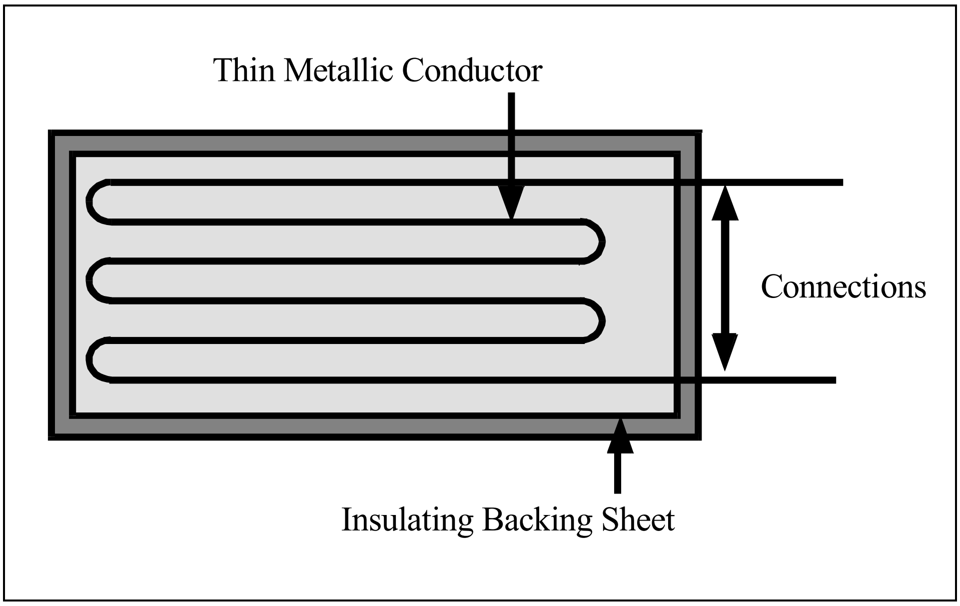
For example: The level of liquid within a tank is illustrated in the figure below:



As the level of the liquid increases, the float rises and so the shaft of the potentiometer is displaced. This will change the voltage at the wiper.

Strain Gauge

These sensors have a long, very thin metallic conductor mounted on a flexible backing sheet. The metallic conductor is folded to fit into a small area.



The gauge is bonded to a rigid body (eg: beam). Any bending of the body (**strain**) will also bend the gauge. This will stretch the conductor at various sites, changing the cross-sectional area. Now, the **electrical resistance** of any conductor will vary with its cross-sectional area. So any mechanical strain in the body will produce a change in the electrical resistance of the gauge. Thus, the resistance of the gauge is a measure of the mechanical strain in the body.

The electrical resistance of the strain gauge can be measured by various electrical circuits (eg: Wheatstone Bridge). Often the output will be in the form of a **voltage**, which could be measured by a voltmeter. The voltmeter can then be calibrated in terms of mechanical strain. Alternatively, the output voltage can be used to control some other system or sub-system. Strain gauges can also be used to measure other physical variables **indirectly**.

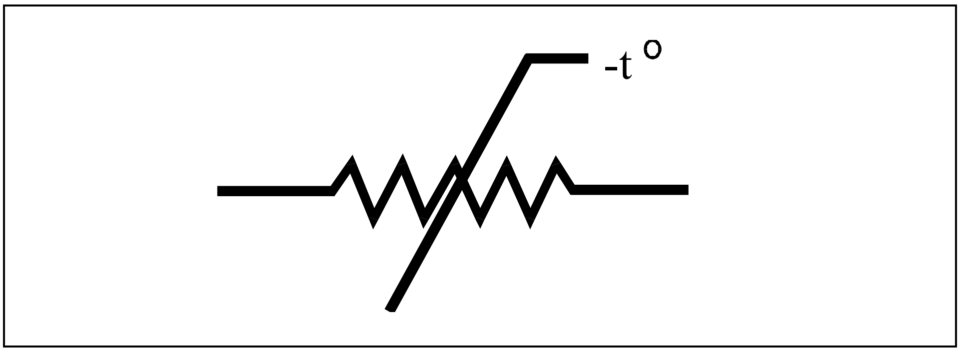
For example:

A diaphragm will stretch in response to increasing **pressure**. Strain gauges can be bonded to the diaphragm. The resistances of these gauges will change in response to a change in pressure.

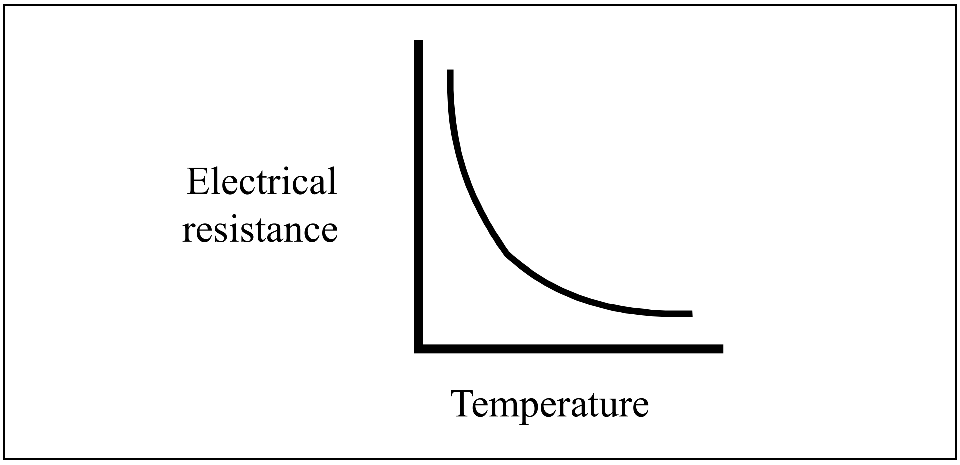
A Strain Gauge Module (Product Code MS7) is available from LJ Technical Systems.

Thermistor

A thermistor has an electrical resistance which changes significantly for a relatively small change in temperature. The circuit symbol for a thermistor is shown below:



Usually the electrical resistance will **decrease** as temperature rises. **This is called a Negative Temperature Coefficient (NTC).**



So a change in temperature will produce a change in the electrical resistance of the thermistor. In a similar fashion to the strain gauge, any change in resistance can be translated into a change in output voltage.

This voltage can then be displayed, using a voltmeter calibrated in degrees Celsius. Alternatively, the output voltage can be used to control some other system or sub-system.

Thermistors can also be used to measure other physical variables **indirectly**.

For example:

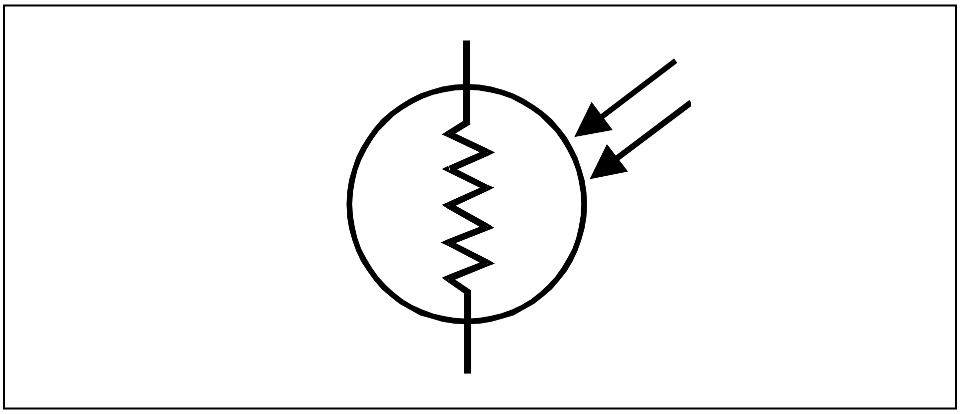
The cooling effect of a fluid flow will depend upon the **flow-rate**. The resistance of the thermistor will therefore change in response to a change in fluid flow-rate.

A Semiconductor Temperature Sensor module (Product Code MS11) is available from LJ Technical Systems.

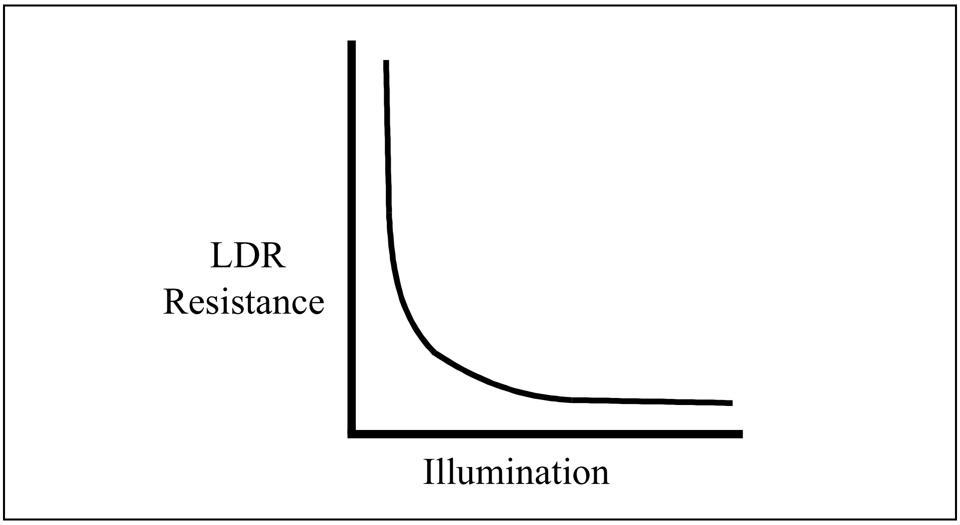
Photoconductive Cell

Photoconductive cells are often called Light Dependent Resistors (LDR), since the electrical resistance of the device will change with changing light intensity.

The circuit symbol for a photoconductive cell is shown below:



The electrical resistance of the LDR will decrease as the intensity of light rises.



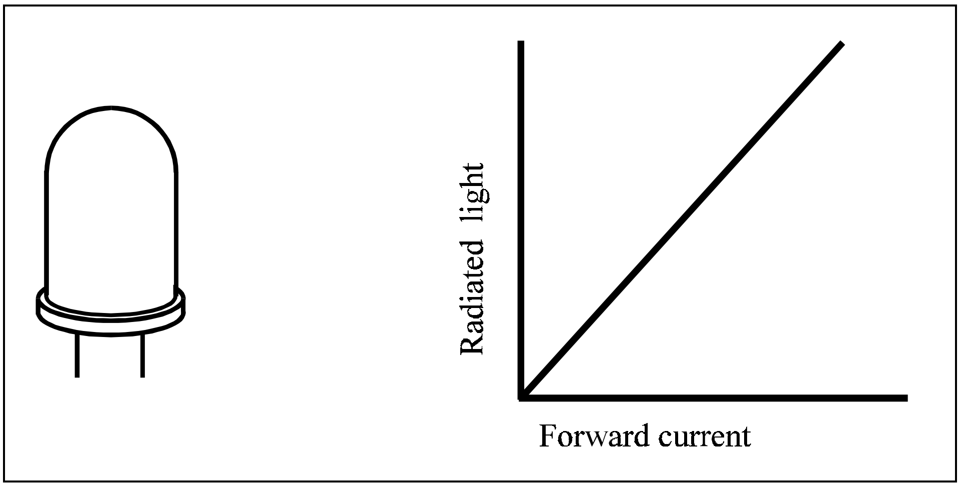
As with a number of other transducers, changes in resistance can be translated into a change in voltage. This voltage can then be displayed, using a voltmeter calibrated in degrees Celsius. Alternatively, the output voltage can be used to control some other system or sub-system. LDRs can also be used to measure other physical variables **indirectly**.

For example:

Some types of smoke detector use a constant light source and LDR arrangement. In the presence of smoke the light falling upon the LDR is reduced. The consequent rise in resistance can be measured or used to trigger an alarm circuit.

Light Emitting Diode (LED)

As electrical current passes through the LED, some energy is released in the form of visible or infra-red light. The color (ie. wavelength) of light emitted will depend upon the exact type of semiconductor material used but it is always of one particular color (ie. the range of wavelength is small). The brightness (ie. intensity) of the light emitted will be directly proportional to the electrical current flowing through the device.



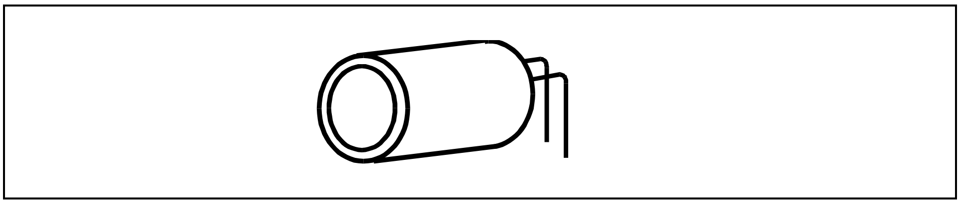
LED's are commonly used as indicator lamps and readout displays on a wide range of scientific and consumer equipment. LED's are also used as light sources in Optical Fiber Communications.

Many LJ products use LED's as indicators. Some of these use 7 - segment LED's. Each of these devices has seven bar - shaped LED's which can be illuminated separately to form numbers and some letters.

Some LJ products use LED's as optical transmission devices.

Phototransistor

The Phototransistor operates in the opposite way to an LED. Visible or infra-red light is converted into a small electrical current which operates a transistor. They are also only sensitive to light within a small range of wavelength.

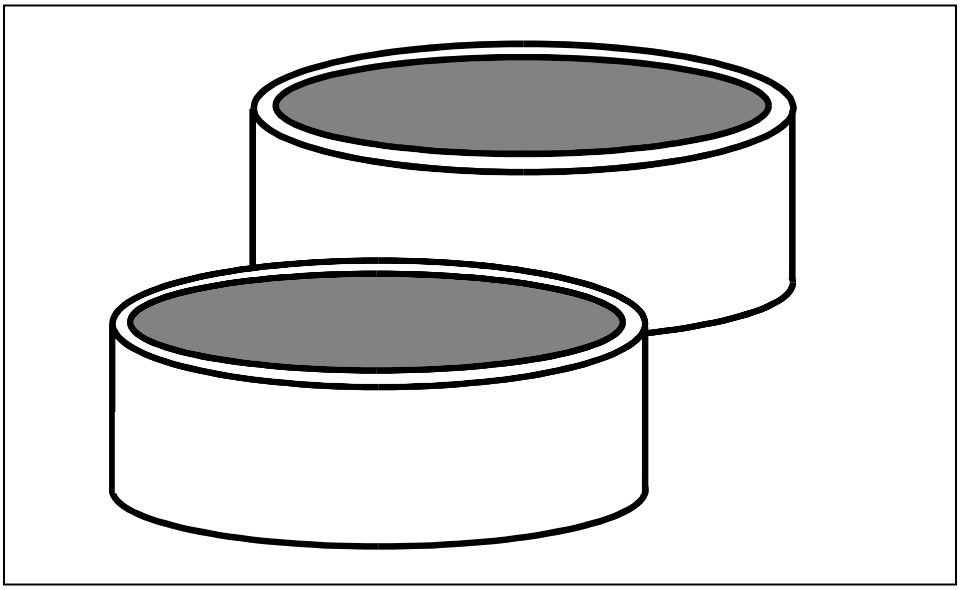


Phototransistors can be used in systems which switch according to the level of ambient light. For example: Street Lighting. They can also be used in conjunction with LED's to provide a means of detecting interruption of a light beam. This technique is used in applications such as process counters, smoke detectors and intruder alarms.

Phototransistors are used in a number of LJ products (eg. as an optical Receiver device).

Ultrasonic Transducers

Ultrasonic soundwaves have a frequency (or pitch) which is well above the range of human hearing. The highest frequency that can be heard by the human ear is about 20kHz. Ultrasonic transducers will normally operate at a frequency of about 40kHz.



An ultrasonic transmitter will generate these high frequency soundwaves which can then be detected by an ultrasonic receiver.

Ultrasonic transmitters and receivers are unaffected by ambient lighting conditions and so may be preferred to photo-electric devices in applications where this is a problem.

Student Assessment 2

1. **The type of device which converts energy from one form to another is called a:**
   - a semiconductor
   - b conductor
   - c transducer
   - d transponder

2. **The potentiometer can be used to directly measure:**
   - a displacement
   - b temperature
   - c current
   - d light intensity

3. **The transducer which converts electrical current into visible light of a specific wavelength is the:**
   - a phototransistor
   - b photoconductive cell
   - c light emitting diode
   - d thermistor

4. **A transducer which converts visible or infra-red light into an electrical current is a:**
   - a potentiometer
   - b phototransistor
   - c thermistor
   - d strain gauge

Continued ...

Student Assessment 2 Continued …

5. Ultrasonic transducers operate at a frequency:

a between 5kHz & 25kHz

b below that of human hearing

c of 4kHz

d above that of human hearing

6. Which transducer has its properties changed when it is bent?

a a thermistor

b a strain gauge

c a potentiometer

d a light emitting diode

Chapter 3 System Flowcharts

Objectives of this Chapter

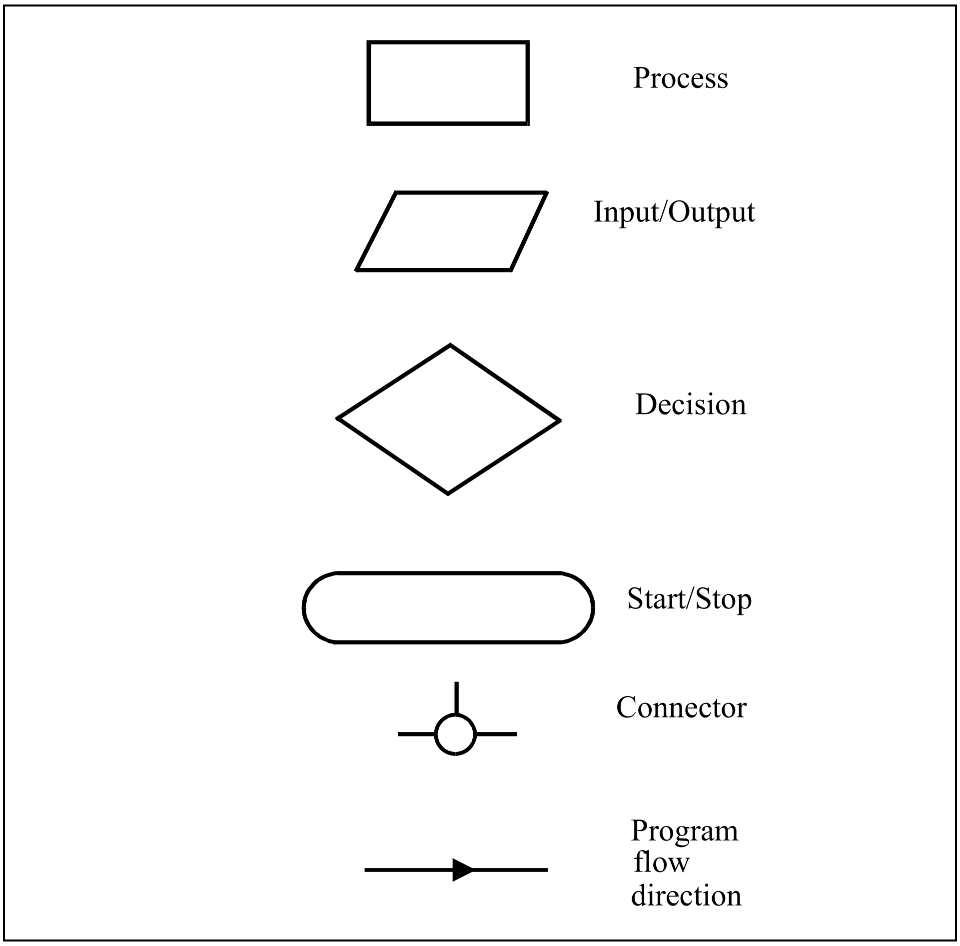
Having studied this chapter you will be able to:

- Explain how a flowchart can be used to describe processes in a system.

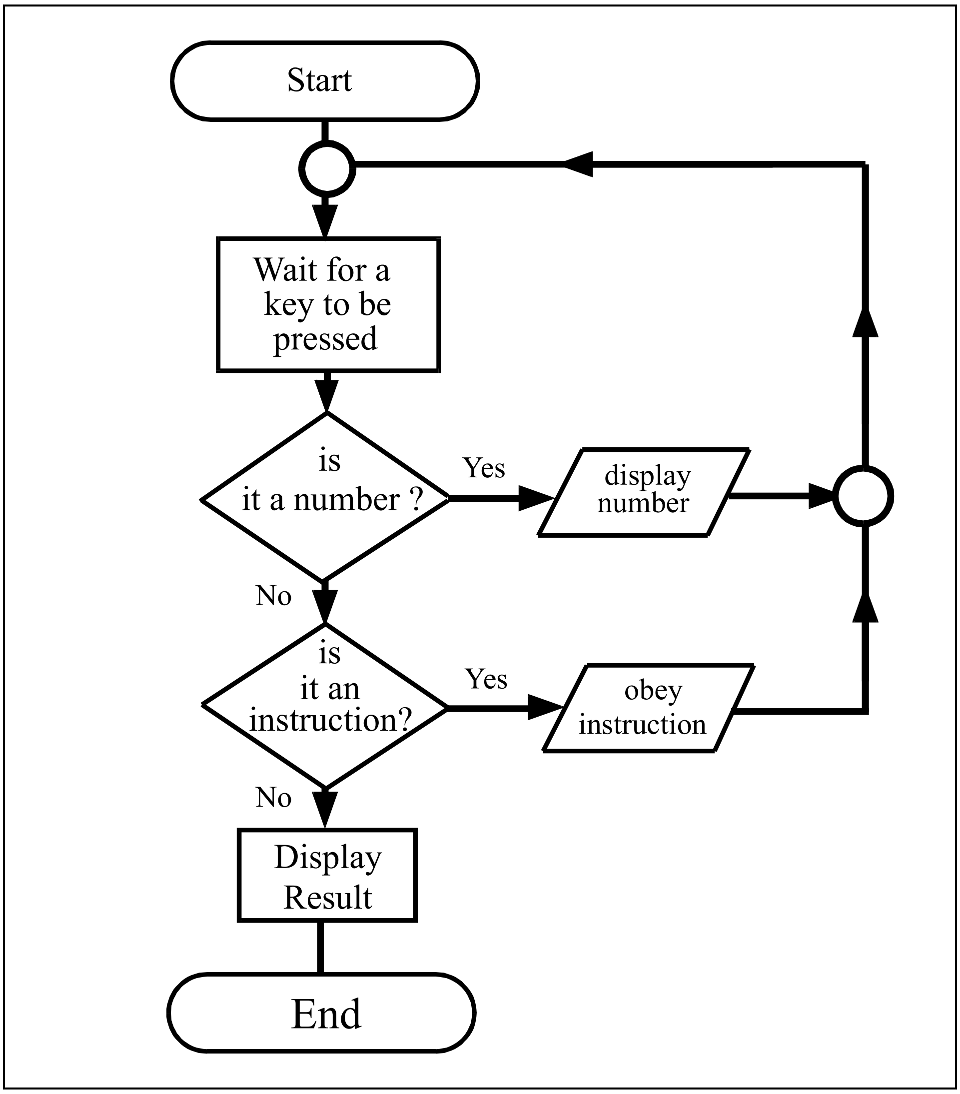
- State the meaning and use of symbols in flowcharts.

A flowchart is really a shorthand technique for describing the processes within a system. Flowcharts can also be used to describe computer programs.

Flowcharts use several symbols. The basic symbols are shown below:



A simplified flowchart which describes the operation of a pocket calculator is shown on the next page.



Notice that it is not necessary to ask the question "is it the '=' key ?" since if a key is neither a number nor an instruction then it must be '='.

So, flowcharts clearly show the order of operations to be carried out and the relationships between different parts of a system.

Student Assessment 3

1. A flowchart is used:

a to describe processes within a system

b because it is easier to follow than a written program

c because it looks nice

d because it is difficult for people to understand

2. A typical use for a flowchart is to describe:

a how to read this text book

b the behavior of humans

c television programs

d computer programs

3. The operation represented by the flowchart symbol shown below is:

a A decision box

b a process

c an input/output

d start/stop

4. A decision symbol is used where:

a more than one outcome is possible

b only one outcome is ever possible

c there is always more than one input

d you have to end the program

Student Assessment 3 Continued ...

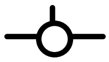
5. The symbol shown below is only used when:

[a] a decision is to be made

[b] a connection is to be made

[c] a program is about to end

[d] a process is being performed



6. If a flowchart symbol had the statement "Wait for a button to be pressed", the type of symbol used would be:

[a] input/output

[b] decision

[c] process

[d] procedure

Chapter 4 Complex System Case Study

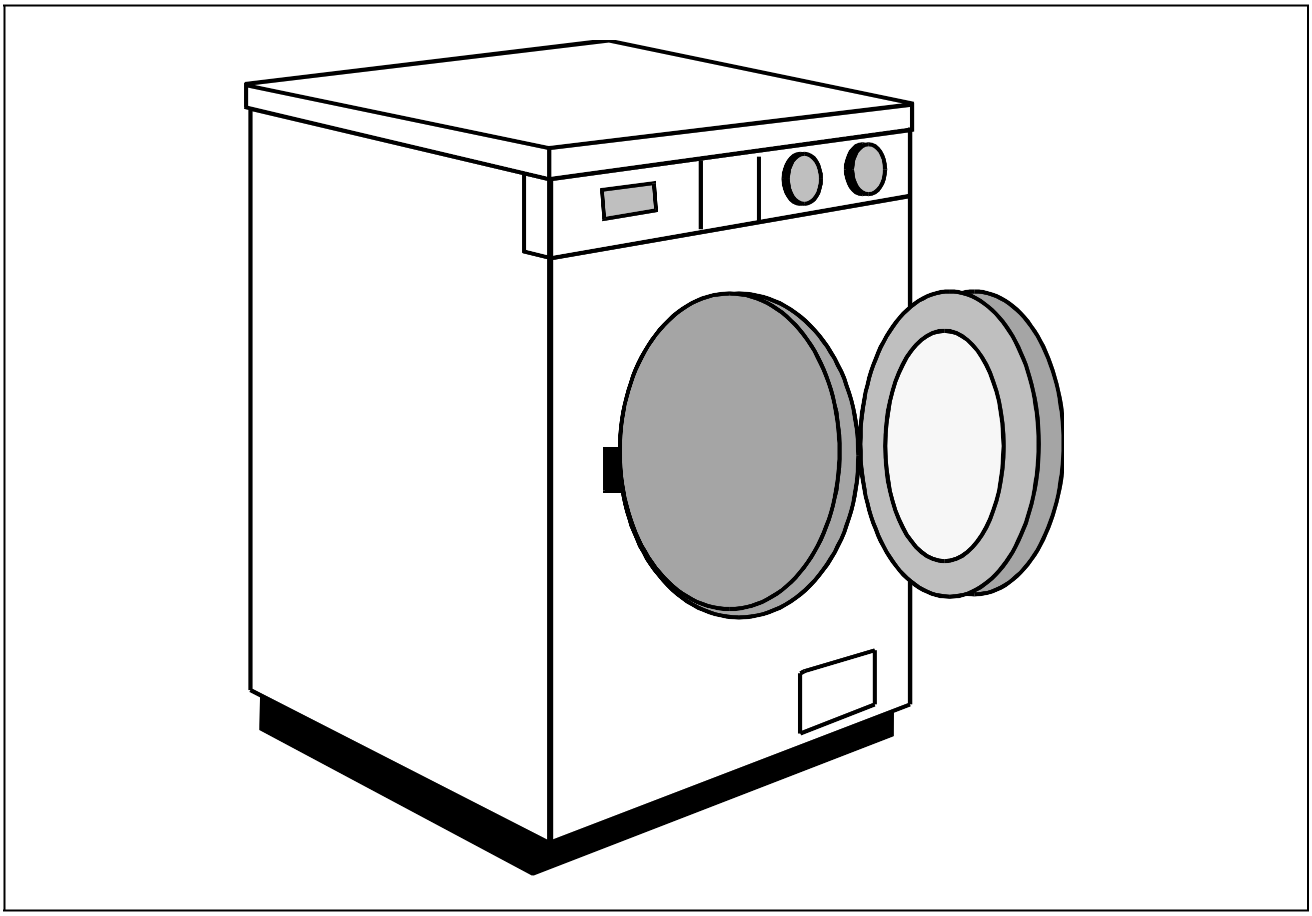
Objectives of this Chapter

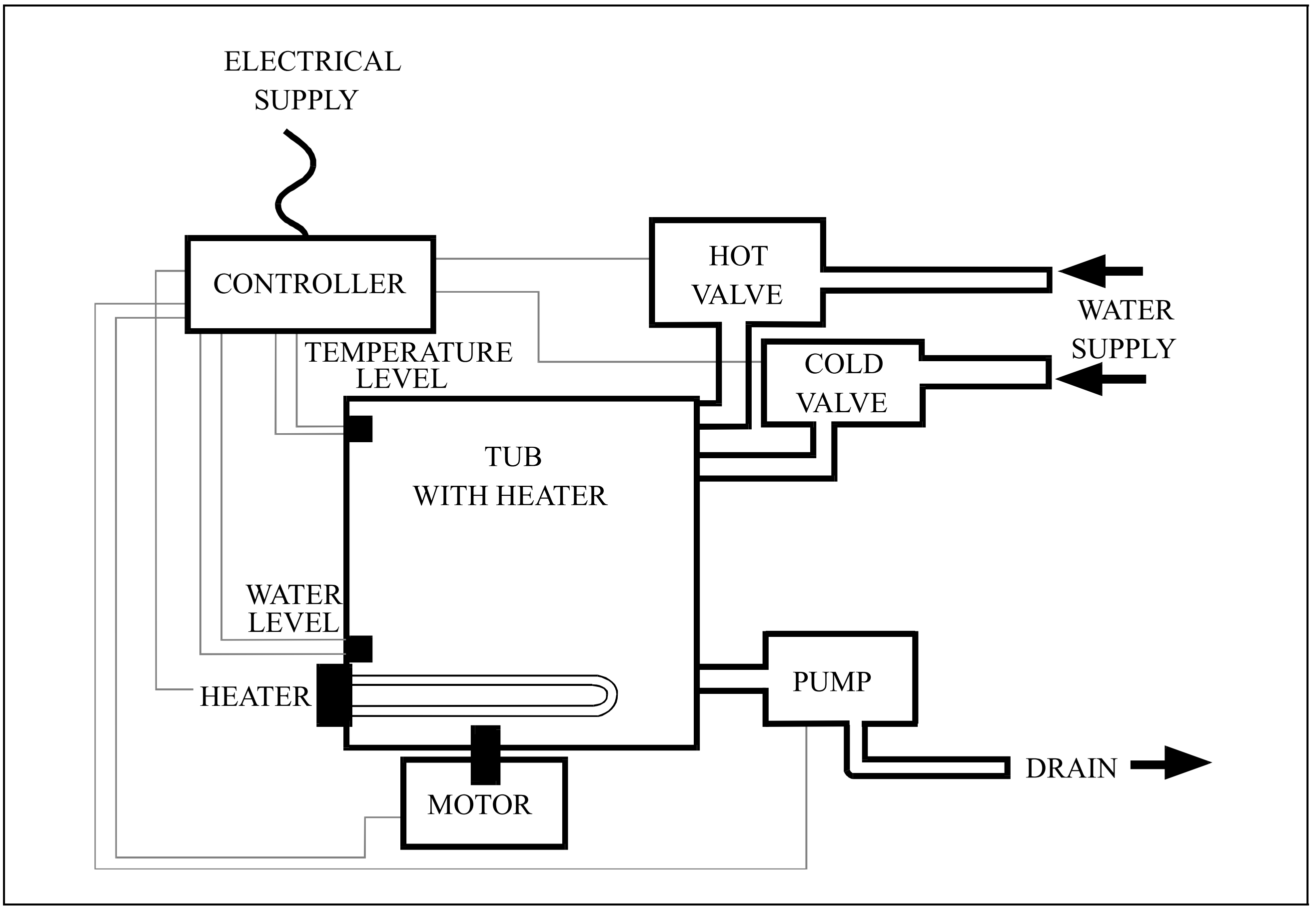
Having studied this chapter you will be able to:

- State examples of a system and sub-systems within a Complex System.
- For a Sub-System, state:
  Sub-System Inputs
  Sub-System Processes
  Sub-System Outputs
- Describe the sequence of events which occur within a Complex System.

Domestic Washing Machine System

A domestic washing machine is a familiar example of a system. A block diagram of a typical washing machine is shown on the next page.





Each of the sub-systems will have inputs, outputs and processes:

Hot Water Valve:

Inputs Hot Water
"Open"/"Close" Signal

Process Electrical signal is converted into mechanical force to "Open"/"Close" the valve.

Outputs Hot Water

Cold Water Valve:

Inputs Cold Water
"Open"/"Close" Signal

Process Electrical signal is converted into mechanical force to open or close the valve.

Outputs Cold Water

Discharge Pump:

Inputs Dirty water from the tub
"Pump" Signal

Process Electrical signal is converted into mechanical force to pump water from the tub.

Outputs Dirty water

Tub With Heater:

Inputs Hot Water
Cold Water
Mechanical energy from the motor shaft
Electrical energy to the heater.

Process Mixes Hot and Cold water and switches heater on or off to wash at correct temperature. Washes and Spin-dries the load.

Outputs Dirty Water
"Temperature Correct "Signal
"Water Level Correct" Signal

Motor:

Inputs Electrical energy
"Fast"/"Slow" Speed Signal

Process Electrical Energy is converted to Mechanical Energy to agitate wash at the correct speed.

Outputs Mechanical Energy

Controller:

Inputs "Temperature Correct" Signal
"Water Level Correct" Signal
Electrical Supply
Preset Program.

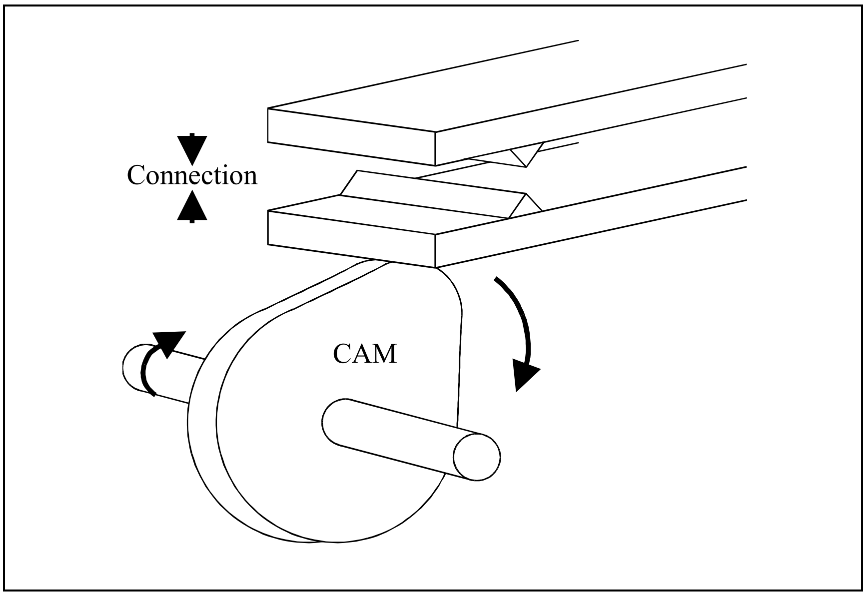
Process Switches Electrical Power to each Sub-System in the correct sequence to achieve the overall function.

Outputs "Open"/"Close" Hot Water Valve Signal
"Open"/"Close" Cold Water Valve Signal
"Operate Discharge Pump" Signal
Electrical Power to the heater element
"Motor Fast"/"Slow" Speed Signal

The overall process is to wash, rinse and spin-dry the load. To achieve this overall function, each sub-system must operate at the correct time and in the correct order. The controller determines this operating sequence. The operating sequence can also be called a program.

In older or less sophisticated washing machines the program is specified by a bank of cam-operated switches.

The cams are slowly rotated by a small electric motor. As the cam rotates, a point is reached where it will operate the switch. Each switch will usually operate one of the sub-systems (eg: Discharge Pump). So, each cam/switch set can be thought of as an instruction within the washing machine control program.



In the newer, more elaborate washing machines the simple electro-mechanical controller is usually replaced by a **microelectronic** controller. Such controllers are potentially much more versatile and reliable.

Washing Machine Operating Sequence

The operating sequence of the washing machine sub-systems can be thought of as a **program**. This program will have 3 major steps:

- **Wash**
- **Rinse**
- **Spin Dry**

Each of these steps will contain a number of much simpler steps. So, the complete program will comprise a great number of very simple steps which follow in a logical sequence.

The controller will contain the program and ensure that each step is carried out in the correct order and for the correct length of time.

WASH:

- Open Hot Water Valve
- Close Hot Water Valve when water reaches desired level
- Open Cold Water Valve
- Close Cold Water Valve when water reaches desired level
- Switch on Heater
- Wait until temperature is correct
- Switch off Heater
- Start Motor at Wash Speed
- Wait until Washing Time has expired
- Stop Motor
- Switch on Discharge Pump
- Wait for tub to empty
- Switch off Discharge Pump

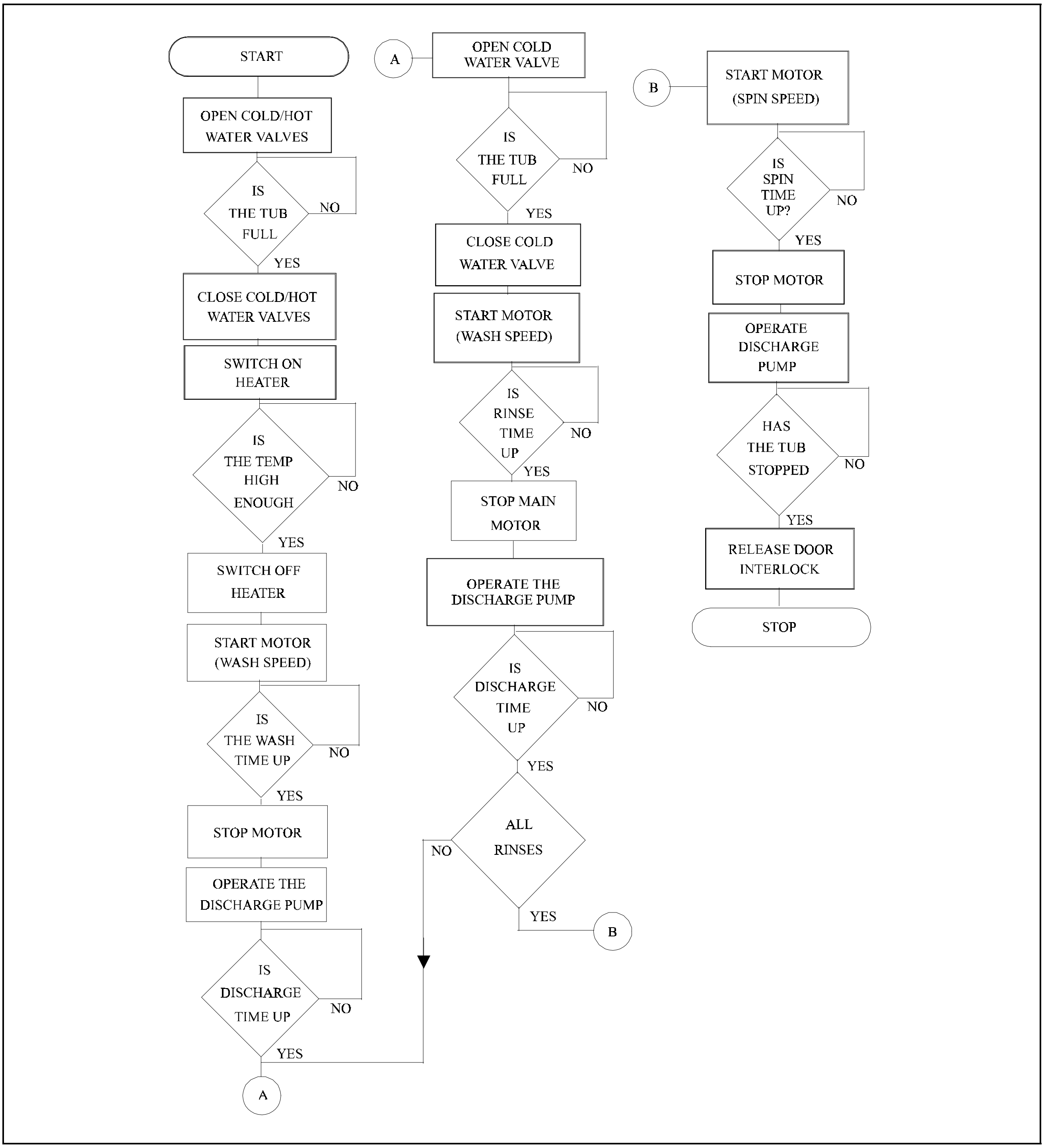
RINSE:

- Open Cold Water Valve
- Close Cold Water Valve when water reaches desired level
- Start Motor at Wash Speed
- Wait until Washing Time has expired
- Stop Motor
- Close Cold Water Valve when water reaches level
- Switch on Discharge Pump
- Wait for tub to empty
- Switch off Discharge Pump
- Repeat all "Rinse" instructions until required number of Rinses have been completed

SPIN DRY:

- Start Motor at Spin Speed
- Switch on Discharge Pump
- Wait for tub to empty
- Switch off Motor
- Switch off Discharge Pump

This method of describing the operation of a program is both difficult to follow and time-consuming to write. It is much easier in both these respects if the program is presented in flowchart form. A flowchart for the operation of a washing machine is shown on the next page.


START
OPEN COLD/HOT WATER VALVES
IS THE TUB FULL
NO
YES
CLOSE COLD/HOT WATER VALVES
SWITCH ON HEATER
IS THE TEMP HIGH ENOUGH
NO
YES
SWITCH OFF HEATER
START MOTOR (WASH SPEED)
IS THE WASH TIME UP
NO
YES
STOP MOTOR
OPERATE THE DISCHARGE PUMP
IS DISCHARGE TIME UP
NO
YES
A
A
OPEN COLD WATER VALVE
IS THE TUB FULL
NO
YES
CLOSE COLD WATER VALVE
START MOTOR (WASH SPEED)
IS RINSE TIME UP
NO
YES
STOP MAIN MOTOR
OPERATE THE DISCHARGE PUMP
IS DISCHARGE TIME UP
NO
YES
ALL RINSES
NO
YES
B
B
START MOTOR (SPIN SPEED)
IS SPIN TIME UP?
NO
YES
STOP MOTOR
OPERATE DISCHARGE PUMP
HAS THE TUB STOPPED
NO
YES
RELEASE DOOR INTERLOCK
STOP

Student Assessment 4

1. A domestic washing machine is an example of a:

a System

b Process

c Sub-system

d Sub-system process

2. An example of a sub-system would be:

a the washing machine

b the hot water valve of a washing machine

c the wash cycle of a washing machine

d the electrical supply to a washing machine

3. In a washing machine, the "Temperature Correct" signal would be an output of:

a the pump

b the controller

c the tub

d the hot water valve

4. An electrical signal converted to a mechanical force is an example of:

a a sub-system process

b a sub-system input

c a sub-system output

d a sub-system program

Student Assessment 4 Continued ...

5. In the flowchart shown earlier for a washing machine program, the first decision to be made in the program is:

a open the water valves?

b is the tub full?

c is the water temperature correct?

d start the motor?

6. Many years ago, program instructions were operated by:

a levers and pulleys

b cam/switch sets

c microswitches

d microelectronic controllers

7. The controller is an example of:

a a system

b a process

c a sub-system

d a sub-system process

Chapter 5 Analog and Digital Systems

Objectives of this Chapter

Having studied this chapter you will be able to:

- Explain the difference between Analog and Digital Signals

- Give examples of :

  Analog Systems
  Digital Systems

- Describe the operation of :

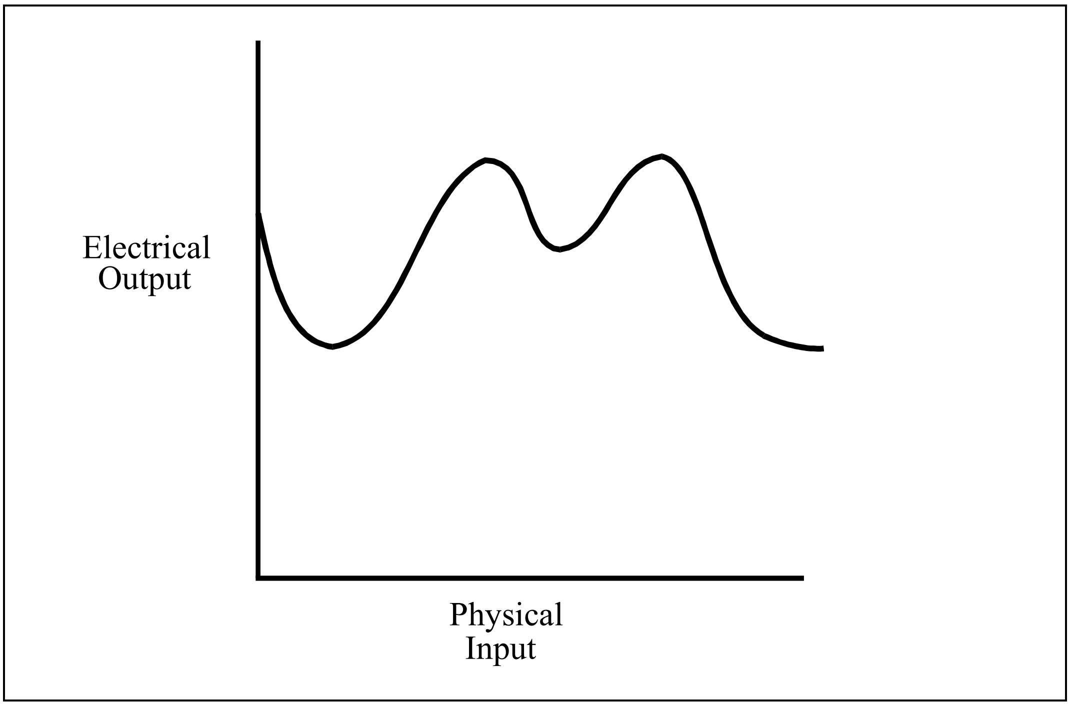
  An Analog to Digital Converter
  A Digital to Analog Converter

Systems and Signal Parameters

Electronic signals can be classified as **Analog** or **Digital**, according to their method of processing.

Analog Signals

Analog signals may have **any value**, between defined limits and are therefore **continuous**. In Analog Electronic Systems, physical inputs (eg: temperature, pressure, etc.) are **represented** by a continuous value of voltage or current.



For example: The resistance of a Thermistor is analogous to temperature.

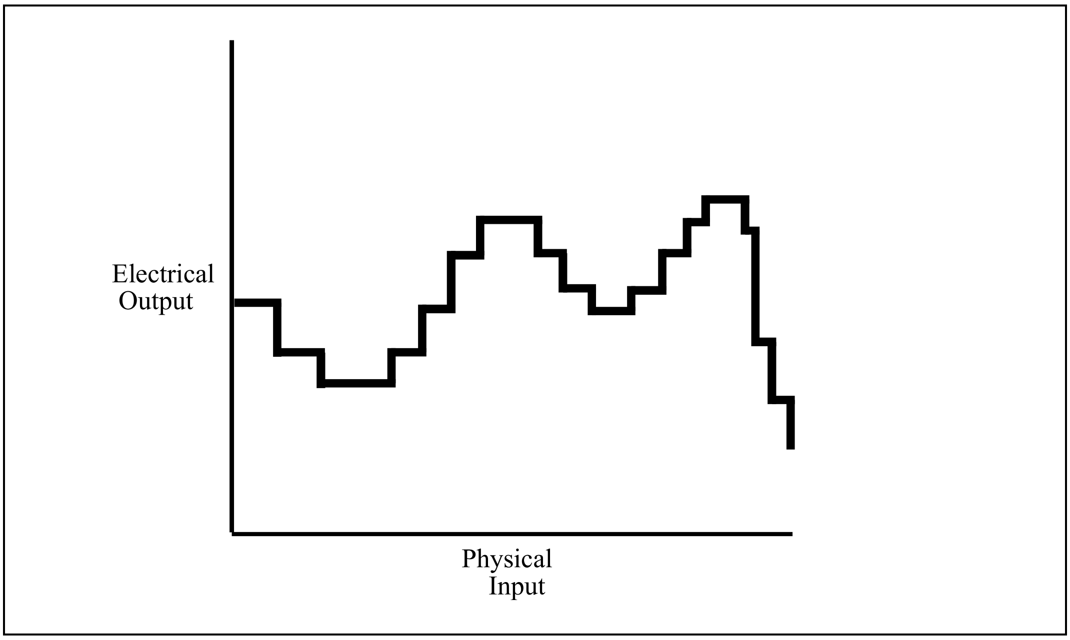
Other examples of Analog Systems are:

- **Mercury Thermometer**: Height of Mercury Column is **analogous** to Temperature.
- **Hi-Fi Amplifier**: Large Output power is **analogous** to Small Input Power.
- **Panel Meter**: Needle deflection is **analogous** to current.

There are many types of Analog Electronic Systems. This branch of Electronics is also called **Linear** Electronics.

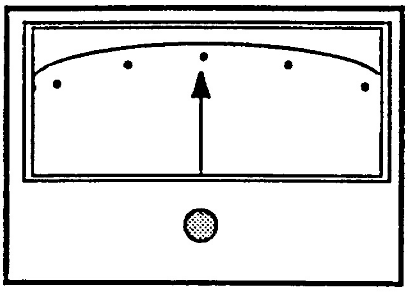
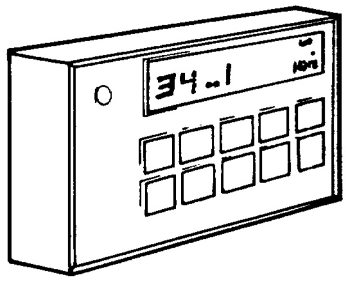
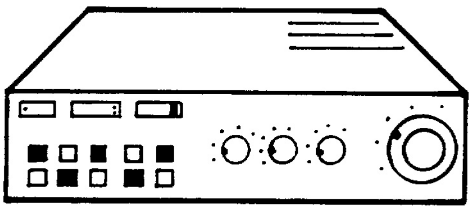
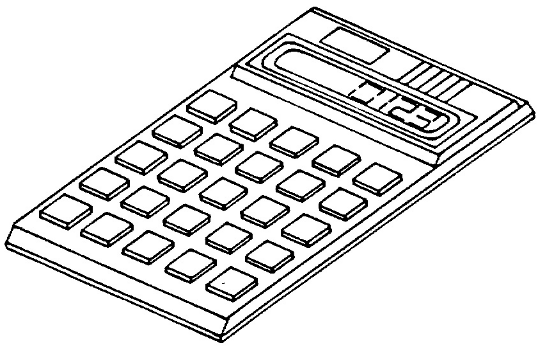
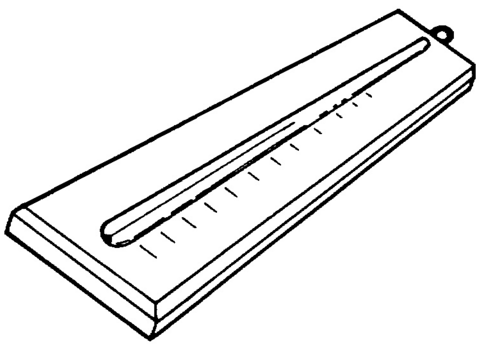
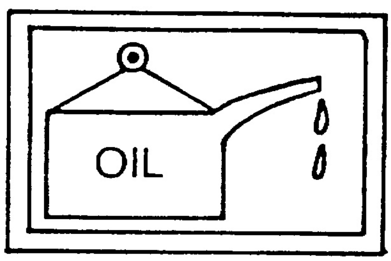
Digital Signals

Digital signals may only have one of a **fixed number** of values (often just **two**) and are therefore **discrete**. In Digital Electronic Systems, physical inputs (eg: temperature, pressure, etc.) are **represented** by discrete **steps** of voltage or current.



For example: A switch may only be in **one** of **two** possible states: "**off**" or "**on**". Other examples of Digital Systems are:

- **Thermostat**: Electrical Switch operates at a predetermined temperature.
- **Pocket-Calculator**: Digital Display shows result of last calculation. The degree of precision is limited by the number of display digits available.
- **Automobile Oil Pressure Warning**: Dashboard lamp is switched on by a pressure-sensitive switch at a predetermined pressure level.

Analog	**Digital**
 Panel Meter	 Electronic Thermostat
 Analog Amplifier	 Pocket Calculator
 Mercury Thermometer	 Oil-Pressure Warning Lamp

Now, many digital electronic systems only have two signal levels: "On" and "Off". These are usually represented by two **voltage levels** (commonly +5V and 0V).

For example:
Microcomputers comprise a large number of two-state devices called **"Flip-Flops"** and so will only process two-state signals.

Two-state systems will operate on **Binary Data**. This will be explained in Chapter 6.

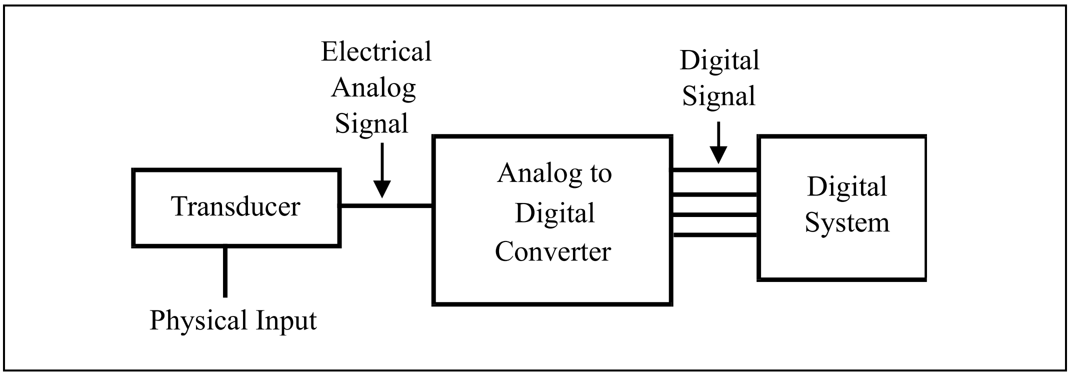
Digital signals may be **parallel** or **serial**. In a parallel system, each data word is transmitted or received in a single operation using a number of interconnections. In a serial system, each data word is transmitted or received using a single interconnection, over a period of time. So the transmission or reception of a serial data word requires a much longer time period.

Clearly then, parallel data communications will be much **faster**. However, for distances longer than a few meters the cost of parallel cable and the risk of data corruption outweigh the speed advantage and serial data communication is usually preferred.

Conversion Between Analog and Digital Signals

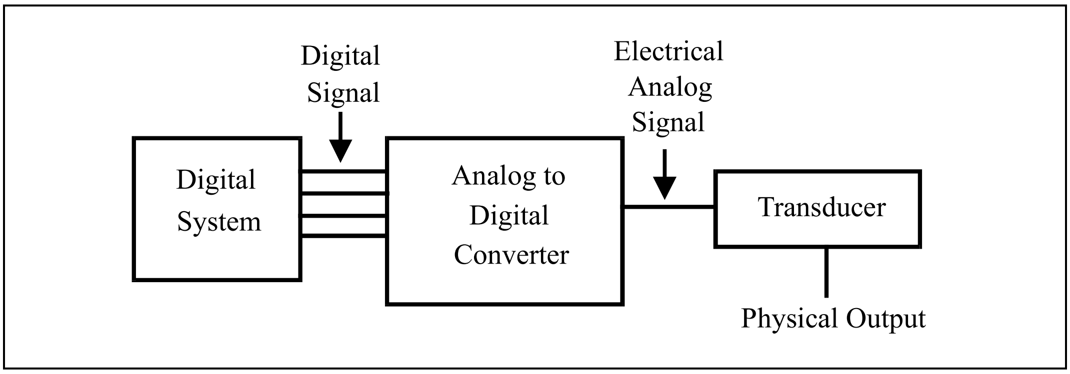
The majority of physical quantities will be essentially analog. For example: Heat, Pressure, etc. However, the microcomputer will only operate on digital signals. Clearly then, some means of converting between analog and digital signals is required for systems which have both types of signal.

Analog to Digital Converter (ADC) circuits take a continuous analog input voltage or current and produce an output which is a parallel binary code. This code can then be processed by the Digital System as shown on the next page.



Sometimes it is necessary for a microcomputer to produce an analog output. For example: To control a dc motor. In these situations a **Digital to Analog Converter** (DAC) can be used.

The DAC will take a parallel binary code and produce an output which is a continuous analog voltage or current. This voltage or current can then be used to provide the required analog output.



The LJ Applications Module (Product Code DT35) includes an A-D Converter and a D-A Converter. These can be used to convert data from one form to another since microcomputers can only operate on digital information. However, a number of the inputs and outputs on the devices which the microcomputer may communicate with can only handle data in an analog form.

Student Assessment 5

1. Signals which may only have a fixed number of levels are called:

- a analog signals
- b digital signals
- c continuous signals
- d linear signals

2. Analog signals are also known as:

- a discrete signals
- b continuous signals
- c stepped signals
- d curvaceous signals

3. An example of an analog system is a:

- a bar-code reader
- b microcomputer
- c transistor radio
- d pocket calculator

4. An automobile oil pressure warning lamp is controlled by a:

- a digital signal
- b analog signal
- c linear signal
- d continuous signal

Continued ...

Student Assessment 5 Continued ...

5. Why is a D-A converter required on a microcomputer?

- a Because a microcomputer can only read analog data
- b Because input devices only use digital signals
- c So that humans can understand what is going on
- d Some output devices may require analog signals

6. If a potentiometer was used as an input device, the device required by the microcomputer to read this data would be:

- a an A-D converter
- b a D-A converter
- c a transducer
- d a transistor

Chapter 6 Binary and Decimal Numbering Systems

Objectives of this Chapter

Having studied this chapter you will be able to:

- Convert a binary number into decimal
- Calculate the binary equivalent of a decimal number

Binary Numbering System

The ordinary everyday numbering system is the **decimal** (or **denary**) system. This operates on data which may be one of ten digits (0 to 9). This is a **weighted column** numbering system since the **position** of a digit within a **number** gives its **weight**.

Consider the decimal number 764: This can be written 764_{10} or 764_D to emphasize that it is **decimal** or **to the base ten**.

Hundreds	Tens	Units
7	**6**	**4**

So this number represents: 700 + 60 + 4

Numbers are made up of digits between 0 and 9, arranged in columns to represent orders of magnitude.

This type of system allows arithmetic to be relatively straightforward, since simple "carrys" and "borrows" can be transferred between columns.

The Binary Numbering System operates on data which may be one of **two** digits (0 **or** 1). Binary Digits are called "**Bits**" (**Bi**nary Digi**t**). The binary system also uses **weighted columns**. The position of a bit within a binary number gives it its weight.

In the Decimal system, columns represent increasing powers of **10** (1, 10, 100, 1000, 10000, etc.). In the Binary system, columns represent increasing powers of 2 (1, 2, 4, 8, 16, 32, 64, etc.).

Consider the binary number 101: This can be written 101_2 to emphasize that it is binary or to the base two.

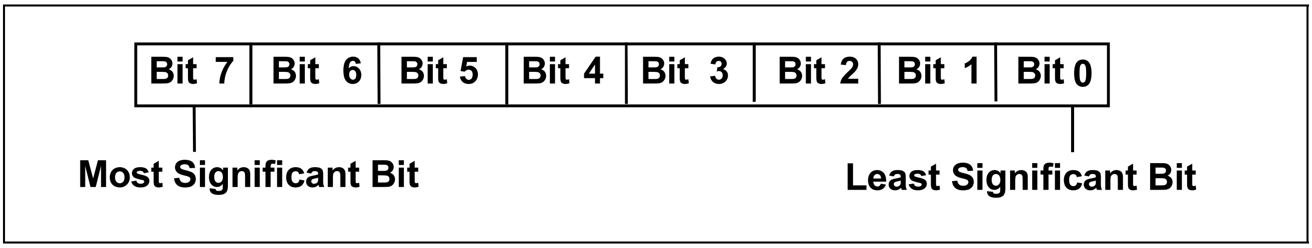
Fours	**Twos**	**Units**
1	**0**	**1**

So this number represents: $4 + 0 + 1 = 5_{10}$

Binary numbers are also made up of digits arranged in columns to represent orders of magnitude. However, each digit may only be a 1 or a 0.

Arithmetic in binary is essentially similar to decimal arithmetic. However "carrys" and "borrows" will occur for values exceeding 1 rather than those exceeding 9.

As previously stated, a single Binary Digit is called a **bit**. Microcomputers may operate on 8, 16 or 32 bits at a time. A group of eight bits is called a **Byte**. Conventionally, the bits within a byte are numbered from 7 to 0 thus:



A group of 4 bits is referred to as a nibble. So one byte is made up from two nibbles.

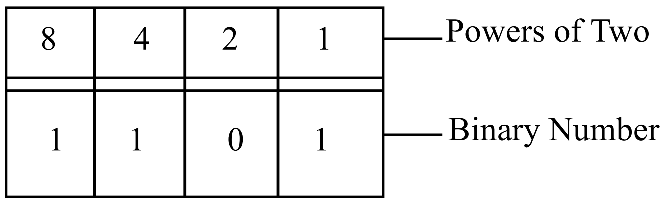
Conversion between Binary and Decimal

Clearly it will be necessary to convert from decimal to binary and vice-versa. There are a number of techniques available but by far the most straightforward and least prone to error is to use **Weighted Columns**.

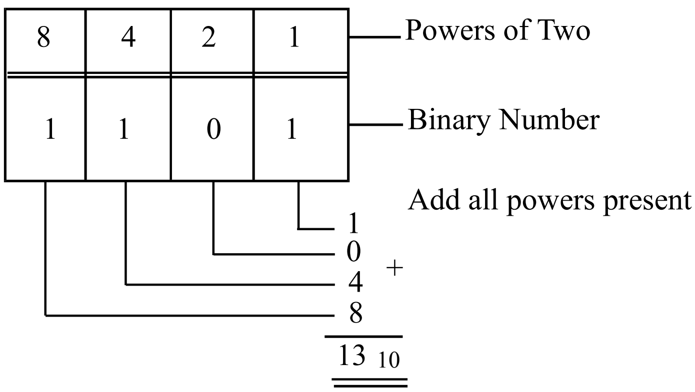
For example: To convert 1101_2 to decimal:

Binary to Decimal Conversion

The binary number to be converted is placed in its weighted columns thus:



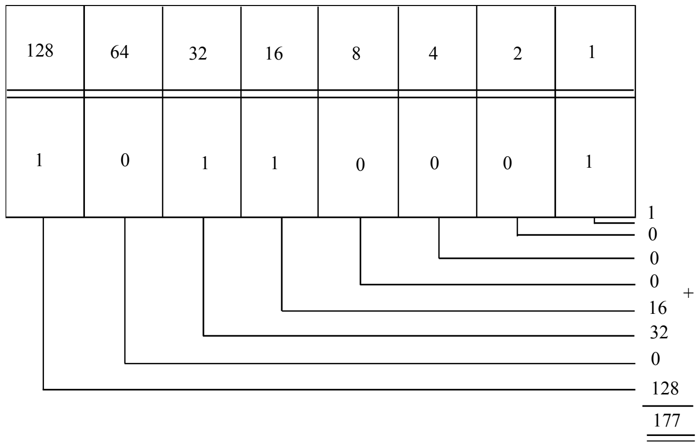
The powers of two which are present (8, 4 and 1 in this case) **are then added**. The sum of these powers gives the decimal equivalent thus:



So the decimal equivalent of $\mathbf{1101_2}$ is $\mathbf{13_{10}}$.

Larger binary numbers may be converted by simply extending the number of weighted columns.

For example: To convert 10110001_2 to decimal:

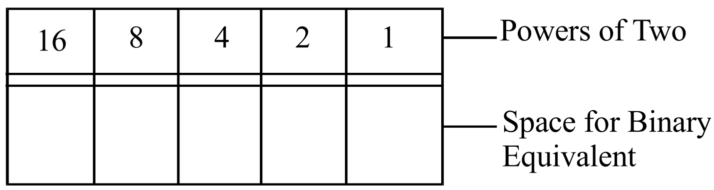


So the decimal equivalent of $\mathbf{10110001_2}$ is $\mathbf{177_{10}}$.

Decimal to Binary Conversion

For example:
To convert 18_{10} to binary:

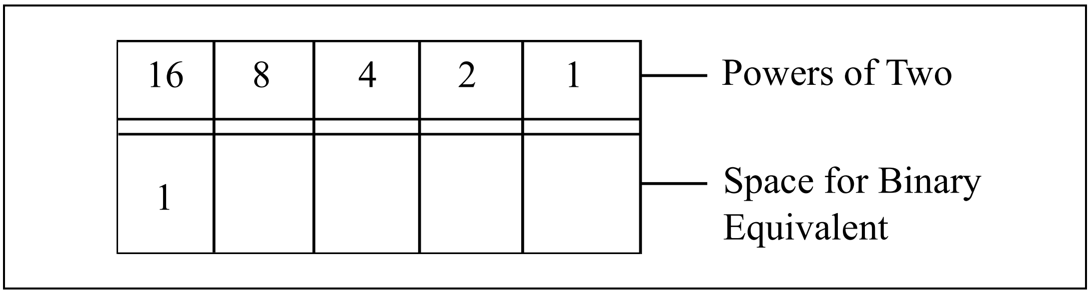
The weighted column technique can again be used. In this case however a number of blank columns are first drawn thus:



The decimal number is then examined to determine the highest power of two it contains. In this example the highest power is 16 (the next highest is 32 which is **greater** than 18).

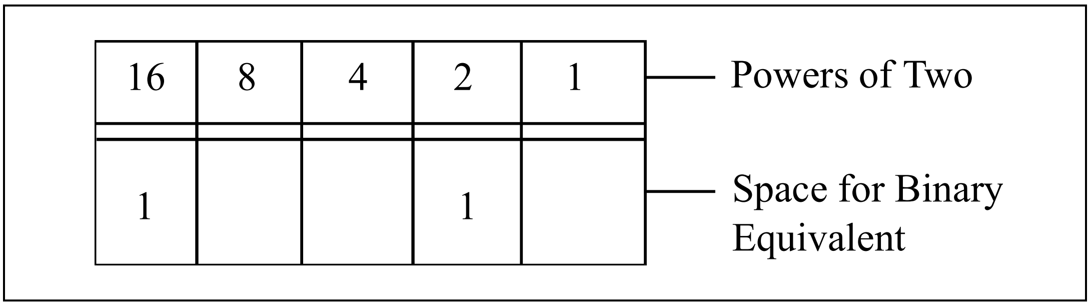
A "1" is then placed in that column and the power of two **subtracted** from the decimal number thus:

18 - 16 = 2



The highest power of two contained within the remainder is then also determined. Clearly this will be 2 in this case.

A "1" is then placed in that column and the power of two again **subtracted** from the remainder thus:



2 - 2 = 0

Once the remainder reduces to **zero**, conversion is complete. Any columns which do **not** contain a "1" should be filled by a **zero** thus:

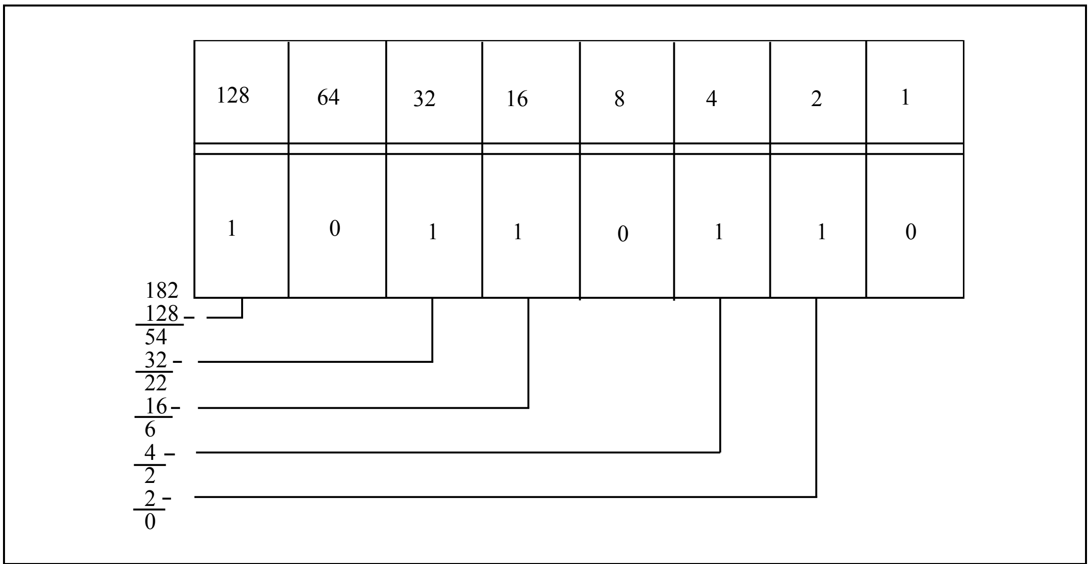
16	8	4	2	1	Powers of Two
1	0	0	1	0	Space for Binary Equivalent

So the binary equivalent of $\mathbf{18_{10}}$ is $\mathbf{10010_2}$.

Larger decimal numbers may again be converted by simply extending the number of weighted columns.

For example:

To convert 182_{10} to binary:



So the binary equivalent of $\mathbf{182_{10}}$ is $\mathbf{10110110_2}$.

Student Assessment 6

1. Binary numbers are represented in powers of:

$\boxed{a}$ 2

$\boxed{b}$ 8

$\boxed{c}$ 10

$\boxed{d}$ 16

2. To convert a binary number into decimal, the most straightforward method to use is:

$\boxed{a}$ weighted numbers

$\boxed{b}$ an abacus

$\boxed{c}$ weighted columns

$\boxed{d}$ weights

3. The decimal equivalent of 1010_2 is:

$\boxed{a}$ 12_{10}

$\boxed{b}$ 10_{10}

$\boxed{c}$ 7_{10}

$\boxed{d}$ 5_{10}

4. The decimal equivalent of 0111_2 is:

$\boxed{a}$ 12_{10}

$\boxed{b}$ 10_{10}

$\boxed{c}$ 7_{10}

$\boxed{d}$ 5_{10}

Student Assessment 6 Continued ...

5. The decimal equivalent of 110101_2 is:

- [a] 81_{10}
- [b] 69_{10}
- [c] 53_{10}
- [d] 43_{10}

6. The decimal equivalent of 11011001_2 is:

- [a] 217_{10}
- [b] 215_{10}
- [c] 155_{10}
- [d] 153_{10}

7. The binary equivalent of 6_{10} is:

- [a] 1011_2
- [b] 1010_2
- [c] 0101_2
- [d] 0110_2

8. The binary equivalent of 11_{10} is:

- [a] 1011_2
- [b] 1010_2
- [c] 0101_2
- [d] 0110_2

Continued ...

Student Assessment 6 Continued …

9. The binary equivalent of 396_{10} is:

$\boxed{a}$ 110010010_2

$\boxed{b}$ 110001100_2

$\boxed{c}$ 110001110_2

$\boxed{d}$ 110001010_2

10. The binary equivalent of 638_{10} is:

$\boxed{a}$ 1011111110_2

$\boxed{b}$ 1010111110_2

$\boxed{c}$ 1001111110_2

$\boxed{d}$ 100111110_2

Chapter 7 The Hexadecimal Numbering System

Objectives of this Chapter

Having studied this chapter you will be able to:

- Explain the reasons for the use of hexadecimal in microcomputer systems.
- Calculate the hexadecimal equivalent of a binary number.
- Convert a decimal number into hexadecimal.

Hexadecimal Numbering System

Most microprocessors will operate upon 8-bit, 16-bit or 32-bit data. Binary numbers of these lengths are difficult for most people to write down accurately, particularly when there are more than a few numbers to be written.

Decimal is far more convenient, since numbers are shorter and use 10 symbols rather than two. However, repeated conversions between decimal and binary are rather time consuming. An alternative is the hexadecimal numbering system. This system is to the base 16. Usually the 16 symbols used are the numbers 0-9 and the letters A-F:

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Conversion Between Decimal and Hexadecimal

All LJ microcomputers use hexadecimal numbers. These can be keyed in and displayed in hexadecimal form. Clearly it will be necessary to convert from decimal to hexadecimal and vice-versa. There are a number of techniques available but probably the easiest and most reliable is to use **binary** as an intermediate stage. Experience has shown this method to be the most straightforward and least prone to error.

Binary to Hexadecimal Conversion

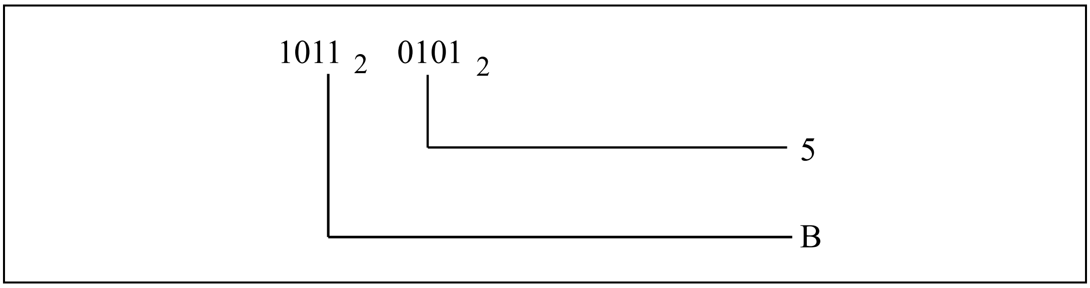
For example:

To convert 10110101_2 to hexadecimal:

The binary number must first be rewritten in groups of **4**, counting from the **right** (ie: the **least** significant bit):

1011 0101

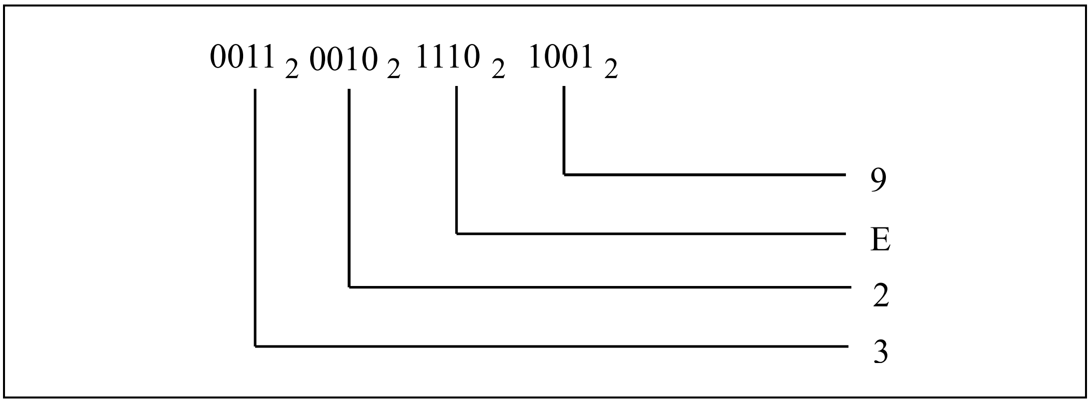
Each group of four can be identified as a hexadecimal digit by reference to the table on the previous page.



So the hexadecimal equivalent of $\mathbf{10110101_2}$ is $\mathbf{B5_H}$. Note the 'H' subscript, indicating that the number is a hexadecimal value. Larger binary numbers can be converted by extending this technique.

For example:

To convert **110010111101001_2** to hexadecimal:



So the hexadecimal equivalent of **110010111101001_2** is **$32E9_H$**

Hexadecimal to Binary Conversion

For example:

To convert $7C_H$ to binary:

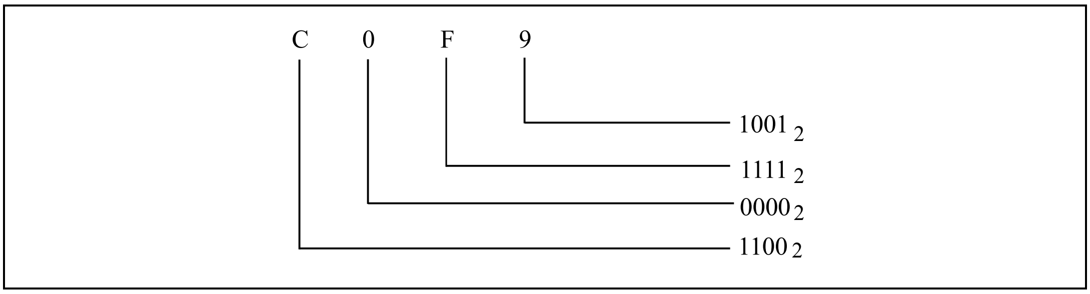
By reference to the table of hexadecimal and binary values, the binary equivalent of each hexadecimal digit can be found:

So the binary equivalent of **$7C_H$** is **$0111\ 1100_2$**.

Larger hexadecimal numbers can be treated in just the same way.

For example:

To convert $C0F9_H$ to binary:

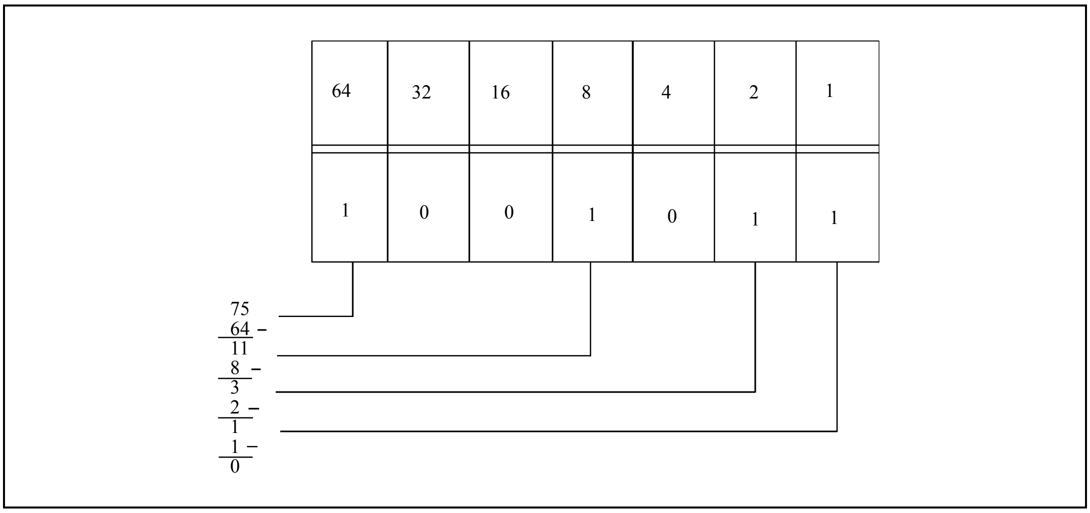


So the binary equivalent of **$C0F9_H$** is **$1100\ 0000\ 1111\ 1001_2$**. With practice the table of hexadecimal and binary equivalents will become committed to memory and these conversions can be completed "on sight".

Decimal to Hexadecimal Conversion

The decimal number is first converted to binary. This binary equivalent is then converted to hexadecimal form. For example: To convert 75_{10} to hexadecimal:

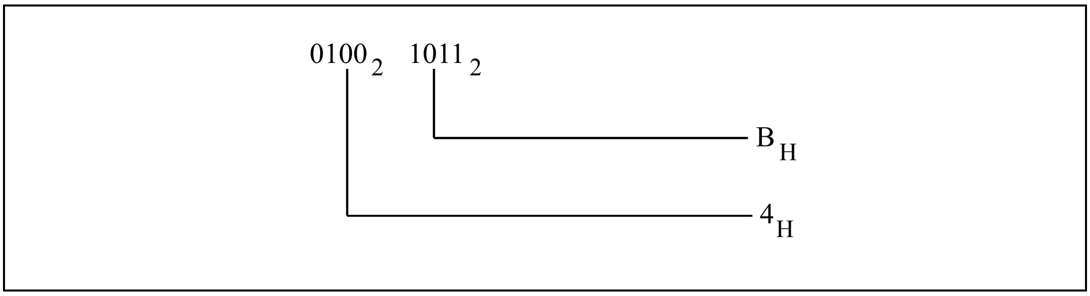
Converting 75_{10} to binary:



So the binary equivalent of 75_{10} is 1001011_2.

Converting 1001011_2 to hexadecimal:

The binary equivalent is rewritten in four-bit groups and the corresponding hexadecimal values read off the table of equivalents:



So the hexadecimal equivalent of $\mathbf{75_{10}}$ is $\mathbf{4B_H}$.

Hexadecimal to Decimal Conversion

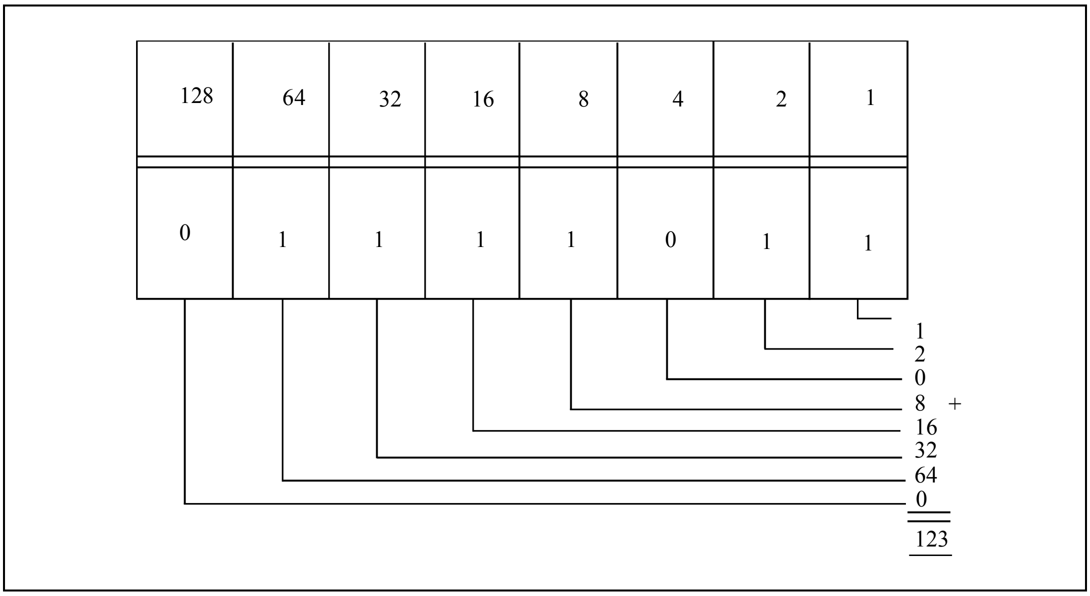
The hexadecimal number is first converted to binary. This binary equivalent is then converted to decimal form. For example: To convert $7B_H$ to decimal:

Converting $7B_H$ to binary:

By reference to the table of hexadecimal and binary values, the binary equivalent of each hexadecimal digit can be found:

So the binary equivalent of $\mathbf{7B_H}$ is $\mathbf{0111\ 1011_2}$.

This may now be converted to decimal form using a weighted column technique thus:



So the decimal equivalent of $\mathbf{7B_H}$ is $\mathbf{123_{10}}$.

Kilobytes and Megabytes

Thousands of bytes are referred to as kilobytes, often abbreviated to "K". 1Kbyte (1K) is 1024_{10} bytes (ie. $100\ 0000\ 0000_2$ or 400_H bytes). Thus:

1K $=1024_{10}$	$100\ 0000\ 0000_2$	400_H
2K $=2048_{10}$	$1000\ 0000\ 0000_2$	800_H
4K $=4096_{10}$	$1\ 0000\ 0000\ 0000_2$	1000_H
8K $=8192_{10}$	$10\ 0000\ 0000\ 0000_2$	2000_H
16K $=16386_{10}$	$100\ 0000\ 0000\ 0000_2$	4000_H
32K $=32768_{10}$	$1000\ 0000\ 0000\ 0000_2$	8000_H
64K $=65536_{10}$	$1\ 0000\ 0000\ 0000\ 0000_2$	10000_H

Millions of bytes are referred to as megabytes, often abbreviated to "M". 1Mbyte (1M) is 1048576_{10} bytes (ie.100000_H bytes). Thus:

1M $= 1048576_{10}$	100000_H
2M $= 2097152_{10}$	200000_H
10M $= 10485760_{10}$	$A00000_H$
20M $= 20971520_{10}$	1400000_H
40M $= 41943040_{10}$	2800000_H

Student Assessment 7

1. Hexadecimal numbers are represented in powers of:

a 2

b 8

c 10

d 16

2. Hexadecimal is used because:

a Conversions between hexadecimal and binary are easier and less time consuming than between decimal and binary.

b Microcomputers only understand hexadecimal.

c It is easier for humans to understand hexadecimal than decimal.

d Hexadecimal uses letters as well as numbers

3. The hexadecimal equivalent of 1011_2 is:

a E_H

b B_H

c 9_H

d 5_H

4. The hexadecimal equivalent of 10100110_2 is:

a $E6_H$

b $B6_H$

c $A6_H$

d 96_H

Continued ...

Student Assessment 7 Continued …

5. The hexadecimal equivalent of 11011011_2 is:

a ED_H

b DB_H

c CB_H

d BD_H

6. The hexadecimal equivalent of 111010101_2 is:

a $EA8_H$

b $75A_H$

c $2D5_H$

d $1D5_H$

7. The binary equivalent of $F3_H$ is:

a $0011\ 1111_2$

b $1100\ 1111_2$

c $1110\ 0011_2$

d $1111\ 0011_2$

8. The binary equivalent of $D9A_H$ is:

a $1101\ 1001\ 1010_2$

b $1101\ 1000\ 1011_2$

c $1110\ 1010\ 1011_2$

d $1110\ 1001\ 1011_2$

Student Assessment 7 Continued ...

9. The hexadecimal equivalent of the decimal number 67_{10} is:

a $C6_H$

b 86_H

c 43_H

d $3D_H$

10. The hexadecimal equivalent of the decimal number 127_{10} is:

a $F7_H$

b $7F_H$

c $E6_H$

d $6E_H$

11. The decimal equivalent of the hexadecimal number 88_H is:

a 58_{10}

b 98_{10}

c 136_{10}

d 166_{10}

12. The decimal equivalent of the hexadecimal number 200_H is:

a 1024_{10}

b 512_{10}

c 256_{10}

d 128_{10}

Chapter 8 Microelectronic Circuits

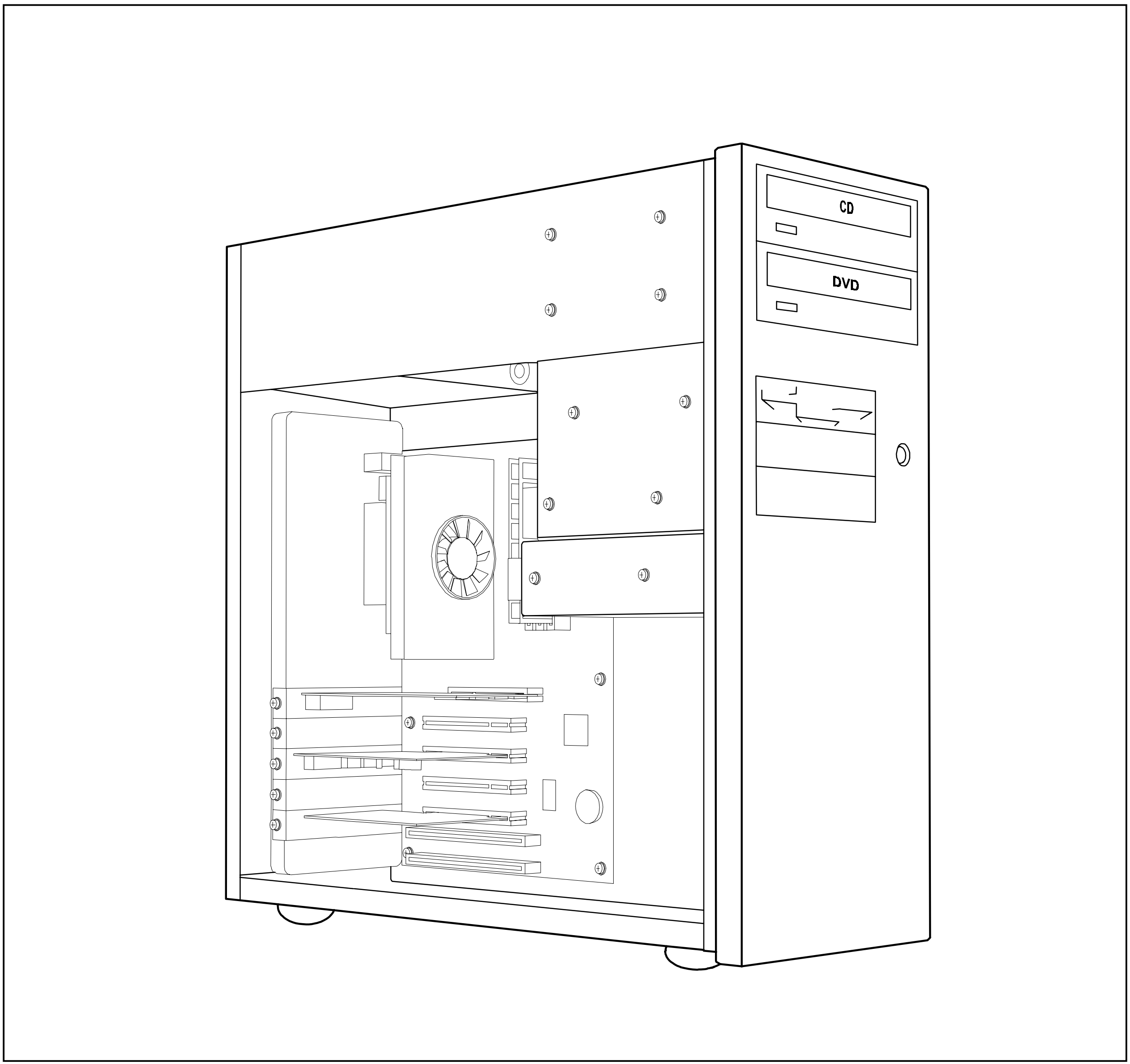
Objectives of this Chapter

Having studied this chapter you will be able to:

- Explain the reasons for using Integrated Circuits.
- Describe the functions of the microcomputer system units.
- Describe the functions of the microcomputer system buses.

Introduction

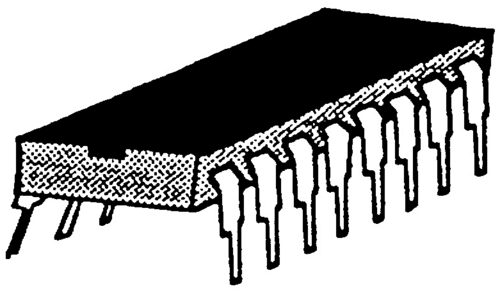
The physical parts of a computer system (eg. Circuit boards, wiring, peripherals, etc.) are often referred to as hardware. The program will cause data to be routed between the various **hardware** sections of the microcomputer.



Integrated Circuits

An **Integrated Circuit (IC)** is a complete functional electronic circuit which is encapsulated so that it can be used as a single component. Most of the mass-produced IC's are **monolithic silicon** IC's. This means that all of the individual components necessary to complete the overall function are implemented on a **single** piece of silicon, often referred to as a "**chip**". Such IC's are very small, being typically just a few millimeters square.

IC's can be found on all LJ microcomputer Boards. These are mainly of the "Dual in Line" (DIL) type. Most DIL IC's have the basic construction shown below.



IC's were developed for a number of reasons.

These include:

- Much smaller size and lower weight
- Lower unit costs
- Ease of circuit construction
- Greater reliability
- Ease of fault location and replacement

IC's can have analog or digital circuit functions. Examples of both types may be found in LJ products, but microcomputers will have predominantly digital IC's.

IC's can be classified according to the degree of integration (ie: miniaturization) thus:

Small Scale Integration (SSI)	up to 10 transistors per Integrated Circuit
Medium Scale Integration (MSI)	up to 100 transistors per Integrated Circuit
Large Scale Integration (LSI)	up to 1000 transistors per Integrated Circuit
Very Large Scale Integration (VLSI)	over 1000 transistors per Integrated Circuit

The Microcomputer

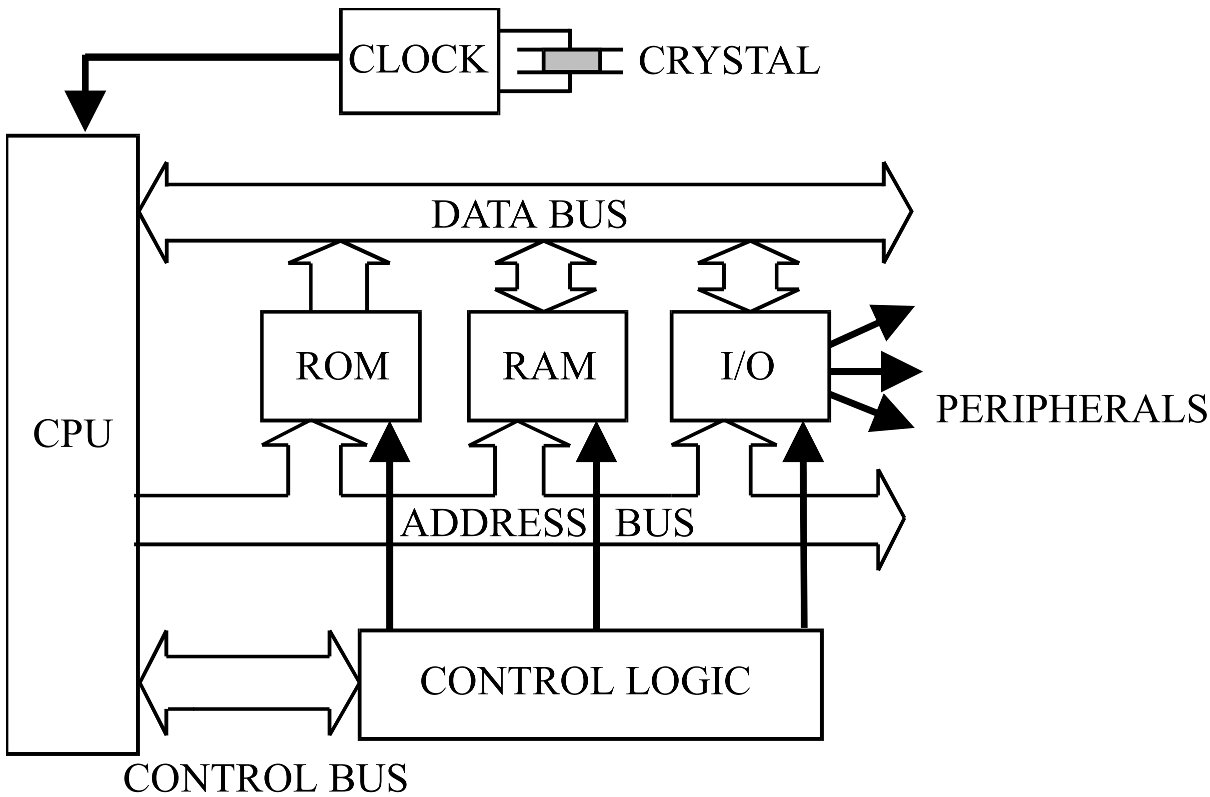
Many manufacturers produce one or more **microprocessors**.

A microprocessor is a VLSI device which can be programmed to process data in a wide variety of ways.

The microprocessor functions as the **Central Processing Unit (CPU)** for a microcomputer. However a microprocessor is not a microcomputer in itself. It requires the support of two other units. A **Memory Unit** is required to store programs and data. An **Input/Output Unit** is required to allow the microcomputer to communicate with the outside world. It is also necessary to provide interconnecting data transmission pathways, called "**buses**", and a timing **clock**. These basic elements are common to all computers.

Microcomputers which perform a **single** function are referred to as **dedicated systems**. Those which may perform a **variety** of tasks are called general purpose. All LJ microcomputer systems are **general purpose** but some can be used as **dedicated** microcontrollers by replacing the Monitor EPROM with a suitable user-programmed device.

A generalized block diagram of a microcomputer is shown on the next page.



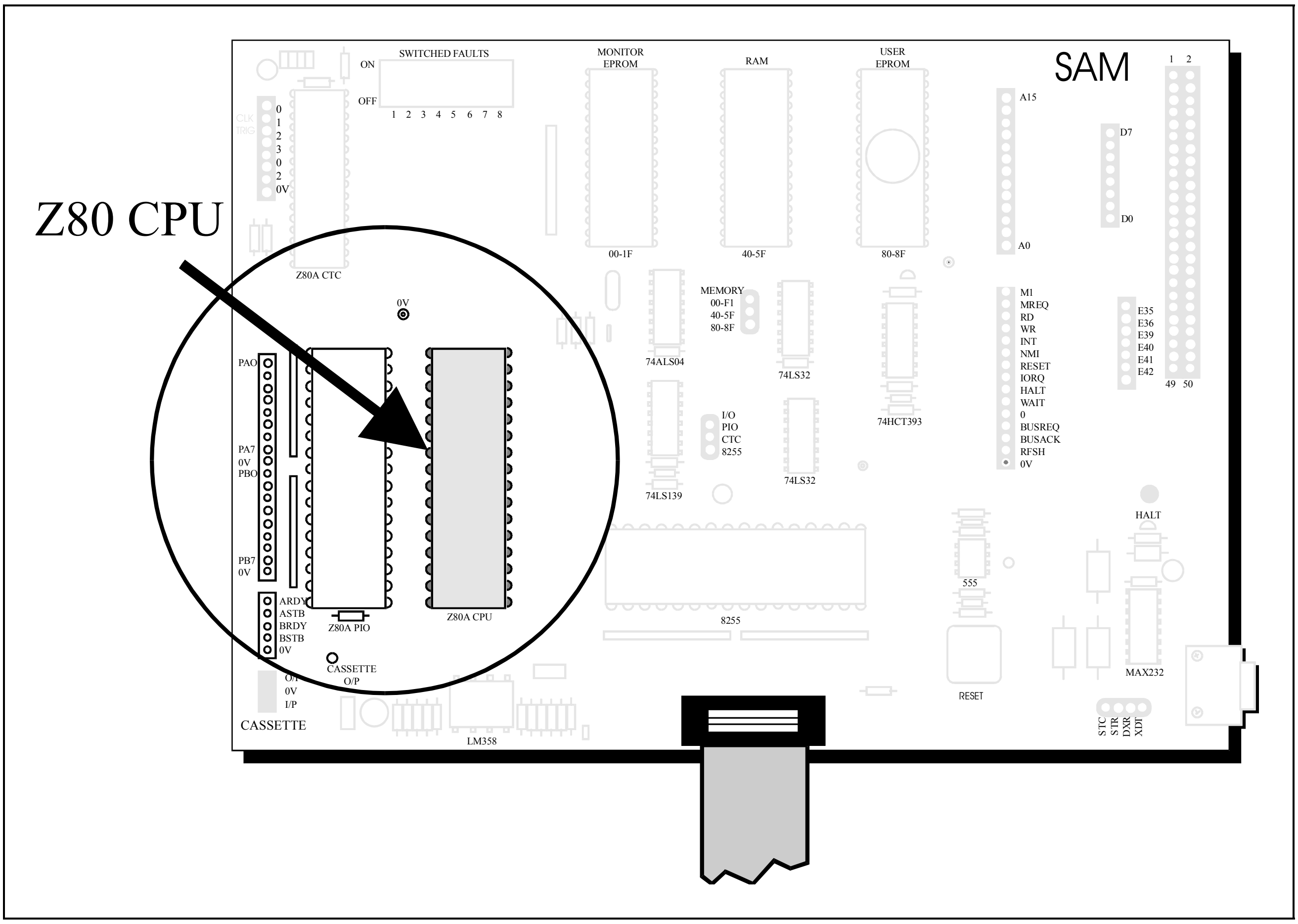
Block diagram of a Microcomputer

Central Processing Unit (CPU)

This part of the microcomputer is responsible for the following functions:

- **Fetching instructions in the correct order from memory.**
- **Interpreting instructions received.**
- **Carrying out instructions received.**

The location of the CPU device on the LJ SAM Microcomputer board is shown below:



Memory Unit

This consists of a number of memory **locations**, each containing a binary value which has the **same** number of bits. The **contents** of each location may be interpreted as an instruction or data. Each location has a unique identifying label, called the **address**. No two locations may have the same address.

In LJ microcomputers these functions are carried out by the devices labeled "MONITOR EPROM", "RAM" and "USER EPROM" (if fitted).

Input/Output Unit

All computers must have some form of access to the "outside world". This part of the microcomputer connects (or interfaces) the microcomputer to external devices and systems. Very often a special interfacing Input/Output (I/O) IC is used. These I/O devices typically have 16 I/O data lines, arranged in two groups of 8. This gives two 8-bit data ports. The ports can be accessed by the CPU to allow data to be transmitted or received from the external environment.

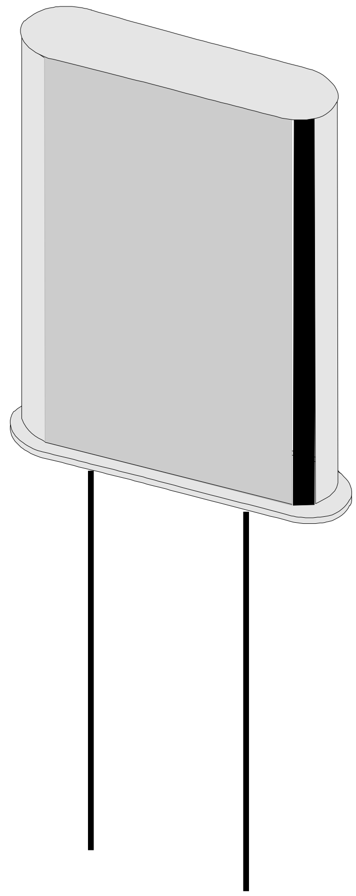
Examples of the names given to such devices are PIO (Parallel Input/Output), PIA (Peripheral Interface Adaptor) and VIA (Versatile Interface Adapter).

Clock

Most digital circuits are **synchronous**. This means that they require a repetitive train of well defined pulses to control the timing of all operations. The microprocessor will then require such an input to synchronize internal and external events.

Microprocessor clocks are based on **crystal controlled** oscillators since these give a stable repetitive waveform.

Microprocessor clocks usually operate at frequencies between 1MHz and 10MHz.



A Typical Crystal

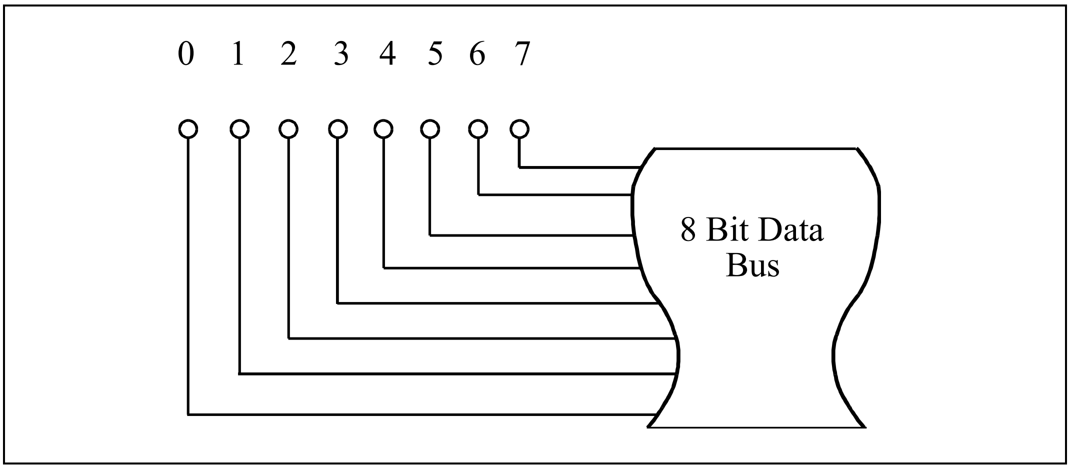
System Buses

A **bus** is a group of parallel conductors, all having broadly the same function. **Unidirectional** buses carry data in **one** direction only, whereas **bidirectional** buses carry data in **both** directions.

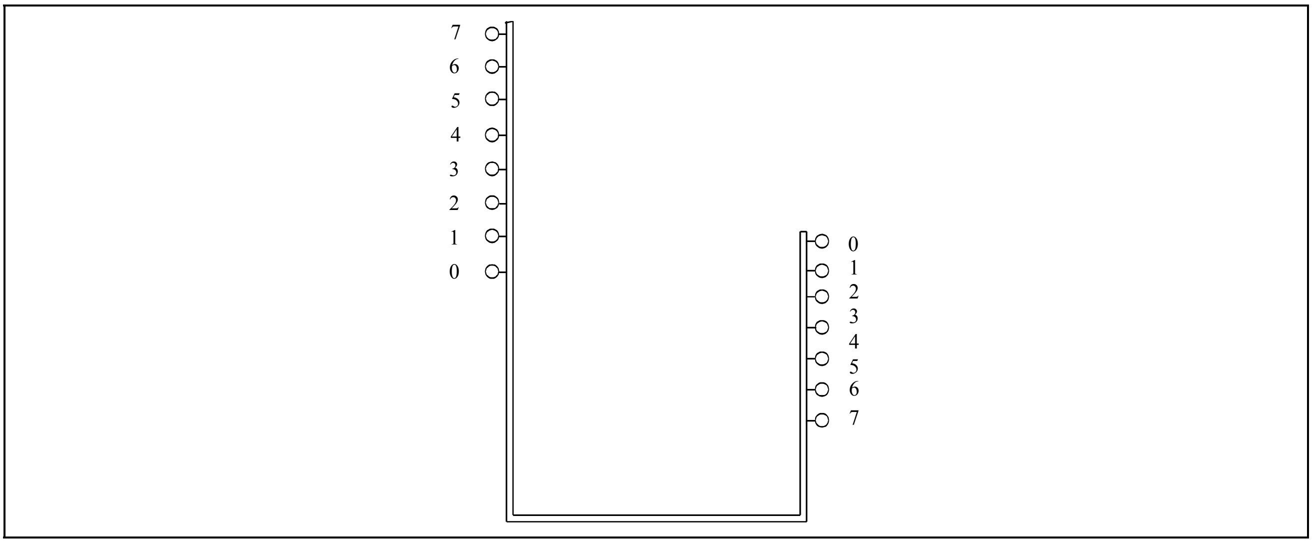
All LJ microcomputer boards have a row of connector pins which allow access to the system buses.

For convenience it is not usual to draw each individual bus line.

Buses are often drawn as a block thus:



Sometimes the bus is drawn as a single, thick line thus:



Address Bus

The address bus is unidirectional. The lines within this bus convey information from the CPU to Memory and I/O Units concerning the location of data. The Z80 and 6502 microprocessors have a 16-bit address bus, allowing 2^{16} (= 65536_{10}) locations to be uniquely identified.

Data Bus

The data bus is bidirectional. It conveys data in either direction between the CPU and the Memory or I/O Units.

However, data may not flow directly between the Memory and I/O Units. Such data transfers must be via the CPU. The number of bits in the data bus depends upon the number of bits which the CPU can work on at one time. (e.g. 8 bits for Z80 and 6502).

Control Bus

The control bus is really just a collection of CPU inputs and outputs which are grouped together for convenience. Some control bus signals flow away from the CPU and some towards it. However, signals only ever travel in a single direction on any given line and so the control bus can be considered unidirectional.

The control signals are required to direct memory read and write operations and to ensure correct responses to specified events.

Student Assessment 8

1. Any miniaturized, complete functional electronic circuit can be called:

- a a microcomputer
- b a CPU
- c a microprocessor
- d an integrated circuit

2. Which of the following is <u>not</u> a reason for the development of the IC?

- a Ease of circuit construction.
- b Greater reliability.
- c An IC can be soldered into a circuit board.
- d Ease of fault location and replacement.

3. The number of transistors contained within an LSI IC is:

- a Up to 10
- b Up to 100
- c Up to 1000
- d Over 1000

4. Which 3 main units are required to form a microcomputer?

- a Microprocessor, Memory Unit, I/O Unit.
- b Microprocessor, Memory Unit, Clock.
- c Microprocessor, Memory Unit, Buses.
- d Microprocessor, I/O Unit, Buses.

Student Assessment 8 Continued ...

5. **Microcomputers which perform a single function are called:**
   - a general purpose systems
   - b dedicated systems
   - c multi-purpose systems
   - d universal systems

6. **Which of the following is an example of a memory unit?**
   - a CPU
   - b PIO
   - c RAM
   - d VDU

7. **Which of the following is an example of an I/O device?**
   - a CPU
   - b PIO
   - c RAM
   - d VDU

8. **Which of the following is <u>not</u> a function of the CPU?**
   - a Interpreting instructions
   - b Carrying out instructions
   - c Storing programs
   - d Fetching instructions from memory

Continued ...

Student Assessment 8 Continued ...

9. Which of the following is a bidirectional bus:

- a Address bus
- b Memory bus
- c Data bus
- d Control bus

10. How many locations can an 8-bit Address bus identify?

- a 131072_{10}
- b 65536_{10}
- c 512_{10}
- d 256_{10}

Chapter 9 The Central Processing Unit

Objectives of this Chapter

Having studied this chapter you will be able to:

- Explain the functions of the ALU and Associated Registers within the CPU.
- Explain the functions of the Control Section of the CPU.

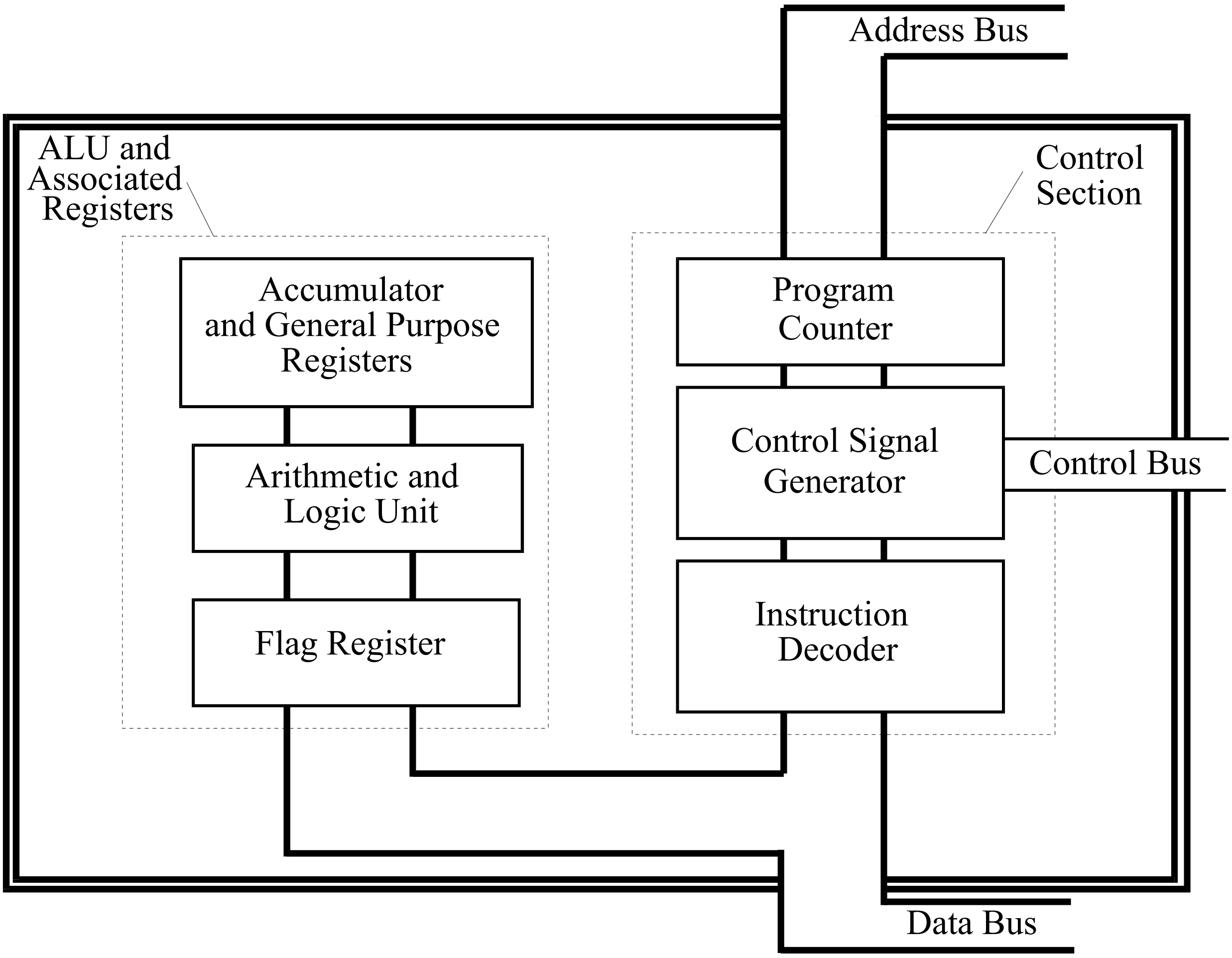
Introduction

Registers are rather like individual memory locations **within** the CPU itself. However, the number of registers is very much smaller than the number of external memory locations. The advantage to the CPU of using internal registers is that they may be accessed via **internal** buses, without the need for an external memory read or write operation. Some registers will be under program control but others will not.

A very generalized diagram of the internal architecture of a microprocessor is shown on the next page. This can be applied to most microprocessors.

Notice that there are two basic sections:

- **Control Section**
- **Arithmetic and Logic Unit and Associated Registers**


Address Bus
ALU and
Associated
Registers
Control
Section
Accumulator
and General Purpose
Registers
Program
Counter
Arithmetic and
Logic Unit
Control Signal
Generator
Control Bus
Flag Register
Instruction
Decoder
Data Bus

Arithmetic and Logic Unit and Associated Registers

This section of the CPU is concerned with the execution of instructions.

Accumulator and General Purpose Registers
The accumulator and general purpose registers act as temporary storage registers during calculations. They are the CPU registers most frequently accessed directly by the user.

Arithmetic and Logic Unit (ALU)
This unit contains circuits which will add, subtract and perform logical operations. The result of an ALU operation is usually returned to the accumulator.

Flag Register
This consists of a collection of flags which indicate the status of the most recent ALU operation. For example a carry flag will be set (ie. becomes logic "1") if the result of the last ALU operation exceeded the length of the accumulator (8-bits for the Z80 and 6502).

Control Section

This section of the CPU is concerned with the transfer of instructions and data between the CPU and memory or input/output unit.

Program Counter
This register holds the address of the next instruction or data word in the program sequence. This address is output on the address bus so that the appropriate location may be accessed by the CPU.

Control Signal Generator
This produces the necessary internal and external control signals for each instruction to be completed. This unit will also respond to certain external events.

Instruction Decoder
This unit will interpret instructions received from memory via the data bus and ensure that the control signal generator produces the correct signals in the correct sequence for the instruction to be completed.

Student Assessment 9

1. Which of the following is <u>not</u> contained within the Control Section of the CPU?

a Program Counter

b Control Signal Generator

c Accumulator

d Instruction Decoder

2. The advantage of internal registers is that:

a There are more internal registers than external memory locations.

b They can be accessed via internal buses.

c They can be accessed via external buses.

d They are always under program control.

3. The value held by the Program Counter is output on the:

a Data bus

b Control bus

c Address bus

d Memory bus

4. The function of the Accumulator is to:

a act as a temporary store.

b perform addition, subtraction and logical operations.

c hold the address of the next instruction in sequence.

d interpret instructions received from memory.

Continued ...

Student Assessment 9 Continued ...

5. The unit of the CPU which is connected directly to the Control bus is the:

- a Instruction decoder
- b Accumulator and general purpose registers
- c Program Counter
- d Control Signal Generator

6. The result of an ALU operation is usually returned to the:

- a Control Signal Generator
- b Accumulator
- c Program Counter
- d Memory Unit

7. The unit of the CPU which indicates the status of the most recent ALU operation is the:

- a Control Signal Generator
- b Program Counter
- c Flag Register
- d Arithmetic and Logic Unit

Chapter 10 The Memory Unit

Objectives of this Chapter

Having studied this chapter you will be able to:

- State the fundamental features of :
  RAM
  ROM

- Describe the function of a memory map.

Introduction

The microprocessor performs the tasks required of it by carrying out its **program**. This is a series of **instructions**. The program will be stored in **memory**. The memory unit will consist of one or more **memory devices**. These are IC's which are designed to store data.

There are two basic types of memory device: RAM and ROM.

Read/Write Memory (RAM)

This type of memory can be read or written by the user. The contents of RAM are **completely lost** when the power is switched off. So this type of memory is said to be **volatile**. RAM is used to store data which may **change**.

A program stored in RAM is often referred to as **software**.

All LJ microcomputers contain RAM for temporary data storage.

Read Only Memory (ROM)

This type of memory has the program written in by the system manufacturer and cannot be easily altered by the user. This type of memory is said to be **non-volatile** since the contents are not lost when the power is switched off. ROM is used to store data which does **not** change.

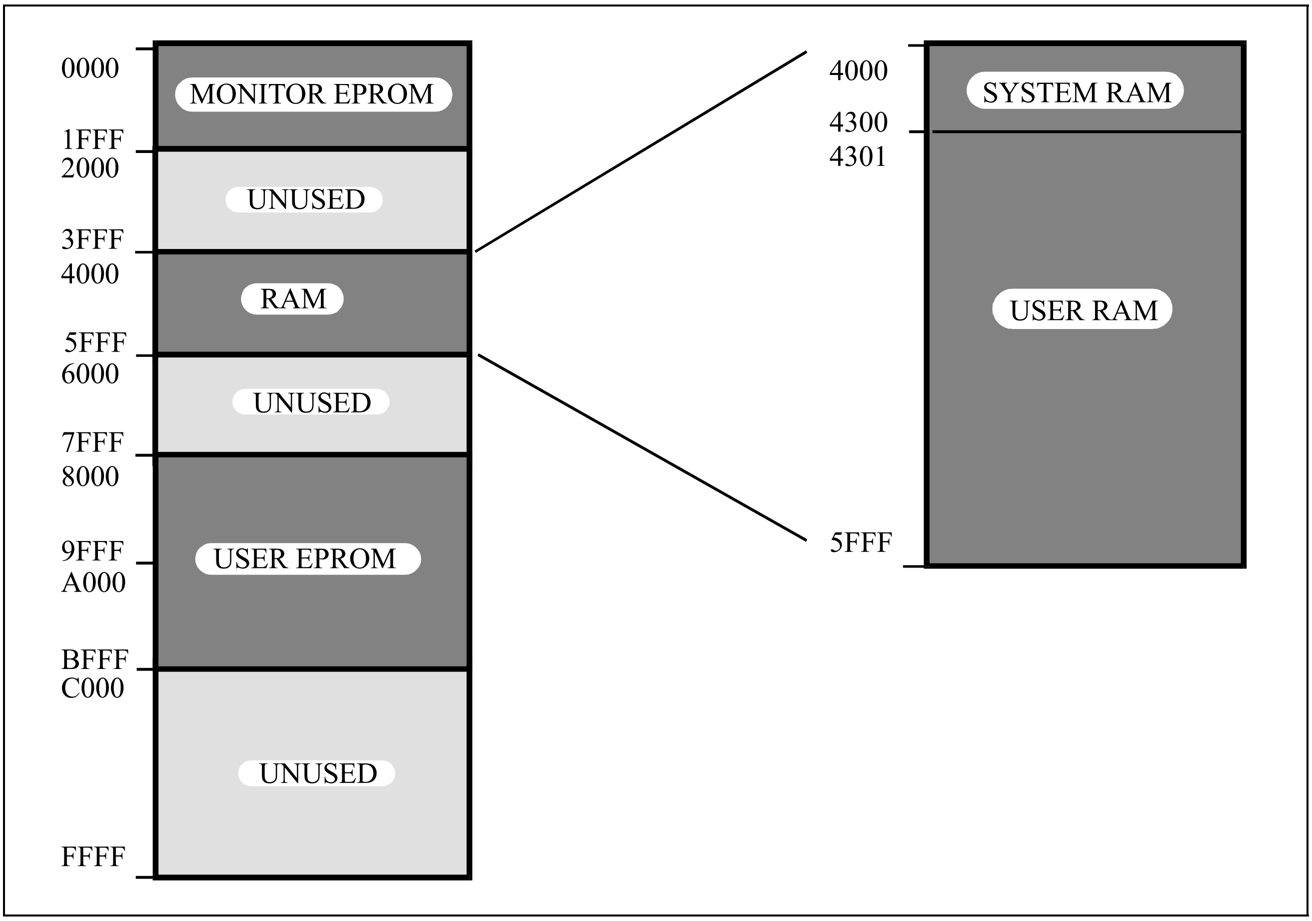
A program stored in ROM is often referred to as **firmware**. Some types can be erased by exposure to ultra-violet (UV) light. All LJ microcomputers use a Monitor EPROM (Erasable Programmable Read Only Memory). These devices must have opaque labels to cover the window by which they are erased, since the UV component of sunlight will erase the contents of EPROM with time. An EPROM programming unit is available from LJ Technical Systems.

Address Decoding

The memory devices within a system must be given an address range to label each location.

A diagram showing the decoding of memory within your LJ microcomputer is shown in the user manual. This is called a **memory map**. Address decoding IC's are used to place memory devices within the memory map.

An example of a microcomputer memory map is shown below:



The memory map shows the user where programs and data may and may not be stored.

For example, memory addresses 6000_H to $7FFF_H$ in the example memory map shown above are unused. This means that no memory device occupies this range of addresses, so this area cannot be used for the storage of user programs and data.

Student Assessment 10

1. Memory which loses its contents when the power is removed, is said to be:

a volatile

b non-volatile

c unstable

d changeable

2. A program stored in ROM is often referred to as:

a software

b hardware

c secureware

d firmware

3. The term EPROM means:

a Electrically Programmable Read Only Memory

b Erasable Programmable Read Only Memory

c Electrically Permanent Read Only Memory

d Erasable Programs Read Only Memory

4. Continuous exposure of an EPROM's window to sunlight, will, over time:

a have no effect

b melt its circuits

c erase its contents

d make it un-programmable

Student Assessment 10 Continued ...

5. RAM is used for:

a permanent data storage

b temporary data storage

c placing memory devices within the memory maps

d interpreting instructions from memory

6. A memory map is required to show:

a whether a program is correct

b how fast the microcomputer can operate

c what data each byte of memory holds

d where in memory programs may or may not be stored

7. In the example memory map shown earlier, the type of memory which is available between addresses 4000_H and 4300_H is:

a Monitor EPROM

b System RAM

c User RAM

d User EPROM

Chapter 11 The Input/Output Unit

Objectives of this Chapter

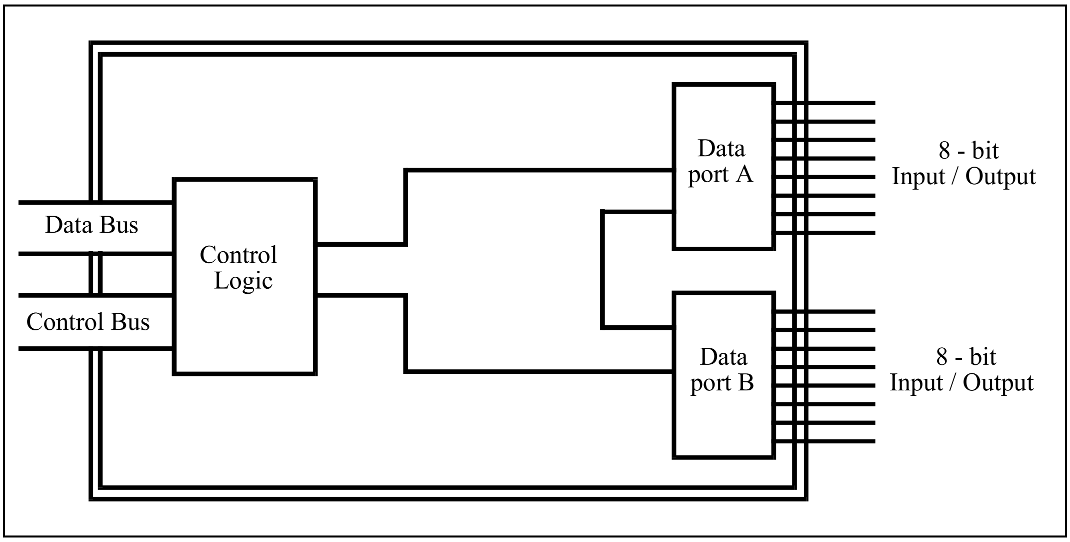
Having studied this chapter you will be able to:

- State why I/O devices are required in microcomputer systems.
- Describe interfacing and the incompatibilities between computers and their peripherals.
- Explain the difference between serial and parallel communication.

Introduction

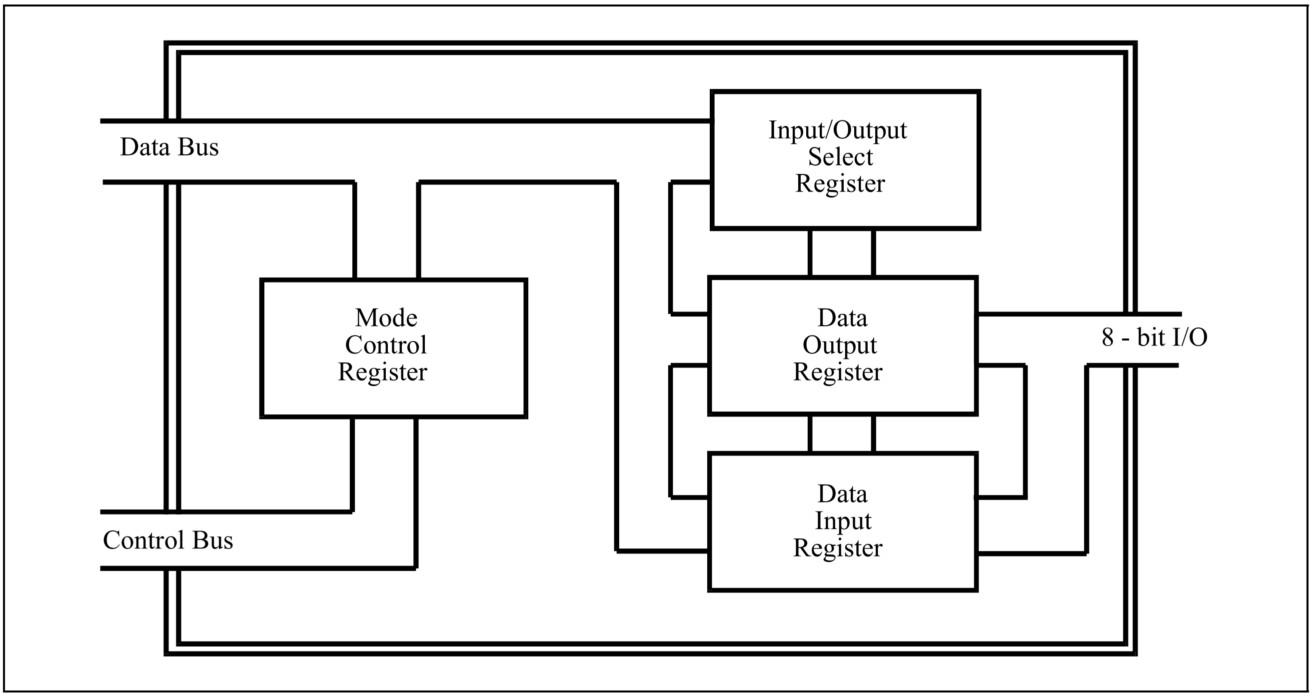
It is vital for the microcomputer to communicate with its external environment. A personal computer must be able to output characters to a screen and printer. It must be able to read inputs from the keyboard and be able to save and recall disk-based data. A microcomputer-based controller must read its input sensors and cause the outputs to change, according to the control program.

The microprocessor will usually communicate with external devices and systems by means of a "family" Input/Output (I/O) device. Typical internal architecture for such a device is shown below:



I/O devices are usually **programmable**. This means that setting up the way the device operates is achieved by running a program which writes to **registers** inside the I/O device.

The internal registers for one particular type of I/O device are shown below:



In LJ Technical Systems microcomputers, such a device can be used for user I/O applications such as controlling a stepper motor or monitoring a bank of switches. Another I/O device may also be used by the monitor program, to interface the microcomputer to the keypad and display.

Typically, a programmable I/O device will provide two 8-bit parallel I/O **ports**. Any of the bits within these ports may be programmed to act as an **input** or an **output** by the user. This may also be changed during the execution of a program.

Interfacing

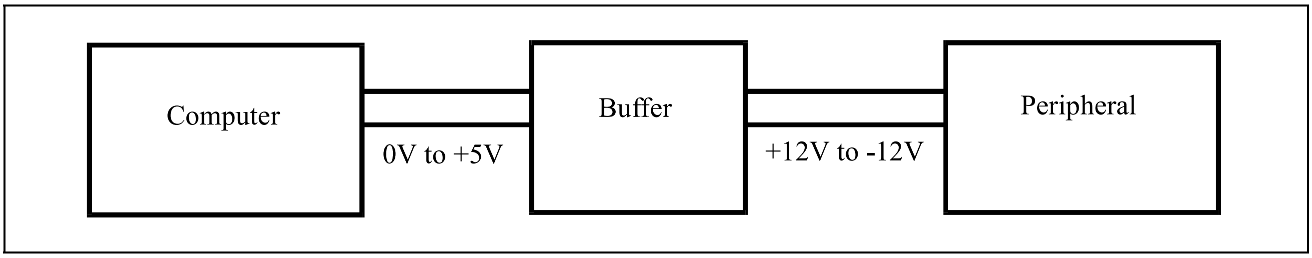
Often it is not possible to directly connect a peripheral to an I/O device, due to some incompatibility between them. An interface is rather like an "adaptor" which is used to "match" different systems together.

Incompatibilities between computers and their peripherals can be classified in four main groups:

1. Electrical Incompatibility

Sometimes a peripheral will require voltages or currents which the microcomputer's I/O device cannot provide. In such circumstances **electrical buffering** is required to change voltage or current levels to those required.

For example:

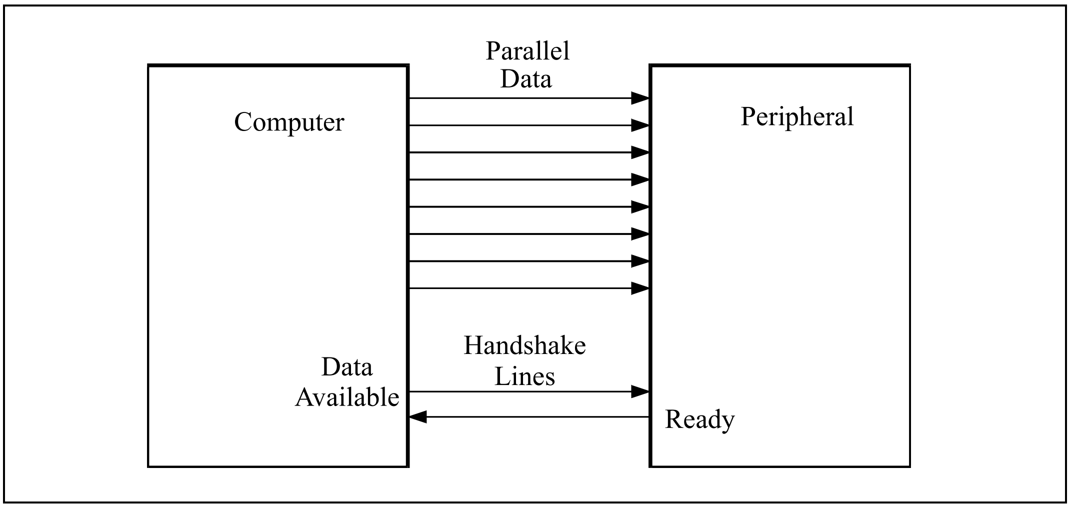


2. Timing Incompatibility

The microcomputer will usually operate very much faster than any peripheral to which it may be connected. There are two methods for overcoming this difficulty:

Software Delay
This technique uses a simple delay routine to prevent the computer from transmitting data at too high a rate. The method relies upon the assumption that if enough time has been allowed for the completion of a task then the task has been completed. This can be very inefficient since the necessary time delay must assume "worst case" conditions.

Handshaking
Handshaking provides a more precise synchronization of data transfers. It requires the use of additional data lines called handshake lines.



The peripheral will signal the computer via the READY line when it is ready to receive data. Only then can the computer write valid data to the Input/Output Port. The computer will also issue a "DATA AVAILABLE" signal to indicate to the peripheral that valid data is now on the I/O Port.

3. Code Incompatibility

The computer manipulates data in **pure binary** form. However, some peripherals use different binary codes.

Clearly then, **code conversion** will be necessary in such cases. For example the conversion of pure binary to gray code:

BINARY	GRAY
0000	0000
0001	0001
0010	0011
0011	0010
0100	0110
0101	0111
0110	0101
0111	0100

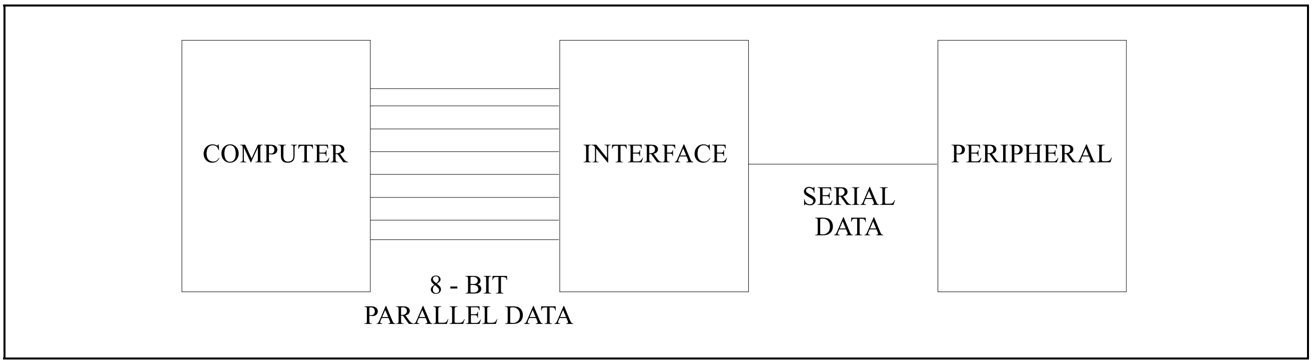
Code conversion is often carried out by **software** means but sometimes **hardware** techniques are used.

Occasionally a mixture of both is appropriate.

4. Data Lines Incompatibility

The computer will always operate upon a **fixed** number of bits. However, a peripheral may require a **different** number of bits so it is sometimes necessary to interface between one system and another.

The most common data lines incompatibility is where a peripheral works on **serial data** (for example RS232). Serial data only requires a **single** data line, whereas the computer will operate upon **parallel data**.



Since this is a common problem, manufacturers produce integrated circuits which will perform this conversion.

Serial communication is inherently much slower than parallel communication since it must transmit data one bit at a time.

Parallel communication will be much faster since a whole data word may be transmitted at a time. Serial communication does have the advantage that far fewer data lines are required. For these reasons parallel communication is usually reserved for fairly short distance communication (a few meters).

Student Assessment 11

1. I/O devices are required so that microcomputers can:

- a read data from memory
- b communicate with external devices
- c perform ALU arithmetic
- d transfer internal data on buses

2. Typically a programmable I/O device provides two parallel I/O ports, each of which contains:

- a 4 bits
- b 8 bits
- c 16 bits
- d 32 bits

3. Interfacing is required to:

- a match different systems together
- b prevent different systems from being matched
- c allow internal data to be transferred on the data bus
- d run all microcomputer programs.

4. Which of the following is not a possible cause of interfacing problems?

- a Data lines incompatibility
- b Code incompatibility
- c Register incompatibility
- d Electrical incompatibility

Continued ...

Student Assessment 11 Continued ...

5. One method of overcoming a timing incompatibility is by:

a code conversion

b handshaking

c electrical buffering

d using parallel communication techniques

6. When interfacing a computer to a peripheral, code conversion may be required because:

a binary is difficult for humans to understand

b not all peripherals use pure binary codes

c microprocessors can only read hexadecimal

d peripherals only use pure binary codes

7. Which of the following statements is <u>not</u> true?

a Parallel communication is faster than Serial communication

b Serial data is transmitted one bit at a time

c Parallel communication requires more data lines

d Serial communication is faster than parallel communication