GL BAJAJ
Institute of Management
FIND YOUR SPARK
Affiliated to CCS University, Meerut.

Computer Architecture & Assembly Language

BCA-303

MR.SURESH KUMAR

Assistant Professor

AT

GLBIM Greater Noida

GL BAJAJ Institute of Management
FIND YOUR SPARK
Affiliated to CCS University, Meerut.

Computer Architecture & Assembly Language

BCA-303

Course Contents / Syllabus

Unit – I

Basic computer organization and design, Instructions and instruction codes, Timing and control/ instruction cycle, Register/ Types of register/ general purpose & special purpose registers/ index registers, Register transfer and micro operations/ register transfer instructions, Memory and memory function, Bus/ Data transfer instructions, Arithmetic logic micro-operations/ shift micro-operations, Input/ Output and interrupts, Memory reference instructions, Memory interfacing memory/ Cache memory.

Computer Architecture & Assembly Language

BCA-303

Course Contents / Syllabus

Unit – II
Central Processing Unit

General Register Organization/ stacks organizations instruction formats, addressing modes, Data transfer and manipulation. Program control reduced computer, pipeline/ RISC/ CISC pipeline vector processing/ array processing. Arithmetic Algorithms: Integer multiplication using shift and add, Booth’s algorithm, Integer division, Floating-point representations.

GL BAJAJ
Institute of Management
FIND YOUR SPARK
Affiliated to CCS University, Meerut.

Computer Architecture & Assembly Language

BCA-303

Course Contents / Syllabus

Unit – III
Computer Arithmetic

Addition, subtraction and multiplication algorithms, divisor algorithms. Floating point, arithmetic operations, decimal arithmetic operations, decimal arithmetic operations.

GL BAJAJ Institute of Management
FIND YOUR SPARK
Affiliated to CCS University, Meerut.

Computer Architecture & Assembly Language

BCA-303

Course Contents / Syllabus

Unit – IV
Input - Output Organization

Peripheral devices, Input/output interface, ALU Asynchronous Data transfer, mode of transfer, priority interrupts, Direct memory Address (DMA), Input/ Output processor (IOP), serial communication.

GL BAJAJ
Institute of Management
FIND YOUR SPARK
Affiliated to CCS University, Meerut.

Computer Architecture & Assembly Language
BCA-303

Course Contents / Syllabus

Unit –V

Evaluation of Microprocessor

Overview of Intel 8085 to Intel Pentium processors Basic microprocessors, architecture and interface, internal architecture, external architecture memory and input/ output interface.

GL BAJAJ Institute of Management
FIND YOUR SPARK
Affiliated to CCS University, Meerut.

Computer Architecture & Assembly Language
BCA-303

Course Contents / Syllabus

Unit –VI

Assembly language, Assembler, Assembly level instructions, macro, use of macros in I/C instructions, program loops, programming arithmetic and logic subroutines, Input-Output programming.

GL BAJAJ
Institute of Management
FIND YOUR SPARK
Affiliated to CCS University, Meerut.

Computer Architecture & Assembly Language

BCA-303

Course Contents / Syllabus

Unit – I

Basic computer organization and design, Instructions and instruction codes, Timing and control/ instruction cycle, Register/ Types of register/ general purpose & special purpose registers/ index registers, Register transfer and micro operations/ register transfer instructions, Memory and memory function, Bus/ Data transfer instructions, Arithmetic logic micro-operations/ shift micro-operations, Input/ Output and interrupts, Memory reference instructions, Memory interfacing memory/ Cache memory.

Computer Architecture & Assembly Language

BCA-303

Computer architecture

can be defined as a set of rules and methods that **describe the functionality**, **management** and **implementation of computers**. To be precise, it is nothing but rules by which a system performs and operates.

Role of computer Architecture

The main role of Computer Architecture is to **balance the performance**, **efficiency**, **cost** and **reliability** of a computer system.

For Example − **Instruction set architecture (ISA) acts as a bridge between computer's software and hardware.** It works as a programmer's view of a machine.

Computer Architecture & Assembly Language

BCA-303

Computers can only understand binary language (i.e., 0, 1) and **users understand high level language (i.e., if else, while, conditions, etc).** So to communicate between user and computer, **Instruction set Architecture plays a major role here, translating high level language to binary language.**

Computer Architecture & Assembly Language

BCA-303

Structure

Let us see the example structure of Computer Architecture as given below.

Generally, **computer architecture consists of the following**

- Processor
- Memory
- Peripherals

All the above parts are connected with the help of system bus, which consists of address bus, data bus and control bus.

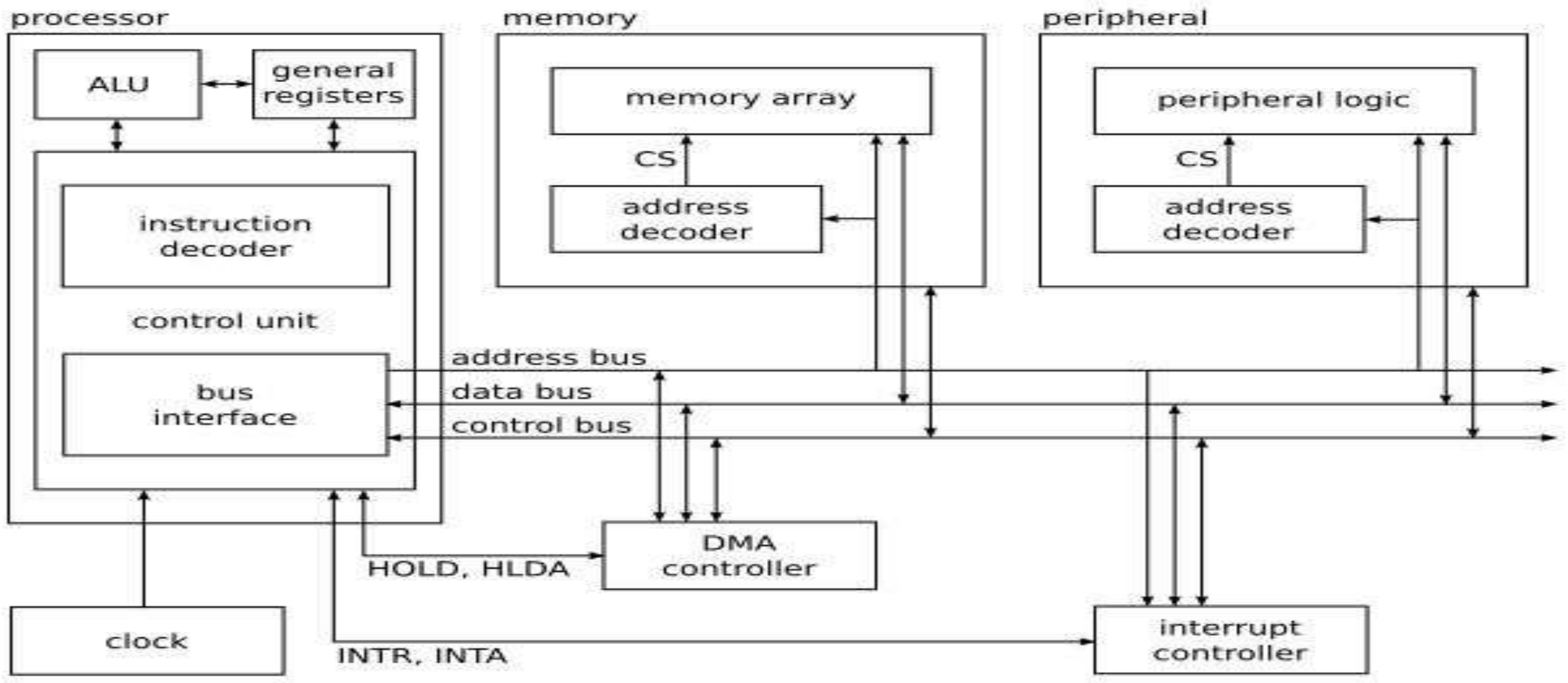
The diagram given below depicts the computer architecture −

Continue…

GL BAJAJ Institute of Management
FIND YOUR SPARK
Affiliated to CCS University, Meerut.

Computer Architecture & Assembly Language

BCA-303



Computer Architecture & Assembly Language

BCA-303

Instruction

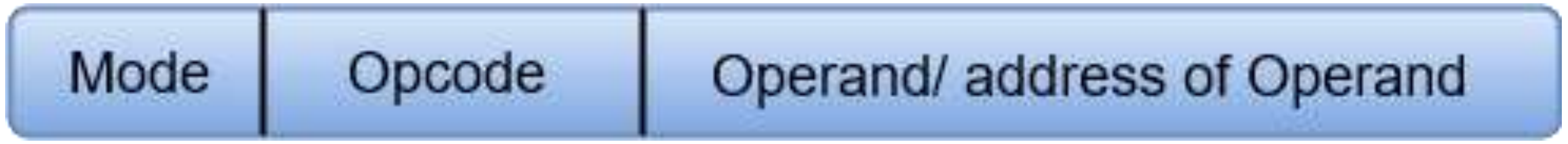
In a computer architecture , an instruction is a single operation of a processor define by the processor instruction set. The size of a instruction is 4 bits.
A process is controlled by a program.

- A program is a set of instructions that specify the operations, data, and the control sequence.
- An instruction is stored in binary code that specifies a sequence of microoperations.
- Instruction codes together with data are stored in memory (Stored Program Concept)

Computer Instructions

Computer instructions are a set of machine language instructions that a particular processor understands and executes. A computer performs tasks on the basis of the instruction provided.

An instruction comprises of groups called fields. These fields include:



Computer Architecture & Assembly Language

BCA-303

- The Operation code (Opcode) field which specifies the operation to be performed.
- The Address field which contains the location of the operand, i.e., register or memory location.
- The Mode field which specifies how the operand will be located.

Computer Architecture & Assembly Language

BCA-303

A basic computer has three instruction code formats which are:

- Memory - reference instruction
- Register - reference instruction
- Input-Output instruction

Instruction Cycle (Fetch -decode execute cycle)

A program residing in the memory unit of a computer consists of a **sequence of instructions**.

These instructions are executed by the processor by going through a **cycle for each instruction.**

In a basic computer, each instruction cycle consists of the following phases:

Fetch instruction from memory.

Decode the instruction.

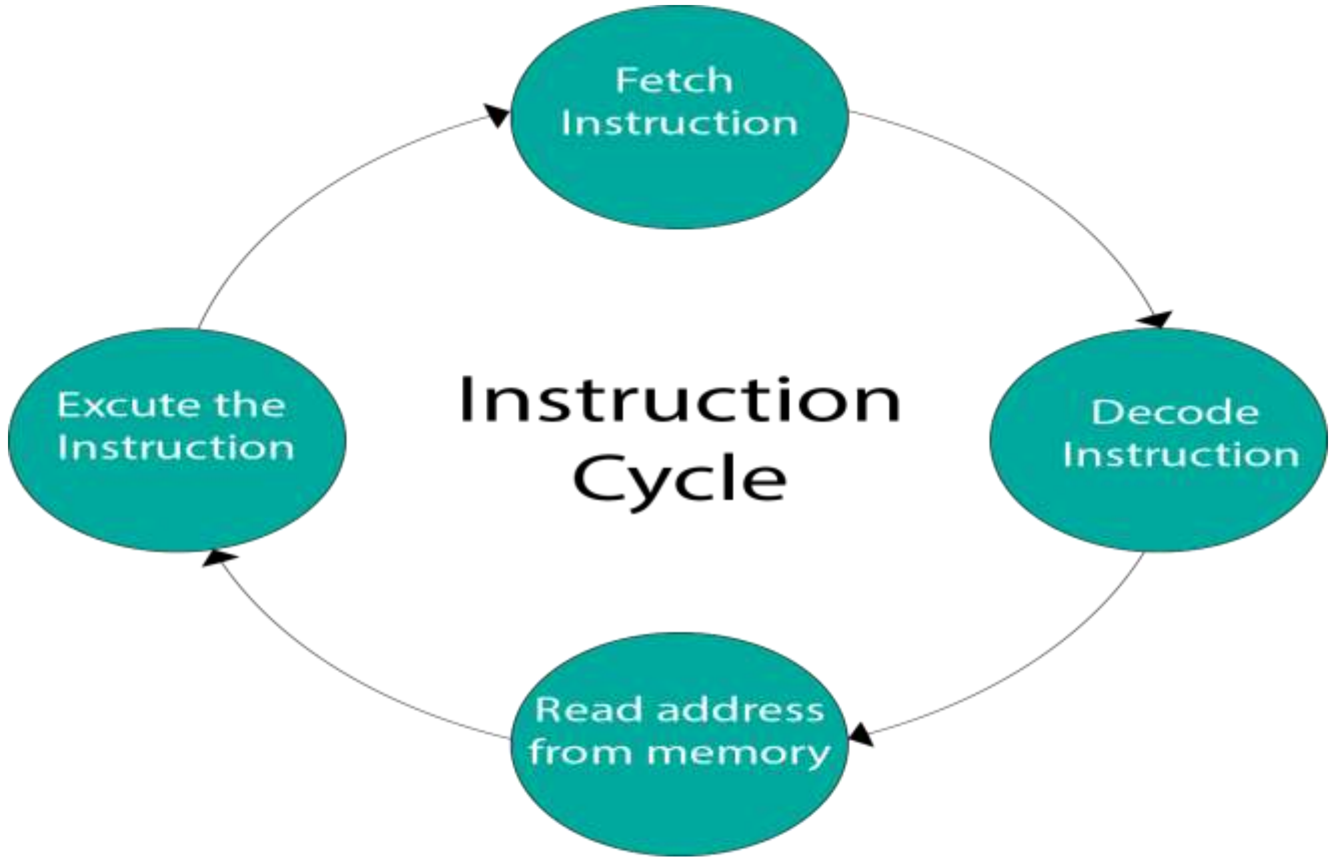
Read the effective address from memory.

Execute the instruction.

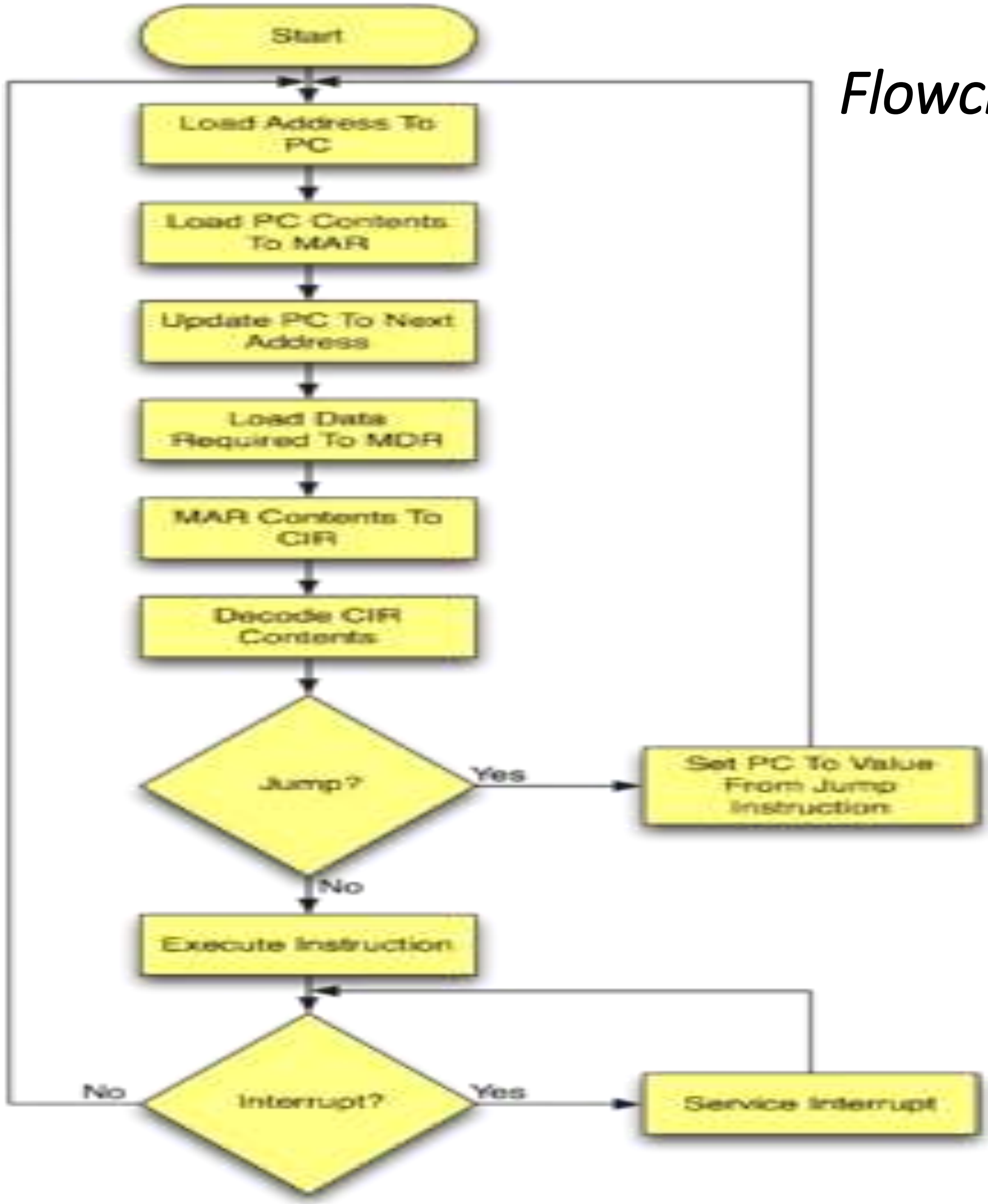
Computer Architecture & Assembly Language

BCA-303

Instruction Cycle



Flowchart of instruction cycle



GL BAJAJ
Institute of Management
FIND YOUR SPARK
Affiliated to CCS University, Meerut.

Computer Architecture & Assembly Language

BCA-303

Input-Output Configuration

In **computer architecture, input-output devices act as an interface between the machine and the user.**

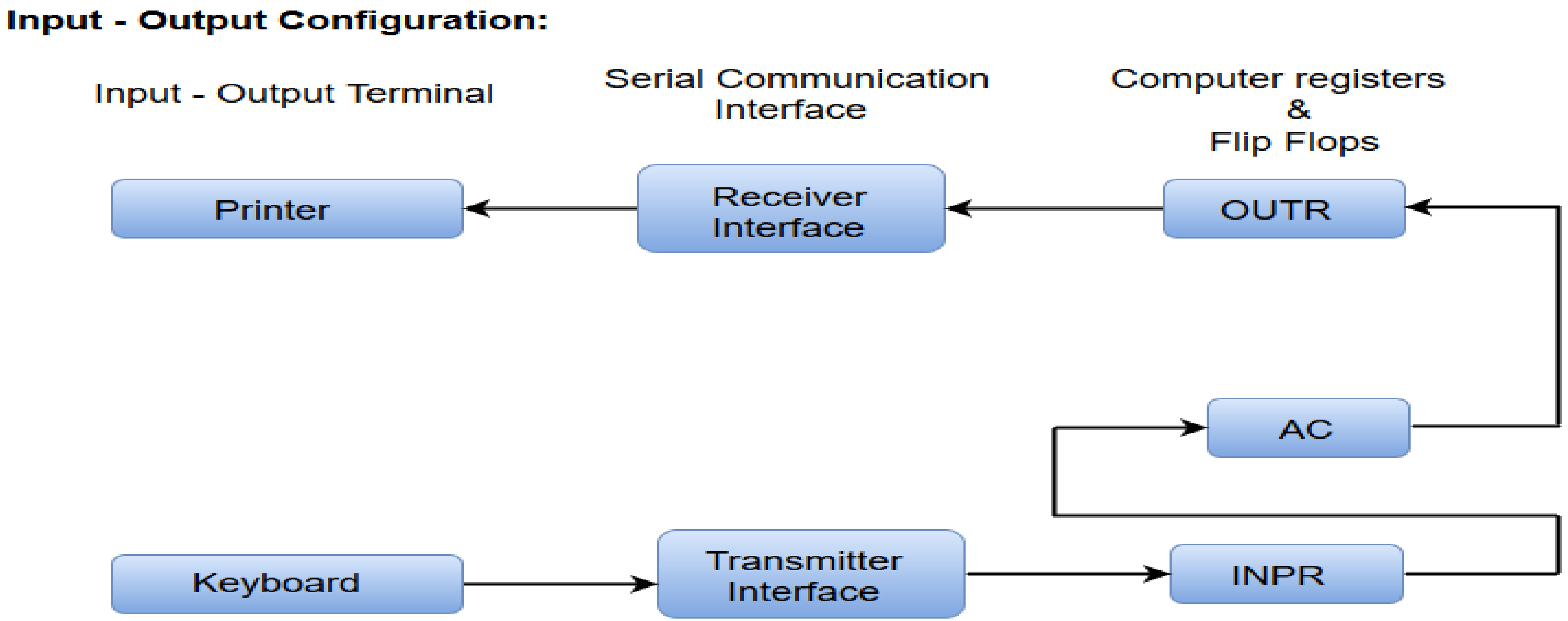
Instructions and data stored in the memory must come from some input device. The results are displayed to the user through some output device.

The following block diagram shows the input-output configuration for a basic computer.

Computer Architecture & Assembly Language

BCA-303

Input-Output Configuration


Input - Output Configuration:
Input - Output Terminal
Serial Communication Interface
Computer registers & Flip Flops
Printer
Receiver Interface
OUTR
AC
Keyboard
Transmitter Interface
INPR

Input-Output Configuration

- The input-output terminals send and receive information.
- The amount of information transferred will always have eight bits of an alphanumeric code.
- The information generated through the keyboard is shifted into an input register 'INPR'.
- The information for the printer is stored in the output register 'OUTR’.

GL BAJAJ
Institute of Management
FIND YOUR SPARK
Affiliated to CCS University, Meerut.

Computer Architecture & Assembly Language

BCA-303

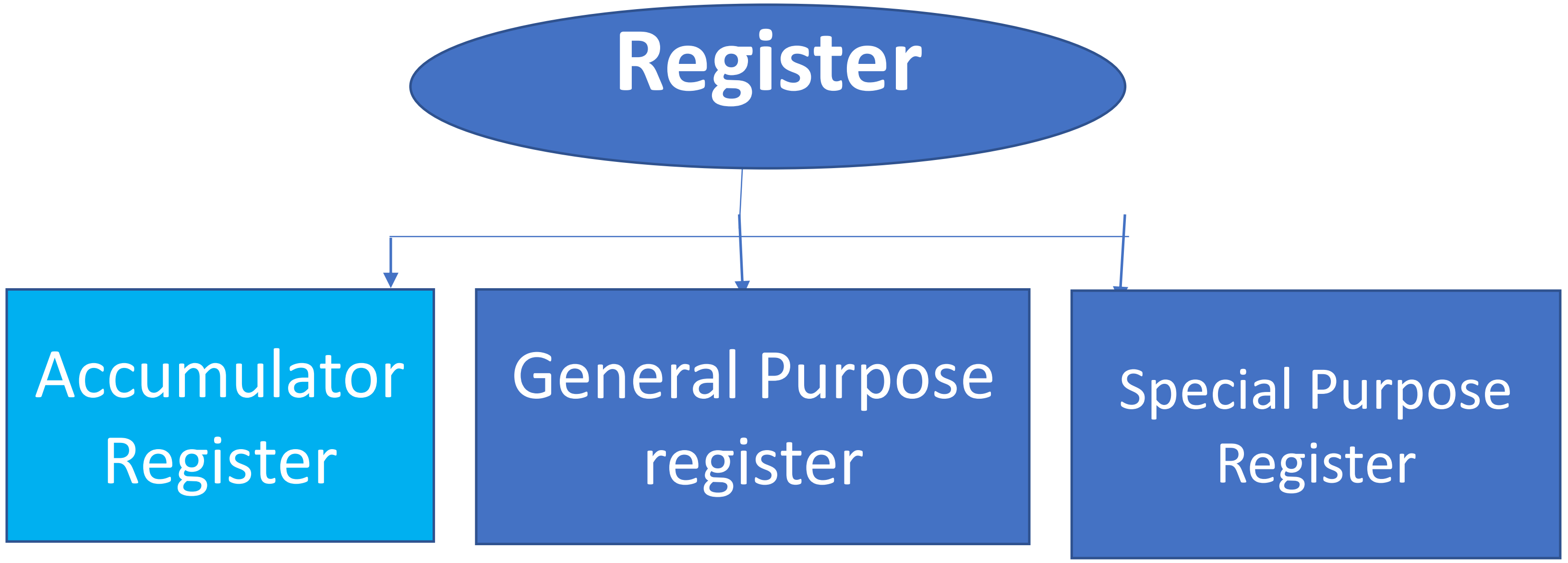
Input-Output Configuration

- Registers INPR and OUTR communicate with a communication interface serially and with the AC in parallel.
- The transmitter interface receives information from the keyboard and transmits it to INPR.
- The receiver interface receives information from OUTR and sends it to the printer serially.

Register and its type:-

A processor register (CPU Register) is one of smallest set of data holding place that are part of CPU processor. A register may hold instruction , storage address or any kind of data.Register is a very fast computer primary memory used to store data/instruction in execution.

TYPES:-



Accumulator Register :-

In computer central processing unit (CPU) an accumulator register is a register which perform connection b/w general purpose register and special purpose register.Accumulator register perform intermediate b/w arithmetic and logical operation.This register is used to storing the results those produce by CPU,when the CPU with generates some results after the processing then all result after the processing then all result will be stored in Accumulator register.

General Purpose Register:-

This is used to store data intermediate results during program execution . It can be excessed by user or assembly programming. General purpose register can store both data and address.

Special Purpose Register:-

User do not access these registers . These register reserved for computer system . These are many types special purpose register according to their purpose:-

1.MAR- Memory Address Register

2.PC-Program Counter.

3.MBR-Memory Buffer Register.

4.MDR-Memory Data Register.

5.IR-Instruction Register.

6. CIR-Current Instruction Register.

7.Index Register.

GL BAJAJ Institute of Management

FIND YOUR SPARK

Affiliated to CCS University, Meerut.

Computer Architecture & Assembly Language

BCA-303

These **registers are utilized for playing out the different operations**. **When we perform some operations, the CPU utilizes these registers to perform the operations.**

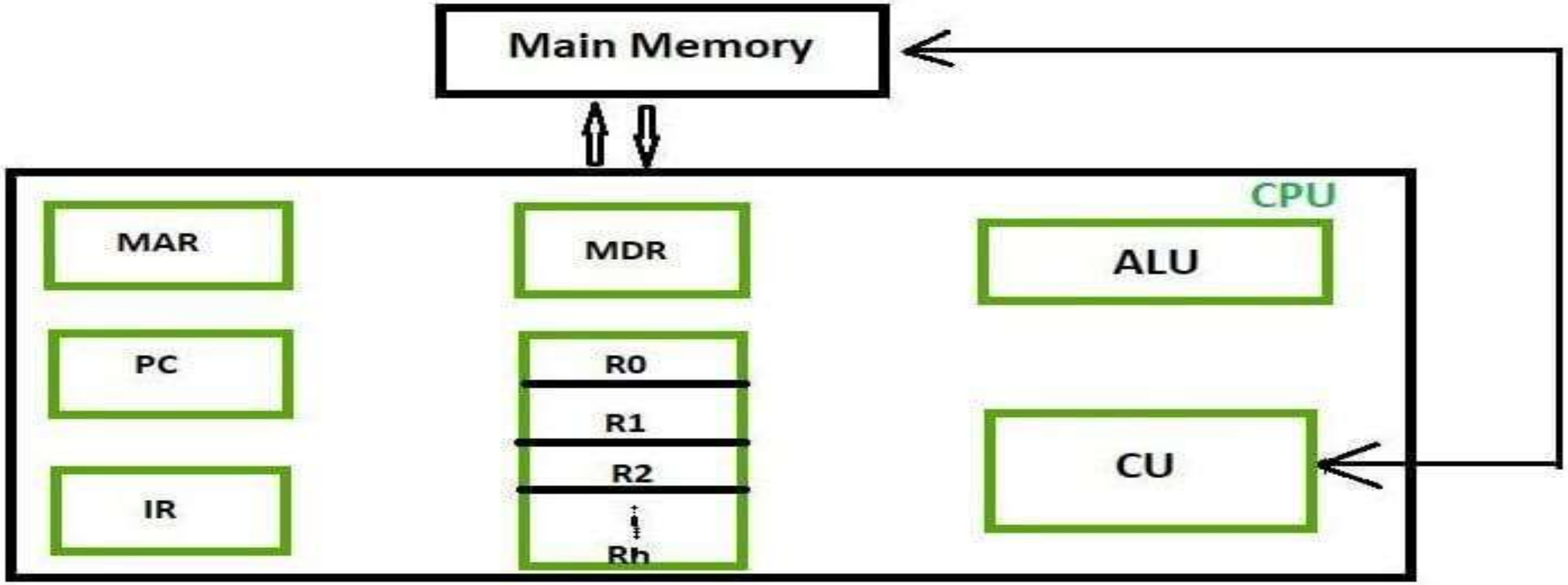
When we provide input to the system for a certain operation, the provided information or the input gets stored in the registers.

Once the ALU arithmetic and logical unit process the output, the processed data is again provided to us by the registers.

Computer Architecture & Assembly Language

BCA-303

Following major operations performed by registers, such as:



Computer Architecture & Assembly Language

BCA-303

Following major operations performed by registers, such as:

- **Fetch:** The fetch operation is utilized for taking the directions by the client. The instructions that are stored away into the main memory for later processing are fetched by registers.
- **Decode:** This operation is utilized for read the instructions , implies the instructions are decoded. the CPU will discover which operation is to be performed on the instructions.
- **Execute:** The CPU performs this operation. Also, results delivered by the CPU are then stored in the memory, and after that, they are shown on the client Screen.

GL BAJAJ Institute of Management
FIND YOUR SPARK
Affiliated to CCS University, Meerut.

Computer Architecture & Assembly Language

BCA-303

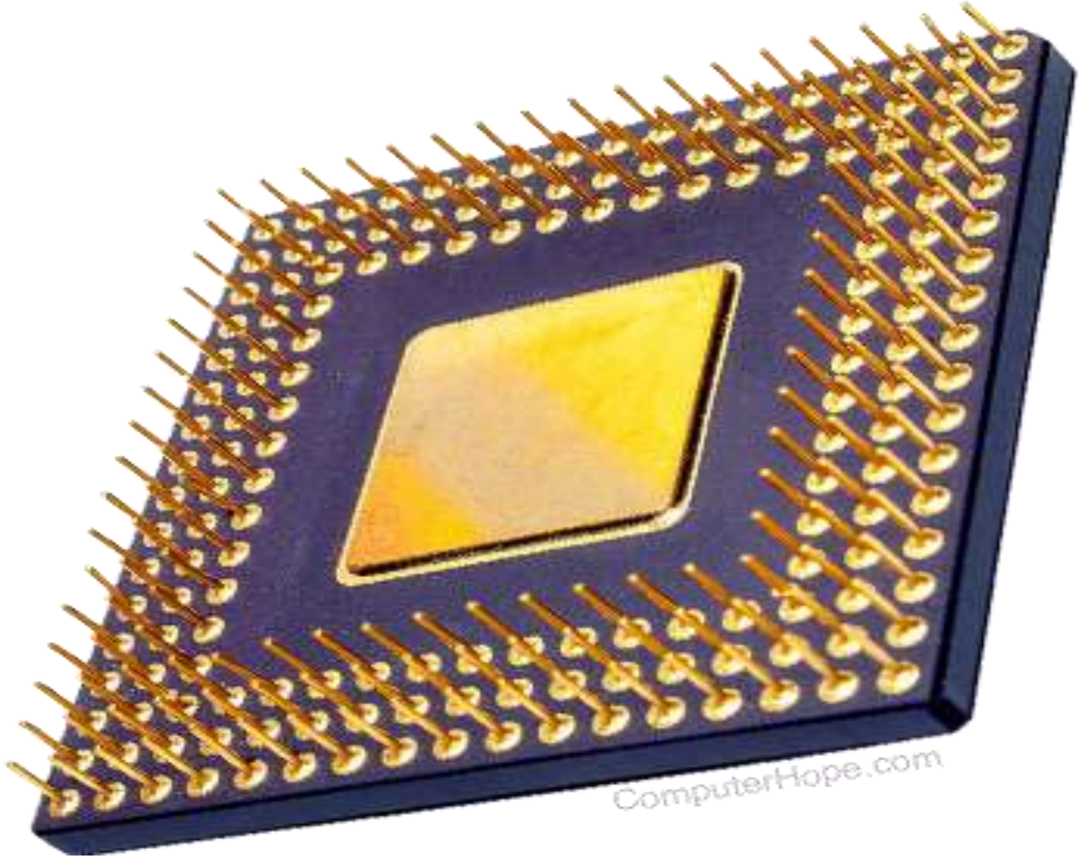
Accumulator Register (AC) : An **accumulator** is a type of register included in a CPU. **It acts as a temporary storage location which holds an intermediate value in mathematical and logical calculations.** Intermediate results of an operation are progressively written to the accumulator, overwriting the previous value. **For example, in the operation "3 + 4 + 5," the accumulator would hold the value 3, then the value 7, then the value 12.** The benefit of an accumulator is that it does not need to be clearly referenced, which conserves data in the operation statement.

GL BAJAJ
Institute of Management
FIND YOUR SPARK
Affiliated to CCS University, Meerut.

Computer Architecture & Assembly Language

BCA-303

Accumulator Register (AC) :

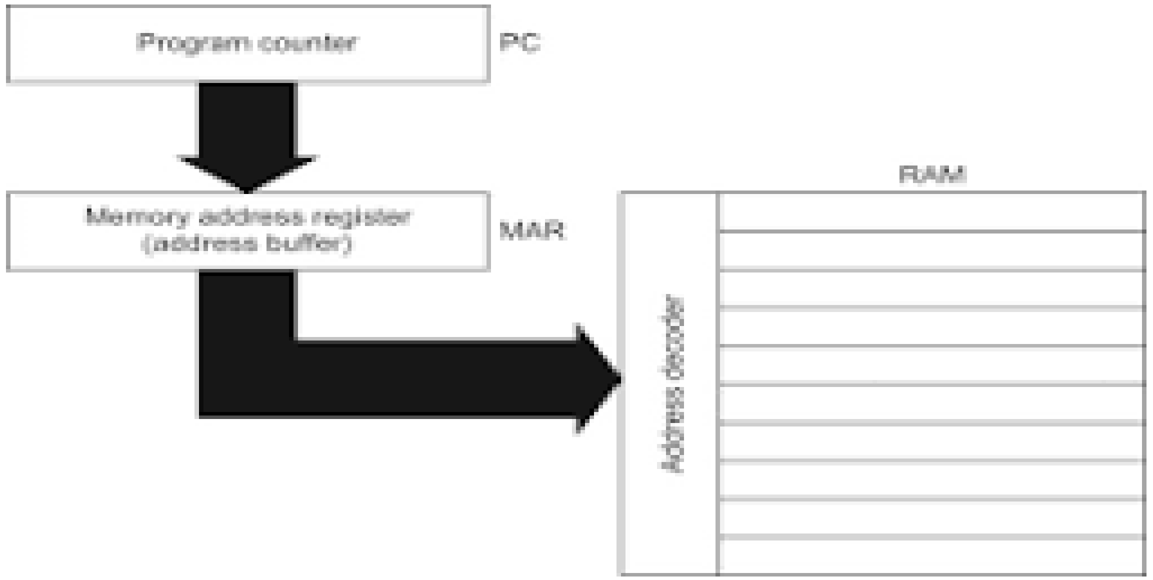


Computer Architecture & Assembly Language

BCA-303

Memory address registers(MAR)

Address location of memory is stored in this register to be accessed later.



- Computer Architecture & Assembly Language
- BCA-303

Memory data registers(MDR)

All the information that is supposed to be written or the information that is supposed to be read from a certain memory address is stored here.

Current Instruction registers(CIR)

In Instruction register which process or called currently being executed or decoded that register is called CIR.

Program Counter (PC):

It is a 16-bit special purpose register. **It is used to hold the address of memory of the next instruction to be executed.** It keeps the track of the instruction in a program while they are being executed. The microprocessor increments the content of the next program counter during the execution of an instruction so that at the end of the execution of an instruction it points to the next instructions address in the program

Instruction Register: The instruction register holds the opcode (operation code or instruction code) of the instruction which is being decoded and executed.

INDEX Register:-

An **Index Register** in a CPU is a processing register used for modifying operand addresses during the run of program or execute the program.A hardware element which hold a number That can be added to (in case of subtract form).the address portion of a computer instruction to form an effective address.It is also known as **BASE REGISTER**.

Stack Pointer (SP)

It is a 16-bit special function register used as memory pointer. A stack is nothing but a portion of RAM. In the stack, the contents of only those registers are saved, which are needed in the later part of the program. The stack pointer (SP) controls the addressing of the stack. The Stack Pointer contains the address of the top element of data stored in the stack.

Computer Architecture & Assembly Language

BCA-303

A **Register** is a group of flip-flops with each flip-flop capable of storing **one bit** of information. An *n-bit* register has a group of *n flip-flops* and is capable of storing binary information of *n-bits*.

A flip-flop is **a device which stores a single bit (binary digit) of data**; one of its two states represents a "one" and the other represents a "zero". Such data storage can be used for storage of state, and such a circuit is described as sequential logic in electronics.

GL BAJAJ
Institute of Management
FIND YOUR SPARK
Affiliated to CCS University, Meerut.

Computer Architecture & Assembly Language

BCA-303

REGISTER TRANSFER AND MICROOPERATIONS

The term Register Transfer refers to the availability of hardware logic circuits that can perform a given micro-operation and transfer the result of the operation to the same or another register.

Most of the standard notations used for specifying operations on various registers are stated below.

- The memory address register is designated by **MAR**.

Computer Architecture & Assembly Language

BCA-303

REGISTER TRANSFER AND MICROOPERATIONS

Most of the standard notations used for specifying operations on various registers are stated below.

- Program Counter **PC** holds the next instruction's address.
- Instruction Register **IR** holds the instruction being executed.
- **R1** (Processor Register).
- We can also indicate individual bits by placing them in parenthesis. For instance, PC (8-15), R2 (5), etc.

REGISTER TRANSFER AND MICROOPERATIONS

- Data Transfer from one register to another register is represented in symbolic form by means of replacement operator. For instance, the following statement denotes a transfer of the data of register R1 into register R2.

$$R2 \leftarrow R1$$

Typically, most of the users want the transfer to occur only in a predetermined control condition. This can be shown by following if-then statement: **If (P=1) then** ($R2 \leftarrow R1$); Here P is a control signal generated in the control section.

REGISTER TRANSFER AND MICROOPERATIONS

It is more convenient to specify a control function (P) by separating the control variables from the register transfer operation. For instance, the following statement defines the data transfer operation under a specific control function (P).

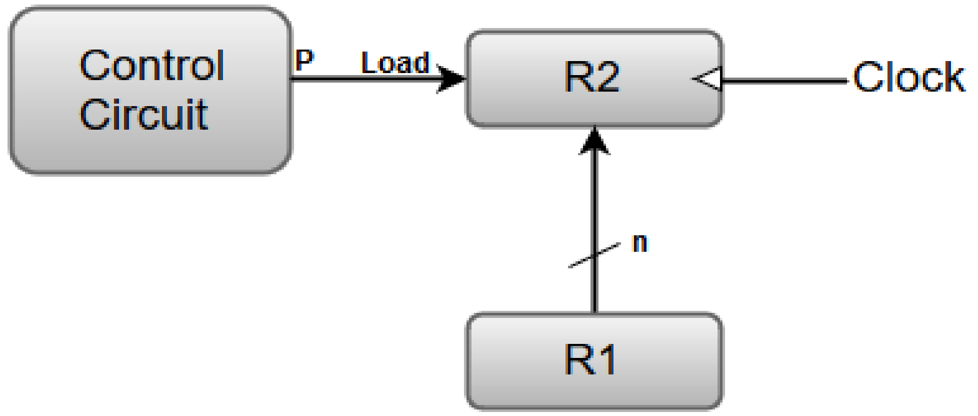
$P: \ R2 \leftarrow R1$

REGISTER TRANSFER AND MICROOPERATIONS

The following image shows the block diagram that depicts the transfer of data from R1 to R2.

Transfer from R1 to R2 when P = 1:

Here, the letter 'n' indicates the number of bits for the register. The 'n' outputs of the register R1 are connected to the 'n' inputs of register R2.



Computer Architecture & Assembly Language

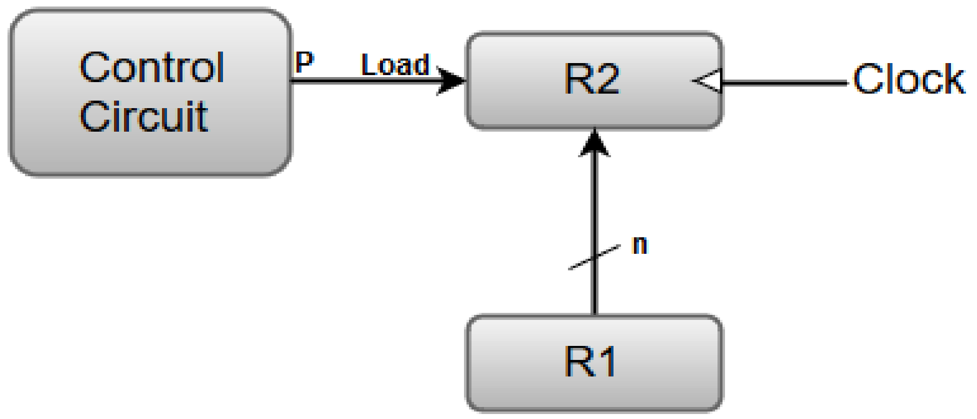
BCA-303

REGISTER TRANSFER AND MICROOPERATIONS

The following image shows the block diagram that depicts the transfer of data from R1 to R2.

Transfer from R1 to R2 when P = 1:

Here, the letter 'n' indicates the number of bits for the register. The 'n' outputs of the register R1 are connected to the 'n' inputs of register R2.



Bus and Memory Transfers

A **digital system composed of many registers**, and paths must be provided to transfer information from one register to another. The number of wires connecting all of the registers will be excessive if separate lines are used between each register and all other registers in the system.

A **bus consists of a set of common lines**, one for each bit of register, through which binary information is transferred one at a time. **Control signals determine which register is selected by the bus during a particular register transfer.**

GL BAJAJ Institute of Management
FIND YOUR SPARK
Affiliated to CCS University, Meerut.

Computer Architecture & Assembly Language

BCA-303

Bus and Memory Transfers

The following block diagram shows a **Bus system for four registers**. It is constructed with the help of four **4 * 1 Multiplexers** each having four data inputs (0 through 3) and two selection inputs (S1 and S2).

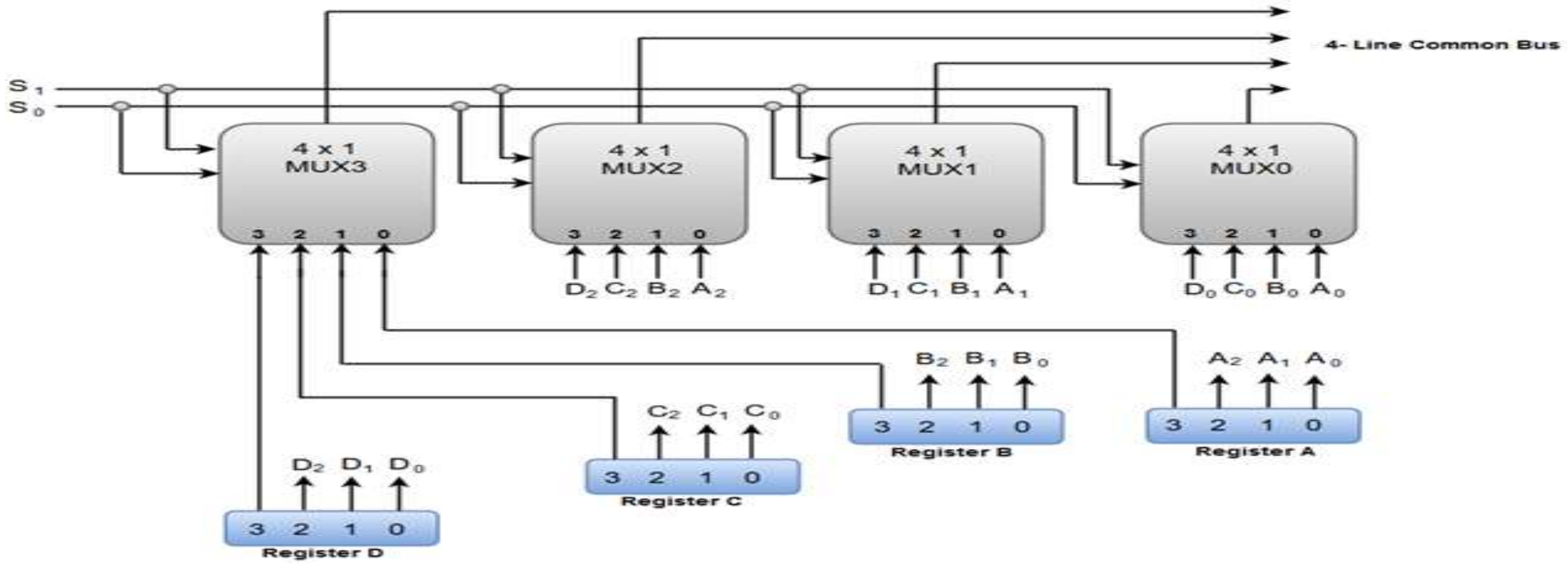
We have used labels to make it more convenient for you to understand the input-output configuration of a Bus system for four registers. For instance, output 1 of register A is connected to input 0 of MUX1.

GL BAJAJ Institute of Management
FIND YOUR SPARK
Affiliated to CCS University, Meerut.

Computer Architecture & Assembly Language

BCA-303

Bus System for 4 Registers:



Computer Architecture & Assembly Language

BCA-303

The two selection lines S1 and S2 are connected to the selection inputs of all four multiplexers. The selection lines choose the four bits of one register and transfer them into the four-line common bus.

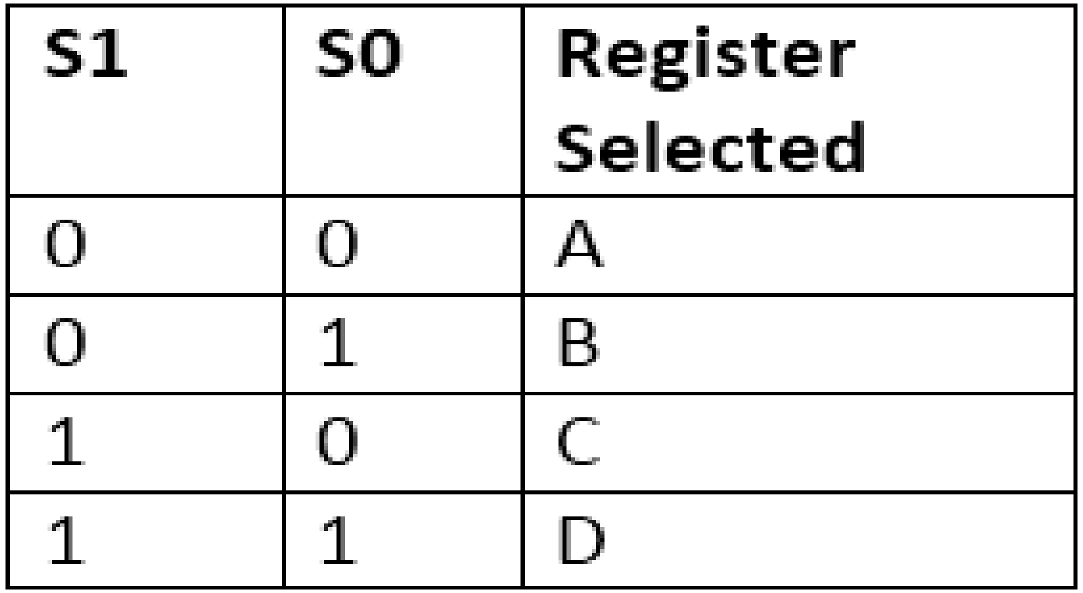
When both of the select lines are at low logic, i.e. **S1S0 = 00, the 0 data inputs of all four multiplexers are selected and applied to the outputs that forms the bus.** This, in turn, causes the bus lines to receive the content of register A since the outputs of this register are connected to the 0 data inputs of the multiplexers.

Similarly, when S1S0 = 01, register B is selected, and the bus lines will receive the content provided by register B.

The following function table shows the register that is selected by the bus for each of the four possible binary values of the Selection lines.

Computer Architecture & Assembly Language

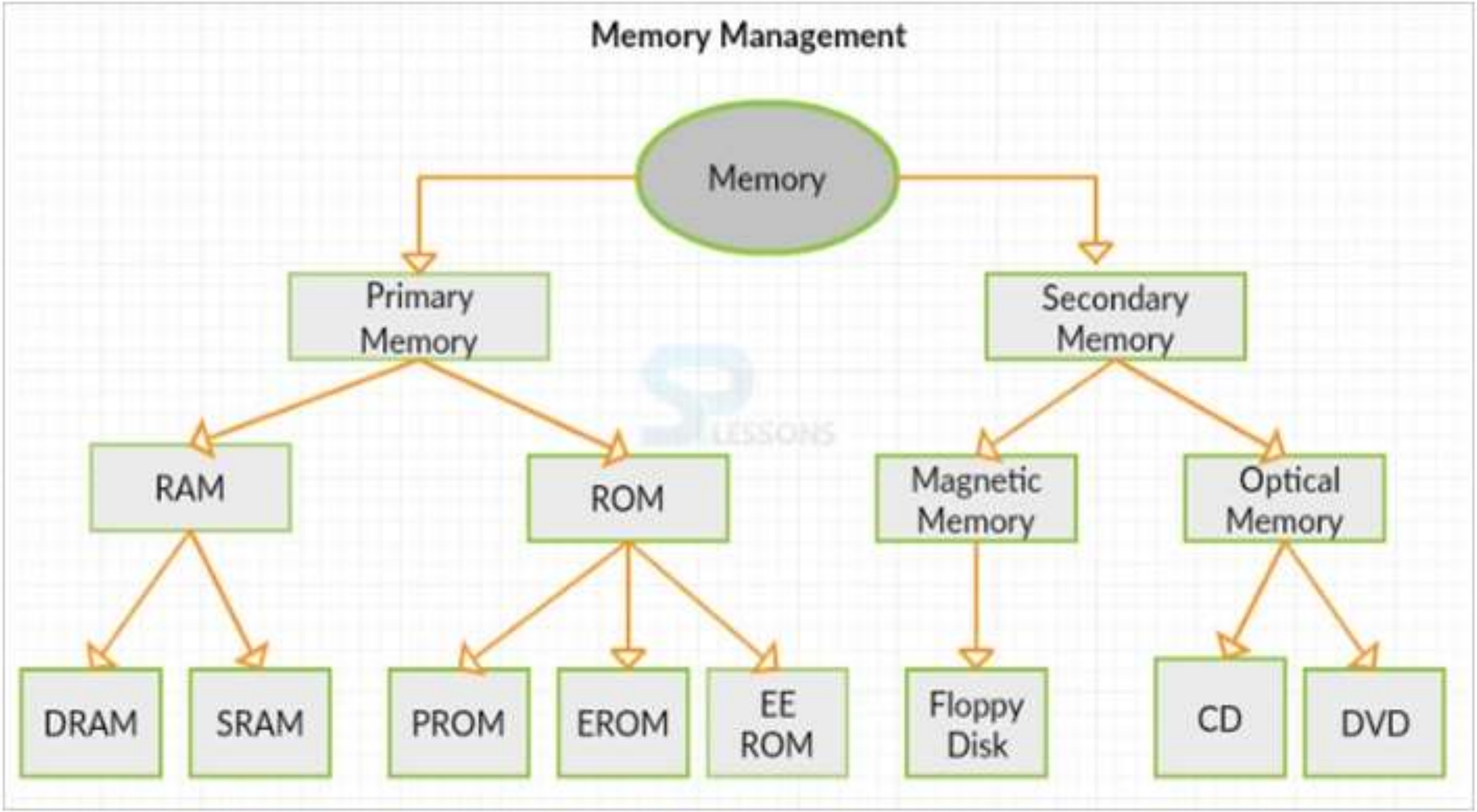
BCA-303



S1	S0	Register Selected
0	0	A
0	1	B
1	0	C
1	1	D

What is Memory?

Computer memory is just like the human brain. It is used to store data/information and instructions. It is a data storage unit or a data storage device where data is to be processed and instructions required for processing are stored. It can store both the input and output can be stored here.


Memory Management
Memory
Primary Memory
Secondary Memory
RAM
ROM
Magnetic Memory
Optical Memory
DRAM
SRAM
PROM
EROM
EE ROM
Floppy Disk
CD
DVD

Characteristics of Main Memory:

- It is faster computer memory as compare to secondary memory.
- It is semiconductor memories.
- It is usually a volatile memory.
- It is the main memory of the computer.
- A computer system cannot run without primary memory.

In general, memory is of three types:

- Primary memory
- Secondary memory
- Cache memory

Primary Memory: It is also known as the main memory of the computer system. It is used to store data and programs or instructions during computer operations. It uses semiconductor technology and hence is commonly called semiconductor memory. Primary memory is of two types:

RAM (Random Access Memory): It is a volatile memory. Volatile memory stores information based on the power supply. If the power supply fails/ interrupted/stopped, all the data & information on this memory will be lost. RAM is used for booting up or start the computer. It temporarily stores programs/ data which has to be executed by the processor. **RAM is of two types:**

Computer Architecture & Assembly Language

BCA-303

S RAM (Static RAM): It uses transistors and the circuits of this memory are capable of retaining their state as long as the power is applied. This memory consists of the number of flip flops with each flip flop storing 1 bit. It has less access time and hence, it is faster.

D RAM (Dynamic RAM): It uses capacitors and transistors and stores the data as a charge on the capacitors. They contain thousands of memory cells. It needs refreshing of charge on capacitor after a few milliseconds. This memory is slower than S RAM.

ROM (Read Only Memory): It is a non-volatile memory. Non-volatile memory stores information even when there is a power supply failed/ interrupted/stopped. ROM is used to store information that is used to operate the system. As its name refers to read-only memory, we can only read the programs and data that is stored on it. ROM is of four types:

MROM(Masked ROM): Hard-wired devices with a pre-programmed collection of data or instructions were the first ROMs. Masked ROMs are a type of low-cost ROM that works in this way.

Computer Architecture & Assembly Language

BCA-303

EPROM (Erasable Programmable Read Only Memory): It is an extension to PROM where you can erase the content of ROM by exposing it to Ultraviolet rays for nearly 40 minutes.

EEPROM (Electrically Erasable Programmable Read Only Memory): Here the written contents can be erased electrically. You can delete and reprogramme EEPROM up to 10,000 times.

Secondary Memory: It is also known as **auxiliary memory** and backup memory. It is a **non-volatile memory and used to store a large amount of data or information**. The **data or information stored in secondary memory is permanent, and it is slower than primary memory.** A CPU cannot access secondary memory directly. The data/information from the auxiliary memory is first transferred to the main memory, and then the CPU can access it.

Characteristics of Secondary Memory:

It is a slow memory but reusable.

It is a reliable and non-volatile memory.

It is cheaper than primary memory.

The storage capacity of secondary memory is large.

A computer system can run without secondary memory.

In secondary memory, data is stored permanently even when the power is off.

Computer Architecture & Assembly Language

BCA-303

Types of secondary memory:

Magnetic Tapes: Magnetic tape is a long, narrow strip of plastic film with a thin, magnetic coating on it that is used for magnetic recording. Bits are recorded on tape as magnetic patches called RECORDS that run along many tracks.

Magnetic Disks: A magnetic disc is a circular metal or a plastic plate and these plates are coated with magnetic material. The disc is used on both sides. Bits are stored in magnetized surfaces in locations called tracks that run in concentric rings.

Types of secondary memory:

Optical Disks: It’s a laser-based storage medium that can be written to and read. It is reasonably priced and has a long lifespan. The optical disc can be taken out of the computer by occasional users. **Types of Optical Disks :**

CD – ROM:

- It’s called Compact Disk. Only read from memory.
- Information is written to the disc by using a controlled laser beam to burn pits on the disc surface.
- The diameter of the disc is 5.25 inches.
- 16000 tracks per inch is the track density.
- The capacity of a CD-ROM is 600 MB, with each sector storing 2048 bytes of data.
- The data transfer rate is about 4800KB/sec. & the new access time is around 80 milliseconds.

WORM-(WRITE ONCE READ MANY):

- A user can only write data once.
- The information is written on the disc using a laser beam.
- It is possible to read the written data as many times as desired.
- They keep lasting records of information but access time is high.
- Data that has already been written cannot be changed.
- Usual size – 5.25 inch or 3.5 inch diameter.
- The usual capacity of 5.25 inch disk is 650 MB,5.2GB etc.

DVDs:

The term "DVD" stands for "Digital Versatile/Video Disc," and there are two sorts of DVDs:

(i) DVDR (writable) and

(ii) DVDRW (Re-Writable)

Cache Memory: It is a type of high-speed semiconductor memory that can help the CPU run faster. Between the CPU and the main memory, it serves as a buffer. It is used to store the data and programs that the CPU uses the most frequently.

Advantages of cache memory:

- It is faster than the main memory.
- When compared to the main memory, it takes less time to access it.
- It keeps the programs that can be run in a short amount of time.
- It stores data in temporary use.

CPU is much faster than main memory .To overcome this speed mis match between CPU and main memory , a cache memory , whose access time is close to the CPU processing speed , is introduced between the CPU and main memory . Now , instead of everytime reading from the main memory , CPU first check the cache memory for the required data . If it is present , the data is read from cache memory at a very high speed . If the data is not present ,it is read from the main memory and put in the cache for further readings.

Cost of cache memory is very high as compared to main memory. Therefore , capacity of cache memory is much smaller as compared to main memory.

Principal of cache :-

A) Temporal Locality Reference

B) Spatial Locality Reference

TEMPORAL LOCALITY REFERENCE

It is states that recently accessed data and instructions are likely to be accessed in the near future . For example , look at the case of loop , when a program loop is executed, the same set of instructions are referenced and fetched repeatedly.

SPATIAL LOCALITY REFERENCE :-

It is states that data and instruction which are close to each other are likely to be accessed in near future. For Example, in a program the instruction are stored in memory locations also when in a program array are used , values are stored in same memory locations.

Hence ,it can be said that a processor spends 90% of its time in 10 % of memory . Now if we can place this 10 % of memory in cache memory , the memory access time would be greatly reduced.

The performance of cache memory is frequently measured in terms of a quantity called **hit ratio.**

If the word found in cache then it produced hit otherwise it produce /count aas a **miss.**

So Hit Ratio (h) =

$$\frac{\textbf{Number of hits}}{\textbf{Number of hits + Number of misses}}$$

In generally , a hit ratio of at least 0.8 is desired.

Performance of cache memory can be calculated as:-

$$Te = hT\,c + (1-h)\,Tm$$

Where ,

h : hit ratio

Te : Effective memory access time in cache memory system

Tc : Cache access time

Tm : Main memory access time

For example If Tc = 100 ns

Tm = 500 ns

h = 80 % (i.e. 0.8)

Then Te = 0.8 *100 +(1-0.8)*500

= 180 ns.

Cache mapping :-

There are three different way for mapping

1. Associative mapping (Free Mapping)
2. Direct mapping (Fixed Mapping)
3. Set associative Mapping

1. Associative mapping (Free Mapping) :-

In this mapping both word and address of the any word are stored in the cache (same as in the main memory). Address

of word In cache memory is also known as **address tag**. Now , instead of everytime reading from the main memory , CPU first check the cache memory for the required data . If it is present , the data is read from cache memory at a very high speed . If the data is not present ,it is read from the main memory and put in the cache for further readings.

Associative mapping use some replacement algoritham:-

- 1. Word Replacement :-

The designer chooses the replacement algorithm which determines which word is replaced.

Let us discuss two replacement algorithm

A) **FIFO** :- In this algorithm , the word which arrived first i.e the oldest word , is removed when a new word needs to be brought to the cache memory.

B) **Least Recently used (LRU)** : This algorithm chooses the word that has been used by the CPU minimum number of times in the recent past.

2.Direct Mapping :-

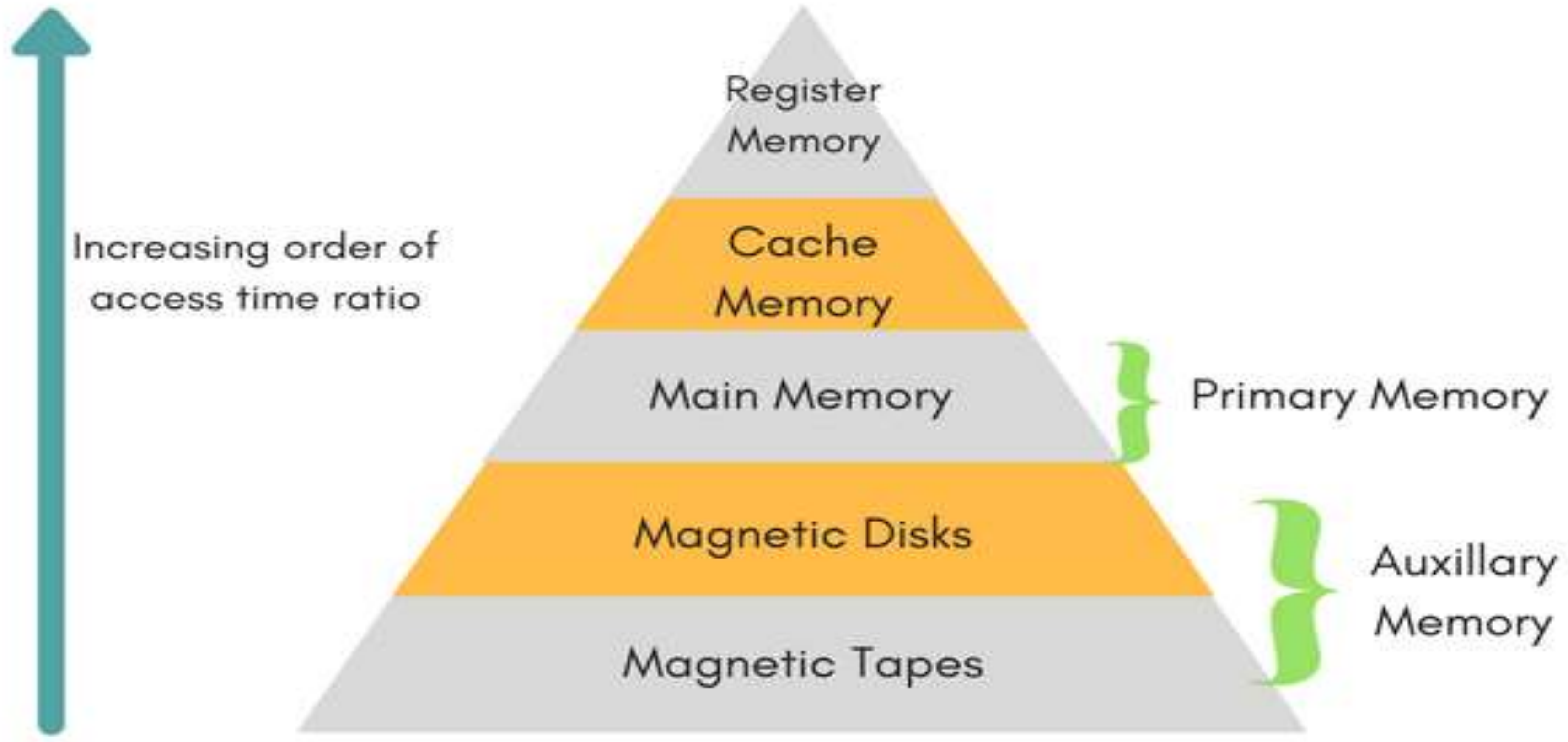
In direct mapping method , the address sent by CPU is divided into two parts called **Tag Field and Index Field . The Index Field has number of bits equal to the number of bits required to address a word in cache.** In direct mapping method , the cache memory stores the word as the tag field. The words will be stored at that location in cache which is represented by the index field of their addresses.

3. Set Associative Memory :-

The Problem of direct mapping is that two word with the same index in their address but with different tag values cannot reside in cache memory at the same time. The problem is overcome in set associative mapping where more than one word –tag pair can be stored in cache against a single index . If two word tag can be stored , it is called a two-way set associative mapping and so on……

Computer Architecture & Assembly Language

BCA-303



MICRO -OPERTIONS:-

Micro operation in computer CPU micro-operation are the functional and atomic operations of a processor. Micro –operation are low level instructions and perform basic operation on data(which is stored in register) and also transfer the data between registers and buses.

TYPES OF MICRO-OPERATION :-

1. **Arithmetic Micro operation.**
2. **LOGIC Micro operation.**
3. **SHIFT Micro operation.**
4. **REGISTER TRANSFER.**

Computer Architecture & Assembly Language

BCA-303

Arithmetic Micro-operations

In general, the Arithmetic Micro-operations deals with the operations performed on numeric data stored in the registers.

The basic Arithmetic Micro-operations are classified in the following categories:

1. Addition
2. Subtraction
3. Increment
4. Decrement
5. Shift

GL BAJAJ Institute of Management
FIND YOUR SPARK
Affiliated to CCS University, Meerut.

Computer Architecture & Assembly Language

BCA-303

Arithmetic Micro-operations

Some additional Arithmetic Micro-operations are classified as:

1. Add with carry
2. Subtract with borrow
3. Transfer/Load, etc.

Computer Architecture & Assembly Language

BCA-303

The following table shows the symbolic representation of various Arithmetic Micro-operations.

Symbolic Representation	**Description**
R3 ← R1 + R2	The contents of R1 plus R2 are transferred to R3.
R3 ← R1 - R2	The contents of R1 minus R2 are transferred to R3.
R2 ← R2'	Complement the contents of R2 (1's complement)

Computer Architecture & Assembly Language

BCA-303

Symbolic Representation	Description
R2 ← R2' + 1	2's complement the contents of R2
R3 ← R1 + R2' + 1	R1 plus the 2's complement of R2
R1 ← R1 + 1	Increment the contents of R1 by one
R1 ← R1 - 1	Decrement the contents of R1 by one

Computer Architecture & Assembly Language

BCA-303

Shift Micro-Operations

Shift micro-operations are those micro-operations that are used for serial transfer of information. These are also used in conjunction with arithmetic micro-operation, logic micro-operation, and other data-processing operations.

There are three types of shifts micro-operations:

Shift Micro-Operations

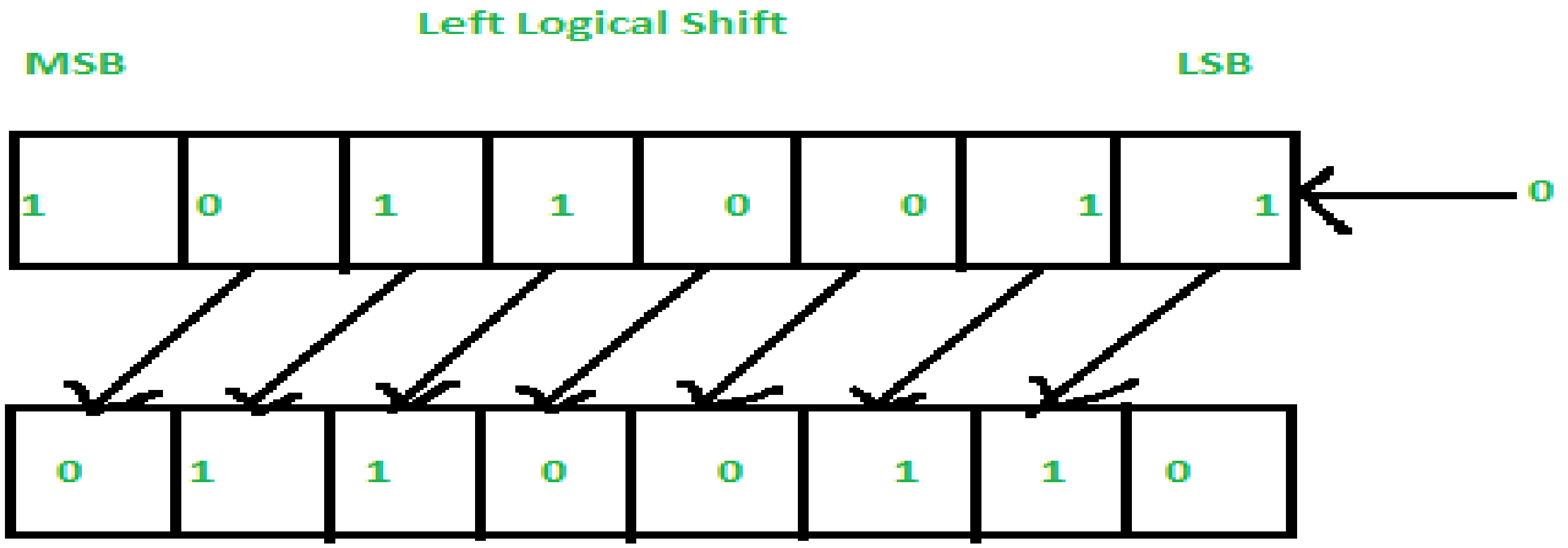
Logical

It transfers the 0 zero through the serial input. We use the symbols **shl** for logical shift-left and **shr** for shift-right.

Logical Shift Left –

In this shift one position moves each bit to the left one by one. The Empty least significant bit (LSB) is filled with zero (i.e., the serial input), and the most significant bit (MSB) is rejected.

Shift Micro-Operations



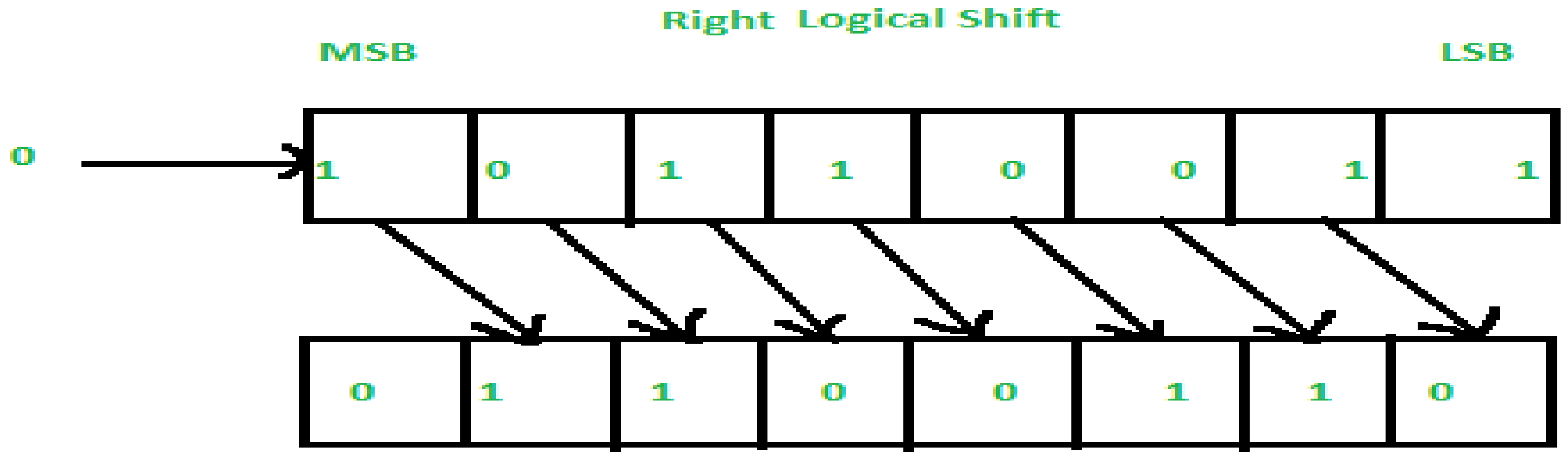
GL BAJAJ
Institute of Management
FIND YOUR SPARK
Affiliated to CCS University, Meerut.

Computer Architecture & Assembly Language

BCA-303

Right Logical Shift –

In this one position moves each bit to the right one by one and the least significant bit(LSB) is rejected and the empty MSB is filled with zero.



GL BAJAJ Institute of Management
FIND YOUR SPARK
Affiliated to CCS University, Meerut.

Computer Architecture & Assembly Language

BCA-303

Arithmetic Micro-Operations :-

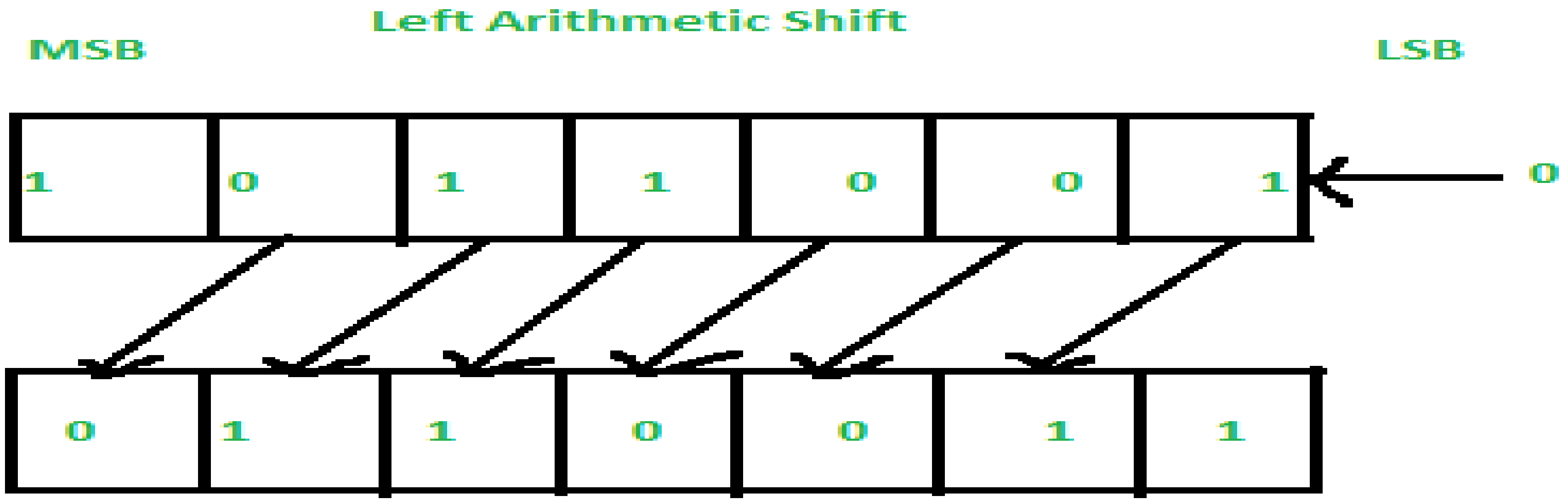
This micro-operation shifts a signed binary number to the left or to the right position. In an arithmetic shift-left, it multiplies a signed binary number by 2 and In an arithmetic shift-right, it divides the number by 2.

Left Arithmetic Shift –

In this one position moves each bit to the left one by one. The empty least significant bit (LSB) is filled with zero and the most significant bit (MSB) is rejected. Same as the Left Logical Shift.

Computer Architecture & Assembly Language

BCA-303



Computer Architecture & Assembly Language

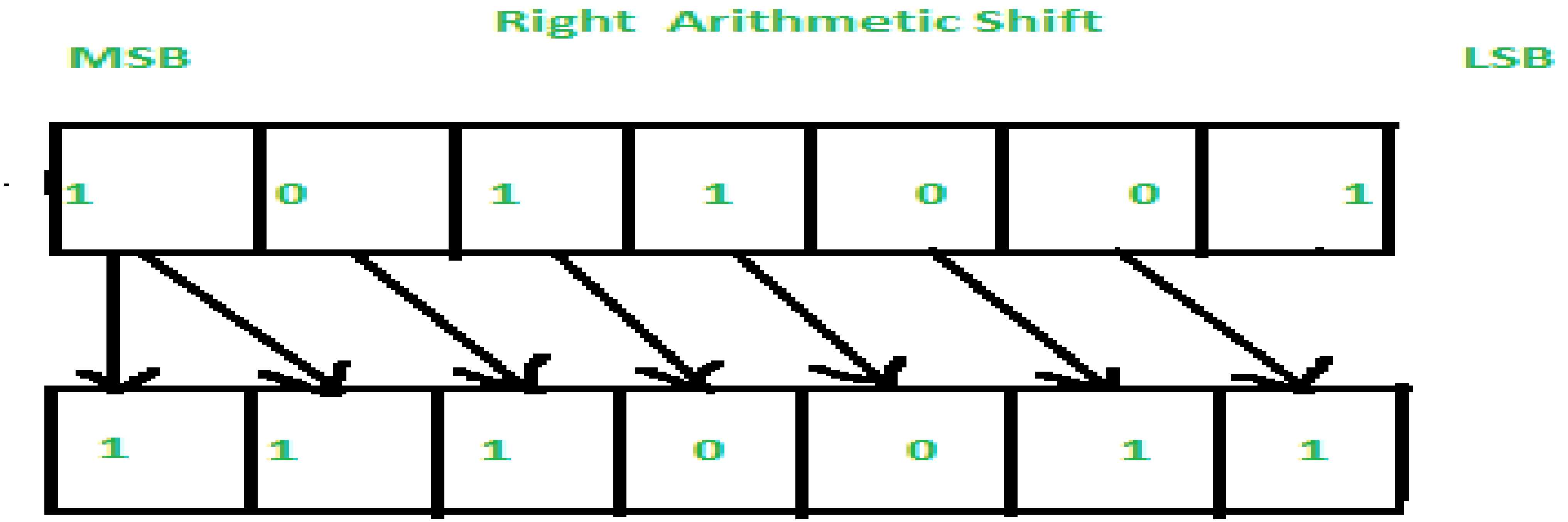
BCA-303

Right Arithmetic Shift –

In this one position moves each bit to the right one by one and the least significant bit is rejected and the empty MSB is filled with the value of the previous MSB.

Computer Architecture & Assembly Language

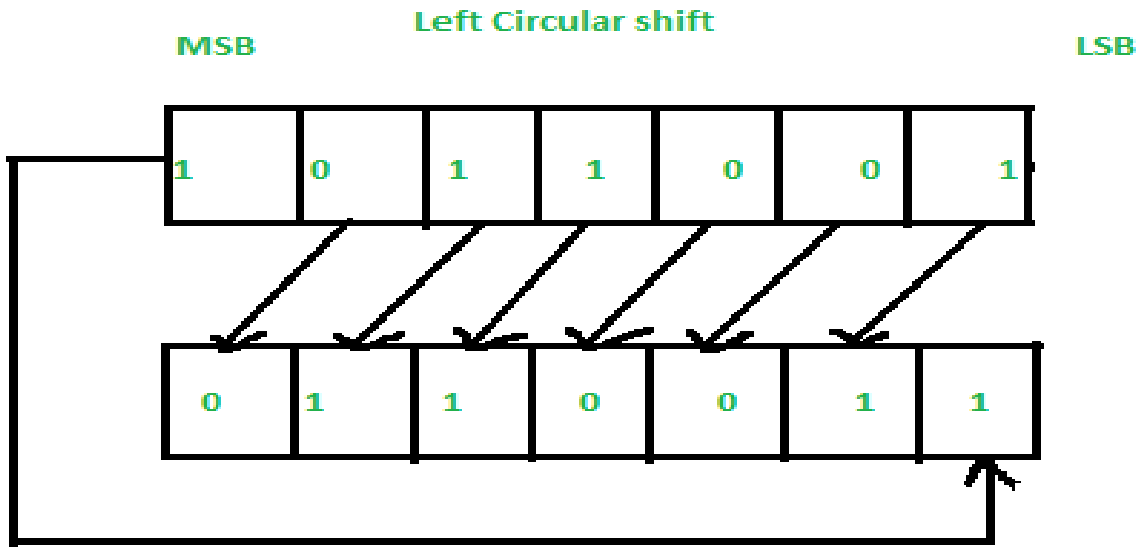
BCA-303



Circular :

The circular shift circulates the bits in the sequence of the register around the both ends without any loss of information.

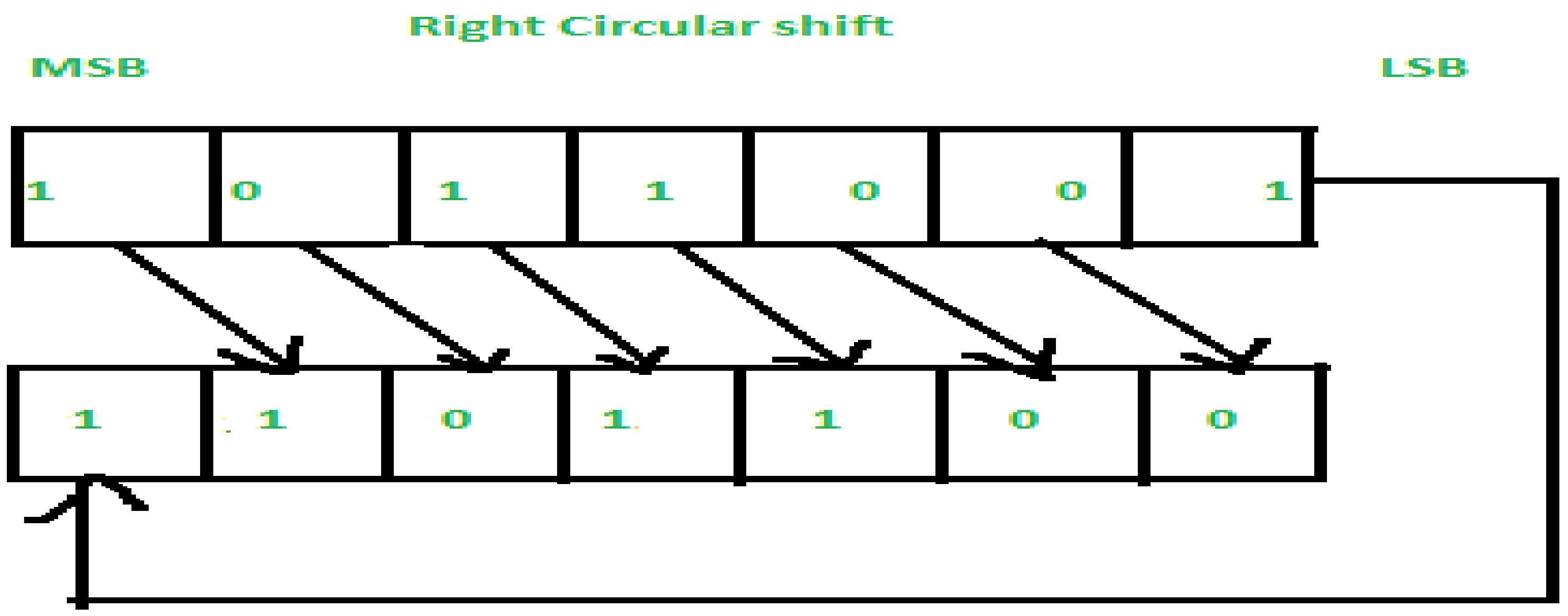
Left Circular Shift –



Computer Architecture & Assembly Language

BCA-303

Right Circular Shift –



Register transfer micro-operations :-

Information transferred from one register to another register (one source to another source) is designed in a symbolic form by means of replacement operator.

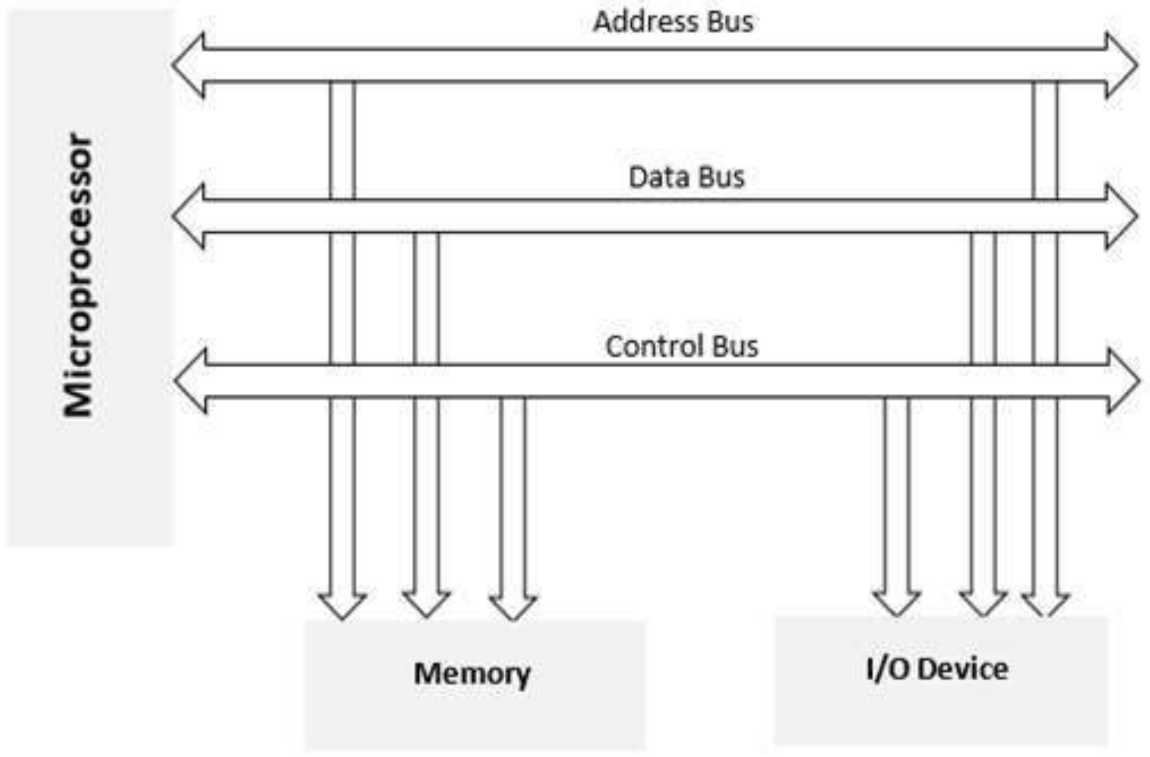
Computer Architecture & Assembly Language

BCA-303

Memory and I/O Interfacing

Several memory chips and I/O devices are connected to a microprocessor.

The following figure shows a schematic diagram to interface memory chips and I/O devices to a microprocessor.



GL BAJAJ Institute of Management
FIND YOUR SPARK
Affiliated to CCS University, Meerut.

Computer Architecture & Assembly Language

BCA-303

Memory and I/O Interfacing

Memory Interfacing

When we are executing any instruction, the address of memory location or an I/O device is sent out by the microprocessor. The corresponding memory chip or I/O device is selected by a decoding circuit.

Memory requires some signals to read from and write to registers and microprocessor transmits some signals for reading or writing data.

BUSES:-

Components communicate to each other using buses . A bus is a set of Parallel wires connecting different components .The processor is connected to the main memory by three separate buses.

The three type of bus are:-

1) **Address Buses :-** Address buses is a path or a set of parallel wires that carries the location of data to be read from or written to. An address bus is one directional only that is processor to the memory. The number of lines for address buses determines the maximum number of bits it can carry , and it turn determines the maximum possible memory capacity of the computer . EXAMPLE :-Think if the postal office only allows 3 digits for house numbers , the maximum addressable house will be 999 so if the computer has 32 -bit buses ,the maximum addressable memory locations will be 2^32 which is 4 GB , assuming each memory location stores one byte of data.

2) Data Buses :-

- Data bus is a bi-directional pathways or wires that carries data or instructions between computer components. The width of data bus is a key factor in determining the overall computer performance . Typical data bus is 8,16,32 or 64 bits wide , If a computer has a word size of 32 but with a 16 bit data bus ,then the data bus has to fetch the word twice from the main memory.
- **3) Control Bus :-** Control buses is a bi-directional pathway that carries command , timing and specific status information among components. The following are the some of control information a control bus may carry :-
- Write to memory
- Read from memory
- Write to I/O port
- Read from I/O port
- Request for data bus
- Grant for data bus
- Sync clock
- Reset all components.

GL BAJAJ Institute of Management
FIND YOUR SPARK
Affiliated to CCS University, Meerut.

Computer Architecture & Assembly Language

BCA-303

What is Interrupt ?

An interrupt is a signal from a device attached to a computer or from a program within the computer that requires the operating system to stop and figure out what to do next.

Interrupt systems are nothing but while the CPU can process the programs if the CPU needs any IO operation. Then, it is sent to the queue and it does the CPU process. Later on Input/output (I/O) operation is ready.

Types of interrupts

There are two types of interrupts which are as follows −

Hardware interrupts

The interrupt signal generated from external devices and i/o devices are made interrupt to CPU when the instructions are ready.

For example − In a keyboard if we press a key to do some action this pressing of the keyboard generates a signal that is given to the processor to do action, such interrupts are called hardware interrupts.

Hardware interrupts are classified into **two types** which are as follows −

Maskable Interrupt − The hardware interrupts that can be delayed when a highest priority interrupt has occurred to the processor.

Non Maskable Interrupt − The hardware that cannot be delayed and immediately be serviced by the processor.

Software interrupts

The interrupt signal generated from internal devices and software programs need to access any system call then software interrupts are present.

Software interrupt is divided into two types. They are as follows −

Normal Interrupts − The interrupts that are caused by the software instructions are called software instructions.

Exception − Exception is nothing but an unplanned interruption while executing a program. For example − while executing a program if we got a value that is divided by zero is called an exception.

GL BAJAJ
Institute of Management
FIND YOUR SPARK
Affiliated to CCS University, Meerut.

Computer Architecture & Assembly Language

BCA-303