RGPVNOTES.IN

Program : **B.Tech**

Subject Name: **Computer Architecture**

Subject Code: **IT-402**

Semester: **4th**

UNIT-1

Computer architecture refers to those parameters of a computer system that are visible to a programmer or those parameters that have a direct impact on the logical execution of a program. Examples of architectural attributes include the instruction set, the number of bits used to represent different data types, I/O mechanisms, and techniques for addressing memory.

Computer Architecture refers to those attributes of a system visible to a programmer or those attributes that have a direct impact on the logical execution of a program.

Examples of architectural attributes include:

a) Instruction set designing
b) Instruction format
c) No of bits used to represent various types of data
d) Different addressing mechanism to access data

Computer organization refers to the operational units and their interconnections that realize the architectural specifications. Examples of organizational attributes include those hardware details transparent to the programmer, such as control signals, interfaces between the computer and peripherals, and the memory technology used.

Ex: Two different models from a same vendor like Intel are brought to analyze. Both the models (laptop and desktop) have same processor like core 2 duo. That means both models understand the same instruction set as we know each processor understands a fixed no of instructions. Henceforth their architecture is same. Due to the placement of various hardware components, one model (laptop) is slim and other is bulky. Hence their organization is different.

Computer Generations

First Generation (1940-1956) Vacuum Tubes

The first computers used vacuum tubes for circuitry and magnetic drums for memory, and were often enormous, taking up entire rooms. They were very expensive to operate and in addition to using a great deal of electricity, the first computers generated a lot of heat, which was often the cause of malfunctions. They relied on machine language, the lowest-level programming language understood by computers, to perform operations.They could only solve one problem at a time, and it could take days or weeks to set-up a new problem. Input was based on punched cards and paper tape, and output was displayed on printouts. The UNIVAC and ENIAC computers are examples of first-generation computing devices.

Advantages

- Vacuum tubes were the only electronic component available during those days.
- Vacuum tube technology made possible to make electronic digital computers.
- These computers could calculate data in millisecond.

Disadvantages

- The computers were very large in size.
- They consumed a large amount of energy.
- They were heated very soon due to thousands of vacuum tubes.
- They were not very reliable.
- Air conditioning was required.
- Constant maintenance was required.
- Non-portable.
- Costly commercial production.

- Very slow speed.
- Limited programming capabilities.
- Used machine language only.
- Used magnetic drums which provide very less data storage.

Second Generation (1956-1963) Transistors

The period of second generation was from 1956-1963. In this generation, transistors were used that were cheaper, consumed less power, were more compact in size, more reliable and faster than the first-generation machines made of vacuum tubes. In this generation, magnetic cores were used as the primary memory and magnetic tape and magnetic disks as secondary storage devices. In this generation, assembly language and high-level programming languages like FORTRAN, COBOL was used. The computers used batch processing and multiprogramming operating system.

Advantages

- Smaller in size as compared to the first-generation computers.
- The 2nd generations Computers were more reliable.
- Used less energy and were not heated.
- Wider commercial use.
- Better portability as compared to the first-generation computers.
- Better speed and could calculate data in microseconds.
- Used faster peripherals like tape drives, magnetic disks, printer etc.
- Used Assembly language instead of Machine language.
- Accuracy was improved.

Disadvantages

- Cooling system was required.
- Constant maintenance was required.
- Commercial production was difficult.
- Only used for specific purposes.
- Costly and not versatile.
- Punch cards were used for input.

Third Generation (1964-1971) Integrated Circuits

The period of third generation was from 1965-1971. The computers of third generation used Integrated Circuits (ICs) in place of transistors. A single IC has many transistors, resistors, and capacitors along with the associated circuitry. The IC was invented by Jack Kilby. This development made computers smaller in size, reliable, and efficient. In this generation remote processing, time-sharing, multiprogramming operating system were used. High-level languages (FORTRAN-II TO IV, COBOL, PASCAL PL/1, BASIC, ALGOL-68 etc.) were used during this generation.

Advantages

- Smaller in size as compared to previous generations.
- Used less energy.
- Produced less heat as compared to the previous two generations of computers.
- Better speed and could calculate data in nanoseconds.
- Used fan for heat discharge to prevent damage.
- Totally general purpose.
- Could be used for high-level languages.
- Good storage.
- Less expensive.

- Better accuracy.
- Commercial production increased.
- Used mouse and keyboard for input.

Disadvantages

- Air conditioning was required.
- Highly sophisticated technology required for the manufacturing of IC chips.

Fourth Generation (1971-Present) Microprocessors

Fourth generation computers became more powerful, compact, reliable, and affordable which gave rise to Personal Computer (PC) revolution. In this generation, time sharing, real time networks, distributed operating system were used. All the high-level languages like C, C++, DBASE etc., were used in this generation.

Advantages

- More powerful and reliable than previous generations.
- Small in size.
- Fast processing power with less power consumption.
- Fan for heat discharging and thus to keep cold.
- No air conditioning required.
- Totally general purpose.
- Commercial production.
- Cheapest among all generations.
- All types of High level languages can be used in this type of computers.

Disadvantages

- The latest technology is required for manufacturing of Microprocessors.

Fifth Generation (Present and Beyond) Artificial Intelligence

The period of fifth generation is 1980-till date. In the fifth generation, VLSI technology became ULSI (Ultra Large-Scale Integration) technology, resulting in the production of microprocessor chips having ten million electronic components.

This generation is based on parallel processing hardware and AI (Artificial Intelligence) software. AI is an emerging branch, which interprets the means and method of making computers think like human beings. All the high-level languages like C and C++, Java, .Net etc. are used in this generation.

AI includes –

- Robotics
- Neural Networks
- Game Playing
- Development of expert systems to make decisions in real-life situations
- *Natural language understanding and generation*

Von Neumann Model

It was developed in 1945 by John Von Neumann. Von Neumann model consist of a CPU, memory and I/O devices. The program is stored in the memory. The CPU fetches an instruction from the memory at a time and executes it.

The Von Neumann architecture is a design model for a stored-program digital computer that uses a processing unit and a single separate storage structure to hold both instructions and data.

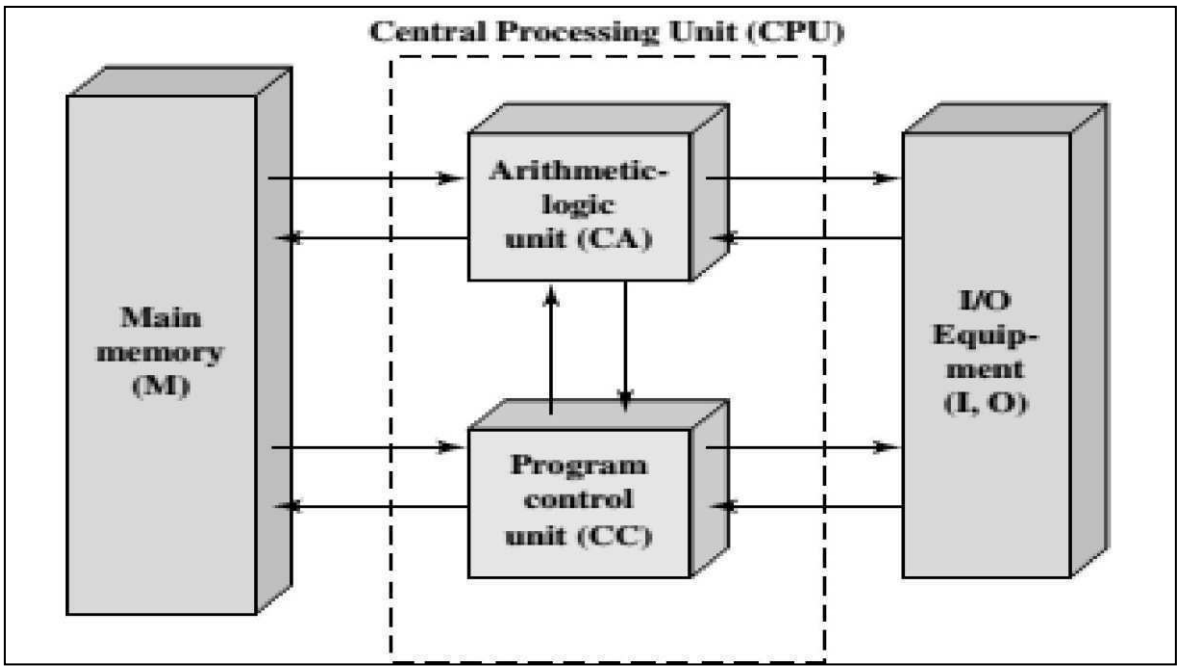

Fig 1.1 Von Neumann Model

A. Central Processor Unit [CPU]:

Central processor unit consists of two basic blocks:

The program control unit has a set of registers and control circuit to generate control signals. The execution unit or data processing unit contains a set of registers for storing data and an Arithmetic and Logic Unit (ALU) for execution of arithmetic and logical operations. In addition, CPU may have some additional registers for temporary storage of data.

B. Input Unit:

With the help of input unit data from outside can be supplied to the computer. Program or data is read into main storage from input device or secondary storage under the control of CPU input instruction. Example of input devices: Keyboard, Mouse, Hard disk, Floppy disk, CD-ROM drive etc.

C. Output Unit:

With the help of output unit computer results can be provided to the user or it can be stored in storage device permanently for future use. Output data from main storage go to output device under the control of CPU output instructions.

Example of output devices: Printer, Monitor, Plotter, Hard Disk, Floppy Disk etc.

D. Memory Unit:

Memory unit is used to store the data and program. CPU can work with the information stored in memory unit. This memory unit is termed as primary memory or main memory module. These are basically semiconductor memories.

There are two types of semiconductor memories -

- Volatile Memory: RAM (Random Access Memory).
- Non-Volatile Memory: ROM (Read only Memory), PROM (Programmable ROM) EPROM (Erasable PROM), EEPROM (Electrically Erasable PROM).

Secondary Memory:

There is another kind of storage device, apart from primary or main memory, which is known as secondary memory. Secondary memories are non-volatile memory and it is used for permanent storage of data and program.

Example of secondary memories:

Hard Disk, Floppy Disk, Magnetic Tape

CPU organization

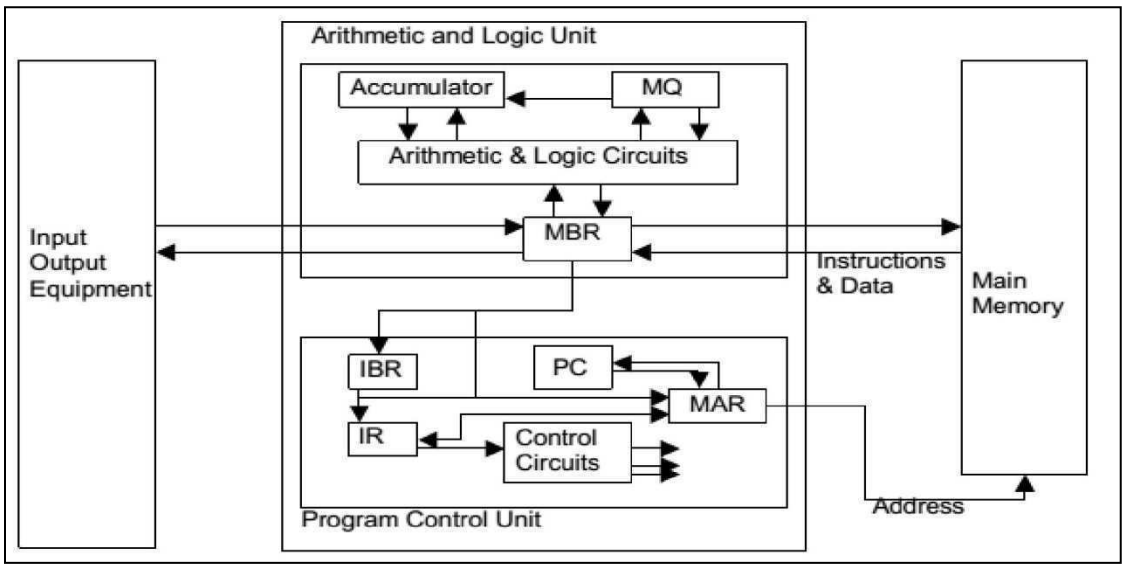

Fig 1.2 CPU Block Diagram

Depending on the internal organization computers can be categorized into one of the three CPU organization

1. Single Accumulator Organization
2. Stack Organization
3. General Register Organization

1. The single accumulator organizations are performed with the implied accumulator register. The instruction format in this type of organization uses one address field.

Example:

ADD X --> AC

2. The stack organized computer can use 1 or 0 address instruction and uses on y two instructions POP and PUSH

Example:

PUSH X --> TOS ss-M[X]

POP --> TOS

3. When there are more than one register, general register organization can be used and the instruction format may contain 2 or 3 address field.

Example:

ADD R1, R2 --> R1

ADD R1, R2, R3 --> R1

Register organization

The CPU is made up of three major parts: Register Set, ALU, and Control Unit as shown in figure below. The register set stores intermediate data used during the execution of the instructions. The arithmetic logic unit (ALU) performs the required microoperations for executing the instructions. The control unit supervises the transfer of information among the registers and instructs the ALU as to which operation to perform.

A bus organization for 7 CPU register is shown in a figure below. All registers are connected to two multiplexers (MUX) that select the registers for bus A and bus B. Registers selected by multiplexers are sent to ALU. Another selector (OPR) connected to ALU selects the operation for the ALU. Output produced by ALU is stored in some register and this destination register for storing the result is activated by the destination decoder (SELD).

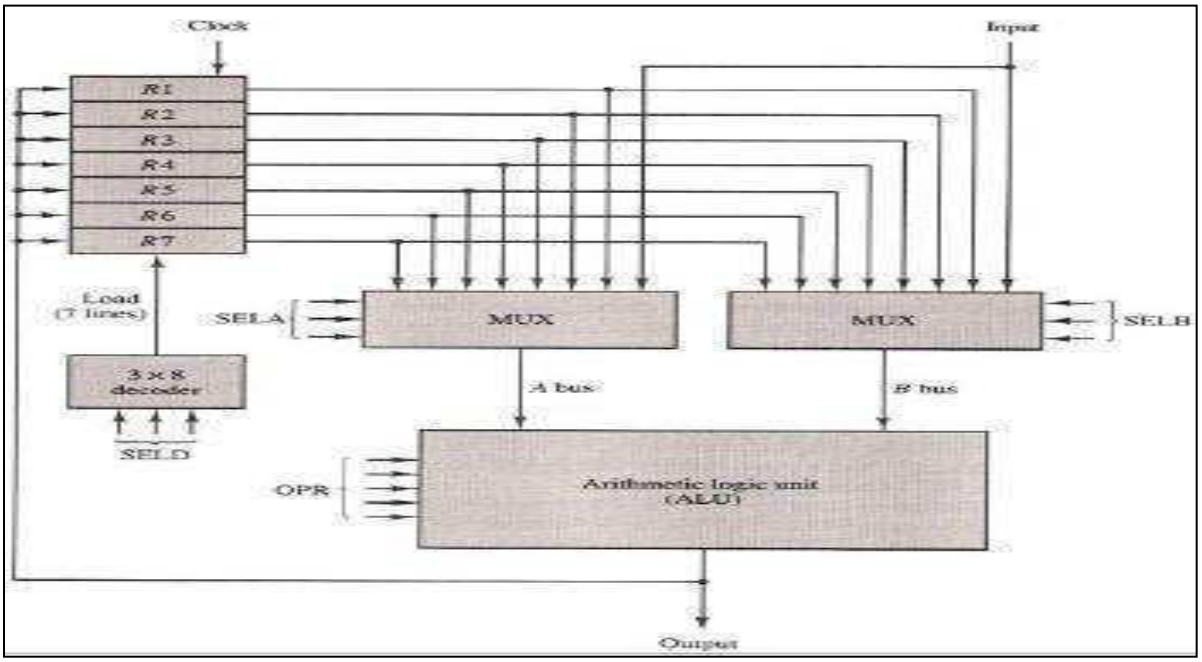

Fig 1.3 General register organization

Example: R1
– MUX selector (SELA): BUS A
– MUX selector (SELB): BUS B
– ALU operation selector (OPR): ALU to ADD
– Decoder destination selector (SELD): R1

Various CPU Registers

1. Memory Address Register (MAR):

This register holds the address of memory where CPU wants to read or write data. When CPU wants to store some data in the memory or reads the data from the memory, it places the address of the required memory location in the MAR.

2. Memory Buffer Register (MBR):

This register holds the contents of data or instruction read from, or written in memory. The contents of instruction placed in this register are transferred to the Instruction Register, while the contents of data are transferred to the accumulator or I/O register.

3. I/O address Register (I/O AR):

I/O Address register is used to specify the address of a particular I/O device.

4. I/O Buffer Register (I/O I3R):
I/O Buffer Register is used for exchanging data between the I/O module and the processor.

5. Program Counter (PC)
Program Counter register is also known as Instruction Pointer Register. This register is used to store the address of the next instruction to be fetched for execution. When the instruction is fetched, the value of IP is incremented. Thus, this register always points or holds the address of next instruction to be fetched.

6. Instruction Register (IR):
Once an instruction is fetched from main memory, it is stored in the Instruction Register. The control unit takes instruction from this register, decodes and executes it by sending signals to the appropriate component of computer to carry out the task.

7. Accumulator Register:
The accumulator register is located inside the ALU, it is used during arithmetic & logical operations of ALU. The control unit stores data values fetched from main memory in the accumulator for arithmetic or logical operation.

8. Stack Control Register:
A stack represents a set of memory blocks; the data is stored in and retrieved from these blocks in an order, i.e. First In and Last Out (FILO). The Stack Control Register is used to manage the stacks in memory. The size of this register is 2 or 4 bytes.

9. Flag Register:
The Flag register is used to indicate occurrence of a certain condition during an operation of the CPU. It is a special purpose register with size one byte or two bytes. Each bit of the flag register constitutes a flag (or alarm), such that the bit value indicates if a specified condition was encountered while executing an instruction.

Register Transfer:
Information transferred from one register to another is designated in symbolic form by means of replacement operator.
R2 ← R1 (It denotes the transfer of the data from register R1 into R2)

- Normally we want the transfer to occur only in predetermined control condition. This can be shown by following if-then statement: if (P=1) then (R2 ← R1)
- Here P is a control signal generated in the control section.

Control Function
A control function is a Boolean variable that is equal to 1 or 0. The control function is shown as:
P: R2 ← R1
The control condition is terminated with a colon. It shows that transfer operation can be executed only if P=1.

Basic symbols for register transfer language

Symbol	Description	Examples
Uppercase letters	Denotes a register	A, R1, MDR
Subscript	An individual cell	A5,B9
Parenthesis	A portion of register	PC(H), MDR(ADR)

Arrow	A transfer	R1 ss R2
Colon	Terminates a control function	T1:
Comma	Multiple operation	T:R1ss R2, R3 ss R4
Square Brackets	Address for memory	MDR ss M[MAR]

Bus & Memory Transfers:

- Memory Read: The read operation for the transfer of a memory unit M from an address register MAR to another data register DR can be illustrated as:
    - Read: DR ←M[MAR]
- Memory Write : The write operation transfer the contents of a data register to a memory word M selected by the address. Assume that the input data are in register R1 and the address in the MAR. The write operation can be stated symbolic as follows:
    - Write: M[MAR] ← R1
- This cause a transfer on information from R1 into the memory word M selected by the address in AR

BUS transfer

Using Multiplexers

Rather than connecting wires between all registers, a common bus is used A bus structure consists of a set of common lines, one for each bit of a register Control signals determine which register is selected by the bus during each transfer Multiplexers can be used to construct a common bus Multiplexers select the source register whose binary information is then placed on the bus the select lines are connected to the selection inputs of the multiplexers and choose the bits of one register

In general, a bus system will multiplex k registers of n bits each to produce an n-line common bus

- This requires n multiplexers – one for each bit
- The size of each multiplexer must be k x 1
- The number of select lines required is log k
- To transfer information from the bus to a register, the bus lines are connected to the inputs of all destination registers and the corresponding load control line must be activated rather than listing each step as

BUS ← C, R1 ← BUS,
R1 ← C, since the bus is implied

Three-state gates

Instead of using multiplexers, three-state gates can be used to construct the bus system
A three-state gate is a digital circuit that exhibits three states

- Two of the states are signals equivalent to logic 1 and 0
- The third state is a high-impedance state – this behaves like an open circuit, which means the output is disconnected and does not have a logic significance

The three-state buffer gate has a normal input and a control input which determines the output state With control 1, the output equals the normal input With control 0, the gate goes to a high-impedance state This enables a large number of three-state gate outputs to be connected with wires to form a common bus line

without endangering loading effects Decoders are used to ensure that no more than one control input is active at any given time This circuit can replace the multiplexer in figure 1.4.
To construct a common bus for four registers of n bits each using three-state buffers, we need n circuits with four buffers in each Only one decoder is necessary to select between the four registers.

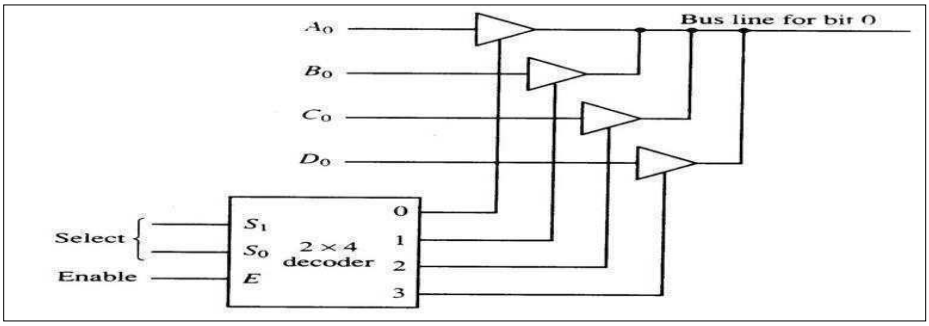

Fig 1.4 Three state bus buffer

Memory Transfer

We designate a memory word by the letter M. It is necessary to specify the address of M when writing memory transfer Operations Designate the address register by AR and the data register by DR.
The read operation can be stated as:

Read: DR ← M [AR]

The write operation can be stated as:

Write: M [AR] ← R1

The address register (AR) is used to select a memory address, and the data register (DR) is used to send and receive data. Both these registers are connected to the internal bus. DR is a bridge between the internal BUS and the memory data BUS.
Memory can also be connected directly to the internal BUS in theory.

M[AR] ← DR
DR ← M[AR]

Hence, accessing memory outside the CPU requires at least two clock cycles. First, we load AR with the desired memory address, and then transfer to or from DR. In most typical computer systems, memory transfers take many clock cycles, known as wait states.

Arithmetic, Logic and Shift micro-operations
Micro-operations perform basic operations on data stored in one or moreregisters, including transferring data between registers or between registers and external buses of the central processing unit (CPU), and performing arithmetic or logical operations on registers.

Types of Micro-operations

- **Register transfer micro-operations**: These types of micro operations are used to transfer from one register to binary information.
- **Arithmetic micro-operations**: These micro-operations are used to perform on numeric data stored in the registers some arithmetic operations.
- **Logic micro-operations:** These micro operations are used to perform bit style operations / manipulations on non-numeric data.

- **Shift micro operations:** As their name suggests they are used to perform shift operations in data store in registers.

Arithmetic Micro-Operations: Some of the basic micro-operations are addition, subtraction, increment and decrement.

Add Micro-Operation
It is defined by the following statement:
R3 → R1 + R2
The above statement instructs the data or contents of register R1 to be added to data or content of register R2 and the sum should be transferred to register R3.

Subtract Micro-Operation
Let us again take an example:
R3 → R1 + R2' + 1
In subtract micro-operation, instead of using minus operator we take 1's compliment and add 1 to the register which gets subtracted, i.e R1 - R2 is equivalent to R3 → R1 + R2' + 1

Increment/Decrement Micro-Operation
Increment and decrement micro-operations are generally performed by adding and subtracting 1 to and from the register respectively.
R1 → R1 + 1
R1 → R1 – 1

Symbolic Designation	Description
R3 ← R1 + R2	Contents of R1+R2 transferred to R3.
R3 ← R1 - R2	Contents of R1-R2 transferred to R3.
R2 ← (R2)'	Compliment the contents of R2.
R2 ← (R2)' + 1	2's compliment the contents of R2.
R3 ← R1 + (R2)' + 1	R1 + the 2's compliment of R2 (subtraction).
R1 ← R1 + 1	Increment the contents of R1 by 1.
R1 ← R1 - 1	Decrement the contents of R1 by 1.

Logic Micro-Operations
These are binary micro-operations performed on the bits stored in the registers. These operations consider each bit separately and treat them as binary variables.

Let us consider the X-OR micro-operation with the contents of two registers R1 and R2.
P: R1 ← R1 X-OR R2
In the above statement we have also included a Control Function.
Assume that each register has 3 bits. Let the content of R1 be 010 and R2 be 100. The X-OR micro-operation will be:

010 → R1

100 → R2

110 → R1 after P=1

Shift Micro-Operations

These are used for serial transfer of data. That means we can shift the contents of the register to the left or right. In the shift left operation the serial input transfers a bit to the right most position and in shift right operation the serial input transfers a bit to the left most position.

There are three types of shifts as follows:

a) Logical Shift

It transfers 0 through the serial input. The symbol "shl" is used for logical shift left and "shr" is used for logical shift right.

R1 ← she R1

R1 ← she R1

The register symbol must be same on both sides of arrows.

b) Circular Shift

This circulates or rotates the bits of register around the two ends without any loss of data or contents. In this, the serial output of the shift register is connected to its serial input. "cil" and "cir" is used for circular shift left and right respectively.

c) Arithmetic Shift

This shifts a signed binary number to left or right. An arithmetic shift left multiplies a signed binary number by 2 and shift left divides the number by 2. Arithmetic shift micro-operation leaves the sign bit unchanged because the signed number remains same when it is multiplied or divided by 2.

RGPVNOTES.IN

We hope you find these notes useful.

You can get previous year question papers at https://qp.rgpvnotes.in .

If you have any queries or you want to submit your study notes please write us at rgpvnotes.in@gmail.com

LIKE & FOLLOW US ON FACEBOOK
facebook.com/rgpvnotes.in