EXPERIMENT NO. 01

DATE OF PERFORMANCE: **GRADE:**

DATE OF ASSESSMENT: **SIGNATURE OF LECTURER/ TTA:**

AIM: Introduction to Computer Organization and Architecture.

Computer Organization:

Computer Architecture is a view of the whole design with the important characteristics visible to programmer.

Von-Neumann architecture:

Historically there have been 2 types of Computers:

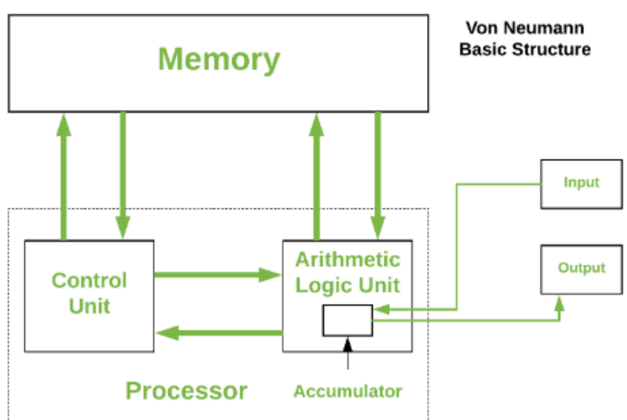
1. **Fixed Program Computers – Their function is very specific and they couldn't be programmed, e.g. Calculators.**

2. **Stored Program Computers – These can be programmed to carry out many different tasks, applications are stored on them, hence the name.]**

The modern computers are based on a stored-program concept introduced by John Von Neumann. In this stored-program concept, programs and data are stored in a separate storage unit called memories and are treated the same.

This novel idea meant that a computer built with this architecture would be much easier to reprogram.

The basic structure is like,



It is also known as IAS computer and is having three basic units:

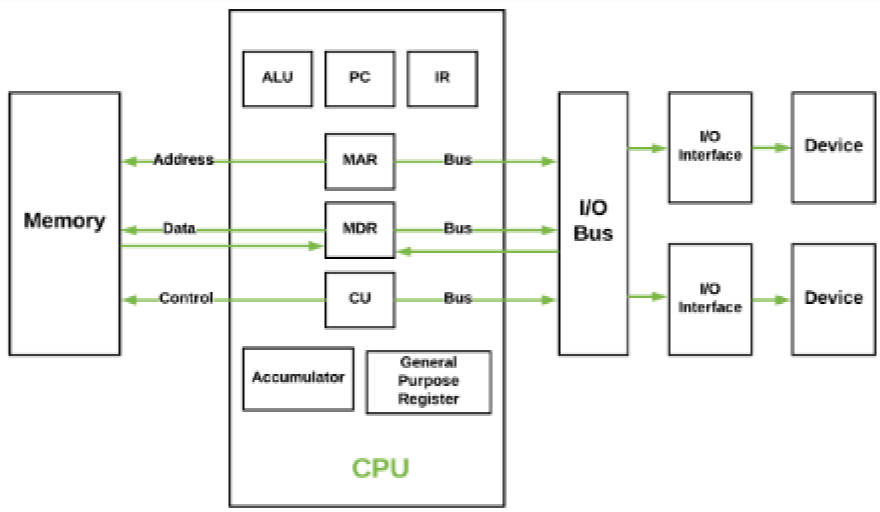
1. The Central Processing Unit (CPU)
2. The Main Memory Unit
3. The Input/Output Device

Control Unit:

A control unit (CU) handles all processor control signals. It directs all input and output flow, fetches code for instructions and controlling how data moves around the system.

Arithmetic and Logic Unit (ALU) :

The arithmetic logic unit is that part of the CPU that handles all the calculations the CPU may need, e.g. Addition, Subtraction, Comparisons. It performs Logical Operations, Bit Shifting Operations, and Arithmetic Operation.



Main Memory Unit (Registers):

1. **Accumulator:** Stores the results of calculations made by ALU.

2. **Program Counter (PC):** Keeps track of the memory location of the next instructions to be dealt with. The PC then passes this next address to Memory Address Register (MAR).

3. **Memory Address Register (MAR):** It stores the memory locations of instructions that need to be fetched from memory or stored into memory.

4. **Memory Data Register (MDR):** It stores instructions fetched from memory or any data that is to be transferred to, and stored in, memory.

5. **Current Instruction Register (CIR):** It stores the most recently fetched instructions while it is waiting to be coded and executed.

6. **Instruction Buffer Register (IBR):** The instruction that is not to be executed immediately is placed in the instruction buffer register IBR.

- **<u>Input/Output Devices:</u>** **Program or data is read into main memory from the *input device* or secondary storage under the control of CPU input instruction.**

  Output devices **are used to output the information from a computer. If some results are evaluated by computer and it is stored in the computer, then with the help of output devices, we can present it to the user.**

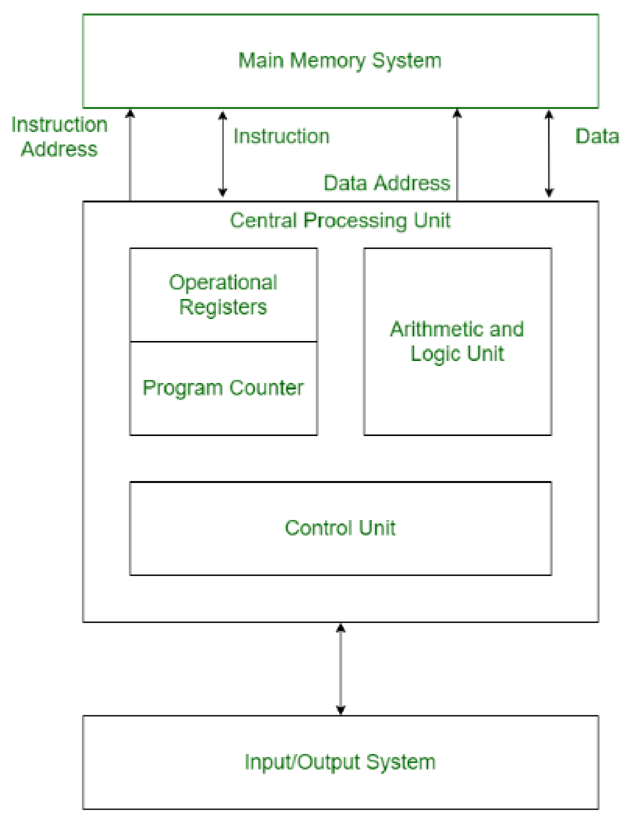
- **<u>Buses :</u> Data is transmitted from one part of a computer to another, connecting all major internal components to the CPU and memory, by the means of Buses. Types:**

1. **<u>Data Bus:</u> It carries data among the memory unit, the I/O devices, and the processor.**
2. **<u>Address Bus:</u> It carries the address of data (not the actual data) between memory and processor.**
3. **<u>Control Bus:</u> It carries control commands from the CPU (and status signals from other devices) in order to control and coordinate all the activities within the computer.**

<u>Harvard Architecture:</u>

Harvard Architecture is the computer architecture that contains separate storage and separate buses (signal path) for instruction and data. It was basically developed to overcome the bottleneck of Von Neumann Architecture. The main advantage of having separate buses for instruction and data is that the CPU can access instructions and read/write data at the same time.

Structure of Harvard architecture:



Buses:

Buses are used as signal pathways. In Harvard architecture, there are separate buses for both instruction and data.

Types of Buses:

- **Data Bus:** **It carries data among the main memory system, processor, and I/O devices.**
- **Data Address Bus:** **It carries the address of data from the processor to the main memory system.**
- **Instruction Bus:** **It carries instructions among the main memory system, processor, and I/O devices.**
- **Instruction Address Bus:** **It carries the address of instructions from the processor to the main memory system.**

Operational Registers:

There are different types of registers involved in it which are used for storing addresses of different types of instructions.

***For example*, the Memory Address Register and Memory Data Register are operational registers.**

Program Counter:

It has the location of the next instruction to be executed. The program counter then passes this next address to the memory address register.

Arithmetic and Logic Unit:

The arithmetic logic unit is that part of the CPU that operates all the calculations needed. It performs addition, subtraction, comparison, logical Operations, bit Shifting Operations, and various arithmetic operations.

Control Unit:

The Control Unit is the part of the CPU that operates all processor control signals. It controls the input and output devices and also controls the movement of instructions and data within the system.

Input/output System:

Input devices are used to read data into main memory with the help of CPU input instruction. The information from a computer as output is given through Output devices. The computer gives the results of computation with the help of output devices.

Advantage of Harvard Architecture:

Harvard architecture has two separate buses for instruction and data. Hence, the CPU can access instructions and read/write data at the same time. This is the major advantage of Harvard architecture.

In practice, Modified Harvard Architecture is used where we have two separate caches (data and instruction). This is common and used in X86 and ARM processors.

Software:

Software is a set of instructions, data or programs used to operate computers and execute specific tasks.

It enables a computer to perform specific task as opposed to the physical components of a system.

There are 3 types of software:

1) **System software**
2) **Application software**
3) **Programming software**

System software:

These software programs are designed to run a computer's application programs and hardware. System software coordinates the activities and functions of the hardware and software. In addition, it controls the operations of the computer hardware and provides an environment or platform for all the other types of software to work in. The OS is the best example of system software; it manages all the other computer programs. Other examples of system software include the firmware, computer language translators and system utilities.

Application software:

The most common type of software, application software is a computer software package that performs a specific function for a user, or 0069n some cases, for another application. An application can be self-contained, or it can be a group of programs that run the application for the user. Examples of modern applications include office suites, graphics software, databases and database management programs, web browsers, word processors, software development tools, image editors and communication platforms.

Programming software:

Computer programmers use programming software to write code. Programming software and programming tools enable developers to develop, write, test and debug other software programs. Examples of programming software include assemblers, compilers, debuggers and interpreters.

Hardware:

The physical part of a computer including the digital circuitry as distinguish from the computer software that executes within the hardware.

Operation:

Computer Software has to be loaded into the computer storage as a hardware memory.

Register transfer language:

A digital computer system exhibits an interconnection of digital modules such as registers, decoders, arithmetic elements, and Control logic.

These digital modules are interconnected with some common data and control paths to form a complete digital system.

Moreover, digital modules are best defined by the registers and the operations that are performed on the data stored in them.

The operations performed on the data stored in registers are called Micro-operations.

The internal hardware organization of a digital system is best defined by specifying:

- **The set of registers and the flow of data between them.**
- **The sequence of micro-operations performed on the data which are stored in the registers.**
- **The control paths that initiates the sequence of micro-operation**

The Register Transfer Language is the symbolic representation of notations used to specify the sequence of micro-operations.

In a computer system, data transfer takes place between processor registers and memory and between processor registers and input-output systems.

Register Transfer:

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.**
- **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.**

- **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:

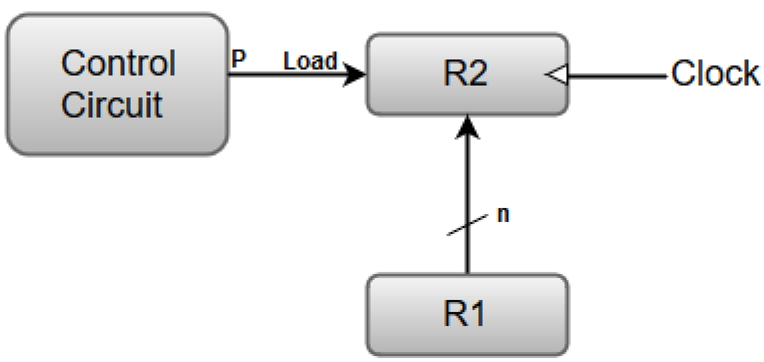
  If (P=1) then ($R2 \leftarrow R1$); Here P is a control signal generated in the control section.

  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).

1. P: $R2 \leftarrow R1$

   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;



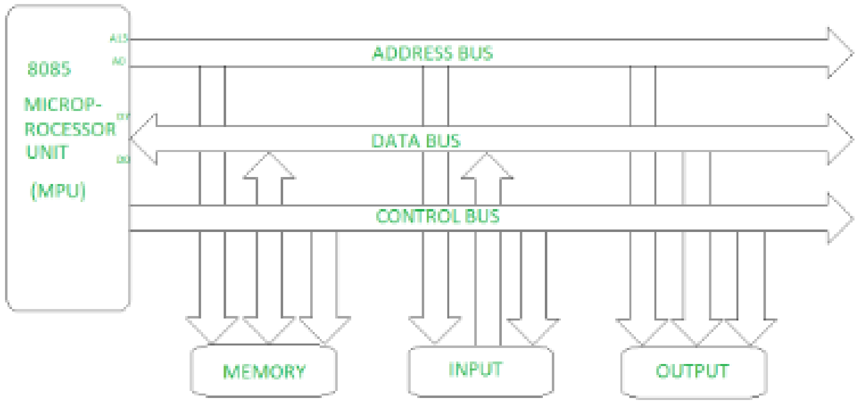
Here, the letter 'n' indicates the number of bits for the register. The 'n' outputs of the register R1 are connected to the 'n' inputs of register R2.

A load input is activated by the control variable 'P' which is transferred to the register R2.

Buses:

Bus is a group of conducting wires which carries information; all the peripherals are connected to microprocessor through Bus.

Diagram to represent bus organization system of 8085 Microprocessor.



There are three types of buses.

1. **Address bus –**

   It is a group of conducting wires which carries address only. Address bus is unidirectional because data flow in one direction, from microprocessor to memory or from microprocessor to Input/output devices (That is, Out of Microprocessor).

   Length of Address Bus of 8085 microprocessor is 16 Bit (That is, Four Hexadecimal Digits), ranging from 0000 H to FFFF H, (H denotes Hexadecimal). The microprocessor 8085 can transfer maximum 16 bit address which means it can address 65, 536 different memory location.

   The Length of the address bus determines the amount of memory a system can address. Such as a system with a 32-bit address bus can address 2^32 memory locations.

   If each memory location holds one byte, the addressable memory space is 4 GB. However, the actual amount of memory that can be accessed is usually much less than this theoretical limit due to chipset and motherboard limitations.

2. **Data bus–**

   It is a group of conducting wires which carries Data only.Data bus is bidirectional because data flow in both directions, from microprocessor to memory or Input/Output devices and from memory or Input/Output devices to microprocessor.

   Length of Data Bus of 8085 microprocessor is 8 Bit (That is, two Hexadecimal Digits), ranging from 00 H to FF H. (H denotes Hexadecimal).

   When it is write operation, the processor will put the data (to be written) on the data bus, when it is read operation, the memory controller will get the data from specific memory block and put it into the data bus.

   The width of the data bus is directly related to the largest number that the bus can carry, such as an 8 bit bus can represent 2 to the power of 8 unique values, this equates to the number 0 to 255.A 16 bit bus can carry 0 to 65535.

3. **Control bus –**

   It is a group of conducting wires, which is used to generate timing and control signals to

control all the associated peripherals, microprocessor uses control bus to process data, that is what to do with selected memory location. Some control signals are:

- Memory read
- Memory write
- I/O read
- I/O Write
- Opcode fetch

If one line of control bus may be the read/write line.If the wire is low (no electricity flowing) then the memory is read, if the wire is high (electricity is flowing) then the memory is written.

Bus and memory transfer:

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.

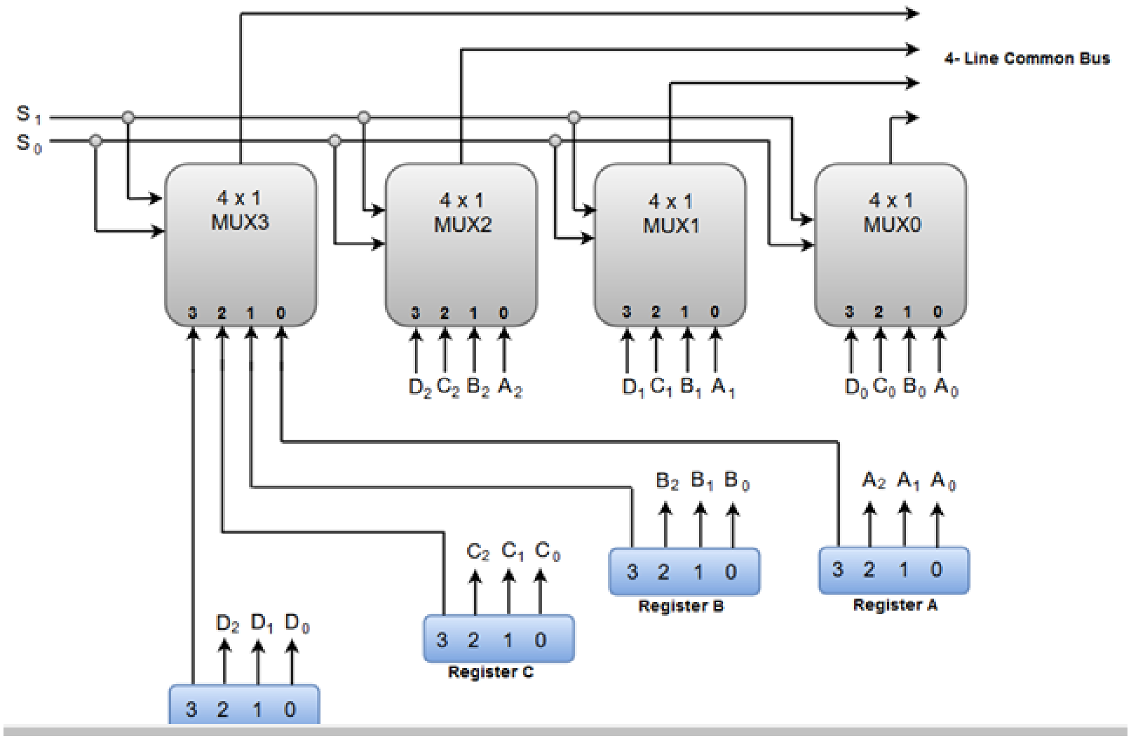
A bus structure, on the other hand, is more efficient for transferring information between registers in a multi-register configuration system.

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.

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.

Bus systems for 4 Registers:



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.

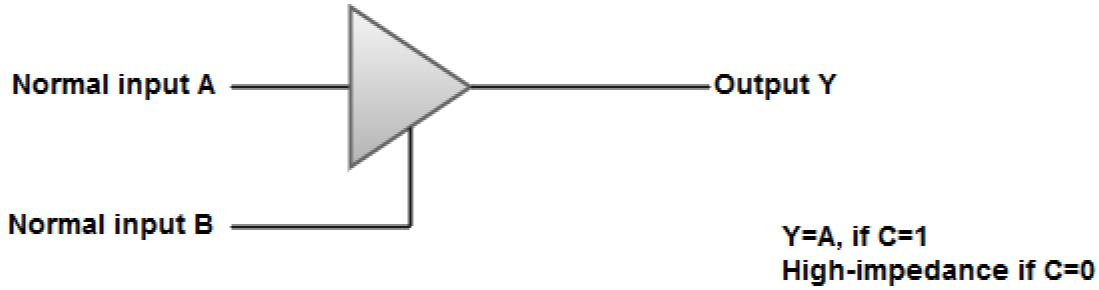
S1	S0	Register Selected
0	0	A
0	1	B
1	0	C
1	1	D

Tristate bus buffer or Three-state bus buffer:

A bus system can also be constructed using three-state gates instead of multiplexers.

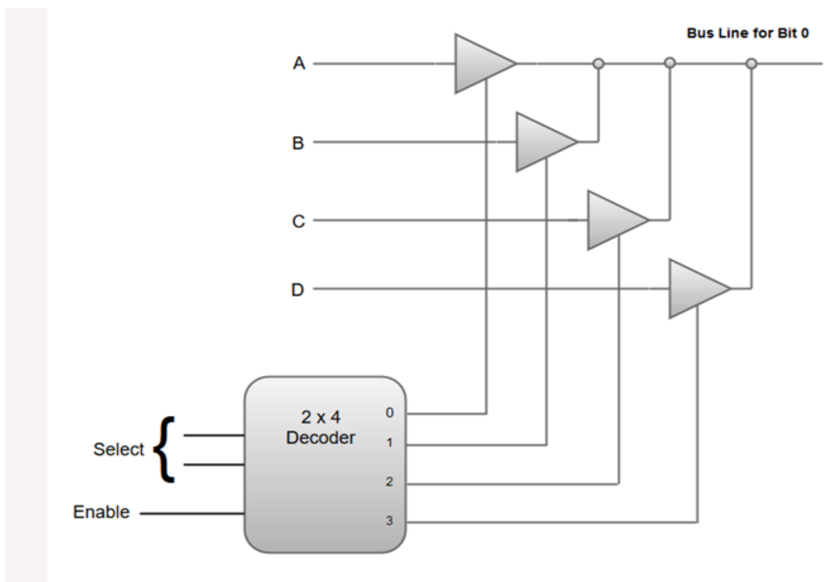
The three state gates can be considered as a digital circuit that has three gates, two of which are signals equivalent to logic 1 and 0 as in a conventional gate. However, the third gate exhibits a high-impedance state.

The most commonly used three state gates in case of the bus system is a buffer gate.



The following diagram demonstrates the construction of a bus system with three-state buffers.

Bus line with three state buffer:



- The outputs generated by the four buffers are connected to form a single bus line.
- Only one buffer can be in active state at a given point of time.
- The control inputs to the buffers determine which of the four normal inputs will communicate with the bus line.
- A 2 * 4 decoder ensures that no more than one control input is active at any given point of time.

Memory Transfer:

Most of the standard notations used for specifying operations on memory transfer are stated below.

- **The transfer of information from a memory unit to the user end is called a Read operation.**
- **The transfer of new information to be stored in the memory is called a Write operation.**
- **A memory word is designated by the letter M.**
- **We must specify the address of memory word while writing the memory transfer operations.**
- **The address register is designated by AR and the data register by DR.**
- **Thus, a read operation can be stated as:**

1. **Read: DR ← M [AR]**
   - **The Read statement causes a transfer of information into the data register (DR) from the memory word (M) selected by the address register (AR).**
   - **And the corresponding write operation can be stated as:**

1. **Write: M [AR] ← R1**
   - **The Write statement causes a transfer of information from register R1 into the memory word (M) selected by address register (AR).**