

Assignment - I

Date: _____

Page No. _____

- ① Explain the difference of Computer architecture and computer organization.

Computer Architecture is a functional description of requirement and design implementation for the various part of a computer. It deals with the functional behaviour of computer system. It comes before the computer organization while designing a computer.

Computer organization comes after the decision of Computer Architecture first. Computer organization is how operational attributes are linked together and contribute to realizing the architectural specification.

Computer Organization deals with a structural relationship.

Computer Architecture

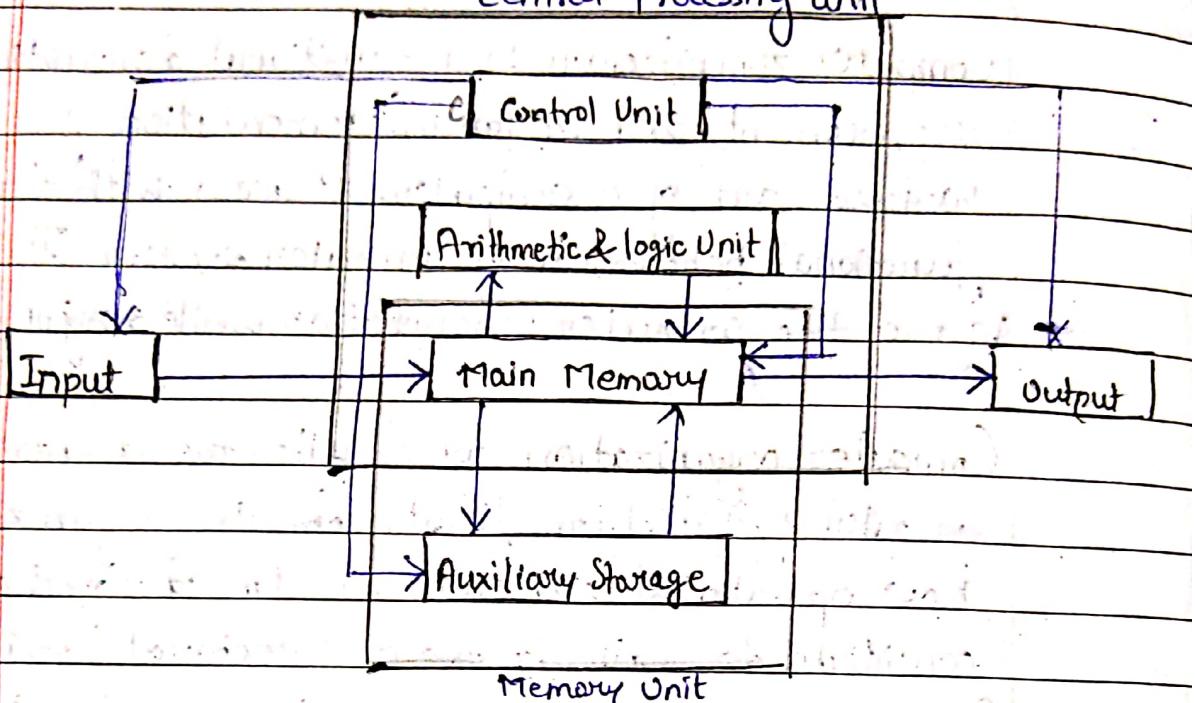
- Architecture describe what the computer does
- It deals with the functional behaviour of computer
- Architecture indicate its hardware
- Computer Architecture comprises logical functions such as instruction sets, registers, data types, and addressing modes

Computer Organization

- The organization describe how it does it.
- It deals with a structural relationship.
- Organization indicate its performance
- Computer organization consist of physical units like circuit design, peripherals and adders.

(2) Show the various functional units and interconnection of computer. Explain with diagram.

central Processing unit



(Block Diagram of Computer).

- i) **Input Unit**: The input Unit consists of input devices that are attached to the computer.
 - Computer input devices are keyboard, mouse, joysticks, scanner, etc.
- ii) **Central Processing Unit**: The CPU is called the brain of computer because it is the control centre of the computer.
 - It fetch first instruction from memory and then interprets them to as to know what is to be done.
- iii) **Arithmetic & logic Unit**: It performs mathematical calculation and takes logical decisions.

- Addition, subtraction, multiplication, and division like all calculations.
- logical decision involve comparison of two data items to see which one is larger or smaller or equal.

v) Control Unit : It coordinate and control the data flow in and out of CPU and also controls all the operations of ALU, memory register and input / output units.

- Sends control signals to I/O devices until the required operation is done properly by ALU and memory.

vi) Primary / Main memory : The memory unit that establish direct communication with the CPU is called main memory.

- It holds the data and instructions that the processor is currently working.

vii) Secondary Memory : The contents of the secondary memory first get transferred to the primary and then are accessed by the processor.

- The memory unit that provide backup storage are called Auxiliary Memory. e.g. magnetic disk, tapes etc.

viii) Output Unit : ~~Memory attached to the CPU is used for storage of data~~

vix) Output Unit :- The output unit consists of output devices that are attached with the computer to provide the processed result.
e.g. monitor, printer, plotter etc.

③

A) Write down the various types of Registers and Explain in details.

B) Explain the push and pop for the type of Stack organization.

A) Various types of Registers will be following.

i) MAR (Memory Address Register): This register holds the memory addresses of data and instruction. This register is used to access data and instructions from memory during the execution phase of an instruction.

ii) PC (Program Counter): This register commonly called instruction pointer (IP). It is a 16-bit special function register. It keeps track of the next memory address of the instruction that is to be executed once the execution of the current instruction is completed.

iii) Accumulator Register (AC): This register is used for storing the results those are produced by the system. When the CPU will generate some results after the processing then all the results stored into the AC register.

iv) MDR (Memory data Register): MDR is the register of a computer's control unit that contains the data to be stored in the computer storage (e.g. RAM).

or the data after a fetch from the computer storage. It acts like a buffer and holds anything that is copied from the memory ready for the processor to use it.

v) Index Register : A hardware element which holds a number that can be added to the address portion of a computer instruction to form an effective address. Also known as base register.

vi) MBR (Memory buffer Register). MBR holds the contents of data or instruction read from or written in memory. It means that this register is used to store data/instruction coming from the memory or going to the memory.

vii) Data Register : A register used in microcomputers to temporarily store data being transmitted to or from a peripheral device.

B) PUSH Operation : The push operation refers to inserting an element in the stack. There's only one position at which the new element can be inserted : at the top of the stack.

- If the stack is not full ($FULL=0$), a new item is inserted with a push operation. The push operation consists of the following sequence of micro operations.

- $SP \leftarrow SP + 1$ Increment stack pointer
- $M[SP] \leftarrow DR$ Write item on top of the stack
- IF $(SP=0)$ then $(FULL \leftarrow 1)$ Check if stack is full
- $EMTY \leftarrow 0$ Mark the stack not empty.

POP Operation : The pop operation is refer to the removal of an element. The operation `?op()` removes the top item from the stack, and return that item.

- A new item is deleted from the stack if the stack is not empty (if $EMTY = 0$). The pop operation consists of the following microprograms.

- $DR \leftarrow M[SP]$ Read item on the top of the stack
- $SP \leftarrow SP - 1$ Decrement stack pointer
- IF $(SP=0)$ then $(EMTY \leftarrow 1)$ Check if stack empty
- $FULL \leftarrow 0$ Mark the stack not full.

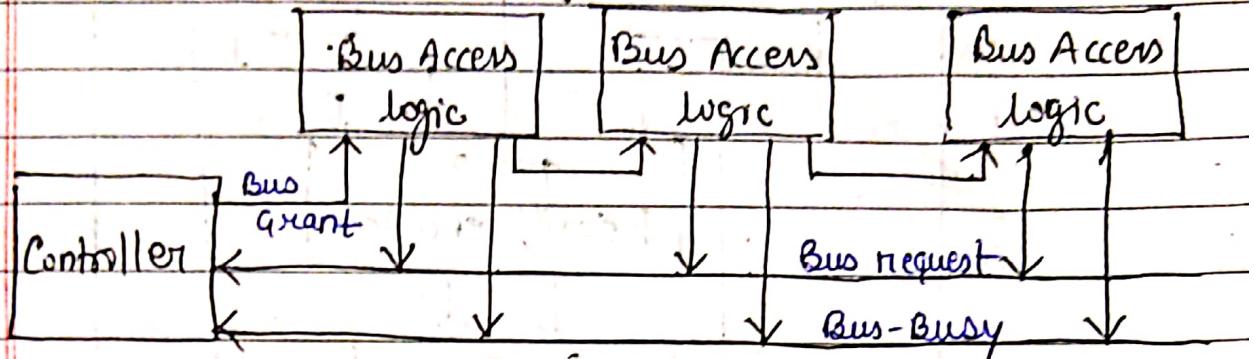
④ Elaborate Bus Arbitration (Static & Dynamic) with Suitable diagram.

- Static priority Arbitration algorithm.
 - a) Serial Arbitration
 - b) Parallel Arbitration.

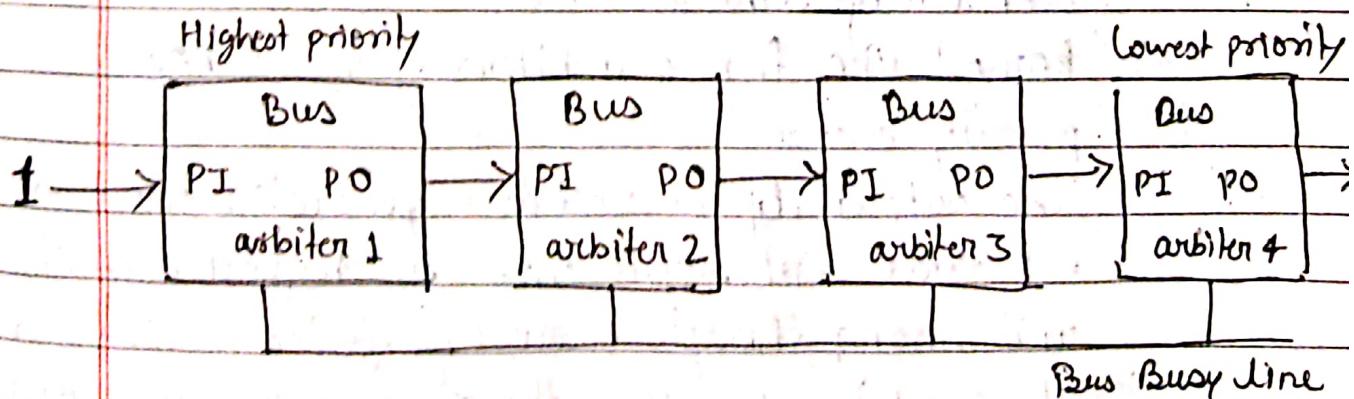
Serial Bus Arbitration is also known as daisy chain arbitration and in this method all the arbiter connect in serial.

→ first arbiter has highest priority and last arbiter has lowest priority.

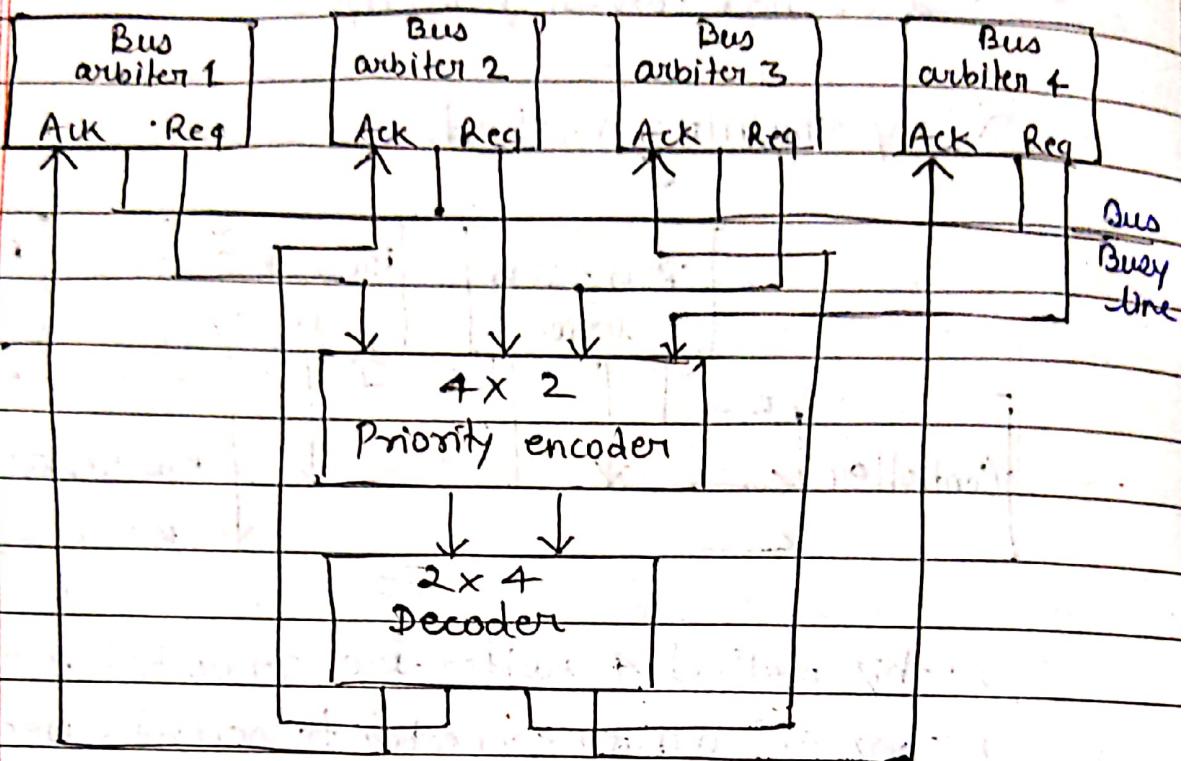
Daisy chaining method.



- In this method 4 arbiters are connected serially means 1st arbiter connected to second, second to third.
- Each arbiter to have a PI and PO means priority in and priority out.
- First arbiter PO connected with PI of second arbiter
- All the arbiter connected with common bus busy line
- 1st arbiter default PI value of 1.
- position of arbiter is related to priority. 1st arbiter has highest priority & last has lowest priority



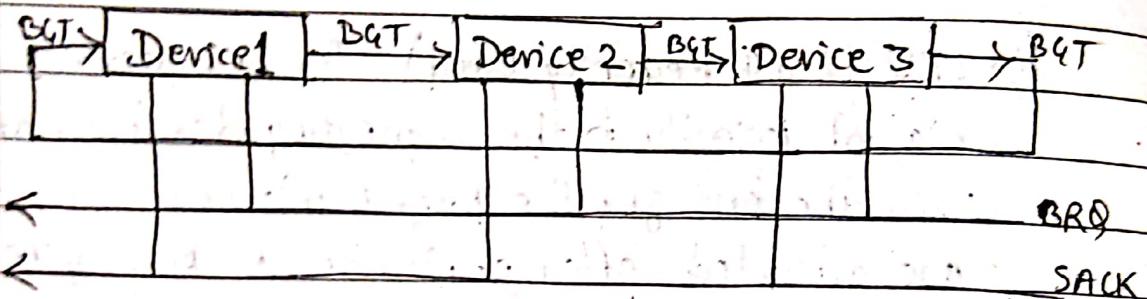
Parallel arbitration



- The parallel bus arbitration technique uses priority encoder & decoder
- Each bus arbiter in parallel arbitration a bus request output line (REQ) & bus acknowledgement line (ACK) input line.
- Each arbiter enables the request line when its processor is requesting access to the system bus
- The processor takes control of the bus if its acknowledge line is enable.
- Request line from 4 arbiters going to 4x2 priority encoder
- The output of the encoder generates the two bit code with represents the highest priority unit among those requesting for common bus.
- The two bit code from the encoder output drives a 2x4 decoder which enables the proper acknowledge line to grant bus access to highest priority.

- Dynamic priority arbitration algorithm.
 - a) Time slice - b) LRU c) FIFO d) polling
 - e) Rotating Daisy chain
- a) Time slice : The time slice algorithm allocates a fixed length time slice of bus that is offered sequentially to each processor, in round-robin fashion
- b) LRU (Least Recently Used) - The LRU algorithm gives the highest priority to the requesting device that has not used the bus for the longest interval. The priority are adjusted after a number of bus cycle according to the LRU algorithm.
- c) FIFO (First in First Out) : In FIFO the first come first serve scheme, request are served in the order received. The bus controller establish a queue arranged according to the time that the bus request arrive.
Each processor must wait for its turn to use the bus on a first in first out (FIFO) basis.
- d) Polling : In a bus system that uses polling, the bus grant signal is replaced by a set of lines called poll lines which are connected to all units. These lines are used by the bus controller to define an address for each device connected to the bus.
- The polling sequence is normally programmable and as a result the selection priority can be altered under program control.

e) Rotating Daisy chain : The rotating daisy chain procedure is a dynamic extension of the daisy chain algorithm. In this scheme there is no bus controller and the priority line is connected from the priority out of the last device back to the priority in of the first device in a closed loop.



(5) Define Various Addressing mode with one suitable example.

→ Implied mode :- In this mode the operands are specified implicitly in the definition of the instruction.

e.g. Complement accumulator

→ Immediate mode : In this mode the operands are specified ~~impliedly~~ in the instruction itself.

In immediate mode instruction has an operand field rather than address field.

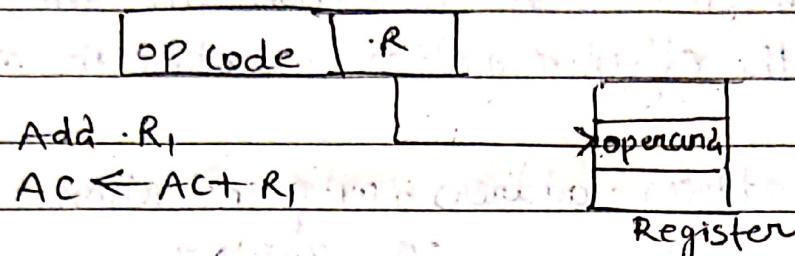
e.g. MV 106 Move 06 to accumulator

ADI 04 Add 05 to the content of
accumulator.

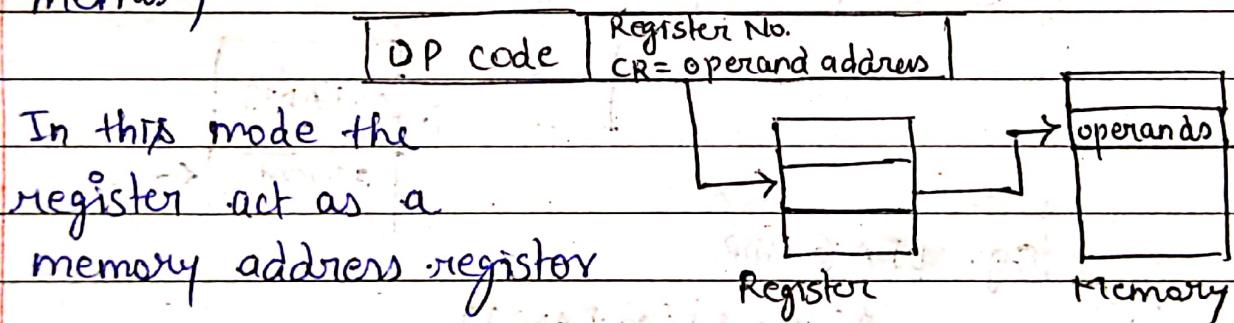
→ Register Addressing Mode: In this mode the operands are in register that resides with in the CPU.

The content of the register is the operand itself

→ EA (Effective address = R)



→ Register indirect addressing mode: - In this mode the instruction specifies a register in the CPU whose content give the address of the operand in the memory

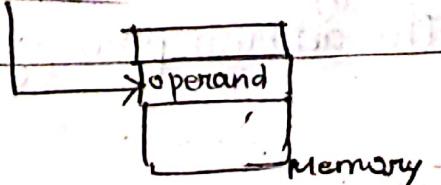
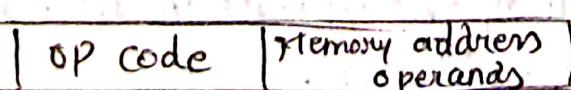


e.g. LDACR₁ (Load in accumulator)

AC ← M[R₁]

→ Direct addressing mode: - In this mode the effective address is equal to the address part of the instruction.

The operand resides in memory & its address is given directly by the address field of the instruction.



e.g. LDA ADR.

$$AC \leftarrow M[ADR]$$

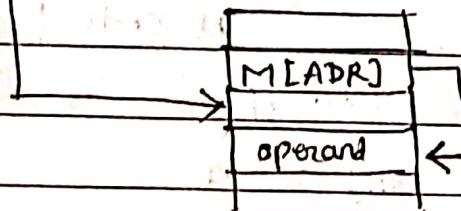
- Indirect addressing mode :- In this mode the address field of the instruction gives the address where the effective address is stored in memory.

Effective address = address part of instruction + content of CPU register.

OP. Code | (memory-address = operand address)

Instruction

opcode | .ADR)



e.g. LDA @ADR

$$AC \leftarrow M[M[ADR]]$$

- Auto increment / Auto decrement addressing mode :- It is like a register indirect mode except that the register is incremented (+) or decremented (-) after or before its value is used to access memory.

- Relation addressing mode :- In this mode the content of PC (program counter) is added to the address part of the instruction in order to obtain the effective address.

The address part of the instruction is usually a

Align number (2's complement) which can be + or -.

- ★ Displacement addressing :- In this mode combine the capabilities of direct addressing & register in direct addressing.

$$EA = A + R.$$

- ★ Base register addressing :- In this mode the content of base register is added to the address part of the instruction to obtain the effective address

$$EA = A + BR.$$

A - Content of the address field

BR - effective.

- ★ Index addressing mode : In this mode the content of an index register is added to the address part of the instruction to obtain the effective ~~address~~ register.

The index register is a special CPU register that contains an index value.

The distance between the beginning address and the address of the operand is the index value. Started in index register.

