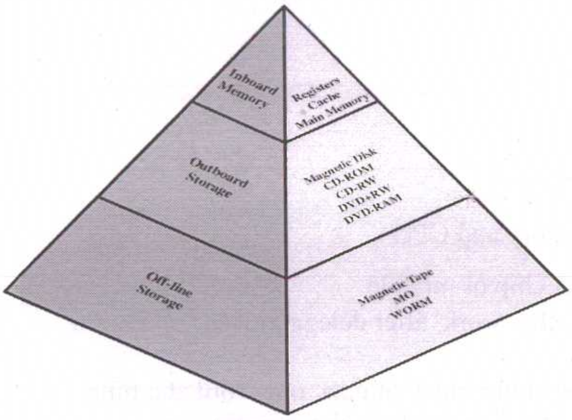
QID 87751

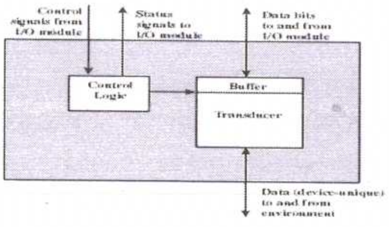
TED(15)-3131 Revision -2015

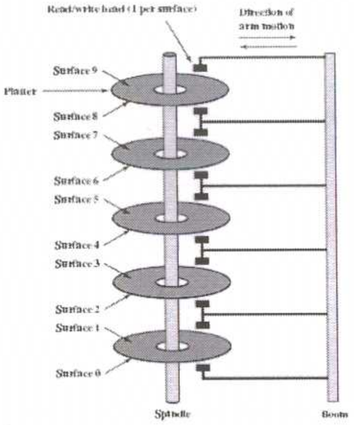
DIPLOMA EXAMINATION IN ENGINEERING/ TECHNOLOGY

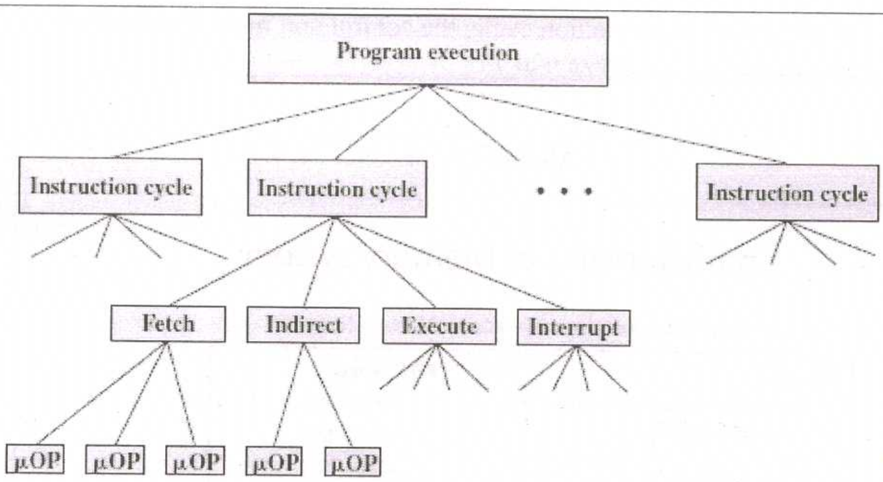
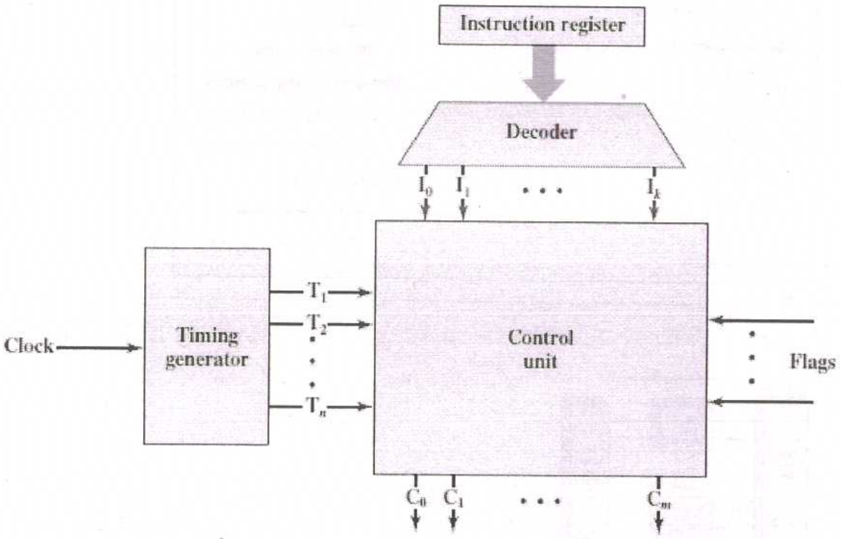
COMPUTER ARCHITECTURE– Scheme of Evaluation

Qn No	Answer	Mark splitup
	PART A	
1	– Small amount of fast memory – Sits between main memory and CPU – May be located on CPU chip or module	2x1
2	The processor continues with other work after delegating I/O operation to the DMA module. The DMA module transfers the entire block of data, one word at a time, directly to or from memory, without going through the processor. When the transfer is complete, the DMA module sends an interrupt signal to the processor. **the processor is involved only at the beginning and end of the transfer**	2x1
3	Contains a word of data to be **written to** memory or the word **most recently read**	2x1
4	In a two stage instruction pipeline, when second stage is executing the instruction,the first stage takes advantage of any unused memory cycles to fetch and buffer the next instruction. This is called *instruction prefetch* or *fetch overlap*	2
5	MAR, MBR, PC, IR	2x.5
	Part B	
1	○ Registers — In CPU ○ Internal or Main memory — May include one or more levels of cache — “RAM” ○ External memory-Backing store ○ goes down the hierarchy, the following occur: **a. Decreasing cost per bit** **b. Increasing capacity** **c. Increasing access time**	

	d. Decreasing frequency of access of the memory by the processor	
	 Inboard Memory Registers Cache Main Memory Outboard Storage Magnetic Disk CD-ROM CD-RW DVD+RW DVD-RAM Off-line Storage Magnetic Tape MO WORM **Figure 4.1 The Memory Hierarchy**	
2	• Both volatile — Power needed to preserve data • **Dynamic cell** — Simpler to build, smaller — More dense — Less expensive — Larger memory units — Used in main memory Bits stored as charge in capacitors Charges leak Need refreshing even when powered Simpler construction Smaller per bit Slower Essentially analogue-Level of charge determines value — • **Static** — Faster — Used in Cache memory Bits stored as on/off switches No charges to leak No refreshing needed when powered More complex construction Larger per bit	

	More expensive Does not need refresh circuits Digital	
3	 Control signals from I/O module Status signals to I/O module Data bits to and from I/O module Control Logic Buffer Transducer Data (device-unique) to and from environment Figure 7.2 Block Diagram of an External Device *Control logic* associated with the device controls the device's operation in response to direction from the I/O module. The *transducer* converts data from electrical to other forms of energy during output and from other forms to electrical during input. a buffer is associated with the transducer to temporarily hold data being transferred between the I/O module and the external environment; a buffer size of 8 to 16 bits is common.	fig -3 Exp - 3
4		Fig-3 Expl-3

	 Read/write head (1 per surface) Direction of arm motion Surface 9 Platter Surface 8 Surface 7 Surface 6 Surface 5 Surface 4 Surface 3 Surface 2 Surface 1 Surface 0 Spindle Boom Figure 6.5 Components of a Disk Drive	
5	**Control Registers** ➢ variety of processor registers used to control the operation of the processor. Not visible to the user. ➢ Some may be visible to machine instructions executed in a control or operating system mode. ➢ different machines will have different register organizations Four registers are essential to instruction execution: • **Program counter (PC):** Contains the address of an instruction to be fetched • **Instruction register (IR):** Contains the instruction most recently fetched • **Memory address register (MAR):** Contains the address of a location in memory • **Memory buffer register (MBR):** ➢ Contains a word of data to be written to memory or the word most recently read	2+4x1
6	execution of a program consists of the sequential execution of instructions. Each instruction is executed during an instruction cycle made up of shorter subcycles (e.g., fetch, indirect, execute, interrupt).The execution of each subcycle involves one or more shorter operations, that is, micro-operations. Micro-operations are the functional, or atomic, operations of a processor.	Fig-3 Expl-3

	 Program execution Instruction cycle Instruction cycle • • • Instruction cycle Fetch Indirect Execute Interrupt µOP µOP µOP µOP µOP Figure 15.1 Constituent Elements of a Program Execution	
7	 Instruction register Decoder I_0 I_1 I_k Clock Timing generator T_1 T_2 T_n Control unit Flags C_0 C_1 C_m Figure 15.10 Control Unit with Decoded Inputs ○ instruction register - The control unit makes use of the opcode from instruction register and will perform different actions (issue a different combination of control signals) for different instructions. ○ To simplify the control unit logic, decoder geneates a unique logic input for each opcode. ○ The clock portion of the control unit issues a repetitive sequence of pulses for measuring the duration of micro-operations. ○ Period of the clock pulses must be long enough to allow the propagation of signals along data paths and through processor circuitry. ○ the control unit emits different control signals at different time units within a single instruction cycle. So a counter as input to the control unit, with a different control signal being used.	Fig- 3 Expl-3

- At the end of an instruction cycle, the control unit must feed back to the counter to reinitialize it at T1.

Part C

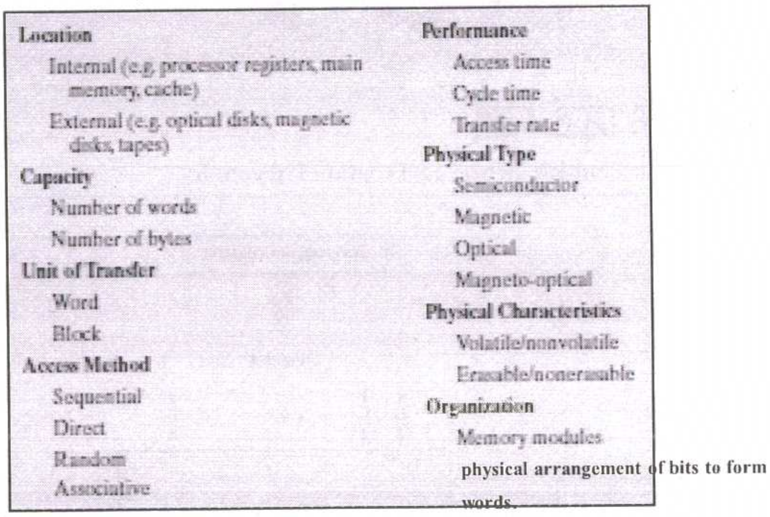
MODULE – 1

III a.

Characteristics of Memory system

~~Fig -8~~ List -4
Expl - 4

Table 4.1 Key Characteristics of Computer Memory Systems



Location
Internal (e.g. processor registers, main memory, cache)
External (e.g. optical disks, magnetic disks, tapes)
Capacity
Number of words
Number of bytes
Unit of Transfer
Word
Block
Access Method
Sequential
Direct
Random
Associative
Performance
Access time
Cycle time
Transfer rate
Physical Type
Semiconductor
Magnetic
Optical
Magneto-optical
Physical Characteristics
Volatile/nonvolatile
Erasable/nonerasable
Organization
Memory modules
physical arrangement of bits to form words.

b.

Fig-4

Expl-3

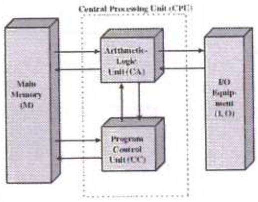

Central Processing Unit (CPU)
Main Memory (M)
Arithmetic-Logic Unit (CA)
Program Control Unit (CC)
I/O Equipment (I, O)

Figure 2.1 Structure of the IAS Computer

- Stored Program concept
- Main memory for storing programs and data
- ALU operating on binary data
- Control unit interpreting instructions from memory and executing
- Input and output equipment operated by control unit

OR

IV a.

Fig-4

Exp-3

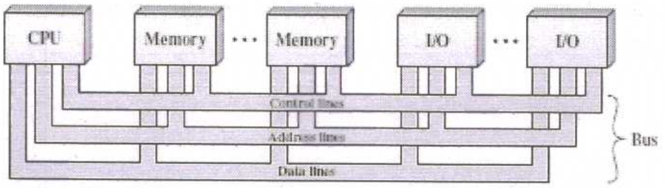
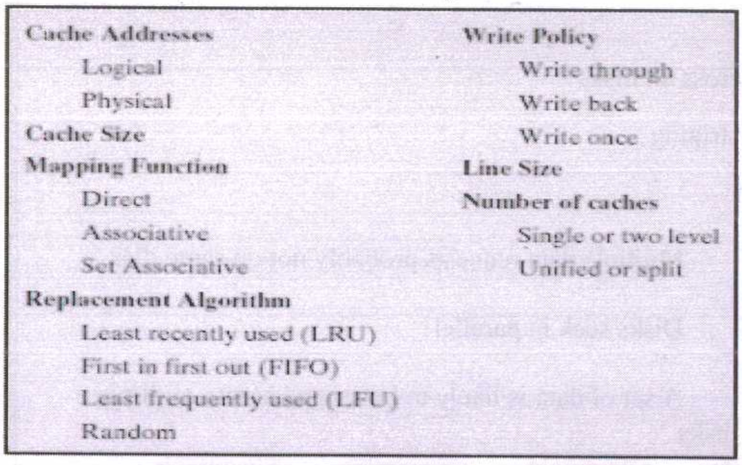


Figure 3.16 Bus Interconnection Scheme

data lines provide a path for moving data among system modules.
The data bus may consist of 32, 64, 128, or even more separate lines, the number of lines being referred to as the *width* of the data bus. Because each line can carry only 1 bit at a time
address lines are used to designate the source or destination of the data on the data bus.
The **control lines** are used to control the access to and the use of the data and address lines. Because the data and address lines are shared by all components,
there must be a means of controlling their use. Control signals transmit both command and timing information among system modules. Timing signals indicate the validity of data and address information. Command signals specify operations to be performed.

b.

Table 4.2 Elements of Cache Design



Cache Addresses	**Write Policy**
Logical	Write through
Physical	Write back
Cache Size	Write once
Mapping Function	**Line Size**
Direct	**Number of caches**
Associative	Single or two level
Set Associative	Unified or split
Replacement Algorithm	
Least recently used (LRU)	
First in first out (FIFO)	
Least frequently used (LFU)	
Random	

Fig -5

Expl-3

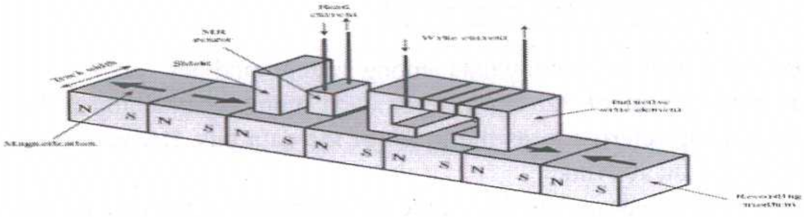
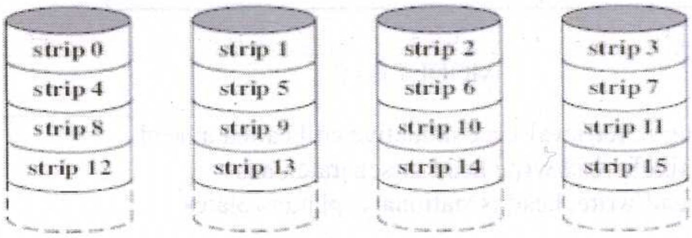
MODULE - 2

V
a.

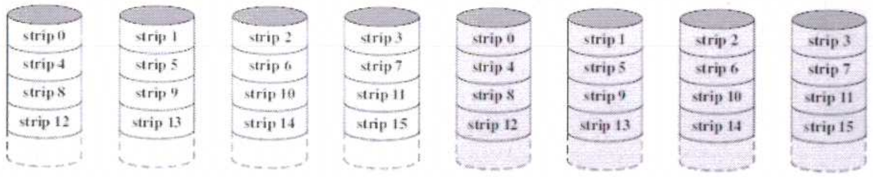
- Recording & retrieval via conductive coil called a head
- May be single read/write head or separate ones
- During read/write, head is stationary, platter rotates
- Write
  - — Current through coil produces magnetic field
  - — Pulses sent to head
  - — Magnetic pattern recorded on surface below
- Read (contemporary)
  - — Separate read head, close to write head
  - — Partially shielded magneto resistive (MR) sensor used for

Fig-3

Expl-3

	reading whose Electrical resistance depends on direction of magnetic field  Read current MR sensor Shield Write current Track width Magnetization Inductive write element Recording medium N S Figure 6.1 Inductive Write/Magnetoresistive Read Head	
b.	RAID 0 — No redundancy — Data striped across all disks — Round Robin striping — Increase speed —— Multiple data requests probably not on same disk —— Disks seek in parallel —— A set of data is likely to be striped across multiple disks  strip 0 strip 1 strip 2 strip 3 strip 4 strip 5 strip 6 strip 7 strip 8 strip 9 strip 10 strip 11 strip 12 strip 13 strip 14 strip 15 (a) RAID 0 (Nonredundant) RAID 1 — Mirrored Disks	3x3

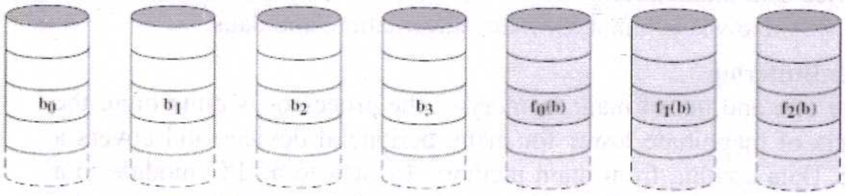
- Data is striped across disks
- 2 copies of each stripe on separate disks
- Read from either
- Write to both
- Recovery is simple
  - Swap faulty disk & re-mirror
  - No down time
- Expensive



(b) RAID 1 (Mirrored)

RAID 2

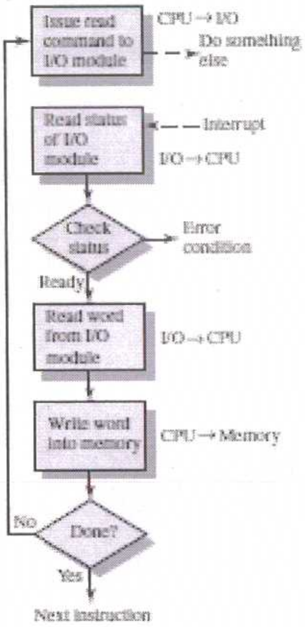
- Disks are synchronized
  - Very small stripes
- Often single byte/word
- Error correction calculated across corresponding bits on disks
- Multiple parity disks store Hamming code error correction in corresponding positions
- Lots of redundancy
  - Expensive
  - Not used



(c) RAID 2 (Redundancy through Hamming code)

OR

VI a.		Fig -3 Expl -4

	 Issue read command to I/O module CPU → I/O Do something else Read status of I/O module Interrupt I/O → CPU Check status Error condition Ready Read word from I/O module I/O → CPU Write word into memory CPU → Memory No Done? Yes Next instruction (b) Interrupt-Driven I/O	
b.	— Control & Timing - to coordinate the flow of traffic between internal resources and external devices. — Processor Communication- ○ **Command decoding** I/O module accepts commands from the processor,typically sent as signals on the control bus ○ **Data:** Data are exchanged between the processor and the I/O module over the data bus. ○ **Status reporting:** peripherals are so slow, it is important to know the status of the I/O module. ○ **Address recognition:** each I/O device has an address. I/O module must recognize one unique address for each peripheral it controls. — Device Communication ○ involves commands, status information, and data — Data Buffering transfer rate into and out of main memory or the processor is quite high, the rate is orders of magnitude lower for many peripheral devices and covers a wide range. Data coming from main memory are sent to an I/O module in a rapid burst. The data are buffered in the I/O module and then sent to the peripheral device at its data rate and vice versa. — Error Detection I/O module is often responsible for **error detection** and for subsequently reporting errors to the processor. One class of errors includes mechanical and electrical malfunctions reported by the device (e.g., paper jam, bad disk track).	Listing - 2.5 Expl-5.5

	MODULE - 3	
VII a.	**User-Visible Registers** A user-visible register is one that may be referenced by means of the machine language that the processor executes. categories: ➢ General purpose ➢ Data ➢ Address ➢ Condition codes **General-purpose registers** can be assigned to a variety of functions by the programmer. general-purpose register can contain the operand for any opcode. example, there may be dedicated registers for floating-point and stack operations. general-purpose registers can be used for addressing functions (e.g., register indirect, displacement) in some cases. **Data registers** Used only to hold data and cannot be employed in the calculation of an operand address. **Address registers** Devoted to a particular addressing mode. Examples: • **Segment pointers:** A segment register holds the address of the base of the segment. There may be multiple registers: for example, one for the operating system and one for the current process. • **Index registers:** These are used for indexed addressing and may be autoindexed. • **Stack pointer:** dedicated register that points to the top of the stack. This allows implicit addressing; push, pop, and other stack instructions need not contain an explicit stack operand. **condition codes** (also referred to as *flags*). partially visible to the user ➢ Condition codes are bits set by the processor hardware as the result of operations. ➢ Eg. an arithmetic operation may produce a positive, negative, zero, or overflow result.	4x2
b.		Structure-4 Function-3

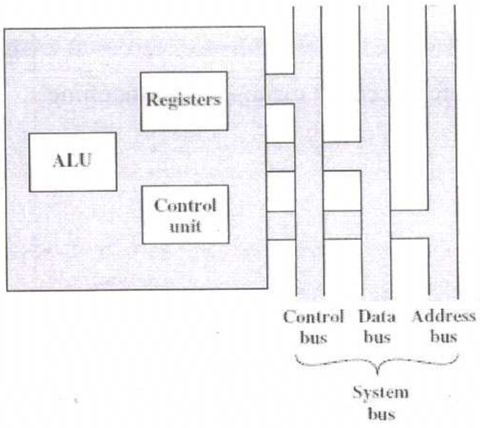


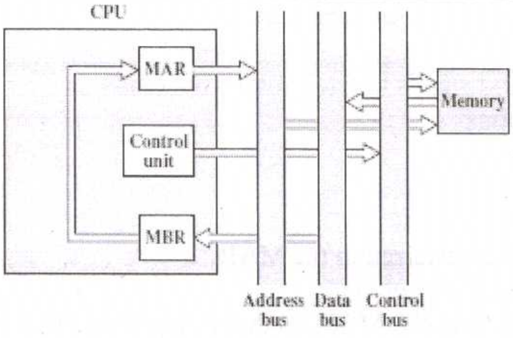
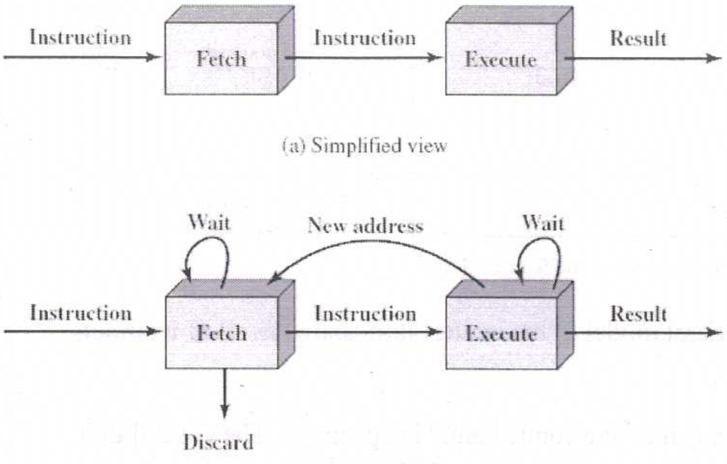
Figure 12.1 The CPU with the System Bus

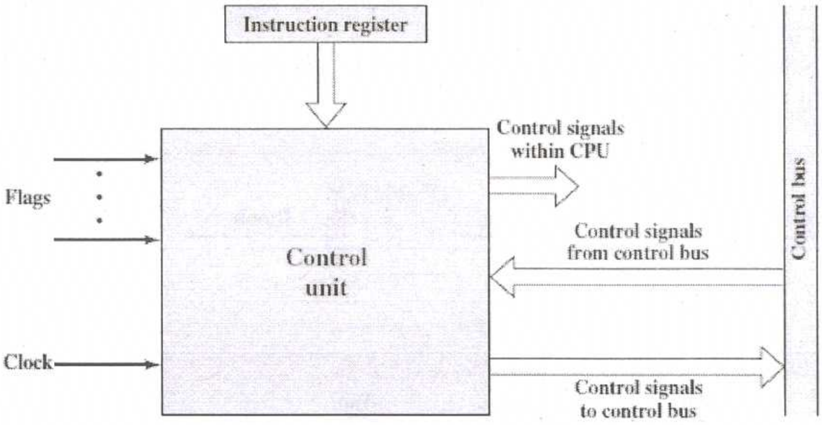
- major components of the processor are
  - an *arithmetic and logic unit* (ALU) and
  - a *control unit* (CU).
- The ALU does the actual computation or processing of data.
- The control unit controls the movement of data and instructions into and out of the processor and controls the operation of the ALU.
- Minimal internal memory, consisting of a set of storage locations, called *registers.*
- The data transfer and logic control paths together known as system *bus.* This element is needed to transfer data between the various registers and the ALU because the ALU operates only on data in the internal processor memory.

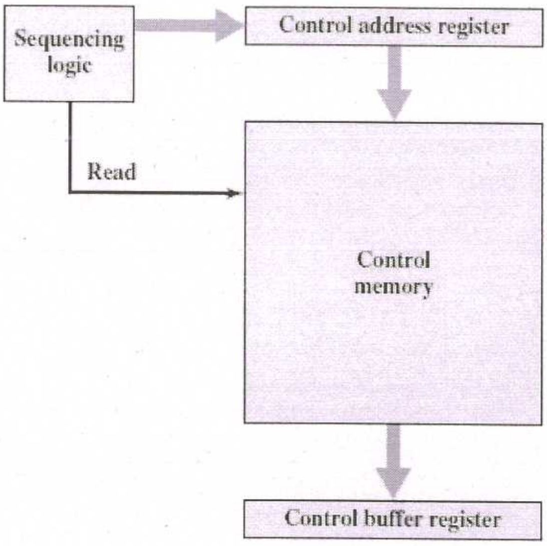
Function of processor is

- ✓ **Fetch instruction:** The processor reads an instruction from memory (register, cache, main memory).
- ✓ **Interpret instruction:** The instruction is decoded to determine what action is required.
- ✓ **Fetch data:** The execution of an instruction may require reading data from memory or an I/O module.
- ✓ **Process data:** The execution of an instruction may require performing some arithmetic or logical operation on data.
- ✓ **Write data:** The results of an execution may require writing data to memory or an I/O module.

	OR	
VIII a.		Fig -4 Expl – 4

	 CPU MAR Control unit MBR Memory Address bus Data bus Control bus Figure 12.7 Data Flow, Indirect Cycle Once the fetch cycle is over, the control unit examines the contents of the IR to determine if it contains an operand specifier using indirect addressing. If so, an *indirect cycle* is performed. The rightmost N bits of the MBR, which contain the address reference, are transferred to the MAR. Then the control unit requests a memory read, to get the desired address of the operand into the MBR.	
b.	 Instruction Fetch Instruction Execute Result Wait New address Wait Instruction Fetch Instruction Execute Result Discard (a) Simplified view (b) Expanded view Figure 12.9 Two-Stage Instruction Pipeline	Fig-3 Expl -4
	MODULE – 4	
IX a.	<u>Fetch cycle</u> t1: MAR <-(PC) t2: MBR <- Memory PC -<(PC) + I t3: IR <- (MBR)	4x2

	where I is the instruction length. Indirect cycle Once an instruction is fetched, the next step is to fetch source operands. includes the following micro-operations: t1: MAR <- (IR(Address)) t2: MBR <- Memory t3: IR(Address) <- (MBR(Address)) The address field of the instruction is transferred to the MAR. This is then used to fetch the address of the operand. the address field of the IR is updated from the MBR, so that it now contains a direct rather than an indirect address. The IR is now in the same state as if indirect addressing had not been used, and it is ready for the execute cycle.	
b.	 Instruction register Control signals within CPU Flags Control unit Control signals from control bus Control bus Clock Control signals to control bus Figure 15.4 Block Diagram of the Control Unit Figure 15.4 is a general model of the control unit, showing all of its inputs and outputs. The inputs are ➢ **Clock:** This is how the control unit “keeps time.” The control unit causes one micro-operation (or a set of simultaneous micro-operations) to be performed for each clock pulse.This is sometimes referred to as the processor cycle time, or the clock cycle time. ➢ **Instruction register:** The opcode and addressing mode of the current instruction are used to determine which micro-operations to perform during the execute cycle. ➢ **Flags:** These are needed by the control unit to determine the status of the processor and the outcome of previous ALU operations. For example, for the increment-and-skip-if-zero (ISZ) instruction, the control unit will increment the PC if the zero flag is set. ➢ **Control signals from control bus:** The control bus portion of the system bus provides signals to the control unit. The outputs are as follows:	Fig-3 Expl-4

	➢ **Control signals within the processor:** These are two types: those that cause data to be moved from one register to another, and those that activate specific ALU functions. ➢ **Control signals to control bus:** These are also of two types: control signals to memory, and control signals to the I/O modules.	
	OR	
X a.	use of multiple processors that can execute in parallel to support a given workload is parallel processing. **Single instruction, single data (SISD) stream:** A single processor executes a single instruction stream to operate on data stored in a single memory. Uniprocessors fall into this category. • **Single instruction, multiple data (SIMD) stream:** A single machine instruction controls the simultaneous execution of a number of processing elements on a lockstep basis. Each processing element has an associated data memory, so that each instruction is executed on a different set of data by the different processors. Vector and array processors fall into this category, • **Multiple instruction, single data (MISD) stream:** A sequence of data is transmitted to a set of processors, each of which executes a different instruction sequence. This structure is not commercially implemented. • **Multiple instruction, multiple data (MIMD) stream:** A set of processors simultaneously execute different instruction sequences on different data sets. SMPs, clusters, and NUMA systems fit into this category.	1+4x1.5
b.	 Sequencing logic Control address register Read Control memory Control buffer register Figure 16.3 Control Unit Microarchitecture The set of microinstructions is stored in the *control memory.* The *control address register* contains the address of the next microinstruction	Fig-4 Expl-4

	to be read. When a microinstruction is read from the control memory, it is transferred to a *control buffer register.* The left-hand portion of that register connects to the control lines emanating from the control unit. Thus, *reading* a microinstruction from the control memory is the same as *executing* that microinstruction. The third element is a sequencing unit that loads the control address register and issues a read command.	