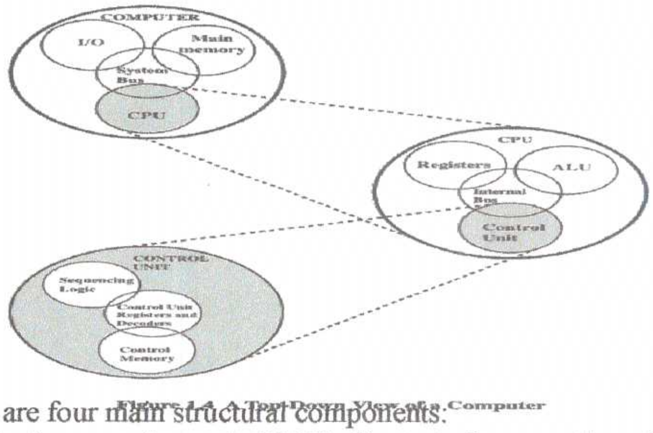
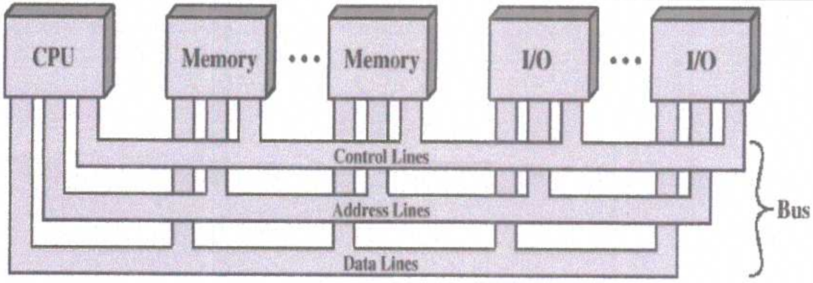
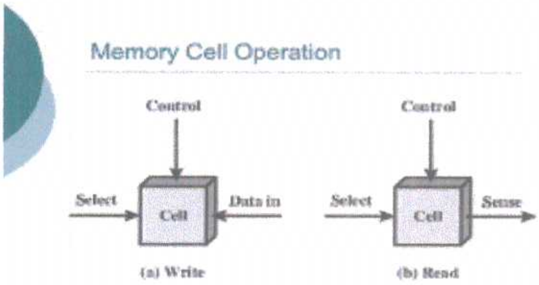
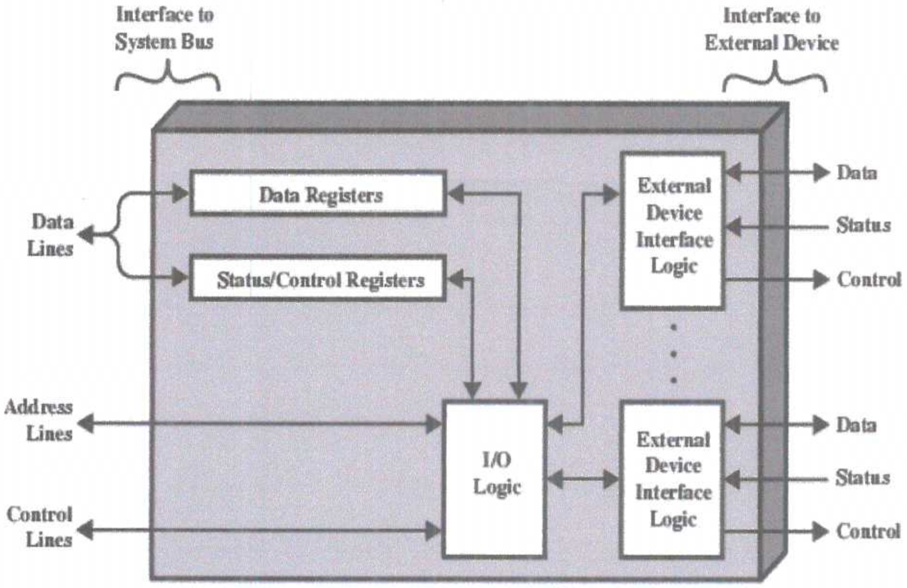
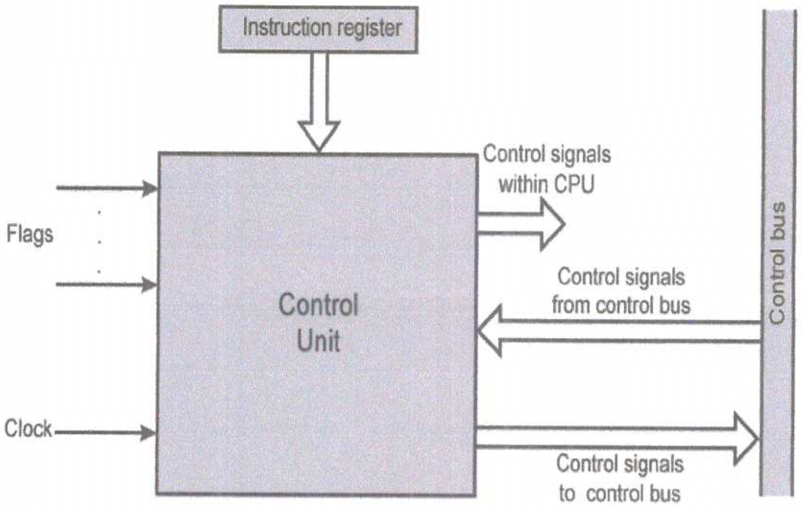
Code 3131 (15)

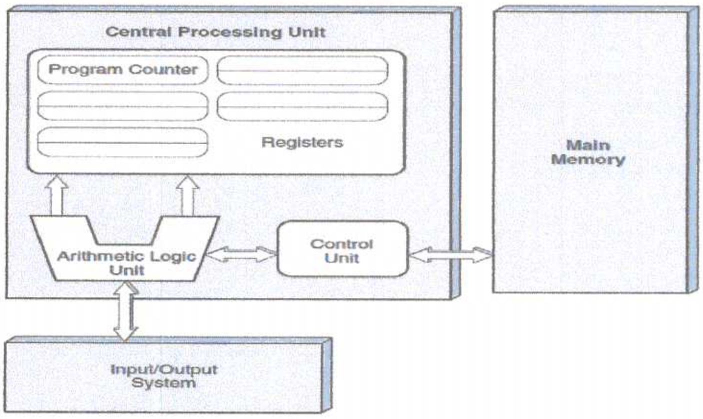
COMPUTER ARCHITECHTURE

Qno.	Key	Split Score	Total Scor e
I.1	1. focuses on the structure and behavior of the computer system 2. refers to the logical aspects of system implementation as seen by the programmer. 3. includes many elements such as instruction sets and formats, data types, addressing modes, number and type of registers, ... 4. helps us to answer the question: How do I design a computer? (Any one)	2	2
I.2.	Data processing,Data Storage, Data movement, Contro;	1/2 *4	2
I.3.	Fetch Instruction, Interpret Instruction,Fetch data,Process data and write data	2	2
I.4.	RAID – Redundent ARRAY of Independent Disk CD – Compact Disk DVD – Digital Verstile Disk HD -High Desity	1/2 *4	2
I.5.	Micro-Operations are the <u>atomic operations</u> of the Processor such as data transfer between registers, transfer between a register & an external bus ALU operation	2	2
Part B			
II.1	Diagram 2 marks explanation 4  COMPUTER I/O Main memory System Bus CPU CPU Registers ALU Internal Bus Control Unit CONTROL UNIT Sequencing Logic Control Unit Registers and Decoders Control Memory Figure 1.4 A Top-Down View of a Computer There are four main structural components: • **Central processing unit (CPU):** Controls the operation of the computer and performs its data processing functions; often simply referred to as processor. Consists of *- Control unit: Controls the operation of the CPU and hence the computer •* *-Arithmetic and logic unit (ALU): Performs the computer's data processing functions*	2+4	6

	-Registers: Provides storage internal to the CPU *-CPU interconnection: Some mechanism that provides for communication among the control unit, ALU, and registers* ***•Main memory:***Stores data. • **I/O:** Moves data between the computer and its external environment. • **System interconnection:** Some mechanism that provides for communication among CPU, main memory, and I/O. A common example of system interconnection is by means of a system bus, consisting of a number of conducting wires to which all the other components attach.		
II 2.	 CPU Memory Memory I/O I/O Control Lines Address Lines Data Lines Bus Data Bus Function of a data bus is to send data from one device to another Data is passed in parallel or serial manner – Parallel will normally pass in a multiple of 8-bits at a time – Serial passes one bit at a time **Address Bus** CPU needs to read an instruction (data) from a given location in memory. address bus provides address the source or destination of data. Address Bus width determines maximum memory capacity of system **Control Lines** Controls the access to the data and address lines Controls the use of the data and address lines Typical control lines include the following: Memory write, Memory read I/O write, I/O read Transfer ACK, Bus request, Bus grant, Interrupt request Interrupt ACK, Clock, Reset	6	6
II 3.	The basic element of a semiconductor memory is the memory cell. Although a variety of electronic technologies are used, all semiconductor memory cells share certain properties: • They exhibit two stable (or semistable) states, which can be used to represent binary 1 and 0. • They are capable of being written into (at least once), to set the state. • They are capable of being read to sense the state.	2+2+2	6

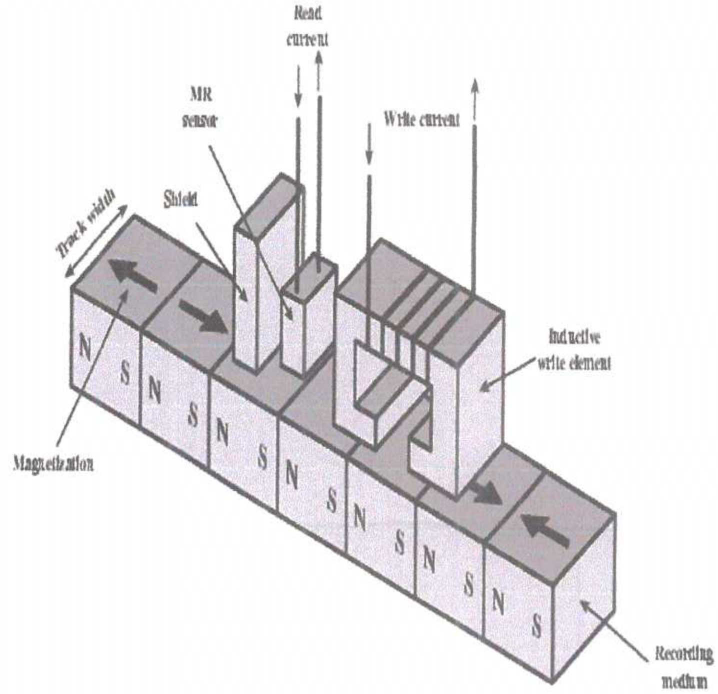
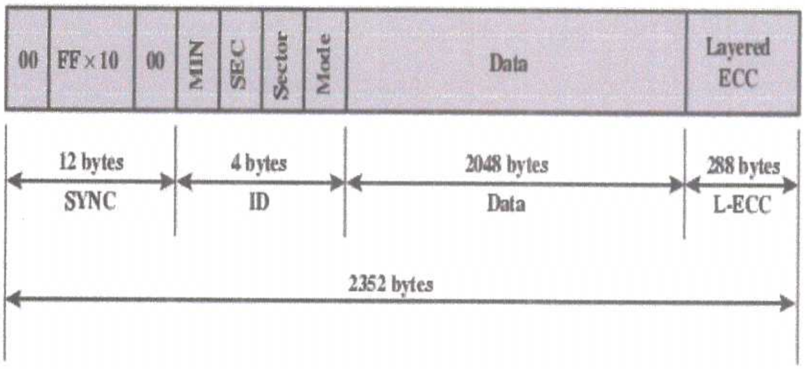
	 Memory Cell Operation Control Select Cell Data in (a) Write Control Select Cell Sense (b) Read Most commonly, the cell has three functional terminals capable of carrying an electrical signal. The select terminal, as the name suggests, selects a memory cell for a read or write operation. The control terminal indicates read or write. For writing, the other terminal provides an electrical signal that sets the state of the cell to 1 or 0. For reading, that terminal is used for output of the cell's state.		
II4.	• Improvement in the uniformity of the magnetic film surface to increase disk reliability • A significant reduction in overall surface defects to help reduce read-write errors • Ability to support lower fly heights (described subsequently) • Better stiffness to reduce disk dynamics •Greater ability to withstand shock and damage	6	6
II 5.	The major functions or requirements for an I/O module fall into the following categories: Control and timing, Processor communication, Device communication, Data buffering, Error detection  Interface to System Bus Interface to External Device Data Lines Data Registers Status/Control Registers Address Lines Control Lines I/O Logic External Device Interface Logic Data Status Control External Device Interface Logic Data Status Control Explanation of the above	2 + 2+ 2	6

II 6.	Fetch Cycle t1 : MAR <-- (PC) t2 : MBR <-- Memory PC <-- (PC) t3 : IR <-- (MBR) Indirect Cycle t1 : MAR <-- (PC) t2 : MBR <-- Memory t3 : IR (Address) <-- (MBR(Address)) Interrupt Cycle t1 : MBR <-- (PC) t2 : MAR <-- Save Address PC <-- Routine_Address t3 : Memory <-- (MBR) Explain each. Each carry 2 ma	2 + 3	6
II 7.	 Instruction register Flags Clock Control Unit Control signals within CPU Control signals from control bus Control signals to control bus Control bus Functions of control unit **Sequencing**	2+4	6

	Causing the CPU to step through a series of micro-operations **Execution** Causing the performance of each micro-op. This is done using Control Signals Sequencing Causing the CPU to step through a series of micro-operations Control Signals Clock One micro-instruction (or set of parallel micro- instructions) per clock cycle Instruction register, Op-code for current instruction Determines which micro-instructions are performed, Flags State of CPU, Results of previous operations From control bus, Interrupts, Acknowledgements,		
Part C			
III a)	The Von Neumann arxchitecture is based on three key concepts: • *Data and instructions are stored in a single read–write memory.* • *The contents of this memory are addressable by location, without regard to the type of data contained there.* • *Execution occurs in a sequential fashion (unless explicitly modified) from one instruction to the next.*  Central Processing Unit Program Counter Registers Arithmetic Logic Unit Control Unit Main Memory Input/Output System FIGURE 1.4 The von Neumann Architecture	4+3	7

III b)	SRAM Vs DRAM -bothare volatile -dynamic memory cell is simpler and smaller and hence densely packed than sram -dram is less expensive than sram -refresher circuit is required for dram and not required for sram - dram suitable for large size memory - sram fasterthan dram - sram used for cache memory -dram is used for main memory.	8	8
IV a)	**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 listing 4 and explanation 4	4 +4	8
IV b)	**Cache Addresses** **Write Policy** Logical Write through Physical Write back	3+4	7

	Write once **Cache Size** **Mapping Function** Direct Associative Set Associative **Replacement Algorithm** Least recently used (LRU) First in first out (FIFO) Least frequently used (LFU) Random **Line Size** **Number of caches** Single or two level Unified or split Listing 3 and explanation 4		
V a)	**Head Motion** Fixed head (one per track) Movable head (one per surface) **Disk Portability** Nonremovable disk Removable disk **Sides** Single sided Double sided **Platters** Single platter Multiple platter **Head Mechanism** Contact (floppy) Fixed gap Aerodynamic gap (Winchester) Listing 3 and explanation 4	3 + 4	7
V b)	**Magnetic read and write mechanism** -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 (traditional)** -Magnetic field moving relative to coil produces current -Coil is the same for read and write **Read (contemporary)** -Separate read head, close to write head -Partially shielded magneto resistive (MR) sensor -Electrical resistance depends on direction of magnetic field -High frequency operation Higher storage density and speed		

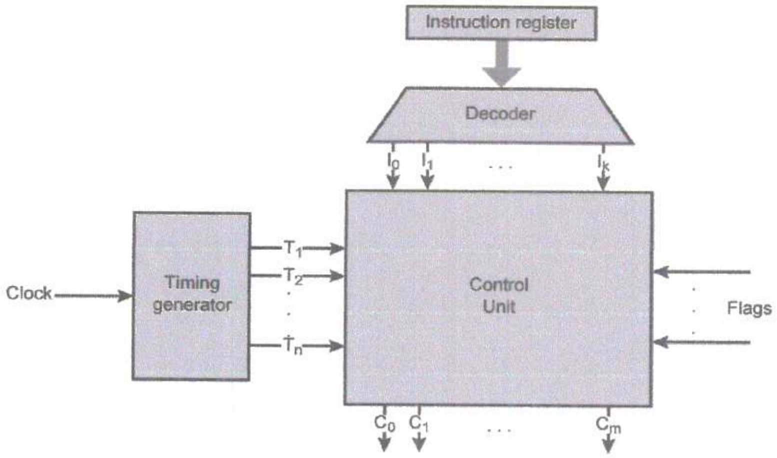
	 Read current MR sensor Write current Shield Track width Inductive write element Magnetization Recording medium N S N S N S N S N S N S N S		
VI a)	 00 FF × 10 00 MIN SEC Sector Mode Data Layered ECC 12 bytes SYNC 4 bytes ID 2048 bytes Data 288 bytes L-ECC 2352 bytes 4marks diagram • ***Sync:*** The sync field identifies the beginning of a block. It consists of a byte of all 0s, 10 bytes of all 1s, and a byte of all 0s. • ***Header:***The header contains the block address and the mode byte. Mode 0 specifies a blank data field; mode 1 specifies the use of an error-correctingcode and 2048 bytes of data; mode 2 specifies 2336 bytes of user data with no error-correcting code. • ***Data:***User data. • ***Auxiliary:*** Additional user data in mode 2. In mode 1, this is a 288-byte error-correcting code.	4+4	8

VI b)	**Table attached**	7	7
VII a)	A user-visible register is one that may be referenced by means of the machine language that the processor executes. We can characterize these in the following categories: • **General purpose** - can be assigned to a variety of function - ussage is orthogonal to instructions - also there are dedicated register such as floating point or stack reg. -some support use of these registers in address calculation • **Data** -may be used only to hold data and not in the calculation of address • **Address** -segment pointer – holds the address of the base of the segment - index registers -used for indexed addressing - stack pointer – points to the top of the stack. • **Condition codes** - also called flag - set by the processor as the result of operation listing 2 Explain each 6	2+6	8
VII b)	Instructiojn Cycle Fetch, Decode , operand read, Execute, Store result, Interrupt List 3 and explanation 4	3+4	7
VIII a)	A Pipelining is a series of stages, where some work is done at each stage in parallel. The stages are connected one to the next to form a pipe. Instructions enter at one end, progress through the stages, and exit at the other end, Pipelining is a speedup technique instructions are overlapped in execution on a processor. Instruction cycle has number of subcycles such as instruction fetch, decode, operand fetch, execute, store result and interrupt. Each sub cycle is considered as one stage. Therefore Instruction cycle has six stages. 3 marks if n instructions are executed one by one the time consumed is n x 6 (considering each stage consumes 1 unit of time)units. 1 mark pipeline explanation 5 marks diagram 5 marks time calculation 1 mark	+1+5+5+1	15
IX	Hardwired Implementation of control unit. **Control unit inputs** **Flags and control bus** -Each bit means something **Instruction register** -Op-code causes different control signals for each different instruction	(2+5) + (2+6)	15

-Unique logic for each op-code
-Decoder takes encoded input and produces single output
-n binary inputs and 2^n outputs

Clock

-Repetitive sequence of pulses
-Useful for measuring duration of micro-ops
-Must be long enough to allow signal propagation
-Different control signals at different times within instruction cycle
-Need a counter with different control signals for t1, t2 etc.

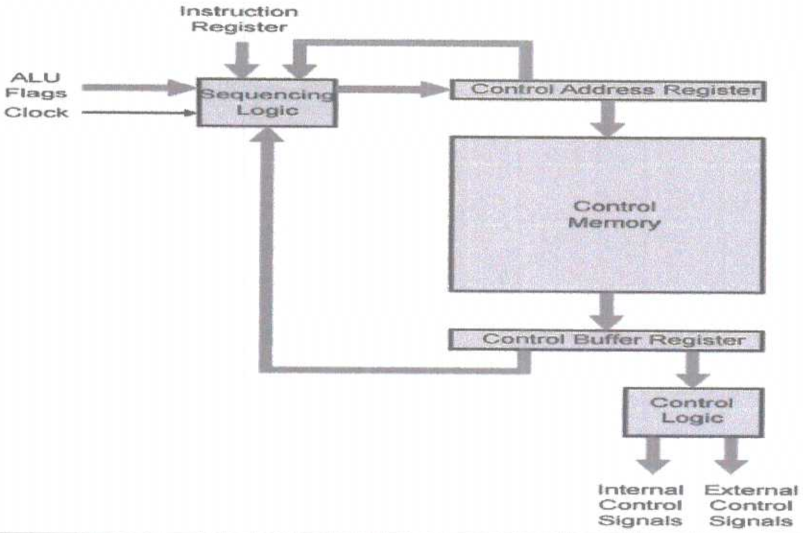


(2 marks diagram)

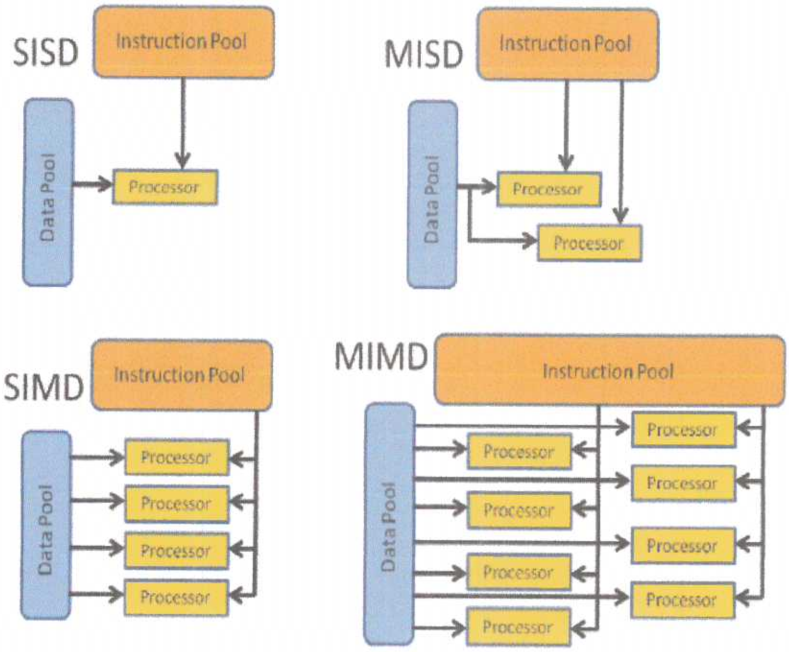
Disadvantages

-Complex sequencing & micro-operation logic
-Difficult to design and test
-Inflexible design
-Difficult to add new instructions

Micro program controlled



	(2marks for diagram) In addition to the use of control signals, each micro-operation is described in symbolic notation. This notation looks suspiciously like a programming language. In fact it is a language, known as a microprogramming language. Each line describes a set of micro-operations occurring at one time and is known as a microinstruction. A sequence of instructions is known as a microprogram, or firmware. Two types of micro instruction Horizontal and vertical. A microinstruction is interpreted as follows: 1. To execute this microinstruction, turn on all the control lines indicated by a1 bit; leave off all control lines indicated by a 0 bit. The resulting control signalswill cause one or more microperations to be performed. 2. If the condition indicated by the condition bits is false, execute the next microinstruction in sequence. 3. If the condition indicated by the condition bits is true, the next microinstruction to be executed is indicated in the address field. The control unit functions as follows: 1. To execute an instruction, the sequencing logic unit issues a READ command to the control memory. 2. The word whose address is specified in the control address register is read into the control buffer register. 3. The content of the control buffer register generates control signals and next address information for the sequencing logic unit. 4. The sequencing logic unit loads a new address into the control address register based on the next-address information from the control buffer register and the ALU flags. All this happens during one clock pulse.		
X	A taxonomy first introduced by Flynn • 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. • 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. (7 marks)	7+8	15

	 SISD Instruction Pool Data Pool Processor MISD Instruction Pool Data Pool Processor Processor SIMD Instruction Pool Data Pool Processor Processor Processor Processor MIMD Instruction Pool Data Pool Processor Processor Processor Processor Processor Processor Processor Processor 8marks		

VI

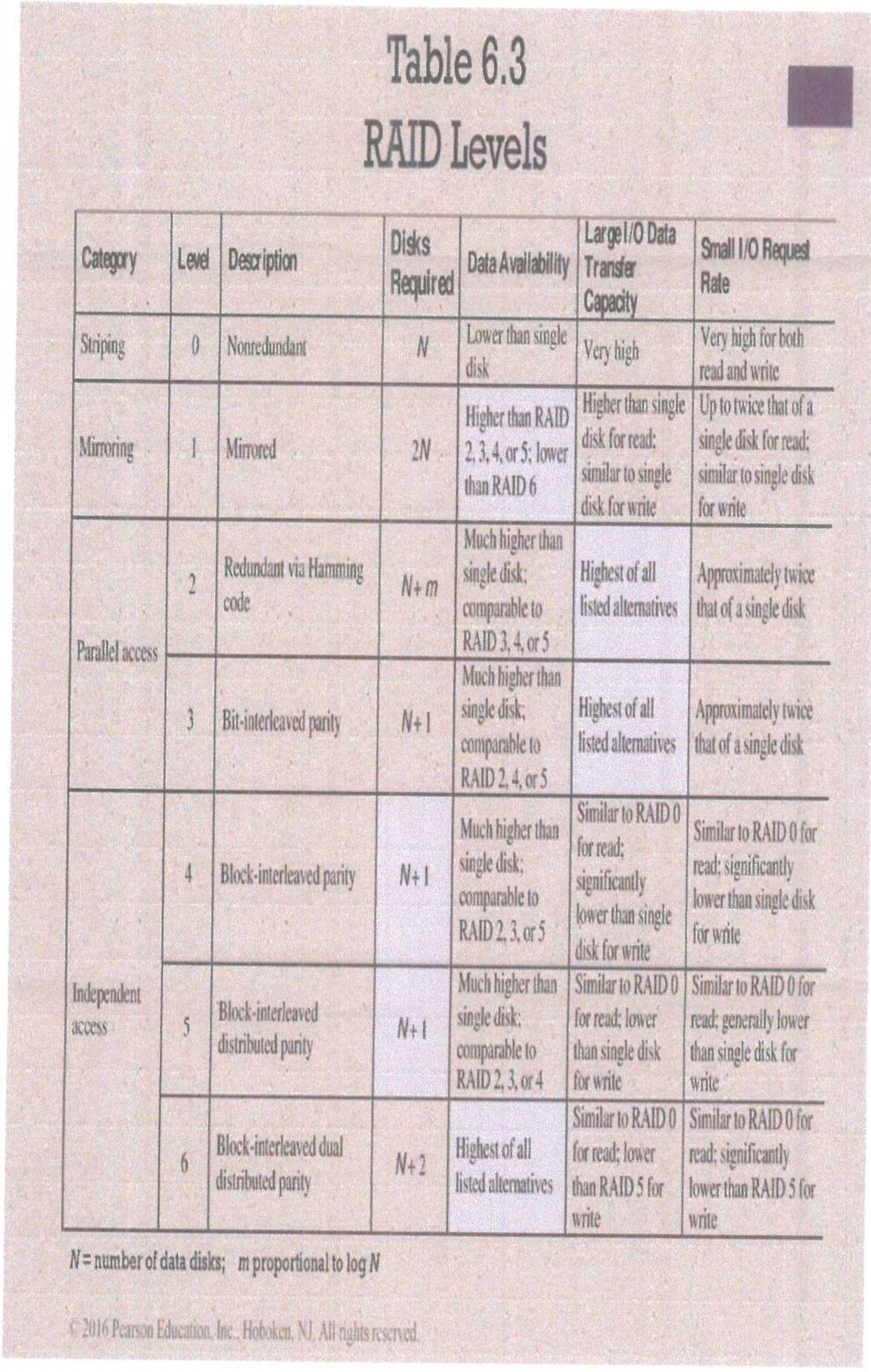


Table 6.3
RAID Levels

Category	Level	Description	Disks Required	Data Availability	Large I/O Data Transfer Capacity	Small I/O Request Rate
Striping	0	Nonredundant	N	Lower than single disk	Very high	Very high for both read and write
Mirroring	1	Mirrored	$2N$	Higher than RAID 2, 3, 4, or 5; lower than RAID 6	Higher than single disk for read; similar to single disk for write	Up to twice that of a single disk for read; similar to single disk for write
Parallel access	2	Redundant via Hamming code	$N+m$	Much higher than single disk; comparable to RAID 3, 4, or 5	Highest of all listed alternatives	Approximately twice that of a single disk
	3	Bit-interleaved parity	$N+1$	Much higher than single disk; comparable to RAID 2, 4, or 5	Highest of all listed alternatives	Approximately twice that of a single disk
Independent access	4	Block-interleaved parity	$N+1$	Much higher than single disk; comparable to RAID 2, 3, or 5	Similar to RAID 0 for read; significantly lower than single disk for write	Similar to RAID 0 for read; significantly lower than single disk for write
	5	Block-interleaved distributed parity	$N+1$	Much higher than single disk; comparable to RAID 2, 3, or 4	Similar to RAID 0 for read; lower than single disk for write	Similar to RAID 0 for read; generally lower than single disk for write
	6	Block-interleaved dual distributed parity	$N+2$	Highest of all listed alternatives	Similar to RAID 0 for read; lower than RAID 5 for write	Similar to RAID 0 for read; significantly lower than RAID 5 for write

N = number of data disks; m proportional to $\log N$