1) Describe the three key concepts of the von Neumann architecture.

Ans → i) Central Processing Unit (CPU) → The brain of computer, responsible for executing instructions. Comprises of several key components → CU (Control Unit), ALU (Arithmetic Logic Unit, Registers

ii) Main Memory Unit → Stores data as well as instructions. Programs and data share same memory space. Stored program concept enables easier programming.

iii) Input / Output (I/O) Devices → Facilitate communication between the computers and external world. Ex → keyboards, mouse, displays, printers, etc.

2) Discuss the two approaches for handling multiple interrupts.

Ans → i) Disable Interrupts → In this method, whenever an interrupt ~~is odi~~ occurs, the processor temporarily disables all further interrupts. This ensures current interrupt is ~~being~~ handled without being interrupted by another. Once the first Interrupt's service routine (ISR) finishes, the processor re-enables interrupts.

ii) Interrupt Priority → This approach assigns priorities to different interrupts. A higher priority interrupt can interrupt a lower priority interrupt that is currently being serviced. The processor stores the state of the lower-priority ISR and jumps to the higher priority one.

3) a) One byte = reserved for op code

remaining 8 bytes (32 bits - 8 bits) = 24 ~~bytes~~ 24 bits

So, max^m addressable memory capacity = 2^{24} = 16 MB

b) i) 32-bit address bus & 16 bit data bus:

Advantage → Wider address bus allows accessing the full 16 mB memory

Disadvantage → Narrower data bus means it takes two transfers to fetch a complete 32-bit instruction. This can slow down instruction fetching compared to wider data bus.

ii) 16-bit address bus & 16-bit data bus →

Advantage → Single transfer can fetch a complete instruction due to matching data bus width.

Disadvantage → Significantly limited address space (2^{16} bytes = 64kB). Most programs would exceed their memory limit.

4) Both the program counter (PC) & instruction register (IR) need to hold a complete 32 bit instructions. Therefores, they both require 32 bits.

Q4) The improvement achieved when fetching instructions and operands using a 32 bit microprocessor as compared to a 16-bit microprocessor.

Given: 20% instructions 32 bits long
40% instructions 16 bits long
40% instruction 8 bit long

16-bit (no. of bus cycle)	Instruction Length	32-bit (no. of bus cycle)
2 bus cycles	32 bit.	1 bus cycle
1 cycle	16-bit	0.5 bus cycle
0.5 cycle	8-bit	0.25 cycle

Avg: 20% × 2 + 40% × 1 + 0.5 × 40%
= 1.2 bus cycle

Avg: 20% × 1 + 40% × 0.5 + 40% × 0.25
= 0.7 bus cycles

So, improvement is 1.2 - 0.7 = 0.5 bus cycles/instruction.

In percentage terms, the improvement is 0.5 / 1.2 × 100 = 41.67%.

Q5) What are the advantages and disadvantages of the logical cache over the physical cache

Ans→

Logical Cache (Virtual)	Physical Cache
Advantages → i) Faster access due to avoiding MMU translation. ii) Simplest cache design	Advantages → i) Simpler cache coherence ii) Reduced MMU overhead
Disadvantages → i) more complex MMU with translation cache ii) Cache coherence challenges in multi-processor systems.	Disadvantages → i) Slower access due to MMU trans ii) more complex cache design with internal trans

Q6) Cache size = N words
No. of bits to address cache size = $\log_2 N$
Block size = B words
Block offset = $\log_2 B$
No. of blocks in cache = N/B
No. of bits to represent block = $\log_2 (N/B)$
No. of sets in cache = $(N/B)/16$
Bits to represent sets : $\log_2 (N/B)/16$
Length (direct) = 10 bits
Length (4-set associative) = x

Here, direct mapped cache is compared with set associative cache, but in both block offset is same.

So, $10 + \log_2(N/B) = x + \log_2(N/B \div 16)$

$\Rightarrow 10 + \log_2(N/B) = x + \log_2(N/B) - \log_2(16)$

$\Rightarrow 10 = x - 4 \Rightarrow x = 14$. Length of bits = 14 for each set associative

7) System is a 32-bit address.

Cache size = 64kB = 64×1024 bytes

$= 2^6 \times 10^{10} = 2^{16}$

So, cache bits = 16

So, tag bits = 32 - 16 = 16 bit

As, cache is 8-way set associative. So, we have to transfer 3 bits to tag side.

So, final tag bits = 16 + 3 = 19

No. of bits in Tag field = 19

8) a) Field lengths of direct mapped cache:

	Tag	Line	Word
111111	11	444	1
BBBBBB	BB	2EEE	3
666666	66	1999	2

b) Field lengths of associative ~~set~~ cache:

Tag → 22 bits, word → 2 bits

	Tag	Word
111111	44444	1
B66666	19999	2
BBBBBB	2EEEE	3

Field lengths of set-associative cache →

Tag → 9 bits, Set - 13 bits, Words → 2 bit

	Tag	Set	Word
111111	82	444	1
C66666	CC	1999	2
BBBBBB	177	EEE	3