FIT3159 Applied 04
Name: Chong Jet Shen
ID: 33517495

Question 1 (150 words max):
You are developing firmware for a new line of smart thermostats. The device uses a simplified instruction set for power-efficient operations.

A. Explain the concept of an instruction set and how it supports efficient firmware execution.
- A set of possible instruction types that a CPU can execute directly.
- Simplified instructions reduce transistor switching, lowering power consumption.
- Fixed-length instructions streamline fetch and decode.
- Dedicated opcodes for common tasks reduce software overhead.

B. Illustrate how a typical "Load Accumulator" (LDA) instruction would behave in this CPU, considering the fetch-decode-execute phases.
- Address of LDA opcode sent to memory and fetch into instruction register during fetch phase.
- Control unit identifies LDA and extracts operand address during decode phase.
- Loads operand address to MAR and data moved to accumulator (ACC) during execute phase.

C. Describe how a "Jump to Subroutine" (JMS) instruction helps manage modular control logic and what steps are involved in executing it.
- Uses shared routines which reduce firmware size as code is reused.
- JMS operand makes PC jump to the subroutine address, executes the subroutine, then returns and resumes the main program.

D. Explain how Load/Store instructions facilitate interaction between processing and memory, particularly in low-power embedded systems.
- Low-power embedded systems often perform operations on data stored in registers.
- Load/Store instructions facilitates moving data between memory and register.

Question 2 (150 words max):

You are consulting on a lightweight processor design for a fitness tracker.

A. Compare 1-, 2-, and 3-operand instruction formats in terms of encoding efficiency and instruction decoding time.

1-operand	2-operand	3-operand
- Very compact instructions. - Simple decoding with few operand fields. - Higher overhead as more instructions needed for encoding and decoding.	- Balanced code size with fewer instructions, reducing overhead. - Simple register-based decoding which avoids frequent memory access.	- Fewer instructions per task due to larger instructions. - Longer decoding time for more fields to process.

B. Define operand encoding and explain how choosing the right format reduces instruction size and memory footprint.

- Defines how instruction fields are represented in binary.
- Fewer operands reduce instruction size.
- Smaller instruction size reduces time to fetch it from memory.
- Smaller instructions waste less memory when in smaller devices.

C. Describe how a stack machine might benefit expression evaluation or compact control logic in this setting.

- Very compact executable code, resulting in smaller storage requirements and shorter fetch time.
- Using a LIFO list instead of registers, fewer bits per instructions needed.
- Smaller opcodes as stack machine has implicit operands, always top of the stack.
- No need for dedicated link register as stack is use to return address.
- Simplified state management by pushing registers to the stack.

D. Justify the need for rapid stack memory access in maintaining responsiveness and minimising energy usage during user interaction.

- For low latency interrupt handling, CPU state must be pushed/popped rapidly to resume execution as slow stack access results in delayed response.
- For real time sensor processing, CPU needs to process data fast enough, else samples are dropped, leading to inaccurate readings.
- Reduce energy usage with reduced memory traffic as main memory access consumes more power than stack operations.
- Slow stack leaves CPU active longer, consuming more energy.

Question 3 (150 words max):

Your team is building an edge computing device to process video feeds on-site using AI inference.

A. Explain how operand encoding in the instruction set affects energy consumption and compute time for AI workloads.

- Defines how many operands an instruction uses where requiring fewer operands leads to lower fetch energy from memory.

B. Discuss the role of stack-based architectures in reducing hardware complexity and power draw.

- Uses a LIFO stack for operands without the need for a large register file, reducing the area of silicon chip needed.
- No operands fields in instructions needed, resulting in smaller decoders and logic gates.

C. Evaluate how using instructions with different operand counts might impact performance in high-throughput inference pipelines.

- Lower operand counts reduces memory increases code density with fewer memory fetches, but requires more instructions, lowering throughput.
- Higher operand counts makes fewer instructions per op, resulting in higher overall throughput

D. Explain why stack memory speed is vital for real-time AI models operating on limited compute resources.

- For real-time responsiveness like object detection, CPU must save registers to the stack quickly to avoid missing the next frame.
- Slow stacks forces more access to high power memory, wasting energy.

Question 4 (150 words max):

A robotics startup is designing a control board for a delivery drone. Timely I/O operations are crucial for sensor feedback and motor control.

A. Compare polled and interrupt-driven I/O in terms of drone system responsiveness.

Polled	Interrupt-driven
- Works well for periodic tasks. - Sensor event may be missed until next check. - Best for low complexity drones with slow sensor rates	- Interrupts are handled near instant. - Best for high performance drones requiring real time sensor fusion and motor control.

B. Within an interrupt-driven approach, explain how polled servicing, register-decoded servicing, and vectored interrupts differ in response time and scalability.

Polled servicing	Register-decoded servicing	Vectored interrupts
- High latency if it's checked last. - Simple to implement, but poor for high speed systems when adding more devices.	- Decent response time as only checks one register. - Supports 10s of devices as limited by register width, but no inherent prioritization.	- Fastest possible response time due to no software checks. - High priority interrupts preempts lower ones. - Supports 100s of interrupts

C. Define Direct Memory Access (DMA) and explain how it reduces latency in real-time telemetry processing.

- A hardware feature that allows peripherals to transfer data directly to/from memory without CPU intervention.
- DMA controller autonomously transfer entire data blocks to memory, bypassing CPU bottlenecks.
- Transfers run in parallel with CPU tasks, resulting in very low latency with no interrupt overhead per byte.

Question 5 (150 words max):

A data center operator must balance performance with cost when upgrading storage infrastructure.

A. Discuss how the speed-cost tradeoff is managed across cache, SSDs, and HDDs in modern systems.

Cache	SSDs	HDDs
- Very fast data transfer speed till near instant access. - Most expensive storage per GB	- Fast data transfer speeds. - Cheaper storage compared to cache	- Slowest data transfer speeds. - Cheapest storage with high capacity

B. Explain how Moore's Law has historically shaped the economics of different storage layers.

- All storage layers over the years benefitted from increased storage density and cutting cost.
- Enabled faster and larger caches and SSDs
- HDDs growth decreased due to physical limitations.

C. Identify key access delay factors in hard disks and provide techniques used in enterprise setups to mitigate them.

Key access delay factors:
- Rotational latency as disk must rotate to position data under the head.
- Contention due to multiple requests queued

Mitigation techniques:
- Higher disk rotation per minute (RPM) for lower rotational latency.
- Distribute load access disks to mitigate contention.

D. Compare SSDs and HDDs for use in large-scale video streaming services, focusing on throughput and durability.

SSDs	HDDs
- Higher throughput as it can handle more concurrent streams. - Near instant seek times. - SSDs degrade over time when it handles heavy writes.	- Struggles with high concurrency. - Slower seek times due to mechanical delay. - Doesn't degrade over time with no write limits

Question 6 (150 words max):

A company is planning to deploy a hybrid cloud backup solution with RAID-based redundancy.

A. Define Kryder's Law and describe how it influences the long-term viability of large-scale storage systems.
- Kryder's Law observes the areal density of magnetic disks doubles every 18 – 24 months.
- HDDs becomes more affordable for up to petabyte-scale backups to be economically feasible.
- Fewer physical disks needed for same capacity, reducing datacenter footprint in space and maintenance.
- RAID redundancy becomes cheaper to implement at scale with more terabytes per drive.

B. Explain how RAID 0 and RAID 1 could be used to optimise either speed or redundancy in backup staging areas.

RAID 0	RAID 1
- Data is split across multiple disks with no redundancy. - Able to write 2x faster, but no fault tolerance. - Best used in short-term staging where it holds large backup batches before archiving.	- Data is duplicated on two identical disks. - Immediate redundancy for critical backups with fault tolerance where when one disk fails, the other disk has a full copy. - Best used for final backup destination for critical data.

C. Compare RAID 5 and RAID 6 in terms of failure tolerance and overhead, and identify which is more suitable for archival purposes.

RAID 5	RAID 6
- Single "check" block spread across all drives in the array which helps survive 1 disk failure. - Less overhead as single check blocks leaves more usable storage.	- Pairs of "check" blocks spread across all drives in the array which help survive 2 disk failures. - More overhead as twice as many check blocks leaves less usable storage.

Which is more suitable?

- RAID 6 is generally more suitable for higher fault tolerance.
- Larger drives increases rebuild failure risks, so fault tolerance is a critical factor.
- RAID 6's higher overhead isn't a huge factor as archiving is usually write once and read rarely.