# Month Quiz

**Topic:** Computer Organization  
**Difficulty:** Medium  
**Questions:** 5  
**Type:** MCQ

**Question 1:** In a classic 5-stage pipeline (IF, ID, EX, MEM, WB), a load instruction is immediately followed by an instruction that uses the loaded value. Why is a one-cycle stall (load-use stall) typically required even when forwarding paths exist?

A) A. The loaded data is only produced at the end of the MEM stage, so it is not available in time for the dependent instruction's EX stage even with forwarding.

B) B. Forwarding cannot be implemented for register-to-register dependencies because it violates register file semantics.

C) C. The dependent instruction must wait for the WB stage to complete to preserve program order and avoid write hazards.

D) D. Load instructions change the program counter alignment, which requires an extra cycle to synchronize the pipeline.

**Answer:** A. The loaded data is only produced at the end of the MEM stage, so it is not available in time for the dependent instruction's EX stage even with forwarding.

**Explanation:** A load instruction fetches data from memory during the MEM stage, and that data becomes available only at the end of MEM. A dependent instruction needs the operand during its EX stage one cycle later. Even with forwarding, there is no earlier pipeline stage producing the loaded data to forward from in time, so a single-cycle stall is required (or other techniques like load-to-use interlock elimination via reordering or hardware prefetching). Options B, C and D are incorrect: forwarding is specifically to handle register-to-register dependencies; waiting for WB is unnecessary when forwarding can be used for most ALU results; and loads do not change PC alignment in a way that requires an extra cycle.

**Question 2:** A direct-mapped cache has total data storage of 64 KB with a block (line) size of 16 bytes. For a 32-bit byte-addressable memory, how many index bits are required to select the cache line?

A) A. 10

B) B. 12

C) C. 16

D) D. 8

**Answer:** B. 12

**Explanation:** Number of cache lines = cache size / block size = 64 KB / 16 B = (65536) / 16 = 4096 lines. Index bits = log2(4096) = 12. (Offset bits = log2(16) = 4; tag bits = 32 - 12 - 4 = 16.) Option B is correct. Options A, C, and D are incorrect because they correspond to different numbers of lines (2^10 = 1024, 2^16 = 65536, 2^8 = 256) which do not match 4096 lines.

**Question 3:** On a little-endian machine, a 32-bit value 0x12345678 is stored starting at memory address 0x1000. What byte value is found at address 0x1002?

A) A. 0x12

B) B. 0x34

C) C. 0x56

D) D. 0x78

**Answer:** B. 0x34

**Explanation:** Little-endian stores the least-significant byte at the lowest address. The 32-bit value 0x12345678 has bytes (low to high): 0x78 (LSB), 0x56, 0x34, 0x12 (MSB). These map to addresses: 0x1000 -> 0x78, 0x1001 -> 0x56, 0x1002 -> 0x34, 0x1003 -> 0x12. Therefore address 0x1002 contains 0x34. Options A, C, and D are other bytes of the word but at different offsets.

**Question 4:** When a CPU encounters a TLB miss but the page table entry indicates the page is present (valid) in memory, what is the typical sequence of actions the hardware performs?

A) A. Trigger a page fault and let the OS bring the page into memory.

B) B. Perform a page table walk to obtain the physical frame mapping, load the translation into the TLB, then retry the faulting access.

C) C. Flush the entire TLB and caches to ensure coherence, then continue execution.

D) D. Immediately abort the instruction and mark the process as terminated due to an invalid address translation.

**Answer:** B. Perform a page table walk to obtain the physical frame mapping, load the translation into the TLB, then retry the faulting access.

**Explanation:** A TLB miss means the translation is not cached in the TLB. If the page table entry shows the page is present, the hardware (or OS via a trap on some designs) performs a page table walk to obtain the physical frame number, inserts that translation into the TLB, and then retries the instruction. This is not a page fault (option A) because the page is present. Flushing the entire TLB/caches (option C) is unnecessary; and aborting the process (option D) is incorrect for a valid page.

**Question 5:** Which characteristic is most commonly associated with classic RISC (Reduced Instruction Set Computer) architectures?

A) A. A large number of complex addressing modes and multi-step complex instructions.

B) B. A load-store approach where only explicit load and store instructions access memory; arithmetic instructions operate on registers.

C) C. Instructions are typically microcoded and may perform several memory accesses in a single instruction.

D) D. Highly variable-length instruction encodings to pack many addressing forms into the ISA.

**Answer:** B. A load-store approach where only explicit load and store instructions access memory; arithmetic instructions operate on registers.

**Explanation:** Classic RISC designs emphasize a simple instruction set with fixed-size instructions, a load-store architecture (memory accessed only by load/store instructions), and simple addressing modes so that most instructions operate directly on registers and can complete in a single cycle. Option B captures this key trait. Options A and C describe CISC characteristics (complex addressing modes and microcoding), and option D (highly variable-length encodings) is more typical of some CISC ISAs; RISC tends to prefer fixed-length encodings.