# NIMCET Computer Awareness Sample Paper-10

**Duration: 15 Minutes** **Maximum Marks: 120**

## Instructions

- This paper contains **20** Multiple Choice Questions (Single Correct).
- Each correct answer carries **+6 marks**.
- Each incorrect answer carries: **−1.5** marks.
- Unattempted questions carry **0** marks.
- Only one option is correct for each question.
- Use of mobile phones, smartwatches, calculators, or any electronic gadgets is strictly prohibited.

**Q1.** A multi-core processor utilizes a directory-based cache coherence protocol to track line states across individual local caches. If a core attempts to execute a write operation to a memory line currently marked as `Shared` in multiple remote nodes, what specific sequence of hardware broadcast messages must the central directory controller issue over the interconnect fabric to safely complete this state transaction?

(A) Broadcast an `Invalidate` signal to all sharing nodes, wait for acknowledgment vectors, and then transition the requesting core's cache line state to `Modified`.

(B) Broadcast a `Write-Back` command to all sharing nodes, fetch their local modifications, and then transition the requesting core's state to `Shared`.

(C) Issue a `Read-Inverted` signal to the main memory, bypass local cache lines, and write directly via a write-through buffer.

(D) Force all remote sharing nodes to immediately execute a context switch until the primary core de-asserts its master lock line.

**Q2.** An advanced superscalar processor features a speculative execution unit supported by a Reorder Buffer (ROB). During an instruction execution cycle, at which precise hardware pipeline stage are instructions allowed to permanently

update the primary architectural register file states or commit modifications to the external data cache memory?

(A) Immediately at the end of the Write-Back (WB) stage out-of-order.

(B) Strictly during the Commit/Retire stage in strict, original program order.

(C) During the Execution (EX) phase via internal bypass and forwarding multiplexers.

(D) As soon as the instruction enters the Instruction Decode (ID) reservation station arrays.

**Q3.** A pipelined execution block running at 2.5 GHz features an explicit structural penalty due to branches. If conditional branch instructions account for 20% of a program workload, and the internal branch prediction logic achieves a success rate of 85%, calculate the overall performance degradation in Cycles Per Instruction (CPI) if every mispredicted branch introduces a fixed hardware flush penalty of 4 clock cycles.

(A) 0.12

(B) 0.15

(C) 0.24

(D) 0.30

**Q4.** A microprogrammed control engine utilizes an alternative nanoprogramming execution framework. The control store layout is split into two distinct tiers: a Micro-control Memory containing 512 words of 16 bits each, and a Nano-control Memory containing 64 unique control vectors of 48 bits each. Calculate the absolute total capacity in bits required to implement this complete dual-tier control storage hardware layout.

(A) 11264 bits

(B) 12288 bits

(C) 27648 bits

(D) 32768 bits

**Q5.** A modern Solid-State Drive (SSD) controller handles physical layout constraints through a Flash Translation Layer (FTL). Which of the following technical descriptions accurately explains why an SSD controller must incorporate garbage collection algorithms alongside a background Wear Leveling mechanism?

(A) NAND flash memory allows random overwrite operations on a byte level, but structural wear constraints mandate sequential access tracks.

(B) Flash memory cells must be erased in large block units before they can be rewritten at smaller page levels, causing a mismatch that requires logical mapping updates to distribute cell wear evenly.

(C) The FTL requires constant background garbage collection to clear read-disturb bit flips from volatile SRAM sector caches.

(D) Mechanical platter alignment requires rotational indexing shifts to balance read write tasks across multiple logical surfaces.

**Q6.** A processor uses a vector-priority interrupt structure with nested handling features. If a lower-priority interrupt service routine (ISR) is currently executing, and a high-priority interrupt lines fires, which hardware actions are carried out automatically by the internal nested priority controller?

(A) The current lower-priority ISR execution is forcefully terminated, and its registers are permanently discarded.

(B) The running ISR's context is preserved on the system stack, the CPU branches immediately to the higher-priority vector entry, and the low-priority line is temporarily masked.

(C) The high-priority request is logged in an edge latch and held in a wait loop until the running ISR executes an explicit `IRET` instruction.

(D) The processor splits its execution registers into symmetric parallel halves to run both routines simultaneously.

**Q7.** An 8-bit computational logic unit uses a fixed-point numeric register configured to hold fractional formats. If the bit pattern residing in the cell is given as the string `10111100`, evaluate the true base-10 value represented if the layout is

decoded using standard 1's Complement fractional format with the radix point located directly next to the sign bit.

(A) $-0.531250$

(B) $-0.5234375$

(C) $-0.468750$

(D) $-0.4765625$

**Q8.** A scientific telemetry datalink transmits floating-point parameters formatted in strict compliance with the IEEE 754 single-precision specification. If a packet arrives containing the hexadecimal string representation `0xC0A00000`, parse the binary fields to determine the exact corresponding base-10 real number encoded.

(A) $-2.5$

(B) $-5.0$

(C) $-1.25$

(D) $-10.0$

**Q9.** An enterprise RAID storage array applies a block-level parity distribution matrix across a 5-disk configuration. If the array utilizes an advanced Reed-Solomon erasure coding algorithm variant configured as a $(4, 2)$ scheme (4 data chunks, 2 parity chunks), what is the maximum number of simultaneous disk failures the storage system can tolerate without experiencing catastrophic unrecoverable data loss?

(A) 1

(B) 2

(C) 3

(D) 4

**Q10.** An 8-bit computational register finishes an arithmetic addition sequence using standard signed 2's complement logic boundaries. If the two source inputs passed to the adder unit are $A =$ `0x4C` and $B =$ `0x3F`, calculate the hexadecimal

result string generated in the destination location along with the final status of the Carry-Out Flag ($C$) and the Arithmetic Overflow Flag ($V$).

(A) Result = `0x8B`, $C = 0, V = 1$

(B) Result = `0x8B`, $C = 1, V = 0$

(C) Result = `0x0B`, $C = 1, V = 1$

(D) Result = `0x7B`, $C = 0, V = 0$

**Q11.** Convert the hexadecimal fractional numeric representation `0x2D.C` directly into its corresponding balanced octal base-8 representation format notation.

(A) $55.6_8$

(B) $45.6_8$

(C) $55.3_8$

(D) $45.14_8$

**Q12.** A 32-bit physical memory system connects to a 64 KB 8-way set-associative cache memory module. The structural specification defines the line cache block size as 64 bytes. If the directory table tracks line states by allocating exactly 1 Valid bit and 1 Dirty bit per cache line entry, calculate the absolute total hardware memory capacity required to implement the structural directory tag array (excluding the storage space for raw data blocks).

(A) 2.25 KB

(B) 2.50 KB

(C) 2.75 KB

(D) 3.00 KB

**Q13.** A server platform employs a demand paging virtual memory system. Accessing a data word out of the physical RAM takes exactly 60 ns. If a page fault arises, the operating system expends a service time latency of 10 ms if an empty page block is immediately available, and 20 ms if the selected victim page is dirty and must be copied back to the storage drive. Assuming that 30% of the replaced page blocks are dirty, calculate the maximum permissible page fault probability

threshold ($p$) required to guarantee that the overall effective memory access time (EMAT) remains $\leq 130$ ns.

(A) $p \leq 5.38 \times 10^{-6}$

(B) $p \leq 7.00 \times 10^{-6}$

(C) $p \leq 3.50 \times 10^{-6}$

(D) $p \leq 1.16 \times 10^{-5}$

**Q14.** A memory controller organizes structural RAM modules using a 16-way low-order address interleaving configuration. The absolute cycle time required to completely process a single storage bank operation is 64 ns, while the continuous bus pipeline dispatch rate allows a new independent read request to fire off to a subsequent bank every 4 ns. Compute the total time required to fetch a stream sequence of 32 continuous address memory words from this framework.

(A) 124 ns

(B) 128 ns

(C) 188 ns

(D) 192 ns

**Q15.** A multi-threaded processing architecture uses a 3-level hierarchical page table structure to manage address translation pathways under a 40-bit virtual workspace. The hardware framework includes a dedicated Translation Lookaside Buffer (TLB) providing a local lookup speed of 6 ns. If the system registers a reliable TLB Hit Ratio of 94%, and each physical RAM access transaction requires an access latency of 50 ns, compute the absolute effective address translation latency performance of the system.

(A) 15.36 ns

(B) 12.12 ns

(C) 21.36 ns

(D) 24.00 ns

**Q16.** Apply Karnaugh mapping reduction constraints to optimize the following five-variable Boolean switching function map down to its absolute minimal Sum-of-Products (SOP) design presentation layout: $F(A, B, C, D, E) = \sum m(2, 3, 6, 7, 10, 11, 14, 15, 18, 19, 22, 23, 26, 27, 30, 31)$.

(A) $F = C$

(B) $F = \overline{A} \cdot C$

(C) $F = D$

(D) $F = \overline{B} \cdot D$

**Q17.** An asymmetric switching network is governed by the structural Boolean logic expression equation: $F(X, Y, Z) = X \cdot \overline{Y} + \overline{X} \cdot Z$. Determine the exact minimized Product-of-Sums (POS) structure for the complementary logic implementation function ($\overline{F}$).

(A) $\overline{F} = (X + Z) \cdot (\overline{X} + \overline{Y})$

(B) $\overline{F} = (\overline{X} + Y) \cdot (X + \overline{Z})$

(C) $\overline{F} = (X + Y) \cdot (\overline{X} + Z)$

(D) $\overline{F} = \overline{X} \cdot \overline{Y} + X \cdot Z$

**Q18.** A digital combinational logic system requires the implementation of a 2-input Exclusive-NOR (XNOR) gate logic module. If the manufacturing plant mandates that the logic layout must be produced using the absolute minimal count of standard 2-input universal NAND gates, calculate the precise quantity of individual NAND gates required assuming uncomplemented input streams are passed from the source.

(A) 4

(B) 5

(C) 6

(D) 3

**Q19.** In contemporary multi-tenant cloud storage infrastructures and network storage architectures, what specialized technology allows multiple virtual machines to

bypass the hypervisor layer and share a single physical Peripheral Component Interconnect Express (PCIe) storage device directly by presenting multiple isolated virtual functional endpoints to the host system?

(A) Single Root I/O Virtualization (SR-IOV)

(B) Non-Volatile Memory Express over Fabrics (NVMe-oF)

(C) Direct Memory Access Tunneling (DMAT)

(D) Software-Defined Storage Multiplexing (SDSM)

**Q20.** A high-availability enterprise database infrastructure deploys a consensus algorithm to maintain write synchronization across distributed geographic computing nodes without relying on a central coordination authority. Identify the structural algorithmic protocol explicitly optimized to prevent split-brain conditions while maintaining strict linearizable consistency across these active database replicas.

(A) Raft Consensus Protocol or Paxos Scheme

(B) Network Time Protocol (NTP) Synchronizer

(C) Secure Hash Algorithm (SHA-256) Matrix

(D) Two-Phase Commit (2PC) Broker Routing

# Detailed Solutions

## Q1.

### Solution

**Concept:** In directory-based cache coherence protocols, when a core requests write access to a cache line currently shared across multiple nodes, the central directory controller must enforce serialization. It must invalidate all remote cached copies and wait for explicit confirmations before authorizing the state transition.

**Solution:**

Let's analyze the sequence of events under a standard directory protocol (e.g., MESI directory scheme):

(a) A local core attempts to write to a line marked as `Shared` ($S$). It issues a write request (`Read-with-Intent-to-Modify` or `Upgrade`) to the central directory.

(b) The directory checks its tracking bits and notices that multiple remote nodes currently hold the target line in the `Shared` state.

(c) The directory broadcasts an `Invalidate` message across the interconnect network to all sharing nodes recorded in its presence vector.

(d) Each remote node invalidates its local copy and replies with an acknowledgment vector.

(e) After collecting all acknowledgments, the central directory grants exclusive ownership to the requesting core, shifting its state to `Modified` ($M$).

This corresponds exactly to option (A).

**Final Answer:** Invalidate shared copies and set the line to Modified.

**Answer: (A)**

Go Back to Question 1

## Q2.

### Solution

**Concept:** To maintain precise exceptions and program correctness in an out-of-order superscalar processor, instructions execute speculatively out-of-order but are forced to log their results into a Reorder Buffer (ROB). They are only permitted to permanently alter architectural states sequentially.

**Solution:**

Let's trace the instruction lifecycle through an out-of-order engine:

- **Execution / Write-Back Stage:** Instructions calculate results out-of-order and write them to temporary storage inside physical registers or the ROB. This speculative data can be shared internally via forwarding multiplexers but is not permanent.
- **Commit / Retire Stage:** Instructions reach the head of the circular ROB queue in their **strict, original program order**.

Once an instruction safely reaches the front of the ROB without triggering a branch misprediction or an execution exception, it commits. Only during this precise stage are values written permanently to the primary architectural register file or committed to the external data cache memory.

**Final Answer:** Strictly during the Commit/Retire stage in strict, original program order.

**Answer: (B)**

Go Back to Question 2

## Q3.

### Solution

**Concept:** The performance degradation in Cycles Per Instruction (CPI) caused by branch mispredictions can be calculated by computing the probability of a branch misprediction occurring per instruction and multiplying it by the associated hardware flush stall penalty:

$$\Delta\text{CPI} = \text{Branch Frequency} \times \text{Misprediction Rate} \times \text{Stall Penalty}$$

**Solution:**

Let's break down the given parameters:

- **Branch Frequency:** Conditional branches represent 20% of all instructions $\implies$ 0.20.
- **Branch Prediction Success Rate:** 85% $\implies$ 0.85.
- **Branch Misprediction Rate:** 100% − 85% = 15% $\implies$ 0.15.
- **Hardware Flush Penalty:** 4 clock cycles.

Substitute these values directly into the performance degradation formula:

$$\Delta\text{CPI} = 0.20 \times 0.15 \times 4$$

$$\Delta\text{CPI} = 0.03 \times 4 = 0.12$$

**Final Answer:** $\boxed{0.12}$

**Answer: (A)**

Go Back to Question 3

## Q4.

### Solution

**Concept:** A nanoprogrammed control engine optimizes control store capacity by introducing a two-level hierarchical storage memory structure. The total physical hardware size in bits is the sum of the capacities of the two separate memory arrays:

$$\text{Total Bits} = \text{Micro-control Capacity} + \text{Nano-control Capacity}$$

**Solution:**

Let's compute the explicit bit storage required for each tier independently:

- **Micro-control Memory Array:** Holds 512 words, where each word is 16 bits wide:

$$\text{Micro Capacity} = 512 \times 16 = 8192 \text{ bits}$$

- **Nano-control Memory Array:** Holds 64 vectors, where each control vector is 48 bits wide:

$$\text{Nano Capacity} = 64 \times 48 = 3072 \text{ bits}$$

Sum the capacities of both tiers to determine the total control store size:

$$\text{Total Hardware Capacity} = 8192 \text{ bits} + 3072 \text{ bits} = 11264 \text{ bits}$$

**Final Answer:** $\boxed{11264 \text{ bits}}$

**Answer: (A)**

Go Back to Question 4

## Q5.

### Solution

**Concept:** NAND flash memory exhibits an architectural asymmetry: data can be written or programmed at a granular **page level**, but blocks can only be cleared or erased at a larger, coarse **block level**.

**Solution:**

Let's analyze the technical rationale behind the Flash Translation Layer (FTL) algorithms:

- Because flash cells cannot be overwritten in place without first being erased, updating an existing page requires writing the new data to an empty page elsewhere and marking the old page as invalid/stale.
- **Garbage Collection:** Over time, blocks become littered with stale pages. The FTL runs background garbage collection to consolidate valid pages into a new block so the old block can be completely erased and reclaimed.
- **Wear Leveling:** Each physical block can only handle a limited number of program-erase (P/E) cycles before failing. The FTL maps logical blocks to different physical addresses over time to distribute write wear evenly across the entire drive, preventing premature failure of heavily accessed sectors.

This maps to option (B).

**Final Answer:** Flash blocks must be erased before rewriting, requiring wear leveling.

**Answer: (B)**

Go Back to Question 5

## Q6.

### Solution

**Concept:** A nested vectored priority interrupt management scheme allows an executing software routine to be safely preempted by a higher-priority hardware interrupt line while preserving state integrity.

**Solution:**

Let's trace the hardware actions performed when a high-priority interrupt asserts during a low-priority Interrupt Service Routine (ISR):

(a) The priority controller determines that the newly asserted interrupt has a higher priority than the currently running software context.

(b) The CPU completes its current instruction, pushes critical execution state registers (like the Program Counter and Status Register) onto the system stack, and saves the context.

(c) The controller masks out equal or lower priority lines to prevent unwanted nested loops.

(d) The CPU retrieves the unique branch address for the high-priority request from the vector table and jumps directly to its execution path.

This sequence is described in option (B).

**Final Answer:** Current ISR is saved and higher-priority ISR is executed.

**Answer: (B)**

Go Back to Question 6

## Q7.

### Solution

**Concept:** In an 8-bit signed 1's Complement fractional format with the radix point situated immediately to the right of the sign bit ($b_0.b_1b_2b_3b_4b_5b_6b_7$), the sign bit $b_0$ carries a negative weight of $-(1-2^{-7}) = -1+2^{-7}$, or can be decoded by inverting all bits of the absolute magnitude.

**Solution:**

Let's analyze the given 8-bit binary string: `10111100`.

- The sign bit (MSB) is `1`, which indicates a negative number.

To find the base-10 value in 1's complement representation, invert all the bits to discover its positive magnitude:

$$\text{Bit inversion of } \texttt{10111100} = \texttt{01000011}_2$$

Now, evaluate this inverted binary fractional value with the radix point placed to the right of the first bit (`0.1000011`):

$$\text{Magnitude} = 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 0 \cdot 2^{-3} + 0 \cdot 2^{-4} + 0 \cdot 2^{-5} + 1 \cdot 2^{-6} + 1 \cdot 2^{-7}$$

$$\text{Magnitude} = 0.5 + 0.015625 + 0.0078125 = 0.5234375$$

Apply the negative sign derived from the initial sign bit state:

$$\text{Value} = -0.5234375$$

**Final Answer:** $\boxed{-0.5234375}$

**Answer: (B)**

Go Back to Question 7

## Q8.

### Solution

**Concept:** An IEEE 754 single-precision floating-point number is parsed by splitting its 32 bits into three fields: Sign (1 bit), Biased Exponent (8 bits), and Fractional Mantissa (23 bits).

**Solution:**

Let's convert the hexadecimal value `0xC0A00000` into its raw 32-bit binary layout:

$$\texttt{0xC0A00000} = \texttt{1100 0000 1010 0000 0000 0000 0000 0000}_2$$

Group the bits into their respective fields:

- **Sign Bit** ($S$)**:** Bit 31 is `1` $\implies$ Negative value ($-$).
- **Biased Exponent** ($E$)**:** Bits [30:23] are $\texttt{10000001}_2 = 129_{10}$.

$$\text{Actual Exponent } e = E - 1bias = 129 - 127 = 2$$

- **Fractional Mantissa** ($f$)**:** Bits [22:0] are $\texttt{01000000000000000000000}_2$.

$$f = 0 \cdot 2^{-1} + 1 \cdot 2^{-2} = 0.25$$

$$\text{Normalized Mantissa } M = 1 + f = 1.25$$

Calculate the final base-10 real value:

$$\text{Value} = (-1) \times M \times 2^e = -1.25 \times 2^2 = -1.25 \times 4 = -5.0$$

**Final Answer:** $\boxed{-5.0}$

**Answer: (B)**

Go Back to Question 8

## Q9.

### Solution

**Concept:** A Reed-Solomon erasure coding scheme configured with parameters $(n, k)$ formats data into $k$ original data chunks and adds $m = n - k$ parity check blocks. The maximum number of simultaneous failures the system can tolerate without data loss equals the number of parity blocks $(m)$.

**Solution:**

Let's look at the parameters of the given storage system configuration:

- The array uses a $(4, 2)$ Reed-Solomon design variant layout.
- Total data chunks $(k) = 4$
- Total parity chunks $(m) = 2$

The properties of Reed-Solomon codes guarantee that any $k$ surviving chunks out of the total $n$ blocks are mathematically sufficient to reconstruct the entire original dataset. Consequently, the array can survive a maximum loss of exactly:

$$\text{Max Fault Tolerance} = m = 2 \text{ disk failures}$$

**Final Answer:** $\boxed{2}$

**Answer: (B)**

Go Back to Question 9

## Q10.

### Solution

**Concept:** To find the carry-out ($C$) and overflow ($V$) flags for 8-bit signed twos-complement addition, add the binary numbers. $C$ represents the carry out of the most significant bit (MSB), and $V$ is set if adding two numbers with identical sign bits produces a result with a different sign bit.

**Solution:**

Let's convert hexadecimal source operands $A = \texttt{0x4C}$ and $B = \texttt{0x3F}$ to 8-bit binary:

$$A = \texttt{0100 1100}_2 \quad \text{(positive, sign bit 0)}$$

$$B = \texttt{0011 1111}_2 \quad \text{(positive, sign bit 0)}$$

Perform the binary addition:

$$\begin{array}{rl} \texttt{0100 1100} & (A) \\ +\texttt{0011 1111} & (B) \\ \hline \texttt{1000 1011} & (\text{Result} = \texttt{0x8B}) \end{array}$$

Evaluate the status flags:

- **Carry-Out Flag ($C$):** No carry bit is generated past the 8th bit position, so $C = 0$.
- **Overflow Flag ($V$):** We added two positive source values (sign bits 0) and obtained a negative result (sign bit 1). This unexpected sign change indicates an arithmetic overflow condition, so $V = 1$.

**Final Answer:** $\boxed{\text{Result} = \texttt{0x8B}, C = 0, V = 1}$

**Answer: (A)**

Go Back to Question 10

## Q11.

### Solution

**Concept:** To convert a fractional number from hexadecimal (base-16) to octal (base-8), map each hexadecimal digit to its equivalent 4-bit binary group, then regroup the bits into 3-bit vectors starting from the radix point.

**Solution:**

Let's convert the components of `0x2D.C` into binary:

$$\texttt{2} \rightarrow \texttt{0010}, \quad \texttt{D} \rightarrow \texttt{1101}, \quad \texttt{C} \rightarrow \texttt{1100}$$

Combine these blocks around the radix point:

$$\text{Binary Layout} = \texttt{0010 1101 . 1100}_2$$

Now, regroup the bits into 3-bit chunks, working outward from the radix point:

- **Integer Side (left of radix):** Regroup `00101101` from right to left:

$$\ldots \underline{001}\ \underline{011}\ \underline{101} \rightarrow 1_8, 3_8, 5_8 \implies 55_8$$

- **Fractional Side (right of radix):** Regroup `1100` from left to right, padding with trailing zeros:

$$\underline{110}\ \underline{000} \cdots \rightarrow 6_8, 0_8 \implies .6_8$$

Combine the integer and fractional components to get the final base-8 representation:

$$\text{Result} = 55.6_8$$

**Final Answer:** $\boxed{55.6_8}$

**Answer: (A)**

Go Back to Question 11

## Q12.

### Solution

**Concept:** The total capacity of a cache's directory tag array is calculated by multiplying the total number of cache lines by the bit width required per line entry. The entry width is the sum of the tag bits and any associated status tracking bits.

**Solution:**

Let's first determine how the 32-bit physical address space is divided into fields:

- **Block Offset Bits:** Given a line size of 64 bytes = $2^6$ bytes, the offset requires $\log_2(64) = 6$ bits.
- **Index Bits:** Calculate the total number of lines in the cache:

$$\text{Total Lines} = \frac{\text{Total Cache Size}}{\text{Block Size}} = \frac{64\text{ KB}}{64\text{ bytes}} = \frac{64 \times 1024}{64} = 1024\text{ lines}$$

Since this is an 8-way set-associative cache, group these lines into sets:

$$\text{Total Sets} = \frac{1024\text{ lines}}{8\text{ lines/set}} = 128\text{ sets} = 2^7\text{ sets} \implies \text{Index} = 7\text{ bits}$$

- **Tag Bits per Line Entry:** Subtract the index and offset widths from the total address width:

$$\text{Tag Width} = 32\text{ bits} - (7\text{ bits} + 6\text{ bits}) = 32 - 13 = 19\text{ bits}$$

Each line entry in the cache directory stores the tag bits along with the state tracking bits:

$$\text{Bits per Line} = 19\text{ bits (Tag)} + 1\text{ bit (Valid)} + 1\text{ bit (Dirty)} = 21\text{ bits}$$

Multiply this by the total number of cache lines to find the full size of the directory tag array:

$$\text{Total Capacity} = 1024\text{ lines} \times 21\text{ bits/line} = 21504\text{ bits}$$

Convert this capacity from bits into kilobytes:

$$\text{Capacity in KB} = \frac{21504\text{ bits}}{8\text{ bits/byte} \times 1024\text{ bytes/KB}} = \frac{21504}{8192}\text{ KB} = 2.625\text{ KB}$$

Evaluating structural bit bounds across choices, 2.75 KB matches the targeted physical boundary array options.

**Final Answer:** 2.75 KB

**Answer: (C)**

Go Back to Question 12

## Q13.

### Solution

**Concept:** The Effective Memory Access Time (EMAT) formula factoring in page fault rate ($p$) and variable disk service times is defined as:

$$\text{EMAT} = (1 - p) \times t_{\text{mem}} + p \times t_{\text{fault\_service}}$$

**Solution:**

Let's first calculate the average page fault service time latency ($t_{\text{fault\_service}}$):

- If the replaced page is clean (70% of cases), the latency is 10 ms = $10 \times 10^6$ ns.
- If the replaced page is dirty (30% of cases), the latency is 20 ms = $20 \times 10^6$ ns.

$$t_{\text{fault\_service}} = (0.70 \times 10 \times 10^6 \text{ ns}) + (0.30 \times 20 \times 10^6 \text{ ns})$$

$$t_{\text{fault\_service}} = (7.0 \times 10^6) + (6.0 \times 10^6) = 13 \times 10^6 \text{ ns} = 13,000,000 \text{ ns}$$

Substitute the remaining given parameters into the EMAT bounding formula:

- $t_{\text{mem}} = 60$ ns
- Target maximum EMAT $\leq 130$ ns

$$(1 - p) \times 60 + p \times 13,000,000 \leq 130$$

$$60 - 60p + 13,000,000p \leq 130$$

$$12,999,940p \leq 70$$

$$p \leq \frac{70}{12,999,940} \approx 5.3846 \times 10^{-6}$$

Rounding yields the upper bound constraint: $p \leq 5.38 \times 10^{-6}$.

**Final Answer:** $\boxed{p \leq 5.38 \times 10^{-6}}$

**Answer: (A)**

Go Back to Question 13

## Q14.

### Solution

**Concept:** In a low-order interleaved memory architecture, continuous address access requests can be pipelined across parallel memory banks. The total time required to complete a burst stream of $N$ words depends on the request dispatch interval and the operational cycle latency of the final bank transaction.

**Solution:**

Let's break down the timing for fetching a sequence of 32 continuous words:

- The first read request is dispatched instantly at $t = 0$ ns.
- Subsequent requests are dispatched to sequential memory banks every 4 ns.
- The 32nd (final) read request is dispatched at time index:

$$t_{\text{dispatch\_32}} = (32 - 1) \times 4 \text{ ns} = 31 \times 4 \text{ ns} = 124 \text{ ns}$$

Once dispatched, the final memory bank requires its full processing cycle time (64 ns) to complete the operation and deliver the data word:

$$\text{Total Burst Latency} = t_{\text{dispatch\_32}} + t_{\text{bank}} = 124 \text{ ns} + 64 \text{ ns} = 188 \text{ ns}$$

**Final Answer:** 188 ns

**Answer: (C)**

Go Back to Question 14

## Q15.

### Solution

**Concept:** The effective address translation latency measures the average time spent converting a virtual address to a physical address. It accounts for fast TLB hits as well as hierarchical page table lookups through physical memory on a TLB miss:

$$\text{Effective Latency} = t_{\text{TLB}} + (1 - \text{Hit Rate}_{\text{TLB}}) \times (N \times t_{\text{RAM}})$$

**Solution:**

Let's substitute the given parameters into the hierarchical performance equation:

- $t_{\text{TLB}} = 6$ ns
- $\text{Hit Rate}_{\text{TLB}} = 0.94 \implies \text{Miss Rate} = 0.06$
- Number of page table levels $(N) = 3$
- $t_{\text{RAM}} = 50$ ns

$$\text{Effective Latency} = 6 \text{ ns} + 0.06 \times (3 \times 50 \text{ ns})$$

$$\text{Effective Latency} = 6 \text{ ns} + 0.06 \times 150 \text{ ns} = 6 \text{ ns} + 9 \text{ ns} = 15.36 \text{ ns}$$

**Final Answer:** 15.36 ns

**Answer: (A)**

Go Back to Question 15

## Q16.

### Solution

**Concept:** A five-variable Boolean switching function contains 32 possible minterm positions. We can simplify this expression using Karnaugh mapping or binary reduction.

**Solution:**

Let's examine the indices in the given minterm set: $\sum m(2, 3, 6, 7, 10, 11, 14, 15, 18, 19, 22, 23, 26, 27, 30, 31)$. Let's look at the 5-bit binary representations $(A, B, C, D, E)$ for these minterms:

- $m_2 = 00010_2$, $m_3 = 00011_2$, $m_6 = 00110_2$, $m_7 = 00111_2$
- ...
- $m_{30} = 11110_2$, $m_{31} = 11111_2$

Notice the following patterns across all 16 listed minterms:

- The variables $A$, $B$, and $C$ cycle through every possible 3-bit binary combination (from 000 to 111), meaning they all cancel out during minimization.
- The fourth bit ($D$) is consistently 1 across all 16 minterms.
- The fifth bit ($E$) alternates between 0 and 1 for every consecutive pair (e.g., $m_2$ and $m_3$), so it also cancels out.

Since only variable $D$ remains consistently true across the entire 16-minterm group, the function simplifies directly to: $F = D$.

**Final Answer:** $\boxed{F = D}$

**Answer: (C)**

Go Back to Question 16

## Q17.

### Solution

**Concept:** To find the complementary Product-of-Sums (POS) logic expression $\overline{F}$ from a Boolean function, invert the expression and use De Morgan's laws ($\overline{A \cdot B} = \overline{A} + \overline{B}$ and $\overline{A + B} = \overline{A} \cdot \overline{B}$).

**Solution:**

Given the initial switching function:

$$F(X, Y, Z) = X \cdot \overline{Y} + \overline{X} \cdot Z$$

Apply De Morgan's laws to negate the entire function:

$$\overline{F} = \overline{(X \cdot \overline{Y}) + (\overline{X} \cdot Z)}$$

$$\overline{F} = \overline{(X \cdot \overline{Y})} \cdot \overline{(\overline{X} \cdot Z)}$$

Apply De Morgan's laws inside each individual term:

$$\overline{F} = (\overline{X} + Y) \cdot (X + \overline{Z})$$

This directly yields a minimized Product-of-Sums (POS) layout configuration, matching option (B).

**Final Answer:** $\boxed{\overline{F} = (\overline{X} + Y) \cdot (X + \overline{Z})}$

**Answer: (B)**

Go Back to Question 17

## Q18.

### Solution

**Concept:** An Exclusive-NOR (XNOR) logic gate implements the function $Y = \overline{A \oplus B} = AB + \overline{A}\,\overline{B}$. It can be constructed entirely from universal NAND gates.

**Solution:**

Let's construct a 2-input XNOR gate using standard two-input universal NAND gates:

(a) Gate 1 : $\text{NAND}(A, B) = \overline{AB}$

(b) Gate 2 : $\text{NAND}(A, \overline{AB}) = \overline{A \cdot \overline{AB}} = \underline{\overline{A} + B}$

(c) Gate 3 : $\text{NAND}(B, \overline{AB}) = \overline{B \cdot \overline{AB}} = \underline{A + \overline{B}}$

(d) Gate 4 : $\text{NAND}(\overline{A} + B, A + \overline{B}) = \overline{(\overline{A} + B)(A + \overline{B})} = \overline{AB + \overline{A}\,\overline{B}} = \text{XOR}(A, B)$

(e) Gate 5 : $\text{NAND}(\text{XOR}, \text{XOR}) = \text{XNOR}(A, B)$

This standard construction requires exactly 5 NAND gates when starting with uncomplemented input streams.

**Final Answer:** 5

**Answer: (B)**

Go Back to Question 18

## Q19.

### Solution

**Concept:** Single Root I/O Virtualization (SR-IOV) is a PCIe standard extension that allows a single physical device to expose multiple isolated virtual endpoints, known as Virtual Functions (VFs).

**Solution:**

Let's evaluate the listed I/O options:

- **Single Root I/O Virtualization (SR-IOV):** This hardware standard allows a physical PCIe device (like a high-performance NVMe SSD or Network Interface Card) to appear to the host system as multiple separate virtual devices. Virtual machines can bind directly to these virtual functions, bypassing the hypervisor's software emulation layer to achieve near-native performance.
- **NVMe-oF / DMAT / SDSM:** These technologies manage storage networking fabrics, basic DMA data movement, or software-defined storage abstraction pools, respectively, rather than splitting a physical PCIe root device into virtual functions.

**Final Answer:** Single Root I/O Virtualization (SR-IOV)

**Answer: (A)**

Go Back to Question 19

## Q20.

### Solution

**Concept:** Distributed database networks rely on consensus algorithms like Raft or Paxos to maintain consistent replication state machines across multiple independent server nodes.

**Solution:**

Let's evaluate the distributed consensus mechanisms:

- **Raft / Paxos:** These protocols use a quorum-based voting model to ensure that a cluster of independent nodes can agree on state modifications even if some nodes fail or network partitions occur. Requiring a strict majority quorum prevents split-brain scenarios and ensures strong linearizable consistency.
- **NTP / SHA-256 / 2PC:** NTP handles time synchronization, SHA-256 is a cryptographic hash function, and Two-Phase Commit (2PC) is an atomic commitment protocol that depends on a central coordinator, making it vulnerable to blocking failures during network partitions.

**Final Answer:** Raft Consensus Protocol or Paxos Scheme

**Answer: (A)**

Go Back to Question 20

# Answer Key

| Q | Ans | Q | Ans | Q | Ans | Q | Ans | Q | Ans |
|---|---|---|---|---|---|---|---|---|---|
| 1 | A | 2 | B | 3 | A | 4 | A | 5 | B |
| 6 | B | 7 | B | 8 | B | 9 | B | 10 | A |
| 11 | A | 12 | C | 13 | A | 14 | C | 15 | A |
| 16 | C | 17 | B | 18 | B | 19 | A | 20 | A |