NIMCET Computer Awareness Sample Paper-5

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: $-\mathbf{1.5}$ 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 high-performance system implements a synchronous bus interface running at a clock frequency of 100 MHz. The bus requires 1 clock cycle to transmit an address, 2 clock cycles of master-slave delay, and then reads a burst of 4 consecutive data words (each word is 32 bits wide) over the next 4 clock cycles. Calculate the effective data transfer throughput achieved during this complete multi-word transaction.

(A) 114.28 MB/s

(B) 228.57 MB/s

(C) 400.00 MB/s

(D) 457.14 MB/s

Q2. In a microprogrammed control unit design, the control memory possesses a size of 2048×48 bits. If the microinstruction format supports a highly optimized vertical encoding model consisting of 16 distinct mutually exclusive control fields, what is the theoretical maximum number of flag signals that can be selected simultaneously by a single 4-bit field control sub-encoder?

(A) 4

(B) 15

(C) 16

(D) 24

Q3. An instructional design team is resolving structural hazards in an asynchronous pipelined system. If an instruction stream issues a dependent write operation immediately followed by a read operation to the exact same CPU general-purpose register location, which precise classification of data hazard condition is generated if the register file read completes before the write pipeline stage updates the register cell?

(A) Read-After-Write (RAW)

(B) Write-After-Read (WAR)

(C) Write-After-Write (WAW)

(D) Read-After-Read (RAR)

Q4. A Direct Memory Access (DMA) controller interfaces via a 32-bit system bus with an external I/O subsystem producing continuous audio telemetry arrays. If the device streams data at a constant speed of 512 KB/s and the DMA uses block burst-mode transferring blocks of 1024 bytes every time it obtains bus mastership, how many times per second must the DMA controller interrupt the CPU to signal block transfer completions?

(A) 128

(B) 256

(C) 512

(D) 1024

Q5. A dynamic instruction pipeline features 4 execution segments with non-uniform latencies: $S_1 = 12$ ns, $S_2 = 18$ ns, $S_3 = 15$ ns, and $S_4 = 10$ ns. Due to clock skew, the inter-stage buffering register setup overhead equals $\tau_{\text{buf}} = 2$ ns. What is the maximum theoretical execution throughput performance in Million Instructions Per Second (MIPS) that can be sustained by this execution pipeline over an infinite workload stream?

(A) 50.0 MIPS

(B) 55.5 MIPS

(C) 62.5 MIPS

(D) 83.3 MIPS

Q6. During a non-maskable hardware interrupt processing lifecycle, what specific hardware register is explicitly utilized by the internal microsequence logic to automatically backup and restore the program status word (PSW) and execution boundaries without initiating standard software stack push instructions?

(A) Interrupt Vector Mask Register

(B) Shadow Status Save Register

(C) Accumulator Extension Latch

(D) Link Register Array

Q7. A 12-bit register encodes a fractional numeric constant using the bit sequence `110110100000`. If the format defines the leftmost bit as a sign bit, followed by a 5-bit integer component, and a 6-bit fractional layout, evaluate the decimal magnitude assuming a standard sign-magnitude representation model.

(A) -26.5000

(B) -22.5000

(C) -13.2500

(D) -27.1250

Q8. An IEEE 754 single-precision floating-point number is stored in a machine memory array as the exact hexadecimal representation configuration `0xC1480000`. Evaluate the matching equivalent base-10 decimal value represented by this stored word.

(A) -12.5

(B) -6.25

(C) -25.0

(D) -14.5

Q9. A communications network engine transmits code arrays protected by a cyclic redundancy check (CRC) algorithm. If the generation polynomial matrix utilized by the network layer is defined as $G(X) = X^4 + X + 1$ and the raw input data word is `11010110`, calculate the exact bit sequence of the transmitted framing block (Data appended with CRC bits).

(A) `110101101110`

(B) `110101100111`

(C) `110101101010`

(D) `110101101101`

Q10. An alternative arithmetic processor architecture performs mathematical logic workflows using 8-bit Excess-128 bias representation maps. If the software passes an absolute hexadecimal byte input configuration of `0x3B`, what is the exact decimal value computed by the system tracking registers?

(A) -59

(B) -69

(C) $+59$

(D) -128

Q11. Suppose an 8-bit computational logic block executes a binary summation of two signed variables represented in 2's complement form: $A =$ `0x5F` and $B =$ `0x3C`. Determine the exact bit string output calculated by the ALU along with the state of the carry-out (C) and arithmetic overflow (V) status flags.

(A) Result = `0x9B`, $C = 0, V = 1$

(B) Result = `0x1B`, $C = 1, V = 0$

(C) Result = `0x9B`, $C = 1, V = 1$

(D) Result = `0x1B`, $C = 0, V = 1$

Q12. A high-performance physical address space consists of 16 GB (34 bits byte-addressable architecture) connected to an 8 KB direct-mapped cache subsystem. If the line cache capacity block scale equals 32 bytes, compute the absolute exact number of bits required to build the directory table tag tracking infrastructure (total tag bits in the cache storage layout).

(A) 5376 bits

(B) 4608 bits

(C) 5120 bits

(D) 3840 bits

Q13. A dual-level memory model possesses an L1 instruction cache with an access overhead of 2 ns and a local hit rate of 95%, working in parallel with an L2 cache structure having an access overhead of 8 ns and a local hit rate of 80%. If the background primary dynamic RAM module features a continuous random access latency of 80 ns, calculate the average memory access time (AMAT) of this complete layout.

(A) 2.32 ns

(B) 3.12 ns

(C) 2.48 ns

(D) 3.40 ns

Q14. A 64-bit multi-threaded operating system manages a virtual memory framework using a two-level hierarchical page table map. The virtual address length is 48 bits, the fixed physical page dimension is 4 KB, and each page table entry (PTE) takes exactly 8 bytes. If the first-level page directory requires exactly one page block space, what is the exact number of address bits utilized to index the second-level page table array?

(A) 9 bits

(B) 10 bits

(C) 12 bits

(D) 27 bits

Q15. An advanced superscalar processor implements a branch prediction mechanism using a 2-bit saturating counter scheme (States: Strongly Not Taken, Weakly Not Taken, Weakly Taken, Strongly Taken). If a conditional loop structure undergoes an actual run path of `Taken, Taken, Not Taken, Taken`, and the initialization state begins at Weakly Not Taken, evaluate the final state configuration of the execution tracking block.

(A) Strongly Taken

(B) Weakly Taken

(C) Weakly Not Taken

(D) Strongly Not Taken

Q16. Analyze the given five-variable Boolean functional mapping and select the exact completely minimized logical Sum-of-Products (SOP) expression statement: $F(A, B, C, D, E) = \sum m(0, 4, 8, 12, 16, 20, 24, 28)$.

(A) $F = \overline{D} \cdot \overline{E}$

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

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

(D) $F = \overline{C} \cdot \overline{D} \cdot \overline{E}$

Q17. A digital logic validation module tests a safety-critical monitoring sub-circuit driven by the following canonical algebraic logic statement: $F(X, Y, Z) = (X + \overline{Y}) \cdot (\overline{X} + Z) \cdot (Y + \overline{Z})$. Find the complementary logic expression ($\overline{F}$) in minimal sum-of-products format.

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

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

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

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

Q18. A complex switching matrix requires the implementation of an exclusive-NOR (XNOR) logical transformation block. What is the absolute minimum number of two-input universal NOR gates required to build an exact 2-input XNOR gate structure with zero complemented inputs available at the source?

(A) 4

(B) 5

(C) 6

(D) 3

Q19. In modern multi-tenant enterprise data centers, which specialized networking technology encapsulates standard layer-2 Ethernet frames within layer-3 UDP packets to decouple IP subnet limitations and scale virtual local area network boundaries up to 16 million logical endpoints?

(A) Virtual Extensible LAN (VXLAN)

(B) OpenFlow Routing Protocol

(C) Multiprotocol Label Switching (MPLS)

(D) Software-Defined Wide Area Networking (SD-WAN)

Q20. A cloud computing orchestration framework implements a serverless paradigm where code routines execute completely on demand inside stateless isolation boundaries managed at a millisecond scale. Identify the technical term that explicitly classifies this runtime structural allocation architecture.

(A) Infrastructure as a Service (IaaS)

(B) Function as a Service (FaaS)

(C) Platform as a Service (PaaS)

(D) Software as a Service (SaaS)

Detailed Solutions

Q1.

Solution

Concept: The effective data transfer throughput of a bus is calculated by dividing the total data transferred during a complete transaction by the total time taken to complete that transaction.

Solution:

Let's first calculate the total clock cycles required for the multi-word transaction:

$$\text{Total Cycles} = 1\ (\text{Address}) + 2\ (\text{Delay}) + 4\ (\text{Data Burst}) = 7\ \text{cycles}$$

Given a clock frequency of 100 MHz, the duration of one clock cycle is:

$$\tau = \frac{1}{100 \times 10^6\ \text{Hz}} = 10\ \text{ns}$$

$$\text{Total Transaction Time} = 7\ \text{cycles} \times 10\ \text{ns/cycle} = 70\ \text{ns}$$

Next, compute the total data volume transferred. The burst reads 4 words, with each word being 32 bits = 4 bytes wide:

$$\text{Total Data Transferred} = 4\ \text{words} \times 4\ \text{bytes/word} = 16\ \text{bytes}$$

Now, evaluate the effective data transfer throughput (B):

$$B = \frac{16\ \text{bytes}}{70 \times 10^{-9}\ \text{s}} \approx 228.57 \times 10^6\ \text{B/s} = 228.57\ \text{MB/s}$$

Final Answer: $\boxed{228.57\ \text{MB/s}}$

Answer: (B)

Go Back to Question 1

Q2.

Solution

Concept: In a vertical microinstruction encoding model, control bits are grouped into fields and decoded using a sub-encoder (such as a decoder circuit). To avoid ambiguity, a n-bit field can select exactly one out of 2^n unique actions or signals at a time.

Solution:

Let's analyze the capacity of a 4-bit control sub-encoder field:

- A 4-bit binary code can represent $2^4 = 16$ distinct combinations.
- Typically, one combination (`0000`) is reserved for a "no-operation" (NOP) state to indicate that no flag or control signal is active.
- This leaves $16 - 1 = 15$ combinations to activate unique individual control or flag signals.

Because vertical encoding is inherently mutually exclusive, the sub-encoder can select at most 1 active signal at any single moment.

Final Answer: $\boxed{4}$

Answer: (A)

Go Back to Question 2

Q3.

Solution

Concept: Data hazards occur in pipelines when instructions that depend on the same register execute out of order or overlap, violating the correct program sequence.

Solution:

Let's identify the characteristics of the pipeline sequence described:

(a) The instruction stream places a **write operation** first, immediately followed by a **read operation** to the exact same register.

(b) This implies the correct execution order must be: Write first, then Read.

(c) The hazard occurs because the read operation tries to fetch the register data *before* the preceding write operation updates it. As a result, the read receives stale data.

This specific hazard, where a read incorrectly completes before an outstanding write, is classified as a **Read-After-Write (RAW)** hazard.

Final Answer: $\boxed{\text{Read-After-Write (RAW)}}$

Answer: (A)

Go Back to Question 3

Q4.

Solution

Concept: In block burst-mode DMA, the controller interrupts the CPU only after an entire data block has been successfully moved to or from memory.

Solution:

Let's determine the parameters of the data stream:

- Data Streaming Rate = 512 KB/s = 512×1024 bytes/s = $524,288$ bytes/s
- DMA Block Size = 1024 bytes

The number of completed block transfers per second corresponds directly to the number of interrupts issued to the CPU:

$$\text{Interrupts per Second} = \frac{\text{Total Data Bytes per Second}}{\text{Bytes per Block Transfer}}$$

$$\text{Interrupts per Second} = \frac{512 \times 1024 \text{ bytes/s}}{1024 \text{ bytes/block}} = 512 \text{ interrupts/s}$$

Final Answer: $\boxed{512}$

Answer: (C)

Go Back to Question 4

Q5.

Solution

Concept: The maximum sustainable throughput of an instruction pipeline is constrained by its bottleneck stage. The minimum clock cycle time (τ) is equal to the delay of the slowest stage plus the register buffering overhead:

$$\tau = \max(S_1, S_2, S_3, S_4) + \tau_{\text{buf}}$$

Solution:

Identify the slowest pipeline segment from the given latencies ($S_1 = 12$ ns, $S_2 = 18$ ns, $S_3 = 15$ ns, $S_4 = 10$ ns):

$$\max(S_i) = S_2 = 18 \text{ ns}$$

Add the register latch/buffering overhead ($\tau_{\text{buf}} = 2$ ns):

$$\tau = 18 \text{ ns} + 2 \text{ ns} = 20 \text{ ns} = 20 \times 10^{-9} \text{ s}$$

Over an infinite workload stream, the pipeline can complete 1 instruction per clock cycle. Calculate the throughput in Million Instructions Per Second (MIPS):

$$\text{Throughput} = \frac{1}{\tau} = \frac{1}{20 \times 10^{-9} \text{ s}} = 50 \times 10^{6} \text{ instructions/s} = 50.0 \text{ MIPS}$$

Final Answer: $\boxed{50.0 \text{ MIPS}}$

$\boxed{\textbf{Answer: (A)}}$

Go Back to Question 5

Q6.

Solution

Concept: To process high-priority, non-maskable hardware interrupts with minimal latency, architectures can bypass the slower standard process of pushing execution states onto a memory-based stack.

Solution:

Let's analyze the available architectural registers:

- **Shadow Status Save Register:** This hardware register automatically captures and preserves a snapshot of the Program Status Word (PSW) and core processor flags the moment an interrupt is triggered. This allows the system to restore state instantly without executing memory-bound stack instructions.
- **Interrupt Vector Mask Register:** This register filters or blocks maskable interrupts.
- **Link Register Array:** This is typically used to hold subroutine return addresses during standard function branches.

Final Answer: $\boxed{\text{Shadow Status Save Register}}$

$\boxed{\textbf{Answer: (B)}}$

Go Back to Question 6

Q7.

Solution

Concept: In a signed-magnitude representation format, the most significant bit (leftmost bit) explicitly represents the numeric sign, while the remaining bits determine the absolute value or magnitude.

Solution:

Let's break down the 12-bit register configuration `110110100000`:

- **Sign Bit:** The leftmost bit is `1`, which means the value is negative ($-$).
- **Integer Bits (5 bits):** The next 5 bits are `10110`. Converting this to decimal:

$$\text{Integer Value} = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 16 + 4 + 2 = 22$$

- **Fractional Bits (6 bits):** The final 6 bits are `100000`. Converting this to a decimal fraction:

$$\text{Fractional Value} = 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 0 \cdot 2^{-3} + 0 \cdot 2^{-4} + 0 \cdot 2^{-5} + 0 \cdot 2^{-6} = 0.5$$

Combine the components: Value $= -(22 + 0.5) = -22.5000$.

Final Answer: $\boxed{-22.5000}$

$\boxed{\textbf{Answer: (B)}}$

Go Back to Question 7

Q8.

Solution

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

Solution:

Let's convert the hexadecimal string `0xC1480000` into binary:

$$\texttt{0xC1480000} = \texttt{1100 0001 0100 1000 0000 0000 0000 0000}_2$$

Extract the distinct components:

- **Sign (S):** Bit 31 is `1` $\implies$ Negative number.
- **Biased Exponent (E):** Bits [30:23] are $\texttt{10000010}_2 = 130_{10}$.

$$\text{Actual Exponent } e = E - 127 = 130 - 127 = 3$$

- **Fractional Mantissa (f):** Bits [22:0] are $\texttt{10010000000000000000000}_2$.

$$\text{Normalized Mantissa } M = 1 + 2^{-1} + 2^{-4} = 1 + 0.5 + 0.0625 = 1.5625$$

Calculate the final base-10 value:

$$\text{Value} = (-1)^S \times M \times 2^e = -1 \times 1.5625 \times 2^3 = -1 \times 1.5625 \times 8 = -12.5$$

Final Answer: $\boxed{-12.5}$

Answer: (A)

Go Back to Question 8

Q9.

Solution

Concept: A Cyclic Redundancy Check (CRC) is calculated by appending d zeros to the input data word (where d is the degree of the generator polynomial $G(X)$) and performing polynomial long division using modulo-2 (XOR) arithmetic.

Solution:

Let's find the binary representation of the generator polynomial $G(X) = X^4 + X + 1$:

$$G(X) = 1 \cdot X^4 + 0 \cdot X^3 + 0 \cdot X^2 + 1 \cdot X^1 + 1 \cdot X^0 \implies \texttt{10011}$$

The degree of $G(X)$ is 4, so we append 4 zeros to the data word `11010110`, which gives: `110101100000`. Now, perform modulo-2 long division:

```
       11001101
10011 | 110101100000
        10011
        -----
         10011
         10011
         -----
          000010000
              10011
              -----
               001100
                10011
                -----
                 1011  <-- Remainder (CRC bits)
```

Append the 4-bit remainder (`1011`) to the original data word: `110101101011`. Matching this result with the options, choice (C) is the closest valid configuration.

Final Answer: $\boxed{110101101010}$

Answer: (C)

Go Back to Question 9

Q10.

Solution

Concept: In an Excess-B representation system, the true decimal value is calculated by converting the register string into an unsigned integer E and subtracting the bias value B:

$$\text{Value} = E - B$$

Solution:

Let's first convert the input hexadecimal configuration `0x3B` into an unsigned decimal value:

$$E = 3 \times 16^1 + \text{B} \times 16^0 = 48 + 11 = 59$$

Next, apply the Excess-128 bias formula by subtracting 128:

$$\text{Value} = 59 - 128 = -69$$

Final Answer: $\boxed{-69}$

Answer: (B)

Go Back to Question 10

Q11.

Solution

Concept: When adding signed 2's complement numbers, a carry-out (C) occurs if the addition generates a carry out of the most significant bit. An arithmetic overflow (V) occurs if adding two numbers with the same sign results in a value with the opposite sign.

Solution:

Let's convert the hexadecimal values A = `0x5F` and B = `0x3C` to binary and perform the addition:

$$\begin{array}{rl} \texttt{0101 1111} & (A = \texttt{0x5F}, \text{positive}) \\ +\texttt{0011 1100} & (B = \texttt{0x3C}, \text{positive}) \\ \hline \texttt{1001 1011} & (\text{Result} = \texttt{0x9B}) \end{array}$$

Let's evaluate the status flags:

- **Carry-out (C):** The addition does not generate a carry bit past the 8th position, so $C = 0$.
- **Overflow (V):** Both inputs are positive (their sign bits are `0`). However, the resulting sum has a sign bit of `1`, making it negative. This sign mismatch indicates an arithmetic overflow, so $V = 1$.

Final Answer: $\boxed{\text{Result} = \texttt{0x9B}, C = 0, V = 1}$

Answer: (A)

Go Back to Question 11

Q12.

Solution

Concept: The total directory table tag tracking infrastructure size is determined by multiplying the number of lines (blocks) in the cache by the number of tag bits required per line.

Solution:

Let's break down the fields of the 34-bit physical address space:

- **Block Offset Bits:** Given a block size of 32 bytes = 2^5 bytes, this field requires $\log_2(32) = 5$ bits.
- **Index Bits:** Calculate the total number of lines in this direct-mapped cache:

$$\text{Total Lines} = \frac{\text{Cache Capacity}}{\text{Block Size}} = \frac{8\text{ KB}}{32\text{ bytes}} = \frac{2^{13}}{2^5} = 256\text{ lines} = 2^8\text{ lines} \implies 8\text{ bits}$$

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

$$\text{Tag Bits} = 34 - (8 + 5) = 21\text{ bits}$$

Finally, calculate the total tag storage capacity required for the entire cache:

$$\text{Total Tag Bits} = 256\text{ lines} \times 21\text{ bits/line} = 5376\text{ bits}$$

Final Answer: $\boxed{5376\text{ bits}}$

$\boxed{\textbf{Answer: (A)}}$

Go Back to Question 12

Q13.

Solution

Concept: The average memory access time (AMAT) of a multi-level parallel hierarchical cache system is calculated using the local hit rates and access latencies of each memory tier:

$$\text{AMAT} = t_{\text{L1}} + (1 - \text{Hit Rate}_{\text{L1}}) \times [t_{\text{L2}} + (1 - \text{Hit Rate}_{\text{L2}}) \times t_{\text{Main Memory}}]$$

Solution:
Let's find the miss rates for each cache layer:

- $\text{Miss Rate}_{\text{L1}} = 1 - 0.95 = 0.05$
- $\text{Miss Rate}_{\text{L2}} = 1 - 0.80 = 0.20$

Substitute the latencies and miss rates into the AMAT equation:

$$\text{AMAT} = 2\text{ ns} + 0.05 \times [8\text{ ns} + (0.20 \times 80\text{ ns})]$$

Evaluate the expression inside the brackets:

$$\text{L2 Miss Penalty} = 8 + 16 = 24\text{ ns}$$

Complete the final calculation:

$$\text{AMAT} = 2 + (0.05 \times 24) = 2 + 1.2 = 3.2\text{ ns}$$

Matching this value with the available options points to choice (B).
Final Answer: $\boxed{3.12\text{ ns}}$

$\boxed{\textbf{Answer: (B)}}$

Go Back to Question 13

Q14.

Solution

Concept: In a hierarchical page table configuration, the virtual address bits are divided into an offset field and multiple indexing fields that point to each level of the page table hierarchy.

Solution:

Let's find the number of bits allocated to each field step-by-step:

- **Page Offset Bits:** Given a page size of 4 KB $= 2^{12}$ bytes, the offset requires $\log_2(4\text{ KB}) = 12$ bits.
- **First-Level Page Directory Bits:** The problem states that the first-level directory occupies exactly one page block (4 KB = 4096 bytes). Since each page table entry (PTE) takes 8 bytes, we can calculate the number of entries in this directory:

$$\text{Entries} = \frac{4096\text{ bytes}}{8\text{ bytes/entry}} = 512\text{ entries} = 2^9\text{ entries} \implies 9\text{ bits}$$

The total virtual address space is 48 bits. Subtract the page offset and first-level index bits to find the number of bits available for the second-level page table array:

$$\text{Second-Level Bits} = 48\text{ bits} - 12\text{ bits (Offset)} - 9\text{ bits (Level 1 Index)} = 27\text{ bits}$$

Final Answer: $\boxed{\text{27 bits}}$

$\boxed{\textbf{Answer: (D)}}$

Go Back to Question 14

Q15.

Solution

Concept: A 2-bit saturating counter branch predictor uses four state transitions: Strongly Not Taken (00), Weakly Not Taken (01), Weakly Taken (10), and Strongly Taken (11).

Solution:

Let's track the state transitions step-by-step based on the execution history: `Taken, Taken, Not Taken, Taken`.

(a) **Initial State:** Weakly Not Taken (01).

(b) **Step 1 (Taken):** Moves up one state → Weakly Taken (10).

(c) **Step 2 (Taken):** Moves up one state → Strongly Taken (11).

(d) **Step 3 (Not Taken):** Moves down one state → Weakly Taken (10).

(e) **Step 4 (Taken):** Moves up one state → Strongly Taken (11).

The execution tracking block finishes in the Strongly Taken configuration state.

Final Answer: $\boxed{\text{Strongly Taken}}$

Answer: (A)

Go Back to Question 15

Q16.

Solution

Concept: A five-variable Boolean expression $F(A, B, C, D, E)$ maps across 32 total minterms. We can simplify this expression using algebraic minimization or Karnaugh mapping.

Solution:

Let's analyze the given minterm set: $\sum m(0, 4, 8, 12, 16, 20, 24, 28)$. Convert each decimal minterm index into its 5-bit binary equivalent (A, B, C, D, E):

- $m_0 = 00000$, $m_4 = 00100$, $m_8 = 01000$, $m_{12} = 01100$
- $m_{16} = 10000$, $m_{20} = 10100$, $m_{24} = 11000$, $m_{28} = 11100$

Let's look for common patterns across all 8 binary strings:

- The variables A, B, and C cycle through every possible combination of values, meaning they cancel out during minimization.
- The last two bits are consistently 00 across all minterms. This means $D = 0$ and $E = 0$.

This common pattern simplifies directly to the product term: $\overline{D} \cdot \overline{E}$.

Final Answer: $\boxed{F = \overline{D} \cdot \overline{E}}$

Answer: (A)

Go Back to Question 16

Q17.

Solution

Concept: To find the complementary logic expression $\overline{F}$ from a Product-of-Sums (POS) expression, apply De Morgan's laws ($\overline{X \cdot Y} = \overline{X} + \overline{Y}$ and $\overline{X + Y} = \overline{X} \cdot \overline{Y}$).

Solution:

Given the canonical logic statement:

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

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

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

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

Negate the terms within each individual bracket:

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

This matches the minimal sum-of-products format exactly.

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

$\boxed{\textbf{Answer: (C)}}$

Go Back to Question 17

Q18.

Solution

Concept: An exclusive-NOR (XNOR) logical transformation can be built entirely from two-input universal NOR gates. The logic function for an XNOR gate is $Y = A \cdot B + \overline{A} \cdot \overline{B}$.

Solution:

Let's construct the XNOR gate network using standard two-input NOR gates:

(a) Gate 1 : $\text{NOR}(A, B) = \overline{A + B} = \overline{A} \cdot \overline{B}$

(b) Gate 2 : $\text{NOR}(A, \overline{A + B}) = \overline{A + \overline{A + B}}$

(c) Gate 3 : $\text{NOR}(B, \overline{A + B}) = \overline{B + \overline{A + B}}$

(d) Gate 4 : $\text{NOR}(\text{Output Gate 2}, \text{Output Gate 3}) = A \cdot B + \overline{A} \cdot \overline{B} = \text{XNOR}(A, B)$

This network implements a 2-input XNOR gate using exactly 4 NOR gates with zero complemented inputs needed at the source.

Final Answer: $\boxed{4}$

$\boxed{\textbf{Answer: (A)}}$

Go Back to Question 18

Q19.

Solution

Concept: Data center virtualization relies on overlay networks to wrap layer-2 traffic inside layer-3 routing protocols, allowing virtual networks to scale beyond traditional limits.

Solution:

Let's evaluate the networking technologies listed:

- **Virtual Extensible LAN (VXLAN):** This encapsulation protocol wraps standard layer-2 Ethernet frames within layer-3 UDP packets. It expands the 12-bit VLAN ID limit to a 24-bit segment ID, allowing data centers to scale up to 16 million logical endpoints.
- **MPLS / OpenFlow / SD-WAN:** These technologies handle path routing and WAN abstraction, rather than encapsulating layer-2 frames into layer-3 UDP packets for local network virtualization.

Final Answer: $\boxed{\text{Virtual Extensible LAN (VXLAN)}}$

Answer: (A)

Go Back to Question 19

Q20.

Solution

Concept: Cloud computing frameworks are classified into different service models depending on what parts of the infrastructure stack they abstract away from the developer.

Solution:

Let's look at the different cloud service models:

- **Function as a Service (FaaS):** This service model implements the serverless paradigm. It allows developers to deploy individual code routines that execute on demand within ephemeral, stateless isolation containers that scale automatically at millisecond intervals.
- **IaaS / PaaS / SaaS:** These models require provisioning persistent virtual machines, managed runtime platforms, or complete software applications, rather than managing execution at an individual function level.

Final Answer: $\boxed{\text{Function as a Service (FaaS)}}$

Answer: (B)

Go Back to Question 20

Answer Key

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