NIMCET Computer Awareness Sample Paper-12

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. Which of the following memory technologies requires a periodic electrical refresh cycle to maintain its stored data bits, making its internal circuitry highly dense but slightly slower?

(A) Flash Memory

(B) SRAM

(C) DRAM

(D) EEPROM

Q2. In a CPU pipeline architecture, a structural hazard occurs when:

(A) A conditional branch instruction changes the flow of control, invalidating pre-fetched instructions.

(B) Two or more concurrent instructions require the use of the same physical hardware resource.

(C) An instruction depends on the result of a previous instruction that has not yet completed execution.

(D) The memory management unit experiences a page fault during data operand access.

Q3. When adding two 8-bit signed numbers represented in 2's complement form, an arithmetic overflow condition can be accurately detected when:

(A) The carry-out from the most significant bit position is equal to 1.

(B) The carry-in into the sign bit position does not equal the carry-out from the sign bit position.

(C) The operands have opposite signs and the result matches the sign of the smaller operand.

(D) The carry-in into the sign bit position is 1 regardless of the carry-out.

Q4. Which of the following statements is TRUE regarding the microprogrammed control unit compared to a hardwired control unit?

(A) It is faster in execution speed and simpler to design.

(B) It utilizes combinational logic circuits exclusively to generate control signals.

(C) It is slower in execution speed but offers higher flexibility for instruction set updates.

(D) It requires less control memory space but is harder to optimize.

Q5. How many 2-input NAND gates are minimally required to implement a standard 2-to-1 multiplexer circuit?

(A) 4

(B) 6

(C) 3

(D) 5

Q6. An 8-bit registers stores a signed integer in 2's complement representation. If the hexadecimal value stored in the register is 0x9C, what is its corresponding decimal value?

(A) -60

(B) -28

(C) -100

(D) -99

Q7. In the context of the Von Neumann architecture, the bottleneck that limits the processing throughput is primarily caused by:

(A) The volatile nature of the primary storage unit.

(B) The lack of registers within the Arithmetic Logic Unit.

(C) The shared bus structure for both instructions and data causing sequential access delays.

(D) The mismatch between disk rotation speed and system bus frequency.

Q8. The main advantage of using Direct Memory Access (DMA) over interrupt-driven I/O architectures is that:

(A) It performs data parity checking automatically without operating system intervention.

(B) It eliminates the need for any internal system bus structure.

(C) It executes internal ALU instructions faster during the transfer phase.

(D) The CPU is completely freed from managing high-speed data transfers after the initial setup.

Q9. Consider a 4-way set-associative cache memory with a total capacity of 64 KB and a block size of 64 bytes. If the main memory address space is 32-bit byte-addressable, how many bits are used for the 'Set Index' field?

(A) 8 bits

(B) 12 bits

(C) 6 bits

(D) 10 bits

Q10. During the execution of a machine instruction, which register in the Control Unit of the CPU holds the address of the next instruction to be fetched from memory?

(A) Program Counter (PC)

(B) Memory Address Register (MAR)

(C) Accumulator (AC)

(D) Instruction Register (IR)

Q11. Which of the following architectural features of a computer system minimizes the performance gap between the fast processor and relatively slow main memory by utilizing spatial and temporal locality?

(A) Direct Memory Access (DMA) channel

(B) Cache memory alignment

(C) Instruction pre-fetch buffer

(D) Pipelining unit

Q12. Which of the following binary representations accurately corresponds to the decimal fraction 0.4375?

(A) $(0.1011)_2$

(B) $(0.0101)_2$

(C) $(0.0111)_2$

(D) $(0.1101)_2$

Q13. The Boolean expression $X \oplus Y \oplus (X \cdot Y)$ can be simplified to which of the following standard logic forms?

(A) $X \oplus Y$

(B) $X \cdot Y$

(C) $X'Y'$

(D) $X + Y$

Q14. In the IEEE 754 single-precision floating-point format, a 32-bit value consists of 1 sign bit, 8 biased exponent bits, and 23 mantissa bits. If the biased exponent field contains the binary value 10000001, what is the actual unbiased exponent value in decimal?

(A) +1

(B) -1

(C) +2

(D) +129

Q15. Perform the subtraction $(45)_{10} - (23)_{10}$ using 8-bit 2's complement binary arithmetic. Which of the following represents the correct 8-bit binary output?

(A) 00010111

(B) 00010101

(C) 00010110

(D) 11101010

Q16. In a computer system, the system bus is typically divided into data, address, and control lines. If a processor has a 32-bit address bus and a 64-bit data bus, what is the maximum byte-addressable memory capacity it can directly reference?

(A) 8 GB

(B) 64 GB

(C) 16 GB

(D) 4 GB

Q17. Which of the following public-key cryptographic algorithms relies explicitly on the mathematical difficulty of factoring large prime numbers for its security infrastructure?

(A) AES

(B) RSA

(C) DES

(D) ECC

Q18. Convert the hexadecimal number $(A4C.8)_{16}$ into its equivalent octal representation.

(A) $(5124.4)_8$

(B) $(5114.2)_8$

(C) $(2444.4)_8$

(D) $(5114.4)_8$

Q19. Simplify the Boolean function $F(A, B, C) = \Sigma m(0, 2, 4, 5, 6)$. Which of the following represents the minimal Sum-of-Products (SOP) expression?

(A) $C' + AB'$

(B) $A'C' + BC$

(C) $C' + AB$

(D) $AC' + B'$

Q20. The purpose of the Translation Lookaside Buffer (TLB) in a modern CPU virtual memory architecture is to:

(A) Cache recent virtual-to-physical address translations to accelerate page table lookups.

(B) Synchronize data transfers between the L1 cache and the L2 cache layers.

(C) Provide a temporary storage area for instructions before they are decoded.

(D) Manage the priority queues for external hardware interrupts.

Detailed Solutions

Q1.

Solution

Concept: The question tests knowledge of different types of semiconductor memory. Dynamic Random Access Memory (DRAM) stores each bit of data in a separate passive configuration consisting of a microscopic capacitor and a transistor. Because capacitors leak charge over time, the data fades rapidly unless the capacitor charge is periodically refreshed by reading and rewriting every cell.

Solution:

(a) DRAM utilizes a single transistor and a single capacitor for each bit of storage, which allows for exceptionally high structural density on a silicon chip.

(b) Due to the natural leakage currents in the sub-micron silicon substrate, the stored charge on these tiny capacitors degrades within milliseconds.

(c) To preserve the stored information, a dedicated memory controller must perform periodic refresh operations, reading out the contents of each row and rewriting them before data corruption occurs.

(d) This refresh overhead cycle temporarily blocks processor access requests, making DRAM slightly slower than Static RAM (SRAM), which uses a 6-transistor latch configuration that retains data without refreshing as long as power is continuously applied.

(e) Flash memory and EEPROM are non-volatile storage technologies that utilize floating-gate transistors, which do not require refreshing to maintain state.

Final Answer: The memory technology is DRAM.

Answer: (C)

Go Back to Question 1

Q2.

Solution

Concept: Pipeline hazards are conditions in instruction-level pipelining that prevent the next instruction in the instruction stream from executing in its designated clock cycle. Hazards are categorized into structural, data, and control hazards. A structural hazard arises explicitly due to physical hardware resource conflicts.

Solution:

(a) A structural hazard occurs when the underlying hardware architecture lacks sufficient execution units, buses, or memory ports to support the concurrent execution of different pipeline stages.

(b) A classic example occurs when a processor features a unified memory architecture for both instructions and data. If one instruction is in the Fetch stage while a previous instruction is in the Memory Access stage, both attempt to access the unified cache simultaneously over a single memory port.

(c) This resource limitation forces the pipeline control logic to insert a stall cycle (bubble) to resolve the execution conflict.

(d) Option (A) describes a control hazard, which is triggered by branch instructions that alter the sequential flow of execution.

(e) Option (C) describes a data hazard, where an instruction requires data that has not yet been computed or written back by an earlier instruction in the pipeline.

Final Answer: Two or more concurrent instructions require the use of the same physical hardware resource.

Answer: (B)

Go Back to Question 2

Q3.

Solution

Concept: In signed 2's complement arithmetic, fixed-width registers have a bounded range. For an 8-bit signed integer, the representable range is from -128 to $+127$. Arithmetic overflow happens when the exact mathematical result of an addition falls entirely outside this valid representable interval.

Solution:

(a) Overflow can only occur when adding two numbers of the same sign. Adding a positive number to a negative number can never cause an overflow because the absolute value of the sum is always smaller than at least one of the operands.

(b) When adding two positive numbers, an overflow results in an incorrect negative value. When adding two negative numbers, an overflow results in an incorrect positive value.

(c) Digitally, this condition is monitored by checking the carry values at the Most Significant Bit (MSB) position, which functions as the sign bit.

(d) Let C_{in} be the internal carry-in bit into the MSB position, and C_{out} be the external carry-out bit from the MSB position.

(e) An overflow condition is mathematically identical to the logic expression $V = C_{in} \oplus C_{out}$. If C_{in} does not equal C_{out}, the result is invalid due to an arithmetic overflow.

Final Answer: The carry-in into the sign bit position does not equal the carry-out from the sign bit position.

Answer: (B)

Go Back to Question 3

Q4.

Solution

Concept: The Control Unit (CU) directs the operations of the processor. It can be implemented using two completely distinct methodologies: Hardwired control (built using fixed combinational logic circuits) or Microprogrammed control (built using microinstructions stored in a dedicated internal control memory).

Solution:

(a) A hardwired control unit uses state machines, decoders, and logic gates to generate control signals. Because the signals travel entirely through direct logic gates, execution is highly optimized and exceptionally fast, but modifying the instruction set requires a complete physical redesign.

(b) A microprogrammed control unit translates a machine instruction into a sequence of microinstructions stored in an internal Control Read-Only Memory (CROM).

(c) Fetching control words sequentially from an internal memory introduces a clock delay, which makes microprogrammed control units slower than hardwired implementations.

(d) However, microprogrammed architectures offer high flexibility. Upgrading or modifying the instruction set can be accomplished simply by rewriting the microcode within the internal control memory.

(e) Thus, microprogrammed units provide higher flexibility at the cost of execution speed.

Final Answer: It is slower in execution speed but offers higher flexibility for instruction set updates.

Answer: (C)

Go Back to Question 4

Q5.

Solution

Concept: A standard 2-to-1 multiplexer selects between two input lines I_0 and I_1 based on the value of a single select line S. The boolean expression representing this combinational function is given by the SOP equation $Y = S'I_0 + SI_1$. This function can be completely implemented using universal NAND logic.

Solution:

(a) To convert the expression $Y = S'I_0 + SI_1$ into NAND-only logic, we apply the double involution principle: $Y = ((S'I_0 + SI_1)')'$.

(b) Applying De Morgan's theorem to the inner expression yields the equivalent formulation: $Y = ((S'I_0)' \cdot (SI_1)')'$.

(c) We can break this down into individual NAND operations. First, we need S', which can be produced by a 2-input NAND gate acting as an inverter: $G_1 = (S \cdot S)' = S'$.

(d) Next, we generate the intermediate product terms using two separate 2-input NAND gates: $G_2 = (G_1 \cdot I_0)' = (S'I_0)'$ and $G_3 = (S \cdot I_1)' = (SI_1)'$.

(e) Finally, these two terms are combined using a fourth 2-input NAND gate: $G_4 = (G_2 \cdot G_3)' = ((S'I_0)' \cdot (SI_1)')' = S'I_0 + SI_1$. This requires exactly 4 NAND gates.

Final Answer: 4

Answer: (A)

Go Back to Question 5

Q6.

Solution

Concept: An 8-bit register storing a signed integer in 2's complement uses the most significant bit (bit 7) as a sign indicator. If the MSB is 0, the number is positive and can be read directly. If the MSB is 1, the number is negative, and its magnitude is found by computing its 2's complement.

Solution:

(a) The given value is hexadecimal $0x9C$. Converting this value into an 8-bit binary pattern yields: $9 = 1001$ and $C = 1100$, giving the complete bit sequence $(10011100)_2$.

(b) The leftmost bit (MSB) is 1, which clearly establishes that the stored integer is a negative value.

(c) To calculate its absolute decimal magnitude, we must find the 2's complement of the binary string $(10011100)_2$.

(d) Step 1: Perform bitwise inversion (1's complement), which transforms the sequence into $(01100011)_2$.

(e) Step 2: Add 1 to the least significant bit of the inverted sequence: $01100011 + 1 = (01100100)_2$.

(f) Converting the resulting magnitude $(01100100)_2$ into decimal gives $2^6 + 2^5 + 2^2 = 64 + 32 + 4 = 100$. Appending the negative sign gives -100.

Final Answer: -100

Answer: (C)

Go Back to Question 6

Q7.

Solution

Concept: The classic Von Neumann architecture relies on a design where the Central Processing Unit (CPU) and memory are separate entities. Crucially, a single, shared system bus is utilized to transmit both instruction opcodes and data operands between the memory unit and the CPU.

Solution:

(a) Because instructions and data must share the exact same physical bus infrastructure, the CPU cannot read an instruction and read/write data at the same exact instant in time.

(b) This structure forces a strictly sequential access pattern: the CPU must first fetch an instruction, decode it, and then execute it, which often involves a separate memory access for data operands.

(c) The throughput of the processor is bounded because the execution speed of the fast ALU is constantly restricted while waiting for sequential memory accesses over the shared bus.

(d) This performance barrier is known as the Von Neumann bottleneck.

(e) Modified architectures, such as the Harvard architecture, overcome this limitation by using separate physical buses and caches for instructions and data.

Final Answer: The shared bus structure for both instructions and data causing sequential access delays.

Answer: (C)

Go Back to Question 7

Q8.

Solution

Concept: Direct Memory Access (DMA) is a specialized data transfer technique used to move blocks of data between high-speed I/O peripherals and the primary system memory without routing the data directly through the Central Processing Unit.

Solution:

(a) In traditional programmed or interrupt-driven I/O architectures, the CPU must execute specific instructions to read a data word from a peripheral register into an internal CPU register, and then write that word out to memory.

(b) For high-speed transfers, this instruction-overhead consumes significant CPU cycles and creates an processing bottleneck.

(c) With a DMA architecture, a dedicated hardware DMA controller takes control of the system buses.

(d) The CPU only initializes the controller with the transfer direction, starting memory address, and total byte count, then immediately returns to executing other user-level application threads.

(e) The DMA hardware manages the entire block transfer directly. Once the transfer completes, the DMA controller issues a single interrupt to notify the processor, offloading the CPU from high-speed data overhead.

Final Answer: The CPU is completely freed from managing high-speed data transfers after the initial setup.

Answer: (D)

Go Back to Question 8

Q9.

Solution

Concept: In a set-associative cache mapping scheme, a physical main memory address is divided into three distinct bit-fields: Tag, Set Index, and Block Offset. The number of bits in each field depends on the total cache capacity, block size, and the degree of associativity.

Solution:

(a) First, determine the number of bytes contained within a single cache block. The problem statement defines the block size as 64 bytes. Since $64 = 2^6$, the Block Offset field requires exactly 6 bits.

(b) Next, calculate the total number of blocks in the cache: Total Blocks $= \frac{\text{Cache Capacity}}{\text{Block Size}} = \frac{64 \text{ KB}}{64 \text{ bytes}} = \frac{65536}{64} = 1024$ blocks.

(c) The cache is configured as a 4-way set-associative system, meaning each set contains exactly 4 lines/blocks.

(d) Compute the total number of distinct sets: Total Sets $= \frac{\text{Total Blocks}}{\text{Associativity}} = \frac{1024}{4} = 256$ sets.

(e) Since $256 = 2^8$, the Set Index field requires exactly 8 bits to uniquely reference any set within the cache structure.

Final Answer: 8 bits

Answer: (A)

Go Back to Question 9

Q10.

Solution

Concept: The Control Unit relies on several internal registers to orchestrate the basic instruction cycle: Fetch, Decode, and Execute. Each register serves a dedicated, distinct operational purpose during these sequential phases.

Solution:

(a) The Program Counter (PC) is a special-purpose register that holds the memory address of the next sequential instruction scheduled to be fetched and executed.

(b) During the instruction fetch phase, the address stored in the PC is placed onto the address bus. The PC is then automatically incremented to point to the next instruction in sequence.

(c) The Memory Address Register (MAR) holds the immediate address currently being read from or written to by the system bus.

(d) The Instruction Register (IR) stores the actual instruction opcode that was just fetched from memory while it is being decoded by the control logic.

(e) The Accumulator (AC) is a general-purpose working register associated with the ALU that holds intermediate arithmetic and logical operands and results.

Final Answer: Program Counter (PC)

Answer: (A)

Go Back to Question 10

Q11.

Solution

Concept: The execution speed of a CPU is orders of magnitude faster than the access latency of main DRAM. To prevent the processor from idling during memory reads, hardware designers implement a small, high-speed memory hierarchy near the core known as cache memory, which leverages the principle of locality of reference.

Solution:

(a) Locality of reference states that computer programs tend to access a relatively small, localized portion of their address space at any given time window.

(b) Temporal locality implies that if a specific memory location is referenced once, it is highly likely to be accessed again in the near future. Cache capitalizes on this by retaining recently accessed instructions and data words.

(c) Spatial locality implies that accessing a specific memory address increases the probability that nearby adjacent addresses will be accessed soon after. Cache architectures capitalize on this by fetching an entire continuous block or line of memory rather than a single isolated byte.

(d) By loading entire cache lines into high-speed Static RAM (SRAM) cells, subsequent sequential instructions or array traversals hit the cache directly, bypassing the slow system bus.

(e) Other pipeline or pre-fetch mechanisms optimize internal throughput but do not address the physical memory latency gap directly like cache alignment does.

Final Answer: Cache memory alignment

Answer: (B)

Go Back to Question 11

Q12.

Solution

Concept: To convert a fractional decimal number into its equivalent binary representation, the fractional part is iteratively multiplied by the base 2. The integer part resulting from each multiplication step forms the successive bits of the binary fraction, ordered from most significant to least significant.

Solution:

(a) Begin with the decimal fraction 0.4375. Multiply this value by 2: $0.4375 \times 2 = 0.8750$. The integer part is 0, which becomes our first fractional bit after the binary point.

(b) Take the remaining fractional part 0.8750 and multiply it by 2: $0.8750 \times 2 = 1.7500$. The integer part is 1, which becomes our second fractional bit.

(c) Take the remaining fractional part 0.7500 and multiply it by 2: $0.7500 \times 2 = 1.5000$. The integer part is 1, which becomes our third fractional bit.

(d) Take the remaining fractional part 0.5000 and multiply it by 2: $0.5000 \times 2 = 1.0000$. The integer part is 1, which becomes our fourth fractional bit.

(e) Since the remaining fractional component is now exactly zero, the algorithm terminates. Combining the recorded integer bits in chronological sequence yields the final binary representation $(0.0111)_2$.

Final Answer: $(0.0111)_2$

Answer: (C)

Go Back to Question 12

Q13.

Solution

Concept: The problem requires the logical simplification of a Boolean expression containing Exclusive-OR (XOR) operators and an AND operator. The algebraic definition of the two-input XOR function is given by the relation $A \oplus B = A'B + AB'$. We apply basic theorems of Boolean algebra to simplify the combined expression step by step.

Solution:

(a) Let the initial logical function be defined as $W = X \oplus Y \oplus (X \cdot Y)$. We first group and expand the first XOR term: $X \oplus Y = X'Y + XY'$.

(b) Now substitute this back into the expression with the remaining terms, viewing $(X \oplus Y)$ as a single composite variable to XOR with $(X \cdot Y)$: $W = (X \oplus Y)'(XY) + (X \oplus Y)(XY)'$.

(c) Let us analyze the first product term, $(X \oplus Y)'(XY)$. The complement of XOR is XNOR, which is $XY+X'Y'$. Multiplying this by XY yields: $(XY+X'Y')(XY) = XY \cdot XY + X'Y'XY = XY + 0 = XY$.

(d) Now analyze the second product term, $(X \oplus Y)(XY)'$. Expanding $(XY)'$ via De Morgan's theorem gives $X' + Y'$. Multiplying this by the XOR expansion yields: $(X'Y + XY')(X' + Y') = X'Y \cdot X' + X'Y \cdot Y' + XY' \cdot X' + XY' \cdot Y' = X'Y + 0 + 0 + XY' = X'Y + XY'$.

(e) Combining both simplified product terms together gives $W = XY + X'Y + XY'$. Factoring out Y from the first two terms gives $Y(X + X') + XY' = Y(1) + XY' = Y + XY'$. By the distributive absorption law, $Y + XY' = Y + X$, which is equivalent to $X + Y$.

Final Answer: $X + Y$

Answer: (D)

Go Back to Question 13

Q14.

Solution

Concept: The IEEE 754 single-precision floating-point format specifies a biased representation for exponents to avoid dealing with negative binary numbers directly in the hardware comparison units. For 32-bit floats, the exponent field is biased by a constant value of +127.

Solution:

(a) The problem states that the 8-bit biased exponent field contains the binary sequence 10000001.

(b) First, we must convert this unsigned binary sequence into its equivalent decimal integer value. The bit positions correspond to powers of two: $2^7 + 2^0 = 128 + 1 = 129$.

(c) The relationship between the actual, true mathematical exponent ($E_{unbiased}$) and the value encoded within the binary bit pattern (E_{biased}) is defined by the formula: $E_{biased} = E_{unbiased} + \text{Bias}$.

(d) Substituting the known single-precision bias constant of 127 and our calculated decimal value of 129 into the formula yields: $129 = E_{unbiased} + 127$.

(e) Solving for the true value gives: $E_{unbiased} = 129 - 127 = +2$. This means the floating-point number scales its binary mantissa by a factor of 2^2.

Final Answer: +2

Answer: (C)

Go Back to Question 14

Q15.

Solution

Concept: Binary subtraction can be elegantly performed by adding the 2's complement representation of the subtrahend to the minuend. This unifies addition and subtraction hardware inside the CPU Arithmetic Logic Unit. The numbers must first be converted into standard 8-bit true binary notation.

Solution:

(a) Convert the base-10 minuend 45 into an 8-bit unsigned binary sequence: $45 = 32+8+4+1$, which equals the bit pattern 00101101.

(b) Convert the base-10 subtrahend 23 into an 8-bit unsigned binary sequence: $23 = 16+4+2+1$, which equals the bit pattern 00010111.

(c) To perform the subtraction using addition, find the 2's complement of 23. First invert the bits of 23 to get the 1's complement: 11101000. Then add 1 to the least significant bit, yielding 11101001.

(d) Now add the binary representation of 45 to the 2's complement representation of -23: 00101101 + 11101001.

(e) Performing the column-by-column addition yields 100010110. Because we are restricted to an 8-bit register architecture, the leftmost carry-out bit beyond the 8th position is discarded, leaving the final 8-bit result as 00010110, which equals decimal 22.

Final Answer: 00010110

Answer: (C)

Go Back to Question 15

Q16.

Solution

Concept: The maximum direct memory addressability of a processor is strictly determined by the width of its physical address bus, completely independent of the data bus width. The address bus determines how many unique memory locations the processor can reference, while the data bus determines how much data can be transferred per cycle.

Solution:

(a) The system specifies a 32-bit address bus. This implies that the CPU can generate 2^{32} unique binary addresses to reference distinct memory locations.

(b) Since the memory system is defined as byte-addressable, each unique binary address corresponds to exactly one byte (8 bits) of physical data storage.

(c) Therefore, the maximum addressable space is calculated as: Capacity = $2^{32} \times 1$ byte.

(d) Using standard binary prefixes, we can break down the power of two: $2^{32} = 2^2 \times 2^{30} = 4 \times 2^{30}$.

(e) Since 2^{30} represents a Gigabyte (GB), the total addressable physical memory capacity evaluates precisely to 4 GB. The 64-bit width of the data bus allows for wider parallel data words to be fetched at one time but does not increase the number of addressable slots.

Final Answer: 4 GB

Answer: (D)

Go Back to Question 16

Q17.

Solution

Concept: Asymmetric or public-key cryptography relies on mathematical trapdoor functions that are computationally easy to perform in one direction but extraordinarily difficult to invert without private key information. Different algorithms base their security on different mathematical problems.

Solution:

(a) The RSA algorithm (developed by Rivest, Shamir, and Adleman) derives its cryptographic strength directly from the prime factorization problem. It is simple to compute the product of two ultra-large prime numbers, but factoring that composite integer back into its constituent primes is computationally infeasible within realistic human timeframes using current technology.

(b) Advanced Encryption Standard (AES) and Data Encryption Standard (DES) are symmetric-key cryptographic algorithms that rely on substitution-permutation networks rather than prime factor mathematical problems.

(c) Elliptic Curve Cryptography (ECC) represents an alternative public-key methodology that bases its security on the algebraic structure of elliptic curves over finite fields, relying on the discrete logarithm problem rather than prime factorization.

Final Answer: RSA

Answer: (B)

Go Back to Question 17

Q18.

Solution

Concept: To convert a number from hexadecimal (base 16) to octal (base 8), it is most efficient to use binary (base 2) as an intermediate step. Each hexadecimal digit maps directly to a group of 4 binary bits, and each octal digit maps directly to a group of 3 binary bits.

Solution:

(a) Convert each digit of the given hexadecimal value $(A4C.8)_{16}$ into its 4-bit binary equivalent: $A = 1010$, $4 = 0100$, $C = 1100$, and the fractional part $8 = 1000$. Combining these yields the continuous binary string: 101001001100.1000.

(b) To convert this binary sequence into octal, regroup the bits into clusters of 3 starting from the radix point and moving outward in both directions.

(c) Grouping the integer portion from right to left gives: $(100)_2$, $(001)_2$, $(001)_2$, and $(101)_2$.

(d) Convert each of these 3-bit integer clusters back into their decimal equivalents: $101 = 5$, $001 = 1$, $001 = 1$, and $100 = 4$. This forms the integer component 5114.

(e) Grouping the fractional portion from the radix point going left to right gives: $(100)_2$ followed by remaining trailing zeros $(000)_2$, which evaluate simply to 4. Combining both segments yields the final base-8 representation $(5114.4)_8$.

Final Answer: $(5114.4)_8$

Answer: (D)

Go Back to Question 18

Q19.

Solution

Concept: The problem asks for the minimal Sum-of-Products expression for a 3-variable Boolean function defined by minterms $F(A, B, C) = \Sigma m(0, 2, 4, 5, 6)$. We can utilize a standard 2 by 4 Karnaugh Map (K-Map) structure to visually group adjacent minterms and eliminate redundant literals.

Solution:

(a) Set up a K-Map where the rows correspond to variable A (A' and A) and the columns correspond to variables B and C ($B'C'$, $B'C$, BC, BC').

(b) Place binary 1s into the grid cells corresponding to the given minterm positions: cell 0 ($A'B'C'$), cell 2 ($A'BC'$), cell 4 ($AB'C'$), cell 5 ($AB'C$), and cell 6 (ABC').

(c) Identify groupings of adjacent 1s. We observe that cells 0, 2, 4, and 6 occupy the four outer corners of the K-Map matrix. These four cells can be combined together into a single quad group.

(d) Analyzing this quad group across the rows, variable A changes state and drops out. Analyzing across the columns, variable B changes state, and variable C remains fixed at 0. Thus, this quad groups simplifies strictly to the single product term C'.

(e) There is still an unisolated 1 remaining in cell 5 ($AB'C$). It can be combined with the adjacent 1 in cell 4 ($AB'C'$) to form a pair group. For this pair, variable A is constant at 1, variable B is constant at 0, and variable C changes state. This pair simplifies to the product term AB'.

(f) Summing the simplified terms from the quad group and the pair group yields the minimal SOP expression: $C' + AB'$.

Final Answer: $C' + AB'$

Answer: (A)

Go Back to Question 19

Q20.

Solution

Concept: Modern operating systems implement virtual memory to allow programs to execute without requiring their entire address space to reside concurrently in physical RAM. Virtual addresses must be dynamically translated to physical memory addresses via page tables. Because page tables reside in main memory, looking them up directly adds a severe latency penalty.

Solution:

(a) The Translation Lookaside Buffer (TLB) is an associative, high-speed hardware cache integrated directly inside the processor's Memory Management Unit (MMU).

(b) The TLB stores a small subset of recent virtual-page to physical-frame address translations.

(c) When the CPU generates a virtual address request, the MMU first checks the TLB. If a matching virtual page entry exists (a TLB hit), the physical frame address is retrieved in a fraction of a clock cycle.

(d) If a TLB miss occurs, the system must perform a page table walk through the multi-level page tables located in slow main memory, causing significant latency.

(e) Thus, the primary purpose of the TLB is to accelerate address translation speed, bypassing sluggish page table lookups.

Final Answer: Cache recent virtual-to-physical address translations to accelerate page table lookups.

Answer: (A)

Go Back to Question 20

Answer Key

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