# Computer Organization and Architecture Test 4

**Number of Questions: 35** **Section Marks: 30**

***Directions for questions 1 to 35:*** Select the correct alternative from the given choices.

**1.** The primary memory of a computer consists of $M$ locations, and the computer has a 4-way set associative cache consisting of '$C$' locations. On an average, how many different locations in primary memory map to a particular location in the cache?

(A) 4 (B) $\frac{M}{C}$
(C) $\frac{4M}{C}$ (D) $M$

**2.** Which of the following statement is FALSE?

(A) The Program Counter and Stack Pointer are updated when a procedure call is executed on a processor.
(B) When a Procedure call is executed on a processor, data cache is flushed to avoid aliasing problems.
(C) A direct mapped cache can have a higher miss-rate than an associative mapped cache, each are of same size (i.e., same number of blocks).
(D) Programs with high temporal locality have a low cache miss rate penalty because, exactly the same addresses are re-referenced.

**3.** Let multiplicand be 23, multiplier be 29 and each number is represented using 6-bits. Then the difference between the total number of add/subtract operations required using unsigned binary multiplication and booths algorithm will be ________.

(A) 0 (B) 1
(C) 2 (D) 3

**4.** Consider a cache with a line size of 16 bytes and a main memory that requires 20 ns to transfer a 4-byte word. For any line that is written at least once before being swapped out of the cache, what is the average number of times that the line must be written before being swapped out for a write-back cache to be more efficient than a write through cache?

(A) more than 2 times (B) more than 3 times
(C) more than 4 times (D) more than 8 times

**5.** A certain pipelined RISC machine has 8 general purpose registers $R_0, R_1, ..., R_7$ and supports the following operations:

ADD $RS1$, $RS2$, Rd (Adds $RS1$ and $RS2$ and put the sum in Rd.)

MUL $RS1$, $RS2$, Rd (Multiply $RS1$ by $RS2$ and put the product in Rd.)

An operation normally takes one cycle; however, an operation takes two cycles if it produces a result required by the immediately following operation in an operation sequence. Consider the expression $xy + xz + xyz$, where variables $x$, $y$, $z$ are located in registers $R_0$, $R_1$, $R_2$. The contents of these registers must not be modified, then the minimum number of clock cycles required for an operation sequence that compute the value of $xy + xz + xyz$ is ______.

(A) 5 (B) 6
(C) 7 (D) 8

**6.** Consider two processors, 5-stage pipeline $P_5$ and 7-stage pipeline $P_7$ to implement the same instruction set. Processor $P_5$ has a clock cycle of 20 ns and $P_7$ has a clock cycle of 17.5 ns. Which of the following is/are TRUE?

I. $P_7$ pipeline has better maximum throughput than $P_5$'s pipeline.
II. The latency of a single instruction is shorter on $P_7$'s pipeline than on $P_5$'s pipeline.
III. Programs executing on $P_7$ will always run faster than the programs executing on $P_5$.

(A) I only (B) II only
(C) I and III only (D) II and III only

**7.** What is the size of the cache memory required for a direct-mapped cache with 16 KB of data and 16 B blocks, assuming a 32-bit address of main memory?

(A) 145 K bits
(B) 146 K bits
(C) 147 K bits
(D) 148 K bits

**8.** In a computer the exponents of a floating point number are represented as 'excess-64' integers. Two such exponents are input to a conventional 7-bit parallel adder. Which of the following should be accomplished in order to obtain a sum that is also in excess-64 notation?

(A) An end-around carry should be added to LSB.
(B) The adder outputs should be computed bitwise.
(C) The most significant adder output bit should be complemented.
(D) The adder outputs should be left unchanged.

**9.** The format of a double operand instruction of a CPU is shown below:

| opcode | src data | dest data |
|---|---|---|
| ← 6-bits → | ← 4-bits → | ← 4-bits → |

20 double-operand instructions and 40 single-operand instructions must be implemented and the opcode field must identify 3-groups of $n$-operand instructions. The total number of zero-operand instructions that can be implemented is ______.

(A) 704 (B) 11264
(C) 16384 (D) 10496

**10.** A CPU has 32-bit memory address and a 256KB cache memory. The cache is organized as a 4-way set-associative cache with a cache block size of 4 words (32-bits per word). The smallest addressable unit is a byte. Then the percentage of cache memory used for tag bits is _________.

(A) 11.11 (B) 12
(C) 80.8 (D) 88.89

**11.** Assume that we have three significant decimal digits. What is the sum of 3.67 and $3.45 \times 10^2$ using guard and round digits and the sum of the two numbers without using guard and round digits respectively?

(A) 348, 349 (B) 349, 348
(C) 350, 347 (D) 348, 348

**12.** If an encoded micro-instruction format is used, then how a 10-bit micro-operation field can be divided into subfields to specify 41 different actions efficiently?

(A) 5, 5 (B) 2, 4, 4
(C) 5, 3, 2 (D) 5, 4, 1

**13.** The two-stages of a pipelined processor are Bus Interface Unit (BIU) and Execution Unit (EU). The BIU fetches instructions from a 4-byte instruction queue. It also calculates address, fetches operands and writes results on to memory as requested by EU. If no such requests are outstanding and the bus is free, the BIU fills vacancies (if any) in the instruction queue. When the EU completes execution of an instruction, it passes results to BIU and proceeds to the next instruction. Let the tasks performed by the BIU and EU take about equal time. The factor by which pipelining improves the performance of the processor (Ignore the effect of branch Instructions) is _________.

(A) 2 (B) 4
(C) 5 (D) 8

**14.** Consider a processor that includes a base with indexed addressing mode. Suppose an instruction is encountered that employs this addressing mode and specifies a displacement of 2970, in decimal. Currently the base and index register contain the decimal numbers 58022 and 8 respectively. Then the address of operand would be _________.

(A) 58022 (B) 60992
(C) 61000 (D) 2978

**15.** Consider a RAID level 5 organization comprising of five disks, with the parity for sets of four blocks on four disks stored on the fifth disk. The number of blocks that are accessed to write one block of data is _________.

(A) 3 (B) 4
(C) 5 (D) 6

**16.** Consider a static two-issue pipeline for MIPS. The two issue packets are, one ALU/branch instruction, one load/store instruction. Which of the following is the correct re-ordering of the below loop to avoid as many pipeline stalls as possible? Assume branches are predicted, so that control hazards are handled by the hardware.

| | |
|---|---|
| loop: Load $R_0$, $M[100]$; | $R_0 \leftarrow M[100]$ |
| ADD $R_0$, $R_0$, 5; | $R_0 \leftarrow R_0 + 5$ |
| Store $R_0$, $M[100]$ ; | $M[100] \leftarrow R_0$ |
| ADD $R_1$, $R_1$, –4; | $R_1 \leftarrow R_1 - 4$ |
| BNE $R_1$, 0, loop; | branch if $R_1 \neq 0$ |

(A) loop: Load $R_0$, $M[100]$
ADD $R_1$, $R_1$, –4
ADD $R_0$, $R_0$, 5
BNE $R_1$, 0, loop; Store $R_0$, $M[100]$
(Both in single cycle).

(B) loop: Load $R_0$, $M[100]$
ADD $R_1$, $R_1$, –4
BNE $R_1$, 0, loop; ADD $R_0$, $R_0$, 5
(Both in single cycle)
Store $R_0$, $M[100]$

(C) Loop: Load $R_0$, $M[100]$
ADD $R_0$, $R_0$, 5
ADD $R_1$, $R_1$, –4
Store $R_0$, $M[100]$
BNE $R_1$, 0, loop

(D) loop: ADD $R_1$, $R_1$, –4
BNE $R_1$, 0, loop
Load $R_0$, $M[100]$
ADD $R_0$, $R_0$, 5; Store $R_0$, $M[100]$;
(Both in single clock cycle)

**17.** Consider a single level cache with an access time of 2.5 ns, a line size of 64 Bytes and a hit ratio of 0.95. Main memory uses a block transfer capability that has a first word (4 bytes) access time of 60 ns and an access time of 5 ns for each word thereafter.
What is the access time when there is a cache miss, Assume that the cache waits until the line has been fetched from main memory and then re-executes for a hit.

(A) 9.25 ns (B) 9.375 ns
(C) 9.625 ns (D) 9.5 ns

**18.** For the data given in Q.No:17, Suppose that increasing the line size to 128 bytes increases hit ratio to 0.97, then the average memory access time

(A) increases
(B) decreases
(C) remains constant
(D) will be unpredictable

**19.** Consider an instruction pipeline with five stages without any branch prediction: Fetch Instruction (FI), Decode instruction (DI), Fetch operand (FO), Execute instruction (EI) and Write operand (WO). The stage delays for FI, DI, FO, EI and WO are 3 ns, 4 ns, 6 ns, 2 ns and 4 ns respectively. There are intermediate storage buffers after each stage and delay of each buffer is 1 ns. A program consisting of 10 instructions $I_1, I_2, ..., I_{10}$ is executed in this pipelined processor. Instruction $I_3$, is the only branch instruction and its branch target is $I_7$. If the branch is taken during the execution of this program, the time needed to complete the program (in nano seconds) is ________.

(A) 49 (B) 56
(C) 98 (D) 105

**20.** Consider the following program segment for a hypothetical CPU having three user registers $R_1$, $R_2$ and $R_3$.

| Instruction | Operation | Instruction size (in words) |
|---|---|---|
| MOV $R_1$, 6000 | $R_1 \leftarrow$ M[6000] | 2 |
| MOV $R_2$,($R_1$) | $R_2 \leftarrow$ M[($R_1$)] | 1 |
| SUB $R_2$, $R_3$ | $R_2 \leftarrow R_2$ - $R_3$ | 1 |
| MOV 6000, $R_2$ | M[6000] $\leftarrow R_2$ | 2 |
| HALT | Machine halts | 1 |

Let the clock cycles required for various operations be as follows:
Instruction fetch and decode: 2 clock cycles per word.
Register to/from memory transfer: 3 clock cycles.
SUB with both operands in registers: 1 clock cycle
Then the total number of clock cycles required to execute the program will be ________.

(A) 2 (B) 23
(C) 24 (D) 19

**21.** Consider a 32KB, two-way set associative cache with 64-byte block size. The CPU generates 32-bit addresses, A 2-to-1 multiplexer has a latency of 0.5ns while a K-bit comparator has a latency of K/10 ns, then the hit latency of the cache is _______.

(A) 1.8 (B) 2.3
(C) 2.4 (D) 3

**22.** According to IEEE standard, a 32-bit, single precision, floating point number $N$ is defined to be:
$N = (-1)^S \times 1.M \times 2^{E-127}$
Where $S$: The sign bit
$M$: Fractional mantissa
$E$: The Biased Exponent
A floating point number is stored as SEM, where $S$, $E$ and $M$ are stored in 1-bit, 8-bits and 23-bits respectively.
What is the value of the floating point number $C0E80000$ (hexa decimal notation)?

(A) –7.25 (B) –29
(C) –0.453 (D) –7

**23.** For the data given in Q.No. 22, using given notation, what is the hexa decimal equivalent of 565?

(A) 048$D$4000 (B) 440$D$4000
(C) 480$D$4000 (D) 444$D$4000

**24.** Consider an 8 KB direct-mapped write-back data cache with 16 byte blocks. The elements of two arrays $A$ and $B$ are 8 bytes long. Array $A$ has 5 rows and 100 columns and array $B$ has 5 rows and 101 columns. (Initially cache is empty). To execute below code, how many cache misses will occur? (Arrays are stored in row-major order).
for ($i$ = 0; $i$ < 5; $i$++ )
for ($j$ = 0; $j$ < 100; $j$++ )
A[$i$][$j$] = $B$[$j$] [0] * $B$[$j$+1][0];

(A) 251 (B) 351
(C) 451 (D) 551

**25.** Consider a processor with a 5-stage MIPS pipeline (IF, ID, EX, MEM and WB) with no forwarding and each stage takes 1 cycle. The register file can be written and then read in same cycle. In the below program, the destination is the first (left most) register.

| Instruction | Operation |
|---|---|
| LD $R_0$, ($R_2$)<br>LD $R_2$, 4($R_3$)<br>ADD $R_3$, $R_2$, $R_0$<br>SUB $R_3$, $R_0$, $R_0$<br>OR $R_4$, $R_3$, 0<br>SW $R_3$, 5($R_2$) | $R_0 \leftarrow M(R_2)$<br>$R_2 \leftarrow M[R_3 + 4]$<br>$R_3 \leftarrow R_2 + R_0$<br>$R_3 \leftarrow R_0 - R_0$<br>$R_4 \leftarrow R_3$ OR 0<br>$R_3 \leftarrow M[R_2 + 5]$ |

How many stall cycles do the program incur?

(A) 1 (B) 2
(C) 3 (D) 4

**26.** Consider a deeper pipeline in which, it takes atleast three pipeline stages before the branch-target address is known and an additional cycle before the branch condition is evaluated, assuming no stalls on the registers in the conditional comparison. The 3-stage delay leads to the branch penalties for three simple branch prediction schemes, which are listed below:

| **Prediction Scheme** | **Frequency** |
|---|---|
| Unconditional Branch | 5% |
| Conditional branch, Taken | 5% |
| Conditional branch, Untaken | 10% |

| **Branch scheme** | **Penalty (unconditional)** | **Penalty (taken)** | **Penalty (Untaken)** |
|---|---|---|---|
| Flush Pipeline | 2 | 3 | 3 |
| Predicted Taken | 2 | 2 | 3 |
| Predicted Untaken | 2 | 3 | 0 |

Find the effective addition to CPI arising from each branch scheme for this pipelining (By taking above frequencies into count)?

(A) 0.5, 0.5, 0.3 (B) 0.3, 0.4, 0.6
(C) 0.55, 0.5, 0.25 (D) 1.3, 1.4, 1.6

**27.** Assume that the number of clock cycles for a polling operation is 200. Consider a processor which executes at 60 MHz, what is the overhead of polling in percentage

(i) For a mouse that is polled 20 times per second.

(ii) For a hard disk transferring data in 1 word chunks at 1 MB/sec.

(A) 0.006%, 100% (B) 0.0067%, 87.4%
(C) 0.0067%, 43.7% (D) 0.0033%, 43.7%

**28.** A CPU, which addresses the data through its 8 registers in one of the 15 different modes, is to be designed to support 15 arithmetic instructions, 10 logic instructions, 24 data-moving instructions, 6 branch instructions, 5 control type instructions. Of these instructions respectively 20%, 60%, 50%, 50% and 60% are either single-operand or no-operand instructions and the rest are of double-operand type. What is the minimum size of the CPU's instruction word?

(A) 15-bits (B) 20-bits
(C) 19-bits (D) 21-bits

**29.** Consider the following sequence of micro-operations:

MAR ← (IR (address))
MBR ← (PC)
PC ← (IR(address))
Memory ← (MBR)
PC ← (PC) + I
I is instruction Length

Which one of the following is a possible operation performed by this sequence?

(A) Instruction fetch
(B) Initiation of interrupt service
(C) Branch-and-save address
(D) Increment and skip if zero.

**30.** Suppose that a 1 GHz processor needs to read 2000 bytes of data from a particular I/O device. The I/O device supplies 1 byte of data every 0.02 ms. The code to process the data and store it in a buffer takes 1000 cycles.

If the processor detects that a byte of data is ready through polling and a polling iteration takes 80 cycles, how many cycles does the entire operation take?

(A) 42000 K cycles (B) 40000 K cycles
(C) 40080 K cycles (D) 39920 K cycles

**31.** For the data given in Q. No. 30, If instead, the processor is interrupted when a byte is ready and the processor spends the time between interrupts on another task; then how many cycles are saved by using interrupt-driven over polling technique. The overhead of handling an interrupt is 200 cycles.

(A) 39680 K cycles (B) 39600 K cycles
(C) 38960 K cycles (D) 37680 K cycles

**32.** A CPU has 64 KB direct mapped cache with 128-byte block size. Suppose $X$ is a two dimensional array of size 512 × 512 with elements that occupy 8-bytes each. Consider the following C-code segment:

```
for (i = 0; i < 512 ; i ++)
{
for (j = 0; j < 512 ; j ++)
{
a+ = X [i] [j];
}
}
```

Initially array $X$ is not in the cache and $i$, $j$, $a$ are in registers. The number of cache misses experienced by given code fragment is ________.

(A) 8196 (B) 16384
(C) 32768 (D) 65536

**33.** Consider the execution of the following sequence of instructions on a five-stage pipeline consisting of:

IF: Instruction Fetch
ID: Instruction Decode
OF: Operand Fetch
IE: Instruction Execution
IS: Store results

| Instruction | Operation |
|---|---|
| LOAD $R_1$, 1; | $R_1 \leftarrow 1$ |
| LOAD $R_2$, 5; | $R_2 \leftarrow 5$ |
| SUB $R_2$, $R_2$, 1; | $R_2 \leftarrow R_2 - 1$ |
| ADD $R_3$, $R_1$, $R_2$; | $R_3 \leftarrow R_1 + R_2$ |
| ADD $R_6$, $R_4$, $R_5$ | $R_6 \leftarrow R_4 + R_5$ |
| SL $M$[1000], $R_3$; | $M[1000] \leftarrow R_3$ |
| ADD $R_7$, $R_4$, $R_6$; | $R_7 \leftarrow R_4 + R_6$ |

The speedup achieved by executing this program on a pipelined processor compared to a sequential processor (Let each stage takes one clock cycle) is _______.

(A) 1.365 (B) 2
(C) 2.1875 (D) 3.236

**34.** In a program, an array is declared as int $A$[1024]. Each array element is 2 bytes in size. In main memory the array is stored at the address 0X00000000. This program is run on a computer that has a direct mapped data cache of size 4 KB, with line size of 8 bytes. Which of the following element of the array conflict with the element $A$[0] in the data cache?

(A) $A$[256] (B) $A$[512]
(C) $A$[63] (D) No conflict

**35.** For the data given in Q. No. 34, If the program accesses the elements of this array one by one in reverse order that is starting with the last element and ending with the first element, how many data cache misses would occur?

(Assume that the data cache is initially empty and that no other data or instruction accesses are to be considered)

(A) 64 (B) 256
(C) 512 (D) 1024

## Answer Keys

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **1.** C | **2.** B | **3.** A | **4.** C | **5.** B | **6.** A | **7.** B | **8.** C | **9.** D | **10.** A |
| **11.** B | **12.** C | **13.** A | **14.** C | **15.** B | **16.** A | **17.** B | **18.** B | **19.** C | **20.** C |
| **21.** B | **22.** A | **23.** B | **24.** B | **25.** B | **26.** C | **27.** B | **28.** B | **29.** C | **30.** C |
| **31.** D | **32.** B | **33.** C | **34.** D | **35.** B | | | | | |

## Hints and Explanations

**1.** If primary memory has '$M$' locations and cache has '$C$' locations then each location on the cache is responsible for $M/C$ locations in direct mapping. For 4-way set associative, the cache can hold $4 \times \frac{M}{C}$ locations.
Choice (C)

**2.** Choice (B)

**3.** Multiplicand ($M$) = 23 = 010111
Multiplier ($Q$) = 29 = 011101
Using unsigned binary multiplication, perform addition if
$Q_0 = 1$ i.e., LSB of $Q$ register is 1 so number of additions equal to number of '1's in $Q$. Here number of 1's in $Q = 4$.
In booth algorithm, perform addition or subtraction if $Q_0, Q_{-1}$ = 01 or 10 respectively. i.e., whenever a bit change occurred, perform either addition or subtraction.
Multiplier ($Q$) = 01 1 1010 ← initially $Q_{-1} = 0$
∴ 4 ADD/SUB operations required.
Hence difference is zero. Choice (A)

**4.** In write-back case, each dirty line is written back once, i.e., at a swap-out time, taking $\frac{16}{4} \times 20 = 80$ ns.
In write-through case, each update of the line requires that one word be written out to main memory, taking 20 ns.
∴ If the average line that gets written atleast once gets written more then $\frac{80}{20} = 4$ times before swap out, then write back is more efficient. Choice (C)

**5.** $R_0 \leftarrow x, R_1 \leftarrow y, R_2 \leftarrow z$
To compute the value $xy + xz + xyz$ on this RISC processor, the operations are
$I_1$: MUL $R_0, R_1, R_3$ ... $R_3 \leftarrow xy$
$I_2$: MUL $R_0, R_2, R_4$ ... $R_4 \leftarrow xz$
$I_3$: MUL $R_3, R_2, R_5$ ... $R_5 \leftarrow xyz$
$I_4$: ADD $R_3, R_4, R_6$ ... $R_6 \leftarrow xy + xz$
$I_5$: ADD $R_5, R_6, R_6$ ... $R_6 \leftarrow xy + xz + xyz$
Each of $I_1, I_2, I_3, I_4$ require one clock cycle. $I_5$ requires two clock cycles as $R_6$ is used in $I_5$.
∴ Minimum clock cycles required = 6 Choice (B)

**6.** Increasing number of stages will increase the throughput but increases latency of single instruction. (III) Condition not valid always. Choice (A)

**7.** Data in cache = 16 KB = $2^{14}$B
Block size = 16 B
Number of blocks $= \frac{2^{14}}{2^4} = 2^{10}$
CPU generates 32-bit addresses.

| Tag | Block | Word |
|---|---|---|

Word field requires 4-bits as block size is 16 B. Block field requires 10-bits as there are $2^{10}$ blocks.
Tag = 32 – (10+4) = 18-bits
∴ Cache size = $2^{10} \times (4 \times 32 + 18)$ = 146 K bits
(as each block has 4 × 32-bit data and 18-bits tag).
Choice (B)

**8.** Let the two exponents to be added are 13 and 15.
13 in excess - 64 = 13 + 64 = 77 = 1001101
15 in excess - 64 = 15 + 64 = 79 = 1001111
Output from 7-bit parallel adder = 0011100
But 13 + 15 + 64 = 92 = 1011100
Hence complement the MSB of the output.
Choice (C)

**9.** Opcode field size = 6-bits
∴ Total number of operations possible = $2^6 = 64$
Out of these 64, only 20 operations are used by double-operand instructions.
Remaining = 64 – 20 = 44.
In single-operand instructions, src data will be included in opcode field.
∴ Possible single-operand operations = $44 \times 2^4$
Out of these only 40 are used for single-operand instructions.
∴ Remaining = $41 \times 2^4$
(∵ Three $2^4$'s are used for single operand instruction)
In zero-operand instructions there will be no operands, so it will be included in opcode field.
∴ Zero-operand instructions
$= 41 \times 2^4 \times 2^4 = 10496$ Choice (D)

**10.** Address generated by CPU has 32-bits.
Cache memory = 256 KB = $2^{18}$ Bytes
Block size = 4 words
= 4 × 32 bits (∵ 1 word = 32-bits)
$= \frac{4 \times 32}{8}$ bytes = 16 bytes = $2^4$ B

$$\text{Number of lines} = \frac{2^{18}}{2^4} = 2^{14}$$

$$\text{Number of sets} = \frac{2^{14}}{2^2} = 2^{12}$$

| Tag | Set | Word |
|---|---|---|
| 16 | 12 | 4 |

32 bits

Percentage of the cache memory is used for tag bits

$$= \frac{16 \times 2^{14}}{(16+128) \times 2^{14}} \times 100 \ (\because \text{ each block has 128-bits})$$

$$= \frac{16}{16+128} \times 100 = 11.11\%$$

Choice (A)

**11.** Given numbers 3.67 and $3.45 \times 10^2$. To add these two numbers, first align the exponents.
i.e., shift the smaller number to right.
$3.67 = 0.0367 \times 10^2$
The guard digit holds 6 and round digit holds 7.
The sum will be

0.0367
<u>3.4500</u>
<u>3.4867</u>

$\therefore$ Sum $= 3.4867 \times 10^2$
Here we need to round two digits; after rounding sum will be $3.49 \times 10^2 = 349$
Without round and guard digits,

3.45
<u>0.03</u>
<u>3.48</u>

$\therefore$ Sum $= 3.48 \times 10^2 = 348$ Choice (B)

**12.** 10-bit micro-operation field can be divided into 3 sub-fields 5, 3, 2 to yield 31, 7, 3 actions respectively.
Choice (C)

**13.** Each stage takes equal amount of time. Let '$x$' be the time taken by BIU and EU stages.
Time taken to execute '$n$' instructions on this pipeline
$= [2 + n - 1]\, x = (n + 1)x$
Without pipelining time taken will be $= 2.n.x$

$$\text{Speed up ratio} = \frac{2nx}{(n+1)x}$$

For large number of instructions, $n >> x$.

$$\therefore \text{ Speed up ratio} = \frac{2n}{n} = 2.$$

Hence pipeline improves the performance by a factor of 2. Choice (A)

**14.** The addressing mode is base with indexing mode.
Displacement = 2970
Base register value = 58022
Index register value = 8
Address of operand = 2970 + 58022 + 8 = 61000
Choice (C)

**15.** To write one block of data the tasks performed are:
1. Read the parity
2. Read target block old data
3. Compute new parity
4. Write new block
5. Write new parity

$\therefore$ 4 times the blocks are accessed. Choice (B)

**16.** The first three instructions have data dependencies and so do the last two. The best reorder for these instructions is:

| Loop: | ALU/branch instruction | LOAD/STORE instruction | Clock cycle |
|---|---|---|---|
| | | LOAD $R_0$, $M$[100] | 1 |
| | ADD $R_1$, $R_1$, –4 | | 2 |
| | ADD $R_0$, $R_0$, 5 | | 3 |
| | BNE $R_1$, 0, Loop | STORE $R_0$, $M$[100] | 4 |

Hence only one pair of instructions has both issue slots used. Choice (A)

**17.** Cache access time = 2.5 ns
Line size = 64 Bytes
Hit ratio = 0.95
Access time of first byte in main memory = 60 ns
Next word access time = 5 ns

$$\text{Number of words in each line} = \frac{64}{4} = 16 \text{ words}$$

For 1st word 60 ns and for remaining 15 words, $15 \times 5$ ns required.
Cache access time when there is a miss
$= 0.95 \times 2.5 + 0.05\,(2.5 + 60 + 15 \times 5 + 2.5)$
$= 9.375$ ns Choice (B)

**18.** New line size = 128 bytes
$H = 0.97$

$$\text{Now, words in each line} = \frac{128}{4} = 32$$

First word requires 60 ns and remaining 31 words require $31 \times 5$ ns.
$\therefore$ Cache access time when there is a miss
$= 0.97 \times 2.5 + 0.03 \times (2.5 + 60 + 31 \times 5 + 2.5)$
$= 9.025$
$\therefore$ Average memory access time reduced.
Choice (B)

**19.** Stage delays for FI, DI, FO, EI and WO are 3, 4, 6, 2, 4 ns respectively.
Maximum time is taken by FO. i.e., 6 ns and additional 1 ns is required for delay. So total time for an instruction to pass from one stage to another is 7 ns. The instruction execution sequence is $I_1, I_2, I_3, I_7, I_8, I_9, I_{10}$.
When $I_3$, is in its execution stage, we detect the branch and when $I_3$ is in 'WO' stage we fetch $I_7$.
Execution time for $I_1, I_2, I_3 = [5 + (3 - 1)]\, 7 = 49$ ns.

Execution time for $I_7, I_8, I_9, I_{10} = [5 + (4 - 1)]7 = 56$ ns
But we fetch $I_7$ in $I_3$'s WO stage, so we saved 7 ns
Hence total time = 49 + 56 – 7 = 98 ns.

Choice (C)

**20.** Each instruction requires fetch and decode stage.

| | Fetch & Decode | Operation |
|---|---|---|
| MOV $R_1$, 6000 | 2 × 2 | 3 |
| MOV $R_2$, ($R_1$) | 2 | 3 |
| SUB $R_2$, $R_3$ | 2 | 1 |
| MOV 6000, $R_2$ | 2 × 2 | 3 |
| HALT | 2 | – |

∴ Total clock cycles required = 7 + 5 + 3 + 7 + 2 = 24

Choice (C)

**21.** Cache capacity = 32 KB = $2^{15}$B
Block size = 64B = $2^6$B
∴ word size = 6-bits.

The cache is 2-way set associative, so $2 \times 2^v = \dfrac{2^{15}}{2^6}$

⇒ $2^V = 2^8$
∴ set size = 8-bits
CPU generates 32-bit addresses.

← 32 →

| Tag | Set | Word |
|---|---|---|
| ← 18 → | ← 8 → | ← 6 → |

First a 2-to-1 multiplexer is required whose latency is 0.5 ns. After that the tag bits are compared using a comparator.

For 18-bit comparator $\dfrac{18}{10}$ ns required.

∴ Hit latency = 0.5 + 1.8 = 2.3 ns

Choice (B).

**22.** Given number $C0E80000$
In binary, it will be

← 32 →

1 | 100 00001 | 110 1000 0000 0000 0000 0000

↑ sign ↑ exponent ↑ mantissa

$S = 1$
$E$ = 100 00001 = 129
Mantissa = 110 1000 0000 0000 0000 0000
∴ $N = (-1)^1 \times 1.1101 \times 2^{129-127}$
$= -1.1101 \times 2^2$
$= -111.01 = -7.25$

Choice (A)

**23.** 565 = 1000110101
$N = (-1)^0 \times 1.000110101 \times 2^{136-127}$
IEEE standard binary equivalent of
$N = \underline{0\,100}\ \underline{0\,100}\ \underline{0\,000}\ \underline{1101}\ \underline{0100}\ \underline{0000}\ \underline{0000}\ \underline{0000}$
4 4 0 D 4 0 0 0
= 440$D$4000

Choice (B)

**24.** Elements of "$A$" are written in the order in which they are stored in memory. So $A$ will benefit from spatial locality. The even values of $j$ will miss and odd values will hit. i.e., Each block consists two values. (For example $a$[0] [0] , $a$[0] [1] or $a$[0] [2], $a$[0] [3] etc.)
Since $A$ has 5 rows and 100 columns, its accesses will lead to $5 \times \dfrac{100}{2} = 250$ misses.
The array '$B$' does not benefit from spatial locality since the accesses are not in the order it is stored. But it will benefit from temporal locality.
The misses due to '$B$' will be for $B[j + 1]$ [0] accesses when $i = 0$ and also the first access to $B[j][0]$ when $j = 0$. Since $j$ goes from $j = 0$ to 99 when $i = 0$, accesses to $B$ lead to 100 + 1 or 101 misses.
∴ Total cache misses = 250 + 101 = 351.

Choice (B)

**25.** There are only two stall cycles. One is after 2nd load as $R_2$ is immediately required in ADD instruction. Another stall is after 'SUB' as $R3$ is immediately required by 'OR' instruction.

Choice (B)

**26.** We find the CPI's of each prediction by multiplying the relative frequencies with respective penalties.

| **Branch scheme** | **Frequency** | **Unconditional** | **Conditional, Taken** | **Conditional, Untaken** |
|---|---|---|---|---|
| Stall Pipeline | 5% | $\frac{5}{100} \times 2 = 0.1$ | $\frac{5}{100} \times 3 = 0.15$ | $\frac{10}{100} \times 3 = 0.3$ |
| Predicted Taken | 5% | $\frac{5}{100} \times 2 = 0.1$ | $\frac{5}{100} \times 2 = 0.1$ | $\frac{10}{100} \times 3 = 0.3$ |
| Predicted Untaken | 10% | $\frac{5}{100} \times 2 = 0.1$ | $\frac{5}{100} \times 3 = 0.15$ | $\frac{10}{100} \times 0 = 0$ |

Total branches frequency = 5 + 5 + 10 = 20%

Using stall pipeline, additional CPI = 0.1 + 0.15 + 0.3 = 0.55
Using predicted taken, additional CPI = 0.1 + 0.1 + 0.3 = 0.5
Using predicted Untaken, additional CPI = 0.1 + 0.15 + 0 = 0.25 Choice (C)

**27.** Clock cycles per second used for polling a mouse
= 20 × 200 = 4000
Fraction of processor cycles used for polling

$$= \frac{4000}{60 \times 10^6} \times 100 = 0.0067\%$$

Rate of polling for hard disk $\dfrac{\frac{1\text{ MB}}{\text{sec}}}{4\text{ bytes per access}}$

= 262144 polling access per second
Clock cycles = 262144 × 200 = 52428800
∴ Fraction of CPU cycles

$$= \frac{52428800}{60 \times 10^6} \times 100 \times 100 = 87.4\%$$ Choice (B)

**28.** Double-operand instruction format will be

| Opcode | Src data | Dest Data |
|---|---|---|

Data will consist both mode and register field.
There are 8-registers, so 3-bits required. For 15-different modes 4-bits required.
∴ Each data field requires 7-bits.
Now we need to find opcode size.

| | Instruction | Single/ no-operand | Double-operand |
|---|---|---|---|
| Arithmetic | 15 | $3\left(\frac{20}{100} \times 15\right)$ | 12 |
| Logic | 10 | 6 | 4 |
| Data-moving | 24 | 12 | 12 |
| Branch | 6 | 3 | 3 |
| Control | 5 | 3 | 2 |

∴ Total double operand instructions = 12 + 4 + 12 + 3 + 2 = 33

6-bits required for double operand. (∵ $2^6 > 33$).
∴ Minimum size of CPU's instruction word = 6 + 7 + 7 = 20-bits. Choice (B)

**29.** Given micro-operations specifies Branch-and-save address. The address part in instruction is extracted. Next instruction address is saved to memory location and execution continues at location (PC) + I. Choice (C)

**30.** Polling is same as programmed-I/O
Number of cycles required for polling
$= 1 \times 10^9 \times 0.02 \times 10^{-3} = 20000$
Before processing the data it will be
20000 – 1000 = 19000 cycles
Each polling iteration takes 80 cycles.
∴ Number of polls required for one byte $= \frac{19000}{80}$
= 237.5 polls ≃ 238 polls
Cycles spend on polling = 238 × 80 = 19040 cycles
∴ Each byte thus take 19040 + 1000 = 20040 cycles
To read 2000 bytes it requires
2000 × 20040 = 40080000
= 40080 K cycles Choice (C)

**31.** Each byte takes 200 + 1000 = 1200 cycles
i.e., 1200 cycles to process a byte.
For 2000 bytes it will required
1200 × 2000 = 2400000 cycles
Number of cycles saved using interrupt method
= 40080000 – 2400000 = 37680 K cycles Choice (D)

**32.** Cache size = 64 KB = $2^{16}$ B
Block size = 128-Bytes = $2^7$ B
Array size = 512 × 512
Each element requires 8 bytes.
Memory required to store $X$ is $512 \times 512 \times 8 = 2^{21}$ Bytes.
Number of Blocks required to store $2^{21}$ Bytes

$$= \frac{2^{21}}{2^7} = 2^{14} = 16384$$

∴ Number of cache misses = 16384 Choice (B)

**33.** Time required to execute given program on a sequential processor = 7 × 5 = 35

On a pipelined processor the execution sequence is shown below:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I1 | IF | ID | OF | IE | IS | | | | | | | | | | | |
| I2 | | IF | ID | OF | IE | IS | | | | | | | | | | |
| I3 | | | IF | ID | | | OF | IE | IS | | | | | | | |
| I4 | | | | IF | ID | | | | | OF | IE | IS | | | | |
| I5 | | | | | IF | ID | | | | | OF | IE | IS | | | |
| I6 | | | | | | IF | ID | | | | | | OF | IE | IS | |
| I7 | | | | | | | IF | ID | | | | | | OF | IE | IS |

∴ Total number of clock cycles required = 16

∴ Speed-up

$$= \frac{\text{Time required on sequential processor}}{\text{Time required on pipelined processor}} = \frac{35}{16}$$

= 2.1875 Choice (C)

**34.** Array is $A[1024]$

Array requires 1024 × 2 bytes.

Cache size = 4 KB = $2^{12}$ B

Line size = 8 bytes

Number of lines $= \frac{2^{12}}{2^3} = 2^9$ lines

Array requires 2048 bytes

i.e., $\frac{2048}{8}$ = 256 blocks. As number of lines is greater than the blocks required for array storage, there will be no conflict with element $A[0]$.

Choice (D)

**35.** As the array requires 2048 bytes i.e., $\frac{2048}{8} = 256$ blocks, 256 misses will be there. Choice (B)