SLOT: A2+TA2

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CONTINUOUS ASSESSMENT TEST - II
WINTER SEMESTER 2025-2026

Programme Name & Branch : **BCE, BAI, BCB, BCI, BCT, BCT, BDS, BKT, BME, BEL, BYB**
Course Code and Course Name : **BCSE205L & Computer Architecture and Organization**
Class Number(s) : **Common to All**
Date of Examination : **15-03-2026**
Exam Duration : **90 minutes** **Maximum Marks: 50**

General instruction(s):

- Answer All Questions

Q. No	Question	M

1. A processor executes a program in 6 seconds. The processor operates at a clock rate of 3 GHz. The instruction mix of the program is as follows:

Instruction Type	CPI	Percentage of Instructions
Arithmetic	1	35%
Memory	3	30%
Branch	4	35%

Assume there are no pipeline stalls other than those reflected in the CPI values.

Calculate the following:

a) The average CPI of the program.
b) The total number of clock cycles executed.
c) The total number of instructions executed.
d) The MIPS rating of the processor during this execution.

Further, assume that after compiler optimization, the branch instruction percentage reduces to 15% (with arithmetic increasing to 50%), and the new average CPI becomes 2.2.

e) Calculate the new execution time.
f) Determine the percentage improvement in performance after optimization.

10

Answer

a) 2.65
b) 1.8×10^{10}
c) 6.79×10^{9}
d) 1132.08
e) 4.98 s
f) 20.45%

Solution

To solve for the processor's performance metrics, we first identify the given parameters:

- Execution Time (T_{exec}) = 6 s
- Clock Rate (f) = 3 GHz $= 3 \times 10^{9}$ Hz
- Instruction Mix:
    - Arithmetic: $\text{CPI}_1 = 1, P_1 = 35\% = 0.35$
    - Memory: $\text{CPI}_2 = 3, P_2 = 30\% = 0.30$

- Branch: $\text{CPI}_3 = 4, P_3 = 35\% = 0.35$

a) Average CPI of the program The **average CPI** (CPI_{avg}) is the weighted sum of the CPIs for each instruction type:

$$\begin{aligned}\text{CPI}_{\text{avg}} &= \sum_{i=1}^{n} (\text{CPI}_i \times P_i) \\ &= (1 \times 0.35) + (3 \times 0.30) + (4 \times 0.35) \\ &= 0.35 + 0.90 + 1.40 \\ &= 2.65\end{aligned}$$

b) Total number of clock cycles executed The total number of **clock cycles** (C) is the product of the execution time and the clock rate:

$$\begin{aligned}C &= T_{\text{exec}} \times f \\ &= 6\,\text{s} \times 3 \times 10^9\,\text{Hz} \\ &= 1.8 \times 10^{10}\,\text{cycles}\end{aligned}$$

c) Total number of instructions executed The **instruction count** (IC) is derived from the total cycles and the average CPI:

$$\begin{aligned}IC &= \frac{C}{\text{CPI}_{\text{avg}}} \\ &= \frac{1.8 \times 10^{10}}{2.65} \\ &\approx 6{,}792{,}452{,}830\ \text{instructions}\end{aligned}$$

d) MIPS rating of the processor The **MIPS** (Millions of Instructions Per Second) rating is calculated as:

$$\begin{aligned}\text{MIPS} &= \frac{IC}{T_{\text{exec}} \times 10^6} = \frac{f}{\text{CPI}_{\text{avg}} \times 10^6} \\ &= \frac{3 \times 10^9}{2.65 \times 10^6} \\ &= \frac{3000}{2.65} \\ &\approx 1132.08\end{aligned}$$

e) New execution time After optimization, the new average CPI is given as $\text{CPI}_{\text{new}} = 2.2$. Assuming the total instruction count (IC) remains constant for the task:

$$
\begin{aligned}
T_{\text{new}} &= \frac{IC \times \text{CPI}_{\text{new}}}{f} \\
&= \frac{6{,}792{,}452{,}830 \times 2.2}{3 \times 10^9} \\
&= \frac{1.8 \times 10^{10}}{2.65} \times \frac{2.2}{3 \times 10^9} \\
&= \frac{6 \times 2.2}{2.65} \\
&\approx 4.98\,\text{s}
\end{aligned}
$$

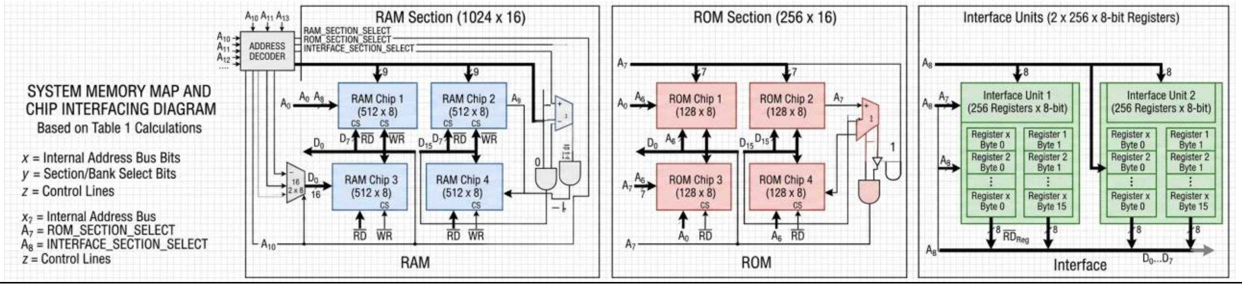
f) Percentage improvement in performance is defined as the reciprocal of execution time ($P = 1/T$). The percentage improvement is:

$$
\begin{aligned}
\text{Improvement} &= \left(\frac{P_{\text{new}} - P_{\text{old}}}{P_{\text{old}}}\right) \times 100\% \\
&= \left(\frac{T_{\text{old}}}{T_{\text{new}}} - 1\right) \times 100\% \\
&= \left(\frac{6}{4.9811} - 1\right) \times 100\% \\
&= \left(\frac{2.65}{2.2} - 1\right) \times 100\% \\
&\approx 20.45\%
\end{aligned}
$$

2. A computer employs RAM chips of 512 x 8 and ROM chips of 128 x 8. The computer system needs 1024 x 16 of RAM, 256 x 16 of ROM, and two interface units with 256 registers each. 10
   a. Compute total number of decoders are needed for the above system?
   b. Design a memory-address map for the above system.
   c. Show the chip layout for the above design.

Solution:

S. No.	Memory	N × W (chip)	N¹ × W¹ (required)	p	q	p × q (total chips)	x (addr lines)	y (sel lines)	z (CS lines)	Total
1	RAM	512 × 8	1024 × 16	2	2	4	9	2	2	13
2	ROM	128 × 8	256 × 16	2	2	4	7	2	2	11
3	Interface	256	512	2	1	2	8	1	2	11



Component	Address Range (Hex)	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
RAM Set 1	$000 - 1FF$	0	0	x	x	x	x	x	x	x	x	x
RAM Set 2	$200 - 3FF$	0	1	x	x	x	x	x	x	x	x	x
ROM Set 1	$400 - 47F$	1	0	0	0	x	x	x	x	x	x	x
ROM Set 2	$480 - 4FF$	1	0	0	1	x	x	x	x	x	x	x
Interface 1	$500 - 5FF$	1	0	1	x	x	x	x	x	x	x	x
Interface 2	$600 - 6FF$	1	1	0	x	x	x	x	x	x	x	x

3. a) In cache memory management policies, interpret the following Main Memory block addresses to the cache for FIFO and LRU. Considering a fully associative mapping cache of size 256 B with block size 64 bytes. Show each step in-detail for hit and miss. **5**

1, 2, 3, 4, 5, 3, 4, 1, 6, 7, 8, 7, 8, 9, 7, 8, 9, 5, 4, 5, 4, 2

Summarize details in the following prescribed format:

Algorithm	**FIFO**	**LRU**
No. of Misses	13	13
No. of Hits	9	9
Total no. of Reference	22	22
Miss Ratio	13/22 = 0.591	13/22 = 0.591
Hit Ratio	9/22 = 0.409	9/22 = 0.409

Compare the results and write your observations.

b) A two way set associative cache memory unit with a capacity of 16 KB is built using a block size of 8 words. The word length is 32 bits. The memory address space is 4 GB. Find the number of bits in the TAG and SET fields. **5**

1. Address Space and Total Address Bits

The memory address space is 4 GB.

$$\text{Address Space} = 4 \times 2^{30} \text{ bytes} = 2^{32} \text{ bytes}$$

$$\text{Total Address Bits} = 32 \text{ bits}$$

2. Block Offset Bits

Each block contains 8 words8words, and each word is 32 bits32bits (4 bytes).

$$\text{Block Size} = 8\ \text{words} \times 4\ \text{bytes/word} = 32\ \text{bytes}$$
$$\text{Block Size} = 2^5\ \text{bytes}$$
$$\text{Block Offset Bits} = 5\ \text{bits}$$

3. Set Index Bits

The cache capacity is 16 KB16KB, and it is 22-way set associative.

- **Total number of blocks in cache:**

$$N_{\text{blocks}} = \frac{\text{Cache Capacity}}{\text{Block Size}} = \frac{16 \times 1024\ \text{bytes}}{32\ \text{bytes}} = \frac{2^{14}}{2^5} = 2^9 = 512\ \text{blocks}$$

Number of sets (SS):

Since it is 22-way set associative, each set contains 22 blocks.

$$S = \frac{N_{\text{blocks}}}{\text{Associativity}} = \frac{512}{2} = 256\ \text{sets}$$
$$S = 2^8\ \text{sets}$$
$$\text{Set Index Bits} = 8\ \text{bits}$$

4. Tag Bits

The total address bits are divided into Tag, Set Index, and Block Offset.

$$\begin{aligned} \text{Tag Bits} &= \text{Total Bits} - (\text{Set Index Bits} + \text{Block Offset Bits}) \\ &= 32 - (8 + 5) \\ &= 32 - 13 \\ &= 19\ \text{bits} \end{aligned}$$

4. a) Asume in a program variable *j* has 32-bit hex value 0x07654321 in memory starting at address 0x2000. Draw the internal byte storage representations if program is executed in big endian and little-endian architectures. Briefly explain one advantage of Little Endian over Big Endian. 5

Solution:

To represent the 32-bit hex value 0x076543210x07654321 in memory, we first break it into its constituent bytes: 0707, 6565, 4343, and 2121.

In **Big-Endian**, the most significant byte (MSB) is stored at the lowest memory address.

Address	Value
0x2000	07
0x2001	65
0x2002	43
0x2003	21

In **Little-Endian**, the least significant byte (LSB) is stored at the lowest memory address.

Address	Value
0x2000	21
0x2001	43
0x2002	65
0x2003	07

A primary advantage of **Little-Endian** is the ease of **type casting** (e.g., converting a 32-bit integer to a 16-bit or 8-bit integer).

- Since the least significant bytes are stored at the starting address, the CPU can read a smaller data type from the same starting address without performing any address arithmetic or pointer offsets.
- For example, to read the 8-bit value of the variable above, the CPU simply reads the byte at 0x20000x2000 (2121), which is the same starting address used for the full 32-bit word.

b) Identify the I/O techniques used in the following scenario and justify the answer. Meera sends a document to the printer and then continues editing another file. The CPU doesn't keep checking the printer's status. Instead, when the printer finishes printing a page, it sends a signal to the CPU, saying, "I'm ready for more data." The CPU then pauses its current work briefly to send the next chunk. 5

The I/O technique described in this scenario is **Interrupt-driven I/O**.

5\. a) Consider a computer system with DMA support. The DMA module transfers one 8-bit character in one CPU cycle from a device to memory through cycle stealing at regular intervals. If the DMA clock frequency is 2 MHz, calculate the data transfer rate of the device in bits per second. 4

Solution:

- **DMA Clock Frequency (f):** $2\,\text{MHz} = 2 \times 10^6\,\text{cycles} \cdot \text{s}^{-1}$
- **Data per cycle:** $1\,\text{character} = 8\,\text{bits}$
- **Transfer mechanism:** Cycle stealing (one character is transferred every cycle at regular intervals).

The data transfer rate (RR) is the product of the clock frequency and the amount of data transferred per clock cycle:

$$\begin{aligned} R &= f \times \text{bits per cycle} \\ &= (2 \times 10^6\,\text{cycles} \cdot \text{s}^{-1}) \times (8\,\text{bits} \cdot \text{cycle}^{-1}) \\ &= 16 \times 10^6\,\text{bits} \cdot \text{s}^{-1} \end{aligned}$$

The data transfer rate is 16 Mbps16Mbps (Megabits per second): 16000000 bps

b) In a research workstation, the CPU is engaged in executing a real-time simulation that requires continuous responsiveness while simultaneously handling background data transfers from multiple devices. Two DMA requests occur in close succession: one from a satellite sensor module that streams data at a sustained rate of 500 KB/sec, sending packets of 4 KB each into main memory, and another from a solid-state storage unit that transfers a 10 MB dataset into memory. During the sensor transfer, the CPU continues execution but experiences intermittent slowdowns, whereas during the storage transfer, the CPU undergoes longer pauses and is unable to access the bus until large chunks of data are fully moved. 6

From these observations, analyze the CPU behaviour and the characteristics of each transfer to determine the corresponding DMA modes of operation, providing justification in terms of bus control duration, CPU responsiveness, and the nature of the data being transferred.

Transfer 1 — Satellite Sensor Module → **Cycle Stealing Mode**

Why this is Cycle Stealing:

In cycle stealing, the DMA controller does not seize the bus for an extended period. Instead, it *steals* individual bus cycles from the CPU — one memory cycle at a time — inserts a single word or small unit of data into memory, then immediately releases the bus back to the CPU. The CPU is not halted outright; it is only delayed for the duration of that stolen cycle, after which it resumes from exactly where it was interrupted.

Transfer 2 — Solid-State Storage Unit → **Burst Mode**

Why this is Burst Mode:

In burst mode, the DMA controller asserts bus control and *holds* it for the entire duration of a contiguous block transfer. The CPU is completely locked out of the bus — unable to fetch instructions, access memory-mapped peripherals, or perform any bus transaction — until the DMA controller releases control after moving the full block. This produces the "longer pauses" and the inability to access the bus described in the scenario..

****