(*) Ans (1)

(A) → A computer system consists of several components working together to execute task.

(A) → central processing unit (CPU): The CPU, or processor. It the brain of the computer. Responsible for executing Instruction. It include the (ALU) → Arithmetic logic units for calculations the control unit (CU) for directing operations. and register for temporary holding data :-

(*) memory:

* primary memory (RAM): store data and Instruction while task are executed.
* cache memory: fast memory located close to or within the CPU, stores frequently accessed data to speed up processing.
* Secondary memory: non-volatile storage like hard drive and SSDs, used for long-term storage. Input/output (I/O) Devices
* Input Devices: Allows user to provide data (eg. keyboard)
* output Devices: Display result to user (eg: monitors, printers).
* System Bus: A communication BUS system that connect the CPU, memory, and I/O Devices, allowing data transfer between them.

(8)

(*) Data Bus: Transfer data.

(*) Address Bus: carries memory address.

(*) control Bus: manage control signals.

(*) To execute an Instruction, the cpu perform a fetch-decode-execute cycle. the control unit fetches an Instruction from memory decode it to understand the required operation, and the ALU executes it. the result is store in memory or a register.

(*) computer Architecture refers to the high-level design, including the Instruction set and data formats, while computer organization deals with physical components and data pathways. Architectures define the system capability (the "what"), whereas organization defines how these capabilities (the "what") whereas organization define how to this capabilities (the "what) (and how) together they determines a computer's performance and functionality.

(*) Both computer architecture and organization crucial for understanding how computer operate. with architecture providing the "what" and organization providing "how".

Cpu execution time

$$T = \frac{\text{Instruction count} \times \text{C.P.I}}{\text{clock speed}}$$

clock speed $= 2.5$ GHz
$= 2.5 \times 10^9$

Instruction count $=$ 10 billion
$= 10 \times 10^9$

CPI $= 1.5$ cycles per Ins

$T = 6$ sec

CPU execution time In 6 sec

$$\text{MIPS} = \frac{(\text{million Instruction In second})}{\text{Execution time} \times 10^6}$$

~~MIPS = (million~~

$$\text{MIPS} = \frac{10 \times 10^9}{6 \times 10^6}$$

$$\text{MIPS} = (1666.67)$$

→ Influence of Factors

1) clock Speed : If the clock speed is increased the cpu execution time will decrease

2) CPI : Reducing the CPI will lower the CPU execution time leading to faster program execution

3) Instruction count

Instruction are executed to complete a task the overall execution time will decrease improving the system's performance ☺

Q3. ⇒ Assembly language is a low-level programming language that use mnemonic code to represent machine-level instruction, allowing programmer to write programs closely tied to hardware. Unlike high-level language, assembly provide direct control over hardware giving programmer more efficiency and precision but requiring more detailed knowledge of the system architecture.

(*) In assembly language each instruction typically corresponding to a single machine operation. Such as loading data into register or performing an arithmetic calculation. Instruction are written in form of mnemonics (like ADD, SUB, MOV) that directly translate into machine code, which the CPU can execute.

→ Example program: Basic arithmetic operations.

* consider a program that perform addition and subtraction on two number. with and without carry.

→ MOV AX, 5 : load 5 into register
→ MOV BX, 3 : load 3 into register
: Addition without carry
ADD AX, BX ; AX = AX + BX (AX = 5+3=8)
: Subtraction with out carry

(11)

```
mov  Ax, 10   ; Load 10 Into Ax
mov  Bx, 2    ; Load 2 Into Bx
SUB  Ax, Bx   ; Ax = Ax - Bx (Ax = 10-2 = 8)
```

; Addition with carry

```
CLC           ; clear carry flag
ADD  Ax, Bx   ; Ax = Ax + Bx (If there is carry,
it is Added to result)
```

; Subtraction with borrow

```
STC           ; set carry flag (Simulate a borrow)
SBB  Ax, Bx   ; Ax = Ax - Bx - CF (Subtract with
Borrow)
```

Programmer structs and two Instruction.

Mov → Transffer Data Between register or from memory to a register.

ADD, SUB : Perform Addition and Subtraction.

CLC, STC : clear and set the carry flag effects operation on rely on carry or Borrow.

(*) Interaction with hardware :

(*) Each instruction interact with cpu registers and flag directly allowing control over operation and status. Assembly instruction execute step by step manner on hardware, with the cpu performing specified tasks at the hardware level. this close cpu perform this close interaction enable efficient and optimized code execution. critical in performance sensitive application like embedded.

4

Try Floating - point Representation Is a method used (7) In computing to Represent real number that can contain fraction (decimals). the IEEE 754 standard define format for Floating point Representation Including Single - precious (32 - bit) and double precious (64 Bits) formats.

1) step 1 convert Integer part to Binary

* 1259 /2 = 629 Remainder 1
629 /2 = 314 " 1
314 /2 = 157 " 0
157 /2 = 78 " 1
78 /2 = 39 " 0
39 /2 = 19 " 1
19 /2 = 9 " 1
9 /2 = 4 " 1
4 /2 = 2 " 0
2 /2 = 1 " 0
1 /2 = 0 " 1

* 1259 In Binary is 10011101011

) Convert the fractional part (0.125) to Binary. (8)

1) $0.125 \times 2 = 0.25 \rightarrow 0$

2) $0.25 \times 2 = 0.5 \rightarrow 0$

3) $0.5 \times 2 = 1.0 \rightarrow 1$

* 0.125 In binary Is 0.001

3) Combine both of them

~~159.125~~ Is ~~1001111011.001~~

10011101011.001

(*) Normalize the Binary part

159.125(10) =

= 10011101011.001(2) =

= $10011101011.001_{(2)} \times 2^0$ =

= $1.0011101011001_{(2)} \times 2^{10}$

(*) Sign 0 (a positive number)

Exponent (unadjusted): 10

mantissa (not normalize)

1.0011101011001

(*) → Adjust the exponent

Exponents (adjusted) =

Exponent (unadjusted) + $2^{(8-1)} - 1 = 0$

$10 + 2^{(8-1)} - 1 = 1$

$(10 + 127)_{10} =$

$137_{(10)}$

→ division = quotient + remainder;

$137 \div 2 = 68 + 1$;

$68 \div 2 = 34 + 0$;

$34 \div 2 = 17 + 0$

$17 \div 2 = 8 + 1$

$8 \div 2 = 4 + 0$

$4 \div 2 = 2 + 0$

$2 \div 2 = 1 + 0$;

$1 \div 2 = 0 + 1$;

$137_{(10)} = 10001001_{(2)}$

Normalised the mantissa

$0.10001001 \times 2^{8} \Rightarrow 0.10001001 \times 2^{(135-127)}$