1) The main structural components of a single processor computer are:-

(i) Central Processing Unit (CPU):-

* The CPU is the brain of the computer. It performs essential tasks such as data processing, storage and communication b/w input/output devices
* Key functions:
  - Arithmetic Logic Unit (ALU): Performs arithmatic and logic operations (e.g: addition, comparision)
  - Processor Registers: Store operands for the ALU and hold results from ALU operations.
  - Control Unit: Orchestrates fetching, decoding and executing instructions by coordinating the ALU, registers, and other components.

(ii) Main Memory (RAM):-

* Also known as primary storage, the main memory stores instructions, data & intermediate results.
* Responsible for transferring information to other units when needed.
* Its size impacts system speed, power, and performance.

(iii) Input/Output (I/O) Devices:-

* These devices facilitate communication between the computer and the external coorld.
* Examples include keyboards, mice, monitors, printers and network interfaces.

(iv) System Interconnection:-

* The system bus connects various components within the computer, allowing them to exchange data and instructions.
* It ensures seamless communication between the CPU, memory and I/O devices.

2) i)

Micro processor	Micro Controller
- It is the basic unit of computer system.	- It is the basic unit of embeded system.
- The circuit is larger	- The circuit is very compact.
- Cost is more.	- Cost is less
- Power saving is less	- Power saving is more.
- Ex - Desktop, Computers	- Ex - AC, Micro wave Oven

ii)

Computer Architecture	Computer Organization
- Computer architecture refers to those attributes of a system that is visible to a programmer.	- Computer organization refers to the operational units & their interconnection that realise the architectural specification
- Architecture is always decided first.	- It is decided after the architecture has been fixed
- It deals with high level design issues	- It deals with low level design issues.
- Architecture generally refers to the hardware models.	- Organization refers to the actual performance of a processor

iii)

Embedded System	Deeply embedded System
- Integrated computing device performing specific functions	→ Specialised embedded system tightly integrated into hardware after with limited resources.

3) Usual computing services after a variety of resources and functionalities to users and business. Here is a list of some common cloud computing services

(i) IaaS :- Infrastructure as a Service

IaaS offers the customer processing, storage & other fundamental computing resources

Ex:- Amazon Elastic Compute Cloud

(ii) PaaS :- Platform as a Service

Cloud provides service to customers in the form of a platform on which the customer's applications can run. It provides a no. of developments tools such as programming languages run time environments.

Ex:- Google App Engine

iii) SaaS :- Software as a Service

Cloud provides service to costumers in the form of software, running on and accesible in the cloud. The applications are accessible through web brocoser.

Ex- Gmail, Sales Force, Google Email Service.

4) With the following methods, we can increase the microprocessor speed.

(i) Pipelining :-

Pipelinging enables the processor to coork simultaneously on multiple operations at the same time by performing the diff^{n} phases of an instruction execution cycle.

(ii) Branch Prediction :-

The processor looks ahead in the instruction code faced from memory & predicts branches or groups of instructions are likly to be processed next.

(iii) Data Flow Analysis :-

The processor analyzers which instructions are dependents on each others result on data to create an optimised schedule instruction

(iv) Superscalar execution :-

This is the ability of the processor to execute more than one instruction in a single clock cycle, so that multiple parallel pipelines can be executed.

(v) Speculative Analysis :-

With the help of branch prediction & Data flow analysis some of the processor can theoretically execute information prior to their actual appearance

(vi) By increasing cache memory

5) Effective CPI :-

Machine A $(CPI)_A = \dfrac{\left(\frac{50}{100}\times 2\right)+\left(\frac{15}{100}\times 3\right)+\left(\frac{15}{100}\times 4\right)+\left(\frac{20}{100}\times 2\right)}{(50+15+15+20)/100}$

$= 1 + 0.45 + 0.6 + 0.4 = 2.45$

Machine B $(CPI)_B = \dfrac{\left(\frac{65}{100}\times 1\right)+\left(\frac{15}{100}\times 4\right)+\left(\frac{10}{100}\times 3\right)+\left(\frac{10}{100}\times 2\right)}{\frac{65}{100}+\frac{15}{100}+\frac{10}{100}+\frac{10}{100}}$

$= 1.75$

Execution Time:-

For Machine A $= \dfrac{100000 \times 2.45}{400\times 10^6} = \dfrac{2.45}{4000} = 0.0006125$ seconds

For Machine B $= \dfrac{1\times 10^5 \times 1.75}{400\times 10^6} = \dfrac{1.75}{4000} = 0.0004375$ seconds

MIPS Rate:-

For machine A $= \dfrac{400\times 10^6}{2.45\times 10^6} = 163.26$ seconds

For machine B $= \dfrac{400\times 10^6}{1.75\times 10^6} = 228.57$ seconds

6) Given, $L = 3$

$\lambda = 6$

By using Little's Law, $(L = \lambda W)$

$W = \dfrac{L}{\lambda} = \dfrac{3}{6} = [0.5]$ hrs

7) From Amdhal's Law,

Speed up $= \dfrac{1}{(1-f)+\frac{f}{N}}$

Given, Speed up $= 2.25$,

$N = 15$

By using Amdhal's Law,

$2.25 = \dfrac{1}{(1-f)+\frac{f}{15}} \Rightarrow 2.25\left(1-f+\frac{f}{15}\right) = 1$

$\Rightarrow 2.25 - 2.25f + \dfrac{2.25f}{15} = 1$

$\Rightarrow \dfrac{33.75 - 33.75f + 2.25f}{15} = 1$

$\Rightarrow -31.25f = -18.75$

$\Rightarrow f = 18.75/31.25 = 0.6 \simeq 60\%$

8) Given $I_c = 10,000,000$

MIPS value :-

Prog-1

Comp A = $\frac{10,000,000}{50 \times 10^6} = \underline{0.2}$

Comp B = $\frac{10,000,000}{80 \times 10^6} = \underline{0.5}$

Comp C = $\frac{10,000,000}{10 \times 10^6} = \underline{1}$

Program-2

Comp A = $\frac{10,000,000}{100 \times 10^6} = \underline{0.1}$

Comp B = $\frac{10,000,000}{200 \times 10^6} = \underline{0.05}$

Comp C = $\frac{10,000,000}{40 \times 10^6} = \underline{0.25}$

Arithmatic Mean

Comp A = $\frac{0.2+0.1}{2} = \underline{0.15}$

Comp B = $\frac{0.5+0.05}{2} = \underline{0.275}$

Comp C = $\frac{1+0.25}{2} = \underline{0.625}$

$\therefore$ Comp C > Comp B > Comp A

Harmonic Mean

Comp A = ~~$\sqrt{0.2 \times 0.1}$~~ = $\frac{2}{(\frac{1}{0.2})+(\frac{1}{0.1})} = \frac{2}{5+10} = \frac{2}{15} = \underline{0.33}$

Comp B = $\frac{2}{(\frac{1}{0.5})+(\frac{1}{0.05})} = \frac{2}{2+20} = \frac{2}{22} = \frac{1}{11} = \underline{0.0909}$

Comp C = $\frac{2}{(\frac{1}{0.25})+(\frac{1}{1})} = \frac{2}{4+1} = \frac{2}{5} = \underline{0.4}$

$\therefore$ Comp C > Comp A > Comp B

9) Using Amdhal's law, Speed up = $\frac{1}{(1-f)+f/Sup}$

Given, $f = 0.4$, $Sup = 2.3$

Speed up = $\frac{1}{(1-0.4)+0.4/2.3} = \frac{1}{0.6+0.1739} = \frac{1}{0.7739} = \underline{1.29}$

10) Some addressing modes of the 8086 microprocesior.

(i) Immediate Addressing Mode :-

* In immediate addressing mode, the operand is specified directly within the instruction.
* Ex- 'MOV AX, 1234H', moves the immediate value '1234H' into the AX register.

(ii) Register Addressing Mode :-

* In register addressing mode, the operand is stored in a register or is the contents of a register.
* Ex- 'MOV BX, AX', moves the contents of the AX register into the BX register.

(iii) Direct Addressing Mode :-

* In direct addressing mode, the operand is located in memory & the effective address is directly specified in the instruction.
* Ex - 'MOV AL, [2000H]' moves the byte at memory address 2000H into the AL register.
* ADDRESS {(10*DS)H + 5000H}

(iv) Indirect Addressing Mode :-

* In indirect addressing mode, the effective address of the operand is held in a register or a memory location.
* Ex - 'MOV AL, [BX]', moves the byte at memory address specified by the contents of the BX register into the AL register.
* ADDRESS {(10*DS)H + [BX]}

(v) Indexed Addressing Mode :-

* An index register is added to a base address to form the effective address of the operand.
* Ex - 'MOV AX, [SI]', moves the word at the memory address specified by the contents of the SI register into the AX register.
* ADDRESS {(10*DS)H + [SI]}

(vi) Base - Indexed Addressing:

* Both a base register and an index register are added to form the effective address.
* Ex - 'MOV AL, [BX+SI]' moves the byte at the memory address specified by the sum of the contents of the BX and SI registers in to the AL register.
* ADDRESS {(10*DS)H + [BX] + [SI]}

(viii) Relative Addressing :-

* The operand is specified as an offset relative to the current instruction point (IP) or relative to a segment register.

* Ex- MOV AX, [BP]1001H

II) i) General Purpose Registers

- AX (Accumulator): AX is a 16-bit register primarily used for arithmatic and data manipulation operations.

  Ex:- 'MOV AX, 1234H'

- BX (Base): BX is a 16-bit register commonly used as a base address pointer or for general data storage.

  Ex:- 'MOV BX, CX'

- CX (Count): CX is a 16-bit register used as a loop counter or for the counting operations.

  Ex:- 'Loop Label' decrements the CX register & becomes zero.

- DX (Data): DX is a 16-bit register used for various data operations, including I/O operations.

  Ex:- 'IN DX, AL'

ii) Index Registers

- SI (Source Index) and DI (Destination Index): These 16-bit index registers are used for string manipulation and memory copying operations.

  Ex:- 'MOVSB'

iii) Segment Registers:

- CS (Code Segment), DS (Data Segment), SS (Stack Segment) & ES (Extra Segment) These 16-bit segment registers store segment selectors used for memory addressing

  Ex:- 'MOV AX, [DS:SI]'

iv) Pointer Registers:

- SP (Stack Pointer) and BP (Base Pointer): These 16 bit registers are used for stack operations & addressing data on the stack.

  Ex:- 'MOV SP, 1000H'

v) Instruction Pointer (IP): IP is a 16-bit register that holds the offset of the next instruction to be executed within the current code segment

Ex:- After executing an instruction, IP automatically increments to point to the next instruction in memory.

12) (a) Assembly Language Program to Multiply 40H with 8H

```
; initialize the values
MOV AL, 40H ; Load 40H into AL
MOV BL, 8H  ; Load 8H into BL
;Multiply using logical instructions
MUL BL      ; Multiply AL by BL

; Result is stored in AX register
;AX = AL * BL

; Halt the program
HLT
```

(b) 'MOV ax, 23f0h': Moves the value '23F0H' into the AX register

'MOV bx, ax': Copies the value of AX into the BX register

'MOV [bx], ax': Stores the value of AX at the memory location pointed to by BX. So, the output memory location is '23F0H' and its content is '23F0H'

'MOV cx, 503fh': Moves the value '503FH' into the CX register.

'MOV ax, cx': Copies the value of CX into the AX register.

'Sub ax, [bx]': Subtracts the value at the memory location pointed to by Bx from AX. Since the content of the memory location pointed to by BX is '23F0H', the result of the subtraction would be 503FH - 23F0H = 2F4FH.

'Inc bx': Increments the value of BX register

'Inc bx': Increments the value of BX register again.

'MOV [bx], ax': Stores the value of AX at the memory location pointed to by BX. So, the output memory location is '23F1H', and its content is '2F4FH'.

'Hlt': Halt the program

∴ Memory location '23F1H' and content is '2F4FH'.