NAME: FARDOUSE LOIMAT JAHAN RUMPA

ID: 2219170041

Dept: CSE (E)

Batch: 17th

Course code: CSE 313

Course Title: Computer Architecture

Ans to the Qus No: 01 (a)

Ans: IBM z EC12 I/O Channel Subsystem structure

The z enterprise EC12 is IBM's latest Mainframe Computer offering (All the time of this writing). This system is based on the use of the zEC12 processor chip, which is a 5.5-GHz Multicore chip with six cores. The zEC12 architecture can have maximum of 101 processor chip with 6 cores.

[1] Channel Structure:

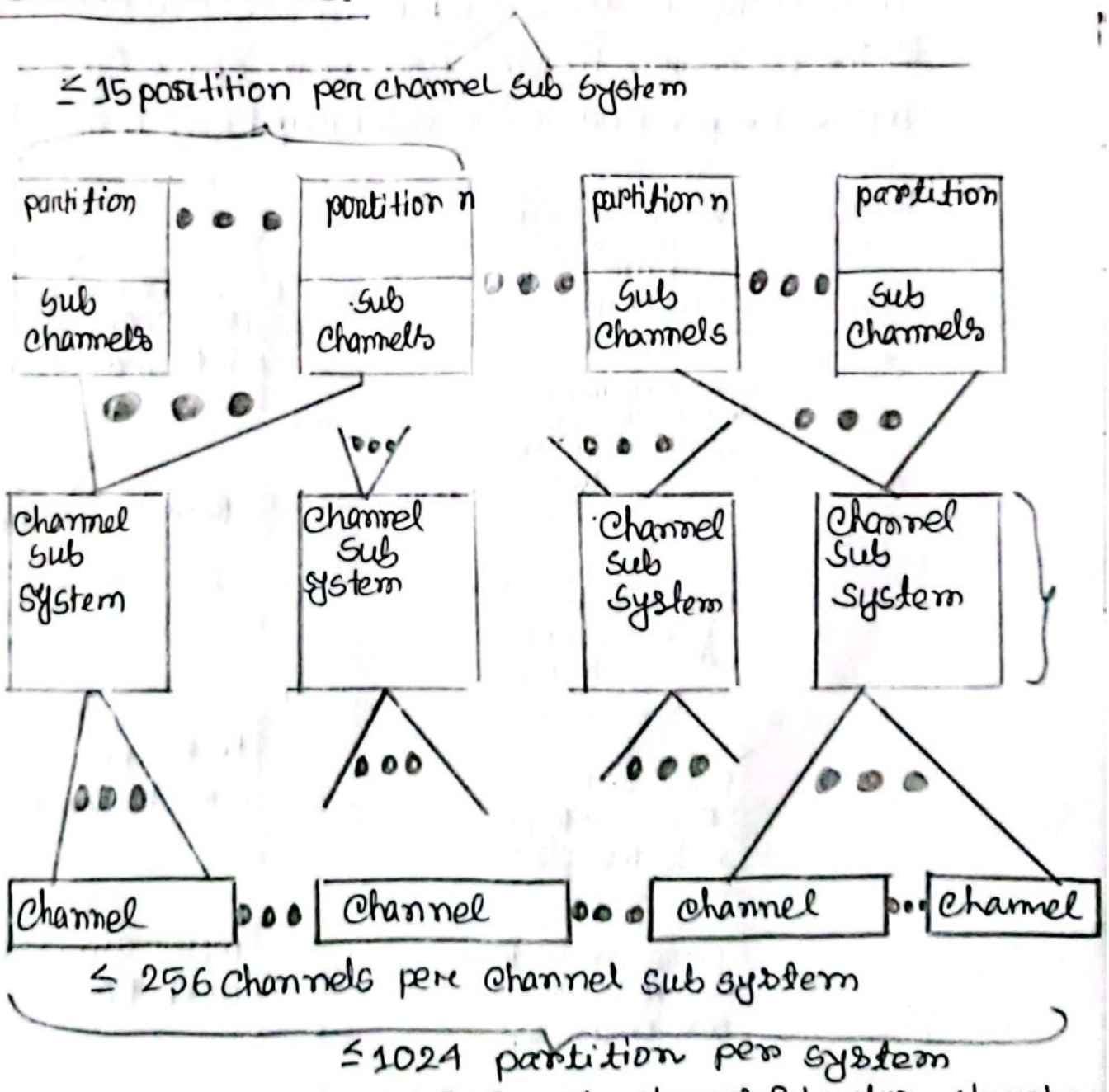


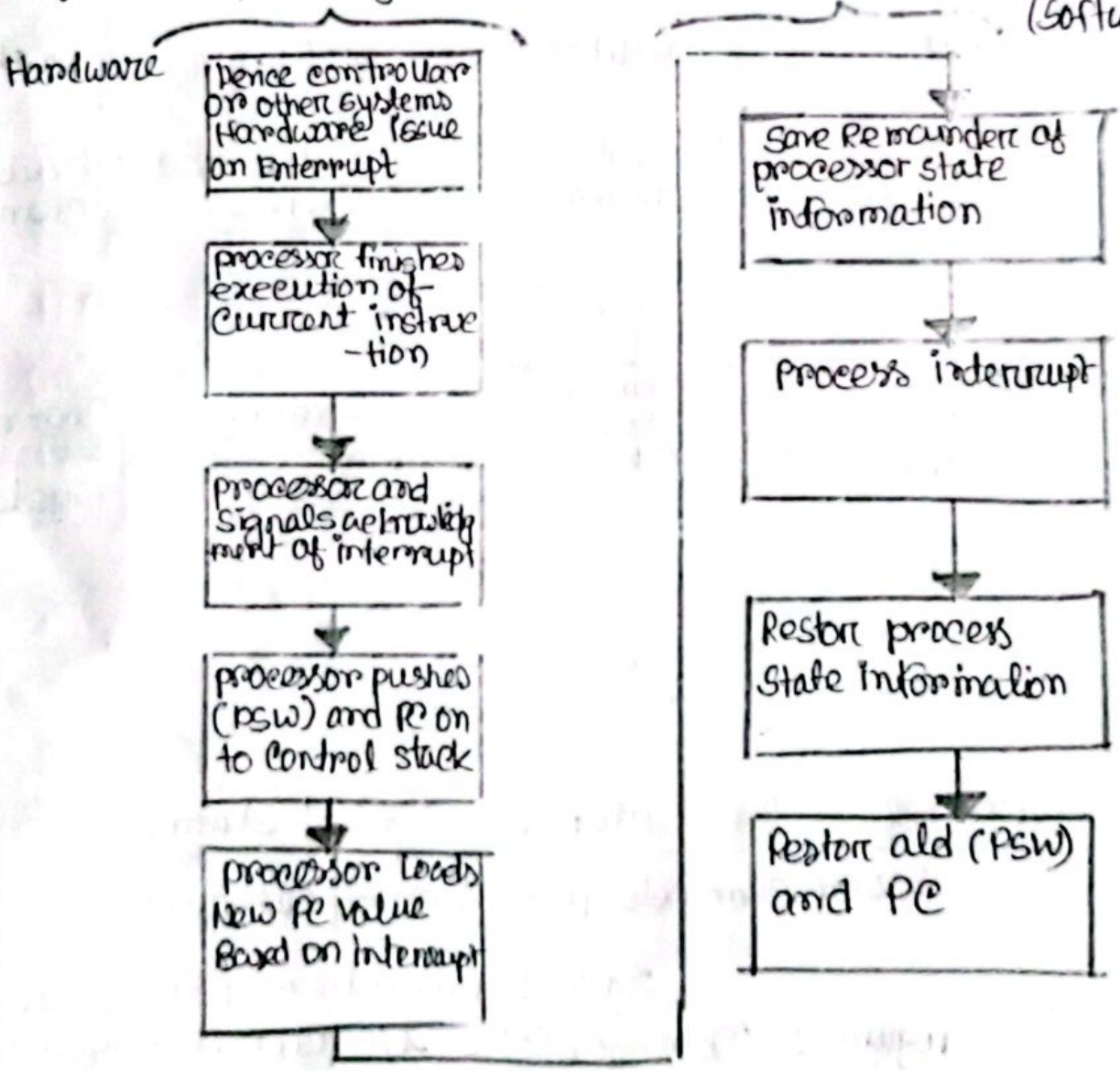
Figure: IBM z EC12 I/O channel Subsystem structure

Ans to the Qus No: 01(b)

Ans: **I/O Device completes an I/O operents sequence of Hardware:**

The Occurrence of an interrupt triggers a number of events, both, in the processore. Hardware And in software. Figure 7.6 shows atypical sequeeeal. when an 110 device complets an 110 operation. the following sequen of Hardware events occurs.

- The Device issues an interrupt Signal to the proceess
- The processor finishes execution of the current instruct before responding to the interrupt, as indicated in.1



The processor now checks to proper to transfer to control to the interrupt routine. To begin it needs to save information needed to resum the current program at the point of interrupt. The minimum information required is (a) the states of the processor which contained in a register called the program states word (PWS) and (b) the location of the next instruction to be executed which is contained in the program counter. This can be pushed on-to the system control stack

The processor Now loads the program counter with the entry location of the interrupt handling program that will respond to this interrupt. Depending, on the computer Architecture and Operating system design, there may be a single program on program for each type of interrupt. or on one program for each device and each type of interrupt. or if there is more then on interrupt handelling roution. the processor must determind which one to invoke. The information may have been include in the original interrupt signal. or the processor may have to issue a request to the device that issued the interrupt to get a response that contains the needed information.

At this point, the program Counter and (PSW) relating to the interrupted program have been saved on the system stack. However, there is other information that is coreided part of the "state" of the executing program. Inparticulan, the contents of the processon registers Need to be saved, because this values plus any other state information, need to be saved. Typically the interrupt handler.

The interrupt handler next process the interrupt. This include an examination of states information.

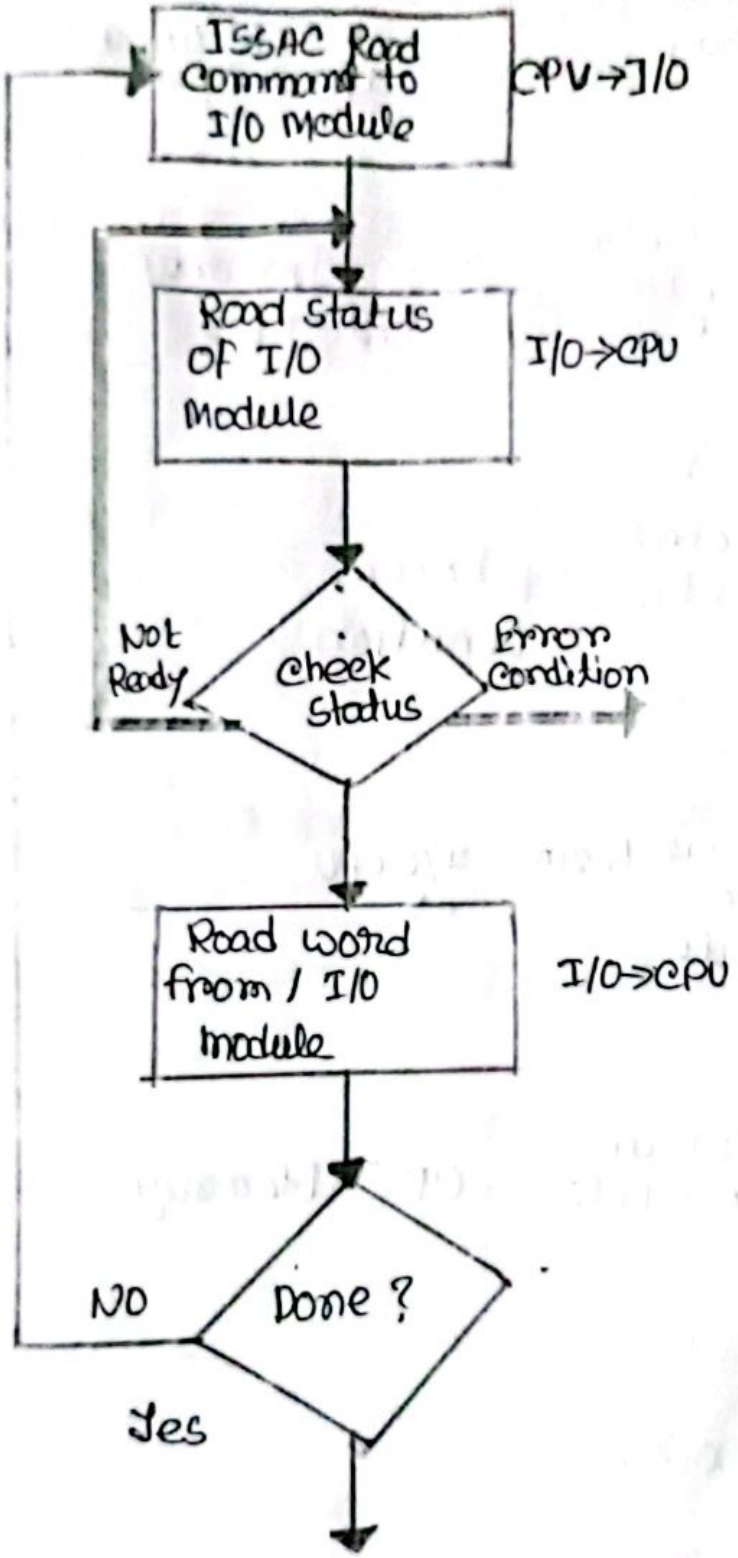
Relating to the I/O operation or other event that caus an interrupt. It may Also involve sending Additional Commands or Acknowledgment to the I/O Device.

- When interrupt processing is Completed the save request- register value are Retrieved from the stack and restor to the registor.
- The final act is to restore the (PSW) and program counter value from the stack. As result, the next instruction to be executed will be from the previously interrupted program. Note that if is important to save all the state information about the interrupted program for later resumption. This is because the interrupt is not a routin called from the program. Rather the interrupt can occure at any time and there for at any point in the Execution of a user program. It's occurance is unpredictable.

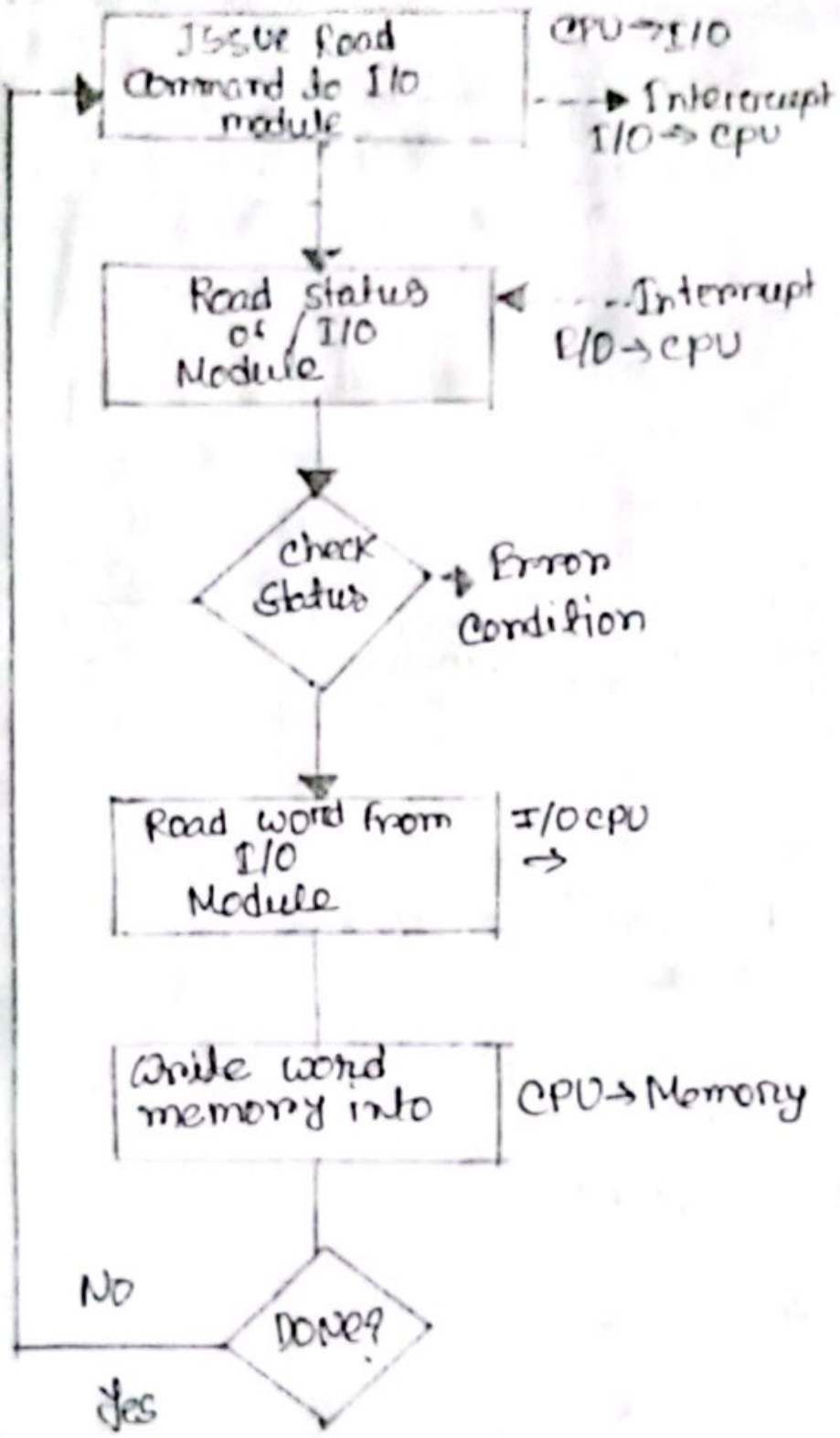
Indeed, as we will see in the next Chapter the two programs may not Have Anythings in Common and may belong to two different users.

Ans to the Que No : 01 (c)

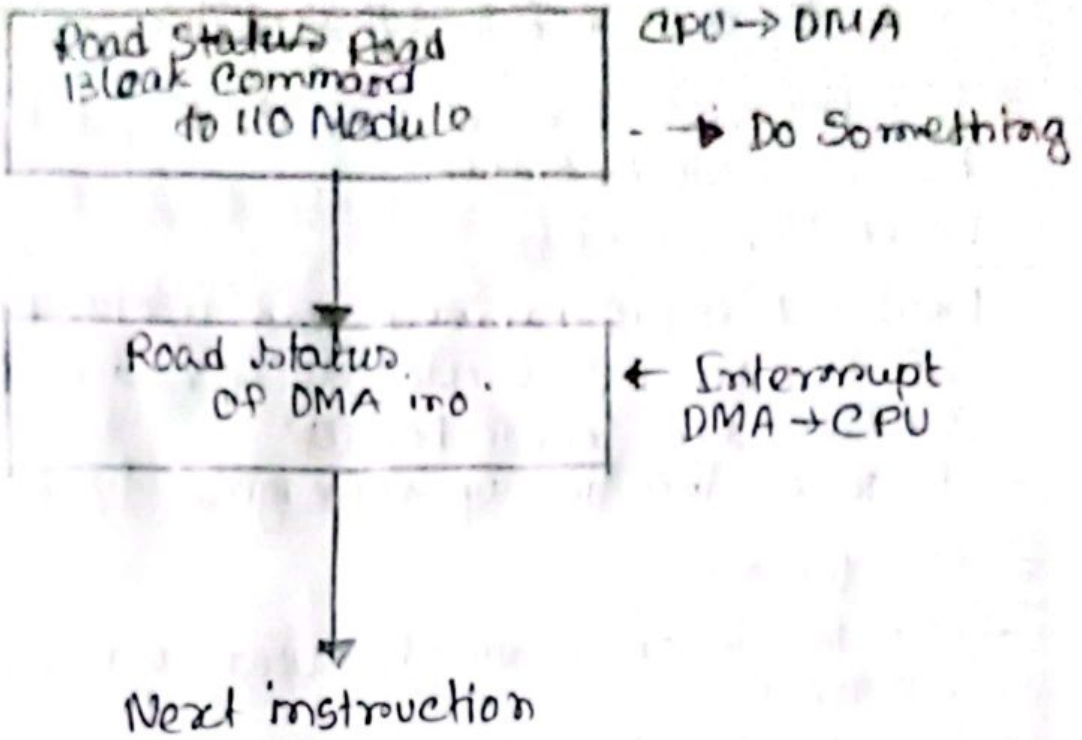
Ans: Three techniques for input of a block diagram:



(a) program I/O.



(b) Interrupt-Driven I/O



© Direct memory Access

Three techniques for input of block Data:

I/O instruction: processor views I/O operation in similar manner as memory operations Each Device is given a Unique identifier or Address. processor issue Commands containing Device address - I/O module must check Address line to see if the command if for itself.

⇒ I/O mapping: → memory - mapped I/O
→ Isolated - I/O

⇒ Interrupted - Driven I/O:

→ Over Comes the processor having to wait long periods of time for I/O Modules.

→ The processor does not have to repeatedly check the I/O module status.

⇒ Direct Memory Access:

Draw back of programmed and interrupt-Driven I/O.

→ I/O transfer rate limited to speed that processor can test and service Devices.

→ Processor tied up managing I/O transfer.

⇒ DMA Function:

→ DMA Module on system has used to mimic the processor.

→ DMA Module only used system bus when processor Doesn't need it.

→ DMA Module may temporarily force processor to suspend operation - cycle sta stealing.

Ans to the QUS NO: 02(a)

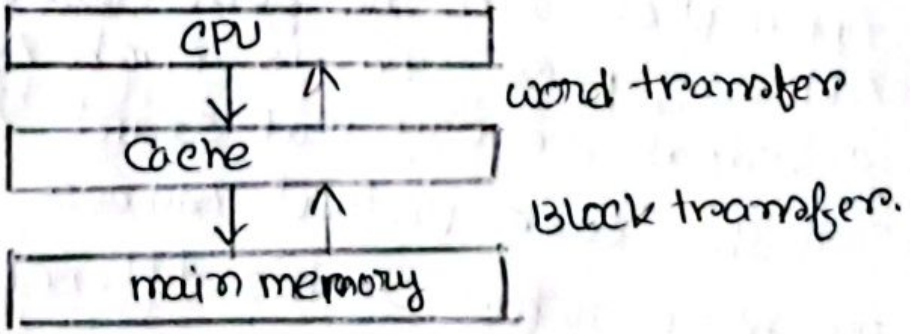
Ans: Cache memory:

Cache memory is a small-sized type of volatile computer memory that provides high speed data access to a processor. and stores frequently used computer programs, applications and data.

A temporary storage of memory, cache makes data retrieving easier and more efficient. It is the fastest memory in a computer, and it is typically integrated onto the motherboard and directly integrated embedded in the processor or main as random access memory (RAM)

main memory:

Main memory is the primary, internal workplace in the computer, commonly known as RAM (random access memory) specifications such as 4GB, 8GB, 12GB and 16GB almost always refer to the capacity of RAM. In contrast, disk or solid state storage capacities in a computer are typically 128GB or 256GB and higher. In a smartphone or tablet, solid state storage generally starts at 32GB or 64GB. The exception to these rules is the chromebook, which may have only 16GB of solid state storage.



Ans to the Qus No: 02 (b)

Ans: Logical Cache:

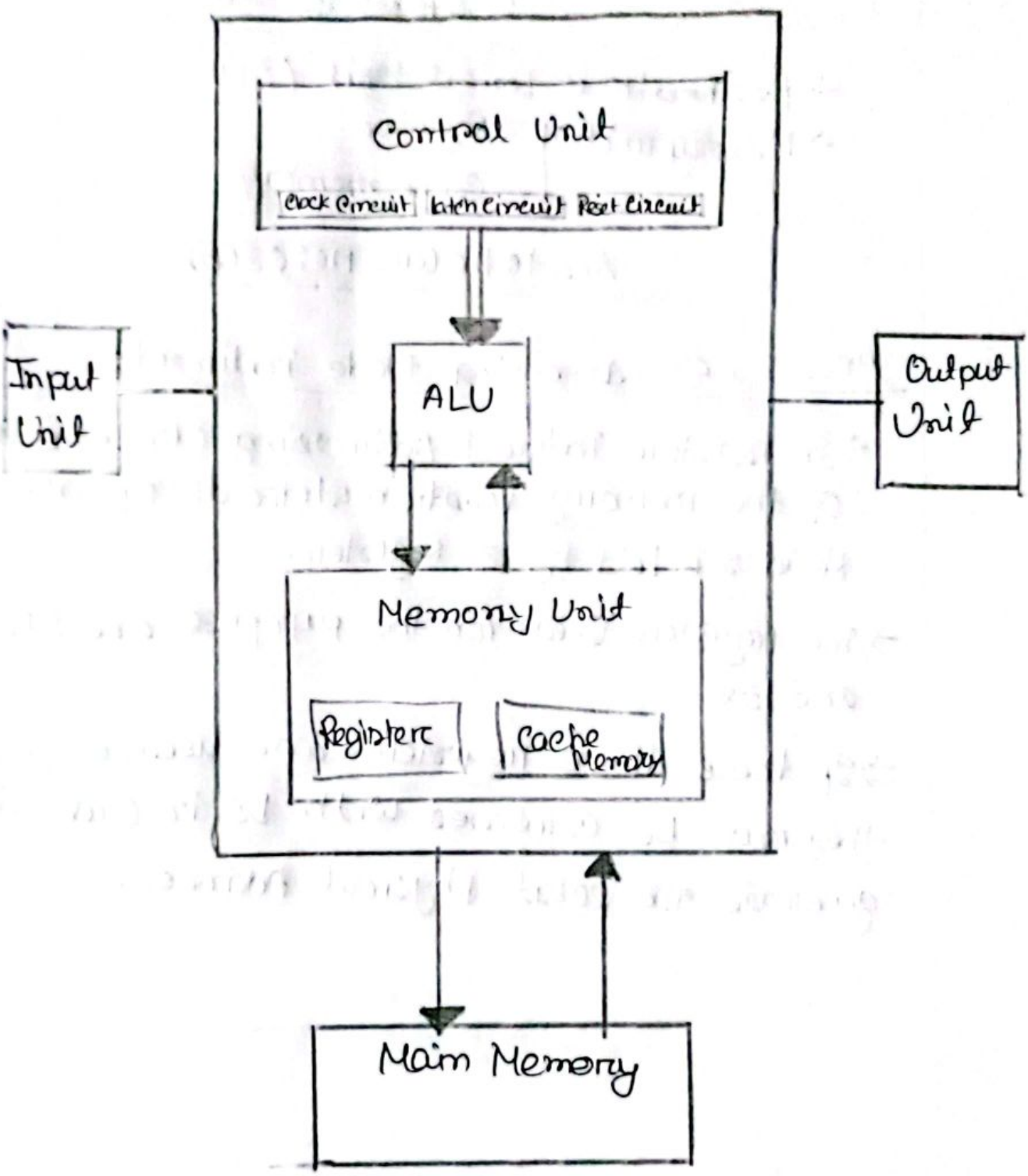
→ It is also known as a virtual cache
→ It is store data using virtual Address.
→ The processor Access cache Directly without going through MMU.
→ Advantage: It is faster then physical cache. Because, the cache can respond before the MMU performs an address translation.

Physical Cache:

The processore Access the cache directly without going through the MMU. A physical caches stores Data using main memory physical address. One obvious Advantage of the logical cache. It that cache Access speed is faster then for a physical cache. Because the cache can Response before the MMU Performs on Address translation. The Disadvantage his to do with the fact that most virtual memory system supply cache Application with the same virtual Memory. Address space That is each Application sees a Virtual memory that start At Address 0. Thus the same virtual Address in two Different Application refers to two different physicle Address. The cache Memory must therefor completly flushed with Cache Application context.

Ans to the Qus No: 03 (a)

Ans:



⇒ Central processing main Unit :

→ Control Unit →
- → Clock Circuit
- → Latch Circuit
- → Reset Circuit.

→ Arithmatic & Logical Unit (ALU)

→ Memory unit
- → Register
- → Cache memory.

Ans to the Qus No: 03(b)

Ans: x86 Addressing Mode indirect:

→ In Register indirect Addressing Mode, the Address of the memory location where the opearand Resides is held by a Register.

→ The registers Used for this purpose are SI, DI, and BX.

→ If these three registers are used as pointers they must be Combined with DS in Order to generate the 20bit physical Address.

Processor Register		
AX	AH	AL
BX	BH	BL
CX	CH	CL
DX	DH	DL
SI		
DI		
BP		
SP		
CS		
DS		
SS		
ES		

Memory	
Memory Address (in Hex)	Data
10000	AAH
10001	BBH
10002	CCH
10003	EEH
10004	DDH
10005	EEH
10006	FFH
10007	11H
10008	22H
10009	33H
1000A	44H