

Name - Sahil Dossa

Roll No - RK 20PD A10

Section : K 20 PD

Registration No :

12005608

CLASSMATE

Date :

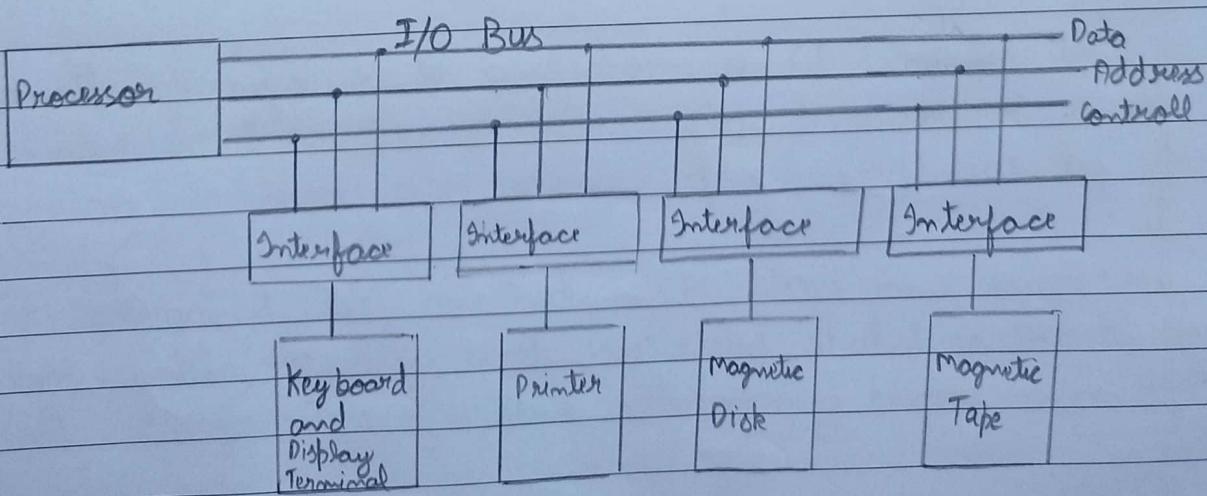
Page :

1

Signature: Sahil

(Q1)

- Ans • The major differences that exist between central computer and the peripheral devices are:
- * Data transfer rate of peripheral devices are slower but faster in central computer
 - * Peripheral devices are generally Electromagnetic devices while central computer is Electronic device.
 - * Data format of peripheral is Byte, Block while for central computer it is Word
 - * Eg of Peripheral devices are keyboard, printer
Eg of Central Computer processor, register, ALU unit.
- Interface provides a method for transferring information between internal storage and external I/O devices:



Name: Sahil Dossa

Roll No: RK20PD010

Section: K20PD

Registration No.: 12005608

CLASSMATE

Date:

Page:

2

Signature: Sahil

Interface:

- Decodes the device address.
- Decodes the commands.
- Provides signals for peripheral controller.
- Synchronizes the data flow and supervises the transfer rate between peripheral and CPU or Memory.

Name : Sahil Dossa

Roll No : RK20 PDA10

Registration No. : 12005608

Signature : Sahil

(Q2)

Ans The data transfer between central computer and I/O devices can be handled in a variety of modes. Some modes use CPU as intermediate path and other transfer data directly to and from memory unit.

- Data transfer to or from peripheral can be handled in one of three possible modes
 - Programmed I/O
 - Interrupt - Initiated I/O
 - Direct Memory Access (DMA)

i) Programmed I/O

- Programmed I/O operations are the result of I/O instruction written in computer program. Each data item transfer is initiated by an instruction in the program.
- Usually transfer is to and from a CPU register to peripheral. Transferring data under program control requires constant monitoring of the peripheral by CPU.
- In programmed I/O method, CPU stays in a program loop until the I/O unit indicated that it is ready for data transfer. This is a time consuming process since it keeps

Signature: Sahil

the processor busy.

It can be avoided by using interrupt facility and special commands to inform the interface to issue an interrupt request signal when data are available for the device.

ii) Interrupted I/O

- The interrupt I/O method helps in reducing the wait time of CPU by using interrupt
- In mean time CPU can proceed to execute another program
- The interface meanwhile will keep on monitoring the device. When the interface determine that device is ready for data transfer it generate interrupt request to the computer.
- Upon detecting the external interrupt signal the CPU momentarily stops the task it was processing, branches to a service program to process I/O transfer and then return to the task it was originally performing.

iii) Direct Memory Access (DMA)

Removing the CPU path from both and letting the peripheral devices manage the memory Buses directly would improves the speed of transfer. This technique is known as DMA

Name: Sahil Dossa

Roll NO.: RK20PDA10

Registration No.: R2005608

CLASSMATE

Date :

Page :

5

Signature: Sahil

- In this CPU initiates the transfer by supplying the interface the start address and the number of words needed to be transferred and then proceeds to execute other task.
- The DMA request memory cycle through Memory Bus. When the request is granted by the memory controller, the DMA transfers the data directly into the memory.
- The CPU merely delays its memory access operation to allow the direct memory I/O transfer.
- Since the peripheral speed is usually slower than processor speed, I/O memory transfer are infrequent compared to processor access to memory.

Name: Sahil Dossa

Registration NO: 12005608

Roll NO: RK20PDF10

CLASSMATE

Date :

Page:

3
6

Signature: Sahil

(Q3)

Ans Types of Addressing Modes:

- Direct Addressing:

- In this Addressing mode the address field of the instruction contains the effective address of the operand.
- Only one reference to memory is required to fetch the operand.
- It is also known as Absolute Addressing Mode.

Eg

$$AC \leftarrow AC + [X]$$

Add X will increment the value stored in the accumulator by the value stored at memory location X

- Indirect Addressing Mode:

- The address field of the instruction specifies the address of memory location that contains the effective address of the operand.
- Two references to memory are required to fetch the operand.

Eg

$$A C \leftarrow AC + [[X]]$$

Add X will increment value stored in the accumulator by the value

Name : Sahil Dursa

Roll No. : R~~R~~K20PDA10

Registration No. : 12005608

CLASSMATE
Date :
Page :

67

Signature: Sahil

stored at memory location specified by X

• Register Direct Addressing Mode:

- The operand is contained in a register set.
- The address field of the instruction refers to a CPU register that contains the operand.
- No reference to memory is required to fetch the operand.

• Eg

$$AC \leftarrow AC + [R]$$

Add R will increment the value stored in the accumulator by the content of register R

• Register Indirect Addressing Mode:

- The address field of the instruction refers to a CPU register that contains the effective address of the operand.
- Only one reference to memory is required to fetch the operand.

• Eg

$$AC \leftarrow AC + [(R)]$$

Add R will increment the value stored in the accumulator by the content of memory location specified in register R.

Name : Sahil Dossa

Roll NO. : RK20 PDA10

CLASSMATE

Date :

Page:

8

Signature: Sahil

- Relative Addressing Mode

- Address field of an instruction specifies the part of the address which can be used along with a designated register to calculate the address of the operand
- 3 different Relative Addressing Modes

- PC Relative Addressing Mode

$$EA = PC + IR \text{ (address)}$$

- Indexed Addressing Mode

(IX : Index Register)

$$EA = TX + IR \text{ (address)}$$

- Base Register Addressing Mode

$$EA = BAR + IR \text{ (address)}$$

(BAR : Base Address Register)

- Eg

$$EA = PC + IR$$

$$8 + 5$$

$$= 13$$

4	
5	
6	
7	0 Add 25
8	

Name : Sahil Dossa

Roll No: RK20PPDA10

CLASSMATE

Date :

Page:

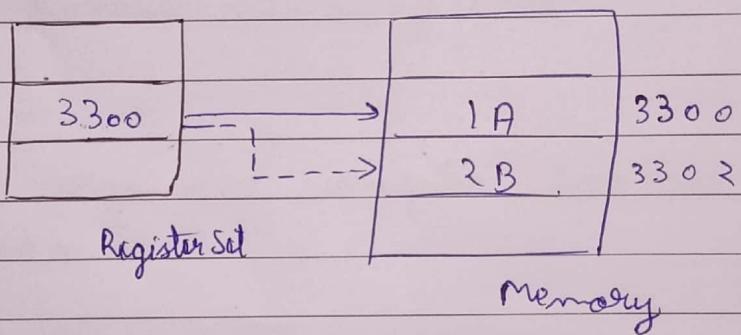
89

Signature: Sahil

→ Auto Increment Addressing Mode :

- This addressing mode is a special case of Register Indirect Addressing mode
- In this addressing mode after the operand, the content of the register is automatically incremented.

Eg

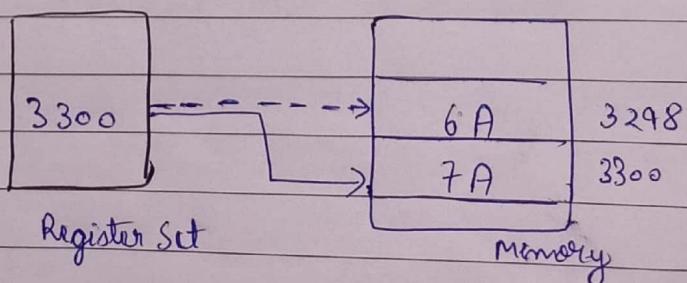


This will get updated to $3300 + 2$
 $= 3302$

→ Auto Decrement Addressing Mode

- This is also special case of Register Indirect addressing mode.
- The value in the register is decremented by 1 automatically.

Eg



Name: Sahil Dossa

Roll No: RK20 PDA10

CLASSMATE

Date :

Page:

10

Signature: Sahil

This will become to 3300 - 3
3298

- Implied Mode:

- Address of the operands are specified implicitly in the definition of the instruction.
- No need to specify address in the instruction. It is also known as Implicit Addressing Mode.
- Eg
In a stack organized computer, Zero Address Address instructions are implied mode.

- Immediate Mode:

- Instead of specifying the address of the operand, operand itself is specified.
- No need to specify address in the instruction. Fast to acquire an operand.
- Eg
ADD 10 will increment the value stored in the accumulator by 10

Name: Sahil Dossa

CLASSMATE
Date :
Page :

Roll No : RK20PDA10

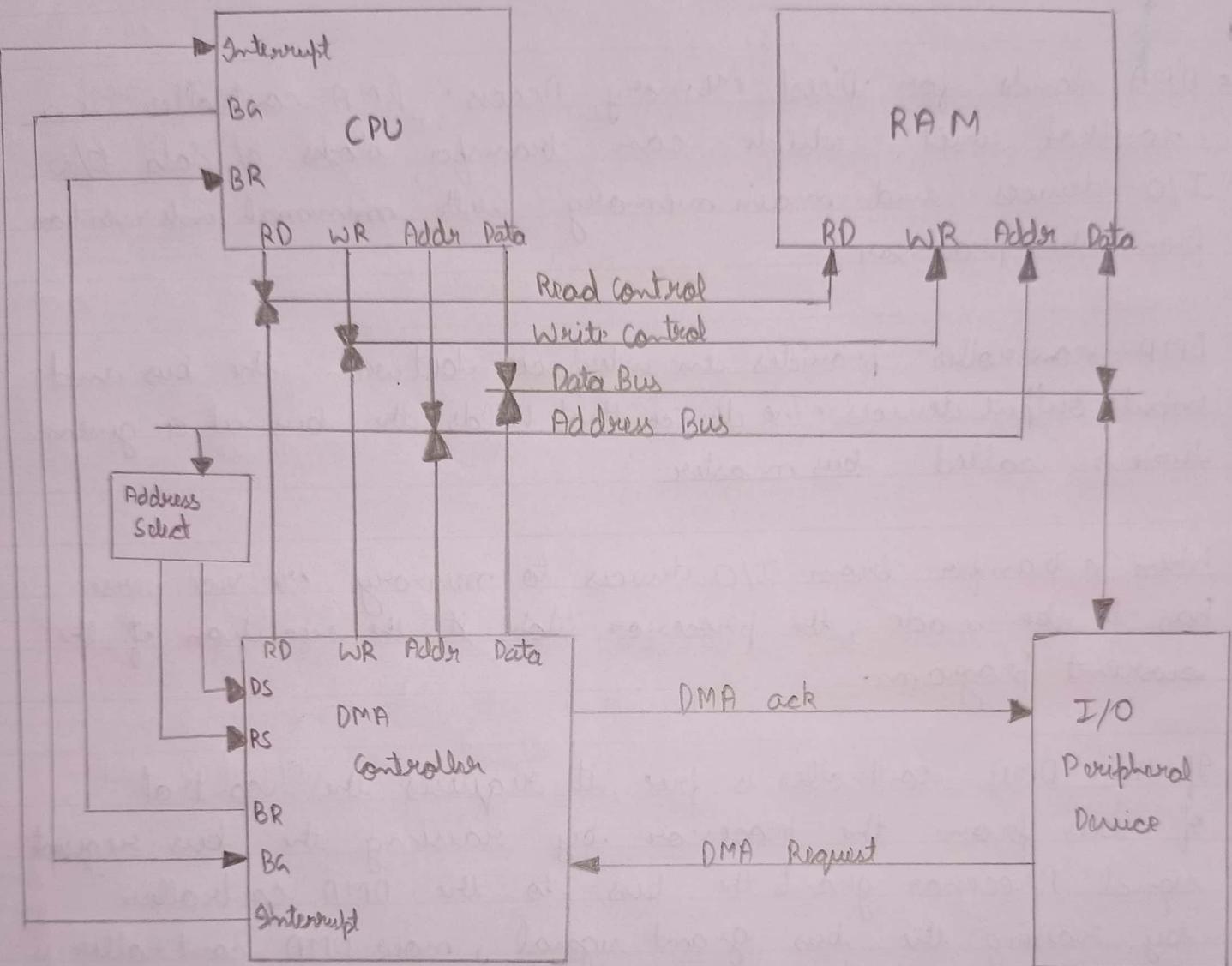
110

Signature: Sahil

Q4)

Ans. DMA stands for Direct Memory Access. DMA controller is a control unit which can transfer blocks of data b/w I/O devices and main memory with minimal intervention from the processor.

- DMA controller provides an interface between the bus and input-output devices. The device that holds the bus at a given time is called bus master.
- When a transfer from I/O devices to memory or vice versa has to be made, the processor stops the execution of the current program.
- If the DMA controller is free, it requests the control of bus from the processor by raising the bus request signal. Processor grants the bus to the DMA controller by raising the bus grant signal, now DMA controller is the bus master.
- DMA controller now has the full control of buses and can interact directly with memory and I/O devices independent of CPU. After completion of data transfer, it disables the bus request signal thereby handing control of buses to the CPU.
- When an I/O device wants to initiate the transfer then it sends



Signature: Sahil

- a DMA request signal to the DMA controller, for which the controller acknowledge if it is free . Then DMA controller request the processor for the bus, generating the bus request signal.
- After receiving the bus grant signal it transfer the data from the device .
- The DMA controller ~~transferring~~ transfers the data in Two modes:
 - **Burst Mode**: In this mode DMA handover the buses to CPU only after completion of whole data transfer.
- This mode of transfer is needed for fast devices such as Disk where data transmission cannot be stopped until an entire block is transferred.
 - **Cycle Stealing**: In this mode, DMA gives control of buses to CPU after transfer of every byte.
- It continuously issue a request for bus control , makes the transfer of one byte and returns the bus.
- CPU merely delays its operation for one memory cycle to allow direct memory I/O transfer to "steal" one memory cycle.