CHAPTER-3

Dt - 04-03-24

A Top Level Vieco of Computer - Function & Structure.

1) What are the 3 main concepts of Von - Neumann's Architecture.
2) Differentiate b/w hardcoare & softcoare programming approaches
3) Describe the Instruction ~~fetch~~ Execute & Fetch cycle with/coithout intercept.
4) Problems from a hypothetical machine
5) What do you understand by ISR. (Intercept Service Routine)
6) Describe the program flow of control without or coith short I/o coait or long I/o coait.
7) Discuss the diff^n mechanism to handle multiple intercept.
8) Short note on inter connection structure.
9) What are the diff^n types of transfers supported by interconnection structure.
10) Discuss about the diff^n types of bus interconnections

* Computer Components

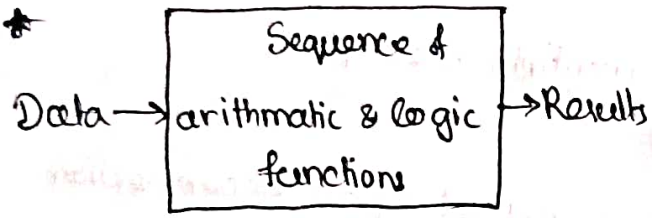
→ Von Neumann architecture is based on 3 concepts

(i) Data & instruction are stored in a single read-write memory

(ii) The contents of this memory are addressable by location irrespective of the type of data present there.

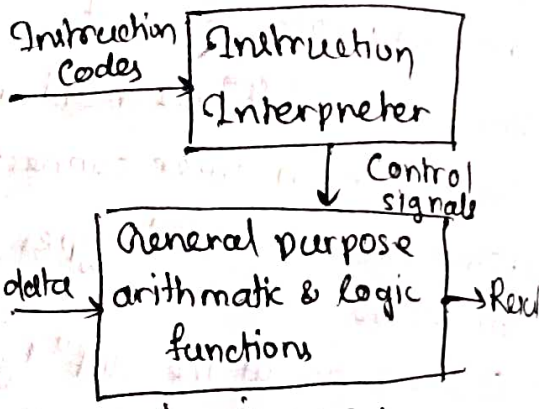
(iii) Execution occurs in the sequential manner from one instruc to the next.

→ Hardware Prog Approach



* Programming is difficult.
* We have to rewire the hardware for each new program.

Software Prog Approach

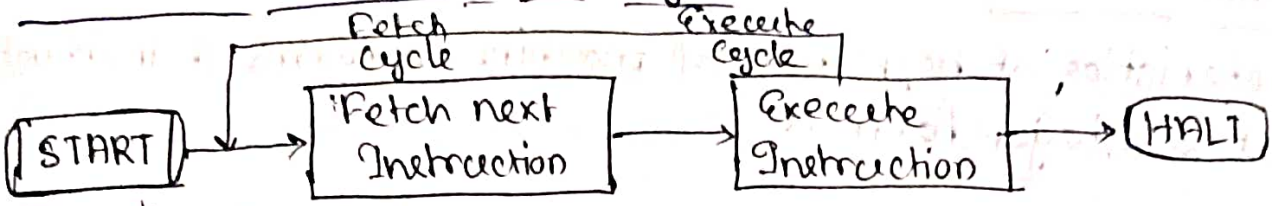


* Programming is easy.
* Instead of rewiring the hardw for each new programs, we ha to provide a new sequence of co only.

(A) Computer Functions

Refers to chapter 1

* Instruction Execeute & Fetch Cycle:-



Describe Fetch Cycle:-

- At the beginning of each instruction cycle, the processor fetches an instruction from the memory.
- A register called PC (Program Counter) holds the address of the instruction to be fetched next.
- The processor always increments the PC after each instruction fetch (So that it will fetch the next instruction in sequence)

Dt - 12-03-24

- The instruction cycle is of 4 diff^n types.

(i) Processor - memory

(ii) Data may be transfered from processor to memory or from memory to processor.

(iii) Processor - I/O

→ Data may be transfered to or from a peripheral device by transfering b/w the processors & I/O modules.

(iii) Data Processing

→ The processor may perform some arithmatic or logical operations on data.

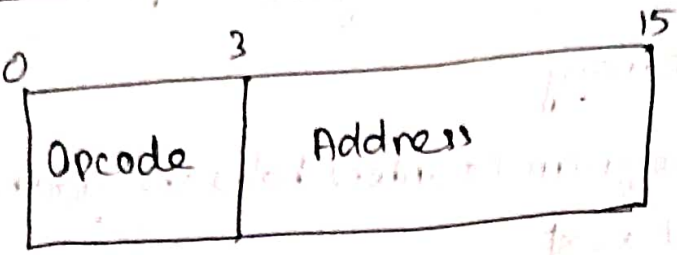
(iv) Control

→ An instruction may specify that the sequence of execution to be alter.

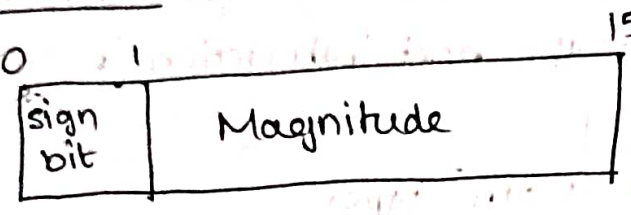
Hypothetical Machine

(i) Characteristics of hypothetical machine in terms of instruction format & integer format (Given in the question)

Instruction Format



Integer Format



- For this hypothetical machine, the processor contains a single data register called accumulator (denoted as AC). Both instructions & data are 16 bits long. So, the memory can be organized into 16 bit word. The instruction format supports the 4 bit opcode value.
- The memory can be addressed upto $2^{12} = 4K$ location.

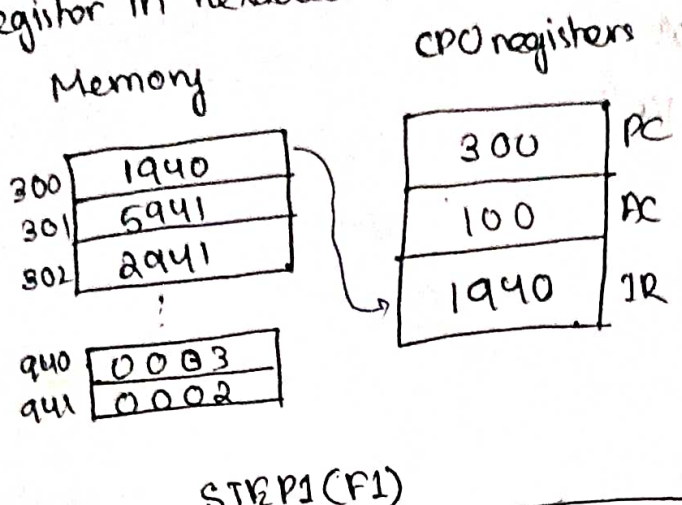
To Find out :-

- Illustrate the partial program execution, Memory location value & processors register contents. (PC, AC, IR)

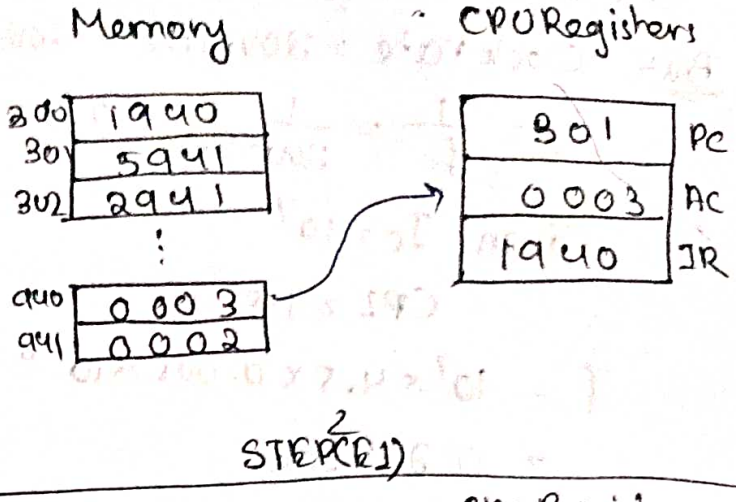
Ex:- The below program section shows the addition of the contents of memory word at address 0940H (content/data is 0003H) to the content of memory word at address 0941H (data = 0002H) & Stores the result in the later location

```
MOV AX, [0940H]   // AX = 0003
ADD AX, [0941H]   // AX ← AX + [0941]
                  //      ← 5
MOV [0941H], AX   // [0941H] ← 0005H
```

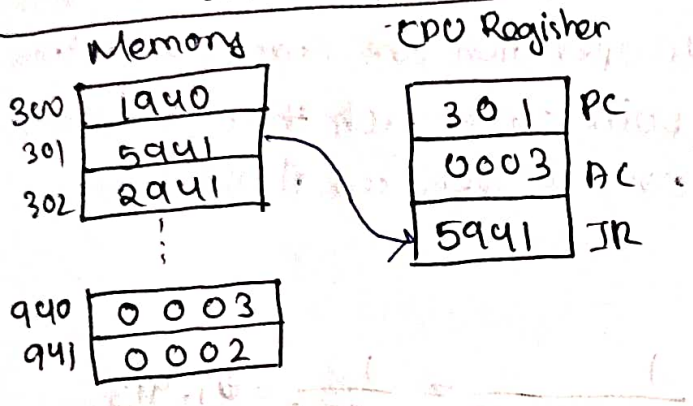
Program execution showing the contents of memory locations & processor registor in hexadecimal.



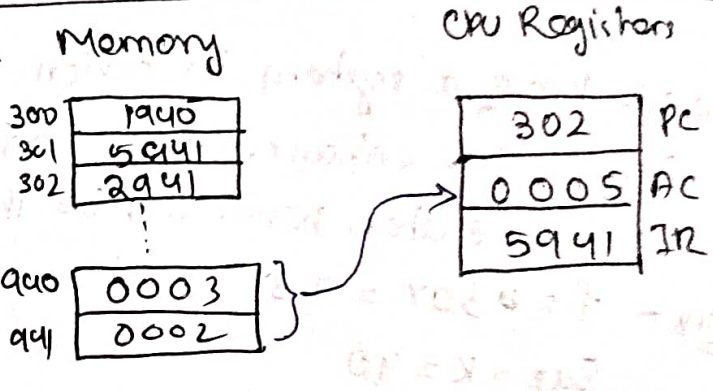
STEP1 (F1)



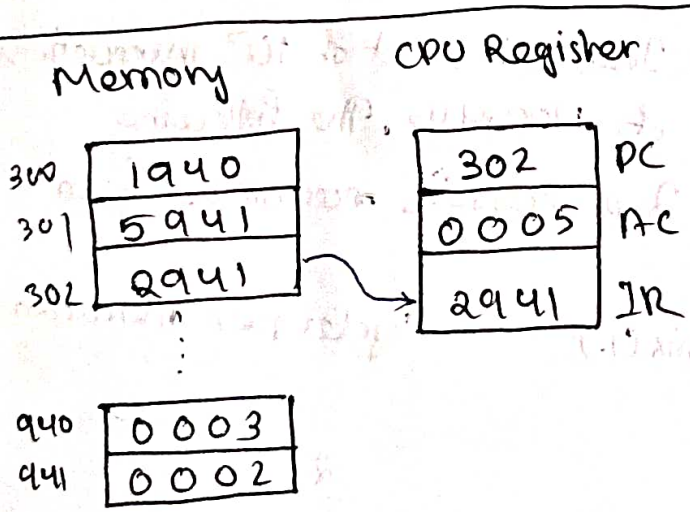
STEP (F1)



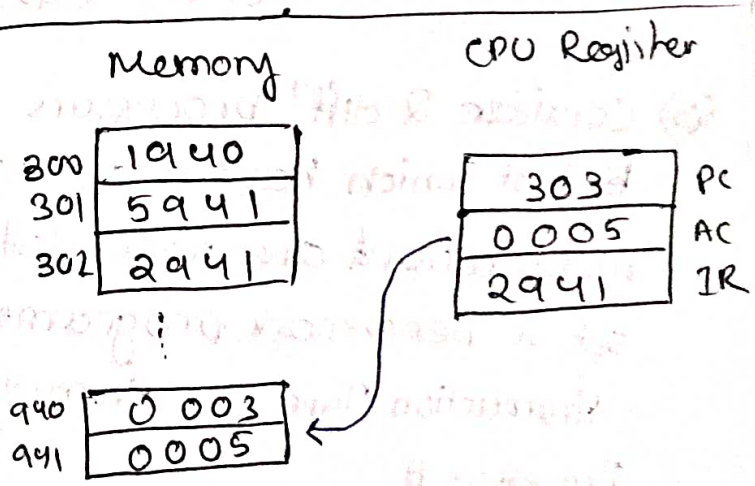
Step-3 (F2)



Step-4 (F2)



Step-5 (F3)

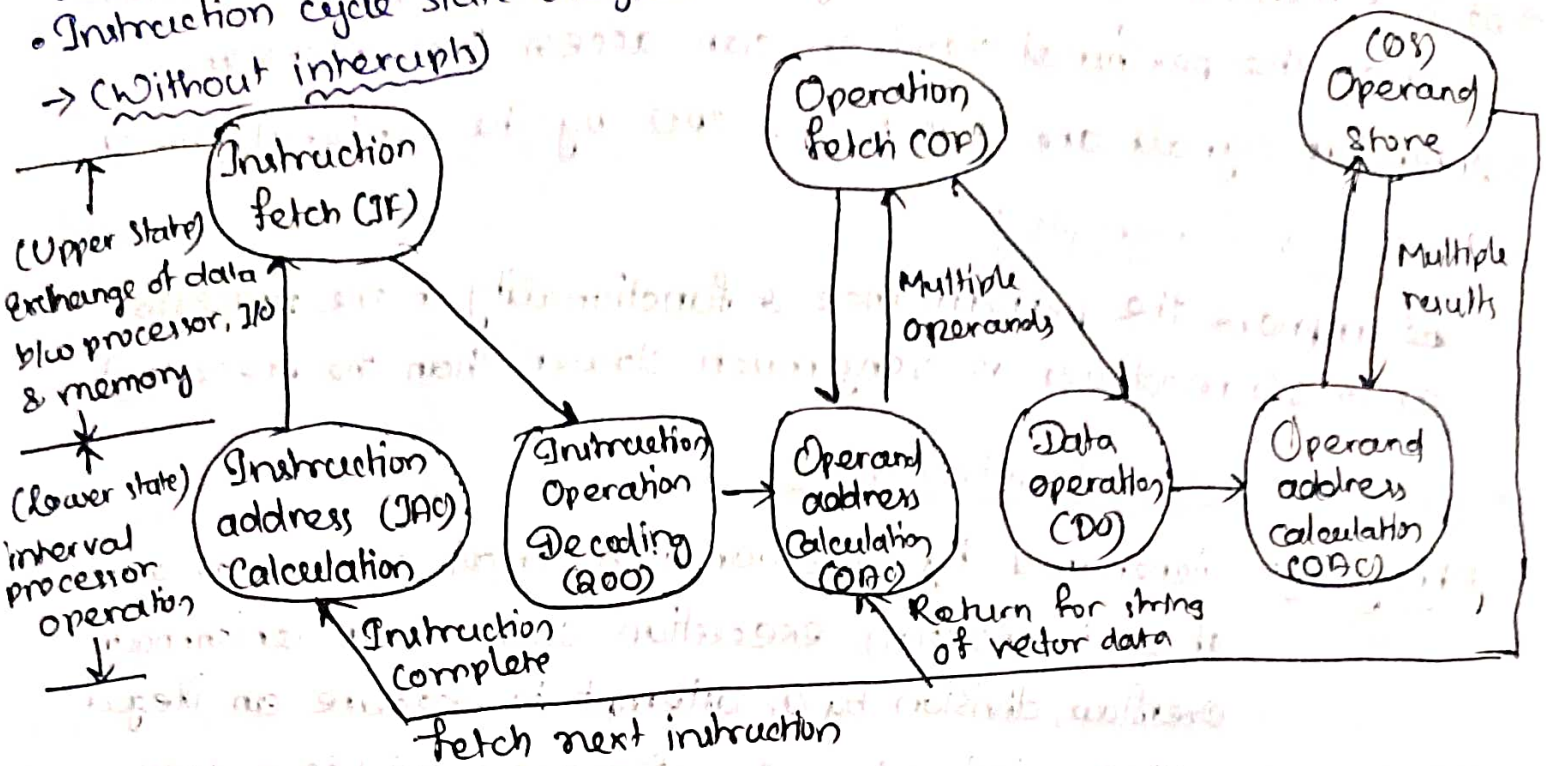


STEP 6 (F3)

D:- 14-03-24

- Instruction cycle state diagram

→ (Without interrupts)



- States in the upper part involves an exchange & either the memory or the I/O devices
- States in the lower part involves only interval processor operation (ALU, Registers)
- The diff^{n} states can be described as follows:
  - * IF - Read ~~input~~ instruction from its memory location into the processor.
  - * IOD - Analyze instruction to determine the type of operation to be performed & operands to be used.
  - * OAC - If the operation involves reference to an operand in memory or is available through I/O then it determines the address of the operands.
  - * OF - Fetch the operand from memory or read it from the I/O module.
  - * DO - Perform the operation indicated by the instruction
  - * OAC - Already written (above)
  - * OS - Write the result into memory or to the ~~do~~ I/O device

* IAC - Determine the address of the next instruction to be executed.

Interrupts :-

→ It is a method of temporarly halting the program execution so that the peripheral devices can access the processor.

→ Interrupt signals are sent to the CPU by the peripheral devices

→ Advantages of intercupts :-
improve the performance & functionality of the processor (as the I/O modules is very much slower than the processor)

→ Common classes of interrupts :-

i) Program :- Generated by some condition that occurs as a result of an instruction execution such as an arithmatic overflow, division by 0, attempt to execute an illegal machine instruction & references outside a user allowed memory space

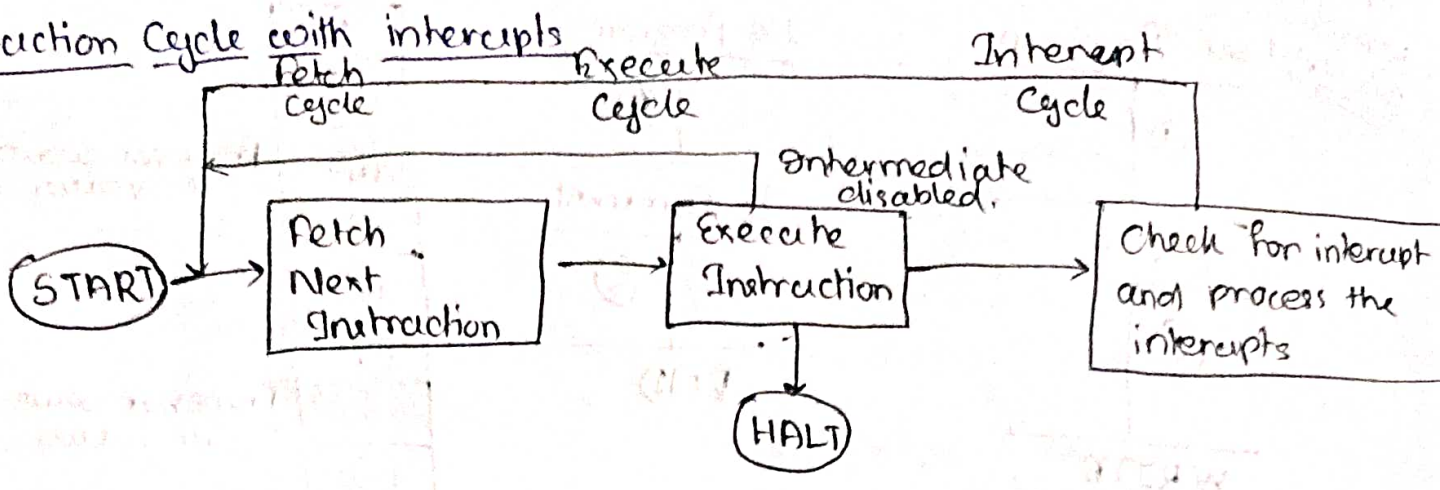
ii) Timer - Generated by the timer circuit present within the processor

iii) I/O - Generated by the I/O controller

iv) Hardware - Generated by a failure such as power failure, memory failure.

Dt - 28 - 03 - 24

Instruction Cycle with interrupts

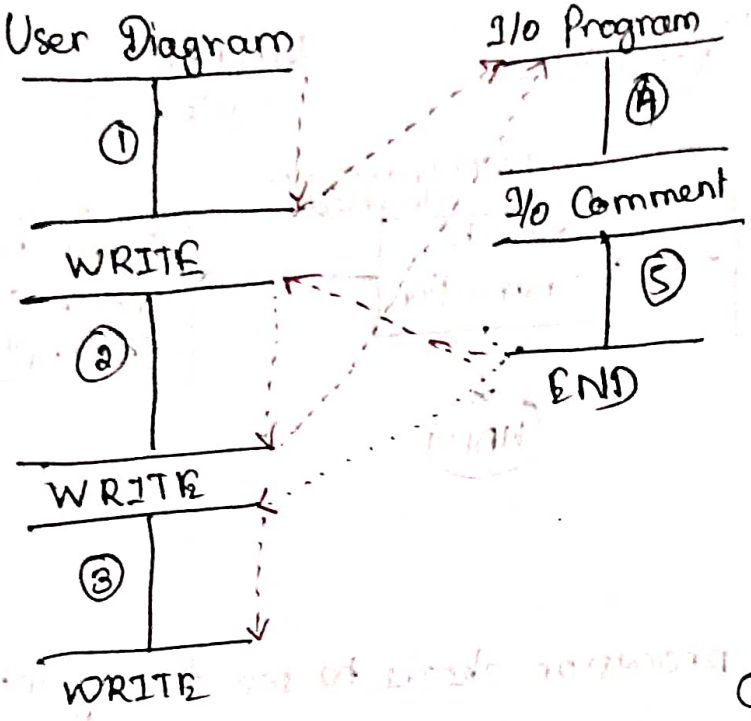


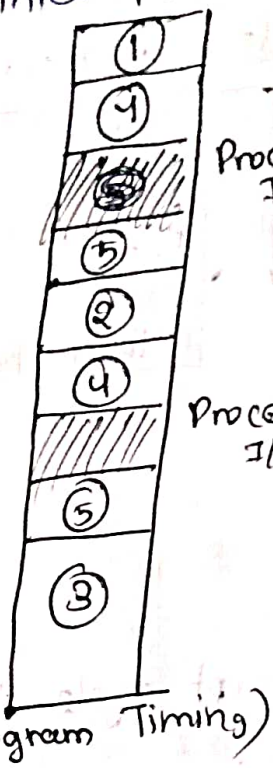
~~How the~~

- On the interrupt cycle, the processor checks to see if any interrupts have occcured.
- If no interrupts are pending, the processor proceeds to the fetch cycle and fetches, the next instruction of the current program.
- If an interrupt is pending, then the processor
  (i) Suspends the execution of the current program & saves its data
  (ii) Sets the PC (Program Count) to the starting address of the interrupt handler
- The processor now proceeds to the fetch cycle and fetches the first instruction in the interrupt handler program (part of OS), which provide service to the interrupt.
- Interrupt handler determines the nature of the interrupt & performs the action to be taken.
- These actions can be of 2 diff^{n} types.
  (i) Interrupts with Short I/O wait
  (ii) Interrupts with Long I/o wait.

Program flow of control with & without interrupts

Without Interrupts





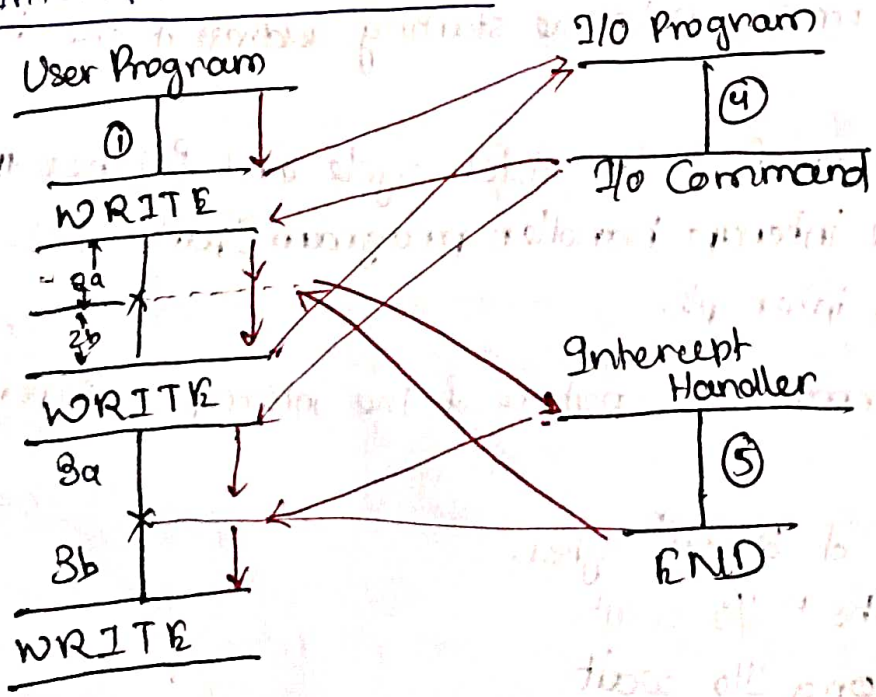
(Program Timing)

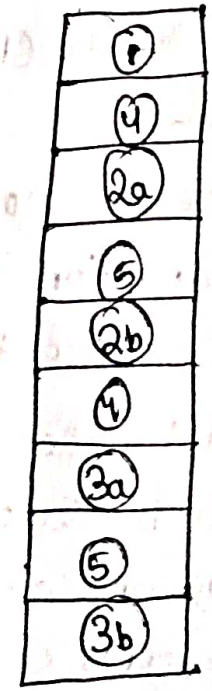
Program flow of control without Interrupts

Disadvantage:-

- Processor waits to process I/O commands & there it remains idea

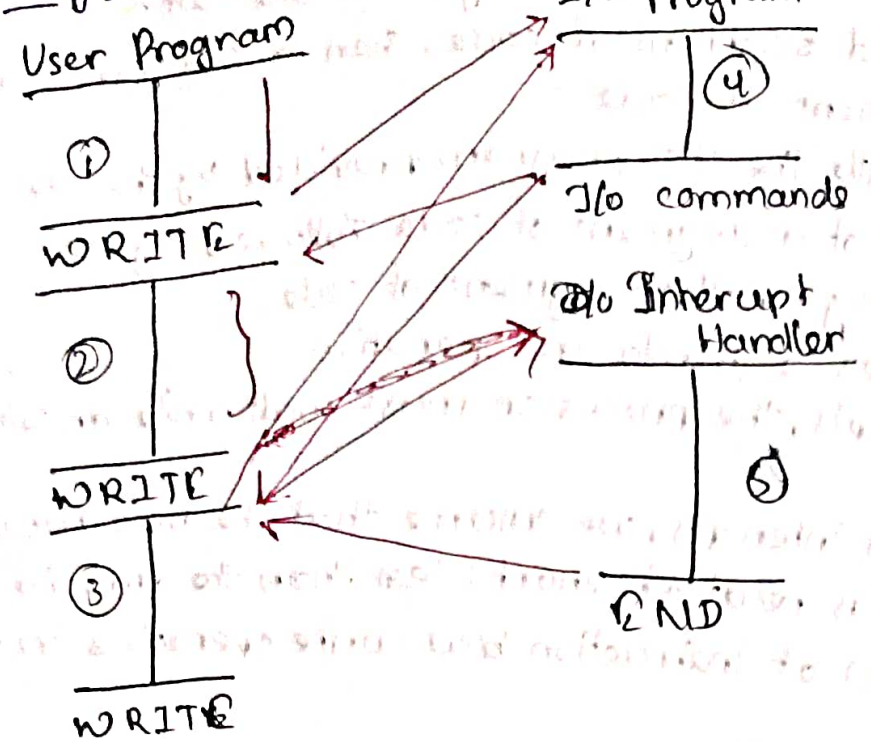
With Interrupts short I/O wait



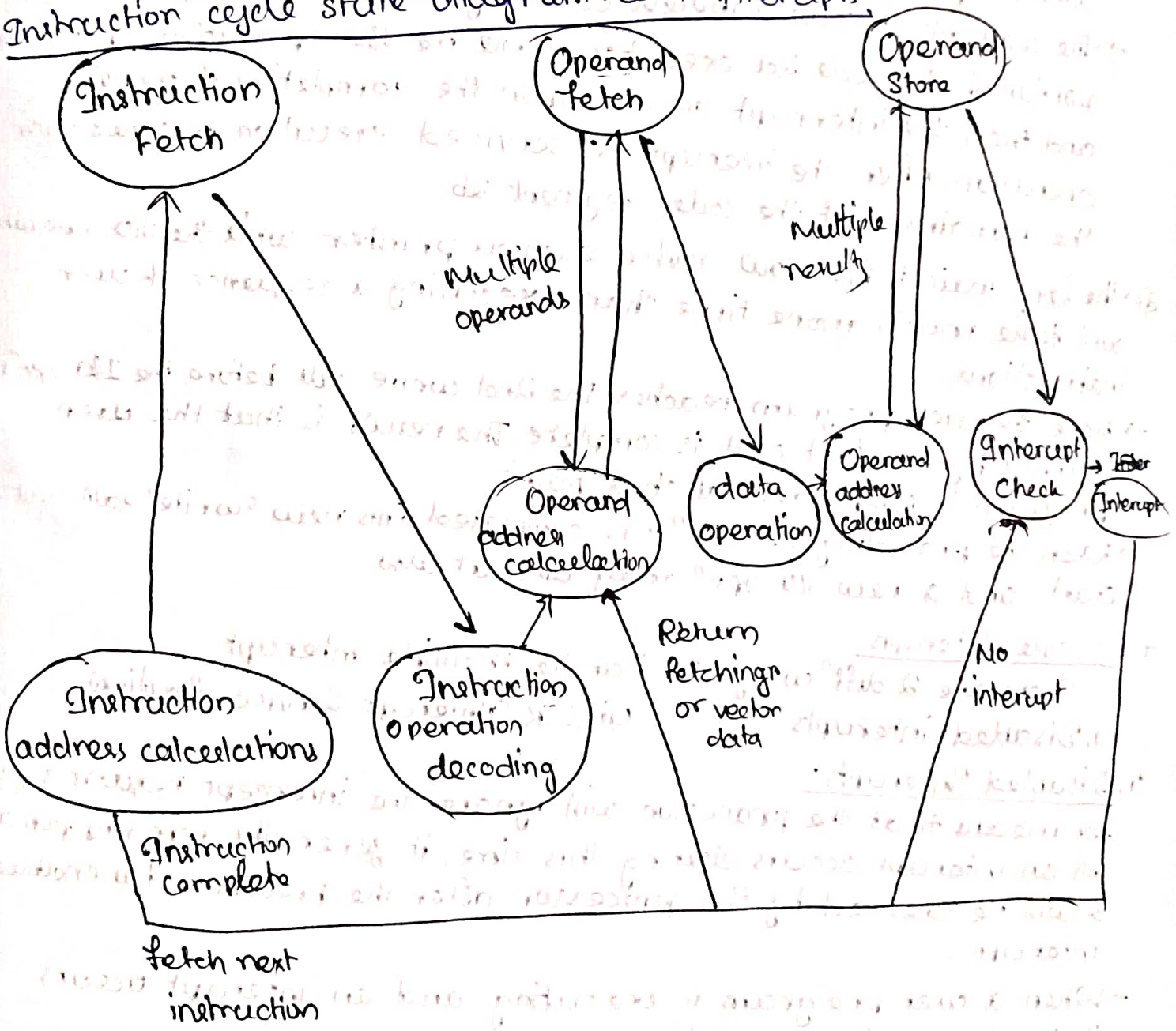


Program flow control with short I/O interrupt

With Long I/O bit



Instruction cycle state diagram with interrupts:-



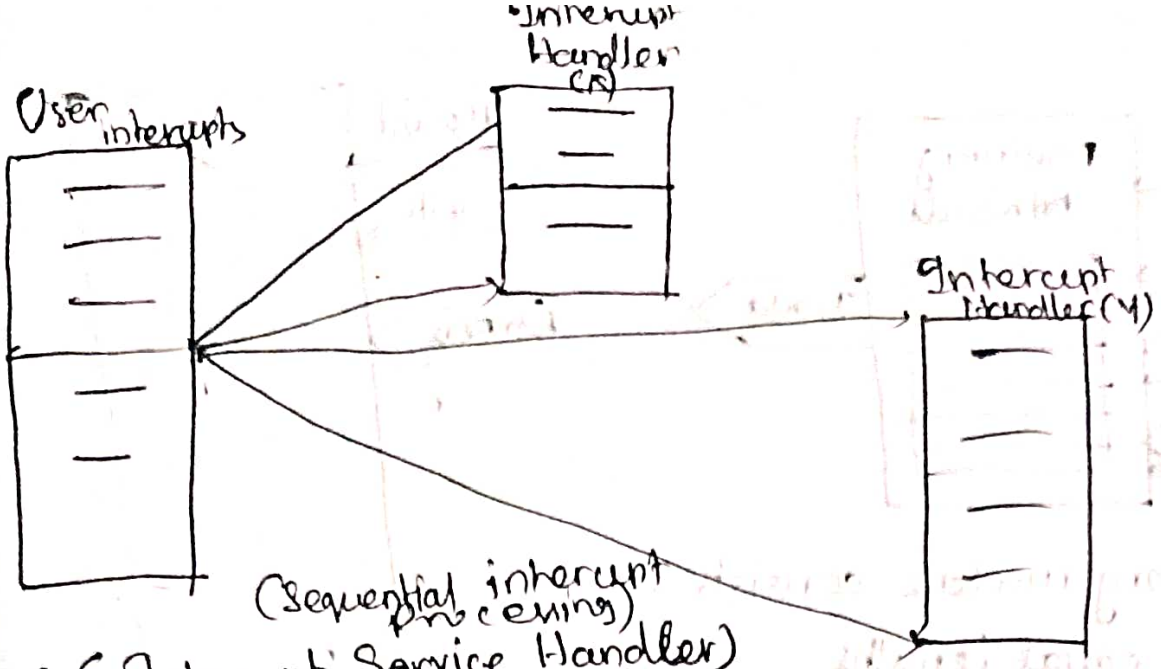
1) Description for interrupt :-
- We have a user program that contains 2 write operations
- There is a segment of code at the beginning, then one 'write' command, then a second segment of code, then a 2nd 'write' command then a 3rd & final segment of code
- The 'write' command calls the I/O program provided by the OS.
- The I/O program consists of a segment of code followed by an I/O command, followed by another segment of code.
- The I/O command involves a hardware operation
- When there is no interrupts, the processor must wait while an I/O operation is performed.

2) → In short I/O wait with interrupts, we assume that the time required for the I/O operation is relatively short: less than the time to complete the execution of instruction blw write operations in the user program.
→ The segment of code labeled code segment 2 is interrupted. A portion of the code 2a executes, while the I/O operation is performed and then the interrupt occurs upon the completion of the I/O operation. After the interrupt is serviced, execution resumes with the remainder of the code segment 2b

3) → The long wait is for slow device such as printer, while the I/O operation will take much more time than executing a sequence of user instructions.
→ Here the user program reaches the 2nd write call before the I/O opr's started by the first call is complete. The result is that the user program is suspended at that point.
→ When the preceding I/O opr's is completed, this new 'write' call will start, and a new I/O opn may start also

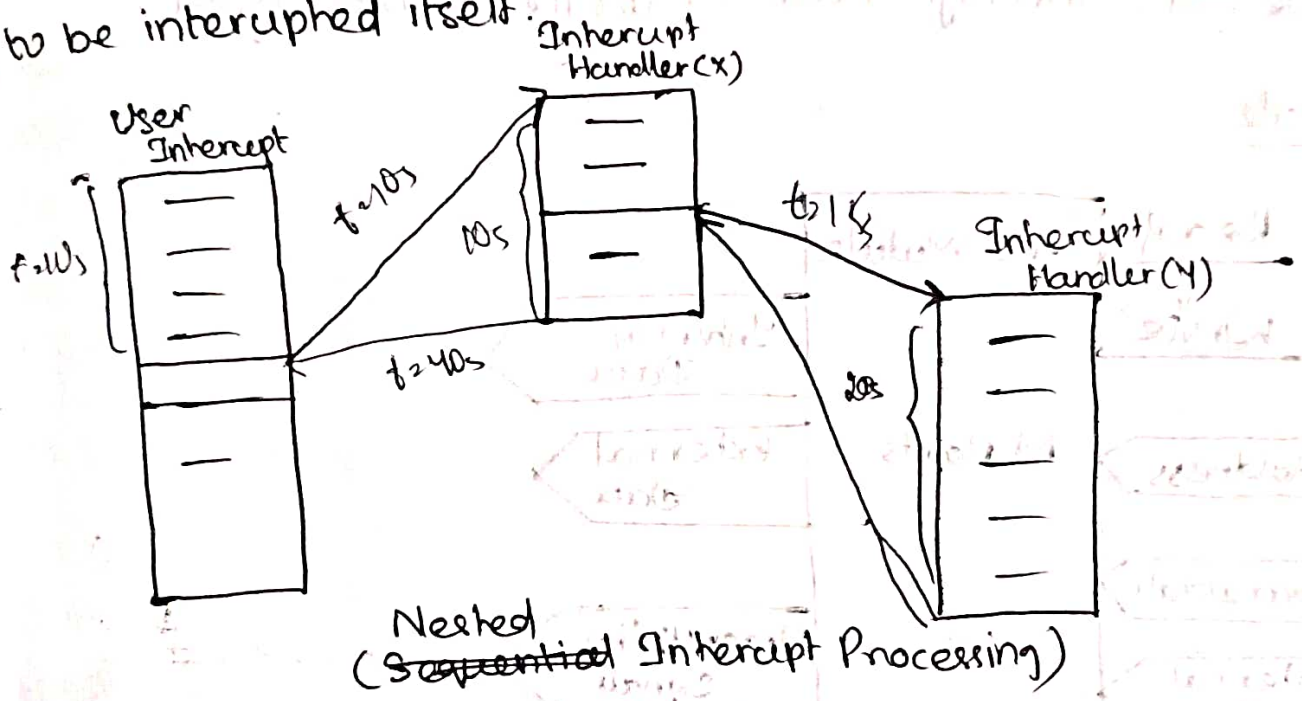
* Multiple Interrupts
There are 2 diff ways to handle multiple interrupt
(i) Disabled interrupts (ii) ISR (Interrupt Service Routine)

(i) Disabled Interrupts :-
- It means that the processor will ignore the interrupt request signal of an interrupt occurs during this time, it generally remains pending & will be checked by the processor after the processor has enabled interrupt.
- When a user program is executing and an interrupt occurs it is disabled immediately.



ISR (Intercept Service Handler)

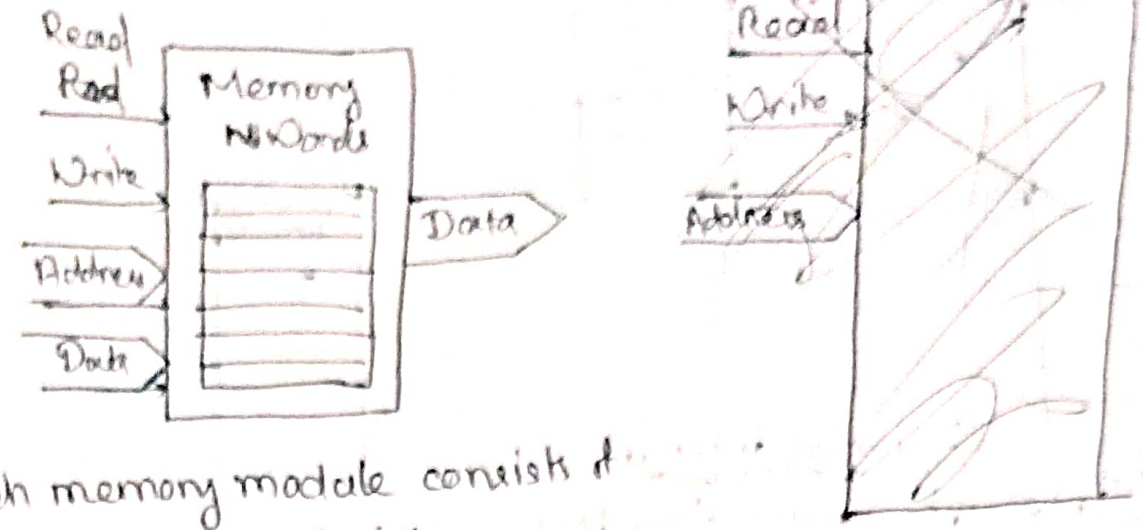
It defines the priorities for the intercepts & allows an intercept of higher priority to cause the low priority intercept handler to be interrupted itself.



+ Interconnection Structure :-

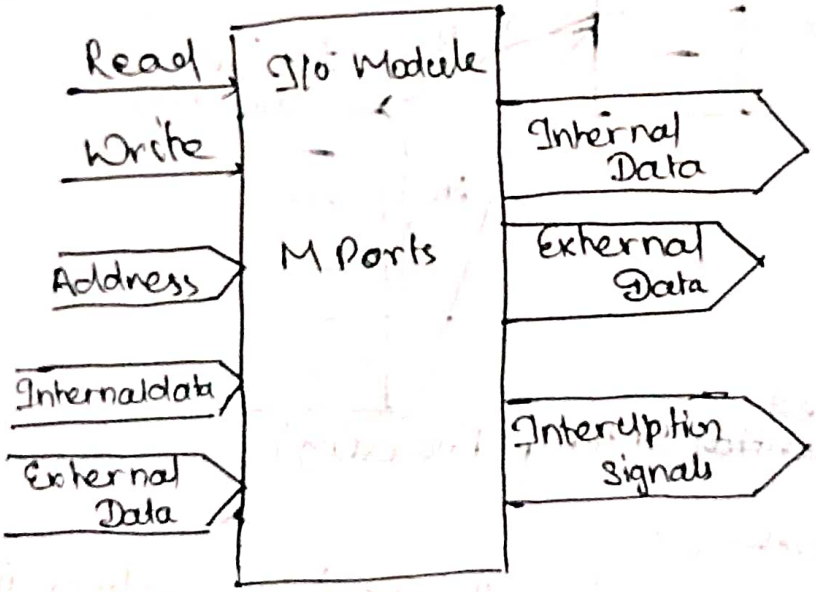
- The collection of paths connecting the various modules is called the interconnection structure.

- Memory

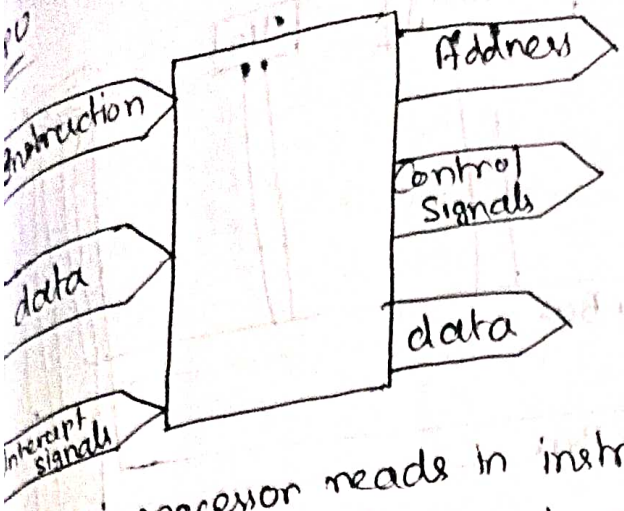


- Each memory module consists of N-words of equal lengths
- Each word is assigned with a physical address written
- A word of data can be read from or ~~within~~ into memory it can be done through read or write control signal

- I/O Module



- There are 2 operations ~~on~~ Read & write
- Each of the interfaces to an external device is known as ~~a~~ ports & provided with an unique address.
- ~~There~~ are external data paths for the input & output of data with an external device.
- An I/O modules should also be able to send intercept signals to processors



• The processor reads in instructions and data & writes out data after processing and ~~usage~~ uses control signals to control the overall operations of the system.

interconnection structure must support the following types of ansfers.

i) Memory to processor

ii) Processor to memory

iii) I/O to processor

iv) Processor to I/O

v) I/O to or from memory

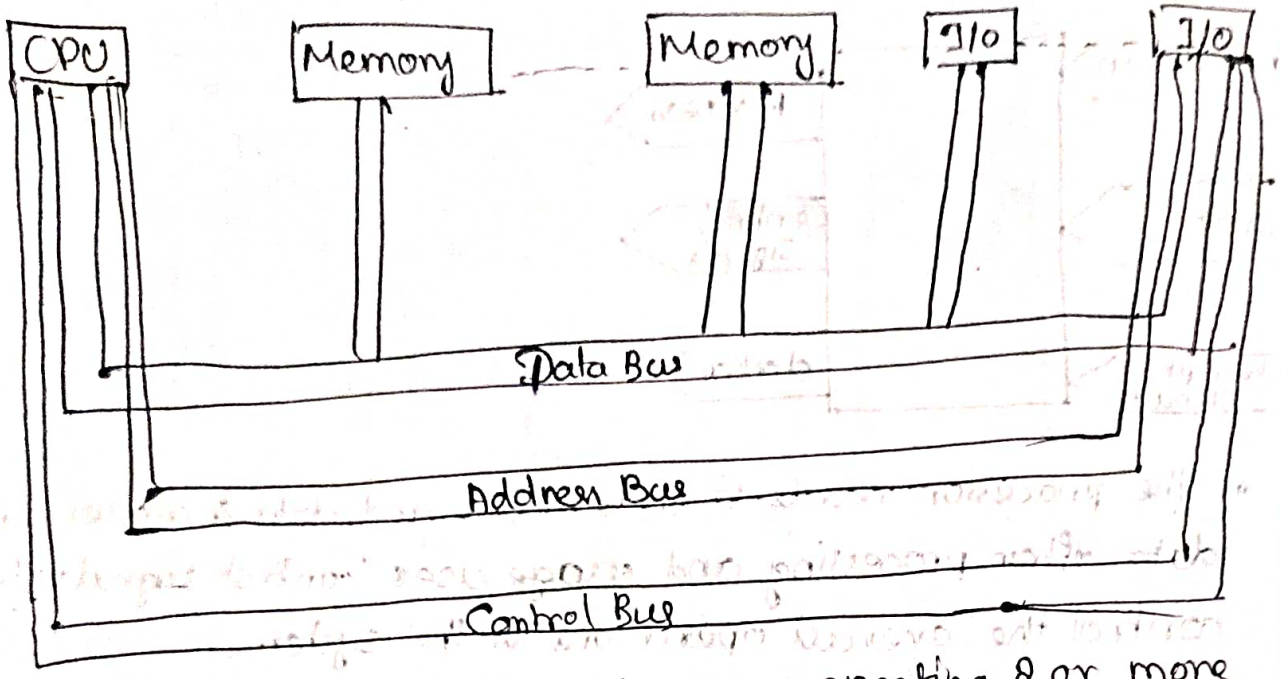
cuss the diff" mechanism to handle multiple interupt

ort note on inter connection structure

nat are the diff" types of transfers supported by interconnection ucture.

iscuss about the diff" types of bus interconnections

* Bus Interconnections :-



- A bus is a communication pathway connecting 2 or more devices. Bus is a share transmission medium.
- Each bus/line is capable of transmitting the binary signals only.
- Data Bus :-
  The data lines provide a path for exchange of data amo the system modules.
  Ex- 8086 ~~is a 16~~ Data bus ~~width~~ width 16 bit
- Address Bus :-
  The address lines are used to designate the source / destin of data on the data bus.
  Ex :- 8086 - Width of address bus = 20 bit
  The width of the address bus determines the maximum possible memory capacity of the processor.
- Control Bus :-
  - The control lines are used to control the ~~access~~ access to t processor and also to the uses of data bus & address bus
  - Timing signals are used to validate the transmitting dev

Dt- 02-04-26

Q) Consider a hypothetical 42 bit micro processor having 32 bit instruction composed of 2 ~~bits~~ fields. The 1st field contains the ~~operand~~ opcode of one bite & the remainder contains the immediate operand or an operand address. what is the impact of the system speed of the microprocessor bus has
(i) 32 bit local address bus
(ii) 16 bit local data bus.

Ans :-

(i) 32 bit local address bus can support 32 bit of data and so, we need 1 clock cycle for the data transfer.

(ii) The 16 bit local data bus can carry 16 bits of data only. But we have to transfer 32 bits of data, so, we need 2 clock cycles to transfer the 32 bits of data.

(3) Consider a hypothetical 32 bit micro processor having 32 bit instruction composed of 2 fields. The 1st byte contains the opcode & the remaining fields contains the immidiate data. (a) what is the maximum directly addressable memory capacity in bytes.

(b) Discuss the impact on the system speed if the microprocessor bus has (i) 32 bit local address bus & 16 bit local data bus.

(ii) 16 bit local address bus & 16 bit local data bus

(c) How many bytes are needed for the program counter & a instruction register?

Sol?:-

(a) 32 - 8 = 24 bit (left)

Max addressable memory capacity = 2^{24} = 16,777,216.

(b)(i) 32 bit local address bus can support 32 bit of address in 1 clock cycle & we need 2 clock cycles to transfer 32 bit of data on 16 bit local data bus line.

Total we require = 2+1 = 3 clock cycles to transfer data

(ii) 16 bit local address bus can support 32 bit of address in 1 clock cycles & we need 2 clock cycles to transfer 32 bit data on 16 bit local data bus line

Total we require = 2+2 = 4 clock cycles to transfer data.

c) No. of bits needed for PC = 24
(Program counter)

No. of bits required for IR = 32
(Instruction register)