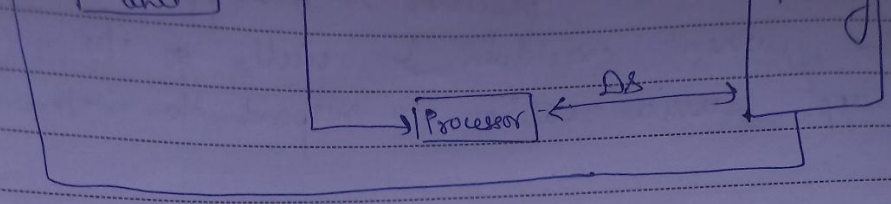


CPU

Set-2

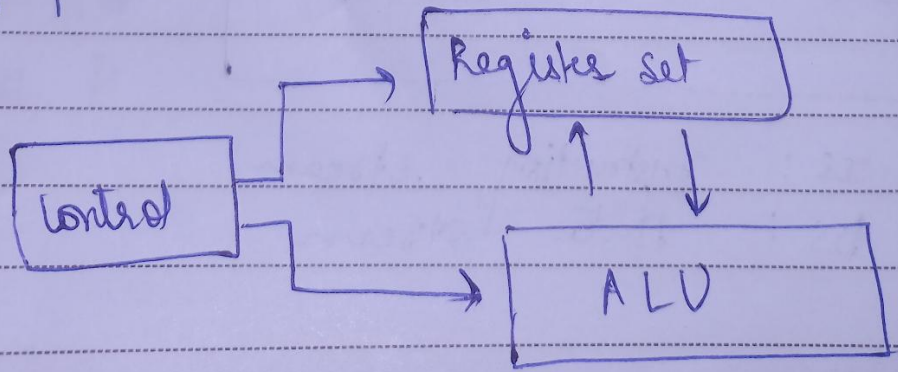
The part of computer that performs the bull of data-processing operations is called the central processing unit.

CPU is made up of 3 major parts : Register set, ALU & control unit.

The register set stores intermediate data used during the execution of the instructions.

The ALU performs the required microoperations for executing the instructions.

The control unit supervise the transfer of information among the registers and instructs the ALU as to which operation to perform.



- All operations are performed with an implied accumulator register.
- The instruction format in this type of computer uses 1 address field.

⇒ Types of CPU organisation

① Single accumulator organisation

ADD B

or

→ for this type of organization, value should be in accumulator

ADD B AC ← AC + M[B]

AC is the accumulator register & M[B] is memory word located at B.

② General Register Organisation

- When we are using multiple general-purpose registers instead of single accumulator register, in the CPU organization then it's called general-register based CPU organisation.
- In this type of organization, computer uses 2 or 3 address fields in their instruction format.

* Three address field :

M	opcode	Destination Add.	source¹ Add.	source² Address
			A	B

X = (A+B) * (C+D)

ADD R1 A, B R1 ← A+B

ADD R2 C, D R2 ← C+D

MUL X R1, R2 X ← R1 * R2

Note :

No. of instructions are less so speed is high.

* Two Address field :

$X = (A+B) * (C+D)$

```
MOV   R1  A
ADD   R1  B       R1 ← R1 + B
MOV   R2  C
ADD   R2  D       R2 ← R2 + D
MUL   R1  R2      R1 ← R1 * R2
```

→ here no. of instructions are more so, the speed is less.

(3) Register Stack Organisation

- A useful feature that included in the CPU of most computers is stack or Last-in, first-out LIFO. A stack ~~storage~~ is a storage device that stores information in such a manner that the item stored in last is first retrieved.

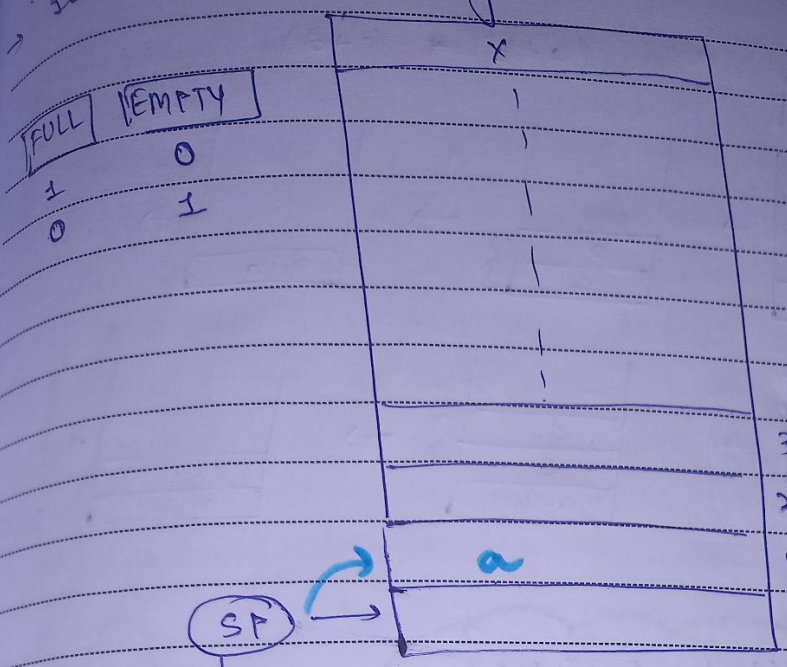
It contains a register that holds the address for the stack called stack Pointer (SP), because its value always points at the top item in the stack.

Stack has 2 operations :

PUSH : The operation of insertion is called push, which is used for pushing a new item in stack.

OP : The operation of deletion is called POP, it's used for removing the top most item from stack.

→ it's a 64 words register



stack pointer (6 bit register) Data Register (DR)

PUSH

SP ← SP + 1

M[SP] ← DR

If (SP=0) then (FULL ← 1)

EMPTY ← 0

POP

DR ← M[SP]

SP ← SP − 1

If (SP = 0) then (EMPTY ← 1)

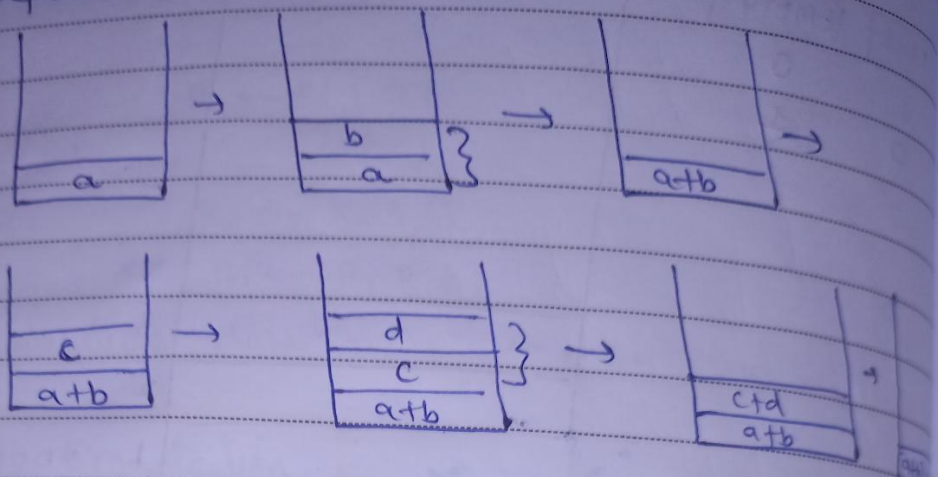
FULL ← 0

- Initially SP will points to 0.
- let in data register we have a & we want to push a in stack, for this we will perform SP ← SP+1 & M[SP] ← DR
- Now SP will point to 1, a will be inserted there.
- In next step it will check If (SP=0), No then value of FULL = 1 & Empty = 0

* Zero Address Instruction

(a+b) * (c+d)

Push a
Push b
Add
Push c
Push d
Add
Mul



INSTRUCTION FORMAT

Instruction : X = (A+B) X (C+D)

1) Three Address Instruction

ADD	R_1	A, B	$R_1 \leftarrow A+B$
ADD	R_2	C, D	$R_2 \leftarrow C+D$
MUL	X	R_1, R_2	$X \leftarrow R_1 \times R_2$

It result in short program, when evaluating arithmetic expressions.

The binary-coded instruction requires too many bits to specify three addresses.

2) Two Address Instruction

MOV	R_1	A	$R_1 \leftarrow M[A]$
ADD	R_1	B	$R_1 \leftarrow R_1 + M[B]$
MOV	R_2	C	$R_2 \leftarrow M[C]$
ADD	R_2	D	$R_2 \leftarrow R_2 + M[D]$
MUL	R_1	R_2	$R_1 \leftarrow R_1 * R_2$
MOV	X	R_1	$M[X] \leftarrow R_1$

→ MOV is used for moving the do operand to and from memory and processor registers.

3) One Address Instruction

LOAD A → content of A will be stored in accumulator

STORE A → content of Accumulor will be stored in A

LOAD	A	$AC \leftarrow M[A]$	
ADD	B	$AC \leftarrow AC + M[B]$	{Acumulator A+B}
STORE	T	$M[T] \leftarrow AC$	
LOAD	C	$AC \leftarrow M[C]$	
ADD	D	$AC \leftarrow AC + M[D]$	(Acumulator C+D}
~~STORE~~			
MUL	T	$AC \leftarrow AC * M[T]$	
STORE	X	$M[X] \leftarrow AC$	

4) Zero Address Instruction (back)

ADDRESSING MODES

- The operation field of an instruction specifies the operation to be performed. This operation must be executed on some data stored in computer register or memory words. The way the operands are chosen during program execution is dependent on the addressing mode.
- It specifies a rule for interpreting or modifying the address field of the instruction before the operand is actually executed
- computer use addressing modes for 2 purposes:
  1) To reduce the no. of bits in the addressing field of instruction
  2) To give programming versatility to the user by providing such facilities as pointers to memory, counters for loop control, indexing of data etc
- To understand various addressing modes its necessary to go through instruction cycle, that is divided in 3 parts:
  * Fetch the instruction from memory
  * Decode the instruction
  * Execute the instruction

→ Various Addressing Modes

① Implied Mode

- In this mode the operand are specified implicitly in the definition of the instruction.
- For example instruction CMA (complement Accumulator) is an implied-mode instruction bcoz the operand in the accumulator register is implied in the definition of the instruction.
- It used for 0 or 1 address registers.

ex

INC A (Increment Accumulator)

AC ← AC + 1

ADD

Instruction

Data

② Immediate Addressing Mode

- In this mode operand is specified in the instruction itself.
- Operand is directly provided as a constant
- No computation is required to calculate effective Address
- Size of the operand/constant depends on the size of Address field.

opcode	Addres

→ Data is directly stored here

③ Register Mode

- Here the operands are in registers that resides within the CPU.
- Register No. is written in the instruction.

opcode	operand

000000 (Register nos.)

000000

200

111111

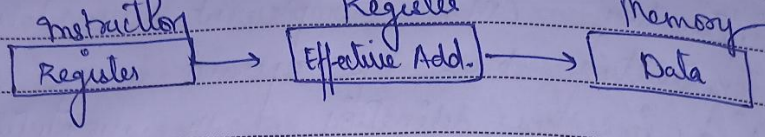
R1

R2

16 bit

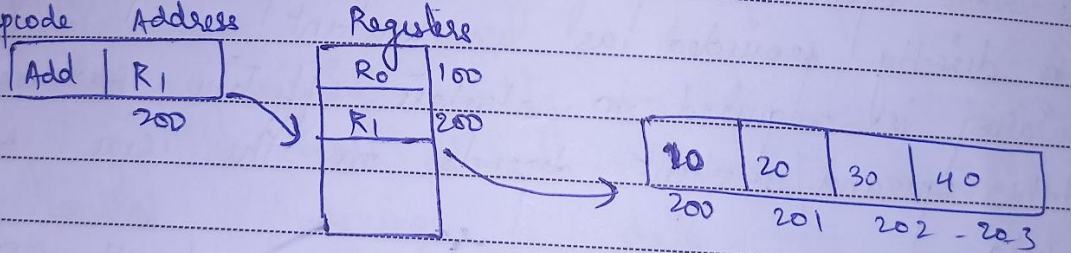
R16

* Register Indirect Mode: In this mode the selected register contain the address of the operand rather than the operand itself.



(4) Autoincrement or Autodecrement Mode:

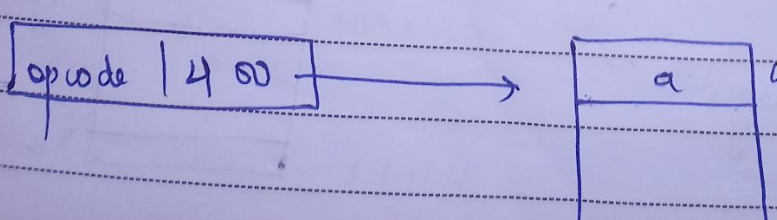
This is similar to the register indirect mode except that the register is incremented or decremented after/before its value is used to access memory. When the address stored in the register refers to a table of data in memory, it is necessary to increment or decrement the register after every access to the table. This can be achieved by using increment/decrement instruction



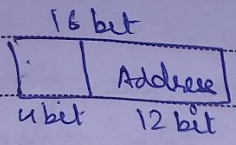
LD (R1)+

Direct Addressing Mode (Absolute Addressing mode)

In this mode the effective Address is equal to the address part of the instruction.
It is used to access variables.



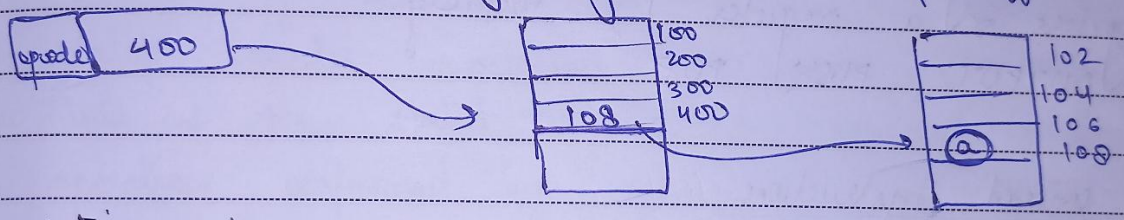
- limitation : size of address must be within the size of addressing part of instruction



size → $0 - 2^{12}$

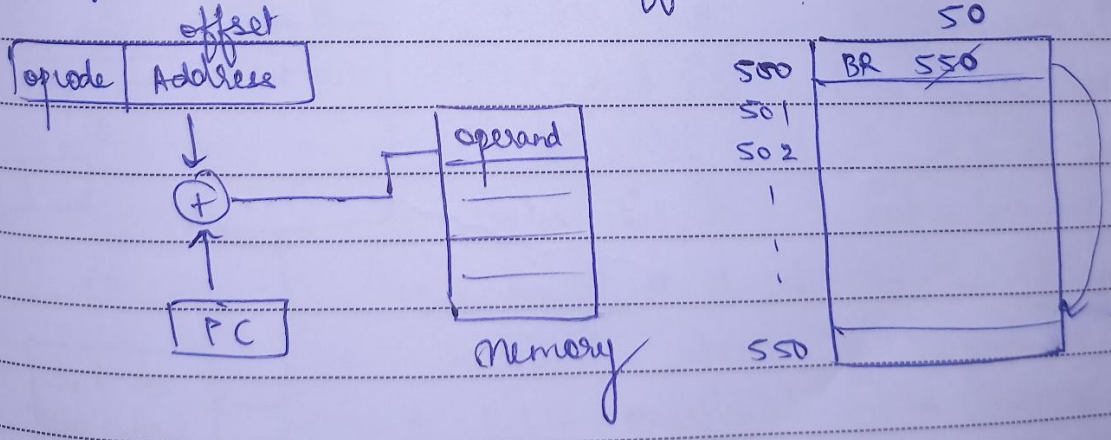
(6) Indirect Addressing Mode

- In this mode address field of the instruction gives the address where the effective Address is stored in memory.
- control fetches the instruction from memory & uses its address part to access memory again to read the effective address.



(7) Relative Address Mode

Effective Address = PC + offset



Note:

PC → shift bydefault by 1 so, offset → 49

COMPUTER INSTRUCTIONS

Computer instructions are classified in 3 categories :

Data Transfer Instructions :

These instructions are used to move data from one place in the computer to another without changing the data content

LOAD : It's used to transfer data from memory to accum

STORE : It's used " " " " accumulator to memo

MOV : It's has been used in computers with multiple CPU registers to designate a transfer from one register to anoth

XCH : (exchange) It swaps information b/w two registers or a register & memory

INPUT, OUTPUT, PUSH, POP

Program control Instructions

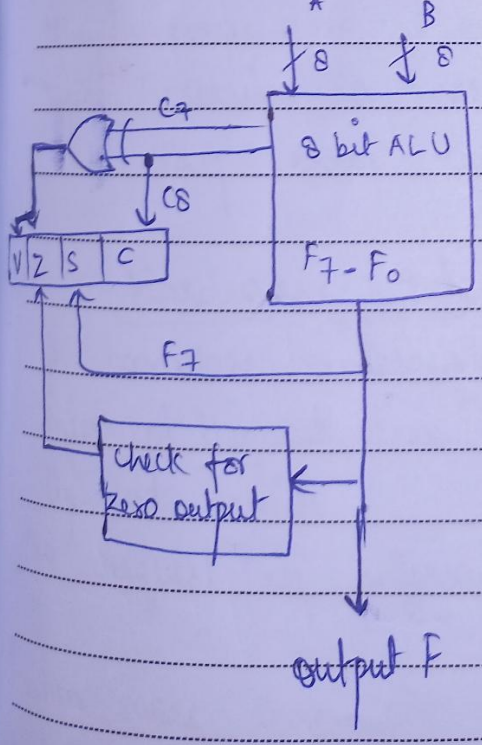
(3) Data Manipulation Instructions

- These instructions perform operations on data & provide the computational capabilities for the computer.
- They are divided into 3 parts.
  - * Arithmetic
  - * Logical & bit manipulation
  - * Shift

→ INR, DEC, ADD, SUB, MUL, DIV, ADDC, SUBB, CLR etc

STATUS BITS

- The status register compromises of status bits
- The bits of the status register are modified according to the operations performed by ALU.
- Block diagram of an 8 bit ALU with a 4-bit status register
- Four status bits are symbolized by C, S, Z, V



- bit C (carry) is set to 1 if the end carry C_8 is 1, if carry is 0 it's cleared to 0.
- Bit S (sign) is set to 1 if the highest order bit F is 1, otherwise 0.
- Bit Z is 1 if the output is 0, but it's 0 when the output is not zero
- Bit V (overflow) is 1 if the XOR of last two carries is equal to 1, other 0.
- Result of 8-bit ALU operation is either 127 or -127

INTERRUPTS AND IT'S TYPES

- The interrupt is a signal emmitted by hardware or software when a process or event needs immediate attention.
- It Alerts the processor to a high priority process requiring interruption of the current working process.
- In I/O devices one of bus control line is dedicated for this purpose & it's called (ISR) Interrupt Service Routine.
- when the devices raises an interrupt, at let's say process i, the processor first completes the execution of i. Then it loads the program counter with the address of the first instruction of ISR.

Software Interrupt

- A sort of interrupt called a software interrupt is one that is produced by software or a system as opposed to hardware.
- Traps and exceptions are other names for them.
- They serve as a signal for the operating system or a system service to carry out a certain function or respond to an error conditions.

Hardware Interrupt

- In hardware interrupt, all the devices are connected to the Interrupt Request line. A single request line is used for all the n devices.
- To request an interrupt, a device closes its associated switch.
- when more than one device raises an interrupt request signal, the additional information is needed to decide which device to consider first.

* Interrupt Nesting : Here I/O devices are organised in priority structure. Hence interrupt request from highest priority is recognized.

* Polling : devices encountered with IRQ is served first.

CHARACTERISTICS OF RISC/CISC

RISC	CISC
• It stands for Reduced instruction set Architecture	• It stands for complex instruction set Architecture.
• Simpler instruction, hence simple instruction decoding	• Complex instruction, hence complex instruction decoding.
• Instruction come undersize of one word	• Instruction are larger than one word size.
• Instruction take a single clock cycle to get executed.	• Instruction may take more than single clock cycle to get executed.
• More general purpose registers	• Less general purpose registers
• Simple Addressing modes	• Complex Addressing mode.
• Fewer data types	• More data types
• Focus on Software	• Focus on Hardware.
Pro 1) simple instructions	Pro 1) Reduce code size
2) Faster executions	2) widely used
3) Lower power consumption	
Con 1) more instruction required	Con 1) ↑ power consumption
2) ↑ memory usage	2) More complex
3) high cost	

PIPELINING PROCESSING

- Pipelining is a process of arrangement of hardware elements of the CPU such that its overall performance is increased.
- Simultaneous execution of more than one instruction take place in pipelined processor.
- Ex→ consider a bottle packaging plant. Let there be 3 stages that bottle should pass through I (insertion), F (filling), S (sealed)

→ without pipelining

I F S | | | | | |
| | | I F S | | |
| | | | | I F S

9/3 ⇒ 3 minutes

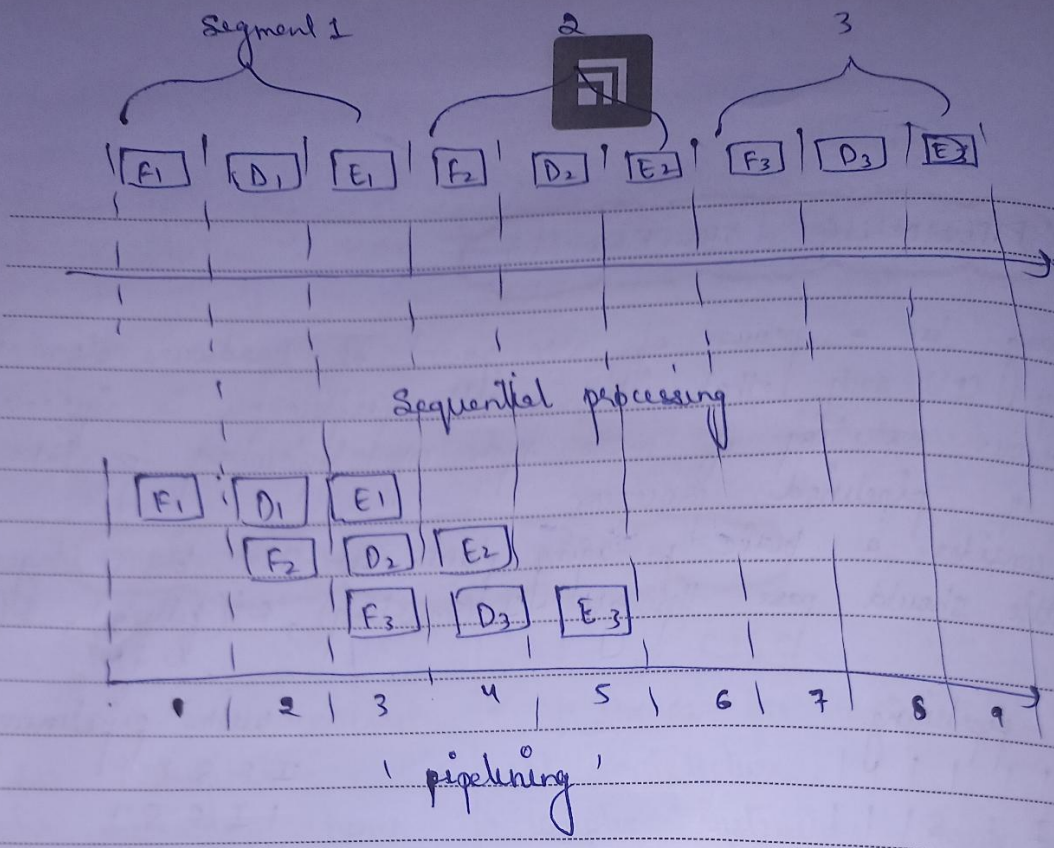
"with" pipelining

I F S | |
| I F S |
| | I F S

5/3 ⇒ 1.67 minutes

- Pipelining is a technique of breaking a sequential process into small fragments or sub operations. The execution of each of these sub-procedure takes place in a certain dedicated segment that functions together with all other segments.

→



Parallel Processing

- Parallel processing is a term used to denote a large class of techniques that are used to provide simultaneous data-processing task for the purpose of increasing computational speed of a computer system.
- Instead of processing each instruction sequentially, a parallel processing system is able to perform concurrent data processing to achieve faster execution time.

For example, while an instruction is being executed in the ALU, the next instruction can be read from memory.

System may have 2 or more ALU & be able to execute 2 or more instruction at the same time.

The purpose of Parallel processing is to increase

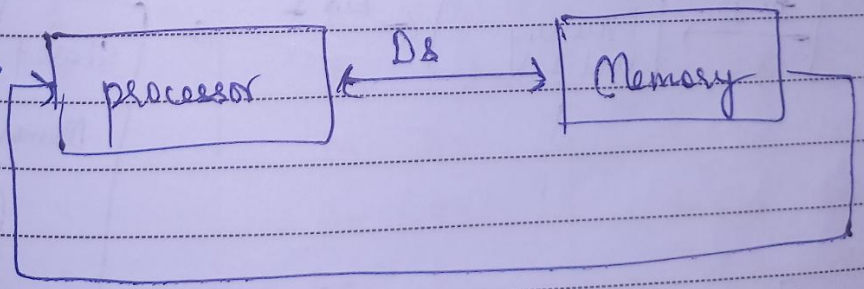
throughput, that is the amount of processing that can be accomplished during a given interval of time.

Flynn's Classification

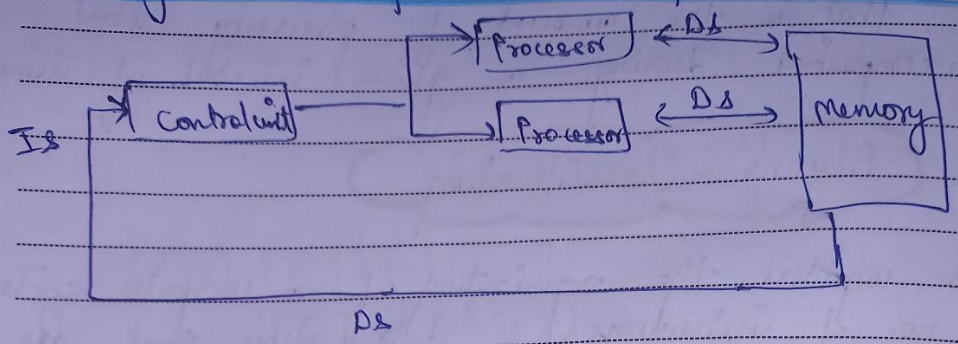
- M.J Flynn considers the organization of a computer system by the no. of instructions & data items that are manipulated simultaneously.
- The normal operation of a computer is to fetch instructions from memory & execute them in processor.
- The sequence of instructions read from memory constitutes an instruction stream.
- The operations performed on data in the processor constitutes a data stream.

→ It's divided into 4 parts

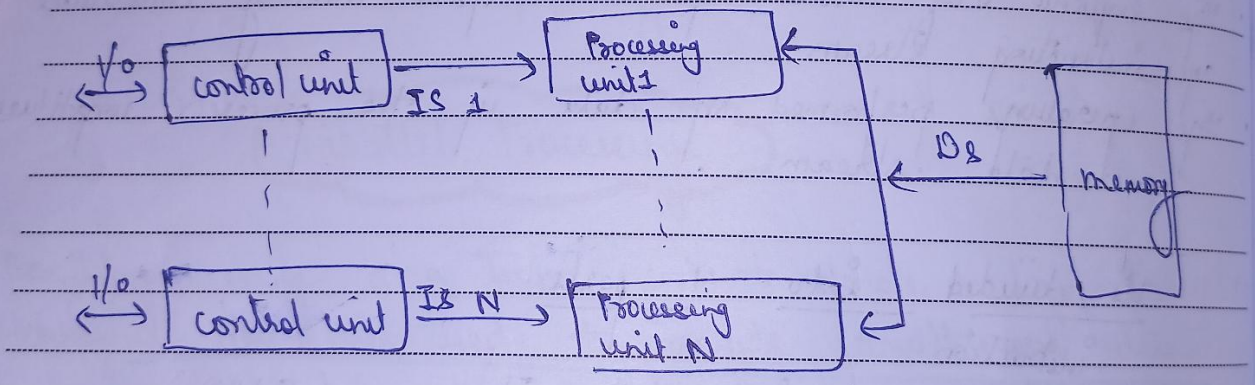
1) Single instruction stream, single data stream (SISD)



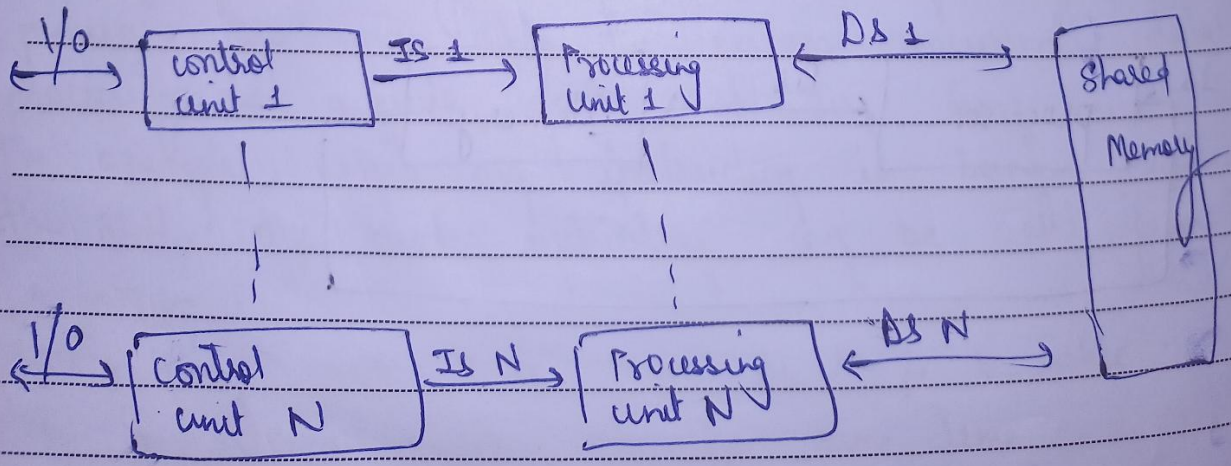
2) Single instruction stream Multiple data stream (SIMD)



3) Multiple instruction stream, single data stream (MISD)



4) Multiple instruction stream, Multiple data stream (MIMD)



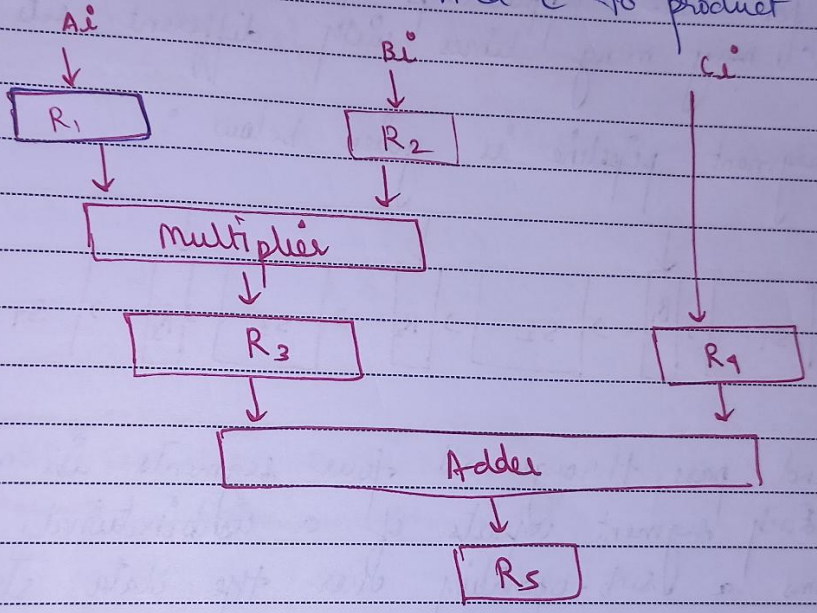
Ques Demonstrate $A_i * B_i + C_i$ for $i = 1, \ldots 7$ using pipeline concept.

$R_1 \leftarrow A$, $R_2 \leftarrow B$ Input A, B

$R_3 \leftarrow R_1 * R_2$, $R_4 \leftarrow C$ Multiply & input C

$R_5 \leftarrow R_3 + R_4$ Add C to product

figure



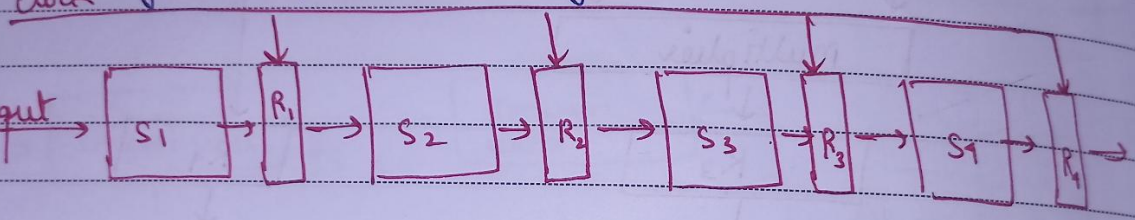
table

Clock pluse No.	Segment 1 R_1	R_2	Segment 2 R_3	R_4	Segment 3 R_5
1	A_1	B_1			
2	A_2	B_2	$A_1 * B_1$	C_1	C_1
3	A_3	B_3	$A_2 * B_2$	C_2	$A_1 * B_1 + C_1$
4	A_4	B_4	$A_3 * B_3$	C_3	$A_2 * B_2 + C_2$
5	A_5	B_5	$A_4 * B_4$	C_4	$A_3 * B_3 + C_3$
6	A_6	B_6	$A_5 * B_5$	C_5	$A_4 * B_4 + C_4$
7	A_7	B_7	$A_6 * B_6$	C_6	$A_5 * B_5 + C_5$
8			$A_7 * B_7$	C_7	$A_6 * B_6 + C_6$
9					$A_7 * B_7 + C_7$

PIPELINING GENERAL CONSIDERATIONS

- Any operation that can be decomposed into a sequence of sub operations of about the same complexity can be implemented by a pipline processor. This technique is efficient for those applications that need to repeat the same task many many times with different sets of data

→ 4 stage segment pipeline is given below :



The operand pass through all four segments in a fixed sequence. Each segment consists of a combinational circuit S, that performs a sub-operation over the data stream flowing through the pipe.

The segments are separated by registers R, that hold the intermediate results b/w the stages.

Task is defined as the total operation performed going through all the segments in the pipeline.

→ **Space-Time diagram for pipeline**

Segment	1	2	3	4	5	6	7	8	9	clock
1	T_1	T_2	T_3	T_4	T_5	T_6				
2		T_1	T_2	T_3	T_4	T_5	T_6			
3			T_1	T_2	T_3	T_4	T_5	T_6		
4				T_1	T_2	T_3	T_4	T_5	T_6	

Ques find the no. of clock cycles when segments & task are given.

$T = 6$ $\quad S = 4$

clock cycles $= k + (n-1)$

$= s + (T-1) \Rightarrow 4 + 6 - 1$

$4 + 5 = 9$ Ans

ARITHMETIC PIPELINE

Ques $x = 0.9504 \times 10^3 \rightarrow$ mantissas

$y = 0.8200 \times 10^2 \rightarrow$

Step 1 : compare the powers, & y has less power $(3-2) = 1$

so, we have to shift y by 1

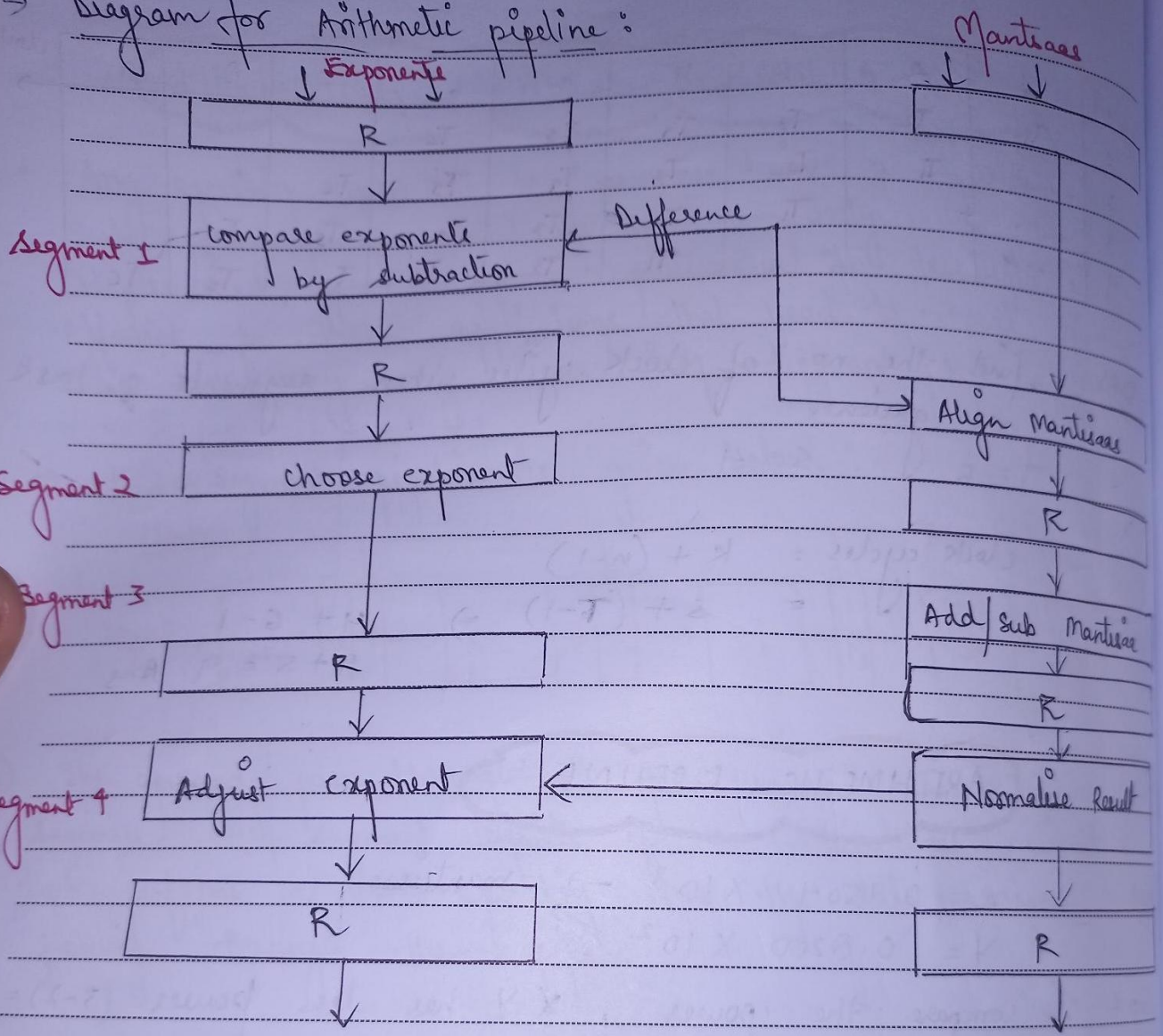
$x = .9504 \times 10^3 \quad y = 0.0820 \times 10^3$

Step 2 Now add both mantissas

$z = 1.0324 \times 10^3$

Step 3 Normalize the result, so that it can be a fraction with nonzero first digit $\quad z = 0.10324 \times 10^4$

→ Diagram for Arithmetic pipeline :



INPUT - OUTPUT ORGANISATION

Peripheral Devices

All the devices, that are connected to the computer system whether they are used for taking the input or displaying the output are called peripheral devices.

- All the input/output devices attached to the computer system, designed to read information into or out of the memory unit upon command from the CPU are called peripheral Devices.
- Peripheral that provides auxiliary storage → magnetic disk, tapes

1) **Monitor & keyboard**:
- Video monitor are most commonly used peripherals, consist of keyboard as input device & display unit as output devices.
- most popular Monitor is CRT (cathode ray tube).

2) **Printer**:
- Printers provide a permanent record on paper of computer output data/text.

Types: daisywheel, laser, dot matrix

3) **Magnetic tape & disk**
- Magnetic tapes are used mostly for storing files of data for example a company's payroll record.
- Magnetic disks are used mostly for bulk storage of program & data

I/O INTERFACE

- Input-output interface provides a method for transferring information b/w internal storage & external I/O devices.
- Peripherals connected to a computer need special communication links for interfacing them with CPU. The purpose of the communication link is to resolve the differences that exist b/w the central computer & each peripheral.

→ Major differences are:

1) Peripherals are electromechanical & electromagnetic devices & their manner of operation is different from the operation of the CPU & memory, which are electronic devices. The conversion of signal values may required

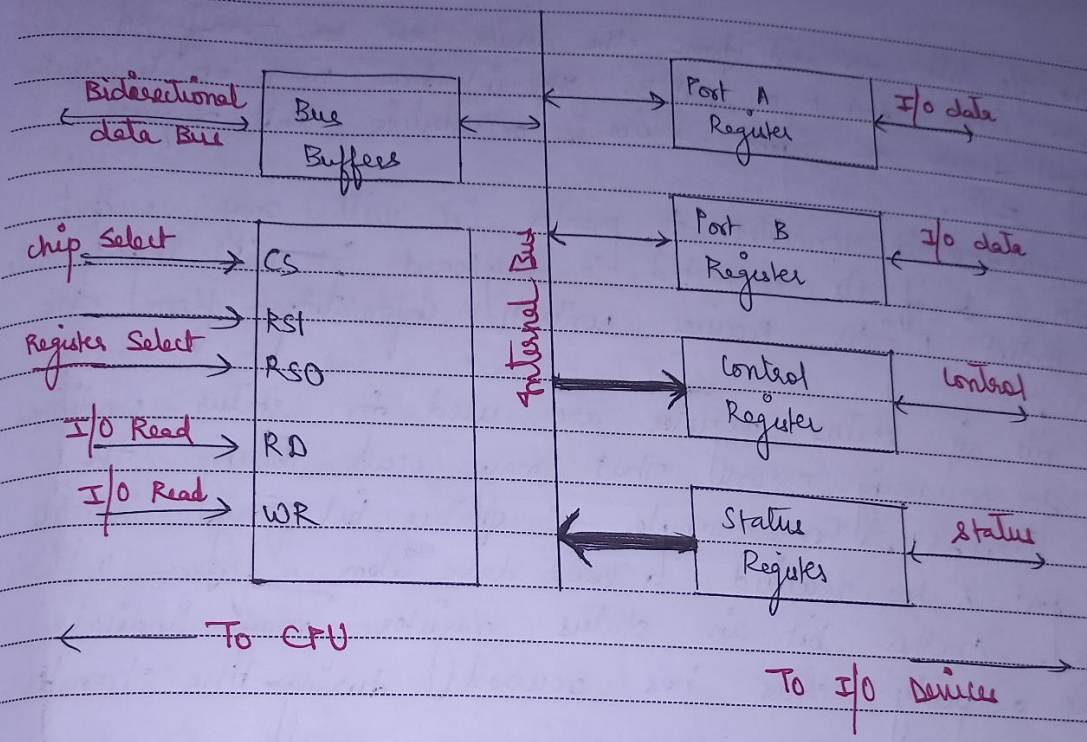
2) The data transfer rate of peripherals is usually slower than the transfer rate of CPU & consequently a synchronization mechanism may be needed

3) Data codes & formats in peripherals differ from the word format in the CPU & memory.

4) The operating modes of peripherals are different from each other & each must be controlled so as not to disturb the operation of other peripherals connected to CPU.

To Resolve these difference, computer system include special hardware components b/w the CPU & peripherals to supervise & synchronize all input & output transfers, these components are interface units because they interface b/w processor bus & peripheral devices.

block diagram



CS	RS1	RS0	Register Selected
0	X	X	~~Port A Register~~ (None data bus in high impedance)
1	0	0	Port A Register
1	0	1	Port B "
1	1	0	Control "
1	1	1	Status Register

- The I/O data to and from the device can be transferred into either port A or port B. The interface may operate with output device or input device, or device that require both I/O.
- If interface is connected to printer, it will only output data, & if it's connected to keyboard it will input data
- The control Register Receives control information from the CPU.
- The bits in status register are used for status conditions & for recording errors that may occur during the data transfer. For example, a status bit may indicate that port A has received a new data item from I/O device. Another bit in status register may indicate that a parity error has occured during the transfer.

ASYNCHRONOUS DATA TRANSFER

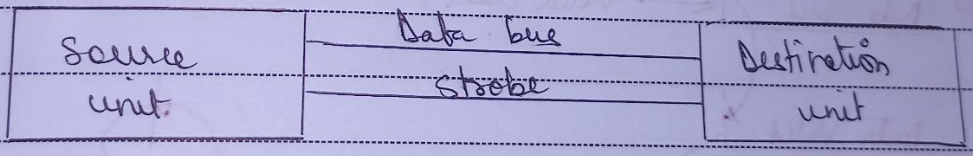
- Two units such as a CPU & I/O interface are designed independently of each other. If the registers in the interface shares a common clock pulse with the CPU registers, the data transfer b/w two units is called Synchronous.

In most cases the internal timing in each unit is independent from the other in that each uses its own private clock for internal registers.

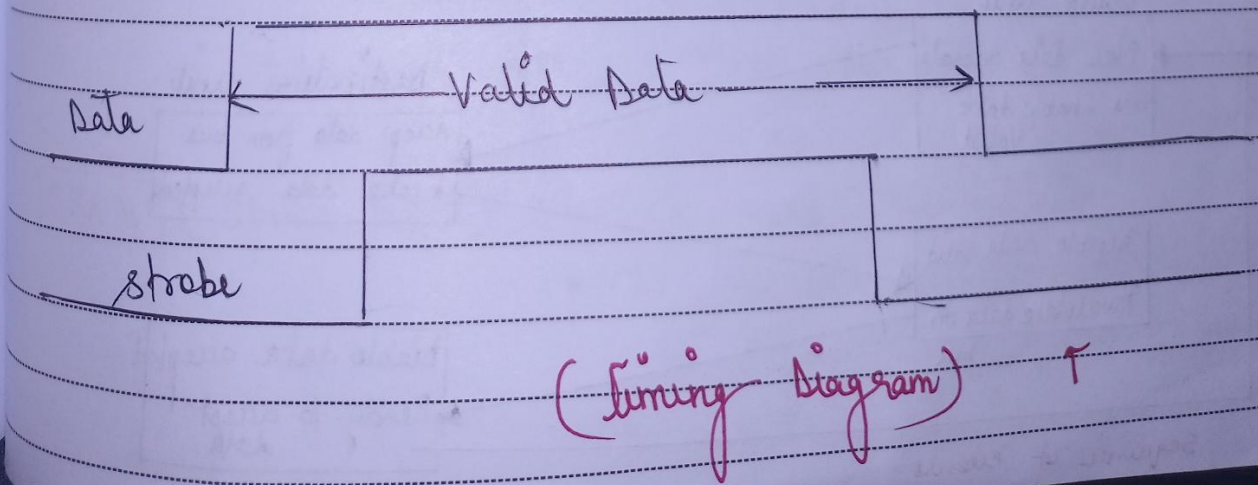
In that case the two units are said to be asynchronous to each other.

① Strobe control

- The strobe control method of asynchronous data transfer employs a single control line to time each transfer. The strobe may be activated by source or destination unit.
- The data bus carries the binary information from source unit to the destination unit. Typically the bus has multiple lines to transfer an entire byte. The strobe is a single line that informs the destination unit when a valid data word is available in bus.
- The source unit first places the data on the data bus. After a brief delay to ensure that the data settle to a steady value, the source activates the strobe pulse.
- The information on the data bus & the strobe signal remain in the active state for a sufficient time period to allow destination unit to receive data



(Block Diagram) ↑

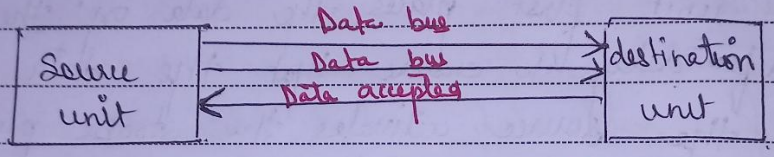


(Timing Diagram) ↑

Handshaking :

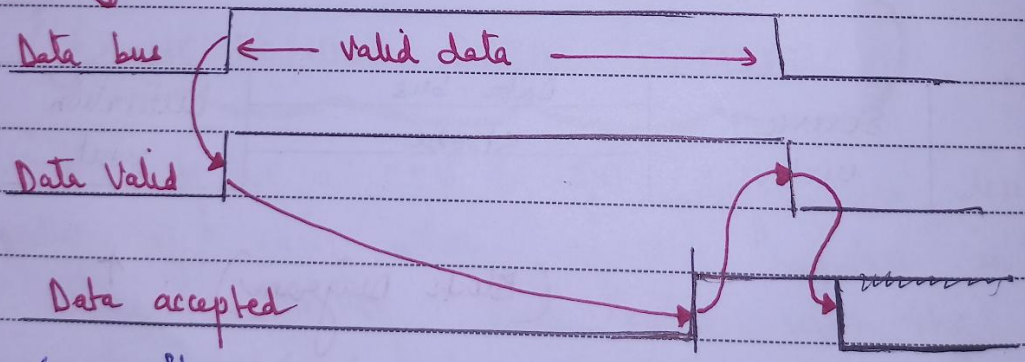
The dis advantage of the strobe method is that the source unit that initiates the transfer has no way of knowing whether destination unit has actually received the data item that was placed in the bus.

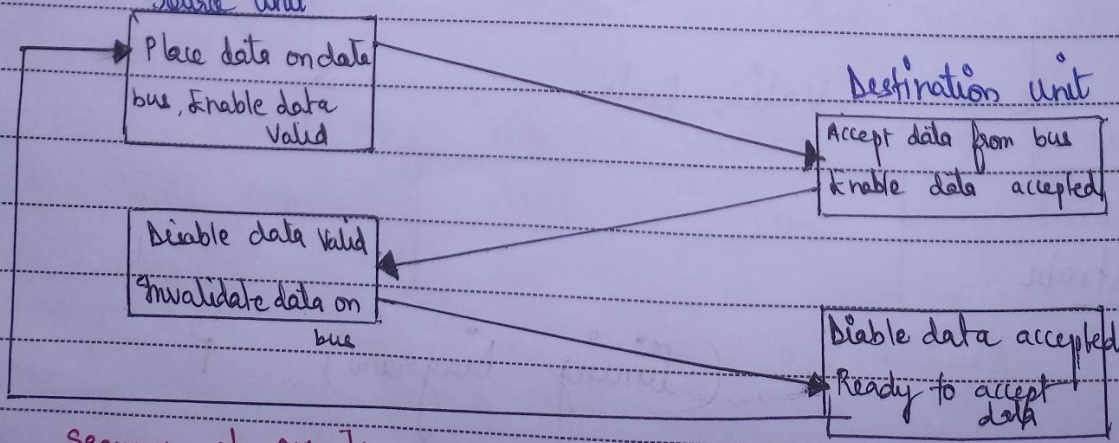
The 2 handshaking lines are data valid, which is generated by source unit, and data accepted generated by the source unit.



↑ [Block diagram]

Timing Diagram ↘

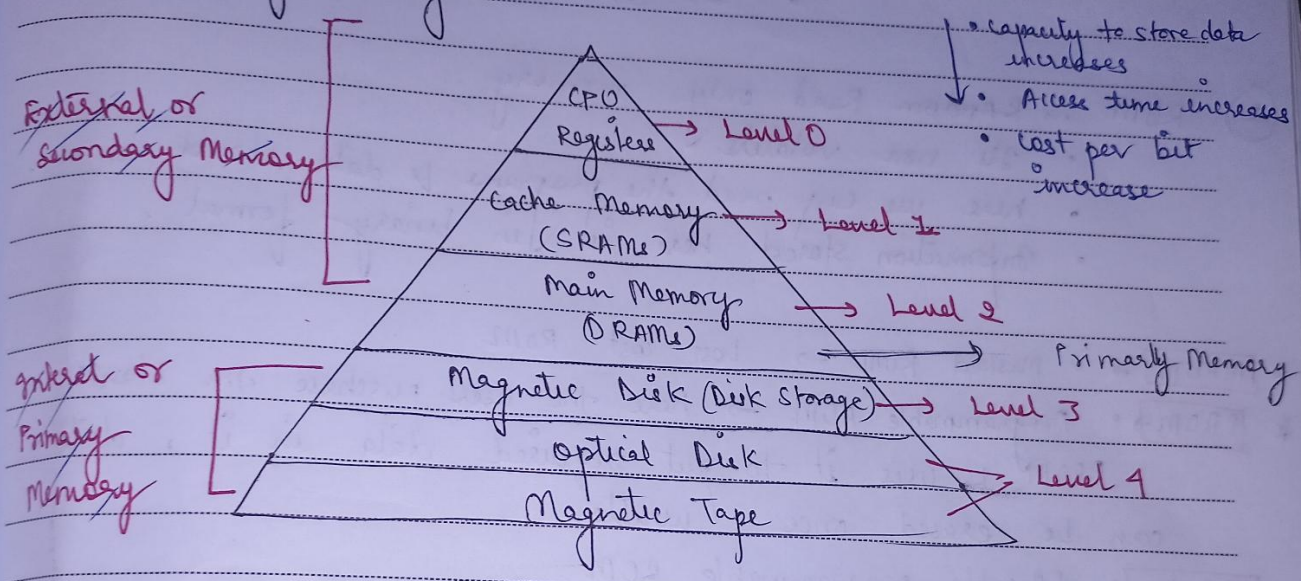




Sequence of events

MEMORY ORGANIZATION IN COMPUTERS

Hierachy Design



Main Memory

- It's also called primary memory, which is used to store data, program or instruction during computer operation.

RAM : • Random Access Memory
- It's volatile in nature (If power supply cuts, data loss occurs)
- It stores data temporarily.
- It's used for booting or start the computer

SRAM (Static RAM) : • It uses transistors
- It has no. of flip flops containing 1 bit
- It's faster

* DRAM (Dynamic RAM) : • It uses capacitor & transistors
• It contains thousands of memory cells.
• It's slow than SRAM

(2) ROM : • RAndom Read only memory.
• It's non volatile
• here we can read the programs & data stored on it.
• Information stored here is in binary formal.

* MROM → • Masked ROM → Lost cost ROM
* PROM → • Programmable ROM → here the user purchase the blank PROM & uses it to put required data in it, data can be erased once written.
* EPROM → • ERasable programmable ROM
• U can erased its content by exposing it to UV Rays
* EEPROM → • Electrically Erasable Programmable ROM
• written contents can be erased electrically (4-10) ms