

Date: / /

Computer organisation TCS 404
Computer is a machine which convert data into a meaningful form (information) with the help of instruction.

MIPS (Million Instruction Per Second) - It is a method of measuring the raw speed of a computer's processor.

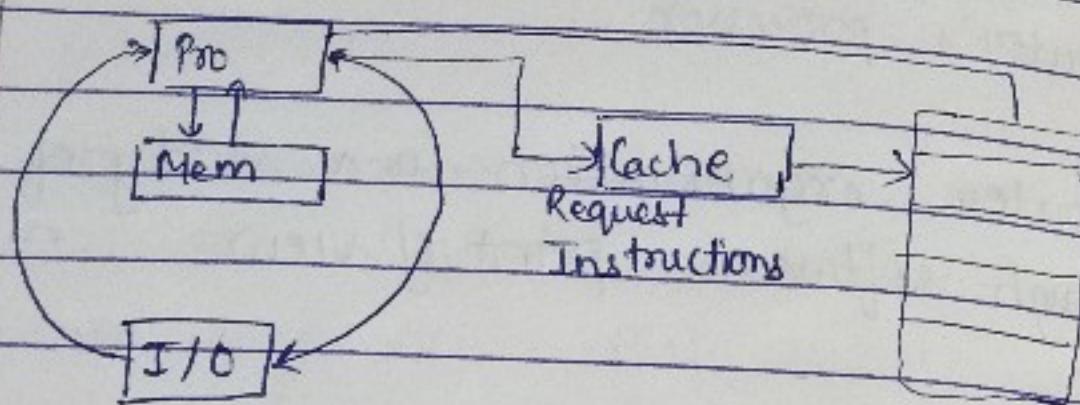
Computer organization is a study of system through software point of view.

Computer Organisation	Architecture	Computer Organisation	Architecture
^{Hardware} 1. Software point of view		^{Software} 1. Hardware point of view	
2. It helps us to understand the functionalities of a system.		2. It tells us how exactly all the units in the system are arranged & interconnected.	
3. It deals with high-level design issues.		3. It deals with low-level design issues.	

Processor request the data from memory & memory accepts the requesting instruction & give the data to the processor. The processor calculate very fast but the processor has to wait as the memory is slow so until

Date: / /

the data is fetched, cache memory comes into existence. The cache memory lies b/w processor & main memory. The internal architecture design of computer system differ from one another but the basic function & organisation remain same for all.



Computer

fixed

stored

ALU

all

ADD

Bas

1. RE

for

me

C

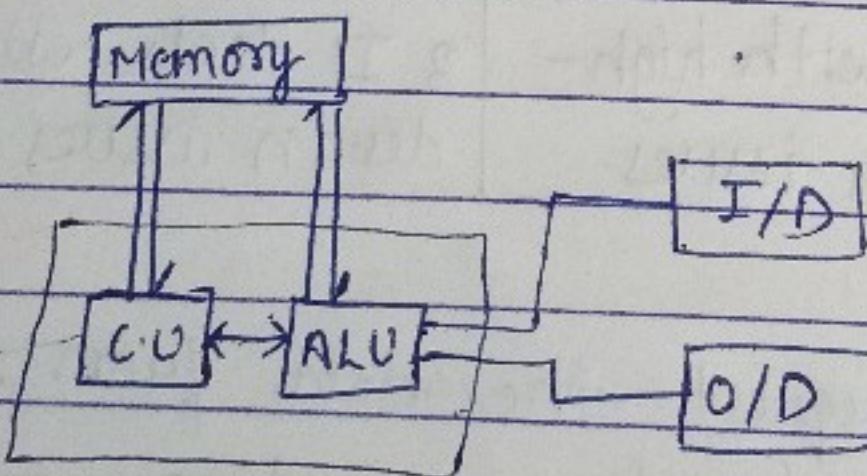
ad

vi

M

ac

The modern computers are based on stored program concept introduced by Sir John Von Neumann



Input Devices - It accept the list of instruction & data from the user, converts these instructions & data in computer acceptable format & supply

the converted instruction & data to the computer for further processing.

Output Devices - Accepts the result produced by the computer which are in coded form & cannot be easily understand by the user. Converts this coded result to human acceptable format.

CPU -

CU (control unit) - It handles all processor control signal. It directs all input & output flow, fetches code for instruction & controlling how data moves around the system.

ALU - The ALU is that part of CPU that handles all the calculation the CPU has to perform. eg.

ADD, SUB, COMP, MOV

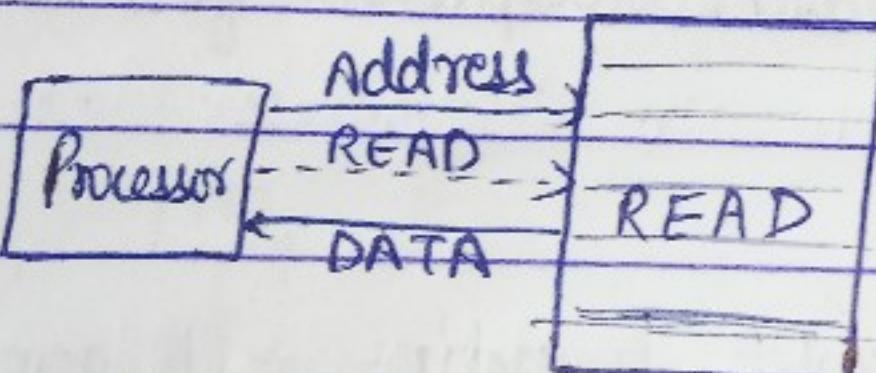
Basic operations of CPU

1. READ - The CPU addresses the memory both for memory READ & WRITE operation, so a memory READ operation is :-
CPU first send address of the location (memory address) & then sends the memory READ signal. On receiving this the memory starts reading from the location pointed to by the address. After Access Time the content of the

Date: / /

location put up by the memory on the data lines.

- * Access Time - It is the time how long it takes for a character in RAM (any memory) to be transferred to or from the CPU. Fast RAM chip has access time of $\leq 10\text{ms}$.



Evolution of Computers

1. First generation (vacuum tube) 1940-1956 - The 1st generation computer system used vacuum tubes for circuitry & magnetic drums for memory, & were often enormous, taking up entire rooms. These computers were very expensive to operate & generated a lot of heat, which was often the cause of malfunctions. These computers relied on machine language to perform operations, & they could solve one problem at a time eg - UNIVAC, ENIAC etc.
2. Second generation (Transistors) (1956-1963) - Vacuum tubes were replaced by Transistors in this generation. The transistor was far superior to the vacuum tube, allowing computers to become smaller, faster, cheaper, more energy-efficient & more reliable than their first-generation predecessors. Though the transistor still generated a great deal of heat that subjected the computer to damage, it was a vast improvement over the vacuum tube. 2nd generation computers moved from cryptic binary machine language to symbolic, or assembly language which allowed programmers to specify instruction in words.
3. Third generation (Integrated circuits) (1964-1971) - The development of the IC was the hallmark of the 3rd generation of computers. Transistors were

Date: / /

miniaturized & placed on silicon chips, called semiconductors which drastically increased the speed & efficiency of computers. Instead of punched cards & printouts, users interacted with 3rd generation computers through keyboards & monitor & interfaced with an O.S., which allowed the device to run many different applications at one time with a central program that monitored the memory.

Fourth generation (Microprocessor) (1971 - present) -

The 4th generation computer took advantage of the invention of the microprocessor or C.P.U. Microprocessor along with IC, helped make it possible for computers to fit easily on a desk.

These small computers became more powerful, they could be linked together to form networks which eventually led to the development of Internet. 4th generation computers also saw the development of GUI, mouse & handheld devices.

Fifth Generation (Artificial Intelligence) (Present &

Beyond) - 5th generation computing devices, based on AI are still in development, though there are some applications, such as voice recognition that are being used today. Quantum computation & molecular & nanotechnology will

Date: / /

radically change the face of computers in years to come. The goal of 5th generation computing is to develop devices that respond to natural languages input & are capable of learning & self-organization.

Types of Computer

On the basis of data handling capabilities, the computer is of 3 types -

1. Analogue Computer - These computers are designed to process analog data. It is the data that changes continuously & cannot have discrete values such as speed, temp. Analogue computer directly accept the data from the measuring device without first converting it into numbers & codes.

2. Digital computer - Digital computer is designed to perform calculations & logical operations at high speed. It accepts the raw data as digits or no & processes it with programs stored in its memory to produce output. eg. laptops & desktops

3. Hybrid computer - It has features of both analogue & digital computer. It is fast like analogue computer & has memory & accuracy like digital computer.

It can process both continuous & discrete data so it is widely used in specialized applications where both analogue & digital data is processed.

Date: / /

- On the basis of size, the computer is of 5 types:
1. Supercomputer - These are biggest & fastest computers. They are designed to process huge amt. of data. A supercomputer can process millions of instructions in a sec. It has thousands of interconnected processors. These are particularly used in scientific & engineering applications.
 2. Mainframe computer - These are designed to support hundreds or thousands of users simultaneously. They can support multiple programs at the same time. It means they can execute different process simultaneously.
 3. Miniframe computer - It is a midsize multiprocessor computer. It consists of 2 or more processors & can support 4 to 200 users at one time. Miniframe computers are used in institutes & departments for the tasks such as billing, accounting & inventory management.
 4. Workstation - Workstation is a single user computer that is designed for technical or scientific applications. It has faster microprocessor, large amount of RAM & high speed graphic adapters. It generally performs a specific job.

Date: / /

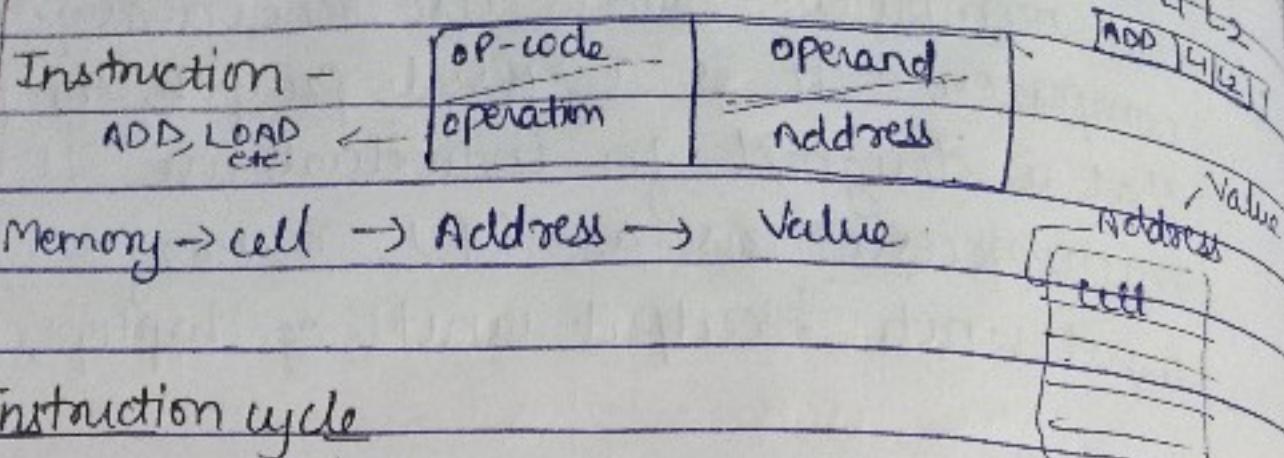
with great expertise

5 Microcomputer - It is also known as personal computer. It is a general purpose computer that is designed for individual use. It has a microprocessor as a CPU, memory, storage area, input unit & output unit. eg. Laptop, ^{desktop} Computer

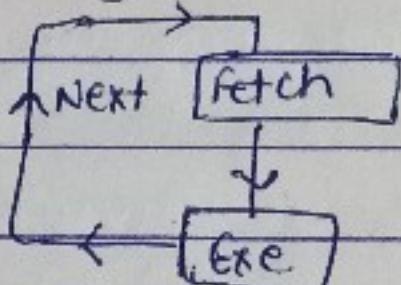
Memory = Register + Cache + PM + S.M

Date: / /

Registers - Address, operands, data are stored in register.

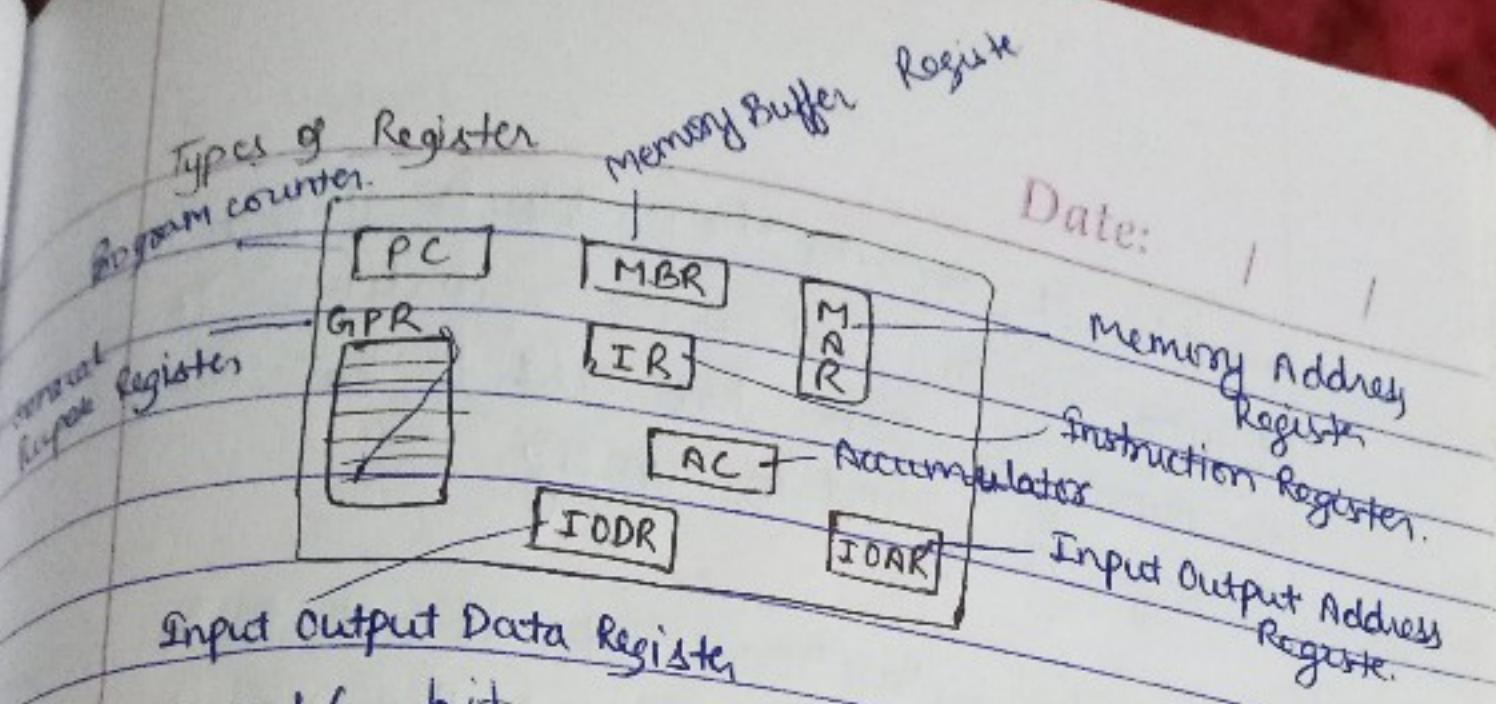


Instruction cycle



- * Bit pattern in the op code field identifies it as the type of instruction.
- * The other 2 field identify the location where the other 2 operands are available.

Register - A type of a computer memory used to quickly accept, store, transfer data & instruction that are being used immediately by the CPU. A register have instruction, storage, address, data, operands & global variable. Registers are basically a group of flip-flop capable of storing one bit of information.



Input Output Data Register

$$AC = 16 \text{ bit}$$

$$PC = 12$$

$$MAR = 16$$

$$MBR = 16$$

$$IR = 16$$

$$GPR = 16$$

$$IODR = 8+8=16$$

$$IOAR = 8+8=16$$

AC - It stores the results of calculation made by ALU. It also used as input registers to the adder

PC - Keeps track of the memory location of the next instruction to be dealt with. The PC ~~taken~~ then passes this next address to MAR. As soon as the instruction fetch is complete, the content of PC are incremented, so as to point to the next instruction address.

IR - The IR stores the current instruction fetched from memory. It stores the most recently fetched instruction while it is waiting to be coded & executed.

Date: / /

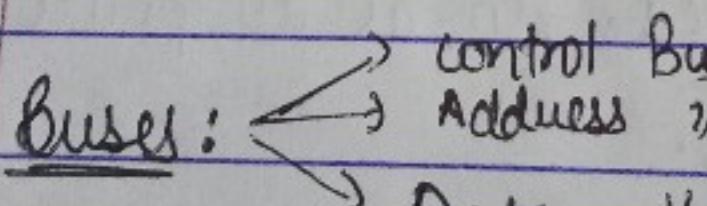
MAR - It contains the address of memory location during memory Read/Write operation. It stores the memory location that need to be transferred to & from stored in memory.

MBR - The instruction that is not to be immediately executed is placed in the buffer register. The data read from memory (during Read operation) or data to be written into memory.

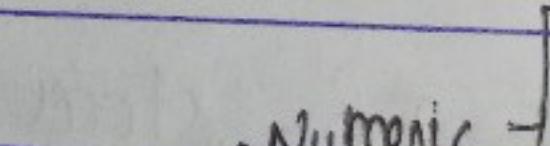
GPR - This is used for multiple purposes, storing operand, addresses, constants, global variables etc. In lieu to GPR the CPU also has some working registers known as Scratch Pad. These are used to keep the intermediate result within an instruction for cycle for complex instruction.

IOPR - It stores input, output data.

IOAR - " " " " address.

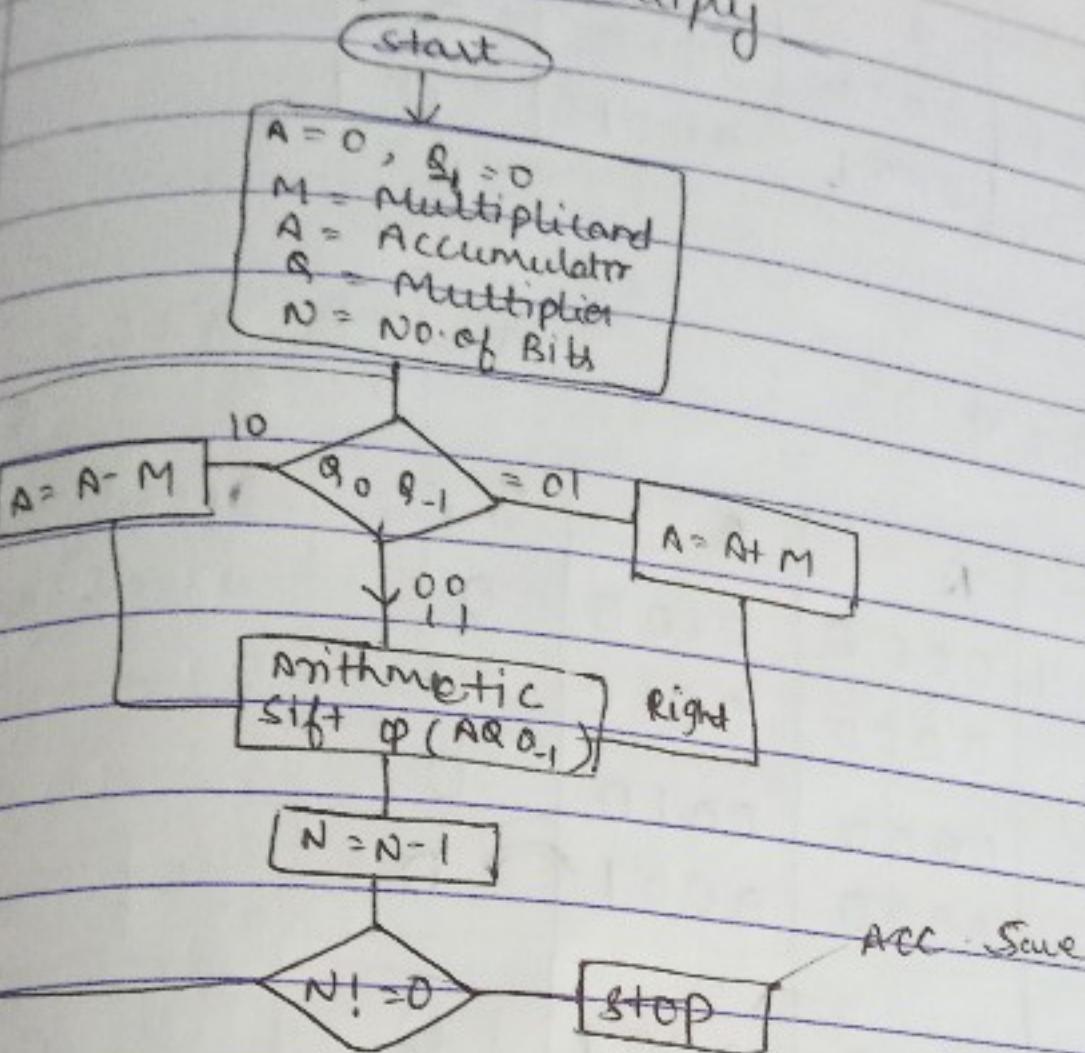
Buses:  control Bus (control signal transfer)
Address "
Data "

Data representation

Numeric -  Signed
Unsigned
Non-numeric

Booth's Algo for Multiply

Date: / /



$$\begin{array}{cccc|c}
 & Q_3 & Q_2 & Q_1 & Q_0 & Q_{-1} \\
 7 \times 3 = 21 & | & | & | & | & \\
 \end{array}
 \quad
 \begin{array}{l}
 7 = M = 0111_2 \\
 3 = Q = 0011_2 \\
 A - M = A + 2^3 M
 \end{array}$$

n	A	Q	Q ₋₁	Operation
n = 4	0000	0011	0	$A - M \Rightarrow A + 2^3 M$ 0000 $+ 1001$ \hline 1001 ACC
	1001	0011	0	ASR L.S.B (discard)

1.	n = 3	1100	1001	1	→ ASR (discard)
		1110	0100	1	A = A + M = 1110
3.	n = 2	1110	0100	1	discard
		0101	0100	1	discard
		0010	1010	0	0111 0101 → discard

Date: / /

Step	n	A	Q	Q_{-1}	Operation
4	$n=1$	0010	1010	0	
		[0001]	0101	0	↓21

Q. $(9 \times 4)^{-8}$

$$q = 1001 \quad a'2'c = \begin{array}{r} 100 \\ +1 \\ \hline \end{array}$$

ab11

Step	n	A	Q	Q_{-1}	Operation
1	$n=4$	0000	0100	0	→ discard
		0000	0010	0	
2.	$n=3$	0000	0010	0	
		0000	0001	0	
3.	$n=2$	0000	0001	0	$\begin{array}{r} 0000 \\ 0111 \\ \hline 0111 \end{array}$
		0111	0001	0	
		0011	1000	1	
4.	$n=1$	0011	1000	1	$\begin{array}{r} 0011 \\ 1100 \\ \hline 1001 \end{array}$
		1100	1000	1	
		0110	0010	0	$\begin{array}{r} 0010 \\ 1100 \\ \hline 1100 \end{array}$

Remove (MSB)

Q. 8x5

$$\begin{array}{r} 218 \\ 240 \\ 220 \\ \hline 710 \end{array}$$

$= 1000$

$$\begin{array}{r} 11 \\ 0111 \\ \hline 1000 \end{array} \quad 2^7 \text{ S com}$$

$$\begin{array}{r} 215 \\ 221 \\ \hline 110 \end{array}$$

0101

32bit

45072

		Date:	/	/
n	A	Q	Op	Operation
4	0000	0101	0	$A + 2^7 S.M$ = 0000 1000 1000
	1000	0101	0	
	1100	0010	1	
3	1100	0010	1	1100 1000 10100
	0100	0010	1	
	0010	0001	0	
2	0010	0001	0	0010 1000 1010
	1010	0001	0	
	1101	0000	1	discard
1	1101	0000	1	1000
	0101	0000	1	1101 0101
	0010	1000	0	$+2^5 \times 1^3$ = 32 + 8 = 40

Data Representation

1. Fixed Point is $\frac{1}{2}$ Radix point

Sign	Integer	Frac
31	30	1615

$$x = \underbrace{x_n}_{\text{(sign)}} \underbrace{x_{n-1} x_{n-2} \cdots x_1}_{\text{Integer}} \underbrace{x_0}_{\text{Fraction}} \underbrace{+ x_{-1} x_{-2} \cdots x_{-m}}_{\text{Fraction}}$$

sign integer frac
 $0001 \cdot 10^1$

$$x_3 \ x_2 \ x_1 \ x_0 + x_{-1} \ x_{-2} \ x_{-3} + \dots + x_{-m}$$

string 1000000000001001 · 101000000000000000000

$$-(1001 \cdot 101)_2 = (-9, 625)_{10}$$

~~550~~ 0.425
48
20
Date: 16

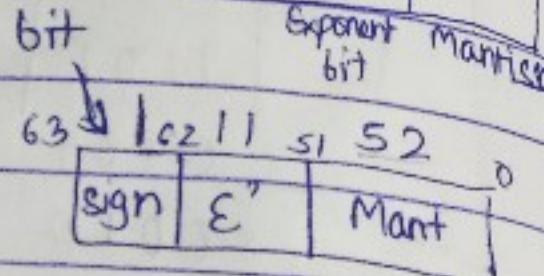
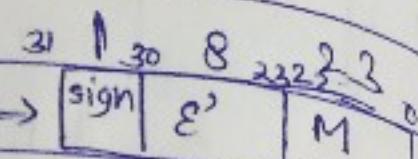
$$\begin{array}{r}
 100011.101 \\
 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\
 32 + 8 + 2 + 1 + \frac{1}{2} + \frac{1}{8} \\
 = 43 + \frac{5}{8} \quad \Rightarrow \quad -(43.625)_{10}
 \end{array}$$

2. Floating point Representation

I E E - 754

→ Single Precision

→ Double " - 64 bit



$$\epsilon' = \text{Exp} + \text{bias} \quad (\text{no. which is both } 32 \text{ & } 64) \\ + (127) \quad 32 \text{ bit}$$

$$E' = e + 1023 \quad (\text{for } 64 \text{ bit})$$

$$Q. \quad (1460 : 125)_{10}$$

$$1) \text{ No. to Binary} \Rightarrow (10110110100.001)_2$$

2) Normalise . 1.011011010000_x 2¹⁰ Expt

$$\text{Sign bit } (s) = 0$$

$$\text{Exp} = 10 \Rightarrow \varepsilon' = 10 + 127 = 137 \Rightarrow 1000/00$$

$$M = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

0. 00001001110110100000000000000000

for 64 bit

Date:

$$E' = 10 + 102 \cdot 3 = 1033 = 10000001001$$

0 | 10000001001 | 011 011 010000 | 000 = 1000000100

(1248.175),₁₀

1)	10011100000.00101100110	175x2 = 350	0	2 1248
	1: 001110000000	350x2 = 700	0	2 624
		700x2 = 1400	1	2 312
		400x2 = 800	0	2 156
		800x2 = 1600	1	2 78
		600x2 = 1200	1	2 39
		200x2 = 400	0	2 14
				2 9
				2 4
				2 2
				1 0

Binary → decimal

sign c' M

(ii) $s = 0 \Rightarrow$ the

$$2) \quad (10000110)_2 = (134)_{10}$$

$$e^3 = 134$$

$$c = 134 - 127 = 7$$

$$3) \quad 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} + 1 \times 2^{-5} \dots$$

•65625

$$4) (-1)^5 \times (1 + M) \times 2^e$$

$$-1^{\circ} \times (1 + 0.65625) \times 2^7$$

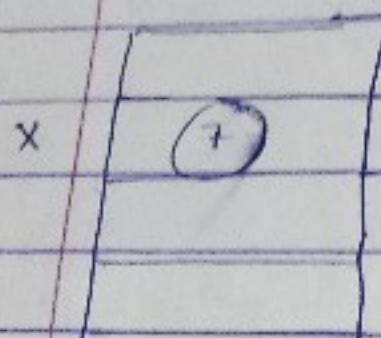
$$= 1.65625 \times 2^7$$

$$= \boxed{212} - \text{Ans}$$

10B@C

AB@BC+CA

ADD X



Date: / /
RTL (Register Transfer Language)

(address of X)

MAR \leftarrow X, memory

MBR \leftarrow M[MAR] } RT

AC \leftarrow AC + MBR.

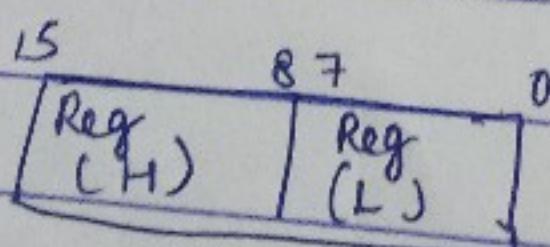
(u/op)

Microoperation - The operation executed on data stored in registers are called microoperation.

e.g. SHIFT, COUNT, CLEAR, LOAD. There are 4 types of microoperation.

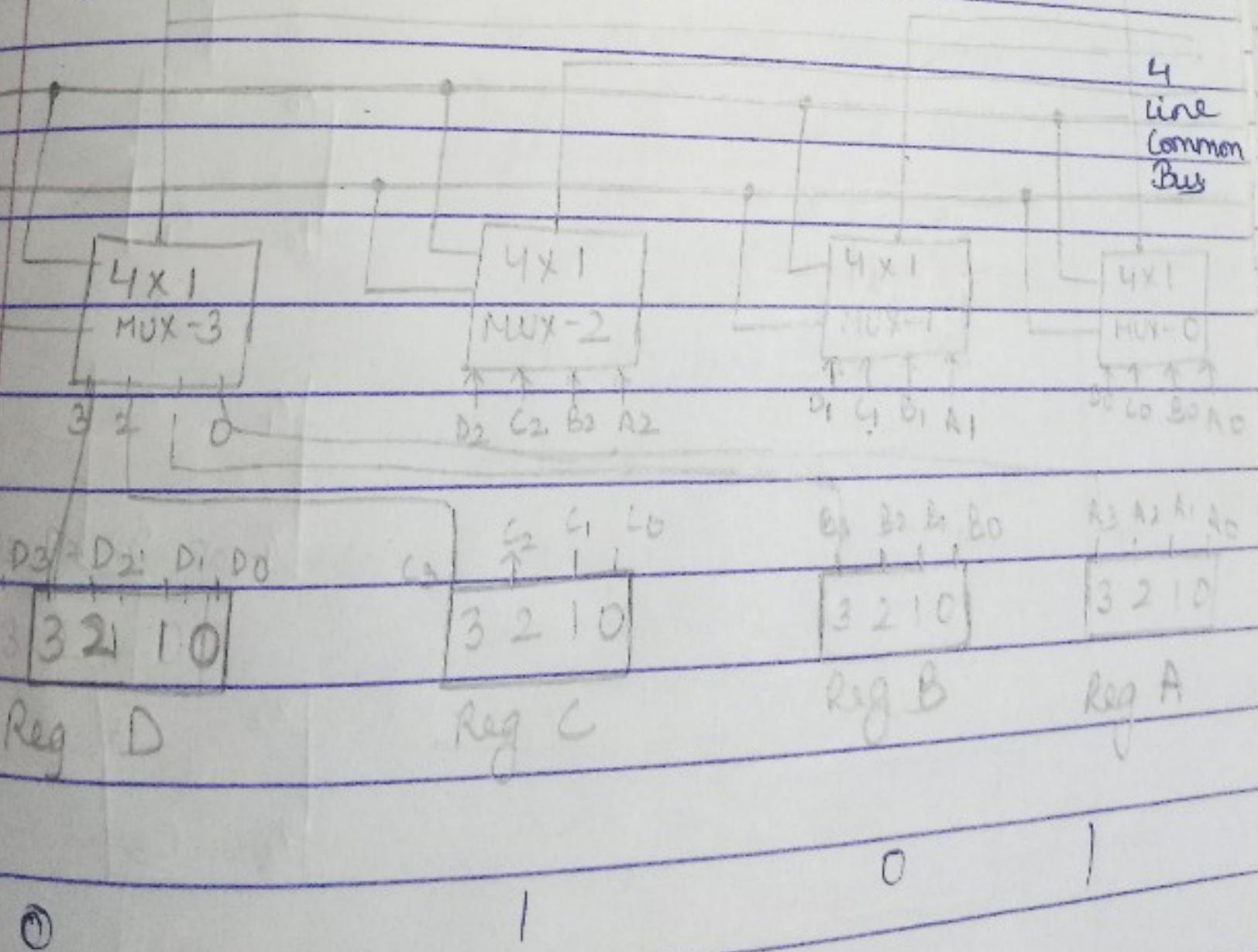
- (i) Register Transfer u/op - SHIFT, LOAD, MOV etc.
- (ii) Arithmetic u/op - ADD, MUL, MOD, SUB
- (iii) Logic u/op - AND, OR, NOT
- (iv) Shift u/op - shift register

RTL - It is a symbolic notation used to describe the u/op transfer among the register. It is not executed by the computer. RTL is a system for expressing in symbolic form the u/op sequence among the register of a digital module.



Date: / /

BUS & Memory Transfer
A computer has many registers & path must be provided to transfer information from one register to another. The no. of wires will be excessive if separate lines are used b/w each register, so the most efficient scheme for transferring information b/w register in a multi register configuration through which binary information is transferred at a time. (control) signal determine which register is selected by the bus during each particular register transfer. The one way of constructing a common bus system is with multiplexer. The multiplexer selects the source register whose binary information is then placed on the bus.

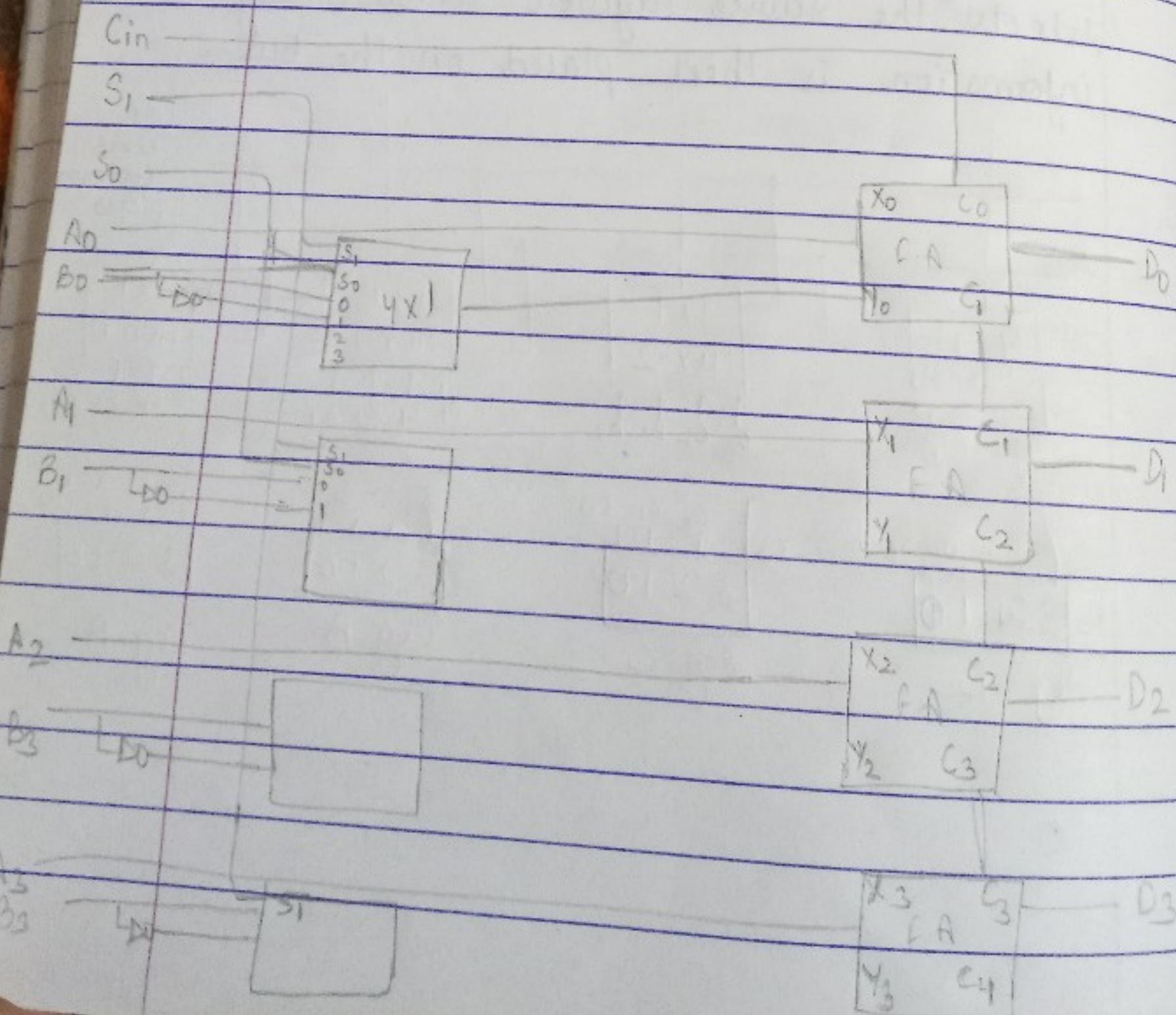


Date: / /

S	S ₀	Reg.
0	0	A
0	1	B
1	0	C
1	1	D

Arithmetic Circuit - By controlling the data inputs to binary adder, it is possible to obtain different types of arithmetic o/p. The output is calculated from the arithmetic sum.

So Based on Y changing value we can get different O/P, so different arithmetic operation can be done $[D = A + Y + C_{in} \rightarrow \text{any}]$.



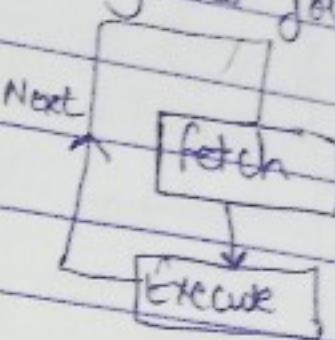
Contd

CPU organisation

Date: / /

Instruction Cycle - A program residing in the memory unit of the computer consists of sequence of instruction. Each instruction cycle is divided into sub-sequencing phases. They are followed as:

- i) Fetch the instruction.
- ii) Decode the instruction
- iii) Read effective address
- iv) Execute the instruction.



Interrupt

- Hardware
- Software
- Maskable
- Extend
- Internal
- Non-maskable

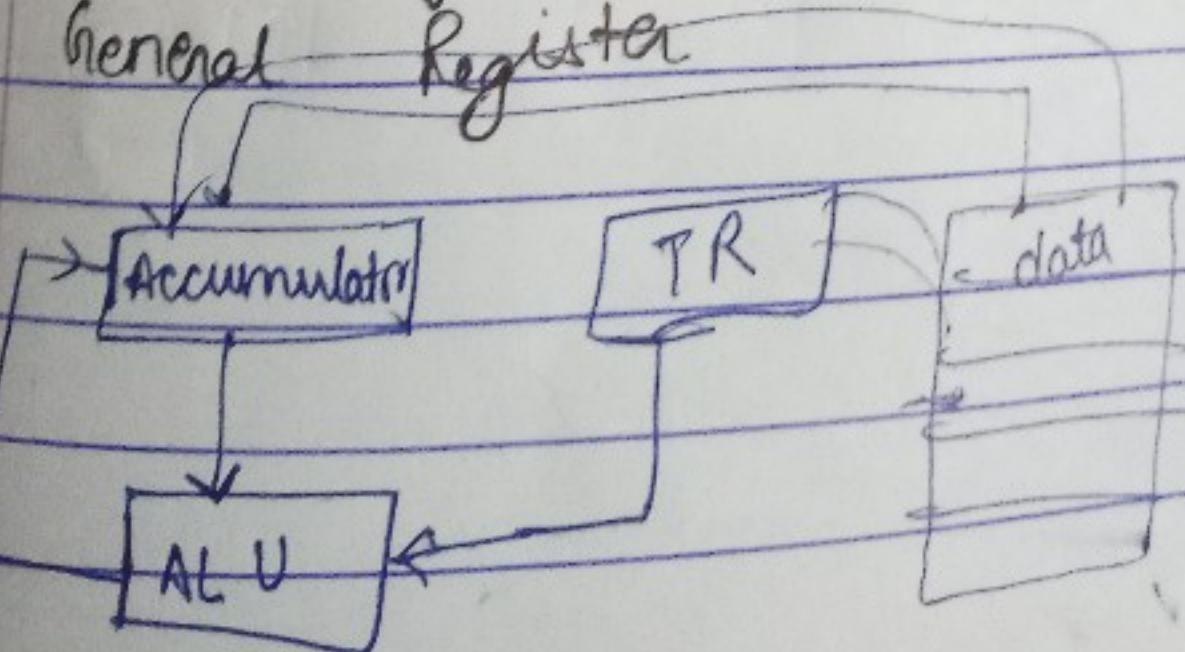
Interrupt is a concept can be said as an unusual event to disturb the normal flow of execution. Types of interrupt are -

Cpu organisation Architecture

1. Accumulator Machine

2. Stack organization

3. General Register



CU
↓
Instruction → Memory Referencing
Register " "

I/O "

[7] OPCODE Operated

D₀ - D₄ → Memory
(Data Line) D₇ → Register

Instruction Referencing in CU (control unit)

The program counter (PC) loaded with the address of first instruction in the program.

The sequence counter (SC) is cleared to zero providing a decoded time (T₀) after each clock pulse SC is incremented by 1

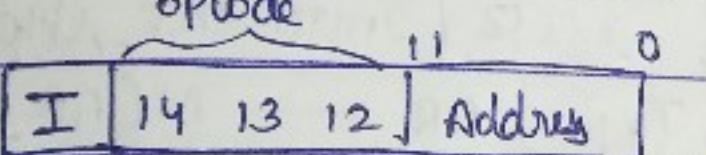
so that the timing signal go through a sequence T₀, T₁, T₂ ... & so on

T₀ → AR ← PC

T₁ → IR ← M[AR]

PC → PC + 1

T₂ → D₀ → D₇



If I₀ = 0 direct data accessing

I₀ = 1 Indirect " "

D ₀	0 0 0	AND
		D ₀ T ₄ : DR ← M[AR] SC → 0
		D ₀ T ₅ : AC ← AC AND DR
D ₁	0 0 1	ADD
		D ₁ T ₄ : DR ← M[AR] SC → 0
		D ₁ T ₅ : AC + DR E → 1
D ₂	0 1 0	LDA / LDR (Load)
		D ₂ T ₄ : DR ← M[AR] SC → 0
		D ₂ T ₅ : AC ← DR

BUN - Branch Unconditionally (Pass control
to different parts of program.)

Date: / /

D₃ 011 D₃ → Store [STR]
D_{3 T₄}: M[AR] ← AC

D₄ 100 D₄ → BUN
D_{4 T₄}: PC ← AR
D_{4 T₅}: PC ← PC + 1 $\xrightarrow{SC \rightarrow 0}$

D₅ 101 D₅-BSA [Branch & save Return Address]
D_{5 T₄}: M[AR] ← PC
AR ← AR + 1
SC ← 0
D_{5 T₅}: PC ← [AR]

D₆ 110 D₆ → ISZ [Increment Skip if zero]
D_{6 T₄}: DR ← M[AR]
D_{6 T₅}: DR ← DR + 1, M[AR] ← DR $\xrightarrow{SC \rightarrow 0}$
 $\text{If } DR = 0 ; PC \leftarrow PC + 1$

Control unit Architecture:

The timing for all the register in the basic computer is controlled by a MCG (Master Clock Generator). The clock pulses are applied to all the flip-flops & register in the system. The clock pulses do not change the state of a register unless the register is enabled by a control signal.

(i) Handwired - The control logic is implemented with gates, flip-flops, decoder & other

digital circuits.

Advantage: - Produce fast mode operation.

(ii) microprogram - The control logic is stored in control memory. The control memory is programmed to initiate the required sequence of operation. Any modification, updation done is to be performed by updating the micro-program in control memory.

Addressing Mode

- Microprocessor executes the instruction stored in memory i.e. RAM.
- It execute one single instruction at a time.
- Each of the instruction contain operation & operand.

These different ways of getting these operands referred in memory or addresses is addressing mode.

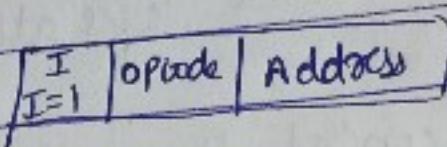
(i) Direct Addressing - Address field contains the address of operand, single memory reference to access data.

No addition calculation to work out effective address.

(ii) Indirect addressing - Memory all pointed to by address field contains the address of

Date: / /

the operand

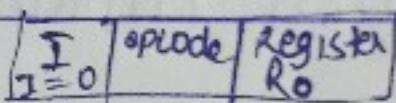


Pointers to

operand

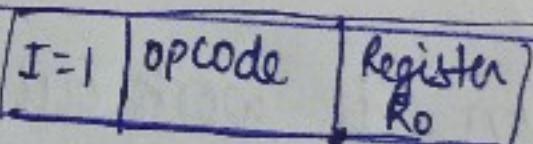
^{Direct}
Register Addressing

- Operand held in register named in address field.
- Limited no of register.
- Very small address field needed as shorter instruction.
- fast execution.



Register Indirect Addressing

- Operand is in memory ~~cell~~ self pointed to have content of register R₀
- Large address space complexity (2^n).



Memory

register

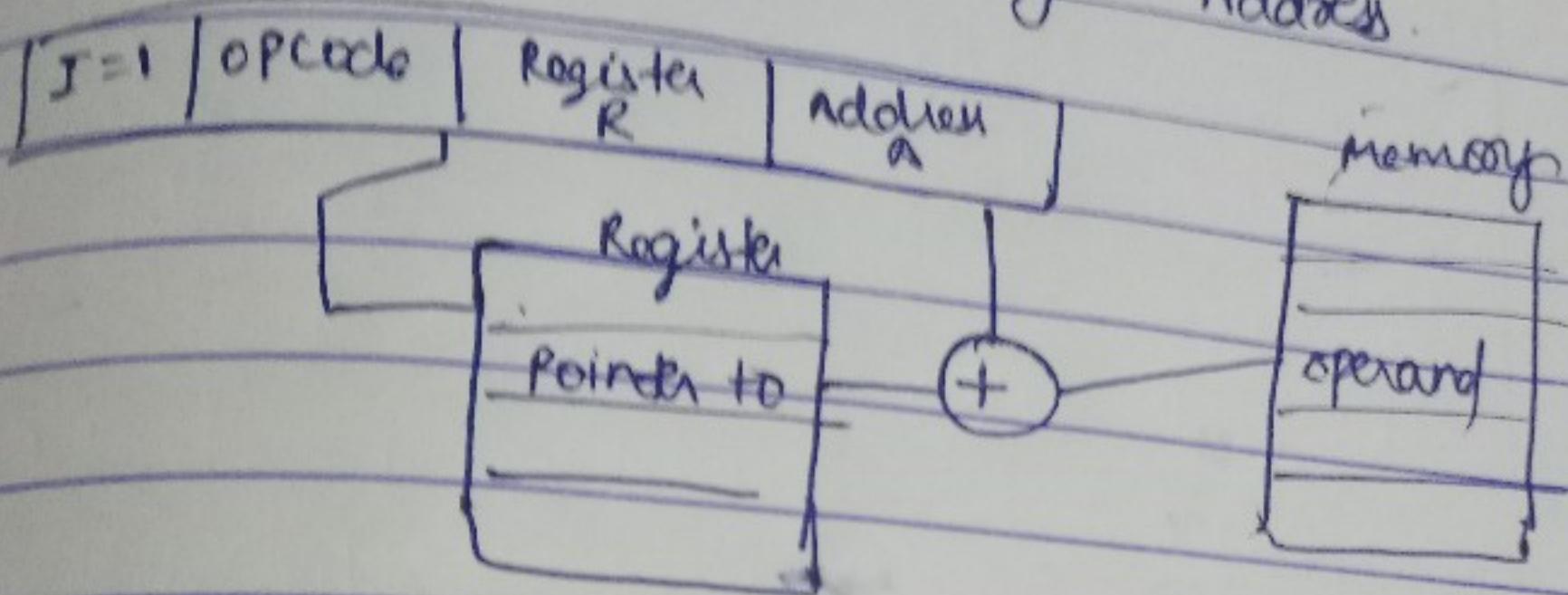
Pointers

operand

Date: / /

Displacement Address

EA (Effective Address) = Reg + Address.



Relative addressing

A version of displacement addressing

Base Register Address

$$EA = A + (R)$$

(displacement)

eg: Segment Reg. 80x86

Index Address

$$EA = A + R$$

$R++$

- A instruction stored in location 300 with addressing field 301 having a value of 600 stored.

- A register R having a value of 200. Find a
 - i) direct address
 - ii) index address
 - iii) immediate address
 - iv) relative address
 - v) register indirect address.

i) $EA = 301$, value = 600
 $\Rightarrow R++$

ii) $EA = A + R$
 $301 + 200 \Rightarrow 501$

$R++ = 200 + 1 = 201$

R	Pointer	300
200	600	301
		302

iii) $EA = 302$

iv) $EA = PC + A$
 $= 302 + 301$
 $= 603$