

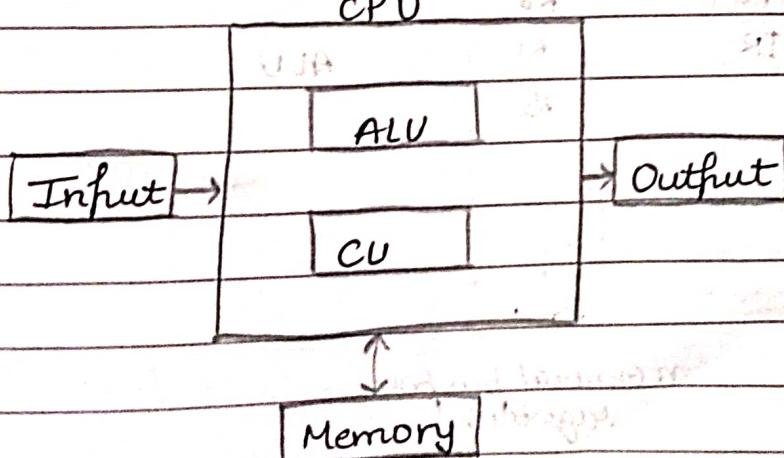
### Computer Architecture

- 1) It is concerned with the way hardware components are connected together to form a computer system.
- 2) It act as the interface b/w hardware and software.
- 3) It help us to understand the functionality of a system.
- 4) A programmer can view architecture in terms of instruction, addressing modes and registers.
- 5) While designing a computer system architecture is considered first.
- 6) Architecture involves datatypes and cache optimization.

### Computer organization

- It is concerned with the structure and behaviour of a computer system as seen by the user.
- It deals with the component of a connection in a system.
- It tell us how exactly all the units in system are arranged & interconnected.
- Organization expresses the realization of architecture.
- Organization is done on the basis of architecture.
- Organization involves circuit design addx to signals and peripherals.

### Simplified BD of Computer System



## Interfacing with Primary Memory

MAR:	1 Byte = 8 bits	10 add. line	address
(Mem. address register)	$1024 = 2^{10}$	= 1 KB	0
MDR:	10 add. line	$2^{10}$ add. space	1
(Mem. data register)	(mem. location)		2
			3
			4
			5
			6
			7
			8
			9
			10
			11
			12
			13
			14
			15
			16
			17
			18
			19
			20
			21
			22
			23
			24
			25
			26
			27
			28
			29
			30
			31
			32
			33
			34
			35
			36
			37
			38
			39
			40
			41
			42
			43
			44
			45
			46
			47
			48
			49
			50
			51
			52
			53
			54
			55
			56
			57
			58
			59
			60
			61
			62
			63
			64
			65
			66
			67
			68
			69
			70
			71
			72
			73
			74
			75
			76
			77
			78
			79
			80
			81
			82
			83
			84
			85
			86
			87
			88
			89
			90
			91
			92
			93
			94
			95
			96
			97
			98
			99
			100

- To read data from memory

Step 1: Load the memory address into MAR

Step 2: Issue the control signal "read"; "RD"

Step 3: Assume the data read from memory is stored into MDR

Step 4:

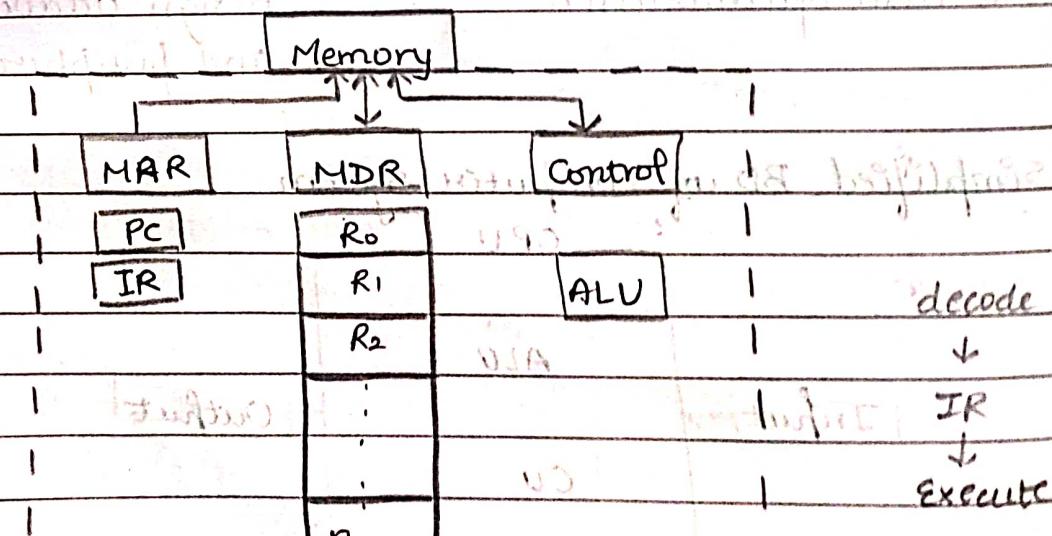
- To write data into memory

Step 1: Load memory address into MAR

Step 2: Load data to be written into MDR

Step 3: Issue the control signal "write"; "WR".

## Architecture of Processor



n General purpose  
registers.

MIPS  $\xrightarrow{RISC}$   
32 bits

$R_1 \leftarrow R_1 + LOCA$

Page No. \_\_\_\_\_  
Date: \_\_\_\_\_

Ex: ADD R1, LOCA

$MAR \leftarrow PC \text{ (addr.)}$

1.  $PC = 1000$

(mem. location)

$MDR \leftarrow \text{Mem}[MAR]$

$MAR = 1000$

$IR \leftarrow MDR \text{ (decod.)}$

$PC = PC + 4 = 1004$

$PC \leftarrow PC + 4$

$MDR = ADD R1, LOCA$

$JR = ADD R1, LOCA$

$MAR = LOCA = 5000 \text{ (add.)}$

$MDR = 75 \text{ (data)}$

$R1 = R1 + MDR = 50 + 75 = 125$

$R1 \boxed{125}$

Ins.  
mem.  
loc.

Address	Content
1000	ADD R1, LOCA
1004	-
5000	75

LOCA

PC : Program counter

PC : It stores address of next instruction which is to be executed

JR : Instruction Register

IR : It stores instruction

which is currently executed

Ex:

$R1 \boxed{250}$

$PC = 1500$

$R2 \boxed{200}$

$MAR = 1500$

$PC = PC + 4 = 1504$

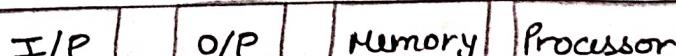
address	Instruction
1500	ADD R1, R2
1504	.....

$MDR = ADD R1, R2$

$IR = ADD R1, R2$

$R1 = R1 + R2 = 450$   
(destination) (source)

System Level single bus architecture

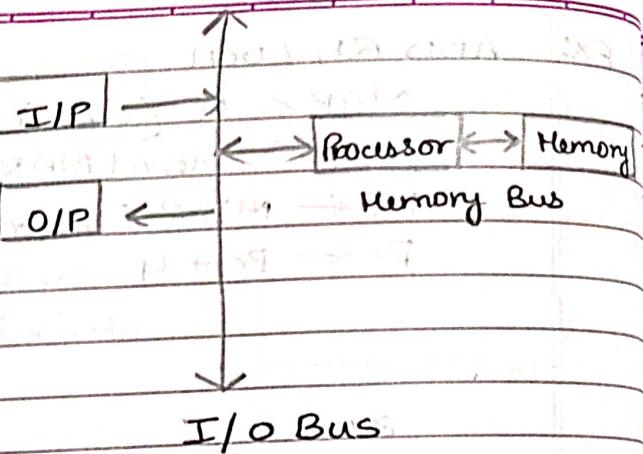


• Delay

• Path for all

### Double Bus

- Need more space
  - cost is more
  - Memory is not connected to I/O so need to introduce more paths
- (Good faster)*



- Software (Collection of programs to perform particular task)
  - Application Software: Made by users

OS: (Windows, Android, iOS)

Compiler & Assembler:

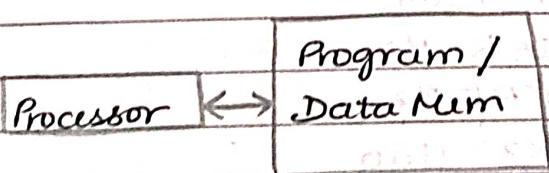
high level language to machine level language

Linker

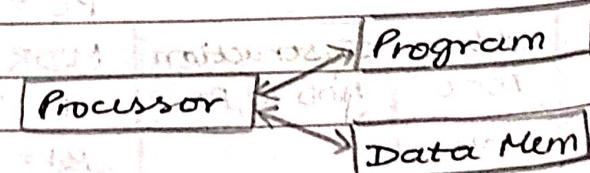
Editor & Debugger

- Types of Architecture

#### Von - Neumann



#### Harvard



Adv: Ease of access

Cost reduce

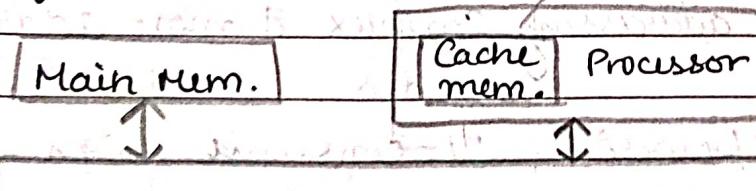
Complexity reduce

Dis: Time increase

Speed decrease

- |   |  |
|---|--|
| 1) It is first computer architecture based on stored program computer concept (Parallel-processing) | It is modern computer architecture based on harvard model.         |
| 2) Same physical memory address is used for instruction and data.                                   | Separate physical memory address is used for instruction and data. |
| 3) There is a common bus for data and instruction transfer for transferring data and instruction.   | Separate buses are used for data and instruction transfer.         |
| 4) Minimum two clock cycles are required to execute single instruction.                             | An instruction is executed in a single cycle.                      |
| 5) It is cheaper in cost.   | It is costly than Von-Neumann.                                     |
| 6) CPU cannot access instruction and read/write at the same time.                                   | It can access both at the same time.                               |
| 7) It is used in personal & small computers.  | It is used in micro controllers & signal processing.               |

### • Performance

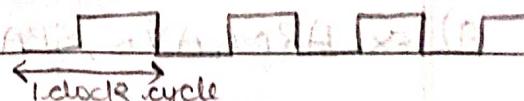


### • Processor clock clock cycle

clock Rate  $R = 1$  cycles/sec or Hz

$$\text{Ex: } 500 \text{ Millions cycles/sec} = 500 \text{ MHz}$$

$$1250 \text{ " " " } = 1.25 \text{ GHz}$$



## • Basic Performance Eqn:

$$T = \frac{N \times S}{R}$$

$T$  = processor time required to execute a program

$R$  = clock rate

$N$  = No. of machine language instructions

$S$  = Average no. of basic steps needed to execute one machine instruction where each step is completed in one clock cycle

## • Instruction set:

RISC	CISC
1) Reduced instruction set computer	Complex instruction set
2) Focus on software	Focus on hardware
3) Use only hard wired control unit	Use both hard wired and microprogram control unit
4) Fixed size instructions	Variable size instructions
5) <del>Code</del> size is large	Code size is small
6) Simple & limited addressing modes	Complex & more addressing modes
Ad 7) It consumes less power	It consumes high power
8) RISC required more RAM	CISC required less RAM
9) ex: ARM, AVR, SPARC etc	ex: Motorola family, AMD and intel - 86 processor

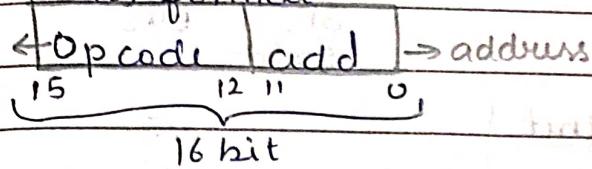
Program Mem is divided into two types  
Data Mem is divided into single part

Page No.

Date : 2/9/23

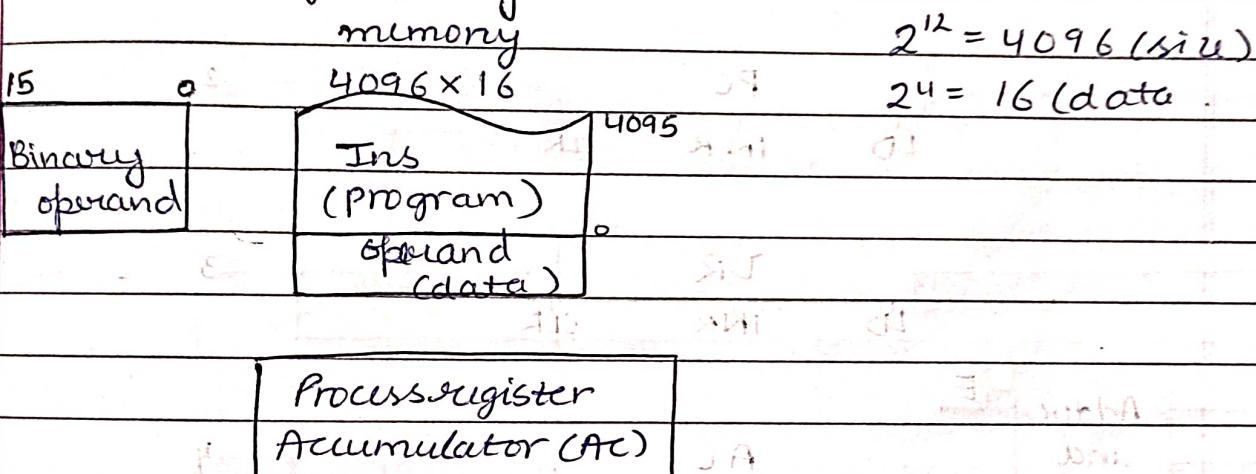
## Instruction Code (Ins code)

Ins format

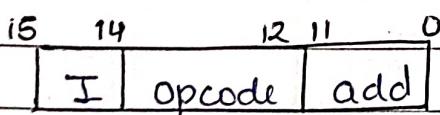


4-bits for op code =  $2^4 = 16$  operations stored  
12-bits for address

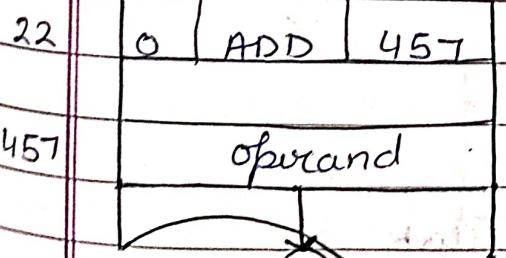
## Stored Program organization



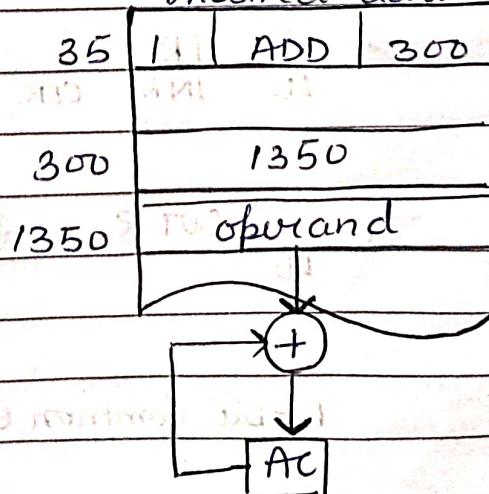
## Direct and indirect addressing



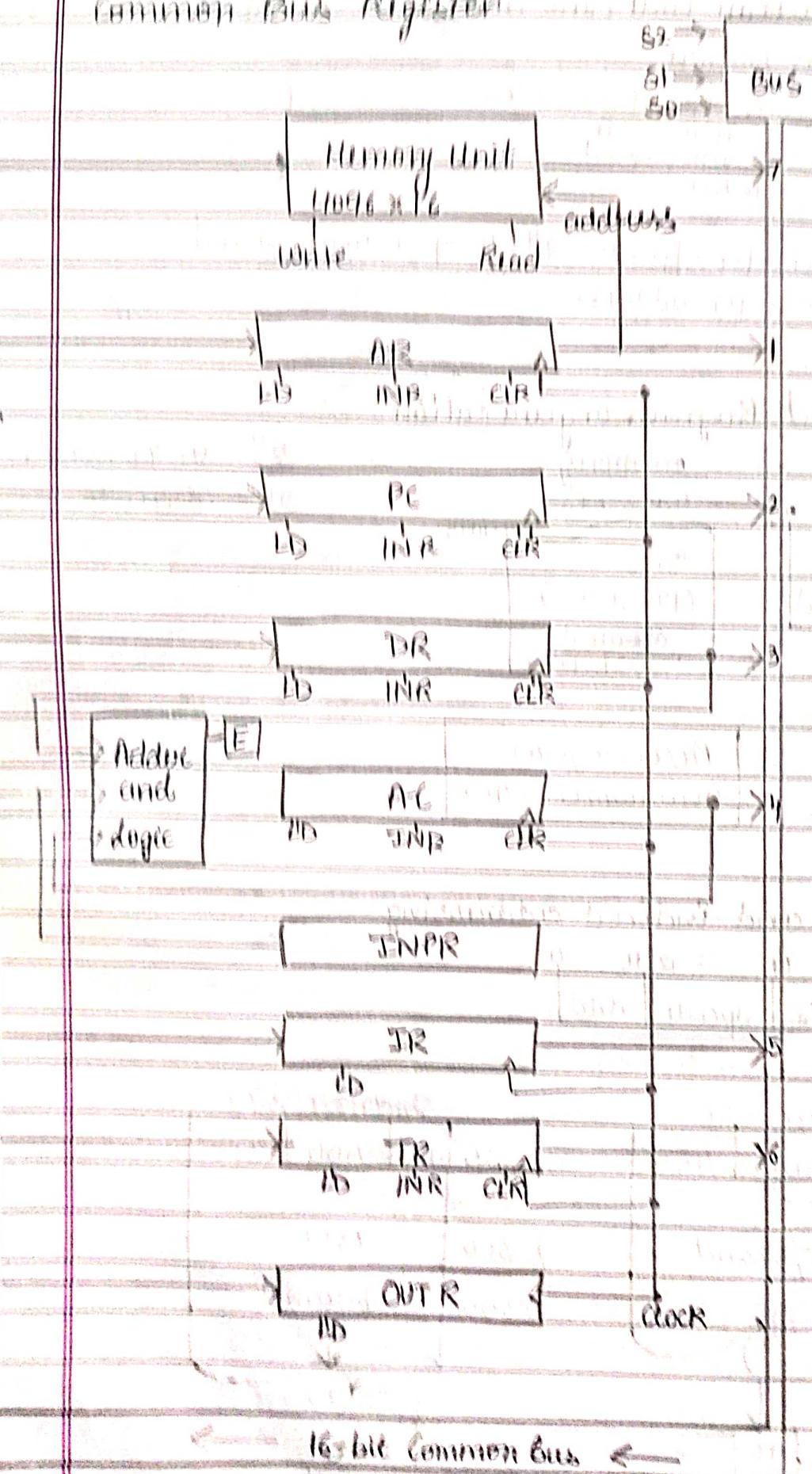
Direct add



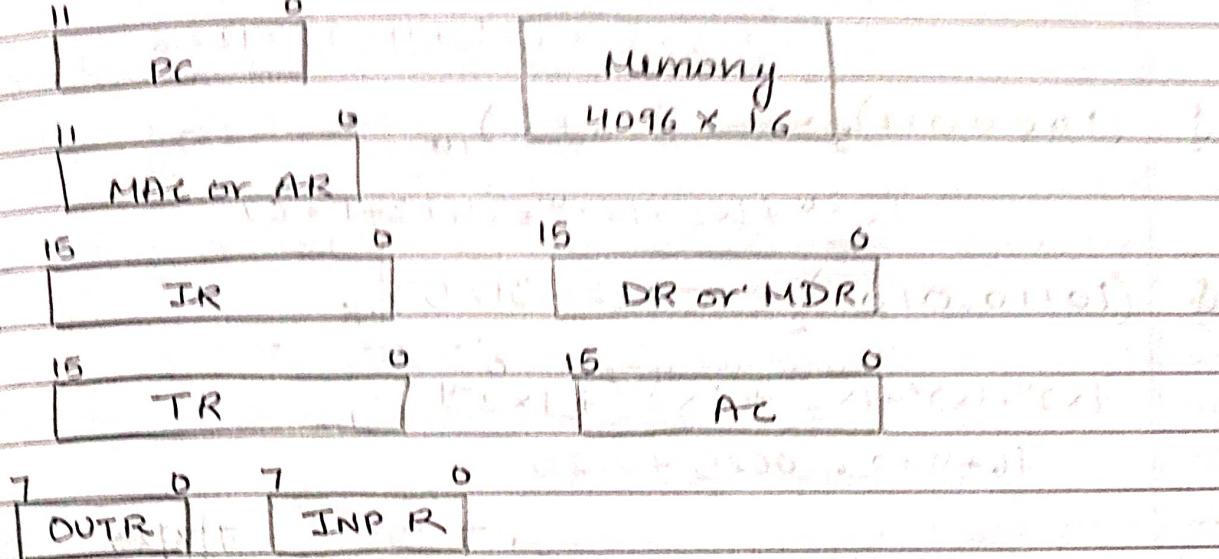
Indirect add



## Common Bus Register



## Computer Register



## Unit - 3

## Digital Number System

$$\begin{aligned}
 & \text{Q} \quad (234)_{10} \rightarrow (11101010)_2 \\
 & \quad \quad \quad \rightarrow (352)_8 \\
 & \quad \quad \quad \rightarrow (EA)_{16} \\
 & \quad \quad \quad \rightarrow (0010\ 0011\ 0100)_{BCD}
 \end{aligned}$$

$$\text{Q} \quad (292)_{10} = (1204)_b$$

Find the value of b.

$$(4 \times b^0) + (2 \times b^2) + (0 \times b^{10}) + (1 \times b^3)$$

$$4 + 2b^2 + b^3 = 292$$

$$b^3 + 2b^2 = 288$$

$$b^2(b+2) = 288$$

$b=6$ , which is not enough to form a basic

$$(BC)_{16} = (-188)_{10} \quad | 11 \times 16^1 + 12 \times 16^0$$

Q)  $(2314)_8 \rightarrow (1228)_{10}$

$$= 2 \times 8^3 + 3 \times 8^2 + 1 \times 8 + 4 \times 8^0 = 1228$$

Q)  $(1000011)_2 \rightarrow (67)_{10}$

$$1 \times 2^6 + 1 \times 2^1 + 1 \times 2^0 = 64 + 2 + 1 = 67$$

Q)  $(10110.0101)_2 \rightarrow (22.3125)_{10}$

$$1 \times 2^4 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^{-2} + 1 \times 2^{-4}$$

$$16 + 4 + 2 + 0.25 + 0.0625 = 22.3125$$

7/8/23

## Computer Instruction

	15	14	12-11	6	5-4, 11	I	I	0 Direct	i Indirect
I.	I	opcode	address						
Mem. ref. ins.									

II.	15	12-11	6	0
Reg. ref. ins	0111	register op <sup>n</sup>		

III.	15	12-11	6	0
I/O ins	1111	I/O operation		

## Timing & Control

Hardwired: (Digital & (MSB))  
Microprogrammed:

### Hardwired Control:

- The logic is implemented with gates, flip flops, decoders and other digital circuits.

Advantage: It is fast in response

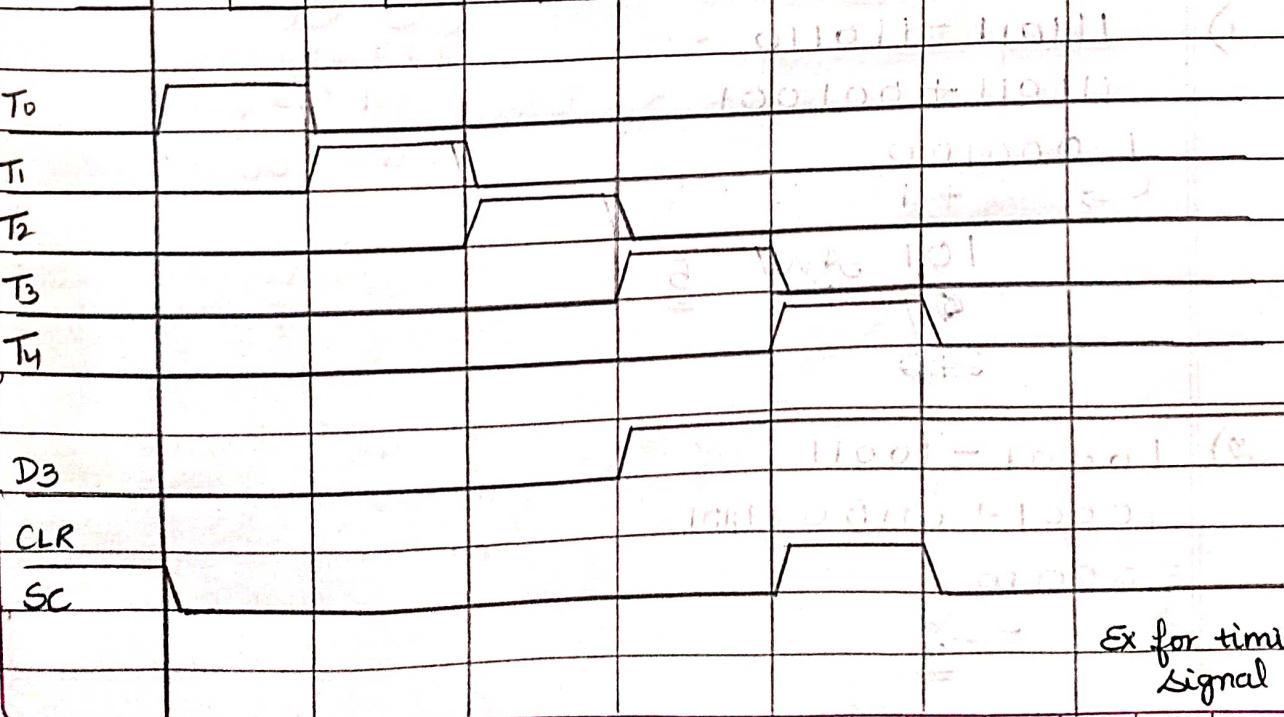
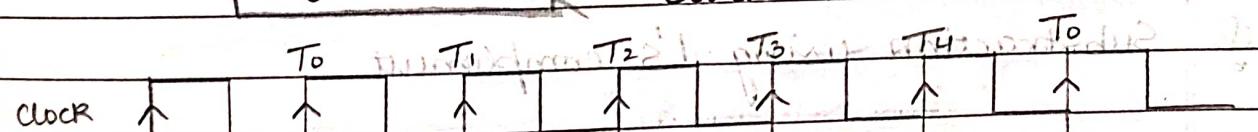
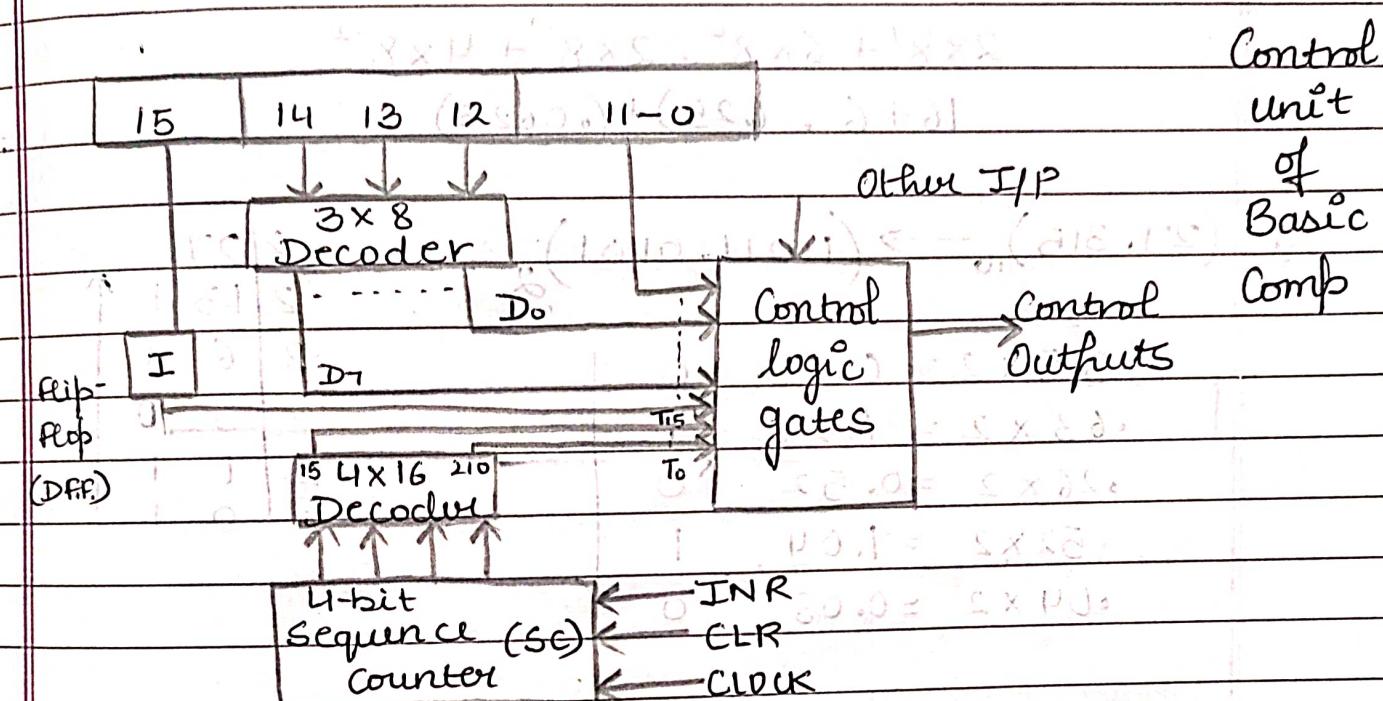
Disadvantage: If the design has to be modified or changed it requires changes in wiring among various components.

## Microprogrammed Control:

- The control information is stored in control memory, the control memory is programmed to initiate required sequence of micro-operations.

**Advantages:**

- Required changes or modifications can be done by updating microprogramme in control memory.



$D_3 T_4 : SC \leftarrow 0$

Q

$$(16.5)_{16} \rightarrow (22.3125)_{10}$$

$$1 \times 16^0 + 6 \times 16^{-1} \cdot 5 \times 16^{-2}$$

$$16 + 6 \cdot (25) + (0.0625)$$

Q

$$(26.24)_8 \rightarrow (22.3125)_{10}$$

$$2 \times 8^1 + 6 \times 8^0 \cdot 2 \times 8^{-1} + 4 \times 8^{-2}$$

$$16 + 6 \cdot (25) + (0.0625)$$

Q

$$(27.315)_{10} \rightarrow (11.011.0101)_2$$

$$0.315 \times 2 = 0.63 \quad 0$$

$$0.63 \times 2 = 1.26 \quad 1$$

$$0.26 \times 2 = 0.52 \quad 0$$

$$0.52 \times 2 = 1.04 \quad 1$$

$$0.04 \times 2 = 0.08 \quad 0$$

2	27
2	13
2	6
2	3
2	1
0	1

Q

Subtraction using 1's complement.

1)

$$111011 - 110110$$

$$111011 + 001001$$

$$\begin{array}{r} 1000100 \\ \swarrow +1 \\ \hline \end{array}$$

$$101 \text{ ANS. } 5$$

$$\begin{array}{r} 00\ 00 \\ 111011 \\ \hline \end{array}$$

$$\begin{array}{r} 00\ 00 \\ 001001 \\ \hline \end{array}$$

$$\begin{array}{r} 1000\ 100 \\ \hline \end{array}$$

$$\cancel{0}\cancel{1}\cancel{1}\cancel{0}$$

2)

$$10001 - 10011$$

$$10001 + 01100 = 11101$$

$$\begin{array}{r} 00010 \\ \swarrow - \\ \hline \end{array}$$

$$\begin{array}{r} 00010 \\ \swarrow - \\ \hline \end{array}$$

$$\begin{array}{r} 10001 \\ 01100 \\ \hline \end{array}$$

$$\begin{array}{r} 11101 \\ \hline \end{array}$$

1's comp

carry: carry forward +ve

no carry: 1's comp -ve

2's comp

carry: ignore +ve

no carry: 2's comp -ve

Page No.

Date:

||

### 1) Subtraction using 2's complement

$$1) \begin{array}{r} 111001 \\ - 1010 \\ \hline 111001 + 0110 \\ = +101111 \end{array}$$

= 47

$$\begin{array}{r} 111001 \\ - 0101 \\ \hline 111110 \end{array}$$

$$2) \begin{array}{r} 10011 - 10001 \\ = 10 \\ = +2 \end{array}$$

$$\begin{array}{r} 11111 \\ - 0110 \\ \hline 111111 \end{array}$$

$$\begin{array}{r} 111001 \\ + 110110 \\ \hline 1101111 \end{array}$$

$$\begin{array}{r} 1010 \\ + 01111 \\ \hline 100010 \end{array}$$

$$\begin{array}{r} 10111 \\ + 1010 \\ \hline 111001 \end{array}$$

$$\begin{array}{r} 111001 \\ - 10001 \\ \hline 111111 \end{array}$$

$$\begin{array}{r} 00010 \\ \hline 00010 \end{array}$$

$$\begin{array}{r} 000000 \\ + 1 \\ \hline 000001 \end{array}$$

$$1) 54321 - 41245 \quad 99999 \quad 1) 9's \text{ comp}$$

$$54321 + 58754 \quad 41245 \quad 2) \text{ add with A}$$

$$113075$$

$$58754$$

$$3) \text{ If carry } +1$$

$$\begin{array}{r} +1 \\ \hline 13076 \end{array}$$

$$4) \text{ No carry } 9's \text{ comp } (-\text{ve})$$

$$1231 - 4145$$

$$9999$$

$$9999$$

$$1231 + 5854$$

$$- 4145$$

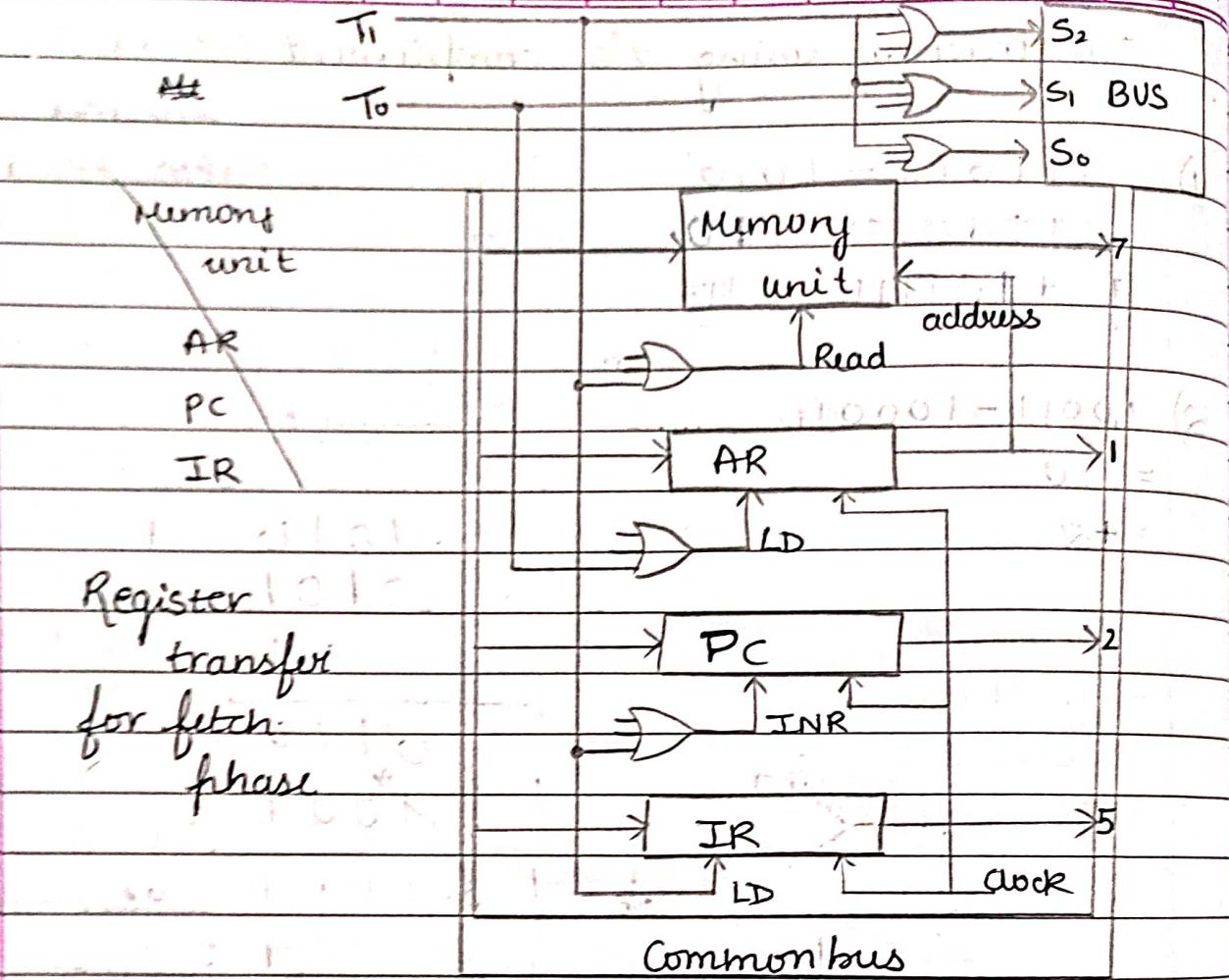
$$- 7085$$

$$7085$$

$$5854$$

$$2914$$

$$= -2914$$



### Fetch & Decode

T<sub>0</sub>: AR ← PC

T<sub>1</sub>: IR ← M[AR], PC ← PC + 1

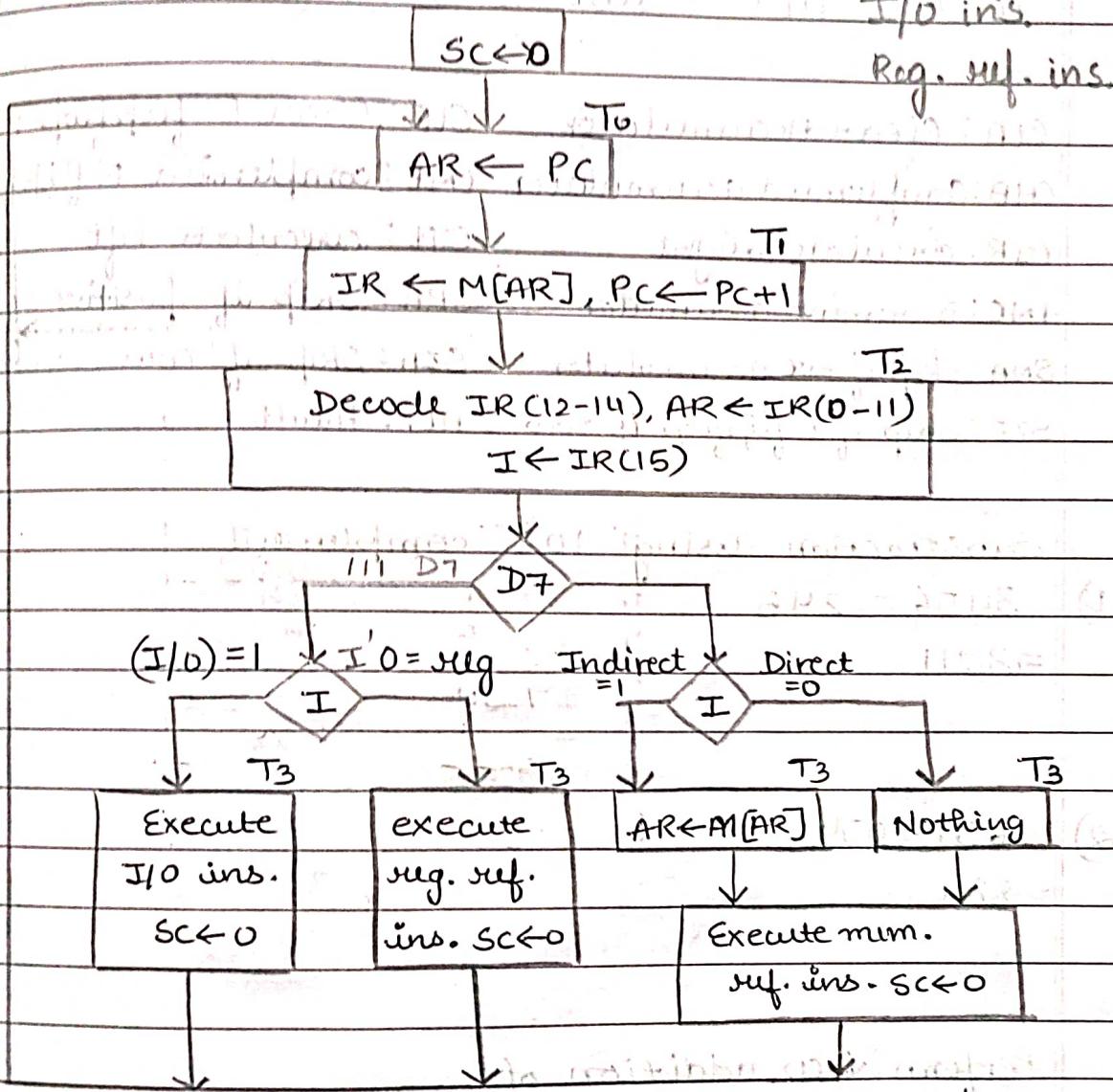
T<sub>2</sub>: Do... D<sub>7</sub> ← Decode IR (12-14)

AR ← IR (0-11), I ← IR (15)

### Instruction Cycle

1. Fetch 'm' instruction from the memory.
2. Decode the instruction.
3. Read the effective address from memory if the instruction has indirect address.
4. Execute the instruction.

Determine the type of instruction : Memory reference

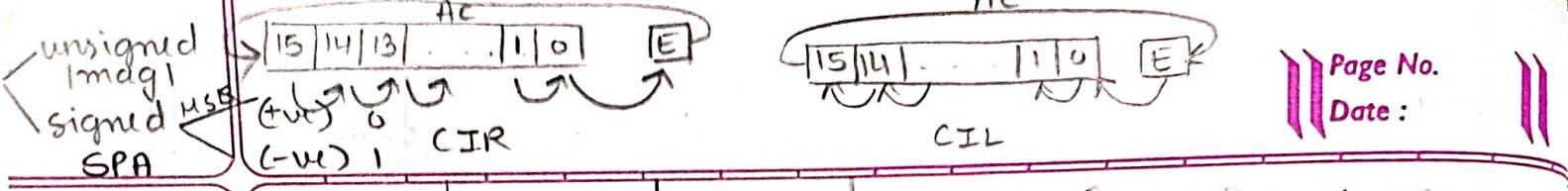


### Reg. Reference Instruction

				$M = D_7 I' T_3$
7800	CLA	MBii	$AC \leftarrow 0$	Initial $I = 1$
7400	CLE	MB10	$E \leftarrow 0$	0 clock cycle
7200	CMA	MB9	$AC \leftarrow \bar{AC}$	
7100	CME	MB8	$E \leftarrow \bar{E}$	
7080	CIR	MB1	$AC \leftarrow \text{shl}(AC), AC(15) \leftarrow E, E \leftarrow AC(0)$	
7040	CIL	MB6	$AC \leftarrow \text{shl}(AC), AC(0) \leftarrow E, E \leftarrow AC(15)$	
7020	INC	MB5	$AC \leftarrow AC + 1$	
7010	SPA	MB4	$If (AC(15=0)) \text{ then } PC \leftarrow PC + 1$	
7008	SNA	MB3	$If (AC(15=1)) \text{ then } PC \leftarrow PC + 1$	
7004	SZA	MB2	$If (AC=0) \text{ then } PC \leftarrow PC + 1$	

Reg. ref.

Processes



7002	SZE	$\mu$ B.	If $(E=0)$ then $PC \leftarrow PC + 1$
7001	HLT	$\mu$ B.	$S \leftarrow 0$ (S is a start stop FF)

CLA: Clear Accumulator

CLE: Clear E-flip flop

CMA: Complement accumulator

CME: complement E-flip flop

CIR: circulate right

CIL: circulate left

INC: increment accumulator

SPA: skip if positive accumulator  
(all bits: 0)

SNA: skip if -ve accumulator

SZA: skip if zero accumulator

SZE: skip if E-flip flop is zero

HLT: Hault

Subtraction using 10's complement

$$1) \quad 3456 - 245 \quad \begin{array}{r} 9999 \\ 0245 \\ \hline 9754 \\ + 1 \\ \hline 9755 \end{array} \quad \begin{array}{r} 0001 \\ 3456 \\ 9755 \\ \hline 13211 \end{array}$$

$$2) \quad 1631 - 0745 \quad \begin{array}{r} 9999 \\ 0745 \\ \hline 9254 \\ + 1 \\ \hline 9255 \end{array} \quad \begin{array}{r} 0001 \\ 1631 \\ 0000 \\ \hline 10886 \end{array}$$

Perform BCD addition of

$$8765 + 3943$$

$$\begin{array}{r} 1000 \ 0111 \ 0110 \ 0101 \\ + 0011 \ 1001 \ 0100 \ 0011 \\ \hline 110000000 \ 1010 \ 1000 \\ 0110 + 0010 \ \underline{\hline 0110} \ 12708 \\ \hline 0010 \ 0111 \ 0000 \ 12708 \\ = 12708 \end{array}$$

x = don't care (0 or 1)

$\wedge$  and

$\vee, \bar{I}$  OR

7 → resolve for Reg.ref. AC: accumulator:

F → resolve for I/O ref. performs all arithmetic op

Page No.

Date: 16/8/23

## Memory Reference Instruction

Hexadecimal code

I = 0	I = 1	Symbol	Op <sup>n</sup>	
0XXX	8XXX	AND	D <sub>0</sub>	AC $\leftarrow$ AC $\wedge$ M[AR]
1XXX	9XXX	ADD	D <sub>1</sub>	AC $\leftarrow$ AC + M[AR], E $\leftarrow$ last
2XXX	AXXX	LDA	D <sub>2</sub>	AC $\leftarrow$ M[AR]
3XXX	BXXX	STA	D <sub>3</sub>	M[AR] $\leftarrow$ AC
4XXX	CXXX	BUN	D <sub>4</sub>	PC $\leftarrow$ AR
5XXX	DXXX	BSA	D <sub>5</sub>	M[AR] $\leftarrow$ PC, PC $\leftarrow$ AR + 1
6XXX	EXXX	JSZ	D <sub>6</sub>	M[AR] $\leftarrow$ M[AR] + 1, If M[AR] + 1 = 0, then PC $\leftarrow$ PC + 1

AND: It and memory content with accumulator

ADD: It adds to memory contents with accumulator.

E: Temp. flip flop associated with carry (final)

LDA: Load content of memory to accumulator

STA: Store content of accumulator into memory

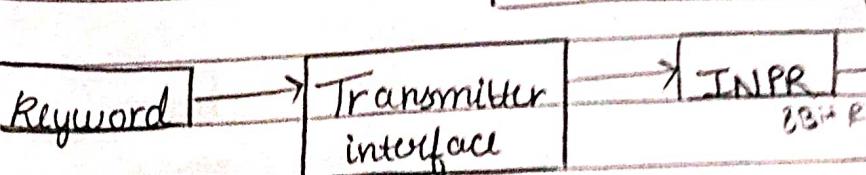
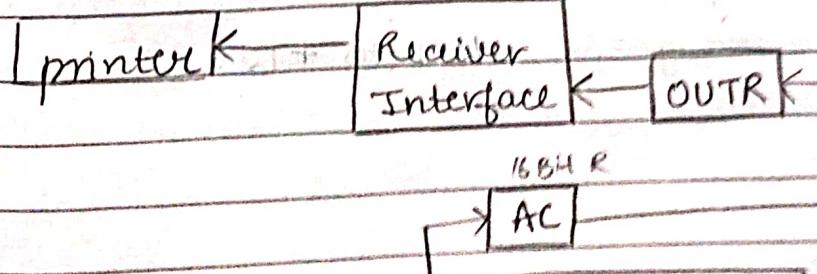
BUN: Branch Unconditionally (precursor term: subroutine)

BSA: Branch and save return address

JSZ: Increment and skip if zero

## I/O configuration

FGIO = 1, 0



FGTI = 1, 0

## Binary number Representation

(Draw back: 2 values for 0)

Unsigned

- (no significance to signs)

Unsigned

- (Range:  $0 \text{ to } 2^n - 1$ )

000 0

001 1

010 2

011 3

100 4

101 5

110 6

111 7

Range

$0 \text{ to } (2^n - 1)$

$0 \text{ to } (2^3 - 1)$

$= 0 \text{ to } 7$

Signed

Signed

-ve (bit 0)

+ve (bit 1)

sign-Magnitude 1's comp 2's comp

form

form

form

Signed

sign & mag 1's 2's

sign & mag 1's 2's

-7 -0 -1

+7 +7 +7

-6 -1 -2

+6 +6 +6

-5 -2 -3

+5 +5 +5

-4 -3 -4

+4 +4 +4

-3 -4 -5

+3 +3 +3

-2 -5 -6

+2 +2 +2

-1 -6 -7

+1 +1 +1

-0 -7 -8

+0 +0 +0

$[-2^{n-1} \text{ to } 2^{n-1}]$  2's comp.

$[-2^{n-1} \text{ to } 2^{n-1}]$  2's comp.

$[-(2^{n-1}) \text{ to } 2^{n-1}]$  1's comp

$[-(2^{n-1}) \text{ to } 2^{n-1}]$  sign (ignoring)

Q find range of 5 bit unsigned binary no., also find min & max value in this range

$$0 - 2^5 - 1 = 0 \text{ to } 31$$

$$\min = 0 \quad \max = 31$$

Q

Sign & Mag.

$$n = 8, \text{ Range} = ? \quad -127 \text{ to } +127$$

Sign & Mag  
add/sub

$$\text{Range} = -(2^n - 1) \text{ to } (2^n - 1)$$

8 cases

		$A \geq B$	$A < B$
$(+A) + (+B)$	$+ (A+B)$		
$(-A) + (+B)$		$-(A-B)$	$+(B-A)$
$(+A) + (-B)$		$+(A-B)$	$-(B-A)$
$(-A) + (-B)$	$-(A+B)$		

23/8/23 Input / Output Ins.

F800	INP	PB11	AC(0-7) $\leftarrow$ INPR, FG1T $\leftarrow$ 0
F400	OUT	PB10	OUTR $\leftarrow$ AC(0-7), FG1O $\leftarrow$ 0
F200	SKI	PB9	If (FG1T = 1) then PC $\leftarrow$ PC + 1
F100	SKO	PB08	If (FG1O = 1) then PC $\leftarrow$ PC + 1
F080	ION	PB07	IEN $\leftarrow$ 1
F040	IOF	PB06	IEN $\leftarrow$ 0

$$P = D_7 I I_3$$

$$B_i = IR(i) [IR(6-11)]$$

INP: Input character

OUT: output character

SKI: skip on Input flag

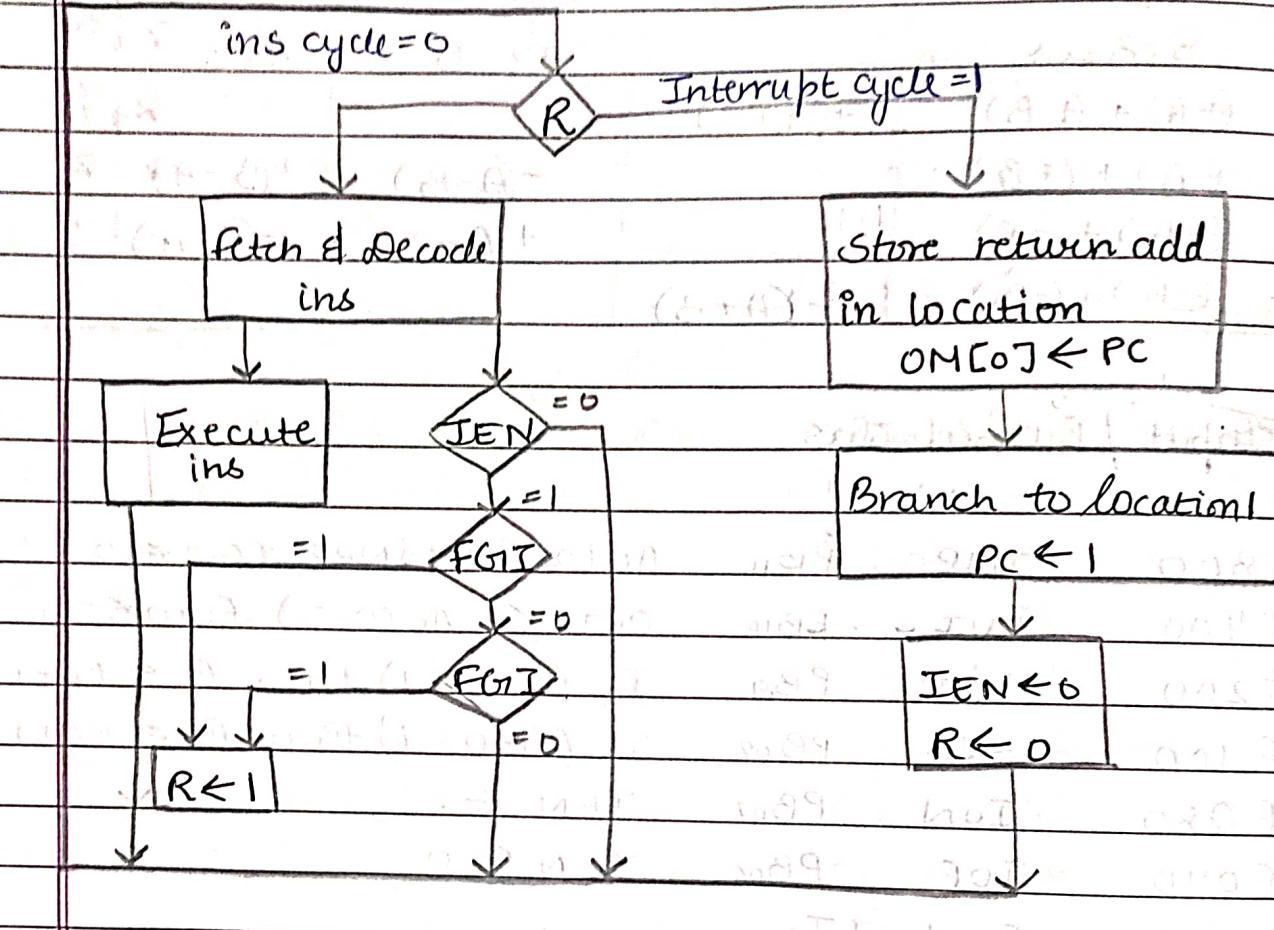
SKO: skip on output flag.

ION: Interrupt enable on

IOF: Interrupt enable off

IEN: Interrupt Enable

## Program Interrupt



Memory

Memory

0	
(PC=256) 1	0 BUN 1120
255 Main mem	
1120 I/O Program(Subroutine)	
1 BUN 0	

0	256 (Returnaddr)
PC=1	0 BUN 1120
255	Main
256	Program
1120	I/O
	Program
	BUN 0

Before int

After int cycle

Unit-3

Q +36 as a 8 bit binary notation

$$+36 = 00100100$$

$$36 = \begin{array}{r} 00 \\ 11011011 \\ +1 \end{array}$$

$$2^5 = \begin{array}{r} 11011100 \\ -128 \\ 64 \\ 32 \\ 16 \\ 8 \\ 4 \\ 2 \\ 1 \end{array}$$

$$-128 \quad 64 \quad 32 \quad 16 \quad 8 \quad 4 \quad 2 \quad 1$$

Decimal  $\Rightarrow -36$

$$2^S = \begin{array}{r} 00 \\ 00100011 \\ +1 \end{array}$$

$$00100100$$

Decimal  $\Rightarrow +36$

$A + (2^S \text{ comp of } A) = 0$  } If carry discarded it

Q Using 8-bit 2's complement

$$(-8) + (+3)$$

$$-8 = 11111000$$

$$+3 = 00000011$$

$$= 11111000$$

$$00000011$$

$$11111011$$

$$-128$$

Decimal  $= -5$

$$(+8) = 00001000$$

$$\begin{array}{r} 00001000 \\ +1 \\ \hline 11111011 \end{array}$$

$$11111000$$

$$00000100$$

$$+1$$

$$00000101$$

$$(-5)$$

$$\stackrel{Q}{=} (-5) + (+3)$$

$$(-5) + (+3)$$

$\begin{array}{r} 11111011 \\ + 11111110 \\ \hline \end{array}$

$$(-5) + (+3)$$

$\begin{array}{r} 11111101 \\ - 1 = -1 \\ \hline \end{array}$

$$-5 = 11111010$$

$$+ 3 = 11111110$$

$\begin{array}{r} 11111110 \\ - 1 = 11111101 \\ \hline \end{array}$

$$-2 = 11111010$$

$$-2 = 000000010$$

$$-2 = 000000010$$

$$-2 = 11111010$$

$$\stackrel{Q}{=} (+7) + (-4)$$

$\downarrow$

$$00000101$$

$$00000100$$

$$00001011$$

$$+ 11$$

$$-7 = 00000111$$

$$00000100$$

$\begin{array}{r} 00000100 \\ - 7 = 11111001 \\ \hline \end{array}$

$$-2 = 11111001$$

$$-2 = 00000100$$

$$00000100$$

$\begin{array}{r} 00000100 \\ - 2 = 11111000 \\ \hline \end{array}$

$$\stackrel{Q}{=} (-3) + (+4)$$

$\downarrow$

$$11111001$$

$$11111100$$

$$(-3) + (+4)$$

$$+ 11$$

$$-3 = 11111000$$

$$00000111$$

$\begin{array}{r} 00000111 \\ - 3 = 11111001 \\ \hline \end{array}$

$$-2 = 11111001$$

$$-2 = 00000100$$

$$00000100$$

$\begin{array}{r} 00000100 \\ - 2 = 11111000 \\ \hline \end{array}$

22/3/23

### Carry vs Overflow

- i) Overflow will for addition

$$(+4) + (+3) = (+7)$$

$$(-6) + (-5) = (-11)$$

$$\text{Ex: } (-7) + 60$$

$$1001 - 1010$$

$$= 110011$$

$$= 10111$$

$$(+42) = (+3), \text{ overflow}$$

$$-7 \rightarrow 0111$$

$$+ 7 \rightarrow 1000$$

$$-5 \rightarrow 0110$$

$$+ 5 \rightarrow 1010$$

Second step result

$(-7) + 60 = (-7) + 42 = (-7) + 3 = (-4)$

Ex:  $(+7) + (+6)$

$$\begin{array}{r}
 \textcircled{1} \textcircled{1} \\
 \textcircled{0} \textcircled{1} \textcircled{1} \\
 \hline
 0110 \\
 1101 \\
 \hline
 -8 \ 4 \ 1 = (-3) \ \underline{\text{overflow}}
 \end{array}$$

### 2) Overflow rule for subtraction

If two numbers in their 2's complement form are subtracted and their sign are different then overflow occurs iff the result has the same sign as subtrahand.

- $(+A) - (-B) = (+ve)$
- $(-A) - (+B) = (-ve)$

Ex:  $-(-6) - (+7) - (-6)$

$$\begin{array}{r}
 +6 = 0110 \\
 -6 = 1010 \\
 \hline
 \end{array}$$

$$\begin{array}{r}
 (011) + (-1010) \\
 (011) + (0110) \\
 \hline
 = 1101 \\
 -8 \ 4 \ 1 = (-3) \ \underline{\text{overflow}}
 \end{array}$$

### 3) Multiplication and division for unsigned numbers.

$$(101)_{10} \times (61)_{10}$$

$$\begin{array}{r}
 2 | 61 \\
 2 | 30 \ 1
 \end{array}$$

$$(1100101)_2 \times (111101)_2$$

$$\begin{array}{r}
 2 | 15 \ 0 \\
 2 | 7 \ 1
 \end{array}$$

$$\begin{array}{r}
 2 | 3 \ 1 \\
 2 | 1 \ 1
 \end{array}$$

$$\begin{array}{r}
 2 | 2 \ 1 \ 1 \\
 0 \ 1 \ 2
 \end{array}$$

0 1

	1	1	0	0	1	0	1
	X	1	1	1	1	0	1
①	②	③	④	⑤	⑥	⑦	⑧
①	1	1	0	0	0	0	1
①	0	0	0	0	0	0	X
①	1	1	0	0	1	0	1
①	1	1	0	0	1	0	X
①	1	1	0	0	1	0	X
①	1	1	0	1	0	1	X
1	1	0	0	1	0	1	X
1	0	0	1	0	1	X	X
1	0	0	0	0	1	0	0

$$\text{Lokalwert} = \left( \begin{smallmatrix} 1000000 & 10001 \end{smallmatrix} \right)_2$$

$$\text{Ex: } \frac{(37)}{18} \div \frac{(5)}{10}$$

$$\underline{(100101) \div (101)}$$

$$\begin{array}{r}
 \boxed{101} & 100101 & \boxed{111} \\
 - 101 & \downarrow & | \\
 \hline
 & 1000 & \\
 & 101 & \downarrow
 \end{array}$$

$$\begin{array}{r} 00111 \\ \times 101 \\ \hline 010 \end{array}$$

## Signed Multiplication.

$$\text{Ex: } (+7) \times (-6)$$

$$\begin{array}{r} \underline{0\ 0\ 0\ 0\ 1\ 1\ 1} \\ \times 0\ 0\ 0\ 0\ 1\ 1\ 0 \\ \hline \end{array}$$

0 0 0 0 0 0 0

00000111XX

$$00101010 = (11)_2$$

$$5 = 101$$

$$6 = 110$$

Page No.  
Date:

Ex

$$(-7) \times (-6)$$

$$\begin{array}{r} 11111001 \\ \times 11111010 \\ \hline \textcircled{10} 00000000 \end{array}$$

$$\begin{array}{r} 11111001 \times^3 \\ 00000000 \end{array}$$

$$\begin{array}{r} 1111001 \times \times \times \\ 111001 \times \times \times \times \end{array}$$

$$\begin{array}{r} 1001 \times \times \times \times \times \\ 001 \times \times \times \times \times \end{array}$$

$$\begin{array}{r} 0010101010 \\ \textcircled{10} 0010101010 \end{array}$$

$$\begin{array}{r} 11111010 \\ \times 11111010 \\ \hline \textcircled{10} 00000000 \end{array}$$

$$(11111010) \times (11111010) = (11111010)$$

$$Ex: (+7) \times (-6)$$

$$\begin{array}{r} 0000(0101) \\ \times 111110 \\ \hline \textcircled{10} 00000000 \end{array}$$

$$\begin{array}{r} 0000111X \\ 0001111X \end{array}$$

$$\begin{array}{r} 0001111X \\ 0110101010 \end{array}$$

$$\begin{array}{r} 0110101010 \\ 1111101010 \end{array}$$

$$\begin{array}{r} 1111101010 \\ 1111101010 \end{array}$$

$$\begin{array}{r} 1111101010 \\ 1111101010 \end{array}$$

$$\begin{array}{r} 1111101010 \\ \textcircled{10} 1111101010 \end{array}$$

$$-6 = 110$$

$$\begin{array}{r} 0010101010 \\ = (0010101010) \end{array}$$

$$= (42)$$

## Floating point Representation.

- 1) Mantissa (M)
- 2) Base (b)
- 3) Exponent (E)

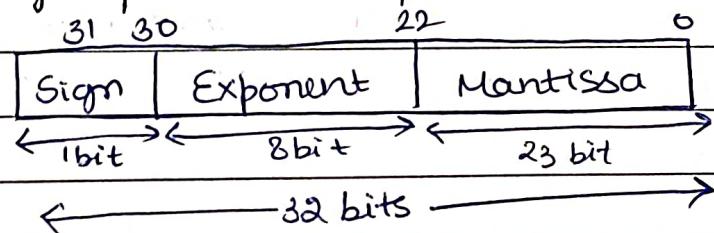
$$\pm M \times b^E$$

In floating point representation binary point float to the right of most significant one and an exponent is used.

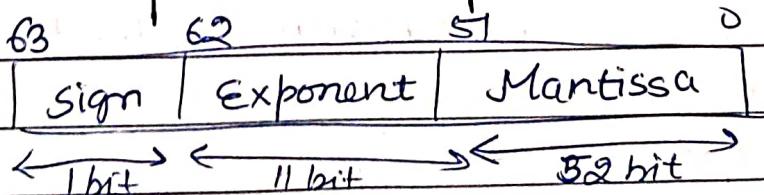
for ex:	M	b	E
$9 \times 10^8$	9	10	8
$110 \times 2^7$	110	2	7
4995.653	4295653	10	-3

## IEEE 754 floating pt. Representation.

### 1) Single precision Representation (32 bit)



### 2) Double precision Representation (64-bit)



Q) Represent 1259.125 in single and double precision floating point representation.

$$(1259.125)_{10}$$

$$\begin{array}{r} 1259 \\ \times 2 = .950 \\ 0 \\ 0.950 \times 2 = .5 \\ 0 \\ \vdots \end{array}$$

1) Step 1: convert into binary

$$(100111010111.001)$$

2) Step 2: Normalize no.

for single precision:

$$(1.N)_2^{E-127}$$

$$1.\underline{00111\ 01011001} \times 2^{10}$$

(mantissa)

$$E - 127 = 10$$

$$E = 127$$

Step 3:

$$2 | 137$$

$$2 | 68 \ 1$$

$$2 | 34 \ 0$$

$$2 | 17 \ 0$$

$$2 | 8 \ 1$$

$$2 | 4 \ 0$$

$$2 | 2 \ 0$$

$$2 | 1 \ 0$$

$$0 \ 1$$

	1	8	23
sign	0	10001001	0011101011001000...

Single  $\rightarrow$

$$(10001001)_2$$

	1	11	52
sign	0	10000001001	0011101011001000...

Double  $\rightarrow$

2) Step 2: For Double precision:

$$(1.N)_2^{E-1023}$$

$$E - 1022 = 10$$

$$E = 1033$$

$$(10000001001)_2$$

$$2 | 1033$$

$$2 | 516 \ 1$$

$$2 | 258 \ 0$$

$$2 | 129 \ 0$$

$$2 | 64 \ 1$$

$$2 | 32 \ 0$$

$$2 | 16 \ 0$$

$$2 | 8 \ 0$$

$$2 | 4 \ 0$$

$$2 | 2 \ 0$$

Q

0110100000000011

Convert following floating point binary no. into decimal.

i)

Assume 9-bit Mantissa, 6-bit exponent

0	110100000	000011
sign	Mantissa	Exponent

$$000011 \rightarrow +3$$

$$110.100000 \times 2^3 \rightarrow \text{binary}$$

2<sup>2</sup>

$$+2^1 + 2^0 = 2^1$$

$$= (5.5)_{10}$$

$\rightarrow$  decimal

ii)

0101010000000.010

0|101010000|000010

$$000010 \rightarrow +2$$

$$10.1010000 \times 2^2 \rightarrow \text{binary}$$

$$2^1 \cdot 2^{-1} + 2^{-3}$$

$$(2.625)_{10}$$

$\rightarrow$  decimal

iii)

Assume 10-bit Mantissa, 6-bit exponent both in 2's complement.

01101100000000100

+ve so no 2's comp  
first bit is 0...

0|1101100000|000100

$$000100 \rightarrow +4$$

$$1101.100000 \times 2^4 \rightarrow \text{binary}$$

$$2^3 + 2^2 + 2^0 \cdot 2^{-1}$$

$$(13.5)_{10} \rightarrow \text{decimal}$$

Q How to represent a no. in IEEE 754 32-bit floating point number?

263.3

$$3 \times 2^1 = .6 \quad 6 \quad 2 \quad | \quad 263.3$$

Step 1:  $(100000111.010011001)$   
so on...

$$\begin{array}{r} -6 \times 2 = 1.2 \quad 1 \quad 2 \\ 1.2 \times 2 = .4 \quad 0 \quad 2 \end{array} \quad | \quad 131 \quad 1$$

Step 2:  $1.0000011101001 \times 2^8$

$$\epsilon - 127 = 8$$

$$\epsilon = 127 + 8$$

$$\begin{array}{r} .4 \times 2 = .8 \quad 0 \quad 2 \\ .8 \times 2 = 1.6 \quad 1 \quad 2 \end{array} \quad | \quad 32 \quad 1$$

$$\begin{array}{r} .8 \times 2 = 1.6 \quad 1 \quad 2 \\ .6 \times 2 = 1.2 \quad 1 \quad 2 \end{array} \quad | \quad 16 \quad 0$$

$$\begin{array}{r} .2 \times 2 = .4 \quad 0 \quad 2 \\ .4 \times 2 = .8 \quad 0 \quad 2 \end{array} \quad | \quad 4 \quad 0$$

Step 3:  $135$  here we have to find 8 bits for mantissa  
 $135 \div 2 = 67$

$$\begin{array}{r} 2 \mid 67 \quad 1 \\ 2 \mid 33 \quad 1 \\ 2 \mid 16 \quad 1 \\ 2 \mid 8 \quad 0 \end{array} \quad | \quad 2 \quad 1 \quad 0$$

$$\begin{array}{r} 2 \mid 4 \quad 0 \\ 2 \mid 2 \quad 0 \end{array} \quad | \quad 0 \quad 1$$

$$\begin{array}{r} 2 \mid 1 \quad 0 \\ 2 \mid 0 \quad 1 \end{array} \quad | \quad 0 \quad 1$$

$$\begin{array}{r} 2 \mid 0 \quad 1 \\ 2 \mid 0 \quad 0 \end{array} \quad | \quad 0 \quad 1$$

$$\begin{array}{r} 2 \mid 0 \quad 0 \\ 2 \mid 0 \quad 0 \end{array} \quad | \quad 0 \quad 1$$

$$\begin{array}{r} 2 \mid 0 \quad 0 \\ 2 \mid 0 \quad 0 \end{array} \quad | \quad 0 \quad 1$$

$$\begin{array}{r} 2 \mid 0 \quad 0 \\ 2 \mid 0 \quad 0 \end{array} \quad | \quad 0 \quad 1$$

$$\begin{array}{r} 2 \mid 0 \quad 0 \\ 2 \mid 0 \quad 0 \end{array} \quad | \quad 0 \quad 1$$

fixed point representation

In fixed point representation there is a fixed no. of digits after decimal point.

→ Unsigned

→ Signed

→ sign & mag

→ 2's comp

Q) Represent fixed point representation of unsigned binary number 0110 110 using 4 int bit and 3 fractional bit.

$$\rightarrow 0110.110$$

$$\rightarrow 2^3 + 2^2 + 2^1 + 2^0$$

$$\rightarrow (6.75)_{10}$$

Q) Represent (-7.5) using 8 bit binary Q's comp representation with 4 digit int and 4 fractional bit

$$-7.5 ; +7.5 = 0111.1000$$

$$\begin{array}{r} 0000 \\ 1000.0111 \\ \hline \end{array}$$

+1

$$-7.5 = 1000.1000$$

$$.75 \times 2 = 1.0$$

Q) Compute  $0.75 + (0.625)$  using fixed point representation.

$$.75 \times 2 = 1.5 \quad | \quad + 0.625 \times 2 = 1.25 = 1$$

$$.5 \times 2 = 1.0 \quad | \quad .25 \times 2 = .5 = 0$$

$$0.75 + 0.625 \times 2 = 1.375 = 1$$

$$0000.1100 + 0000.0110 = (-0.625) 0101_2$$

$$0000.0010$$

$$\begin{array}{r} 0101 \\ +1 \\ \hline 10110 \end{array}$$

Q) Assume no. using 32 bit format which reserves 1 bit for sign, 15 bit for int, 16 bit for fractional.

$$-43.625$$

111101011.

2103

(100000000101011, 1010000000000)<sub>2</sub>

2101

2150

2121

2110

101

$$03 = (101011)$$

$$625 = 101$$

- Q) The following no. are stored using 8's comp. in a 12-bit register with 4-bits after binary point.  
 Convert them into decimal numbers

01111111.1111

$$-2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0, 2^7 = 2^2 + 2^3 + 2^0$$

$$64 - 32 - 16 - 8 - 4 - 2 - 1 - 0.5 - 0.25 - 0.125 = 0.2625$$

 $(127.9375)_{10}$ 

Q) 11111111.1111  
 $2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 2^7 + 2^2 + 2^3 + 2^4$

 $(-0.0625)_{10}$ 

- Q) Using 2's complement convert following decimal number into fixed point binary number to be stored in K-bit register with 4-bits after binary point.

-45.75

2105

2100

2100

2100

2100

000000101101.1100

2100

2100

2100

2100

00

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1

2100

+1