Page No. _____
Date _____

Computer Architecture & Organization

Unit-I

1) ~~Basic~~ Difference betn Computer Organization & Computer Architecture, Computer Types, functional Units, Basic operational Concepts: Bus Structures, Generation of Computer, ~~Evolution of performance~~ Introduction to Computer Operation with a Simple 8 bit - Instruction Computer illustrating assembly and machine language. [Register Transfer language, Register Transfer Bus & memory transfers, Arithmetic Microoperations, Logic micro operation, ~~to~~ Shift microoperations, Arthmetic logic Shiftunit]

Unit II

Instruction Codes, Registers, Buses, Design of computer Instruction Timing & Control, Instruction cycle, Memory Reference Instruction, I/O Interrupt, Design of Basic Computer, Accumulator logic. [Programming the basic Computer - Machine language, Assembly language, Assembler, Address Sequencing, Microprogram, ~~Design of Control unit, Register and Stack Organization~~, Instruction format, Addressing Modes. ~~Data Transfer and manipulation~~.

Unit III:-

~~RISC and CISC processor~~, Computer Arithmetic - Addition & substraction with signed magnitude, multiplication and Division Algorithms, Booth Multiplication Algorithm, ~~Array multiplier~~, Hardware Divide overflow.

Scanned by CamScanner

bit.ly/Medicaps
Aman - 9977770831
20/9/18

Page No. _____
Date _____

implementation for signed - Magnitude and hardware algorithm, Divide overflow, floating point Arithmetic operations - Add, sub, Muliplicatt & Div.

Unit IV :-
I/O organization, I/O Interface, Synchronous Vs Asynchronus Data Transfer, Modes of Transfer, Interrupt & its priority, DMA, I/O Processor, Memory Hierachy, Main Memory, Auxillary Memory, Associative Memory, Cache Memory, Virtual Memory, Memory Managment Hardware

Unit V :-
Flynn's classification, RISC & CISC Processor, Pipeling and Vector processing, parallel processing, Array processor, Multiprocessor Architechure Organization - Performance characteristics of Multiprocessors Multi-Core Architecture, Inter-processor - Communication, Cluster Computing, Graphic Processor Units, System-on-Chips, Flynn's Classification.

Scanned by CamScanner

designing of structure

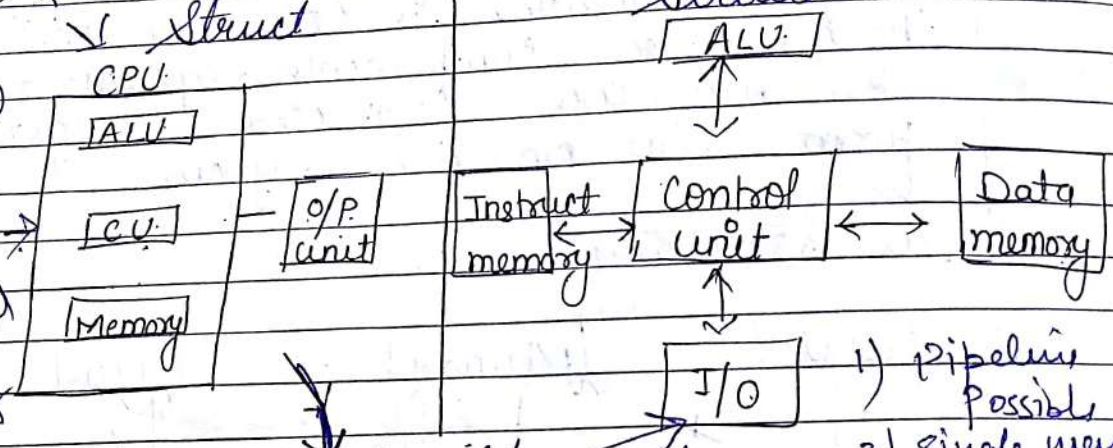
Von-neumann model

Harward model (latest)

(Pipeling) 2 cycle (Data & Ins)

Bottom up

Struct



Von-neumann model = ideal
2) double memory
3) ~~double~~ single side buses

1) pipeline Possible
2) single memo
3) Double side buses.

pipe-lining not possible

Parallel inset is executed in a particular manner

★ generation of Computer

I – Vacuum tubes, Magnetic drums
II – Transistors
III – IC
IV – Microprocessor
V – AI

★ ★ Computer Types

→ Super Computer
→ Mainframe
→ Mini
→ Micro

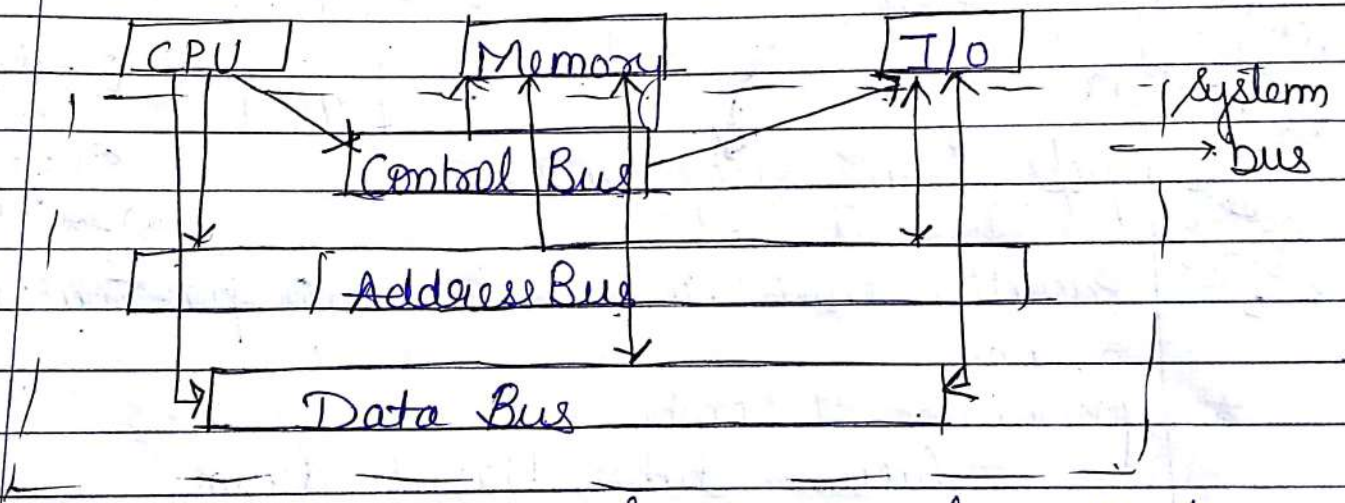
Scanned by CamScanner

Page No. _____
Date _____

★ CAO :- Computer arch is concerned with the Structure of or hardware design of Computer Compondo. It also includes the info formate & Techniques for addressing memory.

Comp organisation is concerned with the way the hardware components operate & the way they are connected together to form the Computer system.

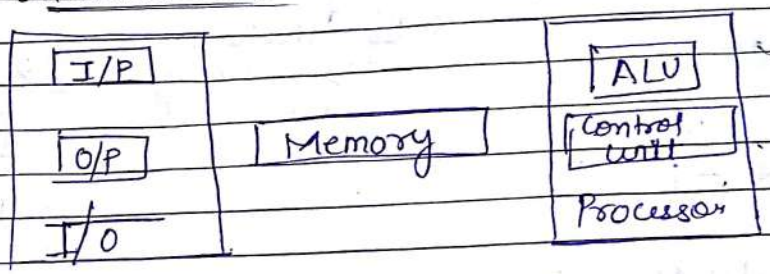
★ Bus Structure



System bus is a single comp bus that connects the major component of a computer syt^m, combing the fu^n of all 3 data bus, to carry inf^n, an address bus to determine where it should be send & a control bus to determine its operations

Page No. _____
Date _____

Functional unit



I/P :- Input unit are used by a Computer which read the data.
ex: joystic, tracball, mouse, Keyboard.

O/P :- Output unit are used to send the processed result to the user.
~~ex:- memory~~.

Memory :- Memory unit is used to stored program and data.

ALU :- most of all operation of a Computer are executed in the ALU of the processor.
ex:- +, -, *, %

Control unit :- Control unit is used to Cordinate the operations of input, Output, memory, ALU. It coordinate the operation of all the unit using Control signals like timing signal.

* Timing Signal are the Signal determing the time when a given action must take place.

Scanned by CamScanner

LSB
least significant bit

MSB
Most significant Bit

Page No.
Date

Instru
Program control
MAR
Memory Address

Register Transfer language (RTL)

(betn data of Registers only)

Excess wire
↓
Bus transfer
Multiplixer

RTL

$R_2 \leftarrow R_1$

$R_3 \leftarrow R_1 + R_2$

Notation

P: $R_2 \leftarrow R_1$
(boolean variable)

R1	87654321.0
15 0	15 87 0
R2	P.C(H) PC(L)

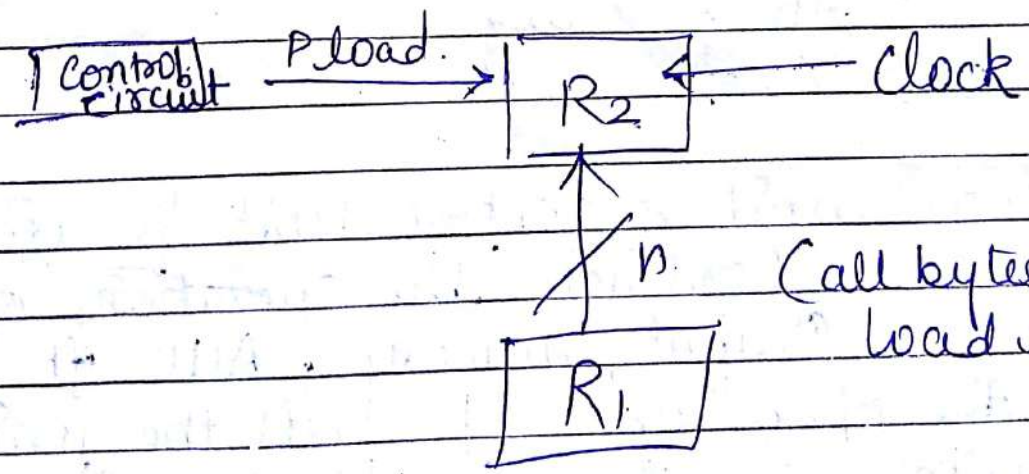
Types of Resiger

IR (Instruction Register)
PC (Program Control)
MAR (Magnetic address Register)
R1 Processor Resiger

Block diagram

* If (P=1) then ($R_2 \leftarrow R_1$)

P: $R_2 \leftarrow R_1$



(all bytes of R_1 is load in R_2)

Scanned by CamScanner

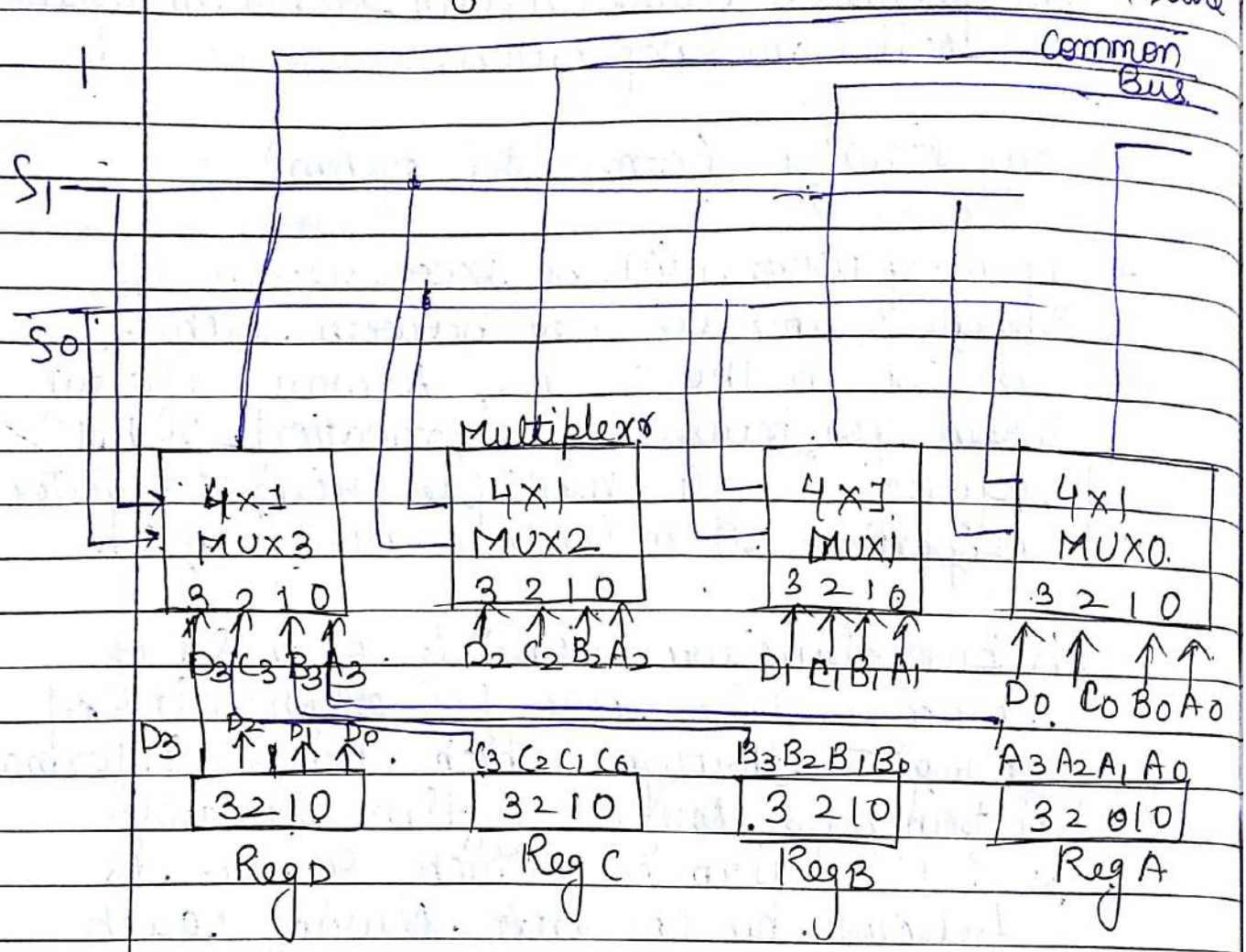
S_0	S_1	
0	0	A
0	1	B
1	0	C
1	1	D

Selection input

Page No. _____

Date _____

Bus transfer



$Bus \leftarrow C$

$R_1 C \leftarrow Bus$

Memory transfer (transfer data betn memory, 00)

Read : $DR \leftarrow M[AR]$ (Memory [address Register] data)

Write : $M[AR] \leftarrow R_1$

$DR \leftarrow M[AR]$

$M[AR] \leftarrow R_1$

$R_1 = 20$

$R_2 = 40$

Scanned by CamScanner

24/8/18

Page No. _____
Date _____

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

Bus Transfer (common bus system)

* The no. of wires will be ~~exec~~ excessive if seperate lines are used between each register in the system. A more efficient schem for transfering information betn register in a multiple ~~resiger~~ register configration is a common bus system.

A bus structure consiste of a set of common lines one for each bit of a register through which binary information is transfered one at a time. control signal determine which register is selected by the bus during each register transfer.

The Multiplexer select the source ~~Resiger~~ resister whoes binary infn is then placed on the bus in general.
A bus system will multiplex K register of n bits each to produce an n line common bus. The required multiplexer is ~~K~~ n. (no of bits in each register).

The size of multiplexer must be K×1 since it multiplexes K data lines.

Scanned by CamScanner

Page No. _____
Date _____

Memory transfer

Read operation :- This causes a transfer of Information into DR from the memory word (M) selected by the address in AR.

Arithmetic Microp-operation (operation perform on Registers)

$R_3 \leftarrow R_1 + R_2$
$R_3 \leftarrow R_1 - R_2$
$D \leftarrow A + 1$
$R_2 \leftarrow \overline{R_2}$
$R_2 \leftarrow \overline{R_2} + 1$

7 basic 8

Logic Micro-operation

$F \leftarrow 0$ clear
$F \leftarrow A \wedge B$ AND
$F \leftarrow \overline{A \wedge B}$ NAND
$F \leftarrow A$ transfer A
$F \leftarrow A \oplus B$ x-Or
$F \leftarrow \overline{A \oplus B}$ X-NOR
$F \leftarrow A \vee B$ - OR
$F \leftarrow \overline{A \vee B}$ NOR

16 basic 4

Shift Micro-operation

→ Logical shift
→ circular shift
→ Arithmetic shift

K = Right
n = no of bit

Page No. _____
Date _____

① Logical :

$R_1 \leftarrow shlR_1$ $R_2 \leftarrow shrR_2$

from Serial input zero
(By default add 0)

100101 shift left
001010

Loss of inf^n

② Circular shift :

$R_1 \leftarrow CirR_1$ $R_2 \leftarrow CilR_2$

10110110 - Shift Right No Loss of inf^n

01011011

③ Arithmetic Shift : (A Shl A Shr)
(Sign unchanged)

-103 ÷2 10011001 — ARS
(At Right shift)
-52 11001100

Arithmetic left Shift

⊕ — +
1 — —

11011001 — -39 ×2
10110010 — -78

(If 0 come in left shift so it will be error of the code)

1 = —
0 = +

Page No. ____
Date ____

(1)

Logical shift :- This two micro-operation specify a one bit shift to the ~~fet~~ left of the content of register R1, and one bit shift to the right of the content of the Resigeter R2.
The bit transfered to the end position through the serial input is asume to be zero. during a logical shift.

(2)

Circular-shift - The circular shift circulate the bits of the register around the two ends without loss of information.

Arithmetic shift :- An arithmetic-shift is micro oper that shift the end binary no to the ~~fe~~ left or right.

~~If~~ An AS left multiply a singed binary no by two.

An AS Right divides the no by 2.

AS must leave the sign bit unchanged because the sign of the no. remains the same when it is multiplied or divide by 2.

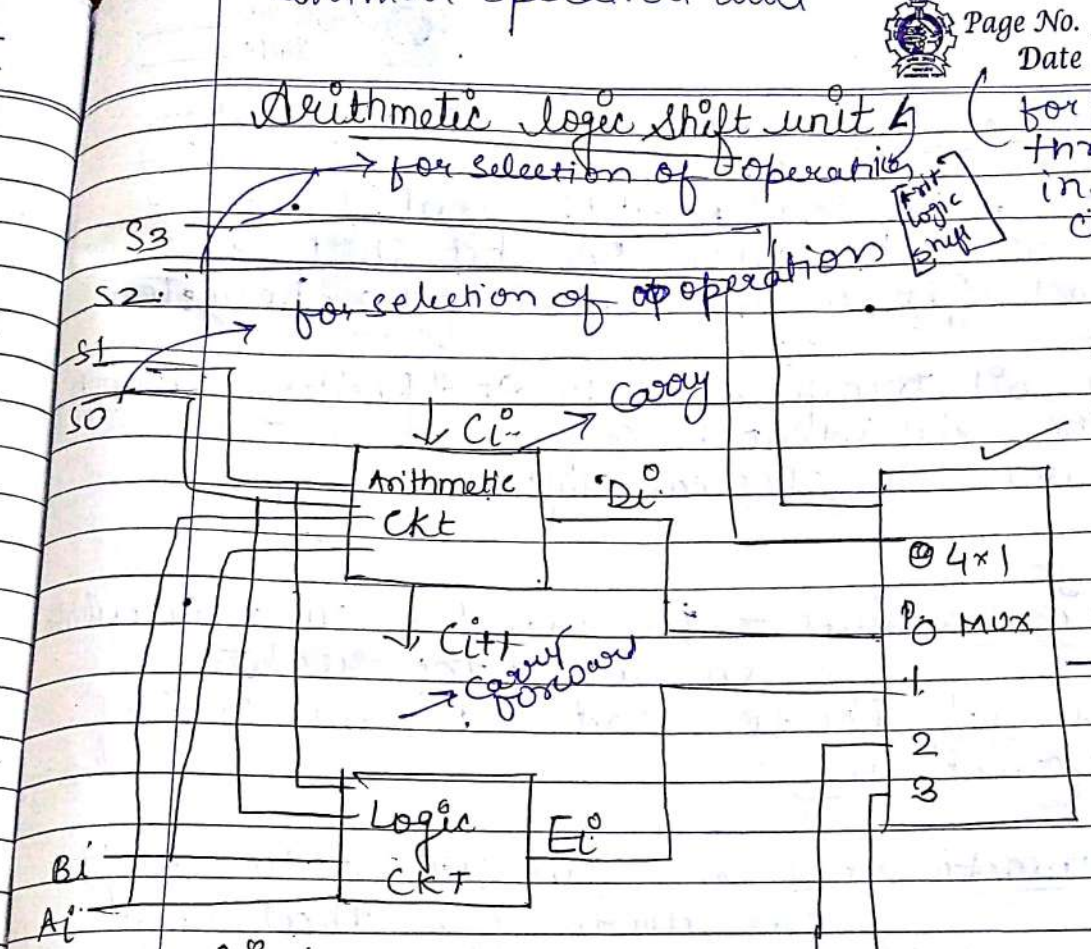
The left most bit in a Register hold the sign bit & Remaine bits hold the magnitude.

Scanned by CamScanner

Common operation unit

Page No. _____
Date _____

Arithmetic logic shift unit (for performing three operations in single circuit)



Operations

Arithmetic:
- F = A
- F = A + 1
- F = A + B
- F = A + B + 1
- F = A + B
- F = A + $\overline{B}$ + 1
- F = A − 1
- F = A

Logical:
- F ∧ B
- F ∨ B
- F ⊕ B
- F = $\overline{A}$

Shift:
- F = shr A
- F = shl A

Page No. _____
Date _____

S_3	S_2	S_1	S_0	C_{in}	
0	0	0	0	0	
0	0	0	0	1	
0	0	0	1	0	
0	0	0	1	1	
0	0	1	0	0	
0	0	1	0	1	
0	0	1	1	0	Arithmetic
0	0	1	1	1	
0	1	0	0	X	
0	1	0	1	X	
0	1	1	0	X	Logical
0	1	1	1	X	
1	0	X	X	X	Shift
1	1	X	X	X	

0 — 00
1 — 01
2 — 10
3 — 11

* Instead of having individual registers performing the microoperation directly computer System employ a no. of storage registers connected to a ALU.

To perform a microoperation the contents of specified registers are placed in the input of the common ALU. the ALU perform an operation and the register of the operation is than transferred to a destination register.

Page No. _____
Date _____

Diagram

→ Input A_i & B_i are applied to both the arthimetic and logic units...
A particular microoperation is selected with input S_1 & S_0
A Multiplixer at the output choose beth an Arithmetic ~~or~~ otput in D_i & logical operation E_i. The data in the Multiplixer are selected with inputs S_2 + S_3.
The other two data input to the multiplixer receive inputs A_{i-1} for the shift right operation and A_{i+1} for the shift left operation.

The Output carry C_{i+1} of a given Arithmetic Stage must be connected to the input carry C_i of the next stage in a sequence.

The input carry to the first stage is C_{in}. which provides the selection variable for the arithmetic operation.

The first 8 are arithmetic operation for which S_3 & S_2 = 0, 0.
Next 4 are logical operation for which S_2 & S_3 = 0, 1
The input carry has no affect during logic operation i.e this it is marked dont care (X)

The two operation are shift operation are selected with S_3, S_2 = (1,0) & S_3, S_2 = (1,1)

Scanned by CamScanner

Page No. ____
Date ____

Unit 8- II

Instruction Code (group of bit)

Decided by action AC to operand power of

operant address is here

15	12 11	0
Opcode	Address	

Inst^n format

4 bit so 16 operation perform

Stored program organisation

Memory 4096×16 words (2^{12})

it helps to [illegible] bit

each word length 16.

Inst^n (Program)
operand (data.

5 + 4

a + b

a

Accumulator Reg. AC → Processor Register

If only one operant is needed so it is present here

Indirect Add → only 8 operation performing

15	14	12 11	0
I	Opcode	Address	

Memory

21	0	Add	450
450	Operand		

→ Effective address

Imidiate = Opcode + 1 operant & other is come from AC

0 – Direct Address

1 = Indirect Adres

Direct

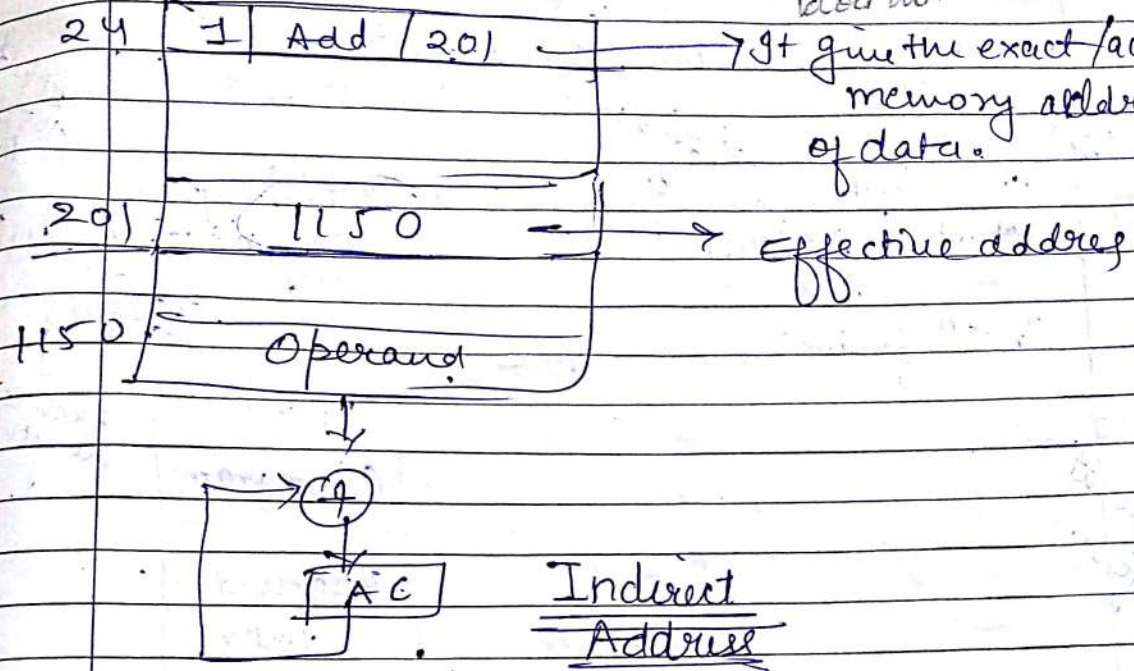
+ → a

AC b

Addressing mode — [Indirect, Direct, Imediate] ①

Page No. _____
Date _____

Memory



Indirect Address

* Effective dress = final address

* A Instruction code is a group of bit that instruct a Computer to perform a specific operation. It is divided into two parts
1) Address
2) opcode

The operation code of an instruction is a group of bits that define operations such as add, multiplication, shift.

The no of bits required for the operation code of an instruction depends on the

Page No. ____
Date ____

total no. of operation available in Computer. The operation code must Consists of atleast n beats for a given 2^n distinct operation.

Stored program Organisation

The first part specify the operation to be performed and the second specify an address.

The Memory address tells the Control where to find an operand in memory

This operand is data from memory and uses as the data to be operated on together with the data stored in the processor register.

Example:-

for the Memory unit 4096 words we need 12 bits to specify and address ($4096 = 2^{12}$) If we store each instruction code in ~~16~~ one 16 bit memory word we have available 4 bits for the operation code to specify 16 possible operation & 12 bits to specify the address of an operand.

The operation is performed with the memory operant & the Content of AC.

Page No. _____
Date _____

If an operation is an Instruction code does not need an operant from memory. the rest of bit in the instruction can be used for other purposes

Example operation such as clear AC, Complement AC. That not need an operant from memory.

21/8/18

Indirect Address :-

When 2nd part of Instruction code specify an operant the instn is said to have an Immediate operant. When second part is specify the adress of operant the instn is said to have direct address when ~~its~~ bits in second part of instruction contains the address of a memory word in which the address of the operant is found is called indirect address. 1 bit of instruction code can be used to distinguish b/w a direct & an indirect address

Computer Register

DR - Held operant
IR - address from memory
AR - that address store in
AC - operant store
PC - address of next

DR	16
AR	12
AC	16
IR	16
PC	12
IR	16
INPR	8
OUTR	8
Temp	

Bus Transfer [Three state buffer or Multiplexer]

Page No. ____
Date ____

DR :- Data R hold memory operant
AR :- Address R hold address for memory register
AC :- Accumulator holds operand (It is precise)
IR :- Instruction R holds instruction code.
P.C :- Program Counter hold address of next instruction.
~~DR~~ TR :- Temporary R hold Temporary data
INPR :- Input register holds input characters
OUTR :- Output register holds the Output Character for the O/P device.

Common Bus

* The no. of ~~mites~~ wires will be excessive collect^n are made b/w the O/P of each registers & the I/P of the other register a more efficient scheme for transfering Information in a System with many registers is to used a common bus. The O/P of 7 registers & memory are connected to the Common bus. The Specific O/P ie selected for the bus lines at any given time is determined from the binary value of the select^n variable S_2, S_1, S_0.

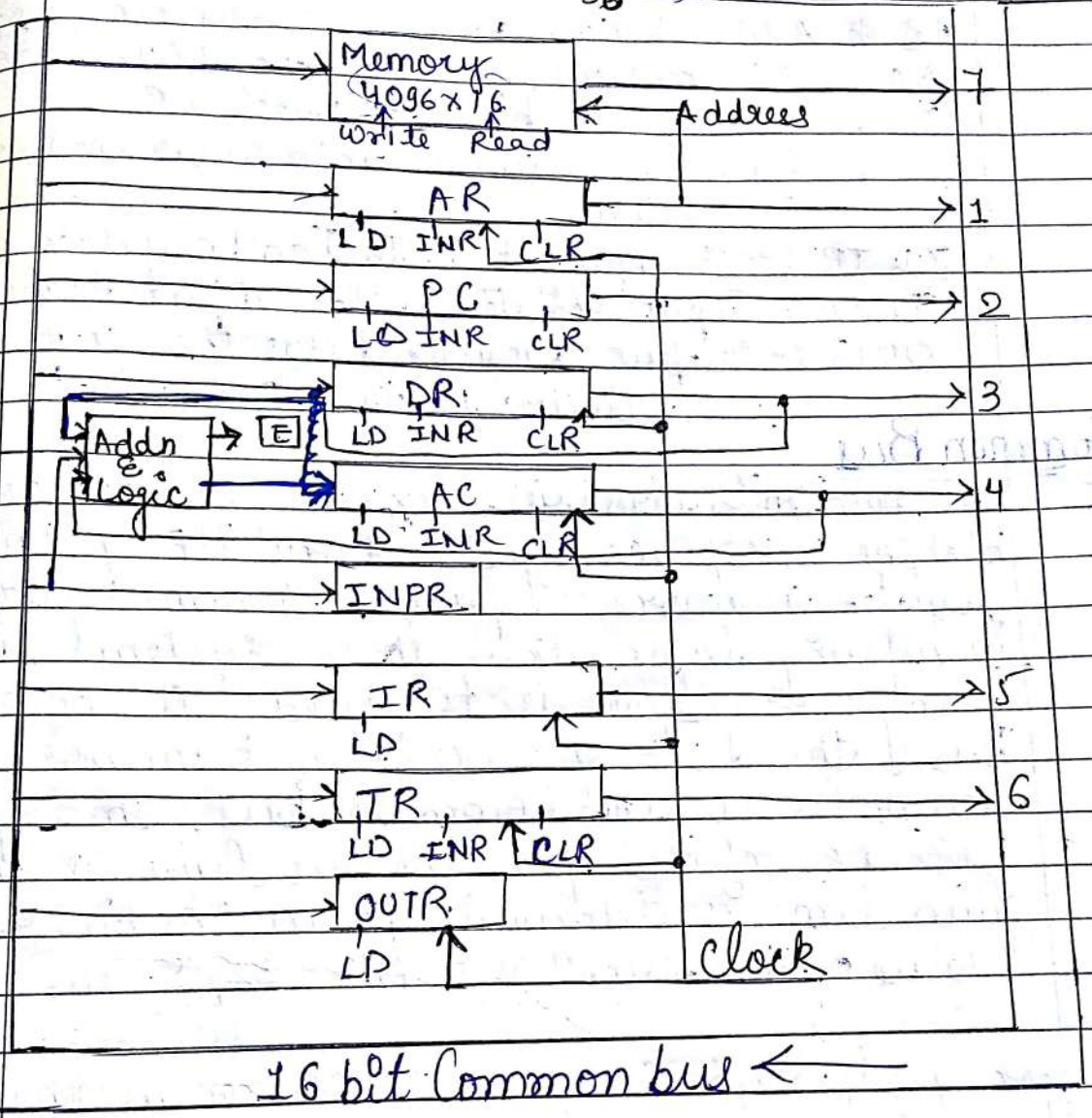
for eg:- If S_2, S_1, S_0 equals to 011 means the DR is selected for placed data on the bus. The particular register whose load signal (LD) I/P is enable receive the data from the bus during next clock pulse transit^n.

The memory plays its 16 bits O/P on the bus when the read I/P is actuated & when $S_2 S_1 S_0 = 111$. The memory receives the contents of the bus when its write I/P is actuated.

Scanned by CamScanner

4096 = 2^12 (3 state buffer)

Buses System / Common Bus



INPR is Connected to provide infⁿ to the bus but OUTR can only receive infⁿ from the bus.

This becoz INPR receives a character from an I/P device which is then transfered to AC.

Page No. _____
Date _____

(Types of Instruction)

Computer Instruction

① Memory Reference Instⁿ

15	14	12–11	0
I	opcode	Address	000110

② Register Reference Instⁿ

15	12–11	0
0111	Registerop^n	

Opcode = 111
I = 0

0 → Reg

③ Input - Output Instⁿ.

15	12–11	0
1111	I/O operation	

Opcode = 111 I = 1

~~Memory RI~~

* The basic Computer has three instruction Code formate each formate has 16 bits the operation code part of the instruction contains three bit and the meaning of the remaining 13 bits dependens on the Operation Code encountered. A nee

2

Page No. ____
Date ____

* A Memory Reference Instn uses 12 bits to specify an address and 1 bit to specifying addresing mode I. (I=0 Direct address, I = IR).

* The Register Reference Instn are recoriged by the opcode (111) with zero in the left most bit of the instruction. A Register Refn Instn specify an operation on the AC Register. An operand from memory is not needed therefore the other 12 bits are used to specify the operation.

* An In/Output Instn does not need a reference to memory and is recognised by opcode (111). With 1 in the left most bit. The remaining 12 bits are used to specify the type of I/P Operation.

The type of Instn is Recognised by the computer control from the 4 bits in positions (12 ~~through~~ to 15) of the instruction.

* A computer should have a set of instruction so that the user can construct machine language program to evaluate any fun. that is known to be computable.

Scanned by CamScanner

Page No. ____
Date ____

the set of instⁿ are said to be complete if the computer include a sufficient no. of instruction in each of the following categories

1) Arithmetic, logical, shift instruction

2) Instⁿ for moving infⁿ to ~~f~~ and from memory and registers.

3) Program control instruction together with instructions that check status condⁿ.

4) I/P instruction.

Memory Refⁿ Instⁿ

1) AND :- It AND memory word to AC.

2) ADD :- It Add memory word to AC

3) LDA :- It load memory word to AC

4) STA :- It store content of AC in memory.

Register

1) CLA :- Clear AC
2) CLE :- Clear E (E flip flop)
3) CMA :- Complement AC

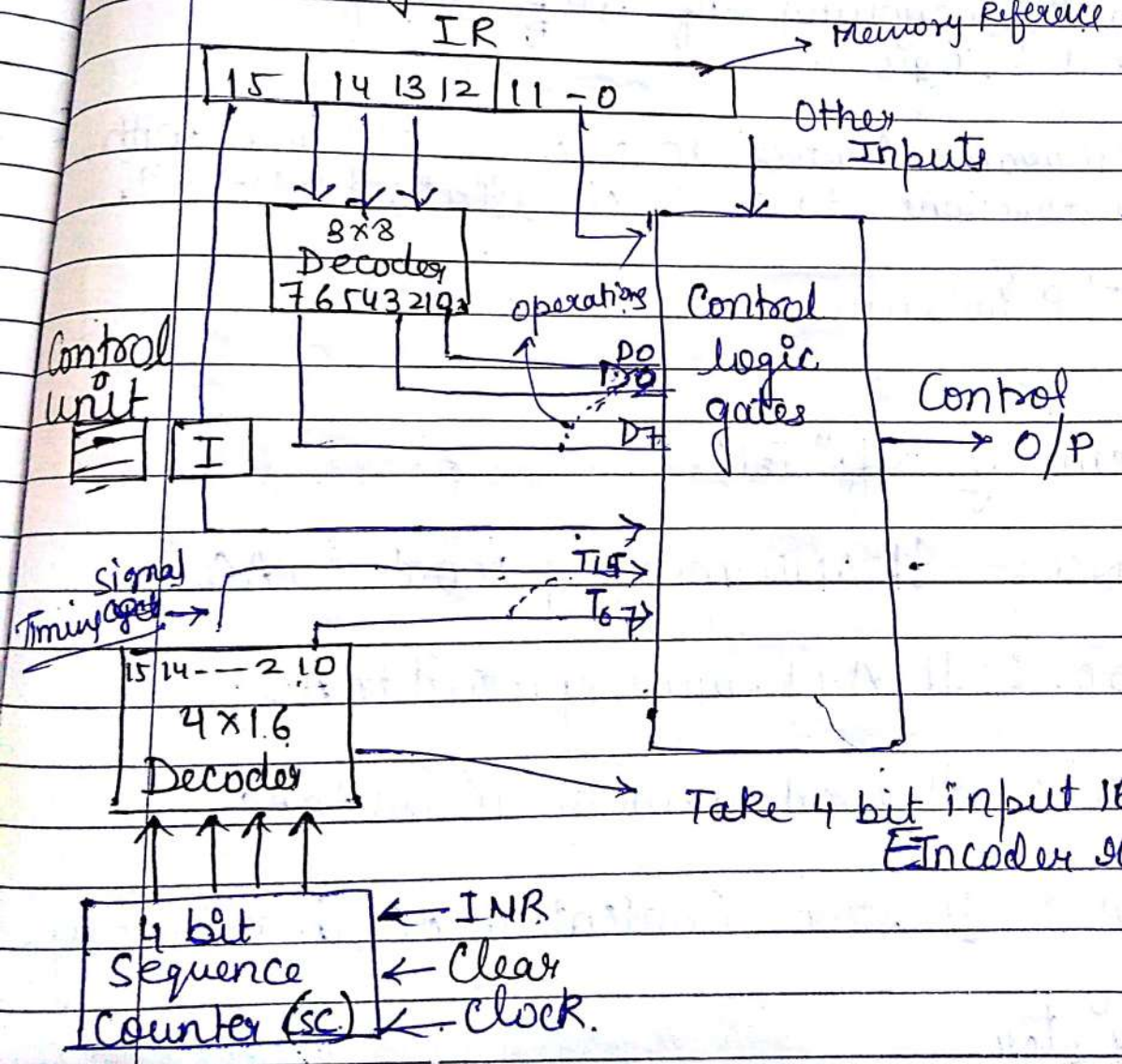
Scanned by CamScanner

Page No. _____
Date _____

(2) I/O

- INP - Input Character to AC
- OUT - Output Char from AC
- ION - Intrupt on
- IOF - Intrupt off

Timing & Control



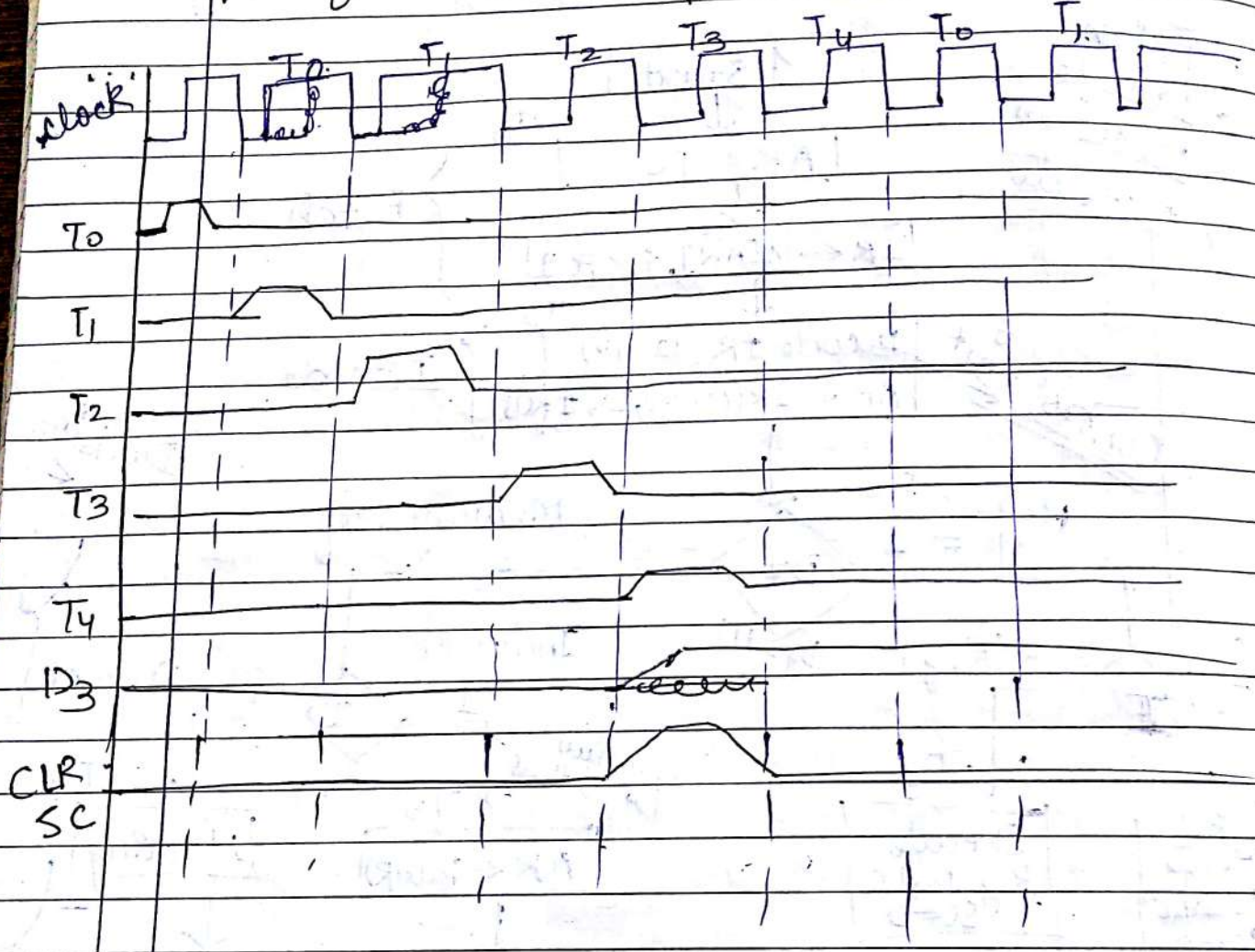
Scanned by CamScanner

Then D_3 action active

Memory operation Page No. Timing signal (4)

$D_3 T_4 : SC \leftarrow 0$

Example of control timing signal



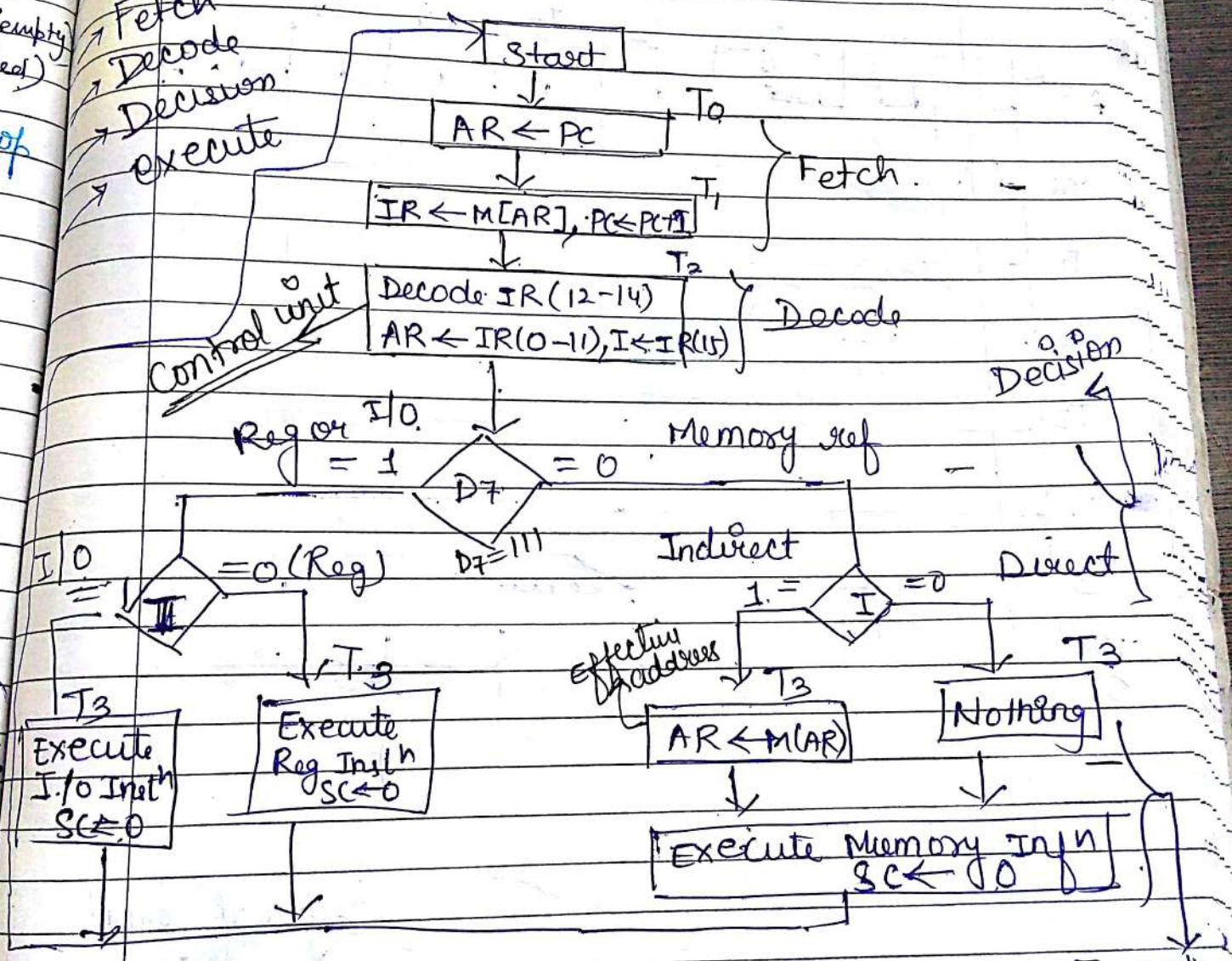
Types of Control organization

- hardwired → gates, flip flop, decoders etc.
- microprogrammed control → Control memory

Scanned by CamScanner

Page No. _____
Date _____

Instruction Cycle



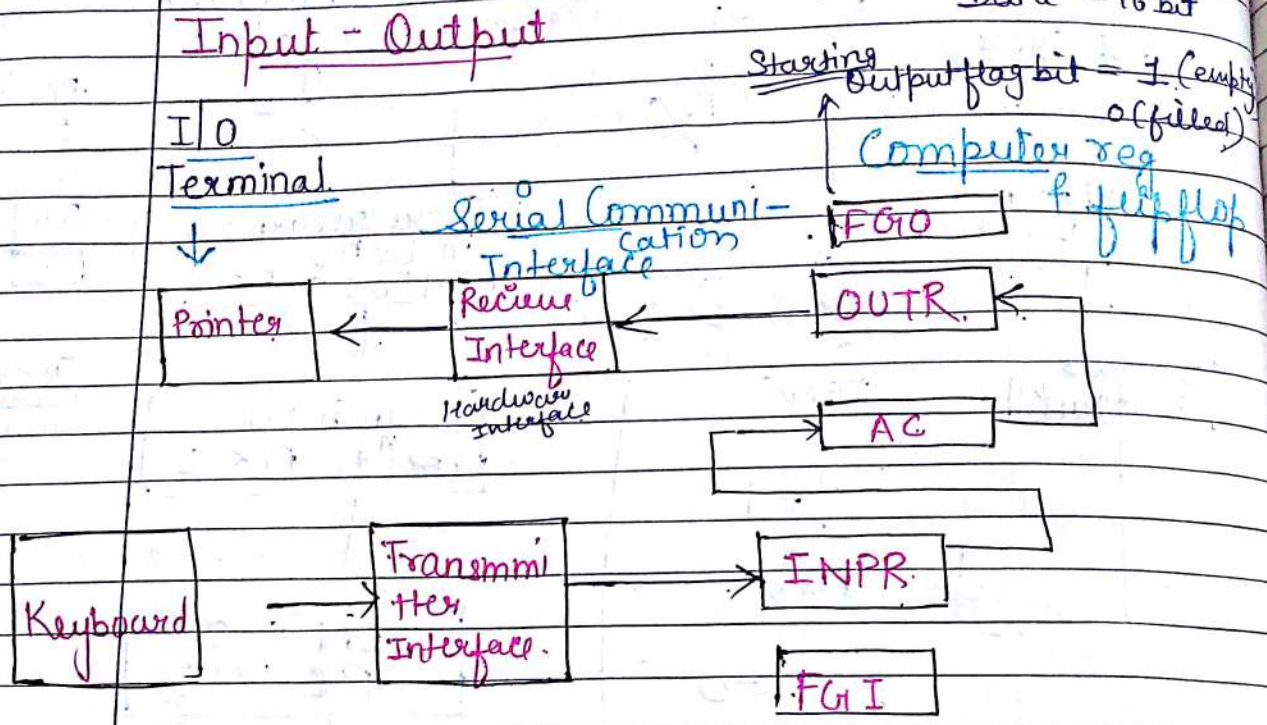
Page No. _____
Date _____

4096x16

INPR = 8 bit
Data = 16 bit

Input - Output

I/O Terminal



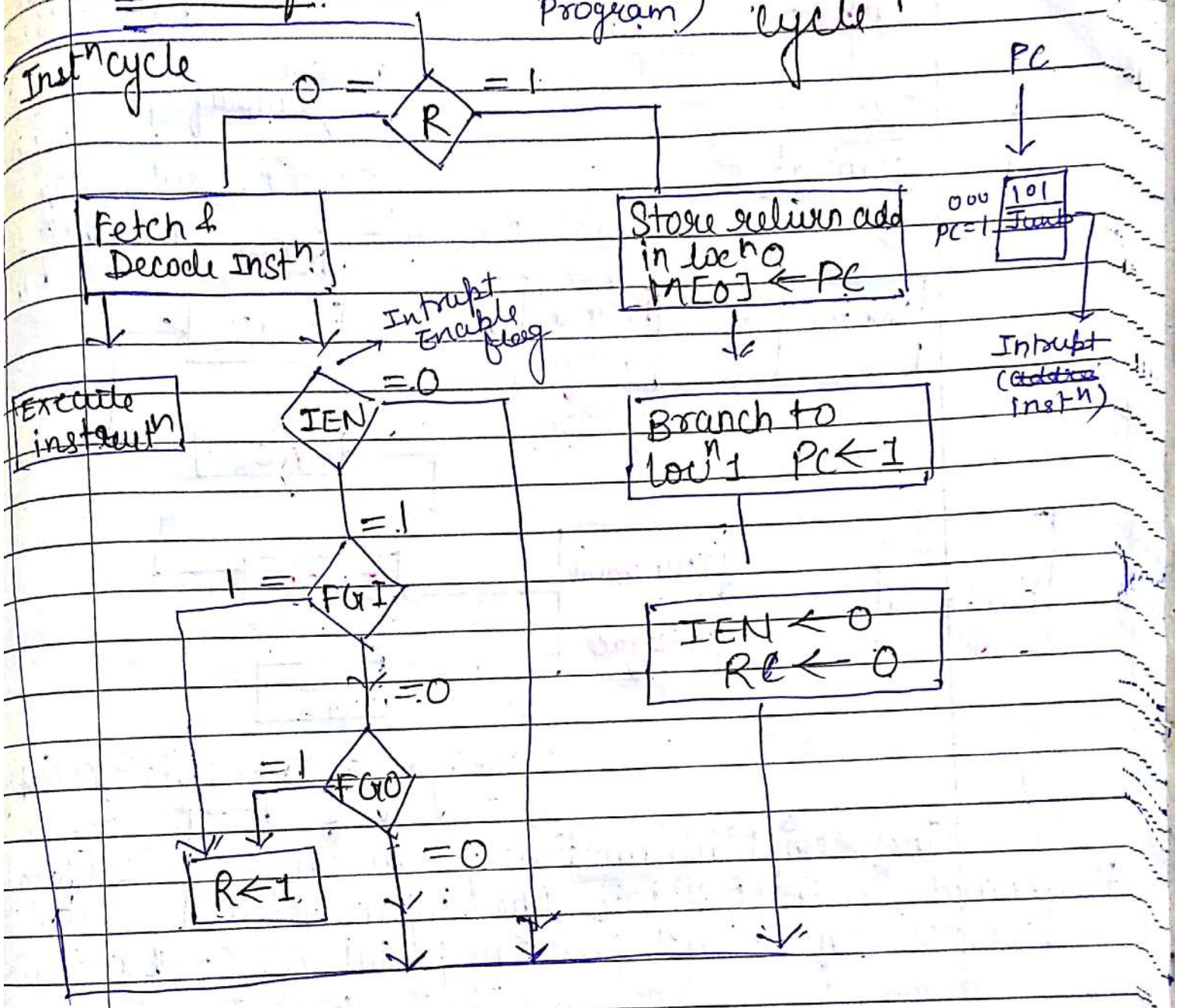
Starting → flag Input bit = 0 (empty)
1 (data filled int)

The Serial information from the Keyboard is shifted into the input register INPR the inf^n for the printer (output device) is stored in the Output Register Outer. The input register INPR consists of 8 bits and hold an input information the 1 bit input flag (FGI). is a Control flip-flop. the flag bit is said to One when new information is available in the input device and is clear to zero when the inf^n is excepted by the Computer. As Along as the flag is set the ~~ft~~ inf^n in INPR cannot be Changed the Computer checks the flag bit if It is One the inf^n from

(PTO)

Interrupt

(if other process is come parallel to Program)



→ Intrupt is executed after the processing process. if intrupt is not come then the PC next address process is executed.

Scanned by CamScanner

Page No. ____
Date ____

INPR is transfered in AC and FGI clear to zero. Once the flag is Cleared new inf^n can be shifted into INPR by striking another key. the OUTR Register work similarly but the direction of inf^n flow is reversed.

Intially the Output flag (FGO) is set to 1 the Computer checks the flag bit if it is 1 the the inf^n from AC is transfered into Outer register and FGO is cleared to zero. The Output device accept the information and print the corresponding ~~car~~ character when the operation is complete it set FGO = 1 the Computer does not load a new inf^n into OutR when FGO is zero because this cond^n indicate that the Output device is in process of printing the Character.

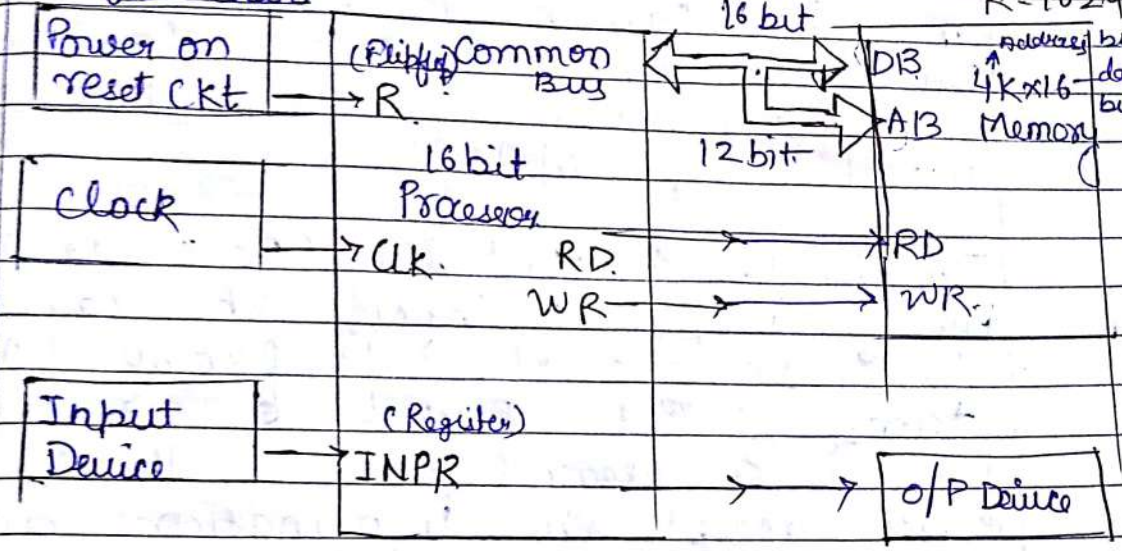
Scanned by CamScanner

Page No. _____
Date _____

Design of Basic Computer

Hypothetic flow Chart

2^{12}
4096×16
K=1024



~~I/P – PB6 – PB11~~
~~Reg 7/0 – R4(B11)~~

~~D7 I' T3 = γ~~
~~D7 I T3 = P~~

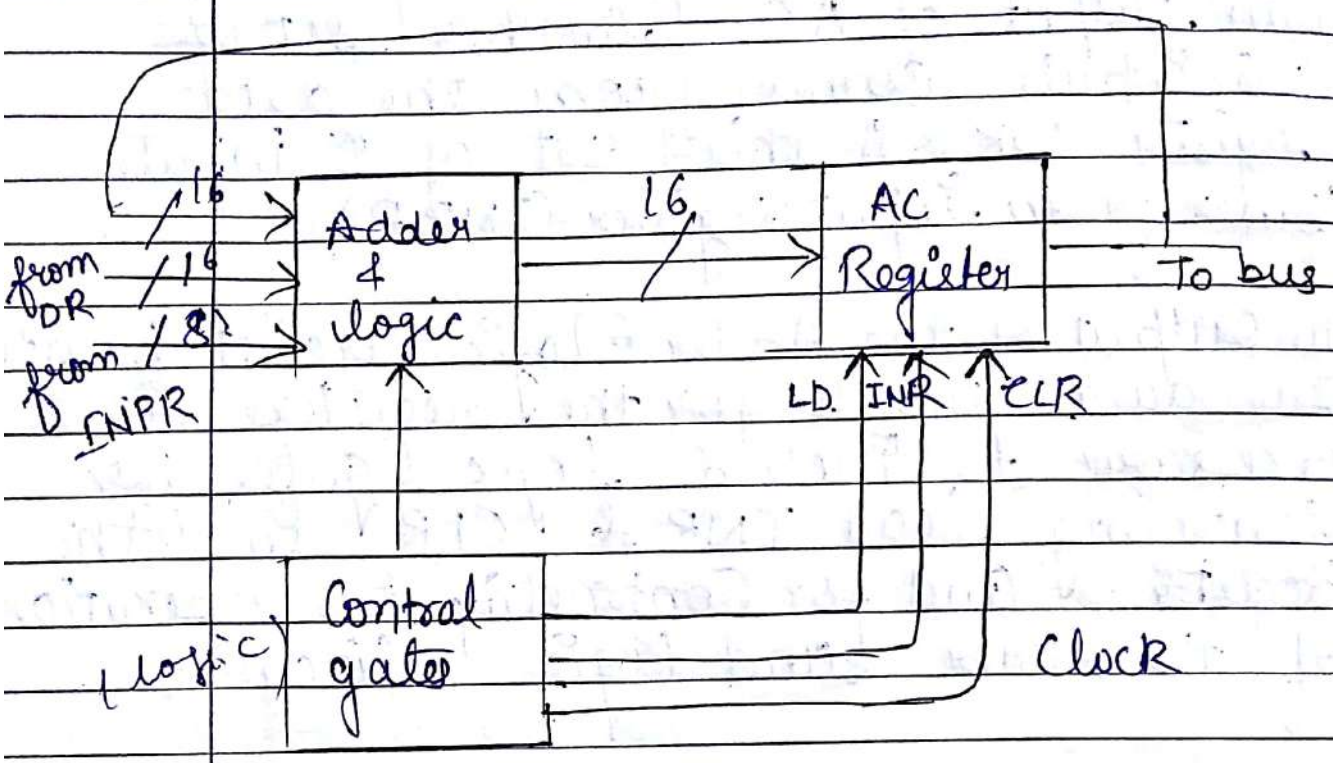
1) A Memory unit with 4096 words of 16 bit each.
2) 9 register, AR, PC, DR, IR, TR, INPR, OUTR, SC, AC.
3) 7 flip-flops, ~~IEN~~ IEN, E, S, R, FGI, FGO
   ↓ Interrupt
4) 2 Decoders :- 3×8 (operation decoder), 4×16 (timing decoder)
   Ctl
5) A 16 bit Common bus.
6) Control logic gates.
7) Adder & logic circuit connected to the input of AC.

Scanned by CamScanner

I/P reg — PB_{11}, PB_6, rB_{11}, rB_9

Design of Accumulator logic

$D_7 I' T_3 = r$
$D_7 I T_3 = p$



D_0T_5 : $AC \leftarrow AC \wedge DR$
D_1T_5 : $AC \leftarrow AC + DR$
D_2T_5 : $AC \leftarrow DR$
PB_{11} : $AC(0-7) \leftarrow INPR$
rB_9 : $AC \leftarrow \overline{AC}$
rB_7 : $AC \leftarrow shrAC$
$AC(15) \leftarrow E$
rB_6 : $AC \leftarrow shlAC$, $A(0) \leftarrow E$
rB_{11} : $AC \leftarrow 0$
rB_5 : $AC \leftarrow AC + 1$

Scanned by CamScanner

Page No. _____
Date _____

Design of AC

* The Adder & logic circuit has 3 sets of inputs ~~on~~ 1 set of 16 input comes from output of AC. Another set of 16 inputs comes from the data register DR. A third set of 8 inputs comes from input register (INPR).

The output of the Adder & logic circuit provide the data inputs for the registers. It is necessary to include logic gates for controling load INR & CLR in the Register and for controlling the operation of the Adder and Logic Circuit.

Scanned by CamScanner

Page No.
Date

Instruction Format

opcode	mode	Address

↳ Single Accumulator Organization
↳ General Register Organization
↳ Stack organization

(1) ADD X one addrs fmt
AC ← AC + M[X]

(2) ADD. R_1, R_2, R_3
$R_1 \leftarrow R_2 + R_3$ Three Register addrs

Scanned by CamScanner

Page No. _____
Date _____

Addressing Modes

- ↳ Implied
- ↳ Immediate
- ↳ Register
- ↳ Register Indirect
- ↳ Autoincrement or Auto-decrement
- ↳ Direct
- ↳ Indirect
- ↳ Relative Address
- ↳ Base Address
- ↳ Index Address

500

Opcode	Mode	Address

Q To ~~address~~ word Instn. at address 200, 201 is a "Load to AC" Memory address field conten 500. Calculat the content of AC & EA.

PC = 200	200	Load to AC	Mode.
R1 = 400	201	Address 500	
XR = 100	202	Next Instn.	
AC			
	399	450	
500 + 202 = 702	400	700	
	500	800	
500 + 100	600	900	
	702	325	
	800	300.	

Address Not Requir & Implied & Immediate mode.

Am { Indexing of Array, pointer

Page No. Date

Register 23

(8)

* The Addressing mode Specify, a rule for interpreting ~~or~~ modifying the address field of the instruction before the operant is used.

PC = 200
R = 400
Index Register XR = 100

* Purpose of Using Addresing mode

1) To Reduce the no. of bits in address field of the Instruction.

2) To provide programing Versatility to the user by providing such facilities as pointer to memory, Indexing of data and program relocation.

* Implied mode :- The operant are specified implecity in the Defination of the instruction.

Ex :- AC

* Immediate mode :- The operant is specified In the Instruction itself means instruction contons an operand field rather than an address field.

* Register mode :- operands are in the Register that ~~resides~~ ~~register~~ resides within the CPU.

Scanned by CamScanner

Page No. _____
Date _____

* Register Indirect mode :- Regis Instn specify a Register in the CPU whose Contents give the address of the operante in the memory.

* Auto In. & Aut Dec :- Similar to the RIM Accept that the register is incremented or decremented after or before its value is used.

* Direct :- Operands reciets in the memory and its address is given directly by the address field of the Instn.

* Indirect :- The address field of the Instn gives the address where the effective address is stored in memory.

* A few addressing mode required that the the address field of Instn be added to the Content of a specific Register in the CPU.

Effective address = address part of Instn + Content of the CPU register

* Relative address = PC + Address field of Instn

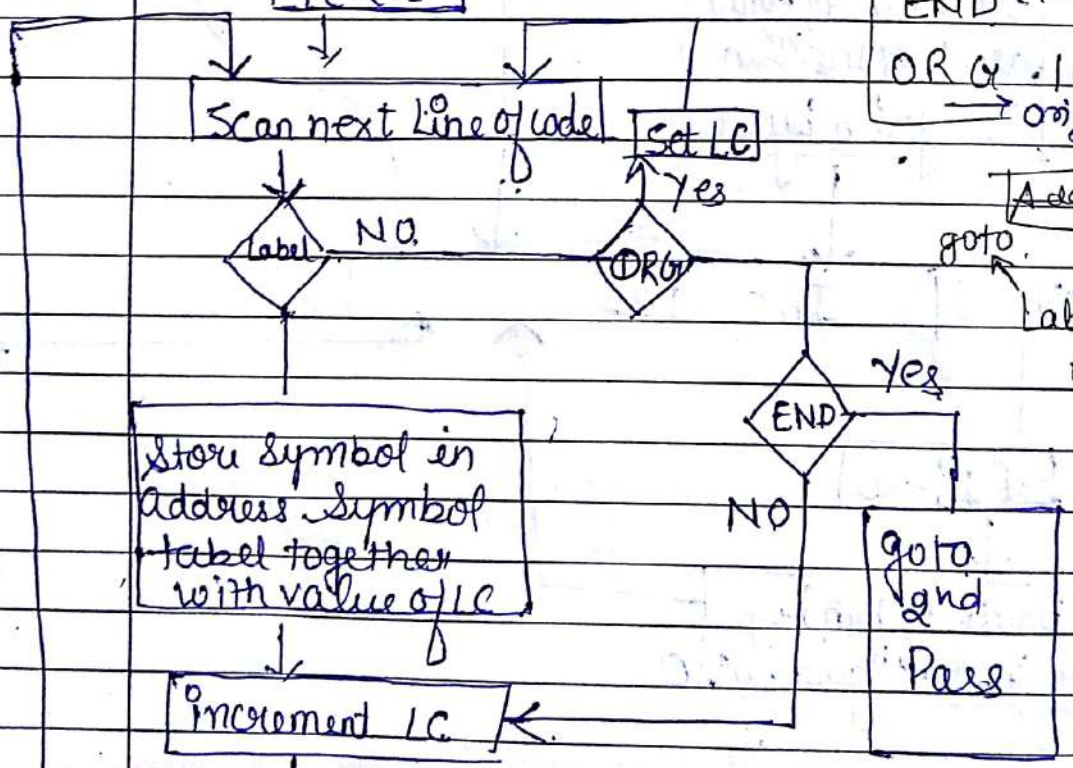
* base address = base address + Address field
* Index address = Index address + Address field.

Scanned by CamScanner

Add Mode	Effective Address	Content of AC
Direct	500	800
Immediate	201	500
Indirect	800	300
Relative	702	325
Indexed	600	900
Register	—	400
Register Indirect	400	700
Auto Inc	400	700
Auto decr	399	450

Assembler First Pass (Istor label & Sydo Instn)

LC = Location Counter



Sydo Instn
DEC
HEX
ORG
END → Program end
ORG 1000 → origin

Non MRI Table
Register & I/O Instn Table

Address symbol table

goto

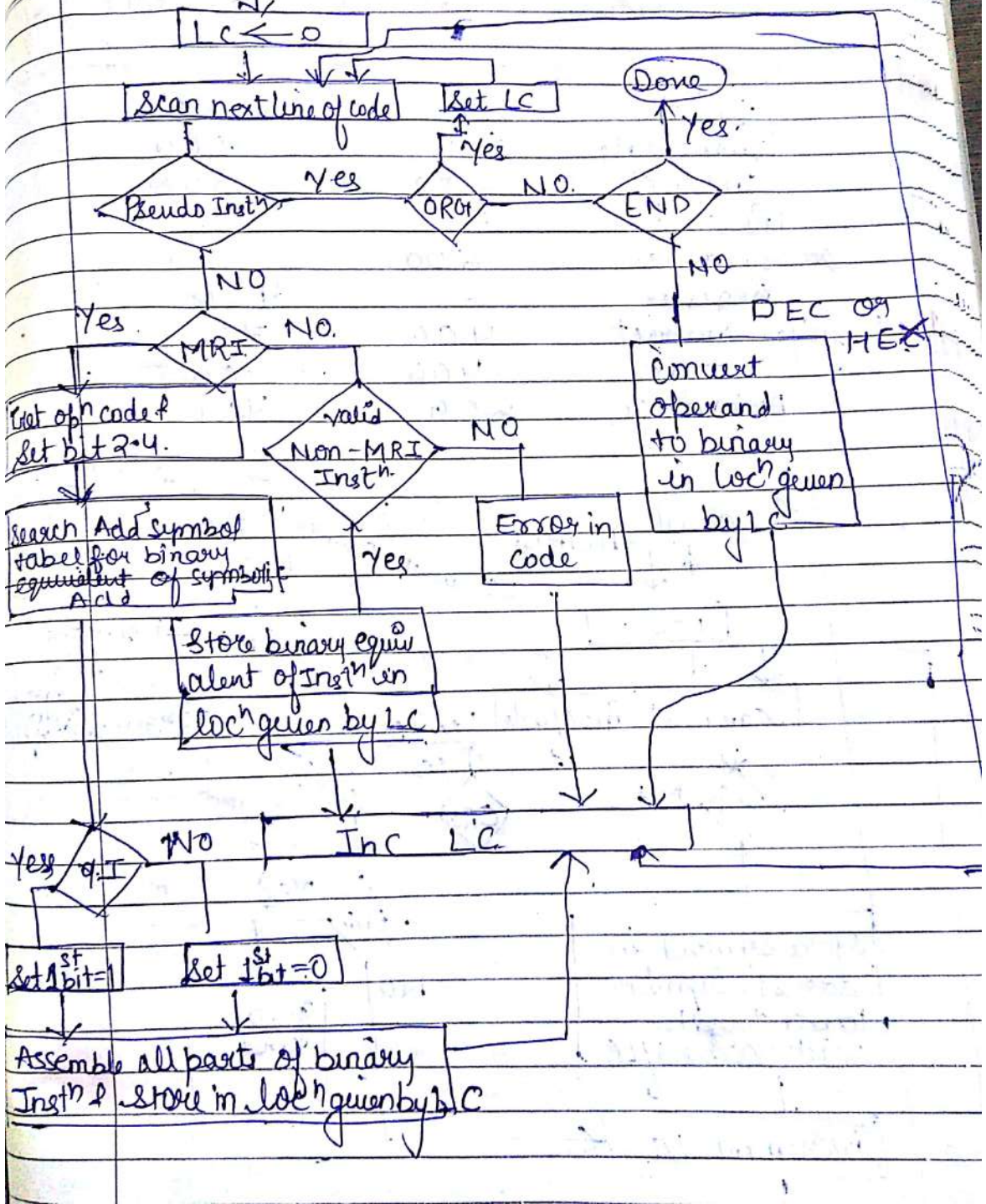
Label: Instn Comment
ABC: ADD B

Label :- User defined Symbol

~~Auto~~ Address symbol table

Page No. _____
Date _____

Second Pass



Scanned by CamScanner

LC → memory address of the binary which is stored at the Memory

Particular location of memory Reference Instn ← MRI Table

Register I/O Instn ← Non MRT Table

Pseudo Table

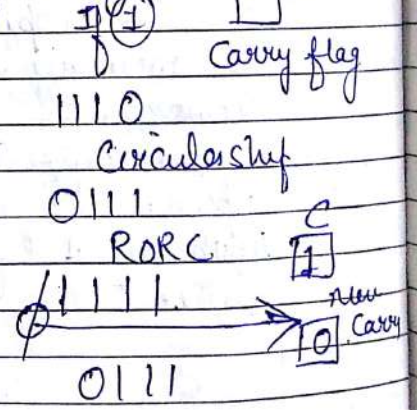
Address Symbol Table

Data Transfer & Manipulation

	Data Transfer	Data Manipulation		
		① Arithmetic	logical	③ shift Instn
LD	Load			
ST.	Store			
XCH	Exchange	INC	CLR	SHR
IN	In	DEC	COM	SHL
OUT	Out	ADD	AND	SHRA
PUSH	Push	SUB	OR	SHRL
POP.	Pop	MVL	XOR	ROR
		DIV.	CLR	ROL
			SETC (Set carry)	RORC
				RORL

RORC :- Rotated Right with carry.

RORL : Rotated left through carry



DEC
NEG
OR
CMP

Page No. _____
Date _____

Instruction Size

(1) One byte instⁿ
eg: MOV C, A
4 bit 4 bit
ADD, SUB, INR, DCR

(2) 2-byte Instⁿ

MVI A, Data
1st byte → 2nd byte

MVI, ADI, ANI, ORI

(3) 3-byte Instⁿ
JUM 2035 H
1st 3rd 2nd
LXI, CALL

(1) One byte instⁿ – Data transfer instⁿs that copy the content from one register or memory into another register or memory.
A one-byte instⁿ includes the operand & opcode in the same byte.
Arithmetic & logic instⁿ without ending with letter I are one-byte instⁿ.

Scanned by CamScanner

Page No. ____
Date ____

(2) 2-byte Instⁿ - In 2 byte instⁿ, the first byte specifies the opcode & the second byte specifies the operand.

Rules :-

(a) Instⁿ that load or manipulate 8-bit data directly are 2-byte Instⁿ.

(b) Instⁿ ending with letter I except LXI are 2-byte Instⁿ.

(3) 3-byte Instⁿ :- Instⁿ that load 16 bit or refer to memory address are 3-byte Instⁿ.

eg: LXI, JUM, CALL, Condⁿ jump.

In 3-byte instⁿ, the first byte specifies the opcode & the following 2 bytes specifies 16 bit address. 2nd byte is low order address & 3rd byte is high order address.

Scanned by CamScanner

Multiplication

Memory

Page No. ____
Date ____

```
MVI  A, 00H        ↓
MVI  B, 06H        Decimal        05
MVI  C, 03H        Accumulator    03
ABC: ADD, B;  00+06 ↓
     DCR C   06+06=12 Hexadecimal
     JNZ ABC.  (z=0)
     STA 0001
```

multiplicand

multiplier count

decrement C.

store the count of MC

If AC 0 or current zero then z flag is 1

0001 = 08 dividend

0002 = 03 divisor

Division

C - divisor

A - dividend

```
LDA  0002H; (divisor)
MOV  C, A;        C=03
LDA  0001H;       A=08
MVI  B, 00H;
L2: CMP. C.  — Compare with
    JC L1  when C = 0 (so AC Jump)
    SUB C. — Subtract C with A.
    INR B
    JMP L2.
```

08 03

if (A) < reg C=1 z=0

if (A) = reg z=1 C=0

if (A) > reg C & z cleared C & z = 0

A = 08 (A = 16 03)
C = 03

```
L1: STA 0003H (remainder)
    MOV A, B; → Hexadecimal (Register)
    STA 0004H; (quotient)
```

A = 05
C = 03

↓ decimal no (memory)

A = 02
C = 03.

JC L1 B = 01 + 0

```
MVI A, 00H
    B, 08H
    C, 03H
ABC: ADD B
     DCR C
     JNZ ABC
```

B

Output

Register	
A	0F
BC	05 00
DE	00 00
HL	00 00
PSW	00 00
PC	42 DF
SP	FF FF
Int-Reg	00

Flag	
S	0
Z	1
AC	0
P	1
C	0

2 | 15
2 | 7 - 1
2 | 3 - 1
1 - 1

0 0 0 0 1 1 1 1
0 F

STA 0002 — 15

STA 0003 0

STA 0004 3

08 0001 = 15
03 0002 = 05

C = 05
A = 15

B = 01
+01

B = 15 < 05
10 < 05
05 = 05

A 15
C 05
A = 10
C = 05
A = 05
C = 05
00000

Z = 1

Register	
A	03
BC	03 05
DE	00 00
HL	00 00
PSW	00 00
PC	42 1A
SP	FF FF
Int-Reg	00

Flag	
S	1
Z	0
AC	0
P	0
C	1

15
16
16
16 + 5
21

0000

2 & 8

Lab

Page No. ____
Date ____

Q. Calculating 1's Complent

```
LDA 0001;
CMA;
STA 0002;
ADI 01;
STA 0004;
```

Complement AC

Add to AC

0	0	0	
0	0	1	2
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Q. Find Largest element in array

Load Register Pair Immediate

```
LXI H, 0002;
MOV B, M;
INX H
MOV A, M;
DCR B;
LOOP: INX H;
      CMP M;
JC    JNC LOOP2
      MOV A, M;
LOOP2: DCR B
      JNZ LOOP
      STA 0009;
```

Register Pair

Increm by 1

C=0 Jump C=1

0	0	0	0	
0	0	0	1	
0	0	1	0	2
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	E
1	1	1	1	F

JNZ = Z = 0

JP – sim bit S=0

JM = Sim bit S=1

JZ = zero flag Z=1

```
00001101
11110010
   +  1
11110011
```

3

```
00000010
1111 1101   12 C
     +1
0000 1110   D
```

EF

Page No. ____
Date ____

Registers

A	FE	
BC	00	00
DE	00	00
HL	00	00
PSW	00	00
PC	42	0D
SP	FF	FE
IntReg	00	

Flags

S	1
Z	0
AC	0
P	0
C	0

0001

Complement AC

Add 10 AC

Load

S=0

2

L=3

A=2

Registers

H 20

L 34

MOA

1/12

2

8

4

6

2

MOA MOA

8 bit | 8 bit

16 bit

2034H.

B=5

A=2

H=8

2<8 C=1

B=4

S2

8

1

3

4

H=1 C=18

1<2

B=2

H=3 C=20 carry

3>2

A=3

A=3

1

B=2

A=3

H<4

A=3

3<4

$[(-13) - (-15)]$

Page No. _____
Date _____

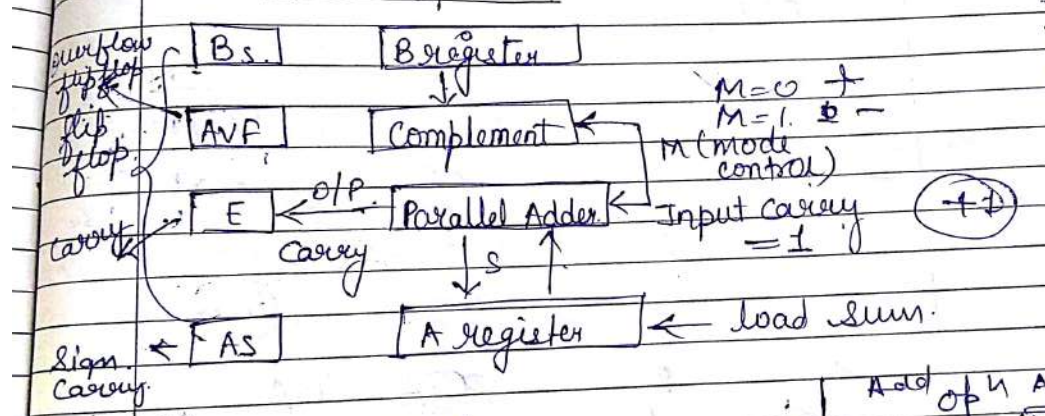
Unit III :

* -ve no representn

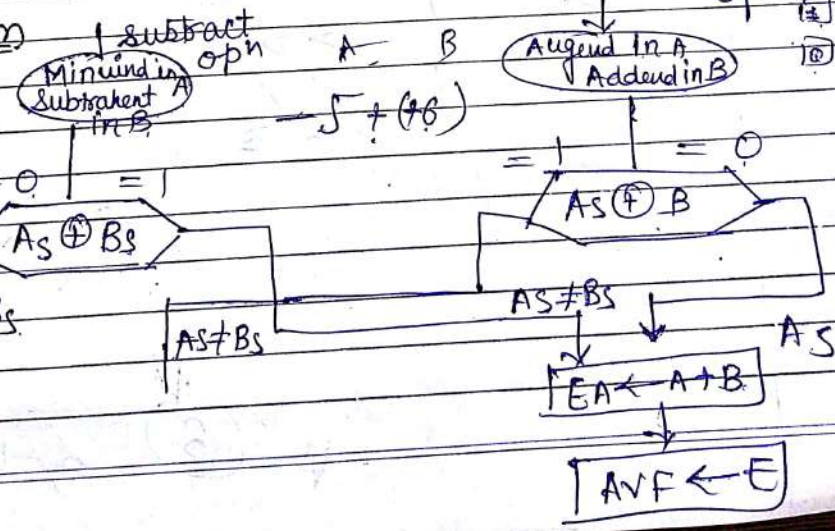
1. Signed Magnitude
2. 1's Complement
3. 2's Complement :

Addition & Subtraction using Signed Magnitude repn

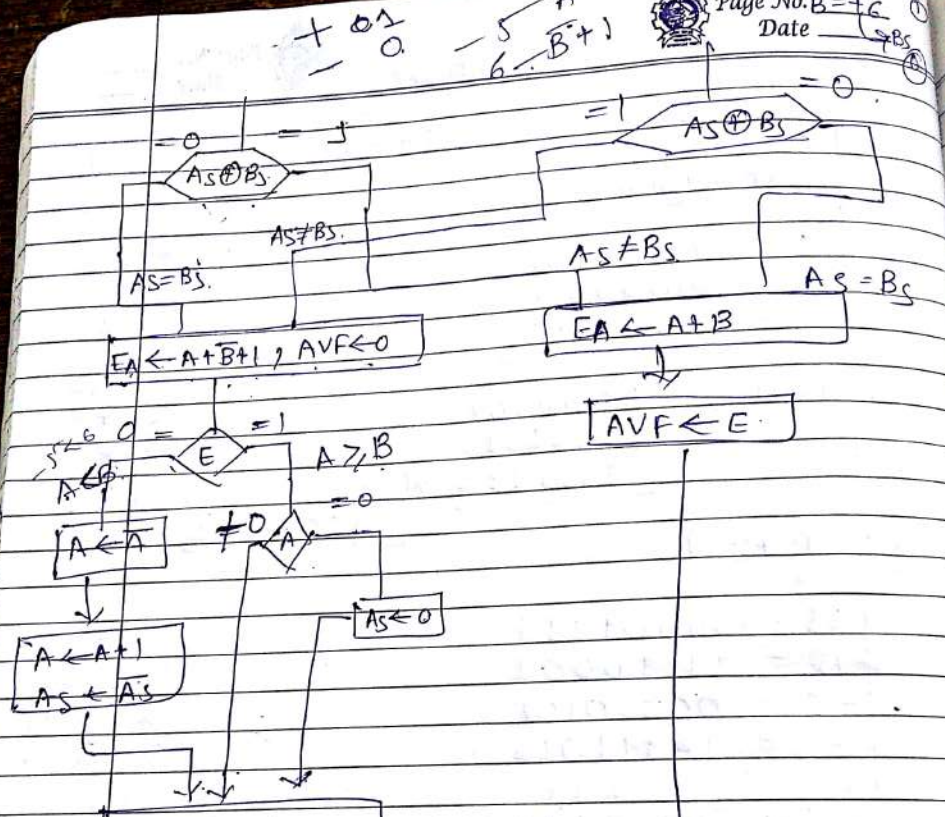
Hardware Implementation



Algorithm



Scanned by CamScanner



Q. $-6 - (-13)$

$A = -6$ $\boxed{As}$ 1

$B = -13$ $\boxed{Bs}$ 1

$A \oplus B = 0$

$6 = 00000110$

$13 = 00001101$

11110010

$+1$

11110011 $\bar{B}+1$

00000110 6

11111001

$00000110 - \bar{A}$

$+1$

$00000111 - \bar{A}+1$

0	00
1	10
1	10
0	01

Page No. _____
Date _____

Q. Add | Subtract

$+5+(-6)$ | $-6-(-13)$

$-15+(-5)$ | $+5-(-15)$

2's Complement

$$\begin{array}{r|l} 2 & 5-1 \\ 2 & 2-0 \\ & 1 \end{array}$$

101

(1) $5 = 00000101$

$6 = 00000110$

$(-6) = 11111010$

$$\begin{array}{r|l} 2 & 15-1 \\ 2 & 7-1 \\ 2 & 3-1 \\ & 1 \end{array}$$

$+5+(-6) = 00000101$

$\quad + 11111010$

$\quad \boxed{11111111} \rightarrow$ (-ve) 2's

$\boxed{00000001}$

(2) $-15+(-5)$

$$\begin{array}{r|l} 2 & 20-0 \\ 2 & 10-0 \\ 2 & 5-1 \\ 2 & 2-0 \\ & 1 \end{array}$$

$15 = 00001111$

$-15 = 11110001$

$+5 = 00000101$

$-5 = 11111011$

$-15+(-5) = 11110001$

$\quad 11111011$

$\quad 11101100 \rightarrow$ 2's

$\boxed{00010100}$

10100

(3) $-6-(-13)$

$$\begin{array}{r|l} 2 & 13-1 \\ 2 & 6-0 \\ 2 & 3-1 \\ & 1 \end{array}$$

$6 = 00000110$

$-6 = 11111010$

$13 = 00001101$

$-13 = 11110010$

$-(-13)$ 00001110

$-6 \quad 11111010$

$-6-(-13) = 00001101 = \boxed{00000111}$

100000111

Scanned by CamScanner

Page No.
Date

(4) $+5-(-15)$

$5 = 00000101$

$-15 = 11110001$

~~11110110~~

~~15 = 00001110~~

~~5 = 00000101~~

~~00010011~~

$5 = 00000101$

$-(-15) = 00001111$

00010100

2 | 20
10

Complement Signed Magnitude

(1) $+5+(-6)$

$A = 5 \quad A_S = 0 \quad A_S \oplus B_S = 1$

$B = 6 \quad B_S = 1$

(-6)

$A = 00000101$

$B = 00000110$

$\overline{B+1} = 11111010$

$A = 00000101$

11111111

$\overline{A} = 00000000$

$A+1 = \quad +1$

$A = \boxed{00000001}$

$A_S = 1$

$\boxed{A = -1}$

(2) $-15+(-5)$

$A = +15 \quad A_S = 1$

$B = +5 \quad B_S = 1$

$A_S \oplus B_S = 0$

$A \leftarrow A+B = 00000101$

(ये 5 नहीं है) 00000110

00001011

$AVF \leftarrow 0$

$\boxed{A = +20}$

$\boxed{A_S = 1}$

$A \leftarrow A+B \quad 00000101$

$5 = 00001111$

$A = 20 \quad 00010100$

$A_S = 1$ (-ve)

Scanned by CamScanner

Subtract

Page No. _____
Date _____

(3) $-6-(-13)$

$A = -6 \quad A_s = 1 \quad A_s \oplus B_s = 0$

$B = -13 \quad B_s = 1$

$\bar{B} = 11110010$

$\bar{B}+1 = \quad +1$

11110011

$A+\bar{B}+1 = 00000110$ $\qquad -6+13$

$A = 11111001$ $\qquad 7$

$\bar{A} = 00000110$

$\bar{A}+1 = \quad +1$

$A = 00000111$

$\boxed{A = 7}$

$\boxed{A_s = 0}$

(4) $+5-(-15)$

$A = 5 \quad A_s = 0 \quad A_s \oplus B_s = 1$

$B = 15 \quad B_s = 1$

$A = 00000101$

$B = 00001111$

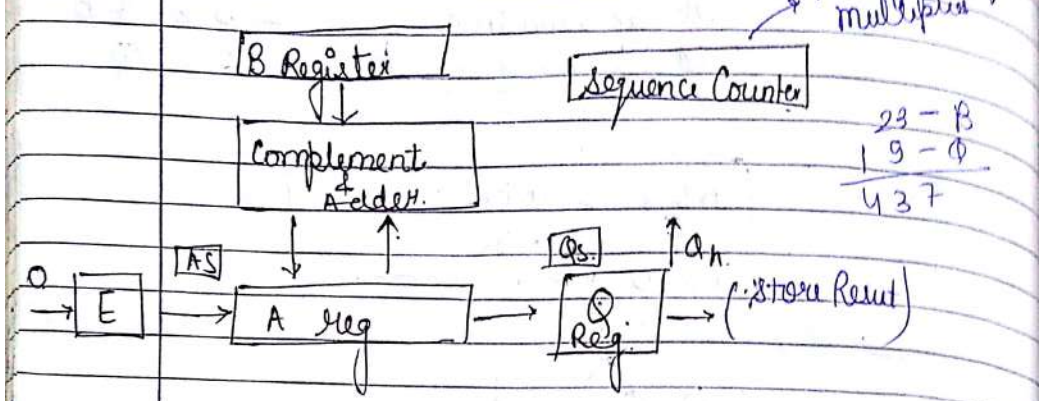
$A+B = 00010100$

$\boxed{A = 20 \quad A_s = 0}$

Scanned by CamScanner

Multiplication using Signed magnitude

Hardware



23 – B
19 – Q
437

Numerical eg.

B = 10111 (23)

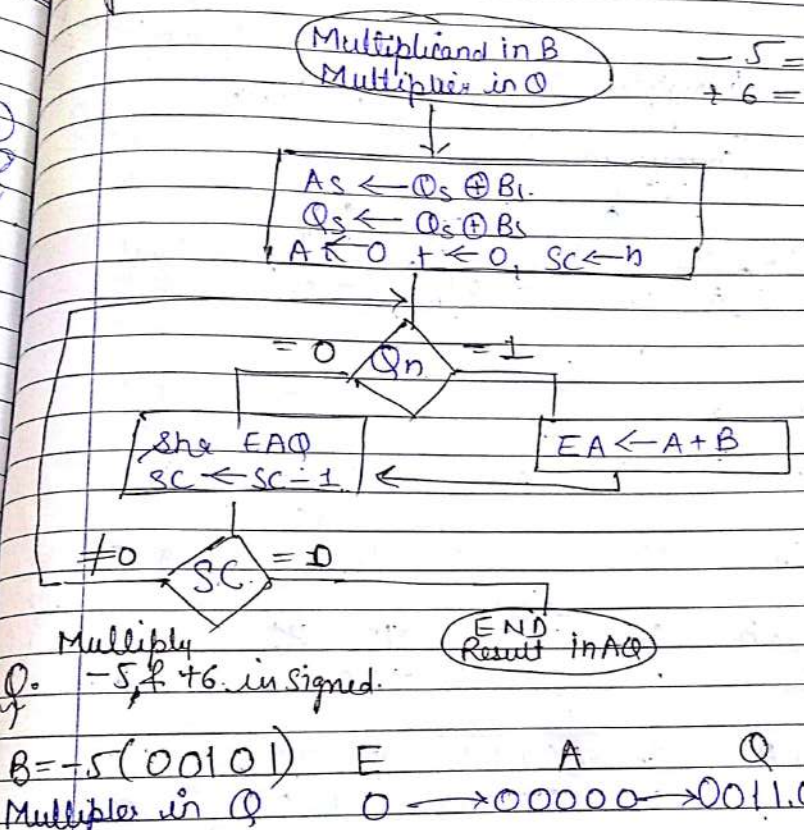
	E	A	Q	Sc
Multiplier in Q	0	00000	10011 (19) → Q_n	5
		10111		
$Q_n = 1$, Add B	0 →	10111 →	10011 → ×	
shr EAQ		01011	11001	4
		10111		
$Q_n = 1$, Add B	1 →	00010 →	11001 → ×	4
shr EAQ	0 →	10001 →	01100 → ×	3
$Q_n = 0$, shr EAQ	0 →	01000 →	10110 → ×	2
$Q_n = 0$	0	00100	01011	1
shr EAQ		10111		
$Q_n = 1$, Add B	0 →	10111 →	01011 → ×	0
shr EAQ		01101	10101	

Scanned by CamScanner

use always 5 bit

Page No. _____
Date _____

Algorithm



$-5 = B$
$+6 = Q$

Multiply
Q. -5 & $+6$ in Signed.

$-5 = B$
$6 = Q$

$B = -5 (00101)$
Multiplier in Q

	E	A	Q	SC
	0 →	00000 →	00110 →	5
$Q_n = 0$ Shr EAQ	0	00000	00011	4
$Q_n = 1$ Add, B	0 →	00101 →	00011 →	3
Shr EAQ	0 →	00010	10001 →	
		+ 00101		
$Q_n = 1$ Add, B Shr EAQ	0	00001	01000	2
Shr EAQ	0 →	00111 →	10001 →	2
	0 →	00011 →	11000 →	1
$Q_n = 0$ Shr EAQ	0 →	00001 →	11100 →	0
$Q_n = 0$ Shr EAQ	0	00000 →	11110	

Scanned by CamScanner

Page No.
Date

LAB

Ascending order

```
LXI H, 0005;
MOV C, M;
DCR C;
L1: MOV D, C;
    LXI H, 0006;
L2: MOV A, M;
    INX H;
    CMP M;
    JC L3
    MOV B, M;
    MOV M, A;
    DCX H;
    MOV M, B;
    INX H;
L3: DCR D;
    JNZ L2;
    DCR C;
    JNZ L1
```

0005 → 5 C=5
0006 → 3 C=4
0007 → 2
4 A=4 (3)
1 M=3 (1)
8 A=3 (0)
M=2
(JNC) 3<2 A<M ✗
3>2 A>M
C+2 result
C=0
JC Cas
C=3
C=2
D=2 B=2 M=3
A=2
M=5
2<5 M=2 M=3
C=1 M=3
D=1 K=1
D=0 C=1
A=0 C=0
D=1 C=3 D=2
C=2 C=1
C=2 A=2 M=5
D=1 2<5 C=1
D=2 D=0
A=5 D=1 K=1
A=2 M=1
A<Reg 5>1 C=0 Z=0
2<5 C=1
K=1 B=1 M=5
M=1 M=5
A=5
A=2 C=0 Z=0 A<M 5>1 K=0 C=0
2<4
B=1
B=1
```

Scanned by CamScanner
```

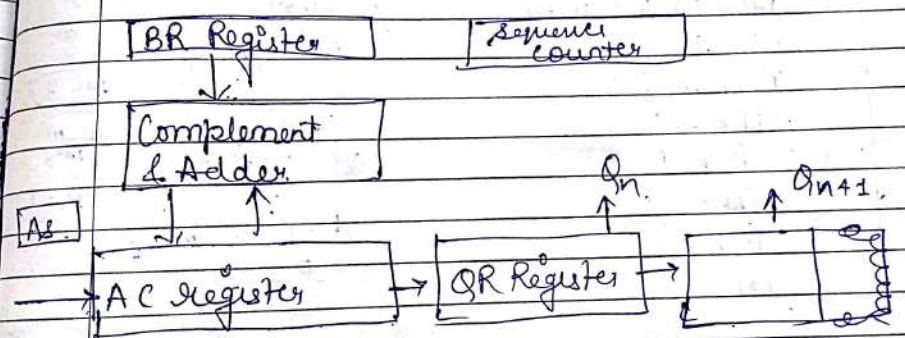
Page No. _____
Date _____

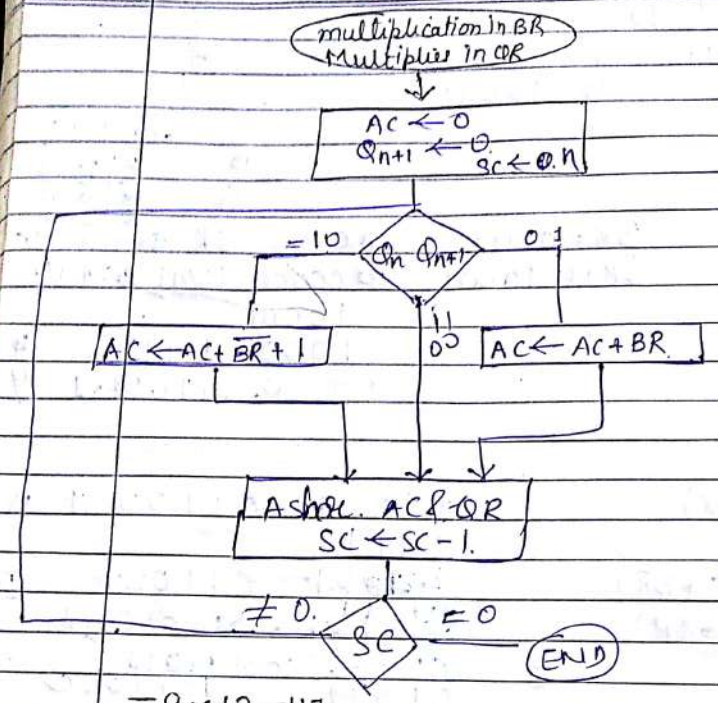
Register		
A	02	
BC	01	00
DE	00	00

2 | 13 - 01
2 | 6 - 0
2 | 3 - 1
| 1

0100
0100
1101
10.011

Booth Multiplication (2's Complement method)





$-9 \times -13 = 117$

BR = 10111
$\overline{BR}$+1 = 01001

Qn	Qn+1		BR = 10111 $\overline{BR}$+1 = 01001	AC	QR	Qn+1	SC
			Intial	00000	10011	0	5
1	0	Add $\overline{BR}$+1		01001			
				01001 →	10011 →		
		Ashr (AC & QR)		00100 →	11001 →	1	4
1	1	Ashr (AC & QR)		00010	01100	1	3
				+ 10111			
0	1	Add BR.		11001 →	01100 →		
				11100 →	10110 →	0	2
0	0	Ashr (AC & QR)		11110	01011	0	1
1	0	Add BR+1		00001			
		Ashr (AC & QR		00111 →	01011 →		0
				00011	10101	1	

(−13)

Page No.
Date

15 × (−13)

15 = 01111 13 = 01101
−13 = 10011

2	15 − 1
2	7 − 1
2	3 − 1
	1

BR = 01111
$\overline{BR}$+1 = 10001

	01111
−13	2 | 13 − 1
	2 | 6 − 0
	2 | 3 − 1

Q	Q_{n+1}	BR = 01111 $\overline{BR}$+1 = 10001	AC	QR	Q_{n+1}	SC
			00000	10011	0	5
1	0		10001			
Ashl			10001	10011		4
			11000	11001	1	4
1	1					
Ashl (Ac+QR)			11100	01100	1	3
			01111			
0	1	(AC+BR)	01011	01100		2
Ashl (Ac+QR)			00101	10110	0	
0	0		01011			
Ashl (Ac+QR)			00101	11011	0	1
			10001			
1	0		10011	11011		0
BR+1			10011			
Ashl (Ac+QR)			11001	11011	1	

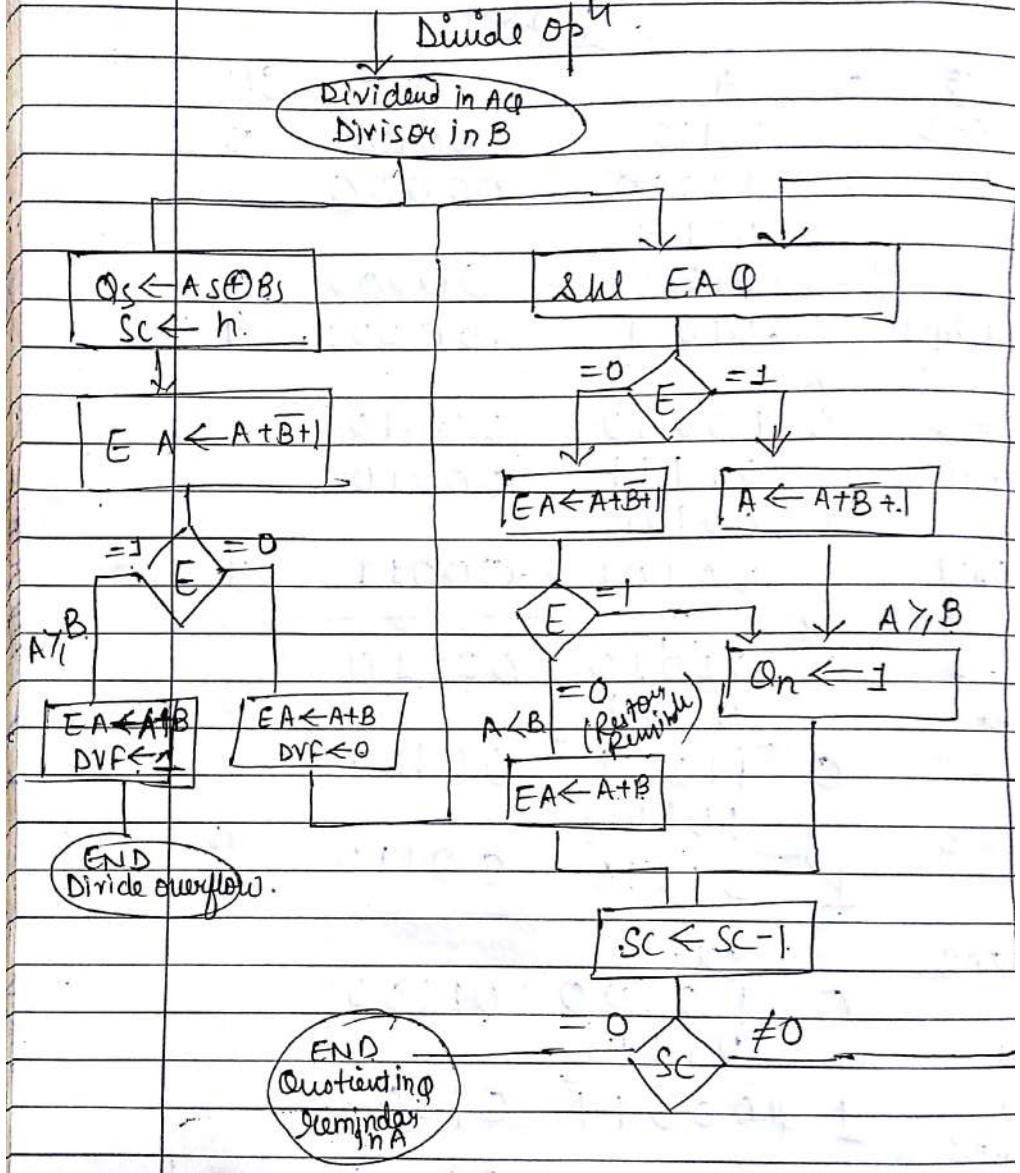
Ans

11100
01111
01011

00101 11011

11011 0

Division Signed Magnitude



Scanned by CamScanner

Divisor

$B = 10001$

$\bar{B}+1 = 01111$

$D1110 - A$ $\quad \bar{B}+1$

01111

11101

Page No.

Date

dividend

	E	A	Q	SC
Shl EAQ		01110	00000	5
	0	11100	00000	
Add $\bar{B}+1$		01111		
	1	01011	00000	
E=1, Set Qn=1		01011	00001	4
Shl EAQ	0	10110	00010	
Add $\bar{B}+1$		01111	00010	
	1	00101		
E=1, Qn=1		00101	00011	3
Shl EAQ	0	01010	00110	
Add $\bar{B}+1$		01111		
	0	11001	00110	
E=0, A+B		10001		
	1	01010	00110	2
Shl EAQ	0	10100	01100	
En=0, $\bar{B}+1$		01111		
	1	00011	01100	
E=1, Qn=1		00011	01101	1
Shl EAQ	0	00110	11010	
$\bar{B}+1$		01111		
	0	10101	11010	
E=0, Q A+B		10001		
	1	00110	11010	0

Scanned by CamScanner

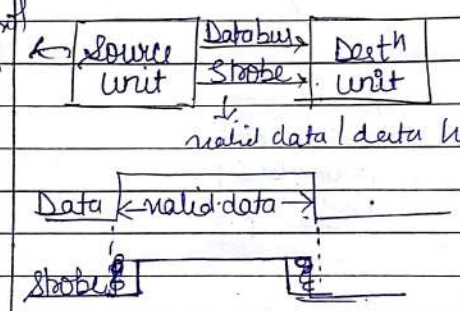
Unit :- IV

I/O Organisation

1. Input Output Interface
2. Isolated verus Memory mapped I/O
3. Synchronous & Asynchronous data transfer

Asynchronous Data Transfer

1. Strobe
2. (i) Source Initiated

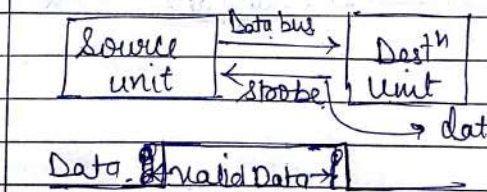


(this is the disadvantage of strobe)

valid data / data hold for a particular time only

Timing Diagram

(ii) Destination Initiated



(Acknowledgment is missing)

time diagram

Dy the I/O interface Required

Electromagnetic — Electronic

Slow — Fast

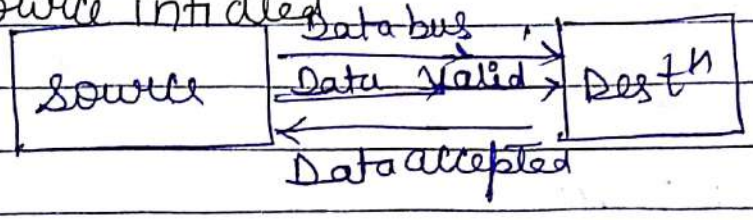
Formate diff^n (I/O) — Formate diff (CPU)

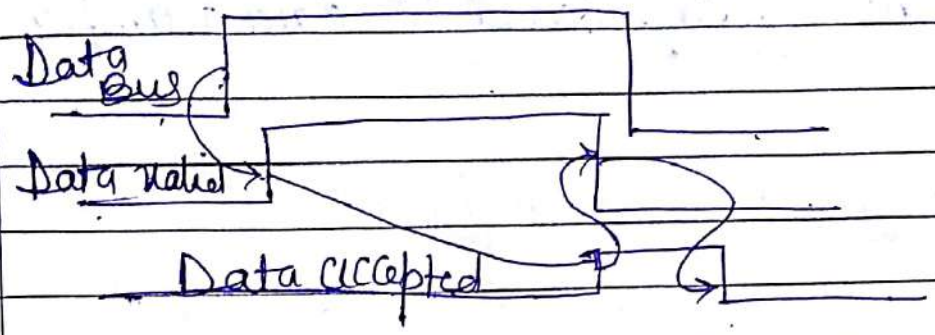
Page No. _____

Date _____

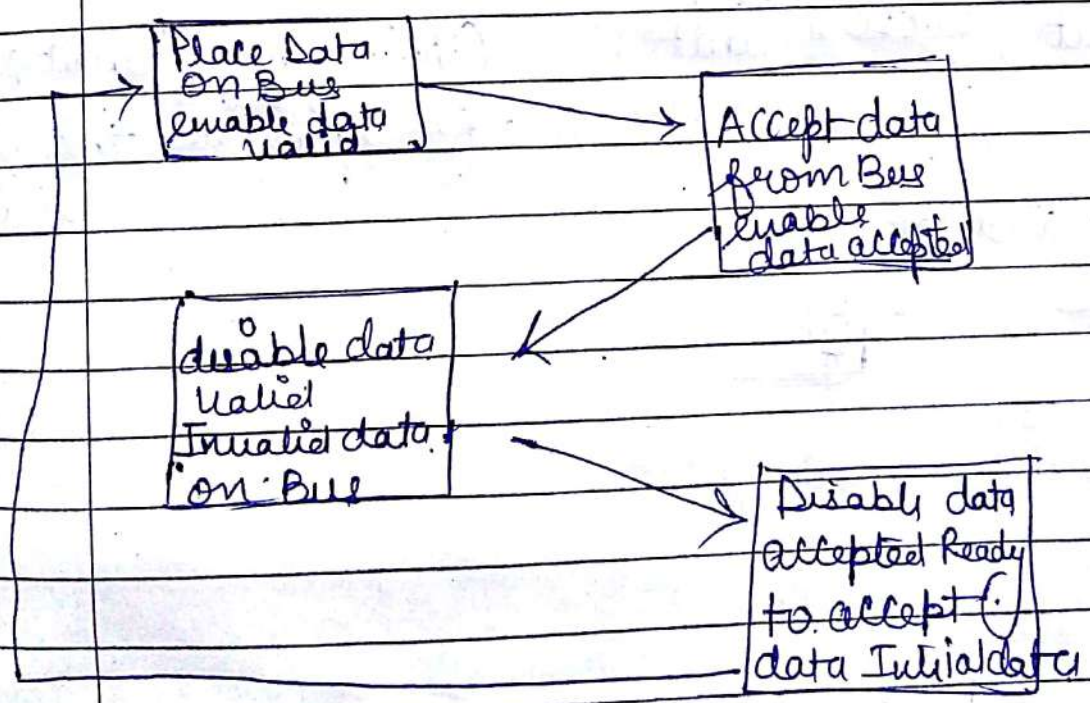
(2) Handshaking Method

(i) source initiated





(ii) Destination Initiated Sequence of events



CPU → Memory

CPU → I/O

Isolated I/O

① bus for diff^n Signals

② bus common but Signal diff^n

③ when both Signal & bus are same (by giving Seperate memory)

M I/O

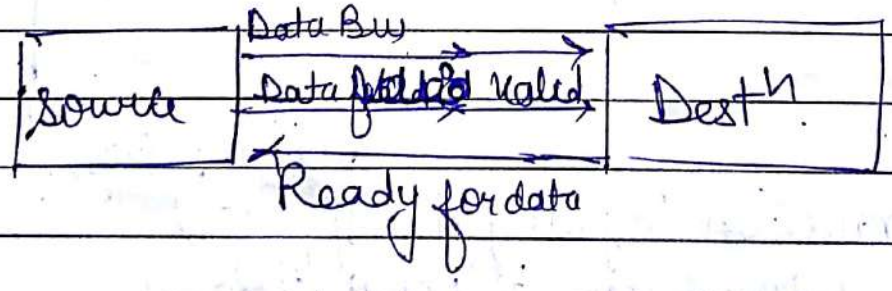
Memory Mapped I/O

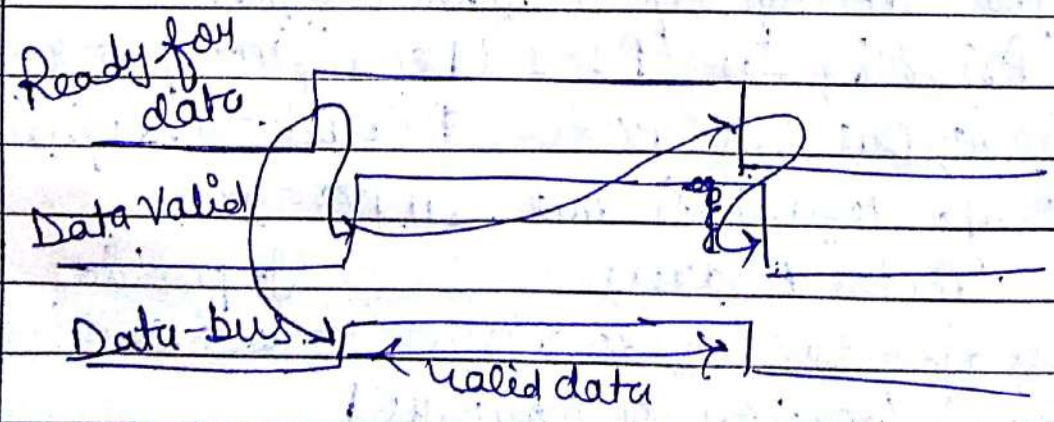
Asynchrony → Stable synchry bet^n two device have diff clck of each which is achieve by Control Signal.

① Strobe

② Handshaking

(ii) Destination Initiated





Scanned by CamScanner

(Return address)

Page No. _____
Date _____

Modes of Data Transfers (Central Computer to I/O)

→ Programmed I/O (Instruction used)
① lenthy
② time taking
③ unnecessary busy CPU.

S/W → Polling

H/W (To decide Priority) what device is taken

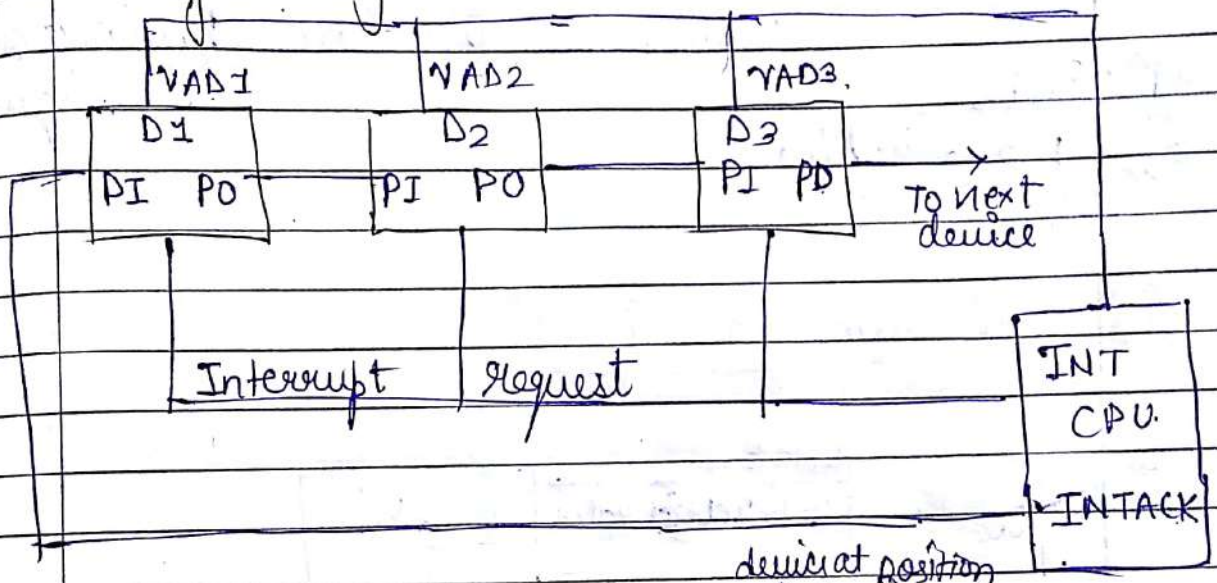
→ Interrupted Initiated I/O
① Two task at a time.

Daisy chain priority

→ DMA (Direct Memory access)

flag bit [1]
[0]

Daisy chaining



→ Serial Connection (Highest priority device at position first)

→ Two signal Interrup request / Acknowledgment

PI = Priority In. PI=1 (Pending interput request)

PO = Priority Out set PO=0 (block the signal)

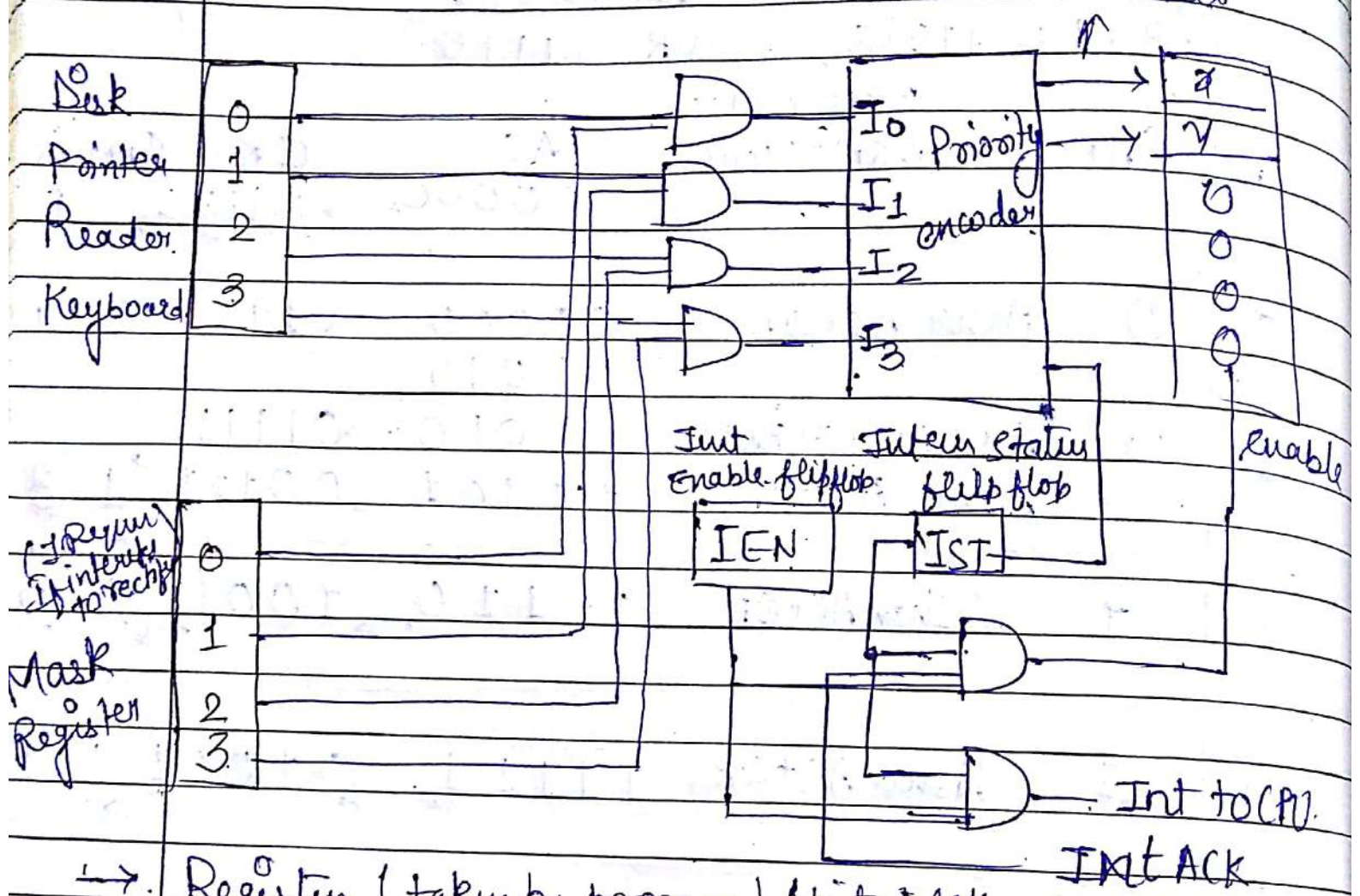
If Interrup request is not so PO=1

VAD = Vector Address

If source produce inter so it provide a branch Addr (Address location of Routine)

(Non Vectored Address) when source not provide branch address (And CPU transfer to the fixed or any location) (By default get the address)

Parallel Priority



→ Register (taken by program) (bits tell the about the priority)

Mask Register → Control the Interrupt signal

IST - If Non Masked bit set

IEN - (If IEN=0 this means Interput is rejected)
(Set by the user)
that interput ko cue deya nahi

Page No.
Date

Q1 Multiply 6 × (-2) using booth

2 = 00010

2 | 6 - 0
2 | 3 - 1
1

BR = 6 = 00110 QR = 00010
BR+1 = 11010 QR = 11110

BR = 00110
BR+1 = 11010

Q_n	Q_{n+1}		AC	QR	Q_{n+1}	SC
			00000 →	11110 →		5
0	0	Ashr AC&QR	00000	01111	0	4
			11010			
1	0	Add AC with BR+1	11010 →	01111 →		
		Ashr AC&QR	11101 →	00111 →	1	3
1	1	Ashr AC&QR	11110 →	10011 →	1	2
1	1	Ashr AC&QR	11111 →	01001 →	1	1
1	1	Ash AC&QR	11111	10100	1	0

= -12

Q2 Multiply -15 × (+4) using signed magnitude

B = +15 (-15) E A Q = (4) SC $\boxed{A_s = Q_s = 1}$

B = 01111 0 → 00000 → 00100 → 5

$Q_n = 0$
Shr EAQ 00000 00010 4

$Q_n = 0$
Shr EAQ 00000 00001 3

01111

$Q_n = 1$
Add A+B 0 → 01111 → 00001 →

Shr EAQ 00111 10000 2

Page No. ____
Date ____

	E	A	Q.	S.C.
B = 01111				
Qn = 0	0 →	00111 →	10000 →	0
Shr EAQ	0 →	00011 →	11000	1

$$\begin{array}{r|l} 2 & 12-0 \\ 2 & 6-0 \\ 2 & 3-1 \\ & 1 \end{array}$$

Qn = 0		
Shr EAQ	00001 11100	0

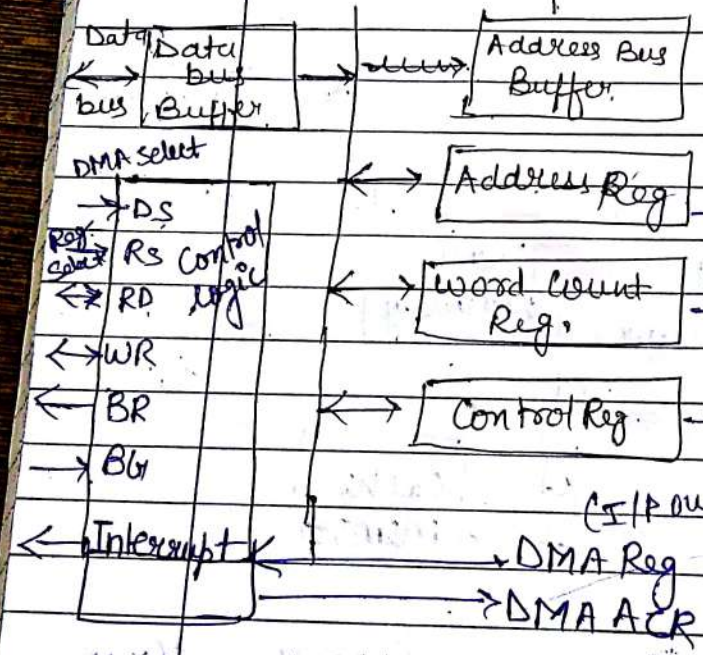
$$\begin{array}{r|l} & 00000\ 01100 \\ & 11111\ 10100 \end{array}$$

$A_S = 1$ AQ = 60

$$\begin{array}{r} 15.2 \\ 4. \\ \hline 6.0 \end{array}$$

$$\begin{array}{r|l} 2 & 60-0 \\ 2 & 30-0 \\ 2 & 15-1 \\ 2 & 7-1 \\ & 3-1 \\ & 1 \end{array}$$

DMA (Direct Memory Access)



00.001 11100

CPU is removed from the data transfer DMA controller do the work of data transfer

BG = 1

All the buss are handover to DMA

(I/P out Request sent) DMA Intilization

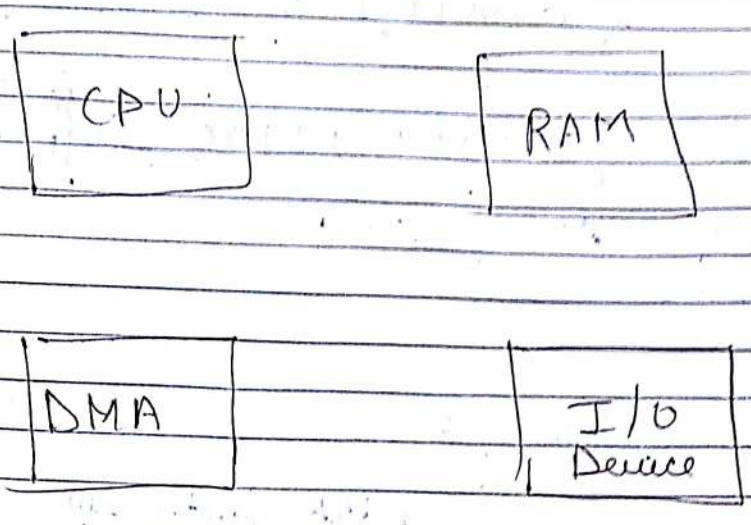
1. Mode of operation
2. Address of Data
3. How many words require
4. Control transfer to DMA

BG = 0 Buss B CPU ke pass h.

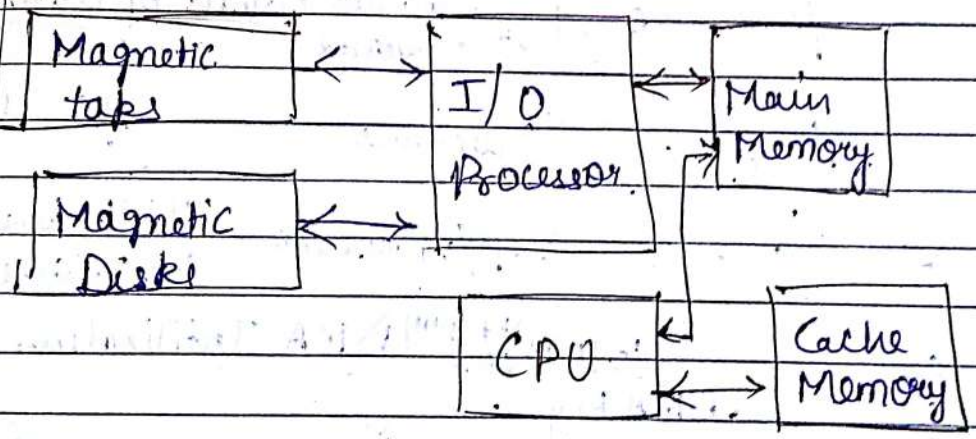
BR :- Bus Request
BG :- Bus Grant
DS – DMA Select
RS – Reg. Select

Read + Write Double sides buy ~~because both Bus~~ DMA use the R/W buy

DMA Transfer

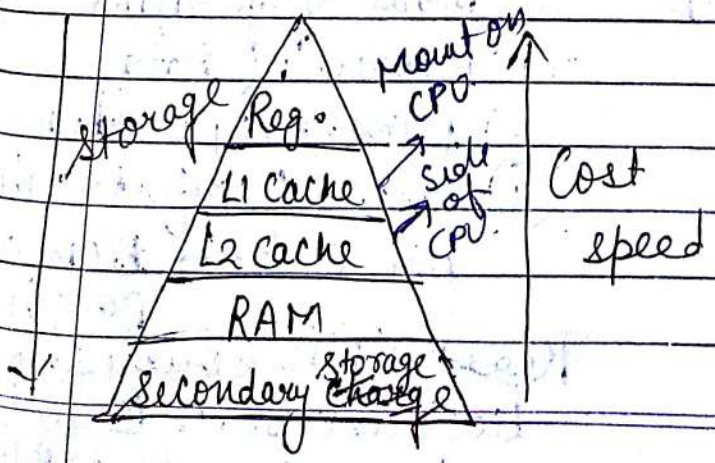


Memory Hierarchy



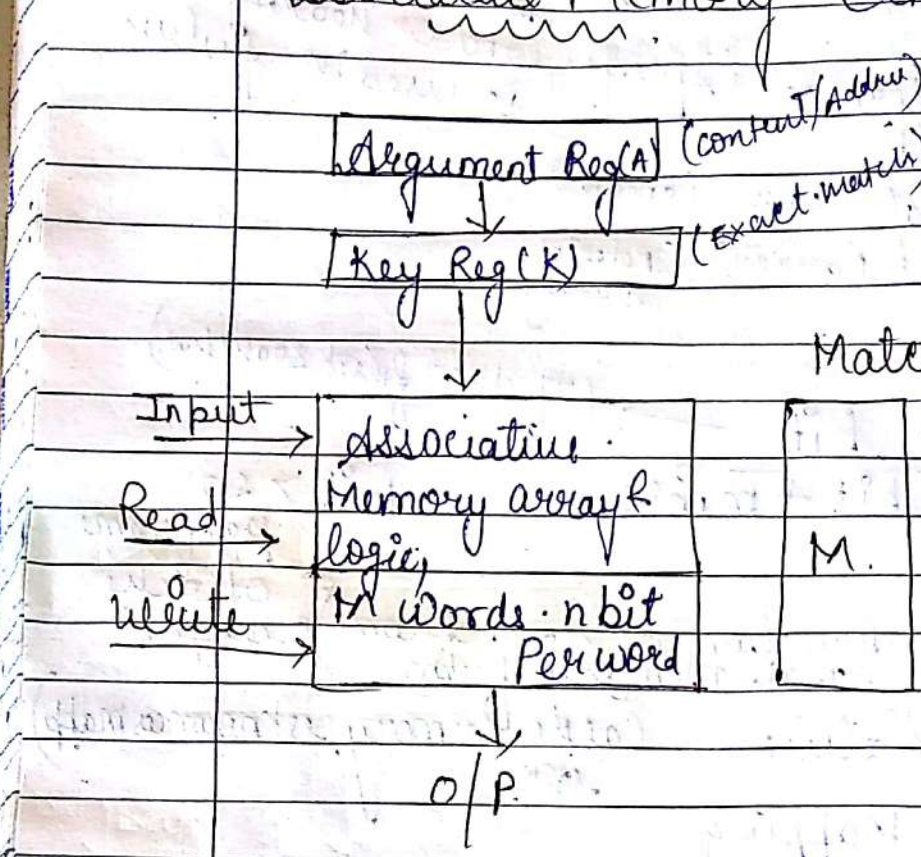
capacitor → Dynamic

FF → Static



Secondary Memo
↓
auxillory memory
(By yourself)

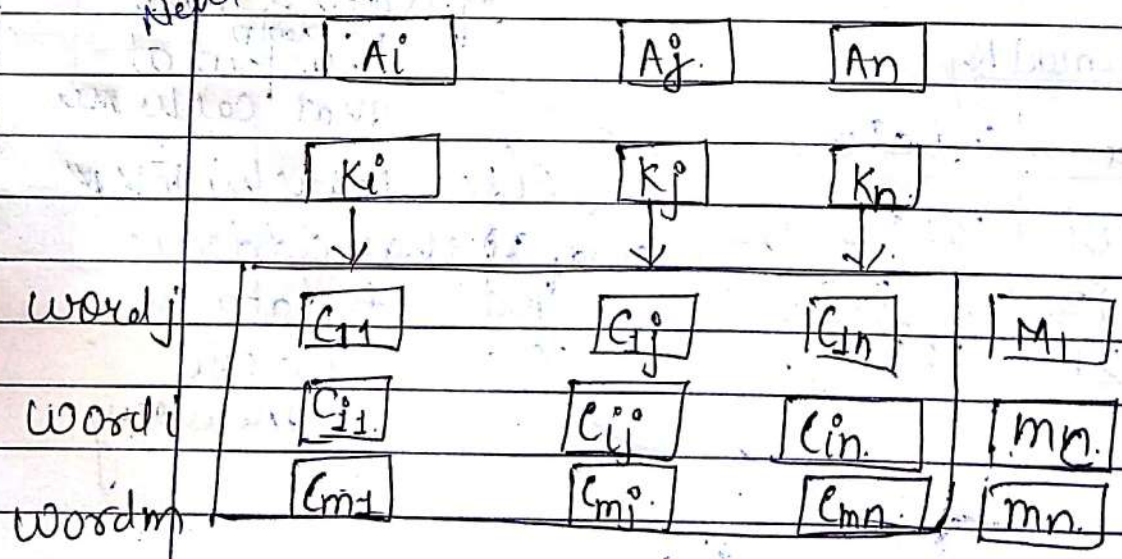
Associative Memory (CAM) Contented Associative Memory



We provide the Content in place of Address

- Search time Short

eg:- A 1001 111100
K 111 000000 → Talraja
Neha ← 100 111100 ✗ wrong dat
101 001001 ✓ correct
Neha



It is very costly because every cell needs a Comparision Logic

Page No. _____
Date _____

Cache Memory

(cache memory based in this principle)

→ Locality of Reference (specific data or nearby data is used in future)

- → temporal (45K) (smart)
- → Spatial (array) (Related nearby)

Hit Ratio = $\frac{hit}{hit + miss}$ → (If data found search that)

hit > so performance of cache is good

Mapping (Transformation of data from main memory from Cache memory is known as mapping)

- → Associative
- → Direct Mapping
- → Set associative

MM 32K×12 Cache = 512×12

(1) Associative [Associative memory used]

↓ 15 bit address

Argument Reg

Add	data
01000	3259
01012	5250
10101	1250

If data found then put data at that cache mem

else searching in MM and put that address & data in Cache memory

cache memory

Octal address

Header Octal

Page No.
Date

To derive Cache memory = RAM is used

(2) Direct Mapping

→ K bit Index (cache size)

→ n-K bit tag

6 bit 9 bit

tag	Index

octal address

00 000 — 77 777

MM	
32K×12	000
Add = 15	
data = 12	777

Cache	
512×12	000
Add = 9	
data = 12	777

00000	1220
00777	2340
01000	1210
02000	4560
02777	6710

MM

Tag	block	word

Index

Index	tag	data
000	~~00~~ 02	1220
	3 + 3 = 6	
777	02	6710

Block	000	01	3110
Block	007	01	1280

$32K = 2^{15}$

Drawback

So index is same & tag is different for

00000 | 02000 (2 way Associative)

$5/2 = 2^{9}$

Cache mem = Index

It is rectify by making two tag

02000 → Index

Tag

5 bit Address → tag + Index

Index | Tag | data | Tag | data

1024

64 block

each block
8 ~~12 bits~~

6 2^8 8 2^3 3

Tag	block	

Index
8+3

Cache

Tag	data

3x4

Scanned by CamScanner