Date: 10/10/17

DC (Module-2)

Data Path & Design of ALU.

① Introduction of ALU & Data path:—

- Def^n: ALU = AU + LU + SU
- Def^n: Data path = CPU registers + ALU.
- Concept: Von Neumann Architecture

CPU → Data Processing Unit + Program Control Unit (CU)
or
Data Path.

- Data path proposed:

Data path = ALU + a collection of Registers. (ACC + MQ + DR or MBR)

② Micro-operation & its classification:

- Def^n: An operation which is performed at micro-level or register level or IC level is said to be micro-operation.
- Classification:—
  a) Register Transfer.
  b) Arithmetic.
  c) Logic.
  d) Shift.

Scanned by CamScanner

③ RTL & its classification:-

• Def^n: It help us to represent micro-operation.

a) Register Transfer Micro-operation:-

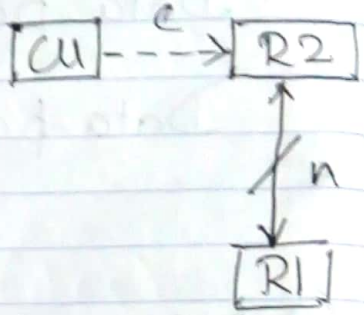
Let R_1 & R_2 be two registers. Let R_1 be the source register & R_2 be the destination register. Now, we want to transfer the value from R1 to R2.

$\boxed{C : R_2 \leftarrow R_1}$

C → Control Signal
R_2 → Destination Register
R_1 → Source.

If C=1 then $R_2 \leftarrow R_1$

else no operation



b) Arithmetic micro-operation

① $C : R_1 \leftarrow R_2 + R_3$	Addition without car
② $C : R1 \leftarrow R_2 + R3 + 1$	" with "
③ $C : R_1 \leftarrow R2 + R3 + 1$	Subtraction without b
④ $C : R1 \leftarrow R_2 + \overline{R3}$	" with borr
⑤ $C : R1 \leftarrow R1 + 1$	Increment
⑥ $C : R1 \leftarrow R1 - 1$	Decrement

Scanned by CamScanner

@ Orunayan Bhattachary @

c) Logical Micro-operation.

1	$C: R1 \leftarrow R1 \vee R3$	OR
2	$C: R1 \leftarrow R2 \wedge R3$	AND
3	$C: R1 \leftarrow \overline{R_2}$	NOT
4	$C: R1 \leftarrow \overline{R_2 \vee R3}$	NOR
5	$C: R1 \leftarrow \overline{R2 \wedge R3}$	NAND
6	$C: R_1 \leftarrow R_2 \oplus R3$	XOR
7	$C: R_1 \leftarrow R2 \odot R3$.	XNOR.

d) Shift Micro-operation

1	C: R1 ← shl R2	logical left Shift.
2	C: R1 ← shr R2	" right "
3	C: R1 ← ashl R2	Arithmetic left "
4	C: R1 ← ashr R2	" right "
5	C: R1 ← cil R2	Circular left "
6	C: R1 ← cir R2	" right "

Scanned by CamScanner

Page :
Date : 17/10/17

DC

④ CPU REGISTERS & ITS CLASSIFICATION :

- Def^n : • It is a part of both CPU & Memory
  - CPU → ALU + CU + CPU registers.
  - Memory → CPU registers + Cache + Primary Memory + Secondary Memory.
  - CPU registers at top of Memory Hirarcy.

- Property :-

a) Speed → Highest
b) Cost → Highest / Costliest
c) Capacity → Lowest
d) Bandwith → Highest.

- Classification :-

a) Accumulator (Mandatory). (ACC).

- Task → Before any arithmetic / Logical / Shift operation, one of the input may be stored in it
  → After any arithmetic / Logical / Shift operation, output or a portion of the output may be stored in Accumulator

- Example :- (Assembly level language statement)

① CMA → [A] ← [$\overline{A}$] (Logical operation)

② ADI07 → [A] ← 07 + [A] (Arithmetic ")

③ SHL A → (Similar to a<<1 in C).

→ [A] ← [A] << 1. (Shift Opr)

Scanned by CamScanner

(b) GPR (General Purpose Register).

- It helps to perform any Arithmetic / Logical / Shift operation.
- Ex: In 8085 MP → B, C, D, E, H, L.
- Ex: ADD B, C → [B] ← [B] + [C].
- SUB C, D, E → [C] ← [D] − [E].

(c) SPR : (Special Purpose Register).

→ NOB. of SPR, Operation, task differs from processor to processor & architecture to architecture.

i) PC → Program Counter.
→ Holds the address of next instruction opcode.

ii) MAR → Memory Address Register
→ Holds the Address of current instruction opcode.

iii) IR → Instruction Register
→ Holds the Opcode of current instruction.

iv) SP → Stack Pointer.
→ Holds the Address of the topmost element of the stack.

v) PSW / Status Register / Flag Register.
~~ss~~
→ Program Status word.
→ After any operation perform by ALU, PSW indicate the status of it.

Scanned by CamScanner

PSW / Status Reg. / Flag. Parts.

→

(i) CS → Carry Status (for carry bit, then 1).

(ii) AC → Auxiliary Carry.

(iii) Z → Zero Flag (If Subtraction is 0 the Op store 1)

(iv) P → Parity (Even no. of 1 then 1 or else if odd parity set to 0).

(v) S → SIGN (Negative then 1 else 0).

(vi) MBR/DR → Memory Buffer Register or Data Register

(Holds temporarily & deals with the data going out or in the CPU).

(vii) IBR → Instruction Buffer Register.

[~~SPR~~ SPR group register does not directly related to the Operation.

Scanned by CamScanner

@ Onneyan Bhattacharya @

DC :—

→ • Construction of a Common Bus System:

1) Using MUX(s)

2) Using tristate buffer.

① Using MUX :-

Prob:— Design a n-bit common bus using MUX(s) to transfer all the bits of a Register given ∃ 4 register & each Register contains n-bits.

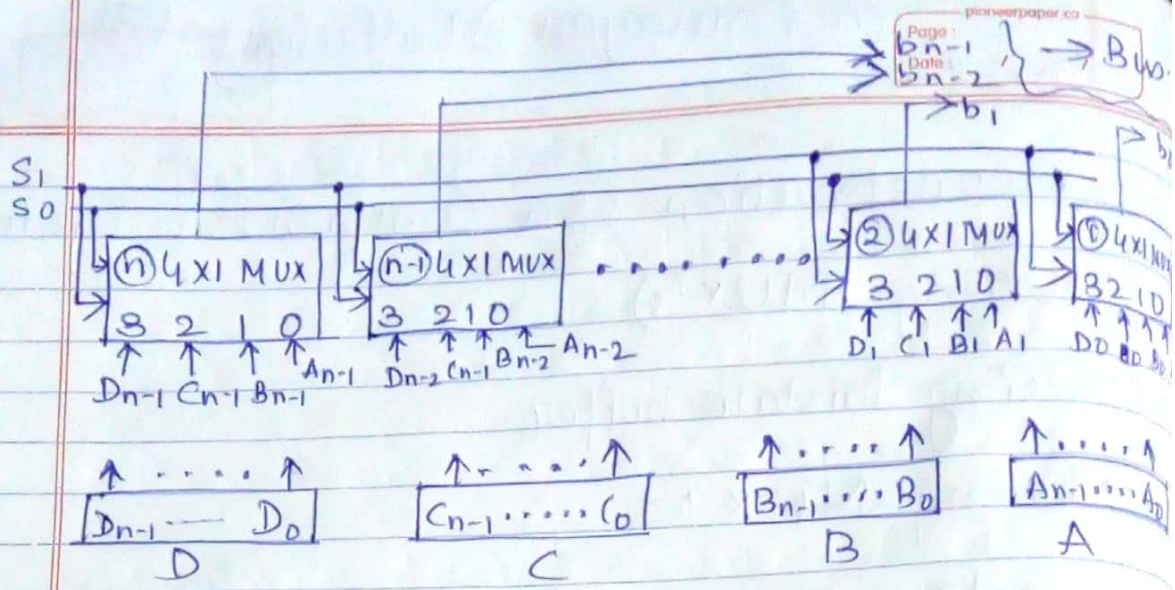
⇒ • no. of lines in the Bus: n.
• no. of register : 4(k), A, B, C, D.
• no. of bits in 1 reg is: n
(0, 1, 2, ..., n-1)
• no. of I/p(s) of 1 mux :- 4 (no. of register)
• no. of select line of 1 MUX: 2 ($\log_2 K$).

• no. of output for 1 MUX :- 1
• no. of MUX(s) : 1 * n = n

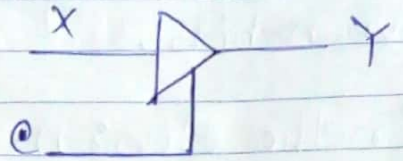
TRUTH TABLE :—

S_1	S_0	O/P
0	0	A
0	1	B
1	0	C
1	1	D.

Scanned by CamScanner



(2) Using Tri-State Buffer: (T).



If $C=1$, then $Y=X$.
else high impedence State.

Prob:-

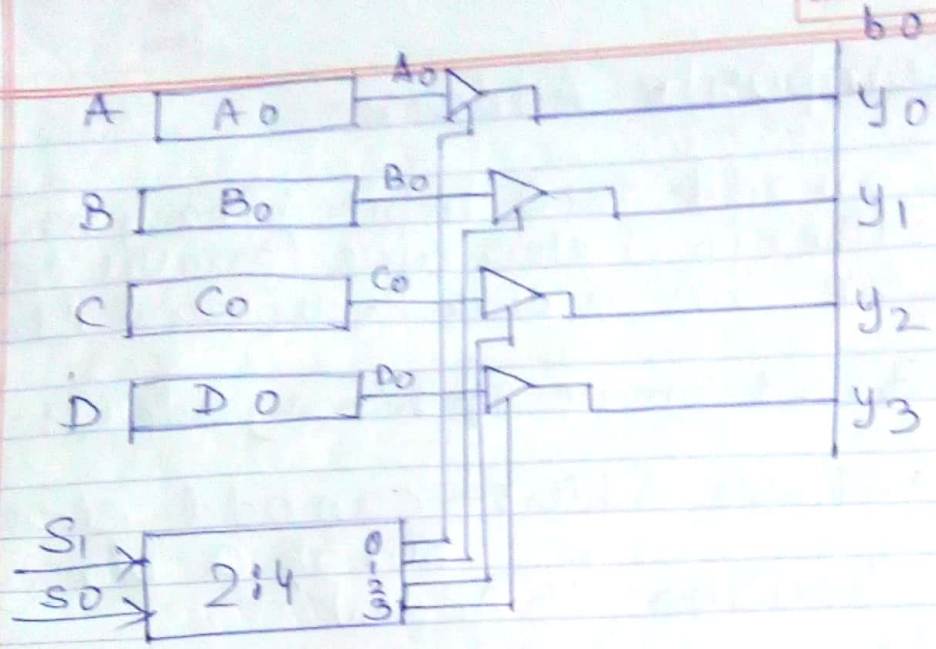
Design a 1-bit common bus using tri state Buffer to transfer 1bit of a register out of 4.

⇒ no. of lines in the Bus = 1
no. of reg(s): 4(k), (A, B, C, D).
no. of bit in 1 reg: 1.

no. of decoder needed = 1.
no. of output of decoder = 4.
no. of input of decoder: $\log_2 k = 2$

No. of T: no. of register x bit
$= 4 \times 1$
$= 4$.

Scanned by CamScanner



TRUTH TABLE:-

S_1	S_0	O/P
0	0	$Y_0 \leftarrow A_0$
0	1	$Y_1 \leftarrow B_0$
1	0	$Y_2 \leftarrow C_0$
1	1	$Y_3 \leftarrow D_0$

2) Introduction of Binary Adder:-

Binary Adder can perform either addition b/w two n-bit binary no.s or addition b/w on n bits binary digit

A = 1101
B = 1000
1 0101
C S

(computer Adder).

- Based on no. of clock cycle needed

Page :
Date : / /

Composite Adder :-

Can add two n-bit binary no.s. either in 1 clock cycle (parallel Adder) or in n clock cycle (serial Adder)

- Based on no. of binary added :-
  - Simple Adder :- can add either 2 bits of a binary bit position (1/2 Adder)

  or 3 bit of a Binary bit position (Full Adder)

Serial Adder	Parallel Adder
1 Composite Adder	1 Composite Adder.
2 Can add two-n bit binary no.s in 'n' clock cycle.	2 Can add it in 1 clock cycle.
3 (Combinational + Sequential) Circuit.	3 Combinational Circuit.
4 Using 1 FA + 1 D Flipflop	4 n no. of ~~D Flip~~ Full Adder.
5 Slow	5 Fast.
6 Inexpensive	6 Expensive.
7 Simple to Design	7 Complex to Design
6 Don't may face carry propagation Delay (CPD)	6 may face CPD.

Scanned by CamScanner

@ Onnayan @ CSR-2C / 66

Page :
Date : / /

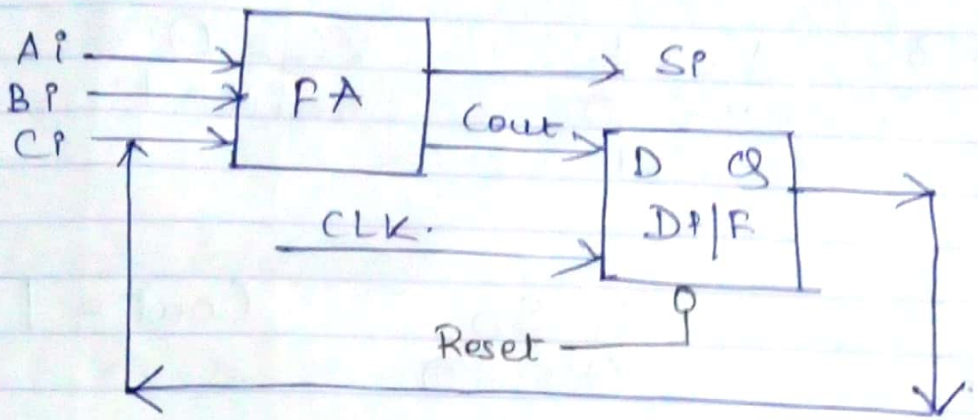
(a) No Sub Category

(a) (a) CPA
(b) CLA
(c) CSA.

Scanned by CamScanner

DC.

1 (*) Design a serial adder circuit to add a ~~b~~ 4-bit number.



D F/F → Store Data (Objective).

$A = A_3\ A_2\ A_1\ A_0$
$= 1\ 1\ 1\ 0$

$B = B_3\ B_2\ B_1\ B_0.$
$0\ 1\ 1\ 0.$

3	2	1	0
1	1	0	0
1	1	1	0
0	1	1	0
(1) 0	1	0	0
C_4	S_3	S_2	S_1	S_0.

(common example) (Q.1, Q.2)

(1) $i = 0$ $A_0 = 0$ $B_0 = 0$ $C_0 = 0$ $S_0 = 0$ $C_{out} = 0 = D.$

(2) $i = 1$ $A_1 = 1$ $B_1 = 1$ $C_1 = 0.$ $S_1 = 0.$ $C_{out} = 1 = D.$

Scanned by CamScanner

Date : / /

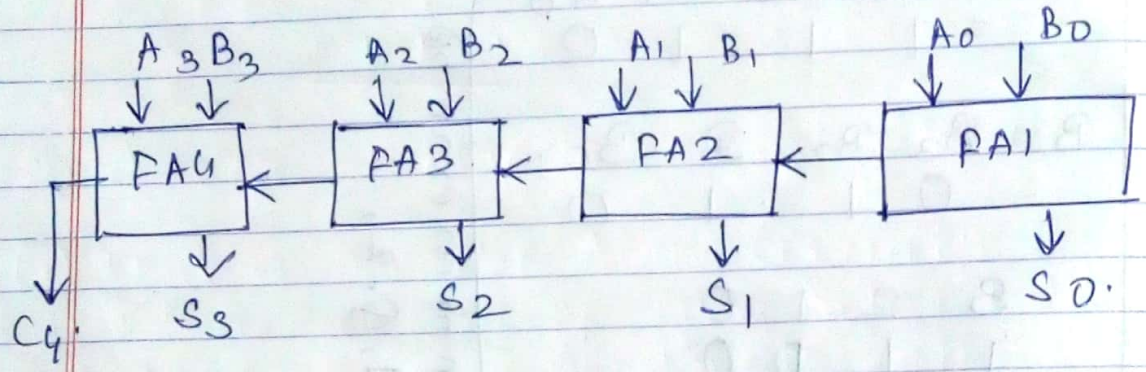
(3) $i = 2$ $A_2 = 1$, $B_2 = 1$, $C_2 = 1$ $S_2 = 1$, $C_{out} = 1 = D$.

(4) $i = 3$ $A_3 = 1$, B_3 0, $C_3 = 1$ $S_3 = 0$. $C_{out} = 1$.

$C_{out} = 1$.

$S = S_3\ S_2\ S_1\ S_0$
$= 0\ 1\ 0\ 0$

2. (*) Design a 4-bit carry propagator adder. (or ripple carry Adder) → (Expensive + less Time Req. to Execu



FA1: $A_0 = 0$, $B_0 = 0$, $C_0 = 0$ } $S_0 = 0$, $C_1 = 0$.

FA2: $A_1 = 1$, $B_1 = 1$, $C_1 = 0$ } $S_1 = 0$. $C_2 = 1$.

FA3: $A_2 = 1$, $B_2 = 1$, $C_2 = 1$ } $S_2 = 1$. $C_3 = 1$.

Scanned by CamScanner

$\Delta \rightarrow$ Carry propagation Delay

FA4: $A_3 = 1$, $B_3 = 0$, $C_3 = 1$ } $S_3 = 0$, $C_4 = 1$.

$S = S_3\ S_2\ S_1\ S_0$ $\quad C_4 = 1$.

$\quad\ \ 0\ \ 1\ \ 0\ \ 0$

Total Addition Time $= 1 + 4 * \Delta$.

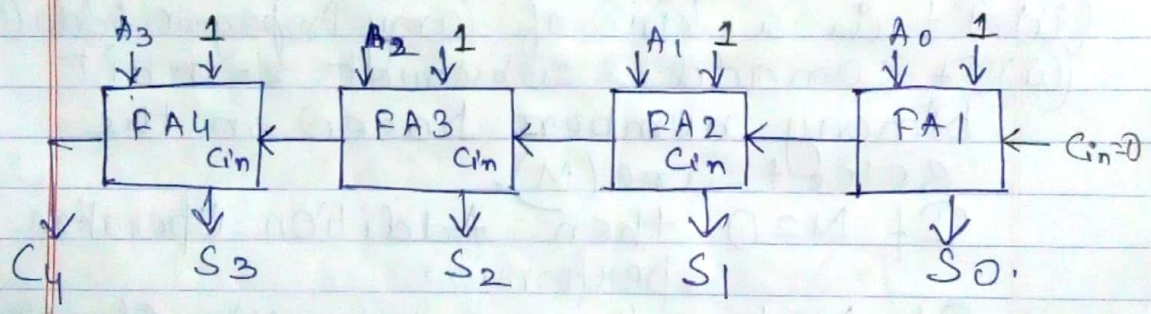
If, m is the total FA used, the.

Time = Total Clock Time + $m * \Delta$.

3. Binary Decrementing Circuit. (4-bit)

→ It is decrementing the value of a binary number by 1-bit.

→ To implement BDC we need multiple-FA.



Neglect the Carry.

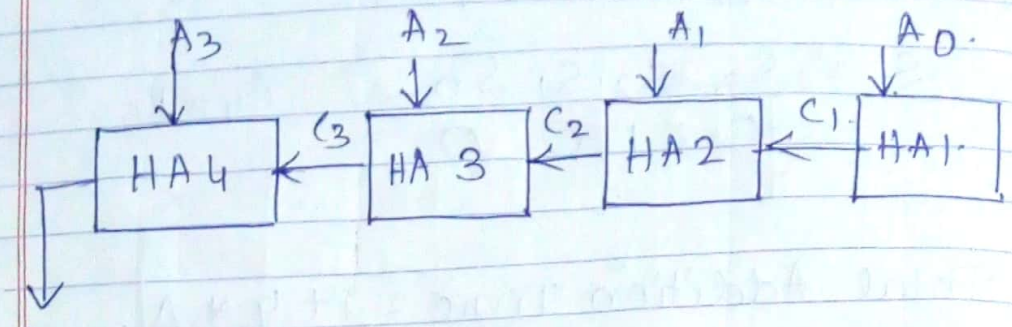
$A = A_3\ A_2\ A_1\ A_0$

$\quad\ \ 1\ \ 1\ \ 1\ \ 0.$

$$\begin{array}{cccc} 3 & 2 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ & & (-) & 1 \\ \hline 1 & 1 & 0 & 1 \\ \hline \end{array}$$

Scanned by CamScanner

4. Binary Incrementing Circuit.



$A = A_3 A_2 A_1 A_0$
1 1 1 0.

3	2	1	0
1	1	1	0
		(+)	1
1	1	1	1

5. Adder subtractor composite unit.

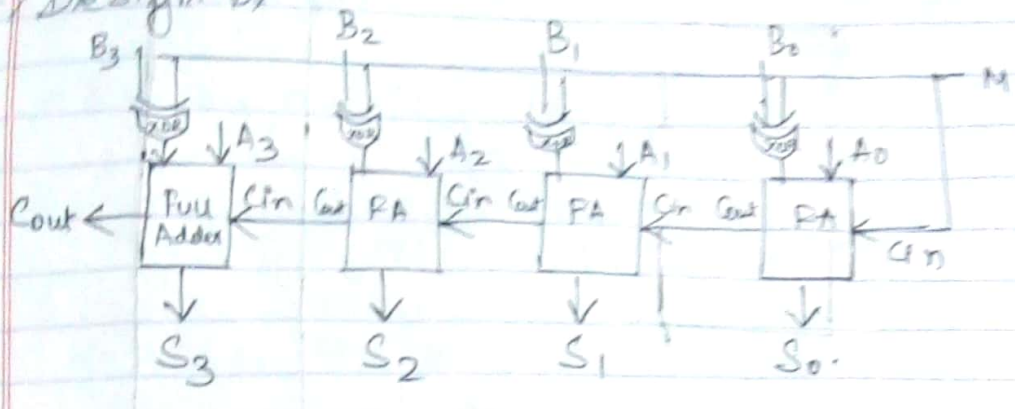
(i) It is a parallel Adder Circuit.
(ii) It is a kind of Carry Propagate Adder (CP
(iii) It can add & substract 2-n-bit binary numbers based on the select line (M).
If M=0 then Addition Operation performed.
If M=1 then Substraction Operation performed. (2's complement addition
(iv) To construct Adder Subtractor Composite unit we require n-no. of FA & n-no. of XOR gates.

(v) Adder-Substractor Unit also suffers from carry propagation delay from A

DC :—

5) Adder Subtractor Composite unit.

* Design :



Say, M = 0 , A = $A_3 A_2 A_1 A_0$, B = $B_3 B_2 B_1 B_0$

1 1 1 0 , 0 1 1 0

```
    1 1 1 0
    0 1 1 0
  ---------
1   0 1 0 0
C4  S3 S2 S1 S0.
```

Total amout of time req. for the result = 1 + 4 * Δ.

6. Design a 4-bit Arithmetic Unit.

S_1	S_0	Cin	Y	K = A+Y+Cin	
0	0	0	B	K = A+B	Add
0	0	1	B	K = A+B+1	Addition with carry
0	1	0	$\overline{B}$	K = A+$\overline{B}$	→ Substraction without borrow
0	1	1	$\overline{B}$	K = A+$\overline{B}$+1	→ Sub. ~~with borrow~~
1	0	0	1	K = A-1	Decrement
1	0	1	1	K = A	Transfer
1	1	0	0	K = A	"
1	1	1	0	K = A+1	Increment

Scanned by CamScanner

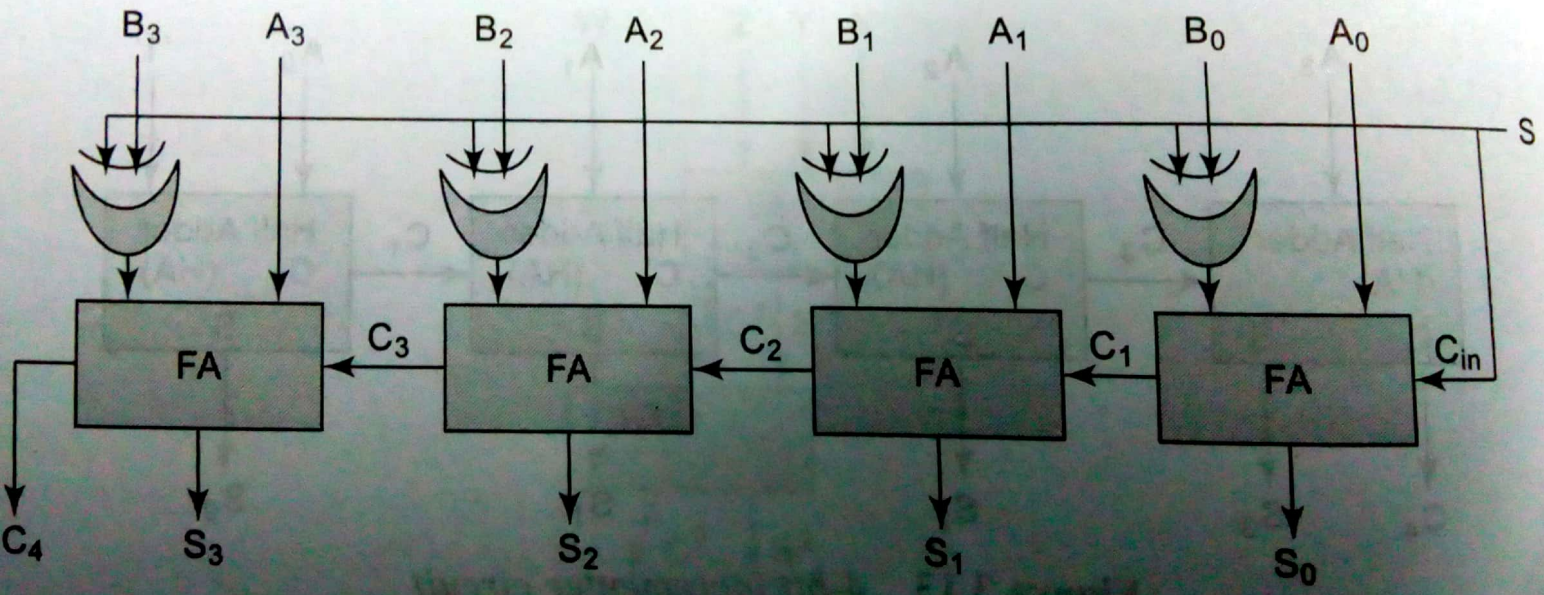


Figure 3.15 *4-bit binary adder-subtractor*

Scanned by CamScanner

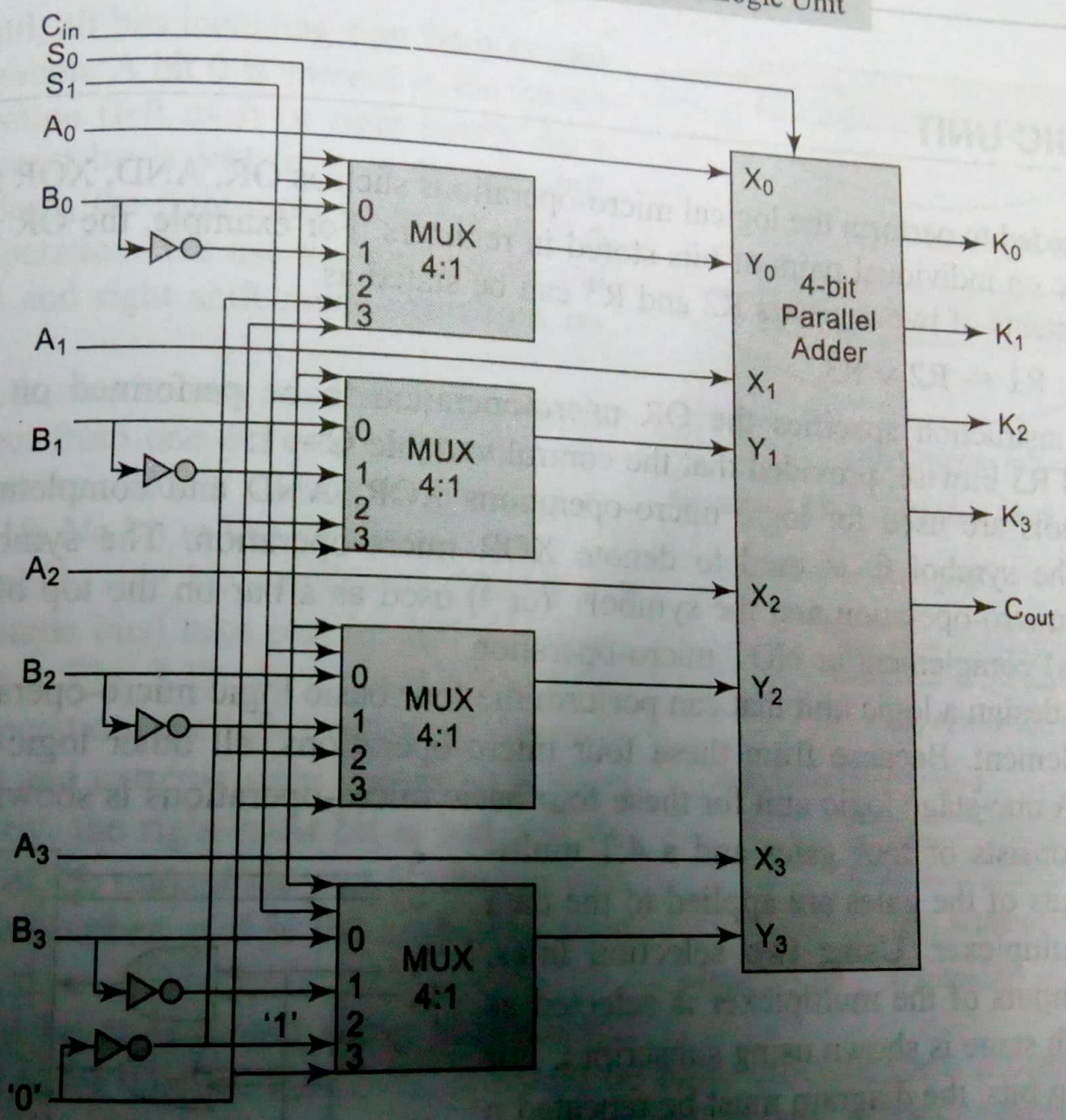


Figure 3.16 *4-bit arithmetic unit*

Scanned by CamScanner

Design of Control Unit :-

Objective :-

1. Introduction of Control Unit.
2. Major task of Control Unit
3. Design of general model of Control Unit
4. Classification of Control Unit.
   1) Hardwired control Unit.
   2) Microprogrammed CU.
5. Design a Hardwired control unit & it's advantages & disadvantages
6. Basic defn of
   (i) Microinstruction.
   (ii) Microprogram.
   (iii) " routine.
   (iv) Control Word.
   (v) Control ~~Memory~~ Memory.
7. Classification of microinstruction.
   (a) Horizontal micro instruction.
   (b) Vertical " " -
8. Horizontal micro instruction
   - Advantages & Disadvantages
9. Vertical micro instruction
   - Advantages & Disadvantages
10. Designing a micro-progm CU.
11. Microprogram vs Hardwired CU.

Scanned by CamScanner

@Arunayan Bhattacharya

(I) Q What is a Control Unit?

Ans → CU is the integral part of CPU, which directs the entire operation by issuing control signals to the data path.
→ It act as the brain of Computer System.
→ Control Unit supervises any operation to be performed inside a computer.
→ CU is responsible taking any decision.

(II) Major Task of CU or Basic task of Control Unit, (or functional req. of CU)

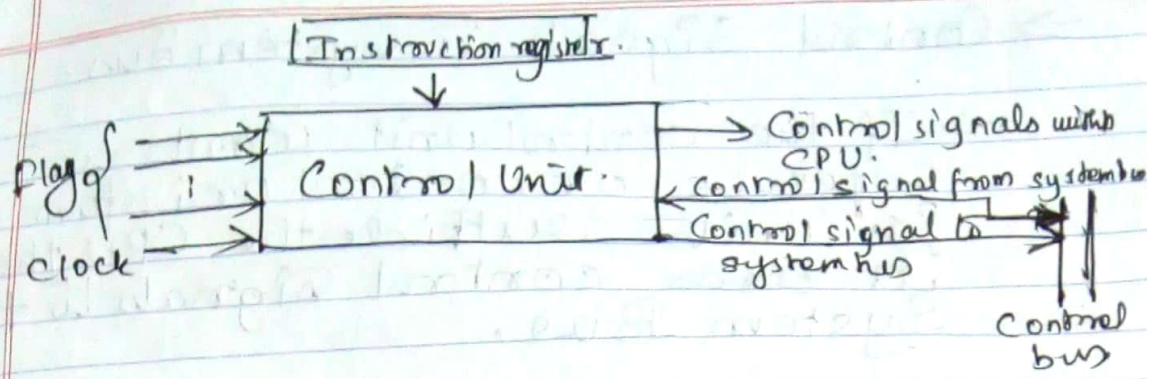
Ans (i) Sequencing
→ The control unit makes the processor to move through the microoperation in proper sequence, depending on the program beign executed.

(ii) Execution
→ The CU causes the execution of microoperation by generating control signals that are required for opening of gates to let data pass, performing ALU operation & closing gates. (CU sends signal to perform which operation Arithmetic/Logical work to be done)
→ The execution of an instruction involves the execution of various subtasks called cycles.

(III) General model of Control Unit.

P.T.O.

Scanned by CamScanner



CLOCK → Input Signal, to measure time. Crystal oscillator is generating the clock.

Flags → Indicate the current State (status of the output) of a processor & result of recent ALU operation to the CU. Belong to Flag Resistor → Flag Bit.

Instruction register → Stores / Holds the OPCODE of the current instruction/register, for it to get executed.

→ Flag Bit will help to determine the Status of the output After the current Instruction is executed. It is given about the old instruction.

→ Control Signal Bus
→ An interrupt or Acknowledgement will come with the help of this this.

→ Control Signal within CPU
→ If control unit wants to inform or Supervise or control any other unit inside the CPU then it can do it using control Signal within CPU

→ Control Signals to System Bus.

→ If the control unit wants to supervise or control or inform any unit outside the CPU then it uses control signals to the System Bus.

④ Classification of Control Unit.

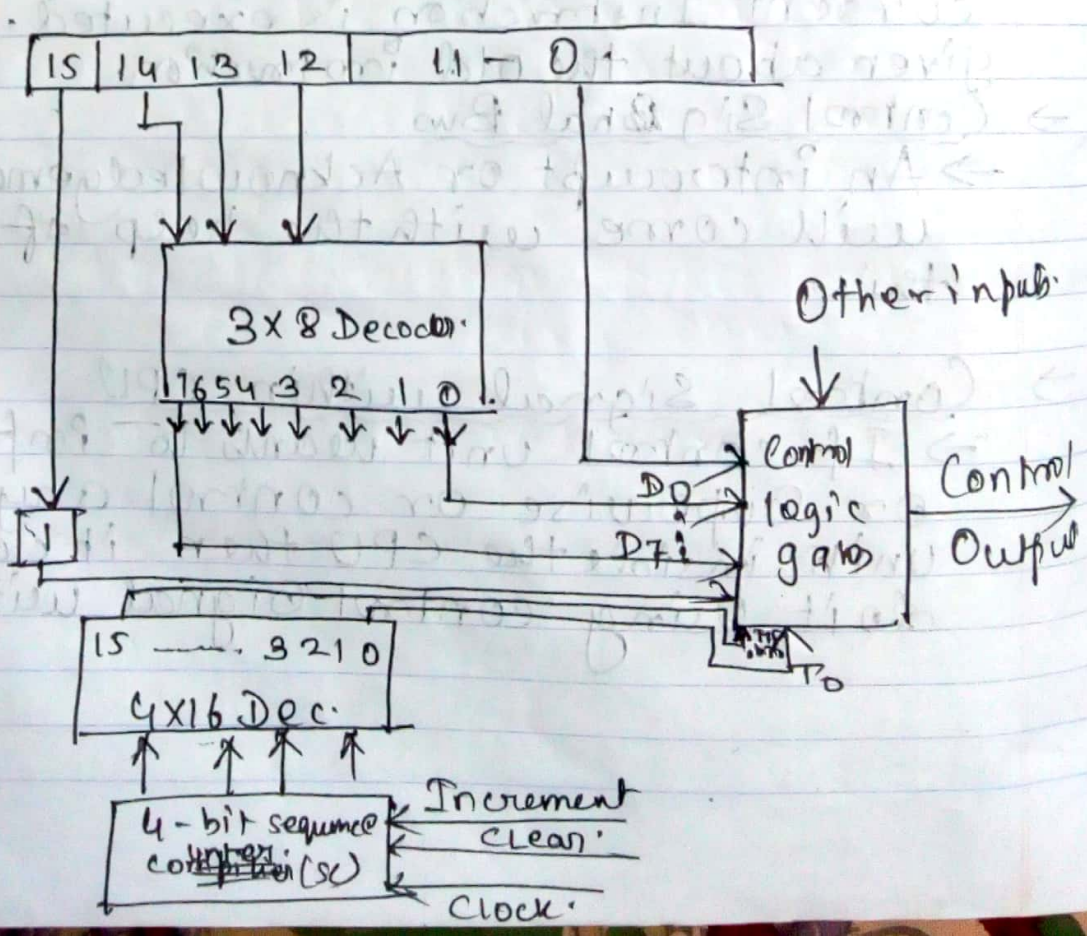
ⓐ Hardwired CU.

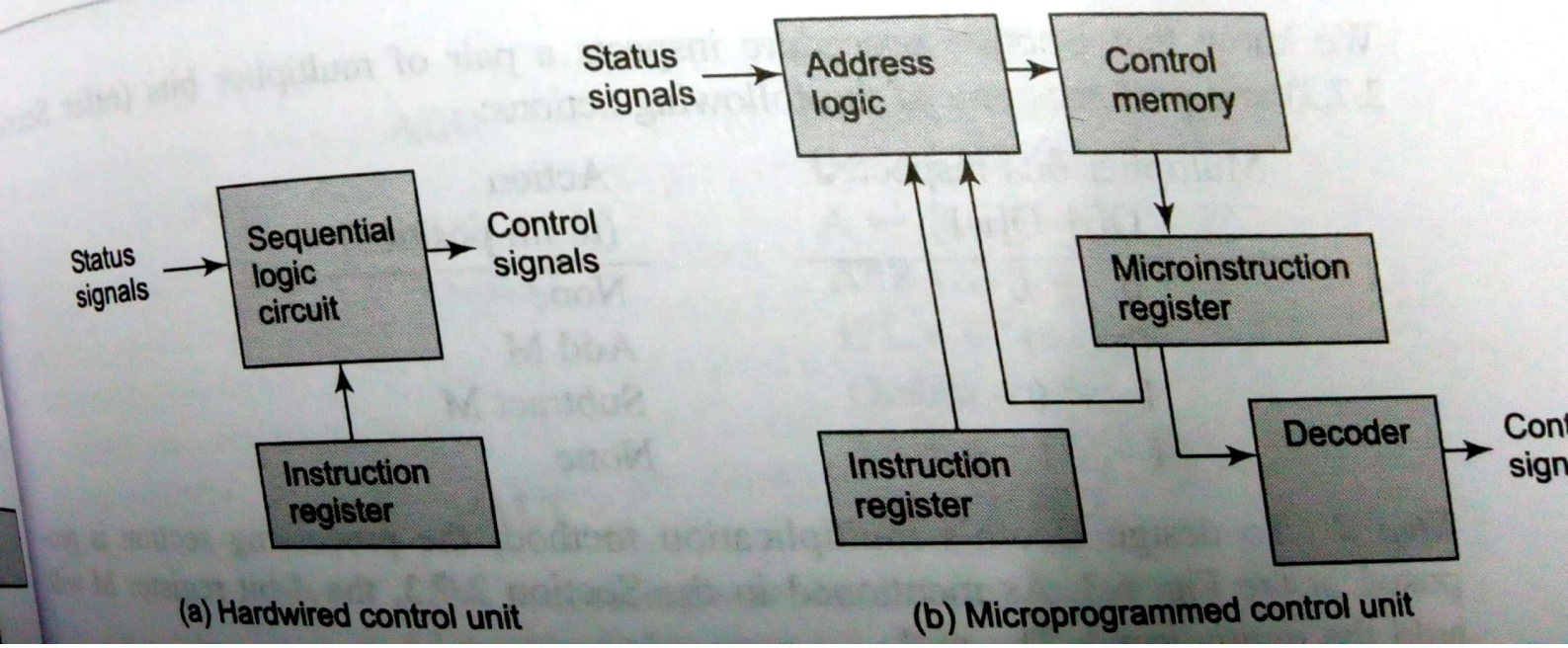
→ If we design a CU using a Hardware approach or using some combinational or sequential circuit.

ⓑ Microprogrammed CU.

→ When CU is designed using a software approach.

⑤ Hardwired CU.





(a) Hardwired control unit

(b) Microprogrammed control unit

Scanned by CamScanner

When clear is clicked → the counter is reset to 0.

4-bit sequence counter
→ It is used to Increment the value and passed to 4x16 Decoder.

Advantages :—

1. It is fast.
2. If we want to construct a CU then we use Hardwired CU.
3. No language Conversion required.

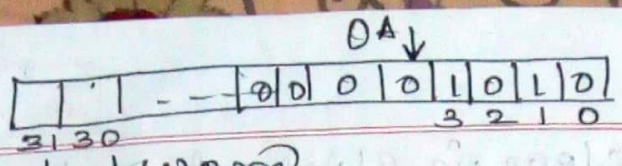
Disadvantage :—

1. Hight cost / Costly.
2. It is not flexible.
3. Debugging is very difficult.
4. Maintainance is not easy.

6. ~~A~~
a) Microinstruction
→ A instruction which is executed at micro / hardware / IC level / (that controls the data flow & sequencing in a processor).
b) Microprogram
→ Set or Series of microinstructions that performs various microoperations.
c) Micro program Routine

→ In the control memory ~~m~~ microinstruction substore in a group, where each group specifies a routine, that routine is micro-program routine.

Scanned by CamScanner



Control word.

(d) Control word
→ It refers to a binary string consisting of 0's & 1's where each bit represent a control line. Suited inside a control memory.

(e) Control memory
→ Control memory holds all the information related to all the microinstruction that are needed for a micro-program.

(7) Horizontal VS Vertical Microinstruction :—

□ Horizontal Microinstruction

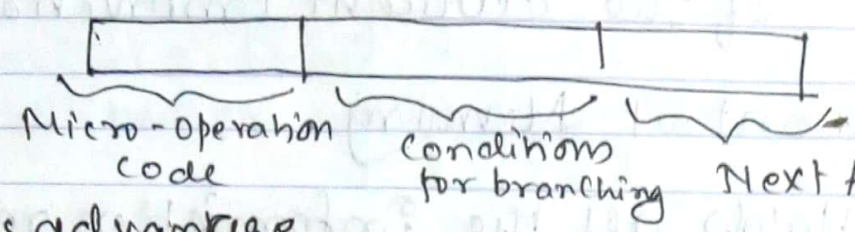
→ (1) Can be called parallely.
(2) In this microinstruction, each individual bit corresponds to an individual control signals required in the CPU. For e.g., if a CPU need 32 control signals, then at least 32-bits are req. in each horizontal microinstruction

Adv :- (1) Operation speed very fast.
(2) Decoding easy.

Dis Adv (1) Control word size is large

Vertical Microinstruction.

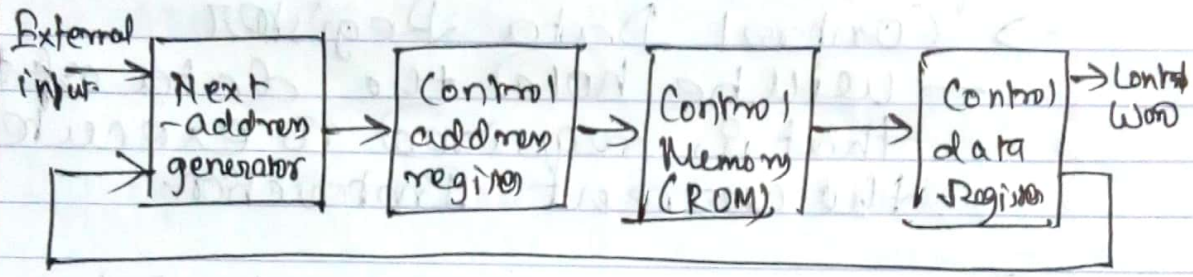
In Vertical Microinstruction, a single field can produce an encoded sequence. It takes more time in generating signals due to decoding time.



Disadvantage

→ ① Decoding is Time Consuming.

(10)



Advantage

→ Micro program CU is less complex.
→ Cheaper & less error prone.
→ It provides flexibility as it is not req. to change the hardware configuration
→ Debugging & maintainance is easy
→ Complex function can be done efficiently.

Disadvantage

→ It works with slow speed
→ Costs too much if we design a small CPU
→ Flexibility is achieved at some extra hardware cost.

Scanned by CamScanner

OA → ADD A, B. (@Orunayan@)

[A] ← [A] + [B].

→ Next - address generator
- will generate the address of the next instruction output.

→ Control Address Register (CAR)
- holds address of the opcode of the current instruction.

→ Control Memory.
- Holds all the information related to all microinstruction for a certain micro-program.

→ Control Data Register
- will be hold the data that is required to execute the current Instruction.

Each by each the control is generated.

Scanned by CamScanner

→ Design of ALU (Logical Unit).

Objective :-

→ To perform Logical Task.

Q. Design a 4-bit Logical Unit using Multiplexer and ~~Basic~~ Logic Gates.

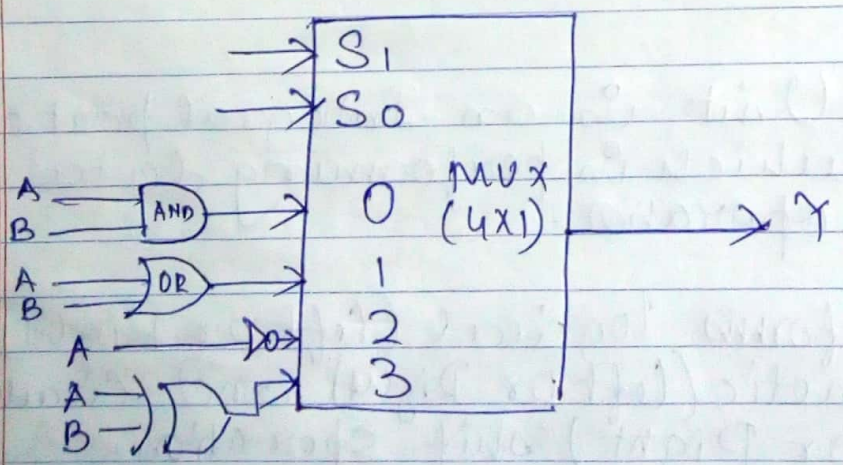
→ Requirement :-

(a) Multiplexer (4 X 1).
(b) AND, OR, NOT, XOR Gate.

No. of Select lines = $\log_2 4 = 2$.

Select line.		O/P (MUX).
S_1	S_0	
0	0	$Y \leftarrow A \wedge B$ (AND)
0	1	$Y \leftarrow A \vee B$ (OR).
1	0	$Y \leftarrow \overline{A}$. (NOT)
1	1	$Y \leftarrow A \oplus B$ (XOR)

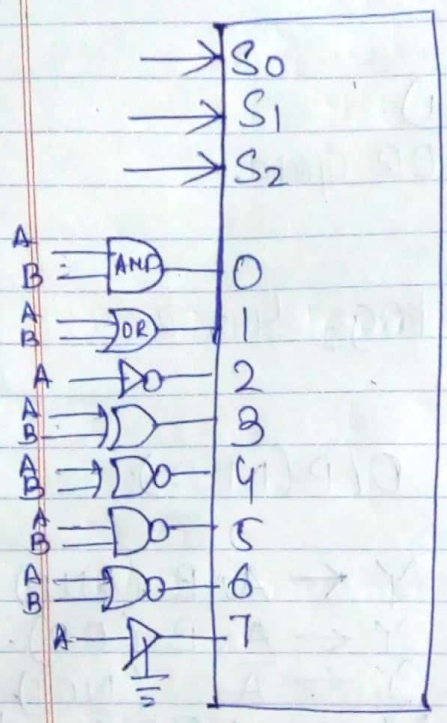
Circuit :-



Scanned by CamScanner

Q. Design a 8-bit logical unit using 8-logic gate:-

	S_0	S_1	S_2
(1) AND	0	0	0
(2) OR	0	0	1
(3) NOT	0	1	0
(4) XOR	0	1	1
(5) XNOR	1	0	0
(6) NAND	1	0	1
(7) NOR	1	1	0
(8) GND	1	1	1



SHIFTER UNIT:-

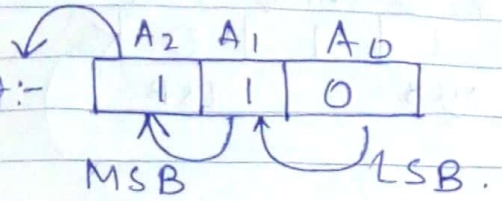
→ Shifter Unit is an Integral part of ALU which is performing data Shift operation.

→ It performs logical (Left or Right), Arithmetic (Left or Right) and Circular (left or Right) shift operation.

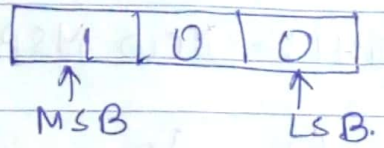
Scanned by CamScanner

logical left: LSB is always replaced by 0.

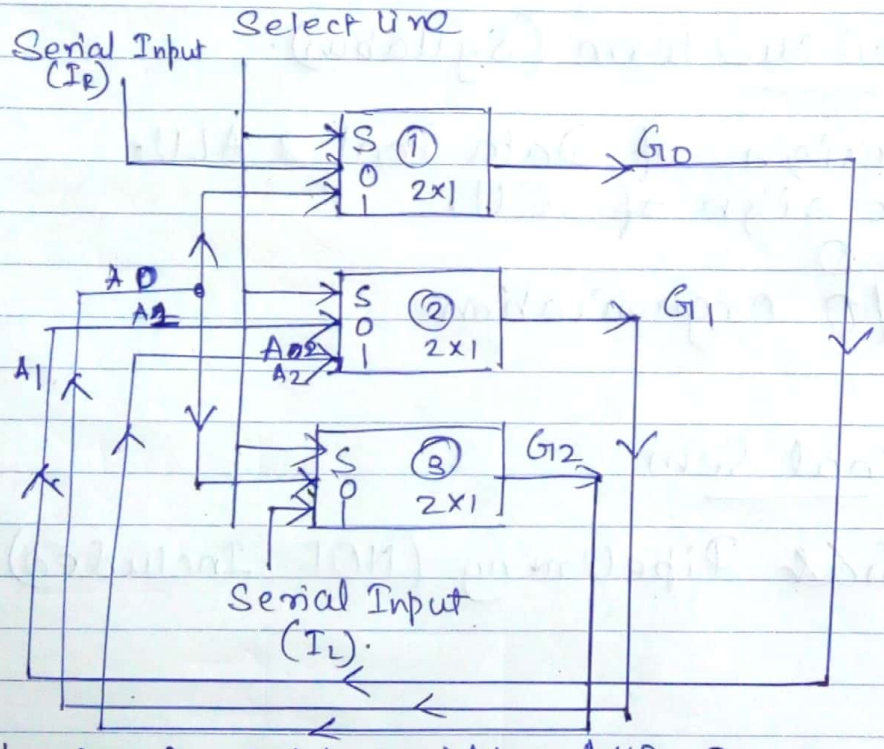
step1 logical left:-



step2



Select line
{ 0 → left shift.
{ 1 → right shift.



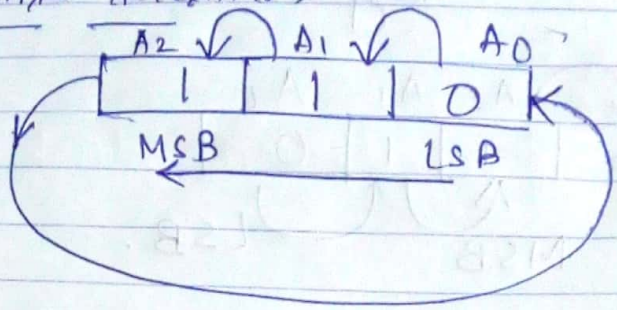
Logical right shift MSB is always filled up by 0.

Arithmetic left shift ⇒ New LSB = OLD LSB.

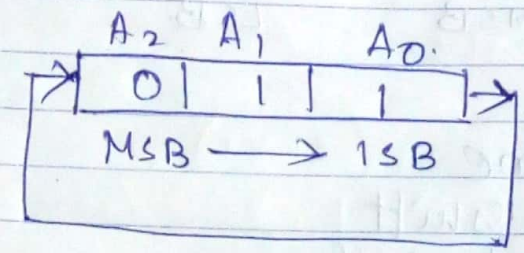
Arithmetic right shift ⇒ New MSB = OLD MSB

CIRCULAR LEFT :- ~~OLD~~ New LSB = Old MSB.

~~CIRCULAR RIGHT :-~~



CIRCULAR RIGHT :- New MSB = Old LSB.



2nd Mid term (Syllabus).

1. Design of Data Part & ALU.
2. Design of CU.
3. MO
4. I/O organisation.

Final Sem

1. ~~Final~~ Pipelining (NOT Included).

Scanned by CamScanner

NO CLASS

MISSED.

ALL DAY CLASS

NOTES PRESENT.

Scanned by CamScanner

NOTES AFTER MID SEM-1.

IF U DON'T HAVE NOTES BEFR MID-SEM, JUST PINGME.

Scanned by CamScanner