28-01-25

Module-2

Register Transfer Logic

A digital system is a sequential logic system constructed with flipflops, gates, adders, decoders, counters, etc. To specify a large digital system with a state table would be very difficult. To describe a digital system in terms of functions such as decoders, adders & registers, it is necessary to employ a higher level mathematical notation. Thus the registers are selected to be the primitive components in the digital system. The register transfer logic describes the information flow & processing tasks among the data stored in the registers in a precise & concise manner.

The components of a digital s/m

1. The set of registers in a system & their fns.
2. The binary coded information stored in the register.
3. The operation performed on the information stored in register.
4. The control fns that initiate the sequence of operations

Registers

Registers is a memory unit which is capable of storing information. For example, a memory unit is considered to be a collection of storage registers where information can be stored.

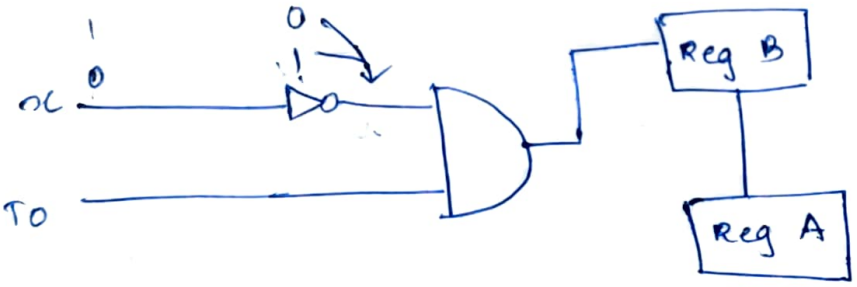
Types of registers

PC,
MAR,
MDR,
IR,
General pupose Registers, etc.

Representation of Register transfers

Example of RTL

$x'T_0 : A \leftarrow B$



The operations performed on the data stored in the register are called microoperations.

Types of microoperation

1. Arithmetic operation
2. Logical "
3. Shift "

Symbolic Representation	Description
$F \leftarrow A+B$	content of A+B is transferred to F
$F \leftarrow A-B$	content of A-B is transferred to F.
$B \leftarrow \overline{B}$	complement the register B.
$B \leftarrow B'+1$	forms the 2's complement of the content of register B
$F \leftarrow A+\overline{B}+1$	A + 2's complement of B (subtraction)
$A \leftarrow A+1$	Increment the register A by 1
$A \leftarrow A-1$	Decrement the register A by 1

24-01-25 Logical M operation.

Symbolic Representation	Description
$A \leftarrow \overline{A}$	Complement all bits of register A.
$F \leftarrow A \vee B$	Perform logical OR M operation
$F \leftarrow A \wedge B$	Logical AND M operation.
$F \leftarrow A \oplus B$	Logical exclusive OR M operation.

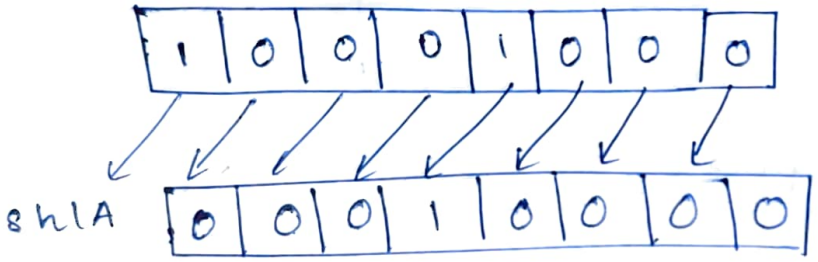
Shift M operation

~~Symbolic Rep~~
~~resentation~~

Shift the content either left or right. These M operation are used for serial transfer of data. There are 3 types of shift M operation logical, arithmetic & circular.

1. Logical Shift Register ~~(sh)~~

Logical left shift (shLA)



Logical shift right (shRA)

2. Arithmetic shift M operation

ashlA ⇒ same as that of shLA

ashrA ⇒ 

↓

copy the MSB as such & perform the right shift operation

3. Circular shift.
round around / change MSB → LSB or LSB → MSB as required

cil

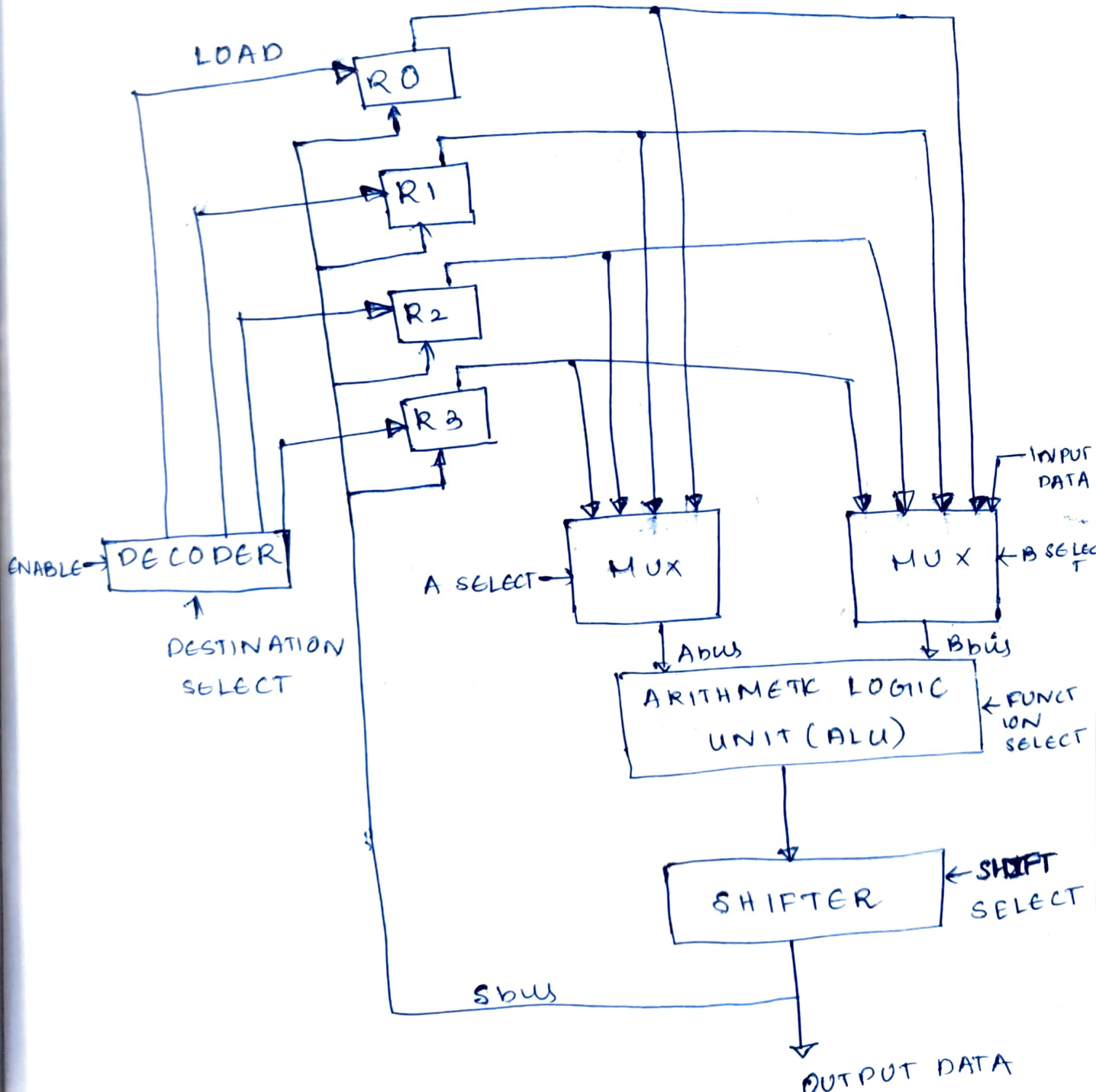
0	0	0	1	0	0	0	1

cir

0	1	0	0	0	1	0	0

Processor Organization

Bus Organization



$R_1 \leftarrow R_2 + R_3$

Step 1: MUX A selector : To place the content of R_2 onto the bus A

Step 2: MUX B selector : To place the content of R_3 onto the bus B

Step 3: ALU fn selector : To provide the arithmetic operation A+B

Step 4: shift selector : To transfer Data

Step 5: Decoder/ Destination selector : Transfer the content of bus S into R_1.

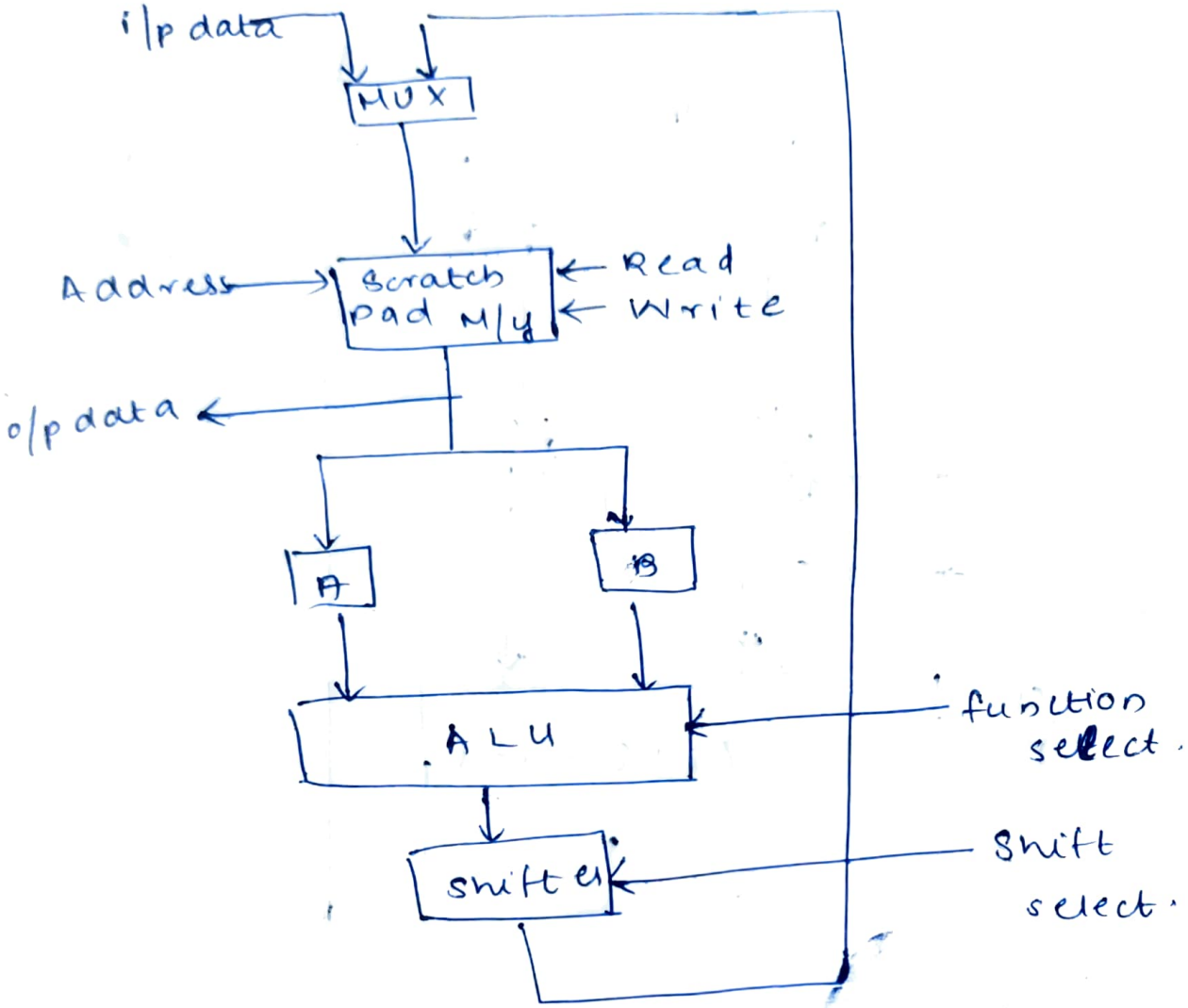
-01-25

Scratchpad memory

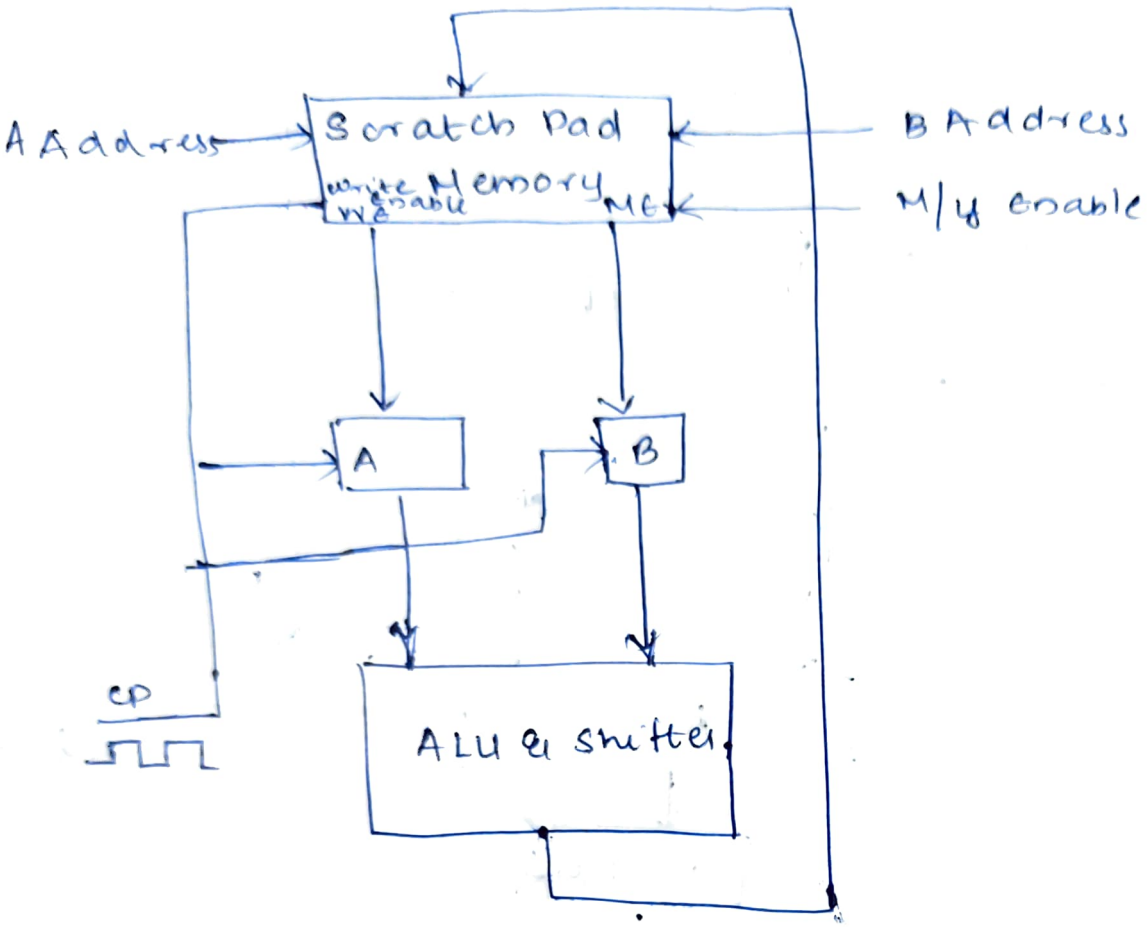
~~Large computer wire re~~

→ The registers in a processor unit can be enclose with in a small memory location unit called scratchpad memory.

→ It is cheaper & it require less wires



Sum processes employee, a 2-port memory inorder to ~~overcome~~ avoid the delay caused in the scratchpad memory. The clock input CP controls the memory read or write operation through the write enable WE. When CP=1, WE=0 read operation takes place. When CP=0, WE=1, write operation takes place.

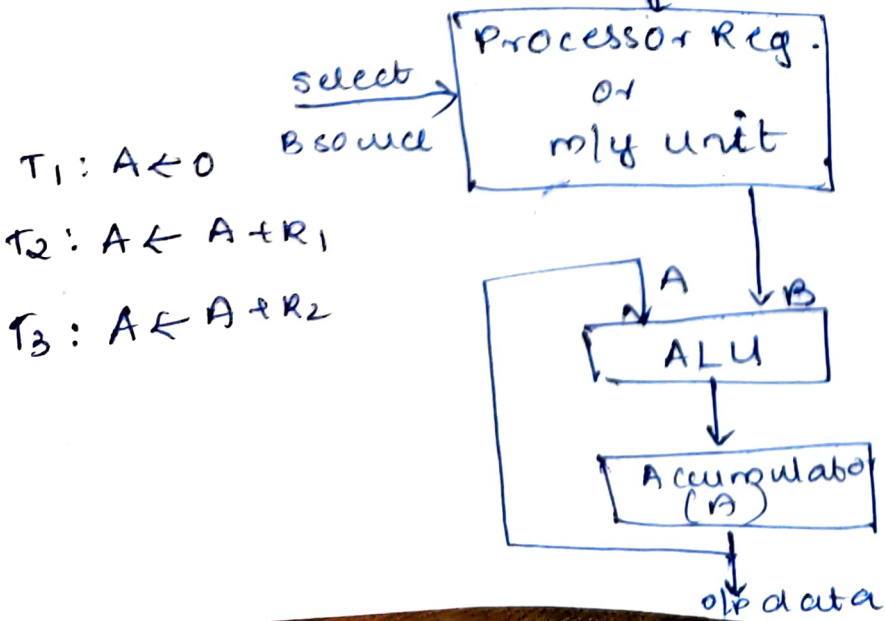


Accumulator

→ used for consecutively adding numbers.

→ used along with ALU but sometimes the accumulator itself is used for shift operations.

→ Accumulator derived its name from addition



$T_1 : A \leftarrow 0$

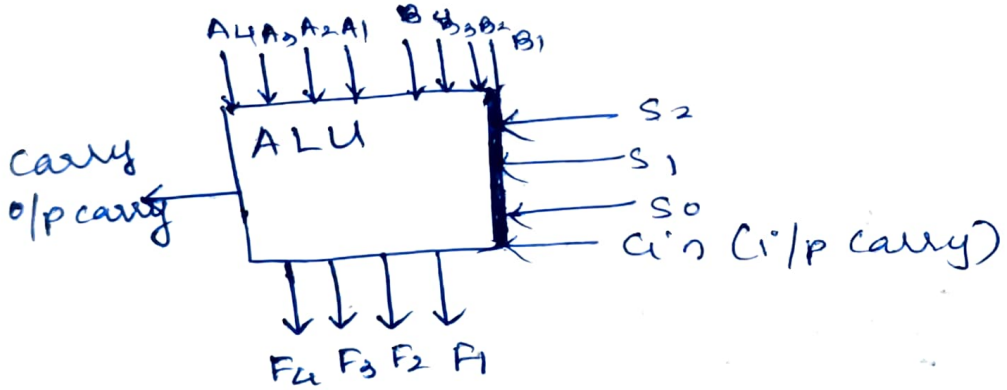
$T_2 : A \leftarrow A + R_1$

$T_3 : A \leftarrow A + R_2$

ALU performs a set of basic arithmetic & logical operation. The number of selection lines S_2, S_1 & S_0 select a particular operation in the unit.

S_2 (mode select): Distinguish b/w arithmetic & logic operation. When $S_2 = 0$, ALU performs arithmetic operation. When $S_2 = 1$, it performs logical operations.

S_1 & S_0 (function select): It specify the operation.



29/01/25

Design of ALU

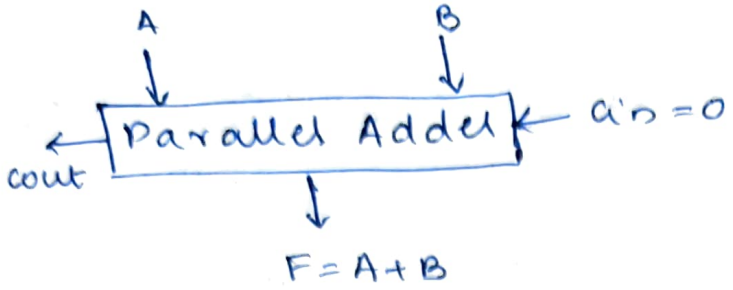
① Designing of Arithmetic section

② Designing of logical "

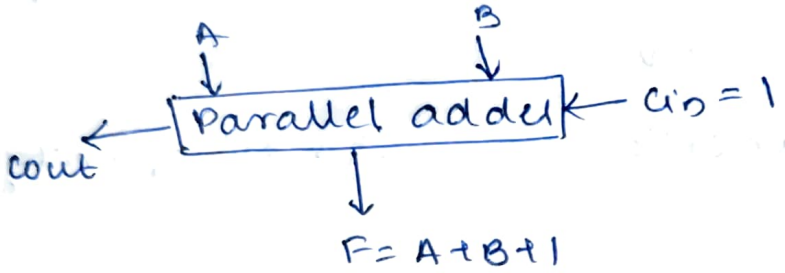
③ ~~Dest~~ Modify arithmetic section so that it can perform both arithmetic & logic section

Designing of arithmetic circuit.

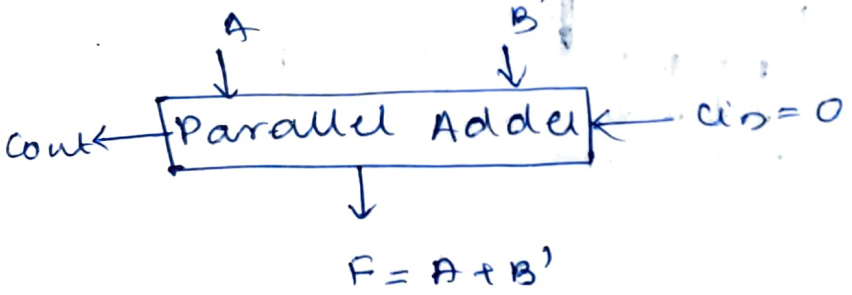
① Addition



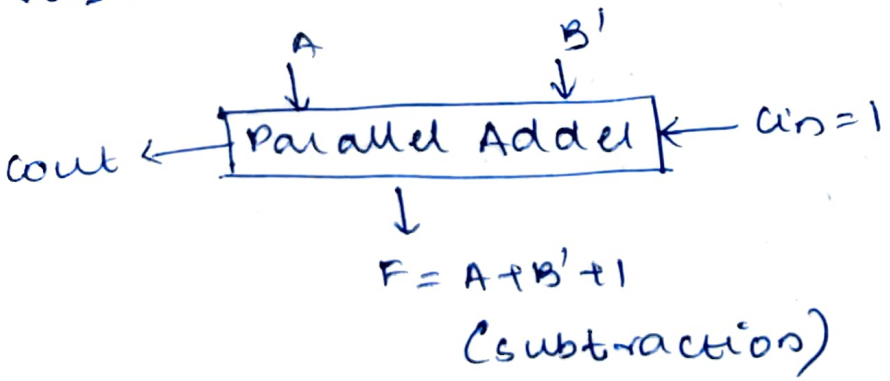
② Addition with carry.



③ A + B'

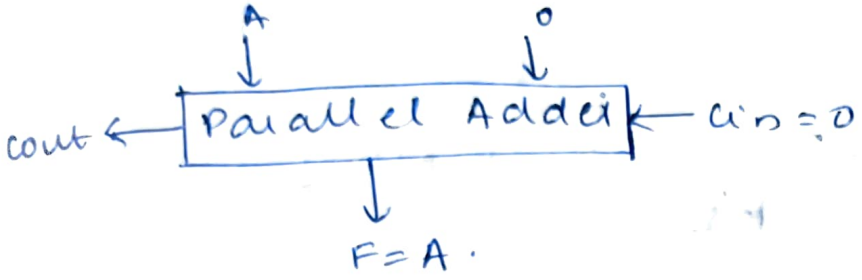


④ Subtraction

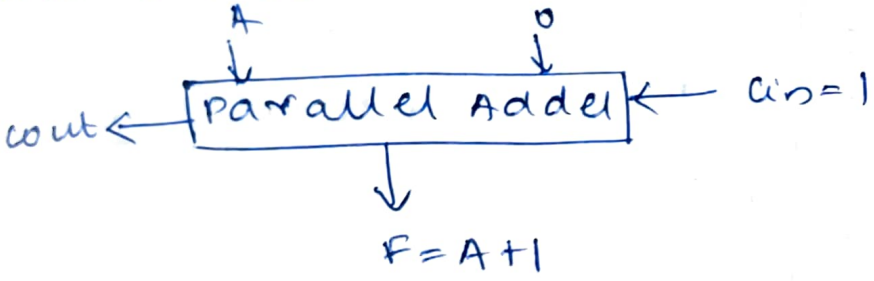


(Subtraction)

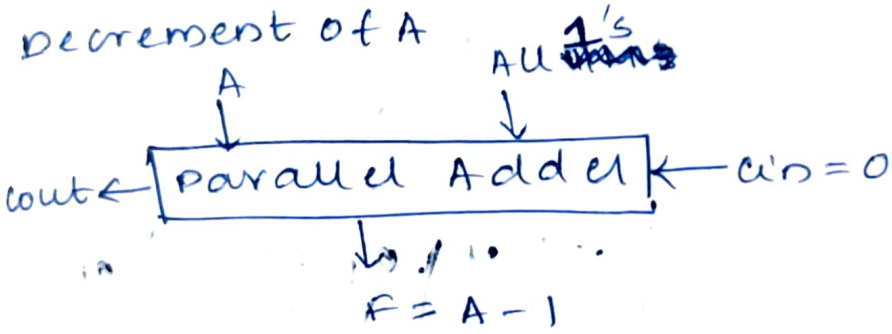
(5) Transfer of A.



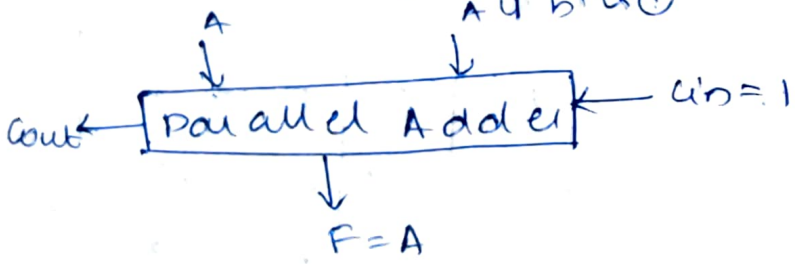
(6) Increment of A.



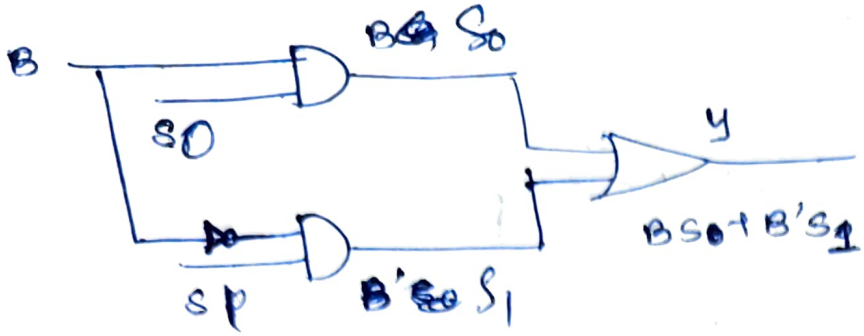
(7) Decrement of A



8) Transfer of A :



By giving diff values to B, we can perform 8 arithmetic operations & the values of B varies from 0, B, B' & 1. The circuit that controls i/p B is called a true or compliment 1 or 0 circuit.



S_1	S_0	y
0	0	0
0	1	B
1	0	B'
1	1	1

$S_1 = 0$
$S_0 = 0$

$B \cdot 0 + B' \cdot 0 = 0$

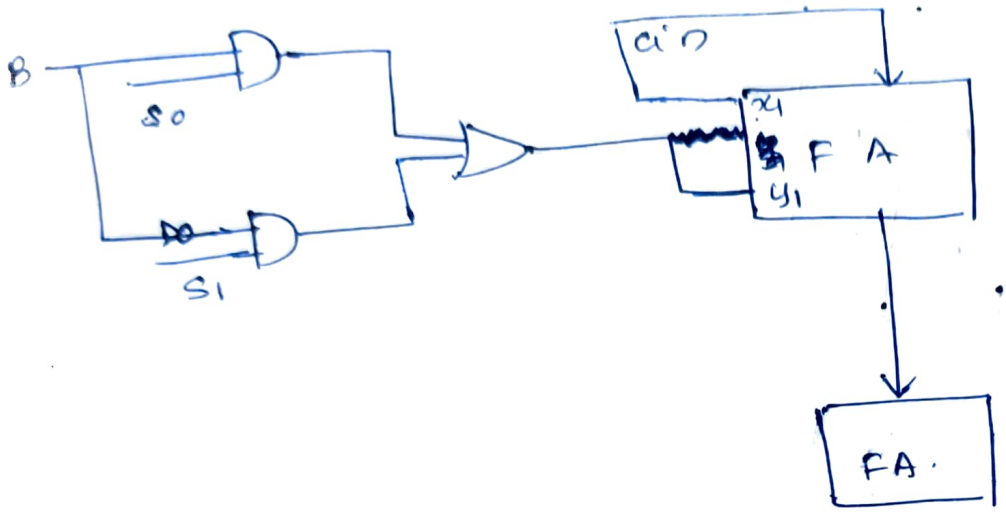
$S_1 = 0$
$S_0 = 1$

$B \cdot 0 + B' \cdot 1$

$= B'$

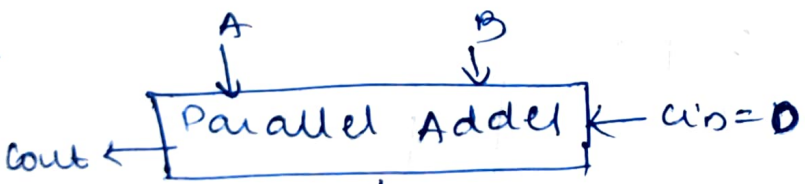
$B + B' = 1$

Function S_1	Function S_0	select c_{in}	x	y	o/p equal $F = x + y$	operation
0	0	0	A	0	$F = A$	Transfer of A
0	0	1	A	0	$F = A + 1$	Incrementation
0	1	0	A	B^2	$F = A + B$	Addition
0	1	1	A	B^2	$F = A + B + 1$	Addition with carry
1	0	0	A	B'	$F = A + B'$	A addition $A + B'$
1	0	1	A	B'	$F = A + B' + 1$	Subtraction
1	1	0	A	1	$F = A - 1$	Decrement of A
1	1	1	A	1	$F = A$	Transfer of A.

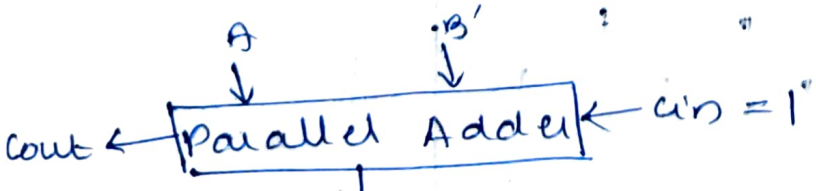


Qn) Design a Adder or subtractor circuit with one selection variables & 2 i/p variables A & B, S=0 perform addition A+B. When S=1, Subtraction takes place.

30/01/25

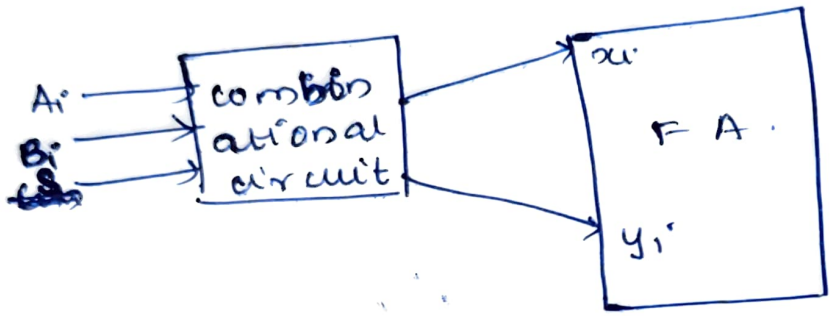


F=A+B.



F=A+B'+1

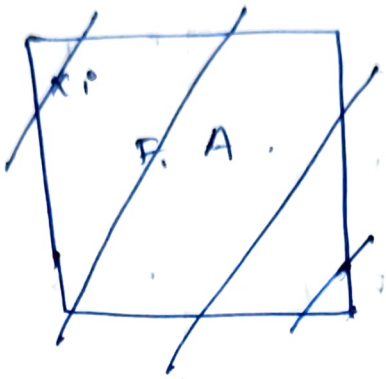
S	x_i	y_i	C_{in}
0	A	B	0
1	A	B'	1



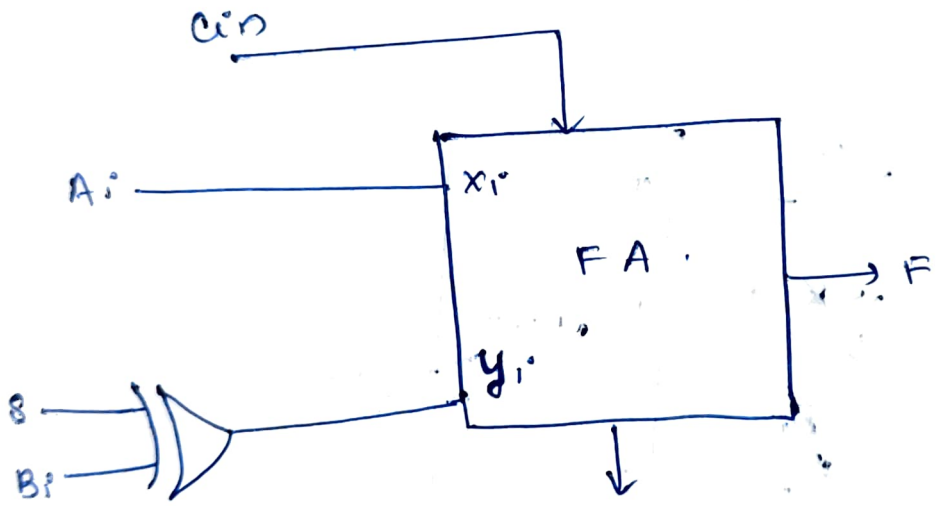
S	A	B	X	Y
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	1
1	0	1	0	0
1	1	0	1	1
1	1	1	1	0

$x_i = A_i$

S	B	y
0	0	0
0	1	1
1	0	1
1	1	0



$y_i = B_i \oplus S$

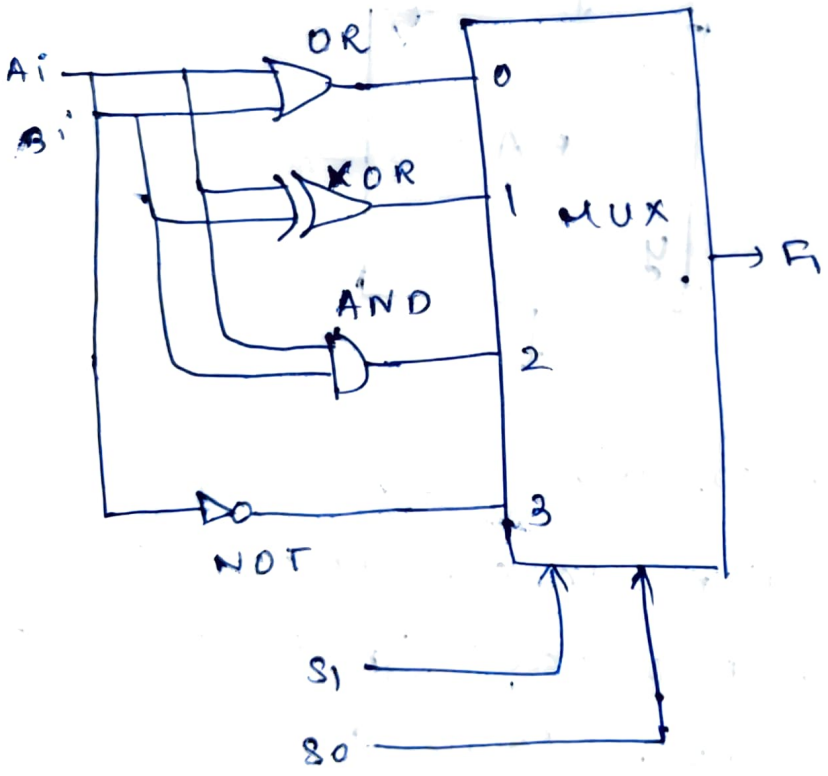


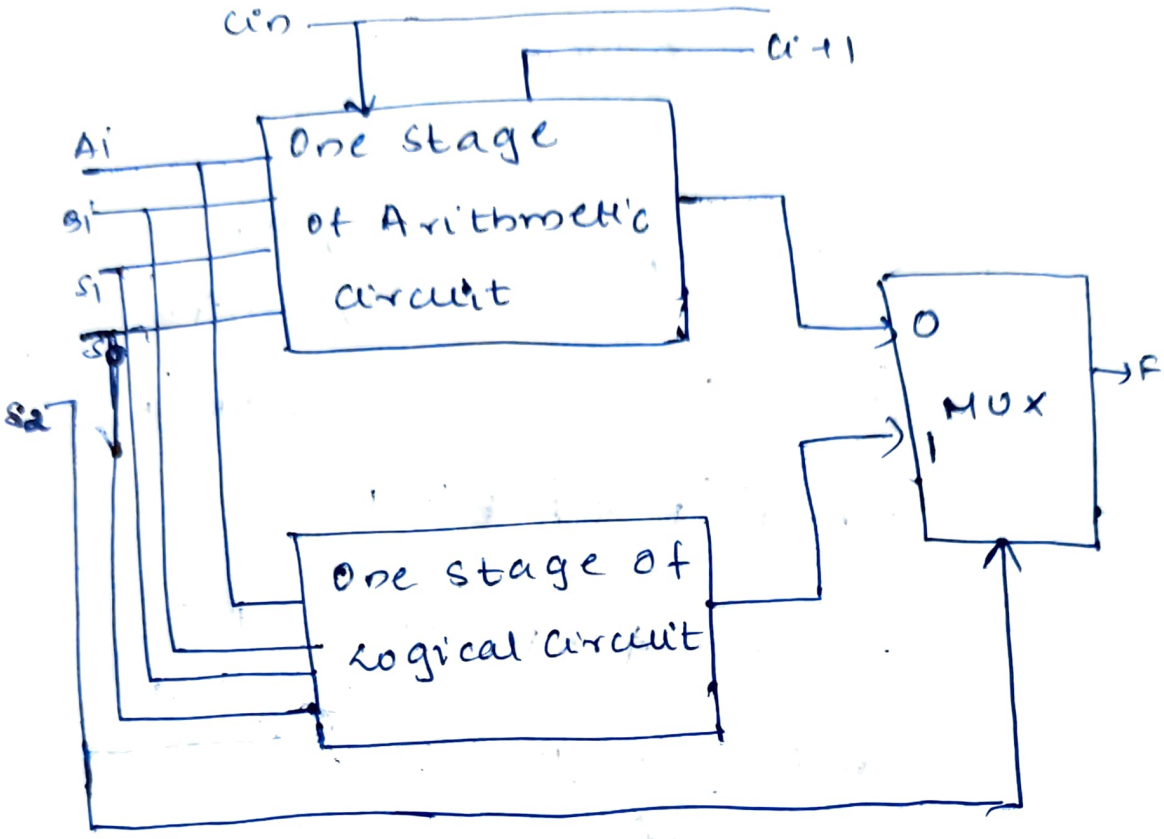
Tutorial (After exam)

1. Design an arithmetic circuit with one selection variable S and two data i/p A and B. When S = 0, the circuit performs addition operation ie, F = A + B. When S = 1, the circuit performs the increment operation F = A + 1.
2. Design an arithmetic circuit that generate the following arithmetic operation. Draw the logic diagram of 1 typical stage

S_1	S_0	$C_{in} = 0$	$C_{in} = 1$
0	0	$F = A + B$	$F = A + B + 1$
0	1	$F = A$	$F = A + 1$
1	0	$F = \bar{B}$	$F = B' + 1$
1	1	$F = A + \bar{B}$	$F = \dot{A} + B' + 1$

Designing of Logical Circuit





22/01/25

Modifying the arithmetic circuit to get logical operation

$S_2\ S_1'\ S_0'$

S_2	S_1	S_0	x	y	Cin	Operation	Expected
1	0	0	A	0	0	Transfer of A	OR
1	0	1	A	B	0	XOR	XOR
1	1	0	A	B'	0	XNOR	AND
1	1	1	A	1	0	NOT	NOT

Full adder, $F = A \oplus B \oplus C \Rightarrow A \oplus B$

$S_1\ S_0$

0 0 $A \oplus 0 = A'0 + A1 = A$

0 1 $A \oplus B = A'B + AB'$

1 0 $A \oplus B' = A'B' + A'B$ = XNOR

1 1 $A \oplus 1 = A'1 + A0 = A'$

$A \oplus B = A'B + AB'$

$S_2\ S_1'\ S_0'$

1 0 0 $\quad A+B \Rightarrow OR$

$S_2\ S_1\ S_0'$

1 1 0 $\Rightarrow (A \oplus B)'$

$A \oplus B$

$(A+K) \oplus B'$

$(A+K)' B' + (A+K) B$

$(A+K) \oplus B'$

$= (A+K)' B' + (A+K) B$

$= A'K'B' + AB + KB$

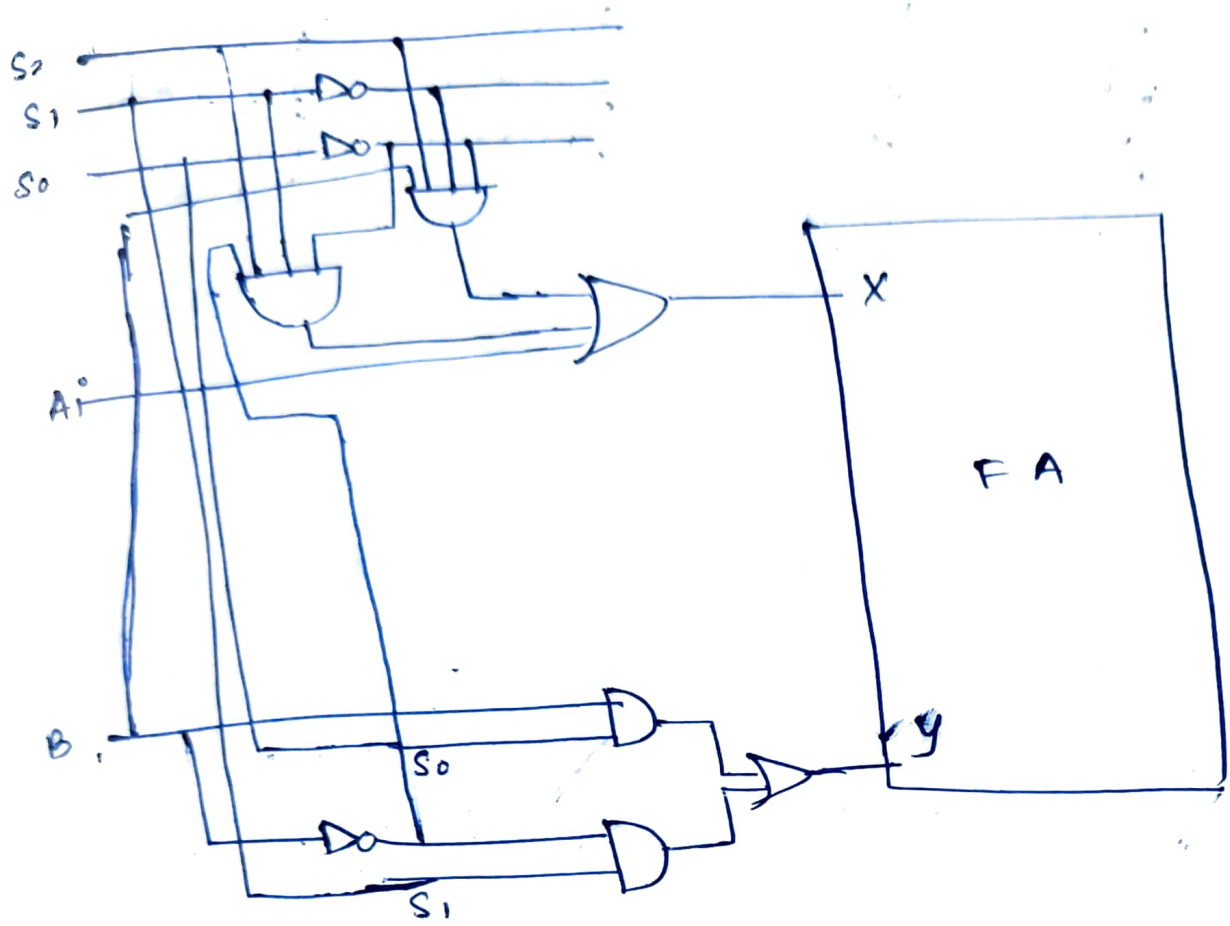
Sub $K = B'$

$= A'B'B' + AB + B'B$

$= A'B'B' + AB + B'B$

$= AB$

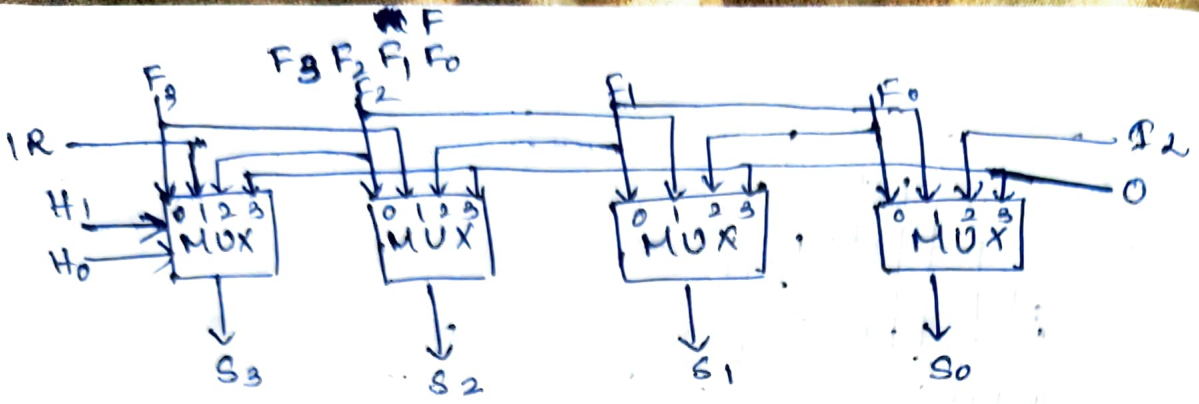
Arithmetic	Logic
$X_i = A_i$	$X_i = A + S_2 S_1' S_0' \cdot B + S_2 S_1 S_0 B$
$Y_i = B S_0 + B' S_1$	$Y_i = B S_0 + B' S_1$



Designing of shifter

shifter help us to transfer the o/p of the ALU on to the o/p bus. The shifter may transfer the information directly without a shift or it may shift the information to the right or left.

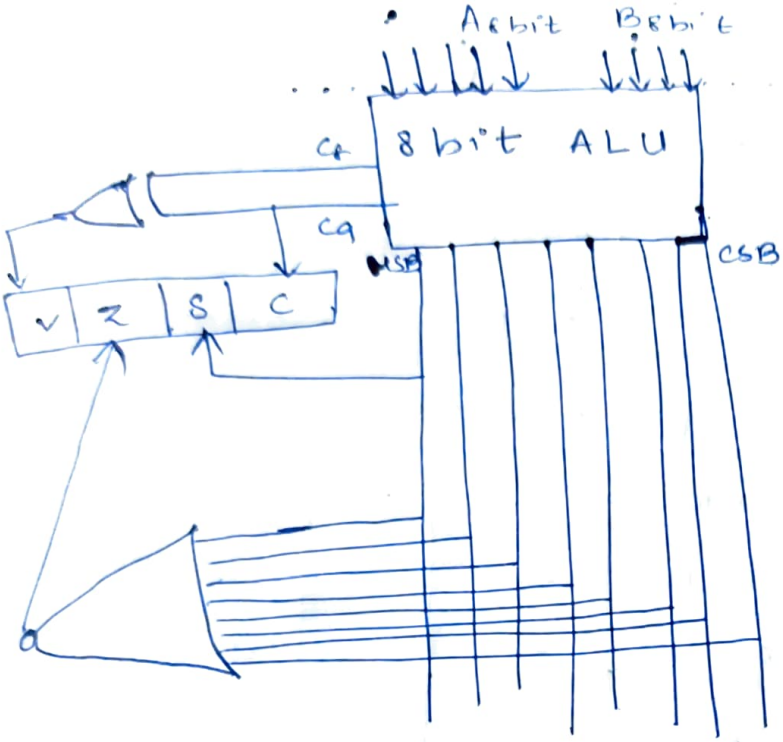
H_1	H_0	
0	0	$S \leftarrow F$
0	1	shr F
1	0	shl F
1	1	clear (0)



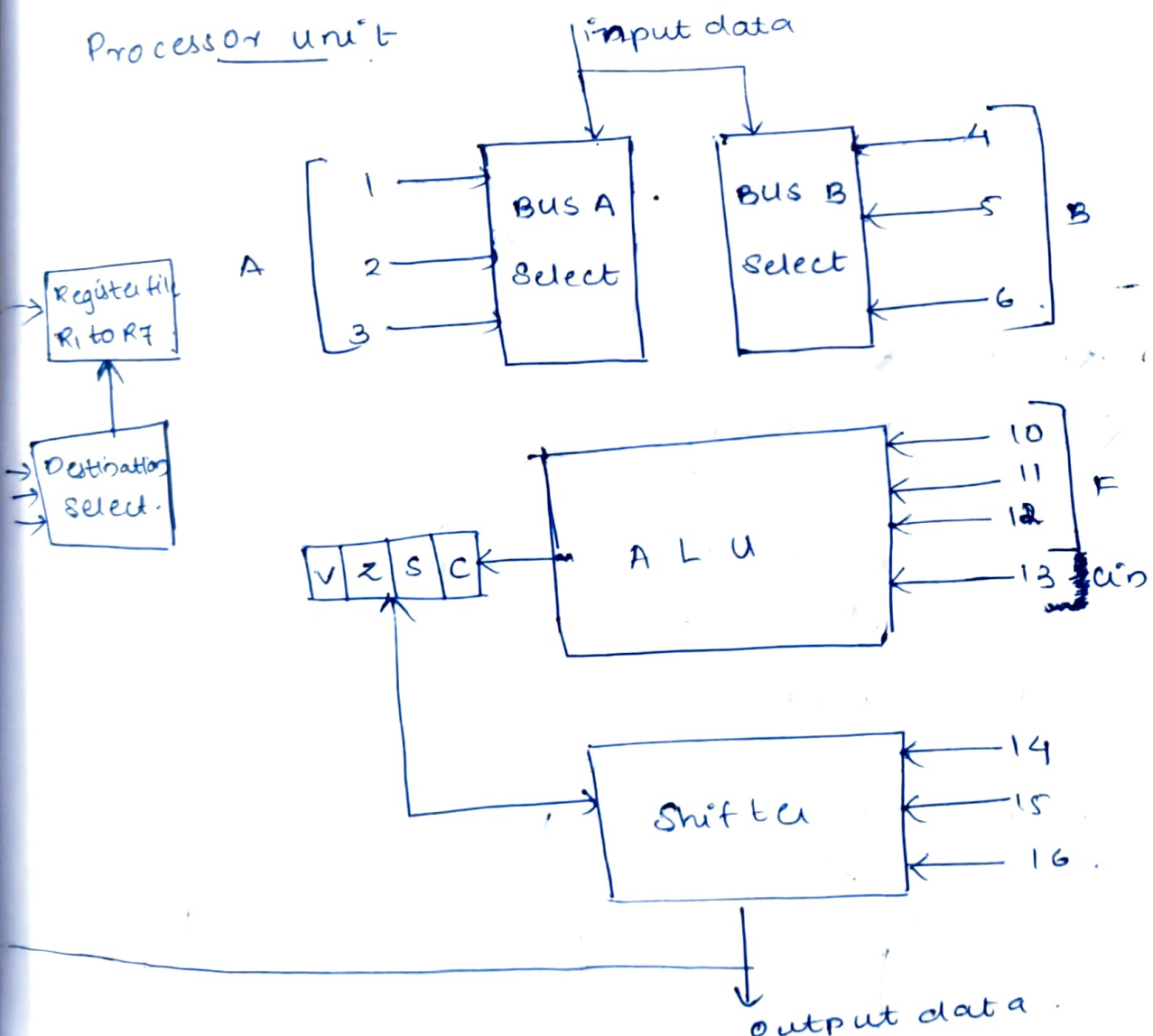
03/02/25 Design of Status Register

A status register is a hardware that contains the information about the state of the processor. It helps to find the relative magnitude of 2 registers. Status register consists of V, Z, S & C flags. C denotes carry flag. Carry flag is set if the o/p of the carry of ALU is one. S - Side flag. Side flag is set if the highest order bit of the result in the o/p is 1. ~~Zero flag~~. Z - Zero flag is set if the o/p of the ALU contains all zeroes.

V - Overflow → Overflow is set if the exclusive OR of C8 & C9 becomes 1.



Processor unit



Binary code	$C_{in}=0$	$C_{in}=1$	
0 0 0	$AC \leftarrow 0$	$A+1$	No shift.
0 0 1	$A+B$	$A+B+1$	shift right
0 1 0	$A-B-1$	$A-B$	~~shift left~~.
~~0 0 0~~	$A-1$	$A, C \leftarrow 1$	shift left.
1 0 0	$A \vee B$	—	0s to o/p bus
1 0 1	$A \oplus B$	—	—
1 1 0	$A \cap B$	—	circulate right with C
1 1 1	$\bar{A}$	—	circulate left with C

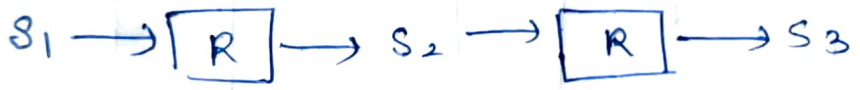
25/02/25

Pipelining

eg:	S_1	S_2	S_3
1.	$R_1 \leftarrow A_1, R_2 \leftarrow B_1$	—	—
2.	$R_1 \leftarrow A_2, R_2 \leftarrow B_2$	$R_3 \leftarrow A_1 * B_1$, $R_4 \leftarrow C_1$	—
3.	$R_1 \leftarrow A_3, R_2 \leftarrow B_3$	$R_3 \leftarrow A_2 * B_2$, $R_4 \leftarrow C_2$	$R_5 \leftarrow A_1 * B_1 + C_1$
4.	$R_1 \leftarrow A_4, R_2 \leftarrow B_4$	$R_3 \leftarrow A_3 * B_3$, $R_4 \leftarrow C_3$	$R_5 \leftarrow A_2 * B_2 + C_2$
5.	$R_1 \leftarrow A_5, R_2 \leftarrow B_5$	$R_3 \leftarrow A_4 * B_4$, $R_4 \leftarrow C_4$	$R_5 \leftarrow A_3 * B_3 + C_3$
6.	$R_1 \leftarrow A_6, R_2 \leftarrow B_6$	$R_3 \leftarrow A_5 * B_5$, $R_4 \leftarrow C_5$	$R_5 \leftarrow A_4 * B_4 + C_4$
7.	$R_1 \leftarrow A_7, R_2 \leftarrow B_7$	$R_3 \leftarrow A_6 * B_6$, $R_4 \leftarrow C_6$	$R_5 \leftarrow A_5 * B_5 + C_5$

8. [illegible]	$R_3 \leftarrow A_7 * B_7$, $R_4 \leftarrow C_7$	$R_5 \leftarrow A_6 * B_6 + C_6$
9. [illegible]		$R_5 \leftarrow A_7 * B_7 + C_7$

Pipelining is a process of dividing a process into several sub-operations where each sub operation is associated into segments. It is called pipelining because the output of S_1 is passed to S_2 & so on.



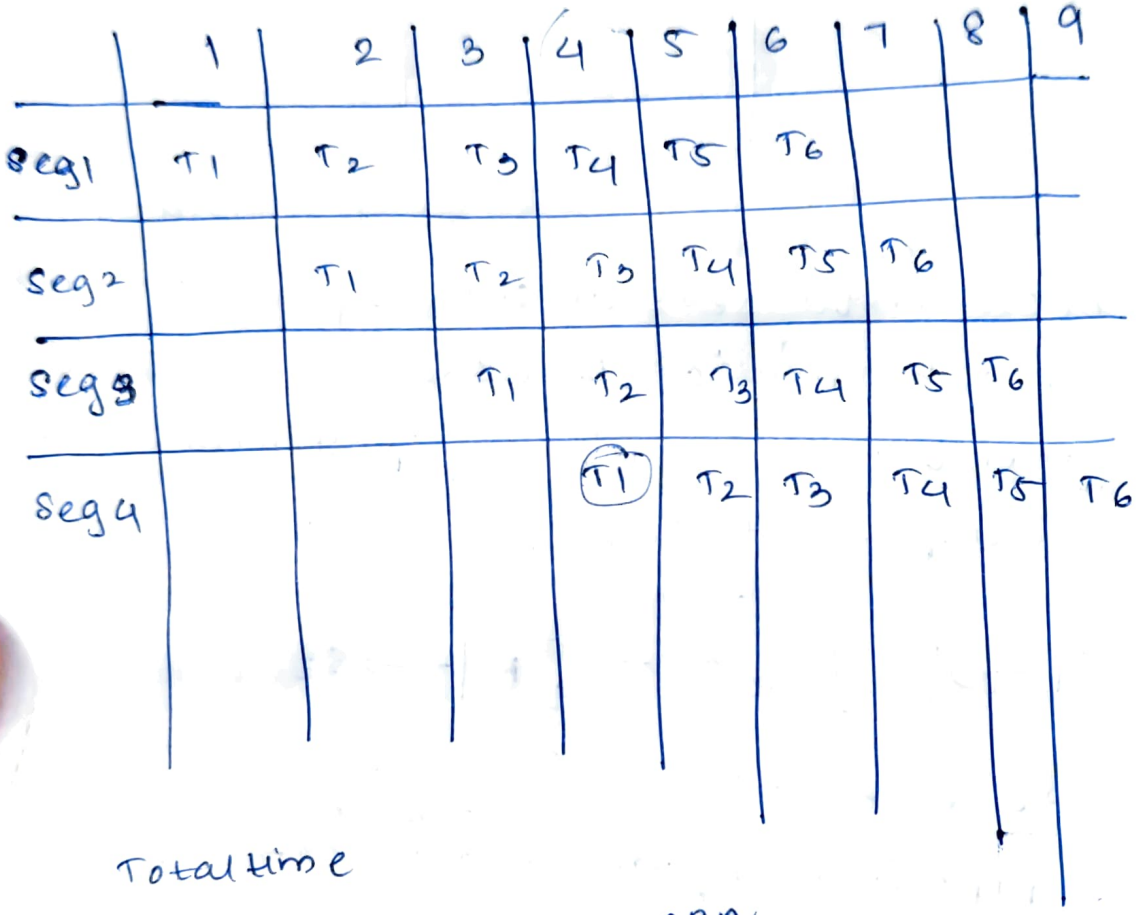
If we execute this, in sequential manner (non pipelining) for execute $A_i * B_i + C_i$ we need 3 segments & for each segment, it need one clock cycle.

Thus for 7 o/ps it needs 21 clock cycles.

Space-Time Diagram

It shows the behaviour of the pipeline.

Consider 6 different tasks are executed in 4 different segments.



	1	2	3	4	5	6	7	8	9
seg1	T1	T2	T3	T4	T5	T6			
seg2		T1	T2	T3	T4	T5	T6		
seg3			T1	T2	T3	T4	T5	T6	
seg4				T1	T2	T3	T4	T5	T6

Total time

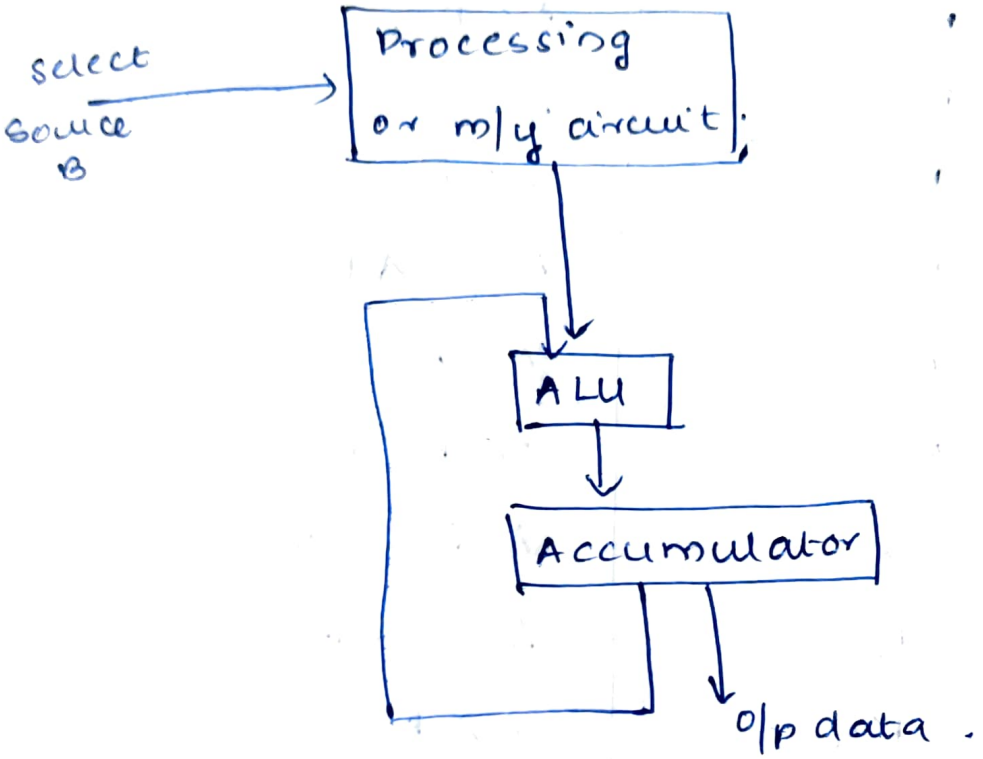
$$k t_p + (n-1) t_p$$

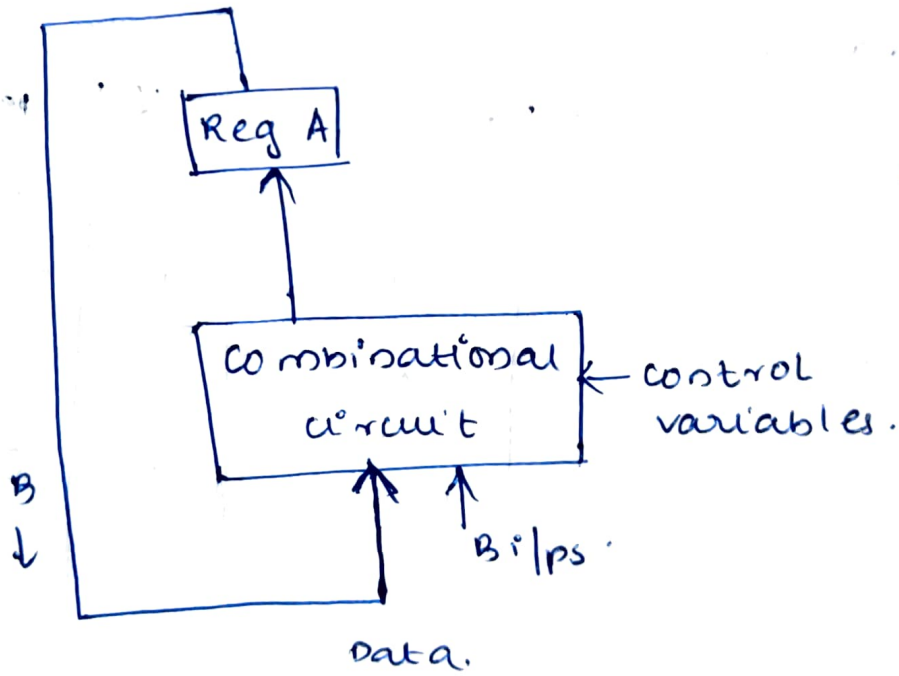
$$4 \cdot 4 + (6-1) \times 4$$

non

06/02/25 Design of Accumulator

Accumulator is a special purpose register capable of performing not only the addition but also many other μ operations.


select
source
B
Processing
or m/y circuit
ALU
Accumulator
o/p data.


Reg A
Combinational
circuit
control
variables.
B
B i/ps.
Data.

control variable	Function
P_1	$A \leftarrow A+B$
P_2	$A \leftarrow 0$
P_3	$A \leftarrow \bar{A}$
P_4	$A \leftarrow A \wedge B$
P_5	$A \leftarrow A \vee B$
P_6	$A \leftarrow A \oplus B$
P_7	$A \leftarrow shr A$
P_8	$A \leftarrow shl A$
P_9	$A \leftarrow A+1$
z	if $(A-0) \geq 1$

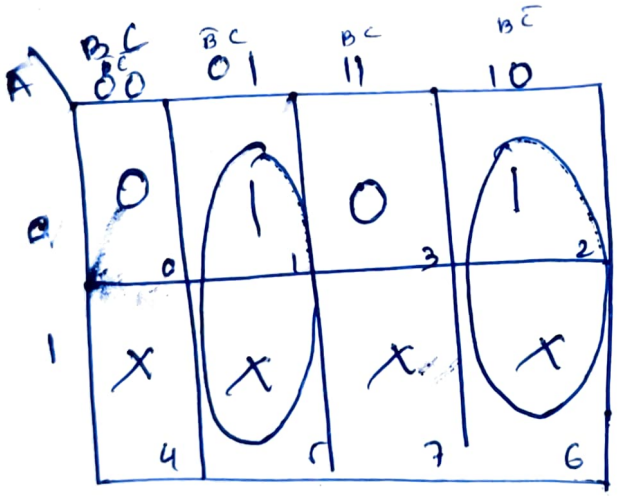
Add .

Present state	Next state	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

J	K	Next state
0	0	No change
0	1	Reset
1	0	set
1	1	Toggle

A	B	C	A_i	JA_i	KA_i	Q_{i+1}
0	0	0	0	0	X	0
0	0	1	1	1	X	0
0	1	0	1	1	X	0
0	1	1	0	0	X	1
1	0	0	1	X	0	0
1	0	1	0	X	1	1
1	1	0	0	X	1	1
1	1	1	1	X	0	1

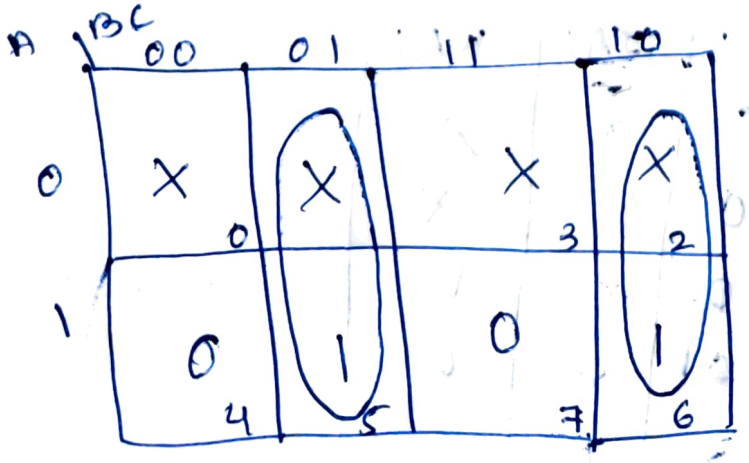
for JA_i



$JA_i = \bar{B} C P_i + B \bar{C} P_i$

~~$= B \oplus C$~~

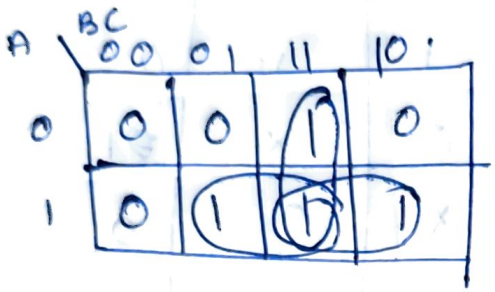
for KA_i



$KA_i = \bar{B} C P_i + B \bar{C} P_i$

~~$= B \oplus C$~~

for $C_i = 1$



$C_{i+1} = BCP_i + ACP_i + ABP_i$

Adding takes place only when P_1 is added.

For clear.

$JA_i = 0$.

$KA_i = P_2$.

Compliment

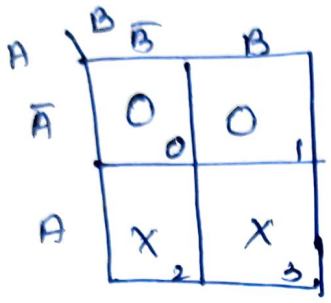
$JA_i = P_3$.

$KA_i = P_3$.

$A \leftarrow A \wedge B$

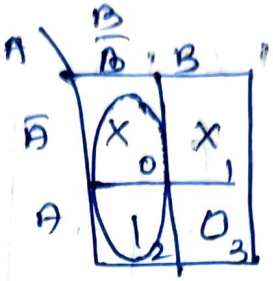
A	B	A_i	JA_i	KA_i
0	0	0	0	X
0	1	0	0	X
1	0	0	X	1
1	1	1	X	0

for JAi



$JA_i = 0$.

for KAi

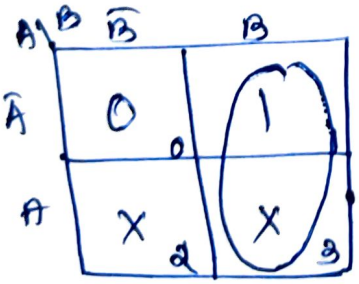


$KA_i = \bar{B}$ P4.

for OR

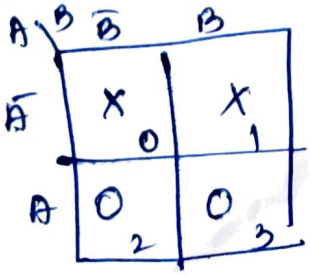
A	B	Ai	JAi	KAi
0	0	0	0	X
0	1	1	1	X
1	0	1	X	0
1	1	1	X	0

for JAi



$JA_i = B$ P5

for KAi

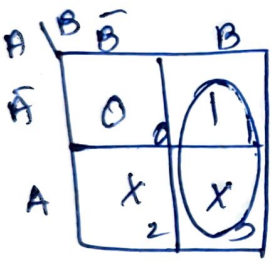


$KA_i = 0$

for XOR

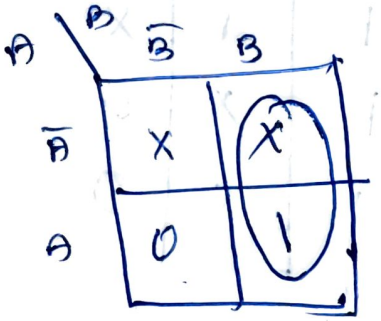
A	B	Ai	JAi	KAi
0	0	0	0	X
0	1	1	1	X
1	0	1	X	0
1	1	0	X	1

for JAi



$JA_i = BP_6$

for KAi



$KA_i = BP_6$