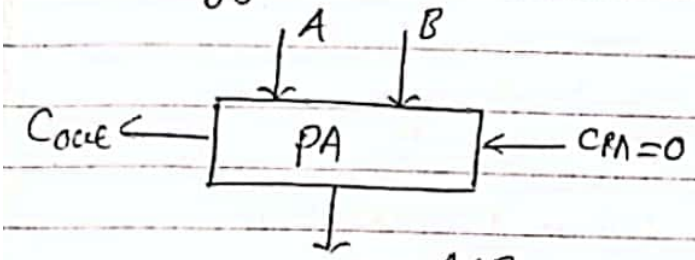
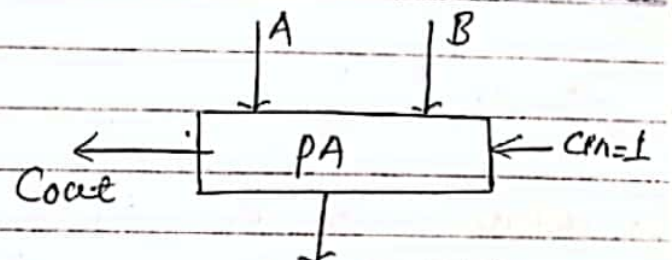
Design of Arithmetic circuits:

→ The basic component of the arithmetic section of an ALU is a parallel adder. A parallel adder is constructed with a number of full adder connected in cascade.

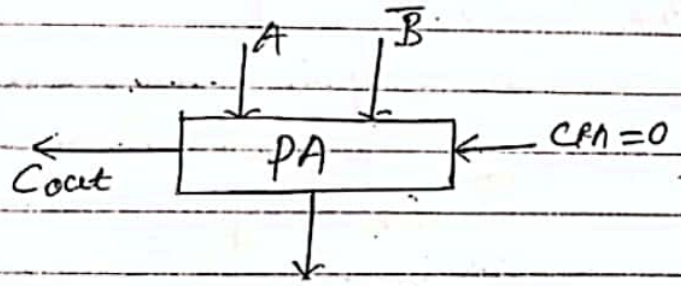
→ 5 different Arithmetic operations are:



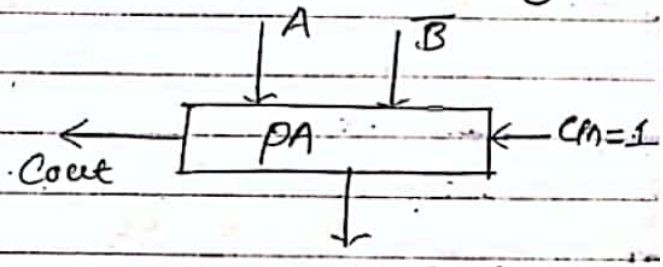
① Addition



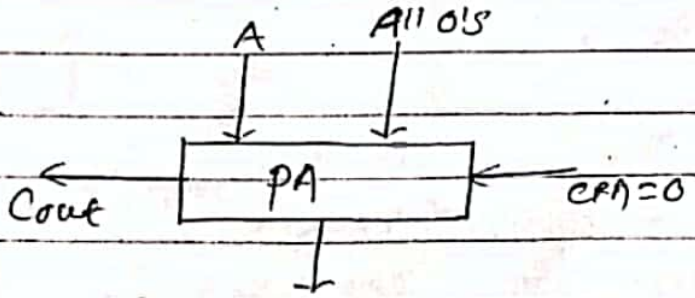
② Addition with carry.



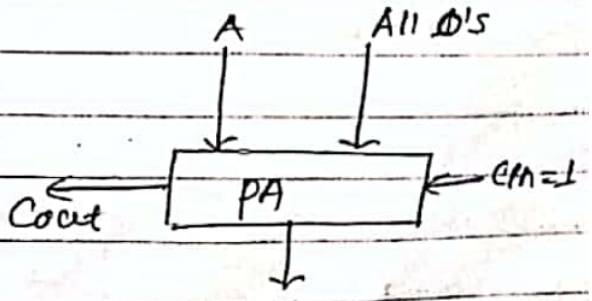
③ A plus 1's complement of B



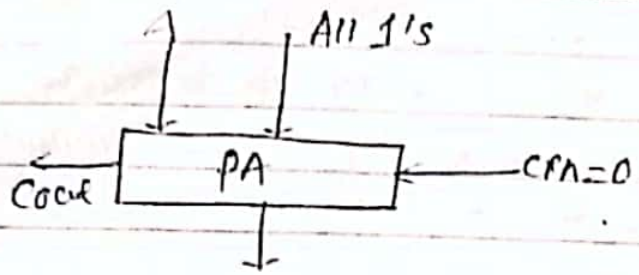
④ Subtraction



⑤ Transfer A

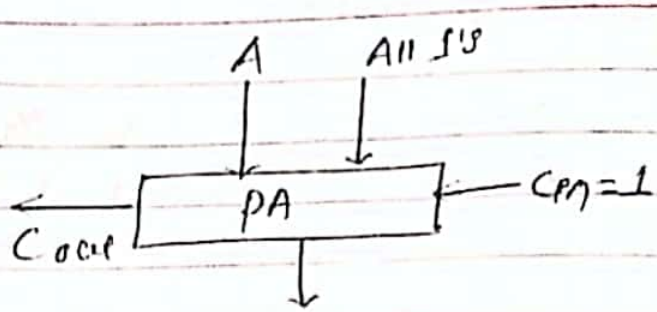


⑥ Increment A



F = A-1

(7) Decrement A

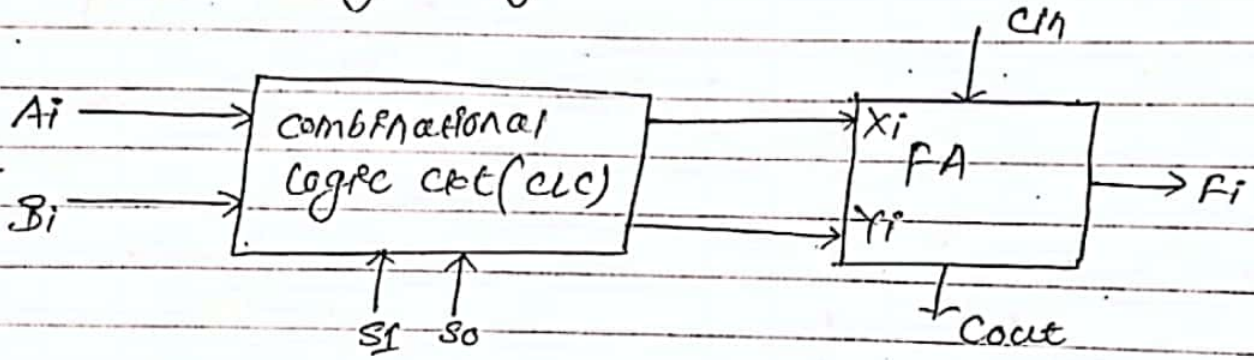


F = A

(8) Transfer A

→ Objective of arithmetic circuit design is to design a combinational circuit before the parallel adder. so that it can produce required F/p according to the select lines S1 & S0.

→ The block diagram of Arithmetic circuit is;



→ Function table:

Function select				
S1	S0	Cin	F	Remarks
→ 0	0	0	A	Transfer
0	0	1	A+1	Increment
→ 0	1	0	A+B	Addition
0	1	1	A+B+1	Addition with Cin = 1
→ 1	0	0	$A+\overline{B}$	A+ 1's comp. of B
1	0	1	$A+\overline{B}+1$	A-B
→ 1	1	0	A+ all 1's	A-1
1	1	1	A+ all 1's +1	A

→ Truth table for CLC fig 8:

I/P's				O/P's	
S1	S0	Ai	Bi	Xi	Yi
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	1	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	0	1	0	1
0	1	1	0	1	0
0	1	1	1	1	1
1	0	0	0	0	1
1	0	0	1	0	0
1	0	1	0	1	1
1	0	1	1	1	0
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	1	1
1	1	1	1	1	1

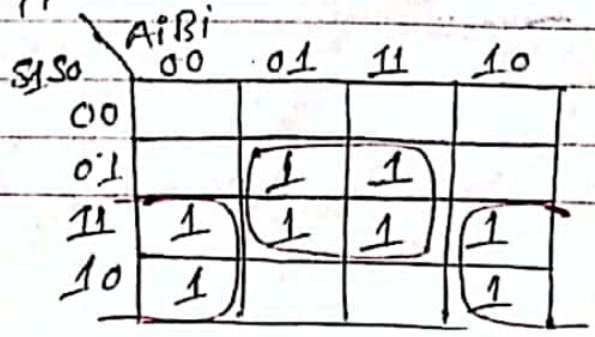
→ Using k-Map for, xi & yi

① for xi

from above truth table, we can see that, ~~the value~~

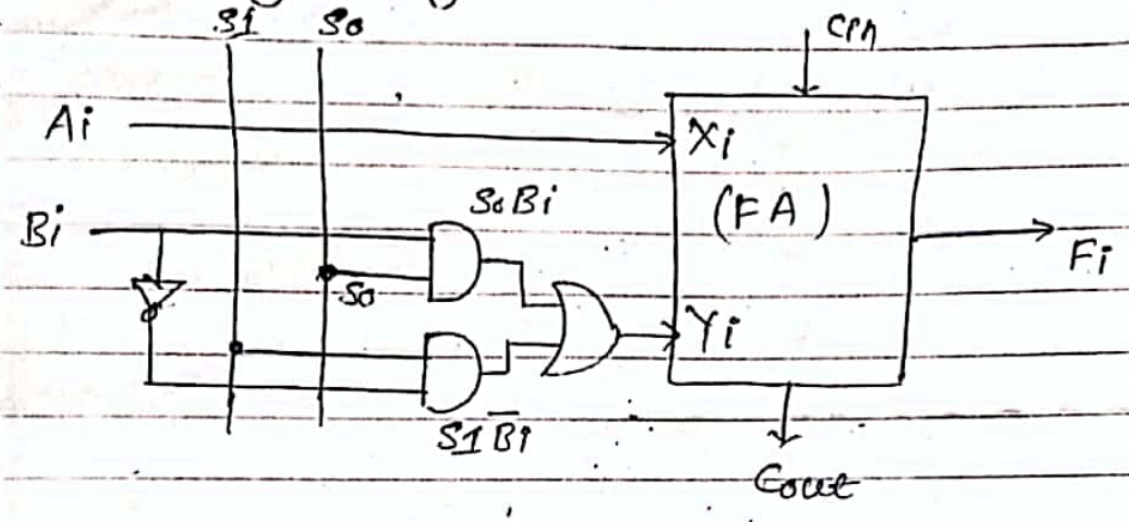
$\therefore X_i = A_i$

② for yi



$\therefore Y_i = S_0 B_i + S_1 \bar{B_i}$

→ The single stage arithmetic ckt is as;



→ Let the inputs A and B are both of 4-bit length, ie

$A = A_4 A_3 A_2 A_1$ &

$B = B_4 B_3 B_2 B_1$

then we draw the following logic diagram for ALU supporting various arithmetic operation on A & B.

→ Logic diagram :-

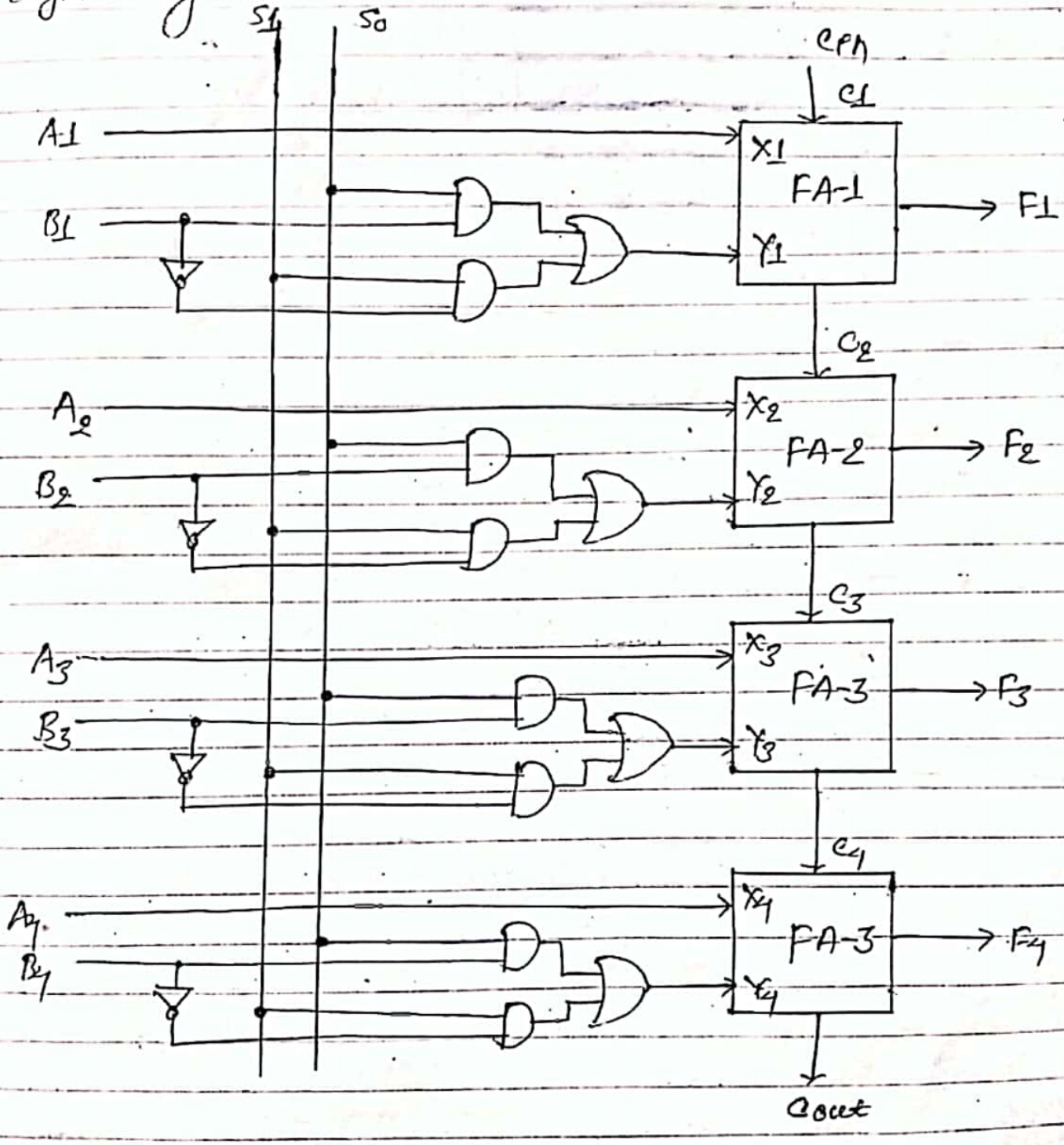


Fig: 4-bit logic diagram of Arithmetic ckt.

(8) Design an arithmetic circuit with two selection variables S1 and S0, that generate the following arithmetic operation. Draw the logic diagram of one typical stage.

S1	S0	$C_{in}=0$	$C_{in}=1$
0	0	$F=A+B$	$F=A+B+1$
0	1	$F=A$	$F=A+1$
1	0	$F=\overline{B}$	$F=\overline{B}+1$
1	1	$F=A+\overline{B}$	$F=A+\overline{B}+1$

A+B:

→ Function table

S1 S0 CIN	F
0 0 0	A+B
0 0 1	A+B+1
0 1 0	A
0 1 1	A+1
1 0 0	$A+\overline{B}$
1 0 1	$\overline{B}+1$
1 1 0	$A+\overline{B}$
1 1 1	$A+\overline{B}+1$

→ Truth table

S1 S0 Ai Bi	Xi Yi
0 0 0 0	0 0
0 0 0 1	0 1
0 0 1 0	1 0
0 0 1 1	1 1
0 1 0 0	0 0
0 1 0 1	0 0
0 1 1 0	1 0
0 1 1 1	1 0
1 0 0 0	0 1
1 0 0 1	0 0
1 0 1 0	0 1
1 0 1 1	0 0
1 1 0 0	0 1
1 1 0 1	0 0
1 1 1 0	1 1
1 1 1 1	1 0

→ k-Map

(i) for Xi

S1S0 \ AiBi	00	01	11	10
00			1	1
01			1	1
11			1	1
10				

$Xi = \overline{S1}Ai + \overline{S0}Ai$

$= Ai(\overline{S1} + \overline{S0})$

(ii) for Yi

B S0 \ AiBi	00	01	11	10
00		1	1	
01				
11	1			1
10	1			1

$Yi = S1\overline{B_i} + \overline{S1}\,\overline{S0}\,Bi$

→ One stage (single stage) Arithmetic logic diagram is,

$X_i = A_i(\overline{S_1} + \overline{S_0})$

$Y_i = S_1 \overline{B_i} + \overline{S_1}\,\overline{S_0}\, B_i$

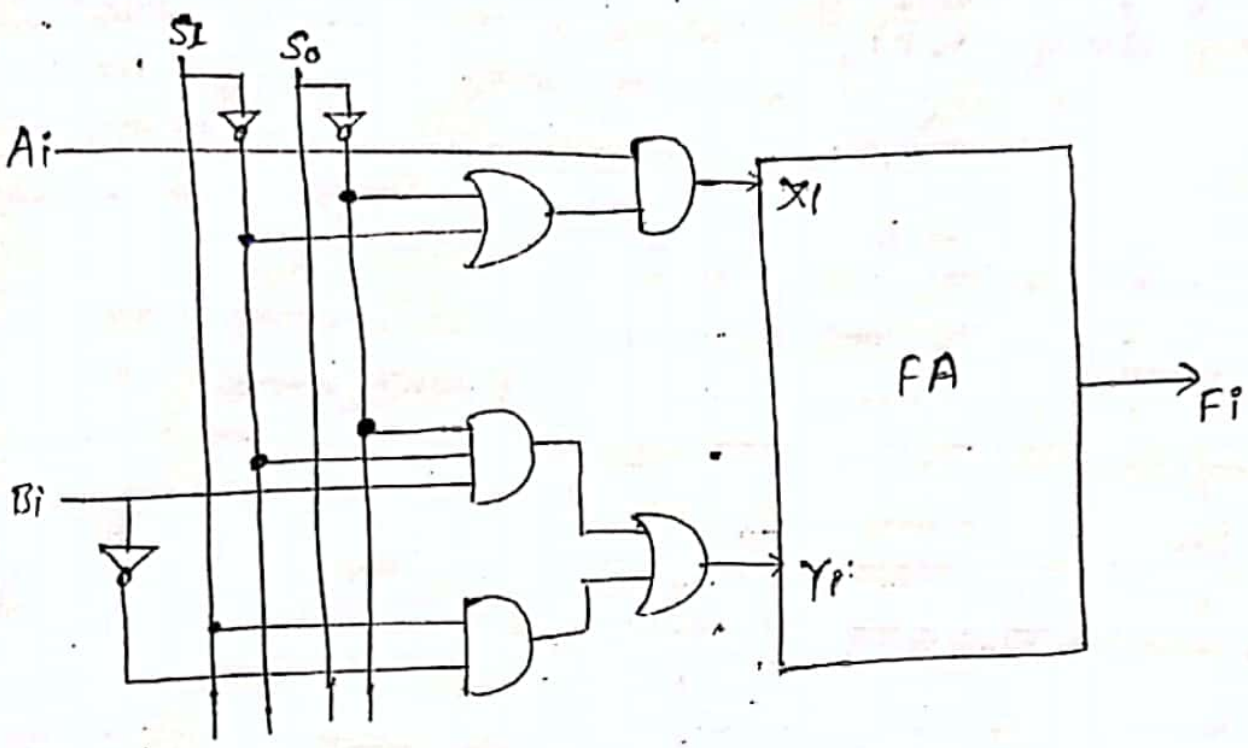


Fig: logic diagram of given arithmetic function table.

Ans (5) Design an arithmetic ckt with the two selection variables S_1 and S_0 that generates the following arithmetic operations. Draw the logic diagram of one typical stage.

S_1	S_0	$C_{in}=0$	$C_{in}=1$
0	0	$F=A$	$F=A+1$
0	1	$F=A-B-1$	$F=A-B$
1	0	$F=B-A-1$	$F=B-A$
1	1	$F=A+B$	$F=A+B+1$

Ans: $X_i = A_i(\overline{S_1} + S_0) + \overline{A_i}\,\overline{S_1}\,\overline{S_0}$; $Y_i = B_i S_1 + \overline{B_i}\,\overline{S_1} S_0$

Status Register

→ Status register (Flag register) is a group of flip-flop used to give status of different operation result.

→ It is connected to ALU.

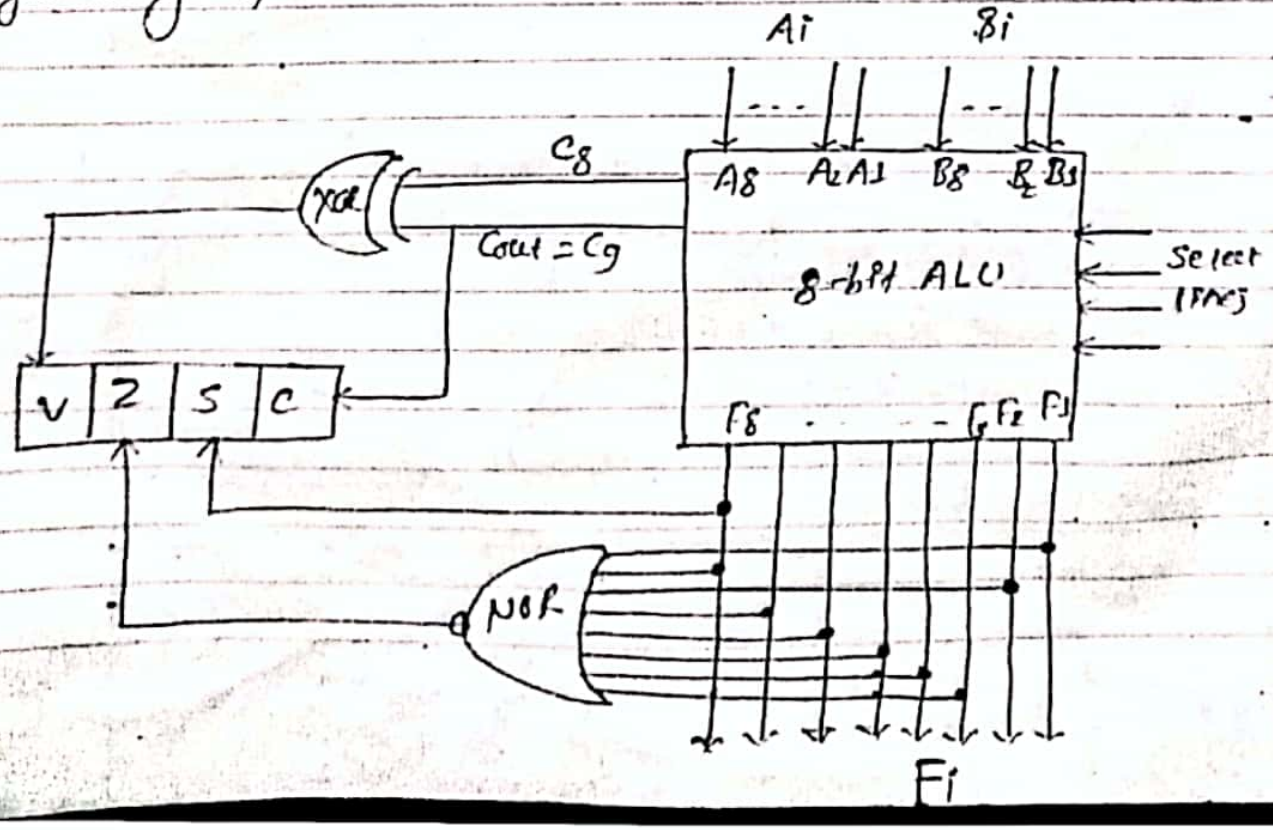
→ Once the operation is performed by ALU (Arithmetic logic unit), result is transferred on external data bus and status of result will be stored in flip-flop.

→ 4-flag register are used

(i) Carry flag (CF) → CF is set if carryout of MSB,
(ii) Sign flag (SF) → SF is set if result is negative,
(iii) Zero flag (ZF) → ZF is set if result is zero,
(iv) Overflow flag (OF) → OF is set if signed result is out of the range.

eg. for 8-bit ALU, OF (or V) is set if the result is greater than +127 and less than -128.

→ logic diagram



* ALU-Flag

V → overflow
Z → zero
S → sign
C → carry

* working operation

(1) Bit C is set if the o/p carry of the ALU is 1. It is cleared if the o/p carry is 0.

(2) Bit S is set if the heighest-order bit (MSB) of the result in the o/p of ALU is 1. It is cleared if the highest-order bit is 0.

(3) Bit Z is set if the o/p of ALU contains all 0's, and cleared otherwise. i.e.

Z = 1 if the result is zero, and Z = 0 if the result is non-zero.

(4) Bit V is set if the XOR of carries C_8 and C_9 is 1 otherwise V is cleared. This is the condition of overflow.

→ States bits after the subtraction of unsigned numbers (A - B):

Relation	condition of steted bit	Boolean function
A > B	C = 1 & Z = 0	CZ'
A ≥ B	C = 1	C
A < B	C = 0	C'
A ≤ B	C = 0 & Z = 1	C' + Z
A = B	Z = 1	Z
A ≠ B	Z = 0	Z'

a) Design an arithmetic circuit with two selection variables S_1 and S_0 that generates the following arithmetic operations. 8

S_1	S_0	$C_{in} = 0$	$C_{in} = 1$
0	0	F = A	F = A + 1
0	1	F = A-B-1	F = A - B
1	0	F = B – A.-1	F = B - A
1	1	F = A + B	F = A + B +1

b) Design a 3-bit synchronous gray code up counter using D flip-flop. 8

b) Design a 4-bit arithmetic circuits which performs eight different arithmetic operations.