Name: MD Abdul Aziz Section: M
ID : 22-47013-1

Answer to the question no-1

My student ID: 22-47013-1

so, a 3 bit adder/subtractor circuit is given below:

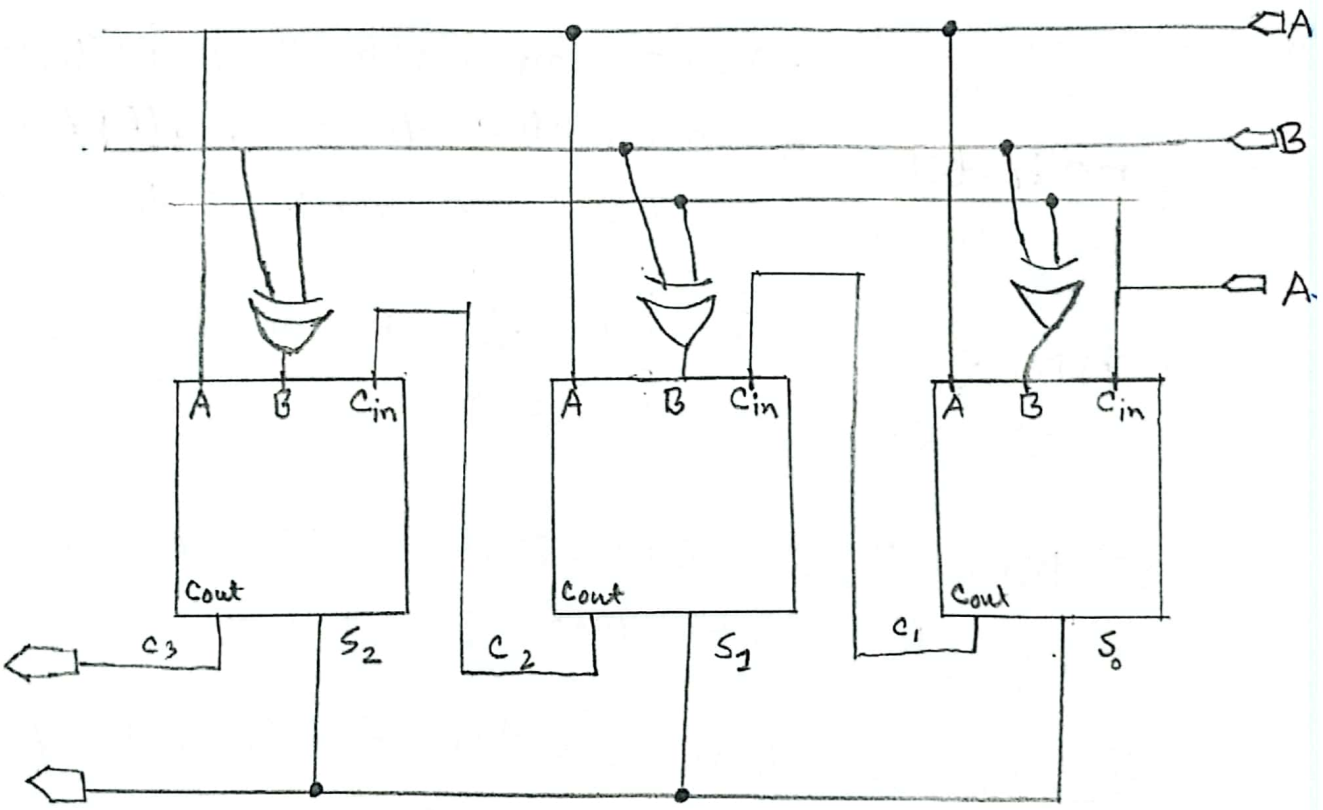


Fig: 3 bit adder/subtractor circuit

Ans to the question no-2

My student ID : 22-47013-1

Here, A = 0 , B = 1 , C = 3

$\therefore Y = A+B+C$
$= 0+1+3$
$= 4$

$\therefore$ A bus organization with Y=4 registers, each register is connected to 2 multiplexers to form 2 buses A and B to Perform $R_1 = R_2 - R_0 + 5$

(1) MUX A selector: to Place the contents of R_2 onto bus A.

(2) MUX B selector: to Place the contents of R_0 onto bus B

3) ALU function selector: to Provide the arithmetic operation A-B

(4) Shift selector: for direct transfer from the output of the ALU onto output bus S (no shift)

(5) Decoder Destination selector: to transfer the contents of bus S into R1.

(6) value Addition: Add the value 5 to the result. the operation is done before it is stored in R1.

Ans to the question no-3

Based on question number 2, the working Process scratch and a two-Port memory system is given below:

The operation: $R1 = R2 - R0 + 5$

In order to Perform the operation binary selection variables must be Provided to following sequence.

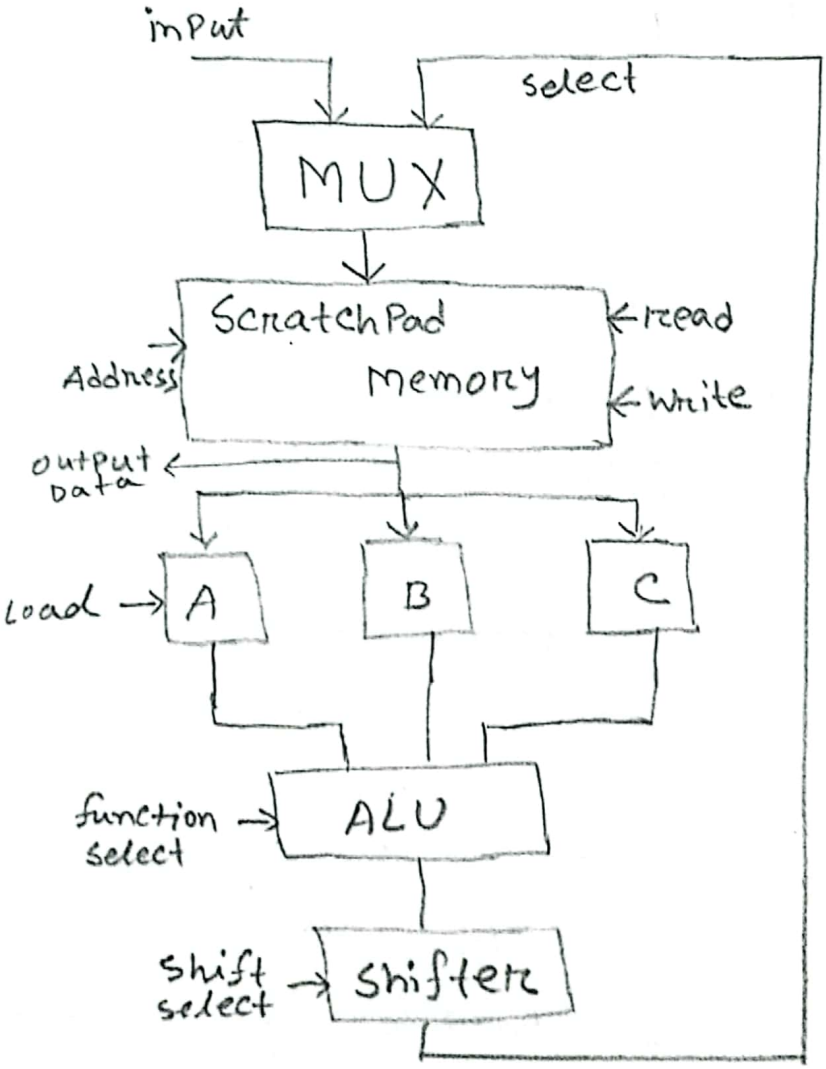


Fig: ScratchPad Memory

T1: $A \leftarrow M[010]$ read the contents R2 into register A

T2: $B \leftarrow M[000]$ read the contents R0 into register B

T3: $C \leftarrow A-B$ Performs the subtraction of the contents of A and B in the ALU storing the results in a temporary register C.

T4: $C \leftarrow C+5$ add the immediate value 5 to the contents of the temporary register C.

T5: $M[001] \leftarrow C$ transfer the final result from the temporary register C to register R1.

2-Port memory

When CP=1, A and B latches are open, accepting information from memory. The WE is also high state. A and B addresses are read from memory and placed into A and B registers.

When CP=0, the latches are closed, retaining the last data entered. The control unit provides the necessary signals to perform the subtraction operation in the ALU.

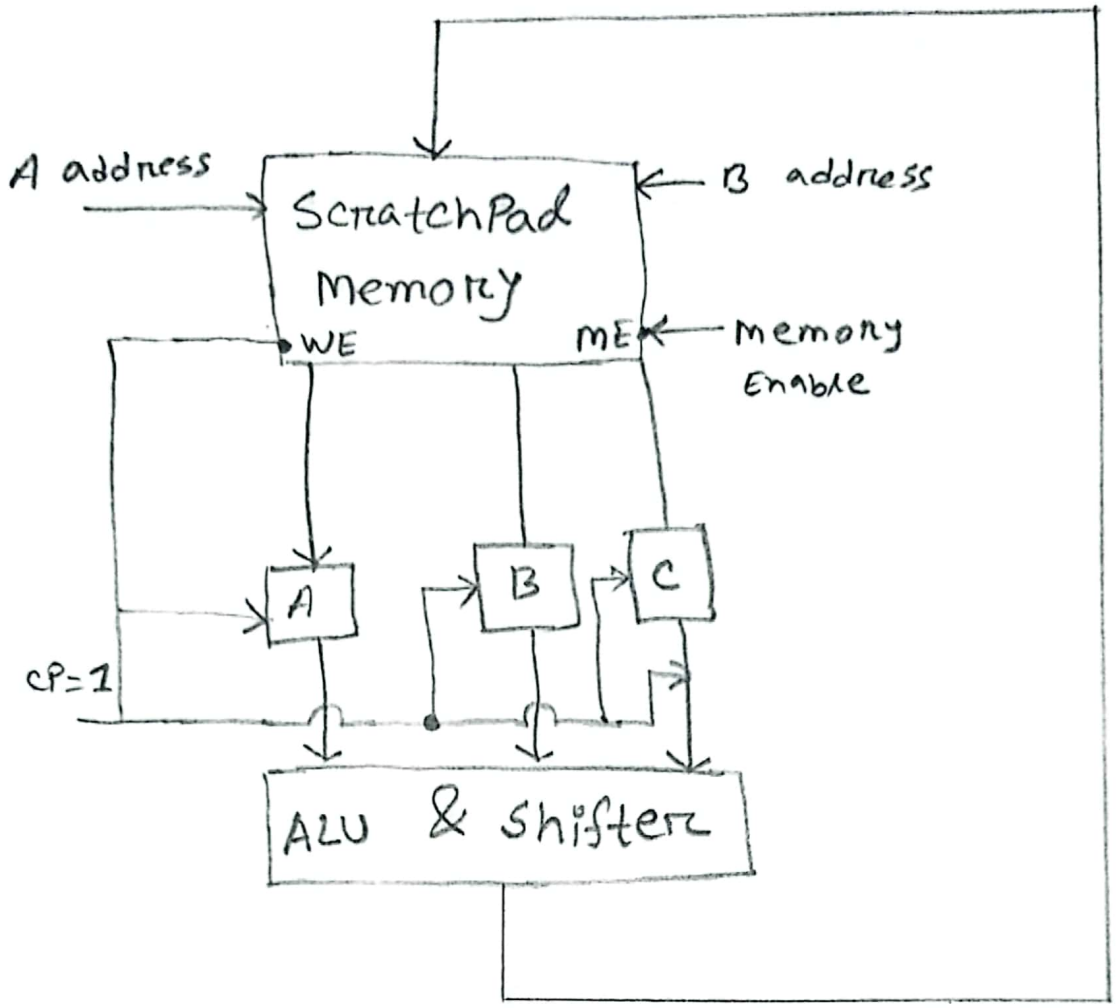


Fig: Processor unit with a 2-Port memory

C ← C+5, the control unit provides the necessary signals to add with the immediate value. when CP=0, the result of the micro-operation is written into memory word defined by B address and then the state of WE is LOW and ME is enabled. The necessary signals to store the result in R1 memory address (001) is provided by the control unit.

Answer to the question no-4

The control logic design process for signed number addition involves creating a complete system that performs binary addition while managing overflow conditions. The steps for a signed addition function is given below:

(1) Understanding Problem Statement: To understand the motive of designing control logic for adding two signed binary numbers and to handle the overflow.

(2) Defining Inputs and outputs: Input includes two signed binary numbers and that results in the result of the operation while an overflow flag also can be generated.

(3) Adder circuit Designing: The inputs to be added by a n-bit adder.

(4) Overflow Detection: There are some scenerios when the overflow occurs such as if both operands are positive, but the result is negative or both operands are negative but the results are positive.

(5) Control signal Implementation: The inputs must be selected. overflow flag should trigger when overflow happens

(6) verification: The accuracy of the sum is verified.

(7) Hardware Implementation: Hardware must be implemented with proper apparatus and connections.

(8) Performance Analysis: Timing analysis must be conducted to ensure the design operates within required time.

Ans to the question no - 5

The control logic design Process for signed number subtraction involves creating a complete system that Performs binary subtraction. The step by step explanation of the control logic design for signed subtraction is given below:

(1) Understanding the Problem Statement:
to understand the motive of designing control logic for subtracting two signed binary numbers.

(2) Defining Inputs and outputs:
Input(A,B) includes two signed binary numbers and that is ~~shown in the~~ resulted in the result of the operation while an overflow flag also can be generated. The output would be, $R = A - B$

(3) Subtraction Process: subtraction is converted to addition by taking the two's complement of B. subtract B from A by calculating two's complement of B (A+)

(4) Adder Circuit Design: A n-bit adder is used to add A and the two's complement of B.

(5) Overflow Detection: There are some scenerios when the overflow occurs such as if the result exceeds the representable range or if the minuend and the subtrahend have different signs but the result has the wrong sign.

(6) Control sign implementation: The inputs must be selected, controls should be enabled when the operation starts and signal overflow must be monitored.

(7) Simulation: subtraction operation is simulated and tested with different inputs.

(8) Hardware Implementation: The logic is implemented in HDL and to verify functionality, the design is mapped.

(9) Timing Analysis: Timing analysis is conducted to ensure the design operates within required time.

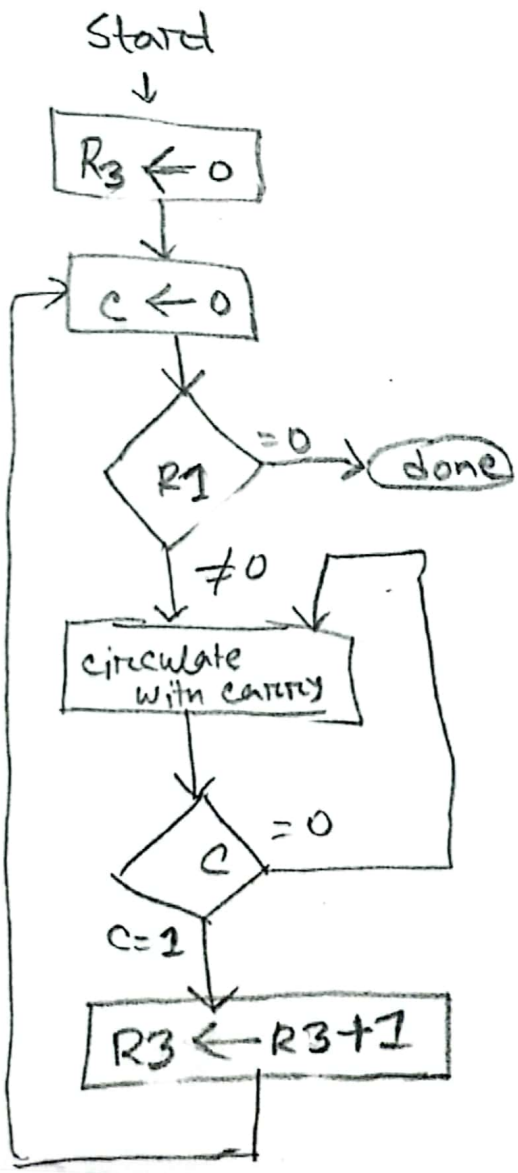
Ans to the question no-6

My student ID: 22-47013-1

$\therefore A = 1 \quad , \quad B = 3$

Register R_1 contains a binary value ~~(R_1=00000001)~~ ($R_1 = 00000001$) and R3 will store the count

Flow Chart:



Microinstruction table of 1's

R1 (Binary)								C	R3 (D)
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	1	1

Ans to the question no - 7

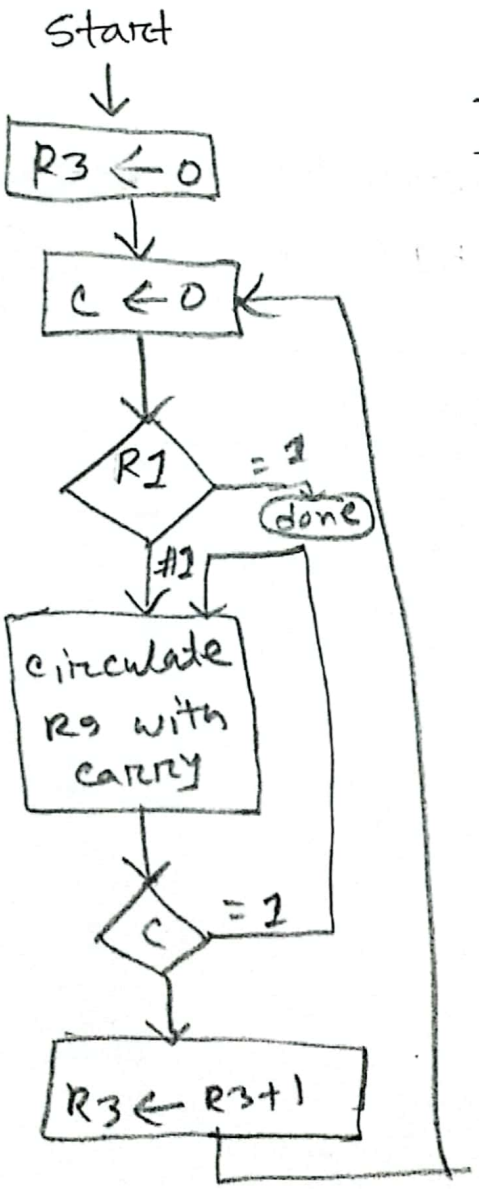
My student ID: 22-47013-1

Here, A = 1, B = 3

Register R1 contains binary value (R1 = 00000001)

Register R3 will store the count of 0's.

Flowchart:



Microinstruction table for 0's

R1 (Binary)								C	R3 (D)
0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	1	0
1	1	0	0	0	0	0	0	0	1
1	1	1	0	0	0	0	0	0	2
1	1	1	1	0	0	0	0	0	3
1	1	1	1	1	0	0	0	0	4
1	1	1	1	1	1	0	0	0	5
1	1	1	1	1	1	1	0	0	6
1	1	1	1	1	1	1	1	0	7

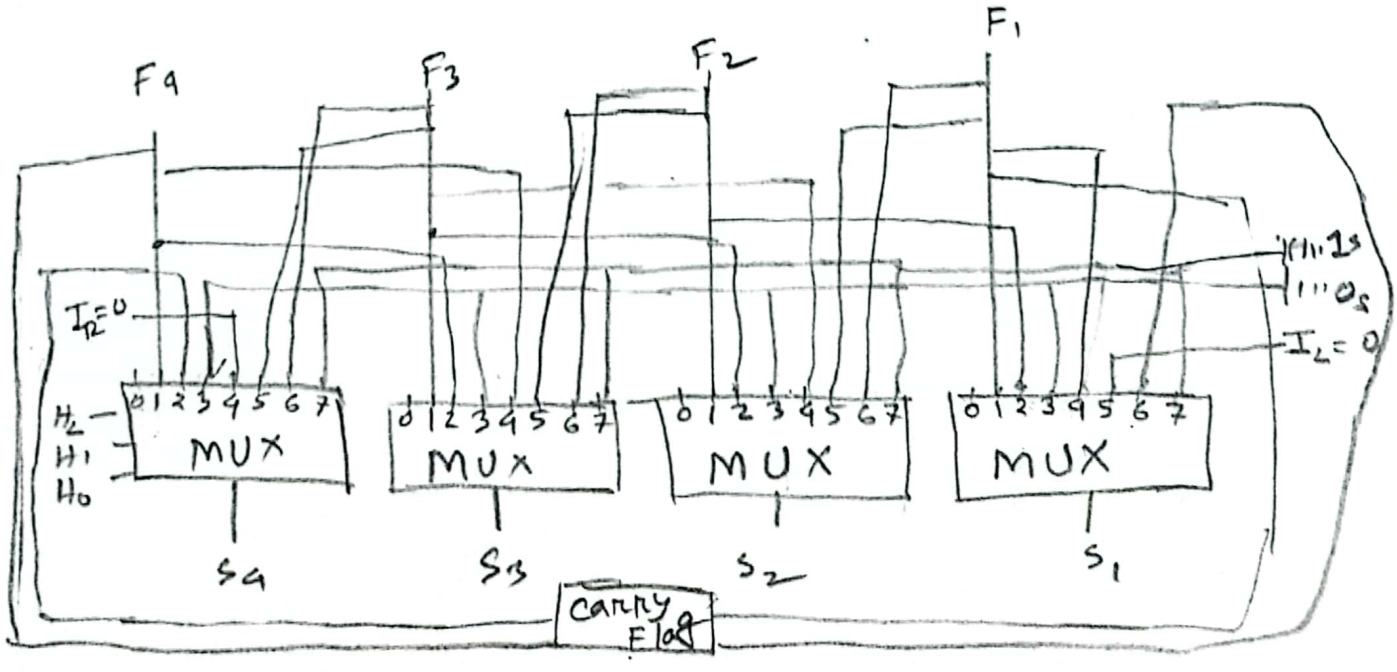
Ans to the question no-8

The Processor Unit with 16-bit control word variable is shown below in the table:

Microoperation	1 2 3	4 5 6	7 8 9	10 11 12	13	14 15 16	
	A	B	D	F	C_{in}	H	Hexa
$R_2 \leftarrow SHL$ (R3 OR R6)	011	110	010	110	0	101	$(7965)_{16}$
R4 ← 1	000	000	100	000	0	111	$(1217)_{16}$
R3+R5+1	011	101	000	010	1	000	$(7428)_{16}$
R4 ← input	000	000	100	000	0	000	$(1211)_{16}$
$R_4 \leftarrow CLCR2$	010	000	100	100	0	110	$(4246)_{16}$

Ans to the question no-9

4-bit shifter design:



Answer to the question no - 10

S_2	S_1	S_0	C_{in}	Function	X_i	Y_i	Z_i
0	0	0	0	A-1	A	1	0
0	0	0	1	A	A	1	1
0	0	1	0	A-B-1	A	$\overline{B}$	0
0	0	1	1	A-B	A	$\overline{B}$	1
0	1	0	0	A+B	A	B	0
0	1	0	1	A+B+1	A	B	1
0	1	1	0	A	A	0	0
0	1	1	1	A+1	A	0	1
1	0	0		$\overline{A}$	A	1	0
1	0	1		$A \oplus B$	A	B	0
1	1	0		$A \cup B$	A+B	0	0
1	1	1		$A \cap B$	$A+\overline{B}$	$\overline{B}$	0

K-maps

(X_i)

S_2S_1 \ S_0	0	1
00	A	A
01	A	A
11	A+B	A+B
10	A	A

$X_i = A_i + S_2 S_1 \overline{S_0} B_i + S_2 S_1 S_0 \overline{B_i}$

(Y_i)

S_2S_1 \ S_0	0	1
00	1	$\overline{B}$
01	B	0
11	0	$\overline{B}$
10	1	B

$Y_i = \overline{S_2}\,\overline{S_1}\,\overline{B_i} + \overline{S_2}\,\overline{S_0} B_i + S_2 S_0 B_i \overline{B}$
$+ S_2 \overline{S_1} B_i$

(Z_i)

S_2S_1 \ S_0	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	0	0	0	0
10	0	0	0	0

$Z_i = \overline{S_2} C_{in}$

4-bit ALU Design

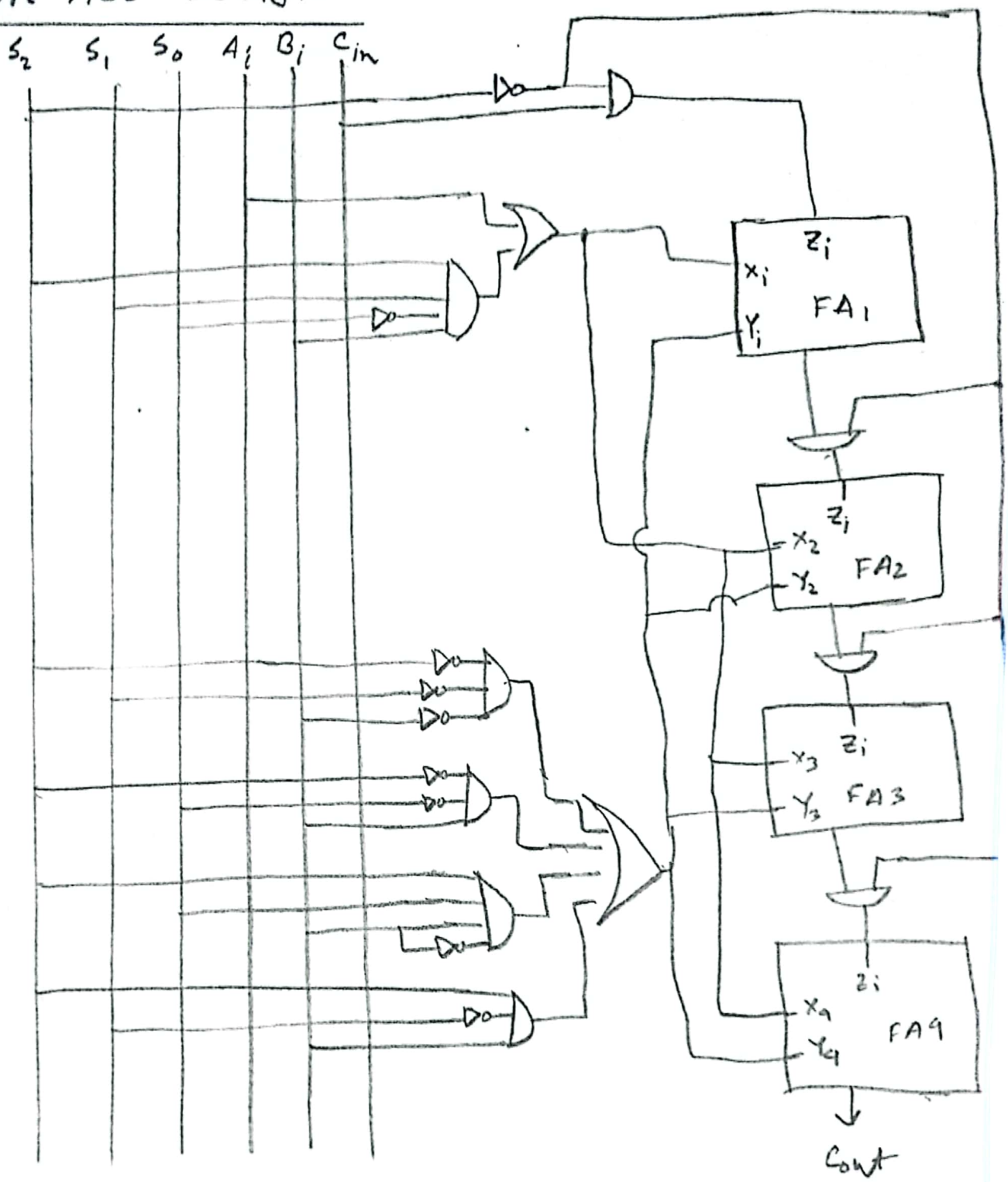


Fig: 4 bit ALU design