Exam Center Code: 205

PU Registration No: 2019-1-18-0070

Exam Roll No: 20180023

Level: Bachelor

Program: Software

Semester: Second (II)

Subject: Logic Circuit

Signature of Student: prashant

Date of Examination: 31/08/2021

Q.N 1; I prefer digital system because.

i. It made possible many scientific, industrial and commercial advances that would have been unattainable otherwise.

ii. The space program will not be possible, industrial enterprises function will not be efficient if those computers were not have invented.

iii. Many complex graphical simulation of gravitational forces cannot be implemented on regular analog system.

iv. Programs / automated programs cannot run through analog system unless digital system converts that process in the signals of 0,1.

v. Many datacenter rely on this digital system because it is much more powerful than analog system.

1. numerical part (23 symbol no last)

I. $(2335.DD)_{16} = ()_5$

$= 2 \times 5^{2} + 3 \times 5^{1} + 5 \times 5^{0} + 13 \times 5^{-1} + 13 \times 5^{-2}$

$= 73.12$

II: $(F@@2)_{16} = (?)_7$

$(F232)_{16} = (?)_7$.

$\Rightarrow F \times 7^{3} + 2 \times 7^{2} + 3 \times 7^{1} + 2 \times 7^{0}$

$\Rightarrow 5145 + 99 + 21 + 2$

$\Rightarrow (5267)_7$.

2 ans: In Half adder, it performs the addition of two bits i.e,

0+0 = 00
0+1 = 01
1+0 = 01
1+1 = 10

But in full adder it performs the sum of 3 inputs and outputs

truth table.

x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Kmap for S:

x \ yz	00	01	11	10
0		1		1
1	1		1	

$$S = x'y'z + x'yz' + xy'z' + xyz$$

$$= z(x'y' + xy) + z'(x'y + xy')$$

$$= z(x \odot y) + z'(x \oplus y)$$

$$= z(x \oplus y)' + z'(x \oplus y)$$

$$= x \oplus y \oplus z.$$

K map for C.

x \ yz	00	01	11	10
0			1	
1		1	1	1

$$C = xz + xy + yz$$

$$= xy + z(x + y)$$

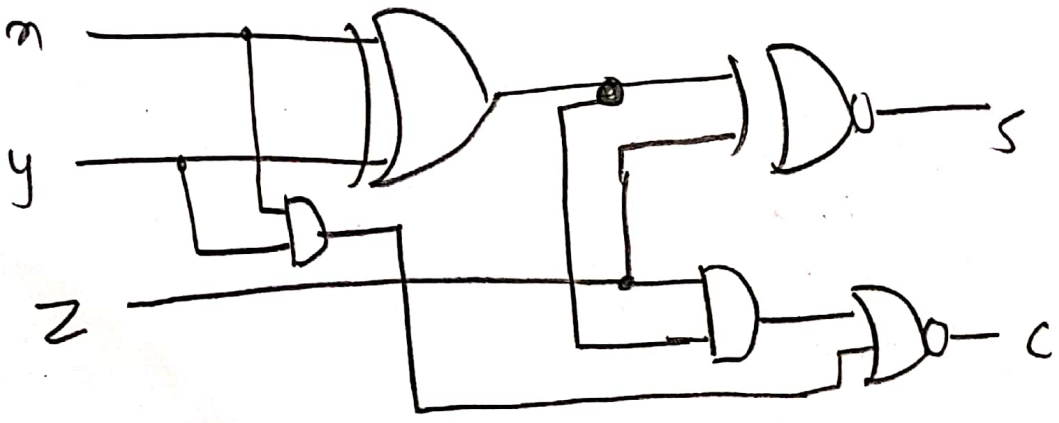


fig: circuit diagram of full adder.

A binary adder-subtractor can do such operation of n-bit data which is combination of many full adders.

To perform addition it simply adds two numbers.

To perform subtraction it adds 1st binary number to 1's complement of 2nd number.

For example.

Let A = 9, B = 7

A = 1001, B = 0111

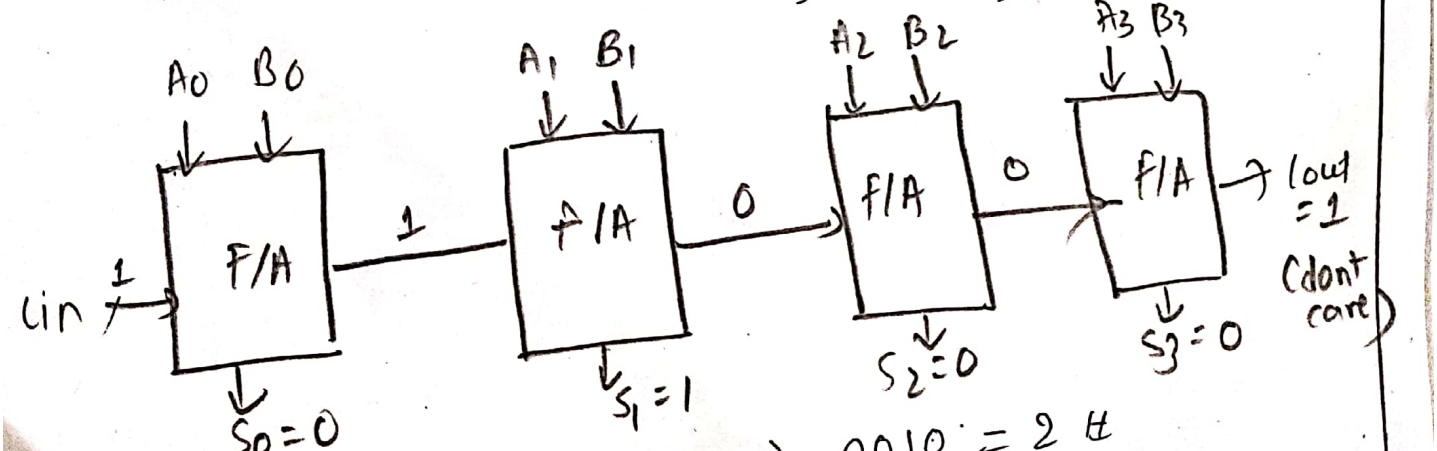
Adding = A + B.

In this operation, I only show subtraction using full adder.

subtraction = $A + \bar{B} + 1$

$= 1001 + 1000 + 1$

$A_3 A_2 A_1 A_0 \quad \bar{B}_3 \bar{B}_2 \bar{B}_1 \bar{B}_0$



Result = $S_3 S_2 S_1 S_0$ => 0010 = 2 #

3.

Ans: The major differences between ROM and PLA are:

ROM	PLA
1. It is a Read Only Memory.	1. It is ~~toH~~ full form is Programmable logic array.
2. ROM cannot take dont care conditions	2. PLA can take dont care condition.
3. It provides full decoding.	3. It ~~p~~ do not ~~provide~~ provide full decoding
4. It is made with a decoder and OR gate	4. It is made up of AND and OR gate.

My symbol no 20180023

So, F = (0, 1, 2, 3, 8)

we have to use $2^3 * 1$ Mux ie 8*1 MUX

Implementation in Mux as:

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
$\bar{A}$	(0)	(1)	(2)	(3)	4	5	6	7
A	(8)	9	10	11	12	13	14	15

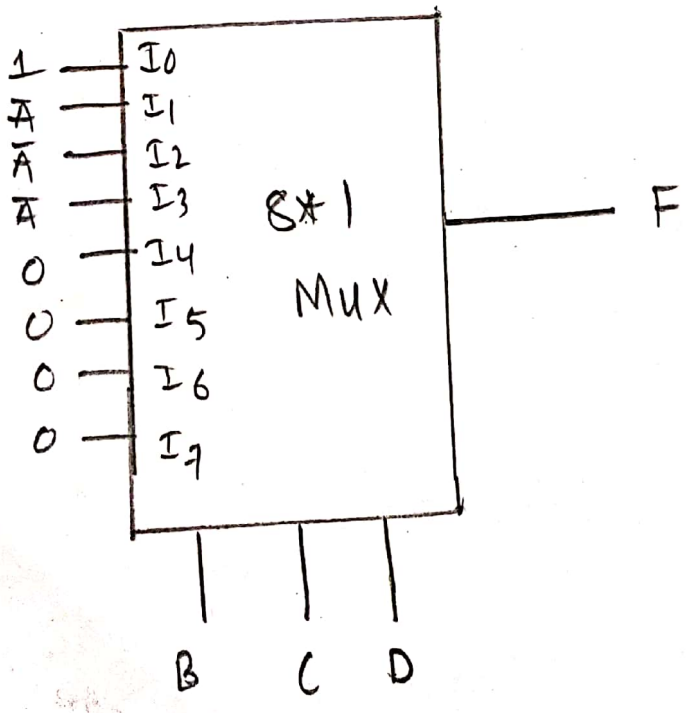
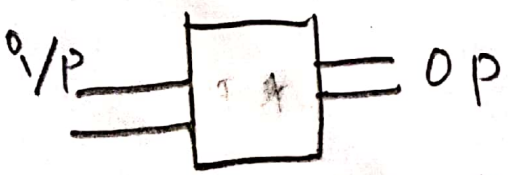


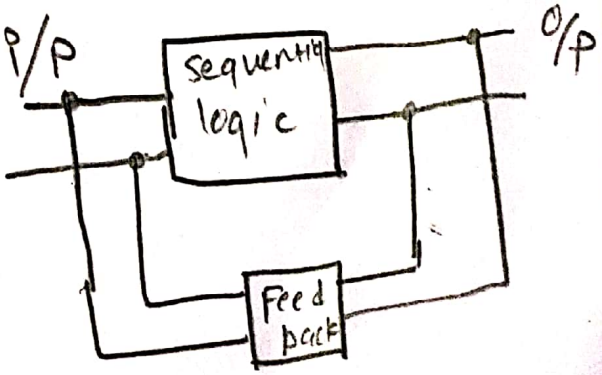
fig: Implementation of my symbolno at MUX

4.

Ans: The difference between combinational Logic and Sequential logic circuit.

Combinational L.C	Sequential L.C.
1. It has one or more inputs and one or more outputs.	1. It has one or more inputs and one or outputs in sequential order.
2. Its output depend only on present input value	2. It's output depend on present input and past output.
3. It doesn't have memory elements	3. It have memory element.
4. It doesn't have feedback path	4. It has feedback path.
5. Example: Adder, Subtractor, MUX, demux, etc	5. Example: flipflop, counter, shift register etc.





In RS flipflop.

S	R	Q	$\bar{Q}$	Remarks.
1	0	1	0	set
0	0	1	0	nochange
0	1	0	1	Reset
0	0	0	1	no change
1	1	0	0	Invalid

In when both R and S are inputed it acts as invalid condition in RS flipflop.

Such flipflop fault can be overcomed by J-K flip flop as it do not give any invalid inputs when clock pulse is applied when latching is done.

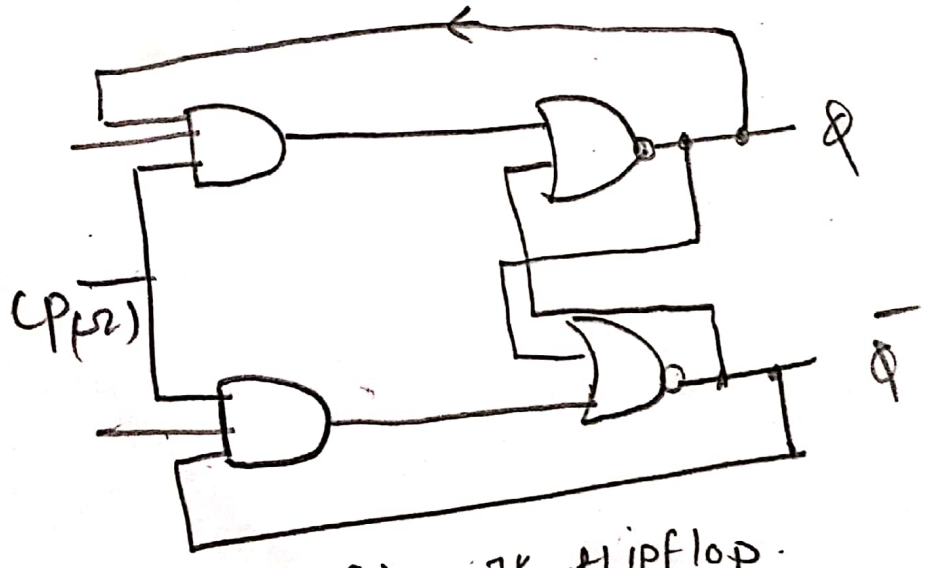


fig: JK flipflop.

5.

ans.. Shift registers are the sequential logic circuit that can be used to store or transfer data which are in binary form.

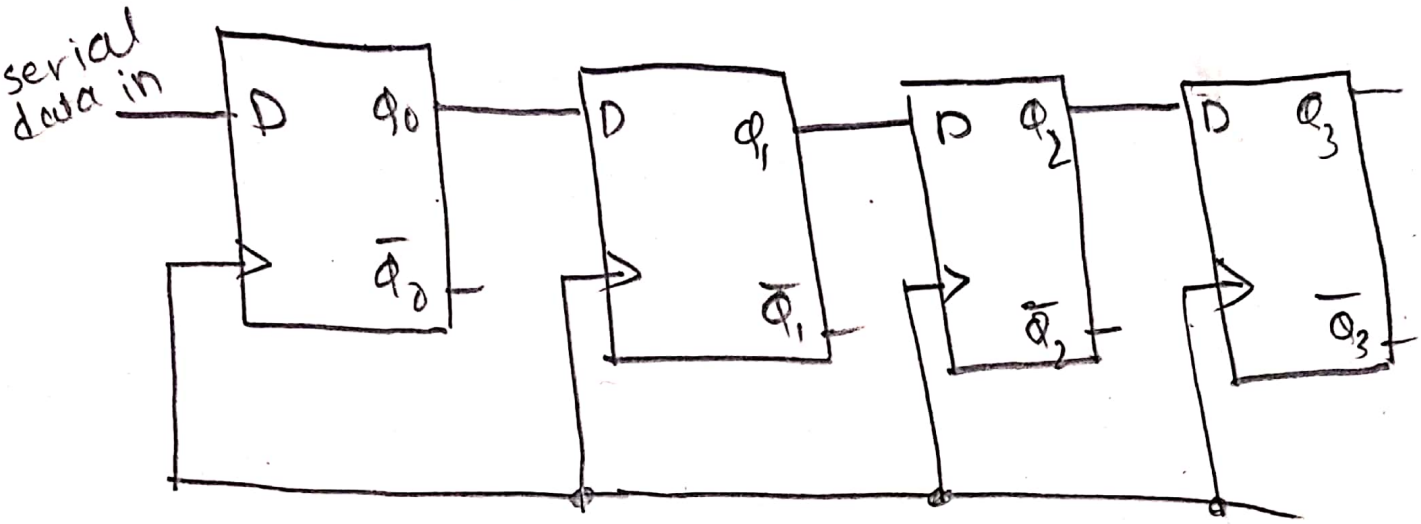
It consist of chain of flipflops which outputs of the flipflop is connected to input of next flip flop.

i) Serial in serial out (SISO)

In this shif register, it accepts the data bit serially ie, one bit at a time on a single line. It produces the stored information on its output also in serial form.

Let us consider the 4 bit SISO shift register. for example: data input is 1101

The data enter serially from LSB (least significant bit)



Operation: Initially, all flipflops are reset ie: $Q_0 = 0$, $Q_1 = 0$, $Q_2 = 0$, $Q_3 = 0$

Data input is 1101

At Cp 1, $Q_0 = 0$, $Q_1 = 0$, $Q_2 = 0$, $Q_3 = 0$

At Cp 2, $Q_0 = 1$, $Q_1 = 0$, $Q_2 = 0$, $Q_3 = 0$

At Cp 3, $Q_0 = 0$, $Q_1 = 1$, $Q_2 = 0$, $Q_3 = 0$

At Cp 4, $Q_0 = 1$, $Q_1 = 0$, $Q_2 = 1$, $Q_3 = 0$

At Cp 5, $Q_0 = 1$, $Q_1 = 1$, $Q_2 = 0$, $Q_3 = 1$

11) Parallel in serial out (PISO).

In PISO shift register, data are entered simultaneously (all bit at a time) and output taken serially (one bit at a time). It needs only 4 clock pulse to complete 4 bit operation.

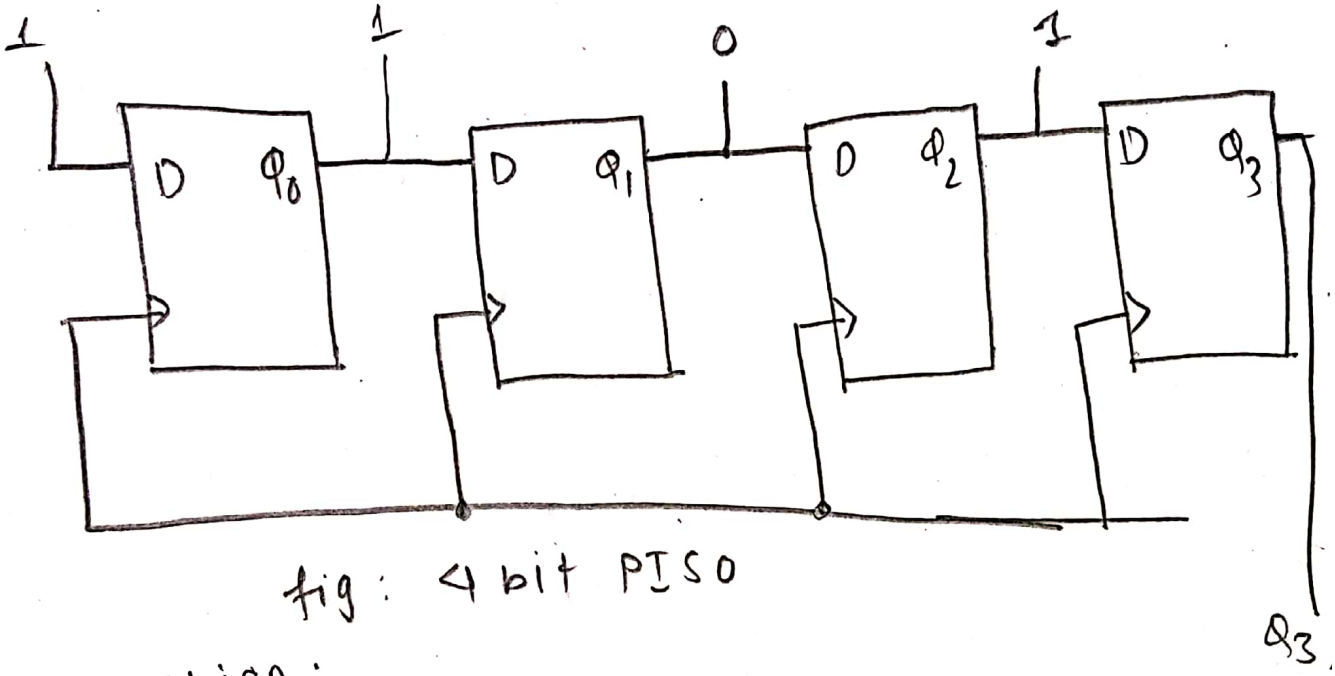


fig: 4 bit PISO

operation:

Initially all flipflop are closed i.e, $Q_0 = 0$, $Q_1 = 0$, $Q_2 = 0$, and $Q_3 = 0$.

At cp1, $Q_0 = 1$, $Q_1 = 1$, $Q_2 = 0$, $Q_3 = 1$

At cp2, $Q_1 = 1$, $Q_2 = 1$, $Q_3 = 0$

At cp 3, $Q_2 = 1$ $Q_3 = 1$

At cp 4, $Q_3 = 1$

6. For educational purpose if i were suggested to make a processor using following

(i) Logic circuit

(ii) Arithmetic circuit.

Design of logic circuit.

logic microoperations to manipulate bits of oprands separately and treat each bit as a binary number.

A simple design of logic for

S_1	S_0	output	Operation.
0	0	$F_i = A_i + B_i$	OR
0	1	$f_i = A_i \oplus B_i$	XOR
1	0	$f_i = A_i B_i$	AND
1	1	$f_i = A_i$	NOT

Function table.

Circuit Realization of logic circuit.

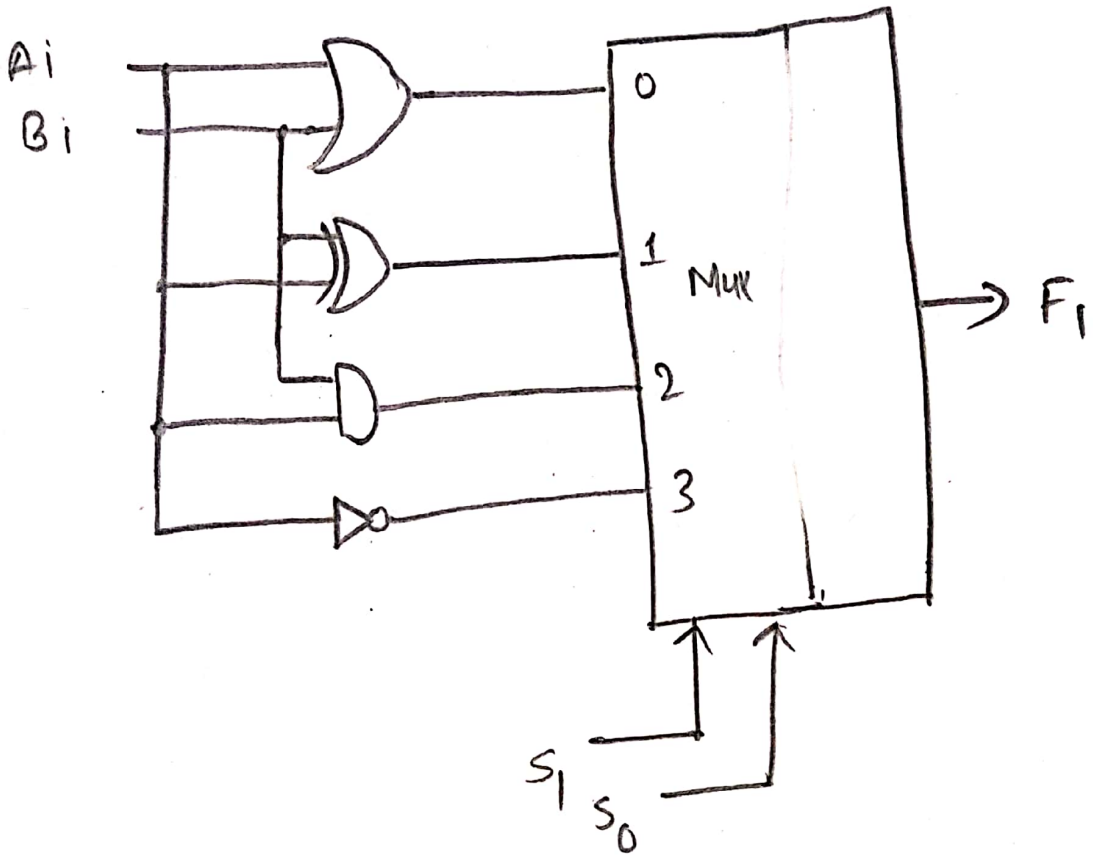


fig: logic diagram.

Design of Arithmetic circuit.

The basic component of the arithmetic section of ALU is a adder. By Controlling the data input to the parallel adder, it is possible to obtain different types of arithmetic operations

Function table for the arithmetic circuit given below

Function select			Yequals	output	function
S_1	S_0	Cin			
0	0	0	0	F = A	Transfer A
0	0	1	0	F = A+1	increment A
0	1	0	B	F = A+B	add B to A
0	1	1	B	F = A+B+1	add B to A+1
1	0	0	$\bar{B}$	F = A+$\bar{B}$	1's B + A
1	0	1	$\bar{B}$	F = A+$\bar{B}$+1	2's B + A
1	1	0	All 1's	F = A-1	Decrement A
1	1	1	All 1's	F = A	Transfer A

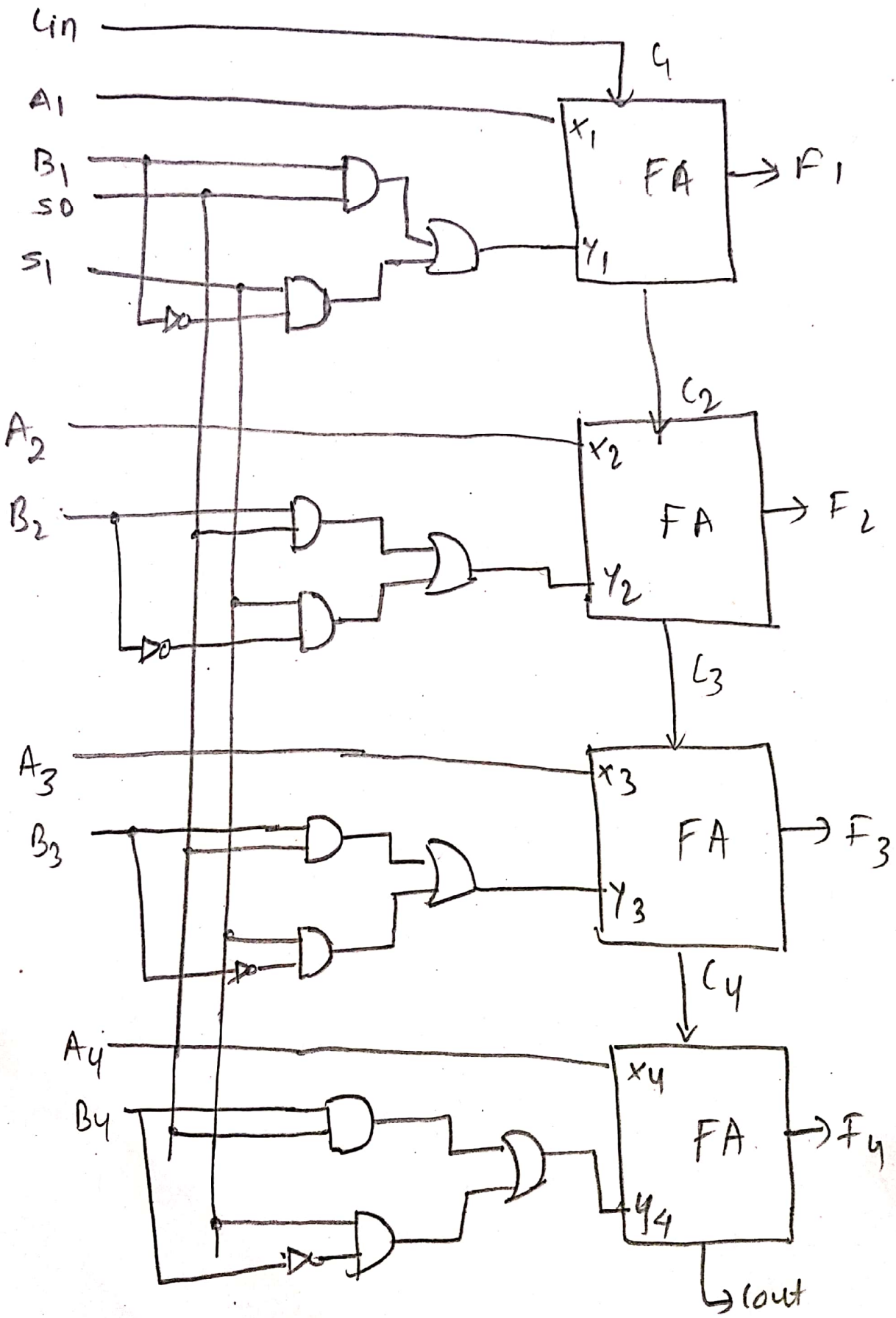


Fig: logic diagram of arithmetic circuit.

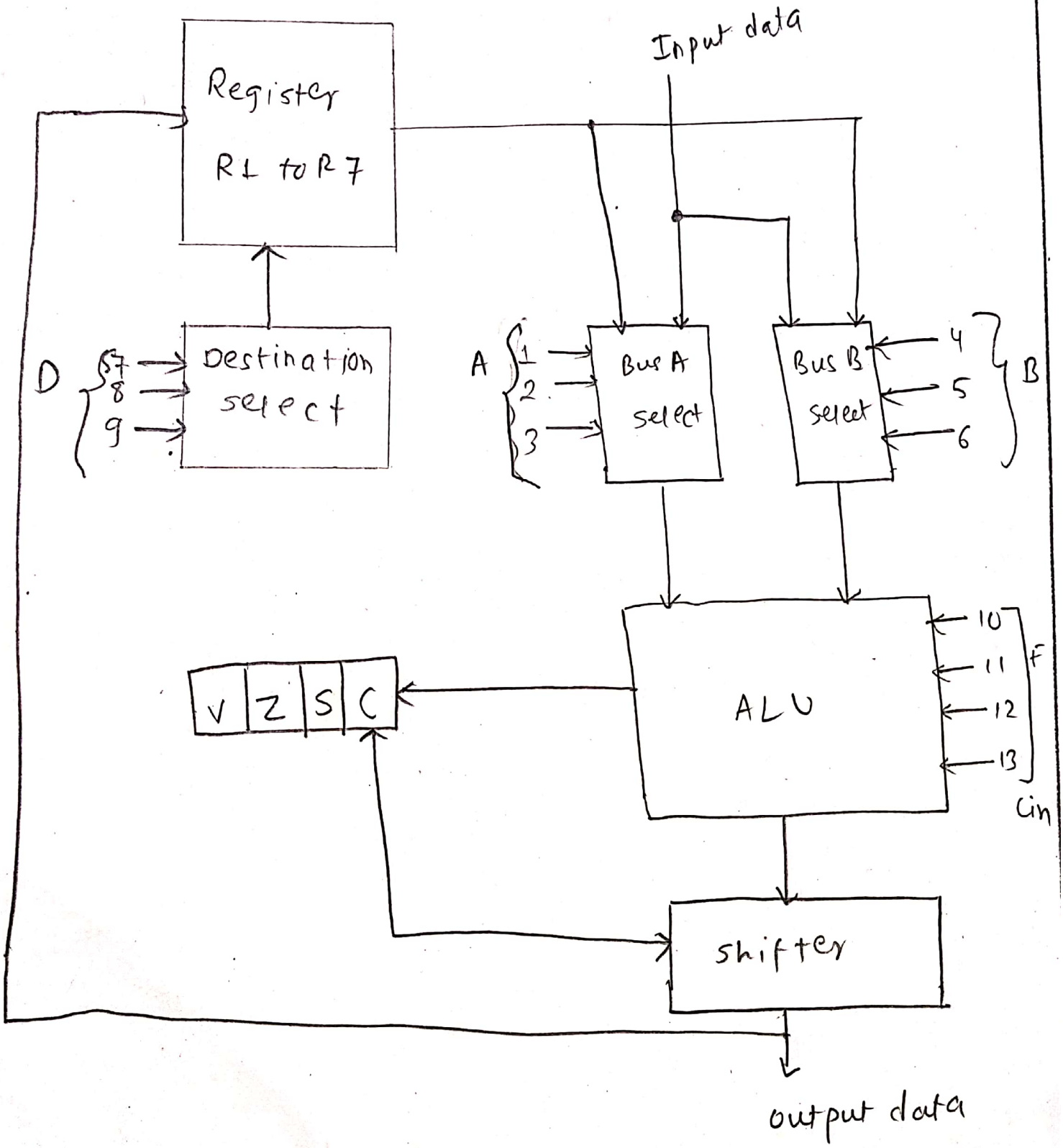


Fig: Processor Unit