DIGITAL LOGIC AND MICROPROCESSOR

Registration Number:	18BIT0272
Name:	PRIYAL BHARDWAJ
Slot:	A2

Q.1 Design a logic circuit that implements the following pseudo-code using the provided Comparator, Adder and Registers, along with as many multiplexers and de-multiplexers as needed. The comparator has two inputs In1 and In2, and three outputs, C1, C2, and C3. If In1 < In2, C1 = 1; if In1 = In2, C2=1; if In1 > In2, C3 =1 (for a given In1 and In2, only one of the comparator outputs can be 1). The Adder takes as inputs two numbers p and q, and produces an output Sum. There are 5 registers for storing the 5 variables, A, B, X, Y, and Z. • Hint: You do not need to use truth table or K-maps. Insert the multiplexers/de-multiplexers as appropriate, and show the signal connections from the input registers A, B, X to the output registers Y and Z, through the multiplexers, comparator, adder, and de-multiplexers. Be sure to show the equations for the select lines of the multiplexers/de-multiplexers in terms of the comparator outputs, C1, C2, and C3.

Pseudo-code:

If A<B then Z=X+A

Else if A=B then Z= X+ B

Else Z = A + B

A.1 Inputs: In1, In2

Output of 3 MUX: C1 (In1<In2), C2 (In1=In2) and C3 (In1>In2) respectively.

Comparator Truth Table:

In1	In2	C1	C2	C3
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

In2 – Select Line

In1 – Input Line

Multiplexer K-Map:

C1:

	In2’	In2
In1’	0	**1**
In1	2	3
	0	In1’

C2:

	In2’	In2
In1’	**0**	1
In1	2	**3**
	In1’	In1

C3:

	In2’	In2
In1’	0	1
In1	**2**	3
	In1	0

Adder Truth Table:

P	Q	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Q – Select Line

P – Input Line

Multiplexer K-Map:

Sum:

	Q’	Q
P’	0	**1**
P	**2**	3
	P	P’

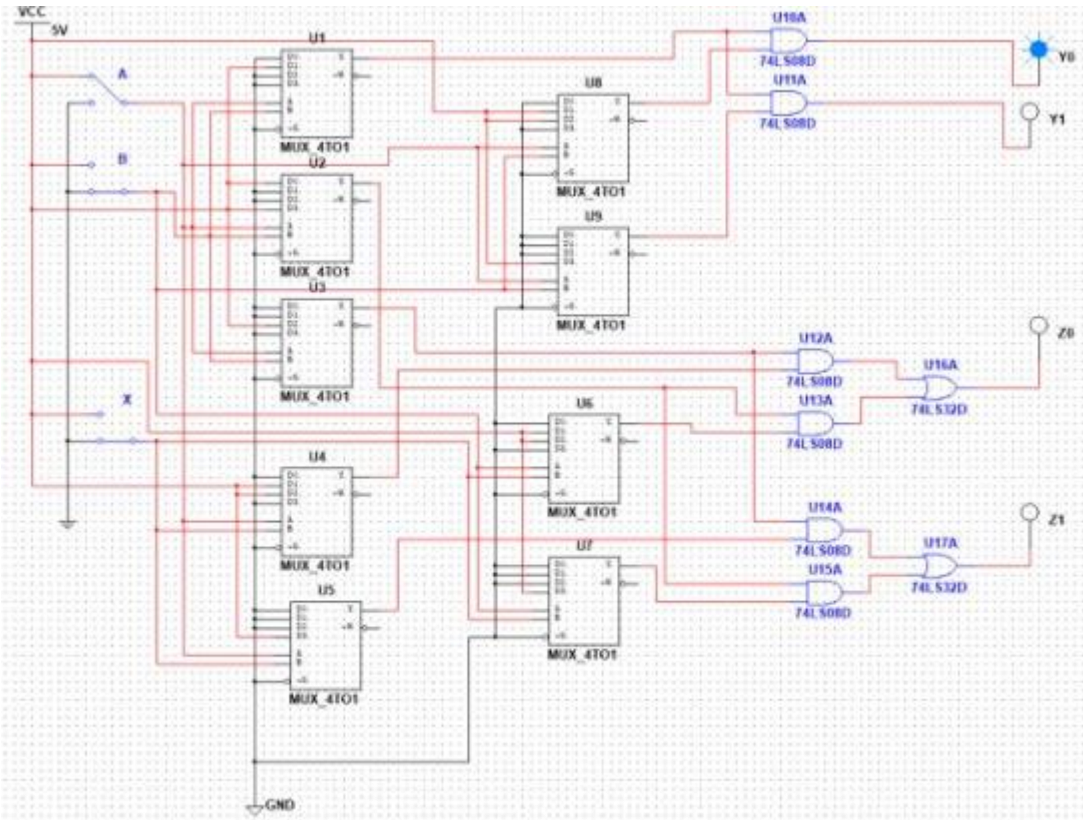
Carry:

	Q’	Q
P’	0	1
P	2	**3**
	0	P

We take 3 input switches A, B and X and 2 outputs Y and Z.
If A<B then Z=X+A
If A=B then Z=X+B
If A>B then Z=A+B

To design the circuit, we will first pass A and B through a magnitude comparator and then using the output we will pass the inputs through an adder and they will produce the final output.

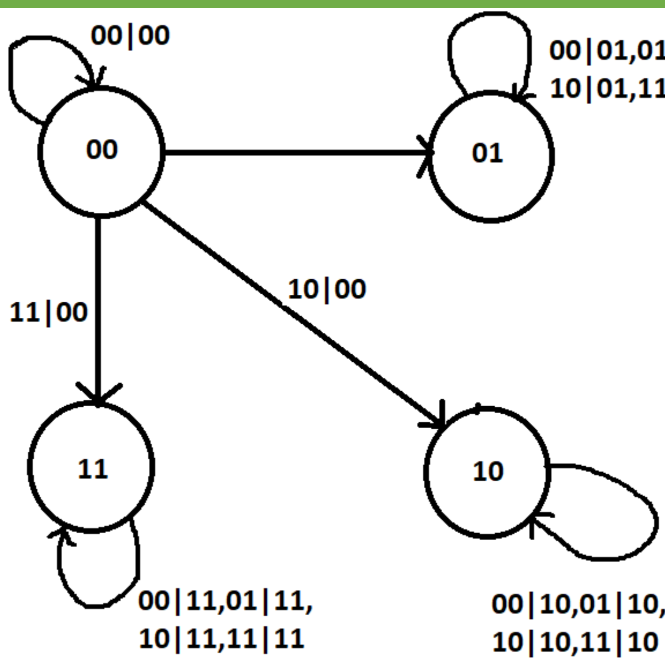
Multi Sim Circuit Diagram:



Q.2 Many game shows use a circuit to determine which of the contestant rings in first. Design a circuit to determine which of two contestant rings in first. It has two inputs S1 and S0 which are connected to the contestants' buttons. The circuit has two outputs Z1 and Z0 which are connected to LED's to indicate which contestant rang in first. There is also a reset button that is used by the game show host to asynchronously reset the flip-flops to the initial state before each question. If contestant 0 rings in first, the circuit turns on LED 0. Once LED 0 is on, the circuit leaves it on regardless of the inputs until the circuit is asynchronously reset by the game show host. If contestant 1 rings in first, the circuit turns on LED 1 and leaves it on until the circuit is reset. If there is a tie, both LED's are turned on. The circuit requires four states: reset, contestant 0 wins, contestant 1 wins, and tie. One way to map the states is to use state 00 for reset, state 01 for contestant 0 wins, state 10 for contestant 1 wins, and state 11 for a tie. With this mapping, the outputs are equal to the current state, which simplifies the output equations.

A.2 In the table below, Q1Q0 is the current state, Q1*Q0* is the next state, and Z1Z0 is the current output. There are four columns of values for the next state Q1*Q0*. The first column contains the values of the next state when S1S0 = 00, the second column contains the next state when S1S0 = 01, and so on.

State Diagram:



State Table:

PRESENT STATE		INPUT		NEXT STATE		FLIP-FLOP INPUTS		OUTPUT	
Q1	**Q0**	**S1**	**S0**	**Q1***	**Q0***	**D1**	**D0**	**Z1**	**Z0**
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	1	0	0
0	0	1	0	1	0	1	0	0	0
0	0	1	1	1	1	1	1	0	0
0	1	0	0	0	1	0	1	0	1
0	1	0	1	0	1	0	1	0	1
0	1	1	0	0	1	0	1	0	1
0	1	1	1	0	1	0	1	0	1
1	0	0	0	1	0	1	0	1	0
1	0	0	1	1	0	1	0	1	0
1	0	1	0	1	0	1	0	1	0
1	0	1	1	1	0	1	0	1	0
1	1	0	0	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

K-Map:

D1:

Q1Q0 S1S0

		1	1
1	1	1	1
1	1	1	1

D2:

Q1Q0 S1S0

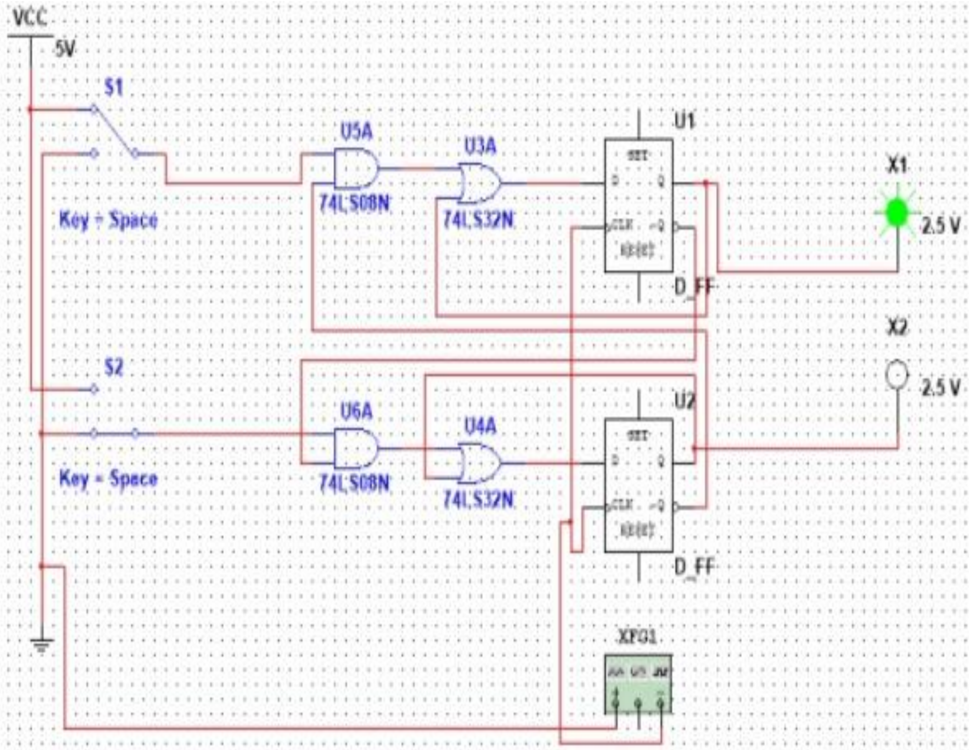
	1	1	
1	1	1	1
1	1	1	1

Expression:

D0 = Q0 + S0Q1'   Z0 = Q0

D1 = Q1 + S1Q0'   Z1 = Q1

Multi Sim Circuit Diagram:



Q.3 Assume a large room has 3 doors and a switch near each door controls a light in the room. The light is turned on or off by changing the state of any one of the switches. More specifically the following should happen:

1. The light is OFF when all 3 switches are open.

2. Closing any one switch will turn the light ON.

3. Then closing the second switch will have to TURN OFF the light.

4. If the light is OFF when the 2 switches are closed, then by closing the third switch the light will TURN ON.

A.3 S1, S2, S3(Switches), D(Door) – Open = 0; Closed = 1

Truth Table:

S1	S2	S3	D
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

K-Map:

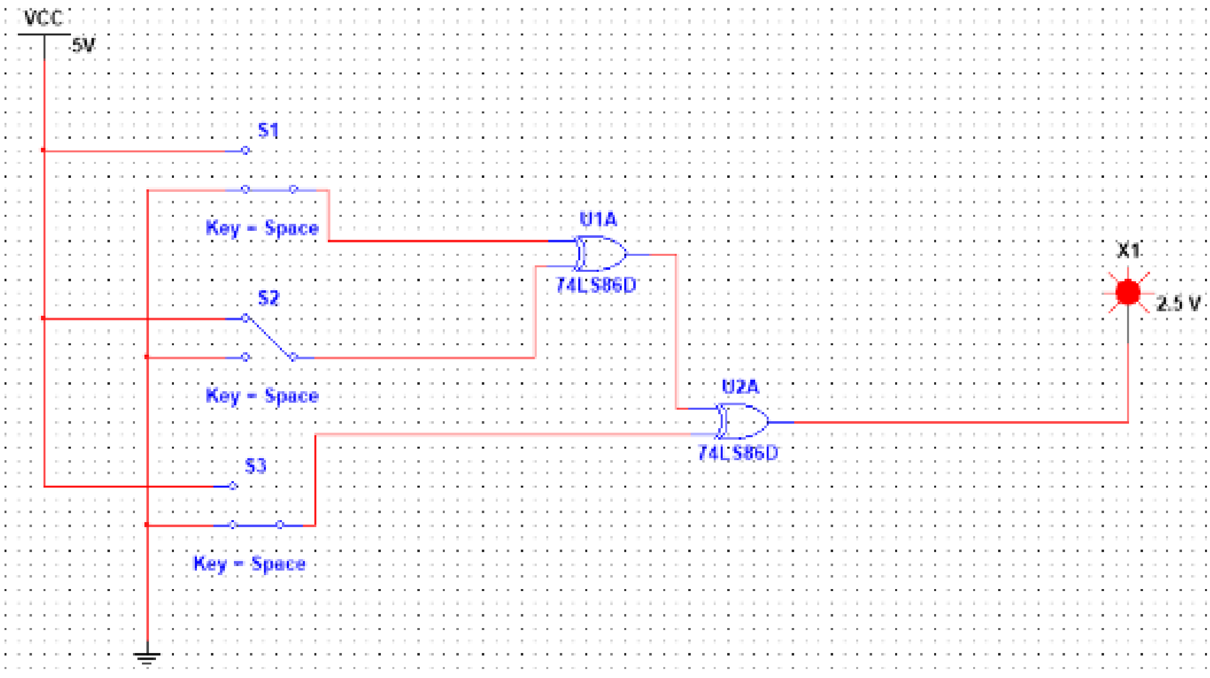
S1 \ S2S3

0	1	0	1
1	0	1	0

Expression:

D = S1 (+) S2 (+) S3

Multi Sim Circuit Diagram:

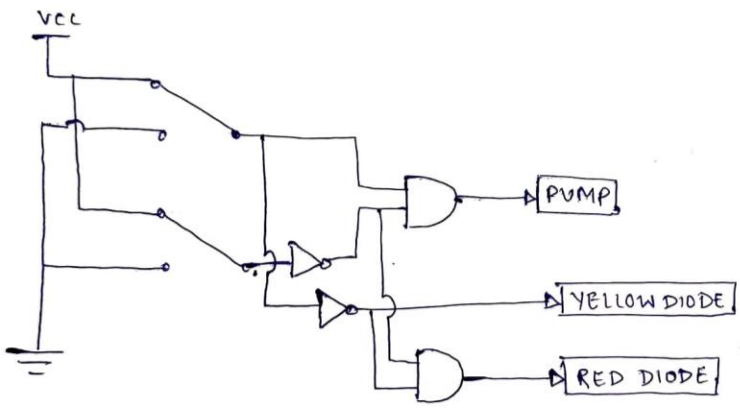


Q.4 Explain in detail about the applications of Combinational Circuits in Real time applications?

A.4

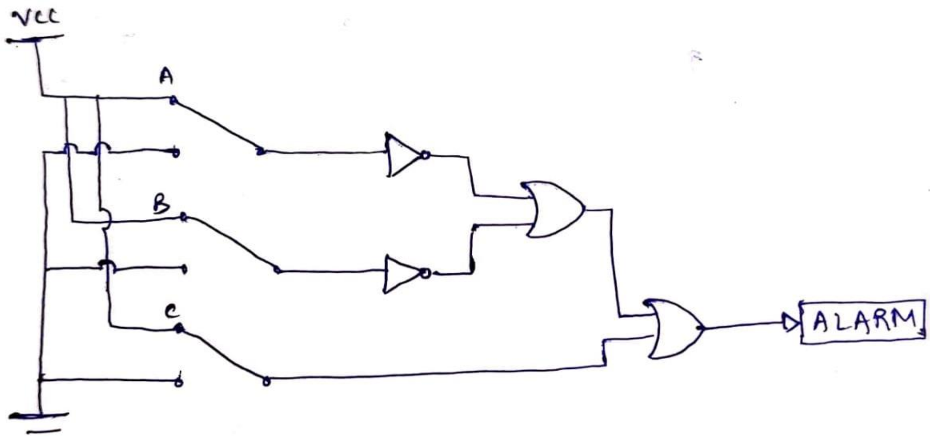
1) CONTROL OF WATER PUMPING

One of possible examples of logic control is design of logic circuit for control of water pumping from a well. Pump will operate if there is insufficient amount of water in the tank inside the house and sufficient amount of water in the well. If there is sufficiency of water in the tank inside the house and insufficiency of water in the well, yellow indicator light will light up. If there is insufficiency of water both in the tank inside the house and in the well, red indicator light will light up.



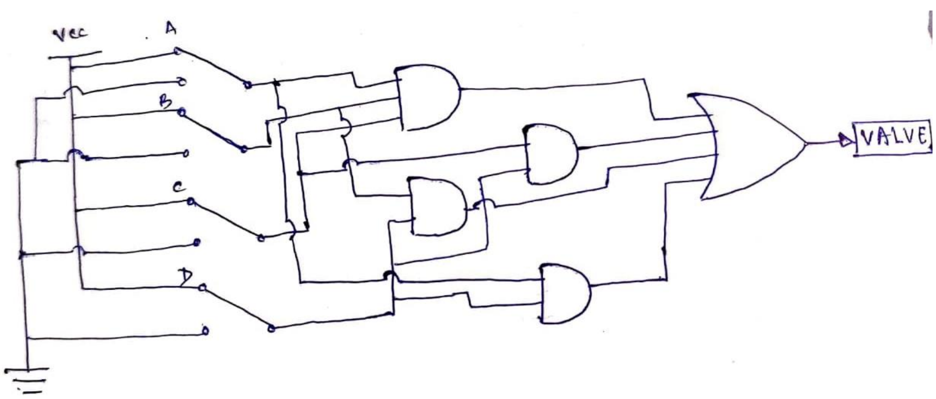
2) BURGLAR ALARM

Another example can be design of building alarm device. Monitored are two windows on ground floor and entry door. System is designed so that alarm is sounded in the event of infiltration of the object by a stranger.



3)CONTROL OF VALVE OF REVERSE PIPELINE

Another application of logic control is control of valve of reserve pipeline. Pumps A, B, C and D are connected to pipeline p1 and p2. It is necessary to design a combination control logic circuit for automatic control of valve of the backup pipeline p2, if pump flow rate exceeds the allowed flow rate of pipeline p1 – Q1= 7 l/s, where the pump flow rates are: - QA = 2 l/s - QB = 3 l/s - QC = 3 l/s - QD = 6 l/s.



4) Automatic switching on and off of street light during the night and day time respectively.

5) Multiplexers and demultiplexers are used in synchronous time domain multiplexing technique, to achieve multiline communication via a single wire.

6) Decoders and encoders are used to design RAM from registers to cut off the power consumption.

7) Combinational logics id used in computer circuits to perform Boolean algebra on input signals and stored data. For example, the part of arithmetic logic unit, or ALU, does mathematical calculations is constructed using combination logics.

8) <u>DOORBELL</u>
An OR gate is used to implement a door bell. If either the front doorbell switch OR the backdoor bell switch is pressed the doorbell rings.

9) <u>TEMPERATURE DETECTOR</u>
NOT gate is used to implement a temperature detector. For example:
If the temperature is above 20 degree centigrade then the central heating is switched off and if the temperature is below 20 degree centigrade the central heating is switched on.

******