IDNO:N210217

NAME:Shaik Akhila

LAB-01

Aim:Verification of logic gates.

Apparatus*:*

Logic gates AND,OR,NOT,NAND,NOR,XOR,XNOR,

Connecting wires,probes/LED's,Digital interactive constants for inputs.

List of logic gates and their IC numbers:

Name of gate	***IC NUMBER***
AND	***7408***
OR	***7432***
NOT	***7400***
XOR	***7486***
XNOR	***74266***
NAND	***7400***
NOR	***7402***

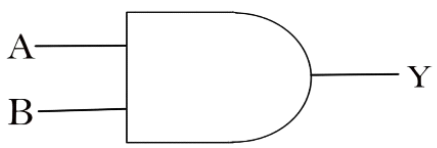
AND Gate:

An AND is a D shaped gate with two or more inputs and produces one output.

Produces a high output (1) only if all the inputs are high.

Produces low output (0) in other cases.

Circuit Diagram:



Truth table:

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Construction of AND gate in multisim:

Open a new blank in multisim.

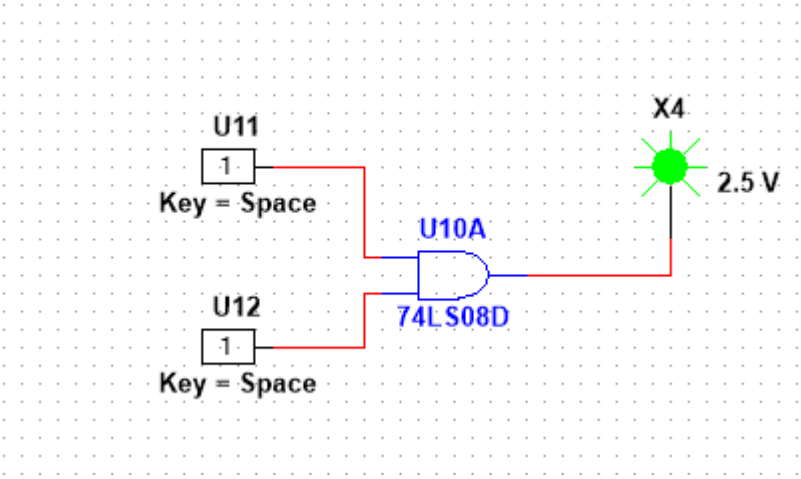
Place the and gate from components section of TTL in the breadboard.

Now connect the two inputs to AND gate and put a probe at output side.

By selecting all family all groups in the components then search for the component we can easily find the required component.

Now verify the truth table.

Multisim Circuit:



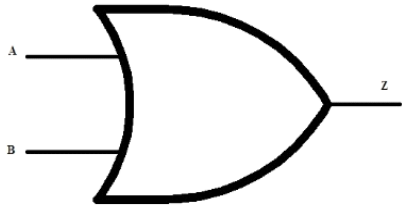
OR Gate:

A curved gate with two or more inputs and one output.

Produces a high output (1) only if one of the inputs are high.

Produces low output (0) if both the inputs are low

Circuit Diagram:



Truth table:

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Construction of OR gate in multisim:

Open a new blank in multisim.

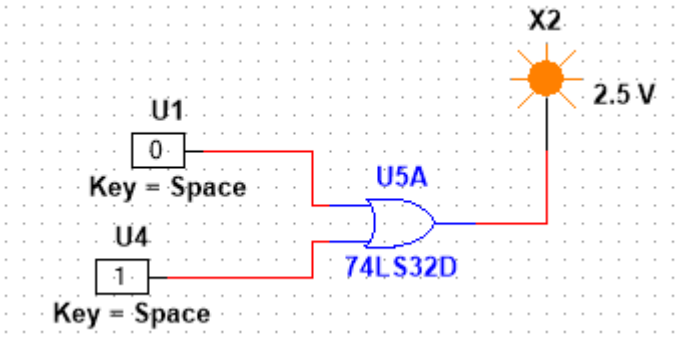
Place the and gate from components section of TTL in the breadboard.

Now connect the two inputs to OR gate and put a probe at output side.

By selecting all family all groups in the components then search for the component we can easily find the required component.

Now verify the truth table.

Multisim Circuit:



NOT Gate:

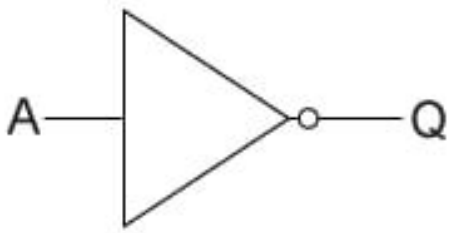
A triangle gate with a small circle at the output.

Produces a high output (1) only if the inputs are low(0).

Produces a low output (0) only if the inputs are high(1).

It is also known as inverter.

Circuit Diagram:



Truth table:

A	

	y
0	1
1	0

Construction of NOT gate in multisim:

Open a new blank in multisim.

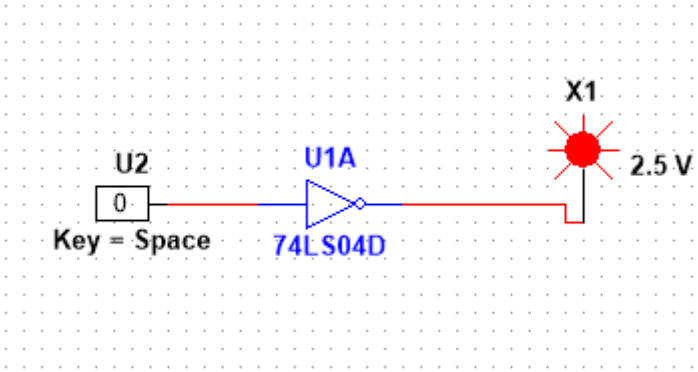
Place the and gate from components section of TTL in the breadboard.

Now connect the two inputs to NOT gate and put a probe at output side.

By selecting all family all groups in the components then search for the component we can easily find the required component.

Now verify the truth table.

Multisim Circuit:

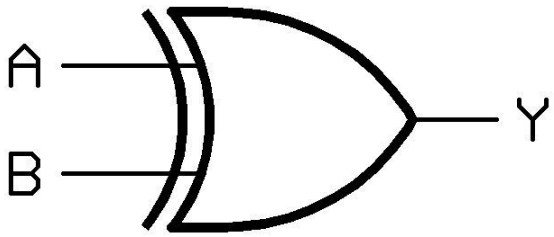


XOR Gate(Exclusive OR):

Produces a high(1) output if exactly one input is high(1)

Otherwise produces low(0).

Circuit Diagram:



Truth table:

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Construction of XOR gate in multisim:

Open a new blank in multisim.

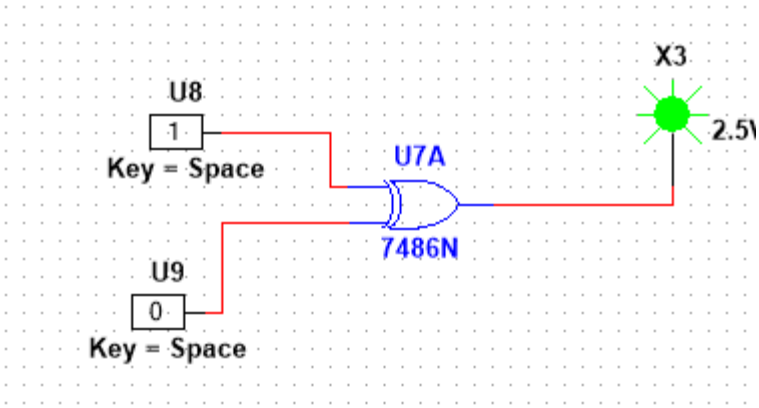
Place the and gate from components section of TTL in the breadboard.

Now connect the two inputs to XOR gate and put a probe at output side.

By selecting all family all groups in the components then search for the component we can easily find the required component.

Now verify the truth table.

Multisim Circuit:

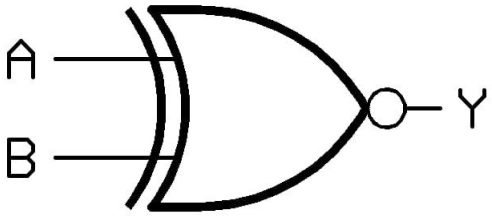


XNOR Gate(Exclusive NOR):

Produces a high(1) output if both input is high(1) or low(0).

Otherwise produces low(0) means when both inputs are different.

Circuit Diagram:



Truth table:

A	B	Y
0	0	1
0	1	0
1		0

	0	
1	1	1

Construction of XNOR gate in multisim:

Open a new blank in multisim.

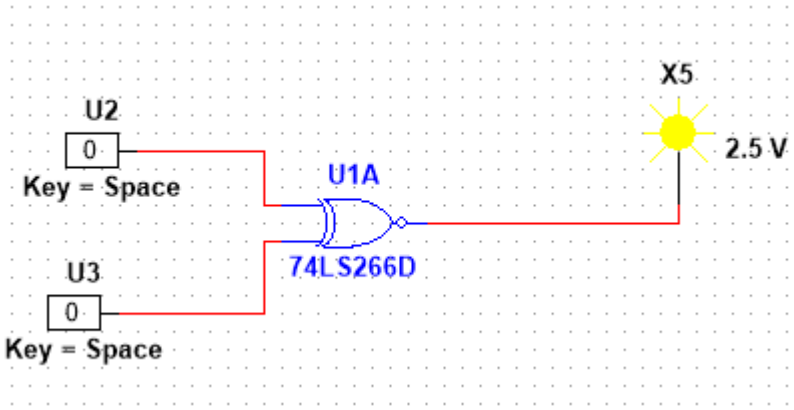
Place the and gate from components section of TTL in the breadboard.

Now connect the two inputs to XNOR gate and put a probe at output side.

By selecting all family all groups in the components then search for the component we can easily find the required component.

Now verify the truth table.

Multisim Circuit:

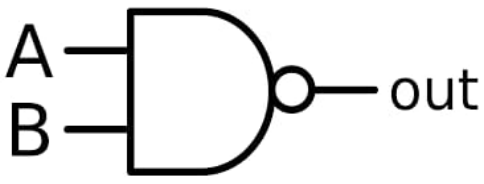


NAND Gate(NOT+AND):

Produces a high(1) output if not all inputs are high.

Otherwise produces low(0).

Circuit Diagram:



Truth table:

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Construction of NAND gate

in multisim:

Open a new blank in multisim.

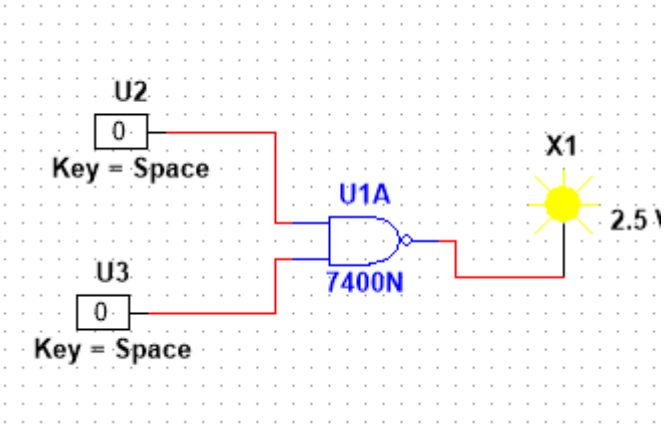
Place the and gate from components section of TTL in the breadboard.

Now connect the two inputs to NAND gate and put a probe at output side.

By selecting all family all groups in the components then search for the component we can easily find the required component.

Now verify the truth table.

Multisim Circuit:

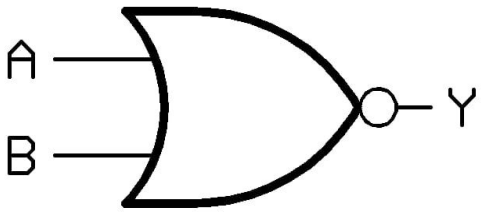


NOR Gate(NOT+OR):

Produces a high(1) output only if not all inputs are low(0).

Otherwise produces high(1).

Circuit Diagram:



Truth table:

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Construction of NOR gate in multisim:

Open a new blank in multisim.

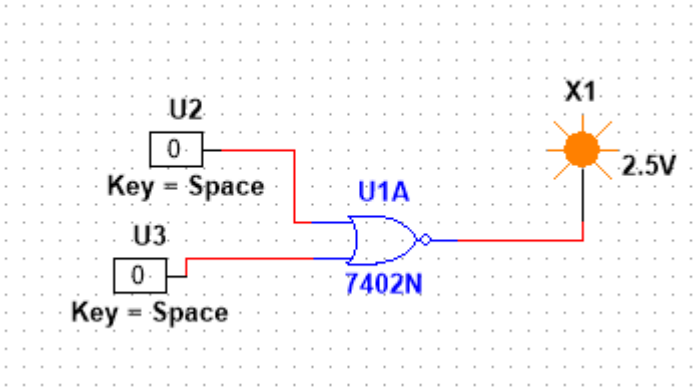
Place the and gate from components section of TTL in the breadboard.

Now connect the two inputs to NOR gate and put a probe at output side.

By selecting all family all groups in the components then search for the component we can easily find the required component.

Now verify the truth table.

Multisim Circuit:



ID NO : N210217

NAME: Shaik . Akhila

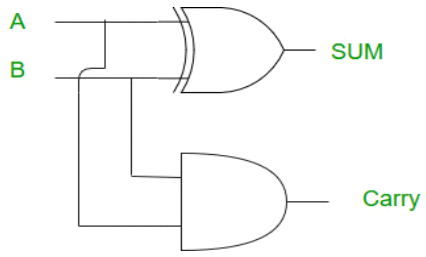
LAB-02

Aim: To verify half adder, full adder using half adders,half subtractor,full subtractor,full subtractor using half subtractors

HALF ADDER:

Description:- A half adder as a type of adder, an electronic circuit that performs the addition of numbers. The half adder is able to add two single binary digits and provide the output plus a carry value.

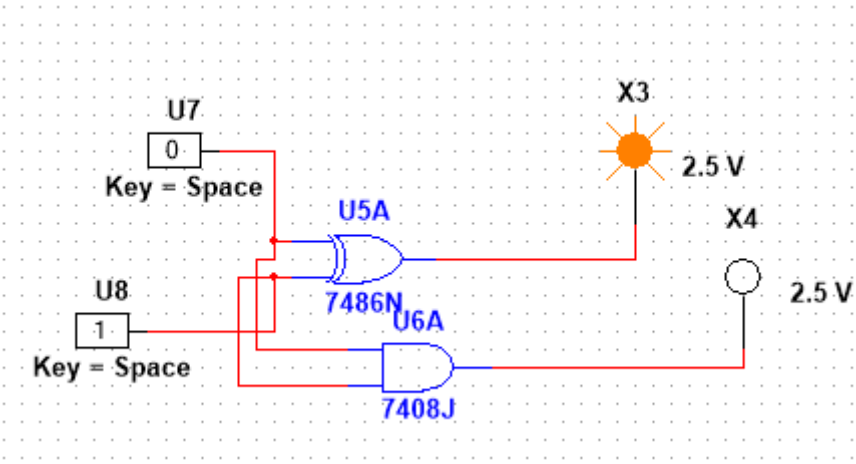
Circuit diagram:-



Truth table:-

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Multisim circuit:-



Components required:- Two interactive digital constants, XOR gate, AND gate ,two LEDs, Two grounds

Procedure:-

1)Open multisim software

2)Create new design

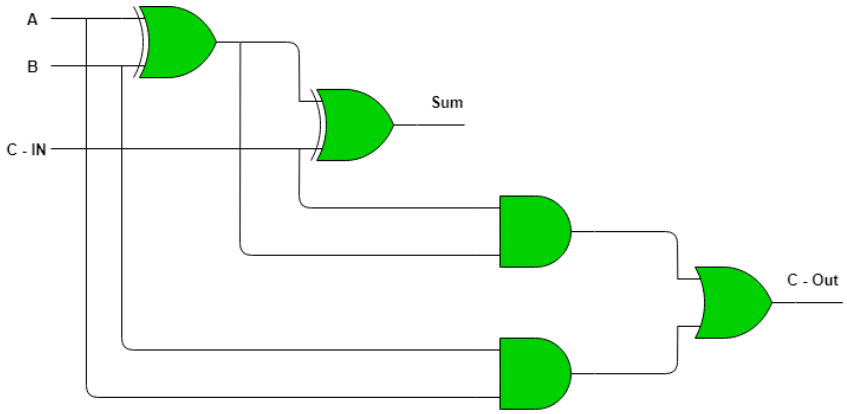
3)Place the compinents and connect them according to the circuit diagram

4)Verify the circuit diagram with the help of truth table

FULL ADDER:

Description:- A full adder adds three one-bit binary numbers, two operands and a carry bit. The adder outputs two numbers, a sum and a carry bit.

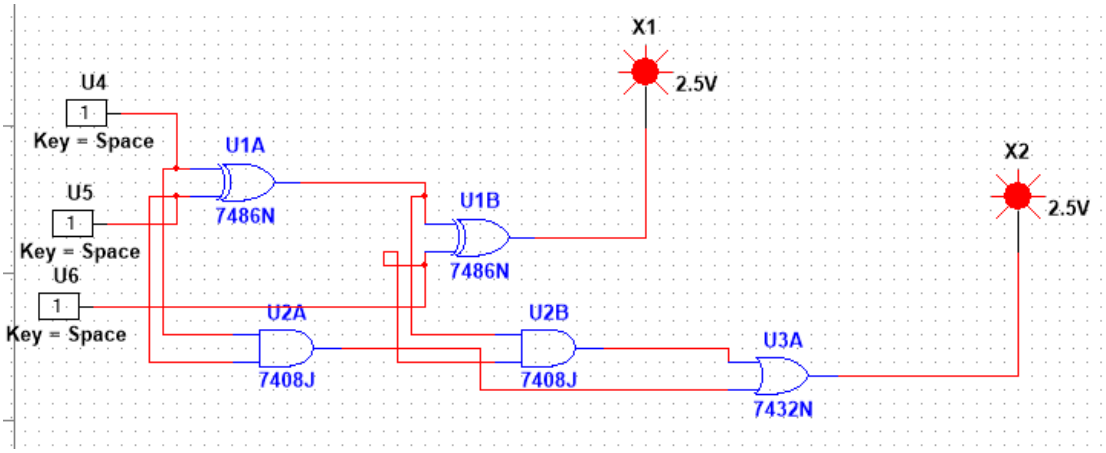
Circuit diagram:-



Truth table:-

Inputs			Outputs	
A	B	C – IN	Sum	C - Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Multisim circuit:-



Components required:- Two interactive digital constants, XOR gate, AND gate, two LEDs, Two grounds

Procedure:-

1)Open multisim software

2)Create new design

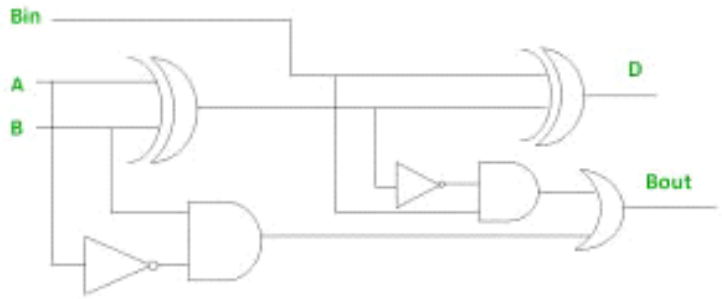
3)Place the compinents and connect them according to the circuit diagram

4)Verify the circuit diagram with the help of truth table

FULL SUBTRACTOR:

Description:- The full subtractor is used to subtract three 1-bit numbers A,B,C which are minuend, subtrahend, and borrow,respectively.

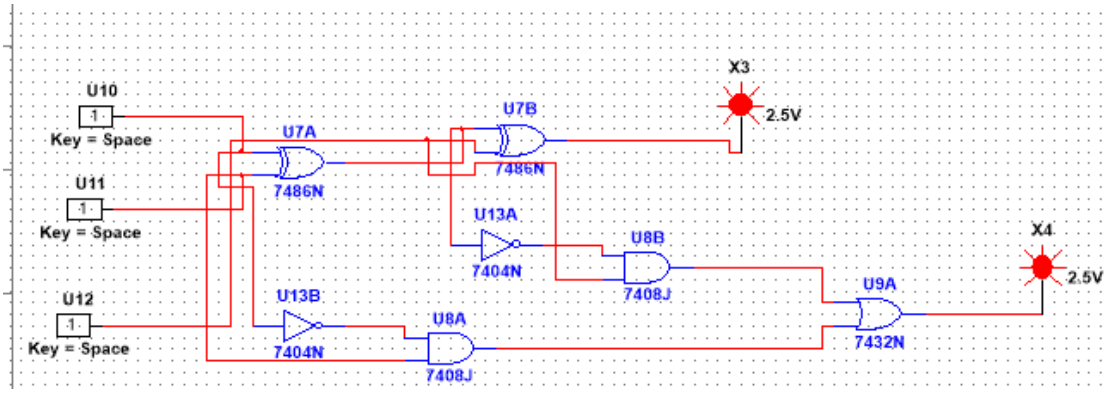
Circuit diagram:-



Truth table:-

INPUT			OUTPUT	
A	B	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Multisim circuit:-



Components required:- Two interactive digital constants, XOR gate, AND gate, NOT gate,two LEDs, Two grounds

Procedure:-

1)Open multisim software

2)Create new design

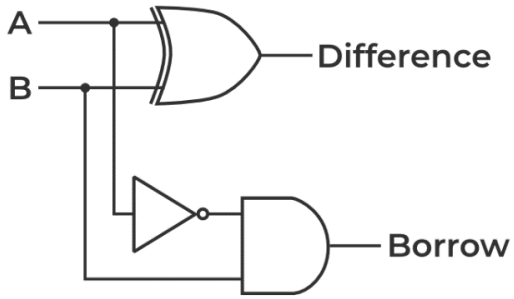
3)Place the compinents and connect them according to the circuit diagram

4)Verify the circuit diagram with the help of truth table

HALF SUBTRACTOR:

Description:- The half subtractor is a combinational circuit which is used to perform subtraction of two bits. A and B and two outputs Difference and Borrow.

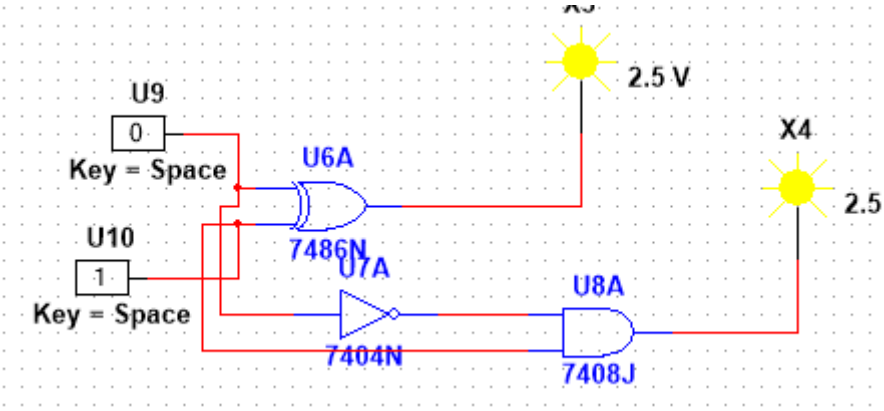
Circuit diagram:-



Truth table:-

A	B	Diff	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Multisim circuit:-



Components required:- Two interactive digital constants, XOR gate, AND gate , NOT gate,two LEDs, Two grounds

Procedure:-

1)Open multisim software

2)Create new design

3)Place the compinents and connect them according to the circuit diagram

4)Verify the circuit diagram with the help of truth table

ID NO : N210217
NAME: Shaik Akhila

LAB-3

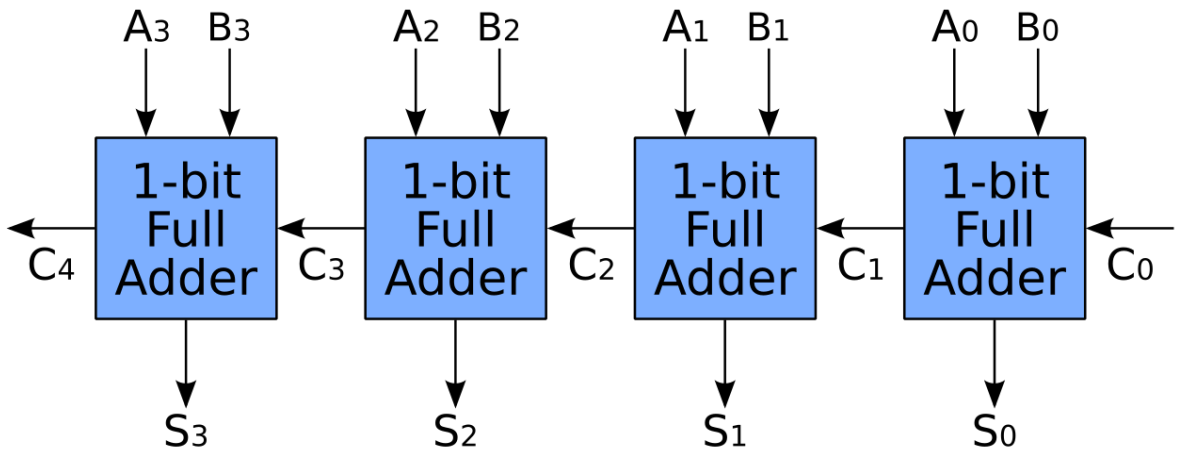
Aim : verification of ‘Ripple Carry Adder’ and ‘Look Ahead Adder’.

1.

Name : Ripple Carry Adder .

Description : A ripple carry adder (RCA) is a logical circuit that adds two or more binary numbers together .

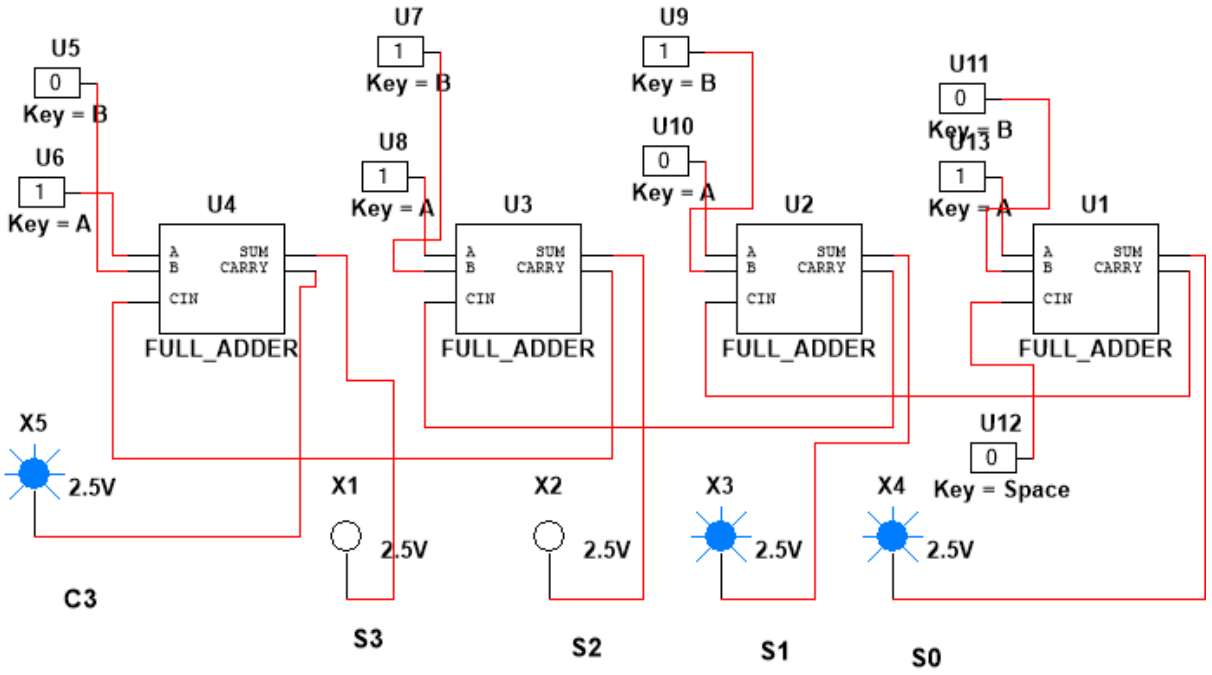
Circuit diagram :



Truth Table:

A3	A2	A1	A0	B3	B2	B1	B0	S3	S2	S1	S0	Carry
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	0	1	1	0	0
0	0	0	1	1	0	0	0	1	0	0	1	0
0	1	0	1	1	0	1	0	1	1	1	1	0
0	0	1	1	1	1	0	0	1	1	1	1	0
0	1	1	1	1	1	1	0	0	1	0	1	1
1	1	1	1	1	1	1	1	1	1	1	0	1

Multisim circuit :



Ripple carry Adder

Procedure:

1.Here I have taken 4 full adder in the multisim.
2.Then connected them as in the circuit diagram.
3.Here the carry is forwarded.
4.And connected probes to them.
5.And check them with the truth table.
6.Thus it acts as ‘Ripple Carry Adder’.

Components required :

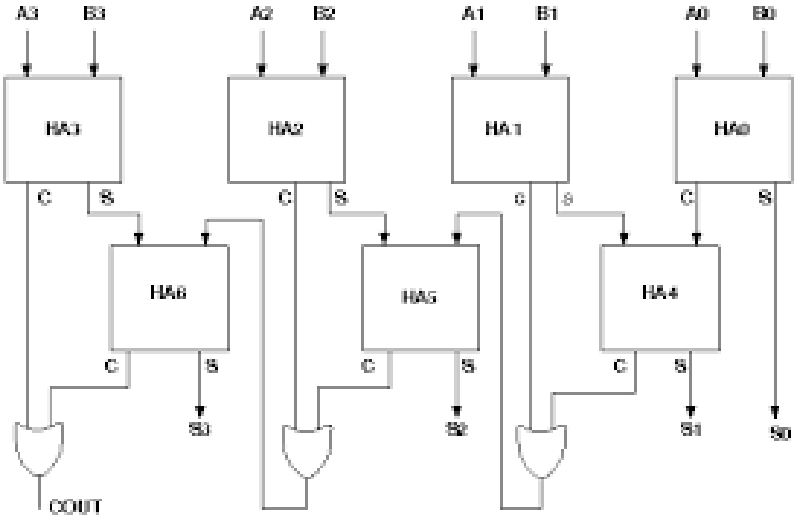
4 Full Adders
Probes
Inputs

2.

Name: Carry Look Ahead Adder

Description: It is also used to add two binary numbers . There is time delay in Ripple carry Adder because of carry forward. It gives output faster than Ripple Carry Adder.

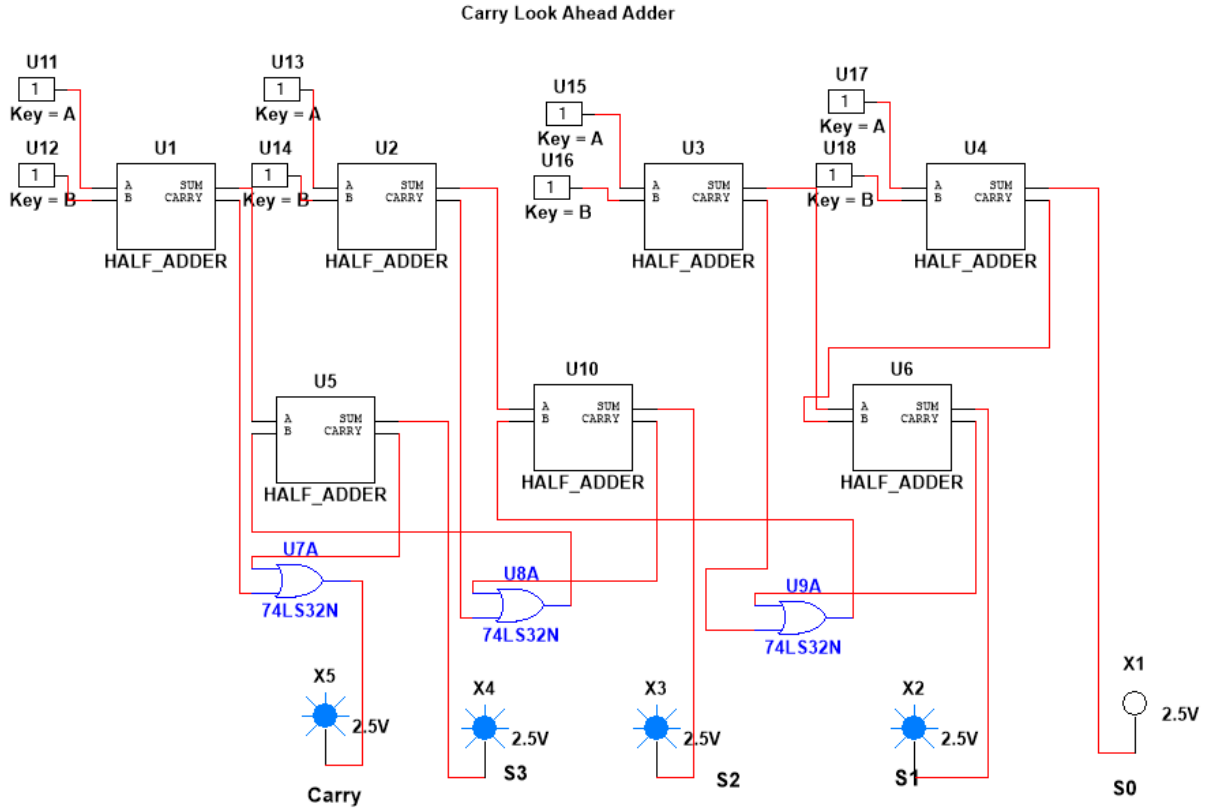
Circuit Diagram:



Truth Table:

A3	A2	A1	A0	B3	B2	B1	B0	S3	S2	S1	S0	Carry
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	0	1	1	0	0
0	0	0	1	1	0	0	0	1	0	0	1	0
0	1	0	1	1	0	1	0	1	1	1	1	0
0	0	1	1	1	1	0	0	1	1	1	1	0
0	1	1	1	1	1	1	0	0	1	0	1	1
1	1	1	1	1	1	1	1	1	1	1	0	1

Multisim Diagram:



Procedure:

1.Here I have taken 7 half adder in the multisim.
2.Then connected them as in the circuit diagram.
3.Connected them to the OR gates .
4.And connected probes to them.
5.And check them with the truth table.
6.Thus it acts as ‘Carry Look Ahead Adder’.

Components Required:

7 Half Adders
3 OR gates
Probes
Inputs

ID NO : N210217

Name : Shaik Akhila

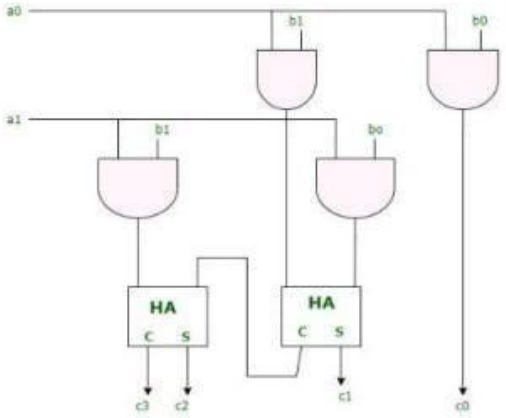
Lab-4

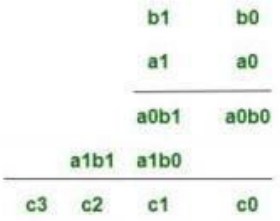
a)Aim: To design and verify the 2 x 2 Array multiplier.

Description:

- An array multiplier is a digital combinational circuit used for multiplying two binary numbers by employing an array of full adders and half adders .
- This array is used for the nearly simultaneous addition of the various product terms involved .To form the various product terms an array of AND gates is used before the Adder array.

Circuit diagram:

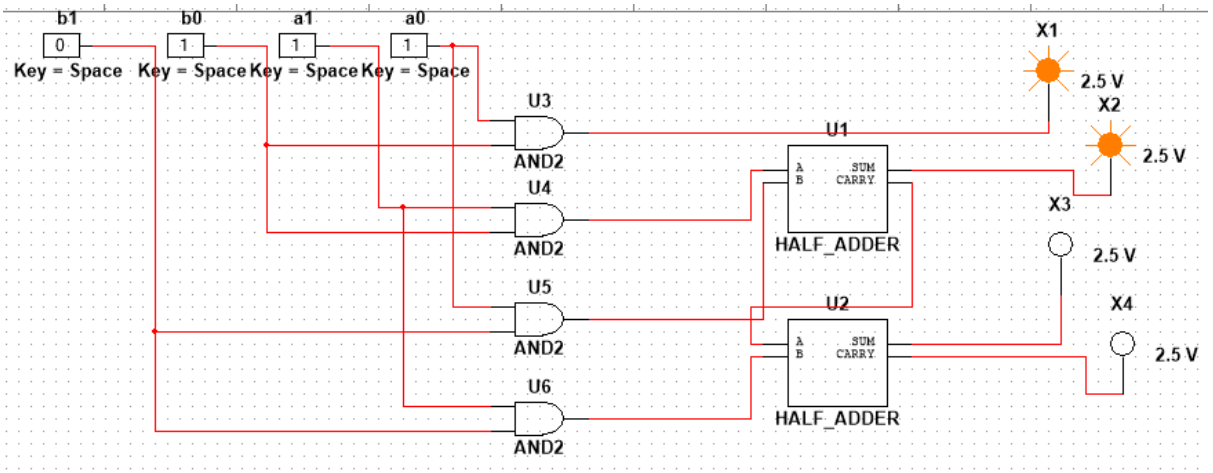




Truth Table:

b1	b0	a1	a0	P3	P2	P1	P0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

Multisim circuit:



Components required:

- 4 Interactive digital constants
- 4 AND gates
- 2 Half Adders
- 4 probes

Procedure:

1. Open multisim and create a new design space.
2. Place 4 AND gates and 2 Half adders and 4 interactive digital constants from the component library.
3. Connect the components as shown in the above circuit .
4. Connect the output to the end of the probe .
5. Simulate the circuit and verify the truth table.

ID NO : N210217

Name: SHAIK AKHILA

LAB-05

Aim: Verification of wallace tree adder.

Components:

To build a wallace tree adder to add three 4-bit input, we need:

4 full adders,4-bit adder.. Display units to check the outputs., Wires to connect.

Wallace Tree Adder:

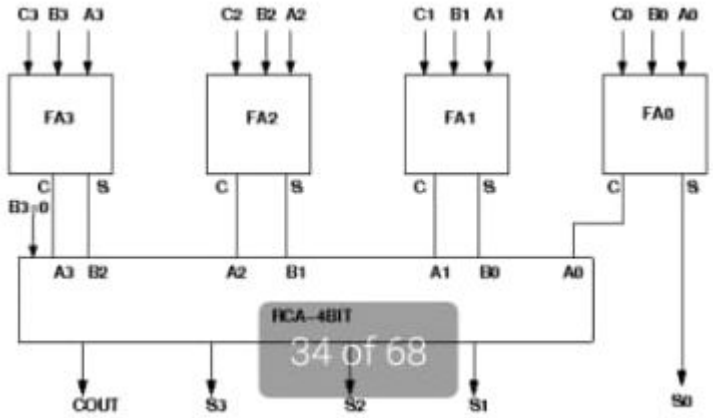
A wallace multiplier is a hardware implementation of a binary multiplier, a digital circuit that multiplies two Integers. It uses a selection of full and half adders to sum partial products in stages until the two numbers are left. The wallace tree has three steps.

1) Multiply each bit of one of the arguments, by each bit of the other.

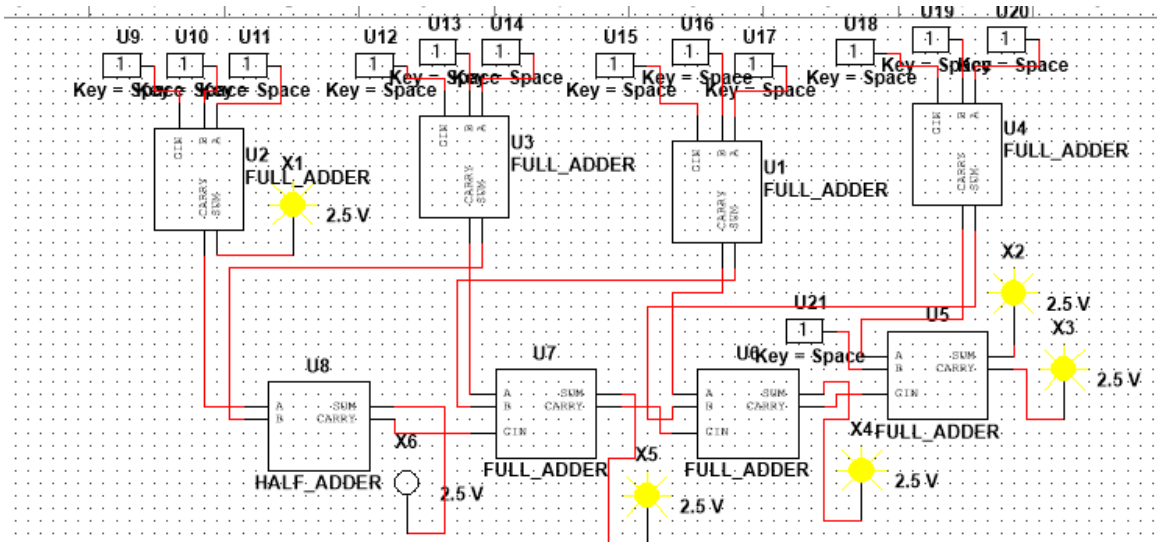
2) Reduce the number of partial products of two by layers of full and half adders.

3) Group the wires in two numbers, and add them with a conventional adder.

Circuit diagram :



Multisim circuit :



Procedure:

1. open a new blank page in multisim
2. Connect all full adders and half adders to the inputs.

3. Connect the probes to the outputs.

4. Verify the truth table.

ID NO : N210217

Name: SHAIK AKHILA

LAB-06

Aim: To design and verify the combinational multiplier.

Components:

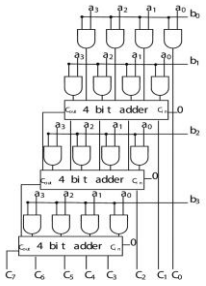
To build a combinational multiplier:

8 full adders,4 half adders,16 and gates,7 inputs, Display units to check the outputs, Wires to connect.

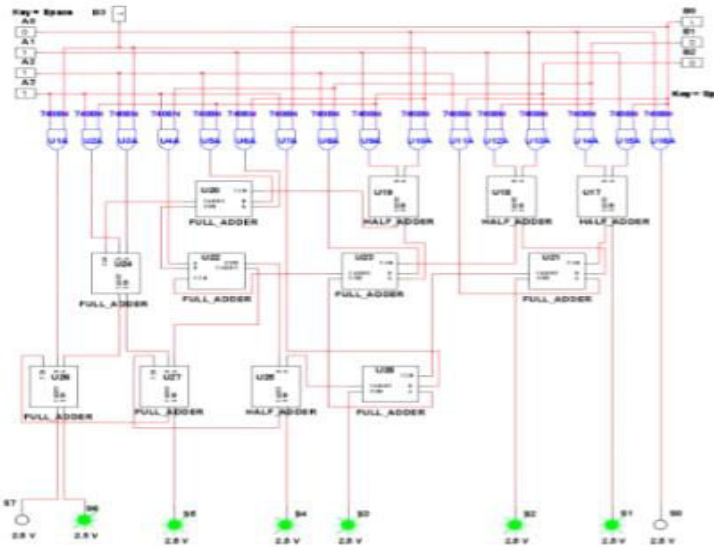
Description:

Combinational Multipliers do multiplication of two unsigned binary numbers. Each bit of the multiplier is multiplied against the multiplicand, the product is aligned according to the position of the bit within the multiplier, and the resulting products are then summed to form the final result.

Circuit diagram :



Multisim circuit :



Truth Table:

Cin	A	B	Product	Carry
	A3 A2 A1 A0	B3 B2 B1 B0	S3 S2 S1 S0	Cout
0	0 0 0 0	0 0 0 0	0 0 0 0	0
0	0 0 0 1	0 0 0 1	0 0 1 0	0
0	0 0 1 0	0 0 1 0	0 1 0 0	0
0	0 0 1 1	0 0 1 1	0 1 1 0	0
0	0 1 0 0	0 1 0 0	1 0 0 0	0
0	0 1 0 1	0 1 0 1	1 0 1 0	0
0	0 1 1 0	0 1 1 0	1 1 0 1	0
0	0 1 1 1	0 1 1 1	1 1 0 0	0
0	1 0 0 0	1 0 0 0	0 0 0 0	0
0	1 0 0 1	1 0 0 1	0 0 1 0	1
0	1 0 1 0	1 0 1 0	0 1 0 0	1
0	1 0 1 1	1 0 1 1	0 1 1 0	1
0	1 1 0 0	1 1 0 0	1 0 0 0	1
0	1 1 0 1	1 1 0 1	1 0 1 0	1
0	1 1 1 0	1 1 1 0	1 1 0 0	1
0	1 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

Procedure:

1. Start the simulator as directed. This simulator supports 5-valued logic.

2. To design the circuit we need 8 full adders, 4 half adders, 16 AND gates, 8 bit switch (to give input, which will toggle its value with a double click), 8 bit displays (to see the output), wires.

3. connect all the components based on the circut diagram.

4. verify the output through probes.

ID no : N210217

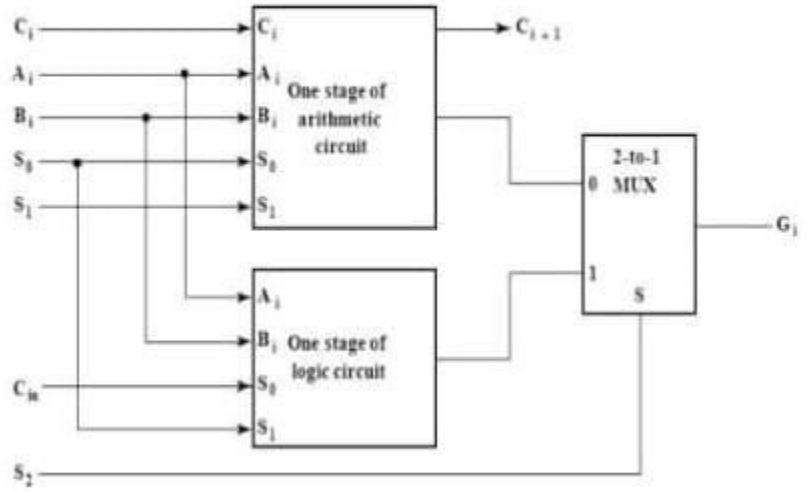
Name: Shaik.Akhila

LAB-7

Arithmetic logic unit:

Description: Arithmetic logic unit is implemented by combining the Arithmetic unit and logic unit with the help of 2 by 1 multiplexer**.**

Circuit Diagram:



Components required:

Two 4 by 1 multiplexers, One 2 by 1 multiplexers, Two full adders, One half adder, Two 2-input xor gates, Two 2-input and gates, One or gate, One nand gate, Probes

Procedure:

1) Open a new blank schematic in multisim and place all the components required.

2)Connect all the components with the help of circuit diagram.

3)Give inputs to the Arithmetic unit by using full adders and half adders and to gates by using input keys.

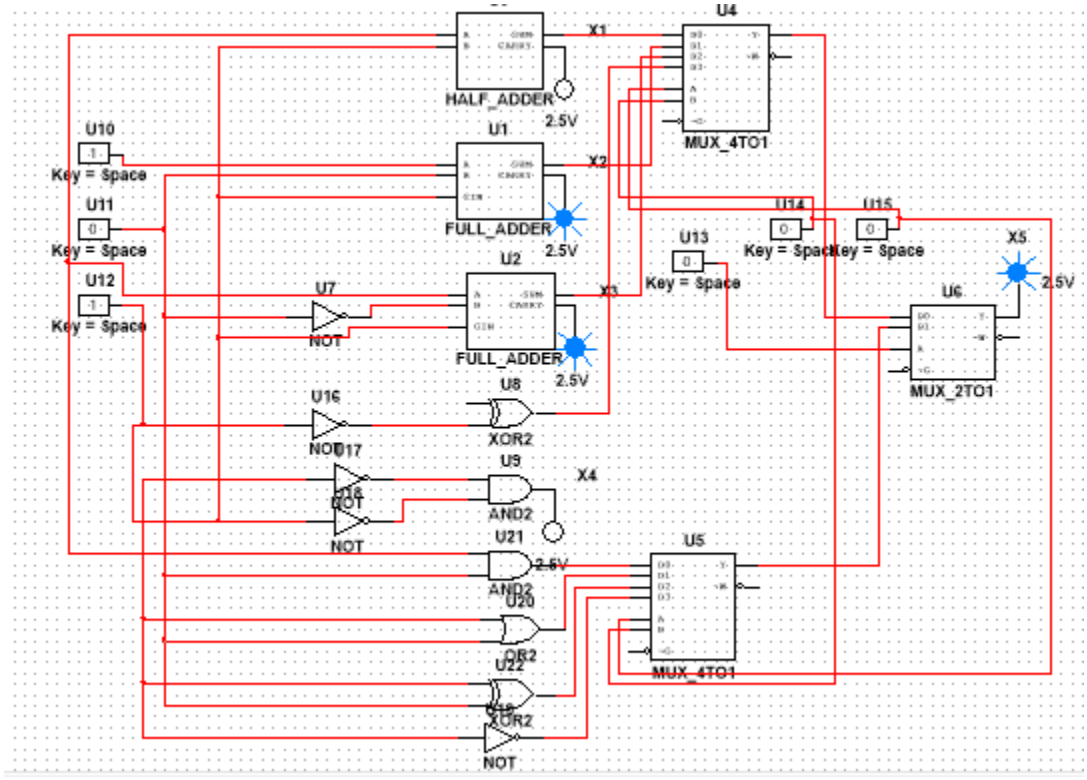
4)Connect probes to the output lines of multiplexer and wherever it is need.

5)Run the schematic and check the implementation of ALU by varying the inputs according to the truth table.

Truth Table:

Operation select					
S_2	S_1	S_0	C_{in}	Operation	Function
0	0	0	0	$F = A$	Transfer A
0	0	0	1	$F = A + 1$	Increment A
0	0	1	0	$F = A + B$	Addition
0	0	1	1	$F = A + B + 1$	Add with carry
0	1	0	0	$F = A + \overline{B}$	Subtract with borrow
0	1	0	1	$F = A + \overline{B} + 1$	Subtraction
0	1	1	0	$F = A - 1$	Decrement A
0	1	1	1	$F = A$	Transfer A
1	0	0	×	$F = A \wedge B$	AND
1	0	1	×	$F = A \vee B$	OR
1	1	0	×	$F = A \oplus B$	XOR
1	1	1	×	$F = \overline{A}$	Complement A

Multisim circuit:



IDNO:N210217

NAME:Shaik Akhila

LAB-09

Aim:Design different types of registers and verify their truthtables.

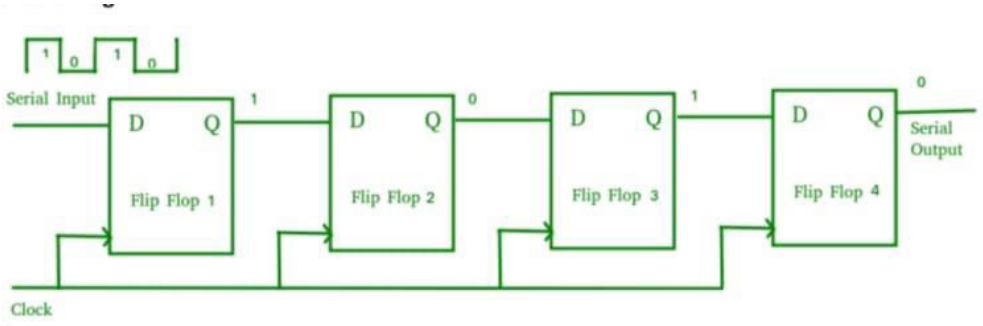
Apparatus*:*

4 D flipflops, and gates ,or gates ,Connecting wires,probes/LED's,digital clock and Digital interactive constants for inputs.

SISO REGISTER :

A Serial-In Serial-Out shift register is a sequential logic circuit that allows data to be shifted in and out one bit at a time in a serial manner.

Circuit Diagram:



Truth table:

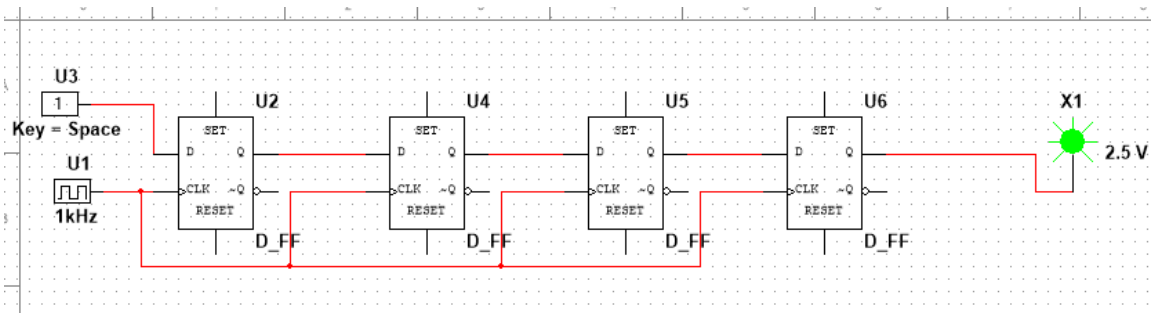
Operation of the Shift-right Register

Timing pulse	Q_A	Q_B	Q_C	Q_D	*Serial output at* Q_D
Initial value	0	0	0	0	0
After 1st clock pulse	1	0	0	0	0
After 2nd clock pulse	1	1	0	0	0
After 3rd clock pulse	0	1	1	0	0
After 4th clock pulse	1	0	1	1	1

Procedure:

1. open a new circuit, place the required number of D flip-flops
2. connect their outputs sequentially to the next flip-flop's data input
3. connect the first flip-flop's data input to your input signal, and finally, take the output from the last flip-flop as your SISO output
4. remember to connect a clock signal to all flip-flops' clock inputs

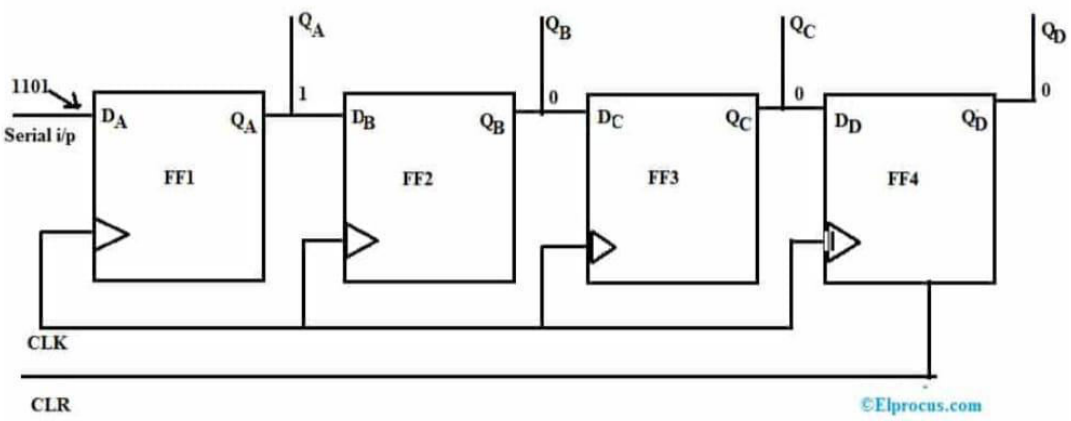
Multisim Circuit:



SIPO REGISTER:

A "SIPO" register, which stands for "Serial-In Parallel-Out," is a type of digital circuit that takes data in one bit at a time (serial input) and outputs all the bits simultaneously on separate pins, effectively converting serial data into parallel data

Circuit Diagram:



SIPO Shift Register Diagram

TRUTHTABLE:

CLK Pulse	QA	QB	QC	QD
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	0	1	0
4	1	1	0	1

SIPO Shift Register Truth Table

Procedure:

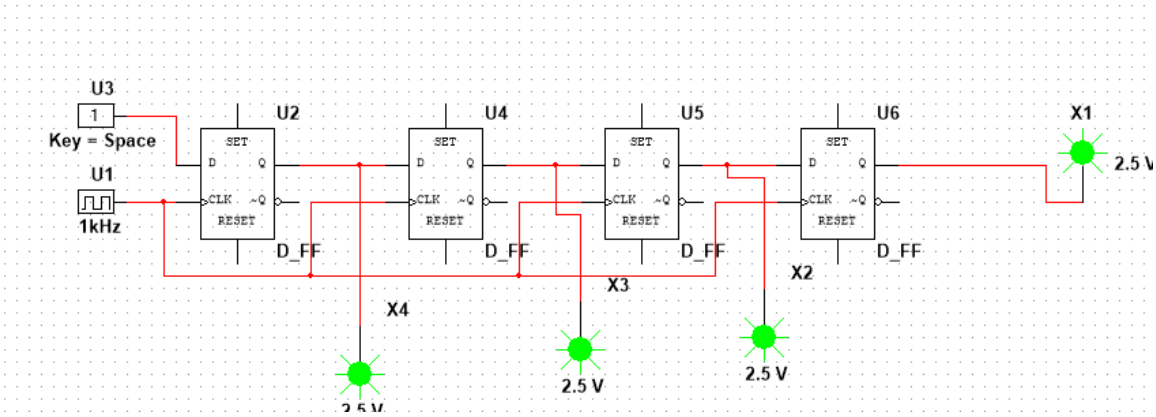
1. open a new circuit, place multiple D-flip flops in a chain

2. connect the output of each flip flop to the data input of the next, then connect the serial data input to the first flip flop's data input

3. the clock signal to the clock input of all flip flops

4. the parallel outputs will be available at the Q outputs of each flip flop

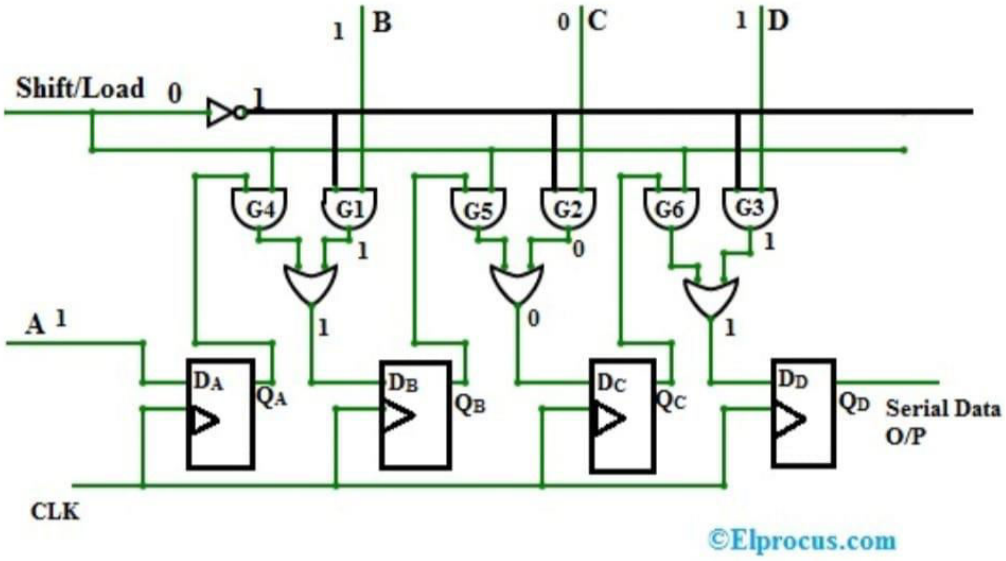
Multisim Circuit:



PISO REGISTER:

A PISO register, which stands for "Parallel-In Serial-Out" register, is a type of shift register that allows data to be loaded in parallel (all bits at once) but outputs the data serially, meaning one bit at a time, with each clock pulse shifting the data along the register.

Circuit Diagram:



PISO Shift Register Circuit

Truth table:

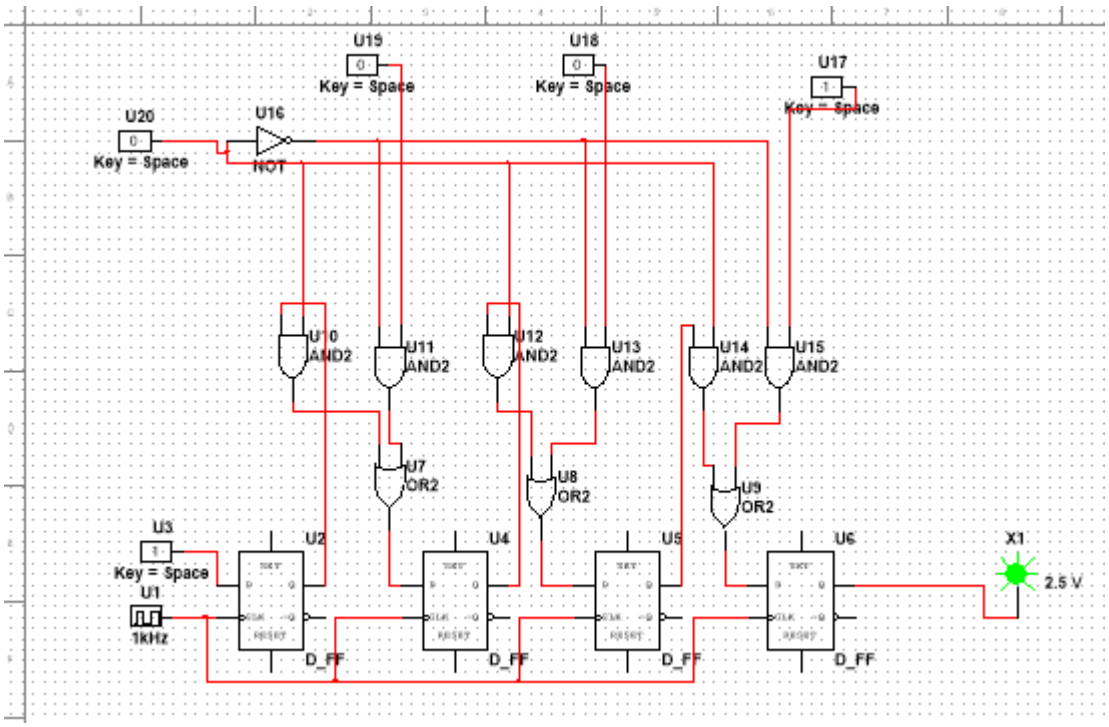
D_A	D_B	D_C	D_D
1	1	0	1

CLK Pulse	Q_A	Q_B	Q_c	Q_D (Data Output)
0	0	0	0	0
1	1	1	0	1
2	0	1	1	0
3	0	0	1	1
4	0	0	0	1

Procedure:

1. Place all d flipflops in parallel
2. Connect the inputs throught and , or gates and then connect them to flipflops
3. Connect a clock to all the flipflops
4. Connect the probes to the outputs.
5. Verify the truthtable.

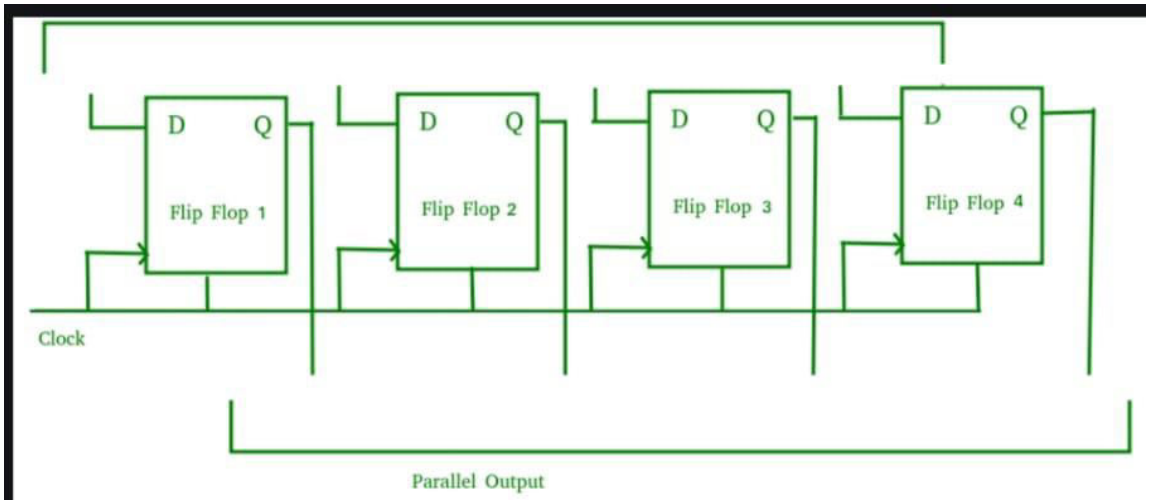
Multisim Circuit:



PIPO REGISTER:

A PIPO register, which stands for "Parallel In, Parallel Out," is a type of digital circuit that allows data to be loaded simultaneously (in parallel) at all input bits and outputs all bits simultaneously

Circuit Diagram:



Truth table:

CLK Pulse	QA	QB	QC	QD
0	0	0	0	0
1	1	1	0	1

Procedure:

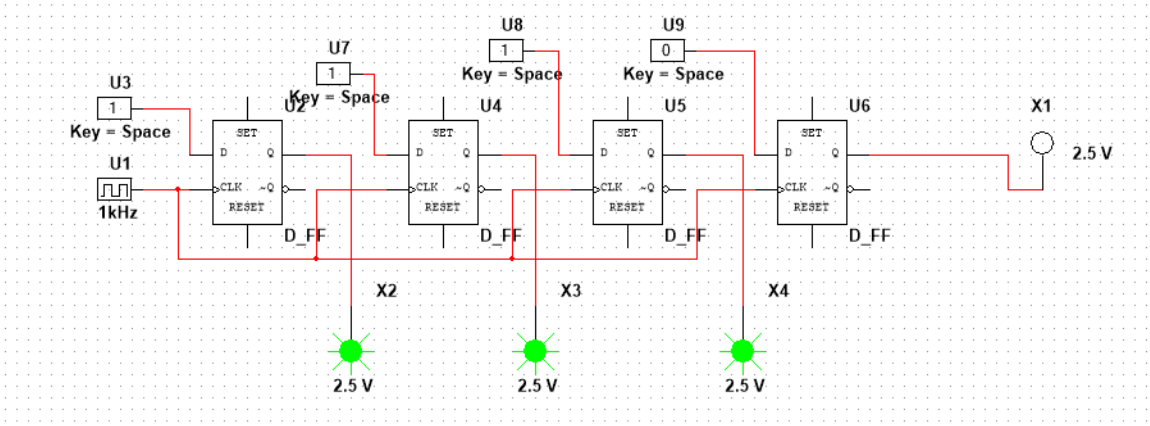
1. place multiple D flip-flops in parallel

2. connect their data inputs to the parallel data input lines

3. connect their clock inputs to a single clock signal, with the outputs of the flip-flops forming the parallel output

4. essentially, each flip-flop represents one bit in the register, allowing simultaneous data input and output on multiple lines

Multisim Circuit:


U3
1
Key = Space
U1
1kHz
U7
1
Key = Space
U8
1
Key = Space
U9
0
Key = Space
U2
U4
U5
U6
SET
D
Q
CLK
~Q
RESET
D_FF
X1
2.5 V
X2
X3
X4
2.5 V
2.5 V
2.5 V

IDNO:N210217

NAME:Shaik Akhila

LAB-10

Aim:Design different types of counters and verify their truthtables.

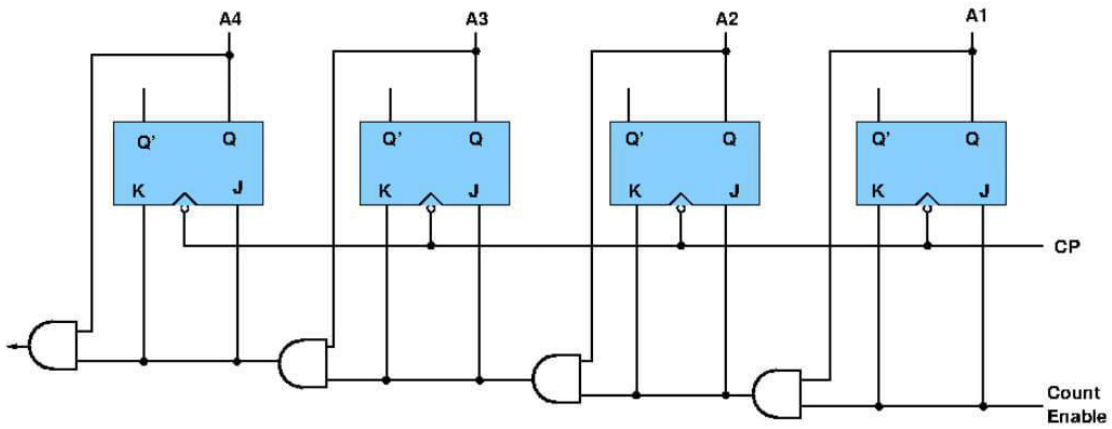
Apparatus*:*

4 D flipflops,4 jk flipflops, and gates ,,Connecting wires,probes/LED's,digital clock and Digital interactive constants for inputs.

4 bit synchronous binary counter :

A counter is a sequential circuit that moves through a predefined sequence of states upon applying of clock pulses. The sequence of states may follow the binary number sequence or an arbitrary manner (no sequence). The simplest example of a counter is the binary counter which follows the binary number sequence. An n-bit binary counter contains n flip-flops and can count binary numbers from 0 to (2n -1)(up counter which is incremental, if it counts decrementally it is then down counter).

Circuit Diagram:



Truth table:(up counter)

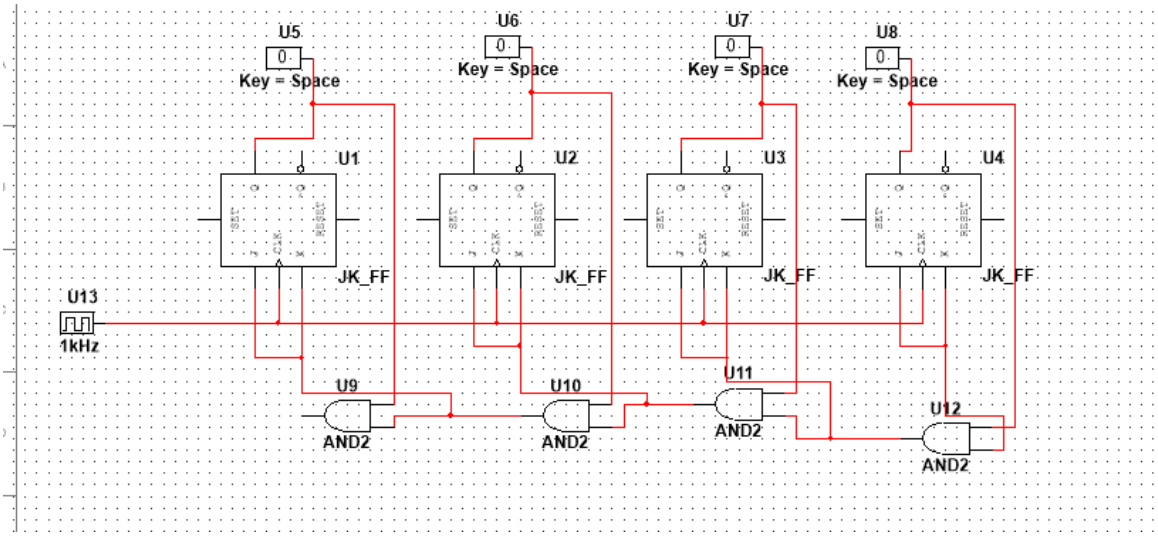
Clock/Time	A_4	A_3	A_2	A_1
1	0	0	0	0
2	0	0	0	1
3	0	0	1	0
4	0	0	1	1
5	0	1	0	0
6	0	1	0	1
7	0	1	1	0
8	0	1	1	1

Clock/Time	A_4	A_3	A_2	A_1
9	1	0	0	0
10	1	0	0	1
11	1	0	1	0
12	1	0	1	1
13	1	1	0	0
14	1	1	0	1
15	1	1	1	0
16	1	1	1	1

Procedure:

1. Use 4 JK Flip-Flops for the 4-bit counter.
2. Connect the clock to all flip-flops' clock inputs.
3. Set J = K = 1 for all flip-flops to toggle.
4. Connect the Q output of each flip-flop to the clock of the next flip-flop in the sequence.

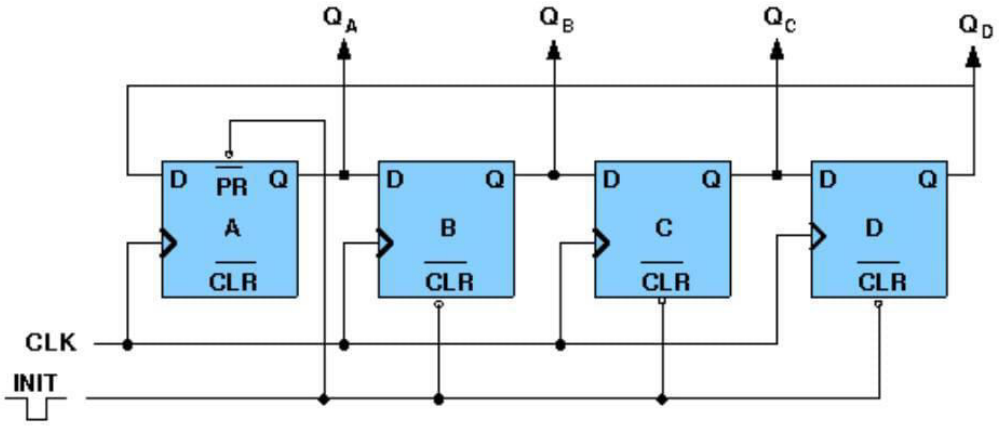
Multisim Circuit:



4 bit synchronous ring counter:

If the output of a shift register is fed back to the input. a ring counter results. The data pattern contained within the shift register will recirculate as long as clock pulses are applied.

Circuit Diagram:



TRUTHTABLE:

Clock pulse	Q_A	Q_B	Q_C	Q_D
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	0	0	0

Procedure:

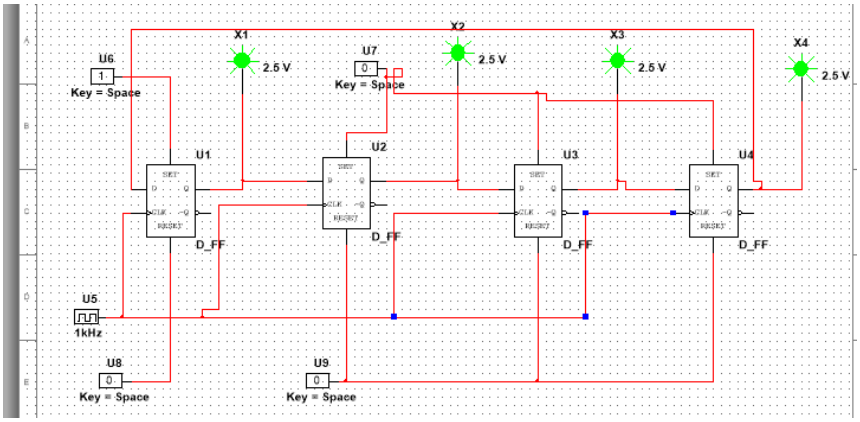
1.Add 4 D flip-flops from the digital components toolbar.

2.Connect Q1 → D2, Q2 → D3, Q3 → D4, Q4 → D1.

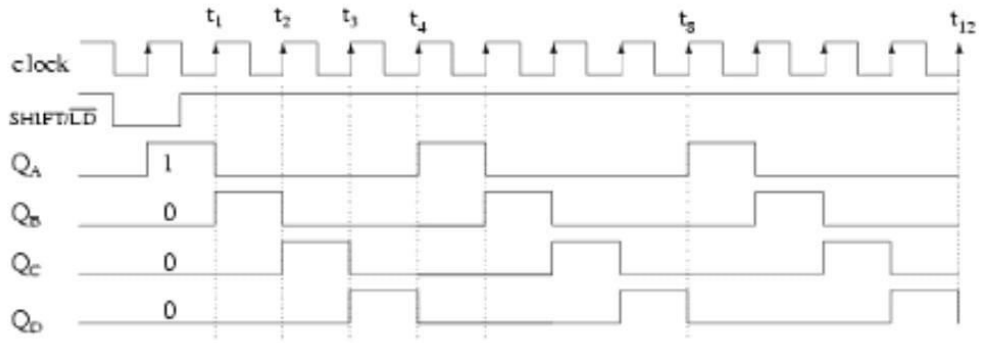
3.Connect a common clock signal to the CLK inputs of all flip-flops

4.Set set of D1 to '1' and D2, D3, D4 to '0'. Observe the output with LEDs or a logic probe

Multisim Circuit:



Timing diagram:

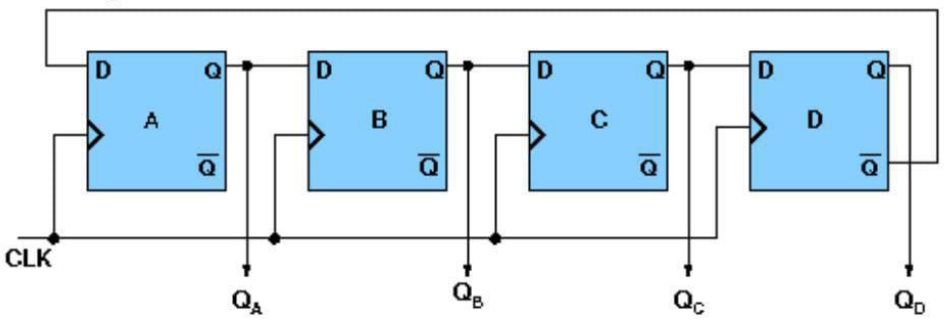


Load 1000 into 4-stage ring counter and shift

4 bit synchronous johnson counter:

If the complement output of a ring counter is fed back to the input instead of the true output, a Johnson counter results. This "reversed" feedback connection has a profound effect upon the behavior of the otherwise similar circuits. Recirculating a single 1 around a ring counter divides the input clock by a factor equal to the number of stages. Whereas, a Johnson counter divides by a factor equal to twice the number of stages.

Circuit Diagram:



Truth table:

Clock pulse	Q_A	Q_B	Q_C	Q_D
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1

Procedure:

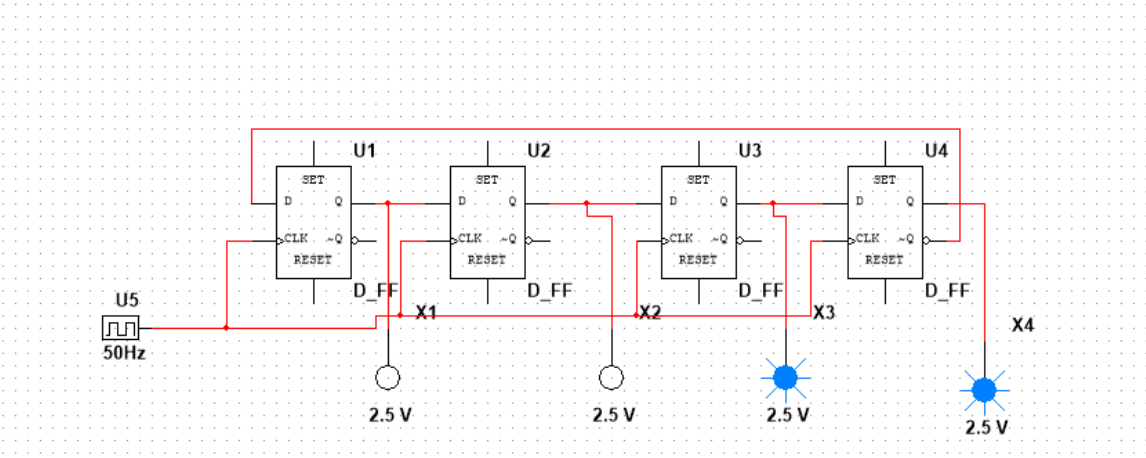
1.Add 4 D flip-flops from the digital components toolbar.

2.Connect Q1 → D2, Q2 → D3, Q3 → D4, Q4 bar → D1.

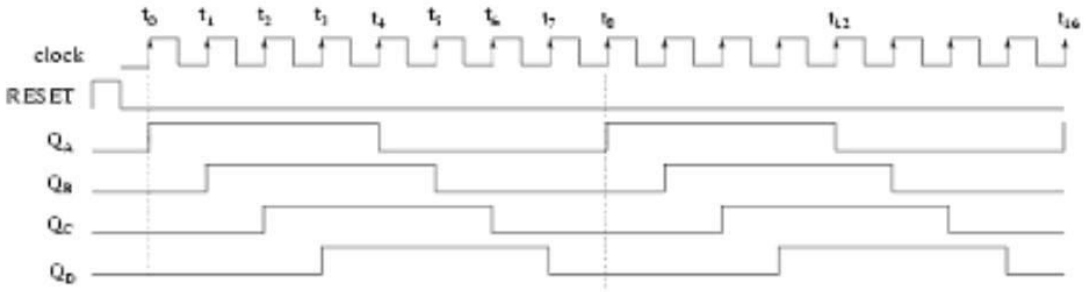
3.Connect a common clock signal to the CLK inputs of all flip-flops

4.Observe the output with LEDs or a logic probe

Multisim Circuit:



Timing diagram:



Four stage Johnson counter waveforms

ID no : N210217

Name: Shaik.Akhila

LAB-10

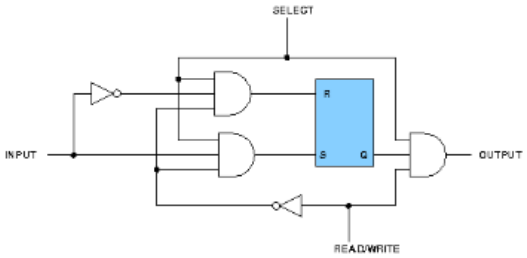
4*4 RAM cell:

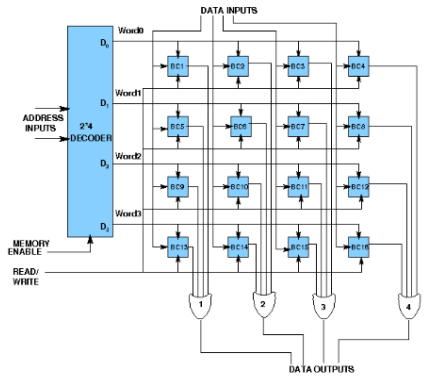
Description:

The binary cell has three inputs and one output. The select input enables the cell for reading or writing and the reda/write input determines the cell operation when it is selected. A 1 in the read/write input provides the read operation by forming a path from the flip-flop to the output terminal. A 0 in the read/write input provides the write operation by forming a path from the input terminal to the flip-flop

A memory with 4 words needs two address lines. The two address inputs go through a 2*4 decoder to select one of the four words. The decoder is enabled with the memory enable input. When the memory enable is 0, all outputs of the decoder are 0 and none of the memory words are selected. With the memory enable at 1, one of the four words is selected, dictated by the value in the two address lines. Once a word has been selected,the read/write input determines the operation.

Circuit Diagram:





Truth Table:

Enable	Read/Write	Data Input	Operation
0	x	x	No operation
1	0	0	write 0 to flipflop
1	0	1	Write 1 to flipflop
1	1	x	Read stored value from flipflop

Components required:

To build a RAM Cell, we need :

3 AND Gates , 2 NOT gates, SR Flip Flop

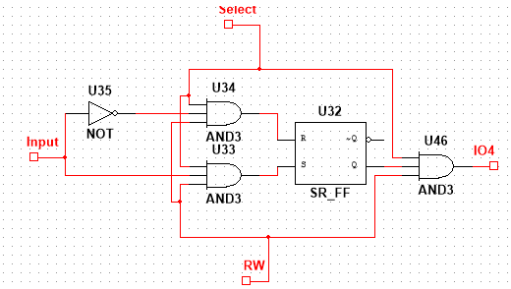
To build a 4X3 RAM, we need :

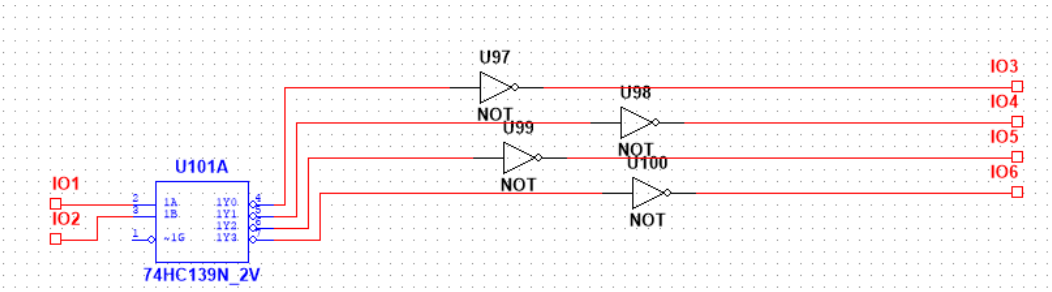
OR Gates , RAM Cell-12 , 2X4 Decoder with Enable

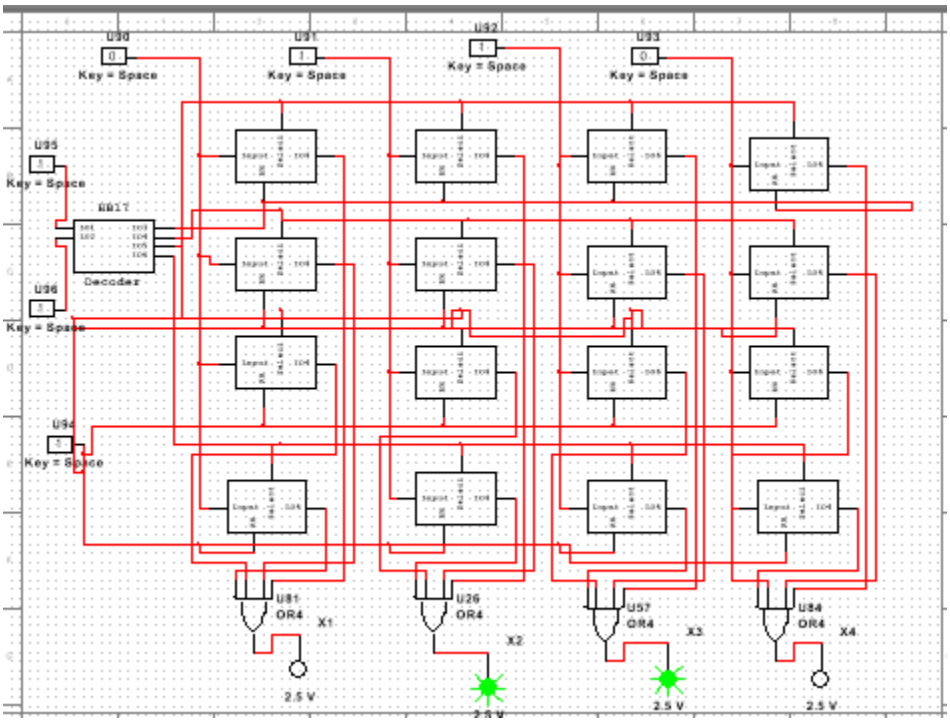
Procedure:

1) Open a new blank schematic in multisim and place all the components required.
2) Place 16 binary cells cells in a 4x4 grid.
3) Add a 2 into 4 decoder.
4) Connect decoder outputs to Select inputs of each row.
5) Connect columns of bianry cells to vertical lines.
6) Place or gates at the bottom, connect to bit lines.
7) Connect all the other components and verify the circuit truth table.

Multisim circuit:




U97
NOT
U98
NOT
U99
NOT
U100
NOT
IO3
IO4
IO5
IO6
IO1
IO2
U101A
74HC139N_2V


U30
Key = Space
U31
Key = Space
U32
Key = Space
U33
Key = Space
U35
Key = Space
U36
Key = Space
U34
Key = Space
U81
OR4
X1
U26
OR4
X2
U57
OR4
X3
U84
OR4
X4
2.5 V
2.5 V
2.5 V
2.5 V