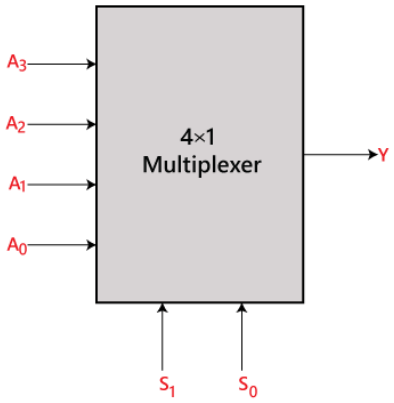
Experiment – 3

Design of basic blocks

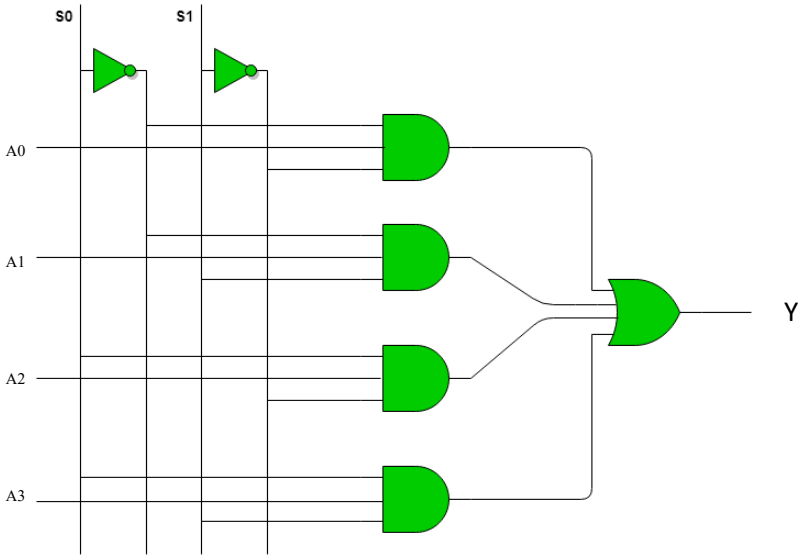
4 x 1 Multiplexer

A multiplexer is a combinational circuit that has many data inputs and a single output, depending on control or select inputs. Input signal is propagated to 2^n bits where n is the number of selected line inputs.



4 x 1 MUX

The 4 x 1 MUX has 4-bit input driven into the output by 2 select lines. Each input combination being given to the select lines results in an input line being connected to the output.

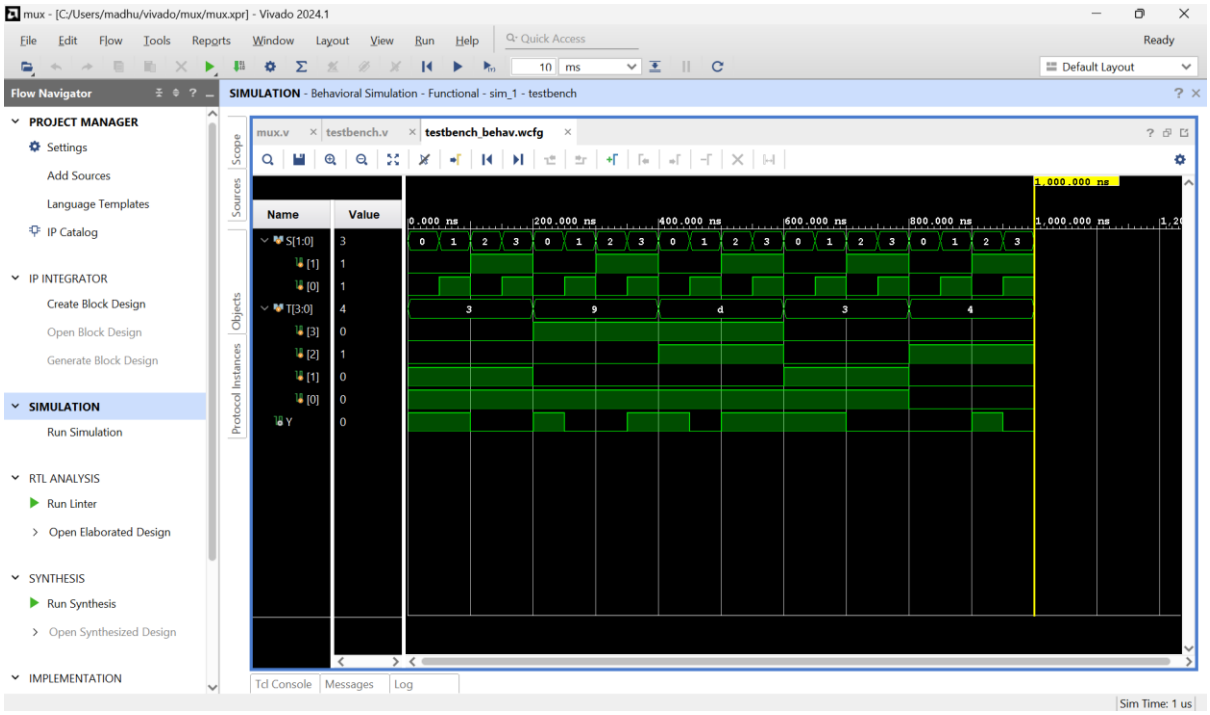


Gate level circuit of 4 x 1 MUX

The output function of the circuit is

$$Y = S_1' S_0' A_0 + S_1' S_0 A_1 + S_1 S_0' A_2 + S_1 S_0 A_3$$

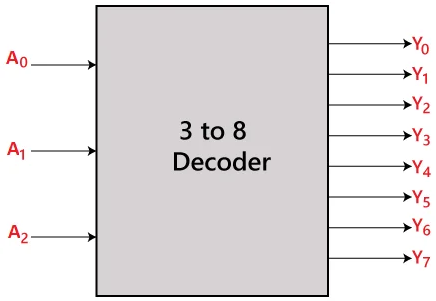
The testbench is provided with 5 inputs for input A and each of the 4 input combinations of select lines are plotted using Verilog.



Output graph of behavioural simulation

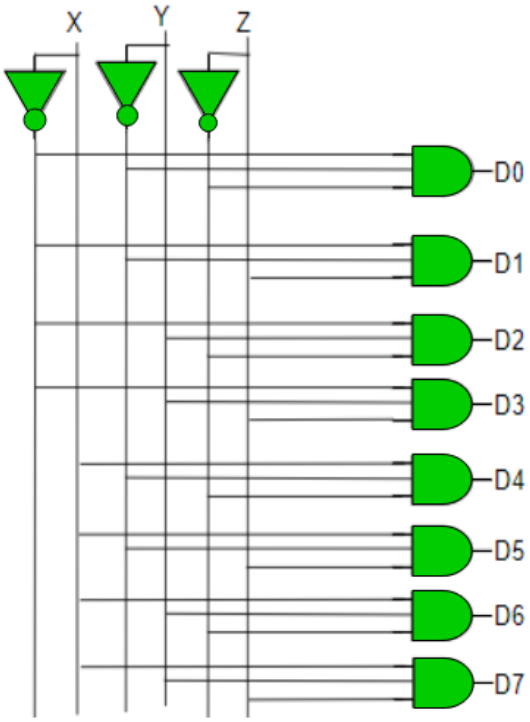
3 to 8 Decoder

A decoder is a combinational logic circuit that is used to change the code into a set of signals. It is the reverse process of an encoder. A decoder circuit takes multiple inputs and gives multiple outputs. The circuit contains n-bit input lines and 2^n bits of output pins.



3 to 8 decoders

A 3 to 8 decoder has 3 input lines and 8 output lines. The gate level circuit is as below. X, Y, and Z represent the 3-bit binary inputs. D0 to D7 are the outputs which are activated with each of the input bit combinations. Each input combination provides a ubique output and an output is mapped to a single input combination.



Gate level realization of 3 to 8 decoder

The output function of the decoder is as follows.

$$D_0 = X' Y' Z'$$

$$D_1 = X' Y' Z$$

$$D_2 = X' Y Z'$$

$$D_3 = X' Y Z$$

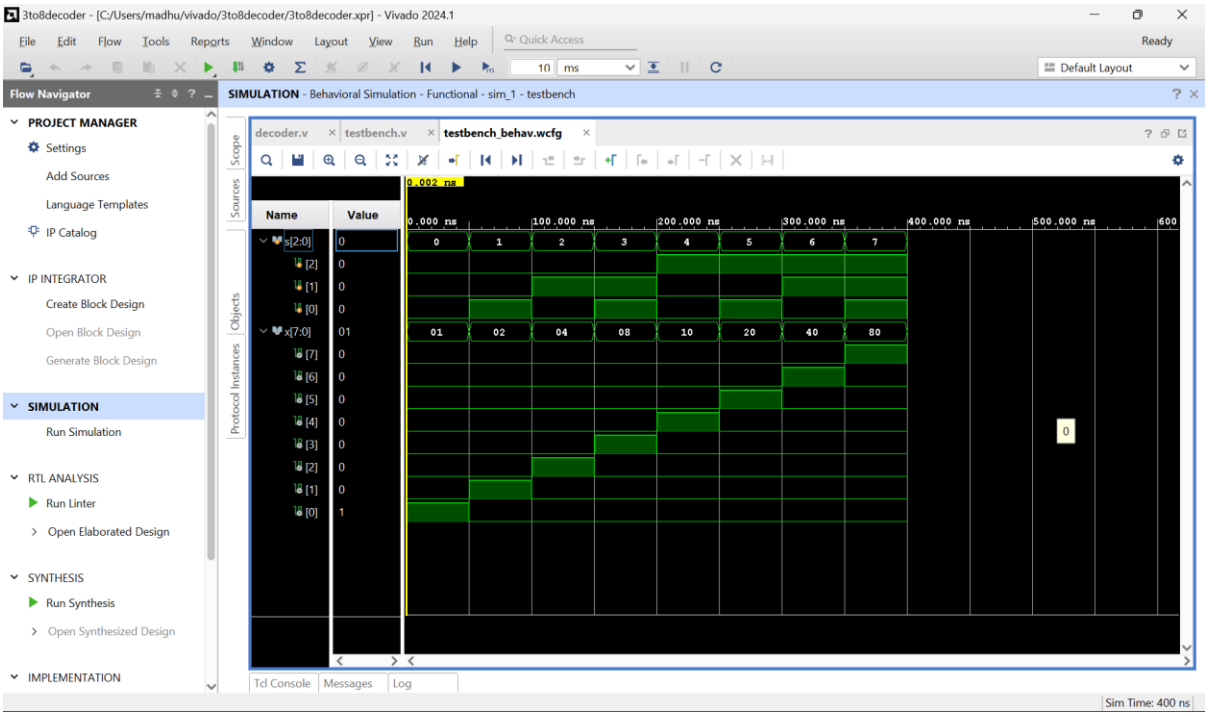
$$D_4 = X Y' Z'$$

$$D_5 = X Y' Z$$

$$D_6 = X Y Z'$$

$$D_7 = X Y Z$$

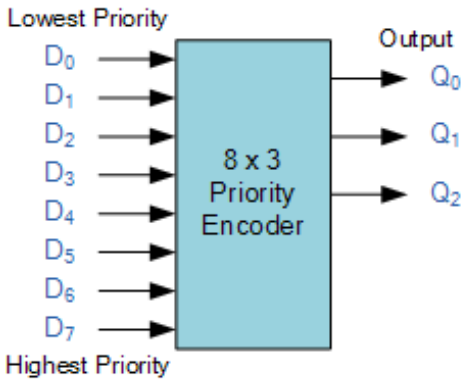
The testbench is provided with 8 input combinations for input X, Y, Z and output in each state is plotted in graph



Output graph of behavioural simulation

Priority Encoder

A priority encoder is a type of encoder which gives binary output of the highest priority input bit. The module consists of 2^n input bits and n-bit output bit.



8 to 3 priority encoders

D7 to D0 are input bits where the priority is highest for D7 and least for D0. Each input pin is associated to the output with the logic algorithm generating 3-bit binary output corresponding to the bit with highest priority turned on. The output does not change when logic state of a bit with smaller priority is changed while a higher priority bit is on and generating output.

Digital Inputs								Binary Output		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Q_2	Q_1	Q_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	X	0	0	1
0	0	0	0	0	1	X	X	0	1	0
0	0	0	0	1	X	X	X	0	1	1
0	0	0	1	X	X	X	X	1	0	0
0	0	1	X	X	X	X	X	1	0	1
0	1	X	X	X	X	X	X	1	1	0
1	X	X	X	X	X	X	X	1	1	1

Truth table of the 8 to 3 priority encoders

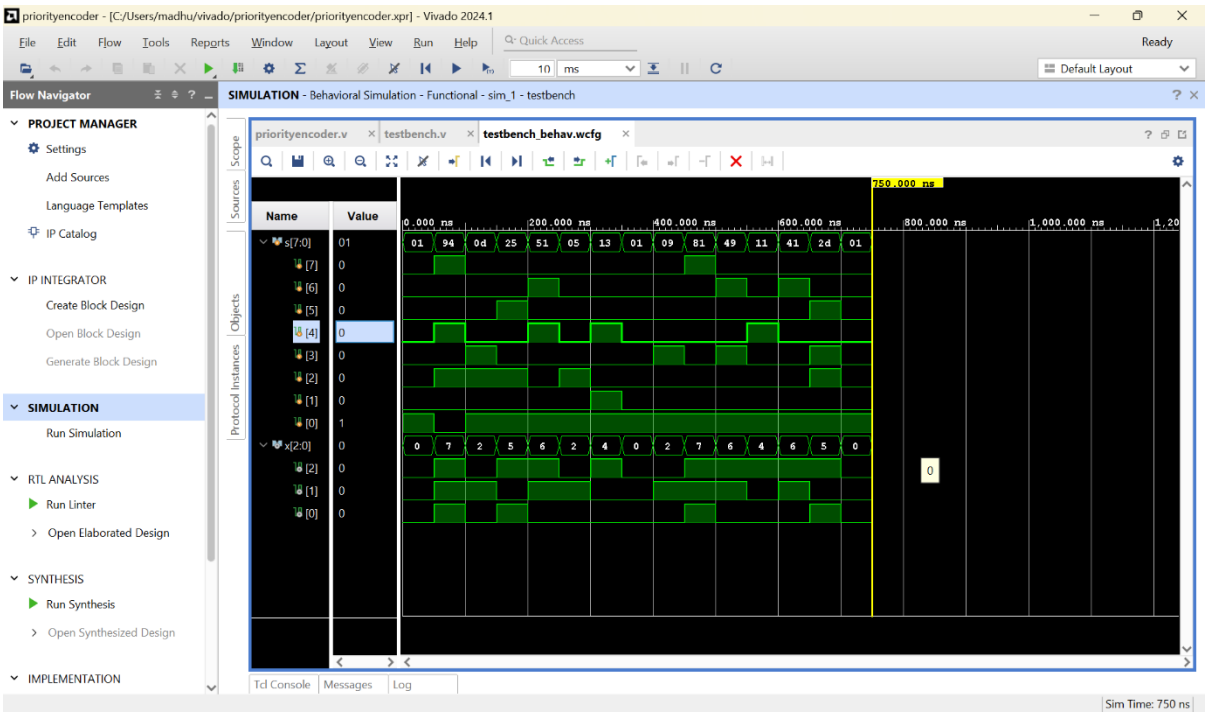
The output function of the encoder is

$Q_0 = D_6' (D_4' D_2' D_1 + D_4' D_3 + D_5) D_7$

$Q_1 = D_5' D_4' (D_2 + D_3) + D_6 + D_7$

$Q_2 = D_4 + D_5 + D_6 + D_7$

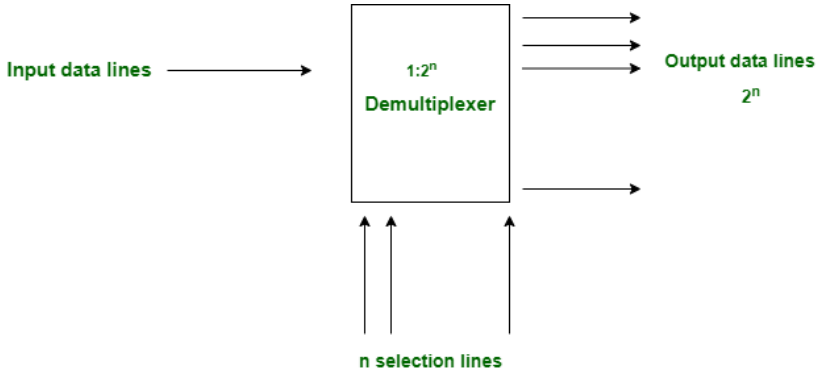
Testbench is designed with various 8-bit inputs and the output is verified using graphical representation.



Output graph of behavioural simulation

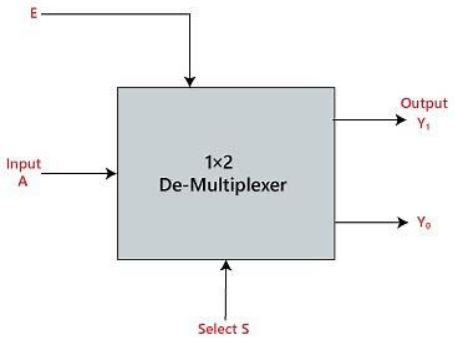
1 x 16 Demultiplexer

A 1 to 2^n demultiplexer/demux is a logical device which transmit the input signal into a specific output according to n-bit select line inputs. Demux are available in 1x2, 1x4, 1x8, 1x16 and more variations. A 1 to 2^n demux can also be designed using a combination of a 1x2 demux and two 1x2^(n-1) demux. We will be designing a 1x16 demux using 1x2 demux and two 1x8 demux.



1x2^n Demux

1x2 demux: The demux has 1 data input, one select input and two output lines.



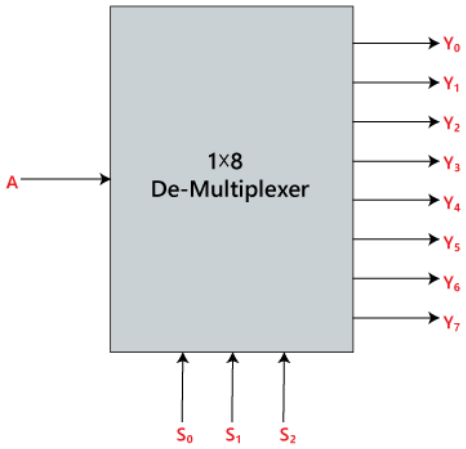
The truth table and output equation are

INPUTS	Output	
S_0	Y_1	Y_0
0	0	A
1	A	0

$Y_0 = S_0' A$

$Y_1 = S_0 A$

1x8 Demux: The demux has 1 data input, three select input and eight output lines



The truth table and output equation are

INPUTS			Output							
S_2	S_1	S_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	A
0	0	1	0	0	0	0	0	0	A	0
0	1	0	0	0	0	0	0	A	0	0
0	1	1	0	0	0	0	A	0	0	0
1	0	0	0	0	0	A	0	0	0	0
1	0	1	0	0	A	0	0	0	0	0
1	1	0	0	A	0	0	0	0	0	0
1	1	1	A	0	0	0	0	0	0	0

$$Y_0 = S_0' S_1' S_2' A$$

$$Y_1 = S0\ S_1' S_2' A$$

$$Y_2 = S0'\ S_1 S_2' A$$

$$Y_3 = S0\ S_1 S_2' A$$

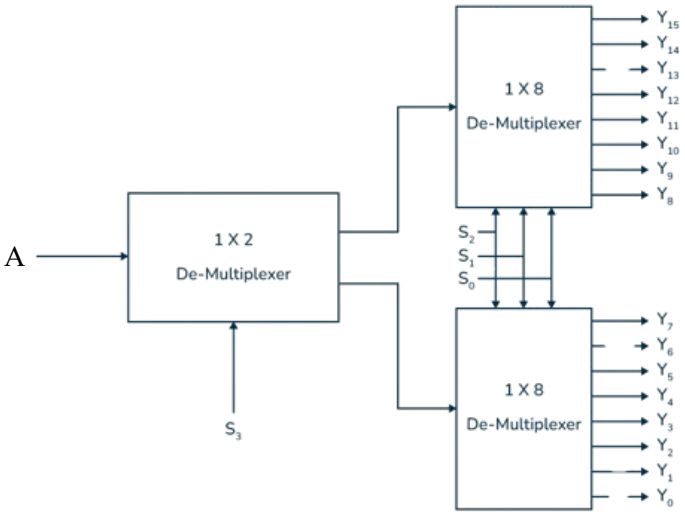
$$Y_4 = S0'\ S_1' S_2 A$$

$$Y_5 = S0\ S_1' S_2 A$$

$$Y_6 = S0'\ S_1 S_2 A$$

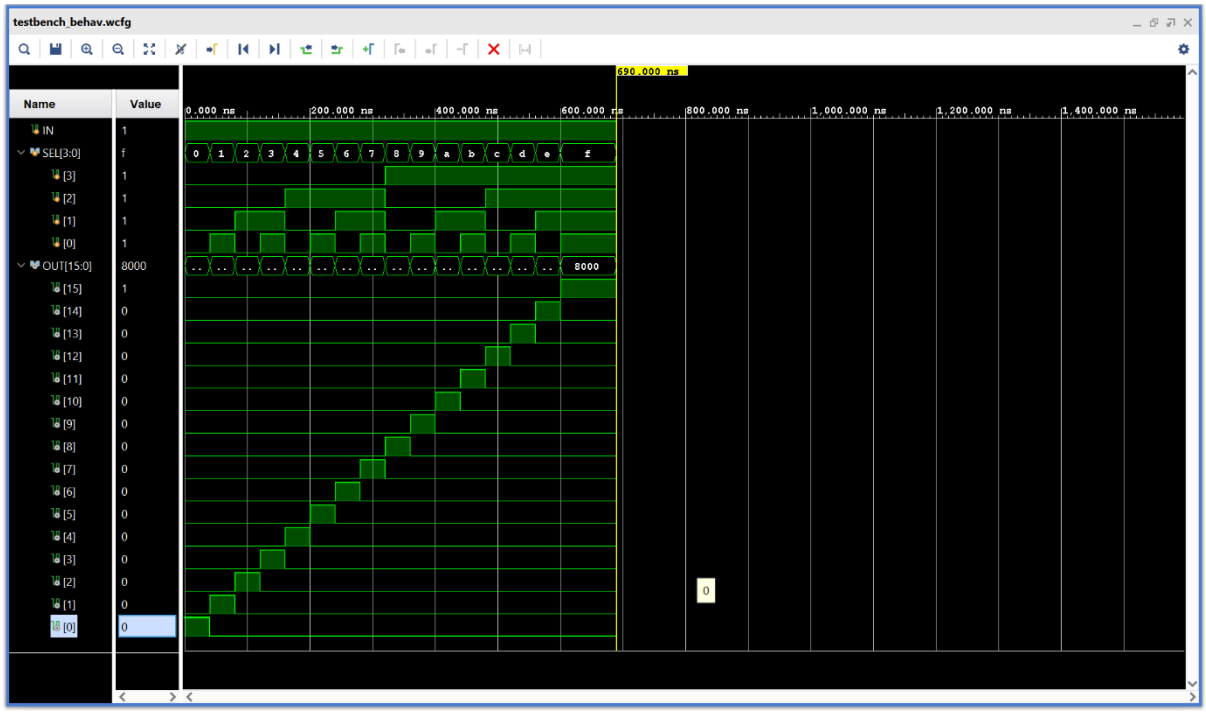
$$Y_7 = S0\ S_1 S_3 A$$

A 1x16 demux is built by connecting a 1x2 demux and two 1x8 demux.



1x16 demux using 1x2 and 1x8 demux block diagram

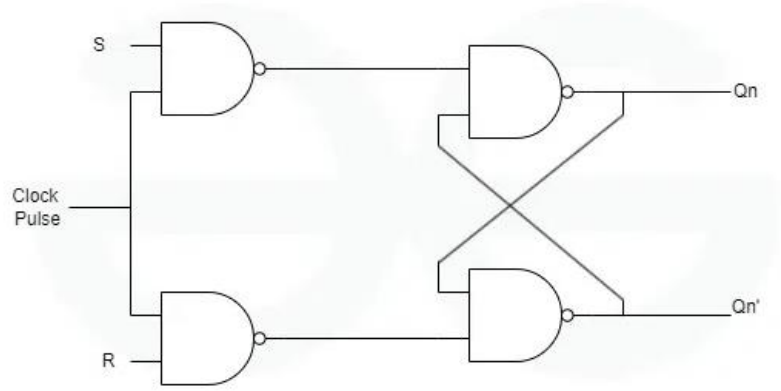
A 1x2 demux and 1x8 demux is implemented in Verilog and a higher-level module is created using the 1x2 demux and 1x8 demux as lower-level modules. Finally, a testbench is created with inputs from 1 to 16 provided using select lines. The output is verified from graph obtained from behavioural simulation.



Output graph of behavioural simulation

HDL model for flipflops; S R flipflop

It is a Flip Flop with two inputs, one is S, and the other is R. S here stands for Set and R here stands for Reset. Set basically indicates set the flip flop which means output 1 and reset indicates resetting the flip flop which means output 0. Here, a clock pulse is supplied to operate this flip-flop, hence it is a clocked flip-flop.



Schematic of SR Flipflop

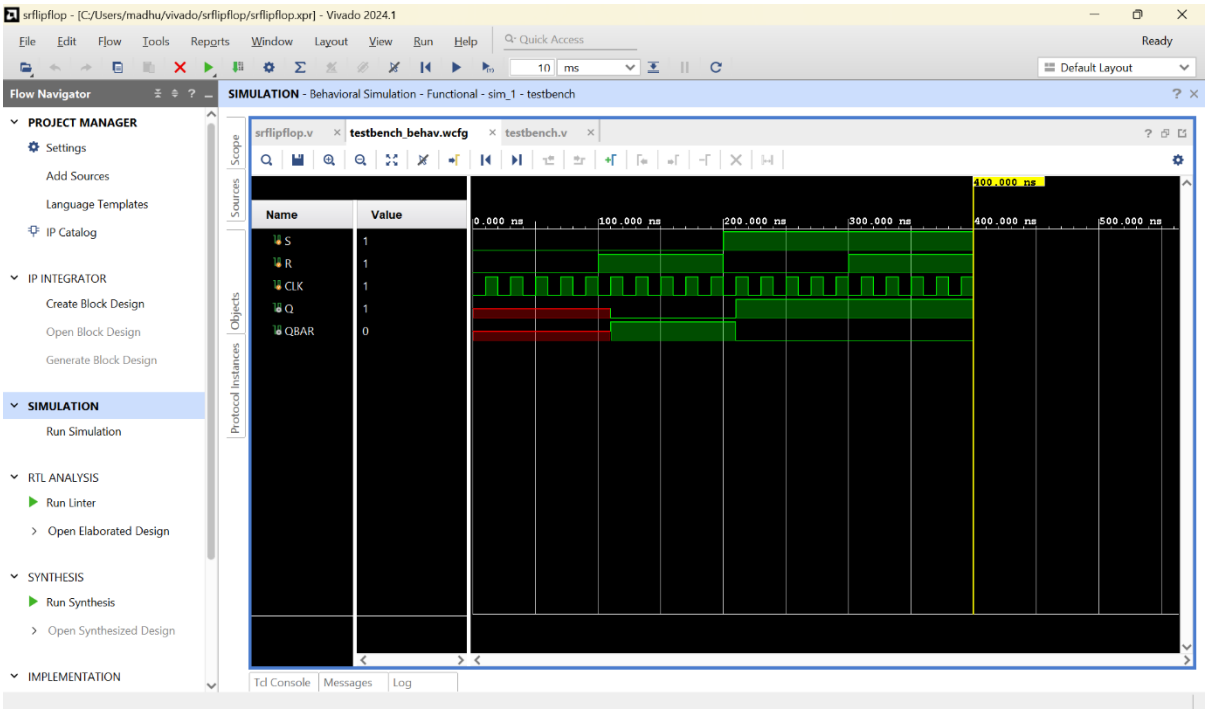
S	R	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	Invalid
1	1	1	Invalid

Function table

Q_n	Q_{n+1}	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

Excitation table

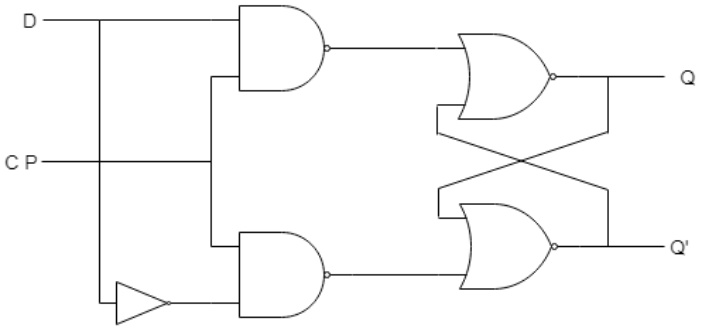
when S = 1, output Q becomes 1, and when R = 1, output Q becomes 0. The same state is continued when S and R are 0. An invalid state is the output while S and R are switched to 1 simultaneously. Testbench is made and output is obtained from behavioural modelling graph. Initial state is provided as Q = 0 and Q' = 1.



Output graph of behavioural simulation

HDL model for flipflops; D flipflop

A D flip-flop stands for data or delay flip-flop. The outputs of this flip-flop are equal to the inputs.

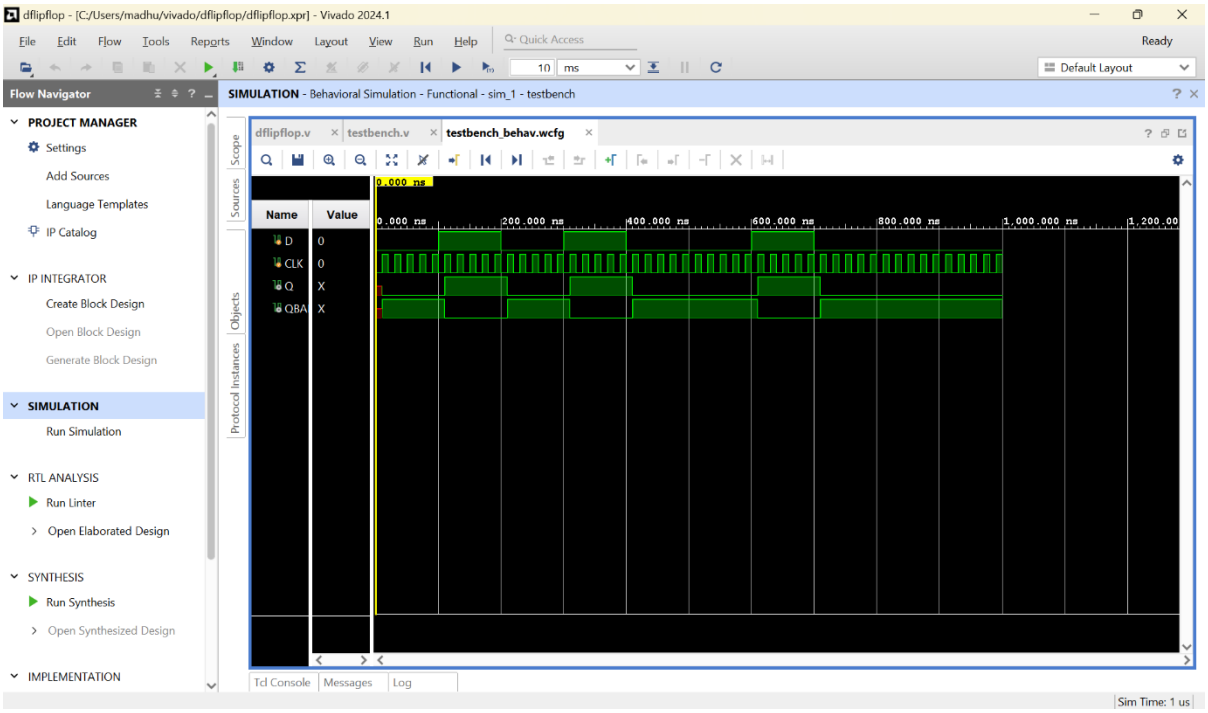


Schematic of D Flipflop

Qn	D	Qn+1
0	0	0
0	1	1
1	0	0
1	1	1

Excitation table

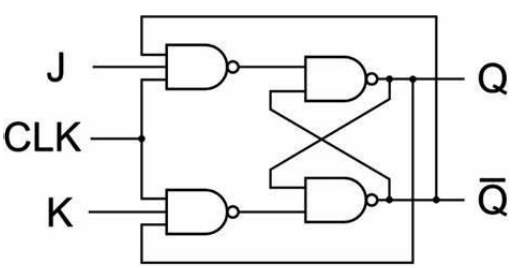
Output is equal to input at each clock edge input. Various input combinations are provided, and output is plotted in graph and analysed



Output graph of behavioural simulation

HDL model for flipflops; J K flipflop

A JK flip – flop is the modification of SR flip – flop with no illegal state. In this the J input is similar to the SET input of SR flip – flop and the K input is similar to the RESET input of SR flip – flop. The schematics of JK flip – flop is shown below.



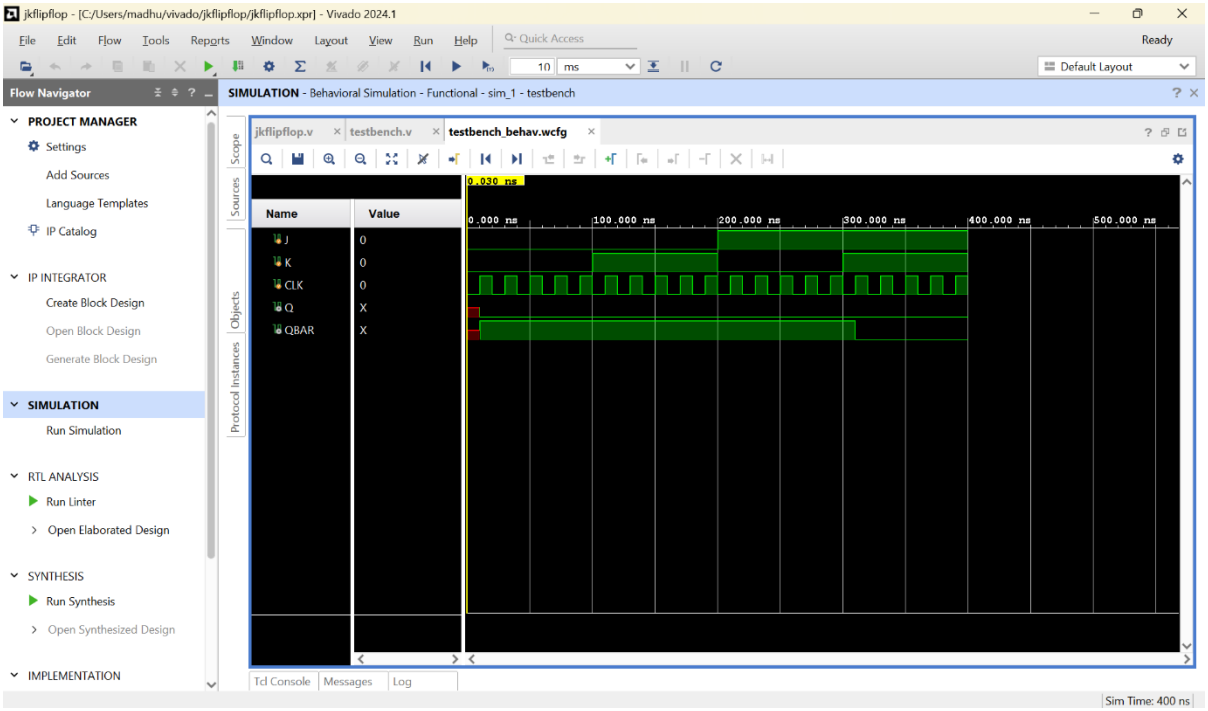
Inputs					Outputs		Comments
PR	CLR	CLK	J	K	Q(n+1)	Q'(n+1)	
0	1	NA	NA	NA	1	0	Set(Preset)
1	0	NA	NA	NA	0	1	Reset(Clear)
1	1	0	NA	NA	Q(n)	Q'(n)	Initial Stage
1	1	1	0	0	Q(n)	Q'(n)	Initial Stage
1	1	1	1	0	1	0	Set
1	1	1	0	1	0	1	Reset
1	1	1	1	1	Q'(n)	Q(n)	Toggle

Truth table

J	K	Q_n	Q_{n+1}	Remark
0	0	x	Qn	Hold state
0	1	x	0	Reset state
1	0	x	1	Set state
1	1	x	Qn'	Toggle state

Reduced truth table

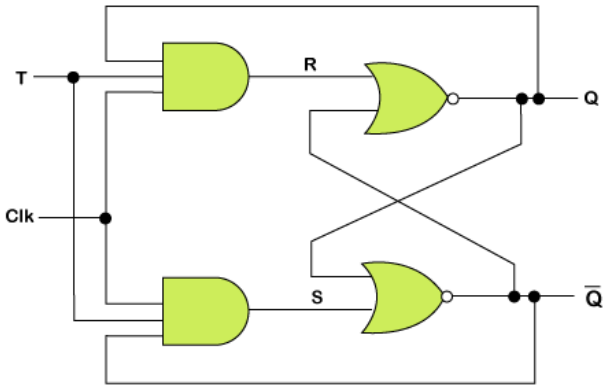
Verilog code is generated using reduced truth table. Output is obtained from behavioural analysis with testbench and different input combinations.



Output graph of behavioural simulation

HDL model for flipflops; T flipflop

T flip flop or Toggle Flip Flop is able to toggle its output depending upon on the input. Output is toggled from previous state while input is 1 at clock pulse. Output remains the same as previous state when input is 0 at the time of clock edge.

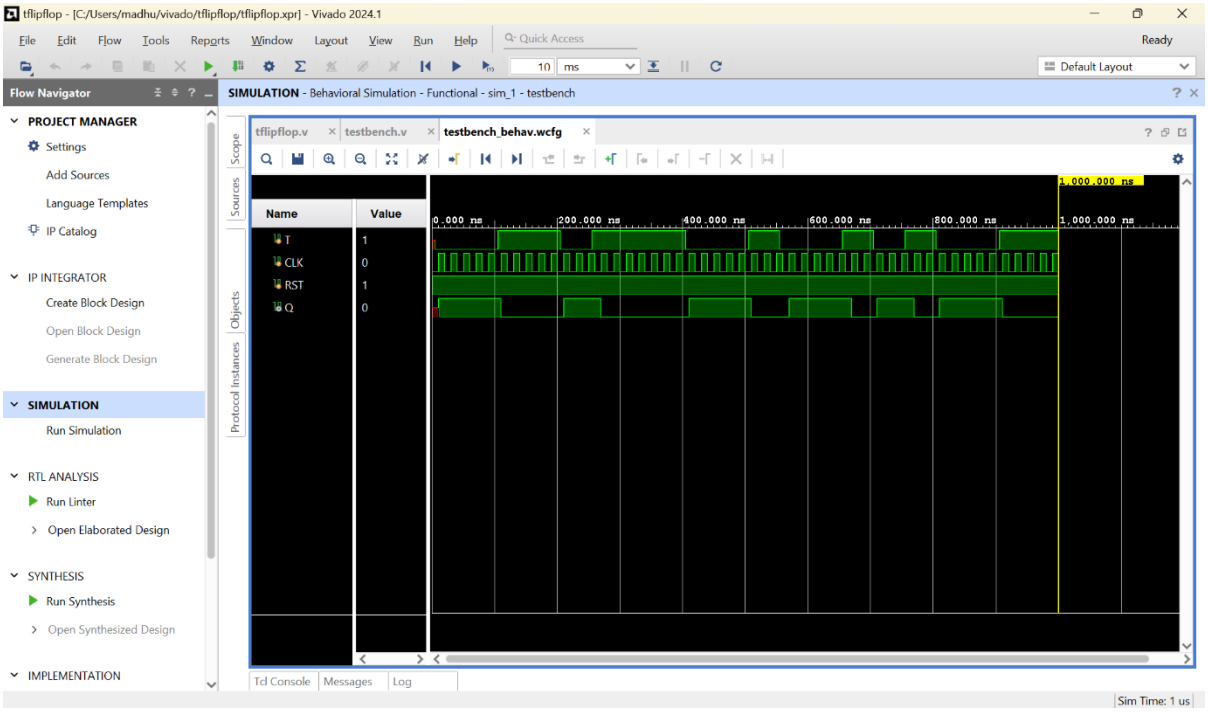


Schematic diagram

T	Qn	Qn+1
0	0	0
1	0	1
0	1	1
1	1	0

Excitation table

Testbench is obtained and different input combinations are provided. Initial state is set as Q = 0 and Q' = 1.



Output graph of behavioural simulation