**4-bits Ripple Carry Adder**

**Lab no# 02**

****

Spring 2022

CSE-308L Digital Systems Design lab

Submitted by: **Ashfaq Ahmad**

Registration No: **19PWCSE1795**

Class Section: **B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Dr: Ma’am Madeha sheer**

**April** 10, 2022

**Department of Computer Systems Engineering**

**University of Engineering and Technology, Peshawar**

**Objectives:**

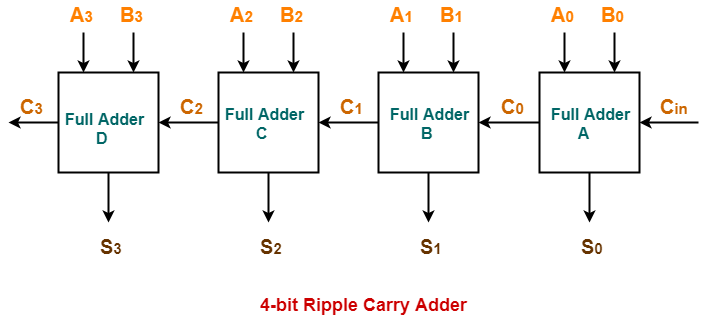
This lab will enable students to:

* + Learn top down and bottom up design methodologies
  + Data flow level modeling

**Task 01:**

Design 4 bits Ripple carry Adder using

1. Gate Level
2. Data flow level



1. **Gate Level**

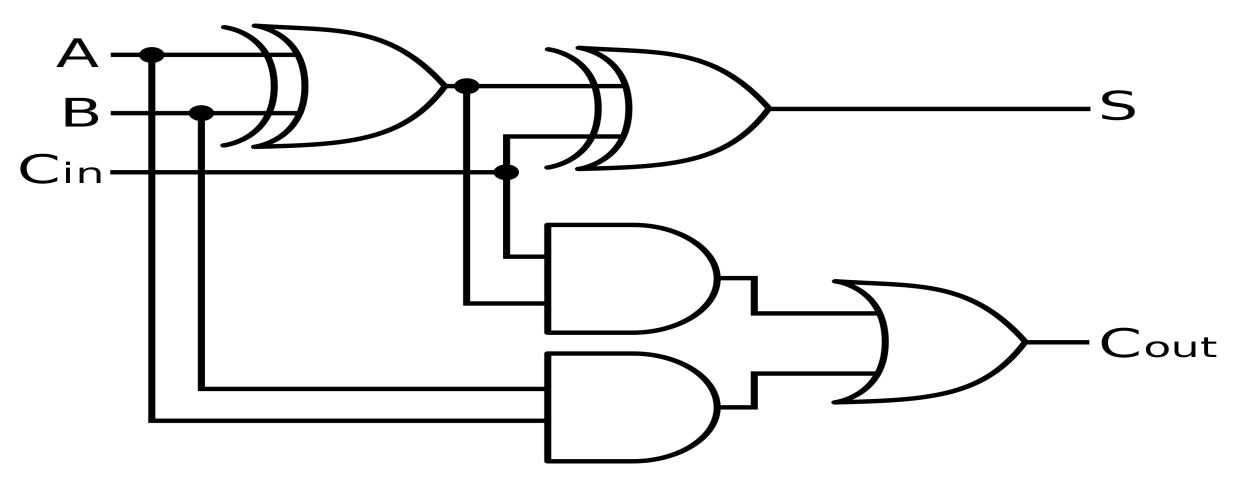
**Steps for task 1a:**

The following steps should be performed while designing a 4 bit RCA adder

**ModelSim:**

1. First implement a Full adder using data gate level modeling.
2. Simulate the Full adder with a test bench.
3. Instantiate the Full adder four times and connect the circuit as shown.
4. Now again write a test bench and simulate the 4 bit RCA.

**Full Adder:**



**Design Code:**

module FA(Cout,sum,A,B,Cin);

input A,B,Cin;

output sum,Cout;

wire w1,w2,w3;

xor x1(w1,A,B);

xor x2(sum,w1,Cin);

and a1(w2,A,B);

and a2(w3,Cin,w1);

or o1(Cout,w2,w3);

endmodule

**Test bench:**

module test\_bench1();

reg A,B,Cin;

wire sum,Cout;

FA td(Cout,sum,A,B,Cin);

initial

begin

#10 A=0;B=0;Cin=0;

#10 A=0;B=0;Cin=1;

#10 A=0;B=1;Cin=0;

#10 A=0;B=1;Cin=1;

#10 A=1;B=0;Cin=0;

#10 A=1;B=0;Cin=1;

#10 A=1;B=1;Cin=0;

#10 A=1;B=1;Cin=1;

end

initial

begin

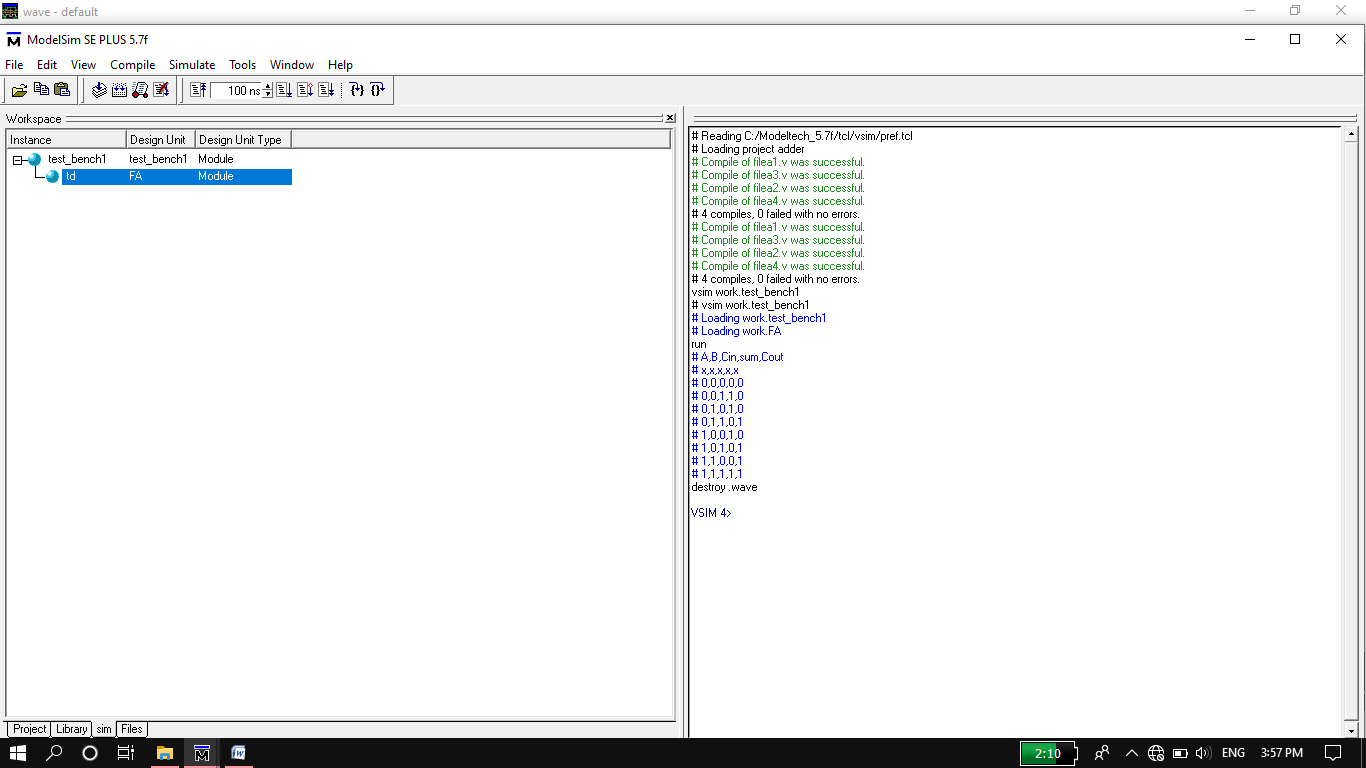
$display("A,B,Cin,sum,Cout");

$monitor("%b,%b,%b,%b,%b ",A,B,Cin,sum,Cout);

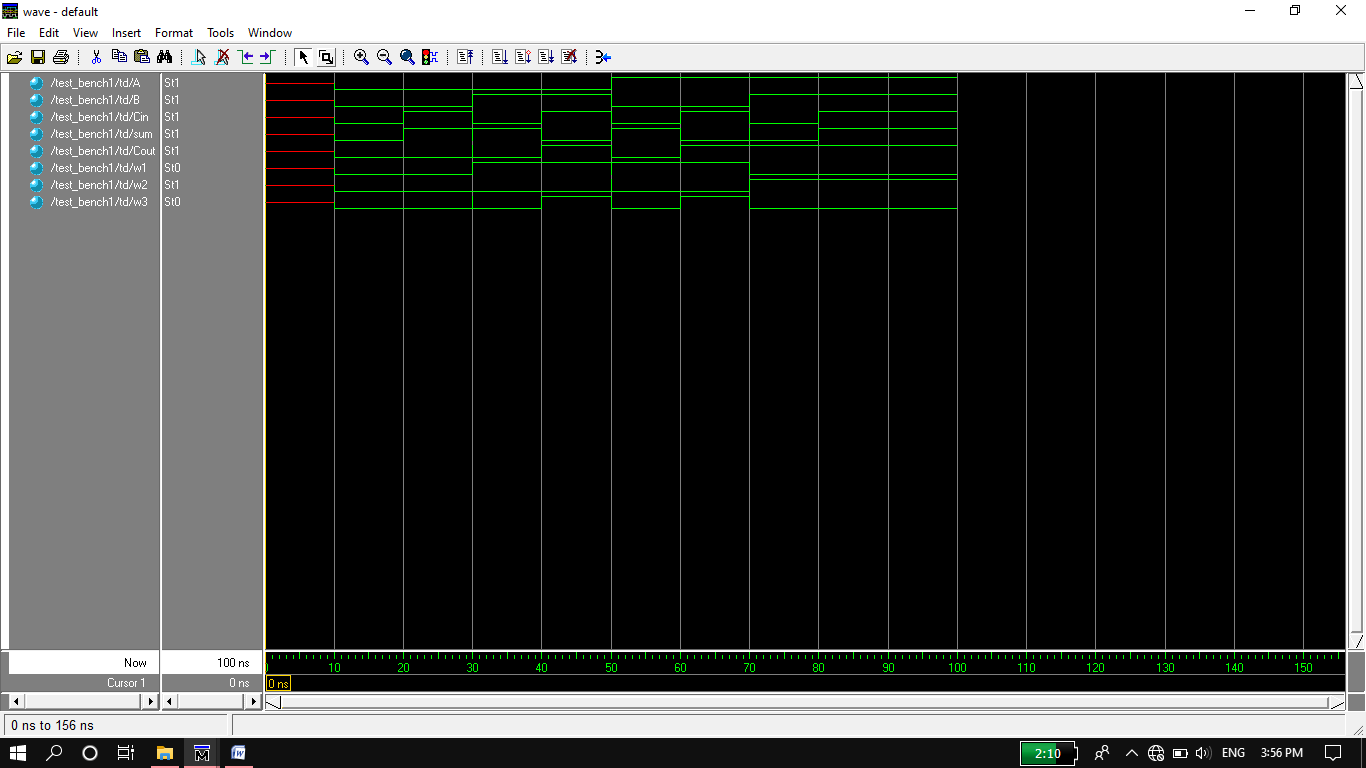
end

endmodule

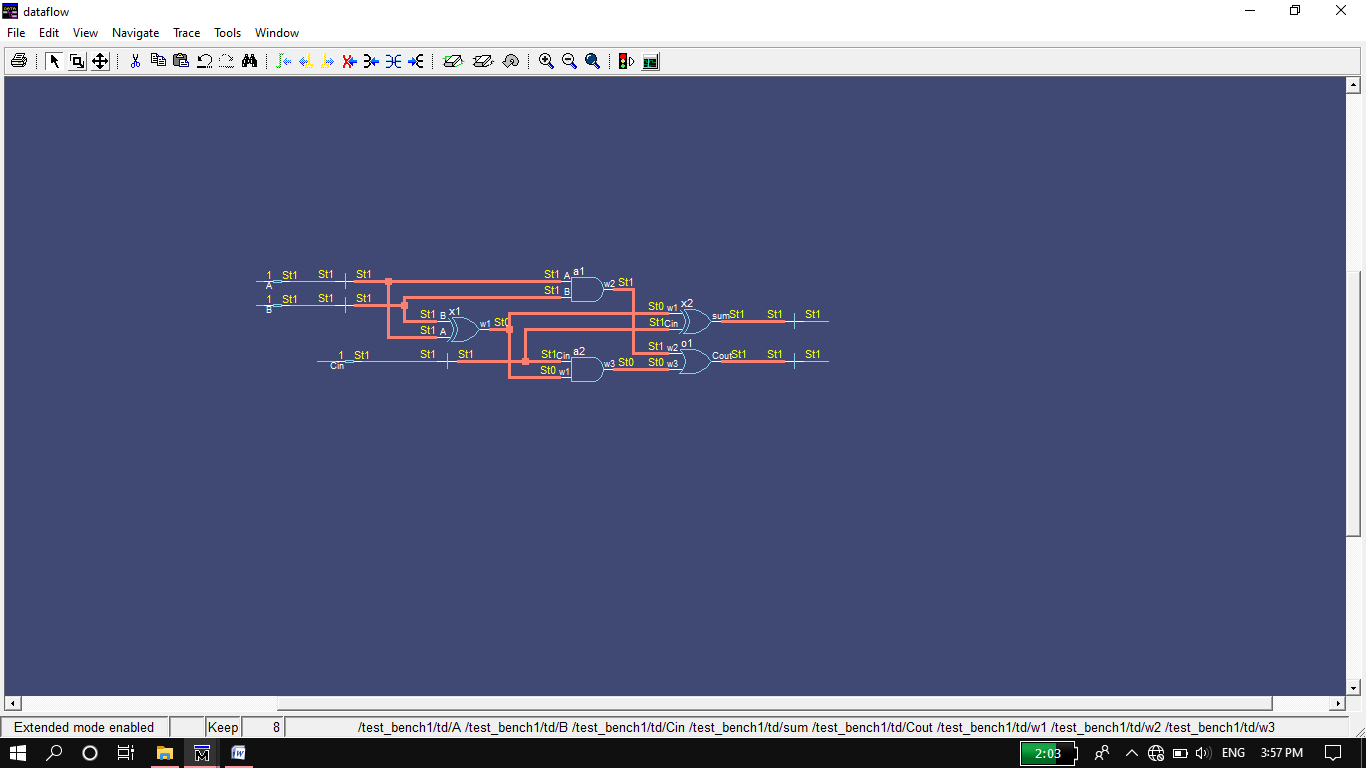
**Truth Table:**



**Wave Form:**



**Circuit :**



**4 Bits Ripple Carry Adder:**

**Design Code:**

module Ripplecarryadder(sum,Cout,A,B);

input [3:0] A,B;

output [3:0]sum;

output Cout;

wire [3:1]Cin;

FA f1(Cin[1],sum[0],A[0],B[0],1'b0); //1'b0=Cin0 1'b0 mean Cin is a 1 bit number which is 0.

FA f2(Cin[2],sum[1],A[1],B[1],Cin[1]);

FA f3(Cin[3],sum[2],A[2],B[2],Cin[2]);

FA f4(Cout,sum[3],A[3],B[3],Cin[3]);

endmodule

**Test Bench:**

module test\_bench2();

reg [3:0] A,B;

wire [3:0]sum;

wire Cout;

Ripplecarryadder td(sum,Cout,A,B);

initial

begin

$display("A B sum Cout");

$monitor("%d,%d,%d,%d",A,B,sum,Cout);

#10 A=4'b0000;B=4'b0000;

#10 A=4'b0001;B=4'b0001;

#10 A=4'b1100;B=4'b0011;

#10 A=4'b1010;B=4'b1001;

#10 A=4'b0010;B=4'b1000;

#10 A=4'b1100;B=4'b0110;

#10 A=4'b1110;B=4'b0111;

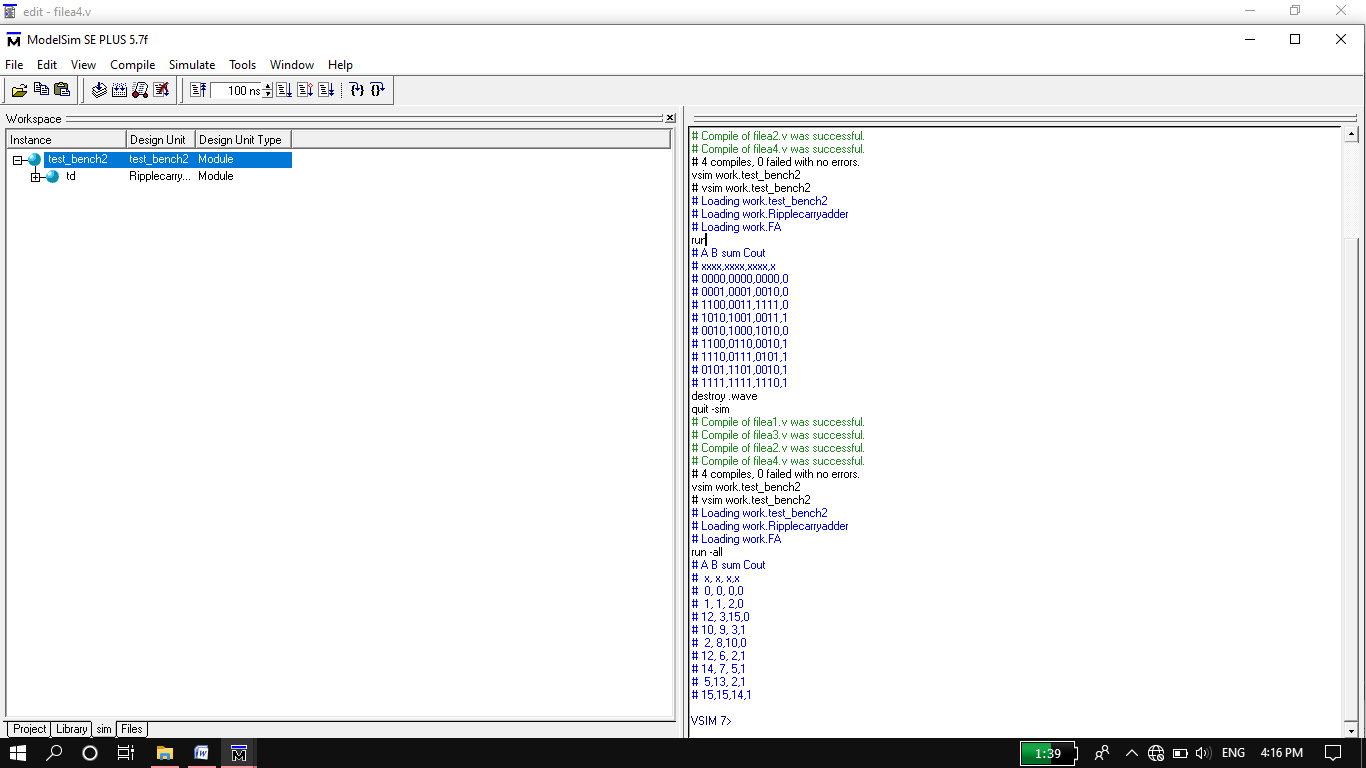
#10 A=4'b0101;B=4'b1101;

#10 A=4'b1111;B=4'b1111;

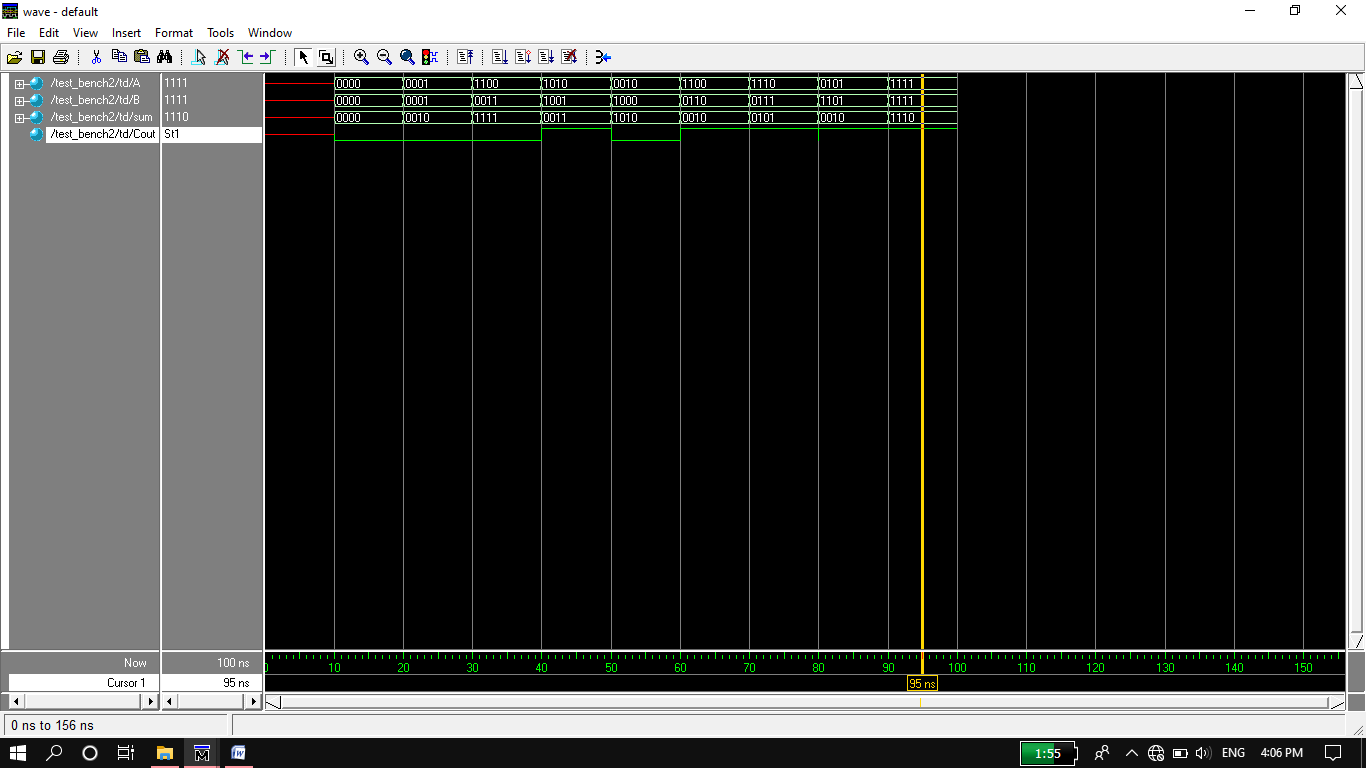
end

endmodule

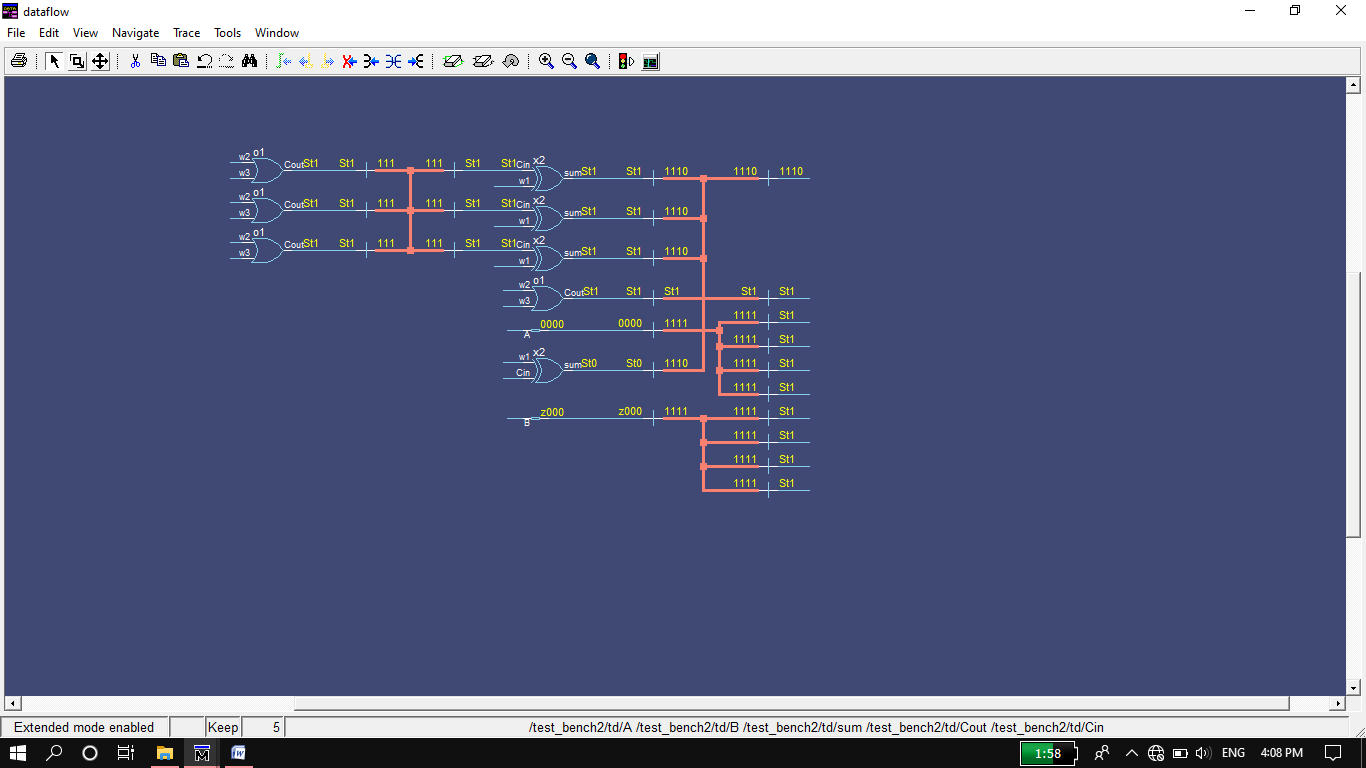
**Truth Table:**



**Wave From:**



**Circuit:**



1. **Data Flow Level:**

**Steps for task 1a:**

Same steps for Data flow level as for data gate level.

**ModelSim:**

1. First implement a Full adder using data flow level modeling.
2. Simulate the Full adder with a test bench.
3. Instantiate the Full adder four times and connect the circuit as shown.
4. Now again write a test bench and simulate the 4 bit RCA.

**Full Adder:**

**Data flow Code:**

module FA(sum,carry,A,B,Cin);

input A,B,Cin;

output sum,carry;

assign {carry,sum}=A+B+Cin;

endmodule

**Test bench:**

module test\_bench3();

reg A,B,Cin;

wire sum,carry;

FA td(sum,carry,A,B,Cin);

initial

begin

$display("A B Cin sum carry");

$monitor("%b,%b,%b,%b,%b",A,B,Cin,sum,carry);

#20 A=0;B=0;Cin=0;

#20 A=0;B=0;Cin=1;

#20 A=0;B=1;Cin=0;

#20 A=0;B=1;Cin=1;

#20 A=1;B=0;Cin=0;

#20 A=1;B=0;Cin=1;

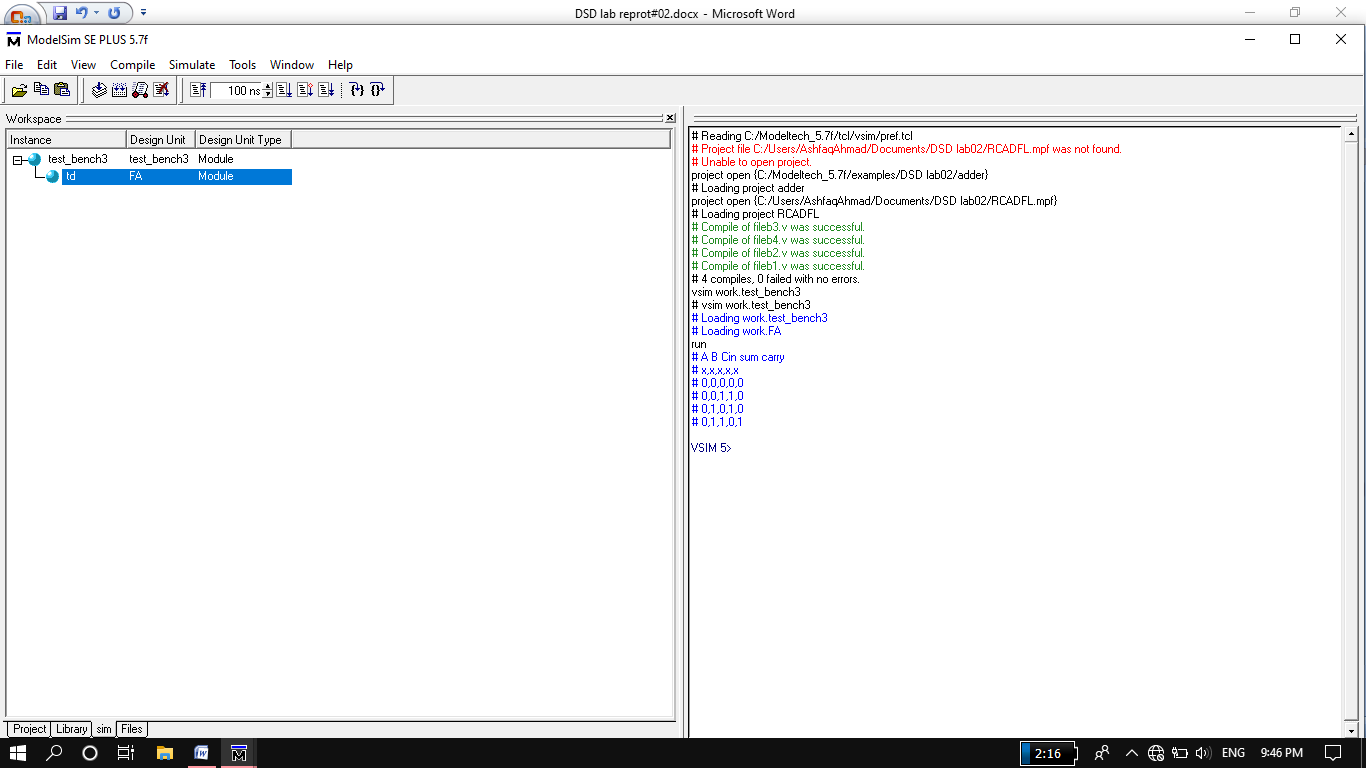
#20 A=1;B=1;Cin=0;

#20 A=1;B=1;Cin=1;

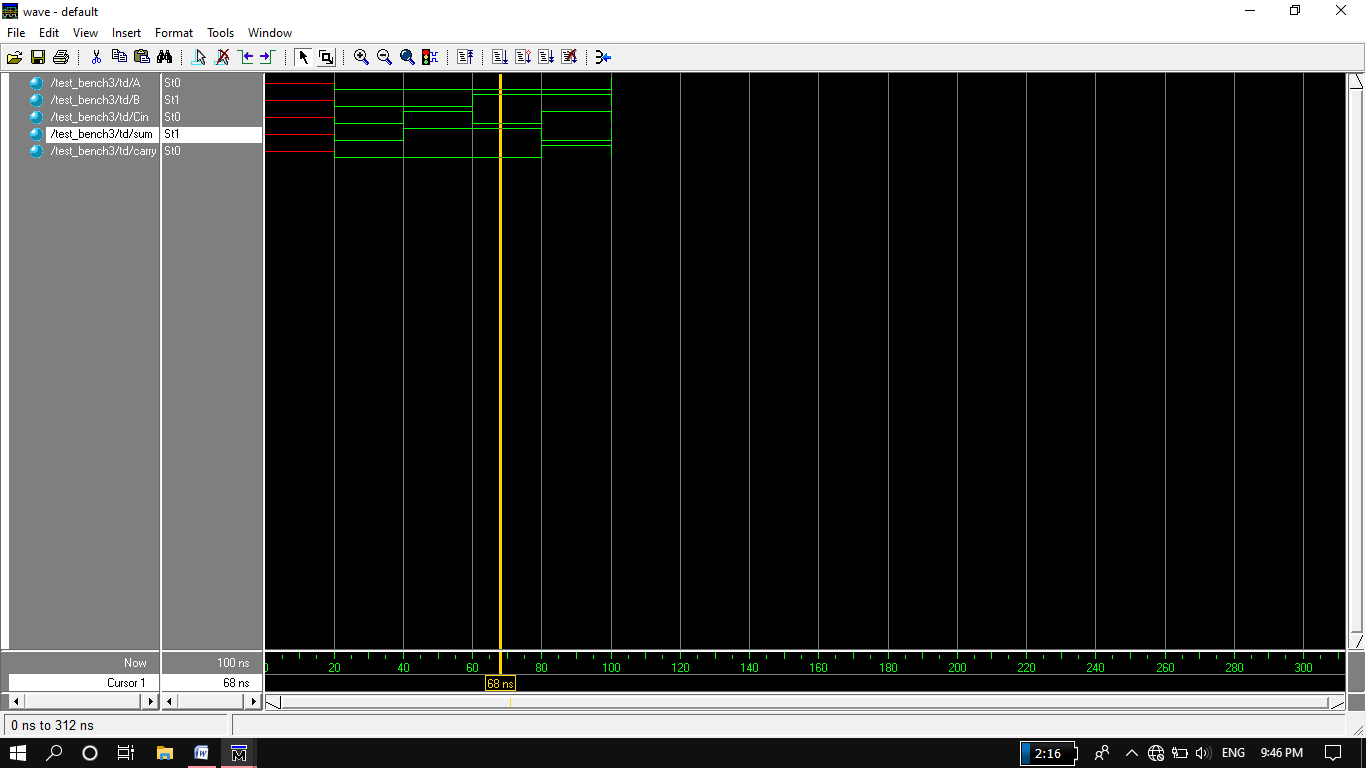
end

endmodule

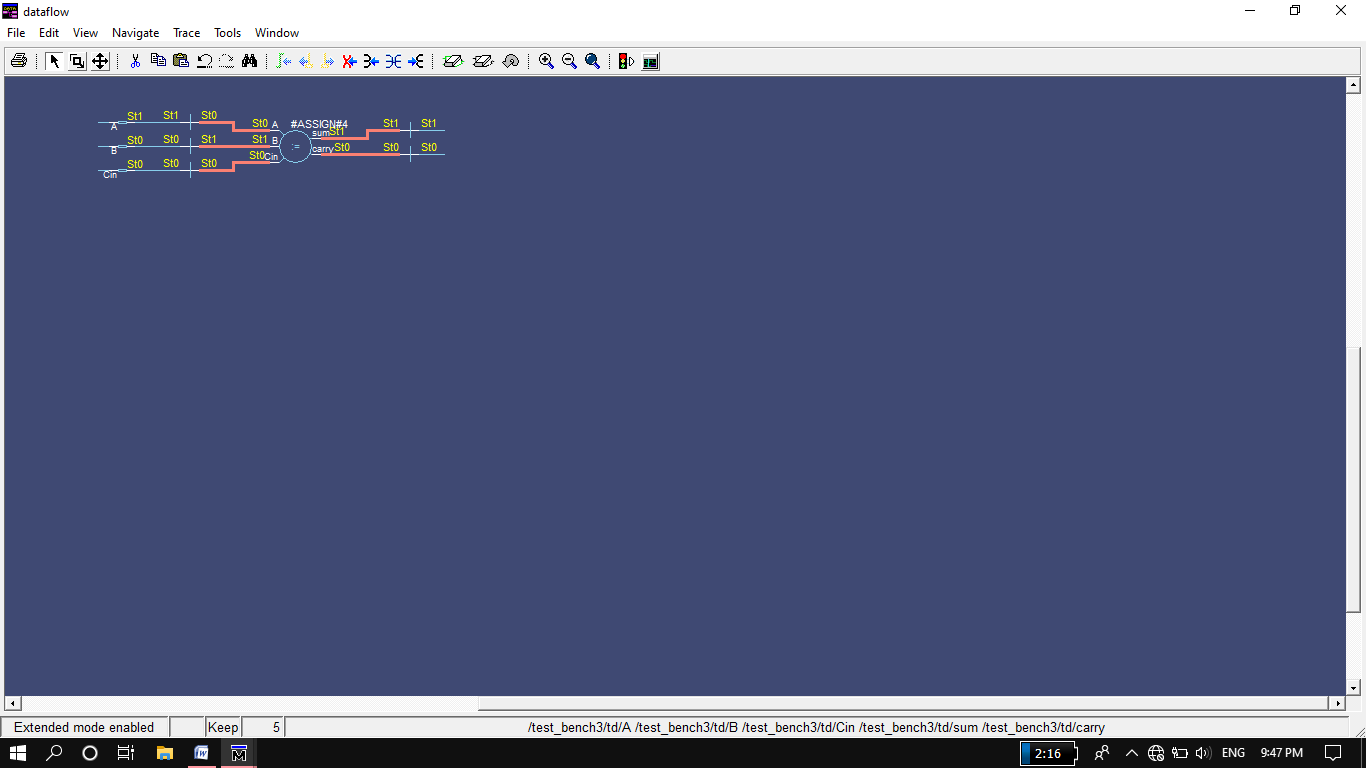
**Truth Table:**



**Wave Form:**



**Circuit:**



**4 bit Ripple Carry Adder:**

**Data Flow Code:**

module RCA(sum,Cout,A,B);

input [3:0] A,B;

output [3:0]sum;

output Cout;

wire [3:1]Cin;

FA f1(sum[0],Cin[1],A[0],B[0],1'b0); //1'b0=cin[0]

FA f2(sum[1],Cin[2],A[1],B[1],Cin[1]);

FA f3(sum[2],Cin[3],A[2],B[2],Cin[2]);

FA f4(sum[3],Cout,A[3],B[3],Cin[3]);

endmodule

**Test Bench:**

module test\_bench4();

reg [3:0] A,B;

wire [3:0]sum;

wire Cout;

RCA td(sum,Cout,A,B);

initial

begin

$display("A B sum Cout");

$monitor("%d,%d,%d,%d",A,B,sum,Cout);

#10 A=4'b0000;B=4'b0000;

#10 A=4'b0001;B=4'b0100;

#10 A=4'b0011;B=4'b1100;

#10 A=4'b1000;B=4'b1100;

#10 A=4'b0011;B=4'b0100;

#10 A=4'b1000;B=4'b1111;

#10 A=4'b0110;B=4'b0001;

#10 A=4'b1110;B=4'b0011;

#10 A=4'b0011;B=4'b0100;

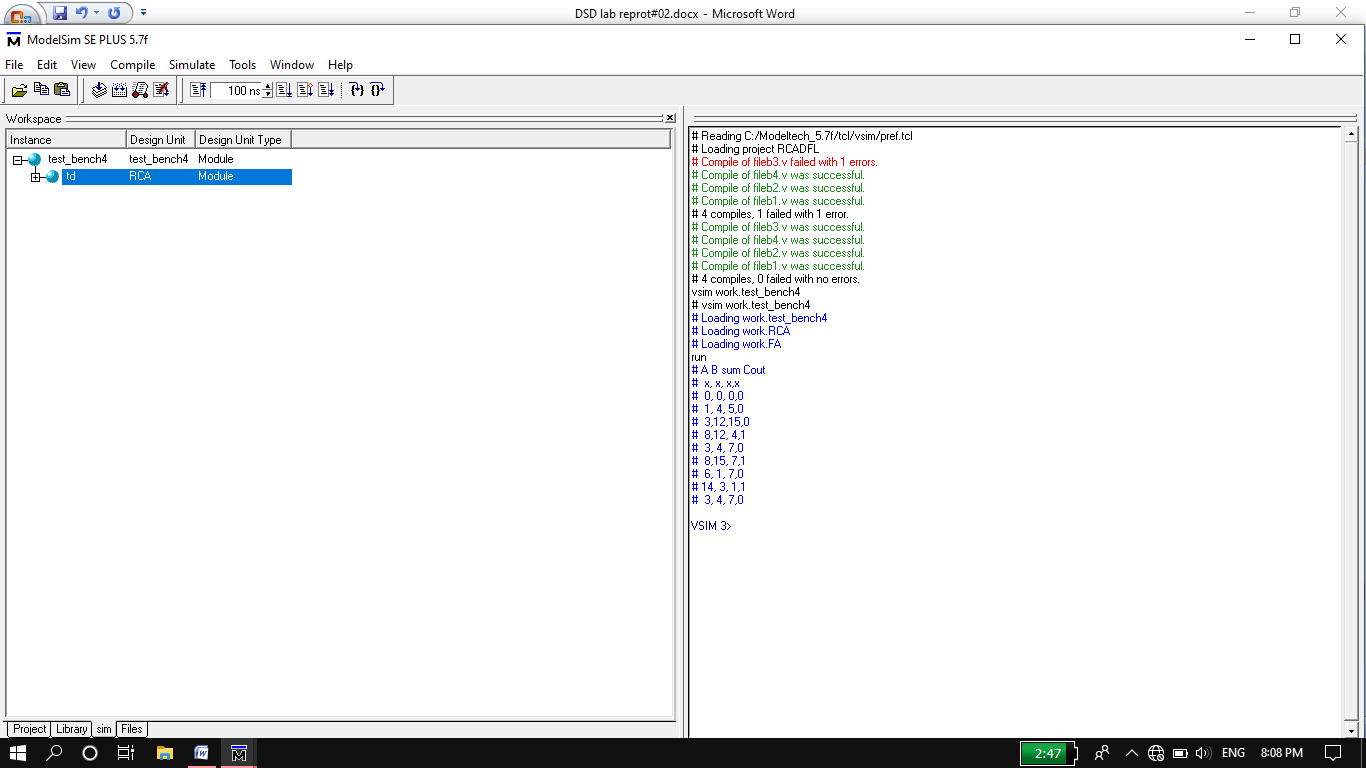
#10 A=4'b0110;B=4'b0010;

#10 A=4'b1111;B=4'b1111;

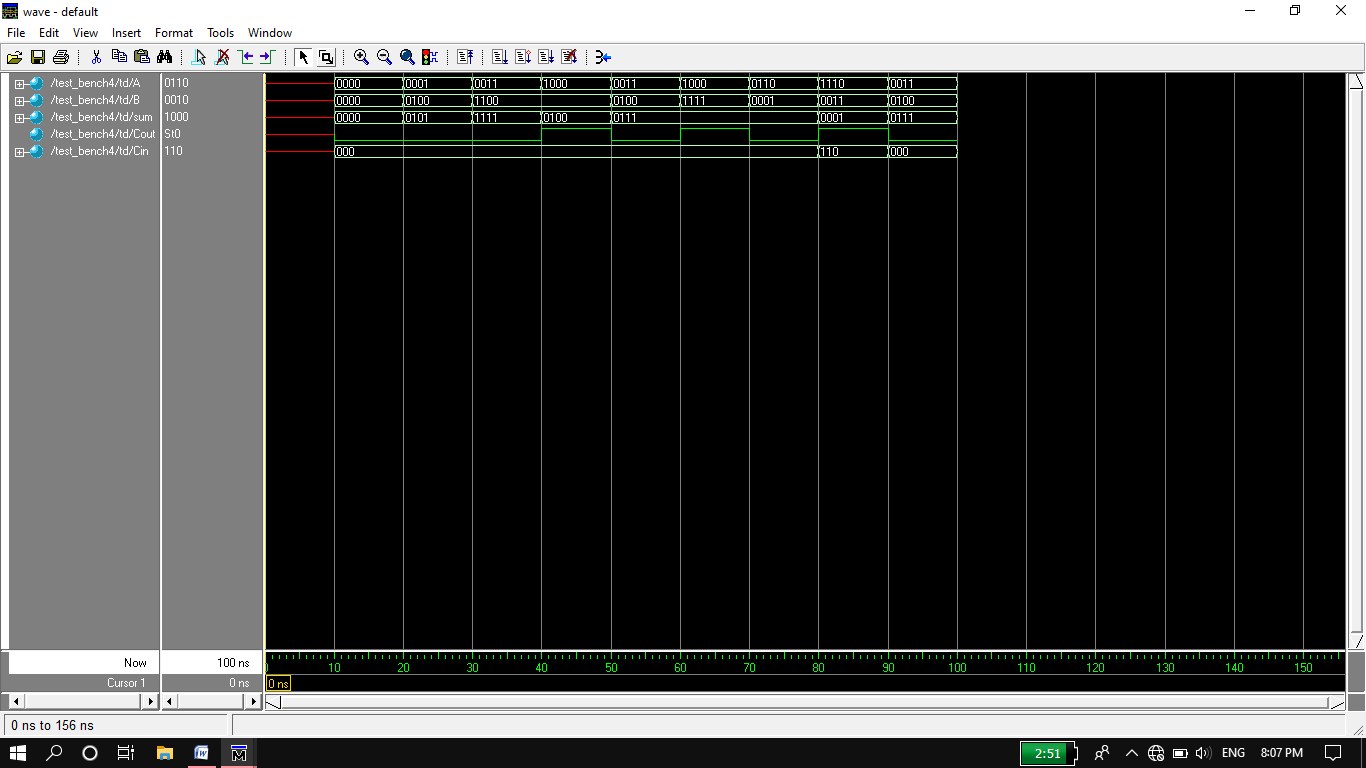
end

endmodule

**Truth Table:**



**Wave Form:**



**Circuit:**

