Lab02

Spring 2025

Submitted by: **Mohsin Sajjad**

Registration No: **22pwsce2149**

Class Section: **A**

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

Mohsin Sajjad

Student Signature: ______________

Submitted to:

Engr. Faheem Jan

Month Day, Year (16 02, 2025)

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

LAB No 2

<u>Objectives:</u>

This lab will enable students to:

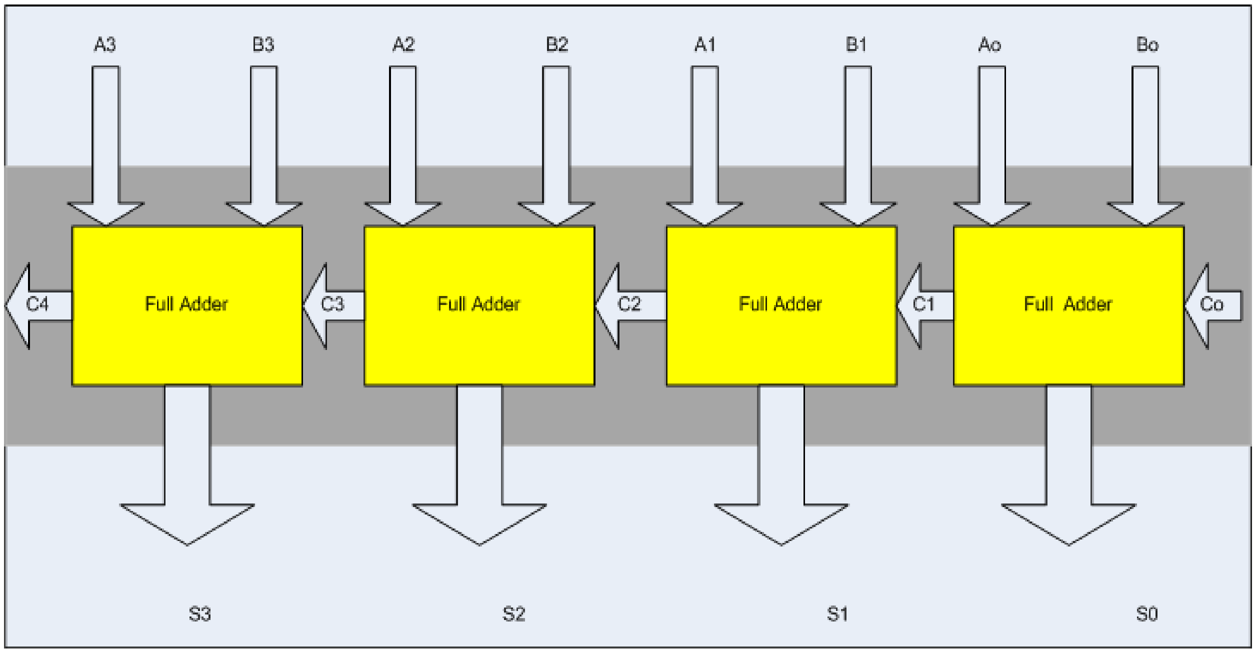
- Learn top down and bottom up design methodologies
- Data flow level modeling

<u>TASK 1:</u>

<u>4 bit Ripple Carry Adder</u>

<u>Block Diagram:</u>

Following is the block diagram of a 4 bit Ripple Carry Adder.



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.

CODE:

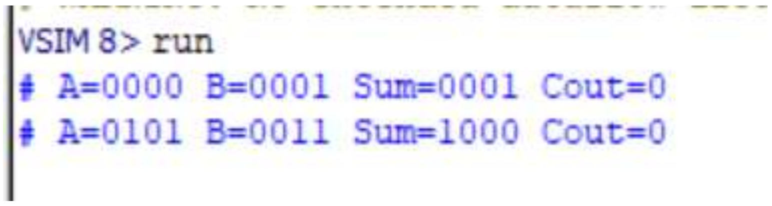
```
module Sum(S,A,B);

   input A,B;

   output S;

   xor x(S,A,B);

endmodule


module Carry(C,A,B);

   input A,B;

   output C;

   and a(C,A,B);

endmodule


module HA(S,C,A,B);

   input A,B;

   output S,C;

   Sum s1(S,A,B);

   Carry c1(C,A,B);

endmodule
```

```
module FA(S,C,A,B,Cin);
    input A,B,Cin;
    output S,C;
    wire S1,C1,C2;

    HA h1(S1,C1,A,B);
    HA h2(S,C2,Cin,S1);
    or o(C,C1,C2);
endmodule

module RCA(S,Cout,A,B,Cin);
    input [3:0] A,B;
    input Cin;
    output [3:0] S;
    output Cout;
    wire [2:0] C;

    FA f1(S[0],C[0],A[0],B[0],Cin);
    FA f2(S[1],C[1],A[1],B[1],C[0]);  // Unique instance name
    FA f3(S[2],C[2],A[2],B[2],C[1]);  // Unique instance name
    FA f4(S[3],Cout,A[3],B[3],C[2]);  // Unique instance name
endmodule

module RCA_tb();
    reg [3:0] A,B;
    reg Cin;
    wire [3:0] S;
    wire Cout;

    RCA rr(S,Cout,A,B,Cin);

    initial begin
```

```
        A = 4'b0000; B = 4'b0001; Cin = 1'b0;
        #20;
        A = 4'b0101; B = 4'b0011; Cin = 1'b0;
    end

    initial
        $monitor("A=%b B=%b Sum=%b Cout=%b", A, B, S, Cout);
endmodule
```

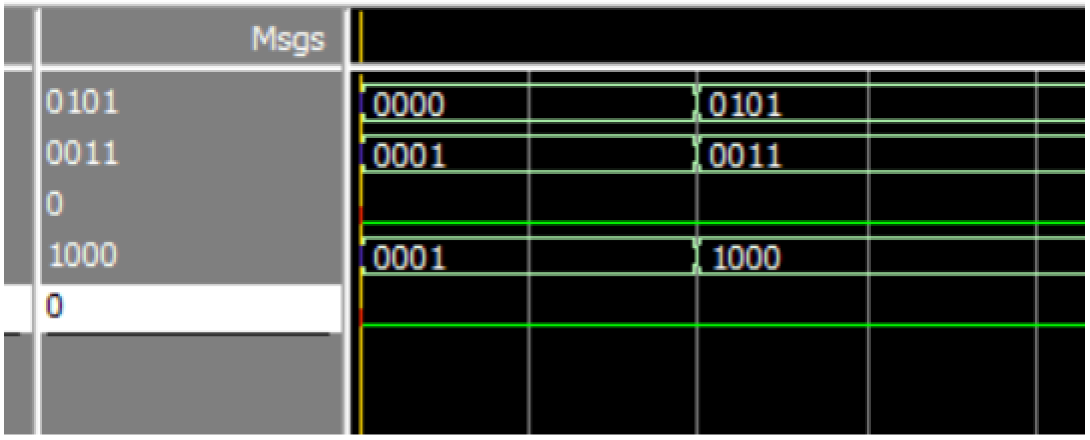
OUTPUT:

Truth Table:



```
VSIM 8> run
# A=0000 B=0001 Sum=0001 Cout=0
# A=0101 B=0011 Sum=1000 Cout=0
```

Wave output:



```
Msgs
0101    0000                0101
0011    0001                0011
0
1000    0001                1000
0
```

Task 1b:

Design the 4 bit full adder using data flow level modeling.

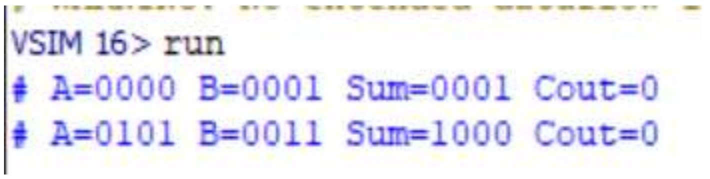
CODE:

```
module HA(S, C, A, B);
   input A, B;
```

```
    output S, C;


    assign S = A ^ B;
    assign C = A & B;
endmodule


module FA(S, C, A, B, Cin);
    input A, B, Cin;
    output S, C;


    assign S = A ^ B ^ Cin;
    assign C = (A & B) | (B & Cin) | (A & Cin);
endmodule


module RCA2(S, Cout, A, B, Cin);
    input [3:0] A, B;
    input Cin;
    output [3:0] S;
    output Cout;
    wire [2:0] C;  // Internal carry wires


    // Ripple Carry Adder using Dataflow
    FA fa0(S[0], C[0], A[0], B[0], Cin);
    FA fa1(S[1], C[1], A[1], B[1], C[0]);

    FA fa2(S[2], C[2], A[2], B[2], C[1]);

    FA fa3(S[3], Cout, A[3], B[3], C[2]);

endmodule


module RCA_tb2();

    reg [3:0] A, B;

    reg Cin;

    wire [3:0] S;

    wire Cout;


    RCA rr(S, Cout, A, B, Cin);


    initial begin

        A = 4'b0000; B = 4'b0001; Cin = 1'b0;

        #20;

        A = 4'b0101; B = 4'b0011; Cin = 1'b0;

    end


    initial

        $monitor("A=%b B=%b Sum=%b Cout=%b", A, B, S, Cout);

endmodule
```

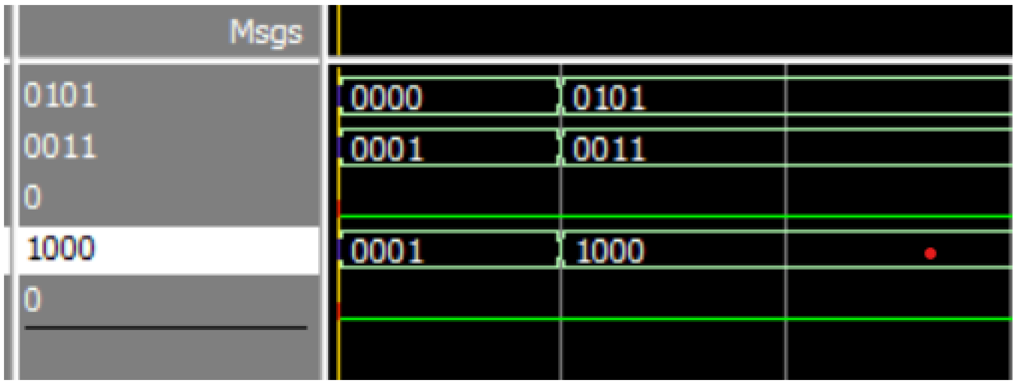
Output:

Truth Table:



```
VSIM 16> run
# A=0000 B=0001 Sum=0001 Cout=0
# A=0101 B=0011 Sum=1000 Cout=0
```

Wave output:



```
Msgs
0101    0000  0101
0011    0001  0011
0
1000    0001  1000
0
```