Structural Design in HDL

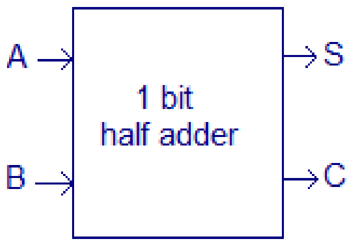
1. Lab Description

Part A: Half Adder

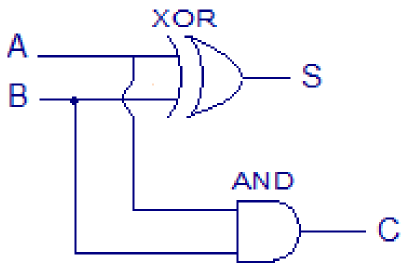
Students are asked to design a 1-bit half adder using HDL. Use the port names based on the picture shown below. Write a testbench code to verify your design.

Inputs		Outputs	
A	B	S	C
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

Truth table



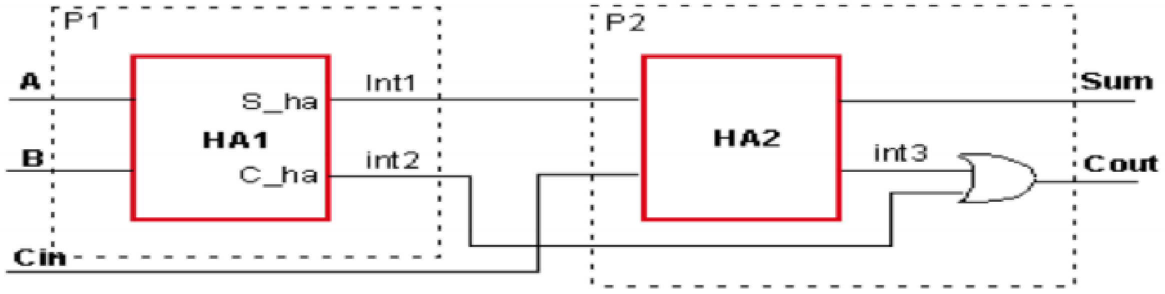
Schematic



Realization

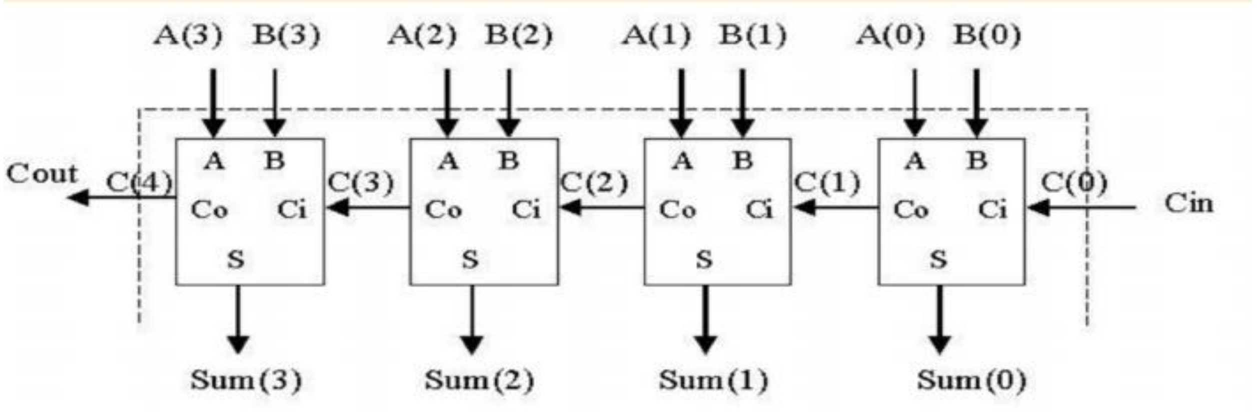
Part B: Full Adder

Using the half adder designed in Part A implement a 1-bit full adder circuit. Verify your design by writing a testbench code



Part C: 4-Bit Ripple Carry Adder

Using the Full Adder designed in Part B create a 4-bit ripple carry adder.



Write your answers from here.

A. Half Adder

Code:

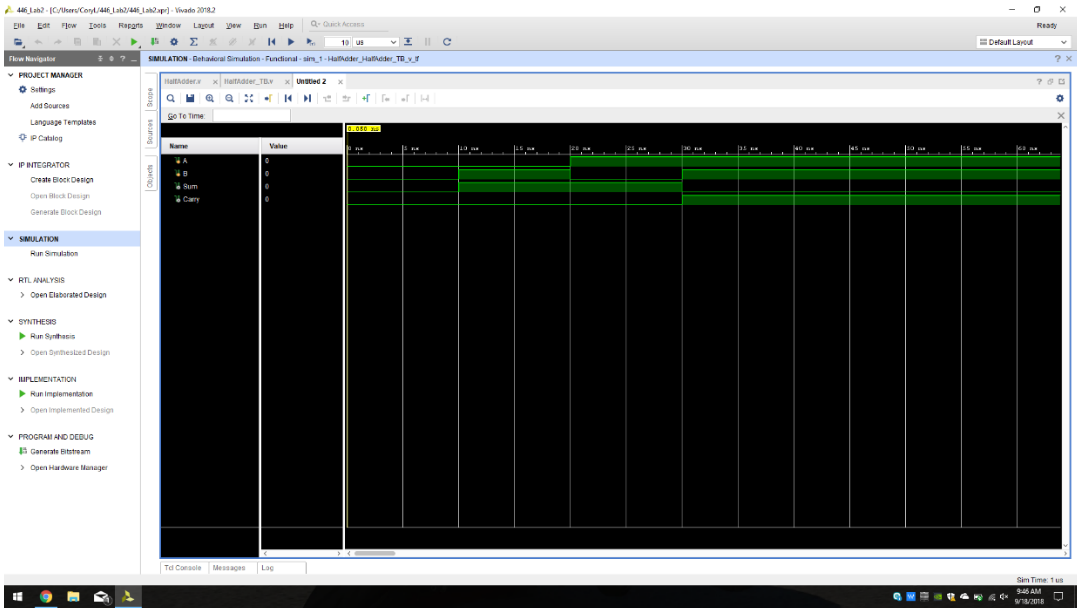
```
module HalfAdder(A, B, Sum, Carry);
   input A,B;
   output Sum,Carry;

   assign Sum = A ^ B;
   assign Carry = A & B;

endmodule
```

Test Bench:

```
module HalfAdder_HalfAdder_TB_v_tf();

// DATE:     21:22:33 09/17/2018
// MODULE:   HalfAdder
// DESIGN:   HalfAdder
// FILENAME: HalfAdder_TB.v
// PROJECT:  EGCP450_Lab2
// VERSION:

// Inputs
   reg A;
   reg B;

// Outputs
   wire Sum;
   wire Carry;
// Bidirs

// Instantiate the UUT
   HalfAdder uut (
      .A(A),
      .B(B),
      .Sum(Sum),
      .Carry(Carry)
      );

// Initialize Inputs
```

```
        initial begin
            A = 0;
            B = 0;
                        #10;
                            A = 0;
            B = 1;
                        #10;
                            A = 1;
            B = 0;
                        #10;
                            A = 1;
            B = 1;
                        #10;
        end
endmodule
```

Wave:


446_Lab2 - [C:/Users/CoryL/446_Lab2/446_Lab2.xpr] - Vivado 2018.2
SIMULATION - Behavioral Simulation - Functional - sim_1 - HalfAdder_HalfAdder_TB_v_tf
Name
Value
A
B
Sum
Carry
0
0
0
0

B. **Full Adder**

Code:

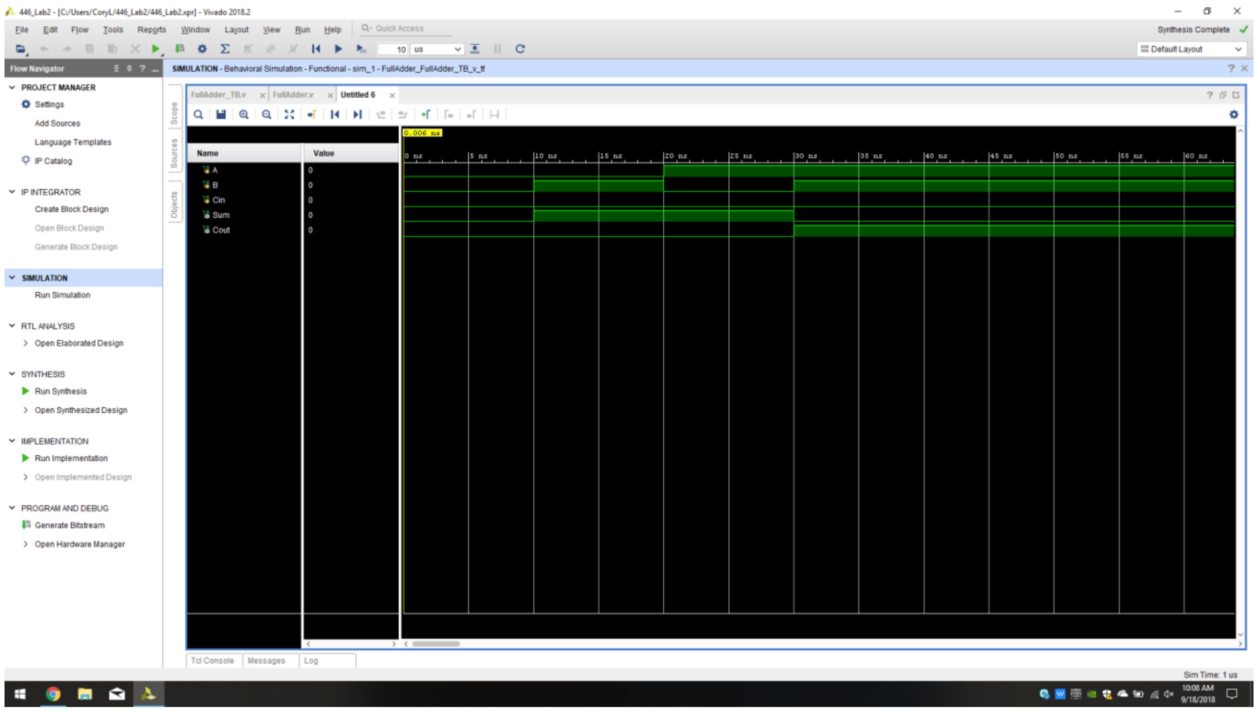
```
module FullAdder(A, B, Cin, Sum, Cout);
   input A, B, Cin;
   output Sum, Cout;

   assign Sum = A ^ B ^ Cin;
   assign Cout = A&Cin | A&B | B&Cin;

endmodule
```

Test Bench:

```
module FullAdder_FullAdder_TB_v_tf();

// DATE:     21:24:44 09/17/2018
// MODULE:   FullAdder
// DESIGN:   FullAdder
// FILENAME: FullAdder_TB.v
// PROJECT:  EGCP450_Lab2
// VERSION:

// Inputs
   reg A;
   reg B;
   reg Cin;

// Outputs
   wire Sum;
   wire Cout;

// Bidirs

// Instantiate the UUT
   FullAdder uut (
      .A(A),
      .B(B),
      .Cin(Cin),
      .Sum(Sum),
      .Cout(Cout)
      );

// Initialize Inputs

   initial begin
         A = 0;
         B = 0;
         Cin = 0;
                    #10;
                       A = 0;
         B = 1;
                    #10;
                       A = 1;
         B = 0;
                    #10;
                       A = 1;
```

```
                B = 1;
                        #10;
        end

endmodule
```

Wave:



C. 4-Bit Ripple Carry Adder

Code:

```
module RippleCarryAdder(A, B, Cin, Sum, Cout);
   input [3:0] A, B;
   input Cin;
   output [3:0] Sum;
   output Cout;

   wire [2:0] C;

   FullAdder
      FA0 (.A(A[0]), .B(B[0]), .Cin(Cin),  .Cout(C[0]), .Sum(Sum[0]) ),
      FA1 (.A(A[1]), .B(B[1]), .Cin(C[0]), .Cout(C[1]), .Sum(Sum[1]) ),
      FA2 (.A(A[2]), .B(B[2]), .Cin(C[1]), .Cout(C[2]), .Sum(Sum[2]) ),
      FA3 (.A(A[3]), .B(B[3]), .Cin(C[2]), .Cout(Cout), .Sum(Sum[3]) );
endmodule
```