California State University, Fullerton

Computer Engineering

**EGCP 446 – Advanced Digital Design using Verilog HDL**

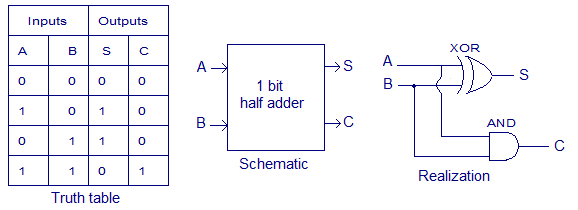
**(Fall 2019)**

**Practice Lab No 1: Structural Design in HDL**

1. **Lab Description**

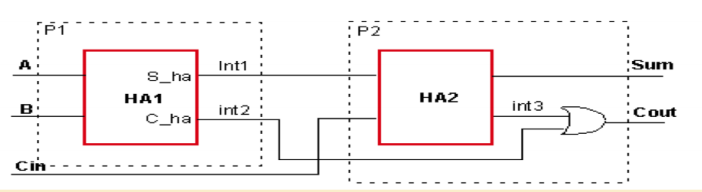
**Part A: Half Adder**

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.



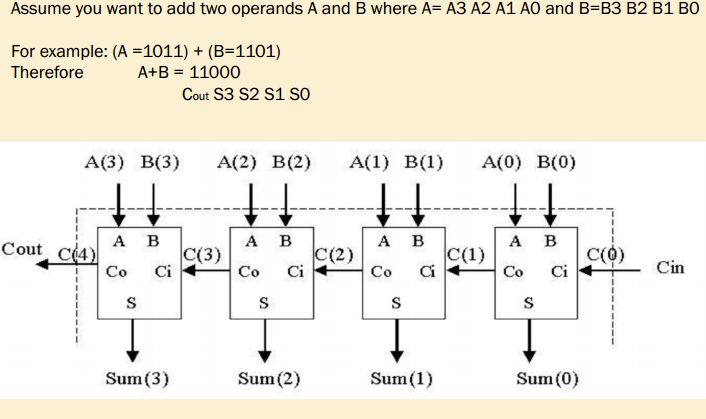
**Part B: Full Adder**

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.

**Brandon Hoang**

**889874731**

Half\_Adder:

module Half\_Adder(a,b,s,c);

input a,b;

output s,c;

xor (s,a,b);

and (c,a,b);

endmodule

Test Bench HA:

module Half\_Adder\_Half\_Adder\_tb\_v\_tf();

reg a;

reg b;

wire s;

wire c;

Half\_Adder uut(

.a(a),

.b(b),

.s(s),

.c(c)

);

initial begin

a = 0;

b = 0;

#100;

a = 0;

b = 1;

#100;

a = 1;

b = 0;

#100;

a = 1;

b = 1;

#100;

end

endmodule

Full\_Adder:

module Full\_Adder(A,B,Cin,Sum,Cout);

input A,B,Cin;

output Sum,Cout;

wire Int1, Int2, Int3;

Half\_Adder HA1(A,B, Int1, Int2) ;

Half\_Adder HA2 (Int1, Cin, Sum, Int3 );

or (Cout, Int3, Int2 );

endmodule

Test Bench FA:

module Full\_Adder\_Full\_Adder\_tb\_v\_tf();

reg A;

reg B;

reg Cin;

wire Sum;

wire Cout;

Full\_Adder uut (

.A(A),

.B(B),

.Cin(Cin),

.Sum(Sum),

.Cout(Cout)

);

initial begin

A = 0;

B = 0;

Cin = 0;

#100;

A = 0;

B = 0;

Cin = 1;

#100;

A = 0;

B = 1;

Cin = 0;

#100;

A = 0;

B = 1;

Cin = 1;

#100;

A = 1;

B = 0;

Cin = 0;

#100;

A = 1;

B = 0;

Cin = 1;

#100;

A = 1;

B = 1;

Cin = 0;

#100;

A = 1;

B = 1;

Cin = 1;

end

endmodule

RCA:

module Four\_RCA(A,B,Cin,Sum,Cout);

input [3:0] A;

input [3:0] B;

input Cin;

output [3:0] Sum;

output Cout;

wire C1,C2,C3;

Full\_Adder FA1( A[0], B[0], Cin, Sum[0], C1);

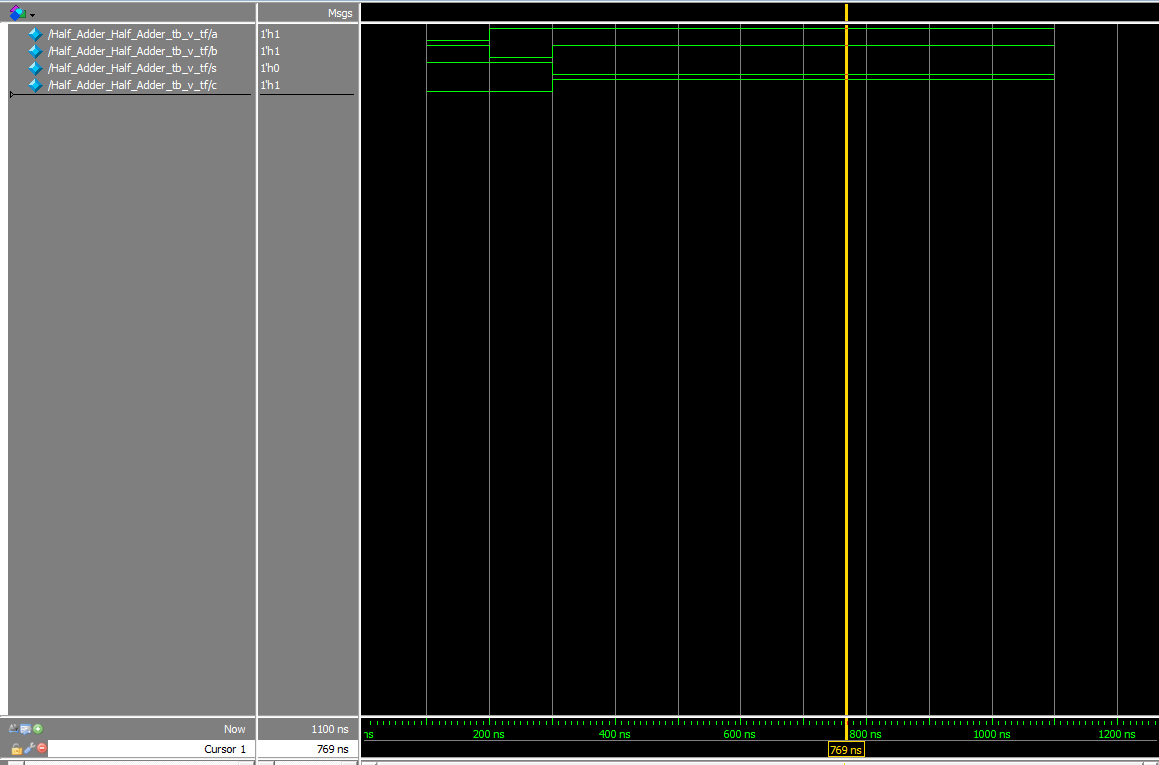
Full\_Adder FA2( A[1], B[1], C1, Sum[1], C2);

Full\_Adder FA3( A[2], B[2], C2, Sum[2], C3);

Full\_Adder FA4( A[3], B[3], C3, Sum[3], Cout);

Endmodule

Half Adder Simulation



Full Adder Simulation

