**Project 2**

**Name:**

Tyler Holland

**Class:**

CPE129 -05

**Full Adder**

Library IEEE;

Use IEEE.STD\_LOGIC\_1164.all;

Entity FullAdder is

Port(CI, A, B : in STD\_LOGIC;

S, CO : out STD\_LOGIC);

End FullAdder;

Architecture adder of FullAdder is

Begin

S <= (CI xor A) xor B;

CO <= (CI and A) or (CI and B) or (A and B);

End adder;

**2’s Complement Adder/Subtractor**

Library IEEE;

Use IEEE\_STD\_LOGIC\_1164.all;

Entity 2compAddSub is

Port(CarryIn, Subtract, AllowCarry : in STD\_LOGIC;

A, B : in STD\_LOGIC\_VECTOR(3 downto 0);

CarryOut, Overflow, ResultZero : out STD\_LOGIC;

Fout : out STD\_LOGIC\_VECTOR(3 downto 0));

End 2compAddSub;

Architecture structural of 2compAddSub is

Component FullAdder is

Port(CI, A, B : in STD\_LOGIC;

S, CO : out STD\_LOGIC);

End component;

Signal CO1, CO2, CO3, CO4, C1, C2, C3, Bsub0, Bsub1, Bsub2, Bsub3 : STD\_LOGIC;

Signal out0, out1, out2, out3, over : STD\_LOGIC;

Begin

--Carry out signals with allow carry functionality

C1 <= AllowCarry AND CO1;

C2 <= AllowCarry AND CO2;

C3 <= AllowCarry AND CO3;

C4 <= AllowCarry AND CO4;

--Allows use of the subtract signal

Bsub0 <= B(0) XOR Subtract;

Bsub1 <= B(1) XOR Subtract;

Bsub2 <= B(2) XOR Subtract;

Bsub3 <= B(3) XOR Subtract;

Fout <= (out0, out1, out2, out3);

over <= CO4 XOR CO3;

CarryOut <= CO4 AND over;

Overflow <= over;

ResultZero <= out0 NOR out1 NOR out2 NOR out3;

**2’s Complement Adder/Subtractor**

Add1: FullAdder

Port map (CarryIn, A(0), Bsub0, out0, CO1);

Add2: FullAdder

Port map (C1, A(1), Bsub1, out1, CO2);

Add3: FullAdder

Port map (C2, A(1), Bsub1, out2, CO3);

Add4: FullAdder

Port map (C3, A(1), Bsub1, out3, CO4);

End structural;

**16-bit Arithmetic Logic Unit**

Library IEEE;

Use IEEE\_STD\_LOGIC\_1164.all;

Entity ALU16 is

Port( A, B : in STD\_LOGIC\_VECTOR(15 downto 0);

M, Sel : in STD\_LOGIC;

ResultZero, SB : out STD\_LOGIC;

F : out STD\_LOGIC\_VECTOR (15 downto 0));

End ALU16;

Architecture structural of ALU16 is

Component 2compAddSub is

Port(CarryIn, Subtract, AllowCarry : in STD\_LOGIC;

A, B : in STD\_LOGIC\_VECTOR(3 downto 0);

CarryOut, Overflow, ResultZero : out STD\_LOGIC;

Fout : out STD\_LOGIC\_VECTOR(3 downto 0));

End component;

Signal C1, C2, C3, C4 : STD\_LOGIC;

Signal zero1, zero2, zero3, zero4 : STD\_LOGIC;

Begin

SB <= C4;

ResultZero <= zero1 AND zero2 AND zero3 AND zero4;

Comp1: 2compAddSub

Port map (‘0’, Sel, M, A(3 downto 0), B(3 downto 0), C1, open, zero1, F(3 downto 0));

Comp2: 2compAddSub

Port map (C1, Sel, M, A(7 downto 4), B(7 downto 4), C2, open, zero2, F(7 downto 4));

Comp2: 2compAddSub

Port map (C2, Sel, M, A(11 downto 8), B(11 downto 8), C3, open, zero3, F(11 downto 8));

Comp2: 2compAddSub

Port map (C3, Sel, M, A(15 downto 12), B(15 downto 12), C4, open, zero4,

F(15 downto 12));

End structural;