Jae Lee

ECEN 220

Lab #5 – Arithmetic Logic Unit (ALU)

10/6/2013

**Full Adder Verilog code**

module FullAdder(

input A,

input B,

input Cin,

output Cout,

output Sum

);

wire temp1,temp2,temp3;

xor(Sum,A,B,Cin);

and(temp1,A,B);

and(temp2,B,Cin);

and(temp3,A,Cin);

or(Cout,temp1,temp2,temp3);

endmodule

**4:1 Mux Verilog code**

module Mux41(

output Q,

input A,

input B,

input C,

input D,

input Sel0,

input Sel1

);

wire sel0bar,sel1bar,a1,b1,c1,d1;

not(sel0bar,Sel0);

not(sel1bar,Sel1);

and(a1,A,sel0bar,sel1bar);

and(b1,B,Sel0,sel1bar);

and(c1,C,sel0bar,Sel1);

and(d1,D,Sel0,Sel1);

or(Q,a1,b1,c1,d1);

endmodule

**Full Adder TCL file**

wave add / -radix hex

isim force add A 0 -time 0 -value 1 -time 10ns -repeat 20ns

isim force add B 0 -time 0 -value 1 -time 20ns -repeat 40ns

isim force add Cin 0 -time 0 -value 1 -time 40ns -repeat 80ns

run 80ns

**4:1 MUX TCL file**

wave add / -radix hex

isim force add A 0

isim force add B 1

isim force add C 0

isim force add D 1

isim force add Sel0 0 -time 0 -value 1 -time 20ns -repeat 40ns

isim force add Sel1 0 -time 0 -value 1 -time 40ns -repeat 80ns

run 80ns

isim force add A 1

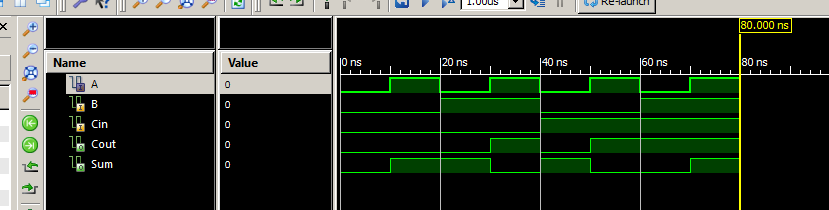
isim force add B 0

isim force add C 1

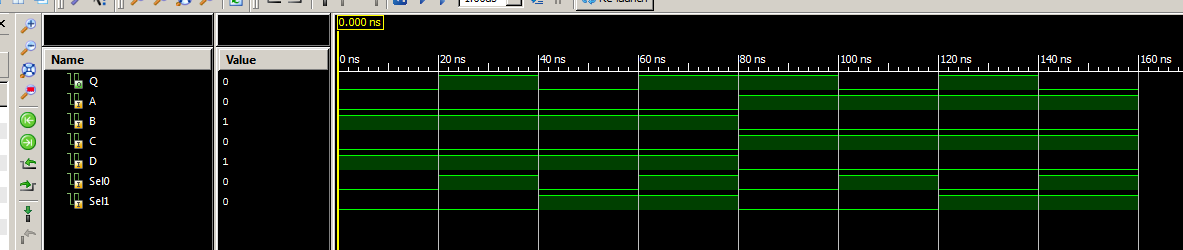
isim force add D 0

run 80ns

**Full Adder Simulation waveform**



**4:1 MUX Simulation waveform**



**1 bit ALU Verilog Code**

module ALU(

input C0,

input C1,

input Cin,

input A,

input B,

output Cout,

output Result

);

wire tempCout,tempSum,c1,d1;

FullAdder F1(A,B,Cin,tempCout,tempSum);

Mux41 M1(Result,A,tempSum,c1,d1,C0,C1);

CarryControl CC1(C0,C1,tempCout,Cout);

and(c1,A,B);

not(d1,A);

endmodule

module CarryControl(

input C0,

input C1,

input Cin,

output Cout

);

wire temp1,temp2;

not(temp1,C1);

and(temp2,temp1,C0);

and(Cout,temp2,Cin);

endmodule

**1 bit ALU TCL file**

wave add / -radix hex

isim force add A 0

isim force add B 0

isim force add Cin 0

isim force add C0 0 -time 0 -value 1 -time 20ns -repeat 40ns

isim force add C1 0 -time 0 -value 1 -time 40ns -repeat 80ns

run 80ns

isim force add A 1

isim force add B 0

isim force add Cin 0

run 80ns

isim force add A 0

isim force add B 1

isim force add Cin 0

run 80ns

isim force add A 1

isim force add B 1

isim force add Cin 0

run 80ns

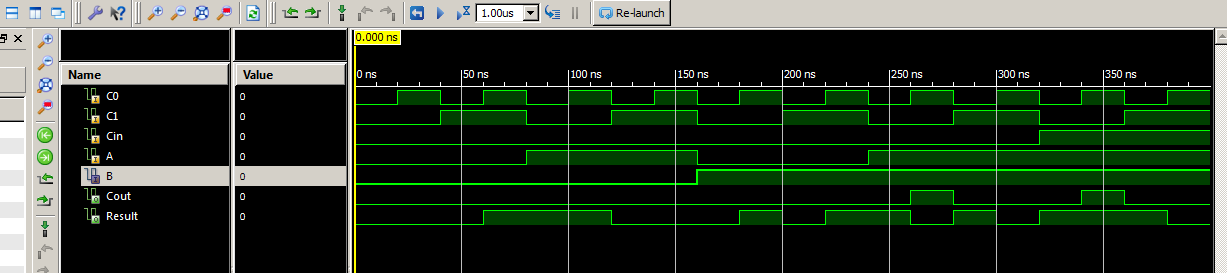
isim force add A 1

isim force add B 1

isim force add Cin 1

run 80ns

**1 bit ALU Simulation waveform**



**4 bit ALU Verilog code**

module ALU4(

input [1:0] C0,

input Cin,

input [3:0] A,

input [3:0] B,

output Cout,

output [3:0] Result

);

wire [2:0] tempCout;

ALU a1(C0[0],C0[1],Cin,A[0],B[0],tempCout[0],Result[0]);

ALU a2(C0[0],C0[1],tempCout[0],A[1],B[1],tempCout[1],Result[1]);

ALU a3(C0[0],C0[1],tempCout[1],A[2],B[2],tempCout[2],Result[2]);

ALU a4(C0[0],C0[1],tempCout[2],A[3],B[3],Cout,Result[3]);

endmodule

**4 bit ALU TCL file**

wave add / -radix hex

isim force add A 1010

isim force add B 0001

isim force add Cin 0

isim force add C0 00 -time 0ns -value 01 -time 20ns -value 10 -time 40ns -value 11 -time 60ns -repeat 80ns

run 80ns

isim force add A 1111

isim force add B 1010

isim force add Cin 0

run 80ns

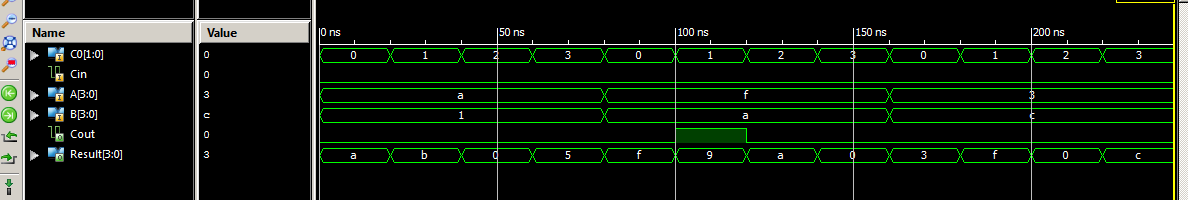
isim force add A 0011

isim force add B 1100

isim force add Cin 0

run 80ns

**4 bit ALU Simulation waveform**



**4 bit ALU UCF file**

## Leds

NET Result<0> LOC = "J14"; # Bank = 1, Pin name = IO\_L14N\_1/A3/RHCLK7, Type = RHCLK/DUAL, Sch name = JD10/LD0

NET Result<1> LOC = "J15"; # Bank = 1, Pin name = IO\_L14P\_1/A4/RHCLK6, Type = RHCLK/DUAL, Sch name = JD9/LD1

NET Result<2> LOC = "K15"; # Bank = 1, Pin name = IO\_L12P\_1/A8/RHCLK2, Type = RHCLK/DUAL, Sch name = JD8/LD2

NET Result<3> LOC = "K14"; # Bank = 1, Pin name = IO\_L12N\_1/A7/RHCLK3/TRDY1, Type = RHCLK/DUAL, Sch name = JD7/LD3

#NET "Led<4>" LOC = "E17"; # Bank = 1, Pin name = IO, Type = I/O, Sch name = LD4

#NET "Led<5>" LOC = "P15"; # Bank = 1, Pin name = IO, Type = I/O, Sch name = LD5

#NET "Led<6>" LOC = "F4"; # Bank = 3, Pin name = IO, Type = I/O, Sch name = LD6

NET Cout LOC = "R4"; # Bank = 3, Pin name = IO/VREF\_3, Type = VREF, Sch name = LD7

## Switches

NET B<0> LOC = "G18"; # Bank = 1, Pin name = IP, Type = INPUT, Sch name = SW0

NET B<1> LOC = "H18"; # Bank = 1, Pin name = IP/VREF\_1, Type = VREF, Sch name = SW1

NET B<2> LOC = "K18"; # Bank = 1, Pin name = IP, Type = INPUT, Sch name = SW2

NET B<3> LOC = "K17"; # Bank = 1, Pin name = IP, Type = INPUT, Sch name = SW3

NET A<0> LOC = "L14"; # Bank = 1, Pin name = IP, Type = INPUT, Sch name = SW4

NET A<1> LOC = "L13"; # Bank = 1, Pin name = IP, Type = INPUT, Sch name = SW5

NET A<2> LOC = "N17"; # Bank = 1, Pin name = IP, Type = INPUT, Sch name = SW6

NET A<3> LOC = "R17"; # Bank = 1, Pin name = IP, Type = INPUT, Sch name = SW7

## Buttons

#NET "btn<0>" LOC = "B18"; # Bank = 1, Pin name = IP, Type = INPUT, Sch name = BTN0

#NET "btn<1>" LOC = "D18"; # Bank = 1, Pin name = IP/VREF\_1, Type = VREF, Sch name = BTN1

NET C0<0> LOC = "E18"; # Bank = 1, Pin name = IP, Type = INPUT, Sch name = BTN2

NET C0<1> LOC = "H13"; # Bank = 1, Pin name = IP, Type = INPUT, Sch name = BTN3

**Anomalies**

It wasn’t extremely difficult. I just had a trouble with setting a variable in array-form. It took a long time to finish, but I enjoyed.