Taylor Cowley

Lab 05: Arithmetic Logic Unit

May 17 2016

**Full Adder Verilog and .tcl files**

module full\_adder(sum, cout, a, b, cin);

// Ports

input a;

input b;

input cin;

output cout;

output sum;

wire a\_b, b\_cin, a\_cin;

and(a\_b, a, b);

and(b\_cin, b, cin);

and(a\_cin, a, cin);

or(cout, a\_b, b\_cin, a\_cin);

xor(sum, a, b, cin);

endmodule

isim force add a 0

isim force add b 0

isim force add cin 0

run 5ns

isim force add cin 1

run 5ns

isim force add cin 0

isim force add b 1

run 5 ns

isim force add cin 1

run 5 ns

isim force add a 1

isim force add b 0

isim force add cin 0

run 5 ns

isim force add cin 1

run 5 ns

isim force add cin 0

isim force add b 1

run 5 ns

isim force add cin 1

run 5 ns

**4:1 Mux Verilog and .tcl files**

module mux\_2(q, sel, a, b);

input a;

input b;

input sel;

output q;

wire a1, a2, selbar;

not(selbar, sel);

and(a1, selbar, a);

and(a2, sel, b);

or(q, a1, a2);

endmodule

module mux\_4(q, sel[1:0], a, b, c, d);

input a;

input b;

input c;

input d;

input[1:0] sel;

output q;

wire a\_or\_b, c\_or\_d;

mux\_2 M0(a\_or\_b, sel[0], a, b); // The LSB selects between a or b

mux\_2 M1(c\_or\_d, sel[0], c, d); // and c or d

mux\_2 M2(q, sel[1], a\_or\_b, c\_or\_d);// the MSB selects between those 2

endmodule

**.tcl file**

isim force add sel 00

isim force add a 0

isim force add b 1

isim force add c 1

isim force add d 1

run 5 ns

isim force add sel 01

run 5 ns

isim force add sel 10

run 5 ns

isim force add sel 11

run 5 ns

isim force add sel 00

isim force add a 1

isim force add b 0

isim force add c 1

isim force add d 1

run 5 ns

isim force add sel 01

run 5 ns

isim force add sel 10

run 5 ns

isim force add sel 11

run 5 ns

isim force add sel 00

isim force add a 1

isim force add b 1

isim force add c 0

isim force add d 1

run 5 ns

isim force add sel 01

run 5 ns

isim force add sel 10

run 5 ns

isim force add sel 11

run 5 ns

isim force add sel 00

isim force add a 1

isim force add b 1

isim force add c 1

isim force add d 0

run 5 ns

isim force add sel 01

run 5 ns

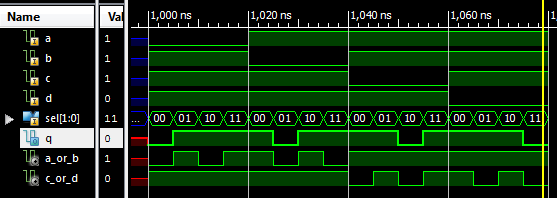
isim force add sel 10

run 5 ns

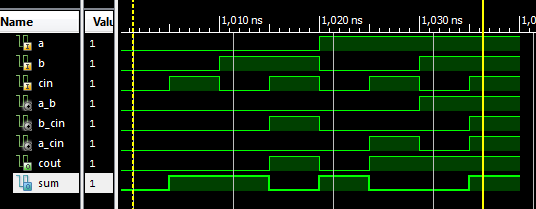
isim force add sel 11

run 5 ns

**4:1 Mux simulation waveform**



**Full Adder simulation waveform**



**1 bit ALU Verilog code and .tcl file**

module alu(out, cout, c[1:0], a, b, cin);

// Ports

input a;

input b;

input cin;

input[1:0] c;

output cout;

output out;

wire a\_b, a\_not, sum;

and(a\_b, a, b);

not(a\_not, a);

full\_adder add(sum, cout, a, b, cin);

mux\_4 select(out, c[1:0], a, sum, a\_b, a\_not);

endmodule

isim force add sel 00

isim force add a 0

isim force add b 1

isim force add c 1

isim force add d 1

run 5 ns

isim force add sel 01

run 5 ns

isim force add sel 10

run 5 ns

isim force add sel 11

run 5 ns

isim force add sel 00

isim force add a 1

isim force add b 0

isim force add c 1

isim force add d 1

run 5 ns

isim force add sel 01

run 5 ns

isim force add sel 10

run 5 ns

isim force add sel 11

run 5 ns

isim force add sel 00

isim force add a 1

isim force add b 1

isim force add c 0

isim force add d 1

run 5 ns

isim force add sel 01

run 5 ns

isim force add sel 10

run 5 ns

isim force add sel 11

run 5 ns

isim force add sel 00

isim force add a 1

isim force add b 1

isim force add c 1

isim force add d 0

run 5 ns

isim force add sel 01

run 5 ns

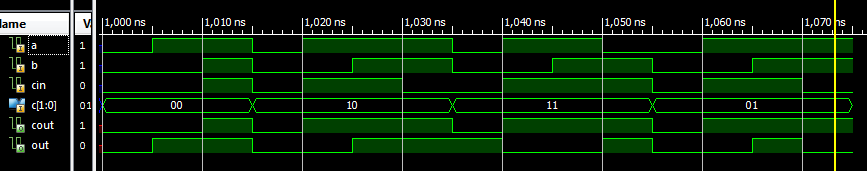
isim force add sel 10

run 5 ns

isim force add sel 11

run 5 ns

**1 Bit ALU simulation waveform**



**4 bit ALU Verilog code and .tcl file**

module alu\_4(out[3:0], cout, c[1:0], a[3:0], b[3:0], cin);

// Ports

input[3:0] a;

input[3:0] b;

input cin;

input[1:0] c;

output cout;

output[3:0] out;

wire carry0, carry1, carry2;

alu alu0(out[0], carry0, c[1:0], a[0], b[0], cin);

alu alu1(out[1], carry1, c[1:0], a[1], b[1], carry0);

alu alu2(out[2], carry2, c[1:0], a[2], b[2], carry1);

alu alu3(out[3], cout, c[1:0], a[3], b[3], carry2);

endmodule

isim force add a 0000 -time 0 -value 0001 -time 5 ns -value 0010 -time 10 ns -value 0011 -time 15 ns -value 0100 -time 20 ns -value 1000 -time 25 ns -value 1100 -time 30 ns -value 1111 -time 35 ns -repeat 50 ns

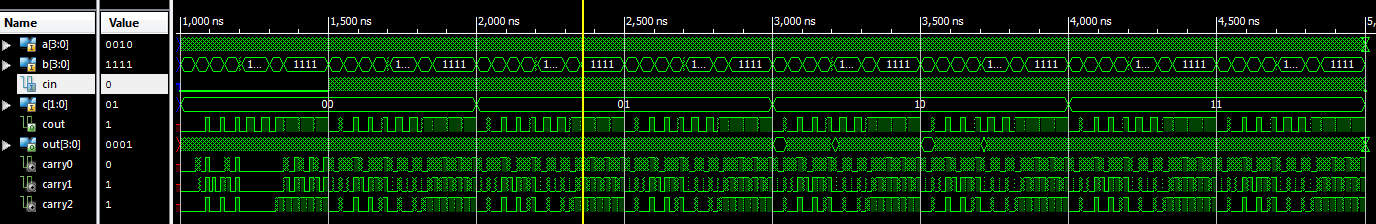
isim force add b 0000 -time 0 -value 0001 -time 50 ns -value 0010 -time 100 ns -value 0011 -time 150 ns -value 0100 -time 200 ns -value 1000 -time 205 ns -value 1100 -time 300 ns -value 1111 -time 350 ns -repeat 500 ns

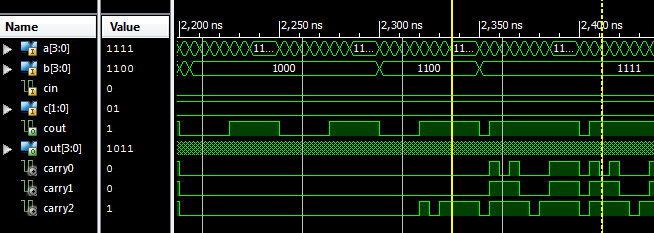
isim force add cin 0 -time 0 -value 1 -time 500 ns -repeat 1000

isim force add c 00 -time 0 -value 01 -time 1000 ns -value 10 -time 2000 ns -value 11 -time 3000 ns

run 4000 ns

**4 bit ALU simulation waveform- both entire thing and more useful snapshot**





**4 bit ALU UCF file**

## Leds

NET "out<0>" LOC = "J14

NET "out<1>" LOC = "J15";

NET "out<2>" LOC = "K15";

NET "out<3>" LOC = "K14";

#NET "Led<4>" LOC = "E17";

#NET "Led<5>" LOC = "P15";

#NET "Led<6>" LOC = "F4";

NET "cout" LOC = "R4";

## Switches

NET "b<0>" LOC = "G18";

NET "b<1>" LOC = "H18";

NET "b<2>" LOC = "K18";

NET "b<3>" LOC = "K17";

NET "a<0>" LOC = "L14";

NET "a<1>" LOC = "L13";

NET "a<2>" LOC = "N17";

NET "a<3>" LOC = "R17";

## Buttons

NET "cin" LOC = "B18";

NET "c<0>" LOC = "E18";

NET "c<1>" LOC = "H13";

**Anomalies (bugs, problems, and suggestions)**

I had major issues with the .tcl file for the 4-bit ALU. I wanted to test most of the possibilities, but I could not hard code everything like I had been doing. I forgot to but the dash in front of value, I had the timing all wrong, but I eventually was able to work it out nicely. And it tested perfectly. ☺