1. **FullAdder.v**

module FullAdder( a, b, cin, cout, sum );

input a, b, cin ;

output sum, cout ;

wire w1, w2, w3 ;

xor xor1( w1, a, b ) ;

xor xor2( sum, w1, cin ) ;

and and1( w2, w1, cin ) ;

and and2( w3, a, b ) ;

or or1( cout, w2, w3 ) ;

endmodule

1. **AddSub4.v**

module AddSub4( a, b, cin, cout, sum, sel );

input [3:0] a, b ;

input cin, sel ;

output [3:0] sum ;

output [3:0] cout ;

wire w0, w1, w2, w3 ;

xor xor1( w0, b[0], sel ) ;

xor xor2( w1, b[1], sel ) ;

xor xor3( w2, b[2], sel ) ;

xor xor4( w3, b[3], sel ) ;

FullAdder FA1( .a(a[0]), .b(w0), .cin(sel), .cout(cout[0]), .sum(sum[0]) ) ;

FullAdder FA2( .a(a[1]), .b(w1), .cin(cout[0]), .cout(cout[1]), .sum(sum[1]) ) ;

FullAdder FA3( .a(a[2]), .b(w2), .cin(cout[1]), .cout(cout[2]), .sum(sum[2]) ) ;

FullAdder FA4( .a(a[3]), .b(w3), .cin(cout[2]), .cout(cout[3]), .sum(sum[3]) ) ;

endmodule

1. **TM\_AddSub4.v**

module TM\_AddSub4() ;

reg [3:0] a, b ;

reg cin, sel ;

wire [3:0] sum ;

wire [3:0] cout ;

AddSub4 AddSub4( .a(a), .b(b), .cin(cin), .cout(cout), .sum(sum), .sel(sel) ) ;

initial begin

a = 4'd5 ;

b = 4'd1 ;

cin = 1'd0 ;

sel = 0 ;

#10

sel = 1 ;

#10

a = 4'd0 ;

b = 4'd0 ;

sel = 0 ;

#10

sel = 1 ;

#10

a = 4'd2 ;

b = 4'd2 ;

sel = 0 ;

#10

sel = 1 ;

#10

a = 4'd1 ;

b = 4'd5 ;

sel = 0 ;

#10

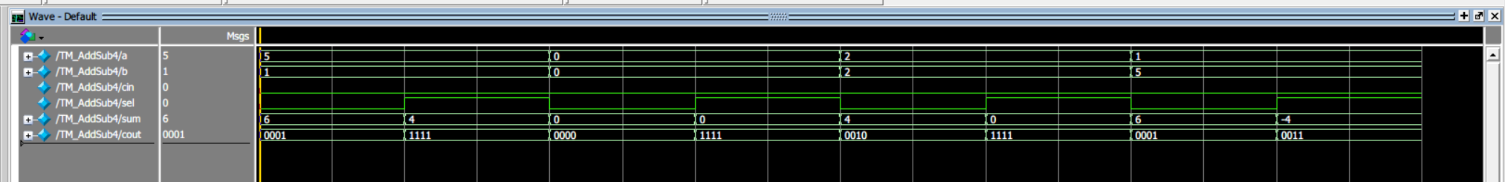
sel = 1 ;

#10

$stop ;

end

endmodule

**WaveForm:**

由左至右依序來看 5 + 1 = 6 正確，5 – 1 = 4 正確， 再來看到最後1 + 5 = 6 正確，1 – 5 = -4 正確(數值用decimal表示)，因此本實驗結果正確無誤。

**心得:本次實驗難度適中，算是讓大家重新拾起對於Verilog Code的記憶，對於我而言，很有幫助，很期待之後實驗的安排，希望能越來越進階。**