Altay İlker Yiğitel

22203024 10.03.2024

CS223 Sec:01 Lab02

adder:

module full\_adder(A, B, Cin, S, Cout);

input wire A, B, Cin;

output reg S, Cout;

always @\*

begin

S = A ^ B ^ Cin;

Cout = (A & B) | ((A ^ B) & Cin);

end

endmodule

module full\_adder\_s (  
 input a,b,cin;  
 output sum,carry;  
);  
  
wire w1,w2,w3,w4;   
  
xor(w1,a,b);  
xor(sum,w1,cin);   
  
and(w2,a,b);  
and(w3,b,cin);  
and(w4,cin,a);  
  
or(carry,w2,w3,w4);   
  
endmodule

substructer:

module full\_subtractor(A, B, Cin,Borrow, Diff);

input wire A, B, Cin;

output reg Borrow, Diff;

always @\* begin

// Difference

Diff = A ^ B ^ Cin;

// Borrow

Borrow = (~A & B) | (~A & Cin) | (B & Cin);

end

endmodule

module full\_substractor\_s (  
 input a,b,c;  
 output borrow,diff;  
);  
wire w1,w4,w5,w6;

xor (diff,a,b,c);

not n1(w1,a);

and a1(w4,w1,b);

and a2(w5,w1,c);

and a3(w6,b,c);

or o1(borrow,w4,w5,w6);  
  
endmodule

towadder:

module two\_adder(a,b,c,d,e,sum,sumtwo,carry);

input a,b,c,d,e;

output sum,sumtwo,carry;

wire w1,w2,w3,w4,w5,w6,w7,w8,cin,Cout;

xor(w1,a,b);

xor(sum,w1,c);

and(w2,a,b);

and(w3,b,c);

and(w4,c,a);

or(Cout,w2,w3,w4);

xor(w5,d,e);

xor(sumtwo,w5,Cout);

and(w6,d,e);

and(w7,e,Cout);

and(w8,Cout,d);

or(carry,w6,w7,w8);

endmodule