Lab I 半加器

sctructural:

module Half\_Add(a,b,sum,carry);

input a, b;

output sum, carry;

and u1(carry, a, b);

xor u2(sum, a, b);

endmodule

data flow:

module Half\_Add(a,b,sum,carry);

input a, b;

output sum, carry;

assign sum = a^b;

assign carry = a&b;

endmodule

Behavioral

module Half\_Add(a,b,sum,carry);

input a, b;

output sum, carry;

reg sum, carry;

always @(a,b)

begin

and u1(carry, a, b);

xor u2(sum, a, b);

end

endmodule

Lab II 全加器

sctructural:

module Full\_Add(sum, c\_out, a, b, c\_in);

input a, b, c\_in;

output sum, c\_out;

wire w1, w2, w3;

Half\_Add u1(w1,w2,a,b);

Half\_Add u2(sum,w3,c\_in,w1);

or u3(c\_out,w2,w3);

endmodule

//額外定義半加器

module Half\_Add(sum,carry,a,b);

input a, b;

output sum, carry;

and u1(carry, a, b);

xor u2(sum, a, b);

endmodule

data flow:

module Full\_Add(sum, c\_out, a, b, c\_in);

input a, b, c\_in;

output sum, c\_out;

assign sum = ((a^b) ^ c\_in);

assign c\_out = (a&b | (a^b)& c\_in);

endmodule

Behavioral

module Full\_Add(sum, c\_out, a, b, c\_in);

input a, b, c\_in;

output sum, c\_out;

reg sum, c\_out;

always @(a,b,c\_in)

begin

sum = ((a^b) ^ c\_in);

c\_out = (a&b | (a^b)& c\_in);

end

endmodule

