Lab 1: Introduction to Verilog HDL

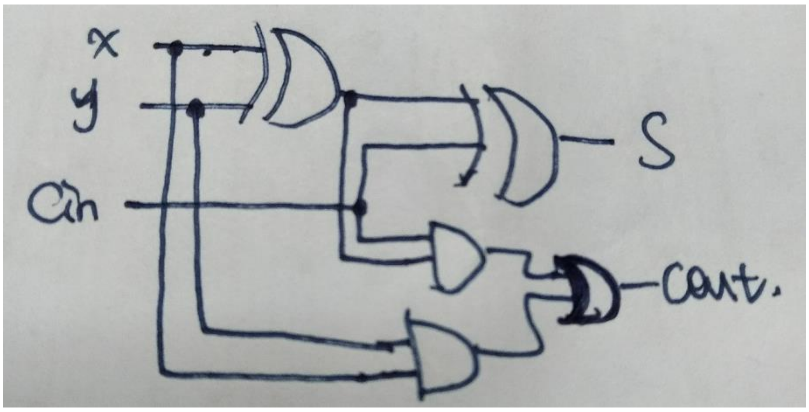
106061146 陳兆廷

1. Full adder:

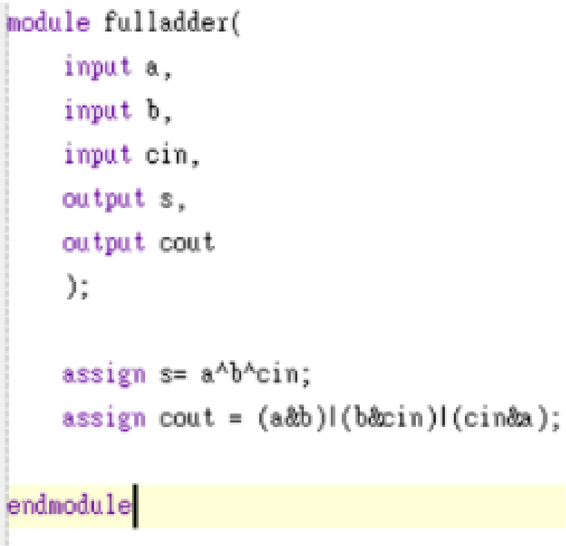
- Logic equations:

  $s = x \oplus y \oplus cin$ 、 $cout = x \cdot y + y \cdot cin + x \cdot cin$

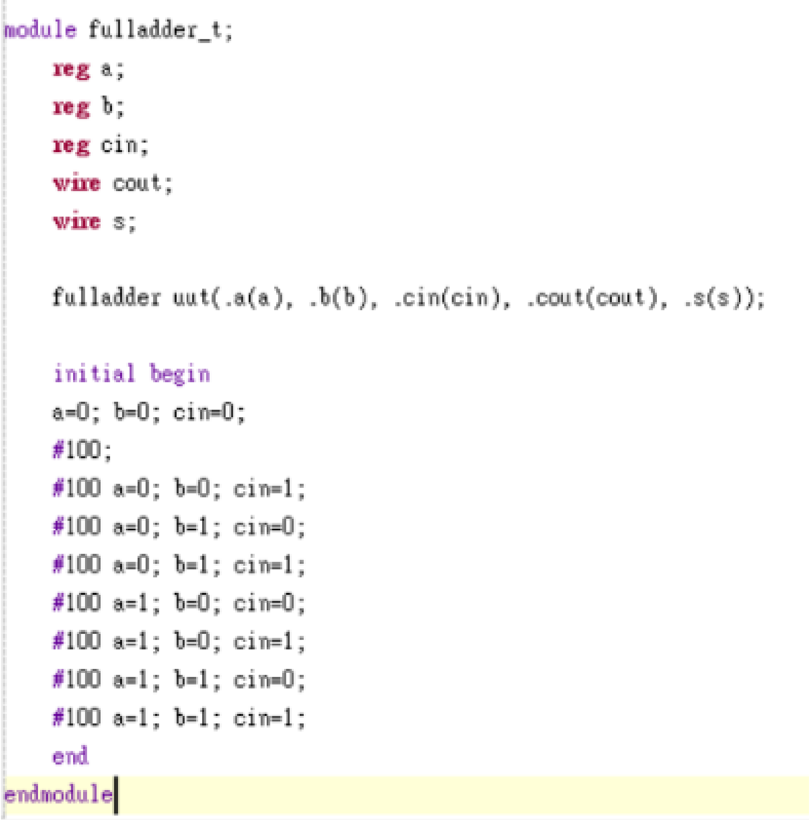
- Related logic diagram:



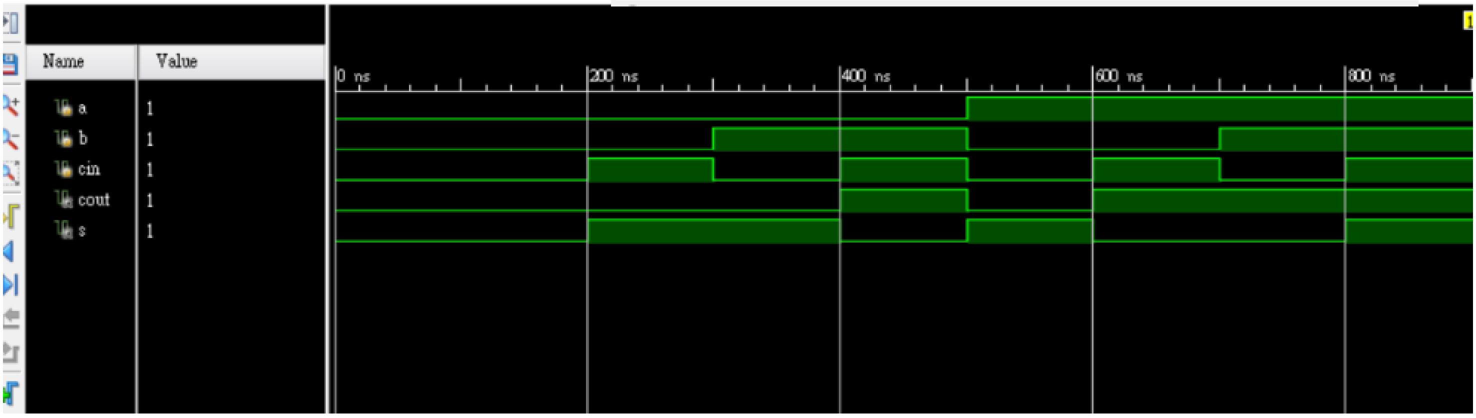
- verification:



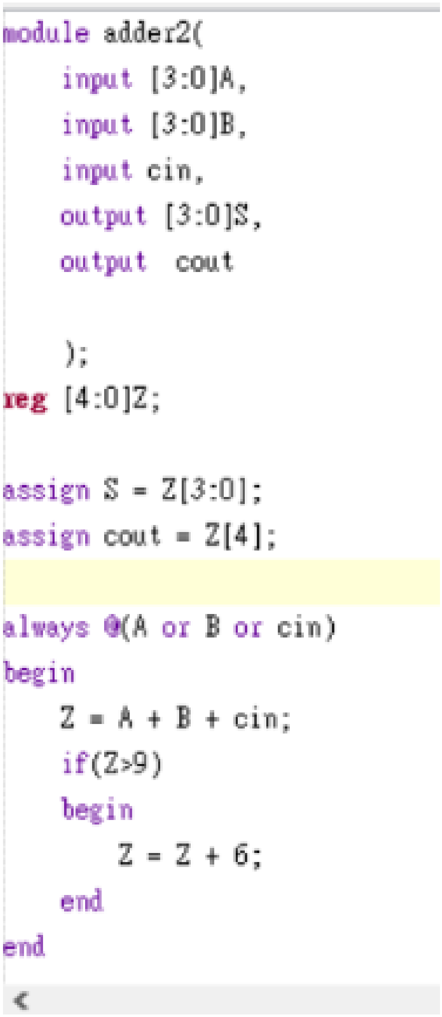
```
module fulladder(
    input a,
    input b,
    input cin,
    output s,
    output cout
    );

    assign s= a^b^cin;
    assign cout = (a&b)|(b&cin)|(cin&a);

endmodule
```



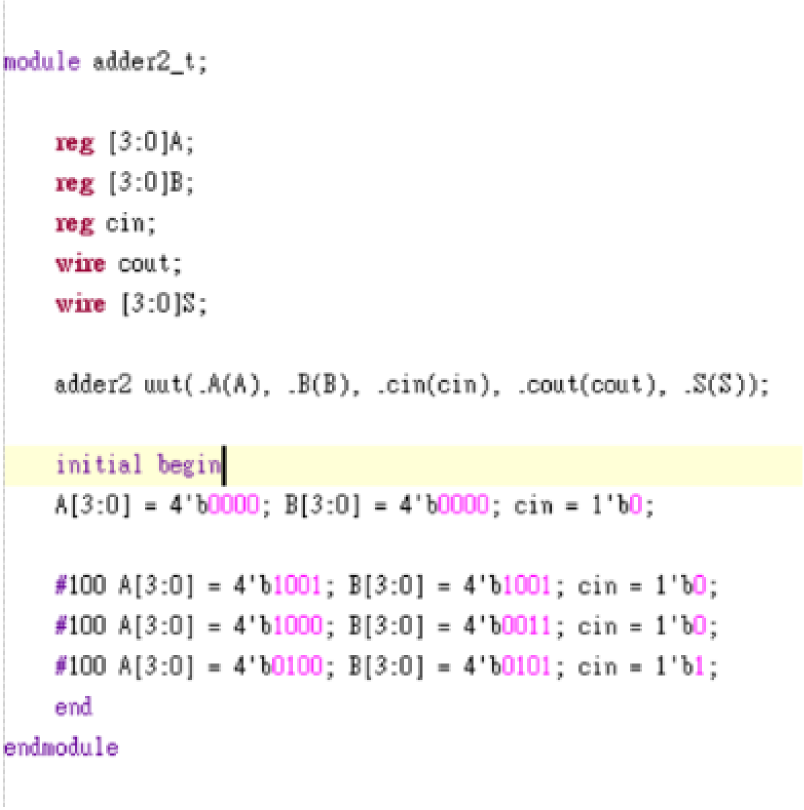
```
module fulladder_t;
    reg a;
    reg b;
    reg cin;
    wire cout;
    wire s;

    fulladder uut(.a(a), .b(b), .cin(cin), .cout(cout), .s(s));

    initial begin
    a=0; b=0; cin=0;
    #100;
    #100 a=0; b=0; cin=1;
    #100 a=0; b=1; cin=0;
    #100 a=0; b=1; cin=1;
    #100 a=1; b=0; cin=0;
    #100 a=1; b=0; cin=1;
    #100 a=1; b=1; cin=0;
    #100 a=1; b=1; cin=1;
    end
endmodule
```



2. Decimal adder:



```
module adder2(
    input [3:0]A,
    input [3:0]B,
    input cin,
    output [3:0]S,
    output  cout

    );
reg [4:0]Z;

assign S = Z[3:0];
assign cout = Z[4];

always @(A or B or cin)
begin
    Z = A + B + cin;
    if(Z>9)
    begin
        Z = Z + 6;
    end
end
```

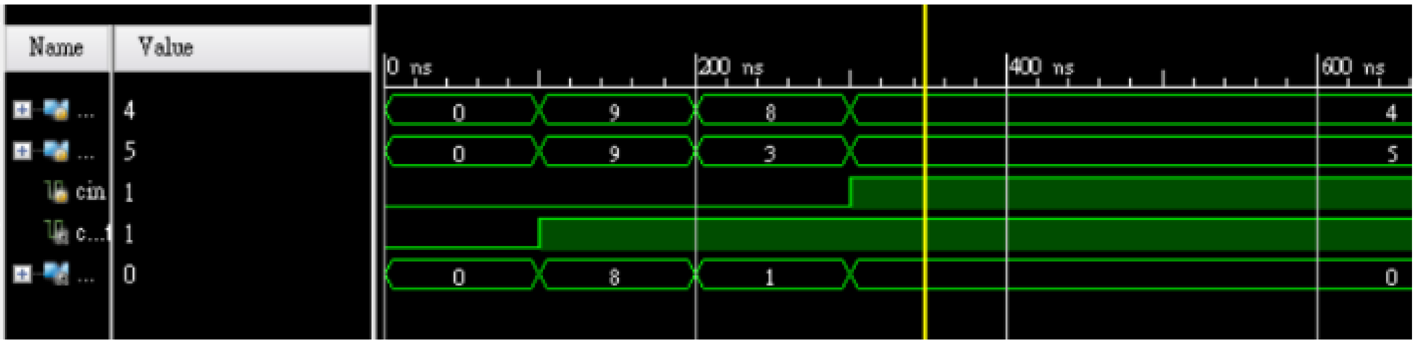


```
module adder2_t;

    reg [3:0]A;
    reg [3:0]B;
    reg cin;
    wire cout;
    wire [3:0]S;

    adder2 uut(.A(A), .B(B), .cin(cin), .cout(cout), .S(S));

    initial begin
    A[3:0] = 4'b0000; B[3:0] = 4'b0000; cin = 1'b0;

    #100 A[3:0] = 4'b1001; B[3:0] = 4'b1001; cin = 1'b0;
    #100 A[3:0] = 4'b1000; B[3:0] = 4'b0011; cin = 1'b0;
    #100 A[3:0] = 4'b0100; B[3:0] = 4'b0101; cin = 1'b1;
    end
endmodule
```

測試值：9+9=18

8+3=11

4+5+1(cin) = 10


Name
Value
...
...
cin
c...
...
4
5
1
1
0
0 ns
200 ns
400 ns
600 ns
0
9
8
4
0
9
3
5
0
8
1
0