# 논리회로설계 (COMP311004) HW_3

2019112130 최부광

| Source code | O |
|---|---|
| Waveform | O |
| Text proof | X |

## 1. Source code

- 1 bit adder

add1.v | add4.v | add4_tb.v

```
1  module add1(in1, in2, cin, sum, cout);
2    input in1,in2, cin;
3    output sum, cout;
4
5    assign sum = ((in1^in2)^cin);
6    assign cout = (in1&in2)|((in1^in2)&cin);
7  endmodule
```

- 4 bit adder

add1.v | add4.v | add4_tb.v

```
1  module add4(a,b,s,cout);
2    input [3:0]a;
3    input [3:0]b;
4    output [3:0]s;
5    output cout;
6
7    add1 b0(a[0], b[0], 1'b0, s[0], q); // LSB
8    add1 b1(a[1], b[1], q, s[1], w);
9    add1 b2(a[2], b[2], w, s[2], e);
10   add1 b3(a[3], b[3], e, s[3], cout); // MSB
11 endmodule
```

- Testbench

add1.v | add4.v | add4_tb.v

```
1  `timescale 1us/1us
2  module add4_tb();
3    reg [3:0]a;        // 4-bit reg vector
4    reg [3:0]b;        // 4-bit reg vector
5    wire [3:0]s_2130; // 4-bit wire vector
6    wire cout;         // scalar wire
7
8    add4 half(a,b,s_2130,cout);
9
10   initial begin      // create output file
11     $dumpfile("output.vcd");
12     $dumpvars(0);
13   end
14
15   initial begin
16     a = 4'b0000;
17     b = 4'b0000;
18   end
19
20   always
21     #1 a = $urandom%16;      // a에 4-bit random value 할당
22
23   always
24     #1 b = $urandom%16;      // b에 4-bit random value 할당
25
26   always
27     #30 $finish;
28
29 endmodule
```

# 논리회로설계 (COMP311004) HW_3

2019112130 최부광

| | |
|---|---|
| Source code | O |
| Waveform | O |
| Text proof | X |

## 2. Waveform