

```
Counter.v
1 module Register32Bit (
2     input clock,
3     input clear,
4     input[31:0] in,
5     output reg[31:0] out);
6
7     always @(posedge clear, negedge clock)
8     if (clear) begin
9         out = 0;
10    end
11    else begin
12        out = in;
13    end
14 endmodule
15
16 module Mux32Bit2To1 (
17     input[31:0] in0,
18     input[31:0] in1,
19     input s,
20     output reg[31:0] out);
21
22     always @(in0, in1, s)
23     case (s)
24         0: out = in0;
25         1: out = in1;
26     endcase
27 endmodule
28
29 module Adder32Bit (
30     input[31:0] a,
31     input[31:0] b,
32     output[31:0] s);
33
34     assign s = a + b;
35 endmodule
36
37 module Counter (
38     input clock,
39     input clear,
40     input enable,
41     input reset,
42     output[31:0] out);
43
44     wire mux_out1;
45     wire mux_out2;
46     wire register_out;
47     wire adder_out;
48
49     Adder32Bit adder(out, 1, adder_out);
50     Mux32Bit2To1 mux1(out, adder_out, enable, mux_out1);
51     Mux32Bit2To1 mux2(mux_out1, 0, reset, mux_out2);
52     Register32Bit register(clock, clear, mux_out2, out);
53 endmodule
```