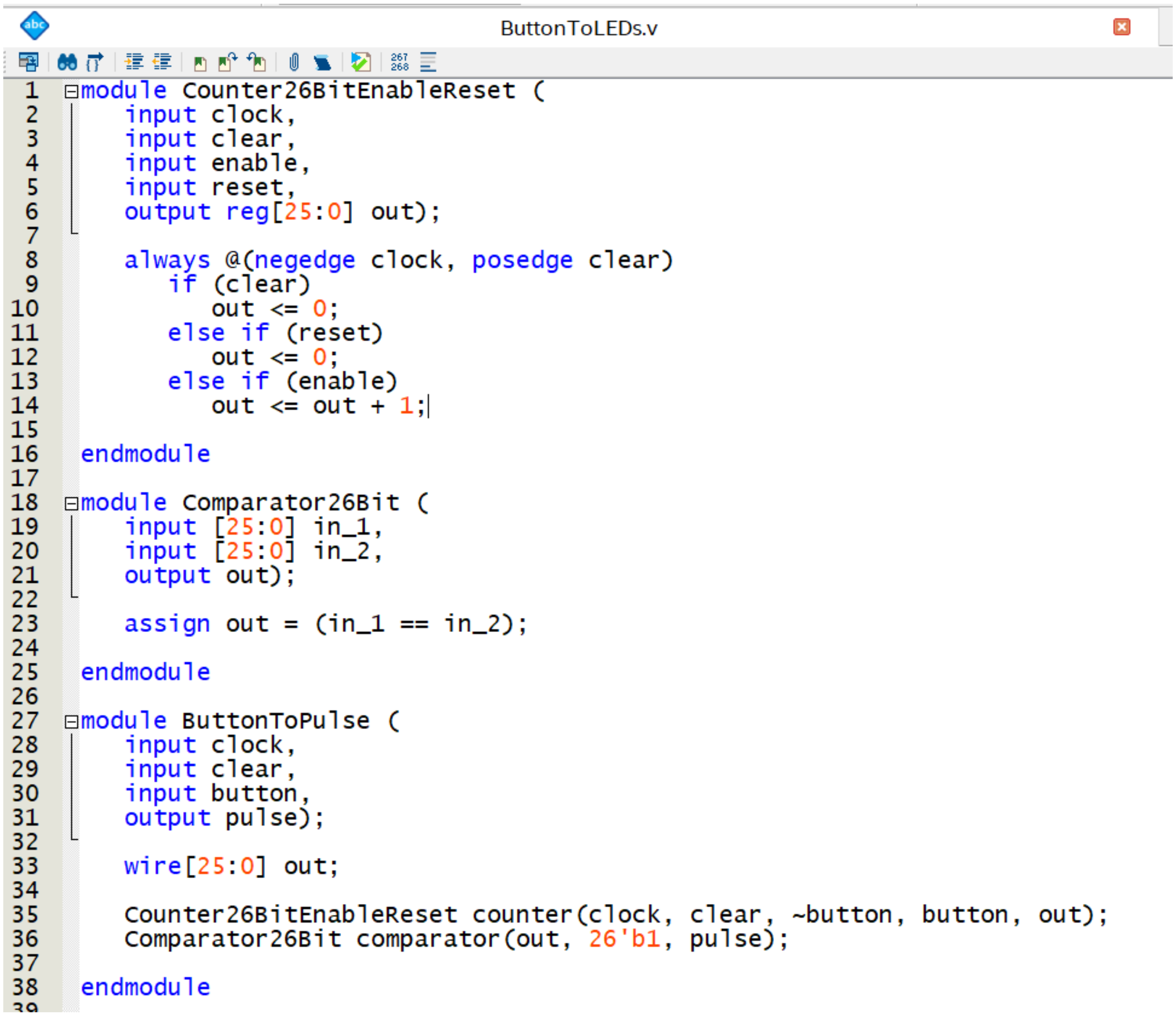
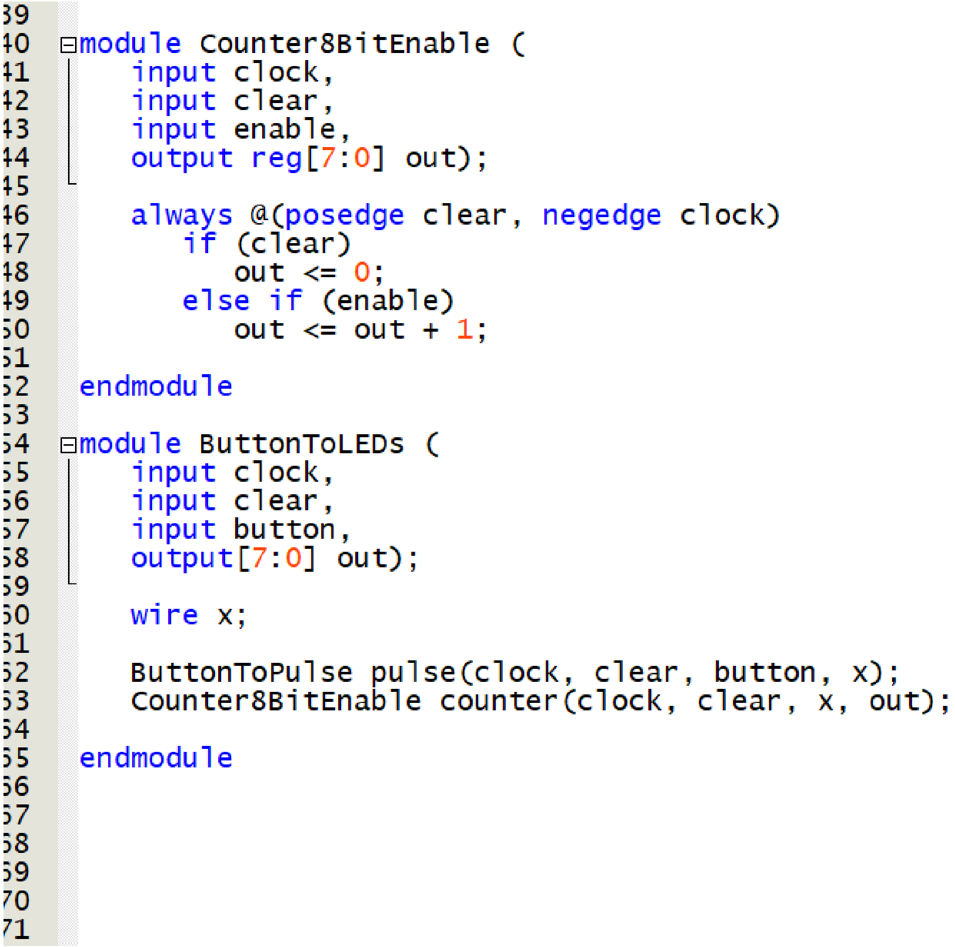


```verilog
module Counter26BitEnableReset (
   input clock,
   input clear,
   input enable,
   input reset,
   output reg[25:0] out);

   always @(negedge clock, posedge clear)
      if (clear)
         out <= 0;
      else if (reset)
         out <= 0;
      else if (enable)
         out <= out + 1;

endmodule

module Comparator26Bit (
   input [25:0] in_1,
   input [25:0] in_2,
   output out);

   assign out = (in_1 == in_2);

endmodule

module ButtonToPulse (
   input clock,
   input clear,
   input button,
   output pulse);

   wire[25:0] out;

   Counter26BitEnableReset counter(clock, clear, ~button, button, out);
   Comparator26Bit comparator(out, 26'b1, pulse);

endmodule
```



```
module Counter8BitEnable (
   input clock,
   input clear,
   input enable,
   output reg[7:0] out);

   always @(posedge clear, negedge clock)
      if (clear)
         out <= 0;
      else if (enable)
         out <= out + 1;

endmodule

module ButtonToLEDs (
   input clock,
   input clear,
   input button,
   output[7:0] out);

   wire x;

   ButtonToPulse pulse(clock, clear, button, x);
   Counter8BitEnable counter(clock, clear, x, out);

endmodule
```