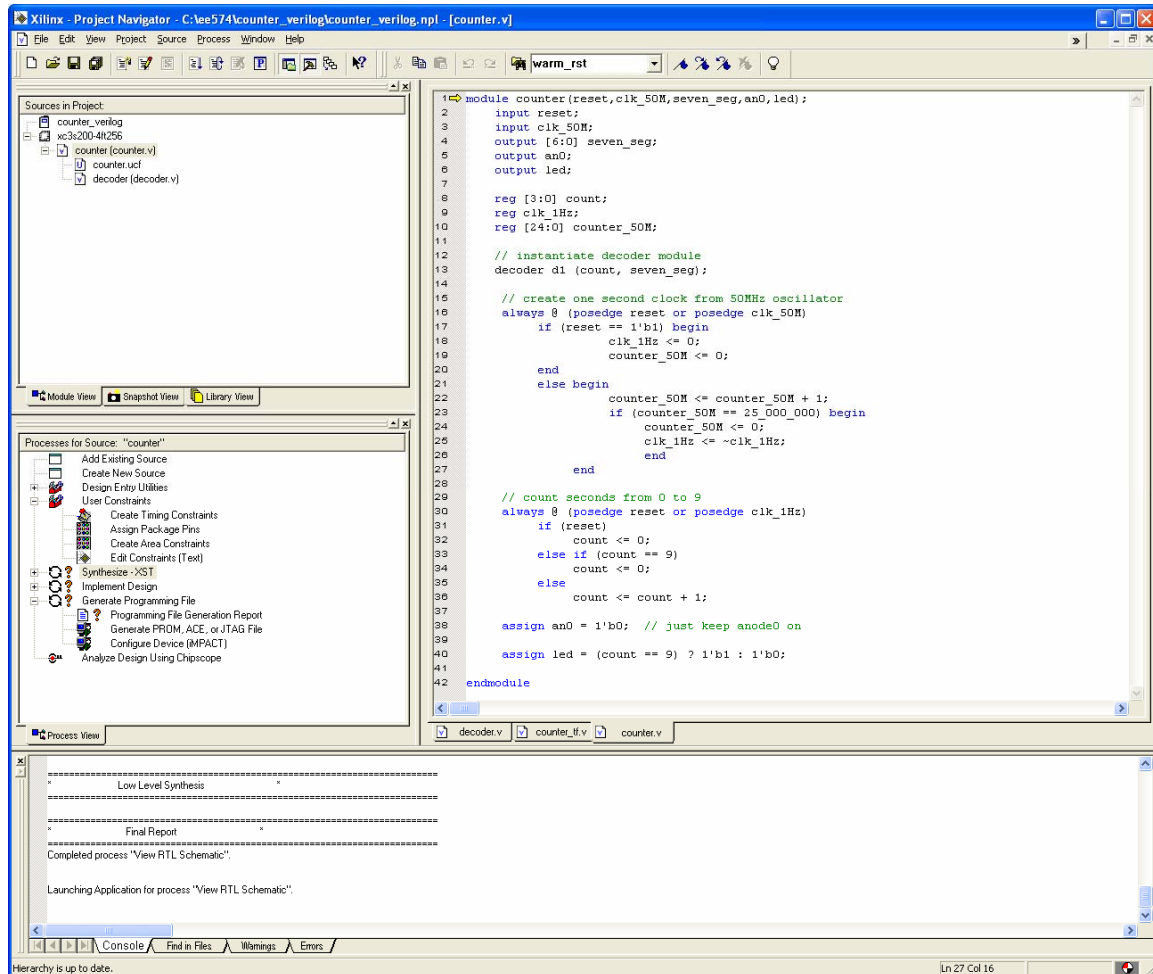


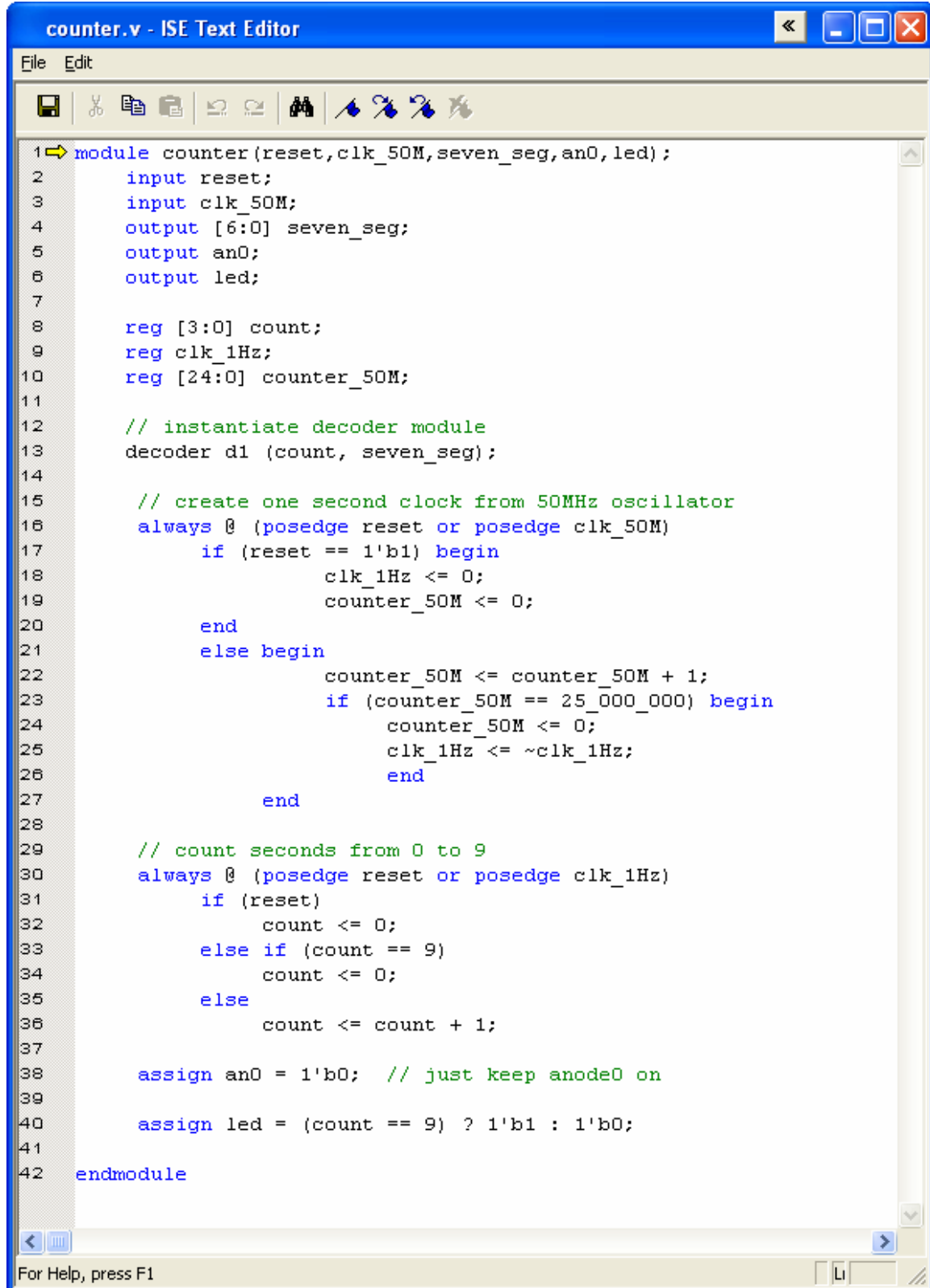
Spartan 3 Starter Board Tutorial (Counter with decoder in Verilog)

Jim Duckworth, March 2005, WPI.

This design shows how to create a simple sequential circuit (a counter). It also demonstrates hierarchical design by using a separate decoder component that converts a binary count value to a seven segment display.

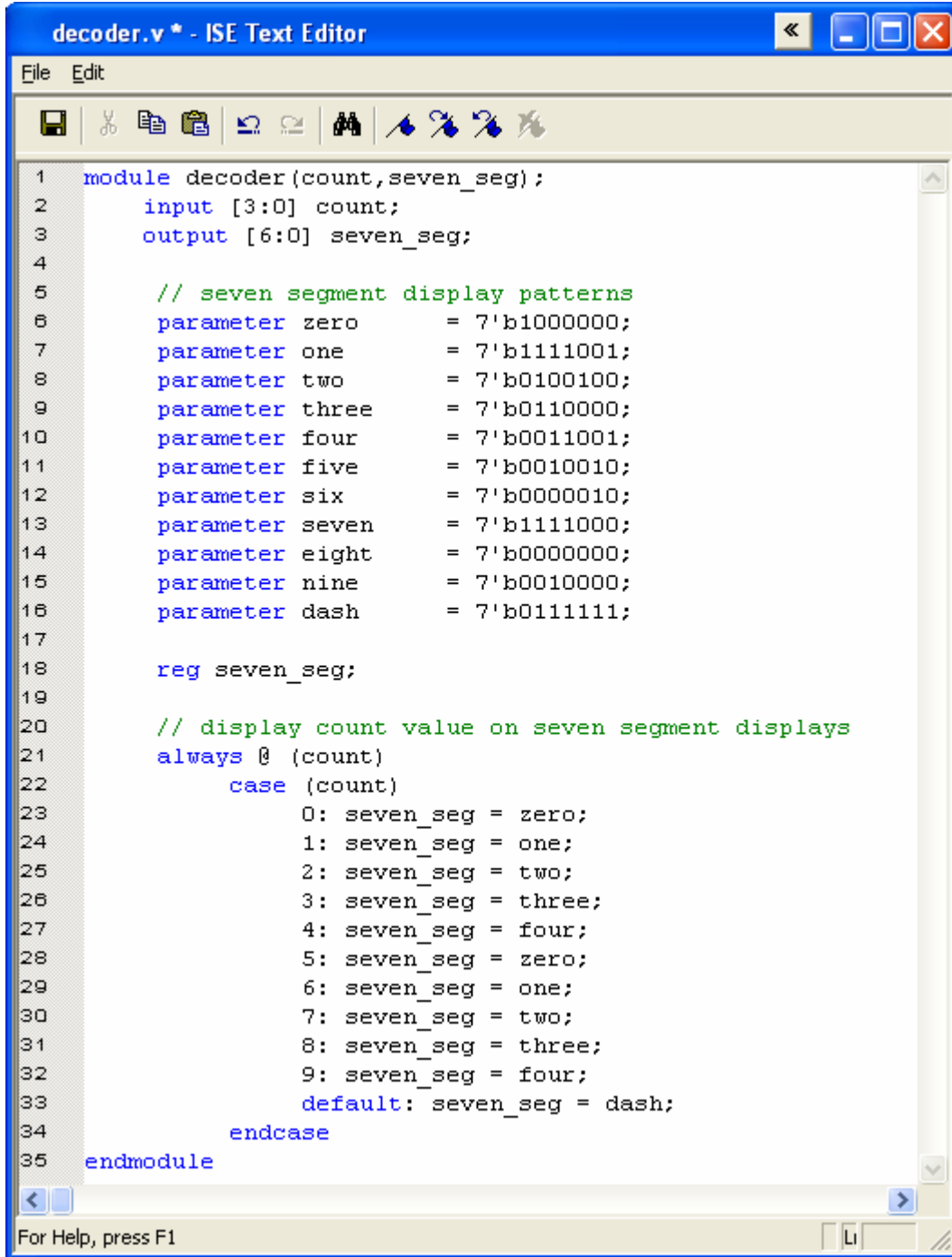
There are three files for this design as shown in the *Sources in Project* Window:



Top level entity and architecture description:

```
1 module counter(reset,clk_50M,seven_seg,an0,led);
2     input reset;
3     input clk_50M;
4     output [6:0] seven_seg;
5     output an0;
6     output led;
7
8     reg [3:0] count;
9     reg clk_1Hz;
10    reg [24:0] counter_50M;
11
12    // instantiate decoder module
13    decoder d1 (count, seven_seg);
14
15    // create one second clock from 50MHz oscillator
16    always @ (posedge reset or posedge clk_50M)
17        if (reset == 1'b1) begin
18            clk_1Hz <= 0;
19            counter_50M <= 0;
20        end
21        else begin
22            counter_50M <= counter_50M + 1;
23            if (counter_50M == 25_000_000) begin
24                counter_50M <= 0;
25                clk_1Hz <= ~clk_1Hz;
26            end
27        end
28
29    // count seconds from 0 to 9
30    always @ (posedge reset or posedge clk_1Hz)
31        if (reset)
32            count <= 0;
33        else if (count == 9)
34            count <= 0;
35        else
36            count <= count + 1;
37
38    assign an0 = 1'b0; // just keep anode0 on
39
40    assign led = (count == 9) ? 1'b1 : 1'b0;
41
42 endmodule
```

For Help, press F1

Decoder Module Description:

```
1  module decoder(count,seven_seg);
2      input [3:0] count;
3      output [6:0] seven_seg;
4
5      // seven segment display patterns
6      parameter zero      = 7'b1000000;
7      parameter one       = 7'b1111001;
8      parameter two       = 7'b0100100;
9      parameter three     = 7'b0110000;
10     parameter four      = 7'b0011001;
11     parameter five      = 7'b0010010;
12     parameter six       = 7'b0000010;
13     parameter seven     = 7'b1111000;
14     parameter eight     = 7'b0000000;
15     parameter nine      = 7'b0010000;
16     parameter dash      = 7'b0111111;
17
18     reg seven_seg;
19
20     // display count value on seven segment displays
21     always @ (count)
22         case (count)
23             0: seven_seg = zero;
24             1: seven_seg = one;
25             2: seven_seg = two;
26             3: seven_seg = three;
27             4: seven_seg = four;
28             5: seven_seg = zero;
29             6: seven_seg = one;
30             7: seven_seg = two;
31             8: seven_seg = three;
32             9: seven_seg = four;
33             default: seven_seg = dash;
34         endcase
35     endmodule
```

For Help, press F1

User constraints file for counter:

```

counter.ucf - ISE Text Editor
File Edit
1 #PACE: Start of Constraints generated by PACE
2
3 #PACE: Start of PACE I/O Pin Assignments
4 NET "an0" LOC = "d14" ;
5 NET "clk_50M" LOC = "t9" ;
6 NET "led" LOC = "k12" ;
7 NET "reset" LOC = "l14" ;
8 NET "seven_seg<0>" LOC = "e14" ;
9 NET "seven_seg<1>" LOC = "g13" ;
10 NET "seven_seg<2>" LOC = "n15" ;
11 NET "seven_seg<3>" LOC = "p15" ;
12 NET "seven_seg<4>" LOC = "r16" ;
13 NET "seven_seg<5>" LOC = "f13" ;
14 NET "seven_seg<6>" LOC = "n16" ;
15
16 #PACE: Start of PACE Area Constraints
17
18 #PACE: Start of PACE Prohibit Constraints
19
20 #PACE: End of Constraints generated by PACE
21
For Help, press F1

```

