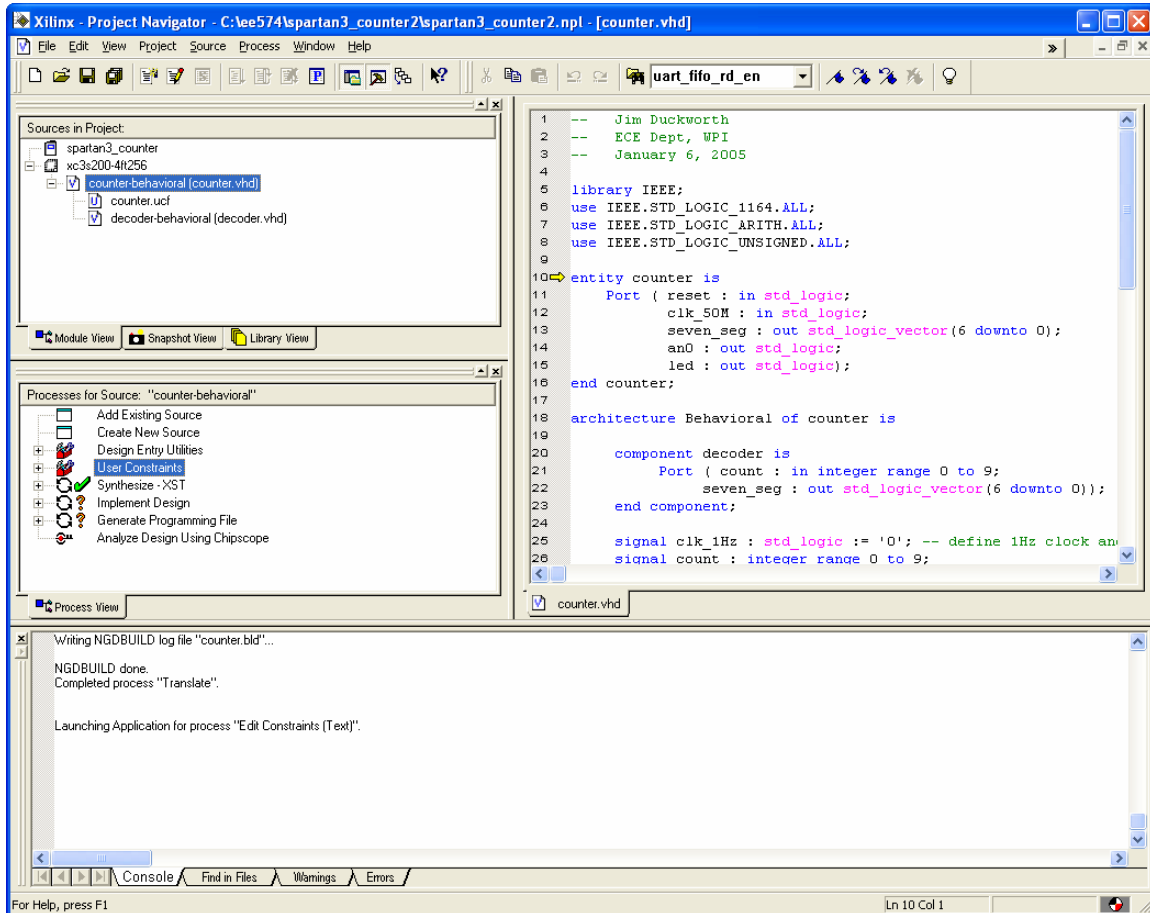


Spartan 3 Starter Board Tutorial (Counter with decoder)

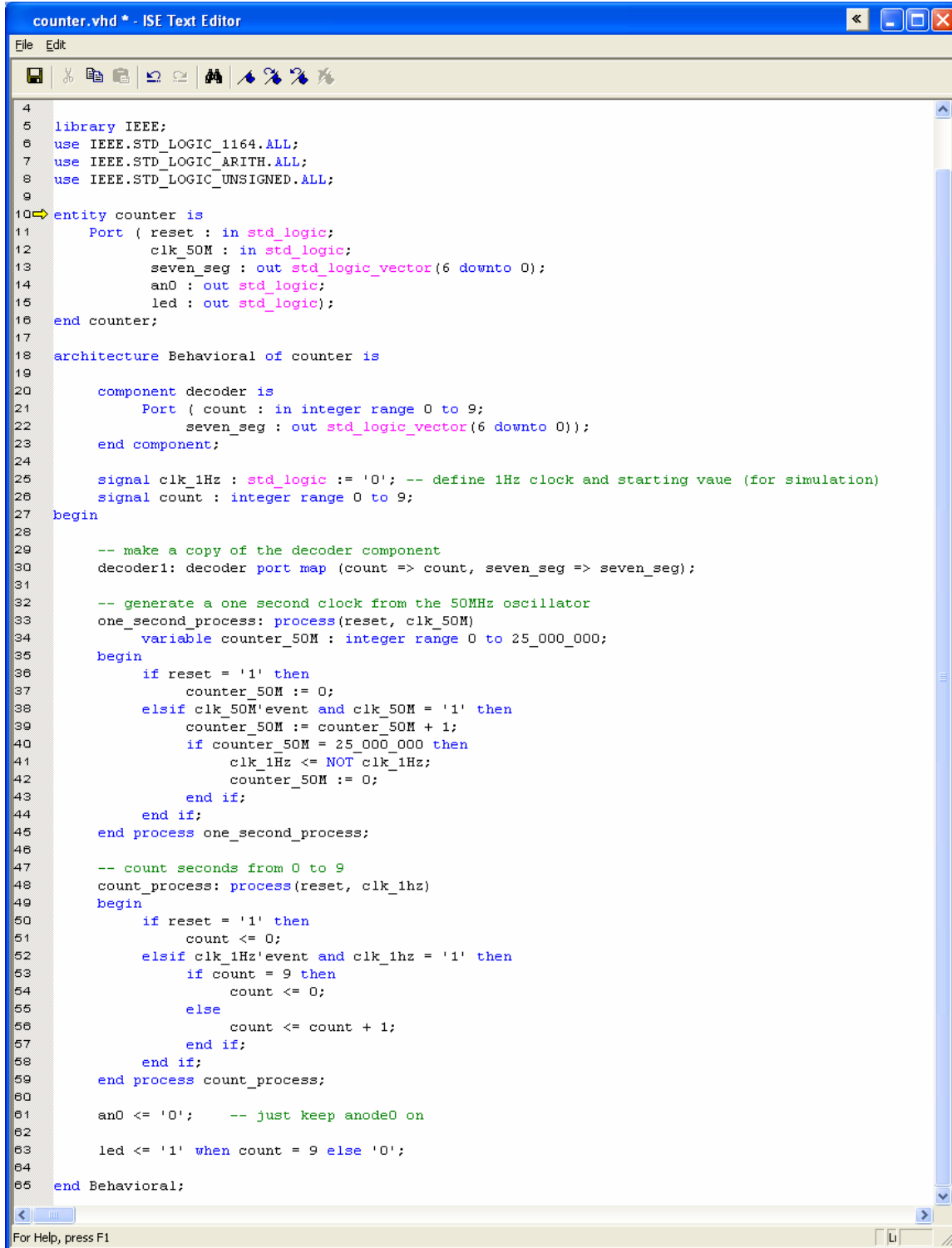
Jim Duckworth, January 2005, WPI.

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:



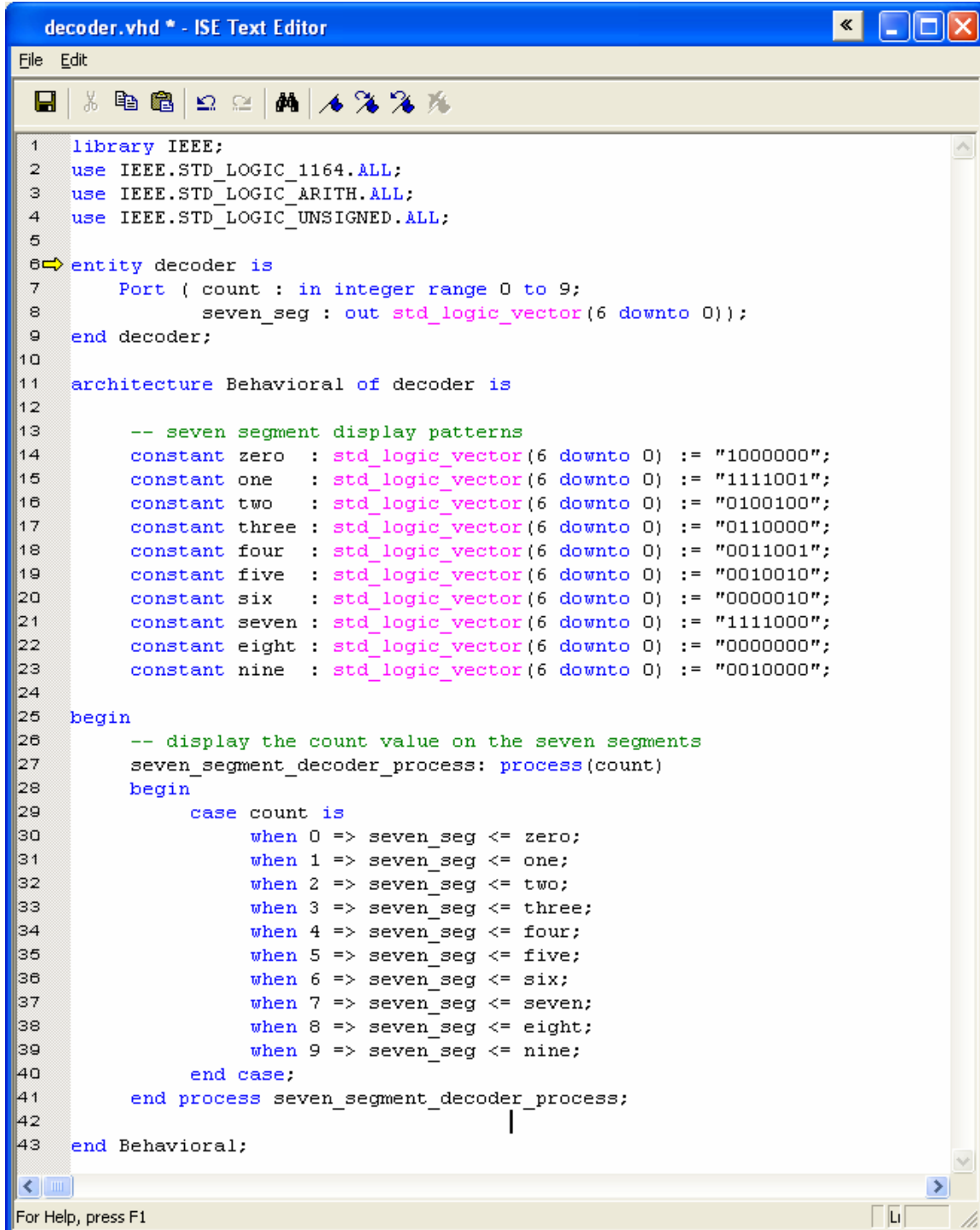
```

counter.vhd * - ISE Text Editor
File Edit
4
5 library IEEE;
6 use IEEE.STD_LOGIC_1164.ALL;
7 use IEEE.STD_LOGIC_ARITH.ALL;
8 use IEEE.STD_LOGIC_UNSIGNED.ALL;
9
10 entity counter is
11     Port ( reset : in std_logic;
12           clk_50M : in std_logic;
13           seven_seg : out std_logic_vector(6 downto 0);
14           an0 : out std_logic;
15           led : out std_logic);
16 end counter;
17
18 architecture Behavioral of counter is
19
20     component decoder is
21         Port ( count : in integer range 0 to 9;
22               seven_seg : out std_logic_vector(6 downto 0));
23     end component;
24
25     signal clk_1Hz : std_logic := '0'; -- define 1Hz clock and starting vaue (for simulation)
26     signal count : integer range 0 to 9;
27 begin
28
29     -- make a copy of the decoder component
30     decoder1: decoder port map (count => count, seven_seg => seven_seg);
31
32     -- generate a one second clock from the 50MHz oscillator
33     one_second_process: process(reset, clk_50M)
34         variable counter_50M : integer range 0 to 25_000_000;
35     begin
36         if reset = '1' then
37             counter_50M := 0;
38         elsif clk_50M'event and clk_50M = '1' then
39             counter_50M := counter_50M + 1;
40             if counter_50M = 25_000_000 then
41                 clk_1Hz <= NOT clk_1Hz;
42                 counter_50M := 0;
43             end if;
44         end if;
45     end process one_second_process;
46
47     -- count seconds from 0 to 9
48     count_process: process(reset, clk_1hz)
49     begin
50         if reset = '1' then
51             count <= 0;
52         elsif clk_1Hz'event and clk_1hz = '1' then
53             if count = 9 then
54                 count <= 0;
55             else
56                 count <= count + 1;
57             end if;
58         end if;
59     end process count_process;
60
61     an0 <= '0'; -- just keep anode0 on
62
63     led <= '1' when count = 9 else '0';
64
65 end Behavioral;

```

For Help, press F1

Decoder Entity and Architecture Description:



```
decoder.vhd * - ISE Text Editor
File Edit
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.STD_LOGIC_ARITH.ALL;
4 use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6 entity decoder is
7     Port ( count : in integer range 0 to 9;
8           seven_seg : out std_logic_vector(6 downto 0));
9 end decoder;
10
11 architecture Behavioral of decoder is
12
13     -- seven segment display patterns
14     constant zero : std_logic_vector(6 downto 0) := "1000000";
15     constant one  : std_logic_vector(6 downto 0) := "1111001";
16     constant two  : std_logic_vector(6 downto 0) := "0100100";
17     constant three: std_logic_vector(6 downto 0) := "0110000";
18     constant four : std_logic_vector(6 downto 0) := "0011001";
19     constant five  : std_logic_vector(6 downto 0) := "0010010";
20     constant six   : std_logic_vector(6 downto 0) := "0000010";
21     constant seven : std_logic_vector(6 downto 0) := "1111000";
22     constant eight : std_logic_vector(6 downto 0) := "0000000";
23     constant nine  : std_logic_vector(6 downto 0) := "0010000";
24
25 begin
26     -- display the count value on the seven segments
27     seven_segment_decoder_process: process(count)
28     begin
29         case count is
30             when 0 => seven_seg <= zero;
31             when 1 => seven_seg <= one;
32             when 2 => seven_seg <= two;
33             when 3 => seven_seg <= three;
34             when 4 => seven_seg <= four;
35             when 5 => seven_seg <= five;
36             when 6 => seven_seg <= six;
37             when 7 => seven_seg <= seven;
38             when 8 => seven_seg <= eight;
39             when 9 => seven_seg <= nine;
40         end case;
41     end process seven_segment_decoder_process;
42
43 end Behavioral;
```

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

```

Synthesized Results:

