```
1   library IEEE;
2   use IEEE.STD_LOGIC_1164.ALL;
3
4   -- Uncomment the following library declaration if using
5   -- arithmetic functions with Signed or Unsigned values
6   --use IEEE.NUMERIC_STD.ALL;
7
8   -- Uncomment the following library declaration if instantiating
9   -- any Xilinx leaf cells in this code.
10  --library UNISIM;
11  --use UNISIM.VComponents.all;
12
13  entity top_tb is
14  --  Port ( );
15  end top_tb;
16
17  architecture Behavioral of top_tb is
18
19      --this entity needs to be defined in your top.vhd file
20      component top is
21      Port (
22          sw       : in  STD_LOGIC_VECTOR (15 downto 0);
23          seg      : out std_logic_vector(6 downto 0);
24          dp       : out std_logic;
25          an       : out std_logic_vector(3 downto 0);
26          led      : out STD_LOGIC_VECTOR (15 downto 0);
27          clk      : in  std_logic
28      );
29      end component;
30
31      --clock period. Set to 100 MHz here
32      constant clk_per : time := 10ns;
33
34      --signals to do the binding to the "top" entity
35      signal sw        : std_logic_vector(15 downto 0);
36      signal seg       : std_logic_vector(6 downto 0);
37      signal dp        : std_logic;
38      signal an        : std_logic_vector(3 downto 0);
39      signal led       : STD_LOGIC_VECTOR (15 downto 0);
40      signal clk       : std_logic := '0';
41
42      --signal counter   : unsigned(15 downto 0) := '0000000000000000';
43  begin
44
45      uut: top port map (
46          sw     => sw,
47          seg    => seg,
48          dp     => dp,
49          an     => an,
50          led    => led,
51          clk    => clk
52      );
53
54      --add code here to test your outputs for correct operation (change the value of "sw")
55      --only needs code to be added in simulation, not in the final file to be uploaded
        to the FPGA
56      --this is step 7 and 8 in Part 1
57
58      sw_proc: process
59      begin
60          sw <= x"7b10";
61          wait for clk_per;
62      end process sw_proc;
63
64      --clock process, high and low for half the clock period
65      clk_proc: process
```

```
66      begin
67          wait for clk_per;
68          clk <= not(clk);
69
70          -- counter <= counter + 1;
71      end process clk_proc;
72
73  end Behavioral;
```