```
1   ----------------------------------------------------------------------------------
2   -- Company: HSRW
3   -- Engineer: Andy Stamm
4   --
5   -- Create Date:    08:49:12 12/17/2015
6   -- Design Name:
7   -- Module Name:    7 segment 6bit to decimal - Behavioral
8   -- Project Name:     Lab 2.3
9   -- Target Devices:
10  -- Tool versions:
11  -- Description:
12  --
13  -- Dependencies:
14  --
15  -- Revision:
16  -- Revision 0.01 - File Created
17  -- Additional Comments:
18  --
19  ----------------------------------------------------------------------------------
20  library IEEE;
21  use IEEE.STD_LOGIC_1164.ALL;
22  use IEEE.STD_LOGIC_ARITH.ALL;
23  use IEEE.STD_LOGIC_UNSIGNED.ALL;
24
25  -- Uncomment the following library declaration if using
26  -- arithmetic functions with Signed or Unsigned values
27  --use IEEE.NUMERIC_STD.ALL;
28
29  -- Uncomment the following library declaration if instantiating
30  -- any Xilinx primitives in this code.
31  --library UNISIM;
32  --use UNISIM.VComponents.all;
33
34  library IEEE;
35  use IEEE.STD_LOGIC_1164.ALL;
36  use IEEE.STD_LOGIC_ARITH.ALL;
37  use IEEE.STD_LOGIC_UNSIGNED.ALL;
38
39  entity seven_segment is
40          Port (  clk, SW0, SW1, SW2, SW3, SW4, SW5  : in     STD_LOGIC;            --
    inputs to the entity
41                  COM0, COM1 : out  STD_LOGIC;
    --ouput anode of the seven segment display
42                  SEGA0, SEGB0, SEGC0, SEGD0, SEGE0, SEGF0, SEGG0 : out  STD_LOGIC; --
    ouputs of the first seven segment dislay
43                  SEGA1, SEGB1, SEGC1, SEGD1, SEGE1, SEGF1, SEGG1 : out  STD_LOGIC  --
    ouputs of the first seven segment dislay
44
45                );
46  end entity seven_segment;
47
48  architecture Behavioral of seven_segment is
49  signal temp_1, temp_2: std_logic_vector(6 downto 0);  -- vector to drive the two
    7-segment displays
50  signal btn: std_logic_vector(5 downto 0);             -- vector to save the switch
    state
51  signal digit0, digit1, result: integer := 0;          -- signals to save integers
```

```
 52
 53       begin
 54          process(clk)
 55              begin
 56               if clk'event and clk='1' then -- wait for the rising edge of the clock
    then assign SW state to vector
 57                  btn(0)<=SW0;
 58                  btn(1)<=SW1;
 59                  btn(2)<=SW2;
 60                  btn(3)<=SW3;
 61                  btn(4)<=SW4;
 62                  btn(5)<=SW5;
 63              end if;
 64          end process;
 65
 66       process
 67          begin
 68             result <= conv_integer(unsigned(btn));    -- convert bitcode to integer
 69
 70             -- split one integer into two singe digits
 71                if result < 10 then
 72                digit0 <= result;
 73                digit1 <= 0;
 74                elsif result < 20 then
 75                digit0 <= result - 10;
 76                digit1 <= 1;
 77                elsif result < 30 then
 78                digit0 <= result - 20;
 79                digit1 <= 2;
 80                elsif result < 40 then
 81                digit0 <= result - 30;
 82                digit1 <= 3;
 83                elsif result < 50 then
 84                digit0 <= result - 40;
 85                digit1 <= 4;
 86                elsif result < 60 then
 87                digit0 <= result - 50;
 88                digit1 <= 5;
 89                elsif result < 70 then
 90                digit0 <= result - 60;
 91                digit1 <= 6;
 92                end if;
 93
 94             -- assign the decoded output to the first seven segment display outputs
 95             SEGA0 <= temp_1(0);
 96             SEGB0 <= temp_1(1);
 97             SEGC0 <= temp_1(2);
 98             SEGD0 <= temp_1(3);
 99             SEGE0 <= temp_1(4);
100             SEGF0 <= temp_1(5);
101             SEGG0 <= temp_1(6);
102             COM0<='0';        -- enable the first seven segment display
103
104             -- assign the decoded output to the first seven segment display outputs
105             SEGA1 <= temp_2(0);
106             SEGB1 <= temp_2(1);
107             SEGC1 <= temp_2(2);
```

```
108            SEGD1 <= temp_2(3);
109            SEGE1 <= temp_2(4);
110            SEGF1 <= temp_2(5);
111            SEGG1 <= temp_2(6);
112            COM1<='0';        -- enable the first seven segment display
113        end process;
114
115            -- decoding for the first seven segment display
116            with digit0 Select
117               temp_1<="0111111" when 0,
118                     "0000110" when 1,
119                     "1011011" when 2,
120                     "1001111" when 3,
121                     "1100110" when 4,
122                     "1101101" when 5,
123                     "1111101" when 6,
124                     "0000111" when 7,
125                     "1111111" when 8,
126                     "1101111" when 9,
127                     "1000000" when others;
128
129            -- decoding for the second seven segment display
130            with digit1 Select
131               temp_2<="0111111" when 0,
132                     "0000110" when 1,
133                     "1011011" when 2,
134                     "1001111" when 3,
135                     "1100110" when 4,
136                     "1101101" when 5,
137                     "1111101" when 6,
138                     "1000000" when others;
139
140    end Behavioral;
141
142
143
```