~\MobaXterm\home\QuartusProjects\Homework\hw8\hw8_square_wave\sqrWv.vhd

```
1  -- Name: Ian Flury
2  -- Assignment: HW8, P1 programmable square wave generator.
3  -- Class: CPEN430 Digital System Design
4  -- Date: 10/18/2023
5
6  library ieee;
7  use ieee.std_logic_1164.all;
8  use ieee.std_logic_unsigned.all;
9  use ieee.numeric_std.all;
10 use work.all;
11
12
13 entity sqrWv is
14     port (
15         clock : in std_logic;
16         RST : in std_logic;
17         H : in std_logic_vector(3 downto 0);
18         L : in std_logic_vector(3 downto 0);
19         wave_out : out std_logic
20     );
21 end sqrWv;
22
23 architecture rtl of sqrWv is
24
25     signal slow_clock : std_logic;
26     signal high_state : std_logic;
27
28 begin
29
30     clock_100ns_period: process(clock)
31         variable clk_cntr : unsigned(3 downto 0);
32         begin
33         if RST = '1' then
34             clk_cntr := (others => '0');
35             slow_clock <= '0';
36         elsif clock'EVENT and clock = '1' then
37             clk_cntr := clk_cntr + 1;
38             if clk_cntr > 4 then
39                 clk_cntr := (others => '0');
40                 slow_clock <= not slow_clock;
41             end if;
42         end if;
43     end process clock_100ns_period;
44
45     logic: process(slow_clock)
46         variable counter : unsigned(3 downto 0);
47         variable down : unsigned(3 downto 0);
48         variable up : unsigned(3 downto 0);
49
50     begin
51         if RST = '1' then
52             counter := (others => '0');
```

```
53              down := (others => '0');
54              up := (others => '0');
55          else
56              if slow_clock'EVENT and slow_clock = '1' then
57                  if high_state = '1' then
58                      if up = 0 then
59                          high_state <= '0';
60                          up := unsigned(H);
61                          wave_out <= '0';
62                      else
63                          up := up - 1;
64                          wave_out <= '1';
65                      end if;
66                  else
67                      if down = 0 then
68                          high_state <= '1';
69                          down := unsigned(L);
70                          wave_out <= '1';
71                      else
72                          down := down - 1;
73                          wave_out <= '0';
74                      end if;
75                  end if;
76              end if;
77          end if;
78      end process logic;
79
80  end rtl;
```