File: W:/ESE382-Lab/Lab10/prelab10/prelab10/src/edge_det.vhd

```
1   ----------------------------------------------------------------------------
    -------
2   --
3   -- Title       : edge_det
4   -- Design      : prelab10
5   -- Author      : Dongyun Lee
6   -- Company     : Stony Brook University
7   --
8   ----------------------------------------------------------------------------
    -------
9   --
10  -- File        : W:\ESE382-Lab\Lab10\prelab10\prelab10\src\edge_det.vhd
11  -- Generated   : Sun Apr 21 16:52:52 2024
12  -- From        : interface description file
13  -- By          : Itf2Vhdl ver. 1.22
14  --
15  ----------------------------------------------------------------------------
    -------
16  --
17  -- Description : positive and negative edge detector
18  --
19  ----------------------------------------------------------------------------
    -------
20
21  library ieee;
22  use ieee.std_logic_1164.all;
23  use ieee.numeric_std.all;
24
25
26  entity edge_det is
27      port(
28          rst_bar   : in  std_logic; -- Asynchronous system reset
29          clk       : in  std_logic; -- System clock
30          sig       : in  std_logic; -- Input signal
31          pos       : in  std_logic; -- '1' for positive edge, '0' for
    negative
32          sig_edge  : out std_logic  -- High for one system clk after edge
    detected
33      );
34  end entity edge_det;
35
36
37  architecture moore_fsm of edge_det is
38  type state is (state_a, state_b, state_c);
39  signal present_state, next_state : state;
40  begin
41      -- first state : detects rst_bar
42      state_reg: process (clk, rst_bar)
43      begin
44          if rst_bar = '0' then
45              present_state <= state_a;
46          elsif rising_edge(clk) then
47              present_state <= next_state;
48          end if;
49      end process;
50
51      -- process where it outputs
52      outputs: process (present_state)
53      begin
54          case present_state is
55              when state_c => sig_edge <= '1';
```

```
56              when others => sig_edge <= '0';
57          end case;
58      end process;
59
60      nxt_state: process (present_state, sig)
61      begin
62          case present_state is
63              when state_a =>
64              if (pos = '1' and sig = '0') or (pos = '0' and sig = '1')
    then
65                  next_state <= state_b;
66              else
67                  next_state <= state_a;
68              end if;
69
70              when state_b =>
71              if (pos = '1' and sig = '1') or (pos = '0' and sig = '0')
    then
72                  next_state <= state_c;
73              else
74                  next_state <= state_b;
75              end if;
76
77              when others =>
78              if (pos = '1' and sig = '0') or (pos = '0' and sig = '1')
    then
79                  next_state <= state_b;
80              else
81                  next_state <= state_a;
82              end if;
83          end case;
84      end process;
85
86
87
88
89  end moore_fsm;
90
```