VHDL Moore FSMs

Agostini Federico

1 Code of `break_measure.vhd` architecture

```
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 break_measure is
14   generic(DONE_TIME : integer := 100);
15   Port (clk    : in  std_logic;
16         rst    : in  std_logic;
17         pulses_in  : in  std_logic;              -- Counter input
18         count_out  : out unsigned(5 downto 0); -- break time measurement in clock cycles
19         done_out   : out std_logic);             -- Calculation done
20 end break_measure;
21
22 architecture Behavioral of break_measure is
23
24 ------------------------------------------------------------
25 ----              WRITE YOUR CODE HERE                  ----
26 ------------------------------------------------------------
27 COMPONENT ila_1
28 PORT (
29         clk : IN STD_LOGIC;
30         probe0 : IN STD_LOGIC_VECTOR(0 DOWNTO 0); -- pulses_in
31         probe1 : IN STD_LOGIC_VECTOR(0 DOWNTO 0); -- pulses_p
32         probe2 : IN STD_LOGIC_VECTOR(5 DOWNTO 0); -- count
33         probe3 : IN STD_LOGIC_VECTOR(0 DOWNTO 0); -- done
34         probe4 : IN STD_LOGIC_VECTOR(2 DOWNTO 0)  -- state
35 );
36 END COMPONENT  ;
37
38 signal pulses_p : std_logic;
39 signal done     : std_logic;
40 signal count    : unsigned(5 downto 0);
41 type state is (s_idle, s_count, s_led); -- idle
42                                         -- [falling pulses]
43                                         -- counting
44                                         -- [rising pulses]
45                                         -- led
```

```
46  signal state_fsm : state;
47
48  begin
49
50  p_fsm : process(clk, rst, pulses_in) is
51  variable cnt : integer;
52  begin
53    if rst = '1' then
54      state_fsm <= s_idle;
55      cnt := 0;
56    elsif rising_edge(clk) then
57      pulses_p <= pulses_in;
58
59      case state_fsm is
60
61      when s_idle =>
62        if pulses_in = '0' and pulses_p = '1' then  -- simulate falling_edge(pulses_in)
63          state_fsm <= s_count;
64        else
65          state_fsm <= s_idle;
66        end if;
67        cnt := 0;
68        count <= to_unsigned(0, 6);
69        done <= '0';
70
71      when s_count =>
72        if pulses_in = '1' and pulses_p = '0' then  -- simulate rising_edge(pulses_in)
73          state_fsm <= s_led;
74        else
75          state_fsm <= s_count;
76        end if;
77        count <= count + 1;
78
79      when s_led =>
80        if cnt < DONE_TIME - 1 then
81          cnt := cnt + 1;
82          state_fsm <= s_led;
83        else
84          state_fsm <= s_idle;
85        end if;
86        done <= '1';
87
88      end case;
89
90    end if;
91
92  end process;
93
94  count_out <= count;
95  done_out  <= done;
96
97  ila : ila_1
98    PORT MAP (
99        clk => clk,
100       probe0(0) => pulses_in,
101       probe1(0) => pulses_p,
102       probe2    => std_logic_vector(count),
103       probe3(0) => done,
```

```
104         probe4      => std_logic_vector(to_unsigned(state'pos(state_fsm),3))
105     );
106
107 end Behavioral;
```

2 Screenshot of ILA

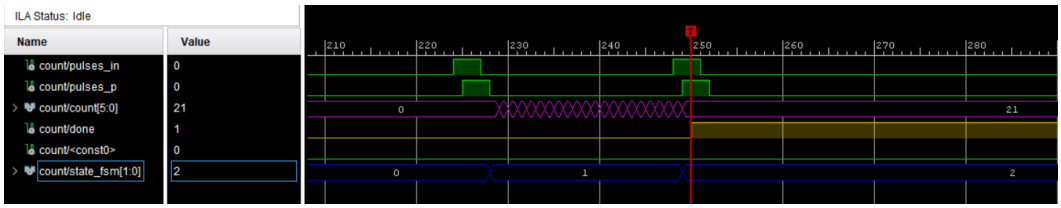

Figura 1: *Screenshot of ILA triggered in the rising edge of the* `done_out` *signal. The* `count` *and* `state_fsm` *signals are represented as* `Unsigned`