Lab7_VHDL_CODE

Tuesday, November 29, 2016 7:10 PM

```
library IEEE;
use IEEE.STD_LOGIC_1164. all;
use IEEE.STD_LOGIC_ARITH. all;
use IEEE.STD_LOGIC_UNSIGNED. all;

entity Divider is
Port (Dividend_in: in std_logic_vector(5 downto 0);
    Divisor: in std_logic_vector(2 downto 0);
    St, Clk, rst: in std_logic;
    Done: out std_logic;
    Quotient: out std_logic_vector(2 downto 0);
    Remainder: out std_logic_vector(2 downto 0);
    Overflow: out std_logic);
end Divider;

architecture Behavioral of Divider is
signal State, NextState: integer range 0 to 5;
signal C, Load, Su, Sh: std_logic;
signal Subout: std_logic_vector(3 downto 0);
signal Dividend: std_logic_vector(6 downto 0);
begin
    Subout <= Dividend(6 downto 3) - ('0' & divisor);
    C <= not Subout (3);
    Remainder <= Dividend(5 downto 3);
    Quotient <= Dividend(2 downto 0);

State_Graph: process (State, St, C)
    begin
        Load <= '0' ; Overflow <= '0'; Sh <= '0'; Su <= '0';
    case State is
        when 0 =>
            if (St = '1') then Load <= '1'; NextState <= 1; Done <= '0';
            else NextState <= 0; end if;
        when 1 =>
            if (C = '1') then Overflow <= '1'; NextState <= 0; Done <= '0';
            else Sh <= '1'; NextState <= 2; end if;
        when 2|3 =>
            if (C = '1') then Su <= '1'; NextState <= State; Done <= '0';
            else Sh <= '1'; NextState <= State + 1; end if;
        when 4 =>
            if (C = '1') then Su <= '1'; end if; Done <= '0';
            NextState <= State + 1;
        when 5 =>
            Done <= '1'; nextState <= 0;
        end case;
    end process State_Graph;


Update: process (Clk,rst)
```

```
    begin
    if rst = '1' then
        state <= 0;
    elsif Clk'event and Clk = '1' then -- rising edge of Clk
        State <= NextState;
        if Load = '1' then Dividend <= '0' & Dividend_in; end if;
        if Su = '1' then Dividend(6 downto 3) <= Subout; Dividend(0) <= '1'; end if;
        if Sh = '1' then Dividend <= Dividend(5 downto 0) & '0'; end if;
    end if;
end process update;
end Behavioral;
```