library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Stopwatch is
  Port ( start_stop : in STD_LOGIC;
         reset : in STD_LOGIC;
         clk_in : in STD_LOGIC;
         DISP0 : out STD_LOGIC_VECTOR (6 downto 0);
         AN : out STD_LOGIC_VECTOR (3 downto 0);
         dp : out std_logic);
end Stopwatch;

architecture Behavioral of Stopwatch is
component four_bit_counter port ( CEn, CLK : in STD_LOGIC;
                                  OUTPUT0,OUTPUT1,OUTPUT2,OUTPUT3 : out STD_LOGIC_VECTOR (3 downto 0));
end component;
component sev_seg_decoder port ( binary_num : in std_logic_vector(3 downto 0);
                                ABCDEFG : out std_logic_vector(6 downto 0));
end component;
component twobitcounter port ( clk : in STD_LOGIC;
                              count_out : out STD_LOGIC_vector(1 downto 0));
end component;
component four_to_one_mux port ( x0,x1,x2,x3 : in std_logic_vector(6 downto 0);
                                sr : in STD_LOGIC_VECTOR (1 downto 0);
                                f : out STD_LOGIC_vector(6 downto 0));
end component;
SIGNAL TEMP,TEMP2 : STD_LOGIC;
signal counter,counter2 : integer range 0 to 11000000 := 0;
signal s_0, s_1, s_2,s_3: std_logic_vector(6 downto 0) :="0000000";--signals to mux from decoders
signal SR : STD_LOGIC_VECTOR(1 DOWNTO 0); --connects 2bit counter to muxes
signal bin0,bin1,bin2,bin3 : std_logic_vector(3 downto 0); --counter outputs go to encoder inputs

begin

frequency_divider: process (clk_in) begin--250 Hz clock divider

if rising_edge(clk_in) then
  if (counter = 200000) then
    temp <= NOT(temp);
    counter <= 0;
  else
    counter <= counter + 1;
  end if;
end if;
end process;

frequency_divider2: process (clk_in) begin --clock divider for cycling through digits

if rising_edge(clk_in) then
  if (counter2 = 5000000) then
    temp2 <= NOT(temp2);
    counter2 <= 0;
  else
    counter2 <= counter2 + 1;
  end if;
end if;
end process;

COMB: PROCESS (SR) -- process to cycle through anodes
BEGIN
CASE SR IS
WHEN "00" =>
  AN<="1110"; dp<='1';
WHEN "01" =>
  AN<="1101"; dp<='0'; -- engage decimal point
WHEN "10" =>
  AN<="1011"; dp<='1';
WHEN "11" =>
  AN<="0111"; dp<='1';
when others => AN<="1111";
END CASE; END PROCESS COMB;

COUNTER2BIT : twobitcounter
  port map (clk => temp,
            count_out=>SR);
COUNTER4BIT0 : four_bit_countr
  port map (clk => temp2,
            OUTPUT0=>bin0,
            OUTPUT1=>bin1,
            OUTPUT2=>bin2,
            OUTPUT3=>bin3,
            CEn => start_stop,
            RST => RESET)
); decode0 : sev_seg_decoder
  port map ( binary_num => bin0,
            ABCDEFG => S_0);
decode1 : sev_seg_decoder
  port map ( binary_num => bin1,
            ABCDEFG => S_1);
decode2 : sev_seg_decoder
  port map ( binary_num => bin2,
            ABCDEFG => S_2);
decode3 : sev_seg_decoder
  port map ( binary_num => bin3,
            ABCDEFG => S_3);
mux : four_to_one_mux
port map ( x0 => S_0,
x1 => S_1,
x2 => s_2,
x3 => s_3,
f => DISP0,
sr => SR
);

end Behavioral;