Electrical and Computer Engineering
ECE-C302

Problem

Implement a running seven-segment display shown below

1. [ ] [ ] [ ] [ ]
2. [ ] [ ] [ ] [E]
3. [ ] [ ] [E] [C]
4. [ ] [E] [C] [E]
5. [E] [C] [E] [ ]

Repeat 1, 2, …

```
-- This solution uses a simple counter as a clock divider.
-- By accessing bits of the counter you can create different
-- synchronous scan rates to drive the display timing.
-- One rate to time division multiplex the characters to be displayed.
-- Another rate to time division multiplex the seven segment control
signals.
-- Display Rate = 50,000,000 * 2^-25 = 1.5 Hz
-- Scan Rate = 50,000,000 * 2^-16 = 763 Hz

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity midterm is
    port(clk   : in  std_logic;
         ssg   : out std_logic_vector(7 downto 0);
         an    : out std_logic_Vector(3 downto 0) );
end midterm;

architecture Behavioral of midterm is
    signal clk_div  : std_logic_vector(27 downto 0);
    signal seg0     : std_logic_vector(6 downto 0);
    signal seg1     : std_logic_vector(6 downto 0);
```

```
   signal seg2     : std_logic_vector(6 downto 0);
   signal seg3     : std_logic_vector(6 downto 0);
begin
-- -------------
-- Clock Divider
-- -------------
   process (clk)
   begin
      if clk'event and clk = '1' then
         if clk_div(27 downto 25) = "101" then
            clk_div <= (others => '0');     -- Explicit to control display
speed
         else
            clk_div <= clk_div + 1;        -- Implement clock divider as
a simple counter
         end if;
      end if;
   end process;

-- --------------------
-- Character Display Mux
-- --------------------
   seg3  <= "1111111" when clk_div(27 downto 25) = "000"   -- " "
      else "1111111" when clk_div(27 downto 25) = "001"   -- " "
      else "1111111" when clk_div(27 downto 25) = "010"   -- " "
      else "1111111" when clk_div(27 downto 25) = "011"   -- " "
      else "0000110" when clk_div(27 downto 25) = "100"   -- "E"
      else "1111111";                                     -- " "

   seg2  <= "1111111" when clk_div(27 downto 25) = "000"   -- " "
      else "1111111" when clk_div(27 downto 25) = "001"   -- " "
      else "1111111" when clk_div(27 downto 25) = "010"   -- " "
      else "0000110" when clk_div(27 downto 25) = "011"   -- "E"
      else "1000110" when clk_div(27 downto 25) = "100"   -- "C"
      else "1111111";                                     -- " "

   seg1  <= "1111111" when clk_div(27 downto 25) = "000"   -- " "
      else "1111111" when clk_div(27 downto 25) = "001"   -- " "
      else "0000110" when clk_div(27 downto 25) = "010"   -- "E"
      else "1000110" when clk_div(27 downto 25) = "011"   -- "C"
      else "0000110" when clk_div(27 downto 25) = "100"   -- "E"
      else "1111111";                                     -- " "

   seg0  <= "1111111" when clk_div(27 downto 25) = "000"   -- " "
```

```
      else "0000110" when clk_div(27 downto 25) = "001"   -- "E"
      else "1000110" when clk_div(27 downto 25) = "010"   -- "C"
      else "0000110" when clk_div(27 downto 25) = "011"   -- "E"
      else "1111111" when clk_div(27 downto 25) = "100"   -- " "
      else "1111111";                                     -- " "

-- -----------------
-- Seven Segment Mux
-- -----------------
   ssg(7) <= '1';
   ssg(6 downto 0) <=   seg0 when clk_div(17 downto 16) = "00"
               else seg1 when clk_div(17 downto 16) = "01"
               else seg2 when clk_div(17 downto 16) = "10"
               else seg3;

-- -----------------
-- Anode Control Mux
-- -----------------
   an <=    "1110" when clk_div(17 downto 16) = "00"
      else "1101" when clk_div(17 downto 16) = "01"
      else "1011" when clk_div(17 downto 16) = "10"
      else "0111";

-- Since you may not be totally familiar with this type of assignment
-- statement, here is an equivalent process statement.  Either syntax
-- will generate a mux.
--
--    process (clk_div)
--    begin
--       if clk_div(17 downto 16) = "00" then
--          an <= "1110";
--       elsif clk_div(17 downto 16) = "01" then
--          an <= "1101";
--       elsif clk_div(17 downto 16) = "10" then
--          an <= "1011";
--       else
--          an <= "0111";
--       end if;
--    end process;

end Behavioral;
```