EMMAN ACE G MENION
BSCPE 3A

20. VHDL CODE FOR SEVEN SEGMENT DISPLAY INTERFACE

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity practice is
   Port (
      clk : in STD_LOGIC;    -- 4 MHz input clock
      rst : in STD_LOGIC;    -- Active-high reset
      seg : out STD_LOGIC_VECTOR(6 downto 0);  -- Segments a-g (active low)
      dig : out STD_LOGIC_VECTOR(3 downto 0)   -- Digit enables (active low)
   );
end practice;

architecture Behavioral of practice is

   -- Constants
   constant MAX_COUNT : unsigned(23 downto 0) := x"F42400";  -- ≈ 4s for 4MHz

   -- Signals
   signal clk_div    : unsigned(23 downto 0) := (others => '0');
   signal hex_value  : unsigned(3 downto 0) := (others => '0');
   signal scan_count : unsigned(1 downto 0) := (others => '0');

   -- Digit enable lookup
   type DIG_ARRAY is array(0 to 3) of STD_LOGIC_VECTOR(3 downto 0);
   constant DIG_EN : DIG_ARRAY := (
      "1110",  -- Enable digit 1
      "1101",  -- Enable digit 2
      "1011",  -- Enable digit 3
      "0111"   -- Enable digit 4
   );

begin

   -- Clock divider and hex counter update (every ~4 seconds)
   process(clk, rst)
   begin
      if rst = '1' then
         clk_div <= (others => '0');
         hex_value <= (others => '0');
      elsif rising_edge(clk) then
         if clk_div = MAX_COUNT then
            clk_div <= (others => '0');
            hex_value <= hex_value + 1;
         else
            clk_div <= clk_div + 1;
         end if;
      end if;
   end process;
```

```
-- Digit scan selector (2 MSBs of clk_div)
scan_count <= clk_div(15 downto 14);
dig <= DIG_EN(to_integer(scan_count));

-- 7-segment display decoder (common cathode, active-low)
with hex_value select
   seg <= "0000001" when x"0",  -- 0
          "1001111" when x"1",  -- 1
          "0010010" when x"2",  -- 2
          "0000110" when x"3",  -- 3
          "1001100" when x"4",  -- 4
          "0100100" when x"5",  -- 5
          "0100000" when x"6",  -- 6
          "0001111" when x"7",  -- 7
          "0000000" when x"8",  -- 8
          "0000100" when x"9",  -- 9
          "0001000" when x"A",  -- A
          "1100000" when x"B",  -- b
          "0110001" when x"C",  -- C
          "1000010" when x"D",  -- d
          "0110000" when x"E",  -- E
          "0111000" when others; -- F

end Behavioral;
```