

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

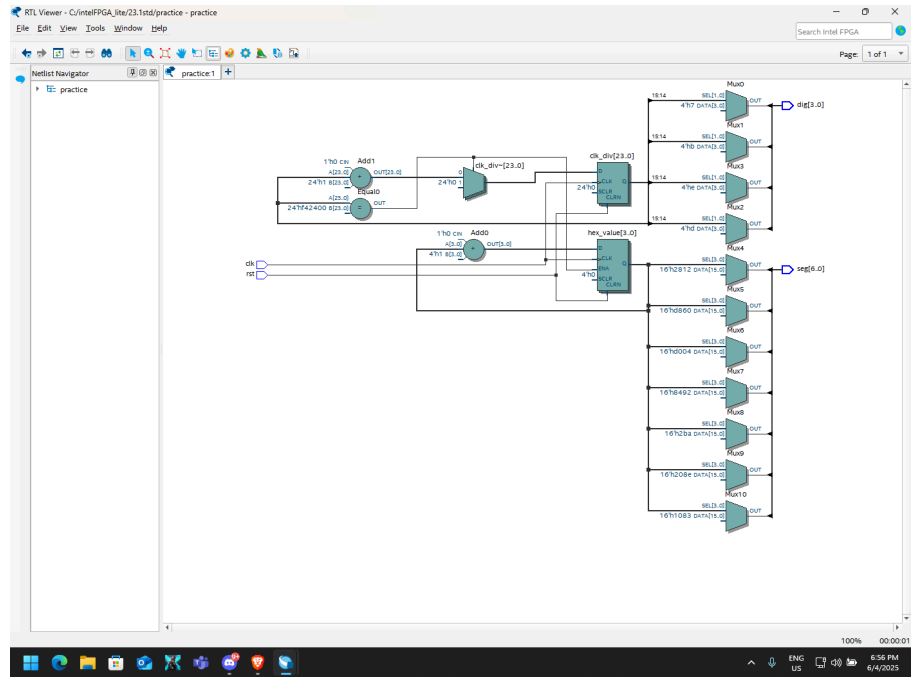
  -- Clock divider and hex counter update (every ~4 seconds)
  process(clk, rst)
  begin
    if rst = '1' then
      clk_div <= (others => '0');
      hex_value <= (others => '0');
    elsif rising_edge(clk) then
      if clk_div = MAX_COUNT then
        clk_div <= (others => '0');
        hex_value <=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



Quartus Prime Lite edition - C:/intelFPGA_lite/23.1std/practice - practice

File Edit View Project Assignments Processing Tools Window Help

Search Intel FPGA

Project Navigator

Entity/Instance

Cyclone IV E: EP4CE6E22C8

practice

Tasks

Compilation

Task

Compile Design 00%

Analysis & Synthesis 00%

Fitter (Place & Route) 00%

Assembler (Generate programming files) 00%

Timing Analysis 00%

EDA Netlist Writer 00%

Edit Settings

Program Device (Open Programmer)

Compilation Report - practice

```
10 dig : out STD_LOGIC_VECTOR(3 downto 0) -- Digit enables (active low)
11 );
12 end practice;
13
14 architecture Behavioral of practice is
15
16 -- Constants
17 constant MAX_COUNT : unsigned(23 downto 0) := x"F42400"; -- ≈ 4s for 4MHz
18
19 -- Signals
20 signal clk_div : unsigned(23 downto 0) := (others => '0');
21 signal hex_value : unsigned(3 downto 0) := (others => '0');
22 signal scan_count : unsigned(1 downto 0) := (others => '0');
23
24 -- Digit enable lookup
25 type DIG_ARRAY is array(0 to 3) of STD_LOGIC_VECTOR(3 downto 0);
26 constant DIG_EN : DIG_ARRAY := (
27   "1110", -- Enable digit 1
28   "1101", -- Enable digit 2
29   "1011", -- Enable digit 3
30   "0111" -- Enable digit 4
31 );
32
33 begin
34
35 -- Clock divider and hex counter update (every ~4 seconds)
36 process(clk, rst)
37 begin
38   if rst = '1' then
39     clk_div <= (others => '0');
40     hex_value <= (others => '0');
41   elsif rising_edge(clk) then
42     if clk_div = MAX_COUNT then
43       clk_div <= (others => '0');
44       hex_value <= hex_value + 1;
45     else
46       clk_div <= clk_div + 1;
47     end if;
48   end if;
49 end process;
```

IP Catalog

Installed IP

Project Directory

No Selection Available

Library

Basic Functions

DSP

Interface Protocols

Memory Interfaces and Controllers

Processors and Peripherals

University Program

Search for Partner IP

Find...

Find Next

Type ID Message

Quartus Prime EDA Netlist writer was successful. 0 errors, 1 warning

293000 Quartus Prime Full compilation was successful. 0 errors, 16 warnings

Running Quartus Prime Netlist Viewers Preprocess

Command: quartus_npp practice -c practice --netlist_type=sgate

18236 Number of processors has not been specified which may cause overloading on shared machines. Set the global assignment NUM_PARALLEL_PROCESSORS in your (