library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

---- Uncomment the following library declaration if instantiating

---- any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity lcd\_V is

generic(rst\_state:std\_logic:='1'); -- for USDP it is Active LOW

Port ( rst : in std\_logic; -- reset

clk\_12Mhz : in std\_logic; -- high freq. clock

lcd\_rs : out std\_logic; -- LCD RS control

lcd\_en : out std\_logic; -- LCD Enable

lcd\_data : out std\_logic\_vector(7 downto 0)); -- LCD Data port

end lcd\_V;

architecture Behavioral of lcd\_V is

signal div : std\_logic\_vector(20 downto 0); --- delay timer 1

signal clk\_t1,lcd\_rs\_s,lcd\_en\_s ,line2,line2\_clk1: std\_logic;

-- LCD controller FSM states

type state is (reset,func,addr\_1st,addr\_2nd,mode,cur,clear,d0,d1,d2,d3,d4,d5,d6,d7,d8,d9,d10,d11,

d12,d13,d14,d15,hold,extra);

signal ps,nx : state;

signal dataout\_s : std\_logic\_vector(7 downto 0); --- internal data command multiplexer

begin

lcd\_en <= lcd\_en\_s;

lcd\_rs <= lcd\_rs\_s;

----- clk divider ---------------------------------

process(rst,clk\_12Mhz)

begin

if(rst = rst\_state)then

div <= (others=>'0');

elsif( clk\_12Mhz'event and clk\_12Mhz ='1')then

div <= div + 1;

end if;

end process;

----------------------------------------------------

clk\_t1 <= div(15);

----- Presetn state Register -----------------------

process(rst,clk\_t1)

begin

if(rst = rst\_state)then

ps <= reset;

line2 <= '0'; ---to display "Smart Logic Tech"

elsif( clk\_t1'event and clk\_t1 ='1')then

ps <= nx;

if ps = extra then

line2 <= '1'; -- to display "lcd demo in vhdl"

end if;

end if;

end process;

----- state and output decoding process

process(rst,ps,line2)

begin

case(ps) is

when reset =>

nx <= func;

lcd\_rs\_s <= '0';

dataout\_s <= "00111000"; -- 38h

when func =>

nx <= mode;

lcd\_rs\_s <= '0';

dataout\_s <= "00111000"; -- 38h

when mode =>

nx <= cur;

lcd\_rs\_s <= '0';

dataout\_s <= "00000110"; -- 06h

when cur =>

nx <= addr\_1st;

lcd\_rs\_s <= '0';

dataout\_s <= "00001100"; -- 0Ch curser at starting point of line1

when addr\_1st => -- line-1 display adress

nx <= clear;

lcd\_rs\_s <= '0';

dataout\_s <= "10110000"; -- 80h

when clear=>

nx <= d0;

lcd\_rs\_s <= '0';

dataout\_s <= "00000001"; -- 01h

when d0 =>

lcd\_rs\_s <= '1';

dataout\_s <= "01010111"; -- W

nx <= d1;

when d1 =>

lcd\_rs\_s <= '1';

dataout\_s <= "01100101"; -- e

nx <= d2;

when d2 =>

lcd\_rs\_s <= '1';

dataout\_s <= "01101100"; -- l

nx <= d3;

when d3 =>

lcd\_rs\_s <= '1';

dataout\_s <= "01100011"; -- c

nx <= d4;

-- LCD Demo in VHDL

when d4 =>

lcd\_rs\_s <= '1';

dataout\_s <= "01101111"; -- o

nx <= d5;

when d5 =>

lcd\_rs\_s <= '1';

dataout\_s <= "01101101"; -- m

nx <= d6;

when d6 =>

lcd\_rs\_s <= '1';

dataout\_s <= "01100101"; -- e

nx <= d7;

when d7 =>

lcd\_rs\_s <= '1';

dataout\_s <= "00100000"; -- space

nx <= d8;

when d8 =>

lcd\_rs\_s <= '1';

dataout\_s <= "01110100"; -- t

nx <= d9;

when d9 =>

lcd\_rs\_s <= '1';

dataout\_s <= "01101111"; -- o

nx <= d10;

when d10 =>

lcd\_rs\_s <= '1';

dataout\_s <= "00100000"; -- space

nx <= d11;

when d11 =>

lcd\_rs\_s <= '1';

dataout\_s <= "01001001"; -- I

nx<=d12;

when d12 =>

lcd\_rs\_s <= '1';

dataout\_s <= "00110010"; -- 2

nx <= d13;

when d13 =>

lcd\_rs\_s <= '1';

dataout\_s <= "01001001"; -- I

nx <= d14;

when d14 =>

lcd\_rs\_s <= '1';

dataout\_s <= "01010100"; -- T

nx <= d15;

when d15 =>

lcd\_rs\_s <= '1';

dataout\_s <= "00100000"; -- space

nx <= addr\_2nd;

when addr\_2nd => --- line-2 display adress

lcd\_rs\_s <= '0';

if line2='1' then

nx <= addr\_2nd;

else

nx <= extra;

dataout\_s <= "11000000"; -- C0h -- curser at starting point of line2

end if;

when extra =>

nx <= d0;

lcd\_rs\_s <= '0';

dataout\_s <= "11000000"; -- C0h

when others=>

nx <= reset;

lcd\_rs\_s <= '0';

dataout\_s <= "00000001"; -- CLEAR

end case;

end process;

lcd\_en\_s <= clk\_t1;

lcd\_data <= dataout\_s;

end Behavioral;