**LCD INTERFACING**

library IEEE; use IEEE.STD\_LOGIC\_1164.ALL; use IEEE.STD\_LOGIC\_ARITH.ALL; use IEEE.STD\_LOGIC\_UNSIGNED.ALL; entity lcd is

Port ( clk,reset : in std\_logic; RS,EN,RW : out std\_logic;

data : out std\_logic\_vector(7 downto 0)); end lcd;

architecture Behavioral of lcd is

type state\_type is (s0,s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11, s12,s13,s14,s15,s16,s17,s18,s19,s20,s21,s22,s23); signal state:state\_type;

SIGNAL count:std\_logic\_vector(22 downto 0); SIGNAL clk1:std\_logic;

begin process(Clk,Reset)

begin

if(Clk' event AND Clk='1')then count<=count+"0001"; end if;

clk1<=count(20);

end process; RW<='0';

process(clk1, reset) begin

if reset = '1' then

state <= s0;

elsif rising\_edge(clk1) then

if state = s0 then state <= s1;

RS<='0'; -- Write commonds to LCD. EN <= '1';

data <= "00110000"; -- Function set for 8 bit interface, 1 line mode and 5x7 dot matrix.

end if;

if state = s1 then state <= s2; EN <= '0';

end if;

if state = s2 then state <= s3;

E43007 Sujit Dhage

EN <= '1';

data <= "00001111"; -- Display cursor and blinking ON. end if;

if state = s3 then state <= s4; EN <= '0';

end if;

if state = s4 then state <= s5; EN <= '1';

data <= "00000001"; -- Clear display.

end if;

if state = s5 then state <= s6; EN <= '0';

end if;

if state = s6 then state <= s7; EN <= '1';

data <= "10000100"; -- Display address. end if;

if state = s7 then state <= s8; EN <= '0';

end if;

if state = s8 then

RS <= '1'; -- Write data to LCD. state <= s9;

EN <= '1';

data <= "00101010"; --(\*) end if;

if state = s9 then state <= s10; EN <= '0';

end if; if state = s10 then

state <= s11; EN <= '1';

data <= "01010011"; --S end if;

if state = s11 then state <= s12; EN <= '0';

end if;

if state = s12 then state <= s13; EN <= '1';

data <= "01001011"; --K

end if;

if state = s13 then state <= s14; EN <= '0';

end if;

if state = s14 then state <= s15; EN <= '1';

data <= "01001110"; --N end if;

if state = s15 then state <= s16; EN <= '0';

end if;

if state = s16 then state <= s17; EN <= '1';

data <= "01000011"; --C end if;

if state = s17 then state <= s18; EN <= '0';

end if;

if state = s18 then state <= s19; EN <= '1';

data <= "01001111"; --O end if;

if state = s19 then state <= s20; EN <= '0';

end if;

if state = s20 then state <= s21; EN <= '1';

data <= "01000101"; --E end if;

if state = s21 then state <= s22; EN <= '0';

end if;

if state = s22 then state <= s23; EN <= '1';

data <= "00101010"; --(\*) end if;

if state = s23 then EN <= '0';

end if; end if;

end process; end Behavioral;

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

**Testbench:**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

-- Uncomment the following library declaration if using -- arithmetic functions with Signed or Unsigned values --USE ieee.numeric\_std.ALL;

ENTITY lcdtest IS END lcdtest;

ARCHITECTURE behavior OF lcdtest IS

-- Component Declaration for the Unit Under Test (UUT) COMPONENT lcd

PORT(

clk1 : IN std\_logic; reset : IN std\_logic; RS : OUT std\_logic; EN : OUT std\_logic; RW : OUT std\_logic;

data : OUT std\_logic\_vector(7 downto 0) );

END COMPONENT;

--Inputs

signal clk1 : std\_logic := '0'; signal reset : std\_logic := '0';

--Outputs signal RS : std\_logic; signal EN : std\_logic; signal RW : std\_logic;

signal data : std\_logic\_vector(7 downto 0); -- Clock period definitions

constant clk1\_period : time := 10 ns; BEGIN

-- Instantiate the Unit Under Test (UUT) uut: lcd PORT MAP (

clk1 => clk1, reset => reset, RS => RS,

EN => EN, RW => RW,

data => data );

-- Clock process definitions clk1\_process :process begin

clk1 <= '0';

wait for clk1\_period/2; clk1 <= '1';

wait for clk1\_period/2; end process;

-- Stimulus process stim\_proc: process begin

-- hold reset state for 100 ns.

reset<='0'; wait for clk1\_period\*10;

-- insert stimulus here wait;

end process; END;

A screenshot of a computer

Description automatically generated

**UCF:**

NET data(0) LOC =P62;

NET data(1) LOC =P63;

NET data(2) LOC =P64;

NET data(3) LOC =P65;

NET data(4) LOC =P67;

NET data(5) LOC =P68;

NET data(6) LOC =P71;

NET data(7) LOC =P72;

NET Clk LOC =P183;

NET reset LOC =P102;

NET RS LOC =P57;

NET EN LOC =P61;

NET RW LOC =P58;

=========================================================================

\* Final Report \*

=========================================================================

Final Results

RTL Top Level Output File Name : lcd.ngr

Top Level Output File Name : lcd

Output Format : NGC

Optimization Goal : Speed

Keep Hierarchy : No

Design Statistics

# IOs : 13

Cell Usage :

# BELS : 92

# GND : 1

# INV : 2

# LUT1 : 20

# LUT2 : 1

# LUT2\_L : 3

# LUT3 : 3

# LUT3\_D : 1

# LUT4 : 14

# LUT4\_D : 1

# LUT4\_L : 4

# MUXCY : 20

# VCC : 1

# XORCY : 21

# FlipFlops/Latches : 55

# FD : 21

# FDC : 22

# FDCE : 1

# FDE : 10

# FDP : 1

# Clock Buffers : 2

# BUFG : 1

# BUFGP : 1

# IO Buffers : 12

# IBUF : 1

# OBUF : 11

=========================================================================

Total REAL time to Xst completion: 4.00 secs Total CPU time to Xst completion: 3.89 secs -->

Total memory usage is 4509380 kilobytes Number of errors : 0 ( 0 filtered) Number of warnings : 3 ( 0 filtered) Number of infos : 0 ( 0 filtered)