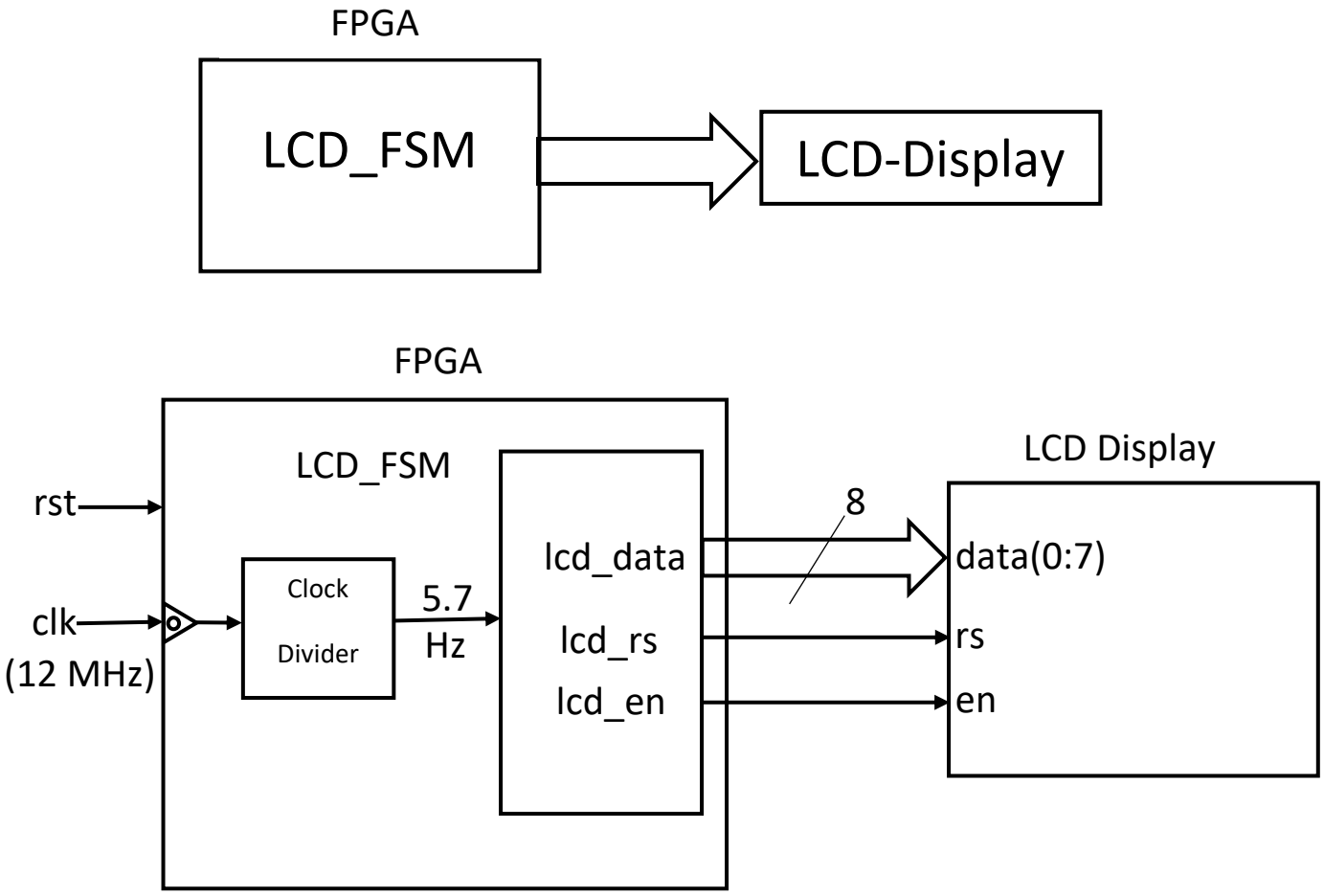
TITLE PAGE

Class	:	BE - 8
Roll. No	:	42428
Assignment No.	:	A. 4
Assignment Name	:	LCD-FPGA Interfacing
Date Of Performance	:	17-10-2020

Block Diagram:



Pin Assignment (UCF Location) for LCD :

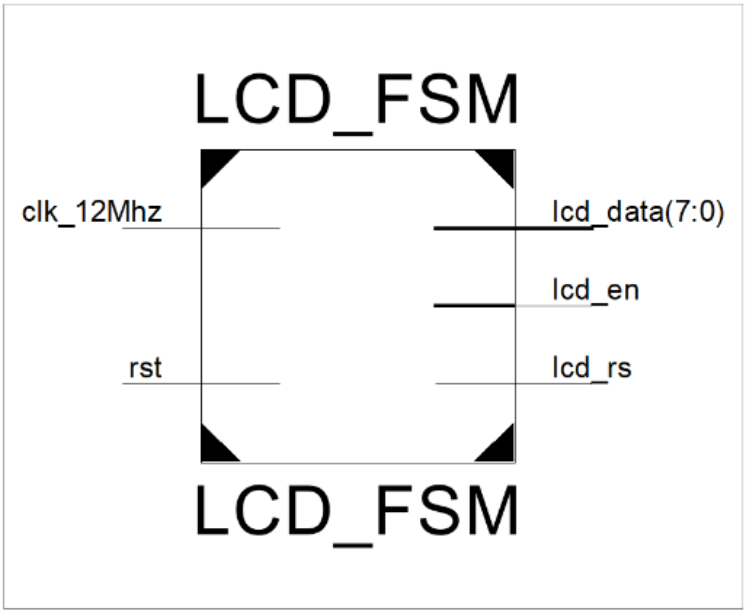
Signal Name	XC3S250E-PQ208	XCS6LX9-TQG144
LCD_RS	P48	P74
LCD_EN	P49	P72
LCD_Data0	P47	NC
LCD_Data1	P41	NC
LCD_Data2	P39	NC
LCD_Data3	P35	NC
LCD_Data4	P33	P66
LCD_Data5	P31	P61
LCD_Data6	P29	P59
LCD_Data7	P24	P58

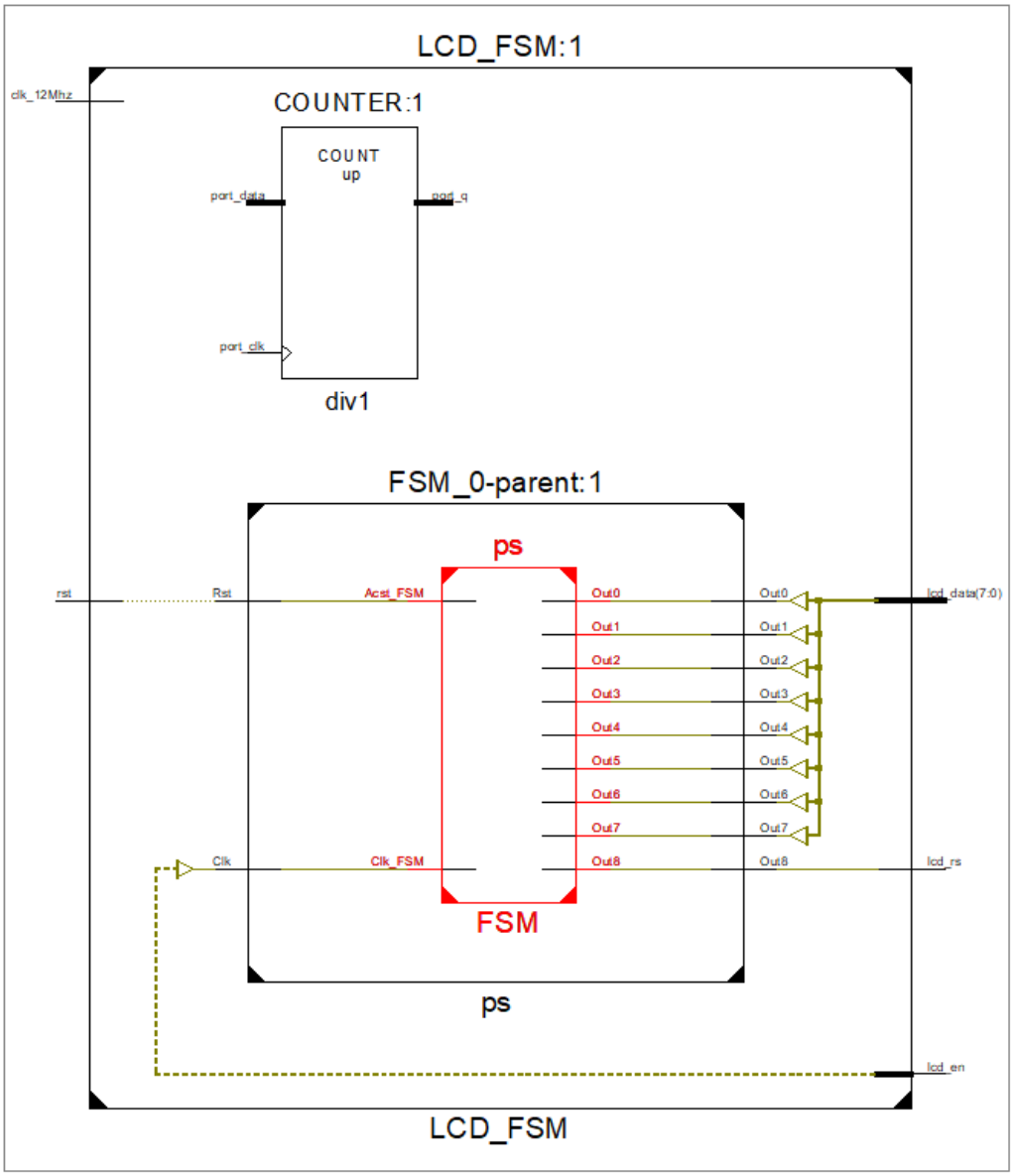
MAIN VHDL PROGRAM

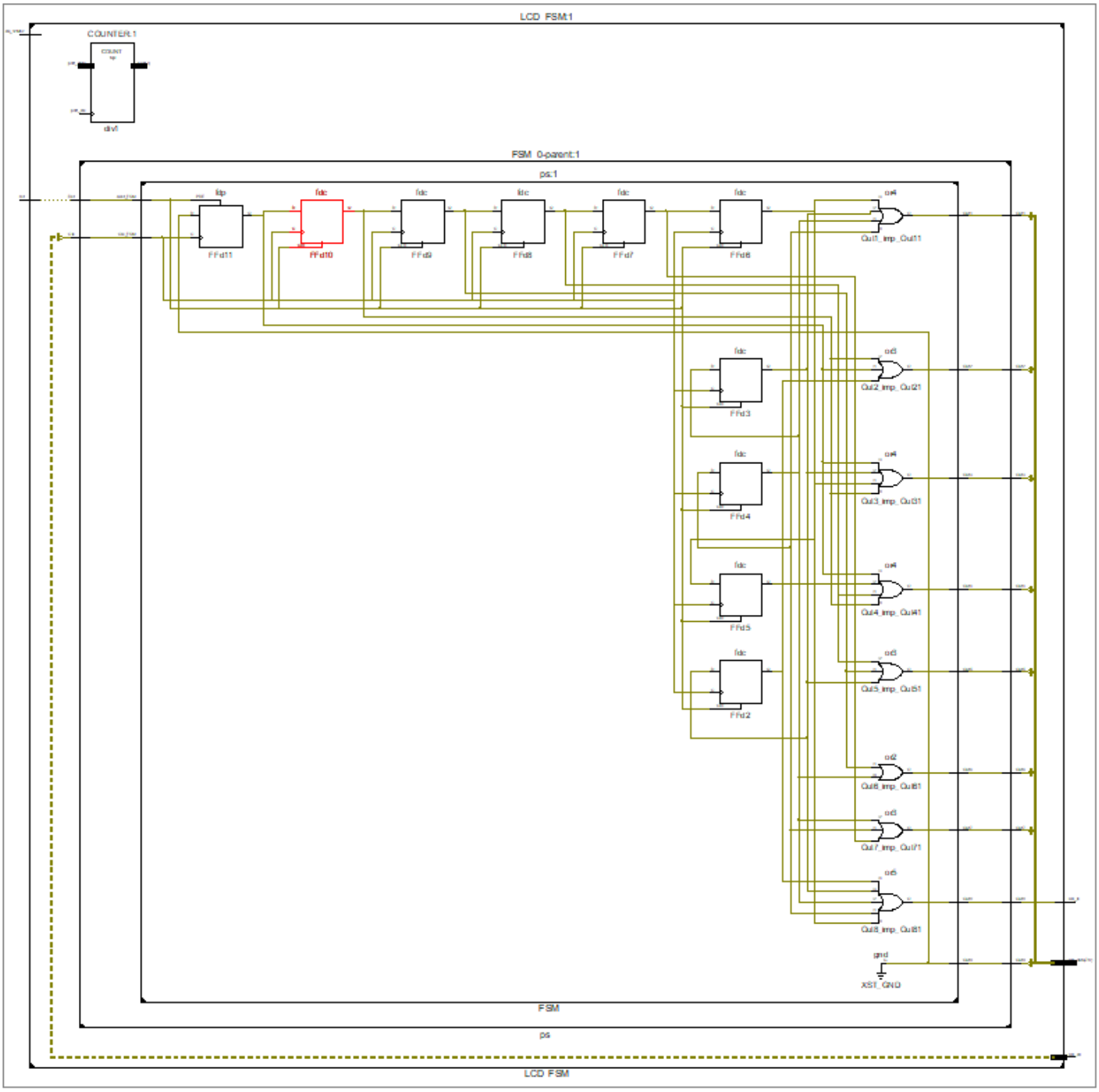
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity LCD_FSM is
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_FSM;

architecture Behavioral of LCD_FSM is
signal div : std_logic_vector(20 downto 0); --- delay timer 1
signal clk_fsm,lcd_rs_s: std_logic;                                  -- LCD controller FSM states
type state is (reset,func,mode,cur,clear,d0,d1,d2,d3,d4,hold);
signal ps,nx       : state;
signal dataout_s  : std_logic_vector(7 downto 0);                    --- internal data command multiplexer
begin

----- clk divider ---------------------------------
process(rst,clk_12Mhz)
begin
          if(rst = '1')then
                    div <= (others=>'0');
          elsif( clk_12Mhz'event and clk_12Mhz ='1')then
                    div <= div + 1;
          end if;
end process;
-------------------------------------------------
clk_fsm <= div(20);

----- Presetn state Register ----------------------
process(rst,clk_fsm)
begin
          if(rst = '1')then
                    ps  <= reset;
          elsif(clk_fsm'event and clk_fsm ='1')then
                    ps <= nx;
          end if;
end process;

----- state and output  decoding process
process(ps)
begin
          case(ps) is
                    when reset =>
                              nx <= func;
                              lcd_rs_s <= '0';
                              dataout_s <= "00111000";                -- 38h
```

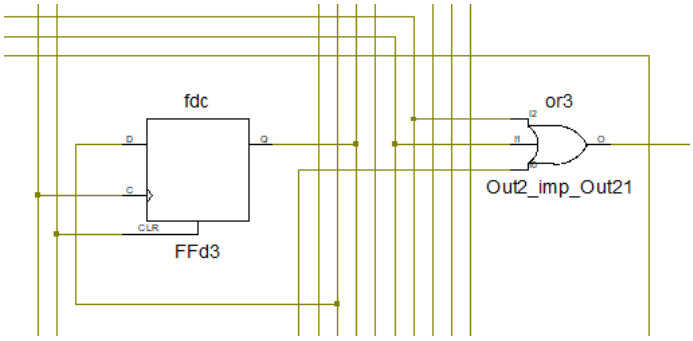
```
                when func =>
                        nx <= mode;
                        lcd_rs_s <= '0';
                        dataout_s <= "00111000";            -- 38h
                when mode =>
                        nx <= cur;
                        lcd_rs_s <= '0';
                        dataout_s <= "00000110";            -- 06h
                when cur =>
                        nx <= clear;
                        lcd_rs_s <= '0';
                        dataout_s <= "00001100";            -- 0Ch  curser at starting point of line1
                when clear =>
                        nx <= d0;
                        lcd_rs_s <= '0';
                        dataout_s <= "00000001";            -- 01h
                when d0 =>
                        lcd_rs_s <= '1';
                        dataout_s <= "01010000";            -- P
                        nx <= d1;
                when d1 =>
                        lcd_rs_s <= '1';
                        dataout_s <= "01001001";            -- I
                        nx <= d2;
                when d2 =>
                        lcd_rs_s <= '1';
                        dataout_s <= "01000011";            -- C
                        nx <= d3;
                when d3 =>
                        lcd_rs_s <= '1';
                        dataout_s <= "01010100";            -- T
                        nx <= d4;
                when d4 =>
                        lcd_rs_s <= '1';
                        dataout_s <= "00100000";            -- space
                        nx <= hold;
                when hold =>
                        lcd_rs_s <= '0';
                        dataout_s <= "00000000";            -- hold
                        nx <= hold;
                when others =>
                        nx <= reset;
                        lcd_rs_s <= '0';
                        dataout_s <= "00000001";            -- CLEAR
        end case;
end process;

lcd_en <= clk_fsm;
lcd_rs <= lcd_rs_s;
lcd_data <= dataout_s;

end Behavioral;
```

RTL SCHEMATIC

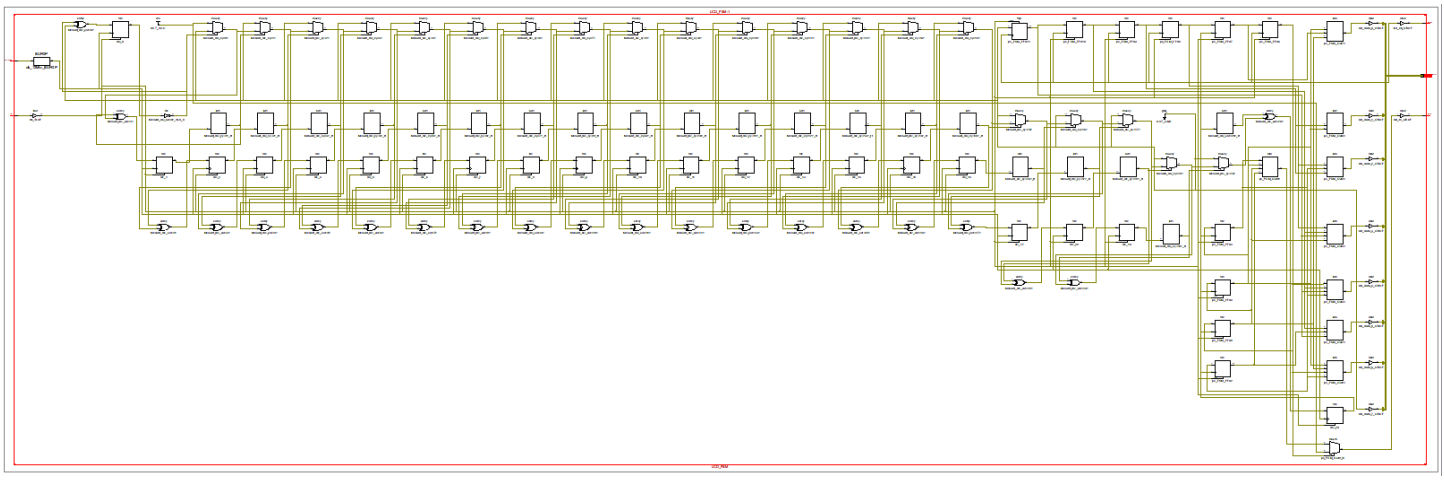


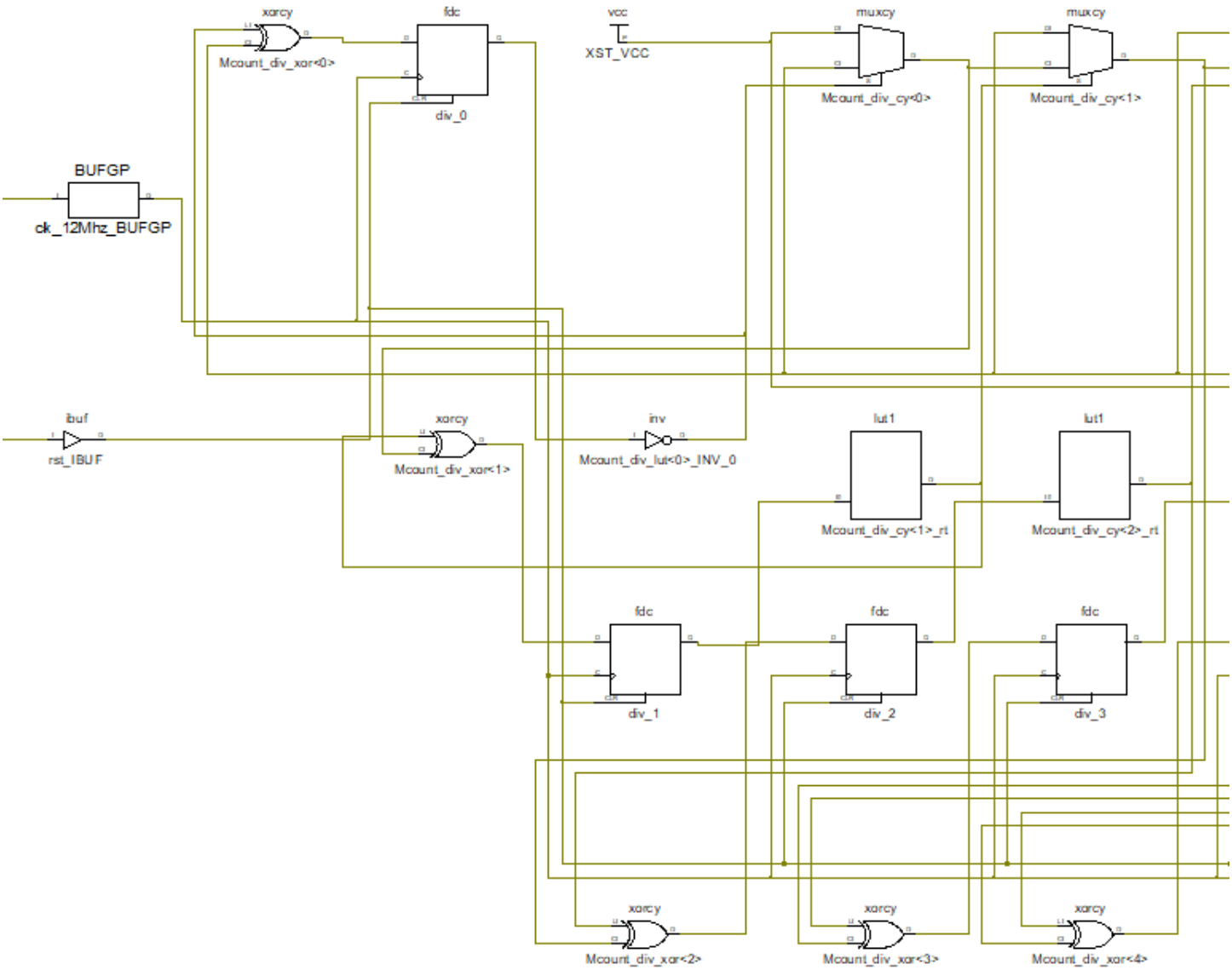



LCD_FSM:1
COUNTER:1
div1
FSM_0-parent:1
ps:1
ldp
fdc
FFd11
FFd10
FFd9
FFd8
FFd7
FFd6
FFd3
FFd4
FFd5
FFd2
or4
Out1_imp_Out11
or3
Out2_imp_Out21
or4
Out3_imp_Out31
or4
Out4_imp_Out41
or3
Out5_imp_Out51
or2
Out6_imp_Out61
or3
Out7_imp_Out71
or5
Out8_imp_Out81
gnd
XST_GND
FSM
ps
LCD_FSM


fdc
D
Q
C
CLR
FFd3
or3
I2
I1
O
Out2_imp_Out21

TECHNOLOGY SCHEMATIC





SYNTHESIS REPORT

1) Device Utilisation Summary

```
=========================================================================
*                      Final Report                     *
=========================================================================
Final Results
RTL Top Level Output File Name     : LCD_FSM.ngr
Top Level Output File Name         : LCD_FSM
Output Format                      : NGC
Optimization Goal                  : Speed
Keep Hierarchy                     : No

Design Statistics
# IOs                              : 12

Cell Usage :
# BELS                             : 73
#      GND                         : 1
#      INV                         : 1
#      LUT1                        : 20
#      LUT2                        : 1
#      LUT3                        : 3
#      LUT4                        : 4
#      MUXCY                       : 20
#      MUXF5                       : 1
#      VCC                         : 1
#      XORCY                       : 21
# FlipFlops/Latches                : 31
#      FDC                         : 30
#      FDP                         : 1
# Clock Buffers                    : 1
#      BUFGP                       : 1
# IO Buffers                       : 11
#      IBUF                        : 1
#      OBUF                        : 10
=========================================================================
Device utilization summary:
---------------------------
Selected Device : 3s250epq208-5

 Number of Slices:              18  out of   2448     0%
 Number of Slice Flip Flops:    31  out of   4896     0%
 Number of 4 input LUTs:        29  out of   4896     0%
 Number of IOs:                 12
 Number of bonded IOBs:         12  out of    158     7%
 Number of GCLKs:               1  out of     24     4%
---------------------------
Partition Resource Summary:
---------------------------
  No Partitions were found in this design.
---------------------------
```

2) TIMING REPORT

Timing Summary:

Speed Grade: -5

Minimum period: 3.933ns (Maximum Frequency: 254.259MHz)
Minimum input arrival time before clock: No path found
Maximum output required time after clock: 5.537ns
Maximum combinational path delay: No path found

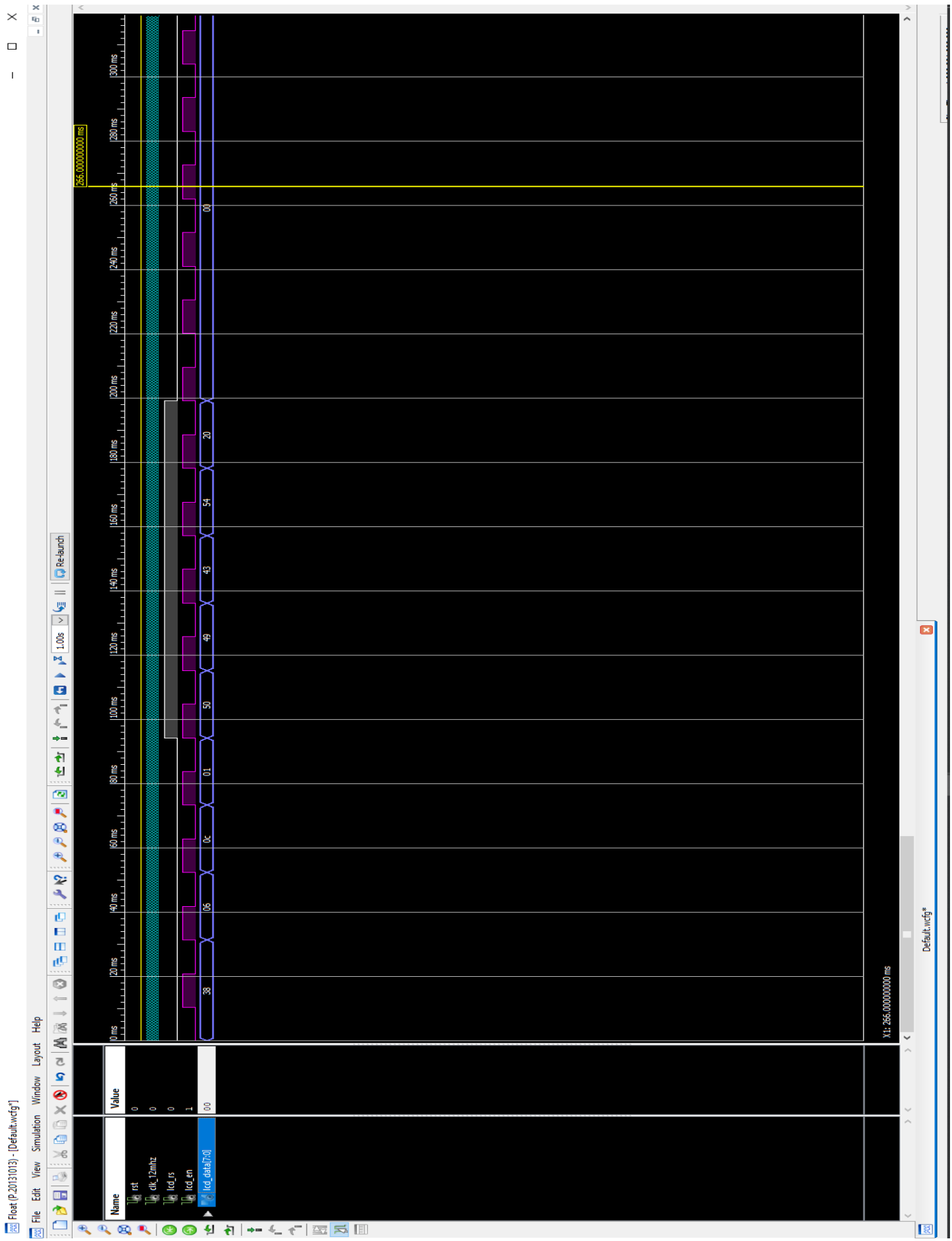
Timing Detail:

All values displayed in nanoseconds (ns)

TESTBENCH PROGRAM

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY LCD_Test IS
END LCD_Test;

ARCHITECTURE behavior OF LCD_Test IS
   COMPONENT LCD_FSM
   PORT(
        rst : IN  std_logic;
        clk_12Mhz : IN  std_logic;
        lcd_rs : OUT  std_logic;
        lcd_en : OUT  std_logic;
        lcd_data : OUT  std_logic_vector(7 downto 0)
       );
   END COMPONENT;

  signal rst : std_logic := '0';
  signal clk_12Mhz : std_logic := '0';

  signal lcd_rs : std_logic;
  signal lcd_en : std_logic;
  signal lcd_data : std_logic_vector(7 downto 0);

  constant clk_12Mhz_period : time := 10 ns;

BEGIN
  uut: LCD_FSM PORT MAP (
        rst => rst,
        clk_12Mhz => clk_12Mhz,
        lcd_rs => lcd_rs,
        lcd_en => lcd_en,
        lcd_data => lcd_data
       );

  clk_12Mhz_process :process
  begin
           clk_12Mhz <= '0';
           wait for clk_12Mhz_period/2;
           clk_12Mhz <= '1';
           wait for clk_12Mhz_period/2;
  end process;

  stim_proc: process
  begin
    rst <= '1';
    wait for 20 ns;
    rst <= '0';
    wait;
  end process;
END;
```

ISIM WAVEFORMS



PIN-LOCKING REPORT

PlanAhead Generated physical constraints

NET "clk_12Mhz" LOC = P80;

NET "rst" LOC = P204;

NET "lcd_rs" LOC = P48;

NET "lcd_en" LOC = P49;

NET "lcd_data[0]" LOC = P47;

NET "lcd_data[1]" LOC = P41;

NET "lcd_data[2]" LOC = P39;

NET "lcd_data[3]" LOC = P35;

NET "lcd_data[4]" LOC = P33;

NET "lcd_data[5]" LOC = P31;

NET "lcd_data[6]" LOC = P29;

NET "lcd_data[7]" LOC = P24;

Conclusion:

Thus we have :

1) Modeled LCD-FPGA Interfacing using Behavioral Modeling Style.
2) Observed following Schematics: **RTL** & **Technology Schematics** generated **Post-Synthesis**.
3) Interpreted **Device Utilization Summary** in terms of LUTs, SLICES, IOBs, Multiplexers & D FFs used out of the available device resources.
4) Interpreted the TIMING Report in terms of Maximum combinational delay as indicative of the Maximum Operating Frequency.
5) Written a TESTBENCH to verify the functionality of LCD- FPGA Interfacing & verified the functionality as per the TRUTH-TABLE by observing ISIM Waveforms.
6) Used PlanAhead Editor for pin-locking.
7) Prototyped the FPGA **XC3S250EPQ208-5** to realize LCD- FPGA Interfacing & verified its operation by giving suitable input combinations.