復旦大學

FDE LAB REPORT

Sun Zhongji

CONTENTS

01

- Development Environment

Development Environment

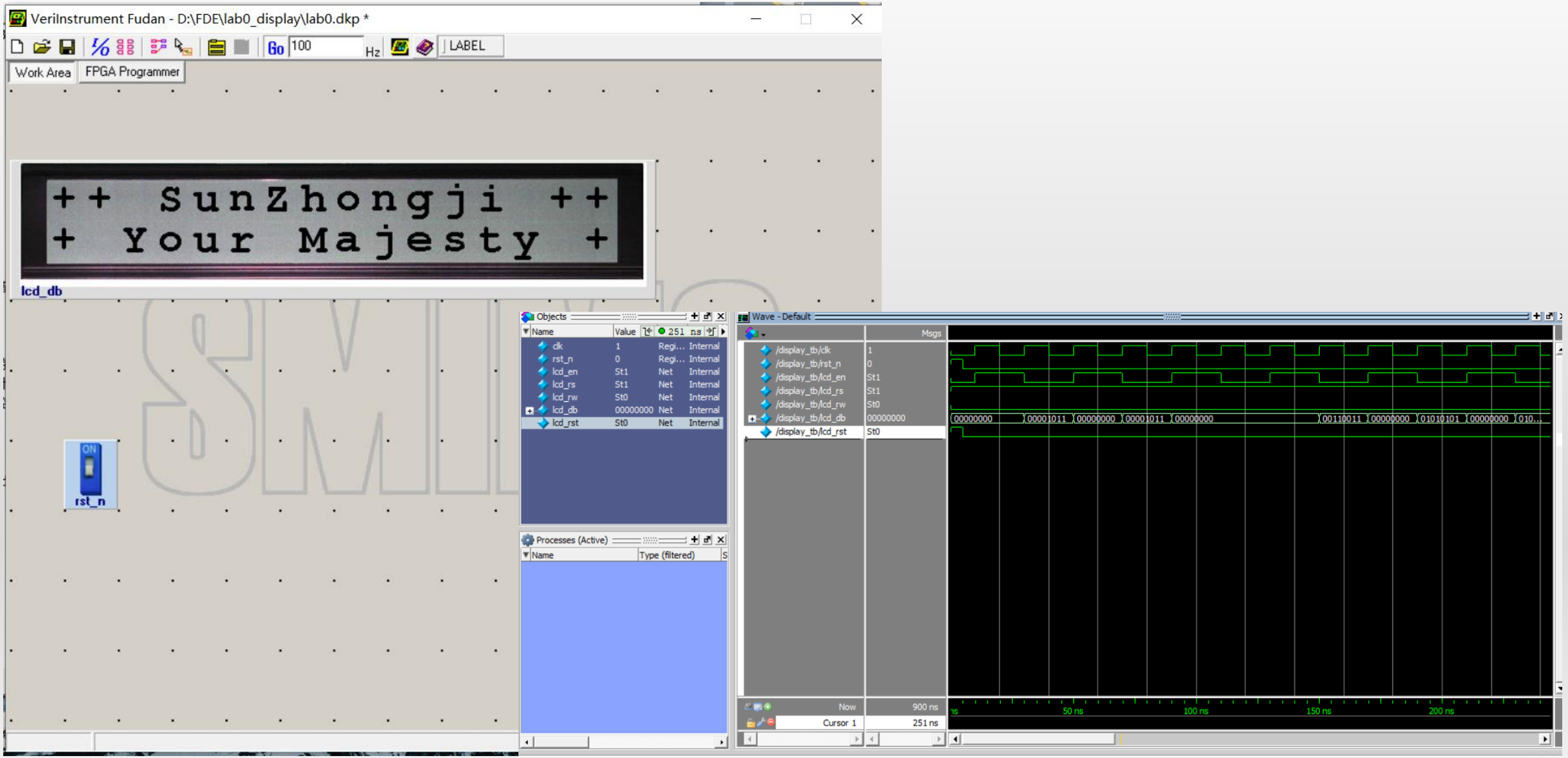
Important for HW-SW Co-design!!

OS: Windows 10
Visual Studio: 2019 x64 (unless underlined)

02

- Lab 0

Lab 0


VeriInstrument Fudan - D:\FDE\lab0_display\lab0.dkp *
Go
100
Hz
LABEL
Work Area
FPGA Programmer
++ SunZhongji ++
+ Your Majesty +
lcd_db
ON
rst_n
Objects
Name
Value
251 ns
clk
rst_n
lcd_en
lcd_rs
lcd_rw
lcd_db
lcd_rst
Processes (Active)
Type (filtered)
Wave - Default
Msgs
/display_tb/clk
/display_tb/rst_n
/display_tb/lcd_en
/display_tb/lcd_rs
/display_tb/lcd_rw
/display_tb/lcd_db
/display_tb/lcd_rst
00000000
00001011
00110011
01010101
Now
900 ns
Cursor 1
251 ns
50 ns
100 ns
150 ns
200 ns

03

- Lab 4

Shortcomings

1. Incompatible with latest version of OpenCV
2. Fixed constraint file and fixed Verilog logic
3. Hard to configure by oneself without experience

Rewritten DLL

1. Supports 64-bit version
2. Does not use FIFO which results in simpler Verilog code
3. Output is a clock cycle after input
4. Update USB protocol
5. Supported by HDLAutoAssign

Verilog Code

Home, 2 weeks ago | 1 author (Home)

```verilog
module inverter(        Home, 2 weeks ago • new
    input               clk,
    input               rstn,   // low active
    input       [15:0]  di,     // data input
    output reg  [15:0]  do      // data output
);

always @(posedge clk or negedge rstn) begin
    if (!rstn)  do <= 16'b0;
    else        do <= ~di;
end

endmodule

```

```verilog
module APP_Example(
  input             APP_CLK,
  input             APP_RSTN,
  input             APP_CS,

  output reg        APP_RD,
  input      [15:0] APP_DI,
  input             APP_Empty,

  output reg        APP_WR,
  output     [15:0] APP_DO,
  input             APP_Full);

  reg        [15:0] app_di_q, app_di_d;
  reg        [15:0] app_do_q, app_do_d;
  reg        [2:0]  cur_state, nxt_state;

  reg               LATCH_READ_DATA;
  reg               LATCH_WRITE_DATA;

  assign APP_DO = app_do_q;

  always@*
  begin
    app_di_d = (LATCH_READ_DATA) ? APP_DI : app_di_q;
    app_do_d = (LATCH_WRITE_DATA) ? ~app_di_q : app_do_q;
  end

  always@*
  begin : FSM
    parameter s0 = 3'd0;
    parameter s1 = 3'd1;
    parameter s2 = 3'd2;
    parameter s3 = 3'd3;
    parameter s4 = 3'd4;
    parameter s5 = 3'd5;

    nxt_state = cur_state;

    APP_RD = 1'b0;
    APP_WR = 1'b0;
    LATCH_READ_DATA = 1'b0;
    LATCH_WRITE_DATA = 1'b0;

    case(cur_state)
      s0 : begin
        nxt_state = (APP_CS & ~APP_Empty) ? s1 : s0;
      end
      s1 : begin
        APP_RD = 1'b1;
        nxt_state = s2;
      end
      s2 : begin
        LATCH_READ_DATA = 1'b1;
        nxt_state = s3;
      end
      s3 : begin
        LATCH_WRITE_DATA = 1'b1;
        nxt_state = (APP_Full) ? s3 : s4;
      end
      s4 : begin
        APP_WR = 1'b1;
        nxt_state = s5;
      end
      s5 : begin
```

Input & Output

Microsoft Visual Studio 调试控制台

index	input	output
[0]	[0000]	[FFFF]
[1]	[0001]	[FFFE]
[2]	[0002]	[FFFD]
[3]	[0003]	[FFFC]
[4]	[0004]	[FFFB]
[5]	[0005]	[FFFA]
[6]	[0006]	[FFF9]
[7]	[0007]	[FFF8]
[8]	[0008]	[FFF7]
[9]	[0009]	[FFF6]
[10]	[000A]	[FFF5]
[11]	[000B]	[FFF4]
[12]	[000C]	[FFF3]
[13]	[000D]	[FFF2]
[14]	[000E]	[FFF1]
[15]	[000F]	[FFF0]
[16]	[0010]	[FFEF]
[17]	[0011]	[FFEE]
[18]	[0012]	[FFED]
[19]	[0013]	[FFEC]
[20]	[0014]	[FFEB]
[21]	[0015]	[FFEA]
[22]	[0016]	[FFE9]
[23]	[0017]	[FFE8]
[24]	[0018]	[FFE7]
[25]	[0019]	[FFE6]
[26]	[001A]	[FFE5]
[27]	[001B]	[FFE4]
[28]	[001C]	[FFE3]

Microsoft Visual Studio 调试控制台

index	input	data	output
[0]	[0000]	[0000]	[0000]
[1]	[0000]	[0000]	[0000]
[2]	[0005]	[0002]	[0000]
[3]	[0007]	[0003]	[FFFD]
[4]	[0009]	[0004]	[FFFC]
[5]	[000B]	[0005]	[FFFB]
[6]	[000D]	[0006]	[FFFA]
[7]	[000F]	[0007]	[FFF9]
[8]	[0011]	[0008]	[FFF8]
[9]	[0013]	[0009]	[FFF7]
[10]	[0015]	[000A]	[FFF6]
[11]	[0017]	[000B]	[FFF5]
[12]	[0019]	[000C]	[FFF4]
[13]	[001B]	[000D]	[FFF3]
[14]	[001D]	[000E]	[FFF2]
[15]	[001F]	[000F]	[FFF1]
[16]	[0021]	[0010]	[FFF0]
[17]	[0023]	[0011]	[FFEF]
[18]	[0025]	[0012]	[FFEE]
[19]	[0027]	[0013]	[FFED]
[20]	[0029]	[0014]	[FFEC]
[21]	[002B]	[0015]	[FFEB]
[22]	[002D]	[0016]	[FFEA]
[23]	[002F]	[0017]	[FFE9]
[24]	[0031]	[0018]	[FFE8]
[25]	[0033]	[0019]	[FFE7]
[26]	[0035]	[001A]	[FFE6]
[27]	[0037]	[001B]	[FFE5]
[28]	[0039]	[001C]	[FFE4]

Image Process

1.GrayScale

2.Binarization

GrayScale

OpenCV Read in Images

24-bit RGB

Red	Green	Blue
0.25	0.5	0.25

8-bit GrayScale

Binarization

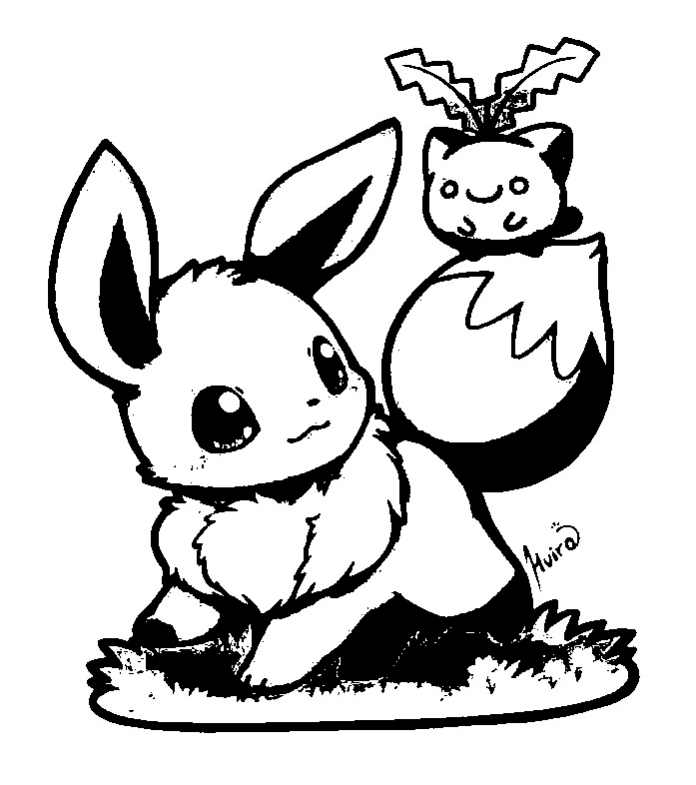
Threshold is 127 of 0-255

OpenCV Write out Images

8-bit GrayScale

8-bit Binarization

Examples



Examples

04

- Problems & Improvements

Deprecated Lab 4

VS 2013 x86 for Doc

Hard to configure without previous projects

Incompatible with OpenCV

Deprecated VideoPlayer in VeriComm

Better effects of image process using VideoPlayer shown in Doc

However, Loss of libraries & incompatible with Windows 10

More Knowledge of Image Process

Found a project on Edge Detection using Sobel Factors

However, Use a BRAM to store 3×3 pixel data

Further, most complicated image process needs clutch of data

Thanks for Your Patience!