Course Design of Computer Organization

Wang Zonghui

College of Computer Science & Technology,
Zhejiang University
March, 2010

topics

- 1. Verilog and Xilinx ISE
- 2. Basic logic component of datapath design
- 3. ALU and the ALU controller design
- 4. R-type instruction design
- 5. CPU controller design
- 6. Design of datapath of single-cycle clock CPU
- 7. Multiple-cycle clock CPU design
- 8. Microprogrammed CPU controller unit design
- 9. Design of datapath of Microprogrammed CPU

Outline

- Experiment Purpose
- Experiment Task
- Basic Principle
- Operating Procedures

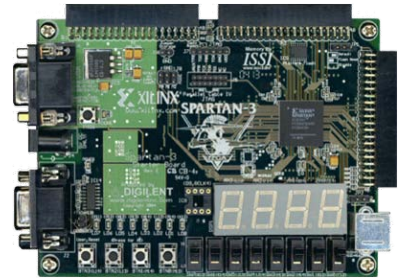
Experiment Purpose

- Familiar with programming and debugging in the Verilog language.
- Familiar with the basic operations on the Xilinx ISE software platform.
- Grasp the I/O applications on the Spartan-3 Starter Board

Experiment Task

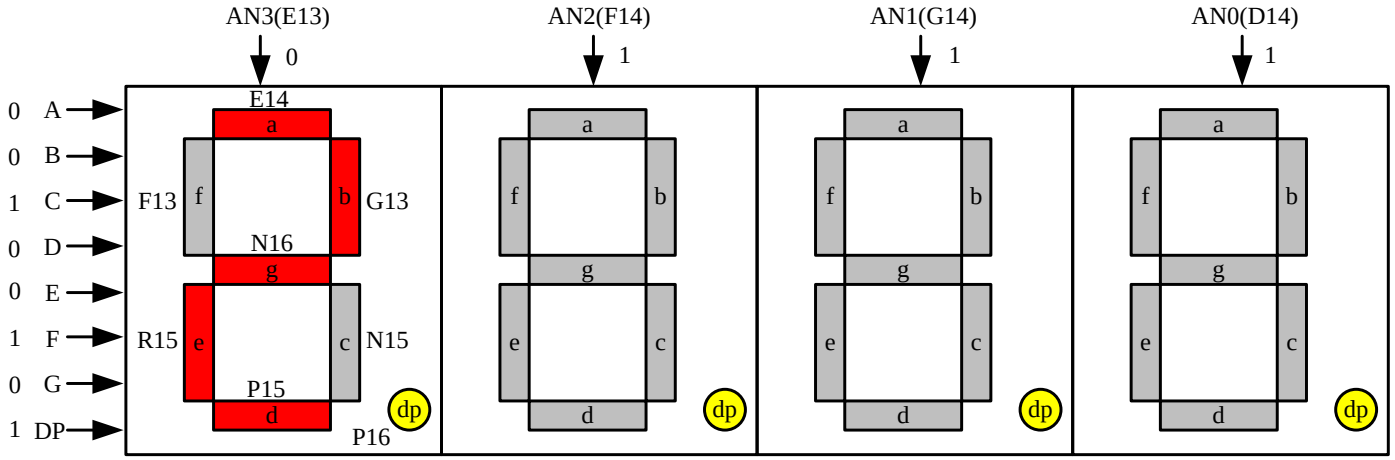
- With the I/O devices of the Spartan-3 development board, concatenate the input /output in the Verilog programming language.
- The I/O devices on the Spartan-3 Starter board consist of eight switches, four push buttons and eight LED(light-emitting diode), four digital tubes.

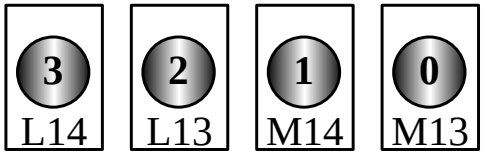
Basic Principle



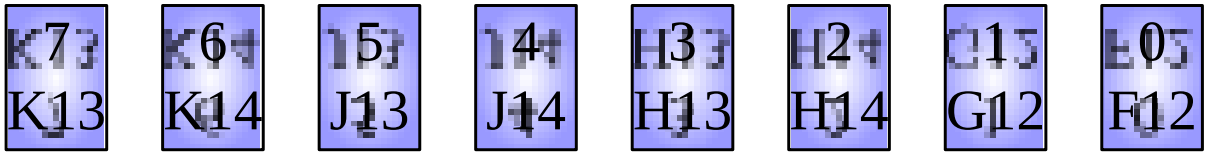
LED

7	6	5	4	3	2	1	0
P	P	N	P	N	L	P	K
11	12	12	13	14	12	14	12





Push Button



Switch Button

Experiment Content

- Display the 8 switch values on the corresponding LED.
- 4 push buttons increase the number of the corresponding four seven-segment digital tubes (plus 1 every button-press).
- The lower four switches of 8 switches control the four seven-segment digital tubes' flash statuses: flashes when the switch high, otherwise does not blink.
- The high four switches of 8 switches control the four seven-segment digital tubes' decimal points: light when the switch high, otherwise extinguish.
- 4 push buttons need dealing with anti-jitter

Top Module

```
module top(clk, button, switch, led, segment, digit_anode);
    input  wire       clk;
    input  wire [3:0] button;
    input  wire [7:0] switch;
    output wire [7:0] led;
    output wire [7:0] segment;
    output wire [3:0] digit_anode;
    reg [15:0] disp_num;
    wire [3:0] blink, dots;
    wire [3:0] button_out;
    wire       clk_500ms;

    anti_jitter  I1(clk, button, button_out);
    timer_500ms  I2(clk, clk_500ms);
    display      I3(clk, disp_num, dots, blink, clk_500ms, digit_anode,
    segment);

    initial disp_num <= 32'b0001_0010_0011_0100; // 0x1234
    assign  led = switch;
    assign  {dots, blink} = {~switch[7:4], switch[3:0]};
    always @(posedge button_out[0]) disp_num[ 3: 0] <= disp_num[ 3: 0] + 1;
    always @(posedge button_out[1]) disp_num[ 7: 4] <= disp_num[ 7: 4] + 1;
    always @(posedge button_out[2]) disp_num[11: 8] <= disp_num[11: 8] + 1;
    always @(posedge button_out[3]) disp_num[15:12] <= disp_num[15:12] + 1;
endmodule
```

UCF

```
NET "clk"             LOC = "T9";

NET "button[0]"       LOC = "M13";
NET "button[1]"       LOC = "M14";
NET "button[2]"       LOC = "L13";
NET "button[3]"       LOC = "L14";
NET "digit_anode[0]" LOC = "D14";
NET "digit_anode[1]" LOC = "G14";
NET "digit_anode[2]" LOC = "F14";
NET "digit_anode[3]" LOC = "E13";
NET "segment[0]"      LOC = "E14";
NET "segment[1]"      LOC = "G13";
NET "segment[2]"      LOC = "N15";
NET "segment[3]"      LOC = "P15";
NET "segment[4]"      LOC = "R16";
NET "segment[5]"      LOC = "F13";
NET "segment[6]"      LOC = "N16";
NET "segment[7]"      LOC = "P16";

                                  NET "switch[0]" LOC = "F12";
                                  NET "switch[1]" LOC = "G12";
                                  NET "switch[2]" LOC = "H14";
                                  NET "switch[3]" LOC = "H13";
                                  NET "switch[4]" LOC = "J14";
                                  NET "switch[5]" LOC = "J13";
                                  NET "switch[6]" LOC = "K14";
                                  NET "switch[7]" LOC = "K13";
                                  NET "led[0]"    LOC = "K12";
                                  NET "led[1]"    LOC = "P14";
                                  NET "led[2]"    LOC = "L12";
                                  NET "led[3]"    LOC = "N14";
                                  NET "led[4]"    LOC = "P13";
                                  NET "led[5]"    LOC = "N12";
                                  NET "led[6]"    LOC = "P12";
                                  NET "led[7]"    LOC = "P11";
```

Operating Procedures

- 1、Coding： Verilog Module & UCF
- 2、Synthesize &Implement
- 3、Generate Programming File
- 4、Program
- 5、Verify and Debug

- Precaution: anti-jitter module