数字电路实验 Lab 3 实验报告

奚项正 PB23000020

必做内容

题目 1：开关与 LED（2 分）

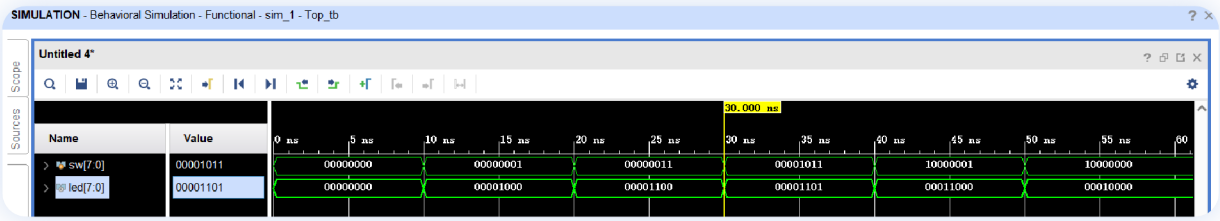
设计

```
module Top (
    input   [7:0]       sw,
    output  [7:0]       led
);
// Write your codes here.

reg [7:0] temp;
always @(*) begin
    temp[7] = sw[4];    //将sw调换顺序并存至temp
    temp[6] = sw[5];
    temp[5] = sw[6];
    temp[4] = sw[7];
    temp[3] = sw[0];
    temp[2] = sw[1];
    temp[1] = sw[2];
    temp[0] = sw[3];
end

assign led = temp;  //led与temp相连
endmodule
```

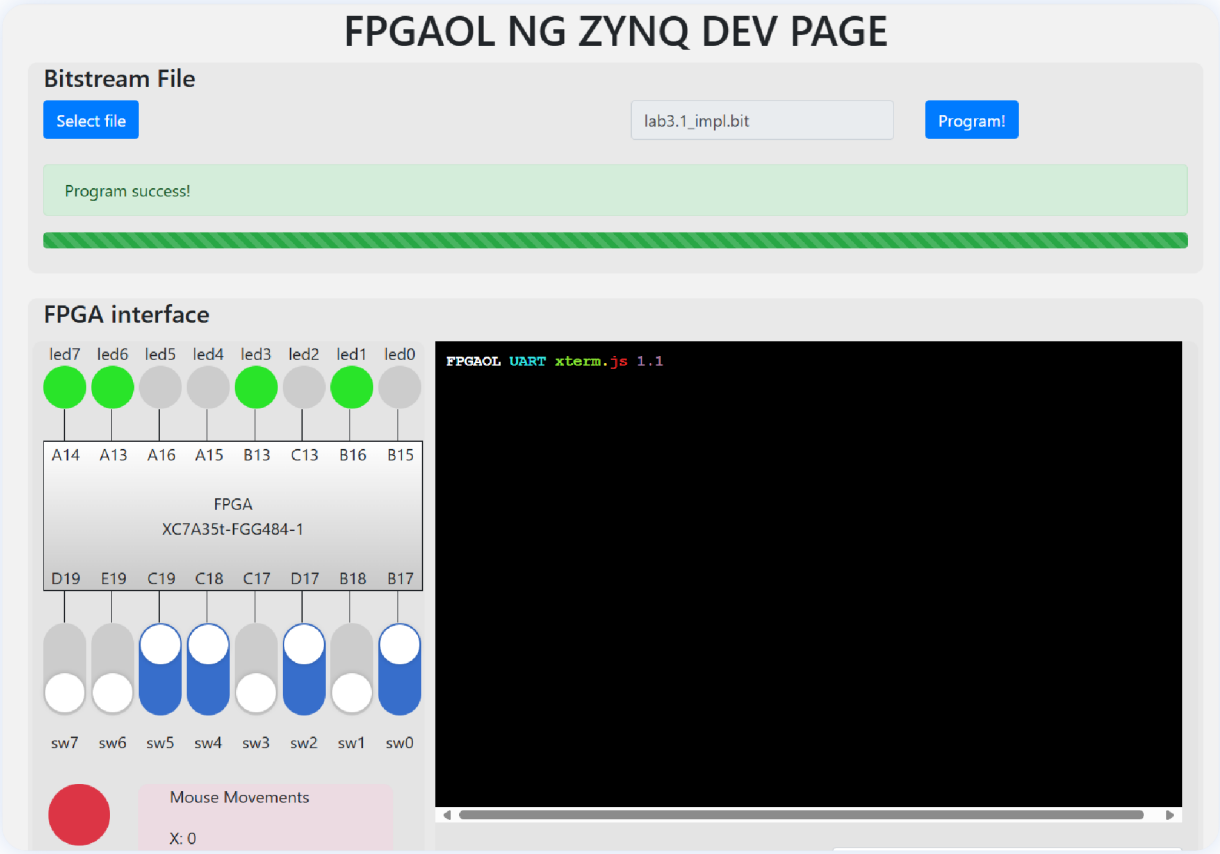
仿真

```
module Top_tb();
reg [7:0]   sw;
wire [7:0]  led;

initial begin
    sw = 8'b0000_0000;
    #10;
```

```
        sw = 8'b0000_0001;
        #10;
        sw = 8'b0000_0011;
        #10;
        sw = 8'b0000_1011;
        #10;
        sw = 8'b1000_0001;
        #10;
        sw = 8'b1000_0000;
    end

    Top top(
        .sw(sw),
        .led(led)
    );
    endmodule
```



实现



题目 2：计数器 Pro Plus（3 分）

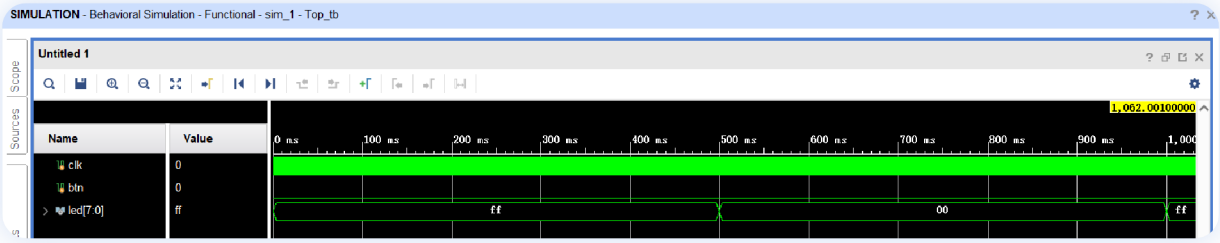
设计

```
module Top (
    input clk,
    input btn,
    output reg[7:0] led
);

reg [31:0] cnt;

initial begin   //初始化led与cnt
    led = 8'b1111_1111;
    cnt = 0;
end

always @(posedge clk) begin //时钟上升沿触发
    if (btn) begin  //用btn进行reset置零
        led <= 8'b0000_0000;
        cnt <= 0;
    end
    else if (cnt >= 50000000) begin //时钟周期为10ns，当cnt=50000000时表示累计0.5s
        cnt <= 0;
        led <= ~led;    //led全部状态改变
    end
    else begin
        cnt <= cnt + 1; //计数器增加1
    end
end

endmodule
```

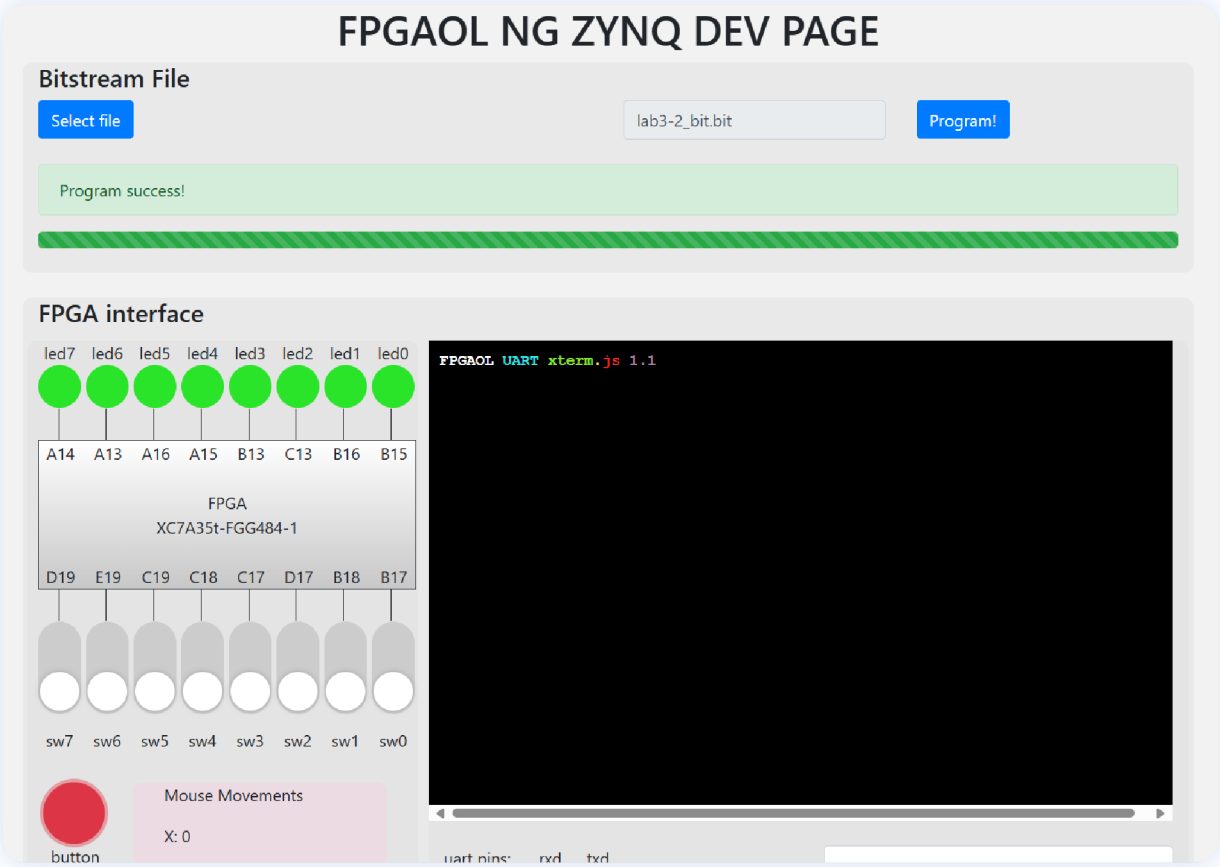
仿真

```
module Top_tb ();
reg clk;
reg btn;
wire [7:0] led;

initial begin
    clk = 2'b0;
    btn = 2'b0;
end

always #5 clk = ~clk;   //时钟周期为10ns

Top top(
    .clk(clk),
    .btn(btn),
    .led(led)
);
```

```
endmodule
```



实现



题目 3：七段数码管（3 分）

设计

```
module Top(
    input                   clk,
    input                   btn,
    output [2:0]            seg_an,
    output [3:0]            seg_data
);
Segment segment(
    .clk(clk),
    .rst(btn),
```

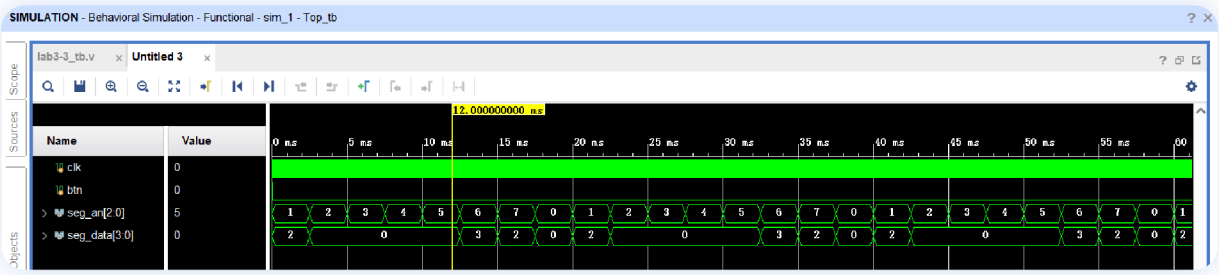
```
    .output_data(32'h23000020), //本人学号为PB23000020
    .seg_data(seg_data),
    .seg_an(seg_an)
);
endmodule
```

```
module Segment(
    input                       clk,
    input                       rst,
    input       [31:0]          output_data,

    output reg  [ 3:0]          seg_data,
    output reg  [ 2:0]          seg_an
);

reg [31:0]  counter;
reg [2:0]   seg_id;

always @(posedge clk) begin //更新counter
    if (rst) begin
        counter <= 0;
    end
    else if (counter >= 250000) begin   //周期为每400Hz
        counter <= 0;
    end
    else begin
        counter <= counter + 1;
    end
end

always @(posedge clk) begin //更新seg_id
    if (rst) begin
        seg_id <= 0;
    end
    else if (counter == 1) begin    //每个counter周期将seg_id增加1
        if (seg_id >= 8) begin  //seg_id范围为0-7间
            seg_id <= 0;
        end
        else begin
            seg_id <= seg_id + 1;
        end
    end
    else begin
        seg_id <= seg_id;
    end
end

always @(*) begin   //更新seg_data至seg_id所对应的数码管
    seg_data = 0;   //需要熄灭其他数码管，保证每次只有一个点亮
    seg_an = seg_id;    //对所有情形皆如此

    case (seg_an)   //采用分时复用的方式轮流点亮每个数码管
```

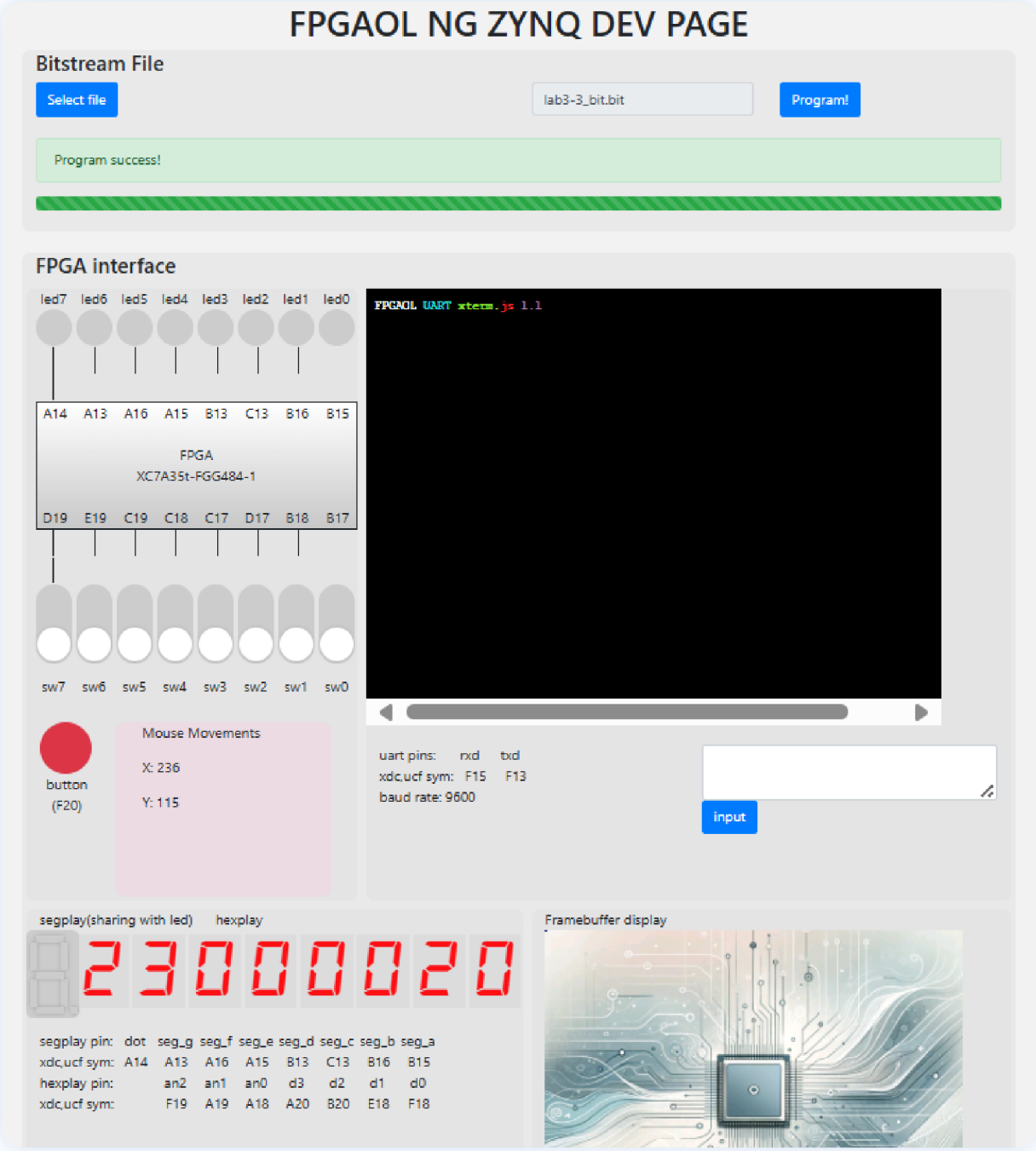
```
        0: seg_data = output_data[3:0];
        1: seg_data = output_data[7:4];
        2: seg_data = output_data[11:8];
        3: seg_data = output_data[15:12];
        4: seg_data = output_data[19:16];
        5: seg_data = output_data[23:20];
        6: seg_data = output_data[27:24];
        7: seg_data = output_data[31:28];
        default: seg_data = 0;
    endcase
end
endmodule
```

仿真

```
module Top_tb();
reg clk;
reg btn;
wire [2:0]  seg_an;
wire [3:0]  seg_data;

initial begin
    clk = 2'b0;
    btn = 2'b1; //先进行reset
    #10 btn = 2'b0;
end

always #5 clk = ~clk;    //时钟周期为10ns

Top top(
    .clk(clk),
    .btn(btn),
    .seg_an(seg_an),
    .seg_data(seg_data)
);
endmodule
```



实现



选择性必做内容

题目 3：带有掩码的数码管（2 分）

设计

```
module Top(
    input                   clk,
    input                   btn,
    input  [7:0]            sw,
    output [2:0]            seg_an,
    output [3:0]            seg_data
);
Segment segment(
```

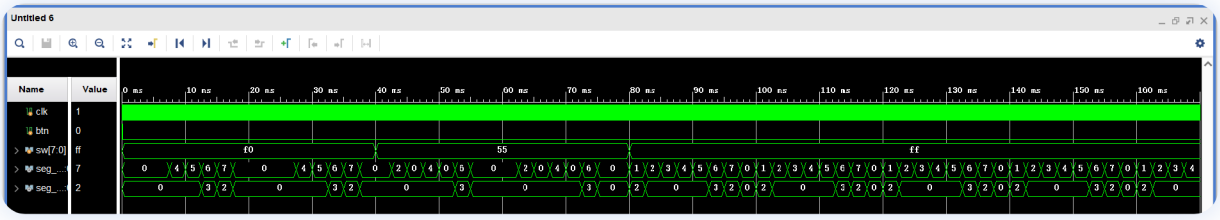
```
    .clk(clk),
    .rst(btn),
    .output_data(32'h23000020),
    .output_valid(sw),  //output_valid连接至sw开关
    .seg_data(seg_data),
    .seg_an(seg_an)
);
endmodule
```

```
module Segment(
    input                       clk,
    input                       rst,
    input       [31:0]          output_data,
    input       [ 7:0]          output_valid,   //补充output_valid输入
    output reg  [ 3:0]          seg_data,
    output reg  [ 2:0]          seg_an
);

reg [31:0]  counter;
reg [2:0]   seg_id;

always @(posedge clk) begin
    if (rst) begin
        counter <= 0;
    end
    else if (counter >= 250000) begin
        counter <= 0;
    end
    else begin
        counter <= counter + 1;
    end
end

always @(posedge clk) begin
    if (rst) begin
        seg_id <= 0;
    end
    else if (counter == 1) begin
        if (seg_id >= 8) begin
            seg_id <= 0;
        end
        else begin
            seg_id <= seg_id + 1;
        end
    end
    else begin
        seg_id <= seg_id;
    end
end

always @(*) begin
    seg_data = 0;
```

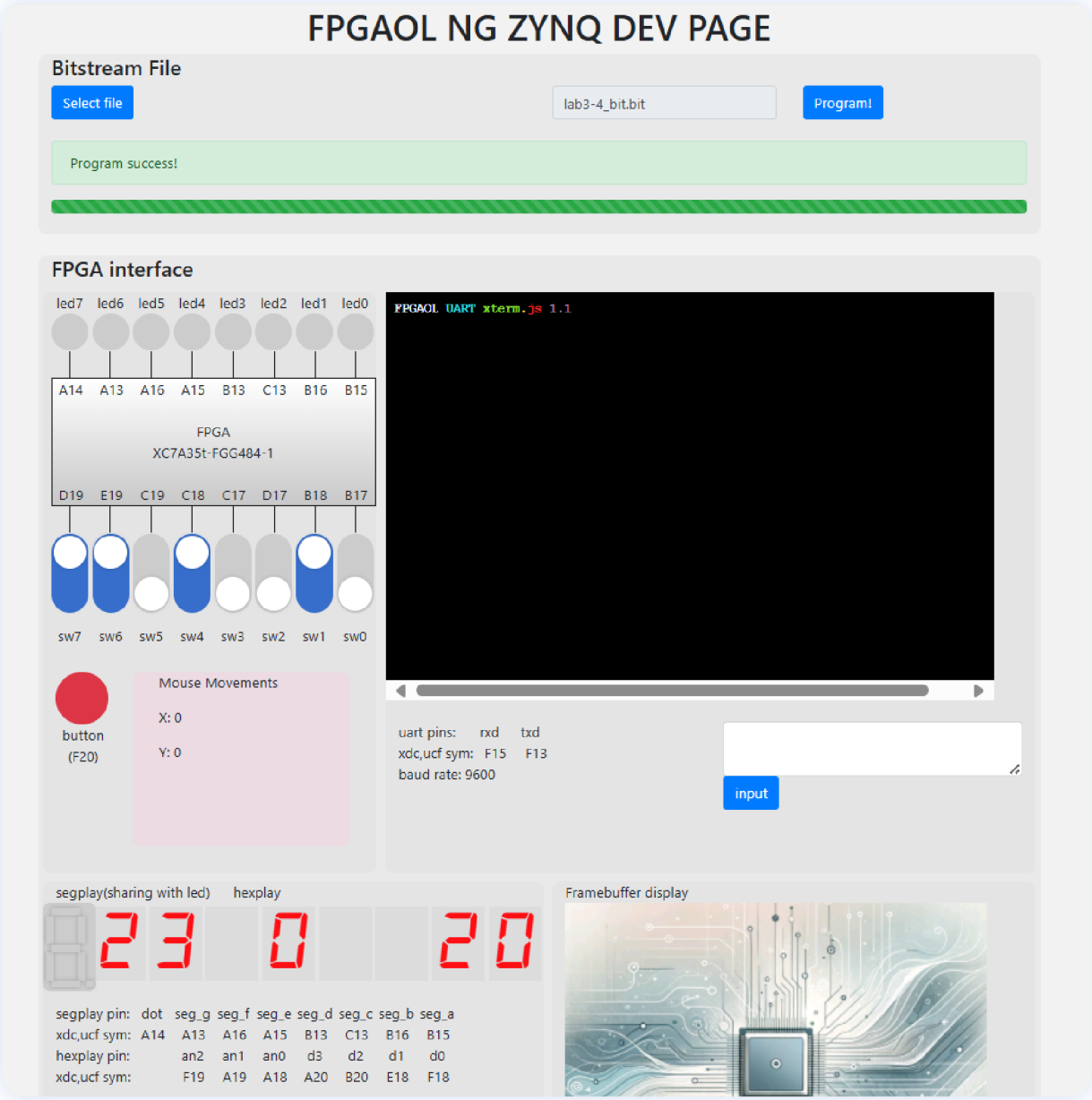
```
    if (output_valid[seg_id] == 1) begin    //补充对开关状态的判断，如果断开则seg_an赋0
        seg_an = seg_id;
    end
    else begin
        seg_an = 0;
    end

    case (seg_an)
        0: seg_data = output_data[3:0];
        1: seg_data = output_data[7:4];
        2: seg_data = output_data[11:8];
        3: seg_data = output_data[15:12];
        4: seg_data = output_data[19:16];
        5: seg_data = output_data[23:20];
        6: seg_data = output_data[27:24];
        7: seg_data = output_data[31:28];
        default: seg_data = 0;
    endcase
end
endmodule
```

仿真

```
module Top_tb();
reg clk;
reg btn;
reg [7:0]   sw;
wire [2:0]  seg_an;
wire [3:0]  seg_data;

initial begin
    clk = 2'b0;
    btn = 2'b1;
    sw = 8'b11110000;   //设定并测试开关
    #10 btn = 2'b0;
    #40000000 sw = 8'b01010101; //间隔40ms
    #40000000 sw = 8'b11111111;
end

always #5 clk = ~clk;

Top top(
    .clk(clk),
    .btn(btn),
    .sw(sw),
    .seg_an(seg_an),
    .seg_data(seg_data)
);
endmodule
```


Untitled 6
Name
Value
clk
btn
sw[7:0]
seg_...
seg_...
ff
f0
55
ff

实现


FPGAOL NG ZYNQ DEV PAGE
Bitstream File
Select file
lab3-4_bit.bit
Program!
Program success!
FPGA interface
led7 led6 led5 led4 led3 led2 led1 led0
A14 A13 A16 A15 B13 C13 B16 B15
FPGA
XC7A35t-FGG484-1
D19 E19 C19 C18 C17 D17 B18 B17
sw7 sw6 sw5 sw4 sw3 sw2 sw1 sw0
FPGAOL UART xterm.js 1.1
button
(F20)
Mouse Movements
X: 0
Y: 0
uart pins: rxd txd
xdc,ucf sym: F15 F13
baud rate: 9600
input
segplay(sharing with led)
hexplay
segplay pin: dot seg_g seg_f seg_e seg_d seg_c seg_b seg_a
xdc,ucf sym: A14 A13 A16 A15 B13 C13 B16 B15
hexplay pin: an2 an1 an0 d3 d2 d1 d0
xdc,ucf sym: F19 A19 A18 A20 B20 E18 F18
Framebuffer display