计算机组成原理 实验报告

姓名：阿卜杜赛米江·萨吾提　学号：PB19111752　实验日期：2021-4-15

一、实验题目：

Lab03　寄存器堆与队列

二、实验目的：

使用 vivado 提供的 IP 核，熟悉寄存器堆的使用，再使用此寄存器堆实现优先队列。

三、实验平台：

Vivado/fpgaol

四、实验过程：

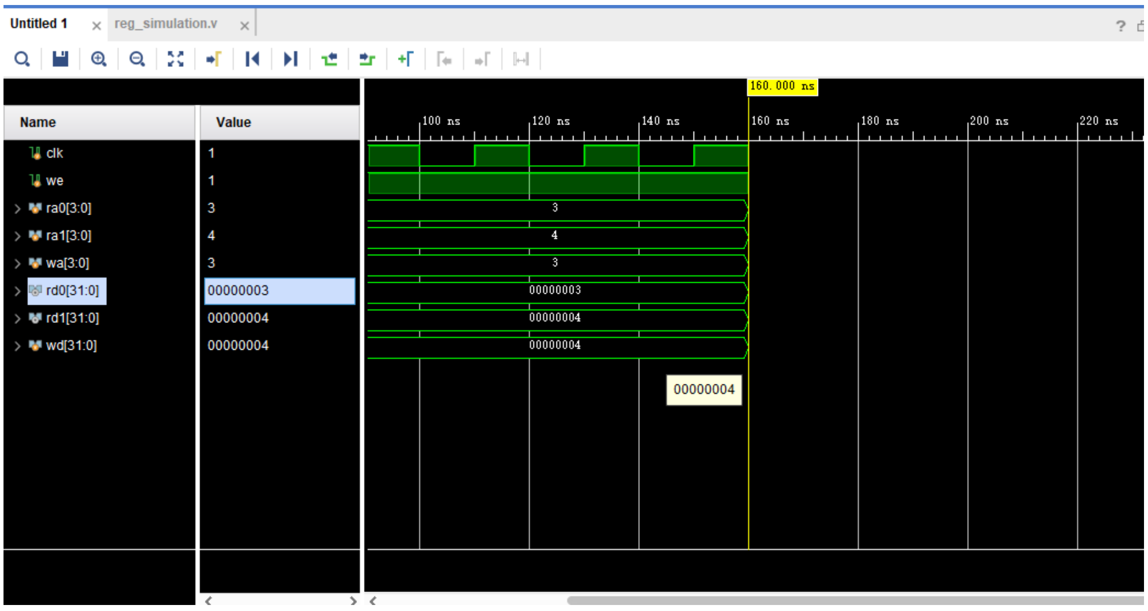
```
    `timescale 1ns / 1ps
module fifo_lihua (
    input clk, BTN,
    input [7:0] sw,
    output [2:0] AN,
    output [3:0] D,
    output [7:0] led
    );
    wire [3:0] front, curr_size;
    wire [3:0] ra, rd;

    display dis_elment (
        .clk(clk), .rst(BTN),
        .front(front), .curr_size(curr_size),
        .led(led[3:0]), .D(D),
        .AN(AN),
        .ra(ra), .rd(rd)
    );
    FIFO FIFO (
        .clk(clk), .rst(BTN),
        .in(sw[3:0]),
        .enq(sw[7]), .deq(sw[6]),
        .full(led[7]), .empty(led[6]),
        .front(front), .curr_size(curr_size),
        .ra(ra), .rd(rd)
    );
endmodule

module display (
    input clk, rst,
    input [3:0] front, curr_size,
    output reg [3:0] led,
    output [3:0] D,
    output reg [2:0] AN,

    //get targeted data by ra
    output reg [3:0] ra,
    input [3:0] rd
    );
    parameter MAX_SIZE = 8;

    wire clk_out1;

    assign D = rd;

    always @(posedge clk_out1) begin
        if(rst == 1 | curr_size == 0 | AN >= curr_size-1) begin
```

```
            ra <= front;
            AN <= 0;
        end
        else begin
            ra <= (ra+1)%(MAX_SIZE+1);
            AN <= AN + 1;
        end

        if(ra == front) begin
            led <= rd;
        end
    end

    clk_10m clk_10m(
        .clk_in1(clk),
        .reset(rst),
        .clk_out1(clk_out1)
        );
endmodule

module fifo (
    input clk, rst,
    input [3:0] in,
    input enq, deq,
    output full, empty,
output reg [3:0] front,     //return top address      output reg
    [3:0] curr_size,

        input [3:0] ra,
output [3:0] rd
);
parameter MAX_SIZE = 8;

wire enq_edge, deq_edge;
reg  [3:0] wa;

assign empty = (front == wa);
assign full  = ((wa+1)%(MAX_SIZE+1) == front);

always @(negedge clk) begin
    if(rst) begin
        front <= 0;
        wa <= 0;
        curr_size <= 0;
    end
    else if(enq_edge == 1 & full == 0) begin // enqueue
        wa <= (wa+1)%(MAX_SIZE+1);
        curr_size <= curr_size + 1;
    end
    else if(deq_edge == 1 & empty == 0) begin // dequeue
        front <= (front+1)%(MAX_SIZE+1);
        curr_size <= curr_size - 1;
    end
end

regfile #(.A_MSB(3), .D_MSB(3)) RegFile (
    .clk(clk), .we(1), //writing always enable
    .wa(wa),   .wd(in),
        .ra0(ra), .rd0(rd)
        );

    signal_edge ENQ_edge(
        .clk(clk),
        .btn(enq),
        .btn_edge(enq_edge)
        );

    signal_edge DEQ_edge(
        .clk(clk),
        .btn(deq),
        .btn_edge(deq_edge)
        );
endmodule

module regfile #(parameter A_MSB = 31, D_MSB = 31, A_LSB = 0,
D_LSB = 0) (
    input clk, we,                     // write enable
    input [D_MSB:D_LSB] wd,            // write data
    input [A_MSB:A_LSB] wa, ra0, ra1, // write, read address
    output [D_MSB:D_LSB] rd0, rd1      // read data
    );
    reg [D_MSB:D_LSB] memory[2**(A_MSB-A_LSB+1):0]; //
2^address

    assign rd0 = memory[ra0];
    assign rd1 = memory[ra1];
```

```
            always @(posedge clk) begin
                if(we) begin
                    memory[wa] <= wd;
                end
            end
        endmodule
        //delay the react of btn
        module signal_edge(
            input clk,
            input btn,
            output btn_edge
            );
            reg r1,r2;
            always@(negedge clk) r1 <= btn;
            always@(negedge clk) r2 <= r1;
            assign btn_edge = r1 & (~r2);
endmodule
```

五、实验结果：

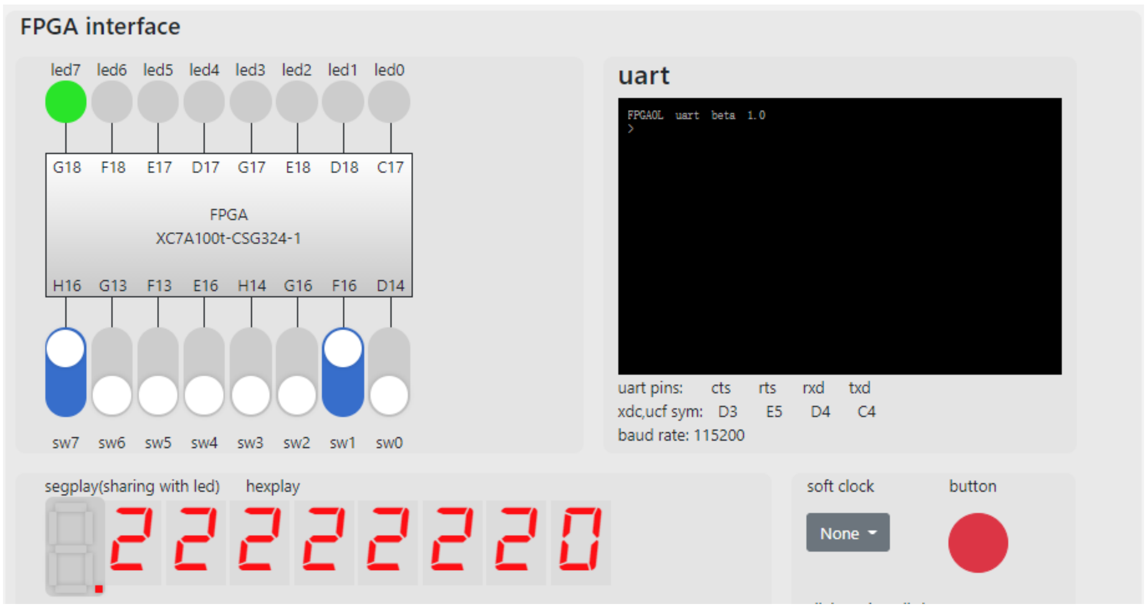
1.寄存器堆的仿真；



```
Name        Value
clk         1
we          1
ra0[3:0]    3
ra1[3:0]    4
wa[3:0]     3
rd0[31:0]   00000003
rd1[31:0]   00000004
wd[31:0]    00000004
```

2. 队列的效果展示：

队满：

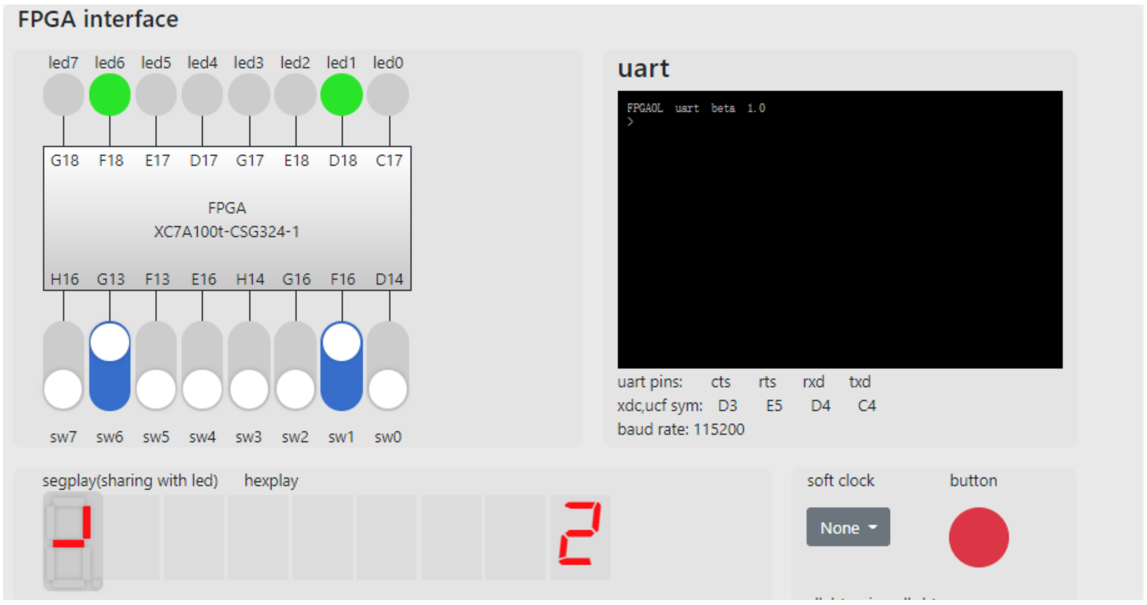

FPGA interface
led7 led6 led5 led4 led3 led2 led1 led0
G18 F18 E17 D17 G17 E18 D18 C17
FPGA
XC7A100t-CSG324-1
H16 G13 F13 E16 H14 G16 F16 D14
sw7 sw6 sw5 sw4 sw3 sw2 sw1 sw0
uart
segplay(sharing with led)
hexplay
soft clock
None
button

```
FPGAOL uart beta 1.0
>
```

uart pins: cts rts rxd txd
xdc,ucf sym: D3 E5 D4 C4
baud rate: 115200

22222220

队空：


FPGA interface
led7 led6 led5 led4 led3 led2 led1 led0
G18 F18 E17 D17 G17 E18 D18 C17
FPGA
XC7A100t-CSG324-1
H16 G13 F13 E16 H14 G16 F16 D14
sw7 sw6 sw5 sw4 sw3 sw2 sw1 sw0
uart
segplay(sharing with led)
hexplay
soft clock
None
button

```
FPGAOL uart beta 1.0
>
```

uart pins: cts rts rxd txd
xdc,ucf sym: D3 E5 D4 C4
baud rate: 115200

2

六、心得体会：

熟悉了寄存器堆的基恩使用，进一步加深了对分模块编写代码的诸多好处