计算机组成原理 实验报告

实验题目：寄存器堆与存储器及其应用
学生姓名：阿非提
学生学号：PB20111633
完成日期：2022.4.5

实验目的

- 掌握寄存器堆（Register File）和存储器的功能、时序及其应用
- 熟练掌握数据通路和控制器的设计和描述方法

实验平台

- Vivado

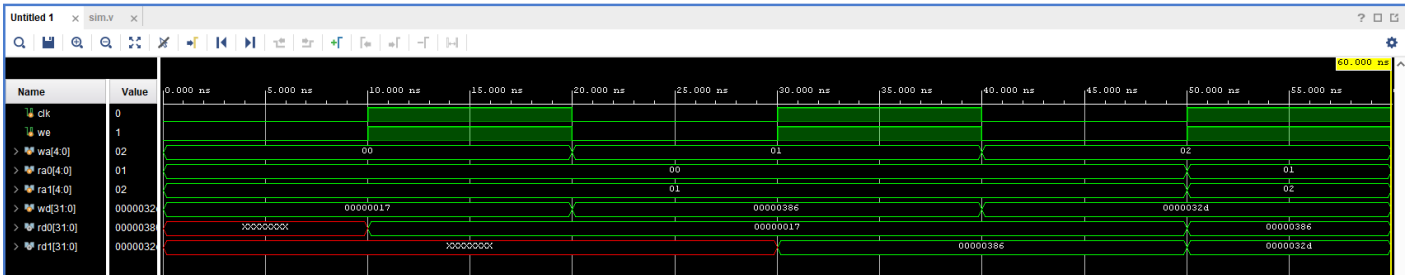
实验练习

- **题目1**

  设计文件

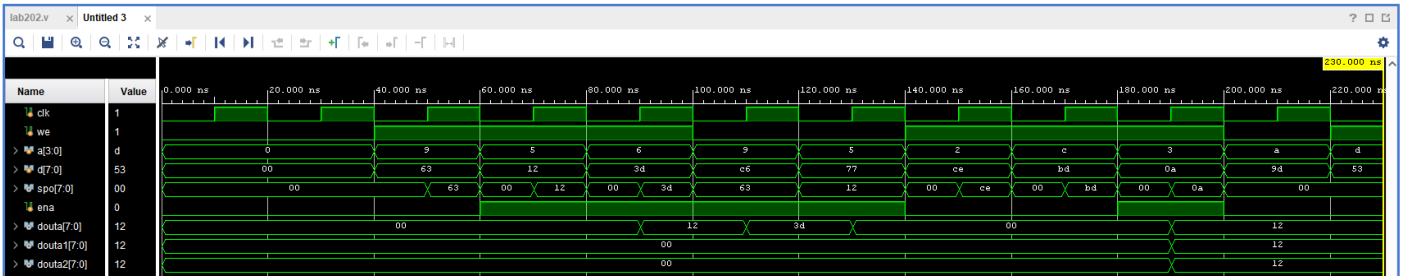
```
module register_file #(parameter WIDTH = 32)(
    input clk,we,
    input [4:0] ra0,ra1,wa,
    input [WIDTH-1:0] wd,
    output [WIDTH-1:0] rd0,rd1
);

reg [WIDTH-1:0] regfile[0:31];

assign rd0 = regfile[ra0], rd1 = regfile[ra1];

always@(posedge clk)begin
    if(we)regfile[wa]<=wd;
end

endmodule
```

  仿真结果



- **题目2**

  仿真结果



✳其中douta 、douta1、douta2 分别是块式RAM 在write first、read first、no change模式下的仿真结果

✳仿真开始前两种RAM里的所有值都初始化为0

对比结果发现块式RAM的读、写均和时钟同步，而分布式RAM写和时钟同步，读不需要时钟同步。块式RAM有三种操作模式分别为，write first、read first、no change。

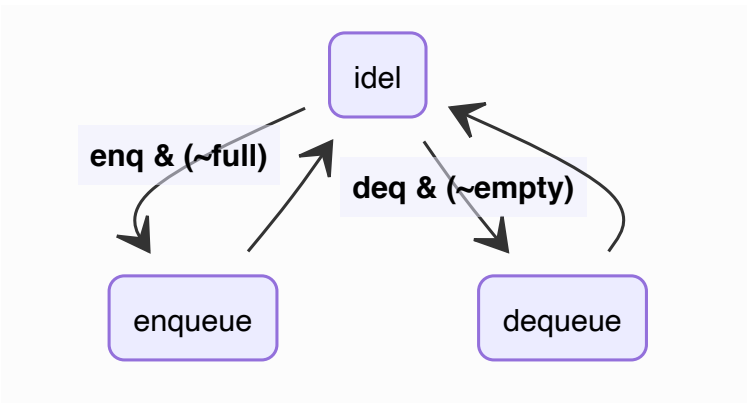
write first 模式下，写入的数据一边写到RAM内核，一边输出到端口。

read first 模式下，写入的数据写到RAM内核，写地址指向的RAM数据输出到端口。

no change 模式下，写入的数据写到RAM内核。输出端口的数据（实际上有个寄存器）保持不变。

- **题目3**

  状态图



数码管需要使用分时复用的方法并根据valid数组判断是否需要显示。

Verilog代码

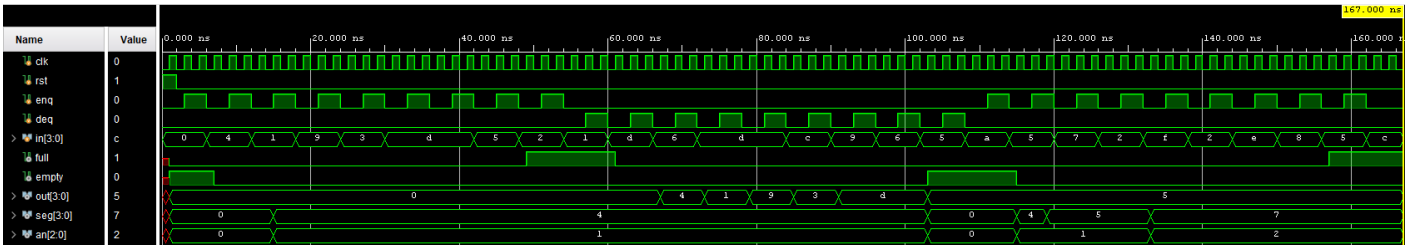
```
module FIFO (
input clk, rst,
input enq,
input [3:0] in,
input deq,
output reg full,empty,
output [3:0] out,
output [2:0] an,
output [3:0] seg
);

//edge detection
reg e1,e2,d1,d2;
wire Enq,Deq;

always@(posedge clk)begin
  e1<=enq;
  d1<=deq;
end
always@(posedge clk)begin
  e2<=e1;
  d2<=d1;
end

assign Enq = e1 & (~e2);
assign Deq = d1 & (~d2);

//LCU
reg [2:0] head,tail;
reg [7:0] valid;
reg [3:0] OUT;


parameter IDLE = 2'd0;
```

```
parameter ENQUEUE = 2'd1;
parameter DEQUEUE = 2'd2;

reg [1:0] currentState,nextState;

//state transition
always@(*)begin
  if(Enq&&(~Deq)&&(~full)) nextState = ENQUEUE;
  else if((~Enq)&&Deq&&(~empty)) nextState = DEQUEUE;
  else nextState = IDLE;
end

//state logic
always@(posedge clk)begin
  if(rst)
      currentState <= IDLE;
  else
      currentState <= nextState;
end

//data
always@(posedge clk)begin
  if(rst)begin
    head <= 3'd0;
    tail <= 3'd0;
    valid <= 8'd0;
  end
  else begin
    case(currentState)
      IDLE:;
      ENQUEUE: begin
        tail <= tail + 3'd1;
        valid[tail] <= 1'd1;
      end
      DEQUEUE: begin
        head <= head + 3'd1;
        valid[head] <= 1'd0;
      end
      default:;
    endcase;
  end
```

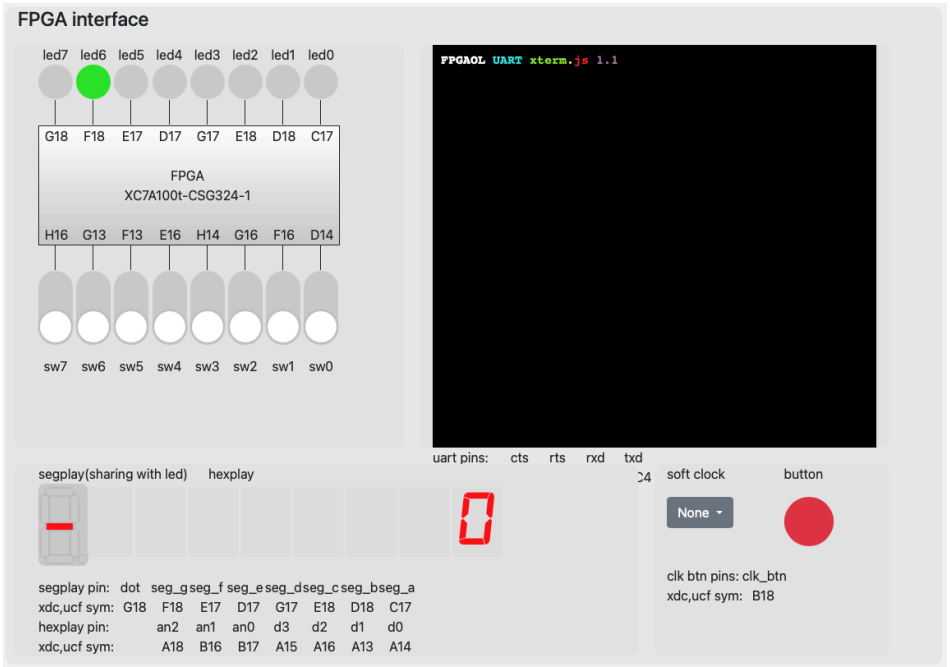
```
end

always@(*)begin
  if(valid==8'd255) full = 1'd1;
  else full = 1'd0;
end

always@(*)begin
  if(valid==8'd0) empty = 1'd1;
  else empty = 1'd0;
end

//segment display unit
reg [5:0] counter;
reg [2:0] current_an,next_an;
always@(posedge clk)begin
  if(rst)
    counter <= 6'd0;
  else
    counter <= counter + 6'd1;
end
always@(posedge clk)begin
    if(rst)
      next_an <= 3'd0;
    else if(counter==0)
      next_an <= next_an + 3'd1;
    else
      next_an <= next_an;
end
always@(posedge clk)begin
    if(rst)
      current_an <= 3'd0;
    else if(valid[next_an])
      current_an <= next_an;
    else
      current_an <= current_an;
end

 //register file

wire we;
```

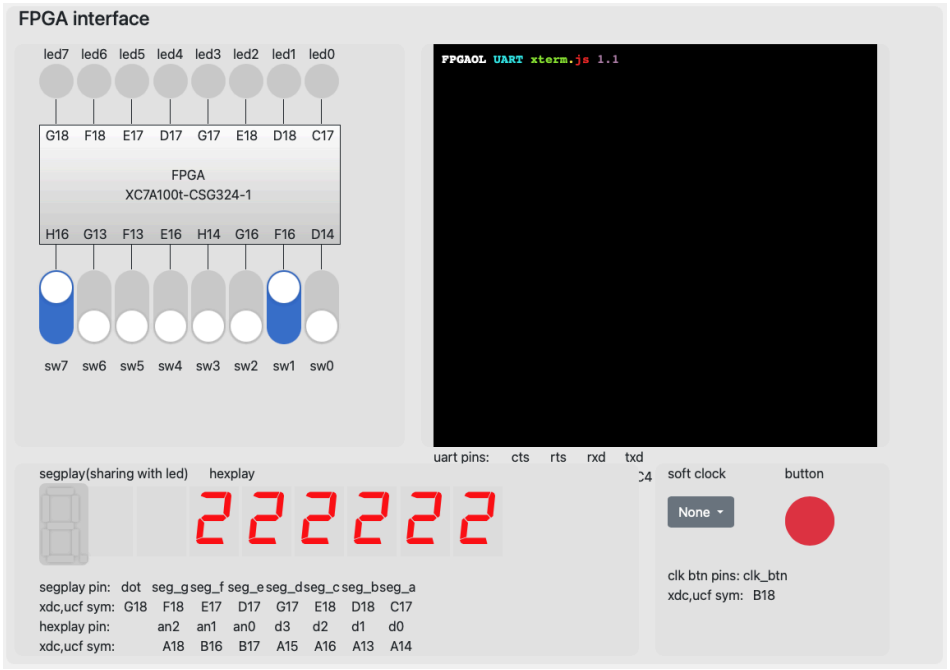
```
wire [3:0] wd,rd0,rd1;
wire [2:0] wa,ra0,ra1;

assign we = Enq & (~full);
assign wa = tail;
assign wd = in;
assign ra0 = head;
assign ra1 = current_an ;


assign an = empty?head:current_an;
assign seg = empty?0:rd1;

always@(posedge clk)begin
  if(rst) OUT <= 4'd0;
  else if(currentState == DEQUEUE) OUT <= rd0;
  else OUT <= OUT;
end

assign out = OUT;

register_file register_file(
    .clk(clk),
    .we(we),
    .wa(wa),
    .wd(wd),
    .ra0(ra0),
    .ra1(ra1),
    .rd0(rd0),
    .rd1(rd1)
);


endmodule
```

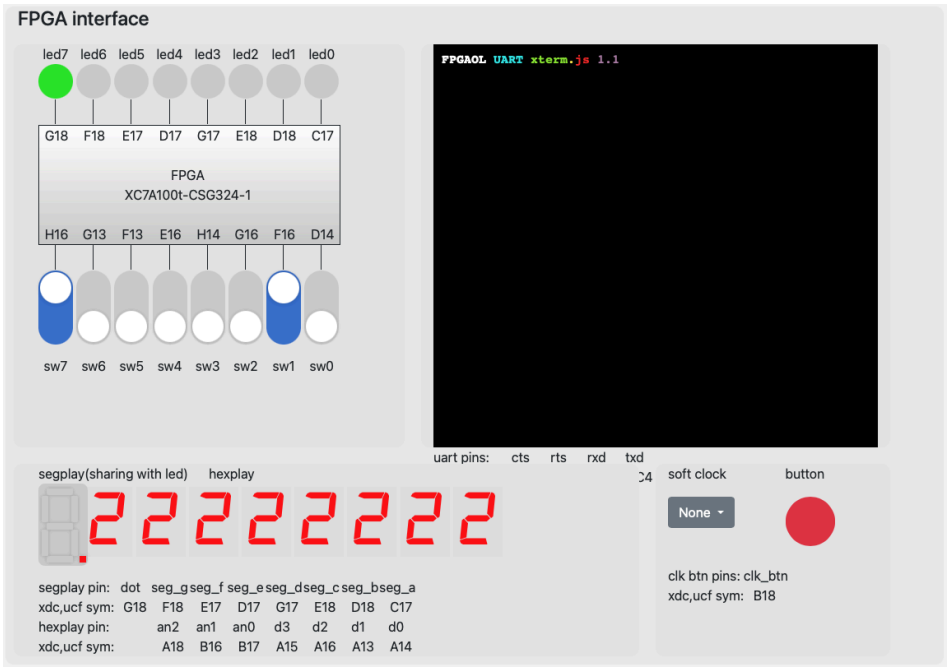
仿真结果


Name
Value
clk
rst
enq
deq
in[3:0]
full
empty
out[3:0]
seg[3:0]
an[2:0]
0.000 ns
20.000 ns
40.000 ns
60.000 ns
80.000 ns
100.000 ns
120.000 ns
140.000 ns
167.000 ns

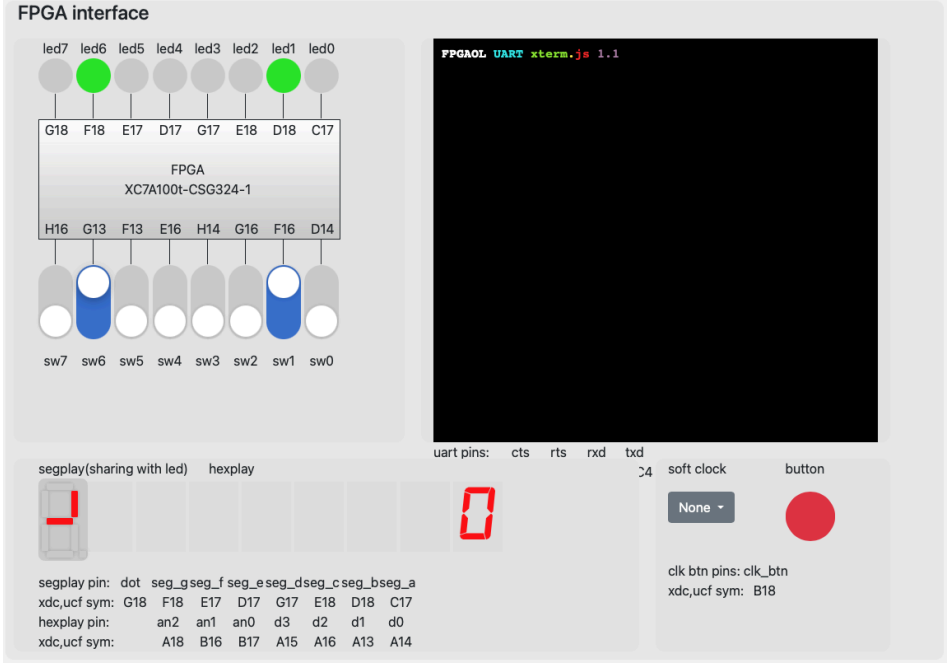
运行结果


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
FPGAOL UART xterm.js 1.1
uart pins: cts rts rxd txd
segplay(sharing with led) hexplay
soft clock
button
None
clk btn pins: clk_btn
xdc,ucf sym: B18
segplay pin: dot seg_g seg_f seg_e seg_d seg_c seg_b seg_a
xdc,ucf sym: G18 F18 E17 D17 G17 E18 D18 C17
hexplay pin: an2 an1 an0 d3 d2 d1 d0
xdc,ucf sym: A18 B16 B17 A15 A16 A13 A14


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
FPGAOL UART xterm.js 1.1
uart pins: cts rts rxd txd
segplay(sharing with led) hexplay
soft clock
button
None
clk btn pins: clk_btn
xdc,ucf sym: B18
segplay pin: dot seg_g seg_f seg_e seg_d seg_c seg_b seg_a
xdc,ucf sym: G18 F18 E17 D17 G17 E18 D18 C17
hexplay pin: an2 an1 an0 d3 d2 d1 d0
xdc,ucf sym: A18 B16 B17 A15 A16 A13 A14


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
FPGAOL UART xterm.js 1.1
uart pins: cts rts rxd txd
segplay(sharing with led) hexplay
soft clock
button
None
segplay pin: dot seg_g seg_f seg_e seg_d seg_c seg_b seg_a
xdc,ucf sym: G18 F18 E17 D17 G17 E18 D18 C17
hexplay pin: an2 an1 an0 d3 d2 d1 d0
xdc,ucf sym: A18 B16 B17 A15 A16 A13 A14
clk btn pins: clk_btn
xdc,ucf sym: B18


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
FPGAOL UART xterm.js 1.1
uart pins: cts rts rxd txd
segplay(sharing with led) hexplay
soft clock
button
None
segplay pin: dot seg_g seg_f seg_e seg_d seg_c seg_b seg_a
xdc,ucf sym: G18 F18 E17 D17 G17 E18 D18 C17
hexplay pin: an2 an1 an0 d3 d2 d1 d0
xdc,ucf sym: A18 B16 B17 A15 A16 A13 A14
clk btn pins: clk_btn
xdc,ucf sym: B18


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
FPGAOL UART xterm.js 1.1
uart pins: cts rts rxd txd
segplay(sharing with led) hexplay
soft clock
button
None
segplay pin: dot seg_g seg_f seg_e seg_d seg_c seg_b seg_a
xdc,ucf sym: G18 F18 E17 D17 G17 E18 D18 C17
hexplay pin: an2 an1 an0 d3 d2 d1 d0
xdc,ucf sym: A18 B16 B17 A15 A16 A13 A14
clk btn pins: clk_btn
xdc,ucf sym: B18

总结与思考

- 通过本次试验我学会了如何通过块式和分布式RAM的使用以及区别 。
- 本次试验难易程偏低。
- 本次试验任务量适中。