

寄存器堆与存储器及其应用

计算机组成原理实验 二

PB19071501 李平治

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

实验环境

- Windows 10 虚拟机
- Vivado
- 在线 FPGA fpgaol.ustc.edu.cn

寄存器堆

1. 逻辑设计

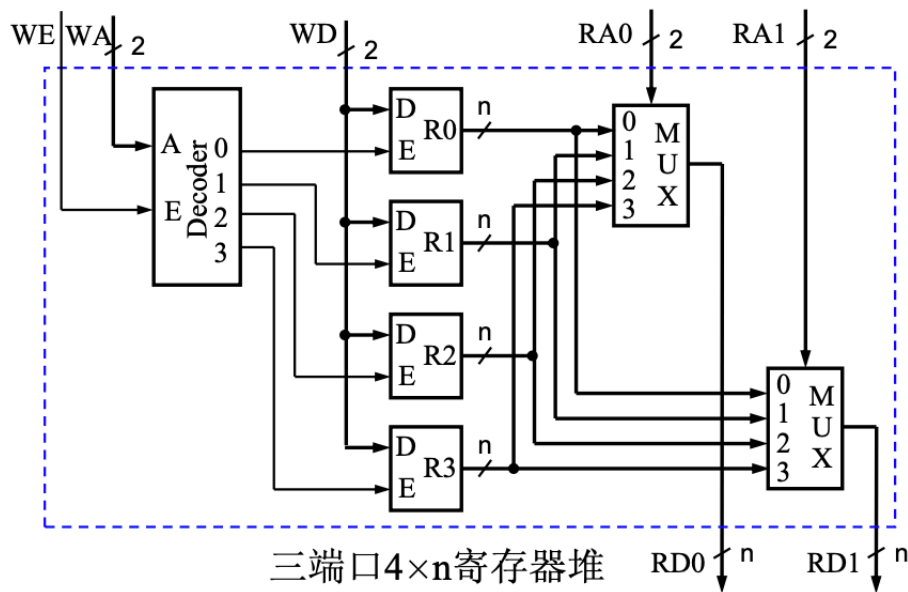
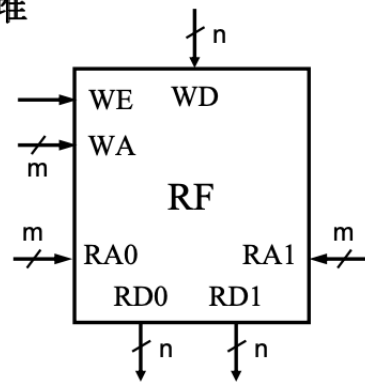
例如，三端口的 $2^m \times n$ 位寄存器堆

1个写端口

- WA: 写地址
- WD: 写入数据
- WE: 写使能

2个读端口

- RA0、RA1: 读地址
- RD0、RD1: 读出数据



2. 核心代码

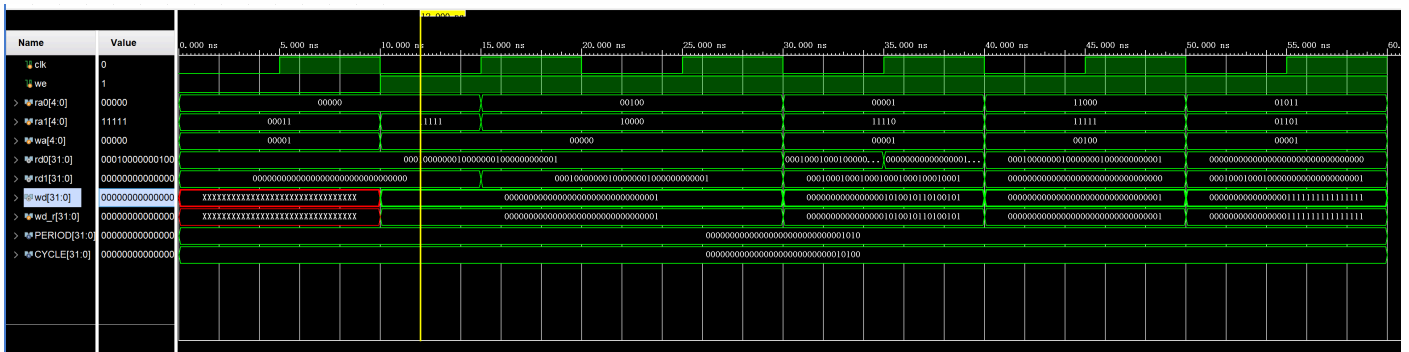
```

1 module register_file
2     #( parameter WIDTH = 32 ) (
3         input clk, //clock
4         input [ 4: 0 ] ra0, //read address 0
5         output [WIDTH - 1: 0] rd0, //read data 0
6         input [4: 0] ra1, //read address 1
7         output [WIDTH - 1: 0] rd1, //read data 1
8         input [4: 0] wa, //write address
9         input we, //write enable
10        input [WIDTH - 1: 0] wd //write data
11    );
12    reg [WIDTH - 1: 0] mem[0: WIDTH - 1];
13    initial
14        $readmemh( "C:/COD/lab2/lab2.srscs/sources_1/new/regfile_init.txt", mem );
15
16    assign rd0 = mem[ra0];
17    assign rd1 = mem[ra1];
18
19    always @( posedge clk )

```

```
20 begin
21     if ( we )
22         begin
23             mem[ wa ] <= wd;
24         end
25     end
26 endmodule
27
```

3. 仿真



IP 例化分布式和块式 16 x 8 位单端口 RAM

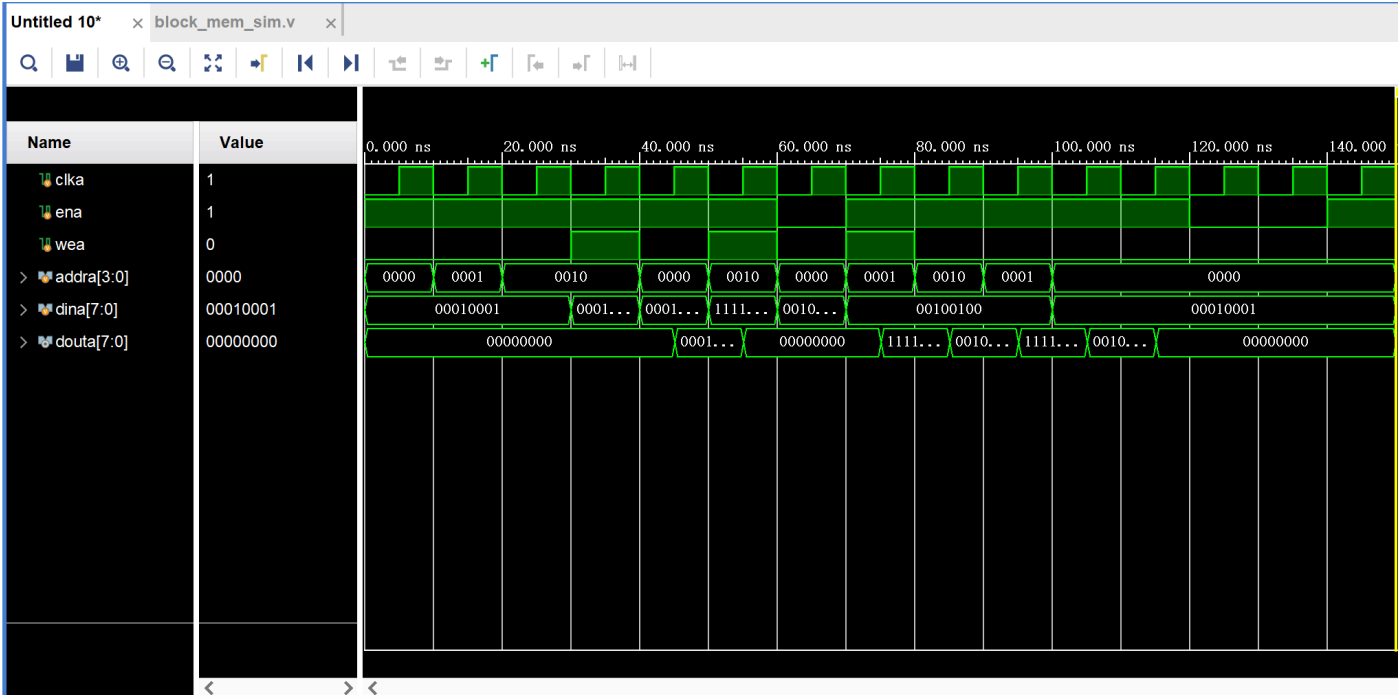
1. 逻辑设计

利用 IP 核中的 **Block Memory Generator** 以及 **Distributed Memory Generator**，生成 16×8 位块式存储器 和 16×8 位分布式存储器。

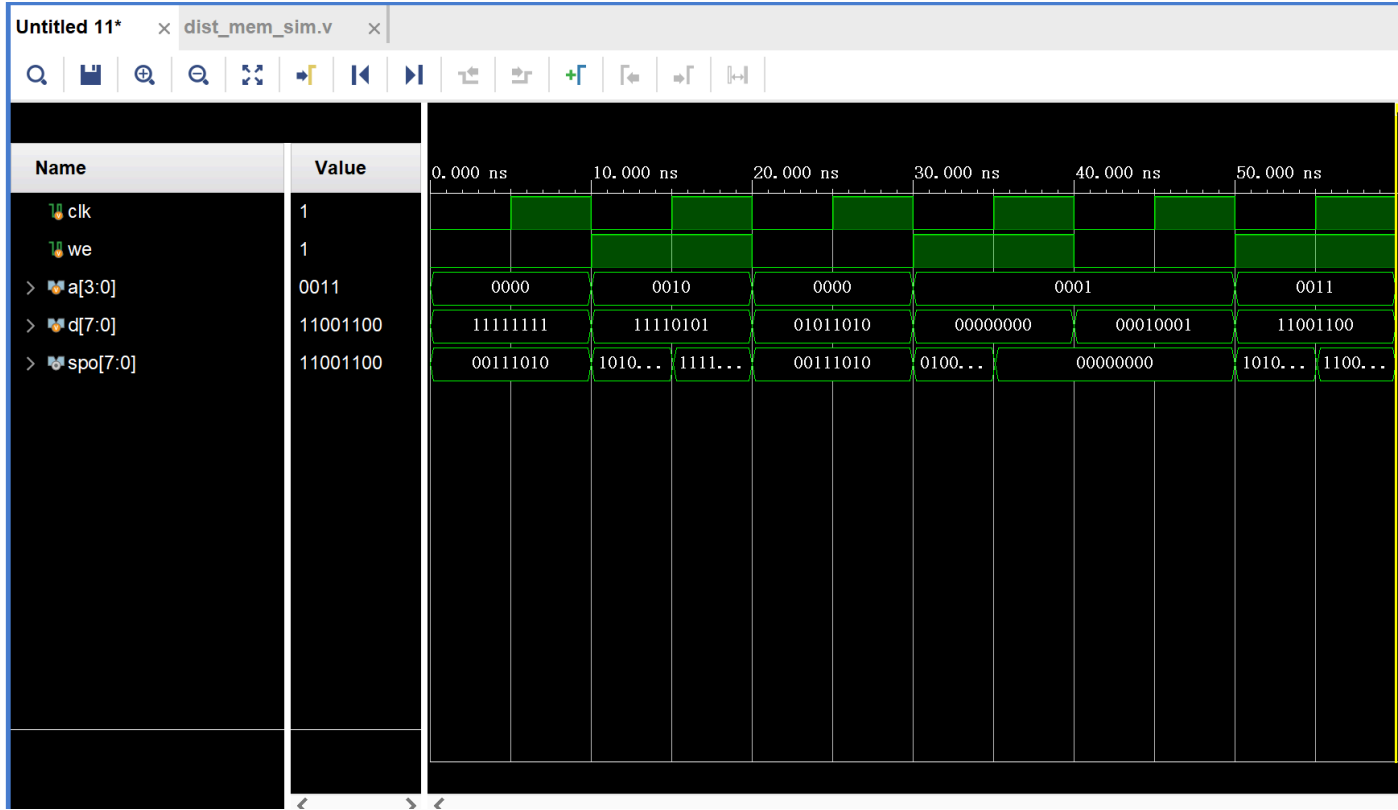
2. 例化端口

```
1 blk_mem_gen_0 ram_16x8 (
2     .clka(clka),    // input wire clka
3     .ena(ena),      // input wire ena
4     .wea(wea),      // input wire [0 : 0] wea
5     .addra(addra),  // input wire [3 : 0] addra
6     .dina(dina),    // input wire [7 : 0] dina
7     .douta(douta)   // output wire [7 : 0] douta
8 );
```

3. 块式仿真



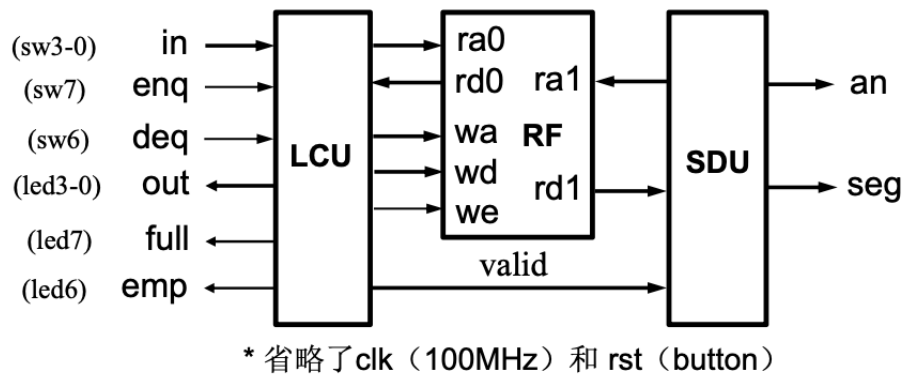
4. 分布式仿真



FIFO 队列

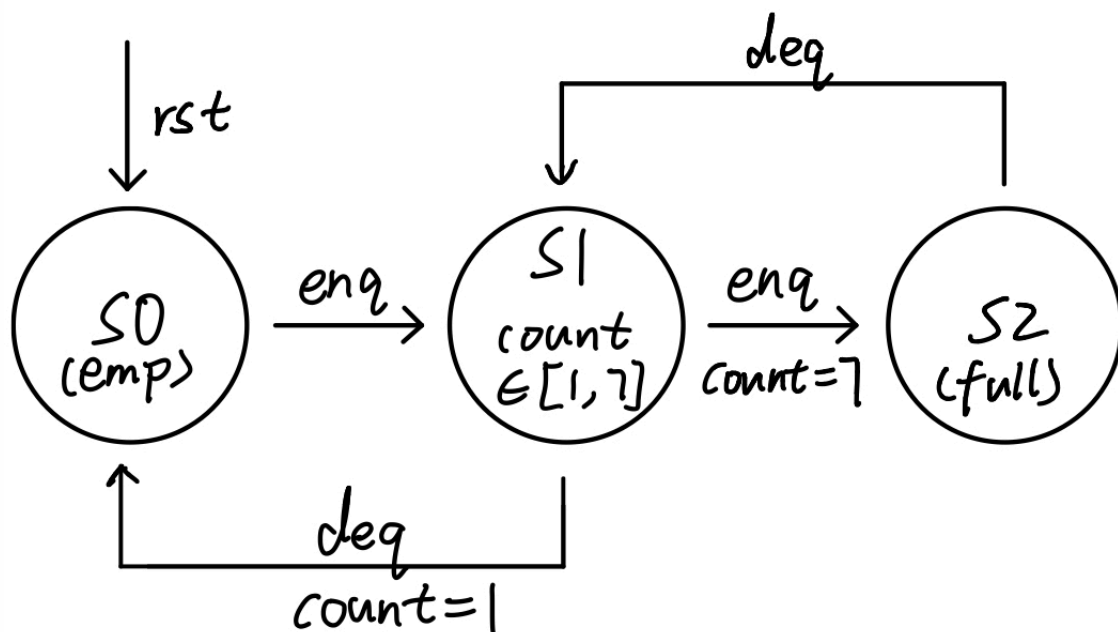
1.逻辑设计

利用三端口 8×4 寄存器堆和适当的逻辑电路，设计实现数据宽度为 4 位、最大长度为 8 的 FIFO 队列，其逻辑符号及数据通路如下图所示。



入队列使能(enq)有效时，将输入数据(in)加入队尾；出队列使能(deq)有效时，将队列头数据输出(out)。队空(emp)和队满(full)用于指示队列是否为空或已满。当队列满时不能执行入队操作，队列空时不能进行出队操作。在入队使能信号的一次有效持续期间，仅允许最多入队一个数据，出队操作类似。

利用板载数码管，还可以显示出队列中数据元素的数值。



2.核心代码

■ 取信号边沿

```
1  module SEDG(  
2      input a,  
3      input clk,  
4      output reg s,  
5      output wire p  
6  );  
7  reg st,pt;  
8  always@(posedge clk)begin  
9      if(a)begin  
10         st <= 1;  
11     end  
12     else begin  
13         st <= 0;  
14     end  
15 end  
16 always@(posedge clk)begin  
17     if(st)begin  
18         s <= 1;  
19     end  
20     else begin  
21         s <= 0;  
22     end  
23 end  
24 always@(posedge clk)begin  
25     if(s)begin  
26         pt <= 1;  
27     end  
28     else begin  
29         pt <= 0;  
30     end  
31 end  
32 assign p = (~pt) & s;  
33 endmodule
```

■ 队列控制单元 LCU

```
1  always @( posedge clk )  
2  begin  
3      if(rst)begin  
4          valid <= 8'b0;  
5          head <= 3'b0;
```

```

6      tail <= 3'b0;
7      count <= 4'd0;
8      out <= 4'd0;
9  end
10 else if(enq)begin
11     if(~full)begin
12         valid[tail] <= 1;
13         tail <= tail + 3'd1;
14         count <= count + 1;
15     end else begin end
16 end
17 else if(deq)begin
18     if(~emp)begin
19         valid[head] <= 0;
20         head <= head + 3'd1;
21         out <= rd0;
22         count <= count - 1;
23     end else begin end
24 end else begin end
25 end
26
27 assign full = (count == 4'h8)?1:0;
28 assign emp = (count == 4'd0)?1:0;

```

■ 数码管显示单元 SDU

```

1  assign x0 = 4'h0;
2  assign an = ra;
3  assign ra = count[15:13];
4  always@(posedge clk)
5  begin
6      if(rst)begin
7          count <= 24'd0;
8      end else begin
9          count <= count + 1;
10     end
11 end
12 assign seg=valid[ra]?rd:x0;

```

3.下载

- 初始队空 (emp 亮起)

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

FPGAOL uart beta 1.0

>

uart pins: cts rts rxd txd

xdc,ucf sym: D3 E5 D4 C4

baud rate: 115200

segplay(sharing with led)

hexplay

-

0

0

0

0

0

0

0

segplay pin: dot seg_gseg_f seg_eseg_dseg_cseg_bseg_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

soft clock

button

None

clk btn pins: clk_btn

xdc,ucf sym: B18

- 3 入队

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

FPGAOL uart beta 1.0

>

uart pins: cts rts rxd txd

xdc,ucf sym: D3 E5 D4 C4

baud rate: 115200

segplay(sharing with led)

hexplay

0

0

0

0

0

0

3

segplay pin: dot seg_gseg_f seg_eseg_dseg_cseg_bseg_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

soft clock

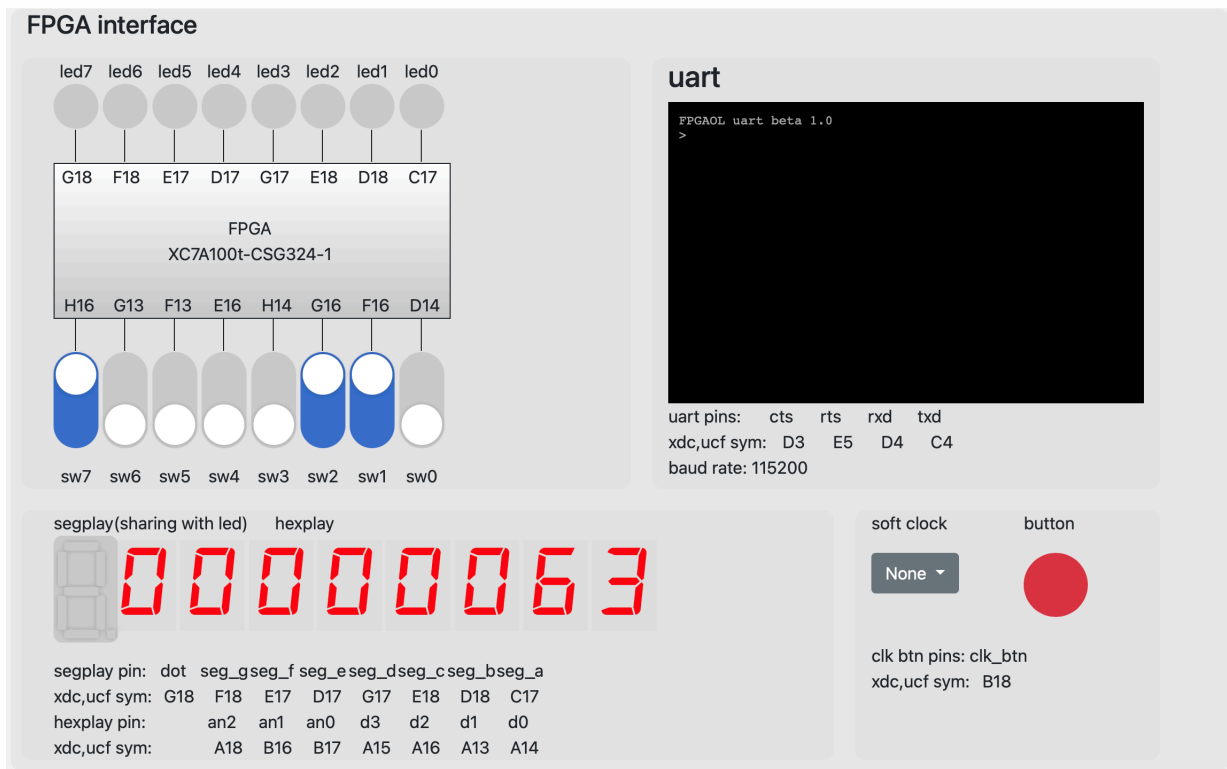
button

None

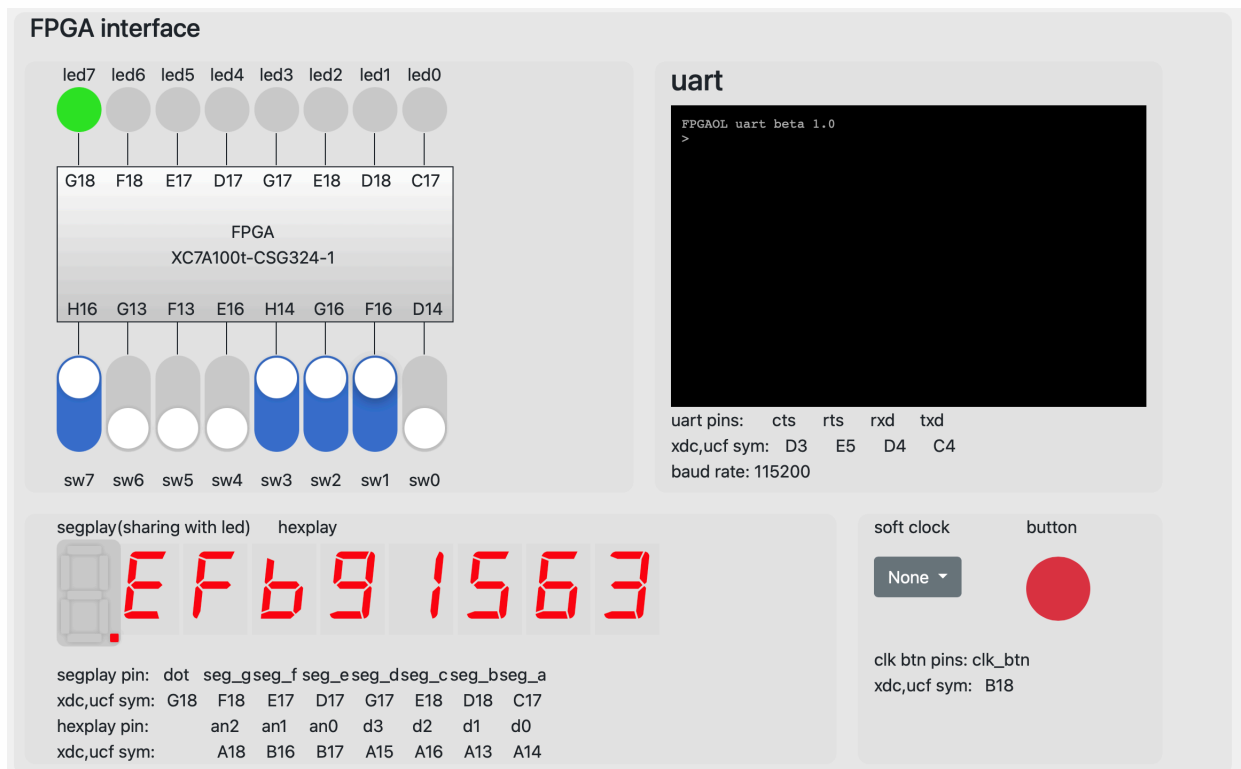
clk btn pins: clk_btn

xdc,ucf sym: B18

- 6 入队



- 5、1、9、11、15、14 依次入队，之后队满 (full 亮起)



- 队首元素 3 出队

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

FPGAOL uart beta 1.0

>

uart pins: cts rts rxd txd

xdc,ucf sym: D3 E5 D4 C4

baud rate: 115200

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: 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

soft clock

button

None

clk btn pins: clk_btn

xdc,ucf sym: B18

- 队首元素 6 出队

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

FPGAOL uart beta 1.0

>

uart pins: cts rts rxd txd

xdc,ucf sym: D3 E5 D4 C4

baud rate: 115200

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: 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

soft clock

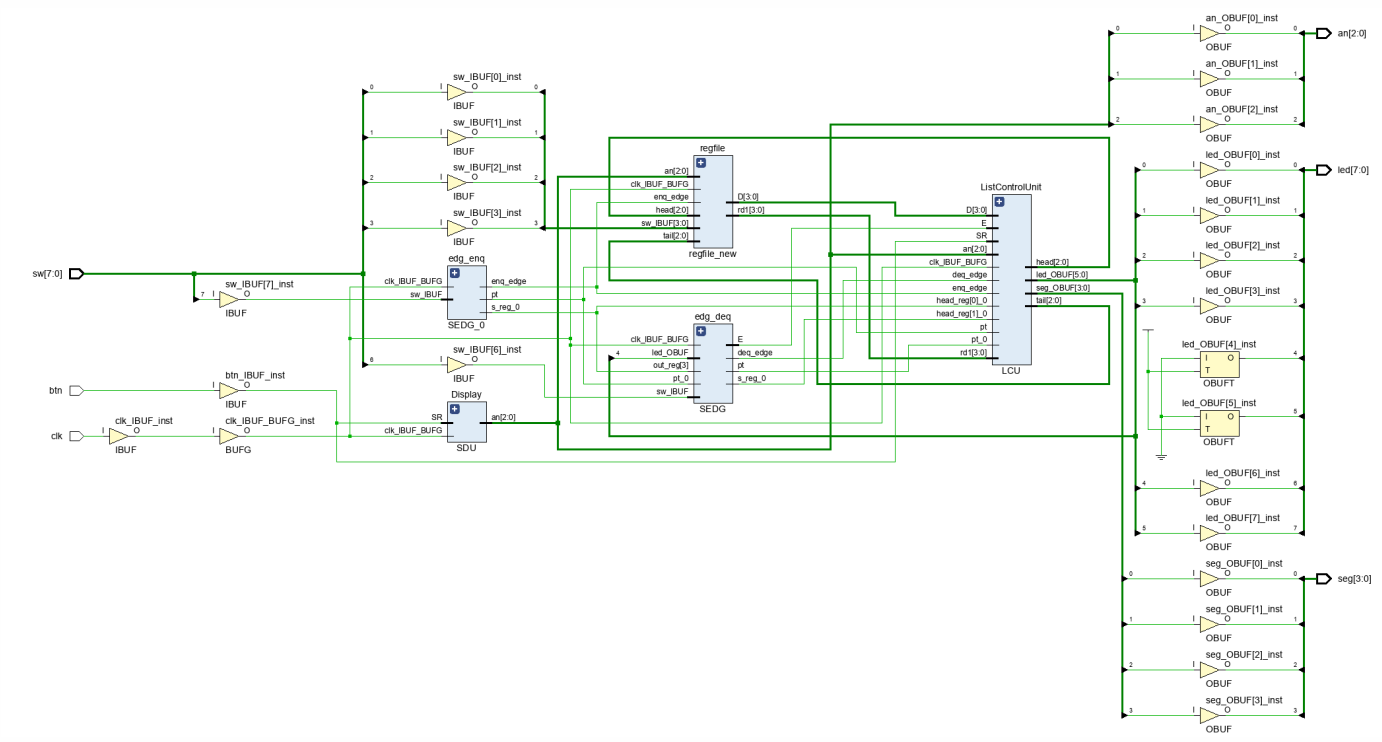
button

None

clk btn pins: clk_btn

xdc,ucf sym: B18

4.电路图



结果分析

寄存器堆、IP 核 RAM 的仿真结果和队列的下载结果符合设计目标

实验总结

本次实验复习了上学期所学的 Verilog 的基本知识，并通过寄存器堆、存储器 RAM 及队列，实现了对组合逻辑电路和时序逻辑电路的巩固，为以后的实验打下基础。