**计算机组成原理 实验二报告**

姓名：曾郅琛 学号：PB20071431 实验日期：2022-3-31

**一、实验题目：**

Lab02 寄存器堆与存储器及其应用

**二、实验目的：**

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

**三、实验平台：**

Vivado

**四、实验过程：**

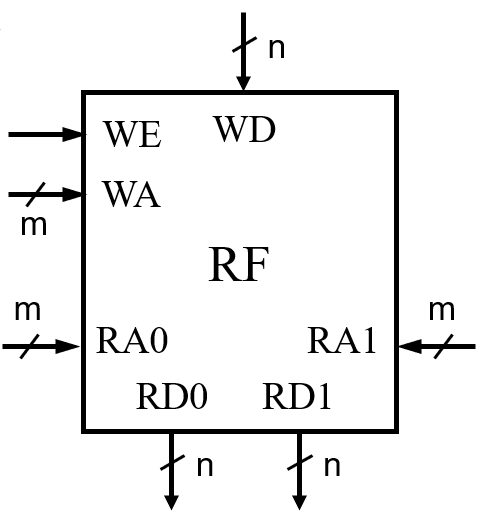
**A. 寄存器堆**

1. 逻辑设计

三端口的2m ╳ 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

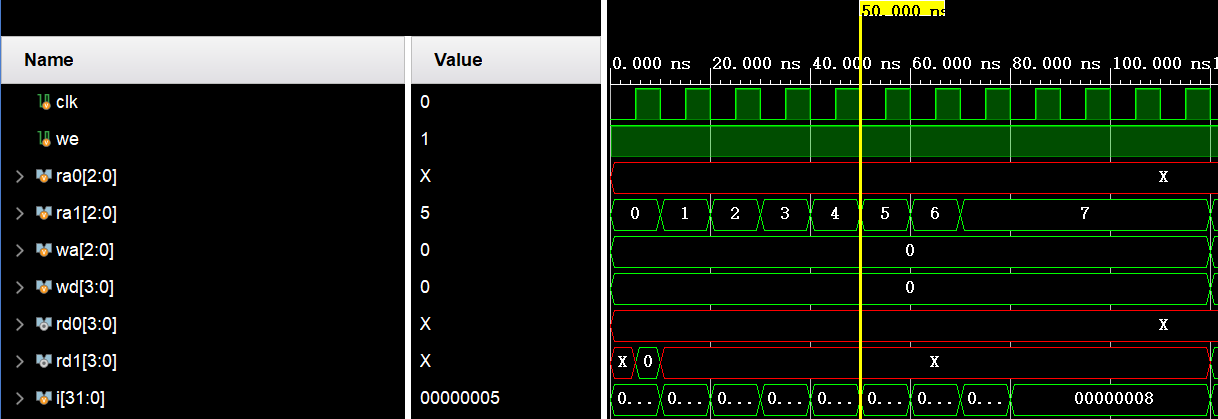


2.代码展示

1. `timescale 1ns/1ns
2. module register\_file
3. #(parameter
4. WIDTH = 32
5. )
6. (
7. input clk,
8. input [2:0] ra0, ra1,
9. output [WIDTH - 1:0] rd0,
10. output [WIDTH - 1:0] rd1,
11. input [2:0] wa,
12. input we,
13. input [WIDTH - 1:0] wd
14. );
15. reg [WIDTH - 1:0] regfile[0：7];
16. assign rd0 = regfile[ra0];
17. assign rd1 = regfile[ra1];
19. always @(posedge clk) begin
20. if (we)
21. regfile[wa] = wd;
22. end
23. endmodule

3.仿真模拟

对寄存器堆进行波形模拟：



**B.存储器IP核**

**1． 分布式**

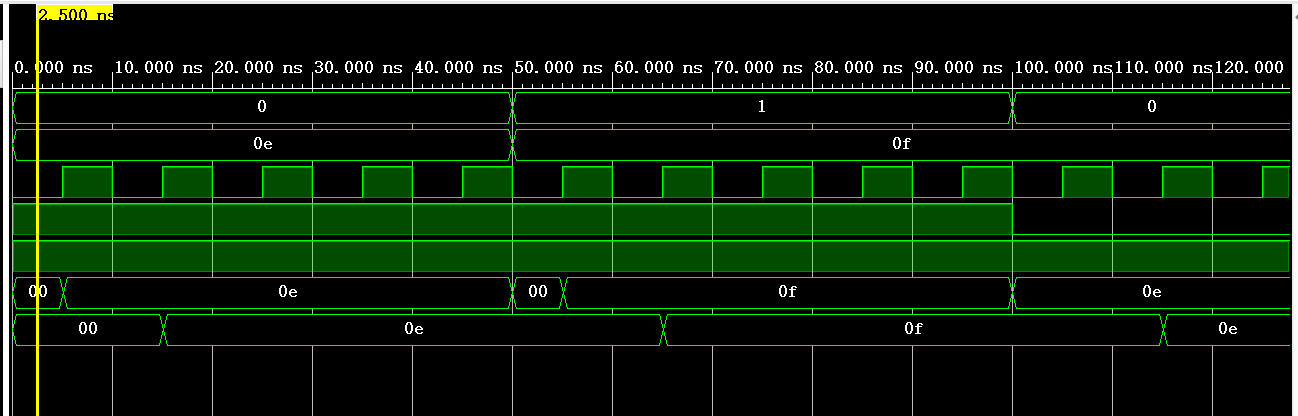
1. dist\_mem\_gen\_0   distributed (
2. .a(a),         *// input wire [15 : 0] a*
3. .d(d),         *// input wire [11 : 0] d*
4. .dpra(dpra),   *// input wire [15 : 0] dpra*
5. .clk(clk),     *// input wire clk*
6. .we(we),       *// input wire we*
7. .dpo(dpo)     *// output wire [11 : 0] dpo*
8. );

**2. 块式**

1. blk\_mem\_gen\_0 block (
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.仿真波形**

将两者波形对比仿真：



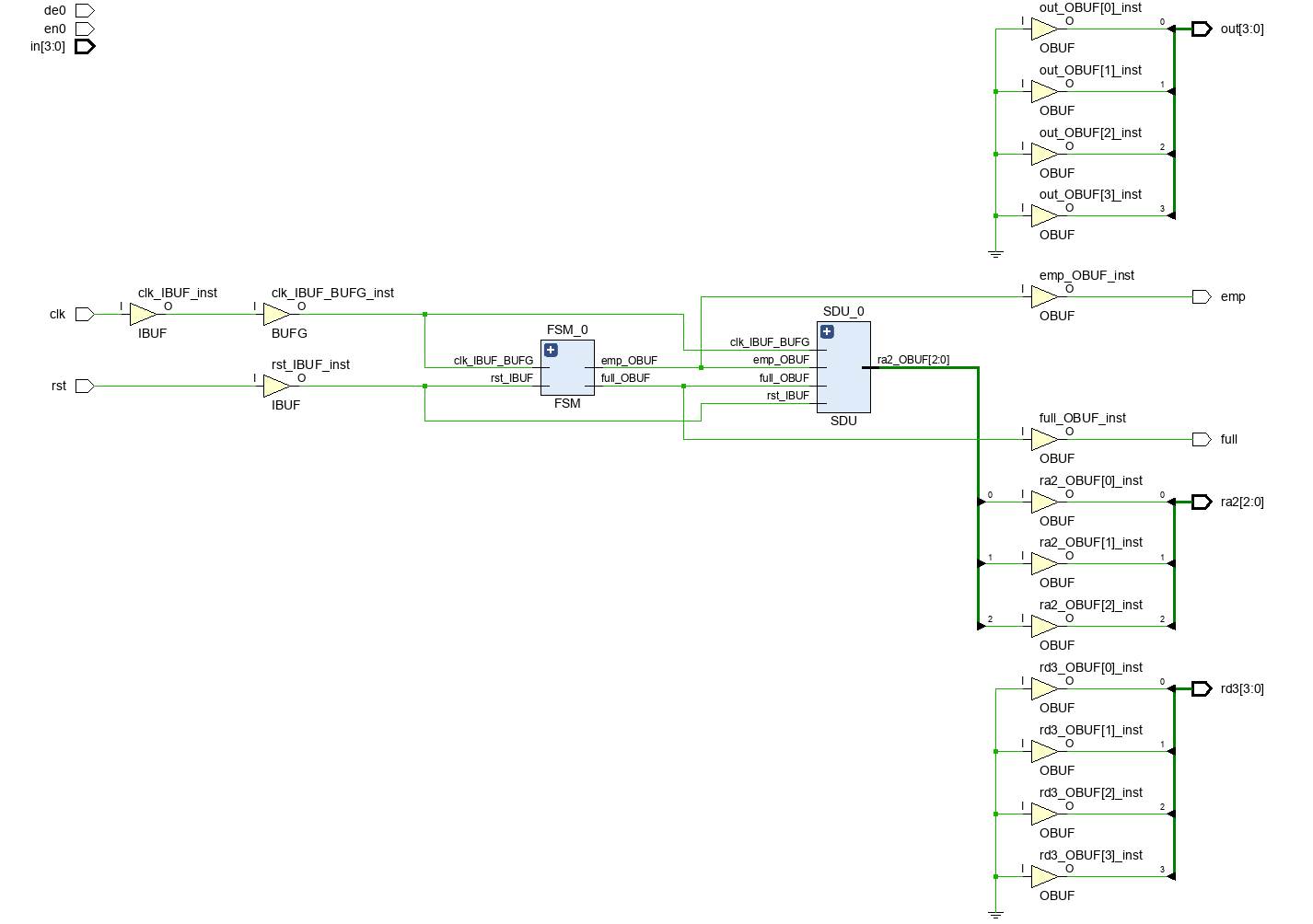
可以看出分布式和块式读写过程大致相同，高电平时，输入的地址被当作写地址，输出的信号为写入的数据；低电平时读写相反。但在读写时间周期上有一定区别。

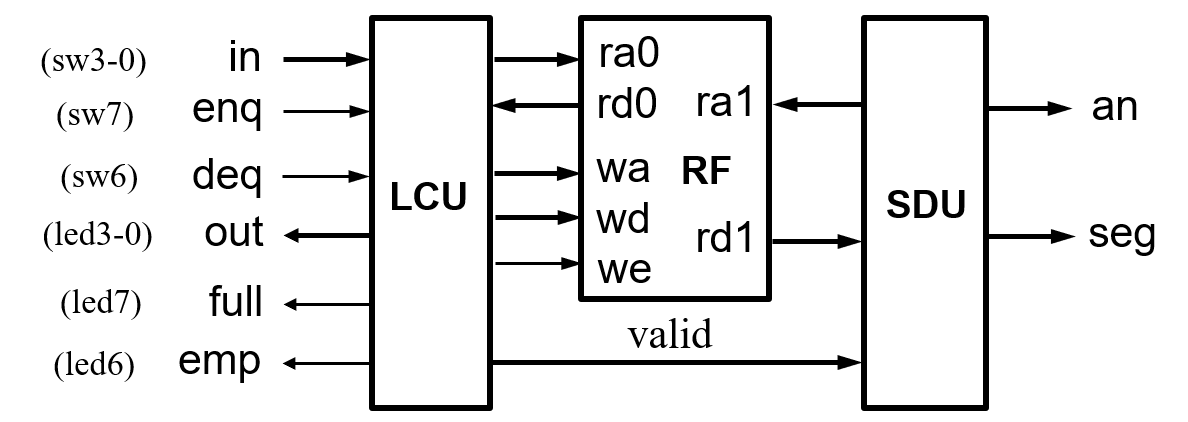
具体如下：

分布式为异步读， 块式为同步读。  
分布式 RAM 和块式都是同步写， 但分布式要快一个时钟周期。

**C.FIFO**

1.数据通路





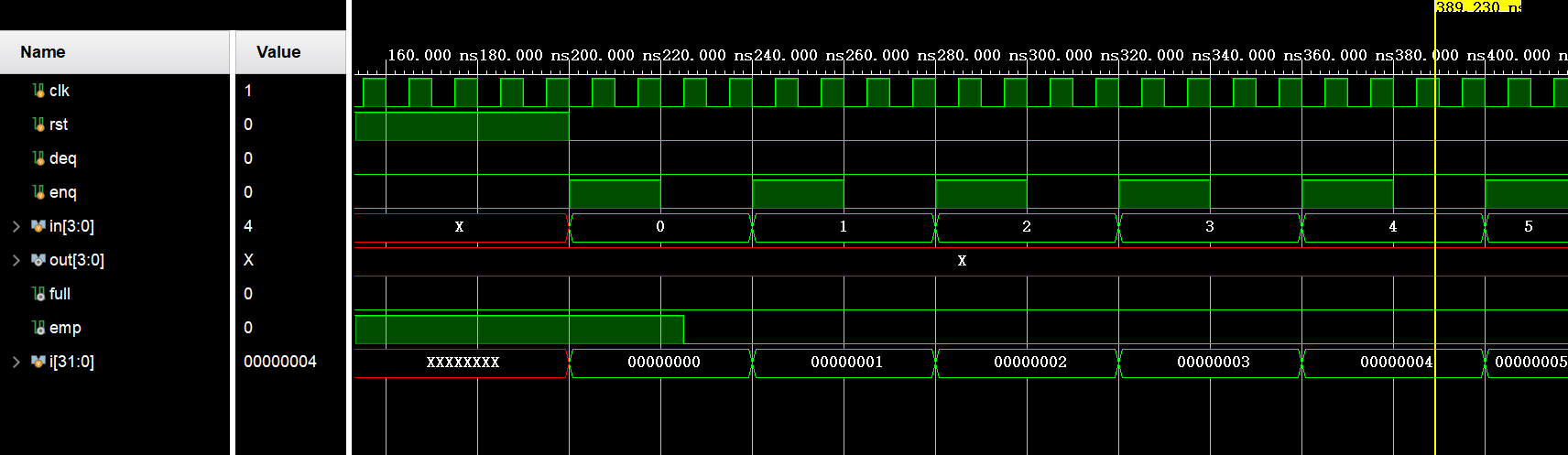
**LCU:**

1. always@(posedge clk or posedge rst) begin
2. if(rst) cs<=0;
3. else cs<=ns;
4. end
5. always@(\*)begin
6. case(cs)
7. s0: begin if(en\_edge) ns<=s1;
8. else ns<=s0;end
9. s1: begin if(en\_edge) begin
10. if(((tail+1)==head)||(tail==7&&head==0)) ns<=s2;
11. else ns<=s1;end
12. else if(de\_edge) begin
13. if(((head+1)==tail)||(head==7&&tail==0))
14. ns<=s0;
15. else ns<=s1;end
16. else ns<=s1;
17. end
18. s2: begin if(de\_edge) ns<=s1;
19. else ns<=s2;end
20. default: ns<=s0;
21. endcase
22. end
23. always@(posedge clk)begin
24. case(cs)
25. s0: begin if(en\_edge)
26. begin tail<=1;head<=0;empty<=0;valid[0]<=1;end
27. else begin tail<=0;head<=0;empty<=1;full<=0;valid<=0;end
28. end
29. s1: begin if(en\_edge) begin tail<=tail+1;head<=head;valid[tail]<=1;end
30. else if(de\_edge) begin head<=head+1;tail<=tail;valid[head]<=0;end
31. end
32. s2: begin if(de\_edge) begin head<=head+1;tail<=tail;full<=0;valid[head]<=0;end
33. else begin head<=head;tail<=tail;full<=1;valid<=valid;end
34. end
35. default: begin head<=head;tail<=tail;valid<=valid;end
36. endcase
37. end

**SDU:**

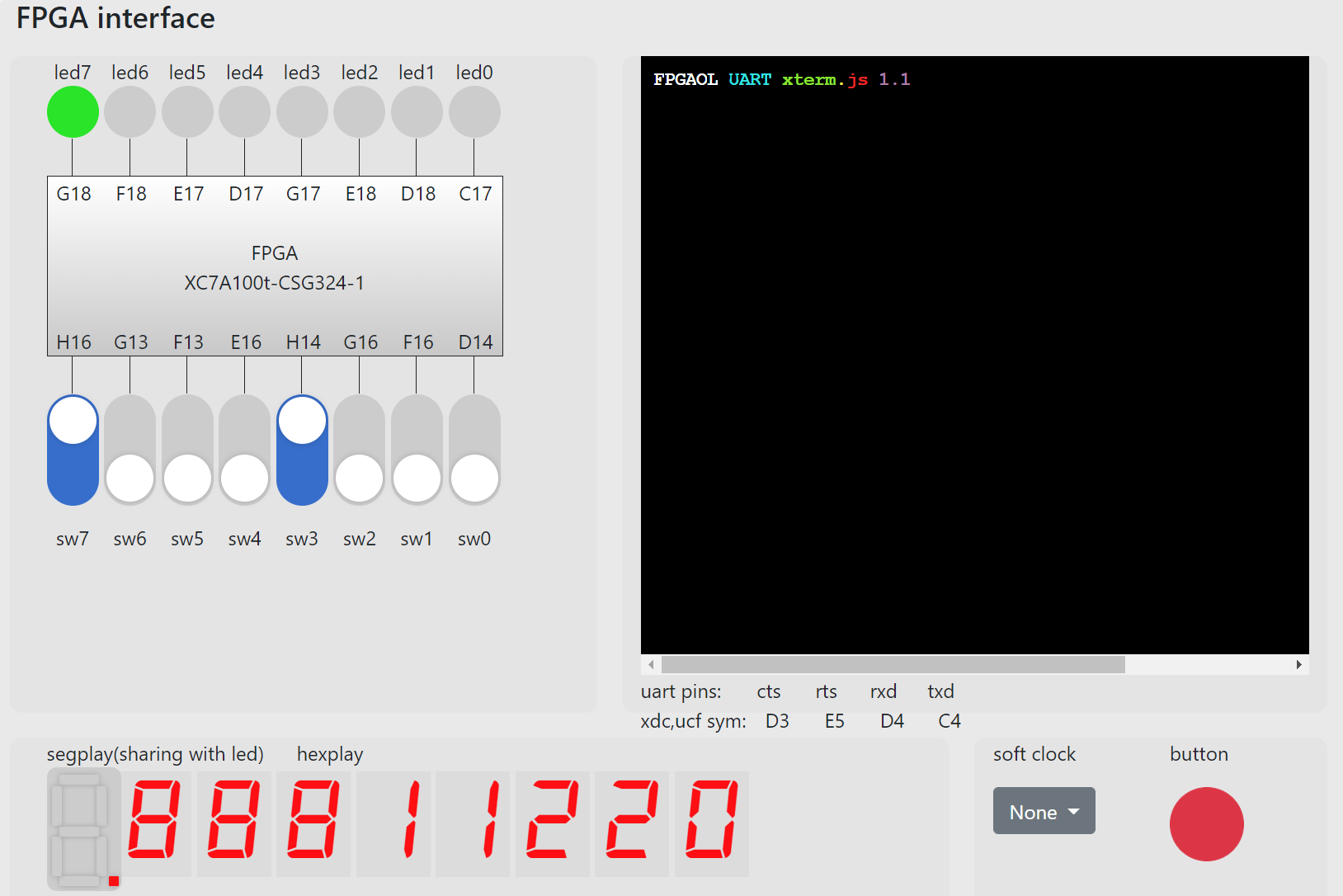
1. initial begin
2. cnt<=0;clk\_cnt<=0;display<=0;
3. end
4. always@(posedge clk or posedge rst)begin
5. if(rst) begin cnt<=0;clk\_cnt<=0;end
6. else begin
7. if(cnt==100000) cnt<=0;
8. else cnt<=cnt+1;
9. if(cnt>100000/2) clk\_cnt<=1;
10. else clk\_cnt<=0;
11. end
12. end
13. always@(posedge clk\_cnt or posedge rst)begin
14. if(rst) display<=0;
15. else display<=display+1;
16. end
17. assign ra1=display;
18. always@(posedge clk)begin
19. if(valid==0) begin
20. seg<=0;an<=0;
21. end
22. else if(valid[display])begin
23. seg<=rd1;an<=display;
24. end
25. End

**3.波形仿真**

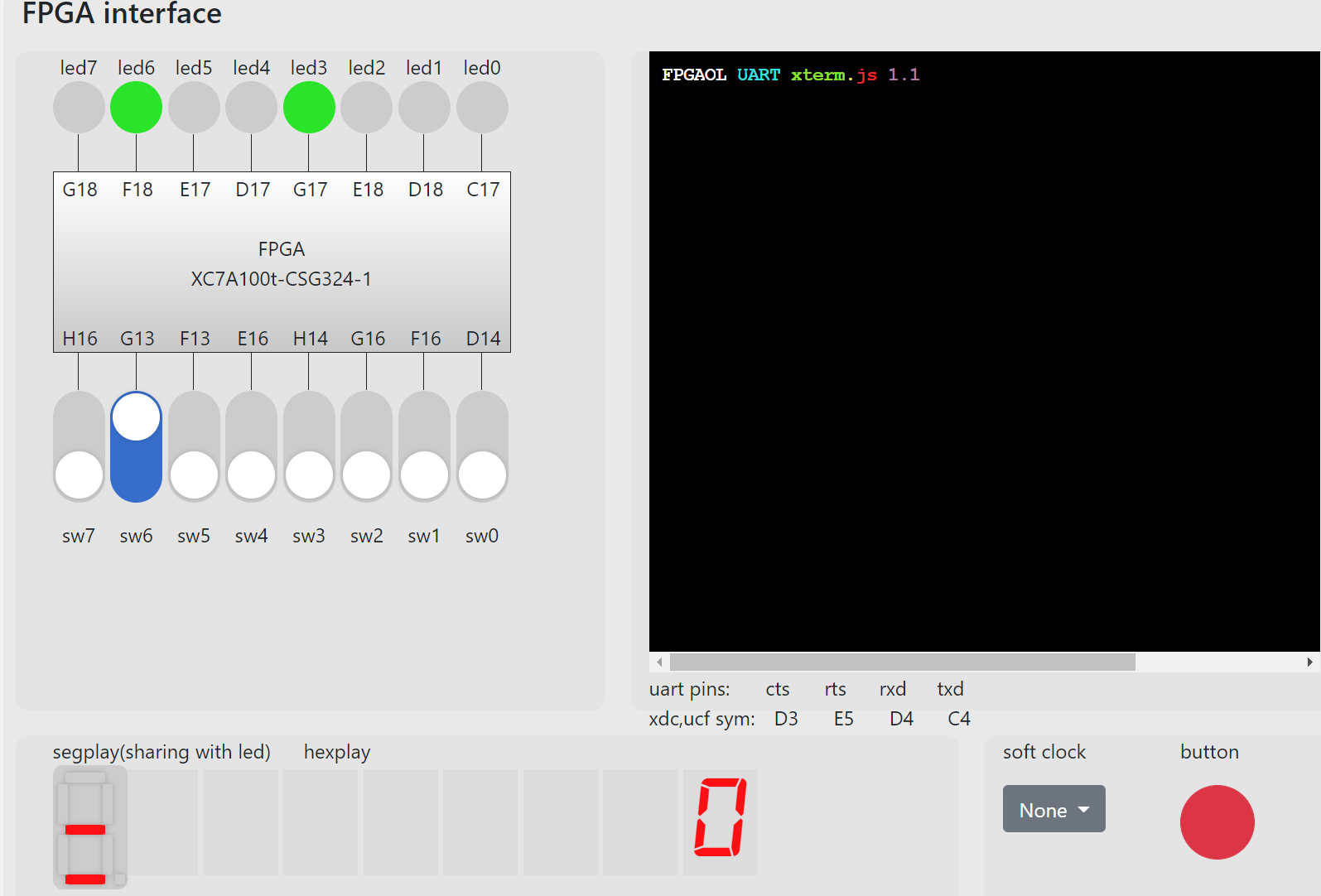


**4.FPGA烧板**

**进队到满：**



**出队到空：**



**六、心得体会：**

本次实验完成寄存器堆比较容易解决，IP核那块做了很久，因为不太理解这种存储，还有vivado第一次用这个，FIFO代码段也很难写，后来问同学还有参考其他组助教给同学的讲解最后才算完成。