数逻实验报告 Lab13

雷远航

December 15, 2022

Abstract

移位寄存器设计与应用

一、操作方法与实验步骤

任务 1: 设计 8 位带并行输入的右移移位寄存器

sift_reg8b.v

利用 verilog 完成能够并行输入的右移移位寄存器.
完成的代码如下:

```
1  `timescale 1ns / 1ps
2
3  module shift_reg8b (
4      input clk,
5      input S_L,  //q=0串行,  q=1并行
6      input s_in, //串行输入的数据
7      input [7:0] p_in,
8      output reg[7:0] Q
9  );
10
11 initial begin
12     Q[7:0] <= 8'b1001_1101;
13 end
14
15 always @(posedge clk) begin:run
16     integer  i;
17     if(S_L == 1'b0)  begin
18         /*串行*/
19         for(i=0; i<=6; i=i+1) begin
20             Q[i] = Q[i+1];
21         end
22         Q[7] <= s_in;
23     end
24
25     else  begin
26         /*并行*/
27         Q[7:0] <= p_in[7:0];
28     end
29
30 end
31
32 endmodule
```

当 S_L 为 0 时进行串行输入,s_in 为将要串行输入的数据.
当 S_L 为 1 时进行并行输入,p_in 为并行输入所要输入的数据.

sift_reg8b_tb.v

通过 testbench 得到波形图, 对设计的移位寄存器进行分析, 测试文件如下:

```
`timescale 1ns / 1ps
`include "shift_reg8b.v"

module shift_reg8b_tb;
    reg clk;
    reg S_L;
    reg s_in;
    reg [7:0]  p_in;
    wire [7:0] Q;

    shift_reg8b uut (
        .clk(clk),
        .S_L(S_L),
        .s_in(s_in),
        .p_in(p_in),
        .Q(Q)
    );

    initial begin
    $dumpfile("shift_reg8b_tb.vcd");
    $dumpvars(0, shift_reg8b_tb);
```

```


        clk = 0;
        S_L = 0;
        s_in = 1;
        p_in[7:0] = 8'b0111_1110;
        #10;
        clk = 1;
        #10;
repeat(5) begin
        clk = 0;
        S_L = 0;
        s_in = 1;
        #10;
        clk = 1;
        #10;
end

        clk = 0;
        S_L = 1;
        #10;
        clk = 1;
        #10;
repeat(3) begin
        clk = 0;
        s_in = 0;
        S_L = 0;
        #10;
        clk = 1;
        #10;
end

   end
```

任务二: 完成 P2S 模块

完善 P2S.v 的内容

将空缺的 S_R 锁存器的部分补充完整

```
    input start,
    input[6:0] PData,
    input clk,
    output sclk,
    output sclrn,
    output sout,
    output EN
);
    wire[7:0] Q;
    wire[7:0] D;
    wire finish;
    wire q;
    wire nq;
    wire set;
    wire reset;

    assign D = {1'b0, PData};
    assign set = start & finish;
    assign reset = ~finish;
    assign q = (set | q) & (~reset);
    assign nq = (reset | nq) & (~set);



    shift_reg8b m0 (.clk(clk), .S_L(q), .s_in(1'b1), .p_in(D), .Q(Q));

    assign finish = &Q[7: 1];
    assign sout = Q[0];
    assign sclrn = 1'b1;

    assign EN = start & finish;
    assign sclk = finish | clk;
```

d 对 P2S 模块进行测试

测试代码如下:

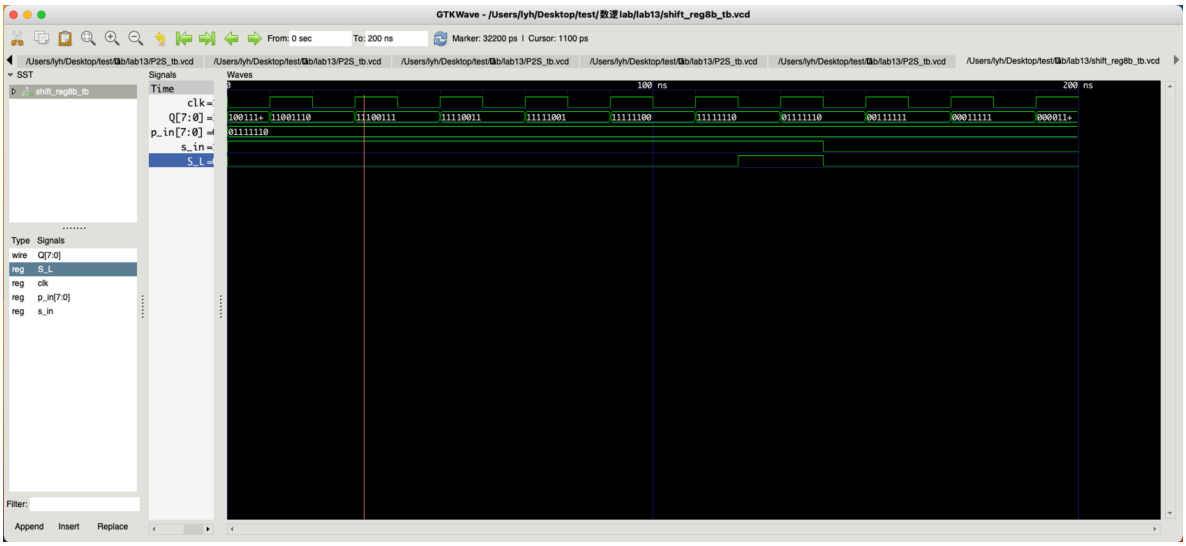
```
1   `timescale 1ns / 1ps
2   `include "P2S.v"
3
4   module P2S_tb;
5
6
7   reg clk;
8   reg [6:0] PData;
9   reg start;
10
11  wire sclk;
12  wire sclrn;
13  wire sout;
14  wire EN;
15
16  P2S uut (.clk(clk),.start(start), .PData(PData), .sclk(sclk), .sclrn
17
18
19  initial begin
20      $dumpfile("P2S_tb.vcd");
21      $dumpvars(0, P2S_tb);
22      start = 0;
23      PData[6:0] = 7'b1010101;
24      repeat(10) begin
25          clk = 0;
26          #5;
27          clk = 1;
28          # 5;
29      end
30      clk=0;
31      start = 1;
32      #5;
```

```
33      clk = 1;
34      start = 0;
35      #5;
36
37      repeat(4) begin
38      clk = 0;
39      #5;
40      clk = 1;
41      #5;
42      end
43
44      clk = 0;
45      start = 1;
46      PData[6:0] = 7'b0010010;
47      #5;
48      clk = 1;
49      start = 0;
50      #5;
51
52      repeat(4) begin
53      clk = 0;
54      #5;
55      clk = 1;
56      #5;
57      end
58
59
60  end
61
62
63  endmodule
```

二、实验结果与分析

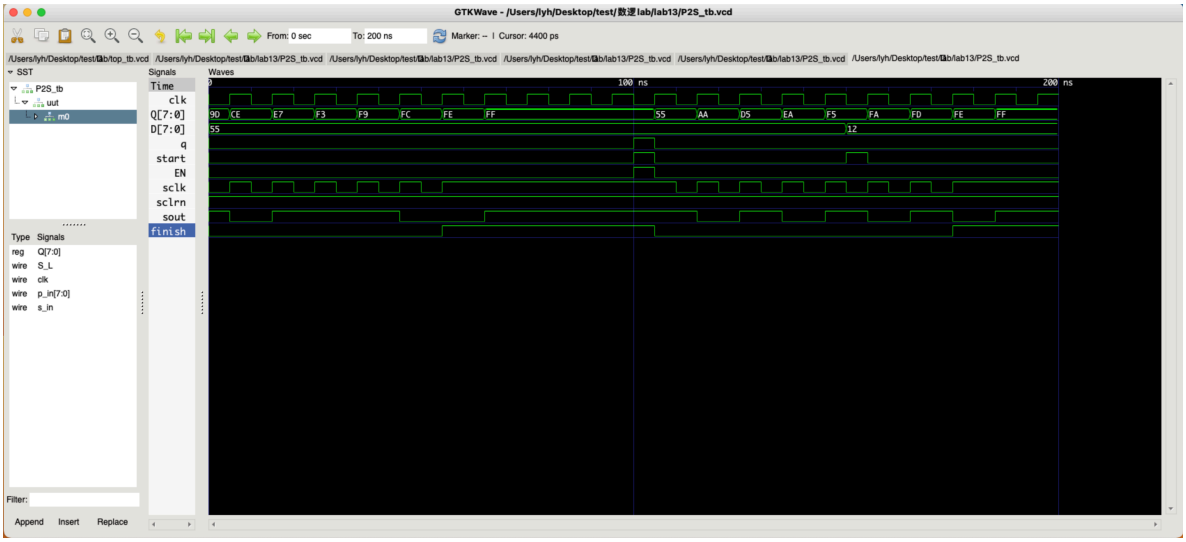
移位寄存器波形图分析:

通过测试文件得到波形图, 测试的结果如下:

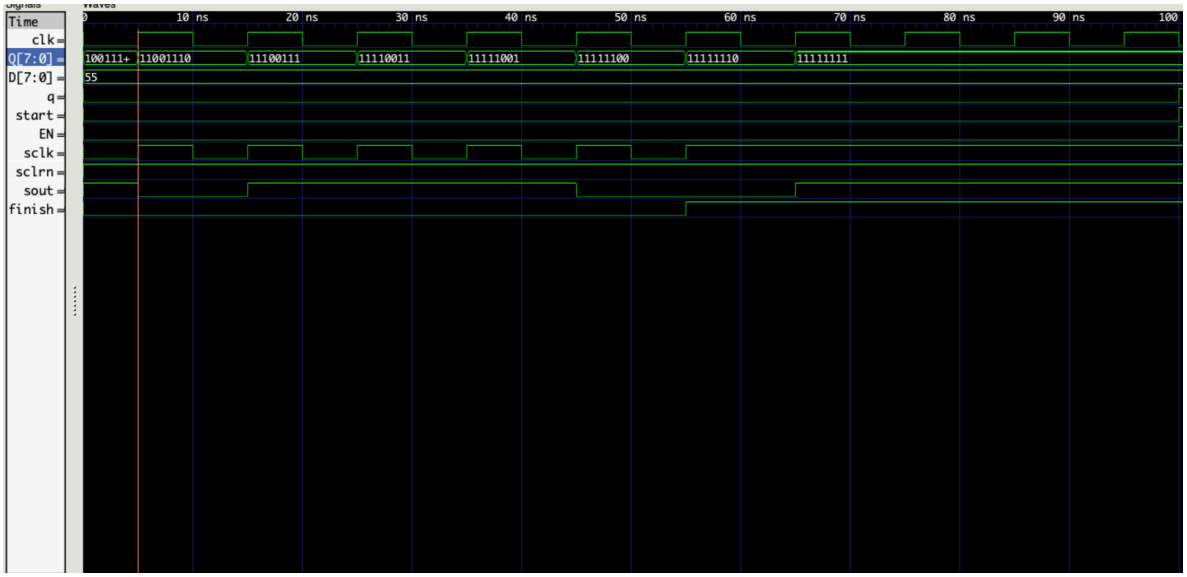


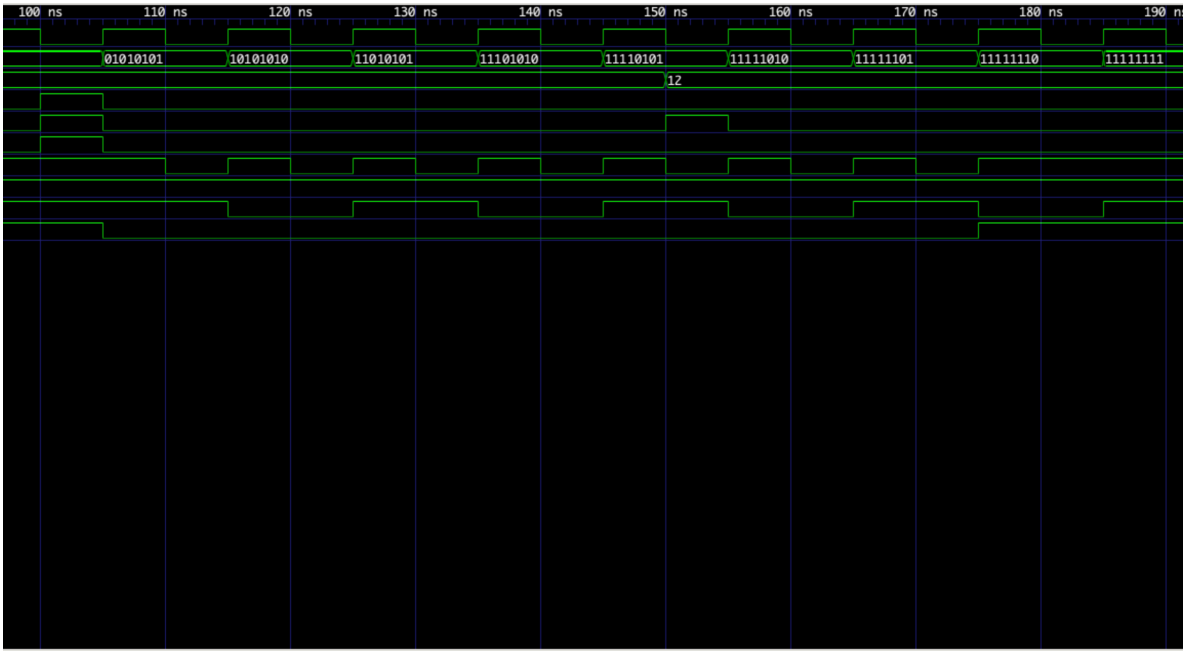
在前六个时钟周期下,S_L 的信号为 0, 进行串行输入的操作, 并且每次串行操作补充的数为 1, 输出的结果进行了 6 次右移, 接下来的时钟周期下 S_L 的信号为 1, 进行并行输入的操作, 此时 p_in 为将要串行输入的数据, 在波形图中可以看出正确进行了传输, 接下来后面的时钟周期进行串行输入, 并且每次串行输入的数据为 0.

P2S 模块的波形分析



二进制形式的输出结果:





在前 7 个时钟周期里先进行的串行输入的操作, 由于载入数据的信号为 0, 并且由 &Q[7:1], 控制的 finish 信号也为 0, 因此 P2S 模块中的 S_R 锁存器的 reset 信号为 1,set 信号为 0, 因此输出的 q 信号为 0, 进行的是串行输入的模式, 并且每次补充的数据都是 1, 当信号不断移位变成 FE 时, 此时的 finish 信号为 1, 但 set 信号和 reset 信号都为 0, 仍然持续进行移位的操作输出的信号为 FF, 在其后面的时钟周期里,start 信号为 1, 代表已经准备好了要输入的数据, 并且 finish 信号也为 1, 所以 set 信号为 1,p 信号此时为 1, 进行串行载入的模式, 可以看到并行的数据被载入到了 Q 中, 在后面的时钟周期中也出现了一个 start 为 1 的状态, 但是此时的 finish 信号为 0 因此不会进行并行载入的模式, 还时持续进行串行输入.

其他输出信号都按照 P2S 模块中的逻辑进行正常的输出.

三、讨论与心得

由于疫情的影响本次实验都通过模拟仿真来进行, 实验中完成了移位寄存器和 P2S 模块, 并且进行了模块的仿真,P2S 模块的一些内容比较复杂, 但在助教讲解之后, 对 P2S 模块的功能的理解也比较顺利, 整体的完整较为轻松.