数电实验EXP6

刘永鹏 191220070 693901492@qq.com 9.30

1. 实验内容

移位寄存器

请根据下表，用 Verilog HDL 语言设计一个移位寄存器，并进行仿真查看 移位寄存器的功能。对于移位寄存器的实现细节，请自行复习数电教科书 8.5 节内容，参考 8.5.9 节内容实现。

其中左端串行输入 1 位数值，并行输出 8 位数值是指每个时钟到来时右移 一位，并且移入的最左位由外部开关决定是 1 还是 0，输出同其他情况一样为 同时输出 8 位。这个功能在进行串行转换为并行时比较有用，可以将时间上顺 序输入的 8 个 bit 存入移位寄存器，在 8 个周期后形成一个 8bit 数一起输出。后 续键盘串行输入可以利用这个功能。

控制位	工作方式
000	清 0
001	置数
010	逻辑右移
011	逻辑左移
100	算术右移
101	左端串行输入 1 位值，并行输出 8 位值
110	循环右移
111	循环左移

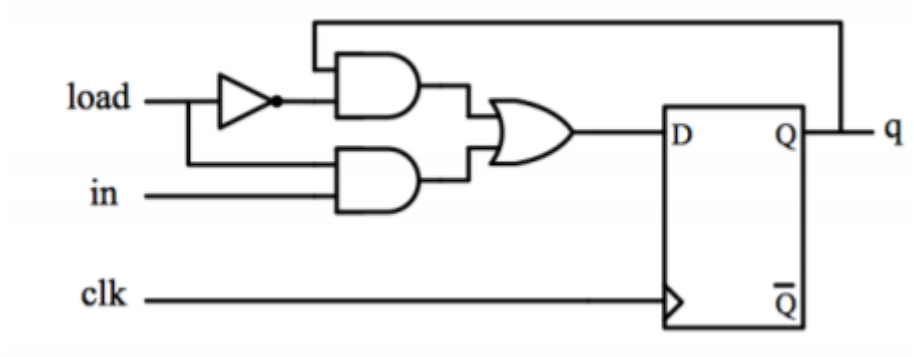
随机数发生器

我们可以利用 8 位移位寄存器来实现一个简单的随机数发生器。参考教科 书第 534 页 LFSR 反馈方程设计一个 n=8，共有 255 种状态的随机数发生器。 请将 8 位二进制数以十六进制显示在数码管上，在 DE10-Standard 开发板上观 察生成的随机数序列。系统需要能够自启动。

2. 实验原理

1. 寄存器也是最常用的时序逻辑电路，是一种存储电路。寄存器的存储电路**是由锁存器或触发器构成的**，因为一个锁存器或触发器能存储 1 位二进制数， 所以由 N 个锁存器或触发器可以构成 N 位寄存器。

2. 简单的寄存器 = **D触发器 + 置数功能。**



3. 寄存器通过串接可以构成多位寄存器。
4. **移位寄存器**则可以在每个时钟沿将存储 在其中的数据向某个方向移动一位。**串行的多个D触发器**可以构成简单的移位寄存器。
5. 在Quartus中，可以非常方便地描述移位寄存器。
6. 对于**随机数生成器**，结合LFSR计数器，可以比较方便地完成随机生成的模拟，采用下列输出方程：

$$X8 = X4 \oplus X3 \oplus X2 \oplus X0 \tag{1}$$

来作为移位输入。

3. 实验环境

1. Quartus环境，手动建立项目。
2. 对于EXP6-1，使用开关（SWITCH）作为数据输入，用按键（KEY）作为模式调节。
3. 对于EXP6-2，仅需要开关即可。

4. 实验代码/实验截图

EXP6-1 寄存器

寄存器的输入输出信号与内部信号。

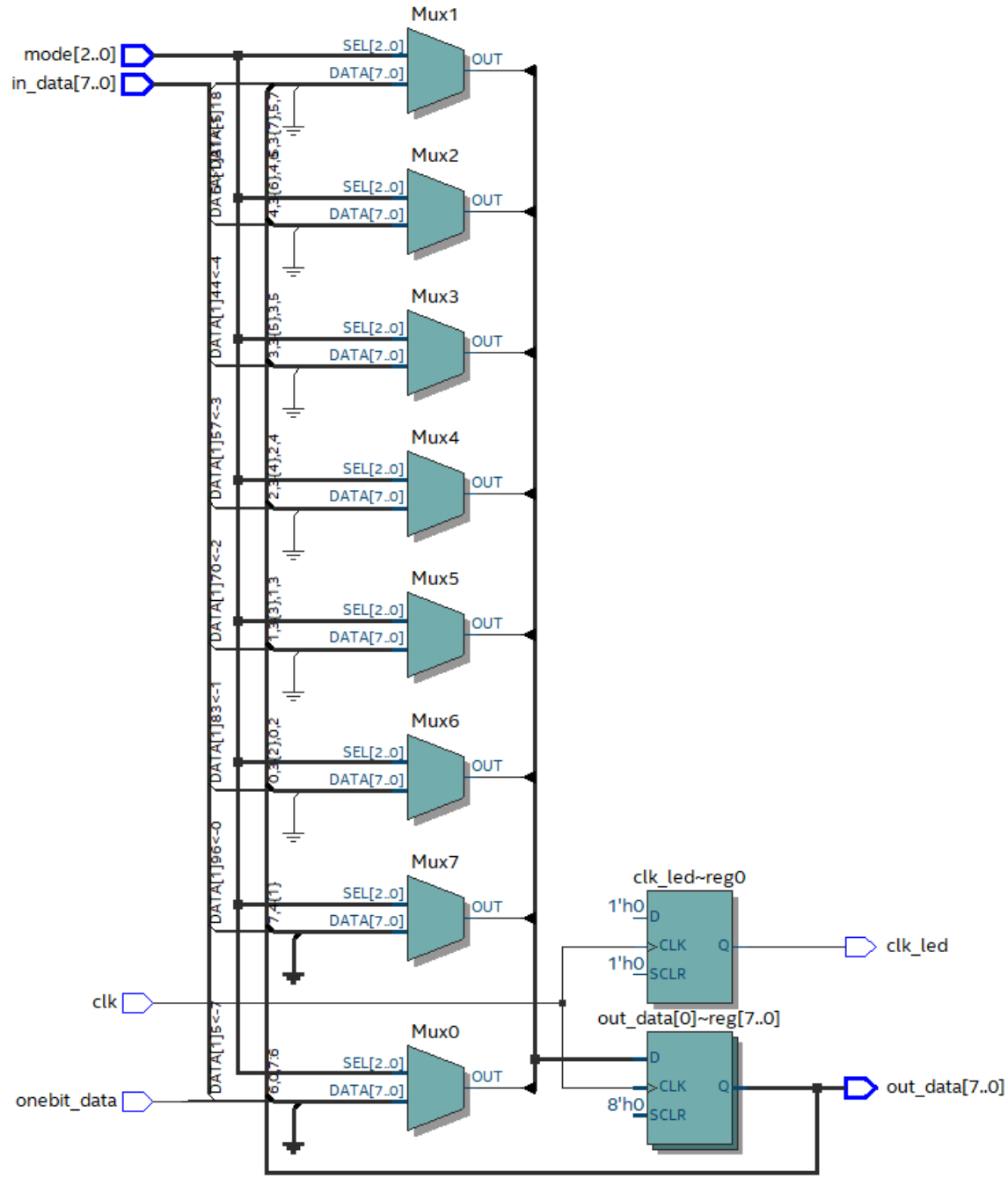
```
module Register(
    input clk, //key[3]
    input [2:0]mode, //key[0],key[1],key[2]
    input [7:0]in_data, //SW[7:0]
    input onebit_data, //SW[8]
    output reg[7:0]out_data, //LEDR[7:0]
    output reg clk_led //SW[9]
);
//inner signal
reg clk_t;
reg [26:0]count_clk;C
```

逻辑功能实现：

```
always @ (posedge clk)
begin
    clk_led <= clk_t;
    case(mode)
    3'b000:begin //清零
        out_data <= 0;
```

```
    end
    3'b001:begin //置数
        out_data <= in_data;
    end
    3'b010:begin //逻辑右移
        out_data <= {1'b0, out_data[7:1]};
    end
    3'b011:begin //逻辑左移
        out_data <= {out_data[6:0], 1'b0};
    end
    3'b100:begin //算术右移
        out_data <= {out_data[7], out_data[7:1]};
    end
    3'b101:begin //左端串行输入，并行输出8位
        out_data <= {onebit_data, out_data[7:1]};
    end
    3'b110:begin //循环右移
        out_data <= {out_data[0], out_data[7:1]};
    end
    3'b111:begin //循环左移
        out_data <= {out_data[6:0], out_data[7]};
    end
    endcase
end
```

RTL门电路视图：



EXP6-2 随机数生成器

输入输出信号/内部信号

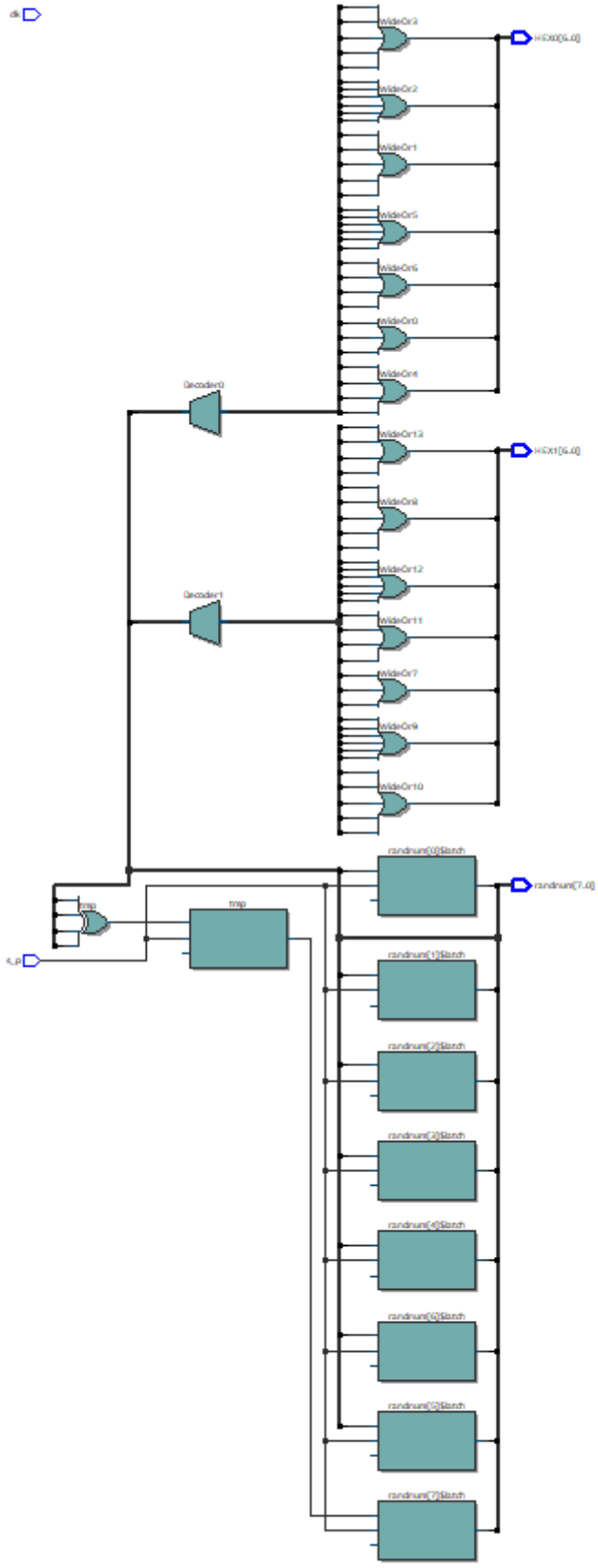
```
module RandomNum(
    input s_p,
    input clk,
    output reg [7:0] randnum,
    output reg [6:0] HEX0,
    output reg [6:0] HEX1
);
//inner signal
reg tmp;
```

逻辑功能实现

```
always @ (clk)
begin
```

```
    if(s_p)//生成随机数中
    begin
        if(randnum == 0)
        begin
            randnum <= 10;
        end
        tmp <= randnum[4] ^ randnum[3] ^ randnum[2] ^ randnum[0];
        randnum <= {tmp, randnum[7:1]};//待优化的式子
    end
    else
    begin
        randnum <= randnum;
        tmp <= tmp;
    end
end
```

RTL门电路视图



5. 实验过程

1. 做出初步设计。
2. Quartus编写代码。
3. 没有采用testbench仿真测试（不太需要）。

4. 烧板测试。

6. 实验结果

已完成验收。

7. 问题与解决方案

1. 采用了书上LFPR移位寄存器的现成方程，使用条件语句将0包含在序列中。
2. 无其他问题。