实验六　寄存器

2020 年秋季学期

A device with two stable positions, such as a relay or a flip-flop circuit, can store one bit of information.

– "A Mathematical Theory of Communication", C. E. Shannon

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。

本实验通过介绍几中常用寄存器的设计方法，复习寄存器的原理，学习寄存器和常用的移位寄存器的设计。

6.1 寄存器

D 触发器可以用于存储比特信号，给 D 触发器加上置数功能就变成了一位寄存器，如图 6-1所示。由图中可以看出，如果 load 信号为 1，则输入信号 in 被送入或门中，或门的另一个输入端为 0，此时 D=in，所以在下一个时钟里 q=in。当 load 值为 0 时，q 值被反馈到或门中，或门的另一个输入值为 0，此时 D=q，因此在下一个时钟周期里 q 值保持先前的值不变。

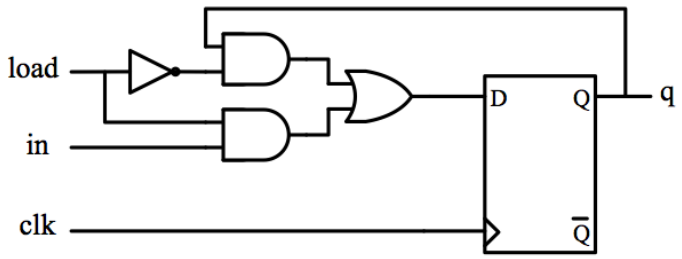


图 6-1: 1 位寄存器

用 Verilog 语言设计寄存器也很简单，如表 6-1所示。

表 6-1: 1 位寄存器代码

```
module register1(load,clk,clr,inp,q);
  input  load,clr,clk,inp;
  output reg q;

  always @(posedge clk)
    if (clr==1)
        q <= 0;
    else if (load == 1)
         q <= inp;
endmodule
```

表 6-1的程序的仿真图如图 6-2所示。

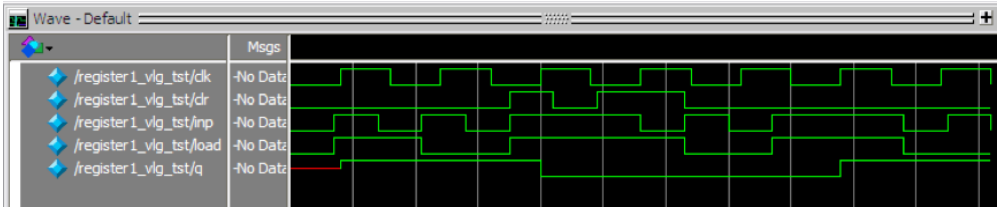


图 6-2: 1 位寄存器仿真结果

本例实现的是一个带有清 0 端和输入端的 1 位寄存器，还有的寄存器带有置位（置 1）端的，图 6-3是同时带有清 0 端、输入端和置位端的寄存器的逻辑示意图，读者可自行设计此寄存器。

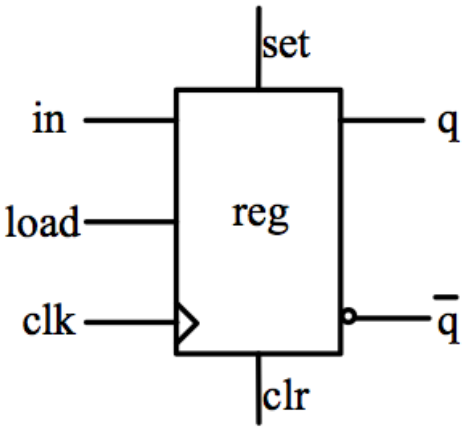


图 6-3: 1 位寄存器框图

将 2 个或者 2 个以上的 1 位寄存器组合在一起，这些寄存器共用一个时钟信号，这就构成了多位寄存器，寄存器常被用在计算机中存储数据，如指令寄

存器、数据寄存器等。表6-2是利用 Verilog 语言设计寄存器的例子。

表 6-2: 4 位寄存器代码

```
module register4(load,clk,clr,d,q);
  input  load,clr,clk;
  input  [3:0] d;
  output reg [3:0] q;

  always @(posedge clk)
    if (clr==1)
         q <= 0;
    else if (load == 1)
         q <= d;
endmodule
```

表6-2的程序的仿真图如图6-4所示。

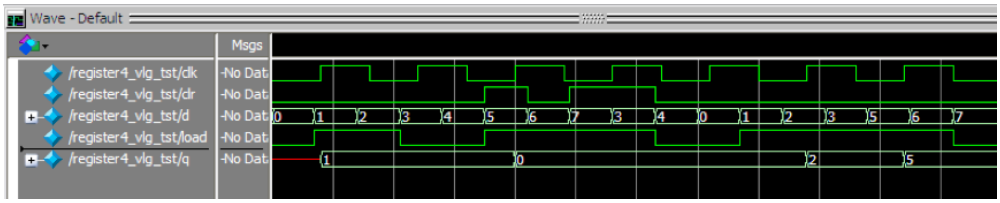
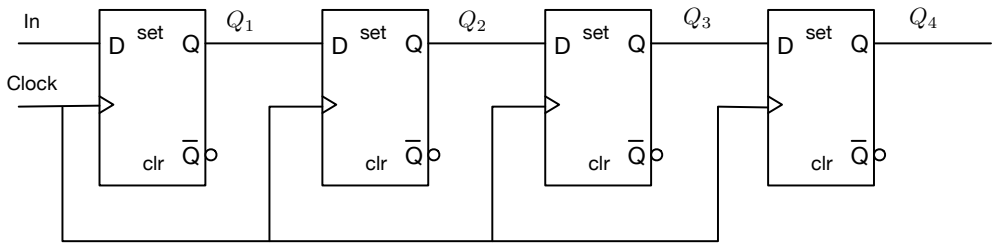


图 6-4: 4 位寄存器仿真结果

6.2 移位寄存器

移位寄存器是一类寄存器，它在时钟的触发沿，根据其控制信号，将存储在其中的数据向某个方向移动一位。移位寄存器也是数字系统的常用器件。

图 6-5(a)中是一个由 4 个 D 触发器构成的简单向右移位寄存器，数据从移位寄存器的左端输入，每个触发器的内容在时钟的正跳变沿（上升沿）将数据传到下一个触发器。图 6-5(b)是一个此移位寄存器的序列传递实例。



(a) 移位寄存器框图

	In	Q1	Q2	Q3	Q4=Out
t0	1	0	0	0	0
t1	0	1	0	0	0
t2	1	0	1	0	0
t3	1	1	0	1	0
t4	1	1	1	0	1
t5	0	1	1	1	0
t6	0	0	1	1	1
t7	0	0	0	1	1

(b) 移位实例

图 6-5: 移位寄存器

6.3 实验内容

6.3.1 算术移位和逻辑移位寄存器

这里的算术移位是指考虑到符号位的移位，算术移位要保证符号位不改变，算术左移同逻辑左移一样，算术右移最左面的空位补符号位。逻辑移位不管是向左移位还是向右移位都是空缺处补 0。循环是将移出去的那一位补充到空出的最高/低位的移位方式。置数是将一个 8 位的数据输入到寄存器中，即给寄存器赋一个初始值。

用 Verilog HDL 语言很容易描述出移位寄存器，如：

```
1 Q <= {Q[0],Q[7: 1]}; //循环右移
2 Q <= {Q[7],Q[7: 1]}; //算术右移
```

请根据表 6-3，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。

表 6-3: 移位寄存器的工作方式

控制位	工作方式
0 0 0	清 0
0 0 1	置数
0 1 0	逻辑右移
0 1 1	逻辑左移
1 0 0	算术右移
1 0 1	左端串行输入 1 位值，并行输出 8 位值
1 1 0	循环右移
1 1 1	循环左移

检查移位寄存器的功能正确后，请将编译好的文件下载到 FPGA 开发平台上进行验证。这里要求用 DE10-Standard 开发板上的按钮作为时钟输入端，即按钮按下，时钟由高电平变为低电平，产生一个下降沿；按钮松开，时钟由低电平变为高电平，产生一个上升沿。

注意，DE10-Standard 开发板上的 Switch 和 Key 都是机械按钮，机械按钮在按下和松开时都会发生机械抖动，产生不可预期的数个上升沿和下降沿。开发板上的 Key 是经过消抖的，但是 Switch 没有消抖。如果用 Switch 做时钟。请参考 “按键消抖” 内容，先对 Switch 进行消抖，然后再将其作为时钟使用，如果用 key 可以忽略此步骤。

6.3.2 利用移位寄存器实现随机数发生器

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。

☞ **思考题**:

生成的伪随机数序列仍然有一定的规律，如何能够生成更加复杂的伪随机数序列?