## 实验二 寄存器设计

# 实验目的

（1）熟悉使用Verilog语言描述时序逻辑电路；

（2）掌握约束文件、仿真文件的编写；

（3）掌握寄存器的设计，加深对寄存器的理解；

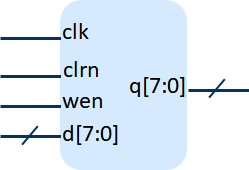
# 1、实验题目一：8位寄存器设计与实现

## 1.1 实验内容

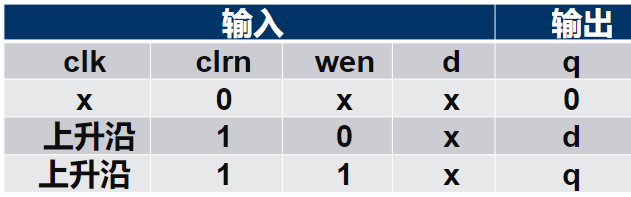
运用Verilog的**结构描述法**，使用理论课介绍的D触发器完成带有异步清零 clrn 和 同步使能wen的8位寄存器reg8。需上板验证，输入接拨码开关，输出驱动led显示。

## 1.2 实验原理

所要实现的8位寄存器原理图如图1-1，具有8位数据输入和输出，异步清零clrn和同步使能wen，具体真值表如图1-2所示。

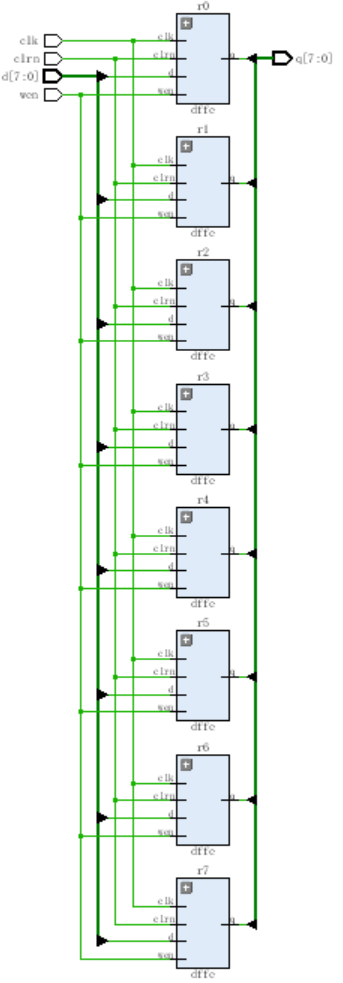


**图1-1 reg8功能框图**



**图1-2 reg8功能表**

8位寄存器的Verilog实现方式有多种，这里要求先实现1位寄存器即D触发器，再通过实例化即结构化描述法实现8位寄存器，所实现的电路结构如下图所示。



**图 1-3 8位寄存器reg8原理图**

新建工程reg8，先完成D触发器dffe.v的实现。图1-2功能表表明：信号clrn(clear\_negative)在低电平时进行异步清零（异步是指不受时钟约束，在任何时钟下均产生结果）。在时钟上升沿时，若信号wen(write\_enable\_negative)为高电平，则写无效，信号保持原来的值；若信号wen为低电平，则写有效，此时信号q的值更新为上升沿处d的值。

然后在reg8.v中实例化dffe模块，完成最终的实现。添加仿真文件reg8\_sim.v，框架代码如下，需自行补全仿真文件,需覆盖复位、跟随、保持三种情况。

`timescale 1ns/1ps   //1ns表示延时单位，1ps表示时间精度

module reg8\_sim(

    );

    reg clk = 1'b0;

    reg clrn = 1'b1;

    reg wen = 1'b1;

    reg [7:0] d = 8'b1;

    wire [7:0] q;

//1 请补全

    always begin

        #10 clk=~clk;

    end

    initial begin

        #25 begin clrn = 1'b0; wen = 1'b0; end

        #10 begin clrn = 1'b1; wen = 1'b1; end

        #30 wen = 1'b0;

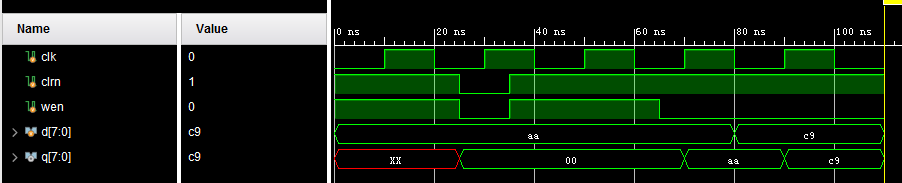
//2 请补全

        #30 $finish;

    end

endmodule

运行仿真，仿真结果如图2-2所示。



**图 1-4 8位寄存器的仿真波形图**

添加约束文件reg8.xdc，reg8 的管脚分配如[表 1-1](#_bookmark158) 所示，自行完成约束文件。

**表 1-1 reg8管脚分配表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 信号 | 部件 | 管脚 | 信号 | 部件 | 管脚 |
| d[7] | SW8-8 | T5 | q[0] | D0 | K2 |
| d[6] | SW8-7 | T3 | q[1] | D1 | J2 |
| d[5] | SW8-6 | R3 | q[2] | D2 | J3 |
| d[4] | SW8-5 | V4 | q[3] | D3 | H4 |
| d[3] | SW8-4 | V5 | q[4] | D4 | J4 |
| d[2] | SW8-3 | V2 | q[5] | D5 | G3 |
| d[1] | SW8-2 | U2 | q[6] | D6 | G4 |
| d[0] | SW8-1 | U3 | q[7] | D7 | F6 |
| clk | SW0 | R1 | clrn | SW2 | M4 |
| wen | SW1 | N4 |  |  |  |

## 1.3 工程文件要求

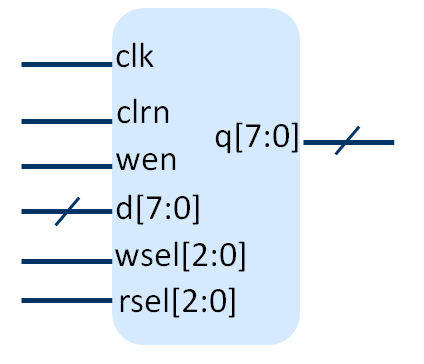
工程文件名称：reg8； 主模块文件名：reg8.v；

仿真文件文件名：reg8\_sim.v； 约束文件文件名：reg8.xdc。

# 2 实验题目二：寄存器文件设计与实现

## 2.1 实验内容

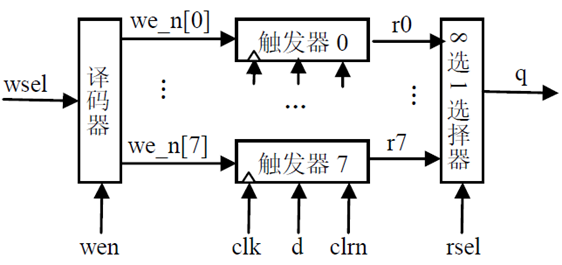
在题目一的基础上，使用reg8，增加译码器和多路选择器，完成8个8位寄存器组成的寄存器文件reg8file的设计。每次访问一个寄存器，可以读寄存器也能写寄存器。



**图2-1 reg8file功能框图**

wsel是写选择端，输入3位写地址；rsel是读选择端，输入3位读地址。wen是低电平有效的写使能端，wen有效的时候d[7:0]写入wsel指定的寄存器。q[7:0]始终输出rsel指定的寄存器的内容。

待实现的寄存器文件内部结构参考图如图2-2。we\_n是内部信号，连接各触发器的写使能端，ri是内部信号，为触发器i的输出。



**图2-2 reg8file内部结构参考图**

reg8file 的管脚分配如[表 2-1](#_bookmark158) 所示，自行完成约束文件。如果Ego1板子标注的部件名顺序与表2-1不一致，不会影响实验，因管脚顺序是不变的，观察现象时分清顺序即可。

**表 2-1 reg8file管脚分配表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 信号 | 部件 | 管脚 | 信号 | 部件 | 管脚 |
| d[7] | SW8-8 | T5 | q[0] | D0 | K2 |
| d[6] | SW8-7 | T3 | q[1] | D1 | J2 |
| d[5] | SW8-6 | R3 | q[2] | D2 | J3 |
| d[4] | SW8-5 | V4 | q[3] | D3 | H4 |
| d[3] | SW8-4 | V5 | q[4] | D4 | J4 |
| d[2] | SW8-3 | V2 | q[5] | D5 | G3 |
| d[1] | SW8-2 | U2 | q[6] | D6 | G4 |
| d[0] | SW8-1 | U3 | q[7] | D7 | F6 |
| clk | SW0 | R1 | clrn | SW2 | M4 |
| wen | SW1 | N4 | rsel[2] | - | - |
| wsel[2] | SW5 | P3 | rsel[1] | SW7 | P5 |
| wsel[1] | SW4 | P2 | rsel[0] | SW6 | P4 |
| wsel[0] | SW3 | R2 |  |  |  |

自行完成仿真和约束文件，仿真需覆盖复位、读寄存器、写寄存器等情况。因拨码开关数不够，rsel[2]不进行约束。

## 2.3 工程文件要求

工程文件名称：reg8file； 主模块文件名：reg8file.v；

仿真文件文件名：reg8file\_sim.v； 约束文件文件名：reg8file.xdc。

在reg8file.v中实例化译码器、触发器、多路选择器，以该文件为顶层文件实现上述设计，并编写仿真文件测试寄存器文件的功能。

# 3、附加题

（1）使用Verilog分别实现RS锁存器、RS触发器、JK触发器、T触发器并上板观察实现现象，分析RTL图不同之处。

（2）自行设计实验通过代码理解阻塞与非阻塞的区别，可以参考https://blog.csdn.net/Reborn\_Lee/article/details/101861516

该题不计分不做检查，有兴趣的同学自行实现。

# 4、实验检查与提交要求

实验题目一检查并解释实验现象（1）分；

实验题目二提交仿真结果截图与文字分析、工程文件（1分）。