2020-9-30

秦嘉余

191220088

1348288404@qq.com

数字电路实验六 寄存器

目录

[1.实验目的 1](#_Toc52393679)

[2.实验原理 1](#_Toc52393680)

[3.实验环境/器材 2](#_Toc52393681)

[4.算术移位和逻辑移位寄存器 2](#_Toc52393682)

[1.设计思路 2](#_Toc52393683)

[2. 实验代码 2](#_Toc52393684)

[3.RTL视图 4](#_Toc52393685)

[4.引脚分配图 4](#_Toc52393686)

[5.实验结果 5](#_Toc52393687)

[5. 利用移位寄存器实现随机数发生器 5](#_Toc52393688)

[1.设计思路 5](#_Toc52393689)

[2. 实验代码 5](#_Toc52393690)

[3.RTL视图 6](#_Toc52393691)

[4.引脚分配图 6](#_Toc52393692)

[5.实验结果 6](#_Toc52393693)

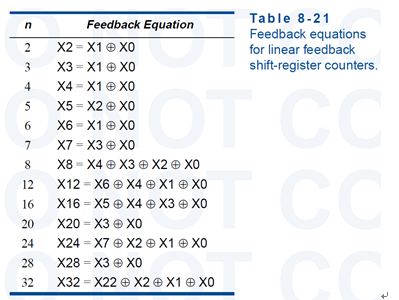
[6.实验实验中遇到的问题及解决办法 7](#_Toc52393694)

1.实验目的

本次实验的目的是熟悉数字电路中的常用原件的寄存器，并使用verilog语言来实现。同时，利用寄存器作为基本原理，完成移位寄存器等一系列原件，最后完成一个随机数生成器。

2.实验原理

寄存器是数字电路中的重要器件，用于存储比特信号，在D触发器的基础上用一定的逻辑门作为输入就可以构成最简单的寄存器。在此门电路可以涵盖清零，置数的功能。在此基础上，将多个D触发器链接在一起就可以构成移位寄存器，然后根据需求拓展不同的功能。完成移位寄存器后通过



上表知X4 xor X3 xor X2 xor X0可以得到随机数序列

3.实验环境/器材

本次实验的环境为Quartus17.1版本

本次实验的器材为DE10 Standard开发板

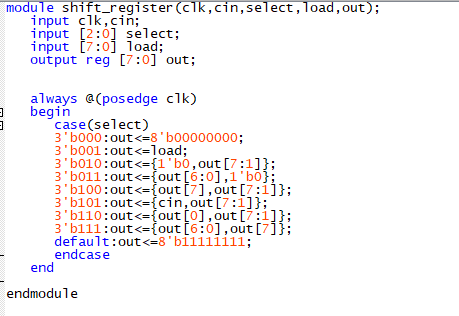
4.算术移位和逻辑移位寄存器

## 1.设计思路

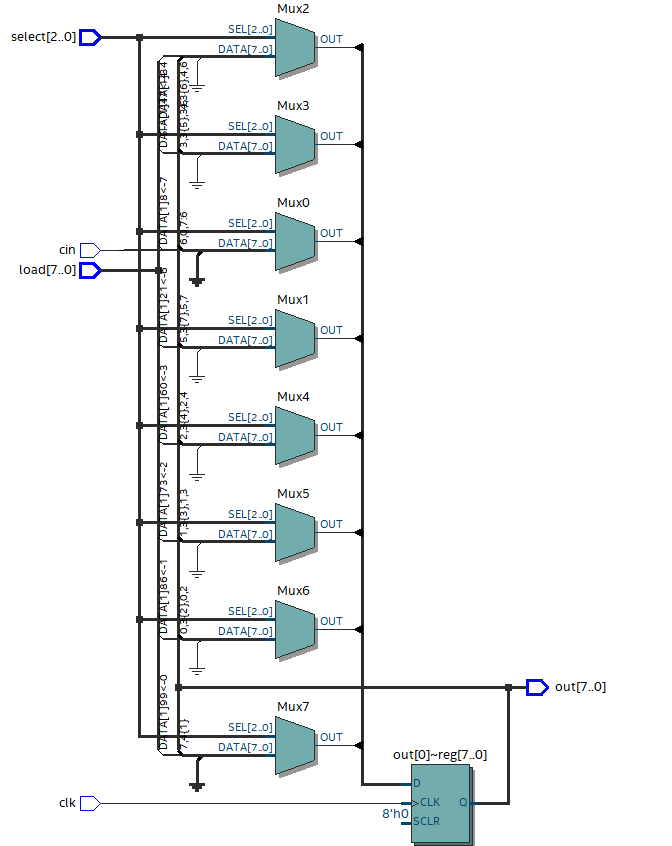
本次实验的形式与ALU类似，同一个器件通过功能选择端完成多种功能，所以大框架使用case语句，always语句检测clk信号，case检测select，根据不同的select选择不同的功能。而每种功能只需要根据select对out进行不同赋值即可。

## 2. 实验代码

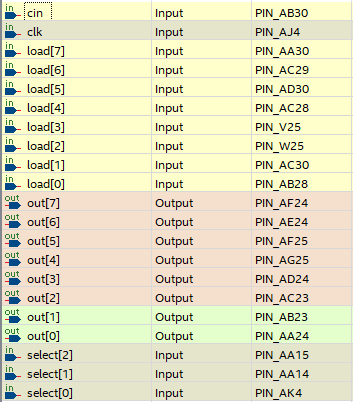
模块名为shift\_register，时钟信号为clk，输入值为cin，选择端为select，载入输入端为load，输出端为8路out



## 3.RTL视图



## 4.引脚分配图



## 5.实验结果

已在开发板上验收

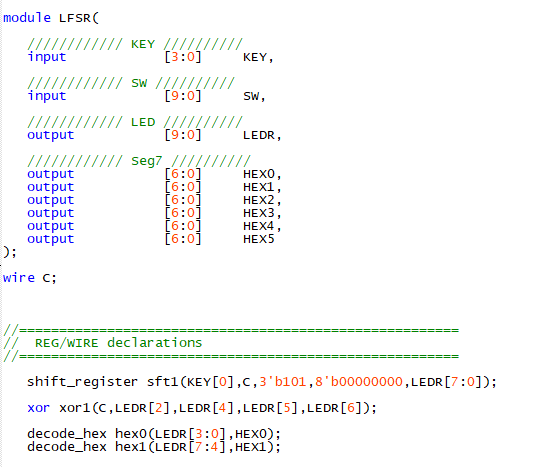
5. 利用移位寄存器实现随机数发生器

## 1.设计思路

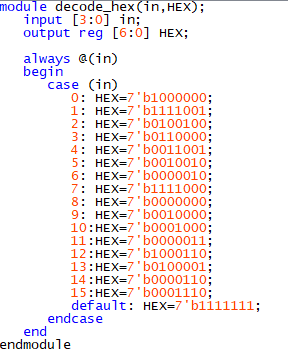
在上实验中我们已经实现了移位寄存器，在本次实验中我们只需要将移位寄存器选择端调至101，选取输入作为移入信号，而移入信号的选取可以根据。这样当初始状态不为00000000,时，即可出现除0外全部状态。最后，设计译码器将8位输出的高四位和第四位接入7段数码管显示。

## 2. 实验代码

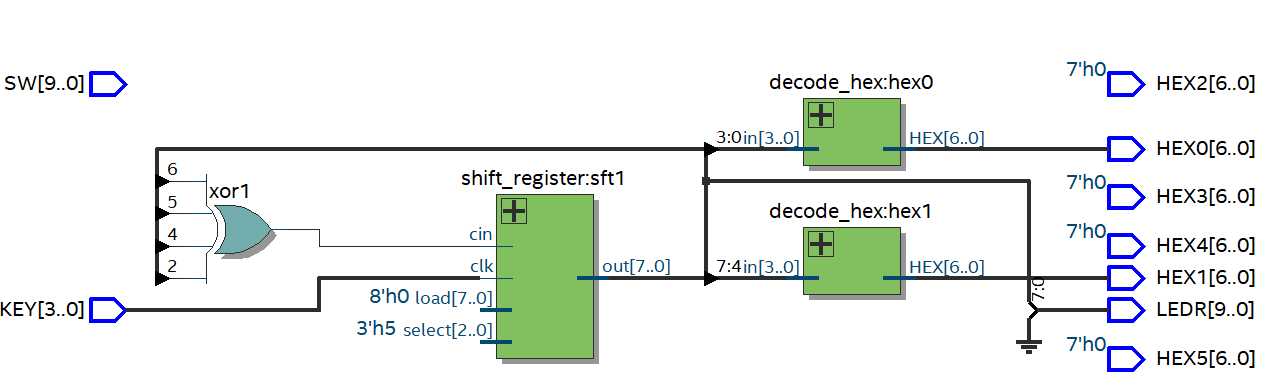
顶层模块名为LFSR。其中shift\_register模块与上实现一致，decode\_hex为显示模块，xor为verilog内置的异或门模块。



下为decode\_hex模块



## 3.RTL视图



## 4.引脚分配图

本项目由build软件自动生成，则不再展示引脚分配图

## 5.实验结果

已在开发板上验收

## 6.补充

由于题目中规定为需要自启动，为了保证全部的256种情况，需要增加对00000000的自启动规则，所以对逻辑反馈门进行改进，完成可以自启动的电路，改进后的方程如下：



其中D为反馈回的值

6.实验实验中遇到的问题及解决办法

对于随机数的生成，由于直接使用异或的门生成电路无法从8’b00000000自启动，所以如果需要自启动，有两种方式：1.对反馈门电路进行修改，补充了从00000000启动的规则，最后完成了自启动 2.将初始值改变，使其不为00000000，这样就可以自启动了