# 实验8&9、加法器、加减法器

# 和ALU基本原理与设计实验报告

姓名： 王祚滨 专业： 信息安全 学号： 3180104933

课程名称： 逻辑与计算机设计基础实验 同组学生姓名： 王国朝、赵卿云

指导老师： 洪奇军 实验地点： 浙江大学紫金港校区东四教学楼509实验日期：2019年 11 月 13 日

# 一、实验目的和要求

1.1掌握一位全加器的工作原理和逻辑功能

1.2掌握串行进位加法器的工作原理和进位延迟

1.3掌握减法器的实现原理

1.4掌握加减法器的设计方法

1.5掌握ALU基本原理及在CPU中的作用

1.6掌握ALU的设计方法

**二、实验内容和原理**

**2.1实验内容：**

任务1：原理图方式设计4位串行进位加法器

任务2：实现4位加法器应用

**2.2实验原理：**

**2.2.1位全加器**

三个输入位：数据位 Ai 和 Bi，低位进位输入 Ci

二个输出位：全加和 Si，进位输出 Ci+1

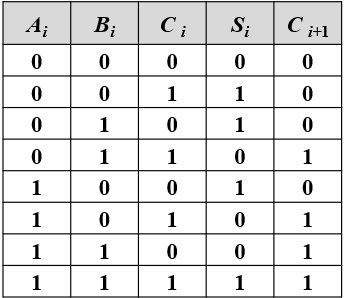




图1 一位全加器真值表和关系式

根据一位全加器的输入输出关系，得到电路图

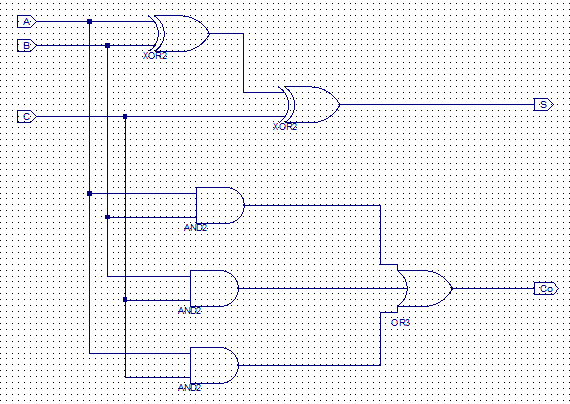
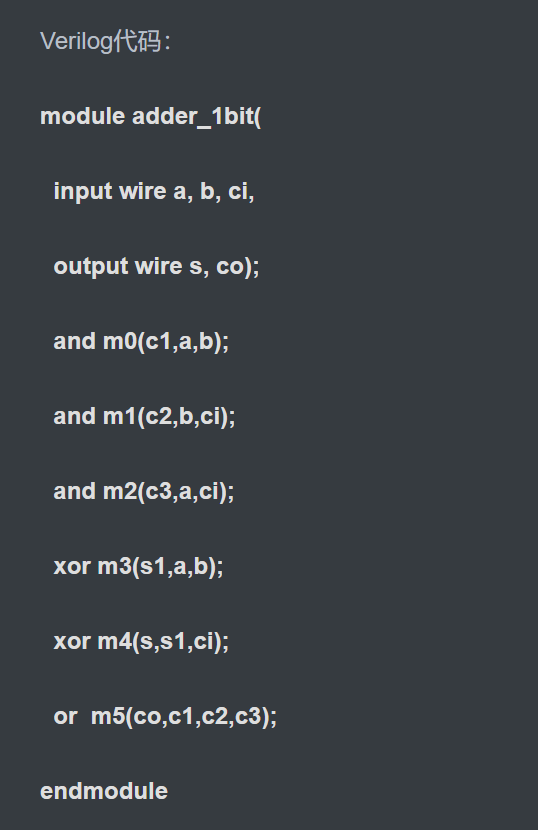


图2 一位加法器电路图



**2.2.2 多位串行进位加法器**

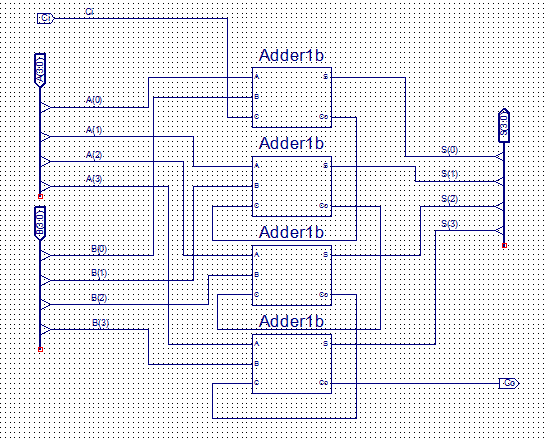
多位全加器可由一位全加器将进位串接构成

高位进位生成速度慢，位数越多时间越长



图3 多位全加器

**2.2.3 4位全加器**



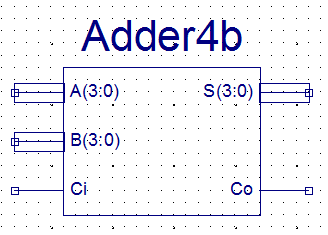


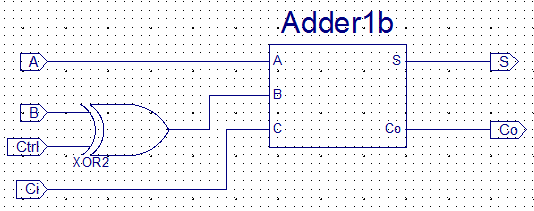
图4 四位全加器电路图

**2.2.4 1位加减法器**

用负数补码加法实现，减数当作负数求补码。

共用加法器。

用“异或”门控制求反，低位进位C0为1。



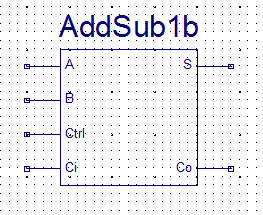


图5 一位加减法器

**2.2.5多位串行进位全减器**

用负数补码加法实现，减数当作负数求补码

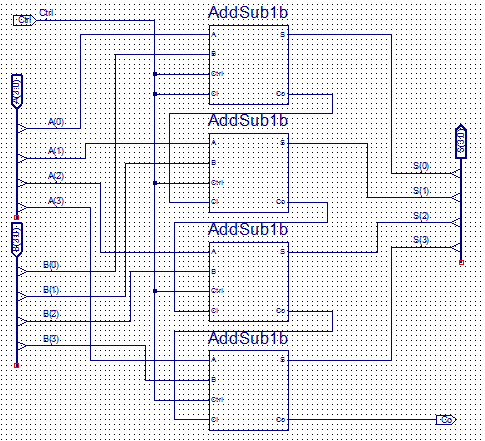
共用加法器

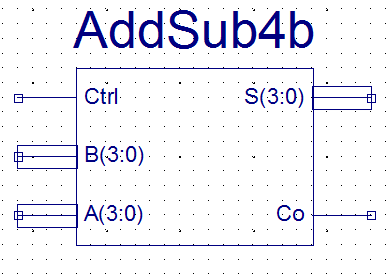
用“异或”门控制求反，低位进位C0为1





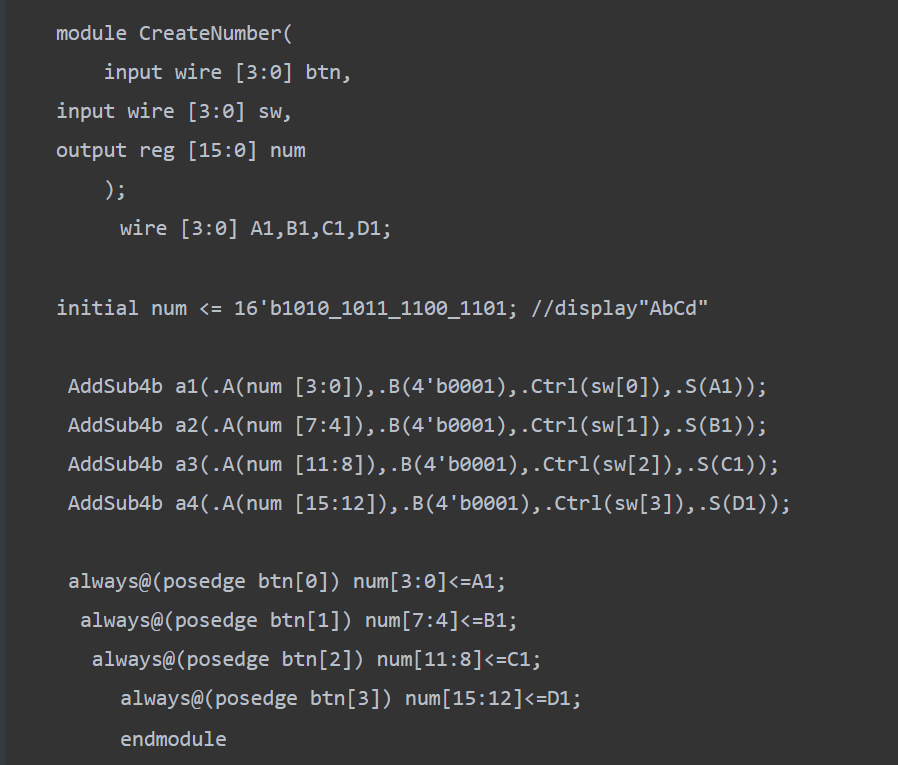
**2.2.6 4位加减法器**



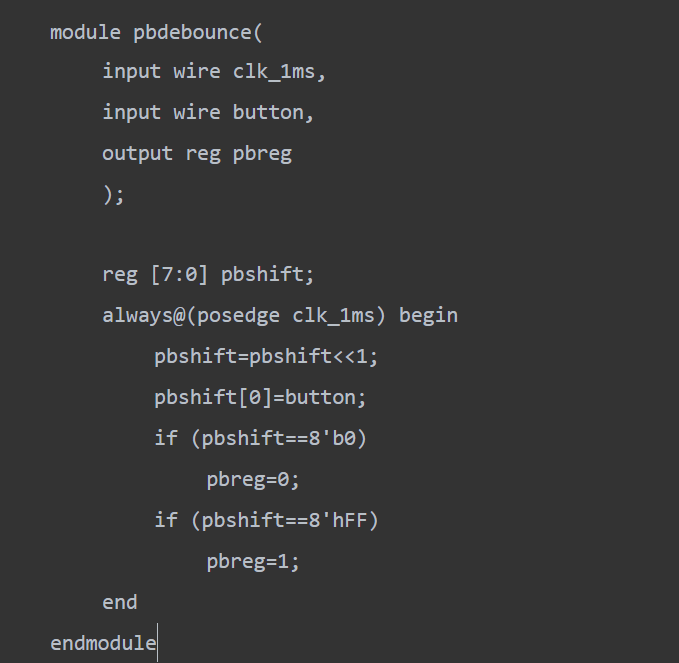


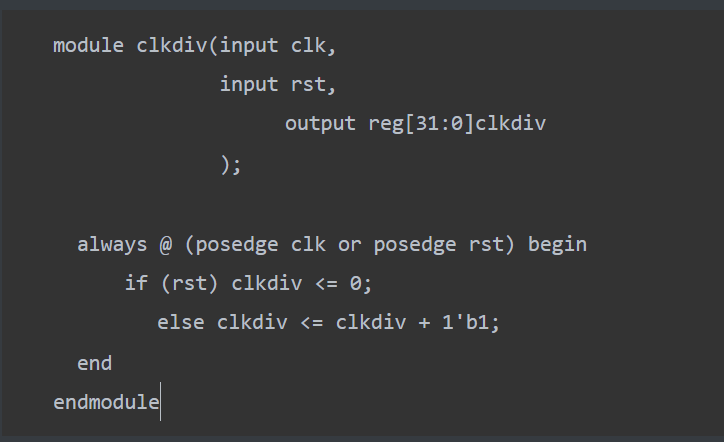
图六 四位加减法器

2.2.7 设计按键数据输入模块



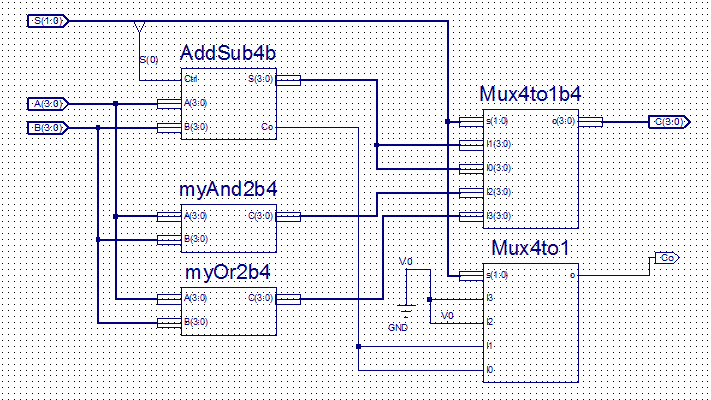
2.2.8 防抖动模块 + 分频器





图七 clkdiv图

**2.2.10 4位ALU原理图**



图八 2.2.10 4位ALU原理图

# 三、主要仪器设备

# 1. 装有Xilinx ISE 14.7的计算机 1台

# 2. SWORD开发板

# 四、操作方法与实验步骤

**4.1实验任务：**

任务1：原理图方式设计4位加减法器

任务2：实现4位ALU及应用设计

**4.2实验步骤：**

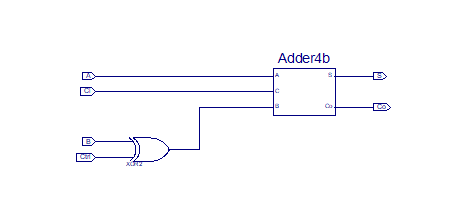
4.2.1 AddSub1b设计

1.新建工程，工程名称用MyALU。

2.Top Level Source Type用HDL

3.新建源文件，类型是Schematic，文件名称用AddSub1b。

4.原理图方式进行设计



图九 AdderSub1b

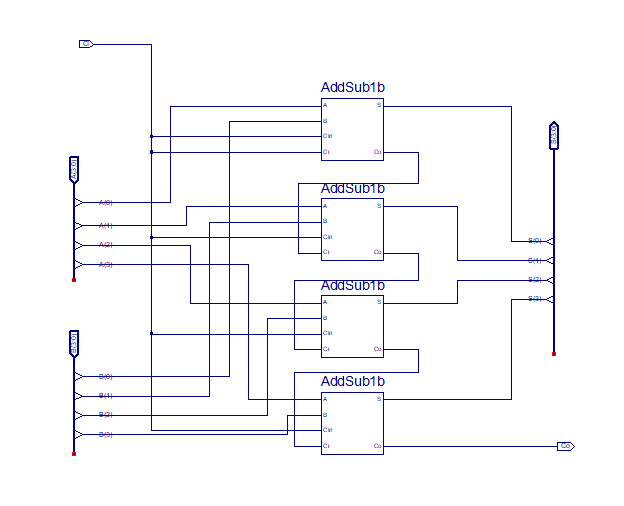
4.2.2 AddSub4b设计

1.新建源文件

2.类型是Schematic

3.文件名称用AddSub4b

4.原理图方式进行设计，调用前面设计的AddSub1b



图十 AdderSub4b

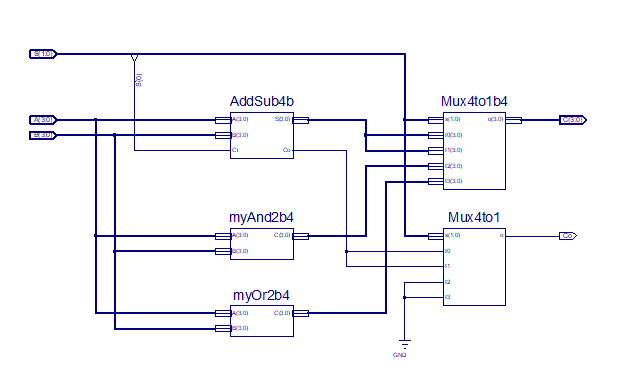
4.2.3 ALU设计

1.新建源文件

2.类型是Verilog或Schematic

3.文件名称用ALU

4.原理图方式进行设计



图十一 ALU

4.2.4 top设计

1.新建源文件，类型是Verilog，文件名Top。，右键设为“Set as Top Module”

2.代码输入进行设计

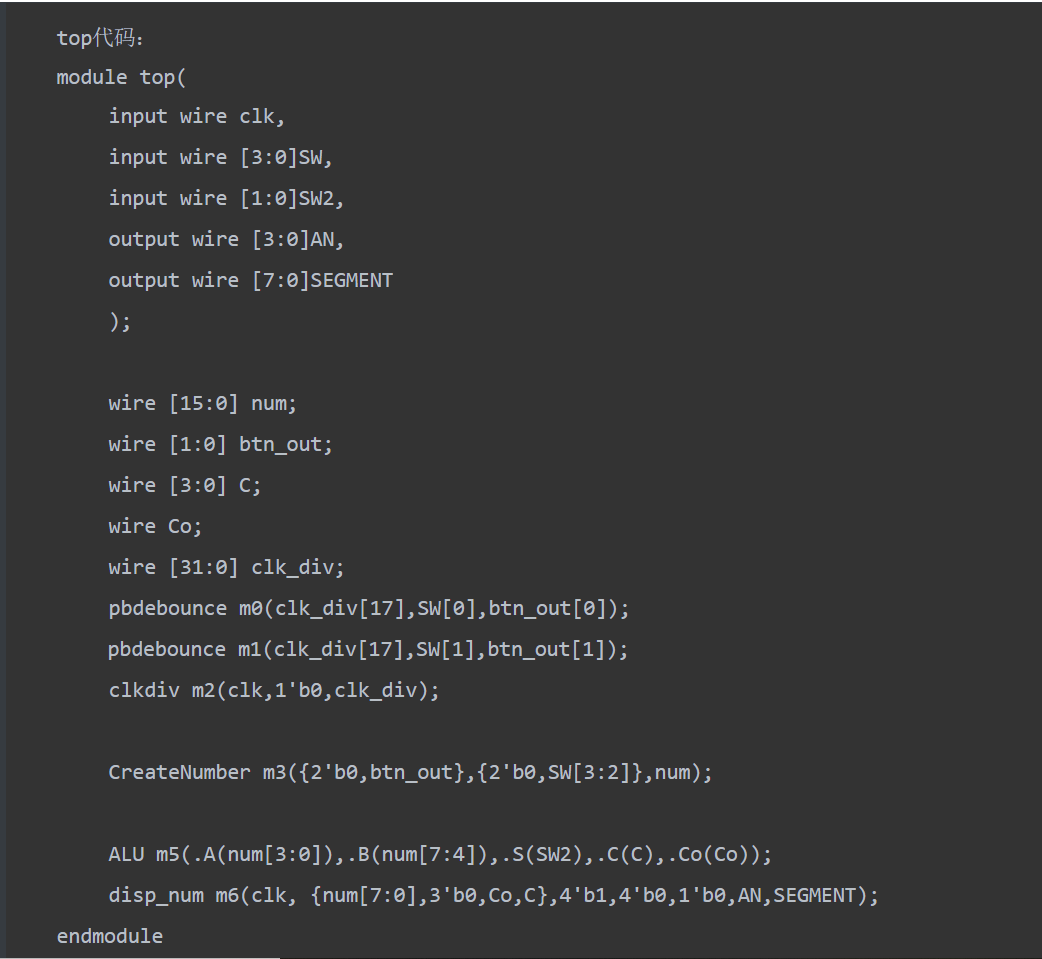
调用pbdebounce模块

调用AddSub4b模块

调用pbdebounce、clkdiv模块

调用DispNum模块

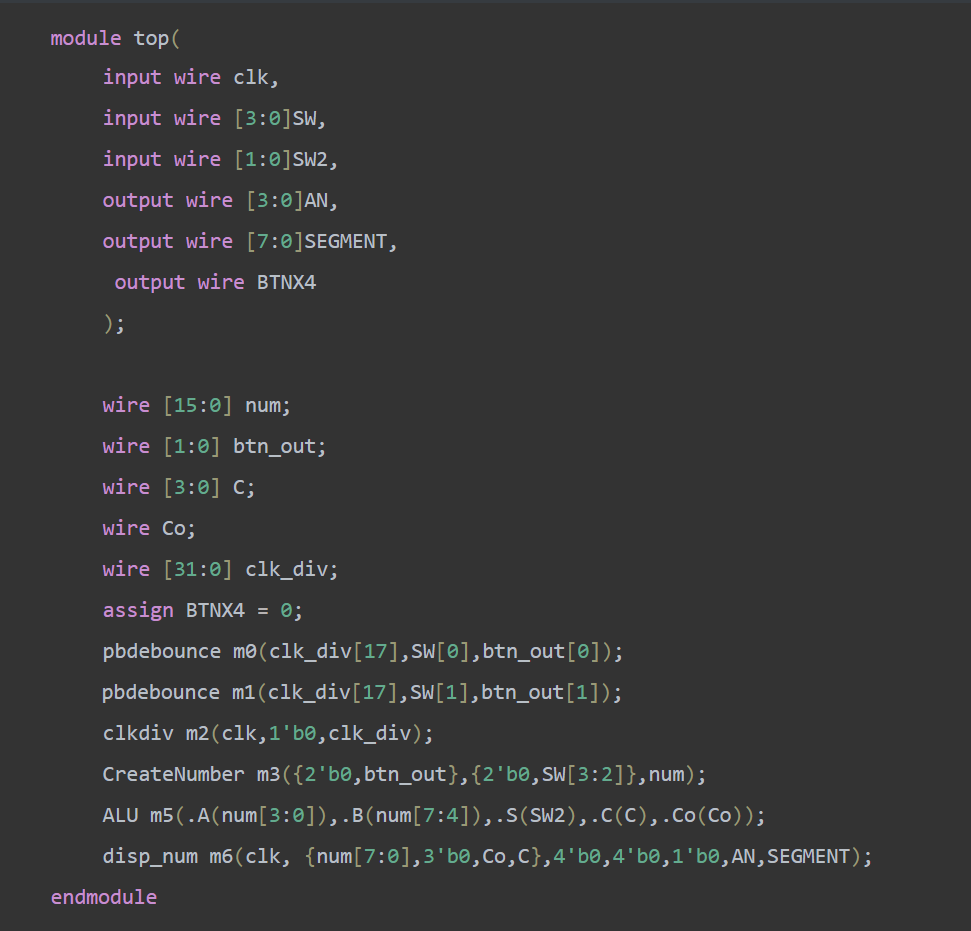
调用CreateNumber模块

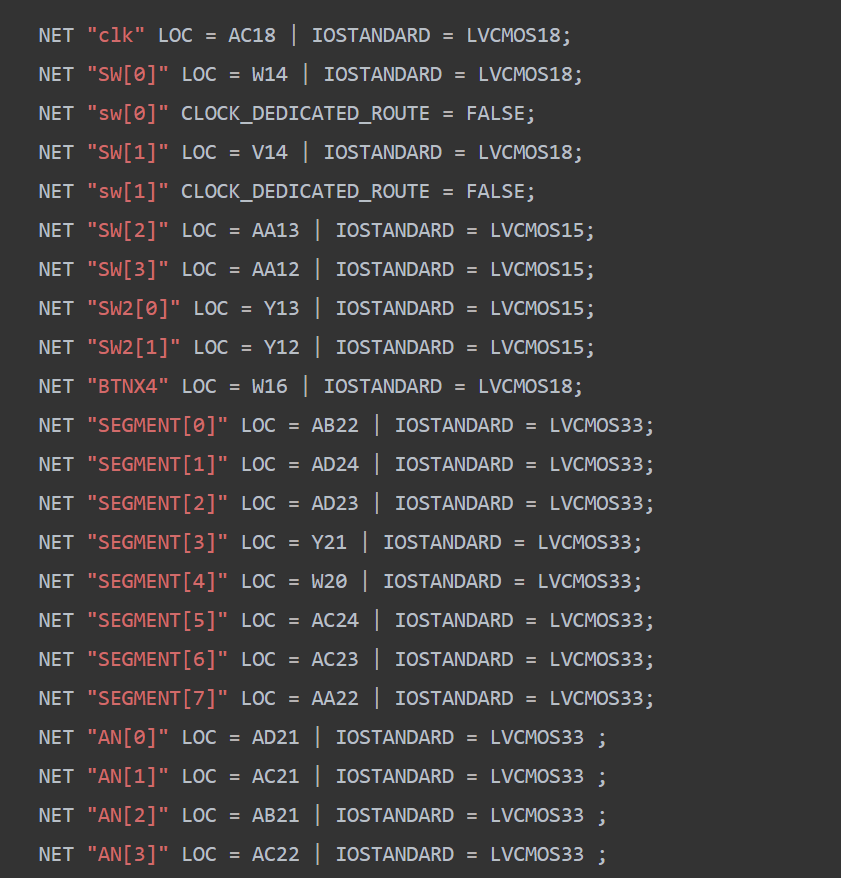


4.2.5. UCF引脚定义

# 

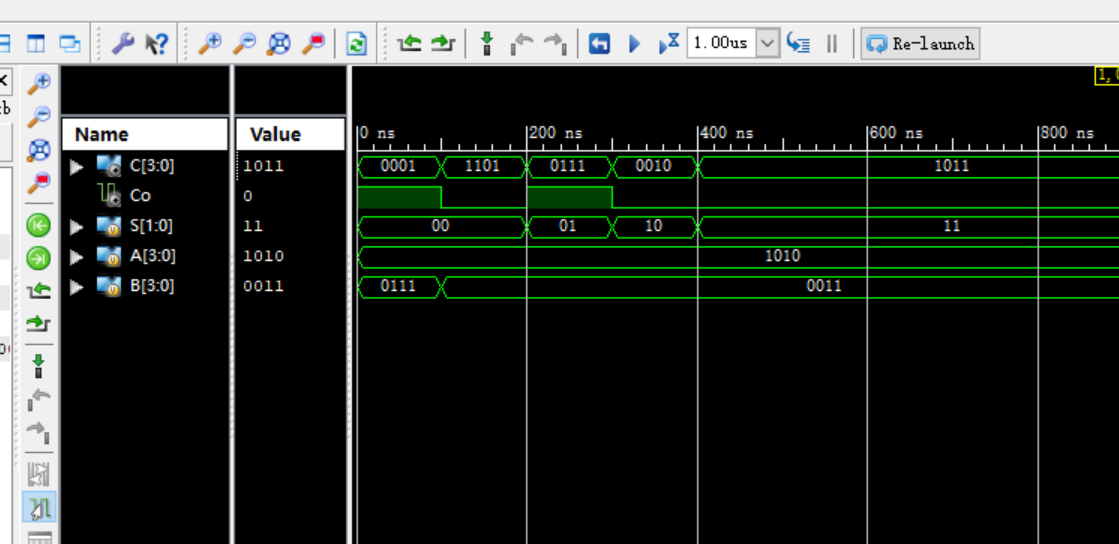
附加：更改为按钮实现





# 五、实验结果与分析

5.1 4bitALU仿真：



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 时间 | 输入 | | | 计算 | | 输出 | |
| A[3:0] | B[3:0] | S[1:0] | 运算符 | 结果 | C0 | S[3:0] |
| 0-50ns | 1010 | 0111 | 00 | + | 1 0001 | 1 | 0001 |
| 50-100ns | 1010 | 0011 | 00 | + | 0 1101 | 0 | 1101 |
| 100-150ns | 1010 | 0011 | 01 | - | 1 0111 | 1 | 0111 |
| 150-200ns | 1010 | 0011 | 10 | AND | 0 0010 | 0 | 0010 |
| 200ns- | 1010 | 0011 | 11 | OR | 0 1011 | 0 | 1011 |

5.2 ALU计算结果

|  |  |  |
| --- | --- | --- |
| **数据** | **分析** | **结果** |
| IMG_20191113_200851 | **输入**  A[3:0] = 3 (0011)  B[3:0] = 6 (0110)  S[1:0] = 00  **计算**  Operator: +  Result = 9 (1001)  **输出**  C0 = 0  C[3:0] = 9 (1001) | **√** |
| IMG_20191113_200924 | **输入**  A[3:0] = F (1111)  B[3:0] = 6 (0110)  S[1:0] = 11  **计算**  Operator: OR  Result = F (0 1111)  **输出**  C0 = 0  C[3:0] = F (1111) | **√** |
| IMG_20191113_201007 | **输入**  A[3:0] = 2 (0010)  B[3:0] = 6 (0110)  S[1:0] = 01  **计算**  Operator: -  Result = -4  **输出**  C0 =1  C[3:0] = 4 | **√** |
| IMG_20191113_201012 | **输入**  A[3:0] =2 (0010)  B[3:0] = 6 (0110)  S[1:0] = 01  **计算**  Operator: AND  Result = 2 (0 0010)  **输出**  C0 = 0  C[3:0] = 2 (0010) | **√** |

# 六、讨论、心得

通过本次实验，我了解到了ALU的设计原理，同时对模块化的设计有了更深的了解，在实验中应用之前设计过的一些模块，让整个过程简单了很多。

在实现加减功能时，更是实践了课上所讲的补码形式，也对理论知识有了进一步的探究，总的来说实验难度并不大，只要细心就可以成功得出结果。

后期将开关改为按钮时，遇到了不小麻烦，没有将使能设置为0，可见对verilog语言仍然比较生疏，还需要进一步提高。

# 实验10、锁存器与触发器基本原理实验报告

姓名： 王祚滨 专业： 信息安全 学号： 3180104933

课程名称： 逻辑与计算机设计基础实验 同组学生姓名： 王国朝、赵卿云

指导老师： 洪奇军 实验地点： 浙江大学紫金港校区东四教学楼509实验日期：2019年 11 月 20 日

# 一、实验目的和要求

1.掌握锁存器与触发器构成的条件和工作原理

2.掌握锁存器与触发器的区别

3.掌握基本SR锁存器、门控SR锁存器、D锁存器、SR锁存器、D触发器的基本功能

4.掌握基本SR锁存器、门控SR锁存器、D锁存器、SR锁存器存在的时序问题

**二、实验内容和原理**

2.1实验内容：

1.实现基本SR锁存器，验证功能和存在的时序问题

2.实现门控SR锁存器，并验证功能和存在的时序问题

3.实现D锁存器，并验证功能和存在的时序问题

4.实现SR主从触发器，并验证功能和存在的时序问题

5.实现D触发器，并验证功能

2.2实验原理：

**2.2.1构成锁存器的充分条件**

能长期保持给定的某个稳定状态

有两个稳定状态：0、1

在一定条件下能随时改变逻辑状态，即：置1或置0

最基本的锁存器有：SR锁存器、D锁存器

锁存器有两个稳定状态，又称双稳态电路

**SR锁存器**

将两个具有2输入端的反向逻辑器件的输出与输入端交叉连起来，另一个输入端作为外部信息输出端，就构成最简单的SR锁存器



|  |  |  |
| --- | --- | --- |
| ***R S*** | ***Q Q*** | **说明** |
| **0 0** | ***Q Q*** | **保持** |
| **0 1** | **1 0** | **置1** |
| **1 0** | **0 1** | **置0** |
| **1 1** | **0 0** | **未定义** |

图一 SR锁存器



图二 SR锁存器

|  |  |  |
| --- | --- | --- |
| ***R S*** | ***Q Q*** | **说明** |
| **0 0** | **1 1** | **未定义** |
| **0 1** | **0 1** | **置0** |
| **1 0** | **1 0** | **置1** |
| **1 1** | ***Q Q*** | **保持** |

**门控SR锁存器**



图三 门控SR锁存器

|  |  |  |
| --- | --- | --- |
| ***C R S*** | ***Q Q*** | **说明** |
| **0××** | ***Q Q*** | **保持** |
| **1 0 0** | ***Q Q*** | **保持** |
| **1 0 1** | **1 0** | **置1** |
| **1 1 0** | **0 1** | **置0** |
| **1 1 1** | **1 1** | **未定义** |

**D锁存器**

基本SR锁存器缺点：存在不确定状态

解决方法：消除不确定状态

只需1个数据输入端 D

输出端Q等于输入端D

采用电平控制 C



图四 D锁存器

|  |  |  |
| --- | --- | --- |
| ***C D*** | ***Q Q*** | **说明** |
| **0 ×** | ***Q Q*** | **保持** |
| **1 0** | **0 1** | **置0** |
| **1 1** | **1 0** | **置1** |

**触发器**

触发：外部输入使锁存器状态改变的瞬间状态

触发器：在锁存器的基础上使每次触发仅使状态改变一次的锁存电路（双稳态）

比D锁存器更有优势，能够避免空翻现象，使每次触发仅使锁存器内部状态改变一次。

分类：主从触发器，边沿触发器。

常见触发器：主从SR触发器、D触发器、JK触发器、T触发器

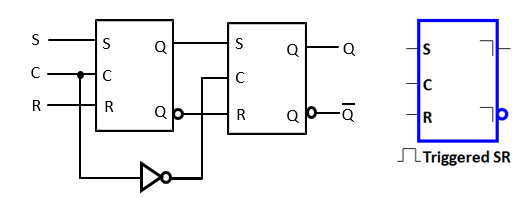
**SR主从触发器**

由两个钟控S-R锁存器串联构成，第二个锁存器的时钟通过反相器取反

当C=1时，输入信号进入第一个锁存器（主锁存器）

当C=0时，第二个锁存器（从锁存器）改变输出

从输入到输出的通路被不同的时钟信号值(C = 1 和 C = 0)所断开



图五 SR主从锁存器

**正边沿维持阻塞型D触发器**



图六 正边沿维持阻塞型D触发器

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **异步控制** | | **上升沿触发** | | | |
| ***R*** | ***S*** | ***CP*** | ***D*** | ***Q*** | ***Q*** |
| **0** | **1** | **×** | **×** | **0** | **1** |
| **1** | **0** | **×** | **×** | **1** | **0** |
| **1** | **1** | **↑** | **0** | **0** | **1** |
| **1** | **1** | **↑** | **1** | **1** | **0** |

# 三、主要仪器设备

# 1. 装有Xilinx ISE 14.7的计算机 1台

# 2. SWORD开发板

# 四、操作方法与实验步骤

**1.基本SR锁存器**

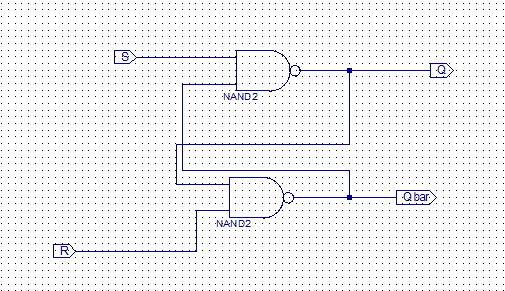
1.新建工程MyLATCHS

2.新建源文件SR\_LATCH.sch

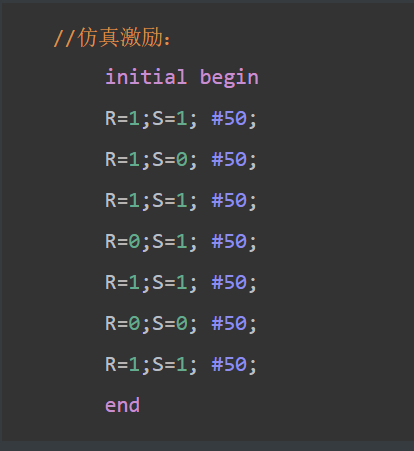
3.用原理图方式设计

4.用NAND2实现

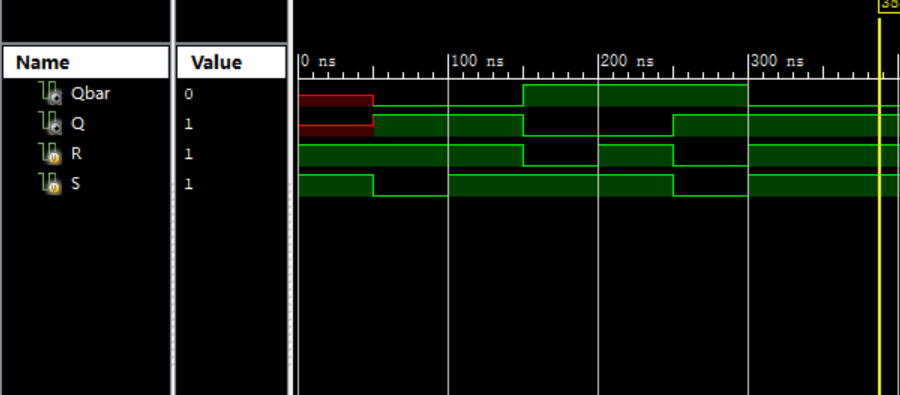
5.仿真



图：SR锁存器SCH图



图：仿真激励



图：仿真波形

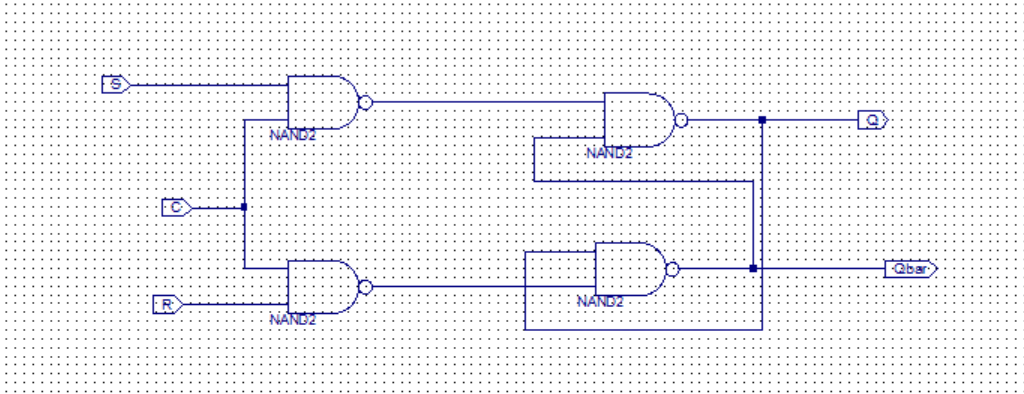
**2.门控SR锁存器**

1.新建源文件CSR\_LATCH.sch，用原理图方式设计。

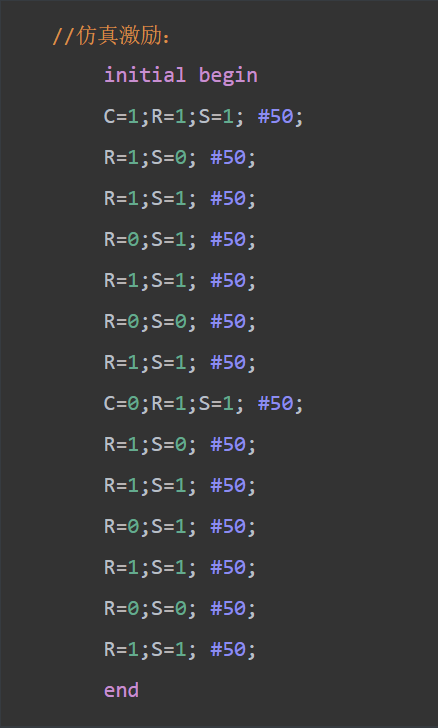
2.用NAND2实现

3.仿真（包含空翻）

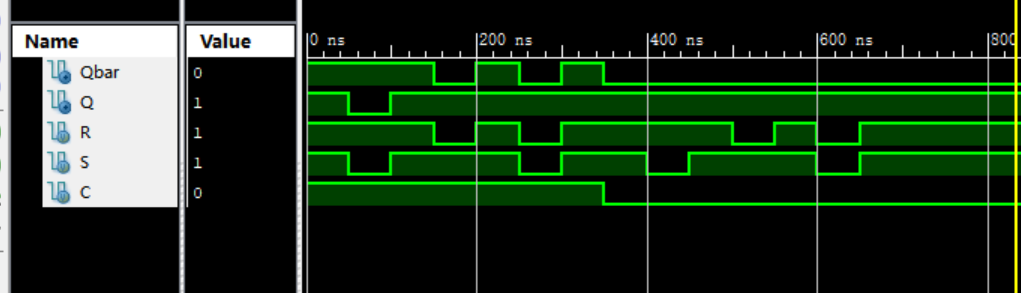
4.生成自定义符号的CSR\_LATCH.sym



图：门控SR锁存器SCH图



图：仿真激励



图：仿真波形

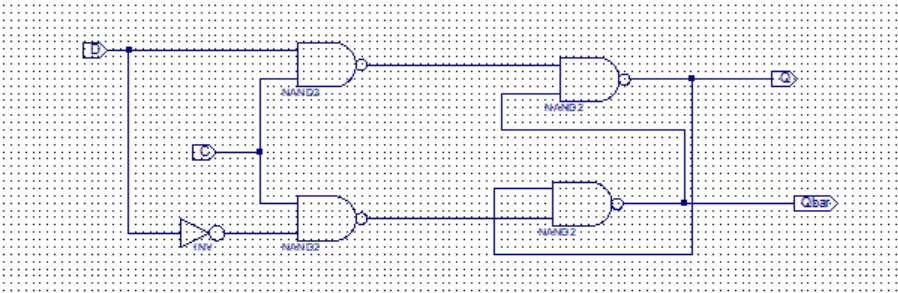
**3.D锁存器**

1.新建源文件D\_LATCH.sch

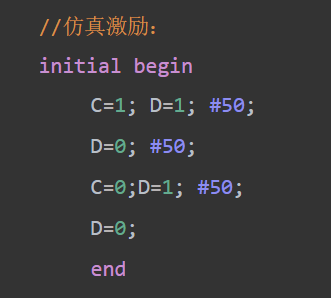
2.用原理图方式设计

3.用NAND2实现

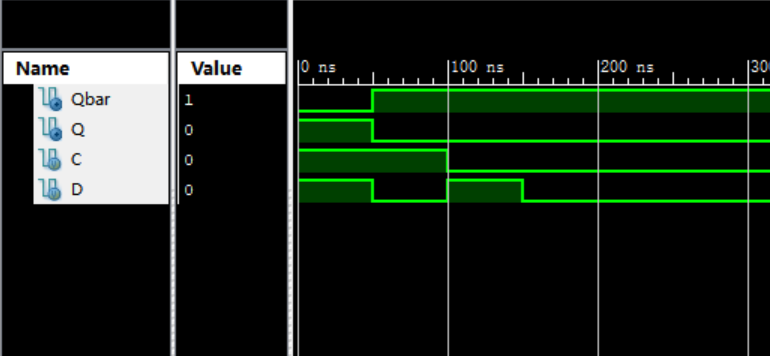
4.仿真（包含空翻）



图：D锁存器SCH图



图：仿真激励



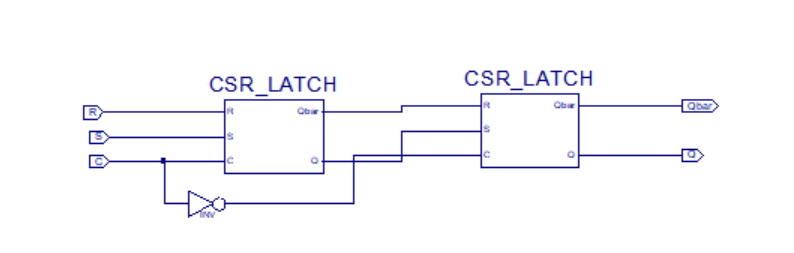
图：仿真波形

**4.SR主从触发器**

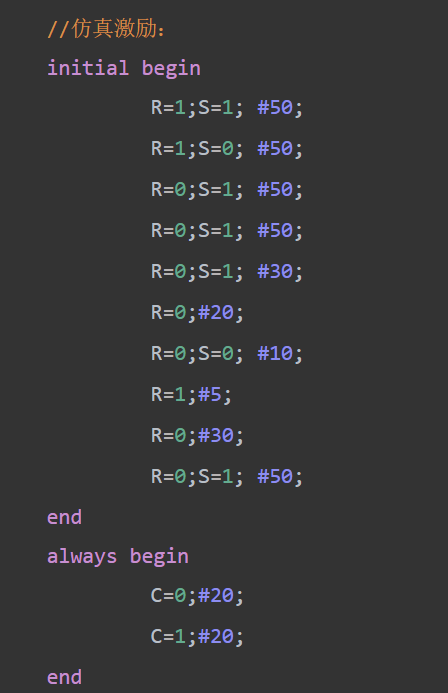
1.新建源文件MS\_FLIPFLOP.sch，用原理图方式设计

2.调用CSR\_LATCH实现

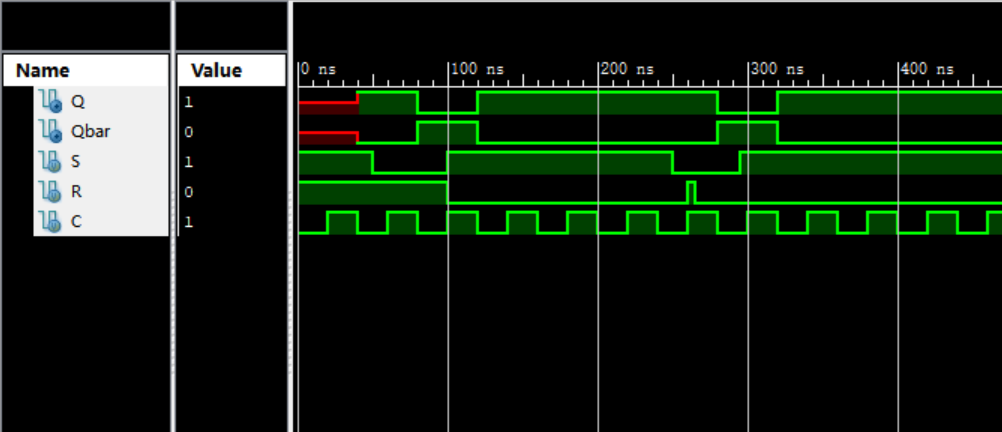
3.仿真（包含一次性采样）



图：SR主从触发器SCH图



图：仿真激励



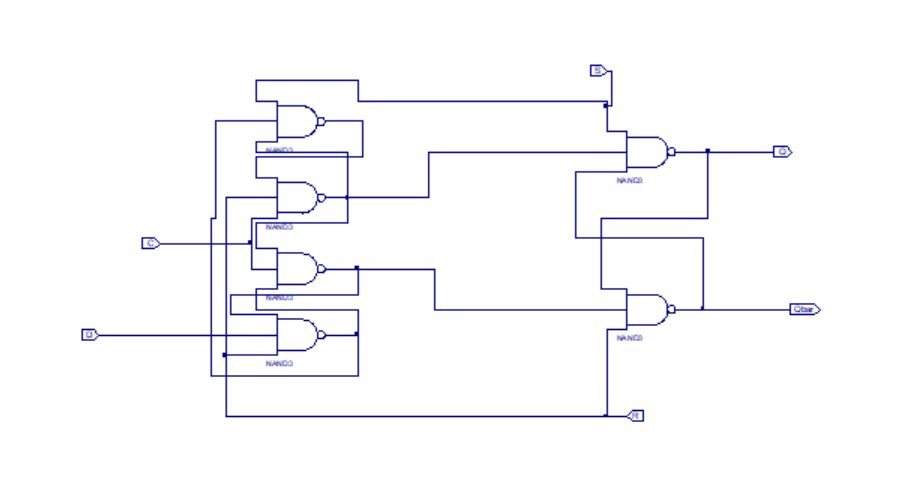
图：仿真波形

**5.D触发器**

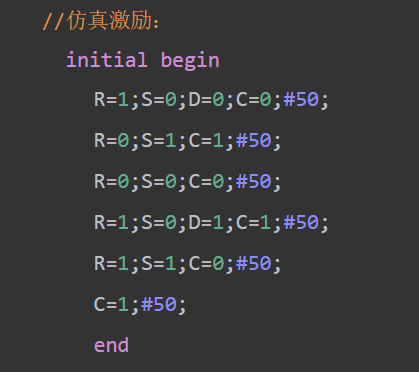
1.新建源文件D\_FLIPFLOP.sch，用原理图方式设计。

2.调用NAND3实现

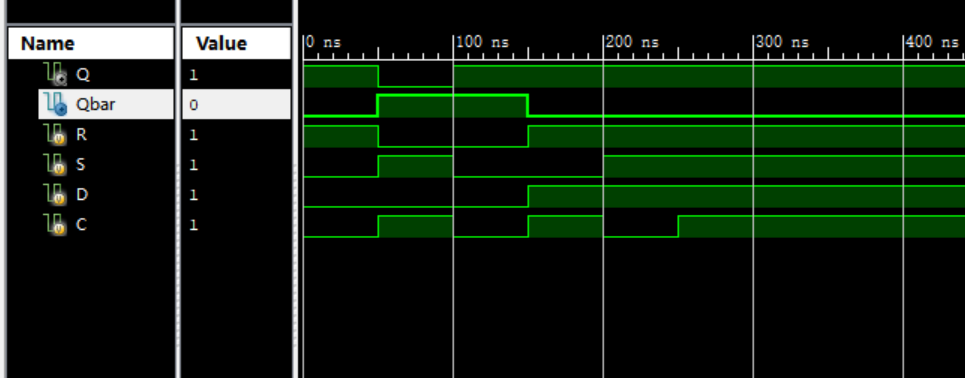
3.仿真



图：D触发器SCH图



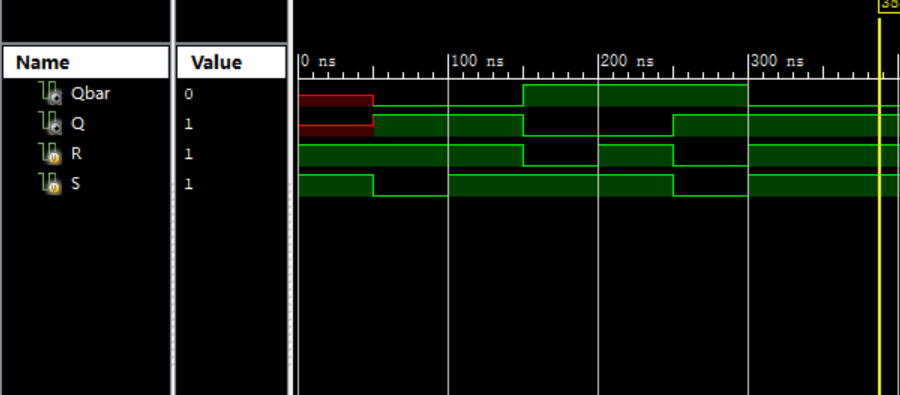
图：仿真激励



图：仿真波形

# 实验结果与分析

1. **基本SR锁存器**



由仿真图可知，0-50ns时，S=1，R=1,保持，但Q、Qbar未定义，所以值未知

50-100ns时，S=0,R=1，置1，所以Q=1,Qbar=0

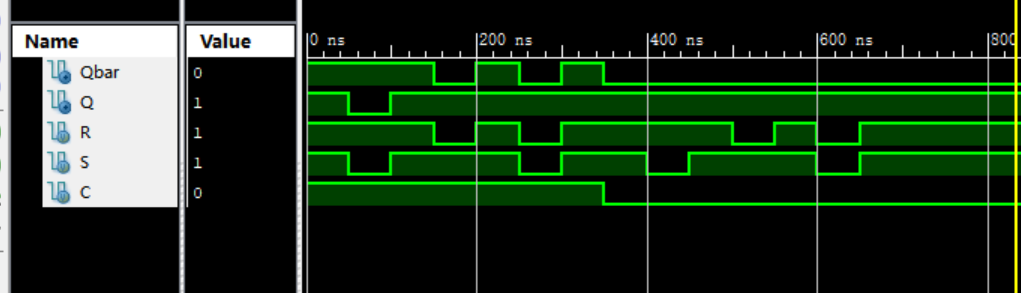
100-150ns时，S=1,R=1，保持，Q=1,Qbar=0不变

150-200ns时, S=1，R=0, 置0,所以Q=0，Qbar=1

200-250ns时, S=1，R=1,保持, Q=0,Qbar=1不变

250ns之后, S=0,R=0,未定义

1. **门控SR锁存器**



此处仿真的是夸张版本，为了更好地观测空翻现象

0-350ns时，C=1，此时门控SR锁存器可以接收数据，

300ns后，C=0，保持，Q和Qbar不变

0-50ns，S=1，R=1，未定义,Q=1,Qbar=1

50-100ns，S=0,R=1，置0，Q=0，Qbar=1,

100-150ns，S=1，R=1，未定义,Q=1,Qbar=1

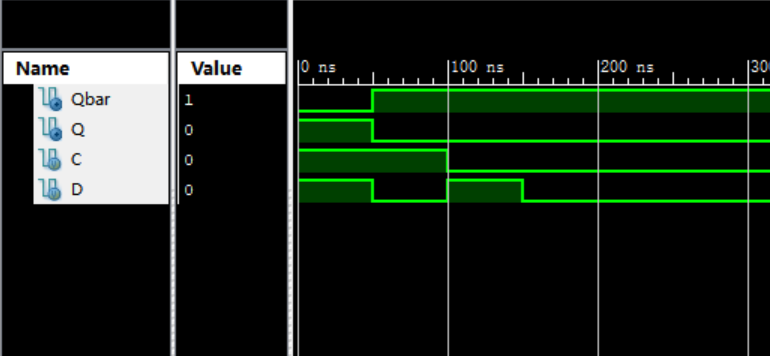
150-200ns，S=1，R=0，置1，Q=1，Qbar=0,

200-250ns，未定义,Q=1,Qbar=1

250-300ns，保持，但Q和Qbar会保持成为互反状态，因此Qbar此时的值变为0

300-350ns,未定义,Q=1,Qbar=1

1. **D锁存器**



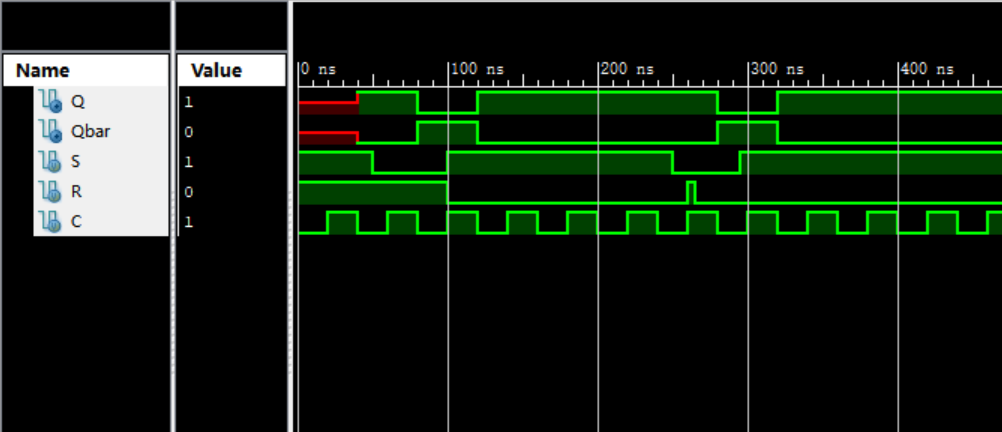
0-100ns时，C=1，此时锁存器可以接收到输入数据

100ns后，C=0，锁存器保持原状态

0-50ns，D=1，置1，Q=1,Qbar=0

50-100ns,D=0,置0，Q=0,Qbar=1

1. **SR主从锁存器**



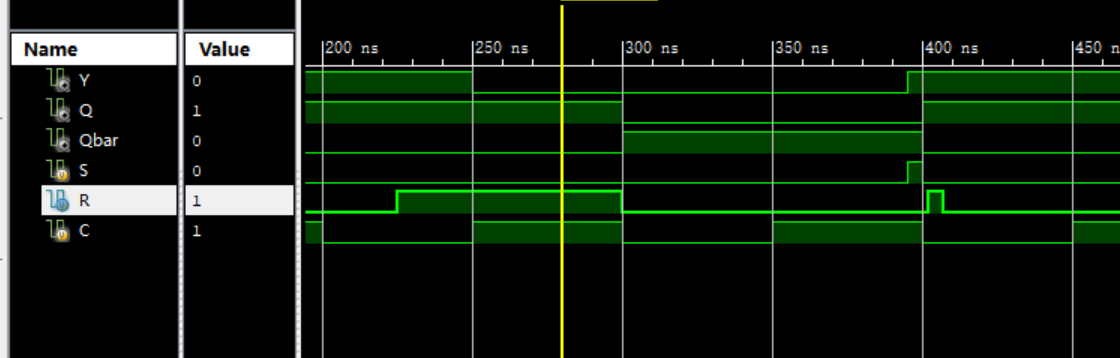
图中可以看出，Q的值与Qbar的值只有在C从1降到0的时刻发生变化

4末，S=0，R=1，置0，Q=0，Qbar=1

6末，S=1，R=0，置1，Q=1，Qbar=0

13末，读取S，R值，此时R产生一窄脉冲，窄脉冲对其产生影响，因此读入S=0，R=1

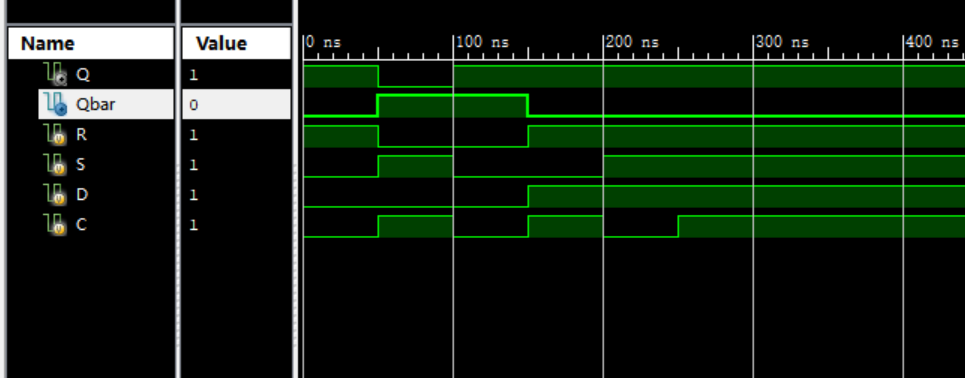
14末，按照S=0，R=1变化，Q置0，Qbar置1



图：一次性采样

由图中可以看出，Y没在下降沿便根据S值改变而产生改变，因此干扰传入了装置内

1. **D触发器**



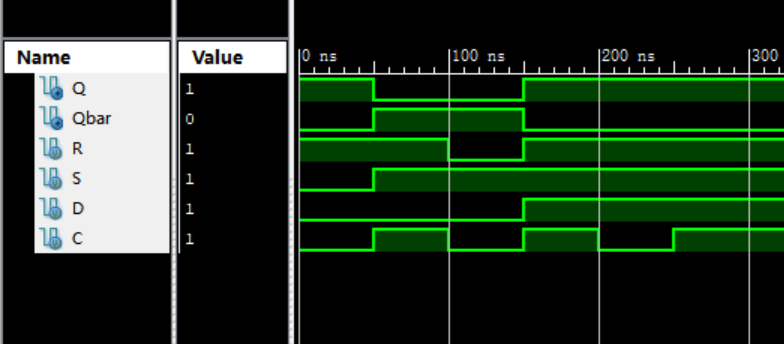
根据D的值，Q和Qbar的值仅在CLK的上升沿处变化（即C由0变化到1的时刻）。同时，Q和Qbar值也受到S与R的影响。

0-50ns时，S=0,R=1置1，所以Q=1，Qbar=0

50ns上升沿，D=0，所以Q=0，Qbar=1

100ns时，S和R值发生改变，S=0,R=0，与老师共同分析后，认为此时为不稳定态，此时Q=1，Qbar=1

150ns上升沿，D=1，Q=1，Qbar=0



对其又进行了一组仿真，可以看出，150ns处，S=1,R=1，上升沿处Q和Qbar值按D值进行改变。

# 讨论、心得

本次实验设计了多种锁存器和触发器，并对各个锁存器和触发器都针对其特点进行了多次仿真。

比如在CSR中将C设置成350ns转换一次从而更清晰的看到空翻现象，在实验过程中也发现了很多以前没有注意到的地方并与老师进行了进一步的分析，对各个锁存器的特点也有了进一步的了解。

通过此次实验，我对锁存器和触发器更加了解，期待早日将其运用到实践中。

# 实验11、同步时序电路设计实验报告

姓名： 王祚滨 专业： 信息安全 学号： 3180104933

课程名称： 逻辑与计算机设计基础实验 同组学生姓名： 王国朝、赵卿云

指导老师： 洪奇军 实验地点： 浙江大学紫金港校区东四教学楼509实验日期：2019年 11 月 27 日

# 一、实验目的和要求

1.掌握典型同步时序电路的工作原理和设计方法

2.掌握时序电路的激励函数、状态图、状态方程的运用

3.掌握用Verilog进行有限状态机的设计、调试、仿真

4.掌握用FPGA实现时序电路功能

**二、实验内容和原理**

2.1实验内容：

1. 4位二进制同步计数器

2.4位可逆二进制同步计数器

3.分频器

2.2实验原理：

4位二进制同步计数器

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***QA*** | ***QB*** | ***QC*** | ***QD*** | ***DA*** | ***DB*** | ***DC*** | ***DD*** |
| **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **0** |
| **1** | **1** | **0** | **0** | **0** | **0** | **1** | **0** | **0** |
| **2** | **0** | **1** | **0** | **0** | **1** | **1** | **0** | **0** |
| **3** | **1** | **1** | **0** | **0** | **0** | **0** | **1** | **0** |
| **4** | **0** | **0** | **1** | **0** | **1** | **0** | **1** | **0** |
| **5** | **1** | **0** | **1** | **0** | **0** | **1** | **1** | **0** |
| **6** | **0** | **1** | **1** | **0** | **1** | **1** | **1** | **0** |
| **7** | **1** | **1** | **1** | **0** | **0** | **0** | **0** | **1** |
| **8** | **0** | **0** | **0** | **1** | **1** | **0** | **0** | **1** |
| **9** | **1** | **0** | **0** | **1** | **0** | **1** | **0** | **1** |
| **10** | **0** | **1** | **0** | **1** | **1** | **1** | **0** | **1** |
| **11** | **1** | **1** | **0** | **1** | **0** | **0** | **1** | **1** |
| **12** | **0** | **0** | **1** | **1** | **1** | **0** | **1** | **1** |
| **13** | **1** | **0** | **1** | **1** | **0** | **1** | **1** | **1** |
| **14** | **0** | **1** | **1** | **1** | **1** | **1** | **1** | **1** |
| **15** | **1** | **1** | **1** | **1** | **0** | **0** | **0** | **0** |





图1 4位二进制同步计数器状态表

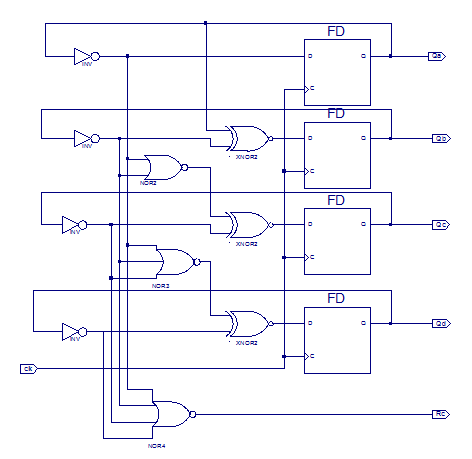


图2 4位二进制同步计数器原理图

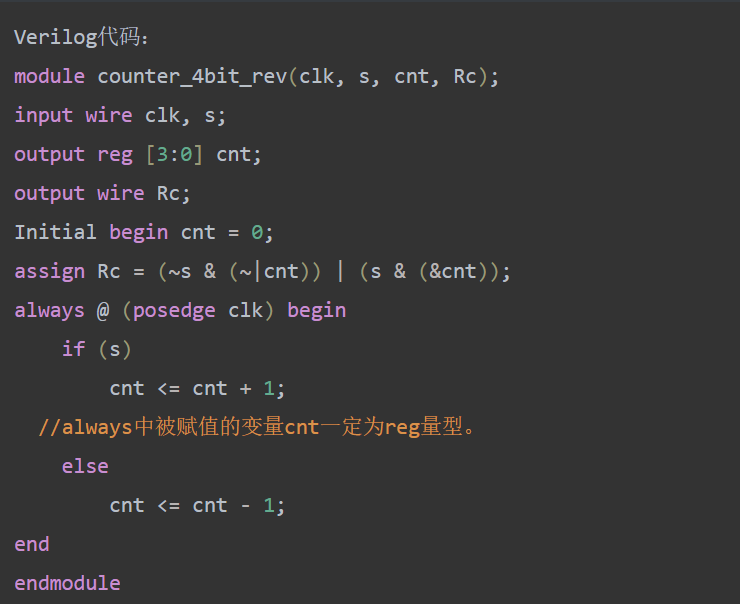
可逆二进制同步计数器

1.可逆二进制同步计数器通过控制端S选择正向或者反向计数

2.S = 1时，正向计数，各触发器逻辑表达式同前面

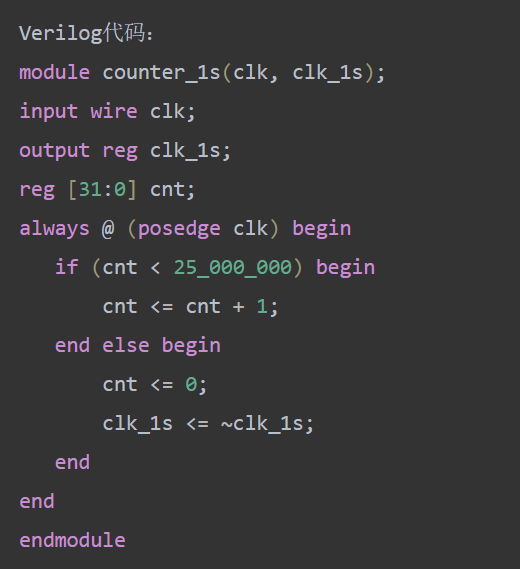
3.S = 0时，反向计数，各触发器逻辑表达式如下式





分频器设计

50MHz信号通过25,000,000次分频后，得到1Hz的秒脉冲方波，作为计数器的脉冲输入。



# 三、主要仪器设备

# 1. 装有Xilinx ISE 14.7的计算机 1台

# 2. SWORD开发板

# 四、操作方法与实验步骤

**任务1：原理图方式设计4位同步二进制计数器**

1.1.新建工程，工程名称用MyCounter。

1.2.Top Level Source Type用HDL

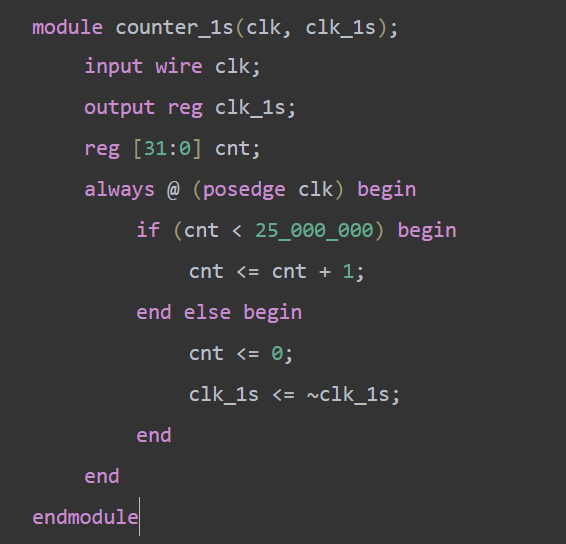
1.3.新建源文件，类型是Schematic，文件名称用Counter4b。

1.4.原理图方式进行设计

1.5.进行波形仿真

2.1新建源文件，用作时钟，类型是Verilog。

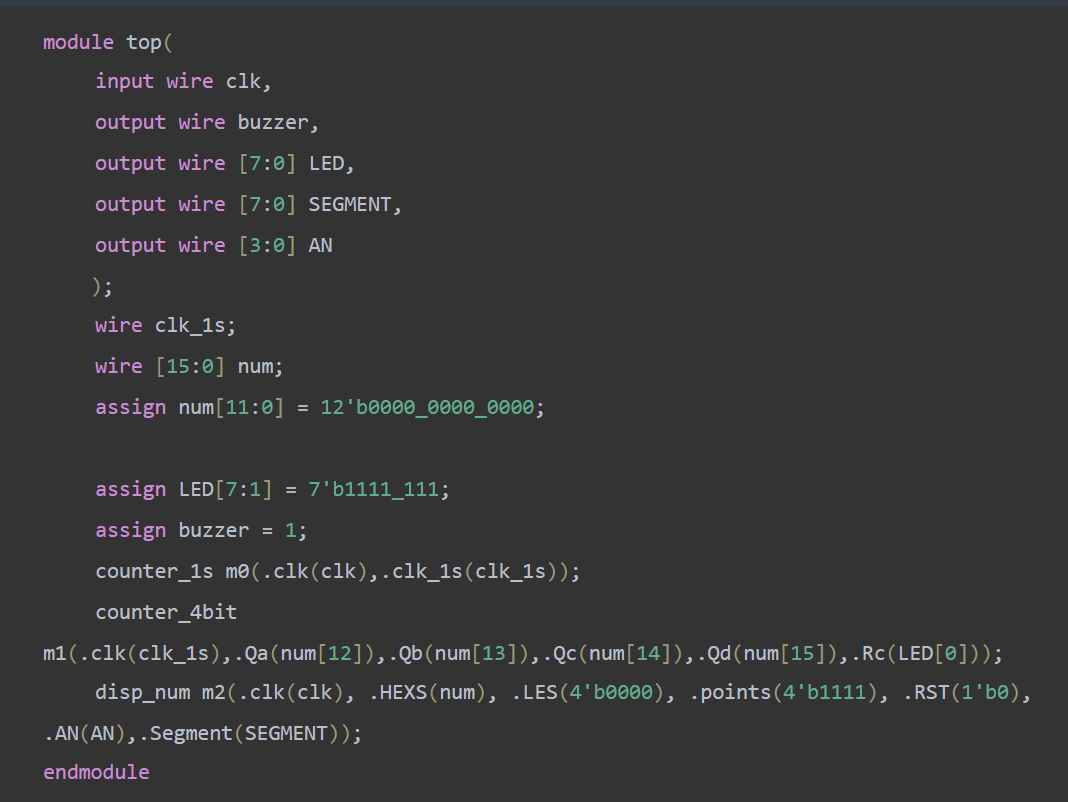
2.2文件名称用clk\_1s，Verilog行为描述。



3.1新建源文件，类型是Verilog，文件名称用Top。

3.2右键设为“Set as Top Module”，输入为clk（100MHZ）时钟

3.3每秒自增1//根据“分频器设计”程序得到1s时钟，显示在1位数码管上。Rc显示在LED灯上



**任务2：设计16位可逆同步二进制计数器**

1.1新建工程，工程名称用myRevCounter。

1.2Top Level Source Type用HDL，新建源文件，类型是Verilog，文件名称用RevCounter。

1.3结构化描述方式进行设计，波形仿真（包含正向计数和反向计数）。

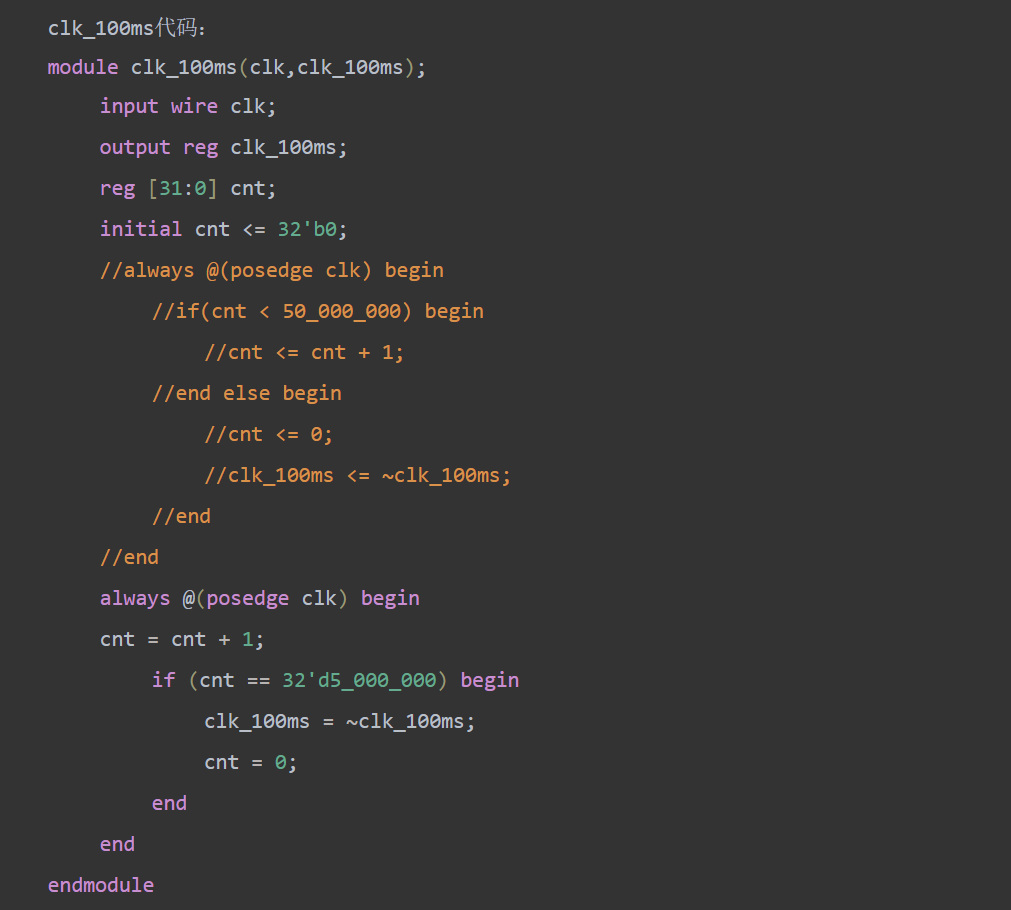
1.4. 新建源文件top，并右键设为“Top Module”

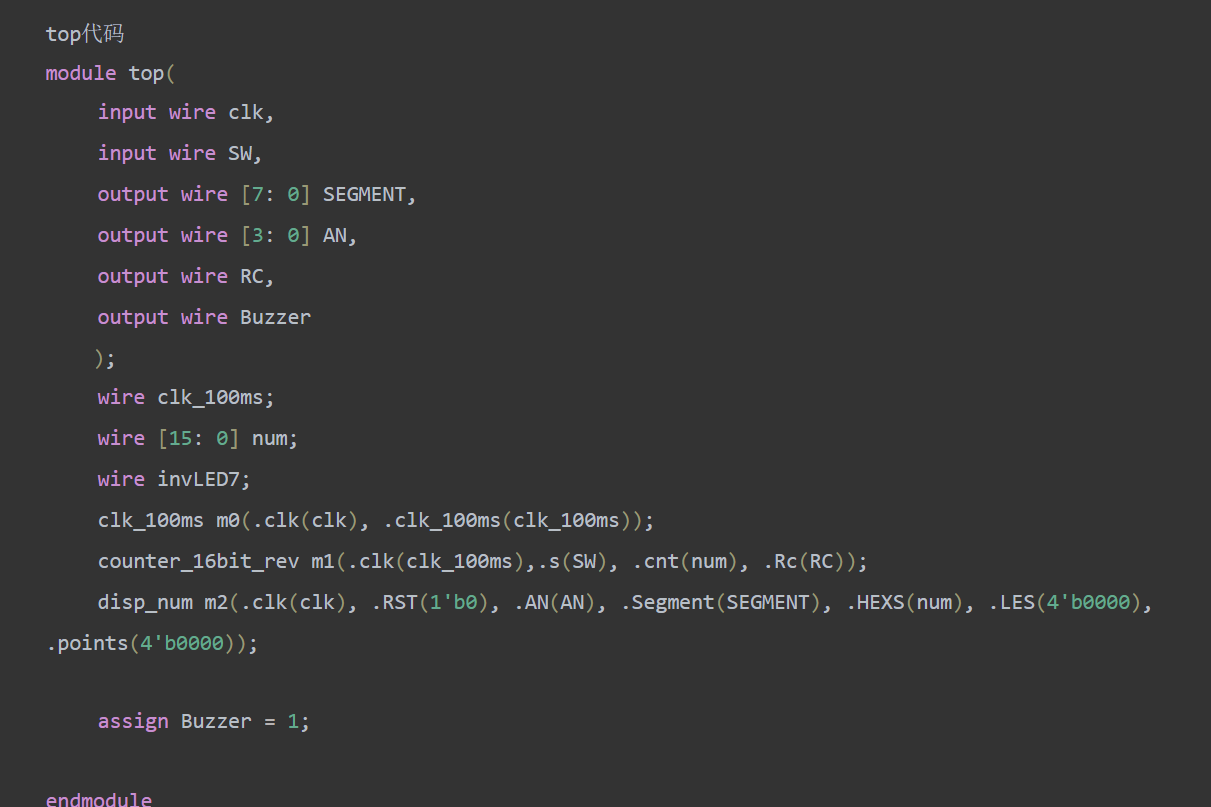
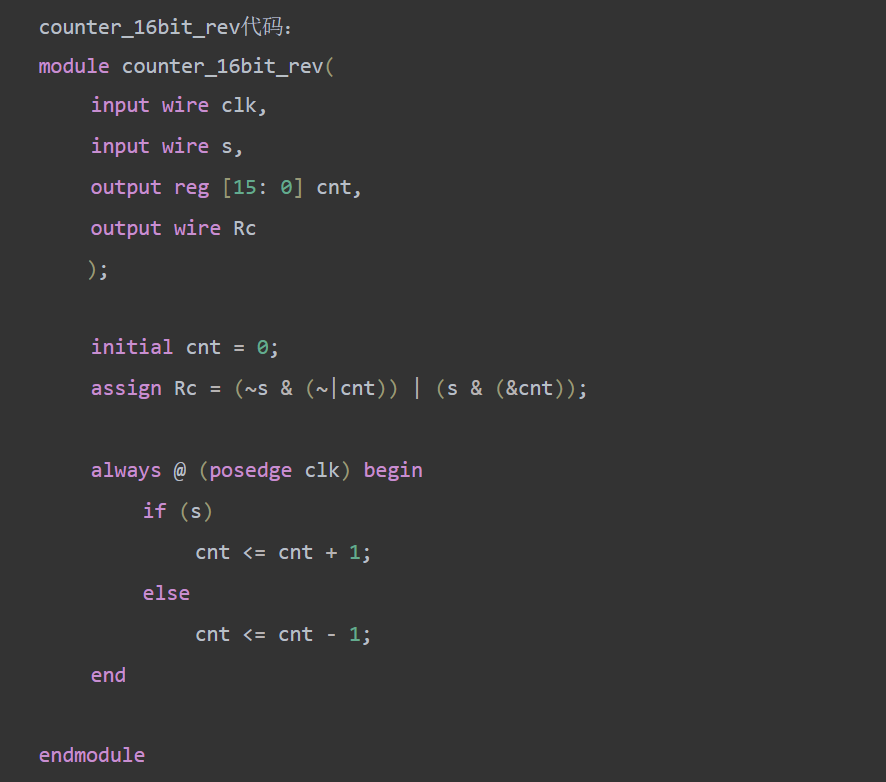
2.1新建源文件，设计100ms时钟，类型是Verilog，文件名称用clk\_100ms。

2.2Verilog行为描述

2.3用sw[0]控制自增/自减1（每0.1秒）

2.4显示在4位数码管上，Rc状态用LED灯来显示。



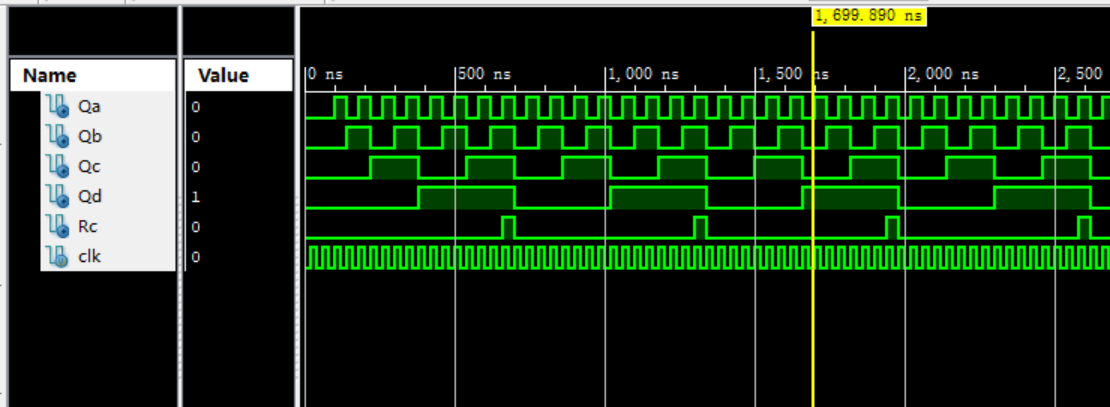


8. UCF引脚定义

# 五、实验结果与分析

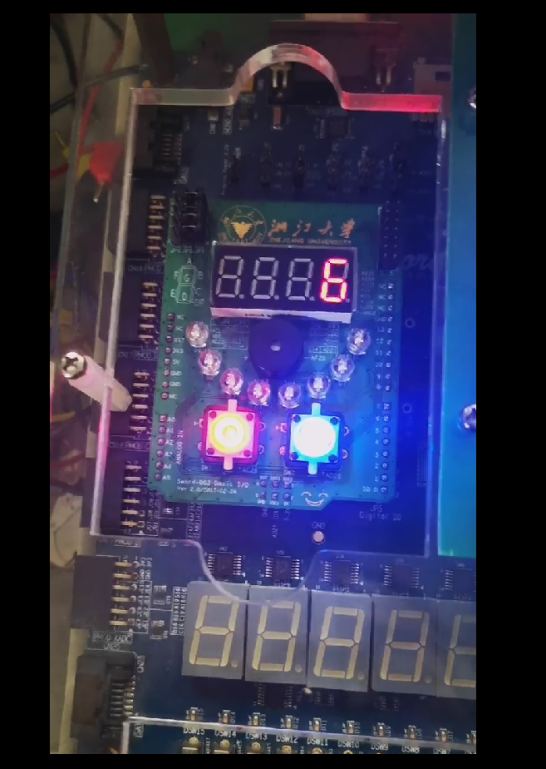
在part3和part4部分已经对代码进行了比较详细的介绍，故在此不做赘述。以下是实验结果。

实验一：仿真图





在板子上的结果：



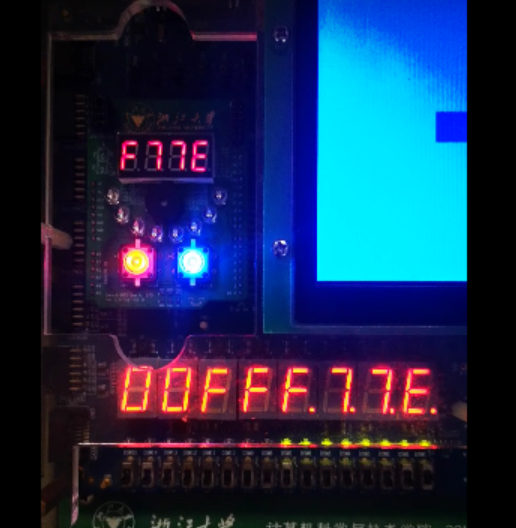
F->0时灯会闪烁

实验二：结果如下



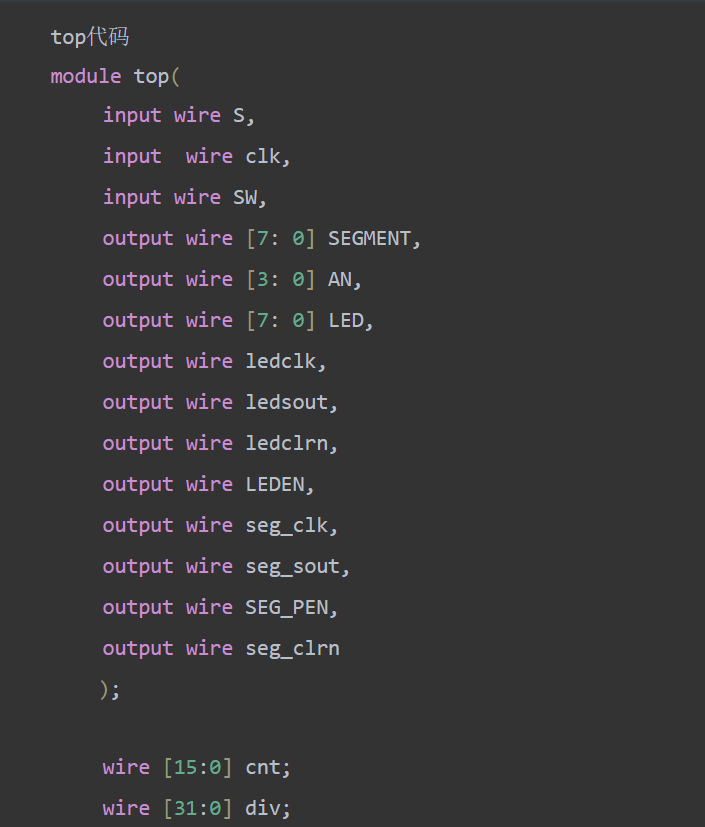
在跨过0000与ffff时，led灯会闪，表示产生rc

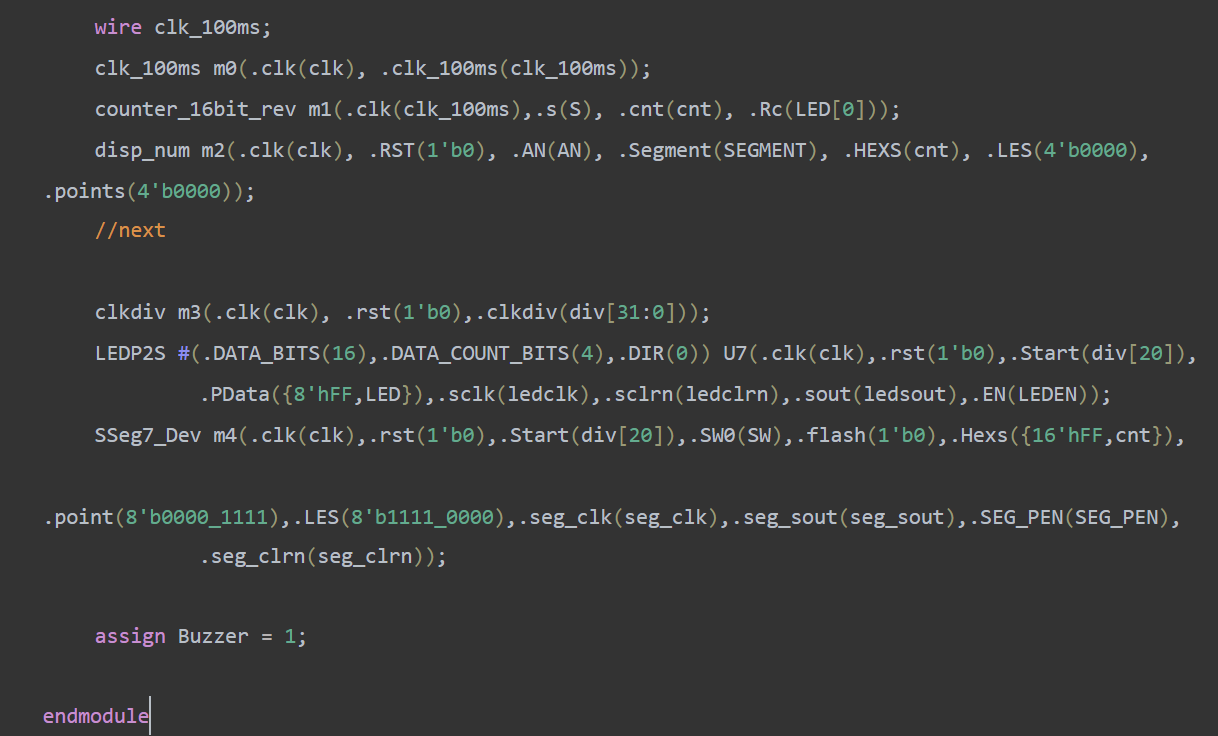
加分项：



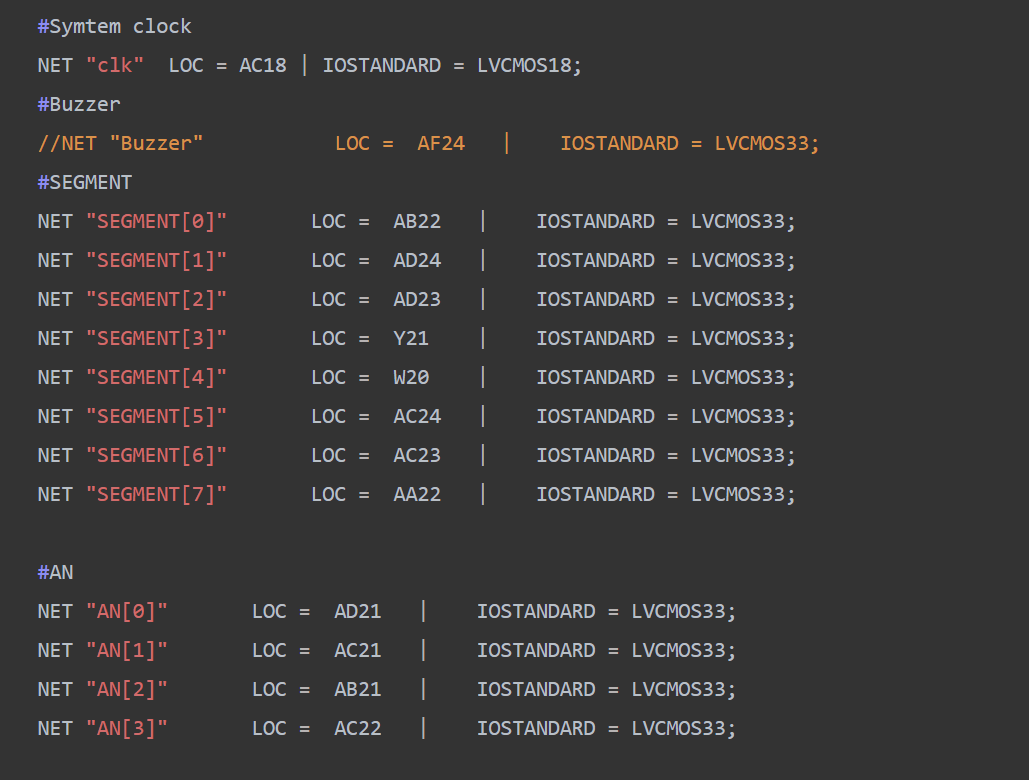
在下方也会产生同样效果，从FFFFFF变为000000时右下角的灯会闪烁表示进位。

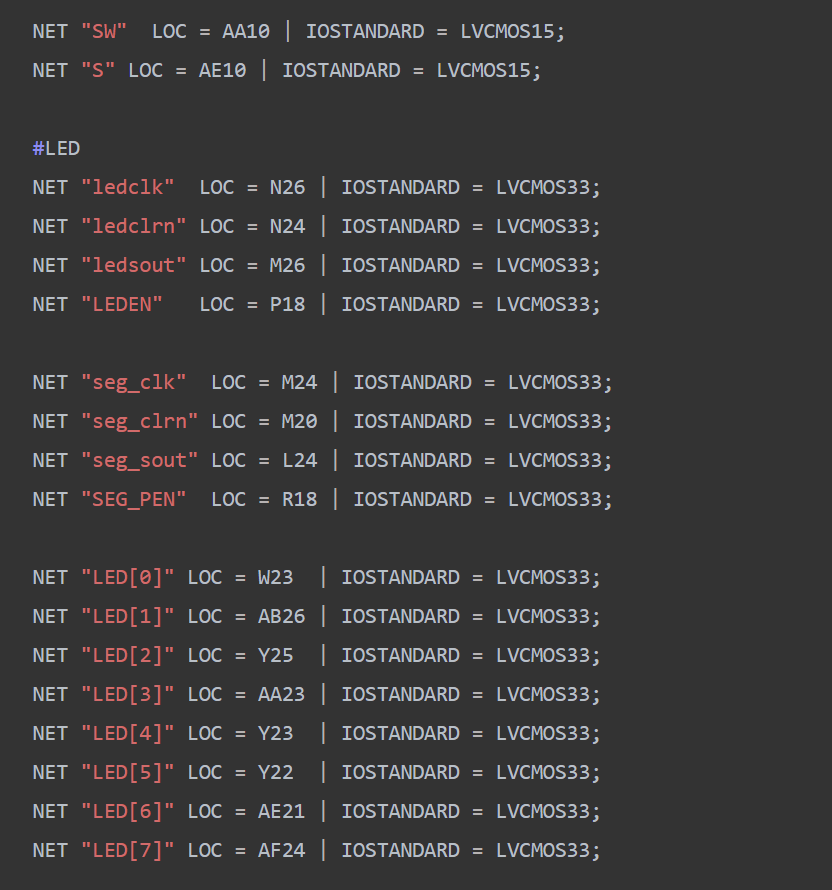
加分项的代码如下：





引脚代码





# 六、讨论、心得

此次代码任务较多，使我们对verilog语言有了进一步的认识，在实验后对verilog语法进行进一步查询，有了很大收获，整体实现难度不大，主要锻炼自己的coding能力。

同时，由于做这次实验时相应的课程老师还没有讲到，因此先对其原理有了一个比较直观的认知，相信在结合课本学习后会有更深的理解。

在课下又完成了加分项的操作，由于最初将SW[0]混用，导致下方总是不能出现正确结果，最后换了一个按钮，完成了实验。