**FPGA显示学号(Verilog代码实现)**

# 介绍

## 问题重述

实现一个可循环显示学号后8位数字的电路，每隔一定时间显示一位，显示到最后一位后停两秒再开始下一轮循环，要求实现学号显示左移和右移的控制，并拓展该电路其他功能，例如显示两位同学的学号，并给出仿真波形，编写测试代码。

## 问题理解

首先，应该在ego板上实现可同时显示学号后8位的数字电路，即板子上可观察到02341925（学号后8位），之后，实现学号的左移和右移的循环显示，拿02341925举例：

|  |  |  |
| --- | --- | --- |
|  | 左移 | 右移 |
| 显示1 | - - - - - - - 5 | 0 - - - - - - - |
| 显示2 | - - - - - - 2 5 | 0 2 - - - - - - |
| 显示3 | - - - - - 9 2 5 | 0 2 3 - - - - - |
| 显示4 | - - - - 1 9 2 5 | 0 2 3 4 - - - - |
| 显示5 | - - - 4 1 9 2 5 | 0 2 3 4 1 - - - |
| 显示6 | - - 3 4 1 9 2 5 | 0 2 3 4 1 9 - - |
| 显示7 | - 2 3 4 1 9 2 5 | 0 2 3 4 1 9 2 - |
| 显示8 | 0 2 3 4 1 9 2 5 | 0 2 3 4 1 9 2 5 |

* + - * 1. 循环显示学号

在之后，拓展功能，即先显示02341925，再显示02341929（二人学号），接着循环显示。

之后给出代码输出的显示波形即可。

# 完整显示学号

## 思路

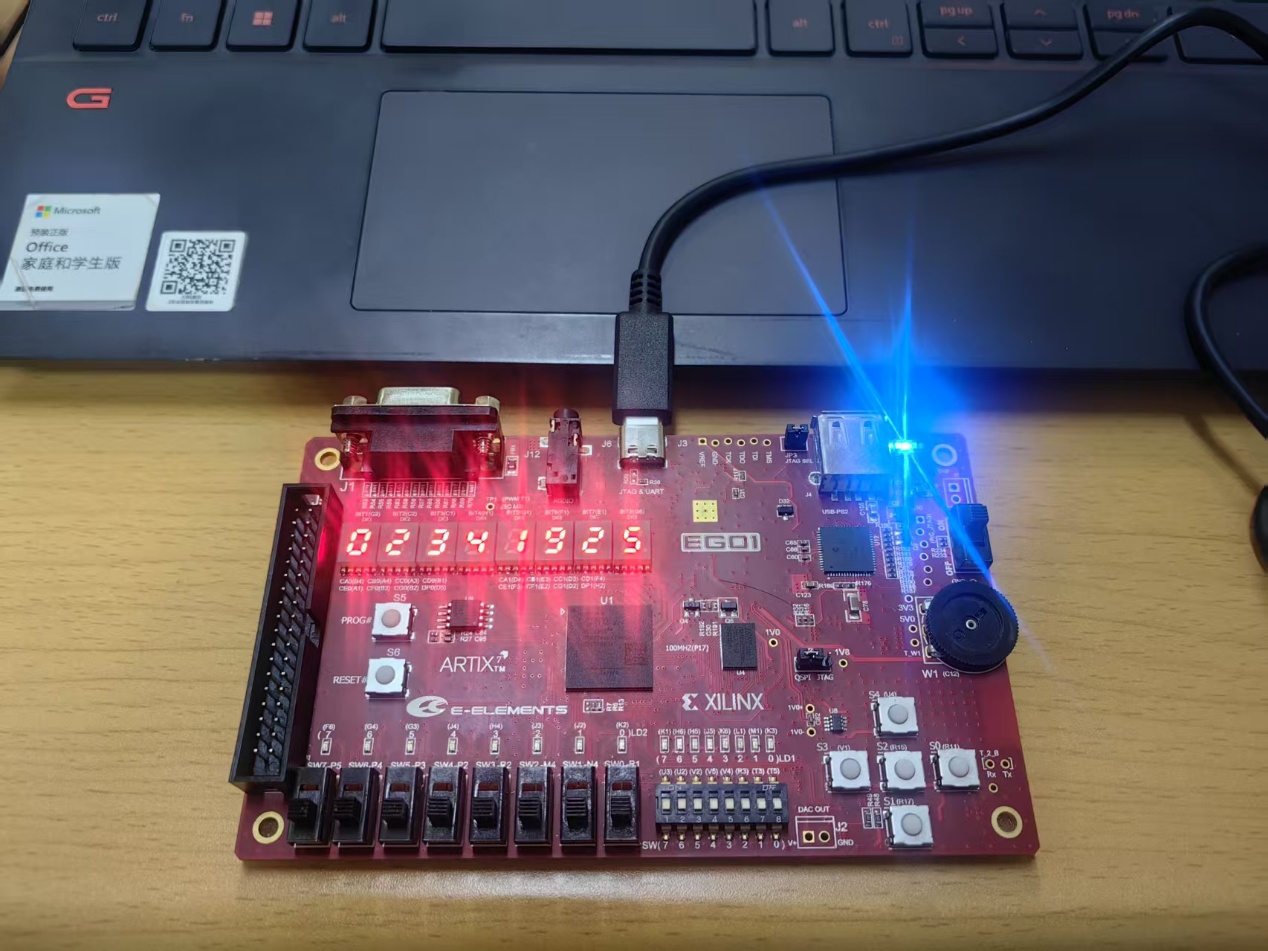
具体的代码和引脚见文件xuehao，这里只介绍思路。xuehao的参考资料见参考文献。

首先，ego1板子的前四位数码位和后四位数码位分别公用一套七段数码管（共阴极）。因此，无法同时做到八位数码位同时显示数字，只能引入高频时钟，数字随着时钟在数码位间高速变化，这样在人眼中它们就是同时显示的。

所以，该代码的关键就是引入了21位的clkdiv的时钟分频器，bitcnt=clkdiv[20:19]作为高速变化的数码位（2^19=524288，时钟主频为100MHz，得到扫描周期为5.24ms），an作为要显示的数码位，digit表示要显示的数字将它与七段数码管相联系（a\_to\_g）

如此，便完成了对x7\_4bit的编写，因为要使八个数码位同时显示，故x7\_4bit函数要调用两次，每次对应与不同的数码位，要用不同的七段数码器。

## 结果



完整学号

# 滚动循环显示学号

## 思路

相关代码见xuehao2，这里只给思路。

xuehao2代码是依据xuehao1代码进行修改的，题目有具体时间即1s、2s的要求，而主时钟频率100MHz，即周期为1/100\_000\_000s，故先用sec\_cnt计数（每当主时钟上升沿到来时加1），当sec\_cnt记到100\_000\_000-1时，sec\_tick+=1，sec\_tick就表示1s的时间脉冲了。

除时间外，还要考虑每隔一秒左移或右移一位，这里用show\_step表示当前显示到第几位，delay\_cnt用于显示完成后隔两秒再继续循环。具体实现是当show\_step<8时，每次sec\_tick到达就要show\_step+=1；到达8时（数码位均有显示），delay\_cnt作为计数条件当计数到2便会使show\_step和delay\_cnt重新置零，完成一次循环。

为了在一个特定数码位输入对应的数字，可以创建类似c++的数组，利用for循环完成该步骤，即在该索引（位置）下填入对应的数字。例如，左移时display[7 - i] = id\_bcd[7 - i]，右移时display[i] = id\_bcd[i]，之后的操作和xuehao代码类似，不再赘述。

## 结果



循环显示学号（请双击该图片观看视频）

# 两个学号交替显示

## 思路

相关代码见xuehao3，这里只讲思路。

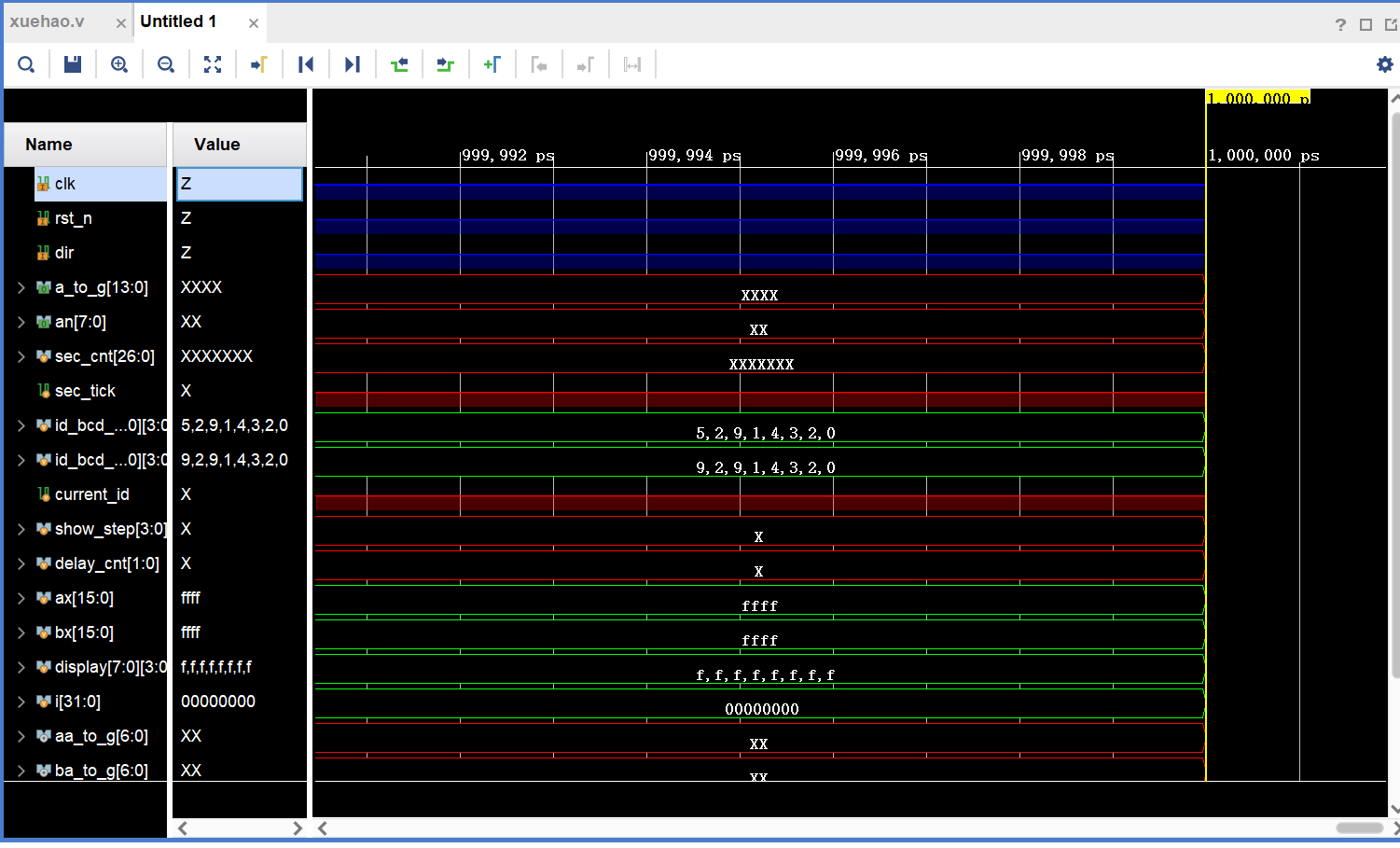
多加一个id\_bcd\_2，current\_id用来判断是学号①还是学号②，在show\_step和delay\_cnt这段代码中引入current\_id，当第一个人的学号展示完时，通过current\_id <= ~current\_id切换学号，后面的display部分也要相应修改（多增加if语句即可）

## 结果



两个学号显示（双击图片）

# 仿真波形



仿真波形

# 参考文献

[vivado 数码管学习（二）数码管显示一位数字和显示八位数字-CSDN博客](https://blog.csdn.net/shenqingdi/article/details/109609231)