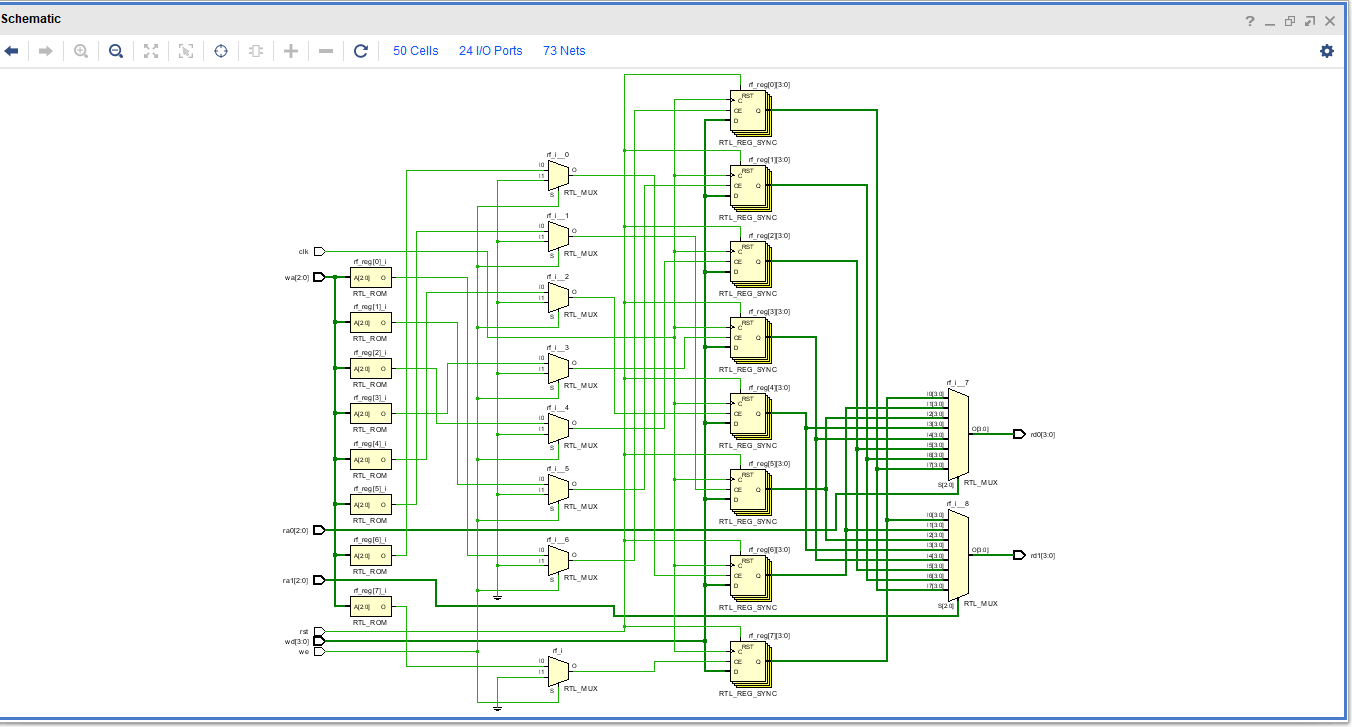
Lab3 寄存器堆与计数器

1. 设计逻辑
2. 寄存器堆

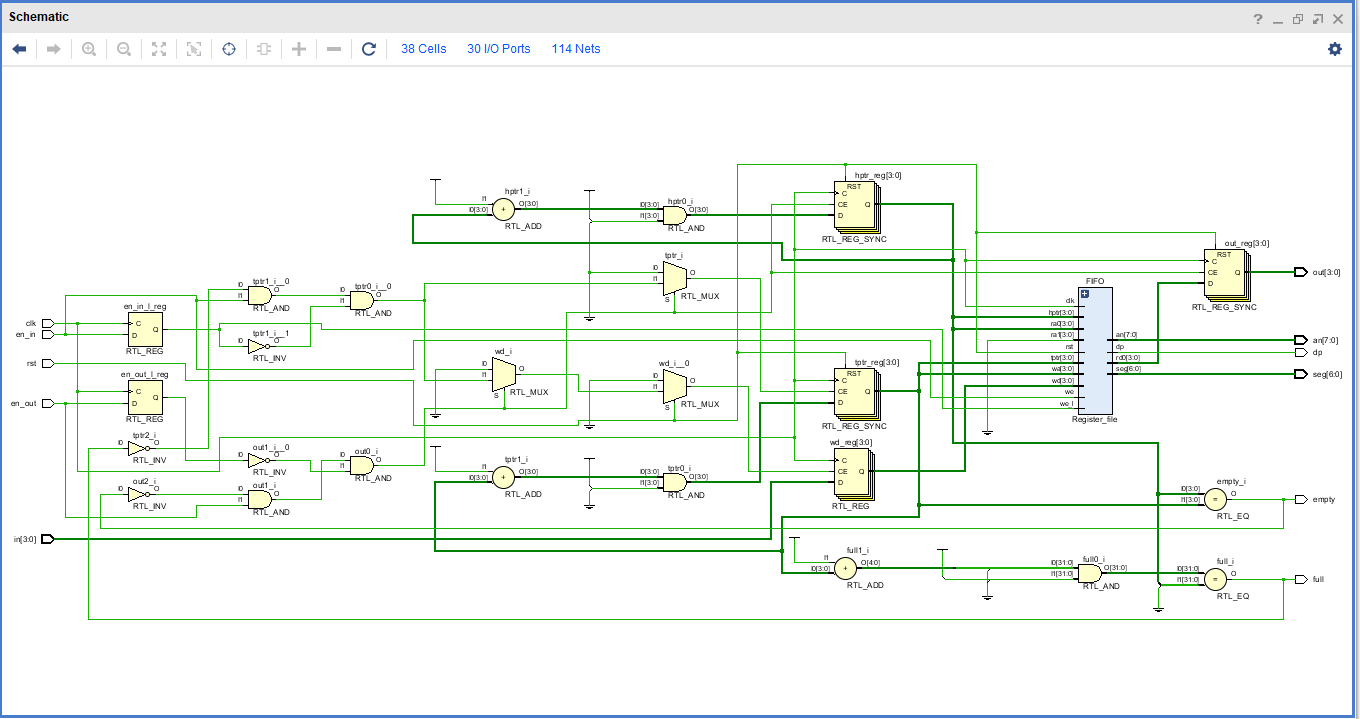
设计8\*4的寄存器堆，8个四位寄存器，wa为写入地址，wd为写入数据，ra0 ra1为读出地址，rd0 rd1为读出数据，we为写入使能，we=1时可以写入，时钟上沿时输入，rst将寄存器中数据清零。



数据通路

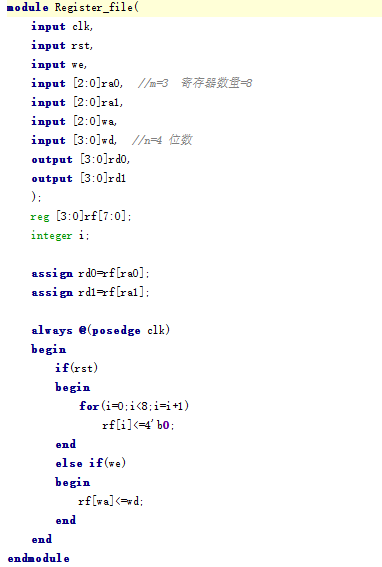
1. FIFO循环队列

设计长度为8的FIFO循环队列，使用头指针hptr和尾指针tptr标记队头和队尾元素，当队空时，hptr=tptr，当队满时，hptr=（tptr+1）%8.由于时钟分频只用来控制数码管，于是使用两个变量en\_in\_l和en\_out\_l来表示上一个时钟周期时en\_in和en\_out的值，当队列没有满且en\_in=1,en\_in\_l=0时，元素入队，当队列没有空且en\_out=1，en\_out\_l=0时出队。rst清空寄存器堆中值。同时用hptr和tptr的相对位置来判断8个七段数码管哪一个亮。即头指针尾指针中间的元素让七段数码管亮。把头尾指针扩展成4位实现循环队列，写入时，hptr+1，且hptr=hptr%8，读出时，tptr+1，且tptr=tptr%8.

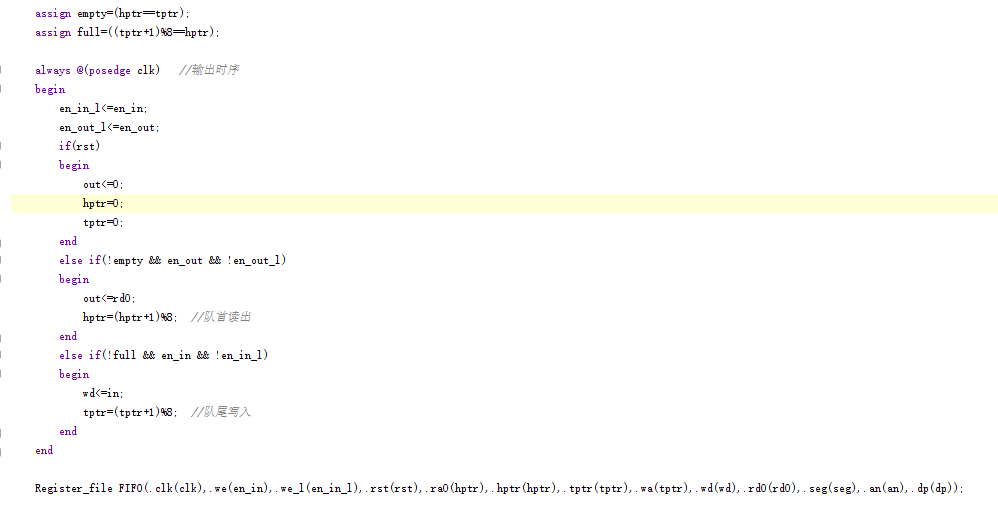


数据通路。

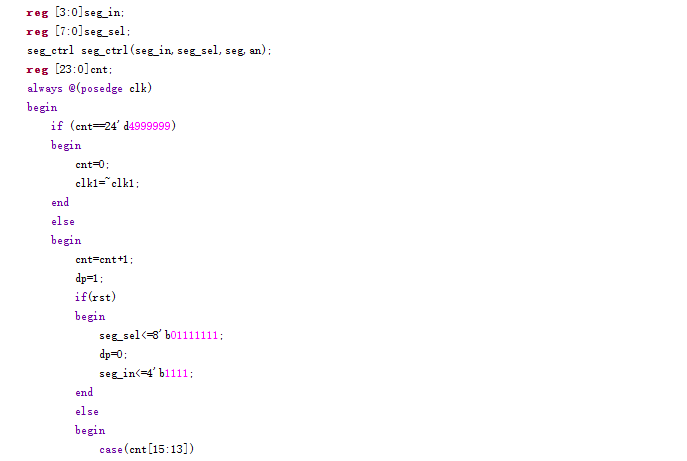
1. 核心代码



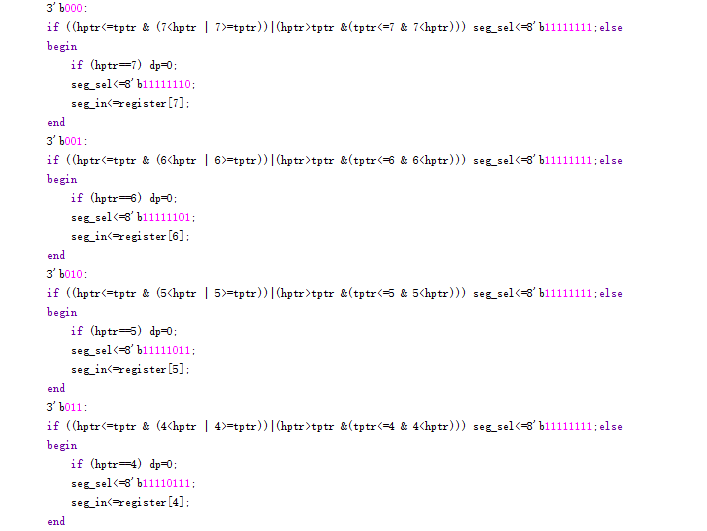
寄存器堆，使用二维数组，rf[wa]<=wd将值写入寄存器堆

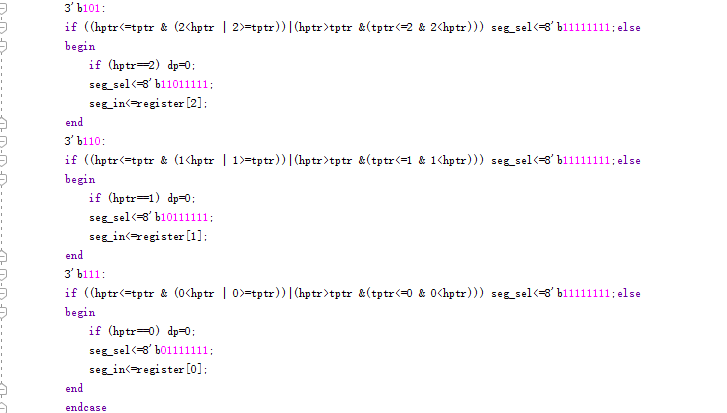


FIFO队列逻辑，rst时将out和头尾指针清零，并在相应条件下写入和读出数据



对100MHz时钟分频以满足七段数码管使用频率，并在rst时使得最左端的小数点亮





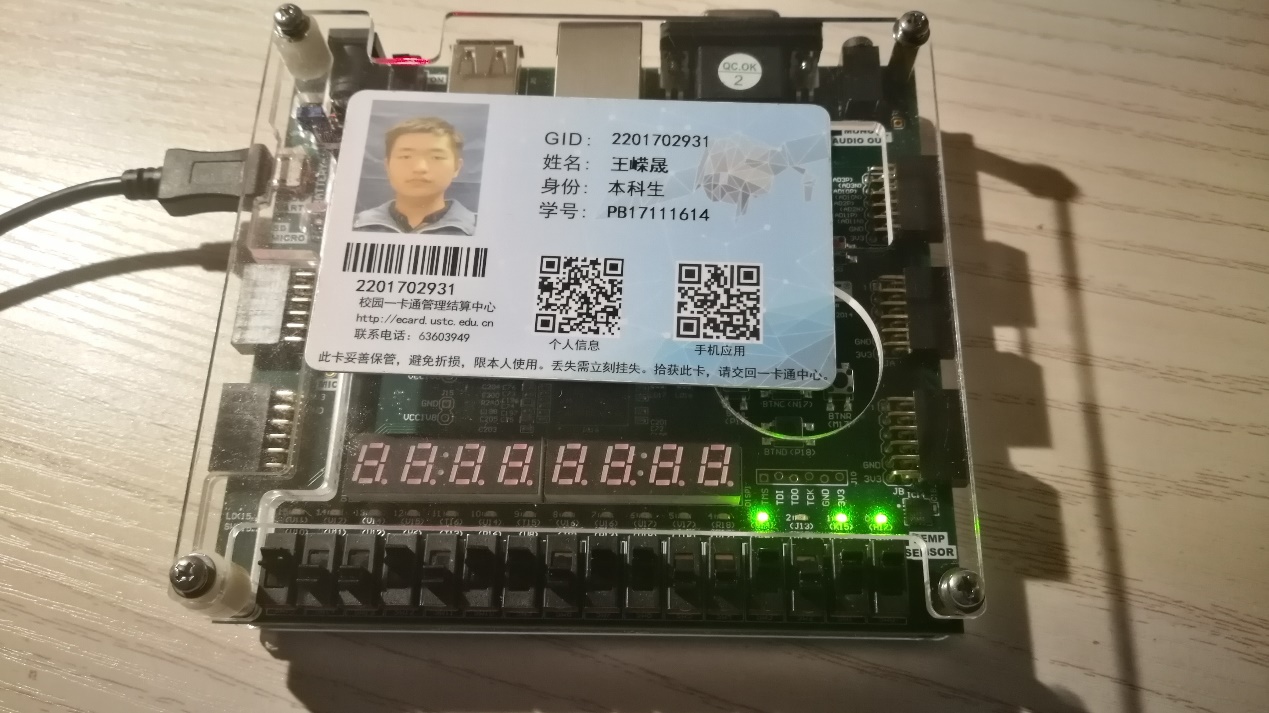
判别哪个七段数码管亮的代码

1. 仿真结果与下载结果：

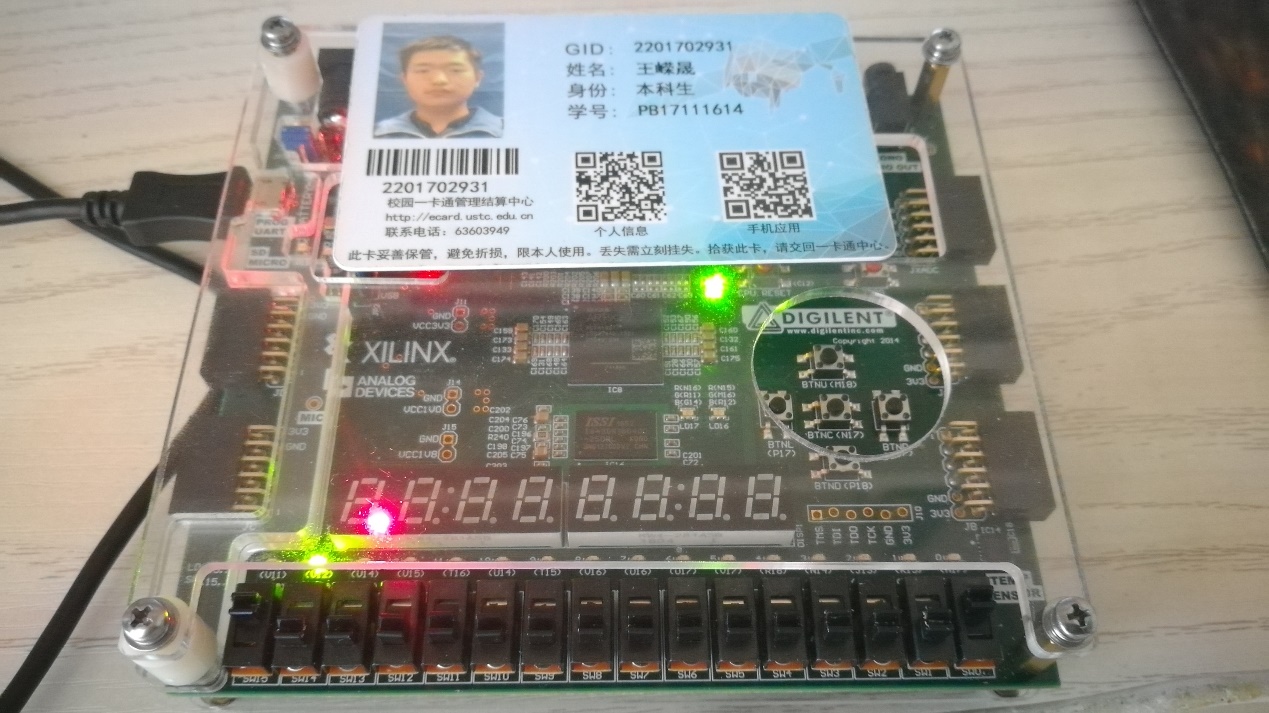
下载结果：



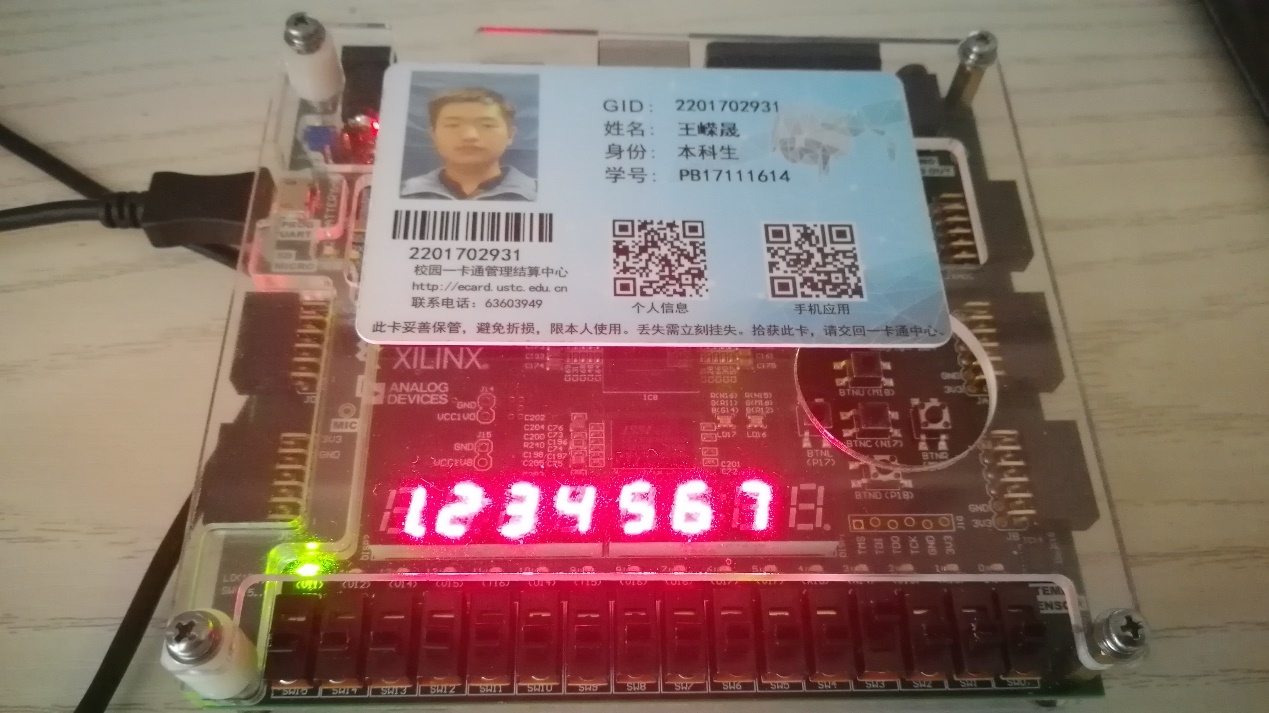
Wa=010,wd=1110,ra0=010,rd0如led显示所示



Wa=011 wd=1011 ra0=011 rd0如led显示所示



Rst，队空，只亮小数点



入队1234567，队满，1位队头

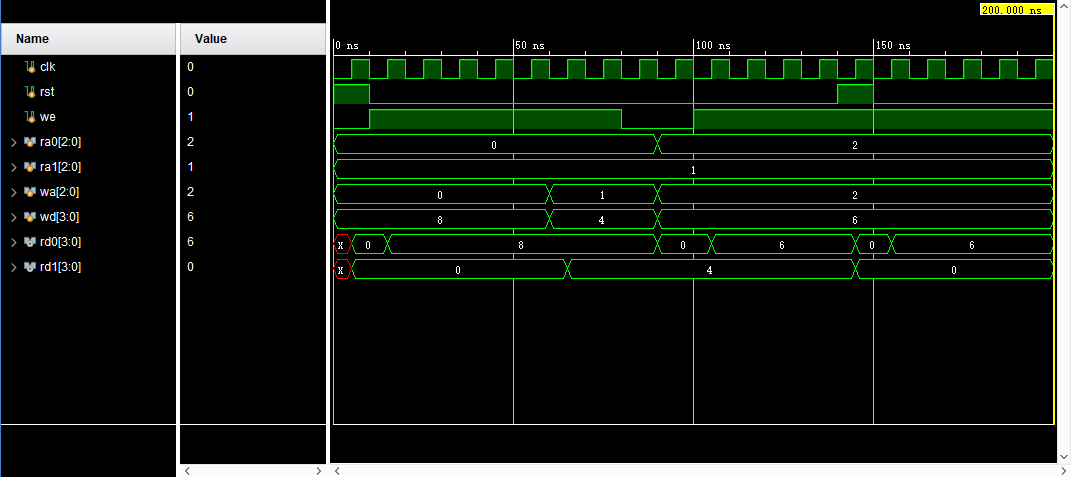


1出队，2为队头

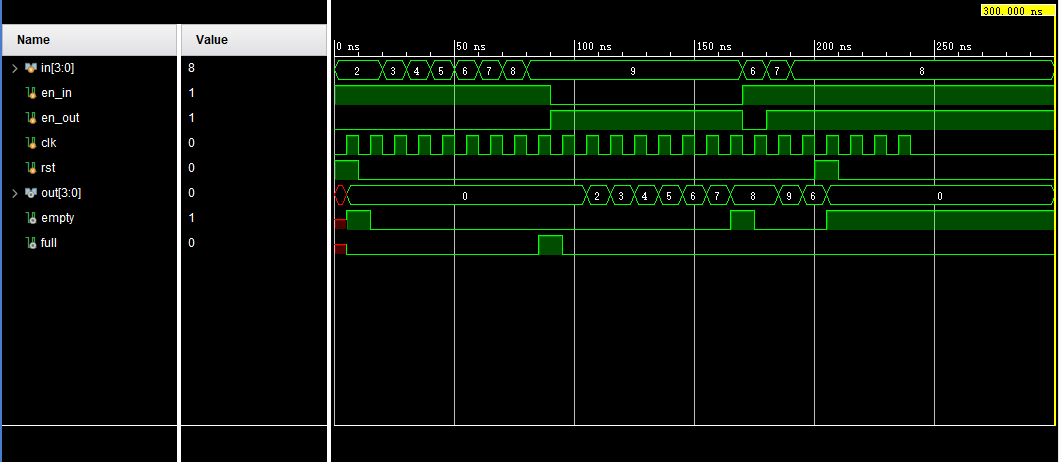


验证为循环队列

仿真截图：

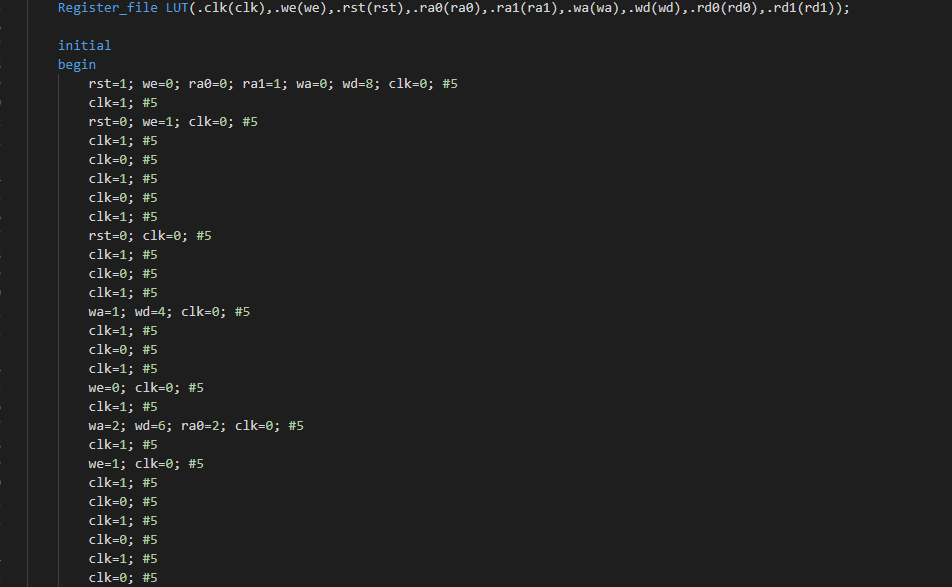


寄存器堆

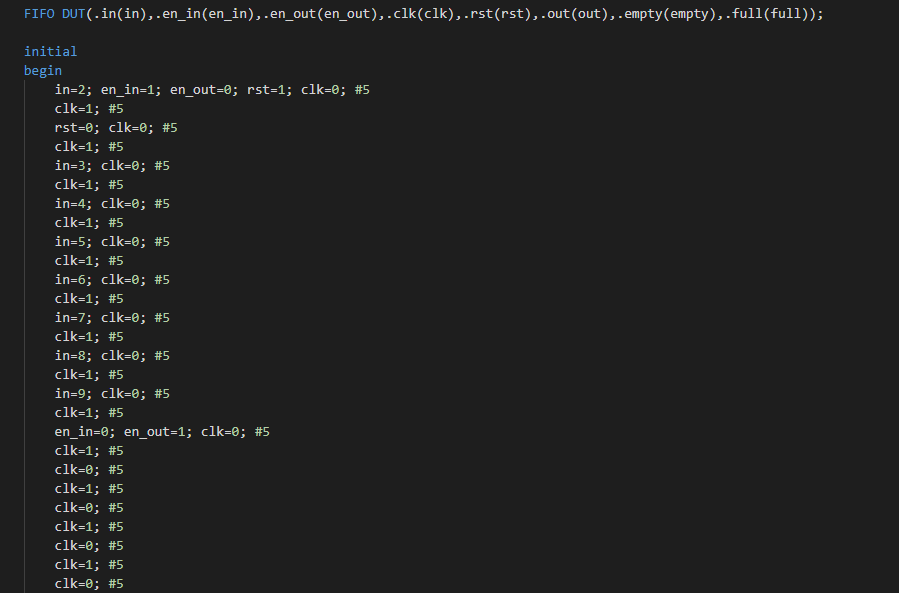


FIFO循环队列

1. 仿真代码：



寄存器堆仿真代码



FIFO仿真代码

1. 结果分析

仿真和下载结果正确，但是FIFO队列有一个状态延迟

1. 实验总结

本次试验学习了寄存器堆的设计以及用寄存器堆实现FIFO循环队列。实现FIFO循环队列时使用七段数码管，难点在于使用板载时钟分频只能用于七段数码管，如何控制输入输出使能信号在时钟周期中单次有效。FPGA开发板似乎有一点小问题，不过更改了按键后下载结果勉强可以通过。

1. 意见/建议：

实验要求不是很具体，导致操作过程中可能会有一些意想不到的问题。