**计算机体系结构实验课程第三次实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | 数字逻辑电路设计实战 | | | 班级 | 张金老师 |
| 学生姓名 | 袁田 | 学号 | 2314022 | 指导老师 | 董前琨 |
| 实验地点 | 实验楼A306 | | 实验时间 | 2025.3.26 | |

**1.实验内容说明**

完成本章的学习后,读者应完成以下三个实践任务:

1. 寄存器堆仿真

针对任务一寄存器堆实验，在项目中导入design source和simulation source后进行仿真观察波形。

1. 同步,异步RAM仿真，综合实现

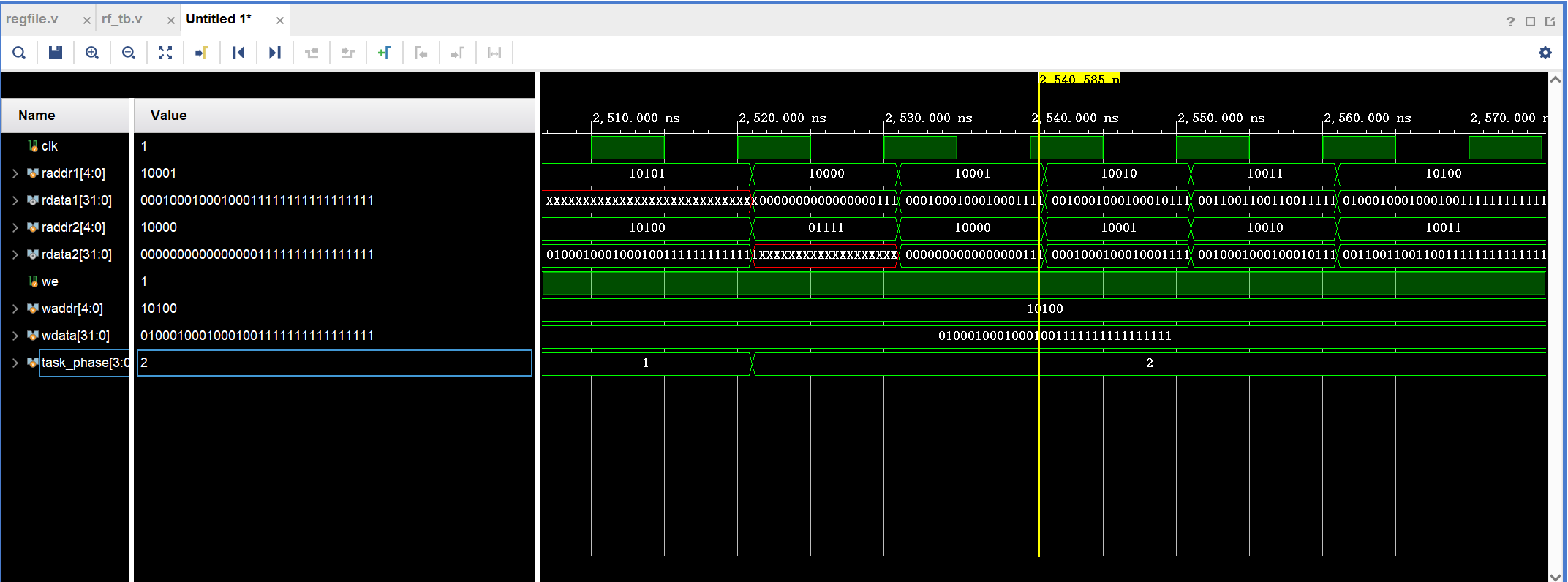
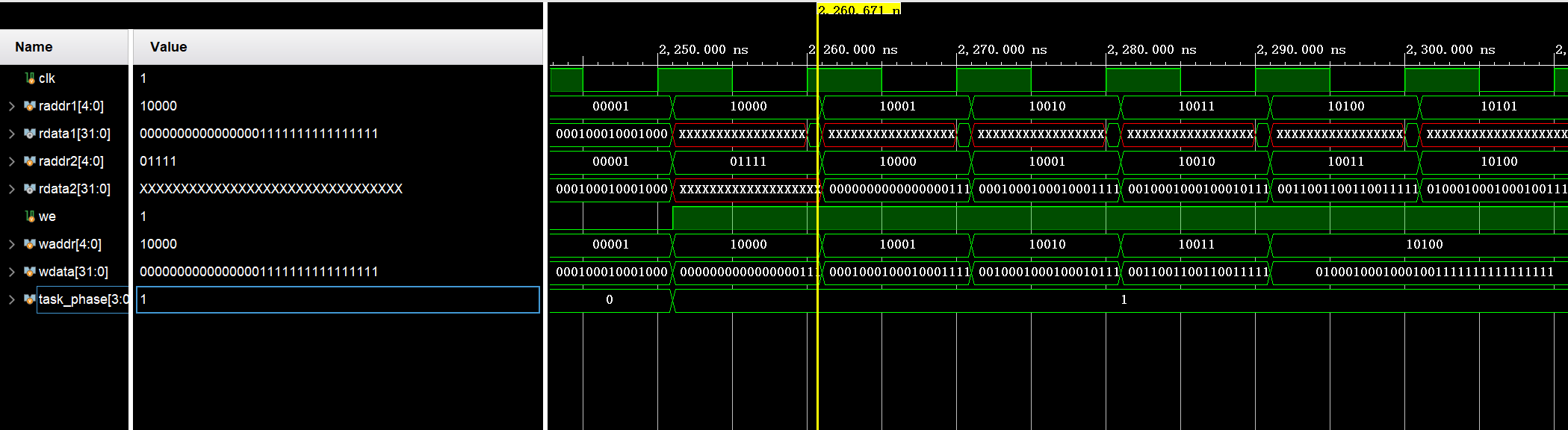
针对任务二同步ram和异步ram实验，可以参考实验指导手册中的存储器实验，注意同步和异步需要分开建工程，然后仿真，在感想收获中分析同步ram和异步ram各自的特点和区别。

1. 数字逻辑电路的设计与调试

针对任务三，介绍清楚发现bug、修改bug和验证的过程，总结使用vivado调试的经验步骤。

**2.实验结果分析**

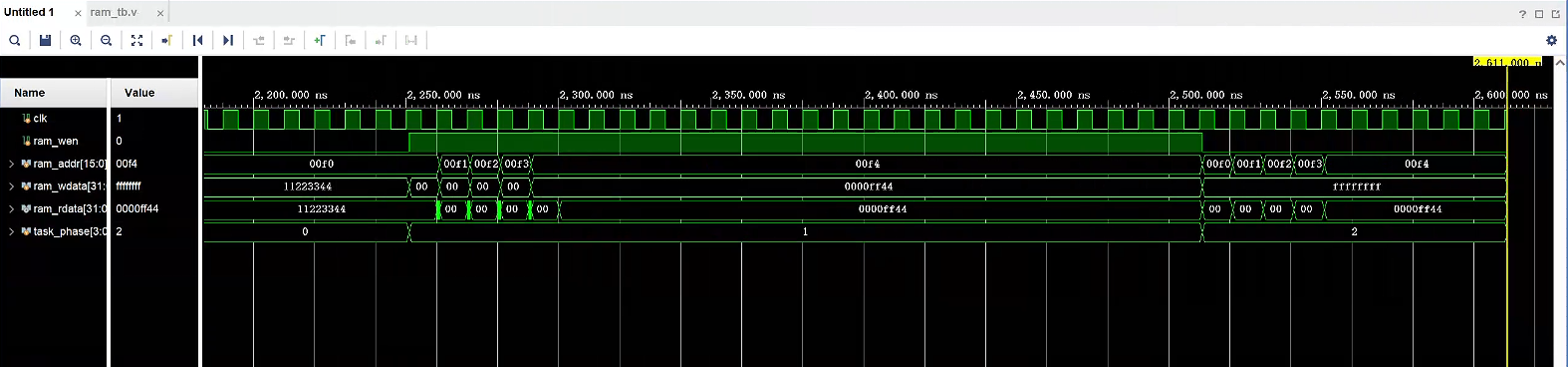
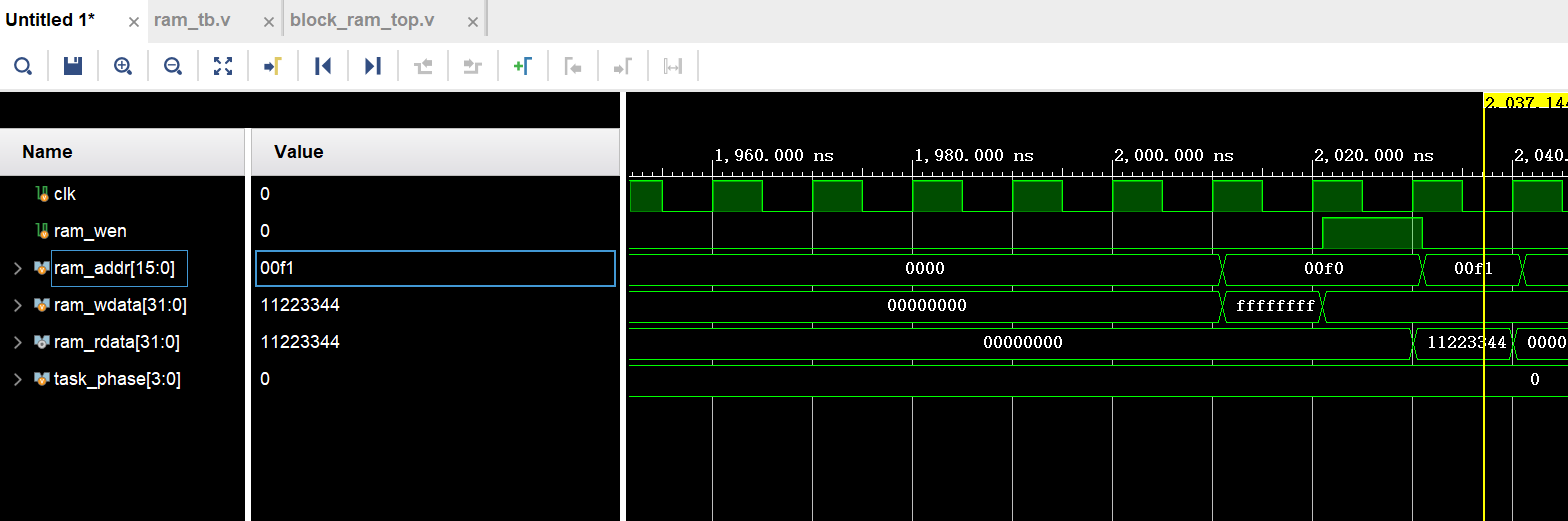
(a)任务一：寄存器堆仿真

在仿真文件中可以看到，除了初始化外进行了三个阶段的仿真，仿真结果如下：

在第零阶段中，验证了基本的读写操作(如写入使能端we=0和we=1的情况)，发现由于写入操作在时钟上升沿时完成，如果读入raddr=waddr，那么需要在下一周期写入完成后才会读取准确的数据；在第一阶段中，测试了寄存器连续写入和读取操作，由于每次改变raddr1和raddr2都进行加一操作，在修改写入的数据和地址后，如果raddr和waddr不同，则rdata立即返回raddr对应的寄存器的值。如果raddr和waddr相同且 we=1，则在下一周期才返回新的写入值；在第二阶段中，主要是将第一阶段写入的值进行读取，以验证数据是否可以保存于寄存器中。

(b)任务二：同步ram和异步ram实验

在同步ram和异步ram实验中，仿真实验同样分为三个阶段，分别检验了其读写功能(利用ram\_wen的值来判断进行读写操作)，检验连续写入多个地址的能力和检验数据保存的能力，下图为两个实验分别的仿真实验情况：



(c)任务三：数字逻辑电路的设计与调试

首先，由于本项目的是实验功能为：

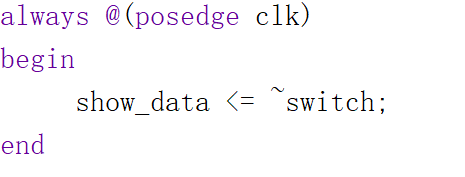
(1)获取开发板最右侧4个拨码开关的状态（记为“拨上为1，拨下为0”，实际开发板上拨码开关的电平是是拨上为低电平而拨下为高电平）,共有l6个状态（数字编号是0-l5）。

(2)最左侧数码管实时显示4个拨码开关的状态。数码管只支持显示0-9，如果拨码开关状态是10-15,则数码管的显示状态不更改（显示上一次的显示值）。

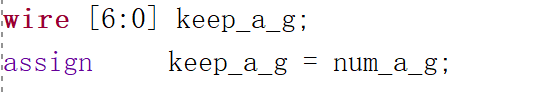
(3)最右侧的4个单色LED灯会显示上一次的拨码开关的状态,支持显示0-l5（拨码开关拨上对应LED灯亮）

通过这些实验功能，从代码中可看出需要修改的部分如下：

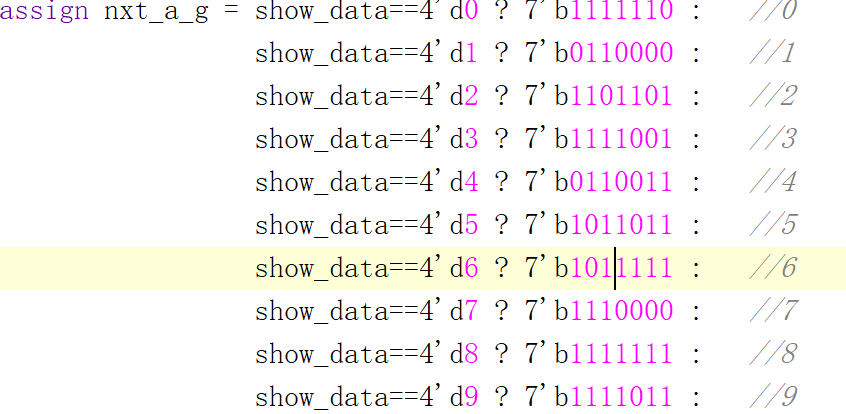
1.进行赋值操作，由于拨码开关在实验箱上的情况和想要实现的功能恰好相反，则需要将switch取反赋给show\_data



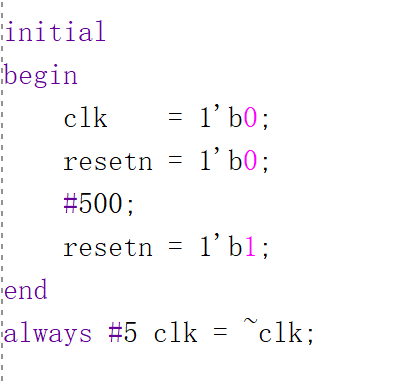
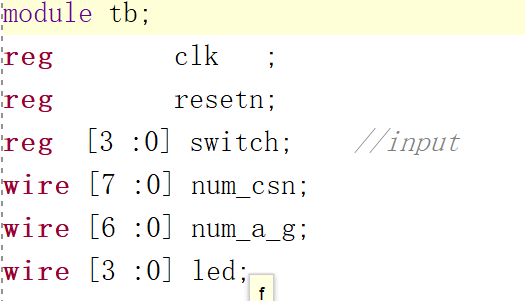
2.由于当show\_data>=10时，不通过数码管显示数据而显示之前的数据，因此将num\_a\_g直接赋值给keep\_a\_g，则nxt\_a\_g的值在show\_data>=10的情况下为之前的num\_a\_g



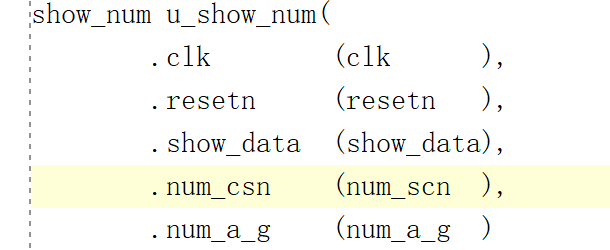
3. nxt\_a\_g 是根据输入信号show\_data的值选择对应的数码管编码（a-g 段）,而输入信号show\_data缺少值为0的情况，需要对其进行补足



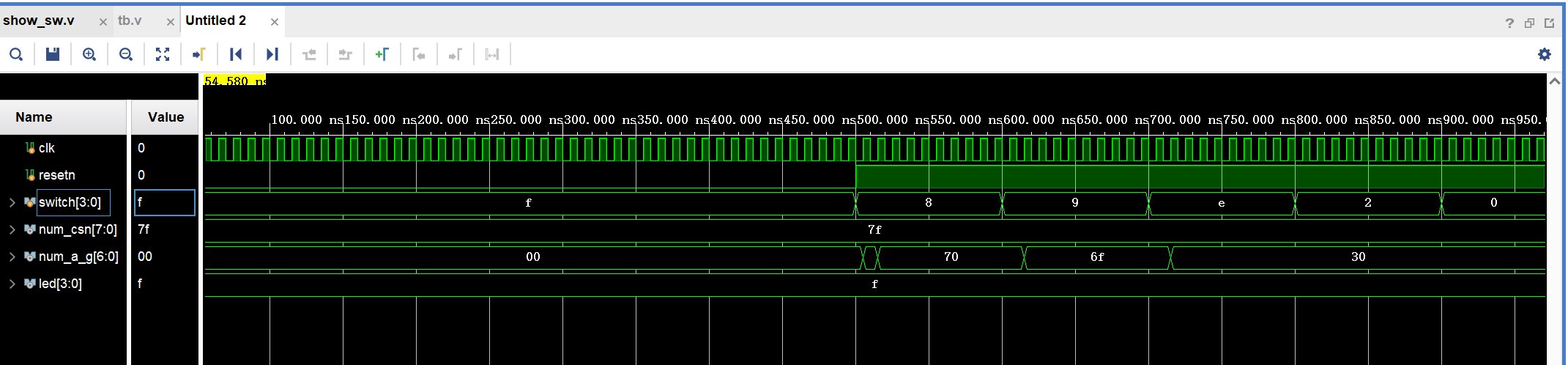
4.观察tb文件，发现会间隔100ns才有初始值，将这部分和后面相关的部分删除，另外在仿真波形中仅有输入情况没有输出情况，增加输出显示



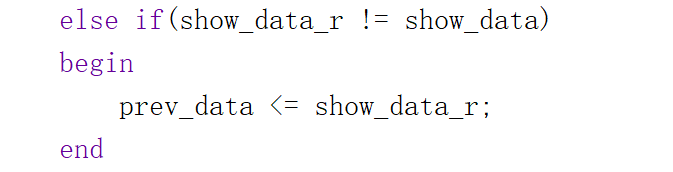
5.实例化模块时有输入错误，将num\_scn改为num\_csn



在进行上述修改后已经基本满足了该项目要求实现的功能，然而此时发现实验的仿真波形图中led的值始终不变：

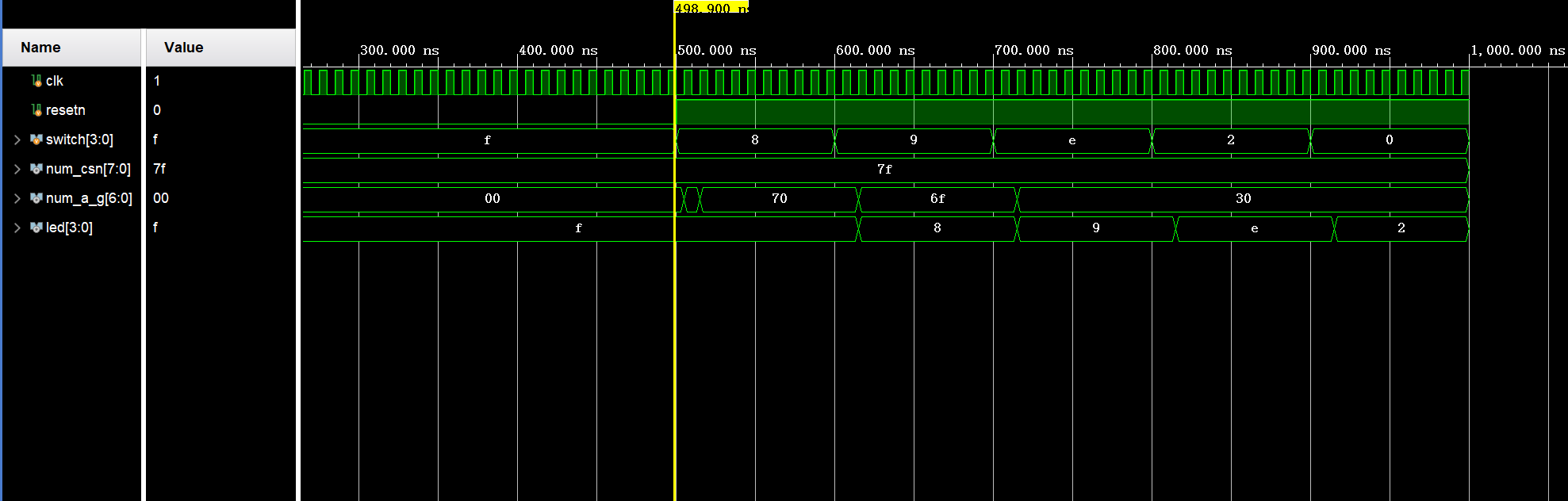


借此可以发现，led=f由于led在代码中是由prev\_data赋值的，则prev\_data始终为初始赋值0，而prev\_data的正确赋值情况应该与show\_data\_r相关(即show\_data的缓存值)，则进行下面的修改：

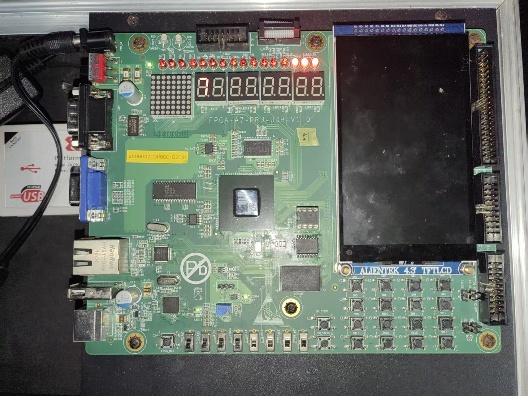
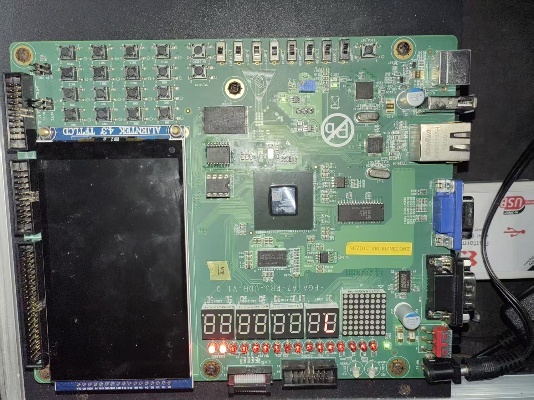


即将阻塞赋值改为非阻塞赋值，使得show\_data\_r赋值给prev\_data也依赖时钟周期

修改最后的仿真波形如下：



实验箱上进行验证结果如下：



其中第一张图片中拨码开关输入0111，数码管中显示7，前一时刻的输入为0011，灯泡对应发光；第二张图片中，拨码开关输入1111，此时由于输入>=10，则数码管中显示上一时刻的输入，同时灯泡符合上一时刻输入0111的情况，实现了要求的功能

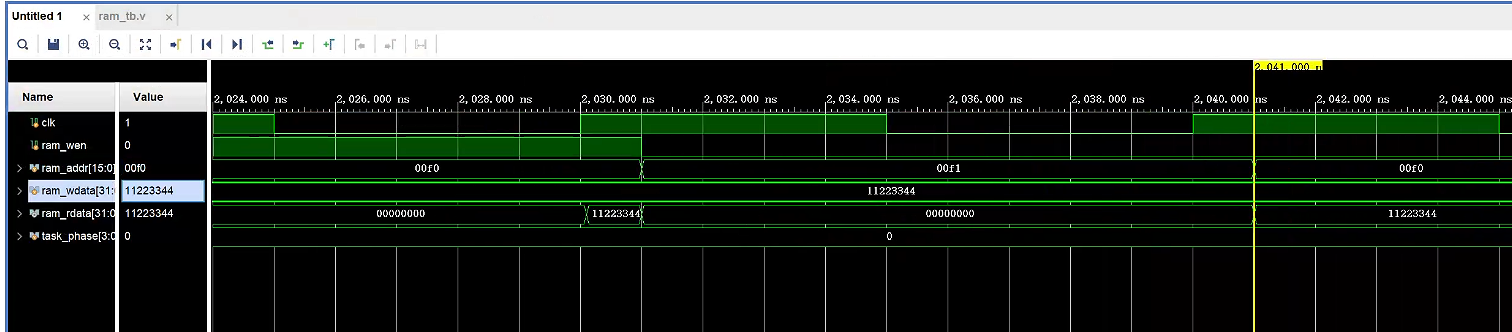
**4.总结感想**

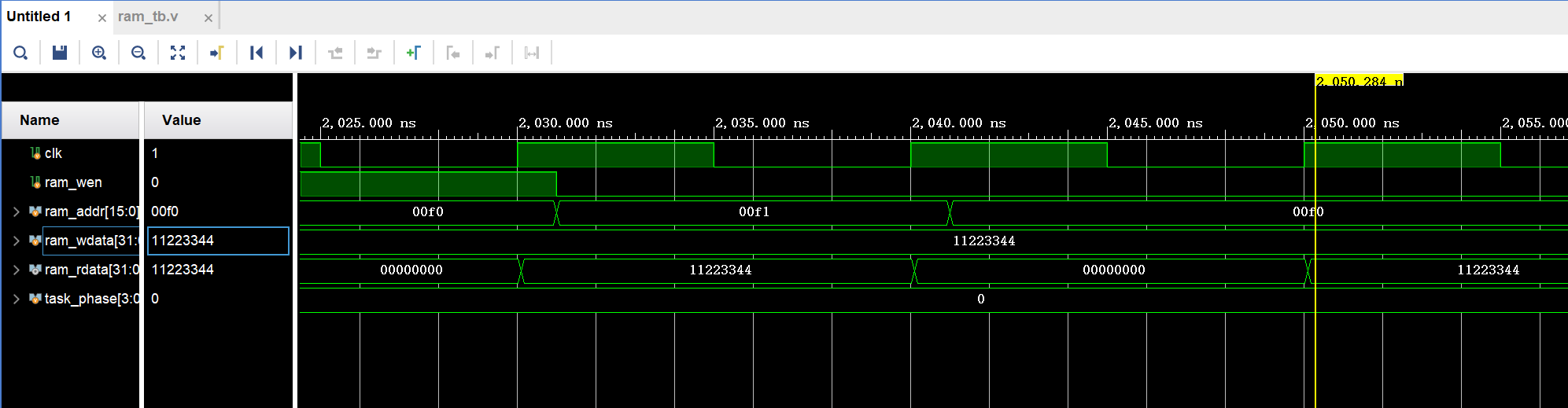
(1) 为什么寄存器堆要设计成“两读一写”？

由于RISC指令通常采取三操作数的各式，比如说add rd,rs1,rs2这个指令，其中需要同时读取两个源操作数rs1和rs2，如果只有一个读取端口，则无法在单周期内同时读取这两个操作数，会导致性能下降；另外，由于CPU通常采用乱序执行，多个指令可能会同时需要读取不同的寄存器，有两个读取端口可以实现并行读取，如果只有 1 个读取端口，乱序执行可能会由于过多的数据依赖而频繁停顿，这样就会造成运行速度的减慢。

(2) 同步ram和异步ram各自的特点和区别

从仿真实验中可以看出，同步ram依赖于时钟周期clk，当读数据时需要在时钟上升沿才可以将数据读出，但异步ram并不依赖于clk，将会在一定延迟时间后自动将数据读出。例如在下面这个时间点上可以看出，异步ram并没有等待clk的上升沿而是在一段时间后自动变化。而在这个时间点前后的同步ram可以看出等待时钟上升沿才发生变化





(3) vivado调试的经验步骤。

主要是通过对代码进行仿真，通过观察仿真中的某些不合功能设计的部分，找到对应关系代码进行修改调试；另外，还可以根据所需功能本身检视代码，确定是否有不合理或者遗漏的部分进行调试修改。