实验二报告

学号：2018K8009929021

姓名：袁欣怡

箱子号：10号

1. 实验任务（10%）

子任务一：完成对寄存器堆的仿真，观察数据在寄存器堆中的读写过程。

子任务二：通过观察同步、异步RAM仿真的结果，对比读写数据行为的异同。

子任务三：已有一段数字电路设计源码，但是其中有五个错误，分析并找出这些错误。

1. 实验设计（0%）
2. 实验过程（90%）
3. 实验流水账

2020.9.16 23:00-24:00 完成子任务一，观察寄存器堆中数据流动的过程。

2020.9.17 13:00-16:30 完成子任务二，观察同步、异步RAM中数据读写行为的异同。

2020.9.17 18:30-23:00 完成子任务三，定位并修改所给源码中的五个错误。

2020.9.18 10:00-14:00 撰写、修改实验报告

1. 子任务一

观察数据在寄存器堆中的读写过程。

先建立一个工程，并将regfile.v和rf\_tb.v添加到工程中。

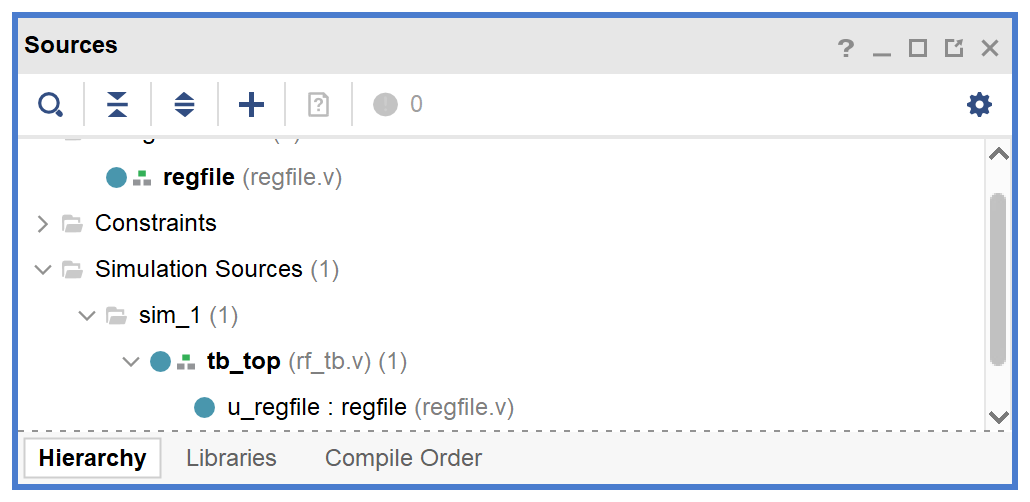


图1 添加设计文件和仿真文件

进行仿真，观察产生的波形图。

通过读代码可以发现，向寄存器堆中赋值的行为需要在时钟上升沿才能触发，但是从寄存器堆中读取数据的行为可以发生在任何时刻，寄存器堆中的数据一旦发生变化，对应的rdata就会发生变化。这一点在后面的分析中有多次体现。

Phase0阶段：

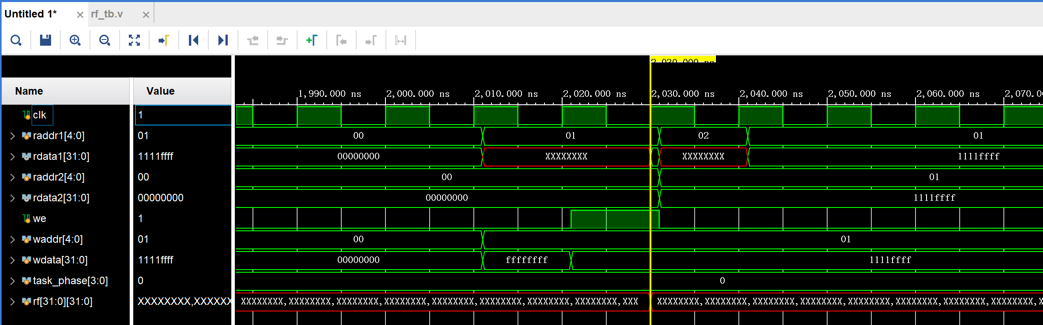


图2 1990ns～2070ns波形图

2030ns之前，读rf[0]的值始终为0，rf[1]的值为不定值X。尽管此时已经有waddr和wdata信号，但是因为we信号始终为低，所以没有向寄存器堆中赋值。

2021ns时，we信号拉高，但是由于赋值过程需要在时钟上升沿触发，因此2030ns时钟上升沿时，rf[1]的值变化为1111ffff，此时raddr1=1，rdata1的值立即发生变化。

Phase1阶段：

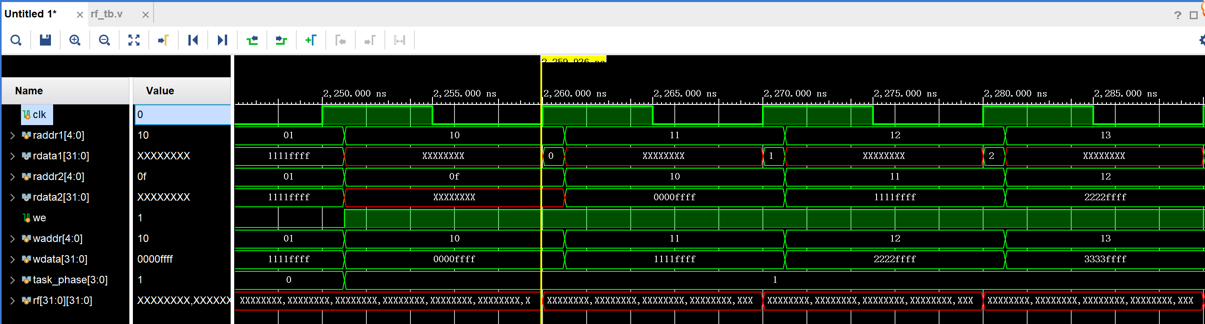


图3 2250ns～2290ns波形图

2251ns时，we信号拉高，waddr和wdata发生变化，但是等到下一个时钟上升沿，即2260ns时，才给rf[10]赋值，同时radta1读出的数据发生变化。后面几个时钟周期中的操作也是类似的。

Phase2阶段：

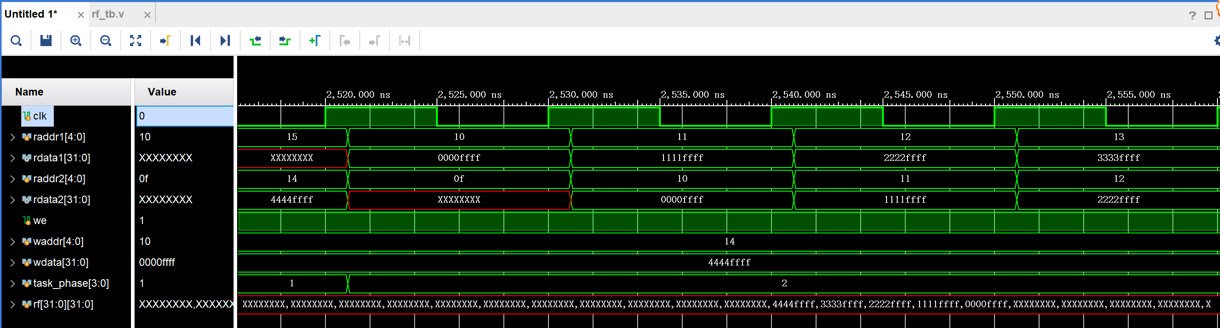


图4 2520ns～2560ns波形图

Phase2阶段中主要是读取数据的操作，其中rf[10], rf[11], rf[12], rf[13]在2250ns至2290ns中赋值。

1. 子任务二
2. 仿真行为对比分析

Phase0阶段：

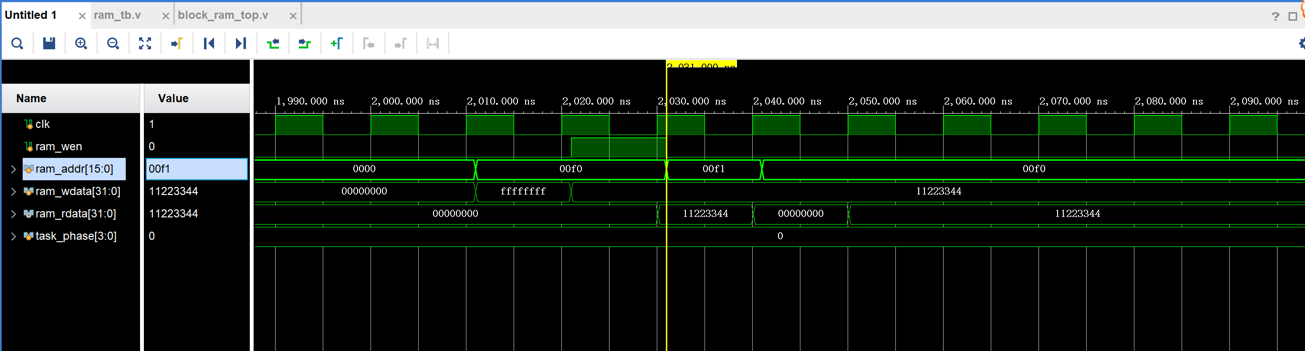


图5 block\_ram\_top 同步RAM 1990ns~2070ns波形图

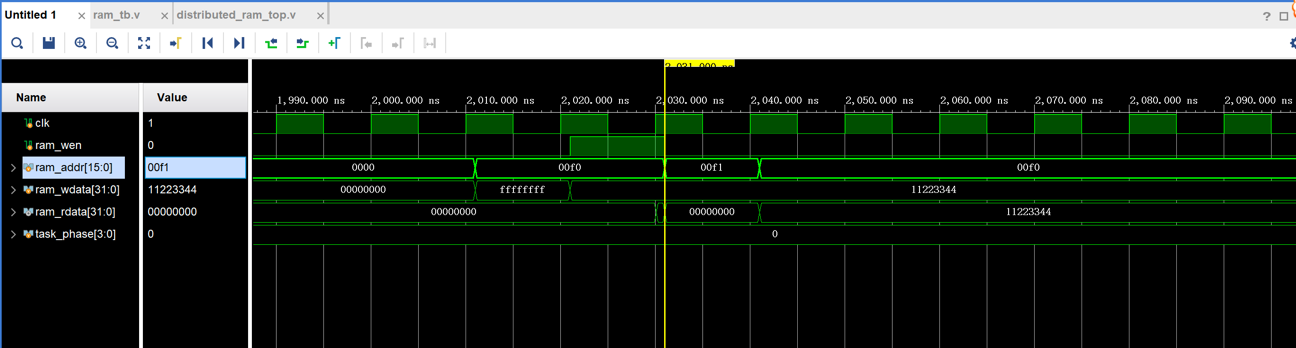


图6 distributed\_ram\_top 异步RAM 1990ns~2070ns波形图

从波形图中可以看出，同步RAM和异步RAM都在2030ns时进行了读数据的操作。此后，同步RAM在2040ns和2050ns时读了两次数据，而异步RAM在2031ns和2041ns时读取数据。

从读出的数据来看，两者在2030ns时读出的数据均为11223344。此后，同步RAM在2040ns读出0，在2050ns时读出11223344；异步RAM在2031ns时读出0，在2041ns时读出11223344。

从波形图中不难看出，同步RAM的ip核中，数据的读写都是同步进行的，也就是说，只有等时钟上升沿到来的时刻才会进行读写。在异步RAM的ip核中，数据的写入时同步进行的，但是数据的读出是异步进行的。体现在仿真波形上，数据11223344都是在2030ns时写入00f0位置的，此时00f0中存储的数据为11223344，00f1中存储的数据为0，并且ram\_rdata也相应发生变化。读取数据的时候，同步RAM中ram\_rdata只在时钟上升沿发生变化，而异步RAM中ram\_rdata随ram\_addr改变而同时改变。

Phase1阶段：

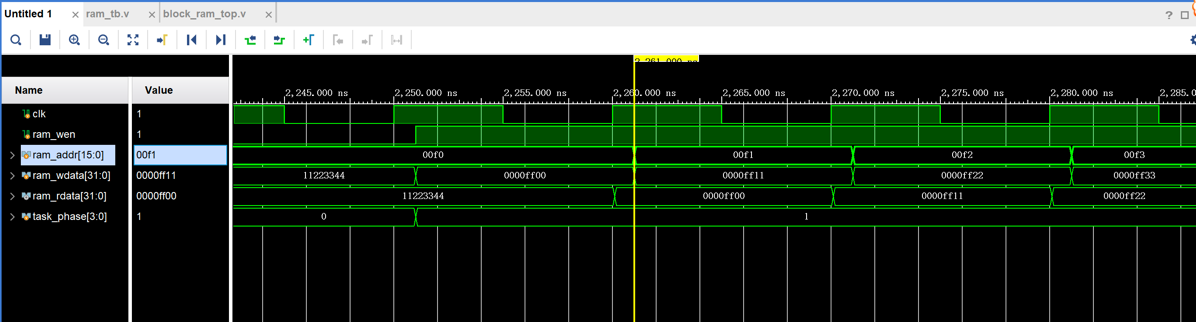


图7 block\_ram\_top 同步RAM 2245ns～2280ns波形图

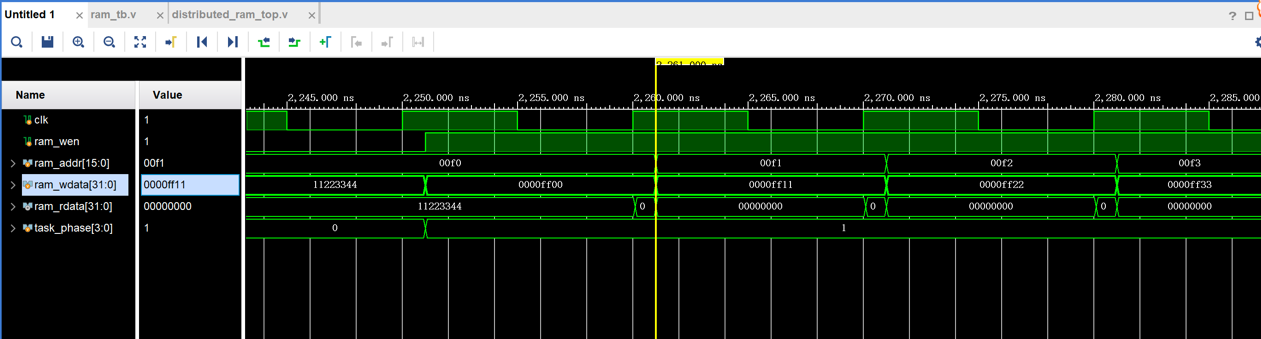


图8 distributed\_ram\_top 异步RAM 2245ns～2280ns波形图

从波形可以看出，数据0000ff00均在2260ns时写入00f0，数据0000ff11均在2270ns时写入00f1，数据0000ff22均在2280ns时写入00f2。在同步RAM中，只在时钟上升沿时读取数据，因此ram\_rdata只在2260ns、2270ns和2280ns发生改变；在异步RAM中，可以随时读取数据，因此2260ns时，ram\_rdata读取00f0中的数据0000ff00，2261ns时ram\_addr变为00f1，所以ram\_rdata也同时变为0，直到2270ns时00f1被赋值，以此类推。

Phase2阶段：

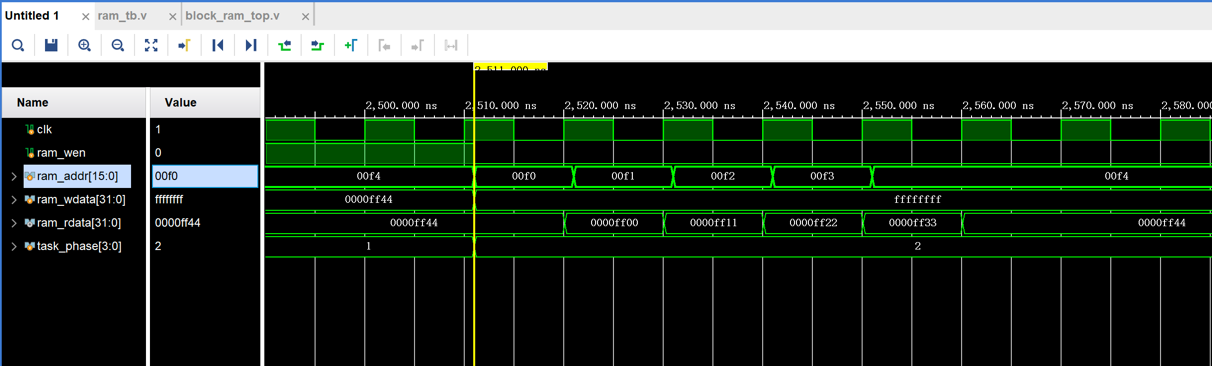


图9 block\_ram\_top 同步RAM 2500ns～2580ns波形图

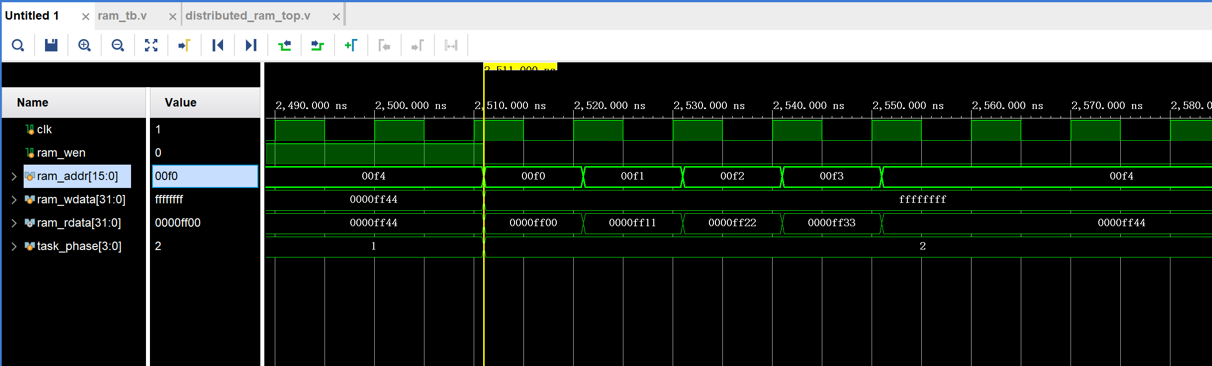


图10 distributed\_ram\_top 异步RAM 2500ns～2580ns波形图

在2050ns时，两个RAM均给00f4赋值0000ff44。此后we信号持续为低，不再写入数据。读取数据时，同步RAM中ram\_rdata只在时钟上升沿到来时发生变化，而异步RAM中ram\_rdata和ram\_addr同时变化。

1. 时序、资源占比对比分析

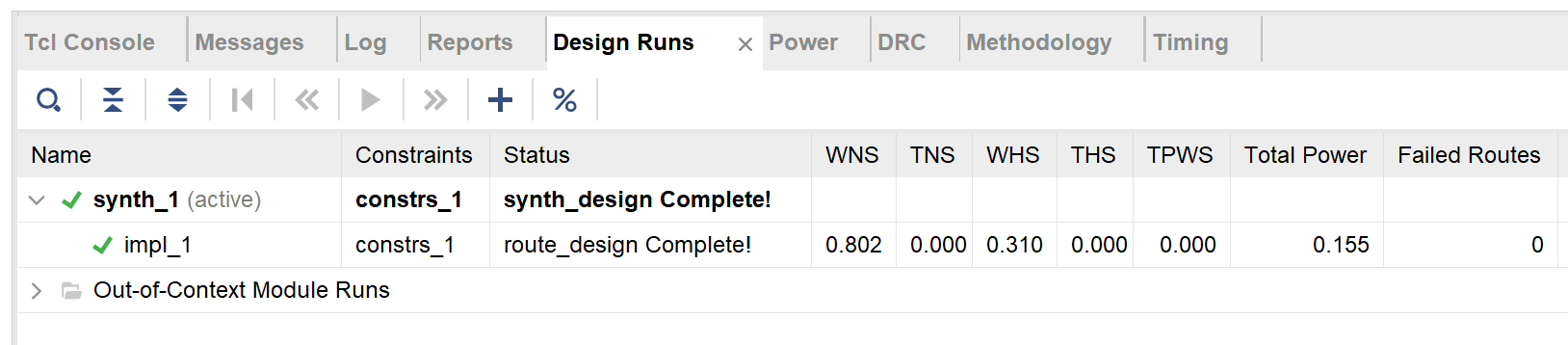


图11 同步RAM时序结果

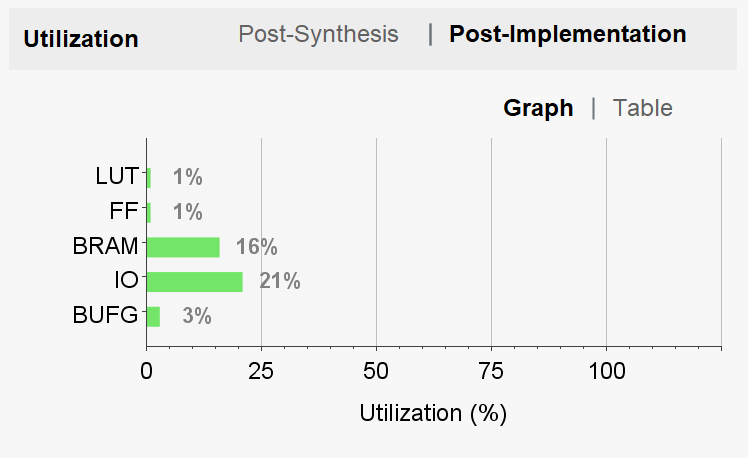


图12 同步RAM资源利用率

社交网络的手机截图

描述已自动生成

图13 异步RAM时序结果

手机屏幕截图

描述已自动生成

图14 异步RAM资源利用率

同步RAM的WNS和TNS值均为非负数，表示时序满足极好；异步RAM中WNS和TNS为负数，但WNS违约不超过300ps，表示时序满足较好。

资源利用率异步RAM明显大于同步RAM，且进行综合时也明显感觉异步RAM所用时间特别长，可见异步RAM占用的资源多得多。

1. 总结

综上所述，同步RAM和异步RAM读写方式不同，可以在不同的地方发挥作用，但是异步RAM所需要的资源比同步RAM多得多，因此，当可以利用的资源受限时，应优先考虑使用同步RAM。

1. 子任务三
2. 错误1:信号为“Z”
3. 错误现象

信号num\_csn持续为Z

1. 分析定位过程

猜测此错误可能因为此信号在模块调用时没有与接口相连，导致没有成功赋值。在查看波形图、向波形图中添加信号时，发现一个名为num\_scn的未知信号，因而怀疑因为拼写错误导致出错。

1. 错误原因

赋值时拼写错误。

1. 修正效果

将num\_scn修改为num\_csn。

1. 归纳总结（可选）

在向波形图中添加信号时，可以先检查一下最下面的几个信号，如果出现不是自己定义的信号，则说明代码中出现了拼写错误，需要首先改正。

1. 错误2:信号为“X”
2. 错误现象

信号show\_data和show\_data\_r持续为X。

1. 分析定位过程

因为这两个信号都是reg类型的信号，所以怀疑它们未被赋值。

1. 错误原因

误将被show\_data赋值的语句注释掉了。

1. 修正效果

将show\_data<=~switch语句恢复到原程序中。

1. 归纳总结（可选）

有时调试的时候可能会将一些语句注释掉，调试完成之后一定要记得检查有没有恢复所有被注释掉的语句。

1. 错误3:波形停止
2. 错误现象

波形在710ns时停止。

1. 分析定位过程

波形停止很有可能是因为循环赋值导致的，因此检查波形图中有没有产生环路。

1. 错误原因

nxt\_a\_g和keep\_a\_g两个信号形成环路。

1. 修正效果

通过分析需要实现的功能发现keep\_a\_g信号不必要存在，因此删除该信号。修改环路后各信号均有正常波形，且可以跑完整个仿真。

1. 归纳总结（可选）

在设计电路时，需要先想好各个信号的功能和它们之间的赋值关系再开始写，否则很容易产生环。

1. 错误4:越沿采样
2. 错误现象

信号prev\_data恒为0，没有发生过变化。

1. 分析定位过程

从代码中可以得知，当show\_data和show\_data\_r不相等的时候，将show\_data\_r的值赋给prev\_data。但是这一赋值行为似乎始终没有发生，推断可能是判定条件出现问题，因而发现show\_data\_r赋值的时候出现越沿采样。

1. 错误原因

给信号show\_data\_r赋值时越沿采样。

1. 修正效果

修正后prev\_data的值正常。

1. 归纳总结（可选）

在always块内赋值时，始终采取非阻塞赋值<=，防止出现越沿采样的情况。

1. 错误5:波形怪异
2. 错误现象

当show\_data=6的时候，num\_a\_g的值没有更新。

1. 分析定位过程

检查num\_a\_g的赋值语句，因而找到nxt\_a\_g信号的赋值语句，于是发现没有考虑show\_data=6的情况。

1. 错误原因

信号nxt\_a\_g的赋值语句中没有考虑到show\_data=6的情况。

1. 修正效果

修正后波形正常。

1. 归纳总结（可选）

在分类讨论的时候要做到不重不漏。

1. 修改后结果

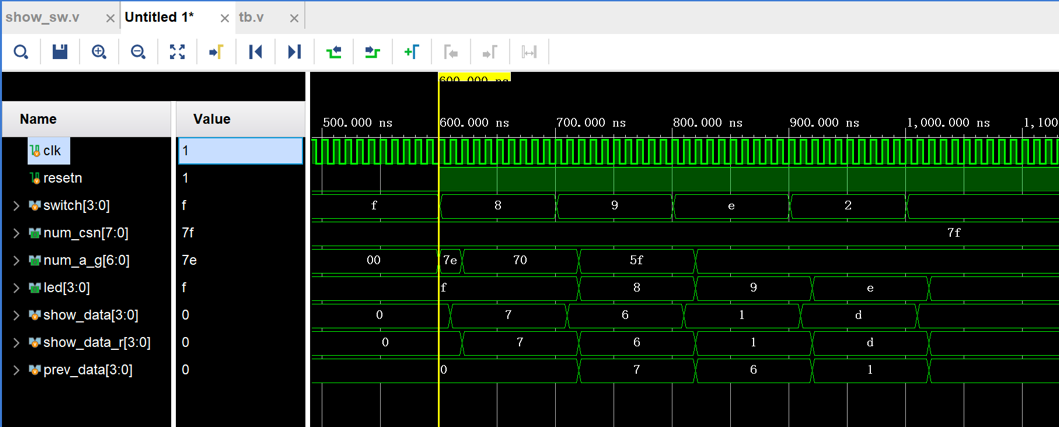


图15 debug后仿真波形

1. 实验总结（可选）

通过本次实验，了解了reg\_file的运行方式，同步异步RAM的区别和读写方式，还锻炼了在代码中寻找错误的能力。