实验二报告

学号：2018K8009929021
姓名：袁欣怡
箱子号：10 号

一、 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。

二、 实验设计（0%）

三、 实验过程（90%）

（一） 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

（二） 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：

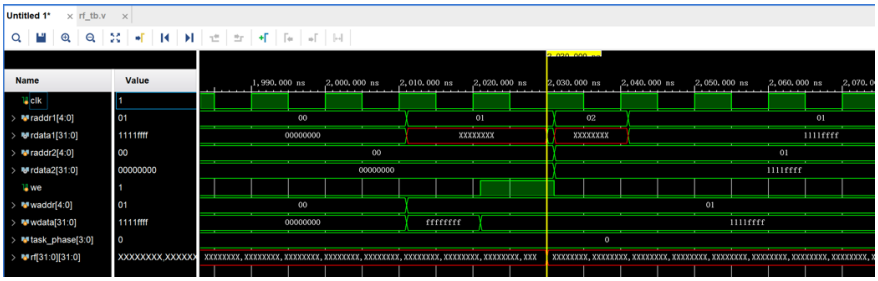

图 2 1990ns ~ 2070ns 波形图

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：

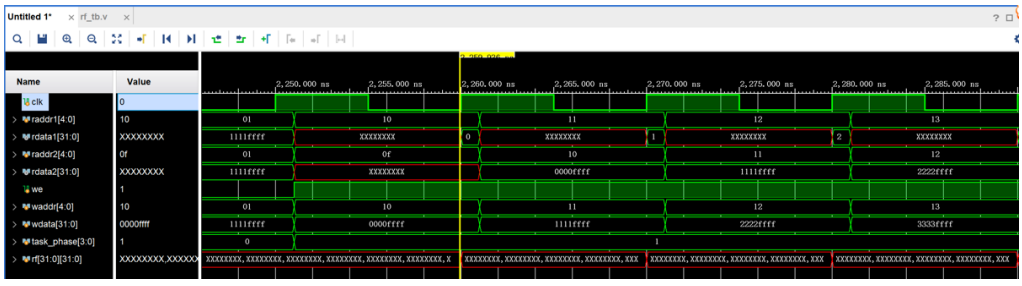

图 3 2250ns ~ 2290ns 波形图

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：

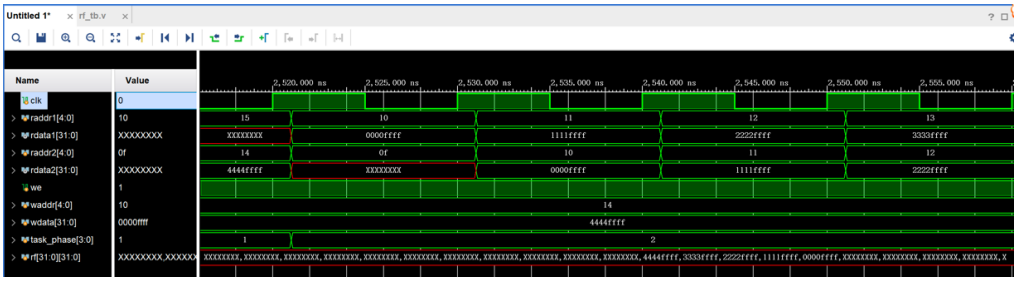

图 4 2520ns ~ 2560ns 波形图

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。

（三） 子任务二

1、 仿真行为对比分析

Phase0 阶段：

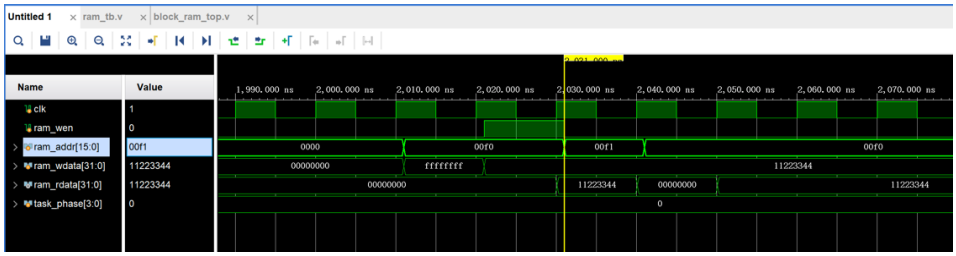

图 5 block_ram_top 同步 RAM 1990ns~2070ns 波形图

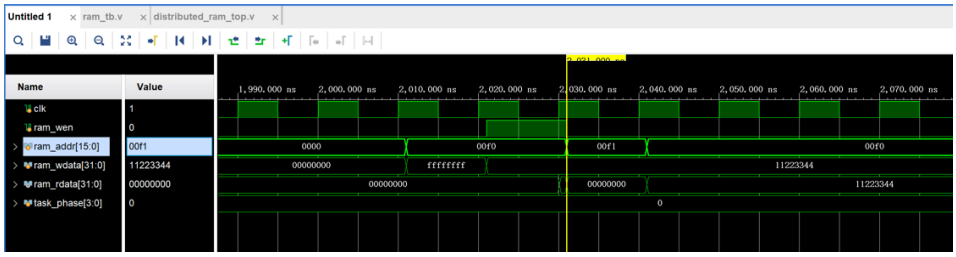

图 6 distributed_ram_top 异步 RAM 1990ns~2070ns 波形图

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：

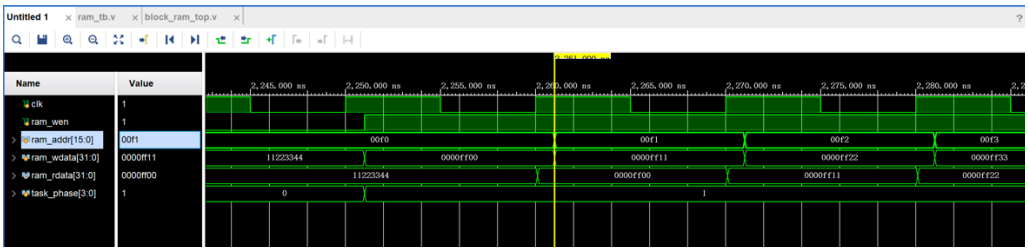

图 7 block_ram_top 同步 RAM 2245ns ~ 2280ns 波形图

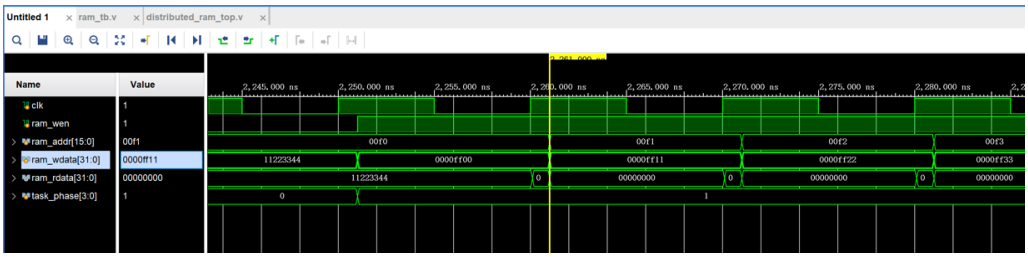

图 8 distributed_ram_top 异步 RAM 2245ns ~ 2280ns 波形图

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：

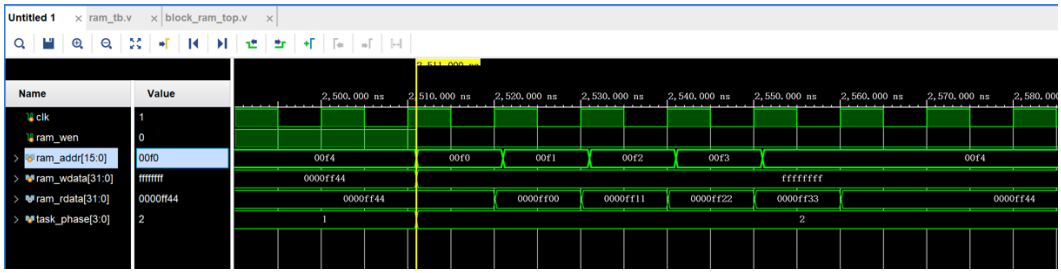

图 9 block_ram_top 同步 RAM 2500ns ~ 2580ns 波形图

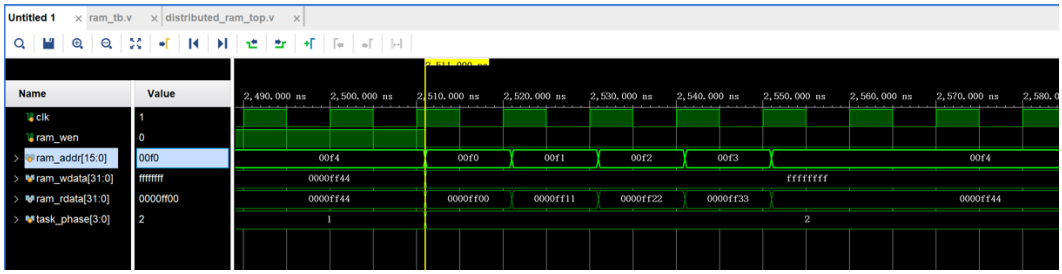

图 10 distributed_ram_top 异步 RAM 2500ns ~ 2580ns 波形图

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。

2、 时序、资源占比对比分析

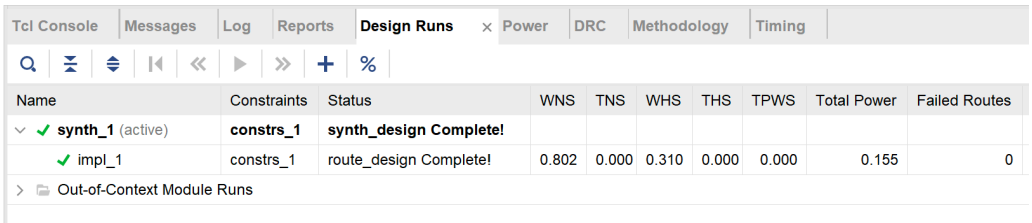

图 11 同步 RAM 时序结果

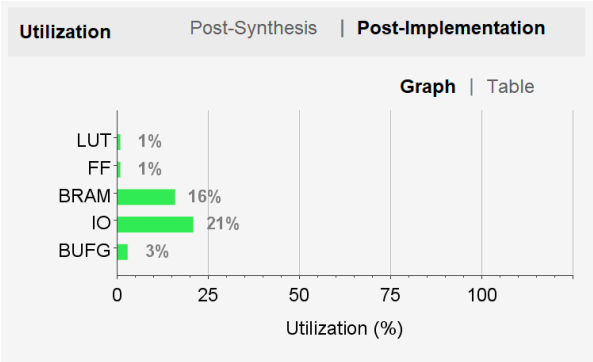

图 12 同步 RAM 资源利用率

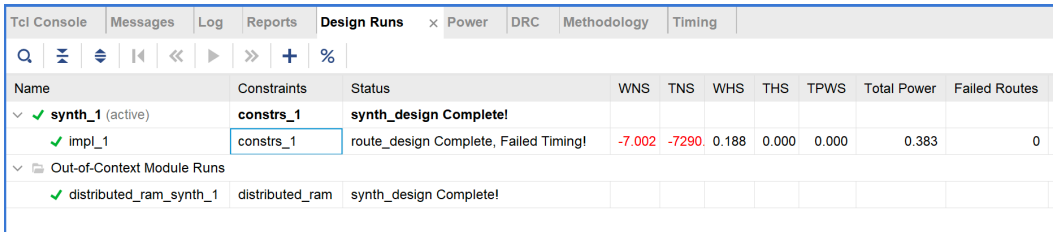

图 13 异步 RAM 时序结果

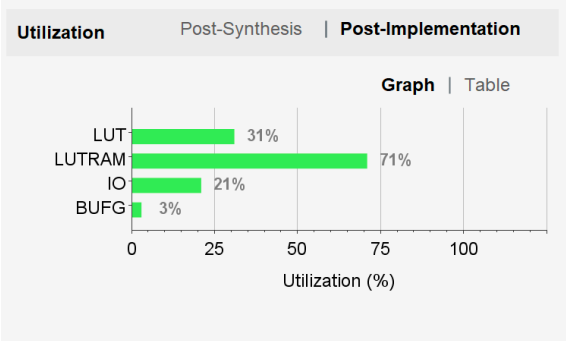

图 14 异步 RAM 资源利用率

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。

3、 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。

（四） 子任务三

1、 错误 1:信号为“Z”

（1） 错误现象

信号 num_csn 持续为 Z

（2） 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。

（3） 错误原因

赋值时拼写错误。

（4） 修正效果

将 num_scn 修改为 num_csn。

（5） 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。

2、 错误 2:信号为“X”

（1） 错误现象

信号 show_data 和 show_data_r 持续为 X。

（2） 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。

（3） 错误原因

误将被 show_data 赋值的语句注释掉了。

（4） 修正效果

将 show_data<=~switch 语句恢复到原程序中。

（5） 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。

3、 错误 3:波形停止

（1） 错误现象

波形在 710ns 时停止。

（2） 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。

（3） 错误原因

nxt_a_g 和 keep_a_g 两个信号形成环路。

（4） 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。

（5） 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。

4、 错误 4:越沿采样

（1） 错误现象

信号 prev_data 恒为 0，没有发生过变化。

（2） 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。

（3） 错误原因

给信号 show_data_r 赋值时越沿采样。

（4） 修正效果

修正后 prev_data 的值正常。

（5） 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。

5、 错误 5：波形怪异

（1） 错误现象

当 show_data=6 的时候，num_a_g 的值没有更新。

（2） 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。

（3） 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。

（4） 修正效果

修正后波形正常。

（5） 归纳总结（可选）

在分类讨论的时候要做到不重不漏。

6、 修改后结果

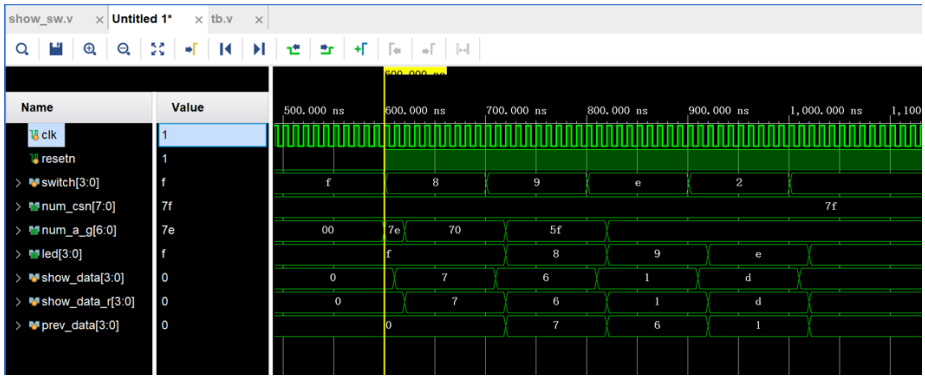

图 15 debug 后仿真波形

四、 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。