**中国科学技术大学计算机学院**

**《计算机组成原理实验》报告**



实验题目：寄存器堆与存储器及其应用

学生姓名：\_\_\_\_ 黄瑞轩\_ \_\_\_\_

学生学号：\_\_ PB20111686 \_\_\_

完成日期：\_\_ 2022.3.28 \_\_

计算机实验教学中心制

2020年09月

**实验题目**

寄存器堆与存储器及其应用

**实验目的**

* 掌握寄存器堆和存储器的功能、时序及其应用
* 熟练掌握数据通路和控制器的设计和描述方法

**实验环境**

* Nexys4-DDR
* Vivado 2019.1

**实验**1: **32×32位的寄存器堆**

* 器件设计
* 要求实现的功能

具有异步读端口和同步写端口，数据宽度为32位，数据深度为32（即5根地址线），寄存器堆的0号寄存器内容恒定为零，且寄存器堆的写操作优先于读操作。

* 逻辑设计

|  |  |
| --- | --- |
|  | ra0, rd0：异步读端口0，a表示地址，d表示内容，下同  ra1, rd1：异步读端口1  wa, wd, we：同步写端口，e表示使能  clk：时钟 |

* 核心代码

见附件register\_file.v。

**问：如何实现寄存器堆的**0**号寄存器内容恒定为零？**

答：用initial语句给0号寄存器内容赋值0，在写入时对wa做特判，仅当wa != 0时才能完成写入。

**问：如何实现寄存器堆的写操作优先于读操作？**

答：做逻辑判断即可：

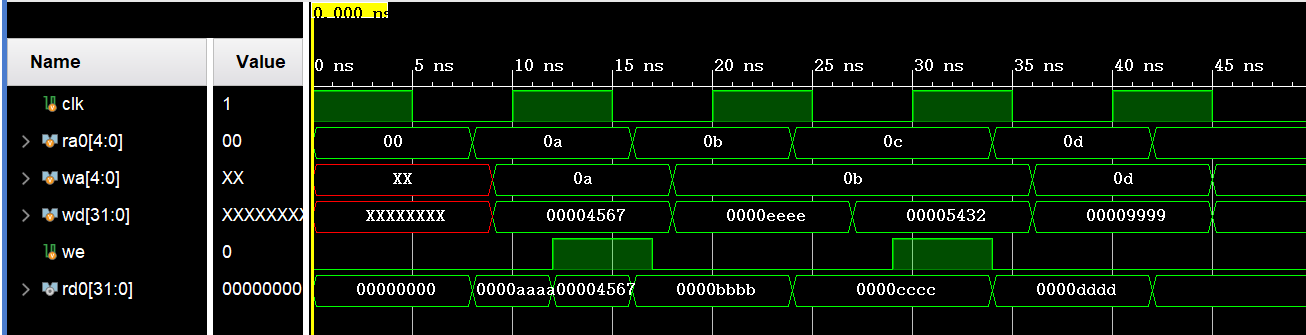
assign rd0 = (ra0 == wa) && we ? wd : rf[ra0];

assign rd1 = (ra1 == wa) && we ? wd : rf[ra1];

* 模拟仿真

testbench文件见附件register\_file\_tb.v。

注：为了仿真方便，在设计文件中初始化了0a处内容为0aaaa、0b处内容为0bbbb、…、0e处内容为0eeee。上交代码时这些初始化会被注释。



经验证与预期一致。

实验2: 256×16位的分布式和块式单端口RAM IP核的功能仿真和对比

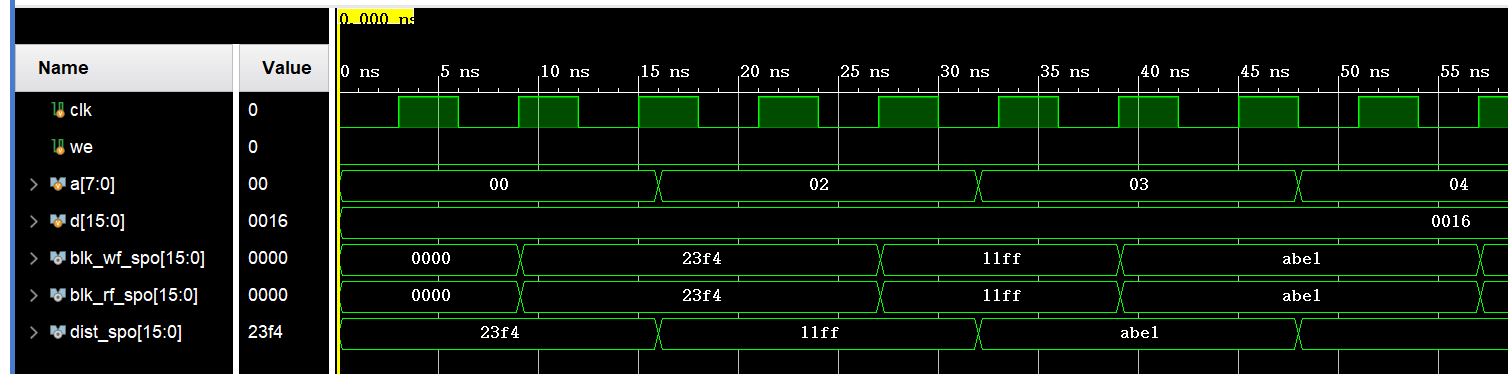
* 256×16位的分布式和块式单端口RAM IP核的创建

|  |  |
| --- | --- |
|  |  |
| 分布式单端口RAM创建参数选择  默认值：16’habcd | 块式单端口RAM创建参数选择  写优先的命名为：blk\_mem\_gen\_WF  读优先的命名为：blk\_mem\_gen\_RF |

COE文件见附件init\_dist.coe、init\_blk\_wf.coe和init\_blk\_rf.coe，为了方便对比，这里把三种存储器的COE文件设置成相同的。

* 256×16位的分布式和块式单端口RAM IP核的仿真
* 分布式和块式存储器的读操作仿真

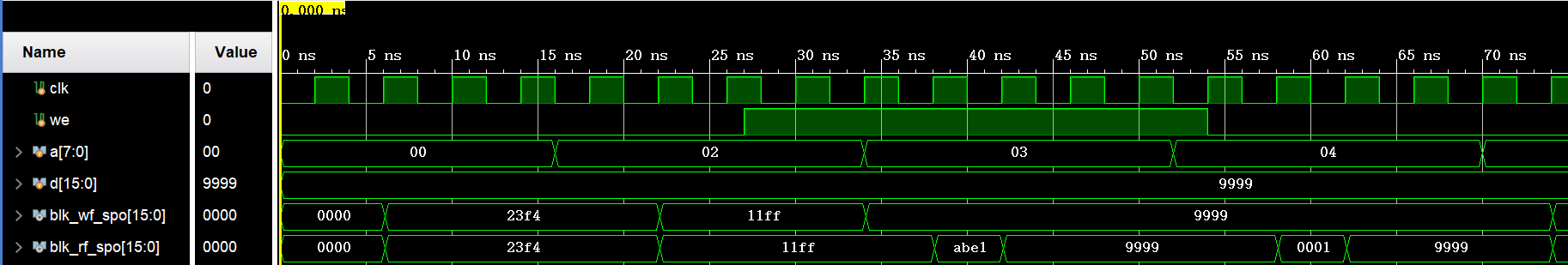
仿真文件见附件dist\_blk\_read\_tb.v。



为了方便进行对比，这里同时例化了一个分布式存储器、一个读优先块式存储器和一个写优先块式存储器，并对clk、we、a、d信号采用相同的输入，把三个存储器的输出端口接到不同的测试位置。

* 分布式存储器的读操作是异步的，见第16 ns，当a改变时，posedge clk未到来，但输出端口值已经改变。
* 两种块式存储器的读操作是同步的，并且第一个posedge clk对块式存储器来说可能是使能信号，在a改变后的第二个posedge clk，输出端口值才会改变。
* 块式存储器的读优先和写优先仿真

仿真文件见blk\_first\_tb.v。为了方便对比，新写入的值都设置为9999。



* 对于写优先的块式存储器，在一个posedge clk到来时，先将din上的数据写入到addr对应的地址中，然后再将addr对应地址中的数据读出。
* 对于读优先的块式存储器，在一个posedge clk到来时，先将addr对应地址中的原始数据读出，然后再将din上的数据写入到addr对应的地址中。

实验3: 排序电路

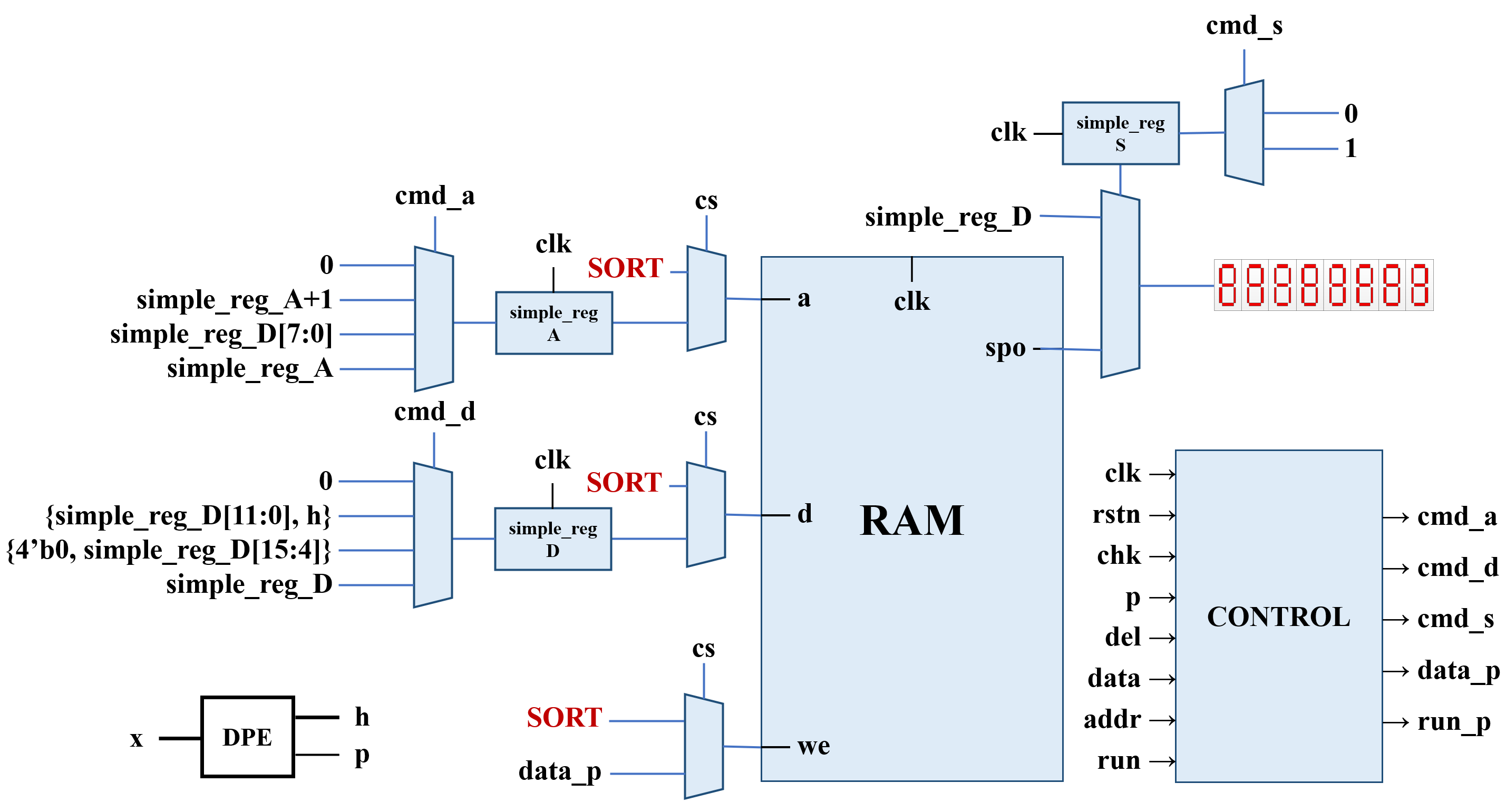
* 排序电路设计
* 需要实现的功能

排序电路采用分布式双端口存储器保存数据，例化时可以初始化数据；查看数据功能：数码管显示存储器的地址和数据；设置地址和修改数据功能，支持回退；可以统计排序所需的时钟周期数；忙碌显示。

* 器件封装图及端口设计

|  |  |
| --- | --- |
|  | x：开关，用于输入数据  del：退格，清除上一个数据  addr：设置地址  data：修改数据  chk：查看下一单元  run：开始排序  rstn：复位，查看第0单元  clk：时钟  an：七段数码管时分复用接口  ca-cg：七段数码管接口  busy：忙碌显示  cnt：排序消耗的时钟周期数 |

* 输入逻辑数据通路和控制器设计



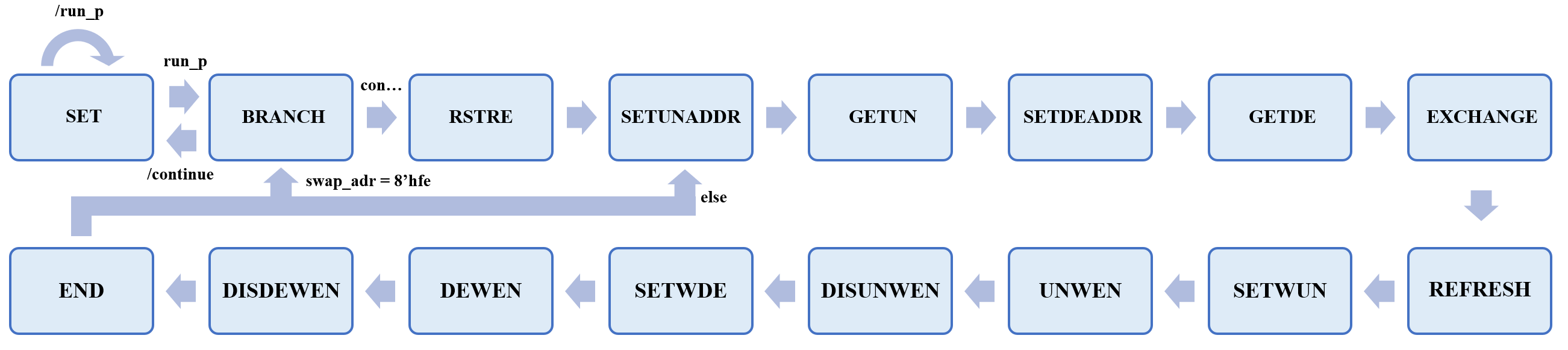
* 核心代码及思路简述

代码见附件sort.v、dpe.v。

DPE模块思路：将输入的x每一位取上升边沿，以此作为h；将各上升边沿获取器出口的归约或结果作为p。

SORT输入逻辑思路：根据run、del等控制信号的取值，改变cmd\_a、cmd\_d、cmd\_s等选择信号的值，以此来操作RAM的读写和七段数码管的显示。

* 排序功能状态图

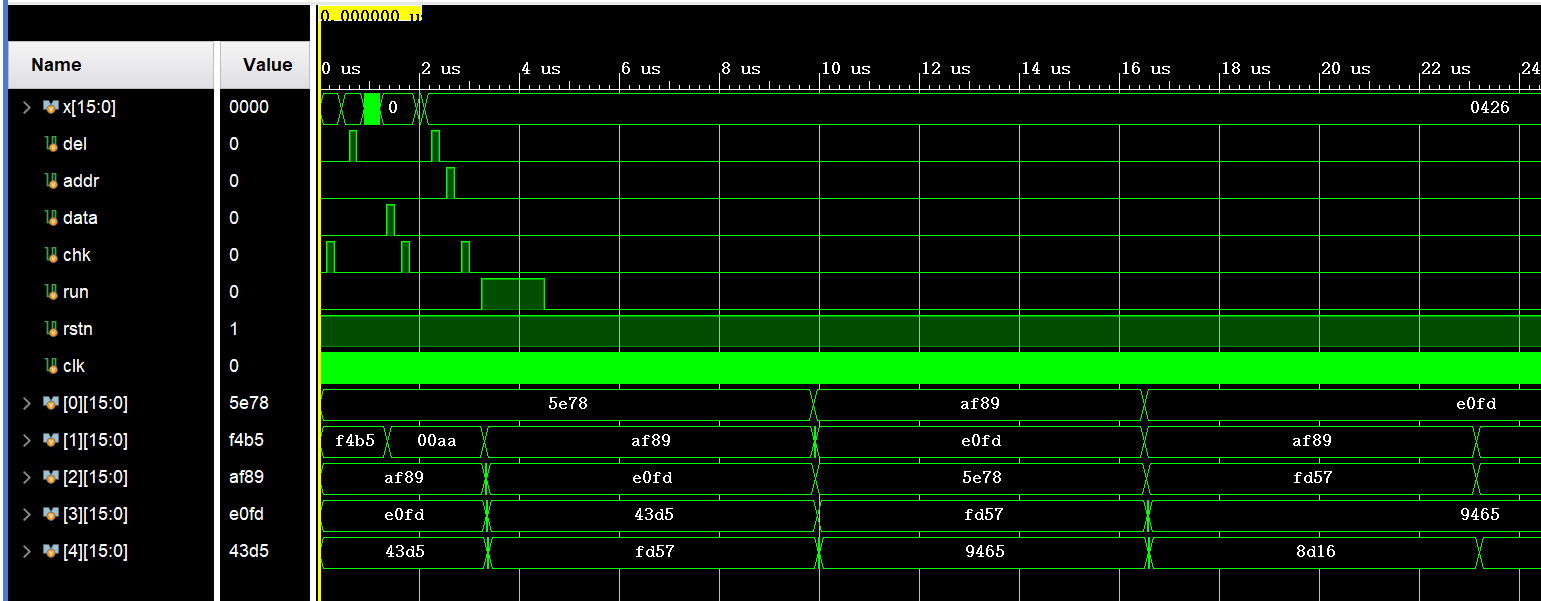


注：某些状态在分布式RAM情况下是冗余的。

本器件排序采用带continue标记优化的冒泡排序算法，按降序排。各状态解释如下：

|  |  |  |
| --- | --- | --- |
| 状态名 | 编码 | 操作 |
| SET | 0000 | 非排序状态，支持数据的写入和展示 |
| BRANCH | 0001 | 根据continue信号判断排序是否结束 |
| RSTRE | 0010 | 设置continue信号为0 |
| SETUNADDR | 0011 | [分布式RAM冗余] |
| GETUN | 0100 | 将前位数据读出保存至寄存器reg\_data |
| SETDEADDR | 0101 | swap\_addr后移 |
| GETDE | 0110 | 将后位数据读出保存至寄存器reg\_data2 |
| EXCHANGE | 0111 | 根据大小关系交换内容，并设置continue信号 |
| REFRESH | 1000 | [分布式RAM冗余] |
| SETWUN | 1001 | swap\_addr前移，并将reg\_d\_设置为reg\_data内容 |
| UNWEN | 1010 | 写使能信号有效 |
| DISUNWEN | 1011 | 写使能信号无效 |
| SETWDE | 1100 | swap\_addr后移，并将reg\_d\_设置为reg\_data2内容 |
| DEWEN | 1101 | 写使能信号有效 |
| DISDEWEN | 1110 | 写使能信号无效 |
| END | 1111 | [分布式RAM冗余] |

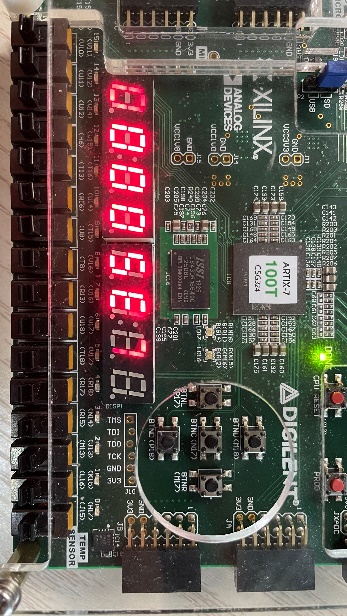
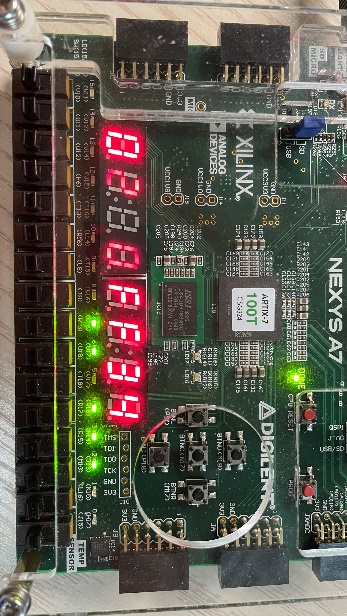
* 排序电路仿真

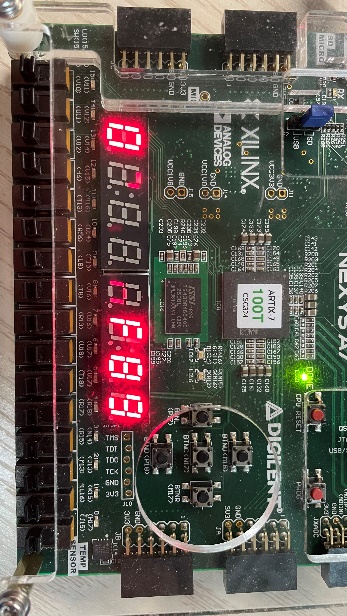
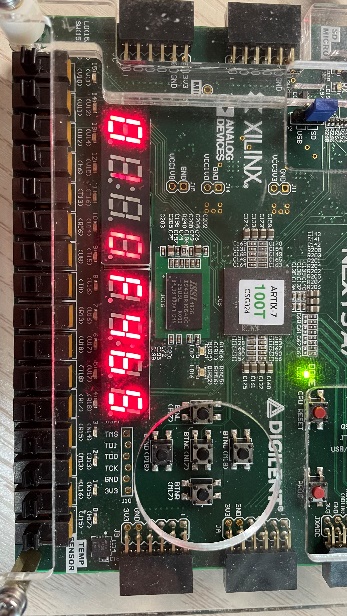


* run信号到来前是进行数据设置，经检验与预期一致。
* run信号到来后开始排序，RAM内置的数据是用Python自动生成的随机数，排序结果经检查与预期一致。
* 排序电路下载测试

前文已述，RAM内置的数据是用Python自动生成的随机数，此COE文件见附件coe\_sort.coe。由于手机拍照无视觉残留，无法正常显示数码管，这里附图仅供参考，以线下检查为主。

约束文件见附件const\_sort.xdc。

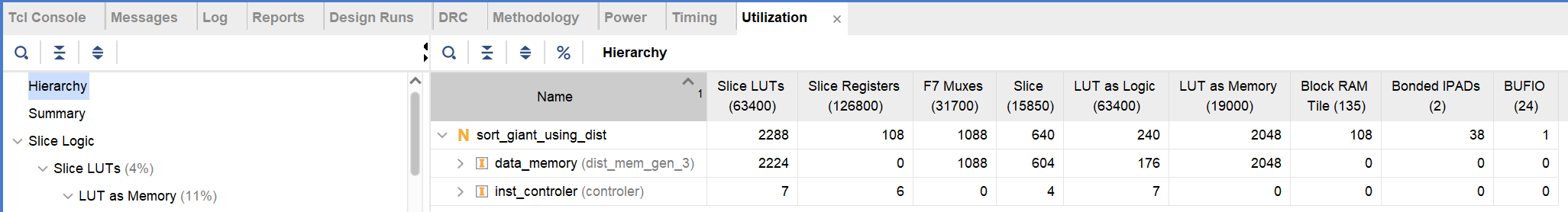




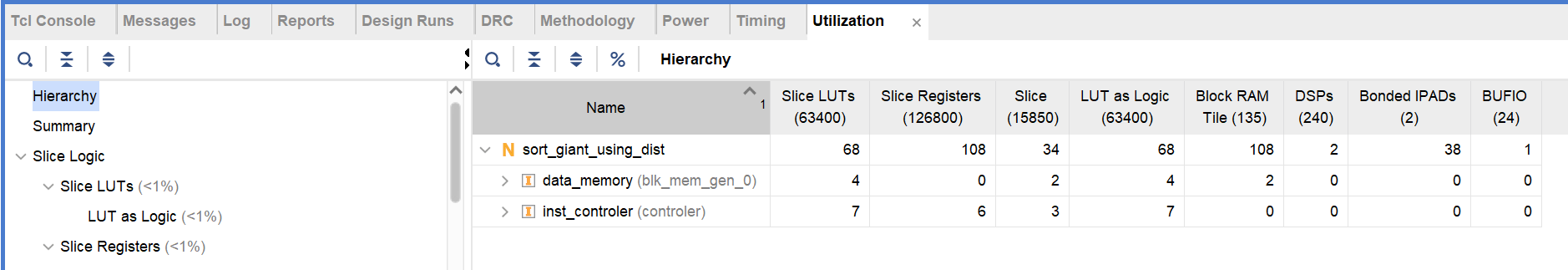
实验选项: 扩大数据量并对比不同RAM下排序电路性能

此实验选项需要对原来的排序电路稍作更改，改为4096×16位后需要12根地址线，这4096个初始数据均由Python随机生成，见附件coe\_giant\_sort.coe。

* 电路资源对比

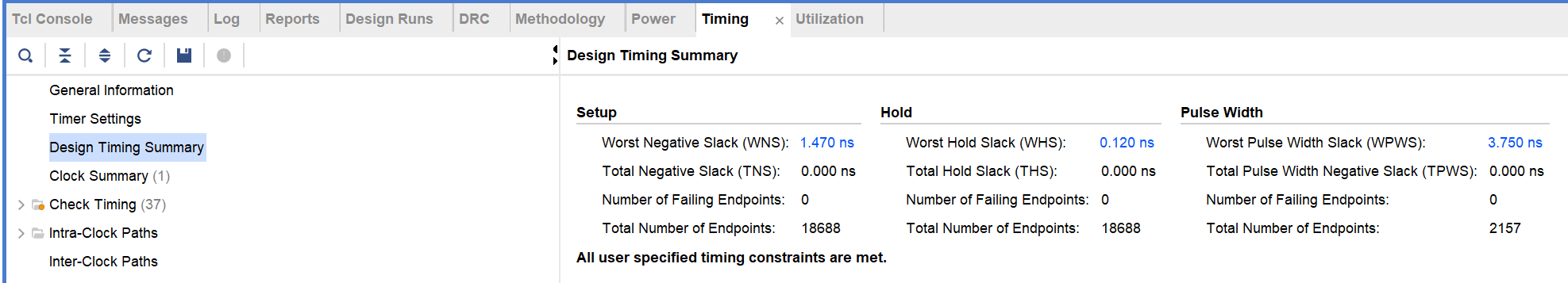


利用分布式存储器的排序电路资源

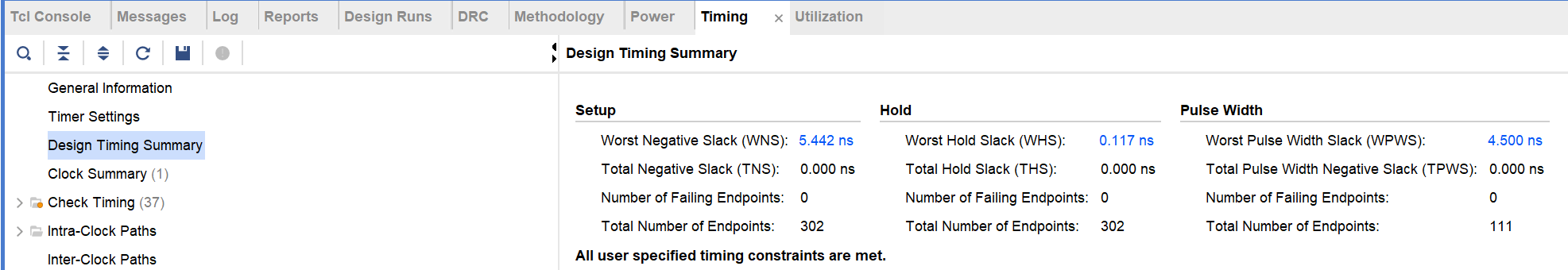


利用块式存储器的排序电路资源

* 电路性能对比



利用分布式存储器的排序电路性能



利用块式存储器的排序电路性能

* 对比结果

就电路资源而言，利用分布式存储器的排序电路消耗的电路资源较利用块式存储器的排序电路多得多；但就电路性能而言，利用分布式存储器的排序电路WNS较利用块式存储器的排序电路少，时序上性能更好。

总结篇

* 收获

这次实验我成功设计了之前提高班做的实验——排序电路，并且通过编写排序电路，理解了从过程式执行到硬件执行的转换，收获很多。本次实验中遇到了调试testbench方面的问题，放在IP核内的数据该如何跟踪？按照以前看一个数据增加一个口的方法不能做到，经过查找资料，发现可以将内部的各线网和寄存器直接拖入波形窗口，这样可以直接在不增加端口的情况方便地进行追踪。另外，在仿真时我为了增加仿真时间，将`timescale 1ns/1ps改成了1ps/1ps，改动时我没有理解这两个参数的含义，结果在使用IP核时出现了问题。这里前一个参数是clk的单位，因为IP核需要一定的建立时间，在1ns下此建立时间可以忽略，而在1ps下就不可忽略，从而导致仿真错误。这也是错误中的收获。

* 建议

1. 对于写优先的寄存器堆，PPT没有讲清楚，也没有要求清楚，建议完善。
2. 应该告知Report timing summary各参数对性能的影响，其中有些参数在互联网上资料甚少。