**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：\_\_\_\_\_\_\_\_\_徐奥\_\_\_\_\_\_\_\_\_\_\_

学生学号：\_\_\_\_\_\_\_PB20061343\_\_\_\_\_\_\_

完成日期：\_\_\_\_\_2022年3月28日\_\_\_\_

计算机实验教学中心制

2020年09月

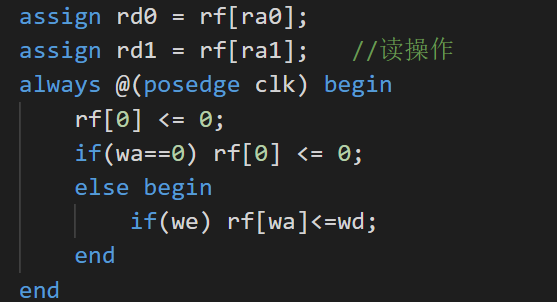
【实验目标】

1. 掌握寄存器堆（Regisster File）和存储器的功能、时许及其应用
2. 熟练掌握数据通路和控制器的设计和描述方法

【实验内容】

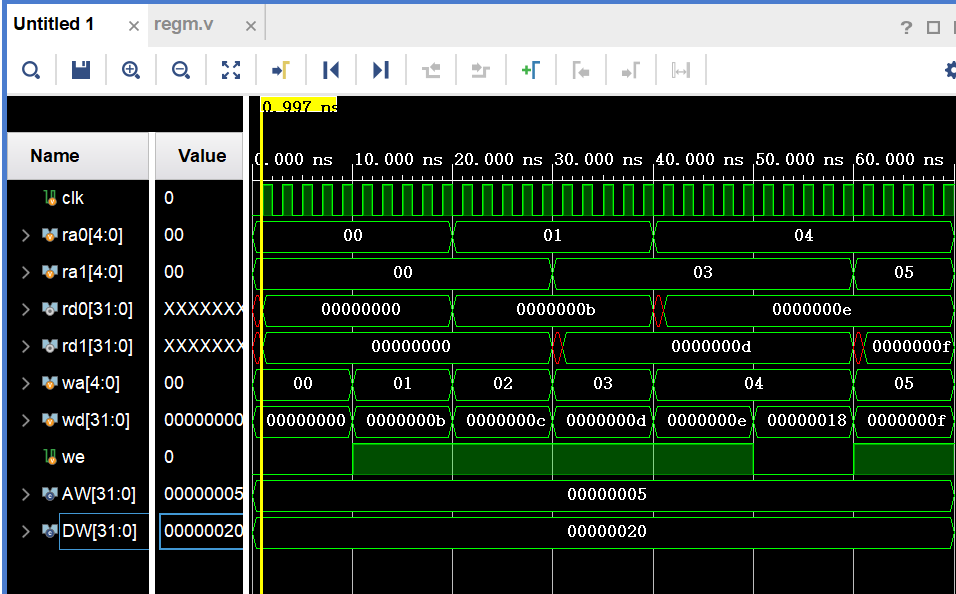
**题目1**.完成32x32位的寄存器堆的功能仿真。寄存器堆的0号寄存器内容恒为零，寄存器堆的写操作优先于读操作。

0号寄存器恒为零，加入特判即可实现。写操作优先，即一旦寄存器写入了新数据，读该寄存器的输出就应为新写入的数据。



**图1**

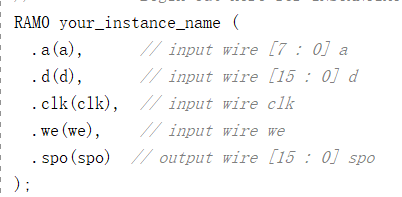
编写仿真文件，仿真结果如下：



**图2**

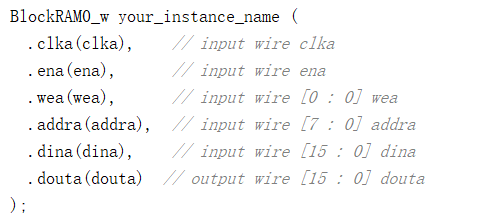
**题目2.**完成256x16位的分布式和块式单端口RAM IP核的功能仿真和对比。

分布式单端口RAM存储器IP核例化



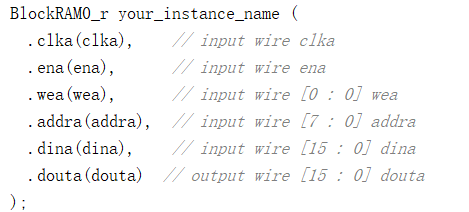
**图3**

例化写操作优先的块式存储器

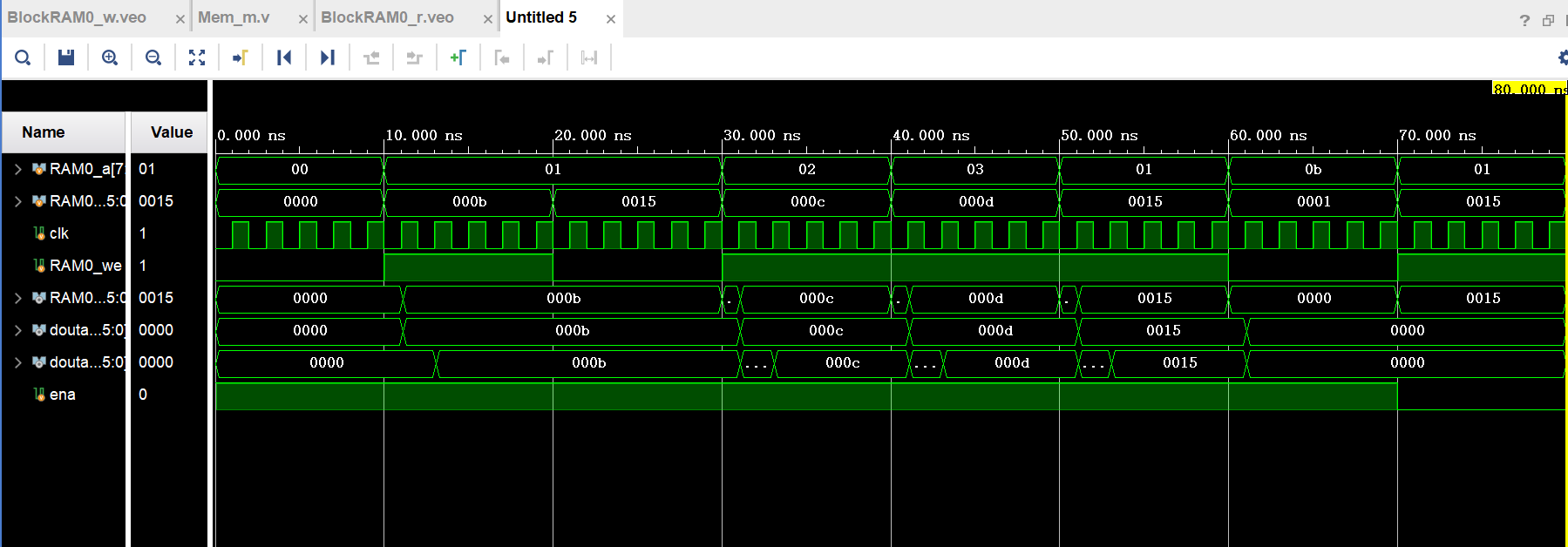


**图4**

例化读操作优先的块式存储器



**图5**

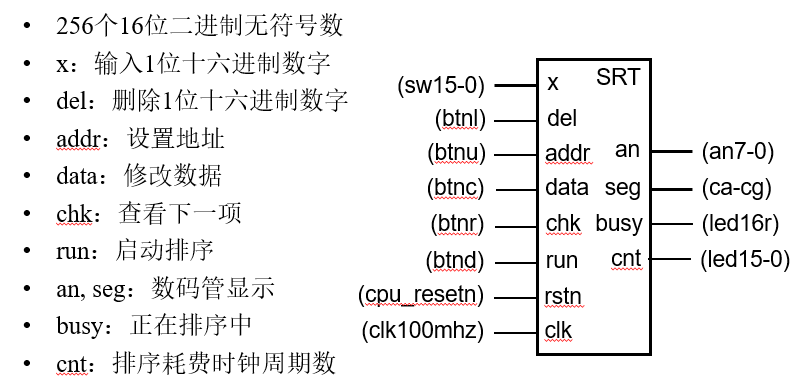
编写仿真文件，仿真结果如下：

比较可知：

1. 从写使能有效，到数据被写入寄存器，三种存储器都需要一个时钟周期
2. 分布式存储器的读操作，在给定读地址后几乎马上就可以读出相应地址的值。
3. 写优先的块式存储器，读操作要在给定读地址后等待一个时钟周期，方可读出数据，在要等待的一个时钟周期内，读出的数据仍为之前读地址的数据。
4. 读优先的块式存储器，读操作滞后读地址修改一个时钟周期，此外，如果在修改读地址时，同时向这个地址发出写信号，那么从修改读地址起，独处的数据依次为：原地址的数据，新地址在写入前的数据，新地址在写入后的数据。其中，前两个数据各占一个时钟周期。

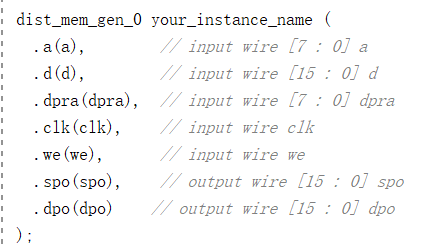
**题目3.**完成排序电路的数据通路和控制器设计和功能仿真，并将排序电路下载至FPGA中测试.

本题目要求实现数据输入、存储、查看、修改和排序等，具体要求如下：



**图7**

采用分布式双端口存储器保存数据。



**图8**

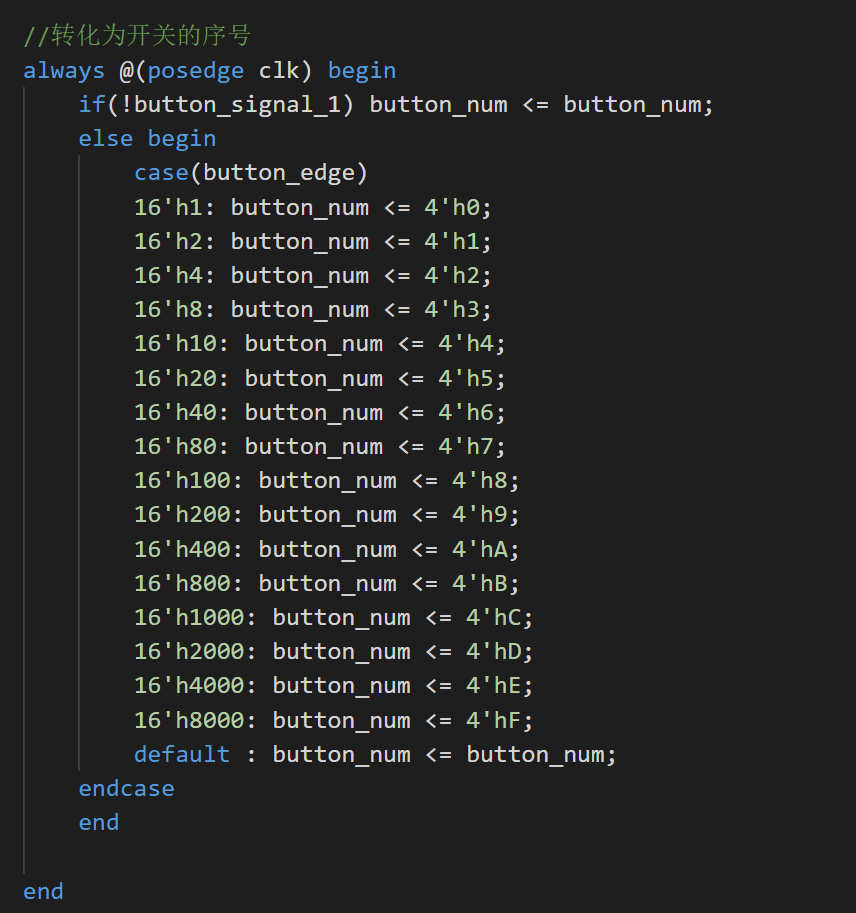
首先，实现由开关输入数据。任何时刻只改变16个开关中的一个开关状态，每次向上或者向下拨动一次开关，生成一位十六进制数h，即4位二进制数，同时产生持续一个时钟周期的脉冲p。

通过异或运算保存开关的边沿，使用缩位或运算产生1时钟周期的脉冲。在按照上述思路的实现过程中，我最初采用了组合电路，导致出现了仿真结果正确但是烧写到板子上却出现错误的问题，究其原因，是因为在组合电路的生成过程中产生了额外的锁存器。后修改为时序逻辑电路实现，达到了预期效果。

开关信号处理的核心代码如下：



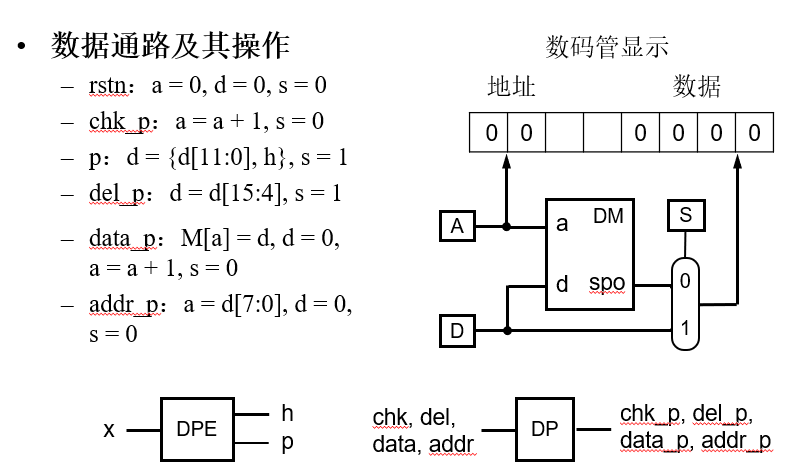
**图9**



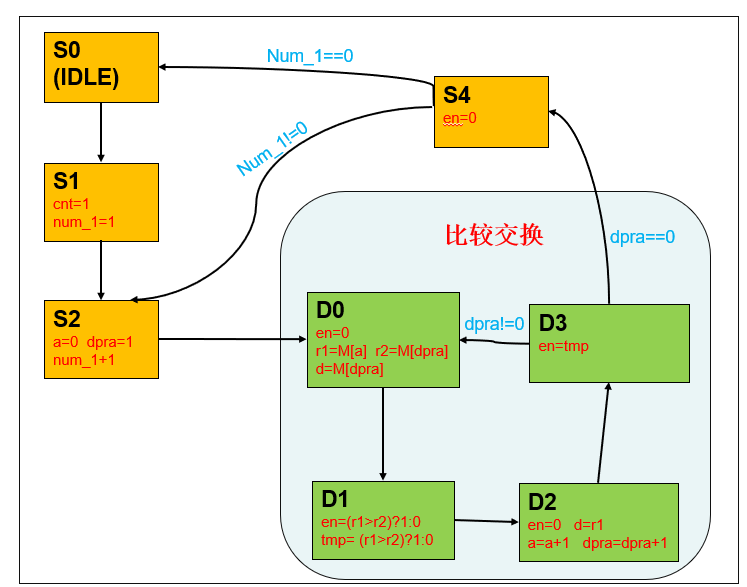
**图10**

之后，对各类按键信号进行去抖动和取上升沿处理。

根据如下图的数据通路，实现对存储器的数据输入、查看、修改、删除操作。

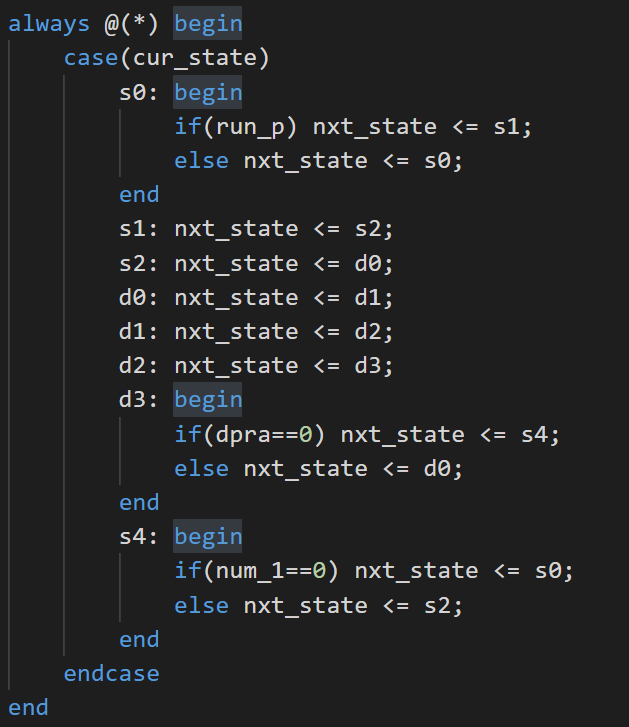


在上述控制器成功实现后，画排序电路的状态图。排序方法选用的是冒泡排序，其核心是两层循环、相邻数据比较大小。状态图如下：

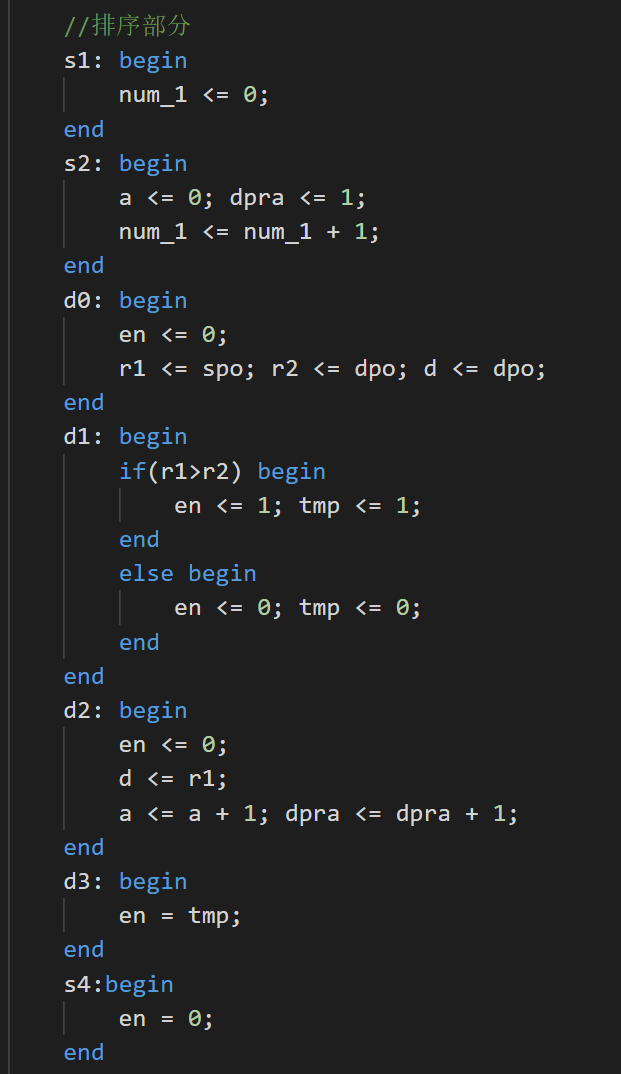


**图12**

核心代码如下：



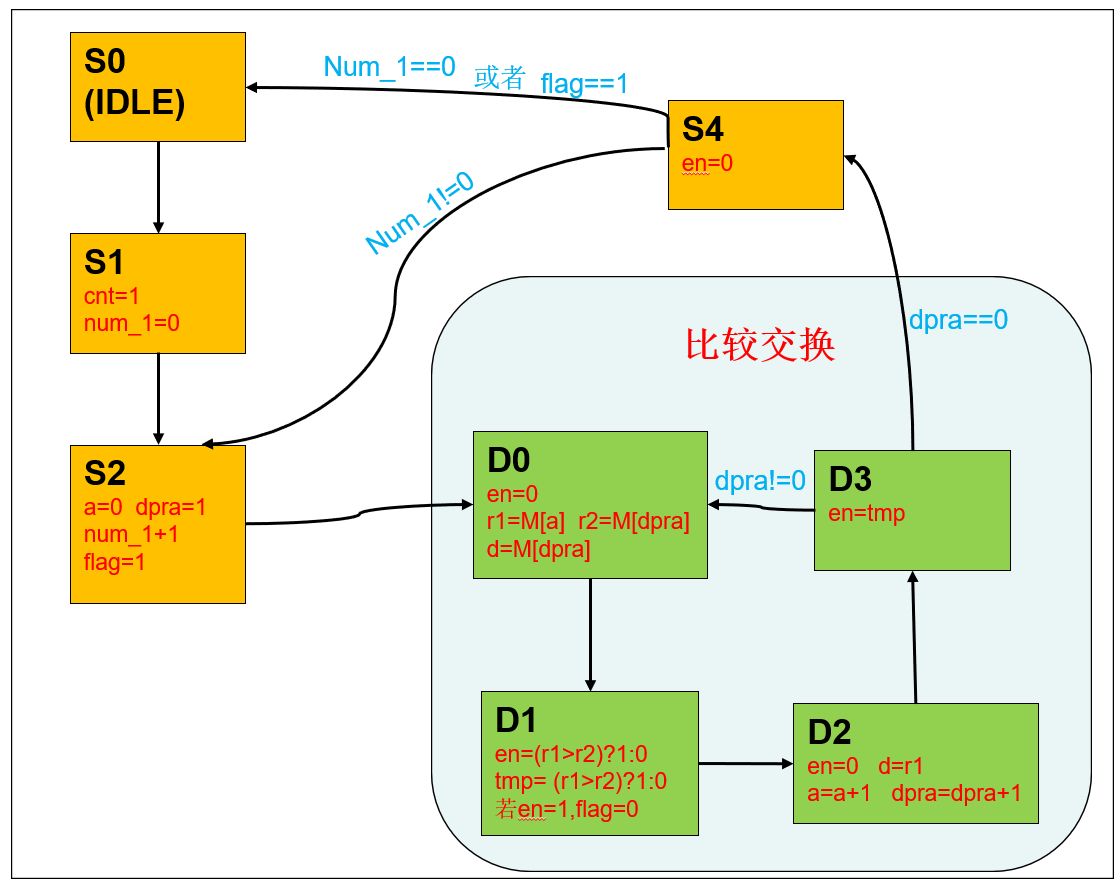
**图13**



**图14**

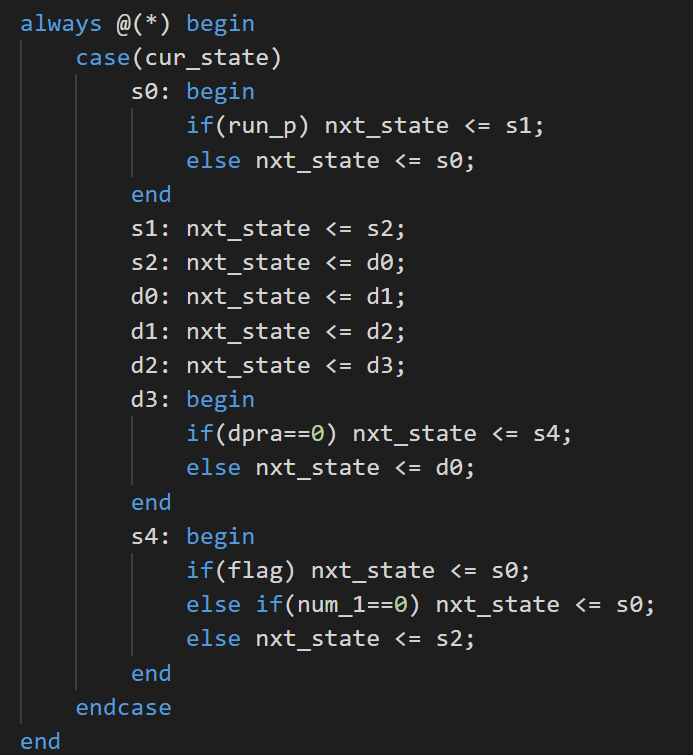
生成比特流文件，烧写到开发板上，经检验，符合预期。

**注：检查代码时，经助教提示，所以我现在加入了一个优化，引入一个标记 flag，如果内层循环没有发生交换，那么直接停止。改进后的状态转换图如下：**



**图15**

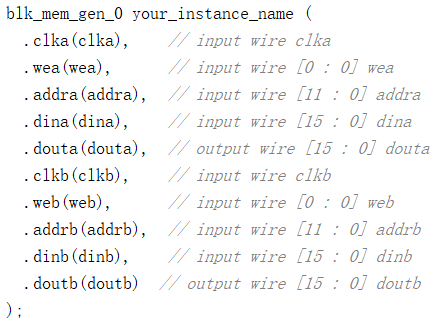
改进后的代码如下：



**题目4**.将数据量增大至4096x16位，分别采用分布式和块式存储器保存数据并排序

分布式存储，则只需要更改生成IP核的数据数量，代码中需要将存储器的地址位宽由8位增加到12位，增加一个数码管用于显示多出来的地址。

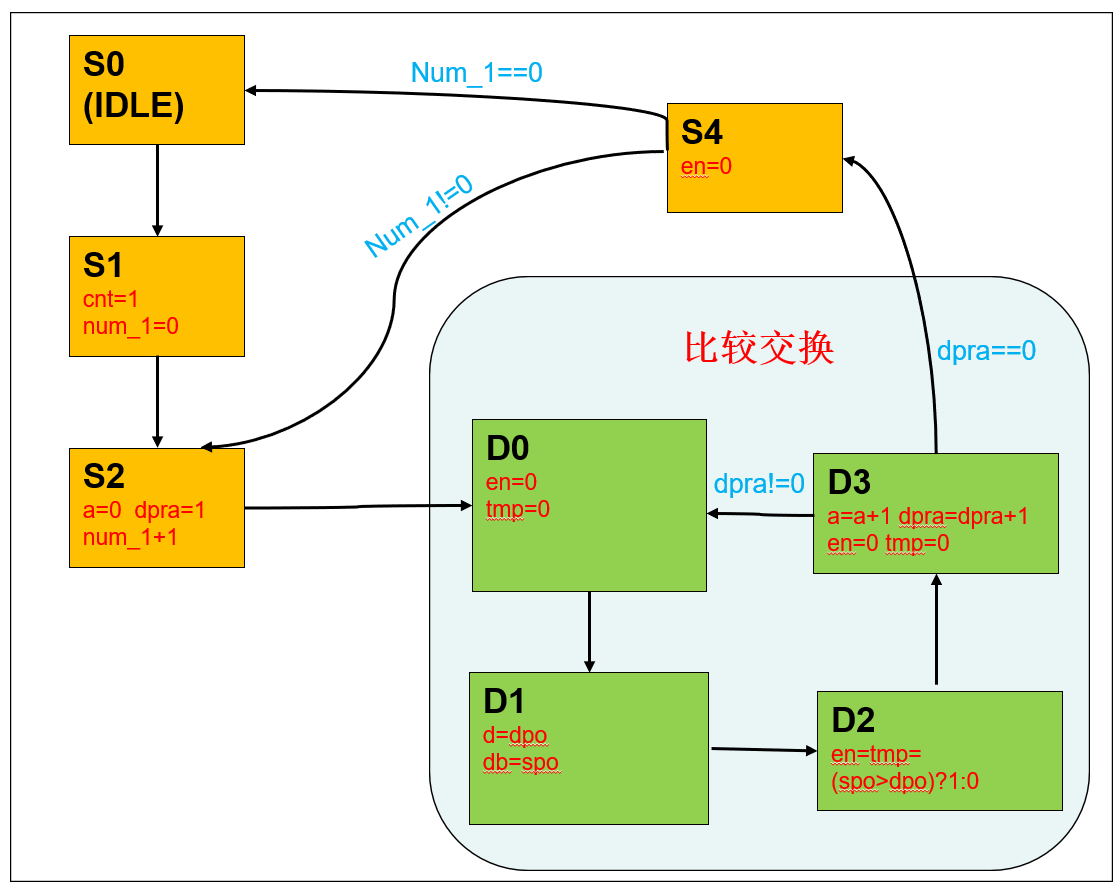
而块式存储，则需要使用真实双端口存储器，它有两个写端口，所以可以更容易实现数据交换。



**图17**

但是需要注意的一点是，它是在数据上升沿读数据，也就是说，如果在当前时钟周期更改了读地址，那么需要在下下一个时钟周期才能获得新地址的稳定内容。

状态图如下：



**图18**

这两个4096位的排序，烧写后，均能正确排序，但是需要的时间比256位的有了明显的增加。

【总结与思考】

1. 本次实验我再次熟悉了IP核的使用，并且对于IP的各项设置有了更清晰的理解，并且对于三种寄存器堆的输出特性有了初步的认识。
2. 本次实验中，我花费了近三个小时的时间进行开关信号处理的debug，最终发现是由于组合逻辑电路生成了额外的锁存器导致，这激励着我更多地去写时序逻辑电路，虽然因为它可能对某些变量存在一个时钟周期的滞后，但是最终结果一定是可以在编写代码时就能预见到的。
3. 在实现排序电路的过程中，我对于状态图、时序逻辑电路的理解进一步加深，明白了状态图在实现复杂电路的重要作用，练习如何划分状态等技巧。

【附录】

1. 题目一的代码和仿真文件
2. 题目二的IP核和仿真文件
3. 题目三的源代码和引脚约束文件
4. 题目四，分布式存储器实现4096位排序的代码
5. 题目四，块式存储器实现4096位排序的代码