**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：\_\_\_\_单周期CPU设计\_\_\_\_

学生姓名：\_\_\_\_\_\_\_\_徐奥\_\_\_\_\_\_\_\_\_\_

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

完成日期：\_\_\_\_2022年4月13日\_\_\_

计算机实验教学中心制

2020年09月

【实验目标】

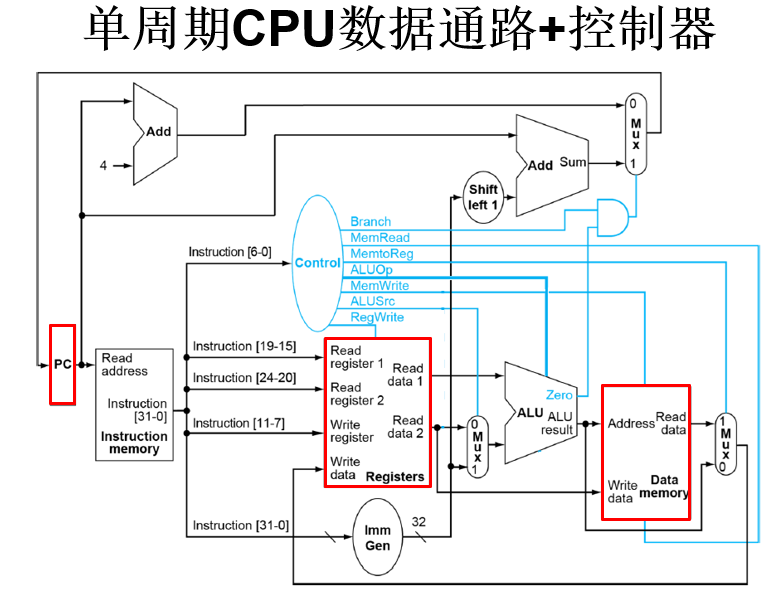
1. 理解单周期CPU的结构和工作原理
2. 掌握单周期CPU的设计和调试方法
3. 熟练掌握数据通路和控制器的设计和描述方法

【实验内容】

**题目1**.设计单周期CPU，将CPU和PDU整合后下载至FPGA，进行逐条指令功能测试。

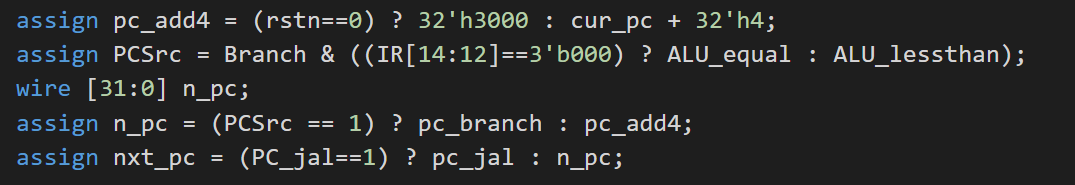
首先，指令寄存器中在FPGA运行过程中，其内容不会改变，所以采用分布式ROM，使用Lab3生成的COE文件初始化。数据寄存器使用双端口分布式RAM。寄存器堆直接在代码中定义。

之后，就根据下图编写各个模块以及各种控制信号。



**图1**

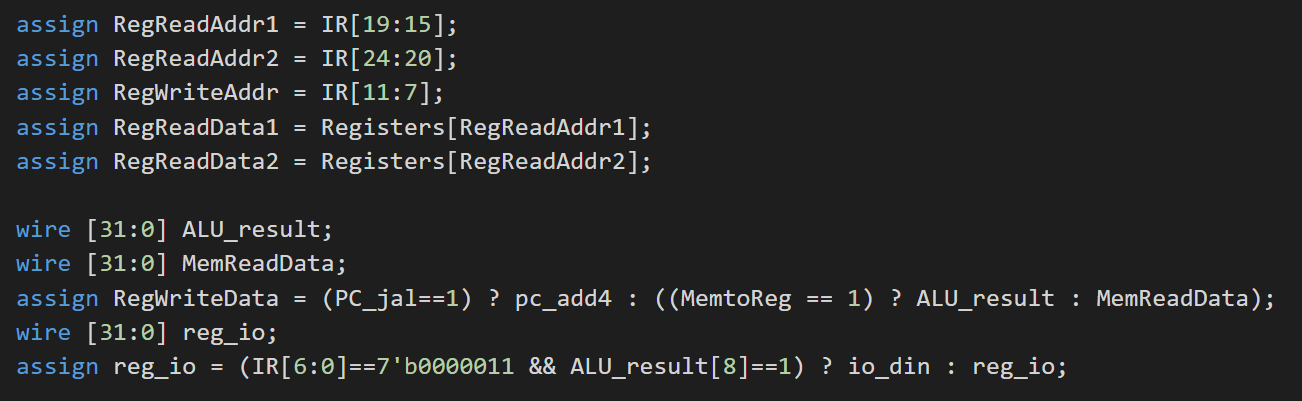
PC部分，在每个上升沿，当前pc变成nxt\_pc，而nxt\_pc有三种来源，一是pc+4，二是条件分支指令在条件满足时的目标地址，三是直接跳转指令的地址。nxt\_pc的核心代码如下：



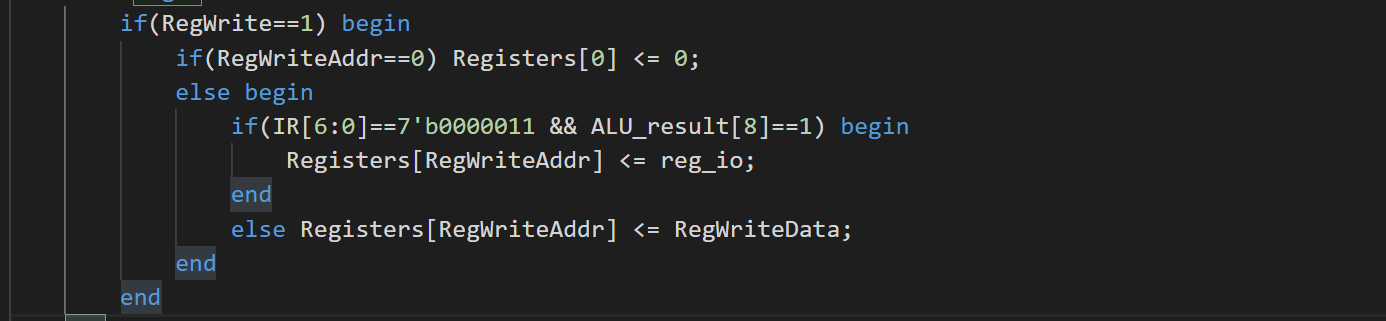
**图2**

指令部分，主要的处理是将当前pc值减去代码段起始地址32’h3000，再除以四。

寄存器堆，直接定义32个寄存器，寄存器的读操作是组合的，写操作是在时钟上升沿、并且寄存器写信号有效时写入。需要特判如果数据来源是IO。核心代码如下：



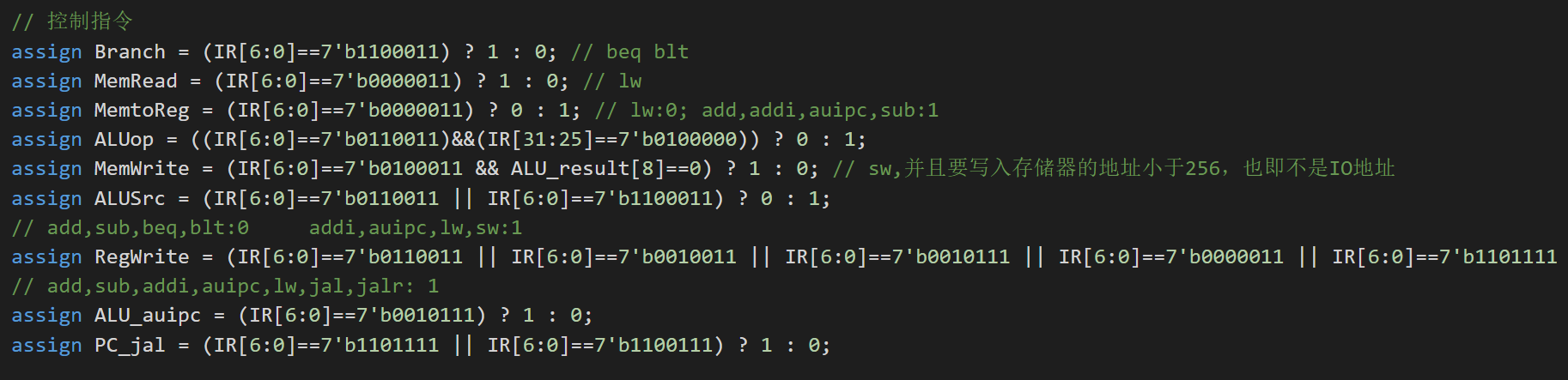
**图3**



**图4**

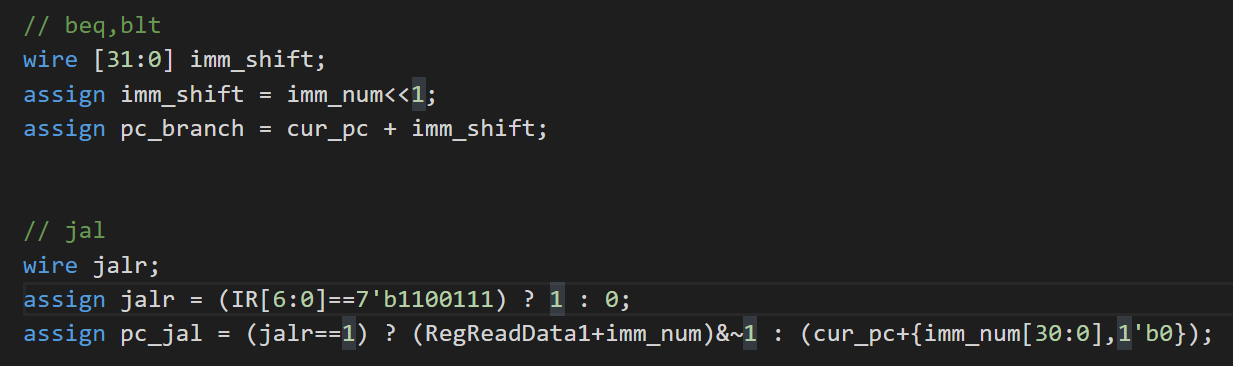
数据段直接用双端口分布式RAM实现。不过在汇编代码中，相邻数据项的地址相差4，而RAM中相邻数据的地址相差1，需要对数据地址除以4.

控制指令，主要是各类读写控制信号以及多路选择器，根据指令的值等确定。除了图1给出的控制信号外，我额外加了几项，分别处理jal和jalr指令、auipc指令时ALU的第一个操作数应选择为为pc。核心代码如下：



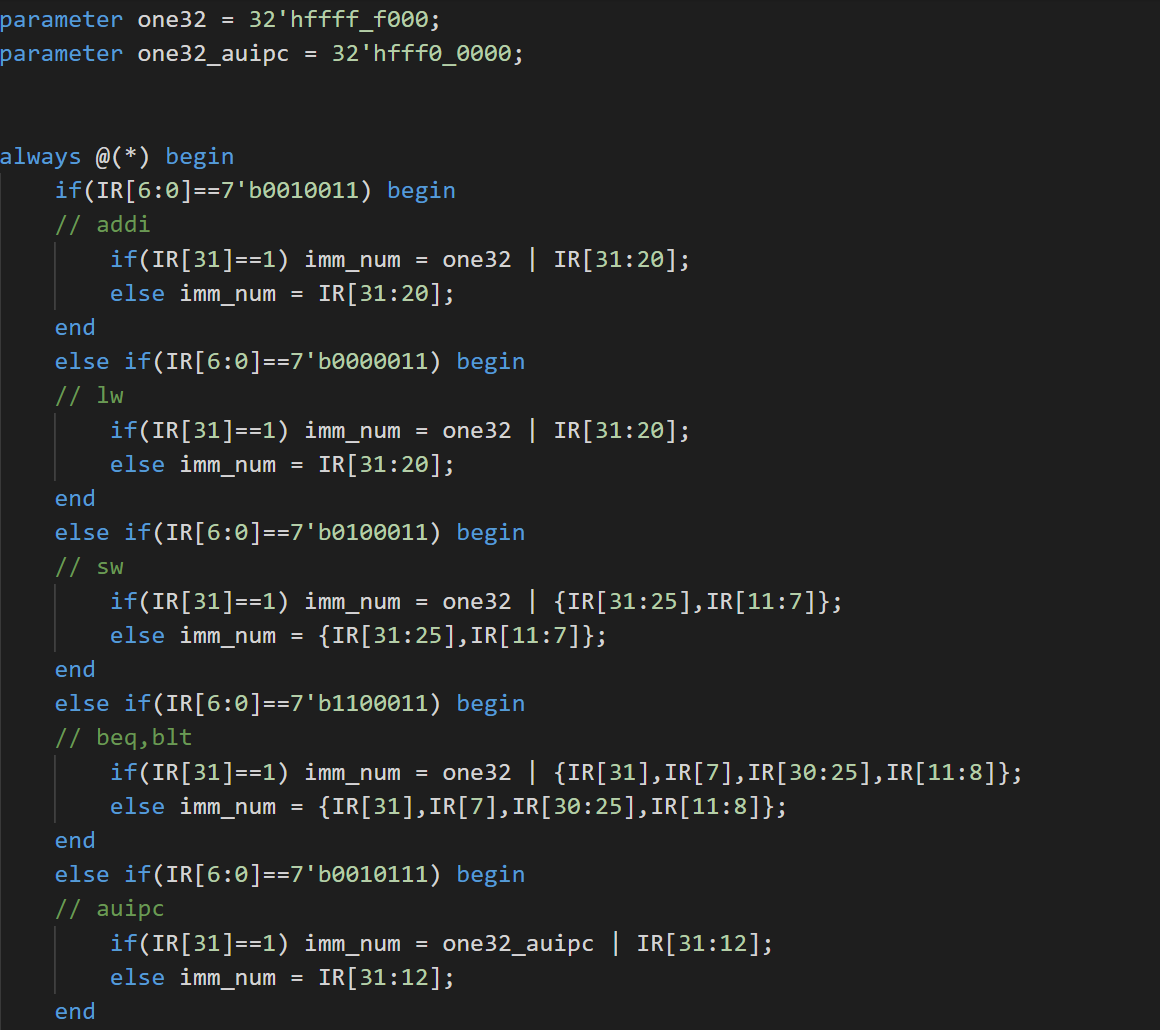
**图5**

条件跳转以及直接跳转指令的目标地址为：

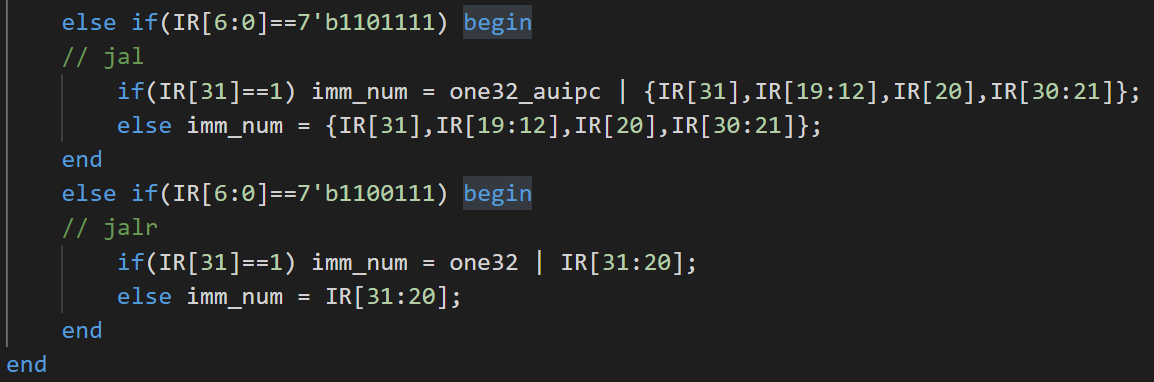


**图6**

立即数的处理，包括三个方面：一是有些指令的立即数并不在指令中连续，需要做拼接；二是对立即数进行符号扩展；三是条件跳转指令的地址需要左移一位。



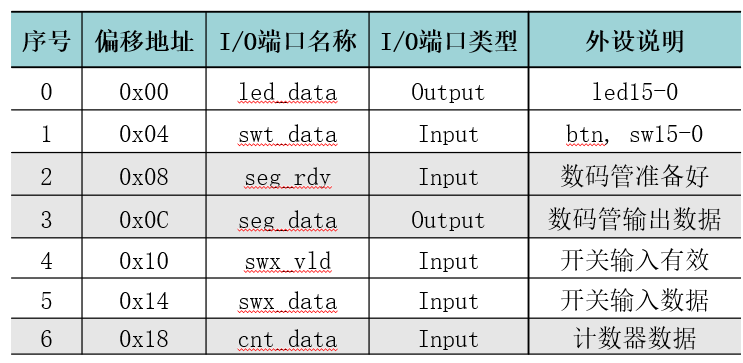
**图7**



**图8**

ALU部分，在要实现的10条指令中，只涉及对ALU的加和减法操作，此外还有两个标志变量，分别是相等以及有符号小于。此外，需要根据指令源操作寄存器和目标操作寄存器在指令中的位置，求出地址。

需要对IO的load和store做特殊处理。本项目规定实际memory的地址为0x0000到0x00ff，IO映射到的地址从0x0100开始。IO地址和外设对应如下：

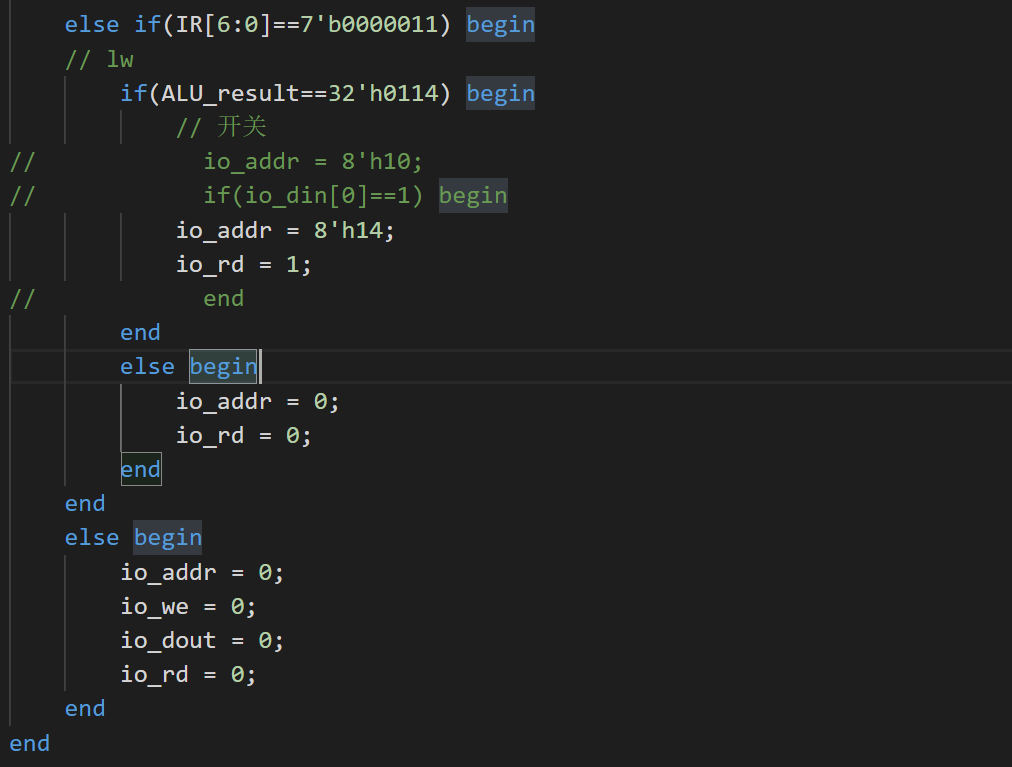


**图9**

核心代码如下：

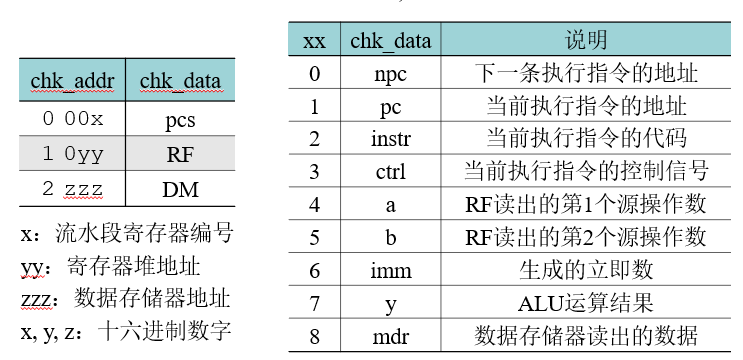


**图10**



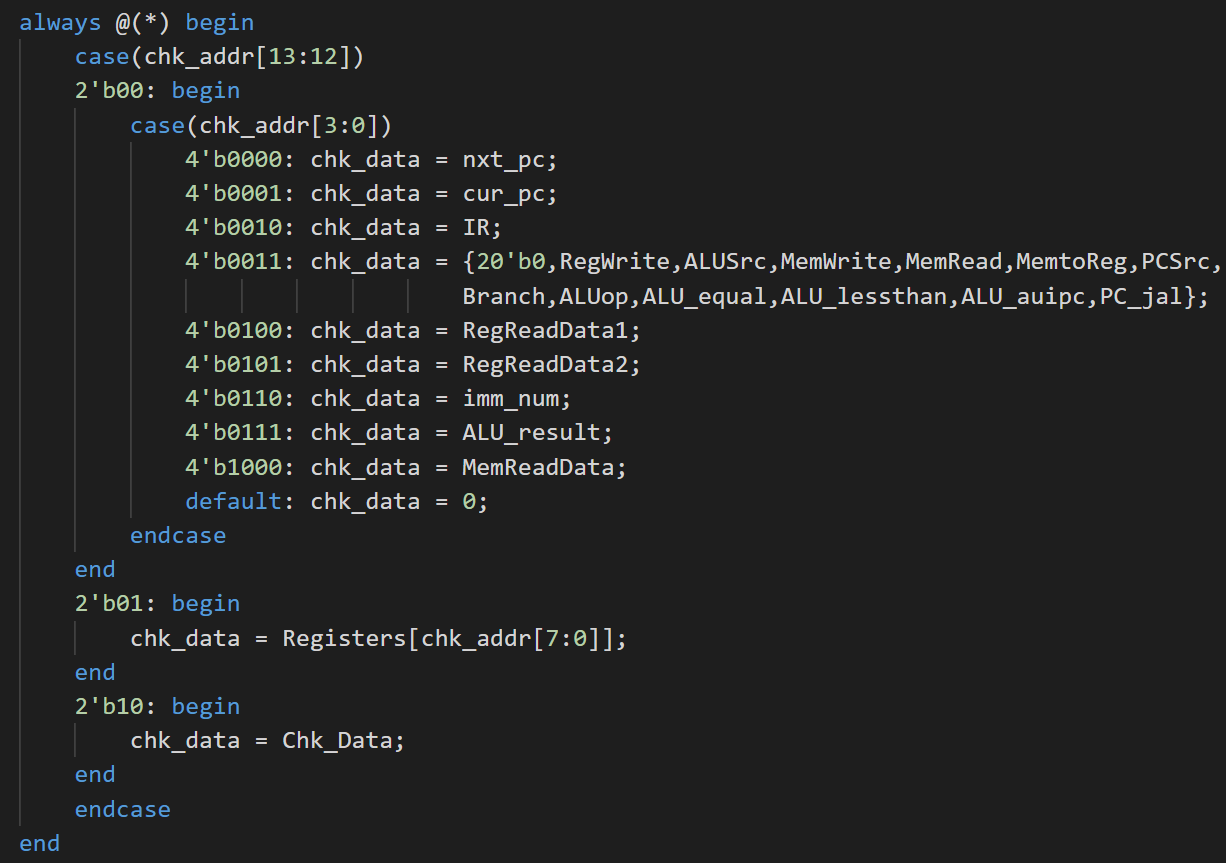
**图11**

最后，需要处理调试总线，调试总线是在pdu处于调试状态时，与cpu之间的数据交互，当CPU停止时，利用开关编写地址，按动chk键，chk\_addr和数据通路状态（chk\_data）分别显示在指示灯led和数码管seg上，再次单独按动chk,将顺序显示后续信息。chk\_addr对应的值如下：



**图12**

核心代码为：



**图13**

之后，编写引脚约束文件，烧写到FPGA开发板上。

Lab3的测试指令的汇编代码，是利用led灯来进行测试，在本项目中，也即要将结果store进0x0100的位置，就可以根据led的亮暗，看到相应的值，进而判断指令的正确与否。

经测试，符合预期。

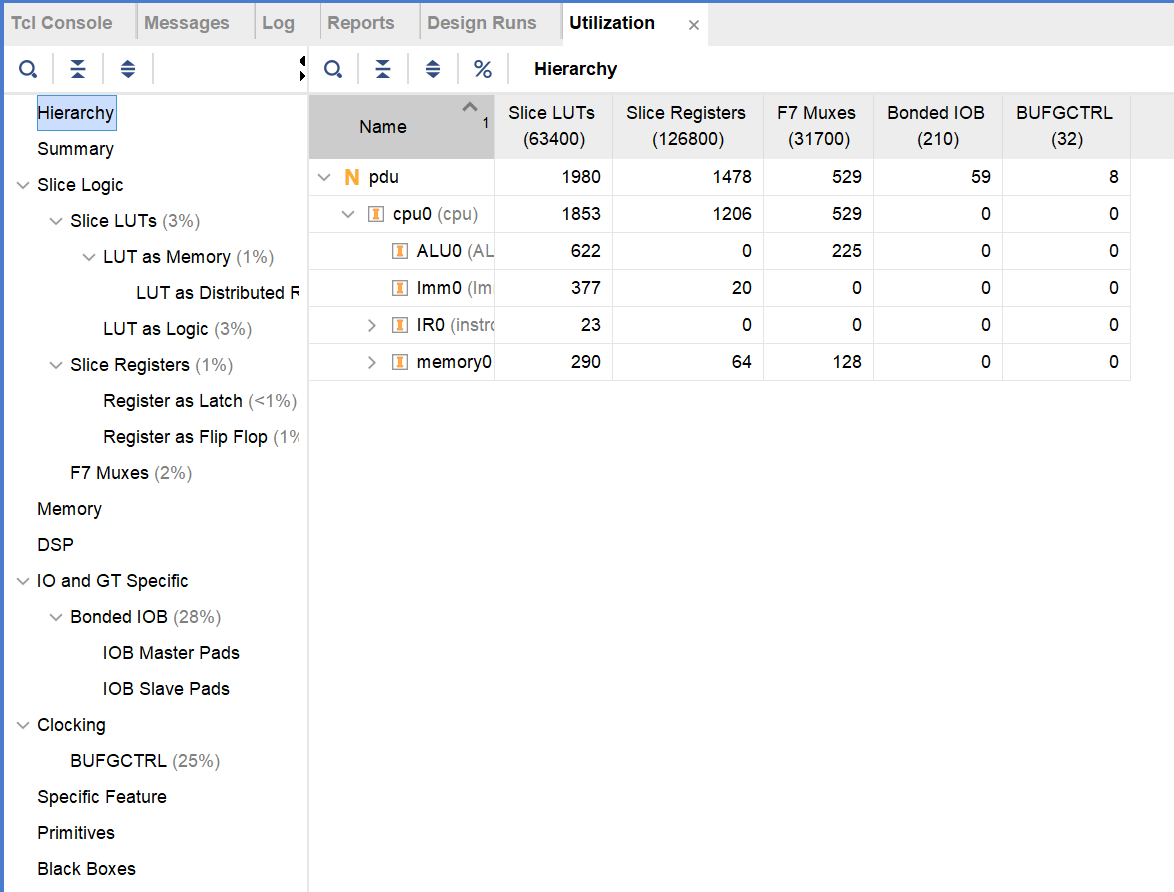
**题目2.**将CPU和PDU整合后下载至FPGA，进行排序程序测试。使用Lab3实验步骤二生成的COE文件初始化。

单周期cpu以及和pdu的交互实现与步骤一的实现过程基本相同，指令寄存器的COE文件是来自第三次实验的排序排序代码生成的，而数据存储器的初始化文件中保存的是待排序的代码。

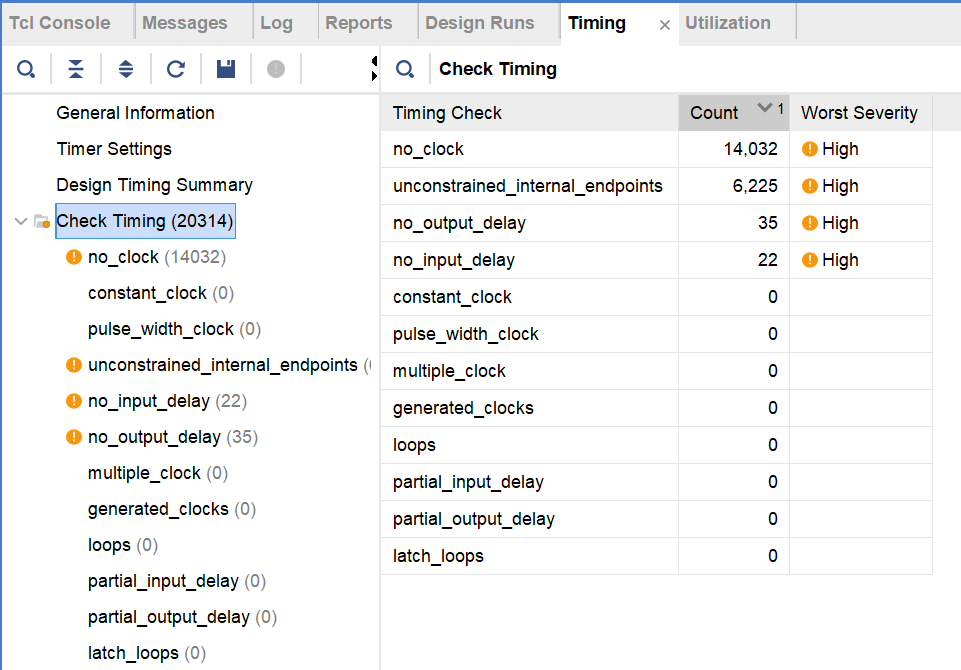
测试过程中，将断点设置在最后一条指令的下一个地址，按动cont，执行结束后，查看数据存储器的值。

经测试，可以将数字按升序排列。

查看电路资源使用情况和电路性能。

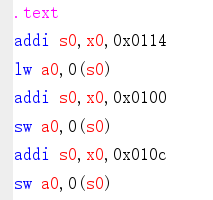


**图14**



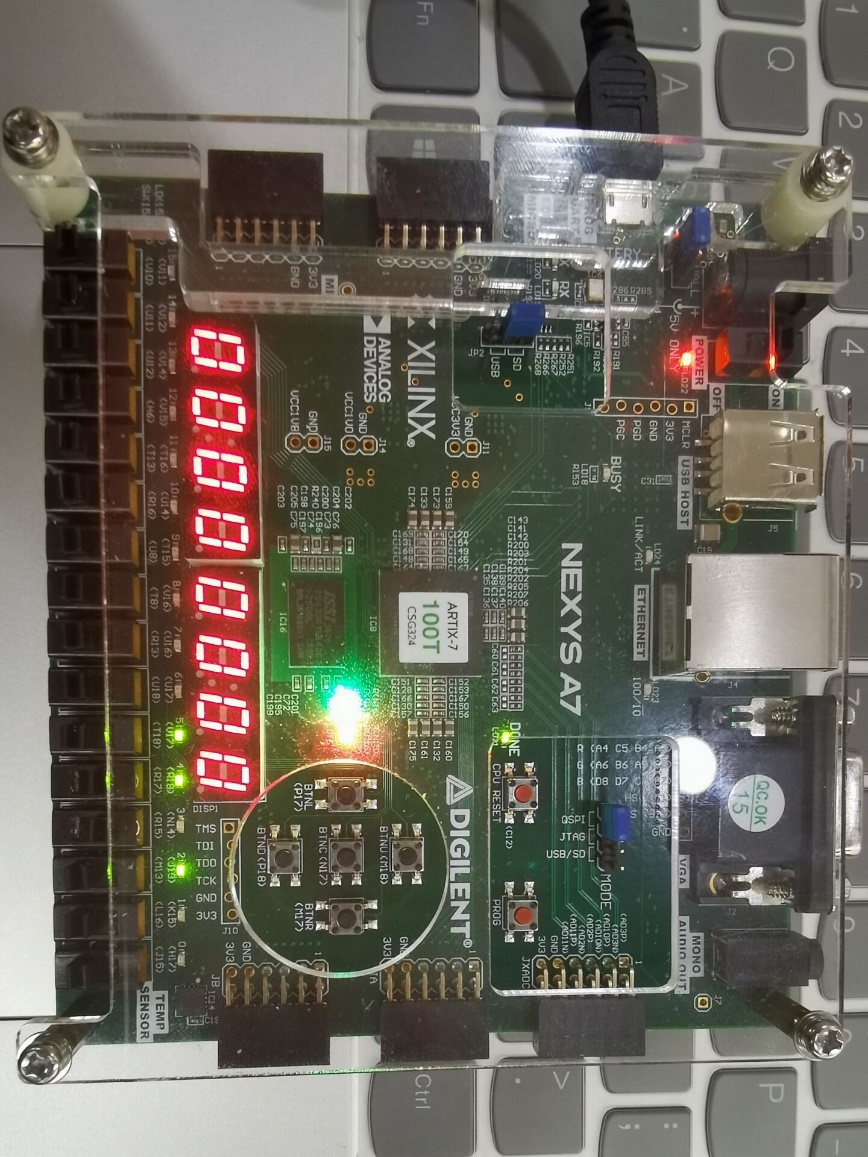
**图15**

此外，上述测试中并没有使用到开关输入和输出到数码管，所以我写了一段简单的测试汇编程序，生成COE文件，初始化指令存储器，进行测试。代码如下：

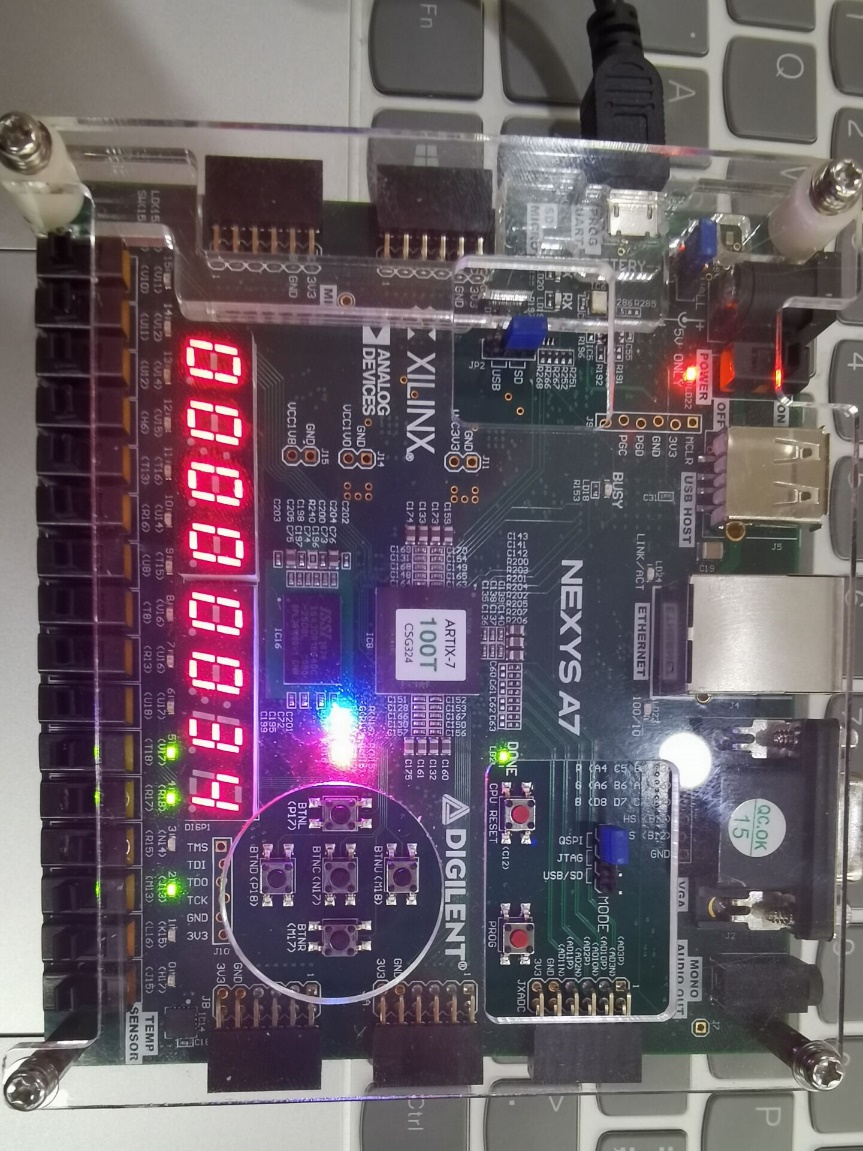


**图16**

首先，会从开关（地址为0x0114）读取数据，存到的寄存器a0，比如将开关拨动为34，即0x34,按动data。然后将a0的值输出到led（地址为0x100）,最后会将a0的值输出到数码管。后两个输出的FPGA开发板图片如下：



**图17**



**图18**

【总结与思考】

1. 本次实验难度较大，不过时间由原来的一周延长至两周，时间安排也相对没有那么紧张。
2. 本次实验收获较多，最重要的是可以深入到代码实现的具体细节上，体会单周期cpu是如何运作的。而且verilog是硬件编程语言，与实际的电路比较接近，所以其实是从电路的角度体会了单周期cpu的执行过程
3. 本次实验提供了pdu，减轻了工作量，虽然存在一些bug，不过pdu的代码相对比较易读，所以bug也不难修改。
4. 最后，本次实验给出的数据通路和控制器图并不全，缺少jal、jalr、auipc的数据通路和控制信号，建议之后的实现文档将这一部分进行补充。

【附录】

1. Lab4\_cpu\_test.srcs题目一的代码文件，包括设计文件、仿真文件和引脚约束文件
2. Lab4\_cpu\_sort.srcs题目二的排序代码，包括设计文件、仿真文件和引脚约束文件