实 验 报 告

学号：2014K8009929022 姓名：孔静 专业：计算机科学与技术

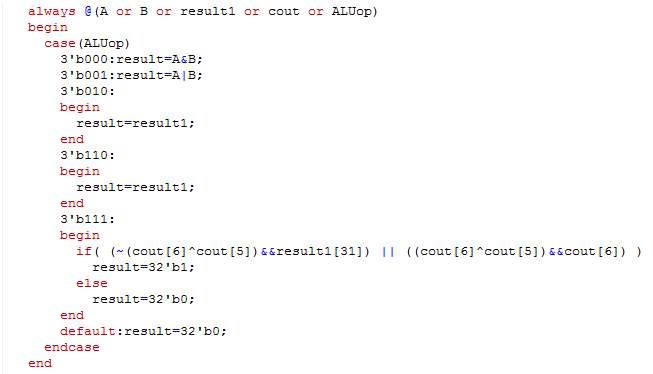
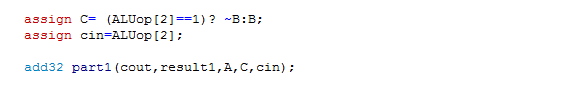
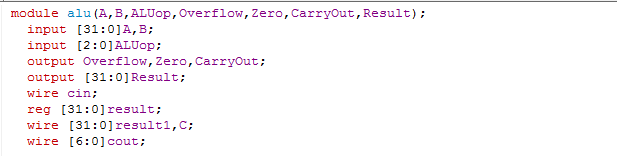
实验序号：1 实验名称：基本部件单元设计

1. **代码以及波形图**

**Alu：**

**Part1代码**

alu部分



根据定义，ALUop，result，以及加法器最高位和符号位的进位，给相应结果赋值

连接加法器进位输出

寄存器，存放结果，最后assign给输出Result

case根据ALUop的不同，给result赋值相应结果

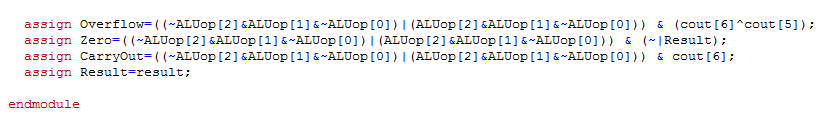
执行slt操作时，先进行减法操作，根据得数的符号位正负判断大于小于：如果结果溢出，根据进位判断正负；如果结果不溢出，根据符号位判断正负

加法，调用加法器A+B+0，减法调用加法器A+~B+1

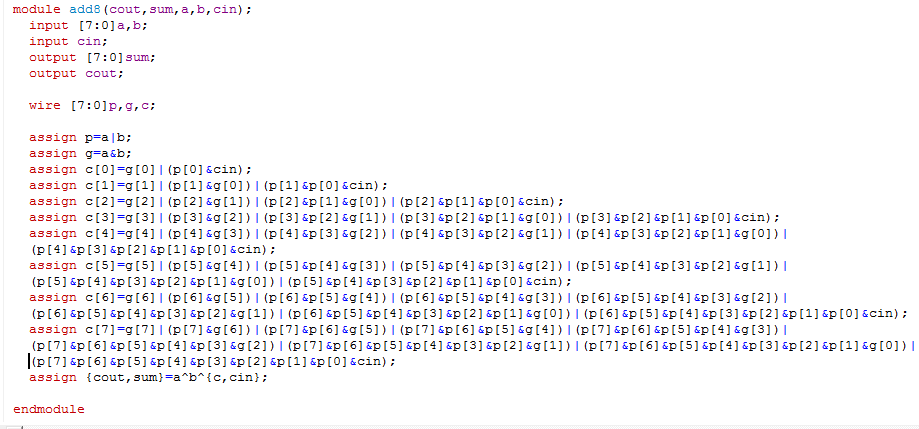
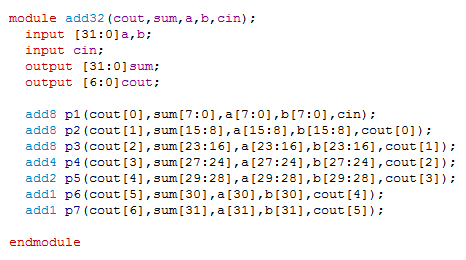
result1连接加法器的输出结果，C为中间变量且连接加法器输入

调用了加法器，增加进位变量

定义信号

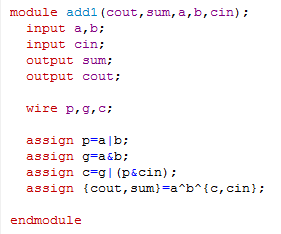
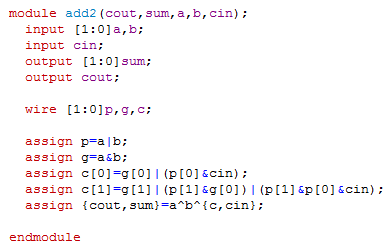
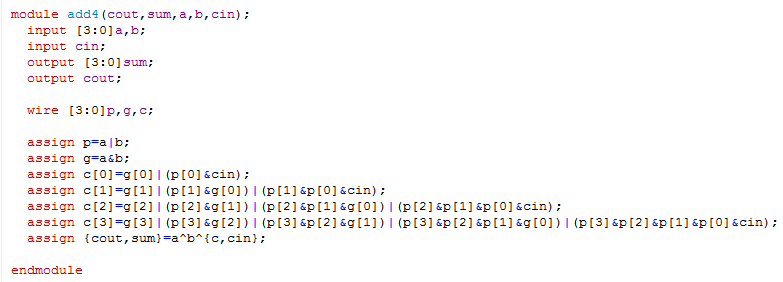


加法器部分

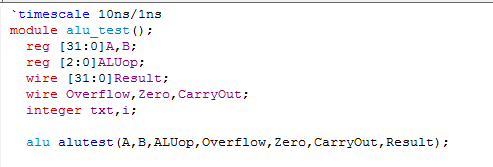
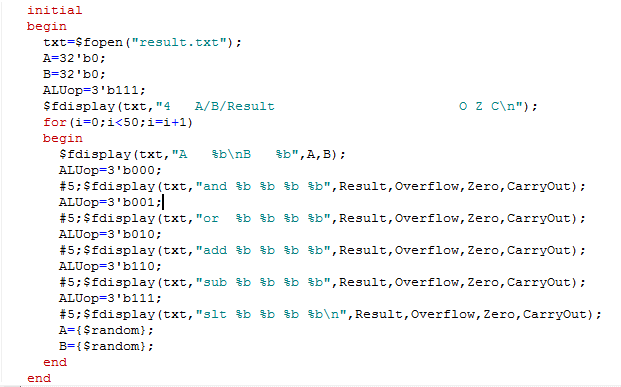


上学期写的8位超前进位加法器，输入[7:0]a，[7:0]b，cin，输出[7:0]sum，cout

因为以前写过8位加法器，因此这个32位本来是由4个8位加法器组成，可是Overflow等输出需要最高位的进位，所以改写成3个8位加法器+1个4位加法器+1个2位加法器+2个1位加法器的7个加法器组成，cout【5】保存最高位的进位，cout【6】保存符号位的进位



**Part2Testbench代码以及部分simulate波形图**



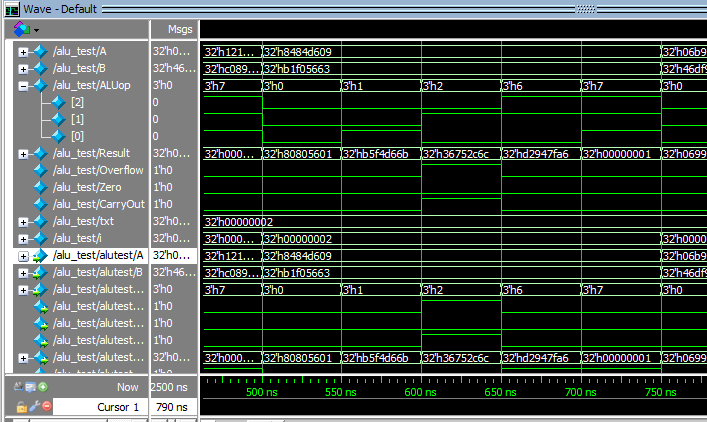
隔50ns，ALUop按顺序改变，000,001,010,110,111

隔250ns，A,B改变

调用alu

定义相应变量

时间精度设定



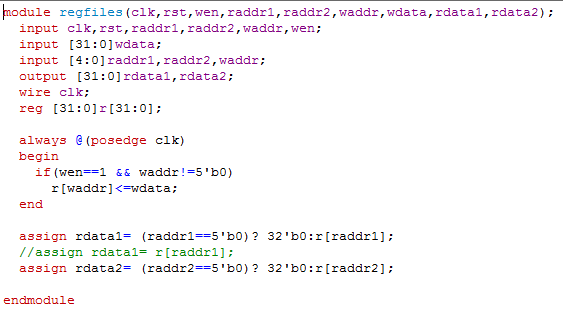
隔250ns，A,B改变

隔50ns，ALUop按顺序改变，000,001,010,110,111

隔50ns，根据输入数据不同，输出不同的结果Result,Overflow,Zero,CarryOut

**Regfiles：**

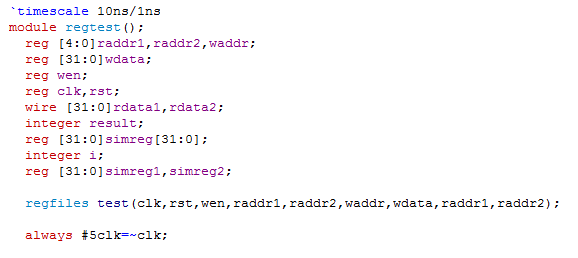
**Part1代码**



当时钟信号上升沿，且写能信号wen为1，且写入位置不在5’b0的时候，在寄存器堆相应位置写入数据

判断读取位置是否是5‘b0，输出相应结果

**Part2Testbench代码以及部分simulate波形图**



每5ns，时钟信号改变

定义相应变量



判断，输出

随机地址，以及数据，以及写能信号

输出文件

图1，最开始的波形图

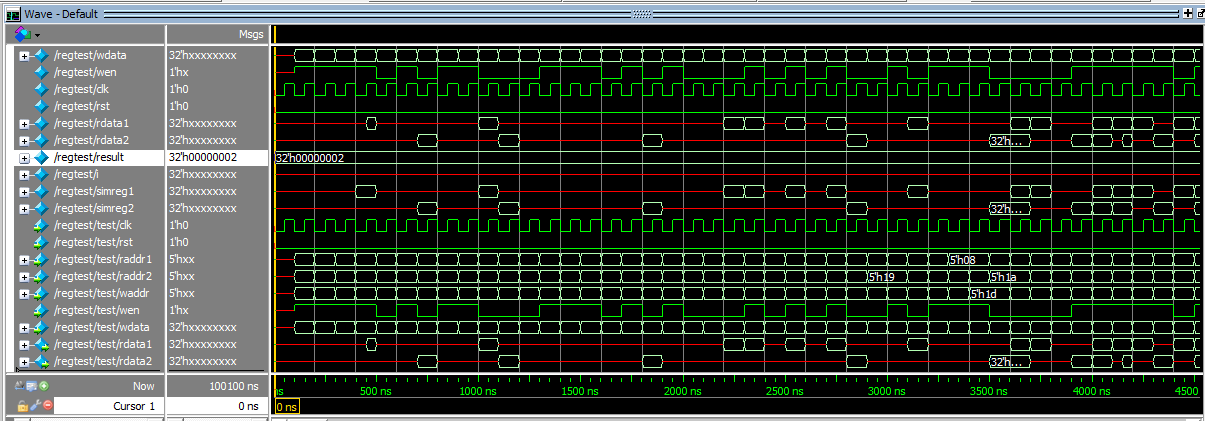
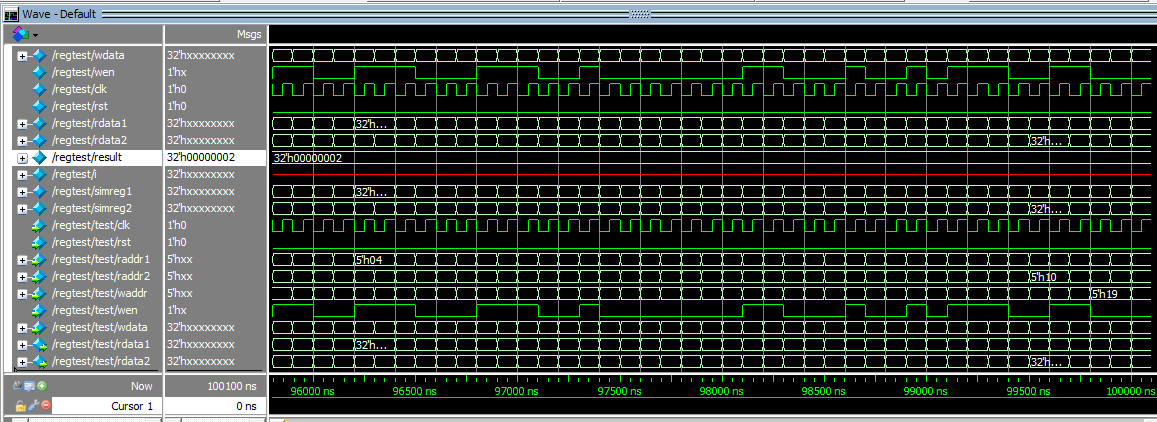


图2，重复1000次读写过程，最后部分的波形图



（因为我的testbench，所有的数据、读取写入地址以及写能信号全都是随机的，所以开始寄存器里基本没有值，所以基本是红线，最后重复次数多了，基本上所有位置都有了数据，所以红线没有了）

每10ns，进行一次随机数据，输入输出，相应数据改变。

1. **问题合集**

问题有很多，大部分是因为粗心等原因造成的，敲键盘打错（一直把data拼成了date），忘记了“；”号，因为在中文输入法下“；”以及“;”（全半角的错误），最后忘记了“endmodule”和“endcase”等等，此类问题略。其他问题依靠着transcript、百度还有同学的帮助下解决，但好多都忘记了是什么问题，以后会记得用//绿色保留错误版本。

（注，以下截图代码，绿色部分均为修正前，其他部分为修正后）

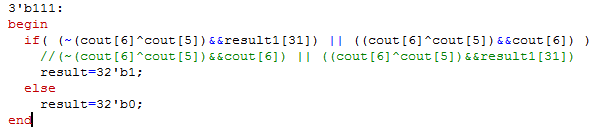
**Problem1**



没有将所有always内部涉及的数据写在（）内，导致程序出错，没有执行always操作。

后将所有涉及的数据都补上了。

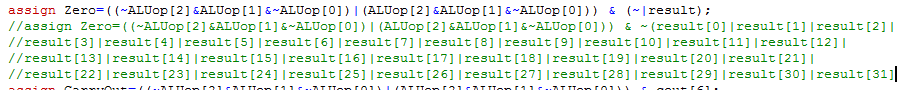
**Problem2**



逻辑错误，开始情况是溢出时候判断符号位，未溢出时判断符号位进位，判断反了，导致slt结果相反。

后根据输出的txt文本发现此问题，并修正。

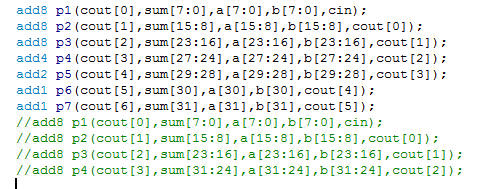
**Problem3**



原本没有使用归约操作符，对每一位进行操作，复杂又容易打错，并且当位数更多时，全部打出显然很复杂。

后在助教老师指导下，看IEEE Standard for Verilog Hardware Description Language, IEEE Std 1364 -2005第5.1节部分，改用归约符号，简洁方便。

**Problem4**



原本32位加法器中，直接由4个8位加法器组成，导致最高位的进位未知，许多结果无法得出（如Overflow)

后将最后一次进行8位加法器拆分为，4,2,1,1位加法器的组成，cout[5]储存着最高位的进位，cout[6]储存着符号位的进位

**Problem5**



最开始版本是最下面and wen，但在compile环节出错，不知道为啥，但我尝试着改成&后编译成功，逻辑上似乎也没啥错误，当wen位0的时候，不管clk如何，永远没有上升沿，所以不执行写入。

后助教老师看过程序之后，告诉我在always里面只能写posedge/negedge/and/or，如果加入&，会综合出一个blablabla，具体忘记了，反正就是容易出错。修正如上，外加在always内部加上if判断wen。

**Problem6**



一开始直接给rdata1赋值r[raddr1]，这个问题也跟我程序有关，我主体程序中并没有对r[0]进行赋值，并且如果写入地址为5’b0的时候，不进行写入，所以如果当读取地址为5’b0的时候，无输出，波形图为红线，高阻态。

后在assign内部加一个？：判断语句，修正了这一错误。

1. **对讲义中思考题的理解和回答**

**Question1**



寄存器属于时序逻辑电路。

时序逻辑电路，与组合逻辑电路的不同就是，它包含触发器，即它不仅和输入信号有关，还跟以前的电路状态有关。

**Question2**



reg [a:0]r[b:0] 表示b+1个a+1位的寄存器

1. **对于此次实验的心得、感受和建议**

在以前做过数字电路大作业alu的基础上，此次作业并不难，并且操作相对之前那个alu少很多，工作量没有那么吓人，（不好意思的说，我之前的alu没好好做，因为功能太多了，打开电脑就崩溃的感觉），并且有清晰的中文解释，所以觉得很好。（如果IEEE standard for verilog也能有中文版就好了，看到英文真的头疼。）

写+调试代码的时候，出现了各种大错误小错误，有的自己解决了，有的在同学（在此感谢谈清扬的mvp，以及张传奇/张北辰/方言歌等同学的助攻）的帮助下顺利解决，还有的由助教帮忙指出并解决。顺便高兴一下，自己也可以帮助解决以前没学过数字电路的同学的部分问题。（话说，助教能给个除了邮件之外的联系方式吗，比如qq比如微信什么的，有的东西问同学，只知道怎么改，不知道为什么这样改。还有和同学交流产生疑问，需要跟助教求证，所以想要一下联系方式。）