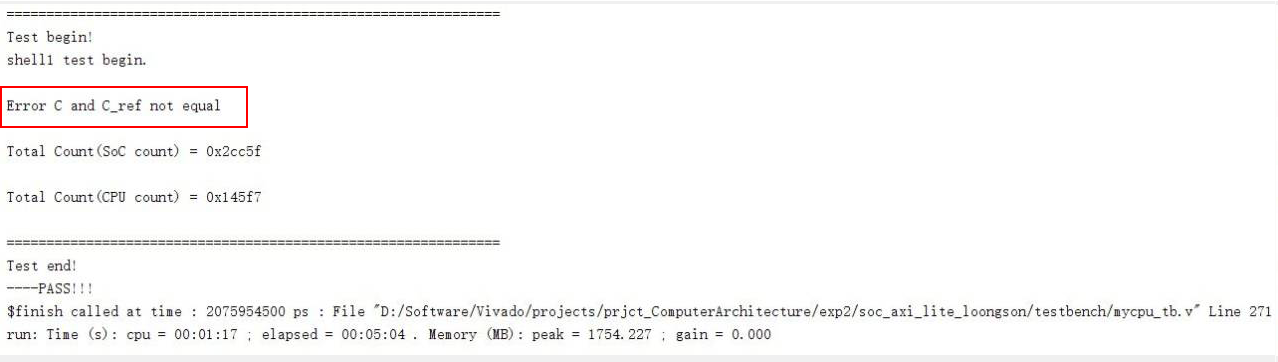
# 实验2实验环境纠错

**错误表现：**

部分同学运行仿真时，TCL控制台输出信息包含：“Error C and C\_ref not equal”。其结果如下图所示：  
 

**解决办法：**

如果出现类似错误或者TCL没有输出比对信息的同学可以使用最新发布的实验包。

如果没有出现类似的同学使用原来的实验发布包也可以。但是也建议其他同学使用最新发布的实验包：cache\_lab\_v0.06；重新进行仿真，即可输出DEBUG信息。

**错误原因：**

1. 同学们代码逻辑错误：cache中写逻辑错误，即store类指令错误。
2. 实验环境有问题，导致没有输出DEBUG信息：我们实验的调试机制为Trace比对机制。思想是关注哪些会修改通用寄存器的指令的执行情况。我们可以通过这种方法检测lw指令是否执行正确，但是我们无法直接的检测sw指令是否执行正确。我们往往是通过检测lw指令的执行情况来判断sw指令是否执行正确的。

举个例子： sw 0x99, addr1;

……

lw t1, addr1;

如果sw指令执行错误，在这种调试机制下，我们是不能立即检测出sw指令执行错误的。只有当我们执行到lw指令，通过读取addr1地址的内存数据到通用寄存器的时候，测试程序才会去比对trace文件。虽然这个时候表现为lw指令错误，其实是sw指令错误。所以同学们在调试过程中，如果遇到前面的lw指令执行正确，但是突然有一条lw指令执行错误的情况。可以考虑是否是sw指令实现错误。这种情况不仅仅出现在cache的实现中。

（不了解不影响实验进行）那为什么没有输出比对信息呢？我们的测试程序主要用于测试性能，会统计运行的时钟数（通过记录开始时间跟结束时间，相关代码可以在主控函数中看见： ./shell1.c）。所以我们会使用lw指令去读时钟（外设）。

但是在这种情况下，不同环境读取的开始时间和结束时间可能是不一样的，所以lw指令写入到通用寄存器的值是不一样的，所以用于读取时钟的lw指令没有一个标准的执行结果。在这种情况下，我们可以通过在记录时钟的时候关闭trace比对机制，来避免对读取时钟的lw指令进行对比。

导致出现没有输出比对信息原因是：

我们只检测了dgemm函数，也就是进行分块矩阵运算的函数，并未检测之后的和C\_ref矩阵比对的代码。这导致如果写入C矩阵时(sw指令)发生了错误，是没有进行trace检测的。因此我们更新了发布包，新的发布包会检测整个matrix\_mult函数。

