深圳大学实验报告

课程名称：计算机系统(3)

实验项目名称：处理器结构实验 二

学院：计算机与软件学院

专业：计算机科学与技术

指导教师：罗秋明

报告人：林浩晟 学号：2022280310 班级：01

实验时间：2024 年 11 月 8 日

实验报告提交时间：2024 年 11 月 15 日

教务处制

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。
2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。

一、试验目的

——控制冒险与分支预测

了解控制冒险分支预测的概念

了解多种分支预测的方法，动态分支预测更要深入了解

理解什么是 BTB（Branch Target Buffer），并且学会用 BTB 来优化所给程序

利用 BTB 的特点，设计并了解在哪种状态下 BTB 无效

了解循环展开，并于 BTB 功能进行对比

对 WinMIPS64 的各个窗口和操作更加熟悉

二、实验内容

按照下面的实验步骤及说明，完成相关操作记录实验过程的截图：

首先，给出一段矩阵乘法的代码，通过开启 BTB 功能对其进行优化，并且观察流水线的细节，解释 BTB 在其中所起的作用；

其次，自行设计一段使得即使开启了 BTB 也无效的代码。

第三，使用循环展开的方法，观察流水因分支停顿的次数减少的现象，并对比采用 BTB 结构时流水因分支而停顿的次数。

（选做：在 x86 系统上编写 C 语言的矩阵乘法代码，用 perf 观察分支预测失败次数，分析其次数是否与你所学知识吻合。再编写前面第二部使用的令分支预测失败的代码，验证 x86 是否能正确预测，并尝试做解释）

三、实验环境

硬件：桌面 PC

软件：Windows

四、实验步骤及说明

背景知识

在遇到跳转语句的时候，我们往往需要等到 MEM 阶段才能确定这条指令是否跳转（通过硬件的优化，可以极大的缩短分支的延迟，将分支执行提前到 ID 阶段，这样就能够将分支预测错误代价减小到只有一条指令），这种为了确保预取正确指令而导致的延迟叫控制冒险（分支冒险）。

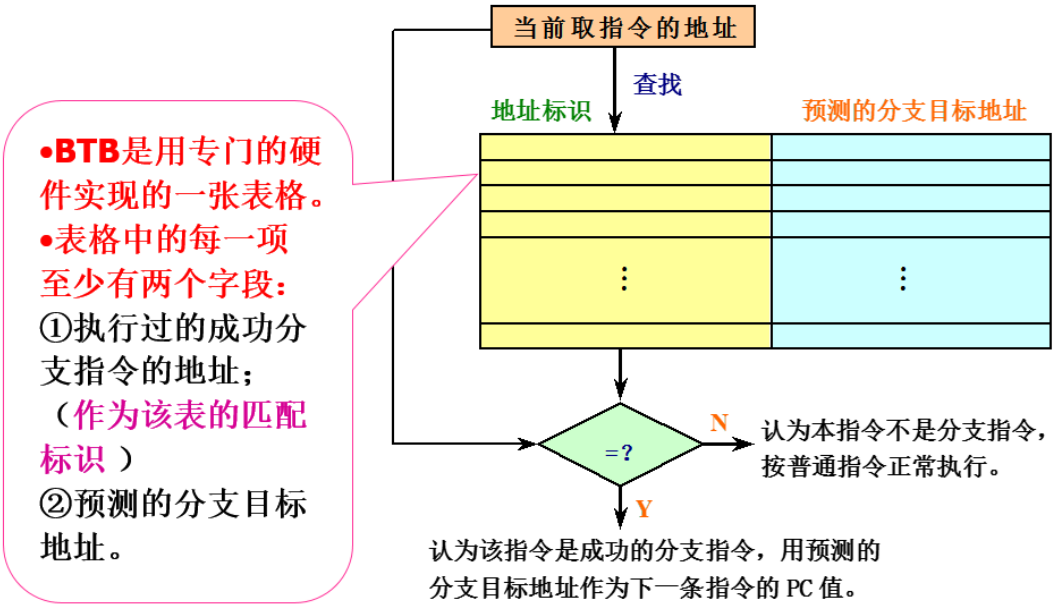
为了降低控制冒险所带来的性能损失，一般采用分支预测技术。分支预测技术包含编译时进行的静态分支预测，和执行时进行的动态分支预测。这里，我们着重介绍动态分支预测中的 BTB（Branch Target Buffer）技术。

BTB 即为分支目标缓冲器，它将分支指令（对应的指令地址）放到一个缓冲区中保存起来，当下次再遇到相同的指令（跳转判定）时，它将执行和上次一样的跳转（分支或不分支）预测。

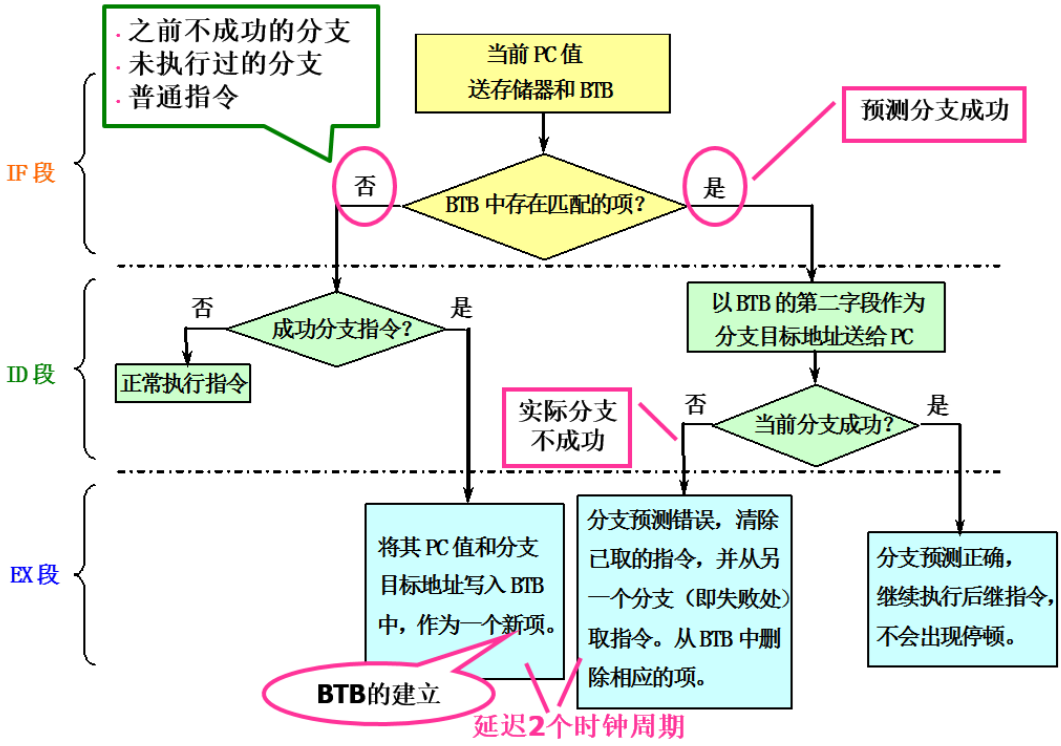
一种可行的 BTB 结构示意图如下：

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。



在采用了 BTB 之后，在流水线各个阶段所进行的相关操作如下：



注意，为了填写 BTB，需要额外一个周期。

一、矩阵乘法及优化

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。
2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。

在这一阶段，我们首先给出矩阵乘法的例子，接着将流水线设置为不带 BTB 功能（configure->enable branch target buffer）直接运行，观察结果进行记录；然后，再开启 BTB 功能再次运行，观察实验结果。将两次的实验结果进行对比，观察 BTB 是否起作用，如果有效果则进一步观察流水线执行细节并且解释 BTB 起作用原因。

矩阵乘法的代码如下：

```
        .data
str:  .asciiz "the data of matrix 3:\n"
mx1:   .space 512
mx2:   .space 512
mx3:   .space 512

            .text
initial:    daddi r22,r0,mx1  #这个initial模块是给三个矩阵赋初值
            daddi r23,r0,mx2
            daddi r21,r0,mx3
input:      daddi r9,r0,64
            daddi r8,r0,0
loop1:      dsll r11,r8,3
            dadd r10,r11,r22
            dadd r11,r11,r23
            daddi r12,r0,2
            daddi r13,r0,3
            sd r12,0(r10)
            sd r13,0(r11)

            daddi r8,r8,1
            slt r10,r8,r9
            bne r10,r0,loop1

mul:        daddi r16,r0,8
            daddi r17,r0,0
loop2:      daddi r18,r0,0    #这个循环是执行for(int i = 0, i < 8; i++)的内容
loop3:      daddi r19,r0,0    #这个循环是执行for(int j = 0, j < 8; j++)的内容
            daddi r20,r0,0    #r20存储在计算result[i][j]过程中每个乘法结果的叠加值
loop4:      dsll r8,r17,6     #这个循环的执行计算每个result[i][j]
            dsll r9,r19,3
            dadd r8,r8,r9
            dadd r8,r8,r22
            ld r10,0(r8)      #取mx1[i][k]的值
            dsll r8,r19,6
            dsll r9,r18,3
            dadd r8,r8,r9
```

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。

```
dadd r8,r8,r23
ld r11,0(r8)       #取mx2[k][j]的值
dmul r13,r10,r11   #mx1[i][k]与mx2[k][j]相乘
dadd r20,r20,r13   #中间结果累加

daddi r19,r19,1
slt r8,r19,r16
bne r8,r0,loop4

dsll r8,r17,6
dsll r9,r18,3
dadd r8,r8,r9
dadd r8,r8,r21     #计算result[i][j]的位置
sd r20,0(r8)       #将结果存入result[i][j]中

daddi r18,r18,1
slt r8,r18,r16
bne r8,r0,loop3

daddi r17,r17,1
slt r8,r17,r16
bne r8,r0,loop2

halt
```

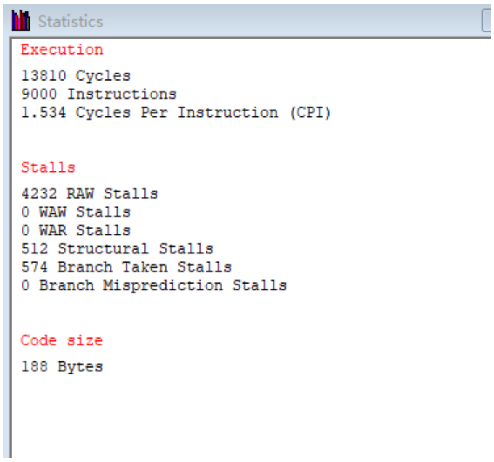
不设置 BTB 功能，运行该程序，观察 Statistics 窗口的结果截屏并记录下来。

1. 将上述代码存入matrix.s文件中,用asm.exe检查并导入WinMIPS64,运行后观察到发生了574次Branch Taken Stalls;

```
PS D:\WinMips\winmips64> ./asm.exe matrix.s
Pass 1 completed with 0 errors
00000000                        .data
00000000          str:  .asciiz "the data of matrix 3:\n"
         74686520
         64617461
         206f6620
         6d617472
         69782033
         3a0a00
00000018          mx1:   .space 512
00000218          mx2:   .space 512
00000418          mx3:   .space 512
```

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。
2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。

```
Pass 2 completed with 0 errors
Code Symbol Table
             initial = 00000000
               input = 0000000c
               loop1 = 00000014
                 mul = 0000003c
               loop2 = 00000044
               loop3 = 00000048
               loop4 = 00000050
Data Symbol Table
                 str = 00000000
                 mx1 = 00000018
                 mx2 = 00000218
                 mx3 = 00000418
```



2. 分析574次的Branch Taken Stalls(BTS)：
   (1) 初始化部分：分别给三个8*8的矩阵赋初值，因为WinMIPS64默认预测不跳转，则仅最后一次循环预测正确，发生8*8－1=63次BTS；
   (2) 矩阵乘法部分：
      ① 第一层循环，执行1轮8次，最后一次循环预测正确，发生1*7=7次BTS；
      ② 第二层循环，执行8轮8次，每轮仅最后一次循环预测正确，发生8*7=56次BTS；
      ③ 第三层循环，执行8*8=64轮8次，每轮仅最后一次循环预测正确，发生8*8*7=448次BTS。
   (3) 总次数为63+7+56+448=574，与程序输出的一致。

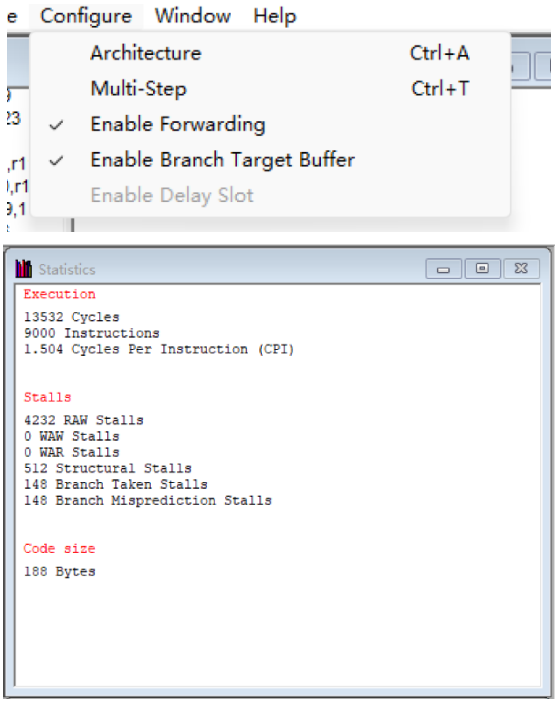
接着，设置 BTB 功能（在菜单栏处选择 Configure 项，然后在下拉菜单中为 Enable Branch Target Buffer 选项划上钩）。并在此运行程序，观察 Statistics 窗口的结果并截屏记录下来。

在这里，我们仅仅观察比较 Stalls 中的最后两项------Branch Taken Stalls 和 Branch Misprediction Stalls。

接下来，对比其结果。我们就结合流水线执行细节分析造成这种情况发生的原因。

3. 开启 BTB 功能后再次执行程序，可以得到下图，可以看到发生了 148 次 Branch Taken Stalls(BTS)，同时也发生了 148 次 Branch Misprediction Stalls(BMS)；

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。
2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。



4. 对于每个循环，BTB 在第一次进入循环时存储跳转地址（BTS），在循环结束时清除（BMS）。因此，循环结构中的 BTS 次数和 BMS 次数会相等。

分析 148 次 BTS 和 BMS：

(1) 初始化部分：一层循环，只发生 1 次 BTS 和 1 次 BMS；

(2) 矩阵乘法部分：

① 第三层循环，执行 8*8=64 轮，每轮发生 1 次 BTS 和 1 次 BMS，共计各 64 次；

② 第二层循环，执行 8 轮，每轮发生 1 次 BTS 和 1 次 BMS，共计各 8 次；

③ 第三层循环，执行 1 轮，发生 1 次 BTS 和 1 次 BMS，共计各 1 次；

(3) 总次数为(1+64+8+1)*2=148 (其中*2 是因为流水线阻塞一次会推迟 2 个时钟周期)，与程序输出一致；

（30 分，结果的获取 10 分，细节分析 20 分）

二、设计使 BTB 无效的代码

在这个部分，我们要设计一段代码，这段代码包含了一个循环。根据 BTB 的特性，我们设计的这个代码将使得 BTB 的开启起不到相应的优化作用，反而会是的性能大大降低。

提示：一定要利用 BTB 的特性，即它的跳转判定是根据之前跳转成功与否来决定的。

给出所用代码以及设计思路，给出运行结果的截屏证明代码实现了目标。

（30 分，代码及思路 20，获取结果并证明目标实现 10 分）

为了降低启用 BTB 后的程序性能，可以通过确保每次分支预测都失败来实现，即让循环中的跳转每次都不一致。

在“一、矩阵乘法及其优化”部分的讨论中，分支预测失效的主要原因是矩阵乘法操作。为了达到每次分支预测都失败的效果，可以通过调整矩阵的尺寸来实现，即将矩阵大小设置为 2x2。

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。

此外，还可以通过将单层循环结构改为双层循环结构来增加分支预测失效的次数。这样做可以进一步提高分支预测错误率，从而降低程序的整体性能。

由此可以给出如下代码

```
.data
str: .asciiz "the data of matrix 3:\n"
mx1: .space 32
mx2: .space 32
mx3: .space 32

    .text
initial:
    daddi r22, r0, mx1
    daddi r23, r0, mx2
    daddi r21, r0, mx3
input:
    daddi r9, r0, 2
    daddi r17, r0, 0
iloop1:
    daddi r18, r0, 0
iloop2:
    dsll r10, r17, 4
    dsll r11, r18, 3
    dadd r11, r11, r10
    dadd r10, r11, r22
    dadd r11, r11, r23
    daddi r12, r0, 2
    daddi r13, r0, 3
    sd r12, 0(r10)
    sd r13, 0(r11)
    daddi r18, r18, 1
    slt r10, r18, r9
    bne r10, r0, iloop2
    daddi r17, r17, 1
    slt r10, r17, r9
    bne r10, r0, iloop1
mul:
    daddi r16, r0, 2
    daddi r17, r0, 0
#循环
loop1:
    daddi r18, r0, 0
loop2:
    daddi r19, r0, 0
```

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。

```
        daddi r20, r0, 0
loop3:
        dsll r8, r17, 4
        dsll r9, r19, 3
        dadd r8, r8, r9
        dadd r8, r8, r22
        ld r10, 0(r8)
        dsll r8, r19, 4
        dsll r9, r18, 3
        dadd r8, r8, r9
        dadd r8, r8, r23
        ld r11, 0(r8)
        dmul r13, r10, r1
        dadd r20, r20, r13
        daddi r19, r19, 1
        slt r8, r19, r16
        bne r8, r0, loop3
        dsll r8, r17, 4
        dsll r9, r18, 3
        dadd r8, r8, r9
        dadd r8, r8, r21
        sd r20, 0(r8)
        daddi r18, r18, 1
        slt r8, r18, r16
        bne r8, r0, loop2
        daddi r17, r17, 1
        slt r8, r17, r16
        bne r8, r0, loop1
        halt
```

1. 将代码用 asm.exe 进行检查，可以发现代码正确；

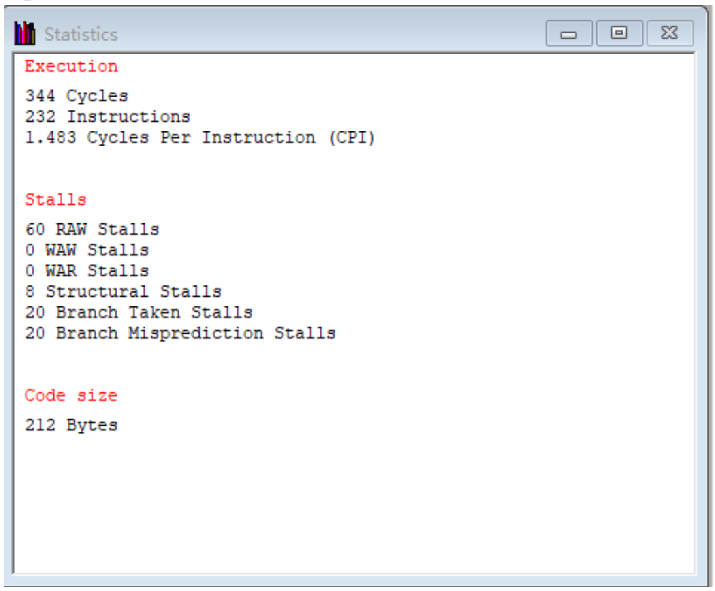
```
Pass 1 completed with 0 errors
00000000          .data
00000000          str: .asciiz "the data of matrix 3:\n"
         74686520
         64617461
         206f6620
         6d617472
         69782033
         3a0a00
00000018          mx1: .space 32
00000038          mx2: .space 32
00000058          mx3: .space 32
```

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。

```
Pass 2 completed with 0 errors
Code Symbol Table
              initial = 00000000
                input = 0000000c
               iloop1 = 00000014
               iloop2 = 00000018
                  mul = 00000054
                loop1 = 0000005c
                loop2 = 00000060
                loop3 = 00000068
Data Symbol Table
                  str = 00000000
                  mx1 = 00000018
                  mx2 = 00000038
                  mx3 = 00000058
```

2. 载入 WinMips 中，开启 BTB 运行，发现发生了 20 次 BTS 和 20 次 BMS；



分析 20 次的 BTS：

(1) 赋值进行了 3 次；

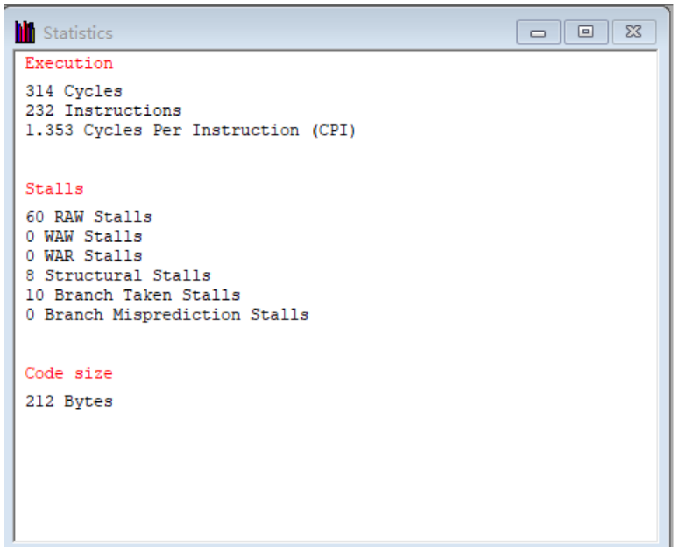
(2) 矩阵中的乘法进行 7 次；

(3) 总次数为(3 + 7)*2=20 次，与输出一致。

3. 关闭 BTB 后运行，发现只进行了 10 次的 BTS，性能提升了一倍。

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。



10 次 BTS 的计算：

(1) 赋值有 3 次；

(2) 矩阵的乘法进行了 7 次

总次数为 3 + 7=10 次。

4. 因此我们知道启用分支目标缓冲（BTB）并不总是能够提升程序性能，其效果很大程度上取决于程序的具体实现方式，尤其是循环的深度和每次循环的迭代次数。

三、循环展开与 BTB 的效果比对

首先，我们需要对循环展开这个概念有一定的了解。

什么是循环展开呢？所谓循环展开就是通过在每次迭代中执行更多的数据操作来减小循环开销的影响。其基本思想是设法把操作对象线性化，并且在一次迭代中访问线性数据中的一个小组而非单独的某个。这样得到的程序将执行更少的迭代次数，于是循环开销就被有效地降低了。

接下来，我们就按照这种思想对上述的矩阵乘法程序进行循环展开。要求将上述的代码通过循环展开将最里面的一个执行迭代 8 次的循环整个展开了，也就是说，我们将矩阵相乘的三个循环通过代码的增加，减少到了两个循环。

比较，通过对比循环展开（未启用BTB）、使用BTB（未进行循环展开）以及未使用BTB且未作循环展开的运行结果。比较他们的Branch Tanken Stalls和Branch Misprediction Stalls的数量，并尝试给出评判。

（30 分，循环展开代码及思路 20 分，评判 10 分）

将（一）中的matrix代码改为8层循环展开，如下

```
.data
str: .asciiz "the data of matrix 3:\n"
mx1: .space 512
mx2: .space 512
mx3: .space 512
     .text
```

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。

```
initial:
    daddi r22, r0, mx1
    daddi r23, r0, mx2
    daddi r21, r0, mx3
input:
    daddi r9, r0, 64
    daddi r8, r0, 0
loop1:
    dsll r11, r8, 3
    dadd r10, r11, r22
    dadd r11, r11, r23
    daddi r12, r0, 2
    daddi r13, r0, 3
    sd r12, 0(r10)
    sd r13, 0(r11)
    daddi r8, r8, 1
    slt r10, r8, r9
    bne r10, r0, loop1
mul:
    daddi r16, r0, 8
    daddi r17, r0, 0
loop2:
    daddi r18, r0, 0
loop3:
    daddi r19, r0, 0
    daddi r20, r0, 0
    #0
    dsll r8, r17, 6
    dsll r9, r19, 3
    dadd r8, r8, r9
    dadd r8, r8, r22
    ld r10, 0(r8)
    dsll r8, r19, 6
    dsll r9, r18, 3
    dadd r8, r8, r9
    dadd r8, r8, r23
    ld r11, 0(r8)
    dmul r13, r10, r11
    dadd r20, r20, r13
    daddi r19, r19, 1
    #1
    dsll r8, r17, 6
    dsll r9, r19, 3
```

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。
2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。

```
dadd r8, r8, r9
dadd r8, r8, r22
ld r10, 0(r8)
dsll r8, r19, 6
dsll r9, r18, 3
dadd r8, r8, r9
dadd r8, r8, r23
ld r11, 0(r8)
dmul r13, r10, r11
dadd r20, r20, r13
daddi r19, r19, 1
#2
dsll r8, r17, 6
dsll r9, r19, 3
dadd r8, r8, r9
dadd r8, r8, r22
ld r10, 0(r8)
dsll r8, r19, 6
dsll r9, r18, 3
dadd r8, r8, r9
dadd r8, r8, r23
ld r11, 0(r8)
dmul r13, r10, r11
dadd r20, r20, r13
daddi r19, r19, 1
#3
dsll r8, r17, 6
dsll r9, r19, 3
dadd r8, r8, r9
dadd r8, r8, r22
ld r10, 0(r8)
dsll r8, r19, 6
dsll r9, r18, 3
dadd r8, r8, r9
dadd r8, r8, r23
ld r11, 0(r8)
dmul r13, r10, r11
dadd r20, r20, r13
daddi r19, r19, 1
#4
dsll r8, r17, 6
dsll r9, r19, 3
dadd r8, r8, r9
```

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。
2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。

```
dadd r8, r8, r22
ld r10, 0(r8)
dsll r8, r19, 6
dsll r9, r18, 3
dadd r8, r8, r9
dadd r8, r8, r23
ld r11, 0(r8)
dmul r13, r10, r11
dadd r20, r20, r13
daddi r19, r19, 1
#5
dsll r8, r17, 6
dsll r9, r19, 3
dadd r8, r8, r9
dadd r8, r8, r22
ld r10, 0(r8)
dsll r8, r19, 6
dsll r9, r18, 3
dadd r8, r8, r9
dadd r8, r8, r23
ld r11, 0(r8)
dmul r13, r10, r11
dadd r20, r20, r13
daddi r19, r19, 1
#6
dsll r8, r17, 6
dsll r9, r19, 3
dadd r8, r8, r9
dadd r8, r8, r22
ld r10, 0(r8)
dsll r8, r19, 6
dsll r9, r18, 3
dadd r8, r8, r9
dadd r8, r8, r23
ld r11, 0(r8)
dmul r13, r10, r11
dadd r20, r20, r13
daddi r19, r19, 1
#7
dsll r8, r17, 6
dsll r9, r19, 3
dadd r8, r8, r9
dadd r8, r8, r22
```

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。
2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。

```
ld r10, 0(r8)
dsll r8, r19, 6
dsll r9, r18, 3
dadd r8, r8, r9
dadd r8, r8, r23
ld r11, 0(r8)
dmul r13, r10, r11
dadd r20, r20, r13
daddi r19, r19, 1



dsll r8, r17, 6
dsll r9, r18, 3
dadd r8, r8, r9
dadd r8, r8, r21
sd r20, 0(r8)

daddi r18, r18, 1
slt r8, r18, r16
bne r8, r0, loop3

daddi r17, r17, 1
slt r8, r17, r16
bne r8, r0, loop2

halt
```

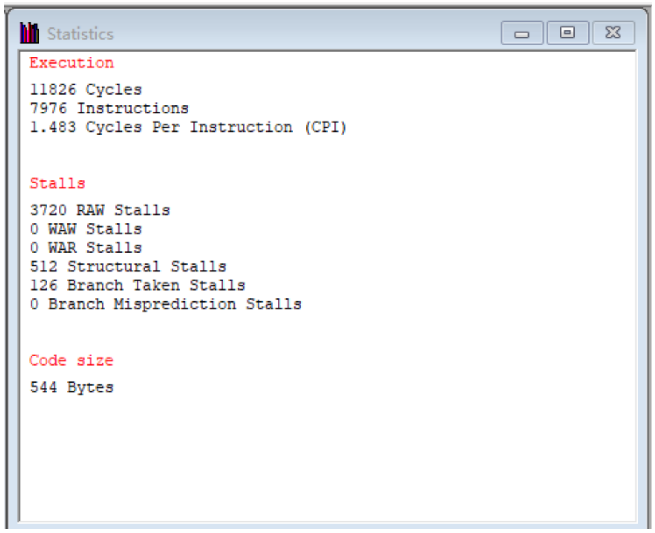
1. 将代码用asm.exe检查，发现没有错误；

```
Pass 1 completed with 0 errors
00000000           .data
00000000          str: .asciiz "the data of matrix 3:\n"
         74686520
         64617461
         206f6620
         6d617472
         69782033
         3a0a00
```

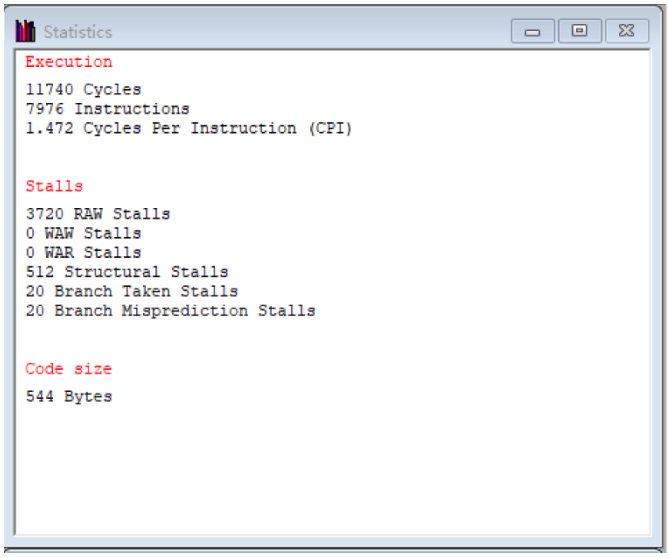
注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。
2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。

```
Pass 2 completed with 0 errors
Code Symbol Table
             initial = 00000000
               input = 0000000c
               loop1 = 00000014
                 mul = 0000003c
               loop2 = 00000044
               loop3 = 00000048
Data Symbol Table
                 str = 00000000
                 mx1 = 00000018
                 mx2 = 00000218
                 mx3 = 00000418
```

2. 载入WinMips中运行，关闭BTB，可以发现进行了126次的BTS；不难发现在进行循环展开之前，共发生了574次BTS，而在实施循环展开优化之后，BTS的次数减少了448次，这个减少的数量正好等于矩阵乘法操作中第三层循环所引发的BTS失效次数；



3. 打开BTB运行，可以发现进行了20次的BTS和20次的BMS；在循环展开之前，程序中出现了148次分支目标缓冲（BTS）错误。实施循环展开优化后，BTS错误的次数减少了120次，这个减少的数量与矩阵乘法中第三层循环产生的BTS错误次数相吻合。



注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。
2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。

4. 因此我们知道，当循环的迭代次数较多时，启用分支目标缓冲（BTB）可以提升程序的执行性能。然而，如果循环的层数过多，内层循环将频繁地出现BTS错误，这会导致性能降低。在这种情况下，可以通过展开那些迭代次数较少的循环来减少分支预测错误的次数，从而提高程序的性能。

四、结束语

写下对于这次试验的所得与感想。

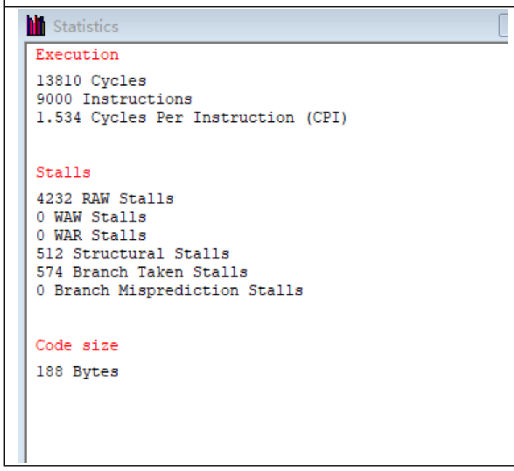
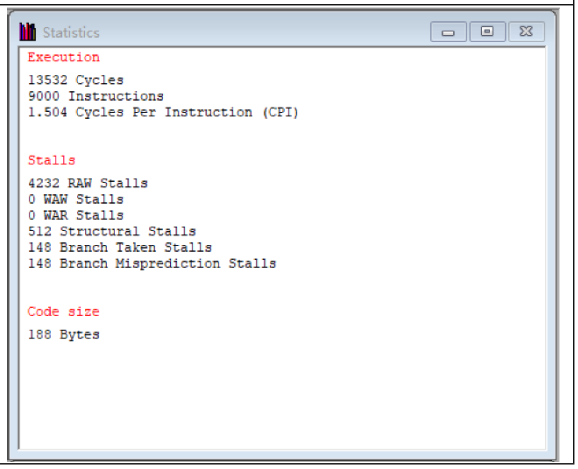
1. 通过这次实验，我对于BTB的作用更加的熟悉；BTB是一种硬件机制，用于预测程序中的分支指令（如if语句）的跳转目标，以减少因分支预测错误而产生的流水线停顿。在矩阵乘法代码中，BTB能够预测循环中的迭代次数，从而提前加载数据和计算指令，减少了因分支预测错误导致的流水线清空和重新填充的开销。
2. 这次实验让我深刻理解了现代处理器中流水线优化技术的重要性和复杂性。BTB和循环展开是提高程序性能的两种重要手段，它们在不同的场景下各有优势。
3. 同时我也意识到，优化技术的选择需要根据具体的程序特性和运行环境来决定。没有一种优化技术是万能的，理解各种技术的原理和适用场景对于编写高效代码至关重要。
4. 最后，这次实验也让我认识到了硬件和软件之间的紧密联系。硬件的优化特性需要软件层面的支持才能发挥最大效能，反之亦然。这种软硬结合的思维方式对于解决实际问题非常有帮助。

（报告撰写质量 10 分）

五、实验结果

实验成功完成。

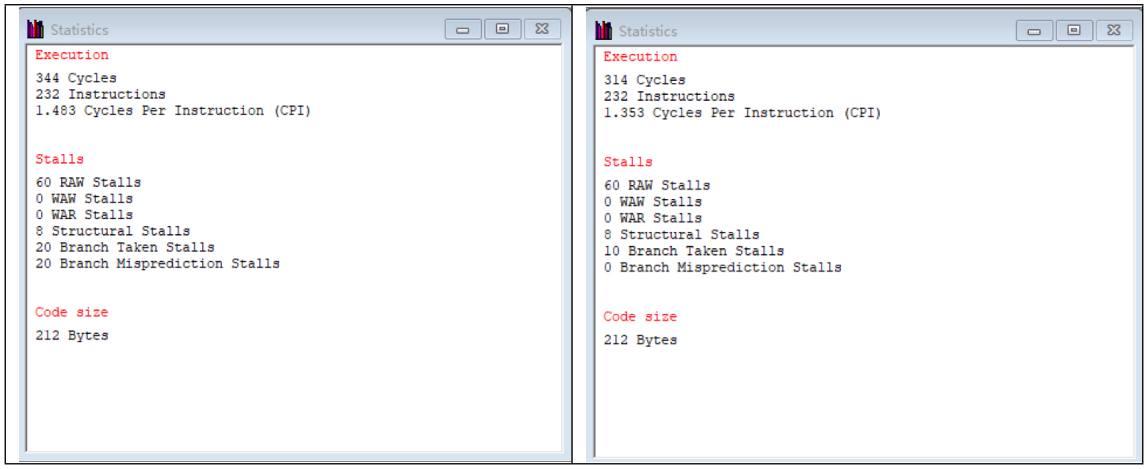
一、矩阵乘法及优化

关闭 BTB	开启 BTB
	

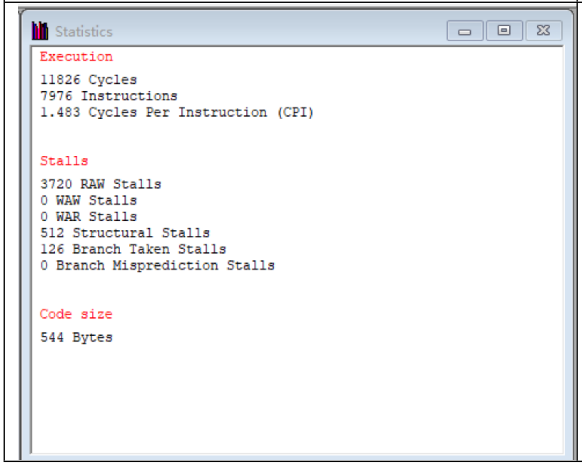
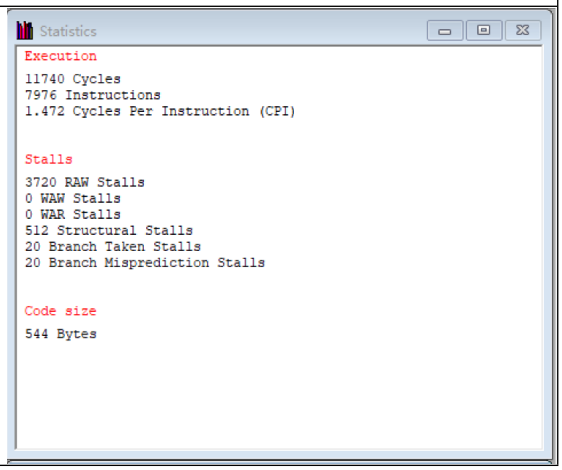
二、设计使 BTB 无效的代码

关闭 BTB	开启 BTB

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。
2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。



三、循环展开与 BTB 的效果比对

关闭 BTB	开启 BTB
	

六、实验总结与体会

在这次处理器结构实验中，我深入理解了控制冒险、分支预测和 BTB 的概念，并通过编程和流水线执行观察，增强了对这些知识的掌握。

1. 控制冒险与分支预测

实验中，我们首先体验了矩阵乘法代码中的控制冒险问题，认识到分支指令可能导致流水线停顿，影响执行效率。了解静态和动态分支预测后，特别是动态分支预测的工作机制，深刻体会到它对提升处理器性能的重要性。

2. BTB 的优化作用

启用 BTB 后，观察到 BTB 能有效提高分支预测的准确性，减少控制冒险对流水线的影响，进一步理解了 BTB 在处理器中的重要作用。

3. 设计 BTB 无效的代码

通过设计 BTB 无效的代码段，我发现 BTB 的性能优势在某些情况下会被削弱，这让我认识到处理器设计中应考虑到各种特殊情况。

4. 循环展开与性能对比

在对比循环展开和 BTB 性能时，理解了循环展开能有效减少分支停顿，从而提升执行效率，学习到优化代码时需综合考虑多种手段。

5. 熟悉 WinMIPS64

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。
2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。

在使用 WinMIPS64 过程中，我对其操作有了深入理解，这一模拟器使得理论知识更具实践性和直观性。

这次实验让我对处理器结构中的控制冒险、分支预测和 BTB 有了更深刻的认识，同时提高了我在处理器设计和性能优化方面的实际应用能力。

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。

指导教师批阅意见： **成绩评定：** 指导教师签字： 年月日
备注：

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。
2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。