

实验五：Branch-Target Buffers

19281171 王雨潇

实验报告要求：实验报告用 Word 排版，文件内一定要有"学号姓名_学号"。实验报告应提交到课程平台，按时提交。实验报告的内容应尽量全面，避免简单化。简述实验内容，实验过程，实验结果，实验分析，心得体会等。答每道题前应将题目拷贝到实验报告上。

一、实验目的

加深对 Branch-Target Buffers 的理解，掌握使用 Branch-Target Buffers 减少或增加分支带来的延迟的情况。

二、实验过程

1. 将以下程序段修改为可利用 WinMIPS64 模拟器运行的程序,假设 R3 的初始值为 R2+40。

程序段 3			
Loop:	LW	R1,0(R2);	load R1 from address 0+R2
	ADDI	R1,R1,#1;	R1=R1+1
	SW	0(R2),R1;	store R1 at address 0+R2
	ADDI	R2,R2,#4;	R2=R2+4
	SUB	R4,R3,R2;	R4=R3-R2
	BNEZ	R4,Loop;	Branch to loop if R4!=0

答：仿照此前实验的代码，为程序段 3 添加数据段，初始化用到的寄存器；

```
.data
.text
daddi r1, $zero, 0    # r1 = 0
daddi r2, $zero, 0    # r2 = 0
daddi r3, r2, 40      # r3 = 40
daddi r4, $zero, 0    # r4 = 0

Loop:
lw     r1, 0(r2)       # load r1 from address 0+r2
daddi  r1, r1, 1       # r1=r1+1
sw     0(r2), r1       # store r1 at address 0+r2
```

```

daddi    r2, r2, 4      # r2=r2+4
dsub     r4, r3, r2     # r4=r3-r2
bnez     r4, Loop       # branch to loop if r4!=0

Finish:
halt

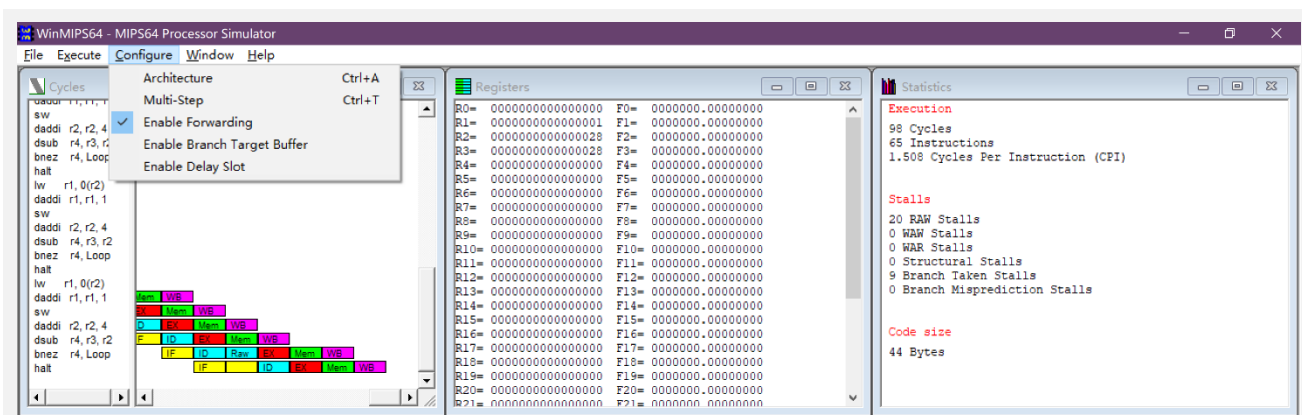
```

2. 在使用 forwarding 的情况下，对比采用 BTB 与不采用 BTB 技术时流水线的变化。

答：流水线对比如下表所示

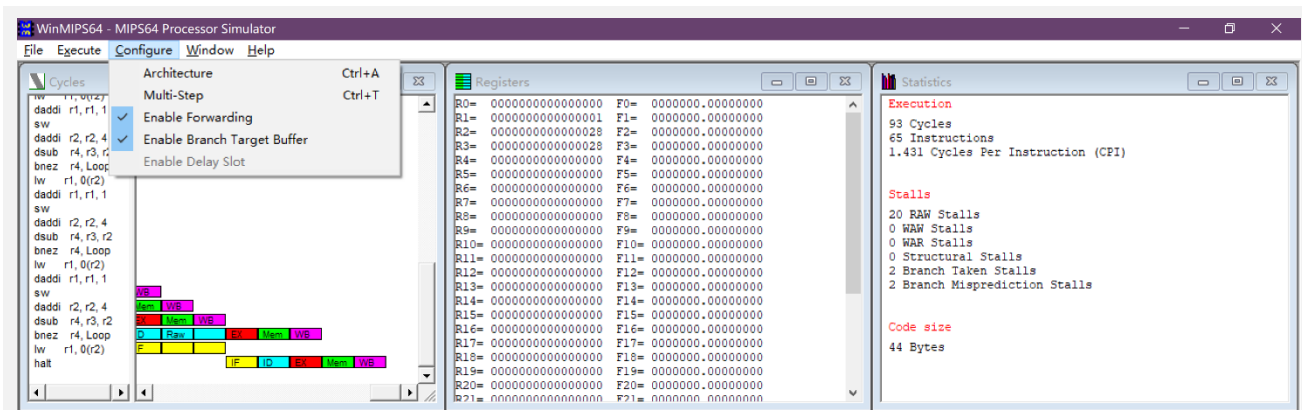
使用 forwarding，不采用 BTB 技术：

共执行 98 个周期，CPI 为 1.508，共有 20 个 RAW 暂停，9 个分支转移暂停；



使用 forwarding，采用 BTB 技术：

共执行 93 个周期，CPI 为 1.431，共有 20 个 RAW 暂停，2 个分支转移暂停，2 个分支预测错误暂停；

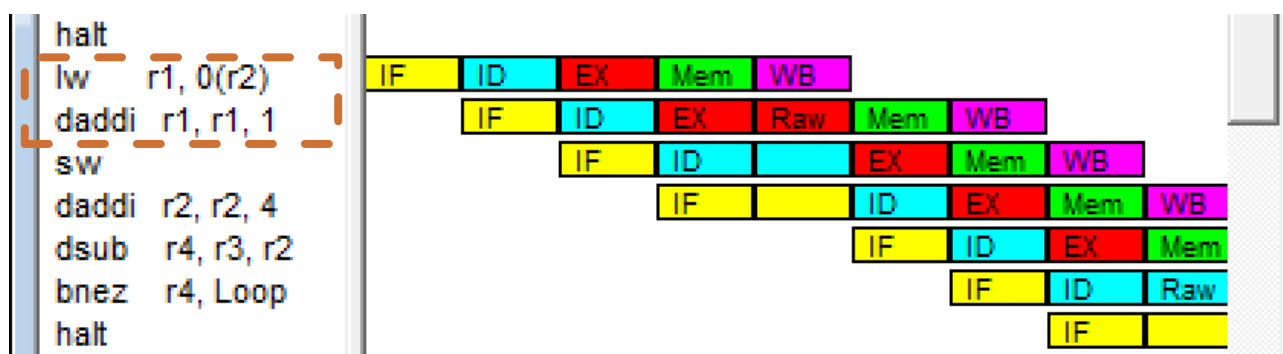


3. 重点分析两种情况下每次循环的 stall 周期数，都是由什么原因造成的？重点分析与分支指令相关的 stall。

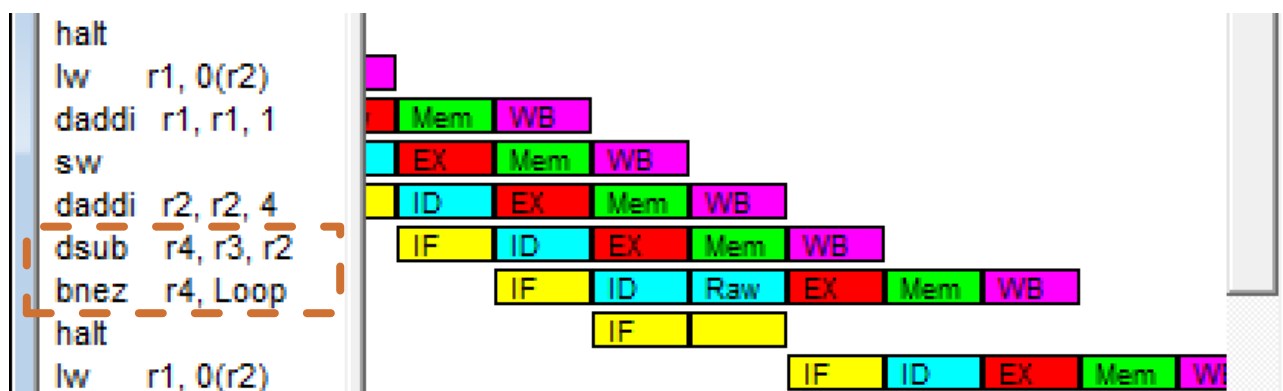
答：不采用 BTB 技术时的暂停原因如下

(1) 每次循环会发生 2 次 **RAW 暂停**

第 1 次是 `lw r1, 0(r2)` 和 `daddi r1, r1, 1` 之间的 RAW 相关引起的 (`lw` 在 MEM 阶段才写完寄存器 `r1`, `daddi` 要在 EX 阶段读寄存器 `r1`)



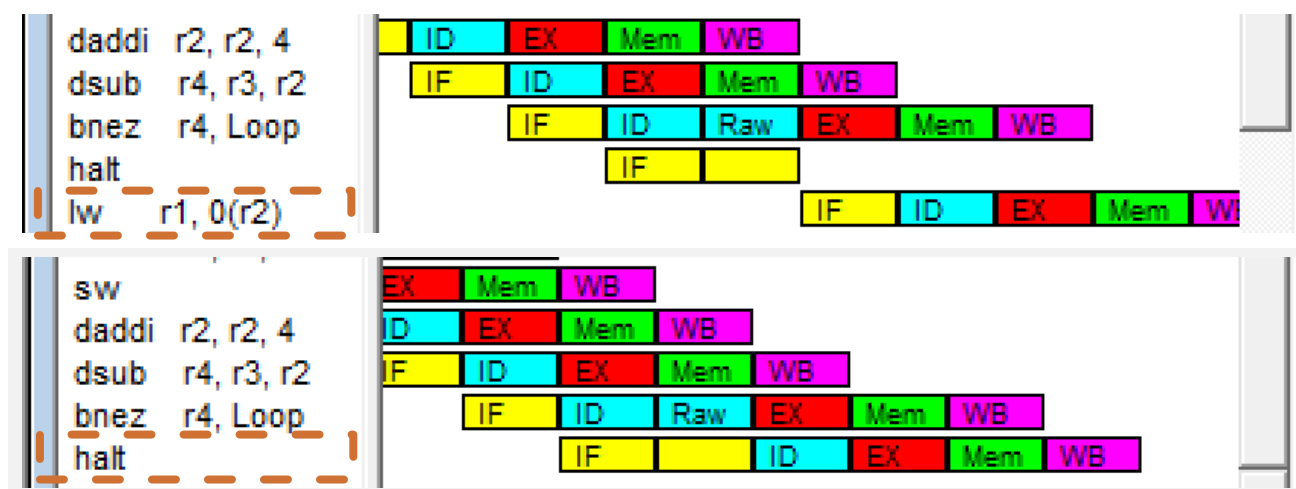
第 2 次是 `dsub r4, r3, r2` 和 `bnez r4, Loop` 之间的 RAW 相关引起的 (`dsub` 要在 MEM 阶段写完 `r4`, `benz` 要读 `r4` 才能在 ID 阶段提前算跳转地址)



(2) 前 9 次循环会发生 1 个周期**分支转移暂停**

WinMIPS64 采用预测分支转移失败降低流水线中的转移造成的惩罚,在 `bnez r4, Loop` 这个分支指令中，转移失败则执行 `halt`。

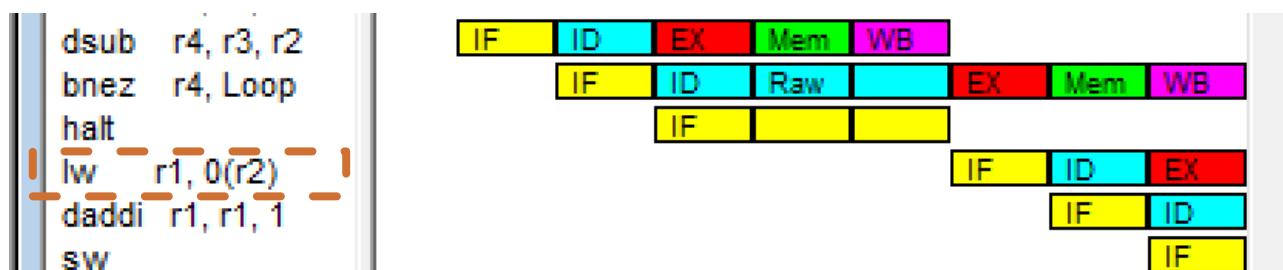
从时空图可以看出，前 9 次 bnez 预判错误让 lw 的 IF 阶段推迟了 1 个周期，第 10 次 bnez 预测成功没有暂停。



而采用 BTB 技术时的暂停原因如下

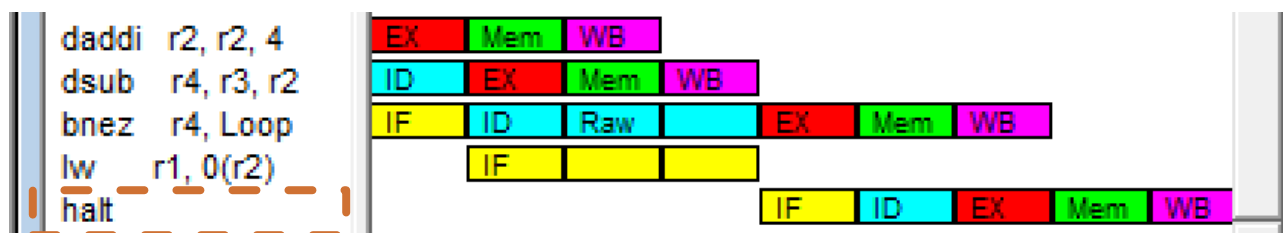
- (1) 每次循环会发生 2 次 **RAW 暂停**，原因与不采用 BTB 技术时一样；
- (2) 第 1 次循环发生了 2 个周期的**分支转移暂停**

此时 BTB 中没有转移的历史记录，也就无从预测第 1 次分支转移；



- (3) 最后 1 次循环发生了 2 个周期的**分支预测错误暂停**

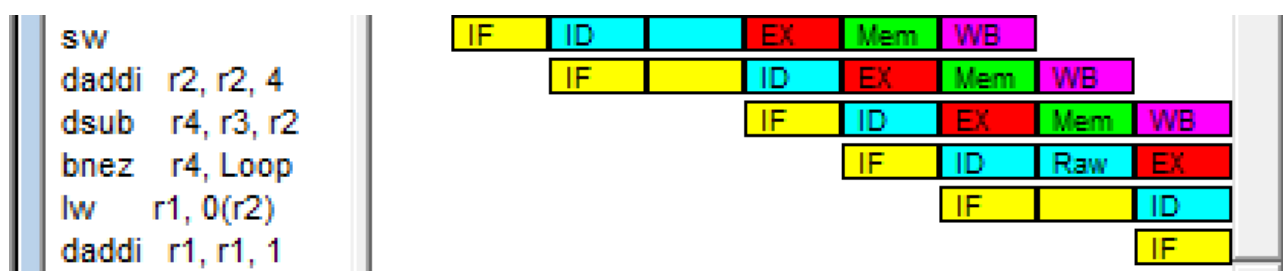
此前 BTB 存储的是预测转移成功的转移记录，但第 10 次 bnez r4, Loop 转移指令在 ID 段得知转移结果是失败，预测错误，所以产生了 2 个周期的错误预测惩罚；



4. 采用 BTB 技术何时能够减少分支指令带来的暂停？何时会增加暂停？为什么？

答：BTB 技术的本质是存储**转移成功后下条指令的地址的缓存**，且只存储预测转移成功时的转移地址，如果 BTB 对分支指令的预测结果和真实转移情况相同就能减少暂停。

采用 BTB 技术后，每当遇到转移分支，如果预测结果正确，程序在 ID 阶段从 BTB 的存储记录里取出地址直接执行，不需要任何暂停（如下图所示）；如果预测结果错误，程序需要做 2 件事：重新取指、修改 BTB 删除这个预测入口，共耗费 2 个周期。



三、 心得体会

在本次实验中，我通过观察 WinMIPS64 模拟器的运行，加深了对 BTB（转移目标缓冲器）原理和执行效果的理解。BTB 通过缓存转移目标地址减少流水线中的转移造成的暂停，如果预测正确就能直接取缓存地址，不需要额外暂停，但 BTB 也需要有转移的历史记录才能进行预测，也无法避免“最后一次预测”的错误。