



第2次实验：分支预测

计算机系统结构

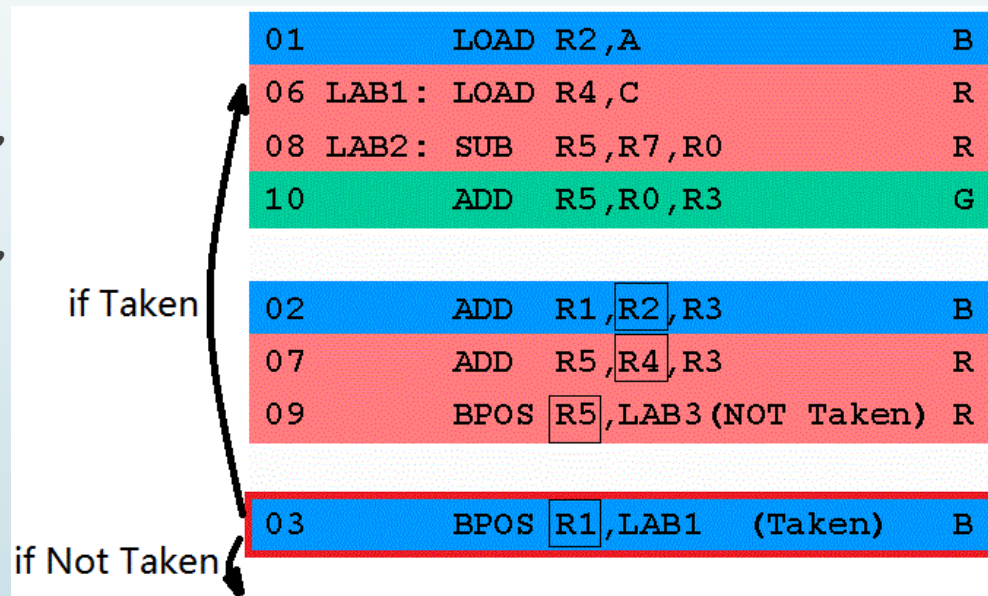
2021-2022春季学期

分支预测

- 在条件转移指令执行前，我们无法知道跳转到底会不会发生。如果能够事先预测跳转是否发生，处理器就可以到适当的地址取下一条指令，使得流水线不中断。在条件转移指令执行前预测跳转是否发生的任务，称为分支预测。

- 简单来说：

- bool
branch_prediction(instruction);





实验内容

- 实现一个分支预测算法，尽量降低测试程序运行时的 Mispredictions per thousand instructions (MPKI)。
- 修改模拟器中的分支预测模块，实现自己的分支预测算法。
- 实现的分支预测算法对存储的需求应小于32KB。

实验环境：模拟器

- 模拟器 (cbp2014.ver3.tar.gz) :
 - 在Linux下使用
- 解压：
 - `tar xzvf cbp2014.ver3.tar.gz`
- 目录结构：
 - results (模拟器运行结果, MPKI数据)
 - scripts (常用工具)
 - sim (模拟器源代码)
 - traces (测试程序, trace形式)

实验环境：分支预测模块

- 修改分支预测模块的代码：
 - sim/predictor.cc、sim/predictor.h
 - 即重新实现PREDICTOR类

```
class PREDICTOR{  
    // The state is defined for Gshare, change for your design  
  
private:  
    UINT32  ghr;           // global history register  
    UINT32  *pht;          // pattern history table  
    UINT32  historyLength; // history length  
    UINT32  numPhtEntries; // entries in pht  
  
public:  
    // The interface to the four functions below CAN NOT be changed  
  
    PREDICTOR(void);  
    bool    GetPrediction(UINT32 PC);  
    void    UpdatePredictor(UINT32 PC, bool resolveDir, bool predDir, UINT32 branchTarget);  
    void    TrackOtherInst(UINT32 PC, opType opType, UINT32 branchTarget);  
  
    // Contestants can define their own functions below  
};
```

实验环境：分支预测模块

► PREDICTOR类public部分：

- 定义了3个成员函数。这3个函数是PREDICTOR类与模拟器其它部分的接口。因此，这3个函数的声明不能修改，只能修改其内部的实现。
- 定义新的函数是可以的。但新定义的函数只能在PREDICTOR类内部使用，无法充当PREDICTOR类与模拟器其它部分的接口。

► PREDICTOR类private部分：

- 分支预测算法所需使用的数据结构。
- 可以自己定义。

实验环境：分支预测模块接口

```
bool    //返回值, true表示跳转, false表示不跳转  
GetPrediction( //预测条件转移指令是否发生跳转  
UINT32 PC    //待预测指令的指令地址  
);
```

注：本函数用于完成条件转移指令的分支预测。关于当前指令可用的信息只有指令地址，其它信息不可用于预测。

实验环境：分支预测模块接口

```
void  
UpdatePredictor( //每当执行条件转移指令时调用  
UINT32 PC, //当前指令的地址  
bool resolveDir, //当前指令实际是否跳转，true表示跳转  
bool predDir, //当前指令是否被预测为跳转，true表示跳转  
UINT32 branchTarget //跳转的目标地址  
);
```

注：本函数通常用于更新分支预测算法所维护的数据。

实验环境：分支预测模块接口

```
void  
TrackOtherInst( //每当执行非条件转移指令时调用  
    UINT32 PC, //当前指令的地址  
    OpType opType, //当前指令的操作码  
    UINT32 branchTarget //跳转的目标地址  
);
```

注：若opType显示当前指令不是转移指令，则branchTarget的取值没有意义。本函数用于跟踪非条件转移指令的执行情况。

实验环境：分支预测模块数据

- PREDICTOR类的private部分定义了分支预测算法所需维护的数据。
- 注意分支预测算法所维护数据的总大小不得超过32KB。
- 存储需求计算：
 - 计算的是用硬件实现相关数据结构所需的开销
 - 不一定等于C++实现中数据结构的大小
- 例：
 - 数据结构：unsigned data[10]，每个元素存储4种状态，用0-3表示。
 - 存储需求为： $2 * 10 = 20$ bits
 - 而不是： $32 * 10 = 320$ bits

实验环境：编译

- 编译：
 - `cd sim`
 - `make`
- 执行（直接运行编译出的可执行文件）：
 - `./predictor <trace_path>`
 - `<trace_path>`：trace所在路径，如`../traces/SHORT-INT-1.cbp4.gz`
 - 只运行参数指定的trace，打印MPKI到stdout。
- 实现完新的分支预测算法后，记得重新编译！

实验环境：执行

- 执行（自动执行所有trace）：
 - `cd scripts`
 - `./doit.sh`
- 查看执行结果：
 - `./getdata.pl -d ../results/GSHARE.32KB/`
 - 显示每个trace的MPKI，及其平均值。



提交内容

- predictor.cc
- predictor.h
- 实验报告

实验报告部分：

- 算法描述
- 存储需求
- 实验结果（MPKI）

- 注：实验成绩由算法对MPKI的优化程度和报告质量综合决定



外部链接

- Championship Branch Prediction (CBP-4):
 - <http://www.jilp.org/cbp2014/>
- ISCA 2014:
 - <http://cag.engr.uconn.edu/isca2014/>