国防科技大学计算机学院

计算机原理

COMPUTER PRINCIPLE

第四章 第四节（4）控制冒险的解决方法

- 有四种方法可以解决流水线控制冒险
  - ① 硬件阻塞（stall）
  - ② 软件插入“NOP”指令
  - ③ 分支预测
  - ④ 延迟分支

❑ 有四种方法可以解决流水线控制冒险

① 硬件阻塞（stall）

② 软件插入“NOP”指令

延迟分支后那条指令的执行，直到分支指令完成（产生新的PC值）

③ 分支预测

④ 延迟分支

1. 分支预测

- 简单（静态）预测

- 动态预测

1. 分支预测

- 简单（静态）预测
  - 总是预测条件不满足，即：继续执行分支指令的后续指令
  - 可使用启发式规则：在特定情况下总是预测转移成功（taken），其他情况则预测转移不成功。比如循环顶（底）部分支总是预测为转移不成功（成功）。能达到65%~85%的预测准确率
- 动态预测

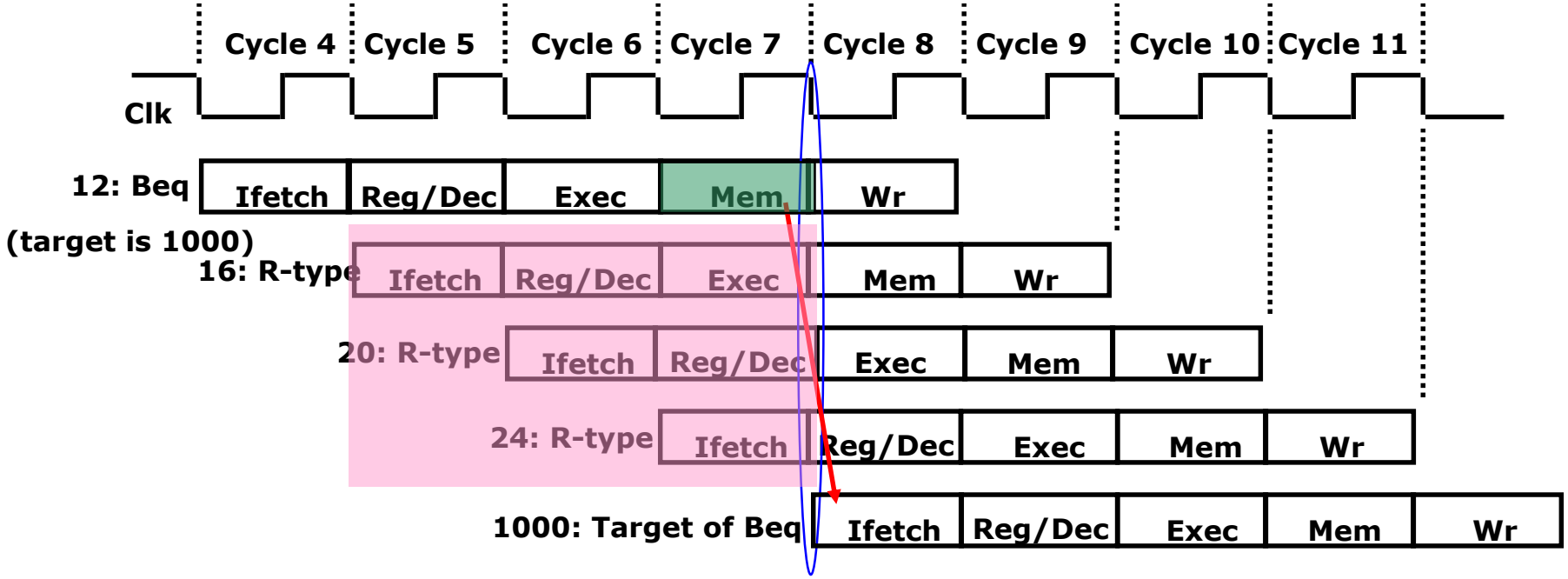
1. 分支预测

- 简单（静态）预测
  - 总是预测条件不满足，即：继续执行分支指令的后续指令
  - 可使用启发式规则：在特定情况下总是预测转移成功（taken），其他情况则预测转移不成功。比如循环顶（底）部分支总是预测为转移不成功（成功）。能达到65%~85%的预测准确率
- 动态预测
  - 根据程序执行的历史情况，进行动态预测调整，能达到90%的预测准确率

1. 分支预测

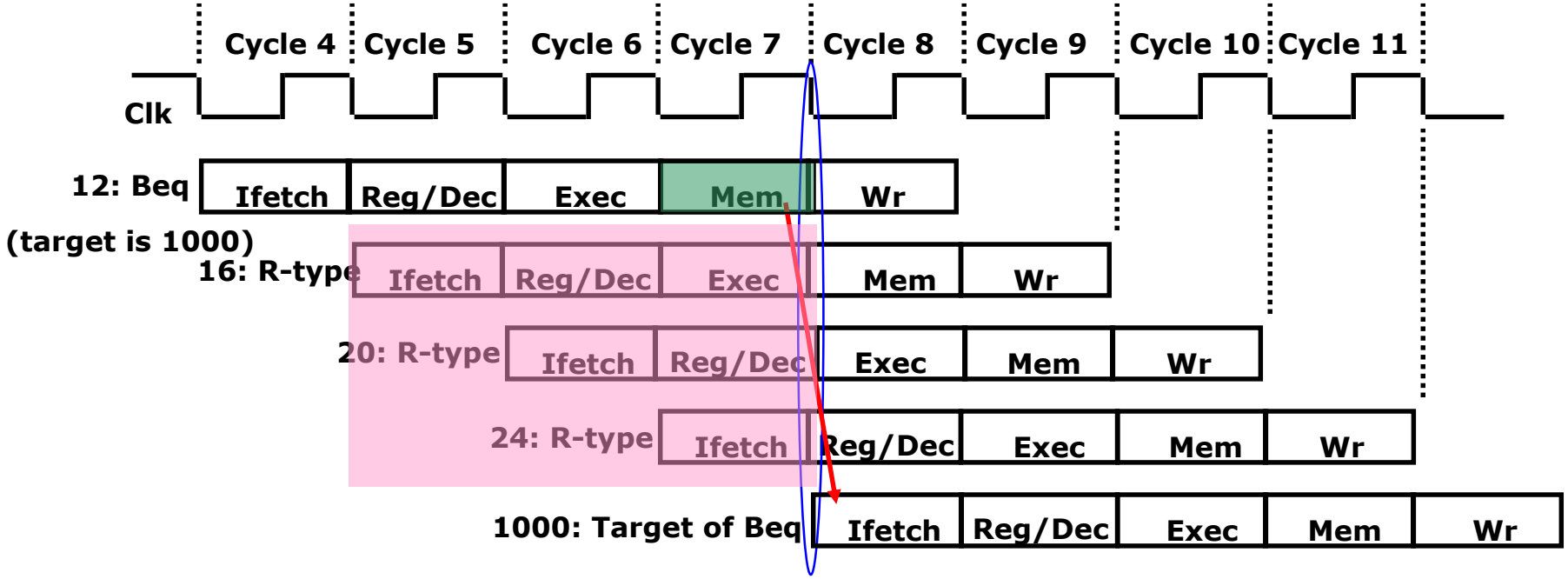
❑ 简单（静态）分支预测方法

- 总是预测条件不满足，即：继续执行分支指令的后续指令
- 预测错误时，需把流水线中三条本不该执行的指令丢弃掉



❑ 简单（静态）分支预测方法

- 总是预测条件不满足，即：继续执行分支指令的后续指令
- 预测错误时，需把流水线中三条本不该执行的指令丢弃掉



延迟损失时间片C：预测错误给流水线带来的延迟损失。这里C=3。

1. 分支预测

- 简单（静态）分支预测方法
  - 总是预测条件不满足，即：继续执行分支指令的后续指令
  - 预测错误时，需把流水线中三条本不该执行的指令丢弃掉
- 性能
  - 如果转移概率是50%，则预测正确率仅有50%。这种方法正确率不高。
- 预测错误的代价
  - 预测错误的代价与何时能确定是否转移有关。越早确定，代价越少。
  - 可以把“是否转移”的确定工作提前，而不要等到MEM阶段才确定。

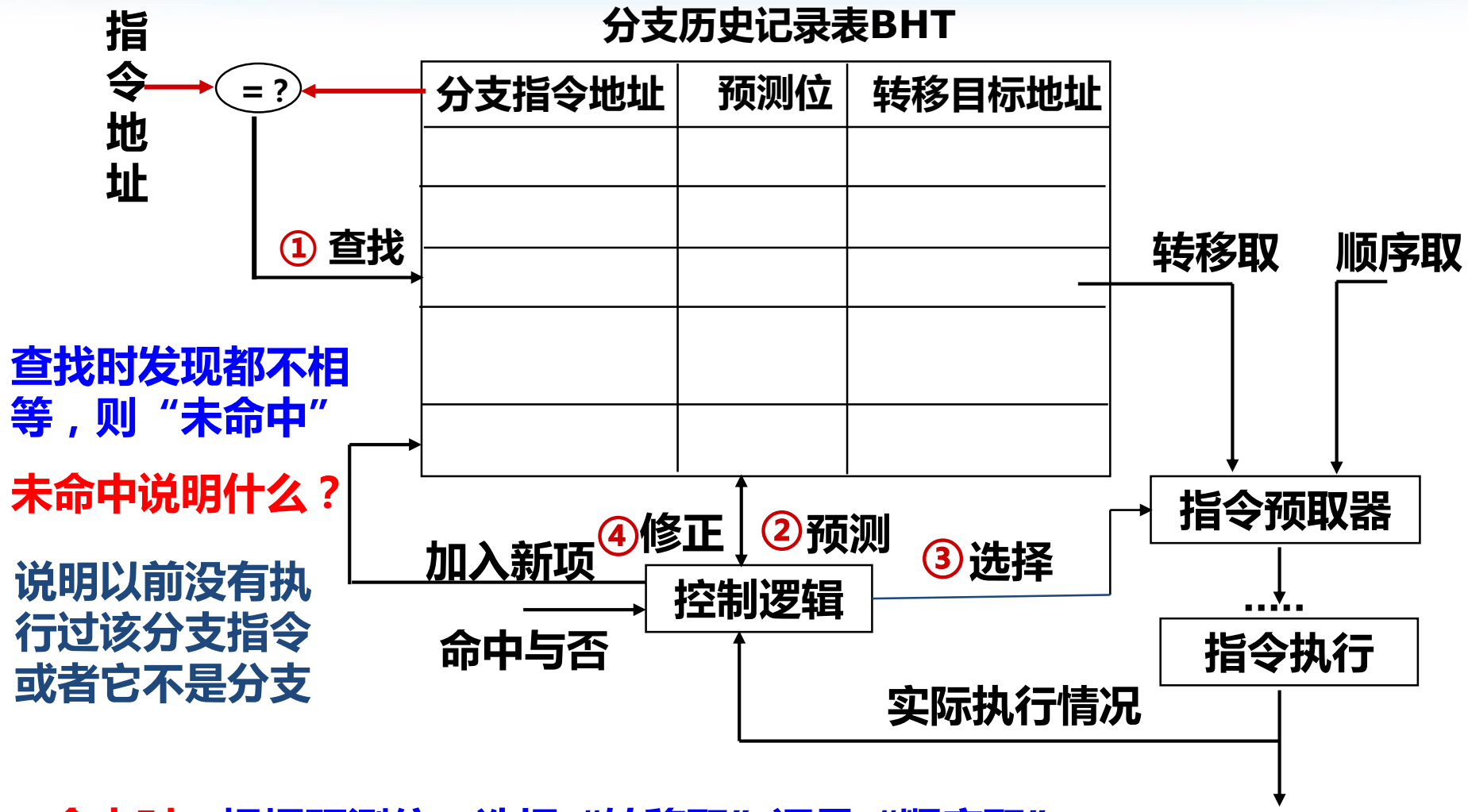
那最早可以提前到哪个阶段呢？

1. 分支预测

❑ 动态分支预测方法

- 利用最近转移发生的情况，来预测下一次可能发生的转移
- 预测后，在实际发生时验证并调整预测
- 转移发生的历史情况记录在特定的缓存中(有多个不同的名称)
  - 分支历史记录表BHT(Branch History Table)
  - 分支预测缓冲BPB(Branch Prediction Buffer)
  - 分支目标缓冲BTB(Branch Target Buffer)
- 每个表项由分支指令地址低位作索引，故在IF阶段就可以取到预测位

1. 分支预测



- 命中时：根据预测位，选择“转移取”还是“顺序取”
- 未命中时：加入新项，并填入指令地址和转移目标地址、初始化预测位

2. 延迟分支

- 属于静态调度技术，由编译器重排指令顺序来实现
- 基本思想
  - 把分支指令前面的与分支指令无关的指令调度到分支指令后面执行，以填充延迟时间片(也称分支延迟槽Branch Delay slot)，找不到可调度的指令时用nop指令填充

2. 延迟分支

如何对以下程序段进行分支延迟调度？

(假定分支延迟时间片为2)

```
lw $1, 0($2)
lw $3, 0($2)
add  $6, $4, $2
beq $3, $5, 2
add $3, $3,$2
sw $1, 0($2)
    ……
```

调度后

```
lw $3, 0($2)
add  $6, $4, $2
beq $3, $5, 2
lw $1, 0($2)
nop
add $3, $3,$2
sw $1, 0($2)
```

若分支条件判断和目标地址计算提前到ID阶段，则分支延迟时间片减为1

```
lw $3, 0($2)
add  $6, $4, $2
beq $3, $5, 2
lw $1, 0($2)
add $3, $3,$2
sw $1, 0($2)
    ……
```

调度后可能带来其他问题：
产生新的load-use数据冒险

调度后，降低了分支延迟损失