

Hardware scheduling

检测 hazard 并进行 forwarding、pipeline、stall 等

Software scheduling (Compiler)

code movement

- Unrolling loop

- Software pipeline

- VLIW

change

- Kill Branch Instruction

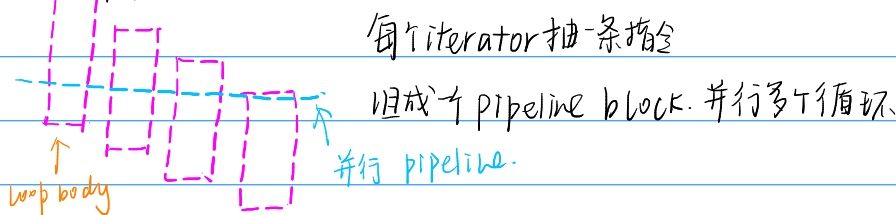
- conditional execution instruction

- superblock

- trace cache

unrolling { advantage: 并行, 大大减少 overhead 总时间
disadvantage: 要求大量 registers, 代码膨胀

software pipeline



产生 stall 的原因: 指令之间的距离太近了, 生产与消费距离太近了

现在把不同 iteration 的指令穿插在一起, 拉开距离

loop-carried independence.

for (int i=0; i<8; ++i) A = A + C[i]

用快速累加的方法 unrolling, 并拉开生产者-消费者距离, 不让 A 是每条的生产兼消费者

cycle 1: $t_0 = C[0] + C[1]$

$t_1 = C[2] + C[3]$

$t_2 = C[4] + C[5]$

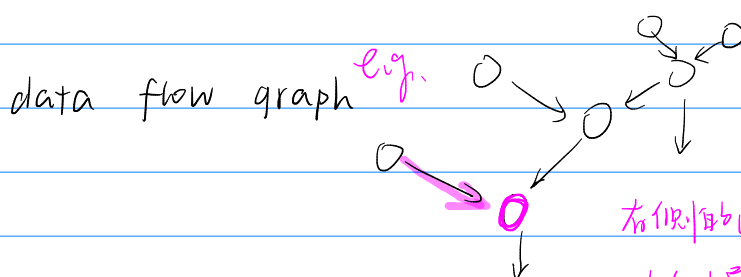
$t_3 = C[6] + C[7]$

cycle 2: $t_4 = t_0 + t_1$

$t_5 = t_2 + t_3$

cycle 3: $res = t_4 + t_5$

用 3 个 add 的时间得到以上 for 的结果



右侧的距离更长, 而且有依赖关系

左侧通过增加节点拉长 →

在左侧添加一些指令, 与右侧并行而不是 stall

two issue (dual issue)



super scalar

{ single scalar } 在同一时刻硬件可以处理多个且数据
{ multi scalar } (data path)

tomasulo 多个 function unit, 并行计算不同的运算.

是 multi-scalar. (super scalar).

有多个 int, float, load, store units. 可以同时处理数据.

tomasulo single issue: 顺 pipeline. 整体是 single scalar. 看起来只处理一个
但其实 tomasulo 可以 multi-issue 并做 super scalar. 充分利用 units

VLZW: Very Long Instruction Word 超长指令字

① 普通指令 opcode | operands ← 仅这个指令相关的 rs1, rs2, rd.

② 德州扑克 opcode | operands 1 | operands 2 | 3 | 4, ... 一条指令, 多条数据
编译器挑出不具有依赖关系的指令, 拼成超长指令中, 同时 issue. 多发
(通用处理器). 充分利用 function unit.

LWC = Conditional Load

LWC R8, 20(R10), R10

{ Load Word to register temp

{ if R10 != 0 commit

LWC 可以用于填充由于 dual/multi-issue 产生的空位.

LWC 相当于一种 prediction execution.

LWC 是一种 conditional execution instruction

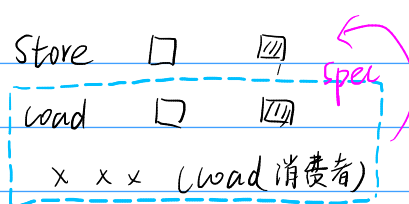
有 conditional Load/store/add/mul ... 预先执行, 等 condition 结果出现决定是否发布

当一个 branch 的 block 比较短, 可以都变为 conditional execution

但是这些预先做就要占用 Mem, ALU, CBR 等有风险开销, Kill Branch

并且预先做有可能面临 page fault, cache miss. 提前取时间开销大, trade-off

所以会要求 block 比较短.



不希望 load 被 store 阻断. 先执行 load, 再 store

这样 load 后面的消费者可以先用

但是 load 取数据跟 store 地址冲突了

则这种预先指令要被取消

把 load 提前并添加 spec 指令

检查相关问题: 冒险失败后可以返回冒险前状态.

bundle & group. { group 编译器挑出可以并行的指令 set
bundle

EPIC: Explicit 显式可并行. 即指 VLIW. 这种. 可多发的. 编译器发给硬件. super scalar

⇔ Inexplicit 隐式可并行.