

→ 11

4/21 第三章

3. (1) 不存在数据依赖。因为存储和加载操作使用不同的虚拟和物理地址。

(2) 存在。因为存储和加载操作使用相同的虚拟地址和物理地址中的不同偏移量。

(3) 不存在。因为存储和加载操作使用不同的虚拟地址和物理地址。

(1) ⁽²⁾ 存在数据依赖, (3) 仍不存在, 因为(3)中地址间隔超过4KB, 而(1)、(2)未超过并被映射到相同物理地址页。

9. (1) ~~5 fdiv~~

fld f2, 0(a0) 4

①: fld, ~~与 fdiv 有关~~ 完成

fdiv.d f8, f0, f2 11

⑤: fld 完成, fdiv 进入, ⑦ 完成

fmul.d f2, f6, f2 5

⑥: fmul.d 进入, ⑧ 完成

fld f4, 0(a1) 4

⑦: fld 进入, ⑨ 完成

fadd.d f4, f0, f4 3

⑪: fadd.d 进入, ⑫ 完成

fadd.d f10, f8, f2 3

⑫: fadd.d f10, f8, f2 进入, ⑬ 完成

fsd f10, 0(a0) 2

⑯: fsd f10, 0(a0) 进入, ⑰ 完成

fsd f4, 0(a1) 2

⑯: fsd f4, 0(a1) 进入, ⑰ 完成

addi a0, a0, 8 1

⑯: addi a0, a0, 8 进入完成

addi a1, a1, 8 1

⑯: addi a1, a1, 8 进入, 完成

sub x20, x4, a0 1

⑯: sub x20, x4, a0 进入, 完成

bnez x20, Loop 2

⑯: bnez x20, Loop 进入, 完成

二. 需要 31 个周期。

(2) # f₂, f₁₀, f₉, f₈, f₇ 中央, 经过 5 周期, fmul.d, fld 进入

①: fld, fdiv.d 进入中央, ②完成中央, fld ④完成, fdiv.d ⑥完成

⑤: fmul.d, fld 进入, ⑥完成中央, ⑦完成 fmul.d ⑧完成, fld ⑩完成.

⑯: fadd.d 和 fadd.d 进入, ⑬完成中央, fadd.d ⑭完成. 一个 18

⑪: fsd 和 fsd 进入, ⑬完成中央, fsd ⑫完成

⑮: addi 和 addi 进入, ⑬完成中央, ⑭完成 addi 完成.

⑯: sub 和 bnez 进入, ⑬完成, ⑭完成, sub ⑭完成, bnez ⑮完成.

二. 需要 29 个周期

(3) fld f₂, 0(a0)

fB f₄, 0(a1)

fdiv.d f₈, f₀, f₂

fadd.d f₇, f₀, f₄

fmul.d f₂, f₆, f₂

fsd f₄, 0(a1)

fadd.d f₁₀, f₈, f₂

addi a₁, a₁, 8

fsd f₁₀, 0(a0)

addi a₀, a₀, 8

sub x₂₀, x₄, a₀

bnez x₂₀, Loop

①: fdiv.d 进入, ④完成

⑤: fdiv.d 和 fadd.d 进入, ③开始, fdiv.d ⑩完成, fadd.d ⑦完成.

⑯: fmul.d 和 fsd 进入, fmul.d ④完成, fsd ⑦完成.

⑪: fadd.d 和 addi 进入, fadd.d ③完成, addi ②完成.

⑫: fsd 和 addi 进入, fsd ④完成, addi ③完成.

⑬: sub 和 bnez 进入, sub ④完成, bnez ⑤完成.

(4). Loop: fld f₄, 0(a0)

fmul.d T₉, f₀, f₂

fdiv.d f₈, f₄, f₂

fld T₁₀, 0(a1)

fadd.d f₆, f₀, T₁₀

fsub.d T₁₁, f₈, f₆

fsd T₁₂, 0(a1)

11. 显式重命名 在编译或运行时将虚拟寄存器映射到物理寄存器，优点是可以高效实现寄存器分配，缺点是需要在程序中插入额外指令，增加代码量；

隐式重命名 在处理器内部自动将虚拟寄存器映射到物理寄存器，优点是不用在程序中插入额外指令，可以有效减少代码量和编译时间，缺点是需要额外的硬件支持。

实现方式：显式重命名可以在程序中插入专门的重命名指令来实现，例如 Intel x86 架构中的 MOV 指令。
隐式重命名在处理器中，指令首先被解码并分配到物理寄存器，将其重命名为表中的条目，例如 Intel x86 架构中的 Pentium Pro 处理器。