

周期

(1) f1d d2, o(a0) - 1~4

fdiv d8, d0, d2 - 5~15

fmul.d f2, f6, d2 - 6~10

fld f4, o(a1) - 7~10

fadd.d f4, f0, f4 - 8~13

fadd.d f10, f8, d2 - 10~18

fsub d10, o(a0) - 11~20

fsub d4, o(a1) - 12~21

addi , a0, a0, 8 - 21~22

addi , a1, a1, 8 - 22~23

sub x20, x4, a0 - 23~29

bz x20, loop - 25~27 \rightarrow (1) 27周月

周期

周期

(2) f1d d2, o(a0) - 1~4

fdiv d8, d0, d2 - 5~15

fmul.d f2, f6, d2 - 6~9

fld f4, o(a1) - 6~9

fadd.d f4, f0, f4 - 10~12

fadd.d f10, f8, d2 - 16~18

fsub d10, o(a0) - 18~19~20

fsub d4, o(a1) - 20~21

addi , a0, a0, 8 - 21~22

addi , a1, a1, 8 - 21

fsub x20, x4, a0 - 22~24

bz x20, loop - 25~26 \rightarrow (2) 26周月

(3) $f1d f2, o(a0)$ $\xrightarrow{fdiv.d f8, f0, f2}$
 $\xrightarrow{fmul.d f10, f6, f2}$ $\xrightarrow{fadd.d f10, f8, f2}$
 $f1d f4, o(a1)$ $\xrightarrow{fadd.d f4, f0, f4} \xrightarrow{fsub f4, o(a1)}$ $\xrightarrow{\text{add}}$

重排后: $f1d f2, o(a0)$ - 1~4

$f1d f4, o(a1)$ - 1~4

$fdiv.d f8, f0, f2$ - 5~15

~~$fsub$~~

$fadd.d f4, f0, f4$ - 5~7

$fmul.d f2, f6, f2$ - 6~10

$fsub f4, o(a1)$ - 8~9

$addi al, a1, 8$ - 10

$fadd.d f10, f8, f2$ - 16~18

$fsub f10, o(a0)$ - 19~20

~~$faddi$~~ $a0, a0, 8$ - 21

$sub x20, x4, a0$ - 22~24

$bz x20, loop$ - 25~26 $\Rightarrow 26$ 循环

这是因为最大具依赖的指令周期 $4+11+3+2+1+3+2=26$ 的原因

10. $f\text{ld } T_9, o(a)$

~~fload~~ $\downarrow \text{fload}$

$f\text{mul.d } T_{10}, T_0, T_2$

$f\text{div.d } T_{11}, T_9, T_{10}$

$f\text{ld } T_{12}, o(a)$

$f\text{add.d } T_{13}, T_0, T_{12}$

$f\text{sub.d } T_{14}, T_{11}, T_{13}$

$f\text{sd } T_{14}, o(a)$

11. 显式重命名通过硬件机制实现对寄存器的重命名；隐式重命名则通过编译器和软件的机制实现对寄存器的重命名。

区别：

(1) 显式重命名通过硬件实现，充分利用硬件并行性，允许更多的指令并行执行，可以减少流水线中的潜在冒险和停顿，提高吞吐量。

隐式重命名通过编译器生成的代码来实现寄存器重命名，因此并不能充分利用硬件并行性。

(2) 显式重命名比隐式重命名更适合于动态发射处理器，因为显式重命名允许发射时对寄存器重命名。

(3) 显式重命名能更好地隐藏指令流水线的细节，使编译器和程序员更容易地进行代码优化。而隐式重命名在编译器中进行寄存器分配使代码可读性和维护性差。

优缺点：(1)同(2)，但显式重命名需要更多硬件成本

(5) 隐式重命名不需要额外资源。但可能限制^{硬件}指令并行执行的能力，因而对编译器和软件进行更多开发工作。

可能的实现方式：(6) 显式重命名常量包括一个重命名表，用于保存物理寄存器和逻辑寄存器之间的映射关系

(7) 隐式重命名会对程序进行静态 ~~编译~~ 分析，将寄存器分配给不同的变量