

3-9

1) 指令

需要周期数

总周期数

fld	$f_2, o(a_2)$	4	4
fdiv.d	f_8, f_0, f_2	11	15
fmul.d	f_2, f_6, f_2	5	16
fld	$f_4, o(a_1)$	4	10
fadd.d	f_4, f_0, f_4	3	13
fadd.d	f_{10}, f_8, f_2	3	19
fsd	$f_{10}, o(a_0)$	2	21
fsd	$f_4, o(a_1)$	2	23
addi	$a_0, a_0, 8$	1	22
addi	$a_1, a_1, 8$	1	23
sub	x_{20}, x_4, a_0	1	24
bnz	x_{20}, loop	2	26

共26个周期

2) 指令

需要周期

发射周期

完成周期

fld	$f_2, o(a_2)$	4	1	4
fdiv.d	f_8, f_0, f_2	11	1	15
fmul.d	f_2, f_6, f_2	5	2	9
fld	$f_4, o(a_1)$	4	2	5
fadd.d	f_4, f_0, f_4	3	3	8
fadd.d	f_{10}, f_8, f_2	3	3	18
fsd	$f_{10}, o(a_0)$	2	4	20
fsd	$f_4, o(a_1)$	2	4	10
addi	$a_0, a_0, 8$	1	5	5
addi	$a_1, a_1, 8$	1	5	5
sub	x_{20}, x_4, a_0	1	6	6

bnz	x20, loop	2	6	8
				共27个周期
b)		需要周期	发射周期	完成周期
	fld f2, 0(a2)	4	1	4
	fld f4, 0(a1)	4	1	4
	fdiiv.d f8, f0, f2	11	2	12
	fmul.d f2, f6, f2	5	2	6
	fadd.d f4, f0, f4	3	3	5
	fadd.d f10, f8, f2	3	13	15
	fsd f4, 0(a1)	2	13	14
	fsd f10, 0(a0)	2	16	17
	addi a0, a0, 8	3	18	20
	addi a1, a1, 8	3	18	20
	sub x20, x4, a0	3	21	23
	bnz x20, loop	2	24	25
				共25周期

3-10 loop: fld T9, 0(a1)
 fmul.d f2, 0, f2
 fdiiv.d f8, T9, f2
 fld T10, 0(a1)
 fadd.d f6, f0, T10
 fsub.d f8, f8, f6
 fsd f8, 0(a1)

3-11

显式重命名和隐式重命名是两种不同的寄存器重命名方式，用于解决处理器中的数据相关性和数据冲突问题。它们有以下区别、优缺点和可能的实现方式：

区别：

显式重命名：程序员在编写代码时明确指定重命名后的寄存器，通过编写特定的指令来实现寄存器重命名。编译器会根据指令中指定的重命名寄存器来进行重命名操作。

隐式重命名：处理器在执行指令时自动进行寄存器重命名，无需程序员显式指定重命名寄存器。处理器会根据硬件设计和指令执行的需求来自动选择重命名寄存器。

优缺点：

显式重命名：优点：程序员可以精确控制寄存器的重命名过程，灵活性较高，可以根据实际需求进行优化。

缺点：需要在代码中显式指定重命名寄存器，增加了编程的复杂性，可能需要修改现有的代码。

隐式重命名：优点：无需程序员显式指定重命名寄存器，减少了编程的复杂性，对现有的代码无需修改。

缺点：重命名过程由硬件自动完成，对处理器的设计和实现要求较高，可能增加硬件的复杂性和成本。

可能的实现方式：

显式重命名：编译器在编译时根据指令中指定的重命名寄存器来生成相应的代码，将原始寄存器替换为重命名寄存器。编译器需要识别指令中的重命名寄存器，并进行寄存器分配和重命名代码生成。

隐式重命名：处理器在执行指令时自动进行寄存器重命名，无需编译器和程序员显式参与。处理器需要具有额外的硬件来实现重命名过程，例如重命名表（Renaming Table）或物理寄存器文件（Physical Register File）来存储重命名寄存器的映射关系。处理器还需要在指令发射和执行阶段进行重命名操作，并保证指令的顺序和正确性。

