

9.

Loop:

I1: fld f2,0(a0)  
I2: fdiv.d f8,f0,f2  
I3: fmul.d f2,f6,f2  
I4: fld f4,0(a1)  
I5: fadd.d f4,f0,f4  
I6: fadd.d f10,f8,f2  
I7: fsd f10,0(a0)  
I8: fsd f4,0(a1)  
I9: addi a0,a0,8  
I10: addi a1,a1,8  
I11: sub x20,x4,a0  
I12: bnez x20,Loop

(1) 29 个周期

(2) 24 个周期

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
I1	I1	I1	I1	I1																					
I2					I	I	I	I	I	I2	I2	I2	I2	I2	I2										
I3					I	I	I	I	I																
I4					I	I	I	I																	
I5					I	I	I	I		I5	I5	I5													
I6																I6	I6	I6							
I7																		I7	I7						
I8																		I8	I8						
I9																			I9						
I10																			I10						
I11																				I11					
I12																					I12	I12			

(3)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
I1	I1	I1	I1	I1																					
I4	I4	I4	I4	I4	I4																				
I2					I2	I2	I2	I2	I2	I2	I2	I2	I2	I2	I2	I2	I2	I2							
I5					I5	I5	I5																		
I3						I3	I3	I3	I3	I3															
I8								I8																	
I10									I10	I10															
I6																I6	I6	I6							
I7																			I7	I7					
I9																			I9						
I11																				I11					
I12																					I12	I12			

24 个周期，感觉因为 div 需要 11 个周期以及后续操作一直到循环最后一步都需要 div 的结果，导致周期数无法继续缩减

10.

fld T9,0(a0)

fmul.d T10,T0,T2

fdiv.d	T11,T9,T10
fld	T12,0(a1)
fadd.d	T13,T0,T12
fsub.d	T14,T11,T13
fsd	T14,0(a1)

11.

在 CPU 中，显式重命名和隐式重命名通常用于描述指令集和寄存器的名称。

显式重命名是指将指令集或寄存器的原名重新命名为一个新的名称。这种重命名方式可以在编译时或运行时进行，通常用于更改指令集或寄存器的名称。

隐式重命名则是指通过在指令集中使用特定的语法或标识符来指定指令集或寄存器的名称。这种重命名方式需要在程序中使用新名称来引用指令集或寄存器，而不是使用旧名称。

显式重命名的优点是可以在编译时或运行时立即看到修改的结果，方便调试和修改程序，同时也能够提高程序的可读性和可维护性。但是，显式重命名可能会导致指令集或寄存器名称的冲突，特别是在大规模程序库中。

隐式重命名的优点是可以悄悄地修改程序中的指令集或寄存器名称，从而减少了修改的时间和成本。同时，隐式重命名也可以避免指令集或寄存器名称的冲突，因为程序员需要使用新名称来引用指令集或寄存器。但是，隐式重命名的缺点是需要手动引用指令集或寄存器，使得程序难以理解和阅读，特别是在大规模程序库中。

在实现方式上，显式重命名可以通过手动修改指令集或寄存器的名称来实现，也可以通过静态代码分析工具来自动识别和修改指令集或寄存器名称的重命名。隐式重命名则通常需要使用特定的语法或标识符来实现。