

作业 6 VLIW

考虑一个 VLIW CPU，每条长指令包含 4 个短指令，具体格式如下。

MEMORY	INTEGER	CONTROL	FLOAT
--------	---------	---------	-------

下表描述了每个短指令的执行周期数。假设每个短指令执行单元都是基于流水线的（即，每个周期都能接收新的请求）。

类别	周期数	指令	描述	功能
CONTROL	1	BEQ Label Rs1, Rs2 NOP	分支指令 空指令	If Rs1 == Rs2: PC = Label PC = Next PC
MEMORY	3	LD Rd, [Rs]	读内存	Rd = MEM[Rs]
INTEGER	2	IADD Rd, Rs1, Rs2	整数加	Rd = Rs1 + Rs2
FLOAT	4	FADD Rd, Rs1, Rs2	浮点加	Rd = Rs1 + Rs2

考虑下面的代码（短指令形式）：

	指令	注释
	<Initialize R0-R2>	R0-R2 指向合法内存
	LOOP:	
1	LD R0, [R0]	R0 <- MEM[R0]
2	LD R1, [R1]	R1 <- MEM[R1]
3	IADD R4, R0, R1	R4 <- R0 + R1
4	FADD R5, R0, R4	R5 <- R0 + R4
5	LD R6, [R2]	R6 <- MEM[R2]
6	LD R2, [R0]	R2 <- MEM[R0]
7	FADD R3, R1, R6	R3 <- R1 + R6
8	IADD R4, R2, R4	R4 <- R2 + R4
9	IADD R5, R5, R4	R5 <- R5 + R4
10	IADD R0, R6, R2	R0 <- R6 + R2
11	IADD R0, R0, R3	R0 <- R0 + R3
12	BEQ LOOP, R0, R5	GOTO LOOP if R0 == R5

请填写下表，将上述短指令形式的代码改写为符合题目给出的 VLIW 格式的代码，目标是让所有指令执行完所用的周期数最少。注意：只考虑 1 次循环（包括 BEQ 指令），不包括初始化部分，也不需要跨循环优化（例如，循环展开），不能增加/减少指令，或修改原指令。

Cycle	MEMORY	INTEGER	CONTROL	FLOAT
1	1 (LD R0, [R0])			
2	2 (LD R1, [R1])			
3	5 (LD R6, [R2])			

4	6 (LD R2, [R0])			
5		3 (IADD R4, R0, R1)		
6				7 (FADD R3, R1, R6)
7		8 (IADD R4, R2, R4)		4 (FADD R5, R0, R4)
8		10 (IADD R0, R6, R2)		
9		8 (IADD R4, R2, R4)		
10		11 (IADD R0, R0, R3)		
11		9 (IADD R5, R5, R4)		
12				
13			12 (BEQ LOOP, R0, R5)	
14				
15				
16				
17				
18				
19				
20				

指令8可以放在上面两个位置中的任意一个。