

【解答】

在给定的 VLIW 架构中，每条长指令包含 4 个短指令槽：MEMORY、INTEGER、CONTROL 和 FLOAT。每个短指令的执行周期数分别为 3、2、1 和 4 周期。为了最小化总执行周期数，我们按依赖关系调度指令，实现了并行化：LD R0, [R0]和 LD R1, [R1]在开始时并行加载，因为彼此独立。接着，IADD R4, R0, R1 依赖于加载完成的 R0 和 R1。之后的 FADD R5, R0, R4 依赖于 R0 和 R4。通过适当安排每条指令的执行顺序，并遵循数据依赖，我们将代码在最小周期内执行完毕。此外，BEQ 指令用于控制循环终止，不需要跨循环优化或增加额外指令。

| Cycle | MEMORY      | INTEGER         | CONTROL          | FLOAT           |
|-------|-------------|-----------------|------------------|-----------------|
| 1     | LD R0, [R0] | -               | -                | -               |
| 2     | LD R1, [R1] | -               | -                | -               |
| 3     | -           | IADD R4, R0, R1 | -                | -               |
| 4     | -           | -               | -                | FADD R5, R0, R4 |
| 5     | LD R6, [R2] | -               | -                | -               |
| 6     | LD R2, [R0] | -               | -                | -               |
| 7     | -           | IADD R4, R2, R4 | -                | -               |
| 8     | -           | IADD R5, R5, R4 | -                | -               |
| 9     | -           | IADD R0, R6, R2 | -                | -               |
| 10    | -           | IADD R0, R0, R3 | -                | -               |
| 11    | -           | -               | BEQ LOOP, R0, R5 | -               |