嵌入式系统 hw4

18300750008 黄剑磊

1 简要分析 CISC 和 RISC 架构各自的优势和劣势。

解. CISC (Complex Instruction Set Computers), 复杂指令集计算机。单个指令完成的任务量大且功能复杂，指令长度灵活，如 PC 上常用的 x86 指令集即为 CISC 架构。

优点：对编译器和程序存储空间的要求较低，

缺点：硬件设计复杂，测试验证难度较高

RISC (Reduced Instruction Set Computers), 精简指令集计算机。单个指令完成的任务量少且功能单一，指令长度相对固定，如 RISC-V、MIPS 以及移动端常见的 ARM 指令集均为 RISC 架构。

优点：硬件设计较为简单，适合利用流水线提升性能

缺点：对编译器设计的要求较高，程序的代码密度较低

□

2 RISC-V 中的基本指令集是什么？列举五个常见的 RISC-V 标准扩展指令集并简要说明它们的作用和应用范围。

解. RISC-V 中的基本指令集包含 RV32I 和 RV64I，为寄存器-寄存器型指令集。

以下是五个常见的 RISC-V 标准扩展指令集及其作用和应用范围：

M 扩展指令集（Multiply/Divide Extension）：该扩展指令集添加了乘法、除法和位移操作指令，用于高性能计算、嵌入式系统和通信应用等。

A 扩展指令集（Atomic Extension）：该扩展指令集添加了原子指令，用于支持多线程并发编程，避免竞争条件，提高程序执行效率。

F 扩展指令集（Floating-Point Extension）：该扩展指令集添加了浮点数操作指令，用于高性能计算和科学计算等应用。

C 扩展指令集（Compressed Extension）：该扩展指令集通过压缩指令长度，实现了更高的指令密度和更小的程序大小，适用于嵌入式系统和存储器受限的应用场景。

Z 扩展指令集（Zero-Operand Extension）：该扩展指令集不需要任何操作数，用于实现一些特殊的操作，如 NOP（空操作指令）、EBREAK（异常断点指令）和 FENCE（内存栅栏指令）等。

□

4 阅读 RISC-V 规范以回答以下问题：

1) RV32I 中的 add 指令和 RV64I 中的 addw 指令均为 32 位整型加法指令，它们是否具有相同的指令操作数（opcode）？此外，RV32I 中的 add 指令和 RV64I 中的 add 指令是否具有相同的指令操作数（opcode）？试分析为什么采取这样的设计。

2) 在 RV64I 中，addw 和 addiw 指令的目标寄存器中存放的 32 位计算结果是否需要进行额外的符号扩展才能用于后续 64 位计算？请说明理由。

解. 1）RV32I 中的 add 指令和 RV64I 中的 addw 指令是不同的指令，它们具有不同的操作码 (opcode)。具体来说，RV32I 中的 add 指令操作码为 0110011，而 RV64I 中的 addw 指令操作码为 0111011。因此，虽然它们都是 32 位整数加法指令，但它们具有不同的操作码，可以区分开来。

这样的设计是为了保持指令集的一致性和可扩展性。RISC-V 指令集被设计为具有可扩展性，可以轻松地添加新的指令和指令集扩展。因此，在 RV64I 中添加了一个新的指令 addw，以处理 32 位有符号整数的加法，而不会与 RV32I 中的现有 add 指令产生冲突。这样的设计允许在不破坏现有指令集的情况下进行扩展，保持了指令集的一致性和可扩展性。

2）在 RV64I 中，addw 指令和 addiw 指令的目标寄存器中存放的 32 位计算结果不需要进行额外的符号扩展，就可以用于后续的 64 位计算。

这是因为，在 RV64I 中，所有的 32 位操作都会自动被符号扩展为 64 位。当使用 32 位指令进行计算时，指令执行的结果会自动被符号扩展为 64 位，并存储到 64 位目标寄存器中。因此，如果在 addw 或 addiw 指令之后使用 64 位指令进行计算，无需进行额外的符号扩展，因为 32 位结果已经被自动扩展为 64 位。 □

5 什么是 RISC-V 的 I 标准指令集中存在的 HINT 指令空间？它有什么作用？

解. RISC-V 是一种开放式指令集架构，其 I 标准指令集是最基本的指令集。其中存在一些被称为 HINT（暗示）指令的特殊指令。HINT 指令集并不是 RISC-V 指令集的一部分，而是被视为可选的指令。

HINT 指令空间包含了一组专门用于提高性能的指令，它们可以向处理器发出各种提示，以帮助优化代码的执行。这些提示可以告诉处理器如何处理指令流，例如预测分支、缓存等。

HINT 指令可以用于提高代码的性能，特别是在高度优化的应用程序中。然而，它们需要开发者具有深入的了解和技能才能使用。在大多数情况下，使用普通的 RISC-V 指令就可以满足大多数应用程序的需要，而不需要使用 HINT 指令。

□

6 考虑如下指令序列：

div a2,a0,a1

rem a3,a0,a1

假设寄存器 a0 和 a1 的初始值分别为 16 和-5，则上述指令序列执行完成后 a2 和 a3 寄存器中的值分别是多少？简要说明 RISC-V 的 M 标准指令集中对除法和余数指令的符号规定。

解. a2 的值为-3，a3 的值为 1

余数总是正的，若 a0 和 a1 符号相反，商的结果为负号，反之为正号。 □

11 写出以下指令使用的寻址模式。

1) jal ra,0x88

2) jalr x0,ra,0

3) addi a0,a1,4

4) mul a0,a1,a2

5) ld a4,16(sp)

解. 1) 偏移量寻址

2) 内存直接寻址

3) 立即数寻址
4) 寄存器直接寻址
5) 偏移量寻址 □