

3.7 第三周

9. 对^{方法}不同微处理器架构进行性能分析和对比的有:

(Dhrystone是测量处理器运算能力的^{基准}最^{常见}程序之一,常用于处理器的整型运算和逻辑运算性能的测量,其输出结果是每秒^{秒钟}运行Dhrystone的次数,单位为DMIPS/MHz)

CoreMark是一个综合基准,用于测量嵌入式系统中使用的中央处理器(CPU)的性能,包含以下算法列表处理(增删改查和排序)、矩阵操作(公共矩阵操作)、状态机(确定输入流是^否包含有效数字)和CRC)

基准测试 (如 Dhrystone 和 CoreMark) 在同一应用程序的多个版本上测试不同架构的处理器,从而比较它们的性能,通常包括一系列的测试套件以测试代表型的性能。

性能模拟: 使用计算机模型来模拟不同架构的处理器性能,可以在不实际运行程序的情况下预测不同架构的性能,通常使用专用的模拟器软件进行。

模拟器: 通过使用特定处理器架构的模拟器来运行相同的应用程序,来比较不同架构之间的性能,可以模拟不同的处理器架构并运行相同的应用程序,以确定性能优劣。

实际测试: 在实际的硬件环境下运行应用程序来测试不同架构的处理器。

Dhrystone 评分是基于每秒钟运行 Dhrystone 的次数。(Dhrystone 基准测试使用^的程序代码包括常见的计算任务)

CoreMark 评分是基于每秒钟可执行 CoreMark 的次数。(CoreMark — — — — 也包括常见的计算任务)



10. 层次化设计在计算机系统中的应用

① 操作系统层次：操作系统可以被看作是计算机硬件和应用程序之间的一个中间层。操作系统提供了一系列的系统调用和接口，让应用程序可以与硬件交互。

② 网络协议层次：计算机网络通信协议通常采用分层的设计方式，将通信过程分解成多个层次，每个层次负责不同的功能，每个层次之间通过接口相互连接。

③ 处理器架构层次：现代处理器架构通常采用分层的设计方式，将处理功能分解成负责不同功能的多个层次。

④ 硬件和软件层次：计算机系统由硬件和软件两个层次构成，不同层次专注于不同的任务。

实际意义：① 提高软件的可移植性和可维护性：分层的设计可以让软件开发者只需要关注当前层次的接口和实现细节，而无需关心底层的实现细节，从而提高了软件的可移植性和可维护性。

② 提高系统的可扩展性：分层的设计可以让系统在不改变底层实现的情况下增加新的功能，从而提高了系统的可扩展性。

③ 提高系统的可靠性和安全性：分层的设计可以让系统在某一层次出现问题时只影响当前层次和上层的功能，而不会对整个系统产生影响，从而提高了系统的可靠性和安全性。

④ 提高系统的性能：分层的设计可以让系统在底层实现中对硬件资源进行更高效的利用。



附加题2: "A New Golden Age for Computer Architecture" 读后感

在"A New Golden Age for Computer Architecture"中,作者探讨了当前计算机体系结构面临的挑战和机遇,以及未来可能带来的技术革新和经济发展。作者提到我们正处在一个黄金时代的边缘,而这个黄金时代中将显著提高成本、性能、能源和安全性。与此同时,我们必须直面:① 针对高级和领域特定语言的软硬件协同设计;② 安全挑战;③ 自由和开放的架构和开源实现;④ agile chip开发等方面的挑战。

具体而言,在高级和领域特定语言的软硬件协同设计上新的进步或可以创建编译器和特定领域的架构,并在这"黄金时代"提供EDA的进步;与此同时,我们需要更加关注数据安全,保护数据免受定时攻击;而自由且开放的ISA将汇集众多人的智慧,使其发展愈见迅速,开放的体系结构还可以为FPGA和提供开源处理器设计,进一步使得为"黄金时代"提供可能性。

综上所述,这个时代面临诸多挑战的同时仍拥有着更多机遇。在遭遇列登内德缩放比例定律和摩尔定律终结、标准微处理器性能增长减速的现状之后,我们并不应该执着于过去而迷失方向,此时着眼未来(如通过高级、特定于领域的语言和体系结构来把架构师从专有指令集中解放出来)。这样我们才能把握佳时代的机遇,享受这"黄金时代"。

