

劣势在于：① 成本更高
② 可靠性不如传统计算机。

3.1.

9. 简述对不同处理器架构进行性能分析和对比的方法：

- ① 指令级模拟 (ILS)：指令级模拟是一种模拟技术，用于模拟指令和微处理器状态的变化，以评估性能和可靠性。
- ② 功能仿真 (FS)：用于模拟机器语言和指令，以及微处理器状态变化，以评估微处理器性能。
- ③ 综合仿真：通过模拟软件来模拟机器语言和指令。
- ④ 性能分析：对微型处理器架构进行性能分析和对比，包括时间分析、空间分析和性能分析等。
- ⑤ 验证：可以利用基于规则的验证工具来验证微处理器架构及其行为是否符合预期。

(2) Dhrystone 标准的测试方法很简单，就是单位时间内跑了多少次 Dhrystone 程序，其指标单位为 DMIPS/MHz，表示在 Dhrystone 标准的测试方法下，每秒处理的百万级的机器语言指令数。~~每秒执行条数~~

CoreMark 跑分是通过运行 C 语言代码得出来的分数，基准测试涵盖了整数计算、浮点计算、内存访问等方面。与 Dhrystone 不同，CoreMark 还考虑了缓存大小对性能的影响，并且提供了一个可扩展的框架，允许用户添加自定义测试。最终得分表示为 CoreMark/MHz，即在 1MHz 条件下，每秒钟可执行多少个 CoreMark。

10. 答：(1). ① 指令集架构方面：将程序设计中的指令集按功能分为多层，从低级指令到高级指令，组成指令集架构。

② 操作系统设计上：操作系统通过设计为用户提供了多个易用的层次，比如系统中的文件系统、内存管理层、设备驱动层等。

③ 网络架构：网络架构是一个多层次化的架构，它将网络系统划分为不同的层次，比如应用层、传输层、网络层、链路层等。

④ 软件体系结构上：软件体系结构也是一个多层次的体系结构，它将软件体系结构划分为不同的层次，如用户界面层、数据访问层等。

(2) 层次化设计的特点为计算机系统提供了模块化、可维护性和可扩展性，使得计算机系统能够更加灵活和可靠地完成任务。此外，层次化的设计有助于将复杂的系统分割成多个独立的、有层次的子系统，各子系统只负责某一特定的功能，从而可以显著减少设计和开发系统的复杂度，使开发和维护更加容易。

附录2. 读后感

随着时代的进步和发展，半导体和集成电路行业迅猛发展，一直指引行业发展的摩尔定律和 Dennard Scaling 开始失效，这让很多人产生了恐慌。但读了 John L. Hennessy 的文章以后，发现作者不仅不认为这是一个值得关注的问题，还认为这为计算机架构师带来了巨大的机遇。分析说理可谓十分生动详细。

文中，Hennessy 提出了两种方法来进一步提高程序运行性能，第一，通过提高现代高级语言的编译性能，这归属于软件层面，且不考虑；其次，通过构建领域特定体系结构，可以大大提高性能和效率，DSL 就是一个例子。我们可以发现，在后摩尔时代，硬件工程师对提高性能和效率的作用大大增加，这即是计算机架构师“黄金时代”最好的说明。

总之，通过这篇文章，我了解到了计算机架构的重要性，以及如何有效解决后摩尔时代挑战的办法。即要想让计算机架构取得突破，就必须落实硬件和软件的协同设计，同时运用节能减排技术，以实现最佳的性能和能效。可以肯定的是，未来计算机架构 确实有望进入一个新的黄金时代。