9. 解：

性能分析对比方法：

(1) 占用率：程序运行占用处理器时间与对应处理器运行总时间的比值，若运行同一程序，a处理器占用率比b处理器低，则a处理器对这一程序进行处理的效率更高

(2) 程序运行指令数：

MIPS (Million Instructions Per Second)：

$MIPS = 指令数/(执行时间\times10^6) = 指令数/(指令数\cdot CPI/freq\cdot10^6)$

$= freq/(CPI\cdot10^6)$

CPI：执行一条指令所需周期数

测试程序 BenchMark：
- SPEC
- Dhrystone：单位每秒计算多少次Dhrystone，把在VAX-11/780上运行757 Dhrystone/s定义为1 DMIPS
- CoreMark：相比Dhrystone算法更有实际价值
- EEMBC
- ……

10. 解：计算机的设计总体分为了两个层次 硬件设计和软件设计

而联 指令集则是沟通软件与硬件间的桥梁

软件层：
- 高级语言层：编程工具为C语言等高级语言，用编译程序执行
- 汇编语言层：编程工具为汇编语言指令级，由汇编程序执行
- 操作系统层：由机器指令扩展指令组成，由操作系统实现

指令集 ISA

硬件层：
- 机器语言层：编程工具为机器语言指令集，由微程序执行
- 微程序设计层：编程工具为微指令集，硬件直接执行
- 数字逻辑层：最底层由逻辑门组成电路

实际意义：有利于人们正确地理解计算机系统的工作原理，软、硬件在系统中的作用；同时使计算机系统更易于维护，调试，及建造，提高工作效率

附加题2：

文章提出了3个核心观点，第一：软件的进步可以激励架构创新；其次：提高硬件/软件接口为架构创新提供了机会；第三：市场最终会解决选择何种架构的争论；

作为沟通硬件与软件的桥梁，ISA在过去的几十年间取得了巨大的发展，从CISC的广泛使用到RISC的诞生与逐步发展，我们总是在寻找一种最优的指令架构，提高计算机性能。但自苹果凭借iphone开启Post-PC时代，从2011年起，ISC的代表架构X86的出货量就每年下降近10%，与之相应的带有RISC处理器的芯片的市场占额却日趋庞大。可以说CISC赢得了PC时代的后期，RISC则正在赢得Post-PC时代，而这也正是市场所决定的。值得注意的是，RISC与CISC并非水火不相容，受到流水线简单指令启发，我们可以将复杂的X86转换为内部类似risc的微指令，将执行外并行指令的思想整合到X86中。

但当我们执行某一特定领域的问题时，使用通用处理器很可能会导致部分资源调用的不足与浪费，为了提高其效率，我们首先可以提高高级语言的性能，如执行矩阵乘法时，通过对Python语言的优化，可以大幅度提升效率，比原版本快62000倍以上；其次，针对该领域或定制专门的架构，如dsl(domain-specific languages就是针对某一领域的硬件-软件接口的一种交互语言，支持对应的a架构，便开发者从手有指令集中解放，提高了处理器能，且能更好地保障其安全性。

随着Dennard缩放定律和Moore定律的终结，以及标准微处理器性能增长的减速，我们将面临当前巨大的挑战，但同时如果我们能够抓住，加深对计算机架构的研究，大胆预测，快速改进，将迎来计算机架构的又一个黄金时代。