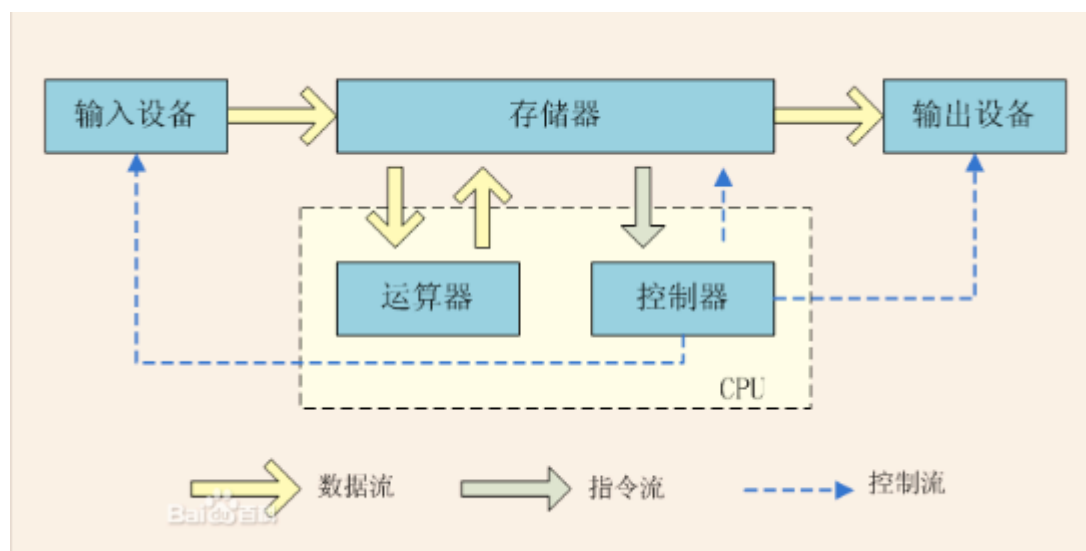


cpu 架构解析

CPU 是中央处理单元（Central Processing Unit）的缩写，它可以被简称做微处理器（Microprocessor），不过经常被人们直接称为处理器（processor）。CPU 是计算机的核心，其重要性好比大脑对于人一样，因为它负责处理、运算计算机内部的所有数据

CPU 架构是按 CPU 的安装插座类型和规格确定的。目前常用的 CPU 按其安装插座规范可分为 Socket x 和 Slot x 两大架构。

以 Intel 处理器为例，Socket 架构的 CPU 中分为 Socket 370、Socket 423 和 Socket 478 三种，分别对应 Intel PIII / Celeron 处理器、P4 Socket 423 处理器和 P4 Socket 478 处理器。Slot x 架构的 CPU 中可分为 Slot 1、Slot 2 两种，分别使用对应规格的 Slot 槽进行安装。其中 Slot 1 是早期 Intel PII、PIII 和 Celeron 处理器采取的构架方式，Slot 2 是尺寸较大的插槽，专门用于安装 P II 和 P III 序列中的 Xeon。Xeon 是一种专用于工作组服务器上的 CPU。



x86 cpu 架构

要了解 X86 和 ARM，就得先了解复杂指令集（CISC）和精简指令集（RISC）从 CPU 发明到现在，有非常多种架构，从我们熟悉的 X86，ARM，到不太熟悉的 MIPS，IA64，它们之间的差距都非常大。但是如果从最基本的逻辑角度来分类的话，它们可以被分为两大类，即所谓的“复杂指令集”与“精简指令集”系统，也就是经常看到的“CISC”与“RISC”。Intel 和 ARM 处理器的第一个区别是，前者使用复杂指令集（CISC），而后者使用精简指令集（RISC）。属于这两种类中的各种架构之间最大的区别，在于它们的设计者考虑问题方式的不同。

8086 是当今 CPU 的鼻祖，所谓 X86 架构也就是指 8086 处理器所开创的指令集体系。为了弥补 8086 在进行浮点运算时的不足，Intel 与 1980 年设计了 8087 数学协处理器，并且为 X86 体系推出了第一个浮点格式 IEEE754。8087 提供两个基本的 32/64bit 浮点资料形态和额外的扩展 80bit 内部支援来改进复杂运算之精度。除此之外，8087 还提供一个 80/17bit 封装 BCD（二进制编码之十进制）格式以及 16/32/64bit 整数资料形态。

CPU 架构，从大的层面（接受和处理信号的方式）分两类——CISC、RISC。

它们的区别在于不同的 CPU 设计理念和方法。CISC 是复杂指令集计算机，目前专指 x86 和 x86-64 两类。其中 x86 又叫 IA32，即 Intel Architecture 32（Intel32 位架构）。x86-64 又叫 AMD64，是 AMD 在 IA32 的基础上扩展出来的一套 64 位 CPU 架构。RISC 是精简指令集计算机，其中比较常用的有 IBM 的 PowerPC 架构，MIPS 的 MIPS 架构，SUN 的 UltraSPARC 架构等等，这儿我就只说 MIPS 架构了。ARM(Advanced RISC Machine)架构是进阶简单指令机器。

早期的 CPU 全部是 CISC 架构，它的设计目的是要用最少的机器语言指令来完成所需的计算任务。比如对于乘法运算，在 CISC 架构的 CPU 上，您可能需要这样一条指令：MUL ADDRA, ADDRb 就可以将 ADDRA 和 ADDRb 中的数相乘并将结果储存在 ADDRA 中。将 ADDRA, ADDRb 中的数据读入寄存器，相乘和将结果写回内存的操作全部依赖于 CPU 中设计的逻辑来实现。这种架构会增加 CPU 结构的复杂性和对 CPU 工艺的要求，但对于编译器的开发十分有利。比如上面的例

子，C 程序中的 `a*=b` 就可以直接编译为一条乘法指令。今天只有 Intel 及其兼容 CPU 还在使用 CISC 架构。

RISC 架构要求软件来指定各个操作步骤。上面的例子如果要在 RISC 架构上实现，将 `ADDRA`，`ADDRB` 中的数据读入寄存器，相乘和将结果写回内存的操作都必须由软件来实现，比如：`MOV A, ADDR A; MOV B, ADDR B; MUL A, B; STR ADDR A, A`。这种架构可以降低 CPU 的复杂性以及允许在同样的工艺水平下生产出功能更强大的 CPU，但对于编译器的设计有更高的要求。

类别	CISC	RISC
指令系统	指令数量很多	较少，通常少于100
执行时间	有些指令执行时间很长，如整块的存储器内容拷贝；或将多个寄存器的内容拷贝到存储器	没有较长执行时间的指令
编码长度	编码长度可变，1-15字节	编码长度固定，通常为4个字节
寻址方式	寻址方式多样	简单寻址
操作	可以对存储器和寄存器进行算术和逻辑操作	只能对寄存器对行算术和逻辑操作，Load/Store体系结构
编译	难以用优化编译器生成高效的目标代码程序	采用优化编译技术，生成高效的目标代码程序

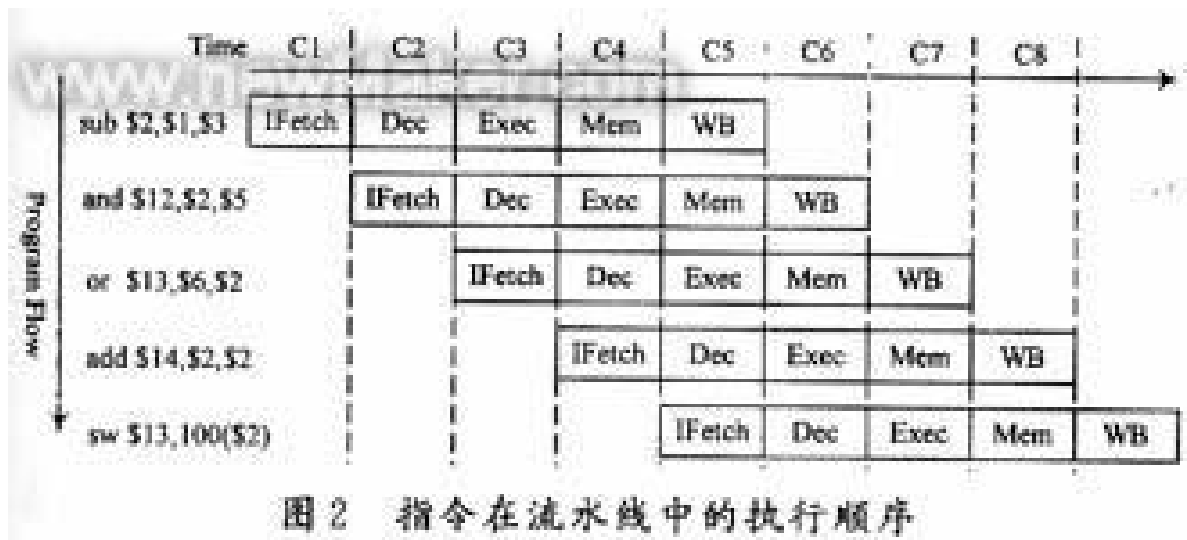
ARM 处理器是 Acorn 有限公司面向低预算市场设计的第一款 RISC 微处理器。全称为 Acorn RISC Machine。ARM 处理器本身是 32 位设计，但也配备 16 位指令集，一般来讲比等价 32 位代码节省达 35%，却能保留 32 位系统的所有优势。在 big.LITTLE 架构里，处理器可以是不同类型的。传统的双核或者四核处理器中包含同样的 2 个核或者 4 个核。一个双核 Atom 处理器中有两个一模一样的核，提供一样的性能，拥有相同的功耗。ARM 通过 big.LITTLE 向移动设备推出了异构计算。这意味着处理器中的核可以有不同的性能和功耗。当设备正常运行时，使用低功耗核，而当你运行一款复杂的游戏时，使用的是高性能的核。为了更快地执行指令，这些流水线可以被设计成允许指令们不按照程序的顺序被执行（乱序执行）。一些巧妙的逻辑结构可以判断下一条指令是否依赖于当前的指令执行的结果。Inter 和 ARM 都提供乱序执行逻辑结构，可想

而知，这种结构十分的复杂，复杂意味着更多的功耗。

此外，ARM 具有其与 X86 架构电脑不可对比的优势，该优势就是：功耗。其实它们的功耗主要是由这几点决定的。首先，功耗和工艺制程相关。ARM 的处理器不管是哪家主要是靠台积电等专业制造商生产的，而 Intel 是由自己的工厂制造的。一般来说后者比前者的工艺领先一代，也就是 2-3 年。如果同样的设计，造出来的处理器应该是 Intel 的更紧凑，比如一个是 22 纳米，一个是 28 纳米，同样功能肯定是 22 纳米的耗电更少。ARM 的处理器有个特点，就是乱序执行能力不如 X86。换句话说，就是用户在使用电脑的时候，他的操作是随机的，无法预测的，造成了指令也无法预测。X86 为了增强对这种情况下的处理能力，加强了乱序指令的执行。此外，X86 还增强了单核的多线程能力。这样做的缺点就是，无法很有效的关闭和恢复处理器子模块，因为一旦关闭，恢复起来就很慢，从而造成低性能。为了保持高性能，就不得不让大部分的模块都保持开启，并且时钟也保持切换。这样做的直接后果就是耗电高。而 ARM 的指令强在确定次序的执行，并且依靠多核而不是单核多线程来执行。这样容易保持子模块和时钟信号的关闭，显然就更省电。

现在 ARM 架构已经具备了进入服务器芯片的能力，众多芯片研发企业纷纷采用 ARM 架构研发服务器芯片无疑将促进其繁荣，2015 年一款采用 ARM 架构的 Windows 10 平板现身，这也是目前曝光的全球首款非 X86 架构、运行 Windows 系统的平板产品。

MIPS 体系结构首先是一种 RISC 架构。MIPS32 架构中有 32 个通用寄存器，其中 \$0 (无论你怎么设置，这个寄存器中保存的数据都是 0) 和 \$31 (保存函数调用 jal 的返回地址) 有着特殊的用途，其它的寄存器可作为通用寄存器用于任何一条指令中。MIPS 架构中没有 X86 中的 PC (程序计数) 寄存器，它的程序计数器不是一个寄存器。因为在 MIPS 这样具有流水线结构的 CPU 中，程序计数器在同一时刻可以有多个给定的值，MIPS32 中不同于其它的 RISC 架构的地方是其有整数乘法部件，这个部件使用两个特殊的寄存器 HI、LO，并且提供相应的指令 mfhi/mthi, mthi/mtlo 来实现整数乘法结果—hi/lo 寄存器与通用寄存器之间的数据交换



在传统的单核时代，提升处理器性能的唯一手段就是提高频率。但受限于物理工艺，频率不能无限提高（例如散热问题等）。对多核处理器来说，可利用的空间增多，散热问题就比较容易解决。这就是 multiprocessor 诞生的背景。

（1）SMT，同时多线程 Simultaneous multithreading，简称 SMT。

SMT 可通过复制处理器上的结构状态，让同一个处理器上的多个线程同步执行并共享处理器的执行资源，可最大限度地实现宽发射、乱序的超标量处理，提高处理器运算部件的利用率，缓和由于数据相关或 Cache 未命中带来的访问内存延时。当没有多个线程可用时，SMT 处理器几乎和传统的宽发射超标量处理器一样。SMT 最具吸引力的是只需小规模改变处理器核心的设计，几乎不用增加额外的成本就可以显著地提升效能。多线程技术则可以为高速的运算核心准备更多的待处理数据，减少运算核心的闲置时间。这对于桌面低端系统来说无疑十分具有吸引力。Intel 从 3.06GHz Pentium 4 开始，所有处理器都将支持 SMT 技术。Intel 的 hyper-threading 其实就是 two-thread SMT。

（2）CMP，片上多处理器（Chip multiprocessors，简称 CMP）

CMP 是由美国斯坦福大学提出的，其思想是将大规模并行处理器中的 SMP（对称多处理器）集成到同一芯片内，各个处理器并行执行不同的进程。与 CMP 比较，SMT 处理器结构的灵活性比较突出。但是，当半导体工艺进入 0.18 微米以后，线延时已经超过了门延迟，要求微处理器的设计通过划分许多

规模更小、局部性更好的基本单元结构来进行。相比之下，由于 CMP 结构已经被划分成多个处理器核来设计，每个核都比较简单，有利于优化设计，因此更有发展前途。目前，IBM 的 Power 4 芯片和 Sun 的 MAJC5200 芯片都采用了 CMP 结构。多核处理器可以在处理器内部共享缓存，提高缓存利用率，同时简化多处理器系统设计的复杂度。

(3) SMP，对称多处理器 (Symmetric Multi-Processors，简称 SMP)

是指在一个计算机上汇集了一组处理器(多 CPU),各 CPU 之间共享内存子系统以及总线结构。在这种技术的支持下，一个服务器系统可以同时运行多个处理器，并共享内存和其他的主机资源。像双至强，也就是我们所说的二路，这是在对称处理器系统中最常见的一种（至强 MP 可以支持到四路，AMD Opteron 可以支持 1-8 路）。也有少数是 16 路的。但是一般来讲，SMP 结构的机器可扩展性较差，很难做到 100 个以上多处理器，常规的一般是 8 个到 16 个，不过这对于多数的用户来说已经够用了。在高性能服务器和工作站级主板架构中最为常见，像 UNIX 服务器可支持最多 256 个 CPU 的系统，其实 qemu 从代码设计上也是最大支持 256 个 virtual cpu。