**复杂指令集与精简指令集**

两种主要的计算机处理器体系结构：

CISC（Complex Instruction Set Computer,复杂指令集计算机）

RISC（Reduced Instruction Set Computer,精简指令集计算机）

* 1
* 2
* 3

**CISC（复杂指令集）**  
即 冯·诺依曼结构（普林斯顿结构），指令与数据存储在同一存储器中；  
采用CISC结构的处理器，指令线与数据线分时复用；  
程序指令存储地址与数据存储地址指向同一个存储器的不同物理位置，则程序指令和数据的宽度相同；  
取指令与取数据不能同时进行，速度受限；  
Intel 8051、Motorola MC68xxx、Atmel AT89

**RISC（精简指令集）**  
即 哈佛结构，指令与数据存储于两个不同的存储空间；  
程序存储器与数据存储器相互独立，独立编址，独立访问；  
分离的程序总线与数据总线在一个机器周期中，可同时获得指令字和操作数，提高执行效率；  
取指令和取数据同时进行，且一般指令线宽与数据线，可包含更多的处理信息；  
Motorola/IBM PowerPC、Atmel AVR、Microchip PIC、ARM

**未来是属于 ARM 为代表的精简指令集还是 x86 为代表的复杂指令集？**  
转载地址：<https://www.zhihu.com/question/20783321>  
分析一：

这里简单来谈一下，ARM和X86之间为什么不太具有可比性的问题。要搞清楚这个问题首先要明白什么是架构，之前也有很多人提到了架构不同，但架构是什么意思？它是一个比较抽象的概念，不太容易用几句话就解释清楚。我们要明白CPU是一个执行部件，它之所以能执行，也是因为人们在里面制作了执行各种功能的硬件电路，然后再用一定的逻辑让它按照一定的顺序工作，这样就能完成人们给它的任务。也就是说，如果把CPU看作一个人，首先它要有正常的工作能力（既执行能力），然后又有足够的逻辑能力（能明白做事的顺序），最后还要听的懂别人的话（既指令集），才能正常工作。而这些集中在一起就构成了所谓的“架构”，它可以理解为一套“工具”、“方法”和“规范”的\*\*。不同的架构之间，工具可能不同，方法可能不同，规范也可能不同，这也造成了它们之间的不兼容——你给一个意大利泥瓦匠看一份中文写成的烹饪指南，他当然不知道应该干什么了。  
如果还看不懂，没关系，我们继续。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”。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。我们可以继续举个例子，比如说我们要命令一个人吃饭，那么我们应该怎么命令呢？我们可以直接对他下达“吃饭”的命令，也可以命令他“先拿勺子，然后舀起一勺饭，然后张嘴，然后送到嘴里，最后咽下去”。从这里可以看到，对于命令别人做事这样一件事情，不同的人有不同的理解，有人认为，如果我首先给接受命令的人以足够的训练，让他掌握各种复杂技能（即在硬件中实现对应的复杂功能），那么以后就可以用非常简单的命令让他去做很复杂的事情——比如只要说一句“吃饭”，他就会吃饭。但是也有人认为这样会让事情变的太复杂，毕竟接受命令的人要做的事情很复杂，如果你这时候想让他吃菜怎么办？难道继续训练他吃菜的方法？我们为什么不可以把事情分为许多非常基本的步骤，这样只需要接受命令的人懂得很少的基本技能，就可以完成同样的工作，无非是下达命令的人稍微累一点——比如现在我要他吃菜，只需要把刚刚吃饭命令里的“舀起一勺饭”改成“舀起一勺菜”，问题就解决了，多么简单。  
这就是“复杂指令集”和“精简指令集”的逻辑区别。可能有人说，明显是精简指令集好啊，但是我们不好去判断它们之间到底谁好谁坏，因为目前他们两种指令集都在蓬勃发展，而且都很成功——X86是复杂指令集（CISC）的代表，而ARM则是精简指令集（RISC）的代表，甚至ARM的名字就直接表明了它的技术：Advanced RISC Machine——高级RISC机。  
到了这里你就应该明白为什么RISC和CISC之间不好直接比较性能了，因为它们之间的设计思路差异太大。这样的思路导致了CISC和RISC分道扬镳——前者更加专注于高性能但同时高功耗的实现，而后者则专注于小尺寸低功耗领域。实际上也有很多事情CISC更加合适，而另外一些事情则是RISC更加合适，比如在执行高密度的运算任务的时候CISC就更具备优势，而在执行简单重复劳动的时候RISC就能占到上风，比如假设我们是在举办吃饭大赛，那么CISC只需要不停的喊“吃饭吃饭吃饭”就行了，而RISC则要一遍一遍重复吃饭流程，负责喊话的人如果嘴巴不够快（即内存带宽不够大），那么RISC就很难吃的过CISC。但是如果我们只是要两个人把饭舀出来，那么CISC就麻烦得多，因为CISC里没有这么简单的舀饭动作，而RISC就只需要不停喊“舀饭舀饭舀饭”就OK。  
这就是CISC和RISC之间的区别。但是在实际情况中问题要比这复杂许许多多，因为各个阵营的设计者都想要提升自家架构的性能。这里面最普遍的就是所谓的“发射”概念。什么叫发射？发射就是同时可以执行多少指令的意思，例如双发射就意味着CPU可以同时拾取两条指令，三发射则自然就是三条了。现代高级处理器已经很少有单发射的实现，例如Cortex A8和A9都是双发射的RISC，而Cortex A15则是三发射。ATOM是双发射CISC，Core系列甚至做到了四发射——这个方面大家倒是不相上下，但是不要忘了CISC的指令更加复杂，也就意味着指令更加强大，还是吃饭的例子，CISC只需要1个指令，而RISC需要5个，那么在内存带宽相同的情况下，CISC能达到的性能是要超过RISC的（就吃饭而言是5倍），而实际中CISC的Core  
i处理器内存带宽已经超过了100GB/s，而ARM还在为10GB/s而苦苦奋斗，一个更加吃带宽的架构，带宽却只有别人的十分之一，性能自然会受到非常大的制约。为什么说ARM和X86不好比，这也是很重要的一个原因，因为不同的应用对带宽需求是不同的。一旦遇到带宽瓶颈，哪怕ARM处理器已经达到了很高的运算性能，实际上根本发挥不出来，自然也就会落败了。  
说到这儿大家应该也已经明白CISC和RISC的区别和特色了。简而言之，CISC实际上是以增加处理器本身复杂度作为代价，去换取更高的性能，而RISC则是将复杂度交给了编译器，牺牲了程序大小和指令带宽，换取了简单和低功耗的硬件实现。但如果事情就这样发展下去，为了提升性能，CISC的处理器将越来越大，而RISC需要的内存带宽则会突破天际，这都是受到技术限制的。所以进十多年来，关于CISC和RISC的区分已经慢慢的在模糊，例如自P6体系（即Pentium Pro）以来，作为CISC代表的X86架构引入了微码概念，与此对应的，处理器内部也增加了所谓的译码器，负责将传统的CISC指令“拆包”为更加短小的微码（uOPs）。一条CISC指令进来以后，会被译码器拆分为数量不等的微码，然后送入处理器的执行管线——这实际上可以理解为RISC内核+CISC解码器。而RISC也引入了指令集这个就逻辑角度而言非常不精简的东西，来增加运算性能。正常而言，一条X86指令会被拆解为2～4个uOPs，平均来看就是3个，因此同样的指令密度下，目前X86的实际指令执行能力应该大约是ARM的3倍左右。不过不要忘了这是基于“同样指令密度”下的一个假设，实际上X86可以达到的指令密度是十倍甚至百倍于ARM的。  
最后一个需要考虑的地方就是指令集。这个东西的引入，是为了加速处理器在某些特定应用上性能而设计的，已经有了几十年的历史了。而实际上在目前的应用环境内，起到决定作用的很多时候是指令集而不是CPU核心。X86架构的强大，很多时候也源于指令集的强大，比如我们知道的ATOM，虽然它的X86核心非常羸弱，但是由于它支持SSE3，在很多时候性能甚至可以超过核心性能远远强大于它的Pentium M，这就是指令集的威力。目前X86指令集已经从MMX，发展到了SSE，AVX，而ARM依然还只有简单而基础的NEON。它们之间不成比例的差距造成了实际应用中成百上千倍的性能落差，例如即便是现今最强大的ARM内核依然还在为软解1080p  
H.264而奋斗，但一颗普通的中端Core i处理器却可以用接近十倍播放速度的速度去压缩1080p H.264视频。至少在这点上，说PC处理器的性能百倍于ARM是无可辩驳的，而实际中这样的例子比比皆是。这也是为什么我在之前说平均下来ARM只有X86几十分之一的性能的原因。  
打了这么多字，其实就是为了说明一点，虽然现在ARM很强大，但它距离X86还是非常遥远，并没有因为这几年的进步而缩短，实际上反而在被更快的拉大。毕竟它们设计的出发点不一样，因此根本不具备多少可比性，X86无法做到ARM的功耗，而ARM也无法做到X86的性能。这也是为什么ATOM一直以来都不成功的原因所在——Intel试图用自己的短处去和别人的长处对抗，结果自然是不太好的，要不是Intel拥有这个星球上最先进的半导体工艺，ATOM根本都不可能出现。而ARM如果尝试去和X86拼性能，那结果自然也好不到哪儿去，原因刚刚也解释过了。不过这也不意味着ARM以后就只能占据低端，毕竟任何架构都有其优点，一旦有应用针对其进行优化，那么就可以扬长避短。X86的繁荣也正是因为整个世界的资源都针对它进行了优化所致。只要能为ARM找到合适的应用与适合的领域，未来ARM也未必不可以进入更高的层次。

分析二：

ARM vs. x86 不是一个技术问题，而是一个商业问题。为什么这么说呢？且听我细细分解：

首先，「x86 比 ARM 功耗高」这一点，在事实上已经不再成立了，具体的数据和分析可以参考 [1]。文章比较长，数据也很多，如果没空看也没关系，看看最后一页的这张表就行了：  
这里比较的是两个平板设备：采用 NVIDIA Tegra 3 (40nm ARM 处理器) 的 Microsoft Surface RT 平板电脑和采用 Intel Clover Trail Atom (32nm x86 处理器) 的 Acer W510 平板电脑。可以看到，每一个测试项目中 Atom 的功耗都比 Tegra 3 低不少。这还不是最要命的，更要命的是  
Atom 的性能比 Tegra 3 高，也就是说 Atom 可以在更短的时间内完成同样的任务然后进入低功耗休眠状态以节省电力，而且在休眠状态下 Atom 的功耗也只有 Tegra 3 的差不多一半。

当然，Atom 并不能完全代表 x86 处理器，Tegra 3 也只是诸多 ARM 处理器中比较费电的一款，但这个例子的教育意义在于它打破了传统上认为「ARM 比 x86 省电」的迷思：没错，在对性能要求不高的情况下，ARM 的确比 x86 省电，因为 ARM 天生就是为低功耗这一目标打造的；但随着应用的不断深入，人们对移动设备的性能要求越来越高，目前在高端平板【或者说普通笔记本替代品】这个尺度的应用上以 Atom 为代表的 x86 处理器已经具有相当的功耗优势。假以时日，在给定性能基础上 x86 做到比 ARM  
功耗更低并不存在太大技术上的障碍。

其次，处理器功耗只是整个设备功耗的一部分，甚至不是最主要的部分，特别是在触屏移动设备上还有屏幕、无线通讯模块这样的耗能大户。比如从上面那张表可以看出，处理器的功耗占总功耗的比例，在绝大多数情况下都不超过 20%，实际使用场合下的功耗占比更是会低于 10%【处理器大部分时候在休眠】。技术的进步可以相对快速的降低处理器的功耗，但由于物理规律的限制【比如屏幕始终需要维持足够的亮度、无线通讯始终是需要发射足够强的电磁波】，设备上的其他耗能大户功耗降低的幅度要小得多。这就意味着单纯削减处理器功耗以延长续航时间的边际效用会越来越小，到一定程度后对整个设备的续航时间的影响变得微乎其微、花大成本提高处理器功耗的回报太低以至于不经济。Intel  
的下一代平台 Haswell 之所以要强调全平台的低功耗也部分是出于这个原因。当然目前我们还没有到那个时候，现在还不用操心这个，但要记住这一点。

————我是技术／商业分割线————

好了，如果现在你和我一样相信 Intel 有能力在功耗和性能上都战胜 ARM，为什么现在绝大部分移动设备采用的是 ARM 而不是 x86 处理器呢？或者说为什么 Intel 会在移动市场上如此「失败」呢？

因为 ARM 代表的产业模式对于 Intel 而言是一场商业灾难。

Intel 这么多年习惯的模式是生产制造几十、几百美元的处理器，在这个价位上的生产制造毛利率高得吓人。而正是丰厚的毛利率使得 Intel 敢于付出高昂的代价研发下一代处理器技术和生产线制程，从而保持领先竞争对手至少一个代际的技术优势。比如现在 Intel 的主流 Core 系列的制程是 22nm，同时期的 ARM 还处于 40nm 到 32nm 或者 32nm 到 28nm 的过渡阶段，而 Intel 已经在新建 14nm 的生产线了。

反观一片 ARM 处理器才卖几美元，利润率微薄，即便出货量大不少【不要忘了出货量和生产线投入之间的准线性关系】，却也不能满足 Intel 这样的巨鳄【巨饿？XD】的胃口，更不能让它保持目前的「高研发／高毛利相互驱动」的商业模式。不玩自家的游戏规则，Intel 沦为二三流厂商那是分分钟可能发生的危险。Intel 管理层不希望看到这个未来，Intel 的股东们更不希望。

另外，移动 SoC 市场中公司之间的合作模式也不适合 Intel。为了节省制造成本和降低功耗，移动 SoC 经常需要集成多家厂商的 IP 块于一片上，那么两个问题就随之而来了：1) Intel 会不会授权自己的 IP 块给其他厂商，比如苹果？这可不是 Intel 擅长的，因为授权设计的收入比销售最终产品的收入要低得多；2) 最终生产的时候用谁的生产线？Intel 可不会白白的让出自己投入巨资研发制造的最新制程的生产线给别人用；就算 Intel 愿意，别人也不见得用得起，毕竟有低毛利率的限制。而 ARM 则没有这些顾虑：ARM  
设计和生产是分离的，设计的 IP 块可以单独授权给各家厂商自行定制整合，而制造采用的是比较成熟的生产线，成本低、可选厂家也多。

这就是为什么 Intel 在移动领域涉足很晚也很勉强、为什么功耗低、单价更低的 Atom 处理器始终比最先进的 Core 处理器落后一两代的原因。非不能也，乃不为也；因何不为？利益使然尔！

然而市场现实是台式机、笔记本的出货量稳步下降，Intel 寄予厚望的 Ultrabook 不成气候也没能让笔记本市场死灰复燃；与此同时，移动设备份额稳步飙升，ARM 处理器的性能在多数场合都能满足需求【good enough】，甚至有威胁 Intel 在服务器市场的潜能。这一切都迫使 Intel 不得不面对 ARM 这个难题；而这个难题的主要症结不在于技术，而在于 Intel 能不能否定自我、推倒重来。