

RISC 与 CISC 的纠结

黄博文

RISC（精简指令集计算机）与 CISC（复杂指令集计算机）两大架构设计哲学的争斗已经成为大家耳熟能详的历史，但是 RISC 的由来，在学术界和工业界以外却很少见到有人提及。本文基于伯克利 RISC 项目领导者之一 David Patterson 的口述自传，以及 ACM 数据库的公开文献整理，向大家介绍 RISC 从发明到广为流传的那段故事。

三十年前的论战

“我们认为，基于 RISC 理念设计的处理器只有在极少数情况下慢于 CISC 处理器……过多的指令使得 CISC 处理器的控制逻辑复杂……研发成本上升……编译器也不知道该如何利用这么复杂的指令集……CISC 的设计思路应当反思。”—— RISC 的早期倡导者之一，David Patterson

“RISC 与 CISC 的区别缺乏明确定义，而且 RISC 缺乏有力实验证明其宣称的优势，仅停留在纸面的设计是不够的，我们在 VAX 架构的设计中发现很多与 RISC 理念相反的地方……实验数据证明 RISC 的出发点有误……”—— VAX CISC 架构的设计者代表，Douglas W. Clark 和 William D. Strecker.

很难相信，观点如此背道而驰的两篇文章，竟然同时刊登在美国计算机学会旗下的同一期《计算机体系结构通讯》杂志上。但事实上，正是论战双方的私下联络以及杂志编辑的有意安排，使得这场 1980 年的论战得以见报，于是留下了可供后人追溯的足够史料。

ACM 数据库收录的影印文档中保留下的点点墨迹，似乎象征着这场论战中四溅的火星。在 1980 年前后，几乎所有的新处理器设计都在按照 CISC 的路线发展，不断加入新的指令，使用微码控制，试图在指令集架构层面对高层编程语言提供更直接的支持，这种发展路线使得硬件研发成本提高，研发周期变长，编译器也不知道该如何利用越来越复杂的指令集。

来自 IBM 研究院的 John Cocke 首先意识到，更加精简清爽的指令集设计将有助于减少硬件开发难度和成本，同时也有利于编译器进行代码优化工作，于是在他领导下的 IBM 801 项目第一次对 RISC 的概念进行了实践。随后加州大学伯克利分校，斯坦福大学的几位科学家也逐渐认清 CISC 的弱点，开始反其道而行之，着手进行新的设计。那时刚刚博士毕业四年，在伯克利任教的年轻老师 David Patterson 就是其中一员，他决定在研究生课程中检验自己的想法，让学生们试着构建一个指令精简化的微处理器作为大作业。在微处理器流片成功之前，David 就撰写了文章发表，描述自己构建处理器的新方法，认为这种精简设计将降低了硬件设计成本，缩短了开发周期，且方便编译器进行代码生成，达到了更高的性能，结果引起争论和质疑，于是有了这场载入体系结构发展史的论战。

反败为胜

当时站在 David 对立面的人包括当时深受尊敬的 VAX CISC 架构设计者 Douglas W. Clark 和 William D. Strecker，David 在文章中多次转述 VAX 的工程经验，试图证明 RISC 的优势，但 VAX 架构设计者们的现身说法使得局面对 David 非常不利，他们以自己的第一手数据将 David 文章中宣称的 RISC 优势逐一驳倒，更加雪上加霜的是，第一组学生们流片回来的处理器，并未能体现出具备说服力的速度优势，于是进一步引发了嘲笑。所幸第二组学生进行的设计较为成功，于是 David 和学生们的成果得以登陆 1983 年国际固态电子电路大会（ISSCC）进行展示，这个会议只接受流片成功的芯片设计投稿，因此门槛较高，同时也因为成果卓越，受到学术界和工业界的广泛关注，是集成电路领域的顶级学术会议。

David 在这里打了一个漂亮的翻身仗, 尽管制造工艺是老旧的 MOSIS, 主频比 Intel 同期制造的处理器 80286 慢上几乎一半, 晶体管数量也只有其三分之一, 但是更加清爽的新式设计在编译器等其他工具的辅助下竟然成功将 Intel 踩在脚下! ISSCC 大会现场所有的大牌人物都目睹了这一历史性时刻, 业界哗然。RISC 提倡简化指令集设计, 固定指令长度, 统一指令编码格式, 加速常用指令, 在当时来看与当时占据主流的 CISC 哲学颇有些背道而驰的意味, RISC 和 David Patterson 早年受到质疑和攻击也就不难理解。但有了流片成功的芯片与性能测试结果在手, 加之 1983 年的 ISSCC 大会上聚集了几位与 David Patterson 观点相同的支持者, RISC 流派已经开始占据上风。

风靡业界

出于兼容性的考虑, David 和学生们设计的芯片从未流入商业市场售卖, 但是在 David 等人的推动和宣传下, 这一设计理念随后如星火燎原般扩散, 一大批公司开始采用这种理念设计新处理器, 包括后来几近统一武林的 Intel。而当初提倡 RISC 的先行者们也纷纷功成名就, John Cocke 在 1987 年将计算机科学领域的最高奖项“图灵奖”收入囊中, 而 David Patterson 在芯片设计与计算机体系结构领域也已经与“泰山北斗”划上等号。

RISC 的设计理念催生的一系列新架构中包含了许多我们耳熟能详的名字, 包括学术上认为比较成功的 DEC Alpha, 后来写入经典教科书的 MIPS, 以线程级并行度绕过指令级并行度障碍的 SUN SPARC, 以及现在统治嵌入式市场的 ARM。这些雨后春笋般涌现的 RISC 处理器将 CISC vs RISC 的世纪之战推向更高潮, 精简指令集和复杂指令级划分为两个阵营, 争论不断。以 x86 为代表的复杂指令集, 其指令编码格式混乱, 导致编码器复杂, 流水线设计较为困难, 指令不定长也带来指令对齐方面的额外挑战。而 RISC 指令集的编码格式相对整洁, 流水线设计容易, 但是由于指令定长, 导致一些比较大的常数/地址必须拆散才能加载, 代码密度不够高, 某些情况下会浪费指令缓存的容量和带宽, 因此两种指令集都不是不败金身, 本质上都属于双刃剑。

在计算机体系结构还不是完全成熟的时候, RISC 的速度优势看上去更加具有吸引力, 连 Intel 都举棋不定, 只得两条腿走路, 在继续更新 CISC 产品线的同时, 也推出了 i860 系列 RISC 处理器, 安迪·格鲁夫当时表示, CISC 处理器是 Intel 一直在做的产品线, 兼容所有软件, 而 RISC 处理器速度更快, 但没有什么软件能在它上面运行。连 Intel 都不知道未来发展方向会是在 RISC 还是 CISC, 只好两头下注。在风头最劲的时候, RISC 处理器一度占据在服务器市场占据统治地位。当时的争论与工业实践中提炼出的 RISC 优势, 直到现在仍被许多人谈论, 但是, 历史的进步无情地碾碎了这一切。

融为一炉

强大的 Intel 在 90 年代中期的 P6 架构里开始引入乱序多发射技术, 这帮助 Intel 第一次在服务器市场上拿出了性能上可以与 RISC 比肩的处理器, 并且第一次实现了 CISC 指令集在解码阶段上向 RISC 类指令的转化, 将后端流水线转换成类 RISC 的形式, 弥补了 CISC 流水线实现上的劣势。许多人认为这是 Intel 在向 RISC 指令集学习, 是在事实上宣布了 RISC 的胜利, 可是 ARM 也于同期引入了代码密度更高的 Thumb 新指令集, 力图提高指令缓存等劣势项目上的效率, 这表明 RISC 也在向 CISC 取经, 双方都在相互取长补短。而 RISC 风格的设计的确能输出速度优势, 这在一部分人心中催生了错误的 RISC 优越论。实质上随着体系结构和微电子技术的进步, CISC 在架构上的所谓劣势逐渐缩小, 乃至消失, 而 RISC 阵营却在 Intel 的猛攻下节节败退, 时至今日, Intel 的服务器 CPU 占据了 95% 的市场份额, RISC 的优越论也逐渐偃旗息鼓。

今年的国际高性能计算机体系结构大会上, 来自美国威斯康辛大学的一个研究小组做了一个测量分析报告, 根据他们披露的测试数据, CISC 与 RISC 在指令集架构层面上的差异已经较小, 由先进的微架构和物理设计、工艺实现带来的改进足以掩盖指令集架构层面的劣势, CISC vs RISC 的世纪之战实际上没有胜负, 双方的精华已经融为一炉。结果这个报告受到一些学者的批评, 有意思的是, 批评的原因并不是在于报告本身结论错误, 而是因为在这个报告所得出的结论, 其实是

大家早都已经认可的共识，没有必要再在大会上宣读！

如何评价一个架构设计理念是好是坏呢？笔者认为，应当是后人们看来理所当然的——“为什么不这样做呢？还好当初没有放弃这条路！”今天我们谈到 RISC 留下的遗产时，已经很难找到负面评价的理由，RISC 不再是缺乏支持的异类，而是现代计算机体系结构的基本组成部分之一，没有人再对 RISC 存在的必要性提出怀疑，这大概就是对它最好的认可。