论文阅读报告

1. 微处理器概要

1.1 微处理器概要

微处理器由一片或几片大规模集成电路组成的中央处理器。这些电路执行控制部件和算术逻辑部件的功能。微处理器能完成取指令、执行指令，以及与外界存储器和逻辑部件交换信息等操作，是微型计算机的运算控制部分。微处理器不仅是微型计算机的核心部件，也是各种数字化智能设备的关键部件。可见微处理器的发展是计算机发展的一个重要的部分。

1.2 微处理器设计概要

在微处理器设计领域，或者说在整个计算机体系结构的设计中，就是一个权衡的过程。计算机结构设计师的工作就是对设计进行各种权衡，这种方式贯穿了微处理器发展的始终。

在微处理器的设计过程中就是综合各种功能元件并将它们组合在一起工作，而这些元件之间往往对于空间资源、能量资源有着各种竞争关系，为了保证整体性能满足要求，必须在各个元件之间进行权衡。同时，很难说存在一个标准的设计格式能够让设计出的微处理器在处理各种任务的时候都能获得最优的性能，因此还需要在特定需求之间进行一定的权衡。

1.3 微处理器发展方向

简单来说，微处理器需要做的就是处理指令，在这过程中进行的就是三件事：取指令，取数据，处理指令。因此微处理器的很多发展就是集中于解决这三部分任务。

1.3.1 取指令

处理器目前一次能取出的指令越发多，如果能取到充分多的指令，处理器的资源就可以被有效利用。但是有三个阻碍妨碍了这一过程：1）指令缓存不命中，2）取指令中断，3）条件分支预测错误。

1.3.2 取数据

取数据最理想的情况是，任意时刻都有需要的数据，但是处理器处理的速度远比访存的速度要快，且这个差距在不断拉大，虽然由于摩尔定律在近年来有逐渐不可靠的趋势，处理器发展速度略有放缓，但是与存储器之间的性能差已经巨大，形成了巨大的剪刀差（见图 1）。这种剪刀差也就意味着在目前的计算机上处理器的处理速度是远比缓存要快的。因此目前的主流方式是使用多级 Cache 存储器的方式减少这一过程的耗时。

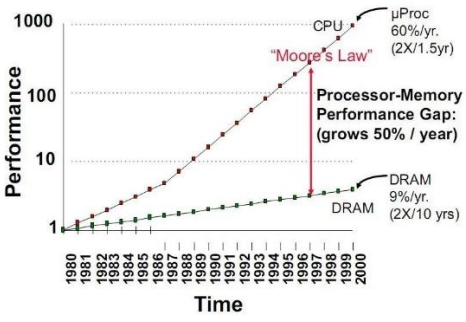


图 1 处理器与存储器性能

1.3.3 指令执行

在指令执行过程中可能存在数据依赖问题，即一个单元的输入需要另一个单元的输出，随着处理器时钟周期的逐渐缩短，这个延迟逐渐变得麻烦。

2. 基于论文的新技术探索

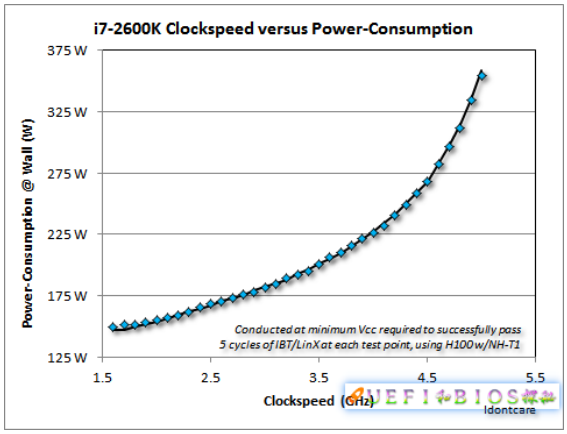
2.1 晶体管、频率

Yale Patt 教授的论文中预测在 2008 年左右 CPU 晶体管数目将达到 10 亿，CPU 频率将达到 6GHz，事实上以 Intel 公司推出的 Core i7 芯片为例，2010 年推出的 i7-980X 晶体管数量达到了 11.7 亿，目前面向中高端用户的 CPU 晶体管数量也大多在 10 亿以上。但是观察目前主流的芯片，可以发现芯片的频率没有达到过 6GHz，而是基本上止步于 4GHz，原因分析如下：

芯片主频停留在 4GHz 或者更低原因主要在于散热，提高主频超过一定范围后，热密度急速提高，很不经济，也造成散热困难。2004 年时，Intel 宣布其奔腾 4 将会发布 4GHz 主频的 CPU，但是最终止步于 3.8GHz，接下来数年芯片主频停留在了这一级别，直到 Haswell 微架构的 CPU 发布之后才真正将主频提升上了 4GHz。

对于 CPU 来说，每一个晶体管的翻转过程中都需要耗费能量，在晶体管充放电的过程中存在一定量的能量消耗，直观上来说，CPU 的主频越高这一消耗的能量必然更高，通过能耗的公式：$P = CV^2 f$ 可以发现能耗与频率是呈正相关的，但是事实上不仅仅是线性关系。

在这里需要引入门延迟的概念，简单来说，组成 CPU 的晶体管充放电需要一定的时间，这个时间就是门延迟，只有在充放电完成之后采样才能保证信号的完整，而这个时间和电压负相关，也就是说电压越高充放电时间越短。当我们不断提高频率的时候就存在过了某个节点之后太快的电平翻转导致门延迟跟不上进而影响了数字信号的完整性，这时候我们就需要提高电压。因此主频事实上与能耗事实上可以近似为一个三次方的关系（见图二），因此当主频超过某个数值之后当主频上升能耗将急速上升，不符合权衡的观点。



图二 i7 芯片能耗图

因此如果希望解决主频难以提高的问题，首要任务是需要降低功耗，目前一种解决方式是使用更小更高效的二维石墨烯材料。2016 年已有科学家设计出一种超低功耗、且最终有望将处理器主频提高至惊人的 100GHz 的石墨烯晶体管[1]。传统晶体管允许电子被一个能量源所激发，跳过能量壁垒并切换到另一状态。尽管这种方法工作得挺好，但在能效上却难以大幅提升。而莫斯科物理技术学院的科学家们已经找到了一种方法来提高隧穿电流，从而使

得隧道晶体管距离实际运用更近了一步，因借助了量子隧穿效应来跳过能量势垒，其操作时的能耗要比标准晶体管少一些，根据论文作者之一 Svintsov 表示：“功率低，电子部件的温度也低，这意味着我们可以让芯片运行在极高的频率——不是 GHz 级别的提升、而是数十上百倍”。但是这一技术距离真正实际商用距离还相当遥远，可以预见的是如果没有更突破性的进展发生，在相当长一段时间内 CPU 的主频并不会有相当大幅的提升。

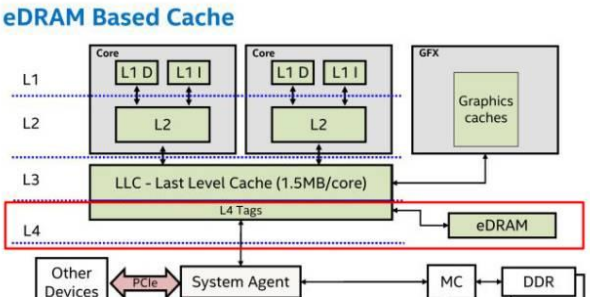
2.2 分支预测技术

处理器流水线的引入大幅度提高了处理器的处理速度与处理效率，但是随之而来的就是分支预测的问题，简单来说就是当代码中遇到一个分支指令，如果分支预测错误，那么在开始处理这条代码到判断分支预测错误之前进入流水线的所有操作都是需要丢弃的，这样就大幅度浪费了处理器的性能，因此需要分支预测技术在预测分支的执行方向。

伴随着指令集并行为代表的超标量处理器的成熟，传统分支预测技术也日趋成熟，因此目前出现对于新的分支预测器的研究，其中神经网络分支器成为目前的一个研究热点。罗格斯大学的丹尼尔.A.希梅内斯最早在 2001 年于 HPAC 会议上将神经网络引入了分支预测领域[2],文章研究表示对于 4K 的硬件开销，使用 SPEC CPU2000 测试程序，神经网络分支无预测率比 Gshare 降低了 26%，比组合分支预测降低了 12%。随着神经网络研究的逐渐深入，逐渐出现了一些针对难预测分支的预测器[3]，引入了 CNN 等在神经网络领域性能优良的神经网络结构。

2.3 片上缓存

在 Yale Patt 的论文中提出在他的时代大多数高性能微处理器都有两个级别的缓存。在当下的主流 CPU 中都出现了三级 Cache，其主要功能是可以进一步降低内存延迟，同时提升大数据量计算时处理器的性能。同时需要注意的是 L2 缓存和 Patt 时代也略有不同，patt 时代的主流中高端微处理器是奔腾系列的处理器，在奔腾 D 处理器中其采用的是双核模式，因此 L1 和 L2 是分别被两个核占用的，也就是说事实上就是两个核各自有一个片上缓存。但是在 Intel 退出 Core 系列处理器之后发生了巨大的变化，L2Cache 变为多核共享模式，而 L1Cache 被每个核单独占有。而目前由于出现了 L3Cache，因此 L2 缓存被每个核单独占据，而 L3Cache 在逻辑上是共享模式。同时，在目前 Intel 的一种新型架构——Haswell 中，一块 DRAM 被加入 CPU 中，被叫做 eDRAM，它在平时可以做显存。也可以被设定为 L4 缓存：



图三 eDRAM 示意图

在服务器领域增加 L3 缓存在性能方面有显著的提升。具有较大 L3 缓存的处理器提供更有效的文件系统缓存行为及较短消息和处理器队列长度。L3 缓存已被部分应用于中高端桌面级处理器中，比如 Intel 的酷睿 i 系列和奔腾 G 系列,AMD 的羿龙 2 全部配备了大小不一的 L3 缓存，这一举措标志着 L3 缓存的应用越来越广泛，CPU 的运算效率和响应速度也将进一步提升。

2.4 乱序处理

在计算机工程领域，乱序执行是一种应用在高性能微处理器中来利用指令周期以避免特定类型的延迟消耗的范式。在 20 世纪 90 年代末出现了乱序执行，CPU 硬件本身在读取指令后重新安排指令的执行顺序。以目前 Intel Core i7 使用的 Nehalem 微架构为例，其代表了当下比较新的中高端微型处理器的性能指标。

在 Nehalem 微架构中乱序引擎显著的扩大了，除了性能原因，还有就是为了提供 SMT，因为 SMT 需要资源共享。Nehalem 的 ROB（重排序缓冲）从 96 项增加到 128 项，RS（保留站）从 32 项增加到 36 项，它们都由两个线程所共享，但是使用不同的策略。ROB 是静态分配给 2 个线程，使得 2 个线程在指令流里都可以预测得一样远。而 RS 则是竞争共享，基于各线程的需求。这是因为许多时候一个线程可能会中止，从内存等待操作数，而使用到很少的 RS 项。这样就不如让另一个更活跃的线程尽可能多地使用 RS 项。在 RS 中的指令当其所有操作数都准备好时，就被分配到执行单元。[4]

事实上 Nehalem 执行单元与 Core2 相比并没有太大的变化，乱序处理的基本框架改变也不大，主要改变就是提高了使用率。

2.5 片上多处理器（CMP）

Yatt 教授的思路是将晶体管的增加全用于 CPU 的单核性能的提升上，在当时也许这是一种看起来可行的方式，但是从目前主流 CPU 的结构上来看，基本上都采用了多核的形式，从 Intel 和 AMD 等大公司的行为推测，发展片上多处理器是比单独发展一个处理器是更有优势的。

拿双核和单核比较，双核的优势不是频率，而是同时处理多件事情。一个核心理论上同时只能干一件事，但是显然在日常使用过程中基本上不可能在同一段实践中只在计算机上打开一个程序，基本上都是多个任务同时打开的，这时候就需要操作系统将每个处理任务划分为多份，分块执行，这时候如果单核的处理器想要达到让客户感觉不到等待的情况，就需要极大提高 CPU 的主频，但是这一点如上文讨论的，就需要克服能耗的问题，因此目前主流观点是单核基本无法支撑当下的 CPU 使用。而如果使用多核 CPU，那么每一个核可以单独处理一个任务，每一个核的主频也许不高，但是实践证明处理的效果是值得发展的。

2.7 可重构的逻辑器件

Yatt 教授的论文中预测了可重构逻辑器件在 CPU 上集成的发展，目前这一方向的研究是微处理器领域的一个方向。可重构芯片的出现打通了“应用定义软件、软件定义芯片”进而实现“应用定义芯片”这一人们长期追求的通道，而广泛的适应性也使其成为替代专用集成电路、可编程器件和经典处理器的有力竞争者。

国内也有团队正在这一方向钻研，2016 年，清华大学、澜起科技与 Intel 合作联手研发融合可重构计算和英特尔 x86 架构技术的新型通用 CPU。其采用的是 CPU+ASIC 的形式进行可重构形式。可重构计算处理器负责实现安全，澜起科技就是把安全向下做到硬件这一级，理论上会检查所有的指令，检查 X86 内核运行时行为是否与预期一致，若一致则执行，如果不符合预期定义的动作都不让执行。而运算功能则由 Intel 做的 X86 内核负责。

根据 Yatt 的预测，CPU 上将会集成 FPGA 的部分，但是直至今日，大部分集成了可重构的微处理器基本上集中于研究 CPU+ASIC 的形式。本人的一点猜测是目前的可重构需求并不需要大幅度的重构灵活性，因此使用 ASIC 能带来更低的能耗。

3. 对未来发展的预测

3.1 GPU 的发展

在当下谈到计算机，不可避免的一个部分就是 GPU，可以说当下很火的机器学习和深度学习，如果不是 GPU 产生了突破性进展，很有可能发展的方向会有所改变。因此未来 CPU 和 GPU 的交互必将更加深入也更能提高程序运行的速度，如果 GPU 普及达到了一定程度，同时与 CPU 交换数据也能达到一定的速率，那么是否 CPU 所有的并行计算的任务可以全部交给 GPU 完成，而 CPU 主要负责完成一些比较复杂的调度以及一些不适合并行计算的部分，这二者之间的数据交互也是未来发展的一个方向，这样分工也许能带来较大的速度突破。

3.2 能耗方面突破

目前如果想要突破能耗的限制有两种方式：

第一种就是使用更高效的散热方式，目前如果搭配液氮散热，搭配一些现在的技术，实际超频之后 CPU 的主频是可以达到 8-9GHz 的，但是如果单纯使用液氮降温肯定不可能达到推广的目标的。也许目前在高效的散热方面可以考虑仿生学的散热。在人体内每一个细胞都在进行氧化还原反应，但是人的体温是恒定在 36.5℃左右的，也就是说生物是有独特的散热形式的，因此也许未来散热形式可以参考生物体内的散热，甚至可以使用液体为晶体管供电的同时带走热量，就像生物体内的血液既给细胞提供氧气，又带走了细胞排出的无用产物。

第二种就是使用新材料，这一点在 2.1 略有涉及。

3.3 新型计算机

新型计算机有很多，包括 DNA 计算机、光电子计算机、量子计算机等。和前二者比起来量子计算机的前景是比较光明的。量子计算机是一类遵循量子力学规律进行高速数学和逻辑运算、存储及处理量子信息的物理装置。从当前的量子计算机发展情况来看，尽管很多人质疑量子计算的数据都是使用单一的计算同类似刷榜的方式产出的，并没有很大的实际意义，但是不可否认的是如果量子计算机可以做出技术性革新，以其当前展示的潜力，传统的计算机的速度极限也许将会被轻松突破。

3.4 小型化计算机

可以预见的是计算机的外形和尺寸大小将随着不同的对象和环境而变化。和前几个预期比起来这一点是在未来一段时间内最可能发生蓬勃发展的，甚至在现在已经正在发展。特别是嵌入计算机，可以遍布生活中的各个场景，使用者可以利用随身携带的信息操作器具不需要使用多余的连接操作就可以接收消息，真正形成万物互联的场面。

4. 参考文献

[1]Svintsov D, Devizorova Z, Otsuji T, et al. Plasmons in tunnel-coupled graphene layers: Backward waves with quantum cascade gain[J]. Physical Review B, 2016, 94(11): 115301.

[2] Jiménez D A, Lin C. Dynamic branch prediction with perceptrons[C]//Proceedings HPCA Seventh International Symposium on High-Performance Computer Architecture. IEEE, 2001: 197-206.

[3] S·J·塔沙, G·凯斯金, G·N·什雅,等. 用于难预测分支的预测器:, CN109213524A[P]. 2019.

[4]https://baike.baidu.com/item/%E4%B9%B1%E5%BA%8F%E6%89%A7%E8%A1%8C%E6%8A%80%E6%9C%AF/5923755?fr=aladdin#6