

# 大小核架构——ARM 处理器的多核设计方法

吴同

计算机科学与技术学院

Email: wutongcs@zju.edu.cn

**摘要：**大小核架构是一种同时使用两种类型核心的异构体系结构。小核旨在实现能效最大化，而大核则旨在提供优越的计算性能。当前，智能手机和平板电脑等移动设备的电池和能耗的改进速度跟不上性能需求的增长速度。ARM 的大小核设计为解决这一矛盾提供了有效的解决方案。

**关键词：**多核处理器 ARM 大小核 多核调度

## 1 架构介绍

ARM 大小核架构 (big.LITTLE) 是由 ARM Holdings 开发的一种异构计算体系结构，将相对省电但速度较慢的小核 (LITTLE) 与性能较强但耗电较大的内核 (big) 结合在一起。所有核心都可以访问相同的内存，因此可以在大小核之间实时切换。这一架构的目的是设计一种多核处理器，既可以更好地适应动态计算需求，又比单独使用时钟缩放技术的耗电更少。

big.LITTLE 最初在 2011 年 10 月与 Cortex-A7 一起发布，后者可以与 Cortex-A15 兼容。2012 年 10 月，ARM 宣布推出 Cortex-A53 和 Cortex-A57 (ARMv8-A) 内核，二者相互兼容，从而可以在 big.LITTLE 芯片中使用。随后，在 Computex 2013 上，ARM 发布了 Cortex-A12，后又在 2014 年 2 月发布了 Cortex-A17，Cortex-A12 和 Cortex-A17 都可以与 Cortex-A7 在 big.LITTLE 芯片上配对使用。

2017 年 5 月，ARM 推出以 big.LITTLE 为基础的新一代架构 DynamIQ。与 big.LITTLE 相比，它将单一集群中的最大内核数增加到 8 个，并最多支持 32 个集群。该技术还为每个内核的电压控制提供了更精细的粒度，并提供了更快的 L2 缓存访问速度。DynamIQ 有望在多核处理器设计上提供更大的灵活性和可扩展性。

## 2 硬件设计

在 big.LITTLE 架构中，大小核在的指令集都是相同的。以 ARM Cortex-A15 和 ARM Cortex-A7 为例，二者都实现了完整的 ARM v7A 架构，包括虚拟化和大型物理地址扩展。因此，所有的指令在 Cortex-A15 和 Cortex-A7 上执行时，其运算结果都是相同的，只有性能差别所带来的执行计算的时间不同。

大核（如 Cortex-A15）和小核（如 Cortex-A7）之间的差异主要体现在微架构中。Cortex-A7 是有序、双发射处理器，其流水线长度在八级到十级之间（图2-1）。而 Cortex-A15 是乱序、三发射（超标量）处理器，其流水线长度在十五级到二十四级之间（图2-2）。

执行一条指令的时间和耗能都与流水线的长度有关。A15 与 A7 的性能差距主要来自于流水线长度，而功耗的差距也来自于流水线长度。一般来说，流水线越长，指令间可重叠的部分越多，CPU 性能越高。但更长的流水线意味着更复杂的电路实现，这会导致发热量的增大。Cortex-A15 与 Cortex-A7 在流水线架构设计上采取了不同的策略。Cortex-A15 偏重性能而牺牲功耗，而 Cortex-A7 会牺牲性能来提高能源的利用率。为了实现两种架构的互补，大小核架构中二者可共享同一个 L2 级缓存。

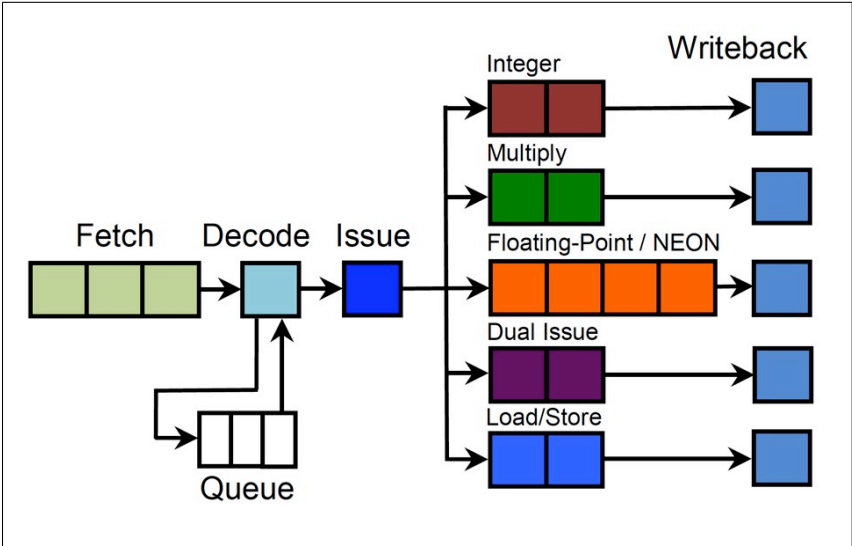


图 2-1 ARM Cortex-A7 流水线结构

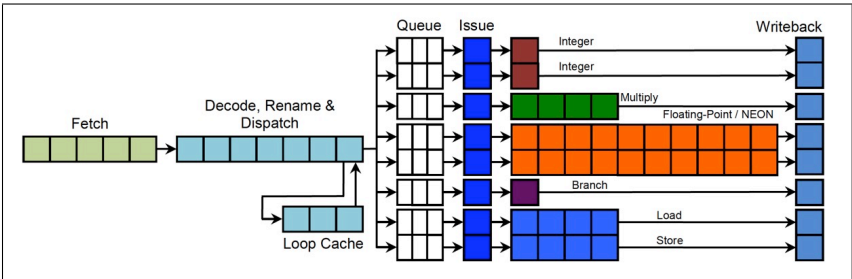


图 2-2 ARM Cortex-A15 流水线结构

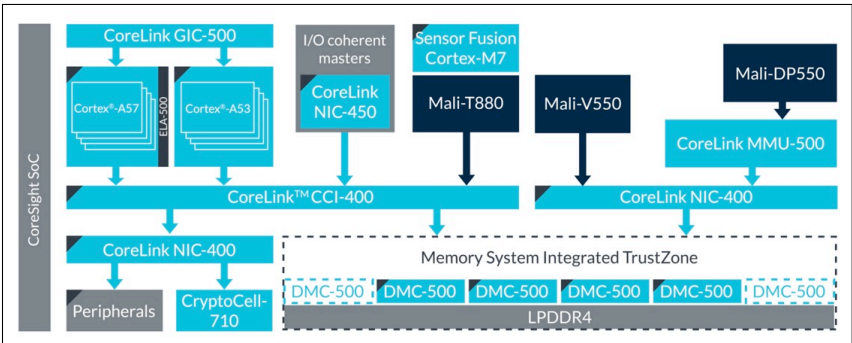


图 2-3 大小核架构通信结构

在通信方面，为使不同处理器之间能够协调一致，大小核架构中采用了 CoreLink CCI 总线，既确保了大核和小核之间的一致性，也确保了与其他组件（如 GPU）的 IO 一致性。CCI 是一种交叉矩阵形的总线，具有可配置、无阻塞性、低延迟、低功耗的特点，可以串联处理器内部各个重要的部分（图2-3）。

初代的 CCI-400 总线采用了图2-4所示的通信架构：处理器组 C1，发一个包含地址信息的特殊读写的命令到总线，然后总线把这个命令转给另一个处理器组 C2。C2 收到请求后，根据地址逐步查找二级和一级缓存，如果发现自己也有，那么就返回数据或者做相应的缓存一致性操作，即监听。在这个过程中，被请求的 C2 中的处理器核心并不参与这个过程，所有的工作由缓存和总线接口单元（BIU）来做。

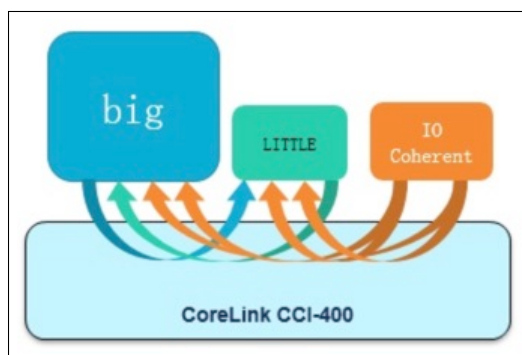


图 2-4 CCI-400 通信架构

后一代的 CCI-500 总线设计则有所不同。如图2-5所示，总线里加入了一个监听过滤器，实际也是一块缓存，将所有处理器组内部的一级二级缓存的状态信息都存放在里面。这样设计的优点是，监听请求不必发送到各组处理器，在总线内部就可以完成，不仅节省时钟周期，也能降低功耗。其代价则是增加了缓存的面积。并且，如果监听过滤器里的某行缓存被替换，同样的操作必须在对应处理器组的一二级缓存也都做一遍，以保持一致性。这个过程会添加额外的负担，因为在更新一二级缓存的同时，监听过滤器本身也需要追踪更新的状态，否则就无法保证一致性。

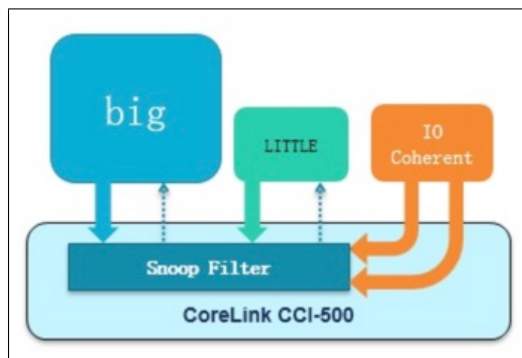


图 2-5 CCI-500 通信架构

在中断方面，大小核架构中 Cortex-A15 和 Cortex-A7 共享一个通用中断控制器 GIC-400（图2-6）。多达 480 个中断可以分配给 Cortex-A15 和 Cortex-A7，并且可编程的中断控制器 GIC-400 允许在 Cortex-A15 或 Cortex-A7 集群中的任何核心之间迁移中断。

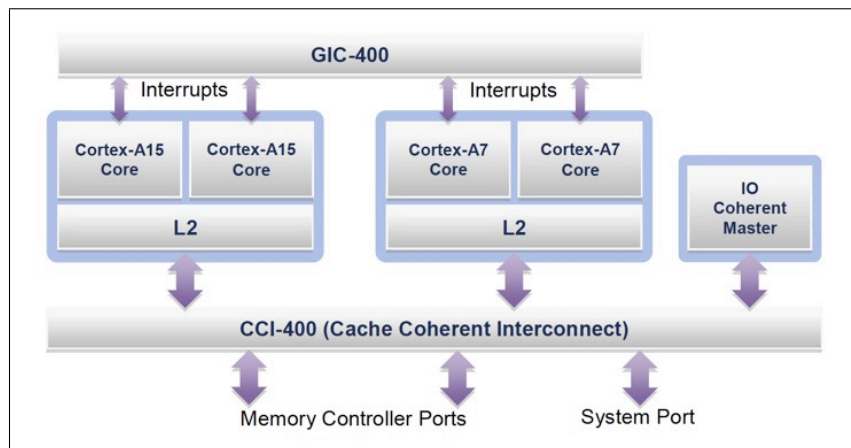


图 2-6 中断架构

### 3 调度算法

早期的大小核架构上的软件模型采取 CPU 迁移（CPU migration）调度算法，即程序可以在核间切换，但不能同时运行所有的核。最新的调度策略——全局任务调度（Global Task Scheduling）能同时使用所有的核，直接控制各核间的线程分配。

在 CPU 迁移算法中，一个任务不会同时使用两个核心（图3-7）。操作系统会比较当前任务所需要的性能与当前使用处理器集群的性能，并设置一个阈值。例如，当前任务在 Cortex-A7 执行时，操作系统将会计算出当前平台处理的最高峰值。一旦 Cortex-A7 处于最高峰值，且还需要更多的资源，操作系统将会连同应用程序移动到 Cortex-A15 上运行。任务在两个集群之间迁移的时间为不超过两万个时钟周期，在 1GHz 的系统中，可控制在 20ms 内。

对于全局任务调度算法，Cortex-A7 和 Cortex-A15 可以同时通电并同时执行代码（图3-8）。这个模型被称为 big.LITTLE MP (Multi-Processing)，它本质上是异构的多核处理 (Multi-Processing)。在这种模型中，Cortex-A7 是一直在线的，只有当需要更多的计算资源时，Cortex-A15 才启动并与 Cortex-A7 同时执行。big.LITTLE MP 最大的优势是各个线程都能在最合适的处理资源上执行。对于计算密集的线程，通常可以分配给 Cortex-A15 执行。而对于大量处理 I/O 或者实时性要求不高的线程，都可分配给 Cortex-A7 执行。

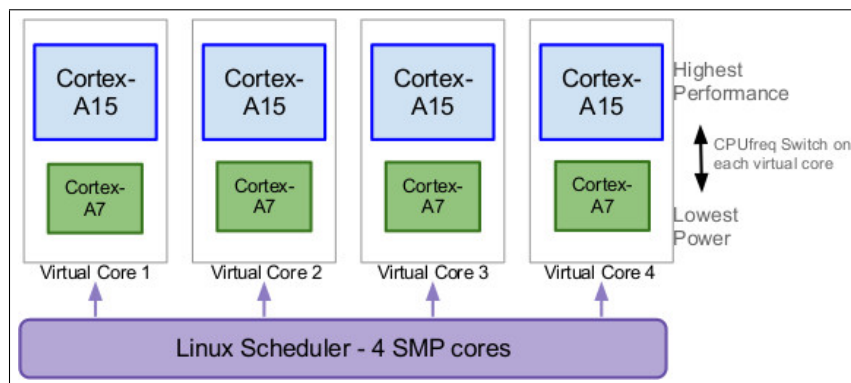


图 3-7 CPU 迁移算法

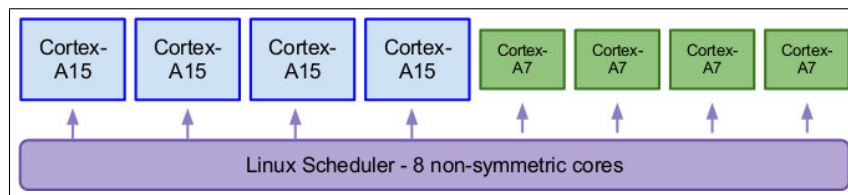


图 3-8 全局任务调度算法

## 4 应用实例

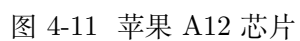
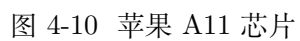
在现今的移动设备的 SoC 上，大小核架构已经被广泛采用。本文以苹果公司的芯片为例，列举若干这一架构的应用实例。

苹果公司于 2016 年 9 月 7 日发布的搭载于 iPhone 7 和 iPhone 7 Plus 上的 A10 芯片是其首次使用大小核架构设计的芯片，其调度模式还是 CPU 迁移模型。A10 中有两个大核和两个小核，面积比为 5.31（图4-9）。一年后的 A11 芯片上则采用了两个大核 + 四个小核的六核设计，并采用 HMP 全局任务调度使得六个核心可以同时运行（图4-10），大小核面积比为 5.06。A12 芯片依旧采用了两个大核 + 四个小核的六核设计，大小核面积比为 4.81（图4-11）。最新的 A13 芯片上，还是采用了两个大核 + 四个小核的六核设计，大小核面积比降为 4.5（图4-12）。



图 4-9 苹果 A10 芯片





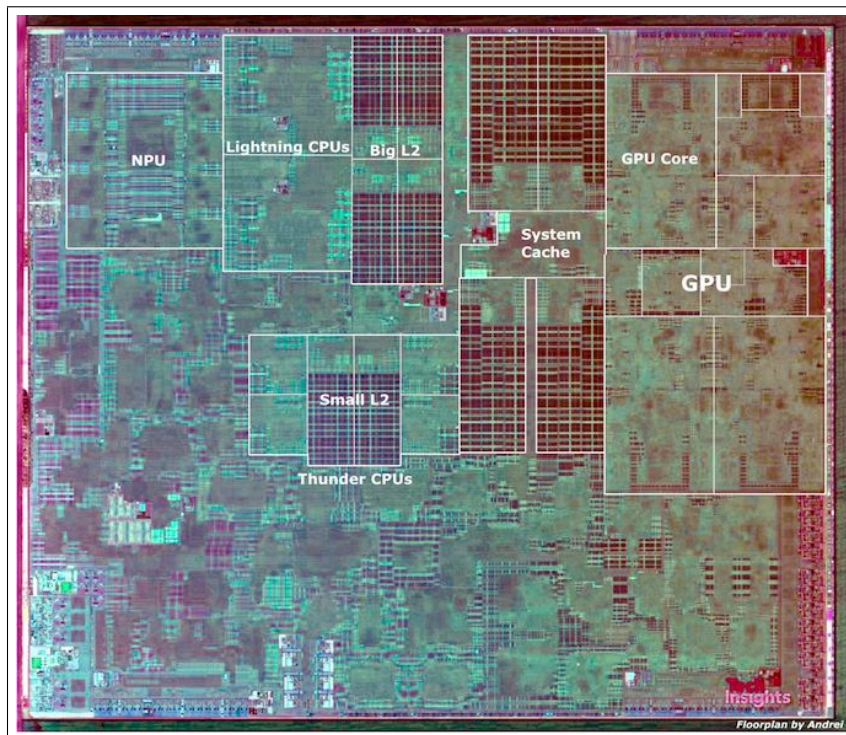


图 4-12 苹果 A13 芯片

## 5 英特尔的模仿

与 ARM 相比，X86 架构过大的功耗导致在移动处理器上的发展并不顺利。为取得性能与功耗之间的平衡，英特尔也吸收了大小核设计的思想，将使用 10nm 制程生产一款代号为 Lakefield 的处理器。目前，英特尔已有强大的高性能 X86 核心，也有针对低功耗平台的 Atom 核心。英特尔将在 Lakefield 处理器上，将使用 Icelake 架构的大核和 Atom 架构的小核。由于英特尔 10nm 制程量产的一次次推迟，导致原本应在 2019 年就发布的这一架构至今仍未发布。受 2020 年新冠疫情的影响，未来何时能正式发布更是没有定数。但英特尔这一架构设计理念的转变以及近十年来苹果、高通等公司的成功实践，标志着大小核架构于移动处理器设计的科学性和有效性。自从计算机体系结构进入多核化以来，性能、功耗、面积三者的矛盾持续加大。大小核设计的理念，为三者之间的平衡找到了一条切实可行的解决途径。

## 6 心得体会

通过写作本文时的查询资料，我了解了当今实际使用的手机、平板上处理器的体系结构，也解决了我的一些疑惑。比如，在使用手机时，有省电模式、平衡模式、性能模式的选择。这三种模式的实现实际是在大小核调度策略中所设置的阈值不同。在省电模式中，程序更多运行在小核上。而在性能模式中，调度策略则比较激进，程序更多运行在大核上。平衡模式则采用较为正常的调度阈值。

这篇论文的写作过程，让我得以对计算机体系结构和操作系统等课程上学到的理论知识是如何应用在最新的日常使用的移动设备中有了更多的认识。同时，通过对这些知识的了解，也让我在平时使用各种电子设备时，让其先进的设计理念得以体现。