ARM，Advanced RISC Machine，英国的一家公司，及其推出的RISC体系架构处理器家族，适用于移动通信领域。

ARM本身不生产销售芯片，而是销售自己的设计授权。客户可以选择：

* IP Core授权：直接购买ARM设计的核心，并基于此设计SoC
* 指令集架构授权：基于某版本的架构自己设计核心，并基于此设计SoC。

来清晰一下**架构**的概念。

在谈到体系结构中，有若干种/层分类方式，非常绕人，其中有很多都可以被叫做“架构（architecture）”，从ARM这个体系结构本身，到最终手机/移动设备上的主板。但其实这些架构的概念，是有概念上和层次上的区分的。目前并没有一个通用的名词列表去定义这些概念，以下的名词定义和解释主要基于ARM官网和维基百科。

注意：

1. 请注意ARMv7和ARM7的区别，前者是指令集版本（或者称为ARM架构版本），后者是指一个核心家族或者核心（Core Family或者Core）
2. 核心的划分被分为两个阶段，ARMv6之前官方核心是按照数字划分的（ARM1-ARM11），ARMv7之后被称为Cortex核心

下面是具体内容：

* **RISC**和**CISC架构**，这是CPU的一个二分类方式，RISC（Reduced Instruction Set Computing，精简指令计算机）的代表就是ARM，此外还有MIPS、PowerPC、RISC-V、SPARC等。CISC（Complex Instruction Set Computing，复杂指令计算机）的代表则是x86体系结构，此外还有Motorola 68000、PDP-10、System Z等。
* 各种体系结构，比如**ARM体系结构（ARM Architecture）**，这里指整个ARM处理器家族
* **ARM架构（ARM Architecture）**划分，也被称为**指令集版本**或者**指令集架构**，目前版本从ARMv1到ARMv8。
  + ARMv6之前的部分**指令集版本**又有不同的子版本，比如ARMv5有ARMv5TE、ARMv5TEJ两个子版本。
  + **Profile**，在ARMv6之前并没有做此种划分，因此被归为Classic，ARMv7开始（以及ARMv6-M）开始将**ARM架构**分为三种，即A（Application），R（Real-time），M（Microcontroller），比如ARMv6-M，ARMv7-M，ARMv7-A，ARMv8-R，ARMv8-A等等。
* **核心家族（Core Family）**，处理器核心的划分，分为官方（ARM Holdings）和第三方（比如三星、高通等）。
  + 官方核心的划分在ARMv6之前跟指令集版本部分对应，也有一定的交错（比如ARM7，ARM9、ARM11）。
    - 根据不同的配置又有细分子类，比如ARM7可分为ARM7、ARM7T、ARM7EJ三个子类
      * T表示支持Thumb指令
      * J表示支持Jazzele
      * D表示支持JTAG Debug
      * M表示支持fast Multiplier
      * I表示支持enhanced ICE
      * S表示Synthesizable
  + ARMv7之后则按照Profile分为Cortex-A、Cortex-R、Cortex-M。
  + 第三方设计的Core指的是其他公司在拿到指令集架构授权之后自己设计的Core，苹果、三星、高通都有做第三方设计
* **片上系统（SoC）**，这个才是真正能在手机主板上看到的封装好的芯片，功能类似PC中的主板，其中主处理器即是上条中的Core。和x86架构的PC不一样，考虑到便携性，手机等移动设备的集成度更高，一个SoC里不仅包含了处理器，还有很多外设比如DMA控制器、LCD控制器、内存控制器、摄像头接口、Flash等等。

以下是详细的指令集版本和不同核心家族，以及所包括的核心列表，可以看到两者之间相关，但也有一定交错。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 指令集版本 | | 官方Core family | | 官方Core | 第三方Core Family | 第三方Core |
| ARMv1 | | ARM1 | | ARM1 |  |  |
| ARMv2 | ARMv2 | ARM2 | | ARM2 |  |  |
| ARMv2a | ARM250 | Amber (Open Source) | Amber23 Amber25 |
| ARM3 | | ARM3 |
| ARMv3 | | ARM6 | | ARM60 ARM600 ARM610 |  |  |
| ARM7 | ARM7 | ARM700 ARM710 ARM710a |  |  |
| ARMv3 | ARMv4T | ARM7T | ARM7TDMI ARM710T ARM720T ARM740T ARM7TDMI-S |  |  |
| ARM9 | ARM9T | ARM9TDMI ARM920T ARM922T ARM940T |  |  |
| SecureCore | | SC100 |  |  |
| ARMv4 | ARM8 | | ARM810 | StrongARM(DEC) | SA-110 SA-1100 |
| Faraday (Faraday) | FA510 FA526 FA626 |
| ARMv5 | ARMv5TE | ARM10 | ARM10E | ARM1020E ARM1022E | FA606TE FA626TE FMP626TE FA726TE |
| ARM9 | ARM9E | ARM946E-S ARM966E-S ARM968E-S ARM996HS | XScale (Intel/Marvell) | XScale Bulverde Monahans |
| ARMv5TEJ | ARM926EJ-S |  |  |
| ARM7 | ARM7EJ | ARM7EJ-S |  |  |
| ARM10 | ARM10E | ARM1026EJ-S |  |  |
| ARMv6 | ARMv6 | ARM11 | | ARM1136J(F)-S |  |  |
| ARMv6T2 | ARM1156T2(F)-S |  |  |
| ARMv6Z | ARM1176JZ(F)-S |  |  |
| ARMv6K | ARM11MPCore |  |  |
| ARMv6-M | SecureCore | | SC000 |  |  |
| Cortex-M | | Cortex-M0 Cortex-M0+ Cortex-M1 |  |  |
| ARMv7 | ARMv7-M | SecureCore | | SC300 |  |  |
| Cortex-M | | Cortex-M3 |  |  |
| ARMv7E-M | Cortex-M4 |  |  |
| Cortex-M7 |  |  |
| ARMv7-R | Cortex-R | | Cortex-R4 Cortex-R5 Cortex-R7 Cortex-R8 |  |  |
| ARMv7-A | Cortex-A | Cortex-A (32bit) | Cortex-A5 Cortex-A7 Cortex-A8 Cortex-A9 Cortex-A12 Cortex-A15 Cortex-A17 | Snapdragon (Qualcomm) | Scorpion Krait |
| Ax (Apple) | A6(Swift) |
| ARMv8 | ARMv8-A | Cortex-A32 | Snapdragon (Qualcomm) | Kryo |
| Cortex-A (64bit) | Cortex-A34 Cortex-A35 Cortex-A53 Cortex-A57 Cortex-A72 Cortex-A73 | Denver (NVidia) | Denver |
| K12 (AMD) | K12 |
| Exynos (Samsung) | M1/M2 (Mongoose) M3(Meerkat) |
| Ax (Apple) | A7(Cyclone) A8(Typhoon) A9(Twister) |
| ARMv8.1-A |  | A10(Hurricane) |
| ARMv8.2-A | Cortex-A55 Cortex-A65E Cortex-A75 Cortex-A76 Cortex-A77 | A11(Monsoon) |
| Neoverse | | NeoverseN1 NeoverseE1 | Exynos (Samsung) | M4(Cheetah) |
| ARMv8.3-A |  | | | Ax (Apple) | A12(Vortex) |
| ARMv8.4-A | A13(Lightning) |
| ARMv8-M | Cortex-M | | Cortex-M23 Cortex-M33 Cortex-M35P |  |  |
| ARMv8-R | Cortex-R | | Cortex-R52 |  |  |

参考：

<https://en.wikipedia.org/wiki/List_of_ARM_microarchitectures>

<https://en.wikipedia.org/wiki/ARM_architecture>