

# ARM Cortex M3 处理器

## 背景

基于从 ARMv6 开始的新设计理念，ARM 进一步扩展了它的 CPU 设计，成果就是 ARMv7 架构闪亮登场。在这个版本中，内核架构首次从单一款式变成了 3 中款式。

(1) 款式 A：设计用于高新能的“开放应用平台”——越来越接近电脑又要快又要实时。

(2) 款式 M：用于深度嵌入式的系统中——本文介绍的主角。

款式 A (ARMv7-A)：需要运行复杂应用的“应用处理器”。支持大型嵌入式操作系统（不一定实时），比如 Symbian（诺基亚智能手机用），Linux，以及微软的 Windows CE 和智能手机操作系统 Windows Mobile。这些应用需要劲爆的处理性能，并且需要硬件 MMU 实现的完整而强大的虚拟内存机制，还基本上会配有 Java 支持，有时还要求一个安全程序执行环境（用于电子商务）。典型的产品包括高端手机和手持仪器，电子钱包以及金融事务处理机。

款式 R (ARMv7-R)：硬实时且高性能的处理器。标的是高端实时市场。那些高级的玩意，像高档轿车的组件，大型发电机控制器，机器手臂控制器等，它们使用的处理器不但要很好很强大，还要极其可靠，对事件的反应也要极其敏捷。

款式 M (ARMv7-M)：认准了旧世代单片机的应用而量身定制。在这些应用中，尤其是对于实时控制系统，低成本、低功耗、极速中断反应以及高处理效率，都是至关重要的。

Cortex 系列是 v7 架构的第一次亮相，其中 Cortex - M3 就是按款式 M 设计的。

## 关于处理器

Cortex-M3 是一款低功耗处理器，具有门数目少，中断延迟短，调试成本低的特点，是为要求有快速中断响应能力的深度嵌入式应用而设计的。该处理器采用 ARMv7-M 架构。

Cortex-M3 处理器整合了以下组件：

处理器内核。这款门数目少，中断延迟短的处理器具备以下特性：

— ARMv7-M：Thumb-2 ISA 子集，包含所有基本的 16 位和 32 位 Thumb-2 指令，用于多媒体，SIMD，E(DSP)和 ARM 系统访问的模块除外。

— 只有分组的 SP

— 硬件除法指令，SDIV 和 UDIV（Thumb-2 指令）

— 处理模式（handler mode）和线程模式（thread mode）

— Thumb 状态和调试状态

— 可中断-可继续（interruptible-continued）的 LDM/STM，PUSH/POP，实现低中断延迟。

— 自动保存和恢复处理器状态，可以实现低延迟地进入和退出中断服务

(ISR)。

- 支持 ARMv6 架构 BE8/LE
- ARMv6 非对齐访问

嵌套向量中断控制器 (NVIC)。它与处理器内核紧密结合实现低延迟中断处理，并具有以下特性：

- 外部中断可配置为 1~240 个
- 优先级位可配置为 3~8 位
- 中断优先级可动态地重新配置
- 优先级分组。分为占先中断等级和非占先中断等级
- 支持末尾连锁 (tail-chaining) 和迟来 (late arrival) 中断。这样，在两个中断之间没有多余的状态保存和状态恢复指令的情况下，使能背对背中断 (back-to-back interrupt) 处理。

— 处理器状态在进入中断时自动保存，中断退出时自动恢复，不需要多余的指令。

存储器保护单元 (MPU)。MPU 功能可选，用于对存储器进行保护，它具有以下特性：

- 8 个存储器区
- 子区禁止功能 (SRD)，实现对存储器区的有效使用。
- 可使能背景区，执行默认的存储器映射属性。

总线接口：

- AHBLite ICode、DCode 和系统总线接口
- APB 专用外设总线 (PPB) 接口
- Bit band 支持，bit-band 的原子写和读访问。
- 存储器访问对齐
- 写缓冲区，用于缓冲写数据。

低成本调试解决方案，具有以下特性：

— 当内核正在运行、被中止、或处于复位状态时，能对系统中包括 Cortex-M3 寄存器组在内的所有存储器和寄存器进行调试访问。

- 串行线 (SW-DP) 或 JTAG (JTAG-DP) 调试访问，或两种都包括。
- Flash 修补和断点单元 (FPB)，实现断点和代码修补。
- 数据观察点和触发单元 (DWT)，实现观察点，触发资源和系统分析 (systemprofiling)
- 仪表跟踪宏单元 (ITM)，支持对 printf 类型的调试。
- 跟踪端口的接口单元 (TPIU)，用来连接跟踪端口分析仪。

## Cortex-M3 的详细框图

CM3 处理器其实是个大礼包，里面除了处理核心外，还有许多其它组件，用于系统管理和调试支持，如图 1 所示：



**CM3Core:** Cortex - M3 处理器的中央处理核心。

**嵌套向量中断控制器 NVIC:** NVIC 是一个在 CM3 中内建的中断控制器。中断的具体路数由芯片厂商定义。NVIC 是与 CPU 紧耦合的, 它还包含了若干个系统控制寄存器。因为 NVIC 支持中断嵌套, 使得在 CM3 上处理嵌套中断时清爽而强大。它还采用了向量中断的机制。在中断发生时, 它会自动取出对应的服务例程入口地址, 并且直接调用, 无需软件判定中断源, 为缩短中断延时做出了非常重要的贡献。

**SysTick 定时器:** 系统滴答定时器是一个非常基本的倒计时定时器, 用于在每隔一定的时间产生一个中断, 即使是系统在睡眠模式下也能工作。它使得 OS 在各 CM3 器件之间的移植中不必修改系统定时器的代码, 移植工作一下子容易多了。SysTick 定时器也是作为 NVIC 的一部分实现的。

**存储器保护单元: MPU** 是一个选配的单元, 有些 CM3 芯片可能没有配备此组件。如果有, 则它可以把存储器分成一些 regions, 并分别予以保护。例如, 它可以某些 regions 在用户级下变成只读, 从而阻止了一些用户程序破坏关键数据。

**BusMatrix:** BusMatrix 是 CM3 内部总线系统的核心。它是一个 AHB 互连的网络, 通过它可以数据在不同的总线之间并行传送——只要两个总线主机不试图访问同一块内存区域。BusMatrix 还提供了附加的数据传送管理设施, 包括一个写缓冲以及一个按位操作的逻辑(位带(bit - band))。

**AHB to APB:** 它是一个总线桥, 用于把若干个 APB 设备连接到 CM3 处理器的有外设总线上(内部的和外部的)。这些 APB 设备常见于调试组件。CM3 还允许芯厂商把附加的 APB 设备挂在这条 APB 总线上, 并通过 APB 接入其外部私有外设线。

以下组件都用于调试, 通常不会在应用程序中使用它们。

**SW - DP/SWJ - DP:** 串行线调试端口 (SW - DP) /串口线 JTAG 调试端口 (SWJ - DP) 都与 AHB 访问端口 (AHB - AP) 协同工作, 以使外部调试器可以发起 AHB 上的数据传送, 从而执行调试活动。在处理器核心的内部没有 JTAG 扫描链, 大多数调试功能都是通过 NVIC 控制下的 AHB 访问来实现的。SWJ - DP 支持 both 串行线协议和 JTAG 协议, 而 SW - DP 只支持串行线协议。

**AHB - AP:** AHB 访问端口通过少量的寄存器, 提供了对全部 CM3 存储器的访问机能。该功能块由 SW - DP/SWJ - DP 通过一个通用调试接口 (DAP) 来控制。当外部调试器需要执行动作的时候, 就要通过 SW - DP/SWJ - DP 来访问 AHB - AP, 从而产生所需的 AHB 数据传送。

**嵌入式跟踪宏单元 ETM:** ETM 用于实现实时指令跟踪, 但它是一个选配件, 所以不是所有的 CM3 产品都具有实时指令跟踪能力。ETM 的控制寄存器是映射到主地址空间上的, 因此调试器可以通过 DAP 来控制它。

**数据观察点及跟踪单元:** 通过 DWT, 可以设置数据观察点。当一个数据地址或数据的值匹配了观察点, 就产生了一次匹配命中事件。匹配命中事件可以用于产生一个观察点事件, 后者能激活调试器以产生数据跟踪信息, 或者让 ETM 联动(以跟踪在哪条指令上发生了匹配命中事件)。

**指令跟踪宏单元 ITM:** ITM 有多种用法。软件可以控制该模块直接把消息送给 TPIU(类似 printf 风格的调试); 还可以让 DWT 匹配命中事件通过 ITM 产生数据跟踪包, 并把它输出到一个跟踪数据流中。

跟踪端口的接口单元 TPIU：TPIU 用于和外部的跟踪硬件（如跟踪端口分析仪）交互。在 CM3 的内部，跟踪信息都被格式化“高级跟踪总线（ATB）包”，TPIU 重新格式化这些数据，从而让外部设备能够捕捉到它们。

FPB：FPB 提供 flash 地址重载和断点功能。Flash 地址重载是指：当 CPU 访问的某条指令匹配到一个特定的 flash 地址时，将把该地址重映射到 SRAM 中指定的位置从而取指后返回的是另外的值。此外，匹配的地址还能用来触发断点事件。Flash 地址重载功能对于测试工作太有用了。例如，通过使用 FPB 来改变程序流程，就可以给那些不能在普通情形下使用的设备添加诊断程序代码（such as adding diagnosis program code to a device that cannot be used in normal situations unless the FPB is used to change the program control.）。

ROM 表：它只是一个简单的查找表，提供了存储器映射信息，这些信息供包括了多种系统设备和调试组件。当调试系统定位各调试组件时，它需要找出相关寄存器在存储器的地址，这些信息由此表给出。绝大多数情况下，因为 CM3 有固定的存储器映射，所以各组件都对号入座——拥有一致的起始地址。但是因为有些组件是可选的，还有些组件是可以由制造商另行添加的，各芯片制造商可能需要定制他们芯片的调试功能。林子大了什么鸟都有，在这种情况下，必须在 ROM 表中给出这些“另类”的信息，这样调试软件才能判定正确的存储器映射，进而可以检测可用的调试组件是何种类型。

## 总线典型的连接方式

由上面的内容及处理器的系统处理方框图可知，CM3 由若干个总线接口，很容易混淆，也不容易弄清楚它们是怎样与其它设备和存储器连接的。这里有个典型连接实例，如图 2 所示：

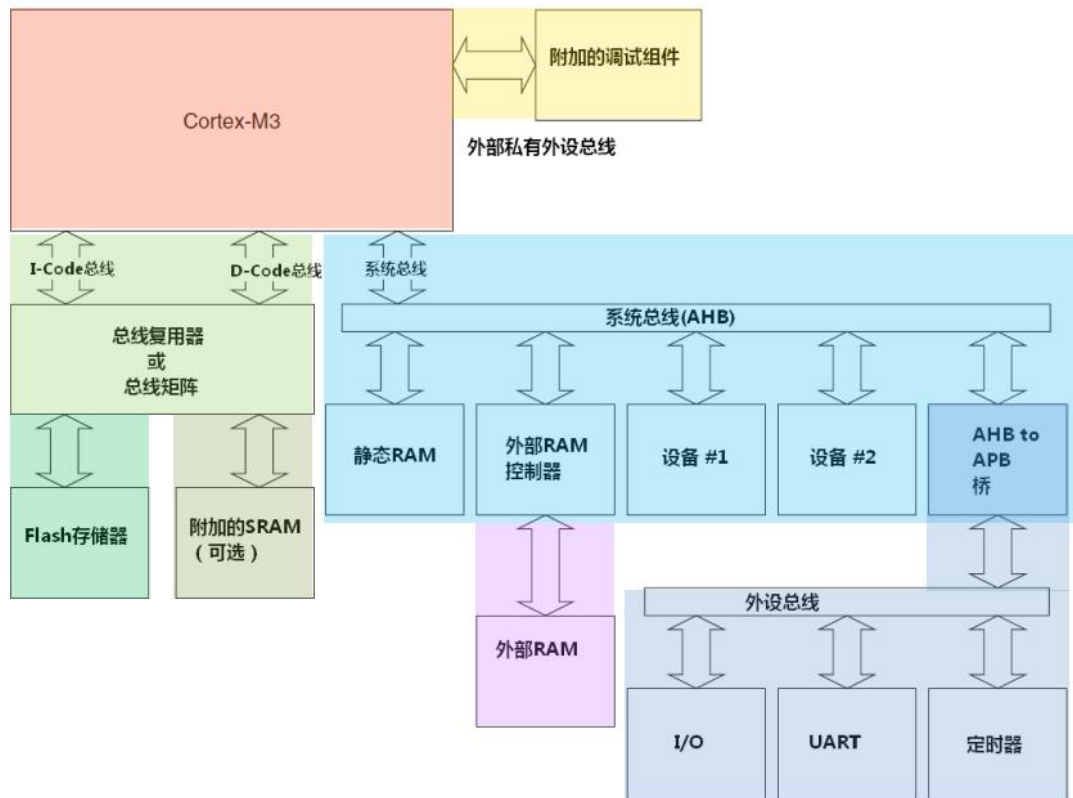


图 2 Cortex-M3 总线连接范例

因为代码存储区既可以由指令总线(I - Code)访问（当从此区取指时），也可以被数据总线（D - Code）访问（当在此区访问数据时），需要在中间插入一个总线开关，称为“总线矩阵[注 1]”；或者使用一个 AHB 总线复用器。如果使用了总线矩阵，则闪存和附加的 SRAM（如果有的话）可以被 both I - Code 和 D - Code 访问。

通过 AHB 总线矩阵把取指和数据访问分开后，如果指令总线和数据总线在同一时刻访问不同的存储器设备（例如，从 flash 中取指的同时从附加的 SRAM 中访问数据），则两者可以并行不悖。但若是只使用了总线复用器，则数据传送就不能同时发生了，然而这时电路尺寸能做得更小。不过，通常的 CM3 单片机设计都使用系统总线来连接 SRAM。而且主 SRAM 确实应该使用系统总线来连接，从而落到 SRAM 存储器的地址区，这样才能利用 CM3 的位带操作能力。

有些脚数比较多的单片机会带外部总线接口（EMI）。这种情况下，需要一个外部存储器控制器，因为 AHB 不接受直接把片外存储器挂在它上面，通常外部存储器控制器也连接到系统总线上。其它的 AHB 设备则可以简单地连接到系统总线上，而不需要额外的总线矩阵。

上图 2 显示出的功能框，像总线矩阵、 AHB - to - APB 总线桥、存储器控制器、 I/O 接口、定时器以及 UART 等，都可以从 ARM 和其它 IP 供应商处取得。不同的 CM3 单片机其片上外设也不同。因此在使用时，你还需要参考器件厂家提供的参考手册。

[注 1]：这里所讲的总线矩阵不是 CM3 内部的总线矩阵，它们是两码事。CM3 内部的总线矩阵是专门设计的，不能作为一个通用的 AHB 开关来使用。