

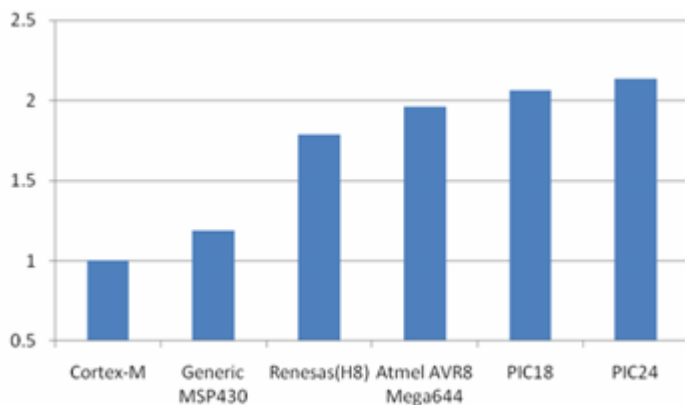
参考 ARM 网站链接:

<http://www.arm.com/zh/products/processors/cortex-m/cortex-m0.php>

从 8/16 位迁移到 ARM Cortex-M

ARM Cortex-M 代码大小优势说明

ARM Cortex-M 处理器提供优于 8 位和 16 位体系结构的代码密度。这在减少对内存的需求和最大限度地提高宝贵的芯片上闪存的使用率方面具有很大的优势。我们将在本部分中分析此优势的原因。



使用相对 EEMBC CoreMark 测试大小的代码大小比较。

指令宽度

认为 8 位微控制器使用 8 位指令，基于 ARM Cortex-M 处理器的微控制器使用 32 位指令，这是一种常见的误解。事实上，PIC18 和 PIC16 指令大小分别是 16 位和 14 位。对于 8051 体系结构，虽然某些指令的长度为 1 字节，但许多其他指令的长度为 2 或 3 字节。通常，对于 16 位体系结构也是如此，其中某些指令可能占用 6 字节或更多内存。

ARM Cortex-M3 和 Cortex-M0 处理器利用可提供极佳代码密度的 ARM Thumb[®]-2 技术。使用 Thumb-2 技术，Cortex-M 处理器可以支持已扩展为包括更强大的 32 位指令的 16 位 Thumb 指令的根本基础。在许多情况下，C 语言编译器将使用 16 位版本的指令，除非使用 32 位版本可以更有效地执行运算。

指令效率

由于没有同时考虑到 ARM Cortex-M 处理器指令更为强大，此图片是不完整的。在许多情况下，单个 Thumb 指令可相当于数个 8/16 位微控制器指令；这意味着 Cortex-M 设备的代码量更少，因此可以以更低的总线速度完成同一任务。

在各处理器体系结构之间比较 16 位乘法运算

8 位示例		16 位示例	ARM Cortex-M
MOV A, XL; 2 字节	MUL AB; 1 字节	MOV R4,&0130h	MULS r0,r1,r0
	ADD A, R1; 1 字节	MOV R5,&0138h	
MOV B, YL; 3 字节	MOV R1, A; 1 字节	MOV SumLo,R6	
MUL AB; 1 字节	MOV A, B; 2 字节	MOV SumHi,R7	
MOV R0, A; 1 字节	ADDC A, R2; 1 字节	(操作数被移入或移出内存映射的硬件乘法单元)	
	MOV R2, A; 1 字节		
MOV R1, B; 3 字节	MOV A, XH; 2 字节		
	MOV B, YH; 3 字节		
MOV A, XL; 2 字节	MUL AB; 1 字节		
MOV B, YH; 3 字节	ADD A, R2; 1 字节		
MUL AB; 1 字节	MOV R2, A; 1 字节		
ADD A, R1; 1 字节	MOV A, B; 2 字节		
MOV R1, A; 1 字节	ADDC A, #0; 2 字节		
	MOV R3, A; 1 字节		
MOV A, B; 2 字节			
ADDC A, #0; 2 字节			
MOV R2, A; 1 字节			
MOV A, XH; 2 字节			
MOV B, YL; 3 字节			

注意: Cortex-M 乘法实际上执行 32 位乘法, 此处我们假设 r0 和 r1 包含 16 位数据。

压缩的数据占用空间

必须注意，Cortex-M 处理器支持 8 位和 16 位数据传输，可以有效利用数据内存。这意味着程序员可以继续使用他们在面向 8/16 位的软件中使用的相同数据类型。

能效优势

对不断增加连接（例如，USB、蓝牙和 IEEE 802.15）、具有复杂模拟传感器（例如，加速计和触摸屏）且成本日益降低的产品需求，已导致需要将模拟设备与数字功能更紧密地集成，以对数据进行预处理和传输。大多数 8 位设备在不显著增加 MHz（并因此不显著增加功率）的情况下不提供支持这些任务的性能，因此嵌入式开发人员需要寻找具有更高级处理器技术的替代设备。16 位设备以前曾被用来解决微控制器应用中的能效问题。但是，16 位设备相对不高的性能意味着它们通常需要较长的活动工作周期或较高的时钟频率才能完成 32 位设备所完成的相同任务。

使软件开发更加容易

基于 ARM Cortex 处理器的微控制器的软件开发可能比 8 位微控制器产品的开发容易得多。Cortex 处理器不但是完全可通过 C 语言进行编程的，而且还附带各种高级调试功能以帮助定位软件中的问题。除 MCU 开发工具包中包括的所有附加资源外，Internet 上还有大量示例和教程，其中许多来自基于 ARM 处理器的 MCU 供应商的网站。