

STM32F4: 高性能的数字信号控制器

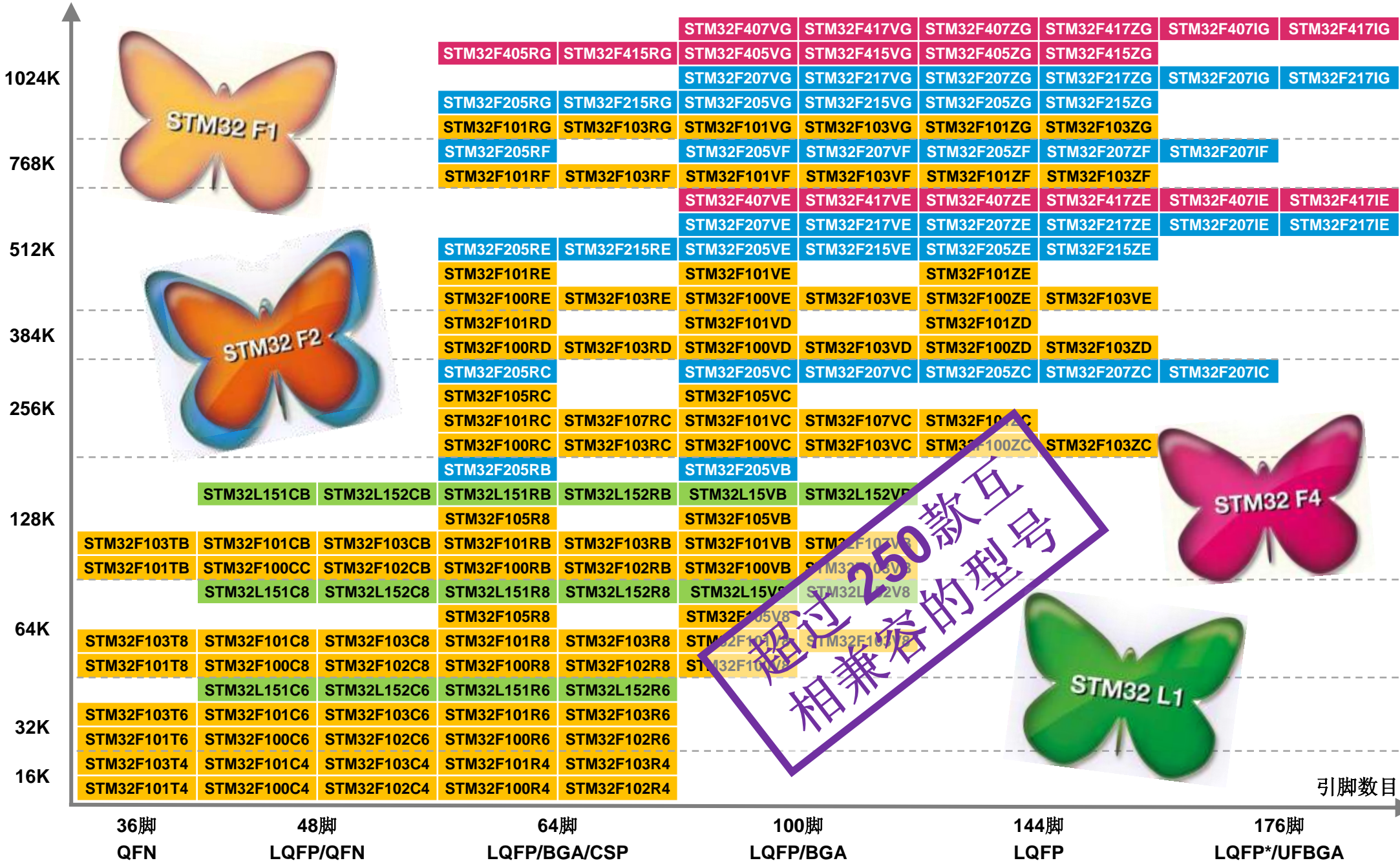
STM32[✦] Releasing your **creativity**



STM32 – 领先的Cortex-M 产品家族



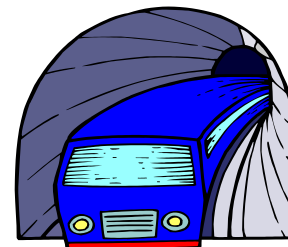
闪存容量 (字节)



STM32家族的新成员：F4系列



- 先进的Cortex-M4内核
 - 浮点运算能力
 - 增强的DSP处理指令
- 更多的存储空间
 - 高达1M字节的片上闪存
 - 高达196K字节的内嵌SRAM
 - FSMC：灵活的外部存储器接口
- 极致的运行速度
 - 以168MHz高速运行时可达到210DMIPS的处理能力
- 更高级的外设
 - 新增功能：照相机接口、加密处理器，USB高速OTG接口...
 - 增强功能：更快的通信接口，更高采样率，带FIFO的DMA控制器...



STM32F4: 数字信号控制器



Cortex-M4内核



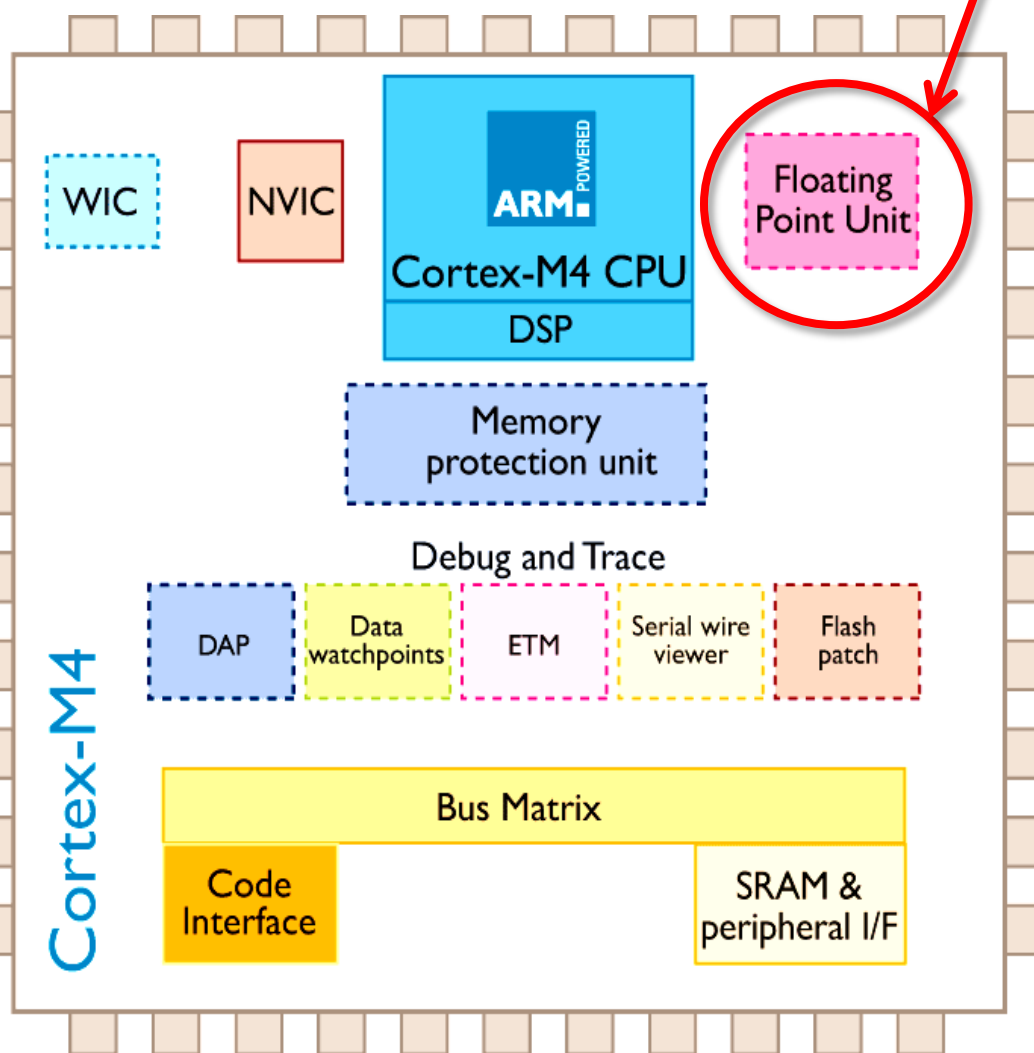
	Cortex-M4	Cortex-M3	Cortex-M0
核心版本	v7ME	v7M	v6M
指令系统	Thumb® / Thumb-2		Thumb® / Thumb-2 subset
指令增强	单周期的 16、32位MAC 单周期的 双16位MAC 8、16位SIMD 计算 硬件除法(2~12周期)	硬件除法(2~12周期) 单周期 (32x32) 乘法 支持饱和算术运算	可选的硬件单周期 (32x32)乘法
流水线	三级 + 分支推测		三级
执行效率	2.19 CoreMark/MHz, 1.25 DMIPS/MHz		1.62 CoreMark/MHz 0.9 DMIPS/MHz
存储器保护	可选。 8 区域管理，可划分子区域和后台区		没有
中断	非屏蔽中断(NMI) + 1~240 个物理中断源		非屏蔽中断(NMI) + 1~32个物理中断源
中断优先级	8~256 个优先级		4级优先
唤醒中断控制器	多达 240 个唤醒中断		可选
睡眠模式	集成 WFI 和 WFE 指令。退出时睡眠功能，睡眠和深度睡眠信号。 使用 ARM 的电源管理部件，可选择保持模式。		
位操作	集成位指令和位带域		
调试	可选的 JTAG 和 SWD 调试接口 支持最多 8 个断点和 4 个察看点		可选的 JTAG 和 SW 调试接口 支持最多 4 个断点和 2 个察看点
跟踪(可选)	指令跟踪(ETM)、数据跟踪(DWT)和仪器跟踪(ITM)模块		

增加单精度浮点运算单元

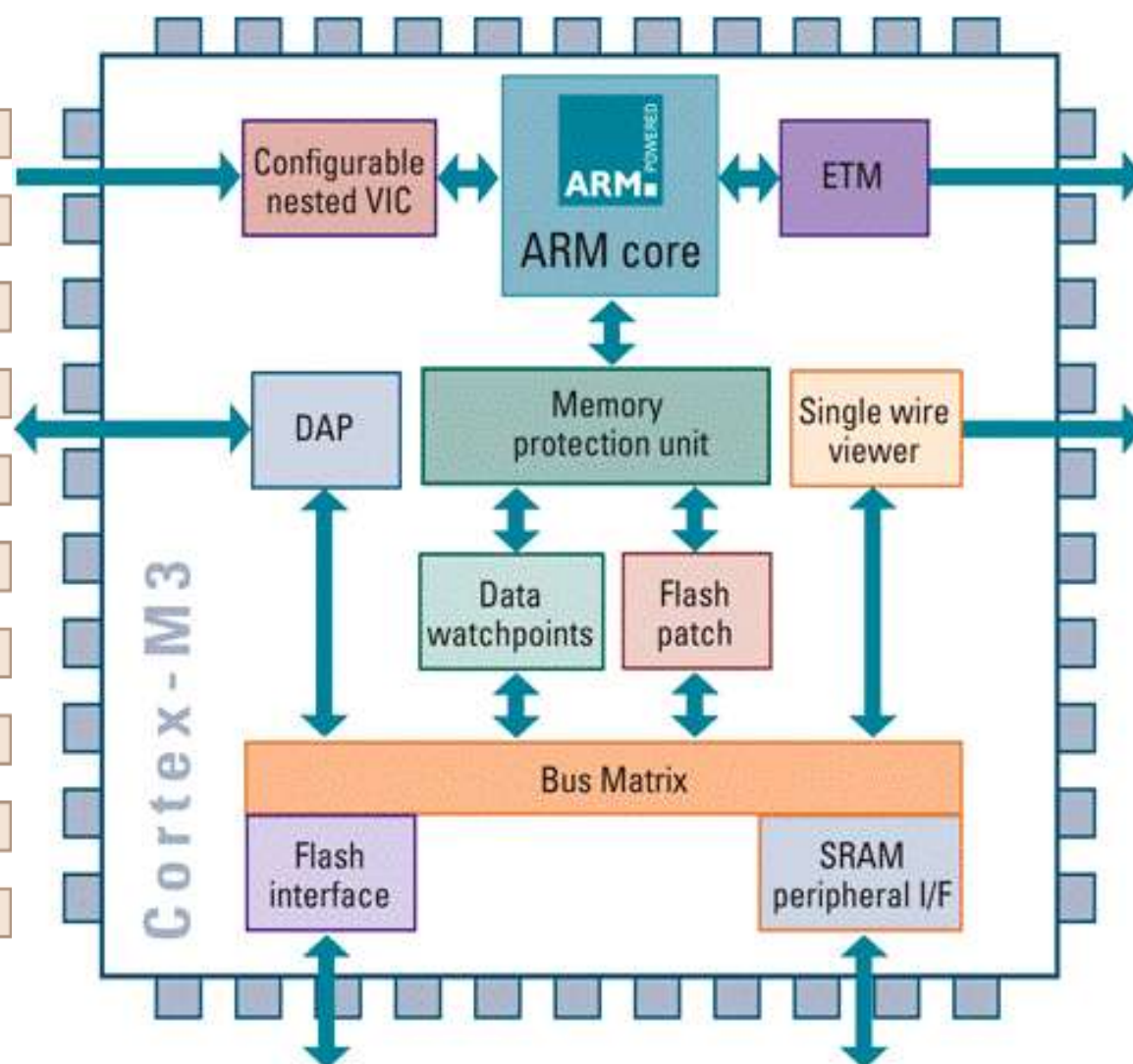


单精度浮点运算单元兼容IEEE 754标准

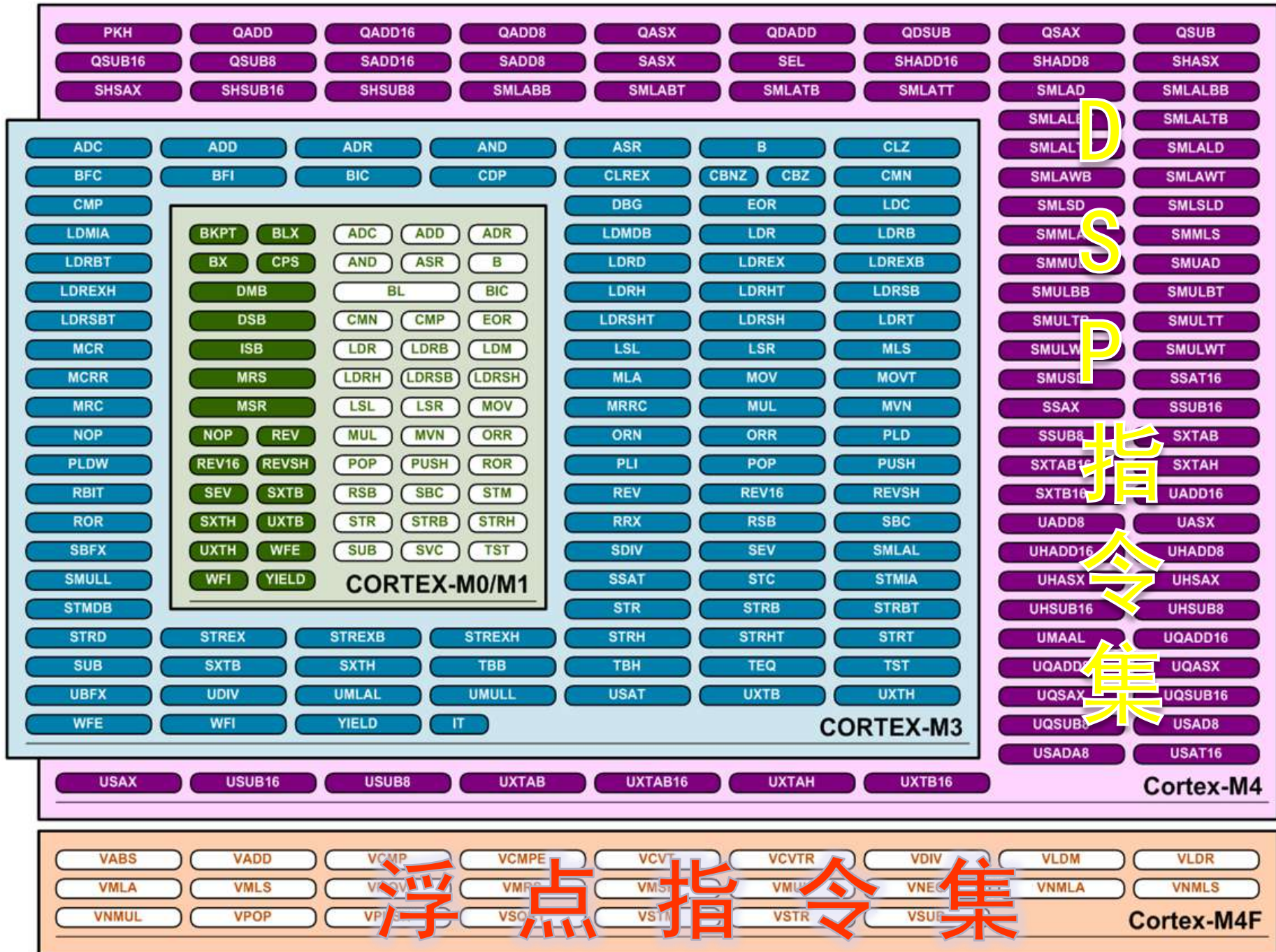
Cortex-M4架构



Cortex-M3架构



增强的指令集



指令示例：单周期的乘加运算

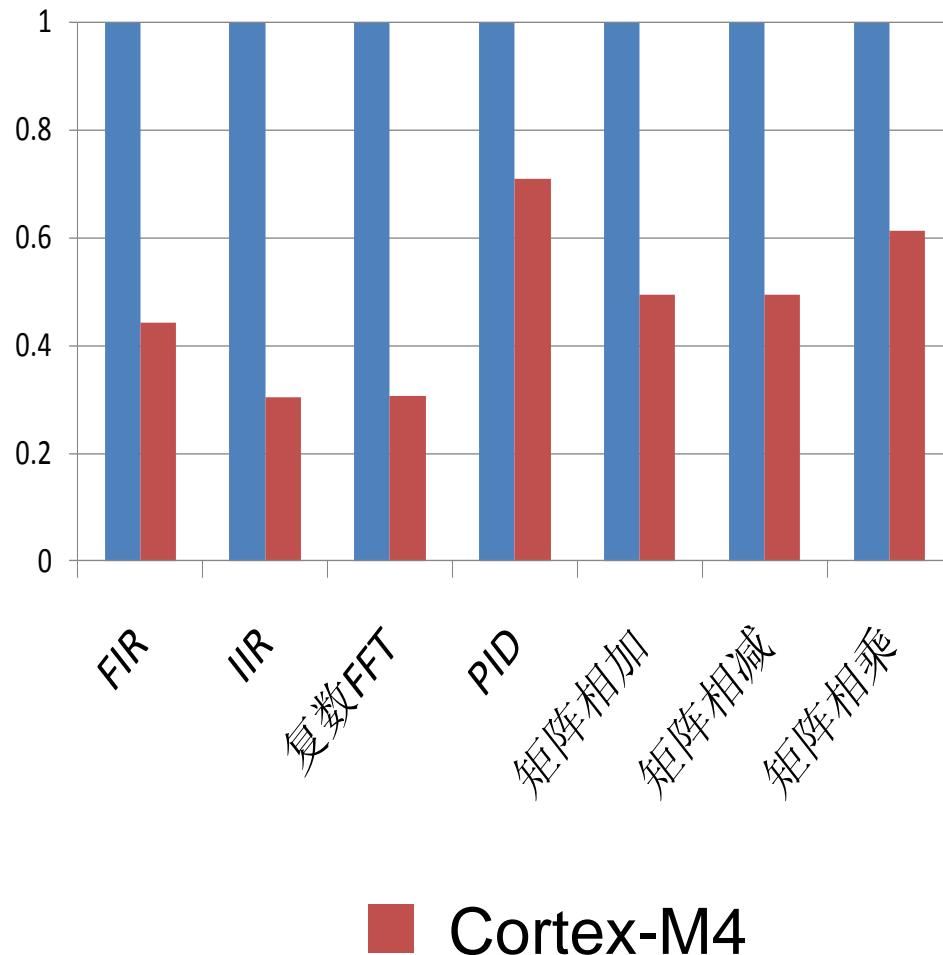


运 算	指 令	CM3	CM4
$16 \times 16 = 32$	SMULBB, SMULBT, SMULTB, SMULTT	n/a	1
$16 \times 16 + 32 = 32$	SMLABB, SMLABT, SMLATB, SMLATT	n/a	1
$16 \times 16 + 64 = 64$	SMLALBB, SMLALBT, SMLALTB, SMLALTT	n/a	1
$16 \times 32 = 32$	SMULWB, SMULWT	n/a	1
$(16 \times 32) + 32 = 32$	SMLAWB, SMLAWT	n/a	1
$(16 \times 16) \pm (16 \times 16) = 32$	SMUAD, SMUADX, SMUSD, SMUSDX	n/a	1
$(16 \times 16) \pm (16 \times 16) + 32 = 32$	SMLAD, SMLADX, SMLSD, SMLSDX	n/a	1
$(16 \times 16) \pm (16 \times 16) + 64 = 64$	SMLALD, SMLALDX, SMLSLD, SMLSLDX	n/a	1
$32 \times 32 = 32$	MUL	1	1
$32 \pm (32 \times 32) = 32$	MLA, MLS	2	1
$32 \times 32 = 64$	SMULL, UMULL	5-7	1
$(32 \times 32) + 64 = 64$	SMLAL, UMLAL	5-7	1
$(32 \times 32) + 32 + 32 = 64$	UMAAL	n/a	1
$32 \pm (32 \times 32) = 32$ (高位)	SMMLA, SMMLAR, SMMLS, SMMLSR	n/a	1
$(32 \times 32) = 32$ (高位)	SMMUL, SMMULR	n/a	1

指令周期数

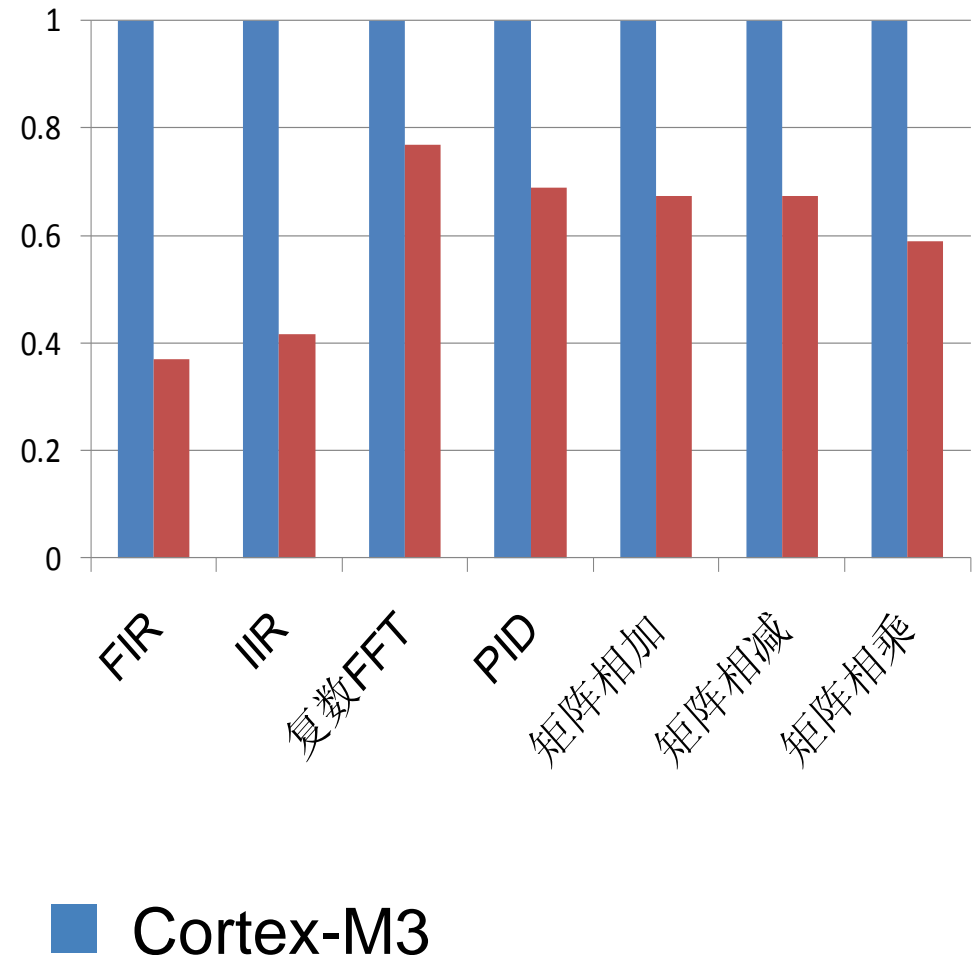
■ 16位DSP功能:

- 相对CM3, CM4只需30%~70%的指令周期

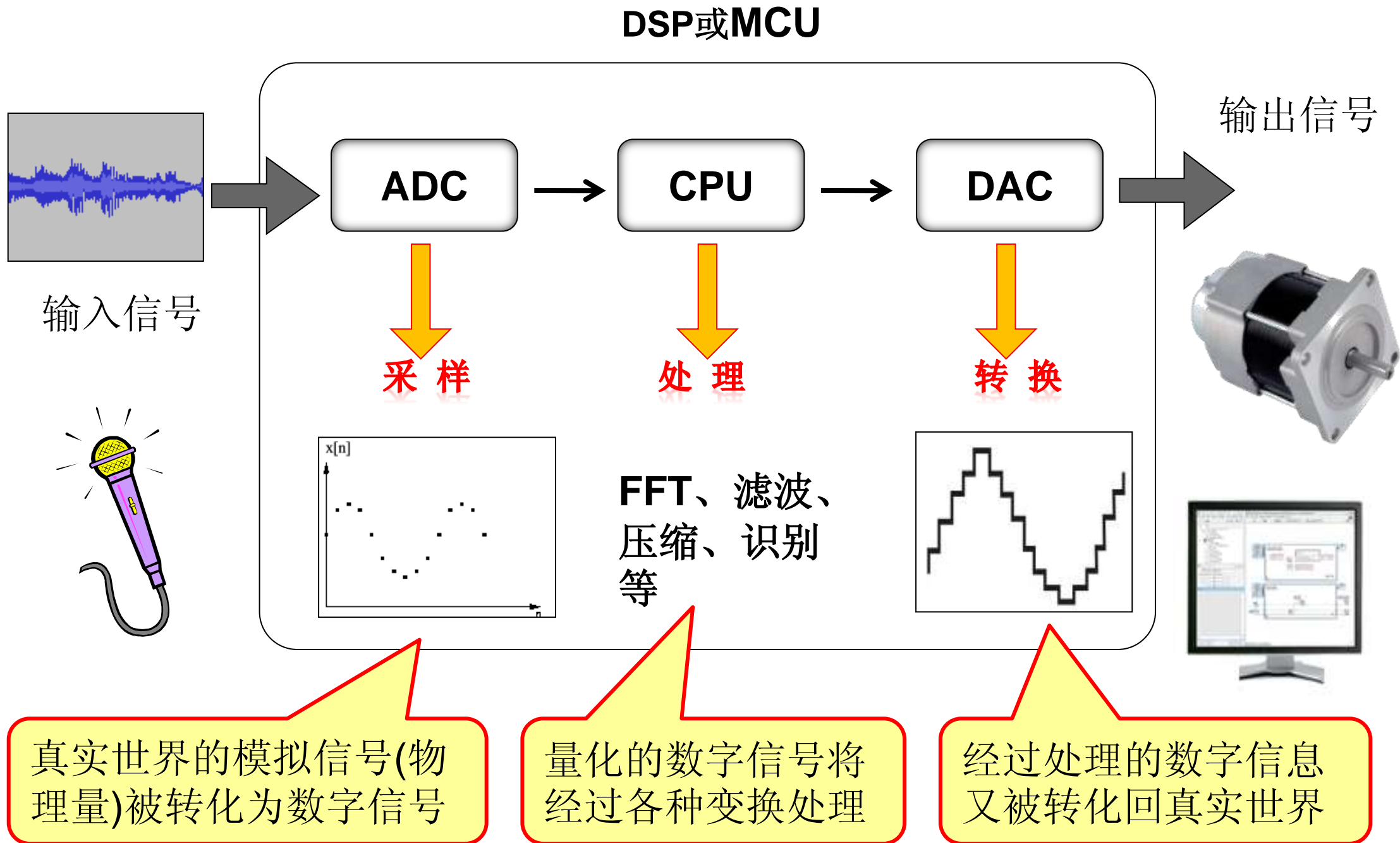


■ 32位DSP功能:

- 相对CM3, CM4只需25%~60%的指令周期



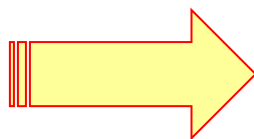
DSP与MCU的关系



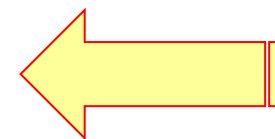
数字信号控制器(DSC)



MCU



STM32 F4



DSP

低成本
低门槛
C编程
高性能中断
超低功耗



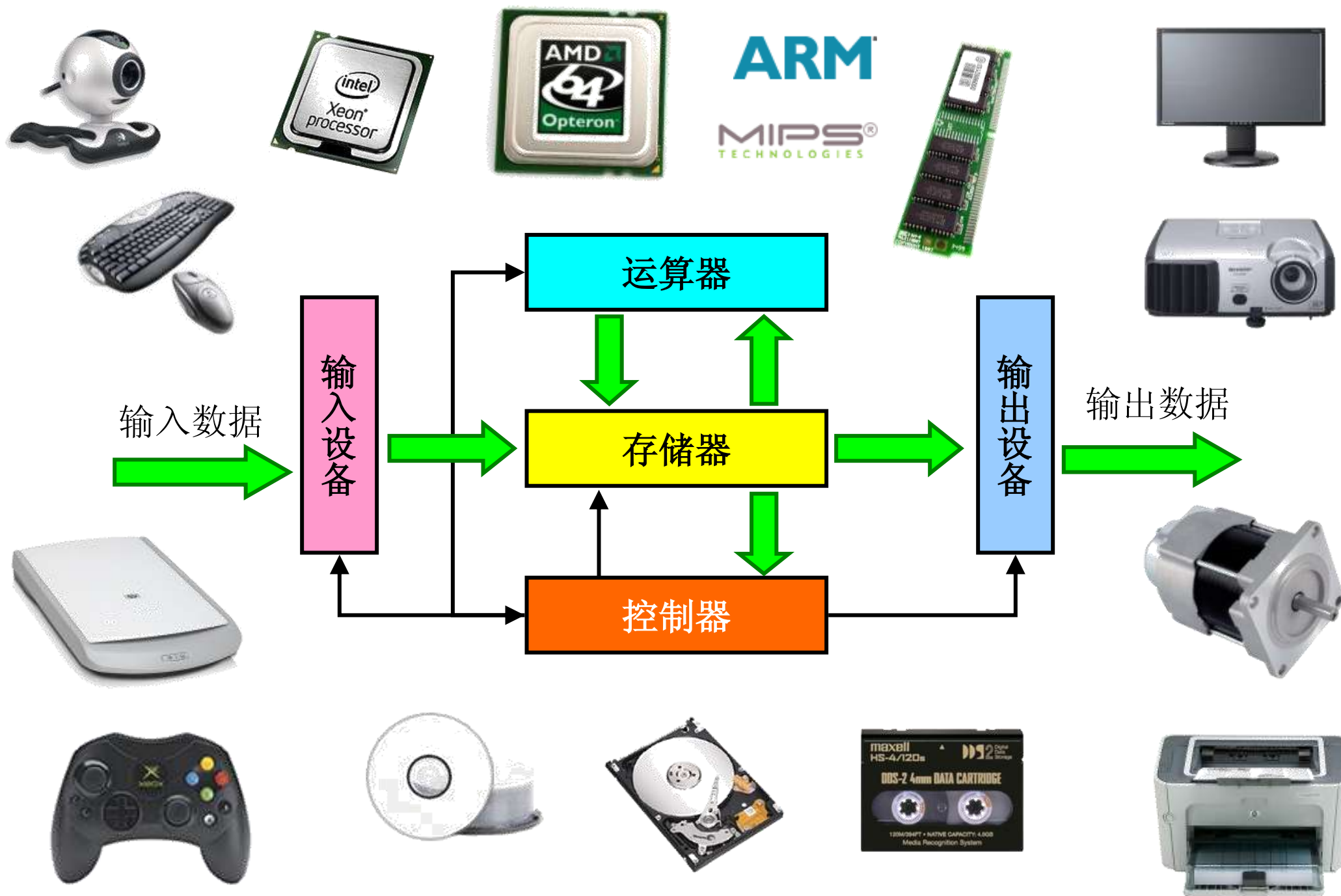
单周期乘加
浮点运算
饱和运算
桶形移位



STM32F4: 高性能的计算能力



冯·诺依曼的计算机模型



■ CPU指令运行的速度

- 流水线结构

- 单周期 vs 多周期指令

■ CPU指令集的选取

- RISC vs CISC

- SIMD vs SISD

- 专用指令集

■ 时钟频率

■ 存储器的速度

- 存储器的构造

- 多级缓冲

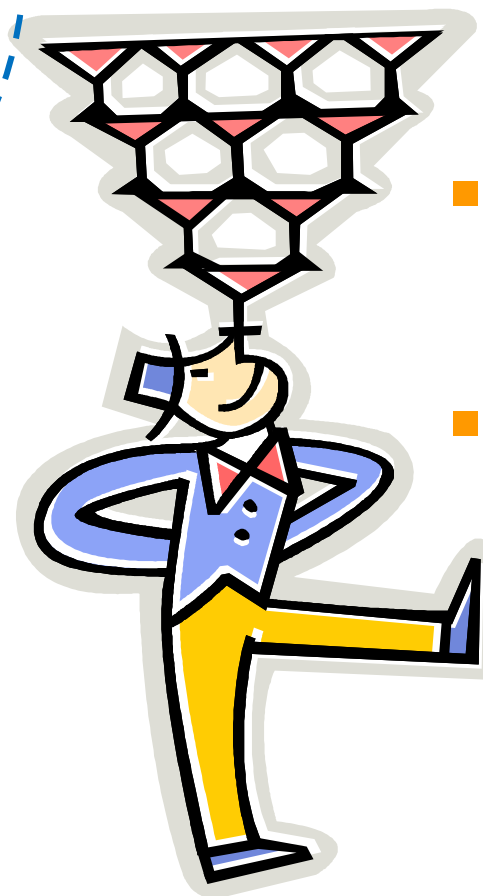
- 代码预测

■ 总线的构造

■ 编译器的优化质量

- 代码的优化

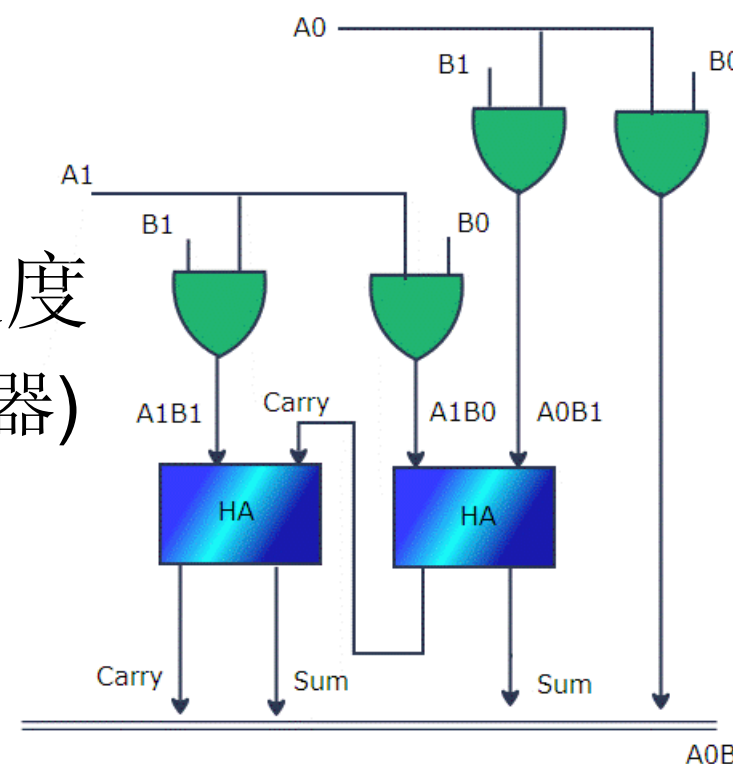
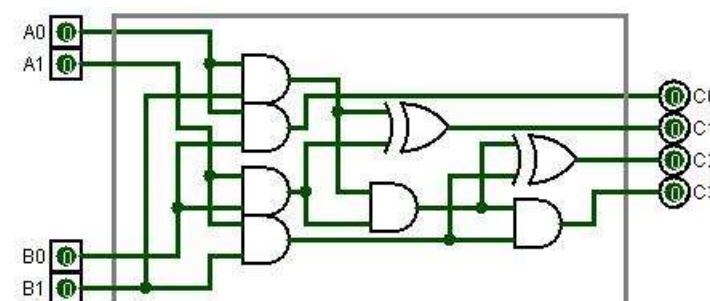
- 指令的选取



显著增强的DSP指令



- 单周期的乘加指令(MAC)
 - 乘法器可以在单周期内完成相乘或乘加指令
 - 有符号和无符号的乘法
 - 有符号和无符号的乘加
 - 有符号和无符号的长数据乘加(64位)
- 单周期的SIMD指令
 - 同时有多个数据参与运算
 - 例如: $16 \times 16 + 16 \times 16 + 32$
- 相对Cortex-M3指令显著提高了速度
 - 16位乘加指令提高4倍(双16位乘加器)
 - 32位乘加指令提高2倍
 - 32位乘加指令提高至7倍



①

DSP指令提高了控制应用的性能



- 这是一个无刷电机控制中用到的一个复数计算
- 获益于LOAD操作和SIMD指令
- 整体性能提高25%至35%

①

Cortex M3 (28-38周期)	Cortex M4 (18-28周期)
LDRSH R12,[R4, #+12]	LDR R10,[R4, #+12]
LDRSH R0,[SP, #+20]	(1条32位load指令代替了2条带符号扩展的16位load指令: 减少2个周期)
SXTH LR,R8	
MUL R8,LR,R0	
LDR R1,[R4, #+44]	
SDIV R0,R1,R7	
LDRSH R2,[R4, #+24]	
LDRSH R3,[R4, #+26]	LDR R2,[R4, #+22]
LDRSH R10,[R4, #+22]	(1条32位load指令代替了2条带符号扩展的16位load指令: 减少2个周期)
SXTH R6,R6	
MLS R5,R6,R10,R5	
MLA R5,R9,R12,R5	SMLSD R5, R10, R6, R5
	(1条SIMD指令代替2条乘加指令: 减少3个周期)
ASR R6,R8,#+15	
MLA R5,R6,R3,R5	
SXTH R0,R0	
MLS R5,R0,R2,R5	SMLSD R5, R0, R2
	(1条SIMD指令代替2条乘加指令: 减少3个周期)
STR R5,[SP, #+12]	

浮点运算更贴近于现实的应用



```
01 1 1 001 1 1 00001 01 01 01 001 01
1 00001 01 01 01 001 01 1 1 01 1 001
1 1 1 00001 01 01 01 001 01 1 1 01 1
01 1 1 001 1 1 00001 0
01 01 1 1 001 1 1
01 01 01 001
001 01 1 1
```

高级抽象模型
(矩阵、数学公式等)

多元的高级开发工具
(MATLab, Scilab,等)

C代码生成
(多数是基于浮点运算)

**Cortex-M4
的适用范围**

**Cortex-M3
的适用范围**

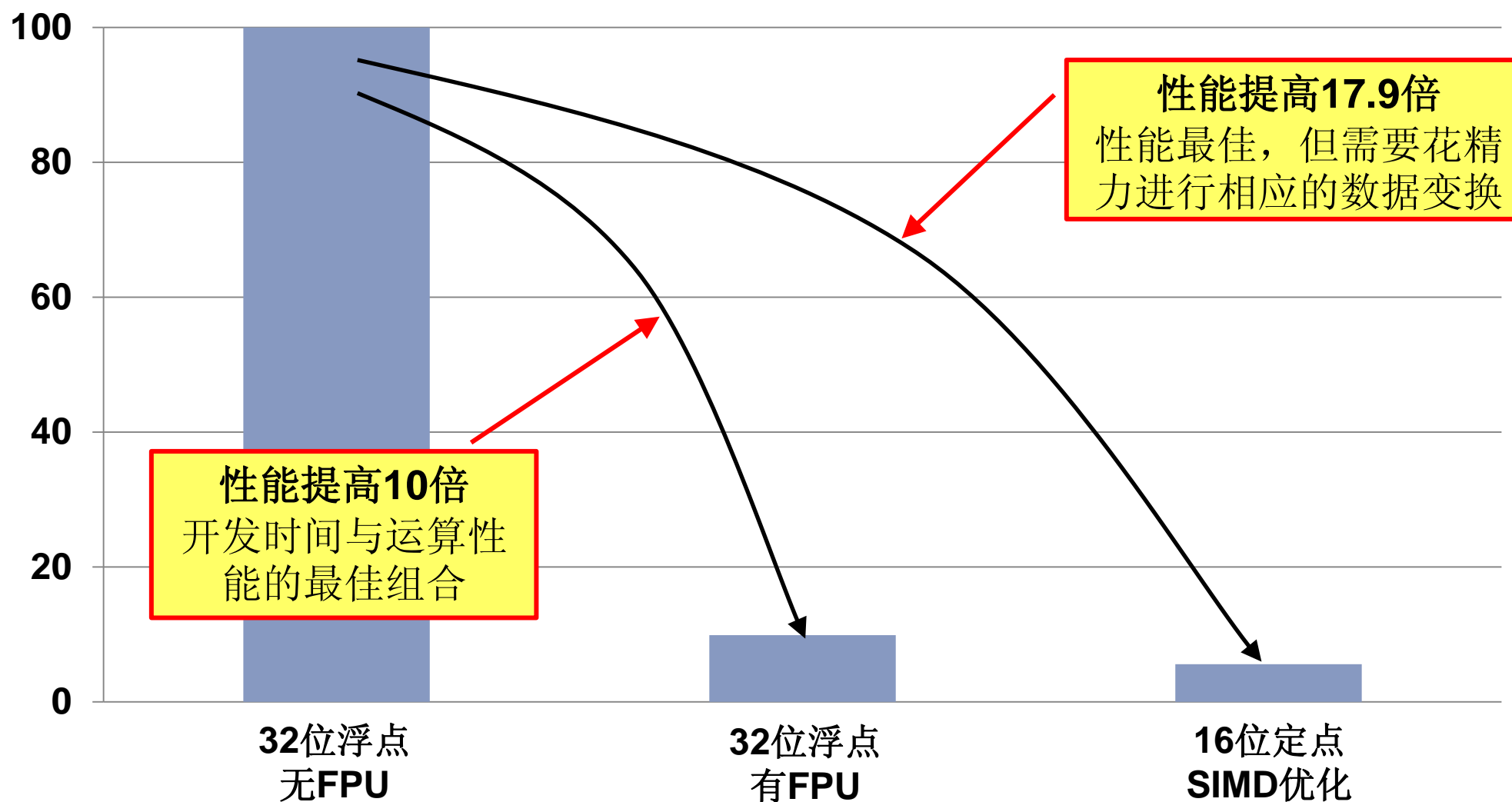
具FPU的处理器
硬件直接处理浮点运算
不需更改代码
很高的性能
最优的代码效率

无FPU的处理器
调用软件库处理浮点运算
不需更改代码
较低的性能
中等的代码效率

无FPU的处理器
使用整数格式计算
需要更改代码
需注意检查边界条件
(例如饱和运算、缩放系数等)
中等或较高性能
中等的代码效率

①

- 这是一个基于浮点的29级FIR计算例子
 - 以CMSIS库函数实现的性能

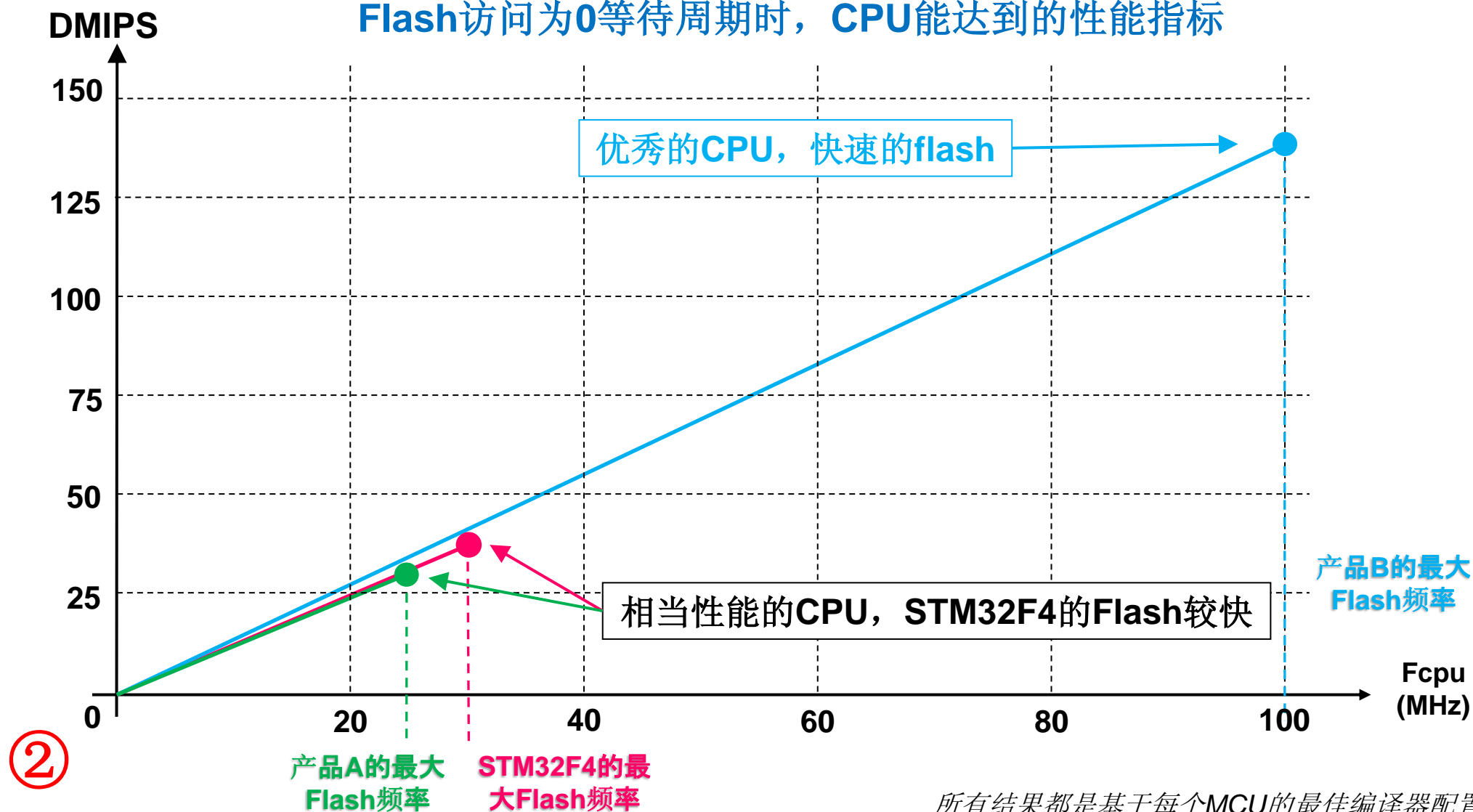


Flash的速度与CPU的速度



- 由于生产工艺的限制，当CPU主频显著提高时，Flash的存取速度却只能处于一个较低的水平。

Flash访问为0等待周期时，CPU能达到的性能指标

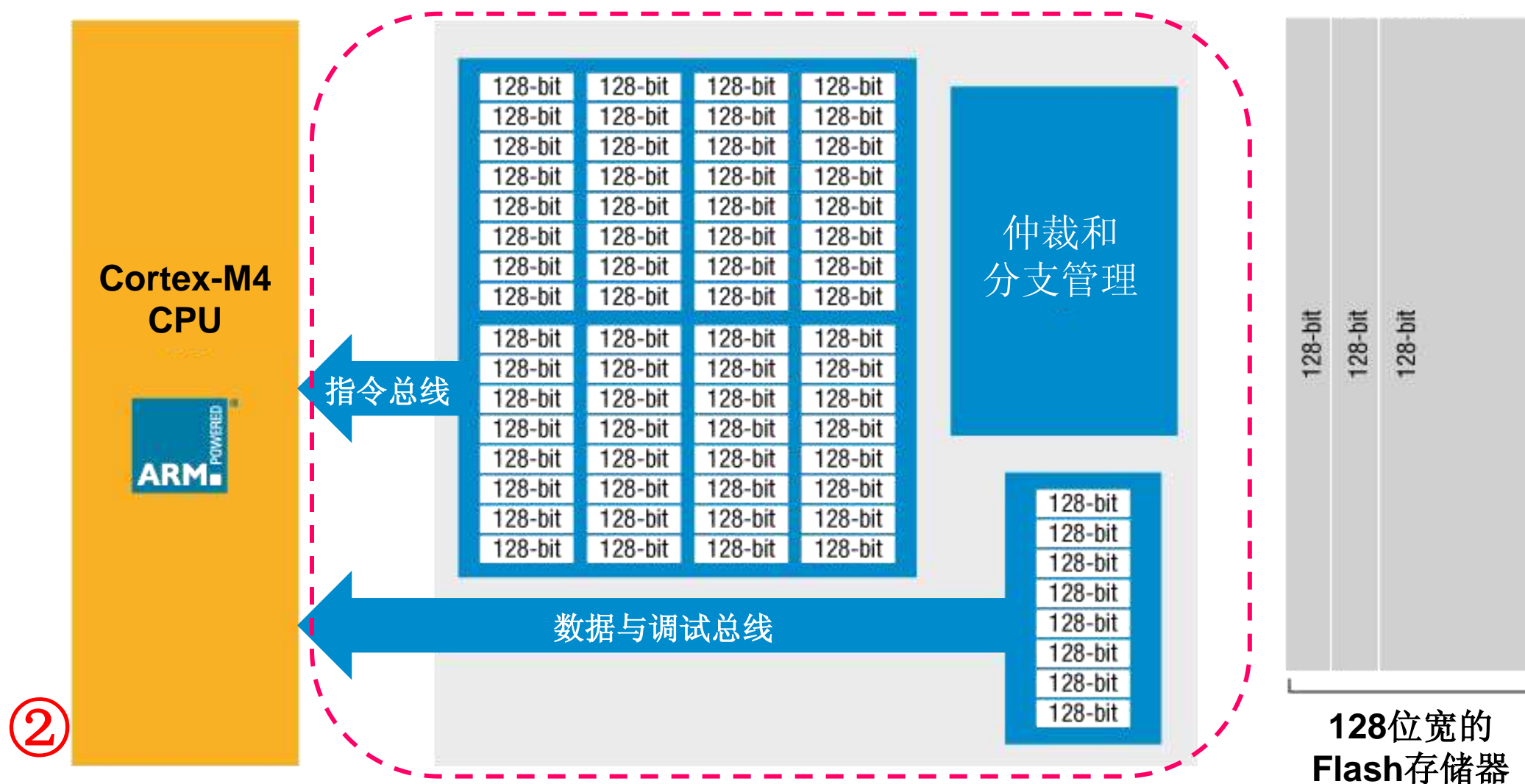


所有结果都是基于每个MCU的最佳编译器配置获得

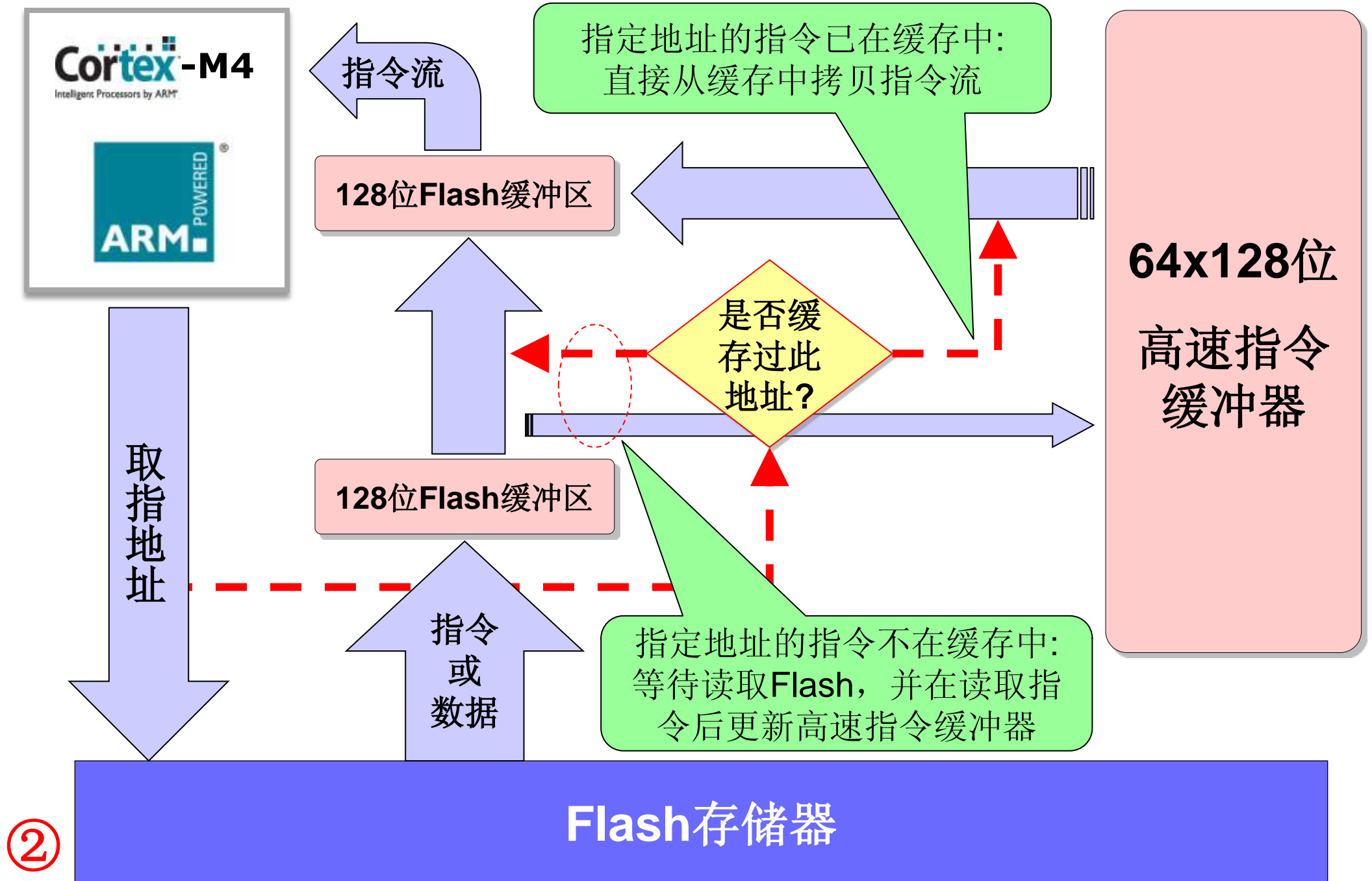
ART自适应实时加速器™



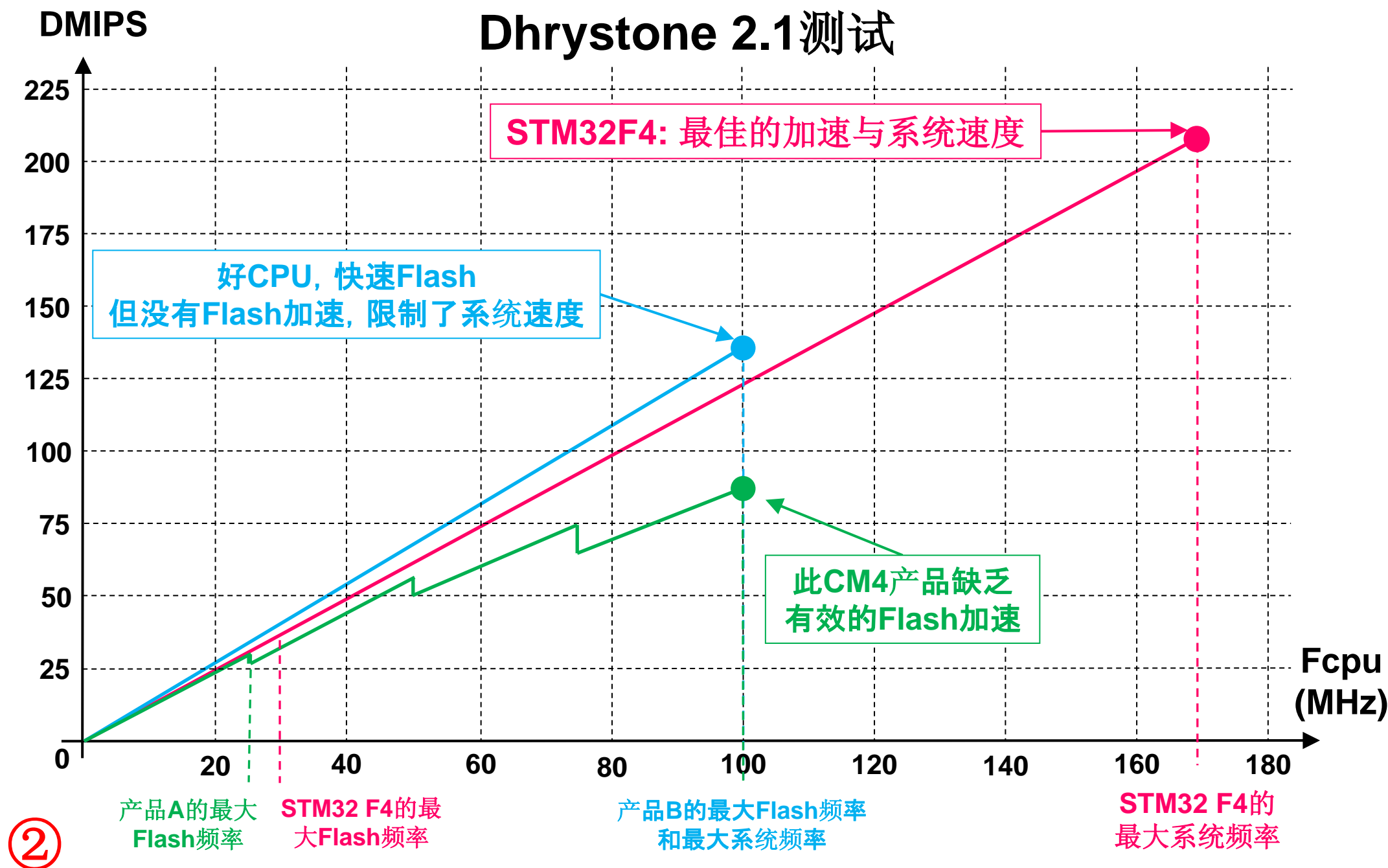
- 自适应实时加速器能够完全释放Cortex-M4内核的性能；当CPU工作于所有允许的频率($\leq 168\text{MHz}$)时，在闪存中运行的程序，可以达到相当于零等待周期的性能。



ART加速原理解析(指令)



Dhrystone的测试结果对比

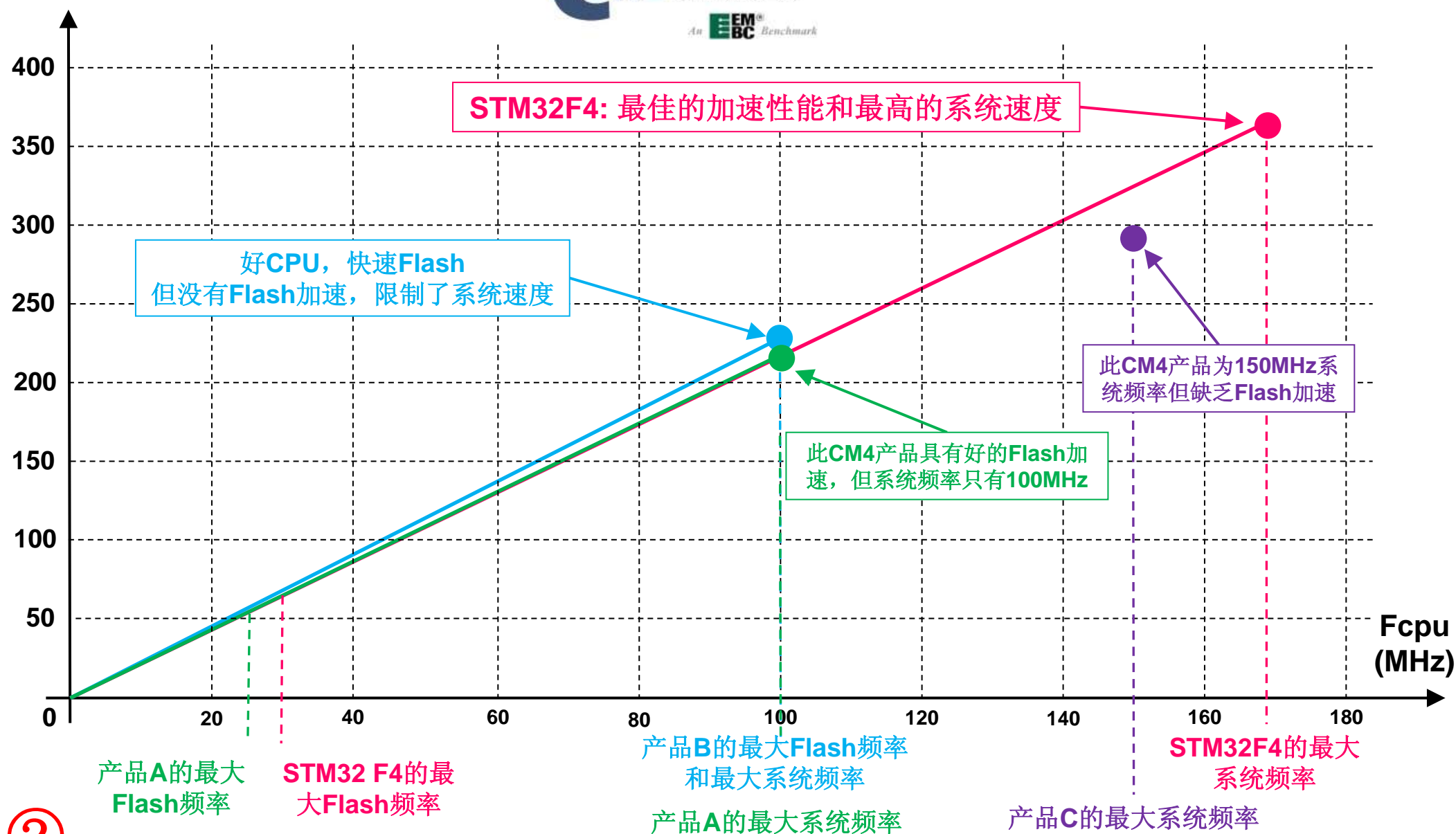


所有结果都是基于每个MCU的最佳编译器配置获得

Coremark的测试结果对比



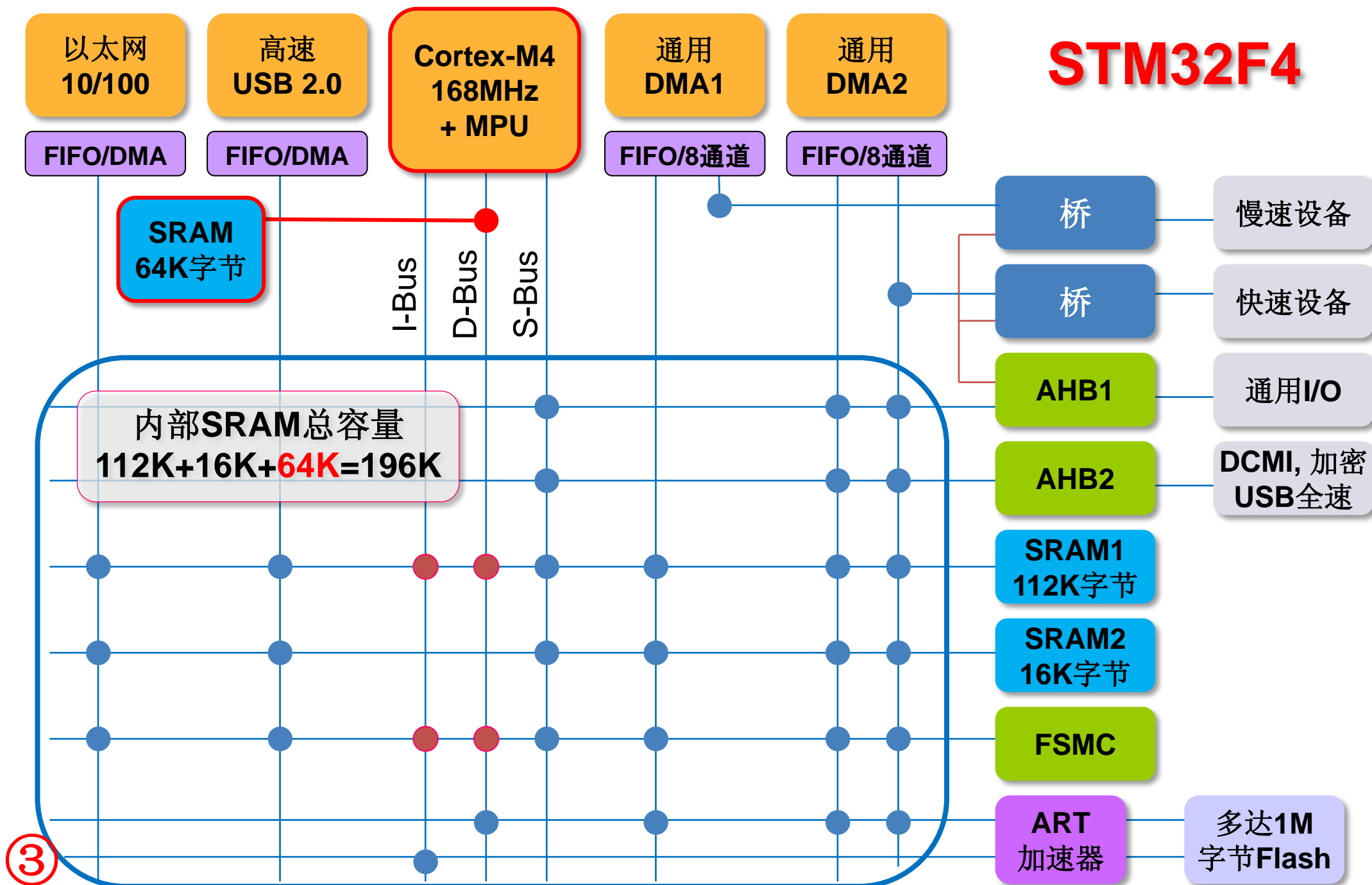
Coremark



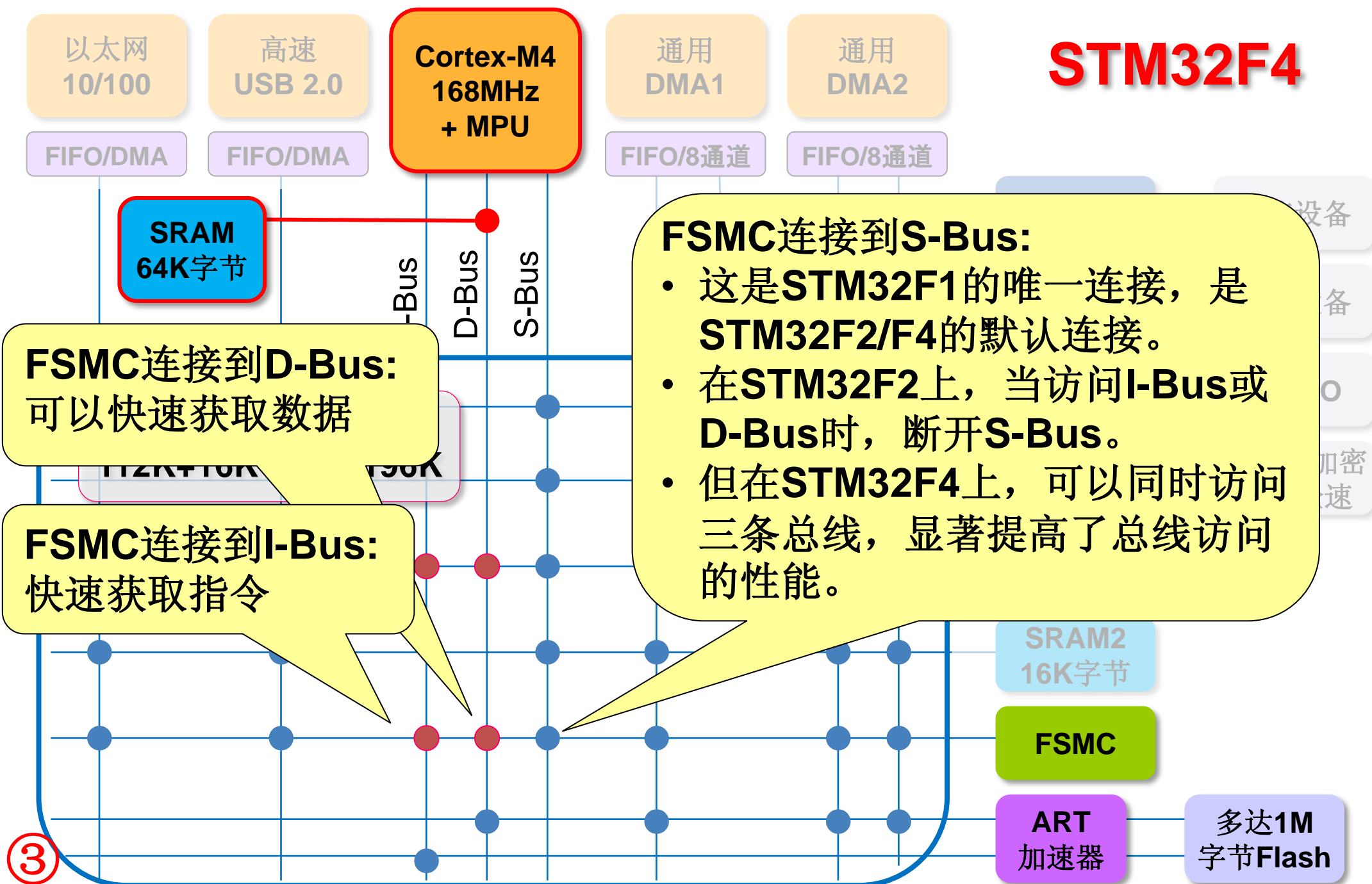
②

所有结果都是基于每个MCU的最佳编译器配置获得

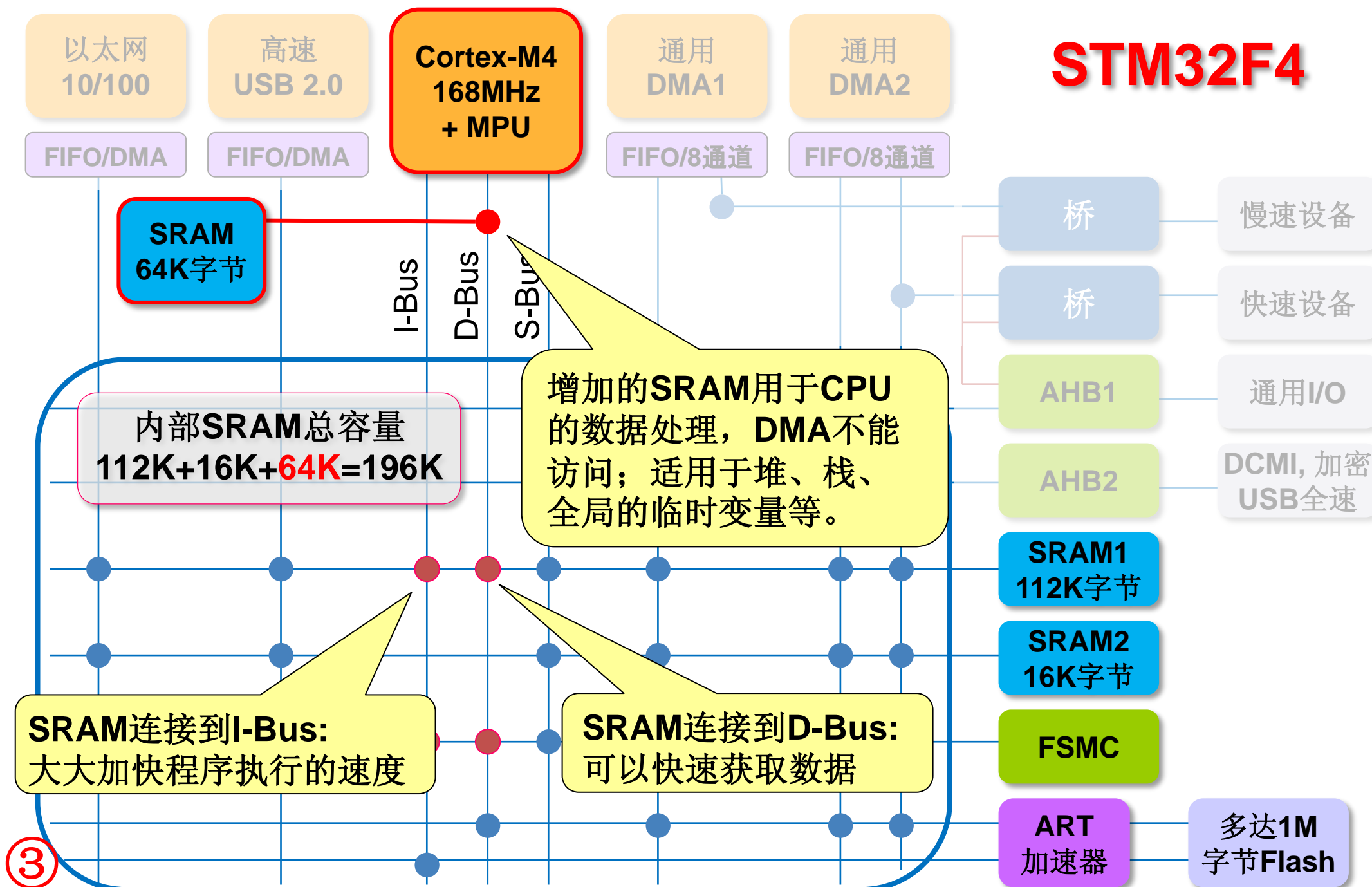
32位多重AHB总线矩阵



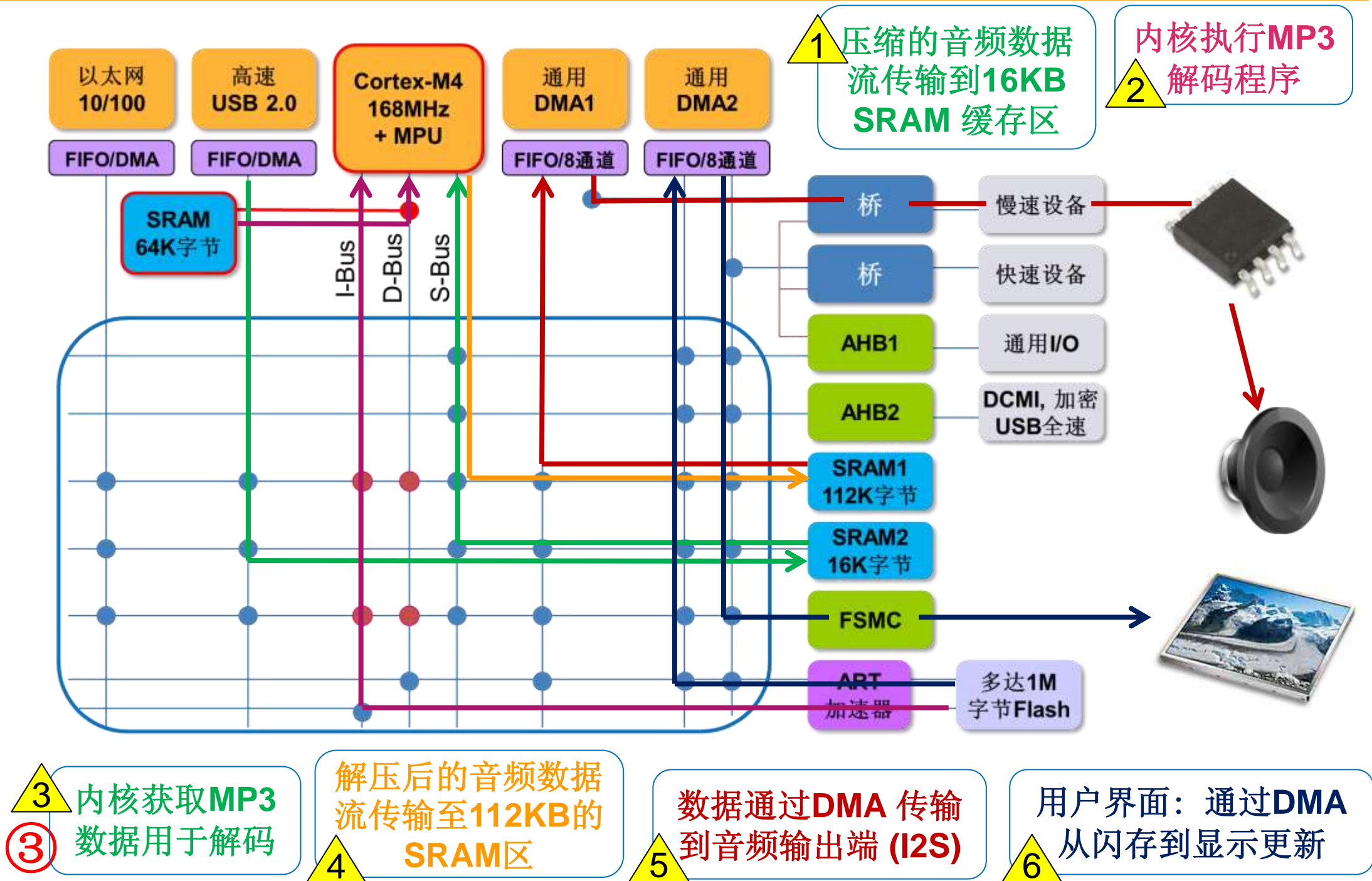
FSMC的重映射功能



内部SRAM至CPU的直连通道



多重总线的并行处理能力



STM32F4: 卓越的功耗效率

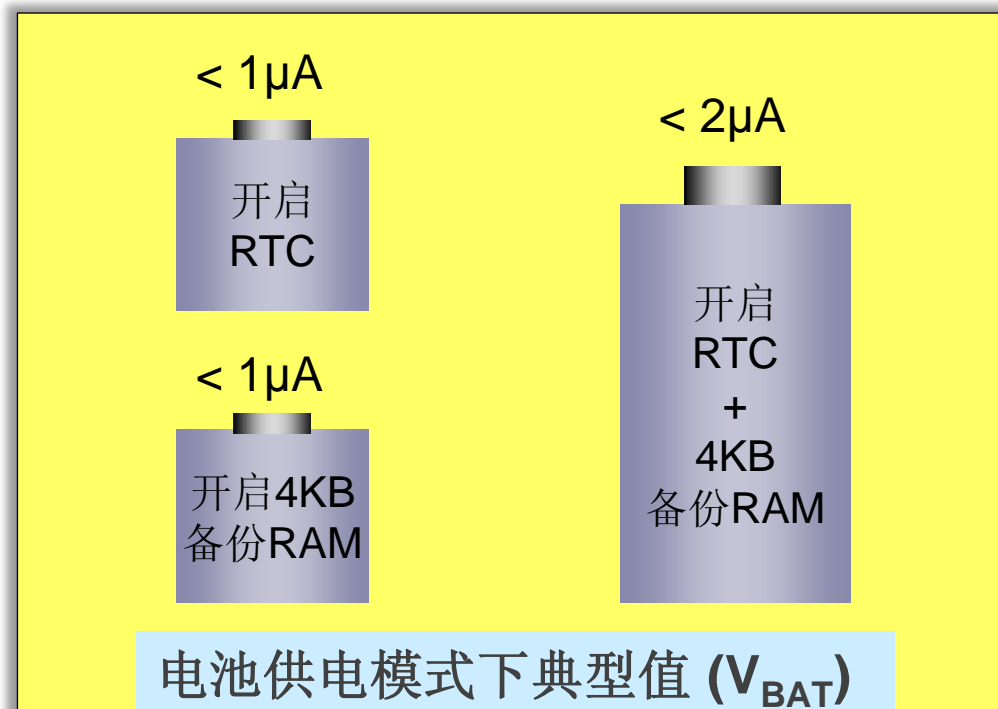


■ 230 $\mu\text{A}/\text{MHz}$:

主频为168MHz情况下，在闪存中执行CoreMark基准测试程序，功耗为38.6mA，这得益于：

- ST的90纳米工艺可使 CPU内核工作电压低至1.2 V
 - 自适应实时加速器减少了访问闪存的次数
 - 电压可调节特性，便于优化性能功耗比
 - 可关闭内部调压器，使用外部调压器为CPU供电
- 供电电源可低至1.7伏
 - 在最低功耗模式下，可以支持后备存储器和实时时钟工作

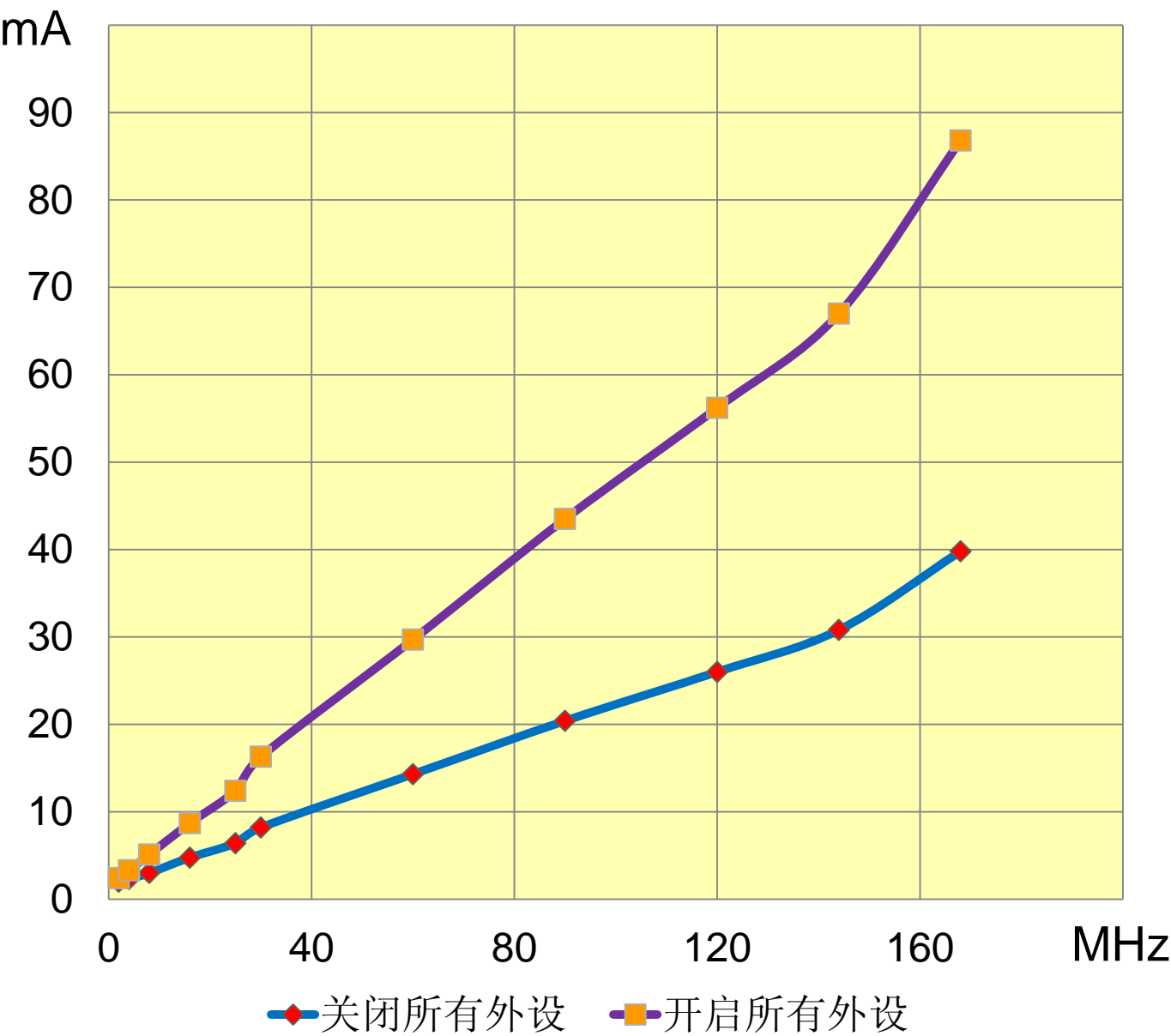
供电范围	最高主频
2.4~3.6V	168MHz
2.1~3.6V	144MHz
1.7~3.6V	128MHz



运行模式下的典型功耗数值



运行模式； 程序和代码在Flash或RAM中； 开启ARM加速

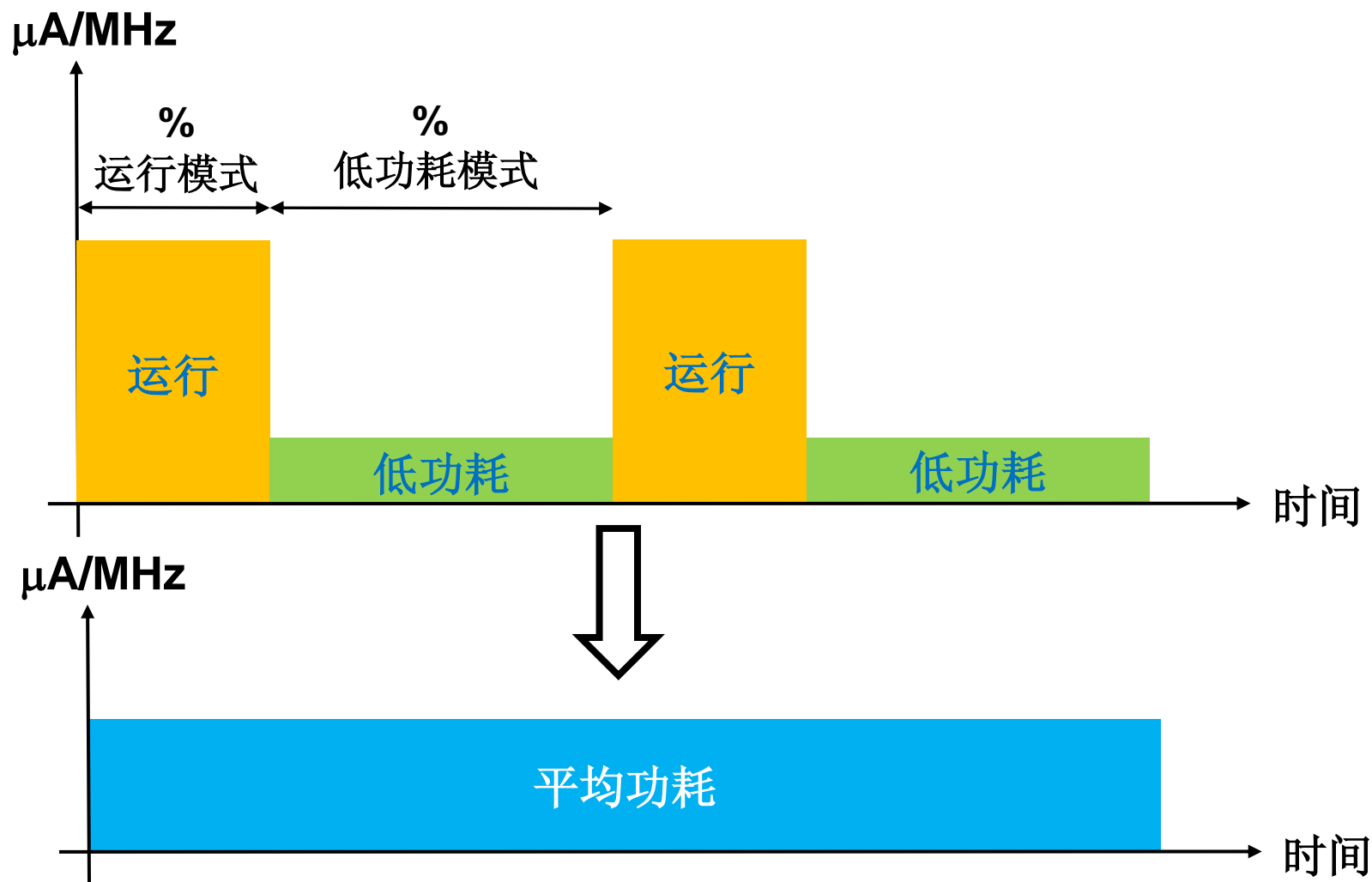


mA	关闭所有外设	开启所有外设
2MHz	2	2.4
4MHz	2.3	3.3
8MHz	3	5.1
16MHz	4.75	8.7
25MHz	6.4	12.4
30MHz	8.2	16.3
60MHz	14.3	29.7
90MHz	20.4	43.5
120MHz	26	56.2
144MHz	30.8	67
168MHz	39.8	86.8

在真实应用中的低功耗



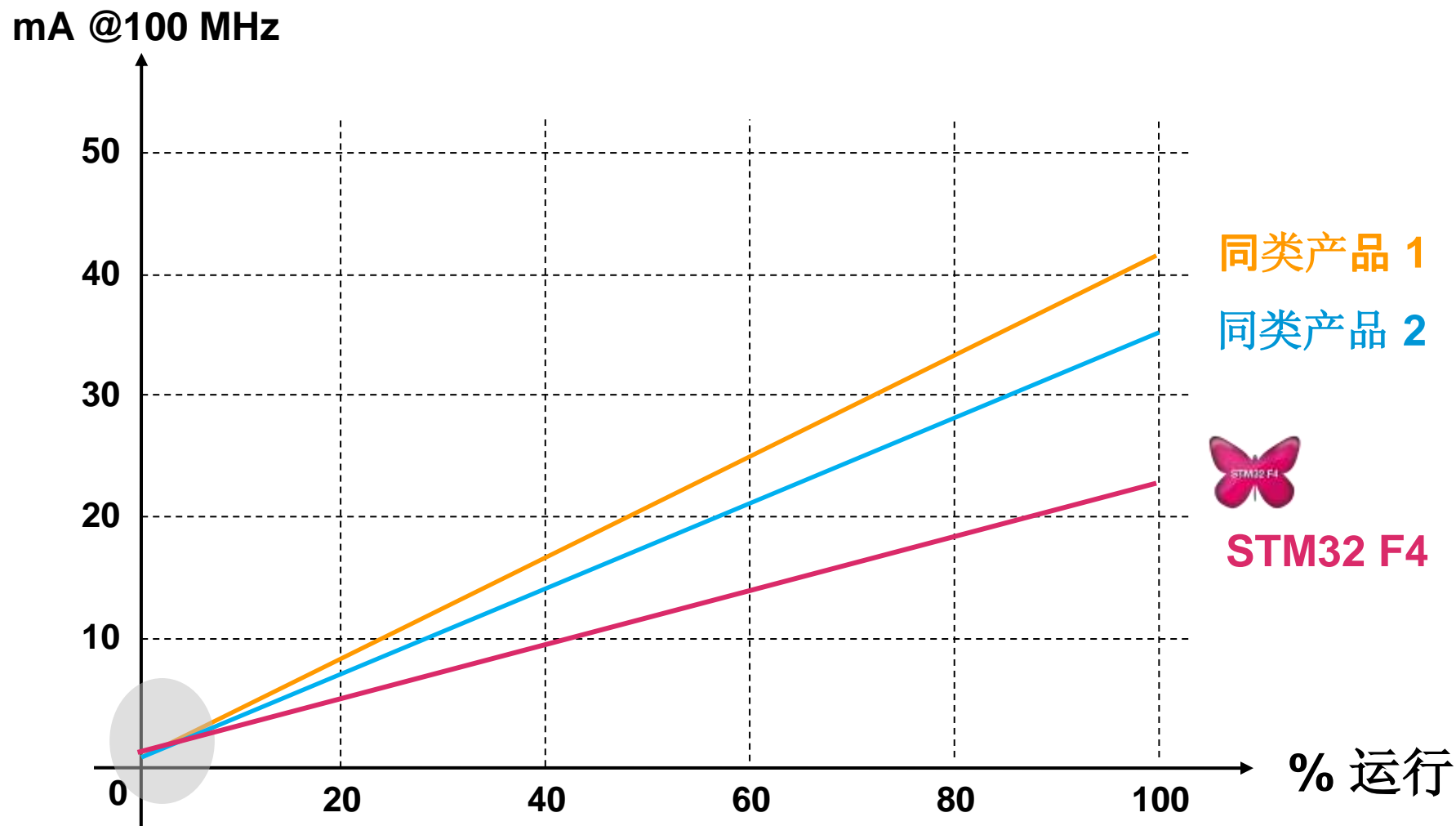
- 低功耗在真实应用中并不仅仅是低功耗模式
- 还需要考虑运行模式和低功耗模式的时间比例



平均功耗



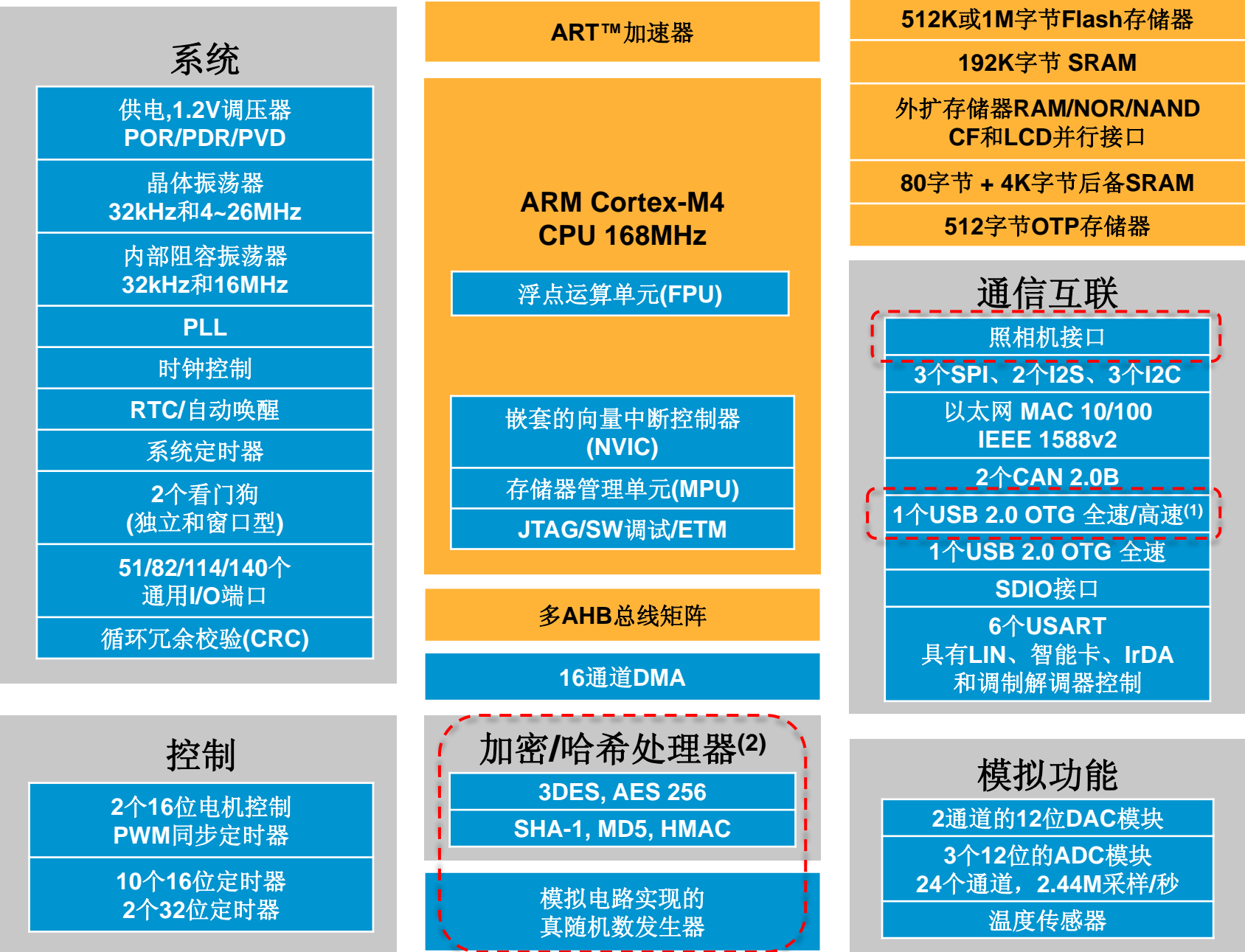
- 运行模式下，STM32 F4具有更高的功耗效率
- STM32F4 特别适合需要有一定运算量的应用场合



STM32F4: 更多的高级外设



STM32F40x/41x框图

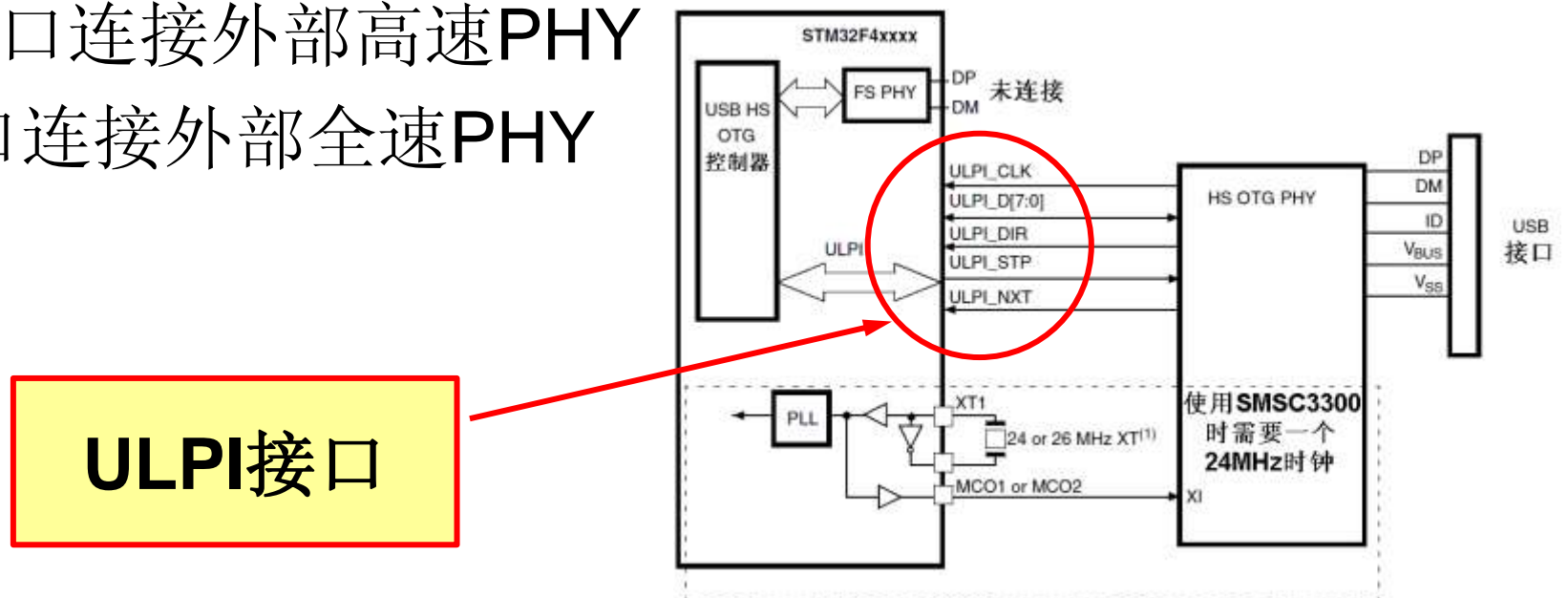


1. 高速功能需通过ULPI接口连接一个外部PHY 2. 只适用于STM32F417x和STM32F415x

USB OTG高速模块



- 主机模式下支持高速/全速/低速
- 从机模式下支持高速/全速
- 内嵌专用DMA，支持突发传输
- 专用4K字节FIFO
- 支持多种PHY接口
 - 片上内嵌全速PHY(与全速OTG模块相同)
 - 以ULPI接口连接外部高速PHY
 - 以I2C接口连接外部全速PHY



- 与全速主机模式相同的性能
 - 多达12个端点



- 支持高速协议特有的特性

- **PING**协议

主机发起**PING**协议以周期性检查从机是否准备好

- **SPLIT**协议

主机通过高速**HUB**连接全速/低速设备时，无须等待设备应答；可以先作其他通信，过段时间再来检查是否受到外设的应答

- **多传输协议**

125uS的微帧时间内对同步传输使用**DATA0**、**DATA1**、**DATA2**和**MDATA**等多种数据**PID**

USB OTG高速模块 – 从机模式



- 与全速从机性能相同
- 新增以下扩展特性
 - 多达5个IN端点
 - 多达5个OUT端点
 - EP1_IN和EP1_OUT有各自的中断向量入口
- NYET
 - 高速模式下从机收到数据包，如果在接收FIFO中没有找到足够的空余空间，返回NYET握手信号



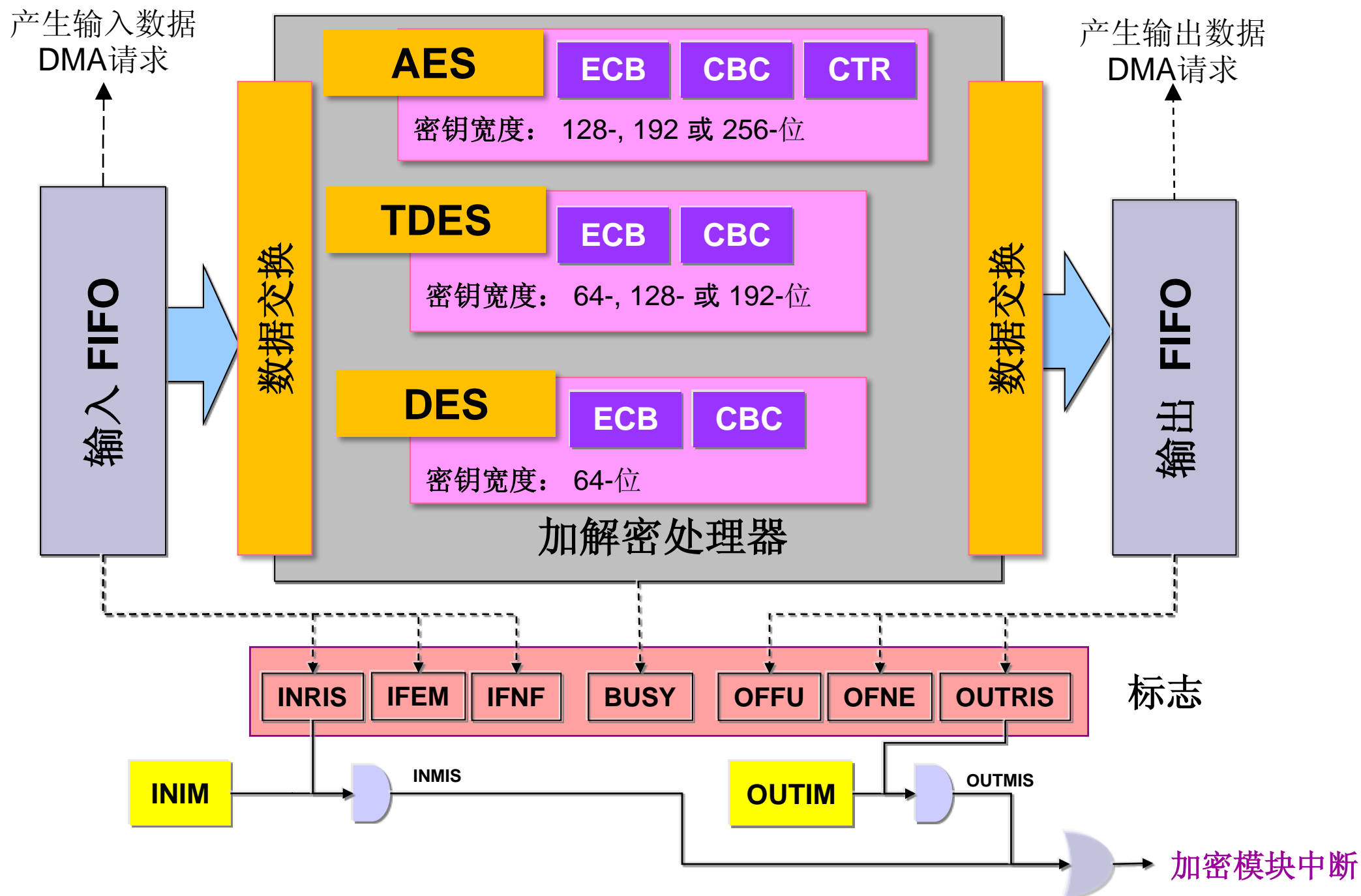
- 主要性能
 - 8/10/12/14位并行接口
 - 连续和快照模式，裁减功能
 - 支持的数据格式有：
 - 8/10/12/14逐行扫描视频信号
 - YCbCr4:2:2逐行扫描视频信号
 - RGB565逐行扫描视频信号
 - 压缩数据支持JPEG格式
- 在48MHz时钟和8位宽度数据接口配置下可接收
 - 15fps的SXGA分辨率每像素2字节的未压缩数据流
 - 30fps的VGA分辨率每像素2字节的未压缩数据流



- 支持加解密算法
 - DES、TDES和AES
- 加解密工作模式
 - 电子密码本(Electronic codebook)模式
 - 密码段链接(Cipher block chaining)模式
 - 计数器(Counter)模式
- 加解密处理器是AHB2上的外设
 - 最高工作频率168MHz
- 支持DMA传输
- 输出/输出端各有8个字的FIFO



加解密处理器框图

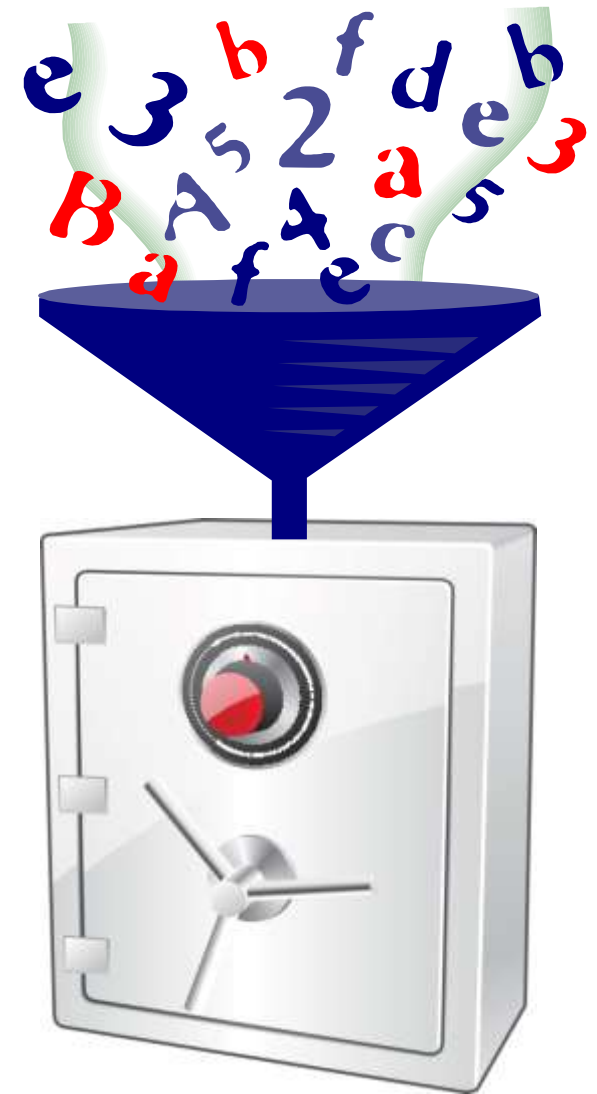


加解密算法特性比较

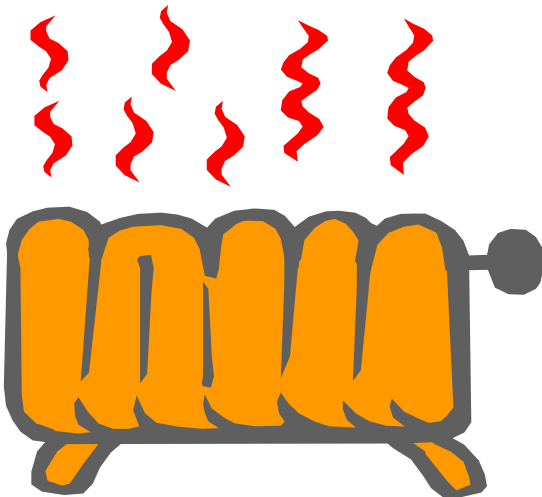


	AES	DES/TDES	共同特性
密钥宽度	128/192/256 位	64位(8位校验) 192/128/64位 (14/16/8位校验)	<p>>> 输入/输出端内嵌各自FIFO(8*32位)</p> <p>>> 支持数据流自动管理 – DMA</p> <p>>> 内嵌数据交换逻辑以支持1/8/16/32位数据</p>
数据块大小	128位	64位	
处理单块数据的时间	14/16/18个HCLK周期	16个HCLK周期 48个HCLK周期	
支持的算法	ECB/CBC/CTR	ECB/CBC/CTR	
加密类型	块加密	块加密	
算法结构	迭代-组合网络	Feistel网络	

- AES加解密算法
 - 安全网络路由
 - 无线通信
 - 加密后的数据保存，包括安全智能卡
- DES/TDES加解密算法
 - 安全数据/文件传输
 - 电子资金转帐
 - 门控：防止用户的密码和个人信息被非法访问

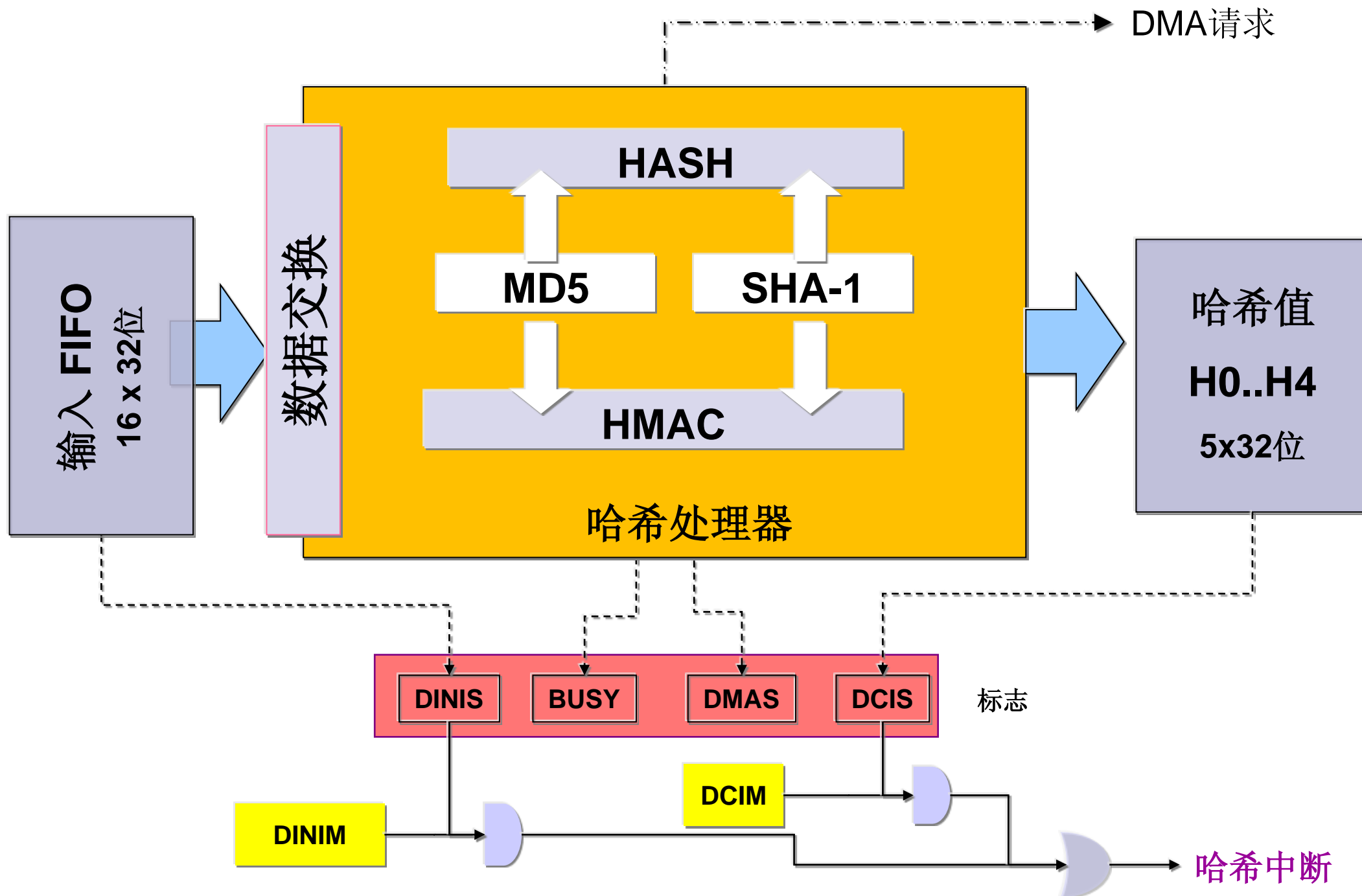


- 主要特性
 - 基于连续模拟噪声的随机数产生器
 - 提供32位的随机数
 - 工作于PLL48CLK时钟下
 - 两个连续随机数之间间隔为40个PLL48CLK周期
 - 为降低功耗可以关闭它



- 主要特性
 - 适用于数据完整性和有效性检查
 - AHB2上的外设，最高工作频率**168MHz**
 - **SHA-1**和**MD5**算法的快速计算
 - **SHA-1**: 66个HCLK周期; **MD5**: 50个HCLK周期
 - **32位**的输入数据，支持**32位/16位/8位/1位**的比特流输入，要求小端对齐方式输入
 - 数据自动交换以遵循**SHA1**计算的大端输入要求
 - 对输入比特流自动填充到**512位**以做哈希计算

哈希处理器框图



- 校验文件或信息的完整性
 - 安全哈希算法一个重用应用就是校验信息的完整性，确定其是否被恶意修改或破坏过。比如可以通过比较传输之前和之后数据的哈希值来校验
 - 检查信息哈希值的有效真实性，从而确定信息本身的有效真实性。通常密码都不是以明文形式存储，而是以哈希值的形式。要确实用户是否合法，只需把用户输入的密码做哈希运算，再把所得哈希值和存储的密码哈希值比较即可。



- 外设性能进一步增强
 - 更快的模数转换速度
 - 更低的ADC/DAC工作电压
 - 带日历功能的实时时钟
 - 4K字节的电池备份SRAM
 - 32位定时器
 - 更快的USART和SPI通信速度
- JTAG引脚熔断保护
- 更多的GPIO



谢谢您！



STM32  Releasing your **creativity**



www.st.com/stm32f4