

认识CPU

浙江大学嵌入式系统课程
2016春夏

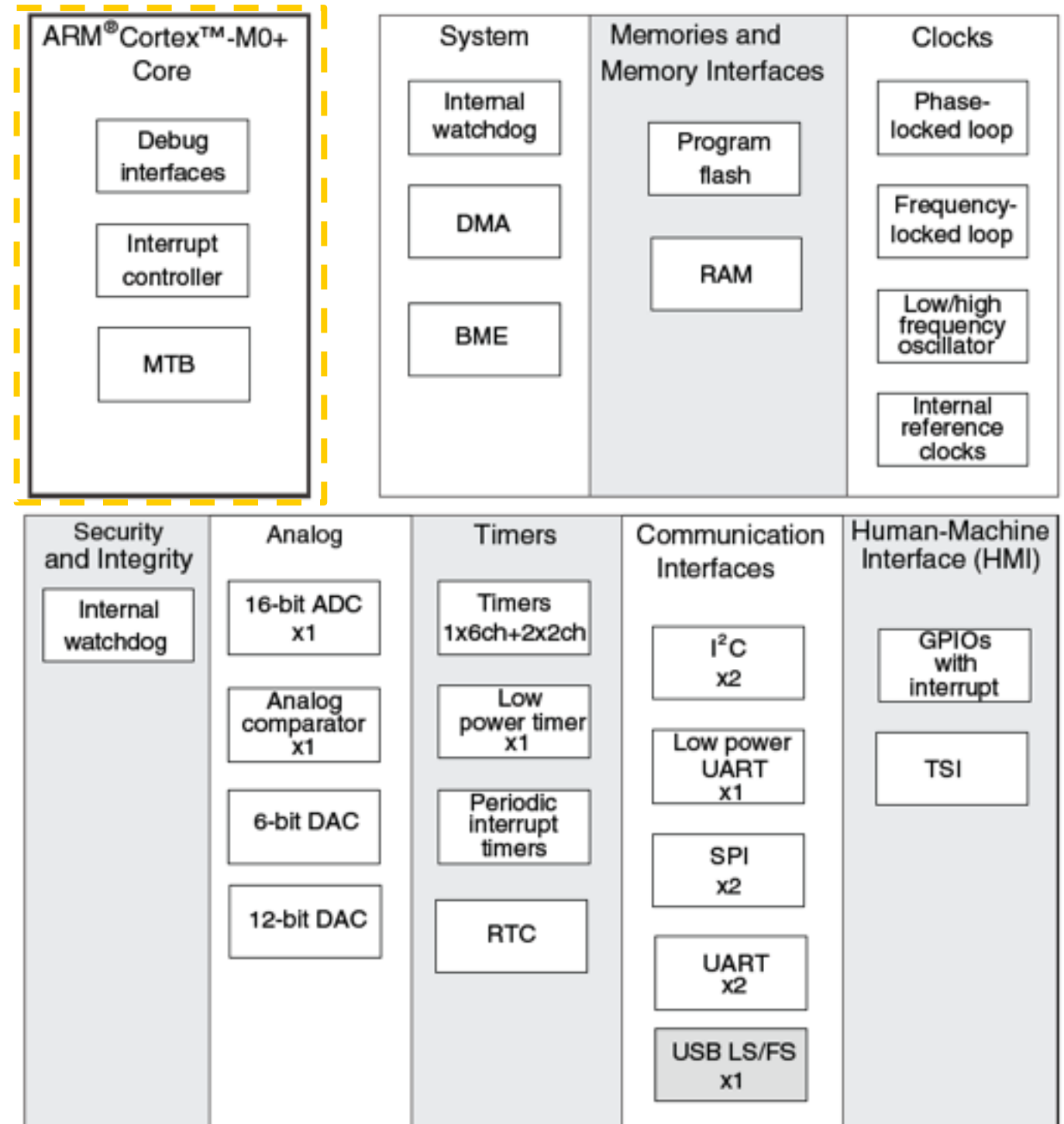
嵌入式CPU

嵌入式CPU——MCU

- 与PC的CPU不同，MCU一定是在CPU之外带有外设
- 至少具有定时器和GPIO
- 一般还具有ADC、UART和PWM等

CPU vs MCU

- 都有执行指令的CPU
- 单片机有外围部件同时来做嵌入式接口和控制
- GPIO
- ADC
- 通信
- 时钟



CPU vs MCU

- CPU：只有指令译码执行部分，不含任何外设，如x86
- MCU：带有外设，至少有定时器和GPIO，一般还有UART、AD等，有的还带有存储器（包括flash和SRAM）
- SoC：带有存储器及某个特定外设（如Wi-Fi、ZigBee）的MCU
- 在嵌入式语境中，CPU指的是MCU，如果要强调不带外设的CPU，则会用MPU

体系与分类

嵌入式CPU分类

- 8位 vs 32位: 51 vs ARM
- CISC vs RISC: 51 vs ARM
- 非ARM vs ARM: 51 vs ARM
- 不带MMU vs 带MMU: cortex-M vs cortex-A

Intel 8051

- CISC架构，典型主频为8~48MHz
- Intel早已不再生产，目前最大的厂家是中国的stc
- 1990年后开始流行带有片内flash和SRAM的型号（Atmel）
- 广泛应用于低端消费类电子产品中
- 许多专用标准产品采用51核
- 许多外设芯片提供51例子代码
- 处理比特级IO任务高效
 - 51的内部总线简单，CPU可以直接控制GPIO
- 价格曾经是优势，现在已不再
- 有商业的C编译器，2000年出现GNU的移植

Atmel的AVR

- 8位RISC，哈佛架构，典型主频8~16MHz
- 片上flash、SRAM和EEPROM
- 第一个有GNU编译器支持的MCU
- 北美大学中最常用的MCU
- CS出身的设计师常用

TI的MSP430

- 16位伪RISC架构
- 低功耗设计是主要优势
- 针对仪表市场，有特殊设计，如LCD驱动
- 只有商业C编译器
- 很少用于校园教学科研

Microchip PIC

- 8位RISC，哈佛架构
- 内存模型复杂
 - 以字为单位编址
 - 复杂的映射
 - 特殊的堆栈空间
- 最初使用BASIC语言来做编程
- 有商业的C编译器
- 在北美的业余电子爱好者中受欢迎
- EE出身的设计师常用

DSP

- 主要是TI和ADI的产品
- 具有特殊的支持傅立叶积分运算的指令
- 普遍为16位、定点运算、字编址
- 音频和视频处理有不同的产品
- 已走入历史
 - Cortex和x86均开始引入DSP指令

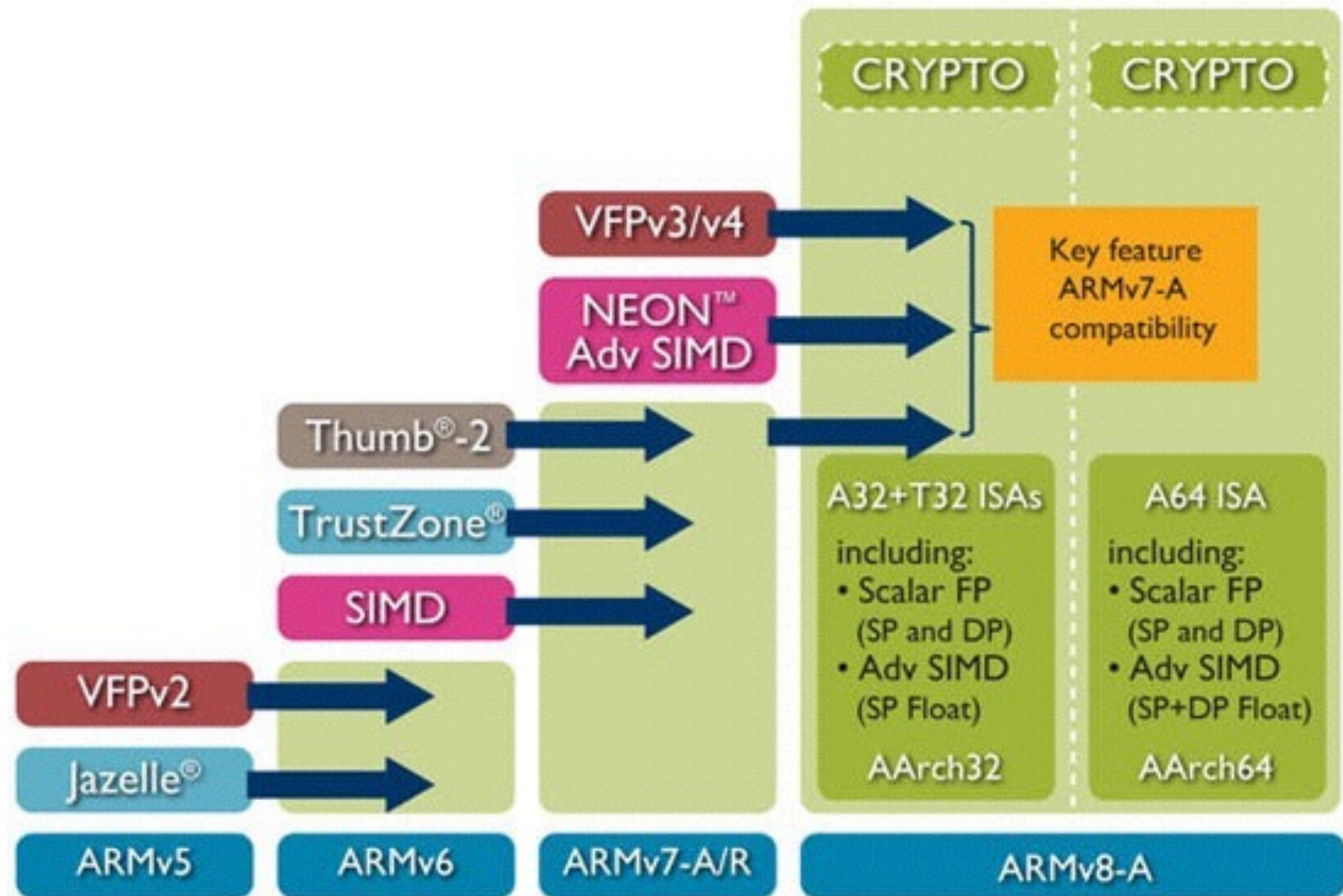
ARM

- Advanced RISC Machine
- 32位，流水线
- 嵌入式市场占有率第一

ARM的分类

- ARM vs Cortex
 - ARM为32位指令，支持16位的Thumb指令，已过时
 - Cortex为增强16位Thumb指令，有部分32位指令
- 没有MMU vs 有MMU
 - ARM和Cortex均有过这两类
 - 目前主要是Cortex-M和Cortex-A的区别

ARM v5-v8的架构



Cortex

- Cortex-A: Application
 - 有MMU，跑Linux/Android/iOS，主频>100MHz，有浮点部件
 - 主要用于手机、平板甚至服务器，目标是取代x86
- Cortex-R: Real-Time
- Cortex-M: Micro-controller
 - 没有MMU，不跑OS或跑RTOS，主频从24MHz~250MHz都有，主流产品75MHz左右，有的支持浮点、DSP甚至多发
 - 低端价格已经低于8位的51

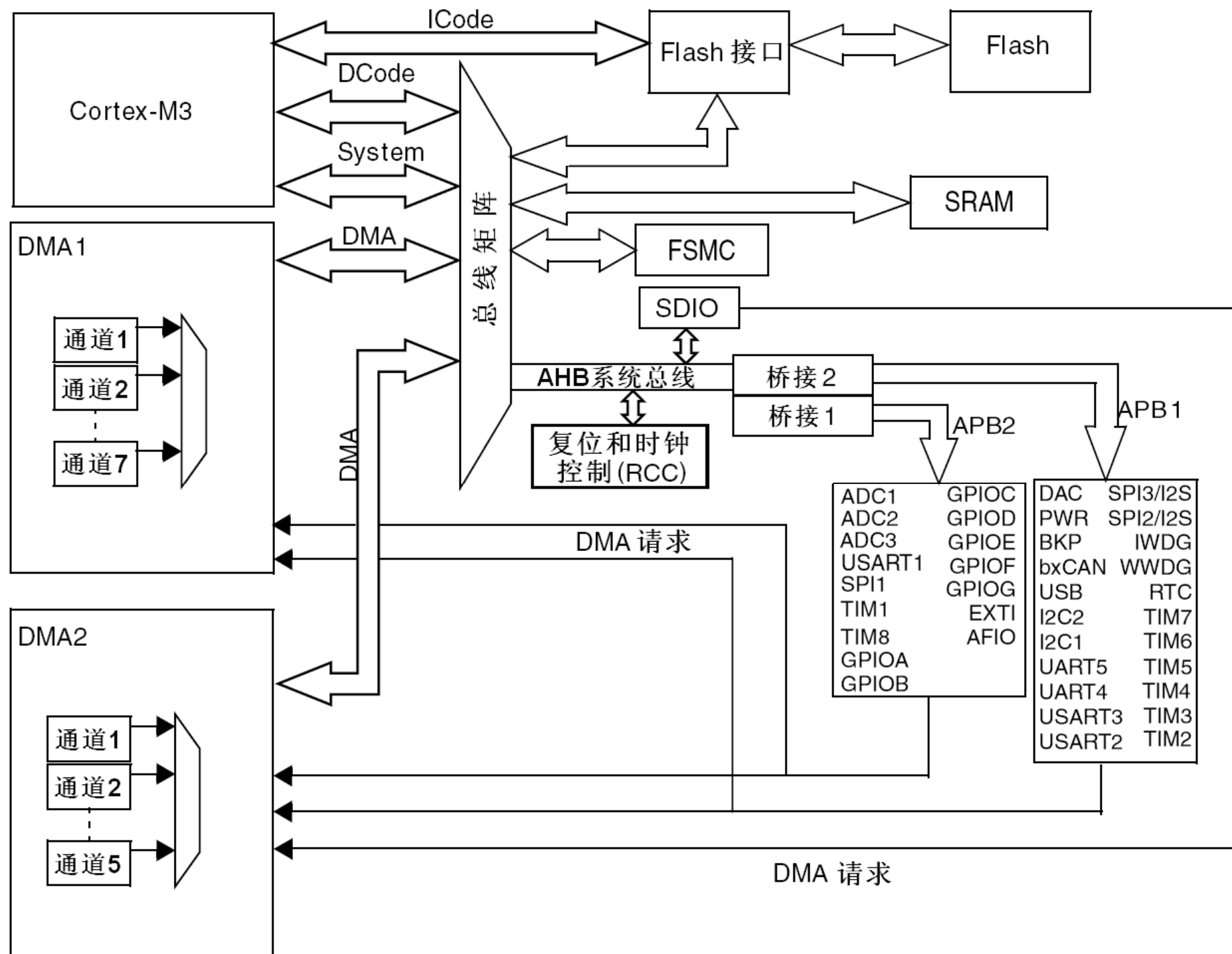
M vs A

- Cortex-M的目标是传统单片机，Cortex-A的目标是平板、手机和桌面及服务器
- Cortex-M的开发过程像单片机
 - 厂家提供器件手册、评估板、参考设计和例子代码，工程师从裸机开始设计
- Cortex-A的开发过程像PC
 - 厂家提供开发好了的底层环境，包括硬件和软件，工程师基于OS和库开始开发
 - 没有器件手册，也不知道内部的特殊功能寄存器

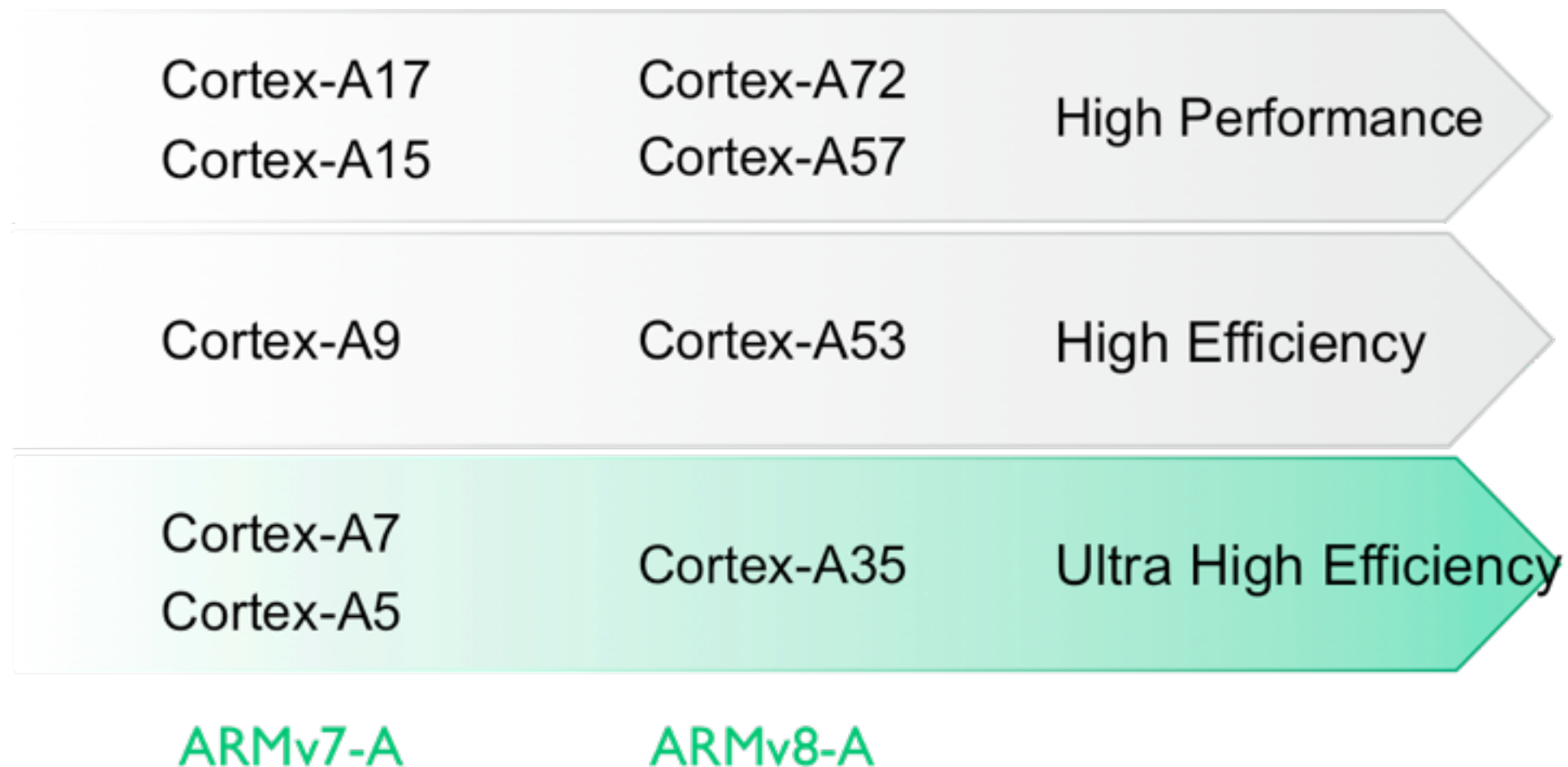
Cortex-M

- M0+: 2级流水线, 56条指令, 单周期32x32位乘法
- M0: 3级流水线
- M1: 支持FPGA
- M3: 除法指令 —> STM32F103
- M4: 3级流水线 + 分支预测, DSP扩展, SIMD
- M7: 6级超标量流水线, 双精度浮点, I/D分离的cache

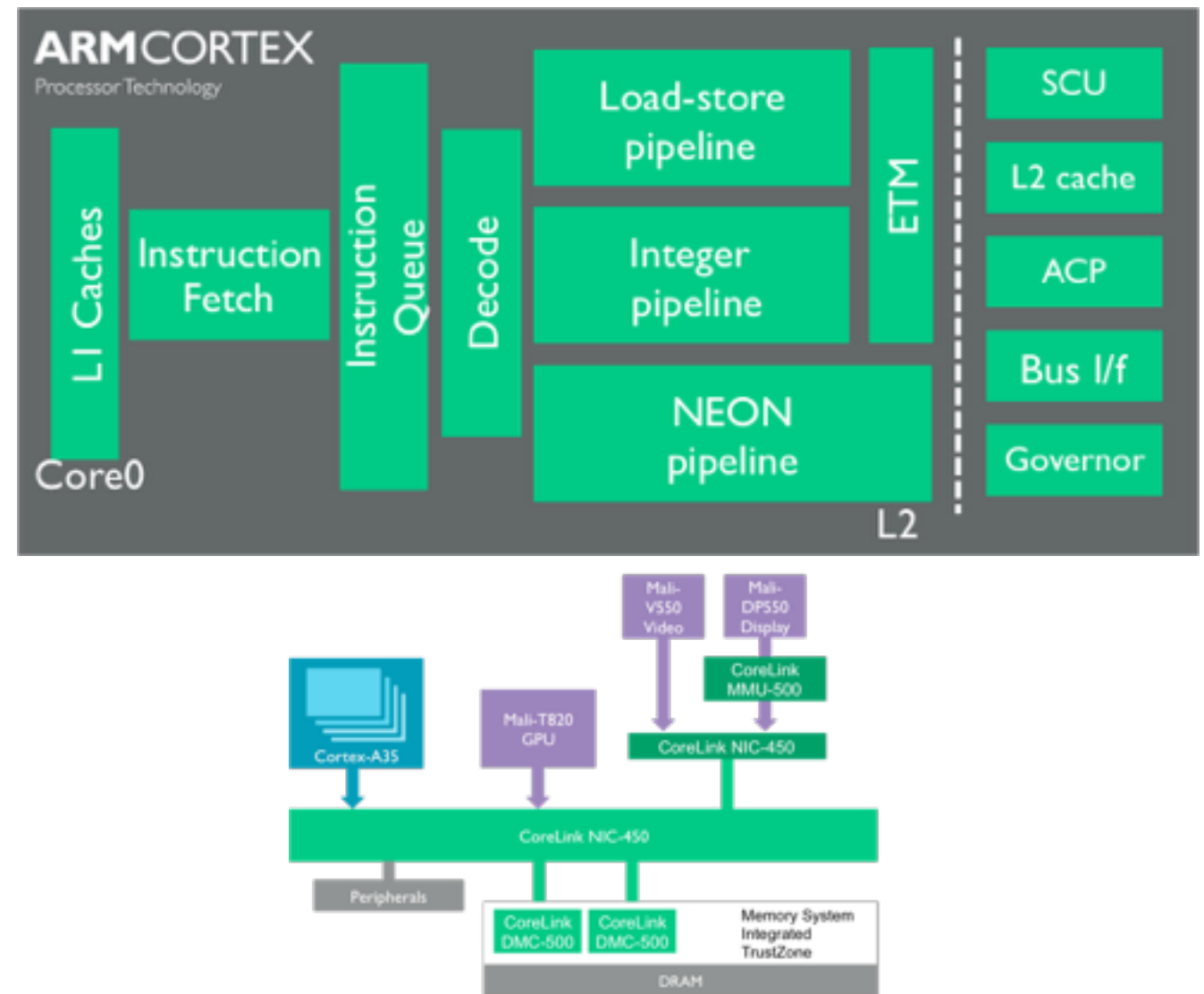
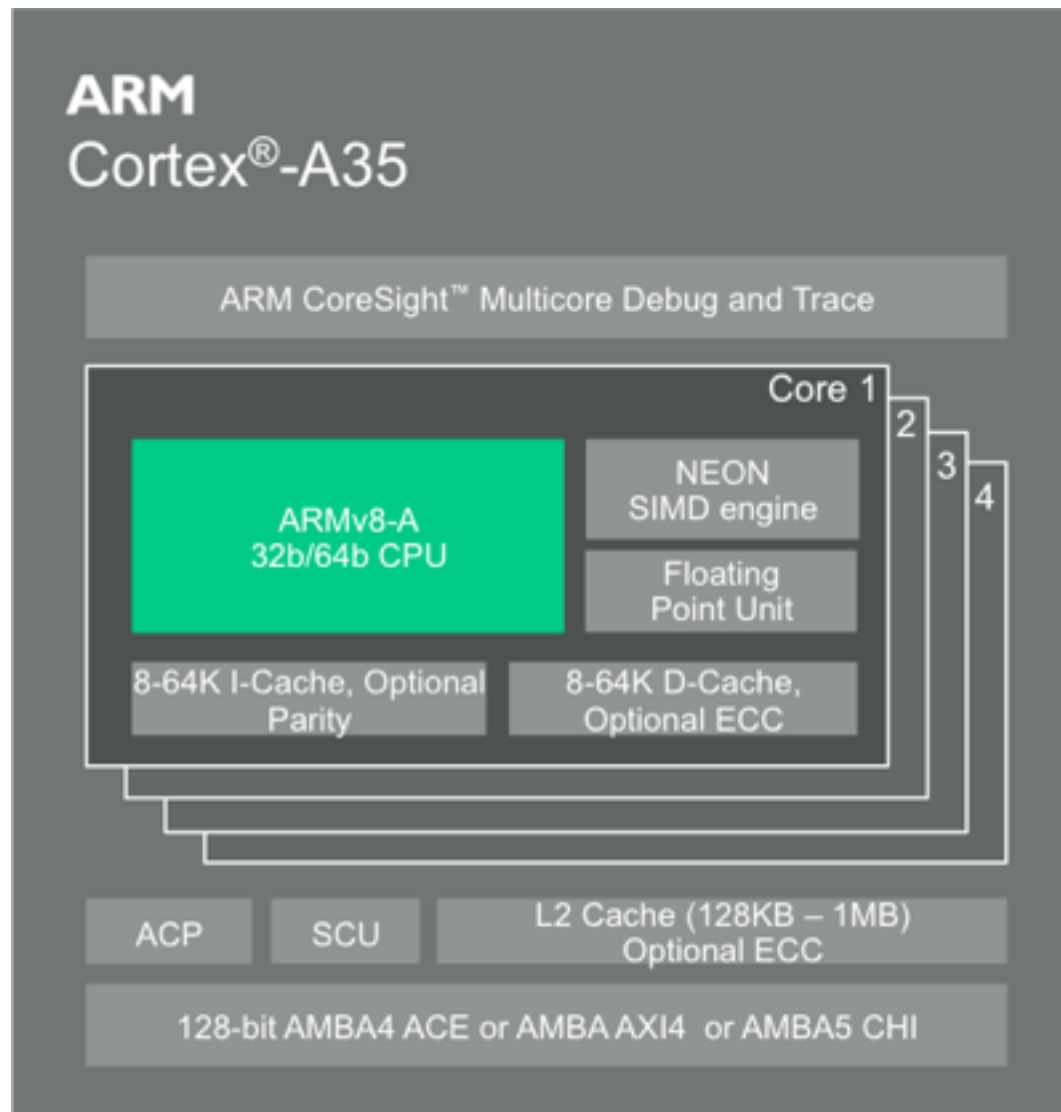
STM32F103



Cortex-A



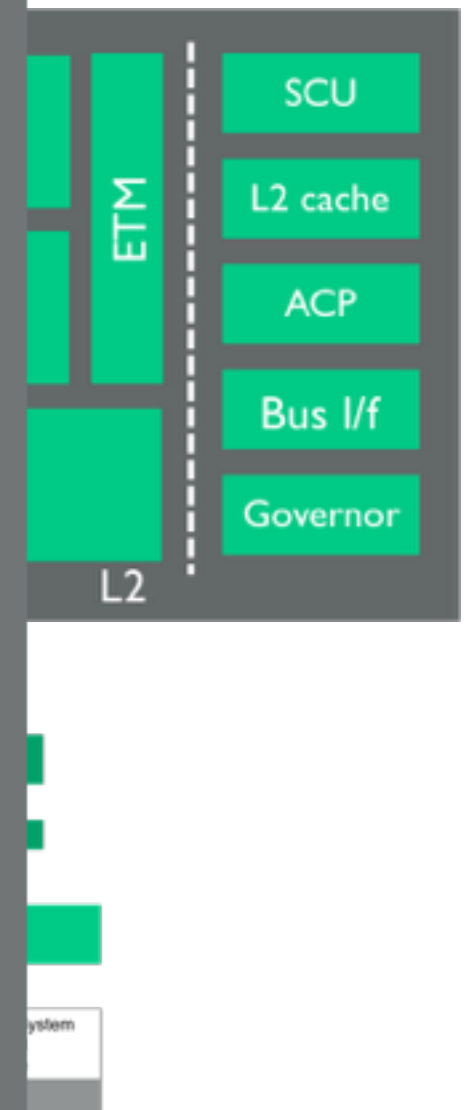
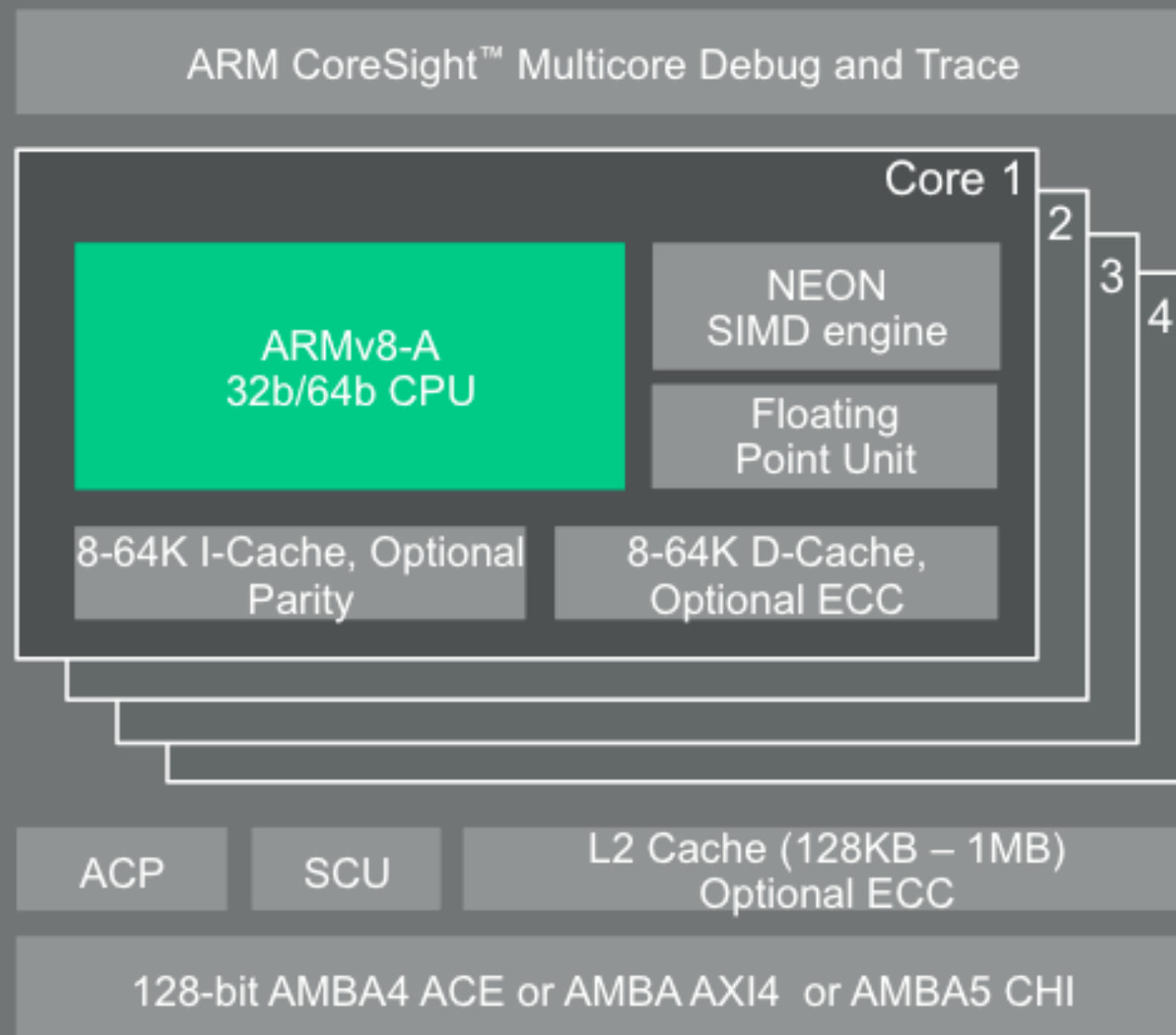
Cortex-A35



- Improvements in efficiency - higher performance and lower power
- New power management capabilities - to lower power consumption and increase battery life
- Full ARMv8-A architecture support – 64-bit capable with backwards compatibility

<https://community.arm.com/groups/processors/blog/2015/11/05/introducing-cortex-a35-arms-most-efficient-application-processor>

ARM Cortex[®]-A35



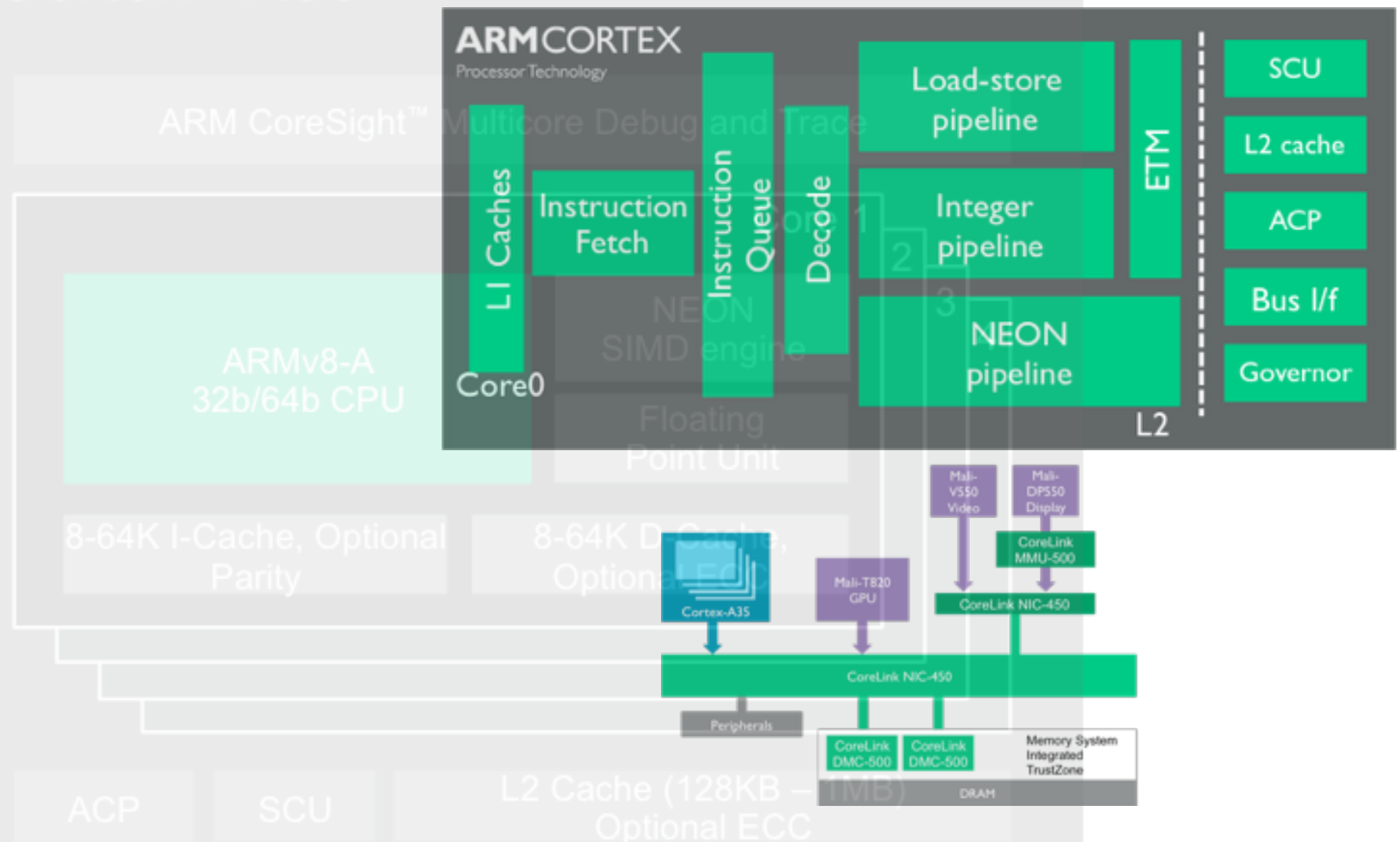
- Improvement
- New power m

base battery life

- Full ARMv8-A architecture support – 64-bit capable with backwards compatibility

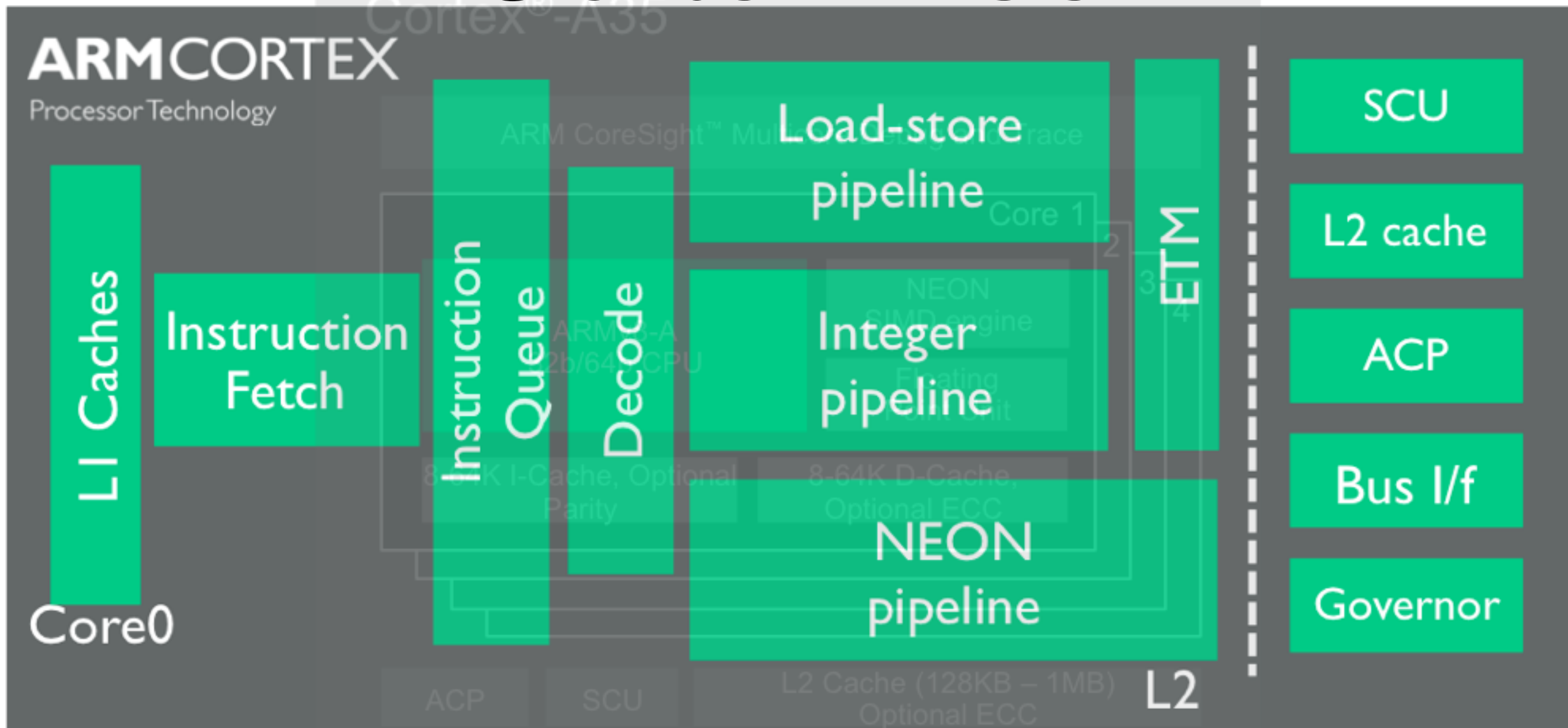
<https://community.arm.com/groups/processors/blog/2015/11/05/introducing-cortex-a35-arms-most-efficient-application-processor>

ARM Cortex-A35



- Improvements in efficiency - higher performance and lower power
- New power management capabilities - to lower power consumption and increase battery life
- Full ARMv8-A architecture support – 64-bit capable with backwards compatibility

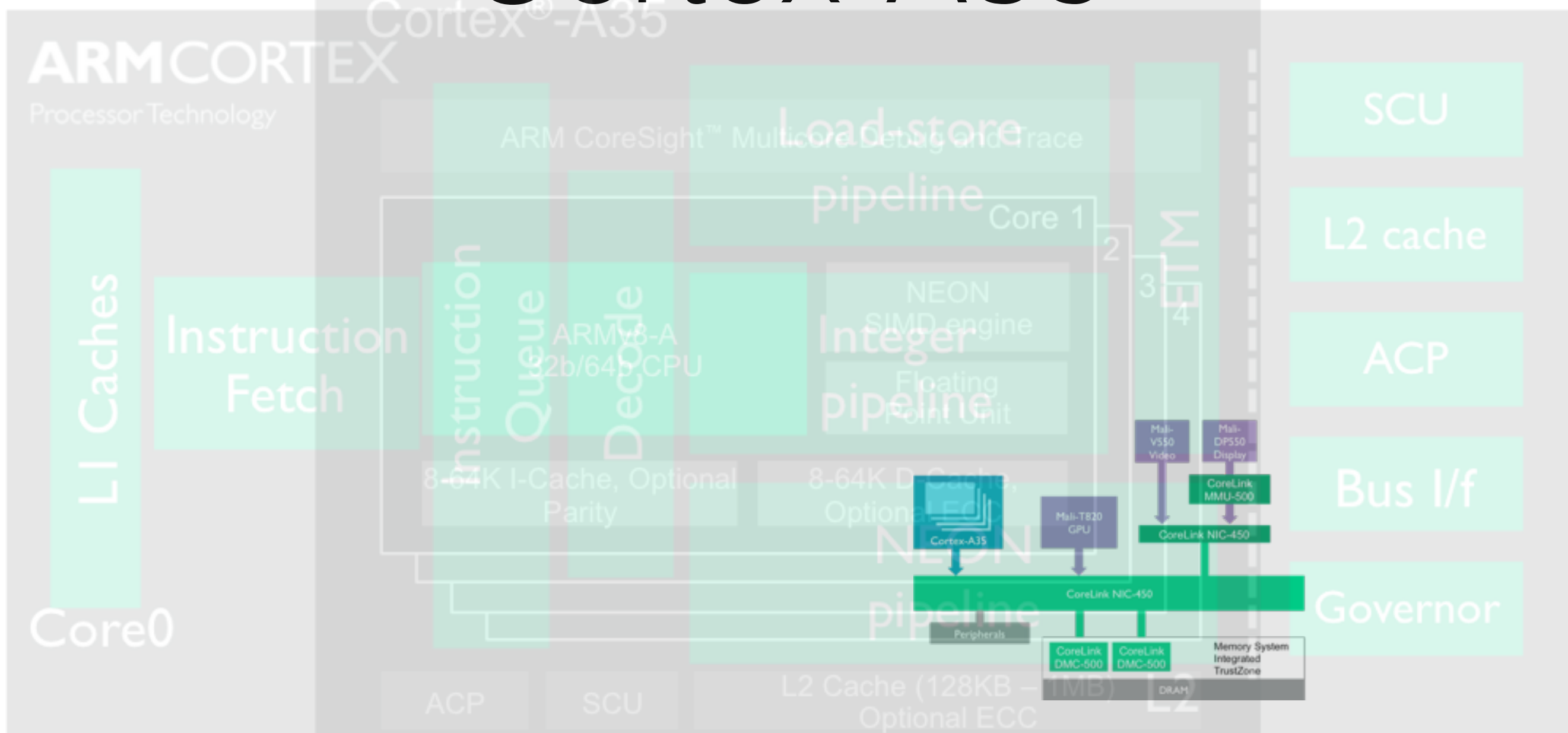
Cortex-A35



- Improvements in efficiency - higher performance and lower power
- New power management capabilities - to lower power consumption and increase battery life
- Full ARMv8-A architecture support – 64-bit capable with backwards compatibility

<https://community.arm.com/groups/processors/blog/2015/11/05/introducing-cortex-a35-arms-most-efficient-application-processor>

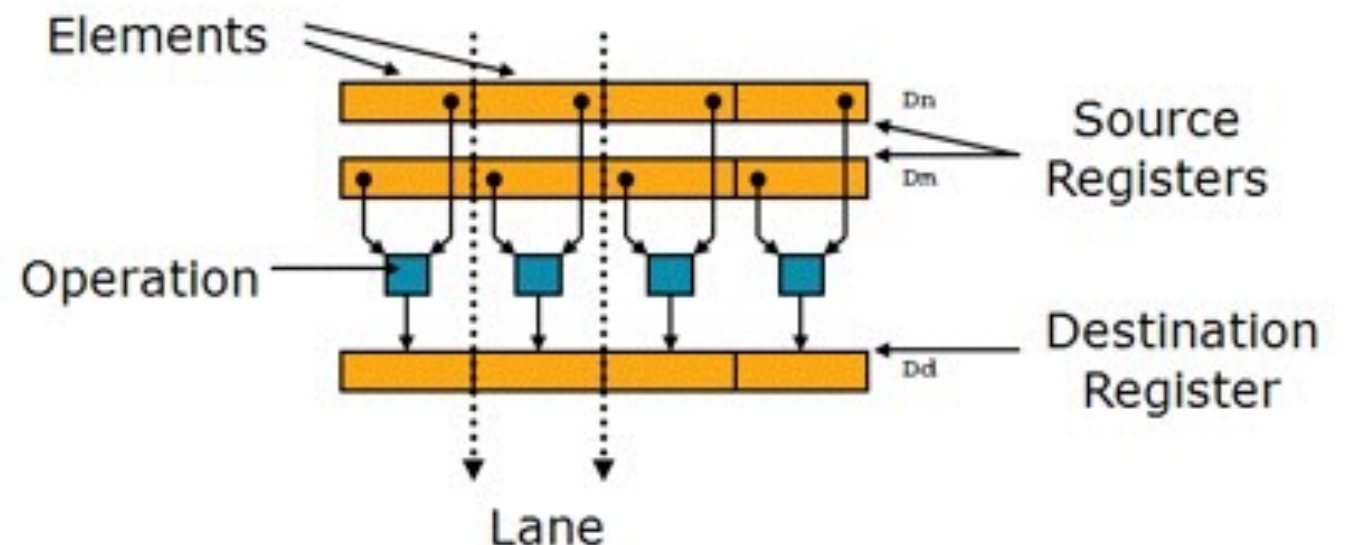
Cortex-A35



- Improvements in efficiency - higher performance and lower power
- New power management capabilities - to lower power consumption and increase battery life
- Full ARMv8-A architecture support – 64-bit capable with backwards compatibility

NEON

- NEON 技术是 ARM Cortex™-A 系列处理器的 128 位 SIMD（单指令，多数据）架构扩展，旨在为消费性多媒体应用程序提供灵活、强大的加速功能，从而显著改善用户体验。它具有 32 个寄存器，64 位宽（双倍视图为 16 个寄存器，128 位宽。）
- NEON 指令可执行“打包的 SIMD”处理：
 - 寄存器被视为同一数据类型的元素的矢量
 - 数据类型可为：有符号/无符号的 8 位、16 位、32 位、64 位单精度浮点
 - 指令在所有通道中执行同一操作



MIPS

- 第一款RISC芯片
 - 也有64位的芯片，也有16位的指令集
- 产品能力<ARM，市场比例小
- 路由器中多见（传统市场）
- 对MIPS的研究在大学教育中比ARM普及
 - MIPS的老大现在是斯坦福的校长，他写了教科书
 - ARM对IP的控制非常严格，研制采用ARM指令集的芯片是违法的

如何选择MCU

选择MCU家族的指标

- 工程师熟悉程度
- 业界评论
- 供货能力
 - 供应商关系
- 价格
- 开发支持：
 - 成熟的方案（成套方案或关键外设方案）
 - 参考设计
 - 编译器
 - 下载方式
 - 功能性能满足要求

下载方式

- 片上存储器
 - JTAG/WDI
 - 专用ICP接口
 - 串口ISP（内置bootloader）
- 外接存储器
 - JTAG
 - 专用烧录工具（焊前/焊后）

功能（资源）

- GPIO：数量、驱动能力
- UART：数量
- AD：分辨率、速度、数量
- 定时器：数量、长度（位数）
- SPI：速度
- I2C：速度
- 其它通信：CAN、USB

性能

- 主频
- 片上flash容量
- 片上SRAM容量
- 外接存储器能力

STM32系列 产品命名规则

示例: **STM32 F 103 C 8 T 6 A xxx**

产品系列

STM32 = 基于ARM®核心的32位微控制器

产品类型

F = 通用类型

产品子系列

101 = 基本型

102 = USB基本型, USB 2.0全速设备

103 = 增强型

105或107 = 互联型

引脚数目

T = 36脚

C = 48脚

R = 64脚

V = 100脚

Z = 144脚

闪存存储器容量

4 = 16K字节的闪存存储器

6 = 32K字节的闪存存储器

8 = 64K字节的闪存存储器

B = 128K字节的闪存存储器

C = 256K字节的闪存存储器

D = 384K字节的闪存存储器

E = 512K字节的闪存存储器

封装

H = BGA

T = LQFP

U = VFQFPN

Y = WLCSP64

温度范围

6 = 工业级温度范围, -40°C~85°C

7 = 工业级温度范围, -40°C~105°C

内部代码

A 或者空 (详见产品数据手册)

选项

xxx = 已编程的器件代号(3个数字)

TR = 卷带式包装

STM32F103

STM32 F1 SERIES - ARM® CORTEX®-M3 FOUNDATION MCUs

Part number	Flash size (Kbytes)	Internal RAM size (Kbytes)	Package	Timer functions		ADC	DAC	I/Os	Serial interface								
				16-bit timers	Others				SPI	I ² S	I ² C	USART + UART ¹	CEC	USB FS	CAN 2.0B	SDIO	E
STM32F103RB	128	20	LQFP64 TFBGA64	4x16-bit	2 x WDG, RTC, 24-bit downcounter	16x12-bit		51	2		2	3		1	1		
STM32F103TB	128	20	VFQFPN36	4x16-bit		10x12-bit		26	1		1	2		1	1		
STM32F103VB	128	20	LFBGA100 LQFP100	4x16-bit		16x12-bit		80	2		2	3		1	1		
STM32F103RC	256	48	LQFP64 WLCSP64	8x16-bit		16x12-bit	2x12-bit	51	3	2	2	3+2		1	1	1	
STM32F103VC	256	48	LFBGA100 LQFP100	8x16-bit		16x12-bit	2x12-bit	80	3	2	2	3+2		1	1	1	
STM32F103ZC	256	48	LFBGA144 LQFP144	8x16-bit		21x12-bit	2x12-bit	112	3	2	2	3+2		1	1	1	
STM32F103RD	384	64	LQFP64 WLCSP64	8x16-bit		16x12-bit	2x12-bit	51	3	2	2	3+2		1	1	1	
STM32F103VD	384	64	LFBGA100 LQFP100	8x16-bit		16x12-bit	2x12-bit	80	3	2	2	3+2		1	1	1	
STM32F103ZD	384	64	LFBGA144 LQFP144	8x16-bit		21x12-bit	2x12-bit	112	3	2	2	3+2		1	1	1	
STM32F103RE	512	64	LQFP64 WLCSP64	8x16-bit		16x12-bit	2x12-bit	51	3	2	2	3+2		1	1	1	
STM32F103VE	512	64	LFBGA100 LQFP100	8x16-bit		16x12-bit	2x12-bit	80	3	2	2	3+2		1	1	1	
STM32F103ZE	512	64	LFBGA144 LQFP144	8x16-bit		21x12-bit	2x12-bit	112	3	2	2	3+2		1	1	1	
STM32F103RF	768	96	LQFP64	12x16-bit		16x12-bit	2x12-bit	51	3	2	2	3+2		1	1	1	
STM32F103VF	768	96	LQFP100	14x16-bit		16x12-bit	2x12-bit	80	3	2	2	3+2		1	1	1	
STM32F103ZF	768	96	LFBGA144 LQFP144	14x16-bit		21x12-bit	2x12-bit	112	3	2	2	3+2		1	1	1	
STM32F103RG	1024	96	LQFP64	12x16-bit		16x12-bit	2x12-bit	51	3	2	2	3+2		1	1	1	
STM32F103VG	1024	96	LQFP100	14x16-bit		16x12-bit	2x12-bit	80	3	2	2	3+2		1	1	1	
STM32F103ZG	1024	96	LFBGA144 LQFP144	14x16-bit		21x12-bit	2x12-bit	112	3	2	2	3+2		1	1	1	

家族内选型基本原则

- 满足资源要求的最小（封装、容量）型号
 - 每件最终产品上省下的每一分钱都是你的利润
- 研制阶段采用相同封装下最大容量的型号

封装

- BGA：体积最小，价格最低，加工成本最高，一般工程师无法焊接，往往用于最终产品
- QFN：手工焊接需要较高技巧，不得已的选择
- QFP：加工成本低，适合手工焊接，常用于研制阶段，也常见于最终产品
- DIP：体积最大，价格最高，不会出现在产品设计中

产品价格组成

- 元件物料成本（BOM成本）： $<50\%$
- PCB成本（两层、4层、6层）
- 加工成本（钢板、贴片、直插）
- 检测成本（烧录、老化、检验）
- 结构成本（外壳、线缆）
- 包装成本

产品价格组成

- 元件物料成本（BOM成本）： $<50\%$
- PCB成本（两层、4层、6层）
- 加工成本（这还没有包括研发、市场、销售和维修成本）
- 检测成本（烧录、老化、检验）
- 结构成本（外壳、线缆）
- 包装成本

资源不足时

- GPIO数量不够：通过移位寄存器（串并转换）扩展低速GPIO
- UART、SPI、I2C不够：通过GPIO软件模拟
- AD不够：
 - 外接I2C或SPI接口的AD芯片
 - 采用CMOS开关矩阵扩展
- 没有DA：用阶梯电阻通过GPIO实现
- 定时器不够：软件扩展

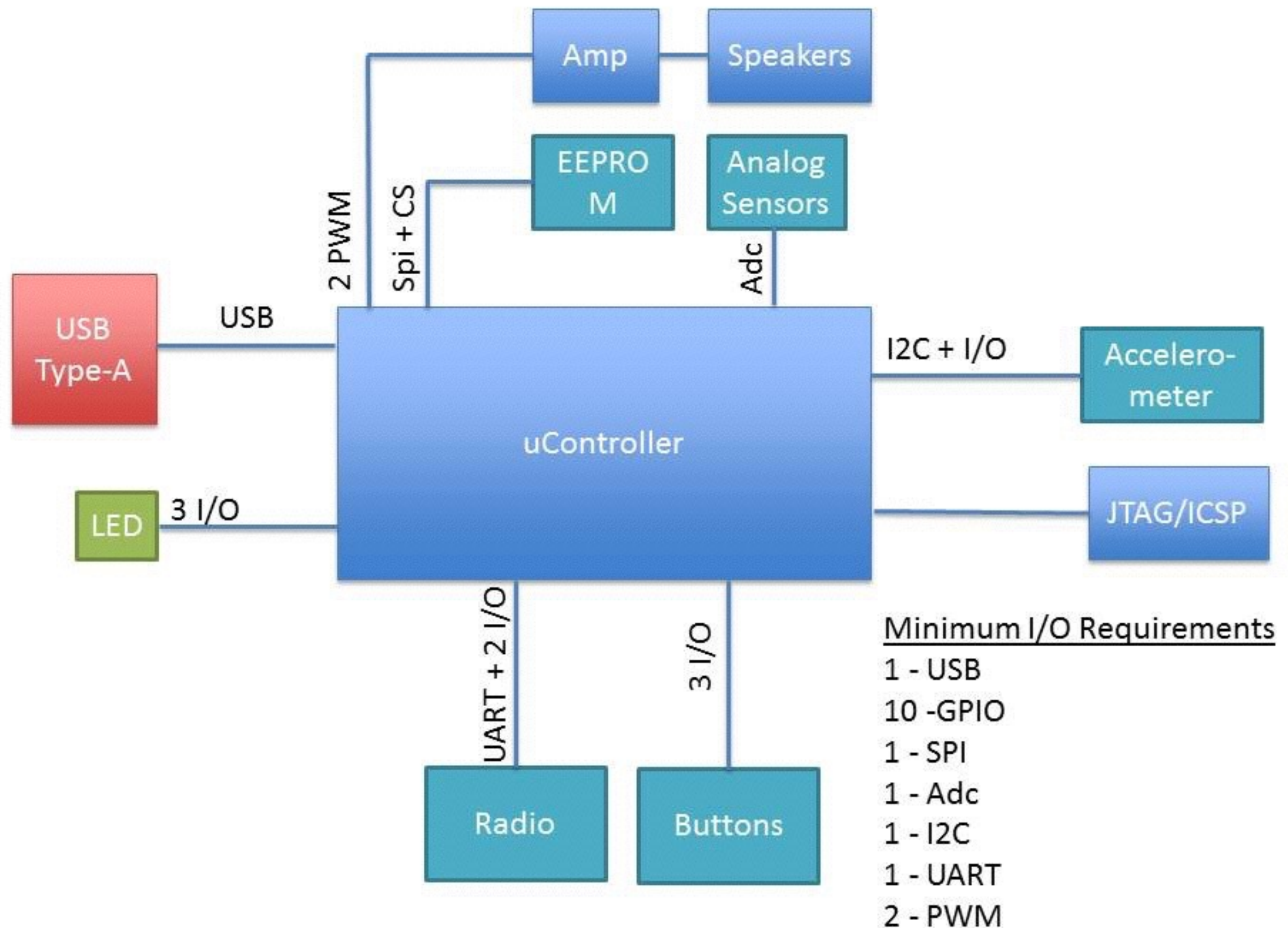
性能不足时

- 速度不够：改进算法挖掘潜力
- SRAM不够：改进算法挖掘潜力
- flash不够：
 - 避免使用浮点：没有浮点硬件的话用的是库
 - 避免使用复杂函数：如printf
 - 链接器是引入整个obj而不是单个函数的：自己写库函数

工作平台

- 一旦确定CPU，最重要的一步就是建立在这块CPU上的工作平台，这包括：
 - 评估板/核心板
 - 编译器、链接脚本、Hello World
 - 串口、AD、定时器等库或测试/示例/基础代码
- 一个企业不可能每个人都很能干，但是一个能干的人也没有一定是不行的

Case Study



10 Steps to Selecting a Micro-controller

1. Make a list of required hardware interfaces
 - Sensing, Control, Communication, Storage
2. Examine the software architecture
 - OS, Floating-Point, Frequency and latency of tasks
3. Select the architecture

4. Identify Memory Needs
5. Start searching for micro-controllers
6. Examine Costs and Power Constraints
7. Check part availability
8. Select a development kit
9. Investigate compilers and tools
10. Start Experimenting