

	CM3	M4K for PIC32	AVR32
计算前导零	CLZ	CLZ, CLO	CLZ
位反转	RBIT	-	BREV
优先级阈值	BASEPRI, 255 级	-	INT0-INT3 可各自关闭, 4 级
位段写访问	BFI, BFC	INS	BFINS
位段读访问	UBFX, SBFX	EXT	BFEXTU, BFEXTS
8051 位寻址	升级成位带别名地址空间	不支持. 必须先读整个字, 再使用 EXT 指令提取长度为 1 个位的位段	不支持. 必须先读整个字, 再使用 bfectu 指令提取长度为 1 个位的位段
8051 位读写	使用位带别名实现 Setb, clrb	-	mems->setb; memc->clrb
硬件乘,除,乘加	支持	支持	支持
硬件除法耗时	最多 12 周期, 典型 4 周期	最多 34 周期, 典型未知	未知
耗时的指令对中断的影响	除法指令放弃再重新开始 多重访问指令中断—继续	未知	未知
双堆栈	MSP/PSP	未知	SP_App/SP_Sys
中断控制器	内置标配 NVIC	外置 VIC	内外混合式 VIC
中断上下文	自动保存若干寄存器		开出若干 banked 寄存器
最多外中断数	256-16=240	96	64 组*32 个/组=32768 个
优先级数	256+3=259 (+3: reset, NMI, hard fault)	7x4=28 级。外置 VIC 提供 7 级主优先级, 每个主优先级中有 4 个子优先级	1+1+1+1=4 级。INT0-INT3, 共 4 级, 每组选择一级
预索引, 后索引, 变址, 预移位	支持	支持	支持
流水线级数	3	5	5
流水线反清洗	IT 指令	“Likely” 机制用于分支指令	可选分支预测
信号量访问	支持位带的互斥访问指令		xchg 指令, 无位带概念
NMI	支持	支持	支持
处理器模式	Handler+thread. 其它模式通过不同类型的系统异常来模拟	User, Kernel, Debug	APP, SVC, EXP, NMI, INT0, INT1, INT2, INT3
特权级+用户级	所有异常为特权级, 非异常时可选特权级或用户级	User 无特权, 其它有特权	应用程序模式只能是用户级 其它特权级
调试异常	调试监视器异常	调试异常	专用的 dbg state
系统调用	SVC 指令	各种 trap	scall 指令
Fault 管理	3 类 faults, 每类 fault 通过各寄存器识别详细信息	为数众多的 faults, 大体对应 CM3 的三大类 faults	通过不同的处理器模式对应各 fault, 分类法类似 CM3
条件自陷	被零除或未对齐自陷成 fault	丰富的条件自陷指令	-
跟踪系统	可选 ITM, ETM, TPIU		集成在调试系统中
周期数管理	-		COUNT/COMPARE 寄存器
周期定时器	内置 SysTick	-	-
存储器管理	可选 MPU	极度简化的 MMU	可选 MPU 或全面的 MMU
cache	需外扩控制器		支持, 支持锁缓存线

协处理器	-	支持	支持
多核支持	弱	中	未知
个人感觉易学度	易	难	A 型易, B 型中。带 DSP, MMU, Cop 与 SIMD 的难
学习资料	丰满	少	丰满
实时操作系统支持	操作系统大量移植, 但大多无法充分挖掘 CM3 的前卫功能。系统特性非常适合写专用操作系统	若操作系统已移植, 但无法充分挖掘 M4K 的前卫功能。系统特性对写专用操作系统的难易度不敢妄加评论	操作系统大量移植, 但大多无法充分挖掘 AVR32 的前卫功能。系统特性比较适合写专用操作系统

新生代 32 位单片机风格处理器的新特性大比较

感觉到单片机迎来了新生代, 我最近肤浅地研究了三款内核, 把我觉得使它们具有青春朝气的各项指标作了一个比较, 如下表所列。大家可以谈谈看法! 由于我对 M4K 和 AVR32 的了解过少, 因此如果有不准确的地方, 还请 in 人们指正。个人感觉, 在 CM3 上写 OS 最容易。