
5.1 计算机中的“ISA”和“ μ arch”各是什么意思？两者之间有何联系？

ISA 指计算机的指令体系结构, μ arch 指微架构。

μ arch 是 ISA 的硬件实现方式。

5.2 请简述哈佛结构的主要优缺点。

优点：将程序和数据分别独立存取，指令可以和数据并行传送；可以消除流水线上取址的资源相关。

缺点：结构更复杂，与外设和外部扩展存储器连接难度更大。

5.3 TCM 与高速缓存 Cache 有什么区别？

TCM 有物理地址，需要占用内存空间，没有 Cache 的不可预测性；一般存放需要快速执行的程序和寄存器数据、局部属性不适合告诉缓存的数据等。

5.5 ARM 指令集、Thumb 指令集和 Thumb-2 指令集之间的主要区别是什么？

Thumb 是 32 位 ARM 指令集中最常用的部分功能经过重新编码和压缩形成的 16 位新指令集，代码密度比 ARM 更高，在 8/16 位的 IO 接口上运行性能比 ARM 更快。

Thumb2 则是结合了 ARM 和 Thumb 指令优点的改进版本。

5.6 MMU 和 MPU 的功能有何异同？

两者都是用于内存管理。

MMU 更为先进，有内存分页转化和虚拟地址到物理地址的转化，支持 MPU 的功能，开销更低，适用于多系统。

5.9 Cortex-M 系列处理器定义的存储器映射关系是固定不变的，这样做有何利弊？

有助于提高基于 cortex-M 系列设备的软件可移植性和代码可重用性；

但可能降低了架构的灵活性，不利于后期调整。

5.10 Cortex-M3 与 Cortex-M4 使用两个堆栈的目的是什么？在中断响应时，程序断点和程序状态寄存器的内容保存在哪个堆栈中？

该处理器有处理模式和线程模式，分别对应不同特权等级和对应的操作；两个堆栈的目的是在线程模式可以使用独立的进程堆栈指针，使应用程序的栈空间和操作系统的栈空间独立，提高可靠性。

断点和状态寄存器保存在主堆栈中。

5.11 Cortex-M3/M4 的 CODE 区选用总线互连矩阵与总线复用器有什么区别？

总线互联矩阵：ICODE 对 FLASH 的取指操作与 D-Code 和 D 对 SRAM 的数据存取操作可以同时进行；

总线复用器：I-CODE 和 D-CODE 对 CODE 区域的访问只能分时复用，不能并行。

5.13 Cortex-M3/M4 从 SRAM 域读取指令执行时有什么缺点？

从 SRAM 区域读取指令效率更低。

5.14 I-Code 和 D-Code 总线全部连接到同一片 Flash 芯片上会有什么问题？

由于无法并行操作，会造成冲突，效率降低。

5.15 私有外设总线（Private Peripheral Bus, PPB）基于哪种总线协议，有何特点？

基于 APB 协议。

此总线是专用的，不用于普通外设。

5.16 如果非特权线程试图访问内核私有区域，将会导致哪一类异常？如果 Cortex-M3 使用了一条 SIMD 运算指令，结果又将如何？

4 内存管理错误。

cortex-M3 没有浮点运算、DSP 等协处理器，会引发 6 用法错误。

- 5.17** 在 Cortex-M3/M4 中，寄存器 R0-R12 有何异同？如果这些寄存器都是空闲的，你觉得首先使用哪些？为什么？
R0-R7 为低位寄存器，R8-R12 为高位寄存器，但都是通用寄存器。
优先用低位寄存器，因为低位使用比较频繁，优先低位可以降低功耗。

- 5.20** 请说明特殊寄存器 PRIMASK 和 FAULTMASK 寄存器的异同。
FAULTMASK 无需主动清理，错误处理程序运行结束返回时，会自动复位 FAULTMASK；
PRIMASK 最低位写入 1 后会屏蔽除复位、NMI 和硬件错误以外所有的系统异常和外部中断。
两者都属于实现 1 位基于优先权一场、中断寄存器。

- 5.22** 某基于 Cortex-M4 的 SOC 芯片共有 64 级外部中断，BASEPRI 寄存器的宽度共有几位？如果想屏蔽所有优先级大于 16 的中断，请写出对 BASEPRI 寄存器进行设置的汇编指令。如果想屏蔽所有优先级大于 0 的中断，又该如何设置？
BASEPRI 7:2 一共 6 位

```
MOV R0, 0x00000004  
MOV BASEPRI, R0
```

```
MOV R0, 0x00000001  
MOV PRIMASK, R0
```

- 5.23** 有人写了一段对 Cortex-M4 的进程栈进行初始化的代码，其中 PSP 的初始值设为 0x8765 4321，并且使用了如下一条语句：“MOVPSP, R0”对 PSP 进行赋值（其中 R0=0x8765 4321）。这样做存在哪些问题？请逐一说明。
堆栈操作以字为单位，PSP 的初始值却没有做到字对齐；
MSP 和 PSP 只能用专门的 MSR、PSR 指令来访问。

- 5.25** 在特权线程模式下如何切换到非特权线程模式？在非特权线程模式下能否采用类似方法切换到特权线程模式？为什么？
特权线程模式下，可以直接修改 CONTROL 寄存器 nPRIV=1，直接进入非特权模式；
非特权模式下，要通过异常状态进入异常处理，在异常处理时修改 nPRIV=0 来进入特权线程模式。

- 5.29** Cortex-M3 存储空间的哪些区域支持位段（bit-band）操作？
SRAM 区域和片上外设区域。

- 5.31** 写出利用位段操作读取 0x4000 1000 的第 3 位的代码。
LDR R0, 0x40020008
LDR R1, [R0]

- 5.32** 存储器访问属性包括哪些？
可缓冲、可缓存、可执行、可共享

- 5.35** Cortex-M 系列处理器不会改变代码的执行顺序，因而不需要存储器屏障指令，这个观点对吗？为什么？
不对。该系列存储器引入缓存，虽然存储器系统不会改变指令执行的顺序，但是顺序执行指令的存储器访问操作完成的时间先后顺序是不定的，需要存储器屏障来保证顺序正确。

- 5.36** 处理器进入异常处理子程序之前保护现场需要把哪些寄存器的值保护起来？
PSR PC LR R0 R1 R2 R3 R12

5.38 解释 Cortex-M 处理器的中断优先级分组机制。

8 位优先级寄存器分为分组优先级和组内优先级。分组优先级对应抢占优先级；当分组优先级相同时，组内优先级更高的异常会优先处理。

5.39 存储器访问属性包括哪些？

向量表重定位用 VTOR 指向向量表的位置，保存向量表对应存储器的偏移量；
处理器可以修改 VTOR 的值来修改向量表的起始位置来实现向量表重定位。

Hint: View this HW github repository at:

https://github.com/cabasky/2021F-Embedded_System_HW