# Cortex M4编程模型

## 工作模式和状态

### 工作状态

有两种工作状态，Thumb状态和Debug状态，如图。

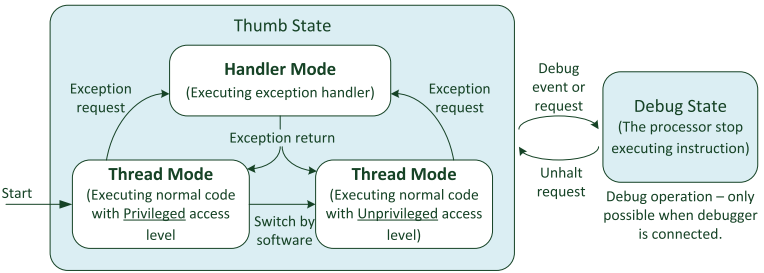


图 1 工作状态和模式

* Thumb状态：当处理器执行Thumb指令的时候，处于Thumb状态。
* Debug状态：当处理器悬停（halted），或触发断点时，进入Debug状态。

### 工作模式

* Handler模式：当执行异常处理例程，如ISR（Interrupt Service Routine）时，处于Handler模式。
* Thread模式：当执行一般的应用程序时，处于Thread模式。根据代码对内存的访问级别，Thread模式又分为特权模式和无特权模式。

分离特权和无特权访问级别，能够帮助系统开发者保护对关键的内存区域的访问。

例如，一个嵌入式系统可能包含一个操作系统内核，这个内核按理应当以特权模式执行；也就是说，操作系统中的应用程序以无特权模式执行。开发者可以通过MPU（Memory Protection Unit）来防止应用程序破坏内核和外设寄存器。这样，当某个程序崩溃时，整个系统仍然能够正常运行。

一般情况下，Cortex M处理器的初始状态为有特权的Thread模式。

## 寄存器

### R0-R12

R0到R12为通用寄存器。其中，R0-R7又被称为低寄存器；大多数16位指令只能访问这些寄存器。R8-R12又被称为高寄存器，可以被32为指令和少数16位指令访问。

### 栈指针（SP）

R13为栈指针，用于PUSH和POP指令访问栈内存。

栈指针在物理层有两个分身：主栈指针和进程栈指针。

* **主栈指针：**Main Stack Pointer, MSP。MSP是默认的栈指针，复位后或者进入Handler模式时，栈指针会被切换为MSP。
* **进程栈指针：**Process Stack Pointer, PSP。只有Thread模式下可以使用PSP。裸机通常不需要PSP；如果使用了嵌入式操作系统，使得内核栈和进程栈分离，这时就有必要使用PSP。

栈指针还有如下性质：

* 在CONTROL寄存器中选择两种栈指针之一。
* 栈指针的最后两位没有意义，这是因为PUSH、POP是对齐到4字节的。
* PSP的初始值是未定义的，MSP的初始值是复位后内存中的第一个dword。

### 链接寄存器（LR）

用来保存被调函数的返回地址。在处理异常的时候，LR会被设置为一个特殊值：EXC\_RETURN，用来触发异常流的返回。

### 程序计数器（PC）

* 读PC：得到当前指令地址+4
* 写PC：引起分支控制流