

6.2.1 工作原理

开始

操作系统和所有使用时基的系统，都必须有一个硬件定时器来产生所需要的“滴答”中断，作为整个系统的时基。滴答中断对操作系统尤其重要。例如，操作系统可以为多个任务许以不同数目的时间片，确保没有一个任务能霸占系统；或者把每个定时器周期的某个时间范围赐予特定的任务等，还有操作系统提供的各种定时功能，都与系统滴答定时器有关。因此，需要一个定时器来产生周期性的中断，而且最好还让用户程序不能随意访问它的寄存器，以维持操作系统“心跳”的节律。

SysTick 是属于 Cortex-M3 内核的组件，是一个 24 位的倒数计数器，即一次最多可以计数 2^{24} 个时钟脉冲。SysTick 是一个倒数计数器，当从某个数倒数到 0 时，将从重装值寄存器中取值作为定时器的初始值，同时可以选择在此时产生中断（异常号：15）。只要不把它在 SysTick 控制及状态寄存器中的使能位清除，就永不停息，即使在睡眠模式下也能继续工作。其原理图如图 6-3 所示。

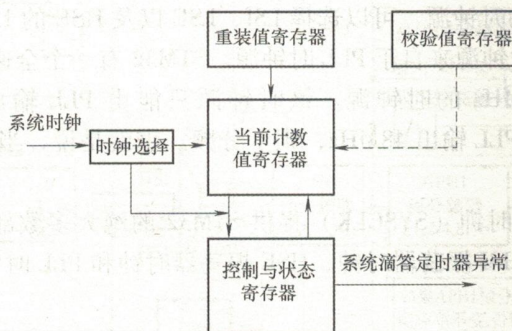


图 6-3 系统滴答定时器原理图

因为所有的 CM3 芯片都带有系统滴答定时器，这使得软件在不同芯片生产厂商的 CM3 器件间的移植工作容易许多。系统滴答定时器的时钟源可以是内部时钟（FCLK，CM3 上的自由运行时钟），也可以是外部时钟（CM3 上的 STCLK 信号）。

结束

6.2.2 使用方法

系统滴答定时器有四个相关的 32 位寄存器，即控制与状态寄存器（STCTRL）、重装值寄存器（STRELOAD）、当前计数值寄存器（STCURR）、校验值寄存器（STCALIB）。ST 公司提供的库函数中，“core_cm3.h”文件中有 SysTick_Type 的结构体如代码 6-1 所示。

代码 6-1 SysTick_Type 结构体

```

1 typedef struct
2 {
3     __IO uint32_t CTRL;           /* !< Offset:0x00 SysTick Control and Status Register */
4     __IO uint32_t LOAD;           /* !< Offset:0x04 SysTick Reload Value Register */
5     __IO uint32_t VAL;            /* !< Offset:0x08 SysTick Current Value Register */
6     __IO uint32_t CALIB;          /* !< Offset:0x0C SysTick Calibration Register */
7 } SysTick_Type;

```

STM32F10x 的 V3.5 版标准库在“core_cm3.h”中还定义了 SysTick_Config 函数，函数代码如代码 6-2 所示。