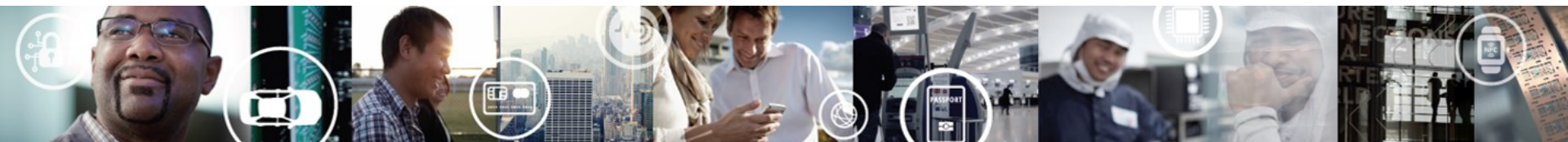


LPC82X 培训资料

计数器/定时器

MAY, 2016



EXTERNAL USE



SECURE CONNECTIONS
FOR A SMARTER WORLD

内容

- 定时器概览
- 窗口看门狗定时器 WWDT
- 自唤醒定时器 WKT
- 多速率定时器 MRT
- SysTick定时器

LPC82X 定时器概览

LPC82X 定时器概览

Timer类型	时钟源	时钟精度	应用方向
SCT	系统时钟	系统时钟精度	普通Timer功能：定时中断，计数捕获，PWM输出
	外部时钟	外部时钟精度	高级Timer功能：实现硬件状态机，复杂波形输出
看门狗WWDT	看门狗振荡器	+/-40%	系统保护复位 低功耗模式唤醒，不能用于Deep power-down模式
唤醒定时器WKT	内部IRC	+/-1.5%	低功耗模式唤醒，任意低功耗模式
	低功耗时钟	+/- 40%	
	外部时钟	外部时钟精度	
MRT	系统时钟	系统时钟精度	产生循环定时中断，单次定时中断
SysTick	系统时钟	系统时钟精度	一般用于产生操作系统tick

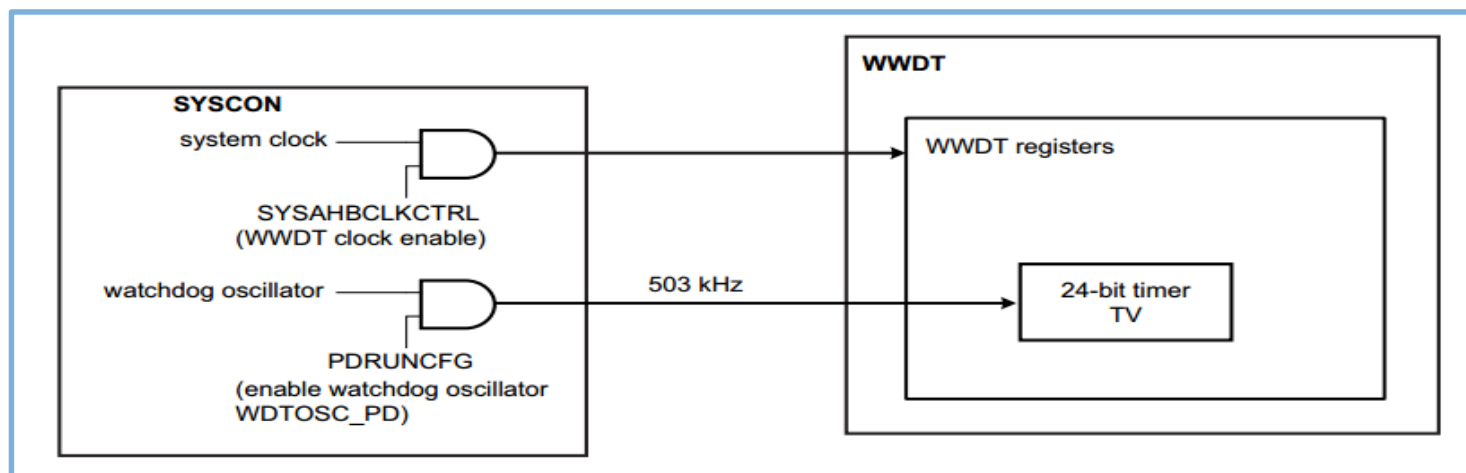
窗口看门狗定时器 WWDT

WWDT特性

- 达到可编程的超时值将产生内部复位信号
- 可选的窗口设置，使得喂狗时间限定在可编程的最小和最大值之间，否则产生看门狗复位
- 超时复位前有警告中断，用户可配置
- 固定的看门狗时钟四分频器，24位向下计数器，可设的超时时间范围($4 \cdot T_{wdclk} \cdot 256 - 4 \cdot T_{wdclk} \cdot 2^{24}$)， T_{wdclk} 可通过看门狗振荡器控制寄存器设置
- 看门狗启动后，只有硬件复位或者看门狗复位才能禁能
- 看门狗定时器可以配置运行在Deep-sleep和Power-down模式下

WWDT时钟和供电设置

- 看门狗模块内部有两个时钟域，分别为看门狗外设时钟和看门狗振荡器时钟，前者源于芯片系统时钟，后者是独立的振荡器时钟，用于驱动看门狗计数器。初始化看门狗外设时，首先要设置好看门狗外设时钟和振荡器供电，前者用于访问外设寄存器，后者使能振荡器时钟输出，驱动看门狗计数器



WWDT时钟和供电设置

• 时钟设置

-使能看门狗外设时钟，以访问WWDT寄存器

▪ `Chip_Clock_EnablePeriphClock(SYSCTL_CLOCK_WWDT);`

-设置看门狗振荡器时钟，该时钟用于驱动看门狗计数器

▪ `Chip_Clock_SetWDTOSC(WDTLFO_OSC_0_60, 64);`

• 供电设置

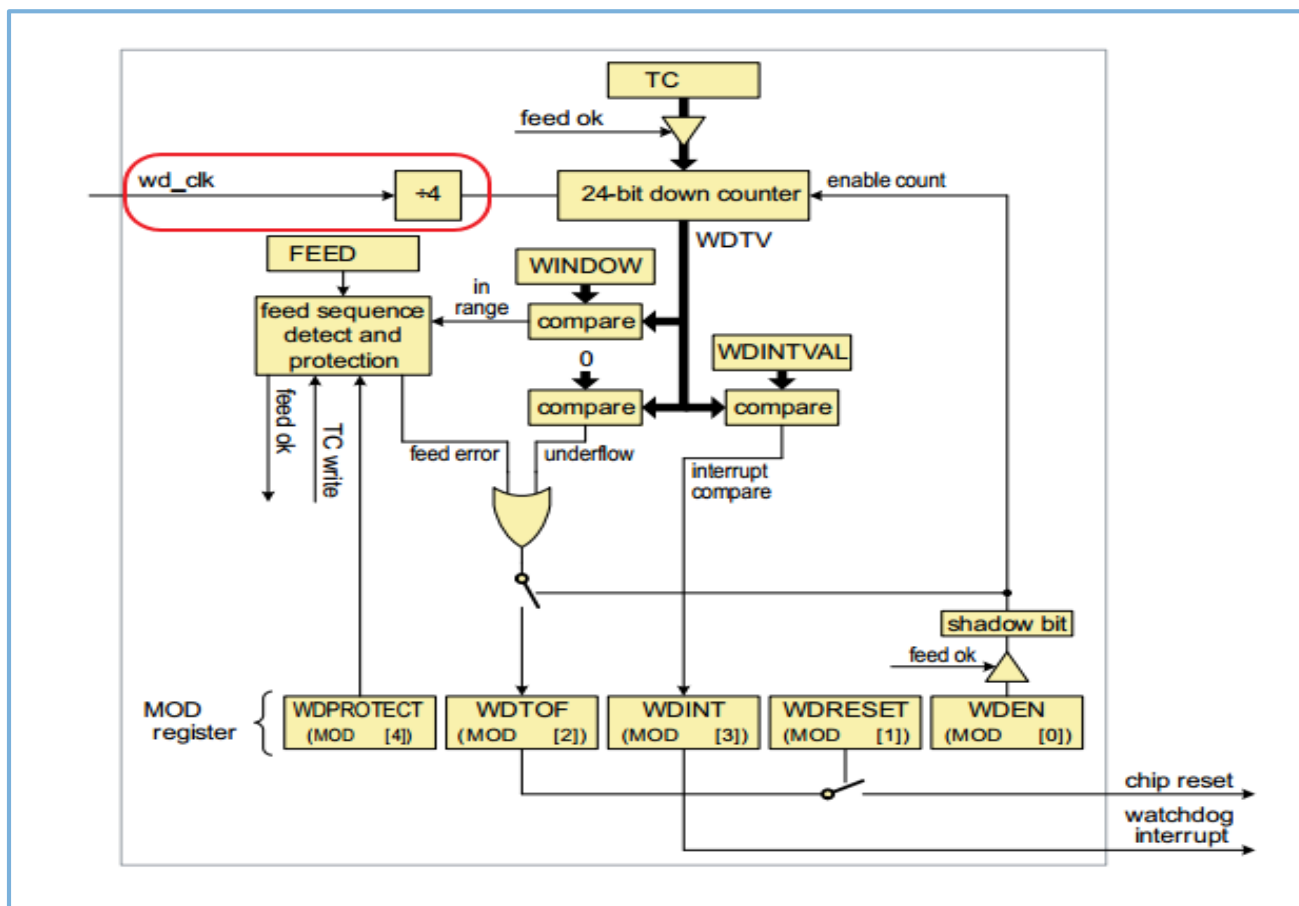
-设置电源配置寄存器PDRUNCFG，使能对看门狗振荡器的供电

▪ `Chip_SYSCTL_PowerUp(SYSCTL_SLPWAKE_WDTOSC_PD);`

PDRUNCFG寄存器位		符号
6		WDTOSC_PD
位值	供电设置	描述
0	Powered	使能对看门狗振荡器的供电
1	Powered down	禁能对看门狗振荡器的供电

WWDT内部框图

- 内部框图显示了看门狗的内部运行原理和机制



WWDT低功耗模式应用

- 看门狗有独立的振荡器时钟，可作为芯片低功耗模式下的唤醒源
- 除了Deep power-down模式，看门狗振荡器可设置运行在Sleep，Deep-sleep和Power-down等低功耗模式下，从而驱动产生看门狗中断唤醒芯片
- 看门狗计数值达到警告中断寄存器设定值时，将产生看门狗外设中断
- 看门狗警告中断寄存器有10位有效值，可设的最大值为1023，所以从警告中断产生到看门狗复位操作的最长可设时间为1023个看门狗计数值，即4096个看门狗时钟
- 看门狗外设中断默认使能，警告中断寄存器复位默认值为0，所以当计数器为0时，将产生看门狗外设中断，看门狗模式寄存器中断标志位置位

WWDT低功耗模式应用 唤醒设置

第一步

- 进入低功耗模式前，设置看门狗定时器常数寄存器TC和看门狗警告中断寄存器WARNINT，确定唤醒时间
 - Chip_WWDT_SetTimeOut(LPC_WWDT, wdtFreq * 2);
 - Chip_WWDT_SetWarning(LPC_WWDT, 512);

第二步

- 在寄存器PDSLEEPCFG中进行相应配置
- 在寄存器PDAWAKECFG中进行相应配置

第三步

- 使能看门狗中断在低功耗模式下唤醒内核
 - Chip_SYSCCTL_EnablePeriphWakeup(SYSCCTL_WAKEUP_WWDTINT);

第四步

- 使能看门狗振荡器在低功耗模式下运行
 - Chip_SYSCCTL_SetDeepSleepPD(SYSCCTL_DEEPSLP_WDTOSC_PD);

WWDT使用注意事项

- 看门狗计数器时钟固定为振荡器时钟输出的四分频，时钟精度为40%
- 看门狗模式寄存器MOD中的看门狗使能位WDEN、复位使能位WDRESET、保护位WDPROTECT和锁定位LOCK，使能后软件将无法清零，即看门狗启动后将一直运行，直到芯片复位操作
- 保护模式下，计数器常数寄存器只能在看门狗计数值低于警告中断寄存器和窗口寄存器值时更改，否则将产生看门狗复位
- 喂狗操作将加载TC寄存器值到看门狗计数器
- 固定的喂狗序列，顺序写0xAA和0x55到看门狗Feed寄存器，如果在写入0xAA和0x55之间有其他看门狗寄存器操作，且看门狗处于使能状态，将立即产生看门狗复位
- 使能看门狗后，必须有正确的喂狗操作才会启动计数器
- 设置看门狗窗口寄存器后，如果喂狗操作发生在看门狗计数值大于窗口值时，将产生看门狗复位。窗口值默认为计数最大值

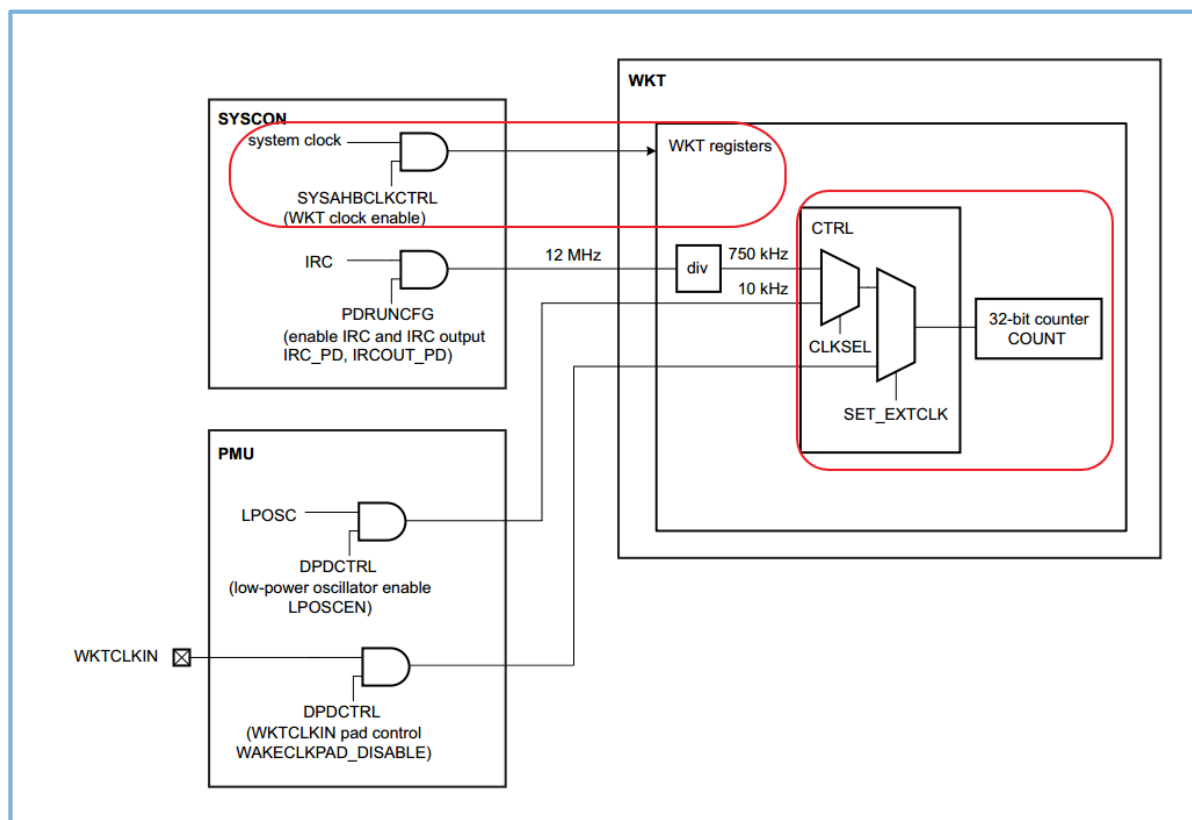
自唤醒定时器 WKT

WKT特性

- 32位向下计数器，写入计数值自动开始计数，计数超时后产生中断或者唤醒芯片
- WKT模块处于独立的常开供电区域
- 三种时钟源可选：IRC，内部低功耗振荡器或者外部时钟（通过WKTCLKIN引脚接入）
- 低功耗振荡器模式和外部时钟模式可用于低功耗模式下唤醒，包括Deep power-down模式

WKT时钟配置 内部时钟框图

- WKT内部有两个时钟域：
 - 内部寄存器接口时钟
 - 32位计数器时钟



WKT时钟配置

- WKT寄存器接口时钟使能

- Chip_Clock_EnablePeriphClock(SYSCTL_CLOCK_WKT);

SYSAHBCLKCTRL寄存器位		符号
9		WKT
位值	时钟控制	描述
0	禁能WKT外设时钟	设置WKT外设时钟的开关
1	使能WKT外设时钟	

WKT计数器时钟配置

• 三种输入时钟源比较

时钟源	低功耗模式下应用	时钟频率	精度
IRC	只可用于Sleep模式	750KHz	高, $\pm 1.5\%$
低功耗振荡器时钟	任意低功耗模式, 包括Deep Power-down	10KHz	差, $\pm 40\%$
外部时钟输入	任意低功耗模式, 包括Deep Power-down	根据输入时钟源	根据输入时钟源

WKT计数器时钟源设置

- 设置WKT控制寄存器CTRL来选择时钟源
 - Chip_WKT_SetClockSource(LPC_WKT, WKT_CLKSRC_10KHZ);

CTRL寄存器位	符号	
0	CLKSEL	
位值	时钟选择	描述
0	IRC时钟	选择IRC时钟作为WKT时钟源, SEL_EXTCLK位清零时有效
1	低功耗时钟	选择低功耗时钟作为WKT时钟源, SEL_EXTCLK位清零时有效
3	SEL_EXTCLK	
位值	时钟选择	描述
0	内部时钟	选择内部时钟作为WKT时钟源, CLKSEL位确定内部时钟源
1	外部时钟	选择外部WKTCLKIN引脚时钟输入作为WKT时钟源

使能IRC时钟

- 设置电源控制寄存器PDRUNCFG，使能对IRC的电源
 - Chip_SYSCTL_PowerUp(SYSCTL_SLPWAKE_IRCOUT_PD);
 - Chip_SYSCTL_PowerUp(SYSCTL_SLPWAKE_IRC_PD);

PDRUNCFG寄存器位		符号
0		IRCOUT_PD
位值	电源开关控制	描述
0	Powered	使能对IRC输出的电源控制
1	Powered down	禁能对IRC输出的电源控制
1		IRC_PD
位值	电源开关控制	描述
0	Powered	使能对IRC振荡器的电源控制
1	Powered down	禁能对IRC振荡器的电源控制

使能低功耗振荡器时钟

- 设置Deep power down控制寄存器DPDCTRL，使能低功耗振荡器
-Chip_PMU_SetPowerDownControl(LPC_PMU, PMU_DPDCTRL_LPOSCEN);
- 如果WKT不作为Deep power down模式下的唤醒源，应该禁能LPOSCDPDEN，以节省能耗
-Chip_PMU_ClearPowerDownControl(LPC_PMU, PMU_DPDCTRL_LPOSCDPDEN);

DPDCTRL寄存器位		符号
2		LPOSCEN
位值	使能/禁能控制	描述
0	禁能	禁能低功耗振荡器
1	使能	使能低功耗振荡器
3		LPOSCDPDEN
位值	使能/禁能控制	描述
0	禁能	在Deep power down模式下，禁能低功耗振荡器
1	使能	在Deep power down模式下，使能低功耗振荡器

使能外部时钟输入

- 设置Deep power down控制寄存器DPDCTRL，使能外部时钟输入

-Chip_PMU_SetPowerDownControl(LPC_PMU, (1 << 5));

DPDCTRL寄存器位		符号
5		WAKECLKPAD_DISABLE
位值	使能/禁能控制	描述
0	禁能	禁能WKT外部时钟输入
1	使能	使能WKT外部时钟输入

WKT低功耗模式应用

- WKT配置为低功耗振荡器模式或者外部时钟模式时，可以工作在任意低功耗模式下，作为芯片的唤醒源
- 低功耗模式唤醒配置事项：
 - 使能WKT外设时钟，设置WKT控制寄存器选择低功耗振荡器时钟或者外部时钟
 - `Chip_Clock_EnablePeriphClock(SYSCTL_CLOCK_WKT);`
 - `Chip_WKT_SetClockSource(LPC_WKT, WKT_CLKSRC_10KHZ);`
 - 设置Deep power down控制寄存器，使能低功耗时钟或者外部时钟
 - `Chip_PMU_SetPowerDownControl(LPC_PMU, PMU_DPDCTRL_LPOSCEN);`
 - 设置使能中断和唤醒
 - `NVIC_EnableIRQ(WKT_IRQn);`
 - `Chip_SYSCTL_EnablePeriphWakeup(SYSCTL_WAKEUP_WKTINT);`
 - 根据选择时钟频率和唤醒时间，加载WKT计数器寄存器，启动计数
 - `Chip_WKT_LoadCount(LPC_WKT, count)`

多速率定时器 MRT

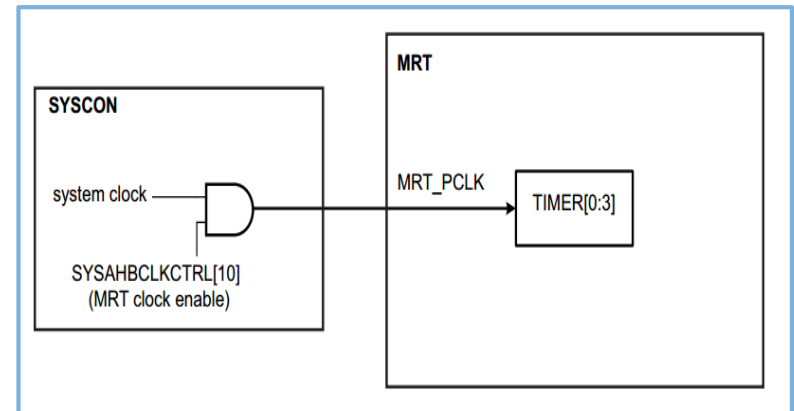
MRT特性

- 31位中断定时器
- 4个独立通道的向下计数器，计数值独立设置
- 3种可配置工作模式：重复中断触发、单次总线停止、单次中断触发

MRT时钟设置

- MRT时钟框图
- MRT外设时钟驱动MRT定时器，外设时钟即为系统时钟，设置系统时钟控制寄存器，使能MRT外设时钟
 - Chip_Clock_EnablePeriphClock(SYSCTL_CLOCK_MRT);

SYSAHBCLKCTRL		
寄存器位	符号	
10	MRT	
位值	时钟控制	描述
0	禁能MRT外设时钟	设置MRT外设时钟的开关
1	使能MRT外设时钟	



MRT工作模式

- MRT模块共有四个独立的MRT定时器，分别有独立的寄存器列表，可设置工作在不同的模式
- MRT共有三中工作模式可设置
 - 重复中断触发模式
 - 单次中断触发模式
 - 单次总线停止模式
- 设置MRT控制寄存器CTRL[0:3]选择工作模式
 - Chip_MRT_SetMode(LPC_MRT_CH0, MRT_MODE_REPEAT);

CTRL寄存器位	符号	
2: 1	MODE	
位值	工作模式	描述
0x0	重复中断触发模式	设置MRT工作在重复中断触发模式
0x1	单次中断触发模式	设置MRT工作在单次中断触发模式
0x2	单次总线停止模式	设置MRT工作在单次总线停止模式

MRT重复中断触发模式

- 根据定时间隔寄存器INTVAL[0:30]的设定值IVALUE，将 IVALUE - 1值加载到计数器，计数器开始向下计数，计数到0时触发中断，间隔寄存器IVALUE - 1值重新自动加载到计数器，继续开始计数，如此多次循环触发中断
- 设置定时间隔寄存器，配置中断触发间隔值及计数器加载方式

INTVAL寄存器位	符号	
30:0	IVALUE	
位值	描述	
0 - $2^{31}-1$	设置MRT计数间隔，将被加载到MRT计数器，并从IVALUE-1开始向下计数。 如果计数器处于空闲状态时，写非0到该值将立即启动计数器。 如果在计数器运行时写0到该值： 若LOAD = 0，则计数到0时计数器停止；若LOAD = 1，则计数立即停止	
INTVAL寄存器位	符号	
31	LOAD	
位值	计数器加载方式	描述
0	非强行加载	重发触发模式下，计数器完成计数间隔后，加载IVALUE-1值
1	强行加载	即使计数器在运行状态下，IVALUE-1值也将加载到计数器

MRT单次中断触发模式

- 根据定时间隔寄存器设定的计数间隔，计数器从IVALUE - 1值开始向下计数，计数到0时产生中断，并停止计数器，计时器将处于空闲状态
- 计数器处于空闲状态时，写非0值到时间间隔寄存器IVALUE，将立即启动计数，从IVALUE - 1值开始向下计数
- 定时间隔寄存器的LOAD值设定同重复中断触发模式

MRT单次总线暂停模式

- 计数器空闲状态下，写非0值IVALUE到定时间隔寄存器将触发计数器开始向下计数
- 计数器启动后，将暂停总线及所有CPU活动，计数器向下计数到0，计数间隔到时后，并不产生中断，而是恢复总线活动，计数器也将停止，处于空闲状态
- 总线停止模式可以用于暂停程序执行，实现设定的延时。相较于软件循环或者定时器查询实现的延时，可以节省CPU运行的功耗
- 该模式只适用于较短的程序延时，微秒级或者10到50个时钟周期，较长时间的延时应该选用单次中断触发模式，因为在单次总线停止模式下，所有总线活动被暂停，任何中断服务程序都无法执行
- 实际的总线停止时间为 $IVALUE + 3$ 个系统时钟周期

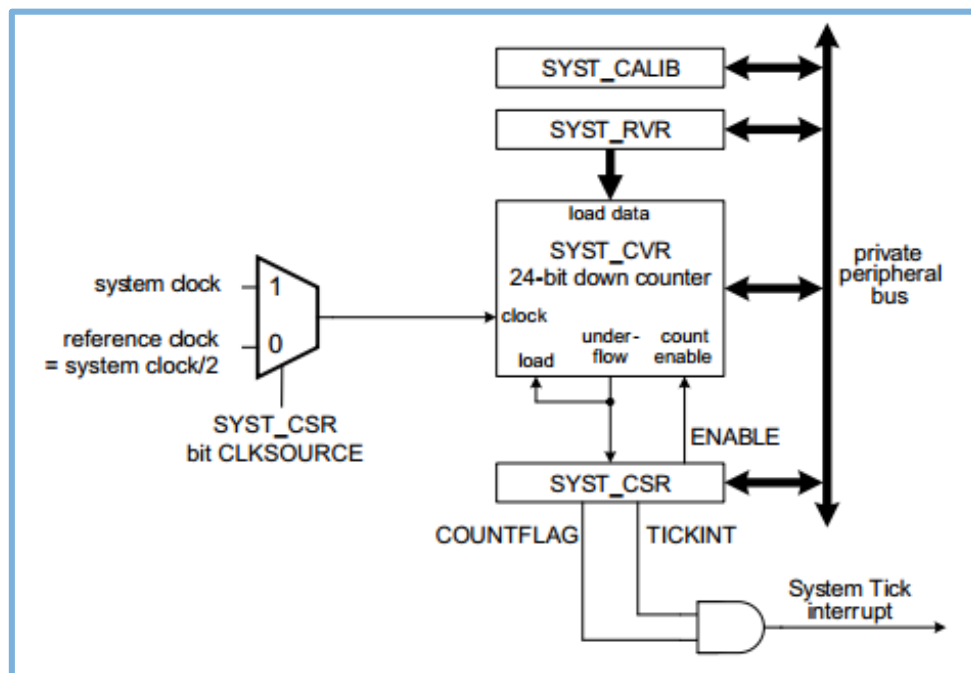
SYSTICK定时器

SysTick特性

- 24位向下计数器
- 可配置选用系统时钟或者1/2系统时钟驱动
- 集成于Cortex内核中，方便软件移植，主要用于实时操作系统或者系统管理软件，作为系统时钟

SysTick内部框图

- SysTick控制和状态寄存器(SYST_CSR)设置时钟源为系统时钟或者系统时钟二分频
- 计数器向下计数为0时，自动加载重载寄存器(SYST_RVR)值，并置位SYST_CSR的COUNTFLAG位，如果中断使能位TICKINT有效的话，产生SysTick中断



SysTick时钟设置

- 设置SysTick控制和状态寄存器SYST_CSR，选择时钟源

SYST_CSR寄存器位	符号	
2	CLKSOURCE	
位值	时钟源选择	描述
0	选择系统时钟	选择SysTick时钟源为系统时钟或者系统时钟二分频
1	选择系统时钟二分频	

SysTick使用设置

第一步

- 根据系统时钟频率和中断间隔要求，设置重载寄存器 (SYST_RVR)

第二步

- 写任意值到计数器(SYST_CVR)将其清零

第三步

- 设置控制和状态寄存器(SYST_CSR)选择时钟源，并使能 SysTick计数器和计数器中断

- 注：重载寄存器的设定值应为需求中断间隔值(Tick)-1



SECURE CONNECTIONS
FOR A SMARTER WORLD