

Sonix (松翰) 8bit 单片机

26 系列 I/O 型原理及基础课件 (三)

复位/系统时钟/电源的管理 (低功耗模式) 系统操作模式

主要内容：

1. 复位，看门狗定时器
2. 系统时钟 (晶振输入，RC 输入，外部时钟输入)
3. 电源的管理 (低功耗模式) 系统操作模式

为了让更多的工程师更加方便、快捷的了解和使用 SONIX 单片机，从而我们编写 SONIX 单片机系列的培训课件，主要详细的介绍了 SONIX 26 系列芯片的硬件模块、指令以及开发环境，供业界工程师交流经验，由于时间仓促，请提出宝贵的意见。

SONIX 官方网站：<http://www.sonix.com.tw>

Sonix 单片机咨询邮箱：howard_tone@hotmail.com

Sonix 单片机咨询电话：13725134515

SONIX 单片机应用推广中心忠心竭诚为你服务

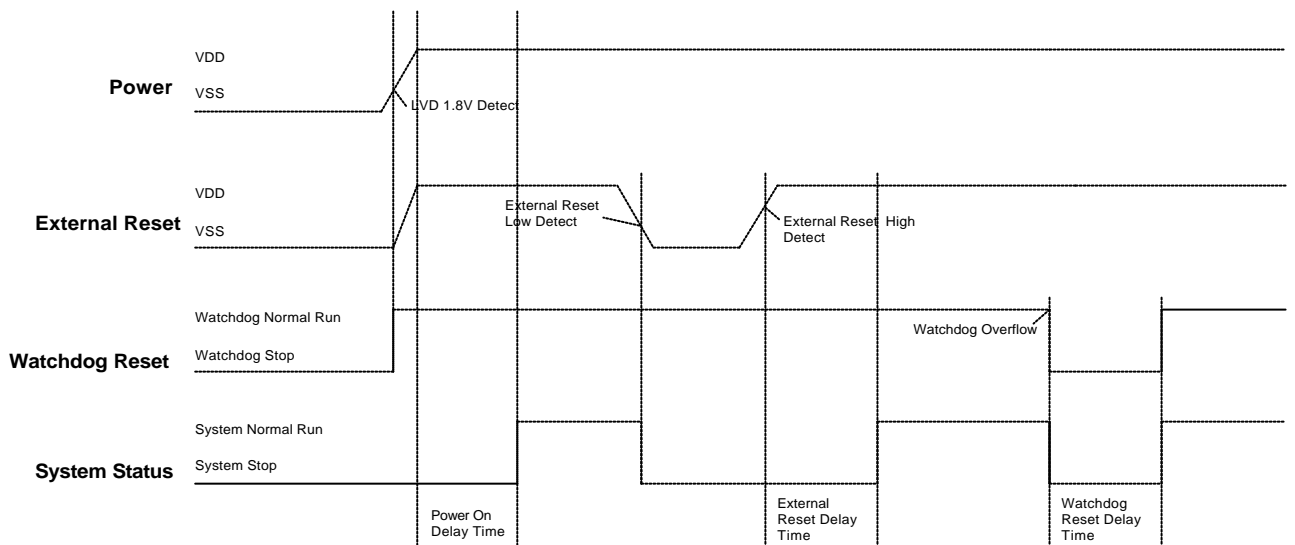
1. 复位

26系列单片机共提供4种复位方式，分别是：

1. 上电复位 (POR)
2. 外部复位 (RESET)
3. 低电压复位 (LVD)
4. 看门狗复位 (WDTR)

1.1 概述

MCU的复位方式有上电复位，外部复位和看门狗复位。当任何一种复位发生，系统复位并初始化系统寄存器。复位后，系统进入普通模式并操作。复位时序图如下：



上电复位时序图

1.2 上电复位

上电或者电源从 VSS 上升到 VDD 时，系统就会复位，跳到普通模式。从上电到系统正常运行之间的这段时间叫做“上电延迟时间”，系统会在系统延迟时间后正常运行。

VDD	晶振	上电延迟时间
3V	4MHz	大约 131ms
5V	4MHz	大约 66ms
3V	32768Hz	大约 239ms
5V	32768Hz	大约 174ms

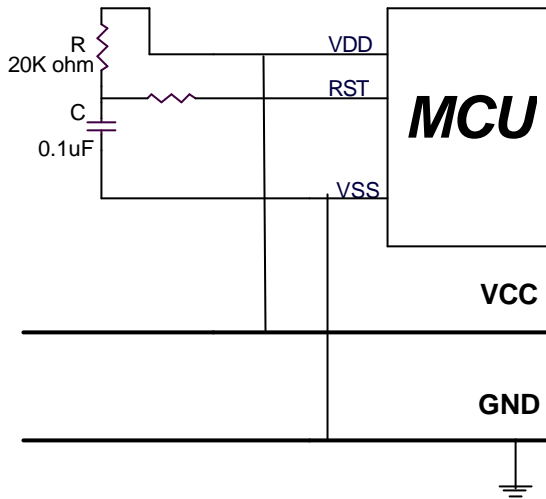
1.3 外部复位

外部复位引脚为施密特触发结构，当它处于复位模式时，则使能复位功能。外部复位为低电平有效，当复位引脚侦测到低电压（低于 $0.3V_{DD}$ ）时，系统开始复位，直到侦测到的电压达到高电平（高于 $0.7V_{DD}$ ）。用户可以使用外部复位电路控制系统。

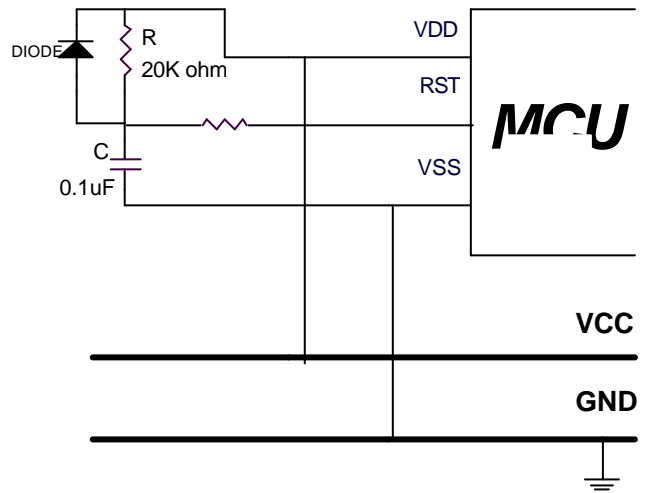
VDD	晶振	外部复位延迟时间
3V	4MHz	大约 131ms
5V	4MHz	大约 66ms
3V	32768Hz	大约 193ms
5V	32768Hz	大约 127ms

1.3.1 外部复位电路

外部复位电路就是一个简单的 RC 回路，如下所示：



简单 RC 复位电路



添加二极管后的节电复位电路

➤ 注：使用 R2 可以避免外部干扰对 MCU 复位的影响，建议 R2 的值为 100 ~ 2000。

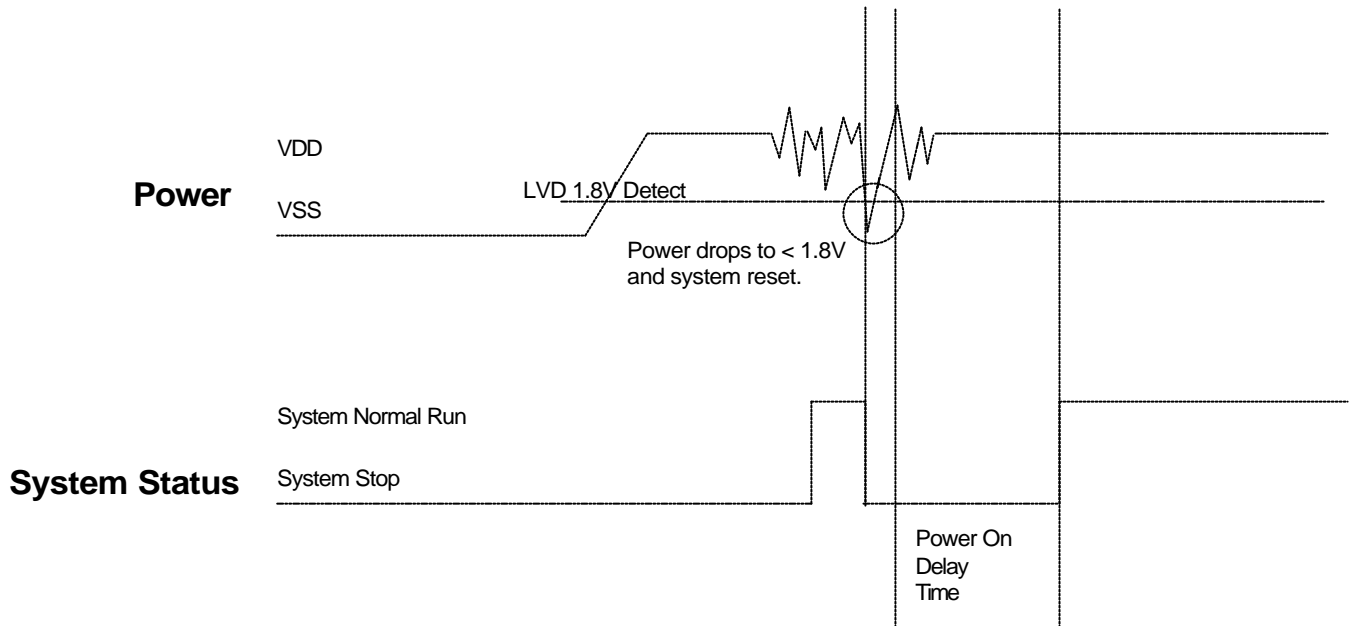
1.4 看门狗复位

看门狗复位是对系统的一种保护。在正常情况下，系统可以正常运行，也可以由程序对看门狗清零；但在错误状态下，系统位于未知状态，且在看门狗定时器溢出前不能由程序对看门狗清零。看门狗定时器溢出后，系统复位，系统重启并进入普通模式。

VDD	晶振	看门狗复位延迟时间
3V	4MHz	大约 145ms
5V	4MHz	大约 73ms
3V	32768Hz	大约 207ms
5V	32768Hz	大约 127ms

1.5 低电压侦测 (LVD)

LVD 为低电压侦测 当侦测到 VDD 低于设定值时 系统就会复位。低电压侦测电压为 1.8V 如果 VDD 低于 1.8V , 系统就会复位。如果 VDD 返回到 1.8V 以上, 系统开始执行上电程序, 在上电延迟时间结束后, 系统开始正常运行, 这叫做节电复位。在高干扰环境下, 上电会有些不稳定, LVD 避开噪音以免影响系统。当 VDD 保持 1.8V 以上的电压时, 系统就会保持稳定的工作状态; 当 VDD 低于 1.8V 时, LVD 使系统复位并等待 VDD 返回到正常电压。LVD 复位延迟就是上电复位延迟时间。



2 系统时钟

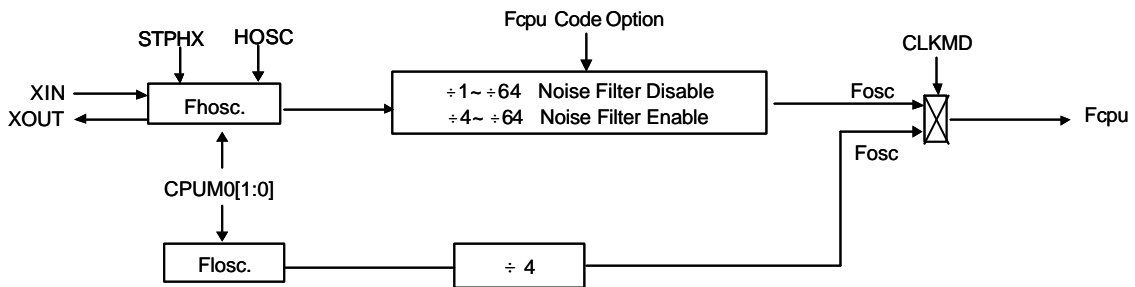
2.1 概述

26 系列是具有高速时钟和低速时钟的双时钟微控制器。高速时钟是由外部振荡电路或内置 16MHz 的高速 RC 振荡电路 (IHRC 16MHz) 提供, 而低速时钟则是由内置低速 RC 振荡电路 (ILRC 16KHz @3V, 32KHz @5V) 提供。高低速时钟均可作为系统时钟 (Fosc)。

普通模式 (高速时钟): $F_{cpu} = F_{osc}/N$, $N = 1\sim64$, N 由 Fcpu 编译选项选择。

低速模式 (低速时钟): $F_{cpu} = F_{osc}/4$

2.2 时钟框图



- HOSC : High_Clk 编译选项
- Fhosc : 外部高速时钟/内部高速 RC 时钟
- Fosc : 内部低速 RC 时钟

2.3 OSCM 寄存器

OSCM 寄存器是一个振荡器控制寄存器, 它控制着振荡器的状态和系统模式。

0CAH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OSCM	0	0	0	CPUM1	CPUM0	CLKMD	STPHX	0
读/写	-	-	-	R/W	R/W	R/W	R/W	-
复位后	-	-	-	0	0	0	0	-

- Bit1 **STPHX** : 外部高速振荡器控制位
 0 = 运行
 1 = 停止, 内部低速振荡器仍处于运行状态。
- Bit2 **CLKMD** : 系统高/低速模式控制位
 0 = 普通模式 (双重时钟), 系统时钟为高速时钟。
 1 = 低速模式, 系统时钟为内部低速时钟
- Bit[4:3] **CPUM1, CPUM0** : CPU 运行模式控制位。
 00 = 普通模式
 01 = 睡眠 (省电) 模式
 10 = 绿色模式
 11 = 保留

☞ 例 : 停止高速振荡器
 BOBSET FSTPHX ; 仅停止高速振荡器

☞ 例：在省电模式下，高速振荡器和内部低速振荡器都停止。
BOBSET FCPUM0 ；停止外部高速和内部低速振荡器进入省电（睡眠）模式

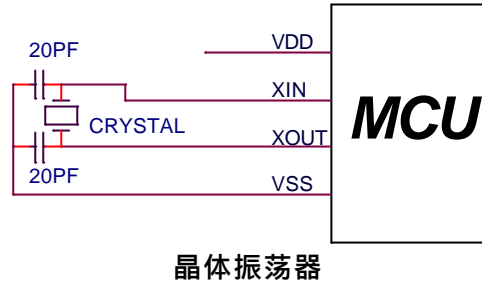
2.4 系统高速时钟

2.4.1 外部高速时钟

外部高速时钟有三种（晶体，RC和外部时钟信号）。高速振荡电路由编译选项的HIGH_CLK选项控制。

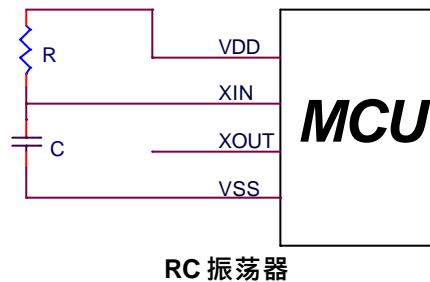
2.4.1.1 晶体振荡器

晶体振荡由 XIN，XOUT 引脚驱动。当使用高/一般/低的频率时，驱动电流是不一样的。HIGH_CLK 的编译选项支持不同的频率。 高速：12M-option.(如：12M)； 一般：4M-option (如：4M)；低速：32k_option(如：32.768K)



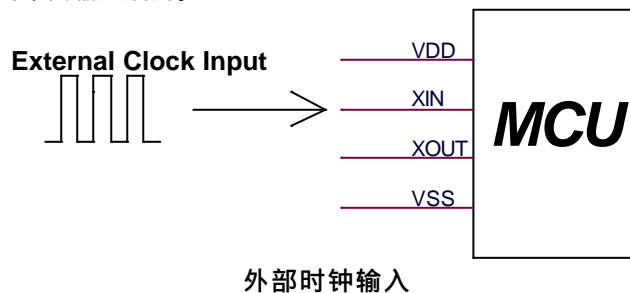
2.4.1.2 RC 振荡器

可以由 HIGH_CLK 选项中的 RC 选项来选择 RC 振荡器。RC 振荡器的频率可以达到 16M。建议改变“R”的值来改变频率。“C”的大小一般为 50P~100P。P1.4 是 XOUT，为无上拉电阻的单向输入引脚。



2.4.1.3 外部时钟输入

外部时钟输入做为系统时钟信号，由 HIGH_CLK 选项中的 RC 选项来选择。外部时钟输入信号由 XIN 引脚输入。P1.4 为 XOUT，是无上拉电阻的单向输入引脚。



* 注：外部振荡器电路的地线必须和微控制器的 Vss 引脚直接相连接。

4.4.2 内部高速 RC

芯片内置 16M RC 时钟，由编译选项中的 IHRC_16M 和 IHRC_RTC 控制。在 IHRC_16M 模式下，系统时钟由内部 16M RC 振荡器提供，XIN/XOUT 引脚和 P1.2/P1.3 共享。在 IHRC_RTC 模式下，系统时钟由内部 16M RC 时钟提供，XIN/XOUT 引脚和外部 32768 晶振连在一起，组成实时时钟（RTC）。

RTC 由寄存器 OPTION 控制，且 RTC 定时器为 T0。详情请参阅“定时器 T0”。

编译选项	XIN 引脚	XOUT 引脚	系统时钟源	RTC 功能
IHRC_16M	P1.3	P1.2	IHRC	禁止

IHRC_RTC	32768Hz 晶振	32768Hz 晶振	IHRC	使能
----------	------------	------------	------	----

2.5 系统低速时钟

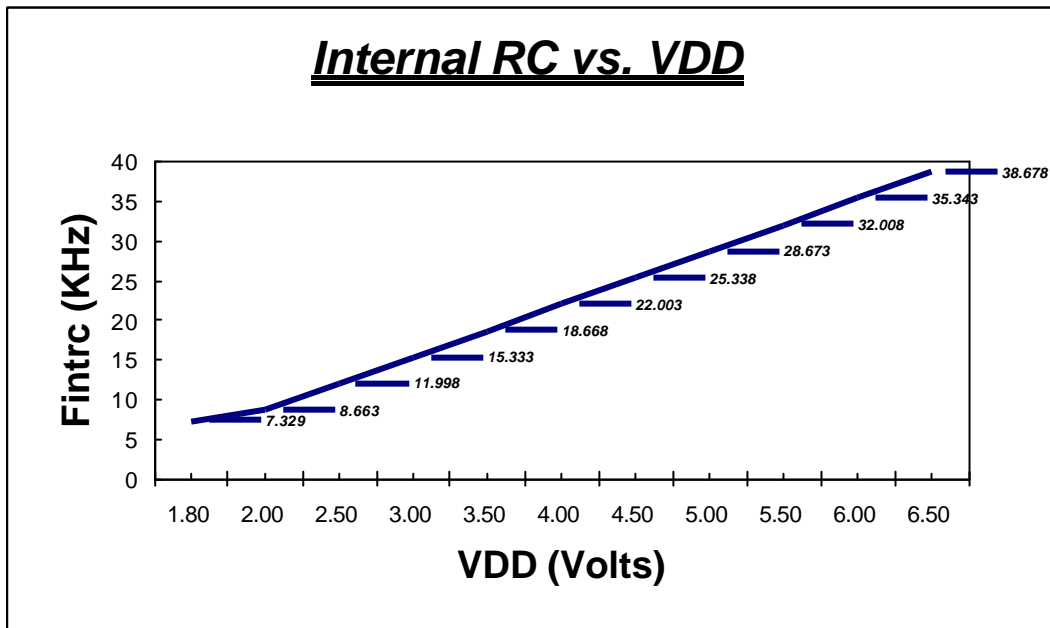
26 系列内置内部低速振荡器，其时钟源由 RC 振荡电路提供。

☞ 系统低速时钟 (Fcpu) = 内部低速时钟 (16KHz @3V , 32KHz @5V) / 4

☞ 例：停止内部低速时钟。
`BOBSET FCPUM0` ; 停止外部高速和内部低速振荡器进入省电（睡眠）模式

* 注：内部低速时钟不能被单独停止，由 OSCM 寄存器的 CPUM0 位控制。

低速振荡器采用 RC 振荡电路，频率受系统电压和温度的影响。通常情况下 RC 振荡器的频率约为 16KHz (3V) , 32KHz (5V) 。RC 频率和电压之间的关系如下图所示：



☞ 例：由 Fcpu 测试内部 RC 频率。内部 RC 频率等于 4 倍的 Fcpu。我们可以从 Fcpu 得到内部 RC 的频率。
`BOBSET P1M.0` ; 设置 P1.0 为输出模式

@@:
`BOBSET FCLKMD` ; 进入内部低速模式
`BOBSET P1.0` ; 在低速模式下输出频率信号
`BOBCLR P1.0` ; 通过测量指令周期来测试频率
`JMP @B`

2.5.1 ? ? ? ? ? ? ? ?

在 RC 模式，用户可以由指令周期 (Fcpu) 来测试 Fosc 的周期。

☞ 例：由测量指令周期来测量外部振荡器的周期
`BOBSET P1M.0` ; 设置 P1.0 为输出模式

@@:
`BOBSET P1.0` ; 输出方波信号
`BOBCLR P1.0` ; 通过测量指令周期来测试频率
`JMP @B`

* 注：不能直接由 XIN 测试 RC 频率；探针会影响 RC 的频率值。

3 系统操作模式

3.1 概述

SN8P2501A 具有低功耗的特性，能在 4 种不同的操作模式下相互转换。

- 普通模式
- 低速模式
- 省电模式（睡眠模式）
- 绿色模式

3.1.1 普通模式

普通模式下，系统时钟源是高速时钟（外部或内部 16MHz RC 振荡器）。上电复位、看门狗复位、LVD 复位或由省电模式唤醒后，系统就在普通模式下运行。系统可以在普通模式下转入省电模式，低速模式和绿色模式。

3.1.2 低速模式

低速模式时，系统时钟源为内部低速时钟。设置 CLKMD=1，系统就进入内部低速模式。低速模式下的运行与普通模式一样，仅是时钟频率有所降低。系统可在低速模式下转入普通模式及低功耗的绿色模式和省电模式。

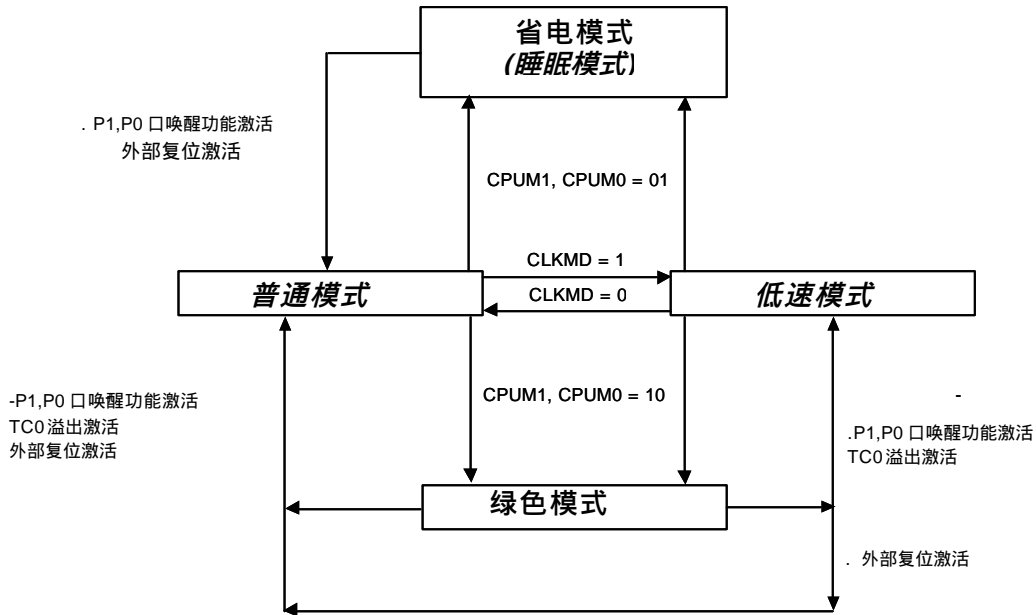
3.1.3 绿色模式

绿色模式提供一个定时唤醒功能。用户可通过设置定时器 T0 来确定系统的唤醒时间。系统可以从普通模式和低速模式进入绿色模式，并且可以由 T0 定时器或由 P0，P1 的电平变换来唤醒。

3.1.4 省电（睡眠）模式

省电模式也称睡眠模式，系统进入睡眠状态时，将停止工作，功耗低至近似于零。省电模式常用于电池供电等节电系统。设 CPUM0=1，系统进入省电模式，外部高速和内部低速振荡器均停止工作，P0、P1 的触发信号可将系统唤醒。

3.2 系统模式控制



操作模式说明

模式	普通模式	低速模式	绿色模式	省电模式 (睡眠模式)	备注
EHOSC	运行	由 STPHX 位决定	由 STPHX 位决定	停止	
IHRC	运行	由 STPHX 位决定	由 STPHX 位决定	停止	
ILRC	运行	运行	运行	停止	
EHOSC (带 RTC)	运行	由 STPHX 位决定	运行	停止	
IHRC (带 RTC)	运行	由 STPHX 位决定	停止	停止	
ILRC (带 RTC)	运行	运行	停止	停止	
CPU	执行	执行	停止	停止	
T0	*有效	*有效	*有效	无效	*程序激活
TC0	*有效	*有效	无效	无效	*程序激活
看门狗定时器	编译选项	编译选项	编译选项	编译选项	Enable
					Disable
					Always_on
内部中断	全部有效	全部有效	T0	全部无效	
外部中断	全部有效	全部有效	全部有效	全部无效	
系统唤醒	-	-	P0, P1, T0 复位	P0, P1 复位	

3.2.1 系统模式转换

☞ 例：普通/低速模式进入省电（睡眠）模式
 B0BSET FCPUM0 ; 设置 CPUM0 = 1.

* 注：系统在睡眠模式中，只有具有唤醒功能的引脚和复位引脚能够将系统唤醒。

☞ 例：普通模式转换为低速模式
 B0BSET FCLKMD ; 设置 CLKMD = 1, 进入低速模式
 B0BSET FSTPHX ; 停止高速振荡器以省电

☞ 例：低速模式转换为普通模式(外部高速振荡器仍然运行)
 B0BCLR FCLKMD ;设置 CLKMD = 0

☞ 例：低速模式转换为普通模式（外部高速振荡器停止）
 如果外部高速时钟停止时程序欲重新回到普通模式，就必须延迟至 10mS 以等待外部时钟稳定下来。
 B0BCLR FSTPHX ; 启动外部高速振荡器

@@: B0MOV Z, #27 ; 若 VDD = 5V, 则内部 RC 的频率为 32KHz (典型值)
 DECMS Z ; 高速振荡器稳定时间 0.125ms X 81 = 10.125ms
 JMP @B ;
 B0BCLR FCLKMD ;进入普通模式

☞ 例：普通/低速模式进入绿色模式
 B0BSET FCPUM1 ; 设置 CPUM1 = 1.

* 注：在没有 T0 唤醒功能的绿色模式下，只有具有唤醒功能的引脚才能将系统唤醒返回上一个操作模式。

☞ 例：普通模式/低速模式进入绿色模式（使能 T0 的唤醒功能）
 ; 设置定时器 T0 的唤醒功能。
 B0BCLR FT0IEN ; 禁止 T0 中断
 B0BCLR FT0ENB ; 禁止 T0 计数
 MOV A, #20H ;
 B0MOV T0M, A ; 设置 T0 时钟源 = Fcpu / 64
 MOV A, #74H ;
 B0MOV T0C, A ; 设置 T0C 初始值= 74H (设置 T0 定时间隔 = 10 ms)
 B0BCLR FT0IEN ; 禁止 T0 中断
 B0BCLR FT0IRQ ; 清 T0 中断请求标志
 B0BSET FT0ENB ; 使能 T0 计数
 ; 进入绿色模式
 B0BCLR FCPUM0 ;设置 CPUMx = 10
 B0BSET FCPUM1

* 注：绿色模式下，打开 T0 的唤醒功能，具有唤醒功能的引脚和 T0 都能将系统唤醒返回到上一个操作模式。T0 的唤醒周期由程序控制。

☞ 例：普通/低速模式进入绿色模式，使能 T0 带 RTC 功能的唤醒功能。
 ; 设置 T0 每 0.5 秒唤醒一次（RTC）。
 B0BCLR FRTC1 ; 设置 RTC 的时间为 0.5 秒。
 B0BCLR FRTC0
 B0BSET FT0ENB ; 使能 T0 计数
 B0BSET FT0TB ; 使能 RTC 功能
 ;进入绿色模式
 B0BCLR FCPUM0 ;设置 CPUMx = 10
 B0BSET FCPUM1

* 注：在 T0 具有 RTC 唤醒功能的绿色模式下，具有唤醒功能的引脚和 T0 都能将系统唤醒返回到上一个操作模式。T0 的唤醒周期由 OPTION 寄存器控制。

088H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPTION	-	-	-	-	RTC1	RTC0	-	-
读/写	-	-	-	-	R/W	R/W	-	-
复位后	-	-	-	-	0	0	-	-

RTC1, RTC0：实时时钟周期时间控制位
 00 = 0.5s, 01 = 1s, 10 = 2s, 11 = 4s.

3.3 唤醒时间

3.3.1 概述

外部高速振荡器从停止到运行需要一段时间的延迟，这段延迟时间对振荡器的稳定工作是必需的。在有些应用中，高速时钟（外部或内部 16MHz RC 振荡器）可能需要经常的开停。外部高速振荡器重新启动需要的这一延迟时间称为唤醒时间。

下面两种情况需要唤醒时间：一种是从省电模式转换到正常模式，另一种是从低速模式转换到普通模式。对前一种情况，26 系列 提供了 2048 个振荡时钟作为唤醒时间，后一种情况需要用户自行计算唤醒时间。

在省电（睡眠）模式下，P0 和 P1 都具有唤醒功能，可以将系统唤醒。P0 的永远具有唤醒功能，P1 的唤醒功能则是由 P1W 寄存器控制的。改变其电平即能产生唤醒信号。

3.3.2 唤醒时间

当系统处于省电（睡眠）模式时，外部高速振荡器停止运行。从睡眠模式唤醒时，SN8P2501A 提供 4096 个外部高速振荡周期作为唤醒时间，以使振荡电路达到稳定状态。唤醒时间结束后，系统进入正常模式，唤醒时间的计算方法如下：

$$\text{唤醒时间} = 1/\text{Fosc} \times 4096 \text{ (sec)} + \text{X'tal 稳定时间}$$

X'tal 稳定时间决定于 X'tal 的类型，一般的，约为 2~4ms（4MHz 晶体振荡器）

☞ 例：省电（睡眠）模式下，系统由 P0、P1 的触发信号唤醒。唤醒时间后，系统进入普通模式，P0、P1 唤醒功能的唤醒时间如下：

$$\text{唤醒时间} = 1/\text{Fosc} \times 4.96 = 1.14\text{ms} \text{ (Fosc} = 3.58\text{MHz)}$$

$$\text{总的唤醒时间} = 1.14\text{ms} + \text{X'tal 稳定时间}$$

3.3.3 P1W 唤醒控制寄存器

省电（睡眠）模式下，具有唤醒功能的 P0 和 P1 都能将系统唤醒，P0 永远具有唤醒功能，而 P1 的唤醒功能受寄存器 P1W 控制。改变其电平就可产生唤醒信号。

0C0H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1W	-	-	-	-	P13W	P12W	P11W	P10W
读/写	-	-	-	-	W	W	W	W
复位后	-	-	-	-	0	0	0	0

Bit[3:0] P10W~P13W：P1 唤醒功能控制位

0=禁止 P1n 唤醒功能

1=使能 P1n 唤醒功能