

## CMT2189B 用户指南

### 概要

CMT2189B 是一款低功耗，高性能，Flash 型的 OOK 无线发射单片机。它能够覆盖 240 - 960MHz 的无线通信频段，同时内嵌一个 RISC 的 Flash 型 MCU。它们都属于 CMOSTEK NextGenRF™ 系列的产品。该产品系列包含了发射机，接收机，收发一体机和 SoC 等短距离无线通信芯片。

本文档涵盖的产品型号如下表所示。

表 1. 本文档涵盖的产品型号

产品型号	工作频率	调制方式	发射功率	发射电流	配置方式	封装
CMT2189B	240 - 960MHz	OOK	+13dBm	17.5mA	内嵌 MCU	SOP14

注意：发射功率和发射电流测试条件为 433.92MHz，CW 模式（即一直处于发射载波模式）；按 Duty 50% 的发射模式，发射电流约为 8.5mA

# 目录

<b>1</b>	<b>芯片架构介绍 .....</b>	<b>6</b>
1.1	总体工作原理 .....	6
1.2	IO 管脚说明 .....	6
<b>2</b>	<b>RF 配置和控制机制 .....</b>	<b>9</b>
2.1	工作模式 .....	9
2.2	简易工作模式 .....	9
2.3	高级配置模式 .....	9
2.4	SPI 接口时序 .....	10
2.5	RF 配置参数 .....	11
2.6	配置寄存器 .....	15
2.7	数据包相关寄存器 .....	15
2.7.1	发射速率 .....	16
2.7.2	硬件数据包格式 .....	16
2.7.3	Preamble .....	17
2.7.4	Head/Sync .....	18
2.7.5	Addr/ID .....	19
2.7.6	Key Value .....	20
2.7.7	LBD Status 配置 .....	21
2.7.8	Stop Bit 配置 .....	22
2.7.9	Pause/Interval 配置 .....	23
2.7.10	Tcycle 配置 .....	23
2.8	状态及功能寄存器 .....	24
2.8.1	软复位 .....	25
2.8.2	工作状态以及状态切换 .....	25
2.8.3	工作状态查询 .....	26
2.9	硬件数据包发射模式 .....	26
2.9.1	上电初始化 .....	26
2.9.2	发射流程 .....	26
2.10	直通发射模式（高级配置模式） .....	27
2.10.1	上电初始化 .....	27
2.10.2	发射流程 .....	27
2.10.3	相关寄存器 .....	28
<b>3</b>	<b>程序存储器 .....</b>	<b>29</b>
<b>4</b>	<b>功能寄存器（SFR） .....</b>	<b>30</b>

4.1	地址映射 .....	30
4.1.1	Bank0 SFR .....	30
4.1.2	BANK1 SFR .....	31
4.1.3	TMR0 (Addr:0x01) .....	32
4.1.4	STATUS (Addr:0x03) .....	32
4.1.5	PORTA (Addr:0x05) .....	33
4.1.6	PORTC (Addr:0x07) .....	33
4.1.7	INTCON (Addr:0x0B) .....	34
4.1.8	PIR1 (Addr:0x0C) .....	35
4.1.9	TMR2 (Addr:0x11) .....	35
4.1.10	T2CON (Addr:0x12) .....	36
4.1.11	WDTCON (Addr:0x18) .....	36
4.1.12	CMCON0 (Addr:0x19) .....	37
4.1.13	PR0 (Addr:0x1A) .....	38
4.1.14	MSCKCON (Addr:0x1B) .....	39
4.1.15	SOSCPR (Addr:0x1C/0x1D) .....	39
4.1.16	OPTION (Addr:0x81) .....	40
4.1.17	TRISA (Addr:0x85) .....	40
4.1.18	TRISC (Addr:0x87) .....	41
4.1.19	PIE1 (Addr:0x8C) .....	41
4.1.20	PCON (Addr:0x8E) .....	42
4.1.21	OSCCON (Addr:0x8F) .....	42
4.1.22	PR2 (Addr:0x92) .....	43
4.1.23	WPUA (Addr:0x95) .....	44
4.1.24	IOCA (Addr:0x96) .....	44
4.1.25	VRCON (Addr:0x99) .....	44
4.1.26	EEDAT (Addr:0x9A) .....	45
4.1.27	EEADR (Addr:0x9B) .....	45
4.1.28	EECON1 (Addr:0x9C) .....	45
4.1.29	EECON2 (Addr:0x9D) .....	46
4.1.30	配置寄存器 UCFGx .....	46
4.1.31	PCL 和 PCLATH .....	48
4.1.32	INDF 和 FSR 寄存器 .....	48
5	系统时钟源 .....	49
5.1	时钟源模式 .....	49
5.2	外部时钟模式 .....	50

5.2.1	EC 模式 .....	50
5.2.2	LP 和 XT 模式 .....	50
5.3	内部时钟模式 .....	50
5.3.1	频率选择位 (IRCF) .....	50
5.3.2	HFINTOSC 和 LFINTOSC 时钟切换时序 .....	51
5.4	时钟切换 .....	52
5.4.1	系统时钟选择 (SCS) 位 .....	52
5.4.2	振荡器起振超时状态 (OSTS) 位 .....	52
5.5	双速时钟启动模式 .....	52
5.5.1	双速启动模式配置 .....	52
5.5.2	双速启动顺序 .....	53
5.6	故障保护时钟监控器 .....	53
5.6.1	故障保护检测 .....	53
5.6.2	故障保护操作 .....	54
5.6.3	故障保护条件清除 .....	54
5.6.4	复位或从休眠中唤醒 .....	54
6	复位时序 .....	55
6.1	POR 上电复位 .....	56
6.2	外部复位 MCLR .....	56
6.3	PWRT (上电计时器) .....	56
6.4	BOR (LVR) 低电压复位 .....	56
6.5	错误指令复位 .....	57
6.6	超时动作 .....	57
7	BOOT .....	60
8	看门狗定时器 .....	61
9	定时器 0 .....	62
9.1	Timer0 简介 .....	62
9.2	Timer0 定时器模式 .....	62
9.3	Timer0 计数器模式 .....	62
9.3.1	软件可配置预分频电路 .....	63
9.3.2	Timer0 中断 .....	63
9.3.3	用外部时钟驱动定时器 0 .....	64
10	定时器 2 .....	65
11	比较器 .....	67
12	数据 EEPROM .....	68
13	时钟测量 .....	69

<b>14</b>	<b>中断模式</b>	<b>70</b>
14.1	INT 中断	70
14.2	PORTA 电平变化中断	71
14.3	中断响应	71
14.4	中断过程中的现场保存	73
<b>15</b>	<b>MCU 睡眠省电模式</b>	<b>74</b>
15.1	唤醒模式	74
15.2	看门狗唤醒	74
<b>16</b>	<b>I/O 端口</b>	<b>75</b>
16.1	PORTA 端口和 TRISA 寄存器	75
16.2	端口的其它功能	75
16.2.1	弱上拉	75
16.2.2	状态变化中断	75
16.3	端口描述	76
16.3.1	PORTA<2:0>	76
16.3.2	PORTA3/PA3	77
16.3.3	PORTA4/PA4	78
16.3.4	PORTA5/PA5	79
16.3.5	PORTA7/PA7	79
16.3.6	PORTC<7:0>	80
<b>17</b>	<b>指令集列表</b>	<b>81</b>
<b>18</b>	<b>文档变更记录</b>	<b>83</b>
<b>19</b>	<b>联系方式</b>	<b>84</b>

# 1 芯片架构介绍

## 1.1 总体工作原理

CMT2189B 是数模一体化发射单片机，它们采用晶体振荡器提供 PLL 的参考频率和数字时钟，支持数据率从 1Kbps 到 40Kbps 的 OOK 调制发射，并支持基于 MCU 程序进行状态控制，以便实现各种低功耗发射应用场合。

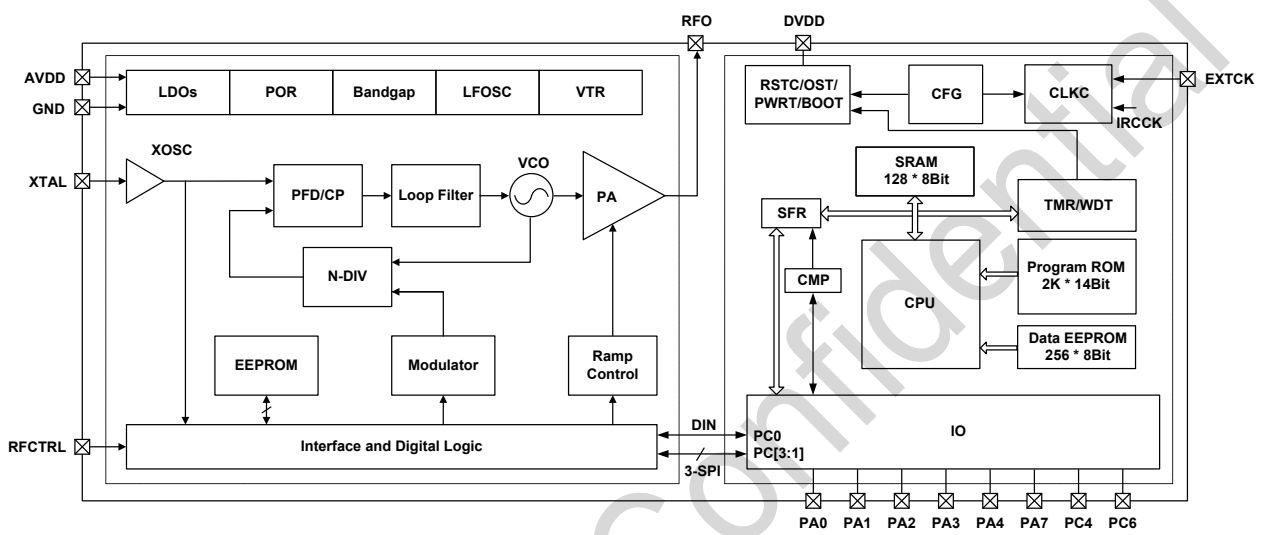


图 1-1. CMT2189B 系统结构

芯片采用 PLL+PA 结构实现 1G 以下频率的无线发射功能。支持数据包 FIFO 模式和直通模式(数据包 FIFO 模式内嵌编码器，可选择合适的编码格式)，将处理后的数据送到调制器（对于直通模式，不经过编码器处理，直接去毛刺后送到调制器），调制器控制 PLL 和 PA，对数据进行 OOK 调制并发射出去。

芯片 MCU 通过 3 线 SPI 接口对 RF 部分进行控制，可实现各种状态切换，模式选择以及低功耗方面的控制。

## 1.2 IO 管脚说明

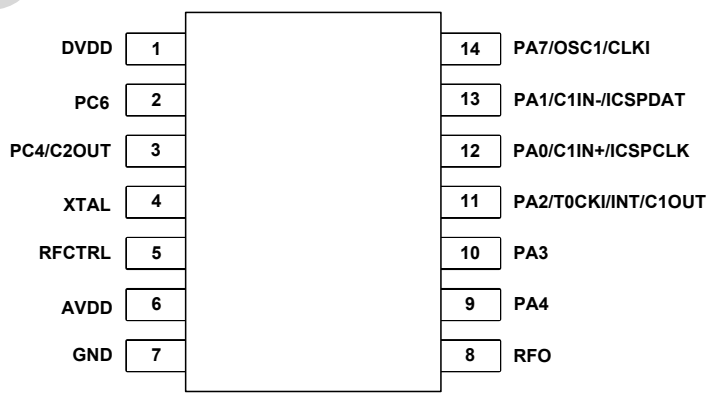


图 1-2. CMT2189B 管脚俯视图

表 1-2. CMT2189B SOP14 封装管脚说明

管脚号	名称	类型	I/O	功能说明	
1	DVDD	数字	I	芯片供电电源正	
2	PC6	数字	IO	通用 IO	
3	PC4/C2OUT	数字	IO	PC4	通用 IO
				C2OUT	比较器 2 输出
4	XTAL	模拟	I	RF 部分晶体振荡器输入	
5	RFCTRL	数字	I	RF 部分 SPI 接口使能控制, 低有效, 内部上拉	
6	AVDD	模拟	I	RF 部分供电电源正	
7	GND	数字	--	芯片供电电源地	
8	RF0	模拟	O	RF 部分 PA 输出	
9	PA4	数字	IO	通用 IO, 支持 IOC, 可配置上拉	
10	PA3	数字	IO	通用 IO, 支持 IOC, 可配置上拉	
11	PA2/T0CKI/INT/C1OUT	数字	IO	PA2	通用 IO, 支持 IOC, 可配置上拉
				T0CKI	Timer0 时钟源输入 (Max=4MHz)
				INT	外部中断输入
				C1OUT	比较器 1 输出
12	PA0/C1IN+/ICSPCLK	数字	IO	PA0	通用 IO, 支持 IOC, 可配置上拉
				C1IN+	比较器 1 输入+
				ICSPCLK	Debug/烧录模式串口 Clock 信号
13	PA1/C1IN-/ICSPDAT	数字	IO	PA1	通用 IO, 支持 IOC, 可配置上拉
				C1IN-	比较器 1 输入-
				ICSPDAT	Debug/烧录模式串口 Data 信号
14	PA7/OSC1/CLKI	数字	IO	PA7	通用 IO, 支持 IOC, 可配置上拉
				OSC1	MCU 晶体脚
				CLKI	外部时钟输入脚
内部引脚	PC0/C2IN+/RFDIN	数字	IO	PC0	通用 IO
				C2IN+	比较器 2 输入+
				RFDIN	RF 直通模式数据输入脚
内部引脚	PC1/C2IN-/SDIO	数字	IO	PC1	通用 IO, 芯片内部连接到 RF 部分
				C2IN-	比较器 2 输出-
				SDIO	RF 部分 3 线 SPI 串行总线数据 SDIO, SDIO 本身不带上拉或下拉电阻, 是双向端口
内部引脚	PC2/SCLK	数字	IO	PC2	通用 IO, 芯片内部连接到 RF 部分
				SCLK	RF 部分 3 线 SPI 串行总线时钟 SCLK
内部引脚	PC3/CSB	数字	IO	PC3	通用 IO, 芯片内部连接到 RF 部分
				CSB	RF 部分 3 线 SPI 串行总线片选 CSB, 内部带上拉电阻

注意:

1. MCU 内集成 2 个比较器, 但由于封装引出管脚, 以及同时部分管脚复用到 RF 部分, 所以内部的比较器不能使用。但 MCU 初始化需要对比较器进行必要的初始化后设置, 以免影响其它功能的工作。
2. MCU 的时钟源系统, 支持内部振荡、外部振荡。其中外部振荡支持双端晶体和单端时钟源输入。但由于封装引出管脚的限制, 没有把 PA6 引出, 所以实际不支持双端晶体的外部振荡模式, 仅支持单端时钟源的外部振荡模式。
3. PC<3:0>为芯片内部控制引脚, 没有引出到芯片管脚, 但内部是作为控制 RF 的总线使用。

CMOSTEK Confidential



## 2 RF 配置和控制机制

### 2.1 工作模式

CMT2189B 内置的 OOK 发射功能有两种工作模式：

- 简易工作模式：上电默认进入的模式，即免配置模式，此模式下仅支持直通发射模式；
- 高级配置模式：通过 SPI 总线配置寄存器和控制工作状态的 mode，此模式下支持 FIFO 和直通模式；

注意：两种工作模式下的直通发射输入源不同，下文有详细介绍。

### 2.2 简易工作模式

在简易工作模式下，外围只需要搭配一个晶体振荡器，芯片上电后通过内部的 PC1（SDIO）进行控制发射的数据，便可以实现相应的发射频率的发送。该模式下，倍频系数固定为 16.5，即按下面计算公式：

$$F_{XTAL} = \frac{F_{RF}}{16.5}, 314 \text{ MHz} \leq F_{RF} \leq 480 \text{ MHz}$$

其中， $F_{XTAL}$  为晶体频率， $F_{RF}$  为工作目标频率，能实现频率范围是 314~480MHz。

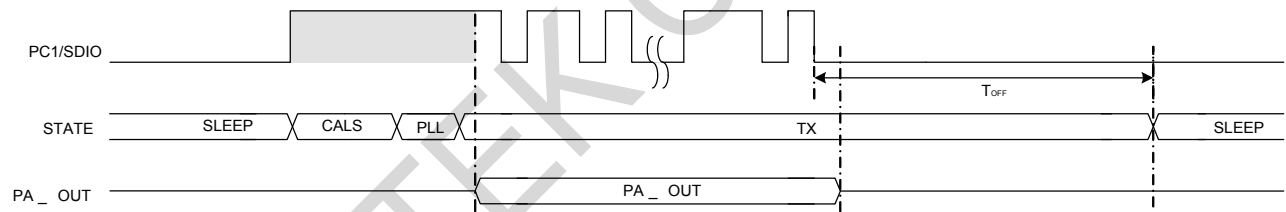


图 2-1. 简易工作模式发射时序图

注意：

1. 该模式下支持 1Kbps~20Kbps 的空中速率范围；
2. 功率固定为+13dBm 最大输出模式，用户需要降低发射功率时，只能通过在 Chock 电感前串接限流电阻；
3. 内部的 SPI 总线不用发任何控制命令，只需要控制 PC<3:0>（PC1 除外）均为高阻输入状态即可；
4. RFCTRL 引脚此模式下悬空即可；
5. 此模式发射数据脚为 PC1（SDIO），低功耗（即常态）为置 0 输出状态；需要发射时，产生上升沿触发即可进入发射状态；发射完毕后，置 0 大于 20ms（ $T_{OFF}$  时间），RF 即自动退出发射模式，进入低功耗状态。
6. 此模式下的低功耗内部管脚处理，PC1/SDIO 需要置 0 输出，PC2/SCLK、PC3/CSB 和 PC4/RFDIN 都设置为高阻输入。

### 2.3 高级配置模式

当用户需要实现更多功能以及更高性能，如：目标工作频率为 868MHz、发射过程芯片自动发包等等情况

下，就需要使用高级配置模式。在此工作模式下：

- 可以选择更多倍频比系数，从而实现 240~960MHz 的频率范围覆盖；
- 可以通过填充 FIFO，发射过程自动发送报文（重复、多次、周期发送），以让释放 MCU 可以做更多其它工作（直通模式下，这个过程占用 MCU 资源）；
- 可以支持更准确的发射速率，精度是取决于晶体，相当准确（直通模式下，速率由 MCU 软件控制，精度取决于软件和内部 RC）；
- 可以支持 RF 内部自带的电压检测功能，实现简单供电电压检测和分析判断处理；
- 可以根据供电电压降低，相应动态调整发射功率，以节省功耗，延长电池使用寿命；

该模式下，芯片内部 MCU 需要通过 3 线 SPI（PC3/CSB、PC2/SCLK、PC1/SDIO）对 RF 部分进行操作寄存器的模式进行控制，可以实现两种发射模式：

**1. 通过 SPI 填充 FIFO 的硬件数据包发送模式（详见本章 2.9 节）**

把需要发射的内容填充到指定的寄存器当中，按配置的速率、编码方式、数据包数量以及包间隔等进行自动发射（数据包格式等相关信息详见本章 2.7 节）。

**2. 通过 PC0/RFDIN 作为发射数据脚的直通模式（详见本章 2.10 节）；**

即 1-Pin 发射模式，数据流由 MCU 产生，并通过指定的管脚进行输出，实现“数据进、天线出”的最基本工作方式。

注意：

1. 上面提及的两种发送模式，前提都是属于高级配置模式下，即通过 SPI 配置和操作芯片的；
2. 在高级配置模式下，需要控制 RFCTRL，即需要通过其它引脚对 RFCTRL 进行控制，所以 RFCTRL 不能悬空；
3. 在高级配置模式下所支持的直通模式，与简易工作模式的直通模式有类似地方，都是通过 1 个数据引脚控制发射的过程，但数据输入源不同。在简易工作模式下，发射输入脚是 PC1/SDIO，而在高级配置模式下的直通模式，由于 PC1/SDIO 为 SPI 的串行数据线，所以更改到 PC0/RFDIN 使用。当然，高级配置模式下的直通模式比简易工作模式下，同样通过配置寄存器方式，能支持更多的频率选择和功率选择。

**2.4 SPI 接口时序**

芯片内部的 RF 功能是通过 3-线的 SPI 口进行控制，与 MCU 部分的 IO 对应关系如下：

表 2-1. SPI 总线与控制端口对应关系

3 线 SPI	MCU 控制端口	功能
CSB	PC3	总线片选使能，低使能，内置上拉
SCLK	PC2	总线时钟线，上升沿触发
SDIO	PC1	总线双向数据

注意：SDIO，双向端口，用于输入和输出数据。地址和数据部分都是从 MSB 开始传送。

当访问 RF 部分的时候，要把 RFCTRL 拉低<sup>[1]</sup>使能 SPI 串口功能，然后片选（PC3/CSB）要拉低，发送一

个 R/W 位，后面跟着 7 位的寄存器地址。在片选（PC3/CSB）拉低之后，必须等待至少半个时钟（PC2/SCLK）周期，才能开始发送 R/W 位。在发送出最后一个时钟（PC2/SCLK）的下降沿之后，必须等待至少半个时钟（PC2/SCLK）周期，再把片选（PC3/CSB）拉高。

在下图 SPI 读操作时，特别注意 PC1/SDIO，因为 SDIO 是双向端口，PC2/SCLK 第 8 个时钟下降沿时候（图中中间虚线），SDIO 会从输入状态切换为输出状态，而 PC2 需要在此前（即第 8 个时钟下降沿之前）从输出状态切换为输入状态。

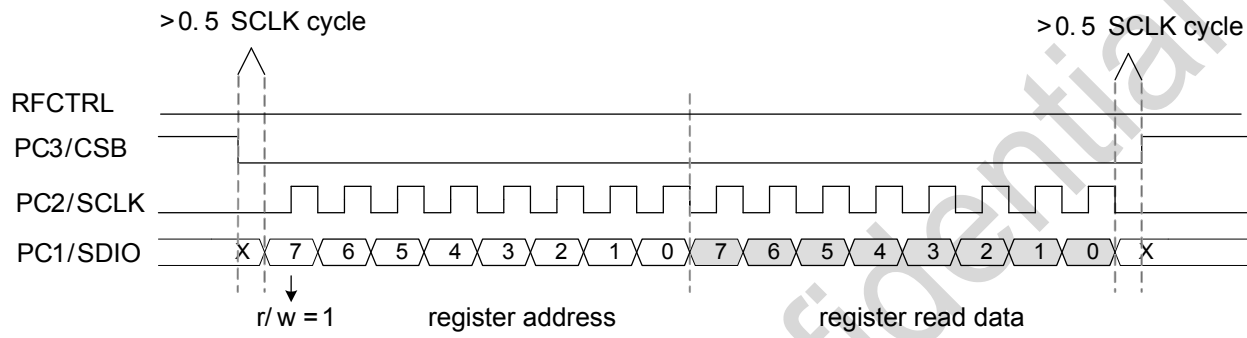


图 2-2. SPI 读寄存器时序

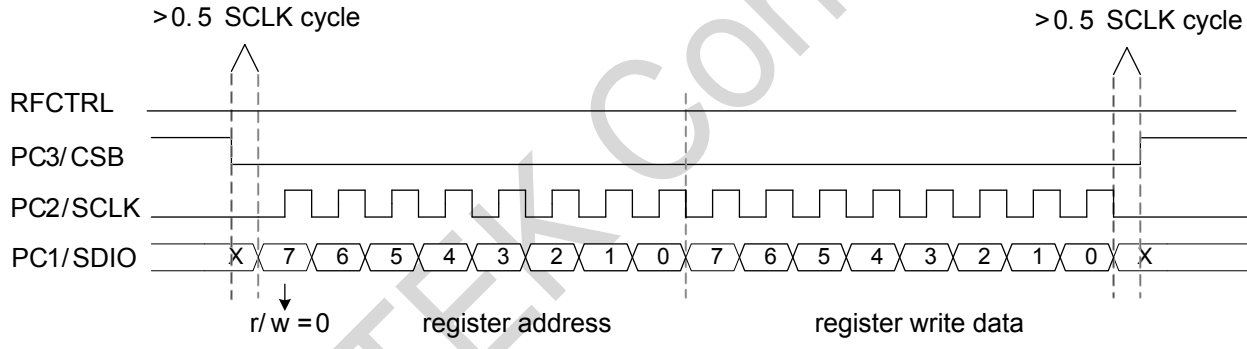


图 2-3. SPI 写寄存器时序

注意：RFCTRL 为输入端口，需要通过外部控制；建议用户通过 CMT2189B 中任意一个功能端口对此进行控制，拉低为使能 SPI 接口功能。整个工作过程，保持 RFCTRL 为低都可以，但在需要睡眠低功耗模式时，需要把 RFCTRL 拉高，因为 RFCTRL 内部自带上拉，拉低会产生功耗。

## 2.5 RF 配置参数

CMT2189B 工作在高级配置模式，可以实现更宽广的工作频率、数据包格式等功能，实现这些功能需要相应的配置参数，它们可以通过 RFPDK 软件导出。具体做法是打开 RFPDK 软件，选择 CMT2157B 型号（与 CMT2189B 内置 RF 同等规格和性能），如下图所示。

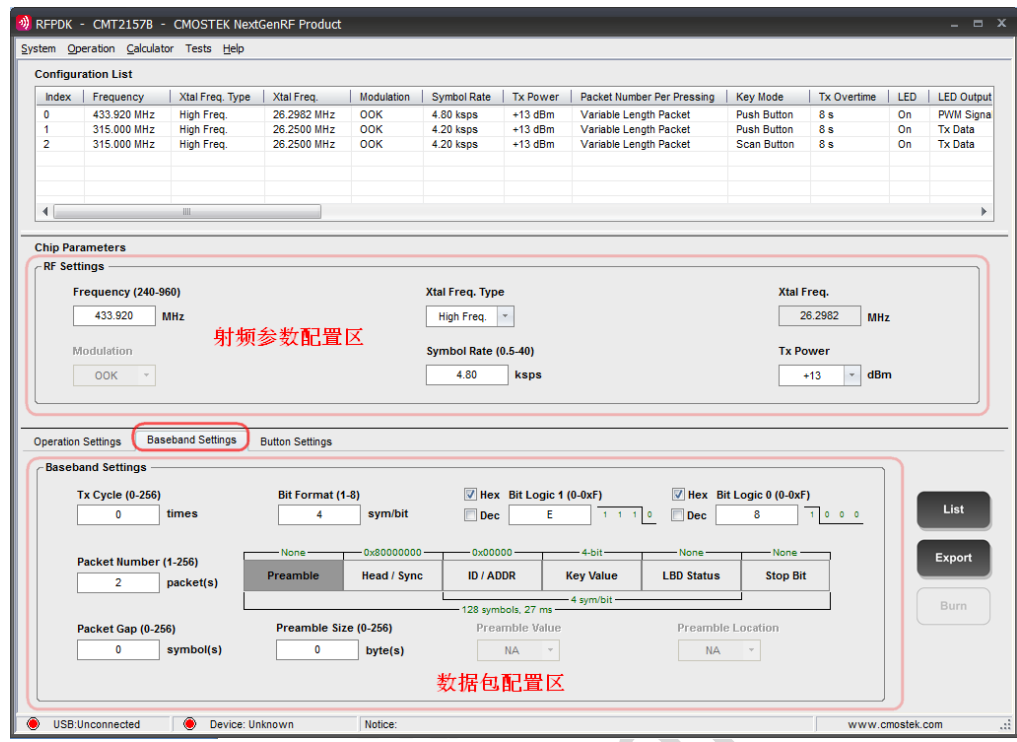


图 2-4. RFPDK CMT2157B 界面

在界面上，主要设置两个区域：**射频参数配置区**和**数据包配置区**，用户根据下文介绍相关的寄存器，按需求进行配置，然后点击 **Export**，生成一个 **exp** 文件，打开如下：

```
-----
;
; CMT2157B Configuration File
; Generated by CMOSTEK RFPDK 1.46
; 2017.11.14 13:47
;
; (中间带“;”注释内容省略)
;
; The following are the EEPROM contents
;
-----
0x7E4F
0x2134
0x017F
0x8015
0x0018
0x7F00
0x0000
0x8000
0x0000
0x0000
0xA073
```

0xE080  
0x2010  
0x8040  
0x5030  
0x6090  
0xC0A0  
0x0000  
0x0100  
0x027C  
0x957B  
0x70F0  
0x0083  
0x0000

-----  
; The following is the CRC result for  
; the above EEPROM contents

-----  
0xEDFA

-----  
; The following are for CMOSTEK  
; use, customers can ignore them

-----  
0x0000  
0x0018

其中，红色字体部分为具体的配置内容，均为 16bits 的字，一共 24 个字，所以用户需要将 16bits 的字，转换为 8bits 的寄存器内容的格式。转换的方法是，每一个 16bits 的字，高 8bits 是奇数地址，低 8bits 是偶数地址。24 个字转换得到 48 个寄存器配置值，地址范围是 0x00~0x2E（最后一个 8bit 无效）。按上面文件，得到的寄存器内容如下：

表 2-2. 16 位 EEPROM 字到 8 位寄存器内容的转换表

16 位 Word	寄存器地址	8 位寄存器配置值
0x7E4F	0x00	0x4F
	0x01	0x7E
0x2134	0x02	0x34
	0x03	0x21
0x017F	0x04	0x7F
	0x05	0x01
0x8015	0x06	0x15
	0x07	0x80
0x0018	0x08	0x18
	0x09	0x00
0x7F00	0x0A	0x00

16 位 Word	寄存器地址	8 位寄存器配置值
	0x0B	0x7F
0x0000	0x0C	0x00
	0x0D	0x00
0x8000	0x0E	0x00
	0x0F	0x80
0x0000	0x10	0x00
	0x11	0x00
0x0000	0x12	0x00
	0x13	0x00
0xA073	0x14	0x73
	0x15	0xA0
0xE080	0x16	0x80
	0x17	0xE0
0x2010	0x18	0x10
	0x19	0x20
0x8040	0x1A	0x40
	0x1B	0x80
0x5030	0x1C	0x30
	0x1D	0x50
0x6090	0x1E	0x90
	0x1F	0x60
0xC0A0	0x20	0xA0
	0x21	0xC0
0x0000	0x22	0x00
	0x23	0x00
0x0100	0x24	0x00
	0x25	0x01
0x027C	0x26	0x7C
	0x27	0x02
0x957B	0x28	0x7B
	0x29	0x95
0x70F0	0x2A	0xF0
	0x2B	0x70
0x0083	0x2C	0x83
	0x2D	0x00
0x0000	0x2E	0x00

用户仅需把上表内容作为参数，通过 SPI 写入时序，相应写入到 0x01~0x2E 寄存器地址即可。

## 2.6 配置寄存器

上面导出的配置参数地址从 0x01~0x2E，按功能划分可以分为三个区域，分别如下：

表 2-3. 配置寄存器区域划分表

区域	地址	涉及的内容
发射区	0x00 – 0x03	发射频率、发射功率
数据包区	0x04 – 0x27	发射速率、数据包格式（仅适用于硬件数据包发射）
系统区	0x28 – 0x2E	系统工作参数

### ● 发射区

发射区参数关联的寄存器地址范围是 0x00~0x03，这些参数主要是发射的中心频率、发射功率大小。因此，当用户需要在应用中发射多个频率（如跳频发射），或根据电量多少调整发射功率大小等，就可以通过在 RFPDK 进行设置导出不同的参数表，仅需要取其中 0x00~0x03 的这一段即可。所以在此不详加论述这部分各寄存器的含义。

### ● 数据包区

数据包区参数关联的寄存器地址范围是 0x04~0x27，这些参数主要是硬件数据包格式方面的设置项，也仅适用于硬件数据包的发射模式下。具体各寄存器的含义，请阅读本章 2.7。

### ● 系统区

系统区参数关联的寄存器地址范围是 0x28~0x2E，这些参数是与 RF 内部特有参数，与用户应用无关，用户无须理会，仅需按 RFPDK 导出的参数进行配置即可，所以在此也不详加论述。

## 2.7 数据包相关寄存器

下面给出数据包区寄存器的概览。

表 2-4. 数据包区寄存器概览表

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x04	CUS_DIG1	SYMBOL_TIME<7:0>							
0x05	CUS_DIG2	SYMBOL_TIME<15:8>							
0x06	CUS_DIG3			LBD_TH<3:0>				LBD_OUT_EN	LBD_ON
0x07	CUS_DIG4					DEGLITCH_EN	TX_OVERTIMES<2:0>		
0x08	CUS_PKT1	TCYCLE_EN	INTERVAL_EN	STOP_EN	KEY_EN	SYNC_EN	PREAMBLE_LOCATION	PREAMBLE_SEL	PREAMBLE_EN
0x09	CUS_PKT2	TXCYCLE<7:0>							
0x0A	CUS_PKT3	PREAMBLE_LENGTH<7:0>							
0x0B	CUS_PKT4	KEY_LENGTH<2:0>			SYNC_LENGTH<4:0>				
0x0C	CUS_PKT5	SYNC_HEADER<7:0>							
0x0D	CUS_PKT6	SYNC_HEADER<15:8>							
0x0E	CUS_PKT7	SYNC_HEADER<23:16>							

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0F	CUS_PKT8	SYNC_HEADER<31:24>							
0x10	CUS_PKT9	ADDR_ID<7:0>							
0x11	CUS_PKT10	ADDR_ID<15:8>							
0x12	CUS_PKT11	ADDR_ID<23:16>							
0x13	CUS_PKT12	ADDR_ID<31:24>							
0x14	CUS_PKT13	BIT_FORMAT<2:0>			ADDR_LENGTH<4:0>				
0x15	CUS_PKT14					STOP_LENGTH<3:0>			
0x16	CUS_PKT15	BIT_LOGIC_L<7:0>							
0x17	CUS_PKT16	BIT_LOGIC_H<7:0>							
0x18	CUS_PKT17	KEY<7:0>							
0x19	CUS_PKT18								
~	~								
0x21	CUS_PKT26								
0x22	CUS_PKT27	STOP_BIT_L<7:0>							
0x23	CUS_PKT28	STOP_BIT_H<15:8>							
0x24	CUS_PKT29	INTERVAL_LENGTH<7:0>							
0x25	CUS_PKT30	PKT_NUM<7:0>							
0x26	CUS_RESV1								
0x27	CUS_RESV2							INTERVAL_STBY_DIS	STBY_PLLOFF_DIS

注意：

1. 灰色区域表示用户无需理解，并不是没有内容，用户只需要按 RFPDK 导出参数配置即可，用户操作当中个别位置 1 或清 0，需要按“读-改-写”流程。
2. 蓝色区域表示用户需要了解，下文会逐一说明清楚。
3. CMT2189B 内置包结构模式与 CMT2157B 一样，用户可以通过 RFPDK 在 CMT2157B 的配置界面上进行好包结构的选择，生成的 exp 参数已经所需要的方式，按参数顺序配置，软件执行过程中，只需填充具体数据内容。

## 2.7.1 发射速率

发射速率由 SYMBOL\_TIME<15:0>决定的，可以通过 RFPDK 生产。

寄存器名	位数	R/W	比特名	功能说明
CUS_DIG1 (0x04)	7:0	RW	SYMBOL_TIME<7:0>	包格式发射数据率
CUS_DIG2 (0x05)	7:0	RW	SYMBOL_TIME<15:8>	

## 2.7.2 硬件数据包格式

CMT2189B 内部支持硬件数据包结构，其数据帧结构如下所示：



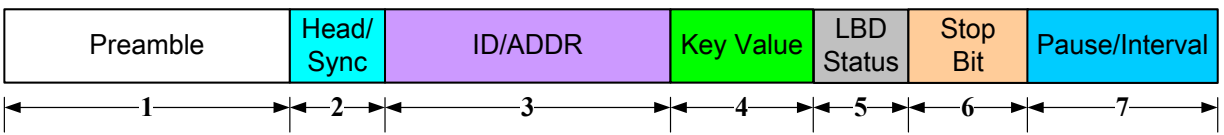


图 2-5. 数据包结构

从上图数据包结构，可以看到主要有 7 个部分构成：

- 1. Preamble: 前导码，可选项，值可以选择 0x55 或 0xAA<sup>[1]</sup>，长度范围从 0~256Byte，任意选择；
- 2. Head/Sync: 同步字，可选项，范围从 0~32 Bits<sup>[2]</sup>，任意选择；
- 3. ID/ADDR: 序列号，必选项，范围从 1~32 逻辑位<sup>[3]</sup>，任意选择；
- 4. Key Value: 键值，可选项，范围从 0~8 逻辑位<sup>[3]</sup>，任意选择；
- 5. LBD Status: 低电状态位，可选项，占 1 逻辑位<sup>[3]</sup>；
- 6. Stop Bit: 结束位，可选项，范围从 0~16 Bits<sup>[2]</sup>，任意选择；
- 7. Pause/Interval: 数据包间隔，固定发“0”，范围从 0~256 Bits<sup>[2]</sup>；

注意：

- [1]. 前导码不进行编码，即采用 NRZ 格式，如选择 0x55，按设置的速率，数据流为 0b01010101（0 代表低电平，1 代表高电平）；
- [2]. Head/Sync、Stop Bit 和 Pause/Interval 都是不进行编码，按设置的速率进行输出，如同前导码一样；
- [3]. 逻辑位，表示编码后的位，下文有详细说明；上述报文结构中编号 3~5 都是支持内部编码器进行编码；

举例来说，编码规则选择最少 1 符号作为 1 逻辑位，即 0b0 就是逻辑 0，0b1 就是逻辑 1（即 NRZ 编码）情况，设置如下：

- Preamble: 设置开，值为 0xAA，长度配置为 5 Bytes；
- Head/Sync: 设置开，值为 0x2DD4，长度 2 Bytes；
- ID/ADDR: 值为 0x12345678，长度 4 Bytes；
- Key Value: 值为 0x9A；
- LBD Status: 设置关闭；
- Stop Bit: 设置开，值为 0xBCDE，长度 16 bits（即 2 Bytes）；
- Pause/Interval: 设置开，长度为 32 bits（即 4Bytes）；

那么数据流如下：

0xAA AA AA AA AA 2D D4 12 34 56 78 9A BC DE 00 00 00 00 AA AA AA AA AA 2D D4 12 34 56 78 9A BC DE 00...

2.7.3 Preamble

表 2-5. Preamble 配置寄存器相关

寄存器名	位数	R/W	比特名	功能说明
CUS_PKT1 (0x08)	2	RW	PREAMBLE_LOCATION	Tcycle 使能时，Preamble 在包结构中的位置： 0: 在 1 个 cycle 内，每包都包含 1 个 Preamble， 即若 1 个 cycle 内有 N 个包，则包含 N 个 Preamble 1: 在 1 个 cycle 内，只包含 1 个 Preamble，且只

寄存器名	位数	R/W	比特名	功能说明
				出现在第一包 注：关于 Tcycle 的概念，后续有详细说明。
	1	RW	PREAMBLE_SEL	Preamble 值： 0: 0x55 1: 0xAA
	0	RW	PREAMBLE_EN	Preamble 使能： 0: 不使能 1: 使能
CUS_PKT3 (0x0A)	7:0	RW	PREAMBLE_LENGTH<7:0>	Preamble 的长度，可配置为 0~255，0 表示发送 1 个 byte 的 Preamble，如此类推，255 表示发送 256 个 byte 的 Preamble。

对用户而言，若配置 PREAMBLE\_EN 为 0，则不发送 Preamble，若配置为 1，则发送 1-256 个 byte 的 Preamble。

## 2.7.4 Head/Sync

表 2-6. Head/Sync 配置寄存器相关

寄存器名	位数	R/W	比特名	功能说明
CUS_PKT1 (0x08)	3	RW	SYNC_EN	Sync 使能： 0: 不使能 1: 使能
CUS_PKT4 (0x0B)	4:0	RW	SYNC_LENGTH<4:0>	Sync 长度，可配置为 0~31，0 表示发送 1 个 Symbol 的 Sync，如此类推，31 表示发送 32 个 Symbol 的 Sync，Symbol 长度随意。
CUS_PKT5 (0x0C)	7:0	RW	SYNC_HEADER<7:0>	Sync 的值，根据不同的 SYNC_LENGTH 设置来填入不同的寄存器，请看下表。
CUS_PKT6 (0x0D)	7:0	RW	SYNC_HEADER<15:8>	
CUS_PKT7 (0x0E)	7:0	RW	SYNC_HEADER<23:16>	
CUS_PKT8 (0x0F)	7:0	RW	SYNC_HEADER<31:24>	

表 2-7. Head/Sync 长度选择与寄存器对应关系

SYNC_LENGTH	SYNC/HEADER			
	<31:24>	<23:16>	<15:8>	<7:0>
0~7	√			
8~15	√	√		
16~23	√	√	√	
24~31	√	√	√	√

表中打勾的地方表示要填入的寄存器。举例说明,如果 SYNC\_LENGTH 设置为 15,即长度为 16 个 Symbol, Sync 的值为 0x5678,那么用户就将这个值填入 SYNC\_HEADER<31:24>和 SYNC\_HEADER<23:16>两个寄存器,msb 对应的是第 31 位,lsb 对应的是第 16 位,即将 0x56 填入 SYNC\_HEADER<31:24>, 0x78 填入 SYNC\_HEADER<23:16>。对用户而言,若配置 Sync 使能为 0,则不发送 Sync,若配置使能为 1,则发送 1-32 个 Symbol 的 Sync。

## 2.7.5 Addr/ID

表 2-8. Addr/ID 配置寄存器相关

寄存器名	位数	R/W	比特名	功能说明
CUS_PKT13 (0x14)	4:0	RW	ADDR_LENGTH<4:0>	Addr ID 长度,可配置为 0~31,0 表示发送 1 个 Logic bit 的 Addr,如此类推,31 表示发送 32 个 Logic bit 的 Addr, Logic bit 长度随意。
	7:5	RW	BIT_FORMAT<2:0>	1 个 Logic bit 包含的 Symbol 个数,可配置为 0~7,0 表示 1 个 Symbol,如此类推,7 表示 8 个 Symbol。
CUS_PKT15 (0x16)	7:0	RW	BIT_LOGIC_L<7:0>	Logic 0 定义
CUS_PKT16 (0x17)	7:0	RW	BIT_LOGIC_H<7:0>	Logic 1 定义
CUS_PKT9 (0x10)	7:0	RW	ADDR_ID<7:0>	Addr ID 值
CUS_PKT10 (0x11)	7:0	RW	ADDR_ID<15:8>	
CUS_PKT11 (0x12)	7:0	RW	ADDR_ID<23:16>	
CUS_PKT12 (0x13)	7:0	RW	ADDR_ID<31:24>	

表 2-9. Addr/ID 长度选择与寄存器对应关系

ADDR_LENGTH	ADDR_ID			
	[31:24]	[23:16]	[15:8]	[7:0]
0~7	√			
8~15	√	√		
16~23	√	√	√	
24~31	√	√	√	√

表中打勾的地方表示要填入的寄存器。举例说明,如果 ADDR\_LENGTH 设置为 15,即长度为 16 个 Logic bit,值为 0x5678,那么用户就将这个值填入 ADDR\_ID<31:24>和 ADDR\_ID<23:16>两个寄存器,msb 对应的是第 31 位,lsb 对应的是第 16 位,即将 0x56 填入 ADDR\_ID<31:24>, 0x78 填入 ADDR\_ID<23:16>。

表 2-10. 编码长度选择与位使能对应关系

BIT_FORMAT	BIT_LOGIC_L/ BIT_LOGIC_H							
	7	6	5	4	3	2	1	0
0	√							
1	√	√						
2	√	√	√					
3	√	√	√	√				
4	√	√	√	√	√			
5	√	√	√	√	√	√		
6	√	√	√	√	√	√	√	
7	√	√	√	√	√	√	√	√

表中打勾的地方表示要填入的寄存器对应位。举例说明，如果 BIT\_FORMAT 设置为 3，即 1 个 Logic bit 包含 4 个 Symbol，若 Logic 0 = 0b'1000'，那么用户就将这个值填入 BIT\_LOGIC\_L<7:4>，若 Logic 1 = 0b'1110'，那么用户就将这个值填入 BIT\_LOGIC\_H<7:4>，msb 对应的是第 7 位，lsb 对应的是第 4 位。

将上面的例子结合起来，若 ADDR\_LENGTH 设置为 20，ADDR\_ID<31:12>为 0x56789，BIT\_FORMAT 设置为 4，BIT\_LOGIC\_L<7:4>为 0b'1000'，BIT\_LOGIC\_H<7:4>为 0b'1110'，将 ADDR\_ID 展开成 Symbol，则发射的数据如下：

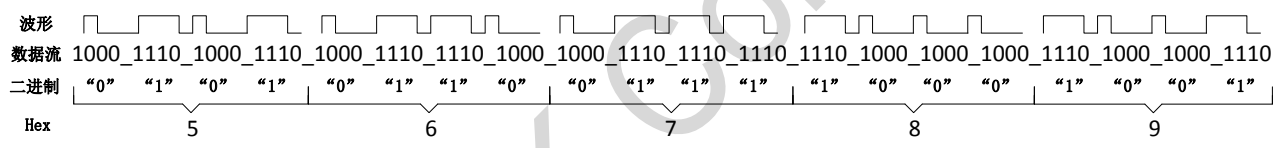


图 2-6. ADDR ID 示例图

即为，ID/ADDR = 0h'8E8E\_8EE8\_8EEE\_E888\_E88E'，共 80 个 Symbol，从高位开始发射。

2.7.6 Key Value

表 2-11. Key Value 配置寄存器相关

寄存器名	位数	R/W	比特名	功能说明
CUS_PKT1 (0x08)	4	RW	KEY_EN	Key 使能位 0: 不使能 1: 使能
CUS_PKT4 (0x0B)	7:5	RW	KEY_LENGTH<2:0>	Key Value 长度，可配置为 0~7，0 表示发送 1 个 Logic bit 的 Key，如此类推，7 表示发送 8 个 Logic bit 的 Key，Logic bit 长度随意。
CUS_PKT17 (0x18)	7:0	RW	KEY<7:0>	Key Value

发射包结构中对应的键值长度最大可配置为 8，下表中打勾的地方表示要填入的寄存器对应位。举例说明，如果 KEY\_LENGTH 设置为 3，即长度为 4 个 Logic bit，值为 0x5，那么用户就将这个值填入 KEY0<7:4>。至于将 Logic bit 展开为 Symbol 的配置，请参考 Addr ID 的相关说明。

表 2-12. Key Value 长度选择与位使能对应关系

KEY_LENGTH	KEY Value							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	√							
1	√	√						
2	√	√	√					
3	√	√	√	√				
4	√	√	√	√	√			
5	√	√	√	√	√	√		
6	√	√	√	√	√	√	√	
7	√	√	√	√	√	√	√	√

对用户而言，若配置 KEY\_EN 为 0，则不发送键值，若配置为 1，则发送 1-8 个 Logic bit 的键值。

## 2.7.7 LBD Status 配置

表 2-13. LBD Status 配置寄存器相关

寄存器名	位数	R/W	比特名	功能说明
CUS_DIG3 (0x06)	0	RW	LBD_ON	LBD（低电压检测）使能： 0：不使能 1：使能
	1	RW	LBD_OUT_EN	LBD 结果输出使能（即是否添加在报文中） 0：不使能 1：使能
	5:2	RW	LBD_TH<3:0>	LBD 的电压比较阈值，若实际电压大于该值，LBD 结果为 1（逻辑 1），反之，为 0（逻辑 0）。
CUS_LBD_RESULT (0x4B)	3:0	R	LBD_RESULT<3:0>	电压测量值

注意：CUS\_LBD\_RESULT 寄存器是功能寄存器，不是通过 RFPDK 导出的参数寄存器，详细可见本章 2.8。

LBD 的工作原理是，用户设置电压值 LBD\_TH，代表的是 LBD 的阈值，然后芯片通过测试得到实际的电压值 LBD\_RESULT，并将 LBD\_RESULT 跟 LBD\_TH 进行对比，如果发现 LBD\_RESULT 比 LBD\_TH 小，那么就代表低电压已经发生，反之，电压正常。根据比较结果内部会输出 LBD 指示信号，如果配置 LBD\_OUT\_EN 为 1，那么 LBD 结果会作为数据包的一部分发射出去，具体参见 2.7.1 小节包结构。

芯片内部采用 4bit 的 ADC 对电压值进行转换，每 0.2v 一个步进，得到 LBD\_RESULT，电压值与 LBD\_RESULT 的关系如下：

表 2-14. LBD\_TH 配置寄存器相关

NUM	LBD_TH	LBD_TH<3:0> or LBD_RESULT<3:0>
1	<1.45v	4'b0000
2	1.45~1.65v	4'b0001
3	1.65~1.85v	4'b0010
4	1.85~2.05v	4'b0011
5	2.05~2.25v	4'b0100
6	2.25~2.45v	4'b0101
7	2.45~2.65v	4'b0110
8	2.65~2.85v	4'b0111
9	2.85~3.05v	4'b1000
10	3.05~3.25v	4'b1001
11	3.25~3.45v	4'b1010
12	3.45~3.65v	4'b1011
13	3.65~3.85v	4'b1100
≥14	>3.85	4'b1101

注意:

1. LBD 不是实时做的, 而是在 PLL 做频点校正之后做一次, 所以用户需要通过先切换到 STBY 状态后, 再切换到 Tx 状态来触发这个测量过程。
2. LBD\_RESULT 可以作为一个测量判断的条件, 所以用户可以在触发 LBD 测量后, 通过读取 LBD\_RESULT 作为定量分析处理。

## 2.7.8 Stop Bit 配置

表 2-15. Stop Bit 配置寄存器相关

寄存器名	位数	R/W	比特名	功能说明
CUS_PKT1 (0x08)	5	RW	STOP_EN	Stop Bit 使能: 0: 不使能 1: 使能
CUS_PKT14 (0x15)	3:0	RW	STOP_LENGTH<3:0>	STOP_BIT 的值
CUS_PKT27 (0x22)	7:0	RW	STOP_BIT<7:0>	
CUS_PKT28 (0x23)	7:0	RW	STOP_BIT<15:8>	

表 2-16. Stop Bit 长度选择与位使能对应关系

STOP_LENGTH	STOP_BIT	
	<15:8>	<7:0>
0~7	√	
8~15	√	√

表中打勾的地方表示要填入的寄存器。举例说明，如果 STOP\_LENGTH 设置为 7，即长度为 8 个 Symbol，值为 0x56，那么用户就将这个值填入 STOP\_BIT[15:8]，msb 对应的是第 15 位，lsb 对应的是第 8 位。

对用户而言，若配置 STOP\_EN 为 0，则不发送 Stop Bit，若配置为 1，则发送 1-8 个 Symbol 的 Stop Bit。

2.7.9 Pause/Interval 配置

表 2-17. Pause/Interval 配置寄存器相关

寄存器名	位数	R/W	比特名	功能说明
CUS_PKT1 (0x08)	6	RW	INTERVAL_EN	Interval 使能： 0：不使能 1：使能
CUS_PKT29 (0x24)	7:0	RW	INTERVAL_LENGTH<7:0>	Interval 长度，可配置为 0~255，0 表示发送 1 个 Symbol 的 0，如此类推，255 表示发送 256 个 Symbol 的 0，Symbol 长度随意。 注：Interval 固定发送 0。

注意：严格意义上讲，Pause/Interval 为发包间隔，只发若干个 Symbol 的 0，不算是包结构的有效部分。

对用户而言，若配置 INTERVAL\_EN 使能为 0，则不发送 Pause/Interval，若配置为 1，则发送 1-256 个 Symbol 的 Pause/Interval。

2.7.10 Tcycle 配置

Tcycle 是指硬件数据包发射模式下的一个发射周期（发射过程），这个发射周期可以包含发射 N 次重复的数据包，同时也支持数据包中间增加间隔停顿时间，或是按 n 个数据包组成一组，一个发射周期包含 m 组的分组方式。这几个方式如下图所示。

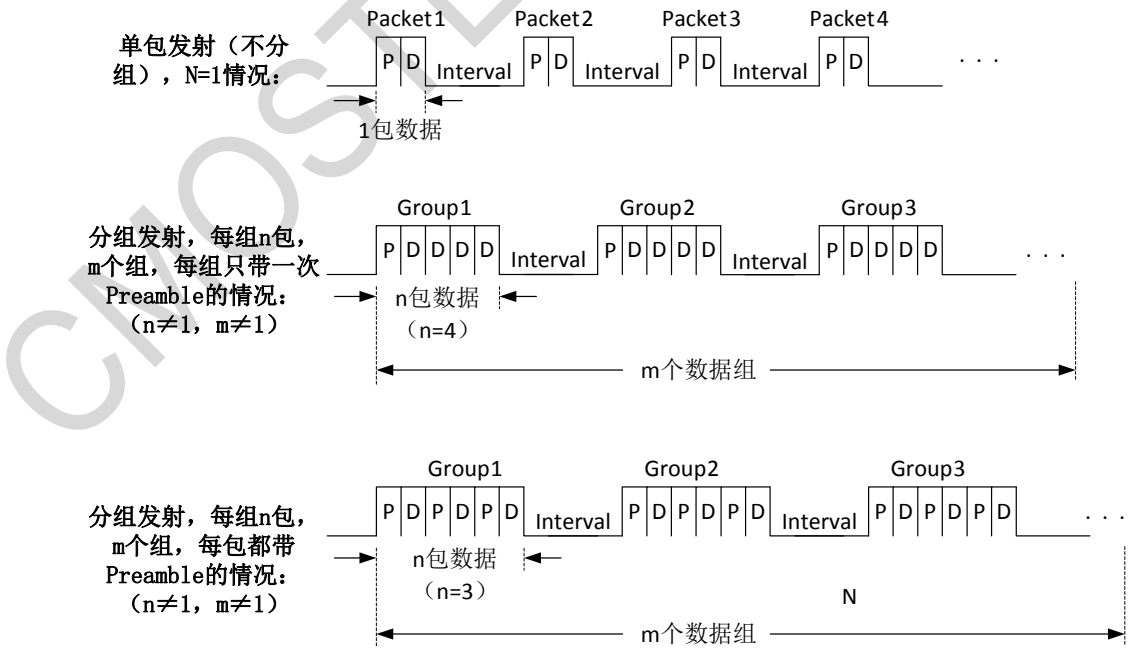


图 2-7. 各种发包方式图示

- 若 Tcycle 不使能，则按上图第一种情况，如果 Preamble 使能，则 Preamble 固定出现在每包的开始。
- 若 Tcycle 使能，Preamble 使能，且 PREAMBLE\_LOCATION = 1，则按上图第二种情况。即 Preamble 在一个分组内出现一次，且出现在每个分组的开始，在一个发射周期内共出现 m（m = TXCYCLE）次。
- 若 Tcycle 使能，Preamble 使能，且 PREAMBLE\_LOCATION = 0，则按上图第三种情况，即 Preamble 在每包的开始出现，在一个分组内出现 n（n = PKT\_NUM）次，在一个发射周期内出现 m\*n（TXCYCLE \* PKT\_NUM）次。

注意：

1. 上图的 P = Preamble，D = Sync/Head + Addr/ID + Key Value + LBD Status + Stop Bit
2. Preamble 不使能，则不发送 Preamble。
3. Interval 不使能，则不发送 Interval。
4. Tcycle 不使能，则 preamble\_location 需配置为 0。
5. Tcycle 不使能，则 Tcycle 次数需配置为 0。

表 2-18. Tcycle 配置寄存器相关

寄存器名	位数	R/W	比特名	功能说明
CUS_PKT1 (0x08)	2	RW	PREAMBLE_LOCATION	Tcycle 使能时，Preamble 在包结构中的位置： 0：在 1 个 cycle 内，每包都包含 1 个 Preamble， 即若 1 个 cycle 内有 N 个包，则包含 N 个 Preamble 1：在 1 个 cycle 内，只包含 1 个 Preamble，且只出现在第一包
	7	RW	TCYCLE_EN	Tcycle 使能： 0：不使能 1：使能
CUS_PKT2 (0x09)	7:0	RW	TXCYCLE[7:0]	1 次发射周期内，cycle 循环的次数： 0-255 表示发送 1-256 个 Tcycle
CUS_PKT30 (0x25)	7:0	RW	PKT_NUM[7:0]	1 个 cycle 发的包个数： 0-255 表示发送 1-256 个包

2.8 状态及功能寄存器

表 2-19. 状态及功能寄存器概览表

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x2F	CUS_SOFTRST	0xFF							
0x33	CUS_MODE					STBY		TX	SLEEP
0x34	CUS_DATA					DATAIN_EN			
0x4B	CUS_LBD_RESULT					LBD_RESULT			
0x4D	CUS_STATUS					MAIN_STATUS			

上面的状态及功能寄存器不能在 RFPDK 中导出，但用户在高级配置模式下是需要使用到的，当中涉及到芯片工作状态的切换、状态读取以及复位等。



2.8.1 软复位

通过 SPI 总线，往 CUS\_SOFRST（0x2F）写入 0xFF，可以实现对 RF 部分的复位处理。

表 2-20. 软复位寄存器

控制寄存器	命令类型	命令	写入值	功能说明
CUS_SOFRST (0x2F)	软复位	soft_rst	0xFF	复位芯片，使芯片重新进行初始化流程，并回到 SLEEP 状态

注：

- 1. 执行 soft\_rst 后，芯片重新进行复位流程，用户需要重新按需求更新配置寄存器；
- 2. 在上电后，配置寄存器（CFG）之前，配置寄存器按内部默认值（即按出厂烧录 EERPOM 值）。

2.8.2 工作状态以及状态切换

CMT2189B 的 RF 有 3 个主要工作状态，在高级配置模式下状态以及切换如下图所示：

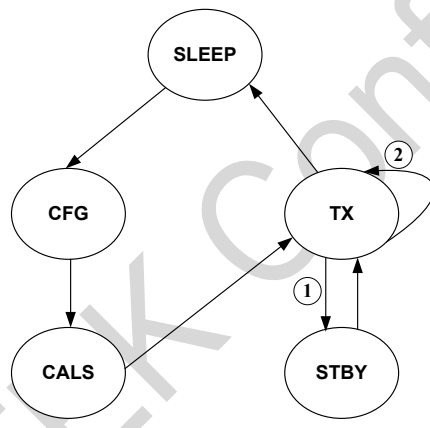


图 2-8. RF 工作状态以及切换

表 2-21. 各状态说明

状态/过程	描述
SLEEP	低功耗模式，关闭所有模块
CFG	配置寄存器过程
CALS	模拟模块的校正
TX	发射数据包，所有关于发射机的模块都会打开
STBY	发射间隔省电模式，关闭 PA，PLL 可根据配置开启，振荡器保持工作

注意：

- 1. CFG 为 MCU 通过 SPI 配置芯片工作参数的过程；
- 2. CALS 为 RF 内部模拟模块校正的过程，这个在启动发射后，RF 内部会自动会执行并处理完毕后切换到 TX 状态；

表 2-22. 状态切换命令及复位命令

控制寄存器	命令类型	命令	写入值	功能说明
CUS_MODE (0x33)	状态跳转	go_sleep	0x01	回到 SLEEP 状态
		go_tx	0x02	进入 TX 状态
		go_stby	0x08	进入 STBY 状态

### 2.8.3 工作状态查询

用户需要查询当前工作状态，可以通过读取 CUS\_STATUS (0x4D) 状态寄存器，其中低 4 位表示当前工作状态。具体参照下表。

表 2-23. 状态切换命令及复位命令

状态寄存器	状态编码	说明
CUS_STATUS (0x4D<3:0>)	0b0000	SLEEP
	0b1010	TX
	0b1101	STBY

## 2.9 硬件数据包发射模式

### 2.9.1 上电初始化

1. 芯片上电，端口初始化，等待 20ms 稳定时间；
2. 控制 RFCTRL = 1；
3. PC3/CSB、PC2/SCLK 和 PC4/RFDIN 配置为高阻输入；PC1/SDIO 配置为输出，置 0；
4. 初始化完成，芯片可以进入睡眠低功耗状态，或做其它处理；

### 2.9.2 发射流程

要实现硬件数据包模式，需要按如下流程进行：

1. 控制 RFCTRL = 0，使能 RF 部分的 SPI 接口，并立刻发送 soft\_rst 命令，等待 20ms 稳定时间，确保进软复位完成；
2. 通过 SPI 发送 go\_stby 命令，并确认 RF 处于 Standby 状态（建议发送命令后，等待 2ms）；
3. 按用户需求配置相关的寄存器（发射区、数据包格式等，配置范围 0x00~0x2E）
  - a) 把 RFPDK 导出的参数全部按顺序配置一遍；
  - b) 如有需要，就针对需要修改的个别寄存器再修改（如填充 FIFO 内容等）；
4. 把寄存器 0x34<5>置 0，然后把寄存器 0x2A<1:0>设为 0b00；
5. 通过 SPI 发送 go\_tx 命令，启动芯片按上述配置进行硬件数据包发射；
6. 通过 SPI 读取状态（按一定周期间隔读取），判断芯片是否发射完毕（返回到 Sleep 状态则为发射完毕）；
7. 发射完毕后，需要把 RFCTRL 置高，SPI 和 RFDIN 切换为高阻输入（SPI 和 RFDIN 内部带上拉），MCU 此后也进入 SLEEP 模式，即整机进入低功耗状态。

注意:

1. 需要再次启动发射时, 按发射流程重新执行一遍;
2. 第 4 步操作时, 需要采用“读一改一写”的方式, 以免错改其它位的值;
3. 发射完毕不能判断是否 Standby, 因为存在数据包间隔过程可能是 Standby 状态;
4. 发射结束后的内部管脚状态处理, 与上电初始化有区别, 需要注意。

## 2.10 直通发射模式 (高级配置模式)

### 2.10.1 上电初始化

1. 芯片上电, 端口初始化, 等待 20ms 稳定时间;
2. 控制 RFCTRL = 1;
3. PC3/CSB、PC2/SCLK 和 PC4/RFDIN 配置为高阻输入; PC1/SDIO 配置为输出, 置 0;
4. 初始化完成, 芯片可以进入睡眠低功耗状态, 或做其它处理;

### 2.10.2 发射流程

要实现配置模式下的直通发射, 需要按如下流程进行:

1. 把 PC0/RFDIN 设置为置 0 输出;
2. 控制 RFCTRL = 0, 使能 RF 部分的 SPI 接口, 并立刻发送 soft\_rst 命令, 等待 20ms 稳定时间
3. 把寄存器 0x34<3>置 1, 并确认修改成功;
4. 通过 SPI 发送 go\_stby 命令, 确认 RF 处于 Standby 状态 (建议发送命令后, 等待 2ms);
5. 按用户需求配置相关的寄存器 (配置范围 0x00~0x03)
6. 把寄存器 0x34<5>置 0, 然后把寄存器 0x2A<1:0>设为 0b01;
7. 通过 SPI 发送 go\_tx 命令, 启动芯片按上述配置进行发射 (建议发送命令后, 等待 1ms);
8. 控制 PC0/RFDIN 发送需要的数据流;
9. 发送完毕后, 保持 PC0/RFDIN 置 0 输出;
10. 把寄存器 0x02<6:0>设置为 0, 然后把寄存器 0x4E<7>置 1;
11. 把寄存器 0x34<3>置 0, 并确认修改成功;
12. 通过 SPI 发送 go\_sleep 命令, 进入 Sleep 状态;
13. 需要把 RFCTRL 置高, PC3/CSB、PC2/SCLK 和 PC4/RFDIN 切换为高阻输入 (内部带上拉), PC1/SDIO 切换为输出置 0, 然后 MCU 此后也进入 SLEEP 模式, 即整机进入低功耗状态。

注意:

1. 第 5 步配置寄存器, 只需要配置 0x00~0x03 即可, 可以实现发射频率和发射功率的调整, 其它可以忽略;
2. 需要再次启动发射时, 按发射流程重新执行一遍;
3. 第 3、6、10、11 步操作时, 需要采用“读一改一写”的方式, 以免错改其它位的值;

### 2.10.3 相关寄存器

表 2-24. 直通模式相关寄存器

寄存器名	位数	R/W	比特名	功能说明
CUS_DIG4 (0x07)	2:0	RW	TX_OVERTIMES[2:0]	结束发射的超时退出时间，可选配置为 0~7。 超时的计算公式为： $T = 20\text{ms} + \text{TX\_OVERTIMES} * 10\text{ms}$ 即：超时退出时间范围为 [20 ms, 90 ms]
CUS_DATA (0x34)	3	RW	DATAIN_EN	RFDIN 发射模式的数据输入脚使能 0：不使能 1：使能

在直通模式下，发射数据的数据率受控于 RFDIN（PC0），当 PC0 保持输出 0 大于 TX\_OVERTIMES 设置的时间，则会自动进入超时退出机制，停止当前的发射，直到 PC0 下次的上升沿触发。所以用户需要根据编码格式进行考虑，TX\_OVERTIMES 默认是 20ms，可以按步进 10ms 增加到 90ms。

### 3 程序存储器

程序地址寄存器为 13 位，最多支持访问 8K 空间 (0x0000~0x1FFF)，但实际芯片程序存储器为 2K Words，外加上 4 个额外的用户配置 (UCFGx)、工程配置区 (FCFGx)，共有 64 Words，它们均由 EEPROM 构成。其中，0~0x7FF 对应主程序区访问，其中未实现部分 0x800~0x1FFF 保留。用户和工厂配置信息区从 0x2000 开始，到 0x203F 结束。

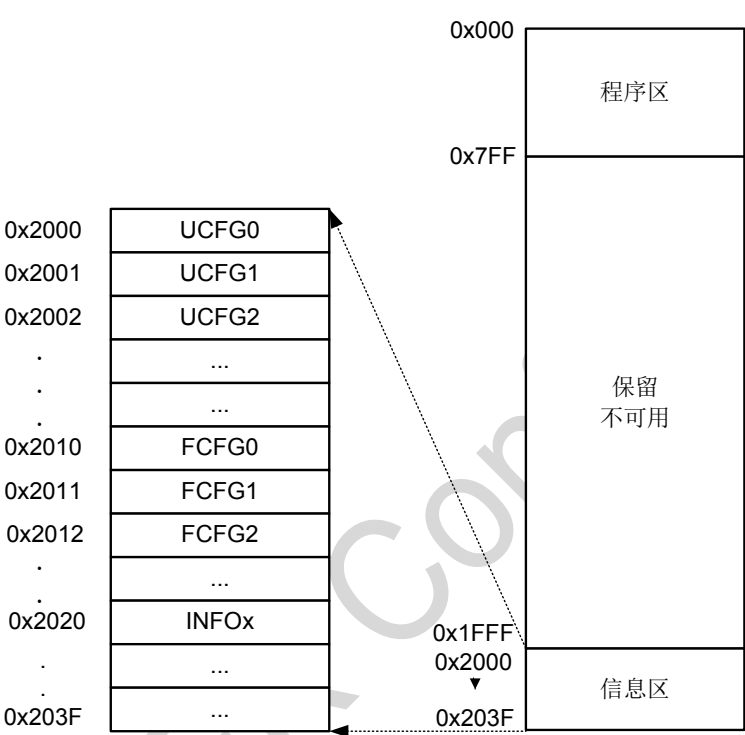


图 3-1. 程序空间地址映射

## 4 功能寄存器（SFR）

### 4.1 地址映射

#### 4.1.1 Bank0 SFR

表 4-1. Bank0 寄存器列表

ADDR	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	POR reset
0	INDF	使用 FSR 的内容对数据存储寄存器进行访问（非物理寄存器）								xxxx xxxx
1	TMR0	Timer0<7:0>								xxxx xxxx
2	PCL	程序计数器 PC<7:0>								0000 0000
3	STATUS	-	-	PAGE	/TF	/PF	Z	HC	C	--01 1xxx
4	FSR	间接寻址指针寄存器								
5	PORTA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	00x0 0000
6										---- ----
7	PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	0000 0000
8										---- ----
9										---- ----
A	PCLATH	-	-	-	程序计数器 PC<13:8>					---0 0000
B	INTCON	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000
C	PIR1	EEIF	CKMEAIF	-	C2IF	C1IF	OSFIF	TMR2IF	-	00-0 000-
D										---- ----
E										---- ----
F										---- ----
10										---- ----
11	TMR2	Timer2<7:0>								0000 0000
12	T2CON	-	TOUTPS<3:0>				TMR2ON	T2CKPS<1:0>		-000 0000
13										---- ----
14										---- ----
15										---- ----
16										---- ----
17										---- ----
18	WDTCON	-	-	-	WDTPS<3:0>				SWDTEN	---0 1000
19	CMCON0	C2OUT	C1OUT	C2INV	C1INV	CIS	CM<2:0>			0000 0000
1A	PR0	PR0<7:0>								1111 1111
1B	MSCKCON	-	-	-	SLVREN	-	CKMAVG	CKCNTI	-	---0 -00-
1C	SOSCPRL	SOSCPR<7:0>								1111 1111
1D	SOSCPRH	-	-	-	-	SOSCPR<11:8>				---- 1111
1E										---- ----
1F										---- ----
20-7F		Bank0 的 SRAM, 96Byte 通用 RAM								xxxx xxxx

## 4.1.2 BANK1 SFR

表 4-2. Bank1 寄存器列表

ADDR	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	POR reset
80	INDF	使用 FSR 的内容对数据存储寄存器进行访问（非物理寄存器）								xxxx xxxx
81	OPTION	/PAPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111
82	PCL	程序计数器 PC<7:0>								0000 0000
83	STATUS	-	-	PAGE	/TF	/PF	Z	HC	C	--01 1xxx
84	FSR	间接寻址指针寄存器								
85	TRISA	TRISA<7:6>		PA5	TRISA<4:0>					11x1 1111
86										---- ----
87	TRISC	TRISC<7:0>								1111 1111
88										---- ----
89										---- ----
8A	PCLATH	-	-	-	程序计数器 PC<13:8>					---0 0000
8B	INTCON	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000
8C	PIE1	EEIE	CKMEAIE	-	C2IE	C1IE	OSFIE	TMR2IE	-	00-0 000-
8D										---- ----
8E	PCON							/POR	/BOR	---- --q q
8F	OSCCON	LFMOD	IRCF<2:0>			OSTS	HTS	LTS	SCS	0101 x000
90										---- ----
91										0000 0000
92	PR2	PR2<7:0>, Timer2 period register								1111 1111
93										---- ----
94										---- ----
95	WPUA	WPUA<7:6>		-	WPUA<4:0>					11-1 1111
96	IOCA	IOCA<7:0>								---- ----
97										---- ----
98										---- ----
99	VRCON	VREN	-	VRR	-	VR<3:0>				0-0- 0000
9A	EEDAT	EEDAT<7:0>								0000 0000
9B	EEADR	EEADR<7:0>								0000 0000
9C	EECON1	-	-	WREN3	WREN2	WRERR	WREN1	-	RD	--00 x0-0
9D	EECON2	-	-	-	-	-	-	-	WR	---- ---0
9E										---- ----
9F										---- ----
A0-BF		Bank1 的 SRAM, 32Byte 通用 RAM								xxxx xxxx
C0-EF										---- ----
F0-FF		SRAM,访问为 Bank0 的 0x70~0x7F								xxxx xxxx

注意:

1. INDF 不是物理寄存器;
2. 灰色部分表示没有实现, 请不要访问;
3. “-”表示未实现, 未实现的寄存器位不要写 1 或使用, 以后芯片升级需要用到。

### 4.1.3 TMR0 (Addr:0x01)

表 4-3. TMR0 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMR0	Timer0<7:0>, 计数结果寄存器							
Reset	X	X	X	X	X	X	X	X
Type	RW							

### 4.1.4 STATUS (Addr:0x03)

表 4-4. STATUS 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
STATUS	-	-	PAGE	/TF	/PF	Z	HC	C
Reset	-	-	0	1	1	X	X	X
Type	-	-	RW	R	R	RW	RW	RW

表 4-5. STATUS 位功能描述

Bit	Name	Function
7:6	-	没有功能, 读 0
5	PAGE	寄存器 BANK 选择: 0 = BANK0 (00h-7Fh) 1 = BANK1 (80h-FFh)
4	/TF	Time Out Bit 1 = 上电后 (POR), 执行过 CLRWDWT 或 SLEEP 指令 0 = 看门狗 (WDT) 发生过 Time-out
3	/PF	Power Down Bit 1 = 上电后 (POR), 或执行过 CLRWDWT 指令 0 = 执行 SLEEP 指令
2	Z	Zero Bit 1 = 加减计算结果或逻辑运算为 0 0 = 加减计算结果或逻辑运算非 0
1	HC	半进/借位 (ADDWF、ADDLW、SUBLW、SUBWF 指令) 1 = 计算结果产生 Bit4 进/借位 0 = 计算结果没有产生 Bit4 进/借位
0	C	进/借位 (ADDWF、ADDLW、SUBLW、SUBWF 指令) 1 = 计算结果产生进/借位 0 = 计算结果没有产生进/借位



表 4-6. 各复位状态下标志情况

/TF	/PF	条件
1	1	上电或者低电复位
0	u	WDT 复位
0	0	WDT 唤醒
u	u	正常运行下发生 MCLR 复位
1	0	睡眠状态下发生 MCLR 复位

注意:

1. 和其他寄存器一样，状态寄存器也可以作为任何指令的目标寄存器。如果一条指令影响 Z、HC 或 C 位的指令以状态寄存器作为目标寄存器，将禁止对这三位的写操作，他们只受逻辑结果影响，被置 1 或清 0。因此，当执行一条把状态寄存器作为目标寄存器的指令后，STATUS 内容可能和预想的不一致；
2. 建议只使用 BCR、BSR、SWAPR 和 STR 指令来改变状态寄存器。

#### 4.1.5 PORTA (Addr:0x05)

表 4-7. PORTA 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Reset	X	X	X	X	X	X	X	X
Type	RW	RW	R	RW	RW	RW	RW	RW

表 4-8. PORTA 位功能描述

Bit	Name	Function
7	PA7	PORTA7 数据
6	PA6	PORTA6 数据
5	PA5	PORTA5 只有输入功能，不存在相应的输出数据寄存器
4	PA4	PORTA4 数据
3	PA3	PORTA3 数据
2	PA2	PORTA2 数据
1	PA1	PORTA1 数据
0	PA0	PORTA0 数据

#### 4.1.6 PORTC (Addr:0x07)

表 4-9. PORTC 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Reset	X	X	X	X	X	X	X	X
Type	RW	RW	RW	RW	RW	RW	RW	RW

表 4-10. PORTC 位功能描述

Bit	Name	Function
7	PC7	PORTC7 数据
6	PC6	PORTC6 数据
5	PC5	PORTC5 数据
4	PC4	PORTC4 数据
3	PC3	PORTC3 数据
2	PC2	PORTC2 数据
1	PC1	PORTC1 数据
0	PC0	PORTC0 数据

## 4.1.7 INTCON (Addr:0x0B)

表 4-11. INTCON 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTCON	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

表 4-12. INTCON 位功能描述

Bit	Name	Function
7	GIE	全局中断使能 1 = 使能所有已经设置打开的中断 0 = 禁止所有中断
6	PEIE	外设中断使能 1 = 使能所有已经设置打开的外设中断 0 = 禁止所有外设中断
5	T0IE	Timer0 溢出中断使能 1 = 使能 0 = 禁止
4	INTE	PA2/INT 外部中断使能 1 = 使能 0 = 禁止
3	PAIE	PORTA 端口变化中断 1 = 使能 PORTA<7:0>端口变化中断 0 = 禁止 PORTA<7:0>端口变化中断
2	T0IF	Timer0 溢出中断标志位 1 = Timer0 产生溢出 (必须通过软件清除) 0 = Timer0 仍未产生溢出
1	INTF	PA2/INT 外部中断标志位 1 = PA2/INT 外部中断产生 (必须通过软件清除) 0 = PA2/INT 外部中断未产生
0	PAIF	PORTA 端口变化中断标志位

1 = PORTA<7:0>任意一个或多个端口产生变化（必须通过软件清除） 0 = PORTA<7:0>中任何一个端口状态没有变化
--

#### 4.1.8 PIR1 (Addr:0x0C)

表 4-13. PIR1 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PIR1	EEIF	CKMEAIF	-	C2IF	C1IF	OSFIF	TMR2IF	-
Reset	0	0	-	0	0	0	0	-
Type	RW	RW	-	RW	RW	RW	RW	-

表 4-14. PIR1 位功能描述

Bit	Name	Function
7	EEIF	EEPROM 写中断标志 1 = EE 写操作完成（必须软件清除） 0 = EE 写操作未完成
6	CKMEAIF	快时钟测量慢时钟操作完成中断标志 1 = 快时钟测量慢时钟操作完成（必须软件清除） 0 = 快时钟测量慢时钟未完成
5	-	保留位，不要写 1
4	C2IF	比较器 2 中断标志位 1 = 比较器 2 输出发生了变化 0 = 比较器 2 输出未发生改变
3	C1IF	比较器 1 中断标志位 1 = 比较器 1 输出发生了变化 0 = 比较器 1 输出未发生改变
2	OSFIF	振荡器故障中断标志位 1 = 系统振荡器发生故障，时钟输入切换为 INTOSC（必须软件清除） 0 = 系统时钟运行正常
1	TMR2IF	Timer2 与 PR2 比较相等中断标志位 1 = Timer2 的值等于 PR2（必须软件清除） 0 = Timer2 的值不等于 PR2
0	-	保留位，不要写 1

#### 4.1.9 TMR2 (Addr:0x11)

表 4-15. TMR2 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMR2	TMR2<7:0>							
Reset	0000 0000							
Type	RW							

## 4.1.10 T2CON (Addr:0x12)

表 4-16. T2CON 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2CON	-	TOUTPS<3:0>				TMR2ON	T2CKPS<1:0>	
Reset	-	0000				0	00	
Type	-	RW				RW	RW	

表 4-17. T2CON 位功能描述

Bit	Name	Function
7	-	保留位，读 0
6:3	TOUTPS<3:0>	Timer2 输出后分频比选择 0000 = 1:1 后分频比 0001 = 1:2 后分频比 0010 = 1:3 后分频比 0011 = 1:4 后分频比 0100 = 1:5 后分频比 0101 = 1:6 后分频比 0110 = 1:7 后分频比 0111 = 1:8 后分频比 1000 = 1:9 后分频比 1001 = 1:10 后分频比 1010 = 1:11 后分频比 1011 = 1:12 后分频比 1100 = 1:13 后分频比 1101 = 1:14 后分频比 1110 = 1:15 后分频比 1111 = 1:16 后分频比
2	TMR2ON	Timer2 开关 1 = Timer2 打开 0 = Timer2 关闭
1:0	T2CKPS<1:0>	Timer2 驱动时钟分频比选择 00 = 1:1 分频比； 01 = 1:4 分频比； 1x = 1:16 分频比

## 4.1.11 WDTCON (Addr:0x18)

表 4-18. WDTCON 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDTCON	-	-	-	WDTPS<3:0>				SWDTEN
Reset	-	-	-	0	1	0	0	0
Type	-	-	-	RW	RW	RW	RW	RW

表 4-19. WDTCON 位功能描述

Bit	Name	Function
7:5	-	保留位, 读 0
4:1	WDTPS<3:0>	看门狗定时器周期选择 0000 = 1:32 0001 = 1:64 0010 = 1:128 0011 = 1:256 0100 = 1:512 (复位值) 0101 = 1:1024 0110 = 1:2048 0111 = 1:4096 1000 = 1:8192 1001 = 1:16384 1010 = 1:32768 1011 = 1:65536 11xx = 1:65536
0	SWDTEN	软件开关看门狗定时器 1 = 打开 0 = 关闭

## 4.1.12 CMCON0 (Addr:0x19)

表 4-20. CMCON0 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMCON0	C2OUT	C1OUT	C2INV	C1INV	CIS	CM<2:0>		
Reset	0	0	0	0	1	0	0	0
Type	R	R	RW	RW	RW	RW	RW	RW

表 4-21. CMCON0 位功能描述

Bit	Name	Function
7	C2OUT	比较器 2 输出位 当 C2INV=0 时, 1: C2VIN+ > C2VIN- 0: C2VIN+ < C2VIN- 当 C2INV=1 时, 1: C2VIN+ < C2VIN- 0: C2VIN+ > C2VIN-

6	C1OUT	比较器 1 输出位 当 C1INV=0 时, 1: C1VIN+ > C1VIN- 0: C1VIN+ < C1VIN- 当 C1INV=1 时, 1: C1VIN+ < C1VIN- 0: C1VIN+ > C1VIN-
5	C2INV	比较器 2 输出反向控制位 0 = 不反向 1 = 反向
4	C1INV	比较器 1 输出反向控制位 0 = 不反向 1 = 反向
3	CIS	比较器输入切换 当 CM[2:0]=010 时, 1 = C1IN+接 C1VIN+, C2IN+接 C2VIN+ 0 = C1IN-接 C1VIN-, C2IN-接 C2VIN- 当 CM[2:0]=001 时, 1 = C1IN+接 C1VIN+ 0 = C1IN-接 C1VIN-
2:0	CM<2:0>	比较器模式选择 000 = 比较器关闭, CxIN 管脚为模拟 IO 管脚 001 = 3 个输入共用到 2 个比较器上 010 = 4 个输入共用到 2 个比较器上 011 = 2 个共参考比较器 100 = 2 个独立比较器 101 = 1 个独立比较器 110 = 2 个带输出共参考比较器 111 = 比较器关闭, CxIN 管脚为数字 IO 管脚

#### 4.1.13 PR0 (Addr:0x1A)

表 4-22. PR0 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PR0	PR0<7:0>							
Reset	0xFF							
Type	RW							

表 4-23. PR0 功能描述

Bit	Name	Function
7:0	PR0<7:0>	Timer0 周期 (比较) 寄存器

## 4.1.14 MSCKCON (Addr:0x1B)

表 4-24. MSCKCON 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MSCKCON	-	-	-	SLVREN	-	CKMAVG	CKCNTI	-
Reset	-	-	-	0	-	0	0	-
Type	-	-	-	RW	-	RW	RW	-

表 4-25. MSCKCON 位功能描述

Bit	Name	Function
7:5	-	保留位, 不能写 1
4	SLVREN	软件控制 LVR 使能位 1 = 当 UCFG<1:0>为 00 时, 打开 LVR 0 = 无论 UCFG<1:0>为何值, 禁止 LVR
3	-	保留位, 不能写 1
2	CKMAVG	快时钟测量慢时钟周期的测量平均模式 1 = 打开平均模式 (自动测量并累加 4 次) 0 = 关闭平均模式
1	CKCNTI	使能快时钟测量慢时钟周期 1 = 使能快时钟测量慢时钟周期 0 = 关闭快时钟测量慢时钟周期 注: 该位在测量完毕后会自自动归零
0	-	保留位, 不能写 1

## 4.1.15 SOSCPR (Addr:0x1C/0x1D)

表 4-26. SOSCPR 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SOSCPR	SOSCPR<7:0>							
Reset	0xFF							
Type	RW							

表 4-27. SOSCPRH 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SOSCPRH	-				SOSCPR<11:8>			
Reset	-				1111			
Type	-				RW			

表 4-28. SOSCPR 功能描述

Bit	Name	Function
11:0	SOSCPR<11:0>	低速振荡器周期 (单位: 快时钟周期数) 用于慢时钟测量功能

## 4.1.16 OPTION (Addr:0x81)

表 4-29. OPTION 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPTION	/PAPU	INTEDG	T0CS	T0SE	PSA	PS<2:0>		
Reset	1	1	1	1	1	111		
Type	RW	RW	RW	RW	RW	RW		

表 4-30. OPTION 位功能描述

Bit	Name	Function																											
7	/PAPU	PORTA 上拉控制位 1 = PORTA 所有端口上拉关闭 0 = PORTA 上拉总使能, 可以通过寄存器独立开关各端口																											
6	INTEDG	外部中断边沿触发选择位 1 = PA2/INT 选择上升沿触发 0 = PA2/INT 选择下降沿触发																											
5	T0CS	Timer0 时钟源选择位 1 = 来自于 PA2/T0CKI 0 = 内部时钟源 (FOSC/2)																											
4	T0SE	Timer0 时钟源边沿选择 1 = PA2/T0CKI 下降沿触发计数 0 = PA2/T0CKI 上升沿触发计数																											
3	PSA	分频器用途配置 1 = 分频器配置给 WDT 用 0 = 分频器配置给 Timer0 用																											
2:0	PS<2:0>	分频器分频比选择 <table border="1"> <thead> <tr> <th>值</th><th>Timer0</th><th>WDT</th></tr> </thead> <tbody> <tr><td>000</td><td>1:2</td><td>1:1</td></tr> <tr><td>001</td><td>1:4</td><td>1:2</td></tr> <tr><td>010</td><td>1:8</td><td>1:4</td></tr> <tr><td>011</td><td>1:16</td><td>1:8</td></tr> <tr><td>100</td><td>1:32</td><td>1:16</td></tr> <tr><td>101</td><td>1:64</td><td>1:32</td></tr> <tr><td>110</td><td>1:128</td><td>1:64</td></tr> <tr><td>111</td><td>1:256</td><td>1:128</td></tr> </tbody> </table>	值	Timer0	WDT	000	1:2	1:1	001	1:4	1:2	010	1:8	1:4	011	1:16	1:8	100	1:32	1:16	101	1:64	1:32	110	1:128	1:64	111	1:256	1:128
值	Timer0	WDT																											
000	1:2	1:1																											
001	1:4	1:2																											
010	1:8	1:4																											
011	1:16	1:8																											
100	1:32	1:16																											
101	1:64	1:32																											
110	1:128	1:64																											
111	1:256	1:128																											

## 4.1.17 TRISA (Addr:0x85)

表 4-31. TRISA 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	R	RW	RW	RW	RW	RW



表 4-32. TRISA 位功能描述

Bit	Name	Function
7:6	TRISA<7:6>	PORTA<7:6>端口方向控制 1 = 输入 0 = 输出
5	TRISA<5>	PORTA5 端口方向控制 仅为输入，固定为 1
4:0	TRISA<4:0>	PORTA<4:0>端口方向控制 1 = 输入 0 = 输出

## 4.1.18 TRISC (Addr:0x87)

表 4-33. TRISC 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	R	RW	RW	RW	RW	RW

表 4-34. TRISC 位功能描述

Bit	Name	Function
7:0	TRISC<7:0>	PORTC<7:0>端口方向控制 1 = 输入 0 = 输出

## 4.1.19 PIE1 (Addr:0x8C)

表 4-35. PIE1 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PIE1	EEIE	CKMEAIE	-	C2IE	C1IE	OSFIE	TMR2IE	-
Reset	0	0	-	0	0	0	0	-
Type	RW	RW	-	RW	RW	RW	RW	-

表 4-36. PIE1 位功能描述

Bit	Name	Function
7	EEIE	EEPROM 写中断使能 1 = 使能 EEPROM 写操作完成中断 0 = 关闭 EEPROM 写操作完成中断
6	CKMEAIE	快时钟测量慢时钟操作完成中断使能位 1 = 使能快时钟测量慢时钟操作完成中断 0 = 关闭快时钟测量慢时钟操作完成中断

4	C2IE	比较器 2 中断使能位 1 = 使能比较器 2 中断 0 = 禁止比较器 2 中断
3	C1IE	比较器 1 中断使能位 1 = 使能比较器 1 中断 0 = 禁止比较器 1 中断
2	OSFIE	振荡器故障中断使能位 1 = 使能振荡器故障中断 0 = 禁止振荡器故障中断
1	TMR2IE	Timer2 与 PR2 比较相等中断使能位 1 = 使能 0 = 禁止

#### 4.1.20 PCON (Addr:0x8E)

表 4-37. PCON 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	-	-	-	-	-	-	/POR	/BOR
Reset	-	-	-	-	-	-	q	q
Type	-	-	-	-	-	-	RW	RW

表 4-38. PCON 位功能描述

Bit	Name	Function
1	/POR	上电复位标志，低有效 0 = 发生上电复位 1 = 没发生上电复位或由软件置 1 /POR 在上电复位后值为 0，此后软件应该将其置 1
0	/BOR	低电压复位标志，低有效 0 = 发生了低电压复位 1 = 没有发生低电压复位或由软件置 1

#### 4.1.21 OSCCON (Addr:0x8F)

表 4-39. OSCCON 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSCCON	LFMOD	IRCF<2:0>			OSTS	HTS	LTS	SCS
Reset	0	101			1	0	0	0
Type	RW	RW			R	R	R	RW

表 4-40. OSCCON 位功能描述

Bit	Name	Function
7	LFMOD	低频内振模式: 1 = 256K 振荡频率模式 0 = 32K 振荡频率模式
6:4	IRCF<2:0>	内部振荡器频率选择 111 = 16MHz 110 = 8MHz 101 = 4MHz(默认) 100 = 2MHz 011 = 1MHz 010 = 500KHz 001 = 250KHz 000 = 32KHz (LFINTOSC)
3	OSTS	振荡器起振超时状态位 1 = 器件运行在 FOSC<2:0>指定的外部时钟之下 0 = 器件运行在内部振荡器之下
2	HTS	内部高速时钟状态 1 = HFINTOSC 状态稳定 0 = HFINTOSC 状态未稳定
1	LTS	内部低速时钟状态 1 = LFINTOSC 状态稳定 0 = LFINTOSC 状态未稳定
0	SCS	系统时钟选择位 1 = 系统时钟选择为内部振荡器 0 = 时钟源由 FOSC<2:0>决定

## 4.1.22 PR2 (Addr:0x92)

表 4-41. PR2 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PR2	PR2<7:0>							
Reset	0xFF							
Type	RW							

表 4-42. PR2 功能描述

Bit	Name	Function
7:0	PR2<7:0>	Timer2 周期 (比较) 寄存器 (详见 Timer2 描述章节)

## 4.1.23 WPUA (Addr:0x95)

表 4-43. WPUA 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WPUA	WPUA7	WPUA6	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0
Reset	1	1	-	1	1	1	1	1
Type	RW	RW	-	RW	RW	RW	RW	RW

表 4-44. WPUA 功能描述

Bit	Name	Function
7:6	WPUA<7:6>	PORTA 弱上拉使能 1 = 使能 0 = 禁止
4:0	WPUA<4:0>	PORTA 弱上拉使能 1 = 使能 0 = 禁止

## 4.1.24 IOCA (Addr:0x96)

表 4-45. IOCA 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOCA	IOCA<7:0>							
Reset	0x00							
Type	RW							

表 4-46. IOCA 功能描述

Bit	Name	Function
7:0	IOCA<7:0>	PORTA 端口状态触发中断使能控制位 1 = 使能端口状态触发中断 0 = 关闭端口状态触发中断

## 4.1.25 VRCON (Addr:0x99)

表 4-47. VRCON 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
VRCON	VREN	-	VRR	-	VR<3:0>			
Reset	0	-	0	-	0			
Type	RW	-	RW	-	RW			

表 4-48. VRCON 位功能描述

Bit	Name	Function
7	VREN	CVref 使能位 1 = CVref 电路通电工作 0 = CVref 电路断电，无泄漏电流
5	VRR	CVref 范围选择位 1 = 低电平范围 0 = 高电平范围
3:0	VR<4:0>	CVref 值选择控制位： VRR = 1 时， $CVref = (VR<4:0> \div 24) \times VDD$ VRR = 0 时， $CVref = (VDD \div 4) + (VR<4:0> \div 32) \times VDD$

## 4.1.26 EEDAT (Addr:0x9A)

表 4-49. EEDAT 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EEDAT	EEDAT<7:0>							
Reset	0x00							
Type	RW							

## 4.1.27 EEADR (Addr:0x9B)

表 4-50. EEADR 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EEADR	EEADR<7:0>							
Reset	0x00							
Type	RW							

## 4.1.28 EECON1 (Addr:0x9C)

表 4-51. EECON1 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EECON1	-	-	WREN3	WREN2	WRERR	WREN1	-	RD
Reset	-	-	0	0	X	0	-	0
Type	-	-	RW	RW	RW	RW	-	W

表 4-52. EECON1 位功能描述

Bit	Name	Function
5,4,2	WREN<2:0>	数据 EEPROM 写使能 111 = 允许软件对 EEPROM 编程，编程完成后各位自动归 0 其它值 = 禁止软件对 EEPROM 编程

3	WRERR	数据 EEPROM 写错误标志位 1 = 在 EEPROM 编程周期发生了看门狗或者外部复位导致中止 0 = 在 EEPROM 编程周期正常完成
0	RD	数据 EEPROM 读控制位，此位只写，读永远返回 0 1 = 启动一次数据 EEPROM 读周期 0 = 不启动读

#### 4.1.29 EECON2 (Addr:0x9D)

表 4-53. EECON2 寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EECON2	-	-	-	-	-	-	-	WR
Reset	-	-	-	-	-	-	-	0
Type	-	-	-	-	-	-	-	RW

表 4-54. EECON2 位功能描述

Bit	Name	Function
0	WR	数据 EEPROM 写控制位 读操作，1 = 数据 EEPROM 编程周期进行中 0 = 数据 EEPROM 不处于编程周期 写操作，1 = 启动一次数据 EEPROM 编程周期 0 = 无功能

#### 4.1.30 配置寄存器 UCFGx

软件不能访问 UCFG0、UCFG1、UCFG2，它们只在上电过程由硬件写入（烧录）。

##### ● UCFG0, PROM 地址 0x2000

表 4-55. UCFG0 配置寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UCFG0	-	CPB	MCLRE	PWRTEB	WDTE	FOSC<2:0>		

表 4-56. UCFG0 位功能描述

Bit	Name	Function
6	CPB	1 = Flash 内容不保护 0 = 启动 Flash 内容保护，MCU 能读，串口不能读 注意： 该位只能由 1 改写为 0，而不能由 0 改写为 1。由 0 改写成 1 的唯一方法是进行一次包括 USER_OPT 在内的片擦除操作，并且重新上电后 CPB 才变为 1。
5	MCLRE	1 = PA5/MCLR 脚执行 MCLR 功能，是复位脚 0 = PA5/MCLR 脚执行 PA5 功能，是数字输入引脚
4	PWRTEB	1 = PWRT 禁止 0 = PWRT 使能

3	WDTE	1 = WDT 使能，程序不能禁止 0 = WDT 禁止，但程序可通过设置 WDTCON 的 SWDTEN 位将 WDT 使能
2:0	FOSC<2:0>	000 = 32K 晶振模式，PA6/PA7 接低速晶体； 001 = 20MHz 晶体模式，PA6/PA7 接高速晶体； 010 = 外部时钟模式，PA6 为 IO 功能，PA7 接时钟输入 011 = INTOSC 模式，PA6 输出系统时钟的 2 分频，PA 为 IO 引脚； 1xx = INTOSCIO 模式，PA6 和 PA7 均为 IO 引脚

● UCFG1, PROM 地址 0x2001

表 4-57. UCFG1 配置寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UCFG1	-	-	TSEL	FCMEN	IESO	RD_CTRL	LVDEN<1:0>	

表 4-58. UCFG1 位功能描述

Bit	Name	Function
5	TSEL	指令周期选择位 1 = 指令周期为 2T 0 = 指令周期为 4T
4	FCMEN	时钟故障监视使能 1 = 使能时钟故障监视 0 = 禁止时钟故障监视
3	IESO	双速时钟使能 1 = 使能 0 = 禁止
2	RD_CTRL	输出模式时读端口控制 1 = 读数据端口返回的 PAD 上的值 0 = 读数据端口返回的 Latch 上的值
1:0	LVDEN<1:0>	低电压复位选择 00 = 低电压复位使能 其它 = 禁止低电压复位

● UCFG2, PROM 地址 0x2002

表 4-57. UCFG12 置寄存器

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UCFG2	-	-	-	-	LVDS<3:0>			

表 4-58. UCFG2 位功能描述

Bit	Name	Function
7:4	-	保留位

3:0	LVDS[3:0]	低电压复位阈值选择	
		数值	电压
		0010	1.8V
		0011	2.0V
		0100	2.2V
		0110	2.8V
		其它	保留

4.1.31 PCL 和 PCLATH

程序计数器（PC）为 11 位宽，其低 8 位来自可读写的 PCL 寄存器，高 3 位（PC<10:8>）来自 PCLATH，不能直接读写。只要发生复位，PC 将被清 0。下图显示了装载 PC 值的两种情形。注意图右边的 LCALL 和 LJUMP 指令，由于指令中的操作码为 11 位，而芯片的 PC 刚好是 11 位，所以这时 PCLATH 并不需要用到。

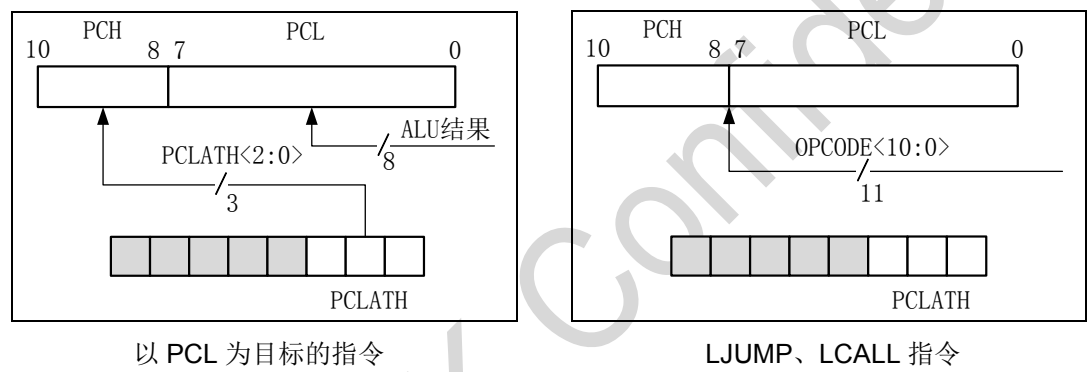


图 4-1. 不同情况下 PC 的加载示意图

修改 PCL

执行任何以 PCL 寄存器为目标寄存器的指令将同时使程序计数器 PC<10:8>被 PCLATH 内容取代。这样可通过将所需的高 3 位写入 PCLATH 寄存器来改变程序计数器的所有内容。

计算 LJUMP 指令是通过向程序计数器加入偏移量（ADDWR PCL）来实现的。通过修改 PCL 寄存器跳转到查找表或程序分支表（计算 LJUMP）时应特别谨慎。假定 PCLATH 设置为表的起始地址，如果表长度大于 255 条指令，或如果存储器地址的低 8 位在表的中间从 0xFF 计满返回到 0x00，那么在每次表起始地址与表内的目标地址之间发生计满返回时，PCLATH 必须递增。

4.1.32 INDF 和 FSR 寄存器

INDF 不是物理存在的寄存器，对 INDF 进行寻址将产生间接寻址，可寻址范围为 0~255。任何使用 INDF 寄存器的指令，实际上是对文件选择寄存器 FSR 所指向的单元进行存取。间接对 INDF 进行读操作将返回 0。间接对 INDF 进行写将导致控操作（可能会影响状态标志位）。



## 5 系统时钟源

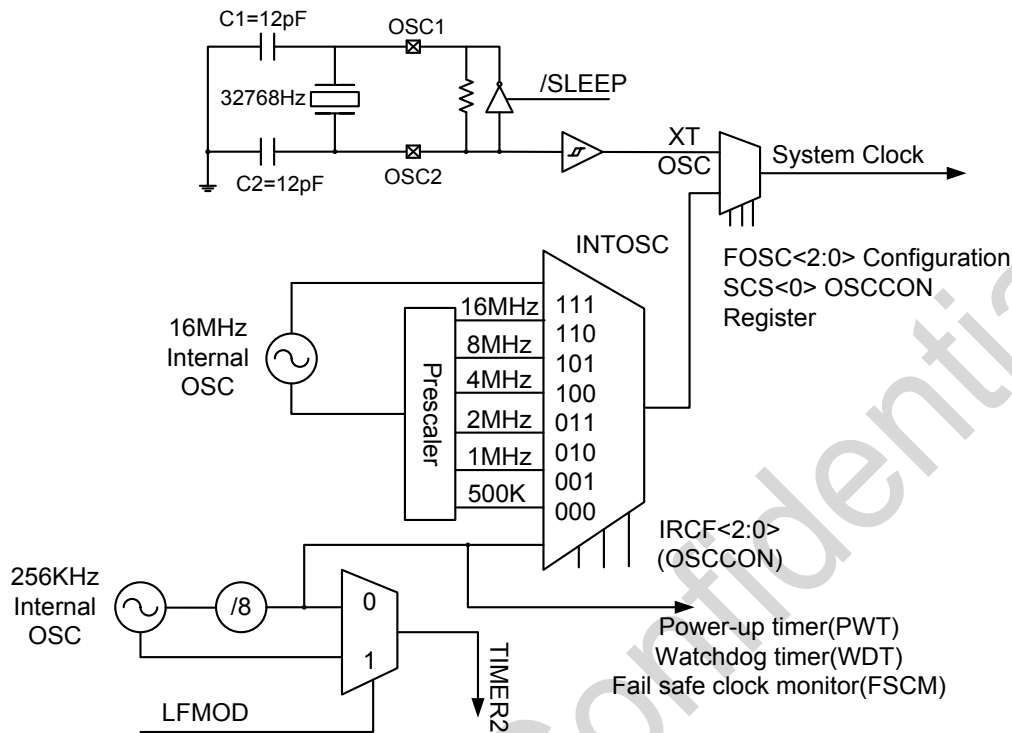


图 5-1. 系统时钟源框图

本芯片包含 3 个时钟源：2 个内置振荡器作为各种时钟源，1 个外部时钟输入源。内置振荡器包括：1 个内部 16M 高速精准振荡器（HFINTOSC），1 个内部 32K/256K（LFINTOSC）低速低功耗振荡器。这些时钟或振荡器结合预分频器可以给系统提供各种频率的时钟源。系统时钟源的预分频器比例由 OPTION 寄存器里的 IRCF<2:0>位控制。

注意：

看门狗、系统时钟源（IRCF=000）以及 PWRT 统一使用 8 分频之后的输出，即 32KHz，而不管 LFMOD 为何值。

### 5.1 时钟源模式

时钟源模式分为外部和内部模式。

- 外部时钟模式依靠外部电路提供时钟源，比如外部时钟 EC 模式，晶体谐振器 XT、LP 模式
- 内部时钟模式内置于振荡器模块中，振荡器模块有 16MHz 高频振荡器和 32KHz 低频振荡器

可通过 OSCCON 寄存器的系统时钟选择位（SCS）来选择内部或者外部时钟源。

## 5.2 外部时钟模式

### 5.2.1 EC 模式

外部时钟模式允许外部产生的逻辑电平作为系统时钟源。工作在此模式下时，外部时钟源连接到 OSC1 输入。

当选取 EC 模式时，振荡器起振定时器（OST）被禁止。因此，上电复位（POR）后或者从休眠中唤醒后的操作不存在延时。MCU 被唤醒后再次启动外部时钟，器件恢复工作，就好像没有停止过一样。

### 5.2.2 LP 和 XT 模式

LP 和 XT 模式支持连接到 OSC1 和 OSC2 的石英晶体谐振器或陶瓷谐振器的使用。模式选择内容反相放大器的低或高增益设定，以支持各种谐振器类型及速度。

LP 振荡器模式选择内部反相放大器的最低增益设定。LP 模式的电流消耗在两种模式中最小。该模式设计仅用于驱动 32.768KHz 音叉式晶振（钟表晶振）。

XT 振荡器模式选择内部反相放大器的高增益设定。

## 5.3 内部时钟模式

振荡器模式有两个独立的内部振荡器，可配置或选取为系统时钟源。

1. HFINTOSC（高频内部振荡器）出厂时已校准，工作频率为 16MHz。
2. LFINTOSC（低频内部振荡器）未经校准，工作频率为 32KHz。软件对 OSCCON 寄存器的内部振荡器频率选择位 IRCF<2:0>进行操作，可选择系统时钟速度。

可通过 OSCCON 寄存器的系统时钟选择（SCS）位，在外部或内部时钟源之间选择系统时钟。

注意：

OSCCON 寄存器的 LFMOD 可以选择 LFINTOSC 是 32KHz 或者 256KHz，但看门狗固定使用 32KHz，不管 LFMOD 为何值。

### 5.3.1 频率选择位（IRCF）

16MHz HFINTOSC 和 32KHz LFINTOSC 的输出连接到预分频器和多路复用器（见图 xx）。OSCCON 寄存器的内部振荡器频率选择位 IRCF<2:0>用于选择内部振荡器的频率输出。可通过软件选择以下 8 个频率之一：

- 16MHz
- 8MHz
- 4MHz（复位后的缺省值）
- 2MHz

- 1MHz
- 500KHz
- 250KHz
- 32KHz

5.3.2 HFINTOSC 和 LFINTOSC 时钟切换时序

当在 LFINTOSC 和 HFINTOSC 之间切换时，新的振荡器可能为了省电已经关闭（见 XX）。在这种情况下，OSCCON 寄存器的 IRCF 位被修改之后、频率选择生效之前，存在一个延时。OSCCON 寄存器的 LTS 和 HTS 位将反映 LFINTOSC 和 HFINTOSC 振荡器的当前活动状态。频率选择时序如下：

1. OSCCON 寄存器的 IRCF<2:0>位被修改
2. 如果新时钟是关闭的，开始一个时钟启动延时
3. 时钟切换电路等待当前时钟下降沿的到来
4. CLKOUT 保持为低，时钟切换电路等待两个新时钟下降沿的到来
5. 现在 CLKOUT 连接到新时钟，OSCCON 寄存器的 HTS 和 LTS 位按要求被更新
6. 时钟切换完成

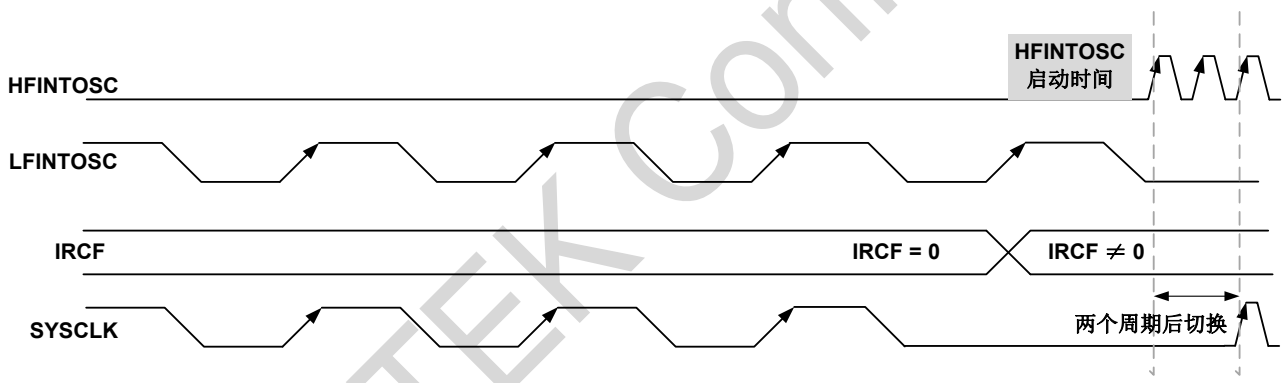


图 5-2. 由慢时钟切换到快时钟示意图

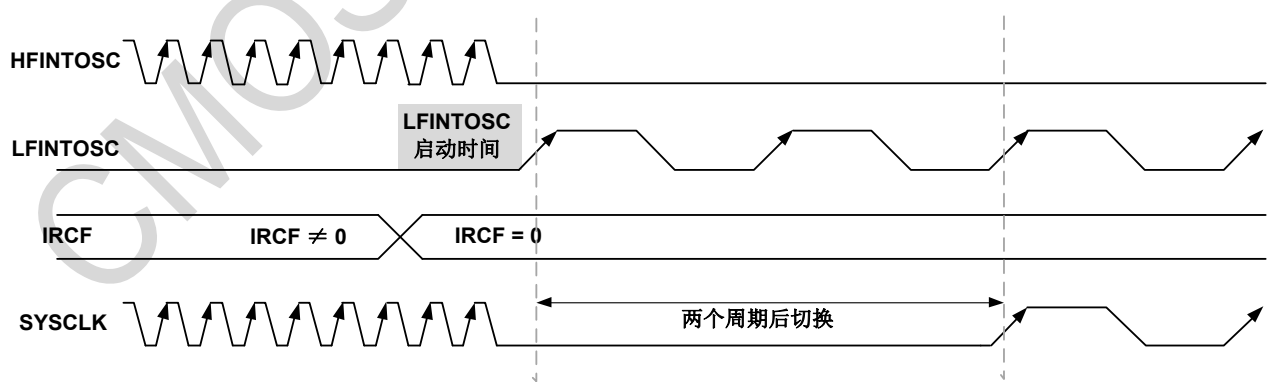


图 5-3. 由快时钟切换到慢时钟示意图

## 5.4 时钟切换

通过软件对 OSCCON 寄存器的系统时钟选择 (SCS) 位进行操作, 可将系统时钟源在外部和内部时钟源之间切换。

### 5.4.1 系统时钟选择 (SCS) 位

OSCCON 寄存器的系统时钟选择 (SCS) 位选择用于 CPU 和外设的系统时钟源。

- OSCCON 寄存器的位 SCS=0 时, 系统时钟源由配置字寄存器 (UCFG0) 中 FOSC<2:0>位的配置决定
- OSCCON 寄存器的位 SCS=1 时, 根据 OSCCON 寄存器的 IRCF<2:0>位所选的内部振荡器频率选取系统时钟源。复位后, OSCCON 寄存器的 SCS 总是被清零。

注意:

任何由硬件引起的时钟切换 (可能产生自双速启动或故障保护时钟监控器) 都不会更新 OSCCON 寄存器的 SCS 位。用户应该监控 OSCCON 寄存器的 OSTS 位以确定当前的系统时钟源。

### 5.4.2 振荡器起振超时状态 (OSTS) 位

OSCCON 寄存器的振荡器起振超时状态 (OSTS) 位用于指示系统时钟是来自外部时钟源, 还是来自内部时钟源。外部时钟源由配置字寄存器 (UCFG0) 的 FOSC<2:0>定义。OSTS 还特别指明在 LP 或 XT 模式下, 振荡器起振定时器 (OST) 是否已超时。

## 5.5 双速时钟启动模式

双速启动模式通过最大限度地缩短外部振荡器起振与代码执行之间的延时, 进一步节省了功耗。对于频繁使用休眠模式的应用, 双速启动模式将在器件唤醒后除去外部振荡器的起振时间, 从而可降低器件的总体功耗。该模式使得应用能够从休眠中唤醒, 将 INTOSC 用作时钟源执行数条指令, 然后再返回休眠状态而无需等待主振荡器的稳定。

注意:

执行 SLEEP 指令将中止振荡器起振时间, 并使 OSCCON 寄存器的 OSTS 位保持清零。

当振荡器模块配置为 LP 或 XT 模式时, 振荡器起振定时器 (OST) 使能 (见第 7.2.1 节“振荡器起振定时器”)。OST 将暂停程序执行, 直到完成 1024 次振荡计数。双速启动模式在 OST 计数时使用内部振荡器进行工作, 使代码执行的延时最大限度地缩短。当 OST 计数到 1024 且 OSCCON 寄存器的 OSTS 位置 1 时, 程序执行切换至外部振荡器。

### 5.5.1 双速启动模式配置

通过以下设定来配置双速启动模式:

- 配置字寄存器（UCFG1）中的位 IESO=1；内部 / 外部切换位（使能双速启动模式）。
- OSCCON 寄存器的位 SCS=0
- 配置字寄存器（CONFIG）中的  $F_{OSC}<2:0>$  配置为 LP 或 XT 模式

在下列操作之后，进入双速启动模式：

- 上电复位（POR）且上电延时定时器（PWRT）
- 延时结束（使能时）后，或者从休眠状态唤醒

如果外部时钟振荡器配置为除 LP 或 XT 模式以外的任一模式，那么双速启动将被禁止。这是因为 POR 后或从休眠中退出时，外部时钟振荡不需要稳定时间。

### 5.5.2 双速启动顺序

1. 从上电复位或休眠中唤醒
2. 使用内部振荡器以 OSCCON 寄存器的 IRCF<2:0>位设置的频率开始执行指令
3. OST 指令，计数 1024 个时钟周期
4. OST 超时，等待内部振荡器下降沿的到来
5. OSTS 置 1
6. 系统时钟保持为低，直到新时钟下一个下降沿的到来（LP 或 XT 模式）
7. 系统时钟切换到外部时钟源

## 5.6 故障保护时钟监控器

故障保护时钟监控器（FSCM）使得器件在出现外部振荡器故障时仍能继续工作。FSCM 能在振荡器起振延时定时器（OST）到期后的任一时刻检测振荡器故障。FSCM 通过将配置字寄存器（UCFG1）中的 FCMEN 位置 1 来使能。FSCM 可用于所有外部振荡器模式（LP、XT 和 EC）。

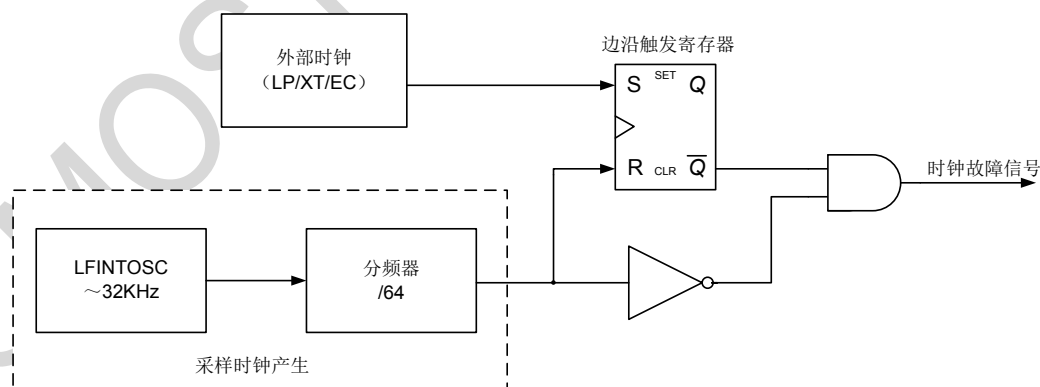


图 5-4. FSCM 原理图框图

### 5.6.1 故障保护检测

FSCM 模块通过将外部振荡器与 FSCM 采样时钟比较来检测振荡器故障。LFINTOSC 除以 64，就产生了采样时钟。请参见图 5-4。故障检测器内部有一个锁存器。在外部时钟的每个下降沿，锁存器被置 1。在采

样时钟的每个上升沿，锁存器被清零。如果采样时钟的整个半周期流逝而主时钟依然未进入低电平，就检测到故障。

### 5.6.2 故障保护操作

当外部时钟出现故障时，FSCM 将器件时钟切换到内部时钟源，并将 PIR1 寄存器的 OSFIF 标志位置 1。如果在 PIR1 寄存器的 OSFIE 位置 1 的同时将该标志位置 1，将产生中断。器件固件随后会采取措施减轻可能由故障时钟所产生的问题。系统时钟将继续来自内部时钟源，直到器件固件成功重启外部振荡器并切换回外部操作。FSCM 所选的内部时钟源由 OSCCON 寄存器的 IRCF<2:0>位决定。这使内部振荡器可以在故障发生前就得以配置。

### 5.6.3 故障保护条件清除

复位、执行 SLEEP 指令或翻转 OSCCON 寄存器的 SCS 位后，故障保护条件被清除。OSCCON 寄存器的 SCS 位被修改后，OST 将重新启动。OST 运行时，器件继续从 OSCCON 中选定的 INTOSC 进行操作。OST 超时后，故障保护条件被清除，器件将从外部时钟源进行操作。必须先清除故障保护条件，才能清零 OSFIF 标志位。

### 5.6.4 复位或从休眠中唤醒

FSCM 设计为能在振荡器起振延时定时器（OST）到期后的任一时刻检测振荡器故障。OST 的使用场合为从休眠状态唤醒后以及任何类型的复位后。OST 不能在 EC 时钟模式下使用，所以一旦复位或唤醒完成，FSCM 就处于激活状态。当 FSCM 被使能时，双速启动也被使能。因此，当 OST 运行时，器件总是处于代码执行阶段。

注意：

由于振荡器起振时间的范围变化较大，在振荡器起振期间（从复位或休眠中退出时），故障保护电路不处于激活状态。经过一段适当的时间后，用户应检查 OSCCON 寄存器的 OSTS 位，以验证振荡器是否已成功起振以及系统时钟是否切换成功。

## 6 复位时序

CMT2189B 有以下几种不同的复位：

1. 上电复位 POR
2. WDT 看门狗复位——在常规运行期间
3. WDT 看门狗唤醒——在睡眠期间
4. /MCLR 管脚复位——在常规运行期间
5. /MCLR 管脚复位——在睡眠期间
6. 低点压（BOR/LVR）复位
7. 指令错误复位（可禁止）

有些寄存器是不被任何复位影响的，这些寄存器的状态在上电复位时是未知的，也不受复位事件影响。大多数其它寄存器都会在以下复位事件时恢复到其“复位状态”。

- 上电复位 POR
- WDT 看门狗复位——在常规运行期间
- WDT 看门狗复位——在睡眠期间
- /MCLR 管脚复位——在常规运行期间
- 低电压（BOR）复位
- 错误指令复位

WDT(看门狗)睡眠唤醒不会造成和在常规运行状态下 WDT(看门狗)超时所造成的复位。因为睡眠唤醒本身就是一种继续的意思，而不是复位/TO 和/PD 位的设置和清零在不同复位条件下的动作是不同的。具体可参考表 6-1 和表 6-2。

/MCLRB 管脚背后的电路带有防抖功能，能够滤除一些干扰造成的尖细脉冲信号。下图为复位电路的总体概述框图：

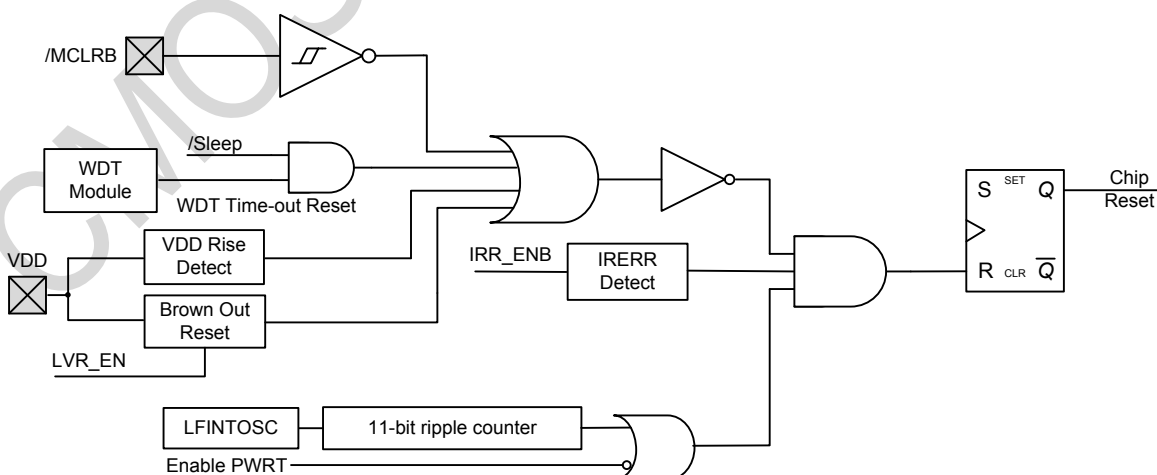


图 6-1. 复位功能框图

## 6.1 POR 上电复位

片上的 POR 电路会将芯片保持在复位状态直到 VDD 电源电压达到足够高，为充分利用片内复位电路的功能，用户可以简单地直接在 VDD 和/MCLR 之间结一个电阻。这样外部就无需任何 RC 复位电路。不过这要求 VDD 电压上升时间为最大。上电完成后，系统复位不会立即释放，还要等一个约 4ms 的延时，期间数字电路保持在复位状态。

## 6.2 外部复位 MCLR

需要注意的是，WDT 复位不会把/MCLR 管脚拉低。在/MCLR 管脚上施加超过指标的电压（例如 ESD 事件）会造成/MCLR 复位，而且在管脚上产生超标的大电流，因此我们推荐用户不再直接用一个电阻将/MCLR 和 VDD 连接起来，而是采用以下电路。

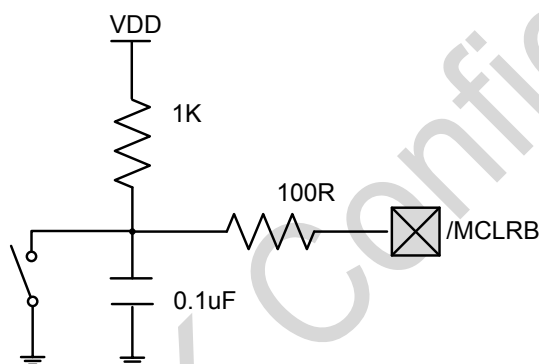


图 6-2. 外部复位参考电路图

在芯片的 CONFIG OPTION 寄存器（UCFG0）中有一个 MCLRE 使能位，将此位清零会使得复位信号由芯片内部产生。当此位为 1 时，芯片的 PA5/MCLR 脚成为外部复位脚。在这个模式下，/MCLR 脚上有个对 VDD 的弱上拉。

## 6.3 PWRT（上电计时器）

PWRT 为上电复位，低电压复位提供一个固定的 64ms（正常情况下）的定时。这个定时器由内部慢时钟驱动。芯片在定时器超时之前都是被保持在复位状态。这段时间能保证 VDD 上升到足够高的电压使得系统能正常工作。PWRT 也可以通过系统 CONFIG 寄存器（UCFG0）来使能。在开启低电压复位功能时，用户应该也打开 PWRT。PWRT 定时是由 VDD 电压超过 VBOR 门限事件启动的。另外需要注意的是，由于由内部慢时钟驱动，定时的实际时间长度是随温度，电压等条件变化而变化的。这个时间不是一个精准参数。

## 6.4 BOR（LVR）低电压复位

低电压复位由 UCFG1<1:0>位来控制。低电压复位就是指当电源电压低于 VBOR 门限电压时所产生的复



位。不过当 VDD 电压低于 VBOR 不超过 TBOR 时间时，低电压复位可能不会发生。VBOR 电压在芯片出货之前需要校准，校准可通过串口写入内部校准寄存器来完成。如果 BOR（低电压复位）是使能（UCFG1<1:0>=00）的，那么最大 VDD 电压上升时间的要求就不存在。BOR 电路会将芯片控制在复位状态，一直到 VDD 电压达到 VBOR 门限电压以上。需要注意的是，当 VDD 低于系统能正常工作的门限时，POR 电路并不会产生复位信号。如果要 BOR 电路产生复位信号，VDD 电压必须在 VSS 电平上保持 100uS 以上。

## 6.5 错误指令复位

当 CPU 的指令寄存器取指到未定义指令时，系统将进行复位，利用此功能可增加系统的抗干扰能力。

## 6.6 超时动作

在上电过程中，芯片内部的超时动作顺序按以下流程执行：POR 结束后启动 PWRT 计时，由于计时是由 POR 脉冲结束启动的，如果/MCLR 在低电平状态下保持足够长的时间，超时事件就会发生。那么将/MCLR 拉高会让 CPU 立即开始执行。这在测试或者需要多个 MCU 同步的情况下会很有用。

### PCON（Power Control Register）

PCON 寄存器里有 2 位指示哪一种复位发生了。Bit0 是/BOR 指示位，其在上电复位是未知态，软件必须将其置 1，然后检查其是否为 0。Bit1 是/POR 指示位，其在上电复位后为 0，软件必须将其置 1。

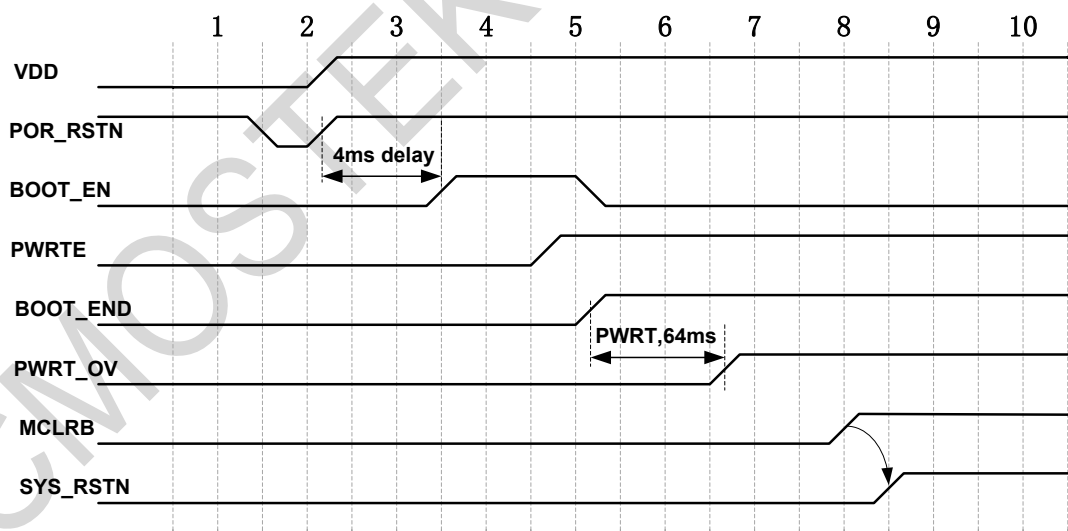
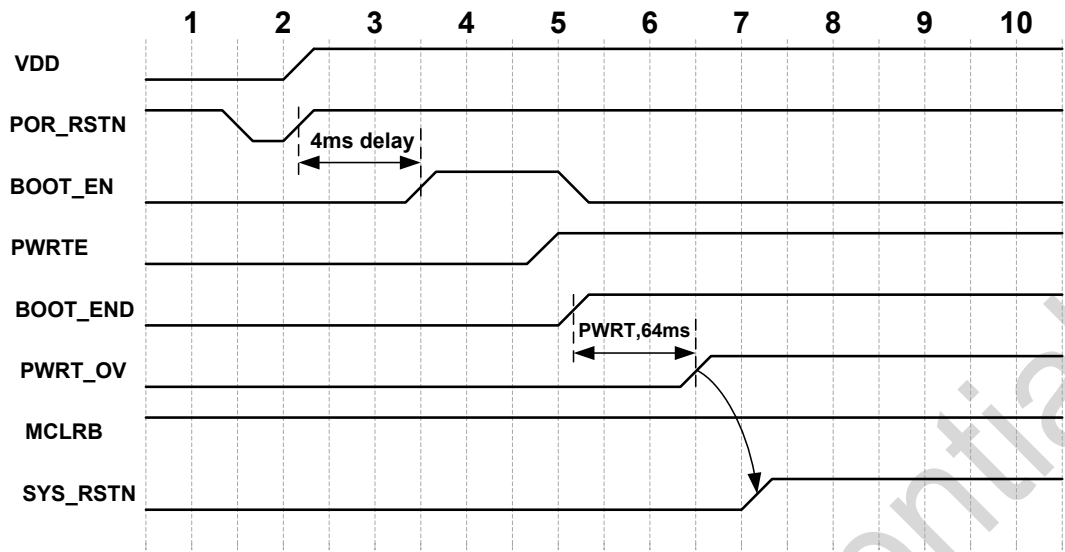


图 6-3. 上电复位，使用了 MCLR



6-4. 上电复位，没有使用 MCLR

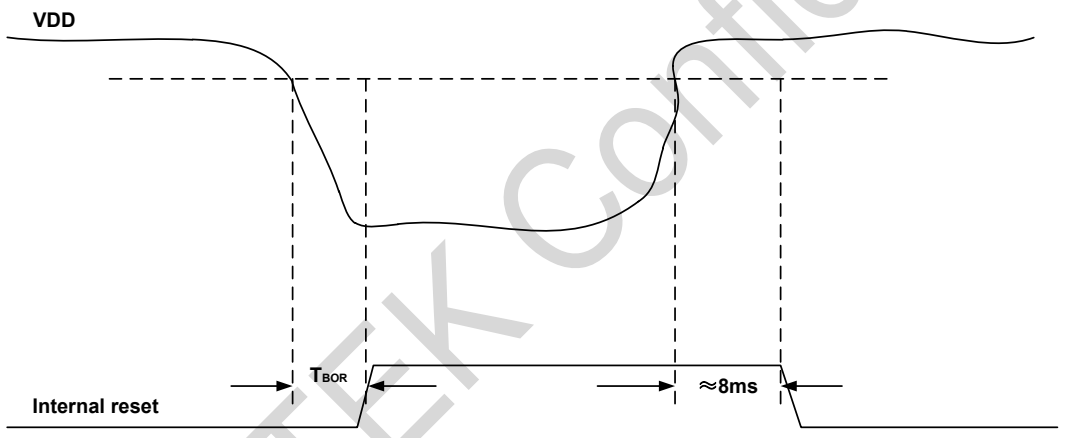


图 6-5. BOR 复位

注意：

- 1. 上电复位或低电压复位后，并且在 PWRT<sub>EB</sub>（UCFG0.4）为低时，PWRT 有效。它是 2048 个内部慢时钟周期，约 64ms；
- 2. TBOR 时间约为 157us；
- 3. 电压恢复正常之后，内部复位不会立即释放，而是要等约为 4ms 的时间。

表 6-1. 各种情况下的超时

振荡器配置	上电复位		低电压复位		睡眠醒来
	/PWRT <sub>EB</sub> =0	/PWRT <sub>EB</sub> =1	/PWRT <sub>EB</sub> =0	/PWRT <sub>EB</sub> =1	
INTOSC	TPWRT	-	TPWRT	-	-

表 6-2. STATUS/PCON 位及其意义 (u-没变化, x-未知)

/POR	/BOR	/TO	/PD	条件
0	X	1	1	POR
U	0	1	1	BOR
U	U	0	U	WDT复位
U	U	0	0	WDT唤醒
U	U	U	U	常规运行下/MCLR复位
U	U	1	0	睡眠下/MCLR复位

## 7 BOOT

在上电复位或低电压复位之后，插入一个状态，把程序 EEPROM 的 2000H 开始的单元映射配置寄存器。系统复位要等 BOOT 结束之后才能释放，如图 6-3 和图 6-4 所示，该过程大概需要 17us。

CMOSTEK Confidential

8 看门狗定时器

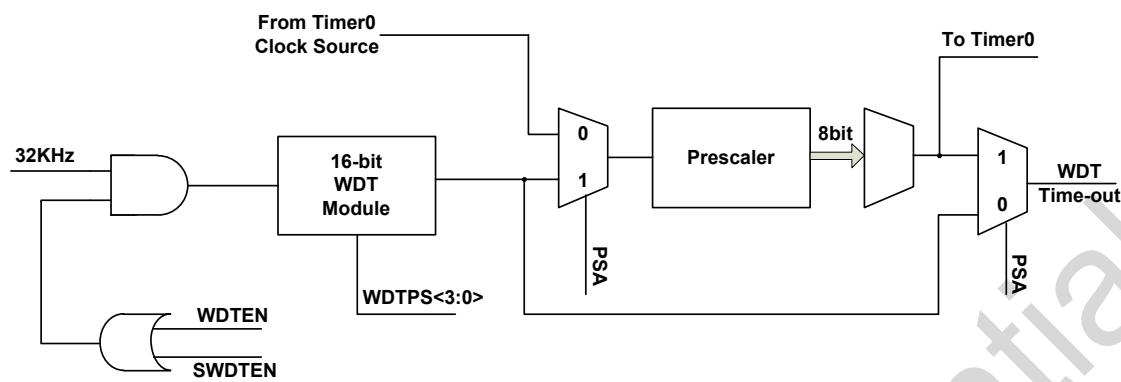


图 8-1. 看门狗和定时器 0 框图

看门狗的时钟源为内部慢时钟（32KHz），它是一个 16 位的计数器，和定时器 0 共用一个 8 位的预分频器，使能位位于配置寄存器 UCFG0 的第 3 位，WDTEN。为 1 时表示使能看门狗，为 0 时禁止，由上电启动过程 BOOT 决定，或者可通过外部串口写入。清看门狗指令 CLRWDT、SLEEP 会清除看门狗计数器。在使能了看门狗的情况下，MCU 睡眠时看门狗溢出事件可以作为一个唤醒源，而 MCU 正常工作时作为一个复位源。

表 8-1. 看门狗状态

条件	看门狗状态
WDTEN和SWDTEN同时为0	清零
CLRWDT指令	
进入SLEEP、退出SLEEP时刻	

注意：

如果内部慢时钟从 32K 切换到 256K 模式（或反之从 256K 切换到 32K 模式），都不影响看门狗计时，因为 WDT 固定使用 32K 时钟源。

## 9 定时器 0

### 9.1 Timer0 简介

定时器 0 为 8 位，可配置为计数器或定时器使用，当作为外部事件（T0CKI）计数器时，可以配置为上升沿或者下降沿计数。作为定时器时，其计数时钟为系统时钟的 2 分频，即每一指令周期递增一次。有一个与 WDT 共用的 8 位预分频器，PSA 为 0 时该预分频器分配给定时器 0 使用。

注意：当改变 PSA 的值时，硬件会自动把预分频器清 0。

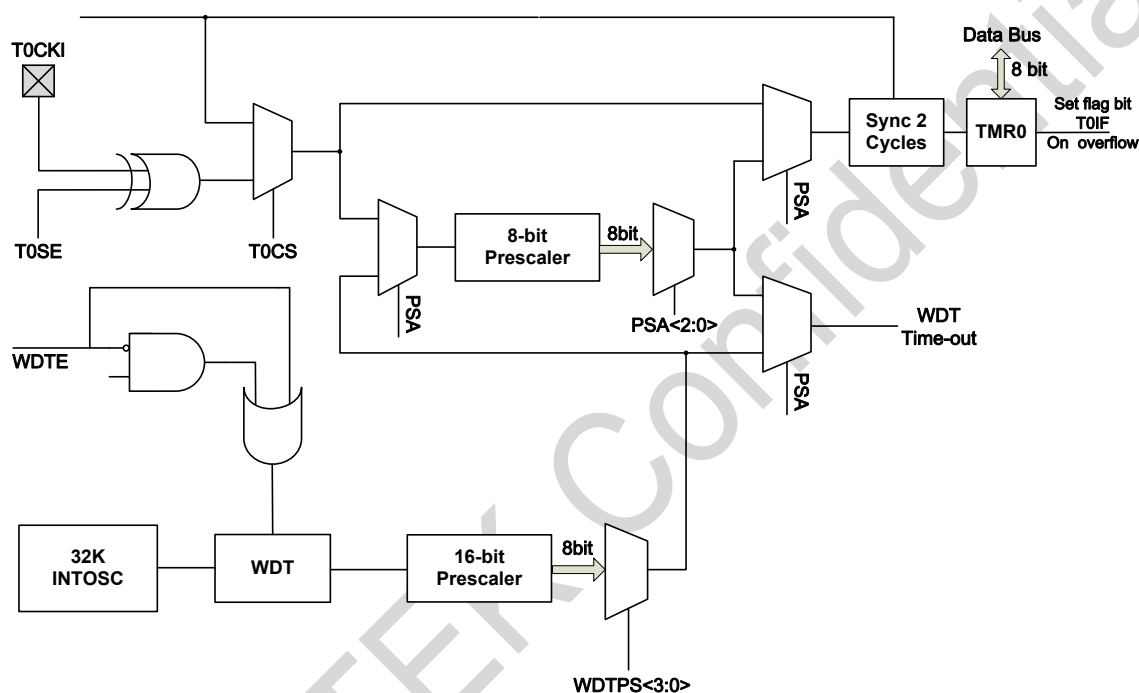


图 9-1. 看门狗和定时器 0 框图

### 9.2 Timer0 定时器模式

该模式下，定时器 0 在每个指令周期加 1(不带预分频)。软件可以清零 OPTION 寄存器里的 T0CS 位以进入定时器模式。当软件对 TMR0 进行写操作时，定时器在写后面 2 个周期内不会递增。

### 9.3 Timer0 计数器模式

该模式下，定时器 0 由每个 T0CKI 管脚的上升沿或下降沿触发加 1（不带预分频）。具体哪一钟沿触发由 OPTION 寄存器里的 T0SE 位决定。软件可以将 OPTION 寄存器里的 T0CS 位置 1 以进入计数器模式。

9.3.1 软件可配置预分频电路

芯片在 Timer0 和 WDT 定时器前面有一个分频电路，可以分配给 Timer0 或者 WDT 定时器用，但二者不能同时使用这个预分频电路。具体分配给 Timer0 还是 WDT 由 OPTION 寄存器里的 PSA 位决定，PSA 为 0 时，预分频分配给 Timer0 使用。在 Timer0 预分频模式下，总共有 8 个预分频比（1:2 到 1:256）可以通过 OPTION 寄存器里的 PS<2:0>位设置。

注意：

- 1. 预分频电路既不可读也不可写。任何对 TMR0 寄存器的写动作会清零预分频电路。
- 2. 当预分频电路分配给 WDT 时，1 条 CLRWDT 指令即可清零预分频电路。
- 3. 在定时器 0 和 WDT 之间切换预分频电路由于分频电路可以分配给 Timer0 或者 WDT 定时器用，在二者之间切换预分频器是有可能导致误复位。

在将预分频电路从分配给 TMR0 切换到分配给 WDT 时，请务必执行以下指令顺序：

```
BANKSEL  TMR0
CLRWDT                                ; Clear WDT
CLRR    TMR0                          ; Clear TMR0 and prescaler
BANKSEL  OPTION_REG
BSR     OPTION_REG, PSA               ; Select WDT
CLRWDT

LDWI    b'11111000'                  ; Mask prescaler bits
ANDWR   OPTION_REG, W
IORWI   b'00000101'                  ; Set WDT prescaler to 1:32
LDWI    OPTION_REG
```

在将预分频电路从分配给 watchdog 切换到分配给 TMR0 时，请务必执行以下指令顺序：

```
CLRWDT                                ; Clear WDT and prescaler
BANKSEL  OPTION_REG
LDWI    b'11110000'                  ; Mask TMR0 select and prescaler bits
ANDWR   OPTION_REG, W
IORWI   b'00000011'                  ; Set prescaler to 1:16
STR     OPTION_REG
```

9.3.2 Timer0 中断

芯片在定时器 0 从 0xFF 溢出到 0x00 时会置起 T0IF 标志，并产生中断（如果使能了的话）。

注意：

Timer0 中断无法唤醒 CPU 因为在睡眠状态下，定时器是被冻结的

### 9.3.3 用外部时钟驱动定时器 0

在计数其模式下，T0CKI 管脚输入和 Timer0 寄存器之间的同步是由在 Q1, Q2 内部时钟相位采样实现的，所以外部时钟源周期的高电平时间和低电平时间必须满足相关时序要求。

CMOSTEK Confidential



## 10 定时器 2

定时器 2 为 8 为定时器包含以下功能：

- 8 位计数寄存器
- 8 位周期寄存器
- TMR2 值等同 PR2 时产生中断
- 1:1, 1:4, 1:16 预分频比
- 1:1~1:16 后分频比

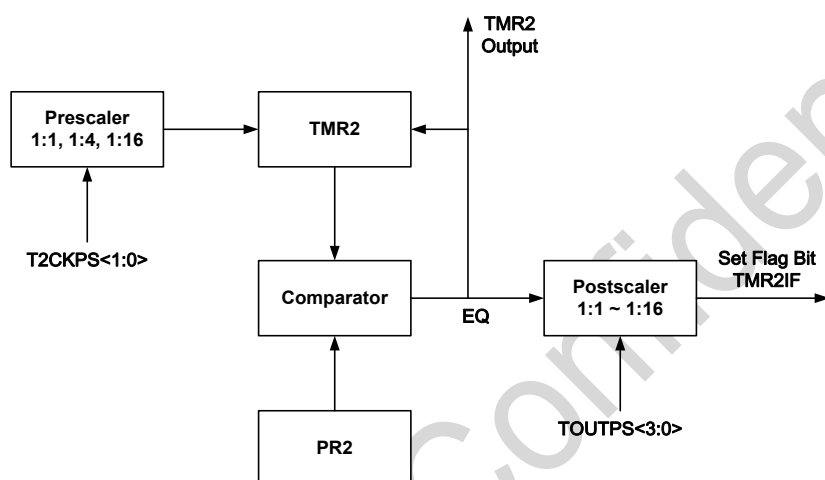


图 10-1. 定时器 2 框图

### Timer2 工作原理：

Timer2 模块的时钟输入是系统指令时钟 ( $F_{OSC}/2$ )。该时钟送入 Timer2 预分频器，其预分频比有 1:1、1:4、1:16 三种选择。随后预分频器的输出被用于递增 TMR2 寄存器。

TMR2 和 PR2 的值被不断比较以确定何时匹配。TMR2 将从 00h 开始递增直到与 PR2 的值相同匹配时将发生以下两种情况：

- TMR2 在下一递增周期复位为 0x00
- Timer2 后分频比递增

Timer2/PR2 比较器的匹配输出送入 Timer2 后分频器。后分频器的选项范围为 1:1 至 1:16。Timer2 后分频器的输出用于将 PIR1 寄存器的 TMR2IF 中断标志置 1

注意：

1. TMR2 和 PR2 都是可读写寄存器。在复位时，他们的值分别是 0 和 0xFF。
2. 将 T2CON 寄存器中的 TMR2ON 位置 1 可打开 Timer2，反之将 TMR2ON 位清零关闭 Timer2。
3. Timer2 预分频器由 T2CON 寄存器的 T2CKPS 位控制。
4. Timer2 后分频器由 T2CON 寄存器的 TOUTPS 位控制。
5. 预分频和后分频计数器会在写以下寄存器时清零：
  - 写 TMR2

- 写 T2CON
- 任何 Reset 动作

6. 写 T2CON 并不会清零 TMR2 寄存器

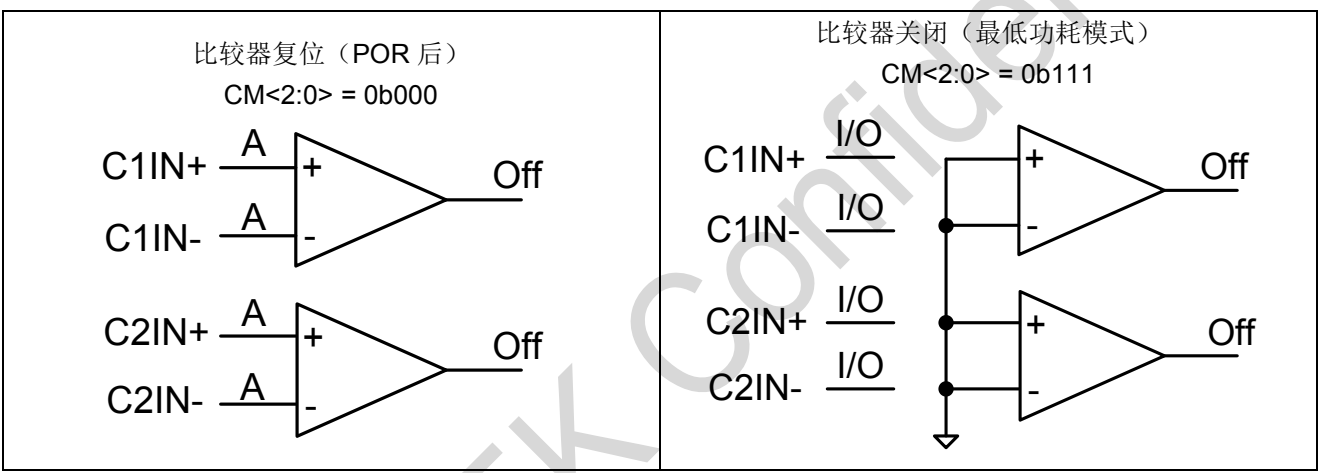
CMOSTEK Confidential

# 11 比较器

片内集成 2 个模拟比较器，由于其中比较器 2 功能管脚同时用于 RF 串行控制总线，所以 2 个比较器均不能使用。

需要注意的是，当读一个端口寄存器时，如果改管脚被配置为模拟信号管脚，软件会读出 0 值。当管脚被设置为数字输入管脚时，比较器仍然会以为该管脚输入一个模拟信号，并输出相应结果。如果一个管脚被设置为数字输入，而这个管脚上面的实际电压又是一个模拟电平，这可能造成输入缓存电路消耗比说明书上标的更大的电流。

模拟比较器总共有 8 种配置模式，由 CMCON0 寄存器的 CM<2:0>这 3 位来选择。由于功能管脚被用于 RF 串行控制线，所以只有如下两种状态：



- 模拟功能（A）：数字输入缓存被屏蔽
- 数字功能（D）：比较器数字输出会覆盖管脚上的其他功能
- 正常端口功能（I/O）：独立于比较器

当端口上标注“A”字样，读的时候无论当前管脚上的状态或 I/O 控制寄存器 TRIS 位的状态都会返回 0 值。用户应该把与被用作模拟输入的管脚相对应的 TRIS 位置为 1 来关闭其数字输出驱动电路。

当端口上标注“D”字样，用户应该将其相应的 TRIS 位职位 0 来打开数字输出驱动电路。

另外，比较器配置切换是应该屏蔽比较器中断以避免不必要的误触发事件。

# 12数据 EEPROM

片内集成有 256 个字节的 EEPROM，通过 EEADR 进行寻址访问。软件可通过 EECON1 和 EECON2 对 EEPROM 进行编程操作，硬件实现了擦除和编程的自定时功能，无需软件查询，节省有限的代码空间，同时利用此特性，启动编程周期之后可以进入睡眠模式，以降低功耗。

数据 EEPROM 在使用（无论是读还是写）之前必须进行以下初始化操作：在未使用到的 EEPROM 某个单元写两次 0xAA，后续程序不要对此单元操作。如：

```
SYSTEM_INIT
.....
.....
LDWI    0x55
STR     EEROM_ADDR
LDWI    0xAA
STR     EEPROM_DATA
LCALL   EEPROM_WRITE
LCALL   EEPROM_WRITE
```

## 编程数据 EEPROM 步骤

要读取数据存储单元，用户必须将地址写入 EEADR 寄存器，然后将 EECON1 寄存器的控制位 RD 置 1。在紧接着的下一周期，EEDAT 寄存器就被 EEPROM 数据写入。因此该数据可由下一条指令读取。EEDAT 将保持这个值直到用户下一次从该单元读取或向该单元写入数据时（在写操作过程中）。

```
BANKSEL EEADR
LDWI    dest_addr
STR     EEADR
BSR     EECON1, RD
LDR     EEDAT, W
```

## 13 时钟测量

此功能可以比较精准的测量内部慢时钟周期。

在此模式下，TIMER2 的预分频、后分频配置自动变为 1:1，组成一个 12 位的定时器，TIMER2 的计数时钟为系统时钟  $F_{osc}$ ，而不是普通模式下的指令时钟  $F_{osc}/2$ 。计数结束后结果自动存到 SOSCPR 寄存器，其单位是系统时钟  $F_{osc}$  的个数。

操作步骤：

1. 为提高计量精度，建议设置 IRCF 为 111，SCS=1，选择 16M 的系统时钟；
2. 把 T2CON.2 置 1，使能 TIMER2；
3. 如果选择 4 次平均，则把 MSCKCON.2 置 1，否则把它清 0；
4. 置位 MSCKCON.1，开始测量；
5. 测量结束后 MSCKCON.1 自动清 0，中断标志置 1；
6. 可以用查询或中断的方式等待结束；
7. 当查询到中断标志为 1 时读取得到的 SOSCPR 即为最终结果。

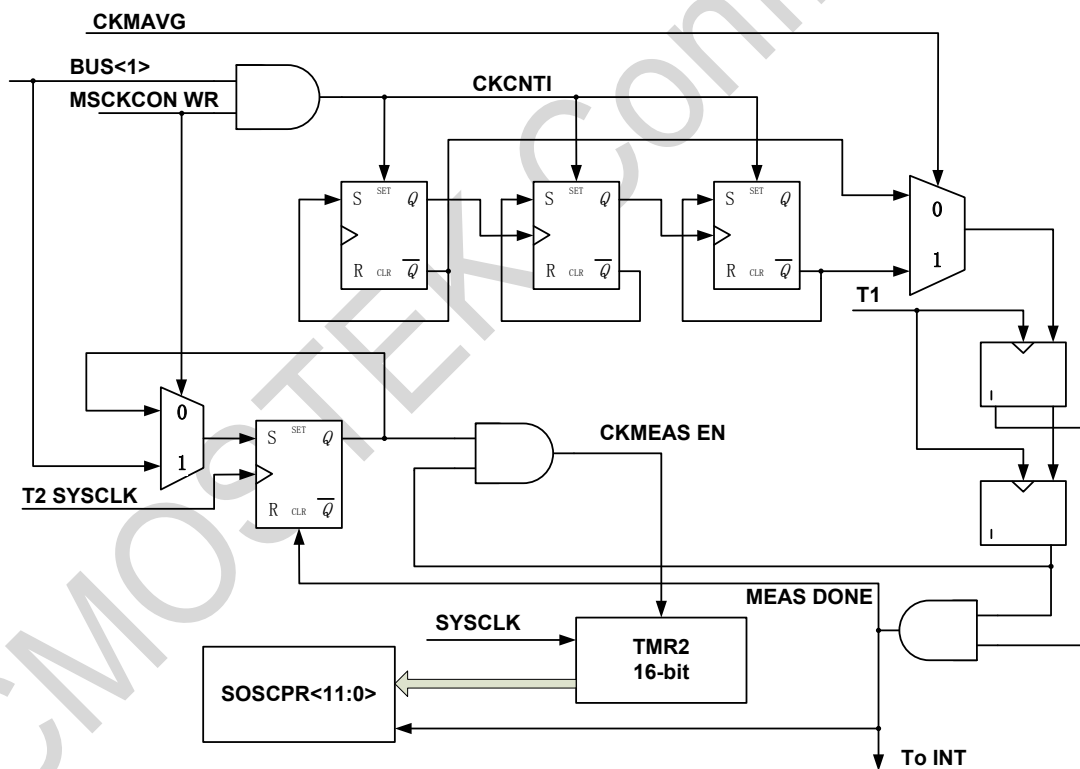


图 13-1. 慢时钟测量模式原理框图

## 14 中断模式

CMT2189B 有以下中断源：

- PA2/INT 管脚进来的外部中断
- Timer0 溢出中断
- PORTA 变化中断
- Timer2 比对相等中断
- EEPROM 数据写中断
- 故障保护时时钟监控器中断
- 比较器中断

中断控制寄存器（INTCON）和外围中断请求寄存器（PIR1）记录了中端标志位。INTCON 同时也包含全局中断使能位 GIE。

当中断被服务后，以下动作自动发生：

- GIE 被清零，从而关闭中断
- 返回地址被推上堆栈
- 程序指针被加载 0004h 地址

中断返回指令，RETFIE，退出中断函数时同时设置 GIE 位，重新使能未屏蔽的中断。

INTCON 寄存器包含以下中断标志位：

- INT 管脚中断
- PORTA 变化中断
- Timer0 溢出中断

PIR1 中包含着外围中断标志位。PIE1 中包含着其对应的中断使能位。

### 14.1 INT 中断

INT 引脚上的外部中断是边沿触发的；当 OPTION 寄存器的 INTEDG 位被置 1 时在上升沿触发，而当 INTEDG 位被清零时在下降沿触发。当 INT 引脚上出现有效边沿时，INTCON 寄存器的 INTF 位置 1。可以通过将 INTCON 寄存器的 INTE 控制位清零来禁止该中断。在重新允许该中断前，必须在中断服务程序中先用软件将 INTF 位清零。如果 INTE 位在进入休眠状态前被置 1，则 INT 中断能将处理器从休眠状态唤醒。

注意：

使用 INT 中断时，必须对 ANSEL 和 CM2CON0 寄存器进行初始化，以便将模拟通道配置为数字输入。配置为模拟输入的引脚总是读为 0。

14.2 PORTA 电平变化中断

PORTA 输入电平的变化会使 INTCON 寄存器的 PAIF 位置 1。可以通过置 1/清零 INTCON 寄存器的 PAIE 位来使能/禁止该中断。此外，可通过 IOCA 寄存器对该端口的各个引脚进行配置。

注意：

- 1. 使用 PORTA 电平变化中断时，必须对 ANSEL 和 CM2CON0 寄存器进行初始化，以便将模拟通道配置为数字输入。配置为模拟输入的引脚总是读为 0；
- 2. 初始化电平变化中断时，应先配置为数字输入 IO，把相应的 IOCA 置 1，然后读取一下该 PORTA；
- 3. 当 IO 电平发生变化时，PAIF 被置 1；
- 4. 清中断标志位之前应该读取一下 PORTA，然后再对 PAIF 清 0。

14.3 中断响应

外部中断包括 INT 管脚进来的或者 PORTA 变化中断的延时一般为 1 到 2 个指令周期。具体视中断发生的实际情况而定。

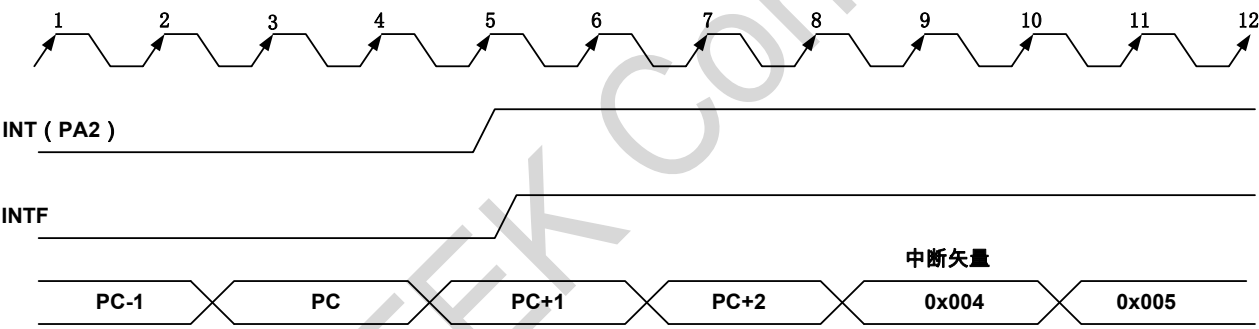


图 14-1. 中断响应时序图

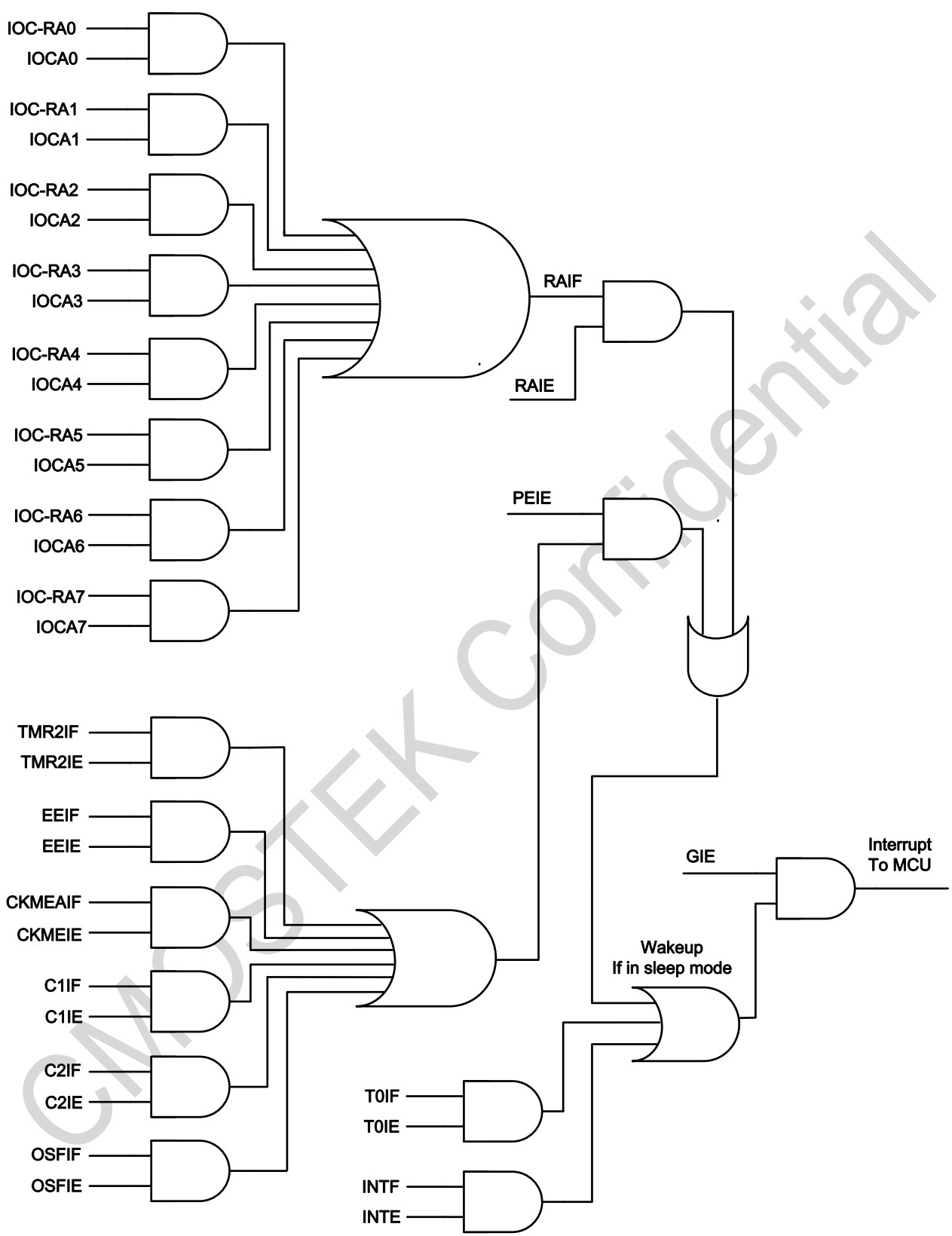


图 14-2. 中断产生电路架构框图



## 14.4 中断过程中的现场保存

在中断过程中，只有返回 PC 被自动保存在堆栈上。一般来说，用户可能需要保存重要的寄存器值在堆栈上，例如 W，STATUS 寄存器等。这些必须由软件来完成。临时寄存器 W\_TEMP 和 STATUS\_TEMP 应该被放置在 GPR 的最后 16byte 里。这 16 个 GPR 落在两个页区间，因此可以稍微节省代码。

CMOSTEK Confidential

## 15 MCU 睡眠省电模式

芯片在执行完 SLEEP 指令后进入睡眠状态。

为了达到最低睡眠功耗，软件应该将所有 IO 置高或低，而且没有外部电路从 IO 耗电。I/O 作为输入的，外部电路应将其拉高或拉低，避免翻转耗电。/MCLR 应该在高电平。

为达到最低功耗，建议配置为晶体模式或者外部时钟模式时，把时钟缺失检测关闭掉，即把 UCFG1 的 FCMEN 位清 0，同时比较器的配置位 CM<2:0>写为 0b111，以关闭比较器模块。

### 15.1 唤醒模式

以下事件可以唤醒芯片：

- /MCLR 管脚上有外部复位
- WDT 超时
- PA2/INT 管脚上有中断，PORTA 变化或其他外围中断

清看门狗指令 CLRWDTC、SLEEP（进入睡眠模式）或者从睡眠模式唤醒，都将清除看门狗计数器。

### 15.2 看门狗唤醒

看门狗工作在内部慢时钟（32KHz），它是一个 16 位的计数器，和定时器 0 共用一个 8 位的预分频器，使能位位于配置寄存器 UCFG0 的第 3 位 WDTEN，为 1 时表示使能看门狗，为 0 时将由 SWDTEN 位决定使能与否，SWDTEN 位于 WDTCON 寄存器。

清看门狗指令 CLRWDTC、SLEEP 会清除看门狗计数器。

在使能了看门狗的情况下，MCU 睡眠时看门狗溢出事件可以作为一个唤醒源，而 MCU 正常工作时作为一个复位源。

## 16 I/O 端口

片内共包含 16 个 GPIO，但限于封装尺寸，仅把 PORTA<7:0>的 6 个 IO 全部引出到芯片管脚（除 PA6 和 PA5）；而 PORTC 把 PC4 和 PC6 引出到芯片管脚，其它均在内部，没有引出来。这些 IO 除了作为普通输入/输出端口以外，还通常具备一些与内核周边电路通讯的功能，具体见下。

### 16.1 PORTA 端口和 TRISA 寄存器

PORTA 是一个 8 位双向端口。与其相应的进出方向寄存器就是 TRISA 寄存器。不过注意到这里第 5 位没有用到，因为 PORTA<5>为单输入方向端口。在 TRISA 寄存器中将某一位设置为“1”会将该对应 PORTA 端口设置为输入端口（此时，输出驱动电路会被关断）。反之，将某一位设置为“0”会将该对应 PORTA 端口设置为输出端口。在置为输出端口时，输出驱动电路会被打开，输出寄存器里的数据会被放置到输出端口。在 PORTA 上进行读动作时，PORTA 内容会是反映输入端口的状态。在 PORTA 上进行写动作时，PORTA 内容会被写入输出寄存器。所有的写操作都是“读-更改-写”这样一个微流程，即数据被读，然后更改，再写入输出寄存器的过程。当 MCLRE 为 1 时，PORTA<5>读的值为 0，此时它是作为外部复位管脚。

### 16.2 端口的其它功能

芯片在 PORTA 的每个端口都有一个状态变化中断选项和弱上拉选型。

#### 16.2.1 弱上拉

PORTA 的每个端口（除了 PORTA<5>）都有一个可以单独设置的内部弱上拉功能。控制 WPUAx 寄存器里的位就可使能或关断这些弱上拉电路。当 GPIO 被设置为输出时，这些弱上拉电路会被自动关断。弱上拉电路在上电复位期间可以被置为关断。这是由 OPTION 寄存器中的/PAPU 位决定的。PORTA<5>内部也有弱上拉功能，它是在将 PORTA<5>设置为/MCLR 功能时自动使能的。当 PORTA<5>被设置为 GPIO 时，该弱上拉电路被自动关断。

#### 16.2.2 状态变化中断

PORTA 的每个端口都可以被单独设置成一个中断源(端口状态变化触发中断)。控制 IOCAx 寄存器里的位就可使能或关断这些端口的中断功能。端口状态变化触发中断的功能在上电复位时无效的。

当端口状态变化触发中断的功能被使能时，当前端口电平值会被与上次读动作所读取数据寄存器的旧值作对比。所有错误匹配结果会被或在一起形成中断标志位 INTCON 寄存器中的 PAIF 标志位该中断可以将芯片从睡眠状态中唤醒。用户需要在中断服务程序中执行以下程序来清除该标志位：

1. 对 PORTA 进行一次读或写得动作，这将结束任何错误匹配的状态；
2. 清零 PAIF 标志位。

错误匹配的条件会一直设置 PAIF 位。对 PORTA 做一次读就可以结束任何错误匹配的状态，使得 PAIF 能被清零。数据寄存器里保持的上一次读的值不会被/MCLR 或低电压复位所影响。只要错误匹配状态存在，PAIF





### 16.3.3 PORTA4/PA4

下图描述了此端口的内部电路结构，PA4 可以被配置为以下功能端口：

- GPIO

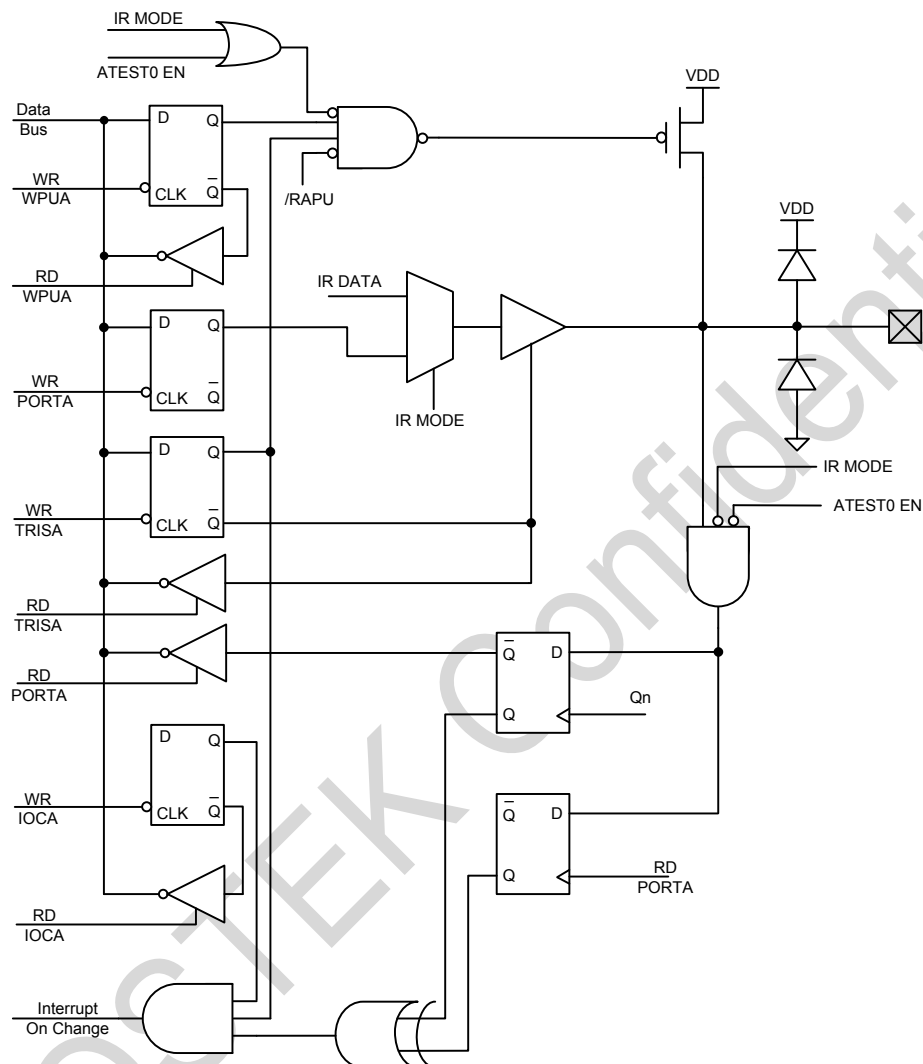


图 16-3. PA4 架构框图

注意：ATEST0 和 IR 均为内部测试用功能，没对用户开放，用户可以忽略。

### 16.3.4 PORTA5/PA5

PA5 由于封装原因，没有引出，所以用户只能通过 UCFG 配置为内部复位，不建议配置为外部复位。

### 16.3.5 PORTA7/PA7

下图描述了此端口的内部电路结构，PA7 可以被配置为以下功能端口：

- GPIO
- 晶振、谐振器连接
- 时钟输入

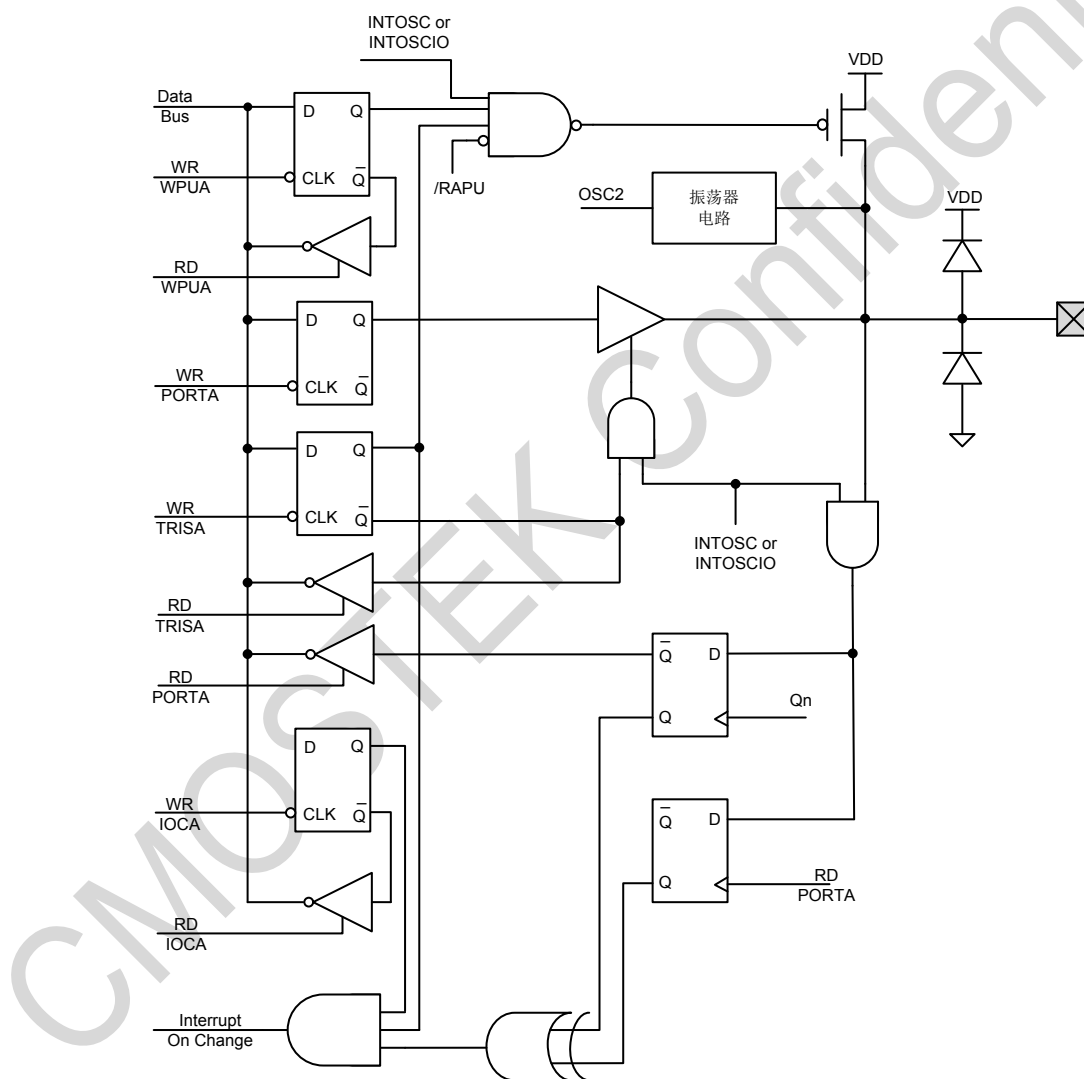


图 16-4. PA7 架构框图





## 17 指令集列表

CMT2189B 采用精简指令架构，一共 37 条指令，以下是各指令的描述。

表 17-1. 指令表

CMT	指令周期	Function	Operation	Status
BCR R, b	1	Bit clear	0-> R(b)	NONE
BSR R, b	1	Bit set	1-> R(b)	NONE
BTSC R, b	1 (2)	Bit test, skip if 0	Skip if R(b)=0	NONE
BTSS R, b	1 (2)	Bit test, skip if 1	Skip if R(b)=1	NONE
NOP	1	No operation	None	NONE
CLRWDT	1	Clear WDT	0-> WDT	/PF, /TF
SLEEP	1	ENTER SLEEP MODE	0-> WDT, STOP OSC	/PF, /TF
STTMD	1	Store W TO TMODE	W-> TMODE	NONE
CTLIO R	1	Control IO direction reg	W-> IODIRr	NONE
STR R	1	Store W to reg	W-> R	NONE
LDR R, d	1	Load reg to d	R-> d	Z
SWAPR R, d	1	Swap halves reg	[R(0-3)R(4-7)]-> d	NONE
INCR R, d	1	Increment reg	R+ 1-> d	Z
INCRSZ R, d	1 (2)	Increment reg, skip if 0	R+ 1-> d	NONE
ADDWR R, d	1	Add W and reg	W+ R-> d	C, HC, Z
SUBWR R, d	1	Sub W from reg	R- W-> d R+ /W+ 1-> d	C, HC, Z
DECR R, d	1	Decrement reg	R- 1-> d	Z
DECRSZ R, d	1 (2)	Decrement reg, skip if 0	R- 1-> d	NONE
ANDWR R, d	1	AND W and reg	R& W-> d	Z
IORWR R, d	1	Inclu.OR W and reg	W  R-> d	Z
XORWR R, d	1	Exclu.OR W and reg	W^ R-> d	Z
COMR R, d	1	Complement reg	/R-> d	Z
RRR R, d	1	Rotate right reg	R(n)-> R(n-1), C-> R(7), R(0)-> C	C
RLR R, d	1	Rotate left reg	R(n)-> R(n+1), C-> R(0), R(7)-> C	C
CLRW	1	Clear working reg	0-> W	Z
CLRR R	1	Clear reg	0-> R	Z
RETI	2	Return from interrupt	Stack-> PC, 1-> GIE	NONE
RET	2	Return from subroutine	Stack-> PC	NONE
LCALL N	2	Long CALL subroutine	N-> PC, PC+1-> Stack	NONE
LJUMP N	2	Long JUMP address	N-> PC	NONE
LDWI I	1	Load immediate to W	I-> W	NONE
ANDWI I	1	AND W and imm	W& I-> W	Z
IORWI I	1	Inclu.OR W and imm	W  I-> W	Z
XORWI I	1	Exclu.OR W and imm	W^ I-> W	Z
RETW I	2	Return, place imm to W	Stack-> PC, I-> W	NONE

CMT		指令周期	Function	Operation	Status
ADDWI	I	1	Add imm to W	$W+I \rightarrow W$	C, HC, Z
SUBWI	I	1	Subtract W from imm	$I-W \rightarrow W$	C, HC, Z

注意：在芯片里，TMODE 寄存器是指 OPTION，即 STTMD 指令的操作是把 W 存到 OPTION。

CMOSTEK Confidential

## 18 文档变更记录

表 18-1. 文档变更记录表

版本号	章节	变更描述	日期
1.0	全部	初始版本发布	2017-11-23
1.1	2	增加 2.7.1 发射速率描述	2017-11-29

## 19 联系方式

无锡泽太微电子有限公司深圳分公司

中国广东省深圳市南山区前海路鸿海大厦 203 室

邮编: 518000

电话: +86 - 755 - 83235017

传真: +86 - 755 - 82761326

销售: sales@cmostek.com

技术支持: support@cmostek.com

网址: www.cmostek.com

**Copyright. CMOSTEK Microelectronics Co., Ltd. All rights are reserved.**

The information furnished by CMOSTEK is believed to be accurate and reliable. However, no responsibility is assumed for inaccuracies and specifications within this document are subject to change without notice. The material contained herein is the exclusive property of CMOSTEK and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of CMOSTEK. CMOSTEK products are not authorized for use as critical components in life support devices or systems without express written approval of CMOSTEK. The CMOSTEK logo is a registered trademark of CMOSTEK Microelectronics Co., Ltd. All other names are the property of their respective owners.