Fremont Micro Devices

FT61F14

数据手册

主要特性

8-bit 基于 EEPROM 的 RISC MCU

Program: 4k x 14; RAM: 512 x 8; Data: 128 x 8

8 / 10 / 14 / 16 / 20 引脚

高精度 12-bit ADC

4 个定时器, 7 路独立 PWM - 3 路带死区控制

通信接口 USART

低 Standby, WDT 和工作电流

POR, LVR, LVD – 单输入比较器

可配置源电流和灌电流

高 ESD, 高 EFT

低 V_{DD} 工作电压

HIRC 可微调

Rev2.00
www.fremontmicro.com

8−bit CPU (EEPROM)

- 49 条 RISC 指令: 2T or 4T
- 16 MHz / 2T (V_{DD} ≥ 2.7)
- 多达 20 个引脚

Memory

- PROGRAM: 4k x 14 bit (读/写保护)
- DATA: 128 x 8 bits
- RAM: 512 x 8 bits
- 16 层硬件堆栈
- 扇区加密，支持 IAP

工作条件 (5V, 25°C)

- V_{DD} (V_{POR} ≤ 1.9V) V_{POR} − 5.5 V
  (通过 POR 自动调整，0°C 以上 ≤1.7V)
- 工作温度等级 1 ⁻40 − ⁺125 °C
- 工作温度等级 2 ⁻40 − ⁺105 °C
- 工作温度等级 3 ⁻40 − ⁺85 °C
- 低 Standby 0.4 μA
- WDT 3.1 μA
- 正常模式 (16 MHz / 2T) 286 μA/mips

高可靠性

- 100 万次擦写次数 (typical)
- > 20 年 / 85°C 存储 (typical)
- ESD > 7 kV, EFT > 5.5 kV

ADC (12−bit)

- 12−bit 精度 (≤ 1 MHz ADC 时钟)
- 7 + 1 通道
- $V_{ADC-REF}$
  - ✓ 内部： 0.5, 2.0, 3.0, V_{DD}
  - ✓ 外部： +, − 可选
- 自动阈值比较
- 触发方式：手动，自动可选
- 支持延时触发

PWM (Total 7 路)

- 支持在 SLEEP 下运行
- 7 个捕捉/比较/PWM 通道：
  - ✓ 独立：占空比，极性
- 3 个通道 (多达 6 个 I/O)：
  - ✓ 互补输出+死区
- 前沿消隐，自动故障刹车 (I/O, LVD, ADC)
- 沿对齐，中心对齐，单脉冲模式

Timers

- WDT (16−bit): 3−bit 预分频
- Timer1 (16−bit): 16−bit 预分频
- Timer2 (16−bit): 4−bit 预分频
- Timer4 (8−bit): 3−bit 预分频
- 自动重载
- 支持在 SLEEP 下运行
- LIRC, 1 or 2x {指令时钟, HIRC，晶振, EC}

通信接口

- USART

I/O PORTS (多达 18 个 I/O)

- 上拉/下拉电阻
- 开漏
- 18 个 I/O 源电流: 4, 8 or 26mA (5V, 25°C)
- 18 个 I/O 灌电流: 53 or 62 mA (5V, 25°C)
- 18 个 I/O: 中断/唤醒

电源管理

- SLEEP
- LVR: 2.0, 2.2, 2.5, 2.8, 3.1, 3.6, 4.1 (V)
- LVD: 2.0, 2.4, 2.8, 3.0, 3.6, 4.0 (V)
  (LVD 可用作极性可选的单输入比较器功能)

系统时钟 (SysClk)

- HIRC 高速内部振荡器
  - ✓ 16MHz <±0.5% typical (1.9−5.5V, 25°C)
  - ✓ 可微调
  - ✓ 1, 2, 4, 8, 16, 32, 64, 128 分频
- LIRC 低功耗低速内部振荡器
  - ✓ 32 kHz 或 256 kHz
- EC 外部时钟 (I/O 输入)
- LP / XT 晶振输入
  - ✓ 双速时钟启动 (HIRC 或 LIRC)
  - ✓ 故障保护时钟监控

其他特性 (欢迎垂询)

- ½ V_{DD} LCD 偏置

集成开发环境 (IDE)

- 片上调试 (OCD)，ISP
- 3 个硬件断点
- 软复位，暂停，单步，跳跃等

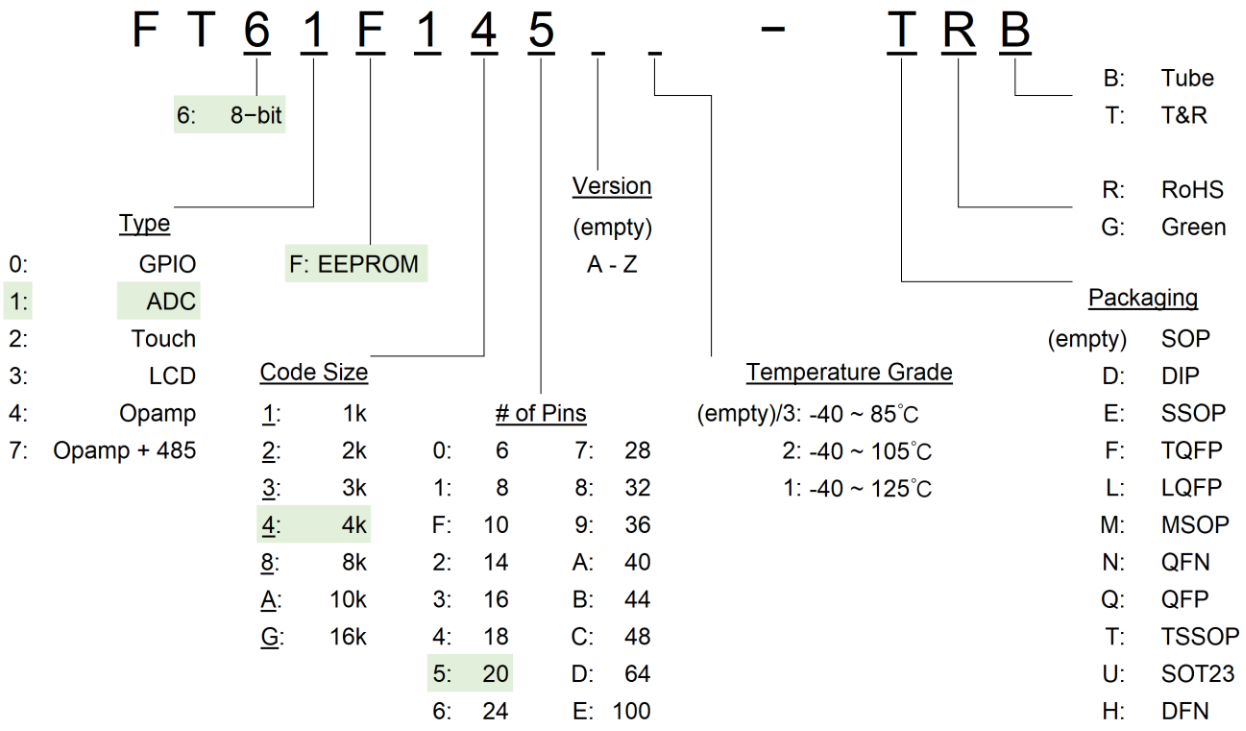
封装

- SOP8 MSOP10 SOP14 SOP16
  SOP20 TSSOP20 QFN20

产品信息和选型表

型号	I/O 数	封装
FT61F141- ab	6	SOP8
FT61F14F- Mab	8	MSOP10
FT61F142- ab	12	SOP14
FT61F143- ab	14	SOP16
FT61F143A- ab		
FT61F143B- ab		
FT61F145- ab	18	SOP20
FT61F145A- ab		
FT61F145- Tab		TSSOP20
FT61F145A- Tab		
FT61F145- Nab		QFN20
FT61F145A- Nab		

此处 a = R; RoHS = G; Green b = B; Tube = T; T&R



MCU 产品订购信息

目录

1. 结构框图和引脚 9
1.1. 引脚图 10
1.2. 管脚描述---按功能分类 13
2. GPIO 16
2.1. IO 端口相关寄存器汇总 19
2.2. I/O 配置 23
2.3. PORTx 功能及优先级 25
2.4. PORT 端口变化中断 26
2.5. 关于读端口 PORTx 27
3. 上电复位 29
3.1. 上电复位流程 29
3.2. 复位时序 29
3.3. 上电复位延时 30
4. 系统复位 31
4.1. 系统复位相关寄存器汇总 32
4.2. 低电压复位 32
4.3. 非法指令复位 33
4.4. 软件复位 33
4.5. EMC 复位 33
4.6. 上电配置过程(BOOT) 33
4.7. 复位源标志位 34
5. LVD 低电压侦测 35
5.1. LVD 相关寄存器汇总 35
5.2. 检测外部电压 36
5.3. LVD 中断 36
5.4. LVD 电平输出 36
6. 振荡器和系统时钟 37
6.1. 振荡器模块相关寄存器汇总 38
6.2. 时钟源模式 40

6.3. 外部时钟模式 40
6.3.1. 振荡器起振定时器 (OST) 40
6.3.2. EC 模式 40
6.3.3. LP 和 XT 模式 40
6.3.4. 内部时钟模式 41
6.3.5. 频率选择位 (MCKCF) 41
6.3.6. HIRC 和 LIRC 时钟切换时序 41
6.3.7. HIRC 时钟特殊功能 42
6.4. 时钟切换 43
6.4.1. 系统时钟选择 (SCS) 位 43
6.4.2. 振荡器起振超时状态 (OSTS) 位 43
6.4.3. 双速时钟启动模式 43
6.4.4. 双速启动模式配置 44
6.4.5. 双速启动顺序 44
6.4.6. 故障保护时钟监控器 44
6.4.7. 故障保护检测 45
6.4.8. 故障保护操作 45
6.4.9. 故障保护条件清除 45
6.4.10. 复位或从休眠中唤醒 45
6.5. 外设时钟门控 46
6.6. 时钟输出 46
7. 慢时钟测量 48
7.1. 慢时钟测量相关寄存器汇总 48
7.2. 测量原理 49
7.3. 上电自动测量 49
7.4. 操作步骤 51
8. 看门狗定时器 52
8.1. 看门狗时钟源 52
8.2. WDT 相关寄存器汇总 53
9. 高级定时器 TIM1 54

9.1. 特性 54
9.2. Timer1 相关寄存器汇总 55
9.3. 功能描述 71
9.3.1. 计数基本单元 71
9.3.2. 计数控制器 80
9.3.3. 捕捉比较通道 90
9.3.4. 同步模式 102
9.3.5. 群组模式 102
9.3.6. TIM1/TIM2 的同步触发模式 102
9.3.7. TIM1 中断 102
9.3.8. 故障刹车源 103
9.3.9. 前沿消隐 105
10. 通用定时器 TIM2 106
10.1. 特性 106
10.2. Timer2 相关寄存器汇总 107
10.3. 功能描述 114
10.3.1. 计数基本单元 114
10.3.2. 捕捉比较通道 115
10.3.3. TIM2 中断 116
11. 基本定时器 TIM4 117
11.1. 特性 117
11.2. TIM4 相关寄存器汇总 117
11.3. TIM4 时钟源 119
11.4. 预分频器 119
11.5. TIM4 中断 119
12. 睡眠模式 120
12.1. 进入 SLEEP 120
12.2. 睡眠的唤醒 121
12.2.1. 使用中断唤醒 122
12.3. 睡眠的系统时钟 122

13. 中断 123
13.1. 中断相关寄存器汇总 123
13.2. 外部中断管脚选择 128
13.3. 中断的使能 129
13.4. 中断的响应时间 129
13.5. 睡眠下的中断 129
13.6. 现场保护 129
14. 数据 EEPROM 和 PROM 130
14.1. DATA EEPROM 相关寄存器汇总 131
14.2. EEADRL 和 EEADRH 寄存器 132
14.2.1. EECON1 和 EECON2 寄存器 132
14.3. 使用数据 EEPROM 132
14.3.1. 读数据 EEPROM 存储器 133
14.3.2. 写数据 EEPROM 存储器 133
14.3.3. 防止误写操作的保护措施 133
14.3.4. 关于 GIE 的清 0 134
14.4. 闪存程序存储器概述 135
14.4.1. 读程序存储器 135
14.4.2. 擦除闪存程序存储器 136
14.4.3. 写程序存储器 137
14.5. 修改闪存程序存储器 139
14.6. 配置字 UCFGx 读访问 140
14.7. 写校验 140
14.8. PROM 内容保护 140
15. 12bit ADC 模块 141
15.1. ADC 相关寄存器汇总 142
15.2. ADC 的配置 145
15.2.1. 端口配置 145
15.2.2. 通道选择 145
15.2.3. 触发方式选择 146

15.2.4. 触发源选择 146
15.2.5. 触发类型选择 146
15.2.6. 触发延时配置 146
15.2.7. ADC 参考电压 146
15.2.8. 转换时钟 147
15.2.9. 中断 148
15.2.10. 转换结果的格式 148
15.2.11. 阈值比较 149
15.3. ADC 的工作原理 149
15.3.1. 启动转换 149
15.3.2. 转换完成 150
15.3.3. 终止转换 150
15.3.4. 休眠模式下 ADC 的工作 150
15.3.5. 外部触发器 150
15.3.6. A/D 转换步骤 151
15.4. A/D 采集时间要求 153
16. USART 接口 154
16.1. USART 接口相关寄存器汇总 155
16.2. 功能描述 160
16.2.1. 一般描述 160
16.2.2. 异步工作模式 160
16.2.3. 同步工作模式 161
16.2.4. 半双工模式 162
16.2.5. 红外工作模式 162
16.2.6. 智能卡模式 163
16.2.7. LIN Master 模式 163
16.2.8. 多芯片通信模式 164
16.2.9. 自动波特率检测 165
17. 存储区读/写保护 166
18. 程序存储器 167

将程序存储器当作数据存储器读取 167
19. 特殊功能寄存器(SPECIAL FUNCTION REGISTERS, SFR) 169
19.1. 初始化配置寄存器 169
19.2. 用户寄存器 171
19.3. 堆栈 184
19.4. STATUS 寄存器 186
19.5. 间接寻址 187
19.5.1. 传统数据存储器 187
19.5.2. 线性数据存储器 188
19.5.3. 程序存储器 189
20. 指令集汇总 (INSTRUCTION SET) 190
20.1. 读-修改-写 (RMW) 指令 192
20.1. 指令详细描述 193
21. 芯片的电气特性 203
21.1. 极限参数 203
21.2. 工作特性 203
21.3. POR, LVR, LVD 204
21.4. I/O 端口电路 205
21.5. 工作电流 (I_{DD}) 205
21.6. 内部振荡器 206
21.7. ADC(12bit)和 ADC VREF 207
21.8. Program 和 Data EEPROM 208
21.9. EMC 特性 208
21.10. 特性图 209
22. 芯片封装信息 214
附录 1，寄存器类型 221
附录 2，文档更改历史 222
联系信息 223

1. 结构框图和引脚

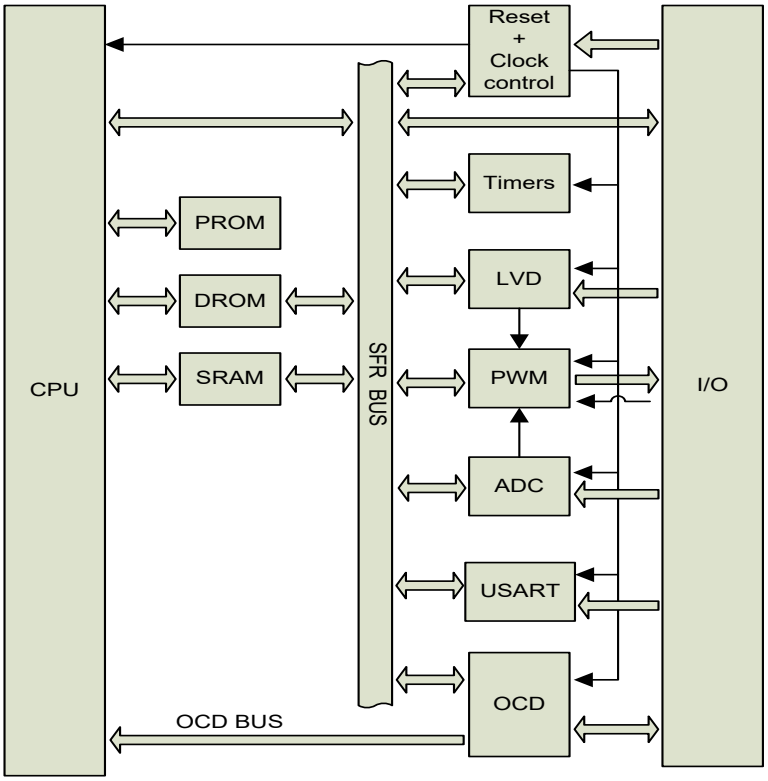


图 1-1 系统结构框图

标准缩写列表如下:

缩写	描述
CPU	Central Processing Unit
SFR	Special Function Registers
SRAM	Static Random Access Memory
DROM	Data EEPROM
PROM	Program EEPROM
TIMx	WDT, Timer1, Timer2, Timer4
PWM	Pulse Width Modulator
ADC	Analog to Digital Converter
LVD	Low Voltage Detect / comparator
USART	Universal Synchronous Asynchronous Receiver Transmitter
OCD	On Chip Debug
I/O	Input / Output

1.1. 引脚图

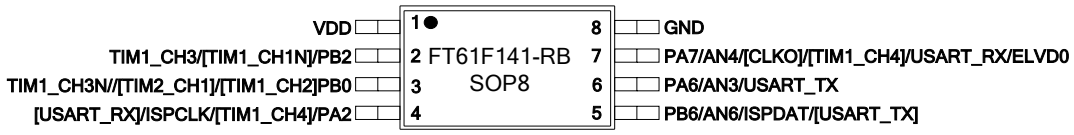


图 1-2 SOP8

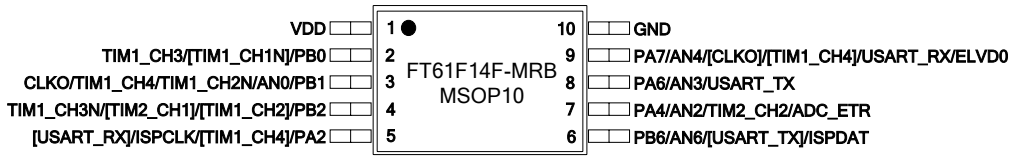


图 1-3 MSOP10

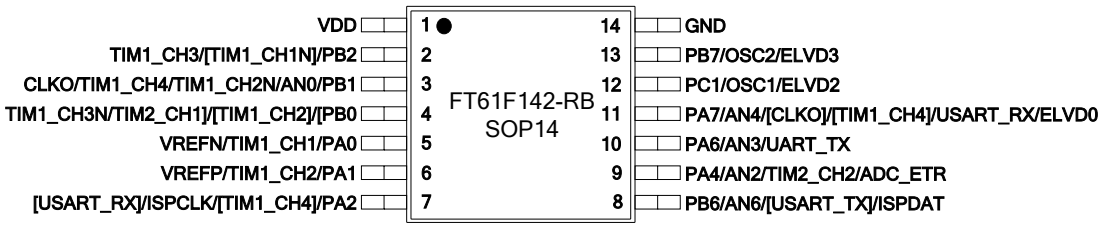


图 1-4 SOP14

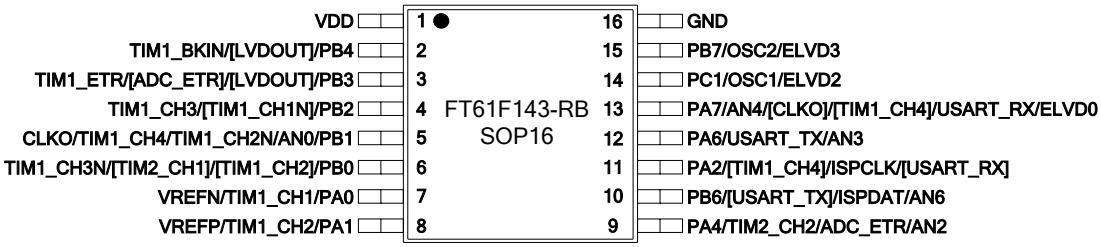


图 1-5 SOP16 (A)

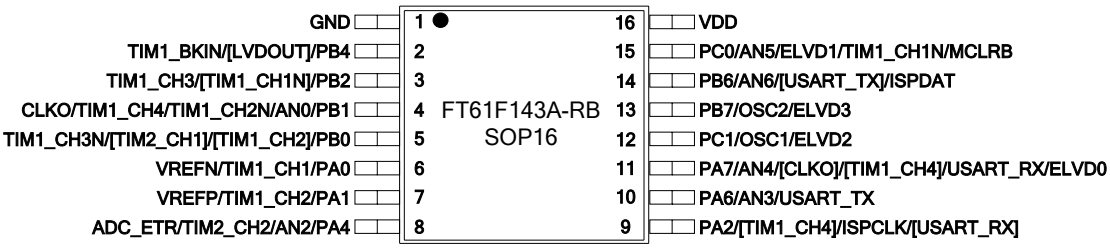


图 1-6 SOP16 (B)

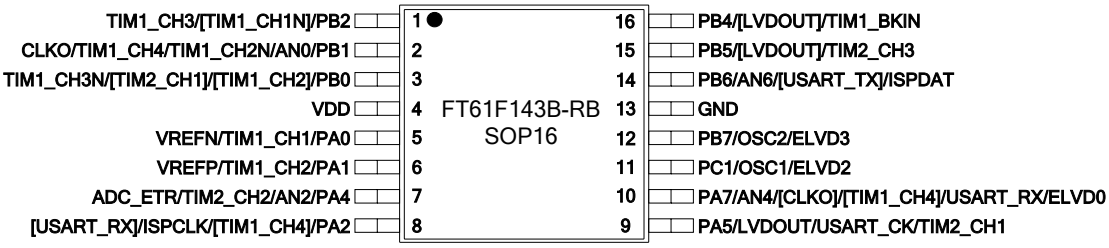


图 1-7 SOP16 (C)

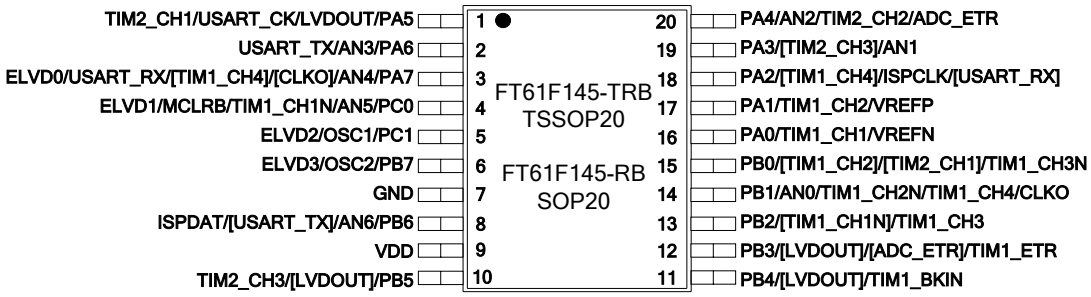


图 1-8 TSSOP20 / SOP20 (A)

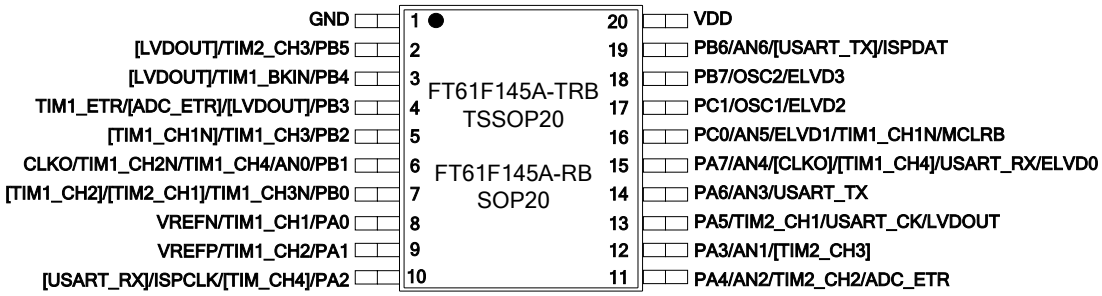


图 1-9 TSSOP20 / SOP20 (B)

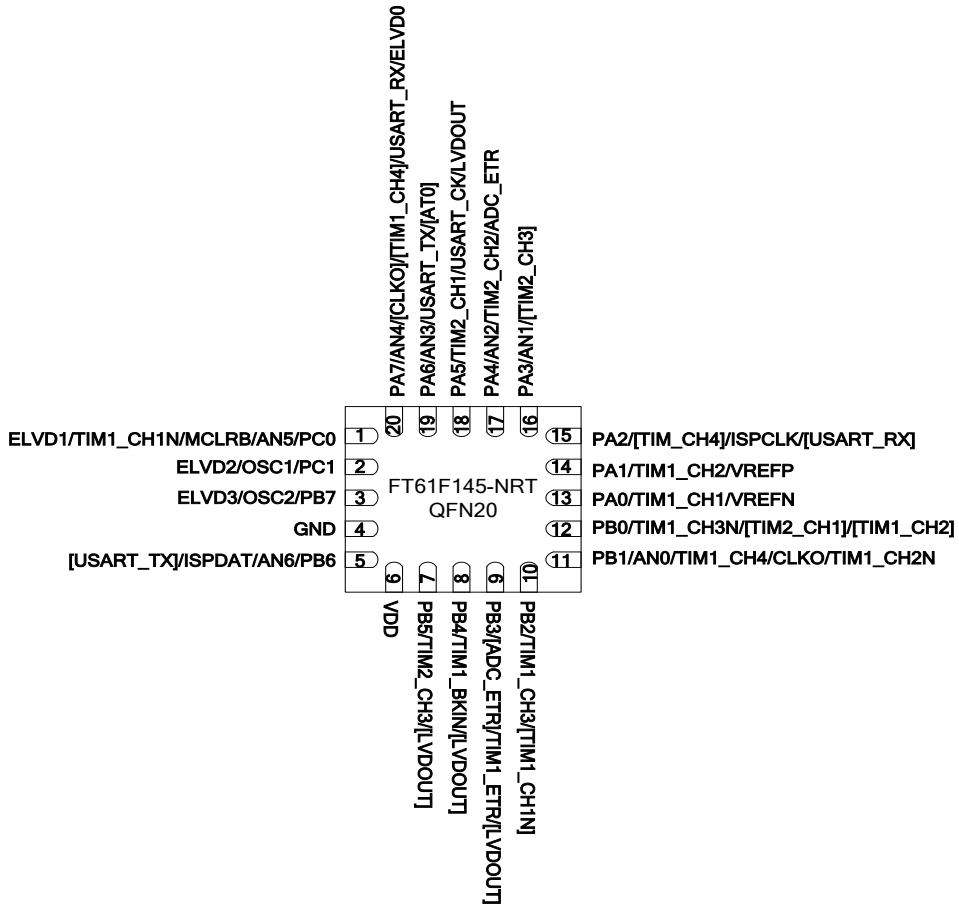


图 1-10 QFN20 (A)

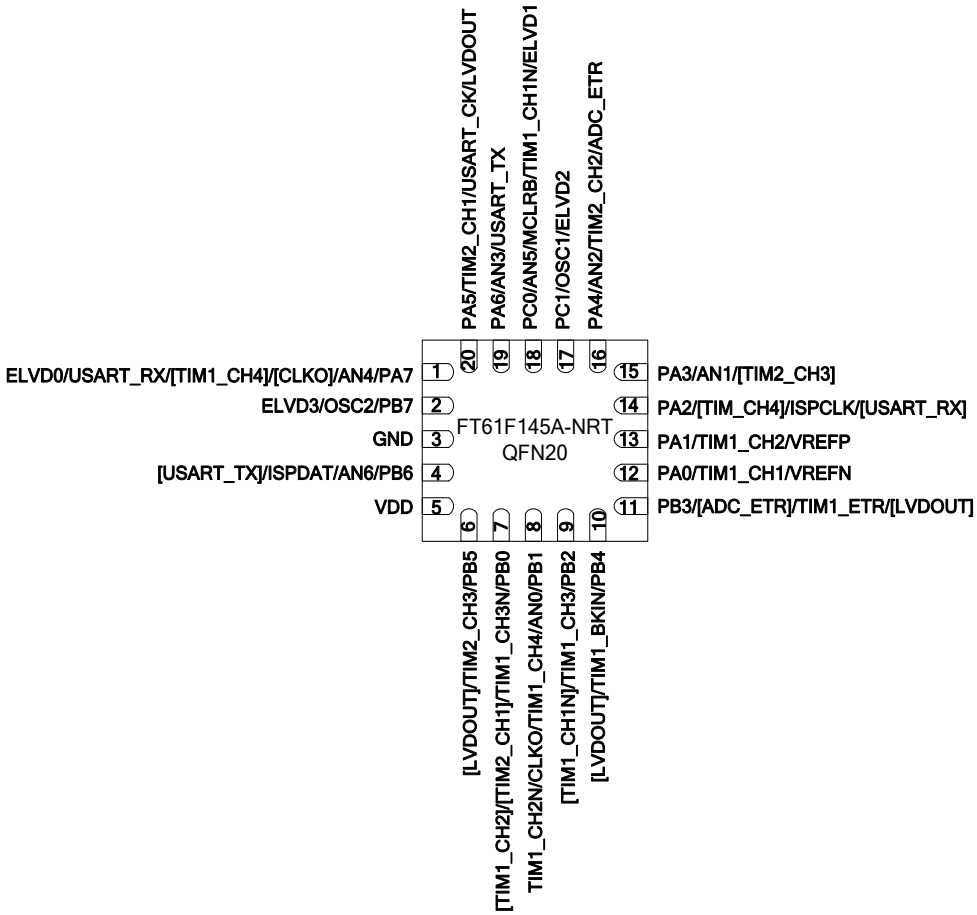

PA5/TIM2_CH1/USART_CK/LVDOUT
PA6/AN3/USART_TX
PC0/AN5/MCLRB/TIM1_CH1N/ELVD1
PC1/OSC1/ELVD2
PA4/AN2/TIM2_CH2/ADC_ETR
ELVD0/USART_RX/[TIM1_CH4]/[CLKO]/AN4/PA7
ELVD3/OSC2/PB7
GND
[USART_TX]/ISPDAT/AN6/PB6
VDD
PA3/AN1/[TIM2_CH3]
PA2/[TIM_CH4]/ISPCLK/[USART_RX]
PA1/TIM1_CH2/VREFP
PA0/TIM1_CH1/VREFN
PB3/[ADC_ETR]/TIM1_ETR/[LVDOUT]
FT61F145A-NRT
QFN20
[LVDOUT]/TIM2_CH3/PB5
[TIM1_CH2]/[TIM2_CH1]/TIM1_CH3N/PB0
TIM1_CH2N/CLKO/TIM1_CH4/AN0/PB1
[TIM1_CH1N]/TIM1_CH3/PB2
[LVDOUT]/TIM1_BKIN/PB4

图 1-11 QFN20 (B)

1.2. 管脚描述---按功能分类

功能	描述	引脚名	对应 GPIO	8 pins	10 pins	14 pins	16(A) pins	16(B) pins	16(C) pins	20(A) pins	20(B) pins	QFN20 (A) pins	QFN20 (B) pins
电源		VDD		1	1	1	1	16	4	9	20	6	5
		GND		8	10	14	16	1	13	7	1	4	3
GPIO	上拉/下拉, 数字输入, 数字输出	PC1				12	14	12	11	5	17	2	17
		PC0						15		4	16	1	18
		PB7				13	15	13	12	6	18	3	2
		PB6		5	6	8	10	14	14	8	19	5	4
		PB5							15	10	2	7	6
		PB4					2	2	16	11	3	8	10
		PB3					3			12	4	9	11
		PB2		2	4	2	4	3	1	13	5	10	9
		PB1			3	3	5	4	2	14	6	11	8
		PB0		3	2	4	6	5	3	15	7	12	7
		PA7		7	9	11	13	11	10	3	15	20	1
		PA6		6	8	10	12	10		2	14	19	19
		PA5							9	1	13	18	20
		PA4			7	9	9	8	7	20	11	17	16
		PA3								19	12	16	15
		PA2		4	5	7	11	9	8	18	10	15	14
		PA1				6	8	7	6	17	9	14	13
		PA0				5	7	6	5	16	8	13	12
ISP 调试	ISP-Data	ISPDAT	PB6	5	6	8	10	14	14	8	19	5	4
	ISP-CLK	ISPCLK	PA2	4	5	7	11	9	8	18	10	15	14
外部复位	上拉	/MCLRB	PC0					15		4	16	1	18
LVD	输入	ELVD0	PA7	7	9	11	13	11	10	3	15	20	1
		ELVD1	PC0					15		4	16	1	18
		ELVD2	PC1			12	14	12	11	5	17	2	17
		ELVD3	PB7			13	15	13	12	6	18	3	2
	输出	LVDOUT	PA5						9	1	13	18	20
		[LVDOUT]	PB5						15	10	2	7	6
		[LVDOUT]	PB4				2	2	16	11	3	8	10
		[LVDOUT]	PB3				3			12	4	9	11

功能	描述	引脚名	对应 GPIO	8 pins	10 pins	14 pins	16(A) pins	16(B) pins	16(C) pins	20(A) pins	20(B) pins	QFN20 (A) pins	QFN20 (B) pins
时钟	输出	CLKO	PB1		3	3	5	4	2	14	6	11	8
		[CLKO]	PA7	7	9	11	13	11	10	3	15	20	1
	OSC +	OSC1	PC1			12	14	12	11	5	17	2	17
	OSC −	OSC2	PB7			13	15	13	12	6	18	3	2
Timer1 (死区)	PWM1	TIM1_CH1	PA0			5	7	6	5	16	8	13	12
	/PWM1	TIM1_CH1N	PC0					15		4	16	1	18
		[TIM1_CH1N]	PB2	2	4	2	4	3	1	13	5	10	9
	PWM2	TIM1_CH2	PA1			6	8	7	6	17	9	14	13
		[TIM1_CH2]	PB0	3	2	4	6	5	3	15	7	12	7
	/PWM2	TIM1_CH2N	PB1		3	3	5	4	2	14	6	11	8
	PWM3	TIM1_CH3	PB2	2	4	2	4	3	1	13	5	10	9
	/PWM3	TIM1_CH3N	PB0	3	2	4	6	5	3	15	7	12	7
	PWM4	TIM1_CH4	PB1		3	3	5	4	2	14	6	11	8
		[TIM1_CH4]	PA7	7	9	11	13	11	10	3	15	20	1
	PWM 故障 刹车输入	TIM1_BKIN	PB4				2	2	16	11	3	8	10
	触发	TIM1_ETR	PB3				3			12	4	9	11
Timer2	PWM5	TIM2_CH1	PA5						9	1	13	18	20
		[TIM2_CH1]	PB0	3	2	4	6	5	3	15	7	12	7
	PWM6	TIM2_CH2	PA4		7	9	9	8	7	20	11	17	16
	PWM7	TIM2_CH3	PB5						15	10	2	7	6
		[TIM2_CH3]	PA3							19	12	16	15
ADC	输入	AN6	PB6	5	6	8	10	14	14	8	19	5	4
		AN5	PC0					15		4	16	1	18
		AN4	PA7	7	9	11	13	11	10	3	15	20	1
		AN3	PA6	6	8	10	12	10		2	14	19	19
		AN2	PA4		7	9	9	8	7	20	11	17	16
		AN1	PA3							19	12	16	15
		AN0	PB1		3	3	5	4	2	14	6	11	8
	触发	ADC_ETR	PA4		7	9	9	8	7	20	11	17	16
		[ADC_ETR]	PB3				3			12	4	9	11
	V_{REF}−	VREFN	PA0			5	7	6	5	16	8	13	12
	V_{REF}+	VREFP	PA1			6	8	7	6	17	9	14	13

功能	描述	引脚名	对应 GPIO	8 pins	10 pins	14 pins	16(A) pins	16(B) pins	16(C) pins	20(A) pins	20(B) pins	QFN20 (A) pins	QFN20 (B) pins
端口变化中断	低电平, 上升沿, 下降沿, 双边沿	PC1				12	14	12	11	5	17	2	17
		PC0						15		4	16	1	18
		PB7				13	15	13	12	6	18	3	2
		PB6		5	6	8	10	14	14	8	19	5	4
		PB5							15	10	2	7	6
		PB4					2	2	16	11	3	8	10
		PB3					3			12	4	9	11
		PB2		2	4	2	4	3	1	13	5	10	9
		PB1			3	3	5	4	2	14	6	11	8
		PB0		3	2	4	6	5	3	15	7	12	7
		PA7		7	9	11	13	11	10	3	15	20	1
		PA6		6	8	10	12	10		2	14	19	19
		PA5							9	1	13	18	20
		PA4			7	9	9	8	7	20	11	17	16
		PA3								19	12	16	15
		PA2		4	5	7	11	9	8	18	10	15	14
		PA1				6	8	7	6	17	9	14	13
		PA0				5	7	6	5	16	8	13	12
USART	USART_CK	USART_CK	PA5						9	1	13	18	20
	USART_TX (开漏)	USART_TX	PA6	6	8	10	12	10		2	14	19	19
		[USART_TX]	PB6	5	6	8	10	14	14	8	19	5	4
	USART_RX	USART_RX	PA7	7	9	11	13	11	10	3	15	20	1
		[USART_RX]	PA2	4	5	7	11	9	8	18	10	15	14

表 1-1 按功能分类的引脚描述

2. GPIO

本芯片共包含 18 个 GPIO。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。

每个端口有 8 个标准寄存器供其操作使用。这些寄存器包括：

- TRISx 寄存器 (数据方向寄存器)
- PORTx 寄存器 (用于读器件引脚上的电平)
- LATx 寄存器 (输出锁存器)
- WPUx 寄存器 (上拉控制)
- WPDx 寄存器 (下拉控制)
- PSRCx 寄存器 (源电流选择)
- PSINKx 寄存器 (灌电流选择)
- ITYPEx 寄存器 (中断类型选择)

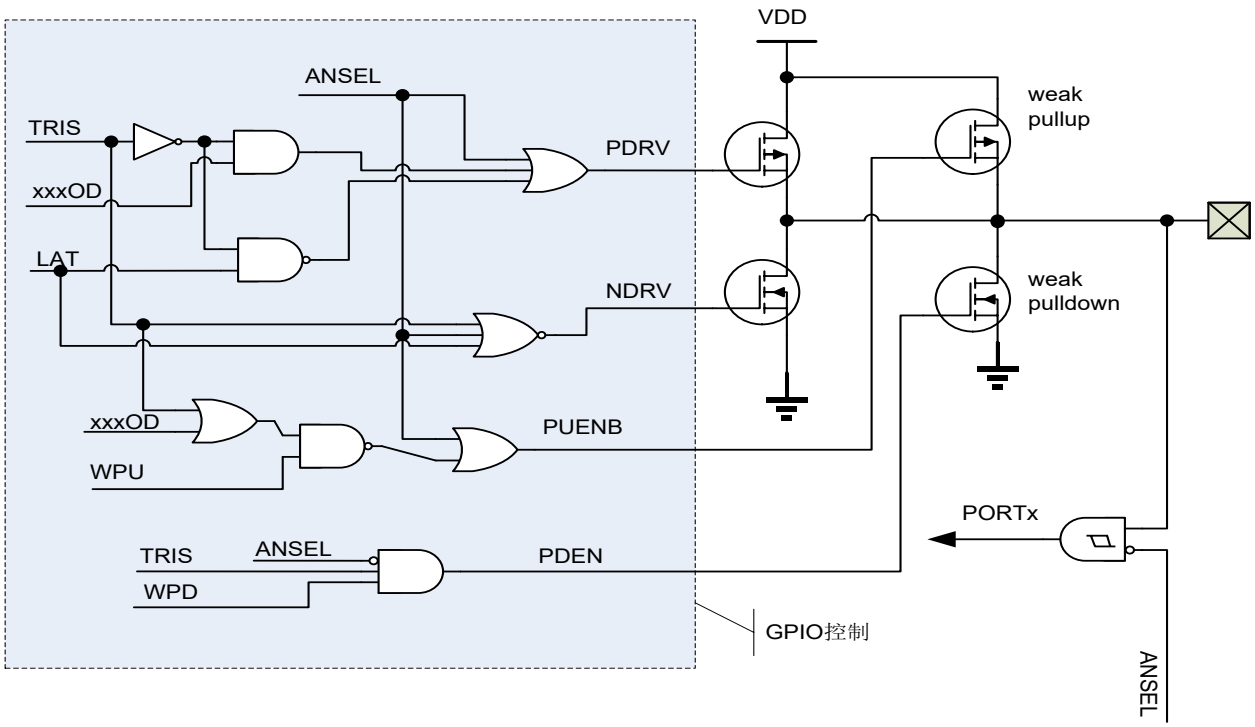


图 2-1 PORT 端口结构原理

某些端口可能有以下额外寄存器：

- ANSELAx (模拟选择寄存器)

通常情况下，当某个端口上的外设使能时，其相关引脚可能不能用作通用输出引脚，但可读取该引脚。

数据锁存器 (LATx 寄存器) 用于对 I/O 引脚所驱动的值进行“读-修改-写”操作。对 LATx 寄存器的写操作与对相应 PORTx 寄存器的写操作具有相同的效果。读 LATx 寄存器将读取保存在 I/O 端口锁存器中的值，而读 PORTx 寄存器将读取实际的 I/O 引脚值。

支持模拟输入的端口具有相关的 ANSELAx 寄存器。当 ANSELA 位置 1 时，禁止与该位相关的数字输入缓冲器。禁止输入缓冲器可防止逻辑输入电路产生短路电流。

所有 I/O 引脚均具有以下功能（**表 2-2，表 2-3**）：

- 数字输出
- 数字输入
- 开漏（USART 相应端口）
- 弱上拉
- 弱下拉

此外，部分 I/O 具有以下特殊功能：

1. 烧录调试引脚（ISP-Data，ISP-CLK），硬件内部连接，不需设置。
2. 通过 IDE 界面配置，且在芯片初始化配置时加载的功能（**表 2-7**）：
   - 外部时钟/晶振输入（OSC1, OSC2）
   - 系统外部复位（/MCLRB）
3. 通过指令对相应 I/O 引脚进行配置的其他功能，可分为 4 类：
   a. 数字输出
      - PWM
      - 内部时钟输出
      - LVD 输出
   b. 数字输入
      - PWM 故障刹车
      - Timer1 触发
      - GPIO 端口变化中断
      - ADC 触发（ADC_ETR）
   c. 模拟输入
      - LVD / BOR
      - ADC
      - $V_{REF}+$
      - $V_{REF}-$
   d. 通信接口
      - USART

引脚名	ISP 调试	时钟	ADC	USART	中断	LVD	PWM	数字 I/O 上拉/下拉	开漏	源电流 (mA)	灌电流 (mA)
PA0			(V_{REF}−)		√		PWM1	√		4, 26	53, 62
PA1			(V_{REF}+)		√		PWM2	√		4, 26	53, 62
PA2	CLK			[RX]	√			√		4, 26	53, 62
PA3			AN1		√		[PWM7]	√		4, 26	53, 62
PA4			AN2/ Trigger		√		PWM6	√		4, 26	53, 62
PA5				CLK	√	LVDOUT	PWM5	√		4, 26	53, 62
PA6			AN3	TX	√			√	√	4, 26	53, 62
PA7		输出	AN4	RX	√	ELVD0	[PWM4]	√		4, 26	53, 62
PB0					√		[PWM2]/ PWM3N/ [PWM5]	√		8,26	53, 62
PB1		输出	AN0		√		PWM4/ PWM2N	√		8,26	53, 62
PB2					√		PWM3/ [PWM1N]	√		8,26	53, 62
PB3			Trigger		√	[LVDOUT]	TIM1_ETR	√		8,26	53, 62
PB4					√	[LVDOUT]	BKIN	√		8,26	53, 62
PB5					√	[LVDOUT]	PWM7	√		8,26	53, 62
PB6	DATA		AN6	[TX]	√			√	√	8,26	53, 62
PB7		OSC−			√	ELVD3		√		8,26	53, 62
PC0			AN5		√	ELVD1	PWM1N	√		4, 8, 26	53, 62
PC1		OSC+			√	ELVD2		√		4, 8, 26	53, 62
注					/MCLRB = PC0					V_{DD}=5, V_{DS}=0.5	

表 2-1 I/O 端口功能

注：PC0-1 支持 3 档可配置源电流驱动能力(参阅 “PSRCx”，**表 2-3**)，和 2 档可配置灌电流驱动能力 (参阅 “PSINKx”，**表 2-3**)。

2.1. IO 端口相关寄存器汇总

名称	地址	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	复位值
ANSELA	0x197	–	ANSELA[6:0]							-000 0000
TRISA	0x8C	TRISA[7:0], PORTA 方向控制								1111 1111
TRISB	0x8D	TRISB[7:0], PORTB 方向控制								1111 1111
TRISC	0x8E	–	–	–	–	–	–	PORTC 方向控制		---- --11
PORTA	0x0C	PORTA 输出寄存器								xxxx xxxx
PORTB	0x0D	PORTB 输出寄存器								xxxx xxxx
PORTC	0x0E	–	–	–	–	–	–	PORTC 输出寄存器		---- --xx
LATA	0x10C	PORTA 数据锁存器								xxxx xxxx
LATB	0x10D	PORTB 数据锁存器								xxxx xxxx
LATC	0x10E	–	–	–	–	–	–	PORTC 数据锁存器		---- --xx
WPUA	0x18C	PORTA 弱上拉								0000 0000
WPUB	0x18D	PORTB 弱上拉								0000 0000
WPUC	0x18E	–	–	–	–	–	–	PORTC 弱上拉		---- --00
WPDA	0x20C	PORTA 弱下拉								0000 0000
WPDB	0x20D	PORTB 弱下拉								0000 0000
WPDC	0x20E	–	–	–	–	–	–	PORTC 弱下拉		---- --00
ODCON0	0x21F	–	–	–	–	–	–	–	UROD	---- ---0
PSRC0	0x11A	PORTA 源电流设置								1111 1111
PSRC1	0x11B	PORTB 源电流设置								1111 1111
PSRC2	0x11C	–	–	–	–	PORTC 源电流设置				---- 1111
PSINK0	0x19A	PORTA 灌电流设置								0000 0000
PSINK1	0x19B	PORTB 灌电流设置								0000 0000
PSINK2	0x19C	–	–	–	–	–	–	PORTC 灌电流设置		---- --00
ITYPE0	0x11E	PORTx[1:0] (x = A, B, C)和 PORTy[3:2] (y = A, B)外部管脚中断类型设置								0000 0000
ITYPE1	0x11F	PORTy[7:4] (y = A, B) 外部管脚中断类型设置								0000 0000
AFP0	0x19E	–	–	管脚重映射寄存器 0					–	--00 000-
AFP1	0x19F	–	–	管脚重映射寄存器 1						--00 0000
EPS0	0x118	外部中断 EINT3~0 管脚选择								0000 0000
EPS1	0x119	外部中断 EINT7~4 管脚选择								0000 0000
EPIE0	0x94	外部管脚中断使能位								0000 0000
EPIF0	0x14	外部管脚中断标志位								0000 0000

表 2-2 I/O 相关用户寄存器的地址和复位值

名称	状态		寄存器	地址	复位值
TRISA	PORTA	PORT 端口数字输出 (方向控制) 1 = 关闭 0 = 使能 (关闭上拉/下拉)	TRISA[7:0]	0x8C	RW-1111 1111
TRISB	PORTB		TRISB[7:0]	0x8D	RW-1111 1111
TRISC	PORTC		TRISC[1:0]	0x8E	RW-11
ANSELA	1 = 关闭上拉/下拉及数字输入(仅适用于 7 个 ADC 通道) 0 = (无动作)		ANSELA[6:0]	0x197	RW-000 0000
WPUA	PORTA	弱上拉 1 = 使能 0 = 关闭	WPUA[7:0]	0x18C	RW-0000 0000
WPUB	PORTB		WPUB[7:0]	0x18D	RW-0000 0000
WPUC	PORTC		WPUC[1:0]	0x18E	RW-00
WPDA	PORTA	弱下拉 1 = 使能 0 = 关闭	WPDA[7:0]	0x20C	RW-0000 0000
WPDB	PORTB		WPDB[7:0]	0x20D	RW-0000 0000
WPDC	PORTC		WPDC[1:0]	0x20E	RW-00
PORTA	PORTA	数据输出寄存器 读：返回 IO 引脚上的电平 写：写入相应的 LATx 寄存器	PORTA[7:0]	0x0C	RW-xxxx xxxx
PORTB	PORTB		PORTB[7:0]	0x0D	RW-xxxx xxxx
PORTC	PORTC		PORTC[1:0]	0x0E	RW-xx
LATA	PORTA	数据锁存器寄存器	LATA[7:0]	0x10C	RW-xxxx xxxx
LATB	PORTB		LATB[7:0]	0x10D	RW-xxxx xxxx
LATC	PORTC		LATC[1:0]	0x10E	RW-xx
UROD	USART_TX 开漏	1 = 使能 0 = 关闭	ODCON0[0]	0x21F	RW-0
AFP0	TIM1_CH2 复用管脚选择 [1]	1 = PB0 0 = PA1	AFP0[5]	0x19E	RW-0
	TIM2_CH1 复用管脚选择	1 = PB0 0 = PA5	AFP0[4]		RW-0
	TIM2_CH3 复用管脚选择	1 = PA3 0 = PB5	AFP0[3]		RW-0
	TIM1_CH1N 复用管脚选择	1 = PB2 0 = PC0	AFP0[2]		RW-0
	ADC_ETR 复用管脚选择	1 = PB3 0 = PA4	AFP0[1]		RW-0
AFP1	USART_TX 复用管脚选择 00 = PA6 10 = PA7 01 = PB6 11 = PA2		AFP1[5:4]	0x19F	RW-00
	USART_RX 复用管脚选择 00 = PA7 10 = PA6 01 = PA2 11 = PB6		AFP1[3:2]		
	TIM1_CH4 复用管脚选择 00/01 = PB1 10 = PA7 11 = PA2		AFP1[1:0]		

[1] 对于 VerA 版芯片，当 TIM1_CH2 输出使能时 (T1CC2S[1:0]=00，T1CC2E=1，MOE=1)，如果 AFP0[5]选择 PB0，此时 PA1 仍被 TIM1_CH2 功能占据，不能作为 GPIO 输出，但可以作为输入，开发应用时要留意。

名称	状态		寄存器	地址	复位值
PSINK0	PA7−PA0	灌电流 (mA) 1 = 62 0 = 53	PSINK0[7:0]	0x19A	RW−0000 0000
PSINK1	PB7−PB0		PSINK1[7:0]	0x19B	RW−0000 0000
PSINK2	PC1−PC0		PSINK2[1:0]	0x19C	RW−00
PSRCA[7:0]	PA7−PA0	源电流 (mA) 1 = 26 0 = 4	PSRC0[7:0]	0x11A	RW−1111 1111
PSRCB[7:0]	PB7−PB0	源电流 (mA) 1 = 26 0 = 8	PSRC1[7:0]	0x11B	RW−1111 1111
PSRCC[3:2]	PC1	源电流 (mA) 11 = 26 01/10 = 8 00 = 4	PSRC2[3:0]	0x11C	RW−1111
PSRCC[1:0]	PC0				

表 2-3 I/O 相关用户寄存器

名称	状态		寄存器	地址	复位值
ITYPE0[1:0]	PORTx.0	外部管脚中断触发类型 00 = 低电平 01 = 上升沿 10 = 下降沿 11 = 双边沿	ITYPE0[1:0]	0x11E	RW−00
ITYPE0[3:2]	PORTx.1		ITYPE0[3:2]		RW−00
ITYPE0[5:4]	PORTy.2		ITYPE0[5:4]		RW−00
ITYPE0[7:6]	PORTy.3		ITYPE0[7:6]		RW−00
ITYPE1[1:0]	PORTy.4		ITYPE1[1:0]	0x11F	RW−00
ITYPE1[3:2]	PORTy.5		ITYPE1[3:2]		RW−00
ITYPE1[5:4]	PORTy.6		ITYPE1[5:4]		RW−00
ITYPE1[7:6]	PORTy.7		ITYPE1[7:6]		RW−00

表 2-4 I/O 中断触发寄存器 (x = A, B, C; y = A, B)

名称	状态		寄存器	地址	复位值
EINT0	EINT0 管脚选择	00 = PA0 10 = PC0 01 = PB0 11 = 保留	EPS0[1:0]	0x118	RW-00
EINT1	EINT1 管脚选择	00 = PA1 10 = PC1 01 = PB1 11 = 保留	EPS0[3:2]		RW-00
EINT2	EINT2 管脚选择	00 = PA2 1x = 保留 01 = PB2	EPS0[5:4]		RW-00
EINT3	EINT3 管脚选择	00 = PA3 1x = 保留 01 = PB3	EPS0[7:6]		RW-00
EINT4	EINT4 管脚选择	00 = PA4 1x = 保留 01 = PB4	EPS1[1:0]	0x119	RW-00
EINT5	EINT5 管脚选择	00 = PA5 1x = 保留 01 = PB5	EPS1[3:2]		RW-00
EINT6	EINT6 管脚选择	00 = PA6 1x = 保留 01 = PB6	EPS1[5:4]		RW-00
EINT7	EINT7 管脚选择	00 = PA7 1x = 保留 01 = PB7	EPS1[7:6]		RW-00

表 2-5 外部中断管脚选择寄存器

名称	状态		寄存器	地址	复位值
EPIEx	外部中断使能位	1 = 使能 0 = 禁止	EPIE0[7:0]	0x94	RW-00000000
EPIF0x [2]	外部中断标志位	1 = Yes (锁存) 0 = No	EPIF0[7:0]	0x14	R_W1C-00000000

表 2-6 外部中断使能和中断标志寄存器

名称	功能	默认
MCLRE	外部 I/O 复位	关闭
FOSC	• LP：PC1 (+) 和 PB7 (−) 接外部低速晶振 • XT：PC1 (+) 和 PB7 (−) 接外部高速晶振 • EC：PC1 (+) 接外部时钟输入，PB7 为 I/O • INTOSCIO：PC1 和 PB7 为 I/O	INTOSCIO

表 2-7 I/O 相关初始化配置寄存器

[2] 写 1 清 0，写 0 无效。建议只使用 STR、MOVWI 指令进行写操作，而不要用 BSR 位操作。

2.2. I/O 配置

每个 PORT 端口，均需根据其相应功能配置以下 4 个模块（**表 2-3**）：

- 数字输出
- 数字输入
- 开漏
- 弱上拉
- 弱下拉

功能	数字输入	上拉/下拉	数字输出	设置
ISP-DATA	On	Off	On	(硬件内置，忽略指令)
ISP-CLK	On	Off	Off	(硬件内置，忽略指令)
/MCLRB	On	上拉	Off	(初始化配置，忽略指令)
时钟输出	(忽略)	Off	On	(初始化配置，忽略指令)
OSC+ (EC)	On	(可选)	Off	(初始化配置，忽略指令)
OSC+ / OSC− (LP, XT)	Off	Off	Off	(初始化配置，忽略指令)
ADC	Off	Off	Off	TRISx = 1; ANSELAx = 1
USART 输出	On	Off	On	TRISx = 0
LVD	Off (5)	Off	Off	TRISx = 1; ANSELAx = 1(除 PC1, PB7 外)
V_{REF}+ / V_{REF}−	Off	Off	Off	TRISx = 1
ADC 触发	On	(可选)	Off	TRISx = 1
USART 输入	On	(可选)	Off	TRISx = 1
端口变化中断	On	(可选)	Off	TRISx = 1
BKIN	On	(可选)	Off	TRISx = 1
数字输入	On	(可选)	Off	TRISx = 1
PWM	On	Off	On	TRISx = 0
数字输出	On	Off	On	TRISx = 0

表 2-8 I/O 配置标志和用户寄存器

注：

1. TRISx = 0：“数字输出”使能，“上拉/下拉”自动关闭 (忽略 WPDx, WPUx)。
2. TRISx = 1：“数字输出”关闭。
3. ANSELAx = 1：“上拉”、“下拉”、“数字输入”自动关闭 (忽略 WPDx, WPUx)。
4. 可关闭“数字输入”的唯一指令为“ANSELAx = 1”。
5. 将 PORT 端口设置为 LVD 输入时，其“数字输入”、“上拉”和“下拉”功能被自动关闭。当 LVD 输入需要在不同的通道之间切换使用时，通过设置“ANSELAx = 1”可关闭当前未被选择通道的“数字输入”，但 PC1 和 PB7 无 ANSELAx 控制，无法关闭“数字输入”，因此不应仅在部分时间作为 LVD 输入。
6. /MCLR 使能：PC0 的弱上拉功能自动使能 (忽略 WPUC[0])；读 PORTC[0] 的值为“0”。
7. 对 PORTx 数据输出寄存器进行写操作，I/O 端口将输出相应的逻辑电平。每组多达 8 个 I/O 的数据寄存器共用相同的地址，写操作实际执行‘读-修改-写’的过程，即先读取该组 PORTx 端口锁存器值 (输出或输入)，然后修改，再写回 PORTx 数据寄存器。
8. 数字输出和数字输入功能可以共存，有些应用需要同时使能数字输出和数字输入。

9. 当 TRISx = 0 时，通过软件可读取 PORTx 输出或输入锁存器 LATx 的值。
10. ODCON0x = 1：“ USART_TX ”管脚开漏输出。管脚的开漏功能和内部上拉功能可以同时打开。
11. 完全复位或系统复位时，PORTx 寄存器不会复位，但 TRISx 将被重置为“1”，从而关闭输出。
12. 部分管脚输入/输出支持重映射功能，AFPx 寄存器可在两个管脚之间选择。

2.3. PORTx 功能及优先级

每个 I/O 管脚均复用了多个功能，当某管脚复用的功能模块都使能输出的情况下，就存在优先级的问题。

因为输入是连到各个功能模块的，故输入不存在优先级问题，例如 PB0 作为 GPIO 输入功能时，同时也作为 TIM2 的捕捉输入。

管脚名称	功能优先级 0	功能优先级 1	功能优先级 2	功能优先级 3
PA0	PA0	TIM1_CH1	–	–
PA1	PA1	TIM1_CH2	–	–
PA2	PA2	[TIM1_CH4]	ISPCLK (调试模式)	
PA3	PA3	[TIM1_CH3]	–	–
PA4	PA4	TIM2_CH2	–	–
PA5	PA5	LVDOUT	USART_CK	TIM2_CH1
PA6	PA6	USART_TX	–	–
PA7	PA7	[CLKO]	[TIM1_CH4]	–
PB0	PB0	[TIM1_CH2]	[TIM2_CH1]	TIM1_CH3N
PB1	PB1	TIM1_CH2N	TIM1_CH4	CLKO
PB2	PB2	[TIM1_CH1N]	TIM1_CH3	–
PB3	PB3	[LVDOUT]	–	–
PB4	PB4	[LVDOUT]	–	–
PB5	PB5	[LVDOUT]	TIM2_CH3	–
PB6	PB6	[USART_TX]	ISPDAT (调试模式)	–
PB7	PB7	OSC2 (XT 模式)	–	–
PC0	PC0	TIM1_CH1N	MCLRB (复位脚)	–
PC1	PC1	OSC1 (XT 模式)	–	–

表 2-9 IO 功能优先级

注：

A 版芯片的输出映射有以下限制，使用时应注意：

1. TIM1_CH2 捕捉输入没有映射到 PB0，即在 PB0 管脚上：TIM1_CH2 只具备比较输出和 PWM 输出；

2. 当 TIM2_CH1 输出使能时 (T2CC1E=1)，PB0 不能配置为 TIM1_CH2 输出，否则 PB0 将输出 TIM2_CH1 波形；

对于 B 版芯片，则无该限制。

2.4. PORT 端口变化中断

所有 I/O 都可选择为外部中断源 (通过 EPS0 和 EPS1 设置)， 同一时刻最多只有 8 个 IO 可以作为外部中断管脚，中断类型可通过寄存器 ITYPE0 和 ITYPE1 设置。它们具备以下特性：

- 上升沿中断
- 下降沿中断
- 双边沿中断
- 低电平中断

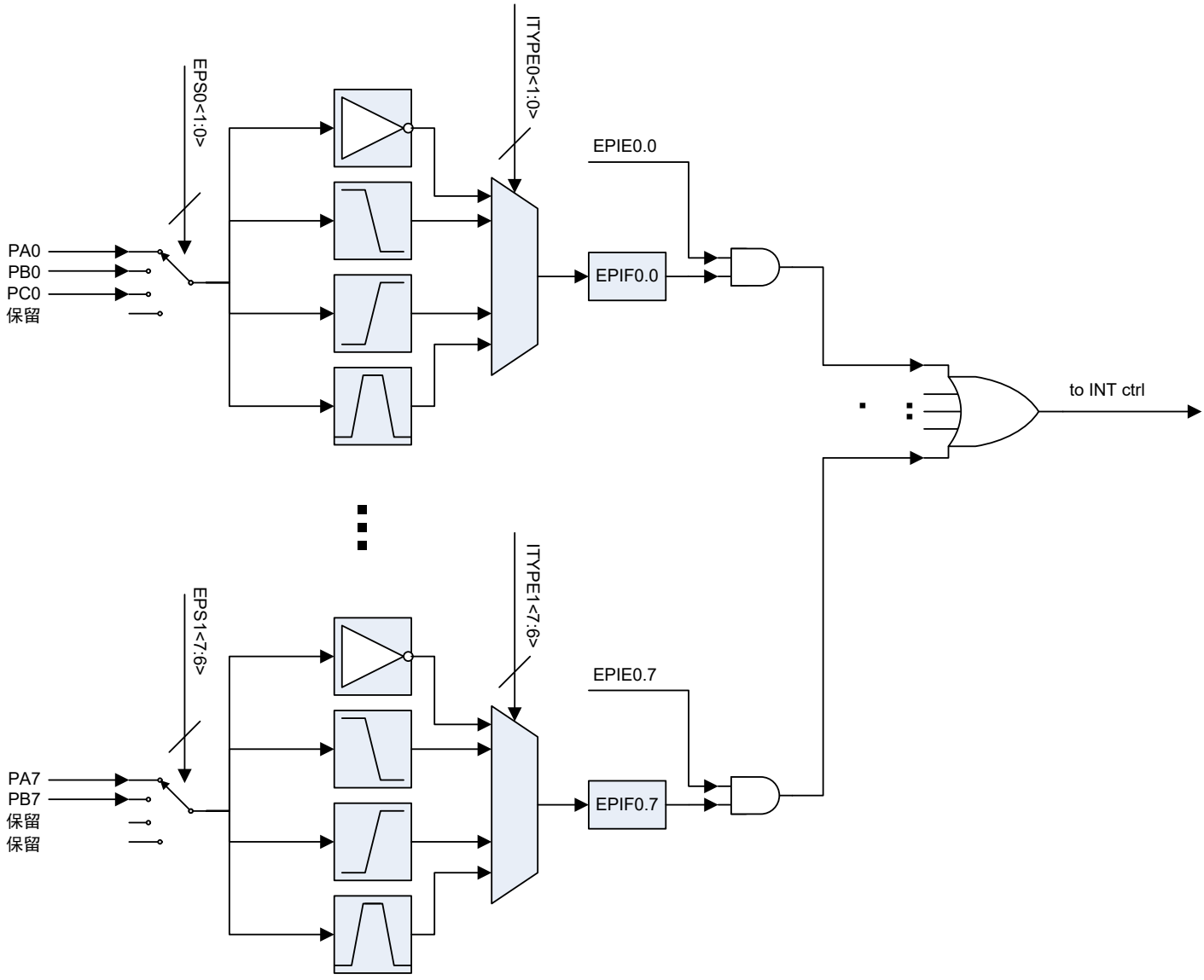


图 2-2 端口变化中断结构框图

2.5. 关于读端口 PORTx

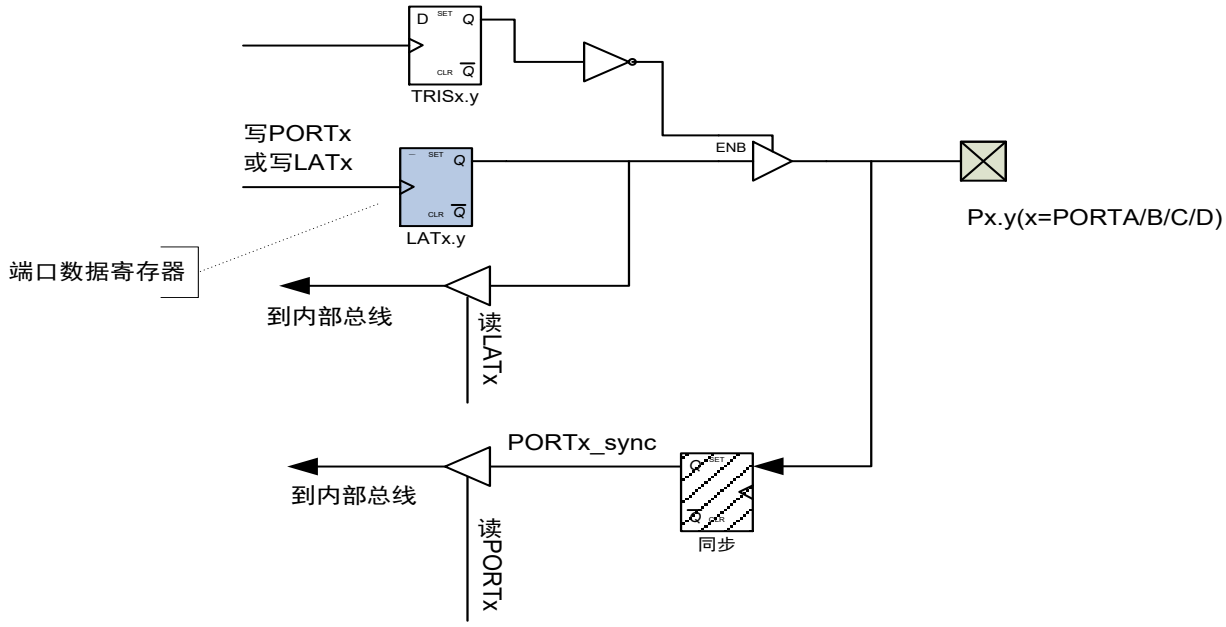


图 2-3 端口读操作原理框图

在 FT61F14x 系列芯片中，操作 GPIO 有两种方式：访问 PORTx 寄存器或者 LATx 寄存器，它们有不同的 SFR 地址。

对于读操作：“读 PORTx”返回的是管脚经过同步寄存器后的值，而“读 LATx”返回的是端口数据寄存器的值；换言之，软件对端口数据寄存器写操作之后，至少要经过一个系统时钟之后，才能通过“读 PORTx”的方式得到新值，而“读 LATx”则无需等待；

对于写操作：无论是写 PORTx 还是 LATx，都是对端口数据寄存器进行写；

由于以上特性，当软件使用“读-修改-写” 指令对 PORTx 进行写操作时，需要特别注意以下情形：

```
BSR     PORTx, n    ;    对 PORTx 第 n 位置 1
BSR     PORTx, m    ;    对 PORTx 第 m 位置 1
…
```

软件期望的波形如下：

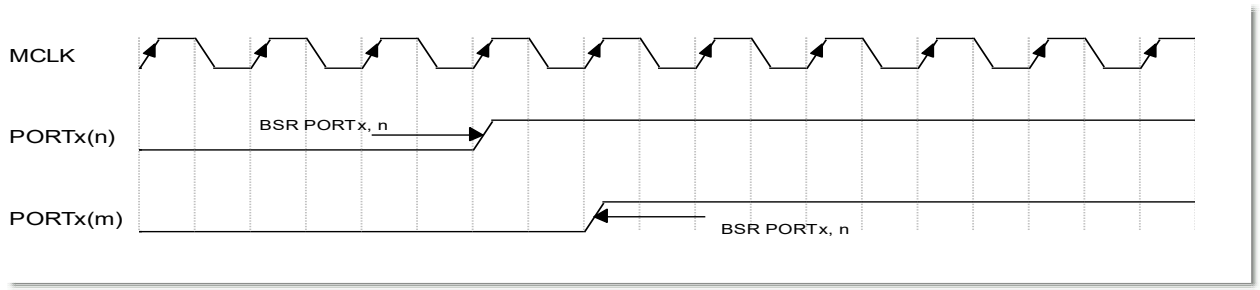


图 2-4 连续使用 RMW 指令对 PORTx 写操作的期望时序

实际输出波形如下：

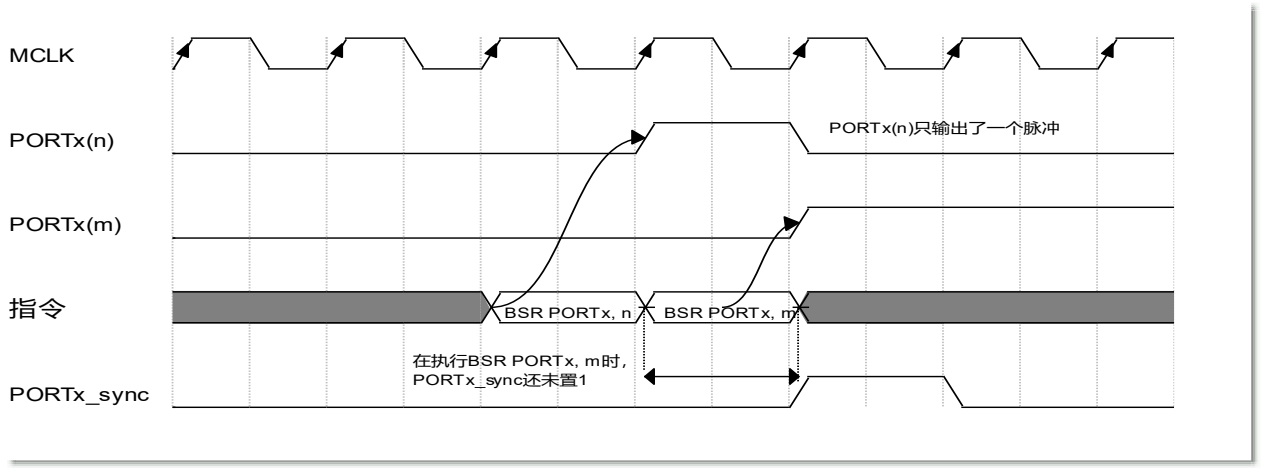


图 2-5 连续使用 RMW 指令对 PORTx 写操作的输出时序

出现这个现象的原因是在执行“BSR PORTx, m” (回顾一下 RMW 指令的执行流程：先读取 PORTx，修改数据，写 PORTx (LATx))时，由于同步的原因，PORTx_sync 还保持为 0，那么写回 PORTx 时刻，这一位的“0”又被写回到 LATx，导致管脚 PORTx.n 只有一个高脉冲。

有以下两种方式解决这一问题：

a) 在 PORTx 连续写操作中间插入一个 NOP；

```
BSR PORTx, n    ;    对 PORTx 第 n 位置 1
NOP             ;    插入 NOP 等待
BSR PORTx, m    ;    对 PORTx 第 m 位置 1
```

b) 或者，写操作用 LATx 寄存器而不是 PORTx；

```
BSR LATx, n     ;    直接操作端口数据寄存器 LATx
BSR LATx, m     ;    直接操作端口数据寄存器 LATx
```

注意：只有 1T 速度模式下才有该现象，不存在于其它 2T/4T 模式，原因是处于 2T/4T 模式下，执行后续指令时，PORTx_sync 已经同步到最新的值。

但如果 I/O 直接驱动 LED 或三极管，或者其它会导致 I/O 被拉低的电路时，无论是在何种速度模式下，软件读到的 PORTx 值将是 0，这种情况下，对 I/O 数据寄存器只能使用 LATx，而不能是 PORTx。

3. 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，上电复位释放后，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。

3.1. 上电复位流程

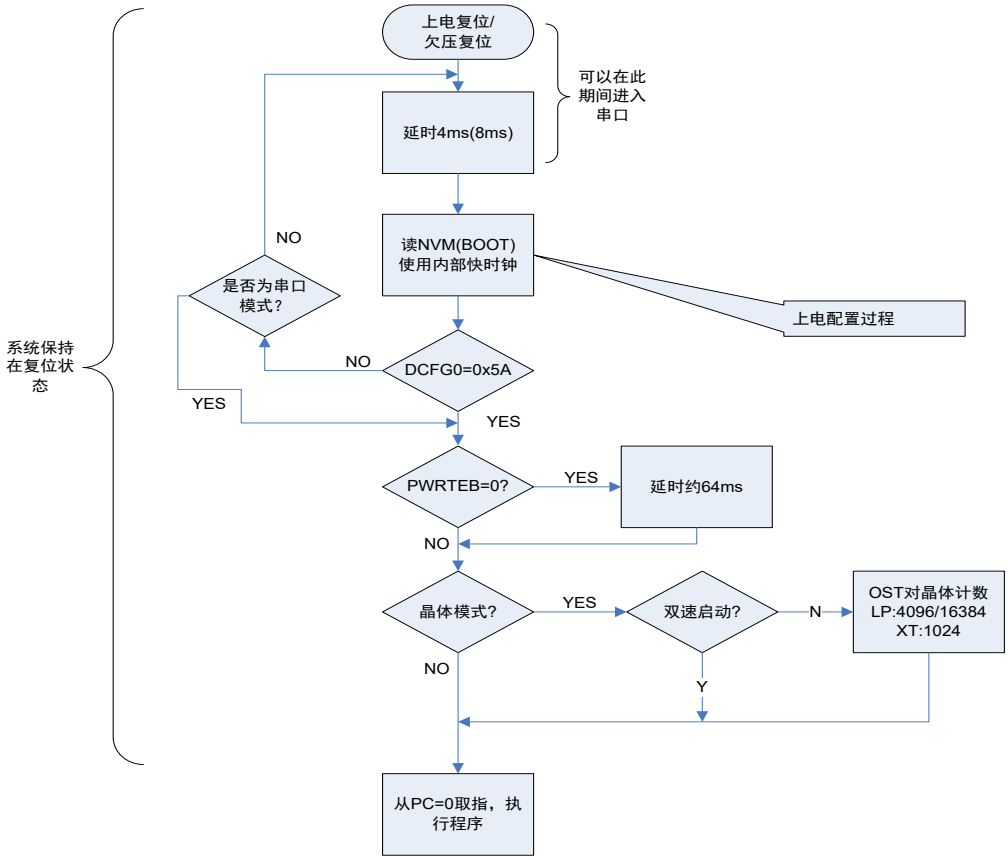


图 3-1 上电复位流程图

3.2. 复位时序

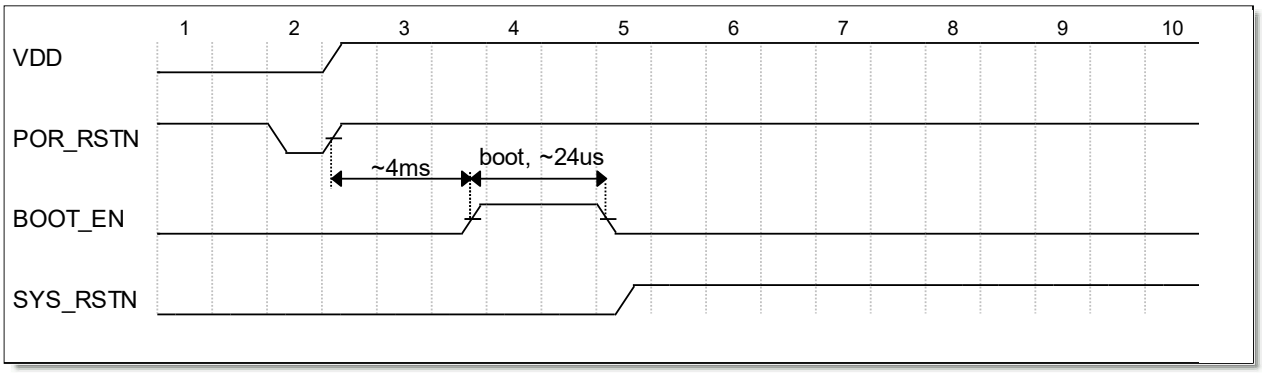


图 3-2 上电复位时序 1 (内部时钟模式，禁止 PWRT)

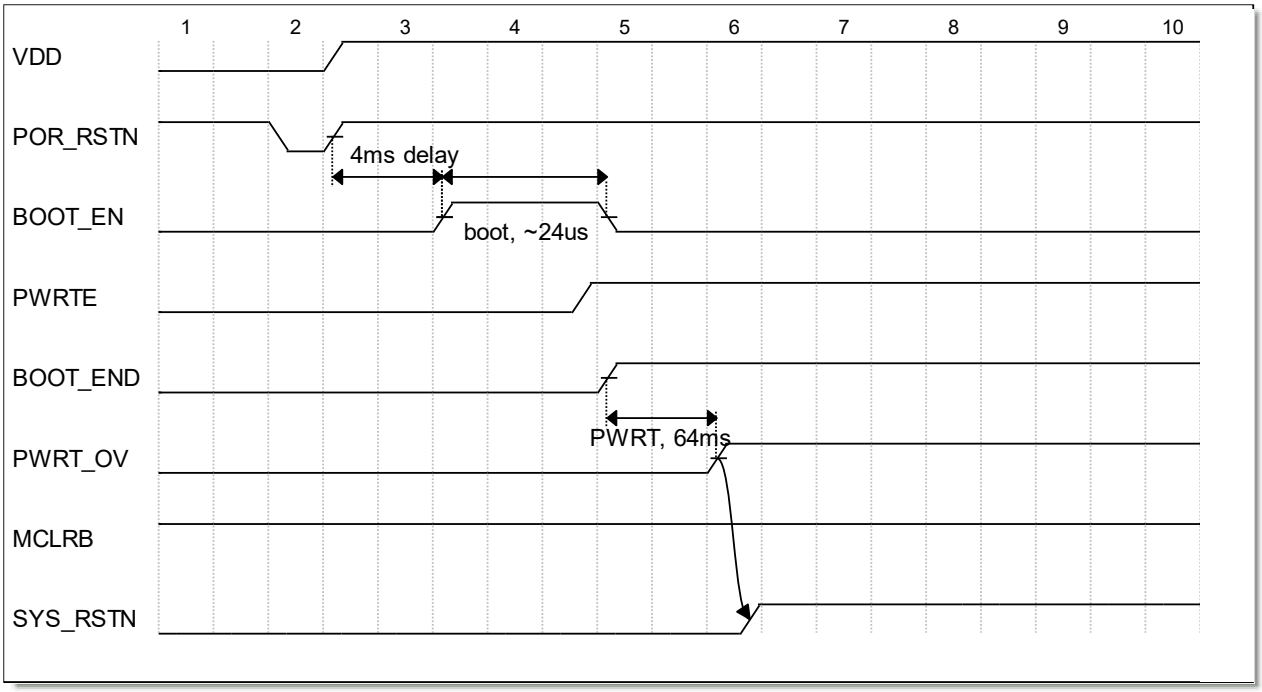


图 3-3 上电复位时序 2 (内部时钟模式，使能 PWRT)

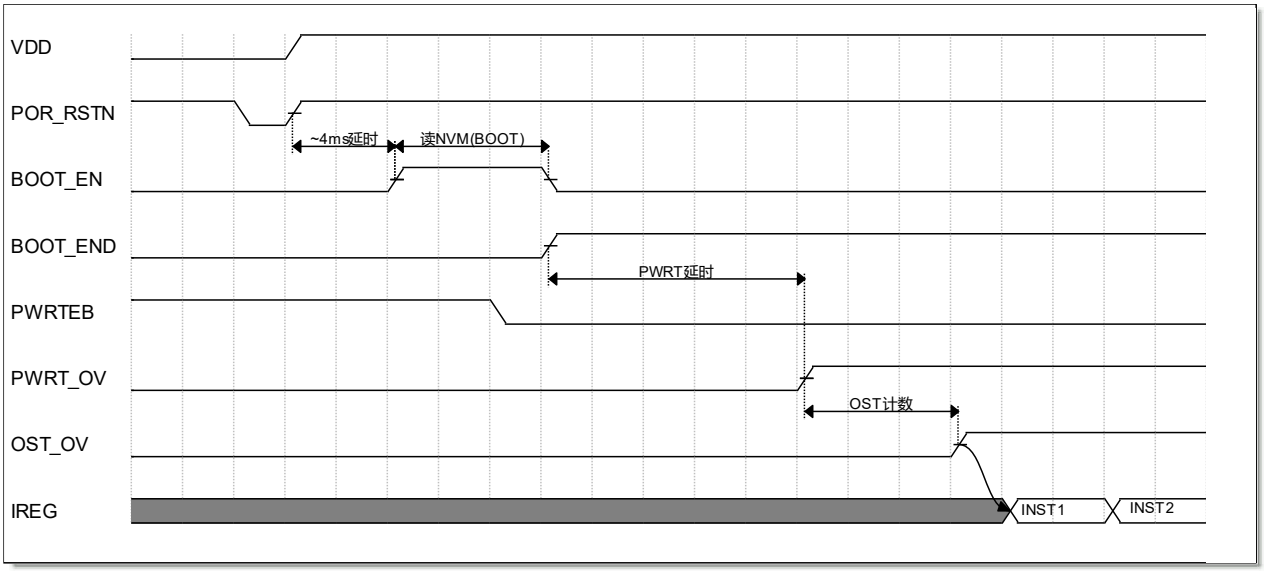


图 3-4 上电复位时序 3 (内部时钟模式，使能 PWRT，双速启动禁止)

3.3. 上电复位延时

复位模块内置了一个 11 位的上电复位定时器 PWRT 模块，它和 WDT 复用同一个计数器，它为上电复位和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 PWRT 溢出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。

可通过系统寄存器 (UCFG0) 来使能。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。

4. 系统复位

有以下 7 种复位源：

- 上电复位
- 低电压复位
- 看门狗复位
- 非法指令复位
- 软件复位
- EMC 复位
- 外部管脚复位
- 堆栈溢出复位

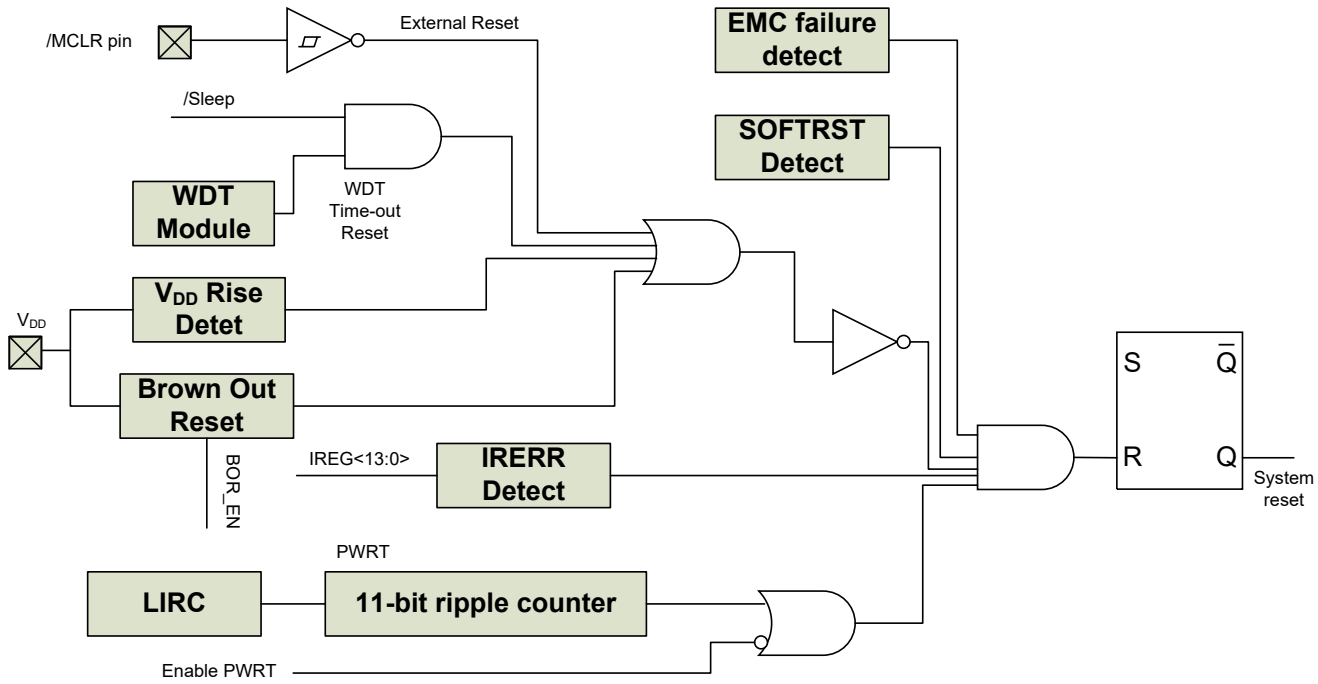


图 4-1 复位框图

4.1. 系统复位相关寄存器汇总

系统复位的大多数设置均由 IDE 界面配置，而不能通过指令修改。

名称	功能	默认
LVRS	7 档 V_{BOR} 电压(V)： 2.0 / 2.2 / 2.5 / 2.8 / 3.1 / 3.6 / 4.1	2.0
LVREN	LVR • 使能 • 关闭 • 非 SLEEP 模式下使能 • 通过指令控制 (SLVREN)	关闭
WDTE	WDT • 使能 (指令不能禁止) • 由指令控制 (SWDTEN)	SWDTEN 控制
MCLRE	外部 I/O 复位功能	关闭

表 4-1 复位相关初始化配置寄存器

4.2. 低电压复位

低电压复位由 UCFG1[1:0]位和 SLVREN 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 T_{BOR} (3~4 个慢时钟周期) 时间时，低电压复位可能不会发生。

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。

当 UCFG1[1:0] = 10 时，BOR 电路关闭将由 CPU 的运行状态决定：CPU 正常工作时 BOR 电路工作，CPU 处于睡眠模式时 BOR 电路自动关闭，这样可以方便的使系统功耗降至更低水平。

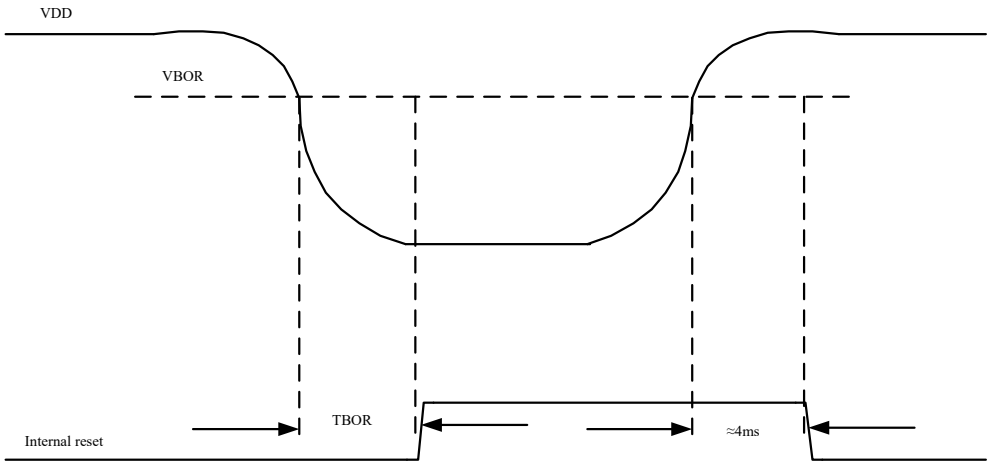


图 4-2 欠压复位

注意：

1. T_{BOR} 时间约为 94~125μs；
2. 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。

名称	状态	寄存器	地址	复位值
SLVREN [1]	仅适用于LVREN配置成由指令SLVREN控制LVR 1 = 使能 LVR 0 = 关闭 LVR	LVDCON0[7]	0x199	RW-0

表 4-2 LVR 用户寄存器

4.3. 非法指令复位

当 CPU 的指令寄存器取指到非法指令 (未定义的操作码) 时，标志位 IERRR (PCON.4) 将被置位，同时系统将进行复位。利用此功能可增加系统的抗干扰能力。

4.4. 软件复位

增强型内核实现了一条软件复位指令，助记符为 RESET，它提供给软件执行硬件复位的方法，复位标志为 SRSTF (PCON.2)。

4.5. EMC 复位

配置区有两个冗余寄存器DCFG0和DCFG1，上电复位值为0x5A，DCFG0在配置过程中它被位于NVM区的0x8047单元低8位覆盖，DCFG1在配置过程中它被位于NVM区的0x807F单元低8位覆盖,如果它们的值不等于0x5A，则EMC硬件检测逻辑会发出一次复位，标志位EMCF置1，重启上电配置过程 (读NVM区)，直到读到正确的值为止 (处于调试模式时，该功能自动禁止)。
在程序运行过程中，EMC 检测模块一直判断 DCFG0 和 DCFG1 的值，当发生某种 EMC 干扰导致其值发生变化时， (DCFG0≠0x5A 或者 DCFG1≠0x5A) 也将引发 EMC 复位，重启读 NVM 区的过程。

4.6. 上电配置过程(BOOT)

发生上电复位、低电压复位后，除了固有的 4ms 复位延时外，还有一个初始化配置寄存器 UCFGx 的动作。该动作从 PROM 的保留地址读取内容写到 UCFGx，待所有配置地址读取完成后，才可以释放系统复位，如**图 3-2** 和**图 3-3** 的所示，该过程大概需要 24μs。

可触发 BOOT 过程的复位源汇总

复位源	可触发 Boot
上电复位	√
低电压复位	√
EMC 复位	√
外部管脚复位	√ (仅 B 版芯片)
看门狗复位	–
非法指令复位	–
堆栈溢出复位	–

[1] 发生欠压复位时，此位不会清 0。其它任何复位会将此位清 0。

4.7. 复位源标志位

名称	状态		寄存器	地址	复位值
STKOVF [2]	堆栈上溢标志位		PCON[7]		R_W1C-0
STKUNF [2]	堆栈下溢标志位	1 = Yes	PCON[6]		R_W1C-0
EMCF [2]	EMC 复位标志位	0 = No，或已清 0	PCON[5]		R_W1C-0
IERRR [2]	非法指令复位标志位		PCON[4]		R_W1C-0
/MCLRR [3]	外部复位标志位		PCON[3]	0x96	RW1-1
/SRSTF [3]	软件复位标志位	1 = No (软件置 1)	PCON[2]		RW1-1
/PORF [3]	上电复位标志位	0 = Yes (锁存)	PCON[1]		RW1-0
/BORF [3]	低电压复位标志位		PCON[0]		RW1-x

表 4-3 复位标志寄存器

复位源	STKOVF	STKUNF	EMCF	IERRR	/MCLRR	/SRSTF	/PORF	/BORF	/TO	/PD
	PCON[7]	PCON[6]	PCON[5]	PCON[4]	PCON[3]	PCON[2]	PCON[1]	PCON[0]	STATUS[4]	STATUS[3]
	0x96								Bank 首地址 + 0x03	
POR	0	0	0	0	1	1	0	1	1	1
LVR	0	0	0	0	1	1	1	0	-	-
正常模式下(非 SLEEP) WDT 溢出(复位)	-	-	-	-	-	-	-	-	0	-
SLEEP 模式下 WDT 溢出(唤醒)	-	-	-	-	-	-	-	-	0	0
MCLR 复位 (≥ VerB)	0	0	0	0	1	1	0	1	1	1
MCLR 复位 (< VerB)	-	-	-	-	0	-	-	-	-	-
非法指令复位	-	-	-	1	-	-	-	-	-	-
堆栈上溢复位	1	-	-	-	-	-	-	-	-	-
堆栈下溢复位	-	1	-	-	-	-	-	-	-	-
软件复位	-	-	-	-	-	0	-	-	-	-
EMC 复位	-	-	1	-	-	-	-	-	-	-
CLRWDT 指令	-	-	-	-	-	-	-	-	1	1
SLEEP 指令	-	-	-	-	-	-	-	-	1	0
片上调试 (OCD)	-	-	-	-	-	-	-	-	-	-

表 4-4 复位相关状态标志位 ("- " 无变化)

[2] 写 1 清 0，写 0 无效。建议只使用 STR、MOVWI 指令进行写操作，而不要用 BSR 或 IOR 指令。
[3] 只能写 1，写 0 无效。

5. LVD 低电压侦测

除了低电压复位功能外，芯片还内置有低电压侦测功能。当电源电压低于设置的电压档位（由 LVDCON0 的 LVDL[2:0]选择）超过 T_{LVD}（3 到 4 个慢时钟周期）以上时，标志位 LVDW 将会被置 1，软件可以利用此位来监控电源电压。如果电源电压大于 LVDL 设置的电压档，该标志位会自动清除，换言之，LVDW 位不具有锁存功能。

5.1. LVD 相关寄存器汇总

名称	状态		寄存器	地址	复位值
LVDM	LVD 输入	1 = 外部管脚 (ELVDx) 0 = VDD	LVDCON0[6]	0x199	RW-0
LVDEN	LVD	1 = 使能 0 = 关闭	LVDCON0[4]		RW-0
LVDW	LVD 触发?	当 LVDP = 1 时: 1 = 检测电压 > $V_{LVD-REF}$ (不锁存) 0 = 检测电压 < $V_{LVD-REF}$ 当 LVDP = 0 时: 1 = 检测电压 < $V_{LVD-REF}$ (不锁存) 0 = 检测电压 > $V_{LVD-REF}$	LVDCON0[3]		RO-0
LVDL	$V_{LVD-REF}$	000 = 保留 100 = 2.8 001 = 保留 101 = 3.0 010 = 2.0 110 = 3.6 011 = 2.4 111 = 4.0	LVDCON0[2:0]		RW-000
LVDP	LVDW 极性	1 = 检测电压 > $V_{LVD-REF}$ 0 = 检测电压 < $V_{LVD-REF}$	LVDCON1[4]	0xF8E	RW-0
LVDDEB	LVD 消抖	1 = 使能 0 = 关闭	LVDCON1[3]		RW-1
LVDOE	LVD 输出	1 = 使能 0 = 关闭	LVDCON1[2]		RW-0
LVDOS	LVD 输出管脚	00 = PA5 10 = PB4 01 = PB5 11 = PB3	LVDCON1[1:0]		RW-00
LVDCAL	LVDL 修调位		LVDTUNE[7:4]	0xF92	RW-1000
ELVDS	LVD 外部输入管脚选择	00 = ELVD0 10 = ELVD2 01 = ELVD1 11 = ELVD3	ADCON3[1:0]	0x41A	RW-00
LVDIE	LVD 中断	1 = 使能 0 = 关闭	INTCON[4]	0x0B	RW-0
LVDIF [1]	LVD 产生中断?	1 = Yes 0 = No，或已被清零	INTCON[1]		R_W1C-0

表 5-1 LVD 用户设置和标志寄存器

[1] 写 1 清 0，写 0 无效。建议只使用 STR、MOVWI 指令进行写操作，而不要用 BSR 或 IOR 指令。

5.2. 检测外部电压

除了可以监控片内 VDD 外，LVD 模块还具备检测外部电压的功能。寄存器位 LVDM 决定了 LVD 作用于 VDD 还是外部电压，当它为 1 时表示对外部管脚 ELVDx 进行监控。外部 LVD 管脚一共有 4 种选择，由寄存器 ELVDS[1:0]控制。当配置为 ELVD 功能时，管脚斯密特输入被关闭以防漏电。

注意：PC0 的外部复位功能优先级高于外部 LVD 功能。当配置为外部复位管脚时，外部 LVD 的检测是无效的，不管 ELVDS 以及 LVDM 为何值。

5.3. LVD 中断

除了通过轮询 LVDW 位了解低电压侦测事件外，软件还可以通过中断的方式来获得低电压的情况。当低电压侦测事件发生后，LVDIF 位自动置 1，它是一个电平触发锁存器，只能通过软件清 0，清 0 的前提是电源电压恢复到 LVDL[2:0]设置的水平以上，LVD 事件结束。

当 PEIE 和 LVDIE 被置 1 且 LVDIF 为 1 时，睡眠状态下 LVD 中断标志位还可以作为一个唤醒源，如果之前 GIE = 1，则唤醒后 CPU 进入中断处理。

5.4. LVD 电平输出

除了使能 LVDOE 从 PA5 输出 LVD 电平外，软件还可以通过配置 LVDCON1 的 LVDOS 来改变电平输出的管脚。此外，用户也可以通过配置 LVDDEB 位来选择 LVD 电平是否需要经过内部的去抖电路输出到管脚上。

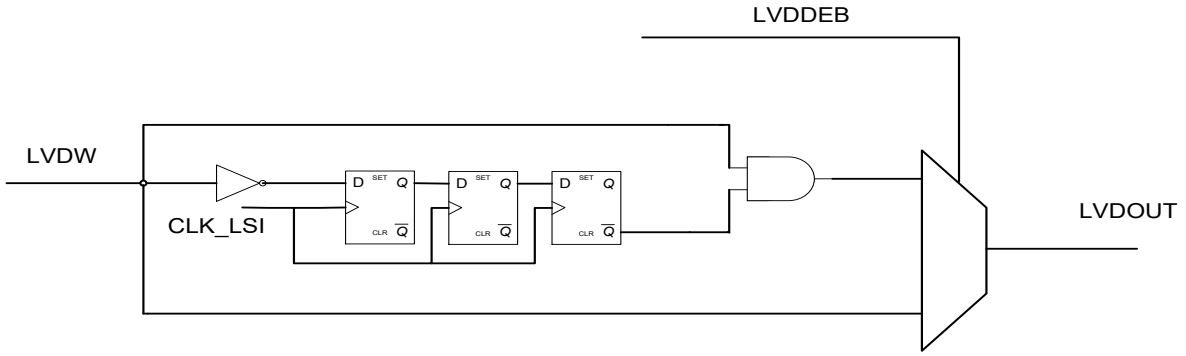


图 5-1 LVD 输出控制电路

6. 振荡器和系统时钟

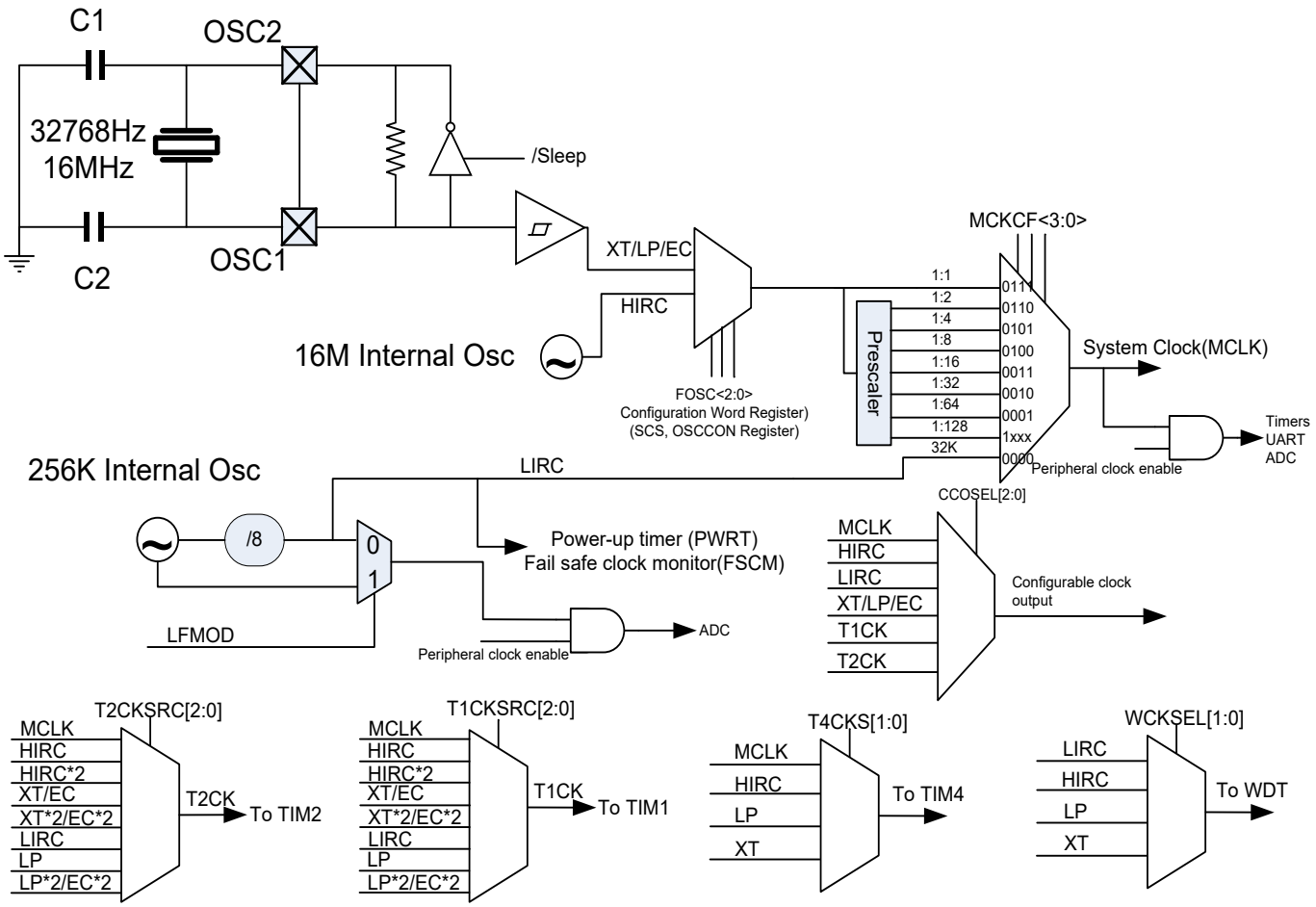


图 6-1 系统时钟源框图

时钟源包含 4 个时钟源：2 个内置振荡器，1 个外部晶体振荡器，1 个外部时钟灌入源。内置振荡器包括 1 个内部 16M 高速精准振荡器 (HIRC)，1 个内部 32k/256k(LIRC)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。同时内部高速振荡器可以通过 OSCTUNE 寄存器对振荡器的频率进行调节校准。

6.1. 振荡器模块相关寄存器汇总

名称	功能		默认
FOSC	• LP：PC1 (+) 和 PB7 (−) 接外部低速晶振 • XT：PC1 (+) 和 PB7 (−) 接外部高速晶振 • EC：PC1 (+) 接外部时钟输入，PB7 为 I/O • INTOSCIO：PC1 和 PB7 为 I/O		INTOSCIO
IESO	XT / LP 双速时钟启动	• 使能 • 关闭	使能
FSCMEN	故障保护时钟监控器	• 使能 • 关闭	使能
TSEL	指令时钟与系统时钟的对应关系 (2T or 4T) • 2 (指令时钟 = SysClk/2) • 4 (指令时钟 = SysClk/4)		2
OSTPER	OST 定时器周期选择 (XT / LP 适用)	• 512 • 1024 • 2048 • 4096 (LP 模式时为 32768)	1024

表 6-1 FOSC 和双速启动初始化配置寄存器

SysClk 系统时钟源			配置			
			SCS	MCKCF	LFMOD	OST
			OSCCON[0]	OSCCON[7:4]	TCKSRC[7]	(可选，参阅 OSTPER)
			0x99			
			RW-0	RW-0100	RW-0	
外部	EC		0	-	-	-
	XT		0	-	-	1,024 (默认值)
	LP		0	-	-	1,024 (默认值)
内部	HIRC	16 MHz	1	0111	-	-
		8 MHz	1	0110	-	-
		4 MHz	1	0101	-	-
		2 MHz	1	0100	-	-
		1 MHz	1	0011	-	-
		500 kHz	1	0010	-	-
		250 kHz	1	0001	-	-
		125 kHz	1	1xxx	-	-
	LIRC	256 kHz [1]	1	0000	1	-
		32 kHz [2]	1	0000	0	-

表 6-2 SysClk 系统时钟源设置相关用户寄存器

[1] 256 kHz LIRC 只供 ADC (参阅 ADCS 和 LFMOD，**表 15-1**) 使用。

[2] 系统时钟源 (MCKCF=000)、PWRT、FSCM 和 WDT (WCKSEL =00)统一使用 LIRC 的 8 分频，即 32 kHz，而不管 LFMOD 为何值。

名称	状态		寄存器	地址	复位值
OSTS	振荡器启动超时状态位(锁存) 1 = 运行在外部振荡器下(启动成功) 0 = 运行在内部振荡器下		OSCCON[3]	0x99	RO-0
HTS	HIRC ready (锁存)	1 = Yes 0 = No	OSCCON[2]		RO-0
LTS	LIRC ready (锁存)	1 = Yes 0 = No	OSCCON[1]		RO-0
SYSON	Sleep 模式下，系统时钟控制 1 = 保持运行 0 = 关闭		CKOCON[7]	0x95	RW-0
CCORDY	时钟输出标志位	1 = Yes 0 = No	CKOCON[6]		RO-0
DTYSEL	TIM1/TIM2 倍频时钟占空比调节位 00 = 2ns 延迟 10 = 4ns 延迟 01 = 3ns 延迟 11 = 7ns 延迟		CKOCON[5:4]		RW-10
CCOSEL	输出时钟选择位 000 = Sysclk 100 = T1CK 001 = HIRC 101 = T2CK 010 = LIRC 110 = LP 011 = XT 111 = EC		CKOCON[3:1]		RW-000
CCOEN	时钟输出	1 = 使能 0 = 禁止	CKOCON[0]		RW-0
CLKOS [3]	时钟输出引脚 1 = CLKO 映射到 PB1 0 = CLKO 映射到 PA7		CKAUX[1]	0xF8D	RW-1
TUN	内部高速时钟 HIRC 频率调节寄存器		OSCTUNE[7:0]	0x98	RW-xxxx xxxx
UARTEN	USART 模块时钟	1 = 使能 0 = 禁止	PCKEN[5]	0x9A	RW-0
TIM4EN	Timer4 模块时钟		PCKEN[3]		RW-0
TIM2EN	Timer2 模块时钟		PCKEN[2]		RW-0
TIM1EN	Timer1 模块时钟		PCKEN[1]		RW-0
ADCEN	ADC 模块时钟		PCKEN[0]		RW-0

表 6-3 振荡器控制位/状态位

[3] 该寄存器写之前需要把 EECON1.CFGS 置 1。

名称	状态		寄存器	地址	复位值
GIE	全局中断	1 = 使能 (PEIE, OSFIE 适用) 0 = 全局关闭 (唤醒不受影响)	INTCON[7]	Bank 首地址 +0x0B	RW-0
PEIE	外设总中断	1 = 使能 (OSFIE 适用) 0 = 关闭 (无唤醒)	INTCON[6]		RW-0
OSFIE	外部振荡器故障中断	1 = 使能 0 = 关闭 (无唤醒)	INTCON[3]		RW-0
OSFIF	外部振荡器故障标志位	1 = Yes (锁存) 0 = No	INTCON[0]		R_W1C-0

表 6-4 振荡器中断使能/状态位

6.2. 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) 来选择内部高速或者外部时钟源。

6.3. 外部时钟模式

6.3.1. 振荡器起振定时器 (OST)

如果振荡器模块配置为 LP、XT 模式，振荡器起振定时器 (OST) 根据配置字 OSTPER[1:0]对来自 OSC1 的振荡计数。用户可以根据不同应用要求，通过设置 OSTPER 位调节 OST 计数的次数。这发生在上电复位 (POR) 之后或上电延时定时器 (PWRT) 延时结束 (如果被使能) 时，或从休眠中唤醒后，或故障保护条件清除后。在此期间，程序计数器不递增，程序执行暂停。OST 确保使用石英晶体谐振器或陶瓷谐振器的振荡器电路已经启动并向振荡器模块提供稳定的系统时钟信号。当在时钟源之间切换时，需要一定的延时以使新时钟稳定。

注意：OST 复用了 WDT 定时器，故在 OST 对晶体时钟计数时，WDT 功能被屏蔽，待 OST 发生溢出后，WDT 功能才恢复 (如果此前 WDT 被使能的话)。当系统时钟切换到 LP 或者 XT 模式时，看门狗计数器会被清零。

6.3.2. 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，OSC2 引脚可用作通用 I/O。

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。

6.3.3. 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，模式选择内部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。该模式设计仅用于驱动 32.768 kHz 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。

6.3.4. 内部时钟模式

振荡器模块有两个独立的内部振荡器，可配置或选取为系统时钟源。

1. HIRC (高频内部振荡器) 出厂时已校准，工作频率为 16MHz。

2. LIRC (低频内部振荡器) 未经校准，工作频率为 32 kHz。软件对 OSCCON 寄存器的系统时钟选择位 MCKCF[3:0]进行写操作，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，在外部或内部高速时钟源之间选择系统时钟。

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

6.3.5. 频率选择位 (MCKCF)

外部晶体时钟，16MHz HIRC 和 32kHz LIRC 的输出连接到预分频器和多路复用器（见**图 6-1**）。OSCCON 寄存器的内部振荡器频率选择位 MCKCF[3:0] 用于选择不同的分频输出。可选择以下各分频比或频率：

- 1:1
- 1:2
- 1:4
- 1:8
- 1:16 (复位后的缺省值)
- 1:32
- 1:64
- 1:128
- 32 kHz (LIRC)

6.3.6. HIRC 和 LIRC 时钟切换时序

当在 LIRC 和 HIRC 之间切换时，新的振荡器可能为了省电已经关闭 (见图 6.2 和图 6.3)。在这种情况下，OSCCON 寄存器的 MCKCF 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 LIRC 和 HIRC 振荡器的当前活动状态。频率选择时序如下：

- OSCCON 寄存器的 MCKCF[3:0]位被修改
- 如果新时钟是关闭的，会有一个时钟的启动延时
- 时钟切换电路等待当前时钟的 2 个下降沿的到来
- CLKOUT 保持为低，时钟切换电路等待两个新时钟下降沿的到来
- 现在 CLKOUT 连接到新时钟。OSCCON 寄存器的 HTS 和 LTS 位被更新
- 时钟切换完成

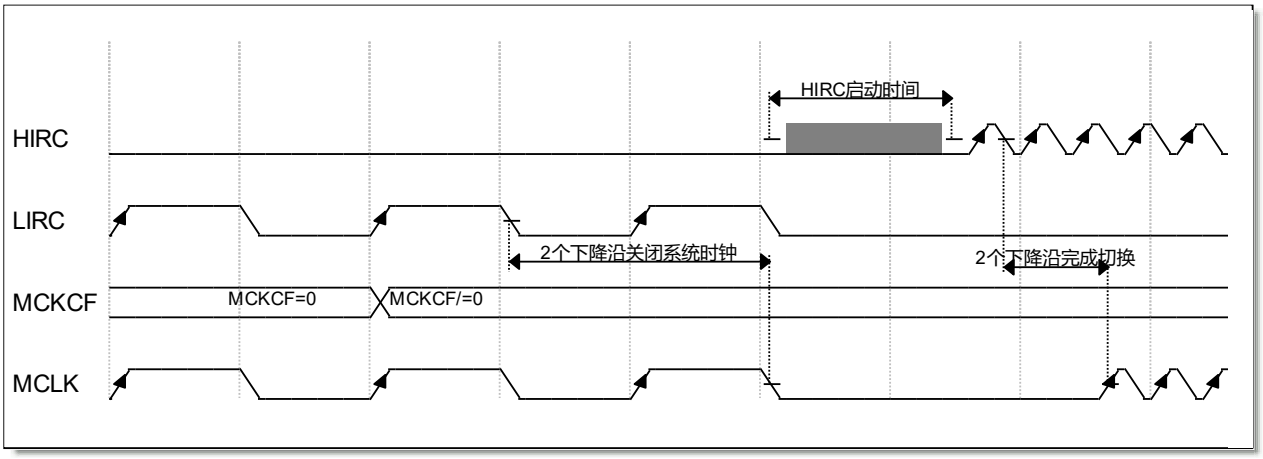


图 6-2 由慢时钟切换到快时钟

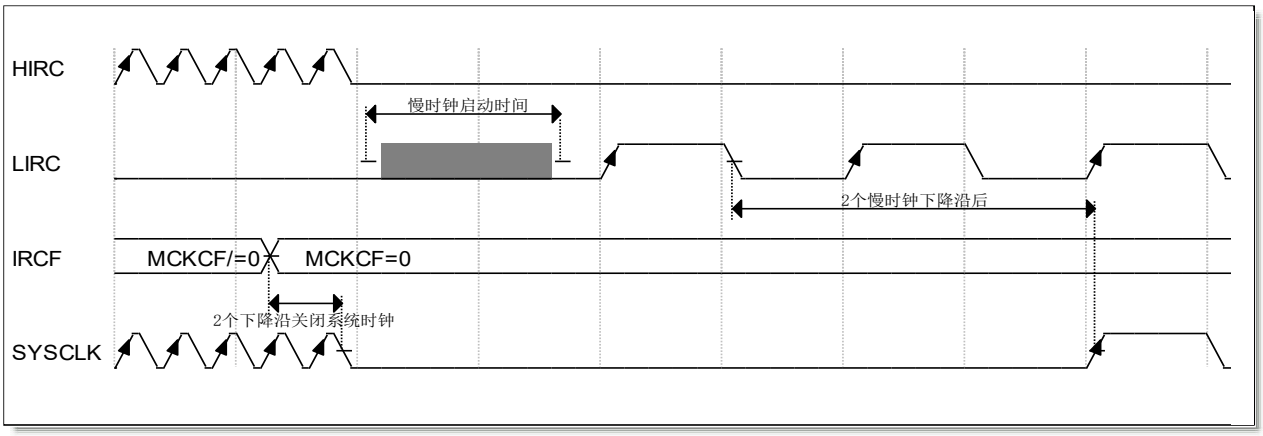


图 6-3 由快时钟切换到慢时钟

6.3.7. HIRC 时钟特殊功能

FT61F14X 内建了一个振荡频率可调的高精度 HIRC 作为系统时钟，出厂时被精确地调校至 16MHz@5V/25℃，用户可以通过编程器的 Code Option 校准系统时钟。校准过程是过滤掉制程上的偏差对精度造成的影响。此 HIRC 受工作的环境温度和工作电压影响会有一定的漂移，对于压漂(4.5V~5.5V)以及 (-20℃~85℃) 的温漂一般状况会在±1%以内。

FT61F14X 有一个特殊的功能：用户可修改 OSCTUNE 的值来对 HIRC 频率作调整。

OSCTUNE 的值确保 HIRC 在上电后准确工作在 16MHz。此数值的初始值每颗 IC 都会有差异。初始值为 OSCTUNE[s]，此时芯片工作在 16MHz，每改变 1 个 LSb 则 HIRC 频率变化约为 40kHz。OSCTUNE[7:0]和 HIRC 输出的关系如下：

OSCTUNE[7:0]值	HIRC 实际输出频率 (16M 为例),单位 kHz
OSCTUNE[s]-n	(16000-n*40)
…..	…..
OSCTUNE[s]-2	16000-2*40=15920
OSCTUNE[s]-1	16000-1*40=15960
OSCTUNE[s]	16000
OSCTUNE[s]+1	16000+1*40=16040
OSCTUNE[s]+2	16000+2*40=16080
…..	…..
OSCTUNE[s]+n	(16000+n*40)

6.4. 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高速时钟源之间切换。

6.4.1. 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，忽略 FOSC[2:0]位，根据 OSCCON 寄存器的 MCKCF[3:0]位决定系统时钟源：HIRC 的分频时钟或者 32k 时钟。

注：

1. 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；
2. 当 MCKCF[3:0]等于 0 时，无论 SCS 为何值，系统时钟都选择内部慢时钟。

6.4.2. 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。

6.4.3. 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：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**章节 6.3.1** “振荡器起振定时器”)。OST 将暂停程序执行，直到完成配置字 OSTPER[1:0]位要求的计数次数。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 OSTPER[1:0]位要求的计数次数且 OSCCON 寄存器的 OSTS 位置 1 时，程序执行切换至外部振荡器。

注：

1.系统时钟配置为外部晶振模式时，同时使能了双速模式，在 OST 未计数到 OSTPER 要求的数值时，CLRWDT 指令不能清除看门狗计数器，也就是说此时 OST 计数不能被中断；

2.系统时钟配置为外部晶振模式时，同时使能了双速模式，在 OST 未计数到 OSTPER 要求的数值时，执行 sleep 指令时，看门狗计数器被清零，此时的 OST 计数被清零。

6.4.4. 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； (使能双速启动模式)
- OSCCON 寄存器的位 SCS = 0 (内部/外部切换位)
- 配置字寄存器 (UCFG0) 中的 FOSC[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，外部时钟振荡器不需要稳定时间。

6.4.5. 双速启动顺序

- 从上电复位或休眠中唤醒
- 使用内部振荡器以 OSCCON 寄存器的 MCKCF[3:0]位设置的频率开始执行指令
- OST 使能，计数 OSTPER[1:0]位要求的计数次数
- OST 超时，等待内部振荡器下降沿的到来
- OSTS 置 1
- 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)
- 系统时钟切换到外部时钟源

6.4.6. 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 FSCMEN 位置 1 来使能。FSCM 可用于所有外部振荡模式 (LP、XT 和 EC)。

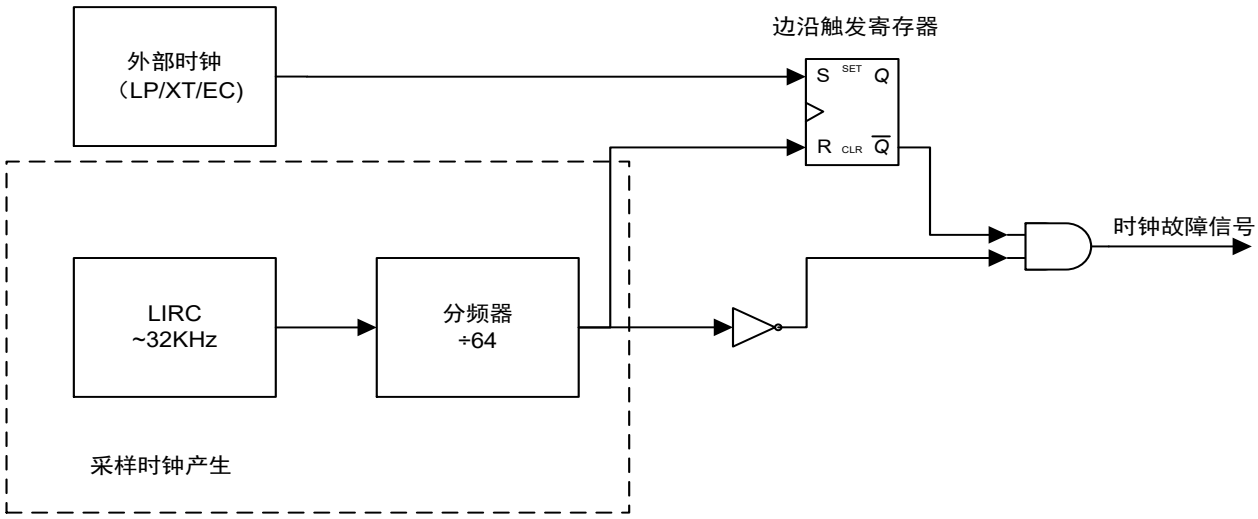


图 6-4 FSCM 原理框图

6.4.7. 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。LIRC 除以 64，就产生了采样时钟。请参见图 6-4。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。

6.4.8. 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。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 MCKCF[3:0]位决定。这使内部振荡器可以在故障发生前就得以配置。

6.4.9. 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。

6.4.10. 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：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。

6.5. 外设时钟门控

关闭未使用外设的时钟可降低功耗。外设的时钟门控模式可以对以下外设的时钟随时打开或者关闭：

- ADC
- USART
- TIM1/2/4

系统复位后，所有的外设时钟均处于关闭的状态。用户可以通过配置 PCKEN 寄存器对应的位来打开相应的外设时钟。但是如果需要禁止某个外设，必须在其时钟被停止之前进行。

如果要使能某个外设，则需要先使能对应外设的时钟，然后再使能对应外设。外设时钟控制只是控制其模块的系统时钟。

TIM1/2/4 计数时钟也受 PCKEN 寄存器控制，当 PCKEN 寄存器对应的未置 1 时，则 TIMER 的计数时钟和寄存器操作时钟同时打开。

注意：

1. 在睡眠模式下，当 SYSON=1 时，无论 TIM1/2 的时钟源是否为系统时钟，其时钟源都会打开；当 SYSON=0 时，TIM1/2 时钟源关闭。
2. ADC 的转换时钟源打开或者关闭与系统进入睡眠模式无关。
3. 慢时钟测量时，慢时钟的打开与关闭也与系统时钟进入睡眠模式无关。

6.6. 时钟输出

可以将芯片内部的时钟源输出到芯片的 CLKO 管脚上，可以选择以下几种时钟源输出：

- 系统时钟
- 内部高速时钟
- XT 晶振时钟
- LP 晶振时钟
- 外部时钟
- 内部慢时钟
- TIM1/TIM2 时钟

时钟输出寄存器控制/状态位说明

时钟的输出选择和输出控制位于 CKOCON 寄存器。

当时钟正在输出时，CCORDY 被硬件置 1，如果 CCOBSY 被置为 1，则 CCOSEL 位处于保护状态，不能被写入。

当输出某个时钟时，对应的时钟源也同时被打开。通过清零 CCOEN 位禁止时钟的输出。时钟输出关闭动作完成之后，CCORDY 位才能被硬件清零。

时钟输出注意

1. 在睡眠模式下，CCO 的输出与 SYSON 相关，当 SYSON=1 时，CCO 继续输出进入睡眠之前所选

的时钟；当 SYSON=0 时，CCO 暂停输出时钟，当系统退出睡眠状态后，继续输出所选时钟；

2. 当系统时钟为 XT 模式时，即使输出时钟选择了 LP 时钟，输出时钟也是 XT 时钟；
3. 当系统时钟为 LP 模式时，即使输出时钟选择了 XT 时钟，输出时钟也是 LP 时钟；
4. 当 FOSC[2:0]选择为内部时钟源时，输出时钟选择了 LP、XT 或者 EC 时钟，输出时钟端口则不会输出时钟；
5. MCU 处于工作模式时，被选择为时钟输出的时钟源将自动使能，例如当前系统时钟为 HSI，时钟输出选择为 XT 晶振时钟且 CCOEN 为 1，则对应的 XT 晶振电路打开，相关管脚也变成 OSC1, OSC2 功能。
6. UCFG0 配置选项的优先级要高于 CKOCON 寄存器的优先级，例如配置选项选择 XT 作为系统时钟，时钟输出配置 LP 或 EC 是不起作用的，反之亦然。
7. 时钟可以选择在 PB1 或者 PA7 输出，由寄存器 CKAUX.CLKOS 位控制。跟普通寄存器不一样，该寄存器的写需要执行以下步骤，否则无法写入：
   (1) 对 EECON1.CFGS 置 1；
   (2) 往 CKAUX 写入目标值；
   (3) 对 EECON1.CFGS 清 0

7. 慢时钟测量

芯片集成了两个内部 RC 振荡器，一个是经过出厂校准的高速高精度的 16M 快时钟 HIRC，一个是低速低功耗的 32k 时钟 LIRC，利用慢时钟测量功能可以把 LIRC 的周期用系统时钟计算出来。此功能可以比较精准地测量内部慢时钟周期。

7.1. 慢时钟测量相关寄存器汇总

名称	状态	寄存器	地址	复位值
CKMAVG	LIRC 和 HIRC 交叉校准时 4 次平均测量模式 1 = 使能 0 = 关闭	MSCKCON[1]	0x41D	RW-0
CKCNTI	启动 LIRC 和 HIRC 的交叉校准功能 1 = 启动 0 = 完成 (自动清零)	MSCKCON [0]		RW-0
SOSCPR	校准 LIRC 周期所需的 HIRC 周期数	SOSCPR[11:0]	0x41F[3:0] 0x41E[7:0]	RW-FFF

表 7-1 LIRC 和 HIRC 交叉校准控制/状态位

名称	状态		寄存器	地址	复位值
GIE	全局中断	1 = 使能 (PEIE, CKMIE 适用) 0 = 全局关闭 (唤醒不受影响)	INTCON[7]	Bank 首地址 +0x0B	RW-0
PEIE	外设总中断	1 = 使能 (CKMIE 适用) 0 = 关闭 (无唤醒)	INTCON[6]		RW-0
CKMIE	LIRC 和 HIRC 交叉校准完成中断	1 = 使能 0 = 关闭 (无唤醒)	PIE1[1]	0x91	RW-0
CKMIF [1]	LIRC 和 HIRC 交叉校准完成标志位	1 = Yes (锁存) 0 = No	PIR1[1]	0x11	R_W1C-0

表 7-2 LIRC 和 HIRC 交叉校准中断使能/状态位

[1] 写 1 清 0，写 0 无效。建议只使用 STR、MOVWI 指令进行写操作，而不要用 BSR 或 IOR 指令。

7.2. 测量原理

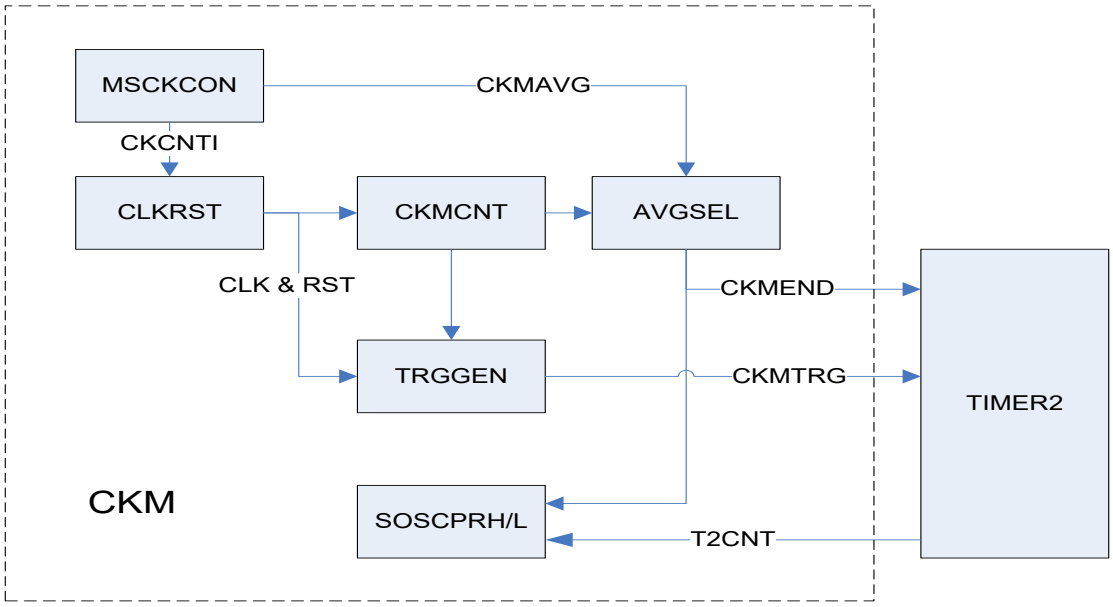


图 7-1 慢时钟测量模式原理框图

慢时钟测量类似于定时器的捕捉模式，处于这种模式下，被测量时钟 LIRC 的边沿（任意沿）将会触发定时器，在另一高速时钟（如 HIRC）的作用下开始计数，在此后的第 2 个（或第 8 个，平均模式时）LIRC 边沿到来时，定时器停止计数，同时把定时器的值锁存到 SOSCPRL/H 寄存器。

慢时钟测量使用的定时器是 TIM2。

注意：

1. 在慢时钟测量过程中软件不要写 SOSCPRH/L；
2. 不要在单步调试下做慢时钟测量，因为暂停模式下 TIM2 被停止，这样会导致测量结果不正确；
3. 若 SYSON = 0 时，慢时钟测量无法在 SLEEP 模式下进行，不要在测量运行时进入 SLEEP 模式。

7.3. 上电自动测量

在上电后慢时钟测量将会自动启动，此时 CKCNTI 置 1，CKMAVG 为 0，打开 LIRC 和 HIRC。TIM2 的时钟被自动配置为 16M 的内部高速时钟，即类似设置 T2CKSRC 为 001、TIM2EN=1 的功能，但未配置这些位。TIM2 使用默认配置，无需置位 CEN 使能 TIM2 计数，此时不能配置 TIM2。

在测量过程中，实际应用程序已在运行，若要使用 TIM2 则需要查询 CKCNTI。若 CKCNTI 为 0 即可使用 TIM2，此时 SOSCPR 寄存器的值为有效值，其单位为 FHSI 时钟的个数。

注：上电自动测量不会置位 CKM 中断标志。

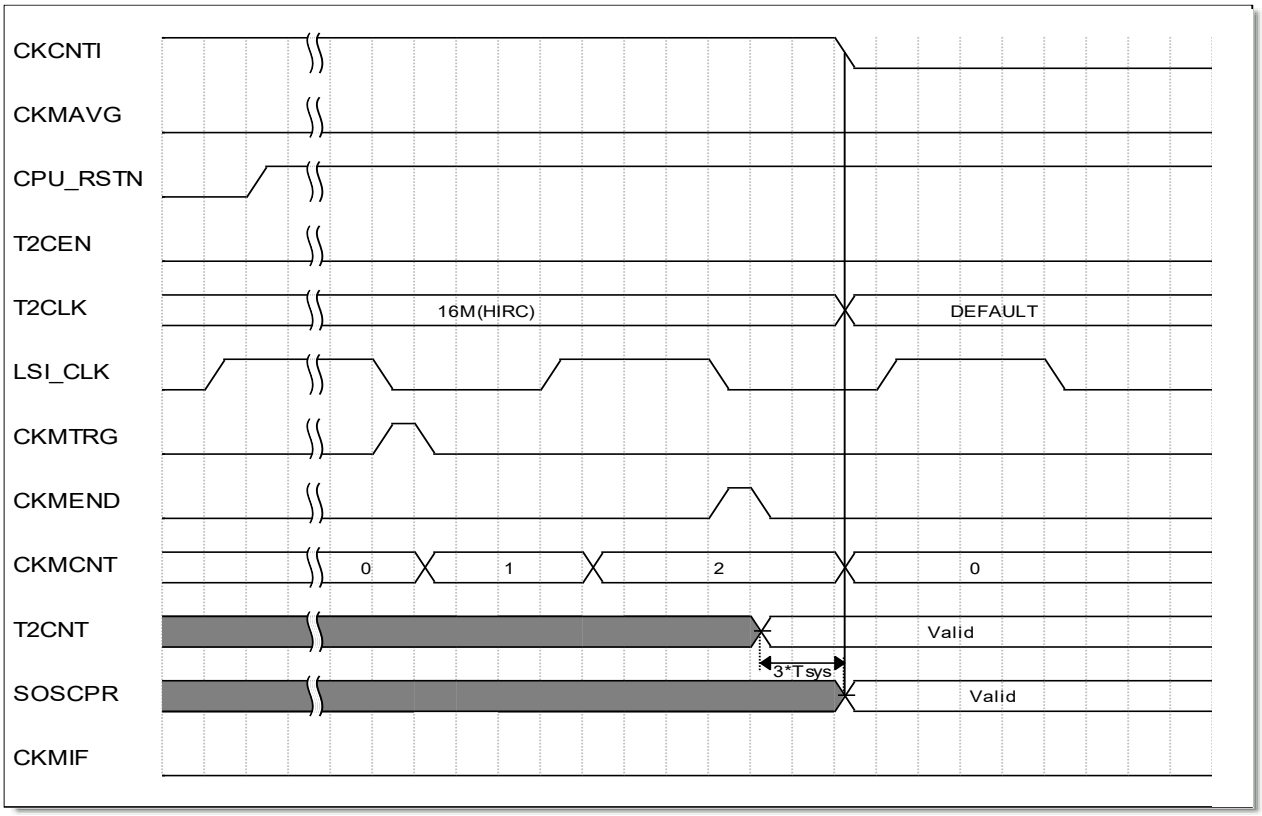


图 7-2 上电慢时钟自动测量时序图

7.4. 操作步骤

1. 为提高计量精度，建议设置 T2CKSRC 为 001，TIM2EN=1，选择 16M 的内部高速时钟；
2. 关闭 TIM2 的相关中断使能，设置 TIM2ARRH/L 为最大值，设置 TIM2PSC 为 0000；
3. 设置 TIM2CR1 为复位值，再将 CEN 置 1，使能 TIM2；
4. 如果选择 4 次平均，则把 MSCKCON.1 置 1，否则把它清 0；
5. 置位 MSCKCON.0，开始测量；
6. 测量结束后 MSCKCON.0 自动清 0，中断标志置 1；
7. 可以用查询或中断的方式等待结束；
8. 当查询到中断标志为 1 时读取得到的 SOSCPR 即为最终结果。

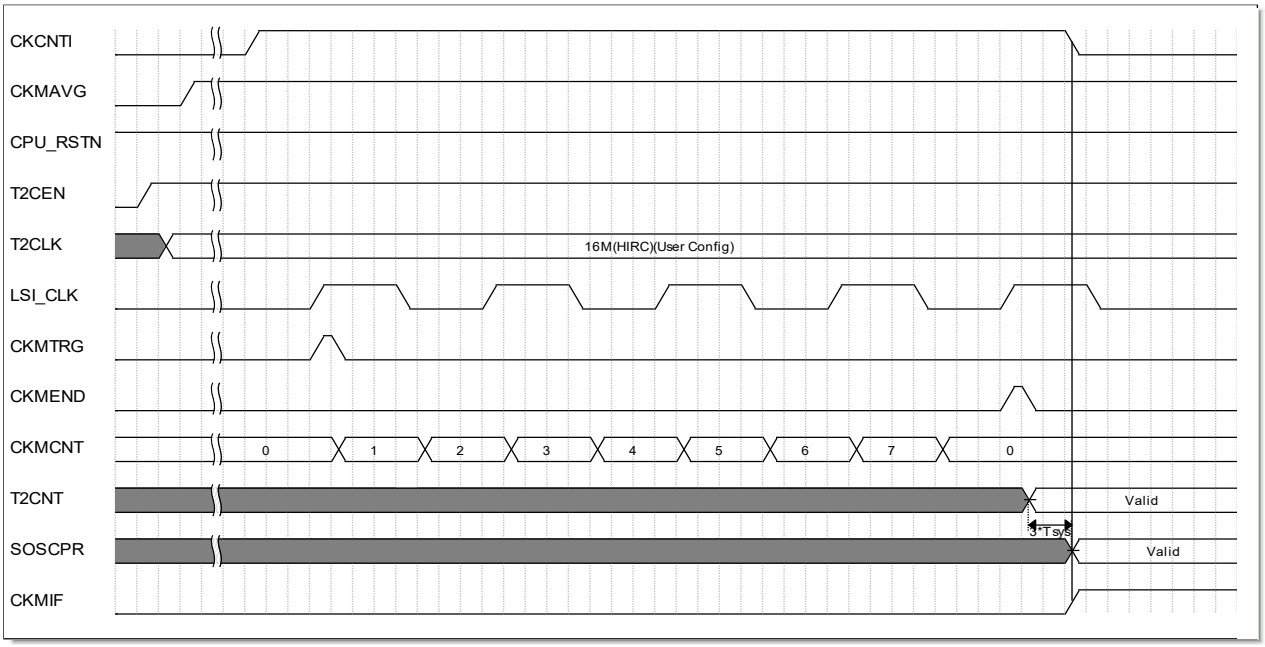


图 7-3 慢时钟测量模式时序图

8. 看门狗定时器

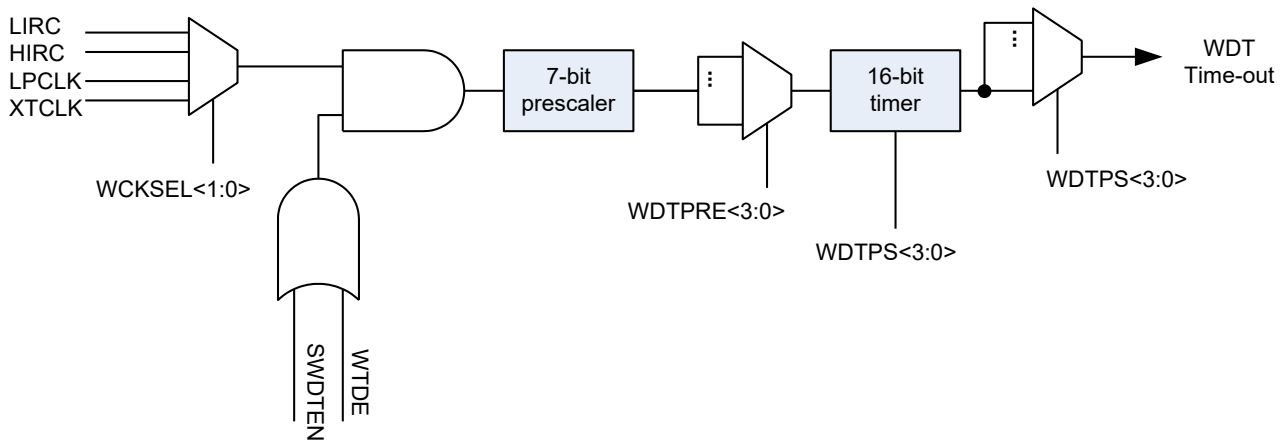


图 8-1 看门狗结构框图

看门狗是一个带 7 位预分频的 16 位计数器，其中预分频和周期可编程，分别由 WDTPRE 和 WDTPS 设置。

WDT 的硬件使能位位于配置寄存器 UCFG0 的第 3 位，WDTEN，软件使能位位于 WDTCON 寄存器的第 0 位，为 1 时表示使能看门狗，为 0 时禁止。

指令 CLRWDT、SLEEP 会清除看门狗计数器。

在使能了看门狗的情况下，处于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 WDT 则是作为一个复位源。

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	
写 WDTCON	
写 WCKSEL	

注意：

1. 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，由 LFMOD 位控制)，都不影响看门狗计时。因为 WDT 时钟源选择 LIRC 时，固定使用 32k 时钟源，见**图 6-1** 的时钟框图；
2. PWRT 和 OST 复用了 WDT 定时器，故 PWRT 或 OST 工作时，看门狗的复位功能是暂时屏蔽的；

8.1. 看门狗时钟源

WDT 有 4 种时钟源可选，由寄存器 MISC0 的 WCKSEL 位设置。在 WDT 使能的情况下，所选择的时钟源被自动使能，并在 SLEEP 模式下保持。

8.2. WDT 相关寄存器汇总

名称	状态	寄存器	地址	复位值
WCKSEL	WDT 时钟源 00 = LIRC 01 = HIRC 10 = LP (仅当 FOSC 为 LP 或 INTOSCIO 模式*) 11 = XT (仅当 FOSC 为 XT 或 INTOSCIO 模式*) *否则配置错误，无 WDT 时钟源	MISC0[1:0]	0x19D	RW-00
WDTPRE	WDT 预分频器 000 = 1　100 = 16 001 = 2　101 = 32 010 = 4　110 = 64 011 = 8　111 = 128 (默认)	WDTCON[7:5]		RW-111
WDTPS	WDT 周期 0000 = 32　0111 = 4,096 0001 = 64　1000 = 8,192 0010 = 128　1001 = 16,384 0011 = 256　1010 = 32,768 0100 = 512 (默认)　1011 = 65,536 0101 = 1,024　11xx = 65,536 0110 = 2,048	WDTCON[4:1]	0x97	RW-0100
SWDTEN	1 = WDT 使能 0 = WDT 关闭 (当 WDTE 选择由 SWDTEN 控制时)	WDTCON[0]		RW-0

表 8-1　WDT 相关用户寄存器

名称	功能	默认
WDTE	WDT • 使能 (指令不能禁止) • 通过指令控制 (SWDTEN)	SWDTEN 控制

表 8-2　WDT 选项初始化配置寄存器

9. 高级定时器 TIM1

9.1. 特性

- 16bit 的向上计数、向下计数或者上/下计数器，支持自动重载；
- 支持可编程预分频的计数时钟；
- 支持 4 个独立的捕捉比较通道，通道可支持：
  - ✓ 输入捕捉
  - ✓ 输出比较
  - ✓ 边沿或中心对称 PWM
  - ✓ 单脉冲输出
  - ✓ 6 步 PWM
- PWM 互补输出和可编程死区时间；
- 可编程的重复计数器；
- 刹车功能，使输出停止在一个复位态或者一个预设状态
- 中断事件：
  - ✓ 更新事件：计数器溢出，计数器初始化
  - ✓ 触发事件：触发计数开始与停止，计数器初始化或外部触发事件
  - ✓ 输入捕捉事件
  - ✓ 输出比较事件
  - ✓ 刹车输入有效事件
- 外部时钟的触发计数
- 前沿消隐

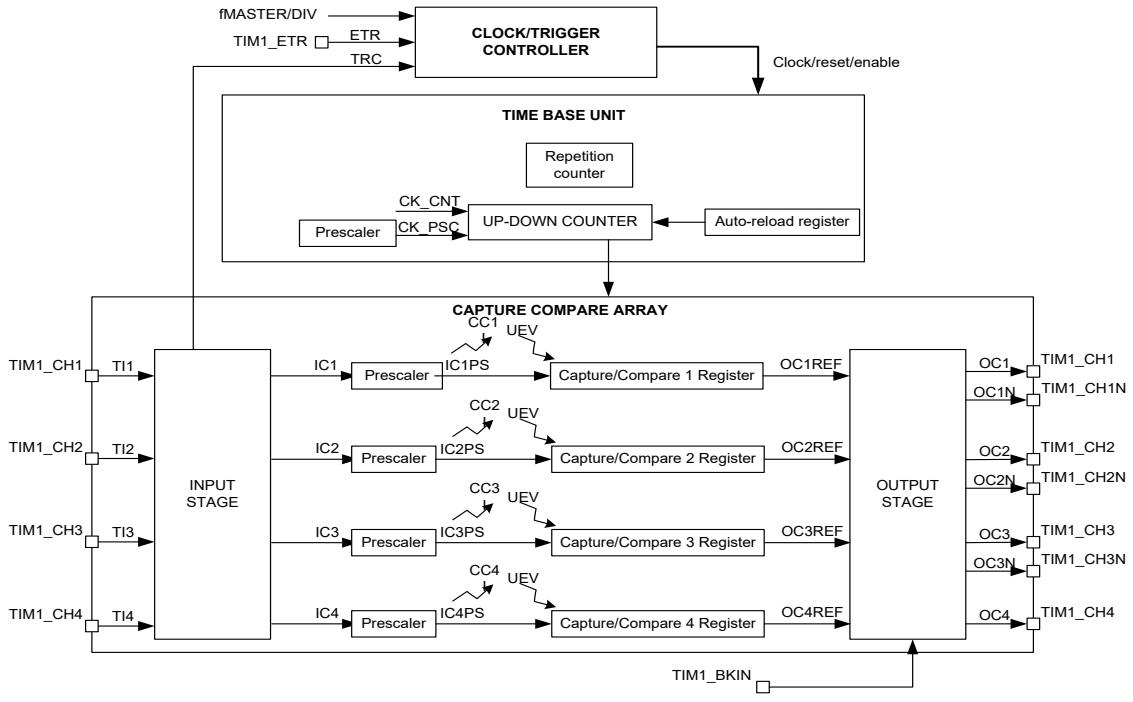


图 9-1 TIM1 原理框图

9.2. Timer1 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
PCKEN	0x9A	—	—	UARTEN	—	TIM4EN	TIM2EN	TIM1EN	ADCEN	0000 0000
CKOCON	0x95	SYSON	CCORDY	DTYSEL		CCOSEL[2:0]			CCOEN	0010 0000
TIM1CR1	0x211	T1ARPE	T1CMS[1:0]		T1DIR	T1OPM	T1URS	T1UDIS	T1CEN	0000 0000
TIM1CR2	0x212	T1ENCTRL	—	—	—	—	T1COMS	—	T1CCPC	0--- -0-0
TIM1SMCR	0x213	—	T1TS[2:0]			—	T1SMS[2:0]			-000 -000
TIM1ETR	0x214	T1ETP	T1ECE	T1ETPS[1:0]		T1EFT[3:0]				0000 0000
TIM1IER	0x215	T1BIE	T1TIE	T1COMIE	T1CC4IE	T1CC3IE	T1CC2IE	T1CC1IE	T1UIE	0000 0000
TIM1SR1	0x216	T1BIF	T1TIF	T1COMIF	T1CC4IF	T1CC3IF	T1CC2IF	T1CC1IF	T1UIF	0000 0000
TIM1SR2	0x217	—	—	—	T1CC4OF	T1CC3OF	T1CC2OF	T1CC1OF	—	---0 000-
TIM1EGR	0x218	T1BG	T1TG	T1COMG	T1CC4G	T1CC3G	T1CC2G	T1CC1G	T1UG	0000 0000
TIM1CCMR1 (output mode)	0x219	T1OC1CE	T1OC1M[2:0]			T1OC1PE	T1OC1FE	T1CC1S[1:0]		0000 0000
TIM1CCMR1 (input mode)		T1IC1F[3:0]				T1IC1PSC[1:0]		T1CC1S[1:0]		0000 0000
TIM1 CCMR2 (output mode)	0x21A	T1OC2CE	T1OC2M[2:0]			T1OC2PE	T1OC2FE	T1CC2S[1:0]		0000 0000
TIM1CCMR2 (input mode)		T1IC2F[3:0]				T1IC2PSC[1:0]		T1CC2S[1:0]		0000 0000
TIM1CCMR3 (output mode)	0x21B	OC3CE	T1OC3M[2:0]			OC3PE	OC3FE	T1CC3S[1:0]		0000 0000
TIM1CCMR3 (input mode)		T1IC3F[3:0]				T1IC3PSC[1:0]		T1CC3S[1:0]		0000 0000
TIM1CCMR4 (output mode)	0x21C	OC4CE	T1OC4M[2:0]			OC4PE	OC4FE	T1CC4S[1:0]		0000 0000
TIM1CCMR4 (input mode)		T1IC4F[3:0]				T1IC4PSC[1:0]		T1CC4S[1:0]		0000 0000
TIM1CCER1	0x21D	T1CC2NP	T1CC2NE	T1CC2P	T1CC2E	T1CC1NP	T1CC1NE	T1CC1P	T1CC1E	0000 0000
TIM1CCER2	0x21E	T1GP	T1SMOD	T1CC4P	T1CC4E	T1CC3NP	T1CC3NE	T1CC3P	T1CC3E	0000 0000
TIM1CNTRH	0x28C	T1CNT[15:8]								0000 0000
TIM1CNTRL	0x28D	T1CNT[7:0]								0000 0000
TIM1PSCRH	0x28E	T1PSC[15:8]								0000 0000
TIM1PSCRL	0x28F	T1PSC[7:0]								0000 0000
TIM1ARRH	0x290	T1ARR[15:8]								1111 1111
TIM1ARRL	0x291	T1ARR[7:0]								1111 1111
TIM1RCR	0x292	T1REP[7:0]								0000 0000
TIM1CCR1H	0x293	T1CCR1[15:8]								0000 0000
TIM1CCR1L	0x294	T1CCR1[7:0]								0000 0000
TIM1CCR2H	0x295	T1CCR2[15:8]								0000 0000
TIM1CCR2L	0x296	T1CCR2[7:0]								0000 0000
TIM1CCR3H	0x297	T1CCR3[15:8]								0000 0000
TIM1CCR3L	0x298	T1CCR3[7:0]								0000 0000
TIM1CCR4H	0x299	T1CCR4[15:8]								0000 0000
TIM1CCR4L	0x29A	T1CCR4[7:0]								0000 0000
TIM1BKR	0x29B	T1MOE	T1AOE	T1BKP	T1BKE	T1OSSR	T1OSSI	T1LOCK[1:0]		0000 0000
TIM1DTR	0x29C	T1DTG[7:0]								0000 0000
TIM1OISR	0x29D	—	T1OIS4	T1OIS3N	T1OIS3	T1OIS2N	T1OIS2	T1OIS1N	T1OIS1	-000 0000
LEBCON	0x41C	LEBEN	LEBCH[1:0]		—	EDGS	BKS[2:0]			000- 0000

表 9-1 Timer1 相关用户寄存器汇总 (- 保留位必须保持为复位值，不可更改)

名称	状态		寄存器	地址	复位值
T1CNT	TIM1 计数值	高 8 位	TIM1CNTRH[7:0]	0x28C	RW-0000 0000
		低 8 位	TIM1CNTRL[7:0]	0x28D	RW-0000 0000
T1PSC	TIM1 预分频器	高 8 位	TIM1PSCRH[7:0]	0x28E	RW-0000 0000
		低 8 位	TIM1PSCRL[7:0]	0x28F	RW-0000 0000
T1ARR	输出比较模式：PWM 周期的自动重装载寄存器(预装载值) 注：此值为 0 时，计数器不工作；	高 8 位	TIM1ARRH[7:0]	0x290	RW-1111 1111
		低 8 位	TIM1ARRL[7:0]	0x291	RW-1111 1111
T1REP	重复向下计数值		TIM1RCR[7:0]	0x292	RW-0000 0000
T1CCR1	输出比较模式: TIM1_CH1 占空比(预装载值)	高 8 位	TIM1CCR1H[7:0]	0x293	RW-0000 0000
		低 8 位	TIM1CCR1L[7:0]	0x294	RW-0000 0000
	输入捕获模式: 上一次捕获事件(IC1)捕获的计数值	高 8 位	TIM1CCR1H[7:0]	0x293	RO-0000 0000
		低 8 位	TIM1CCR1L[7:0]	0x294	RO-0000 0000
T1CCR2	输出比较模式: TIM1_CH2 占空比(预装载值)	高 8 位	TIM1CCR2H[7:0]	0x295	RW-0000 0000
		低 8 位	TIM1CCR2L[7:0]	0x296	RW-0000 0000
	输入捕获模式: 上一次捕获事件 (IC2)捕获的计数值	高 8 位	TIM1CCR2H[7:0]	0x295	RO-0000 0000
		低 8 位	TIM1CCR2L[7:0]	0x296	RO-0000 0000
T1CCR3	输出比较模式: TIM1_CH3 占空比(预装载值)	高 8 位	TIM1CCR3H[7:0]	0x297	RW-0000 0000
		低 8 位	TIM1CCR3L[7:0]	0x298	RW-0000 0000
	输入捕获模式: 上一次捕获事件 (IC3)捕获的计数值	高 8 位	TIM1CCR3H[7:0]	0x297	RO-0000 0000
		低 8 位	TIM1CCR3L[7:0]	0x298	RO-0000 0000
T1CCR4	输出比较模式: TIM1_CH4 占空比(预装载值)	高 8 位	TIM1CCR4H[7:0]	0x299	RW-0000 0000
		低 8 位	TIM1CCR4L[7:0]	0x29A	RW-0000 0000
	输入捕获模式: 上一次捕获事件 (IC4)捕获的计数值	高 8 位	TIM1CCR4H[7:0]	0x299	RO-0000 0000
		低 8 位	TIM1CCR4L[7:0]	0x29A	RO-0000 0000

表 9-2 Timer1 周期相关寄存器

名称	状态		寄存器	地址	复位值
TIM1EN	TIM1 模块时钟	1 = 使能 0 = 关闭	PCKEN[1]	0x9A	RW-0
SYSON	睡眠模式下，系统时钟控制	1 = 使能 0 = 关闭	CKOCON[7]	0x95	RW-0
T1CKSRC	TIM1 时钟源 (Fmaster) 000 = Sysclk　100 = 2 x (XT or EC) (*) 001 = HIRC　101 = LIRC 010 = XT or EC (*)　110 = LP or EC (*) 011 = 2 x HIRC　111 = 2 x (LP or EC) (*) (*) FOSC 需相应配置成 LP/XT/EC 模式或 INTOSCIO 模式，否则振荡器将不会运行。		TCKSRC[2:0]	0x31F	RW-000
DTYSEL	TIM1/TIM2 倍频时钟占空比调节位 00 = 2ns 延迟　10 = 4ns 延迟 01 = 3ns 延迟　11 = 7ns 延迟		CKOCON[5:4]	0x95	RW-10
T1ARPE	PWM 周期的自动预装载 1 = 使能 (T1ARR 预装载值在更新事件到来时被加载) 0 = 禁止 (T1ARR 立即被加载)		TIM1CR1[7]	0x211	RW-0
T1CMS	计数器对齐模式 00 = 边沿对齐模式 (计数方向由 T1DIR 决定) 01 = 中央对齐模式 1 (向下计数时 T1CCxIF 置 1) 10 = 中央对齐模式 2 (向上计数时 T1CCxIF 置 1) 11 = 中央对齐模式 3 (向上和向下计数时 T1CCxIF 均置 1) 注： 1. 中央对齐模式为计数器交替地向上和向下计数。 2. 计数器关闭后(T1CEN=0)，才允许切换模式。		TIM1CR1[6:5]		RW-00
T1DIR	计数器计数方向 (中央对齐/编码器模式时，该位只读) 1 = 向下 0 = 向上		TIM1CR1[4]		RW-0
T1OPM	单脉冲模式 1 = 使能 (在发生下一次更新事件时，T1CEN 自动清零，计数器停止) 0 = 关闭 (发生更新事件时，计数器不停止)		TIM1CR1[3]		RW-0

名称	状态	寄存器	地址	复位值
T1URS	当 T1UDIS = 0 时，更新事件中断请求源 1 = 计数器上溢/下溢 0 = 计数器上溢/下溢、软件设置 T1UG 位、或复位触发事件	TIM1CR1[2]		RW-0
T1UDIS	产生更新事件控制 1 = 禁止 0 = 允许	TIM1CR1[1]		RW-0
T1CEN	TIM1 计数器 1 = 使能 (此时编码器、门控和外部时钟模式才能工作) 0 = 关闭	TIM1CR1[0]		RW-0
T1ENCTRL	TIM1/TIM2 同步触发控制 1 = TIM1/TIM2 同时由 T1CEN 使能 0 = TIM2 由 T2CEN 使能	TIM1CR2[7]	0x212	RW-0
T1COMS	当 T1CCPC = 1 时，捕获/比较控制位更新事件源 1 = 当 T1COMG 置 1 或产生触发事件时，更新控制位 0 = 当 T1COMG 置 1 时，更新控制位 注：仅对带互补输出的通道有效	TIM1CR2[2]		RW-0
T1CCPC	捕获/比较控制位(T1SMOD, T1GP, T1CCxE, T1CCxNE, T1CCxP, T1CCxNP, T1OCxM)的自动预装载 1 = 使能 (控制位的预装载值在换相事件到来时被加载) 0 = 禁止 (控制位立即被加载) 注：仅对带互补输出的通道有效	TIM1CR2[0]		RW-0
T1GP	PWM 群组模式 1 = 使能 (通道 CH1/2/3 及 CH1N/2N/3N 的占空比由 T1CCR1 决定，原配置 T1CCR2/T1CCR3 失效) 0 = 禁止	TIM1CCER2[7]	0x21E	RW-0
T1SMOD	PWM 同步模式 1 = 使能 (通道 CH1N/2N/3N 与 CH1/2/3 的信号完全相同，且无死区功能) 0 = 禁止	TIM1CCER2[6]		RW-0

表 9-3 Timer1 相关用户控制寄存器

名称	状态	寄存器	地址	复位值
T1TS	同步计数器的触发输入源 0xx = 保留 100 = TI1 的边沿检测器 (TI1F_ED) 101 = 滤波后的定时器输入 1 (TI1FP1) 110 = 滤波后的定时器输入 2 (TI2FP2) 111 = 外部触发输入 (ETRF) 注： 1. 仅在 T1SMS = 000 时可更改这些位； 2. 通道 3/4 的输入只能用作对应通道的捕捉源，而不能作为触发源；	TIM1SMCR[6:4]	0x213	RW-000
T1SMS	触发模式 000 = 内部时钟 001 = 编码器模式 1 (根据 TI1FP1 的电平，计数器在 TI2FP2 的有效沿向上/下计数) 010 = 编码器模式 2 (根据 TI2FP2 的电平，计数器在 TI1FP1 的有效沿向上/下计数) 011 = 编码器模式 3 (根据 TI1FP1/ TI2FP2 的电平，计数器在 TI2FP2/TI1FP1 的有效沿向上/下计数) 100 = 复位模式 (在触发输入有效沿时重新初始化计数器和预分频器) 101 = 门控模式 (计数器在触发输入有效电平期间计数，无效电平则停止计数，但不复位) 110 = 触发模式 (计数器在触发输入有效沿时计数且不复位) 111 = 外部时钟模式 1 (计数器在触发输入有效沿时计数) 注： 1. 门控模式不能选择 TI1F_ED 触发输入； 2. 触发输入有效沿的极性参阅 T1CC1P/T1CC2P 或 T1ETP；	TIM1SMCR[2:0]		RW-000

<table>
<tr><th>名称</th><th>状态</th><th>寄存器</th><th>地址</th><th>复位值</th></tr>
<tr><td>T1ETP</td><td>外部触发极性
1 = ETR 反相，即低电平或下降沿有效
0 = ETR 不反相，即高电平或上升沿有效</td><td>TIM1ETR[7]</td><td rowspan="5">0x214</td><td>RW-0</td></tr>
<tr><td>T1ECE</td><td>外部时钟模式 2
1 = 使能，计数时钟为 ETR 有效沿
0 = 关闭
注：
1. 当外部时钟模式 1 选择外部触发输入 ETR 时，与外部时钟模式 2 的功能相同；
2. 外部时钟模式 2 可与触发/复位/门控模式同时使用，但此时其他模式的触发输入源不能选择 ETRF；
3. 外部时钟模式 2 与外部时钟模式 1 不能同时使用；</td><td>TIM1ETR[6]</td><td>RW-0</td></tr>
<tr><td>T1ETPS[1:0]</td><td>外部触发预分频比
00 = 1 10 = 4
01 = 2 11 = 8
注：外部触发信号的频率需满足≤Fmaster/4</td><td>TIM1ETR[5:4]</td><td>RW-00</td></tr>
<tr><td>T1ETF[3:0]</td><td>外部触发滤波器采样频率及数字滤波器长度
<table>
<tr><th>Value</th><th>采样频率($f_{SAMPLING}$)</th><th>数字滤波器长度 N</th></tr>
<tr><td>0000</td><td>Fmaster</td><td>0</td></tr>
<tr><td>0001</td><td>Fmaster</td><td>2</td></tr>
<tr><td>0010</td><td>Fmaster</td><td>4</td></tr>
<tr><td>0011</td><td>Fmaster</td><td>8</td></tr>
<tr><td>0100</td><td>Fmaster / 2</td><td>6</td></tr>
<tr><td>0101</td><td>Fmaster / 2</td><td>8</td></tr>
<tr><td>0110</td><td>Fmaster / 4</td><td>6</td></tr>
<tr><td>0111</td><td>Fmaster / 4</td><td>8</td></tr>
<tr><td>1000</td><td>Fmaster / 8</td><td>6</td></tr>
<tr><td>1001</td><td>Fmaster / 8</td><td>8</td></tr>
<tr><td>1010</td><td>Fmaster / 16</td><td>5</td></tr>
<tr><td>1011</td><td>Fmaster / 16</td><td>6</td></tr>
<tr><td>1100</td><td>Fmaster / 16</td><td>8</td></tr>
<tr><td>1101</td><td>Fmaster / 32</td><td>5</td></tr>
<tr><td>1110</td><td>Fmaster / 32</td><td>6</td></tr>
<tr><td>1111</td><td>Fmaster / 32</td><td>8</td></tr>
</table></td><td>TIM1ETR[3:0]</td><td>RW-0000</td></tr>
</table>

表 9-4　Timer1 触发相关用户控制寄存器

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	Bit0	复位值
TIM1CCMR1	0x219	T1IC1F[3:0]				T1IC1PSC[1:0]		T1CC1S[1:0]		RW-0000 0000
TIM1CCMR2	0x21A	T1IC2F[3:0]				T1IC2PSC[1:0]		T1CC2S[1:0]		RW-0000 0000
TIM1CCMR3	0x21B	T1IC3F[3:0]				T1IC3PSC[1:0]		T1CC3S[1:0]		RW-0000 0000
TIM1CCMR4	0x21C	T1IC4F[3:0]				T1IC4PSC[1:0]		T1CC4S[1:0]		RW-0000 0000

名称	状态			寄存器	地址	复位值
T1ICxF	通道 x 输入捕获采样频率和数字滤波器长度			TIM1CCMRx[7:4] x = 1, 2, 3, 4	0x219/ 0x21A/ 0x21B/ 0x21C	RW-0000
	Value	采样频率($f_{SAMPLING}$)	数字滤波器长度 N			
	0000	Fmaster	0			
	0001	Fmaster	2			
	0010	Fmaster	4			
	0011	Fmaster	8			
	0100	Fmaster / 2	6			
	0101	Fmaster / 2	8			
	0110	Fmaster / 4	6			
	0111	Fmaster / 4	8			
	1000	Fmaster / 8	6			
	1001	Fmaster / 8	8			
	1010	Fmaster / 16	5			
	1011	Fmaster / 16	6			
	1100	Fmaster / 16	8			
	1101	Fmaster / 32	5			
	1110	Fmaster / 32	6			
	1111	Fmaster / 32	8			
T1ICxPSC	通道 x 输入捕获预分频器（几个事件触发一次捕获） 00 = 1 个　　10 = 4 个 01 = 2 个　　11 = 8 个 注：当 T1CCxE = 0 时，该预分频器复位为 00；			TIM1CCMRx[3:2]		RW-00
T1CC1S [1]	通道 1 模式选择	00 = 输出 01 = 输入，输入脚映射在 TI1FP1 10 = 输入，输入脚映射在 TI2FP1 11 = 输入，输入脚映射在 TRC		TIM1CCMR1[1:0]	0x219	RW-00

[1] 仅在通道 x 关闭（T1CCxE = 0 和 T1CCxNE = 0）时可写。

名称	状态		寄存器	地址	复位值
T1CC2S [2]	通道 2 模式选择	00 = 输出 01 = 输入，输入脚映射在 TI2FP2 10 = 输入，输入脚映射在 TI1FP2 11 = 输入，输入脚映射在 TRC	TIM1CCMR2[1:0]	0x21A	RW-00
T1CC3S [2]	通道 3 模式选择	00 = 输出 01 = 输入，输入脚映射在 TI3FP3 10 = 输入，输入脚映射在 TI4FP3 11 = 保留	TIM1CCMR3[1:0]	0x21B	RW-00
T1CC4S [2]	通道 4 模式选择	00 = 输出 01 = 输入，输入脚映射在 TI3FP4 10 = 输入，输入脚映射在 TI4FP4 11 = 保留	TIM1CCMR4[1:0]	0x21C	RW-00

表 9-5 TIM1CCMRx 作为输入配置寄存器

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	Bit0	复位值
TIM1CCMR1	0x219	T1OC1CE	T1OC1M[2:0]			T1OC1PE	T1OC1FE	T1CC1S[1:0]		RW-0000 0000
TIM1CCMR2	0x21A	T1OC2CE	T1OC2M[2:0]			T1OC2PE	T1OC2FE	T1CC2S[1:0]		RW-0000 0000
TIM1CCMR3	0x21B	T1OC3CE	T1OC3M[2:0]			T1OC3PE	T1OC3FE	T1CC3S[1:0]		RW-0000 0000
TIM1CCMR4	0x21C	T1OC4CE	T1OC4M[2:0]			T1OC4PE	T1OC4FE	T1CC4S[1:0]		RW-0000 0000

T1OCxM	通道 x 输出比较模式		OCxREF (输出参考信号)
000	冻结 (不比较)		禁止
001	当 TIM1_CNT = CCRx_SHAD 时		1
010	当 TIM1_CNT = CCRx_SHAD 时		0
011	当 TIM1_CNT = CCRx_SHAD 时		电平翻转
100	强制为无效电平		0
101	强制为有效电平		1
110	PWM1 模式	TIM1_CNT < CCRx_SHAD	1
		TIM1_CNT > CCRx_SHAD	0
111	PWM2 模式	TIM1_CNT < CCRx_SHAD	0
		TIM1_CNT > CCRx_SHAD	1

1. OCxREF 与 T1CCxP 共同决定输出引脚 OCx 的值；
2. PWM 模式下比较结果改变时，或输出比较模式下从冻结模式切换到 PWM 模式时，OCxREF 电平才会改变；

表 9-6 T1OCxM 配置为输出比较模式

[2] 仅在通道 x 关闭 (T1CCxE = 0 和 T1CCxNE = 0) 时可写。

名称	状态		寄存器	地址	复位值
T1OCxCE	通道 x 输出参考信号 OC1REF 由 ETR 清零控制 1 = 使能 (当 ETR 为有效电平时，OC1REF 清零) 0 = 禁止 (OC1REF 不受 ETR 输入的影响)		TIM1CCMRx[7] x = 1, 2, 3, 4	0x219/ 0x21A/ 0x21B/ 0x21C	RW-0
T1OCxPE	通道 x 输出比较占空比的自动预装载 1 = 使能 (T1CCRx 预装载值在更新事件到来时被加载) 0 = 禁止 (T1CCRx 立即被加载) 注：PWM 模式下必须使能，单脉冲模式可选		TIM1CCMRx[3] x = 1, 2, 3, 4		RW-0
T1OCxFE	通道 x 输出信号快速使能 1 = 使能 (输入触发有效沿到通道 x 输出的延时≥ 3*$T_{Fmaster}$，且通道 x 输出为固定值，PWM1 模式输出值=T1CEN，PWM2 模式输出值=!T1CEN) 0 = 禁止 (输入触发有效沿到通道 x 输出的延时≥ 5*$T_{Fmaster}$) 注：该位只在通道 1 被配置成 PWM1/PWM2 模式时起作用		TIM1CCMRx[2] x = 1, 2, 3, 4		RW-0
T1CC1S [3]	通道 1 模式选择	00 = 输出 01 = 输入，输入脚映射在 TI1FP1 10 = 输入，输入脚映射在 TI2FP1 11 = 输入，输入脚映射在 TRC	TIM1CCMR1[1:0]	0x219	RW-00
T1CC2S [3]	通道 2 模式选择	00 = 输出 01 = 输入，输入脚映射在 TI2FP2 10 = 输入，输入脚映射在 TI1FP2 11 = 保留	TIM1CCMR2[1:0]	0x21A	RW-00
T1CC3S [3]	通道 3 模式选择	00 = 输出 01 = 输入，输入脚映射在 TI3FP3 10 = 输入，输入脚映射在 TI4FP3 11 = 保留	TIM1CCMR3[1:0]	0x21B	RW-00
T1CC4S [3]	通道 4 模式选择	00 = 输出 01 = 输入，输入脚映射在 TI3FP4 10 = 输入，输入脚映射在 TI4FP4 11 = 保留	TIM1CCMR4[1:0]	0x21C	RW-00

表 9-7 TIM1CCMRx 作为输出配置寄存器

[3] 仅在通道 x 关闭 (T1CCxE = 0 和 T1CCxNE = 0) 时可写。

名称	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	地址	复位值
TIM1CCER1	T1CC2NP	T1CC2NE	T1CC2P	T1CC2E	T1CC1NP	T1CC1NE	T1CC1P	T1CC1E	0x21D	RW-0000 0000
TIM1CCER2	T1GP	T1SMOD	T1CC4P	T1CC4E	T1CC3NP	T1CC3NE	T1CC3P	T1CC3E	0x21E	RW-0000 0000

名称	功能	输出比较模式	输入捕获/触发模式
T1CCxP	通道 x 引脚 输出极性选择	1 = OCx 低电平有效 0 = OCx 高电平有效	1 = 捕获 / 触发发生在 TIxF 低电平或下降沿 0 = 捕获 / 触发发生在 TIxF 高电平或上升沿 注：输入触发源仅可选通道 1 和 2
T1CCxE	通道 x 引脚 输出使能	1 = 使能 (OCx 输出到对应的引脚) 0 = 禁止	1 = 使能 (捕获计数器的值到 TIM1CCRx 寄存器中) 0 = 禁止
T1CCxNP	通道 x 互补引脚 输出极性选择	1 = OCxN 低电平有效 0 = OCxN 高电平有效	–
T1CCxNE	通道 x 互补引脚 使能	1 = 使能 (OCxN 输出到对应的引脚) 0 = 关闭	–
注：通道输出电平由 T1MOE, T1OSSI, T1OSSR, T1OISx, T1OISxN, T1CCxE 和 T1CCxNE 位的值共同决定，参阅**表 9-10**。			

表 9-8 Timer1 通道输出和极性选择

名称	状态		寄存器	地址	复位值
TIM1_CH1N	通道 1 互补输出重映射	1 = PB2 0 = PC0	AFP0[2]	0x19E	RW-0
TIM1_CH2	通道 2 输出重映射	1 = PB0 0 = PA1	AFP0[5]		
TIM1_CH4	通道 4 输出重映射	0x = PB1 10 = PA7 11 = PA2	AFP1[1:0]	0x19F	RW-00

表 9-9 Timer1 管脚功能重映射寄存器

名称	控制	寄存器	地址	复位值
T1MOE [4]	主输出控制 (仅对配置为输出的通道有效) 1 = 使能 (若 T1CCxE= 1，则使能 OC 和 OCN 输出) 0 = 禁止 (禁止 OC 和 OCN 输出或强制为空闲状态)	TIM1BKR[7]	0x29B	RW-0
T1AOE	自动输出控制 1 = T1MOE 在下一个更新事件到来时被自动置 1(当刹车输入无效时) 或由软件置 1 0 = T1MOE 只能由软件置 1	TIM1BKR[6]		RW-0
T1BKP	故障源 TIM1_BKIN 刹车输入极性 1 = 高电平有效 0 = 低电平有效	TIM1BKR[5]		RW-0
T1BKE	刹车输入(BRK)功能 1 = 使能 0 = 禁止	TIM1BKR[4]		RW-0
T1OSSR	运行模式下(当 T1MOE = 1 时)输出“关闭状态”选择 详情请见**表 9-11 Timer1 输出控制和状态**”	TIM1BKR[3]		RW-0
T1OSSI	空闲模式下(当 T1MOE=0 时)输出“关闭状态”选择 详情请见**表 9-11 Timer1 输出控制和状态**”	TIM1BKR[2]		RW-0
T1LOCK [5]	锁定设置 (写保护，防止软件错误)	TIM1BKR[1:0]		RW-00

00	01	10	11
关闭	锁定级别 1	锁定级别 2	锁定级别 3
寄存器无写保护	T1BKE, T1BKP, T1AOE, T1OISx, T1OISxN, T1DTG	包含级别 1, T1CCxP, T1CCxNP, T1OSSR, T1OSSI	包含级别 2, T1OCxM, T1OCxPE

表 9-10 Timer1 主输出使能、刹车和锁定级别寄存器

[4] 当刹车输入有效时，该位将被硬件异步清 0。

[5] 系统复位后只能写一次 LOCK 位，一旦写入则其内容保持不变直至复位。

控制位					输出状态	
T1MOE	T1OSSI	T1OSSR	T1CCxE	T1CCxNE	OCx 输出状态	OCxN 输出状态
1	X	0	0	0	OCx = 0 (输出关闭)	OCxN = 0 (输出关闭)
		0	0	1	OCx = 0 (输出关闭)	OCxREF + 极性 OCxN = OCxREF ^ T1CCxNP
		0	1	0	OCxREF + 极性 OCx = OCxREF ^ T1CCxNP	OCxN = 0 (输出关闭)
		0	1	1	OCxREF + 极性 + 死区	OCxREF 的互补信号 + 极性 + 死区
		1	0	0	OCx = T1CCxP (输出关闭)	OCxN = T1CCxNP (输出关闭)
		1	0	1	关闭状态(运行模式下输出使能) OCx = T1CCxP	OCxREF + 极性 OCxN = OCxREF ^ T1CCxNP
		1	1	0	OCxREF + 极性 OCx = OCxREF ^ T1CCxNP	关闭状态(运行模式下输出使能) OCxN = T1CCxNP
		1	1	1	OCxREF + 极性 + 死区	OCxREF 的互补信号 + 极性 + 死区
0	0	X	0	0	OCx = T1CCxP (输出关闭)	OCxN = T1CCxNP (输出关闭)
	0		0	1	(输出关闭) OCx = T1OISx, OCxN = T1OISxN	
	0		1	0		
	0		1	1		
	1		0	0	OCx=T1CCxP (输出关闭)	OCxN=T1CCxNP (输出关闭)
	1		0	1	关闭状态(空闲模式下输出使能) OCx = T1OISx, OCxN = T1OISxN	
	1		1	0		
	1		1	1		

表 9-11 Timer1 输出控制和状态

名称	状态	寄存器	地址	复位值
T1OIS4	当 T1MOE=0 时，通道 4(OC4) 空闲状态输出 1 = OC4 输出 1 0 = OC4 输出 0	TIM1OISR[6]	0x29D	RW-0
T1OIS3	当 T1MOE=0 时，通道 3/2/1(OCx) 空闲状态输出	TIM1OISR[4]		RW-0
T1OIS2	1 = 死区时间后，OCx 输出 1	TIM1OISR[2]		RW-0
T1OIS1	0 = 死区时间后，OCx 输出 0	TIM1OISR[0]		RW-0
T1OIS3N	当 T1MOE=0 时，互补通道 3/2/1(OCxN) 空闲状态输出	TIM1OISR[5]		RW-0
T1OIS2N	1 = 死区时间后，OCxN 输出 1	TIM1OISR[3]		RW-0
T1OIS1N	0 = 死区时间后，OCxN 输出 0	TIM1OISR[1]		RW-0

表 9-12 Timer1 通道输出空闲状态寄存器

<table>
<tr><th>名称</th><th colspan="3">控制</th><th>寄存器</th><th>地址</th><th>复位值</th></tr>
<tr><td rowspan="11">T1DTG</td><td colspan="3">死区发生器设置</td><td rowspan="11">TIM1DTR[7:0]</td><td rowspan="11">0x29C</td><td rowspan="11">RW-0000 0000</td></tr>
<tr><td>T1DTG[7:5]</td><td>DT(死区持续时间)</td><td>t_{DTG}</td></tr>
<tr><td>0xx</td><td>T1DTG[7:0] x t_{DTG}</td><td>$T_{Fmaster}$ (f1)</td></tr>
<tr><td>10x</td><td>(64+T1DTG[5:0]) x t_{DTG}</td><td>2 x $T_{Fmaster}$ (f2)</td></tr>
<tr><td>110</td><td>(32+T1DTG[4:0]) x t_{DTG}</td><td>8 x $T_{Fmaster}$ (f3)</td></tr>
<tr><td>111</td><td>(32+T1DTG[4:0]) x t_{DTG}</td><td>16 x $T_{Fmaster}$ (f4)</td></tr>
<tr><td colspan="3">* $T_{Fmaster}$ 为 TIM1 时钟源
若 $T_{Fmaster}$ =125 ns (8 MHz)时，死区时间如下：</td></tr>
<tr><td>T1DTG[7:0]</td><td>死区时间 (μs)</td><td>步长时间</td></tr>
<tr><td>0 ~ 7Fh</td><td>0 ~ 15.875</td><td>125 ns (f1)</td></tr>
<tr><td>80h ~ BFh</td><td>16 ~ 31.75</td><td>250 ns (f2)</td></tr>
<tr><td>C0h ~ DFh</td><td>32 ~ 63</td><td>1 μs (f3)</td></tr>
<tr><td>E0h ~ FFh</td><td>64 ~ 126</td><td>2 μs (f4)</td></tr>
</table>

表 9-13 Timer1 互补输出死区时间配置

名称	状态	寄存器	地址	复位值
LEBEN	前沿消隐 1 = 使能 0 = 关闭 注：必须在 GO/DONE=0 时进行切换，否则 ADC 结果不确定；	LEBCON[7]	0x41C	RW-0
LEBCH	前沿消隐通道 00 = TIM1_CH1 10 = TIM1_CH3 01 = TIM1_CH2 11 = TIM1_CH4	LEBCON[6:5]		RW-00
EDGS	PWM 消隐沿 1 = PWM 下降沿 0 = PWM 上升沿	LEBCON[3]		RW-0
BKS	TIM1 的故障源 000 = 关闭 001 = BKIN 管脚 010 = LVD 检测 100 = ADC 阈值比较	LEBCON[2:0]		RW-000

表 9-14 LEB 用户寄存器

名称	状态		寄存器	地址	复位值
GIE	全局中断 1 = 使能 (PEIE, T1BIE, T1BG, T1TIE, T1TG, T1COMIE, T1COMG, T1CCxIE, T1CCxG, T1UIE 适用) 0 = 全局关闭 (唤醒不受影响)		INTCON[7]	Bank 首地址 +0x0B	RW-0
PEIE	外设总中断 1 = 使能 (T1BIE, T1BG, T1TIE, T1TG, T1COMIE, T1COMG, T1CCxIE, T1CCxG, T1UIE 适用) 0 = 关闭 (无唤醒)		INTCON[6]		RW-0
T1BIE	刹车中断	1 = 使能 0 = 关闭	TIM1IER[7]	0x215	RW-0
T1BG [6]	刹车软件中断		TIM1EGR[7]	0x218	WO-0
T1BIF [7]	刹车中断标志位 1 = 刹车输入上检测到有效电平 0 = 无刹车事件产生		TIM1SR1[7]	0x216	R_W1C-0
T1TIE	触发中断	1 = 使能 0 = 关闭	TIM1IER[6]	0x215	RW-0
T1TG [6]	触发软件中断		TIM1EGR[6]	0x218	WO-0
T1TIF [7]	触发中断标志位 1 = 已触发 0 = 无触发事件产生		TIM1SR1[6]	0x216	R_W1C-0
T1COMIE	换相 (COM) 中断	1 = 使能 0 = 关闭	TIM1IER[5]	0x215	RW-0
T1COMG [6]	换相 (COM) 软件中断		TIM1EGR[5]	0x218	WO-0
T1COMIF [7]	换相 (COM) 中断标志位 1 = 已换相 0 = 无换相事件产生		TIM1SR1[5]	0x216	R_W1C-0
T1CC4IE	通道 4 捕获/比较中断	1 = 使能 0 = 关闭	TIM1IER[4]	0x215	RW-0
T1CC3IE	通道 3 捕获/比较中断		TIM1IER[3]		RW-0
T1CC2IE	通道 2 捕获/比较中断		TIM1IER[2]		RW-0
T1CC1IE	通道 1 捕获/比较中断		TIM1IER[1]		RW-0
T1CC4G [6]	通道 4 捕获/比较软件中断		TIM1EGR[4]	0x218	WO-0
T1CC3G [6]	通道 3 捕获/比较软件中断		TIM1EGR[3]		WO-0
T1CC2G [6]	通道 2 捕获/比较软件中断		TIM1EGR[2]		WO-0
T1CC1G [6]	通道 1 捕获/比较软件中断		TIM1EGR[1]		WO-0

[6] 软件置 1，硬件自动清 0。
[7] 写 1 清 0，写 0 无效。建议只使用 STR、MOVWI 指令进行写操作，而不要用 BSR 或 IOR 指令。

名称	状态		寄存器	地址	复位值
T1CC4IF [8]	通道 x 捕获/比较中断标志位 • 输出模式： 1 = CNT 值与 T1CCRx 值匹配 0 = 不匹配 注：若 T1CCRx>T1ARR，则当 CNT 计数到 T1ARR 值时，T1CCxIF 置 1。 • 输入模式： 1 = 计数器值已被捕获至 TIM1CCR 0 = 无捕获产生		TIM1SR1[4]	0x216	R_W1C-0
T1CC3IF [8]			TIM1SR1[3]		R_W1C-0
T1CC2IF [8]			TIM1SR1[2]		R_W1C-0
T1CC1IF [8]			TIM1SR1[1]		R_W1C-0
T1CC4OF [8]	通道 x 重复捕获标志位 1 = 发生重复捕获 (计数器的值被捕获到 TIM1CCRx 寄存器时，T1CCxIF 的状态已经为 1) 0 = 无重复捕获 注：仅通道配置位捕获输入时有效		TIM1SR2[4]	0x217	R_W1C-0
T1CC3OF [8]			TIM1SR2[3]		R_W1C-0
T1CC2OF [8]			TIM1SR2[2]		R_W1C-0
T1CC1OF [8]			TIM1SR2[1]		R_W1C-0
T1UIE	允许更新中断	1 = 使能 0 = 关闭	TIM1IER[0]	0x215	RW-0
T1UG [9]	允许更新软件中断		TIM1EGR[0]	0x218	WO-0
T1UIF [8]	更新中断标志位 1 = 更新事件等待响应 0 = 无更新事件		TIM1SR1[0]	0x216	R_W1C-0

表 9-15 Timer1 中断使能和状态位

[8] 写 1 清 0，写 0 无效。建议只使用 STR、MOVWI 指令进行写操作，而不要用 BSR 或 IOR 指令。

[9] 软件置 1，硬件自动清 0。

9.3. 功能描述

整个 TIM1 可以分为三个大的功能部分：计数基本单元、计数控制和捕捉比较通道。计数基本单元分为向上/向下计数器、自动加载寄存器、重复计数器和预分频器；计数控制器又分为计数触发源，模式控制；捕捉比较通道分为捕捉输入通道，输出比较通道，死区产生和输出控制。

9.3.1. 计数基本单元

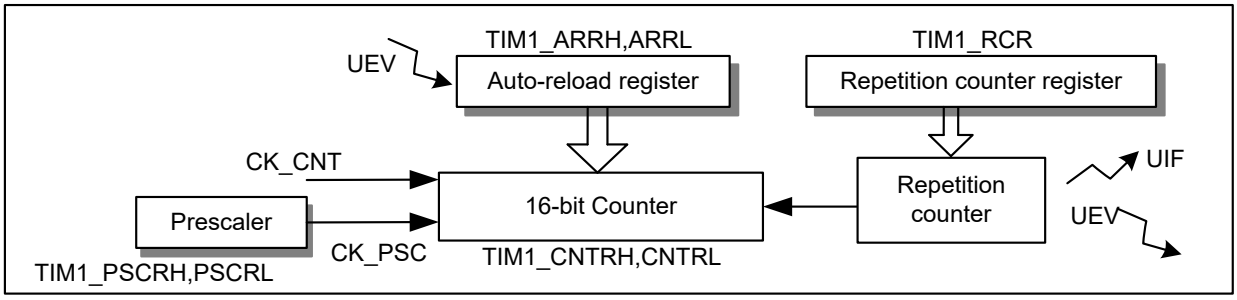


图 9-2 计数基本单元

16 位计数器，预分频器，自动重载寄存器和重复计数寄存器都能由软件进行读写。

9.3.1.1. 计数基本单元组成

9.3.1.1.1. 16 位计数器的读写

- TIM1CNTRH/L 能在任何时候进行写操作；但是建议为了避免出现不正确的中间状态，不要在计数器运行的时候进行写操作
- TIM1CNTRH/L 的写操作是没有顺序限制的；可以先写高位也可以先写低位
- TIM1CNTRH/L 能在任何时候进行读操作；但是因为此设计是异步设计，所以在计数器运行期间进行读操作可能读出不正确的数值，需要读两次，比较两次数值是否一致；如果一致，则读出的数值是正确的数值；否则，读出数值是错误的。

9.3.1.1.2. 预分频器

计数时钟可以进行 16bit 的时钟预分频，分频系数为 1~65536。计算计数器时钟分频的公式，如下：

$$f_{CK_CNT} = Fmaster / (PSCR[15:0] + 1)$$ (PSCR 为实际装入预分频器影子寄存器的值)

预分频支持分频自动更新，即在更新事件发生后，能够自动改变预分频值。当 T1CEN 为 0 时，写入预分频寄存器的值也能直接加载实际应用的预分频寄存器中。配置步骤示例如下：

1. 使能 TIM1 模块时钟并选择 TIM1 时钟源
2. 配置计数周期
3. 配置占空比
4. 配置预分频
5. 使能计数器

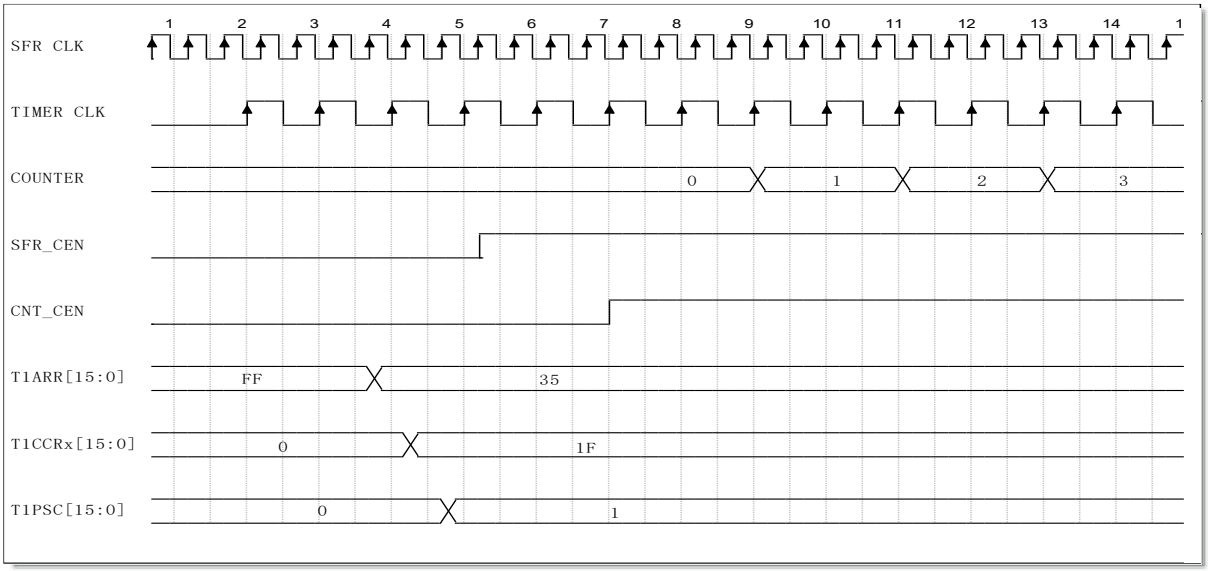


图 9-3 预分频为 1 时的计数器计数时序图

9.3.1.1.3. 自动重载寄存器

自动重载寄存器由一个预加载寄存器和一个影子寄存器组成。写自动重载寄存器的三种方式：

- 方式 1：计数器使能位打开并且周期预加载使能(T1ARPE=1)。在这种模式下，写入自动重载寄存器的值保存在预加载寄存器中，并在下一个更新事件到来时传送到影子寄存器中进行使用。如下图所示：

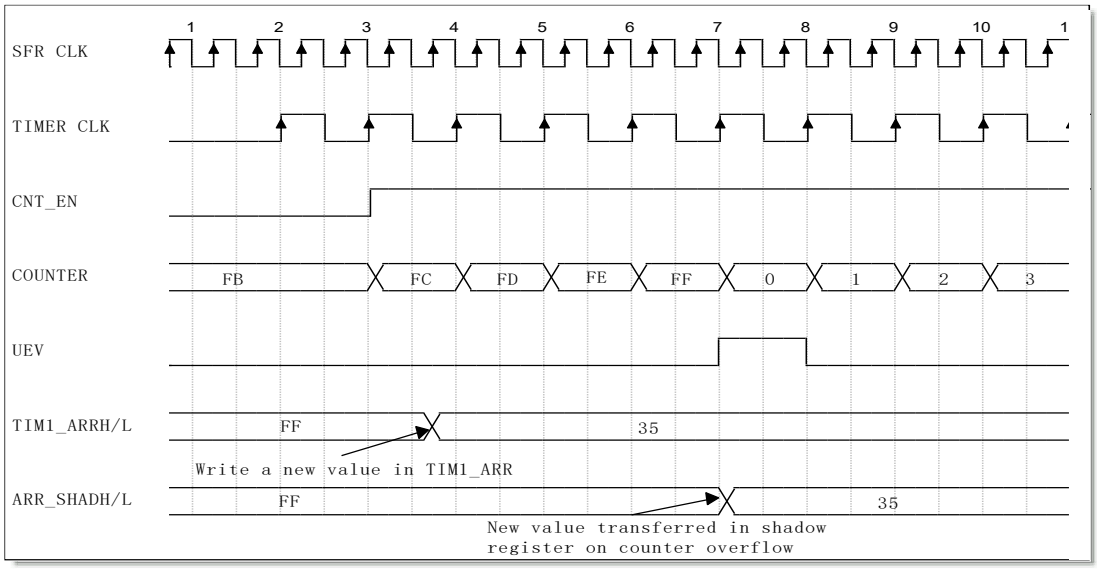


图 9-4 T1CEN = 1 且 T1ARPE = 1，周期寄存器 (T1ARR) 加载图

- 方式 2：计数器使能位打开并且周期预加载关闭(T1ARPE=0)。在这种模式下，写入自动重载寄存器的数值直接传送到影子寄存器中进行使用。如下图所示：

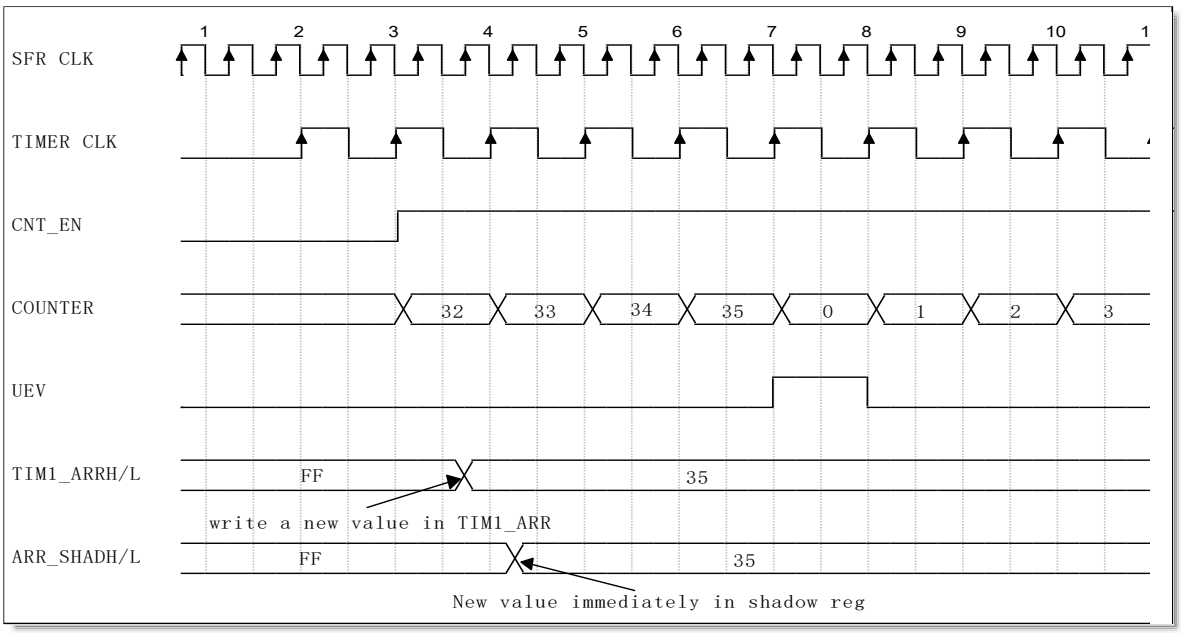


图 9-5 T1CEN = 1 且 T1ARPE = 0，周期寄存器 (T1ARR) 加载图

- 方式 3：当计数器使能位(T1CEN)关闭时，不管周期预加载(T1ARPE)使能还是关闭，写入自动重载寄存器的数值直接传送到影子寄存器中进行使用。如下图所示：

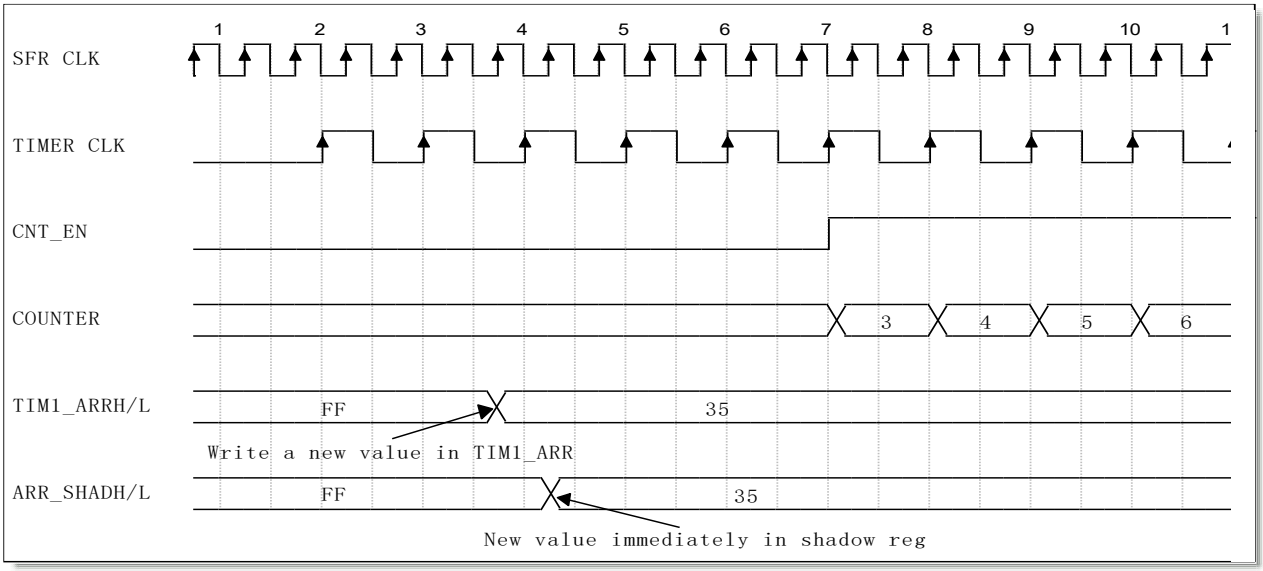


图 9-6 T1CEN = 0，周期寄存器 (T1ARR) 加载图

9.3.1.1.4. 更新事件

更新事件(UEV)产生的条件：

- 计数器上溢或下溢
- 通过软件设置 TIM1EGR 寄存器中的 T1UG 位
- 设置为复位模式时，触发事件的到来

更新事件的影响：

- 影响 1：某些预加载的寄存器(具体寄存器可查看寄存器表格)在预加载使能的情况下都能被更新为最新值。各类预加载寄存器的总结如下表所示：

在更新事件下，可进行预加载的寄存器	TIMARRH/L	TIM1PSCRH/L	TIM1CCRxH/L
相应的预加载使能位	T1ARPE	无使能位，预加载在计数器使能(T1CEN=1) 时一直有效	T1OCxPE

表 9-16 更新事件相关的预加载寄存器 vs 预加载使能位

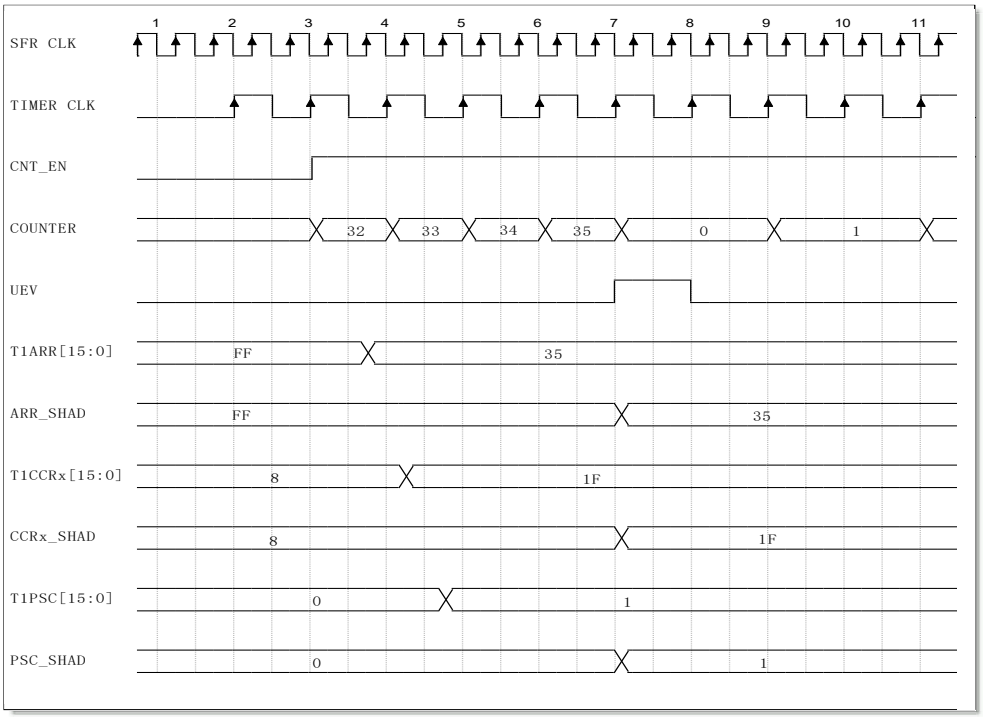


图 9-7 更新事件下，预加载寄存器的更新图

- 影响 2：若 T1UDIS=0，当产生更新事件时，更新标志位(T1UIF)被置位；反之，T1UDIS=1 时，不产生更新事件，更新标志位(T1UIF)也不会被置位。如下图所示：

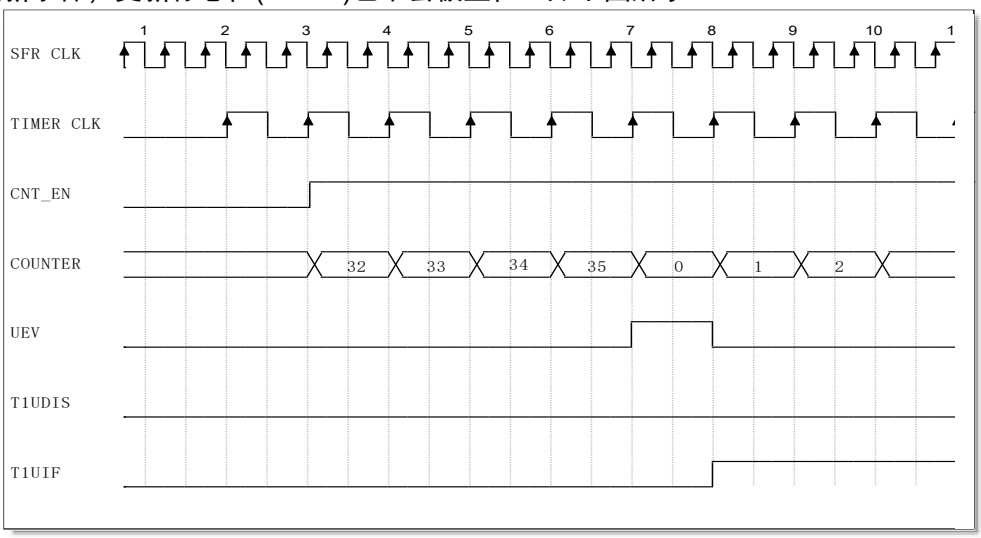


图 9-8　更新事件下且 T1UDIS = 0，更新标志位变化图

- 影响 3：单次脉冲模式下，更新事件的到来会使计数器使能位 (T1CEN) 关闭，计数器停止计数。关于单次脉冲模式的详细说明可查看**章节 9.3.3.3** 内容。

- 影响 4：故障事件撤消后，如果 T1AOE=1，PWM 将在更新事件后恢复正常输出。关于故障刹车事件的详细说明可查看**章节 9.3.8** 内容。

9.3.1.1.5. 换相事件

换相事件实际上是捕捉/比较控制位更新事件，用于更新控制输入输出的预加载位；换相事件预加载功能使得计数器在运行过程中改变输出控制位，而不影响上一个 PWM 波形的完整输出。

换相事件的产生：

- 通过软件设置 TIM1EGR 寄存器中的 T1COMG 位
- 配置为输入通道并且开启更新控制选择位(T1COMS=1)时，触发事件的到来用于产生换相事件的触发事件实际是计数触发源，关于计数触发源的详细描述可查看 10.3.2.2 章节内容。

换相事件的影响

- 影响 1：某些预加载的寄存器(具体寄存器可查看寄存器表格)在预加载使能的情况下都能被更新为最新值。各类预加载寄存器位的总结如下表所示：
- 影响 2：当产生换相事件时，换相事件标志位(T1COMIF)被置位。

换相事件下，可进行预加载的寄存器位	T1SMOD	T1GP	T1CC1E T1CC2E T1CC3E	T1CC1NE T1CC2NE T1CC3NE	T1CC1P T1CC2P T1CC3P	T1CC1NP T1CC2NP T1CC3NP	T1OC1M T1OC2M T1OC3M
预加载使能位	T1CCPC						

表 9-17 换相事件相关的预加载寄存器 vs 预加载使能位

9.3.1.2. 向上计数模式

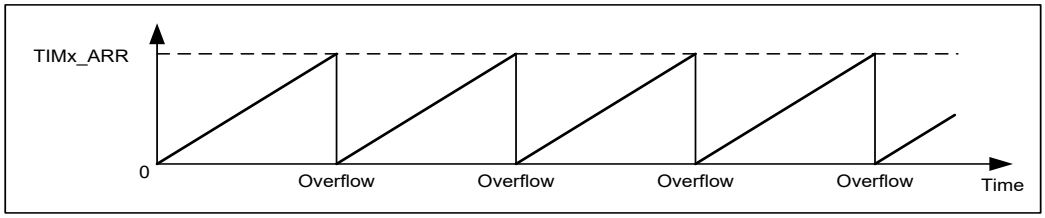


图 9-9 向上计数模式

在向上计数模式中，计数器从 0 开始计数向上计数，计到 TIM1_ARR 寄存器所设数值。然后重新从 0 开始计数并产生一个计数器上溢事件；如果 T1UDIS 设为 0，那么还会产生一个更新事件 UEV。

9.3.1.3. 向下计数模式

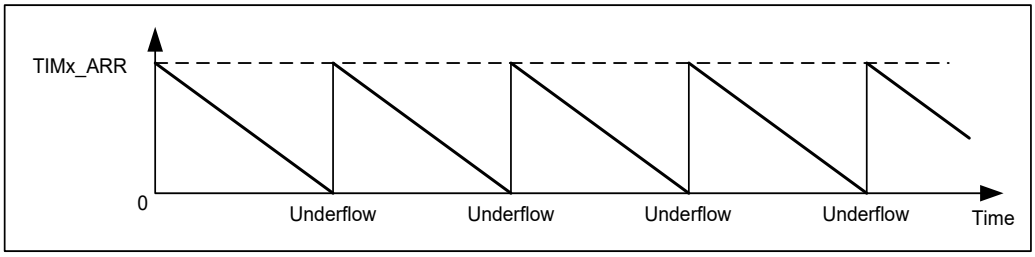


图 9-10 向下计数模式

在向下计数模式中，计数器从 TIM1_ARR 寄存器设置的自动重载值开始向下计数，直到计到 0。然后重新从自动重载值开始计数并产生一个计数器下溢事件；如果 T1UDIS 设为 0，那么还会产生一个更新事件 UEV。

9.3.1.4. 中心对齐模式

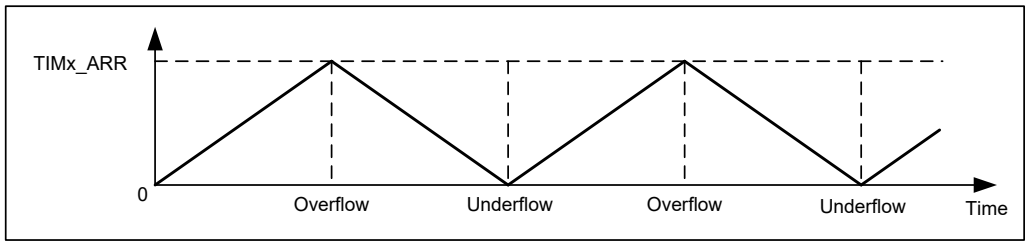


图 9-11 中心对齐模式，T1DIR 初始化为 0

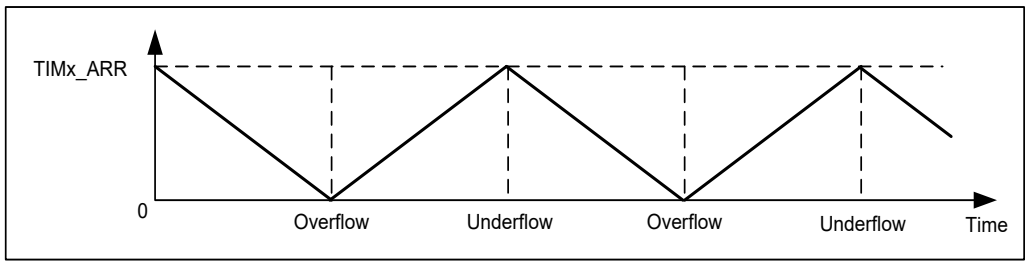


图 9-12 中心对齐模式，T1DIR 初始化为 1

在中心对齐模式中，计数器从 0 开始向上计数，计到自动重载值。此时会产生一个计数器上溢事件。然

后计数器开始向下计数计到 0，产生一个下溢事件。计数器不断地重复上述的计数过程。

中心对齐模式所需注意事项：

- 方向位(T1DIR)不能进行写操作。方向位会由硬件设置成当前计数器的计数方向
- 当在中心对齐模式下开始计数时，当前的配置会被使用 – 计数开始值为写入 TIM1CNTRH/L 中的值，计数开始方向决定于写入 TIM1CR1 寄存器中的 T1DIR 位。注意 T1DIR 位和 T1CMS 值不能被软件同时改写。
- T1DIR 位在 T1CMS 不等于 00 时，为只读寄存器，无法进行写操作；所以如果想要配置计数器的初始计数方向，需要先配置计数方向(T1DIR)，再配置计数模式(TICMS)。
- 运行在中心对齐模式下时，不建议写计数器值(TIM1CNTRH/L)，因为可能会产生意想不到的结果。

  如果写入计数器的值大于自动加载值(TIM1_CNT > T1ARR)，计数方向可能不会进行更新。

  如果写入计数器的值为 0 或为 T1ARR，计数方向会进行更新但更新事件(UEV)不会产生。

配置步骤示例：

1. 使能 TIM1 模块时钟并选择 TIM1 时钟源
2. 配置计数周期预加载使能(T1ARPE=1)
3. 配置计数周期(T1ARR=06H)，占空比
4. 配置初始计数方向为向上计数(T1DIR=0)
5. 配置计数模式为中央对齐模式 1(T1CMS=01)
6. 配置预分频(T1PSC=0)
7. 使能计数器

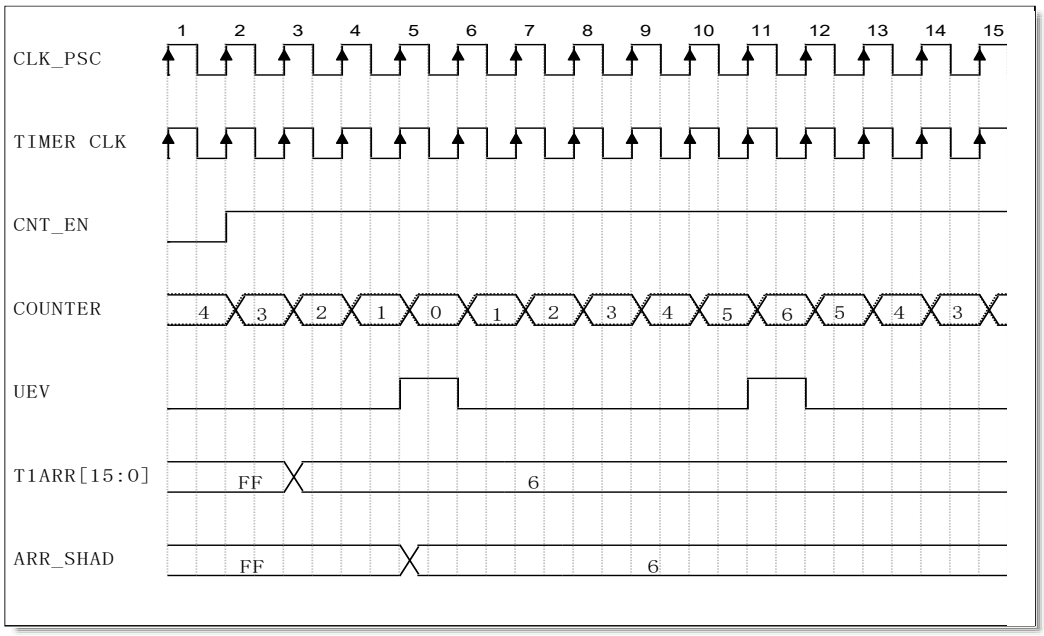


图 9-13　中心对齐模式下，计数时序图

9.3.1.5. 重复向下计数器

重复计数器是 8bit 的向下计数器，会在每次 timer 上溢或下溢时-1；只有当重复向下计数器减到 0 时，计数器上溢或下溢才会产生更新事件(UEV)；使用重复计数器能够设定更新事件的频率，这在产生特定数量 PWM 信号时非常有用，如图 10.14 所示。

重复向下计数器自减事件：

- 计数器向上计数模式下，每个计数上溢事件都会使重复计数器减 1。
- 计数器向下计数模式下，每个计数下溢事件都会使重复计数器减 1。
- 计数器中心对齐模式下，每个计数上溢或下溢事件都会使重复计数器减 1。

重复向下计数器是自动重载的，当发生了更新事件(UEV)时，会将 TIM1RCR 寄存器中的值自动重载到重复向下计数器中。如下图所示：

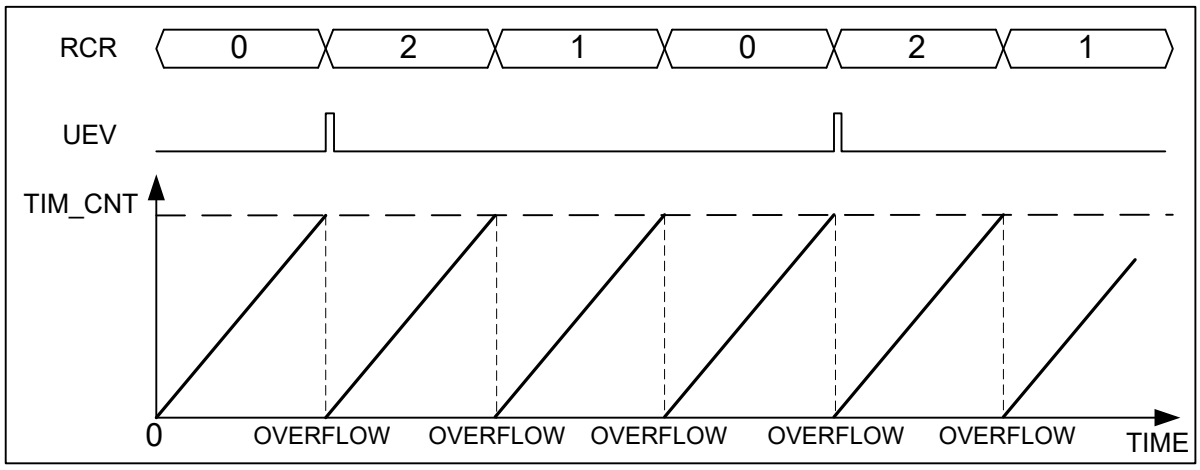


图 9-14 T1REP = 2，重复计数器计数时序图

配置产生特定个数 PWM 信号的步骤示例：

1. 使能 TIM1 模块时钟并选择 TIM1 时钟源
2. 配置 TIM1 通道相对应的端口为输出端口
3. 开启更新事件中断
4. 配置计数周期(T1ARR)，占空比(T1CCRx)
5. 需要开启周期预加载(T1ARPE)和占空比预加载功能(T1OcxPE)
6. 配置计数方向为向上计数(T1DIR=0)
7. 配置输出比较模式(T1OcxM=3’b111)为 PWM2 输出模式，并配置通道使能
8. 打开自动主输出使能(T1AOE=1)位
9. 使能计数器
10. 在更新事件中断中，重新更改计数周期，占空比等配置

以下是一段示例代码：

```
        BANKSEL     PCKEN
        BSR         PCKEN,0             ; 使能 TIM1 模块时钟

        BANKSEL     INTCON
        LDWI        H'C0'
        STR         INTCON              ; 开启全局中断使能和外设中断使能

        BANKSEL     TCKSRC
        LDWI        H'01'
        STR         TCKSRC              ; 选择 TIM1 时钟源为 HIRC

        BANKSEL     TRISA
        LDWI        H'FE'
        STR         TRISA               ; 配置 PA0 为通道 1 的输出通道

        BANKSEL     TIM1ARRL            ;
        LDWI        H'1F'
        STR         TIM1ARRL            ; 将输出波形周期配置为 32

        LDWI        H'10'
        STR         TIM1CCR1L           ; 将输出波形占空比配置为 16

        LDWI        H'02'
        STR         TIM1RCR             ; 将重复计数器配置为 2

        BSR         TIM1BKR,6           ; 打开自动主输出使能位

        BANKSEL     TIM1CCMR1
        LDWI        H'70'
        STR         TIM1CCMR1           ; 配置通道 1 为 PWM2 模式输出

        BSR         TIM1IER,0           ; 开启更新事件中断

        LDWI        H'01'               ;
        STR         TIM1CCER1           ; 使能通道 1

        BANKSEL     TIM1CR1
        LDWI        H'81'               ; 开启计数器计数使能位
        STR         TIM1CR1             ; 开启计数器计数使能位和周期预加载使能位
INT:
        BANKSEL     TIM1ARRL            ;
        LDWI        H'14'
        STR         TIM1ARRL            ; 将输出波形周期配置为 20
```

上述示例对应示意图：

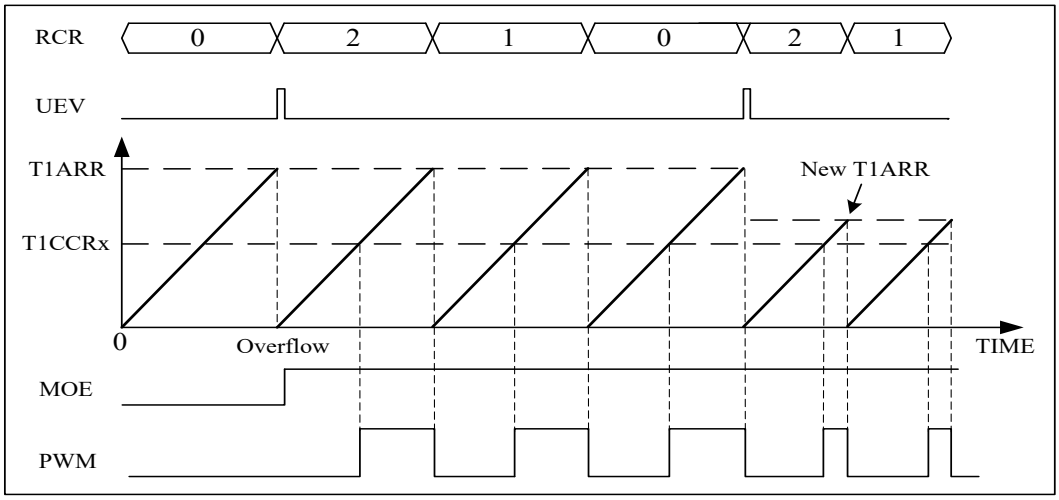


图 9-15 利用重复计数器输出 3 个特定的 PWM 的时序图

注意：

由于重复计数器只有在周期更新事件(UEV)发生时才重载 T1REP 值，对 TIM1RCR 寄存器写入的新值只在下次周期更新事件发生时才起作用，所以建议当配置 T1REP 不为 0 时，在第一个更新事件之后再打开更新事件中断。

9.3.2. 计数控制器

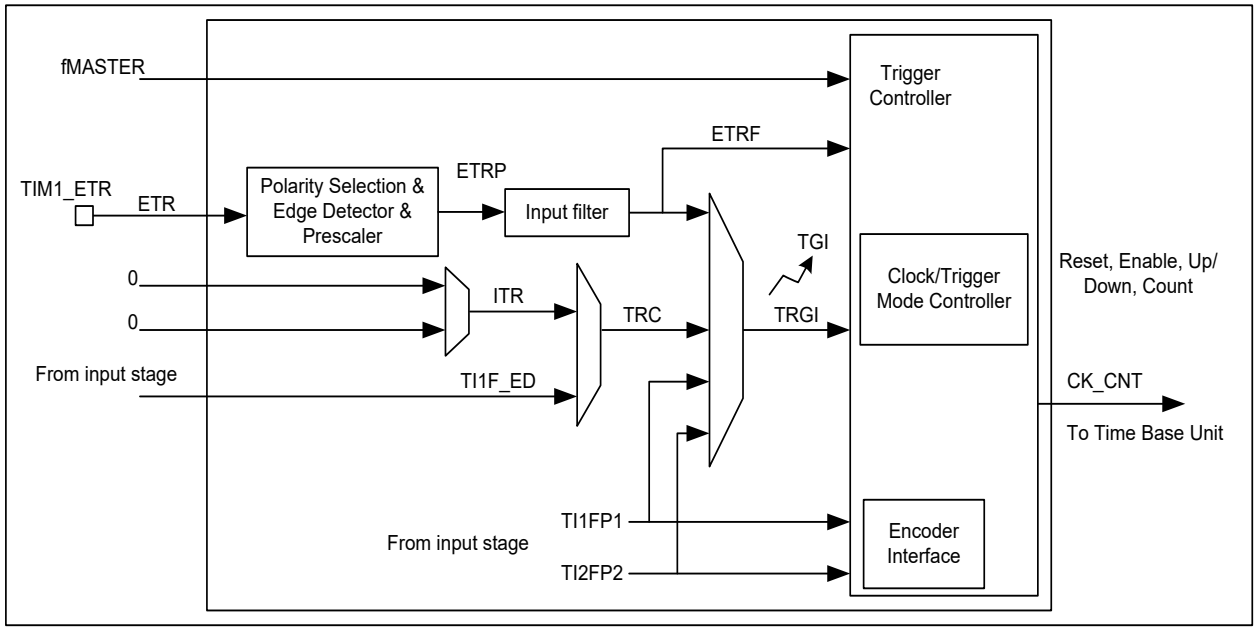


图 9-16 时钟/触发控制器框图

时钟/触发控制器允许配置各种计数器时钟源，输入触发和输出触发。

9.3.2.1. 计数器时钟源

计数器总共有 3 种时钟源类别的选择：

- 内部时钟源(CK_CNT)
- 外部时钟源(ETR)
- 输入触发源(TRGI)

其中 ETR 是外部输入的异步信号，由于时钟同步的需求，因此 ETR 的时钟周期必须大于内部的计数器时钟周期的 2 倍。

计数器的计数时钟（CK_CNT）可由 TCKSRC 寄存器进行选择，总共有以下 8 种时钟源：

- 系统时钟/主时钟
- HIRC
- XT 时钟/外部时钟
- HIRC 的 2 倍频
- XT 时钟/外部时钟的 2 倍频
- LIRC

- LP 时钟/外部时钟
- LP 时钟/外部时钟的 2 倍频

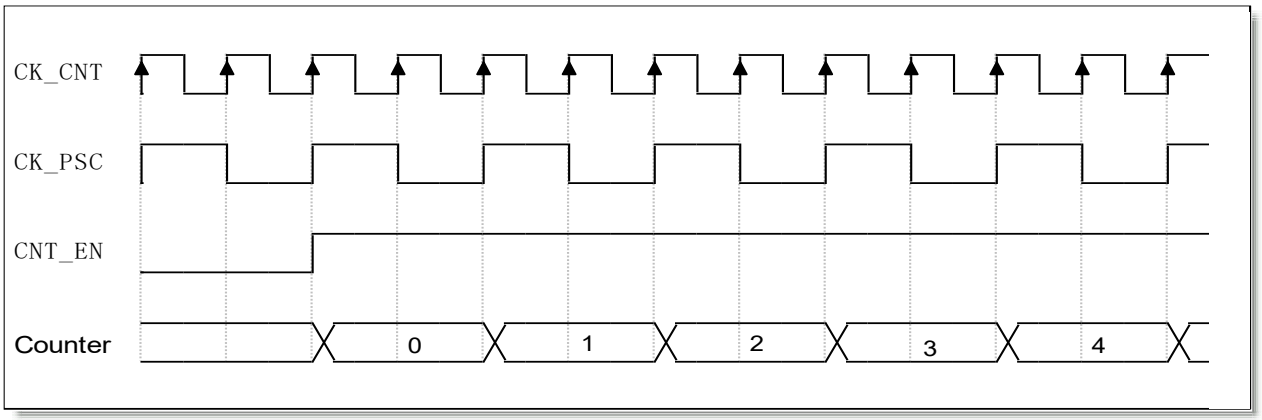


图 9-17 内部时钟源且预分频为 1 时，计数器计数时序图

9.3.2.2. 计数触发源

本章节所述的触发源是计数触发源,而非捕捉源;计数触发源来自 ETR 端口和通道 1/2 对应的输入端口，来自通道 1/2 的计数触发源与通道 1/2 的捕捉源是一致的；计数触发源不能来自通道 3/4，通道 3/4 的输入只能作为对应的通道捕捉源。关于捕捉源的详细描述可查看 9.4.3.1 章节内容。

当 T1SMS=000 时,计数由内部时钟驱动,使能 T1CEN 即可触发计数;但是在计数控制模式(T1SMS !=0)下，需要触发源，这些触发源由 TIM1CR1 寄存器中的 T1TS[2:0]位进行选择，总共有以下 4 种计数触发源：

- 输入源 T1 的边沿检测(TI1F_ED)；
- 滤波后的通道 1 输入(TI1FP1)；
- 滤波后的通道 2 输入(FI2FP2)；
- 外部触发输入(ETRF)；

9.3.2.3. 计数控制模式选择

TIM1 除了有向上计数、向下计数、中央对齐计数方式之外，还有 4 种计数控制模式，需要计数触发源的配合使用；计数模式的选择由 TIM1SMCR 寄存器中的 T1SMS[2:0]去控制，下列是 9 种计数模式：

1. 内部时钟模式：计数由内部时钟(CK_CNT)驱动。
2. 译码模式 1：根据 TI1FP1 的电平，在 TI2FP2 的边沿进行上计数或下计数；
3. 编码模式 2：根据 TI2FP2 的电平，在 TI1FP1 的边沿进行上计数或下计数；
4. 译码模式 3：根据其他输入电平，在 TI1FP1 和 TI2FP2 的边沿进行上计数或下计数；
5. 复位模式： 在选中的触发输入(TRGI)的上升沿时重新初始化计数器，并且产生一个更新寄存器的信号。
6. 门控模式：当触发输入(TRGI)为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止(但不复位)。计数器的启动和停止都是受控的。

7. 触发模式：计数器在触发输入 TRGI 的上升沿启动(但不复位)，只有计数器的启动是受控的。
8. 外部时钟模式 1：计数由触发输入(TRGI)信号进行驱动
9. 外部时钟模式 2：计数由外部触发输入信号(ETRF)进行驱动

各个模式更详细的描述参照 TIM1SMCR 寄存器的描述，使用各种控制模式时需注意以下事项：

1. 在配置复位模式和触发模式时，可以通过配置 T1CCxNP/T1CCxP 来配置选择触发的边沿，具体参照 T1CCxNP/T1CCxP 的描述。
2. 译码模式下，其输入的信号可以选择极性，通过配置 T1CCxP 进行选择。
3. 门控模式只能选择部分触发源；因为输入源 T1 的边沿检测(TI1F_ED)是脉冲式触发源，所以不能在门控模式下选择为触发源。
4. 外部时钟模式 1 比外部时钟模式 2 的优先级更低。

- 内部时钟模式：

内部时钟 (CK_CNT) 模式下，在软件配置计数器使能 (T1CEN) 之后，计数器开始由内部时钟 (CK_CNT) 驱动下进行计数；如**图 9-17** 所示。

- 编码器模式：

编码器模式主要应用在电机控制。总共有以下 3 种计数方式：

1. 当 T1SMS=001 时，计数器只在 TI2 沿计数
2. 当 T1SMS=010 时，计数器只在 TI1 沿计数
3. 当 T1SMS=011 时，计数器在 TI1 沿和 TI2 沿计数

详情如下表所示：

有效计数沿	相对信号的电平值(TI1FP1 for TI2) (TI2FP2 for TI1)	TI1FP1 信号		TI2FP2 信号	
		上升	下降	上升	下降
TI1	高	——	——	向下	向上
	低	——	——	向上	向下
TI2	高	向上	向下	——	——
	低	向下	向上	——	——
TI1 or TI2	高	向上	向下	向下	向上
	低	向下	向上	向上	向下

表 9-18 计数方向 vs 译码信号

编码器模式的工作方式像是带方向选择的外部时钟。根据 TI1 和 TI2 信号的组合，计数器在 0 至 T1ARR 之间连续的进行计数，计数方向是随着 TI1 和 TI2 信号的组合不断变化的。在这个模式下，捕捉功能，比较功能和输出都能正常工作；预分频功能在此模式下不起作用并且此模式无法与外部时钟模式 2 兼容。

编码模式的步骤示例：

1. 配置输入捕捉寄存器的值 – 配置输入捕捉滤波器；配置捕捉预分频器
2. 将通道配置为输入捕捉通道 T1CC1S=01，并将 IC1 映射在 TI1FP1 上
   将通道配置为输入捕捉通道 T1CC2S=01，并将 IC2 映射在 TI2FP2 上
3. 写 T1CC1P=0，选择 IC1 极性不取反，IC1=TI1；写 T1CC2P=0，选择 IC2 极性不取反，IC2=TI2
4. 通过写 T1SMS=011，将 TIM1 配置为编码器 3 模式，上升沿与下降沿均可触发计数
5. 置位 T1CEN，启动计数器

以下是一段示例代码：

```
BANKSEL     PCKEN
BSR         PCKEN,0             ; 使能 TIM1 模块时钟

BANKSEL     TCKSRC
LDWI        H'01'
STR         TCKSRC              ; 选择 TIM1 时钟源为 HIRC

BANKSEL     TRISA
LDWI        H'FF'
STR         TRISA               ; 配置 PA0 为通道 1 的输入通道,
                                ; PA1 为通道 2 的输入通道

BANKSEL     TIM1CCMR1
LDWI        H'01'
STR         TIM1CCMR1           ; 配置通道 1 为 IC1 映射在 TI1FP1 上

LDWI        H'02'
STR         TIM1CCMR2           ; 配置通道 2 的 IC2 映射在 TI1FP2 上

LDWI        H'53'
STR         TIM1SMCR            ; 配置 TIM1 为编码器 3 模式

LDWI        H'11'
STR         TIM1CCER1           ; 使能通道 1/2

BANKSEL     TIM1CR1
BSR         TIM1CR1,0           ; 开启计数器计数使能位
```

上述示例代码对应示意图：

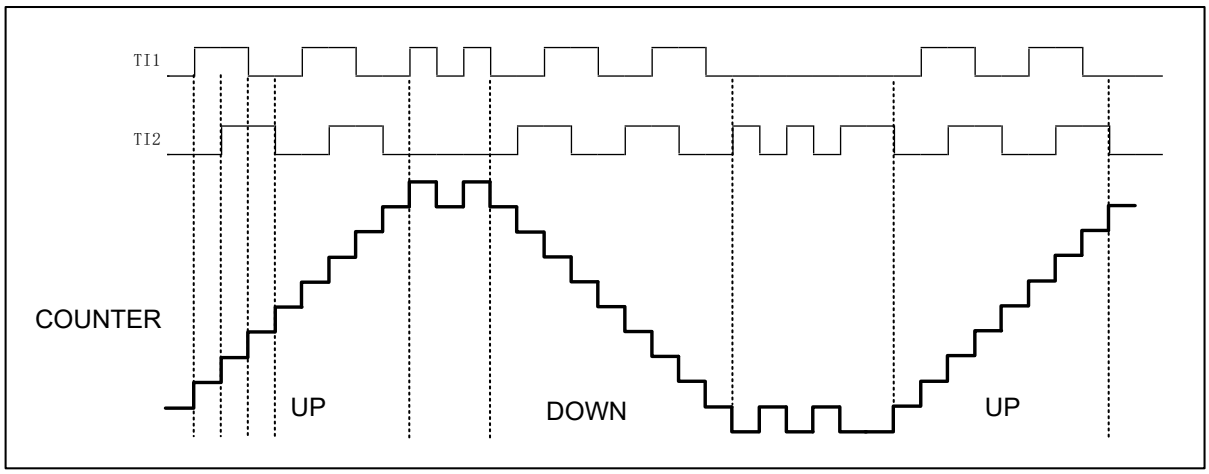


图 9-18 编码器模式 3 下，计数器计数示意图

- 复位模式：

当触发输入事件到来时，计数器和计数器预分频都会被初始化。如果此时 T1URS 为 0 且 T1UDIS 也为 0，则会产生一个更新事件，同时所有的预加载寄存器都会被更新。

复位模式的步骤示例：

1. 配置输入捕捉寄存器的值 – 配置输入捕捉滤波器 T1IC1F=000；配置捕捉预分频器 T1IC1PSC=0
2. 将通道配置为输入捕捉通道 T1CC1S=01，并将 IC1 映射在 TI1FP1 上
3. 写 T1CC1P=0，选择检测触发上升沿的到来
4. 通过写 T1SMS=100，将 TIM1 配置为复位模式。同时写 T1TS=101，选择 TI1 为输入触发源
5. 置位 T1CEN，启动计数器

以下是一段示例代码：

```
BANKSEL     PCKEN           ;
BSR         PCKEN,0         ; 使能 TIM1 模块时钟

BANKSEL     TCKSRC          ;
LDWI        H'01'           ;
STR         TCKSRC          ; 选择 TIM1 时钟源为 HIRC

BANKSEL     TRISA           ;
LDWI        H'FF'           ;
STR         TRISA           ; PA0 为通道 1 的输入通道

BANKSEL     TIM1CCMR1       ;
LDWI        H'01'           ;
STR         TIM1CCMR1       ; 配置通道 1 的 IC1 映射在 TI1FP1 上

LDWI        H'54'           ;
STR         TIM1SMCR        ; 配置 TIM1 为复位模式，触发源为 TI1FP1
LDWI        H'01'           ;
STR         TIM1CCER1       ; 使能通道 1 并且为上升沿触发

BANKSEL     TIM1CR1         ;
BSR         TIM1CR1,0       ; 开启计数器计数使能位
```

当 TI1 的上升沿到来时，计数器被清 0 并从 0 开始重新计数。与此同时，触发标志位 (TIF) 会被置位，在触发中断使能的情况下还会产生一个中断请求。如下列示例对应时序图所示：

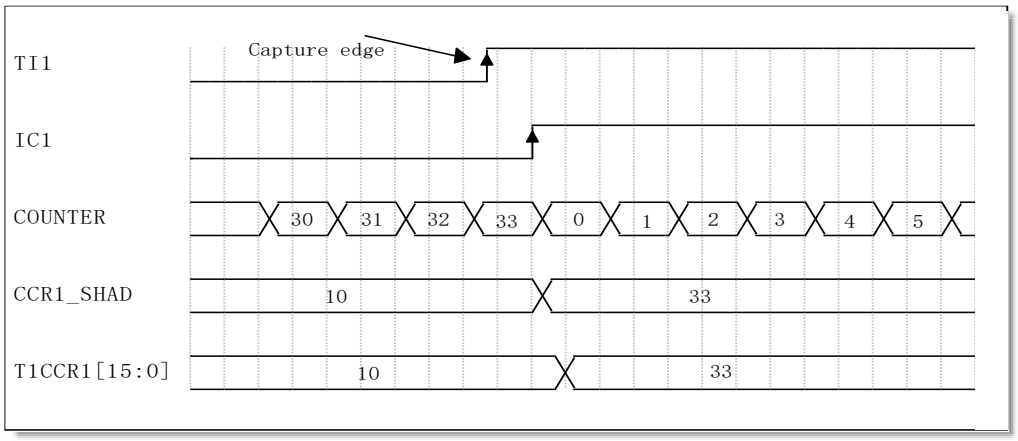


图 9-19 复位模式下，计数器计数时序图

- 门控模式：

依据选择的触发输入的电平值，计数器会被使能。此模式下，计数器的运行和停止都是受控的。

门控模式的步骤示例：

1. 使能 TIM1 模块时钟并选择 TIM1 时钟源
2. 配置输入捕捉寄存器的值 – 配置输入捕捉滤波器 T1IC1F=000；配置捕捉预分频器 T1IC1PSC=0
3. 将通道配置为输入捕捉通道 T1CC1S=01，并将 IC1 映射在 TI1FP1 上
4. 写 T1CC1P=1，选择检测输入低电平的到来
5. 通过写 T1SMS=101，将 TIM1 配置为门控模式。同时写 T1TS=101，选择 TI1 为输入源
6. 置位 T1CEN，使能计数器(在门控模式下，需要开启 T1CEN；在此基础上，才能由输入源控制计数器的运行与停止)

以下是一段示例代码：

```
BANKSEL     PCKEN           ;
BSR         PCKEN,0         ; 使能 TIM1 模块时钟

BANKSEL     TCKSRC          ;
LDWI        H'01'           ;
STR         TCKSRC          ; 选择 TIM1 时钟源为 HIRC

BANKSEL     TRISA           ;
LDWI        H'FF'           ;
STR         TRISA           ; PA0 为通道 1 的输入通道

BANKSEL     TIM1CCMR1       ;
LDWI        H'01'           ;
STR         TIM1CCMR1       ; 配置通道 1 的 IC1 映射在 TI1FP1 上

LDWI        H'55'           ;
STR         TIM1SMCR        ; 配置 TIM1 为门控模式，触发源为 TI1FP1

LDWI        H'03'           ;
STR         TIM1CCER1       ; 使能通道 1 并且低电平输入为有效电平

BANKSEL     TIM1CR1         ;
BSR         TIM1CR1,0       ; 开启计数器计数使能位

BTSS        TIM1SR1,6       ; 判断触发中断标志位是否为高
LJUMP       $-1             ;

BCR         TIM1SR1,6       ; 将触发中断标志位清零
```

当 TI1 为低电平时，计数器在内部时钟的驱动下进行计数；当 TI1 变为高电平时，计数器停止计数。触发标志位 (T1TIF) 会在计数器启动或停止时被置位。如下列示例对应时序图所示：

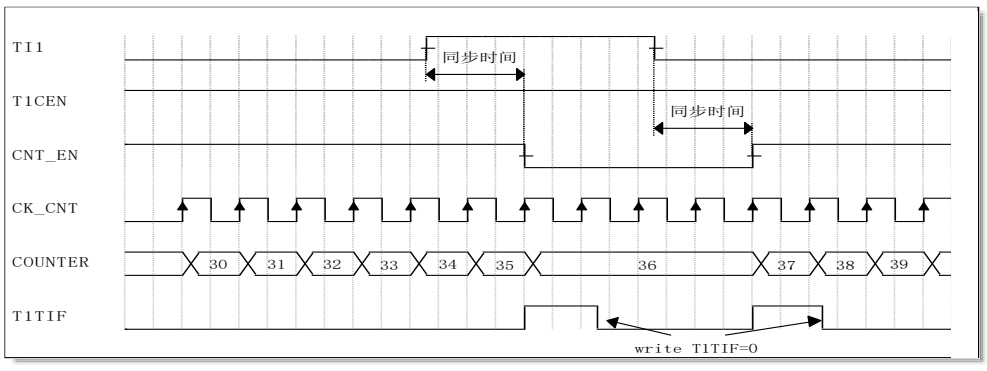


图 9-20 门控模式下，计数器计数时序图

- 触发模式：

依据选择的触发输入的电平值，计数器会被启动(T1CEN 被置位)。

触发模式的步骤示例：

1. 使能 TIM1 模块时钟并选择 TIM1 时钟源
2. 配置输入捕捉寄存器的值 – 配置输入捕捉滤波器 T1IC2F=000；配置捕捉预分频器 T1IC2PSC=0
3. 将通道配置为输入捕捉通道 T1CC2S=01，并将 IC2 映射在 TI2FP2 上
4. 写 T1CC2P=0，选择检测触发上升沿的到来
5. 通过写 T1SMS=110，将 TIM1 配置为触发模式。同时写 T1TS=110，选择 TI2 为输入触发源

以下是一段示例代码：

```
BANKSEL     PCKEN           ;
BSR         PCKEN,0         ; 使能 TIM1 模块时钟

BANKSEL     TCKSRC          ;
LDWI        H'01'           ;
STR         TCKSRC          ; 选择 TIM1 时钟源为 HIRC

BANKSEL     TRISA           ;
LDWI        H'FF'           ;
STR         TRISA           ; PA1 为通道 2 的输入通道

BANKSEL     TIM1CCMR2       ;
LDWI        H'01'           ;
STR         TIM1CCMR2       ; 配置通道 2 的 IC2 映射在 TI2FP2 上

LDWI        H'66'           ;
STR         TIM1SMCR        ; 配置 TIM1 为触发控制模式，触发源为 TI2FP2

LDWI        H'10'           ;
STR         TIM1CCER1       ; 使能通道 2 并且为上升沿触发

BANKSEL     TIM1CR1         ;
BSR         TIM1CR1,0       ; 开启计数器计数使能位
```

当 TI2 的上升沿到来时，计数器在内部时钟的驱动下启动计数，并且触发标志位 (T1TIF) 被置位。如下列示例对应时序图所示：

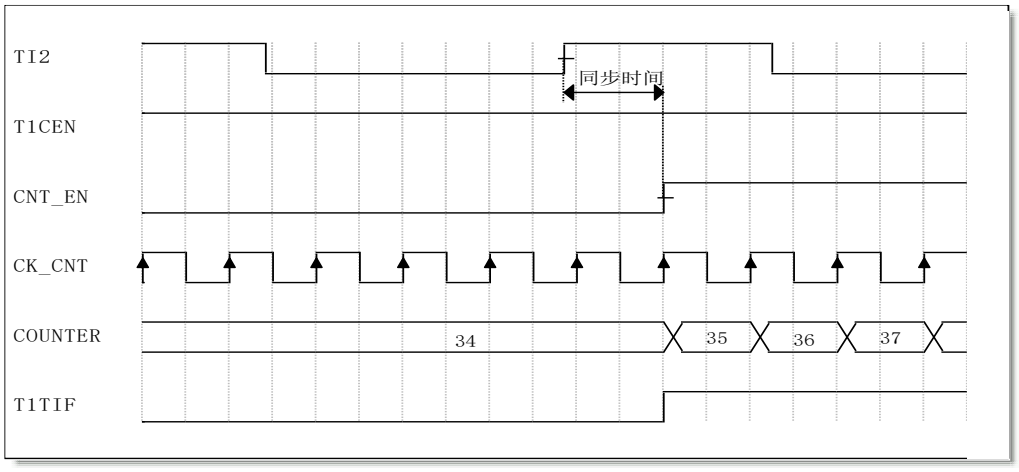


图 9-21 触发模式下，计数器计数时序图

- 外部时钟模式 1：

当 T1SMS = 111 时，此模式被选中。计数器会在每个输入触发的上升沿或下降沿进行计数。

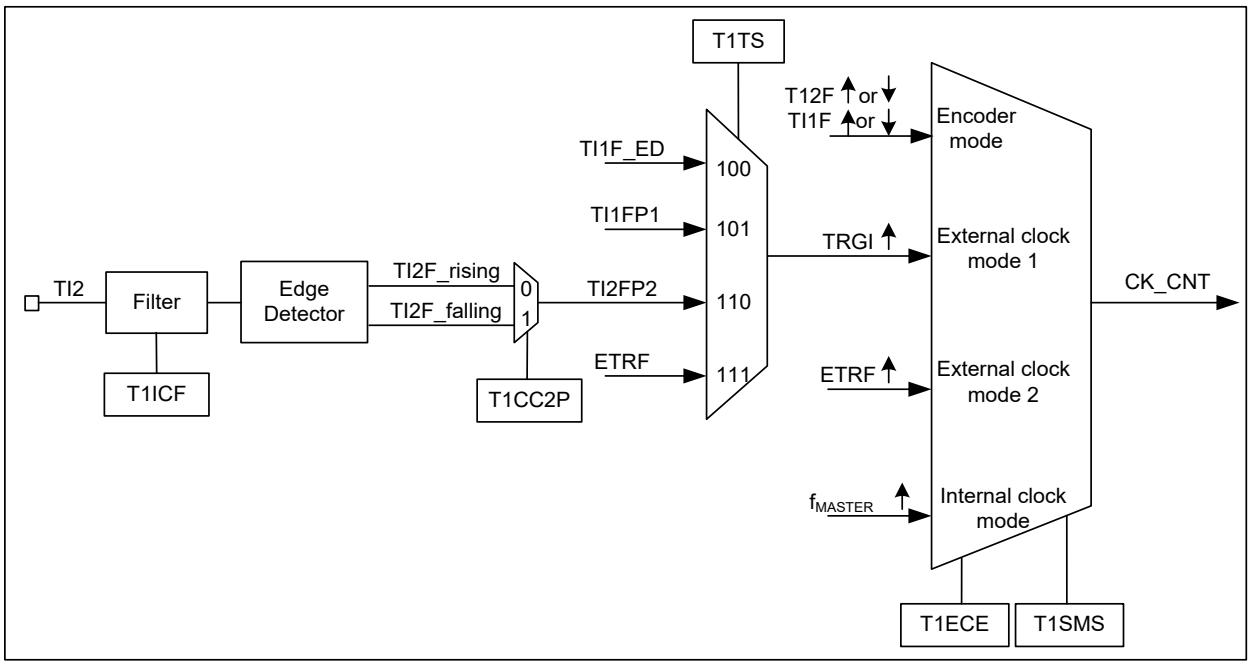


图 9-22 外部时钟模式 1 时，TI2 作为计数时钟源

外部时钟模式 1 的步骤示例：

1. 配置输入捕捉寄存器的值 – 配置输入捕捉滤波器 T1IC1F=000；配置捕捉预分频器 T1IC1PSC=0
2. 将通道配置为输入捕捉通道 T1CC1S=01，并将 IC1 映射在 TI1FP1 上
3. 写 T1CC1P=0，选择检测触发上升沿的到来
4. 通过写 T1SMS=111，将 TIM1 配置为外部时钟模式 1；同时写 T1TS=110，选择 TI1 为输入源
5. 置位 T1CEN，使能计数器

以下是一段示例代码：

```
BANKSEL     PCKEN               ;
BSR         PCKEN,0         ; 使能 TIM1 模块时钟

BANKSEL     TCKSRC              ;
LDWI        H'01'               ;
STR         TCKSRC          ; 选择 TIM1 时钟源为 HIRC

BANKSEL     TRISA               ;
LDWI        H'FF'               ;
STR         TRISA           ; PA0 为通道 1 的输入通道

BANKSEL     TIM1CCMR1           ;
LDWI        H'01'                   ;
STR         TIM1CCMR1           ; 配置通道 1 的 IC1 映射在 TI1FP1 上

LDWI        H'57'                   ;
STR         TIM1SMCR            ; 配置 TIM1 为外部时钟模式 1，触发源为 TI1FP1

LDWI        H'01'                   ;
STR         TIM1CCER1           ; 使能通道 1
```

```
BANKSEL     TIM1CR1             ;
BSR         TIM1CR1,0           ; 开启计数器计数使能位
BTSS        TIM1SR1,6       ; 判断触发中断标志位是否为高
LJUMP       $-1             ;
BCR         TIM1SR1,6           ; 将触发中断标志位清零
```

当 TI1 上升沿到来时，计数器进行一次计数，并且触发标志位(T1TIF)被置位；如果触发中断使能(T1TIE)，则会产生一个中断请求。如下列示例对应示意图所示：

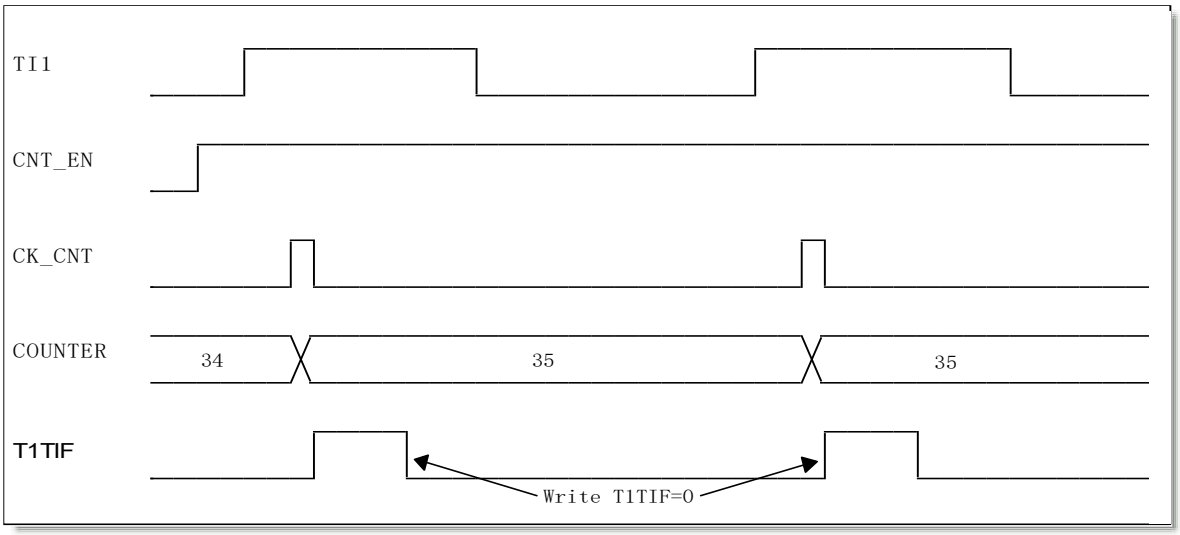


图 9-23 外部时钟模式 1 下，计数器计数示意图

- 外部时钟模式 2：

当 T1ECE=1 时，此模式被选中。计数器会在每个输入触发的上升沿或下降沿进行计数。

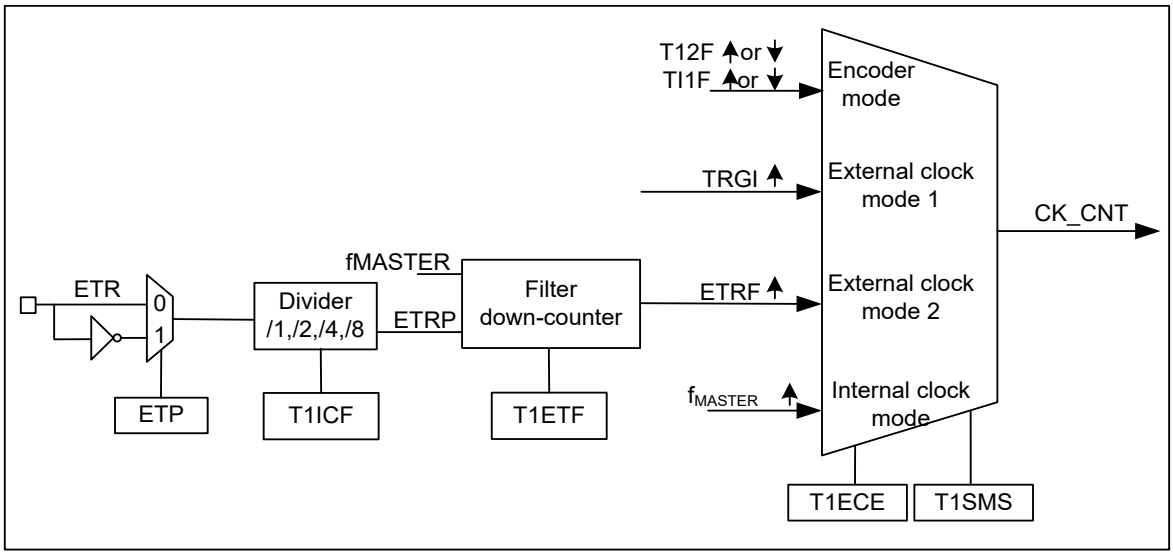


图 9-24 外部时钟模式 2 时，外部时钟源(ETR)作为计数时钟源

外部时钟模式 2 的步骤示例：

1. 配置外部触发寄存器的值 – 配置外部触发滤波器 T1ETF=000；配置外部触发预分频器 T1ETPS=1
2. 写 T1ETP=0，选择上升沿为检测沿
3. 通过写 T1ECE=1，将 timer 配置为外部时钟模式 2
4. 置位 T1CEN，使能计数器

以下是一段示例代码：

```
BANKSEL     PCKEN                ;
BSR         PCKEN,0                ; 使能 TIM1 模块时钟
BANKSEL     TCKSRC               ;
LDWI        H'01'                  ;
STR         TCKSRC                 ; 选择 TIM1 时钟源为 HIRC
BANKSEL     TRISB                  ;
LDWI        H'FF'                  ;
STR         TRISB                    ; PB3 为 ETR 外部信号的输入通道
BANKSEL     TIM1ETR              ;
LDWI      H'50'                ;
STR         TIM1ETR              ; 配置外部触发预分频为 1，使能外部时钟模式 2
BANKSEL     TIM1CR1                ;
BSR       TIM1CR1,0              ; 开启计数器计数使能位
```

ETR 信号上升沿和真正产生计数时钟之间的延时是因为外部输入信号 ETR 需要进行时钟同步；如下列示例对应示意图所示：

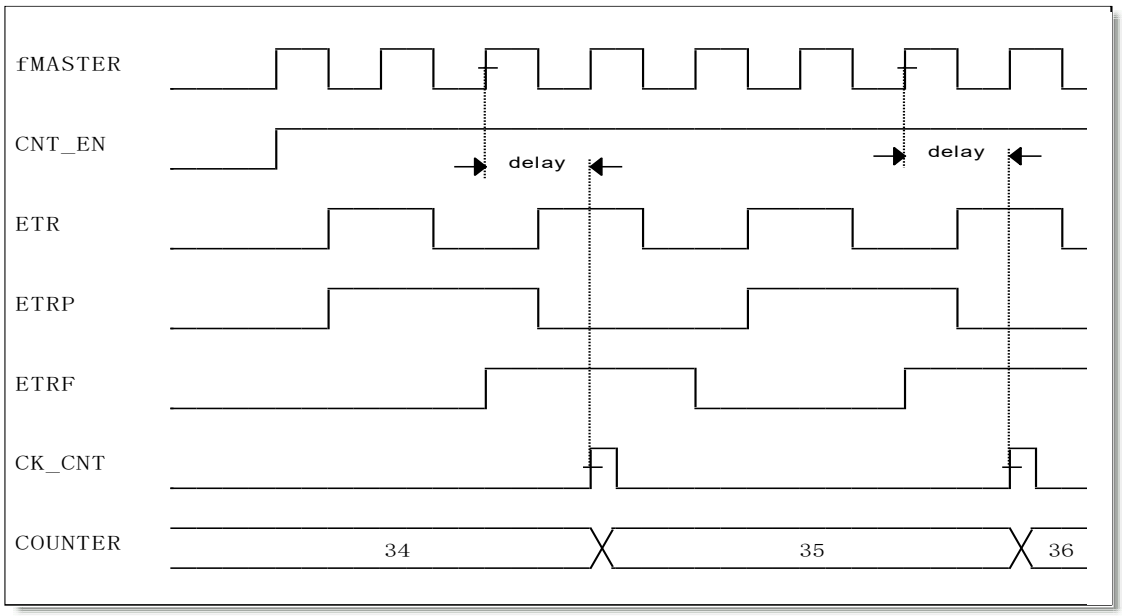


图 9-25 外部时钟模式 2 下，计数器计数示意图

9.3.3. 捕捉比较通道

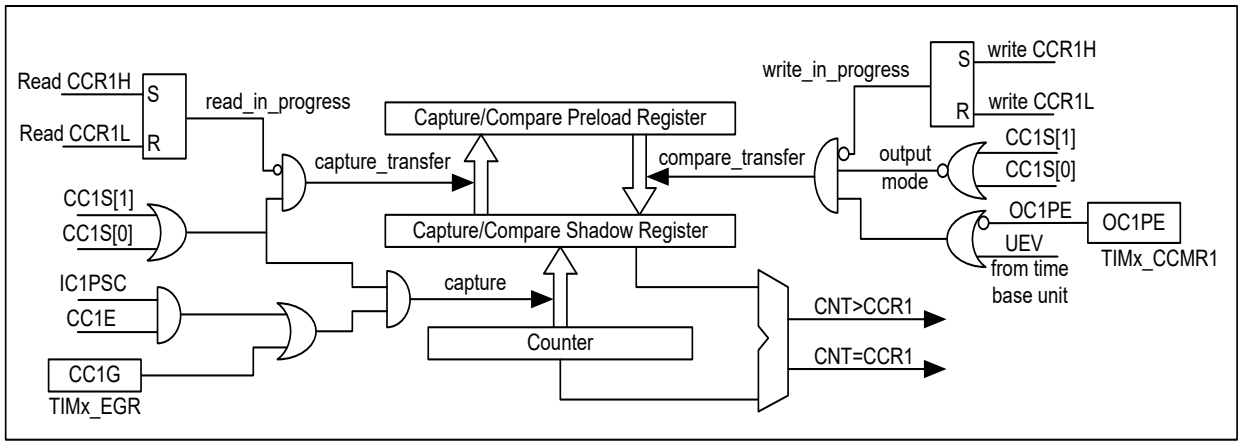


图 9-26 捕捉/比较通道 1 框图

TIMER 的 I/O 口能被配置为输入捕捉或输出比较功能。这个配置由 T1CCxS 通道选择位进行设定；对单个通道而言，输入捕捉功能和输出比较功能是互斥的两个功能。但每个通道都有独立的配置寄存器，所以可以将某些通道配置为输入捕捉功能，另一些通道配置为输出比较功能；例如：配置 T1CC1S=2'b00，T1CC2S=2'b00，T1CC3S=2'b01，T1CC3S=2'b10，这样通道 1 和通道 2 为输出比较通道，可在输出波形，而通道 3 和通道 4 为输入捕捉通道，可进行捕捉功能。

TIM1CCRxH/L 寄存器的读写：

TIM1CCRxH/L 寄存器由一个预加载寄存器和一个影子寄存器组成。

TIM1CCRxH/L 寄存器在输出比较模式和输入捕捉模式下的读写有所不同；在输出比较模式下，为可读可写寄存器；而在输入捕捉模式下，为只读寄存器。

- 在输出比较模式下：

  TIM1CCRxH/L 寄存器的访问没有任何限制，可读可写。

  读 TIM1CCRxH/L： 读出的值来自 CCRx 预加载寄存器的值，跟先前写入 TIM1CCRxH/L 寄存器的值保持一致。

  写 TIM1CCRxH/L：有预加载使能位(T1OcxPE)；如果预加载使能(T1OcxPE=0)关闭，则写入 TIM1CCRxH/L 寄存器的值直接由 CCRx 预加载寄存器传递到 CCRx 影子寄存器。反之，写入 TIM1CCRxH/L 寄存器的值在下一次更新事件发生时才会从 CCRx 预加载寄存器传递到 CCRx 影子寄存器。

- 在输入捕捉模式下：

  TIM1CCRxH/L 寄存器为只读寄存器。在捕捉事件发生时，计数器值会被写入到 CCRx 影子寄存器中，而后再写回到 CCRx 预加载寄存器中。

  读 TIM1CCRxH/L 寄存器时，必须先读高 8 位，再读低 8 位。读高 8 位时，CCRx 预加载寄存器被冻结，此时计数器值无法写回到 CCRx 预加载寄存器中；只有按顺序读完低 8 位后，CCRx 预加载寄存器才能更新为最新一次捕捉值。

TIM1CCMRx 寄存器为复用寄存器。详见**章节 0**

9.3.3.1. 捕捉输入通道

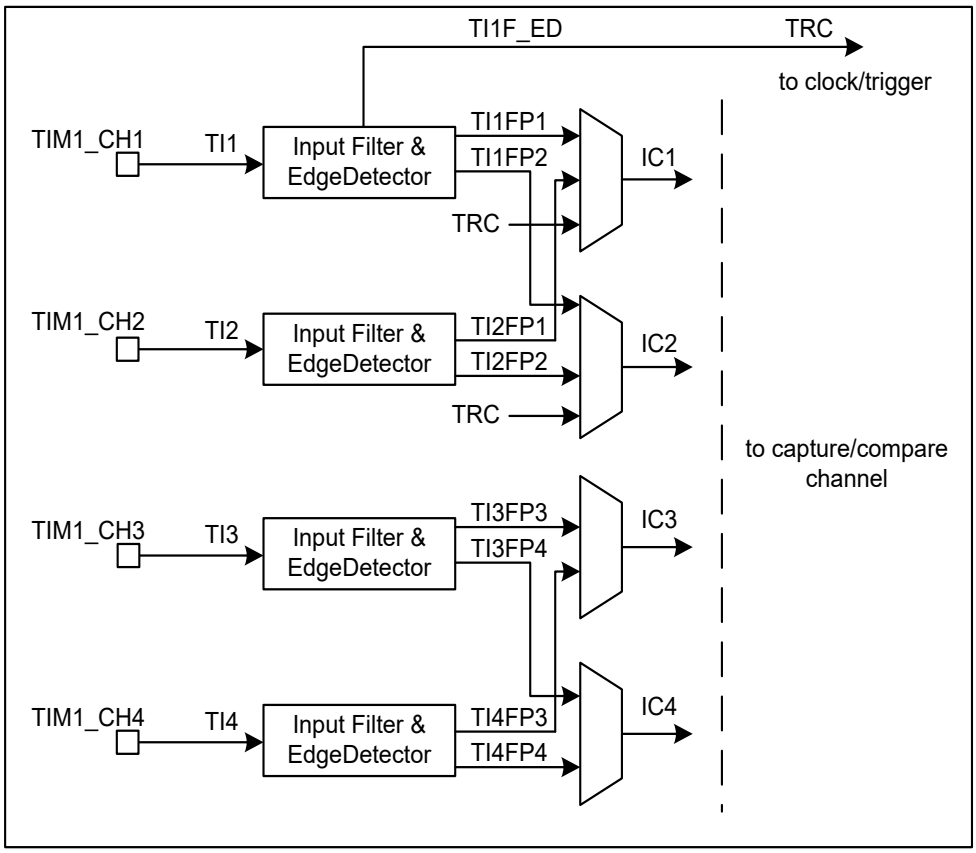


图 9-27 通道输入框图

信号名称	详细说明
TIM1_CH1/2/3/4	通道 1/2/3/4 对应 I/O 口的输入
IC1/2/3/4	通过选择后的真正的通道捕捉源
TI1FP1	来自通道 1 对应 I/O 的输入捕捉信号，作为通道 1 的捕捉源之一
TI1FP2	来自通道 1 对应 I/O 的输入捕捉信号，作为通道 2 的捕捉源之一
TI2FP2	来自通道 2 对应 I/O 的输入捕捉信号，作为通道 2 的捕捉源之一
TI2FP1	来自通道 2 对应 I/O 的输入捕捉信号，作为通道 1 的捕捉源之一
TI3FP3	来自通道 3 对应 I/O 的输入捕捉信号，作为通道 3 的捕捉源之一
TI3FP4	来自通道 3 对应 I/O 的输入捕捉信号，作为通道 4 的捕捉源之一
TI4FP4	来自通道 4 对应 I/O 的输入捕捉信号，作为通道 4 的捕捉源之一
TI4FP3	来自通道 4 对应 I/O 的输入捕捉信号，作为通道 3 的捕捉源之一
TRC	来自通道 1 对应 I/O 的输入双沿捕捉信号，作为通道 1 和通道 2 的捕捉源之一

表 9-19 信号说明列表

当一个通道被配置成输入捕捉通道并且输入捕捉事件有效时，可以将当前的计数值保存在 TIM1CCRx 寄存器。每个通道都有一个数字滤波单元，可配置采样频率 (T1ICxF[3:0])，捕捉预分频 (T1IC1PSC[1:0])，捕捉极性选择 (T1CCxP) 和捕捉触发源 (T1CCxS)。每个通道都有各自的捕捉源，如下表所示：

T1CCxS (捕捉源选择)	通道 1	通道 2	通道 3	通道 4
2'b00	TI1FP1	TI2FP2	TI3FP3	TI4FP4
2'b01	TI2FP1	TI1FP2	TI4FP3	TI3FP4
2'b10	TRC	TRC	—	—

表 9-20 各通道输入捕捉源列表

当一个输入捕捉发生时：

- TIM1CCR1H/L 寄存器得到捕捉发生时计数器的值。
- 输入捕捉标志位 (T1CCxIF) 被置位。如果当 T1CCxIF 保持为 1 时，又一次发生了输入捕捉事件，那么溢出捕捉标志位 (T1CCxOF) 也会被置位。
- 如果 T1CCxIE 为 1，那么捕捉将产生一个中断事件。

配置为输入捕捉通道的示例步骤：

1. 使能 TIM1 模块时钟并选择 TIM1 时钟源
2. 将通道相应的端口配置为输入端口
3. 选择输入触发源 (T1CCxS)
4. 配置采样频率 (T1ICxF[3:0])，捕捉预分频 (T1IC1PSC[1:0])
5. 配置捕捉源的捕捉极性 (T1CCxP)
6. 使能捕捉通道 (T1CCxE)
7. 使能计数器 (T1CEN)

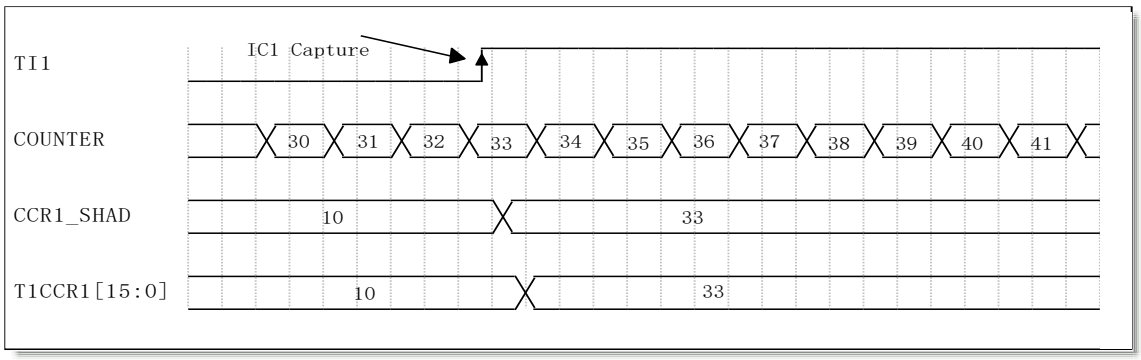


图 9-28 简单输入捕捉时序图

PWM 输入信号测量的应用：

利用捕捉输入模式和复位模式，并且将两个通道的输入捕捉源都选择为同一个通道的 PWM 信号输入；这样就可以测量从通道输入的 PWM 信号的周期以及占空比。

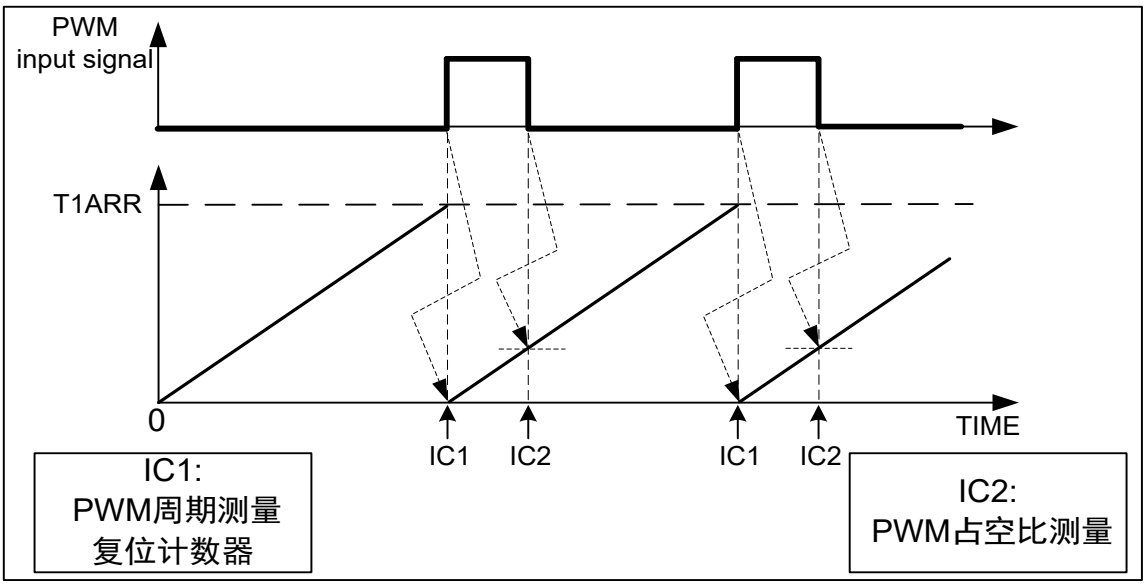


图 9-29 测量 PWM 信号的示意图

具体测量 PWM 的配置步骤如下：

1. 使能 TIM1 模块时钟并选择 TIM1 时钟源
2. 将通道 1/2 相应的端口配置为输入端口
3. 通道 1 配置将 IC1 映射在 TI1FP1 上；通道 2 配置将 IC2 映射在 TI2FP1 上
4. 配置通道 1 为上升沿捕捉(T1CC1P=0)；通道 2 为下降沿捕捉(T1CC2P=1)
5. 配置采样频率(T1IcxF[3:0]=4’b0000)，捕捉预分频(T1IC1PSC[1:0]=2’b00)
6. 将计数控制模式配置为复位模式(T1SMS=101)，计数触发源配置为 TI1FP1(T1TS=101)
7. 使能计数器(T1CEN)
8. 开启通道 1 和通道 2 的输入捕捉功能(T1CC1E=1 且 T1CC2E=1)

注意：

因为捕捉沿先于复位触发源两个计数时钟周期，所以需要软件进行一下操作才能等到准确测量值：

- 当预分频为 0 时，PWM 周期 = T1CCR1H/L+2，占空比 = T1CCR2H/L+2
- 当预分频为 1 时，PWM 周期 = T1CCR1H/L+1，占空比 = T1CCR2H/L+1
- 当预分频大于 1 时，PWM 周期 = T1CCR1H/L，占空比 = T1CCR2H/

以下是一段示例代码：

```
BANKSEL     PCKEN           ;
BSR         PCKEN,0         ; 使能 TIM1 模块时钟

BANKSEL     TCKSRC          ;
LDWI        H'01'           ;
STR         TCKSRC          ; 选择 TIM1 时钟源为 HIRC

BANKSEL     TRISA           ;
LDWI        H'FF'           ;
STR         TRISA           ; 配置 PA0 为通道 1 的输入通道，PA1 为通道 2 的输入通道

BANKSEL     TIM1CCMR1       ;
LDWI        H'01'           ;
STR         TIM1CCMR1       ; 配置通道 1 为 IC1 映射在 TI1FP1 上

LDWI        H'02'           ;
STR         TIM1CCMR2       ; 配置通道 2 的 IC2 映射在 TI1FP2 上

LDWI        H'54'           ;
STR         TIM1SMCR        ; 配置 TIM1 为复位控制模式，触发源为 TI1FP1

LDWI        H'31'           ;
STR         TIM1CCER1       ; 使能通道 1/2，通道 1 为上升沿捕捉，通道 2 为下降沿捕捉

BANKSEL     TIM1CR1         ;
BSR         TIM1CR1,0       ; 开启计数器计数使能位
```

上述示例代码对应波形图：

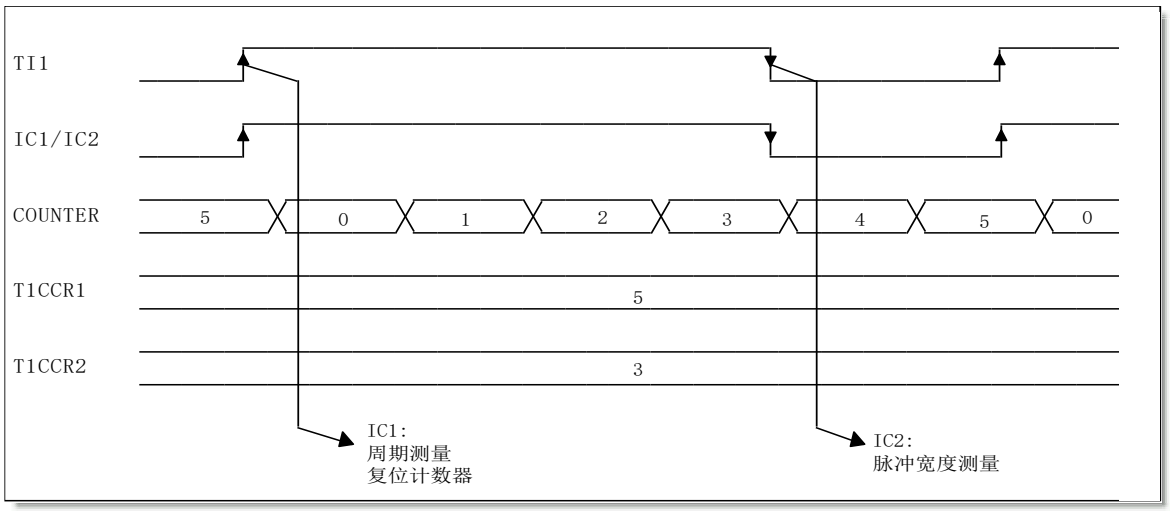


图 9-30 测量 PWM 信号的时序图

9.3.3.2. 输出比较通道

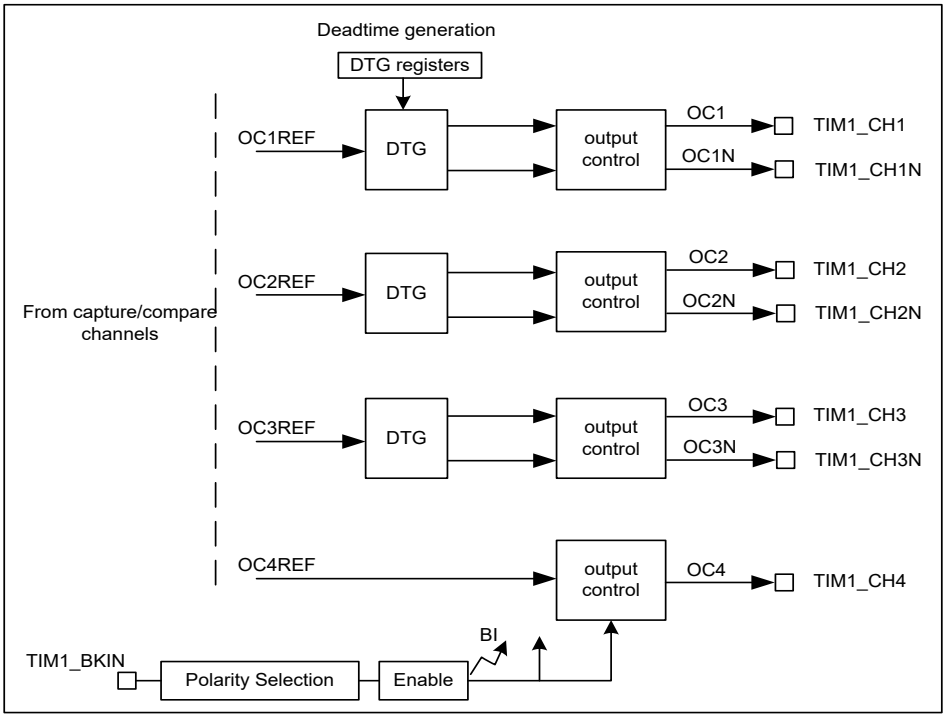


图 9-31 通道输出框图

输出阶段产生立即响应的波形，用来作为参考波形，叫做 OCxREF 信号(高有效)。刹车功能，极性选择和其他输出控制位都在参考波形之后去做控制。

输出比较通道根据计数值与比较值 CCRx，产生 OCxREF 输出并送到死区产生模块，经过死区产生模块后再经过其他输出控制位的控制将波形输出到端口。

具体的输出控制位以及可达到的输出效果可以查看 10.3.3.5 章节内容。

输出比较模式下，可选择不同的输出模式去输出 PWM 波形；输出模式由 T1OcxM[3:0]选择，总共有以下 8 种不同的输出模式(最终的输出还需要取决于极性选择(T1CCxP))：

1. 冻结模式：输出冻结，输出实际比较值(CCRx_SHAD)与计数器 TIM1_CNT 间的比较对 OC1REF 不起作用
2. 匹配有效：当计数值 CNT 与实际比较值(CCRx_SHAD)匹配时，OCxREF 为高电平；
3. 匹配无效：当计数值 CNT 与实际比较值(CCRx_SHAD)匹配时，OCxREF 为低电平；
4. 翻转：当计数值 CNT 与实际比较值(CCRx_SHAD)匹配时，输出翻转；
5. 强制无效：OCxREF 强制为低电平；
6. 强制有效：OCxREF 强制为高电平；
7. PWM1：
   向上计数时，当 CNT<实际比较值(CCRx_SHAD)时，OCxREF 有效；
   向下计数时，CNT>实际比较值(CCRx_SHAD)时，OCxREF 无效；
8. PWM2：
   向上计数时，当 CNT<实际比较值(CCRx_SHAD)时，OCxREF 无效；
   向下计数时，CNT>实际比较值(CCRx_SHAD)时，OCxREF 有效；

配置为输出比较通道的示例步骤和示例代码如下：

1. 使能 TIM1 模块时钟并选择 TIM1 时钟源
2. 将通道相应的端口配置为输出端口
3. 配置输出波形的周期 (T1ARR) 和占空比 (T1CCRx)
4. 配置输出比较模式 (T1OCxM) 和输出极性 (T1CCxP)
5. 使能比较输出通道 (T1CCxE)
6. 打开主输出自动使能位 (T1AOE)，在更新事件发生时硬件会自动使能主输出 (T1MOE)
7. 使能计数器 (T1CEN)

```
BANKSEL     PCKEN               ;
BSR         PCKEN,0             ; 使能 TIM1 模块时钟

BANKSEL     TCKSRC          ;
LDWI        H'01'               ;
STR         TCKSRC          ; 选择 TIM1 时钟源为 HIRC

BANKSEL     TRISA               ;
LDWI        H'FE'               ;
STR         TRISA               ; 配置 PA0 为通道 1 的输出通道

BANKSEL     TIM1ARRL            ;
LDWI        H'05'               ;
STR       TIM1ARRL          ; 将输出波形周期配置为 6

LDWI        H'03'               ;
STR         TIM1CCR1L           ; 将输出波形占空比配置为 3

BANKSEL     TIM1CCMR1           ;
LDWI        H'10'               ;
STR         TIM1CCMR1           ; 配置通道 1 为匹配有效模式输出

LDWI        H'01'               ;
STR         TIM1CCER1           ; 使能通道 1

BANKSEL     TIM1BKR             ;
BSR         TIM1BKR,6           ; 打开主输出自动使能位 T1AOE

BANKSEL     TIM1CR1         ;
BSR         TIM1CR1,0           ; 开启计数器计数使能位
```

上述示例代码对应波形图：

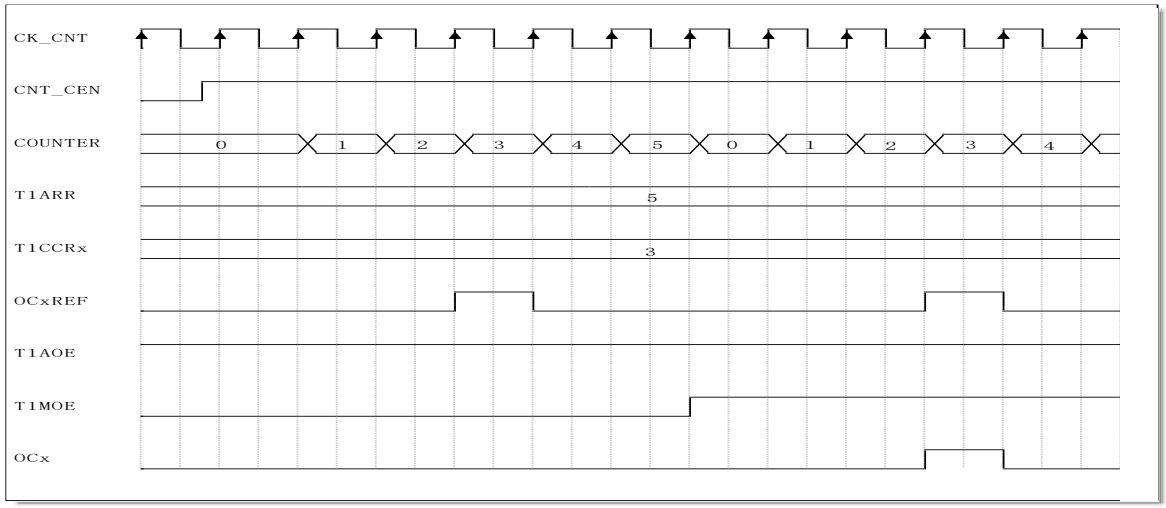


图 9-32 T1OCxM 为匹配有效模式下的输出时序图

示例代码：

```
BANKSEL     PCKEN               ;
BSR         PCKEN,0             ; 使能 TIM1 模块时钟

BANKSEL     TCKSRC          ;
LDWI        H'01'               ;
STR         TCKSRC          ; 选择 TIM1 时钟源为 HIRC

BANKSEL     TRISA               ;
LDWI        H'FE'               ;
STR         TRISA               ; 配置 PA0 为通道 1 的输出通道

BANKSEL     TIM1ARRL            ;
LDWI        H'05'               ;
STR         TIM1ARRL            ; 将输出波形周期配置为 6

LDWI        H'03'               ;
STR         TIM1CCR1L           ; 将输出波形占空比配置为 3

BANKSEL     TIM1CCMR1           ;
LDWI        H'30'               ;
STR         TIM1CCMR1           ; 配置通道 1 为翻转模式输出

LDWI        H'01'               ;
STR         TIM1CCER1           ; 使能通道 1

BANKSEL     TIM1BKR             ;
BSR         TIM1BKR,6           ; 打开主输出自动使能位 T1AOE

BANKSEL     TIM1CR1         ;
BSR         TIM1CR1,0           ; 开启计数器计数使能位
```

上述示例代码对应波形图：

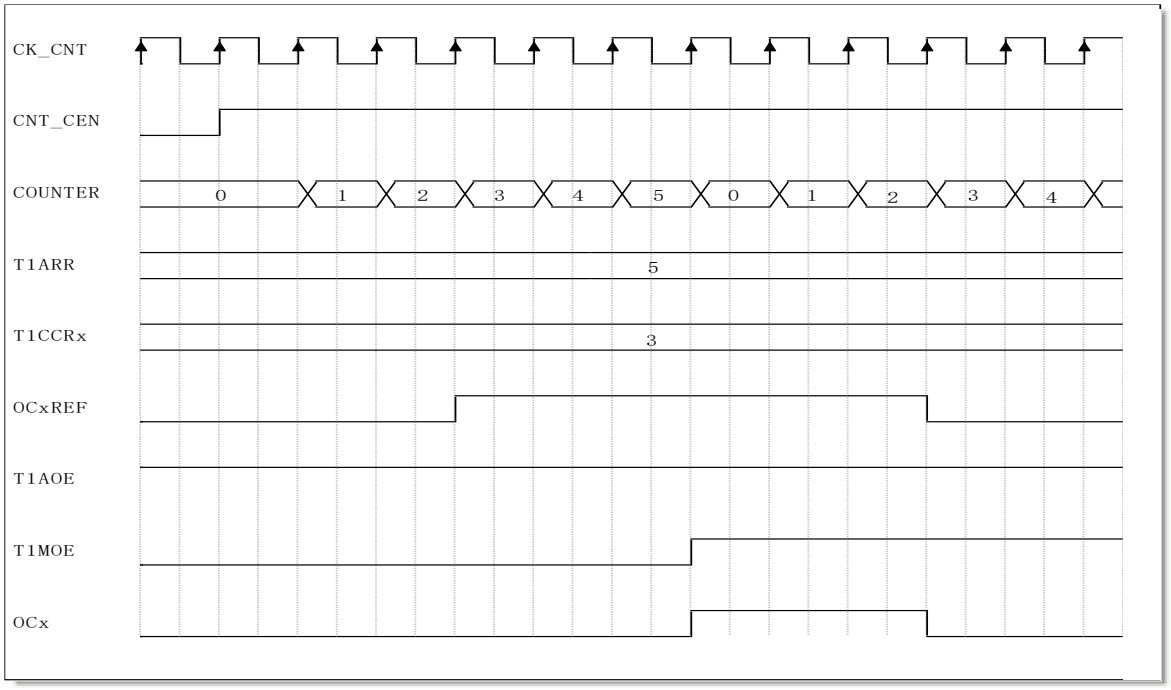


图 9-33 T1OCxM 为翻转模式下的输出时序图

示例代码：

```
BANKSEL     PCKEN           ;
BSR         PCKEN,0         ; 使能 TIM1 模块时钟

BANKSEL     TCKSRC          ;
LDWI        H'01'           ;
STR         TCKSRC          ; 选择 TIM1 时钟源为 HIRC

BANKSEL     TRISA           ;
LDWI        H'FE'           ;
STR         TRISA           ; 配置 PA0 为通道 1 的输出通道

BANKSEL     TIM1ARRL        ;
LDWI        H'05'           ;
STR         TIM1ARRL        ; 将输出波形周期配置为 6

LDWI        H'03'           ;
STR         TIM1CCR1L       ; 将输出波形占空比配置为 3

BANKSEL     TIM1CCMR1       ;
LDWI        H'70'           ;
STR         TIM1CCMR1       ; 配置通道 1 为 PWM2 模式输出

LDWI        H'01'           ;
STR         TIM1CCER1       ; 使能通道 1

BANKSEL     TIM1BKR         ;
BSR         TIM1BKR,6       ; 打开主输出自动使能位 T1AOE

BANKSEL     TIM1CR1         ;
BSR         TIM1CR1,0       ; 开启计数器计数使能位
```

上述示例代码对应波形图：

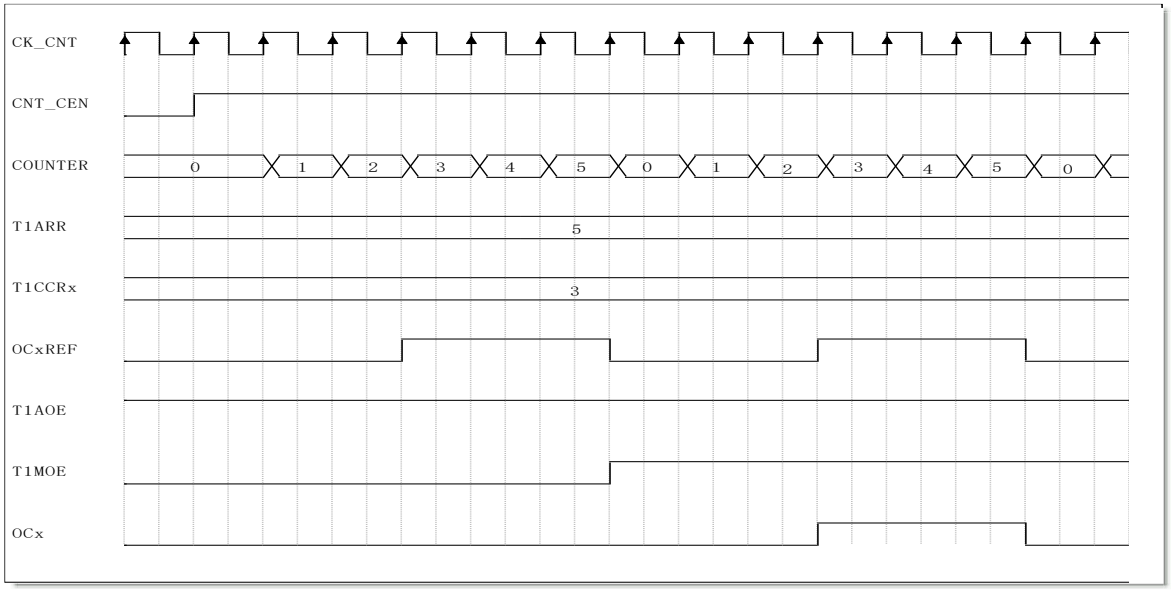


图 9-34 T1OCxM 为 PWM2 模式下的输出时序图

注意：

- 刹车事件的到来只会影响输出，并不影响 TIMER 本身的计数。
- 为了避免在计数器还未开启时，就开始输出脉冲；需要在计数器使能之后再开启 T1CCxE 和 T1CCxNE

9.3.3.3. 单次脉冲模式

开启单次脉冲模式和不开单次脉冲模式的区别在于：开启单次脉冲模式 (T1OPM=1) 并且下一次更新事件到来时，硬件会自动关闭计数器使能位 (T1CEN)，计数器停止计数。

想要产生一个正确的脉冲，比较值 (T1CCRx) 必须与计数器初始值 (T1ARR) 不同；所以在开始计数之前必须满足以下配置：

- 在向上计数模式下：COUNTER < T1CCRx < T1ARR
- 在向下计数模式下：COUNTER > T1CCRx

想要产生一个正确的脉冲，还必须满足以下配置：

- 在输出模式为 PWM1 模式 (T1OCxM=110) 下：T1CCxP 必须为 1
  如果输出模式为 PWM1 模式并且 T1CCxP 为 0，更新事件之后 PWM 输出会一直为有效值
- 在输出模式为 PWM2 模式 (T1OCxM=111) 下：T1CCxP 必须为 0
  如果输出模式为 PWM2 模式并且 T1CCxP 为 1，更新事件之后 PWM 输出会一直为有效值

9.3.3.3.1. 单次脉冲例程

单次脉冲模式可以配合触发模式在特定的时间点产生单个特定的 PWM 输出；配置步骤如下所示：

1. 使能 TIM1 模块时钟并选择 TIM1 时钟源
2. 将通道 2 相应的端口配置为输入端口，通道 1 相应的端口配置为输出端口
3. 通道 2 配置 T1CC2S 为 01，IC2 映射在 TI2FP2 上；并配置通道 2 为上升沿捕捉(T1CC2P=0)
4. 将计数控制模式配置为触发模式(T1SMS=110)，计数触发源配置为 TI2FP2(T1TS=110)
5. 通道 1 配置为输出通道(T1CC1S=00)
6. 通道 1 的比较输出模式配置为 PWM2 模式(T1OC1M=111)，输出极性配置为高电平有效(T1CC1P=0)
7. 使能计数器(T1CEN)
8. 开启通道 2 的输入捕捉功能 T1CC2E=1)和通道 1 的输出比较功能(T1CC1E)

以下是一段示例代码：

```
BANKSEL     PCKEN           ;
BSR         PCKEN,0         ; 使能 TIM1 模块时钟

BANKSEL     TCKSRC          ;
LDWI        H'01'           ;
STR         TCKSRC          ; 选择 TIM1 时钟源为 HIRC

BANKSEL     TRISA           ;
LDWI        H'FE'           ;
STR         TRISA           ; 配置 PA0 为通道 1 的输出通道，PA1 为通道 2 的输入通道

BANKSEL     TIM1CCMR1       ;
LDWI        H'70'           ;
STR         TIM1CCMR1       ; 配置通道 1 为 PWM2 模式输出

LDWI        H'01'           ;
STR         TIM1CCMR2       ; 配置通道 2 的 IC2 映射在 TI2FP2 上

LDWI        H'66'           ;
```

```
STR     TIM1SMCR        ; 配置 TIM1 为触发控制模式，触发源为 TI2FP2
LDWI    H'11'           ;
STR     TIM1CCER1       ; 使能通道 1 和通道 2
BANKSEL TIM1BKR         ;
BSR     TIM1BKR,7       ; 打开主输出使能 T1MOE
BANKSEL TIM1CR1         ;
BSR     TIM1CR1,0       ; 开启计数器计数使能位
```

上述示例代码对应示意图：

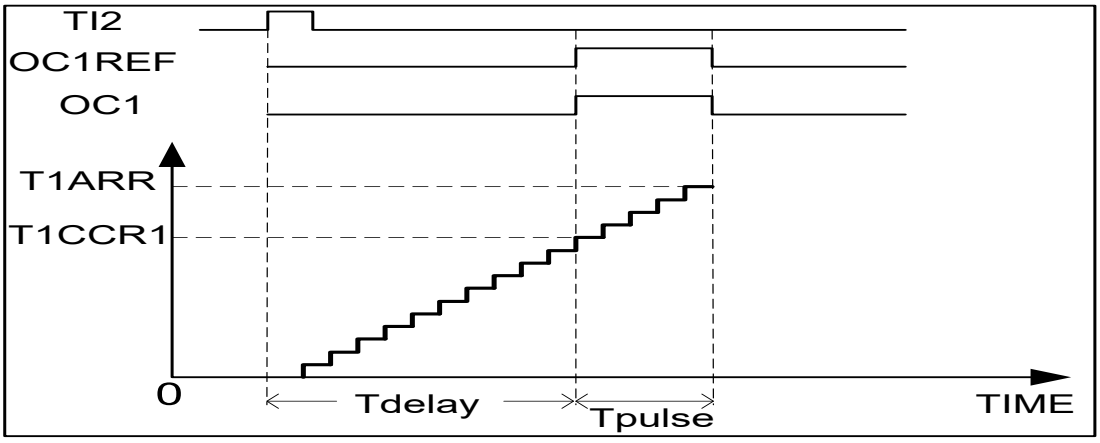


图 9-35 单次脉冲应用示意图

9.3.3.3.2. 快速输出使能

在单脉冲模式下，Tix 输入的沿检测会使能计数器计数；然后，计数器值与 T1CCRxH/L 值的比较会使输出发生变化；但是，从 Tix 输入边沿发生变化到真正输出波形中间会有一段最小延时。

为了缩短输出波形的最小延时，可以设置 TIM1CCMRx 寄存器中的 T1OcxFE 位。此位只能在输出配置为 PWM1 模式或 PWM2 模式时开启。

9.3.3.4. 死区产生

当把通道的互补输出使能时，就自动使能死区功能。每当一个输出信号 (正向输出信号或互补输出信号) 出现下降沿时，就会将另一个信号的上升沿后延一个死区时间长度。如**图 9-36** 和**图 9-37** 所示：

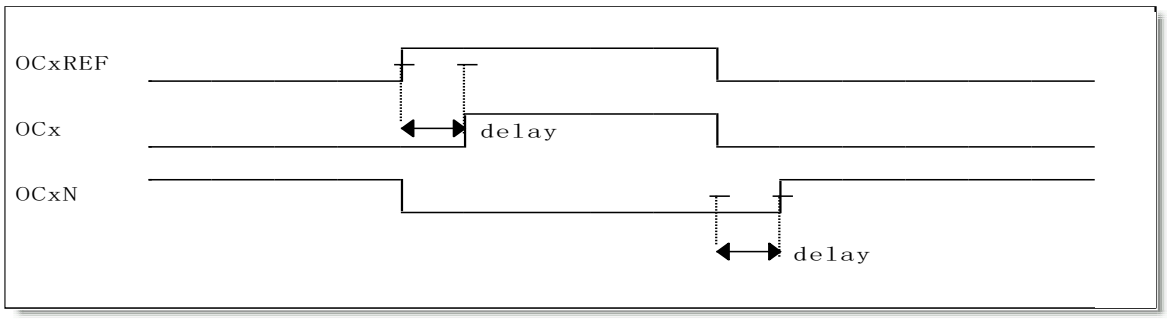


图 9-36 正向输出插入死区时序图

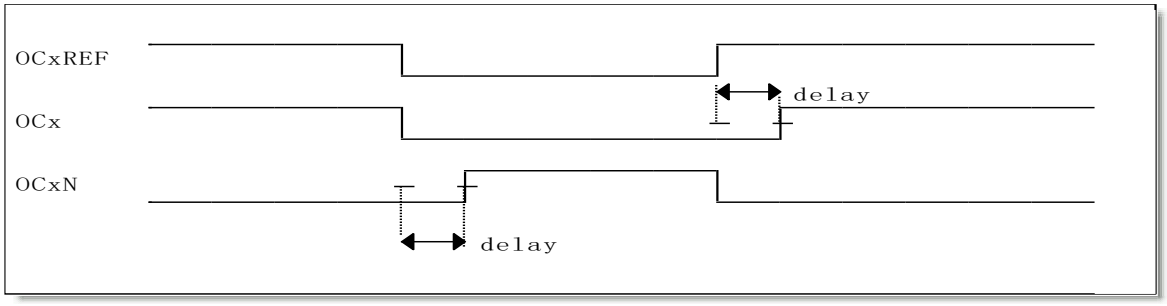


图 9-37 互补输出插入死区时序图

死区时间可以编程:根据寄存器位 T1DTG[7:0]，可以配置死区时间长度，具体参考寄存器描述 TIM1DTR 的 T1DTG[7:0]。

有的 OCXREF 输出的脉冲时间很短 (小于死区时间)，有可能某一脉冲信号 (正向输出信号或反向输出信号) 会被死区覆盖，导致输出不变化。 如图 9-33 和图 9-35 所示：

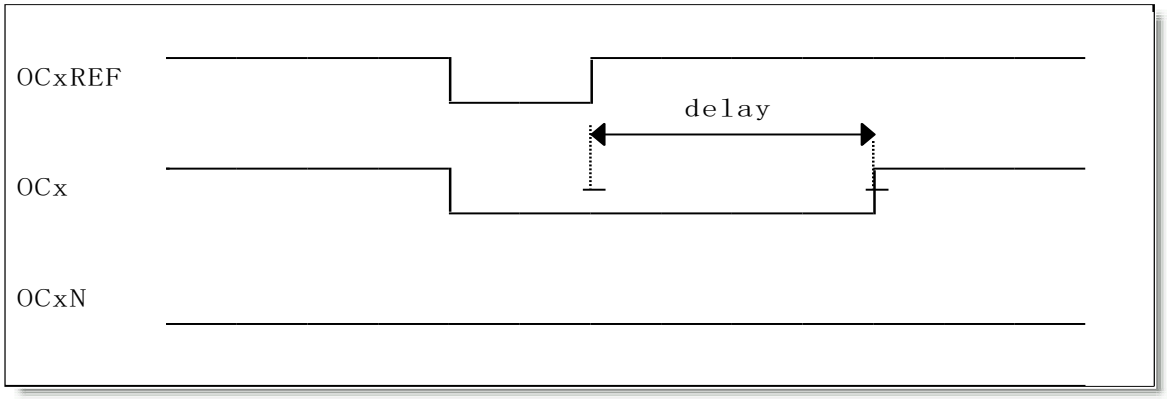


图 9-38 正向输出被死区覆盖时序图

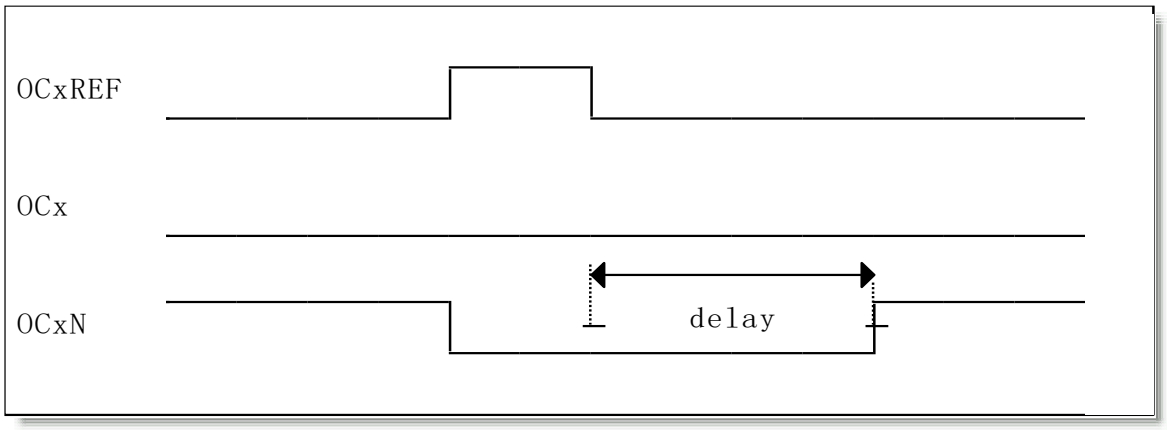


图 9-39 互补输出被死区覆盖时序图

9.3.3.5. 输出控制

参考信号 OCxREF 产生后不直接输出到端口，而是先经过死区控制模块和极性选择，再由以下 5 位控制信号 (T1MOE、T1OSSI、T1OSSR、T1CCxNE 和 T1CCxE) 进行组合控制之后送到端口上。具体的组合控制内容请见**表 9-11** Timer1 输出控制和状态”

输出的状态转换实际是刹车事件异步将 T1MOE 清零实现的。T1MOE 异步清零，但要注意刹车事件撤销时需要同步 2 个 CK_CNT 时钟 (如果有时钟的情况下)。

9.3.4. 同步模式

通过 TIM1CCER2 寄存器中的 T1SMOD 位可配置 TIM1 为同步模式。在同步模式下，通道 1/2/3 的互补输出与其正向输出信号完全相同。在该模式下，硬件自动禁止死区功能。

9.3.5. 群组模式

通过 TIM1CCER2 寄存器中的 T1GP 位可配置 TIM1 为群组模式。在群组模式下，通道 1/2/3 及其互补通道的占空比都由 TIM1CCR1H 和 TIM1CCR1L 决定，原本的配置失效(即通道 2/3 的输出与通道 1 的输出完全一致，通道 2/3 的互补输出与通道 1 的互补输出完全一致)。

9.3.6. TIM1/TIM2 的同步触发模式

通过 TIM1CR2 寄存器中的 T1ENCTRL 位可配置成 TIM1/TIM2 的同步触发模式。在该模式下，TIM2 的内部使能信号完全由 TIM1 的 T1CEN 控制。配置过程中，可先将 TIM1 和 TIM2 配置完，再配置 T1ENCTRL 为 1，最后通过配置 T1CEN 将 TIM1 和 TIM2 同时开启。

9.3.7. TIM1 中断

TIM1 有以下 8 个中断请求源：

- 刹车中断
- 触发中断
- 换相中断
- 捕捉/比较 4 中断
- 捕捉/比较 3 中断
- 捕捉/比较 2 中断
- 捕捉/比较 1 中断
- 更新中断(例如：上溢、下溢、计数初始化)

在用这些中断之前需要提前打开 TIM1IER 寄存器中的中断使能位(T1BIE、T1TIE、T1COMIE、T1CCxIE 和 T1UIE)。

不同的中断源还可以配置通过 TIM1EGR 寄存器来产生(软件产生中断)。

9.3.8. 故障刹车源

TIM1 有以下 3 种刹车事件：

- BKIN 管脚事件
- LVD 事件
- ADC 比较事件

当故障事件有效且被选择为刹车源 (由 BKS0~2 决定)，如果 BKE 位为 1，PWM 输出管脚将被置于预设的状态，预设状态由寄存器 TIM1OISR 决定。

当一个刹车事件发生时：

- T1MOE 位会被异步清 0，强制输出进入无效状态，空闲状态或复位状态。甚至在 MCU 振荡器关闭的情况下，T1MOE 也会被刹车事件清 0。
- 在 T1MOE=0 之后，每个输出通道都会先将输出值置为无效值，等死区时间到之后变成提前设置好的 T1OISx 位的值。如果 T1OSSI=0，timer 会将输出关闭。
- 当互补输出使能时：

  输出首先会设置为无效值(根据极性选择位)。此操作是异步清 0 的，所以即使 TIM1 没有时钟驱动也能进行。

  如果 TIM1 是有时钟进行驱动的，那么死区时间到来之后就会进入由 T1OISx 和 T1OISxN 提前设定的预设状态。(由于 T1MOE 的同步，所以此情况下真正的死区时间会比死区设置值长 2 个 CK_CNT 时钟)
- 刹车状态标志位(T1BIF)被置位。如果 T1BIE 位为 1，那么将会产生一个中断事件。
- 如果 T1AOE 位配置为 1，那么 T1MOE 位在下次更新事件(UEV)到来时，将会由硬件自动置位。如果 T1AOE 位为 0，那么只能由软件将 T1MOE 位重新置位。

当故障事件有效时，T1MOE 清 0，PWM 输出将一直置于预设状态；

故障事件撤消后，如果 T1AOE=1，PWM 将在下一次更新事件后恢复正常输出，否则，软件需要自动打开 T1MOE。

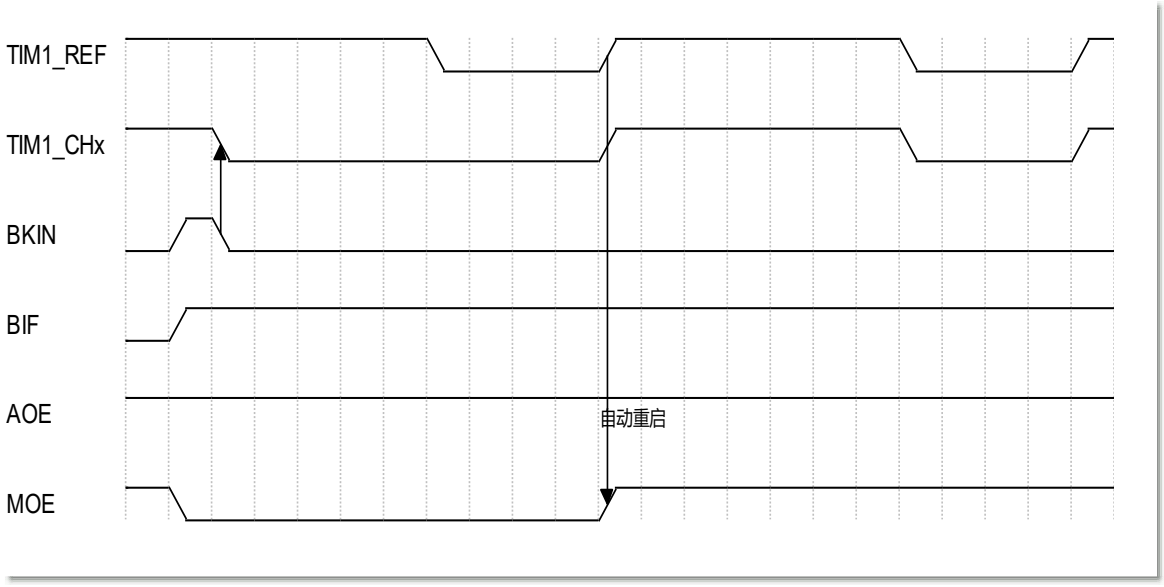


图 9-40 PWM 的自动重启

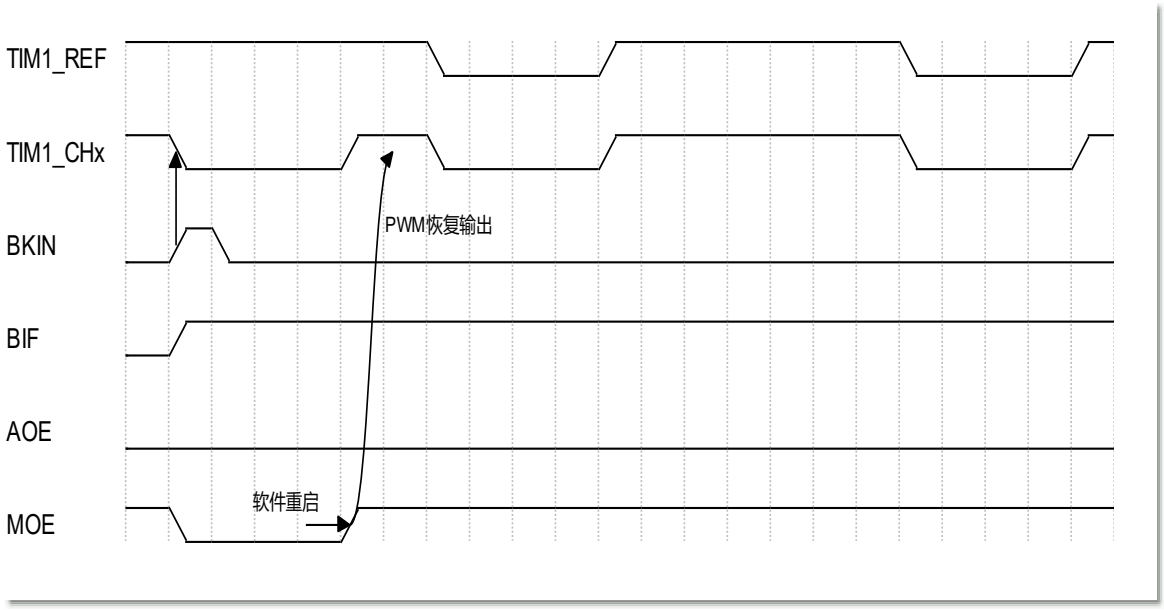


图 9-41 PWM 的软件重启

9.3.9. 前沿消隐

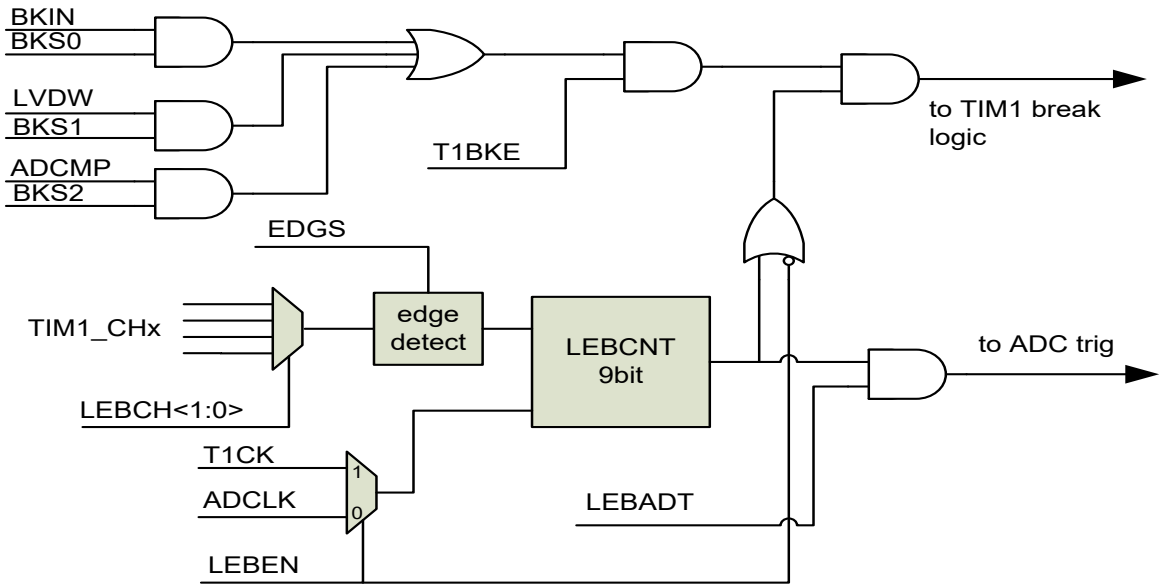


图 9-42 前沿消隐原理框图

在高速开关应用中，开关通常会产生极大的瞬变，这些瞬变可能会导致测量误差。利用前沿消隐 (LEB) 功能，应用程序可以忽略 PWM 输出边沿附近发生的瞬变。

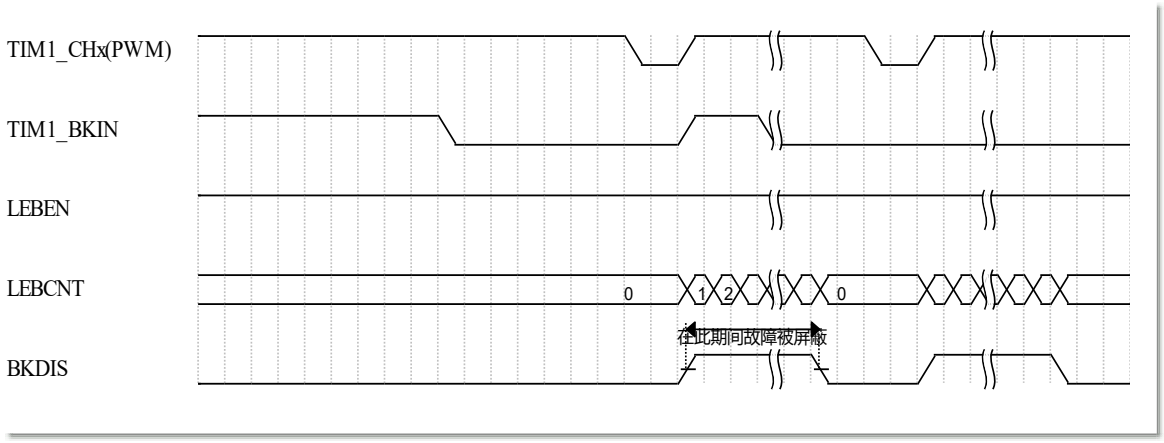


图 9-43 前沿消隐时序示意图

LEBCH 用于选择被消隐 TIM1 的 PWM 通道，EDGS 选择边沿类型。当 LEBEN 为 1，PWM 边沿将触发 LEB 定时器计数，时钟源为 TIM1 时钟，直到计数值等于 LEBPR，LEB 定时器停止计数，这段时间为消隐周期，期间所发生的刹车事件将被忽略；在消隐周期内如果再次发生有效的 PWM 边沿，则 LEB 定时器将清 0，重新开始计数。

注意：

(1) LEB 定时器和 ADC 延时定时器复用了同一个 9bit 计数器，当 LEBEN 为 1 时，原 ADC 的延时触发功能被禁止，但如果 LEBADT 为 1，LEB 定时器溢出将触发一次 AD 转换。

(2) 寄存器 ADCON3 中 ADCMPEN 位的关闭能将 ADCMP 产生的刹车事件清除

10. 通用定时器 TIM2

10.1. 特性

Timer2 的功能除捕捉比较通道数量不同以外，其他相同：

1. 16bit 的向上计数，支持自动重载；
2. 计数时钟预分频；
3. 支持 1/2 个独立的捕捉比较通道，通道可支持：
   - 输入捕捉
   - 输出比较
   - PWM 产生
4. 中断事件：
   - 更新事件：计数器溢出，计数器初始化
   - 输入捕捉事件
   - 输出比较事件

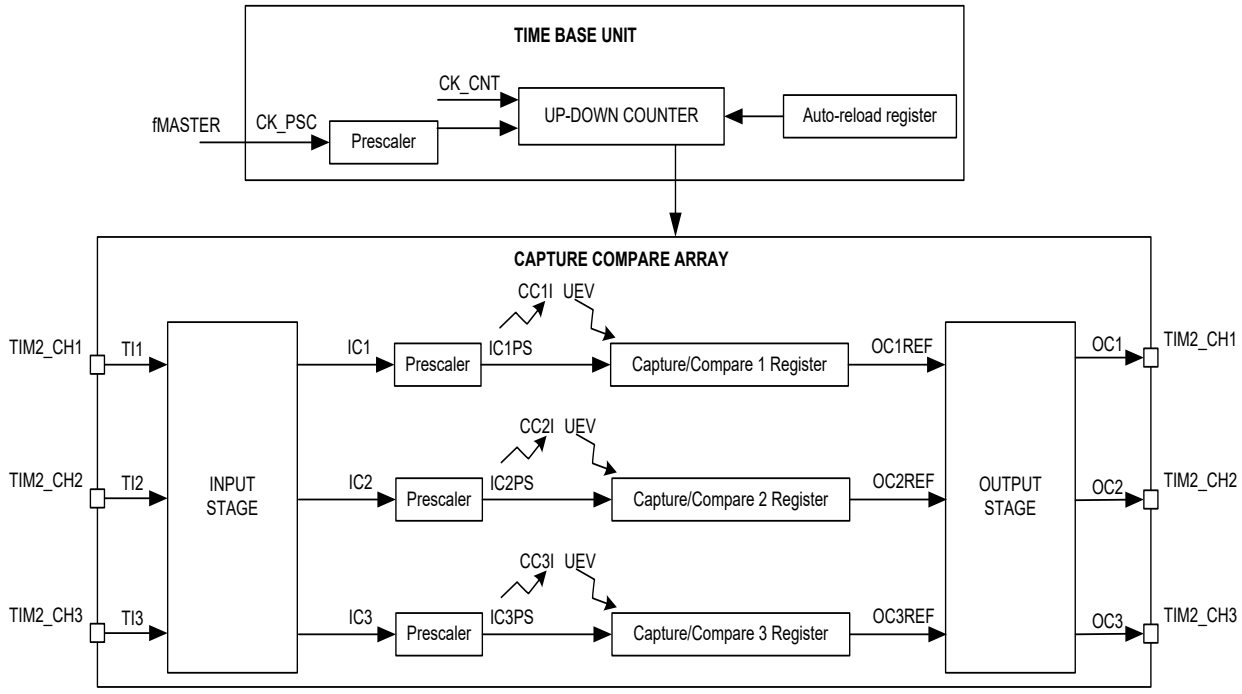


图 10-1 TIM2 原理框图

10.2. Timer2 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
PCKEN	0x9A	—	—	UARTEN	—	TIM4EN	TIM2EN	TIM1EN	ADCEN	0000 0000
CKOCON	0x95	SYSON	CCORDY	DTYSEL		CCOSEL[2:0]			CCOEN	0010 0000
TIM2CR1	0x30C	T2ARPE	—	—	—	T2OPM	T2URS	T2UDIS	T2CEN	0--- 000
TIM2IER	0x30D	—	—	—	—	T2CC3IE	T2CC2IE	T2CC1IE	T2UIE	---- 0000
TIM2SR1	0x30E	—	—	—	—	T2CC3IF	T2CC2IF	T2CC1IF	T2UIF	---- 0000
TIM2SR2	0x30F	—	—	—	—	T2CC3OF	T2CC2OF	T2CC1OF	—	---- 000-
TIM2EGR	0x310	—	—	—	—	T2CC3G	T2CC2G	T2CC1G	T2UG	---- 0000
TIM2CCMR1 (output mode)	0x311	—	T2OC1M[2:0]			T2OC1PE	—	T2CC1S[1:0]		-000 0-00
TIM2CCMR1 (input mode)		T2IC1F[3:0]				T2IC1PSC[1:0]		T2CC1S[1:0]		0000 0000
TIM2CCMR2 (output mode)	0x312	—	T2OC2M[2:0]			T2OC2PE	—	T2CC2S[1:0]		-000 0-00
TIM2CCMR2 (input mode)		T2IC2F[3:0]				T2IC2PSC[1:0]		T2CC2S[1:0]		0000 0000
TIM2CCMR3 (output mode)	0x313	—	T2OC3M[2:0]			OC3PE	—	T2CC3S[1:0]		-000 0-00
TIM2_CCMR3 (input mode)		T2IC3F[3:0]				T2IC3PSC[1:0]		T2CC3S[1:0]		0000 0000
TIM2CCER1	0x314	—	—	T2CC2P	T2CC2E	—	—	T2CC1P	T2CC1E	--00 --00
TIM2CCER2	0x315	—	—	—	—	—	—	T2CC3P	T2CC3E	---- --00
TIM2CNTRH	0x316	T2CNT[15:8]								0000 0000
TIM2CNTRL	0x317	T2CNT[7:0]								0000 0000
TIM2PSCR	0x318	—	—	—	—	T2PSC[3:0]				---- 0000
TIM2ARRH	0x319	T2ARR[15:8]								1111 1111
TIM2ARRL	0x31A	T2ARR[7:0]								1111 1111
TIM2CCR1H	0x31B	T2CCR1[15:8]								0000 0000
TIM2CCR1L	0x31C	T2CCR1[7:0]								0000 0000
TIM2CCR2H	0x31D	T2CCR2[15:8]								0000 0000
TIM2CCR2L	0x31E	T2CCR2[7:0]								0000 0000
TIM2CCR3H	0x29E	T2CCR3[15:8]								0000 0000
TIM2CCR3L	0x29F	T2CCR3[7:0]								0000 0000

表 10-1 Timer2 相关用户寄存器汇总

名称	状态		寄存器	地址	复位值
T2CNT	TIM2 计数值	高 8 位	TIM2CNTRH[7:0]	0x316	RW-0000 0000
		低 8 位	TIM2CNTRL[7:0]	0x317	RW-0000 0000
T2PSC	TIM2 预分频器		TIM2PSCR[3:0]	0x318	RW-0000
T2ARR	输出比较模式：PWM 周期的自动重装载寄存器(预装载值) 注：此值为 0 时，计数器不工作；	高 8 位	TIM2ARRH[7:0]	0x319	RW-1111 1111
		低 8 位	TIM2ARRL[7:0]	0x31A	RW-1111 1111
T2CCR1	输出比较模式: TIM2_CH1 占空比(预装载值)	高 8 位	TIM2CCR1H[7:0]	0x31B	RW-0000 0000
		低 8 位	TIM2CCR1L[7:0]	0x31C	RW-0000 0000
	输入捕获模式: 上一次捕获事件(IC1)捕获的计数值	高 8 位	TIM2CCR1H[7:0]	0x31B	RO-0000 0000
		低 8 位	TIM2CCR1L[7:0]	0x31C	RO-0000 0000
T2CCR2	输出比较模式: TIM2_CH2 占空比(预装载值)	高 8 位	TIM2CCR2H[7:0]	0x31D	RW-0000 0000
		低 8 位	TIM2CCR2L[7:0]	0x31E	RW-0000 0000
	输入捕获模式: 上一次捕获事件(IC2)捕获的计数值	高 8 位	TIM2CCR2H[7:0]	0x31D	RO-0000 0000
		低 8 位	TIM2CCR2L[7:0]	0x31E	RO-0000 0000
T2CCR3	输出比较模式: TIM2_CH3 占空比(预装载值)	高 8 位	TIM2CCR3H[7:0]	0x29E	RW-0000 0000
		低 8 位	TIM2CCR3L[7:0]	0x29F	RW-0000 0000
	输入捕获模式: 上一次捕获事件(IC3)捕获的计数值	高 8 位	TIM2CCR3H[7:0]	0x29E	RO-0000 0000
		低 8 位	TIM2CCR3L[7:0]	0x29F	RO-0000 0000

表 10-2 Timer2 周期相关寄存器

名称	状态		寄存器	地址	复位值
TIM2EN	TIM2 模块时钟	1 = 使能 0 = 关闭	PCKEN[2]	0x9A	RW-0
SYSON	睡眠模式下，系统时钟控制	1 = 使能 0 = 关闭	CKOCON[7]	0x95	RW-0
T2CKSRC	Timer2 时钟源 (Fmaster) 000 = Sysclk　100 = 2 x (XT or EC) (*) 001 = HIRC　101 = LIRC 010 = XT or EC (*)　110 = LP or EC (*) 011 = 2 x HIRC　111 = 2x (LP or EC) (*) (*) FOSC 应相应配置成 LP/XT/EC 模式或选择 INTOSCIO 模式，否则振荡器将不会运行。		TCKSRC[6:4]	0x31F	RW-000
DTYSEL	TIM1/TIM2 倍频时钟占空比调节位 00 = 2ns 延迟　10 = 4ns 延迟 01 = 3ns 延迟　11 = 7ns 延迟		CKOCON[5:4]	0x95	RW-10
T2ARPE	PWM 周期的自动预装载 1 = 使能 (T2ARR 预装载值在更新事件到来时被加载) 0 = 禁止 (T2ARR 立即被加载)		TIM2CR1[7]	0x30C	RW-0
T2OPM	单脉冲模式 1 = 使能 (下一次更新事件到来时，T2CEN 自动清零，计数器停止) 0 = 关闭 (发生更新事件时，计数器不停止)		TIM2CR1[3]		RW-0
T2URS	当 T2UDIS=0 时，更新事件中断源 1 / 0 = 计数器上溢/下溢		TIM2CR1[2]		RW-0
T2UDIS	产生更新事件控制 1 = 禁止 0 = 允许		TIM2CR1[1]		RW-0
T2CEN	TIM2 计数器	1 = 使能 0 = 关闭	TIM2CR1[0]		RW-0

表 10-3　Timer2 相关用户控制寄存器

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	Bit0	复位值
TIM2CCMR1	0x311	T2IC1F[3:0]				T2IC1PSC[1:0]		T2CC1S[1:0]		RO-0000 0000
TIM2CCMR2	0x312	T2IC2F[3:0]				T2IC2PSC[1:0]		T2CC2S[1:0]		RO-0000 0000
TIM2CCMR3	0x313	T2IC3F[3:0]				T2IC3PSC[1:0]		T2CC3S[1:0]		RO-0000 0000

名称	状态			寄存器	地址	复位值
T2ICxF	通道 x 输入捕获采样频率和数字滤波器长度			TIM2CCMRx[7:4] x = 1, 2, 3	0x311/ 0x312/ 0x313	R0−0000
	Value	采样频率 ($f_{SAMPLING}$)	数字滤波器长度 (N)			
	0000	Fmaster /2	0			
	0001	Fmaster	2			
	0010	Fmaster	4			
	0011	Fmaster	8			
	0100	Fmaster / 2	6			
	0101	Fmaster / 2	8			
	0110	Fmaster / 4	6			
	0111	Fmaster / 4	8			
	1000	Fmaster / 8	6			
	1001	Fmaster / 8	8			
	1010	Fmaster / 16	5			
	1011	Fmaster / 16	6			
	1100	Fmaster / 16	8			
	1101	Fmaster / 32	5			
	1110	Fmaster / 32	6			
	1111	Fmaster / 32	8			
T2ICxPSC	通道 x 输入捕获预分频器 (几个事件触发一次捕获) 00 = 1 个 01 = 2 个 10 = 4 个 11 = 8 个 注: 当 T2CCxE = 0 时，该预分频器复位为 00			TIM2CCMRx[3:2]		RO−00
T2CC1S [1]	通道 1 模式选择	00 = 输出 01 = 输入，输入脚映射在 TI1FP1 10 = 输入，输入脚映射在 TI2FP1 11 = 保留		TIM2CCMR1[1:0]	0x311	RO−00

[1] 仅在通道 x 关闭时(即 T2CCxE = 0)可写，x = 1, 2, 3。

名称	状态		寄存器	地址	复位值
T2CC2S [2]	通道2模式选择	00 = 输出 01 = 输入，输入脚映射在 TI2FP2 10 = 输入，输入脚映射在 TI1FP2 11 = 保留	TIM2CCMR2[1:0]	0x312	RO-00
T2CC3S [2]	通道3模式选择	00 = 输出 01 = 输入，输入脚映射在 TI3FP3 1x = 保留	TIM2CCMR3[1:0]	0x313	RO-00

表 10-4 TIM2CCMRx 作为输入配置寄存器

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	Bit0	复位值
TIM2CCMR1	0x311	-	T2OC1M[2:0]			T2OC1PE	-	T2CC1S[1:0]		RW-000 0-00
TIM2CCMR2	0x312	-	T2OC2M[2:0]			T2OC2PE	-	T2CC2S[1:0]		RW-000 0-00
TIM2CCMR3	0x313	-	T2OC3M[2:0]			T2OC3PE	-	T2CC3S[1:0]		RW-000 0-00

T2OCxM	输出模式描述		OCxREF (输出参考信号)
000	冻结 (不比较)		禁止
001	当 TIM2_CNT = CCRx_SHAD 时		1
010	当 TIM2_CNT = CCRx_SHAD 时		0
011	当 TIM2_CNT = CCRx_SHAD 时		电平翻转
100	强制为无效电平		0
101	强制为有效电平		1
110	PWM1 模式	TIM2_CNT < CCRx_SHAD	1
		TIM2_CNT > CCRx_SHAD	0
111	PWM2 模式	TIM2_CNT < CCRx_SHAD	0
		TIM2_CNT > CCRx_SHAD	1

1. OCxREF 与 T2CCxP 共同决定输出引脚 OCx 的值；
2. PWM 模式下比较结果改变时，或输出比较模式下从冻结模式切换到 PWM 模式时，OCxREF 电平才会改变；

表 10-5 T2OCxM 配置为输出比较模式

[2] 仅在通道 x 关闭时(即 T2CCxE = 0)可写，x = 1, 2, 3。

名称	状态		寄存器	地址	复位值
T2OCxPE	通道 x 输出比较占空比的自动预装载 1 = 使能 (T2CCRx 预装载值在更新事件到来时被加载) 0 = 禁止 (T2CCRx 立即被加载)		TIM2CCMRx[3] x = 1, 2, 3	0x311/ 0x312/ 0x313	RW-0
T2CC1S [3]	通道 1 模式选择	00 = 输出 01 = 输入，输入脚映射在 TI1FP1 10 = 输入，输入脚映射在 TI2FP1 11 = 保留	TIM2CCMR1[1:0]	0x311	RW-00
T2CC2S [3]	通道 2 模式选择	00 = 输出 01 = 输入，输入脚映射在 TI2FP2 10 = 输入，输入脚映射在 TI1FP2 11 = 保留	TIM2CCMR2[1:0]	0x312	RW-00
T2CC3S [3]	通道 3 模式选择	00 = 输出 01 = 输入，输入脚映射在 TI3FP3 1x = 保留	TIM2CCMR3[1:0]	0x313	RW-00

表 10-6 TIM2CCMRx 作为输出配置寄存器

名称	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	地址	复位值
TIM2CCER1	-	-	T2CC2P	T2CC2E	-	-	T2CC1P	T2CC1E	0x314	RW-00 --00
TIM2CCER2	-	-	-	-	-	-	T1CC3P	T1CC3E	0x315	RW----- --00

名称	功能	输出比较模式	输入捕获/触发模式
T2CCxP	通道 x 引脚 输出极性选择	1 = OCx 低电平有效 0 = OCx 高电平有效	1 = 捕获 / 触发发生在 TIxF 低电平或下降沿 0 = 捕获 / 触发发生在 TIxF 高电平或上升沿
T2CCxE	通道 x 引脚 使能	1 = 使能 (OCx 输出到对应的引脚) 0 = 禁止	1 = 使能 (捕获计数器的值到 TIM2CCRx 寄存器中) 0 = 禁止

表 10-7 Timer2 通道输出和极性选择

名称	状态	寄存器	地址	复位值
TIM2_CH1	TIM2 通道 1 管脚功能重映射 1 = PB0 0 = PA5	AFP0[4]	0x19E	RW-0
TIM2_CH3	TIM2 通道 3 管脚功能重映射 1 = PA3 0 = PB5	AFP0[3]		RW-0

表 10-8 Timer2 通道管脚功能重映射

[3] 仅在通道 x 关闭时(即 T2CCxE = 0)可写，x = 1, 2, 3。

名称	状态		寄存器	地址	复位值
GIE	全局中断 1 = 使能 (PEIE, T2CCxIE, T2CCxG, T2UIE 适用) 0 = 全局关闭 (唤醒不受影响)		INTCON[7]	Bank 首地址 +0x0B	RW-0
PEIE	外设总中断 1 = 使能 (T2CCxIE, T2CCxG, T2UIE 适用) 0 = 关闭 (无唤醒)		INTCON[6]		RW-0
T2CC3IE	通道 3 捕获/比较中断	1 = 使能 0 = 关闭	TIM2IER[3]	0x30D	RW-0
T2CC2IE	通道 2 捕获或比较中断		TIM2IER[2]		RW-0
T2CC1IE	通道 1 捕获或比较中断		TIM2IER[1]		RW-0
T2CC3G [4]	通道 3 捕获/比较软件中断		TIM2EGR[3]	0x310	WO-0
T2CC2G [4]	通道 2 捕获/比较软件中断		TIM2EGR[2]		WO-0
T2CC1G [4]	通道 1 捕获/比较软件中断		TIM2EGR[1]		WO-0
T2CC3IF [5]	通道 x 匹配/捕获中断标志位 • 输出模式： 1 = CNT 值与 T2CCRx 值匹配 0 = 不匹配 • 输入模式： 1 = 计数器值已被捕获至 TIM2CCR 0 = 无捕获产生 注: 软件清 0 或读 TIM2CCRxL 清 0		TIM2SR1[3]	0x30E	R_W1C-0
T2CC2IF [5]			TIM2SR1[2]		R_W1C-0
T2CC1IF [5]			TIM2SR1[1]		R_W1C-0
T2CC3OF [5]	通道 x 重复捕获中断标志位 1 = 发生重复捕获 (计数器的值被捕获到 TIM2CCRx 寄存器时，T2CCxIF 的状态已经为 1) 0 = 无重复捕获 注：仅通道配置位捕获输入时有效		TIM2SR2[3]	0x30F	R_W1C-0
T2CC2OF [5]			TIM2SR2[2]		R_W1C-0
T2CC1OF [5]			TIM2SR2[1]		R_W1C-0
T2UIE	允许更新中断	1 = 使能 0 = 关闭	TIM2IER[0]	0x30D	RW-0
T2UG [4]	允许更新软件中断		TIM2EGR[0]	0x310	WO-0
T2UIF [5]	更新中断标志位 1 = 更新事件等待响应 0 = 无更新事件		TIM2SR1[0]	0x30E	R_W1C-0

表 10-9 Timer2 中断使能和状态位

[4] 软件置 1，硬件自动清 0。

[5] 写 1 清 0，写 0 无效。建议只使用 STR、MOVWI 指令进行写操作，而不要用 BSR 或 IOR 指令。

10.3. 功能描述

整个 TIM2 可以分为两个大的功能部分：计数基本单元和捕捉比较通道。计数基本单元分为向上计数器、自动加载寄存器、预分频器；捕捉比较通道分为捕捉输入通道，输出比较通道和输出控制。

10.3.1. 计数基本单元

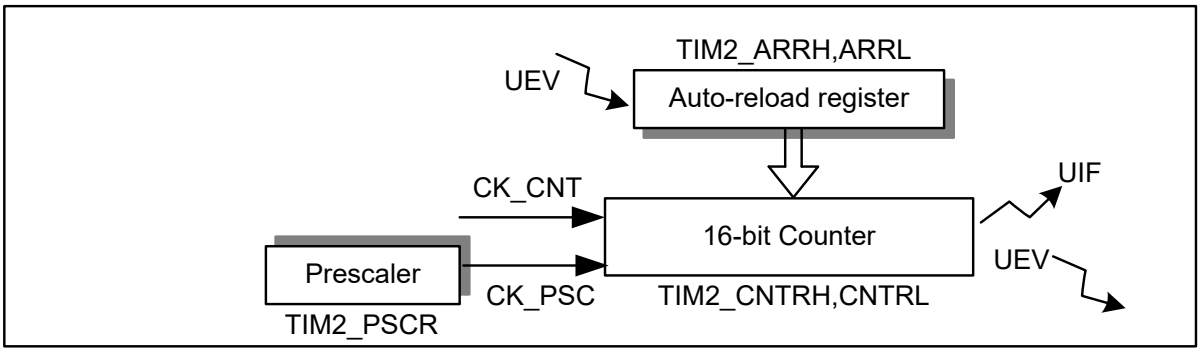


图 10-2 计数基本单元框图

计数基本单元包括：

- 16 位向上计数器
- 16 位自动重加载寄存器
- 4 位可编程预分频器

TIM2 没有重复计数器

10.3.1.1. 时钟源选择

时钟源可由 TCKSRC 寄存器进行配置：

- T2CKSRC[2:0] = 000 时，系统时钟/主时钟为 TIM2 时钟
- T2CKSRC[2:0] = 001 时，HIRC 为 TIM2 时钟
- T2CKSRC[2:0] = 010 时，XT 时钟/外部时钟为 TIM2 时钟
- T2CKSRC[2:0] = 011 时，HIRC 的 2 倍频为 TIM2 时钟
- T2CKSRC[2:0] = 100 时，XT 时钟/外部时钟的 2 倍频为 TIM2 时钟
- T2CKSRC[2:0] = 101 时，LIRC 为 TIM2 时钟
- T2CKSRC[2:0] = 110 时，LP 时钟/外部时钟为 TIM2 时钟
- T2CKSRC[2:0] = 111 时，LP 时钟/外部时钟的 2 倍频为 TIM2 时钟

10.3.1.2. 向上计数器

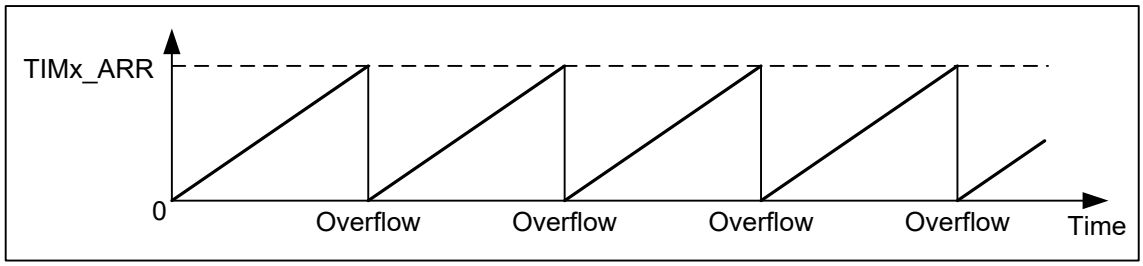


图 10-3 向上计数器

TIM2 计数器只能向上计数。计数器从 0 开始计数向上计数，计到 TIM1_ARR 寄存器所设数值。然后重新从 0 开始计数并产生一个计数器上溢事件；如果 T2UDIS 设为 0，那么还会产生一个更新事件 UEV。

10.3.1.3. 预分频器

计数时钟可以进行 4bit 的时钟预分频：

$$f_{CK_CNT} = Fmaster / 2^{(PSCR[3:0])_2}$$ (PSCR[3:0])

预分频支持分频自动更新，即在更新事件发生后，能够自动改变预分频值。当 T2CEN 为 0 时，写入预分频寄存器的值也能直接加载实际应用的预分频寄存器中。

10.3.2. 捕捉比较通道

TIM2CCMRx 寄存器是复用寄存器。

当作为输出比较通道时，TIM2CCMRx 寄存器作为输出配置寄存器，并且第 7 位和第 2 位禁止配置，保持为默认值；

当作为输入捕捉通道时，TIM2CCMRx 寄存器作为输入配置寄存器；

10.3.2.1. 捕捉输入通道

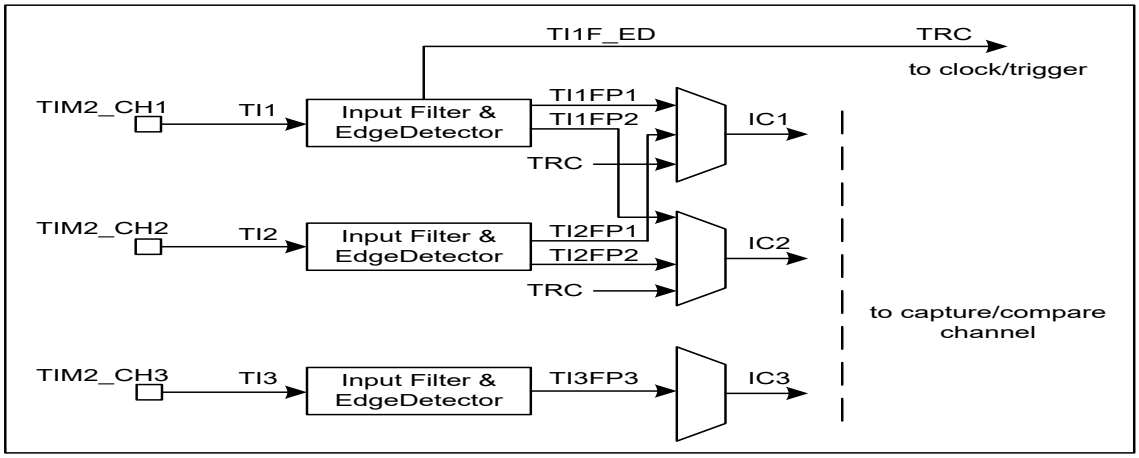


图 10-4 输入通道框图

10.3.2.2. 输出比较通道

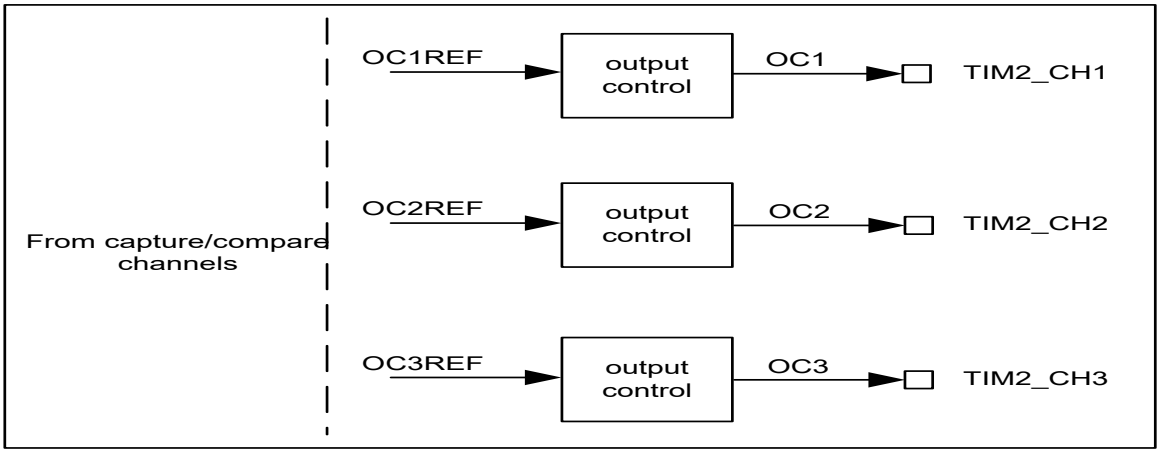


图 10-5 输出通道框图

TIM2 的输出没有死区功能，没有互补输出功能，也没有刹车功能。

10.3.3. TIM2 中断

TIM2 有以下 4 个中断请求源：

- 捕捉/比较 3 中断
- 捕捉/比较 2 中断
- 捕捉/比较 1 中断
- 更新中断

在用这些中断之前需要提前打开 TIM2IER 寄存器中的中断使能位 (T2CCxIE 和 T2UIE)。

不同的中断源还可以配置通过 TIM2EGR 寄存器来产生 (软件产生中断)。

11. 基本定时器 TIM4

11.1. 特性

- 8bit 自动重载向上计数器
- 计数时钟可编程预分频
- 计数器溢出中断

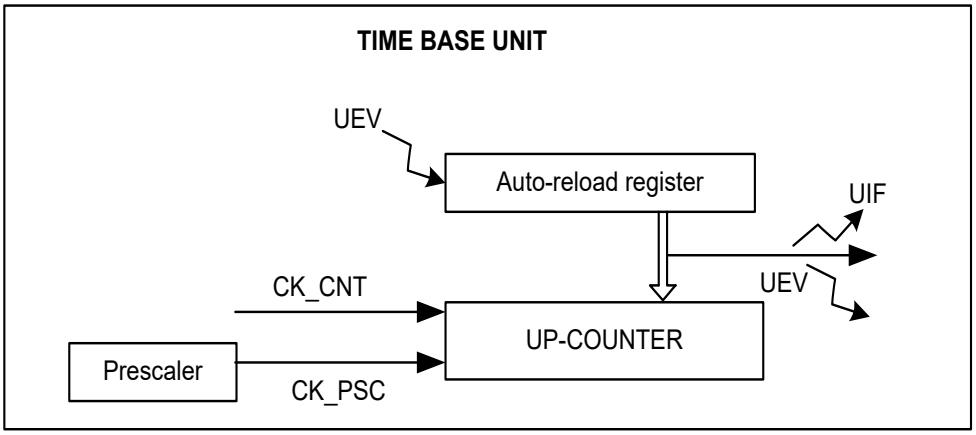


图 11-1 TIM4 原理框图

11.2. TIM4 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
PCKEN	0x9A	—	—	UARTEN	—	TIM4EN	TIM2EN	TIM1EN	ADCEN	0000 0000
CKOCON	0x95	SYSON	CCORDY	DTYSEL		CCOSEL[2:0]			CCOEN	0010 0000
TIM4CR1	0x111	T4ARPE	—	T4CKS[1:0]		T4OPM	T4URS	T4UDIS	T4CEN	0-00 0000
TIM4IER	0x112	—	—	—	—	—	—	—	T4UIE	---- ---0
TIM4SR	0x113	—	—	—	—	—	—	—	T4UIF	---- ---0
TIM4EGR	0x114	—	—	—	—	—	—	—	T4UG	---- ---0
TIM4CNTR	0x115	T4CNT[7:0]								0000 0000
TIM4PSCR	0x116	—	—	—	—	—	T4PSC[2:0]			---- -000
TIM4ARR	0x117	T4ARR[7:0]								1111 1111

表 11-1 Timer4 相关用户寄存器汇总

名称	状态		寄存器	地址	复位值
TIM4EN	TIM4 模块时钟	1 = 使能 0 = 关闭	PCKEN[3]	0x9A	RW-0
SYSON	睡眠模式下，系统时钟控制	1 = 使能 0 = 关闭	CKOCON[7]	0x95	RW-0
T4ARPE	周期的自动预装载 1 = 使能 (T4ARR 预装载值在更新事件到来时被加载) 0 = 禁止 (T4ARR 立即被加载)		TIM4CR1[7]	0x111	RW-0
T4CKS	Timer4 时钟源 00 = Sysclk　10 = LP (*) 01 = HIRC　11 = XT (*) (*)FOSC 应相应配置成 LP/XT 或选择 INTOSCIO 模式，否则振荡器将不会运行		TIM4CR1[5:4]		RW-00
T4OPM	单脉冲模式 1 = 使能 (下一次更新事件到来时，计数器停止) 0 = 关闭 (发生更新事件时，计数器不停止)		TIM4CR1[3]		RW-0
T4URS	当 T4UDIS = 0 时，更新事件中断源 1 = 计数器上溢 0 = 软件设置 T4UG 位或计数器上溢		TIM4CR1[2]		RW-0
T4UDIS	产生更新事件控制 1 = 禁止 0 = 允许		TIM4CR1[1]		RW-0
T4CEN	TIM4 计数器	1 = 使能 0 = 关闭	TIM4CR1[0]		RW-0
T4PSC	Timer4 预分频器 000 = 1　100 = 16 001 = 2　101 = 32 010 = 4　110 = 64 011 = 8　111 =128 注：必须产生更新事件或 T4CEN=0，更新的预分频值才生效		TIM4PSCR[2:0]	0x116	RW-000
T4CNT	Timer4 计数值		TIM4CNTR[7:0]	0x115	RW-0000 0000
T4ARR	周期的自动重装载寄存器(预装载值) 注：此值为 0 时，计数器不工作		TIM4ARR[7:0]	0x117	RW-1111 1111

表 11-2 Timer4 相关用户控制寄存器

名称	状态		寄存器	地址	复位值
GIE	全局中断 1 = 使能 (PEIE, T4UIE, T4UG 适用) 0 = 全局关闭 (唤醒不受影响)		INTCON[7]	Bank 首地址 +0x0B	RW-0
PEIE	外设总中断	1 = 使能 (T4UIE, T4UG 适用) 0 = 关闭 (无唤醒)	INTCON[6]		RW-0
T4UIE	允许更新中断	1 = 使能 0 = 关闭	TIM4IER[0]	0x112	RW-0
T4UG [1]	允许更新软件中断		TIM4EGR[0]	0x114	WO-0
T4UIF	更新中断标志位	1 = 更新事件等待响应 0 = 无更新事件	TIM4SR[0]	0x113	R_W1C-0

表 11-3 Timer4 中断使能和状态位

11.3. TIM4 时钟源

TIM4 有 4 种时钟源可选，由寄存器位 T4CKS 设置。在 TIM4 的被使能 (PCKEN.TIM4EN=1) 的情况下，所选择的时钟源被自动使能。

注意：

1. 如果要选择 LP 晶体时钟，系统时钟配置寄存器位 FOSC 必须选择 LP 模式，否则对应的时钟源将不被使能；
2. 同理，如果要选择 XT 晶体时钟，系统时钟配置寄存器位 FOSC 必须选择 XT 模式，否则对应的时钟源将不被使能；

SLEEP 模式下，如果 SYSON 为 1，且 TIM4EN=1，则所选择的时钟源将保持振荡，TIM4 将继续工作；否则，所选的时钟源取决于其他模块的设置情况。

11.4. 预分频器

计数时钟可以进行 3bit 的时钟预分频：

$$f_{CK_CNT} = Fmaster / 2^{(PSCR[2:0])}$$

预分频支持分频自动更新，即在更新事件发生后，能够自动改变预分频值。当 T4CEN 为 0 时，写入预分频寄存器的值也能直接加载实际应用的预分频寄存器中。

11.5. TIM4 中断

TIM4 只有一个中断请求源：

- 更新中断 (计数器上溢或计数器初始化)

在用这些中断之前需要提前打开 TIM4IER 寄存器中的中断使能位 (T4UIE)。

不同的中断源还可以配置通过 TIM4EGR 寄存器来产生 (软件产生中断 T4UG)

[1] 软件置 1，硬件自动清 0。

12. 睡眠模式

睡眠模式下，指令时钟关闭，指令执行停止，大多数模块掉电以降低功耗。如**表 12-1** 所示，FT61F14x 可根据实际需求在睡眠时有选择地开启各个模块，而无须指令介入，以使其相应功能如 LVR、LVD、WDT、Timers、PWM 和 ADC 能在 SLEEP 模式下保持运行。一些模块也可配置成进入 SLEEP 后自动关闭，而无须由指令关闭。

模块	SLEEP 模式下的各模块配置条件	
	运行	自动关闭?
指令时钟	(始终关闭)	Yes
LVR	LVREN = 00 or (LVREN = 01 & SLVREN=1)	LVREN = 10
LVD	LVDEN = 1	No
WDT	WDTE or SWDTEN	No
TIMER1	SYSON = 1 & TIM1EN = 1	SYSON =0
TIMER2	SYSON = 1 & TIM2EN = 1	SYSON =0
TIMER4	SYSON = 1 & TIM4EN = 1	SYSON = 0
PWM	(跟随 Timer1 或 Timer2)	
HIRC / LIRC / EC / LP / XT	(跟随使用它们的外设状态)	
ADC	(当 ADCEN = 1 & ADON = 1 且 ADC 所选时钟源保持运行时，ADC 即可运行)	
USART	(当 UARTEN = 1 且 USART 所选时钟源保持运行时，USART 即可运行)	
I/O	(除非 SLEEP 时使能 PWM，否则 I/O 将保持其进入 SLEEP 前的状态)	

表 12-1 除指令时钟外，其他模块可根据需求在 SLEEP 模式下保持运行

12.1. 进入 SLEEP

CPU 通过执行 SLEEP 指令进入睡眠模式。进入睡眠时：

1. 若 WDT 使能，则 WDT 的后分频器和定时器将被清零，并重新开始计时
2. 超时标志位 (/TO) = 1
3. 掉电标志位 (/PD) = 0
4. 32kHz LIRC 不受影响，并且由其提供时钟的外设可以在休眠模式下继续工作
5. LP 晶体振荡器不受影响 (当 TIMx 使用它作为工作时钟时)
6. I/O 端口保持执行 SLEEP 指令之前的状态 (驱动为高电平、低电平或高阻态)
7. WDT 之外的复位不受休眠模式影响

关于外设在休眠期间工作的更多详细信息，请参见各个章节。

要最大程度地降低电流消耗，应考虑以下条件：

1. I/O 引脚不应悬空，I/O 作为输入时可打开内部的上拉或下拉
2. 外部电路从 I/O 引脚灌电流
3. 内部电路从 I/O 引脚拉电流
4. 内部弱上拉的引脚
5. 模块使用 31kHz LIRC
6. 模块使用 LP 振荡器

12.2. 睡眠的唤醒

可以通过下列任一事件将器件从休眠状态唤醒：

1. MCLR 引脚上的外部复位输入 (如果使能)
2. BOR 复位 (如果使能)
3. POR 复位
4. 看门狗定时器 (如果使能) 溢出
5. 任何外部中断
6. 能够在休眠期间运行的外设产生的中断 (更多信息请参见各个外设)

前 3 个事件会使器件复位，后 3 个事件认为是程序执行的延续。

当执行 SLEEP 指令时，下一条指令 (PC+1)被预先取出。如果希望通过中断事件唤醒器件，则必须允许相应的中断允许位。唤醒与 GIE 位的状态无关，如果 GIE 位被禁止，器件将继续执行 SLEEP 指令后的指令。如果 GIE 位被允许，器件先执行 SLEEP 指令后的指令，然后将调用中断服务程序。如果不想执行 SLEEP 指令后的指令，用户应该在 SLEEP 指令后面放置一条 NOP 指令。

器件从休眠模式唤醒时，WDT 清零，与唤醒的原因无关。

12.2.1. 使用中断唤醒

当禁止全局中断 (GIE 被清零)，并且有任一中断源中断标志位置 1 且其中断被使能时，将会发生下列某一事件：

- 如果在执行 SLEEP 指令之前发生中断
  - ✓ SLEEP 指令将作为 NOP 执行
  - ✓ WDT 和 WDT 预分频器不会清零
  - ✓ STATUS 寄存器的 TO 位不会置 1
  - ✓ STATUS 寄存器的 PD 位不会清零
- 如果在执行 SLEEP 指令期间或之后发生中断
  - ✓ SLEEP 指令将完全执行
  - ✓ 器件将立即从休眠模式唤醒
  - ✓ WDT 和 WDT 预分频器将清零
  - ✓ STATUS 寄存器的 TO 位将置 1
  - ✓ STATUS 寄存器的 PD 位将清零

要确定是否执行了 SLEEP 指令，可以测试 PD 位。如果 PD 位置 1，则说明 SLEEP 指令被作为一条 NOP 指令执行了。

12.3. 睡眠的系统时钟

进入睡眠状态后，CPU 时钟停止，PC 停留在 SLEEP 的下一条地址。默认情况下，系统时钟也会被关闭。但如果 SYSON 位置 1 时，系统时钟将一直保持运行，在这种情况下，被选择为系统时钟的 HIRC，XT，LIRC 振荡器将不会被关闭。

注意：如果要使用 PROM 或数据 EEPROM 的写完成中断唤醒，SYSON 必须置 1。

13. 中断

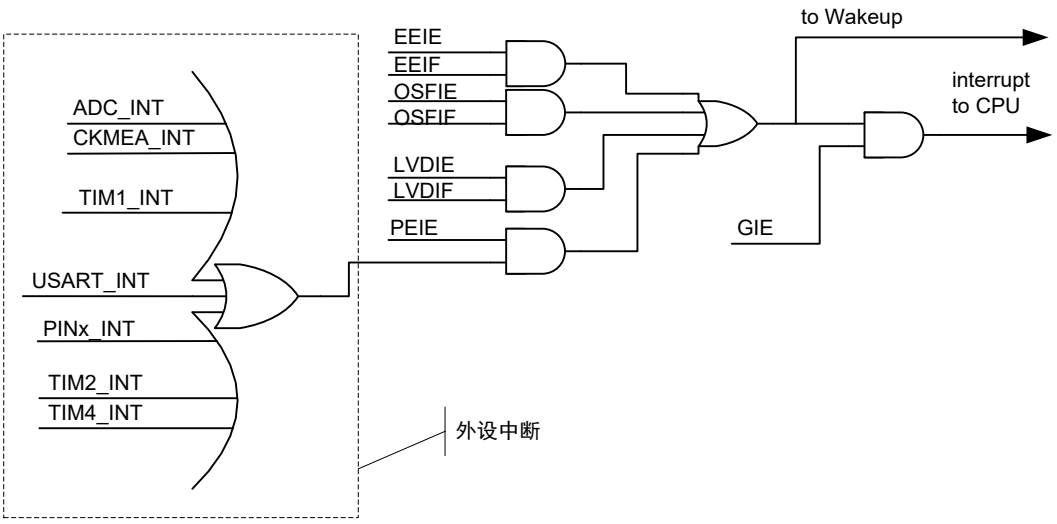


图 13-1 中断逻辑框图

FT61F14X 有以下中断源，部分中断可以把 CPU 从睡眠状态唤醒：

- 外部管脚中断
- ADC 中断
- LVD 中断
- EEPROM 写完成中断
- 慢时钟测量完成中断
- 时钟缺失中断
- TIMx 中断
- USART 中断

13.1. 中断相关寄存器汇总

名称	地址	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	复位值(RW)
INTCON	0x0B	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
PIE1	0x91	–	–	–	–	–	TKIE	CKMIE	ADCIE	---- -000
PIR1	0x11	–	–	–	–	–	TKIF	CKMIF	ADCIF	---- -000
TIM1IER	0x215	T1BIE	T1TIE	–	T1CC4IE	T1CC3IE	T1CC2IE	T1CC1IE	T1UIE	00-0 0000
TIM1SR1	0x216	T1BIF	T1TIF	–	T1CC4IF	T1CC3IF	T1CC2IF	T1CC1IF	T1UIF	00-0 0000
TIM1SR2	0x217	–	–	–	T1CC4OF	T1CC3OF	T1CC2OF	T1CC1OF	–	---0 000-
TIM1EGR	0x218	–	–	–	T1CC4G	T1CC3G	T1CC2G	T1CC1G	–	---0 000-
TIM2IER	0x30D	–	–	–	–	T2CC3IE	T2CC2IE	T2CC1IE	T2UIE	---- 0000
TIM2SR1	0x30E	–	–	–	–	T2CC3IF	T2CC2IF	T2CC1IF	T2UIF	---- 0000
TIM2SR2	0x30F	–	–	–	–	T2CC3OF	T2CC2OF	T2CC1OF	–	---- 000-
TIM2EGR	0x310	–	–	–	–	T2CC3G	T2CC2G	T2CC1G	T2UG	---- 0000
TIM4IER	0x112	–	–	–	–	–	–	–	T4UIE	---- ---0
TIM4SR	0x113	–	–	–	–	–	–	–	T4UIF	---- ---0
TIM4EGR	0x114	–	–	–	–	–	–	–	T4UG	---- ---0

名称	地址	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	复位值(RW)
URIER	0x48E	–	–	TCEN	–	IDELE	RXSE	URTE	URRXNE	--0- 0000
URLSR	0x492	ADDRF	IDLEF	TXEF	BKF	FEF	PEF	OVERF	RXNEF	0010 0000
URTC	0x49C	–	–	–	–	–	–	–	TCF	---- ---1
EPIE0	0x94	外部中断控制寄存器								0000 0000
EPIF0	0x14	外部中断标志位寄存器								0000 0000
TRISA	0x8C	PORTA 方向控制								1111 1111
TRISB	0x8D	PORTB 方向控制								1111 1111
TRISC	0x8E	PORTC 方向控制								1111 1111
EPS0	0x118	外部中断 EINT3 ~ 0 管脚选择寄存器								0000 0000
EPS1	0x119	外部中断 EINT7 ~ 4 管脚选择寄存器								0000 0000

表 13-1 中断相关寄存器地址和默认值

名称	状态		寄存器	地址	复位值
GIE	全局中断 1 = 使能 (PEIE，各中断独立使能位适用) 0 = 全局关闭 (唤醒不受影响)		INTCON[7]	Bank 首地址 +0x0B	RW-0
PEIE	外设总中断 1 = 使能 (各中断独立使能位适用) 0 = 关闭 (无唤醒)		INTCON[6]		RW-0
EEIE	EE 写完成中断	1 = 使能 0 = 关闭 (无唤醒)	INTCON[5]		RW-0
LVDIE	LVD 中断		INTCON[4]		RW-0
OSFIE	时钟缺失中断		INTCON[3]		RW-0
EEIF [1]	EE 写完成标志位	1 = Yes (锁存) 0 = No	INTCON[2]		R_W1C-0
LVDIF [1]	LVD 中断标志位		INTCON[1]		R_W1C-0
OSFIF [1]	时钟缺失中断标志位		INTCON[0]		R_W1C-0

表 13-2 INTCON 寄存器

名称	状态		寄存器	地址	复位值
CKMIE	LIRC 和 HIRC 交叉校准完成中断	1 = 使能 0 = 关闭 (无唤醒)	PIE1[1]	0x91	RW-0
ADCIE	ADC 转换完成中断		PIE1[0]		RW-0
CKMIF [1]	LIRC 和 HIRC 交叉校准完成标志位	1 = Yes (锁存) 0 = No	PIR1[1]	0x11	R_W1C-0
ADCIF [1]	ADC 转换完成标志位		PIR1[0]		R_W1C-0

表 13-3 PIE1 和 PIR1 寄存器

[1] 写 1 清 0，写 0 无效。建议只使用 STR、MOVWI 指令进行写操作，而不要用 BSR 或 IOR 指令。

名称	状态		寄存器	地址	复位值
T1BIE	Timer1 刹车中断		TIM1IER[7]		RW-0
T1TIE	Timer1 触发中断		TIM1IER[6]		RW-0
T1COMIE	Timer1 换相（COM）中断		TIM1IER[5]		RW-0
T1CC4IE	Timer1 捕获/比较通道 4 中断		TIM1IER[4]		RW-0
T1CC3IE	Timer1 捕获/比较通道 3 中断		TIM1IER[3]	0x215	RW-0
T1CC2IE	Timer1 捕获/比较通道 2 中断		TIM1IER[2]		RW-0
T1CC1IE	Timer1 捕获/比较通道 1 中断		TIM1IER[1]		RW-0
T1UIE	Timer1 更新事件中断		TIM1IER[0]		RW-0
T1BG [2]	Timer1 刹车软件中断		TIM1EGR[7]		WO-0
T1TG [2]	Timer1 触发软件中断		TIM1EGR[6]		WO-0
T1COMG [2]	Timer1 换相（COM）软件中断		TIM1EGR[5]		WO-0
T1CC4G [2]	Timer1 捕获/比较通道 4 软件中断		TIM1EGR[4]		WO-0
T1CC3G [2]	Timer1 捕获/比较通道 3 软件中断	1 = 使能	TIM1EGR[3]	0x218	WO-0
T1CC2G [2]	Timer1 捕获/比较通道 2 软件中断	0 = 关闭 (无唤醒)	TIM1EGR[2]		WO-0
T1CC1G [2]	Timer1 捕获/比较通道 1 软件中断		TIM1EGR[1]		WO-0
T1UG [2]	Timer1 更新事件软件中断		TIM1EGR[0]		WO-0
T2CC3IE	Timer2 捕获/比较通道 3 中断		TIM2IER[3]		RW-0
T2CC2IE	Timer2 捕获/比较通道 2 中断		TIM2IER[2]		RW-0
T2CC1IE	Timer2 捕获/比较通道 1 中断		TIM2IER[1]	0x30D	RW-0
T2UIE	Timer2 更新事件中断		TIM2IER[0]		RW-0
T2CC3G [2]	Timer2 捕获/比较通道 3 软件中断		TIM2EGR[3]		WO-0
T2CC2G [2]	Timer2 捕获/比较通道 2 软件中断		TIM2EGR[2]		WO-0
T2CC1G [2]	Timer2 捕获/比较通道 1 软件中断		TIM2EGR[1]	0x310	WO-0
T2UG [2]	Timer2 更新事件软件中断		TIM2EGR[0]		WO-0
T4UIE	Timer4 更新事件中断		TIM4IER[0]	0x112	RW-0
T4UG [2]	Timer4 更新事件软件中断		TIM4EGR[0]	0x114	WO-0

表 13-4 TIMx 中断控制寄存器

[2] 软件置 1，硬件自动清 0。

名称	状态		寄存器	地址	复位值
T1BIF [3]	Timer1 刹车中断标志位	1 = Yes(锁存) 0 = No	TIM1SR1[7]	0x216	R_W1C-0
T1TIF [3]	Timer1 触发事件中断标志位		TIM1SR1[6]		R_W1C-0
T1COMIF [3]	Timer1 换相（COM）中断标志位		TIM1SR1[5]		R_W1C-0
T1CC4IF [3]	Timer1 捕获/比较通道 4 中断标志位		TIM1SR1[4]		R_W1C-0
T1CC3IF [3]	Timer1 捕获/比较通道 3 中断标志位		TIM1SR1[3]		R_W1C-0
T1CC2IF [3]	Timer1 捕获/比较通道 2 中断标志位		TIM1SR1[2]		R_W1C-0
T1CC1IF [3]	Timer1 捕获/比较通道 1 中断标志位		TIM1SR1[1]		R_W1C-0
T1UIF [3]	Timer1 更新事件中断标志位		TIM1SR1[0]		R_W1C-0
T1CC4OF [3]	Timer1 捕获/比较通道 4 重复捕获中断标志位		TIM1SR2[4]	0x217	R_W1C-0
T1CC3OF [3]	Timer1 捕获/比较通道 3 重复捕获中断标志位		TIM1SR2[3]		R_W1C-0
T1CC2OF [3]	Timer1 捕获/比较通道 2 重复捕获中断标志位		TIM1SR2[2]		R_W1C-0
T1CC1OF [3]	Timer1 捕获/比较通道 1 重复捕获中断标志位		TIM1SR2[1]		R_W1C-0
T2CC3IF [3]	Timer2 捕获/比较通道 3 中断标志位		TIM2SR1[3]	0x30E	R_W1C-0
T2CC2IF [3]	Timer2 捕获/比较通道 2 中断标志位		TIM2SR1[2]		R_W1C-0
T2CC1IF [3]	Timer2 捕获/比较通道 1 中断标志位		TIM2SR1[1]		R_W1C-0
T2UIF [3]	Timer2 更新事件中断标志位		TIM2SR1[0]		R_W1C-0
T2CC3OF [3]	Timer2 捕获/比较通道 3 重复捕获中断标志位		TIM2SR2[3]	0x30F	R_W1C-0
T2CC2OF [3]	Timer2 捕获/比较通道 2 重复捕获中断标志位		TIM2SR2[2]		R_W1C-0
T2CC1OF [3]	Timer2 捕获/比较通道 1 重复捕获中断标志位		TIM2SR2[1]		R_W1C-0
T4UIF [3]	Timer4 更新事件中断标志位		TIM4SR[0]	0x113	R_W1C-0

表 13-5 TIMx 中断标志寄存器

[3] 写 1 清 0，写 0 无效。建议只使用 STR、MOVWI 指令进行写操作，而不要用 BSR 或 IOR 指令。

名称	状态		寄存器	地址	复位值
URTE	发送 BUF 为空中断	1 = 使能 0 = 关闭	URIER[1]	0x48E	RW-0
TXEF	发送 BUF 状态	1 = 空 0 = 非空 注：写 DATAL(8bit) / DATAH(9bit)清零	URLSR[5]	0x492	RO-1
URRXNE	接收 BUF 非空中断	1 = 使能 0 = 关闭	URIER[0]	0x48E	RW-0
RXNEF	接收 BUF 状态	1 = 非空 0 = 空，或已被清零 注：读 DATAL(8bit) / DATAH(9bit)清零	URLSR[0]	0x492	RO-0
TCEN	发送完成中断	1 = 使能 0 = 关闭	URIER[5]	0x48E	RW-0
TCF	发送完成标志	1 = 完成 0 = 未完成 注：写 1 清零，或写 DATAL(8bit) /DATAH(9bit)后清零	URTC[0]	0x49C	R_W1C-1
IDELE	空闲帧中断	1 = 使能 0 = 关闭	URIER[3]	0x48E	RW-0
IDLEF [4]	检测到空闲帧标志	1 = 检测到 0 = 未检测到	URLSR[6]	0x492	RW0-0
RXSE [5]	接收状态中断	1 = 使能 0 = 关闭 接收状态中断产生条件： BKF = 1 FEF = 1 PEF = 1 OVERF = 1	URIER[2]	0x48E	RW-0
BKF [4]	接收到断开帧标志	1 = 接收到 0 = 未接收到，或已被清零	URLSR[4]	0x492	RW0-0
FEF [4]	接收帧错误标志	1 = 错误 0 = 正确，或已被清零	URLSR[3]	0x492	RW0-0
PEF [4]	接收奇偶校验错误标志	1 = 错误 0 = 正确，或已被清零	URLSR[2]	0x492	RW0-0

[4] 写 0 清零，写 1 无效。
[5] USART 接收到断开帧，帧错误，奇偶校验错误，接收溢出错误状态。

名称	状态		寄存器	地址	复位值
OVERF [6]	接收 BUF 溢出标志	1 = 溢出 0 = 正常，或已被清零	URLSR[1]	0x492	RW0-0
WAKE	哑模式唤醒方式选择	1 = 地址匹配 0 = IDLE 帧	URMCR[2]	0x491	RW-0
ADDRF	哑模式地址匹配标志	1 = 匹配 0 = 未匹配	URLSR[7]	0x492	RO-0

表 13-6 USART 中断使能和状态位

名称	状态		寄存器	地址	复位值
EPIEx	外部中断使能位	1 = 使能 0 = 禁止	EPIE0[7:0]	0x94	RW-00000000
EPIF0x [7]	外部中断标志位	1 = Yes (锁存) 0 = No	EPIF0[7:0]	0x14	R_W1C-00000000

表 13-7 EPIE0 和 EPIF0 寄存器

13.2. 外部中断管脚选择

名称	状态			寄存器	地址	复位值
EINT0	EINT0 管脚选择	00 = PA0 01 = PB0	10 = PC0 11 = 保留	EPS0[1:0]	0x118	RW-00
EINT1	EINT1 管脚选择	00 = PA1 01 = PB1	10 = PC1 11 = 保留	EPS0[3:2]		RW-00
EINT2	EINT2 管脚选择	00 = PA2 01 = PB2	1x = 保留	EPS0[5:4]		RW-00
EINT3	EINT3 管脚选择	00 = PA3 01 = PB3	1x = 保留	EPS0[7:6]		RW-00
EINT4	EINT4 管脚选择	00 = PA4 01 = PB4	1x = 保留	EPS1[1:0]	0x119	RW-00
EINT5	EINT5 管脚选择	00 = PA5 01 = PB5	1x = 保留	EPS1[3:2]		RW-00
EINT6	EINT6 管脚选择	00 = PA6 01 = PB6	1x = 保留	EPS1[5:4]		RW-00
EINT7	EINT7 管脚选择	00 = PA7 01 = PB7	1x = 保留	EPS1[7:6]		RW-00

表 13-8 外部中断管脚选择寄存器

[6] 写 0 清零，写 1 无效。
[7] 写 1 清 0，写 0 无效。建议只使用 STR、MOVWI 指令进行写操作，而不要用 BSR 或 IOR 指令。

13.3. 中断的使能

由图 13-1 可知，要想使用中断，除了该中断的相关使能位 (xxxIE) 需要置 1 外，总中断开关 (GIE) 也需要打开。ADC，慢时钟测量，TIMx，外部管脚，USART 模块被归类了外设中断，除了其自身的中断使能要打开外，另外一个外设总中断开关 PEIE 也需要置 1。

注：各中断标志位的置起并跟中断使能位无关。

13.4. 中断的响应时间

中断响应延时定义为从发生中断事件到开始执行中断向量处的代码所需要的时间。正常情况下 (即不处于 PROM 的写周期)，同步中断的响应延时为 3 或 4 个指令周期。对于异步中断，响应延时为 3 至 5 个指令周期，具体取决于中断发生的时间和正在执行的指令。

13.5. 睡眠下的中断

睡眠状态下系统时钟被关闭，部分使用系统时钟作为时钟的外设将停止工作。可将 CPU 唤醒的中断源有：

- 外部管脚中断
- EEPROM 写完成中断
- LVD 中断
- TIMx 中断 (使用慢时钟时)
- ADC 中断

注意：

1. 唤醒 CPU 不要求 GIE 使能。GIE 为 0 时，CPU 唤醒后将执行 SLEEP 指令后面的代码，而非中断向量。
2. 由于同步带来的延时，在对中断标志位清 0 之后，至少要等两条指令才可以执行 SLEEP 指令，否则 MCU 将不会进入睡眠。

当 CKOCON.SYSON 为 1 时，系统时钟保持运行，所以其它直接使用系统时钟的外设也可以把 CPU 唤醒，换言之，在这种条件下，所有中断都能唤醒 CPU。

13.6. 现场保护

进入中断时，中断控制单元将返回的 PC 地址保存在堆栈中。且，以下寄存器自动保存在影子寄存器中：

- W 寄存器
- STATUS 寄存器 (TO 和 PD 状态标志位除外)
- BSREG 寄存器
- FSR 寄存器
- PCLATH 寄存器

退出中断服务程序时，这些寄存器将自动恢复。在 ISR 期间对这些寄存器所做的任何修改都将丢失。如果需要修改任何这些寄存器，应修改 Bank31 中相应的影子寄存器，并在退出 ISR 时恢复此新值。根据用户的应用程序要求，可能还需要保存其他寄存器，这些额外的寄存器则需要用户自行处理。

14. 数据 EEPROM 和 PROM

数据 EEPROM 和程序存储器是可读可写的。这两种存储器没有直接映射到数据存储器空间，而是通过特殊功能寄存器 (SFR) 间接寻址。有 6 个 SFR 用于访问这两种存储器：

- EECON1
- EECON2
- EEDATL
- EEDATH
- EEADRL
- EEADRH

当与数据 EEPROM 模块接口时，EEDATL 寄存器存放要读写的 8 位数据，EEADRL 寄存器存放被访问的 EEDATL 单元的地址。这些器件具有 128 字节的数据 EEPROM，地址范围从 0h 到 07Fh。

访问程序存储器模块时，EEDATH:EEDATL 寄存器对形成双字节字，存放要读/写的 14 位数据，而 EEADRL 和 EEADRH 寄存器形成双字节字，存放被读取的程序存储单元的 12 位地址。

EEPROM 数据存储器允许以字节为单位进行读写。EEPROM 字节写操作会自动擦除目标存储单元并写入新数据 (在写入前擦除)。写入时间由片上定时器控制。写入和擦除电压由片上电荷泵产生，此电荷泵能在器件电压范围内正常工作，用于字节或字操作。

器件能否对程序存储器的特定块执行写操作取决于配置字寄存器位 FSECPB0[7:0]的设置。然而，始终允许程序存储器的读操作。

当器件被代码保护时，调试接口将不再能访问数据或程序存储器。在代码保护时，CPU 仍可继续读写数据 EEPROM 存储器和程序存储器。

14.1. DATA EEPROM 相关寄存器汇总

名称	状态		寄存器	地址	复位值
EEADR [1]	PROM / DATA EEPROM 地址	低 8 位	EEADRL[7:0]	0x191	RW-0000 0000
		高 4 位	EEADRH[3:0]	0x192	RW-000 0000
EEDAT [2]	PROM / DATA EEPROM 数据	低 8 位	EEDATL[7:0]	0x193	RW-xxxx xxxx
		高 6 位	EEDATH[5:0]	0x194	RW-xx xxxx
EEPGD	存储器选择位	1 = 访问 PROM 0 = 访问 DATA EEPROM	EECON1[7]	0x195	RW-0
CFGS	PROM / DATA EEPROM 或配置寄存器选择位 1 = 访问配置寄存器 (读访问) 0 = 访问 PROM 或 DATA EEPROM		EECON1[6]		RW-0
FREE [3]	PROM 擦除使能位 1 = 在下一条 WR 命令执行擦除操作 (擦除完成后由硬件清零) 0 = 在下一条 WR 命令执行写操作 注：仅当 CFGS = 0 且 EEPGD = 1 时有效		EECON1[4]	0x195	RW-0
WRERR	PROM / DATA EEPROM 擦除/写错误标志位 1 = 中止 (除 POR 之外的任何复位) 0 = 正常完成		EECON1[3]		RW-x
WREN [4]	编程/擦除使能位 1 = 使能 0 = 禁止		EECON1[2]		RW-0
WR [5]	PROM / DATA EEPROM 写控制位 1 = 启动一次写或写正在进行中(完成后重置为 0) 0 = 完成		EECON1[1]		RW1-0
RD	PROM / DATA EEPROM 读控制位 1 = Yes (保持 4 个 SysClk 周期后清零) 0 = No		EECON1[0]		RW1-0
EECON2	PROM / DATA EEPROM 写操作解锁控制寄存器 在 EECON1 寄存器的 WR 置位前，必须先写 0x55，随后是 0xAA，用于解锁写操作。这些写操作必须在连续的指令周期完成		EECON2[7:0]	0x196	WO-xxxx xxxx

表 14-1 EEPROM 相关用户控制寄存器

[1] 在写周期 (EEPROM: 3 ~ 5ms, PROM: 1.5 ~ 2.5ms) 内，该寄存器不可写。用该寄存器访问程序存储器时，地址范围必须位于 0~0xFFF，否则无法完成读写访问。

[2] 在写周期 (EEPROM: 3 ~ 5ms, PROM: 1.5 ~ 2.5ms) 内，该寄存器不可写。

[3] 访问数据 EEPROM 时：该位不起作用，下一条 WR 命令将启动一个擦除周期和一个写周期。

[4] 在写周期内，禁止对该寄存器位写。

[5] 软件写 1 后至少要等 1 个系统时钟才能回读。

名称	状态		寄存器	地址	复位值
GIE	全局中断 1 = 使能 (PEIE, EEIE 适用) 0 = 全局关闭 (唤醒不受影响)		INTCON[7]	Bank 首地址 +0x0B	RW-0
PEIE	外设总中断	1 = 使能 (EEIE 适用) 0 = 关闭 (无唤醒)	INTCON[6]		RW-0
EEIE	EEPROM 写完成中断	1 = 使能 0 = 关闭 (无唤醒)	INTCON[5]		RW-0
EEIF [6]	EEPROM 写完成中断标志位	1 = Yes (锁存) 0 = No	INTCON[2]		R_W1C-0

表 14-2 EEPROM 中断使能和状态位

14.2. EEADRL 和 EEADRH 寄存器

EEADRH:EEADRL 寄存器对可以寻址最大 128 字节的数据 EEPROM 或最大 4k 字的程序存储器。

当选择程序地址值时，地址的高字节写入 EEADRH 寄存器而低字节被写入 EEADRL 寄存器。当选择 EEPROM 地址值时，只将地址的低字节写入 EEADRL 寄存器。

14.2.1. EECON1 和 EECON2 寄存器

EECON1 是访问 EE 存储器的控制寄存器。
控制位 EEPGD 决定访问的是程序存储器还是数据存储器。当它为 0 时，任何后续操作都将针对 EEPROM 存储器进行。当置 1 时，任何后续操作都将针对程序存储器进行。复位后，EEPGD 清 0，即默认选中 EEPROM。

控制位 RD 和 WR 分别启动读和写。用软件只能将这些位置 1 而无法清零。在读或写操作完成后，由硬件将它们清零。由于无法用软件将 WR 位清零，可避免因意外而过早地终止写操作。

当 WREN 位置 1 时，允许执行写操作。上电时，WREN 位被清零。在正常运行中当写操作被复位中断时，WRERR 位置 1。在这些情况下，复位后用户可以检查 WRERR 位并执行相应的错误处理程序。当写操作完成时，PIR2 寄存器的中断标志位 EEIF 被置 1。该标志位必须用软件清零。

读 EECON2 得到的是全 0。EECON2 寄存器仅在数据 EEPROM 写过程中使用。要使能写操作，必须将特定模式写入 EECON2。

14.3. 使用数据 EEPROM

数据 EEPROM 是高耐久性、可字节寻址的阵列，已将其优化以便存储频繁更改的信息 (例如：程序变量或其他经常更新的数据)。软件开发者应将不频繁更改的变量 (例如常量、ID 和校准值等) 存储在闪存程序存储器中，以免超出 EEPROM 字节最大的写允许次数。

[6] 写 1 清 0，写 0 无效。建议只使用 STR、MOVWI 指令进行写操作，而不要用 BSR 或 IOR 指令。

14.3.1. 读数据 EEPROM 存储器

要读取数据存储单元，用户需要把地址写入 EEADRL 寄存器，清零 EECON1 寄存器的 EEPGD 和 CFGS 控制位，再置 1 控制位 RD。等待 4 个系统时钟周期后，数据将更新到 EEDATL 寄存器中，EEDATL 将把此值保持至下一次读取或用户向该单元写入数据时 (在写操作过程中) 为止。

读数据 EEPROM，示例：

```
BANKSEL   EEADRL          ;
LDWI      DATA_EE_ADDR    ;
STR       EEADRL           ;Data Memory Address to read
BCR       EECON1, CFGS     ;Deselect Config space
BCR       EECON1, EEPGD    ;Point to DATA memory
BSR       EECON1, RD       ;EE Read
; wait 4 system clock
LDR       EEDATL, W        ;W = EEDATL
```

注意：无论 CPB 为何值，软件总是可以读取 EEPROM；

14.3.2. 写数据 EEPROM 存储器

要写 EEPROM 数据存储单元，用户应首先将该单元的地址写入 EEADRL 寄存器并将数据写入 EEDATL 寄存器。然后用户必须按特定顺序开始写入每个字节 (即，首先将 0x55 写入 EECON2，随后将 0xAA 写入 EECON2，最后将 WR 位置 1)。如果未完全按照上述顺序逐字节写入，将不会启动写操作。在该代码段中应禁止中断。

必须将 EECON1 中的 WREN 位置 1 以使能写操作。这种机制可防止由于代码执行错误 (异常) 导致误写数据 EEPROM。除了更新 EEPROM 时，用户应始终保持 WREN 位清零。WREN 位不能由硬件清零。

一个写序列启动后，清零 WREN 位将不会影响此写周期。除非 WREN 位置 1，否则 WR 位将无法置 1。写周期完成时，WR 位由硬件清零并且 EE 写完成中断标志位 (EEIF) 置 1。用户可以允许中断或查询此位。EEIF 必须用软件清零。

注意：软件对 EECON1.WR 写 1 后，至少等待一个系统时钟 (NOP 或者任何别的指令) 软件才能对该位进行读判断，否则将读回 0，进而影响程序的流程 (比如误认为写结束)。

14.3.3. 防止误写操作的保护措施

有些情况下，用户并不希望向数据 EEPROM 存储器写入数据。为了防止 EEPROM 误写操作，器件内建了各种保护机制。上电时，清零 WREN。同时，上电延时定时器 (64ms 的延时) 也会阻止对 EEPROM 进行写操作。

写启动序列和 WREN 位共同防止在以下情况下发生意外写操作：

- 欠压
- 电源故障
- 软件故障

14.3.4. 关于 GIE 的清 0

在启动 EEPROM 和 PROM 写之前，需对 EECON2 顺序写 0x55 和 0xAA，且不能被打断。所以在该开锁动作前应确认 GIE 已清 0 (以屏蔽可能的中断)。由于中断的响应延时为 2 个 NOP，代码示例如下：

```
GIE= 0;
NOP;
NOP;
while (GIE) { GIE= 0;};
```

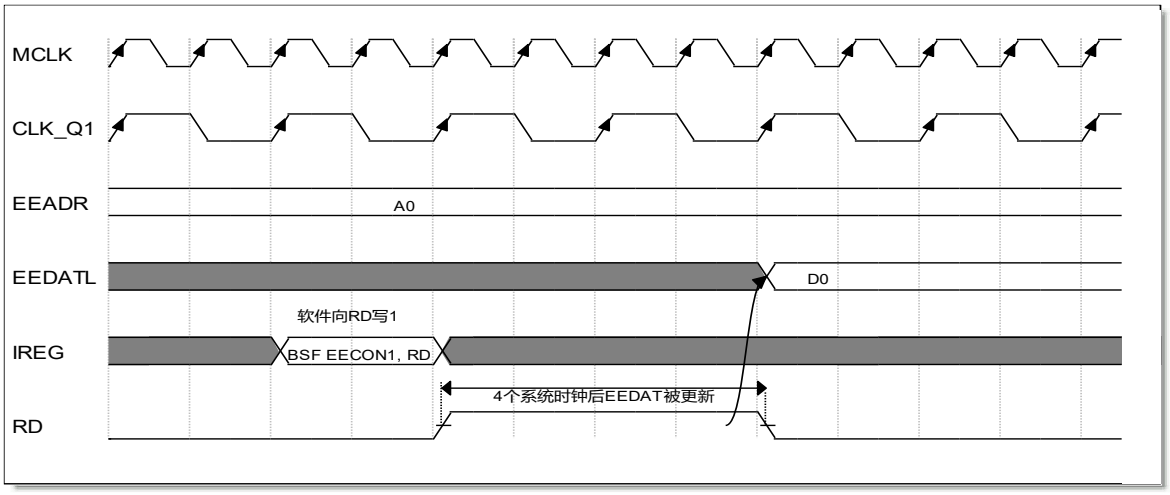


图 14-1 2T 模式下软件读 EEPROM 时序

写数据 EEPROM，示例：

```
BANKSEL EEADRL                  ;
LDWI DATA_EE_ADDR  ;
STR EEADRL                      ;Data Memory Address to write
LDWI DATA_EE_DATA   ;
STR EEDATL                      ;Data Memory Value to write
BCR EECON1, CFGS                ;Deselect Configuration space
BCR EECON1, EEPGD               ;Point to DATA memory
BSR EECON1, WREN                ;Enable writes

BCR INTCON, GIE                 ;Disable INTs.
NOP
NOP
BTSC INTCON, GIE               ;Test again
GOTO $-1


LDWI 55h                        ;
STR EECON2                      ;Write 55h
LDWI 0AAh               ;
STR EECON2                      ;Write AAh

BSR EECON1, WR                   ;Set WR bit to begin write
BSR INTCON, GIE                 ;Enable Interrupts
BCR EECON1, WREN                ;Disable writes
BTSC EECON1, WR                 ;Wait for write to complete
LJUMP $-2                       ;Done
```

该程序流不能被中断

注意：

1. 数据 EEPROM 写周期并不暂停指令的执行。
2. 清完 GIE 之后等待两个 NOP 再判断一个 GIE 是否被清掉，以避免因中断返回的 GIE 置 1。

14.4. 闪存程序存储器概述

了解程序存储器结构对于擦除和编程操作非常重要。程序存储器按行排列。每行包括固定数量的 14 位程序存储器字。行是用户软件可擦除的最小块大小。只有当目标地址位于未受写保护的存储器段内 (由配置字寄存器的 CPB 和 FSECPB0 定义)，才能对程序存储器进行写或擦除操作。擦除某行后，用户可以对该行的部分或全部进行重新编程。写入程序存储器行的数据将被写入 14 位宽的数据写锁存器中。用户不能直接访问这些写锁存器，但是可以通过对 EEDATH:EEDATL 寄存器对的连续写入来加载写锁存器的内容。

数据写锁存器数并不等于行单元数，例如 FT61F14x 一行 (页) 有 32 个单元，但只有 1 个写锁存器。编程时，用户软件需要填充该组写锁存器并多次启动编程操作，才能完全重新编程擦除的行 (页)。例如，具有 32 字的行大小和 1 个写锁存器的器件需要将数据装入写锁存器并启动编程操作 32 次。

注：如果用户只想修改部分以前编程的行，那么必须读取整行的内容并保存在 RAM 中，然后进行擦除。

14.4.1. 读程序存储器

要读程序存储单元，用户必须：

1) 将最低有效地址位和最高有效地址位写入 EEADRH:EEADRL 寄存器对
2) 将 EECON1 寄存器的 CFGS 位清零
3) 将 EECON1 寄存器的 EEPGD 控制位置 1
4) 然后，将 EECON1 寄存器的控制位 RD 置 1

一旦将读控制位置 1，闪存程序存储器控制器将使用第二个指令周期读取数据。这会导致紧跟“BSR EECON1,RD”指令后的第二条指令被忽略。在紧接着的下一个周期，EEDATH:EEDATL 寄存器对中就有数据了，因此可在随后的指令中将该数据读作两个字节。

EEDATH:EEDATL 寄存器对将把此值保存至下一次读操作或用户向该单元写入数据时为止。

注：

1. 要求程序存储器读操作后的两条指令为 NOP。这可以防止用户在 RD 位置 1 后的下一条指令执行双周期指令；
2. 不管 CP 位的设置如何，软件都可以读取程序存储器；

读程序存储器，示例

```
* This code block will read 1 word of program
* memory at the memory address: PROG_ADDR_HI: PROG_ADDR_LO
* data will be returned in the variables: PROG_DATA_HI, PROG_DATA_LO

BANKSEL EEADRL                  ; Select Bank for EEPROM registers
LDWI PROG_ADDR_LO               ;
STR EEADRL                      ; Store LSB of address

LDWI PROG_ADDR_HI               ;
STR EEADRH                      ; Store MSB of address

BCR EECON1,CFGS                 ; Do not select Configuration Space
BSR EECON1,EEPGD                ; Select Program Memory

BCR INTCON,GIE                  ; Disable interrupts

BSR EECON1,RD                   ; Initiate read

NOP                             ; Executed(Figure 8.3.1)
NOP                             ; Ignored(Figure 8.3.1)

BSR INTCON,GIE                  ; Restore interrupts

LDR EEDATL,W                    ; Get LSB of word
STR PROG_DATA_LO                ; Store in user location

LDR EEDATH,W                    ; Get MSB of word
STR PROG_DATA_HI                ; Store in user location
```

14.4.2. 擦除闪存程序存储器

当执行代码时，程序存储器只能按行擦除。要擦除行：

1) 将要擦除的新行地址装入 EEADRH:EEADRL 寄存器对
2) 将 EECON1 寄存器的 CFGS 位清零
3) 将 EECON1 寄存器的 EEPGD、FREE 和 WREN 位置 1
4) 依次将 0x55 和 0xAA 写入 EECON2 (编程解锁序列)
5) 将 EECON1 寄存器的控制位 WR 置 1，以开始擦除操作
6) 查询 EECON1 寄存器的 FREE 位，以确定行擦除何时结束

程序存储器的行擦除，示例

```
; This row erase routine assumes the following:
; 1. A valid address within the erase block is loaded in ADDRH:ADDRL
; 2. ADDRH and ADDRL are located in shared data memory 0x70 - 0x7F (common RAM)

BCR INTCON,GIE        ; Disable ints so required sequences will execute properly

NOP
NOP

BTSC INTCON, GIE
GOTO $-1

BANKSEL EEADRL
LDR ADDRL,W           ; Load lower 8 bits of erase address boundary
STR EEADRL

LDR ADDRH,W           ; Load upper 6 bits of erase address boundary
STR EEADRH

BSR EECON1,EEPGD ; Point to program memory
BCR EECON1,CFGS   ; Not configuration space
BSR EECON1,FREE   ; Specify an erase operation
BSR EECON1,WREN   ; Enable writes

LDWI 55h   ; Start of required sequence to initiate erase
STR EECON2            ; Write 55h
LDWI 0AAh             ;
STR EECON2            ; Write Aah

BSR EECON1,WR         ; Set WR bit to begin erase
NOP                   ; Any instructions here are ignored as processor
                      ; halts to begin erase sequence
NOP                   ; Processor will stop here and wait for erase complete.
                      ; after Erase processor continues with 3rd instruction

BCR EECON1,WREN   ; Disable writes

BSR INTCON,GIE         ; Enable interrupts
```

该程序流不能被中断

在“BSR EECON1,WR”指令后，处理器需要两个周期来设置擦除操作。用户必须在 WR 位置 1 后，执行两条 NOP 指令。处理器将暂停内部操作，通常为 2ms 擦除时间。这不是休眠模式，因为时钟和外设将继续运行。擦除周期后，处理器从 EECON1 写指令后的第三条指令继续操作。

14.4.3. 写程序存储器

使用以下步骤编程程序存储器：

1) 装入要编程的字的起始地址
2) 将数据装入写锁存器
3) 启动编程操作
4) 重复第 1 至 3 步，直到写入所有数据

写入程序存储器之前，要写入的字必须已擦除或者以前未写入过。程序存储器一次只能擦除一行。启动写操作时不会自动擦除。

程序存储器一次只能写入一个字，请参见图 14-2 (对带 1 个写锁存器的程序存储器进行字写操作)。写操

作不会跨这些边界。程序存储器写操作完成时，写锁存器将复位为 0x3FFF。

应完成以下步骤，以装载写锁存器和编程程序存储块。可按以下步骤操作：装载数据到写锁存器，启动编程序列。需要特殊的解锁序列才能将数据装入写锁存器或启动编程操作，不应中断此解锁序列。

1) 将 EECON1 寄存器的 EEPGD 和 WREN 位置 1

2) 将 EECON1 寄存器的 CFGS 位清零

3) 将要写入的单元地址装入 EEADRH:EEADRL 寄存器对

4) 依次将 0x55 和 0xAA 写入 EECON2，然后将 EECON1 寄存器的 WR 位置 1 (编程解锁序列)

5) 等待约 2ms 时间，锁存器数据写入程序存储器

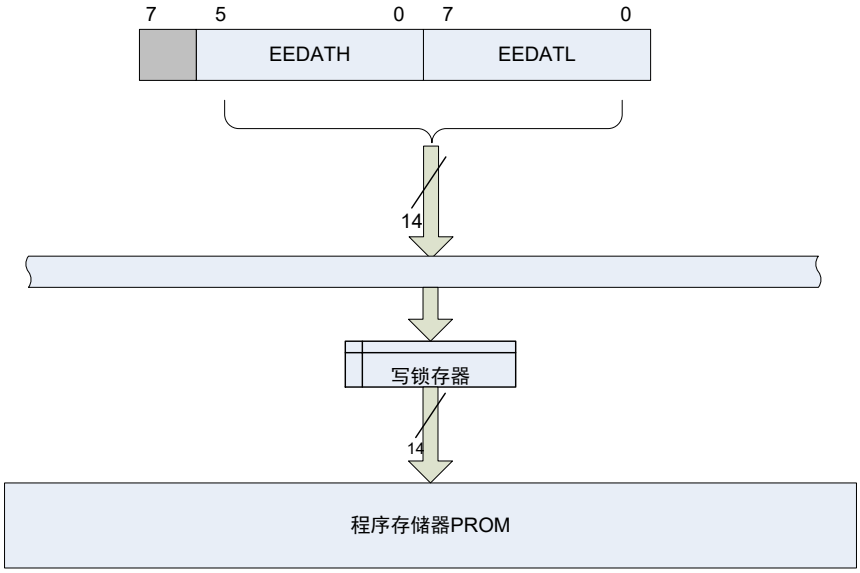


图 14-2 对带 1 个写锁存器的 PROM 进行写操作

注：完整的单字写序列，将初始地址装入 EEADRH:EEADRL 寄存器对，数据通过间接寻址载入。示例中提供的代码序列必须重复多次，以完全编程擦除的程序存储器行。对于 FT61F14X 系列芯片，一行 (页) 有 32 个 WORD，该代码序列需要重复 32 遍。

对包含 1 个写锁存器的 PROM 编程，示例：

```
; This write routine assumes the following:
; 1. The 2 bytes of data are loaded, starting at the address in DATA_ADDR
; 2. Each word of data to be written is made up of two adjacent bytes in DATA_ADDR, stored in little endian format
; 3. A valid starting address is loaded in ADDRH:ADDRL
; 4. ADDRH and ADDRL are located in shared data memory 0x70 - 0x7F (common RAM);

   BCR INTCON,GIE    ; Disable ints so required sequences will execute properly

   NOP
   NOP

   BTSC INTCON, GIE
   GOTO $-1

   BANKSEL EEADRH ; Bank 3
   LDR ADDRH,W       ; Load target address
   STR EEADRH        ;
   LDR ADDRL,W       ;
   STR EEADRL        ;

   LDWI LOW DATA_ADDR ; Load initial data address
   STR FSR0L ;
   LDWI HIGH DATA_ADDR ; Load initial data address
   STR FSR0H ;

   BSR EECON1,EEPGD ; Point to program memory
   BCR EECON1,CFGS              ; Not configuration space
   BSR EECON1,WREN              ; Enable writes

   MOVIW FSR0++      ; Load first data byte into lower
   STR EEDATL        ;
   MOVIW FSR0++      ; Load second data byte into upper
   STR EEDATH        ;

START_WRITE:

   LDWI 55h          ; Start of required write sequence:
   STR EECON2        ; Write 55h
   LDWI 0AAh         ;
   STR EECON2        ; Write Aah

   BSR EECON1,WR     ; Set WR bit to begin write
   NOP               ; Any instructions here are ignored as processor
                     ; halts to begin write sequence
   NOP               ; Processor will stop here and wait for write complete.
                     ; after write processor continues with 3rd instruction

   BCR EECON1,WREN              ; Disable writes

   BSR INTCON,GIE    ; Enable interrupts
```

该程序流不能被中断

14.5. 修改闪存程序存储器

修改程序存储器行中的现有数据，且该行内的数据需保留时，必须先读取它并将其保存在 RAM 映像中。

使用以下步骤修改程序存储器：

1) 装入要修改的行的起始地址
2) 从行中读取现有数据并将其保存到 RAM 映像中
3) 修改 RAM 映像以包含要写入到程序存储器的新数据
4) 装入要重新写入的行的起始地址

5) 擦除程序存储器行

6) 将来自 RAM 映像的数据装入写锁存器

7) 启动编程操作

根据需要重复第 6 至 7 步多次，以对擦除行进行重新编程。

14.6. 配置字 UCFGx 读访问

当 EECON1 寄存器中的 CFGS = 1 时，不是访问程序存储器或 EEPROM 数据存储器，而是访问配置字 UCFGx。这是 PC[15] = 1 时指向的区域，即当软件发起读操作时，地址是 [0x8000 + EADDR]，但不是所有的地址都可访问，对于未实现的单元，读返回未定义。

14.7. 写校验

根据具体应用，将写入数据 EEPROM 或者程序存储器中的值对照期望写入的值进行校验（见如下例程）是一个良好的编程习惯。

对数据 EEPROM 写校验，示例：

```
BANKSEL     EEDATL          ;
LDR         EEDATL, W       ;EEDATL not changed from previous write
BSR         EECON1, RD      ;YES, Read the value written
XORWR       EEDATL, W       ;
BTSS        STATUS, Z       ;Is data the same
LJUMP       WRITE_ERR       ;No, handle error
                            ;Yes, continue
```

14.8. PROM 内容保护

PROM 控制器内置加密保护单元，具备以下特性：

- 全区加密，由 CPB 位控制
- 分扇区加密，1 扇区=1k words，由 FSECPB0 寄存器控制
- 解除加密只能通过执行一次包含 UCFG 页在内的全芯片擦除

全加密和分扇区加密区别如下表：

加密方式	CPU 取指	软件读	软件写	串口读	串口写
无	√	√	√(2)	√	√
全区	√	√	√(2)	×(1)	×(4)
分扇区	√	×(1)	×(3)	×(3)	×(5)

注意：

1) EEDAT 保持旧值不变；

2) 软件不可以编程或擦除 UCFG 页；

3) 只可以读未加密的扇区；

4) 只允许串口做包括 UCFG 在内的全芯片擦除；

5) 只允许串口做包括 UCFG 在内的全芯片擦除，或者对未加密的扇区做页擦除，编程；

6) 任何情况下，软件都不可以做全芯片擦除

15. 12bit ADC 模块

模数转换器 (Analog-to-digital Converter，ADC) 可将模拟输入信号转换为相应的 12 位二进制表征值。该系列器件采用多个模拟输入复用到一个采样保持电路。采样保持电路的输出与转换器的输入相连接。转换器通过逐次逼近法产生 12 位二进制值，并将转换结果保存在 ADC 结果寄存器 (ADRESL:ADRESH) 中。ADC 参考电压可用软件选择为 VDD、外部参考电压或内部产生的参考电压。ADC 可在转换完成时产生中断。该中断可用于将器件从休眠唤醒。

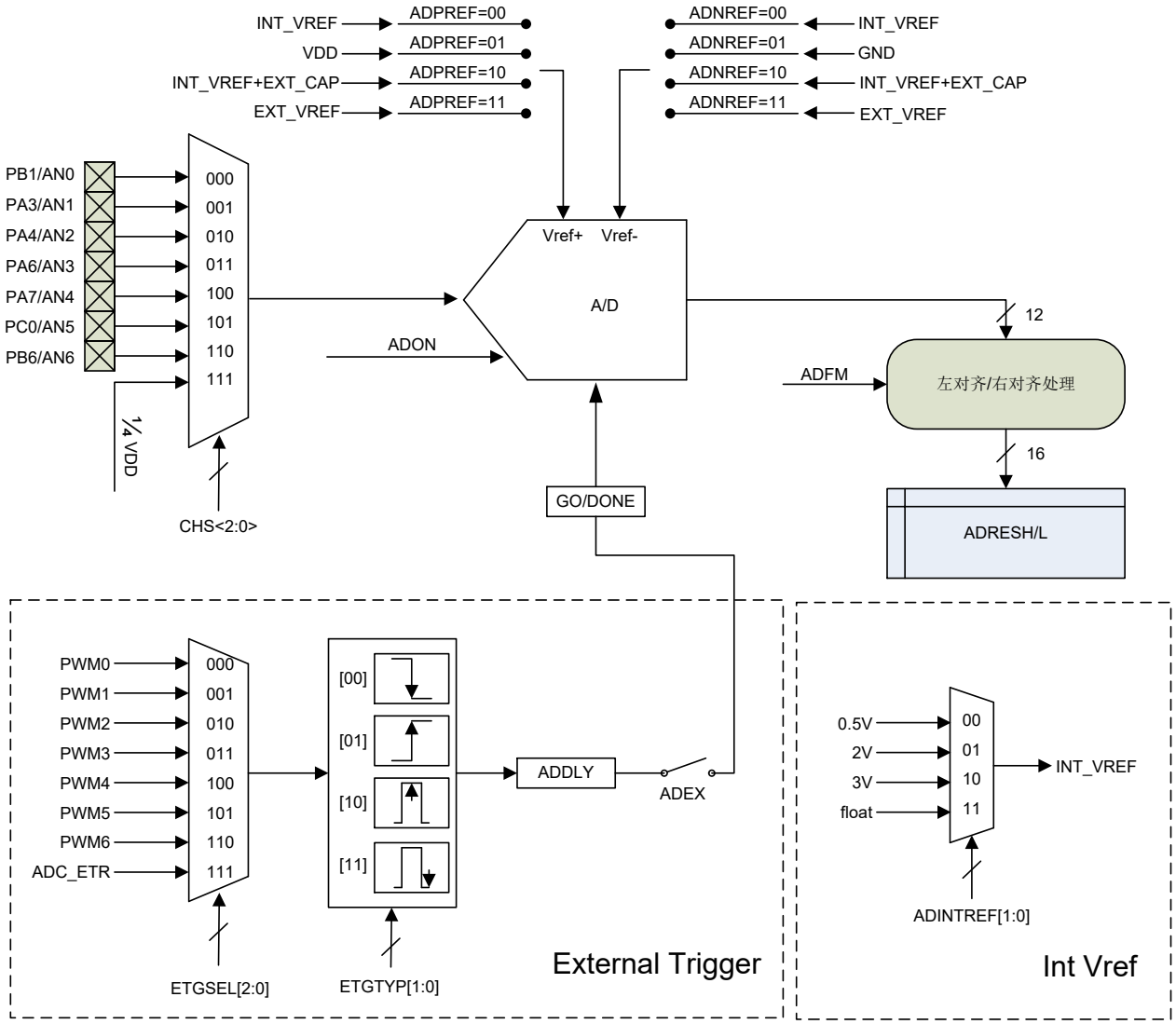


图 15-1 ADC 原理框图

15.1. ADC 相关寄存器汇总

名称	状态	寄存器	地址	复位值
ADRESL	ADC 转换结果低有效位 (LSB) ADFM=0: ADRESL[7:4] = 低 4 位 (其余为“0”) ADFM=1; ADRESL[7:0] = 低 8 位	ADRESL[7:0]	0x9B	RW-0000 0000
ADRESH	ADC 转换结果高有效位 (MSB) ADFM=0: ADRESH[7:0] = 高 8 位 ADFM=1: ADRESH[3:0] = 高 4 位 (其余为“0”)	ADRESH[7:0]	0x9C	RW-0000 0000
ADCEN	ADC 模块时钟 1 = 使能 0 = 关闭	PCKEN[0]	0x9A	RW-0
CHS	ADC 模拟输入通道 000 = AN0 100 = AN4 001 = AN1 101 = AN5 010 = AN2 110 = AN6 011 = AN3 111 = 1/4 V_{DD}	ADCON0[6:4]	0x9D	RW-000
ADEX	ADC 触发条件 (GO/DONE) 1 = 由 PA4 或 PWM 置位 GO/DONE (硬件触发) 0 = 由指令置位 GO/DONE (软件触发)	ADCON0[2]		RW-0
GO/DONE	ADC 转换启动和状态位 1 = 由软件, PA4 或 PWM 启动 A/D 转换 (转换完成后自动清零) 0 = 转换完成 / 未进行转换	ADCON0[1]		RW-0
ADON	1 = ADC 使能 0 = ADC 关闭 (无电流消耗)	ADCON0[0]		RW-0
LFMOD	1: LIRC = 256 kHz 0: LIRC = 32 kHz	TCKSRC[7]	0x31F	RW-0
ADFM	A/D 转换结果格式 (参阅 “ADRESH”) 1 = 右对齐 0 = 左对齐	ADCON1[7]	0x9E	RW-0
ADCS	ADC 转换时钟源 000 = SysClk/2 100 = SysClk/4 001 = SysClk/8 101 = SysClk/16 010 = SysClk/32 110 = SysClk/64 011 = LIRC 111 = LIRC	ADCON1[6:4]		RW-000

名称	状态	寄存器	地址	复位值
ADNREF	$V_{ADC-REF}$ – (负参考电压) 00 = 内部 $V_{ADC-REF}$ 01 = GND 10 = 内部 $V_{ADC-REF}$ + 外部电容 Cap 11 = 外部参考电压 (I/O)	ADCON1[3:2]		RW-00
ADPREF	$V_{ADC-REF}$ + (正参考电压) 00 = 内部 $V_{ADC-REF}$ 01 = V_{DD} 10 = 内部 $V_{ADC-REF}$ + 外部电容 Cap 11 = 外部参考电压 (I/O)	ADCON1[1:0]		RW-00
ADINTREF	内部 $V_{ADC-REF}$ 00 = 0.5 01 = 2.0 10 = 3.0 11 = (未连接)	ADCON2[7:6]		RW-00
ETGTYP	外部触发沿 (当 ADEX=1 时适用) 00 = (PWM 或 PA4-ADC_ETR) 下降沿 01 = (PWM 或 PA4-ADC_ETR) 上升沿 10 = 一个 PWM 周期的中点 11 = 一个 PWM 周期的终点 注：中点和终点触发仅中心对齐模式 PWM 输出使用	ADCON2[5:4]	0x9F	RW-00
ADDLY.8 / LEBPR9	ADC 延迟计数器或 LEB 计数器的第 8 位 (参阅 “ADDLY”)	ADCON2[3]		RW-0
ETGSEL	外部触发源 (当 ADEX=1 时适用) 000 = PWM1, TIM1_CH1　100 = PWM5, TIM2_CH1 001 = PWM2, TIM1_CH2　101 = PWM6, TIM2_CH2 010 = PWM3, TIM1_CH3　110 = PWM7, TIM2_CH3 011 = PWM4, TIM1_CH4　111 = ADC_ETR	ADCON2[2:0]		RW-000
ADDLY / LEBPRL	ADC 延迟/ LEB (非软件触发，ADEX = 1) (此为低 8 位, ADDLY.8 为高有效位) 延迟时间 = (ADDLY+6)/F_{ADC} (如果启用 PWM 输出触发 ADC，在 PWM 运行过程中不得更改 ADDLY)	ADDLY[7:0]	0x1F	RW-0000 0000
ADFBEN	ADC 阈值比较结果匹配事件触发 PWM 故障刹车 1 = 使能 0 = 关闭	ADCON3[7]	0x41A	RW-0

名称	状态	寄存器	地址	复位值
ADCMPOP	ADC 阈值比较的极性 1 = ADC 结果的高 8 位 < ADCMPH[7:0] 0 = ADC 结果的高 8 位 ≥ ADCMPH[7:0]	ADCON3[6]		RW-0
ADCMPEN	ADC 阈值比较 1 = 使能 0 = 关闭	ADCON3[5]		RW-0
ADCMPO	ADC 比较结果输出位 输出比较结果 (每次 AD 转换结束都会更新输出)	ADCON3[4]		RO-0
LEBADT	LEB 结束后，ADC 开始自动转换 1 = 触发 ADC 转换 0 = 不触发 ADC 转换	ADCON3[3]		RW-0
ADCMPH	ADC 比较阈值 (仅高 8 位，0.4% steps)	ADCMPH[7:0]	0x41B	RW-0000 0000
LEBEN	ADC 触发和 BKIN 的 LEB 使能位 1 = 使能 (当 GO/DONE=1 时进行切换将产生不可预知的结果) 0 = 关闭	LEBCON[7]	0x41C	RW-0
LEBCH	LEB 信号源 00 = TIM1_CH1　10 = TIM1_CH3 01 = TIM1_CH2　11 = TIM1_CH4	LEBCON[6:5]		RW-00
EDGS	LEB 触发沿 1 = 下降沿　0 = 上升沿	LEBCON[3]		RW-0

表 15-1 ADC 相关用户寄存器

名称	状态		寄存器	地址	复位值
GIE	全局中断 1 = 使能 (PEIE, ADCIE 适用) 0 = 全局关闭 (唤醒不受影响)		INTCON[7]	Bank 首地址 + 0x0B	RW-0
PEIE	外设总中断	1 = 使能 (ADCIE 适用) 0 = 关闭 (无唤醒)	INTCON[6]		RW-0
ADCIE	ADC 转换完成中断	1 = 使能 0 = 关闭 (无唤醒)	PIE1[0]	0x91	RW-0
ADCIF [1]	ADC 转换完成中断标志位	1 = Yes (锁存) 0 = No	PIR1[0]	0x11	R_W1C-0

表 15-2 ADC 中断使能和状态位

[1] 写 1 清 0，写 0 无效。建议只使用 STR、MOVWI 指令进行写操作，而不要用 BSR 或 IOR 指令。

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
PCKEN	0x9A	—	—	UARTEN	—	TIM4EN	TIM2EN	TIM1EN	ADCEN	0000 0000
CKOCON	0x95	SYSON	CCORDY	DTYSEL		CCOSEL[2:0]			CCOEN	0010 0000
ADRESL	0x09B	A/D 结果寄存器的低字节								0000 0000
ADRESH	0x09C	A/D 结果寄存器的高字节								0000 0000
ADCON0	0x09D	—	CHS[2:0]			—	ADEX	GO/DONE	ADON	-000 -000
ADCON1	0x09E	ADFM	ADCS[2:0]			ADNREF[1:0]		ADPREF[1:0]		0000 0000
ADCON2	0x09F	ADINTREF[1:0]		ETGTYP[1:0]		ADDLY.8	ETGSEL[2:0]			0000 0000
ADDLY	0x01F	ADDLY[7:0] / LEBPRL[7:0]								0000 0000
ADCON3	0x41A	ADFBEN	ADCMPOP	ADCMPEN	ADCMPO	LEBADT	—	ELVDS[1:0]		0000 0-00
ADCMPH	0x41B	ADCMPH[7:0]								0000 0000
LEBCON	0x41C	LEBEN	LEBCH		—	EDGS	BKS2	BKS1	BKS0	000- 0000

表 15-3 ADC 相关用户寄存器地址

15.2. ADC 的配置

配置和使用 ADC 时，必须考虑以下功能：

- 端口配置
- 通道选择
- 触发方式选择
- 触发源选择
- 触发类型选择
- 触发延时配置
- ADC 参考电压的选择
- ADC 转换时钟源
- 中断控制
- 转换结果的格式
- 阈值比较

注意：在进行各项配置更改的时候，需要确保 AD 转换并未正在进行或外部触发功能未开启。建议在 ADON 关闭时进行更改。

15.2.1. 端口配置

ADC 可用于转换模拟和数字信号。转换模拟信号时，应将相关的 TRIS 和 ANSELA 位置 1 将 I/O 引脚应配置为模拟功能。更多信息请参见相应的端口章节。

注意：如果定义为数字输入的引脚上存在模拟电压，会导致输入缓冲器传导过大的电流。

15.2.2. 通道选择

ADCON0 寄存器的 CHS 位决定将哪个通道连接到采样保持电路。改变通道时，根据采样稳定的需要可

在启动转换前加入一定延时。更多信息请参见**章节 15.3** “ADC 的工作原理”。

15.2.3. 触发方式选择

ADCON0 寄存器的 ADEX 位决定是否使用外部触发信号。
若 ADEX=0 时，GO/DONE 可由程序置位，AD 转换完成自动清零。
若 ADEX=1 时，GO/DONE 将由外部硬件触发置位，AD 转换完成清零。

注意：若选择了前沿消隐触发 ADC，即 LEBADT 设为 1 时，需要先置位 ADEX 和 ADON。

15.2.4. 触发源选择

在设定 ADEX 后，ADCON2 寄存器的 ETGSEL 位决定使用哪个外部触发信号。其中可选 I/O 引脚触发，需要配置相关寄存器。具体请参见相应的端口章节。

15.2.5. 触发类型选择

ADCON2 寄存器的 ETGTYP 位决定外部触发信号的触发类型。
其中选择 PWM 的中点或终点类型时，触发源将会默认选择 TIM1 中心对齐的 PWM 输出信号。具体请参见相应的 TIM1 章节。

15.2.6. 触发延时配置

ADCON2 寄存器的 ADDLY.8 位和 ADDLY 寄存器组成 9 位延时计数器，共同决定外部触发信号的触发延时时间。由于需要同步异步信号，实际延迟时间为：$(ADDLY+6)/F_{ADC}$。

注意：若选择了前沿消隐触发功能时，则实际延迟时间为：$(ADDLY+3)/F_{TIM1} + 3/F_{ADC}$。

15.2.7. ADC 参考电压

ADCON1 寄存器的 ADPREF 位提供对正参考电压的控制，ADNREF 位提供对负参考电压的控制。正/负参考电压可以是内部参考电压、VDD/GND、内部参考电压加外部电容、外部参考电压。正/负参考电压可以有各种组合，但不可以同时选择内部参考电压。若发生则强制负参考电压连接到 GND。
ADCON2 寄存器的 ADINTREF 位提供对内部参考电压的控制。内部参考电压可以选择 0.5V、2V、3V 或者悬空。

15.2.8. 转换时钟

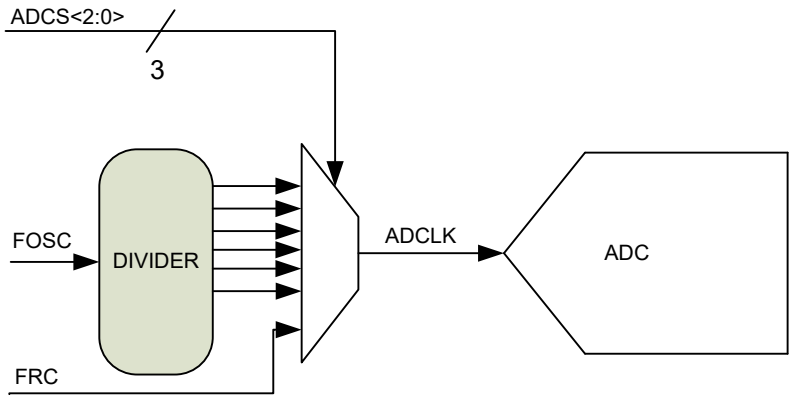


图 15-2 ADC 的时钟配置原理

转换时钟源可通过 ADCON1 寄存器的 ADCS 位用软件选择。有以下 7 种时钟选项：

- FOSC/2
- FOSC /4
- FOSC /8
- FOSC /16
- FOSC /32
- FOSC /64
- FRC (内部慢时钟振荡器)

完成一位 (bit) 的转换时间定义为 T_{AD}。完成 12 位转换需要 14 个 TAD 周期 ($12T_{AD}$ 数据转换时间，$1T_{AD}$ 的数据传输处理时间，$1T_{AD}$ 的数据清除时间)，如图 15.3 和 15.7 所示。

进行正确的转换必须满足相应的 T_{AD} 规范。更多信息请参见**章节 21.7** “电气特性”中的 A/D 转换要求。**表 15-4** 所示为正确选择 ADC 时钟的示例。

注意：

1. 除非使用的是 FRC，否则任何系统时钟频率的变化均会改变 ADC 时钟频率，这将对 ADC 结果产生负面影响；
2. FRC 可以是 256KHz 或者是 32KHz，取决于 LFMOD 为何值；
3. 当 SYSON 为 0 时，在睡眠模式下 AD 转换结束并不能唤醒 MCU，要想实现 AD 转换结束后唤醒，必须把 SYSON 置 1；
4. 下表格中的灰色部分是不支持的频率；

ADC 时钟周期 (T_{AD})		系统时钟频率 (Sysclk)			
ADC 时钟源	ADCS[2:0]	16MHz	8MHz	4MHz	1MHz
F_{OSC} /2	000	0.125μs	0.25μs	0.5μs	2.0μs
F_{OSC} /4	100	0.25μs	0.5μs	1.0μs	4.0μs
F_{OSC} /8	001	0.5μs	1.0μs	2.0μs	8.0μs
F_{OSC} /16	101	1.0μs	2.0μs	4.0μs	16.0μs
F_{OSC} /32	010	2.0μs	4.0μs	8.0μs	32.0μs
F_{OSC} /64	110	4.0μs	8.0μs	16.0μs	64.0μs
F_{RC}	x11	4.0μs	4.0μs	4.0μs	4.0μs

表 15-4 ADC 时钟周期和器件工作频率

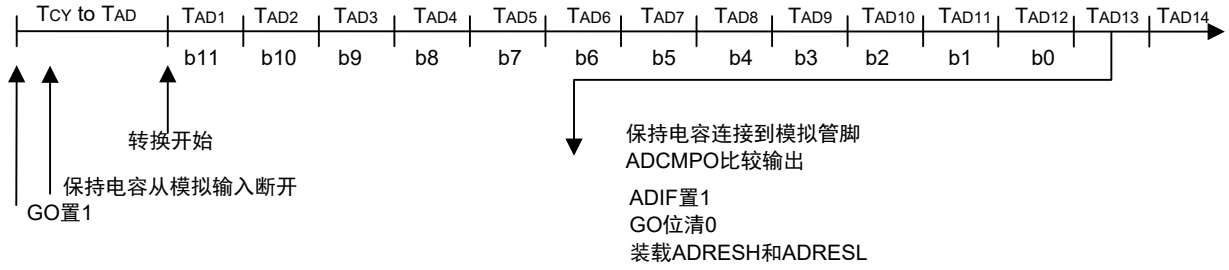


图 15-3 模数转换 T_{AD} 周期

15.2.9. 中断

ADC 模块可使中断在模数转换完成时产生。ADC 中断标志为 PIR1 寄存器中的 ADIF 位。ADC 中断使能为 PIE1 寄存器中的 ADIE 位。ADIF 位必须用软件置 1 清零。

注意：

1、无论 ADC 中断是否被打开，ADIF 位在每次正常转换完成时均置 1。

2、软件停止 AD 转换不会置位 ADIF。

3、仅当置位 SYSON 时，ADC 才能在休眠期间工作。

器件工作或处于休眠状态时均可产生中断。如果器件处于休眠状态，中断可唤醒器件。从休眠唤醒时，始终执行 SLEEP 指令后的那条指令。如果用户试图唤醒器件并恢复顺序执行代码，必须禁止全局中断。如果允许全局中断，代码执行将转至中断服务程序。

15.2.10. 转换结果的格式

12 位 A/D 转换结果有两种格式，即左对齐和右对齐。ADCON1 寄存器的 ADFM 位控制输出格式。
AD 自动校准值也受输出格式影响。

两种输出格式：

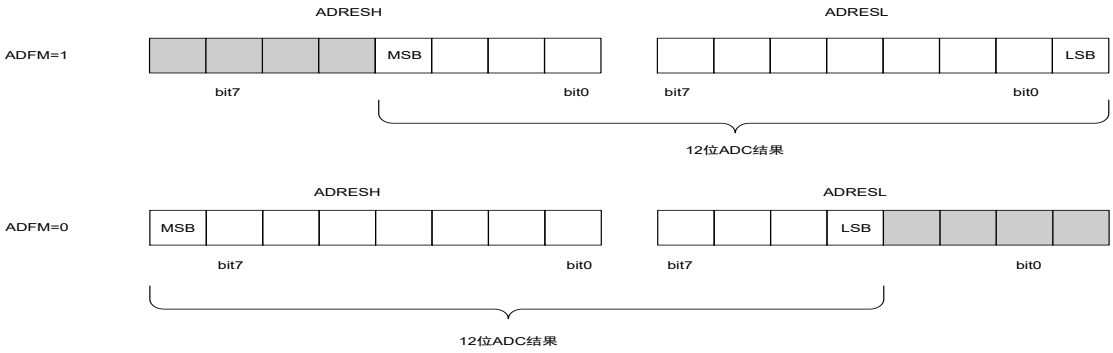


图 15-4 ADC 转换结果格式示意图

15.2.11. 阈值比较

ADCMPH 寄存器为 ADC 结果比较阈值，ADCON3 寄存器的 ADCMPEN 位控制比较功能使能，ADCMPOP 位控制比较极性，ADCMPO 指示比较结果。

可在每次 AD 转换完成时进行比较。比较结果会一直保持，直到下次转换完成时被更新。清零 ADCMPEN 或 ADON 可以关闭比较功能或 AD 模块，且同时清零 ADCMPO。进入睡眠不会清 ADCMPO。在每次比较完成时可以产生故障刹车事件，由 ADCON3 寄存器的 ADFBEN 控制。

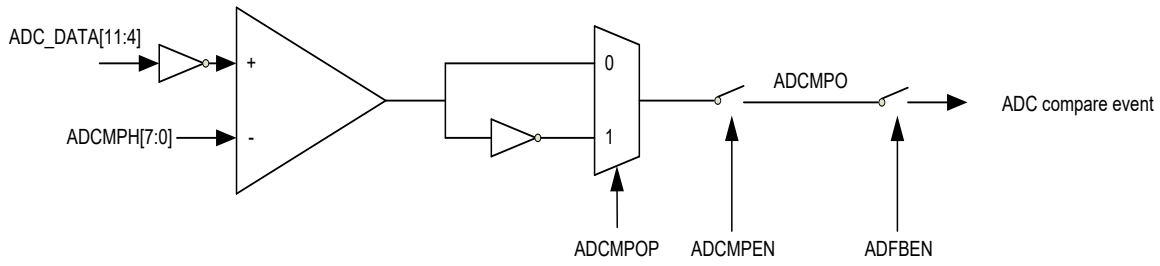


图 15-5 ADC 阈值比较功能框图

由阈值比较框图可知，ADC 结果高 8 位是先做了取反再和 ADCMPH 作比较的，软件在使用时要注意。举例说明：假设要比较 ADC 结果是否大于 Ath，软件需要做以下步骤：

1. 把比较目标值先取反得到~Ath，然后写到 ADCMPH；
2. 把 ADCMPOP 置 1；

15.3. ADC 的工作原理

15.3.1. 启动转换

要使能 ADC 模块，必须将 ADCON0 寄存器的 ADON 位置 1。

若 ADEX=0 时，将 ADCON0 寄存器的 GO/DONE 位置 1 将启动 AD 转换,。

若 ADEX=1 时，由外部触发信号硬件置位 GO/DONE 位，启动 AD 转换。程序置位 GO/DONE 位无效。

注意：

1. 不应在打开 ADC 的那条指令中将 GO/DONE 位置 1。请参见**章节 15.3.6**“A/D 转换步骤”

2. 不应在启动 ADC 转换后或等待外部触发时更改 AD 配置。
3. 置位 GO/DONE 位后需要等待一个系统周期才可读回 GO/DONE 标志位。

15.3.2. 转换完成

转换完成时，ADC 模块将：

- 将 GO/DONE 位清零
- 将 ADIF 标志位置 1
- 用新的转换结果更新 ADRESH:ADRESL 寄存器
- 若使能阈值比较功能，则更新 ADCMPO 比较结果

15.3.3. 终止转换

当 ADEX=0，ADC 处于软件触发状态，如果转换必须在完成前被终止，可用软件将 GO/DONE 清零。那么 ADC 将不会更新这次转换数据。当 ADEX=1，ADC 处于硬件触发状态，如果需要终止转换，则需要将 ADON 置 0，关闭 ADC 的使能开关。

注意：器件复位将强制所有寄存器回到其复位状态。这样，ADC 模块就被关闭，并且任何待处理的转换均被终止。

15.3.4. 休眠模式下 ADC 的工作

ADC 模块可在休眠期间工作，这要求打开 SYSON 位。
ADC 需要等待 $4*T_{AD}$ 后才开始转换。这允许软件在设置 GO/DONE 位后，执行一个 SLEEP 指令置 MCU 于 SLEEP 模式，从而降低 ADC 转换期间的系统噪声。
如果允许 ADC 中断，转换完成后器件将从休眠唤醒。如果禁止 ADC 中断，ADC 模块在转换完成后关闭，尽管 ADON 位保持置 1 状态。
如果 SYSON 未打开，执行一条 SLEEP 指令将使当前转换强制中止，ADC 模块被直接关闭，尽管 ADON 位保持置 1 状态。

如果需要在休眠模式下搭配其它模块一起使用，具体请参见相应的功能配置章节，如 TIMER、GPIO、CLK 管理模块。

15.3.5. 外部触发器

除了通过软件启动 AD 转换外，还可以通过硬件触发方式启动 AD 转换。在 ADEX 置 1 后，可选择 PWM 通道的边沿或周期、管脚边沿等触发信号自动触发启动 AD 转换 (硬件自动置位 GO/DONE)。这允许在没有软件介入的情况下，定期进行 AD 转换。

通过 ETGSEL (ADCON2[2:0]) 和 ETGTYP (ADCON2[5:4]) 设置来选择触发源和触发类型。同时，还可以在外部触发信号与启动 AD 转换之间插入触发延时。

在 AD 模块转换过程中(GO/DONE = 1)，任何软件或硬件触发信号都是无效的。若这时停止转换，清零 ADGO 并不会停止触发延时计数。可清零 ADEX 停止触发延时计数。

只有配置 TIMER 为 PWM 输出模式并且使能 PWM 输出时，才会产生 AD 触发信号。更多信息请参见相应的 TIMER 章节。

注意：
当 LEBEN=1 时，外部触发器被禁止。这种情况下，可以选择把 LEBADT 置 1，此时 ADON 和 ADEX 必须设置为 1。在消隐周期结束后会触发一次 AD 转换 (硬件自动置位 GO/DONE)。

15.3.6. A/D 转换步骤

以下是使用 ADC 进行模数转换的步骤示例：

1. 配置端口：
   - 禁止引脚输出驱动器 (见 TRIS 寄存器)
   - 将引脚配置为模拟
2. 配置 ADC 模块：
   - 选择 ADC 转换时钟
   - 配置参考电压
   - 选择 ADC 输入通道
   - 配置触发源、类型及延时
   - 选择转换结果的格式
   - 配置 ADC 结果阈值比较
   - 打开 ADC 模块
3. 配置 ADC 中断 (可选)：
   - 将 ADC 中断标志清零
   - 允许 ADC 中断
   - 允许外设中断
   - 允许全局中断
4. 等待所需稳定时间 T_{ST}(1)；
5. 将 GO/DONE 位置 1 启动转换或等待硬件触发；
6. 等待一个系统周期才可回读 GO/DONE 位；
7. 通过以下情况之一等待 ADC 转换完成：
   - 查询 GO/DONE 位
   - 等待 ADC 中断 (允许中断时)
8. 读取 ADC 结果；
9. 将 ADC 中断标志清零 (在允许了中断的情况下这一步是必需的)。

以下是一段示例代码：

```
BANKSEL   ADCON1          ;
LDWI      B'11110101'     ;Right justify, ADC Frc clock
STR       ADCON1          ;Vref+ VDD , Vref- GND

BANKSEL   TRISB           ;
BSR       TRISB,7         ;Set PB7 to input

BANKSEL   ANSELA          ;
BSR       ANSELA,0        ;Set PB7 to analog

BANKSEL   ADCON0          ;
LDWI      B'00000001'     ;Select channel AN0,
STR       ADCON0          ;Turn ADC On

CALL      StableTime      ;ADC stable time, tST

BSR       ADCON0,ADGO     ;Start conversion (ADC stable time)
NOP                       ;ADGO ReadBack WaitTime

BTSC      ADCON0,ADGO     ;Is conversion done?
LJUMP     $-1             ;No, test again

BANKSEL   ADRESH          ;
LDR       ADRESH,W        ;Read upper 4 bits
STR       RESULTHI        ;store in GPR space

BANKSEL   ADRESL          ;
LDR       ADRESL,W        ;Read lower 8 bits
STR       RESULTLO        ;Store in GPR space
```

注意：

1. T_{ST} 时间是 ADC 的稳定时间，当使用内部参考时，ADC 首次启动还需要考虑参考电压的稳定时间 T_{VRINT}，等待时间应取两者的较大者，即 max (T_{VRINT}, T_{ST})；

15.4. A/D 采集时间要求

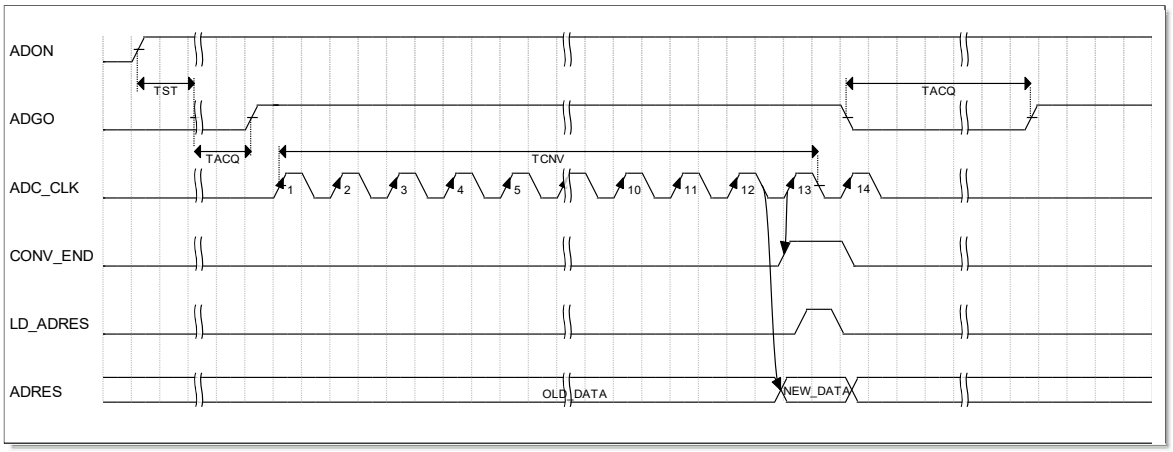
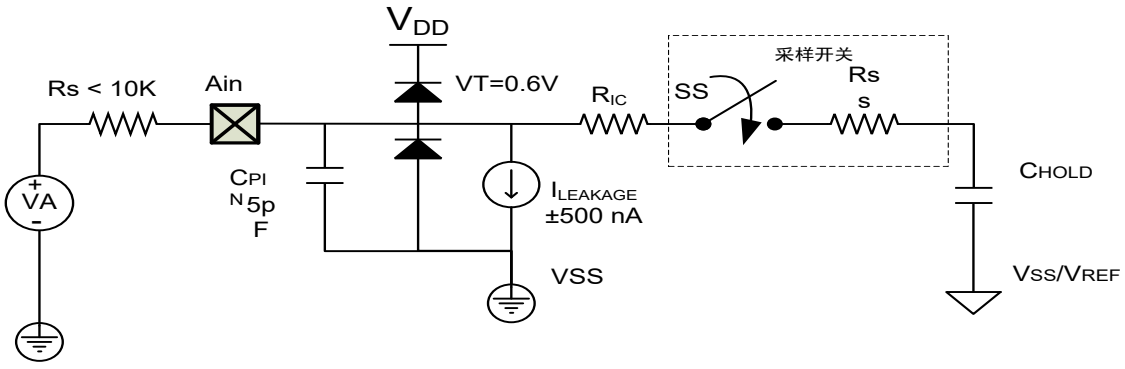


图 15-6 ADC 软件触发转换时序图

为了使 ADC 达到规定的精度，必须使充电保持电容 (CHOLD) 充满至输入通道的电平。模拟输入模型请参见**图 15-7**。源阻抗 (RS) 和内部采样开关 (RSS)阻抗直接影响电容 CHOLD 的充电时间。采样开关 (RSS) 阻抗随器件电压 (VDD) 的变化而变化。建议模拟信号源的最大阻抗为 10kΩ。采集时间随着源阻抗的降低而缩短。在选择 (或改变) 模拟输入通道后，必须在开始转换前完成采集。



图注：

CPIN	= 输入电容
VT	=门限电压
ILEAKAGE	=结点漏电流
RIC	=互联电阻
SS	=采样开关
CHOLD	=采样保持电容

图 15-7 模拟输入模型

16. USART 接口

- 同步模式
  - ✓ 产生同步时钟输出
- 多芯片通信模式
  - ✓ 哑模式唤醒之后，才可以接收数据
  - ✓ 可以通过地址匹配和 IDLE 帧唤醒哑模式
- 异步模式
  - ✓ 可编程的 7,8,9 比特数据模式
  - ✓ 支持 1,2/1.5 bit 停止位
  - ✓ 支持红外 1.0 模式
  - ✓ 单线半双工
  - ✓ 发送接收使能控制
  - ✓ 16bit 波特率设置
  - ✓ URRXNE 中断，URTE 中断，帧断开，帧错误，奇偶校验错误，overrun 中断，IDELE 帧中断，发送完成中断
- 智能卡模式
  - ✓ 1.5 bit 停止位
  - ✓ 时钟输出
  - ✓ guard time
- LIN 主机模式
  - ✓ 支持断开帧的发送与检测
- 自动波特率检测

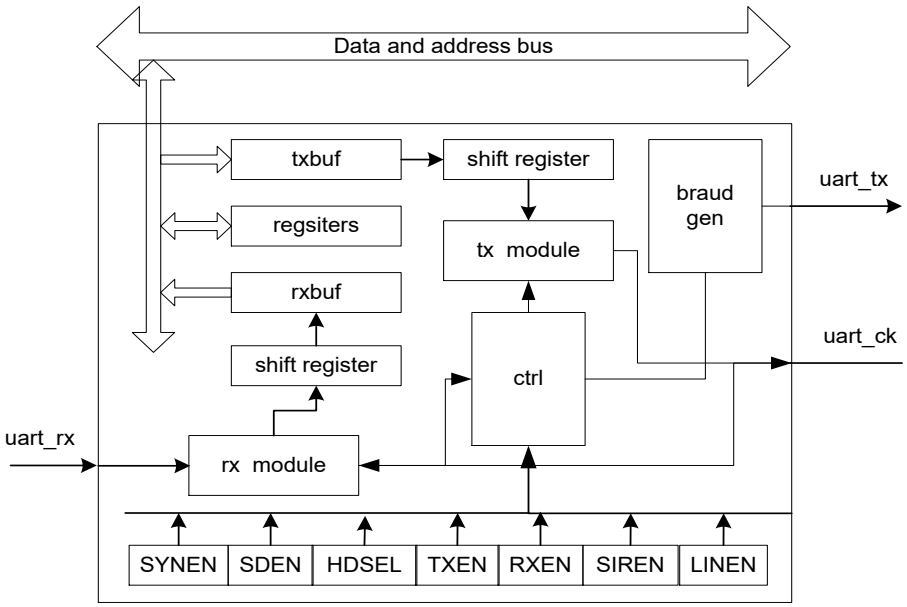


图 16-1 USART 结构框图

16.1. USART 接口相关寄存器汇总

名称	状态		寄存器	地址	复位值
DATAL	数据发送/接收 BUF 低 8 位 (不适宜位操作)		URDATAL[7:0]	0x48C	RW-0000 0000
DATAH	当 EXTEN=1：数据发送/接收 BUF 高 1 位 注：要先写 URxDATAL，再写 URxDATAH 当 EXTEN=0： 1 = URxDATAL 为地址 0 = URxDATAL 为数据		URDATAH[0]	0x48D	RW-0
UARTEN	USART 模块时钟 1 = 打开 0 = 关闭		PCKEN[5]	0x9A	RW-0
SYSON	睡眠模式下，系统时钟控制 1 = 保持运行 0 = 关闭		CKOCON[7]	0x95	RW-0
BKREQ	发送断开帧 1 = 使能，或正在发送中 0 = 关闭，或已发送完成 注：发送完成后该位自动清零，不能在断开帧发送过程中对该位写 0，发送断开帧之前请先设置断开帧的长度		URLCR[6]	0x48F	RW-0
EVEN	奇/偶校验选择	1 = 偶校验 0 = 奇校验	URLCR[4]		RW-0
PEN	校验位	1 = 使能 0 = 关闭	URLCR[3]		RW-0
STOP	停止位长度 1 = 1.5 bit (智能卡模式) 或 2 bit 0 = 1 bit		URLCR[2]		RW-0
LTH	通信数据长度 (不包括校验位) 1 = 8 bit 0 = 7 bit		URLCR[0]		RW-0
RWU	多处理器模式下，接收唤醒进入哑模式 1 = 使能 0 = 关闭，或已退出		URLCREXT[1]	0x490	RW-0
EXTEN	发送数据长度 1 = 9 bit 0 = 7 bit 或 8 bit (由 LTH 决定)		URLCREXT[0]		RW-0

名称	状态		寄存器	地址	复位值
SIRLP	红外低功耗模式	1 = 使能 0 = 关闭	URMCR[5]	0x491	RW-0
TXEN	串口发送 1 = 使能 (相应 IO 会被用作 TX 引脚) 0 = 关闭		URMCR[4]		RW-0
RXEN	串口接收使能 1 = 使能 (相应 IO 会被用作 RX 引脚) 0 = 关闭		URMCR[3]		RW-0
WAKE	哑模式唤醒方式	1 = 地址匹配 0 = IDLE 帧	URMCR[2]		RW-0
HDSEL	半双工	1 = 使能 0 = 关闭	URMCR[1]		RW-0
SIREN	红外模式	1 = 使能 0 = 关闭	URMCR[0]		RW-0
RAR	多处理器模式下的本机地址[3:0]		URRAR[3:0]	0x493	RW-0000
DLL	波特率分频计数器低 8 位和高 8 位 波特率 = Fmaster / (16 * {DLH, DLL}) 注：Fmaster = Sysclk。当{DLH, DLL} = 0x0000 时，USART 不工作；{DLH, DLL} 最小值= 0x0001		URDLL[7:0]	0x494	RW-0000 0000
DLH			URDLH[7:0]	0x495	RW-0000 0000
ABRE	波特率检测溢出标志	1 = 溢出 0 = 正常	URABCR[3]	0x496	RW-0
ABRM	波特率检测模式 1 = 检测 [(起始位+第 1bit 数据) / 2] 的长度 (数据的第 1bit 必须为 1，第 2bit 必须为 0) 0 = 只检测起始位长度 (第 1bit 数据必须为 1)		URABCR[2]		RW-0
ABRF	检测到波特率标志位 1 = 检测到 0 = 未检测到 注：写 0 清零，该位在清零后，会立即再次进入波特率检测，为了保证每次检测到的都是起始位，建议在 RXNEF 被置位后，再清零此位		URABCR[1]		RW-0
ABREN	自动波特率检测	1 = 使能 0 = 关闭	URABCR[0]		RW-0
LBCL	同步模式下，发送最后 1bit 数据(MSB)对应的时钟输出 1 = 使能 0 = 关闭		URSYNCR[3]	0x497	RW-0

名称	状态	寄存器	地址	复位值
URCPHA	同步模式时钟相位 (第一个数据采样点) 1 = 第 2 个时钟转换沿 0 = 第 1 个时钟转换沿	URSYNCR[2]		RW-0
URCPOL	同步模式时钟极性 (总线空闲时，SCK 的状态) 1 = 高电平 0 = 低电平	URSYNCR[1]		RW-0
SYNEN	同步模式 1 = 使能 (相应 IO 会被用作同步时钟输出) 0 = 关闭	URSYNCR[0]		RW-0
LINEN	LIN Master 模式：1 = 使能；0 = 关闭	URLINCR[4]	0x498	RW-0
BLTH	断开帧长度 (bit)设置 注: BLTH>0 有效，建议设置为 12 或 13bit 长度，太短会被认为接收到的为正常帧	URLINCR[3:0]		RW-0000
NACK	智能卡模式，检测到奇偶校验出错时回复 NACK 1 = 发送 NACK 0 = 不发送 NACK	URSDCR0[6]	0x499	RW-0
CKOE	智能卡时钟源输出 1 = 使能 (需配置 PSC 寄存器为有效值) 0 = 关闭	URSDCR0[5]		RW-0
SDEN	智能卡模式 1 = 使能 (停止位必须为 1.5bit) 0 = 关闭	URSDCR0[4]		RW-0
GT	智能卡模式，保护时间 (两字符之间的波特时钟间隔) 注：最小值为 1 (当 GT=0，两个字符之间也有 1 个波特周期的间隔)，保护时间过后，发送完成标志被置位	URSDCR1[7:0]	0x49A	RW-0000 0000
PSC	对系统时钟进行分频，给智能卡或红外低功耗提供时钟	URSDCR2[7:0]	0x49B	RW-0000 0000

	智能卡时钟源	红外低功耗时钟源
0	无效	无效
1	2 分频	1 分频
2	3 分频	2 分频
3	4 分频	3 分频
…	…	…
255	256 分频	255 分频

表 16-1 USART 相关寄存器

名称	状态		寄存器	地址	复位值
GIE	全局中断	1 = 使能 (PEIE, URTE, URRXNE, TCEN, IDELE, RXSE 适用) 0 = 全局关闭 (唤醒不受影响)	INTCON[7]	Bank 首地址 +0x0B	RW-0
PEIE	外设总中断	1 = 使能 (URTE, URRXNE, TCEN, IDELE, RXSE 适用) 0 = 关闭 (无唤醒)	INTCON[6]		RW-0
URTE	发送 BUF 为空中断	1 = 使能 0 = 关闭	URIER[1]	0x48E	RW-0
TXEF [1]	发送 BUF 状态	1 = 空 0 = 非空	URLSR[5]	0x492	RO-1
URRXNE	接收 BUF 非空中断	1 = 使能 0 = 关闭	URIER[0]	0x48E	RW-0
RXNEF [1]	接收 BUF 状态	1 = 非空 0 = 空，或已被清零	URLSR[0]	0x492	RO-0
TCEN	发送完成中断	1 = 使能 0 = 关闭	URIER[5]	0x48E	RW-0
TCF	发送完成标志	1 = 完成 0 = 未完成 注：写 1 清零，或写 DATAL(8bit) /DATAH(9bit)后清零	URTC[0]	0x49C	R_W1C-1
IDELE	空闲帧中断	1 = 使能 0 = 关闭	URIER[3]	0x48E	RW-0
IDLEF [2]	检测到空闲帧标志	1 = 检测到 0 = 未检测到	URLSR[6]	0x492	RW0-0
RXSE	接收状态中断	1 = 使能 0 = 关闭 接收状态中断产生条件： BKF = 1 FEF = 1 PEF = 1 OVERF = 1	URIER[2]	0x48E	RW-0
BKF [2]	接收到断开帧标志	1 = 接收到 0 = 未接收到，或已被清零	URLSR[4]	0x492	RW0-0
FEF [2]	接收帧错误标志	1 = 错误 0 = 正确，或已被清零	URLSR[3]		RW0-0
PEF [2]	接收奇偶校验错误标志	1 = 错误 0 = 正确，或已被清零	URLSR[2]		RW0-0
OVERF [2]	接收 BUF 溢出标志	1 = 溢出 0 = 正常，或已被清零	URLSR[1]		RW0-0

[1] 写 DATAL(8bit) / DATAH(9bit) 清零
[2] 写 0 清零，写 1 无效。

名称	状态		寄存器	地址	复位值
WAKE	哑模式唤醒方式选择	1 = 地址匹配 0 = IDLE 帧	URMCR[2]	0x491	RW-0
ADDRF	哑模式地址匹配标志	1 = 匹配 0 = 未匹配	URLSR[7]	0x492	RO-0

表 16-2 USART 中断使能和状态位

名称	状态		寄存器	地址	复位值
AFP1[5:4]	USART_TX	00 = PA6 10 = PA7 01 = PB7 11 = PA2	AFP1[5:4]	0x19F	RW-00
AFP1[3:2]	USART_RX	00 = PA7 10 = PA6 01 = PA2 11 = PB6	AFP1[3:2]		RW-00
UROD	USART_TX 开漏输出	1 = 使能 0 = 关闭	ODCON0[0]	0x21F	RW-0

表 16-3 USART 接口引脚控制

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
PCKEN	0x9A	—	—	UARTEN	—	TIM4EN	TIM2EN	TIM1EN	ADCEN	0000 0000
CKOCON	0x95	SYSON	CCORDY	DTYSEL		CCOSEL[2:0]			CCOEN	0010 0000
URDATAL	0x48C	DATAL[7:0]								0000 0000
URDATAH	0x48D	—							DATAH	---- ---0
URIER	0x48E	—		TCEN	—	IDELE	RXSE	URTE	URRXNE	--0- 0000
URLCR	0x48F	—	BKREQ	—	EVEN	PEN	STOP	—	LTH	-0-0 00-0
URLCREXT	0x490	—						RWU	EXTEN	---- --00
URMCR	0x491	—		SIRLP	TXEN	RXEN	WAKE	HDSEL	SIREN	--00 0000
URLSR	0x492	ADDRF	IDLEF	TXEF	BKF	FEF	PEF	OVERF	RXNEF	0000 0000
URRAR	0x493	—				RAR[3:0]				---- 0000
URDLL	0x494	DLL[7:0]								0000 0000
URDLH	0x495	DLH[7:0]								0000 0000
URABCR	0x496	—				ABRE	ABRM	ABRF	ABREN	---- 0000
URSYNCR	0x497	—				LBCL	CPHA	CPOL	SYNEN	---- 0000
URLINCR	0x498	—			LINEN	BLTH[3:0]				---0 0000
URSDCR0	0x499	—	NACK	CKOE	SDEN	—				-000 ----
URSDCR1	0x49A	GT[7:0]								0000 0000
URSDCR2	0x49B	PSC[7:0]								0000 0000
URTC	0x49C	—							TCF	---- ---1

表 16-4 USART 相关寄存器地址

16.2. 功能描述

16.2.1. 一般描述

串口模块总共有三只引脚：

1. USART_RX：用作串口数据的输入引脚
2. USART_TX：用作串口数据的输出引脚，在用作半双工模式时也用作串行数据的输入引脚
3. USART_CK：在同步模式时用作同步时钟输出，在智能卡模式时用系统作分频时钟输出

该模块支持同步模式，异步模式，半双工模式，LIN Master 模式，红外模式和智能卡模式，默认的状态是工作于异步全双工模式，在确定使用一种模式后，请确保其他模式的相关使能位已关闭。

16.2.2. 异步工作模式

异步工作模式的串口采用异步的方式进行通信，配置的步骤如下：

1. 配置 DLH/DLL 产生相应的波特率进行通信，DLH 和 DLL 共同组成 16 位的波特率分频器，通信的波特率=fmaster/(16*(DL*))，其中 fmaster 为系统时钟，16 位的波特率分频器的值最小位 1，DL*表示的是 DLL 和 DLH 的组合，设置为零时串口不工作；
2. 配置 LCR 寄存器中的 LTH 位和 LCREXT 寄存器中的 EXTEN 来设置通信的数据长度，配置 LCR 寄存器中的 STOP 位来配置停止位的长度，配置 LCR 寄存器中的 PEN 和 EVEN 来配置奇偶校验位，配置 IER 寄存器中的中断使能位来允许中断；
3. 配置 MCR 寄存器中的 TXEN 和 RXEN 来使能允许发送和接收；

异步模式通信的数据格式是先发送低位数据位，最后发送高比特位，如下图所示的 8 比特数据格式不带奇偶校验和带奇偶校验的帧格式。

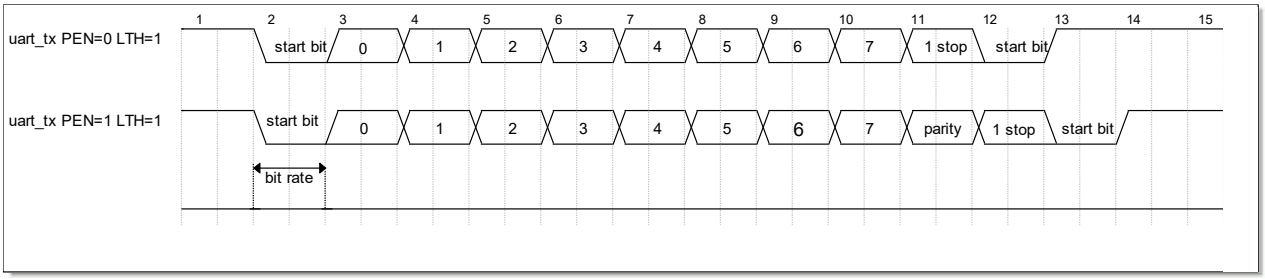


图 16-2 异步模式时序图

异步模式的数据处理流程包括阻塞模式处理和非阻塞模式处理，大致的处理流程如下：

1. 配置完波特率和相关控制位后，发送端可以向 DATAL/H 发送 buf 寄存器写入数据，在阻塞模式下可以查询 TXEF 标志位，如果查询到 TXEF 为 1，则可以继续向 DATAL/H 写入需要发送的数据；在非阻塞模式下，使能发送为空中断，则在 TXEF 为 1 时，就会自动进入中断，向 DATAL/H 写入数据就可清除 TXEF 标志位，当在向 txbuf 写入最后一个要发送的数据时，禁用发送为空中断；
2. 接收端在阻塞模式下可以查询 RXNEF 标志位，在查询到该标志位为 1 时，表示接收到了数据，通过读取 DATAL/H 来清零 RXNEF 标志位；在阻塞模式接收数据时，需要使能 RXNE 中断，在串口接收到数据后，直接进入中断，读取 rxbuf 后清零 RXNEF 标志位；在非阻塞模式接收数据时，建议打开 RXSE 中断使能，在接收数据的过程中如遇到接收错误就会直接进入中断进行相关的处理；

3. 在串口发送数据的时候也可以使用 TCF 标志位来处理，在 TCF 标志位为 1 时，表示当前的数据发送已经完成，可以向 txbuf 写入下一个要发送的数据，这时 TCF 标志位会自动清零；

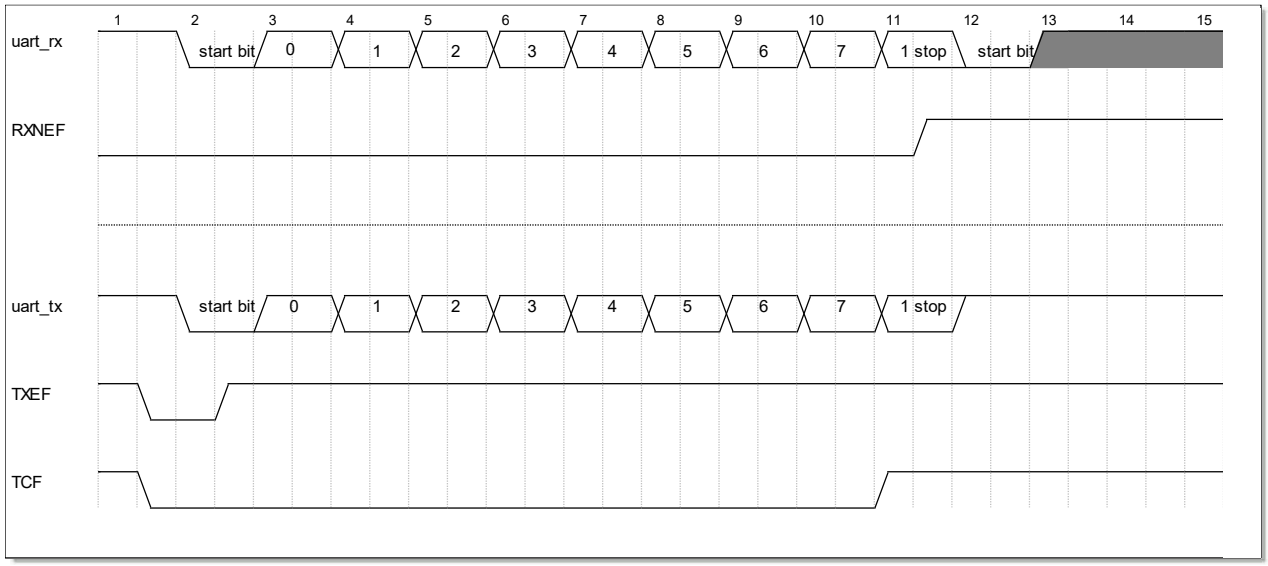


图 16-3 异步模式标志位时序图

16.2.3. 同步工作模式

同步工作模式用于串口模拟 SPI 通信的功能，在串口数据输出的同时，输出一个与数据相关的同步时钟，同步时钟的极性和相位可以通过 URSYNCR 寄存器中的 CPOL 和 CPHA 来配置；URSYNCR 寄存器中的 LBCL 控制的是最后一比特数据的时钟是否输出，该位为零时，只输出数据长度减一个有效同步时钟，最后一个有效时钟不会输出；SYNEN 就是同步时钟输出使能位，在该位为 1 时相应的 IO 会用作同步时钟输出；该模块只能模拟 SPI 主机模式，数据输出是先发送低位数据，然后发送的高位数据，并且时钟引脚只能输出同步时钟，并不能用作时钟输入；

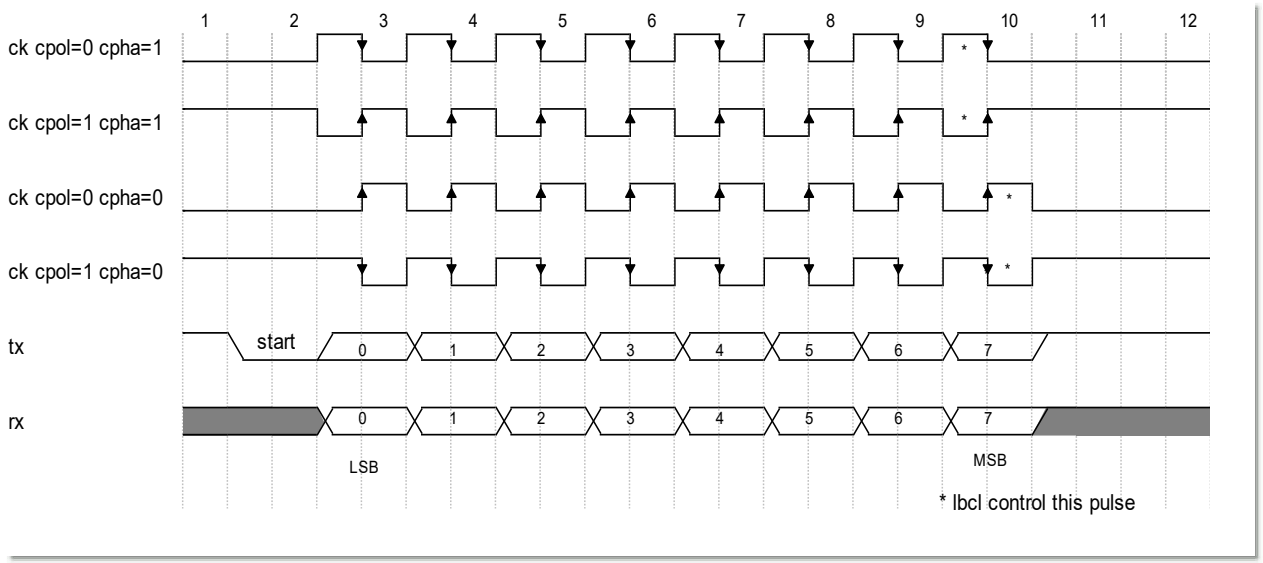


图 16-4 同步模式时序图

如图所示 8bit 数据格式的同步时钟输出，同步模式中如果没有使能 TXEN 也会产生同步脉冲输出，这时候同步模式只用于接收数据，写入到 DATAL/H 寄存器中的数据会发送到内部的移位寄存器中，用于产生同步脉冲输出，tx 引脚的值一直保持为 1，在同步发送的同时如果使能了 RXEN 接收位，则可以同步接收数据。

16.2.4. 半双工模式

半双工模式属于异步工作方式的一种，只是在通信时只用到了 tx 引脚，tx 引脚 IO 应该配置成开漏模式，发送与接收的处理由软件控制 RXEN 和 TXEN 来实现；需要注意的是如果在发送过程中同时使能了接收，则发送的数据也会被本机接收到；置位 HDSEL 即可启用半双工模式。

16.2.5. 红外工作模式

红外模式用于红外通信，置位 SIREN 位可以使能红外模式，同时 LTH 位置位为 1，启用八比特数据格式；通信的波特率设置跟异步串口的配置方法相同。

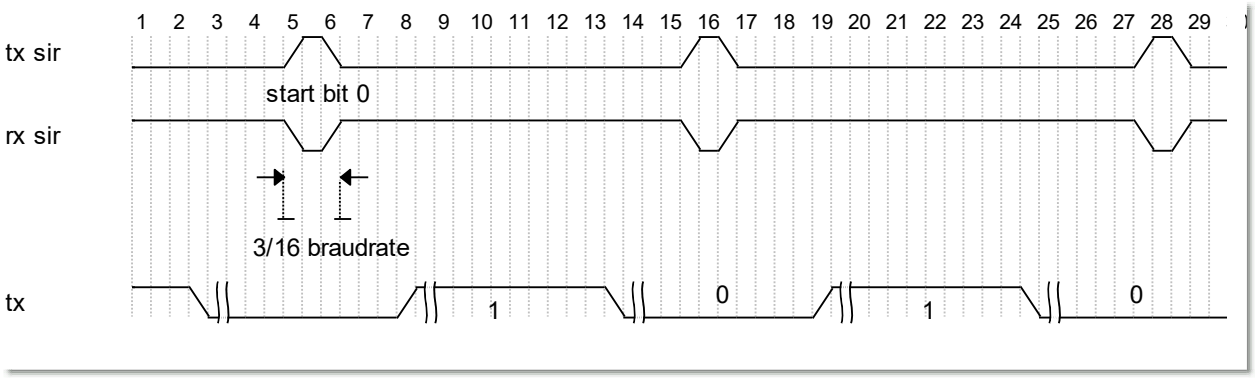


图 16-5 红外模式时序图

如图所示红外模块发送的脉冲宽度是比特周期的 3/16，当发送的数据为零时会产生一个高脉冲；接收时的低脉冲会被解释成零；接收与发送的总线极性是相反的，发送空闲时总线保持低电平，接收空闲时总线保持为高电平。

红外模式可以工作在低功耗模式，红外模式通常工作在系统的时钟频率下，红外的通信波特率=fmaster/(16*DL*)；当使能了 SIRLP 以后，红外的通信波特率= fmaster/(PSC*16*DL*)；这里的 DL*表示 DLL 和 DLH 的组合，在 psc 设置为 0 或 1 时，psc 分频模块无效，波特产生模块直接使用 Fmaster，如下图所示

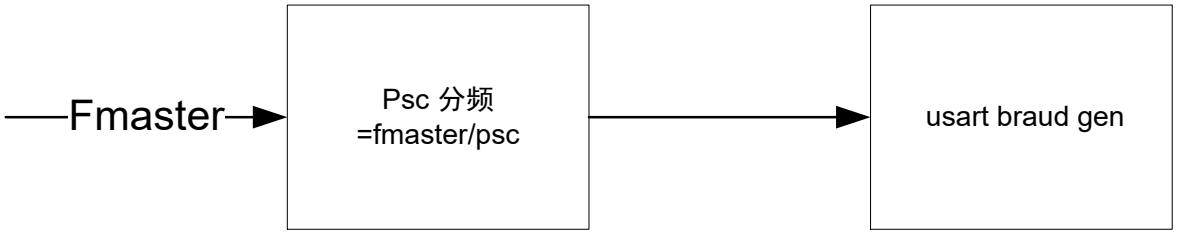


图 16-6 红外低功耗模式原理框图

16.2.6. 智能卡模式

智能卡模式属于半双工模式，支持 ISO7816-3 标准，置位 SDEN 来启用智能卡模式，除此之外根据协议要求需要使能 1.5 比特停止位 STOP 和奇偶校验位 PEN，同时需要配置相应的 IO 为开漏模式。

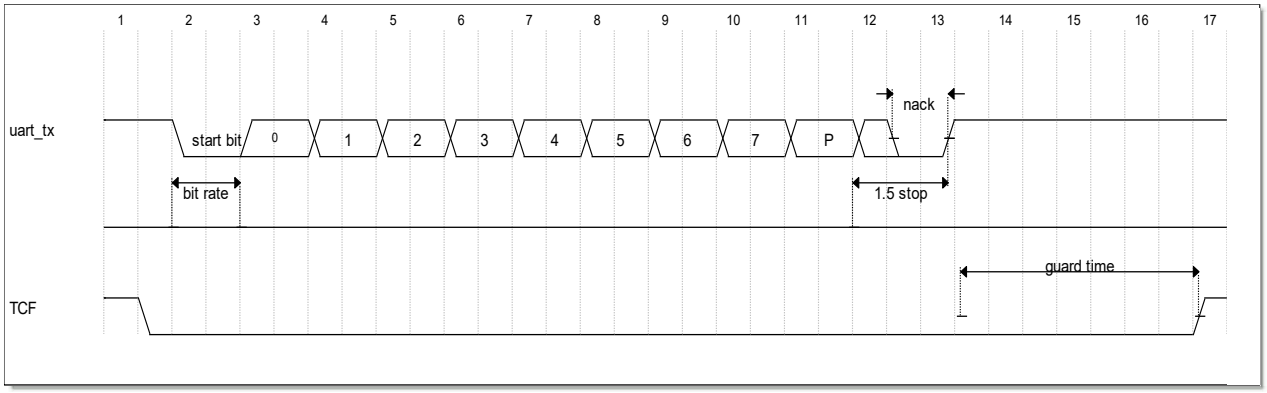


图 16-7 智能卡模式时序图

在使能了 NACK 位以后，接收方在检测到奇偶校验出错以后，会在 0.5 个停止位之后拉低总线一个比特周期，同时发送方会在停止位处检测总线是否被拉低，若检测到总线被拉低，则会产生帧错误标志 FEF，发送方根据要求可以选择重发当前的数据，发送次数由用户决定。在没有使能 NACK 位时，接收方在检测到奇偶校验错误时，不会拉低总线而是会产生一个奇偶检验错误标志位 PEF。

智能卡模式的发送方发送完成数据后，TCF 标志位会在经过 GT 个波特周期后置位；发送与接收的处理由软件控制 TXEN 和 RXEN 来处理。

智能卡模式中，可以通过使能 CKOE 来输出一个时钟供给使能卡使用，输出的时钟频率详见 URSDCR2 寄存器说明；需要注意的是在置位 CKOE 后，请配置 PSC 的值为有效值，否则不应该置位 CKOE。

16.2.7. LIN Master 模式

串口模块支持 LIN Master 模式，使能 LINEN 后进入 LIN Master 模式，在发送断开帧之前请先配置一下断开帧的长度 BLTH；如图所示，在置位 BKREQ 后，tx 引脚会发送 BLTH 个连续的低电平，发送完成后自动清零，在使能该位后，可以查询 BKREQ 的状态，等到 BKREQ 为 0 时则表示断开帧发送完成；在发送断开帧的过程中请勿手动清零 BKREQ。

接收端在接收到大于起始位+数据长度+停止位个数的连续低电平以后，会被认为接收到了断开帧，BKF 会被置 1。

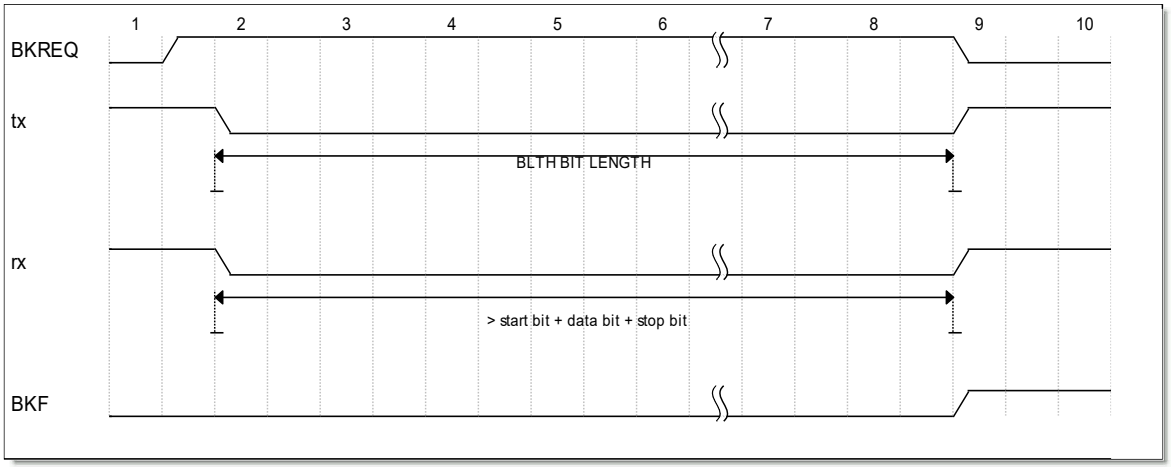


图 16-8 LIN Master 模式时序图

需要注意的是断开帧的接收与发送并不局限于 LIN mode，其他异步模式，红外模式等也是可以应用。

16.2.8. 多芯片通信模式

多处理器通信用于一个芯片用作主机模式，其他芯片用作从机模式，从机的发送引脚通过逻辑与的方式连接到主机的 RX 引脚，这种模式中主机希望接收特定的消息，只有在特定的条件触发后才会接收数据。

置位 RWU 后即可进入哑模式，屏蔽一切接收，根据 WAKE 的配置，可以唤醒接收主机接收数据：

1. WAKE 置 0，在接收到起始位+数据位+停止位总个数波特周期后唤醒；
2. WAKE 置 1，在接收到匹配的地址后唤醒；

- 地址空闲唤醒，在置位 RWU 后，如果总线数据一致繁忙，则不唤醒接收，在检测到连续的一帧空闲时间 (起始位+数据位+停止位) 后唤醒开始接收数据。

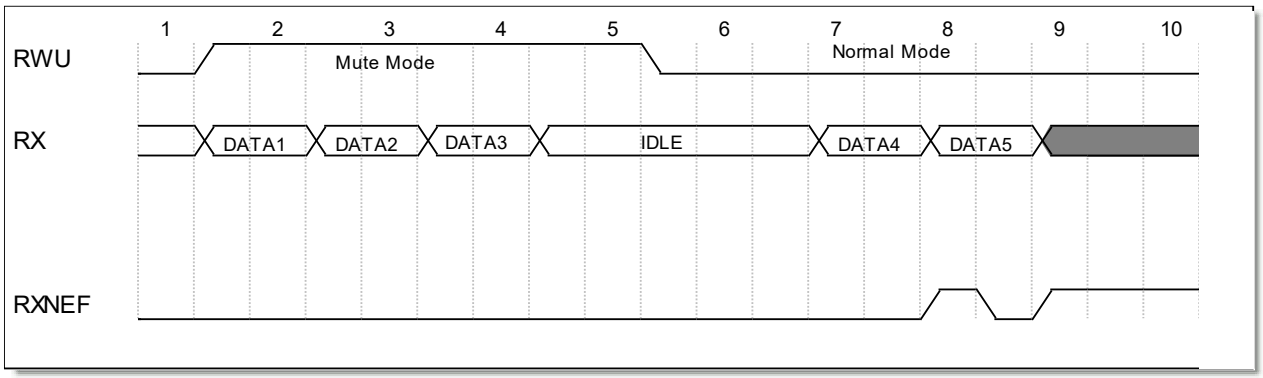


图 16-9 哑模式空闲唤醒时序图

- 地址匹配唤醒，在置位 RWU 后，每次接收到数据后都会判断数据的高位是否为 1，若为 1 则把数据的低四位与 URRAR 的值进行比较，若相等则退出哑模式开始接收之后的数据，后续如果再次接收到地址数据 (该模式下数据的高位为 1 则表示接收到的数据为地址数据)，则还是会与本机的地址 URRAR 进行比较，若不同立即进入哑模式；该模式下每次匹配到地址后 ADDRF 会被置 1，反之没有匹配到地址时一直为零。

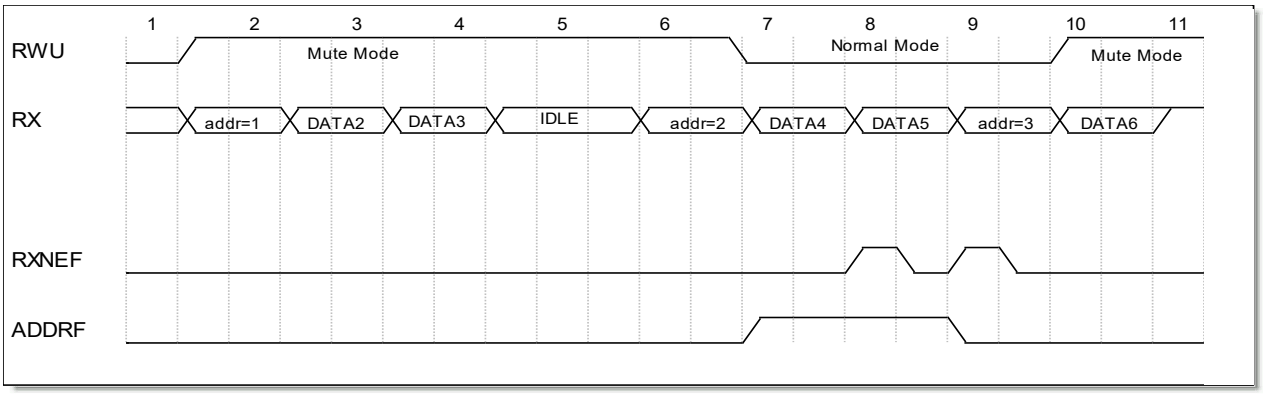


图 16-10 哑模式地址匹配唤醒时序图

16.2.9. 自动波特率检测

自动波特率检测功能用于接收端校准通信波特率，保持与发送端波特率相同，串口模块实现的波特率检测模块有两种模式：

1. 检测起始位的长度 (model0)；这种模式要求数据的第一比特为 1，例如数据 0x03、0x55 等；
2. 检测起始比特和第一比特的长度 (model1)；这种模式要求第一比特的数据为 1，第二比特的数据为 0，例如数据 0x55，0x01 等；

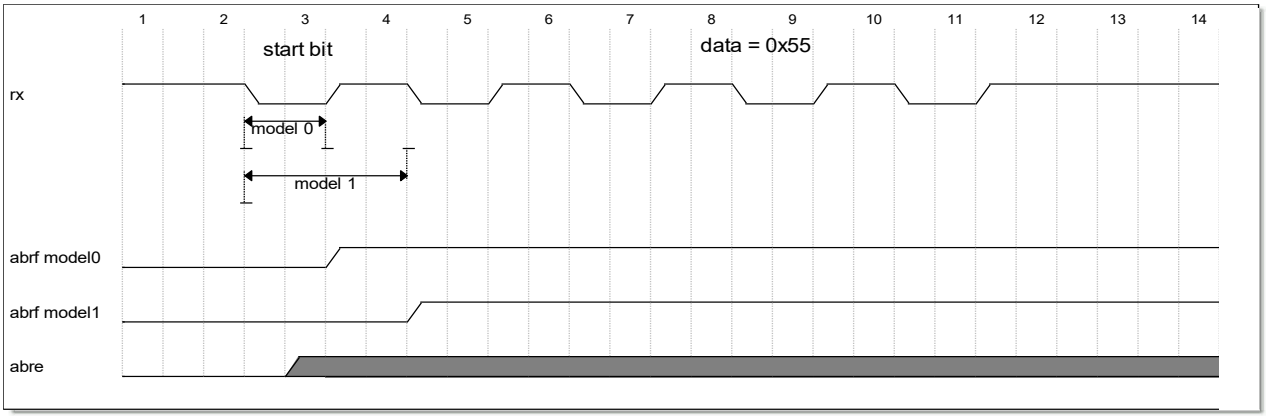


图 16-11自动波特率检测时序图

使用波特率检测功能，首先使能 ABREN，然后根据自己使用的检测模式配置 ABRM，读取 ABRF 是否为 1 (上次使用过后未清零)，如果为 1，则写零清零；然后开始接收数据，波特率检测完成后 ABRF 就会置 1，在 ABRF 置 1 后，不要立即清零 ABRF，因为清理 ABRF 会立即在当前传输的位置 (可能已经不是起始比特的位置) 进行波特率检测，这样会导致错误的结果；当前数据接收完成后会产生 RXNEF 标志位，然后可以清零 ABRF，开始下一次波特率的检测，假如不清零 ABRF 标志位，则下次接收数据时不会启动波特率检测；如果波特率检测超出了范围则会产生 ABRE 标志位 ，表示波特率检测出错。

波特率检测完成后，如果后续还需要检测波特率则不需要立即清零 ABRF，只有在需要再次检测的时候清零 ABRF 即可。

需要注意的是波特率检测的数据是用来配置 DLL/DLH 寄存器用的，如果发送方波特率数据不靠近 Fbraudrate=Fmaster/(16*{DLH,DLL})，则波特率检测模块会自动配置本机为靠近支持的波特率，串口模块并不支持小数波特率，因此该模块的波特率检测存在误差。

17. 存储区读/写保护

程序区 (PROM) 可配置为全区读保护，或扇区读/写保护 (每个扇区空间为 1 k x 14)。这些保护功能均由 IDE 界面进行选择配置。

名称	功能	默认
CPB	PROM 全区读保护	关闭
FSECPB0	PROM 扇区 0(1k x 14)读/写保护	关闭
FSECPB1	PROM 扇区 1(1k x 14)读/写保护	关闭
FSECPB2	PROM 扇区 2(1k x 14)读/写保护	关闭
FSECPB3	PROM 扇区 3(1k x 14)读/写保护	关闭

图 17-1 存储区读/写保护初始化配置寄存器

18. 程序存储器

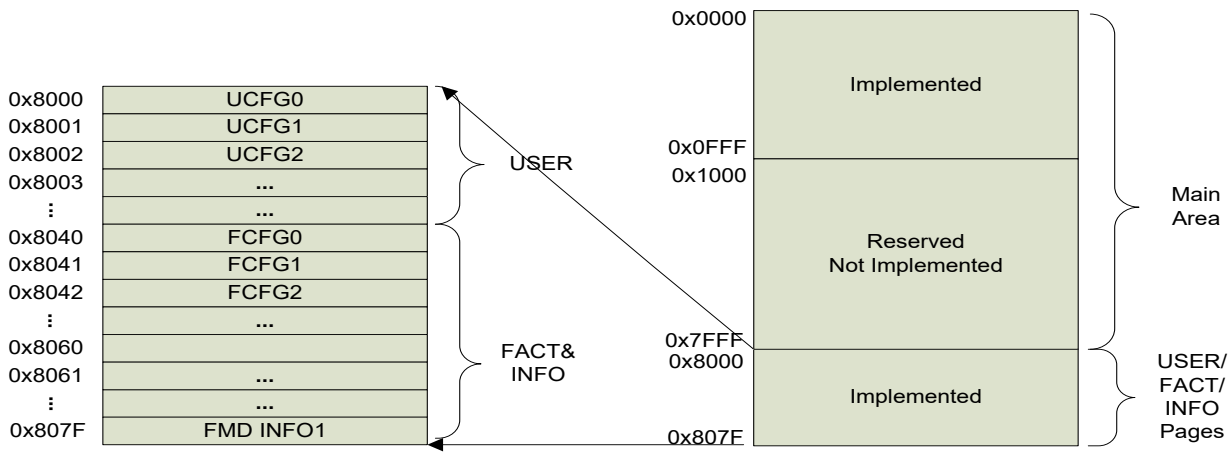


图 18-1 程序存储区地址空间

程序地址计数器 PC 为 15 位(0x0000 ~ 0x7FFF)，最多支持 32K 地址空间。芯片实现了 4K 的程序存储器 (0x0000 ~ 0x0FFF)，外加上 2 个额外的用户配置区，工厂配置区，以及 2 个信息区 INFO0/INFO1。

4K 程序存储器由 128 页组成，每页 32 个 word (1word= 14bits)，地址范围为 0x0000~0xFFF，地址超过 0xFFF 将导致回卷到 0x0000。

另外的 4 个 NVM 区分别占一个单独的页，每页 32 word，其编址从 0x8000 开始，到 0x807F 结束。

将程序存储器当作数据存储器读取

有两种方法可访问程序存储器中的常数。第一种方法是使用 RETW 指令表。第二种方法是设置 FSR 指向程序存储器。

RETW 指令

RETW 指令用于提供对常数表的访问。

如下示例：给出了创建这种表的推荐方法。

```
Constants ; example 18.1
BRW ;     Add Index in W to program counter to ;select data
RETW      DATA0 ;Index0 data
RETW      DATA1 ;Index1 data
RETW      DATA2
RETW      DATA3
my_function：
;… LOTS OF CODE…
LDWI      DATA_INDEX
call      constants      ; THE CONSTANT IS IN W
```

BRW 指令使得这种类型的表实现起来非常简单。如果代码必须保持与前几代单片机之间的可移植性，则 BRW 指令不适用，因此必须使用较老的表读取方法。

使用 FSR 间接读取

可将程序存储器当作数据存储器进行访问，方法是将 FSRxH 寄存器的 bit7 置 1 并读取与之配对的 INDFx 寄存器。MOVIW 指令会将已寻址到的字的低 8 位保存在 W 寄存器中。无法通过 INDF 寄存器执行写程序存储器操作。通过 FSR 访问程序存储器的指令需要一个额外的指令周期才能完成。如下示例演示了通过 FSR 访问程序存储器的过程。

如果标号指向程序存储器中的单元，HIGH 伪指令将 bit[7]置 1。

```
constants    ; example 18.2
RETW     DATA0 ; Index0 data
RETW     DATA1 ; Index1 data
RETW     DATA2
RETW     DATA3
my_function：
;… LOTS OF CODE…
LDWI     LOW constants
STR      FSR1L
LDWI     HIGH constants
STR      FSR1H
MOVIW    0[FSR1]          ; THE PROGRAM MEMORY IS IN W
```

19. 特殊功能寄存器(SPECIAL FUNCTION REGISTERS, SFR)

有 2 种特殊功能寄存器 (SFR)：

- 初始化配置寄存器： 由仿真器界面设置 (Integrated Development Environment, IDE)；
- 用户寄存器；

19.1. 初始化配置寄存器

图 19-1 由 IDE 设置的初始化配置寄存器

名称	功能	默认
CPB	PROM 全区读保护	关闭
MCLRE	外部 I/O 复位	关闭
PWRTEB	上电延时定时器(PWRT)，初始化配置完成后额外延时~64ms	关闭
WDTE	WDT • 使能 (指令不能禁止) • 由指令控制 (WDTCON 的 SWDTEN)	SWDTEN 控制
FOSC	• LP：PC1 (+) 和 PB7 (−) 接外部低速晶振 • XT：PC1 (+) 和 PB7 (−) 接外部高速晶振 • EC：PC1 (+) 接外部时钟输入，PB7 为 I/O • INTOSCIO：PC1 和 PB7 为 I/O	INTOSCIO
STVREN	堆栈上/下溢出复位使能 • 使能 • 关闭	使能
OSTPER	OST 定时器周期选择 (XT / LP 适用) • 512 • 1024 • 2048 • 4096 (LP 模式时为 32768)	1024
TSEL	指令时钟与系统时钟 SysClk 的对应关系 (1T, 2T or 4T)： • 1 (指令时钟= SysClk) • 2 (指令时钟= SysClk/2) • 4 (指令时钟= SysClk/4)	2
FCMEN	故障保护时钟监控器 • 使能 • 关闭	使能
IESO	XT / LP 双速时钟启动 • 使能 • 关闭	使能
LVREN	LVR • 使能 • 关闭 • 非 SLEEP 模式下使能 • 通过指令控制 (SLVREN)	关闭
LVRS	7 档 V_{BOR} 电压(V)： 2.0 / 2.2 / 2.5 / 2.8 / 3.1 / 3.6 / 4.1	2.0

名称	功能	默认
FSECPB0	PROM 扇区 0(1k x 14)读/写保护	关闭
FSECPB1	PROM 扇区 1(1k x 14)读/写保护	关闭
FSECPB2	PROM 扇区 2(1k x 14)读/写保护	关闭
FSECPB3	PROM 扇区 3(1k x 14)读/写保护	关闭

图 19-2 初始化配置寄存器 (由 IDE 设置)

19.2. 用户寄存器

用户寄存器，即特殊功能寄存器 (SFR) 分布在 32 个 bank 中，每个 bank 大小为 128 个字节。在访问寄存器前，必须先切换到相应的 bank。

可通过将存储区号写入存储区选择寄存器 (Bank Select Register，BSREG) 来选择有效存储区。未实现的存储器将读为 0。所有的数据存储器都可以直接访问 (通过使用文件寄存器的指令)，或通过 2 个文件选择寄存器 (FSR) 间接访问。更多信息，请参见**章节 19.5**“间接寻址”。

数据存储器使用一个 12 位地址。地址的高 5 位用于定义存储区地址，低 7 位用于选择该存储区中的寄存器/RAM。

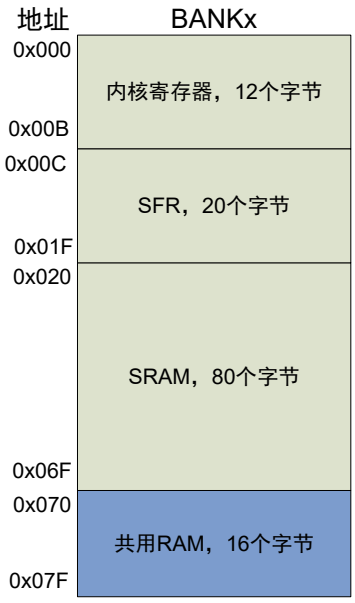


图 19-3 存储区构成

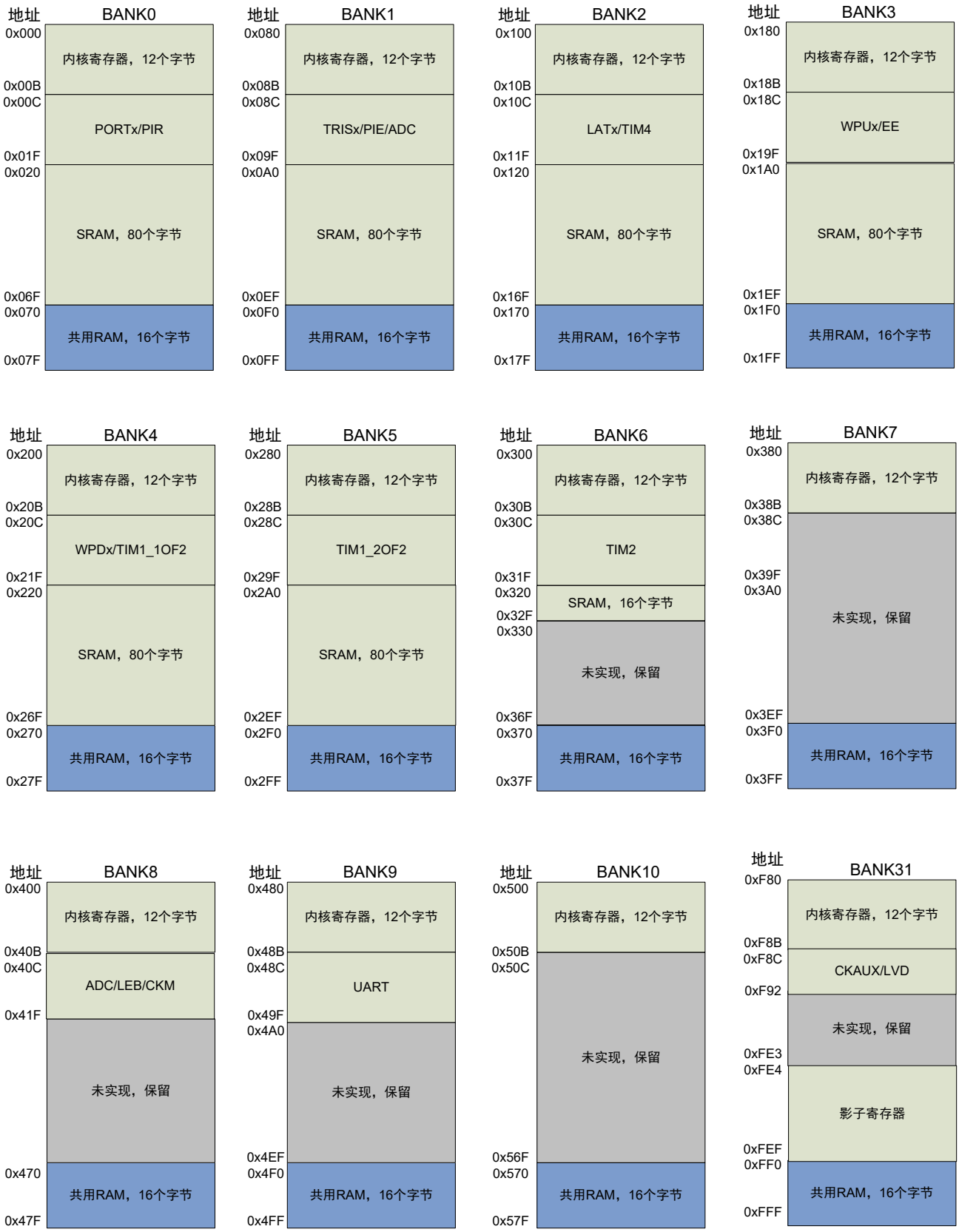


图 19-4 存储区地址映射

因为在切换 bank 时需要额外的指令，因此一些常用的 SFR 同时存储在 13 个 bank 中，以减少切换操作，这些 13 个 bank 所共有的寄存器值是同步的。

Bank	首地址
Bank0	000H
Bank1	080H
Bank2	100H
Bank3	180H
Bank4	200H
Bank5	280H
Bank6	300H
Bank7	380H
Bank8	400H
Bank9	480H
Bank10	500H
Bank31	F80H

表 19-1 BANK 首地址

地址	名称	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	复位值
首地址 + 0H	INDF0	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
首地址 + 1H	INDF1	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
首地址 + 2H	PCL	程序计数器 (PC) 低 8 位								0000 0000
首地址 + 3H	STATUS	-	-	-	/TO	/PD	Z	DC	C	0001 1xxx
首地址 + 4H	FSR0L	间接寻址指针寄存器								xxxx xxxx
首地址 + 5H	FSR0H									xxxx xxxx
首地址 + 6H	FSR1L									xxxx xxxx
首地址 + 7H	FSR1H									xxxx xxxx
首地址 + 8H	BSREG	BANK 选择寄存器								xxxx xxxx
首地址 + 9H	WREG	WORK 寄存器								xxxx xxxx
首地址 + 0AH	PCLATH	-	-	-	程序计数器 (PC) 高 5 位锁存器					---0 0000
首地址 + 0BH	INTCON	GIE	PEIE	EEIE	LVDTE	OSFIE	EEIF	LVDIF	OSFIIF	0000 0000
首地址 + (70 – 7F)		公共 BANK SRAM 区								xxxx xxxx

表 19-2 BANK 共有寄存器

地址	名称	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	复位值
00	INDF0	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
01	INDF1	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
02	PCL	程序计数器低 8 位								0000 0000
03	STATUS	-	-	-	/TF	/PF	Z	DC	C	---1 1xxx
04	FSR0L	间接寻址指针寄存器 0 [15:0]								xxxx xxxx
05	FSR0H									xxxx xxxx
06	FSR1L	间接寻址指针寄存器 1 [15:0]								xxxx xxxx
07	FSR1H									xxxx xxxx
08	BSREG	存储区选择寄存器 (Bank Select Register)								xxxx xxxx
09	WREG	工作寄存器 W								xxxx xxxx
0A	PCLATH	-	-	-	程序计数器高 5 位锁存器					---0 0000
0B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
0C	PORTA	PORTA [7:0]								xxxx xxxx
0D	PORTB	PORTB [7:0]								xxxx xxxx
0E	PORTC	PORTC[7:0]								xxxx xxxx
11	PIR1	-	-	-	-	-	-	CKMIF	ADCIF	---- --00
14	EPIF0	外部管脚中断标志位								0000 0000
1F	ADDLY /LEBPRL	ADC 外部触发启动延时计数器阈值低 8 位 / 复用为 前沿消隐计数阈值								0000 0000
20–6F		SRAM BANK0 (80Bytes)								xxxx xxxx
70–7F		SRAM BANK0 (16Bytes), 物理地址 0x70–0x7F								xxxx xxxx

表 19-3 SFR, BANK 0

地址	名称	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	复位值
80	INDF0	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
81	INDF1	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
82	PCL	程序计数器低 8 位								0000 0000
83	STATUS	-	-	-	/TF	/PF	Z	DC	C	---1 1xxx
84	FSR0L	间接寻址指针寄存器 0 [15:0]								xxxx xxxx
85	FSR0H									xxxx xxxx
86	FSR1L	间接寻址指针寄存器 1 [15:0]								xxxx xxxx
87	FSR1H									xxxx xxxx
88	BSREG	存储区选择寄存器 (Bank Select Register)								xxxx xxxx
89	WREG	工作寄存器 W								xxxx xxxx
8A	PCLATH	-	-		程序计数器高 5 位锁存器					---0 0000
8B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
8C	TRISA	TRISA[7:0]								1111 1111
8D	TRISB	TRISB[7:0]								1111 1111
8E	TRISC	TRISC[7:0]								1111 1111
91	PIE1	-	-	-	-	-	-	CKMIE	ADCIE	---- --00
94	EPIE0	外部中断使能位								0000 0000
95	CKOCON	SYSON	CCORDY	DTYSEL		CCOSEL[2:0]			CCOEN	0010 0000
96	PCON	STKOVF	STKUNF	EMCF	IERRR	/MCLRR	/SRSTF	/PORF	/BORF	qqqq qqqq
97	WDTCON	WDTPRE[2:0]			WDTPS[3:0]				SWDTEN	1110 1000
98	OSCTUNE	HIRC 时钟频率调节位								-xxx xxxx
99	OSCCON	MCKCF[3:0]				OSTS	HTS	LTS	SCS	0011 0000
9A	PCKEN	-	-	UARTEN	-	TIM4EN	TIM2EN	TIM1EN	ADCEN	--0- 0000
9B	ADRESL	ADC 结果寄存器低 8 位								0000 0000
9C	ADRESH	ADC 结果寄存器高 8 位								0000 0000
9D	ADCON0	-	CHS[2:0]			-	ADEX	GO/DONE	ADON	-000 -000
9E	ADCON1	ADFM	ADCS[2:0]			ADNREF[1:0]		ADPREF[1:0]		0000 0000
9F	ADCON2	ADINTREF[1:0]		ETGTYP[1:0]		ADDLY.8	ETGSEL[2:0]			0000 0000
A0–EF		SRAM BANK1 (80Bytes)								xxxx xxxx
F0–FF		SRAM BANK1 (16Bytes), 访问 BANK0's 物理地址 0x70–0x7F								xxxx xxxx

表 19-4 SFR, BANK 1

地址	名称	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	复位值
100	INDF0	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
101	INDF1	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
102	PCL	程序计数器低 8 位								0000 0000
103	STATUS	–	–	–	/TF	/PF	Z	DC	C	---1 1xxx
104	FSR0L	间接寻址指针寄存器 0 [15:0]								xxxx xxxx
105	FSR0H									xxxx xxxx
106	FSR1L	间接寻址指针寄存器 1 [15:0]								xxxx xxxx
107	FSR1H									xxxx xxxx
108	BSREG	存储区选择寄存器 (Bank Select Register)								xxxx xxxx
109	WREG	工作寄存器 W								xxxx xxxx
10A	PCLATH	–	–	–	程序计数器高 5 位锁存器					---0 0000
10B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
10C	LATA	LATA[7:0]								xxxx xxxx
10D	LATB	LATB[7:0]								xxxx xxxx
10E	LATC	–	–	–	–	–	–	LATC[1:0]		---- --xx
111	TIM4CR1	T4ARPE	–	T4CKS[1:0]		T4OPM	T4URS	T4UDIS	T4CEN	0-00 0000
112	TIM4IER	–	–	–	–	–	–	–	T4UIE	---- ---0
113	TIM4SR	–	–	–	–	–	–	–	T4UIF	---- ---0
114	TIM4EGR	–	–	–	–	–	–	–	T4UG	---- ---0
115	TIM4CNTR	T4CNT[7:0]								0000 0000
116	TIM4PSCR	-	-	-	-	-	T4PSC[2:0]			---- -000
117	TIM4ARR	T4ARR[7:0]								1111 1111
118	EPS0	EPS0[7:0]								0000 0000
119	EPS1	EPS1[7:0]								0000 0000
11A	PSRC0	PSRCA[7:0]								1111 1111
11B	PSRC1	PSRCB[7:0]								1111 1111
11C	PSRC2	-	-	-	-	PSRCC[3:2]		PSRCC[1:0]		---- 1100
11E	ITYPE0	ITYPE0[7:0]								0000 0000
11F	ITYPE1	ITYPE1[7:0]								0000 0000
120–16F		SRAM BANK2 (80Bytes)								xxxx xxxx
170–17F		SRAM BANK2 (16Bytes), 访问 BANK0's 物理地址 0x70–0x7F								xxxx xxxx

表 19-5 SFR, BANK 2

地址	名称	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	复位值
180	INDF0	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
181	INDF1	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
182	PCL	程序计数器低 8 位								0000 0000
183	STATUS	-	-	-	/TF	/PF	Z	DC	C	---1 1xxx
184	FSR0L	间接寻址指针寄存器 0 [15:0]								xxxx xxxx
185	FSR0H									xxxx xxxx
186	FSR1L	间接寻址指针寄存器 1 [15:0]								xxxx xxxx
187	FSR1H									xxxx xxxx
188	BSREG	存储区选择寄存器 (Bank Select Register)								xxxx xxxx
189	WREG	工作寄存器 W								xxxx xxxx
18A	PCLATH	-	-	-	程序计数器高 5 位锁存器					---0 0000
18B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
18C	WPUA	WPUA[7:0]								0000 0000
18D	WPUB	WPUB[7:0]								0000 0000
18E	WPUC	-	-	-	-	-	-	WPUC[1:0]		---- --00
191	EEADRL	EEADR[7:0]								0000 0000
192	EEADRH	-	-	-	-	EEADR[11:8]				-000 0000
193	EEDATL	EEDAT[7:0]								xxxx xxxx
194	EEDATH	-	-	EEDAT[13:8]						--xx xxxx
195	EECON1	EEPGD	CFGS	-	FREE	WRERR	WREN	WR	RD	00-0 0000
196	EECON2	EEPROM 控制寄存器 2								xxxx xxxx
197	ANSELA	模拟管脚设置寄存器								-000 0000
199	LVDCON0	SLVREN	LVDM	-	LVDEN	LVDW	LVDL[3:0]			0000 0000
19A	PSINK0	PSINK0[7:0]								0000 0000
19B	PSINK1	PSINK1[7:0]								0000 0000
19C	PSINK2	-	-	-	-	-	-	PSINK2[1:0]		---- --00
19D	MISC0	-	-	-	-	-	-	WCKSEL[1:0]		---- --00
19E	AFP0	-	-	AFP0[5:1]					-	--00 000-
19F	AFP1	-	-	AFP1[5:0]						--00 0000
1A0–1EF		SRAM BANK3 (80Bytes)								xxxx xxxx
1F0–1FF		SRAM BANK3 (16Bytes), 访问 BANK0's 物理地址 0x70–0x7F								xxxx xxxx

表 19-6 SFR, BANK3

地址	名称	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	复位值
200	INDF0	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
201	INDF1	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
202	PCL	程序计数器低 8 位								0000 0000
203	STATUS	–	–	–	/TF	/PF	Z	DC	C	0001 1xxx
204	FSR0L	间接寻址指针寄存器 0 [15:0]								xxxx xxxx
205	FSR0H									xxxx xxxx
206	FSR1L	间接寻址指针寄存器 1 [15:0]								xxxx xxxx
207	FSR1H									xxxx xxxx
208	BSREG	存储区选择寄存器 (Bank Select Register)								xxxx xxxx
209	WREG	工作寄存器 W								xxxx xxxx
20A	PCLATH	–	–	–	程序计数器高 5 位锁存器					---0 0000
20B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
20C	WPDA	WPDA[7:0]								0000 0000
20D	WPDB	WPDB[7:0]								0000 0000
20E	WPDC	–	–	–	–	–	–	WPDC[1:0]		---- --00
211	TIM1CR1	T1ARPE	T1CMS[1:0]		TIDIR	T1OPM	TIURS	T1UDS	T1CEN	0000 0000
212	TIM1CR2	T1ENCTRL	–	–	–	–	T1COMS	–	T1CCPC	0--- -0-0
213	TIM1SMCR	-	T1TS[2:0]			-	T1SMS[2:0]			-000 -000
215	TIM1IER	T1BIE	T1TIE	T1COMIE	T1CC4IE	T1CC3IE	T1CC2IE	T1CC1IE	T1UIE	0000 0000
216	TIM1SR1	T1BIF	T1TIF	T1COMIF	T1CC4IF	T1CC3IF	T1CC2IF	T1CC1IF	T1UIF	0000 0000
217	TIM1SR2	-	-	-	T1CC4OF	T1CC3OF	T1CC2OF	T1CC1OF	-	---0 000-
218	TIM1EGR	T1BG	T1TG	T1COMG	T1CC4G	T1CC3G	T1CC2G	T1CC1G	T1UG	0—0 000-
219	TIM1CCMR1 (output mode)	T1OC1CE	T1OC1M[2:0]			T1OC1PE	T1OC1FE	T1CC1S[1:0]		0000 0000
	TIM1CCMR1 (input mode)	T1IC1F[3:0]				T1IC1PSC[1:0]		T1CC1S[1:0]		0000 0000
21A	TIM1CCMR2 (output mode)	T1OC2CE	T1OC2M[2:0]			T1OC2PE	T1OC2FE	T1CC2S[1:0]		0000 0000
	TIM1CCMR2 (input mode)	T1IC2F[3:0]				T1IC2PSC[1:0]		T1CC2S[1:0]		0000 0000
21B	TIM1CCMR3 (output mode)	T1OC3CE	T1OC3M[2:0]			T1OC3PE	T1OC3FE	T1CC3S[1:0]		0000 0000
	TIM1CCMR3 (input mode)	T1IC3F[3:0]				T1IC3PSC[1:0]		T1CC3S[1:0]		0000 0000
21C	TIM1CCMR4 (output mode)	T1OC4CE	T1OC4M[2:0]			T1OC4PE	T1OC4FE	T1CC4S[1:0]		0000 0000
	TIM1CCMR4 (input mode)	T1IC4F[3:0]				T1IC4PSC[1:0]		T1CC4S[1:0]		0000 0000
21D	TIM1CCER1	T1CC2NP	T1CC2NE	T1CC2P	T1CC2E	T1CC1NP	T1CC1NE	T1CC1P	T1CC1E	0000 0000
21E	TIM1CCER2	T1GP	T1SMOD	T1CC4P	T1CC4E	T1CC3NP	T1CC3NE	T1CC3P	T1CC3E	0000 0000
21F	ODCON0	–	–	–	–	–	–	–	UROD	---- ---0
220–26F		SRAM BANK4 (80Bytes)								xxxx xxxx
270–27F		SRAM BANK4 (16Bytes), 访问 BANK0's 物理地址 0x70–0x7F								xxxx xxxx

表 19-7 SFR, BANK4

地址	名称	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	复位值
280	INDF0	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
281	INDF1	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
282	PCL	程序计数器低 8 位								0000 0000
283	STATUS	–	–	–	/TF	/PF	Z	DC	C	0001 1xxx
284	FSR0L	间接寻址指针寄存器 0 [15:0]								xxxx xxxx
285	FSR0H									xxxx xxxx
286	FSR1L	间接寻址指针寄存器 1 [15:0]								xxxx xxxx
287	FSR1H									xxxx xxxx
288	BSREG	存储区选择寄存器 (Bank Select Register)								xxxx xxxx
289	WREG	工作寄存器 W								xxxx xxxx
28A	PCLATH	–	–	–	程序计数器高 5 位锁存器					---0 0000
28B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
28C	TIM1CNTRH	T1CNT[15:8]								0000 0000
28D	TIM1CNTRL	T1CNT [7:0]								0000 0000
28E	TIM1PSCRH	T1PSC[15:8]								0000 0000
28F	TIM1PSCRL	T1PSC[7:0]								0000 0000
290	TIM1ARRH	T1ARR[15:8]								1111 1111
291	TIM1ARRL	T1ARR[7:0]								1111 1111
292	TIM1RCR	TIREP[7:0]								0000 0000
293	TIM1CCR1H	T1CCR1[15:8]								0000 0000
294	TIM1CCR1L	T1CCR1[7:0]								0000 0000
295	TIM1CCR2H	T1CCR2[15:8]								0000 0000
296	TIM1CCR2L	T1CCR2[7:0]								0000 0000
297	TIM1CCR3H	T1CCR3[15:8]								0000 0000
298	TIM1CCR3L	T1CCR3[7:0]								0000 0000
299	TIM1CCR4H	T1CCR4[15:8]								0000 0000
29A	TIM1CCR4L	T1CCR4[7:0]								0000 0000
29B	TIM1BKR	T1MOE	T1AOE	T1BKP	T1BKE	T1OSSR	T1OSSI	T1LOCK[1:0]		0000 0000
29C	TIM1DTR	T1DGT[7:0]								0000 0000
29D	TIM1OISR	-	T1OIS4	T1OIS3N	T1OIS3	T1OIS2N	T1OIS2	T1OIS1N	T1OIS1	-000 0000
29E	TIM2CCR3H	T2CCR3[15:8]								0000 0000
29F	TIM2CCR3L	T2CCR3 [7:0]								0000 0000
2A0–2EF		SRAM BANK5 (80Bytes)								xxxx xxxx
2F0–2FF		SRAM BANK5 (16Bytes), 访问 BANK0's 物理地址 0x70–0x7F								xxxx xxxx

表 19-8 SFR, BANK5

地址	名称	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	复位值
300	INDF0	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
301	INDF1	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
302	PCL	程序计数器低 8 位								0000 0000
303	STATUS	–	–	–	/TF	/PF	Z	DC	C	0001 1xxx
304	FSR0L	间接寻址指针寄存器 0 [15:0]								xxxx xxxx
305	FSR0H									xxxx xxxx
306	FSR1L	间接寻址指针寄存器 1 [15:0]								xxxx xxxx
307	FSR1H									xxxx xxxx
308	BSREG	存储区选择寄存器 (Bank Select Register)								xxxx xxxx
309	WREG	工作寄存器 W								xxxx xxxx
30A	PCLATH	–	–	–	程序计数器高 5 位锁存器					---0 0000
30B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
30C	TIM2CR1	T2ARPE	-	-	-	T2OPM	T2URS	T2UDIS	T2CEN	0--- 0000
30D	TIM2IER	-	-	-	-	T2CC3IE	T2CC2IE	T2CC1IE	T2UIE	---- 0000
30E	TIM2SR1	-				T2CC3IF	T2CC2IF	T2CC1IF	T2UIF	---- 0000
30F	TIM2SR2	-				T2CC3OF	T2CC2OF	T2CC1OF	-	---- 000-
310	TIM2EGR	-				T2CC3G	T2CC2G	T2CC1G	T2UG	---- 0000
311	TIM2CCMR1 (output mode)	-	T2OC1M[2:0]			T2OC1PE	-	T2CC1S[1:0]		-000 0-00
	TIM2CCMR1 (input mode)	T2OC1M[2:0]				T2OC1PE		T2OC1M[2:0]		0000 0000
312	TIM2CCMR2 (output mode)	-	T2OC2M[2:0]			T2OC2PE	-	T2CC2S[1:0]		-000 0-00
	TIM2CCMR2 (input mode)	T2IC2F[3:0]				T2IC2PSC[1:0]		T2CC2S[1:0]		0000 0000
313	TIM2CCMR3 (output mode)	-	T2OC3M[2:0]			T2OC3PE	-	T2CC3S[1:0]		-000 0-00
	TIM2CCMR3 (input mode)	T2IC3F[3:0]				T2IC3PSC[1:0]		T2CC3S[1:0]		0000 0000
314	TIM2CCER1	-	-	T2CC2P	T2CC2E	-	-	T2CC1P	T2CC1E	--00 --00
315	TIM2CCER2	-	-	-	-	-	-	T2CC3P	T2CC3E	---- --00
316	TIM2CNTRH	T2CNT[15:8]								0000 0000
317	TIM2CNTRL	T2CNT[7:0]								0000 0000
318	TIM2PSCR	-	-	-	-	T2PSC[3:0]				---- 0000
319	TIM2ARRH	T2ARR[15:8]								1111 1111
31A	TIM2ARRL	T2ARR[7:0]								1111 1111
31B	TIM2CCR1H	T2CCR1[15:8]								0000 0000
31C	TIM2CCR1L	T2CCR1[7:0]								0000 0000
31D	TIM2CCR2H	T2CCR2[15:8]								0000 0000
31E	TIM2CCR2L	T2CCR2[7:0]								0000 0000
31F	TCKSRC	LFMOD	T2CKSRC[2:0]			-	T1CKSRC[2:0]			0000 -000
320–32F		SRAM BANK6 (16Bytes)								xxxx xxxx
370–37F		SRAM BANK6 (16Bytes), 访问 BANK0's 物理地址 0x70–0x7F								xxxx xxxx

表 19-9 SFR, BANK6

地址	名称	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	复位值
380	INDF0	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
381	INDF1	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
382	PCL	程序计数器低 8 位								0000 0000
383	STATUS	–	–	–	/TF	/PF	Z	DC	C	0001 1xxx
384	FSR0L	间接寻址指针寄存器 0 [15:0]								xxxx xxxx
385	FSR0H									xxxx xxxx
386	FSR1L	间接寻址指针寄存器 1 [15:0]								xxxx xxxx
387	FSR1H									xxxx xxxx
388	BSREG	存储区选择寄存器 (Bank Select Register)								xxxx xxxx
389	WREG	工作寄存器 W								xxxx xxxx
38A	PCLATH	–	–	–	程序计数器高 5 位锁存器					---0 0000
38B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
3F0–3FF		SRAM BANK7 (16Bytes), 访问 BANK0's 物理地址 0x70–0x7F								xxxx xxxx

表 19-10 SFR, BANK7

地址	名称	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	复位值
400	INDF0	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
401	INDF1	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
402	PCL	程序计数器低 8 位								0000 0000
403	STATUS	–	–	–	/TF	/PF	Z	DC	C	0001 1xxx
404	FSR0L	间接寻址指针寄存器 0 [15:0]								xxxx xxxx
405	FSR0H									xxxx xxxx
406	FSR1L	间接寻址指针寄存器 1 [15:0]								xxxx xxxx
407	FSR1H									xxxx xxxx
408	BSREG	存储区选择寄存器 (Bank Select Register)								xxxx xxxx
409	WREG	工作寄存器 W								xxxx xxxx
40A	PCLATH	–	–	–	程序计数器高 5 位锁存器					---0 0000
40B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
41A	ADCON3	ADFBEN	ADCMPOP	ADCMPEN	ADCMPO	LEBADT	-	ELVDS[1:0]		0000 0-00
41B	ADCMPH	ADCMPH[7:0]								0000 0000
41C	LEBCON	LEBEN	LEBCH[1:0]		-	EDGS	BKS[2:0]			000- 0000
41D	MSCKCON	-	-	-	-	-	-	CKMAVG	CKCNTI	---- --01
41E	SOSCPRL	SOSCPR[7:0]								1111 1111
41F	SOSCPRH	-	-	-	-	SOSCPR[11:8]				---- 1111
470–47F		SRAM BANK8 (16Bytes), 访问 BANK0's 物理地址 0x70–0x7F								xxxx xxxx

表 19-11SFR, BANK8

地址	名称	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	复位值
480	INDF0	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
481	INDF1	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
482	PCL	程序计数器低 8 位								0000 0000
483	STATUS	–	–	–	/TF	/PF	Z	DC	C	0001 1xxx
484	FSR0L	间接寻址指针寄存器 0 [15:0]								xxxx xxxx
485	FSR0H									xxxx xxxx
486	FSR1L	间接寻址指针寄存器 1 [15:0]								xxxx xxxx
487	FSR1H									xxxx xxxx
488	BSREG	存储区选择寄存器 (Bank Select Register)								xxxx xxxx
489	WREG	工作寄存器 W								xxxx xxxx
48A	PCLATH	–	–	–	程序计数器高 5 位锁存器					---0 0000
48B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
48C	URDATAL	DATAL[7:0]								0000 0000
48D	URDATAH	-							DATAH	---- ---0
48E	URIER	–	TCIE	–	IDELE	RXSE	URTE	URRXNE	URRXNE	-0-0 0000
48F	URLCR	-	BKREQ	-	EVEN	PEN	STOP	-	LTH	-0-0 00-0
490	URLCREXT	-	-	-	-	-	-	RWU	EXTEN	---- --00
491	URMCR	-	-	SIRLP	TXEN	RXEN	WAKE	HDSEL	SIREN	--00 0000
492	URLSR	ADDRF	IDLEF	TXEF	BKF	FEF	PEF	OVERF	RXNEF	0000 0000
493	URRAR	-	-	-	-	RAR[3:0]				---- 0000
494	URDLL	DLL[7:0]								0000 0000
495	URDLH	DLH[7:0]								0000 0000
496	URABCR	-	-	-	-	ABRE	ABRM	ABRF	ABREN	---- 0000
497	URSYNCR	-	-	-	-	LBCL	CPHA	CPOL	SYNEN	---- 0000
498	URLINCR	-	-	-	LINEN	BLTH[3:0]				---0 0000
499	URSDCR0	-	NACK	CKOE	SDEN	-	-	-	-	-000 0000
49A	URSDCR1	GT[7:0]								0000 0000
49B	URSDCR2	PSC[7:0]								0000 0000
49C	URTC	-	-	-	-	-	-	-	TCF	---- ---1
4F0–4FF		SRAM BANK9 (16Bytes), 访问 BANK0's 物理地址 0x70–0x7F								xxxx xxxx

表 19-12 SFR, BANK9

地址	名称	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	复位值
500	INDF0	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
501	INDF1	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
502	PCL	程序计数器低 8 位								0000 0000
503	STATUS	–	–	–	/TF	/PF	Z	DC	C	0001 1xxx
504	FSR0L	间接寻址指针寄存器 0 [15:0]								xxxx xxxx
505	FSR0H									xxxx xxxx
506	FSR1L	间接寻址指针寄存器 1 [15:0]								xxxx xxxx
507	FSR1H									xxxx xxxx
508	BSREG	存储区选择寄存器 (Bank Select Register)								xxxx xxxx
509	WREG	工作寄存器 W								xxxx xxxx
50A	PCLATH	–	–	–	程序计数器高 5 位锁存器					---0 0000
50B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
570–57F		SRAM BANK10 (16Bytes), 访问 BANK0's 物理地址 0x70–0x7F								xxxx xxxx

表 19-13 SFR, BANK10

地址	名称	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	复位值
F80	INDF0	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
F81	INDF1	使用 FSR 的内容对数据存储器进行访问 (非物理寄存器)								xxxx xxxx
F82	PCL	程序计数器低 8 位								0000 0000
F83	STATUS	–	–	–	/TF	/PF	Z	DC	C	0001 1xxx
F84	FSR0L	间接寻址指针寄存器 0 [15:0]								xxxx xxxx
F85	FSR0H									xxxx xxxx
F86	FSR1L	间接寻址指针寄存器 1 [15:0]								xxxx xxxx
F87	FSR1H									xxxx xxxx
F88	BSREG	存储区选择寄存器 (Bank Select Register)								xxxx xxxx
F89	WREG	工作寄存器 W								xxxx xxxx
F8A	PCLATH	–	–	–	程序计数器高 5 位锁存器					---0 0000
F8B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
F8D	CKAUX	–	–	–	–	–	–	CLKOS	–	---- --1-
F8E	LVDCON1				LVDP	LVDDEB	LVDOE	LVDOS[1:0]		---0100
F92	LVDTUNE	LVDCAL[3:0]				–	–	–	–	1000 ----
FE4–FEF		Shadow reg								xxxx xxxx
FF0–FFF		SRAM BANK31 (16Bytes), 访问 BANK0's 物理地址 0x70–0x7F								xxxx xxxx

表 19-14 SFR, BANK31

注：

1. INDF 不是物理寄存器；
2. 灰色部分表示没有实现；
3. 不要对未实现的寄存器位进行写操作；

CPU 在进入中断时，硬件会自动把 W 寄存器，STATUS 寄存器 (TO 和 PD 状态标志位除外)，BSR 寄存器，FSR 寄存器以及 PCLATH 寄存器保存到处于 bank31 的影子寄存中，在退出中断时再把它们恢复到对应的寄存器，节省软件的开销。

地址	寄存器名称
FE4h	STATUS_SHAD
FE5h	WREG_SHAD
FE6h	BSREG_SHAD
FE7h	PCLATH_SHAD
FE8h	FSR0L_SHAD
FE9h	FSR0H_SHAD
FEAh	FSR1L_SHAD
FEBh	FSR1H_SHAD
FECh	—
FEDh	STKPTR
FEEh	TOSL
FEFh	TOSH

19.3. 堆栈

FT61F14x 有一个 16 级深 x15 位宽的硬件堆栈。堆栈空间不属于程序存储空间或数据存储空间的一部分。当执行 LCALL 或 LCALLW 指令或由于中断导致程序跳转时，PC 的值会被压入堆栈。当执行 RET、RETW 或 RETI 指令时，PC 值从堆栈弹出。PCLATH 的值不受压栈或出栈操作的影响。

如果 STVREN 位被编程为 0 (配置字寄存器)，则堆栈可作为循环缓冲区使用。这就是说在压栈 16 次后，第 17 次压栈操作会覆盖第 1 次压栈操作存储的值，而第 18 次压栈操作覆盖第 2 次压栈操作存储的值 (以此类推)。不管是否使能复位，STKOVF 和 STKUNF 标志位在上溢/下溢时都会置 1。

访问堆栈

可通过 TOSH、TOSL 和 STKPTR 寄存器来使用堆栈。STKPTR 是堆栈指针的当前值。TOSH:TOSL 寄存器对指向栈顶。这两个寄存器都可读写。由于 PC 的大小为 15 位，故 TOS 划分为 TOSH 和 TOSL 两部分。要访问堆栈，可调整用来定位 TOSH:TOSL 的 STKPTR 值，然后对 TOSH:TOSL 执行读/写操作。STKPTR 为 5 位，允许检测上溢和下溢。

在正常程序运行期间，LCALL、LCALLW 和中断会使 STKPTR 值递增 1，而 RETW、RET 和 RETI 会使 STKPTR 值递减 1。任何时候都可以检查 STKPTR，以查看可用堆栈空间。STKPTR 总是指向堆栈中的

当前使用单元。因此，LCALL 或 LCALLW 指令会使 STKPTR 值递增 1，然后写 PC，而返回操作则会卸载 PC，然后使 STKPTR 值递减 1。

注意：在允许中断的情况下修改 STKPTR 时应谨慎。

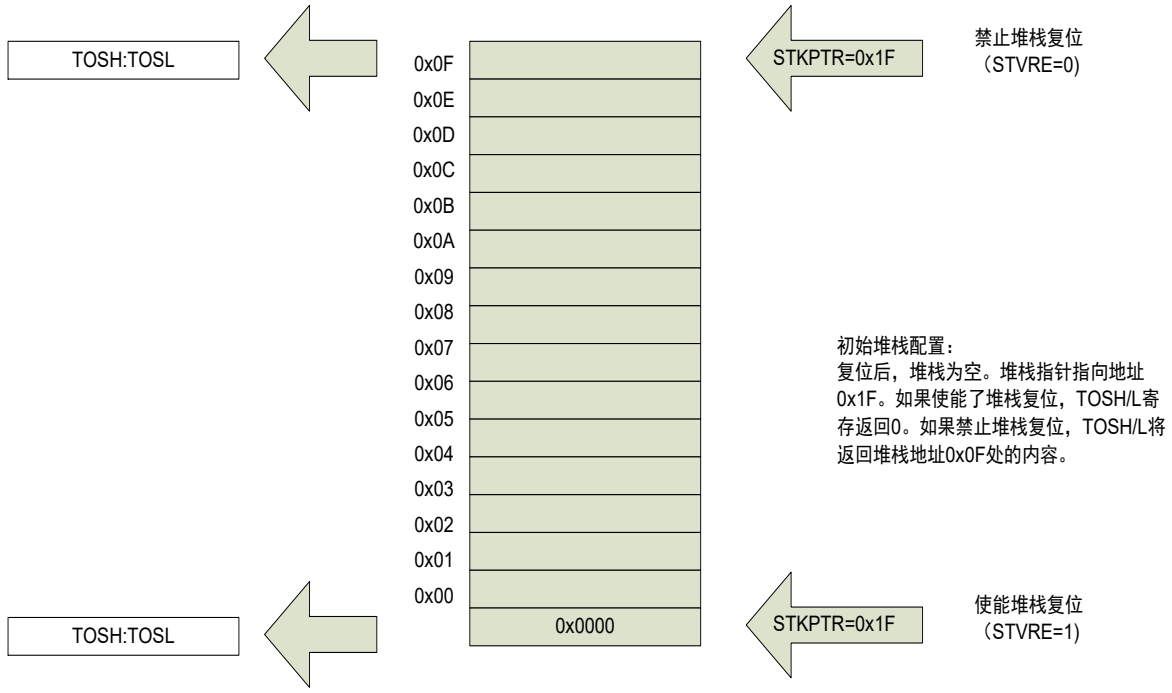


图 19-5 软件访问堆栈

上溢/下溢复位

如果配置字寄存器中的 STVREN 位编程为 1，则在压满 16 级后再执行压栈操作，或者在弹出第 1 级后再执行出栈操作，PCON 寄存器中的相应位 (分别为 STKOVF 或 STKUNF) 会置 1，从而使器件复位。

19.4. STATUS 寄存器

名称	状态	寄存器	地址	复位值
/ TO	超时标志位 1 = 上电后，执行了 CLRWDT 或 SLEEP 指令 0 = 发生 WDT 超时溢出	STATUS[4]	Bank 首地址 + 0x03	RO-1
/PD	掉电标志位 1 = 上电复位后或执行了 CLRWDT 指令 0 = 执行了 SLEEP 指令	STATUS[3]		RO-1
Z	0 标志位：算术或逻辑运算的结果为零？ 1 = Yes 0 = No	STATUS[2]		RW-x
DC	半进位 / 半借位 (ADDWR, ADDWI, SUBWI, SUBWR)：结果的第 4 低位向高位发生了进位或借位？ 1 = 进位，或未借位 0 = 未进位，或借位	STATUS[1]		RW-x
C	进位 / 借位 (ADDWR, ADDWI, SUBWI, SUBWR)：结果的最高位发生了进位或借位？ 1 = 进位，或未借位 0 = 未进位，或借位	STATUS[0]		RW-x

表 19-15 STATUS 寄存器

注：

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。如果一条影响 Z、DC 或 C 位的指令以状态寄存器作为目标寄存器，将禁止写这三位。根据器件逻辑，这些位会被置 1 或清零。 此外，也不能写 TO 和 PD 位。因此，当执行一条把状态寄存器作为目标寄存器的指令后，状态寄存器的结果可能和预想的不一样。
2. 因此，建议仅使用 BCR、BSR、SWAPR 和 STR 指令来改变状态寄存器，因为这些指令不影响任何状态位。
3. 注意：对于借位 C，极性是相反的。减法指令通过加上第二个操作数的二进制补码实现。对于移位指令 (RRR 和 RLR)，此位值来自源寄存器的最高位或最低位。

19.5. 间接寻址

INDFn 寄存器不是物理寄存器。任何访问 INDFn 寄存器的指令，实际上都是访问由文件选择寄存器 (FSR) 指定的地址处的寄存器。如果 FSRn 地址指定了 2 个 INDFn 寄存器中的任何一个，执行读操作会返回 0，而写操作无法实现 (尽管状态位会受影响)。可通过 FSRnH 和 FSRnL 对来创建 FSRn 寄存器值。FSR 寄存器形成的 16 位地址允许对 65536 个地址单元的空间进行寻址。

这些地址单元可划分为 3 个存储区：

- 传统数据存储器
- 线性数据存储器
- 程序存储器

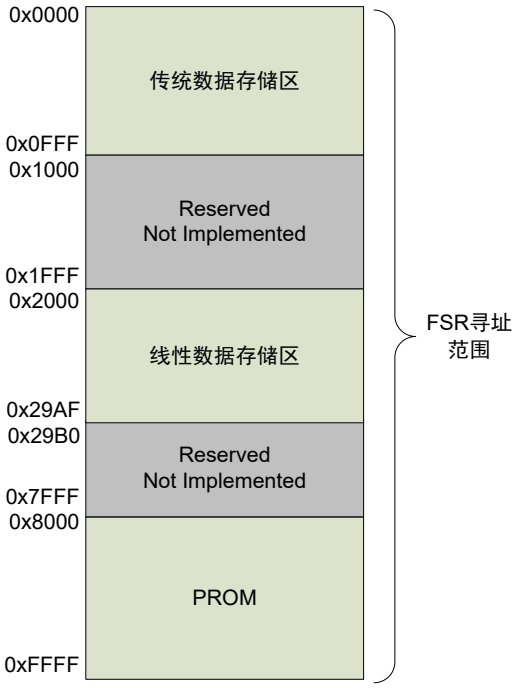


图 19-6 间接寻址

19.5.1. 传统数据存储器

传统数据存储器指的是从 FSR 地址 0x000 到 FSR 地址 0xFFF 的区域。此地址对应于所有 SFR、GPR 和公共寄存器的绝对地址。

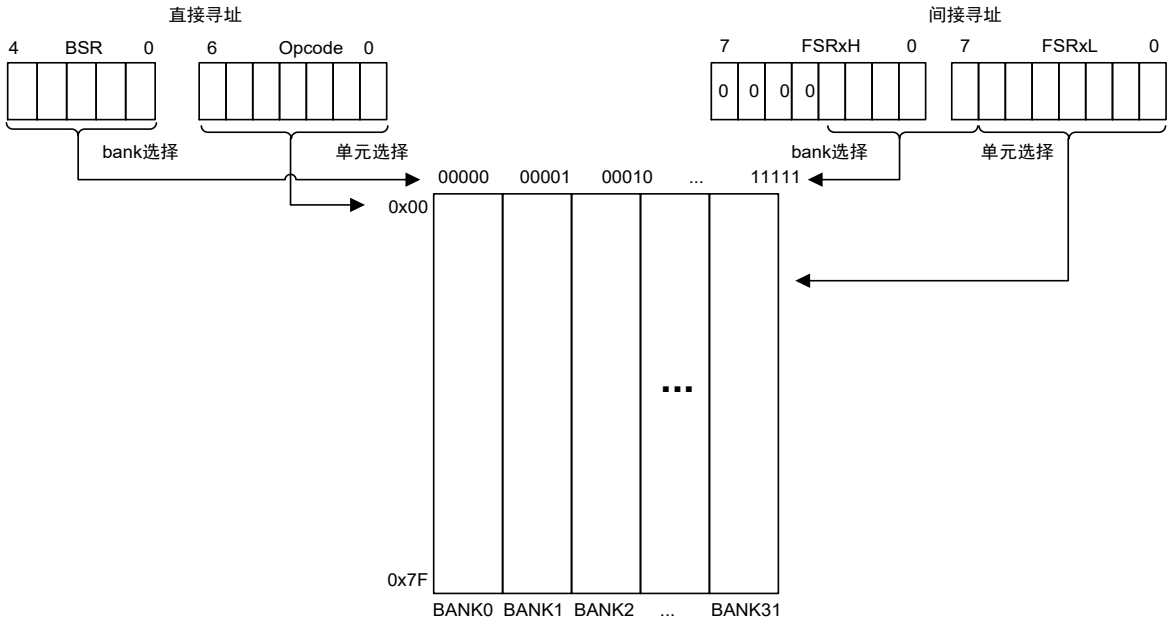


图 19-7 传统数据存储器映射

19.5.2. 线性数据存储器

线性数据存储器指的是从 FSR 地址 0x2000 到 FSR 地址 0x21DF 的区域。该区域为虚拟区域，它指向所有存储区中 80 字节的 GPR 存储区块 (在 FT61F14x 中，只有 Bank0~Bank5 实现了 80B 的 GPR)。

未实现的存储区读为 0x00。使用线性数据存储器区域允许缓冲区大于 80 字节，因为当 FSR 增大到超过一个存储区时，会直接转到下一个存储区的 GPR 存储器。线性数据存储器区域不包含 16 字节的公共存储器。

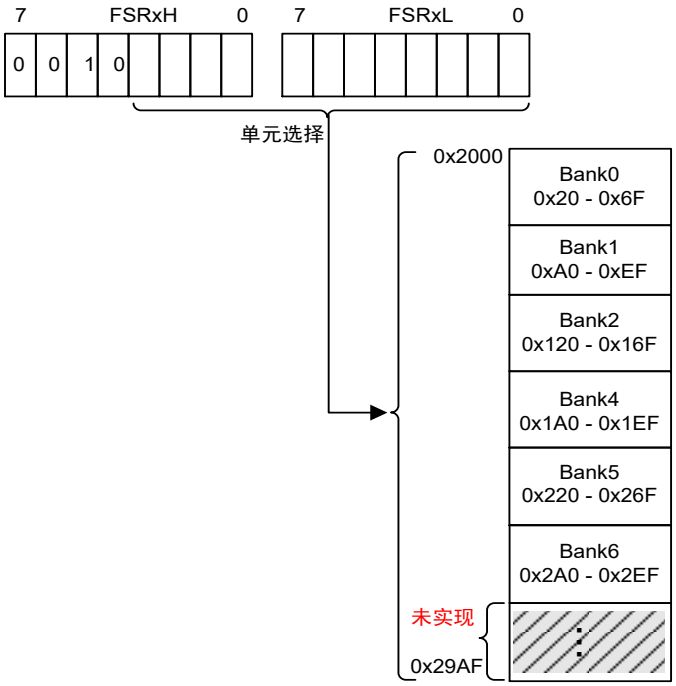


图 19-8 线性数据存储区映射

19.5.3. 程序存储器

要使常数的访问更为容易，可将整个程序存储器映射到 FSR 地址空间的高半部分。当 FSRnH 的 MSB 置 1 时，低 15 位就是可通过 INDF 进行访问的程序存储器的地址。只有每个存储单元的低 8 位可通过 INDF 进行访问。通过 FSR/INDF 接口无法对程序存储器执行写操作。所有通过 FSR/INDF 接口对程序存储器进行访问的指令都需要一个额外的指令周期才能完成。

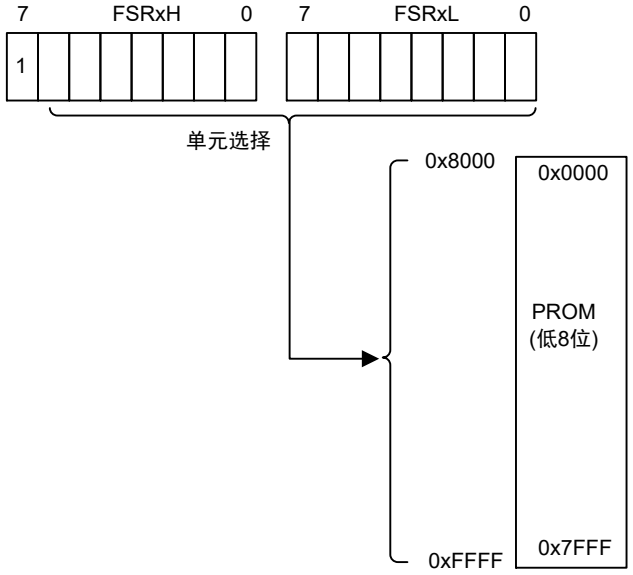


图 19-9 程序存储区映射

下图给出了 CPU 使用 FSRn 间接寻址 PROM 的时序，共占 2 个指令周期。

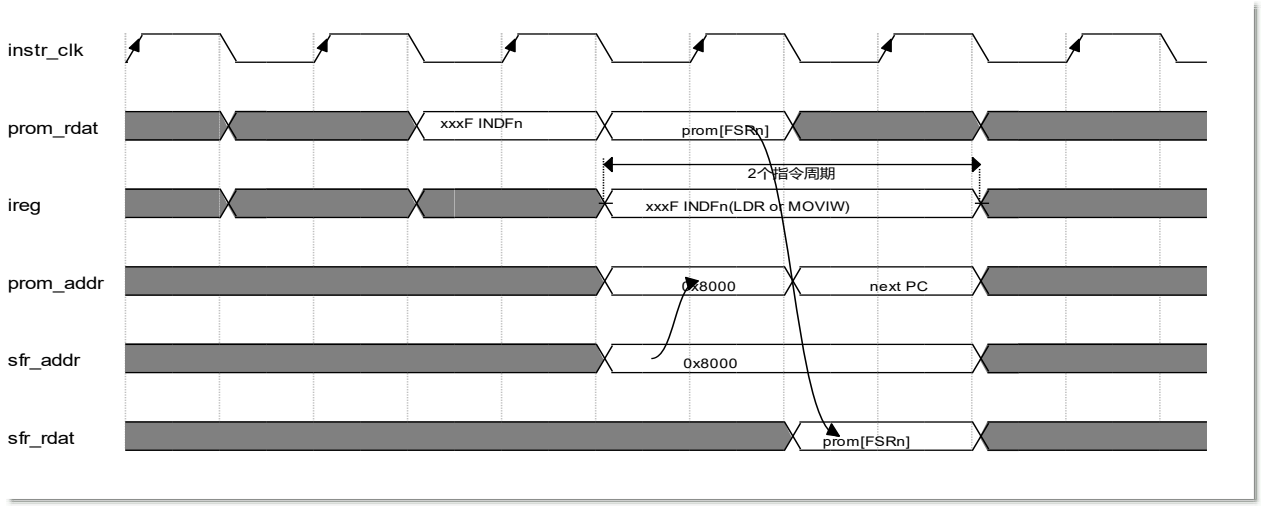


图 19-10 间接寻址读程序存储器

20. 指令集汇总 (INSTRUCTION SET)

汇编语法	功能	运算	指令周期	状态位
NOP	空操作	None	1	NONE
SLEEP	进入 SLEEP 模式	0 → WDT; Stop OSC	1	/PD, /TO
RESET	软件复位	复位寄存器 PCON	1	/SRSTF
CLRWDT	清看门狗 (喂狗)	0 → WDT	1	/PD, /TO
LJUMP N	无条件跳转	N → PC	2	NONE
BRA k	相对跳转 (地址范围有限制)	PC + 1 + k → PC	2	NONE
BRW	将寄存器 W 的值作为偏移量进行相对跳转	PC + w → PC	2	NONE
LCALL N	调用子程序	N → PC; PC + 1 → Stack	2	NONE
CALLW	调用地址由寄存器 W 指定的子程序	W → PC; PC + 1 → Stack	2	NONE
RETW	将立即数 I 送入 W 并返回	I → W, Stack → PC	2	NONE
RETI	从中断返回	Stack → PC; 1 → GIE	2	NONE
RET	从子程序返回	Stack → PC	2	NONE
BCR R, b	将寄存器 R 的 b 位清 0	0 → R(b)	1	NONE
BSR R, b	将寄存器 R 的 b 位置 1	1 → R(b)	1	NONE
CLRR R	将寄存器 R 清 0	0 → R	1	Z
LDR R, d (MOVF)	将 R 存到 d	R → d	1	Z
COMR R, d	R 的补码	/R → d	1	Z
INCR R, d	R + 1	R + 1 → d	1	Z
INCRSZ R, d	R + 1，结果为 0 则跳过	R + 1 → d	1	NONE
DECR R, d	R − 1	R − 1 → d	1	Z
DECRSZ R, d	R − 1，结果为 0 则跳过	R − 1 → d	1	NONE
SWAPR R, d	将寄存器 R 的半字节交换	R(0-3)R(4−7) → d	1	NONE
RRR R, d	R 带进位循环右移	R(0) → C; R(n) → R(n−1); C → R(7);	1	C
RLR R, d	R 带进位循环左移	R(7) → C; R(n) → R(n+1); C → R(0);	1	C
LSRF f, d	F 逻辑右移	0→ f(7); f(n+1) → R(n); f(0)→ C;	1	C, Z
LSLF f, d	F 逻辑左移	f(7) → C; f(n) → R(n+1); 0→ R(0);	1	C, Z
ASRF f, d	算术右移	f(7) → R(7); f(n+1) →R(n); f(0) →C;	1	C, Z
BTSC R, b	位测试，结果为 0 则跳过	Skip if R(b)=0	1	NONE

汇编语法	功能	运算	指令周期	状态位
BTSS R, b	位测试，结果为 1 则跳过	Skip if R(b)=1	1	NONE
CLRW	将工作寄存器 W 清 0	0 → W	1	Z
STR R (MOVWF)	将 W 存到 R	W → R	1	NONE
ADDWR R, d	W 与 R 相加	W + R → d	1	C, DC, Z
ADDWFC R, d	W 与 R 相加(带进位)	W + R + C → d	1	C, DC, Z
SUBWR R, d	R 减 W	R − W → d	1	C, DC, Z
SUBWFB R, d	R 减 W (带借位)	R − W – (/B) → d	1	C, DC, Z
ANDWR R, d	W 与 R 相与	R & W → d	1	Z
IORWR R, d	W 与 R 相或	W | R → d	1	Z
XORWR R, d	W 与 R 异或	W ^ R → d	1	Z
LDWI I (MOVLW)	将立即数存到 W	I → W	1	NONE
ANDWI I	W 与立即数 I 相与	I & W → W	1	Z
IORWI I	W 与立即数 I 相或	I | W → W	1	Z
XORWI I	W 与立即数 I 异或	I ^ W → W	1	Z
ADDWI I	W 与立即数 I 相加	I + W → W	1	C, DC, Z
SUBWI I	立即数 I 减 W	I − W → W	1	C, DC, Z
MOVLB k	将立即数 I 送到存储区选择寄存器 BSR	K → BSR	1	NONE
ADDFSR FSRn, k	立即数 k 与 FSRn 相加	FSRn + k → FSRn	1	NONE
MOVLP	将立即数 k 送到 PCLATH	k → PCLATH	1	NONE
MOVIW mm	将 FSRn 的内容送到 W	FSRn → W	1	Z
MOVWI mm	将 W 的内容送到 FSRn	W → FSRn	1	NONE

表 20-1 49 条指令集

字段	描述	
R(f)	SFR/GPR 地址	
W	工作寄存器	
b	8-bit 寄存器 R / RAM 中的位地址	
I/ Imm (k)	立即数字段、常数或标号	
x	不关心，值可以为 0 或 1	
d	目标寄存器选择	1 = 结果存放到寄存器 R / RAM 0 = 结果存放到 W
mm	预/后增/减模式选择 (++FSRn, --FSRn, FSRn++ , FSRn--, k[FSRn])	
N	程序绝对地址	
PC	程序计数器	
/PD	掉电标志位	

字段	描述
/TO	超时标志位
C	进位 / 借位
DC	半进位 / 半借位
Z	0 标志位

表 20-2 操作码字段

名称	状态	寄存器	地址	复位值
Z	0 标志位：算术或逻辑运算的结果为零？ 1 = Yes 0 = No	STATUS[2]	Bank 首地址 +0x03	RW-x
DC	半进位 / 半借位 (ADDWR, ADDWI, SUBWI, SUBWR)：结果的第 4 低位向高位发生了进位或借位？ 1 = 进位，或未借位 0 = 未进位，或借位	STATUS[1]		RW-x
C	进位 / 借位 (ADDWR, ADDWI, SUBWI, SUBWR)：结果的最高位发生了进位或借位？ 1 = 进位，或未借位 0 = 未进位，或借位	STATUS[0]		RW-x

表 20-3 计算状态标志位

20.1. 读-修改-写 (RMW) 指令

所有需要使用文件寄存器 (**表 20-1** 中助记符带 R(f)的指令) 的指令都会执行读-修改-写 (RMW)操作，即先把目标寄存器内容取出，根据指令修改数据，再把数据写回到目标寄存器或 W (取决于 d 和具体指令)。

举例说明：

```
BSR     FSR0L, 0;
```

上述指令在 CPU 的执行过程如下：

1) 把 FSR0L 读出到临时寄存器 T；
2) 把寄存器 T 或上”0000 0001”形成新数据；
3) 再把新数据写回 FSR0L；

20.1. 指令详细描述

ADDFSR	立即数与 FSRn 相加
语法：	[标号] ADDFSR FSRn, k
操作数：	-32 ≤ k ≤ 31 n∈[0,1]
操作：	FSR(n)+k → FSR(n)
受影响的状态位：	无
说明：	将有符号 6 位立即数 k 与 FSRnH:FSRnL 寄存器对的内容相加。 FSRn 地址范围限制为 0000h-FFFFh。传送地址超出该边界时，FSR 会发生折回。

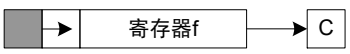
ADDWI	立即数与 W 相加
语法：	[标号] ADDWI k
操作数：	0 ≤ k ≤ 255
操作：	(W)+k → (W)
受影响的状态位：	C、DC 和 Z
说明：	将 W 寄存器的内容与 8 位立即数 k 相加，结果存入 W 寄存器。

ADDWR	W 与 f 相加
语法：	[标号]ADDWR f,d
操作数：	0≤f≤127 d∈[0,1]
操作：	(W)+(f)→(目标寄存器)
受影响的状态位：	C、DC 和 Z
说明：	将 W 寄存器的内容与寄存器 f 的内容相加。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。

ANDWI	立即数和 W 作逻辑与运算
语法：	[标号] ANDWI k
操作数：	0 ≤ k ≤ 255
操作：	(W).AND.(k) → (W)
受影响的状态位：	Z
说明：	将 W 寄存器的内容与 8 位立即数 k 进行逻辑与运算。结果存入 W 寄存器。

ANDWR	W 和 f 作逻辑与运算
语法：	[标号] ANDWR f,d
操作数：	0 ≤ f ≤ 127 d∈[0,1]
操作：	(W).AND.(f) →(目标寄存器)
受影响的状态位：	Z
说明：	将 W 寄存器的内容与寄存器 f 的内容进行逻辑与运算。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。

ASRF	算术右移
语法：	[标号] ASRF f {,d}
操作数：	0≤f≤127 d∈[0,1]
操作：	(f[7])→目标寄存器[7] (f[7:1])→目标寄存器[6:0], (f[0])→C
受影响的状态位：	C 和 Z
说明：	将寄存器 f 的内容连同进位标志位一起右移 1 位。MSb 保持不变。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1,结果存回寄存器 f。



ADDWFC	W 与 f 相加 (带进位)
语法：	[标号] ADDWFC f {,d}
操作数：	0≤f≤127 d∈[0,1]
操作：	(W)+(f)+(C)→目标寄存器
受影响的状态位：	C、DC 和 Z
说明：	将 W 的内容、进位标志位与数据存储单元 f 的内容相加。如果 d 为 0，结果存入 W。如果 d 为 1，结果存入数据存储单元 f。

BCR	将 f 寄存器中的某位清零
语法：	[标号] BCR f, b
操作数：	0≤f≤127 0≤b≤7
操作：	0→(f[b])
受影响的状态位：	无
说明：	将寄存器 f 中的位 b 清零。

BRA	相对跳转
语法：	[标号] BRA 标号 [标号] BRA $+k
操作数：	-256≤标号 - PC+1≤255 -256≤k≤255
操作：	(PC)+1+k→PC
受影响的状态位：	无
说明：	将有符号 9 位立即数 k 与 PC 相加。由于 PC 将递增 1 以便取下一条指令，所以新地址将为 PC+1+k。该指令为一条双周期指令。该跳转的地址范围存在限制。

BRW	将 W 寄存器的内容作为偏移量进行相对跳转
语法：	[标号] BRW
操作数：	无
操作：	(PC)+(W)→PC
受影响的状态位：	无
说明：	将 W 的内容 (无符号) 与 PC 相加。由于 PC 将递增 1 以取出下一条指令，所以新地址将为 PC+1+(W)。该指令为一条双周期指令。

BSR	将 f 中的某位置 1
语法：	[标号] BSR f,b
操作数：	0≤f≤127 0≤b≤7
操作：	1→(f[b])
受影响的状态位：	无
说明：	将寄存器 f 的位 b 置 1。

BTSC	测试 f 中某位，为 0 则跳过
语法：	[标号] BTSC f,b
操作数：	0≤f≤127 0≤b≤7
操作：	如果(f[b])=0，则跳过
受影响的状态位：	无
说明：	如果寄存器 f 的位 b 为 1，则执行下一条指令。如果寄存器 f 的位 b 为 0，则丢弃下一条指令，转而执行一条 NOP 指令，从而使该指令成为双周期指令。

BTSS	测试 f 中某位,为 1 则跳过
语法：	[标号] BTSS f,b
操作数：	0≤f≤127 0≤b≤7
操作：	如果(f[b])=1，则跳过
受影响的状态位：	无
说明：	如果寄存器 f 的位 b 为 0，则执行下一条指令。如果位 b 为 1，则丢弃下一条指令，转而执行一条 NOP 指令，从而使该指令成为双周期指令。

CLRW	将 W 寄存器清零
语法：	[标号] CLRW
操作数：	无
操作：	00h→(W) 1→Z
受影响的状态位：	Z
说明：	W 寄存器被清零。全零位 (Z) 被置 1。

LCALL	调用子程序
语法：	[标号] LCALL k
操作数：	0≤k≤2047
操作：	(PC)+1→TOS， k→PC[10:0]， (PCLATH[4:3])→PC[12:11]
受影响的状态位：	无
说明：	调用子程序。首先，将返回地址 (PC+1)压入堆栈。将 11 位立即数地址装入 PC 的[10:0]位。将 PCLATH 的内容装入 PC 的高位。LCALL 是双周期指令。

CALLW	调用地址由 W 寄存器指定的子程序
语法：	[标号] CALLW
操作数：	无
操作：	(PC)+1→TOS， (W)→PC[7:0]， (PCLATH[6:0])→PC[14:8]
受影响的状态位：	无
说明：	调用地址由 W 寄存器指定的子程序。首先，将返回地址 (PC+1)压入返回堆栈。然后，W 的内容被装入 PC[7:0]，将 PCLATH 的内容装入 PC[14:8]。CALLW 是双周期指令。

CLRR	将 f 清零
语法：	[标号] CLRR f
操作数：	0≤f≤127
操作：	00h→(f) 1→Z
受影响的状态位：	Z
说明：	寄存器 f 的内容被清零，并且 Z 位被置 1。

CLRWDT	将看门狗定时器清零
语法：	[标号] CLRWDT
操作数：	无
操作：	00h→WDT 0→WDT 预分频器 1→/TO 1→/PD
受影响的状态位：	/TO 和/PD
说明：	CLRWDT 指令复位看门狗定时器及其预分频器。 状态位/TO 和/PD 均被置 1。

COMR	f 取反
语法：	[标号] COMR f,d
操作数：	0≤f≤127 d∈[0,1]
操作：	$(\bar{f})$→(目标寄存器)
受影响的状态位：	Z
说明：	将寄存器 f 的内容取反。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。

LJUMP	无条件跳转
语法：	[标号] LJUMP k
操作数：	0≤k≤2047
操作：	k→PC[10:0] PCLATH[4:3]→PC[12:11]
受影响的状态位：	无
说明：	LJUMP 是无条件跳转指令。将 11 位立即数值装入 PC 的[10:0]位。PC 的高位从 PCLATH[4:3]装入。LJUMP 是双周期指令。

DECR	f 递减 1
语法：	[标号] DECR f,d
操作数：	0≤f≤127 d∈[0,1]
操作：	(f)-1→(目标寄存器)
受影响的状态位：	Z
说明：	将寄存器 f 的内容递减 1。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。

INCR	f 递增 1
语法：	[标号] INCR f,d
操作数：	0≤f≤127 d∈[0,1]
操作：	(f)+1→(目标寄存器)
受影响的状态位：	Z
说明：	将寄存器 f 的内容递增 1。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。

DECRSZ	f 递减 1，为 0 则跳过
语法：	[标号]DECRSZ f,d
操作数：	0≤f≤127 d∈[0,1]
操作：	(f)-1→(目标寄存器)； 结果=0 则跳过
受影响的状态位：	无
说明：	将寄存器 f 的内容递减 1。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。如果结果为 1，则执行下一条指令。如果结果为 0，则转而执行一条 NOP 指令，从而使该指令成为双周期指令。

INCRSZ	f 递增 1，为 0 则跳过
语法：	[标号] INCRSZ f,d
操作数：	0≤f≤127 d∈[0,1]
操作：	(f)+1→(目标寄存器)， 结果=0 则跳过
受影响的状态位：	无
说明：	将寄存器 f 的内容递增 1。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。如果结果为 1，则执行下一条指令。如果结果为 0，则转而执行 NOP 指令，从而使该指令成为双周期指令。

IORWI 立即数和 W 作逻辑或运算

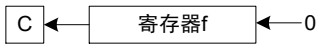
语法：[标号] IORWI k
操作数：0≤k≤255
操作：(W).OR.k→(W)
受影响的状态位：Z
说明：将 W 寄存器的内容与 8 位立即数 k 进行逻辑或运算。结果存入 W 寄存器。

IORWR W 和 f 作逻辑或运算

语法：[标号] IORWR f,d
操作数：0≤f≤127
d∈[0,1]
操作：(W).OR.(f)→(目标寄存器)
受影响的状态位：Z
说明：将 W 寄存器的内容与寄存器 f 的内容进行逻辑或运算。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。

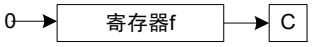
LSLF 逻辑左移

语法：[标号] LSLF f {,d}
操作数：0≤f≤127
d∈[0,1]
操作：(f[7])→C
(f[6:0])→目标寄存器[7:1]
0→目标寄存器[0]
受影响的状态位：C 和 Z
说明：将寄存器 f 的内容连同进位标志位一起左移 1 位。0 移入 LSb。如果 d 为 0，结果存入 W。如果 d 为 1，结果存回寄存器 f。



LSRF 逻辑右移

语法：[标号] LSRF f {,d}
操作数：0≤f≤127
d∈[0,1]
操作：0→目标寄存器[7]
(f[7:1])→目标寄存器[6:0]
(f[0])→C
受影响的状态位：C 和 Z
说明：将寄存器 f 的内容连同进位标志位一起右移 1 位。0 移入 MSb。如果 d 为 0，结果存入 W。如果 d 为 1，结果存回寄存器 f。



LDR 传送 f

语法：[标号] LDR f,d
操作数：0≤f≤127
d∈[0,1]
操作：(f)→(目标寄存器)
受影响的状态位：Z
说明：根据 d 的状态，将寄存器 f 的内容传送到目标寄存器。如果 d=0，目标寄存器为 W 寄存器。如果 d=1，目标寄存器为文件寄存器 f。由于状态标志位 Z 要受影响，可用 d=1 检测文件寄存器。
指令字数：1
指令周期数：1

示例：LDR FSR, 0
执行指令后
W=FSR 寄存器的值
Z=1

MOVIW 将 INDFn 的内容传送到 W

语法： [标号] MOVIW ++FSRn
[标号] MOVIW --FSRn
[标号] MOVIW FSRn++
[标号] MOVIW FSRn--
[标号] MOVIW k[FSRn]

操作数： n∈[0,1]
mm∈[00,01,10,11]
-32≤k≤31

操作： INDFn→W
有效地址通过以下方式确定：
•FSR+1 (预递增 1)
•FSR-1 (预递减 1)
•FSR+k (相对偏移)
执行传送指令后，FSR 值为以下任一项：
•FSR+1 (所有值都加 1)
•FSR-1 (所有值都减 1)
•不变

受影响的状态位：Z

模式	语法	mm
预递增	++FSRn	00
与递减	--FSRn	01
后递增	FSRn++	10
后递减	FSRn--	11

说明： 该指令用于在 W 寄存器和任一间接寄存器 (INDFn)之间传送数据。执行该传送指令之前/之后，将通过预/后增/减来更新指针 (FSRn)。

注：INDFn 寄存器不是物理寄存器。访问 INDFn 寄存器的指令实际上访问的是由 FSRn 指定的地址处的寄存器。

FSRn 地址范围限制为 0000h-FFFFh。地址递增/递减到超出边界时，将导致它发生折回。

MOVWI 将 W 的内容传送到 INDFn

语法： [标号] MOVWI ++FSRn
[标号] MOVWI --FSRn
[标号] MOVWI FSRn++
[标号] MOVWI FSRn--
[标号] MOVWI k[FSRn]

操作数： n∈[0,1]
mm∈[00,01,10,11]
-32≤k≤31

操作： W→INDFn
有效的地址由以下项决定:
•FSR+1 (预递增 1)
•FSR-1 (预递减 1)
•FSR+k (相对偏移)
执行传送指令后，FSR 值为以下任一项：
•FSR+1 (所有值都加 1)
•FSR-1 (所有值都减 1)
不变

受影响的状态位：无

模式	语法	mm
预递增	++FSRn	00
与递减	--FSRn	01
后递增	FSRn++	10
后递减	FSRn--	11

说明： 该指令用于在 W 寄存器和任一个间接寄存器 (INDFn) 之间传送数据。执行该传送指令之前/之后，将通过预/后增/减来更新指针 (FSRn)。

注：INDFn 寄存器不是物理寄存器。访问 INDFn 寄存器的指令实际上访问的是由 FSRn 指定的地址处的寄存器。

FSRn 地址范围限制为 0000h-FFFFh。地址递增/递减到超出边界时，将导致它发生折回。
对于 FSRn 的递增/递减操作不会影响任何状态位。

MOVLB 将立即数传送到 BSR

语法： [标号] MOVLB k
操作数： 0≤k≤15
操作： k→BSR
受影响的状态位： 无
说明： 将 5 位立即数 k 装入存储区选择寄存器 (BSR)。

MOVLP 将立即数传送到 PCLATH

语法： [标号] MOVLP k
操作数： 0≤k≤127
操作： k→PCLATH
受影响的状态位： 无
说明： 将 7 位立即数 k 装入 PCLATH 寄存器。

LDWI 将立即数传送到 W

语法： [标号] MOVLW k
操作数： 0≤k≤255
操作： k→(W)
受影响的状态位： 无
说明： 将 8 位立即数 k 装入 W 寄存器。其余无关位均汇编为 0。
指令字数： 1
指令周期数： 1
示例： LDWI 0x5A
执行指令后
W = 0x5A

NOP 空操作

语法： [标号] NOP
操作数： 无
操作： 空操作
受影响的状态位： 无
说明： 不执行任何操作。
指令字数： 1
指令周期数： 1
示例： NOP

STR 将 W 的内容传送到 f

语法： [标号] STR f
操作数： 0≤f≤127
操作： (W) →(f)
受影响的状态位： 无
说明： 将 W 寄存器的数据传送到寄存器 f。
指令字数： 1
指令周期数： 1
示例： STR OPTION
执行指令前
OPTION = 0xFF
W = 0x4F
执行指令后
OPTION = 0x4F
W = 0x4F

RESET 软件复位

语法： [标号] RESET
操作数： 无
操作： 执行器件复位。复位 PCON 寄存器的 nRI 标志。
受影响的状态位： 无
说明： 此指令可实现用软件执行硬件复位。

RET 从子程序返回

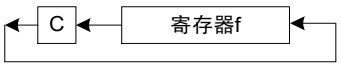
语法： [标号] RET
操作数： 无
操作： TOS→PC
受影响的状态位： 无
说明： 从子程序返回。执行出栈操作，将栈顶 (TOS) 内容装入程序计数器。这是一条双周期指令。

RETI 从中断返回

语法： [标号] RETI
操作数： 无
操作： TOS→PC，
1→GIE
受影响的状态位：无
说明： 从中断返回。执行出栈操作，将栈顶 (Top-of-Stack，TOS) 的内容装入 PC。通过将全局中断允许位 GIE (INTCON[7]) 置 1，来允许中断。这是一条双周期指令。
指令字数： 1
指令周期数： 2
示例： RETI
中断后
PC = TOS
GIE = 1

RLR 对 f 执行带进位的循环左移

语法： [标号] RLR f,d
操作数： 0≤f≤127
d∈[0,1]
操作： 参见如下说明
受影响的状态位：C
说明： 将寄存器 f 的内容连同进位标志位一起循环左移 1 位。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。
指令字数： 1
指令周期数： 1
示例： RLF REG1,0
执行指令前:
REG1 = 1110 0110
C = 0
执行指令后:
REG1 = 1110 0110
W = 1100 1100
C = 1



RETW 返回并将立即数送入 W

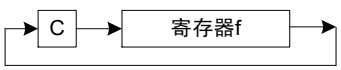
语法： [标号] RETW k
操作数： 0≤k≤255
操作： k→(W);
TOS→PC
受影响的状态位：无
说明： 将 8 位立即数 k 装入 W 寄存器。将栈顶内容 (返回地址) 装入程序计数器。这是一条双周期指令。
指令字数： 1
指令周期数： 2
示例：

```
        LCALL TABLE;W contains
        ;table offset value
        • ;W now has table value
        •
TABLE   •
        ADDWR   PC    ;W = offset
        RETW    k1    ;Begin table
        RETW    k2    ;
        •
        •
        •
        RETW    kn    ;End of table
```

执行指令前
W = 0x07
执行指令后
W = k8 的值

RRR 对 f 执行带进位的循环右移

语法： [标号] RRR f,d
操作数： 0≤f≤127
d∈[0,1]
操作： 参见如下说明
受影响的状态位：C
说明： 将寄存器 f 的内容连同进位标志位一起循环右移 1 位。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f



SLEEP 进入休眠模式

语法： [标号] SLEEP
操作数： 无
操作： 00h→WDT，
0→WDT 预分频器，
1→/TO，
0→/PD
受影响的状态位：/TO 和/PD
说明： 掉电状态位/PD 被清零。超时状态位/TO 被置 1。看门狗定时器及其预分频器被清零。振荡器停振，处理器进入休眠模式。

SWAPR 将 f 中的两个半字节交换

语法： [标号] SWAPR f,d
操作数： 0≤f≤127
d∈[0,1]
操作： (f[3:0])→(目标寄存器[7:4])，
(f[7:4])→(目标寄存器[3:0])
受影响的状态位：无
说明： 寄存器 f 的高半字节和低半字节相互交换。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。

SUBWI 从立即数中减去 W

语法： [标号] SUBWI k
操作数： 0≤k≤255
操作： k-(W)→(W)
受影响的状态位：C、DC 和 Z
说明： 用 8 位立即数 k 减去 W 寄存器的内容 (通过二进制补码方式进行运算) 。结果存入 W 寄存器。

C= 0	W>k
C= 1	W≤k
DC=0	W[3:0] > k[3:0]
DC=1	W[3:0] ≤ k[3:0]

SUBWR f 减去 W

语法： [标号] SUBWR f,d
操作数： 0≤f≤127
d∈[0,1]
操作： (f)-(W)→(目标寄存器)
受影响的状态位：C、DC 和 Z
说明： 用寄存器 f 的内容减去 W 寄存器的内容 (通过二进制补码方式进行运算)。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。

C= 0	W>f
C= 1	W≤f
DC=0	W[3:0]> > f[3:0]
DC=1	W[3:0] ≤ f[3:0]

SUBWFB f 减去 W (带借位)

语法： SUBWFB f {,d}
操作数： 0≤f≤127
d∈[0,1]
操作： (f)–(W)–(/B)→目标寄存器
受影响的状态位：C、DC 和 Z
说明： 用 f 寄存器的内容减去 W 的内容和借位标志 (进位) (通过二进制补码方式进行运算)。如果 d 为 0，结果存入 W。如果 d 为 1，结果存回寄存器 f。

XORWR W 和 f 作逻辑异或运算

语法： [标号] XORWR f,d
操作数： 0≤f≤127
d∈[0,1]
操作： (W).XOR.(f)→(目标寄存器)
受影响的状态位：Z
说明： 将 W 寄存器的内容与寄存器 f 的内容进行逻辑异或运算。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。

XORWI 立即数和 W 作逻辑异或运算

语法：	[标号] XORWI k
操作数：	0≤k≤255
操作：	(W).XOR.k→(W)
受影响的状态位：	Z
说明：	将 W 寄存器的内容与 8 位立即数 k 进行逻辑异或运算。结果存入 W 寄存器。

21. 芯片的电气特性

21.1. 极限参数

工作温度等级 3…… $^{-}40 – ^{+}85°C$
工作温度等级 2…… $^{-}40 – ^{+}105°C$
工作温度等级 1…… $^{-}40 – ^{+}125°C$
存储温度…… $^{-}40 – ^{+}125°C$
电源电压…… V_{SS}-0.3V – V_{SS}+6.0V
端口输入电压…… V_{SS}-0.3V – V_{DD}+0.3V

注：

1. 超过上述“极限参数”所规定的范围，可能会对芯片造成永久性损坏。
2. 除非另作说明，所有特性值的测试条件为 25°C, V_{DD} =1.9 – 5.5V。
3. 本节所示的值和范围基于特性值，并非最终出货的标准值。除非另作说明，生产测试温度为 25°C。

21.2. 工作特性

电气参数		最小值	典型值	最大值	单位	条件/备注
Fsys(系统时钟频率)	2T/4T	—	—	8	MHz	-40~85°C, VDD = 1.9~5.5V
		—	—	16	MHz	-40~85°C, VDD = 2.7~5.5V
指令周期 (Tins)	2T	—	125	—	ns	系统时钟 HIRC
	4T	—	250	—	ns	
	2T	—	61	—	μs	系统时钟 LIRC
	4T	—	122	—	μs	
上电复位保持时间 (T_{DRH})		—	4.2	—	ms	25°C, PWRT disable
外部复位脉冲宽度 (T_{MCLRB})		2000	—	—	ns	25°C
WDT 周期 (T_{WDT})		—	1	—	ms	无预分频，WDTPS[3:0]=0000

21.3. POR, LVR, LVD

上电复位 (POR)

电气参数	最小值	典型值	最大值	单位	条件/备注
I_{POR} 工作电流	—	140	—	nA	25°C, V_{DD} = 3.3V
V_{POR}	—	1.65	—	V	25°C

低电压复位 (LVR)

电气参数	最小值	典型值	最大值	单位	条件/备注
I_{LVR} 工作电流	—	15.7	—	μA	25°C, V_{DD} = 3.3V
V_{LVR}，LVR 阈值	1.94	2.0	2.06	V	25°C
	2.13	2.2	2.27		
	2.42	2.5	2.58		
	2.72	2.8	2.88		
	3.01	3.1	3.19		
	3.49	3.6	3.71		
	3.98	4.1	4.22		
LVR delay	—	125	157	μs	25°C, V_{DD} = 2.5V

低电压检测 (LVD)

电气参数	最小值	典型值	最大值	单位	条件/备注
I_{LVD} 工作电流	—	22.5	—	μA	25°C, V_{DD} = 3.3V
V_{LVD}，LVD 阈值	1.94	2.0	2.06	V	25°C
	2.33	2.4	2.47		
	2.72	2.8	2.88		
	2.91	3.0	3.09		
	3.49	3.6	3.71		
	3.88	4.0	4.12		
LVD delay	—	125	157	μs	25°C, V_{DD} = 2.5V

21.4. I/O 端口电路

电气参数		最小值	典型值	最大值	单位	条件/备注
V_{IL}		0	—	0.3* V_{DD}	V	
V_{IH}		0.7* V_{DD}	—	V_{DD}	V	
漏电流		-1	—	1	μA	V_{DD} = 5V
源电流(source)	L0	—	-4	—	mA	25°C, V_{DD} = 5V, V_{OH} = 4.5V
	L1	—	-8	—		
	L2	—	-26	—		
灌电流(sink)	L0	—	53	—	mA	25°C, V_{DD} = 5V, V_{OL}= 0.5V
	L1	—	62	—		
上拉电阻		—	20	—	kΩ	
下拉电阻		—	20	—	kΩ	

21.5. 工作电流（I_{DD}）

电气参数	Sysclk	典型值@V_{DD}			单位
		2.0V	3.0V	5.5V	
正常模式(2T)，I_{DD}	16MHz	—	2.181	2.286	mA
	8MHz	1.078	1.522	1.592	
	4MHz	0.826	1.178	1.214	
	2MHz	0.624	0.716	0.719	
	1MHz	0.400	0.460	0.461	
	500kHz	0.294	0.361	0.369	
	250kHz	0.244	0.317	0.327	
	125kHz	0.219	0.295	0.306	
	32kHz	0.034	0.046	0.048	
休眠模式（Sleep, ~~WDT OFF~~, ~~LVR OFF~~），I_{SB}	—	0.211	0.274	0.423	μA
休眠模式（Sleep, WDT ON, ~~LVR OFF~~）	—	1.383	2.499	3.062	
休眠模式（Sleep, ~~WDT OFF~~, LVR ON）	—	11.618	15.951	21.956	
休眠模式（Sleep, WDT ON, LVR ON）	—	12.793	18.156	24.469	
休眠模式（Sleep, ~~WDT OFF~~, ~~LVR OFF~~, LVD ON）	—	18.621	22.774	28.738	

注：

1. 睡眠电流的测试条件为 I/O 处于输入模式并外部下拉到 0；

21.6. 内部振荡器

内部低频振荡器 (LIRC)

低频振荡器有双模模式，一种模式下振动频率为 32kHz，另一种模式下振动频率为 256kHz。振荡频率模式由 OSCCON 寄存器中的 LFMOD 位控制，0 为 32kHz 模式，1 为 256kHz 模式。

电气参数	最小值	典型值	最大值	单位	条件/备注
振荡频率	30.4	32	33.6	kHz	25°C, V_{DD} = 2.5V
随温度变化范围	-2.5%	—	2.0%	—	-40 ~ 85°C, V_{DD} = 2.5V
随电源电压变化范围	-4.0%	—	1.0%	—	25°C, V_{DD} = 1.9 ~ 5.5V
I_{LIRC} 工作电流	—	1.3	—	μA	25°C, V_{DD} = 3.0V
启动时间	—	4.6	—	μs	25°C, V_{DD} = 3.0V

内部高频振荡器(HIRC)

电气参数	最小值	典型值	最大值	单位	条件/备注
校准范围	15.84	16	16.16	MHz	25°C, V_{DD} = 2.5V
随温度变化范围	-2.0%	—	2.0%	—	-40 ~ 85°C, V_{DD} = 2.5V
随电源电压变化范围	-0.5%	—	0.5%	—	25°C, V_{DD} = 1.9 ~ 5.5V
I_{HIRC} 工作电流	—	40	—	μA	25°C, V_{DD} = 3.0V
启动时间	—	2.5	—	μs	25°C, V_{DD} = 3.0V

21.7. ADC(12bit)和 ADC VREF

ADC (12bit)

电气参数	最小值	典型值	最大值	单位	条件/备注
ADC 工作电压 V_{DD}	2.7	—	5.5	V	
ADC 工作电流 I_{VDD}	—	90	—	μA	25°C, V_{REFP} = V_{DD} = 2.7V, ADC 转换时钟频率为 250kHz
	—	100	—	μA	25°C, V_{REFP} = V_{DD} = 3.0V, ADC 转换时钟频率为 250kHz
	—	140	—	μA	25°C, V_{REFP} = V_{DD} = 5.5V, ADC 转换时钟频率为 250kHz
模拟输入电压 V_{AIN}	VREFN	—	VREFP	V	
外部参考电压 V_{REF}	—	—	V_{DD}	V	
分辨率	—	—	12	位	
积分误差 E_{IL}	—	±2	—	LSB	25°C, V_{REFP} = V_{DD} = 5.0V, V_{REFN} = GND, ADC 转换时钟频率为 250kHz
微分误差 E_{DL}	—	±2	—	LSB	
偏移误差 E_{OFF}	—	±3	—	LSB	25°C, V_{REFP} = V_{DD} = 5.0V, V_{REFN} = GND, ADC 转换时钟频率为 250kHz
增益误差 E_{GN}	—	±5	—	LSB	
转换时钟周期 T_{AD}	—	2	—	μs	V_{REFP} > 3.0V, V_{DD} > 3.0V
转换时钟数	—	15	—	T_{AD}	
稳定时间 (T_{ST})	—	15	—	μs	
采样时间 (T_{ACQ})	—	1.5	—	T_{AD}	
建议的模拟电压源阻抗(ZAI)	—	—	10	kΩ	

ADC V_{REF}

电气参数	最小值	典型值	最大值	单位	条件/备注
内置参考电压 ADCVref	0.492	0.5	0.508	V	25°C, V_{DD} = 5V
	1.992	2.0	2.008	V	25°C, V_{DD} = 5V
	2.988	3.0	3.012	V	25°C, V_{DD} = 5V
内置参考电压 0.5V 稳定时间 T_{VRINT}	—	400	—	μs	25°C, V_{DD} = 5V
	—	600	—	μs	25°C, V_{DD} = 5V, 1μF
内置参考电压 2.0V 稳定时间 T_{VRINT}	—	450	—	μs	25°C, V_{DD} = 5V
	—	800	—	μs	25°C, V_{DD} = 5V, 1μF
内置参考电压 3.0V 稳定时间 T_{VRINT}	—	450	—	μs	25°C, V_{DD} = 5V
	—	1200	—	μs	25°C, V_{DD} = 5V, 1μF

21.8. Program 和 Data EEPROM

参数		最小值	典型值	最大值	单位	条件
$V_{DD-READ}$	Program/Data EE 读电压	V_{POR}	—	5.5	V	-40 – 85 / 105°C
$V_{DD-WRITE}$	Program EE 写电压	2.5	—	5.5	V	-40 – 85 / 105°C
	Data EE 写电压	1.9	—	5.5		
N_{END}	Program EE 擦/写次数	100 k	—	—	cycle	25 °C
		40 k	—	—		85 °C
		10 k	—	—		105 °C
	Data EE 擦/写次数	1,000 k	—	—		25 °C
		400 k	—	—		85 °C
		100 k	—	—		105 °C
T_{RET}	Program EE 数据保持	20	—	—	年	1k 次擦写后 @ 85 °C
		10	—	—		1k 次擦写后 @ 105 °C
	Data EE 数据保持	20	—	—		10k 次擦写后 @ 85 °C
		10	—	—		10k 次擦写后 @ 105 °C
T_{WRITE}	Program EE 写时间	0.7	—	1.2	ms	—
	Data EE 写时间	3.0	—	5.0		—
I_{PROG}	Data EE 编程电流	—	—	323	μA	25 °C, V_{DD} = 3 V

21.9. EMC 特性

ESD

参数		最小值	典型值	最大值	单位	条件
V_{ESD}	HBM	7000	–	–	V	MIL-STD-883H Method 3015.8
V_{ESD}	MM	400	–	–	V	JESD22-A115

Latch-up

参数	最小值	典型值	最大值	单位	条件
LU, static latch-up	200	–	–	mA	EIA/JESD 78

EFT

参数	最小值	典型值	最大值	单位	条件
V_{EFT}	5.5	–	–	kV	V_{DD} (5V) 与 GND 间的电容：1μF

21.10. 特性图

注意：本节提供的图表基于特性值，仅用作设计参考，未经生产测试。

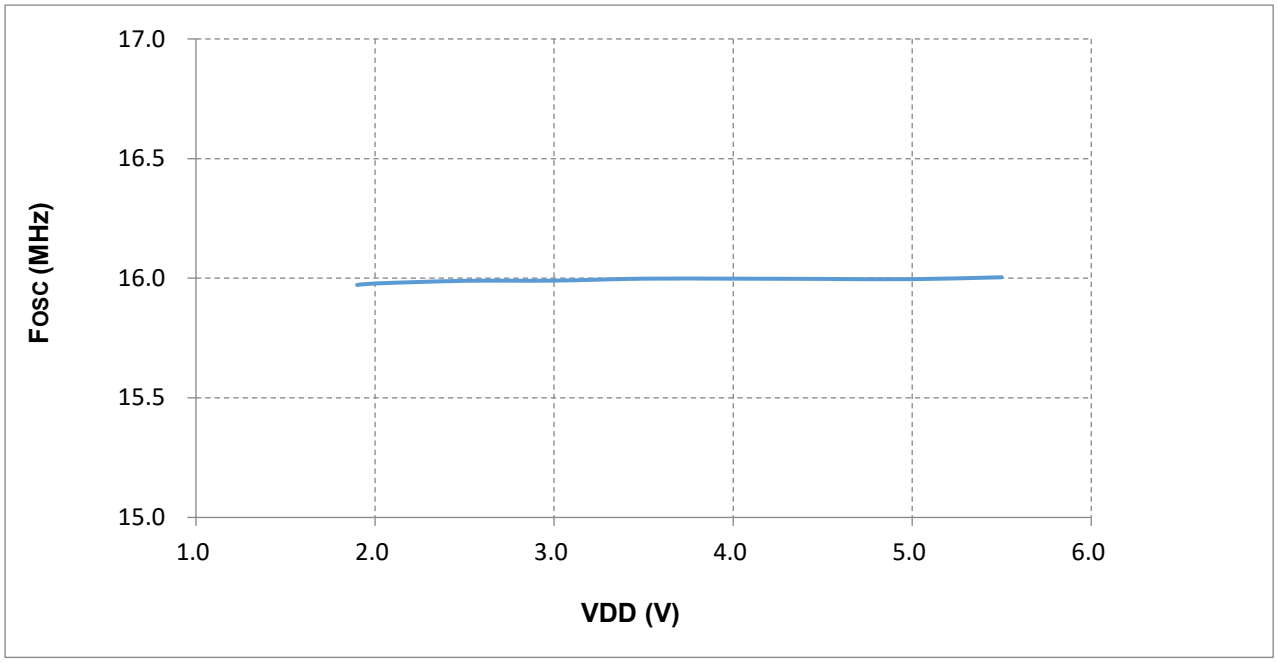


图 21-1 HIRC vs.V_{DD} (T_A = 25°C)

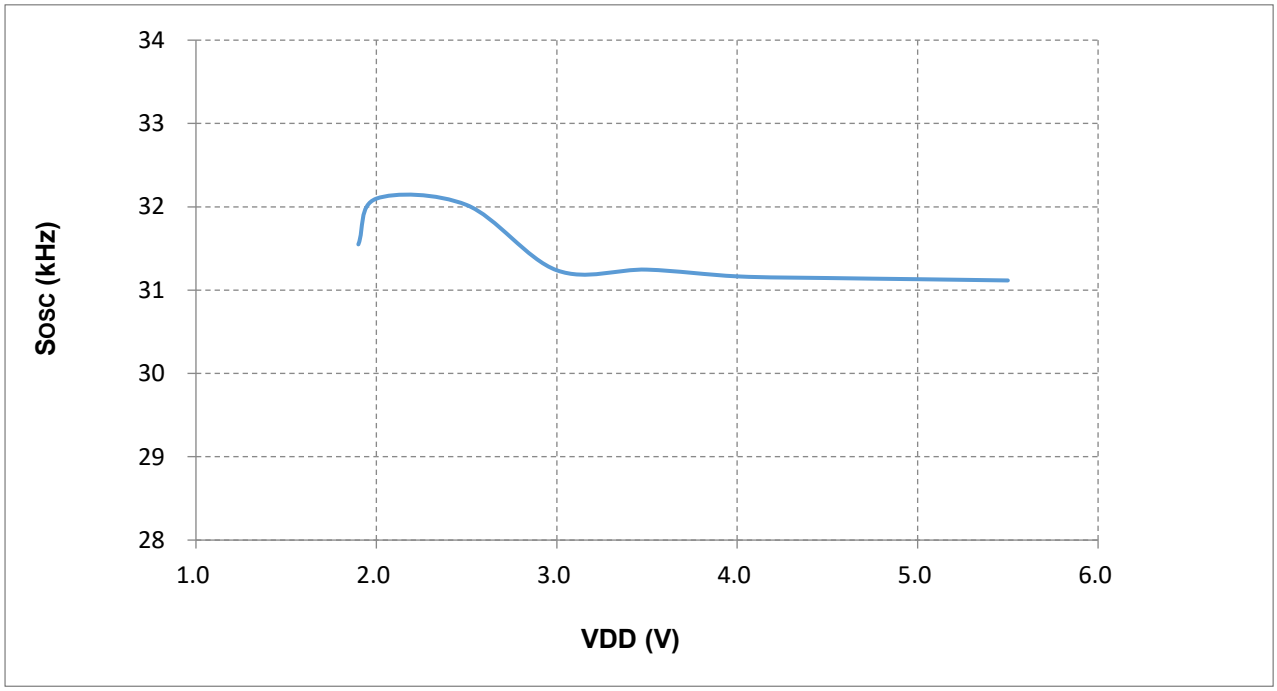


图 21-2 LIRC vs. V_{DD} (T_A = 25°C)

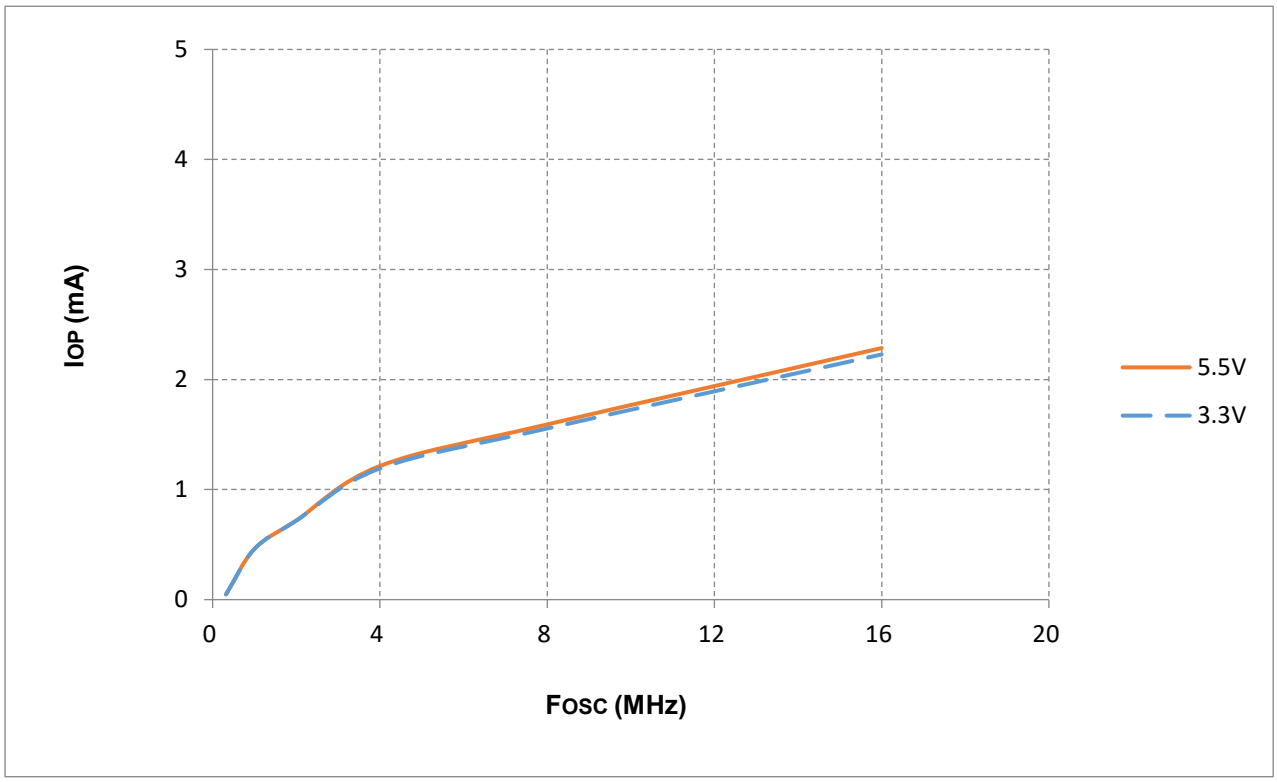


图 21-3 I_{DD} vs Frequency (2T, T_A=25°C)

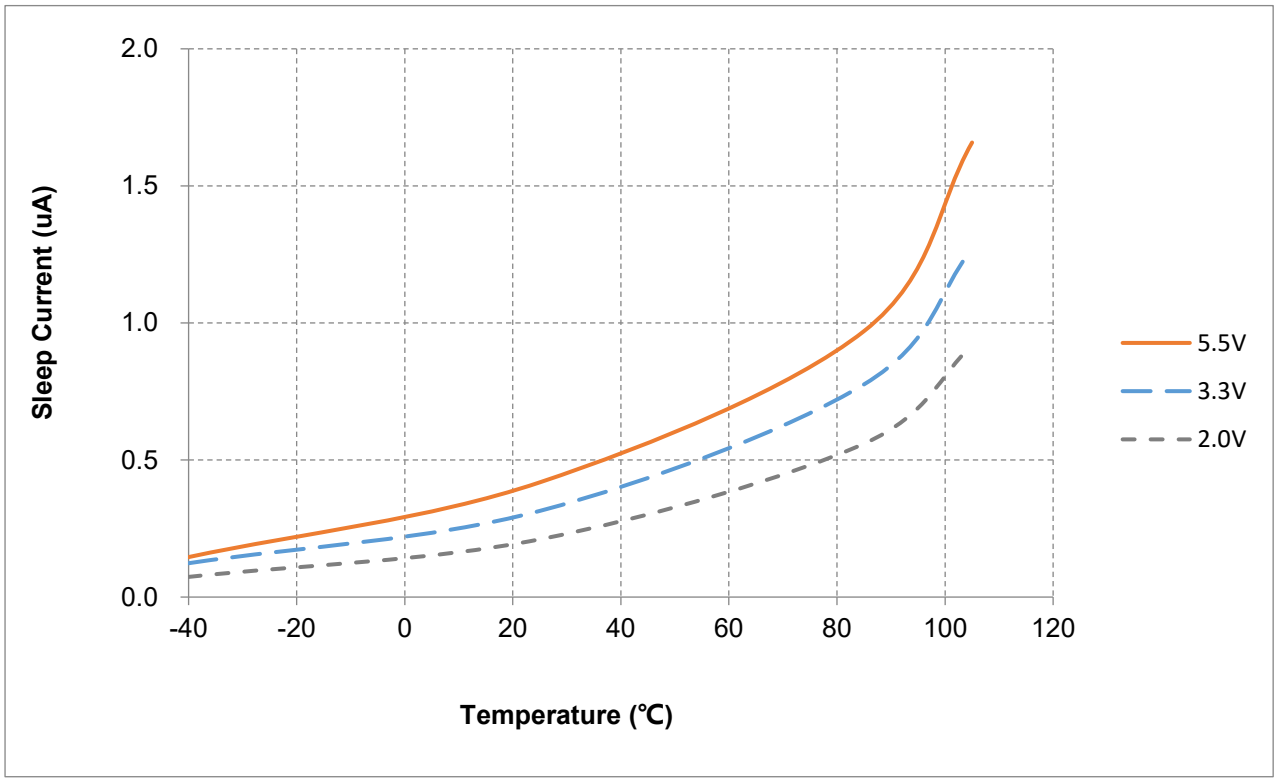


图 21-4 Sleep Current (I_{SB}) vs. Temperature

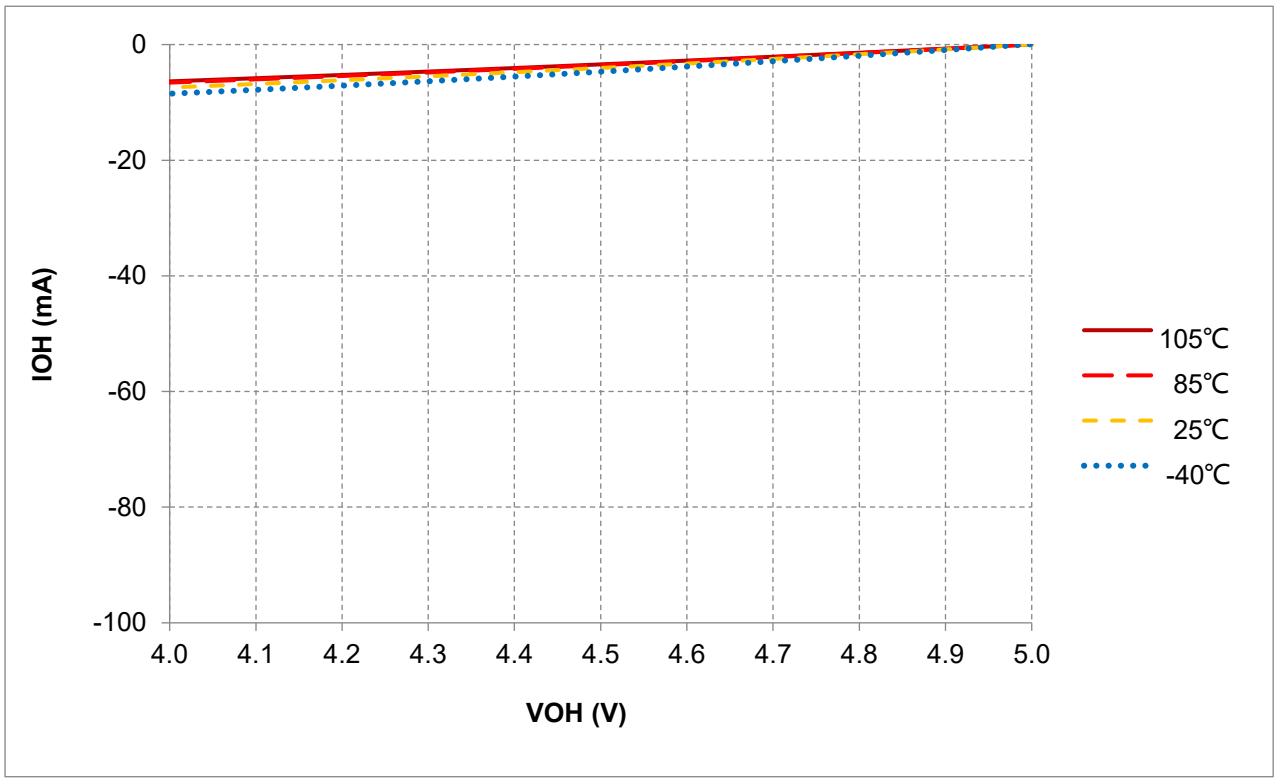


图 21-5 I_{OH} vs V_{OH} @L0 = -4mA , V_{DD} = 5V

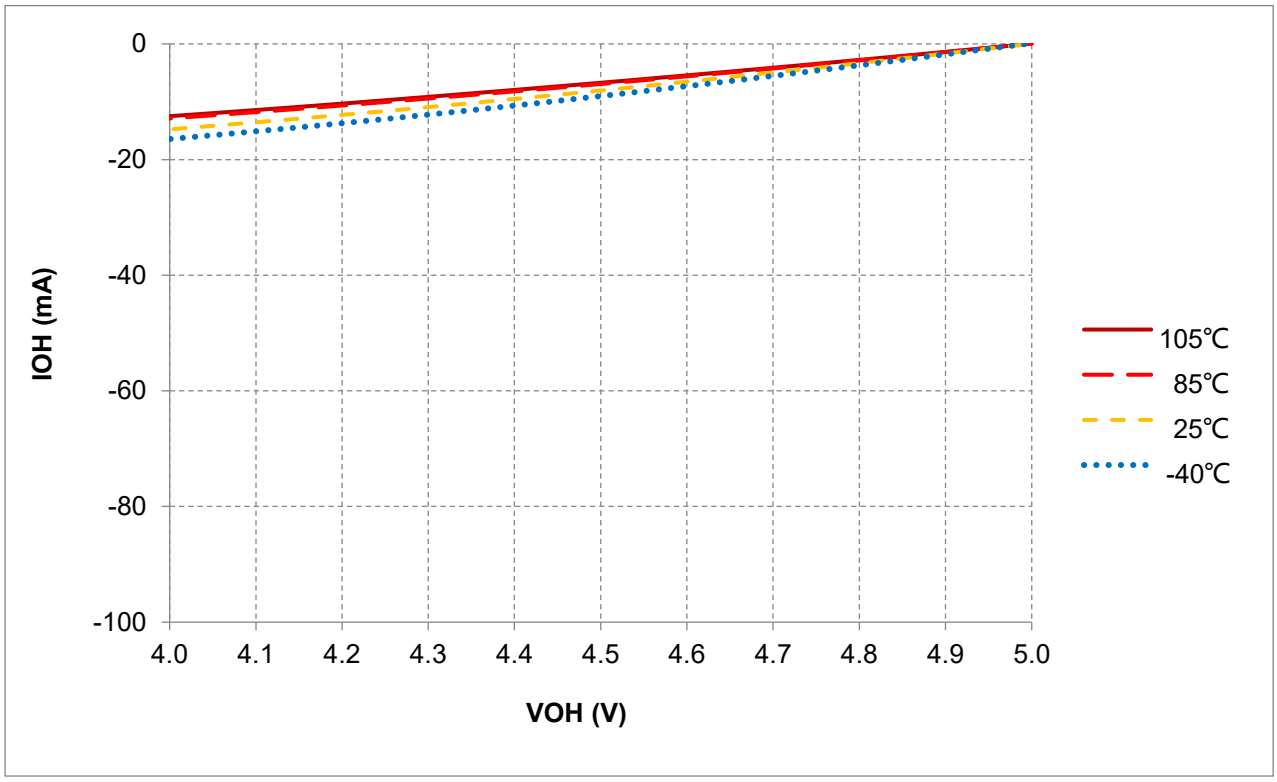


图 21-6 I_{OH} vs V_{OH} @L1 = -8mA , V_{DD} = 5V

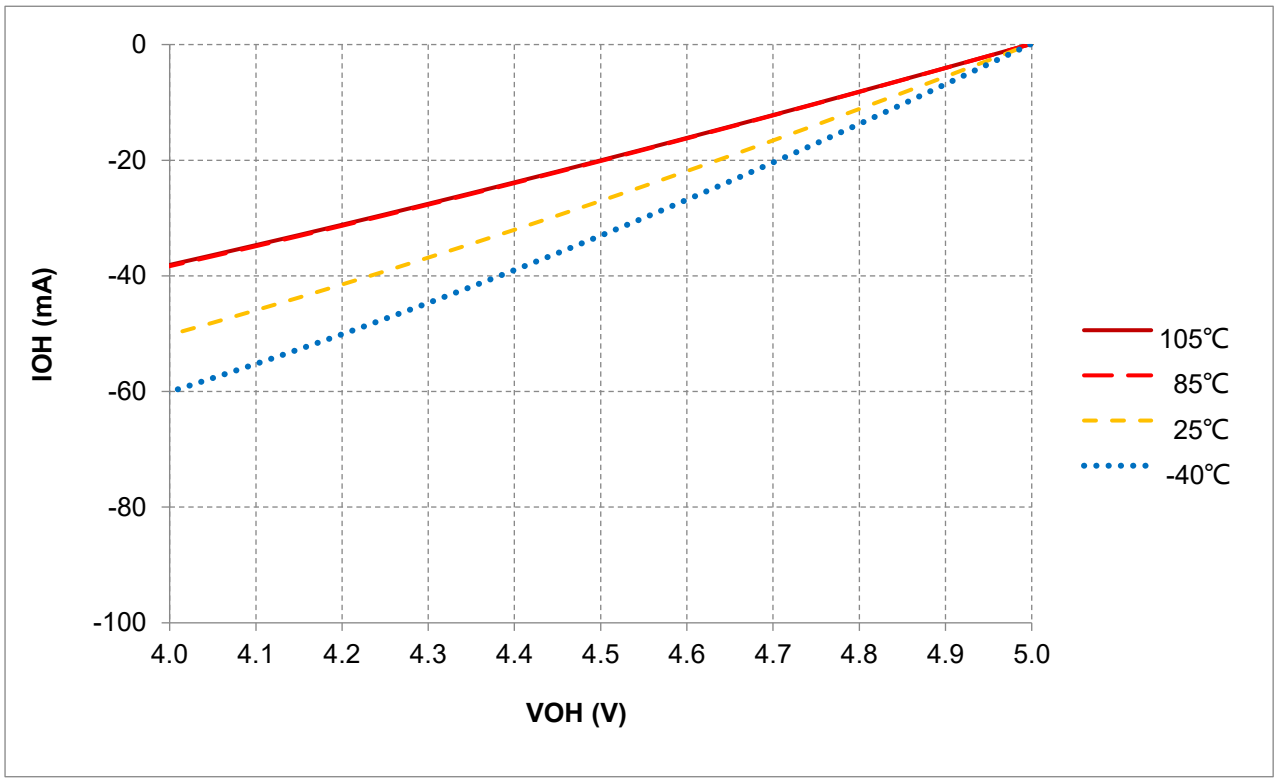


图 21-7 I_{OH} vs V_{OH} @L2 = -26mA , V_{DD} = 5V

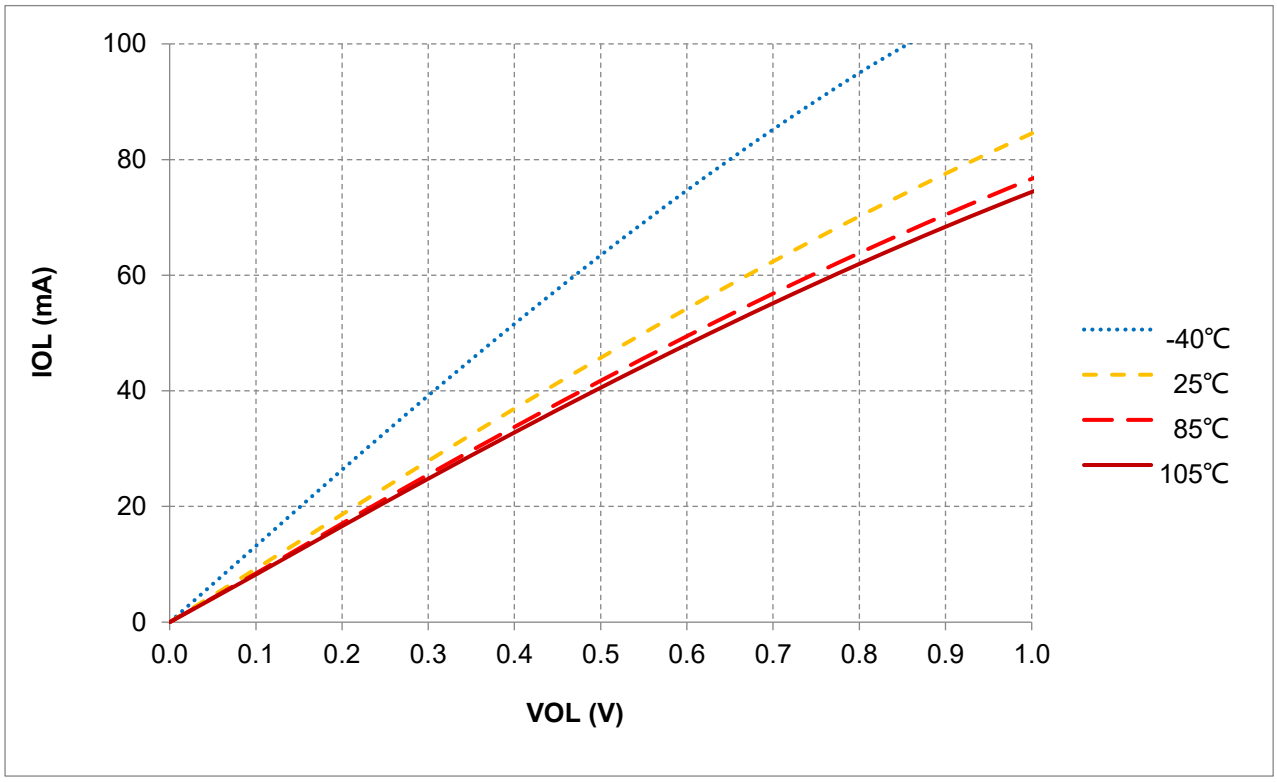


图 21-8 I_{OH} vs V_{OL} @L0 = 53mA , V_{DD} = 5V

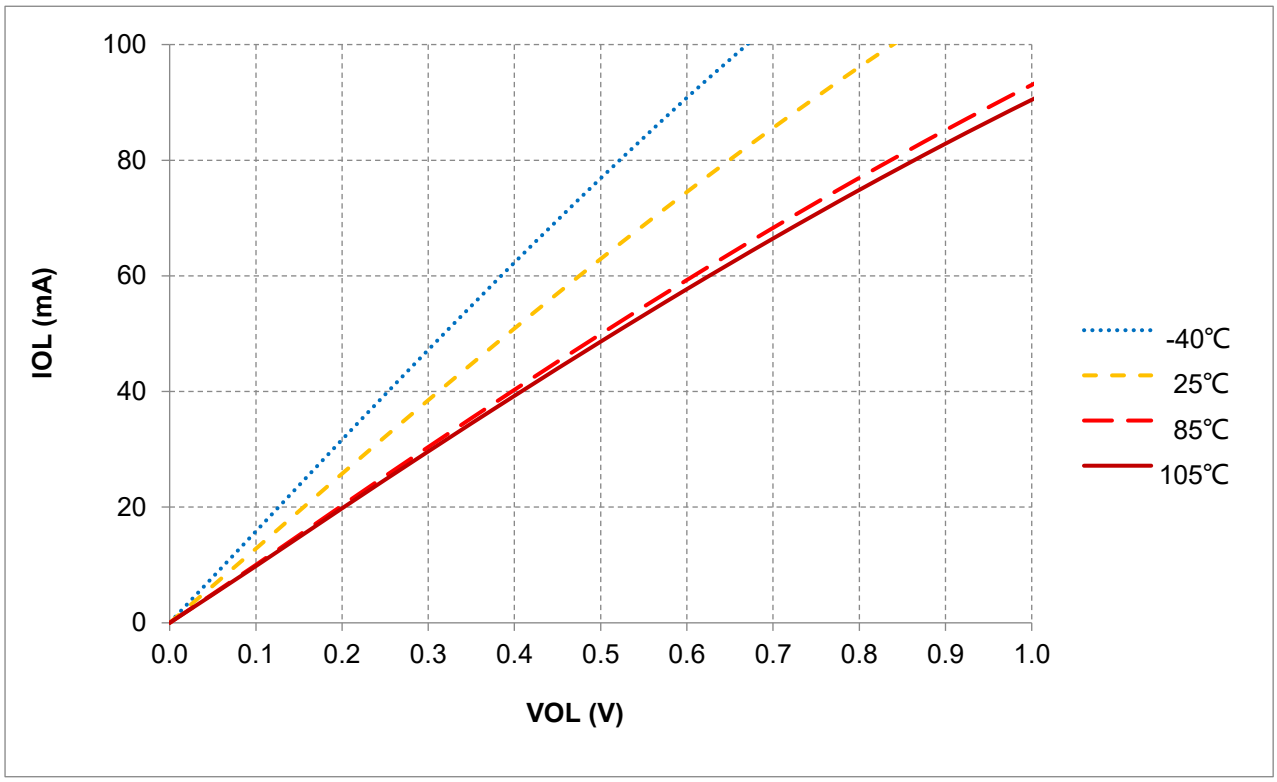
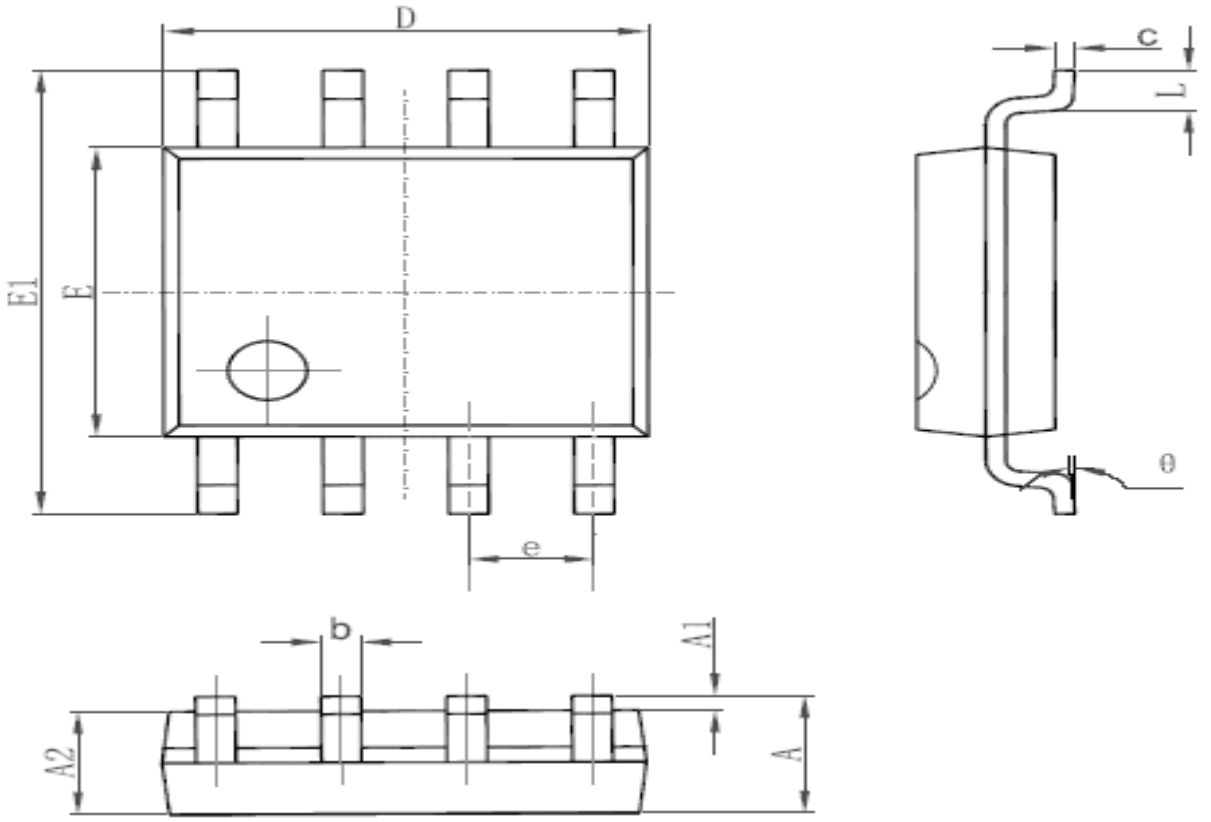


图 21-9 I_{OL} vs V_{OL} @L1 = 62mA , V_{DD} = 5V

22. 芯片封装信息

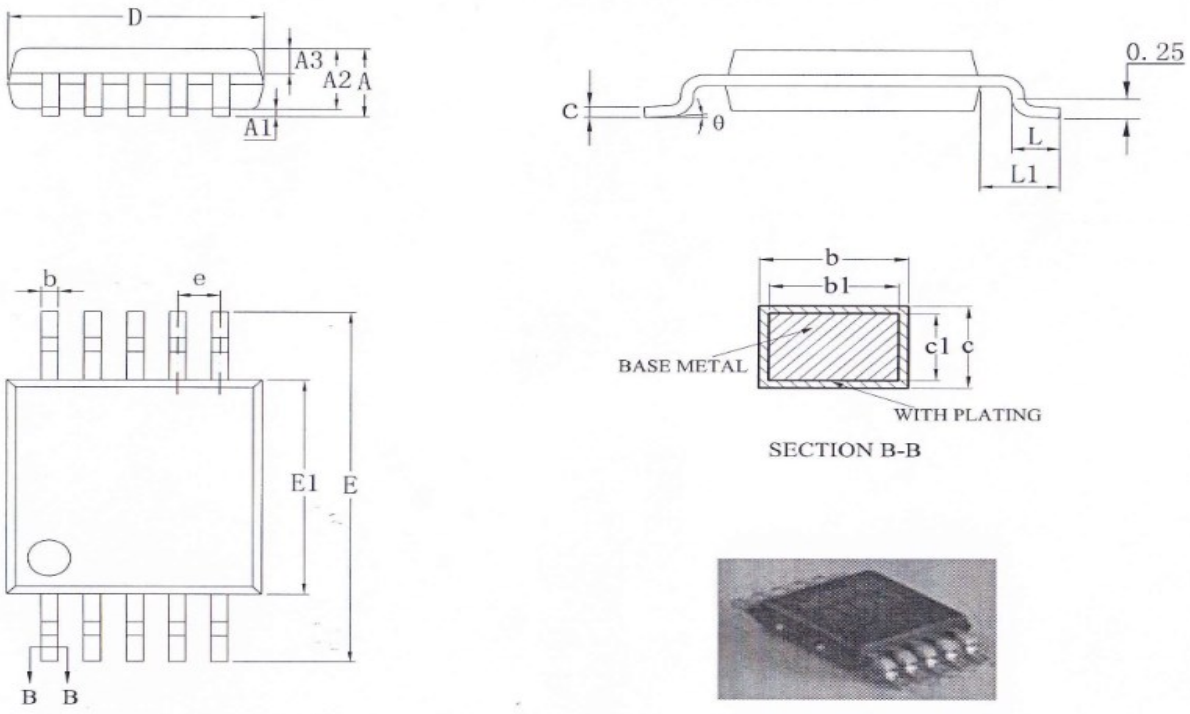
本芯片采用 SOP8、MSOP10、SOP14、SOP16、SOP20、QFN20 和 TSSOP20 封装方式，具体封装尺寸信息如下。

SOP8



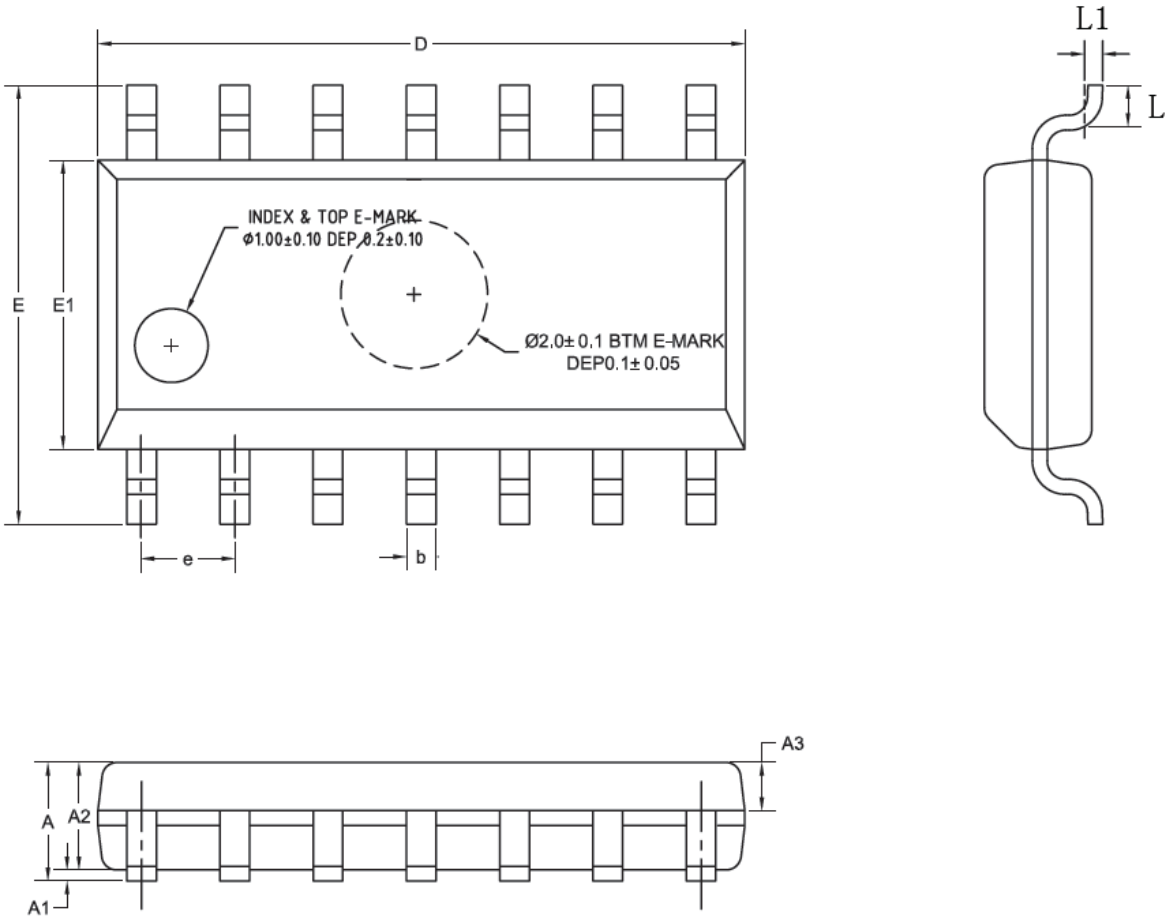
Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	1.350	1.750	0.053	0.069
A1	0.100	0.250	0.004	0.010
A2	1.350	1.550	0.053	0.061
b	0.330	0.510	0.013	0.020
c	0.170	0.250	0.006	0.010
D	4.700	5.100	0.185	0.200
E	3.800	4.000	0.150	0.157
E1	5.800	6.200	0.228	0.244
e	1.270 (BSC)		0.050 (BSC)	
L	0.400	1.270	0.016	0.050
θ	0°	8°	0°	8°

MSOP10



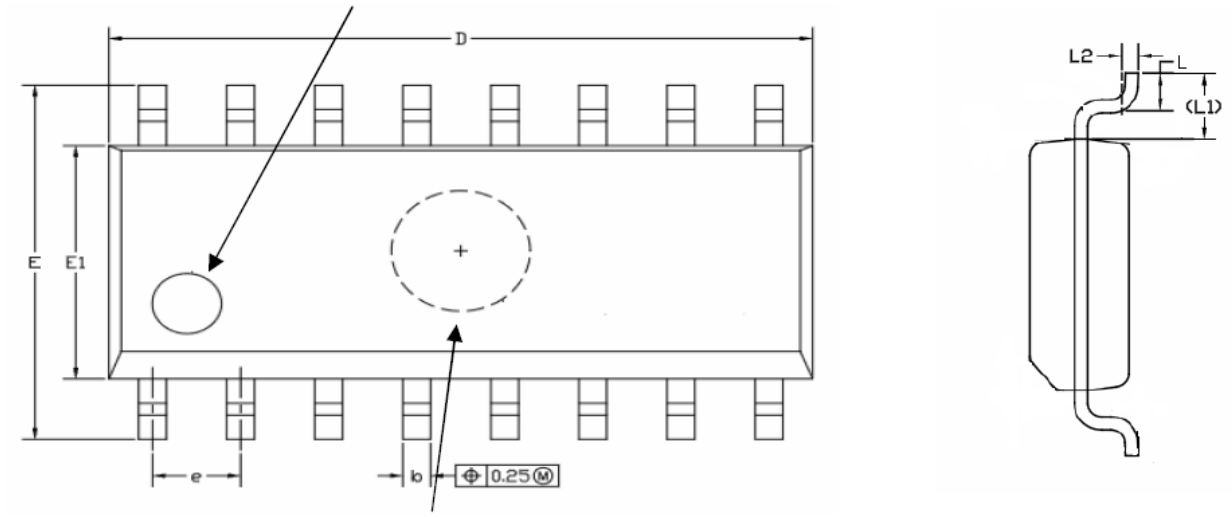
Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	-	1.100	-	0.043
A1	0.050	0.150	0.002	0.006
A2	0.750	0.950	0.030	0.037
A3	0.300	0.400	0.012	0.016
b	0.180	0.260	0.007	0.010
b1	0.170	0.230	0.007	0.009
c	0.150	0.190	0.006	0.007
c1	0.140	0.160	0.006	0.006
D	2.900	3.100	0.114	0.122
E	4.700	5.100	0.185	0.201
E1	2.900	3.100	0.114	0.122
e	0.500(BSC)		0.020(BSC)	
L	0.400	0.700	0.016	0.028
L1	0.950(REF)		0.037(REF)	
θ	0	8°	0	8°

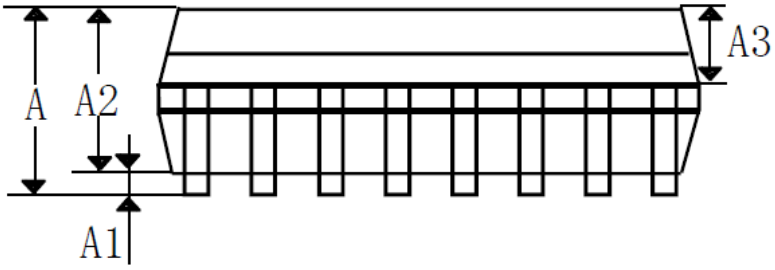
SOP14



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	-	1.700	-	0.066
A1	0.100	0.200	0.004	0.008
A2	1.400	1.500	0.055	0.059
A3	0.620	0.680	0.024	0.027
b	0.370	0.420	0.015	0.016
D	8.710	8.910	0.343	0.347
E	5.900	6.100	0.232	0.238
E1	3.800	3.950	0.150	0.156
e	1.270(BSC)		0.050(BSC)	
L	0.500	0.700	0.020	0.027
L1	0.250(BSC)		0.010(BSC)	

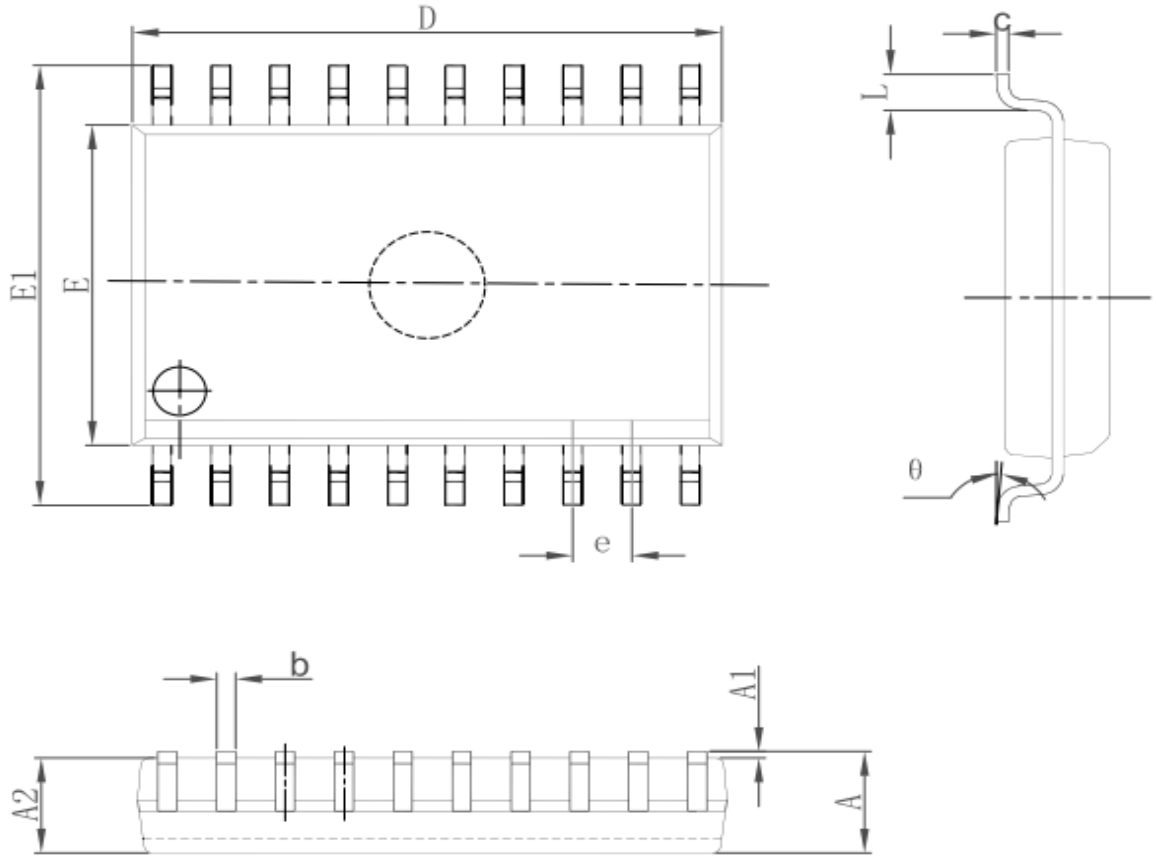
SOP16





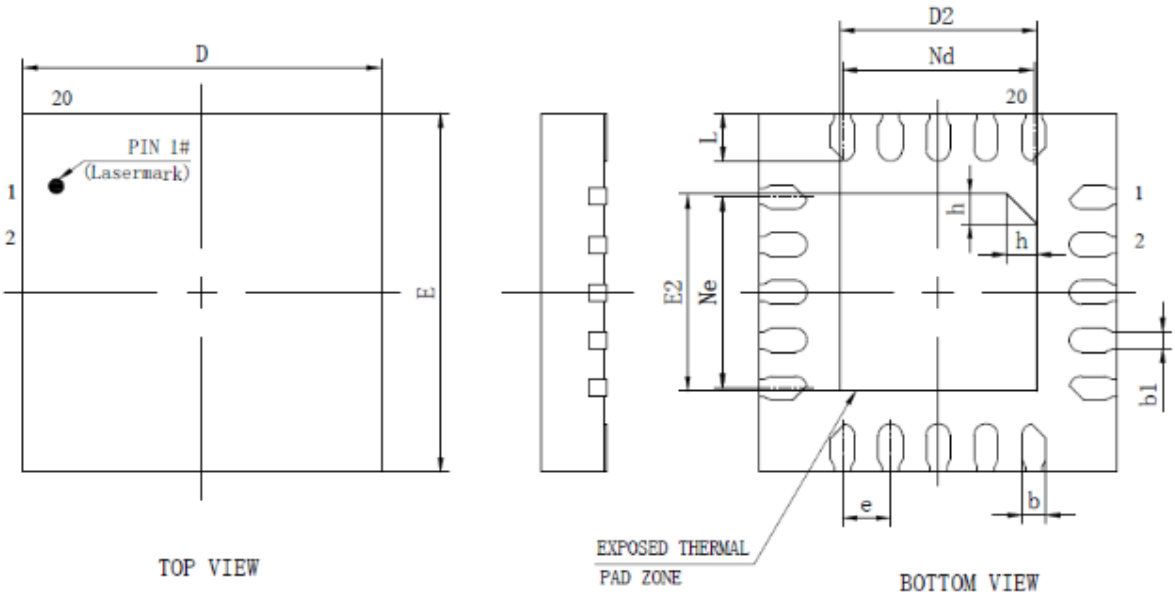
Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	-	1.700	-	0.066
A1	0.100	0.200	0.004	0.008
A2	1.420	1.480	0.056	0.058
A3	0.620	0.680	0.024	0.027
D	9.960	10.160	0.392	0.396
E	5.900	6.100	0.232	0.238
E1	3.870	3.930	0.152	0.153
b	0.370	0.430	0.015	0.017
e	1.240	1.300	0.048	0.051
L	0.500	0.700	0.020	0.027
L1	1.050(REF)		0.041(REF)	
L2	0.250(BSC)		0.010(BSC)	

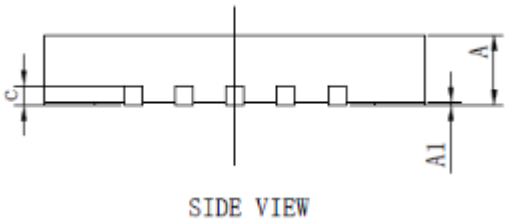
SOP20



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	2.350	2.650	0.093	0.104
A1	0.100	0.300	0.004	0.012
A2	2.100	2.500	0.083	0.098
b	0.330	0.510	0.013	0.020
c	0.204	0.330	0.008	0.013
D	12.520	13.000	0.493	0.512
E	7.400	7.600	0.291	0.299
E1	10.210	10.610	0.402	0.418
e	1.270 (BSC)		0.050 (BSC)	
L	0.400	1.270	0.016	0.050
θ	0°	8°	0°	8°

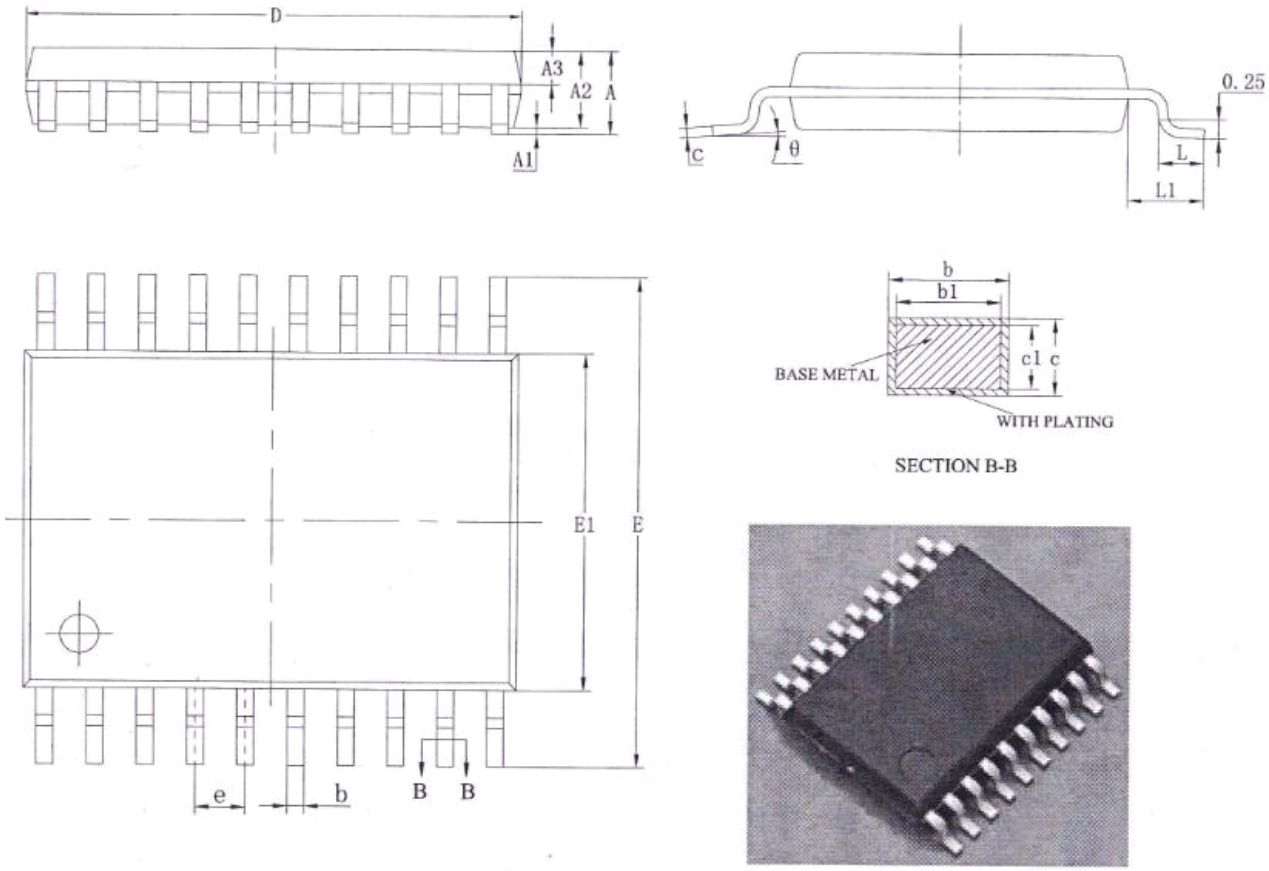
QFN20





Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	0.500	0.600	0.020	0.024
A1	-	0.050	-	0.002
b	0.150	0.250	0.006	0.010
b1	0.140 (REF)		0.006 (REF)	
c	0.100	0.200	0.004	0.008
D	2.900	3.100	0.114	0.122
D2	1.550	1.750	0.061	0.069
e	0.400 (BSC)		0.016 (BSC)	
Ne	1.600 (BSC)		0.063 (BSC)	
Nd	1.600 (BSC)		0.063 (BSC)	
E	2.900	3.100	0.114	0.122
E2	1.550	1.750	0.061	0.069
L	0.350	0.450	0.014	0.018
h	0.200	0.300	0.008	0.012

TSSOP20



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	-	1.20	-	0.472
A1	0.05	0.15	0.020	0.059
A2	0.80	1.05	0.315	0.413
A3	0.39	0.49	0.154	0.193
b	0.20	0.28	0.079	0.110
b1	0.19	0.25	0.075	0.098
c	0.13	0.17	0.051	0.067
c1	0.12	0.14	0.047	0.055
D	6.40	6.60	2.520	2.598
E1	4.30	4.50	1.693	1.771
E	6.20	6.60	2.441	2.598
e	0.65(BSC)		0.256(BSC)	
L	0.45	0.75	0.177	0.295
L1	1.00REF		0.394REF	
θ	0	8°	0	8°

附录 1，寄存器类型

缩写	描述	说明
WO	Write Only, read “0”	只写，读为 0
RO	Read Only	只读
RW	Read, Write	可读，可写
RW0	Read, Write “0” only	可读，只能写 0，写 1 无效
RW1	Read, Write “1” only	可读，只能写 1，写 0 无效
R_W1C	Read, Cleared by Writing “1”	可读，写 1 清零，写 0 无效
Res	Reserved, read “0”	保留位，只读，读为 0

附录 2，文档更改历史

日期	版本	内容
2019-7-15	1.00	初版
2019-8-21	1.01	添加注意事项 “写数据 EEPROM 存储器” 更新 ADC 的“阈值比较”章节内容 把 1T 模式去掉 更正 TIM1CCMR4.2 名字 添加 FT61F145A-RB 脚位
2019-9-5	1.02	更新电气特性 添加注意点到 AFP0 寄存器 PSRCx 寄存器描述添加 L0/L1/L2
2019-9-10	1.03	更新第 5 节系统时钟源框图
2019-10-30	1.04	更新 ADC 转换时序图 更新关于启动 ADC 的相关描述 更新电气特性 更新最小指令周期描述 更新 INTCON 寄存器的笔误 更新 PSINKx 寄存器描述 T4CKS 描述改为可读写
2019-12-2	1.05	PROM 页大小由 64 字更正为 32 字 添加指令集详细说明 FERAE 改名为 FREE
2020-2-22	1.06	更正 FT61F145A-RB/-TRB PIN11/PIN12 错误 添加 FT61F143A-RB 脚位 添加“存储器编程特性” 添加选型表 T_{AD} 改为 2μs
2020-4-26	1.07	更新电气特性 添加“关于 GIE 清 0” 添加“管脚输出映射的限制”
2020-10-30	1.08	添加 61F143B-RB 脚位 更正 61F143A-RB 脚位，pin14 与 pin15 互换位置 删掉 LVROE 更新存储器特性表格 添加“EMC 特性”表格 添加了芯片版本说明
2020-12-7	1.09	修改 TIM2 中寄存器 ICxF 的相关描述 TIM2CCMR3.T2OC2M -> TIM2CCMR3.T2OC3M
2021-10-22	2.00	全面优化寄存器表格，更新 MCU 产品订购信息

联系信息

Fremont Micro Devices (SZ) Corporation

#5−8, 10/F, Changhong Building
Ke-Ji Nan 12 Road, Nanshan District,
Shenzhen, Guangdong, PRC 518057

Tel: (+86 755) 8611 7811
Fax: (+86 755) 8611 7810

Fremont Micro Devices (HK) Corporation

#16, 16/F, Block B, Veristrong Industrial Centre,
34−36 Au Pui Wan Street, Fotan, Shatin, Hong Kong SAR

Tel: (+852) 2781 1186
Fax: (+852) 2781 1144

http://www.fremontmicro.com/

* Information furnished is believed to be accurate and reliable. However, Fremont Micro Devices (SZ) Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents of other rights of third parties, which may result from its use. No license is granted by implication or otherwise under any patent rights of Fremont Micro Devices (SZ) Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. Fremont Micro Devices (SZ) Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of Fremont Micro Devices (SZ) Corporation. The FMD logo is a registered trademark of Fremont Micro Devices (SZ) Corporation. All other names are the property of their respective owners.