



Fremont Micro Devices

FT62F28

数据手册

主要特性

8-bit 基于 EEPROM 的 RISC MCU
Program: 8k x 14; RAM: 616 x 8; Data: 128 x 8
16 / 20 / 24 / 28 / 32 引脚
12-bit 高精度 ADC ($V_{\text{ADC-REF}} = 0.5\text{V}$ 时为 10-bit 精度)
4 个定时器, 4 路独立 PWM – 1 路带死区控制
高可靠性的 28 路触摸按键
运算放大器, 2 路模拟比较器
SPI, I2C, 2x UART
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

Rev2.09

www.fremontmicro.com

8-bit CPU (EEPROM)

- 37 条 RISC 指令: 2T or 4T
- 16 MHz / 2T ($V_{DD} \geq 2.7$)
- 多达 32 个引脚

Memory

- PROGRAM: 8k x 14 bit (读/写保护)
- DATA: 128 x 8 bit (读保护)
- RAM: 616 x 8 bit
- 8 层硬件堆栈
- 扇区加密

工作条件 (5V, 25°C)

- V_{DD} ($V_{POR} \leq 1.9V$) $V_{POR} - 5.5 V$
(通过 POR 自动调整, 0°C 以上 $\leq 1.7V$)
- 工作温度等级 1 $-40 - +125^\circ C$
- 工作温度等级 2 $-40 - +105^\circ C$
- 工作温度等级 3 $-40 - +85^\circ C$
- 低 Standby 0.6 μA
- WDT 1.8 μA
- 正常模式 (16 MHz) 230 $\mu A/mips$
- 低功耗模式 (32 kHz) 11 μA

高可靠性

- 100 万次擦写次数 (typical)
- > 20 年 / 85°C 存储 (typical)
- ESD > 8 kV, EFT > 5.5 kV

ADC (12-bit)

- 12-bit 精度 (≤ 2 MHz ADC 时钟)
- 30 + 2 通道
- $V_{ADC-REF}$
 - ✓ 内部: 0.5, 2.0, 3.0, V_{DD}
 - ✓ 外部: +, - 可选
- 自动阈值比较和中断

PWM (Total 5 路)

- 支持在 SLEEP 下运行
- 4 个 PWM 通道 (同周期): 独立占空比+极性
- 1 个通道 (多达 2 个 I/O): 互补输出+死区
- 自动故障刹车 (I/O, LVD, ADC)
- 单脉冲模式; 蜂鸣器模式

Timers

- WDT (16-bit): 7-bit 后分频
- Timer0 (8-bit): 8-bit 预分频
- Timer1 (12-bit)
- Timer2 (16-bit): 4-bit 预分频和后分频
- 支持在 SLEEP 下运行

- LIRC, 1 or 2x {指令时钟, HIRC, 晶振, EC}

TOUCH

- 多达 28 个触摸按键

运算放大器

- 手动校准, 可输出到 IO / ADC

比较器 (2 路)

- 独立和窗口比较模式, 极性可选
- 128 级可编程参考电压
- 输入可选: IO, 7bit DAC, 运放输出

通信接口

- SPI, I2C, 2x UART

I/O PORTS (多达 30 个 I/O)

- 上拉/下拉电阻, 开漏
- 30 个 I/O 源电流: 3, 6 or 18mA (5V, 25°C)
- 30 个 I/O 灌电流: 53 or 70 mA (5V, 25°C)
- 8 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: 1.2, 2.0, 2.4, 2.7, 3.0, 3.3, 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 (可微调)
 - ✓ 16MHz $< \pm 1\%$ typical (2.5-5.5V, 25°C)
 - ✓ 1, 2, 4, 8, 16, 32, 64 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)

- ADC 最小分辨率为 0.12mV, 精度为 0.24mV, 适用于电流检测
- 13.56MHz 载波
- 用于锂电池的电压监控
- $\frac{1}{2} V_{DD}$ 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

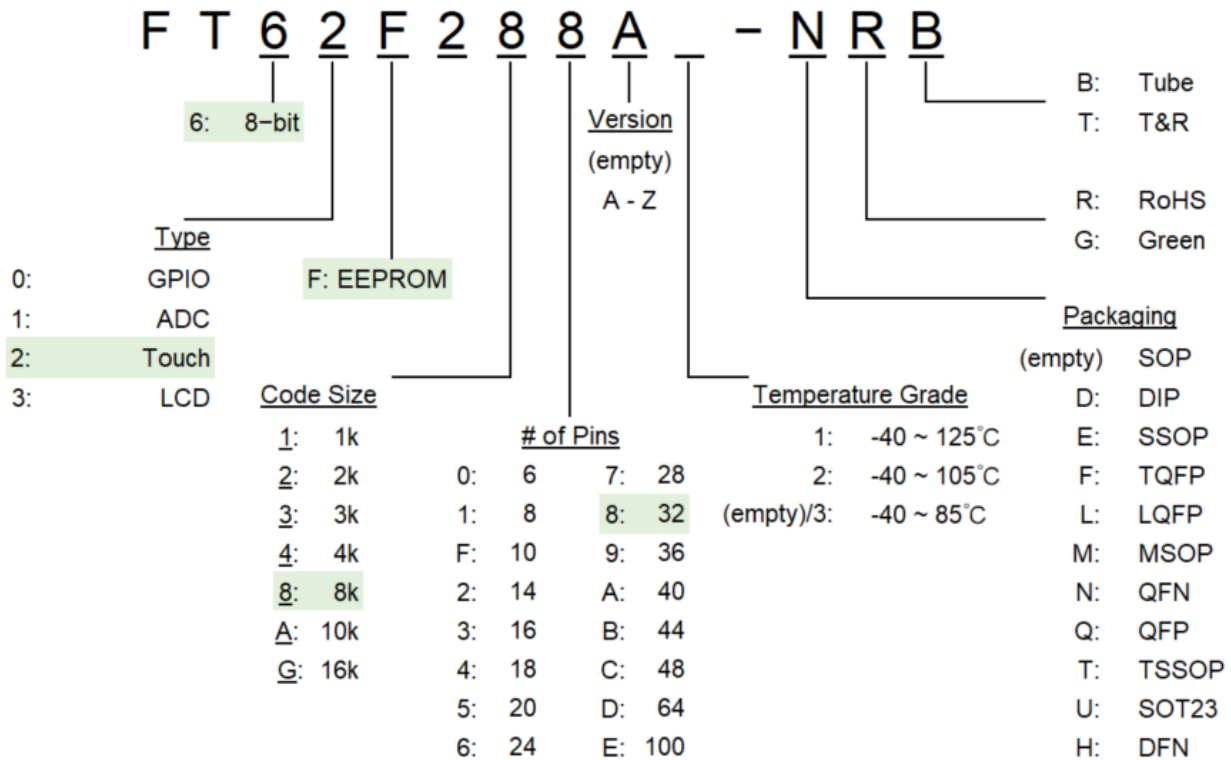
封装

- SOP16 SOP20 TSSOP20 SOP24
TSSOP24 SOP28 TSSOP28 QFN32

产品信息和选型表

型号	I/Os	封装
FT62F283- ab	14	SOP16
FT62F283B- ab		
FT62F285B- ab	18	SOP20
FT62F285B- Tab		TSSOP20
FT62F286A- ab	22	SOP24
FT62F286A- Tab		TSSOP24
FT62F287A- ab	26	SOP28
FT62F287B- ab		
FT62F287C- ab		
FT62F287C- Tab		TSSOP28
FT62F288A- Nab	30	QFN32

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



MCU 产品订购信息

文档修改历史

日期	版本	描述
2022-06-21	1.04	旧版
2021-10-22	2.00	全面优化寄存器表格，更新 MCU 产品订购信息
2021-10-29	2.01	删除型号：FT62F287-RB
2021-11-26	2.02	增加型号：FT62F286A-TRB
2022-03-18	2.03	删除型号：FT62F285A-RB 增加型号：FT62F283-RB, FT62F283A-RB, FT62F285B-RB
2022-03-31	2.04	增加型号：FT62F288-LRB
2022-05-26	2.05	增加型号：FT62F286A-RB, FT62F287C-TRB 更新 SPI、I2C、UARTx 部分寄存器复位值为 x
2022-06-21	2.06	增加型号：FT62F283B-RB
2022-08-25	2.07	全面优化版本 (请忽略旧版)
2022-10-27	2.08	修改 SPI 时序图；TODZ 配置区分版本号； 删除型号：FT62F283A-RB；
2022-11-1	2.09	删除 SPI 的 CRC 功能描述

目录

1. 结构框图和引脚.....	9
1.1. 引脚图.....	11
1.2. 引脚描述---按功能分类.....	15
2. I/O 端口.....	21
2.1. IO 端口相关寄存器汇总.....	25
2.2. I/O 配置.....	31
2.3. 引脚输出优先级.....	33
3. 上电复位 (POR).....	34
3.1. 初始化配置时序.....	34
4. 系统复位.....	36
4.1. 系统复位相关寄存器汇总.....	37
4.2. 欠压复位 (Brown-Out Reset, LVR / BOR).....	37
4.3. 非法指令复位 (Illegal Instruction Reset).....	38
4.4. 看门狗定时器 (Watch Dog Timer, WDT) 复位.....	38
4.5. 外部 I/O 系统复位 /MCLR.....	39
4.6. 检测上次复位类型.....	39
5. 低电压检测/比较器 (LVD).....	40
5.1. LVD 相关寄存器汇总.....	40
6. 振荡器和系统时钟.....	41
6.1. 振荡器模块相关寄存器汇总.....	42
6.2. 内部时钟模式 (HIRC 和 LIRC).....	44
6.3. 外部时钟模式 (EC / LP / XT).....	45
6.3.1. EC 模式.....	45
6.3.2. LP 和 XT 模式.....	46
6.4. HIRC, LIRC 和 EC 时钟的内部切换.....	47
7. 定时器 (TIMERS).....	48
7.1. 看门狗定时器 (Watch Dog Timer, WDT).....	49
7.1.1. WDT 相关寄存器汇总.....	49
7.1.2. WDT 的设置和使用.....	50

7.1.3.	在 Timer0 和 WDT 之间切换分频电路.....	51
7.2.	定时器 0 (TIMER0) 和 PWM0.....	51
7.2.1.	Timer0 相关寄存器汇总.....	52
7.2.2.	PWM0.....	54
7.3.	定时器 1 (TIMER1).....	54
7.3.1.	Timer1 相关寄存器汇总.....	55
7.3.2.	Timer1 寄存器的读/写操作.....	56
7.3.3.	Timer1 自动关闭.....	56
7.4.	定时器 2 (TIMER2).....	57
7.4.1.	Timer2 相关寄存器汇总.....	58
7.4.2.	Timer2 寄存器的读/写操作.....	59
8.	SLEEP 睡眠模式 (POWER-DOWN).....	60
8.1.	进入 SLEEP.....	60
8.2.	从 SLEEP 中唤醒.....	61
9.	中断 (INTERRUPTS).....	62
9.1.	中断相关寄存器汇总.....	64
9.2.	INT 和 PORTA 端口变化中断.....	70
10.	PWM.....	71
10.1.	PWM 相关寄存器汇总.....	72
10.2.	时钟源.....	75
10.3.	周期 (Period).....	75
10.4.	占空比 (Duty Cycle).....	75
10.5.	死区 (Deadband) 时间.....	76
10.6.	故障刹车 (Fault-Break) 功能.....	76
10.7.	周期和占空比寄存器的更新.....	78
10.8.	PWM 输出.....	78
10.9.	(P1C, P1D) 和 (P1B, P1C) 的第 2 功能输出.....	79
11.	数据 EEPROM (DATA EEPROM).....	80
11.1	DATA EEPROM 和 PROM 相关寄存器汇总.....	80
11.2	写 DATA EEPROM.....	81

11.3	读 DATA EEPROM.....	82
11.4	自动擦除功能.....	82
11.5	EEPROM 最后 16 个 byte.....	83
12.	12-bit 模/数转换器 (ANALOG TO DIGITAL CONVERTER, ADC).....	84
12.1.	ADC 相关寄存器汇总.....	85
12.2.	ADC 配置.....	89
12.2.1.	ADC 触发和延时配置.....	90
12.2.2.	ADC 中止转换.....	91
12.2.3.	中断.....	91
12.3.	ADC 采样保持时间.....	92
12.4.	ADC 最短采样时间.....	93
12.5.	ADC 转换步骤示例.....	93
13.	运算放大器 (OPAMP).....	95
13.1.	运放相关寄存器汇总.....	96
13.2.	运放的典型应用.....	97
14.	比较器.....	98
14.1.	比较器相关寄存器汇总.....	99
14.2.	比较器配置.....	101
14.3.	比较器中断.....	104
15.	SPI 接口.....	105
15.1	SPI 相关寄存器汇总.....	107
15.2	SPI 配置.....	111
15.2.1	通信时钟 SCK.....	111
15.2.2	数据处理流程.....	112
15.2.3	从机模式的睡眠唤醒.....	114
16.	I2C 接口.....	115
16.1	I2C 接口相关寄存器汇总.....	116
16.2	I2C 配置.....	119
16.2.1	主机发送模式.....	120
16.2.2	主机接收模式.....	120

16.2.3	从机发送模式.....	121
16.2.4	从机接收模式.....	122
17.	UARTx 接口	123
17.1	UARTx 接口相关寄存器汇总.....	124
17.2	UARTx 异步工作模式	127
18.	TOUCH 模块	129
19.	存储区读/写保护.....	130
20.	低功耗模式 (LOW POWER MODE).....	130
21.	指令集 (INSTRUCTION SET)	131
22.	特殊功能寄存器 (SPECIAL FUNCTION REGISTERS, SFR).....	133
22.1	初始化配置寄存器	133
22.2	用户寄存器.....	136
22.3	STATUS 寄存器.....	144
22.4	PCL 和 PCLATH	145
23.	电气特性.....	147
23.1	极限参数	147
23.2	工作特性	147
23.3	POR, LVR, LVD.....	148
23.4	I/O 端口电路.....	149
23.5	工作电流(I_{DD})	149
23.6	内部振荡器.....	150
23.7	7bit DAC 电路 (比较器参考电压)	150
23.8	ADC(12bit) 和 ADC VREF	151
23.9	运算放大器 0 特性.....	152
23.10	Comparator 比较器.....	153
23.11	Program 和 Data EEPROM	154
23.12	EMC 特性.....	154
24.	特性图	155
25.	芯片封装信息	160
26.	附录：寄存器类型.....	169

联系信息 170

1. 结构框图和引脚

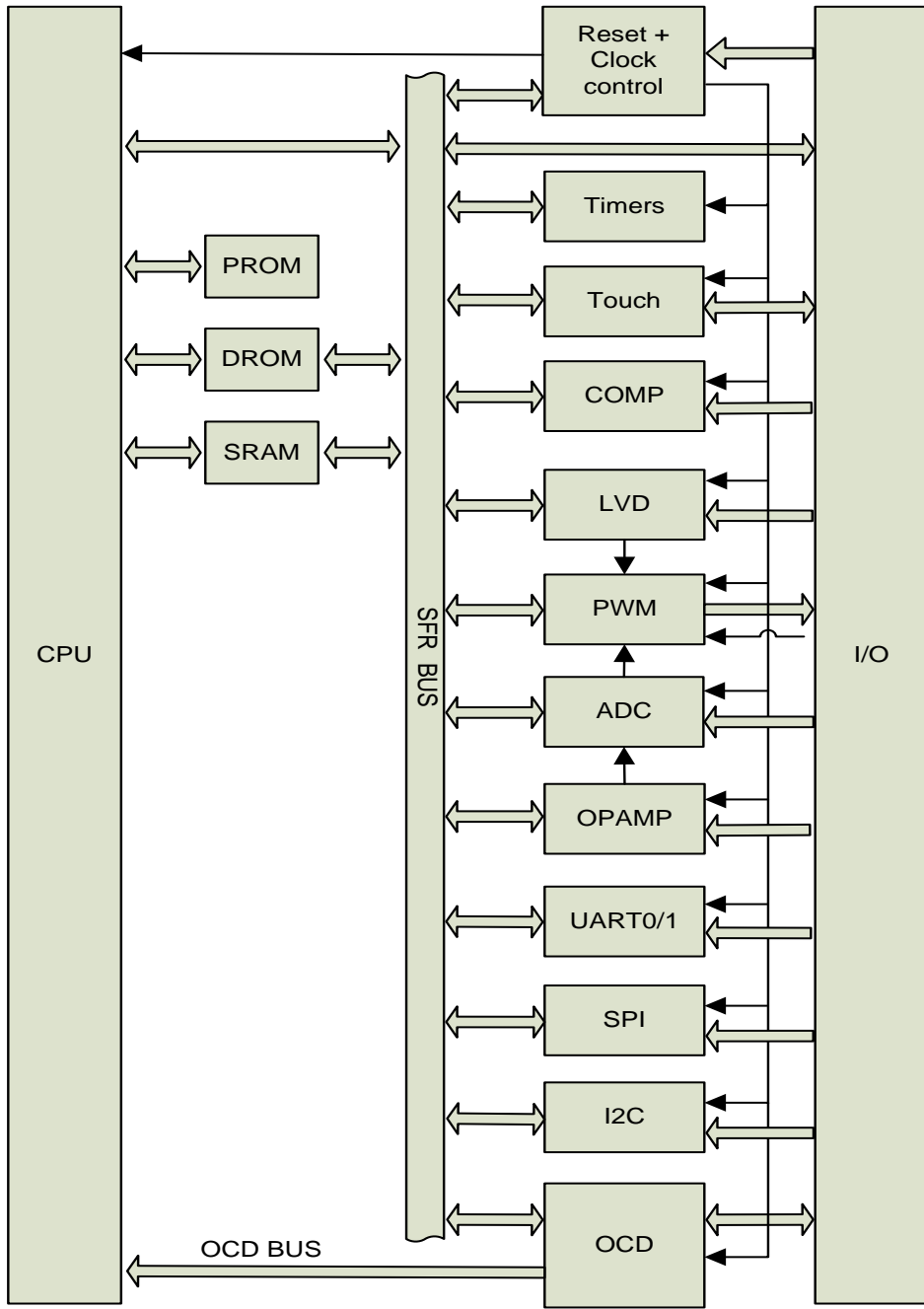


图 1-1 系统结构框图

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
Timers	WDT, Timer0, Timer1, Timer2
PWM	Pulse Width Modulator
ADC	Analog to Digital Converter
OPAMP	Operational amplifier
COMP	Comparator
LVD	Low Voltage Detect / comparator
Touch	Touch
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver Transmitter
I2C	Inter-Integrated Circuit bus (I2C-bus)
OCD	On Chip Debug
I/O	Input / Output

1.1. 引脚图

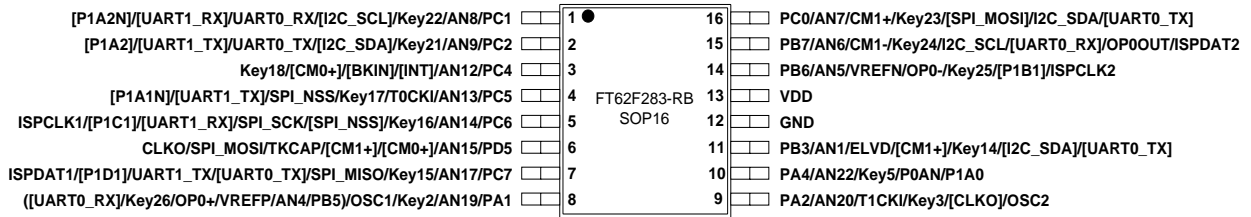


图 1-2 SOP16 ^{1 2}

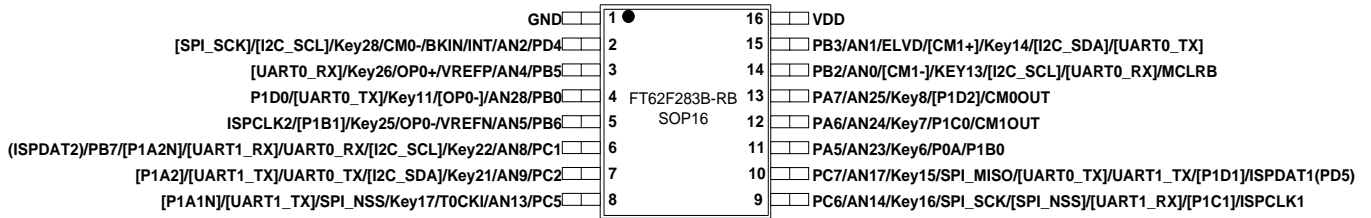


图 1-3 SOP16 (B) ^{1 2 3}

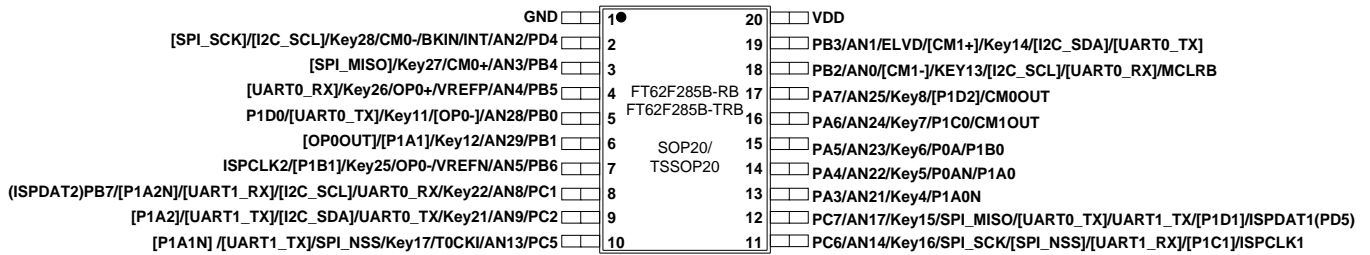


图 1-4 SOP20 / TSSOP20 ^{1 2 3}

¹ 两个 I/O 共同打线到同一个 pin 的注意事项:

封装	两个 I/O 共同打线到同一个 pin			
	Pin	I/O	同时设置为输出时, 需满足:	设置为输入或输出时:
FT62F283	pin8	PA1 PB5	PORTA1 = PORTB5	上拉/下拉的设置不能与输出值相反
FT62F283A	pin10	PC7 PD5	PORTC7 = PORTD5	
FT62F283B	pin6	PB7 PC1	PORTB7 = PORTC1	
	pin10	PC7 PD5	PORTC7 = PORTD5	
FT62F285B	pin8	PB7 PC1	PORTB7 = PORTC1	
	pin12	PC7 PD5	PORTC7 = PORTD5	

² PD3 与 GND 打线在一起, 因此禁止将其设置成输出高或使能内部上拉。

³ 因 PB7 (ISPDAT2) 为烧录调试数据脚, 而 PB7 和 PC1 打线在一起, 因此在芯片复位后的 20ms 内, 禁止将 PC1 设置成输出, 以免影响烧录调试功能。

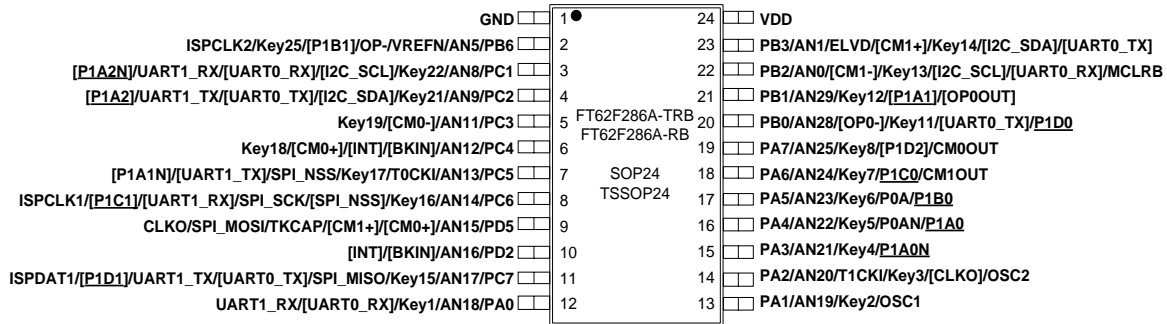


图 1-5 SOP24 / TSSOP24 ⁴

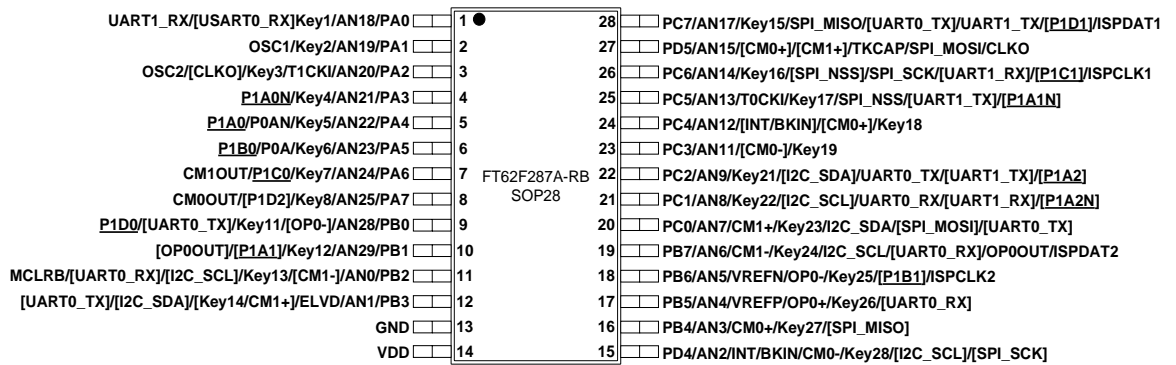


图 1-6 SOP28 (A) ⁴

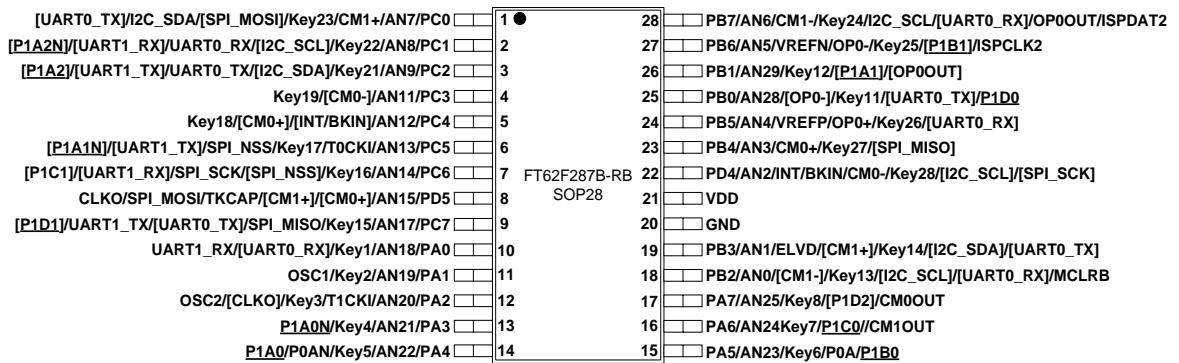


图 1-7 SOP28 (B)

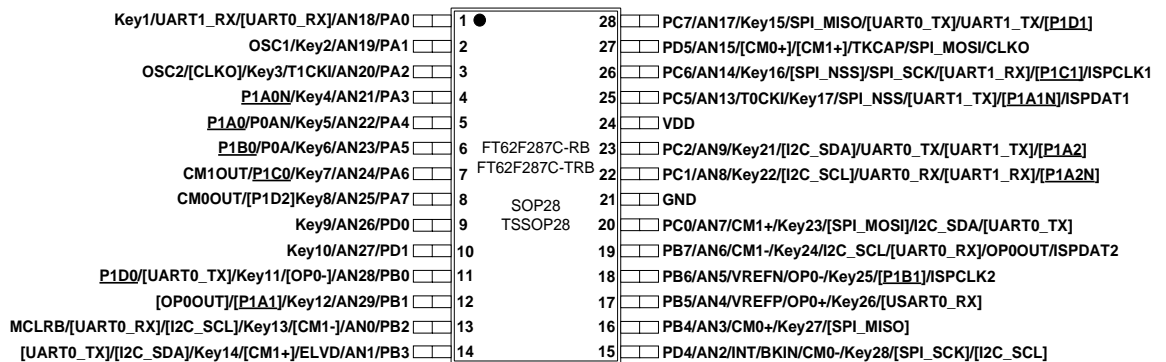


图 1-8 SOP28 (C) / TSSOP28 ⁵

⁴ 对于 FT62F286A / FT62F287A 封装，PD3 与 GND 打线在一起，因此禁止将其设置成输出高或使能内部上拉。

⁵ 对于 FT62F287C 封装，PD3 与 VDD 打线在一起，因此禁止将其设置成输出低或使能内部下拉。

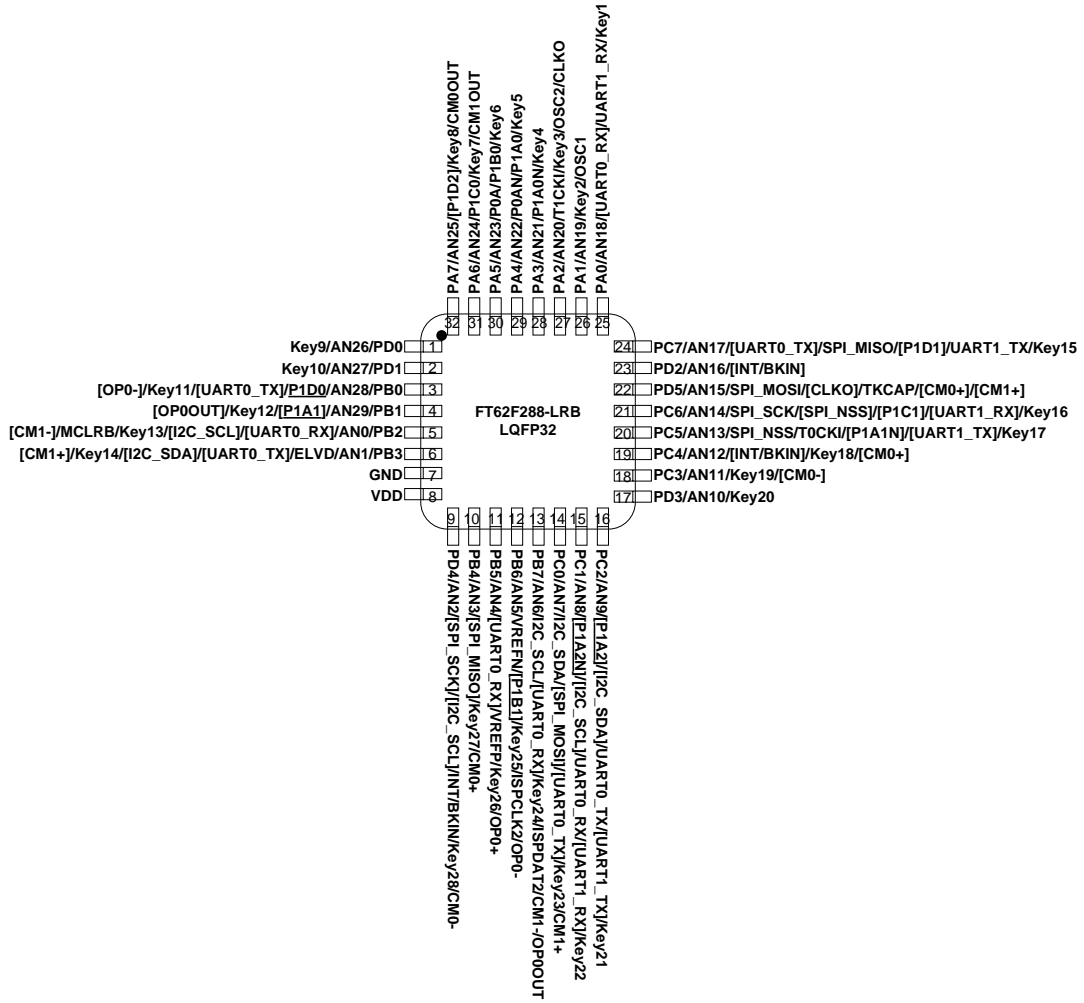
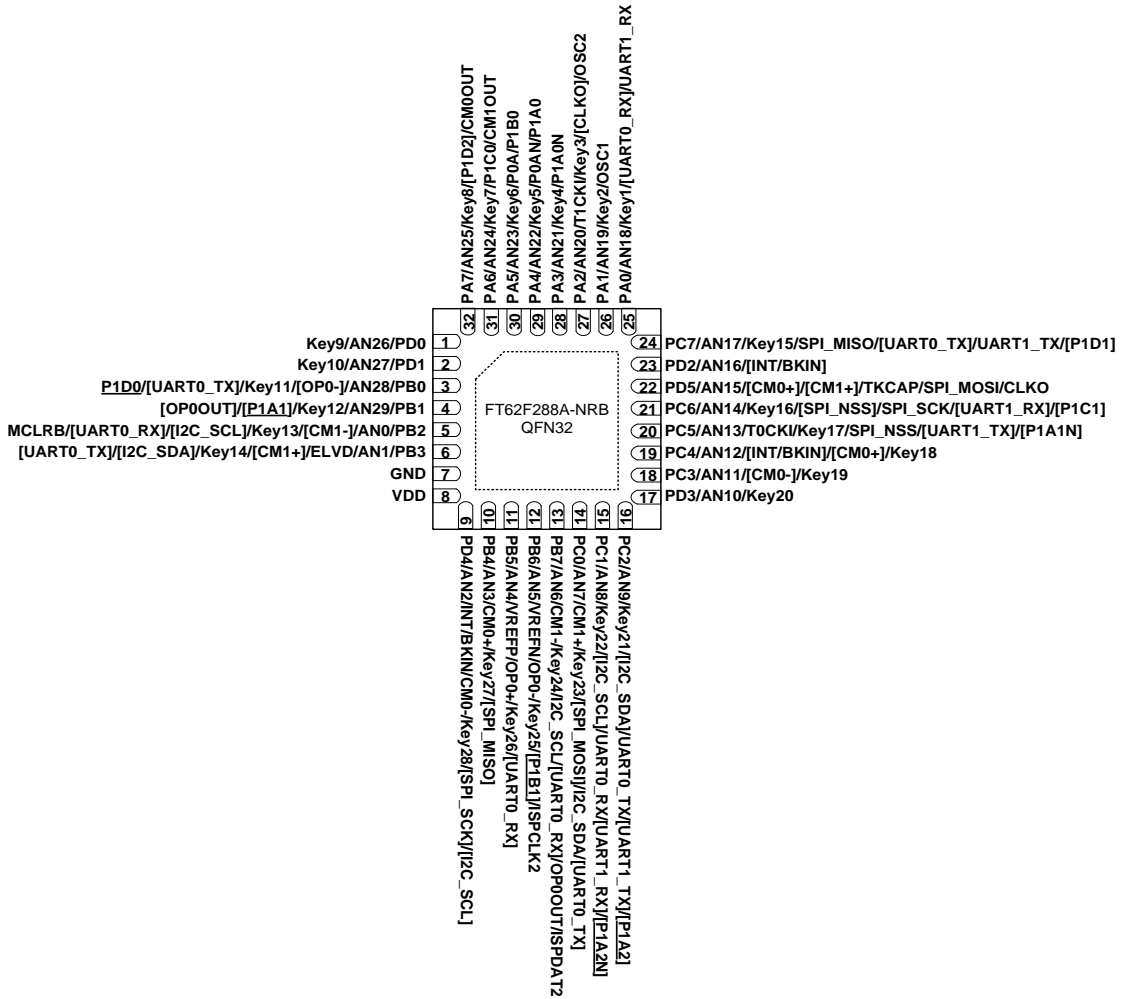


图 1-9 LQFP32



1-10 QFN32 ⁶

⁶ QFN32: 底部裸露的焊垫与 GND 连在一起。

1.2. 引脚描述--按功能分类

功能	描述	引脚名	对应 GPIO	16 pins	16(A) pins	16(B) pins	20 pins	24 pins	28(A) pins	28(B) pins	28(C) pins	32 pins	
电源		VDD		13	16	16	20	24	14	21	24	8	
		GND		12	1	1	1	1	13	20	21	7	
GPIO	上拉/下拉, 数字输入, 数字输出 开漏	PD5		6	10 ¹	10 ¹	12 ¹	9	27	8	27	22	
		PD4			2	2	2		15	22	15	9	
		PD3											17
		PD2						10					23
		PD1										10	2
		PD0										9	1
		PC7		7	10 ¹	10 ¹	12 ¹	11	28	9	28	28	24
		PC6		5	9	9	11	8	26	7	26	26	21
		PC5		4	8	8	10	7	25	6	25	25	20
		PC4		3				6	24	5			19
		PC3						5	23	4			18
		PC2		2	7	7	9	4	22	3	23	23	16
		PC1		1	6	6 ¹	8 ¹	3	21	2	22	22	15
		PC0		16						20	1	20	14
		PB7		15		6 ¹	8 ¹		19	28	19	19	13
		PB6		14	5	5	7	2	18	27	18	18	12
		PB5		8 ¹	3	3	4		17	24	17	17	11
		PB4					3		16	23	16	23	10
		PB3		11	15	15	19	23	12	19	14	19	6
		PB2			14	14	18	22	11	18	13	18	5
		PB1					6	21	10	26	12	26	4
		PB0			4	4	5	20	9	25	11	25	3
		PA7			13	13	17	19	8	17	8	17	32
		PA6			12	12	16	18	7	16	7	16	31
		PA5			11	11	15	17	6	15	6	15	30
		PA4		10			14	16	5	14	5	14	29
PA3					13	15	4	13	4	13	28		
PA2		9				14	3	12	3	12	27		
PA1		8 ¹				13	2	11	2	11	26		
PA0						12	1	10	1	10	25		
LVD	输入	ELVD	PB3	11	15	15	19	23	12	19	14	6	

功能	描述	引脚名	对应 GPIO	16 pins	16(A) pins	16(B) pins	20 pins	24 pins	28(A) pins	28(B) pins	28(C) pins	32 pins
外部复位	上拉	/MCLRB	PB2		14	14	18	22	11	18	13	5
时钟	输出	CLKO	PD5	6	10 ¹	10 ¹	12 ¹	9	27	8	27	22
		[CLKO]	PA2	9				14	3	12	3	27
	Timer0 时钟	T0CKI	PC5	4	8	8	10	7	25	6	25	20
	Timer1 时钟	T1CKI	PA2	9				14	3	12	3	27
	OSC +	OSC1	PA1	8 ¹				13	2	11	2	26
OSC -	OSC2	PA2	9				14	3	12	3	27	
ISP 调试	ISP-Data1	ISPDAT1	PC5								25	
	ISP-Data1	ISPDAT1	PC7	7	10 ¹	10 ¹	12 ¹	11	28			
	ISP-CLK1	ISPCLK1	PC6	5	9	9	11	8	26		26	
	ISP-Data2	ISPDAT2	PB7	15		6 ¹	8 ¹		19	28	19	13
	ISP-CLK2	ISPCLK2	PB6	14	5	5	7	2	18	27	18	12
外部边沿中断		INT	PD4		2		2		15	22	15	9
		[INT]	PC4	3				6	24	5		19
		[INT]	PD2					10				23
PORTA 端口变化 中断	输入	PA7			13	13	17	19	8	17	8	32
		PA6			12	12	16	18	7	16	7	31
		PA5			11	11	15	17	6	15	6	30
		PA4		10			14	16	5	14	5	29
		PA3					13	15	4	13	4	28
		PA2		9				14	3	12	3	27
		PA1		8 ¹				13	2	11	2	26
		PA0						12	1	10	1	25
PWM0		P0A	PA5		11	11	15	17	6	15	6	30
	/PWM0	P0AN	PA4	10			14	16	5	14	5	29
PWM1 (死区)		P1A0	PA4	10			14	16	5	14	5	29
		[P1A1]	PB1				6	21	10	26	12	4
		[P1A2]	PC2	2	7	7	9	4	22	3	23	16
	/PWM1	P1A0N	PA3				13	15	4	13	4	28
	/PWM1	[P1A1N]	PC5	4	8	8	10	7	25	6	25	20
	/PWM1	[P1A2N]	PC1	1	6	6 ¹	8 ¹	3	21	2	22	15
PWM2		P1B0	PA5		11	11	15	17	6	15	6	30
		[P1B1]	PB6	14	5	5	7	2	18	27	18	12
PWM3		P1C0	PA6		12	12	16	18	7	16	7	31
		[P1C1]	PC6	5	9	9	11	8	26	7	26	21

功能	描述	引脚名	对应 GPIO	16 pins	16(A) pins	16(B) pins	20 pins	24 pins	28(A) pins	28(B) pins	28(C) pins	32 pins	
PWM4		P1D0	PB0		4	4	5	20	9	25	11	3	
		[P1D1]	PC7	7	10 ¹	10 ¹	12 ¹	11	28	9	28	24	
		[P1D2]	PA7		13	13	17	19	8	17	8	32	
PWM 故障刹车输入		BKIN	PD4		2	2	2		15	22	15	9	
		[BKIN]	PC4	3				6	24	5		19	
		[BKIN]	PD2					10				23	
ADC	输入	AN29	PB1				6	21	10	26	12	4	
		AN28	PB0		4	4	5	20	9	25	11	3	
		AN27	PD1									10	2
		AN26	PD0									9	1
		AN25	PA7		13	13	17	19	8	17	8	32	
		AN24	PA6		12	12	16	18	7	16	7	31	
		AN23	PA5		11	11	15	17	6	15	6	30	
		AN22	PA4	10			14	16	5	14	5	29	
		AN21	PA3				13	15	4	13	4	28	
		AN20	PA2	9				14	3	12	3	27	
		AN19	PA1	8 ¹				13	2	11	2	26	
		AN18	PA0					12	1	10	1	25	
		AN17	PC7	7	10 ¹	10 ¹	12 ¹	11	28	9	28	24	
		AN16	PD2					10					23
		AN15	PD5	6	10 ¹	10 ¹	12 ¹	9	27	8	27	22	
		AN14	PC6	5	9	9	11	8	26	7	26	21	
		AN13	PC5	4	8	8	10	7	25	6	25	20	
		AN12	PC4	3				6	24	5			19
		AN11	PC3					5	23	4			18
		AN10	PD3										17
		AN9	PC2	2	7	7	9	4	22	3	23	16	
		AN8	PC1	1	6	6 ¹	8 ¹	3	21	2	22	15	
		AN7	PC0	16						20	1	20	14
		AN6	PB7	15		6 ¹	8 ¹			19	28	19	13
		AN5	PB6	14	5	5	7	2	18	27	18	12	
		AN4	PB5	8 ¹	3	3	4			17	24	17	11
		AN3	PB4				3			16	23	16	10
		AN2	PD4		2	2	2			15	22	15	9
AN1	PB3	11	15	15	19	23	12	19	14	6			
AN0	PB2		14	14	18	22	11	18	13	5			

功能	描述	引脚名	对应 GPIO	16 pins	16(A) pins	16(B) pins	20 pins	24 pins	28(A) pins	28(B) pins	28(C) pins	32 pins
	V _{REF-}	VREFN	PB6	14	5	5	7	2	18	27	18	12
	V _{REF+}	VREFP	PB5	8 ¹	3	3	4		17	24	17	11
	ADC_ETR	ADC_ETR	PD4		2	2	2		15	22	15	9
		[ADC_ETR]	PC4	3					6	24	5	
	[ADC_ETR]	PD2						10				23
运放	正相端输入	OP0+	PB5	8 ¹	3	3	4		17	24	17	11
	反相端输入	OP0-	PB6	14	5	5	7	2	18	27	18	12
		[OP0-]	PB0		4	4	5	20	9	25	11	3
	输出	OP0OUT	PB7	15		6 ¹	8 ¹		19	28	19	13
[OP0OUT]		PB1					6	21	10	26	12	4
比较器 0	正相端输入	CM0+	PB4				3		16	23	16	10
		[CM0+]	PC4	3				6	24	5		19
		[CM0+]	PD5	6	10 ¹	10 ¹	12 ¹	9	27	8	27	22
	反相端输入	CM0-	PD4				2		15	22	15	9
		[CM0-]	PC3					5	23	4		18
输出	CM0OUT	PA7		13	13	17	19	8	17	8	32	
比较器 1	正相端输入	CM1+	PC0	16					20	1	20	14
		[CM1+]	PB3	11	15	15	19	23	12	19	14	6
		[CM1+]	PD5	6	10 ¹	10 ¹	12 ¹	9	27	8	27	22
	反相端输入	CM1-	PB7	15		6	8 ¹		19	28	19	13
		[CM1-]	PB2		14	14	18	22	11	18	13	5
输出	CM1OUT	PA6		12	12	16	18	7	16	7	31	
SPI	SPI_MISO	SPI_MISO	PC7	7	10 ¹	10 ¹	12	11	28	9	28	24
		[SPI_MISO]	PB4				3		16	23	16	10
	SPI_MOSI	SPI_MOSI	PD5	6	10 ¹	10 ¹	12 ¹	9	27	8	27	22
		[SPI_MOSI]	PC0	16					20	1	20	14
	SPI_NSS	SPI_NSS	PC5	4	8	8	10	7	25	6	25	20
		[SPI_NSS]	PC6	5	9	9	11	8	26	7	26	21
SPI_SCK	SPI_SCK	PC6	5	9	9	11	8	26	7	26	21	
	[SPI_SCK]	PD4		2	2	2		15	22	15	9	
I2C	I2C_SDA	I2C_SDA	PC0	16					20	1	20	14
		[I2C_SDA]	PC2	2	7	7	9	4	22	3	23	16
		[I2C_SDA]	PB3	11	15	15	19	23	12	19	14	6
	I2C_SCL	I2C_SCL	PB7	15		6 ¹	8 ¹		19	28	19	13
		[I2C_SCL]	PB2		14	14	18	22	11	18	13	5
		[I2C_SCL]	PD4		2	2	2		15	22	15	9

功能	描述	引脚名	对应 GPIO	16 pins	16(A) pins	16(B) pins	20 pins	24 pins	28(A) pins	28(B) pins	28(C) pins	32 pins
		[I2C_SCL]	PC1	1	6	6 ¹	8 ¹	3	21	2	22	15
UART0	UART0_TX	UART0_TX	PC2	2	7	7	9	4	22	3	23	16
		[UART0_TX]	PB0		4	4	5	20	9	25	11	3
		[UART0_TX]	PB3	11	15	15	19	23	12	19	14	6
		[UART0_TX]	PC0	16					20	1	20	14
	[UART0_TX]	PC7	7	10 ¹	10 ¹	12 ¹	11	28	9	28	24	
	UART0_RX	UART0_RX	PC1	1	6	6 ¹	8 ¹	3	21	2	22	15
		[UART0_RX]	PB2		14	14	18	22	11	18	13	5
		[UART0_RX]	PB5	8 ¹	3	3	4		17	24	17	11
		[UART0_RX]	PB7	15		6 ¹	8 ¹		19	28	19	13
[UART0_RX]		PA0					12	1	10	1	25	
UART1	UART1_TX	UART1_TX	PC7	7	10 ¹	10 ¹	12 ¹	11	28	9	28	24
		[UART1_TX]	PC2	2	7	7	9	4	22	3	23	16
		[UART1_TX]	PC5	4	8	8	10	7	25	6	25	20
	UART1_RX	UART1_RX	PA0					12	1	10	1	25
		[UART1_RX]	PC1	1	6	6 ¹	8 ¹	3	21	2	22	15
		[UART1_RX]	PC6	5	9	9	11	8	26	7	26	21
TOUCH	输入	KEY1	PA0					12	1	10	1	25
		KEY2	PA1	8 ¹				13	2	11	2	26
		KEY3	PA2	9				14	3	12	3	27
		KEY4	PA3				13	15	4	13	4	28
		KEY5	PA4				14	16	5	14	5	29
		KEY6	PA5		11	11	15	17	6	15	6	30
		KEY7	PA6		12	12	16	18	7	16	7	31
		KEY8	PA7		13	13	17	19	8	17	8	32
		KEY9	PD0								9	1
		KEY10	PD1								10	2
		KEY11	PB0		4	4	5	20	9	25	11	3
		KEY12	PB1				6	21	10	26	12	4
		KEY13	PB2		14	14	18	22	11	18	13	5
		KEY14	PB3	11	15	15	19	23	12	19	14	6
		KEY15	PC7	7	10 ¹	10 ¹	12 ¹	11	28	9	28	24
		KEY16	PC6	5	9	9	11	8	26	7	26	21
		KEY17	PC5	4	8	8	10	7	25	6	25	20
		KEY18	PC4	3				6	24	5		19
		KEY19	PC3					5	23	4		18

功能	描述	引脚名	对应 GPIO	16 pins	16(A) pins	16(B) pins	20 pins	24 pins	28(A) pins	28(B) pins	28(C) pins	32 pins
		KEY20	PD3									17
		KEY21	PC2	2	7	7	9	4	22	3	23	16
		KEY22	PC1	1	6	6 ¹	8 ¹	3	21	2	22	15
		KEY23	PC0	16					20	1	20	14
		KEY24	PB7	15		6 ¹	8 ¹		19	28	19	13
		KEY25	PB6	14	5	5	7	2	18	27	18	12
		KEY26	PB5	8 ¹	3	3	4		17	24	17	11
		KEY27	PB4				3		16	23	16	10
		KEY28	PD4		2	2	2		15	22	15	9
	C _{REF}	TKCAP	PD5	6	10 ¹	10 ¹	12 ¹	9	27	8	27	22

表 1-1 按功能分类的引脚描述

2. I/O 端口

根据不同的封装类型, FT62F28x 系列芯片最多有 30 个 I/O 引脚。共分为 4 组: PORTA(8)、PORTB(8)、PORTC(8) 和 PORTD(6)。表 2-1 和 表 2-2 列出了所有 I/O 引脚的功能。

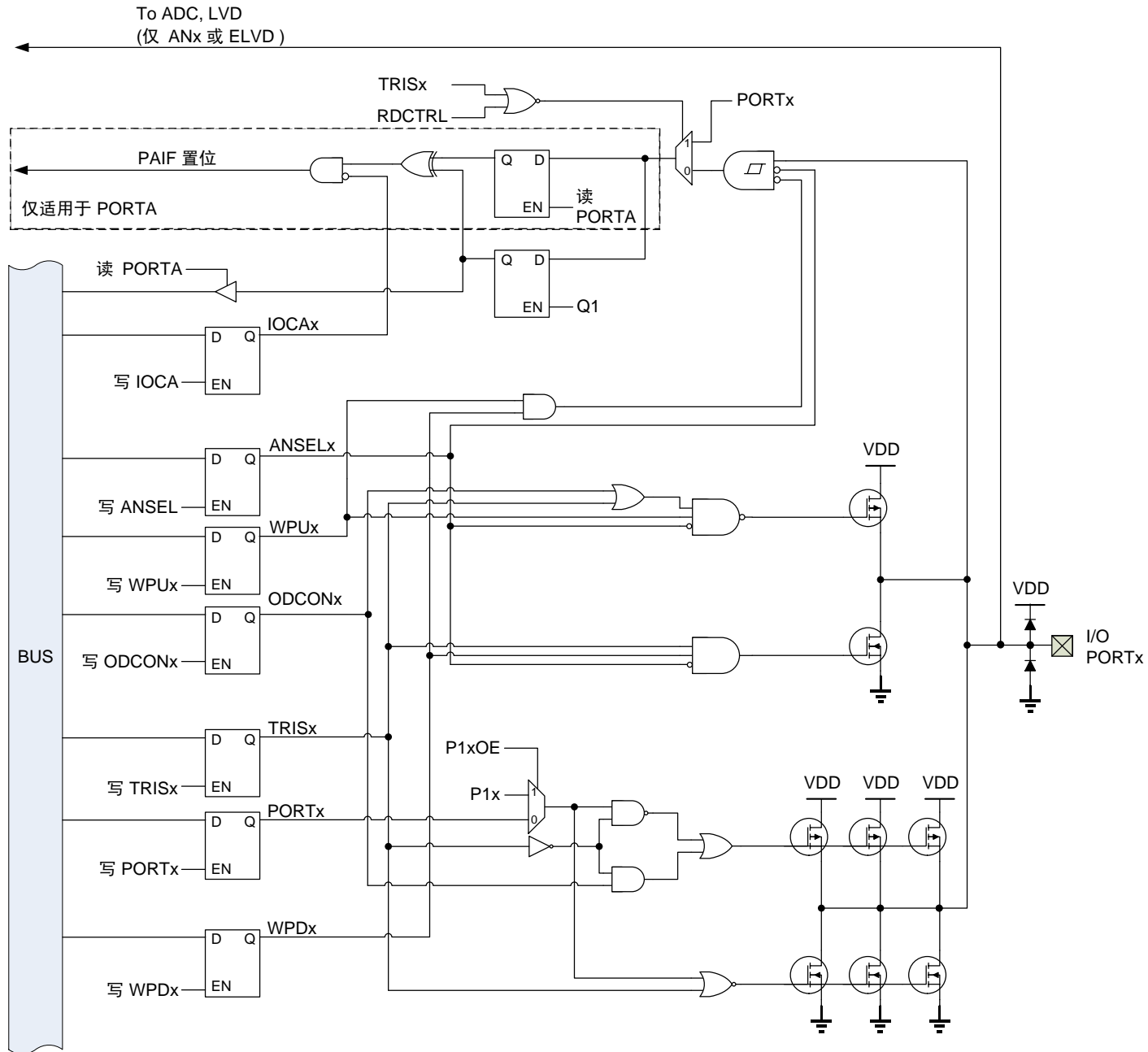


图 2-1 PORT 端口结构框图

所有 I/O 引脚均具有以下功能 (表 2-4):

- 数字输出
- 数字输入
- 开漏
- 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, 不需要设置。
2. 通过 IDE 界面选择, 且在芯片初始化配置时加载的功能 (表 2-6):
 - 外部时钟/晶振输入 (OSC1, OSC2)
 - 系统外部复位 (/MCLR/B)
 - 内部时钟输出
3. 通过指令对相应 I/O 引脚进行配置的其他功能, 可分为 5 类:
 - a) 数字输出
 - PWM
 - 内部时钟输出
 - 比较器输出 CMxOUT
 - b) 数字输入
 - PWM 故障刹车
 - Timer0 时钟输入
 - Timer1 时钟输入
 - 外部边沿中断 (INT)
 - GPIO 端口变化中断
 - ADC 触发 (ADC_ETR)
 - c) 模拟输入
 - LVD / BOR
 - ADC
 - V_{REF+}
 - V_{REF-}
 - TOUCH
 - 运放输入 OP0+ / OP0-
 - 比较器输入 CMx+ / CMx-
 - d) 模拟输出
 - 运放输出 OP0OUT
 - e) 通信接口
 - SPI
 - I2C
 - UART0 / UART1

引脚名	ISP 调试	时钟	中断	LVD	PWM	数字 I/O 上拉/下拉	开漏	源电流 (mA)	灌电流 (mA)
PA0			√			√	√	3, 6, 24	53, 70
PA1		OSC+	√			√	√	3, 6, 24	53, 70
PA2		OSC-/ [CLKO]	√			√	√	3, 6, 24	53, 70
PA3			√		P1A0N	√	√	3, 6, 24	53, 70
PA4			√		P0AN / P1AN	√	√	3, 6, 24	53, 70
PA5			√		P0A / P1B0	√	√	3, 6, 24	53, 70
PA6			√		P1C0	√	√	3, 6, 24	53, 70
PA7			√		[P1D2]	√	√	3, 6, 24	53, 70
PB0					P1D0	√	√	3, 6, 24	53, 70
PB1					[P1A1]	√	√	3, 6, 24	53, 70
PB2			/MCLR			√	√	3, 6, 24	53, 70
PB3				ELVD		√	√	3, 6, 24	53, 70
PB4						√	√	3, 6, 24	53, 70
PB5						√	√	3, 6, 24	53, 70
PB6	CLK2				[P1B1]	√	√	3, 6, 24	53, 70
PB7	DAT2					√	√	3, 6, 24	53, 70
PC0						√	√	3, 6, 24	53, 70
PC1					[P1A2N]	√	√	3, 6, 24	53, 70
PC2					[P1A2]	√	√	3, 6, 24	53, 70
PC3						√	√	3, 6, 24	53, 70
PC4			[INT]		[BKIN]	√	√	3, 6, 24	53, 70
PC5	DAT1	T0CKI			[P1A1N]	√	√	3, 6, 24	53, 70
PC6	CLK1				[P1C1]	√	√	3, 6, 24	53, 70
PC7	DAT1				[P1D1]	√	√	3, 6, 24	53, 70
PD0						√	√	3, 6, 24	53, 70
PD1						√	√	3, 6, 24	53, 70
PD2			[INT]		[BKIN]	√	√	3, 6, 24	53, 70
PD3						√	√	3, 6, 24	53, 70
PD4			INT		BKIN	√	√	3, 6, 24	53, 70
PD5		CLKO				√	√	3, 6, 24	53, 70
注:		T1CKI = PA2						V _{DD} =5, V _{DS} =0.5	

表 2-1 I/O 端口功能

注： 所有 IO 支持 3 档可配置源电流驱动能力(参阅“PSRCx”，表 2-4)，和 2 档可配置灌电流驱动能力 (参阅“PSINKx”，表 2-4)。

引脚名	ADC	TOUCH	比较器	运算放大器	SPI	I2C	UART0	UART1
PA0	AN18	KEY1					[RX]	RX
PA1	AN19	KEY2						
PA2	AN20	KEY3						
PA3	AN21	KEY4						
PA4	AN22	KEY5						
PA5	AN23	KEY6						
PA6	AN24	KEY7	CM1OUT					
PA7	AN25	KEY8	CM0OUT					
PB0	AN28	KEY11		[OP0-]			[TX]	
PB1	AN29	KEY12		[OP0OUT]				
PB2	AN0	KEY13	[CM1-]			[SCL]	[RX]	
PB3	AN1	KEY14	[CM1+]			[SDA]	[TX]	
PB4	AN3	KEY27	CM0+		[MISO]			
PB5	AN4 / (V_{REF+})	KEY26		OP0+			[RX]	
PB6	AN5 / (V_{REF-})	KEY25		OP0-				
PB7	AN6	KEY24	CM1-	OP0OUT		SCL	[RX]	
PC0	AN7	KEY23	CM1+		[MOSI]	SDA	[TX]	
PC1	AN8	KEY22				[SCL]	RX	[RX]
PC2	AN9	KEY21				[SDA]	TX	[TX]
PC3	AN11	KEY19	[CM0-]					
PC4	AN12	KEY18	[CM0+]					
PC5	AN13	KEY17			NSS [NSS]			[TX]
PC6	AN14	KEY16			SCK			[RX]
PC7	AN17	KEY15			MISO		[TX]	TX
PD0	AN26	KEY9						
PD1	AN27	KEY10						
PD2	AN16							
PD3	AN10	KEY20						
PD4	AN2	KEY28	CM0-		[SCK]	[SCL]		
PD5	AN15	TKCAP	[CM0+]/[CM1+]		MOSI			

表 2-2 I/O 端口功能 (续)

2.1. IO 端口相关寄存器汇总

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
PORTA	0x05	PORTA 输出寄存器								xxxx xxxx
PORTB	0x06	PORTB 输出寄存器								xxxx xxxx
PORTC	0x07	PORTC 输出寄存器								xxxx xxxx
PORTD	0x08	—	—	PORTD 输出寄存器						--xx xxxx
TRISA	0x85	PORTA 方向控制								1111 1111
TRISB	0x86	PORTB 方向控制								1111 1111
TRISC	0x87	PORTC 方向控制								1111 1111
TRISD	0x88	—	—	PORTD 方向控制						--11 1111
WPUA	0x294	PORTA 弱上拉								1111 1111
WPUB	0x295	PORTB 弱上拉								0000 0000
WPUC	0x296	PORTC 弱上拉								0000 0000
WPUD	0x297	—	—	PORTD 弱上拉						--00 0000
WPDA	0x290	PORTA 弱下拉								0000 0000
WPDB	0x291	PORTB 弱下拉								0000 0000
WPDC	0x292	PORTC 弱下拉								0000 0000
WPDD	0x293	—	—	PORTD 弱下拉						--00 0000
ODCONA	0x205	PORTA 开漏								0000 0000
ODCONB	0x206	PORTB 开漏								0000 0000
ODCONC	0x207	PORTC 开漏								0000 0000
ODCOND	0x208	—	—	PORTD 开漏						--00 0000
PSINKA	0x214	PORTA 灌电流设置								0000 0000
PSINKB	0x215	PORTB 灌电流设置								0000 0000
PSINKC	0x216	PORTC 灌电流设置								0000 0000
PSINKD	0x217	—	—	PORTD 灌电流设置						--00 0000
PSRCAL	0x20C	PORTA 源电流设置 (低 8 位)								1111 1111
PSRCAH	0x20D	PORTA 源电流设置 (高 8 位)								1111 1111
PSRCBL	0x20E	PORTB 源电流设置 (低 8 位)								1111 1111
PSRCBH	0x20F	PORTB 源电流设置 (高 8 位)								1111 1111
PSRCCL	0x210	PORTC 源电流设置 (低 8 位)								1111 1111
PSRCCH	0x211	PORTC 源电流设置 (高 8 位)								1111 1111
PSRCDL	0x212	PORTD 源电流设置 (低 8 位)								1111 1111
PSRCDH	0x213	—	—	—	—	PORTD 源电流设置 (高 4 位)				---- 1111
ANSEL0	0x11E	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	0000 0000
ANSEL1	0x11F	ANS15	ANS14	ANS13	ANS12	ANS11	ANS10	ANS9	ANS8	0000 0000
ANSEL2	0x28C	ANS23	ANS22	ANS21	ANS20	ANS19	ANS18	ANS17	ANS16	0000 0000
ANSEL3	0x28D	—	—	ANS29	ANS28	ANS27	ANS26	ANS25	ANS24	--00 0000
COMAF0	0x285	SCKPO	NSSPO	MOSIPO	MISOPO	SDAPO[1:0]		SCLPO[1:0]		0000 0000
COMAF1	0x286	INTPO[1:0]		TX0PO[2:0]			RX0PO[2:0]			0000 0000
COMAF2	0x287	UR1SW	UR0SW	—	TX1PO[1:0]		INTFIXB	RX1PO[1:0]		00-0 0100
IOCA	0x98	IOCA[7:0]: PORTA 端口变化中断设置								0000 0000
OPTION	0x81	/PAPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111

表 2-3 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]	0x85	RW-1111 1111
TRISB	PORTB		TRISB[7:0]	0x86	RW-1111 1111
TRISC	PORTC		TRISC[7:0]	0x87	RW-1111 1111
TRISD	PORTD		TRISD[5:0]	0x88	RW-11 1111
ANSEL0	AN[7:0]	1 = 关闭上拉/下拉, 及数字输入 (适用于 30 个 ADC 通道) 0 = (无动作)	ANSEL0[7:0]	0x11E	RW-0000 0000
ANSEL1	AN[15:8]		ANSEL1[7:0]	0x11F	RW-0000 0000
ANSEL2	AN[23:16]		ANSEL2[7:0]	0x28C	RW-0000 0000
ANSEL3	AN[29:24]		ANSEL3[5:0]	0x28D	RW-00 0000
/PAPU	1 = 关闭所有 PORTA 上拉功能 0 = 上拉由 WPUA 控制		OPTION[7]	0x81	RW-1
WPUA	PORTA	<u>弱上拉</u> 1 = 使能 (PORTA 默认值) 0 = 关闭 (PORTB, C, D 默认值)	WPUA[7:0]	0x294	RW-1111 1111
WPUB	PORTB		WPUB[7:0]	0x295	RW-0000 0000
WPUC	PORTC		WPUC[7:0]	0x296	RW-0000 0000
WPUD	PORTD		WPUD[5:0]	0x297	RW-00 0000
WPDA	PORTA	<u>弱下拉</u> 1 = 使能 0 = 关闭	WPDA[7:0]	0x290	RW-0000 0000
WPDB	PORTB		WPDB[7:0]	0x291	RW-0000 0000
WPDC	PORTC		WPDC[7:0]	0x292	RW-0000 0000
WPDD	PORTD		WPDD[5:0]	0x293	RW-00 0000
ODCONA	PORTA	<u>开漏</u> 1 = 使能 0 = 关闭	ODCONA[7:0]	0x205	RW-0000 0000
ODCONB	PORTB		ODCONB[7:0]	0x206	RW-0000 0000
ODCONC	PORTC		ODCONC[7:0]	0x207	RW-0000 0000
ODCOND	PORTD		ODCOND[5:0]	0x208	RW-00 0000
PORTA	PORTA	<u>数据输出寄存器</u>	PORTA[7:0]	0x05	RW-xxxx xxxx
PORTB	PORTB		PORTB[7:0]	0x06	RW-xxxx xxxx
PORTC	PORTC		PORTC[7:0]	0x07	RW-xxxx xxxx
PORTD	PORTD		PORTD[5:0]	0x08	RW--xx xxxx
PSRCA ¹	PORTA	<u>源电流 (mA)</u> (00) = 3 (01) = 6 / (10) = 6 (11) = 24	PSRCAL[7:0]	0x20C	RW-1111 1111
			PSRCAH[7:0]	0x20D	RW-1111 1111
PSRCB ¹	PORTB		PSRCBL[7:0]	0x20E	RW-1111 1111
			PSRCBH[7:0]	0x20F	RW-1111 1111
PSRCC ¹	PORTC		PSRCCL[7:0]	0x210	RW-1111 1111
			PSRCCH[7:0]	0x211	RW-1111 1111
PSRCD ¹	PORTD		PSRCDL[7:0]	0x212	RW-1111 1111
			PSRCDH[3:0]	0x213	RW-1111

¹ PSRCx [(2y+1) : 2y] 控制管脚 PORTx.y (其中 x = A/B/C/D, y = 0-7)。比如 PSRCA[1:0] 控制 PA0 的源电流档位。

名称	状态		寄存器	地址	复位值
PSINKA	PORTA	灌电流 (mA) 1 = 70 0 = <u>53</u>	PSINKA[7:0]	0x214	RW-0000 0000
PSINKB	PORTB		PSINKB[7:0]	0x215	RW-0000 0000
PSINKC	PORTC		PSINKC[7:0]	0x216	RW-0000 0000
PSINKD	PORTD		PSINKD[5:0]	0x217	RW-00 0000
SCKPO ²	<u>SPI_SCK</u> 1 = PD4 0 = <u>PC6</u>		COMAF0[7]	0x285	RW-0
NSSPO ²	<u>SPI_NSS</u> 1 = PC6 0 = <u>PC5</u>		COMAF0[6]		RW-0
MOSIPO ²	<u>SPI_MOSI</u> 1 = PC0 0 = <u>PD5</u>		COMAF0[5]		RW-0
MISOPO ²	<u>SPI_MISO</u> 1 = PB4 0 = <u>PC7</u>		COMAF0[4]		RW-0
SDAPO ²	<u>I2C_SDA</u> 00 = <u>PC0</u> 10 = PC2 01 = PB3 11 = PB4		COMAF0[3:2]		RW-00
SCLPO ²	<u>I2C_SCL</u> 00 = <u>PB7</u> 10 = PC1 01 = PB2 11 = PD4		COMAF0[1:0]		RW-00
INTPO ²	<u>INT/BKIN/ADC_ETR</u> 0x = <u>PD4</u> 10 = PD2 11 = PC4		COMAF1[7:6]		RW-00
TX0PO ²	<u>UART0_TX</u> 当 UR0SW = 0 时: 当 UR0SW = 1 时: 0xx = <u>PC2</u> 0xx = <u>PC1</u> 100 = PC0 100 = PB7 101 = PB3 101 = PB2 110 = PB0 110 = PB5 111 = PC7 111 = PA0		COMAF1[5:3]	0x286	RW-000
RX0PO ²	<u>UART0_RX</u> 当 UR0SW = 0 时: 当 UR0SW = 1 时: 0xx = <u>PC1</u> 0xx = <u>PC2</u> 100 = PB7 100 = PC0 101 = PB2 101 = PB3 110 = PB5 110 = PB0 111 = PA0 111 = PC7		COMAF1[2:0]		RW-000

² 管脚重映射选择。

名称	状态		寄存器	地址	复位值
UR1SW	<u>UART1 TX 和 RX 管脚交换</u> 1 = 交换 (详见 TX1PO 和 RX1PO) 0 = 正常		COMAF2[7]	0x287	RW-0
UR0SW	<u>UART0 TX 和 RX 管脚交换</u> 1 = 交换 (详见 TX0PO 和 RX0PO) 0 = 正常		COMAF2[6]		RW-0
TX1PO ²	<u>UART1 TX</u> 当 UR1SW = 0 时: 当 UR1SW = 1 时: 0x = <u>PC7</u> 0x = <u>PA0</u> 10 = PC5 10 = PC6 11 = PC2 11 = PC1		COMAF2[4:3]		RW-00
INTFIXB	<u>INT 中断管脚选择 (≥ VerG 芯片)</u> 1 = 由 INTPO 决定 0 = PD4 (BKIN 和 ADC_ETR 映射仍由 INTPO 决定) 注: < VerG 芯片, 此位为保留位, 读为 0		COMAF2[2]		RW-1
RX1PO ²	<u>UART1 RX</u> 当 UR1SW=0 时: 当 UR1SW=1 时: 0x = <u>PA0</u> 0x = <u>PC7</u> 10 = PC6 10 = PC5 11 = PC1 11 = PC2		COMAF2[1:0]		RW-00
CLKOS ²	<u>CLKO</u> 1 = <u>PD5</u> 0 = PA2		MSCON0[4]	0x10C	RW-1
OP0PSEL	<u>运放 0 正相端输入</u>	1 = <u>GND</u> 0 = PB5	OP0CR0[6]	0x96	RW-1
OP0NSEL	<u>运放 0 反相端输入</u> 00 = GND 01 = PB6 或 PB0 (由 IPINSW 决定) 10 = <u>串联 4k 电阻连接到 PB6 或 PB0 (由 IPINSW 决定)</u> 11 = 串联 4k 电阻连接到 GND		OP0CR0[5:4]		RW-10
OPINSW	运放 0 输出管脚	1 = PB1 0 = <u>PB7</u>	OP0CR1[5]		0x97
IPINSW	运放 0 反相端外部输入管脚	1 = PB0 0 = <u>PB6</u>	OP0CR1[4]	RW-0	

名称	状态		寄存器	地址	复位值
OPTOIO	运放 0 输出	1 = 使能 0 = 禁止	OP0CR1[0]		RW-0
CM0OE	比较器 0 输出到 PA7	1 = 使能 0 = 禁止	CM0CON0[5]	0x29D	RW-0
CM0PSEL	比较器 0 正相端输入	(00) = PB4 (01) = PC4 (10) = PD5 (11) = FLOAT	CM0CON0[3:2]		RW-00
CM0NSEL	比较器 0 反相端输入	00 = PD4 01 = PC3 10 = 运放 0 输出 11 = DAC1OUT	CM0CON0[1:0]		RW-00
CM1OE	比较器 1 输出到 PA6	1 = 使能 0 = 禁止	CM1CON0[5]		0x29E
CM1PSEL	比较器 1 正相端输入	(00) = PB3 (01) = PC0 (10) = PD5 (11) = FLOAT	CM1CON0[3:2]	RW-00	
CM1NSEL	比较器 1 反相端输入	00 = PB2 01 = PB7 10 = 运放 0 输出 11 = DAC2OUT	CM1CON0[1:0]	RW-00	
P1C0OE	P1C0 输出到 PA6	1 = 使能 0 = 禁止	P1OE0[7]	0x9E	RW-0
P1B0OE	P1B0 输出到 PA5		P1OE0[6]		RW-0
P1A2NOE	P1A2N 输出到 PC1		P1OE0[5]		RW-0
P1A2OE	P1A2 输出到 PC2		P1OE0[4]		RW-0
P1A1NOE	P1A1N 输出到 PC5		P1OE0[3]		RW-0
P1A1E	P1A1 输出到 PB1		P1OE0[2]		RW-0
P1A0NOE	P1A0N 输出到 PA3		P1OE0[1]		RW-0
P1A0OE	P1A0 输出到 PA4		P1OE0[0]		RW-0
P1D2OE	P1D2 输出到 PA7		P1OE1[7]	0x9F	RW-0
P1D1OE	P1D1 输出到 PC7		P1OE1[6]		RW-0
P1D0OE	P1D0 输出到 PB0		P1OE1[5]		RW-0
P1C1OE	P1C1 输出到 PC6		P1OE1[2]		RW-0
P1B1OE	P1B1 输出到 PB6		P1OE1[1]		RW-0

表 2-4 I/O 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, INTE, PAIE 适用) 0 = <u>全局关闭</u> (唤醒不受影响)	INTCON[7]	Bank 首地址 +0x0B	RW-0
PEIE	外设总中断	1 = 使能 (PAIE 适用) 0 = <u>关闭</u> (无唤醒)	INTCON[6]		RW-0
INTE	INT 外部中断	1 = 使能 0 = <u>关闭</u> (无唤醒)	INTCON[4]		RW-0
PAIE	PORTA 端口变化总中断	1 = 使能 0 = <u>关闭</u>	INTCON[3]		RW-0
INTF	INT 外部中断标志位	1 = Yes (锁存) 0 = <u>No</u>	INTCON[1]		RW0-0
PAIF	PORTA 端口变化总中断标志位	1 = Yes (锁存) 0 = <u>No</u>	INTCON[0]		RW0-0
IOCA	PORTA 各端口变化中断	1 = 使能 0 = <u>禁止</u>	IOCA[7:0]	0x98	RW-0000 0000
INTEDG	INT 外部中断有效沿选择	1 = <u>上升沿</u> 0 = <u>下降沿</u>	OPTION[6]	0x14	RW-1

表 2-5 PORTA 端口和外部中断使能和标志寄存器

名称	功能	默认
RDCTRL	当 TRISx = 0 时, 读 PORTx 寄存器的返回值 • 输入锁存器 • 输出锁存器	输出锁存器
MCLRE	外部 I/O 复位	关闭
FOSC	• LP: PA1 (+) 和 PA2 (-) 接外部低速晶振 • XT: PA1 (+) 和 PA2 (-) 接外部高速晶振 • EC: PA1 (+) 接外部时钟输入, PA2 为 I/O • INTOSC: PA2 或 PD5 输出"指令时钟", PA1 为 I/O • <u>INTOSCIO</u> : PA1 和 PA2 为 I/O	INTOSCIO

表 2-6 I/O 相关初始化配置寄存器

2.2. I/O 配置

每个 PORT 端口，均需根据其相应功能配置以下 5 个模块 (表 2-4):

- 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)
LVD	Off ⁽⁵⁾	Off	Off	TRISx = 1; ANSELx = 1
ADC	Off	Off	Off	TRISx = 1; ANSELx = 1
V _{REF+} / V _{REF-}	Off	Off	Off	TRISx = 1
TOUCH	Off	Off	Off	TRISx = 1
OP0+ / OP0- 输入	Off	Off	Off	TRISx = 1
CM0+ / CM0- 输入	Off	Off	Off	TRISx = 1; ANSELx = 1
CM1+ / CM1- 输入	Off	Off	Off	TRISx = 1; ANSELx = 1
SPI / I2C / UARTx 输入	On	(可选)	Off	TRISx = 1
外部边沿中断 (INT)	On	(可选)	Off	TRISx = 1
GPIO 端口变化中断	On	(可选)	Off	TRISx = 1
BKIN	On	(可选)	Off	TRISx = 1
ADC 触发	On	(可选)	Off	TRISx = 1
Timer0 时钟	On	(可选)	Off	TRISx = 1
Timer1 时钟	On	(可选)	Off	TRISx = 1
数字输入	On	(可选)	Off	TRISx = 1
OP0OUT	Off	Off	Off	TRISx = 0
CM0 / CM1 输出	On	(可选)	On	TRISx = 0
SPI / I2C / UARTx 输出	On	Off	On	TRISx = 0
PWM	On	Off	On	TRISx = 0
数字输出	On	Off	On	TRISx = 0

表 2-7 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. ANSELx = 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“ANSELx = 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。
 6. “/PAPU = 1”关闭所有 PAX 端口的“弱上拉”功能。PBx、PCx 和 PDx 没有此类控制位。
 7. /MCLR 使能：PB2 的弱上拉功能自动使能（忽略 WPUB[2]）；读 PORTB[2] 的值为“0”。
 8. 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。
 9. 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。
 10. 当 TRISx = 0 时，通过 IDE 界面可选择读取 PORTx 输出或输入锁存器的值。
 11. ODCONx = 1：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。
 12. 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。
- INT 和 PORTA 端口变化中断的设置，请参阅 [章节 9](#)“中断”。

2.3. 引脚输出优先级

每个 I/O 引脚均复用多个功能，当相应模块都使能输出时，输出优先级从低到高如 表 2-8 所示。因为输入连接到各个功能模块，所以输入不存在优先级问题。

名称	优先级 0	优先级 1	优先级 2	优先级 3	优先级 4
PA0	PA0	-	-	-	-
PA1	PA1	OSC1	-	-	-
PA2	PA2	CLKO	OSC2	-	-
PA3	PA3	P1A0N	-	-	-
PA4	PA4	P0AN	P1A0	-	-
PA5	PA5	P0A	P1B0	-	-
PA6	PA6	P1C0	CM1OUT	-	-
PA7	PA7	[P1D2]	CM0OUT	-	-
PB0	PB0	P1D0	-	-	-
PB1	PB1	[P1A1]	[OP0OUT]	-	-
PB2	PB2	[I2C_SCL]	-	-	-
PB3	PB3	[UART0_TX]	[I2C_SDA]	-	-
PB4	PB4	[SPI_MISO]	-	-	-
PB5	PB5	-	-	-	-
PB6	PB6	[P1B1]	ISPCLK2	-	-
PB7	PB7	I2C_SCL	OP0OUT	ISPDAT2	-
PC0	PC0	I2C_SDA	[SPI_MOSI]	[UART0_TX]	-
PC1	PC1	[P1A2]	[I2C_SDA]	UART0_TX	-
PC2	PC2	[P1A2N]	[I2C_SCL]	[UART1_TX]	-
PC3	PC3	-	-	-	-
PC4	PC4	-	-	-	-
PC5	PC5	SPI_NSS	[P1A1N]	[UART1_TX]	ISPDAT1
PC6	PC6	SPI_SCK	[SPI_NSS]	[P1C1]	ISPCLK1
PC7	PC7	SPI_MISO	[P1D1]	UART1_TX	ISPDAT1
PD0	PD0	-	-	-	-
PD1	PD1	-	-	-	-
PD2	PD2	-	-	-	-
PD3	PD3	-	-	-	-
PD4	PD4	[SPI_SCK]	-	-	-
PD5	PD5	SPI_MOSI	[CLKO]	-	-

表 2-8 管脚输出优先级

3. 上电复位 (POR)

上电过程，即 V_{DD} 从低于 Power-On-Reset 电压 (V_{POR}) 上升至高于 V_{POR} 的过程。当 CPU 重新上电时， V_{DD} 可能没有完全掉电至 0V。

1. 当 V_{DD} 低于 V_{POR} 时，CPU 处于完全复位状态。
 - a. 所有校准配置寄存器不复位。除 INDF、Z、HC、C、FSR、TMR0、PORTx、FOSCCALL/H、ADRESL/H、SPICFG、SPISCR、SPIIER、SPICR1、I2CCR1/2、I2CADDR、I2CCCR、I2CDAT、URxSTAT、URxDATL/H、URxCR1/2、URxBRRL/H、SECCODE、LVDTRIM 和 SRAM 以外 (参阅 [章节 22](#) “特殊功能寄存器”) 的其他特殊功能寄存器 (Special Function Registers, SFR) 均处于复位状态。而不复位的寄存器如 SRAM，将保持其数据直至 V_{DD} 降到 0.6V (典型值)，当 V_{DD} 低于 0.6V 时，其值为不确定值。
 - b. 程序计数器 PC = 0x00，指令寄存器 = “NOP”，堆栈指针 = “TOS” (栈顶)。
2. 当 V_{DD} 上升至 V_{POR} 以上时，芯片开始初始化配置 (BOOT) 过程。
3. 初始化配置完成后，指令将从 PC = 0x00 地址开始执行。

常温 (25°C) 下， V_{POR} 的典型值 ~1.6V，低温 (-40°C) 上升至 ~1.9V。当 $V_{DD} \geq V_{POR}$ 时，CPU 即可在较低的速度 8 MHz / 2T 下正常工作，因此 V_{DD} 的工作范围随温度变化而自动调整。此特性对于电池供电系统来说很重要，在典型的电池工作环境中，当电池电压低至 ~1.6V 时，CPU 仍可工作，从而提高电池使用寿命。

注：

1. V_{POR} 不可配置。
2. POR 的硬件电路默认为开启状态，当 V_{DD} 电压低于 V_{POR} 时即执行芯片电源复位，而不是仅在上电时执行。

3.1. 初始化配置时序

名称	功能	默认
PWRTEB	上电延时定时器, 初始化配置完成后额外延时~64ms	关闭
CSUMENB	程序空间校验和的验证功能	关闭

表 3-1 初始化配置

以上 2 种初始化配置，均由 IDE 界面设置，不能通过指令修改。初始化配置过程：

1. CPU 空闲等待 ~8ms；
2. 从非易失性存储器中加载初始化配置寄存器值，该过程 ~2ms。这些寄存器值由 IDE 预先设置，不受指令影响；
3. 如果使能上电延时定时器 (Power-On-Timer, PWRT)，CPU 将额外空闲等待 ~64ms。
4. 如果使能校验和 (Checksum, CSUM)，该功能将对整个程序空间进行累加并校验。
 - a. 如果校验失败，CPU 将从空闲等待 ~8ms 开始重新启动初始化配置过程；
 - b. 如果校验成功，且没有其他复位条件限制，则 CPU 开始执行指令；

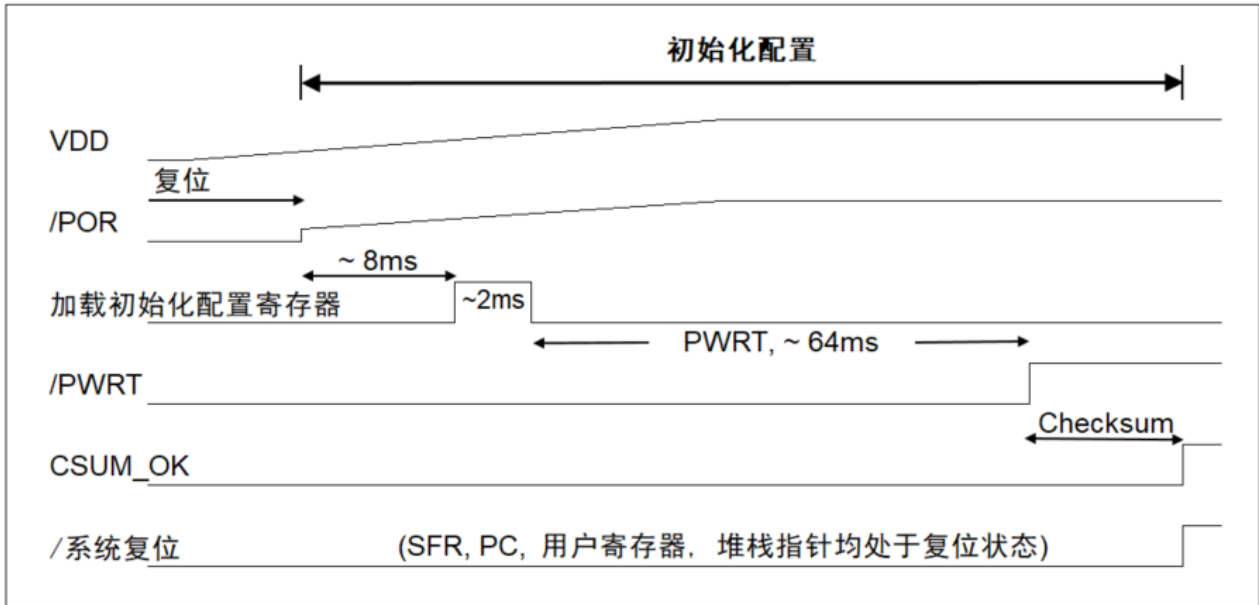


图 3-1 上电时序 (PWRT 和 Checksum 使能)

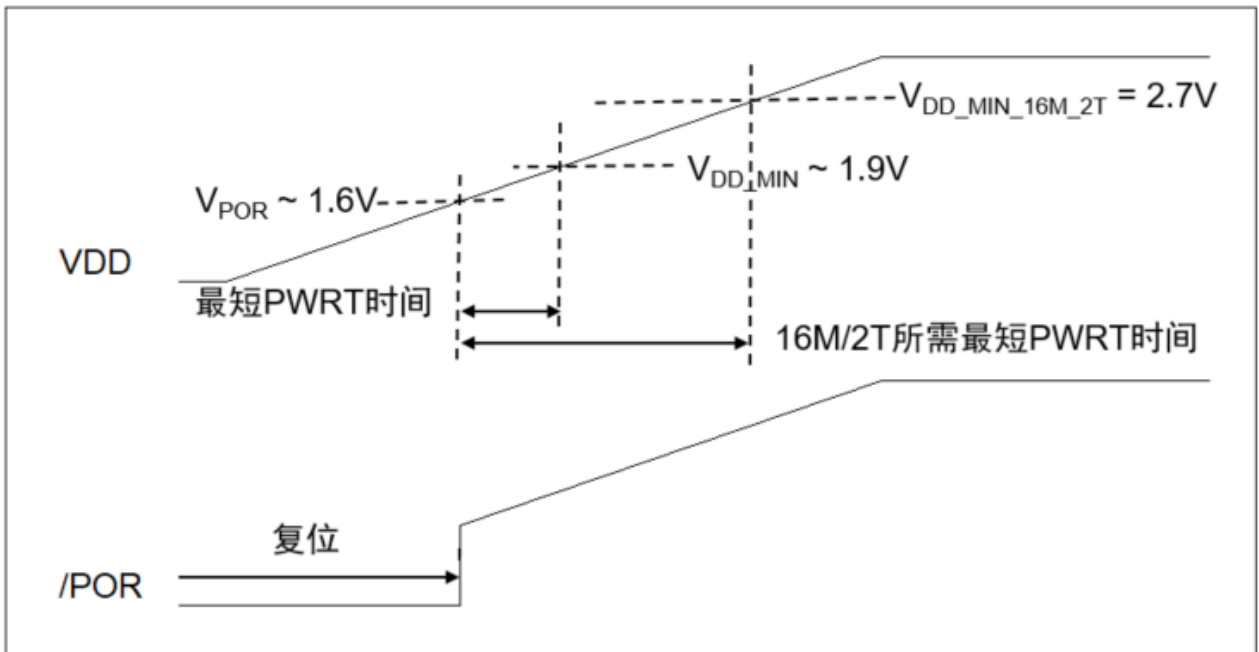


图 3-2 上电过程所需最小 PWRT 时间

如果 CPU 需要在 16MHz / 2T 的速度下运行，那么在初始化配置(BOOT)结束时 V_{DD} 必须高于 2.7V。通过使能 PWRT，可使初始化配置时间从 $\sim 10\text{ms}$ 增加至 $\sim 74\text{ms}$ ，从而提高电源系统的稳定时间。

在以 16MHz / 2T 的速度运行时，应使能 LVR 且设置 $V_{\text{BOR}} \geq 2.7\text{V}$ 。另外，可通过指令控制 LVR 使能的频率以不时地监测 V_{DD} ，而无需一直使能(参阅“LVREN”，“SLVREN”)以降低功耗。

注:

1. V_{DD} 上电过程不可以太慢，另外不建议 V_{DD} 的电容 $C_{\text{VDD}} \geq 22 \mu\text{F}$ ；
2. V_{DD} 电容值以 1 到 $10 \mu\text{F}$ 为佳。出于 EFT 性能考虑， $C_{\text{VDD}} < 1 \mu\text{F}$ 可能太小；
3. 如果可以接受启动延时，那么建议使能 PWRT 以提高 CPU 的稳定性；

4. 系统复位

与 POR 不同，系统复位(system reset) 并不会完全复位。系统复位时，CPU 是否启动初始化配置过程则取决于复位触发类型。若启动初始化配置则空闲等待~8ms，然后重新加载初始化配置寄存器值，如果使能 PWRT 将额外延时~64ms，随后系统正常启动。在系统复位中：

- 除初始化配置寄存器外，POR 过程中被重置的寄存器在系统复位时也会被同样重置；
- 程序计数器 PC = 0x00，指令寄存器 = “NOP”，堆栈指针 = “TOS” (栈顶)；

除仿真调试的 OCD(On-Chip Debugger) 模块外，以下 7 种事件可触发系统复位：

1. 欠压复位 (BOR / LVR) – 总会启动初始化配置；
2. 非法指令复位 – 如果使能 “IRBTE” 则启动初始化配置；
3. 看门狗复位 (WDT) – 如果 CPU 处于非 SLEEP 状态且使能 “WDTBTE” 则启动初始化配置；
4. 外部 I/O 复位 (/MCLR) – 如果使能 “MRBTE” 则启动初始化配置；

注： 如果可以接受更长的系统重启时间，则建议使能初始化配置过程(BOOT)以提高系统的稳定性。

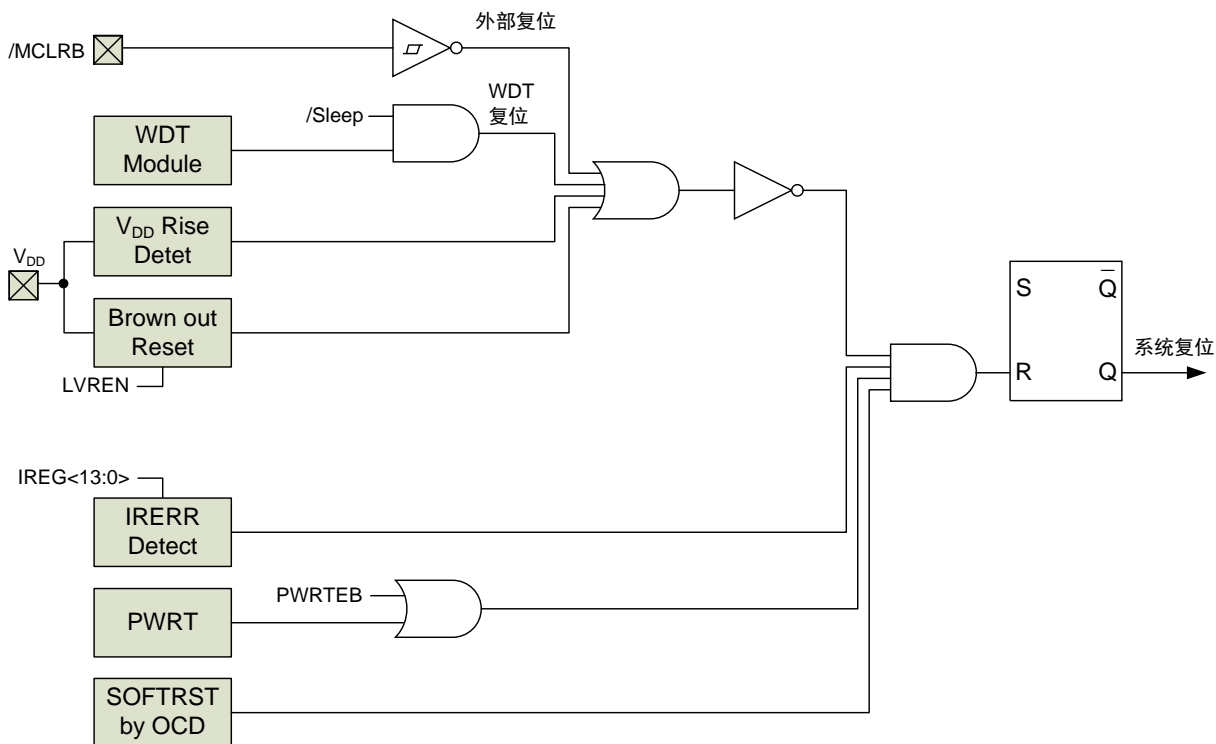


图 4-1 复位电路结构框图

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 <ul style="list-style-type: none"> 使能 关闭 非 SLEEP 模式下使能 通过指令控制 (SLVREN) 	关闭
WDTE	WDT <ul style="list-style-type: none"> 使能 (指令不能禁止) 由指令控制 (SWDTEN) 	SWDTEN 控制
MCLRE	外部 I/O 复位	关闭
WDTBTE	WDT 复位启动初始化配置过程	关闭
IRBTE	非法指令复位启动初始化配置过程	关闭
MRBTE	MCLRE 复位启动初始化配置过程	关闭

表 4-1 复位相关初始化配置寄存器

4.2. 欠压复位 (Brown-Out Reset, LVR / BOR)

当 V_{DD} 值降低并低于预设的欠压值 (V_{BOR}) 超过 T_{BOR} 时间时，就会出现欠压状态。消抖时间 T_{BOR} 大概为 3 到 4 个 LIRC 或 HIRC 时钟周期 (参阅 “LVRDEB” 和 “LVRCKS”，如果未预先启动，LIRC 或 HIRC 将自动开启)。当 $V_{DD} \leq V_{BOR}$ 时，CPU 保持系统复位状态，直至 $V_{DD} > V_{BOR}$ 时 CPU 开始初始化配置过程(BOOT)。

V_{POR} 值不能配置，而 V_{BOR} 值可以设置为 2.0、2.2、2.5、2.8、3.1、3.6、4.1V (参阅 “LVRS”，表 4-1)。

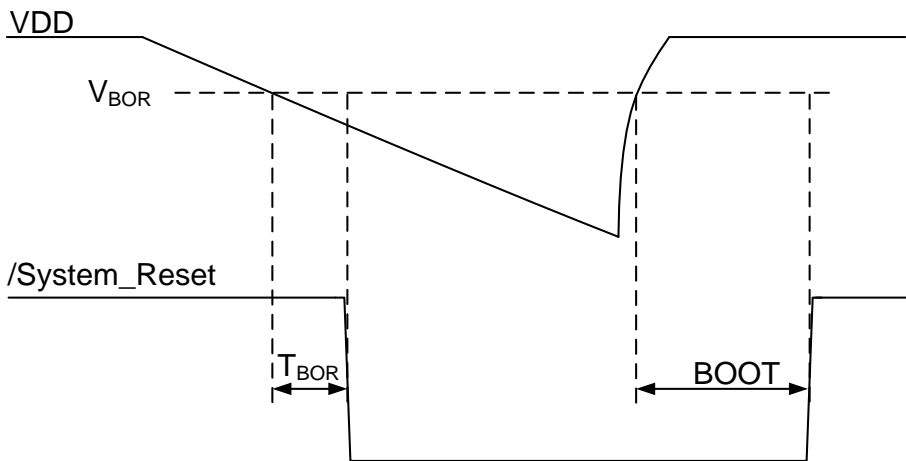


图 4-2 LVR 初始化配置时序图

LVR 可配置成 4 种不同功能(参阅“LVREN”，表 4-1)。

1. LVR 使能；
2. LVR 关闭；
3. 非 SLEEP 模式下 LVR 使能；
4. 由指令控制使能或关闭 LVR (SLVREN, 表 4-2)。

注： SLEEP 模式下，可通过指令关闭 LVR 以降低功耗。但如果系统 V_{DD} 不稳定，CPU 应定时唤醒并使能 LVR 来监测 V_{DD} 。

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		MSCON0[3]	0x10C	RW-0
LVRDEB ²	LVR 消抖	1 = 使能 0 = 禁止	MSCON1[3]	0x101 / 0x301	RW-1
LVRCKS ²	LVR 消抖时钟	1 = HIRC 0 = LIRC	MSCON1[2]		RW-0
LFMOD	LIRC	1 = 256 kHz 0 = 32 kHz	OSCCON[7]	0x8F	RW-0
LVRADJ ²	LVR 修调位, ~3% / step		LVDTRIM[2:0]	0x21C	RO-xxx

表 4-2 LVR 用户寄存器

4.3. 非法指令复位 (Illegal Instruction Reset)

CPU 获取指令错误的原因有很多，最常见的为干扰和 V_{DD} 不稳定。

虽然没有专用的复位指令，但任何蓄意的非法指令都等同于复位指令。出现非法指令时将产生系统复位，随后是否启动初始化配置则取决于 IDE 设置 (参阅“IRBTE”，表 4-1)。

4.4. 看门狗定时器 (Watch Dog Timer, WDT) 复位

SLEEP 模式下，WDT 溢出将导致唤醒。

正常模式 (非 SLEEP 模式)下，WDT 溢出将触发系统复位，随后是否产生初始化配置则取决于 IDE 设置 (参阅“WDTBTE”，表 4-1)。WDT 复位可用于复位挂起的 CPU。应在程序中不时地清除 WDT 以避免错误复位。

关于 WDT 的操作和设置等细节，请参阅 章节 7.1 看门狗定时器 (Watch Dog Timer, WDT)。

¹ 发生欠压复位时，此位不会清 0。其它复位会将此位清 0。

² 只受上电复位影响，其它复位不起作用。

4.5. 外部 I/O 系统复位 /MCLR

如果已相应设置初始化配置寄存器，那么可通过在/MCLR (PB2)脚上施加低电压来使 CPU 复位。/MCLR 脚通常经过一个电阻弱上拉到 V_{DD}，而不是直接连到 V_{DD}，如 图 4-3 所示，建议采用外部 RC 电路以提供故障滤波和过流保护。

/MCLR 系统复位后是否产生初始化配置则取决于 IDE 设置 (参阅“MRBTE”，表 4-1)。

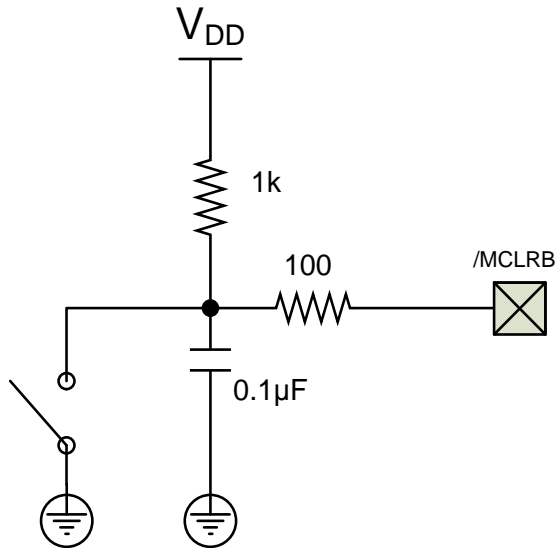


图 4-3 /MCLR 复位电路

4.6. 检测上次复位类型

通过 4 个状态标志位 /POR、/BOR、Time Out (/TF)、Power Down (/PF)的不同组合可以追溯上一次系统复位的类型，但“正常模式下的/MCLR 系统复位”和“非法指令复位”除外。这些状态标志位均需由指令置 1。复位后，相应的标志位将被锁定为“0”。

复位源	/POR	/BOR	/TF	/PF
	PCON[1]	PCON[0]	STATUS[4]	STATUS[3]
	0x8E		Bank 首地址 + 0x03	
POR	0	(未知)	1	1
LVR	-	0	1	1
正常模式下(非 SLEEP) WDT 溢出(复位)	-	-	0	-
SLEEP 模式下 WDT 溢出(唤醒)	-	-	0	0
SLEEP 模式下/MCLR 复位	-	-	1	0
正常模式下(非 SLEEP) /MCLR 复位	-	-	-	-
非法指令复位	-	-	-	-
片上调试 (OCD)	-	-	-	-

表 4-3 复位相关状态标志位 (“-” 无变化)

5. 低电压检测/比较器 (LVD)

LVD 的工作原理与 LVR 类似，但以下几点除外：

- 所有控制位和参数选择位均由用户指令设置，而不是初始化配置时；
- 必须正确设置 I/O： $\text{TRISx} = 1$; $\text{ANSELAx} = 1$ ；
- LVD 事件将置位 LVDW 而不是/BOR；
- 可通过指令配置为中断，且不会触发系统复位；
- 可通过 LVDDEB 使能消抖功能。消抖时间(T_{LVD}) 为 3 – 4x 消抖时钟 LVDCKS (如果未预先启动，所选时钟源将自动开启)；
- LVD 输入可以配置成 V_{DD} 或其他 2 个 I/O, 后者允许将 LVD 当作单输入比较器功能使用, 与 9 档 LVDL 电压值($V_{\text{LVD-REF}}$) 之一进行比较；
- 可设置 LVD 的极性，因此 LVD 可作为 $V_{\text{LVD-REF}}$ 的“高”或“低”比较器；
- 9 档可选 LVD 电压(LVDL)，可通过指令实现~2%/step 的微调精度；

5.1. LVD 相关寄存器汇总

名称	状态	寄存器	地址	复位值
LVDL	$V_{\text{LVD-REF}}$ 0000 = 1.8 0101 = 3.3 0001 = 2.0 0110 = 3.6 0010 = 2.4 0111 = 4.0 0011 = 2.7 1xxx = 1.2 0100 = 3.0	PCON[7:4]	0x8E	RW-0000
LV DEN	<u>LVD</u> 1 = 使能 0 = 关闭	PCON[3]		RW-0
LVDW	<u>LVD 触发?</u> 1 = Yes (不锁存) 0 = No	PCON[2]		RO-x
LFMOD	<u>LIRC</u> 1 = 256 kHz 0 = 32 kHz	OSCCON[7]	0x8F	RW-0
LV DCKS	LVD 消抖时钟 00 = <u>LIRC</u> 10 = Sysclk 01 = <u>HIRC</u> 11 = 指令时钟	LV DCON[6:5]		RW-00
LV DP	LVDW 极性 1 = 检测电压 > $V_{\text{LVD-REF}}$ 0 = 检测电压 < $V_{\text{LVD-REF}}$	LV DCON[4]		RW-0
LV DDEB	LVD 消抖 1 = 使能 0 = 关闭	LV DCON[3]	0x21A	RW-1
LV DM	<u>LVD 检测源</u> 00 = <u>PB3</u> 01 = VDD 1x = PD5	LV DCON[1:0]		RW-00
LV DIE	<u>LVD 中断</u> 1 = 使能 0 = 关闭	PIE1[5]	0x8C	RW-0
LV DIF	<u>LVD 产生中断?</u> 1 = Yes 0 = No, 或已被清零	PIR1[5]	0x0C	RW0-0
LV DADJ	LVDL 修调位, ~2% / step	LV DTRIM[6:3]	0x21C	RW-xxxx

表 5-1 LVD 用户设置和标志寄存器

6. 振荡器和系统时钟

系统时钟 (SysClk) 可通过指令选择为内部高速振荡器 HIRC, 内部低速振荡器 LIRC, 或外部振荡器 (EC, LP, XT, 参阅 “SCS”, 表 6-2)。如果选择外部振荡器, 那么由初始化配置寄存器 “FOSC” (表 6-1) 选定 3 种外部振荡器之一。系统时钟还可通过指令进一步选择为内部振荡器的分频 (参阅 IRCF, 表 6-2)。系统时钟用于产生指令时钟 (Instruction Clock):

$$\text{指令时钟} = \text{SysClk} / N; N = 2 \text{ for } 2T, 4 \text{ for } 4T.$$

外部时钟输入引脚由初始化配置寄存器设置 (参阅 FOSC)。

内部指令时钟输出功能可由初始化配置寄存器 (参阅 FOSC) 或指令 (参阅 SCKEN 和 SCKOE) 设置。如果使能指令时钟输出, 那么则由 “CLKOS” (表 6-3) 选择输出端口为 PA2 或 PD5。

Timers 和 ADC 模块有独立的振荡器, 因此可有多个振荡器同时运行。

当 Timers 使能时, 其选用的振荡器将自动开启, 且在 Timers 运行期间一直保持有效。SLEEP 模式下, 可将其振荡器配置为开启或关闭。当相应的振荡器在 SLEEP 模式下保持运行时, ADC, Timers 和 PWM 功能同样可在 SLEEP 下保持工作。

SLEEP 模式下指令停止运行, 指令时钟默认停止, 时钟输出暂停。当 SYSON = 1 时, 指令时钟将保持运行, 因此选择指令时钟作为时钟源的外设模块也将在 SLEEP 模式下保持工作, 此外时钟输出也将继续。

注: 当 ADC 转换时钟源选择 LIRC 时, 进入 SLEEP 模式后, LIRC 将保持运行, 与 SYSON 无关;

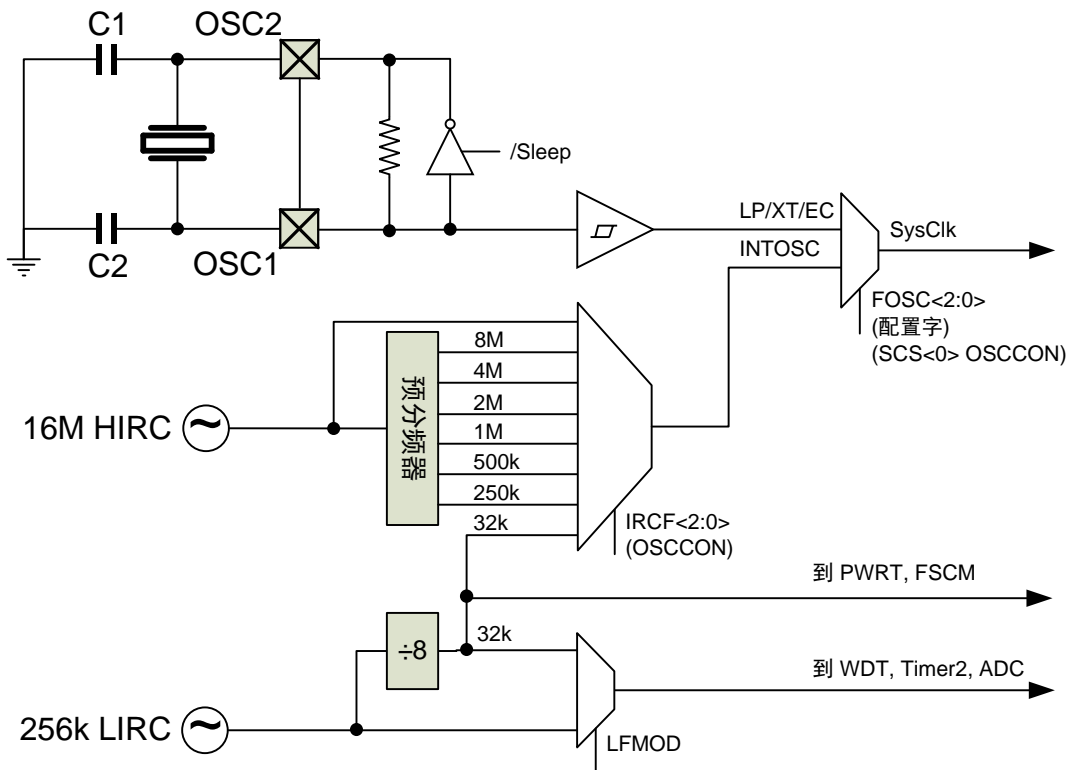


图 6-1 系统时钟 SysClk 的时钟源框图

6.1. 振荡器模块相关寄存器汇总

名称	功能	默认
FOSC	<ul style="list-style-type: none"> LP: PA1 (+) 和 PA2 (-) 接外部低速晶振 XT: PA1 (+) 和 PA2 (-) 接外部高速晶振 EC: PA1 (+) 接外部时钟输入, PA2 为 I/O INTOSC: PA2 或 PD5 输出"指令时钟", PA1 为 I/O INTOSCIO: PA1 和 PA2 为 I/O 	INTOSCIO
IESO	<p><u>XT / LP 双速时钟启动</u></p> <ul style="list-style-type: none"> 使能 关闭 	使能
FCMEN	<p><u>故障保护时钟监控器</u></p> <ul style="list-style-type: none"> 使能 关闭 	使能
TSEL	<p><u>指令时钟与系统时钟的对应关系 (2T or 4T)</u></p> <ul style="list-style-type: none"> 2 (指令时钟 = Sysclk/2) 4 (指令时钟 = Sysclk/4) 	2

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	IRCF	LFMOD	OST	
		OSCCON[0]	OSCCON[6:4]	OSCCON[7]	(固定值)	
		0x8F				
RW-0	RW-100	RW-0				
外部	EC	0	-	-	-	
	XT	0	-	-	1,024	
	LP	0	-	-	32,768	
内部	HIRC	16 MHz	1	111	-	-
		8 MHz	1	110	-	-
		4 MHz	1	101	-	-
		2 MHz	1	100	-	-
		1 MHz	1	011	-	-
		500 kHz	1	010	-	-
	LIRC	250 kHz	1	001	-	-
		256 kHz ¹	1	000	1	-
		32 kHz ²	1	000	0	-

表 6-2 SysClk 系统时钟源设置相关用户寄存器

¹ 256 kHz LIRC (参阅 LFMOD) 只供 WDT(参阅 WCKSRC, 表 7-4)、Timer2 (参阅 T2CKSRC, 表 7-9)、ADC (参阅 ADCS, 表 12-2)、LVR (参阅 LVRCKS, 表 4-2) 和 LVD (参阅 LVDCKS, 表 5-1) 使用。

² 系统时钟源 (IRCF=000)、PWRT、LIRC 和 HIRC 交叉校准、FSCM 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]	0x8F	RO-x
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		LVDCON[7]	0x21A	RW-0
SCKEN ³	内部系统时钟输出	1 = 使能 0 = 禁止	SCKCFG[7]	0x209	RW-0
SCKOE	当 SCKEN = 1 时, 系统时钟输出到 I/O 1 = PA2 或 PD5 (取决于 CLKOS) 0 = 禁止		SCKCFG[1]		RW-0
CLKOS	内部时钟输出引脚 1 = CLKO 映射到 PD5 0 = CLKO 映射到 PA2		MSCON0[4]	0x10C	RW-1
HIRCM	HIRC 频率选择 1 = 13.5 MHz 0 = 16 MHz		MSCON1[0]	0x101/ 0x301	RW-0
FOSCCAL	内部高速时钟 HIRC 频率调节寄存器 注: bit[4] 及 bit[2:0] 需保持为 1;	高 5 位	FOSCCALH[4:0]	0x309	RW-x xxxx
		低 4 位	FOSCCALL[7:0]	0x109	RW-xxx1 x111
CKMAVG	LIRC 和 HIRC 交叉校准时 4 次平均测量模式	1 = 使能 0 = 关闭	MSCON0[2]	0x10C	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)		MSCON0[1]		RW-0
SOSCPR	校准 LIRC 周期所需的 HIRC 周期数		SOSCPR[11:0]	0x219[3:0] 0x218[7:0]	RW-FFF

表 6-3 振荡器控制/状态位

³ 在修改 SCKEN 和 SCKOE 之前需先将 EECON1.WREN3/2/1 位置 1, 写之后需清零 EECON1.WREN3/2/1, 避免误操作。

名称	状态		寄存器	地址	复位值
GIE	全局中断	1 = 使能 (PEIE, CKMIE, 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 (CKMIE, OSFIE 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[6]	0x8C	RW-0
OSFIE	外部振荡器故障中断	1 = 使能 0 = 关闭 (无唤醒)	PIE1[2]		RW-0
CKMIF	LIRC和HIRC交叉校准完成标志位	1 = Yes (锁存) 0 = No	PIR1[6]	0x0C	RW-0
OSFIF	外部振荡器故障标志位	1 = Yes (锁存) 0 = No	PIR1[2]		RW-0

表 6-4 振荡器中断使能/状态位

6.2. 内部时钟模式 (HIRC 和 LIRC)

内部高频时钟 (Internal high frequency clock, HIRC) 出厂时已分别校准到 13.5 MHz 和 16 MHz @ 2.5V/25°C。芯片之间的频率变化典型值 < ±1.5% @2.5 – 5.5V/25°C, 温度变化典型值为 < ±3.5% @ -40 – +85 °C。

HIRC 精度在晶圆测试时已进行校准。封装过程可能会导致 HIRC 频率漂移。烧录器软件可选择对 HIRC (仅 16 MHz) 进行重新校准。HIRC 出厂校准值已存储到 “FOSCCAL” 寄存器中, 用户可通过指令修改 “FOSCCAL” 来微调 HIRC 频率 (16 MHz 或 13.5 MHz, 参阅 “HIRCM”), 微调 steps 是非线性的 (~0.2%/LSB)。粗略估计 (以 16 MHz 为例) 如下:

$$\{FOSCCALH[4:0], FOSCCALL[7:5], FOSCCALL[3]\} \pm N \approx 16000 * (1 \pm N * 0.2\%) \text{ (kHz)}$$

注意:

1. FOSCCAL 由 {FOSCCALH[4:0], FOSCCALL[7:5], FOSCCALL[3]} 共 9 位组成, 另外 FOSCCALL[4] 和 FOSCCALL[2:0] 需要保持为 1;
2. 当系统时钟 SysClk 选择 HIRC, 且 TSEL 选择 2T 指令周期时, 建议先将 HIRC 设置成 8 MHz 或者更低 (参阅 IRCF), 通过 FOSCCAL 进行频率微调后, 如需要, 再将 HIRC 设置成 16 MHz, 从而避免在频率微调的过程中超出芯片的工作范围 (16 MHz / 2T, V_{DD} ≥ 2.7V) 或导致其他不可预期的情况。TSEL 选择 4T 指令周期时, 则无上述限制。

内部低频时钟 (Internal low frequency clock, LIRC) 出厂时已校准至 32kHz。芯片之间的频率变化典型值为 < ±6.0% @2.5 – 5.5V/25°C, 温度变化典型值 < ±4.0% @ -40 – +85 °C。

LIRC 和 HIRC 可相互交叉校准 – 在一个 LIRC 周期 (32kHz) 内使用 Timer2 来测量指令时钟数 (SysClk 选择 16MHz HIRC), 此为内置硬件功能。

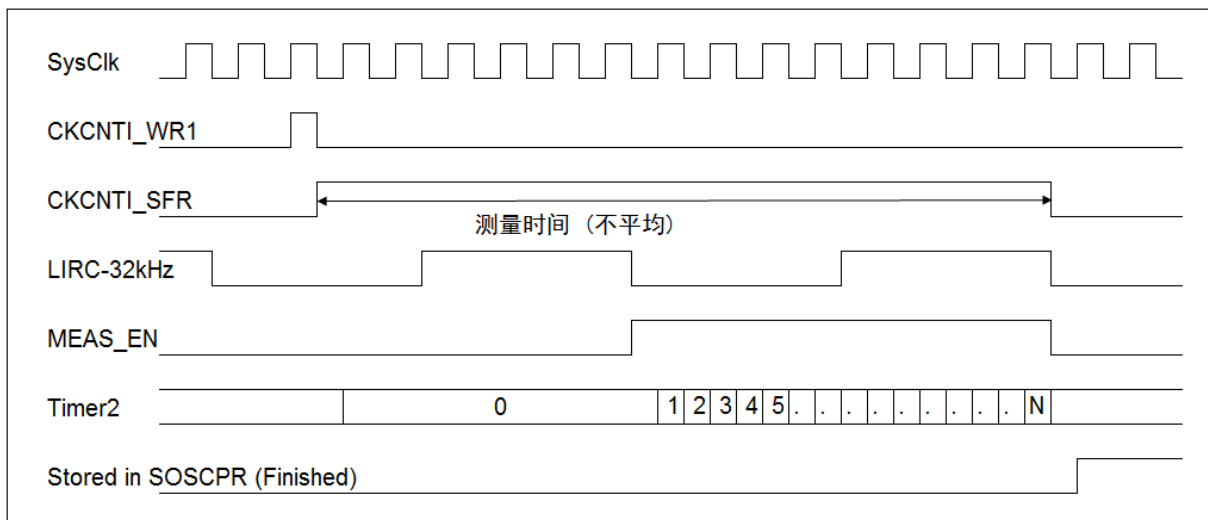


图 6-2 单次测量时序图

LIRC 和 HIRC 交叉校准步骤:

1. 设置 IRCF = 111, SCS = 1 ; SysClk 选择 16MHz HIRC (其他频率设置的精准度会降低)
2. 设置 CKMAVG = 1 ; 4 次测量平均, 选择 0 表示不做平均
3. 设置 TMR2ON = 1 ; 使能 Timer2
4. 设置 CKCNTI = 1 ; 开始校准, 默认 Timer2 预分频比 = 1, 后分频比 = 1,
T2CKSRC = SysClk for 2T; SysClk/2 for 4T
5. 校准完成时, CKCNTI 自动清零("CKCNTI = 0"), CKMIF 自动置位("CKMIF = 1")。
6. 测量值存储在 SOS CPR 寄存器中。
7. LIRC 默认为 32kHz, 且 CPU 运行在 16MHz / 2T 下, 则理想的匹配值为 500。

注:

- LIRC 和 HIRC 交叉校准时, 不要对 SOS CPRH/L 寄存器进行写操作;
- LIRC 和 HIRC 交叉校准时, Timer2 不能被其他外设使用;
- LIRC 和 HIRC 交叉校准功能与 IDE 的单步调试模式不兼容;
- 当 CKCNTI = 1 时, LIRC 自动开启, 且进入 SLEEP 模式后保持运行, 但仅当 SYSON = 1 时, 校准才可在 SLEEP 模式下运行。

6.3. 外部时钟模式 (EC / LP / XT)

6.3.1. EC 模式

外部数字信号作为时钟源连接到 OSC1 脚 (OSC2 用作 I/O)。当 SysClk 选择 EC 模式时, 当 POR 复位或从睡眠中唤醒时, EC 模式不需要设置或转换时间延迟。

6.3.2. LP 和 XT 模式

LP 或 XT 模式下，石英晶体谐振器或陶瓷谐振器作为时钟源连接到 OSC1 和 OSC2 脚。

LP 振荡器模式是 3 种模式 (EC, LP, XT) 中增益设定和电流消耗最低的。该模式仅用于驱动 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最高增益设定。

如果时钟源选择 XT 或 LP 模式，当初始化配置结束或从睡眠中唤醒时，CPU 在振荡器起振定时器 (OST) 计数期间将暂停执行程序，这有利于 XT 或 LP 时钟的稳定。对于 XT 和 LP 模式，OST 分别计数 1,024 和 32,768 个 OSC1 (晶体输入+ve 端)。对于 32.768 kHz 音叉式晶振，OST 计时至少需要 1 秒。

注：

- WDT 将保持清零状态直至 OST 完成计数；
- OST 计数期间，不要对 WDTCON 或 OPTION 寄存器进行写操作，否则将产生不可预期的行为；

双速时钟启动 (参阅“IESO”，表 6-1) 允许 CPU 在 OST 计数期间选择内部振荡器 INTOSC 模式作为 SysClk 进而执行指令。在需要频繁进出睡眠模式的情况下，双速时钟启动功能可使芯片在唤醒后立即执行指令，从而除去外部振荡器所需的起振时间，以降低整机功耗。即 CPU 从睡眠中唤醒，将 INTOSC 作为 SysClk 执行几条指令后，再返回睡眠状态，而无需等待外部振荡器的稳定。

注： EC 模式下双速启动功能关闭，因其振荡器不需要稳定时间。

双速启动时序：

1. 初始化配置结束或从睡眠中唤醒；
2. 选择 INTOSC 作为 SysClk 执行指令直至 OST 超时；
3. SysClk 从 INTOSC 的下降沿一直保持为低，直至新时钟的下降沿到来 (LP 或 XT 模式)；
4. SysClk 切换到外部时钟源；

振荡器起振超时状态位 (OSTS) 用于提示 SysClk 运行在外部时钟源下或内部时钟源下。当开启双速时钟启动功能时，通过 OSTS 可间接查询 LP 或 XT 模式下的振荡器起振定时器(OST)是否已经超时。

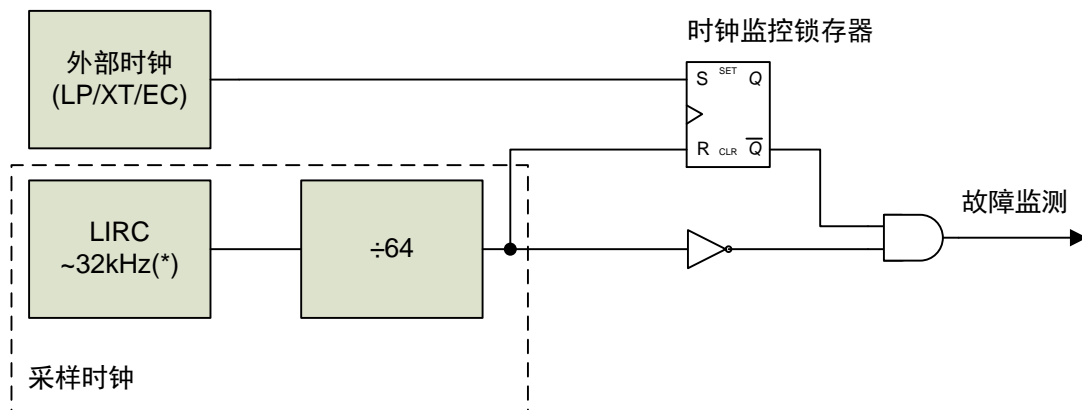
执行 SLEEP 指令将中止 OST 计时，而 OSTS 将保持为“0”。

故障保护时钟监控器 (Fail-Safe Clock Monitor, FSCM, 由“FCMEN”使能，参阅表 6-1) 可使芯片在外部振荡器出现故障时仍能继续工作。在振荡器起振定时器 (OST) 超时后，FSCM 随时检测振荡器故障。FSCM 适用于所有外部振荡器模式(EC, LP 和 XT)。当选择外部振荡器时，建议使能 FSCM 功能。

如果外部振荡器的振荡频率在~1 kHz 及以下时，则判定其出现故障。由 LIRC 除以 64 产生采样时钟，故障监控器内部有一个锁存器，外部时钟在其每个下降沿将锁存器置 1，采样时钟在其每个上升沿将锁存器清 0。当采样时钟的整个半周期结束而主时钟依然未进入低电平时，即检测到故障。

当外部时钟出现故障时，FSCM 自动将 SysClk 切换成内部时钟源并置位 OSFIF。如果 OSFIE 使能，OSFIF 置 1 将产生中断。芯片固件随后应采取措施来减轻可能由故障时钟所导致的问题。SysClk 将继续运行在内部时钟源下，直到芯片固件成功重启外部振荡器。

由“IRCF”决定“FSCM”所用的内部时钟源，这使得内部振荡器在外部时钟发生故障前就得以配置。



注：LFMOD不影响采样时钟。

图 6-3 FSCM 结构框图

复位、执行 SLEEP 指令或翻转 SCS 位后，故障保护条件将被清除。当 SCS 位被修改后，OST 将重新启动计时。OST 运行期间，CPU 选择 INTOSC 模式作为 SysClk 继续执行指令。OST 超时后，故障保护条件被清除，芯片将切换回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。

注：任何可能由双速时钟启动或故障保护时钟监控器所产生的时钟自动切换，都不会更新 SCS 位。程序应监控 OSTS 位以确定当前的 SysClk 系统时钟源。

6.4. HIRC, LIRC 和 EC 时钟的内部切换

图 6-4 为时钟内部切换时序图。若 HIRC 或 LIRC 在切换前已经关闭 (为了省电)，则将会有额外的振荡器设置延迟时间，可通过 HTS 和 LTS 标志位查询相应振荡器的状态。

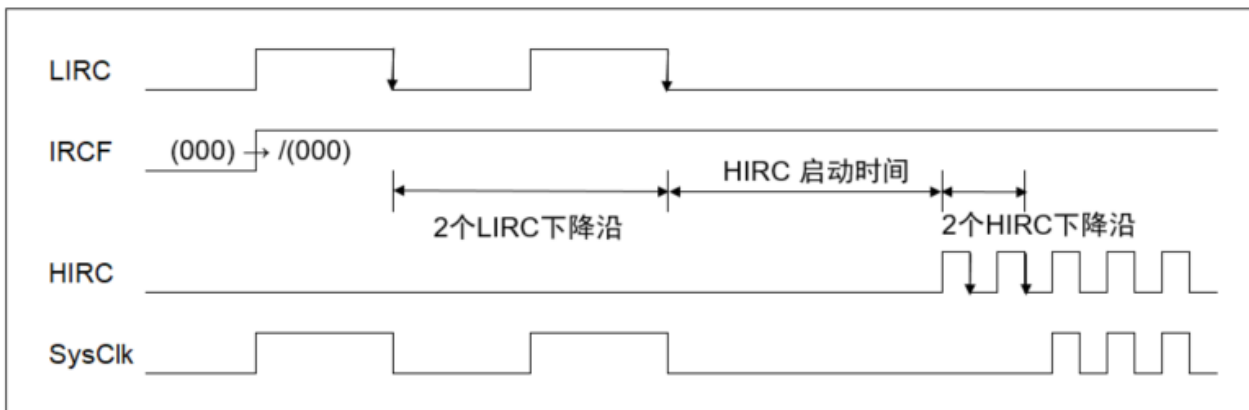


图 6-4 由 LIRC 切换到 HIRC 时序图 (同样的原则页适用于 EC, LIRC, HIRC 之间的切换)

7. 定时器 (TIMERS)

共有 4 个定时器，包括看门狗定时器(WDT)在内。

	WDT	Timer0	Timer1	Timer2
预分频器 (位)	–	8 (与 WDT 共用)	4 (1x, 4x, 16x, Timer1 和 Timer2 共用)	
计数器 (位)	16	8	12	16
后分频器 (位)	7 (与 Timer0 共用)	–	–	4 (1 – 16x)
时钟源	<ul style="list-style-type: none"> • LP • XT • HIRC • <u>LIRC</u> 	<ul style="list-style-type: none"> • LP • XT • HIRC • <u>指令时钟</u> • PC5/T0CKI (转变沿计数器) 	<ul style="list-style-type: none"> • LP • XT • HIRC • <u>指令时钟</u> 	<ul style="list-style-type: none"> • LP • XT • HIRC • <u>指令时钟</u> • LIRC • 2x 指令时钟 • 2x HIRC • 2x (EC, LP or XT)

表 7-1 定时器资源

注：如果定时器的时钟源不是指令时钟，在更改 TMRx 之前需先设置 “TMRxON = 0”。

当定时器使能时，其所选的时钟源会自动开启。指令时钟在 SLEEP 模式下被关闭，因此不能用于 WDT，当 SYSON = 1 时，指令时钟将保持运行。当定时器选择 LP / XT 振荡器作为时钟源时，FOSC 必须相应配置成 LP / XT 模式或选择 INTOSCIO 模式，否则 LP / XT 振荡器将处于关闭状态，不会产生计数。

WDT 的后分频器(postscaler)和 Timer0 的预分频器(prescaler)共用同一个硬件分频电路。该硬件电路由指令选择分配给 WDT 或 Timer0，但二者不能同时使用。未被分配分频器的定时器，其分频比值为“1”。同样的规则也适用于共用同一个预分频器电路的 Timer1 和 Timer2。

当 POR 或系统复位时，除 Timer0 的计数器外，其他所有定时器的计数器、预分频器和后分频器都将复位。以下事件也将复位相应定时器的计数器和分频器：

	WDT	Timer0	Timer1	Timer2
预分频器	–	<ul style="list-style-type: none"> • 写 TMR0 • PSA 切换 	<ul style="list-style-type: none"> • T1ON = 0 & T1CKPSA = 1 • 写 TMR1L/H • 写 T1CON0 	<ul style="list-style-type: none"> • TMR2ON = 0 & T1CKPSA = 0 • 写 TMR2L/H • 写 T2CON0
			<ul style="list-style-type: none"> • LIRC 和 HIRC 交叉校准启动 • T1ON = 0 & TMR2ON = 0 	
计数器	<ul style="list-style-type: none"> • WDT, OST 溢出 • 进入/退出 SLEEP • CLRWDT • 写 WDTCON 	<ul style="list-style-type: none"> • Timer0 溢出 	<ul style="list-style-type: none"> • TMR1 = PR1 (匹配) 	<ul style="list-style-type: none"> • TMR2 = PR2 (匹配)
后分频器	<ul style="list-style-type: none"> • 除写 WDTCON 外的以上所有条件 • PSA 切换 	–	–	<ul style="list-style-type: none"> • 除(T1ON = 0 & TMR2ON = 0)外的以上所有条件

表 7-2 定时器的计数器和分频器的重置事件

一旦发生 PWM 故障刹车，且刹车事件一直保持，那么 Timer2 的计数器、预分频器和后分频器将停止递增，在故障刹车事件清除后将自动恢复。

PWM 单脉冲输出结束时将自动设置 “TMR2ON = 0”，通过设置 “TMR2ON = 1” 可重新启动 Timer2。

7.1. 看门狗定时器 (Watch Dog Timer, WDT)

WDT 用于“从 SLEEP 中唤醒”或“CPU 挂起时产生系统复位”。当 WDT 计数到预设数量的时钟周期数时则产生溢出。

- SLEEP 模式下，WDT 溢出将触发唤醒。CPU 将从其进入 SLEEP 前的位置恢复操作。唤醒不是中断，也不是系统复位事件。
- 正常模式(非 SLEEP 模式)下，WDT 溢出将触发系统复位 (参阅 章节 4 系统复位)。随后是否产生初始化配置，则取决于 WDTBTE 设置。

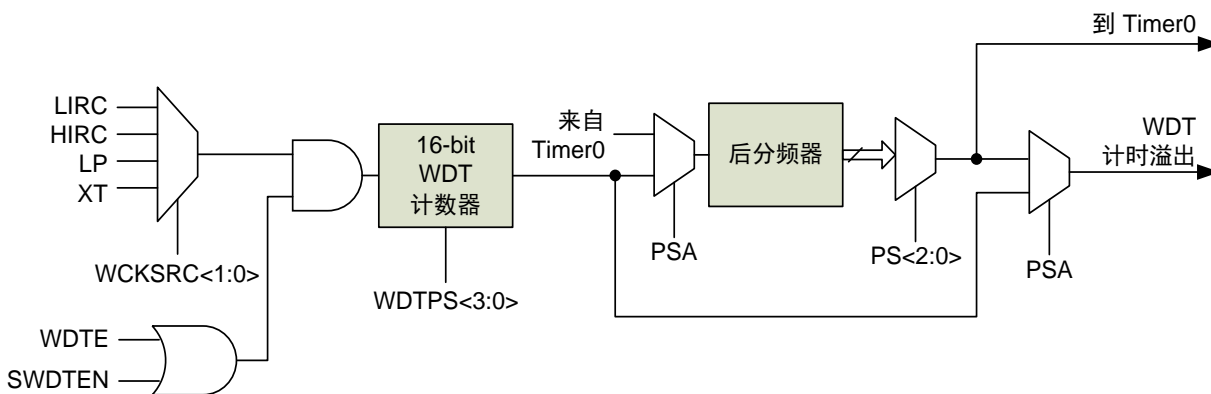


图 7-1 WDT 结构框图

计时超过看门狗定时时间：WDT-周期 x WDT-预分频比 / WDT 时钟频率，WDT 将溢出。

对于选定的时钟源，由于 WDT 预分频器的二进制特性，看门狗定时时间步长呈连续的倍数关系。选用 LIRC 作为时钟源时，WDT 溢出前可设置的最长定时时间为：

$$2^{16} \times 2^7 / 32\text{kHz} = \sim 262 \text{ seconds.}$$

7.1.1. WDT 相关寄存器汇总

名称	功能	默认
WDTE	<u>WDT</u> • 使能 (指令不能禁止) • <u>通过指令控制 (SWDTEN)</u>	SWDTEN 控制
WDTBTE	WDT 复位启动初始化配置	关闭

表 7-3 WDT 选项初始化配置寄存器

名称	状态		寄存器	地址	复位值	
WCKSRC	<u>WDT 时钟源</u>		WDTCON[6:5]	0x99	RW-00	
	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					
WDTPS	<u>WDT 周期</u>		WDTCON[4:1]	0x99	RW-0100	
	0000 = 32 0111 = 4,096 0001 = 64 1000 = 8,192 0010 = 128 1001 = 16,384 0011 = 256 1010 = 32,768 0100 = <u>512 (默认)</u> 1011 = 65,536 0101 = 1,024 11xx = 65,536 0110 = 2,048					
SWDTEN	1 = WDT 使能 0 = <u>WDT 关闭</u> (当 WDTE 选择由 SWDTEN 控制时)		WDTCON[0]		RW-0	
LFMOD	1: LIRC = 256 kHz 0: <u>LIRC = 32 kHz</u>		OSCCON[7]	0x8F	RW-0	
PSA	1 = <u>分频电路分配给 WDT 后分频器</u> 0 = 分频电路分配给 Timer0 预分频器		OPTION[3]		RW-1	
PS		<u>WDT 后分频比</u>	<u>Timer0 预分频比</u>	OPTION[2:0]	0x81	RW-111
	000	1	2			
	001	2	4			
	010	4	8			
	011	<u>(PSA=1)</u> 8	<u>(PSA=0)</u> 16			
	100	16	32			
	101	32	64			
	110	64	128			
	111	<u>128</u>	<u>256</u>			
xxx	<u>(PSA =0)</u> 1	<u>(PSA =1)</u> 1				

表 7-4 WDT 相关用户寄存器

7.1.2. WDT 的设置和使用

由 WDTE (初始化配置寄存器) 以及 SWDTEN (用户寄存器) 使能 WDT, WDT 触发复位后是否产生初始化配置过程则由 WDTBTE (初始化配置寄存器) 决定。

WDT 时钟源由 WCKSRC 选择 (如果选 LIRC 则由 LFMOD 进一步选定频率), 后分频器由 WDTPS, PSA 和 PS 一起设置。当 WDT 使能时, 其所选时钟源自动开启, 且在 SLEEP 模式下保持运行。

如需阻止 WDT 溢出, 必须在设定的定时时间之前清除 WDT, 具体可参阅 表 7-2 中的清除 WDT 事件。WDT 被清除后将重新开始计时。

7.1.3. 在 Timer0 和 WDT 之间切换分频电路

共用的硬件分频电路可分配给 Timer0 或 WDT 使用, 当在 Timer0 和 WDT 之间切换分频电路时可能会导致系统误复位。

将分频电路从分配给 Timer0 切换至 WDT 时, 必须遵循以下指令顺序:

```

BANKSEL TMR0                ; Can skip if already in TMR0 bank
CLRWDT                       ; Clear WDT
CLRR TMR0                    ; Clear TMR0 and scaler
BANKSEL OPTION
BSR OPTION, PSA              ; Select WDT

LDWI b'11111000'            ; Mask scaler bits (PS2-0)
ANDWR OPTION, W
IORWI b'00000101'           ; Set WDT scaler bits to 32 (or any value desired)
STR OPTION
    
```

将分频电路从分配给 WDT 切换至 Timer0 时, 必须遵循以下指令顺序:

```

CLRWDT                       ; Clear WDT and scaler
BANKSEL OPTION
LDWI b'11111000'            ; Mask TMR0 select and scaler bits (PSA, PS2-0)
ANDWR OPTION, W
IORWI b'00000011'           ; Set Timer0 scale to 16 (or any value desired)
STR OPTION
    
```

7.2. 定时器 0 (TIMER0) 和 PWM0

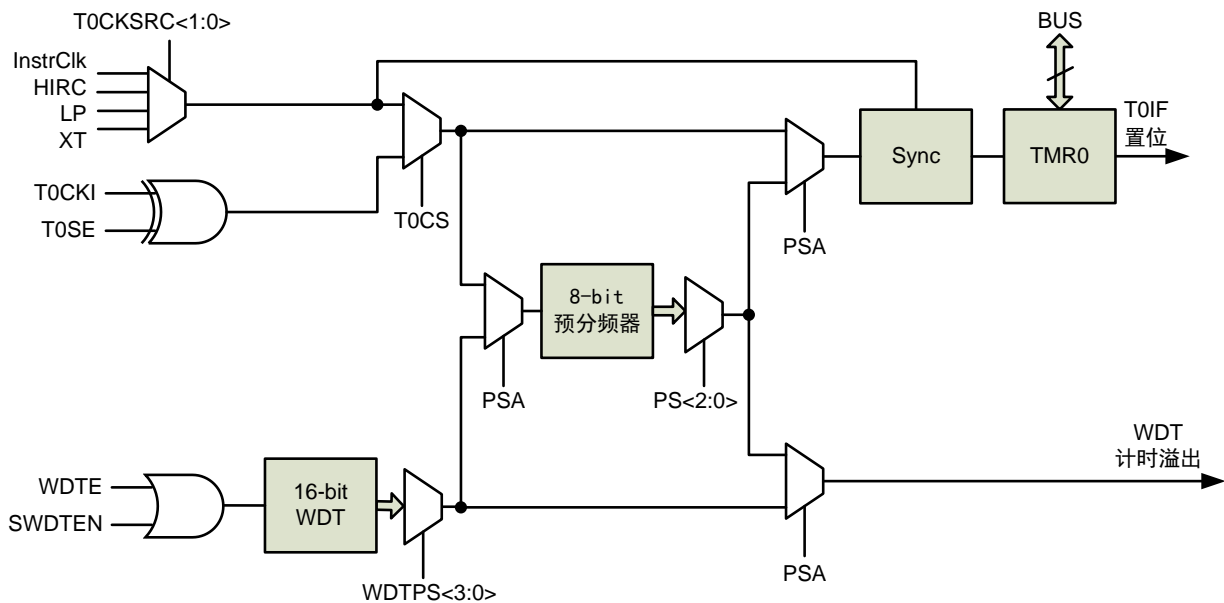


图 7-2 Timer0 结构框图

Timer0 可用作 I/O “PC5-T0CKI” 的上升沿/下降沿计数器, 或计时的定时器 (参阅 T0CKSRC), 也可用

于产生 PWM0。

Timer0 计数和定时溢出时间 = $TMR0[7:0] * Timer0_预分频$

Timer0 溢出将置位中断标志位(TOIF), 是否触发中断和/或从睡眠中唤醒则取决于相应的使能控制位(TOIE 和 GIE)。

注:

1. 对 TMR0 进行写操作后的 2 个指令周期内, Timer0 停止递增;
2. 如需从睡眠中唤醒, 需设置 “T0CKRUN = 1” 和 “T0CKSRC ≠ 00”, 以使 Timer0 的时钟源不是指令时钟且在 SLEEP 模式下保持运行, 或设置 ”SYSON = 1” 和 “T0CKRUN = 1”, 以使指令时钟在 SLEEP 模式下保持运行, 否则 Timer0 将停止计数, 维持其进入睡眠前的计数值;
3. 如果 Timer0 用于对 T0CKI 进行计数, 那么相对于 Timer0, 对 T0CKI 有最小周期、高/低脉冲宽度的要求。除非 T0CKI 非常快且 T_{T0CK} 非常慢, 否则通常都满足这些限制条件;

T0CKI	最小值	单位	条件
高/低脉冲宽度	$0.5 * T_{T0CK} + 20$	ns	无预分频
	10	ns	有预分频
周期	20 和 $(T_{T0CK}+40)/N$ 中的较大者	ns	N = 1, 2, 4, ..., 256 (有预分频) N = 1 (无预分频)

4. 关于 “在 Timer0 和 WDT 之间切换分频电路” 请参阅 [章节 7.1.3](#);

7.2.1. Timer0 相关寄存器汇总

名称	状态		寄存器	地址	复位值
GIE	全局中断	1 = 使能 (TOIE 适用) 0 = 全局关闭 (唤醒不受影响)	INTCON[7]	Bank 首地址 +0x0B	RW-0
TOIE	Timer0溢出 中断控制位	1 = 使能 0 = 关闭 (无唤醒)	INTCON[5]		RW-0
TOIF	Timer0溢出 中断标志位	1 = 已经溢出 (锁存) 0 = 未溢出	INTCON[2]		RW-0

表 7-5 Timer0 中断使能和状态位

名称	状态		寄存器	地址	复位值			
P0ANP	P0AN 极性	1 = 取反	MSCON2[7]	0x10D	RW-0			
P0AP	P0A 极性	0 = 正常	MSCON2[6]		RW-0			
T0DZ	PWM0 死区时间 (<verL 芯片)	00 = 4ns 10 = 2x T0CK 01 = 1x T0CK 11 = 4x T0CK	T0CON0[7:6]	0x105	RW-00			
	PWM0 死区时间 (≥verL 芯片)	00 = 0ns 10 = 2x T0CK 01 = 1x T0CK 11 = 4x T0CK						
P0ANOE	P0AN 通道输出到 PA4	1 = 使能	T0CON0[5]			RW-0		
P0AOE	P0A 通道输出到 PA5	0 = 关闭	T0CON0[4]			RW-0		
T0ON	Timer0 模块	1 = 使能 0 = 关闭	T0CON0[3]			RW-1		
T0CKRUN	SLEEP 模式下 T0CK 保持运行	1 = Yes (时钟源非指令时钟) 0 = No	T0CON0[2]			RW-0		
T0CKSRC	<u>Timer0 时钟源 (T0CS = 0)</u>		T0CON0[1:0]			0x81	RW-00	
	00 = 指令时钟 10 = LP (*) 01 = HIRC 11 = XT (*)							
(*) FOSC 应相应配置成 LP/XT 模式或选择 INTOSCIO 模式，否则振荡器将不会运行；								
SYSON	<u>Sleep 模式下, 系统时钟控制</u>	1 = 保持运行 0 = 关闭	LVDCON[7]					0x21A
T0CS	Timer0 输入源	1 = PC5/T0CKI (计数器) 0 = T0CKSRC (定时器)	OPTION[5]	0x81	RW-1			
T0SE	计数器触发沿	1 = 下降沿 0 = 上升沿	OPTION[4]		RW-1			
PSA	1 = 分频电路分配给 WDT 后分频器 0 = 分频电路分配给 Timer0 预分频器		OPTION[3]		RW-1			
PS	WDT 后分频比		Timer0 预分频比		OPTION[2:0]			RW-111
	000	(PSA=1)	1	2				
	001		2	4				
	010		4	8				
	011		8	16				
	100		16	32				
	101		32	64				
	110		64	128				
	111		128	256				
xxx	(PSA=0)	1	(PSA=1)	1				
TMR0	Timer0 计数值		TMR0[7:0]	0x01	RW-xxxx xxxx			
PR0	PR0 周期寄存器		PR0[7:0]	0x107	RW-1111 1111			
T0DUTY	PWM0 (P0A / P0AN) 占空比		T0DUTY[7:0]	0x108	RW-0000 0000			

表 7-6 Timer0 相关用户控制寄存器

7.2.2. PWM0

Timer0 具有 1 路极性可选、带有互补输出以及死区功能(参阅 "T0DZ")的 PWM0。PWM0 可作为 Touch 模块的驱动信号，或直接输出到 I/O (参阅 "P0AOE" 和 "P0ANOE")。

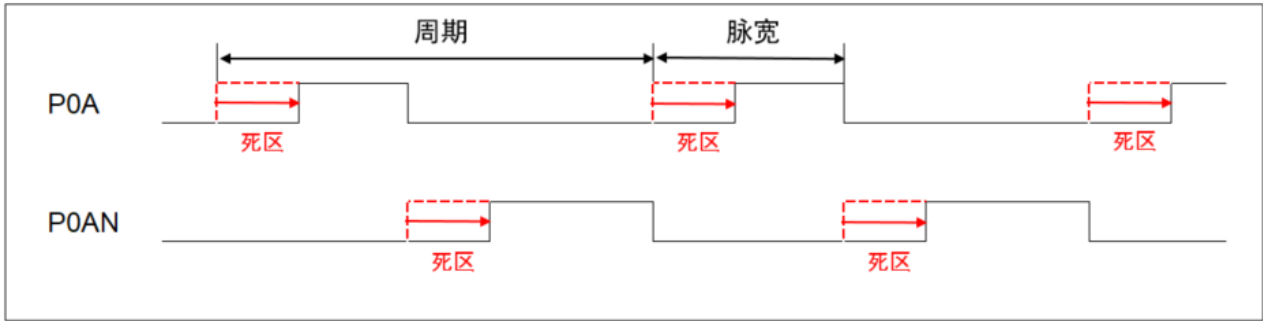


图 7-3 PWM0 时序图

PWM0 周期由 PR0 决定，如公式 7-1:

公式 7-1 $PWM0 \text{ 周期} = (PR0 + 1) * T_{T0CK} * (TMR0 \text{ 预分频值})$

当 Timer0 的计数结果寄存器 TMR0 与 PR0 相等时:

- TMR0 被清零，即 "TMR0 = 0"。
- P0A 输出逻辑 "1"。

PWM0 脉宽和占空比分别由公式 7-2 和公式 7-3 计算得出:

公式 7-2 $\text{脉宽} = T0DUTY * T_{T0CK} * (TMR2 \text{ 预分频值})$

公式 7-3 $\text{占空比} = T0DUTY \div (PR0 + 1)$

7.3. 定时器 1 (TIMER1)

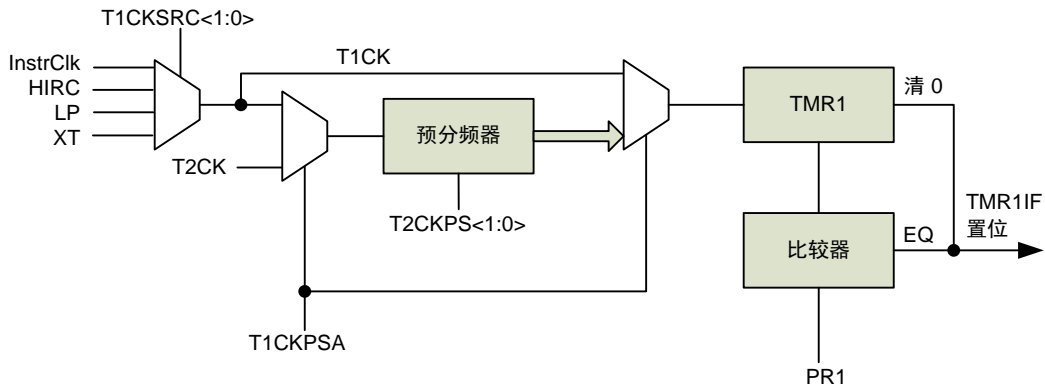


图 7-4 Timer1 结构框图

Timer1 用作计时功能。

共用的预分频器 (由 T2CKPS 设置预分频比值 1, 4 或 16) 可分配给 Timer1 或 Timer2 使用, 如果未分配给 Timer1, 其分频比为“1”。

Timer1 将从 0x00 开始递增, 直至递增到设置时间 =

$$\text{Timer1_预分频} * \text{PR1} / (\text{Timer1 时钟频率})$$

此时, TMR1 与 PR1 匹配, 并置位中断标志位(TMR1IF), 而 TMR1 在下一个递增周期复位为 0x00。匹配后是否触发中断和/或从睡眠中唤醒则取决于相应的使能控制位(GIE, PEIE 和 TMR1IE)。

如需从睡眠中唤醒, 需设置 “T1CKRUN = 1” 和 “T1CKSRC ≠ 00”, 以使 Timer1 的时钟源不是指令时钟且在 SLEEP 模式下保持运行, 或设置 ”SYSON = 1” 和 “T1CKRUN = 1”, 以使指令时钟在 SLEEP 模式下保持运行, 否则 Timer1 将停止计数, 维持其进入睡眠前的计数值。

7.3.1. Timer1 相关寄存器汇总

名称	状态		寄存器	地址	复位值
GIE	全局中断 1 = 使能 (PEIE, TMR1IE 适用) 0 = 全局关闭 (唤醒不受影响)		INTCON[7]	Bank 首地址	RW-0
PEIE	外设总中断	1 = 使能 (TMR1IE 适用) 0 = 关闭 (无唤醒)	INTCON[6]	+0x0B	RW-0
TMR1IE	Timer1与PR1匹配中断	1 = 使能 0 = 关闭 (无唤醒)	PIE1[3]	0x8C	RW-0
TMR1IF	Timer1与PR1匹配中断标志位	1 = 匹配 (锁存) 0 = 不匹配	PIR1[3]	0x0C	RW-0

表 7-7 Timer1 中断使能和状态位

名称	状态		寄存器	地址	复位值
T1AUSTP	Timer1 自动关闭	1 = 使能 0 = 禁止	T1CON0[6]	0x106	RW-0
T1ECKE	Timer1 输入源	1 = PA2/T1CKI (计数器) 0 = T1CKSRC (定时器)	T1CON0[5]		RW-0
T1CKPSA	预分频器分配位	1 = Timer1 0 = Timer2	T1CON0[4]		RW-0
T1CKRUN	睡眠时 T1CK 保持运行	1 = Yes (时钟源非指令时钟) 0 = No	T1CON0[3]		RW-0
T1ON	Timer1	1 = 使能 0 = 关闭	T1CON0[2]		RW-0
T1CKSRC	Timer1 时钟源 (T1ECKE = 0) 00 = 指令时钟 10 = LP ^(*) 01 = HIRC 11 = XT ^(*) ^(*) FOSC 应相应配置成 LP/XT 或选择 INTOSCIO		T1CON0[1:0]		RW-00

名称	状态	寄存器	地址	复位值
	模式，否则振荡器将不会运行			
SYSON	Sleep 模式下，系统时钟控制 1 = 保持运行 0 = 关闭	LVDCON[7]	0x21A	RW-0
T2CKPS (T1CKPS)	Timer2 / Timer1 预分频比 00 = 1 01 = 4 1x = 16	T2CON0[1:0]	0x1E	RW-00
PR1L	PR1 周期寄存器低 8 位	PR1L[7:0]	0x117	RW-1111 1111
PR1H	PR1 周期寄存器高 4 位	PR1H[3:0]	0x118	RW-1111
TMR1L	TMR1 计数结果寄存器低 8 位	TMR1L[7:0]	0x119	RW-0000 0000
TMR1H	TMR1 计数结果寄存器高 4 位	TMR1H[3:0]	0x11A	RW-0000
T1EVTS	当 T1AUSTP = 1 时，Timer1 自动关闭事件源 0xx = LVD 中断事件 100 = CM0 中断事件 101 = CM1 中断事件 110 = LVD 或 CM0 中断事件 111 = LVD 或 CM1 中断事件	MSCON2[2:0]	0x10D	RW-000

表 7-8 Timer1 相关用户控制寄存器

7.3.2. Timer1 寄存器的读/写操作

具体读写操作步骤请参照 [章节 7.4.2](#) “Timer2 寄存器的读/写操作”。

7.3.3. Timer1 自动关闭

当 T1AUSTP = 1 时，以下事件（参阅 T1EVTS）的到来可触发 Timer1 的自动关闭（T1ON = 0）：

- LVD 中断事件
- CM0 中断事件
- CM1 中断事件
- LVD 或 CM0 中断事件
- LVD 或 CM1 中断事件

注：

1. 无论 T1AUSTP 为何值，T1ON 均可由软件进行读写。
2. 当发生上述硬件事件并触发 Timer1 自动关闭后，即使相应的中断标志位一直置 1，T1ON 也可由软件重新置 1。

7.4. 定时器 2 (TIMER2)

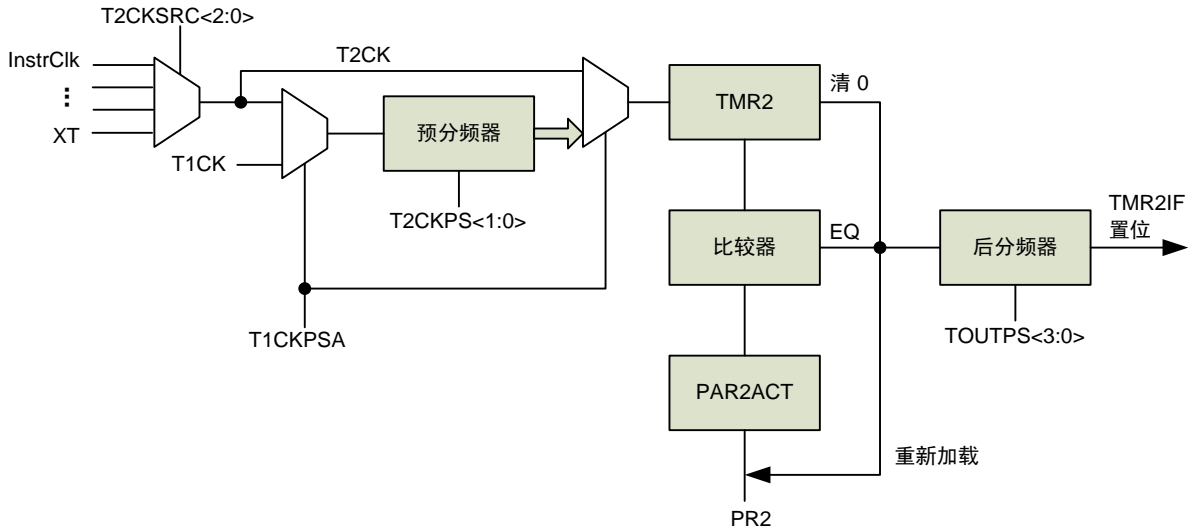


图 7-5 Timer2 结构框图

Timer2 为定时器，也可用于产生 PWM (无后分频器，参阅章节 10 PWM)，或用于 LIRC 和 HIRC 交叉校准计数 (CKCNTI=1)。可同时运用计数匹配和后分频器溢出功能。

Timer2 时钟源由 T2CKSRC 选择 (如果选 LIRC 则由 LFMOD 进一步选定频率)，Timer2 时钟被送入 Timer2 预分频器 (预分频比为 1, 4 或 16)，预分频器的输出用于递增 TMR2 寄存器，TMR2 从 0x00 开始递增直至与 PR2 匹配。匹配时：

1. TMR2 在下一个递增周期复位为 0x00；
2. Timer2 后分频器递增；
3. 当 Timer2 后分频器的递增输出值与后分频比设置值 (1, 2 15 或 16) 相等时，Timer2 溢出；
4. 中断标志位 TMR2IF 置 1，是否触发中断和/或从睡眠中唤醒则取决于相应的使能控制位(GIE, PEIE 和 TMR2IE)；

注：

1. 对 T2CON0 进行写操作并不会清零 TMR2 寄存器。
2. TMR2 和 PR2 都是可读/写寄存器。复位时，其值分别为 0x0000 和 0xFFFF。
3. 如需从睡眠中唤醒，需设置 “T2CKRUN = 1” 和 “T2CKSRC ≠ 000”，以使 Timer2 的时钟源不是指令时钟且在 SLEEP 模式下保持运行，或设置 ”SYSON = 1” 和 “T2CKRUN = 1”，以使指令时钟在 SLEEP 模式下保持运行，否则 Timer2 将停止计数，维持其进入睡眠前的计数值；

7.4.1. Timer2 相关寄存器汇总

名称	状态		寄存器	地址	复位值
T2CKRUN	睡眠时 T2CK 保持运行	1 = Yes (时钟源非指令时钟) 0 = <u>No</u>	MSCON0[0]	0x10C	RW-0
T1CKPSA	预分频器分配位	1 = Timer1 0 = <u>Timer2</u>	T1CON0[4]	0x106	RW-0
PR2U	<u>更新周期和占空比的即时生效控制位</u> 1 = PR2/P1xDTy 缓冲值立即分别更新到 PR2ACT 和 P1xDTyACT 0 = 周期结束后正常更新		T2CON0[7]	0x1E	RW1-0
TOUTPS	<u>Timer2 后分频比</u> 0000 = <u>1</u> 0100 = 5 1000 = 9 1100 = 13 0001 = 2 0101 = 6 1001 = 10 1101 = 14 0010 = 3 0110 = 7 1010 = 11 1110 = 15 0011 = 4 0111 = 8 1011 = 12 1111 = 16		T2CON0[6:3]		RW-0000
TMR2ON	<u>Timer2 (PWM 单脉冲模式下自动清 0)</u>	1 = 使能 0 = <u>关闭</u>	T2CON0[2]		RW-0
T2CKPS (T1CKPS)	<u>Timer2/Timer1 预分频比</u> 00 = <u>1</u> 01 = 4 1x = 16		T2CON0[1:0]		RW-00
T2CKSRC	<u>Timer2 时钟源</u> 000 = <u>指令时钟</u> 100 = HIRC 001 = 2x 指令时钟 101 = LIRC 010 = 2x HIRC 110 = LP (*) 011 = 2x LP, XT or EC (*) 111 = XT (*) (*) 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		T2CON1[2:0]		0x1F
SYSON	<u>Sleep 模式下, 系统时钟控制</u>	1 = 保持运行 0 = <u>关闭</u>	LVDCON[7]	0x21A	RW-0
LFMOD	1: LIRC = 256kHz 0: <u>LIRC = 32kHz</u>		OSCCON[7]	0x8F	RW-0
TMR2L	TMR2 计数结果寄存器低 8 位		TMR2L[7:0]	0x15	RW-0000 0000
TMR2H	TMR2 计数结果寄存器高 8 位		TMR2H[7:0]	0x16	RW-0000 0000
PR2L	PR2 周期寄存器低 8 位		PR2L[7:0]	0x17	RW-1111 1111
PR2H	PR2 周期寄存器高 8 位		PR2H[7:0]	0x18	RW-1111 1111

表 7-9 Timer2 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, TMR2IE 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 (TMR2IE 适用) 0 = 关闭 (无唤醒)	INTCON[6]		RW-0
TMR2IE	Timer2与PR2匹配中 断	1 = 使能 0 = 关闭 (无唤醒)	PIE1[1]	0x8C	RW-0
TMR2IF	Timer2与PR2匹配中 断标志位	1 = 匹配 (锁存) 0 = 不匹配	PIR1[1]	0x0C	RW-0

表 7-10 Timer2 中断使能和状态位

7.4.2. Timer2 寄存器的读/写操作

TMR2H 和 TMR2L 不能同时读或写。通过 TMR2H 的内部缓存 TMR2H_buf 可解决此问题，必须遵循以下读写顺序：

- 读 TMR2 时，先读 TMR2L，此时 TMR2H 的值将被锁存到 TMR2H_buf，然后读 TMR2H。当 Timer2 的时钟源不是指令时钟时，需设置 “TMR2ON = 0” 以停止计数，然后在读 TMR2 之前执行 1 条 NOP 指令。
- 写 TMR2 时，先写 TMR2H，此时 TMR2H 的值将被储存在 TMR2H_buffer 中。然后写 TMR2L，此时 TMR2H 和 TMR2L 将同时更新到计数值中。另外，为了避免写入操作和计数之间的竞争，在写操作前，应设置 “TMR2ON = 0” 以停止计数。

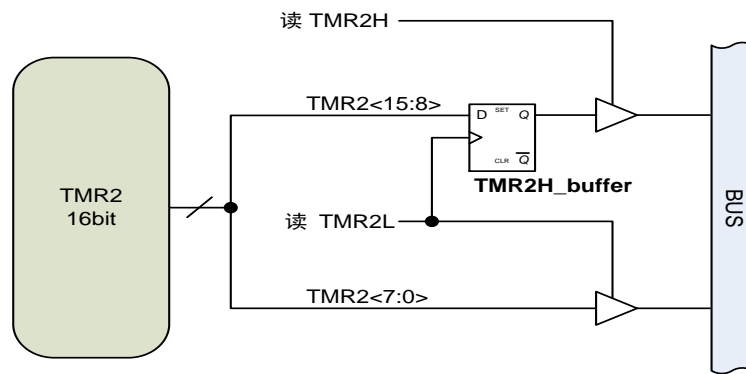


图 7-6 TMR2 读操作结构框图

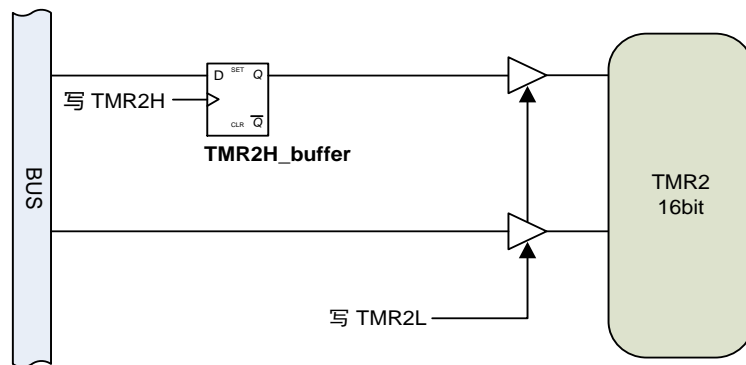


图 7-7 TMR2 写操作结构框图

8. SLEEP 睡眠模式 (POWER-DOWN)

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 8-1 所示，FT62F28x 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、ADC、运放、比较器、SPI、I2C 和 UART0/1 能在 SLEEP 模式下保持运行。有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 (配置 LVREN)	使能或通过指令控制 (SLVREN = 1)	非 SLEEP 模式下使能
LVD	LV DEN = 1	No
WDT	WDTE or SWDTEN	No
TIMER0	SYSON = 1 & T0CKSRC = 00 & T0CKRUN = 1 & T0ON = 1	SYSON = 0
	SYSON = 0 & T0CKSRC ≠ 00 & T0CKRUN = 1 & T0ON = 1	T0CKRUN=0
TIMER1	SYSON = 1 & T1CKSRC = 00 & T1CKRUN = 1 & T1ON = 1	SYSON = 0
	SYSON = 0 & T1CKSRC ≠ 00 & T1CKRUN = 1 & T1ON = 1	T1CKRUN=0
TIMER2	SYSON = 1 & T2CKSRC = 000 & T2CKRUN = 1 & TMR2ON = 1	SYSON = 0
	SYSON = 0 & T2CKSRC ≠ 000 & T2CKRUN = 1 & TMR2ON = 1	T2CKRUN=0
时钟输出	SYSON = 1 & SCKEN = 1	SYSON = 0
ADC	SYSON = 1 & ADON = 1	Yes: SYSON = 0 & ADCS ≠ 111 No: ADCS = 111
运算放大器	OP0ON = 1	No
比较器 0 / 1	CMxEN = 1	No
SPI	SYSON = 1 & SPIEN = 1	SYSON = 0
I2C	SYSON = 1 & I2CEN = 1	SYSON = 0
UART0 / UART1	SYSON = 1 & URxEN = 1 & URxTXEN / URxRXEN = 1	SYSON = 0
PWM	(跟随 Timer0 或 Timer2)	
HIRC / LIRC / EC / LP / XT	(跟随使用它们的外设状态)	
I/O	(除非 SLEEP 时使能 PWM、时钟输出或其他外设，否则 I/O 将保持其进入 SLEEP 前的状态)	

表 8-1 除指令时钟外，其他模块可根据需求在 SLEEP 模式下保持运行

8.1. 进入 SLEEP

CPU 通过执行 SLEEP 指令进入睡眠模式。进入睡眠时：

1. 如果 WDT 使能，则 WDT 的后分频器(如果分配给 WDT)和定时器将被清零，并重新开始计时。
2. 超时标志位 (/TF) = 1。
3. 掉电标志位 (/PF) = 0。
4. 时钟源
 - 当 SYSON = 0 时，指令时钟自动关闭；

- 当 $SYSON = 1$ 时, 指令时钟保持运行, 如果配置了输出内部指令时钟, 则进入睡眠后其输出也将继续;
- 如果 Timer 在 SLEEP 下保持运行, 则其所选用的时钟源 HIRC, LIRC 或外部振荡器(EC, LP, XT) 也将保持运行。如果某个 Timer 在睡眠中自动关闭, 则其时钟源也将自动关闭, 除非此时钟源同时被另一个保持运行的 Timer 所使用。

5. I/O 端口

- 如果 Timers 在 SLEEP 下保持运行, 则 PWM0/PWM1 输出也将保持。如果 Timers 自动关闭, 那么 PWM0/PWM1 的输出将保持其进入 SLEEP 前的状态。
- 当 $SYSON = 1$, 则 ADC、SPI、I2C 或 UART0/1 可保持运行。当 $SYSON = 0$ 时, ADC (ADCS $\neq x11$)、SPI、I2C 或 UART1/2 将自动关闭, 但 ADC 时钟源选择 LIRC (ADCS = x11) 除外。
- 其他数字输出端口, 将保持其进入 SLEEP 之前的状态(高阻态, 低电平或高电平)。

有关各个外设 在 SLEEP 状态下如何工作的详细信息, 请查阅相应章节。

注: 如果在执行 SLEEP 指令之前发生中断(中断标志位置 1 且其中断使能, 但全局中断 GIE 禁止), 则 SLEEP 指令将作为 NOP 执行, 不会进入睡眠模式。

8.2. 从 SLEEP 中唤醒

从睡眠中唤醒有 2 个基本原则:

- 基于时间, 即 CPU 在一定的时间后醒来。建议选择 LIRC 作为计时的时钟源, 因为 LIRC 比 HIRC 的功耗更低。
- 基于事件, 即触发 POR, 系统复位, 仅唤醒而不产生中断, 以及产生中断的事件, 如 LVD, ADC, 端口变化中断, 外部 INT 边沿中断, EEPROM 写完成。

ADC 有自动阈值比较功能, 当满足预设的阈值标准时, 则会触发唤醒和/或中断。

从睡眠中唤醒, 分为以下几种情况:

1. 如果使能, 看门狗定时器可触发唤醒 (参阅 [章节 7.1](#) 看门狗定时器)。
2. 完全复位和系统复位:
 - POR 完全复位 (不能关闭)
 - 通过 /MCLR 进行外部系统复位 (如果使能)
 - LVR 复位 (如果使能)
3. 使能中断 (关闭“全局中断使能”不影响唤醒功能)。请参阅 [章节 9](#) 中断。

注:

1. 从睡眠中唤醒将清零 WDT。
2. 紧跟 SLEEP 指令后必须写为 NOP 指令

使用非中断方式 (即未执行“中断服务程序) 从 SLEEP 中唤醒时, 比如 WDT 唤醒或全局中断控制位 (GIE)未使能时的中断事件唤醒, 下一条指令将被执行两次。

使用中断方式从 SLEEP 中唤醒时, 将先执行下一条指令, 然后再调用中断服务程序。

为了避免重复执行, 建议紧跟 SLEEP 的后面写为 NOP 指令。

SLEEP

NOP // 非中断方式唤醒时, NOP 将执行两遍。

9. 中断 (INTERRUPTS)

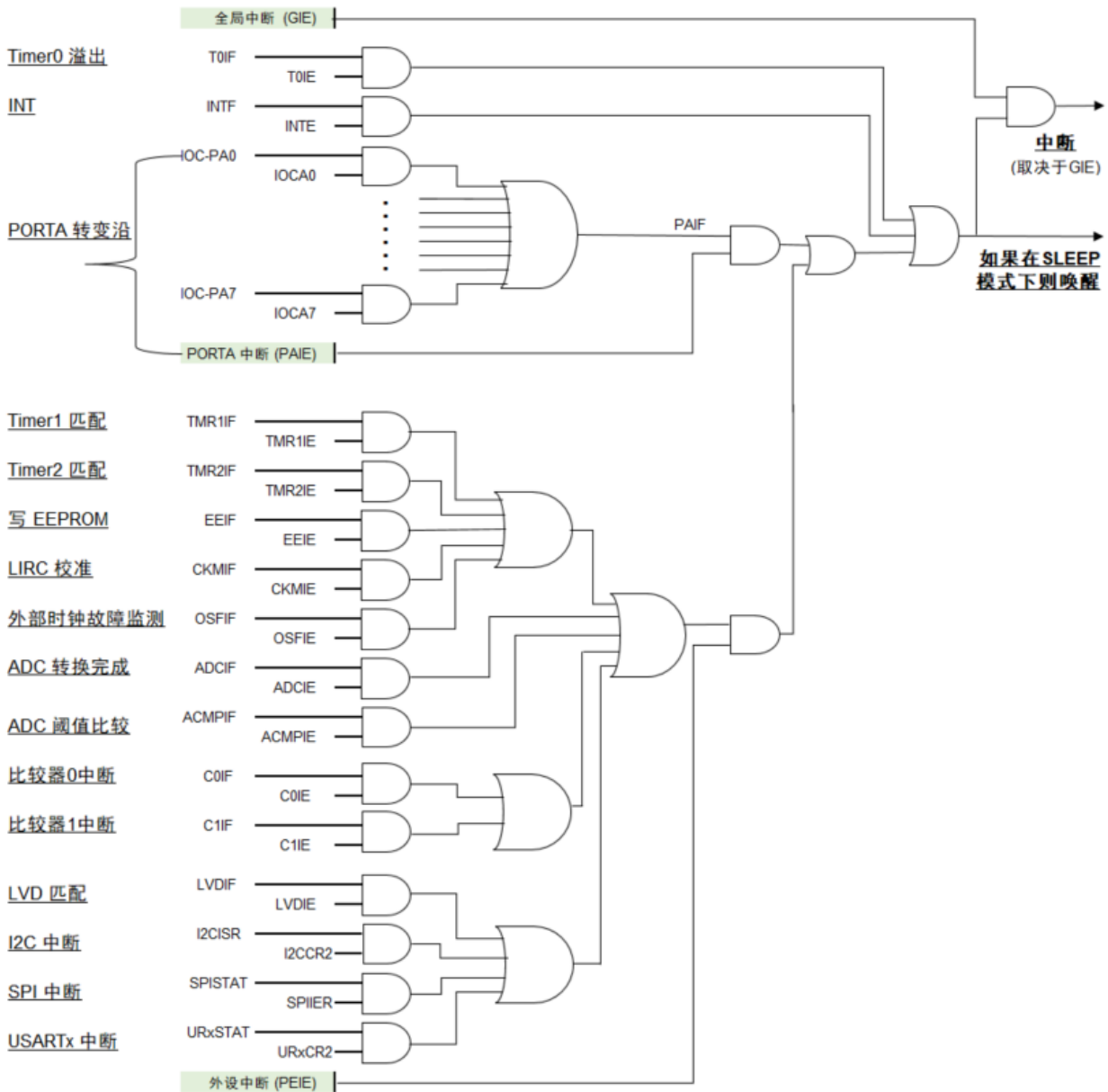


图 9-1 中断结构框图

CPU 支持 17 个中断源，分为 2 组：

1) 非外设中断 (Timer0 和 I/O)

- Timer0 溢出
- INT (上升沿或下降沿中断)
- PORTA 端口变化中断 (软件控制)

2) 外设中断

- Timer1 与 PR1 匹配
- Timer2 后分频器溢出
- DATA EEPROM 写完成
- LIRC 和 HIRC 交叉校准完成
- 故障保护时钟监控器
- 比较器 0/1 中断

- ADC 转换完成
- LVD 条件匹配
- SPI 中断
- ADC 阈值比较匹配
- I2C 中断
- UART0/1 中断

与其他 Timers 不同，WDT 溢出不会产生中断。除外部 I/O 中断外，其他中断请参阅相应章节。

产生中断时，PC 跳转并执行“中断服务程序(ISR)”。中断的关闭/使能有多层控制：

- 每个中断源均有其独立的中断使能位：TOIE, INTE, IOCAx, TMR1IE, TMR2IE, OSFIE, CKMIE, EEIE, C0IE, C1IE, ADCIE, ACMPIE, LVDIE, I2CCR2, SPIIER, UR0CR2, UR1CR2。
- 8 个 PAx 中断输入共用一个端口中断使能位：PAIE (PORTA Interrupt Enable)。
- 外设中断有一个总中断使能位：PEIE (PEripheral Interrupt Enable)。
- 如果关闭以上所有控制位，将不会执行从睡眠中唤醒。
- 所有中断都由全局中断使能位控制：GIE (Global Interrupt Enable)。与其他使能位不同，当关闭全局中断使能位时，依然允许从睡眠中唤醒。
- 关闭中断使能位并不影响中断标志位的置位。

中断处理时序如下：

- 自动设置“GIE = 0”，从而关闭中断。
- 返回地址被压入堆栈，程序指针 PC 加载 0x0004 地址。
- 发生中断后的 1 – 2 个指令周期，跳转到“中断服务程序(ISR)”开始处理中断。
- 执行“从中断返回(RETI)”指令退出 ISR。在 RETI 之前必须清除当前所处理的中断标志位。
- 当 ISR 完成时，PC 返回到中断前的地址，如果在 SLEEP 模式下，则返回到 SLEEP 指令后紧跟的地址。
- 在执行 RETI 时自动设置“GIE = 1”，从而使能中断。

注： 中断过程中，只有返回 PC 地址被自动保存在堆栈上。如果用户需要保存其他重要的寄存器值(如 W, STATUS 寄存器等)，必须通过指令将这些值正确地写入临时寄存器中，建议使用 SRAM 的最后 16 个 bytes 作为临时寄存器，因为所有 bank 共用这 16 个 bytes，而不需要切换 bank 以节省代码。

如果中断服务程序中需要修改 bank 区选择寄存器 AUXPGE，则要对 AUXPGE 进行现场保护，退出中断前再恢复，示例代码如下：

```
void interrupt ISR()
{
    AUXPGE_SHADOW = AUXPGE;           // 处理中断前，保存 AUXPGE
    ...                               // 其它代码
    AUXPGE = AUXPGE_SHADOW;         // 退出中断前，恢复 AUXPGE
}
```


9.1. 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(RW)
INTCON	0x0B	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000
PIE1	0x8C	EEIE	CKMIE	LVDIE	ACMPIE	TMR1IE	OSFIE	TMR2IE	ADCIE	0000 0000
PIR1	0x0C	EEIF	CKMIF	LVDIF	ACMPPIF	TMR1IF	OSFIF	TMR2IF	ADCIF	0000 0000
PIE2	0x89	-	-	-	-	-	-	C1IE	C0IE	---- --00
PIR2	0x09	-	-	-	-	-	-	C1IF	C0IF	---- --00
TRISA	0x85	PORTA 方向控制								1111 1111
IOCA	0x98	PORTA 端口变化中断设置								0000 0000
OPTION	0x81	/PAPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111
COMAF1	0x286	INTPO[1:0]		TX0PO[2:0]			RX0PO[2:0]			0000 0000
COMAF2	0x287	UR1SW	UR0SW	-	TX1PO[1:0]		INTFIXB	RX1PO[1:0]		00-0 0100
UR0CR2 ¹	0x188	UR0BRRH[3:0]				-	UR0ERRIE	UR0RXNEIE	UR0TXEIE	xxx- xxxx
UR1CR2 ¹	0x93	UR1BRRH[3:0]				-	UR1ERRIE	UR1RXNEIE	UR1TXEIE	xxx- xxxx
UR0STAT ¹	0x19F	-	UR0TXBSY	UR0OVF	UR0PEF	UR0RXFULL	UR0FEF	UR0RXNEF	UR0TXEF	-xxx xxxx
UR1STAT ¹	0x95	-	UR1TXBSY	UR1OVF	UR1PEF	UR1RXFULL	UR1FEF	UR1RXNEF	UR1TXEF	-xxx xxxx
I2CCR2 ¹	0x197	I2CADDRH[1:0]		NACK	DUTY	-	-	-	I2CIE	xxxx x--x
I2CISR	0x19E	HOLDF	STOPF	ADDRF	NACKF	I2CBUSY	DIRF / BUSERR	IICRXNEF	IICTXEF	0000 0001
SPIIER ¹	0x193	-	-	-	-	WAKUP	RXERR	RXNE	TXE	---- xxxx
SPISTAT	0x195	SPIF	WCOL	MODF	RXOVRN	-	-	WKF	-	0000 --0-
SPICFG ¹	0x18E	SPIBUSY	MSTEN	CPHA	CPOL	SLAS	NSSVAL	SRMT	RXBMT	xxxx xxxx
SPICR0	0x18D	-	-	-	-	NSSM		TXBMT	SPIEN	---- 1110

表 9-1 中断相关寄存器地址和默认值

¹ 寄存器复位值不确定，应用程序在系统初始化时应将其写为合适的值，特别是各中断使能位，以免产生中断及不必要的功耗。

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
T0IE	Timer0 溢出中断	1 = 使能 0 = 关闭 (无唤醒)	INTCON[5]		RW-0
INTE	INT 外部中断		INTCON[4]		RW-0
PAIE	PORTA 端口变化总中断		INTCON[3]		RW-0
T0IF	Timer0 溢出中断标志位	1 = Yes (锁存) 0 = No	INTCON[2]		RW-0
INTF	INT 外部中断标志位		INTCON[1]		RW-0
PAIF	PORTA 端口变化总中断标志位		INTCON[0]		RW-0

表 9-2 INTCON 寄存器

名称	状态		寄存器	地址	复位值	
EEIE	EE 写完成中断	1 = 使能 0 = 关闭 (无唤醒)	PIE1[7]	0x8C	RW-0	
CKMIE	LIRC 和 HIRC 交叉校准完成中断		PIE1[6]		RW-0	
LVDIE	LVD 中断		PIE1[5]		RW-0	
ACMPIE	ADC 阈值比较匹配中断		PIE1[4]		RW-0	
TMR1IE	Timer1 与 PR1 匹配中断		PIE1[3]		RW-0	
OSFIE	外部振荡器故障中断		PIE1[2]		RW-0	
TMR2IE	Timer2 与 PR2 匹配中断		PIE1[1]		RW-0	
ADCIE	ADC 转换完成中断		PIE1[0]		RW-0	
C1IE	比较器 1 中断	1 = Yes (锁存) 0 = No	PIE2[1]	0x89	RW-0	
C0IE	比较器 0 中断		PIE2[0]		RW-0	
EEIF	EE 写完成标志位		1 = Yes (锁存) 0 = No	PIR1[7]	0x0C	RW-0
CKMIF	LIRC 和 HIRC 交叉校准完成标志位			PIR1[6]		RW-0
LVDIF	LVD 中断标志位			PIR1[5]		RW-0
ACMPPIF	ADC 阈值比较匹配标志位			PIR1[4]		RW-0
TMR1IF	Timer1 与 PR1 匹配标志位			PIR1[3]		RW-0
OSFIF	外部振荡器故障标志位			PIR1[2]		RW-0
TMR2IF	Timer2 与 PR2 匹配标志位	PIR1[1]		RW-0		
ADCIF	ADC 转换完成标志位	PIR1[0]		RW-0		
C1IF	比较器 1 中断标志位	0x09	PIR2[1]	RW-0		
C0IF	比较器 0 中断标志位		PIR2[0]	RW-0		

表 9-3 PIE1/2 和 PIR1/2 寄存器

名称	状态		寄存器	地址	复位值
WAKUP	从机唤醒中断	1 = 使能 0 = 关闭 (无唤醒)	SPIIER[3]	0x193	RW-x
RXERR	接收错误中断 (工作模式错误, 接收溢出)	1 = 使能 0 = 关闭 (无唤醒)	SPIIER[2]		RW-x
RXNE	接收 BUF 为非空中断	1 = 使能 0 = 关闭 (无唤醒)	SPIIER[1]		RW-x
TXE	发送 BUF 为空中断	1 = 使能 0 = 关闭 (无唤醒)	SPIIER[0]		RW-x
WKF ²	从机唤醒(接收到数据)标志位	1 = 已唤醒 (锁存) 0 = 未唤醒, 或已被清零	SPISTAT[1]	0x195	RW0-0
RXOVRN ¹	接收溢出标志位	1 = 溢出 (锁存) 0 = 正常	SPISTAT[4]		RW0-0
MODF ¹	工作模式错误标志位	1 = 错误 (锁存) (主机模式下, NSS 脚使能输入且为低电平, 导致模式错误) 0 = 正常	SPISTAT[5]		RW0-0
TXBMT	发送 BUF 状态位	1 = 空 0 = 非空	SPICR0[1]	0x18D	RO-1
RXBMT	接收 BUF 状态位	1 = 空 0 = 非空	SPICFG[0]	0x18E	RO-x

表 9-4 SPI 中断使能和状态位

¹ 写 0 清零, 写 1 无效

名称	状态		寄存器	地址	复位值
I2CIE	I2C 接口中断	1 = 使能 0 = 关闭 注: I2C 接口中断产生的条件 HOLDF = 1 STOPF = 1 ADDRf = 1 BUSERR = 1 IICRXNEF = 1 IICTXEF = 1	I2CCR2 [0]	0x197	RW-x
HOLDF	主机模式, SCL 拉伸状态	1 = SCL 正被主机拉伸 0 = SCL 未被主机拉伸	I2CISR[7]	0x19E	RO-0
	从机模式, SCL 拉伸状态 ²	1 = SCL 正被从机拉伸 0 = SCL 未被从机拉伸, 或已被清零			RW0-0
STOPF ²	主/从机模式, Stop 标志	1 = 检测到 (ACK 后置位) 0 = 未检测到	I2CISR[6]		RW0-0
ADDRF ³	主机发送地址 / 从机接收地址匹配标志	主机发送地址: 1 = 完成 (ACK 后置位) 0 = 未发送或不匹配 从机接收地址: 1 = 匹配 0 = 不匹配 注: NACK 后不会置位 ADDRf	I2CISR[5]		RW0-0
NACKF ²	应答 ACK 状态	1 = NACK 0 = ACK	I2CISR[4]	0x19E	RW0-0
I2CBUSY	主/从机状态	1 = Busy (繁忙) 0 = IDLE (空闲) 注: 从机模式, 地址匹配成功后即置位, 接收到 Start / Restart / Stop 后清零;	I2CISR[3]		RO-0
BUSERR / DIRF	主机模式, 总线仲裁失败 (BUSERR) 标志	1 = 产生仲裁失败 0 = 未产生仲裁失败	I2CISR[2]		RO-0

² 写 0 清零, 写 1 无效;

³ 10 位地址格式, 低 8 位地址和高 2 位地址匹配时均会置位; 写 0 清零, 写 1 无效;

名称	状态		寄存器	地址	复位值
	从机模式，数据传输方向 (DIRF) 标志 ²	1 = 发送 (从机接收地址字节的读写位为 1 时置位) 0 = 接收			RW0-0
IICRXNEF ²	RX-FIFO 状态	1 = 非空 (读 I2CDAT 或写 0 清零) 0 = 空	I2CISR[1]		RW0-0
IICTXEF ⁴	TX-FIFO 状态	1 = 空 (写 I2CDAT 或写 1 清零) 0 = 非空	I2CISR[0]		RW1-1

表 9-5 I2C 中断使能和状态位

名称	状态		寄存器	地址	复位值
UR0TXEIE	发送 BUF 为空中断	1 = 使能	UR0CR2[0]	0x188	RW-x
UR1TXEIE		0 = 关闭	UR1CR2[0]	0x93	RW-x
UR0TXEF	发送 BUF 状态	1 = 空	UR0STAT[0]	0x19F	RO-x
UR1TXEF		0 = 非空	UR1STAT[0]	0x95	RO-x
UR0RXNEIE	接收 BUF 为非空中断	1 = 使能	UR0CR2[1]	0x188	RW-x
UR1RXNEIE		0 = 关闭	UR1CR2[1]	0x93	RW-x
UR0RXNEF	接收 BUF 状态	1 = 非空	UR0STAT[1]	0x19F	RO-x
UR1RXNEF		0 = 空, 或已被清零	UR1STAT[1]	0x95	RO-x
UR0ERRIE	接收错误中断	1 = 使能 0 = 关闭 注: 接收错误中断产生条件 URxOVF = 1 URxPEF = 1 URxFEf = 1	UR0CR2[2]	0x188	RW-x
UR1ERRIE		UR1CR2[2]	0x93	RW-x	
UR0OVF ⁵	接收 BUF 溢出错误标志	1 = 溢出	UR0STAT[5]	0x19F	RW0-x
UR1OVF ⁵		0 = 正常, 或已被清零	UR1STAT[5]	0x95	RW0-x
UR0PEF ⁵	接收到奇偶校验错误标志	1 = 错误	UR0STAT[4]	0x19F	RW0-x
UR1PEF ⁵		0 = 正确, 或已被清零	UR1STAT[4]	0x95	RW0-x
UR0FEF ⁵	接收到帧错误标志	1 = 错误	UR0STAT[2]	0x19F	RW0-x
UR1FEF ⁵		0 = 正确, 或已被清零	UR1STAT[2]	0x95	RW0-x

表 9-6 UART1/2 中断使能和状态位

⁴ 只能写 1, 写 0 无效;

⁵ 写 0 清零, 写 1 无效;

名称	状态	寄存器	地址	复位值
/PAPU	<u>PORTA 上拉</u> 1 = 全局关闭 0 = 由 WPUA 控制	OPTION[7]	0x81	RW-1
INTEDG	<u>INT 中断沿</u> 1 = 上升沿 0 = 下降沿	OPTION[6]		RW-1
TRISA	<u>PORTA I/O 数字输出 (方向控制)</u> 1 = 输入 (关闭数字输出) 0 = 关闭上拉/下拉	TRISA[7:0]	0x85	RW-1111 1111
IOCA	<u>PORTA 端口变化中断</u> 1 = 使能 0 = 禁止	IOCA[7:0]	0x98	RW-0000 0000
INTPO	<u>INT / BKIN / ADC_ETR 管脚映射</u> 0x = <u>PD4</u> 10 = PD2 11 = PC4	COMAF1[7:6]	0x286	RW-00
INTFIXB	<u>INT 中断管脚映射 (≥ G 版芯片)</u> 1 = 由 INTPO 决定 0 = 映射在 PD4 (BKIN 和 ADC_ETR 复用映射仍由 INTPO 决定) 注: <G 版芯片, 此位为保留位, 读为 0	COMAF2[2]	0x287	RW-1

表 9-7 OPTION, TRISA, IOCA 和 COMAFx 寄存器

9.2. INT 和 PORTA 端口变化中断

名称	INT	PORTA 端口变化中断
通道数	PC4 / PD2 / PD4 (3个通道)	PA0 – PA7 (多达8个通道)
I/O 设置	TRISC[x] / TRISD[x] = 1; ANSELx[x] = 0	TRISA[x] = 1; ANSELx[x] = 0
其他设置	INTEDG, INTPO, INTFIXB, INTE, GIE, INTF	IOCA, PAIE, GIE, PAIF
触发	上升沿或下降沿 (二选一)	0 → 1 或 1 → 0
需要软件监控?	No	需要

表 9-8 PA4-INT 和 PORTA 端口变化中断之间的区别

INT 和 PORTA 端口变化中断均为外部 I/O 中断。如果设置正确，INT 将在后台运行而无需监督。PORTA 端口变化中断则需要持续的软件监控。对于 PORTA 端口变化中断：

1. 将输入寄存器值锁存到端口变化中断锁存器中(通过读 PORTA)。
2. 当输入电平变化时，输入寄存器值与锁存器值之间的差异将置位 PAIF。
3. 输入寄存器的锁存过程(即读 PORTA 的过程)将更新用于比较的参考电平，如果在 PAIF 置位后立即读 PORTA 即可清除端口变化中断的触发条件。当端口不匹配事件不再存在时，PAIF 可通过指令清除。

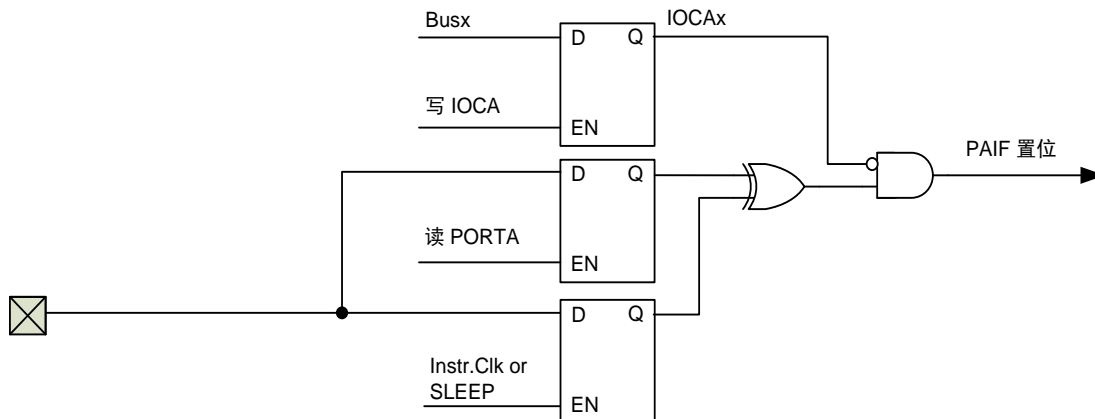


图 9-2 PORTA 转变沿中断

10. PWM

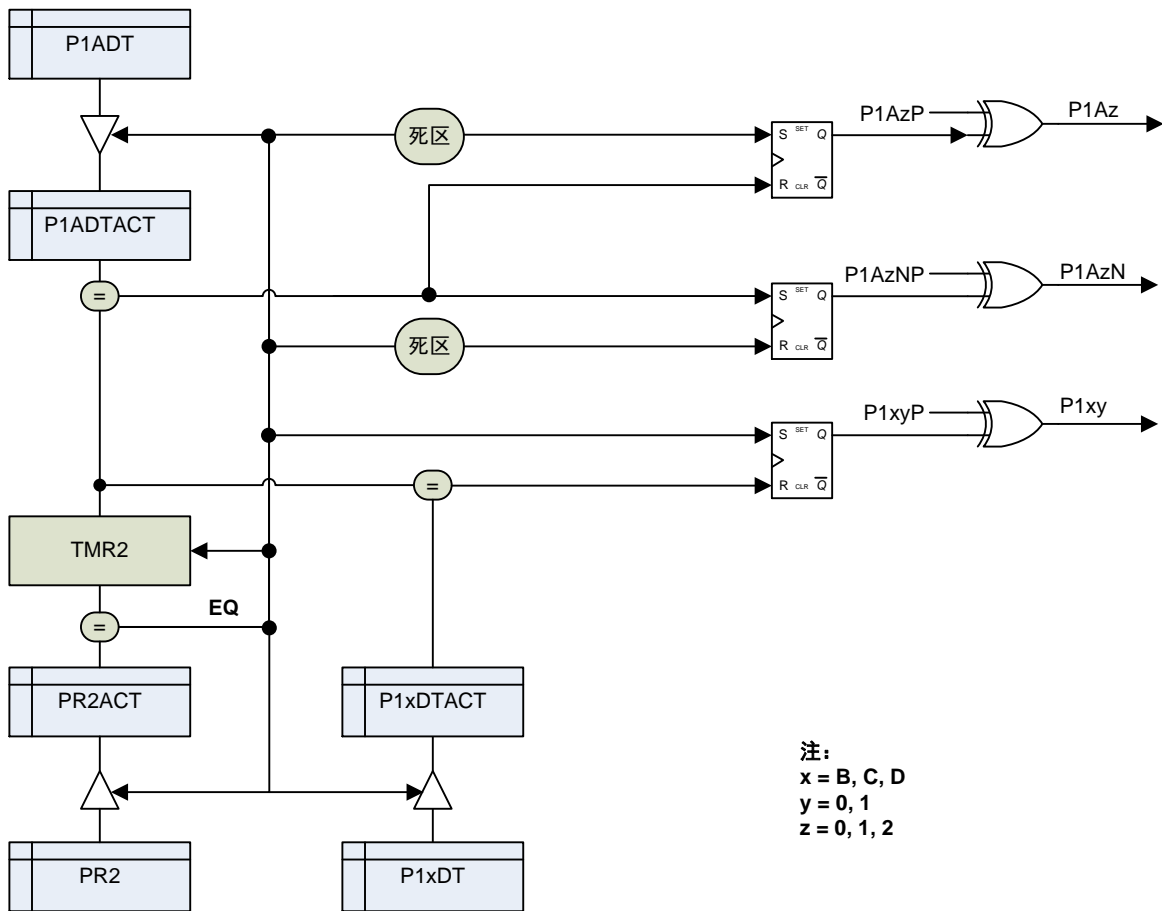


图 10-1 PWM 结构框图

PWM 特性：

- 4 路周期相同 (由 Timer2 控制)，且独立占空比的 PWM 通道：P1A, P1B, P1C, P1D
- 通道 1 带有互补输出：/P1A
- 1 路带死区控制的 PWM 通道：P1A, /P1A
- 16-bit 的分辨率
- 每路 PWM 输出极性可独立控制
- 带可选自动重启功能的多种故障刹车事件
- PWM1 和 PWM4 可映射到 3 个 I/O, PWM2 和 PWM3 可映射到 2 个 I/O
- XOR/XNOR 第 2 功能输出
- 蜂鸣器模式
- 单脉冲输出模式
- 周期和占空比寄存器双缓冲读写设计

SLEEP 模式下的 PWM 操作 – 不管是否处于 SLEEP 状态，只要 Timer2 保持运行(参阅 [章节 7.4](#) Timer2)，且 PWM 使能，那么 PWM 将一直保持运行。如果 SLEEP 模式下 Timer2 自动关闭，那么 PWM 的输出将保持其进入 SLEEP 前的状态。

10.1. PWM 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值	
P1ADTL	0x0D	P1A 占空比低 8 位								0000 0000	
P1ADTH	0x0E	P1A 占空比高 8 位								0000 0000	
P1BDTL	0x0F	P1B 占空比低 8 位								0000 0000	
P1BDTH	0x10	P1B 占空比高 8 位								0000 0000	
P1CDTL	0x11	P1C 占空比低 8 位								0000 0000	
P1CDTH	0x12	P1C 占空比高 8 位								0000 0000	
P1DDTL	0x13	P1D 占空比低 8 位								0000 0000	
P1DDTH	0x14	P1D 占空比高 8 位								0000 0000	
TMR2L	0x15	Timer2 计数器低 8 位								0000 0000	
TMR2H	0x16	Timer2 计数器高 8 位								0000 0000	
PR2L	0x17	PR2 周期低 8 位								1111 1111	
PR2H	0x18	PR2 周期高 8 位								1111 1111	
P1BR0	0x19	P1BEVT	P1BKS			P1BSS	P1ASS				0000 0000
P1BR1	0x1A	P1D2SS		P1DSS		P1C2SS		P1CSS		0000 0000	
P1POL0	0x1B	P1C0P	P1B0P	P1A2NP	P1A2P	P1A1NP	P1A1P	P1A0NP	P1A0P	0000 0000	
P1POL1	0x1C	P1D2P	P1D1P	P1D0P	—	—	P1C1P	P1B1P	—	000- -00-	
P1CON	0x1D	P1AUE	P1DC							0000 0000	
T2CON0	0x1E	PR2U	TOUTPS				TMR2ON	T2CKPS(T1CKPS)		0000 0000	
T2CON1	0x1F	—			P1OS	P1BZM	T2CKSRC			---0 0000	
P1AUX	0x8D	P1BKM	P1CDM	P1B2SS[1:0]		P1CF2E	P1CF2	P1DF2E	P1DF2	0000 0000	
P1OE1	0x9F	P1D2OE	P1D1OE	P1D0OE	—	—	P1C1OE	P1B1OE	—	000- -00-	
P1OE0	0x9E	P1C0OE	P1B0OE	P1A2NOE	P1A2OE	P1A1NOE	P1A1OE	P1A0NOE	P1A0OE	0000 0000	

表 10-1 PWM 相关用户寄存器汇总

	Timer2 周期		Timer2 计数器		占空比		死区时间
	MSB	LSB	MSB	LSB	MSB	LSB	
PWM1	PR2H	PR2L	TMR2H	TMR2L	P1ADTH	P1ADTL	P1DC
/PWM1							
PWM2					P1BDTH	P1BDTL	-
PWM3					P1CDTH	P1CDTL	-
PWM4					P1DDTH	P1DDTL	-

名称	地址	bit 7	bit 6 - 0	复位值(RW)
PR2H	0x18	PR2 周期高 8 位		1111 1111
PR2L	0x17	PR2 周期低 8 位		1111 1111
TMR2H	0x16	Timer2 计数器高 8 位		0000 0000
TMR2L	0x15	Timer2 计数器低 8 位		0000 0000
P1ADTH	0x0E	P1A 占空比高 8 位		0000 0000
P1ADTL	0x0D	P1A 占空比低 8 位		0000 0000
P1BDTH	0x10	P1B 占空比高 8 位		0000 0000
P1BDTL	0x0F	P1B 占空比低 8 位		0000 0000
P1CDTH	0x12	P1C 占空比高 8 位		0000 0000
P1CDTL	0x11	P1C 占空比低 8 位		0000 0000
P1DDTH	0x14	P1D 占空比高 8 位		0000 0000
P1DDTL	0x13	P1D 占空比低 8 位		0000 0000
P1CON	0x1D	-	P1DC (死区时间)	-000 0000

表 10-2 PWM 时序设置

	通道分配			输出使能			极性		
	Ch 0	Ch 1	Ch 2	Ch 0	Ch 1	Ch 2	Ch 0	Ch 1	Ch 2
PWM1	PA4	PB1	PC2	P1A0OE	P1A1OE	P1A2OE	P1A0P	P1A1P	P1A2P
/PWM1	PA3	PC5	PC1	P1A0NOE	P1A1NOE	P1A2NOE	P1A0NP	P1A1NP	P1A2NP
PWM2	PA5	PB6	-	P1B0OE	P1B1OE	-	P1B0P	P1B1P	-
PWM3	PA6	PC6	-	P1C0OE	P1C1OE	-	P1C0P	P1C1P	-
PWM4	PB0	PC7	PA7	P1D0OE	P1D1OE	P1D2OE	P1D0P	P1D1P	P1D2P

名称	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	地址	复位值(RW)
P1OE0	P1C0OE	P1B0OE	P1A2NOE	P1A2OE	P1A1NOE	P1A1OE	PA10NOE	P1A0OE	0x9E	0000 0000
P1OE1	P1D2OE	P1D1OE	P1D0OE	-	-	P1C1OE	P1B1OE	-	0x9F	000- -00-
P1POL0	P1C0P	P1B0P	P1A2NP	P1A2P	P1A1NP	P1A1P	P1A0NP	P1A0P	0x1B	0000 0000
P1POL1	P1D2P	P1D1P	P1D0P	-	-	P1C1P	P1B1P	-	0x1C	000- -00-

表 10-3 PWM 输出极性 (1 = 反向, 0 = 正常) 和输出使能位 (1 = 使能, 0 = 关闭)

	关闭 = 0, 使能 = 1			异或(XOR) = 0, 同或(XNOR) = 1			第 2 功能
	Ch 0	Ch 1	Ch 2	Ch 0	Ch 1	Ch 2	
PWM1	-	-	-	-	-	-	N/A
/PWM1	-	-	-	-	-	-	N/A
PWM2	-	-	-	-	-	-	N/A
PWM3	-	P1CF2E	-	-	P1CF2	-	P1C xor/xnor P1D
PWM4	-	P1DF2E		-	P1DF2		P1B xor/xnor P1C

名称	地址	bit 3	bit 2	bit 1	bit 0	复位值 (RW)
P1AUX	0x8D	P1CF2E	P1CF2	P1DF2E	P1DF2	0000

表 10-4 PWM 第 2 功能

名称	4 路 PWM 通道通用控制	寄存器	地址	复位值
PR2U	<u>更新周期和占空比的即时生效控制位</u> 1 = PR2/P1xDTy 缓冲值立即分别更新到 PR2ACT 和 P1xDTyACT 0 = <u>周期结束后正常更新</u>	T2CON0[7]	0x1E	RW1-0
P1OS	1 = 单脉冲 (One pulse) 模式 0 = <u>正常连续模式</u>	T2CON1[4]	0x1F	RW-0
P1BZM	1 = 蜂鸣器 (Buzzer) 模式, 50%占空比 0 = <u>正常 PWM 模式</u>	T2CON1[3]		RW-0

表 10-5 4 路 PWM 通道的通用功能控制

名称	状态	寄存器	地址	复位值
P1BKS	<u>PWM1 故障源</u> 000: <u>禁止故障刹车功能</u> 100: C1OUT=1 001: BKIN= 0 101: C0OUT=1 or C1OUT=1 010: BKIN=1 110: ADC 阈值比较标志位为'1' 011: C0OUT=1 111: BKIN= 0 or C0OUT=1 or C1OUT=1 or ADC 阈值比较标志位为'1'	P1BR0[6:4]	0x19	RW-00 0
	P1BEVT	<u>PWM1 发生故障事件标志位</u> 1 = Yes (锁存, 直至被清零) 0 = <u>No</u>		P1BR0[7]
P1AUE	<u>PWM1 自动重启</u> 1 = 当故障条件被清除时, P1BEVT 自动清零, PWM1 自动重启 0 = <u>当故障条件被清除时, P1BEVT 由指令清零, PWM1 重启</u>	P1CON[7]	0x1D	RW-0
P1BKM	<u>故障刹车时, Timer2 及其预分频</u> 1 = 不受影响 0 = <u>停止</u>	P1AUX[7]	0x8D	RW-0
P1CDM	<u>故障刹车时, P1C/P1D 输出控制</u> 1 = 保持输出 0 = <u>分别由 P1CSS/P1C2SS/P1DSS/P1D2SS 寄存器控制</u>	P1AUX[6]		RW-0

表 10-6 PWM 故障刹车相关寄存器

名称	地址	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	复位值(RW)
P1BR0	0x19	P1BEVT	P1BKS [2:0]		P1BSS		P1ASS		0000 0000	
P1BR1	0x1A	P1D2SS		P1DSS	P1C2SS		P1CSS		0000 0000	
P1AUX	0x8D	P1BKM	P1CDM	P1B2SS	P1CF2E	P1CF2	P1DF2E	P1DF2	0000 0000	

	故障下，引脚输出状态			备注		
	Ch0	Ch1	Ch2	< G 版芯片		≥ G 版芯片
PWM1	P1ASS			00 = 高阻	⁽¹⁾ 00 = 高阻	00 = 高阻
/PWM1				01 = 逻辑“0”	01 = 0	01 = 0
PWM2	P1BSS	P1B2SS ⁽¹⁾	-	1x = 逻辑“1”	1x = 1	1x = 1
PWM3	P1CSS	P1C2SS ⁽¹⁾	-	如 p1xxp = 0, 逻辑“0” = 0		
PWM4	P1DSS	P1D2SS ⁽¹⁾		如 p1xxp = 1, 逻辑“0” = 1		

表 10-7 PWM 发生故障时的输出状态

10.2. 时钟源

4 路 PWM 通道的专用定时器为 Timer2，其可选择的 6 个时钟源如下：

- 1x or 2x 指令时钟
- 1x or 2x HIRC
- LIRC
- 1x or 2x 外部时钟 (只有当 FOSC 相应地配置成 LP, XT 或 EC 模式时才有效)

具体时钟源设置请参阅 [章节 7.4 Timer2](#)。

10.3. 周期 (Period)

PWM 周期由 Timer2 的 PR2 (PR2H + PR2L) 周期寄存器决定，如 [公式 10-1](#)：

公式 10-1 $PWM \text{ 周期} = (PR2 + 1) * T_{T2CK} * (TMR2 \text{ 预分频值})$

当 Timer2 的计数结果寄存器 TMR2 与 PR2 相等时：

- Timer2 的周期和占空比寄存器 (PR2ACT 和 P1xDTACT) 被更新。
- TMR2 被清零，即 “TMR2 = 0”。
- P1Ax, P1Bx, P1Cx, P1Dx 输出逻辑 “1”。

10.4. 占空比 (Duty Cycle)

4 路 PWM 均具有独立的占空比，由相应的 2 x 8-bit 寄存器 (P1xDTH, P1xDTL) 设置。P1xDTH 为高 8 位而 P1xDTL 为低 8 位。由于内部的双缓冲设计，P1xDTH 和 P1xDTL 寄存器可在任何时刻被更新写入。

PWM 脉宽 (Pulse width) 和占空比 (Duty cycle) 分别由 [公式 10-2](#) 和 [公式 10-3](#) 计算得出：

公式 10-2 $脉宽 = P1xDT * T_{T2CK} * (TMR2 \text{ 预分频值})$

公式 10-3 $占空比 = P1xDT \div (PR2 + 1)$

10.5. 死区 (Deadband) 时间

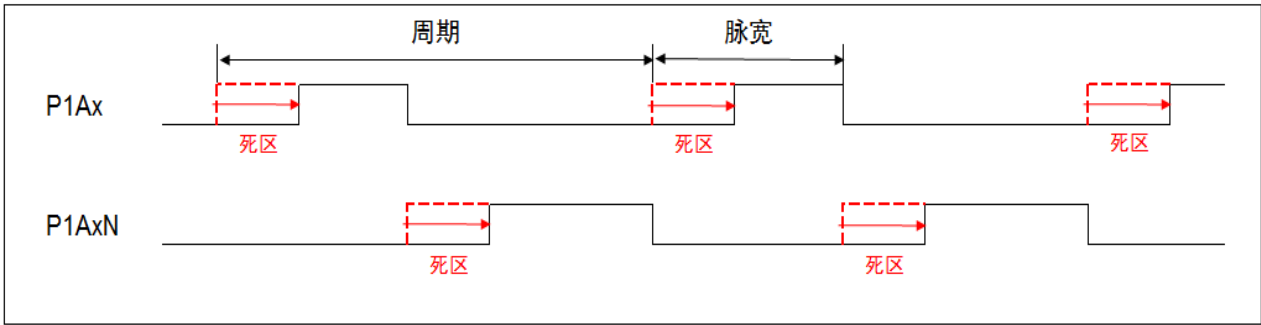


图 10-2 PWM 死区时间时序图

如果 P1DC ≠ “00 0000”，P1Ax 和 P1AxN (/P1A) 的低到高转换沿将产生延迟，延迟时间即为“死区”时间。有效脉宽和占空比也相应减小。死区定时器以 Timer2 时钟作为计数时钟源。

10.6. 故障刹车 (Fault-Break) 功能

4 路 PWM 均支持故障刹车功能。一旦发生故障刹车事件，且只要故障条件一直存在，PWM 输出引脚将根据其设置一直输出预设状态。TMR2ON 受影响。故障刹车事件可以为下列条件之一：

- BKIN = 0
- BKIN = 1
- C0OUT = 1
- C1OUT = 1
- C0OUT = 1 或 C1OUT = 1
- ADC 阈值比较标志位为‘1’
- BKIN=0 或 C0OUT=1 或 C1OUT=1 或 ADC 阈值比较标志位为‘1’

注： P1BEVT 为故障事件状态位。C0OUT/C1OUT 不锁存，反映比较器 0/1 的实时比较结果。每次 ADC 转换结束后都会更新阈值比较结果。

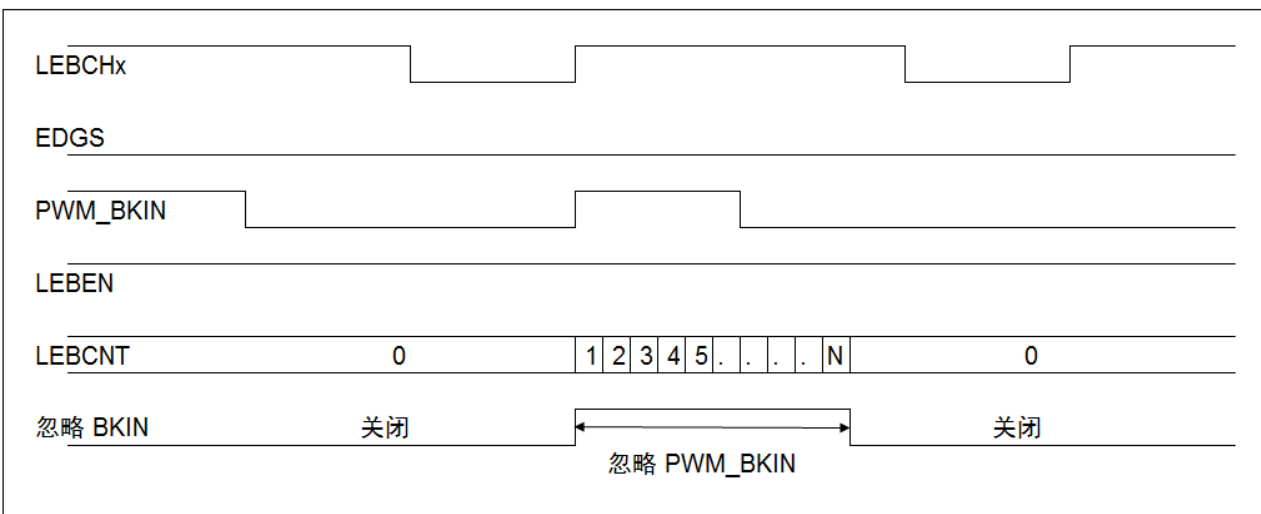


图 10-3 LEB 时序图

如果 “LEBEN = 1”，那么在 LEB 计数时间内将忽略 PWM 故障刹车事件的触发。这种 LEB 特性对 PWM

故障刹车触发(BKIN) 和 ADC 触发均适用，且参数设置也共用。

名称	状态		寄存器	地址	复位值
ETGTYP	<u>LEB 触发沿</u> (≥ VerG 芯片)	00 = <u>PWM 下降沿</u> 01 = PWM 上升沿 1x = 保留值	ADCON2[5:4]	0x115	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	<u>LEB 信号源</u> (≥ VerG 芯片)	000 = P1A0 100 = P1D 001 = P1A0N 101 = ADC_ETR 010 = P1B 110 = (无) 011 = P1C 111 = (无)	ADCON2[2:0]		RW-000
ADDLY / LEBPRL	<u>ADC 延迟或 LEB 低 8 位</u> (非软件触发, ADEX = 1) (延迟时间 = (ADDLY+6)/F _{ADC}) (< G 版芯片) 外部延迟时间 = (ADDLY+3) / F _{ADC} (≥ G 版芯片) (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]	0x110	RW-0000 0000
LEBEN	<u>ADC 触发和 BKIN 的 LEB 使能位</u> 1 = 使能 (当 GO/DONE=1 时进行切换将产生不可预知的结果) 0 = <u>关闭</u>		LEBCON[7]	0x10E	RW-0
LEBCH	<u>LEB 信号源</u> (< VerG 芯片)	00 = P1A0 10 = P1C 01 = P1B 11 = P1D	LEBCON[6:5]		RW-00
(≥ VerG 芯片为保留位)					
EDGS	<u>LEB 触发沿</u> (< VerG 芯片)	1 = PWM 下降沿 0 = <u>PWM 上升沿</u>	LEBCON[3]	0x10E	RW-0
	LEB 周期结束后, 触发 ADC 转换 (≥ VerG 芯片)	1 = 触发 0 = <u>不触发</u>			

表 10-8 LEB 用户寄存器

故障刹车时的输出状态 – 对于 < VerG 芯片, 故障刹车时, P1x 输出可以为输入状态(高阻), 输出逻辑高或逻辑低。注意, P1B1, P1C1, P1D1, P1D2 的故障输出状态的控制逻辑与其他 I/O 不同。对于 ≥ VerG 芯片, 故障刹车时, P1x 输出可以为输入状态(高阻), 输出高或低。

故障清除 – 只要故障条件有效, P1BEVT 便不能由指令清零。只有当故障条件被清除时, P1BEVT 才可被指令清零。

自动重启模式 – 发生故障刹车时, Timer2 将停止计数。当故障事件结束后, Timer2 将从其停止处恢复计数。4 路 PWM 输出可同时配置成自动重启模式, 否则 PWM 输出必须由指令重启。

10.7. 周期和占空比寄存器的更新

周期和各占空比寄存器可随时被更新写入，但除非使用 PR2U 来使其立即强制更新，否则直至下一个周期到来时其更新值才会真正有效。

注： 指令可读 PR2 和 P1xDTL, P1xDTH 寄存器，而 xxxACT 对软件不可见。

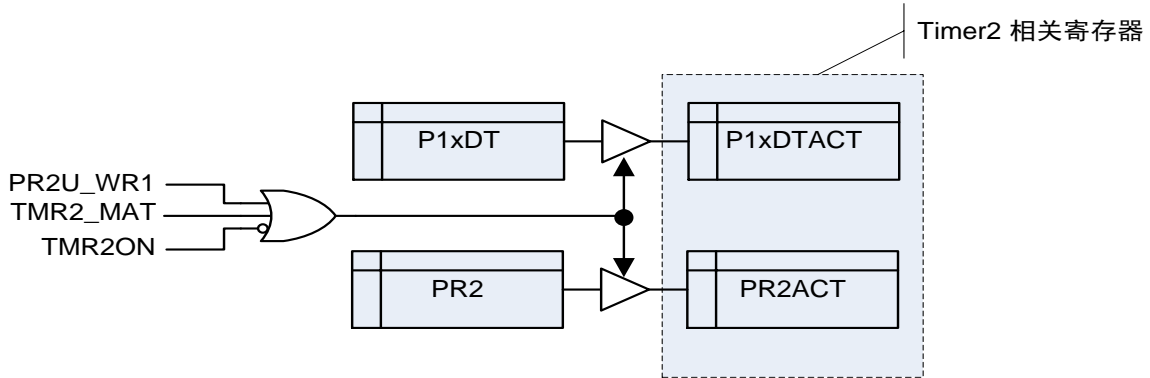


图 10-4 Timer2 寄存器的更新

周期和占空比寄存器的双缓冲读写设计可确保在大部分情况下减少 PWM 输出的毛刺，但如果在非常接近一个周期结束时去更新这些寄存器(特别是在 Timer2 的频率比系统时钟 Sysclk 快的情况下)，则可能发生不可预知的情况，且可能导致 xxxACT 寄存器的值被改为非期望值。

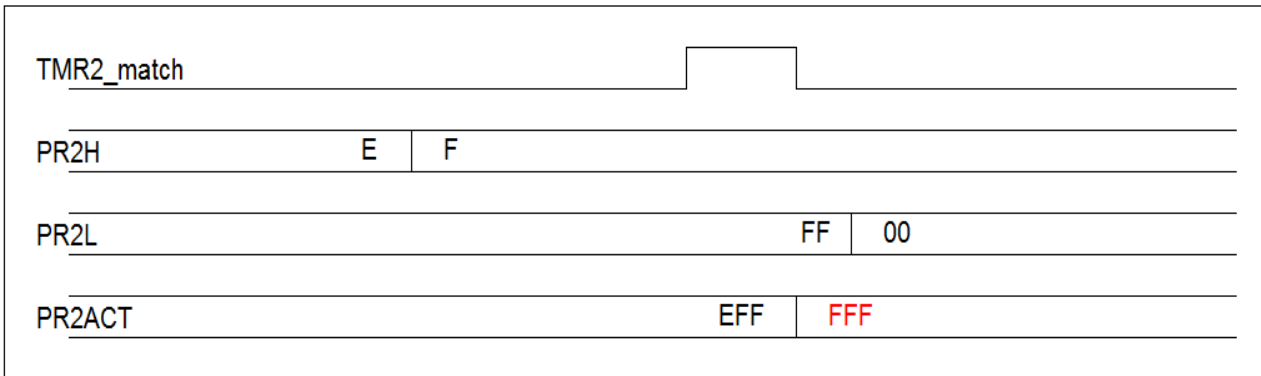


图 10-5 PR2ACT 值被更新成 FFF (期望值为 F00)

因此强烈建议在一个新的周期开始后立即更新 PR2 和 xxxDTx 寄存器。

10.8. PWM 输出

重映射 – 4 路独立占空比的 PWM 通道 P1A, P1B, P1C, P1D, 可映射到不同的 I/O 引脚。PWM1 和 PWM4 可分别映射到 3 个 I/O, PWM2 和 PWM3 可分别映射到 2 个 I/O。

蜂鸣器 (Buzzer) 模式 – 输出周期为 $(2 * (PR2 + 1) * T_{T2CK} * (TMR2 \text{ 预分频值}))$ 。P1A, P1B, P1C 和 P1D 将输出 50% 占空比的方波。

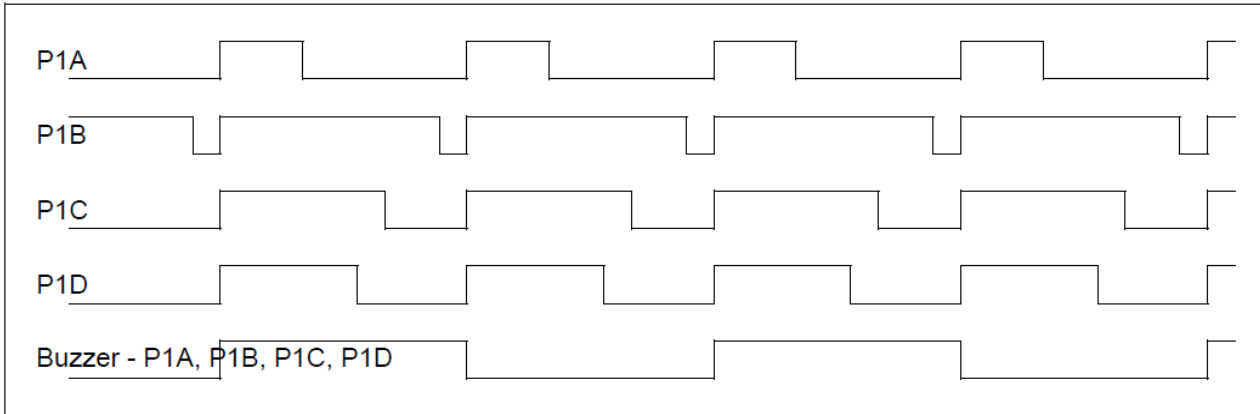


图 10-6 蜂鸣器模式的输出时序图

单脉冲输出 – P1A, P1B, P1C 和 P1D 将只产生一次相应的单脉冲。

10.9. (P1C, P1D) 和 (P1B, P1C) 的第 2 功能输出

PC6 = P1C xor P1D (或 P1C xnor P1D, 参阅 “P1CF2E” 和 “P1CF2”, 表 10-4)。

PC7 和/或 PA7 = P1B xor P1C (或 P1B xnor P1C, 参阅 “P1DF2E” 和 “P1DF2”, 表 10-4)。

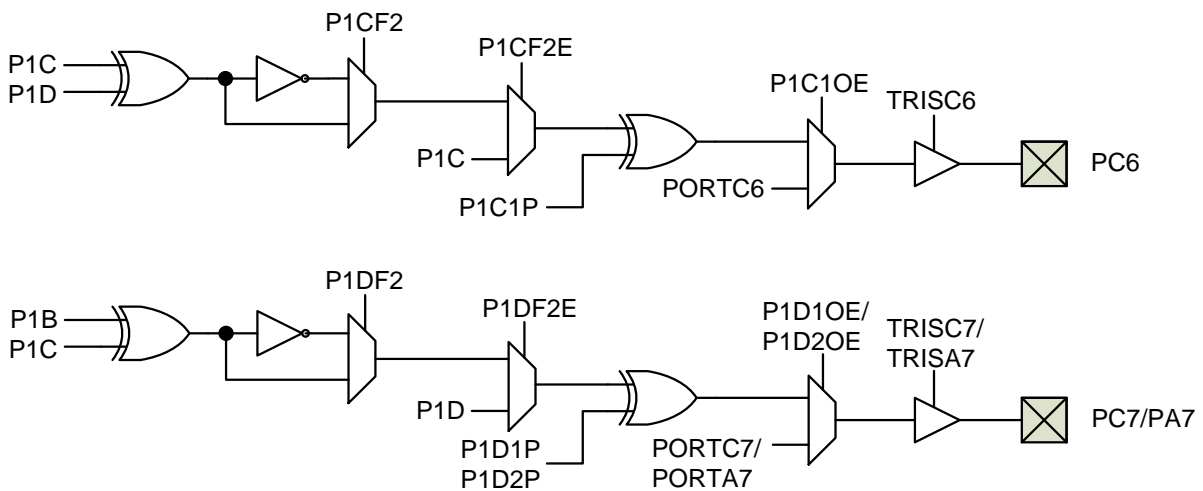


图 10-7 第 2 功能输出结构框图

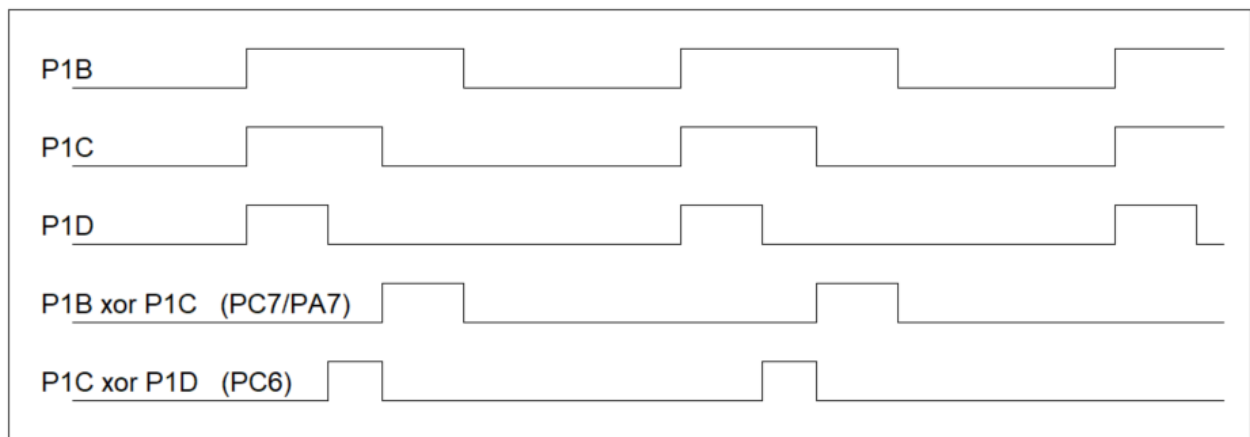


图 10-8 P1B 和 P1C 的第 2 功能时序图

11. 数据 EEPROM (DATA EEPROM)

FT62F28x 片内集成有 128 x 8-bit 的非易失性 DATA EEPROM 存储区，并独立于主程序区。此数据存储区的典型擦写次数可达 100 万次。可通过指令进行读/写访问，每次可读取或写入的单位为 1 个 byte (8-bit)，没有页模式(page mode)。擦除/编程实现了硬件自定时，无需软件查询，以节省有限的代码空间。因此写操作可在后台运行，不影响 CPU 执行其他指令，甚至可进入 SLEEP 状态。

读操作需要 2 个指令时钟周期，而写操作需要的时间为 $T_{WRITE-EEPROM}$ (使能自动擦除为 2 ~ 4 ms，关闭自动擦除则为 0.7 ~ 1.3 ms)。芯片内置有电荷泵，因此不需要提供外部高压，即可对 EEPROM 区进行擦除和编程。写操作完成时将置位相应的中断标志位 EEIF。

不支持连续读(sequential READ) 或连续写(sequential WRITE)，因此每次读/写都必须更新相应的地址。

只要 $V_{DD} \geq V_{POR}$ ，CPU 即可在 8 MHz / 2T 的速度下运行，在高温下甚至可低至 1.5V 左右。而写 DATA EEPROM 所需的电压($V_{DD-WRITE}$) 较高。工作温度等级 2 和等级 1 的最低 $V_{DD-WRITE}$ 分别为 1.9V 和 2.2V。读 DATA EEPROM 没有此最低电压限制(参阅 $V_{DD-READ}$)。

11.1 DATA EEPROM 和 PROM 相关寄存器汇总

名称	状态	寄存器	地址	复位值
EEDAT	DATA EEPROM 数据	EEDAT[7:0]	0x9A	RW-0000 0000
EEADR	DATA EEPROM 地址	EEADR[6:0]	0x9B	RW- 000 0000
WREN3	<u>DATA EEPROM 写使能 (bit 3)</u> 111 = 使能, 完成后重置为 000 (其他) = 关闭	EECON1[5]	0x9C	RW-0
WREN2	DATA EEPROM 写使能 (bit 2)	EECON1[4]		RW-0
WRERR	<u>DATA EEPROM 写错误标志位</u> 1 = 中止 (发生 MCLR 或 WDT 复位) 0 = 正常完成	EECON1[3]		RW-x
WREN1	DATA EEPROM 写使能 (bit 1)	EECON1[2]		RW-0
PONLY	<u>DATA EEPROM 自动擦除</u> 1 = 关闭 (不擦除, 只写) 0 = 使能 (先擦除, 再写)	EECON1[1]		RW-0
RD	<u>DATA EEPROM 读控制位</u> 1 = 使能 (保持 4 个 SysClk 周期, 然后清零) 0 = 关闭	EECON1[0]		RW-0
WR	<u>DATA EEPROM 写控制位</u> 1 = 启动一次写或写正在进行中 (完成后重置为 0) 0 = 完成	EECON2[0]		0x9D

表 11-1 EEPROM 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, 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)	PIE1[7]	0x8C	RW-0
EEIF	EEPROM 写完成中断 标志位	1 = Yes (锁存) 0 = No	PIR1[7]	0x0C	RW-0

表 11-2 EEPROM 中断使能和状态位

11.2 写 DATA EEPROM

1. 设置 “GIE = 0”;
2. 判断 GIE, 如果 “GIE = 1”, 则重复步骤 (1);
3. 将目标地址写入 EEADR;
4. 将目标数据写入 EEDAT;
5. 设置 “WREN3, WREN2, WREN1” = “1, 1, 1”, 并在整个编程过程中保持此设置;
6. 须立即设置 “WR = 1” 以启动写 (否则将中止);
7. 编程完成 (编程时间请参阅 $T_{WRITE-EEPROM}$) 后, “WR” 和 “WREN3, WREN2, WREN1” 都将自动清 0;

示例程序:

```

BCR INTCON, GIE
BTSC INTCON, GIE
LJUMP $-2
BANKSEL EEADR
LDWI 55H
STR EEADR           ; 地址为 0x55
STR EEDAT           ; 数据为 0x55
LDWI 34H
STR EECON1          ; WREN3/2/1 同时置 1
BSR EECON2, 0       ; 启动写
BSR INTCON, GIE     ; GIE 置 1
    
```

注:

1. 当编程正在进行中时, 对 Data EEPROM 进行读操作将导致读取结果错误。
2. 如果编程完成前, WREN3, WREN2 或 WREN1 任意一位被清 0, 在下次编程前需清除 EEIF 标志位。

11.3 读 DATA EEPROM

将目标地址写入 EEADR 寄存器，然后启动读 (“RD = 1”)。2 个指令时钟周期后，EEPROM 数据被写入 EEDAT 寄存器，因而必须在读指令之后紧跟一条 NOP 指令。EEDAT 寄存器将保持此值直至下一次读或写操作。

读 DATA EEPROM 的示例程序如下：

```
BANKSEL EEADR
LDWI dest_addr
STR EEADR
BSR EECON1, RD
NOP                ; 读等待
LDR EEDAT, W       ; 此时，数据可由指令读取
```

11.4 自动擦除功能

将数据写入字节(byte)的过程包括 2 步：先擦除字节，再编程字节。擦除操作将字节的所有 bits 擦成“1”，而编程操作会有选择地将个别 bits 写成“0”。本芯片内置自动擦除功能(设置 PONLY = 0)，即编程前会先自动执行擦除操作。除高温环境外，建议使能自动擦除功能。

如果使能自动擦除，多次编程 FF 数据实际为多次擦除相应字节。然而多次编程非 FF 数据实际只对相应字节进行了一次编程，因为每次编程前都会先自动擦除。只有当自动擦除功能关闭时，重复编程才会有累积效应。某些情况下，比如在非常高的温度下，可能会需要关闭自动擦除功能，并进行重复编程以确保编程成功。流程如下：

1. 确保自动擦除使能。
2. 擦除字节。
3. 读 DATA EEPROM。
4. 如果字节数据为 FF 则继续，否则返回步骤(2)。
5. 再执行相同次数的步骤(2)即擦除操作，以确保擦除强度。
6. 关闭自动擦除。
7. 编程期望值。
8. 读 DATA EEPROM。
9. 如果字节数据为期望值则继续，否则返回步骤(7)。
10. 再执行相同次数的步骤(7)即累积编程，以确保编程强度。

11.5 EEPROM 最后 16 个 byte

EEPROM 的最后 2 页 (0x70~7F, 共 16 byte) 用来存储 ADC 内部参考电压, 以及 OP0 Vos 的 AD 转换值, 软件只能读, 不能写。

地址	名称	说明
0x70	VREF0P5_CAL[7:0]	ADC 内部参考 0.5V 的实测值 (VDD = 5V)
0x71	VREF0P5_CAL[15:8]	
0x72	VREF2P0_CAL[7:0]	ADC 内部参考 2V 的实测值 (VDD = 5V)
0x73	VREF2P0_CAL[15:8]	
0x74	VREF3P0_CAL[7:0]	ADC 内部参考 3V 的实测值 (VDD = 5V)
0x75	VREF3P0_CAL[15:8]	
0x76	N/A	
0x77	N/A	
0x78	VOS_G10_CAL[7:0]	放大倍数配置为 10 (反馈电阻 40k/4k) 时的 AD 转换值 (VREF = 内部 3V, VDD = 5V)
0x79	VOS_G10_CAL[11:8]	
0x7A	VOS_G20_CAL[7:0]	放大倍数配置为 20 (反馈电阻 80k/4k) 时的 AD 转换值 (VREF = 内部 3V, VDD = 5V)
0x7B	VOS_G20_CAL[11:8]	
0x7C	VOS_G40_CAL[7:0]	放大倍数配置为 40 (反馈电阻 160k/4k) 时的 AD 转换值 (VREF = 内部 3V, VDD = 5V)
0x7D	VOS_G40_CAL[11:8]	
0x7E	VOS_G80_CAL[7:0]	放大倍数配置为 80 (反馈电阻 320k/4k) 时的 AD 转换值 (VREF = 内部 3V, VDD = 5V)
0x7F	VOS_G80_CAL[11:8]	

表 11-3 ADC V_{REF} 以及 OP0 Vos 的出厂值

注:

1. ADC 内部参考电压 $VREFxPx_CAL[15:12]$ 表示实测值的整数部分, $[11:0]$ 表示小数部分。例如 $VREF0P5_CAL[15:0] = 0510$, 表示实测值为 0.510V。

12. 12-bit 模/数转换器 (ANALOG TO DIGITAL CONVERTER, ADC)

ADC 模块可将模拟输入信号转换成 12-bit 的数字信号。ADC 可在不同的时钟速度下运行，并且在高达 2MHz 的时钟速度(即 100 kHz 的采样率，10 μs/采样) 下仍具有 11-bit 精度。

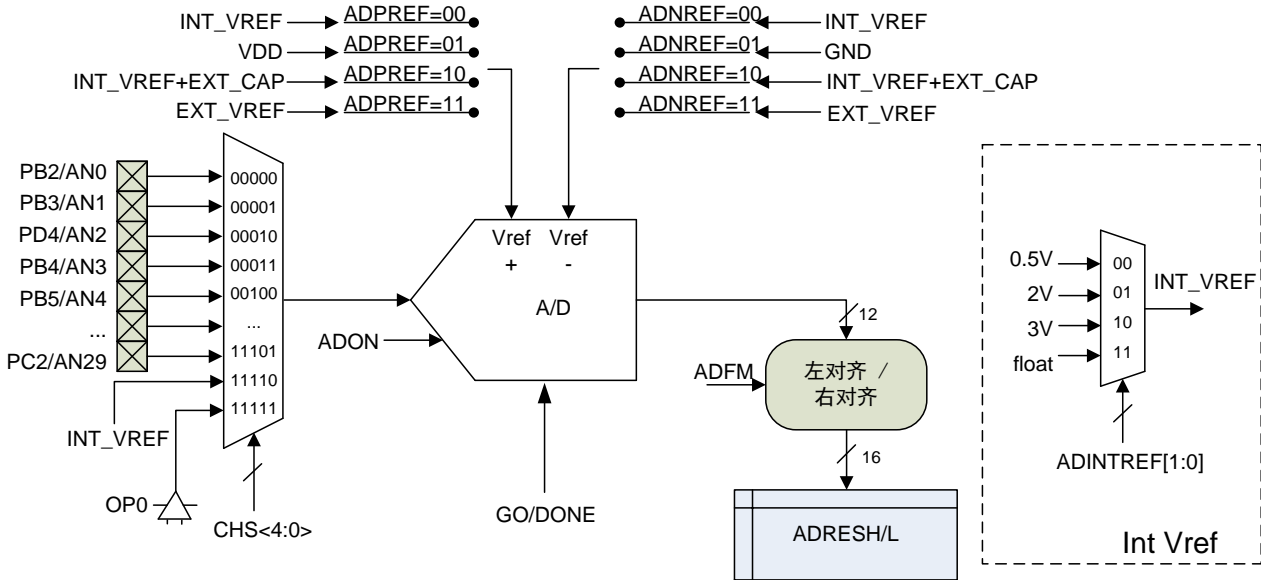


图 12-1 ADC 结构框图

模拟输入信号可选择为 30 个 I/O (ANx) 通道之一、3 个内部参考电压(Internal $V_{ADC-REF}$)之一或内部运放输出。ADC 由指令、I/O (PC4/PD2/PD4) 或 PWM 触发。在触发和 ADC 采样之间可增加延时或前沿消隐(Leading Edge Blanking, LEB)。

当 ADC 转换完成和/或 ADC 阈值比较结果匹配时，将置位相应的中断标志位，并可触发中断和/或从睡眠中唤醒。

ADC 参考电压($V_{ADC-REF}$)可通过指令选择为 V_{DD} ，3 个内部参考电压(0.5V, 2V, 3V)之一，或通过 I/O 输入外部参考电压。

ADC 不需要校准。另外，ADC 转换过程在后台运行，转换期间 CPU 可执行其他指令。

如果 ADC 需要在 SLEEP 下保持运行，且其转换时钟源为 Sysclk 或其分频时，则需设置 SYSON = 1，来使系统时钟 Sysclk 在 SLEEP 下保持运行。当 ADC 的时钟源为 LIRC 时，进入 SLEEP 后 LIRC 将自动开启。

当 ADC 配置为硬件触发(I/O 或 PWM)时，GO/DONE 由硬件触发事件直接置位并启动 A/D 转换，软件置位 GO/DONE 将被忽略。

在高采样率的应用中，使用 ADC 时需注意 3 个时间点：

1. 所选通道开始采样的时刻。
2. 结束采样的时刻。采样保持电路断开前的瞬间，所选通道上的电压值被用于测量转换。
3. 数据转换完成时间。

12.1. ADC 相关寄存器汇总

名称	状态		寄存器	地址	复位值
GIE	全局中断	1 = 使能 (PEIE, ADCIE, ACMPIE 适用) 0 = <u>全局关闭</u> (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, ACMPIE 适用) 0 = <u>关闭</u> (无唤醒)	INTCON[6]		RW-0
ADCIE	ADC 转换完成中断	1 = 使能 0 = <u>关闭</u> (无唤醒)	PIE1[0]	0x8C	RW-0
ADCIF	ADC 转换完成中断标志位	1 = Yes (锁存) 0 = <u>No</u>	PIR1[0]	0x0C	RW-0
ACMPIE	ADC 阈值比较匹配中断	1 = 使能 0 = <u>关闭</u> (无唤醒)	PIE1[4]	0x8C	RW-0
ACMPIF	ADC 阈值比较匹配中断标志位	1 = Yes (锁存) 0 = <u>No</u>	PIR1[4]	0x0C	RW-0

表 12-1 ADC 中断使能和状态位

名称	状态		寄存器	地址	复位值
ADRESL	<u>ADC 转换结果低有效位 (LSB)</u>		ADRESL[7:0]	0x111	RW-xxxx xxxx
	ADFM=0: ADRESL[7:4] = 低 4 位 (其余为“0”) ADFM=1; ADRESL[7:0] = 低 8 位				
ADRESH	<u>ADC 转换结果高有效位 (MSB)</u>		ADRESH[7:0]	0x112	RW-xxxx xxxx
	ADFM=0: ADRESH[7:0] = 高 8 位 ADFM=1: ADRESH[3:0] = 高 4 位 (其余为“0”)				
LEBEN	<u>ADC 触发和 BKIN 的 LEB 使能位</u> 1 = 使能 (当 GO/DONE=1 时进行切换将产生不可预知的结果) 0 = <u>关闭</u>		LEBCON[7]	0x10E	RW-00
LEBCH	<u>LEB 信号源</u> (<G 版芯片)	00 = <u>P1A0</u> 10 = P1C 01 = P1B 11 = P1D	LEBCON[6:5]		
	(≥ G 版芯片为保留位)				
EDGS	<u>LEB 触发沿</u> (<G 版芯片)	1 = PWM 下降沿 0 = <u>PWM 上升沿</u>	LEBCON[3]	RW-0	
	<u>LEB 结束后, ADC 开始自动转换</u> (≥ G 版芯片)	1 = 触发 ADC 转换 0 = <u>不触发 ADC 转换</u>			

名称	状态	寄存器	地址	复位值																																
ADCMPH	ADC 比较阈值 (仅高 8 位, 0.4% steps)	ADCMPH[7:0]	0x10F	RW-0000 0000																																
ADDLY / LEBPRL	<p><u>ADC 延迟/ LEB (非软件触发, 即 ADEX = 1 有效)</u> (此为低 8 位, ADDLY.8 为高有效位)</p> <p>外部延迟时间 = (ADDLY+6) × T_{AD} 外部延迟时间 = (ADDLY+3) × T_{AD} (≥ VerG 版芯片)</p> <p>注: 如果启用 PWM 输出触发 ADC, 在 PWM 运行过程中不得更改 ADDLY;</p>	ADDLY[7:0]	0x110	RW-0000 0000																																
CHS	<p><u>ADC 模拟输入通道</u></p> <table border="0"> <tr><td>00000 = AN0</td><td>10000 = AN16</td></tr> <tr><td>00001 = AN1</td><td>10001 = AN17</td></tr> <tr><td>00010 = AN2</td><td>10010 = AN18</td></tr> <tr><td>00011 = AN3</td><td>10011 = AN19</td></tr> <tr><td>00100 = AN4</td><td>10100 = AN20</td></tr> <tr><td>00101 = AN5</td><td>10101 = AN21</td></tr> <tr><td>00110 = AN6</td><td>10110 = AN22</td></tr> <tr><td>00111 = AN7</td><td>10111 = AN23</td></tr> <tr><td>01000 = AN8</td><td>11000 = AN24</td></tr> <tr><td>01001 = AN9</td><td>11001 = AN25</td></tr> <tr><td>01010 = AN10</td><td>11010 = AN26</td></tr> <tr><td>01011 = AN11</td><td>11011 = AN27</td></tr> <tr><td>01100 = AN12</td><td>11100 = AN28</td></tr> <tr><td>01101 = AN13</td><td>11101 = AN29</td></tr> <tr><td>01110 = AN14</td><td>11110 = (内部 V_{ADC-REF})</td></tr> <tr><td>01111 = AN15</td><td>11111 = 运放 0 输出</td></tr> </table>	00000 = AN0	10000 = AN16	00001 = AN1	10001 = AN17	00010 = AN2	10010 = AN18	00011 = AN3	10011 = AN19	00100 = AN4	10100 = AN20	00101 = AN5	10101 = AN21	00110 = AN6	10110 = AN22	00111 = AN7	10111 = AN23	01000 = AN8	11000 = AN24	01001 = AN9	11001 = AN25	01010 = AN10	11010 = AN26	01011 = AN11	11011 = AN27	01100 = AN12	11100 = AN28	01101 = AN13	11101 = AN29	01110 = AN14	11110 = (内部 V _{ADC-REF})	01111 = AN15	11111 = 运放 0 输出	ADCON0[7:3]	0x113	RW-00000
00000 = AN0	10000 = AN16																																			
00001 = AN1	10001 = AN17																																			
00010 = AN2	10010 = AN18																																			
00011 = AN3	10011 = AN19																																			
00100 = AN4	10100 = AN20																																			
00101 = AN5	10101 = AN21																																			
00110 = AN6	10110 = AN22																																			
00111 = AN7	10111 = AN23																																			
01000 = AN8	11000 = AN24																																			
01001 = AN9	11001 = AN25																																			
01010 = AN10	11010 = AN26																																			
01011 = AN11	11011 = AN27																																			
01100 = AN12	11100 = AN28																																			
01101 = AN13	11101 = AN29																																			
01110 = AN14	11110 = (内部 V _{ADC-REF})																																			
01111 = AN15	11111 = 运放 0 输出																																			
ADEX	<p><u>ADC 触发条件 (GO/DONE)</u></p> <p>1 = 由 ADC_ETR 或 PWM 置位 GO/DONE (硬件触发) 0 = 由指令置位 GO/DONE (软件触发)</p>	ADCON0[2]		RW-0																																
GO/DONE	<p><u>ADC 转换启动和状态位</u></p> <p>1 = 由软件、ADC_ETR 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</p>	ADCON0[1]		RW-0																																
ADON	<p>1 = ADC 使能 0 = <u>ADC 关闭</u> (无电流消耗)</p>	ADCON0[0]		RW-0																																
LFMOD	<p>1: LIRC = 256 kHz 0: LIRC = 32 kHz</p>	OSCCON[7]	0x8F	RW-0																																

名称	状态	寄存器	地址	复位值
ADFM	<u>A/D 转换结果格式 (参阅 “ADRESH”)</u> 1 = 右对齐 0 = 左对齐	ADCON1[7]	0x114	RW-0
ADCS	<u>ADC 转换时钟源</u> TSEL = 2T TSEL = 4T 000 = SysClk/2 000 = SysClk/4 001 = SysClk/8 001 = SysClk/16 010 = SysClk/32 010 = SysClk/64 011 = SysClk 011 = SysClk/2 100 = SysClk/4 100 = SysClk/8 101 = SysClk/16 101 = SysClk/32 110 = SysClk/64 110 = SysClk/128 111 = LIRC 111 = LIRC	ADCON1[6:4]		RW-010
ADNREF	<u>V_{ADC-REF-} (负参考电压)</u> 00 = 内部 V _{ADC-REF} 01 = <u>GND</u> 10 = 内部 V _{ADC-REF} + 外部电容 C _{EXT} 11 = 外部参考电压 (I/O)	ADCON1[3:2]		RW-01
ADPREF	<u>V_{ADC-REF+} (正参考电压)</u> 00 = 内部 V _{ADC-REF} 01 = V _{DD} 10 = 内部 V _{ADC-REF} + 外部电容 C _{EXT} 11 = 外部参考电压 (I/O)	ADCON1[1:0]		RW-00
ADINTREF	<u>内部 V_{ADC-REF}</u> 00 = 0.5 01 = <u>2.0</u> 10 = 3.0 11 = (未连接)	ADCON2[7:6]		RW-01
ETGTYP	<u>外部触发沿 (当 ADEX=1 时适用)</u> 00 = (PWM 或 ADC_ETR) 下降沿 01 = (PWM 或 ADC_ETR) 上升沿 1x = 保留值 注: ≥G 版芯片, 也作 LEB 触发沿选择	ADCON2[5:4]		0x115 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

12.2. ADC 配置

配置 ADC 包括以下设置 (更改配置时需设置 $ADON = 0$ 以关闭 A/D 转换或外部触发):

- 通道选择
- ADC 参考电压
- ADC 转换时钟源
- 转换结果格式
- 触发源
- ADC 延时或前沿消隐 (LEB)
- 阈值比较 (可选)
- 响应 (中断设置)

通道选择 – 由 CHS 寄存器选择输入通道, 连接到用于 ADC 转换的采样保持电路。相应的 I/O 需设置 $TRISx = 1$ 和 $ANSELx = 1$ 来配置成模拟输入。

ADC 参考电压 ($V_{ADC-REF}$) – ADC 以 2 个参考电压作为相对值来测量输入模拟电压: V_{REF+} 和 V_{REF-} 。参考电压可以选择为:

- V_{REF+} 可选 VDD, V_{REF-} 可选 GND
- 内部参考电压
- 内部参考电压加外部电容 C_{EXT}
- 外部参考电压 (V_{REF+} 为 PB5, V_{REF-} 为 PB6)

V_{REF+} 和 V_{REF-} 可以为上述选择的不同组合, 但不可以同时选择内部参考电压, 否则 V_{REF-} 将强制连接到 GND。

内部参考电压可以为 0.5V, 2.0V, 3.0V, 或 “未连接” (参阅 “ADINTREF”, 表 12-2)。其中, 内部参考电压的出厂值存储在 EEPROM 的 0x70~0x75 地址, 具体读取方法请参阅 章节 11.5。

ADC 转换时钟选择 – ADC 可通过指令选择 8 种时钟频率 (参阅 “ADCS”, 表 12-2):

- $TSEL = 2T$ 时为 $SysClk/N$; $TSEL = 4T$ 时为 $SysClk/2N$; $N = 1, 2, 4, 8, 16, 32, 64$
- LIRC (256 kHz 或 32 kHz, 参阅 “LFMOD”, 表 6-2)

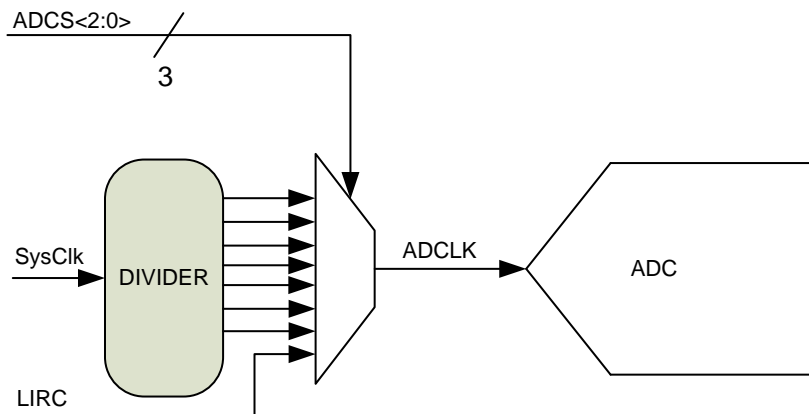


图 12-2 ADC 时钟配置

转换结果格式 – A/D 转换结果可储存为左对齐或右对齐两种格式(参阅 "ADFM", 表 12-2)。

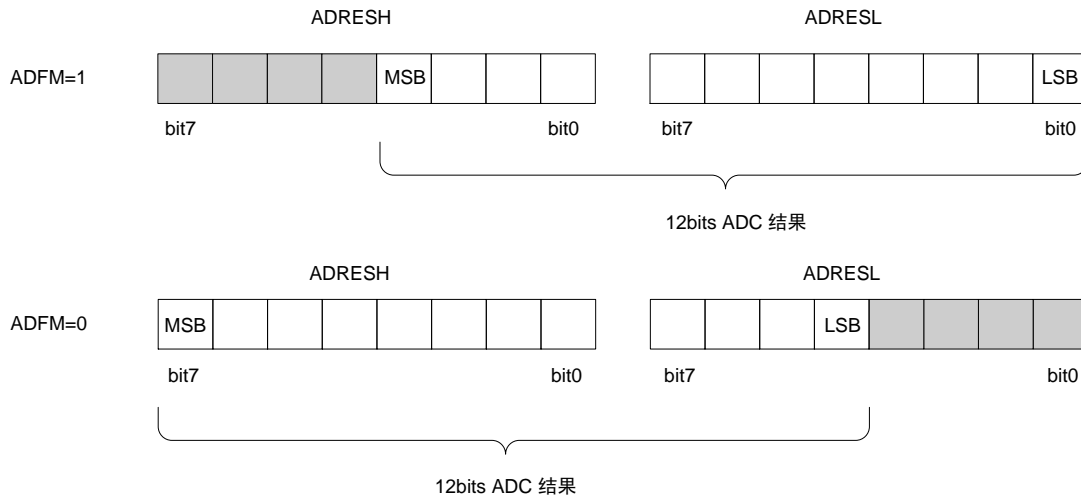


图 12-3 ADC 转换结果格式

12.2.1. ADC 触发和延时配置

ADC 转换可由指令(ADEX = 0)、PWM 边沿或 I/O(PC4/PD2/PD4)转变沿(ADEX = 1)触发。其中, PWM 或 I/O 的触发沿可选择为 "上升沿" 或 "下降沿" (参阅 "ETGTYP", 表 12-2)。

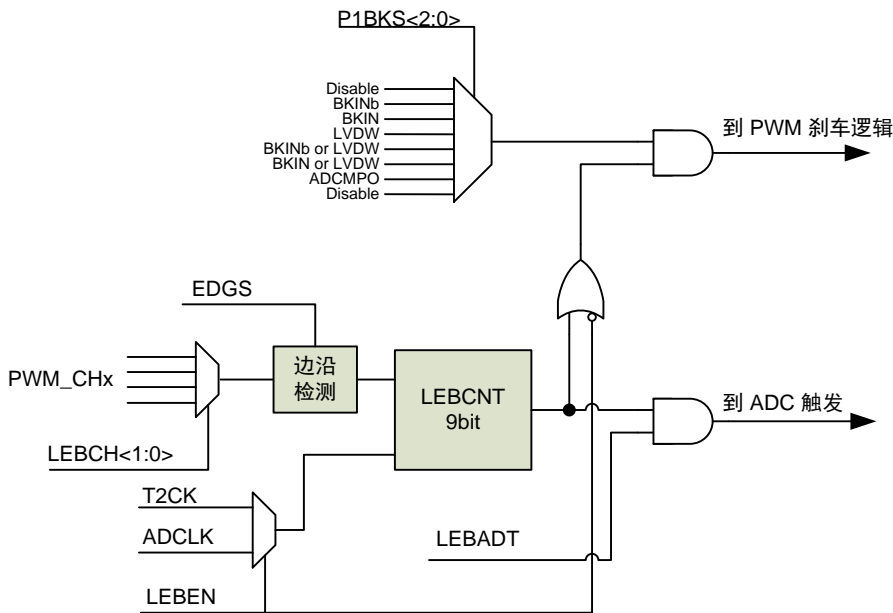


图 12-4 LEB 结构框图

在高速开关应用中, 开关(如 MOSFETs/IGBTs)导通瞬间通常会产生极大的瞬变电流, 而这些瞬变会导致测量误差。利用前沿消隐(LEB)功能, 应用程序可忽略 PWM 输出边沿附近由 MOSFETs/IGBTs 开关所导致的预期瞬变。

LEB 和 PWM 的时钟源均为 T2CK (Timer2 时钟源)。LEB 计时期间, ADC 保持采样, 直至 LEB 计时溢

出(参阅“LEBPR”，表 12-2)。在 LEB 计时周期内如果再次发生有效的 LEB 触发沿，则 LEB 定时器将清 0 并重新开始计数。

触发条件	延迟 / 消隐	触发通道
指令	(无延迟)	(N/A)
I/O (PA4)	(ADDLY+6) × T _{AD} ; (< VerG芯片) (ADDLY+3) × T _{AD} ; (≥ VerG芯片) ADDLY = LEBPR	I/O (PC4/PD2/PD4)
PWM	(LEBPR+6) × T _{AD} ; (< VerG芯片) (LEBPR+3) × T _{AD} ; (≥ VerG芯片)	LEBEN = 0; ETGSEL (LEBCH 忽略)
	(LEBPR+4) × T _{T2CK} + 2 × T _{AD} ; (< VerG芯片) (LEBPR+3) × T _{T2CK} ; (≥ VerG芯片) (T _{T2CK} = Timer2 period)	LEBEN = 1; LEBCH (ETGSEL 忽略)

表 12-4 ADC 触发，延迟和通道设置

如果由软件触发(ADEX = 0)，GO/DONE 由指令置位后立即启动 A/D 转换。如果由 I/O 或 PWM 触发，则有一定的延迟时间(“3 × T_{AD}”或“4 × T_{T2CK} + 2 × T_{AD}”，参阅表 12-4)。另外可通过设置 ADDLY/LEBPR 寄存器在 GO/DONE 置位前增加额外的延迟。ADC 延时定时器(ADDLY)和 LEB 定时器(LEBPR)共用同一个 9-bit 计数器，此计数器由 LEBPR9 和 LEBPRL[7:0]组成。延迟结束后采样保持电路将在“0 – 1 × T_{AD}”时间内断开。

注：

1. 在使能 LEB 前，需先设置 ADEX 和 ADON 寄存器。
2. ADC 转换完成前将忽略新的触发条件。
3. 如果 LEBEN=1，则将忽略 ETGSEL，触发源即为 LEB 的触发源。此时由 LEB 定时器溢出触发 ADC 自动转换(参阅“LEBADT”或“EDGS”，表 12-2)。

12.2.2. ADC 中止转换

有时需中止 ADC 转换，比如需启动新的 ADC 采样。

- 当 ADEX = 0 (指令触发)时，可通过软件设置 GO/DONE = 0 来中止 ADC。
- 当 ADEX = 1 时，必须通过关闭 ADC 模块(ADON = 0)来中止 ADC。
- 当 ADC 转换被中止时，ADRESH 和 ADRESL 不会被更新，而是保持前一次的转换结果值。
- 系统复位时，由于相应的寄存器被复位，因此 ADC 将中止，且 ADC 模块被关闭。

12.2.3. 中断

ADC 模块在发生下列事件时将置位相应的中断标志位：

- ADC 转换完成 (ADCIF)
- ADC 阈值比较匹配 (ACMPIF)

每个中断模块都有其相应的中断使能位(ADCIE 和 ACMPIE), 和更高层级的外设总中断(PEIE), 以及最高级别的全局中断(GIE)。

无论中断使能位是否打开, 发生中断事件时都将置位相应的中断标志位。是否触发中断和/或从睡眠中唤醒则取决于相应的使能控制位(GIE, PEIE, ADCIE 和 ACMPIE)。

注: ADC 转换完成后会自动将结果与 ADCMPH 寄存器里的阈值进行比较(参阅“ADCMPEN”, 表 12-2)。由 ADCMPP 设置比较极性, 当产生相应的匹配条件时将置位中断标志位 ACMPIF。仅转换结果的高 8 位用于阈值比较, 因此 V_{REF+} 和 V_{REF-} 之间的比较 step 为 0.4%。当 ADFBEN = 1 时, 也将使能相应中断。

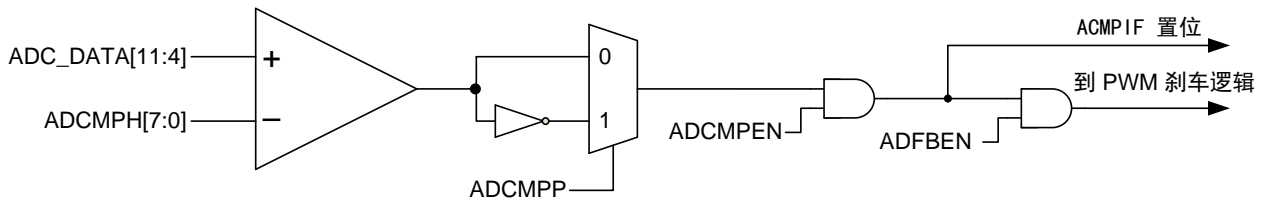


图 12-5 ADC 阈值比较结构框图

12.3. ADC 采样保持时间

采样保持时间 T_{ACQ} , 必须足够长以保证内部 ADC 电压稳定在输入通道电压的 0.01% 误差以内, 从而达到 12bit 的精度(0.024%)。采样保持时间和外部串联电阻的关系如下(表 12-5):

$$T_{ACQ} > 0.09 \times (R + 1) \mu s; R \text{ 的单位为 } k \Omega$$

当采样保持时间 T_{ACQ} 为 $2\mu s$ 时, 外部串联电阻必须 $\leq 21 k\Omega$ 。如果使用更大的串联电阻, 则 T_{ACQ} 将成比例增加。结点漏电流限制了允许使用的最大串联电阻值。对于 5nA 的结点漏电流, 在 $50 k\Omega$ 的串联电阻上将产生 $0.25mV$ ($2V$ 参考电压的 0.0125%) 的压降。而当温度超过 $100^\circ C$ 时, 结点漏电流将大幅提高。因此, 串联电阻越小越好。

串联电阻值	T_{ACQ}
$> 50k\Omega$	(不推荐)
$43 k\Omega$	$\geq 4.0 \mu s$
$21 k\Omega$	$\geq 2.0 \mu s$
$< 21 k\Omega$	$\geq 2.0 \mu s$

表 12-5 不同的外部串联电阻与最短 T_{ACQ} 的对应关系

采样保持时间即为内部 ADC 观测输入通道电压的时间。

采样保持时间的开始 = 通道切换(参阅“CHS”)后或 ADC 稳定(参阅 T_{ST})后, 以时间较长者为准。

采样保持时间的结束 = 硬件触发延迟结束或软件 GO/DONE 置 1 后的 $0 - 1 \times T_{AD}$ 时间内, 延迟时间由触发条件决定(参阅表 12-4), 同时采样保持电路断开。

采样点 = 采样保持电路断开前的瞬间, 有 $0 - 1 \times T_{AD}$ 时间的不确定性。

采样断开后开始数据转换, 转换过程需 $15 \times T_{AD}$ 时间。因此从硬件触发延迟结束或软件 GO/DONE 置 1

后到数据转换完成需要 $15 \times T_{AD}$ 到 $16 \times T_{AD}$ 时间。数据转换完成后，采样保持电路重新闭合，开始下一个采样周期，同样需等待足够长的采样时间 T_{ACQ} 后，才能再次启动 A/D 转换。

12.4. ADC 最短采样时间

T_{AD} 为 ADC 的时钟周期。完整的 12-bit 转换所需最短时间： $T_{ACQ} + 16 \times T_{AD}$

可保证 11-bit 精度的最高转换采样率为 100 kHz (~10 μ s/采样)。

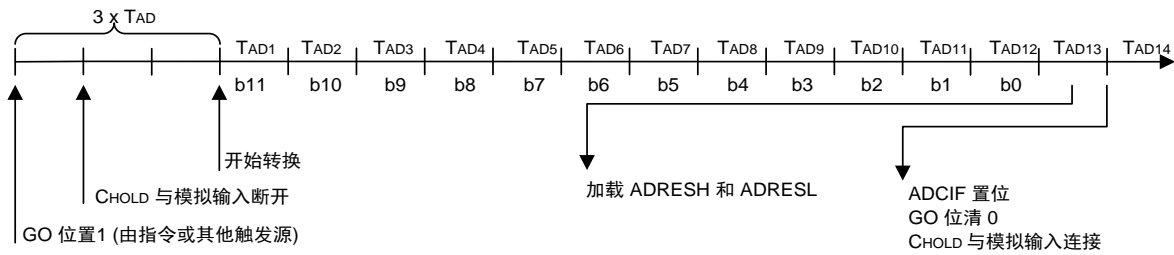


图 12-6 模数转换 T_{AD} 周期

12.5. ADC 转换步骤示例

设置 ADC:

1. 配置端口:
 - a. 设置 $TRISx = 1$ ，禁止引脚输出驱动;
 - b. 设置 $ANSELx = 1$ ，关闭数字输入、弱上拉和弱下拉功能;
2. 配置 ADC 模块:
 - a. 选择 ADC 转换时钟源;
 - b. 选择 ADC 参考电压;
 - c. 选择 ADC 触发条件: 软件、ADC_ETR 或 PWM, 有或无 LEB;
 - d. 选择转换结果格式;
 - e. 使能阈值比较(可选);
3. 配置 ADC 中断(可选):
 - a. 使能 ADC 转换完成和/或阈值比较中断;
 - b. 使能外设总中断;
 - c. 关闭全局中断(如需执行中断服务程序则使能);
4. 打开 ADC 模块。然后等待所需 ADC 稳定时间 T_{ST} (~15 μ s), 当 $V_{ADC-REF}$ 选择内部参考电压时, 则需等待内部参考电压的稳定时间 T_{VRINT} (参阅“ T_{VRINT} ”, 章节 23.8) 和 T_{ST} 时间的较长者, 即 $\max(T_{VRINT}, T_{ST})$ 。

至此, ADC 已准备好对不同的通道进行采样。对输入通道采样时:

1. ADC 输入选择为需测量的通道 (参阅“CHS”)
2. 如有必要, 需清除 ADC 转换完成或阈值比较中断标志位。

3. 对采样时间 T_{ACQ} 有最低要求, T_{ACQ} 需足够长以保证内部 ADC 输入电容充满至输入通道电压的 0.01% 误差以内。另外取决于触发类型, 切换通道后或 ADC 稳定后(以时间较长者为准)可能会有一定的延迟再触发。
 - a. 对于软件触发, 需要额外的 T_{ACQ} 时间。
 - b. 对于 ADC_ETR 或 PWM 触发, 除非使用非常大的串联电阻, 否则内部延迟时间 $(ADDLY+3) \times T_{AD}$ 通常大于 T_{ACQ} , 因此不需要再额外延迟 T_{ACQ} 。
5. 等待所需的延迟后, 由指令置位 GO/DONE, 或等待硬件触发事件自动置位 GO/DONE, 以启动 A/D 转换。
6. 通过以下方式等待 ADC 转换完成:
 - a. 等待 1 个 sysclk 周期后, 查询 GO/DONE 位;
 - b. 等待 ADC 中断 (使能中断时);
7. 读取 ADC 转换结果。
8. 如有必要, 清除 ADC 转换完成或阈值比较中断标志位。

注:

1. 虽然 GO/DONE 和 ADON 在同一个寄存器(ADCON0)中, 但不应同时设置。
2. ADC 转换过程中或等待外部触发时, 不可更改配置。建议在 ADON = 0 时进行更改。

以下为 ADC 程序示例 (输入采样通道为 PB2, ADC 时钟为 LIRC):

```

BANKSEL ADCON1
LDWI B'01110000'           ; ADC LIRC clock
STR ADCON1
BANKSEL TRISB
BSR TRISB, 2               ; Set PB2 to input
BANKSEL ANSEL0
BSR ANSEL0, 0              ; Set-PB2 to analog
BANKSEL ADCON0
LDWI B'10000001'          ; Right justify,
STR ADCON0                 ; VDD, Vref, AN0, On
LCALL StableTime           ; ADC stable time
LCALL SampleTime           ; Acquisition delay, TACQ
BSR ADCON0, GO             ; Start conversion
NOP                        ; GO/DONE ReadBack WaitTime
BTSC ADCON0, GO            ; Conversion done?
LJUMP $-1                  ; No, test again
BANKSEL ADRESH;
LDR ADRESH, W              ; Read upper 4 bits
STR RESULTHI               ; Store in SRAM space
BANKSEL ADRESL;
LDR ADRESL, W              ; Read lower 8 bits
STR RESULTLO               ; Store in SRAMspace

```

13. 运算放大器 (OPAMP)

运算放大器可用于对模拟信号进行处理，特性：

- 输入失调电压可读
- 单位增益带宽：3.64 MHz
- 反相端输入：外部直接连接或串 4kΩ 电阻
- 输出反馈到反相端：40、80、160 或 320kΩ 电阻
- 输出可连接外部管脚

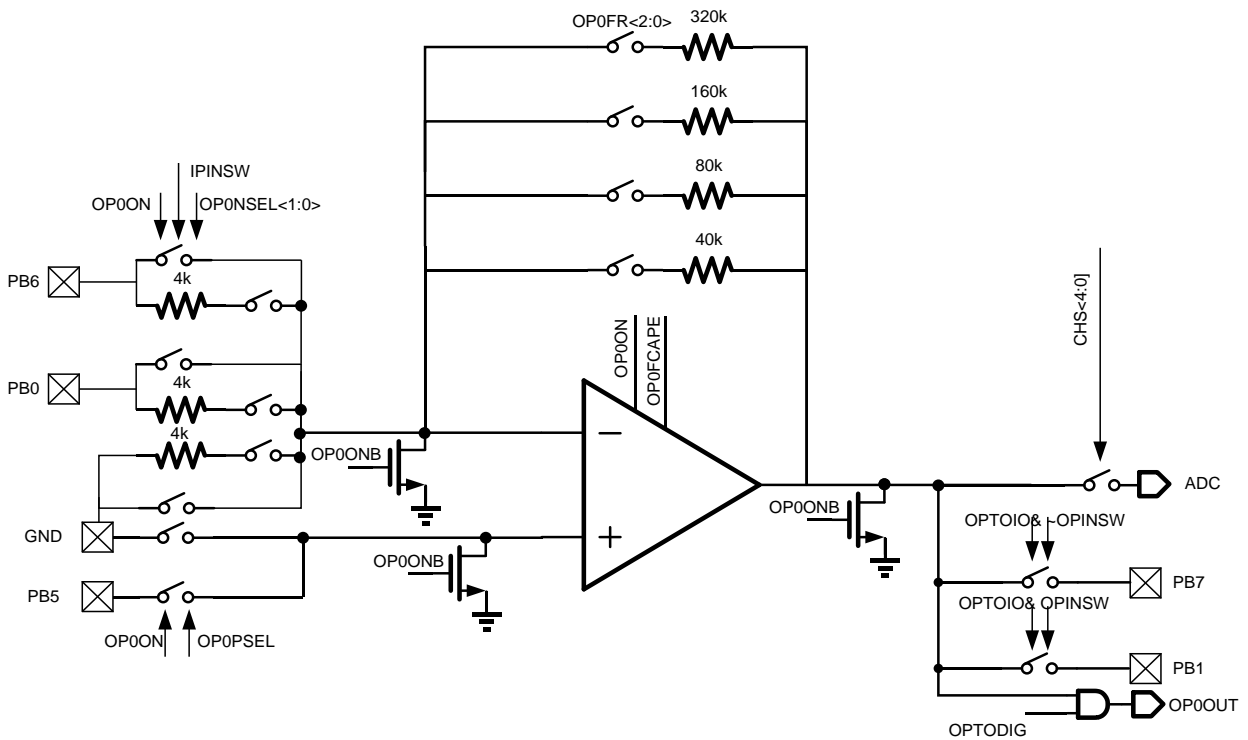


图 13-1 运算放大器结构框图

运放的输入失调电压 V_{os} 存储在 EEPROM 的 0x78~0x7F 地址，具体读取方法请参阅 [章节 11.5](#)。

13.1. 运放相关寄存器汇总

名称	地址	bit 7	bit 6	bit 5	bit 4	bit 3	Bit 2	bit 1	bit 0	复位值
OP0CR0	0x96	OP0OUT	OP0PSEL	OP0NSEL[1:0]		OP0FR[2:0]			OP0ON	x110 1100
OP0CR1	0x97	-	-	OPINSW	IPINSW	OP0FCAPE	-	OPTODIG	OPTOIO	--00 0-00

表 13-1 运放相关寄存器地址

名称	状态		寄存器	地址	复位值
OP0OUT	运放输出寄存器		OP0CR0[7]	0x96	RO-x
OP0PSEL	运放正相端输入 1 = <u>GND</u> 0 = PB5		OP0CR0[6]		RW-1
OP0NSEL	运放反相端输入 00 = GND 01 = 外部输入管脚 (由 IPINSW 决定) 10 = <u>串联 4k 电阻连接到外部输入管脚</u> (由 IPINSW 决定) 11 = 串联 4k 电阻连接到 GND		OP0CR0[5:4]		RW-10
OP0FR	运放反馈电阻 0xx = 无 100 = 40k 101 = 80k 110 = <u>160k</u> 111 = 320k		OP0CR0[3:1]		RW-110
OP0ON	运放	1 = 使能 0 = <u>禁止</u>	OP0CR0[0]	0x97	RW-0
OPINSW	运放输出管脚	1 = PB1 0 = <u>PB7</u>	OP0CR1[5]		RW-0
IPINSW	运放反相端外部输入管脚	1 = PB0 0 = <u>PB6</u>	OP0CR1[4]		RW-0
OP0FCAPE	运放补偿电容	1 = <u>禁止</u> (运放用作比较器) 0 = 使能	OP0CR1[3]		RW-0
OPTODIG	运放输出到寄存器 OP0OUT	1 = 使能 0 = <u>禁止</u>	OP0CR1[1]		RW-0
OPTOIO	运放输出到管脚	1 = 使能 (由 OPINSW 决定) 0 = <u>禁止</u>	OP0CR1[0]		RW-0

表 13-2 运放相关寄存器

13.2. 运放的典型应用

电流检测应用，选择反相端输入串联 4k 电阻到 PB0/PB6 (OP0NSEL = 10)，设置反馈电阻为 160kΩ (OP0FR = 110)，ADC 模拟输入通道选择运放输出 (CHS = 11111)，正相端可配置成以下两种方式：

- 配置正相端输入接地 (OP0PSEL = 1)。

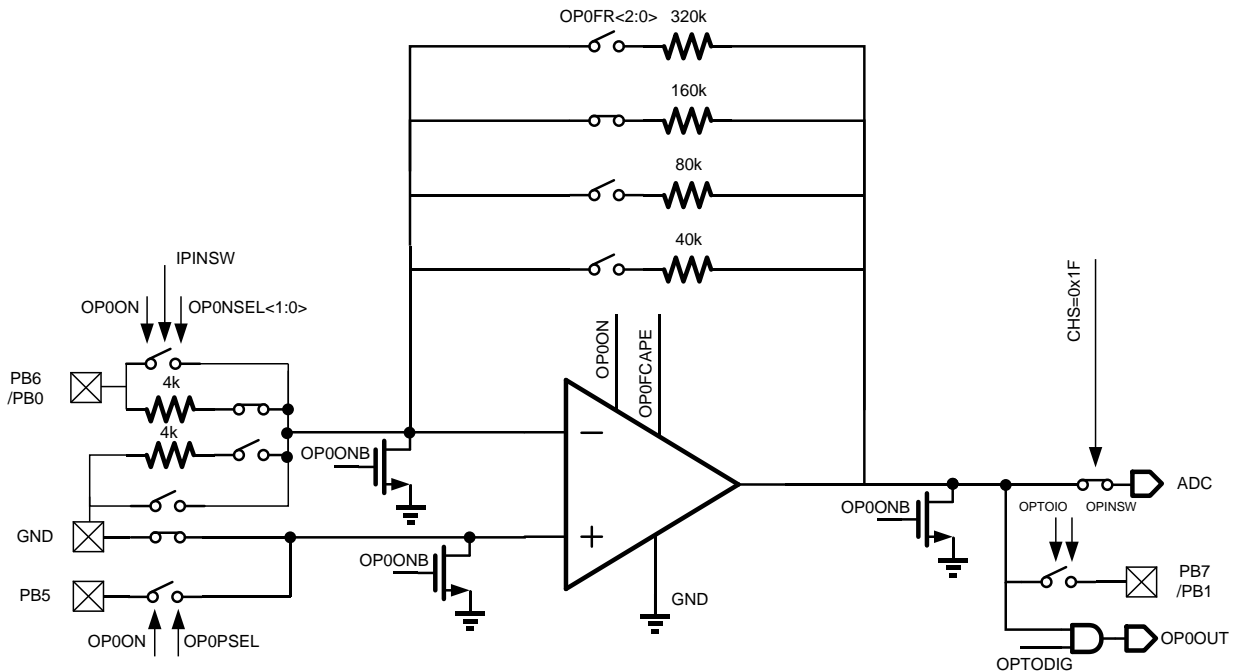


图 13-2 电流检测连接方式 1

- 配置正相端输入直接连接到 PB5 (OP0PSEL = 0)。

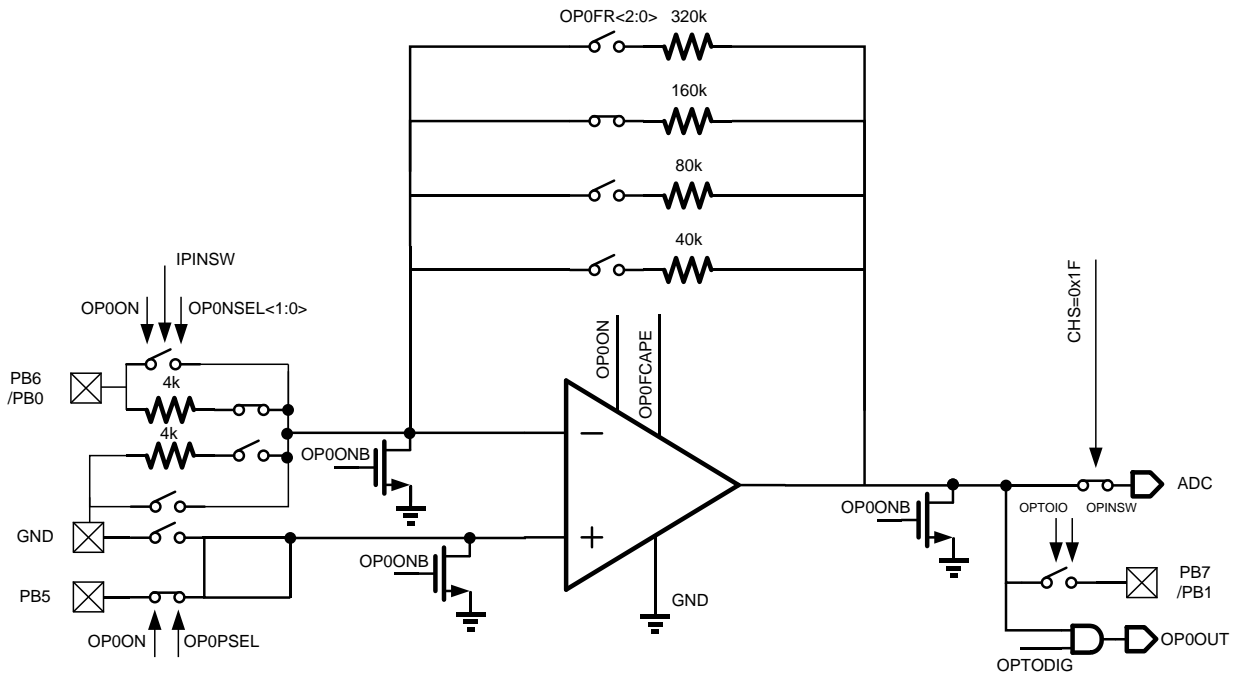


图 13-3 电流检测连接方式 2

14. 比较器

FT62F28x 片内集成 2 个模拟比较器，可分别用于比较 2 个模拟输入电压值的大小。

- 比较器 CM0/CM1 可配置成窗口比较模式
- 反相输入可连接 7bit DAC 输出 (可编程)，或运放输出 OP0OUT
- 输出可去抖
- 输出结果极性可选，并可输出到引脚
- 输出可作为 PWM1 故障刹车源
- 输出可触发唤醒或中断 (电平变化中断)

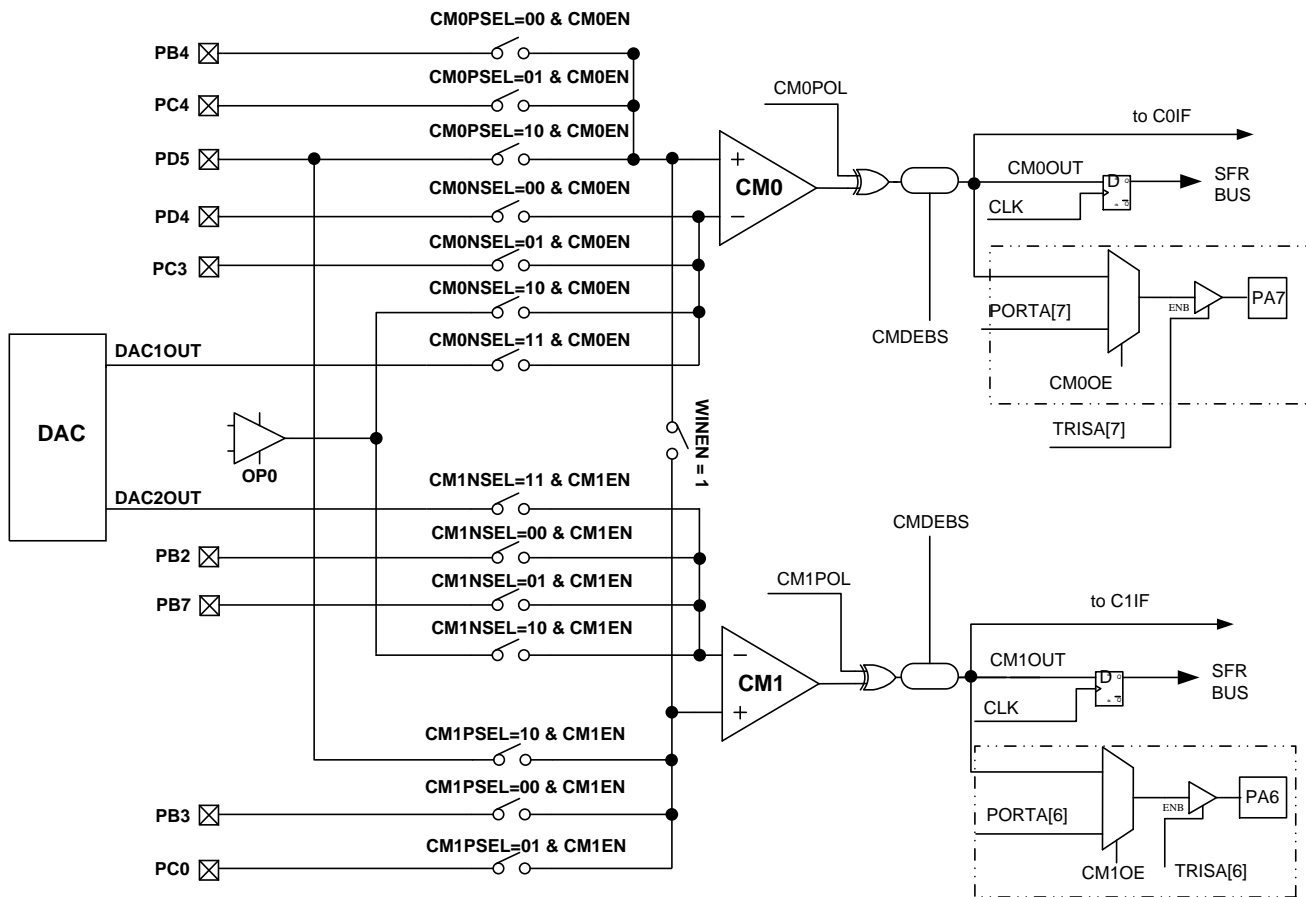


图 14-1 比较器 C0/C1 结构框图

14.1. 比较器相关寄存器汇总

名称	状态		寄存器	地址	复位值
CM0EN	比较器 C0	1 = 使能 0 = <u>禁止</u>	CM0CON0[7]	0x29D	RW-0
CM0POL	比较器 C0 输出极性	1 = 反向 0 = <u>正常</u>	CM0CON0[6]		RW-0
CM0OE	比较器 C0 输出到 PA7	1 = 使能 0 = <u>禁止</u>	CM0CON0[5]		RW-0
C0OUT	<u>比较器 C0 输出结果</u> 当 CM0POL = 0: 当 CM0POL = 1: 1 = CM0+ > CM0- 1 = CM0+ < CM0- 0 = <u>CM0+ < CM0-</u> 0 = <u>CM0+ > CM0-</u>		CM0CON0[4]		RO-x
CM0PSEL	<u>比较器 C0 正相端输入 (CM0+)</u> 00 = <u>PB4</u> 10 = PD5 01 = PC4 11 = (未连接)		CM0CON0[3:2]		RW-00
CM0NSEL	<u>比较器 C0 反相端输入 (CM0-)</u> 00 = <u>PD4</u> 10 = 运放输出 OP0OUT 01 = PC3 11 = DAC1OUT		CM0CON0[1:0]	RW-00	
CM1EN	比较器 C1	1 = 使能 0 = <u>禁止</u>	CM1CON0[7]	0x29E	RW-0
CM1POL	比较器 C1 输出极性	1 = 反向 0 = <u>正常</u>	CM1CON0[6]		RW-0
CM1OE	比较器 C1 输出到 PA6	1 = 使能 0 = <u>禁止</u>	CM1CON0[5]		RW-0
C1OUT	<u>比较器 C1 输出结果</u> 当 CM1POL = 0: 当 CM1POL = 1: 1 = CM1+ > CM1- 1 = CM1+ < CM1- 0 = <u>CM1+ < CM1-</u> 0 = <u>CM1+ > CM1-</u>		CM1CON0[4]		RO-x
CM1PSEL	<u>比较器 C1 正相端输入 (CM1+)</u> (00) = <u>PB3</u> (10) = PD5 (01) = PC0 (11) = (未连接)		CM1CON0[3:2]		RW-00
CM1NSEL	<u>比较器 C1 反相端输入 (CM1-)</u> 00 = <u>PB2</u> 10 = 运放 0 输出 01 = PB7 11 = DAC2OUT		CM1CON0[1:0]	0x29E	RW-00

名称	状态		寄存器	地址	复位值
CMDEBS	去抖选择	1 = 比较器 C1 去抖 0 = 比较器 C0 去抖	DACCON0[7]	0x29C	RW-0
DEBPR	<u>比较器 C0/C1 去抖时间</u> 00 = 不去抖 01 = 1~2 个去抖时钟 10 = 2~3 个去抖时钟 11 = 3~4 个去抖时钟 注：去抖时钟由 LVDCKS (参阅表 5-1) 选择；		DACCON0[6:5]		RW-00
WNDEN	窗口比较模式	1 = 使能 0 = 关闭	DACCON0[3]		RW-0
DACEN	DAC	1 = 使能 0 = 禁止	DACCON0[2]		RW-0
DACVREF	<u>DAC 参考电压</u> 00 = 2V 10 = 4V 01 = 3V 11 = VDD		DACCON0[1:0]		RW-00
DAC1DAT	<u>DAC1 通道输出电压</u> $V_{DAC1OUT} = V_{DACVREF} * (DAC1DAT[6:0] + 1) / 128$ (0x20 ≤ DAC1DAT [6:0] ≤ 0x7F)		DAC1DAT[6:0]	0x29A	RW-000 0000
DAC2DAT	<u>DAC2 通道输出电压</u> $V_{DAC2OUT} = V_{DACVREF} * (DAC2DAT[6:0]) / 128$ (0x00 ≤ DAC2DAT [6:0] ≤ 0x58)		DAC2DAT[6:0]	0x29B	RW-000 0000
CMAUSTR	<u>比较器 C0/C1 的自动开启</u> 1 = 使能 0 = 禁止		MSCON2[5]	0x10D	RW-0
STRTYP	<u>比较器 C0/C1 自动开启类型</u> (仅 CMAUSTR = 1 时有效) 00 = LVD 中断事件置位 CM0EN 01 = LVD 中断事件置位 CM1EN 10 = SLEEP 模式, LVD 中断唤醒 CPU 后置位 CM0EN 11 = SLEEP 模式, LVD 中断唤醒 CPU 后置位 CM1EN		MSCON2[4:3]		RW-00

表 14-1 比较器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, C0IE, C1IE 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 (C0IE, C1IE 适用) 0 = 关闭 (无唤醒)		INTCON[6]		RW-0
C1IE	比较器 C1 中断	1 = 使能 0 = 关闭 (无唤醒)	PIE2[1]	0x89	RW-0
C0IE	比较器 C0 中断		PIE2[0]		RW-0
C1IF	比较器 C1 中断标志位	1 = Yes (锁存) 0 = No	PIR2[1]	0x09	RW-0
C0IF	比较器 C0 中断标志位		PIR2[0]		RW-0

表 14-2 比较器中断使能和状态位

名称	地址	bit 7	bit 6	bit 5	bit 4	bit 3	Bit 2	bit 1	bit 0	复位值
DAC1DAT	0x29A	-	DAC1DAT[6:0]						-000 0000	
DAC2DAT	0x29B	-	DAC2DAT[6:0]						-000 0000	
DACCON0	0x29C	CMDEBS	DEBPR[1:0]		-	WNDEN	DACEN	DACVREF[1:0]		000- 0000
CM0CON0	0x29D	CM0EN	CM0POL	CM0OE	C0OUT	CM0PSEL[1:0]		CM0NSEL[1:0]		000x 0000
CM1CON0	0x29E	CM1EN	CM1POL	CM1OE	C1OUT	CM1PSEL[1:0]		CM1NSEL[1:0]		000x 0000
MSCON2	0x10D	P0ANP	P0AP	CMAUSTR	STRTYP[1:0]		T1EVTS[2:0]		0000 0000	

表 14-3 比较器相关寄存器汇总

14.2. 比较器配置

模拟输入端口 — 比较器输入可选择外部管脚、内部 DAC 或运放输出 OP0OUT (参阅 “CMxPSEL” 和 “CMxNSEL”)。当选择外部管脚时, 相应的 I/O 需设置 TRISx = 1 和 ANSELx = 1 来配置成模拟输入。

除了配置成单比较器以外, 比较器 C0/C1 还可配置成窗口比较器 (WNDEN = 1, CM0OE = 1 且 CM1OE = 1)。

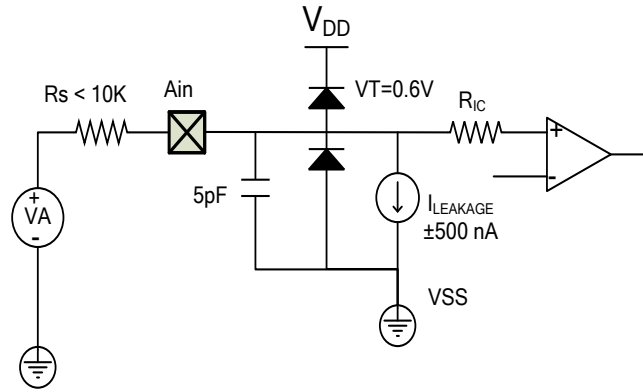


图 14-2 比较器模拟输入模型

模拟输入端口与 I/O 引脚复用，这些引脚与 VDD 和 GND 之间都有反向保护二极管，输入电压禁止偏离此保护范围 0.6V 以上，否则二极管将导通导致门锁效应。外部串联电阻需 < 10 kΩ。此外，如果模拟输入引脚上外接电容或齐纳二极管之类的元件，不应产生泄漏电流，否则可能会造成结果的不准确。

比较结果输出 — 比较器 C0/C1 的比较结果可分别输出到 I/O 引脚 PA7 和 PA6 (CMxOE = 1)，输出极性可选 (参阅 CMxPOL)，相应的 I/O 需设置 TRISx = 0 来使能输出驱动电路。

此外，比较器 C0/C1 具有去抖功能 (参阅 CMDEBS)，去抖时钟与 LVD 的去抖时钟相同 (参阅 LVDCKS, 表 5-1)，且有 3 种去抖时间可选(参阅 DEBPR)。

注：

1. 硬件只实现了比较结果的单边去抖功能，即当 CxPOL= 0 时，只对高脉冲毛刺过滤，而当 CxPOL = 1 时，只对低脉冲毛刺过滤。
2. SLEEP 模式下，需设置 SYSON = 1，去抖功能才有效。

比较器反应时间 — 当输入新的参考电压或输入源变化时，比较器输出有效电平所需的时间，即为反应时间。此外，当参考电压变化时，还需要一定的稳定时间。具体请参阅 章节 23.10。

注：使能比较器模块时，需要 1μs 左右的稳定时间，在此期间，比较器输出无效，应关闭中断以避免误触发。

可编程参考电压 DACxOUT — 比较器 C0/C1 的反相输入端可分别选择为内部可编程参考电压 $V_{DAC1OUT}$ 和 $V_{DAC2OUT}$ (CMxNSEL = 11)，参考电压由 DAC 模块提供且独立使能(参阅 "DACEN")，可输出 128 级电平范围，并与 DAC 的参考电压 (参阅 DACVREF) 成比例。

$$V_{DAC1OUT} = V_{DACVREF} * (DAC1DAT[6:0] + 1)/128 \quad (0x20 \leq DAC1DAT[6:0] \leq 0x7F)$$

$$V_{DAC2OUT} = V_{DACVREF} * (DAC2DAT[6:0]) /128 \quad (0x00 \leq DAC2DAT[6:0] \leq 0x58)$$

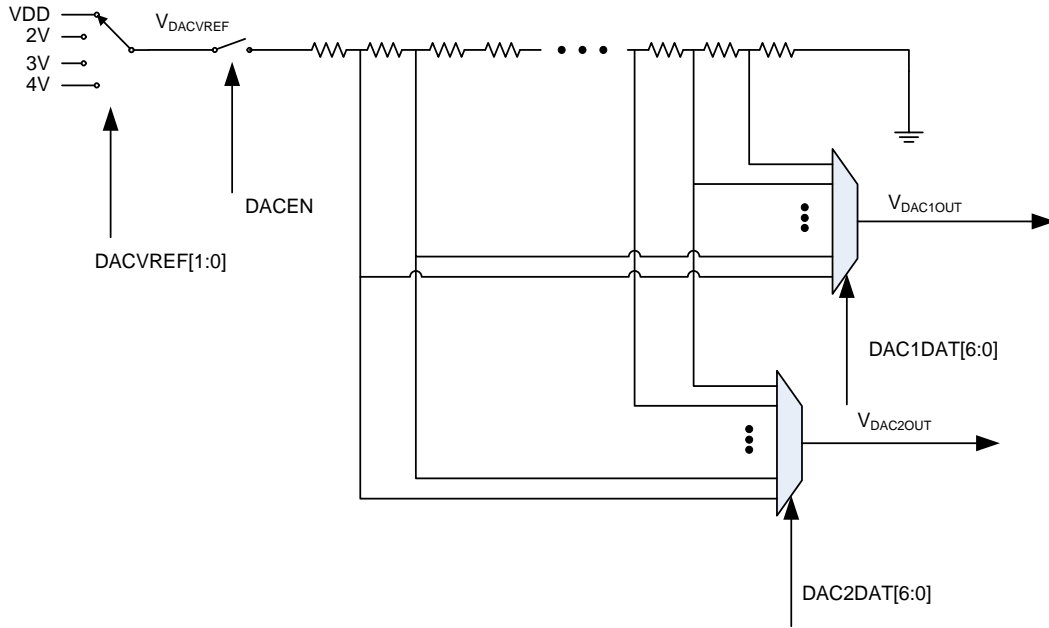


图 14-3 比较器参考电压框图

比较器的自动开启 — 比较器 C0/C1 可以和 LVD 配合使用，达到级联启动，从而节省功耗。在某些应用中，可先使能低速低功耗的 LVD，待 LVD 中断事件发生后，自动使能高速高功耗的比较器（参阅“CMAUSTR”和“STRTYP”）。

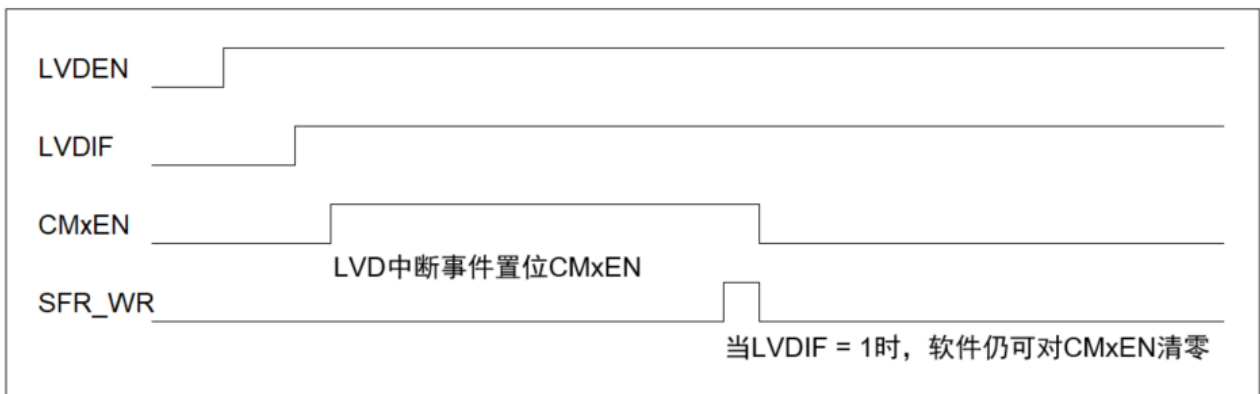


图 14-4 比较器 CMx 的自动开启

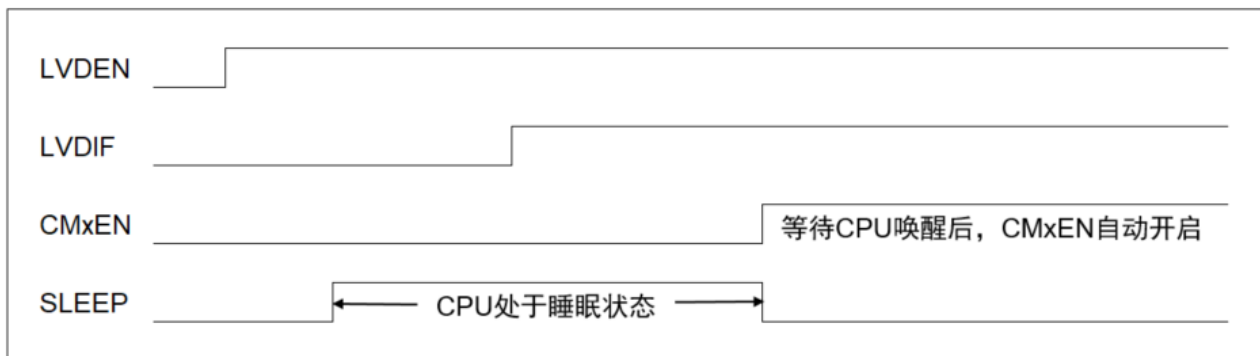


图 14-5 比较器 CMx 在睡眠状态下的自动开启

14.3. 比较器中断

比较器 C0/C1 输出变化时 (当 $CMxPOL = 0$ 时: 由低变到高; 当 $CMxPOL = 1$ 时: 由高变到低), 将置位相应的中断标志位 $CxIF$ 。是否触发中断和/或从睡眠中唤醒则取决于相应的使能控制位(GIE, PEIE, C1IE, C2IE)。

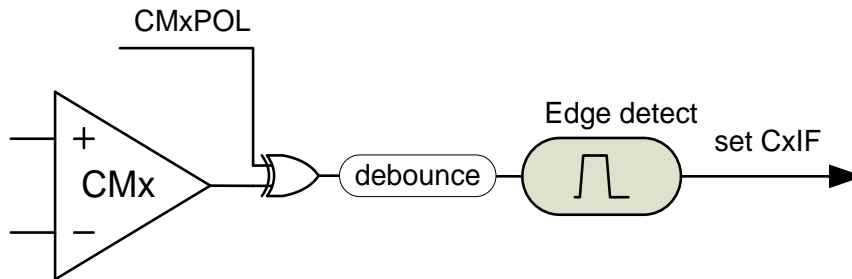


图 14-6 比较器中断检测原理图

15. SPI 接口

SPI 接口可通过 SPI 协议与外部设备进行通信，特性如下：

- 全双工、半双工同步传输
- 主机模式、从机模式
- 可编程主机模式通信速率
- 可编程时钟极性和相位
- 可编程数据传输格式：优先发送 LSB 或 MSB
- 主机和从机模式下均可由硬件或软件管理 NSS 引脚：主/从模式的动态切换
- 支持 SPI 接口 MOSI / MISO 开漏输出
- 发送 BUF 为空中断、接收 BUF 为非空中断
- 工作模式错误中断、接收溢出中断
- 从机模式唤醒中断

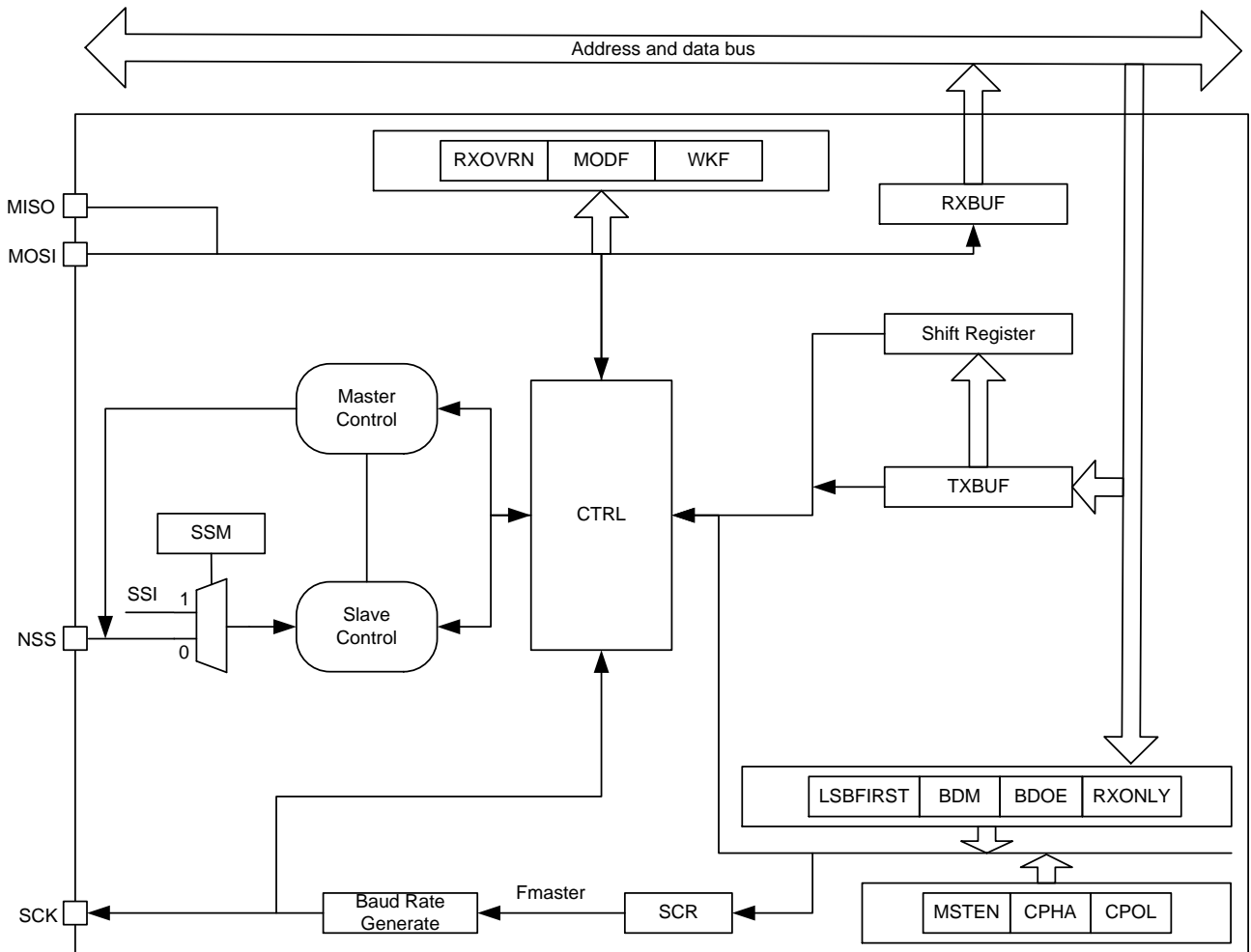


图 15-1 SPI 结构框图

SPI 接口有 4 个引脚:

名称	功能	主机模式	从机模式
MOSI	主机输出 / 从机输入	数据发送	数据接收
MISO	主机输入 / 从机输出	数据接收	数据发送
SCK	串行时钟	时钟输出	时钟输入
NSS	从机片选脚	-	输入, 低电平有效

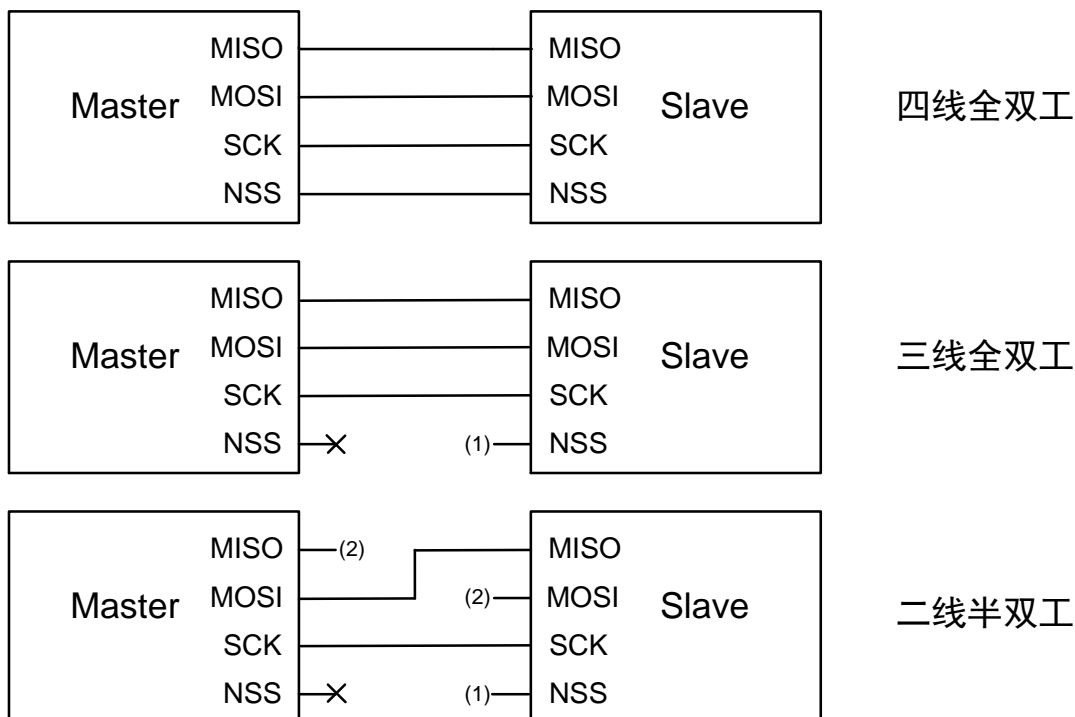
表 15-1 SPI 接口引脚说明

注:

1. 本章节中的 MOSI / MISO / SCK / NSS 分别对应引脚图中的 SPI_MOSI / SPI_MISO / SPI_SCK / SPI_NSS。
2. 从机片选 NSS 引脚配置:
 - NSS 引脚可配置成输入、输出或禁用三种状态 (参阅 "NSSM");
 - NSS 用作输入时, 其输入值 NSSVAL 为端口电平值(硬件) 或 SSI 值(软件, 参阅 "SSM");
 - 从机模式下, 当 NSS 配置成输入且为低电平时, 表示该从机被选中, 可以开始接收或发送数据;
 - 主机模式下, 当 NSS 配置成输入且为低电平时, 将导致工作模式错误(置位 MODF), 此时 SPI 模块自动切换成从机模式, 此特性可用于兼容多主机通信;

SPI 接口支持全双工(四线/三线)和半双工(二线)同步数据传输。SPI 通信总是由主机发起。

全双工模式, 在同一时钟信号 (主机输出的串行时钟) 下, 数据输出和数据输入同步进行。半双工模式, 主机模式的数据脚为 MOSI, 从机模式的数据脚为 MISO。



注:
 (1) 硬件或软件管理;
 (2) GPIO;

图 15-2 SPI 接口引脚连接示意图

15.1 SPI 相关寄存器汇总

名称	状态	寄存器	地址	复位值
SPIDAT	<u>数据发送/接收 BUF (TXBUF/RXBUF)</u> 写时: 将新数据写入到 TXBUF 中 读时: 返回 RXBUF 中未读的数据	SPIDAT[7:0]	0x18C	RW-0000 0000
SPIF ¹	<u>数据传输完成标志</u> 1 = 完成 (锁存) 0 = 未完成, 或已被清零	SPISTAT[7]	0x195	RW0-0
WCOL ¹	<u>BUF 写入失败(非空时写入)标志</u> 1 = 失败 (锁存) 0 = 正常	SPISTAT[6]		RW0-0
NSSM	<u>NSS 引脚模式选择</u> 00 = 禁用 01 = 输入 (输入值 NSSVAL 与 SSM, 端口电平及 SSI 有关) 1x = 输出 (输出值 = NSSM[0])	SPICR0[3:2]	0x18D	RW-01
SPIEN	<u>SPI 接口</u> 1 = 使能 0 = 关闭	SPICR0[0]		RW-0
SPIBUSY	<u>SPI 状态</u> 1 = 忙碌中 0 = 空闲	SPICFG[7]	0x18E	RO-x
MSTEN	<u>工作模式</u> 1 = 主机模式 (MASTER) 0 = 从机模式 (SLAVE)	SPICFG[6]		RW-x
CPHA	<u>SCK 相位选择 (数据采样点)</u> 1 = 第 2 个时钟转换沿 0 = 第 1 个时钟转换沿 注: 在从机发送模式下, 不管 CPHA 为何值, 数据都是在第一个沿开始变化。见图 15-3、图 15-4	SPICFG[5]		RW-x
CPOL	<u>SCK 极性选择 (SPI 空闲时, SCK 时钟状态)</u> 1 = 高电平 0 = 低电平	SPICFG[4]		RW-x
SLAS	<u>从机 (SLAVE) 选中标志位</u> 1 = 被选中 0 = 未被选中	SPICFG[3]		RO-x

¹ 写 0 清零, 写 1 无效;

名称	状态	寄存器	地址	复位值
NSSVAL	<u>NSS 引脚输入值</u> 当 SSM=0 时, NSSVAL= NSS 引脚端口电平值 当 SSM=1 时, NSSVAL= SSI	SPICFG[2]		RO-x
SRMT	<u>内部串行移位寄存器状态</u> 1 = 空 0 = 非空	SPICFG[1]		RO-x
SYSON	<u>睡眠模式下, 系统时钟控制</u> 1 = 保持运行 0 = 关闭	LVDCON[7]	0x21A	RW-0
SPISCR	<u>SCK 速率设置 (仅主机模式有效)</u> 速率 = Fmaster/(2*(SPISCR+1)) (SPI 外设时钟 Fmaster = Sysclk)	SPISCR[7:0]	0x18F	RW-xxxx xxxx
BDM	<u>半双工</u> 1 = 使能 0 = 关闭	SPICR1[7]	0x194	RW-x
BDOE	<u>半双工工作模式</u> 1 = 发送 0 = 接收	SPICR1[6]		RW-x
RXONLY	<u>全双工工作模式</u> 1 = 只允许接收 0 = 允许发送和接收	SPICR1[5]		RW-x
SSI	<u>NSS 引脚软件输入值 (仅当 SSM = 1 时有效)</u> 1 = 输入值为 1 0 = 输入值为 0	SPICR1[4]		RW-x
SSM	<u>从机模式下, NSS 引脚输入值管理</u> 1 = 软件 0 = 硬件	SPICR1[3]		RW-x
LSBFIRST	<u>数据传输格式</u> 1 = 优先发送低比特位 (LSB) 0 = 优先发送高比特位 (MSB)	SPICR1[0]		RW-x

表 15-2 SPI 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, TXE, RXNE, RXERR, WAKUP 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 (TXE, RXNE, RXERR, WAKUP 适用)	INTCON[6]		RW-0

	0 = 关闭 (无唤醒)				
TXE	发送 BUF 为空中断	1 = 使能 0 = 关闭 (无唤醒)	SPIIER[0]	0x193	RW-x
TXBMT	发送 BUF 状态位	1 = 空 0 = 非空	SPICR0[1]	0x18D	RO-1
RXNE	接收 BUF 为非空中断	1 = 使能 0 = 关闭 (无唤醒)	SPIIER[1]	0x193	RW-x
RXBMT	接收 BUF 状态位	1 = 空 0 = 非空	SPICFG[0]	0x18E	RO-x
RXERR	接收错误中断 (工作模式错误, 接收溢出)	1 = 使能 0 = 关闭 (无唤醒)	SPIIER[2]	0x193	RW-x
MODF ²	工作模式错误标志位	1 = 错误 (锁存) (主机模式下, NSS 脚使能输入且为低电平, 导致模式错误) 0 = 正常	SPISTAT[5]	0x195	RW0-0
RXOVRN ²	接收溢出标志位	1 = 溢出 (锁存) 0 = 正常	SPISTAT[4]		RW0-0
WAKUP	从机唤醒中断	1 = 使能 0 = 关闭	SPIIER[3]	0x193	RW-x
WKF ²	从机唤醒(接收到数据)标志位	1 = 已唤醒 (锁存) 0 = 未唤醒, 或已被清零	SPISTAT[1]	0x195	RW0-0

表 15-3 SPI 中断使能和状态位

名称	状态		寄存器	地址	复位值
SCKPO	SPI_SCK	1 = PD4 0 = <u>PC6</u>	COMAF0[7]	0x285	RW-0
NSSPO	SPI_NSS	1 = PC6 0 = <u>PC5</u>	COMAF0[6]		RW-0
MOSIPO	SPI_MOSI	1 = PC0 0 = <u>PD5</u>	COMAF0[5]		RW-0
MISOPO	SPI_MISO	1 = PB4 0 = <u>PC7</u>	COMAF0[4]		RW-0

² 写 0 清零, 写 1 无效;

ODCONA	PORTA 开漏	1 = 使能 0 = 关闭	ODCONA[7:0]	0x205	RW-0000 0000
ODCONB	PORTB 开漏		ODCONB[7:0]	0x206	RW-0000 0000
ODCONC	PORTC 开漏		ODCONC[7:0]	0x207	RW-0000 0000
ODCOND	PORTD 开漏		ODCOND[5:0]	0x208	RW--00 0000

表 15-4 SPI 接口引脚控制

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
SPIDAT	0x18C	数据发送/接收寄存器 (BUF)SPIDAT [7:0]								0000 0000
SPISTAT	0x195	SPIF	WCOL	MODF	RXOVRN	-	-	WKF	-	0000 --0-
SPICR0	0x18D	-	-	-	-	NSSM		TXBMT	SPIEN	---- 0110
SPICFG ³	0x18E	SPIBUSY	MSTEN	CPHA	CPOL	SLAS	NSSVAL	SRMT	RXBMT	xxxx xxxx
SPICR1 ³	0x194	BDM	BDOE	RXONLY	SSI	SSM	-	-	LSBFIRST	xxxx x--x
SPIIER ³	0x193	-	-	-	-	WAKUP	RXERR	RXNE	TXE	---- xxxx
COMAF0	0x285	SCKPO	NSSPO	MOSIPO	MISOPO	SDAPO[1:0]		SCLPO[1:0]		0000 0000

表 15-5 SPI 相关寄存器地址

³ 寄存器复位值不确定，应用程序在系统初始化时应将其写为合适的值，特别是各中断使能位，以免产生中断及不必要的功耗。

15.2 SPI 配置

主机和从机的 SPI 配置流程基本相同：

1. 选择主机或从机模式 (参阅 "MSTEN");
2. 配置 NSS 引脚 (参阅 "NSSM", "SSM", "SSI" 和 "NSSVAL");
3. 配置主机模式的 SCK 通信速率 = $F_{\text{master}} / (2 * (\text{SPISCR} + 1))$, 从机模式的速率高达 $F_{\text{master}} / 4$;
4. 设置 SCK 的相位和极性 (参阅 "CPOL" 和 "CPHA");
5. 选择数据传输格式 (参阅 "LSBFIRST");
6. 设置全双工 (参阅 "RXONLY") 或半双工工作模式 (参阅 "BDM" 和 "BDOE");
7. 设置 SPIEN = 1, 使能 SPI 模块;
8. 如需要, 可使能相应的中断 (参阅 "GIE", "PEIE", "RXERR", "RXNE", "TXE" 和 "WAKUP");

注:

- SPI 外设时钟 $F_{\text{master}} = \text{Sysclk}$;
- SPI 模块使能时, 引脚 MOSI / MISO / SCK / NSS 接口功能自动使能;
- 主机发送 SCK 时钟之前, 需要先使能 SPI 从机;
- 当主机作为发送端, SPI 使能且 TXBUF 为非空时, 主机自动发起传输;
- 当主机作为只接收模式 (RXONLY=1 或 BDM=1 & BDOE=0) 时, SPI 使能后, 主机自动发起传输并一直发送 SCK;
- 主机发起传输之前, 从机的数据寄存器中需提前写入将要发送的数据 (连续通信时, 在正在进行的传输结束之前, 需继续向从机的数据寄存器中写入数据);
- 当 SPIEN 由 0 变 1 时, SPIF / MODF / RXOVRN / WKF 自动清零, TXBMT / RXBMT 自动置位;

15.2.1 通信时钟 SCK

时钟 SCK 的极性和相位可配置为图 15-3、图 15-4 所示的 4 种情况 (参阅 "CPOL", "CPHA")。

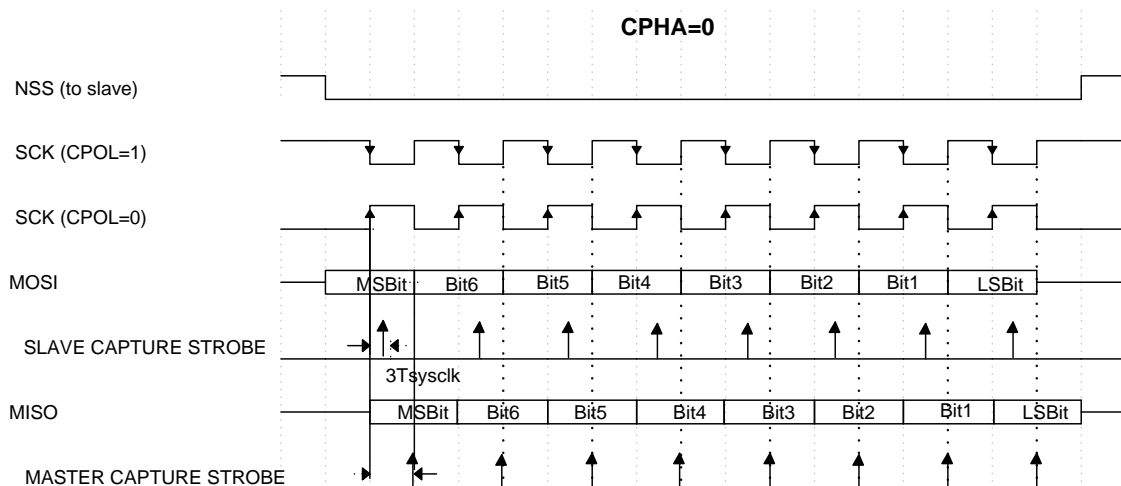


图 15-3 时钟极性 CPHA=0 时序图

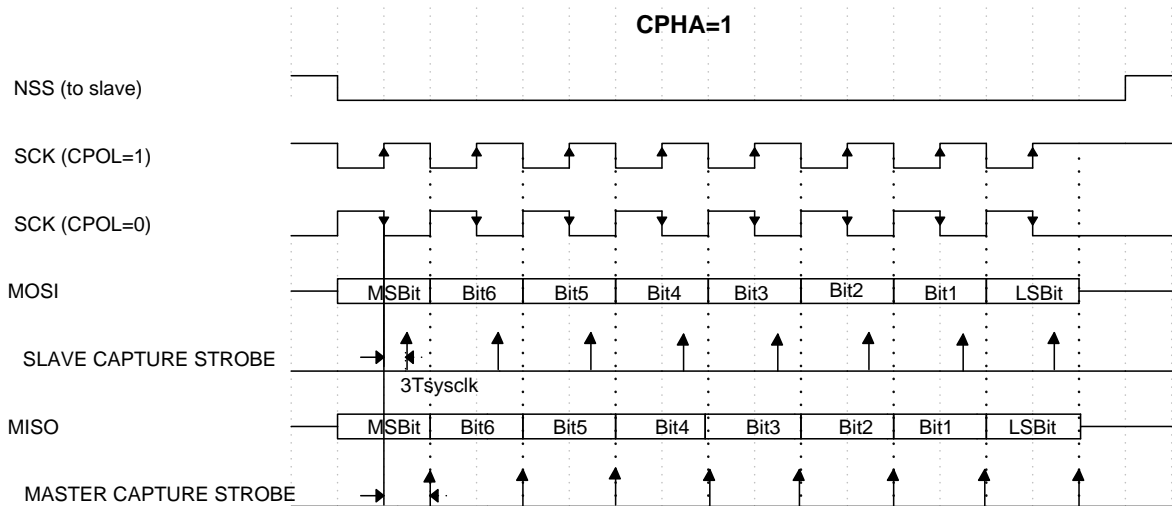


图 15-4 时钟极性 CPHA=1 时序图

15.2.2 数据处理流程

数据通信流程分为阻塞模式和非阻塞模式。

	阻塞模式	非阻塞模式
发送数据	向 SPIDAT (TXBUF) 写入数据后，查询 TXBMT，当其置 1 时，写入下一个数据	当 TXE = 1 时，向 SPIDAT (TXBUF) 写入数据后，TXBMT 置 1 则进入中断
接收数据	查询 RXBMT，当其为 0 时，则可读取 DATA(RXBUF) 的值	当 RXNE = 1 时，RXBMT 复位为 0 后则进入中断
	查询 RXOVRN，当 RXOVRN 置 1 时，需软件清零相应的错误标志位	当 RXERR = 1 时，RXOVRN 置 1 后进入中断 (需软件清零相应的错误标志位)
备注	-	进入中断后，查询相应的状态标志位并处理发送接收流程，处理完成后退出中断

表 15-6 SPI 数据处理流程

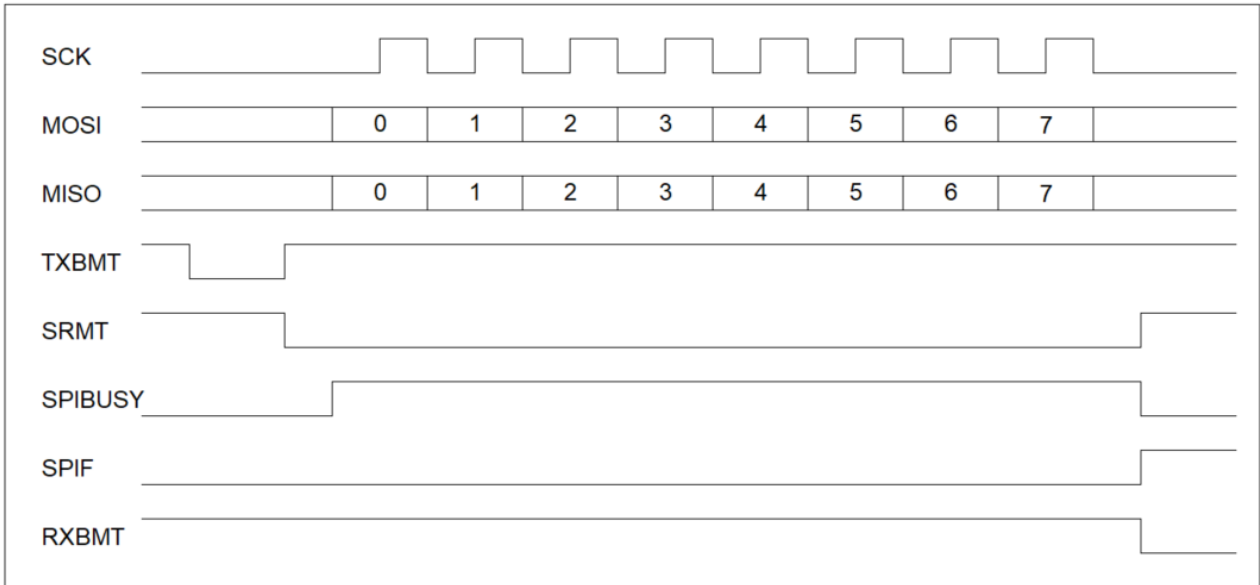


图 15-5 数据处理时序图 (以单字节数据传输为例)

以全双工通信流程为例，无论阻塞模式还是非阻塞模式，通信过程中的相关标志位变化如图 15-5 所示：

1. 向 SPIDAT (TXBUF) 寄存器写入数据后，TXBMT 由 1 变为 0；
2. TXBUF 中的数据传送到内部移位寄存器，SRMT 由 1 变为 0，SPIBUSY 置 1；
3. 移位寄存器中的数据完全移出后，SRMT 由 0 变为 1，SPIBUSY 清零；
4. 当前字节数据传输完成后，SPIF 由 0 变为 1，同时 RXBMT 由 1 变为 0，此时可读取 SPIDAT (RXBUF) 寄存器中的值；

注：全双工或半双工模式下，需在完成发送/接收全部数据 (TXBMT=1 / RXBMT=0) 后，且 SPI 处于空闲状态 (SPIBUSY=0) 时，才能关闭 SPI 模块；

15.2.3 从机模式的睡眠唤醒

睡眠模式下，如果 SYSON、WAKEUP、PEIE 同时使能，从机在接收到数据的第 1 个 bit 时，即可唤醒 MCU。

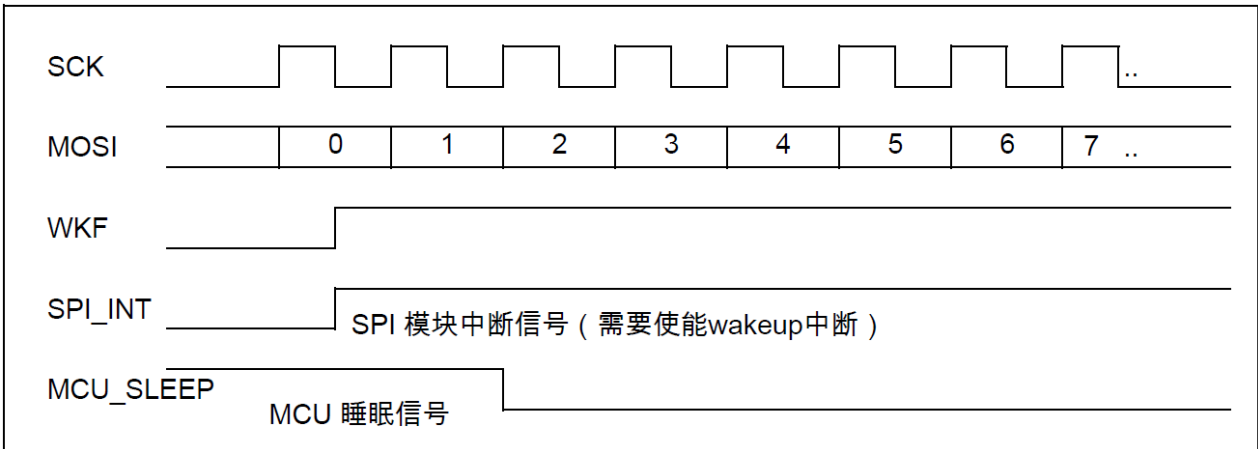


图 15-6 睡眠唤醒时序图

16. I2C 接口

I2C 为双线接口 (数据线 SDA 和串行时钟线 SCL), 可通过 I2C 协议与外部设备进行通信, 特性如下:

- 主机模式、从机模式
- 多主机兼容
- 标准模式(100 kHz)、快速模式(400 kHz)、快速+模式(1 MHz)
- 7 位或 10 位地址格式
- 数据从高位开始发送/接收
- 可选时钟拉低扩展 (Clock stretching)
- 支持 I2C 接口 SCL / SDA 开漏输出
- 支持软件复位
- 事件中断:
 - ✓ TX-FIFO 状态为空中断、RX-FIFO 状态为非空中断
 - ✓ 主机模式下: SCL 拉伸中断、发送 Stop 中断、目标从机地址匹配中断
 - ✓ 从机模式下: SCL 拉伸中断、检测到 Stop 中断、本机地址匹配中断
- 错误中断:
 - ✓ 主机仲裁失败中断

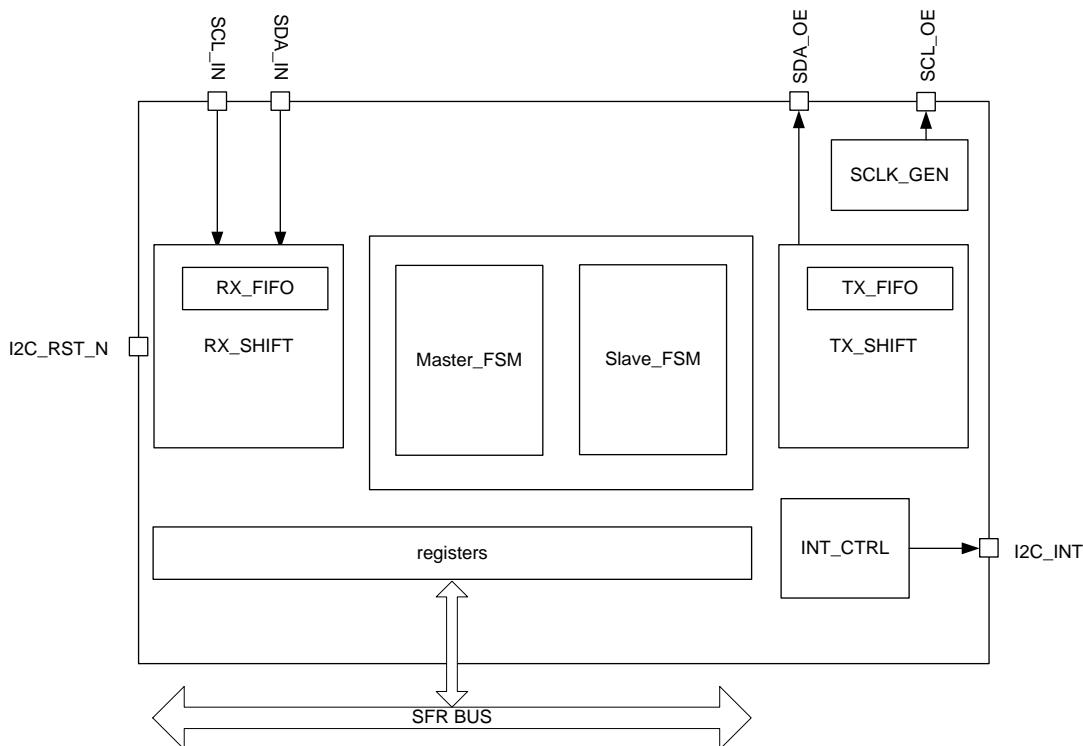


图 16-1 I2C 结构框图

16.1 I2C 接口相关寄存器汇总

名称	状态	寄存器	地址	复位值
SYSON	<u>睡眠模式下，系统时钟控制</u> 1 = 保持运行 0 = 关闭	LVDCON[7]	0x21A	RW-0
STRET	<u>当 TX-FIFO 为空 / RX-FIFO 非空 / 地址匹配时，拉伸 SCL</u> 1 = 使能 0 = 关闭 注：如果主/从机关闭拉伸且为发送模式，当发送缓存 TX-FIFO 为空时，则重复发送最后一次写入到 I2CDAT 的数据；	I2CCR1[7]	0x196	RW-x
ADDRM ¹	<u>地址格式</u> 1 = 10 位 0 = 7 位	I2CCR1[5]		RW-x
STOP	<u>字节传输后，发送 Stop</u> 1 = 发送 (发送成功后自动清零) 0 = 不发送	I2CCR1[4]		RW-x
WRN	<u>主机模式，数据传输方向 (读写位 R/W)</u> 1 = 读取 0 = 发送	I2CCR1[3]		RW-x
START	<u>发送 Start，或字节传输后发送 Restart</u> 1 = 发送 (发送成功后自动清零) 0 = 不发送	I2CCR1[2]		RW-x
MASTER ¹	<u>工作模式</u> 1 = 主机模式 0 = 从机模式	I2CCR1[1]		RW-x
I2CEN	<u>I2C 接口</u> 1 = 使能 0 = 关闭	I2CCR1[0]		RW-x
I2CADDRL	<u>从机地址低有效位 (LSB)</u> 7 位地址：ADD[6:0]有效，ADD[7]忽略； 10 位地址：ADD[7:0] = 低 8 位； 注：主机模式下为目标从机地址，从机模式下为本机地址；	I2CADDR[7:0]		0x198

¹ 当 I2CEN = 0 时可写；

名称	状态	寄存器	地址	复位值
I2CADDRH	<u>从机地址高有效位 (MSB)</u> 7 位地址: ADD[9:8]忽略; 10 位地址: ADD[9:8] = 高 2 位; 注: 主机模式下为目标从机地址, 从机模式下为本机地址;	I2CCR2[7:6]	0x197	RW-xx
NACK	<u>接收应答</u> 1 = 发送 NACK 0 = <u>发送 ACK (地址匹配或接收到数据)</u>	I2CCR2[5]		RW-x
DUTY	<u>SCL 占空比</u> 1: SCLL / SCLH = 2 / 1 0: SCLL / SCLH = 1 / 1	I2CCR2[4]		RW-x
I2CCCR	<u>I2C 通信速率</u> 通信速率 = Sysclk / (2x (I2CCCR + 3)) 注: I2CCCR 最小值为 1;	I2CCCR[7:0]	0x199	RW-xxxx xxxx
I2CDAT	<u>数据寄存器</u> 写时: 将新数据写入到 TX-FIFO 中 读时: 返回 RX-FIFO 中未读的数据 注: TX-FIFO 和 RX-FIFO 的深度均为 1;	I2CDAT[7:0]	0x19A	RW-xxxx xxxx

表 16-1 I2C 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, I2CIE 适用) 0 = <u>全局关闭</u> (唤醒不受影响)	INTCON[7]	Bank 首 地址 +0x0B	RW-0
PEIE	外设总中断 1 = 使能 (I2CIE 适用) 0 = <u>关闭</u> (无唤醒)	INTCON[6]		RW-0
I2CIE	I2C 接口中断 1 = 使能 0 = <u>关闭</u> 注: I2C 接口中断产生的条件 HOLDF = 1 STOPF = 1 ADDRF = 1 BUSERR = 1 IICRXNEF = 1	I2CCR2 [0]	0x197	RW-x

名称	状态	寄存器	地址	复位值	
	IICTXEF = 1				
HOLDF	主机模式, SCL 拉伸状态	1 = SCL 正被主机拉伸 0 = SCL 未被主机拉伸	I2CISR[7]	0x19E	RO-0
	从机模式, SCL 拉伸状态 ²	1 = SCL 正被从机拉伸 0 = SCL 未被从机拉伸, 或已被清零			RW0-0
STOPF ²	主/从机模式, Stop 标志	1 = 检测到 (ACK 后置位) 0 = 未检测到	I2CISR[6]		RW0-0
ADDRF ³	主机发送地址 / 从机接收地址匹配标志	主机发送地址: 1 = 完成 (ACK 后置位) 0 = 未发送或不匹配 从机接收地址: 1 = 匹配 0 = 不匹配 注: NACK 后不会置位 ADDRF	I2CISR[5]		RW0-0
NACKF ²	应答 ACK 状态	1 = NACK 0 = ACK	I2CISR[4]		RW0-0
I2CBUSY	主/从机状态	1 = Busy (繁忙) 0 = IDLE (空闲) 注: 从机模式, 地址匹配成功后即置位, 接收到 Start / Restart / Stop 后清零;	I2CISR[3]	0x19E	RO-0
BUSERR / DIRF	主机模式, 总线仲裁失败 (BUSERR) 标志	1 = 产生仲裁失败 0 = 未产生仲裁失败	I2CISR[2]		RO-0
	从机模式, 数据传输方向 (DIRF) 标志 ²	1 = 发送 (从机接收地址字节的读写位为 1 时置位) 0 = 接收		RW0-0	
IICRXNEF ²	RX-FIFO 状态	1 = 非空 (读 I2CDAT 或写 0 清零) 0 = 空	I2CISR[1]		RW0-0
IICTXEF ⁴	TX-FIFO 状态	1 = 空 (写 I2CDAT 或写 1 清零) 0 = 非空	I2CISR[0]		RW1-1

表 16-2 I2C 中断使能和状态位

² 写 0 清零, 写 1 无效;

³ 10 位地址格式, 低 8 位地址和高 2 位地址匹配时均会置位; 写 0 清零, 写 1 无效;

⁴ 只能写 1, 写 0 无效;

名称	状态		寄存器	地址	复位值
SDAPO	<u>I2C_SDA 管脚映射</u>		COMAF0[3:2]	0x285	RW-00
	00 = <u>PC0</u> 01 = <u>PB3</u>	10 = <u>PC2</u> 11 = <u>PB4</u>			
SCLPO	<u>I2C_SCL 管脚映射</u>		COMAF0[1:0]		RW-00
	00 = <u>PB7</u> 01 = <u>PB2</u>	10 = <u>PC1</u> 11 = <u>PD4</u>			
ODCONA	PORTA 开漏	1 = <u>使能</u> 0 = <u>关闭</u>	ODCONA[7:0]	0x205	RW-0000 0000
ODCONB	PORTB 开漏		ODCONB[7:0]	0x206	RW-0000 0000
ODCONC	PORTC 开漏		ODCONC[7:0]	0x207	RW-0000 0000
ODCOND	PORTD 开漏		ODCOND[5:0]	0x208	RW-00 0000

表 16-3 I2C 接口引脚控制

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
I2CCR1	0x196	STRET	—	ADDRM	STOP	WRN	START	MASTER	I2CEN	x-xx xxxx
I2CCR2	0x197	I2CADDRH[1:0]		NACK	DUTY	—	—	—	I2CIE	xxxx ---x
I2CADDR	0x198	I2CADDRL[7:0]								x-xx xxxx
I2CCCR	0x199	I2CCCR[7:0]								xxxx xxxx
I2CDAT	0x19A	I2CDAT[7:0]								xxxx xxxx
I2CISR	0x19E	HOLDF	STOPF	ADDRF	NACKF	I2CBUSY	DIRF /BUSERR	IICRXNEF	IICTXEF	0000 0001

表 16-4 I2C 相关寄存器地址

16.2 I2C 配置

主机和从机的 I2C 配置流程基本相同：

1. 选择主机或从机模式 (参阅 "MASTER");
2. 主机配置 SCL 占空比及通信速率 (参阅 "DUTY", "I2CCCR[7:0]");
3. 主从机选择 7 位或 10 位地址格式 (参阅 "ADDRM");
4. 设置主机的数据传输方向为发送或接收 (参阅 "WRN"), 从机则由接收地址字节的读写位控制;
5. 设置 I2CEN = 1, 使能 I2C 模块;
6. 如需要, 可使能相应的中断 (参阅 "GIE", "PEIE" 和 "I2CIE");

注:

- 当 I2CEN = 1 时, 引脚 SCL / SDA 接口功能自动使能, SCL / SDA 分别对应引脚图中的 I2C_SCL / I2C_SDA;

I2C 通信由主机产生时钟信号并发起数据传输，且由主机控制 Start 和 Stop 信号。串行数据传输以 Start 条件开始并以 Stop 条件结束。在一个字节 (8bit) 传输后的第 9 个时钟期间，接收端需回送一个应答位 (ACK) 给发送端。

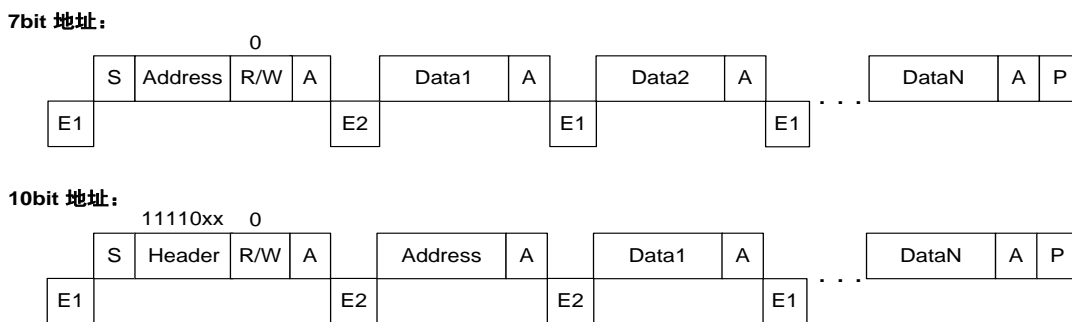
从机检测到 Start 条件后，能识别自己的地址 (可编程，7 位或 10 位)，且具有 Stop 检测功能。

I2C 模块的四种工作模式为：主机发送、主机接收、从机发送、从机接收。

16.2.1 主机发送模式

当 ADDR_M = 0 (7 位地址格式)：主机发送的第 1 个 byte 包括 7 位地址和读写位(0)，然后开始发送 8 位串行数据。

当 ADDR_M = 1 (10 位地址格式)：主机发送的第 1 个 byte 包括地址头段序列 (11110+高 2 位地址) 和读写位(0)，第 2 个 byte 为低 8 位地址，然后开始发送 8 位串行数据。



注：xx 为 Address 高有效位

图 16-2 主机发送流程

注：

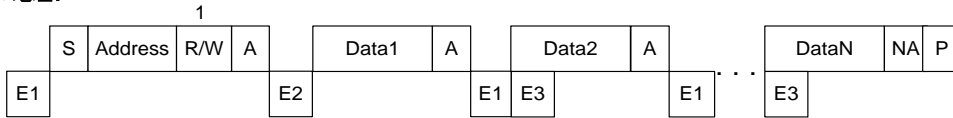
- S = Start 信号，A = ACK 信号，P = Stop 信号；
- E1: IICTXEF = 1, TX-FIFO 为空 (写 1 或写 I2CDAT 将清零该标志)；
- E2: ADDR_F=1 (写 0 清零)；

16.2.2 主机接收模式

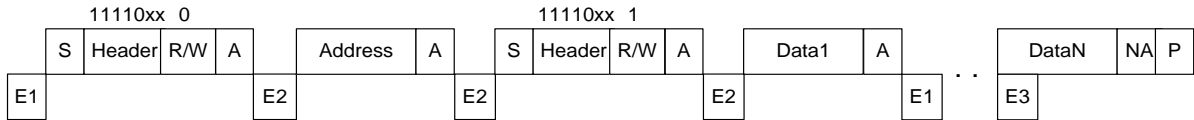
ADDR_M = 0 (7 位地址格式)：主机发送的第 1 个 byte 包括 7 位地址和读写位(1)，然后开始接收 8 位串行数据。

ADDR_M = 1 (10 位地址格式)：主机发送的第 1 个 byte 包括地址头段序列 (11110+高 2 位地址) 和读写位(0)，第 2 个 byte 为低 8 位地址，然后重新发送 Start 信号以及地址头段序列和读写位(1)，开始接收 8 位串行数据。

7bit 地址:



10bit 地址:



注: xx 为 Address 高有效位

图 16-3 主机接收流程

注:

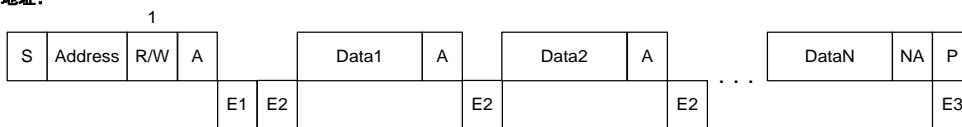
- S = Start 信号, A = ACK 信号, P = Stop 信号;
- E1: IICTXEF=1, TX-FIFO 为空 (写 1 或写 I2CDAT 将清零该标志);
- E2: ADDRDF=1 (写 0 清零);
- E3: IICRXNEF=1, RX-FIFO 非空 (写 0 或读 I2CDAT 将清零该标志);

16.2.3 从机发送模式

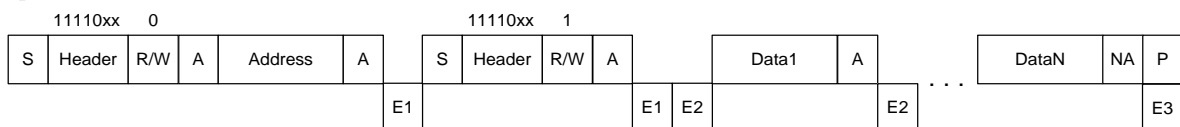
ADDRM = 0 (7 位地址格式): 从机接收的第 1 个 byte 包括 7 位地址和读写位(1), 然后开始发送 8 位串行数据。

ADDRM = 1 (10 位地址格式): 从机接收的第 1 个 byte 包括地址头段序列 (11110+高 2 位地址) 和读写位(0), 第 2 个 byte 为低 8 位地址, 然后重新检测 Start 信号并接收地址头段序列和读写位(1), 开始发送 8 位串行数据。

7bit 地址:



10bit 地址:



注: xx 为 Address 高有效位

图 16-4 从机发送流程

注:

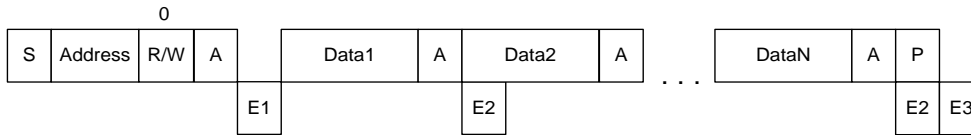
- S = Start 信号, A = ACK 信号, P = Stop 信号;
- E1: ADDRDF=1, 拉低 SCL 线 (写 0 清零);
- E2: IICTXEF=1, TX-FIFO 为空 (写 1 或写 I2CDAT 将清零该标志);
- E3: NACKF=1 (写 0 清零);

16.2.4 从机接收模式

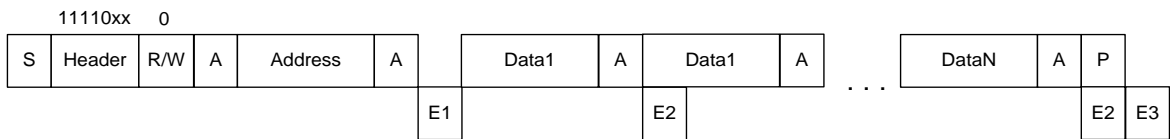
ADDRM = 0 (7 位地址格式): 从机接收的第 1 个 byte 包括地址和读写位(0), 然后开始接收 8 位串行数据。

ADDRM = 1 (10 位地址格式): 从机接收的第 1 个 byte 包括地址头段序列 (11110+高 2 位地址) 和读写位(0), 第 2 个 byte 为低 8 位地址, 然后开始接收 8 位串行数据。

7bit 地址:



10bit 地址:



注: xx 为 Address 高有效位

图 16-5 从机接收流程

注:

- S = Start 信号, A = ACK 信号, P = Stop 信号;
- E1: ADDRFR=1 (写 0 清零);
- E2: IICRXNEF=1, RX-FIFO 非空 (写 0 或读 I2CDAT 将清零该标志);
- E3: STOPF =1 (写 0 清零);

17. UARTx 接口

通用异步收发器 UARTx (x = 0, 1) 可与使用工业标准 NRZ 串行数据格式的外设进行通信。特性如下：

- 全双工、单线半双工异步模式
- 数据传输长度：8 或 9 位
- 奇偶校验位
- 停止位：1 或 2 位
- 16 位可编程波特率发生器，高达 1Mbit/s
- 数据从低位开始发送/接收
- 发送器和接收器可单独使能
- 支持 UART 接口 TX 开漏输出
- 发送 FIFO 为空中断、接收 FIFO 为非空中断
- 接收状态中断：帧错误、奇偶校验错误或接收溢出

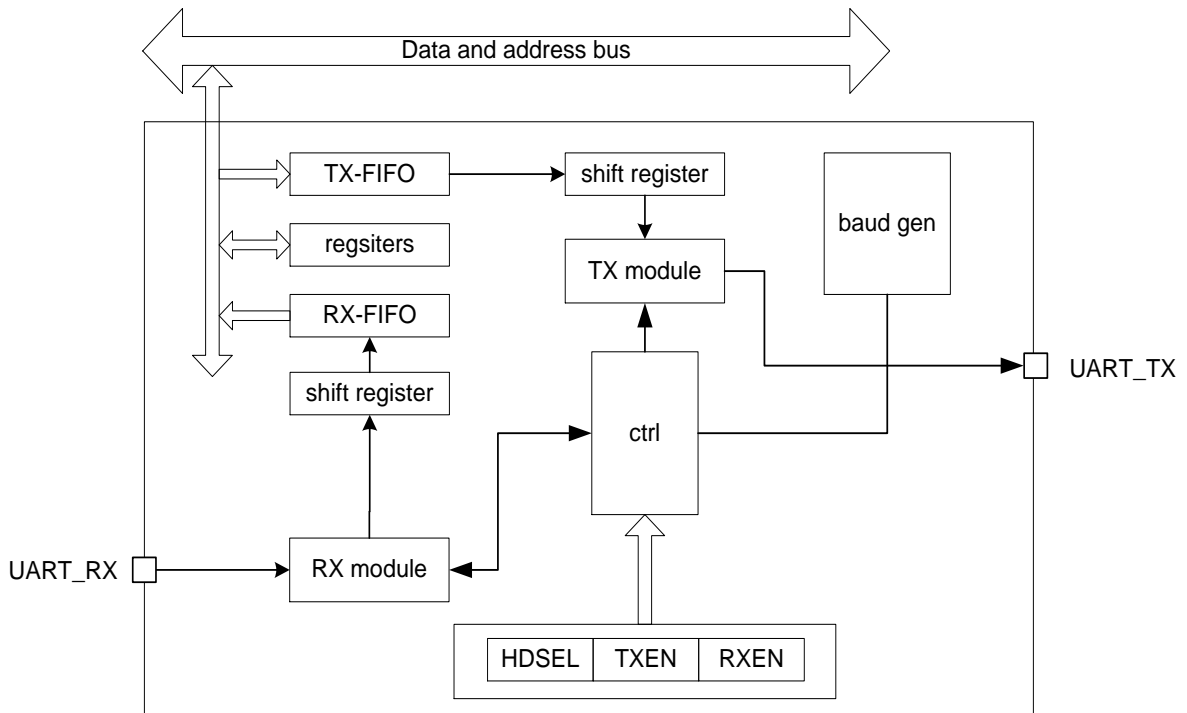


图 17-1 UART 结构框图

UART 串口模块有 2 个引脚：

UART_RX：串行数据输入。

UART_TX：串行数据输出。单线半双工模式下，TX 引脚既用于数据输入也用于数据输出（需配置成开漏模式）。

注：

- 当发送器使能，但不发送数据时，TX 引脚处于高电平。
- 当发送器使能，且发送数据时，TX 引脚在起始位期间处于低电平，在停止位期间处于高电平。

17.1 UARTx 接口相关寄存器汇总

名称	状态		寄存器	地址	复位值	
SYSON	睡眠模式下, 系统时钟控制	1 = 保持运行 0 = 关闭	LVDCON[7]	0x21A	RW-0	
UR0DATL	数据发送 / 接收 FIFO 低 8 位 (不适宜位操作)		UR0DATL[7:0]	0x185	RW-xxxx xxxx	
UR1DATL			UR1DATL[7:0]	0x90	RW-xxxx xxxx	
UR0DATH	数据发送 / 接收 FIFO 高 1 位 (当 URxMODE = 1 时) 注: 需先写 URxDATAL, 再写 URxDATAH;		UR0DATH[0]	0x186	RW-x	
UR1DATH			UR1DATH[0]	0x91	RW-x	
UR0TXBSY	发送器状态		UR0STAT[6]	0x19F	RO-x	
UR1TXBSY	1 = 忙碌	0 = 空闲	UR1STAT[6]	0x95	RO-x	
UR0RXFULL	接收 FIFO 状态		UR0STAT[3]	0x19F	RO-x	
UR1RXFULL	1 = 已满	0 = 未滿	UR1STAT[3]	0x95	RO-x	
UR0HDSSEL	半双工		UR0CR1[7]	0x187	RW-x	
UR1HDSSEL	1 = 使能	0 = 关闭	UR1CR1[7]	0x92	RW-x	
UR0STOP	停止位长度		UR0CR1[6]	0x187	RW-x	
UR1STOP	1 = 2 bit	0 = 1 bit	UR1CR1[6]	0x92	RW-x	
UR0MODE	通信数据格式 (不包括校验位)		UR0CR1[5]	0x187	RW-x	
UR1MODE	1 = 9 bit	0 = 8 bit	UR1CR1[5]	0x92	RW-x	
UR0EVEN	奇/偶校验		UR0CR1[4]	0x187	RW-x	
UR1EVEN	1 = 偶校验	0 = 奇校验	UR1CR1[4]	0x92	RW-x	
UR0PCEN	校验位		UR0CR1[3]	0x187	RW-x	
UR1PCEN	1 = 使能	0 = 关闭	UR1CR1[3]	0x92	RW-x	
UR0RXEN	串口接收		UR0CR1[2]	0x187	RW-x	
UR1RXEN	1 = 使能	0 = 关闭	UR1CR1[2]	0x92	RW-x	
UR0TXEN	串口发送		UR0CR1[1]	0x187	RW-x	
UR1TXEN	1 = 使能	0 = 关闭	UR1CR1[1]	0x92	RW-x	
UR0EN	串口模块		UR0CR1[0]	0x187	RW-x	
UR1EN	1 = 使能	0 = 关闭	UR1CR1[0]	0x92	RW-x	
UR0BRRH	波特率分频计数器低 8 位和高 8 位		高 4 位	UR0CR2[7:4]	0x188	RW-xxxx
UR0BRRL	波特率 = Fmaster / (16 * URxBRR) 注: Fmaster = Sysclk。URxBRR 最小值 = 0x0001, 当其为 0x0000 时, UARTx 不工作;		低 8 位	UR0BRRL[7:0]	0x189	RW-xxxx xxxx
UR1BRRH			高 4 位	UR1CR2[7:4]	0x93	RW-xxxx
UR1BRRL			低 8 位	UR1BRRL[7:0]	0x94	RW-xxxx xxxx

表 17-1 UART0 / UART1 相关用户寄存器

名称	状态		寄存器	地址	复位值
GIE	全局中断		INTCON[7]	Bank 首地址 +0x0B	RW-0
	1 = 使能 (PEIE, URxTXEIE, URxRXNEIE, URxERRIE 适用) 0 = 全局关闭 (唤醒不受影响)				
PEIE	外设总中断		INTCON[6]		RW-0
UR0TXEIE	发送 BUF 为空中断	1 = 使能	UR0CR2[0]	0x188	RW-x
UR1TXEIE		0 = 关闭	UR1CR2[0]	0x93	RW-x
UR0TXEF	发送 BUF 状态	1 = 空	UR0STAT[0]	0x19F	RO-x
UR1TXEF		0 = 非空	UR1STAT[0]	0x95	RO-x
UR0RXNEIE	接收 BUF 为非空中断	1 = 使能	UR0CR2[1]	0x188	RW-x
UR1RXNEIE		0 = 关闭	UR1CR2[1]	0x93	RW-x
UR0RXNEF	接收 BUF 状态	1 = 非空	UR0STAT[1]	0x19F	RO-x
UR1RXNEF		0 = 空, 或已被清零	UR1STAT[1]	0x95	RO-x
UR0ERRIE	接收错误中断	1 = 使能 0 = 关闭	UR0CR2[2]	0x188	RW-x
UR1ERRIE		注: 接收错误中断产生条件 URxOVF = 1 URxPEF = 1 URxFEf = 1	UR1CR2[2]	0x93	RW-x
UR0OVF ¹	接收 BUF 溢出错误标志	1 = 溢出	UR0STAT[5]	0x19F	RW0-x
UR1OVF ¹		0 = 正常, 或已被清零	UR1STAT[5]	0x95	RW0-x
UR0PEF ¹	接收到奇偶校验错误标志	1 = 错误	UR0STAT[4]	0x19F	RW0-x
UR1PEF ¹		0 = 正确, 或已被清零	UR1STAT[4]	0x95	RW0-x
UR0FEF ¹	接收到帧错误标志	1 = 错误	UR0STAT[2]	0x19F	RW0-x
UR1FEF ¹		0 = 正确, 或已被清零	UR1STAT[2]	0x95	RW0-x

表 17-2 UART0 / UART1 中断使能和状态位

¹ 写 0 清零, 写 1 无效

名称	状态		寄存器	地址	复位值	
TX0PO	<u>UART0_TX 管脚映射</u>		COMAF1[5:3]	0x286	RW-000	
	当 UR0SW=0 时:	当 UR0SW=1 时:				
	0xx = <u>PC2</u>	0xx = <u>PC1</u>				
	100 = PC0	100 = PB7				
	101 = PB3	101 = PB2				
RX0PO	<u>UART0_RX 管脚映射</u>		COMAF1[2:0]	0x286	RW-000	
	当 UR0SW=0 时:	当 UR0SW=1 时:				
	0xx = <u>PC1</u>	0xx = <u>PC2</u>				
	100 = PB7	100 = PC0				
	101 = PB2	101 = PB3				
UR1SW	<u>UART1_TX 和 RX 管脚交换</u>		COMAF2[7]	0x287	RW-0	
	1 = 交换 (详见 TX1PO 和 RX1PO) 0 = 正常					
UR0SW	<u>UART0_TX 和 RX 管脚交换</u>		COMAF2[6]	0x287	RW-0	
	1 = 交换 (详见 TX0PO 和 RX0PO) 0 = 正常					
TX1PO	<u>UART1_TX 管脚映射</u>		COMAF2[4:3]	0x287	RW-00	
	当 UR1SW=0 时:	当 UR1SW=1 时:				
	0x = <u>PC7</u>	0x = <u>PA0</u>				
	10 = PC5	10 = PC6				
RX1PO	<u>UART1_RX 管脚映射</u>		COMAF2[1:0]	0x287	RW-00	
	当 UR1SW=0 时:	当 UR1SW=1 时:				
	0x = <u>PA0</u>	0x = <u>PC7</u>				
	10 = PC6	10 = PC5				
ODCONA	PORTA 开漏	1 = 使能 0 = 关闭	ODCONA[7:0]	0x205	RW-0000 0000	
	ODCONB		PORTB 开漏	ODCONB[7:0]	0x206	RW-0000 0000
	ODCONC		PORTC 开漏	ODCONC[7:0]	0x207	RW-0000 0000
	ODCOND		PORTD 开漏	ODCOND[5:0]	0x208	RW---00 0000

表 17-3 UART0 / UART0 接口引脚控制

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
UR0DATL ²	0x185	UR0DATL[7:0]								xxxx xxxx
UR0DATH ²	0x186	—							UR0DATH	---- ---x
UR0CR1 ²	0x187	UR0HDSEL	UR0STOP	UR0MODE	UR0EVEN	UR0PCEN	UR0RXEN	UR0TXEN	UR0EN	xxxx xxxx
UR0CR2 ²	0x188	UR0BRRH[3:0]				—	UR0ERRIE	UR0RXNEIE	UR0TXEIE	xxxx -xxx
UR0BRRL ²	0x189	UR0BRRL[7:0]								xxxx xxxx
UR0STAT ²	0x19F	—	UR0TXBSY	UR0OVF	UR0PEF	UR0RXFULL	UR0FEF	UR0RXNEF	UR0TXEF	-xxx xxxx
UR1DATL ²	0x90	UR1DATL[7:0]								xxxx xxxx
UR1DATH ²	0x91	—							UR1DATH	---- ---x
UR1CR1 ²	0x92	UR1HDSEL	UR1STOP	UR1MODE	UR1EVEN	UR1PCEN	UR1RXEN	UR1TXEN	UR1EN	xxxx xxxx
UR1CR2 ²	0x93	UR1BRRH[3:0]				—	UR1ERRIE	UR1RXNEIE	UR1TXEIE	xxxx -xxx
UR1BRRL ²	0x94	UR1BRRL[7:0]								xxxx xxxx
UR1STAT ²	0x95	—	UR1TXBSY	UR1OVF	UR1PEF	UR1RXFULL	UR1FEF	UR1RXNEF	UR1TXEF	-xxx xxxx

表 17-4 UART0 / UART1 相关寄存器地址

17.2 UARTx 异步工作模式

全双工和半双工的配置流程：

1. 设置通信波特率 = $F_{master} / (16 * \{URxBRRH, URxBRRL\})$ (参阅 "URxBRRH", "URxBRRL");
2. 设置通信数据长度为 8 或 9 位 (参阅 "URxMODE");
3. 设置奇偶校验位 (参阅 "URxPCEN", "URxEVEN");
4. 设置停止位长度为 1 或 2 位 (参阅 "URxSTOP");
5. 选择全双工 (默认) 或半双工工作模式 (参阅 "URxHDSEL");
6. 如需要, 可使能相应的中断 (参阅 "GIE", "PEIE", "URxTXEIE", "URxRXNEIE" 和 "URxERRIE" 等);
7. 根据需要, 设置 URxTXEN = 1 或 URxRXEN = 1, 使能发送或接收功能;
8. 设置 URxEN = 1, 使能 UARTx 模块;

注：

1. UARTx 外设时钟 $F_{master} = Sysclk$;
2. 数据发送 FIFO 深度为 1, 接收 FIFO 深度为 2;
3. 半双工模式下, 如果同时使能发送和接收功能, 则发送的数据也会被本机接收到;
4. 建议在关闭 UARTx 模块 (URxEN = 0) 时, 再更改相关配置, 否则将导致数据传输错误;

异步模式的数据通信格式为先发送低位, 后发送高位。有无奇偶校验位的数据帧格式对比如下图：

² 寄存器复位值不确定, 应用程序在系统初始化时应将其写为合适的值, 特别是各中断使能位, 以免产生中断及不必要的功耗。

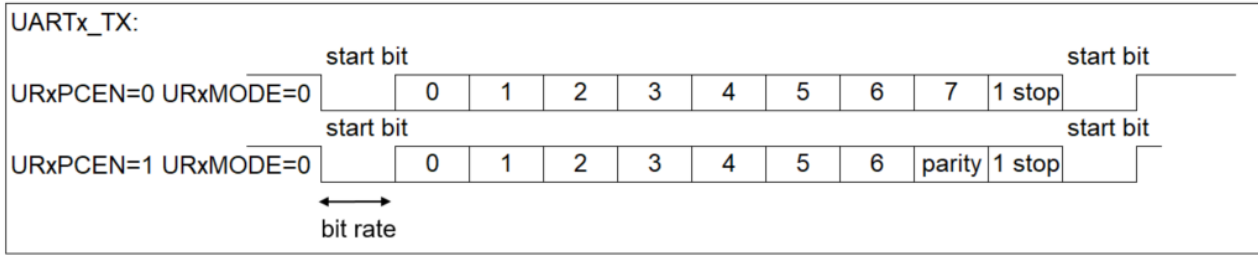


图 17-2 异步模式通信格式 (以 8bit 数据格式为例)

数据处理流程包括阻塞模式和非阻塞模式:

	阻塞模式	非阻塞模式
发送数据	向 URxDATL/H (TX-FIFO) 写入数据后, 有 2 种查询方式如下: 1. 查询 URxTXEF, 当其置 1 时, 写入下一个数据; 2. 查询 URxTXBSY, 当其清 0 后, 写入下一个数据;	当 URxTXEIE = 1 时, 向 URxDATL/H (TX-FIFO) 写入数据后, URxTXEF 置 1 则进入中断
接收数据	查询 URxRXNEF 或 URxRXFULL, 当其置 1 时, 则可读取 URxDATL/H (RX-FIFO) 的值;	当 URxRXNEIE = 1 时, URxRXNEF 置 1 后进入中断; 此外, 建议使能 URxERRIE 中断, 当接收错误时则进入相应的中断进行处理;
备注	-	进入中断后, 查询相应的状态标志位并处理发送接收流程, 处理完成后退出中断

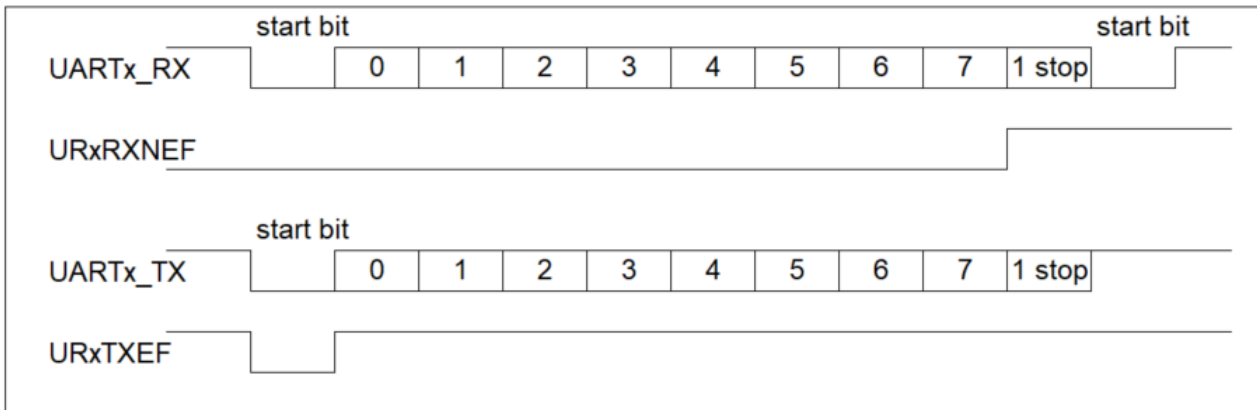


图 17-3 异步模式标志位时序图

18. TOUCH 模块

FT62F28x 片内集成多路触控按键功能，可替代传统的机械式轻触按键，外围简单，安全性高。

- 多达 28 个触摸按键
- 通过外接参考电容 C_{REF} 调节灵敏度 (推荐 10nF)
- 抗干扰能力强

用户可通过仿真器 IDE 内置的 FMDTouchTool 软件及库函数快速开发触摸相关应用, 开发界面示例如下:

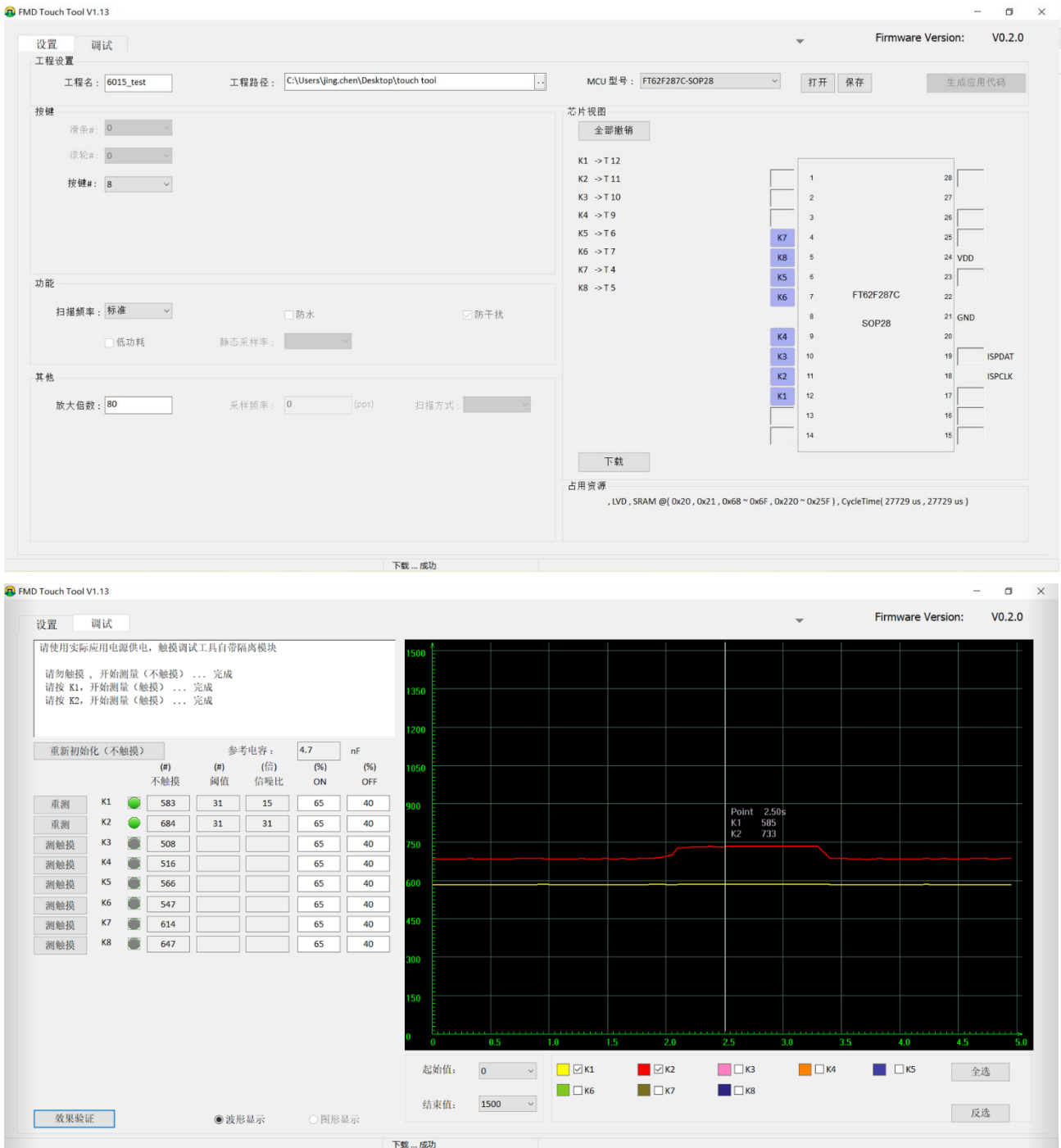


图 18-1 Touch 开发界面

19. 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 1k x 14), 数据 EEPROM 区(DROM)可配置为全区读保护。。这些保护功能均由 IDE 界面进行选择配置。

名称	功能	默认
CPDB	DROM 全区读保护	关闭
CPB	PROM 全区读保护	关闭
SECPB0	PROM 扇区 0 (1k x 14) 读/写保护	关闭
SECPB1	PROM 扇区 1 (1k x 14) 读/写保护	关闭
SECPB2	PROM 扇区 2 (1k x 14) 读/写保护	关闭
SECPB3	PROM 扇区 3 (1k x 14) 读/写保护	关闭
SECPB4	PROM 扇区 4 (1k x 14) 读/写保护	关闭
SECPB5	PROM 扇区 5 (1k x 14) 读/写保护	关闭
SECPB6	PROM 扇区 6 (1k x 14) 读/写保护	关闭
SECPB7	PROM 扇区 7 (1k x 14) 读/写保护	关闭

表 19-1 存储区读/写保护初始化配置寄存器

20. 低功耗模式 (LOW POWER MODE)

随着系统时钟 Sysclk 频率的增加, 功耗也相应增加。但当 Sysclk 为高频时, 完成既定任务所需的时间更短, 从而使 CPU 可以更快地进入 SLEEP 模式。结合这两个相对的因素, 通常情况下, 在所允许的最高频率下运行时, 一个既定任务所需的功耗是最低的。FT62F28x 在 16MHz / 2T 的速度下功耗低至 230 μ A/MIPS, 为业内功耗最低的产品之一。

有些应用需要 CPU 保持运行而不进入睡眠, 此时 SysClk 建议选择 LIRC。为了进一步降低功耗, 可通过使能 ROMLPE 来让 CPU 进入低功耗模式。此低功耗模式仅适用于指令时钟低于 1 MHz。

当 LIRC 为 32 kHz 时, 低功耗模式的工作电流为 ~ 11 μ A。

名称	状态		寄存器	地址	复位值
ROMLPE	低功耗模式	1 = 使能 0 = 关闭	MSCON0[5]	0x10C	RW-0

表 20-1 低功耗模式

21. 指令集 (INSTRUCTION SET)

汇编语法	功能	运算	状态位
NOP	空操作	None	NONE
SLEEP	进入 SLEEP 模式	0 → WDT; Stop OSC	/PF, /TF
CLRWDT	清看门狗 (喂狗)	0 → WDT	/PF, /TF
LJUMP N	无条件跳转	N → PC	NONE
LCALL N	调用子程序	N → PC; PC + 1 → Stack	NONE
RETI	从中断返回	Stack → PC; 1 → GIE	NONE
RET	从子程序返回	Stack → PC	NONE
BCR R, b	将寄存器 R 的 b 位清 0	0 → R(b)	NONE
BSR R, b	将寄存器 R 的 b 位置 1	1 → R(b)	NONE
CLRR R	将寄存器 R 清 0	0 → R	Z
LDR R, d (MOVf)	将 R 存到 d	R → d	Z
COMR R, d	R 的反码	/R → d	Z
INCR R, d	R + 1	R + 1 → d	Z
INCRSZ R, d	R + 1, 结果为 0 则跳过	R + 1 → d	NONE
DECR R, d	R - 1	R - 1 → d	Z
DECRSZ R, d	R - 1, 结果为 0 则跳过	R - 1 → d	NONE
SWAPR R, d	将寄存器 R 的半字节交换	R(0-3)R(4-7) → d	NONE
RRR R, d	R 带进位循环右移	R(0) → C; R(n) → R(n-1); C → R(7);	C
RLR R, d	R 带进位循环左移	R(7) → C; R(n) → R(n+1); C → R(0);	C
BTSC R, b	位测试, 结果为 0 则跳过	Skip if R(b)=0	NONE
BTSS R, b	位测试, 结果为 1 则跳过	Skip if R(b)=1	NONE
CLRW	将工作寄存器 W 清 0	0 → W	Z
STTMD	将 W 内容存到 OPTION	W → OPTION	NONE
CTLIO R	设置 I/O 方向控制寄存器 TRISr	W → TRISr	NONE
STR R (MOVWF)	将 W 存到 R	W → R	NONE
ADDWR R, d	W 与 R 相加	W + R → d	C, HC, Z
SUBWR R, d	R 减 W	R - W → d	C, HC, Z
ANDWR R, d	W 与 R 相与	R & W → d	Z
IORWR R, d	W 与 R 相或	W R → d	Z
XORWR R, d	W 与 R 异或	W ^ R → d	Z
LDWI I (MOVLW)	将立即数存到 W	I → W	NONE
ANDWI I	W 与立即数 I 相与	I & W → W	Z
IORWI I	W 与立即数 I 相或	I W → W	Z
XORWI I	W 与立即数 I 异或	I ^ W → W	Z
ADDWI I	W 与立即数 I 相加	I + W → W	C, HC, Z
SUBWI I	立即数 I 减 W	I - W → W	C, HC, Z
RETW I	返回, 将立即数 I 存到 W	Stack → PC; I → W	NONE

表 21-1 37 条 RISC 指令

字段	描述
R(F)	SFR/SRAM 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
X	不关心, 值可以为 0 或 1
d	<u>目标寄存器选择</u> 1 = 结果存放到寄存器 R / RAM 0 = 结果存放到 W
N	程序绝对地址
PC	程序计数器
/PF	掉电标志位
/TF	超时标志位
TRISr	TRISr 寄存器, r 可以是 A, B, C
C	进位 / 借位
HC	半进位 / 半借位
Z	0 标志位

表 21-2 操作码字段

名称	状态	寄存器	地址	复位值
Z	<u>0标志位: 算术或逻辑运算的结果为零?</u> 1 = Yes 0 = No	STATUS[2]	0x03 0x83 0x103 0x183	RW-x
HC	<u>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?</u> 1 = 进位, 或未借位 0 = 未进位, 或借位	STATUS[1]		RW-x
C	<u>进位 / 借位 (ADDWR, ADDWI, SUBWI, SUBWR): 结果的最高位发生了进位或借位?</u> 1 = 进位, 或未借位 0 = 未进位, 或借位	STATUS[0]		RW-x

表 21-3 计算状态标志位

22. 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;

22.1 初始化配置寄存器

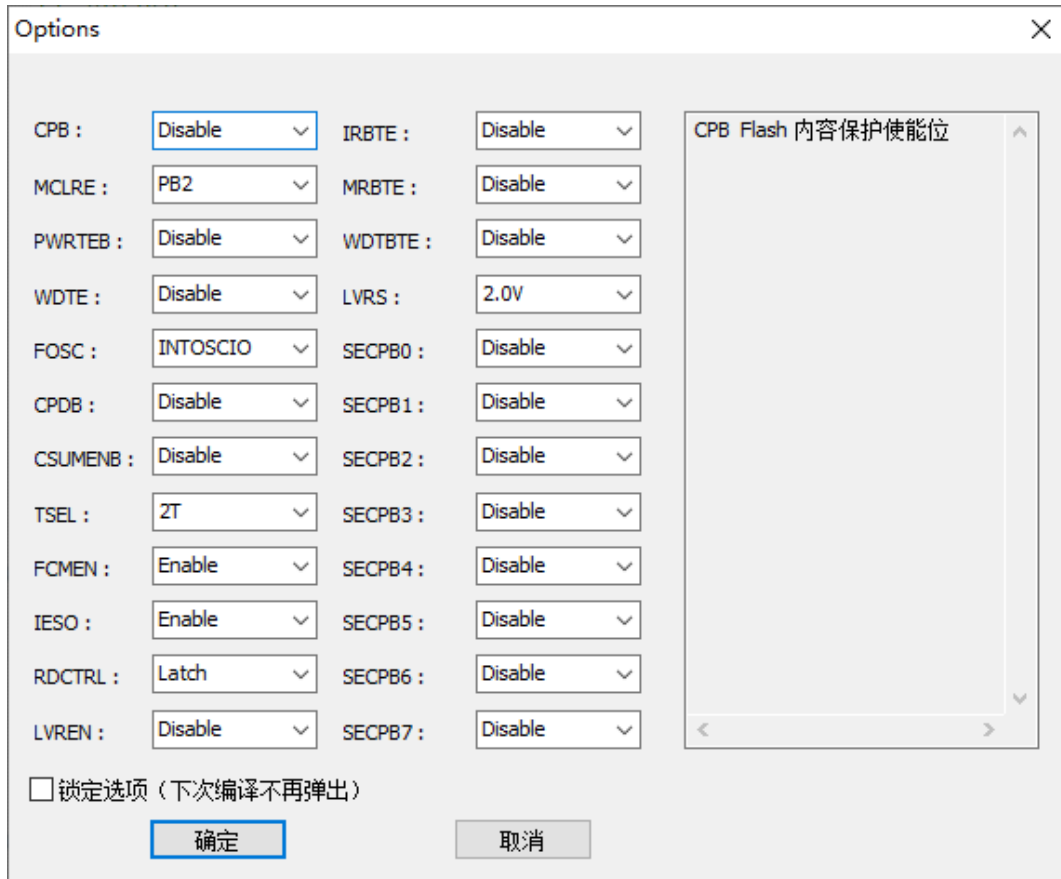


图 22-1 由 IDE 设置的初始化配置寄存器

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	<u>WDT</u> <ul style="list-style-type: none"> 使能 (指令不能禁止) 由指令控制 (SWDTEN) 	SWDTEN 控制
FOSC	<ul style="list-style-type: none"> LP: PA1 (+) 和 PA2(-) 接外部低速晶振 XT: PA1 (+) 和 PA7 (-) 接外部高速晶振 EC: PA1 (+) 接外部时钟输入, PA7 为 I/O INTOSC: PA2 或 PD5 输出"指令时钟", PA1 为 I/O INTOSCIO: PA1 和 PA2 为 I/O 	INTOSCIO
CPDB	DROM 全区读保护	关闭
CSUMEB	程序空间校验和的验证功能	关闭
TSEL	<u>指令时钟与系统时钟 SysClk 的对应关系 (2T or 4T):</u> <ul style="list-style-type: none"> 2 (指令时钟= SysClk/2) 4 (指令时钟= SysClk/4) 	2
FCMEN	<u>故障保护时钟监控器</u> <ul style="list-style-type: none"> 使能 关闭 	使能
IESO	<u>XT / LP 双速时钟启动</u> <ul style="list-style-type: none"> 使能 关闭 	使能
RDCTRL	<u>当 TRISx = 0 时, 读 PORTx 寄存器的返回值</u> <ul style="list-style-type: none"> 输入锁存器 输出锁存器 	输出
LVREN	<u>LVR</u> <ul style="list-style-type: none"> 使能 关闭 非 SLEEP 模式下使能 通过指令控制 (SLVREN) 	关闭
IRBTE	非法指令复位启动初始化配置	关闭
MRBTE	MCLRE 复位启动初始化配置	关闭
WDTBTE	WDT 复位启动初始化配置	关闭
LVRS	<u>7 档 V_{BOR} 电压(V):</u> 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
FSECPB4	PROM 扇区 4 (1k x 14) 读/写保护	关闭
FSECPB5	PROM 扇区 5 (1k x 14) 读/写保护	关闭
FSECPB6	PROM 扇区 6 (1k x 14) 读/写保护	关闭
FSECPB7	PROM 扇区 7 (1k x 14) 读/写保护	关闭

表 22-1 初始化配置寄存器 (由 IDE 设置)

22.2 用户寄存器

用户寄存器，即特殊功能寄存器(SFR)和 SRAM 分布在 8 个 bank (bank0~7) 中，每个 bank 大小为 128 字节。在访问寄存器前，必须先切换到相应的 bank (参阅 "AUXPGE" 和 "PAGE")。

由于切换 bank 需要额外的指令，因此一些常用的 SFR 同时存储在 8 个 bank 中，以减少切换操作，这 8 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

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	INDF	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	FSRB8	PAGE[1:0]		/TF	/PF	Z	HC	C	0001 1xxx
首地址 + 4H	FSR	间接寻址指针寄存器								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	TOIE	INTE	PAIE	TOIF	INTF	PAIF	0000 0000
首地址 + (70 - 7F)	公共 BANK SRAM 区									xxxx xxxx

表 22-2 8 个 BANK 共有的寄存器

地址	名称	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	复位值
00	INDF	使用 FSR 的内容对数据存储寄存器进行访问 (非物理寄存器)								xxxx xxxx
01	TMR0	Timer0 计数器								xxxx xxxx
02	PCL	程序计数器低 8 位								0000 0000
03	STATUS	FSRB8	PAGE[1:0]		/TF	/PF	Z	HC	C	0001 1xxx
04	FSR	间接寻址指针寄存器								xxxx xxxx
05	PORTA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	xxxx xxxx
06	PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	xxxx xxxx
07	PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	xxxx xxxx
08	PORTD	-	-	PD5	PD4	PD3	PD2	PD1	PD0	--xx xxxx
09	PIR2	-	-	-	-	-	-	C1IF	C0IF	---- --00
0A	PCLATH	-	-	-	程序计数器高 5 位锁存器					---0 0000
0B	INTCON	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000
0C	PIR1	EEIF	CKMIF	LVDIF	ACMP1F	TMR1IF	OSFIF	TMR2IF	ADCIF	0000 0000
0D	P1ADTL	P1A 占空比寄存器低 8 位								0000 0000
0E	P1ADTH	P1A 占空比寄存器高 8 位								0000 0000
0F	P1BDTL	P1B 占空比寄存器低 8 位								0000 0000
10	P1BDTH	P1B 占空比寄存器高 8 位								0000 0000
11	P1CDTL	P1C 占空比寄存器低 8 位								0000 0000
12	P1CDTH	P1C 占空比寄存器高 8 位								0000 0000
13	P1DDTL	P1D 占空比寄存器低 8 位								0000 0000
14	P1DDTH	P1D 占空比寄存器高 8 位								0000 0000
15	TMR2L	Timer2 计数器[7:0]								0000 0000
16	TMR2H	Timer2 计数器[15:8]								0000 0000
17	PR2L	PR2[7:0], Timer2 周期寄存器低 8 位								1111 1111
18	PR2H	PR2[15:8], Timer2 周期寄存器高 8 位								1111 1111
19	P1BR0	P1BEVT	P1BKS[2:0]			P1BSS[1:0]		P1ASS[1:0]		0000 0000
1A	P1BR1	P1D2SS[1:0]		P1DSS[1:0]		P1C2SS[1:0]		P1CSS[1:0]		0000 0000
1B	P1POL0	P1C0P	P1B0P	P1A2NP	P1A2P	P1A1NP	P1A1P	P1A0NP	P1A0P	0000 0000
1C	P1POL1	P1D2P	P1D1P	P1D0P	-	-	P1C1P	P1B1P	-	00- -00-
1D	P1CON	P1AUE	P1DC[6:0]							0000 0000
1E	T2CON0	PR2U	TOUTPS[3:0]				TMR2ON	T2CKPS[1:0]		0000 0000
1F	T2CON1	-	-	-	P1OS	P1BZM	T2CKSRC[2:0]			---0 0000
20-6F		SRAM BANK0 (80Bytes)								xxxx xxxx
70-7F		SRAM BANK0 (16Bytes)								xxxx xxxx

表 22-3 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	INDF	使用 FSR 的内容对数据存储寄存器进行访问 (非物理寄存器)								xxxx xxxx
81	OPTION	/PAPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111
82	PCL	程序计数器低 8 位								0000 0000
83	STATUS	FSRB8	PAGE[1:0]		/TF	/PF	Z	HC	C	0001 1xxx
84	FSR	间接寻址指针寄存器								xxxx xxxx
85	TRISA	PORTA 方向控制								1111 1111
86	TRISB	PORTB 方向控制								1111 1111
87	TRISC	PORTC 方向控制								1111 1111
88	TRISD	-	-	PORTD 方向控制						--11 1111
89	PIE	-	-	-	-	-	-	C1IE	C0IE	---- --00
8A	PCLATH	-	-	-	程序计数器高 5 位锁存器					---0 0000
8B	INTCON	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000
8C	PIR1	EEIE	CKMIE	LVDIE	ACMPIE	TMR1IE	OSFIE	TMR2IE	ADCIE	0000 0000
8D	P1AUX	P1BKM	P1CDM	P1B2SS[1:0]		P1CF2E	P1CF2	P1DF2E	P1DF2	0000 0000
8E	PCON	LVLDL[3:0]				LVDEN	LVDW	/POR	/BOR	0000 0xqq
8F	OSCCON	LFMOD	IRCF[2:0]			OSTS	HTS	LTS	SCS	0101 x000
90	UR1DATL	UR1DATL[7:0]								xxxx xxxx
91	UR1DATH	-	-	-	-	-	-	-	UR1DATH	---- ---x
92	UR1CR1	UR1HSEL	UR1STOP	UR1MODE	UR1EVEN	UR1PCEN	UR1RXEN	UR1TXEN	UR1EN	xxxx xxxx
93	UR1CR2	UR1BRRH[3:0]				-	UR1ERRIE	UR1RXNEIE	UR1TXEIE	xxxx -xxx
94	UR1BRRL	UR1BRRL[7:0]								xxxx xxxx
95	UR1STAT	-	URxTXBSY	URxOVF	URxPEF	URxRXFULL	URxFEF	URxRXNEF	URxTXEF	xxxx xxxx
96	OP0CR0	OP0OUT	OP0PSEL	OP0NSEL[1:0]		OP0FR[2:0]			OP0ON	x110 1100
97	OP0CR1	-	-	OPINSW	IPINSW	OP0CAPE	-	OPTODIG	OPTOIO	--00 0-00
98	IOCA	IOCA[7:0]								0000 0000
99	WDTCON	-	WCKSRC[1:0]		WDTPS[3:0]			SWDTEN		-000 1000
9A	EEDAT	EEDAT[7:0]								0000 0000
9B	EEADR	EEADR[7:0]								0000 0000
9C	EECON1	-	Reserved	WREN3	WREN2	WRERR	WREN1	PONLY	RD	-000 x000
9D	EECON2	-	-	-	-	-	-	-	WR	---- ---0
9E	P1OE0	P1C0OE	P1B0OE	P1A2NOE	P1A2OE	P1A1NOE	P1A1OE	P1A0NOE	P1A0OE	0000 0000
9F	P1OE1	P1D2OE	P1D1OE	P1D0OE	-	-	P1C1OE	P1B1OE	-	000- -00-
A0-EF		SRAM BANK1 (80Bytes)								xxxx xxxx
F0-FF		SRAM 访问 BANK0 的 0x70-0x7F								xxxx xxxx

表 22-4 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	INDF	使用 FSR 的内容对数据存储寄存器进行访问 (非物理寄存器)								xxxx xxxx	
101	MSCON1	-	-	-	-	LVRDEB	LVRCKS	AUXPGE	HIRCM	---- 1000	
102	PCL	程序计数器低 8 位								0000 0000	
103	STATUS	FSRB8	PAGE[1:0]		/TF	/PF	Z	HC	C	0001 1xxx	
104	FSR	间接寻址指针寄存器								xxxx xxxx	
105	T0CON0	T0DZ[1:0]		P0ANOE	P0AOE	T0ON	T0CKRUN	T0CKSRC[1:0]		0000 1000	
106	T1CON0	-	T1AUSTP	T1ECKE	T1CKPSA	T1CKRUN	T1ON	T1CKSRC[1:0]		-000 0000	
107	PR0	Timer0 产生的 PWM 波周期设置								1111 1111	
108	T0DUTY	Timer0 产生的 PWM 波占空比设置								0000 0000	
109	FOSCCAL	FOSCCAL[7:0]								xxxx xxxx	
10A	PCLATH	-	-	-	程序计数器高 5 位锁存器					---0 0000	
10B	INTCON	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000	
10C	MSCON0	-	-	ROMLPE	CLKOS	SLVREN	CKMAVG	CKCNT1	T2CKRUN	0001 0000	
10D	MSCON2	P0ANP	P0AP	CMAUSTR	STRTYP[1:0]		T1EVTS[2:0]			0000 0000	
10E	LEBCON	LEBEN	LEBCH[1:0]		-	EDGS	-	-	-	000- 0----	
10F	ADCMPPH	ADCMPPH[7:0]								0000 0000	
110	ADDLY	ADDLY[7:0] / LEBPRL[7:0]								0000 0000	
111	ADRESL	左对齐格式下 AD 结果的低 4 位或者右对齐下结果的低 8 位								xxxx xxxx	
112	ADRESH	左对齐格式下 AD 结果的高 8 位或者右对齐格式下的高 4 位								xxxx xxxx	
113	ADCON0	CHS[4:0]					ADEX	GO/DONE	ADON		0000 0000
114	ADCON1	ADFM	ADCS[2:0]			ADNREF[1:0]		ADPREF[1:0]		0010 0100	
115	ADCON2	ADINTREF[1:0]		ETGTYP[1:0]		ADDLY.8	ETGSEL[2:0]			0100 0000	
116	ADCON3	ADFBEN	ADCMPP	ADCMPE	-	LEBADT	-				000- 0----
117	PR1L	PR1 周期寄存器低 8 位 PR1[7:0]								1111 1111	
118	PR1H	-	-	-	-	PR1 周期寄存器高 4 位 PR1[11:8]				---- 1111	
119	TMR1L	Timer1[7:0]								0000 0000	
11A	TMR1H	-	-	-	-	Timer1[11:8]				---- 0000	
11B	TKCON0	-	-	TKRSEL[1:0]		TKMOD	TKCHGS[1:0]		TKON	--00 0000	
11C	TKEN0	TKEN0[7:0]								0000 0000	
11D	TKEN1	TKREN	-	TKEN1[5:0]						0-00 0000	
11E	ANSEL0	ANSEL[7:0]								0000 0000	
11F	ANSEL1	ANSEL[15:8]								0000 0000	
120-16F		SRAM BANK2 (80Bytes)								xxxx xxxx	
170-17F		SRAM 访问 BANK0 的 0x70-0x7F								xxxx xxxx	

表 22-5 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	INDF	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	FSRB8	PAGE[1:0]		/TF	/PF	Z	HC	C	0001 1xxx
184	FSR	间接寻址指针寄存器								xxxx xxxx
185	UR1DATL	UR1DATL[7:0]								xxxx xxxx
186	T1CON0	-	-	-	-	-	-	-	UR1DATH	---- --x
187	UR0CR1	UR0HSEL	UR0STOP	UR0MODE	UR0EVEN	UR0PCEN	UR0RXEN	UR0TXEN	UR0EN	xxxx xxxx
188	UR0CR2	UR0BRRH[3:0]				-	UR0ERRIE	UR0RXNEIE	UR0TXEIE	xxxx -xxx
189	UR1BRRL	UR1BRRL[7:0]								xxxx xxxx
18A	PCLATH	-	-	-	程序计数器高 5 位锁存器					---0 0000
18B	INTCON	GIE	PEIE	TOIE	INTE	PAIE	TOIF	INTF	PAIF	0000 0000
18C	SPIDAT	SPIDAT[7:0]								0000 0000
18D	SPICR0	-	-	-	-	NSSM		TXBMT	SPIEN	---- 0010
18E	SPICFG	SPIBUSY	MSTEN	CPHA	CPOL	SLAS	NSSVAL	SRMT	RXBMT	xxxx xxxx
18F	SPISCR	SPISCR[7:0]								xxxx xxxx
193	SPIIER	-	-	-	-	WAKUP	RXERR	RXNE	TXE	---- xxxx
194	SPICR1	BDM	BDOE	RXONLY	SSI	SSM	-	-	LSBFIRST	xxxx xxxx
195	SPISTAT	SPIF	WCOL	MODF	RXOVRN	-	-	WKF	-	0000 --0-
196	I2CCR1	STRET	-	ADDRM	STOP	WRN	START	MASTER	I2CEN	x-xx xxxx
197	I2CCR2	I2CADDRH[1:0]			NACK	DUTY	-	-	-	xxxx ---x
198	I2CADDR	2CADDRL[7:0]								xxxx xxxx
199	I2CCCR	I2CCCR[7:0]								xxxx xxxx
19A	I2CDAT	I2CDAT[7:0]								xxxx xxxx
19B	-	未实现, 读 0								---- ----
19C	TKEN2	TKEN2[7:0]								0000 0000
19D	TKEN3	VREFM	TKCAPM	TKEN3[5:0]						0000 0000
19E	I2CISR	HOLDF	STOPF	ADDRF	NACKF	I2CBUSY	DIRF/ BUSERR	IICRXNEF	IICTXEF	0000 0000
19F	UR0STAT	-	UR0TXBSY	UR0OVF	UR0PEF	UR0RXFULL	UR0FEF	UR0RXNEF	UR0TXEF	-xxx xxxx
1A0-1EF		SRAM BANK3 (80Bytes)								xxxx xxxx
1F0-1FF		SRAM 访问 BANK0 的 0x70-0x7F								xxxx xxxx

表 22-6 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	INDF	使用 FSR 的内容对数据存储寄存器进行访问 (非物理寄存器)								xxxx xxxx
201	-	未实现, 读 0								---- ----
202	PCL	程序计数器低 8 位								0000 0000
203	STATUS	FSRB8	PAGE[1:0]	/TF	/PF	Z	HC	C		0001 1xxx
204	FSR	间接寻址指针寄存器								xxxx xxxx
205	ODCONA	ODCONA[7:0]								0000 0000
206	ODCONB	ODCONB[7:0]								0000 0000
207	ODCONC	ODCONC[7:0]								0000 0000
208	ODCOND	-	-	ODCOND[5:0]						--00 0000
209	SCKCFG	SCKEN	-	-	-	-	SCKOE	-		0000 0000
20A	PCLATH	-	-	-	程序计数器高 5 位锁存器					---0 0000
20B	INTCON	GIE	PEIE	TOIE	INTE	PAIE	TOIF	INTF	PAIF	0000 0000
20C	PSRCAL	PSRCAL[7:0], PORTA 源电流设置, 低字节								1111 1111
20D	PSRCAH	PSRCAH[7:0], PORTA 源电流设置, 高字节								1111 1111
20E	PSRCBL	PSRCBL[7:0], PORTB 源电流设置, 低字节								1111 1111
20F	PSRCBH	PSRCBH[7:0], PORTB 源电流设置, 高字节								1111 1111
210	PSRCCL	PSRCCL[7:0], PORTC 源电流设置, 低字节								1111 1111
211	PSRCCH	PSRCCH[7:0], PORTC 源电流设置, 高字节								1111 1111
212	PSRCDL	PSRCDL[7:0], PORTD 源电流设置, 低字节								1111 1111
213	PSRCDH	-	-	-	-	PSRCDH[3:0], PORTD 源电流设置, 高字节			---- 1111	
214	PSINKA	PSINKA[7:0], PORTA 灌电流设置								0000 0000
215	PSINKB	PSINKB[7:0], PORTB 灌电流设置								0000 0000
216	PSINKC	PSINKC[7:0], PORTC 灌电流设置								0000 0000
217	PSINKD	-	-	PSINKD[5:0], PORTD 灌电流设置						--00 0000
218	SOSCPRL	SOSCPRL[7:0], 慢时钟测量结果低 8 位								1111 1111
219	SOSCPRH	-	-	-	-	SOSCPR[11:8], 慢时钟测量结果高 4 位			---- 1111	
21A	LVDCON	SYSON	LVDCKS[1:0]	LVDP	LVDDDB	-	LVDM[1:0]			-000 1100
21B	SECCODE	加密寄存器								xxxx xxxx
21C	LVDTRIM	-	LVDADJ[3:0]			LVRADJ[2:0]				-xxx xxxx
21D	-	-								xxxx xxxx
21E	-	-								xxxx xxxx
21F	-	-								xxxx xxxx
220-26F		SRAM BANK4 (80Bytes)								xxxx xxxx
270-27F		SRAM 访问 BANK0 的 0x70-0x7F								xxxx xxxx

表 22-7 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	INDF	使用 FSR 的内容对数据存储寄存器进行访问 (非物理寄存器)								xxxx xxxx
281	-	未实现, 读 0								---- ----
282	PCL	程序计数器低 8 位								0000 0000
283	STATUS	FSRB8	PAGE[1:0]		/TF	/PF	Z	HC	C	0001 1xxx
284	FSR	间接寻址指针寄存器								xxxx xxxx
285	COMAF0	SCKPO	NSSPO	MOSIPO	MISOPO	SDAPO[1:0]		SCLPO[1:0]		0000 0000
286	COMAF1	INTPO[1:0]		TX0PO[2:0]			RX0PO[2:0]			0000 0000
287	COMAF2	UR1SW	UR0SW	-	TX1PO[1:0]		INTFIXB	RX1PO[1:0]		00-0 0100
28A	PCLATH	-	-	-	程序计数器高 5 位锁存器					---0 0000
28B	INTCON	GIE	PEIE	TOIE	INTE	PAIE	TOIF	INTF	PAIF	0000 0000
28C	ANSEL2	ANSEL2[7:0]								0000 0000
28D	ANSEL3	-	-	ANSEL3[5:0]						--00 0000
28E	TKCTRIM	-	-	Reserved			TKCTRIM[3:0]			--00 0000
28F	-	未实现, 读 0								---- ----
290	WPDA	WPDA[7:0], PORTA 弱下拉使能								0000 0000
291	WPDB	WPDB[7:0], PORTB 弱下拉使能								0000 0000
292	WPDC	WPDC[7:0], PORTC 弱下拉使能								0000 0000
293	WPDD	-	-	WPDD[5:0], PORTD 弱下拉使能						--00 0000
294	WPUA	WPUA[7:0], PORTA 弱上拉使能								1111 1111
295	WPUB	WPUB[7:0], PORTB 弱上拉使能								0000 0000
296	WPUC	WPUC[7:0], PORTC 弱上拉使能								0000 0000
297	WPUD	-	-	WPUD[5:0], PORTD 弱上拉使能						--00 0000
298	-	未实现, 读 0								---- ----
299	-	未实现, 读 0								---- ----
29A	DAC1DAT	-	DAC1DAT[6:0]						-000 0000	
29B	DAC2DAT	-	DAC2DAT[6:0]						-000 0000	
29C	DACCON0	CMDEBS	DEBPR[1:0]		-	WNDEN	DACEN	DACVREF[1:0]		000- 0000
29D	CM0CON0	CM0EN	CM0POL	CM0OE	C0OUT	CM0PSEL[1:0]		CM0NSEL[1:0]		000x 0000
29E	CM1CON0	CM1EN	CM1POL	CM1OE	C1OUT	CM1PSEL[1:0]		CM1NSEL[1:0]		000x 0000
29F	-	未实现, 读 0								---- ----
2A0-2EF		SRAM BANK5 (80Bytes)								xxxx xxxx
2F0-2FF		SRAM 访问 BANK0 的 0x70-0x7F								xxxx xxxx

表 22-8 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	INDF	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	MSCON1	-	-	-	-	LVRDEB	LVRCKS	AUXPGE	HIRCM	---- 1000
302	PCL	程序计数器低 8 位								0000 0000
303	STATUS	FSRB8	PAGE[1:0]	/TF	/PF	Z	HC	C		0001 1xxx
304	FSR	间接寻址指针寄存器								xxxx xxxx
309	FOSCCALH	-	-	-	FOSCCAL12:8]					---- --xx
30A	PCLATH	-	-	-	程序计数器高 5 位锁存器					---0 0000
30B	INTCON	GIE	PEIE	TOIE	INTE	PAIE	TOIF	INTF	PAIF	0000 0000
320-36F		SRAM BANK6 (80Bytes)								xxxx xxxx
370-37F		SRAM 访问 BANK0 的 0x70-0x7F								xxxx xxxx

表 22-9 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	INDF	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	-	未实现, 读 0								---- ----
382	PCL	程序计数器低 8 位								0000 0000
383	STATUS	FSRB8	PAGE[1:0]	/TF	/PF	Z	HC	C		0001 1xxx
384	FSR	间接寻址指针寄存器								xxxx xxxx
38A	PCLATH	-	-	-	程序计数器高 5 位锁存器					---0 0000
38B	INTCON	GIE	PEIE	TOIE	INTE	PAIE	TOIF	INTF	PAIF	0000 0000
3A0-3C7		SRAM BANK7 (40Bytes)								xxxx xxxx
3C8-3EF		未实现, 读 0								---- ----
3F0-3FF		SRAM 访问 BANK0 的 0x70-0x7F								xxxx xxxx

表 22-10 SFR, BANK7

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;

22.3 STATUS 寄存器

名称	状态	寄存器	地址	复位值
FSRB8	FSR 寄存器第 8 位，与 FSR 组成一个 9 位的寄存器，在间接寻址时使用，详见 章节 22.4 。	STATUS[7]	Bank 首地址 +0x03	RW-0
PAGE[1:0]	寄存器存储区(bank)选择位 (用于直接寻址) 00 = Bank0 (0x00h – 0x7Fh) 01 = Bank1 (0x80h – 0xFFh) 10 = Bank2 (0x100 – 0x17Fh) 11 = Bank3 (0x180 – 0x1FFh) 注：访问 Bank4~Bank7 需把 AUXPGE 置 1。	STATUS[6:5]		RW-00
/TF	<u>超时标志位</u> 1 = <u>上电后，执行了 CLRWDT 或 SLEEP 指令</u> 0 = 发生 WDT 超时溢出	STATUS[4]		RO-1
/PF	<u>掉电标志位</u> 1 = <u>上电复位后或执行了 CLRWDT 指令</u> 0 = 执行了 SLEEP 指令	STATUS[3]		RO-1
Z	<u>0 标志位：算术或逻辑运算的结果为零？</u> 1 = Yes 0 = No	STATUS[2]		RW-x
HC	<u>半进位 / 半借位 (ADDWF, ADDLI, SUBL, SUBWF)：</u> <u>结果的第 4 位向高位发生了进位或借位？</u> 1 = 进位，或未借位 0 = 未进位，或借位	STATUS[1]		RW-x
C	<u>进位 / 借位 (ADDWF, ADDWI, SUBWI, SUBWF)：</u> <u>结果的最高位发生了进位或借位？</u> 1 = 进位，或未借位 0 = 未进位，或借位	STATUS[0]		RW-x

表 22-11 STATUS 寄存器

注：

1. 同其他寄存器一样，STATUS 状态寄存器也可以作为任何指令的目标寄存器。但如果一条影响 Z、DC 或 C 位的指令以 STATUS 作为目标寄存器，那么对这三位的写操作将被禁止，Z、DC 和 C 位只受运算结果影响从而被置 1 或清 0。此时，当执行一条以 STATUS 作为目标寄存器的指令后，STATUS 的内容可能与预期不一致。
2. 建议只使用 BCR、BSR、SWAPR 和 STR 指令来操作 STATUS 寄存器。

22.4 PCL 和 PCLATH

程序区分为 Page0 (2k Words), Page1 (2k Words), Page2 (2k Words) 和 Page3 (2k Words)。在 Page0 的末尾 (0x7FF) 将翻转到 Page1 的开头 (0x800), 依此类推, 而在 Page3 的末尾 (0x1FFF) 将回滚到 Page0 的开头 (0x000)。由于指令的地址宽度为 11 位, 只能寻址 2kW。因此对于需要从一个 page 跳转到另一个 page 的 LJUMP 和 LCALL 等长跳转指令, 则需设置 PCLATH 以便先移动到相应的 Page0 到 Page3。从 Page0 移动到 Page1 / Page2 / Page3 的程序示例如下:

```
LDWI 08 ;
STR PCLATH ; PCLATH = 08 (PCLATH = 00 if jump from page1 to page0)
LJUMP label_in_page1 ; Jump to Page1

LDWI 10 ;
STR PCLATH ; PCLATH = 10 (PCLATH = 00 if jump from page2 to page0)
LJUMP label_in_page2 ; Jump to Page2

LDWI 18 ;
STR PCLATH ; PCLATH = 18 (PCLATH = 00 if jump from page3 to page0)
LJUMP label_in_page3 ; Jump to Page3
```

程序计数器(PC)为 13 位宽。其低 8 位来自可读写的 PCL 寄存器, 高 5 位(PC[12:8])来自 PCLATH, 不可直接读写。发生复位时, PC 将被清 0。图 22-2 显示了装载 PC 值的两种情形。

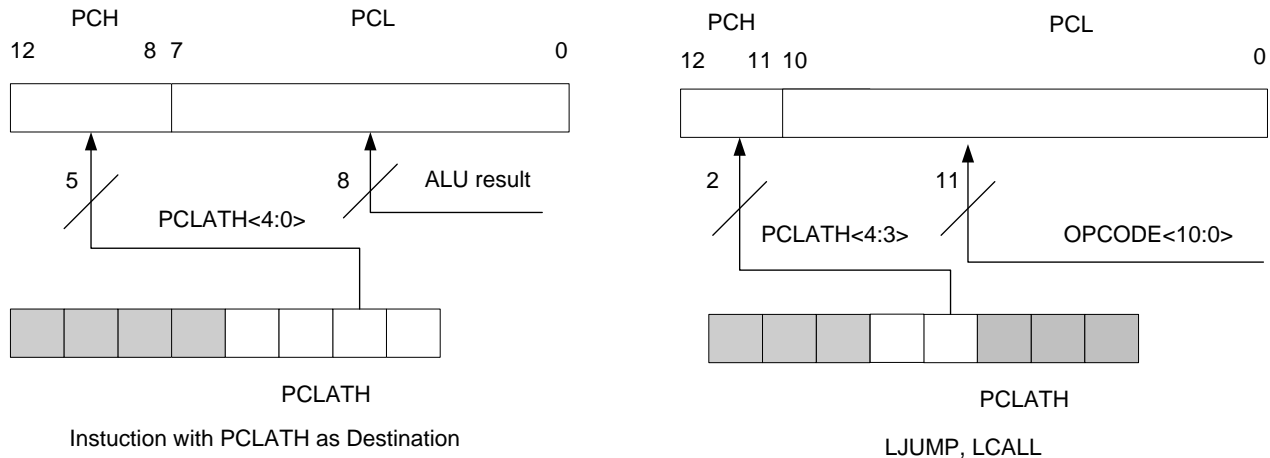


图 22-2 装载 PC 值的不同情况

执行任何以 PCL 寄存器为目标寄存器的指令将同时使程序计数器 PC[12:8]位被 PCLATH 内容所取代。因此可通过将所需的高 5 位先写入 PCLATH 寄存器来更改程序计数器 PC 的全部内容。

计算 LJUMP 指令是通过向程序计数器 PC 加入偏移量(ADDWR PCL)来实现的。因此通过修改 PCL 寄存器来跳转到查找表或程序分支表(计算 LJUMP)时应特别谨慎。假定 PCLATH 设置为表的起始地址, 如果表的长度大于 255 条指令, 或地址低 8 位在表的中间, 计满至 0xFF 返回到 0x00, 那么在每次表的起始地址或表内的目标地址发生计满返回时, PCLATH 必须递增。

INDF 不是物理存在的寄存器, 对 INDF 进行寻址将产生间接寻址。

任何使用 INDF 寄存器的指令，实际上是对文件选择寄存器(File Select Register, [FSRB8, FSR])所指向的单元进行存取。间接对 INDF 进行读操作将返回 0，间接对 INDF 进行写操作将导致空操作（可能会影响状态标志位）。

FT62F28x 系列芯片的 SFR 空间扩展为 1kB。当 AUXPGE = 0 时，间接寻址访问地址为 0x000~0x1FF，当 AUXPGE = 1 时，访问地址为 0x200~0x3FF。

名称	状态	寄存器	地址	复位值
AUXPGE	SFR BANK 间接或直接寻址访问 1 = BANK4 ~ 7 0 = BANK0 ~ 3	MSCON1[1]	0x101 /0x301	RW-0

表 22-12 AUXPGE 寄存器

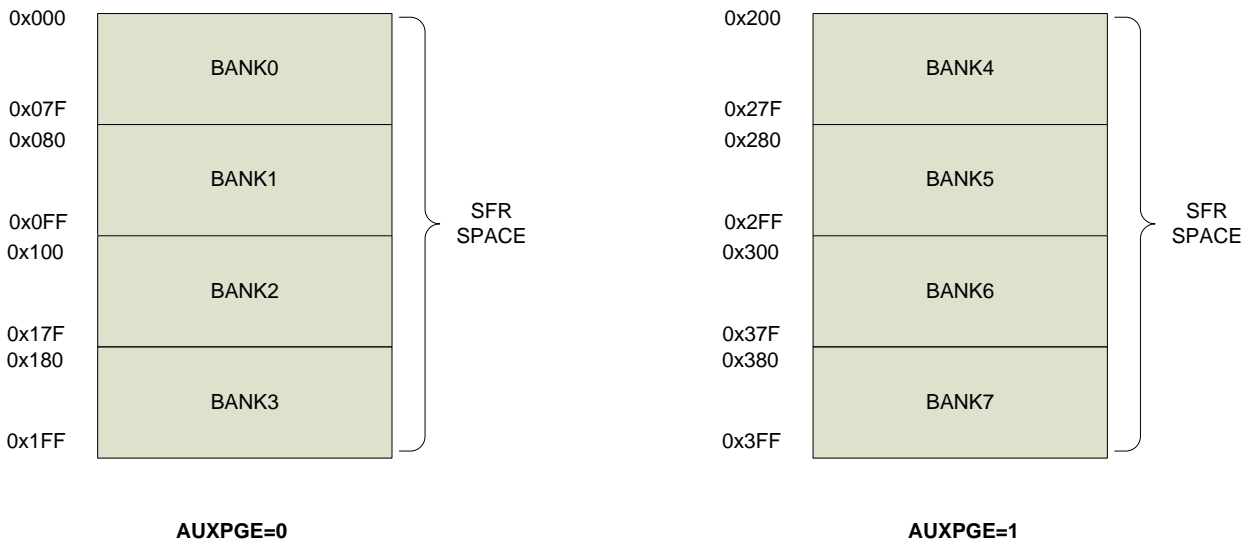


图 22-3 间接寻址

23. 电气特性

23.1 极限参数

工作环境温度等级 3.....	-40 – +85°C
工作环境温度等级 2.....	-40 – +105°C
工作环境温度等级 1.....	-40 – +125°C
存储温度.....	-40 – +125°C
结工作温度范围(Tj).....	-40 – +150°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。
4. 除非另作说明，生产测试温度为 25°C。由于高温筛选不是常规测试流程，超出上述工作环境温度时，芯片的某些性能参数将不能保证。
5. 150°C 下，未经压力测试的典型数据保持时间大于 10 年。

23.2 工作特性

参数		最小值	典型值	最大值	单位	条件
Fsys (SysClk)	2T/4T	-	-	8	MHz	-40 – 85°C, V _{DD} = 1.9 – 5.5V
		-	-	16	MHz	-40 – 85°C, V _{DD} = 2.7 – 5.5V
指令周期 (T _{INSTRCLK})	2T	-	125	-	ns	SysClk = HIRC
	4T	-	250	-	ns	
	2T	-	61	-	µs	SysClk = LIRC
	4T	-	122	-	µs	
T0CKI 高或低脉冲宽度		0.5 * T _{T0CK} + 20	-	-	ns	无预分频
		10	-	-	ns	有预分频
T0CKI 输入周期		Max. 20 and (T _{T0CK} +40)/N	-	-	ns	N = 1, 2, 4, ..., 256 (预分频值)
上电复位保持时间 (T _{DRH})		-	8	-	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	预分频比 = 1:32

注: T_{T0CK} 是指由 T0CKSRC 所选的时钟周期。

23.3 POR, LVR, LVD

上电复位 (POR)

参数	最小值	典型值	最大值	单位	条件
I _{POR} 工作电流	-	0.14	-	μA	25°C, V _{DD} = 3.3V
V _{POR}	-	1.65	-	V	25°C

低电压复位 (LVR)

参数	最小值	典型值	最大值	单位	条件
I _{LVR} 工作电流	-	15.0	-	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	94	-	125	μs	25°C, V _{DD} = 1.9 – 5.5V

低电压检测 (LVD)

参数	最小值	典型值	最大值	单位	条件
I _{LVD} 工作电流	-	20.0	-	μA	25°C, V _{DD} = 3.3V
V _{LVD} , LVD 阈值	1.16	1.2	1.24	V	25°C
	1.94	2.0	2.06		
	2.33	2.4	2.47		
	2.62	2.7	2.78		
	2.91	3.0	3.09		
	3.20	3.3	3.40		
	3.49	3.6	3.71		
	3.88	4.0	4.12		
LVD delay	188	-	250	μs	25°C, V _{DD} = 1.9 – 5.5V

23.4 I/O 端口电路

参数		最小值	典型值	最大值	单位	条件
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	-	-3	-	mA	25°C, V _{DD} = 5V, V _{OH} = 4.5V
	L1	-	-6	-		
	L2	-	-18	-		
灌电流(sink)	L0	-	45	-	mA	25°C, V _{DD} = 5V, V _{OL} = 0.5V
	L1	-	60	-		
上拉电阻		-	20	-	kΩ	
下拉电阻		-	20	-	kΩ	

23.5 工作电流(I_{DD})

参数	SysClk	典型值 @V _{DD}			单位
		2.0V	3.0V	5.5V	
正常模式 (2T) - I _{DD}	16 MHz	-	1.681	1.843	mA
	8 MHz	0.749	1.146	1.231	
	4 MHz	0.524	0.786	0.854	
	2 MHz	0.387	0.497	0.543	
	1 MHz	0.249	0.319	0.357	
	500kHz	0.176	0.228	0.262	
	250kHz	0.139	0.183	0.221	
	32 kHz	0.024	0.031	0.032	
低功耗模式 (2T) - I _{DD}	32 kHz	0.009	0.011	0.011	μA
Sleep 模式 (WDT OFF, LVR OFF), I _{SB}	-	0.20	0.48	0.55	
Sleep 模式 (WDT ON, LVR OFF)	32 kHz	1.11	1.71	1.81	
Sleep 模式 (WDT OFF, LVR ON)	-	11.48	15.40	21.01	
Sleep 模式 (WDT ON, LVR ON)	32 kHz	17.31	20.41	24.63	
Sleep 模式 (WDT OFF, LVR OFF, LVD ON)	-	18.57	20.49	24.80	

注： Sleep 模式 I_{SB} 的测试条件为所有 I/O 设置成输入模式并外部下拉到 GND。

23.6 内部振荡器

内部低频振荡器 (LIRC)

测试条件为 LIRC 选择 32 kHz (LFMOD = 0)。

参数	最小值	典型值	最大值	单位	条件
振荡频率	30.4	32	33.6	kHz	25°C, V _{DD} = 2.5V
随温度变化范围	-4.0%	—	3.5%	—	-40 ~ 85°C, V _{DD} = 2.5V
随电源电压变化范围	-1.0%	—	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

内置高频振荡器(HIRC)

高频振荡器分为两个模块：13.56MHz、16MHz。

参数	最小值	典型值	最大值	单位	条件
13.56M 校准范围	13.42	13.56	13.69	MHz	25°C, V _{DD} = 2.5V
13.56M 随温度变化范围	-3.5%	—	3.5%	—	-40~85°C, V _{DD} = 2.5V
16M 校准范围	15.84	16	16.16	MHz	25°C, V _{DD} = 2.5V
16M 随温度变化范围	-3.5%	—	3.5%	—	-40~85°C, V _{DD} = 2.5V
随电源电压变化范围	-1.0%	—	1.0%	—	25°C, V _{DD} = 1.9 – 5.5V
I _{HIRC} 工作电流	—	136	—	μA	25°C, V _{DD} = 3.0V
启动时间	—	2.5	—	μs	25°C, V _{DD} = 3.0V

23.77bit DAC 电路 (比较器参考电压)

参数	最小值	典型值	最大值	单位	条件
DAC VREF	1.990	2	2.010	V	-40°C ~85°C, V _{DD} =2.5 – 5.5V
	2.985	3	3.015		-40°C ~85°C, V _{DD} =3.5 – 5.5V
	3.980	4	4.020		-40°C ~85°C, V _{DD} =4.5 – 5.5V
相对精度	—	VREF/128	—	V	-40°C ~85°C, V _{DD} =2.0 – 5.5V
绝对精度	—	—	1/2*	LSB	同上
单位电阻 (unit resistor)	—	5000*	—		同上

23.8 ADC(12bit) 和 ADC VREF

ADC (12bit)

参数	最小值	典型值	最大值	单位	条件
ADC 工作电压 V_{DD}	2.7	-	5.5	V	
ADC 工作电流 I_{VDD}	-	120	-	μA	$V_{REF+} = V_{DD} = 2.7V$
	-	125	-	μA	$V_{REF+} = V_{DD} = 3.0V$
	-	155	-	μA	$V_{REF+} = V_{DD} = 5.5V$
模拟输入电压 V_{AIN}	V_{REF-}	-	V_{REF+}	V	
外部参考电压 V_{REF}	-	-	V_{DD}	V	
分辨率	-	-	12	bit	
积分误差 E_{IL}	-	± 2.0	-	LSB	$V_{REF+} = V_{DD} = 5.0V$, $V_{REF-} = GND$, $F_{ADCLK} = 250kHz$
微分误差 E_{DL}	-	± 1.0	-	LSB	
偏移误差 E_{OFF}	-	± 3.0	-	LSB	$V_{REF+} = V_{DD} = 5.0V$, $V_{REF-} = GND$, $F_{ADCLK} = 250kHz$
增益误差 E_{GN}	-	± 5.0	-	LSB	
转换时钟周期 T_{AD}	-	2	-	μs	$V_{REF+} > 3.0V$, $V_{DD} > 3.0V$
转换时钟数	-	16	-	T_{AD}	
稳定时间 (T_{ST})	-	15	-	μs	
采样时间 (T_{ACQ})	-	≥ 2	-	μs	
模拟电压源阻抗 (ZAI)	-	-	10	k Ω	(推荐)

ADC V_{REF}

参数	最小值	典型值	最大值	单位	条件	
内部参考电压 $V_{ADC-REF}$	$V_{ADC-REF} = 0.5V$	0.492	0.5	0.508	V	
	$V_{ADC-REF} = 2.0V$	1.990	2	2.010	V	
	$V_{ADC-REF} = 3.0V$	2.985	3	3.015	V	
稳定时间 T_{VRINT}	$V_{ADC-REF} = 0.5V$	-	400	-	μs	
		-	600	-	μs	$C_{EXT} = 1\mu F$
	$V_{ADC-REF} = 2.0V$	-	450	-	μs	
		-	800	-	μs	$C_{EXT} = 1\mu F$
	$V_{ADC-REF} = 3.0V$	-	450	-	μs	
		-	1200	-	μs	$C_{EXT} = 1\mu F$

注:

- 除非另作说明, 典型值的测试条件为 $25^{\circ}C$, $V_{DD} = 5.0V$ 。
- C_{EXT} 为内部参考电压 $V_{ADC-REF}$ 所接的外部电容 (当 ADPREF 或 ADNREF 配置成 10 时, 参阅表 12-2)。

微分误差 DNL

typical DNL Error (LSB) @ $V_{DD} = 5V$				
V_{REF+} F_{ADCLK}	0.5	2	3	V_{DD}
250 kHz	±2.5	±1.0	±1.0	±1.0
500 kHz	±3.0	±1.0	±1.0	±1.0
1 MHz	±3.0	±1.0	±1.0	±1.0
2 MHz	±6.0	±2.0	±1.5	±1.0
4 MHz	—	—	—	±4.0

积分误差 INL

typical INL Error (LSB) @ $V_{DD} = 5V$				
V_{REF+} F_{ADCLK}	0.5	2	3	V_{DD}
250 kHz	±4.0	±3.0	±2.5	±2.0
500 kHz	±4.0	±3.0	±3.0	±2.0
1 MHz	±4.0	±3.0	±3.0	±2.0
2 MHz	±6.0	±3.5	±3.0	±2.0
4 MHz	—	—	—	±3.5

23.9 运算放大器 0 特性

参数	最小值	典型值	最大值	单位	条件
I_{VDD} 工作电流	—	70	—	μA	$V_{DD} = 5V$, 正常模式
工作电压	2.5	—	5.5	V	—
输入失调电压 V_{OS}	—	12	—	mV	未校准
共模输入范围	V_{SS}	—	$V_{DD} - 1.4$	V	$V_{DD} = 5V$
增益带宽	—	3.64	—	MHz	$R_L = 1M, C_L = 100pF$
电源电压抑制比	—	100	—	dB	$V_{DD} = 5V$
共模抑制比	—	100	—	dB	$V_{CM} = V_{DD}/2, V_{DD} = 5V$
SR 转换速率	—	1.4	—	V/μs	空载, 5V

23.10 Comparator 比较器

参数		最小值	典型值	最大值	单位	条件
I _{VDD} 工作电流	CM0	—	135	—	μA	25°C, V _{DD} = 3V
	CM1	—	105	—	μA	25°C, V _{DD} = 3V
工作电压		1.9	—	5.5	V	
共模输入电压	CM0	1.0	—	V _{DD}	V	-40°C~85°C, V _{DD} = 1.9 – 5.5V
	CM1	V _{SS}	—	V _{DD} -1.0		
输入偏移电压 (V _{OS})	CM0	—	±3	—	mV	-40°C~85°C, V _{in} = V _{DD} /2, V _{DD} = 2.0V~5.5V
	CM1	—	±3	—		
共模抑制比(CMRR)		55	—	—	dB	-40°C~85°C, V _{DD} = 1.9 – 5.5V
迟滞(Hysteresis)		—	0	—	mV	-40°C~85°C, V _{DD} = 1.9 – 5.5V
响应时间 (Response Time)	CM0	—	55	—	ns	正常模式: 输出低->高
		—	50	—	ns	正常模式: 输出高->低
	CM1	—	55	—	ns	正常模式: 输出低->高
		—	40	—	ns	正常模式: 输出高->低
稳定时间(Settle Time)		—	—	10	μs	00000 -> 11111

23.11 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}	Data EE 写时间	-	3.0	-	ms	关闭自动擦除
		-	1.0	-		使能自动擦除
I _{PROG}	Data EE 编程电流	-	150	-	μA	25 °C, V _{DD} = 3 V

23.12 EMC 特性

ESD

参数		最小值	典型值	最大值	单位	条件
V _{ESD}	HBM	8000	-	-	V	MIL-STD-883H Method 3015.8
V _{ESD}	MM	200	-	-	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

24. 特性图

注： 特性图基于特性值，仅供参考，未经生产测试。

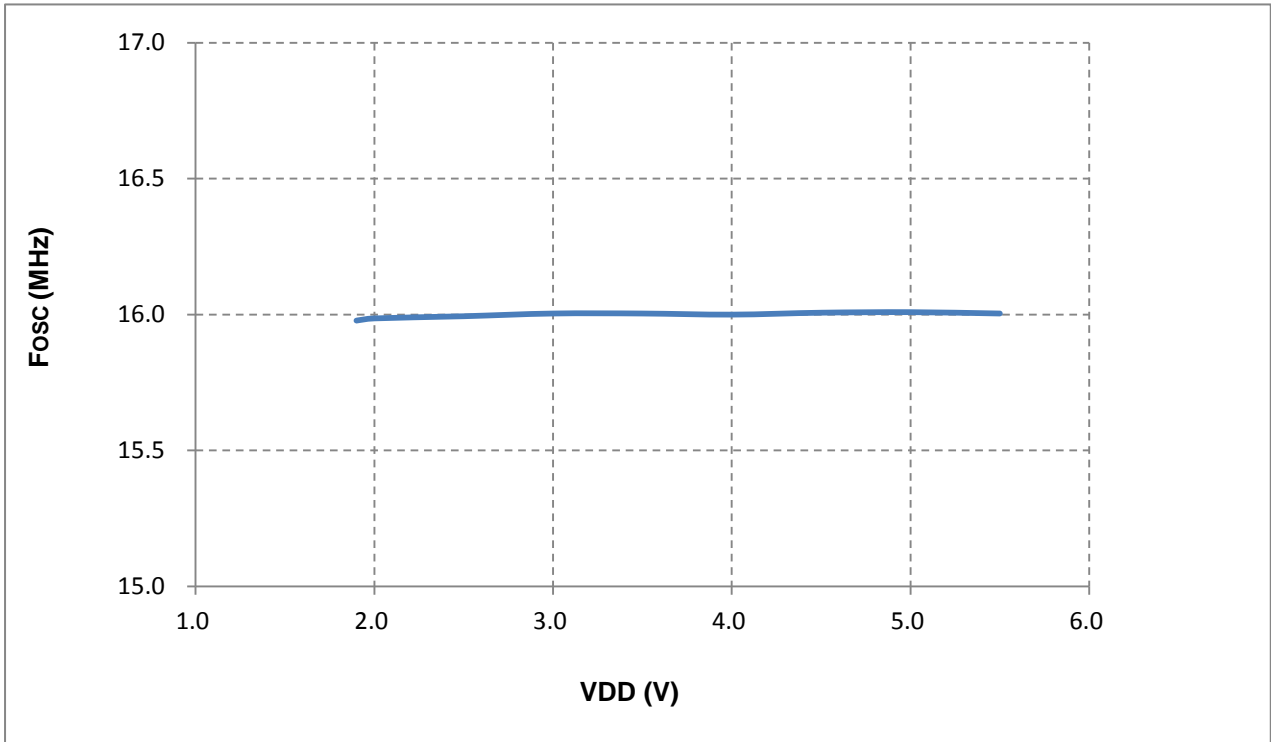


图 24-1 HIRC (16M) vs. V_{DD} (T_A = 25°C)

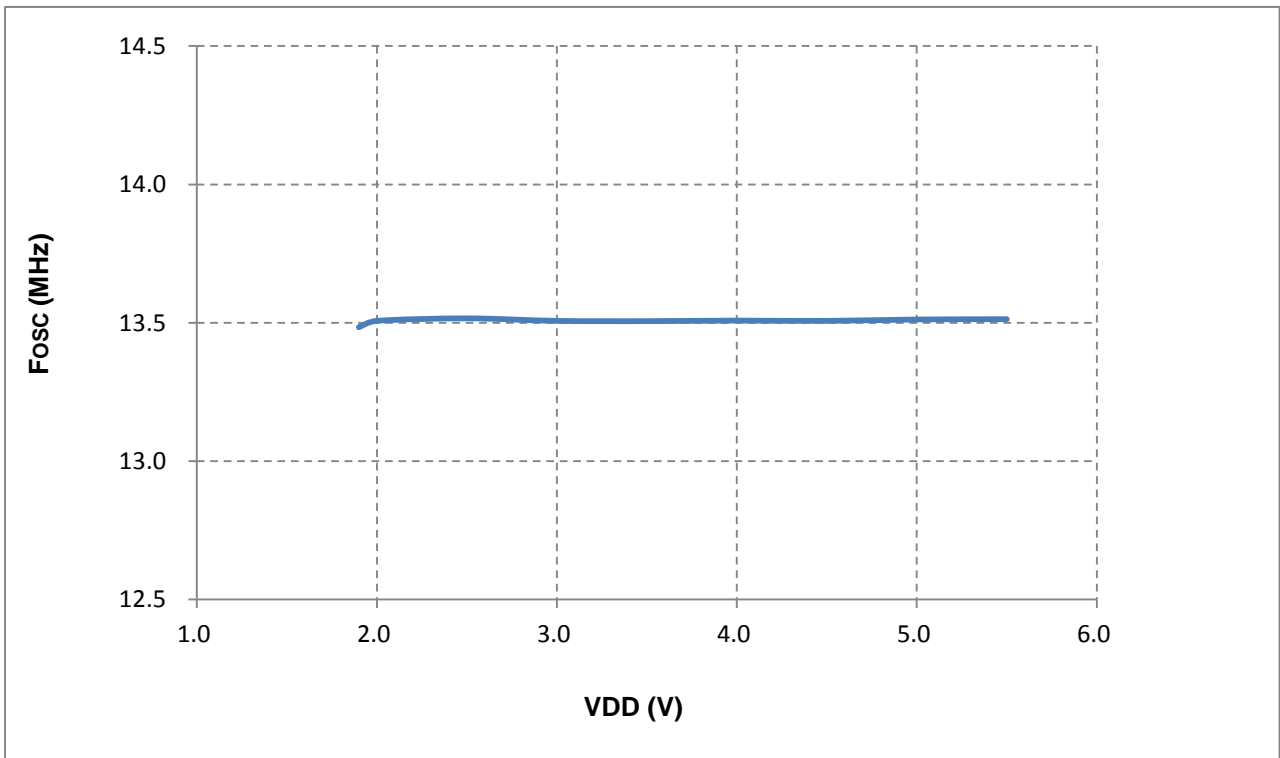


图 24-2 HIRC (13.56M) vs. V_{DD} (T_A = 25°C)

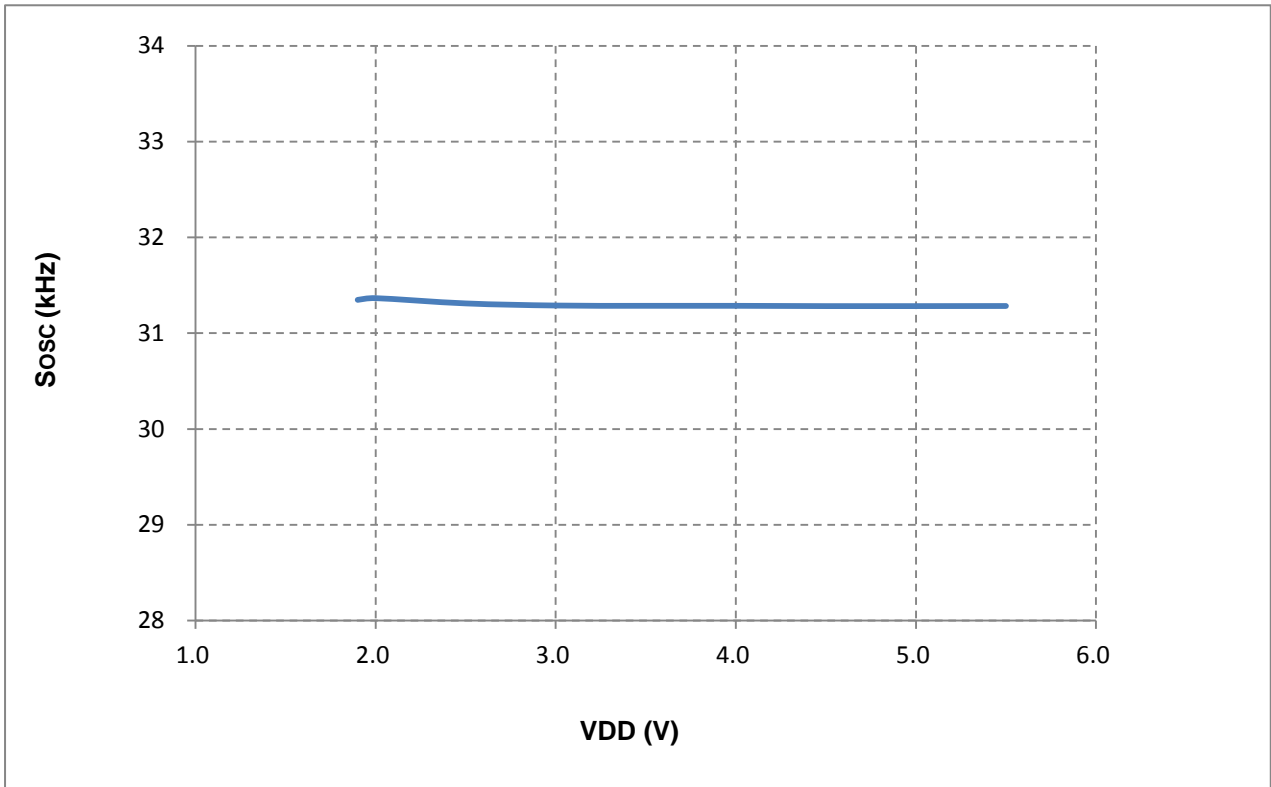


图 24-3 LIRC vs. V_{DD} (T_A = 25°C)

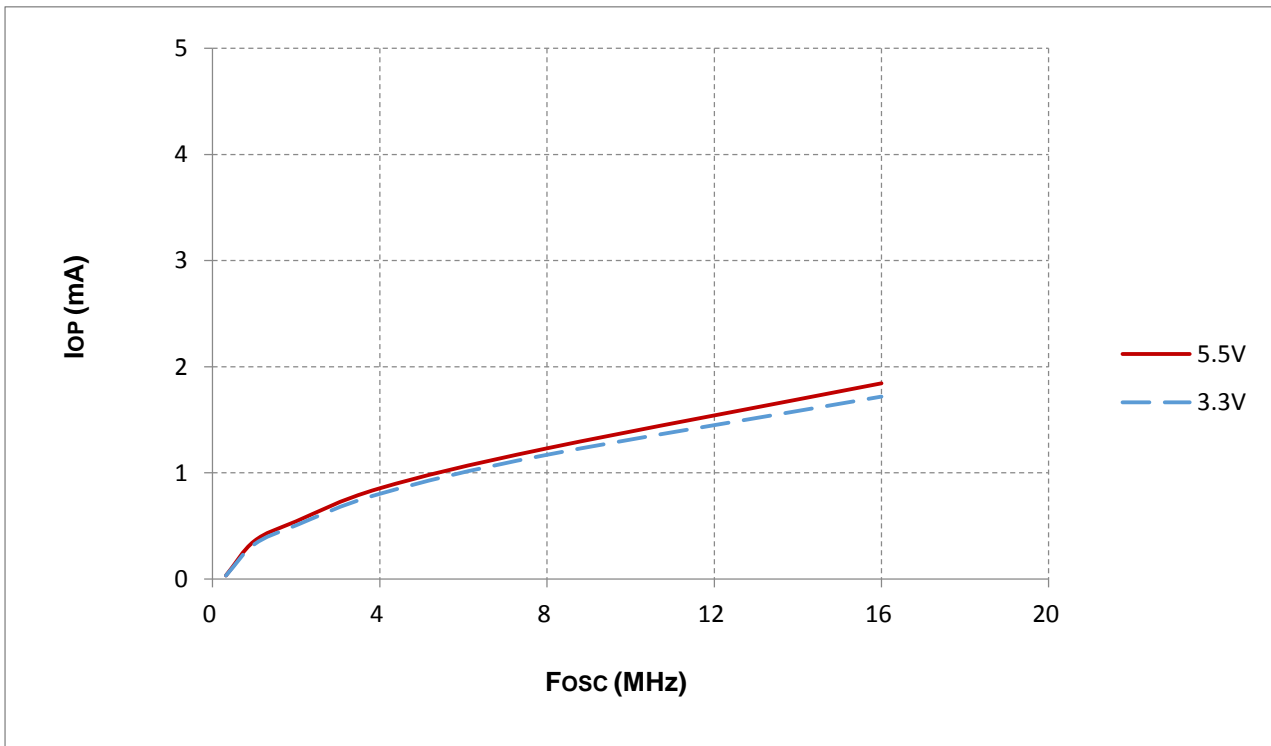


图 24-4 I_{DD} vs. Frequency (2T, T_A = 25°C)

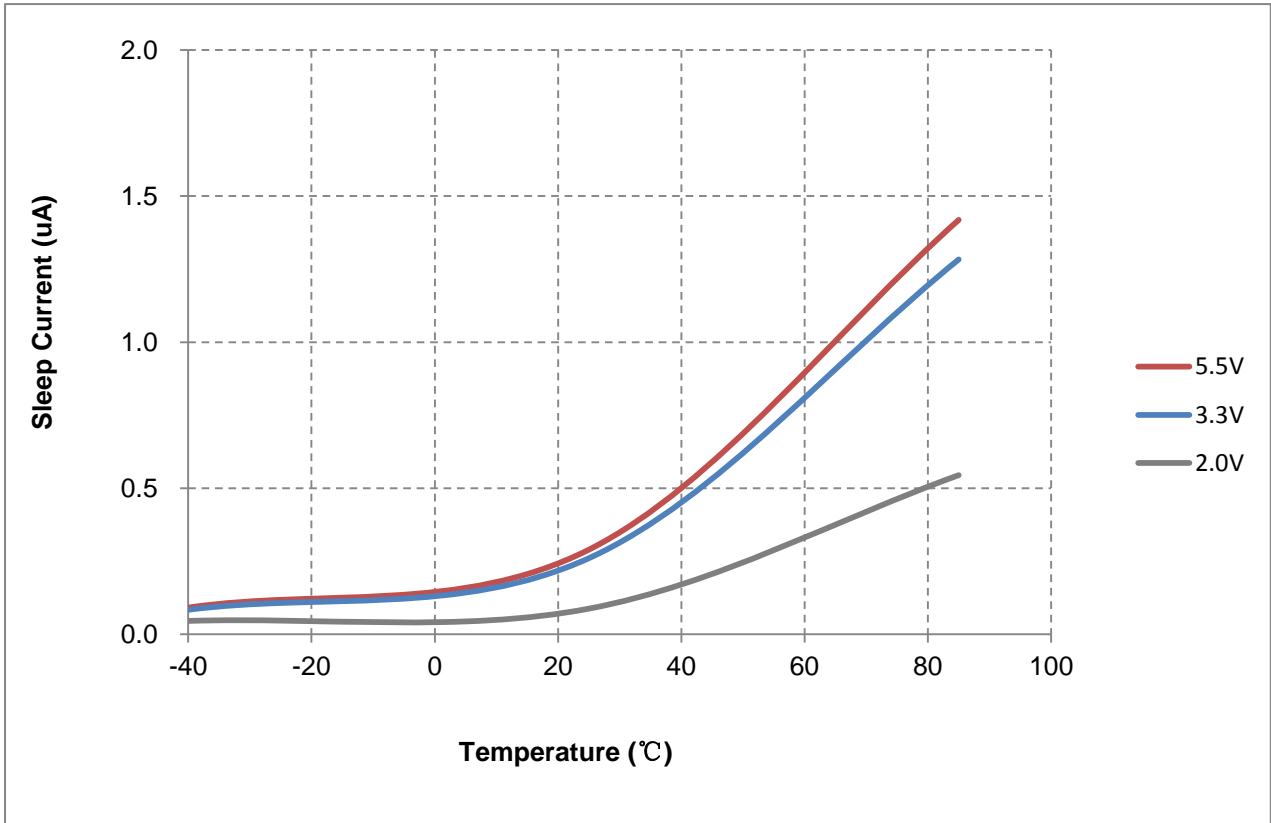


图 24-5 Sleep Current (I_{SB}) vs. Temperature

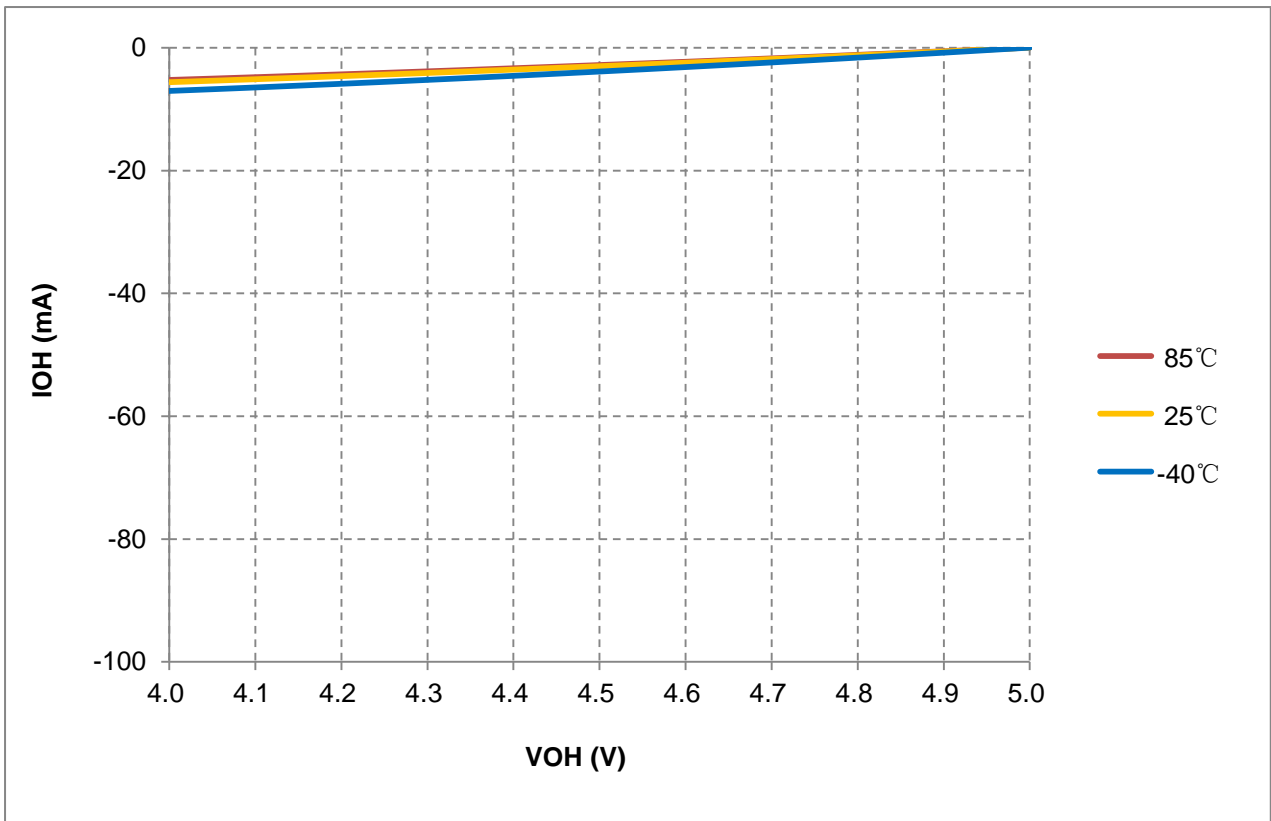


图 24-6 I_{OH} vs. V_{OH} @ $L0 = -3mA$, $V_{DD} = 5V$

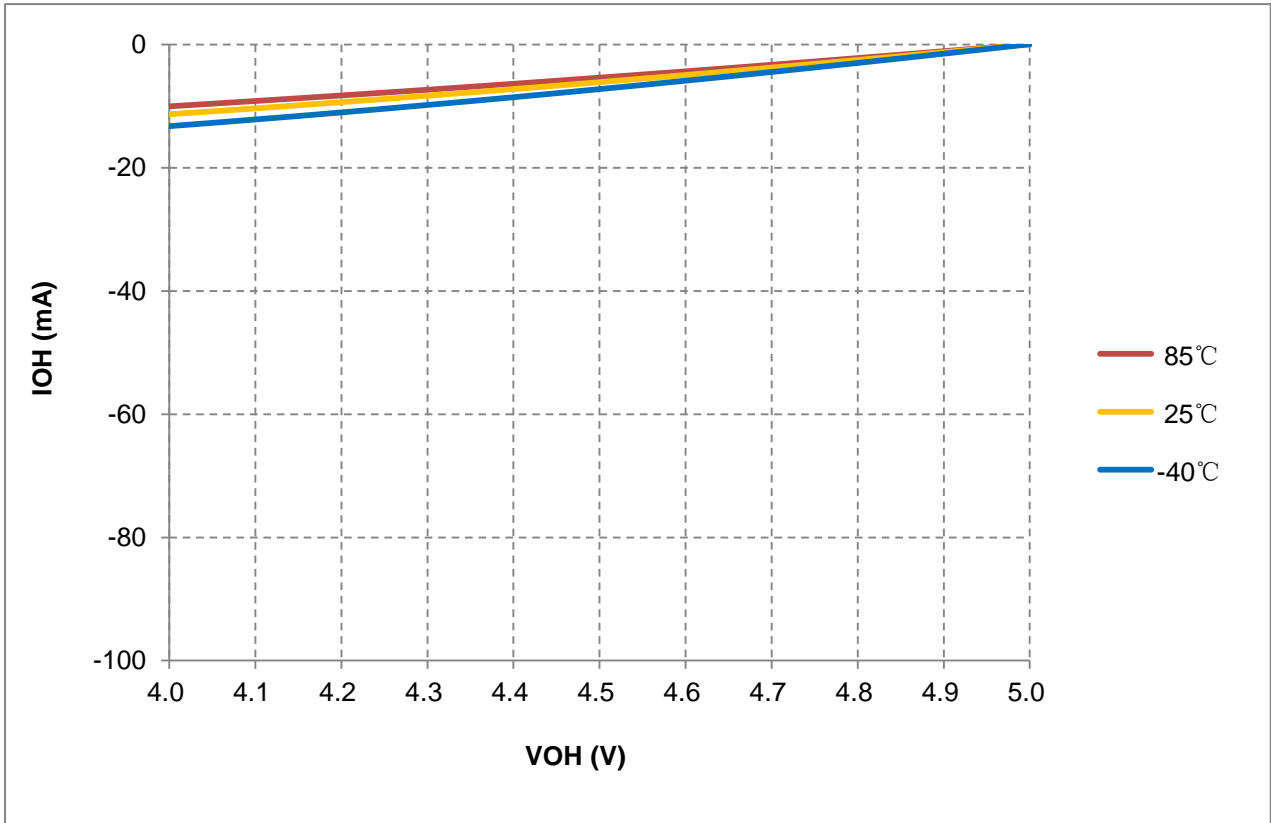


图 24-7 IOH vs. VOH @L1 = -6mA, VDD = 5V

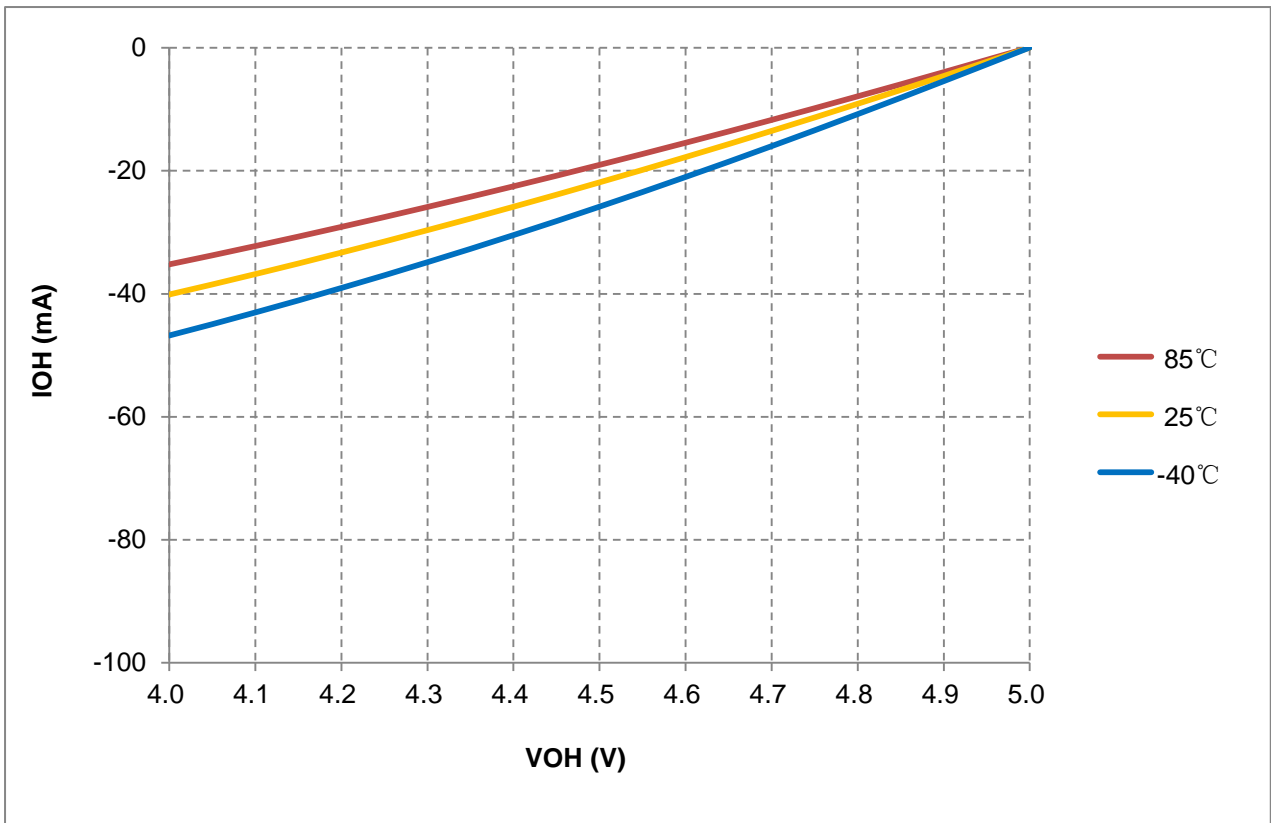


图 24-8 IOH vs. VOH @L2 = -18mA, VDD = 5V

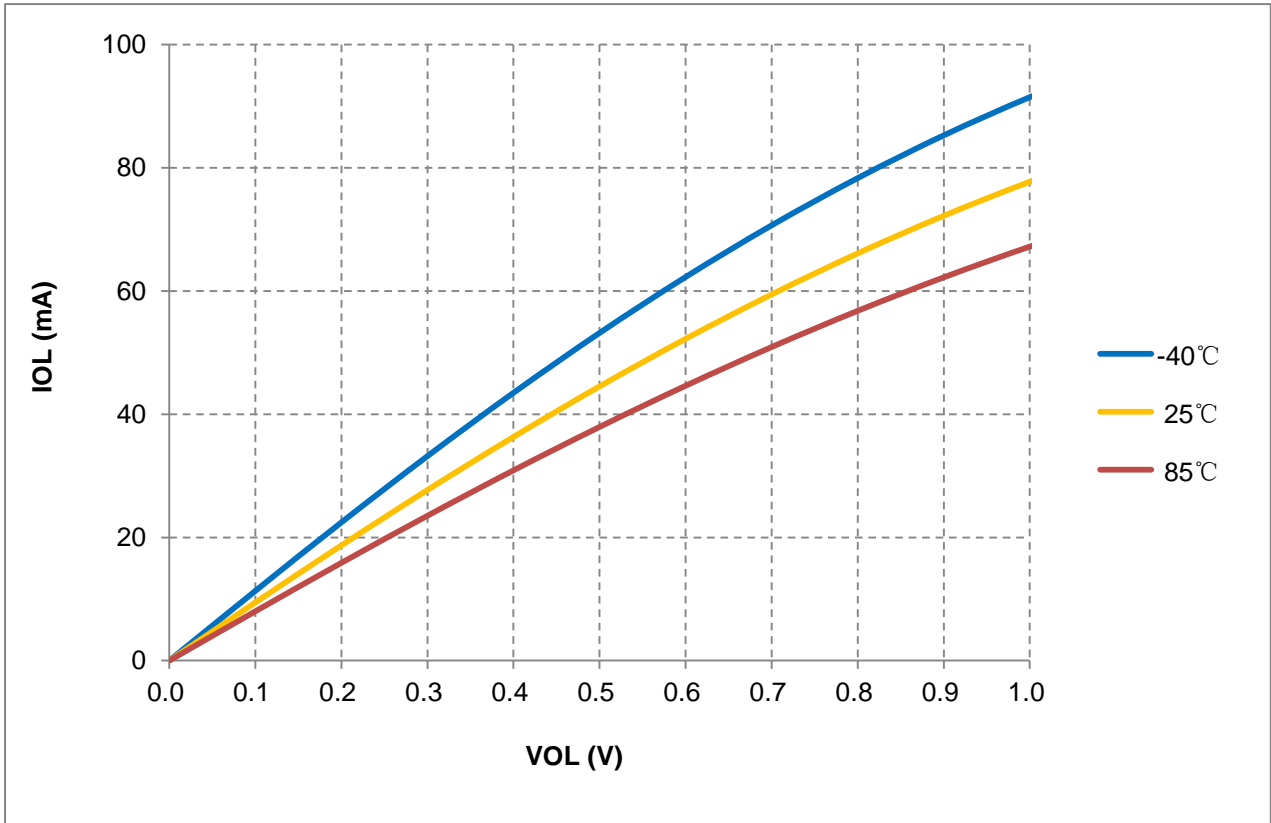


图 24-9 I_{OL} vs. V_{OL} @L0 = 45mA, V_{DD} = 5V

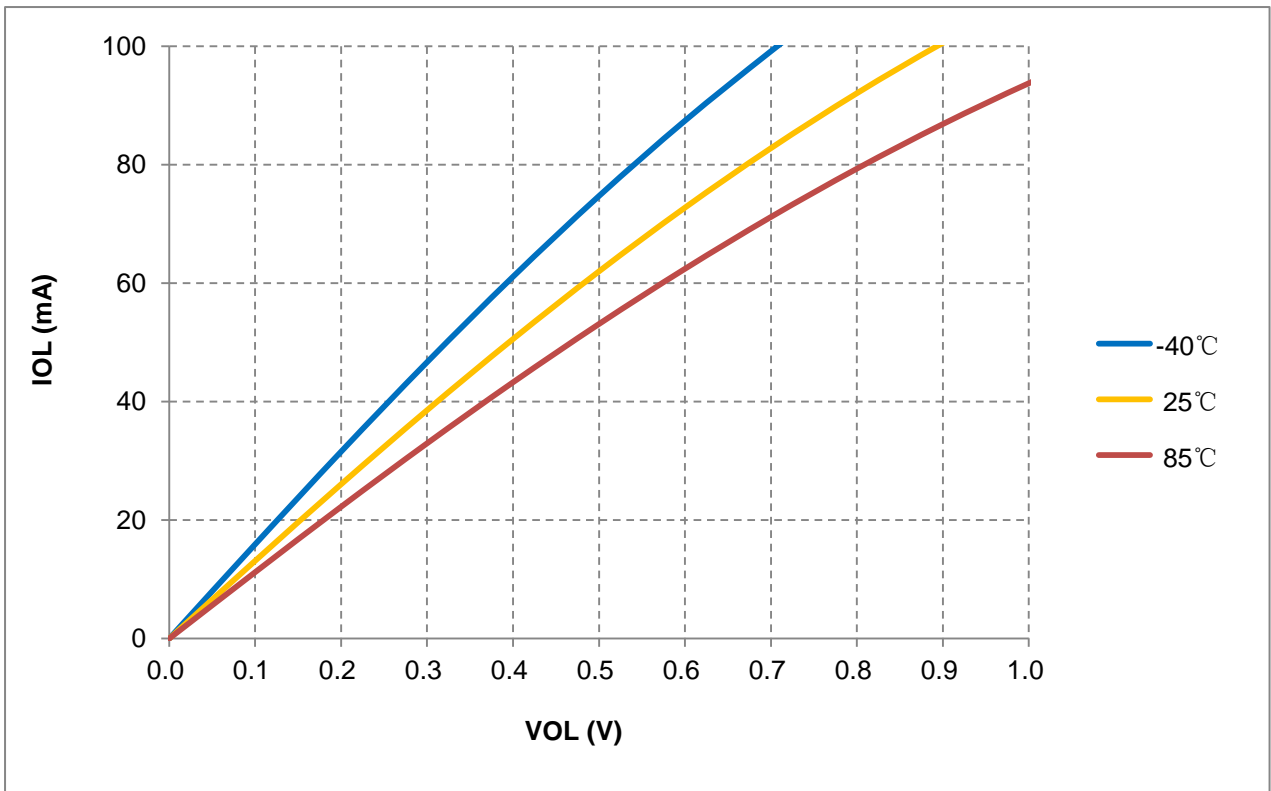
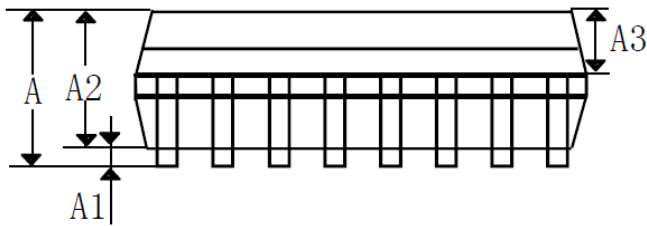
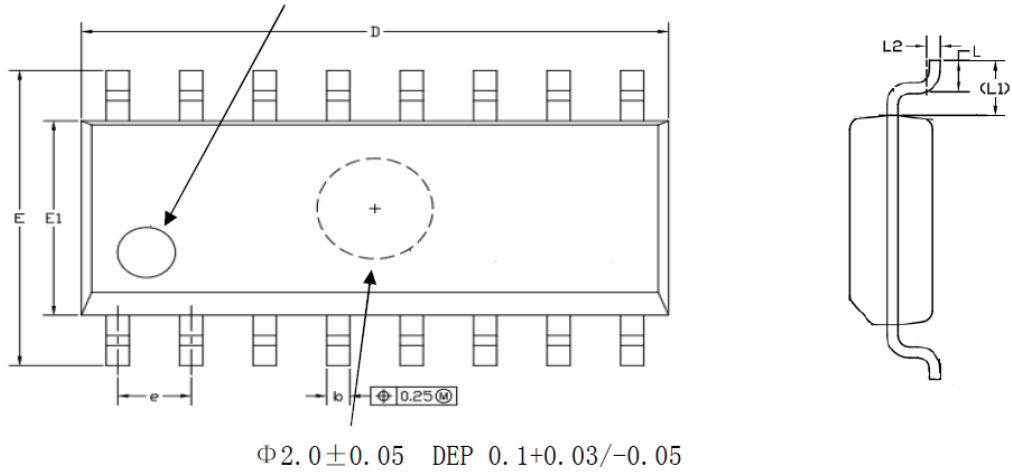


图 24-10 I_{OL} vs. V_{OL} @L1 = 60mA, V_{DD} = 5V

25. 芯片封装信息

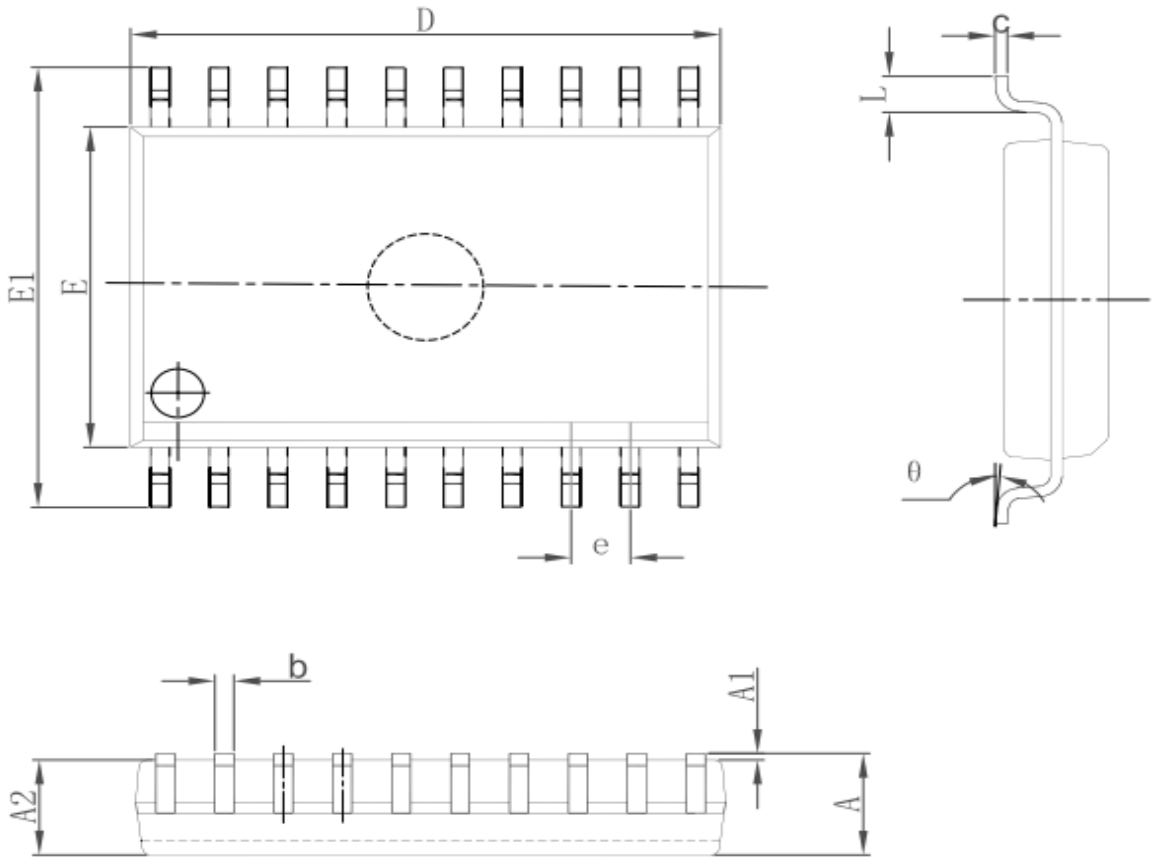
本芯片的封装形式有 SOP16, SOP20, TSSOP20, TSSOP24, SOP28, LQFP32 和 QFN32 封装。
具体封装尺寸信息如下：

SOP16



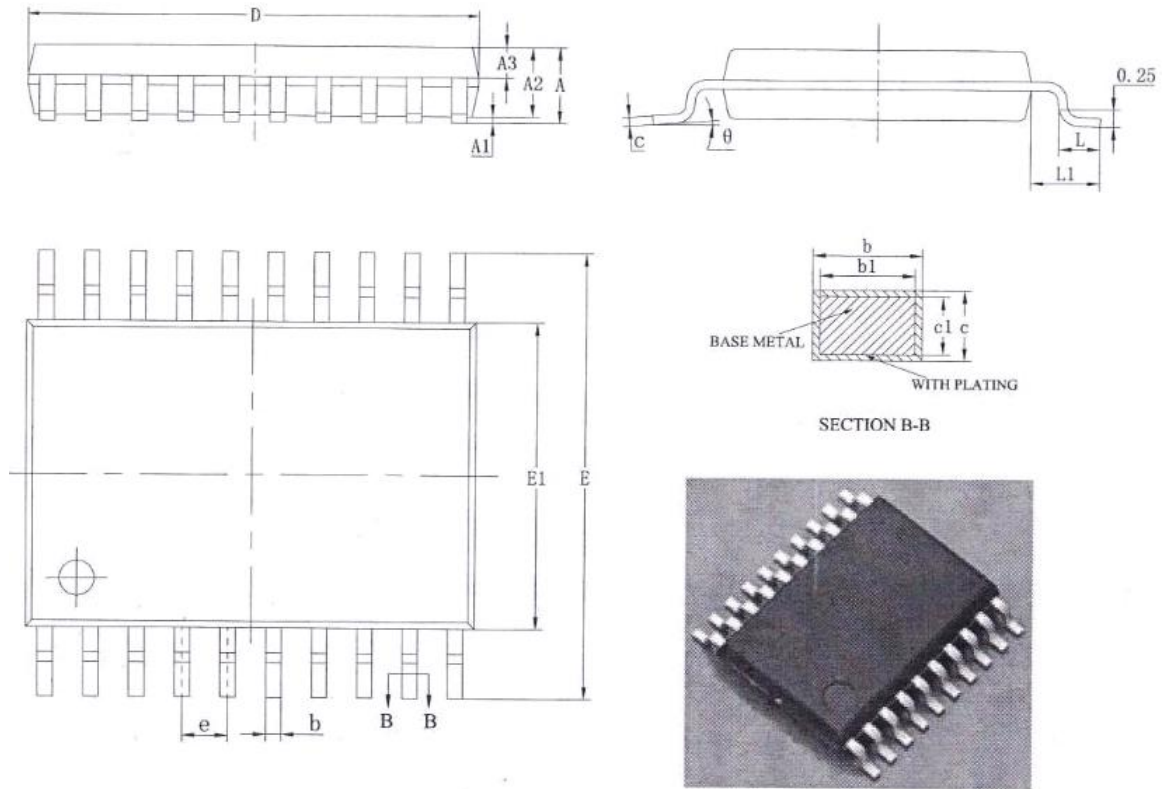
Symbol	Dimensions (mm)		Dimensions (inches)	
	Min	Max	Min	Max
A	-	1.700	-	0.066
A1	0.100	0.200	0.004	0.008
A2	1.420	1.480	0.056	0.058
A3	0.620	0.680	0.024	0.027
D	9.960	10.160	0.392	0.396
E	5.900	6.100	0.232	0.238
E1	3.870	3.930	0.152	0.153
b	0.370	0.430	0.015	0.017
e	1.240	1.300	0.048	0.051
L	0.500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

SOP20



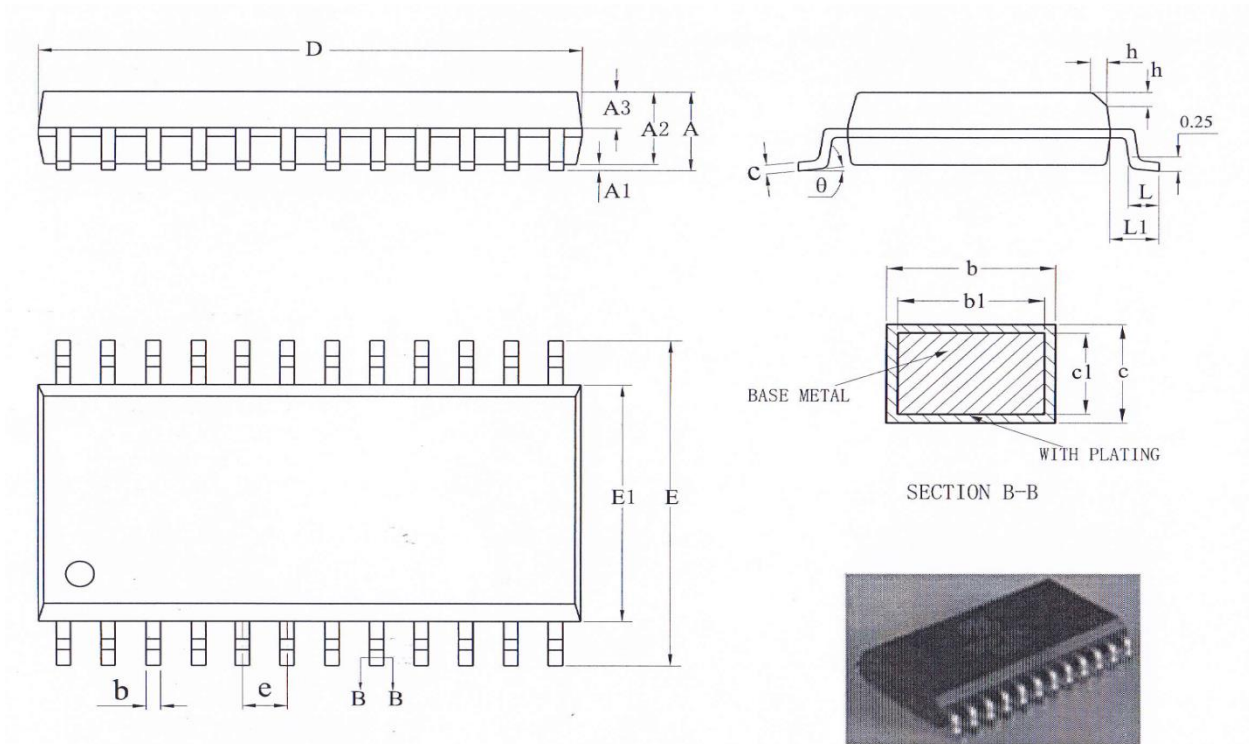
Symbol	Dimensions (mm)		Dimensions (inches)	
	Min	Max	Min	Max
A	2.350	2.650	0.093	0.104
A1	0.100	0.300	0.004	0.012
A2	2.100	2.500	0.083	0.098
b	0.330	0.510	0.013	0.020
c	0.204	0.330	0.008	0.013
D	12.520	13.000	0.493	0.512
E	7.400	7.600	0.291	0.299
E1	10.210	10.610	0.402	0.418
e	1.270 (BSC)		0.050 (BSC)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

TSSOP20



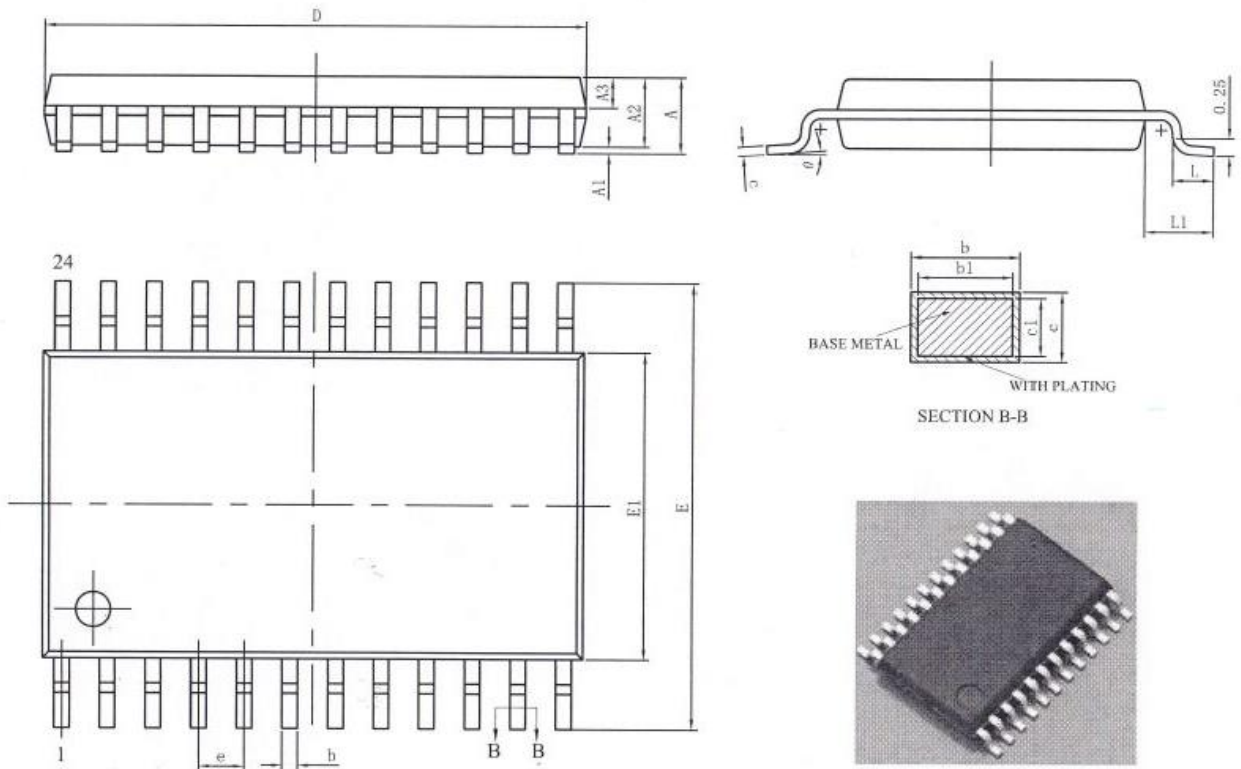
Symbol	Dimensions (mm)		Dimensions (inches)	
	Min	Max	Min	Max
A	-	1.20	-	0.472
A1	0.05	0.15	0.020	0.059
A2	0.80	1.05	0.315	0.413
A3	0.39	0.49	0.154	0.193
b	0.20	0.28	0.079	0.110
b1	0.19	0.25	0.075	0.098
c	0.13	0.17	0.051	0.067
c1	0.12	0.14	0.047	0.055
D	6.40	6.60	2.520	2.598
E1	4.30	4.50	1.693	1.771
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°

SOP24



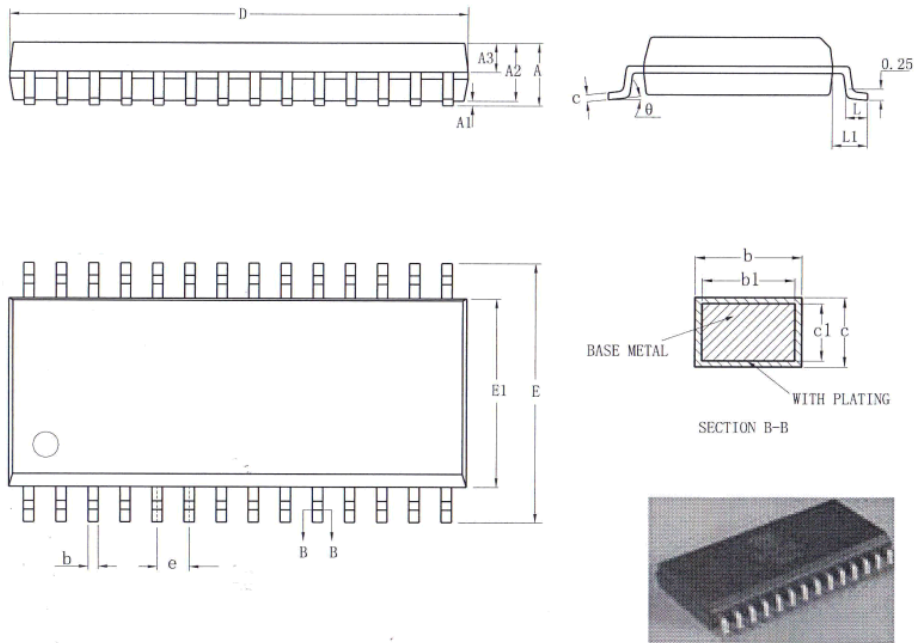
Symbol	Dimensions (mm)		
	Min	Nom	Max
A	2.36	2.54	2.64
A1	0.10	0.20	0.30
A2	2.26	2.30	2.35
A3	0.97	1.02	1.07
b	0.39	—	0.47
b1	0.38	0.41	0.44
c	0.25	—	0.29
c1	0.24	0.25	0.26
D	15.30	15.40	15.50
E	10.10	10.30	10.50
E1	7.40	7.50	7.60
e	1.27BSC		
L	0.70	—	1.00
L1	1.40REF		
h	0.25	—	0.75
theta	0	—	8°

TSSOP24



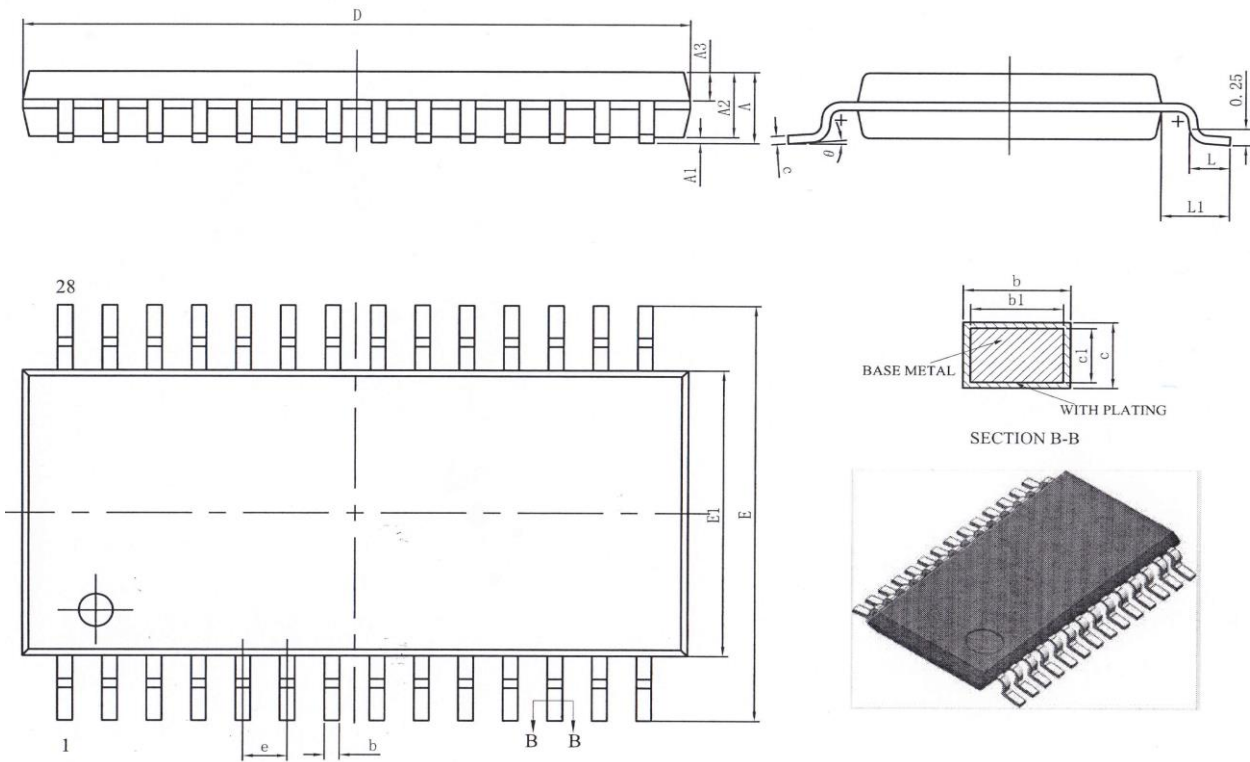
Symbol	Dimensions (mm)		
	Min	Nom	Max
A	—	—	1.20
A1	0.05	—	0.15
A2	0.80	1.00	1.05
A3	0.39	0.44	0.49
b	0.20	—	0.29
b1	0.19	0.22	0.25
c	0.13	—	0.18
c1	0.12	0.13	0.14
D	7.70	7.80	7.90
E	6.20	6.40	6.60
E1	4.30	4.40	4.50
e	0.65BSC		
L	0.45	0.60	0.75
L1	1.00BSC		
θ	0	—	8°

SOP28



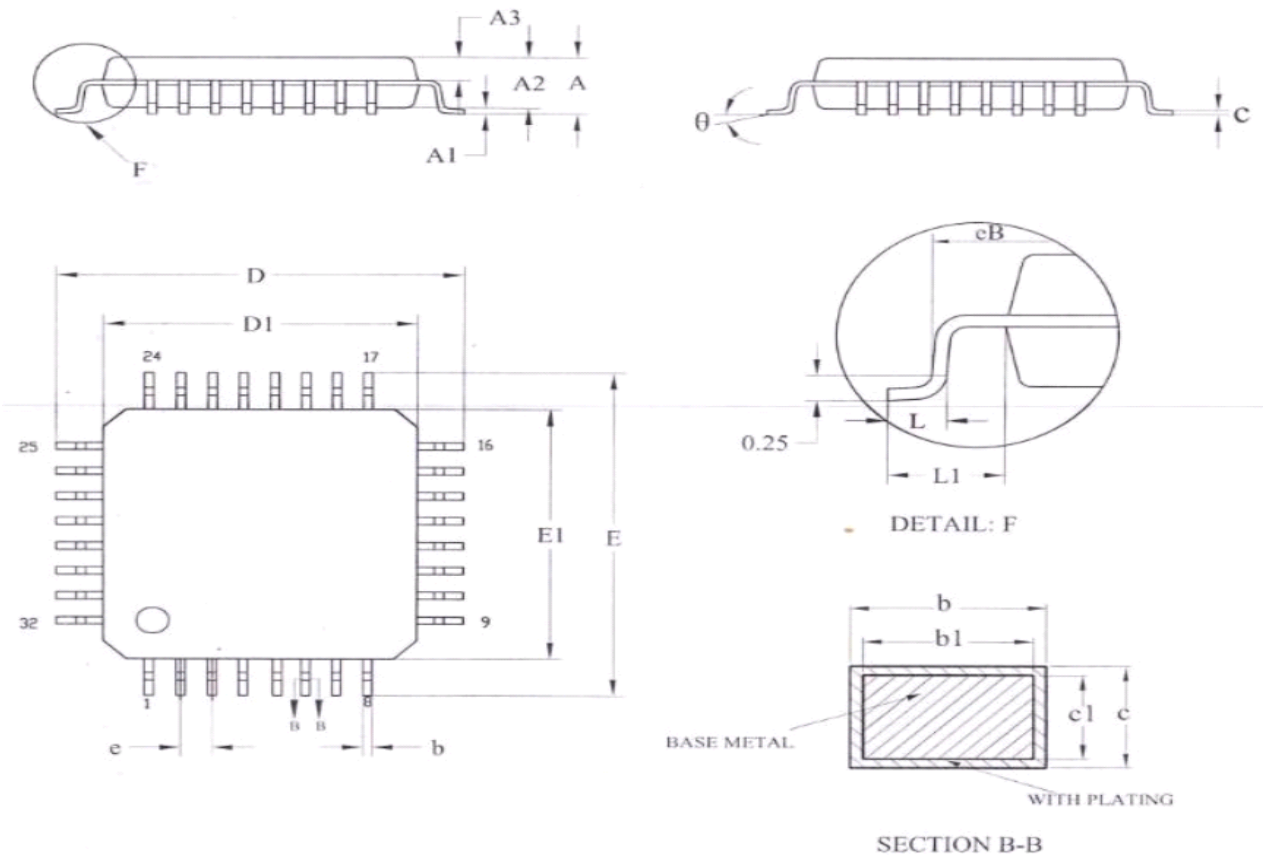
Symbol	Dimensions (mm)		Dimensions (inches)	
	Min	Max	Min	Max
A	-	2.65	-	0.104
A1	0.10	0.30	0.004	0.012
A2	2.25	2.35	0.089	0.093
A3	0.97	1.07	0.038	0.042
b	0.39	0.47	0.015	0.019
b1	0.38	0.44	0.015	0.017
c	0.25	0.29	0.010	0.011
c1	0.24	0.26	0.009	0.010
D	17.90	18.10	0.704	0.712
E	10.10	10.50	0.397	0.413
E1	7.40	7.60	0.290	0.299
e	1.27(BSC)		0.05(BSC)	
L	0.70	1.00	0.027	0.039
L1	1.40REF		0.055REF	
θ	0	8°	0	8°

TSSOP28



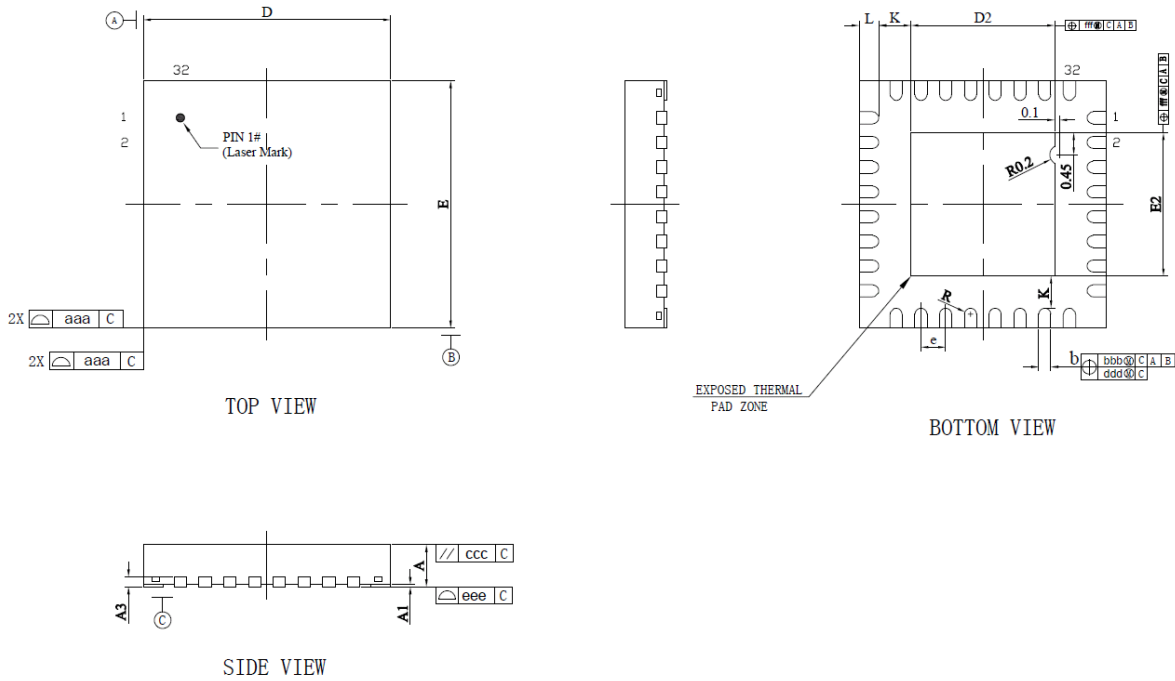
Symbol	Dimensions (mm)		
	Min	Nom	Max
A	-	-	1.20
A1	0.05	-	0.15
A2	0.80	-	1.00
A3	0.39	0.44	0.49
b	0.20	-	0.29
b1	0.19	0.22	0.25
c	0.14	-	0.18
c1	0.12	0.13	0.14
D	9.60	9.70	9.80
E	6.20	6.40	6.60
E1	4.30	4.40	4.50
e	0.65BSC		
L	0.45	0.60	0.75
L1	1.00BSC		
θ	0	-	8°

LQFP32



Symbol	Dimensions (mm)		Dimensions (inches)	
	Min	Max	Min	Max
A	-	1.60	-	0.063
A1	0.05	0.15	0.002	0.006
A2	1.35	1.45	0.053	0.057
A3	0.59	0.69	0.023	0.027
b	0.33	0.41	0.013	0.016
b1	0.32	0.38	0.013	0.015
c	0.13	0.17	0.005	0.006
c1	0.12	0.14	0.005	0.006
D	8.80	9.20	0.346	0.362
D1	6.90	7.10	0.272	0.280
E	8.80	9.20	0.346	0.362
E1	6.90	7.10	0.272	0.280
eB	8.10	8.25	0.319	0.324
e	0.80(BSC)		0.031(BSC)	
L	0.45	0.75	0.018	0.030
L1	1.00REF		0.039REF	
theta	0	7°	0	7°

QFN32



Symbol	Dimensions (mm)		Dimensions (inches)	
	Min	Max	Min	Max
A	0.80	0.90	0.031	0.035
A1	0	0.05	0.000	0.002
b	0.20	0.30	0.008	0.012
A3	0.203REF		0.008REF	
D	4.90	5.10	0.193	0.201
D2	2.80	3.00	0.110	0.118
e	0.50BSC		0.020BSC	
E	4.90	5.10	0.193	0.201
E2	2.80	3.00	0.110	0.118
L	0.35	0.45	0.014	0.018
R	0.075	0.175	0.003	0.007
K	0.65REF		0.026REF	
aaa	0.10		0.004	
bbb	0.07		0.003	
ccc	0.10		0.004	
ddd	0.05		0.002	
eee	0.08		0.003	
fff	0.10		0.004	

26. 附录：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

联系信息**Fremont Micro Devices 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) Limited

#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 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 Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. Fremont Micro Devices Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of Fremont Micro Devices Corporation. The FMD logo is a registered trademark of Fremont Micro Devices Corporation. All other names are the property of their respective owners.