

## 8 位微控制器

# KF8A100GQT\_QP 数据手册

## 产品订购信息

芯片型号	订货号	封装	FLASH (Byte)	RAM(Byte)	DEE(Byte)	内部 HFOOSC	外部 HF/LFOOSC	8位定时器	16位定时器	PWM6/7	增强型 PWM5	增强型 PWM3/4	12位 ADC通道	比较器	SSCI	USART (LIN)	USART(SSO7816/LIN)	RTC	指令系统
KF8A100	KF8A100GQT	LQFP64	64K	4K	256	16M	16M/ 32.768K	1	10	4路输出*2	1组全桥	1组全桥*2	43	4	2	1	2	Y	V2
KF8A100	KF8A100GQP	LQFP32	64K	4K	256	16M	16M/ 32.768K	1	10	3路输出*2	1组全桥	1组全桥*1	23	3	1	N	2	Y	V2

## 文档说明

### 简介

本文档为 KF8A100 系列单片机芯片数据手册。该系列不同的型号拥有不同的芯片封装和资源，具体型号对应的封装及资源情况见对应的数据手册。

更多详细信息请访问 ChipON 官网 [www.chipon-ic.com](http://www.chipon-ic.com)。

## KF8A100 芯片使用注意事项

### 芯片的 ESD 防护措施

KF8A100 芯片提供高达 8KV 的 ESD 标准保护电路。建议用户根据芯片存储/应用的环境采取适当静电防护措施。应注意应用环境的湿度；建议避免使用容易产生静电的绝缘体；存放和运输应在抗静电容器、防静电屏蔽袋或导电材料容器中；包括工作台在内的所有测试和测量工具必须保证接地；操作者应该佩戴静电消除手腕环手套，不能用手直接接触芯片等。

### 芯片的 EFT 防护措施

KF8A100 芯片提供高达 4.2KV 的 EFT 标准的保护电路。当 MCU 芯片应用在 PCB 系统时，需要遵守 PCB 相关设计要求，包括电源线、地线（包括数字/模拟电源分离，单点/多点接地等）、复位管脚保护电路、电源和地之间的去耦电容、高低频电路单独分别处理以及单/多层板选择等。

### 芯片的 LATCH-UP 防护措施

为有效防护 LATCH-UP 损坏芯片，用户需保证在 VDD 引脚上不出现异常高压或者负压。建议用户在 VDD 和 VSS 之间并接两个 105 和 102 大小的电容，电容尽量靠近芯片的 VDD 引脚。

### 芯片的焊接

KF8A100 芯片的焊接应按照工业标准的焊接要求，以免损坏芯片。手工焊接时注意焊接的温度和焊接时间。

### 芯片的上电/断电

KF8A100 芯片提供独立电源管脚。当 KF8A100 芯片应用在多电源供电系统时，应先对 MCU 芯片上电，再对系统其他部件上电；反之，断电时，先对系统其他部件断电，再对 MCU 芯片断电。若操作顺序相反则可能导致芯片内部元件过压或过流，从而导致芯片故障或元件退化。

### 芯片的复位

KF8A100 芯片提供内部上电复位。用户可选择使用外部复位、断电复位、看门狗复位等，确保复位电路正常工作。在系统设计时，若使用外部复位电路，建议采用三极管复位电路、RC 复位电路。若不使用外部复位电路，建议采用复位管脚接电阻到电源，或采取必要的电源抖动处理电路或其他保护电路。具体可参照芯片的数据手册说明。

### 芯片的内部时钟

KF8A100 芯片提供内部时钟源。内部时钟源会随着温度、电压变化而偏移，可能会影响时钟源精度。具体可参照芯片的数据手册说明。

### 芯片的初始化

KF8A100 芯片提供各种内部和外部复位。对于不同的应用系统，有必要对芯片寄存器、内存、功能模块等进行初始化，尤其是 I/O 管脚复用功能进行初始化，避免由于芯片

上电以后，I/O 管脚状态的不确定情况发生。

### 芯片的管脚

KF8A100 芯片提供宽范围的输入管脚电平，用户输入高电平应大于  $V_{IH}$  的最小值，低电平应小于  $V_{IL}$  的最大值，以免波动噪声进入芯片。对于未使用的输入/输出管脚，建议用户设为输入状态，并通过电阻上拉至电源或下拉至地，或设置为输出管脚，输出固定电平并浮空。对未使用的管脚处理因应用系统而异，具体遵循应用系统的相关规定和说明。

VDD 和 VSS 之间需接 104 以上的电容，电容尽量靠近 MCU 芯片的 VDD 引脚。

### 芯片的低功耗设计

KF8A100 芯片提供低功耗设计模式，用户在实际应用中可根据应用系统的要求采用各种不同的低功耗模式，包括系统工作时钟的选择和休眠模式的选择等等。

### 芯片的开发环境

KF8A100 芯片提供完整的软/硬件开发环境，并受知识产权保护。选择上海芯旺微电子技术有限公司指定的汇编器、编译器、编程器、硬件仿真器开发环境，必须遵循与芯片相关的规定和说明。

## 寄存器操作缩写说明

以下为本文中寄存器操作使用的字母缩写含义。

read/write (R/W):	软件可读写该位。
read-only (R):	软件只能读该位。
write-only (W):	软件只能写该位，读该位时返回默认值。
unrealized (U):	未实现
uncertain (x):	该位初始状态不确定。

## 寄存器内容及操作说明

寄存器: XTALCAL:外部低频校准寄存器(地址:362H)

复位值	bit7							bit0
0101 0000	-	-	-	-	-	-	-	-
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BIT<5:3>: 外部低频的功耗档位选择位

BIT<5>: 62nA 偏置电流档位

BIT<4>: 45nA 偏置电流档位

BIT<3>: 27nA 偏置电流档位

000 = 偏置电流不开, XTAL 不工作

...

111 = 偏置电流 134nA 档位

注: XTALCAL<5:3>档位应选择最大, 即“111”。偏置电流越大, 外部低频时钟越容易起振。

如上图所示为寄存器描述的一般形式, 包含寄存器名称、功能、地址、复位后的默认值及各 bit 位 R/W 属性。

配置寄存器时须注意: 对功能没有明确描述, 具有读写操作属性的 bit 位, 配置时须保持其默认值。如该例中 XTALCAL 寄存器, 其 bit6、bit4 复位默认值为 1, bit6 无具体描

述，bit4 为偏置电流档位，配置时，bit6 需要保持其复位默认值，bit4 根据实际需要进行配置即可。若需配置偏执电流档位为 134nA, 则 XTALCAL 应配置为 “01111000” 即 0x78。

引脚示意图

LQFP64:

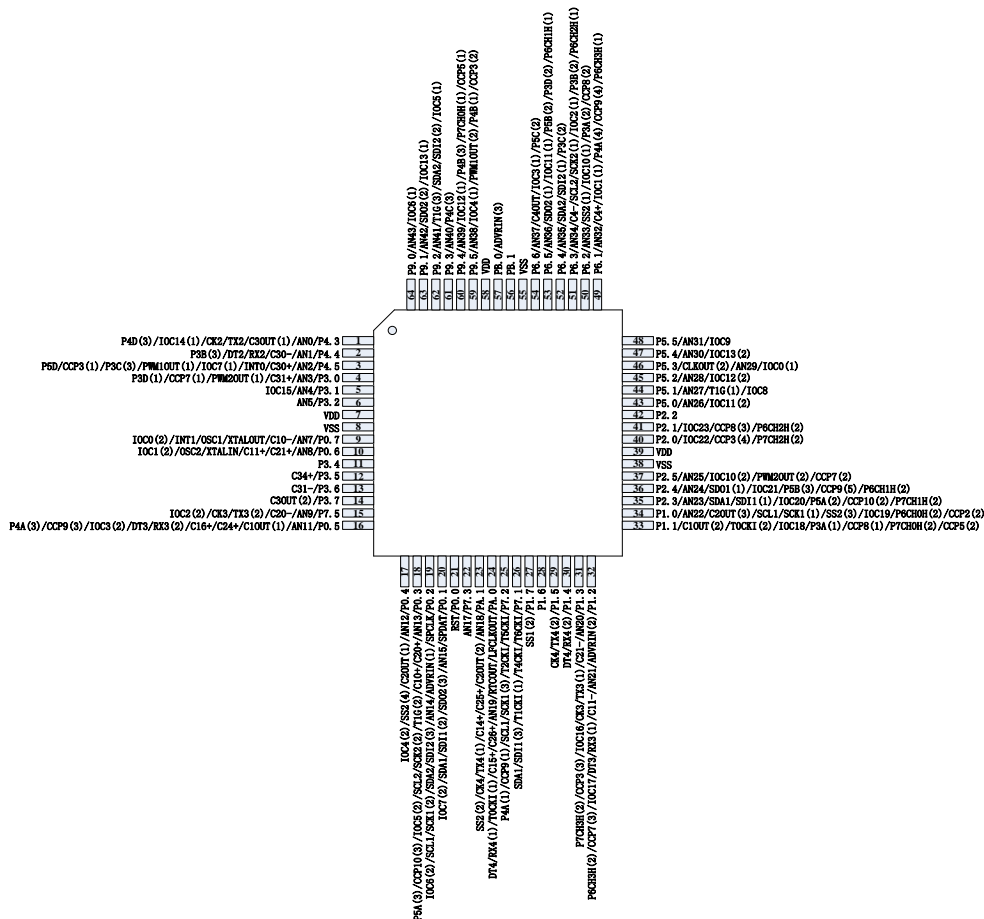


图 1 KF8A100-LQFP64 引脚图

LQFP32:

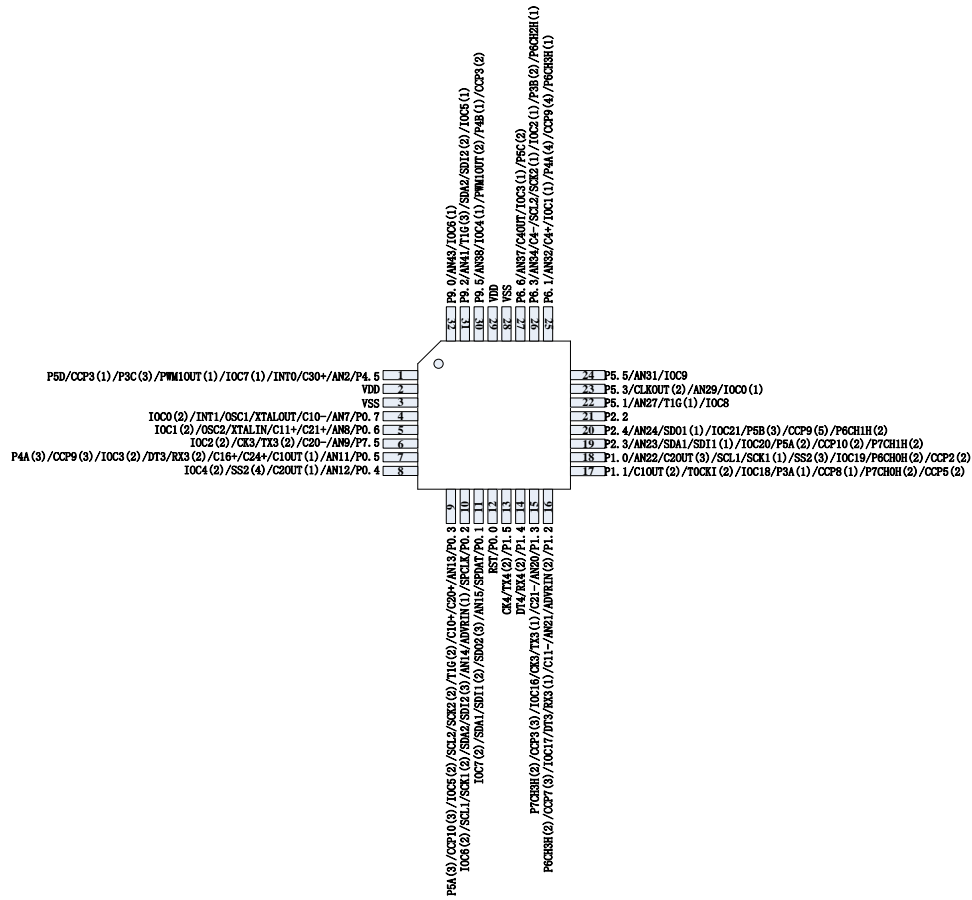


图 2 KF8A100-LQFP32 引脚图

## 引脚功能说明

封装	引脚	PWM			IOC	INT	SSCI(I2C/SPI)	UART	外部振荡器	TIMER	比较器	AD	RST&PROGRAM&其他
32 pin	64 Pin	PWM6/PWM7	PWM3/PWM4/PWM5/T3/T7	PWM1/PWM2									
12	21	P0.0											RST
11	20	P0.1			IOC7(2)		SDA1/SDI1(2)/SDO2(3)					AN15	SPDAT
10	19	P0.2			IOC6(2)		SCL1/SCK1(2)/SDA2/SDI2(3)					AN14/ADVRIN(1)	SPCLK
9	18	P0.3	P5A(3)/CCP10(3)		IOC5(2)		SCL2/SCK2(2)			TIG(2)	C10+/C20+	AN13	
8	17	P0.4			IOC4(2)		SS2(4)				C2OUT(1)	AN12	
7	16	P0.5	P4A(3)/CCP9(3)		IOC3(2)			DT3/RX3(2)			C16+/C24+/C1OUT(1)	AN11	
5	10	P0.6			IOC1(2)			OSC2/XTALIN			C11+/C21+	AN8	
4	9	P0.7			IOC0(2)	INT1		OSC1/XTALOUT			C10-	AN7	
18	34	P1.0	P6CH0H(2)/CCP5(2)		IOC19		SCL1/SCK1(1)/SS2(3)				C2OUT(3)	AN22	
17	33	P1.1	P7CH0H(2)/CCP2(2)	P3A(1)/CCP8(1)	IOC18					T0CKI(2)	C1OUT(2)		
16	32	P1.2	P6CH3H(2)	CCP7(3)	IOC17			DT3/RX3(1)			C11-	AN21/ADVRIN(2)	
15	31	P1.3	P7CH3H(2)	CCP3(3)	IOC16			CK3/TX3(1)			C21-	AN20	
14	30	P1.4						DT4/RX4(2)					
13	29	P1.5						CK4/TX4(2)					
	28	P1.6											
	27	P1.7					SS1(2)						
	40	P2.0	P7CH2H(2)	CCP3(4)	IOC22								
	41	P2.1	P6CH2H(2)	CCP8(3)	IOC23								
21	42	P2.2											
19	35	P2.3	P7CH1H(2)	P5A(2)/CCP10(2)	IOC20		SDA1/SDI1(1)					AN23	
20	36	P2.4	P6CH1H(2)	P5B(3)/CCP9(5)	IOC21		SDO1(1)					AN24	
	37	P2.5		CCP7(2)	PWM2OUT(2)	IOC10(2)						AN25	
	4	P3.0		P3D(1)/CCP7(1)	PWM2OUT(1)						C31+	AN3	
	5	P3.1				IOC15						AN4	
	6	P3.2										AN5	
	11	P3.4											
	12	P3.5									C34+		
	13	P3.6									C31-		
	14	P3.7									C3OUT(2)		
	1	P4.3		P4D(3)		IOC14(1)		CK2/TX2			C3OUT(1)	AN0	
	2	P4.4		P3B(3)				DT2/RX2			C30-	AN1	
1	3	P4.5		P5D/CCP3(1)/P3C(3)	PWM1OUT(1)	IOC7(1)	INT0				C30+	AN2	
	43	P5.0				IOC11(2)						AN26	
22	44	P5.1				IOC8				TIG(1)		AN27	
	45	P5.2				IOC12(2)						AN28	
23	46	P5.3				IOC0(1)						AN29	CLKOUT(2)
	47	P5.4				IOC13(2)						AN30	
24	48	P5.5				IOC9						AN31	
25	49	P6.1	P6CH3H(1)	P4A(4)/CCP9(4)		IOC1(1)					C4+	AN32	
	50	P6.2		P3A(2)/CCP8(2)		IOC10(1)	SS2(1)					AN33	
26	51	P6.3	P6CH2H(1)	P3B(2)		IOC2(1)	SCL2/SCK2(1)				C4-	AN34	
	52	P6.4		P3C(2)			SDA2/SDI2(1)					AN35	
	53	P6.5	P6CH1H(1)	P5B(2)/P3D(2)		IOC11(1)	SDO2(1)					AN36	



27	54	P6.6		P5C(2)		IOC3(1)					C4OUT	AN37	
	26	P7.1						SDA1/SDI1(3)		T1CKI(1)/T4CKI/T6CKI			
	25	P7.2		P4A(1)/CCP9(1)				SCL1/SCK1(3)		T2CKI/T5CKI			
	22	P7.3										AN17	
6	15	P7.5				IOC2(2)			CK3/TX3(2)		C20-	AN9	
32	64	P9.0				IOC6(1)						AN43	
	63	P9.1				IOC13(1)		SDO2(2)				AN42	
31	62	P9.2				IOC5(1)		SDA2/SDI2(2)		T1G(3)		AN41	
	61	P9.3		P4C(3)								AN40	
	60	P9.4	P7CH0H(1)/CCP5(1)	P4B(3)		IOC12(1)						AN39	
30	59	P9.5		P4B(1)/CCP3(2)	PWM1OUT(2)	IOC4(1)						AN38	
	24	PA.0						DT4/RX4(1)		T0CKI(1)	C15+/C26+	AN19	RTCOU/LFCLKOUT
	23	PA.1						SS2(2)	CK4/TX4(1)		C14+/C25+/C2OUT(2)	AN18	
	57	PB.0										ADVRIN(3)	
	56	PB.1											
2	7	VDD											
3	8	VSS											
29	39	VDD											
28	38	VSS											
	58	VDD											
	55	VSS											

1	I/O	P4.3	带弱上拉功能的双向输入输出端口
		AN0	ADC 通道 0
		C3OUT(1)	比较器 3 的输出
		CK2	USART2 半双工同步时钟端
		TX2	USART2 全双工异步数据发送端
		IOC14(1)	电平变化模块引脚输入
2	I/O	P4.4	带弱上拉功能的双向输入输出端口
		AN1	ADC 通道 1
		C30-	比较器 3 的负端输入
		RX2	USART2 全双工异步接收数据输入端
		DT2	USART2 半双工同步数据端
		P3B(3)	PWM3 输出通道
3	I/O	P4.5	带弱上拉功能的双向输入输出端口
		AN2	ADC 通道 2
		C30+	比较器 3 的正端输入
		INT0	外部中断脚 0
		IOC7(1)	电平变化模块引脚输入
		PWM1OUT(1)	PWM1 输出
		P3C(3)	PWM3 输出通道
		CCP3(1)	CCP3 的比较捕捉功能
4	I/O	P5D	PWM5 输出通道
		P3.0	带弱上拉功能的双向输入输出端口
		AN3	ADC 通道 3
		C31+	比较器 3 的正端输入
		PWM2OUT(1)	PWM2 输出
		CCP7(1)	CCP7 的捕捉比较功能
5	I/O	P3D(1)	PWM3 输出通道
		P3.1	带弱上拉功能的双向输入输出端口
		AN4	ADC 通道 4
6	I/O	IOC15	电平变化模块引脚输入
		P3.2	带弱上拉功能的双向输入输出端口
7	P	AN5	ADC 通道 5
8	P	VDD	
9	I/O	VSS	
		P0.7	带弱上拉功能的双向输入输出端口
		AN7	ADC 通道 7
		C10-	比较器 1 负端输入
		XTALOUT	外部低频振荡器输入引脚 1
		OSC1	外部振荡器晶振脚
		INT1	外部中断脚 1

		IOC0(2)	电平变化模块引脚输入
10	I/O	P0.6	带弱上拉功能的双向输入输出端口
		AN8	ADC 通道 8
		C21+	比较器 2 正端输入
		C11+	比较器 1 正端输入
		XTALIN	外部低频振荡器输入引脚 1
		OSC2	外部振荡器晶振脚
		IOC1(2)	电平变化模块引脚输入
11	I/O	P3.4	带弱上拉功能的双向输入输出端口
12	I/O	P3.5	带弱上拉功能的双向输入输出端口
		C34+	比较器 3 正端输入
13	I/O	P3.6	带弱上拉功能的双向输入输出端口
		C31-	比较器 3 的负端输入
14	I/O	P3.7	带弱上拉功能的双向输入输出端口
		C3OUT(2)	比较器 3 的输出
15	I/O	P7.5	带弱上拉功能的双向输入输出端口
		AN9	ADC 通道 9
		C20-	比较器 2 的负端输入
		CK3(2)	USART3 半双工同步时钟端
		TX3(2)	USART3 全双工异步数据发送端
		IOC2(2)	电平变化模块引脚输入
16	I/O	P0.5	带弱上拉功能的双向输入输出端口
		AN11	ADC 通道 11
		C1OUT(1)	比较器 1 的输出
		C24+	比较器 2 的正端输入
		C16+	比较器 1 的正端输入
		RX3(2)	USART3 全双工异步接收数据输入端
		DT3(2)	USART3 半双工同步数据端
		IOC3(2)	电平变化模块引脚输入
		CCP9(3)	CCP9 的比较捕捉功能
P4A(3)	PWM4 输出通道		
17	I/O	P0.4	带弱上拉功能的双向输入输出端口
		AN12	ADC 通道 12
		C2OUT(1)	比较器 2 的输出
		SS2(4)	MSSP2 模块 SPI 从动模式选择引脚
		IOC4(2)	电平变化模块引脚输入
18	I/O	P0.3	带弱上拉功能的双向输入输出端口
		AN13	ADC 通道 13
		C20+	比较器 2 正端输入
		C10+	比较器 1 正端输入
		T1G(2)	T1 门控信号
		SCK2(2)	MSSP2 的 SPI 时钟端
		SCL2(2)	MSSP2 的 I2C 时钟端

		IOC5(2)	电平变化模块引脚输入
		CCP10(3)	CCP10 的比较捕捉功能
		P5A(3)	PWM5 输出通道
19	I/O	P0.2	带弱上拉功能的双向输入输出端口
		SPCLK	编程调试时钟端
		ADVRIN(1)	AD 外部参考电压模拟输入
		AN14	ADC 通道 14
		SDI2(3)	MSSP2 的 SPI 数据输入端
		SDA2(3)	MSSP2 的 I2C 数据端
		SCK1(2)	MSSP1 的 SPI 时钟端
		SCL1(2)	MSSP1 的 I2C 时钟端
		IOC6(2)	电平变化模块引脚输入
20	I/O	P0.1	带弱上拉功能的双向输入输出端口
		SPDAT	编程调试数据端
		AN15	ADC 通道 15
		SDO2(3)	MSSP2 的 SPI 数据输出端
		SDI1(2)	MSSP1 的 SPI 数据输出端
		SDA1(2)	MSSP1 的 I2C 数据端
		IOC7(2)	电平变化模块引脚输入
21	I/O	P0.0	不带弱上拉功能的双向输入输出端口
		RST	外部复位引脚
22	I/O	P7.3	带弱上拉功能的双向输入输出端口
		AN17	ADC 通道 17
23	I/O	PA.1	带弱上拉功能的双向输入输出端口
		AN18	ADC 通道 18
		C2OUT(2)	比较器 2 的输出
		C25+	比较器 2 的正端输入
		C14+	比较器 1 的正端输入
		TX4(1)	USART4 全双工异步数据发送端
		CK4(1)	USART4 半双工同步时钟端
		SS2(2)	MSSP2 的 SPI 从动模式选择引脚
24	I/O	PA.0	带弱上拉功能的双向输入输出端口
		RTCOU	RTC 信号输出
		LFCLKOUT	外设低频时钟输出
		AN19	ADC 通道 19
		C26+	比较器 2 正端输入
		C15+	比较器 1 正端输入
		T0CKI(1)	T0 外部时钟输入引脚
		RX4(1)	USART4 全双工异步接收数据输入端
		DT4(4)	USART4 半双工同步数据端
25	I/O	P7.2	带弱上拉功能的双向输入输出端口
		T5CKI	T5 外部时钟输入
		T2CKI	T2 外部时钟输入

		SCK1(3)	MSSP1 的 SPI 时钟端
		SCL1(3)	MSSP1 的 I2C 时钟端
		CCP9(1)	CCP9 的比较捕捉功能
		P4A(1)	PWM4 输出通道
26	I/O	P7.1	带上拉和电平变化中断功能的双向输入输出端口
		T6CKI	T6 外部时钟输入
		T4CKI	T4 外部时钟输入
		T1CKI(1)	T1 外部时钟输入
		SDI1(3)	MSSP1 的 SPI 数据输入端
		SDA1(3)	MSSP1 的 I2C 数据端
27	I/O	P1.7	带上拉的双向输入输出端口
		SS1(2)	MSSP1 的 SPI 从动模式选择引脚
28	I/O	P1.6	带上拉的双向输入输出端口
29	I/O	P1.5	带上拉和电平变化中断功能的双向输入输出端口
		TX4(2)	USART4 全双工异步数据发送端
		CK4(2)	USART4 半双工同步时钟端
30	I/O	P1.4	带上拉的双向输入输出端口
		RX4(2)	USART4 全双工异步接收数据输入端
		DT4(2)	USART4 半双工同步数据端
31	I/O	P1.3	带弱上拉功能的双向输入输出端口
		AN20	ADC 通道 20
		C21-	比较器 2 的负端输入
		TX3(1)	USART3 全双工异步数据发送端
		CK3(1)	USART3 半双工同步时钟端
		IOC16	电平变化模块引脚输入
		CCP3(3)	CCP3 的比较捕捉功能
P7CH3H(2)	PWM7 通道 3 输出		
32	I/O	P1.2	带弱上拉功能的双向输入输出端口
		ADVRIN(2)	ADC 参考电压模拟输入
		AN21	ADC 通道 21
		C11-	比较器 1 的负端输入
		RX3(1)	USART3 全双工异步接收数据输入端
		DT3(1)	USART3 半双工同步数据端
		IOC17	电平变化模块引脚输入
		CCP7(3)	CCP7 的捕捉比较功能
		P6CH3H(2)	PWM6 通道 3 输出
33	I/O	P1.1	带弱上拉功能的双向输入输出端口
		C1OUT(2)	比较器 1 输出
		T0CKI(2)	T0 外部时钟输入引脚
		IOC18	电平变化模块引脚输入
		P3A(1)	PWM3 输出通道
		CCP8(1)	CCP8 的比较捕捉功能
		P7CH0H(2)	PWM7 通道 0 输出

		CCP2(2)	CCP2 的比较捕捉功能
34	I/O	P1.0	带弱上拉功能的双向输入输出端口
		AN22	ADC 通道 22
		C2OUT(3)	比较器 2 的输出
		SCL1(1)	MSSP1 的 I2C 时钟端
		SCK1(1)	MSSP1 的 SPI 时钟端
		SS2(3)	MSSP2 的 SPI 从动模式选择引脚
		IOC19	电平变化模块引脚输入
		P6CH0H(2)	PWM6 通道 0 输出
		CCP5(2)	CCP5 的比较捕捉功能
35	I/O	P2.3	带弱上拉功能的双向输入输出端口
		AN23	ADC 通道 23
		SDA1(1)	MSSP1 的 I2C 数据端
		SDI1(1)	MSSP1 的 SPI 数据输入端
		IOC20	电平变化模块引脚输入
		P5A(2)	PWM5 输出通道
		CCP10(2)	CCP10 的比较捕捉功能
36	I/O	P2.4	带弱上拉功能的双向输入输出端口
		AN24	ADC 通道 24
		SDO1(1)	MSSP1 的 SPI 数据输出端
		IOC21	电平变化模块引脚输入
		P5B(3)	PWM5 输出通道
		CCP9(5)	CCP9 的比较捕捉功能
		P6CH1H(2)	PWM6 通道 1 输出
37	I/O	P2.5	带弱上拉功能的双向输入输出端口
		AN25	ADC 通道 25
		IOC10(2)	电平变化模块引脚输入
		PWM2OUT(2)	PMW2 输出
		CCP7(2)	CCP7 的捕捉比较功能
38	P	VSS	
39	P	VDD	
40	I/O	P2.0	带弱上拉功能的双向输入输出端口
		IOC22	电平变化模块引脚输入
		CCP3(4)	CCP3 的比较捕捉功能
		P7CH2H(2)	PWM7 的通道 2 输出
41	I/O	P2.1	带弱上拉功能的双向输入输出端口
		IOC23	电平变化模块引脚输入
		CCP8(3)	CCP8 的比较捕捉功能
		P6CH2H(2)	PWM6 的通道 2 输出
42	I/O	P2.2	带弱上拉功能的双向输入输出端口
43	I/O	P5.0	带弱上拉功能的双向输入输出端口
		AN26	ADC 通道 26

		IOC11(2)	电平变化模块引脚输入
44	I/O	P5.1	带弱上拉功能的双向输入输出端口
		AN27	ADC 通道 27
		T1G(1)	T1 门控信号
		IOC8	电平变化模块引脚输入
45	I/O	P5.2	带弱上拉功能的双向输入输出端口
		AN28	ADC 通道 28
		IOC12(2)	电平变化模块引脚输入
46	I/O	P5.3	带弱上拉功能的双向输入输出端口
		CLKOUT(2)	系统时钟输出
		AN29	ADC 通道 29
		IOC0(1)	电平变化模块引脚输入
47	I/O	P5.4	带弱上拉功能的双向输入输出端口
		AN30	ADC 通道 30
		IOC13(2)	电平变化模块引脚输入
48	I/O	P5.5	带弱上拉功能的双向输入输出端口
		AN31	ADC 通道 31
		IOC9	电平变化模块引脚输入
49	I/O	P6.1	带弱上拉功能的双向输入输出端口
		AN32	ADC 通道 32
		C4+	比较器 4 的正端输入
		IOC1(1)	电平变化模块引脚输入
		P4A(4)	PWM4 输出通道
		CCP9(4)	CCP9 的比较捕捉功能
50	I/O	P6.2	带弱上拉功能的双向输入输出端口
		AN33	ADC 通道 33
		SS2(1)	MSSP2 模块 SPI 从动模式选择引脚
		IOC10(1)	电平变化模块引脚输入
		P3A(2)	PWM3 输出通道
		CCP8(2)	CCP8 的比较捕捉功能
51	I/O	P6.3	带弱上拉功能的双向输入输出端口
		AN34	ADC 通道 34
		C4-	比较器 4 的负端输入
		SCL2(1)	MSSP2 的 I2C 时钟端
		SCK2(1)	MSSP2 的 SPI 时钟端
		IOC2(1)	电平变化模块引脚输入
		P3B(2)	PWM3 输出通道
P6CH2H(1)	PWM6 通道 2 输出		
52	I/O	P6.4	带弱上拉功能的双向输入输出端口
		AN35	ADC 通道 35
		SDA2(1)	MSSP2 的 I2C 数据端
		SDI2(1)	MSSP2 的 SPI 数据输入端

		P3C(2)	PWM3 输出通道
53	I/O	P6.5	带弱上拉功能的双向输入输出端口
		AN36	ADC 通道 36
		SDO2(1)	MSSP2 的 SPI 数据输出端
		IOC11(1)	电平变化模块引脚输入
		P5B(2)	PWM5 输出通道
		P3D(2)	PWM3 输出通道
		P6CH1H(1)	PWM6 通道 1 输出
		54	I/O
AN37	ADC 通道 37		
C4OUT	比较器 4 的输出		
IOC3(1)	电平变化模块引脚输入		
P5C(2)	PWM5 输出通道		
55	P	VSS	
56	I/O	PB.1	带弱上拉功能的双向输入输出端口
57	I/O	PB.0	带弱上拉功能的双向输入输出端口
		ADVRIN(3)	ADC 参考电压模拟输入
58	P	VDD	
59	I/O	P9.5	带弱上拉功能的双向输入输出端口
		AN38	ADC 通道 38
		IOC4(1)	电平变化模块引脚输入
		PWM1OUT(2)	PWM1 输出
		P4B(1)	PWM4 输出通道
		CCP3(2)	CCP3 的比较捕捉功能
60	I/O	P9.4	带弱上拉功能的双向输入输出端口
		AN39	ADC 通道 39
		IOC12(1)	电平变化模块引脚输入
		P4B(3)	PWM4 输出通道
		P7CH0H(1)	PWM7 通道 0 输出
61	I/O	CCP5(1)	CCP5 的比较捕捉功能
		P9.3	带弱上拉功能的双向输入输出端口
		AN40	ADC 通道 40
62	I/O	P4C(3)	PWM4 输出通道
		P9.2	带弱上拉功能的双向输入输出端口
		AN41	ADC 通道 41
		T1G(3)	T1 门控信号
		SDA2(2)	MSSP2 的 I2C 数据端
		SDI2(2)	MSSP2 的 SPI 数据输入端
63	I/O	IOC5(1)	电平变化模块引脚输入
		P9.1	带弱上拉功能的双向输入输出端口
		AN42	ADC 通道 42
		SDO2(2)	MSSP2 的 SPI 数据输出端
		IOC13(1)	电平变化模块引脚输入



64	I/O	P9.0	带弱上拉功能的双向输入输出端口
		AN43	ADC 通道 43
		IOC6(1)	电平变化模块引脚输入

## 目 录

产品订购信息.....	2
文档说明.....	3
简介.....	3
KF8A100 芯片使用注意事项.....	3
寄存器操作缩写说明.....	4
寄存器内容及操作说明.....	4
引脚示意图.....	6
引脚功能说明.....	8
目 录.....	18
<b>1 系统概述.....</b>	<b>34</b>
1.1 芯片特征.....	35
1.2 系统框图.....	37
1.3 存储器.....	38
1.4 配置位.....	38
1.5 在线串行编程.....	40
<b>2 I/O 端口介绍.....</b>	<b>41</b>
2.1 I/O 端口的读写.....	42
2.2 P0 口.....	43
2.2.1 P0 口相关的寄存器.....	43
2.2.1.1 P0 口状态寄存器 (P0).....	43
2.2.1.2 P0 口输出锁存寄存器 (POLR).....	43
2.2.1.3 P0 口方向控制寄存器(TR0).....	44
2.2.1.4 P0 口模拟/数字口设置寄存器 (ANS0).....	44
2.2.1.5 P0 上拉功能控制寄存器(PUR0).....	45
2.2.1.6 P0 口电平变化中断控制寄存器(IOCL0).....	45
2.2.2 P0 口各引脚内部原理功能框图.....	46
2.3 P1 口.....	48
2.3.1 P1 口相关的寄存器.....	48
2.3.1.1 P1 口状态寄存器(P1).....	48
2.3.1.2 P1 口输出锁存寄存器 (P1LR).....	48
2.3.1.3 P1 口方向控制寄存器(TR1).....	49
2.3.1.4 P1 上拉功能控制寄存器(PUR1).....	49
2.3.1.5 P1 口模拟/数字口设置寄存器 (ANS1).....	49
2.3.2 P1 口原理功能框图.....	50
2.4 P2 口.....	51
2.4.1 P2 口相关的寄存器.....	51
2.4.1.1 P2 口状态寄存器(P2).....	51
2.4.1.2 P2 口输出锁存寄存器 (P2LR).....	51

2.4.1.3 P2 口方向控制寄存器(TR2).....	52
2.4.1.4 P2 口模拟/数字口设置寄存器 (ANS2) .....	52
2.4.1.5 P2 口上拉功能控制寄存器 (PUR2) .....	52
2.4.2 P2 口原理功能框图.....	53
2.5 P3 口 .....	54
2.5.1 P3 口相关的寄存器.....	54
2.5.1.1 P3 口状态寄存器(P3).....	54
2.5.1.2 P3 口输出锁存寄存器 (P3LR) .....	54
2.5.1.3 P3 口方向控制寄存器(TR3).....	55
2.5.1.4 P3 口模拟/数字口设置寄存器 (ANS3) .....	55
2.5.1.5 P3 上拉功能控制寄存器(PUR3).....	55
2.5.2 P3 口原理功能框图.....	56
2.6 P4 口 .....	57
2.6.1 P4 口相关的寄存器.....	57
2.6.1.1 P4 口状态寄存器(P4).....	57
2.6.1.2 P4 口输出锁存寄存器 (P4LR) .....	57
2.6.1.3 P4 口方向控制寄存器(TR4).....	58
2.6.1.4 P4 口模拟/数字口设置寄存器 (ANS4) .....	58
2.6.1.5 P4 上拉功能控制寄存器(PUR4).....	58
2.6.2 P4 口原理功能框图.....	59
2.7 P5 口 .....	60
2.7.1 P5 口相关的寄存器.....	60
2.7.1.1 P5 口状态寄存器(P5).....	60
2.7.1.2 P5 口输出锁存寄存器 (P5LR) .....	60
2.7.1.3 P5 口方向控制寄存器(TR5).....	61
2.7.1.4 P5 口模拟/数字口设置寄存器 (ANS5) .....	61
2.7.1.5 P5 上拉功能控制寄存器(PUR5).....	61
2.7.2 P5 口原理功能框图.....	62
2.8 P6 口 .....	63
2.8.1 P6 口相关的寄存器.....	63
2.8.1.1 P6 口状态寄存器(P6).....	63
2.8.1.2 P6 口输出锁存寄存器 (P6LR) .....	63
2.8.1.3 P6 口方向控制寄存器(TR6).....	64
2.8.1.4 P6 口模拟/数字口设置寄存器 (ANS6) .....	64
2.8.1.5 P6 上拉功能控制寄存器(PUR6).....	64
2.8.2 P6 口原理功能框图.....	65
2.9 P7 口 .....	66
2.9.1 P7 口相关的寄存器.....	66
2.9.1.1 P7 口状态寄存器(P7) ; 通过 I/O 口到 P7 寄存器 .....	66
2.9.1.2 P7 口输出锁存寄存器 (P7LR) 通过 P7 到 I/O 口寄存器 .....	66
2.9.1.3 P7 口方向控制寄存器(TR7).....	67
2.9.1.4 P7 口模拟/数字口设置寄存器 (ANS7) .....	67
2.9.1.5 P7 上拉功能控制寄存器(PUR7).....	67
2.9.2 P7 口原理功能框图.....	68

2.10 P9 口 .....	69
2.10.1 P9 口相关的寄存器 .....	69
2.10.1.1 P9 口状态寄存器(P9).....	69
2.10.1.2 P9 口输出锁存寄存器 (P9LR) .....	69
2.10.1.3 P9 口方向控制寄存器(TR9).....	70
2.10.1.4 P9 口模拟/数字口设置寄存器 (ANS9) .....	70
2.10.1.5 P9 上拉功能控制寄存器(PUR9).....	70
2.10.2 P9 口原理功能框图 .....	71
2.11 PA 口 .....	72
2.11.1 PA 口相关的寄存器 .....	72
2.11.1.1 PA 口状态寄存器(PA).....	72
2.11.1.2 PA 口输出锁存寄存器 (PALR) .....	72
2.11.1.3 PA 口方向控制寄存器(TRA).....	73
2.11.1.4 PA 口模拟/数字口设置寄存器 (ANSA) .....	73
2.11.1.5 PA 上拉功能控制寄存器(PURA).....	73
2.11.2 PA 口原理功能框图 .....	74
2.12 PB 口 .....	75
2.12.1 PB 口相关的寄存器 .....	75
2.12.1.1 PB 口状态寄存器(PB).....	75
2.12.1.2 PB 口输出锁存寄存器 (PBLR) .....	75
2.12.1.3 B 口方向控制寄存器(TRB).....	76
2.12.1.4 PB 口模拟/数字口设置寄存器 (ANSB) .....	76
2.12.1.5 PB 上拉功能控制寄存器(PURB).....	76
2.12.2 PB 口原理功能框图 .....	77
2.13 复用引脚功能 .....	78
2.13.1 复用引脚功能相关寄存器 .....	78
2.13.1.1 复用引脚功能控制寄存器 APFCTL2 .....	78
2.13.1.2 复用引脚功能控制寄存器 APFCTL3 .....	79
2.13.1.3 复用引脚功能控制寄存器 APFCTL4 .....	80
2.13.1.4 复用引脚功能控制寄存器 APFCTL5 .....	80
2.13.1.5 复用引脚功能控制寄存器 APFCTL6 .....	81
2.13.1.6 复用引脚功能控制寄存器 APFCTL7 .....	81
2.13.1.7 复用引脚功能控制寄存器 APFCTL8 .....	82
2.13.1.8 复用引脚功能控制寄存器 APFCTL9 .....	83
2.13.1.9 复用引脚功能控制寄存器 APFCTL10 .....	84
2.13.1.10 复用引脚功能控制寄存器 APFCTL11 .....	84
2.13.1.11 复用引脚功能控制寄存器 APFCTL12 .....	85
<b>3 振荡器 .....</b>	<b>86</b>
3.1 概述 .....	86
3.2 振荡器相关寄存器 .....	87
3.2.1 系统时钟控制寄存器 (SCLKCTL) .....	87
3.2.2 主时钟状态寄存器 (SCLKSTU) .....	88
3.2.3 高频外设时钟选择控制寄存器 (HFCKCTL) .....	89

3.2.4	低频外设时钟选择控制寄存器 (LFCKCTL)	89
3.2.5	校准寄存器	90
3.3	振荡器工作模式	90
3.3.1	振荡器详细描述	91
3.3.1.1	内部高频振荡器	91
3.3.1.2	内部低频振荡器	91
3.3.1.3	外部高频振荡器	92
3.3.1.4	外部低频振荡器	92
3.3.2	系统主时钟 SCLK	92
3.3.2.1	系统时钟的切换与同步	93
3.3.2.2	双速时钟启动模式	93
3.3.3	高频外设时钟 HFCLK	94
3.3.4	低频外设时钟 LFCLK	94
3.4	时钟信号同步	95
3.5	休眠模式	95
3.6	外部时钟故障检测	95
3.6.1	时钟故障保护检测	96
3.6.2	时钟故障保护处理	96
<b>4</b>	<b>存储器</b>	<b>97</b>
4.1	程序存储器(ROM)区	98
4.1.1	程序计数器(PC)	98
4.1.1.1	PCL/PCH 寄存器的使用	98
4.1.1.2	执行 JMP、CALL 指令时的情况	99
4.1.2	堆栈	99
4.2	数据存储器(RAM)区	100
4.2.1	通用寄存器区	100
4.2.2	特殊功能寄存器(SFR)区	101
4.3	FLASH 自写	103
4.3.1	寄存器 NVMDATAH/L	104
4.3.2	寄存器 NVMADDRH/L	104
4.3.3	写 Flash	104
4.3.4	读 Flash	105
4.4	DATA EEPROM	107
4.4.1	寄存器 NVMDATAL	107
4.4.2	寄存器 NVMADDRL	107
4.4.1	寄存器 NVMCTL0/NVMCTL1	107
4.4.2	写 DATA EEPROM	107
4.4.3	读 DATA EEPROM	108
4.5	寄存器组 RN	109
4.6	ID 地址单元	109
<b>5</b>	<b>汇编指令及寻址方式</b>	<b>110</b>
5.1	寻址方式	110
5.1.1	寄存器寻址	110

5.1.2 直接寻址.....	110
5.1.3 立即数寻址.....	110
5.1.4 寄存器间接寻址.....	111
5.1.5 位寻址.....	111
5.2 汇编指令.....	111
<b>6 中断.....</b>	<b>112</b>
6.1 中断相关的寄存器.....	114
6.1.1 中断控制寄存器 INTCTL.....	115
6.1.2 中断使能寄存器 EIE1.....	116
6.1.3 中断使能寄存器 EIE2.....	116
6.1.4 中断使能寄存器 EIE3.....	117
6.1.5 中断使能寄存器 EIE4.....	118
6.1.6 中断使能寄存器 EIE5.....	118
6.1.7 中断使能寄存器 EIE6.....	119
6.1.8 中断使能寄存器 EIE7.....	120
6.1.9 中断使能寄存器 EIE8.....	120
6.1.10 中断标志寄存器 EIF1.....	121
6.1.11 中断标志寄存器 EIF2.....	121
6.1.12 中断标志寄存器 EIF3.....	122
6.1.13 中断标志寄存器 EIF4.....	123
6.1.14 中断标志寄存器 EIF5.....	123
6.1.15 中断标志寄存器 EIF6.....	124
6.1.16 中断标志寄存器 EIF7.....	124
6.1.17 中断使能寄存器 EIF8.....	125
6.1.18 中断优先级控制寄存器 IP0.....	126
6.1.19 中断优先级控制寄存器 IP1.....	126
6.1.20 中断优先级控制寄存器 IP2.....	127
6.1.21 中断优先级控制寄存器 IP3.....	127
6.1.22 中断优先级控制寄存器 IP4.....	128
6.1.23 中断优先级控制寄存器 IP5.....	129
6.1.24 中断优先级控制寄存器 IP6.....	129
6.1.25 中断优先级控制寄存器 IP7.....	130
6.1.26 中断优先级控制寄存器 IP8.....	130
6.1.27 电源控制寄存器 PCTL.....	131
6.1.28 INT 边沿触发选择.....	132
6.1.29 中断响应.....	132
6.2 INT 中断.....	133
6.2.1 INT0 中断.....	133
6.2.2 INT1 中断.....	133
6.3 定时器中断.....	134
6.4 P0 口中断.....	134
6.5 ADC 中断.....	134
6.6 PWM1/2 中断.....	134

6.7 模拟比较器中断 .....	134
6.8 CCP 2/5 中断 .....	134
6.9 CCP 8/9/10 中断 .....	135
6.10 USART 中断 .....	135
6.11 RTC 中断 .....	135
6.12 SSCI 中断 .....	135
6.13 中断现场保护 .....	135
<b>7 电平变化中断 .....</b>	<b>136</b>
7.1 电平中断原理框图 .....	136
7.2 使能模块 .....	137
7.3 电平变化相关的寄存器 .....	137
7.3.1 IOCAPOS 控制寄存器 .....	137
7.3.2 IOCANEG 控制寄存器 .....	139
7.3.3 IOCMOD 控制寄存器 .....	141
7.3.4 IOCAF 状态寄存器 .....	141
7.4 中断标志 .....	142
7.5 清零中断标志 .....	142
<b>8 定时器/计数器 .....</b>	<b>143</b>
8.1 定时器/计数器 T0 .....	143
8.1.1 T0 原理框图 .....	143
8.1.2 T0 相关的寄存器 .....	143
8.1.2.1 OPTR 选择寄存器 .....	144
8.1.2.2 T0CTL 控制寄存器 .....	145
8.1.3 定时模式 .....	145
8.1.4 计数模式 .....	145
8.1.5 T0 的使用 .....	145
8.2 定时器/计数器 T1 .....	146
8.2.1 T1 原理框图 .....	146
8.2.2 T1 相关的寄存器 .....	146
8.2.2.1 T1 控制寄存器 (T1CTL) .....	147
8.2.2.2 T1 控制寄存器 2 (T1CTL2) .....	148
8.2.3 定时模式 .....	148
8.2.4 门控模式 .....	148
8.2.4.1 T1 门控使能 .....	148
8.2.4.1 T1 门控信号源选择 .....	149
8.2.4.2 T1 门控交替计数模式 .....	149
8.2.4.3 T1 门控单脉冲模式 .....	149
8.2.4.4 T1 门控值状态 .....	149
8.2.4.5 T1 门控中断 .....	150
8.2.5 计数模式 .....	150
8.2.6 重载模式 .....	150
8.2.7 T1 在休眠模式下的运行 .....	150
8.3 定时器 T2/T4 .....	151

8.3.1 T2/T4 相关的寄存器.....	152
8.3.1.1 T2 控制寄存器 0(T2CTL0) .....	152
8.3.1.2 T4 控制寄存器 0(T4CTL0) .....	153
8.3.1.3 T2 控制寄存器 1(T2CTL1) .....	154
8.3.1.4 T4 控制寄存器 1(T4CTL1) .....	154
8.3.2 T2/T4 的工作原理 .....	155
8.3.3 T2 计数模式.....	155
8.3.4 T2/T4 分频器 .....	156
8.3.5 T2/T4 中断 .....	156
8.3.6 T2/T4 在休眠模式 .....	156
8.3.7 T2/T4 分配给 PWM6 .....	157
8.3.8 T2/4 触发 AD .....	157
8.3.9 T2/4 读写操作 .....	157
8.4 定时器 T5/T6.....	158
8.4.1 T5/T6 相关的寄存器.....	159
8.4.1.1 T5 控制寄存器 0(T5CTL0) .....	159
8.4.1.2 T6 控制寄存器 0(T6CTL0) .....	160
8.4.1.3 T5 控制寄存器 1(T5CTL1) .....	161
8.4.1.4 T6 控制寄存器 1(T6CTL1) .....	161
8.4.2 T5/T6 的工作原理 .....	162
8.4.3 T5 计数模式.....	162
8.4.4 T5/T6 分频器 .....	163
8.4.5 T5/T6 中断 .....	163
8.4.6 T5/T6 在休眠模式 .....	163
8.4.7 T5/T6 分配给 PWM7 .....	164
8.4.8 T5/T6 触发 AD.....	164
8.4.9 T5/T6 读写操作.....	164
8.5 定时器/计数器 T3/T7.....	165
8.5.1 T3/T7 原理框图.....	165
8.5.2 T3/T7 相关的寄存器.....	165
8.5.2.1 T3 控制寄存器 .....	166
8.5.2.2 T7 控制寄存器 .....	167
8.5.2.3 CCP3 控制寄存器.....	167
8.5.2.4 CCP7 控制寄存器.....	168
8.5.3 T3/T7 预分频器.....	168
8.5.4 T3/T7 计数时钟选择.....	169
8.5.5 T3/T7 重载功能.....	169
8.5.6 T3/T7 中断 .....	169
8.5.7 T3/T7 工作在休眠模式.....	169
8.5.8 CCP 功能 .....	169
8.5.8.1 捕捉模式.....	169
8.5.8.2 比较模式.....	170
8.6 定时器 T8/9.....	171
8.6.1 T8/T9 的工作原理 .....	171



8.6.2 T8/9 相关的寄存器.....	173
8.6.2.1 T8 控制寄存器 T8CTL .....	173
8.6.2.2 T8 控制寄存器 T8CTL1.....	174
8.6.2.3 T9 控制寄存器 T9CTL .....	174
8.6.2.4 T9 控制寄存器 T9CTL1.....	175
8.6.2.5 T8CCRH/L 和 T9CCRH/L 寄存器.....	175
8.6.3 T8/T9 中断.....	175
8.6.4 T8/T9 在休眠模式.....	176
8.6.5 T8/T9 分配给 PWM3/4 .....	176
8.7 定时器 T10.....	176
8.7.1 T10 的工作原理 .....	176
8.7.2 T10 相关的寄存器.....	177
8.7.2.1 T10 控制寄存器 T10CTL0 .....	178
8.7.2.2 T10 控制寄存器 T10CTL1 .....	178
8.7.2.3 T10CCR1/0H 和 T10CCR1/0L 寄存器.....	179
8.7.3 T10 中断 .....	179
8.7.4 T10 在休眠模式.....	179
8.7.5 T10 分配给 PWM5 .....	179
<b>9 模数(A/D)转换模块.....</b>	<b>180</b>
9.1 与 AD 相关的寄存器 .....	180
9.1.1 AD 控制寄存器 0(ADCCTL0).....	181
9.1.2 AD 控制寄存器 1(ADCCTL1).....	181
9.1.3 AD 控制寄存器 2(ADCCTL2).....	182
9.1.4 AD 中断控制寄存器 (ADCINTCTL) .....	183
9.1.5 AD 控制寄存器 3 (ADCCTL3) .....	184
9.1.6 温度传感器配置寄存器(TEMPSNR).....	185
9.2 通道的选择.....	185
9.3 模拟输入口的配置.....	185
9.4 多通道扫描模式 .....	185
9.4.1 AD 多通道扫描控制寄存器.....	186
9.5 A/D 转换参考电压的选择.....	186
9.6 转换时钟的选择 .....	186
9.7 输出格式.....	186
9.8 A/D 转换的启动和完成.....	187
9.9 A/D 工作在休眠模式.....	187
9.10 复位的影响.....	187
9.11 使用 A/D 转换器的设置.....	188
9.12 TxCCR <sub>x</sub> 触发 AD.....	188
9.13 CCP 触发 AD.....	188
<b>10 PWM1/2 模块 .....</b>	<b>189</b>
10.1 PWM1/2 原理框图.....	189
10.2 PWM 相关的寄存器 .....	192
10.3 PWM 控制寄存器.....	192

10.4 PWM 周期.....	192
10.5 PWM 占空比.....	193
10.6 PWM 分辨率.....	193
10.7 PWM 中断.....	193
10.8 休眠模式下的操作.....	194
10.9 复位的影响.....	194
10.10 PWM 使用方法.....	194
<b>11 CCP2/5(捕捉/比较/PWM6、7)模块.....</b>	<b>195</b>
11.1 捕捉/比较相关寄存器.....	195
11.1.1 PWM6CTL0 寄存器.....	196
11.1.2 PWM7CTL0 寄存器.....	197
11.1.3 PWM6H0/PWM6L0、 PWM7H0/PWM7L0 寄存器.....	197
11.2 捕捉模式.....	198
11.3 比较模式.....	199
11.4 PWM6/7 模式.....	200
11.4.1 PWM6/7 相关寄存器.....	201
11.4.1.1 PWM6CTL2 更新控制寄存器.....	202
11.4.1.2 PWM7CTL2 更新控制寄存器.....	203
11.4.1.3 PWM6PC 极性控制寄存器.....	204
11.4.1.4 PWM7PC 极性控制寄存器.....	204
11.4.1.5 P6ATRCTL 寄存器.....	205
11.4.1.6 P7ATRCTL 寄存器.....	205
11.4.2 PWM6/7 的周期、占空比及分辨率.....	205
11.4.2.1 PWM6/7 周期.....	206
11.4.2.2 PWM6/7 占空比.....	206
11.4.2.3 PWM6/7 分辨率(最小占空比).....	207
11.4.2.4 PWM6/7 中断.....	208
11.4.3 边沿对齐 PWM 信号.....	208
11.4.4 PWM 信号产生和中断.....	208
11.4.5 PWM 输出设置.....	209
11.4.6 PWM 更新锁定.....	209
11.4.7 PWM 复位控制.....	209
<b>12 CCP8/9(捕捉/比较/PWM3、4)模块.....</b>	<b>210</b>
12.1 捕捉/比较相关寄存器.....	210
12.1.1 PWM3CTL0 寄存器.....	210
12.1.2 PWM4CTL0 寄存器.....	211
12.2 捕捉模式.....	212
12.3 比较模式.....	213
12.4 PWM3/4 模块.....	213
12.5 PWM3/4 相关寄存器.....	215
12.5.1 PWM4CTL0 寄存器.....	215
12.5.2 PWM4CTL1 寄存器.....	216
12.5.3 P4ASCTL 寄存器.....	216

12.5.4 P4ATRCTL 寄存器 .....	217
12.6 PWM3/4 的周期、占空比及分辨率 .....	218
12.6.1 PWM3/4 周期 .....	218
12.6.2 PWM3/4 占空比 .....	218
12.6.3 PWM3/4 分辨率 .....	218
12.6.4 PWM3/4 中断 .....	218
12.7 单输出模式 .....	219
12.8 半桥输出模式 .....	220
12.8.1 死区延时 .....	221
12.9 全桥输出模式 .....	222
12.10 自动关断和自动重启模式 .....	225
12.10.1 自动关断模式 .....	225
12.10.2 自动重启模式 .....	226
<b>13 CCP10(捕捉/比较/PWM5)模块 .....</b>	<b>228</b>
13.1 CCP 相关寄存器 .....	228
13.2 捕捉模式 .....	229
13.3 比较模式 .....	230
13.4 PWM5 模式 .....	231
13.4.1 PWM5 相关控制寄存器 .....	233
13.4.1.1 PWM5CTL0 寄存器 .....	233
13.4.1.2 PWM5CTL1 寄存器 .....	234
13.4.1.3 PWM5CTL2 寄存器 .....	235
13.4.1.4 PWM5 极性控制寄存器 (PWM5PC) .....	235
13.4.1.5 PWM5 强制控制寄存器 (PWM5FC) .....	236
13.4.1.6 PWM5 输出控制寄存器 (PWM5OC) .....	236
13.4.1.7 P5ASCTL0 寄存器 .....	237
13.4.1.8 P5ASCTL1 寄存器 .....	238
13.4.1.9 P5STRCTL0 寄存器 .....	238
13.4.1.10 P5STRCTL1 寄存器 .....	239
13.4.2 PWM5 的周期、占空比及分辨率 .....	239
13.4.2.1 PWM5 周期 .....	239
13.4.2.2 PWM5 占空比 .....	240
13.4.2.3 PWM5 分辨率 .....	240
13.4.2.4 PWM5 中断 .....	240
13.4.3 边沿对齐 PWM 信号 .....	241
13.4.4 PWM 信号产生和中断 .....	241
13.4.5 单输出模式 .....	241
13.4.6 半桥输出模式 .....	243
13.4.6.1 死区延时 .....	244
13.4.7 全桥输出模式 .....	245
13.4.8 脉冲转向控制 .....	248
13.4.9 输出控制 .....	249
13.4.10 强制输出模式 .....	249

13.4.11 PWM 输出极性模式 .....	249
13.4.12 PWM 更新锁定.....	249
13.4.13 PWM 复位控制.....	250
13.4.14 自动关断和自动重启模式.....	250
13.4.14.1 自动关断模式.....	250
13.4.14.2 自动重启模式.....	251
<b>14 模拟比较器模块.....</b>	<b>253</b>
14.1 模拟比较器原理.....	254
14.2 与模拟比较器相关的寄存器 .....	255
14.2.1 比较器中断触发控制寄存器 CMCTL0.....	255
14.2.2 比较器 1 控制寄存器 C1CTL.....	256
14.2.3 比较器结果寄存器 COUT.....	257
14.2.4 比较器 2 控制寄存器 C2CTL.....	257
14.2.5 比较器 3 控制寄存器 C3CTL.....	258
14.2.6 比较器 4 控制寄存器 C4CTL.....	259
14.3 零点检测电压 .....	259
14.4 电阻分压 .....	259
14.4.1 电阻分压配置寄存器 0.....	260
14.4.2 电阻分压配置寄存器 1.....	261
14.5 滤波器相关寄存器.....	262
14.5.1 滤波器控制寄存器 FILTxCTL .....	262
14.5.2 滤波器 x 采样时钟分频寄存器 .....	263
14.6 极性选择 .....	263
14.7 范围控制功能 .....	263
14.8 滤波功能 .....	264
14.9 比较器中断.....	264
14.10 比较器使用.....	264
14.11 比较器清零定时器.....	265
14.11.1 比较器清零定时器控制寄存器 CCTCTL.....	265
14.11.2 使用方法 .....	266
<b>15 通用全双工/ 半双工收发器.....</b>	<b>267</b>
15.1 系统概述 .....	267
15.1.1 相关寄存器 .....	267
15.1.2 原理框图.....	269
15.2 波特率发生器 .....	270
15.2.1 相关寄存器 .....	270
15.2.1.1 USARTx 波特率控制寄存器 BRCTLx.....	270
15.2.2 波特率的选择.....	272
15.2.3 自动波特率检测.....	273
15.2.4 接收间隔字符时自动唤醒.....	274
15.2.5 间隔符时序 .....	275
15.2.5.1 间隔和同步发送序列.....	276
15.2.5.2 接收间隔字符.....	276

15.3 USART 全双工模式.....	277
15.3.1.1 USARTx 全双工发送操作.....	277
15.3.1.2 发送状态和控制寄存器 TSCTLx .....	278
15.3.1.3 发送数据.....	279
15.3.1.4 全双工发送的设置.....	279
15.3.2 USART 全双工接收操作 .....	281
15.3.2.1 接收状态和控制寄存器 RSCTLx .....	282
15.3.2.2 接收数据.....	283
15.3.2.3 接收错误.....	283
15.3.2.4 地址检测.....	284
15.3.2.5 全双工接收的设置.....	284
15.3.3 RS-485 发送/接收.....	285
15.3.3.1 RS-485 9 位地址检测模式设置 .....	285
15.3.4 全双工操作时钟的精确性.....	286
15.4 USART 半双工模式.....	286
15.4.1 USARTx 半双工主控模式.....	286
15.4.1.1 半双工主控发送.....	287
15.4.1.2 半双工主控接收.....	288
15.4.2 USART 半双工从动模式 .....	290
15.4.2.1 USART 半双工从动发送 .....	290
15.4.2.2 USART 半双工从动接收.....	291
15.4.3 USART 半双工 RS-485 模式.....	291
15.5 USART 单线通信模式.....	292
15.5.1 USART 引脚配置寄存器 UPINSETx.....	292
15.6 7816 模式.....	293
15.6.1 7816 模式发送.....	293
15.6.1.1 发送设置.....	293
15.6.2 7816 模式接收.....	294
15.6.3 7816 模式寄存器.....	295
15.6.3.1 7816 发送控制寄存器 U7816TXCTL2.....	295
15.6.3.2 7816 接收控制寄存器 U7816RXCTL2 .....	296
15.6.3.3 7816 控制寄存器 U7816CTL2.....	297
15.6.3.4 预分频控制寄存器 CLKDIV2 .....	297
15.6.3.5 EGT 控制寄存器 EGTCTL2.....	297
<b>16 SSCI 模块 .....</b>	<b>299</b>
16.1 概述.....	299
16.2 SSCI 相关寄存器 .....	299
16.2.1 SSCIx 控制寄存器 0 (SSCIxCTL0) .....	300
16.2.2 SSCIx 控制寄存器 1 (SSCIxCTL1) .....	301
16.2.3 SSCIx 控制寄存器 2 (SSCIxCTL2) .....	301
16.2.4 SSCIx 状态寄存器 (SSCIxSTA) .....	302
16.2.5 SSCIx 屏蔽寄存器 (SSCIxMSK) .....	303
16.2.6 SSCIx I2C 地址寄存器 (SSCIxADD) .....	304

16.3 I2C 模式.....	304
16.3.1 工作原理.....	304
16.3.2 I2C 从动模式.....	306
16.3.2.1 寻址.....	306
16.3.2.2 接收.....	307
16.3.2.3 发送.....	308
16.3.2.4 广播呼叫地址支持.....	310
16.3.3 I2C 主控模式.....	310
16.3.3.1 主控模式支持.....	311
16.3.3.2 I2C 主模式操作.....	312
16.3.3.3 波特率发生器.....	312
16.3.3.4 I2C 主控模式启动条件时序.....	313
16.3.3.5 I2C 主控模式重复启动条件时序.....	314
16.3.3.6 I2C 主控模式发送.....	315
16.3.3.7 I2C 主控模式接收.....	316
16.3.3.8 应答序列时序.....	318
16.3.3.9 停止条件序列.....	318
16.3.3.10 时钟仲裁.....	319
16.3.4 多主控器模式.....	320
16.3.4.1 多主机通信，总线冲突与总线仲裁.....	320
16.3.4.2 启动条件期间的总线冲突.....	321
16.3.4.3 重复启动条件期间的总线冲突.....	323
16.3.4.4 停止条件期间的总线冲突.....	324
16.3.4.5 SSCI 屏蔽寄存器.....	325
16.4 SPI 模式.....	326
16.4.1 工作原理.....	327
16.4.2 使能 SPI/IO 与外部链接.....	327
16.4.3 典型连接.....	328
16.4.4 主模式.....	328
16.4.5 从模式.....	329
16.4.5.1 从动模式.....	329
16.4.5.2 从动选择同步.....	330
16.4.6 休眠模式和复位.....	331
16.4.7 SPI 四种工作模式设置.....	332
16.4.7.1 主控发送工作流程.....	332
16.4.7.2 从动接收工作流程.....	332
16.4.7.3 主控接收工作流程.....	333
16.4.7.4 从动发送工作流程.....	333
<b>17 实时时钟 (RTC) 模块.....</b>	<b>334</b>
17.1 概述.....	334
17.1.1 原理框图.....	334
17.2 寄存器描述.....	335
17.2.1 相关寄存器.....	335

17.2.2	实时时钟启动寄存器 RTCSRT.....	335
17.2.3	状态显示寄存器 RTCSTU.....	336
17.2.4	闹钟中断时间寄存器.....	337
17.2.4.1	闹钟中断秒寄存器 RTCALRS.....	337
17.2.4.2	闹钟中断分寄存器 RTCALRM.....	337
17.2.4.3	闹钟中断时寄存器 RTCALRH.....	337
17.2.4.4	闹钟中断星期寄存器 RTCALRW.....	338
17.2.5	时间节拍中断寄存器 RTCTTR.....	338
17.2.6	实时时钟校正寄存器 RTCFCR.....	339
17.2.7	实时时钟寄存器.....	339
17.2.7.1	实时时钟秒寄存器 RTCSEC.....	339
17.2.7.2	实时时钟分寄存器 RTCMIN.....	339
17.2.7.3	实时时钟时寄存器 RTCHOUR.....	340
17.2.7.4	实时时钟星期寄存器 RTCWEK.....	340
17.2.7.5	实时时钟日寄存器 RTCDAY.....	340
17.2.7.6	实时时钟月寄存器 RTCMTH.....	340
17.2.7.7	实时时钟年寄存器 RTCYEAR.....	341
17.2.7.8	实时时钟定时器使能寄存器 RTCTMREN.....	341
17.2.7.9	实时时钟定时器控制寄存器 RTCTMRCTL.....	341
17.2.7.10	实时时钟定时器 0 计数寄存器 RTCTMR0.....	342
17.2.7.11	实时时钟定时器 1 计数寄存器 RTCTMR1.....	342
17.3	功能描述.....	343
17.3.1	RTC 初始化.....	343
17.3.2	RTC 启动.....	343
17.3.3	RTC 时间设置.....	343
17.3.4	RTC 闹钟功能.....	344
17.3.5	时钟校正功能.....	345
17.3.5.1	计算方法.....	345
17.3.5.2	校准原理.....	345
17.3.5.3	校准寄存器设定值表.....	346
17.3.6	12/24 小时模式选择.....	347
17.3.7	闰年显示功能.....	347
17.3.8	RTC 定时器.....	347
17.3.9	RTC 工作在休眠模式.....	348
17.3.10	RTC 中断.....	348
17.3.10.1	闹钟中断时间设置.....	348
17.3.10.2	周期时间节拍中断.....	349
17.3.10.3	进程中断.....	349
17.3.10.4	内置定时器中断.....	349
18	看门狗定时器.....	350
18.1	独立看门狗 IWDT.....	350
18.1.1	WDT 预分频选择寄存器 WDTPS.....	350
18.2	窗口看门狗模块 WWDT.....	351

18.2.1 窗口看门狗相关寄存器 .....	351
18.2.1.1 窗口看门狗控制寄存器 WCTL .....	351
18.2.1.2 窗口看门狗计数寄存器 WCNT .....	352
18.2.1.3 窗口看门狗配置寄存器 WCFR .....	352
18.2.2 窗口看门狗原理 .....	353
18.2.3 窗口看门狗中断 .....	354
<b>19 复位 .....</b>	<b>355</b>
19.1 电源控制状态寄存器(PCTL) .....	356
19.2 上电复位(POR) .....	357
19.3 WDT 复位 .....	357
19.4 RST 复位 .....	357
19.5 欠压检测复位(LVR) .....	358
19.6 上电延时定时器 .....	358
19.7 不同复位条件下对寄存器的影响 .....	359
<b>20 电源管理和功耗模式 .....</b>	<b>360</b>
20.1 电源 .....	360
20.1.1 内置电压调节器 .....	360
20.2 电源管理 .....	360
20.2.1 被封区读写配置寄存器 .....	360
20.2.2 备份区寄存器的读写 .....	361
20.3 功耗模式 .....	362
20.3.1 功耗模式相关寄存器 .....	362
20.3.1.1 功耗模式控制寄存器 .....	362
20.3.1.2 备份区外设复位寄存器 .....	362
20.3.2 正常运行模式 .....	363
20.3.3 普通休眠模式 .....	363
20.3.4 深度休眠模式 .....	363
20.3.5 普通/深度休眠模式下 I/O 口注意事项 .....	363
20.3.6 普通/深度休眠模式唤醒方式 .....	364
<b>21 电气规范 .....</b>	<b>365</b>
21.1 极限参数值 .....	365
21.2 HFINTOSC 的频率精度与 VDD 和温度之间的关系 .....	366
21.3 静态电流特性 .....	367
21.4 休眠电流特性 .....	367
21.5 外设电流特性 .....	368
21.6 I/O 端口电平和芯片供电电压特性 .....	369
21.7 POR .....	370
21.8 LVR .....	370
21.9 INTHF .....	371
21.10 INTLF .....	371
21.11 EXTLF .....	371
21.12 EXTHF .....	371



21.13 ADC12.....	372
21.14 CMP.....	372
21.15 ESD 和 LATCH UP.....	372
21.16 直流特性图表.....	374
<b>22 封装信息.....</b>	<b>382</b>
<b>附录 1 特殊功能寄存器（SFR）地址映射及功能汇总.....</b>	<b>384</b>
BANK0.....	384
BANK1.....	387
BANK2.....	390
BANK3.....	392
BANK4.....	393
BANK5.....	396
BANK6.....	398
<b>附录 2 汇编指令集.....</b>	<b>400</b>
<b>附录 3 全双工异步模式的典型波特率和误差值.....</b>	<b>403</b>
<b>附录 4 版本信息.....</b>	<b>407</b>
<b>产品标识体系.....</b>	<b>409</b>
<b>ROHS 认证.....</b>	<b>410</b>
<b>声明及销售网络.....</b>	<b>411</b>

## 1 系统概述

KF8A100 为哈佛结构的精简指令 CPU。在这种结构中，程序和数据总线是相互独立的。指令字节长度为 16 位，大多数指令能在一个机器周期内执行完成。一共有 73 条指令，效率高，容易进行指令扩展。

芯片内集成了多种外设，包括：

1 个 8 位定时器/计数器 T0；

10 个 16 位定时器/计数器 T1/T2/T3/T4/T5/T6/T7/T8/T9/T10；

2 个看门狗定时器：IWDG/WWDT；

1 个 12 位 43 通道 A/D 模块；

5 路 CCP 模块；

2 个独立的 USART 模块；

2 个带 7816 的串口模块；

4 个模拟比较器；

2 个 SSCI 总线模块；

1 个 RTC 模块；

硬件看门狗及低电压复位模块等。

芯片内集成了 32K×16 位的程序存储器、4K 字节的用户数据存储器 RAM 和 256×8 字节的 Data EEPROM。

## 1.1 芯片特征

### ● CPU

高性能哈佛结构的 RISC CPU  
73 条精简指令  
支持中断优先级处理  
复位向量位于 0000H  
两级中断可选，用不同的入口地址（高 0004H，低 0014H）  
工作时钟软件可选  
系统时钟最高可达 16MHz

### ● 存储器

32K×16 位 FLASH 程序存储器  
4K×8 位的数据存储器  
可选的 FLASH 自写范围  
工作寄存器组 R0~R7  
FLASH 可经受 100 000 次写操作  
DATA EEPROM 可经受 1 000 000 次写操作

### ● 特殊功能

内嵌上电复位电路  
低电压复位  
硬件双看门狗  
内部高频时钟精度 16MHz±1%  
可外接低频振荡器和外接高频振荡器  
4 种时钟源  
3 种时钟信号选择  
支持在线串行编程，低功耗休眠模式

### ● I/O 口配置

输入输出：P0.0 只能作为输入口外其它口均为双向输入输出  
内置上拉功能：P0~PB 口均带有弱上拉功能(P0.0 除外)  
电平变化中断：P0 口均有电平变化中断功能

### ● 定时器/计数器

定时器 0：带有 8 位预分频器的 8 位定时器/计数器  
定时器 1：带重载功能、门控和预分频器的 16 位定时器/计数器  
定时器 2/4/5/6/8/9/10：带 16 位周期寄存器、预分频器和后分频器 16 位定时器  
定时器 3/7：带捕捉比较、重载功能、预分频器的 16 位定时器/计数器

### ● 其它外设

1 个 12 位 ADC 模块  
5 路 CCP 模块  
4 个模拟比较器模块(边沿触发中断)  
2 个 SSCI 总线模块  
2 个独立的 USART 模块

1 个带 7816 的串口模块（带 LIN、IRDA）

1 个独立的 RTC（实时计数）

1 个高低压检测模块；

2 个看门狗定时器：IWDG/WWDT；

● **AEC-Q100**

已通过 AEC-Q100 测试

● **工作条件**

工作电压： 2.7V~5.5V

工作温度范围： -40~125°C

1.2 系统框图

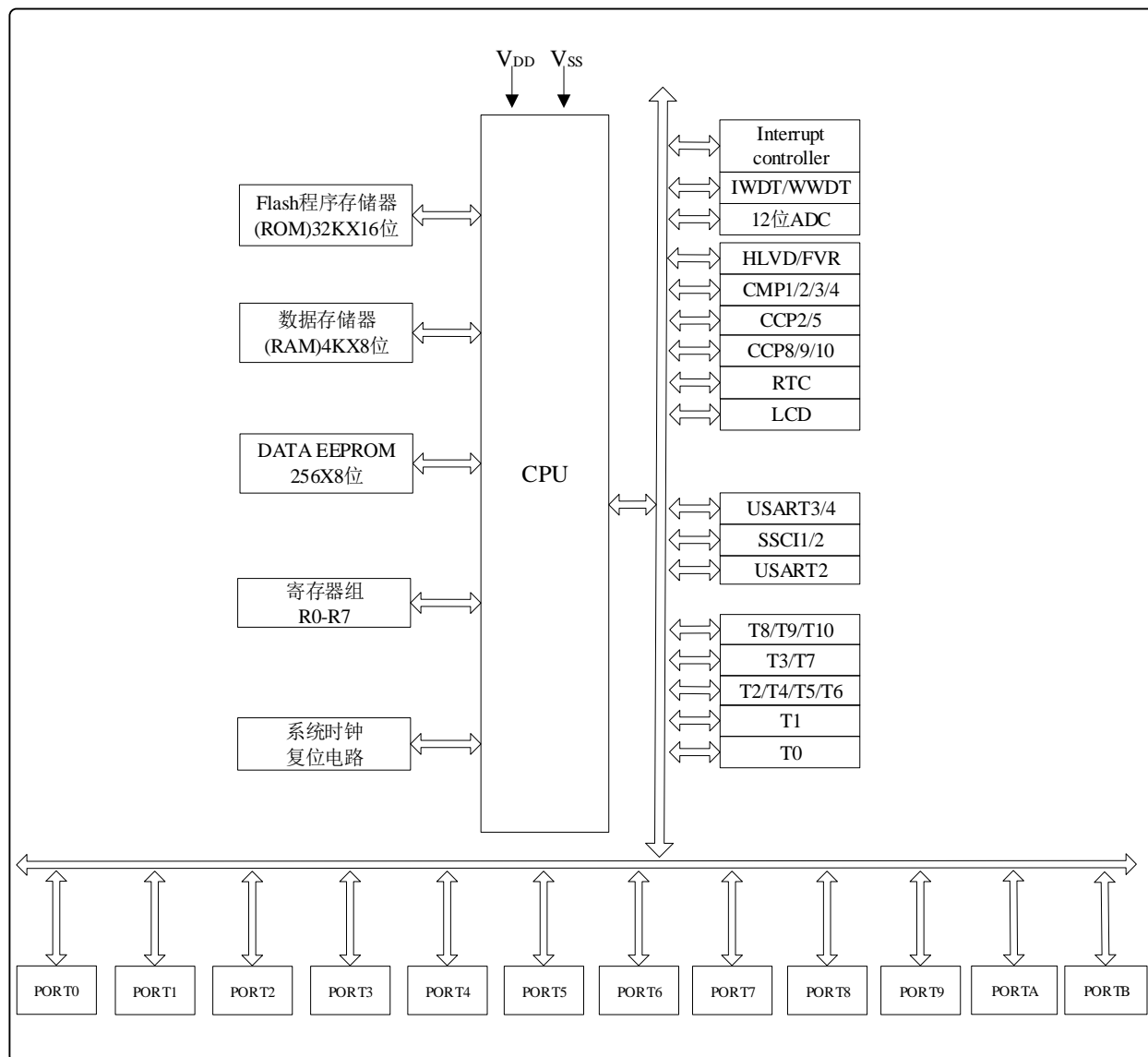


图 1.1KF8A100 单片机系统框图

## 1.3 存储器

KF8A100 单片机的存储器包含:程序存储器(ROM)、数据存储器(RAM)和 DATA EEPROM。

KF8A100 的程序存储器空间为 32K×16 位，寻址范围为 0000H~7FFFH，可擦写次数为 10 万次。数据存储器有 37 个存储区，数据存储器分为特殊寄存器区(SFR)和通用存储器区。特殊功能寄存器区包括 BANK0、BANK1、BANK2、BANK3、BANK4 和 BANK5，地址为 00H~6FH、100H~16FH、200H~26FH、300H~36FH、400H~46FH 和 500H~56FH。而通用寄存器区一共包括 32 个区，每个区有 128×8 位的存储单元，各区的地址请查阅第 4 章。

KF8A100 带有的 256×8 位的 DATA EEPROM 是独立寻址的，其地址为 00H-FFH。有关以上各种存储器的具体介绍请参考第 4 章。

## 1.4 配置位

如下面寄存器所示，用户在烧写程序时，在编程器中通过对配置位进行设置，使单片机启用诸如看门狗、程序代码保护、欠压检测等功能。

**CONFIG: 配置字(地址:8007H)**

R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P
WRT1	WRT0	PWMPIN	HPOL	LPOL	DEBUG	-----	IESO	CODEP	LVREN	RSTEN	PWRT	IWDTE N	DATAP	FSCM	F0SC0		
bit15							bit8							bit0			

注: R=可读 P=编程时可写

**WRT<1:0>: Flash 自写保护使能位**

- 00 = 写保护关闭，全区域可写
- 01 = 0000H 至 07FFH 受写保护
- 10 = 0000H 至 3FFFH 受写保护
- 11 = 0000H 至 7FFFH 受写保护，全区域不可写

**PWMPIN: PWM3/4/5/6/7 复位控制**

- 1 = HPOL/LPOL 控制复位时的引脚输出状态
- 0 = I/O 端口控制复位时的引脚状态

**HPOL: PWM3/4/5/6/7 高边驱动器极性位 (high side polarity)**

- 0 = 引脚P6/7CHxH (x=0,1,2,3) 引脚和PxA/C (x=3,4,5) 上的输出信号极性为低电平有效
- 1 = 引脚P6/7CHxH (x=0,1,2,3) 引脚和PxB/D (x=3,4,5) 上的输出信号极性为高电平有效

**LPOL: PWM3/4/5/6/7 低边驱动器极性位 (low side polarity)**

- 0 = P6/7CHxL (x=0,1,2,3) 引脚和PxB/D (x=3,4,5) 上的输出信号极性为低电平有效
- 1 = P6/7CHxL (x=0,1,2,3) 引脚和PxB/D (x=3,4,5) 上的输出信号极性为高电平有效

注: 复位默认高阻态; 上电复位时因为引脚复用寄存器 APFCTLx 为复位值, 所以 HPOL 和 LPOL 仅控制 PWM3/4/5/6/7 默认输出引脚, 比如只控制 P3A(1), 而 P3A(2) 不受影响

- $\overline{\text{DEBUG}}$ : 在线调试使能位  
 1 = 禁止在线调试  
 0 = 使能在线调试
- IESO: 内部外部时钟切换使能位  
 1 = 内外时钟切换使能  
 0 = 内外时钟切换禁止
- $\overline{\text{CODEP}}$ : 代码保护使能位  
 1 = 禁止程序存储器代码保护  
 0 = 使能程序存储器代码保护
- LVREN: 欠压检测功能使能位  
 1 = 使能欠压检查功能  
 0 = 禁止欠压检查功能
- RSTEN:  $\overline{\text{P0.0/RST}}$  引脚功能选择  
 1 =  $\overline{\text{P0.0/RST}}$  引脚配置为外部复位输入  
 0 =  $\overline{\text{P0.0/RST}}$  引脚功能为数字输入口
- $\overline{\text{PWRT}}$ : 上电延时使能位  
 0 = 使能上电延时  
 1 = 禁止上电延时
- IWDTEN: 独立看门狗定时器(IWDT)使能位  
 1 = 使能 IWDT  
 0 = 禁止 IWDT
- $\overline{\text{DATAP}}$ : DATA EEPROM 加密使能位  
 1 = 禁止数据存储区加密  
 0 = 使能数据存储区加密
- FSCM: 时钟故障检测使能位  
 1 = 时钟故障检测使能  
 0 = 时钟故障检测禁止
- FOSC0: 高频时钟源选择位  
 1 = 外部高频时钟(EXTHF)作为系统时钟  
 0 = 内部高频时钟(INTHF)作为系统时钟

CONFIG: 校验字(地址:8008H)

R	R	R	R	R	R	R	R	R	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P
—	—	—	—	—	—	—	—	LPLVREN	POR3	BOR3	BOR2	BOR1	BOR0	POR2	POR1	POR0	
bit15								bit8								bit0	

注: R=编程器可读 P=编程时可写

- LPLVREN: 低功耗欠压检测功能使能位  
 LPLVREN=1 使能低功耗欠压检查功能  
 LPLVREN=0 禁止低功耗欠压检查功能
- BOR<3:0>: BOD 电压校验位
- POR<3:0>: POR 电压校验位

## 1.5 在线串行编程

如图 1.2、图 1.3 所示，在最终应用电路中可对 KF8A100 单片机进行在线串行编程、调试。实现串行编程仅需要四根线包括：时钟线(SPCLK)、数据线(SPDAT)、电源线(VDD)、地线(VSS)。

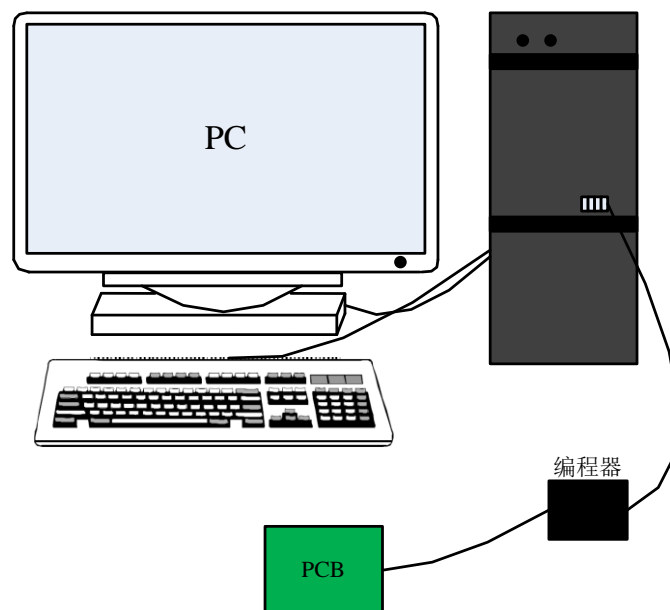


图 1.2 在线调试系统示意图

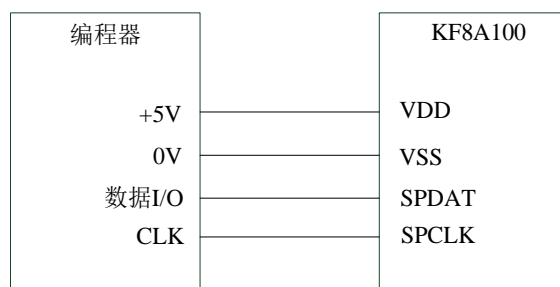


图 1.3 在线串行编程连接图



## 2 I/O 端口介绍

如引脚示意图所示，KF8A100 单片机的封装形式为 LQFP64。

- KF8A100-LQFP64 共有 64 个引脚，包括三组电源地和 58 个功能引脚

注: 用户在正常使用时，通常会有一些用不到的引脚，如果直接把这些管脚悬空，而不做其他处理可能使单片机功耗增大，因此建议将那些不用的引脚设置为模拟输入模式或者数字输出模式；

P0.0 无输出模式，不使用时建议设置为模拟输入模式，当配置为数字输入模式且不用时，建议外接上拉电阻。

## 2.1 I/O 端口的读写

当 IO 端口配置为模拟口时，任何对 IO 口电平的读操作都为低电平。

当 IO 端口配置为数字口时，用户可通过读 Px 寄存器获得端口电平信息。

当 IO 端口配置为数字输出口时，用户可通过写 PxLR 寄存器配置 IO 端口的输出电平。

注：x=0~11。

其原理框图如图 2.1 所示：

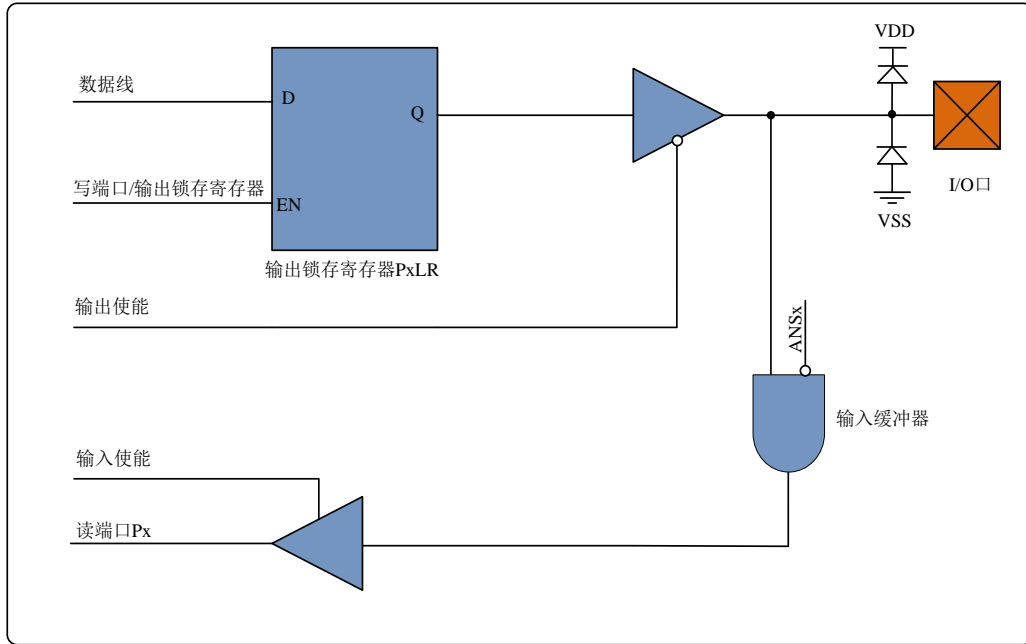


图 2.1 I/O 口读写原理图

## 2.2 P0 口

P0 口共有 8 个引脚。在线编程时 P0 口的 V<sub>PP</sub>、SPCLK、SPDAT 作为编程脚使用。P0.0 只能作为输入口且没有上拉功能，其它端口均可作为普通 I/O 口且带有上拉功能，P0 口所有引脚都有电平变化中断功能。

### 2.2.1 P0 口相关的寄存器

表 2.1 与 P0 端口相关的寄存器

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
05H	P0	P07	P06	P05	P04	P03	P02	P01	P00
45H	P0LR	P0LR7	P0LR6	P0LR5	P0LR4	P0LR3	P0LR2	P0LR1	P0LR0
25H	TR0	TR07	TR06	TR05	TR04	TR03	TR02	TR01	TR00
36H	IOCL0	IOCL07	IOCL06	IOCL05	IOCL04	IOCL03	IOCL02	IOCL01	IOCL00
35H	PUR0	PUR07	PUR06	PUR05	PUR04	PUR03	PUR02	PUR01	-
4AH	ANS0	ANS07	ANS06	ANS05	ANS04	ANS03	ANS02	ANS01	ANS00

#### 2.2.1.1 P0 口状态寄存器 (P0)

P0 寄存器各位对应 P0 口相应引脚当前的状态，如下寄存器所示：

**寄存器：** P0: P0口状态寄存器(地址: 05H)

		bit7							bit0
复位值 xxxx xxxx		P07	P06	P05	P04	P03	P02	P01	P00
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P0<7:0>: P0 口状态位

1 = 对应引脚为逻辑高电平

0 = 对应引脚为逻辑低电平

#### 2.2.1.2 P0 口输出锁存寄存器 (P0LR)

寄存器 P0LR 是 P0 口输出锁存寄存器。在 P0 口作为数字输出时，通过写 P0LR 寄存器来设置 P0 口的输出状态。

**寄存器：** P0LR: P0口输出锁存寄存器(地址: 45H)

		bit7							bit0
复位值 xxxx xxxx		P0LR7	P0LR6	P0LR5	P0LR4	P0LR3	P0LR2	P0LR1	P0LR0
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P0LR<7:0>: 写 P0 口输出状态

1 = 对应引脚输出高电平

0 = 对应引脚输出低电平

### 2.2.1.3 P0 口方向控制寄存器(TR0)

如下寄存器所示，TR0 为 P0 口方向控制寄存器，当 TR0 某位置 1 时，将该引脚设置为输入，此时引脚为三态(悬空)，TR0 某位清 0，对应引脚设置为输出。

**寄存器:** TR0: P0口方向控制寄存器(地址: 25H)

		bit7						bit0	
复位值		TR07	TR06	TR05	TR04	TR03	TR02	TR01	TR00
1111 1111		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TR0<7:0>: P0 口各引脚方向控制位

1 = 对应的引脚设置为输入

0 = 对应的引脚设置为输出

### 2.2.1.4 P0 口模拟/数字口设置寄存器 (ANS0)

P0 口模拟/数字口设置寄存器 ANS0 用于将 P0 口设置为模拟口或者数字口，通过将 ANS0 某位置 1，将对应的引脚设置为模拟口，清 0 设置为数字 I/O 口。

**寄存器:** ANS0: P0口模拟/数字口设置寄存器(地址:4AH)

		bit7						bit0	
复位值		ANS07	ANS06	ANS05	ANS04	ANS03	ANS02	ANS01	ANS00
1111 1111		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ANS0<7:0>: P0 口各引脚模拟/数字口设置位

1 = 将对应引脚配置为模拟口

0 = 将对应引脚配置为数字口或者特殊功能引脚

注:

- 1.端口配置为模拟口时，端口方向必须配置为输入状态；
- 2.当P0口配置为模拟（输入）口时，P0寄存器和POLR寄存器将无效；即无法通过P0寄存器读端口状态，也无法通过POLR寄存器对端口进行电平控制。

### 2.2.1.5 P0 上拉功能控制寄存器(PUR0)

KF8A100 中 P0 引脚除了 P0.0 外均带有上拉功能, 可通过上拉功能控制寄存器和 OPTR 寄存器中的  $\overline{\text{PUPH}}$  来控制上拉功能是否打开。

如果要将某引脚的上拉功能打开, 需要先将  $\overline{\text{PUPH}}$  (上拉功能总使能位) 位清 0, 允许 P0 口上拉功能打开, 然后再将要打开上拉功能的引脚, 所对应的上拉功能控制位置 1 即可。寄存器 PUR0 为上拉功能控制寄存器。

注: 只有将引脚设置为数字输入时才开启上拉电阻功能, 如果将某引脚设置为输出或者设置为模拟输入时将会自动禁止该引脚的上拉电阻。

**寄存器:** PUR0: P0口弱上拉控制寄存器(地址: 35H)

		bit7							bit0
复位值	1111 1111	PUR07	PUR06	PUR05	PUR04	PUR03	PUR02	PUR01	-
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUR0<7:1>: 上拉功能使能位

1 = 使能对应的端口上拉功能

0 = 禁止对应的端口上拉功能

### 2.2.1.6 P0 口电平变化中断控制寄存器(IOCL0)

P0 口每个引脚都具有电平变化中断功能, 当引脚的当前电平与上次读 P0 寄存器时的电平不匹配时将产生电平变化中断。如下寄存器所示, IOCL0 为电平变化中断控制寄存器, 将 IOCL0 某位置 1 将开启对应引脚的电平变化中断功能, 如果该引脚电平发生变化, 不管电平变化中断是否使能, 电平变化中断标志位(POIF)都会置 1, 如果全局中断使能位(AIE)和电平变化中断使能位(POIE)都已置 1, 则会响应中断进入中断服务子程序。P0 口所有引脚的电平变化中断共用一个标志位 POIF。

注:

- 只有将引脚设置为数字输入时才开启电平变化中断功能, 如果将某引脚设置为输出或者设置为模拟输入时将会自动禁止该引脚的电平变化中断功能。
- P0口各引脚的电平变化中断共用一个中断使能位和中断响应标志位。
- 清零POIF前, 需要对P0口状态寄存器(P0)执行读操作, 否则可能无法清零POIF。

**寄存器:** IOCL0: P0口电平变化中断控制寄存器(地址:36H)

		bit7							bit0
复位值	0000 0000	IOCL07	IOCL06	IOCL05	IOCL04	IOCL03	IOCL02	IOCL01	IOCL00
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IOCL0<7:0>: P0 端口引脚电平变化中断使能控制位

1 = 使能对应引脚的电平变化中断

0 = 禁止对应引脚的电平变化中断

2.2.2 P0 口各引脚内部原理功能框图

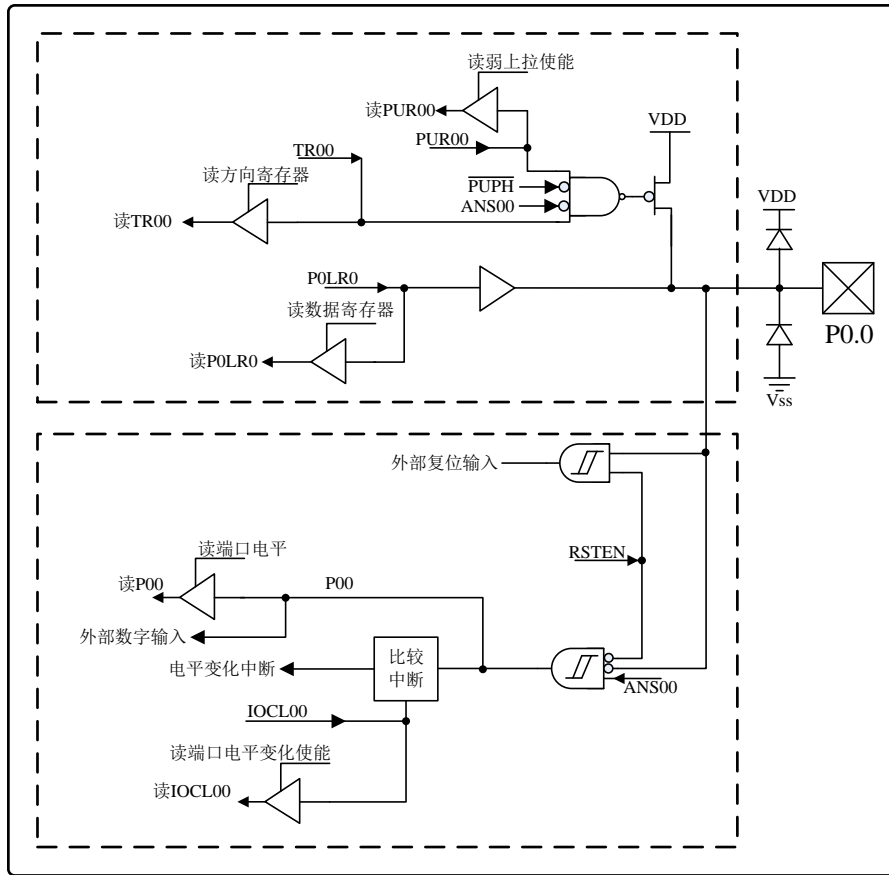


图 2.2 引脚 P0.0 原理功能框图

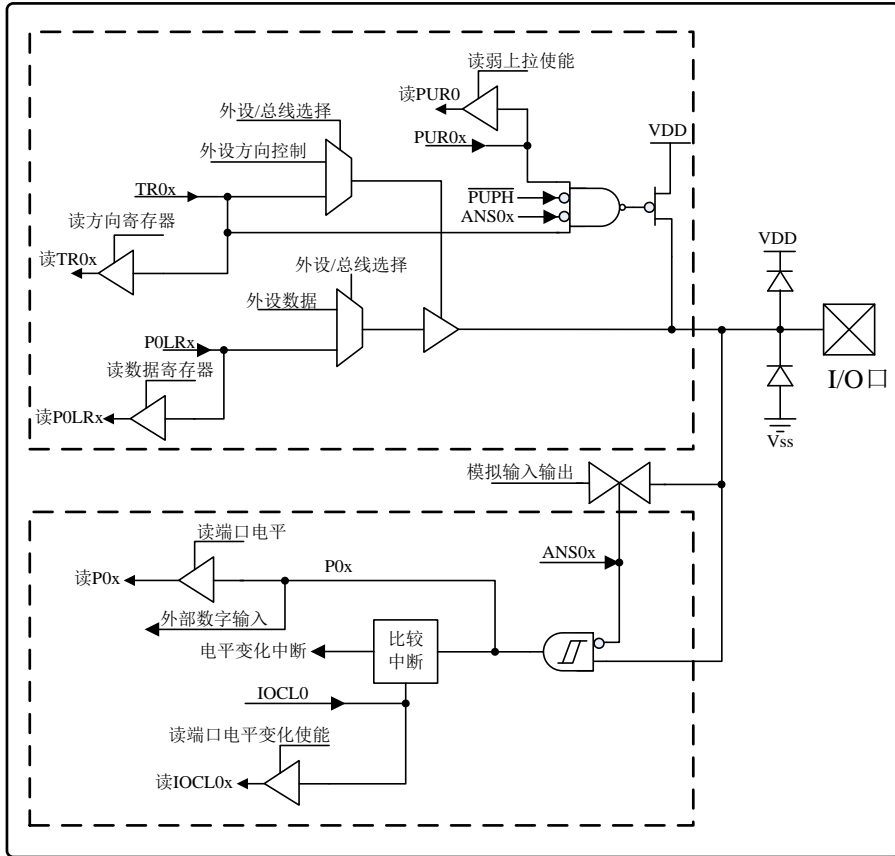


图 2.3 P0.1~P0.7 口引脚原理框图

## 2.3 P1 口

P1 口具有 8 个引脚。所有管脚均可作为普通 I/O 口。

### 2.3.1 P1 口相关的寄存器

表 2.2 与 P1 口相关的寄存器

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
07H	P1	P17	P16	P15	P14	P13	P12	P11	P10
47H	P1LR	P1LR7	P1LR6	P1LR5	P1LR4	P1LR3	P1LR2	P1LR1	P1LR0
27H	TR1	TR17	TR16	TR15	TR14	TR13	TR12	TR11	TR10
241H	PUR1	PUR17	PUR16	PUR15	PUR14	PUR13	PUR12	PUR11	PUR10
4BH	ANS1	ANS17	ANS16	ANS15	ANS14	ANS13	ANS12	ANS11	ANS10

#### 2.3.1.1 P1 口状态寄存器(P1)

寄存器 P1 对应端口 P1 引脚作为普通 I/O 口时的状态。如下寄存器所示

寄存器: P1: P1口状态寄存器(地址: 07H)

		bit7						bit0	
复位值		P17	P16	P15	P14	P13	P12	P11	P10
xxxx xxxx		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P1<7:0>: P1 口各引脚状态位

1 = 对应引脚为逻辑高电平

0 = 对应引脚为逻辑低电平

#### 2.3.1.2 P1 口输出锁存寄存器 (P1LR)

寄存器 P1LR 是 P1 口输出锁存寄存器。在 P1 口作为输出时，我们是通过写 P1LR 寄存器来设置输出 P1 口的状态。

寄存器: P1LR: P1口输出锁存寄存器(地址: 47H)

		bit7						bit0	
复位值		P1LR7	P1LR6	P1LR5	P1LR4	P1LR3	P1LR2	P1LR1	P1LR0
xxxx xxxx		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P1LR<7:0>: 写 P1 口输出状态

1 = 对应引脚输出高电平

0 = 对应引脚输出低电平



### 2.3.1.3 P1 口方向控制寄存器(TR1)

通过将寄存器 TR1 中的某位置 1，将对应管脚设置为输入口。清 0 设置为输出口，系统复位时，P1 口各引脚默认为输入口。

**寄存器:** TR1: P1口方向控制寄存器(地址: 27H)

		bit7						bit0	
复位值	1111 1111	TR17	TR16	TR15	TR14	TR13	TR12	TR11	TR10
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TR1<7:0>: P1 口引脚方向控制位  
 1 = P1 口对应引脚被配置为输入端口  
 0 = P1 口对应引脚被配置为输出端口

### 2.3.1.4 P1 上拉功能控制寄存器(PUR1)

如果要将某引脚的上拉功能打开，需要先将 $\overline{\text{PUPH}}$ (上拉功能总使能位)位清 0，允许 P1 口上拉功能打开，然后再将要打开上拉功能的引脚，所对应的上拉功能控制位置 1 即可。PUR1 为上拉功能控制寄存器。

**寄存器:** PUR1: P1口弱上拉控制寄存器(地址: 241H)

		bit7						bit0	
复位值	1111 1111	PUR17	PUR16	PUR15	PUR14	PUR13	PUR12	PUR11	PUR10
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUR1<7:0>: 上拉功能使能位  
 1 = 使能对应的端口上拉功能  
 0 = 禁止对应的端口上拉功能

### 2.3.1.5 P1 口模拟/数字口设置寄存器 (ANS1)

P1 口模拟/数字口设置寄存器 ANS1 用于将 P1 口设置为模拟口或者数字口，通过将 ANS1 某位置 1，将对应的引脚设置为模拟口，清 0 设置为数字 I/O 口。

**寄存器:** ANS1: P1口模拟/数字口设置寄存器(地址:4BH)

		bit7						bit0	
复位值	1111 1111	ANS17	ANS16	ANS15	ANS14	ANS13	ANS12	ANS11	ANS10
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ANS1<7:0>: P1 口各引脚模拟/数字口设置位  
 1 = 将对应引脚配置为模拟口  
 0 = 将对应引脚配置为数字口或者特殊功能引脚

注:

- 1.端口配置为模拟口时，端口方向必须配置为输入状态；
- 2.当P1口配置为模拟（输入）口时，P1寄存器和P1LR寄存器将无效；即无法通过P1寄存器读端口状态，也无法通过P1LR寄存器对端口进行电平控制。

2.3.2 P1 口原理功能框图

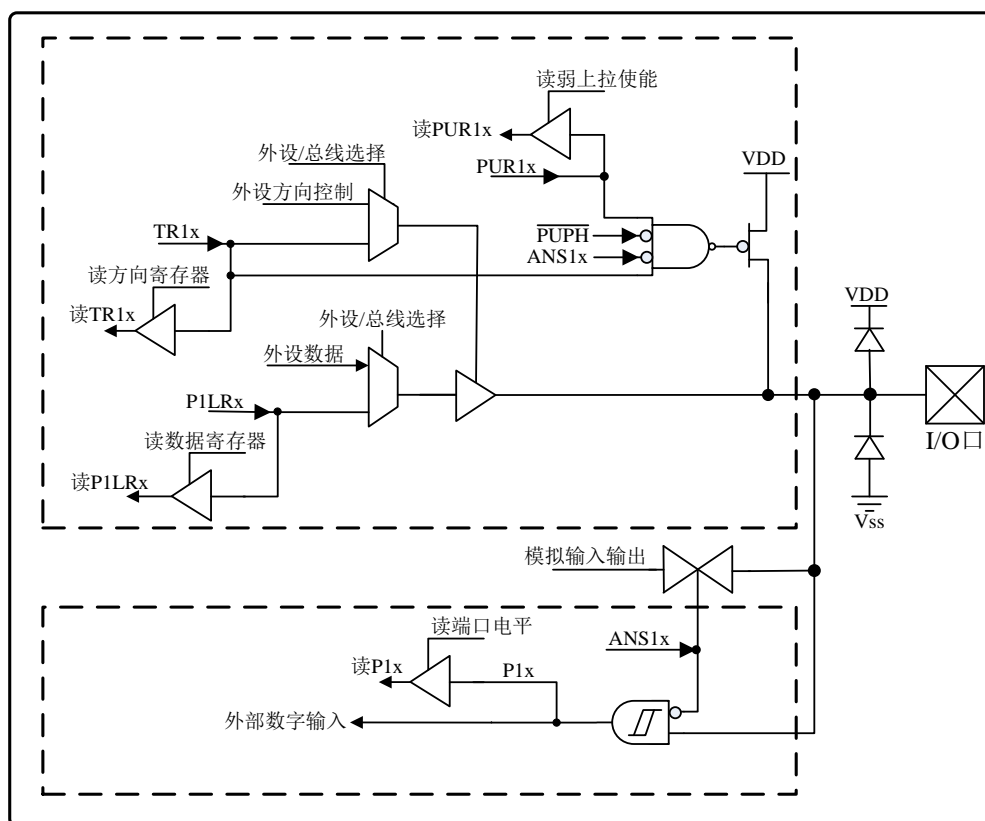


图 2.4 P1 口引脚原理功能框图

## 2.4 P2 口

P2 口共有 6 个引脚。所有管脚均可作为普通 I/O 口。

### 2.4.1 P2 口相关的寄存器

表 2.3 与 P2 口相关的寄存器

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
06H	P2	-	-	P25	P24	P23	P22	P21	P20
46H	P2LR	-	-	P2LR5	P2LR4	P2LR3	P2LR2	P2LR1	P2LR0
26H	TR2	-	-	TR25	TR24	TR23	TR22	TR21	TR20
4CH	ANS2	-	-	ANS25	ANS24	ANS23	ANS22	ANS21	ANS20
242H	PUR2	-	-	PUR25	PUR24	PUR23	PUR22	PUR21	PUR20

#### 2.4.1.1 P2 口状态寄存器(P2)

寄存器 P2 各位对应端口 P2 口各引脚作为普通 I/O 口时的状态。如下寄存器所示：

**寄存器：** P2: P2口状态寄存器(地址: 06H)

		bit7						bit0	
复位值 xxxx xxxx		-	-	P25	P24	P23	P22	P21	P20
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P2<5:0>: P2 口各引脚状态位  
 1 = 对应引脚为逻辑高电平  
 0 = 对应引脚为逻辑低电平

#### 2.4.1.2 P2 口输出锁存寄存器 (P2LR)

寄存器 P2LR 是 P2 口输出锁存寄存器。在 P2 口作为输出时，我们是通过写 P2LR 寄存器来设置输出 P2 口的状态。

**寄存器：** P2LR: P2口输出锁存寄存器(地址: 46H)

		bit7						bit0	
复位值 xxxx xxxx		-	-	P2LR5	P2LR4	P2LR3	P2LR2	P2LR1	P2LR0
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P2LR<5:0>: 写 P2 口输出状态  
 1 = 对应引脚输出高电平  
 0 = 对应引脚输出低电平

### 2.4.1.3 P2 口方向控制寄存器(TR2)

如下寄存器所示，通过将寄存器 TR2 中的某位置 1，将对应管脚设置为输入口。清 0 设置为输出口。

**寄存器:** TR2: P2口方向控制寄存器(地址: 26H)

		bit7						bit0	
复位值	--11 1111	-	-	TR25	TR24	TR23	TR22	TR21	TR20
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TR2<5:0>: P2 口各引脚方向控制位  
 1 = P2 口对应引脚被配置为输入端口  
 0 = P2 口对应引脚被配置为输出端口

### 2.4.1.4 P2 口模拟/数字口设置寄存器 (ANS2)

P2 口模拟/数字口设置寄存器 ANS2 用于将 P2 口设置为模拟口或者数字口，通过将 ANS2 某位置 1，将对应的引脚设置为模拟口，清 0 设置为数字 I/O 口。

**寄存器:** ANS2: P2口模拟/数字口设置寄存器(地址:4CH)

		bit7						bit0	
复位值	--11 1111	-	-	ANS25	ANS24	ANS23	ANS22	ANS21	ANS20
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ANS2<5:0>: P2 口各引脚模拟/数字口设置位  
 1 = 将对应引脚配置为模拟口  
 0 = 将对应引脚配置为数字口或者特殊功能引脚

注:

- 1.端口配置为模拟口时，端口方向必须配置为输入状态；
- 2.当P2口配置为模拟（输入）口时，P2寄存器和P2LR寄存器将无效；即无法通过P2寄存器读端口状态，也无法通过P2LR寄存器对端口进行电平控制。

### 2.4.1.5 P2 口上拉功能控制寄存器 (PUR2)

如果要将某引脚的上拉功能打开，需要先将 $\overline{\text{PUPH}}$ (上拉功能总使能位)位清 0，允许 P2 口上拉功能打开，然后再将要打开上拉功能的引脚，所对应的上拉功能控制位置 1 即可。

**寄存器:** PUR2: P2口弱上拉控制寄存器(地址: 242H)

		bit7						bit0	
复位值	--11 1111	-	-	PUR25	PUR24	PUR22	PUR23	PUR21	PUR20
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUR2<5:0>: 上拉功能使能位  
 1 = 使能对应的端口上拉功能  
 0 = 禁止对应的端口上拉功能

2.4.2 P2 口原理功能框图

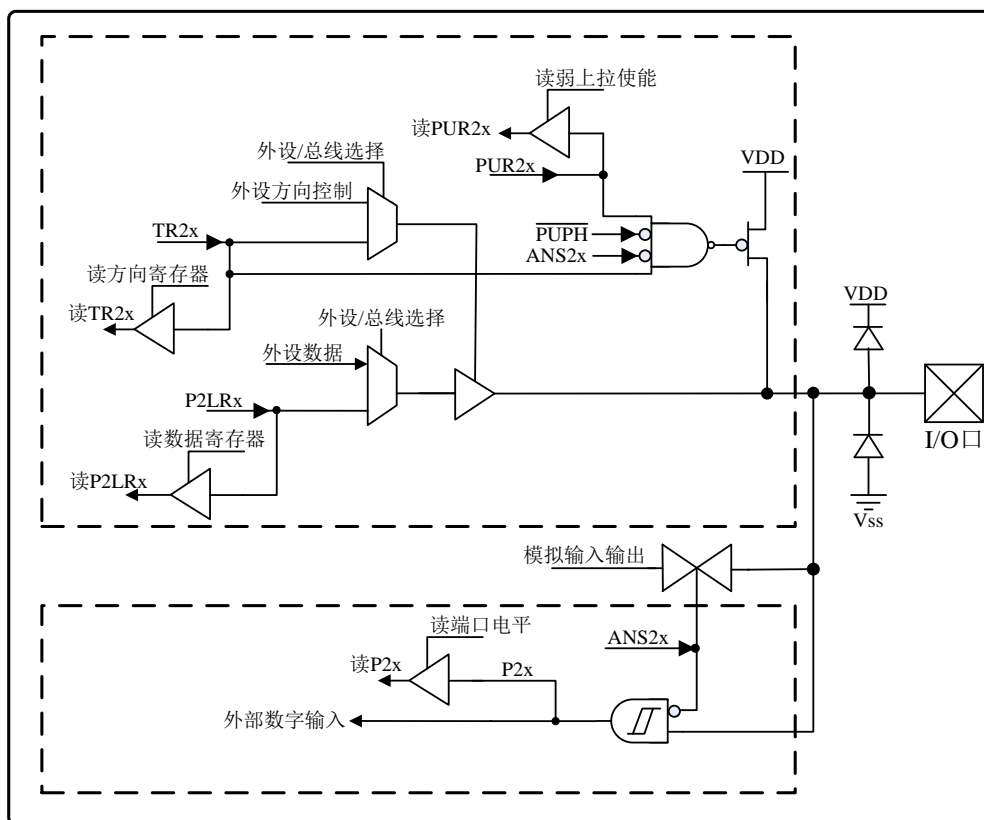


图 2.5 P2 口引脚原理功能框图

## 2.5 P3 口

P3 口共有 7 个引脚，所有管脚均可作为通用 I/O 口。

### 2.5.1 P3 口相关的寄存器

表 2.4 与 P3 口相关的寄存器

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
08H	P3	P37	P36	P35	P34	-	P32	P31	P30
48H	P3LR	P3LR7	P3LR6	P3LR5	P3LR4	-	P3LR2	P3LR1	P3LR0
28H	TR3	TR37	TR36	TR35	TR34	-	TR32	TR31	TR30
4DH	ANS3	ANS37	ANS36	ANS35	ANS34	-	ANS32	ANS31	ANS30
243H	PUR3	PUR37	PUR36	PUR35	PUR34	-	PUR32	PUR31	PUR30

#### 2.5.1.1 P3 口状态寄存器(P3)

寄存器 P3 各位对应端口 P3 口各引脚作为普通 I/O 口时的状态。如下寄存器所示。

**寄存器:** P3: P3口状态寄存器(地址:08H)

		bit7					bit0		
复位值		P37	P36	P35	P34	-	P32	P31	P30
XXXX XXXX		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P3x: P3 口各引脚状态位

1 = 对应引脚为逻辑高电平

0 = 对应引脚为逻辑低电平

#### 2.5.1.2 P3 口输出锁存寄存器 (P3LR)

寄存器 P3LR 是 P3 口输出锁存寄存器。在 P3 口作为输出时，我们是通过写 P3LR 寄存器来设置输出 P3 口的状态。

**寄存器:** P3LR: P3口输出锁存寄存器(地址:48H)

		bit7					bit0		
复位值		P3LR7	P3LR6	P3LR5	P3LR4	-	P3LR2	P3LR1	P3LR0
XXXX XXXX		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P3LRx: 写 P3 口输出状态

1 = 对应引脚输出高电平

0 = 对应引脚输出低电平

### 2.5.1.3 P3 口方向控制寄存器(TR3)

通过将寄存器 TR3 中的某位置 1，将对应管脚设置为输入口。清 0 设置为输出口。

**寄存器:** TR3: P3口方向控制寄存器(地址: 28H)

bit7				bit0				
复位值 1111-111	TR37	TR36	TR35	TR34	-	TR32	TR31	TR30
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TR3x: P3 口各引脚方向控制位

1 = P3 口对应引脚被配置为输入端口

0 = P3 口对应引脚被配置为输出端口

### 2.5.1.4 P3 口模拟/数字口设置寄存器 (ANS3)

P3 口模拟/数字口设置寄存器 ANS3 用于将 P3 口设置为模拟口或者数字口，通过将 ANS3 某位置 1，将对应的引脚设置为模拟口，清 0 设置为数字 I/O 口。

**寄存器:** ANS3: P3口模拟/数字口设置寄存器(地址:4DH)

bit7				bit0				
复位值 1111-111	ANS37	ANS36	ANS35	ANS34	-	ANS32	ANS31	ANS30
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ANS3x: P3 口各引脚模拟/数字口设置位

1 = 将对应引脚配置为模拟口

0 = 将对应引脚配置为数字口或者特殊功能引脚

注:

- 1.端口配置为模拟口时，端口方向必须配置为输入状态；
- 2.当P3口配置为模拟（输入）口时，P3寄存器和P3LR寄存器将无效；即无法通过P3寄存器读端口状态，也无法通过P3LR寄存器对端口进行电平控制。
- 3.清零P0IF前，需要对P0口状态寄存器(P0)执行读操作，否则可能无法清零P0IF。

### 2.5.1.5 P3 上拉功能控制寄存器(PUR3)

如果要将某引脚的上拉功能打开，需要先将 $\overline{\text{PUPH}}$ (上拉功能总使能位)位清 0，允许 P3 口上拉功能打开，然后再将要打开上拉功能的引脚，所对应的上拉功能控制位置 1 即可。

**寄存器:** PUR3: P3口弱上拉控制寄存器(地址: 243H)

bit7				bit0				
复位值 1111-111	PUR37	PUR36	PUR35	PUR34	-	PUR32	PUR31	PUR30
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUR3x: 上拉功能使能位

1 = 使能对应的端口上拉功能

0 = 禁止对应的端口上拉功能

2.5.2 P3 口原理功能框图

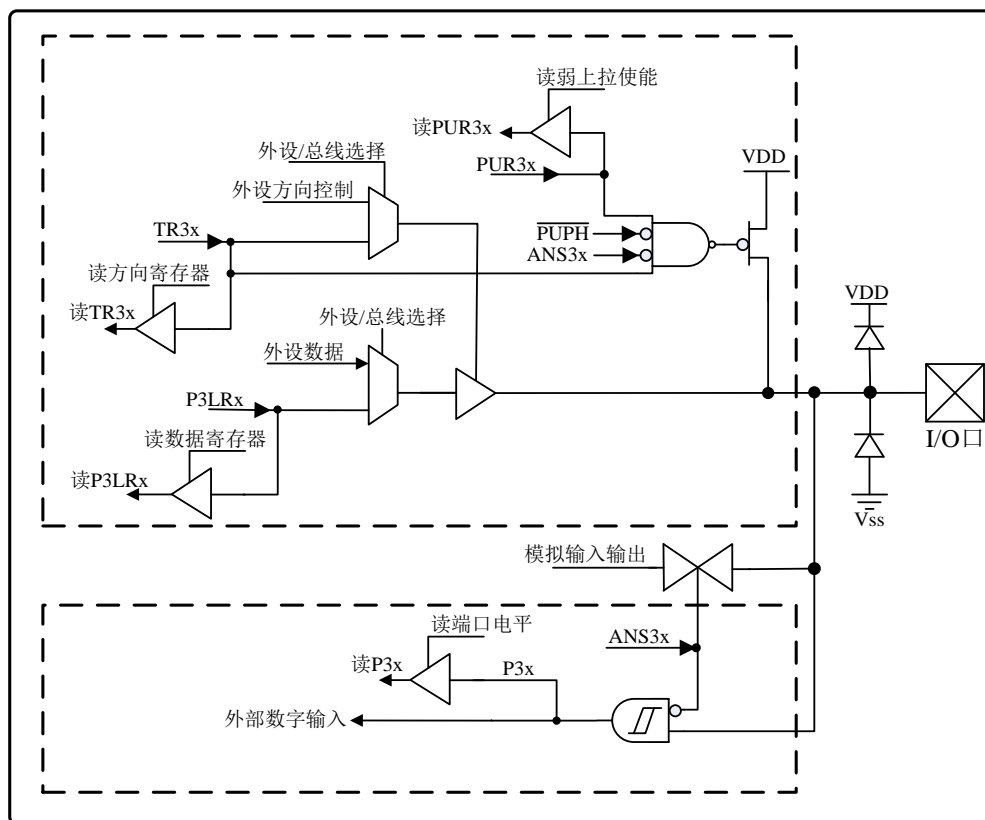


图 2.6 P3 口引脚原理功能框图



## 2.6 P4 口

P4 口共有 3 个引脚，所有管脚均可作为普通 I/O 口。

### 2.6.1 P4 口相关的寄存器

表 2.5 与 P4 口相关的寄存器

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
09H	P4	-	-	P45	P44	P43	-	-	-
49H	P4LR	-	-	P4LR5	P4LR4	P4LR3	-	-	-
29H	TR4	-	-	TR45	TR44	TR43	-	-	-
244H	PUR4	-	-	PUR45	PUR44	PUR43	-	-	-
4EH	ANS4	-	-	ANS45	ANS44	ANS43	-	-	-

#### 2.6.1.1 P4 口状态寄存器(P4)

寄存器 P4 各位对应端口 P4 口各引脚作为普通 I/O 口时的状态。如下寄存器所示：

**寄存器：** P4: P4口状态寄存器(地址: 09H)

		bit7						bit0	
复位值 xxxx xxxx		-	-	P45	P44	P43	-	-	-
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P4x: P4 口各引脚状态位

1 = 对应引脚为逻辑高电平

0 = 对应引脚为逻辑低电平

#### 2.6.1.2 P4 口输出锁存寄存器 (P4LR)

寄存器 P4LR 是 P4 口输出锁存寄存器。在 P4 口作为输出时，我们是通过写 P4LR 寄存器来设置输出 P4 口的状态。

**寄存器：** P4LR: P4口输出锁存寄存器(地址:49H)

		bit7					bit0	
复位值 xxxx xxxx		-	-	P4LR5	P4LR4	P4LR3	-	-
		R/W	R/W	R/W	R/W	R/W	R/W	R/W

P4LRx: 写 P4 口输出状态

1 = 对应引脚输出高电平

0 = 对应引脚输出低电平

### 2.6.1.3 P4 口方向控制寄存器(TR4)

通过将寄存器 TR4 中的某位置 1，将对应管脚设置为输入口。清 0 设置为输出口。

**寄存器:** TR4: P4口方向控制寄存器(地址: 29H)

	bit7						bit0	
复位值 --11 1--	-	-	TR45	TR44	TR43	-	-	-
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TR4x: P4 口各引脚方向控制位

1 = P4 口对应引脚被配置为输入端口

0 = P4 口对应引脚被配置为输出端口

### 2.6.1.4 P4 口模拟/数字口设置寄存器 (ANS4)

P4 口模拟/数字口设置寄存器 ANS4 用于将 P4 口设置为模拟口或者数字口，通过将 ANS4 某位置 1，将对应的引脚设置为模拟口，清 0 设置为数字 I/O 口。

**寄存器:** ANS4: P4口模拟/数字口设置寄存器(地址:4EH)

	bit7						bit0	
复位值 -11 1--	-	-	ANS45	ANS44	ANS43	-	-	-
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ANS4x: P4 口各引脚模拟/数字口设置位

1 = 将对应引脚配置为模拟口

0 = 将对应引脚配置为数字口或者特殊功能引脚

注:

- 1.端口配置为模拟口时，端口方向必须配置为输入状态；
- 2.当P4口配置为模拟（输入）口时，P4寄存器和P4LR寄存器将无效；即无法通过P4寄存器读端口状态，也无法通过P4LR寄存器对端口进行电平控制。

### 2.6.1.5 P4 上拉功能控制寄存器(PUR4)

如果要将某引脚的上拉功能打开，需要先将 $\overline{\text{PUPH}}$ (上拉功能总使能位)位清 0，允许 P4 口上拉功能打开，然后再将要打开上拉功能的引脚，所对应的上拉功能控制位置 1 即可。PUR4 为上拉功能控制寄存器。

**寄存器:** PUR4: P4口弱上拉控制寄存器(地址: 244H)

	bit7						bit0	
复位值 --11 1--	-	-	PUR45	PUR44	PUR43	-	-	-
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUR4x: 上拉功能使能位

1 = 使能对应的端口上拉功能

0 = 禁止对应的端口上拉功能

2.6.2 P4 口原理功能框图

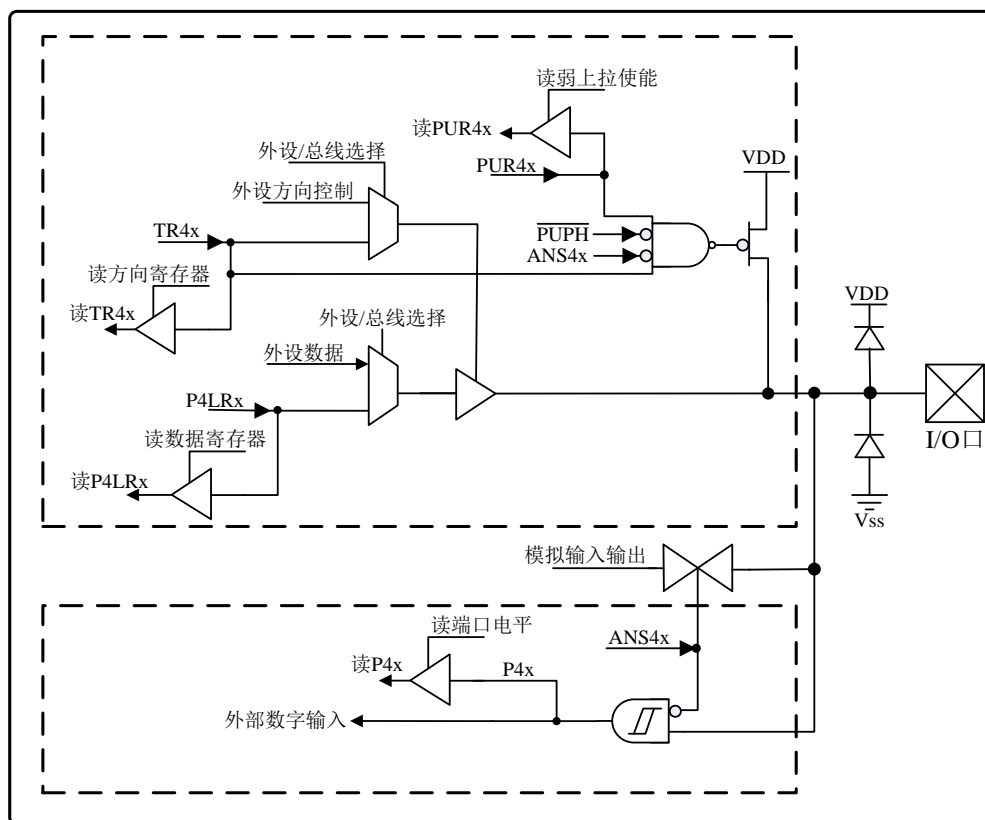


图 2.7 P4 口引脚原理功能框图

## 2.7 P5 口

P5 口共有 6 个引脚，所有管脚均可做普通的 I/O 口。

### 2.7.1 P5 口相关的寄存器

表 2.6 与 P5 口相关的寄存器

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
105H	P5	-	-	P55	P54	P53	P52	P51	P50
145H	P5LR	-	-	P5LR5	P5LR4	P5LR3	P5LR2	P5LR1	P5LR0
125H	TR5	-	-	TR55	TR54	TR53	TR52	TR51	TR50
14AH	ANS5	-	-	ANS55	ANS54	ANS53	ANS52	ANS51	ANS50
245H	PUR5	-	-	PUR55	PUR54	PUR53	PUR52	PUR51	PUR50

#### 2.7.1.1 P5 口状态寄存器(P5)

寄存器 P5 各位对应端口 P5 口各引脚作为普通 I/O 口时的状态。

**寄存器:** P5: P5口状态寄存器(地址: 105H)

复位值 xxxx xxxx	bit7							bit0
	-	-	P55	P54	P53	P52	P51	P50
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P5x: P5 口各引脚状态位

1 = 对应引脚为逻辑高电平

0 = 对应引脚为逻辑低电平

#### 2.7.1.2 P5 口输出锁存寄存器 (P5LR)

寄存器 P5LR 是 P5 口输出锁存寄存器。在 P5 口作为输出时，我们是通过写 P5LR 寄存器来设置输出 P5 口的状态。

**寄存器:** P5LR: P5口输出锁存寄存器(地址:145H)

复位值 xxxx xxxx	bit7							bit0
	-	-	P5LR5	P5LR4	P5LR3	P5LR2	P5LR1	P5LR0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P5LRx: 写 P5 口输出状态

1 = 对应引脚输出高电平

0 = 对应引脚输出低电平

### 2.7.1.3 P5 口方向控制寄存器(TR5)

通过将寄存器 TR5 中的某位置 1，将对应管脚设置为输入口。清 0 设置为输出口。

**寄存器:** TR5: P5口方向控制寄存器(地址: 125H)

		bit7						bit0	
复位值		-	-	TR55	TR54	TR53	TR52	TR51	TR50
--11 1111		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TR5x: P5 口各引脚方向控制位

1 = P5 口对应引脚被配置为输入端口

0 = P5 口对应引脚被配置为输出端口

### 2.7.1.4 P5 口模拟/数字口设置寄存器 (ANS5)

P5 口模拟/数字口设置寄存器 ANS5 用于将 P5 口设置为模拟口或者数字口，通过将 ANS5 某位置 1，将对应的引脚设置为模拟口，清 0 设置为数字 I/O 口。

**寄存器:** ANS5: P5口模拟/数字口设置寄存器(地址:14AH)

		bit7						bit0	
复位值		-	-	ANS55	ANS54	ANS53	ANS52	ANS51	ANS50
--11 1111		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ANS5x: P5 口各引脚模拟/数字口设置位

1 = 将对应引脚配置为模拟口

0 = 将对应引脚配置为数字口或者特殊功能引脚

注:

- 1.端口配置为模拟口时，端口方向必须配置为输入状态；
- 2.当P5口配置为模拟（输入）口时，P5寄存器和P5LR寄存器将无效；即无法通过P5寄存器读端口状态，也无法通过P5LR寄存器对端口进行电平控制。

### 2.7.1.5 P5 上拉功能控制寄存器(PUR5)

如果要将某引脚的上拉功能打开，需要先将 $\overline{\text{PUPH}}$ (上拉功能总使能位)位清 0，允许 P5 口上拉功能打开，然后再将打开上拉功能的引脚，所对应的上拉功能控制位置 1 即可。

**寄存器:** PUR5: P5口弱上拉控制寄存器(地址: 245H)

		bit7						bit0	
复位值		-	-	PUR55	PUR54	PUR53	PUR52	PUR51	PUR50
--11 1111		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUR5x: 上拉功能使能位

1 = 使能对应的端口上拉功能

0 = 禁止对应的端口上拉功能

2.7.2 P5 口原理功能框图

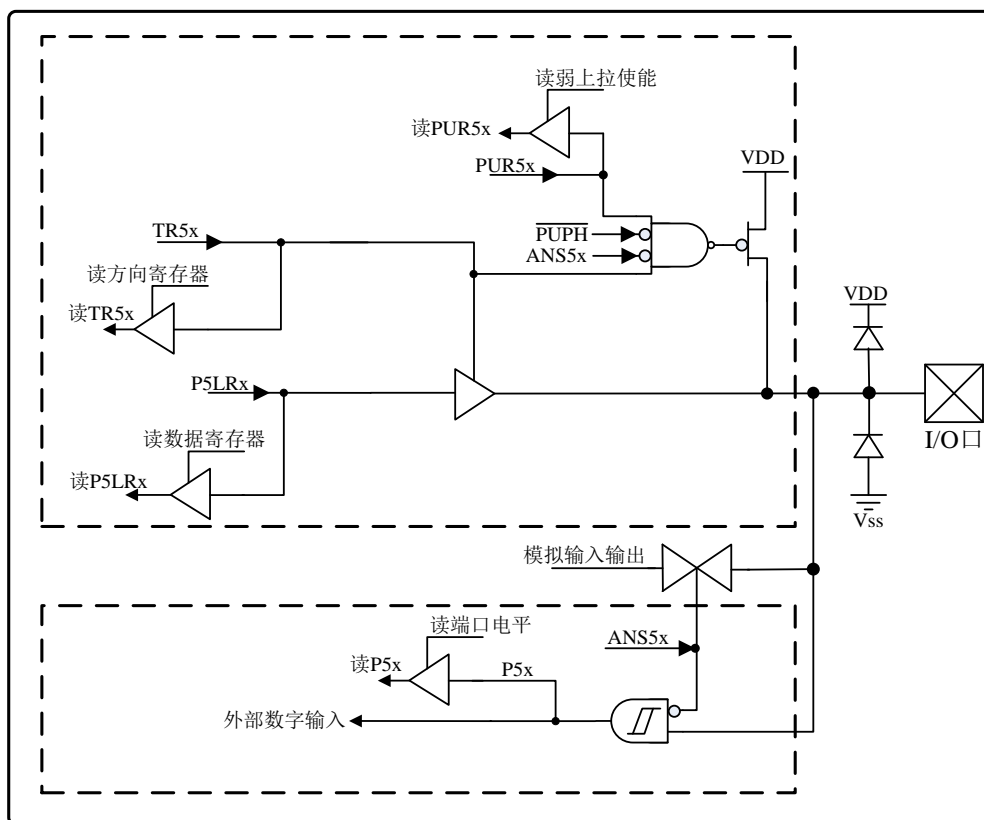


图 2.8 P5 口引脚原理功能框图

## 2.8 P6 口

P6 口共有 6 个引脚，所有管脚均可作为普通 I/O 口。

### 2.8.1 P6 口相关的寄存器

表 2.7 与 P6 口相关的寄存器

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
106H	P6	-	P66	P65	P64	P63	P62	P61	-
146H	P6LR	-	P6LR6	P6LR5	P6LR4	P6LR3	P6LR2	P6LR1	-
126H	TR6	-	TR66	TR65	TR64	TR63	TR62	TR61	-
14BH	ANS6	-	ANS66	ANS65	ANS64	ANS63	ANS62	ANS61	-
246H	PUR6	-	PUR66	PUR65	PUR64	PUR63	PUR62	PUR61	-

#### 2.8.1.1 P6 口状态寄存器(P6)

寄存器 P6 各位对应端口 P6 口各引脚作为普通 I/O 口时的状态。

**寄存器:** P6: P6口状态寄存器(地址: 106H)

		bit7						bit0
复位值	-	P66	P65	P64	P63	P62	P61	-
xxxx xxxx	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P6x: P6 口各引脚状态位

1 = 对应引脚为逻辑高电平

0 = 对应引脚为逻辑低电平

#### 2.8.1.2 P6 口输出锁存寄存器 (P6LR)

寄存器 P6LR 是 P6 口输出锁存寄存器。在 P6 口作为输出时，我们是通过写 P6LR 寄存器来设置输出 P6 口的状态。

**寄存器:** P6LR: P6口输出锁存寄存器(地址:146H)

		bit7						bit0
复位值	-	P6LR6	P6LR5	P6LR4	P6LR3	P6LR2	P6LR1	-
xxxx xxxx	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P6LRx: 写 P6 口输出状态

1 = 对应引脚输出高电平

0 = 对应引脚输出低电平

### 2.8.1.3 P6 口方向控制寄存器(TR6)

如下寄存器所示，通过将寄存器 TR6 中的某位置 1，将对应管脚设置为输入口。清 0 设置为输出口。

**寄存器:** TR6: P6口方向控制寄存器(地址: 126H)

		bit7						bit0
复位值	-	TR66	TR65	TR64	TR63	TR62	TR61	-
-111 111-	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TR6x: P6 口各引脚方向控制位

1 = P6 口对应引脚被配置为输入端口

0 = P6 口对应引脚被配置为输出端口

### 2.8.1.4 P6 口模拟/数字口设置寄存器 (ANS6)

P6 口模拟/数字口设置寄存器 ANS6 用于将 P6 口设置为模拟口或者数字口，通过将 ANS6 某位置 1，将对应的引脚设置为模拟口，清 0 设置为数字 I/O 口。

**寄存器:** ANS6: P6口模拟/数字口设置寄存器(地址:14BH)

		bit7						bit0
复位值	-	ANS66	ANS65	ANS64	ANS63	ANS62	ANS61	-
-111 111-	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ANS6x: P6 口各引脚模拟/数字口设置位

1 = 将对应引脚配置为模拟口

0 = 将对应引脚配置为数字口或者特殊功能引脚

注:

- 1.端口配置为模拟口时，端口方向必须配置为输入状态；
- 2.当P6口配置为模拟（输入）口时，P6寄存器和P6LR寄存器将无效；即无法通过P6寄存器读端口状态，也无法通过P6LR寄存器对端口进行电平控制。

### 2.8.1.5 P6 上拉功能控制寄存器(PUR6)

如果要将某引脚的上拉功能打开，需要先将 $\overline{\text{PUPH}}$ (上拉功能总使能位)位清 0，允许 P6 口上拉功能打开，然后再将要打开上拉功能的引脚，所对应的上拉功能控制位置 1 即可。

**寄存器:** PUR6: P6口弱上拉控制寄存器(地址: 246H)

		bit7						bit0
复位值	-	PUR66	PUR65	PUR64	PUR63	PUR62	PUR61	-
-111 111-	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUR6x: 上拉功能使能位

1 = 使能对应的端口上拉功能

0 = 禁止对应的端口上拉功能



2.8.2 P6 口原理功能框图

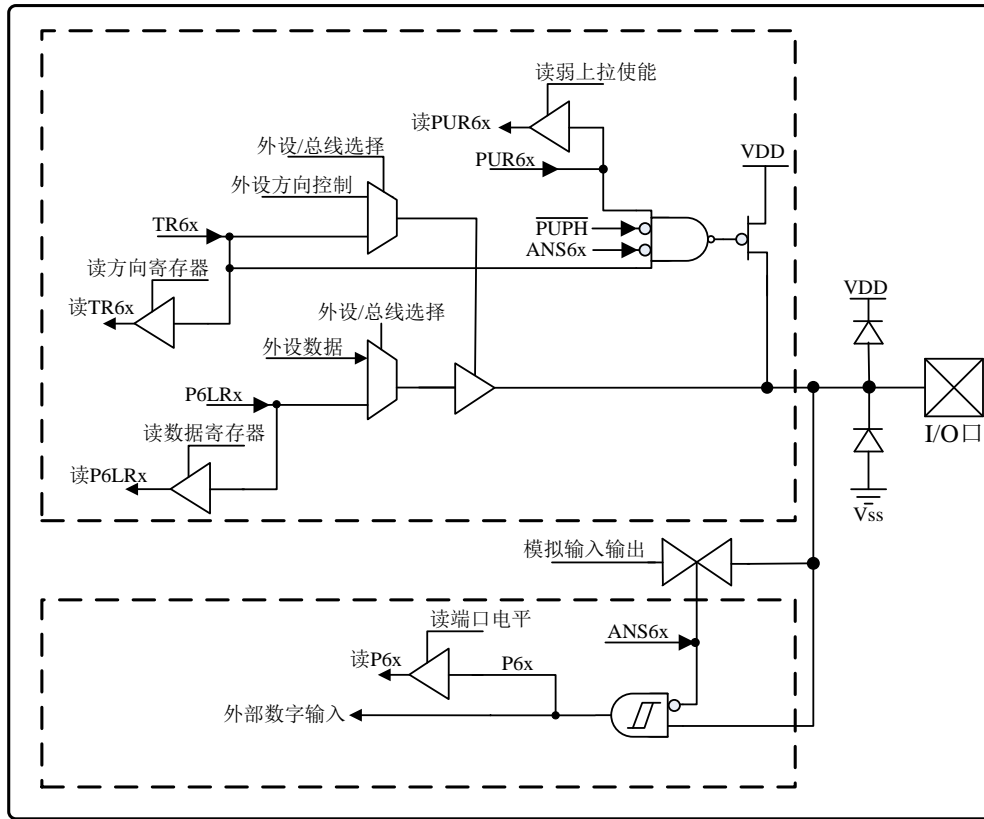


图 2.9 P6 口引脚原理功能框图

## 2.9 P7 口

P7 口共有 4 个引脚，所有管脚均可作为普通 I/O 口。

### 2.9.1 P7 口相关的寄存器

表 2.8 与 P7 口相关的寄存器

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
107H	P7	-	-	P75	-	P73	P72	P71	-
147H	P7LR	-	-	P7LR5	-	P7LR3	P7LR2	P7LR1	-
127H	TR7	-	-	TR75	-	TR73	TR72	TR71	-
247H	PUR7	-	-	PUR75	-	PUR73	PUR72	PUR71	-
14CH	ANS7	-	-	ANS75	-	ANS73	ANS72	ANS71	-

#### 2.9.1.1 P7 口状态寄存器(P7)；通过 I/O 口到 P7 寄存器

寄存器 P7 各位对应端口 P7 口各引脚作为普通 I/O 口时的状态。

**寄存器：** P7: P7口状态寄存器(地址: 107H)

复位值 xxxx xxxx	bit7							bit0
	-	-	P75	-	P73	P72	P71	-
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P7x: P7 口各引脚状态位

1 = 对应引脚为逻辑高电平

0 = 对应引脚为逻辑低电平

#### 2.9.1.2 P7 口输出锁存寄存器 (P7LR) 通过 P7 到 I/O 口寄存器

寄存器 P7LR 是 P7 口输出锁存寄存器。在 P7 口作为输出时，我们是通过写 P7LR 寄存器来设置输出 P7 口的状态。

**寄存器：** P7LR: P7口输出锁存寄存器(地址:147H)

复位值 xxxx xxxx	bit7							bit0
	-	-	P7LR5	-	P7LR3	P7LR2	P7LR1	-
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P7LRx: 写 P7 口输出状态

1 = 对应引脚输出高电平

0 = 对应引脚输出低电平

### 2.9.1.3 P7 口方向控制寄存器(TR7)

通过将寄存器 TR7 中的某位置 1，将对应管脚设置为输入口。清 0 设置为输出口。

**寄存器:** TR7: P7口方向控制寄存器(地址: 127H)

	bit7						bit0	
复位值 --1- 111-	-	-	TR75	-	TR73	TR72	TR71	-
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TR7x: P7 口各引脚方向控制位

1 = P7 口对应引脚被配置为输入端口

0 = P7 口对应引脚被配置为输出端口

### 2.9.1.4 P7 口模拟/数字口设置寄存器 (ANS7)

P7 口模拟/数字口设置寄存器 ANS7 用于将 P7 口设置为模拟口或者数字口，通过将 ANS7 某位置 1，将对应的引脚设置为模拟口，清 0 设置为数字 I/O 口。

**寄存器:** ANS7: P7口模拟/数字口设置寄存器(地址:14CH)

	bit7						bit0	
复位值 --1- 111-	-	-	ANS75	-	ANS73	ANS72	ANS71	-
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ANS7x: P7 口各引脚模拟/数字口设置位

1 = 将对应引脚配置为模拟口

0 = 将对应引脚配置为数字口或者特殊功能引脚

注:

- 1.端口配置为模拟口时，端口方向必须配置为输入状态，包括ANS7的bit6和bit7；
- 2.当P7口配置为模拟（输入）口时，P7寄存器和P7LR寄存器将无效；即无法通过P7寄存器读端口状态，也无法通过P7LR寄存器对端口进行电平控制。

### 2.9.1.5 P7 上拉功能控制寄存器(PUR7)

如果要将某引脚的上拉功能打开，需要先将 $\overline{\text{PUPH}}$ (上拉功能总使能位)位清 0，允许 P7 口上拉功能打开，然后再将要打开上拉功能的引脚，所对应的上拉功能控制位置 1 即可。

**寄存器:** PUR7: P7口弱上拉控制寄存器(地址: 247H)

	bit7						bit0	
复位值 --1- 111-	-	-	PUR75	-	PUR73	PUR72	PUR71	-
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUR7x: 上拉功能使能位

1 = 使能对应的端口上拉功能

0 = 禁止对应的端口上拉功能

2.9.2 P7 口原理功能框图

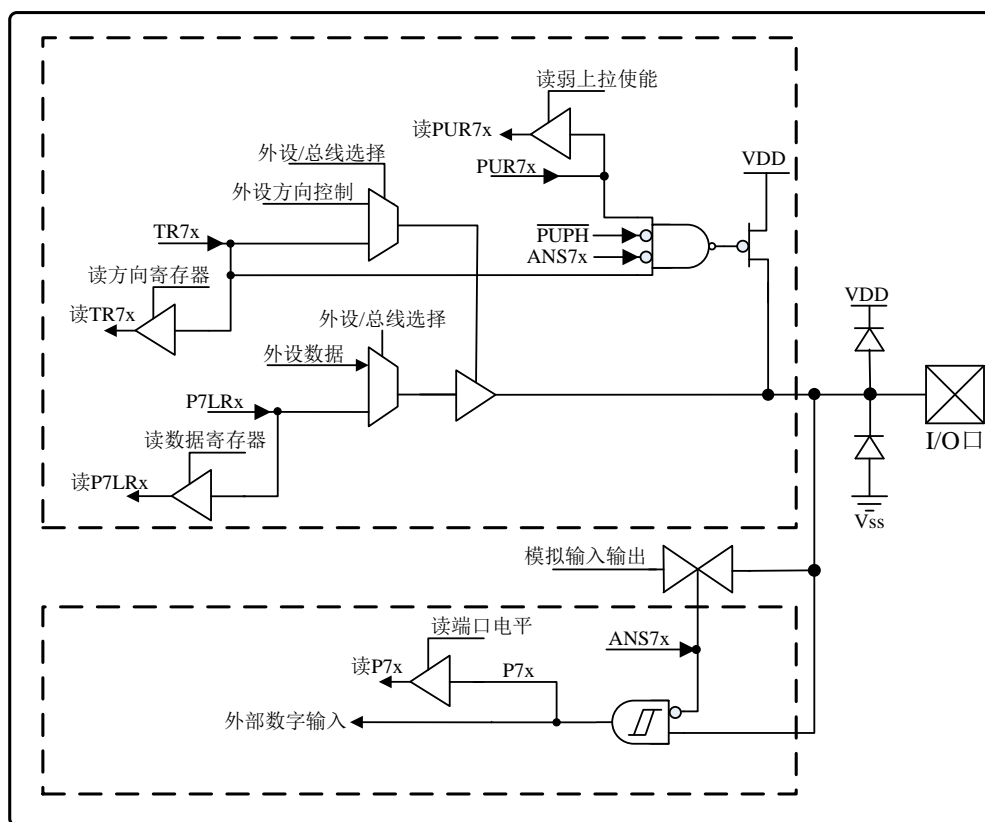


图 2.10 P7 口引脚原理功能框图

## 2.10 P9 口

P9 口共有 6 个引脚，所有管脚均可作为普通 I/O 口。

### 2.10.1 P9 口相关的寄存器

表 2.9 与 P9 口相关的寄存器

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
109H	P9	-	-	P95	P94	P93	P92	P91	P90
149H	P9LR	-	-	P9LR5	P9LR4	P9LR3	P9LR2	P9LR1	P9LR0
129H	TR9	-	-	TR95	TR94	TR93	TR92	TR91	TR90
240H	PUR9	-	-	PUR95	PUR94	PUR93	PUR92	PUR91	PUR90
14EH	ANS9	-	-	ANS95	ANS94	ANS93	ANS92	ANS91	ANS90

#### 2.10.1.1 P9 口状态寄存器(P9)

寄存器 P9 各位对应端口 P9 口各引脚作为普通 I/O 口时的状态。

**寄存器:** P9: P9口状态寄存器(地址:109H)

		bit7						bit0	
复位值	-	-	P95	P94	P93	P92	P91	P90	
--xx xxxx									
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P9<5:0>: P9 口各引脚状态位  
 1 = 对应引脚为逻辑高电平  
 0 = 对应引脚为逻辑低电平

#### 2.10.1.2 P9 口输出锁存寄存器 (P9LR)

寄存器 P9LR 是 P9 口输出锁存寄存器。在 P9 口作为输出时，我们是通过写 P9LR 寄存器来设置输出 P9 口的状态。

**寄存器:** P9LR: P9口输出锁存寄存器(地址:149H)

		bit7						bit0	
复位值	-	-	P9LR5	P9LR4	P9LR3	P9LR2	P9LR1	P9LR0	
--xx xxxx									
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P9LR<5:0>: 写 P9 口输出状态  
 1 = 对应引脚输出高电平  
 0 = 对应引脚输出低电平

### 2.10.1.3 P9 口方向控制寄存器(TR9)

通过将寄存器 TR9 中的某位置 1，将对应管脚设置为输入口。清 0 设置为输出口。

**寄存器:** TR9: P9口方向控制寄存器(地址: 129H)

	bit7							bit0
复位值 --11 1111	-	-	TR95	TR94	TR93	TR92	TR91	TR90
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TR9<5:0>: P9 口各引脚方向控制位  
 1 = P9 口对应引脚被配置为输入端口  
 0 = P9 口对应引脚被配置为输出端口

### 2.10.1.4 P9 口模拟/数字口设置寄存器 (ANS9)

P9 口模拟/数字口设置寄存器 ANS9 用于将 P9 口设置为模拟口或者数字口，通过将 ANS9 某位置 1，将对应的引脚设置为模拟口，清 0 设置为数字 I/O 口。

**寄存器:** ANS9: P9口模拟/数字口设置寄存器(地址:14EH)

	bit7							bit0
复位值 --11 1111	-	-	ANS95	ANS94	ANS93	ANS92	ANS91	ANS90
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ANS9<5:0>: P9 口各引脚模拟/数字口设置位  
 1 = 将对应引脚配置为模拟口  
 0 = 将对应引脚配置为数字口或者特殊功能引脚

注:

- 1.端口配置为模拟口时，端口方向必须配置为输入状态；
- 2.当P9口配置为模拟（输入）口时，P9寄存器和P9LR寄存器将无效；即无法通过P9寄存器读端口状态，也无法通过P9LR寄存器对端口进行电平控制。

### 2.10.1.5 P9 上拉功能控制寄存器(PUR9)

如果要将某引脚的上拉功能打开，需要先将 PUPH (上拉功能总使能位)位清 0，允许 P9 口上拉功能打开，然后再将要打开上拉功能的引脚，所对应的上拉功能控制位置 1 即可。

**寄存器:** PUR9: P9口弱上拉控制寄存器(地址: 240H)

	bit7							bit0
复位值 --11 1111	-	-	PUR95	PUR94	PUR93	PUR92	PUR91	PUR90
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUR9<5:0>: 上拉功能使能位  
 1 = 使能对应的端口上拉功能  
 0 = 禁止对应的端口上拉功能

2.10.2 P9 口原理功能框图

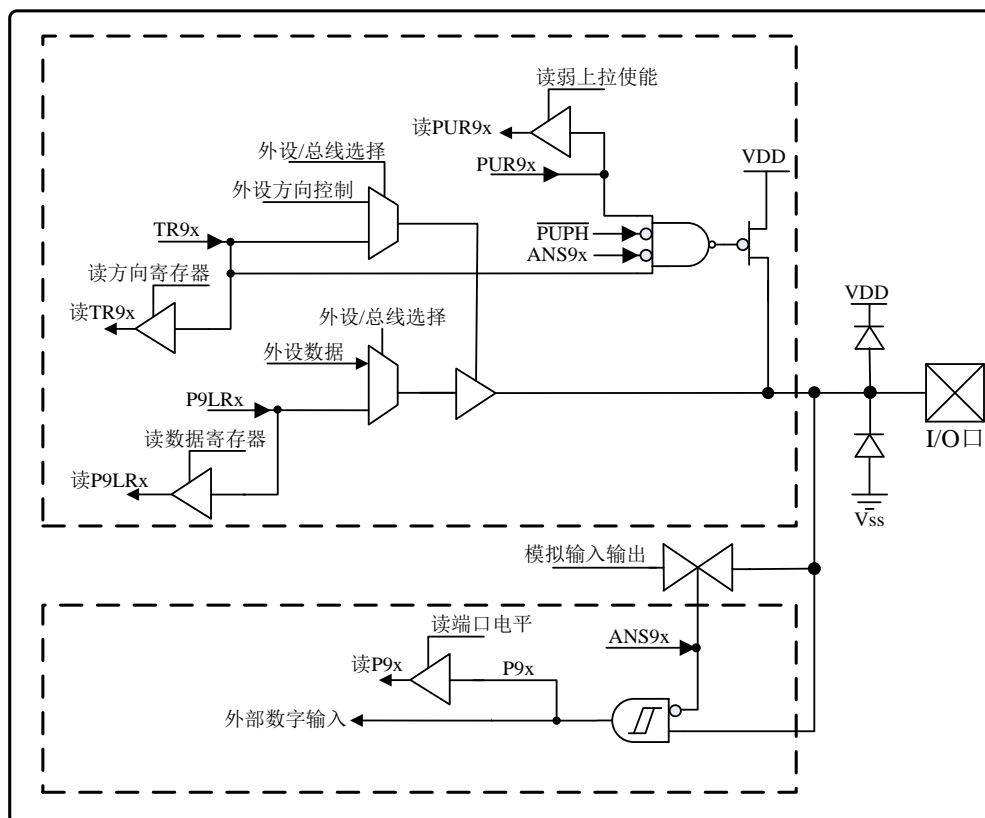


图 2.11 P9 口引脚原理功能框图

## 2.11 PA 口

P10 口共有 2 个引脚，所有管脚均可作为普通 I/O 口。

### 2.11.1 PA 口相关的寄存器

表 2.10 与 PA 口相关的寄存器

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
424H	PA	-	-	-	-	-	-	PA1	PA0
425H	PALR	-	-	-	-	-	-	PALR1	PALR0
426H	TRA	-	-	-	-	-	-	TRA1	TRA0
427H	PURA	-	-	-	-	-	-	PURA1	PURA0
428H	ANSA	-	-	-	-	-	-	ANSA1	ANSA0

#### 2.11.1.1 PA 口状态寄存器(PA)

寄存器 PA 各位对应端口 PA 口各引脚作为普通 I/O 口时的状态。

**寄存器:** PA: PA口状态寄存器(地址:424H)

		bit7						bit0	
复位值	xxxx xxxx	-	-	-	-	-	-	PA1	PA0
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**PAx:** PA 口各引脚状态位  
 1 = 对应引脚为逻辑高电平  
 0 = 对应引脚为逻辑低电平

#### 2.11.1.2 PA 口输出锁存寄存器 (PALR)

寄存器 PALR 是 PA 口输出锁存寄存器。在 PA 口作为输出时，我们是通过写 PALR 寄存器来设置输出 PA 口的状态。

**寄存器:** PALR: PA口输出锁存寄存器(地址:425H)

		bit7						bit0	
复位值	xxxx xxxx	-	-	-	-	-	-	PALR1	PALR0
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**PALRx:** 写 PA 口输出状态  
 1 = 对应引脚输出高电平  
 0 = 对应引脚输出低电平



### 2.11.1.3 PA 口方向控制寄存器(TRA)

通过将寄存器 TRA 中的某位置 1，将对应管脚设置为输入口。清 0 设置为输出口。

**寄存器:** TRA: PA口方向控制寄存器(地址: 426H)

	bit7						bit0	
复位值 -----11	-	-	-	-	-	-	TRA1	TRA0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**TRAx:** PA 口各引脚方向控制位  
 1 = PA 口对应引脚被配置为输入端口  
 0 = PA 口对应引脚被配置为输出端口

### 2.11.1.4 PA 口模拟/数字口设置寄存器 (ANSA)

PA 口模拟/数字口设置寄存器 ANSA 用于将 PA 口设置为模拟口或者数字口，通过将 ANSA 某位置 1，将对应的引脚设置为模拟口，清 0 设置为数字 I/O 口。

**寄存器:** ANSA: PA口模拟/数字口设置寄存器(地址:428H)

	bit7						bit0	
复位值 -----11	-	-	-	-	-	-	ANSA1	ANSA0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**ANSAx:** PA 口各引脚模拟/数字口设置位  
 1 = 将对应引脚配置为模拟口  
 0 = 将对应引脚配置为数字口或者特殊功能引脚

注:

- 1.端口配置为模拟口时，端口方向必须配置为输入状态；
- 2.当PA口配置为模拟（输入）口时，PA寄存器和PALR寄存器将无效；即无法通过PA寄存器读端口状态，也无法通过PALR寄存器对端口进行电平控制。

### 2.11.1.5 PA 上拉功能控制寄存器(PURA)

如果要将某引脚的上拉功能打开，需要先将 $\overline{PUPH}$ (上拉功能总使能位)位清 0，允许 PA 口上拉功能打开，然后再将要打开上拉功能的引脚，所对应的上拉功能控制位置 1 即可。

**寄存器:** PURA: PA口弱上拉控制寄存器(地址:427 H)

	bit7						bit0	
复位值 -----11	-	-	-	-	-	-	PURA1	PURA0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**PURAx:** 上拉功能使能位  
 1 = 使能对应的端口上拉功能  
 0 = 禁止对应的端口上拉功能

2.11.2 PA 口原理功能框图

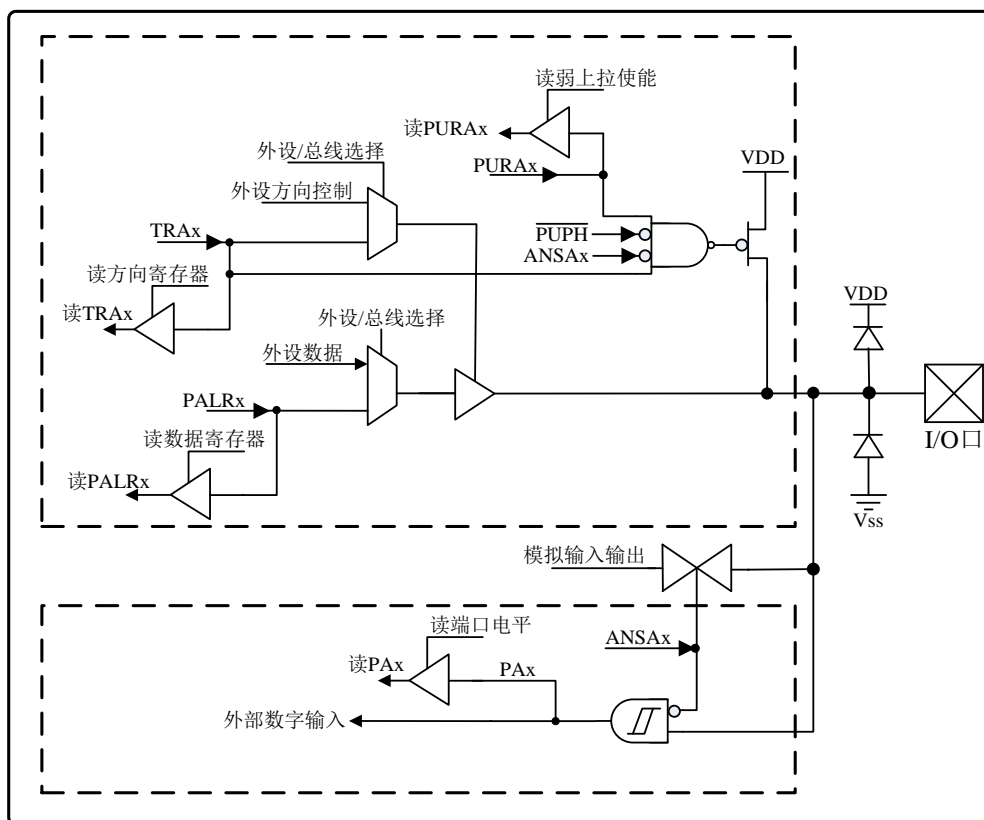


图 2.12 PA 口引脚原理功能框图

## 2.12 PB 口

PB 口共有 2 个引脚，所有管脚均可作为普通 I/O 口。

### 2.12.1 PB 口相关的寄存器

表 2.11 与 PB 口相关的寄存器

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
429H	PB	-	-	-	-	-	-	PB1	PB0
42AH	PBLR	-	-	-	-	-	-	PBLR1	PBLR0
42BH	TRB	-	-	-	-	-	-	TRB1	TRB0
42CH	PURB	-	-	-	-	-	-	PURB1	PURB0
42DH	ANSB	-	-	-	-	-	-	ANSB1	ANSB0

#### 2.12.1.1 PB 口状态寄存器(PB)

寄存器 PB 各位对应端口 PB 口各引脚作为普通 I/O 口时的状态。

**寄存器:** PB: PB口状态寄存器(地址:429H)

复位值 xxxx xxxx	bit7						bit0	
	-	-	-	-	-	-	PB1	PB0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**PBx:** PB 口各引脚状态位  
 1 = 对应引脚为逻辑高电平  
 0 = 对应引脚为逻辑低电平

#### 2.12.1.2 PB 口输出锁存寄存器 (PBLR)

寄存器 PBLR 是 PB 口输出锁存寄存器。在 PB 口作为输出时，我们是通过写 PBLR 寄存器来设置输出 PB 口的状态。

**寄存器:** PBLR: PB口输出锁存寄存器(地址:42AH)

复位值 xxxx xxxx	bit7						bit0	
	-	-	-	-	-	-	PBLR1	PBLR0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**PBLRx:** 写 PB 口输出状态  
 1 = 对应引脚输出高电平  
 0 = 对应引脚输出低电平

### 2.12.1.3 B 口方向控制寄存器(TRB)

通过将寄存器 TRB 中的某位置 1，将对应管脚设置为输入口。清 0 设置为输出口。

**寄存器:** TRB: PB口方向控制寄存器(地址: 42BH)

	bit7						bit0	
复位值 -----11	-	-	-	-	-	-	TRB1	TRB0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**TRBx:** PB 口各引脚方向控制位  
 1 = PB 口对应引脚被配置为输入端口  
 0 = PB 口对应引脚被配置为输出端口

### 2.12.1.4 PB 口模拟/数字口设置寄存器 (ANSB)

PB 口模拟/数字口设置寄存器 ANSB 用于将 PB 口设置为模拟口或者数字口，通过将 ANSB 某位置 1，将对应的引脚设置为模拟口，清 0 设置为数字 I/O 口。

**寄存器:** ANSB: PB口模拟/数字口设置寄存器(地址:42DH)

	bit7						bit0	
复位值 -----11	-	-	-	-	-	-	ANSB1	ANSB0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**ANSBx:** PB 口各引脚模拟/数字口设置位  
 1 = 将对应引脚配置为模拟口  
 0 = 将对应引脚配置为数字口或者特殊功能引脚

注:

- 1.端口配置为模拟口时，端口方向必须配置为输入状态；
- 2.当PB口配置为模拟（输入）口时，PB寄存器和PBLR寄存器将无效；即无法通过PB寄存器读端口状态，也无法通过PBLR寄存器对端口进行电平控制。

### 2.12.1.5 PB 上拉功能控制寄存器(PURB)

如果要将某引脚的上拉功能打开，需要先将  $PURPH$  (上拉功能总使能位)位清 0，允许 PB 口上拉功能打开，然后再将要打开上拉功能的引脚，所对应的上拉功能控制位置 1 即可。

**寄存器:** PURB: PB口弱上拉控制寄存器(地址: 42CH)

	bit7						bit0	
复位值 -----11	-	-	-	-	-	-	PURB1	PURB0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**PURBx:** 上拉功能使能位  
 1 = 使能对应的端口上拉功能  
 0 = 禁止对应的端口上拉功能

2.12.2 PB 口原理功能框图

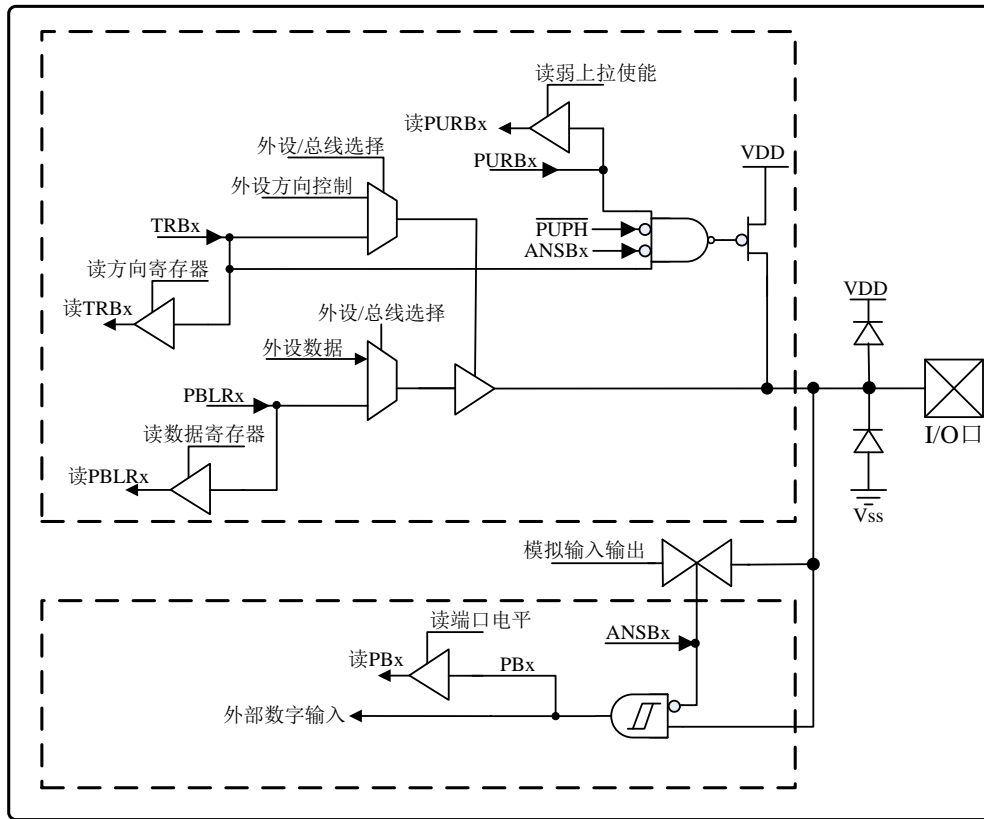


图 2.13 PB 口引脚原理功能框图

## 2.13 复用引脚功能

设置 APFCTL 寄存器可以使外设等输入输出功能映射到到相应的引脚上。

### 2.13.1 复用引脚功能相关寄存器

表 2.12 引脚复用寄存器列表

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
45EH	APFCTL2	P7CH3HSEL	P7CH2HSEL	P7CH1HSEL	P7CH0HSEL	P6CH3HSEL	P6CH2HSEL	P6CH1HSEL	P6CH0HSEL
45FH	APFCTL3	P3DSEL	P3CSEL1	P3CSEL0	P3BSEL1	P3BSEL0	P3ASEL	CCP7SEL1	CCP7SEL0
460H	APFCTL4	P4DSEL1	P4DSEL0	P4CSEL1	P4CSEL0	P4BSEL1	P4BSEL0	P4ASEL1	P4ASEL0
461H	APFCTL5	P5CSEL	P5BSEL1	P5BSEL0	P5ASEL1	P5ASEL0	CCP3SEL1	CCP3SEL0	T0CKSEL
462H	APFCTL6	CCP10SEL1	CCP10SEL0	CCP9SEL2	CCP9SEL1	CCP9SEL0	CCP8SEL1	CCP8SEL0	INTCCPSEL
463H	APFCTL7	IOC7SEL	IOC6SEL	IOC5SEL	IOC4SEL	IOC3SEL	IOC2SEL	IOC1SEL	IOC0SEL
464H	APFCTL8	IOC14SEL	IOC13SEL	IOC12SEL	IOC11SEL	IOC10SEL	TICKSEL	T1GSEL1	T1GSEL0
465H	APFCTL9	SDIDA2SEL 1	SDIDA2SEL 0	SCKCL2SEL	SDO2SEL1	SDO2SEL0	SS2SEL1	SS2SEL0	CLKOUTSEL L
466H	APFCTL10	SDIDA1SEL 1	SDIDA1SEL 0	SCKCL1SEL 1	SCKCL1SEL 0	SDO1SEL	SS1SEL	PWM2OUTS EL	PWM1OUTS EL
467H	APFCTL11	-	-	ADVRINSEL1 L1	ADVRINSEL0 L0	-	-	-	-
468H	APFCTL12	C3OUTSEL	C2OUTSEL1	C2OUTSEL0	C1OUTSEL	CK3TX3SEL	DT3RX3SEL	CK4TX4SEL	DT4RX4SEL

#### 2.13.1.1 复用引脚功能控制寄存器 APFCTL2

**寄存器:** APFCTL2: 复用引脚功能控制寄存器1(地址: 45EH)

复位值 0000 0000	bit7							bit0
	P7CH3HS EL	P7CH2HS EL	P7CH1HS EL	P7CH0HS EL	P6CH3HS EL	P6CH2HS EL	P6CH1HS EL	P6CH0HS EL
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P7CH3HSEL: P7CH3H 输出功能选择位

0= 保留

1= P7CH3H 功能位于 P1.3

P7CH2HSEL: P7CH2H 输出功能选择位

0= 保留

1= P7CH2H 功能位于 P2.0

P7CH1HSEL: P7CH1H 输出功能选择位

0= 保留

1= P7CH1H 功能位于 P2.3

P7CH0HSEL: P7CH0H 输出功能选择位

CCP2 捕捉模式输入通道选择位

0= P7CH0H 功能和 CCP2 捕捉输入通道位于 P9.4

- 1= P7CH0H 功能和 CCP2 捕捉输入通道位于 P1.1
- P6CH3HSEL: P6CH3H 输出功能选择位  
0= P6CH3H 功能位于 P6.1  
1= P6CH3H 功能位于 P1.2
- P6CH2HSEL: P6CH2H 输出功能选择位  
0= P6CH2H 功能位于 P6.3  
1= P6CH2H 功能位于 P2.1
- P6CH1HSEL: P6CH1H 输出功能选择位  
0= P6CH1H 功能位于 P6.5  
1= P6CH1H 功能位于 P2.4
- P6CH0HSEL: P6CH0H 输出功能选择位  
CCP5 捕捉模式输入通道选择位  
0= 保留  
1= P6CH0H 功能和 CCP5 捕捉输入通道位于 P1.0

### 2.13.1.2 复用引脚功能控制寄存器 APFCTL3

寄存器: APFCTL3: 复用引脚功能控制寄存器1(地址: 45FH)

	bit7						bit0	
复位值 0000 0000	P3DSEL	P3CSEL1	P3CSEL0	P3BSEL1	P3BSEL0	P3ASEL	CCP7SEL 1	CCP7SEL 0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- P3DSEL: P3D 输出功能引脚选择位  
0= P3D 功能位于 P3.0  
1= P3D 功能位于 P6.5
- P3CSEL<1:0>: P3C 输出功能引脚选择位  
00=保留  
01= P3C 功能位于 P6.4  
1x= P3C 功能位于 P4.5
- P3BSEL<1:0>: P3B 输出功能引脚选择位  
00=保留  
01= P3B 功能位于 P6.3  
1x= P3B 功能位于 P4.4
- P3ASEL: P3A 输出功能引脚选择位  
0= P3A 功能位于 P1.1  
1= P3A 功能位于 P6.2
- CCP7SEL<1:0>: CCP7 功能引脚选择位  
00= CCP7 功能位于 P3.0  
01= CCP7 功能位于 P2.5  
1x= CCP7 功能位于 P1.2

## 2.13.1.3 复用引脚功能控制寄存器 APFCTL4

**寄存器:** APFCTL4: 复用引脚功能控制寄存器1(地址: 460H)

	bit7						bit0	
复位值 0000 0000	P4DSEL1	P4DSEL0	P4CSEL1	P4CSEL0	P4BSEL1	P4BSEL0	P4ASEL1	P4ASEL0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- P4DSEL<1:0>:** P4D 输出功能引脚选择位  
 00=保留  
 01=保留  
 1x= P4D 功能位于 P4.3
- P4CSEL<1:0>:** P4C 输出功能引脚选择位  
 00=保留  
 01=保留  
 1x= P4C 功能位于 P9.3
- P4BSEL<1:0>:** P4B 输出功能引脚选择位  
 00= P4B 功能位于 P9.5  
 01=保留  
 10= P4B 功能位于 P9.4  
 11=保留
- P4ASEL<1:0>:** P4A 输出功能引脚选择位  
 00= P4A 功能位于 P7.2  
 01=保留  
 10= P4A 功能位于 P0.5  
 11= P4A 功能位于 P6.1

## 2.13.1.4 复用引脚功能控制寄存器 APFCTL5

**寄存器:** APFCTL5: 复用引脚功能控制寄存器1(地址: 461H)

	bit7					bit0		
复位值 0000 0000	P5CSEL	P5BSEL1	P5BSEL0	P5ASEL1	P5ASEL0	CCP3SEL 1	CCP3SEL 0	T0CKSEL
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- P5CSEL:** P5C 输出功能引脚选择位  
 0=保留  
 1= P5C 功能位于 P6.6
- P5BSEL<1:0>:** P5B 输出功能引脚选择位  
 00=保留  
 01= P5B 功能位于 P6.5  
 1x= P5B 功能位于 P2.4
- P5ASEL<1:0>:** P5A 输出功能引脚选择位  
 00=保留  
 01= P5A 功能位于 P2.3



1x= P5A 功能位于 P0.3  
 CCP3SEL<1:0> CCP3 功能引脚选择位  
 00= CCP3 功能位于 P4.5  
 01= CCP3 功能位于 P9.5  
 10= CCP3 功能位于 P1.3  
 11= CCP3 功能位于 P2.0  
 T0CKSEL: T0CK 功能引脚选择位  
 0= T0CK 功能位于 PA.0  
 1= T0CK 功能位于 P1.1

2.13.1.5 复用引脚功能控制寄存器 APFCTL6

寄存器: APFCTL6: 复用引脚功能控制寄存器1(地址: 462H)

	bit7						bit0	
复位值 0000 0000	CCP10SE L1	CCP10SE L0	CCP9SEL 2	CCP9SEL 1	CCP9SEL 0	CCP8SEL 1	CCP8SEL 0	INTCCPS EL
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CCP10SEL<1:0> CCP10 功能引脚选择位  
 00=保留  
 01= CCP10 功能位于 P2.3  
 1x= CCP10 功能位于 P0.3  
 CCP9SEL<2:0> CCP9 功能引脚选择位  
 000= CCP9 功能位于 P7.2  
 001=保留  
 010= CCP9 功能位于 P0.5  
 011= CCP9 功能位于 P6.1  
 1xx= CCP9 功能位于 P2.4  
 CCP8SEL<1:0> CCP8 功能引脚选择位  
 00= CCP8 功能位于 P1.1  
 01= CCP8 功能位于 P6.2  
 1x= CCP8 功能位于 P2.1  
 INTCCPSEL: 外部引脚关断 PWM3/4/5 功能选择位 (配合 PxASS 使用)  
 0=保留  
 1= INT 功能位于 P0.7

2.13.1.6 复用引脚功能控制寄存器 APFCTL7

寄存器: APFCTL7: 复用引脚功能控制寄存器1(地址: 463H)

	bit7						bit0	
复位值 0000 0000	IOC7SEL	IOC6SEL	IOC5SEL	IOC4SEL	IOC3SEL	IOC2SEL	IOC1SEL	IOC0SEL
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W



- IOC7SEL: IOC7 功能引脚选择位  
0= IOC7 功能位于 P4.5  
1= IOC7 功能位于 P0.1
- IOC6SEL: IOC6 功能引脚选择位  
0= IOC6 功能位于 P9.0  
1= IOC6 功能位于 P0.2
- IOC5SEL: IOC5 功能引脚选择位  
0= IOC5 功能位于 P9.2  
1= IOC5 功能位于 P0.3
- IOC4SEL: IOC4 功能引脚选择位  
0= IOC4 功能位于 P9.5  
1= IOC4 功能位于 P0.4
- IOC3SEL: IOC3 功能引脚选择位  
0= IOC3 功能位于 P6.6  
1= IOC3 功能位于 P0.5
- IOC2SEL: IOC2 功能引脚选择位  
0= IOC2 功能位于 P6.3  
1= IOC2 功能位于 P7.5
- IOC1SEL: IOC1 功能引脚选择位  
0= IOC1 功能位于 P6.1  
1= IOC1 功能位于 P0.6
- IOC0SEL: IOC0 功能引脚选择位  
0= IOC0 功能位于 P5.3  
1= IOC0 功能位于 P0.7

### 2.13.1.7 复用引脚功能控制寄存器 APFCTL8

寄存器: APFCTL8: 复用引脚功能控制寄存器1(地址: 464H)

	bit7						bit0	
复位值 0000 0000	IOC14SEL	IOC13SEL	IOC12SEL	IOC11SEL	IOC10SEL	TICKSEL	TIGSEL1	TIGSEL0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- IOC14SEL: IOC14 功能引脚选择位  
0= IOC14 功能位于 P4.3  
1=保留
- IOC13SEL: IOC13 功能引脚选择位  
0= IOC13 功能位于 P9.1  
1= IOC13 功能位于 P5.4
- IOC12SEL: IOC12 功能引脚选择位  
0= IOC12 功能位于 P9.4  
1= IOC12 功能位于 P5.2
- IOC11SEL: IOC11 功能引脚选择位  
0= IOC11 功能位于 P6.5

	1= IOC11 功能位于 P5.0
IOC10SEL:	IOC10 功能引脚选择位
	0= IOC10 功能位于 P6.2
	1= IOC10 功能位于 P2.5
T1CKSEL:	T1CK 功能引脚选择位
	0= T1CK 功能位于 P7.1
	1=保留
T1GSEL<1:0>:	T1G 功能引脚选择位
	00= T1G 功能位于 P5.1
	01= T1G 功能位于 P0.3
	1x= T1G 功能位于 P9.2

### 2.13.1.8 复用引脚功能控制寄存器 APFCTL9

寄存器: APFCTL9: 复用引脚功能控制寄存器1(地址: 465H)

	bit7						bit0
复位值 0000 0000	SDIDA2S EL1	SDIDA2S EL0	SCKCL2S EL	SDO2SEL 1	SDO2SEL 0	SS2SEL1	SS2SEL0 CLKOUT SEL
	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SDIDA2SEL<1:0>:	SDA2/SDI2 功能引脚选择位
	00= SDA2/SDI2 功能位于 P6.4
	01= SDA2/SDI2 功能位于 P9.2
	1x= SDA2/SDI2 功能位于 P0.2
SCKCL2SEL:	SCL2/SCK2 功能引脚选择位
	0= SCL2/SCK2 功能位于 P6.3
	1= SCL2/SCK2 功能位于 P0.3
SDO2SEL<1:0>:	SDO2 功能引脚选择位
	00= SD02 功能位于 P6.5
	01= SDO2 功能位于 P9.1
	1x= SDO2 功能位于 P0.1
SS2SEL<1:0>:	SS2 功能引脚选择位
	00= SS2 功能位于 P6.2
	01= SS2 功能位于 PA.1
	10= SS2 功能位于 P1.0
	11= SS2 功能位于 P0.4
CLKOUTSEL:	CLKOUT 功能引脚选择位
	0=保留
	1= CLKOUT 功能位于 P5.3

## 2.13.1.9 复用引脚功能控制寄存器 APFCTL10

寄存器: APFCTL10: 复用引脚功能控制寄存器1(地址: 466H)

		bit7						bit0	
复位值	0000 0000	SDIDA1S EL1	SDIDA1S EL0	SCKCL1S EL1	SCKCL1S EL0	SDO1SEL	SS1SEL	PWM2OU TSEL	PWM1OU TSEL
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SDIDA1SEL<1:0>: SDA1/SDI1 功能引脚选择位

00= SDA1/SDI1 功能位于 P2.3

01= SDA1/SDI1 功能位于 P0.1

1x= SDA1/SDI1 功能位于 P7.1

SCKCL1SEL<1:0>: SCL1/SCK1 功能引脚选择位

00= SCL1/SCK1 功能位于 P1.0

01= SCL1/SCK1 功能位于 P0.2

1x= SCL1/SCK1 功能位于 P7.2

SDO1SEL: SDO1 输出功能引脚选择位

0= SD01 功能位于 P2.4

1=保留

SS1SEL: SS1 功能引脚选择位

0=保留

1= SS1 功能位于 P1.7

PWM2OUTSEL: PWM2OUT 输出功能引脚选择位

0= PWM2OUT 功能位于 P3.0

1= PWM2OUT 功能位于 P2.5

PWM1OUTSEL: PWM1OUT 输出功能引脚选择位

0= PWM1OUT 功能位于 P4.5

1= PWM1OUT 功能位于 P9.5

## 2.13.1.10 复用引脚功能控制寄存器 APFCTL11

寄存器: APFCTL11: 复用引脚功能控制寄存器1(地址: 467H)

		bit7						bit0	
复位值	0000 0000	保留	保留	ADVRINS EL1	ADVRINS EL0	保留	保留	保留	保留
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ADVRINSEL<1:0>: ADVRIN 功能引脚选择为位

00= ADVRIN 功能位于 P0.2

01= ADVRIN 功能位于 P1.2

1x= ADVRIN 功能位于 PB.0

## 2.13.1.11 复用引脚功能控制寄存器 APFCTL12

寄存器: APFCTL12: 复用引脚功能控制寄存器1(地址: 468H)

	bit7				bit0			
复位值	C3OUTSE	C2OUTSE	C2OUTSE	C1OUTSE	CK3TX3S	DT3RX3S	CK4TX4S	DT4RX4S
0000 0000	L	L1	L0	L	EL	EL	EL	EL
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- C3OUTSEL: C3OUT 功能引脚选择位  
0=C3OUT 功能位于 P4.3  
1=C3OUT 功能位于 P3.7
- C2OUTSEL<1:0>: C2OUT 功能引脚选择位  
00=C2OUT 功能位于 P0.4  
01=C2OUT 功能位于 PA.1  
1x=C2OUT 功能位于 P1.0
- C1OUTSEL: C1OUT 功能引脚选择位  
0=C1OUT 功能位于 P0.5  
1=C1OUT 功能位于 P1.1
- CK3TX3SEL: CK3/TX3 功能引脚选择位  
0= CK3/TX3 功能位于 P1.3  
1= CK3/TX3 功能位于 P7.5
- DT3RX3SEL: DT3/RX3 功能引脚选择位  
0= DT3/RX3 功能位于 P1.2  
1= DT3/RX3 功能位于 P0.5
- CK4TX4SEL: CK4/TX4 功能引脚选择位  
0= CK4/TX4 功能位于 PA.1  
1= CK4/TX4 功能位于 P1.5
- DT4RX4SEL: DT4/RX4 功能引脚选择位  
0= DT4/RX4 功能位于 PA.0  
1= DT4/RX4 功能位于 P1.4

### 3 振荡器

KF8A100 有一个带故障保护时钟检测的时钟选择模块，该模块具有多种时钟源可选，使得应用更广泛，并且可以最大程度的提高性能以及降低功耗。图 3.1 为振荡器模块的原理框图。

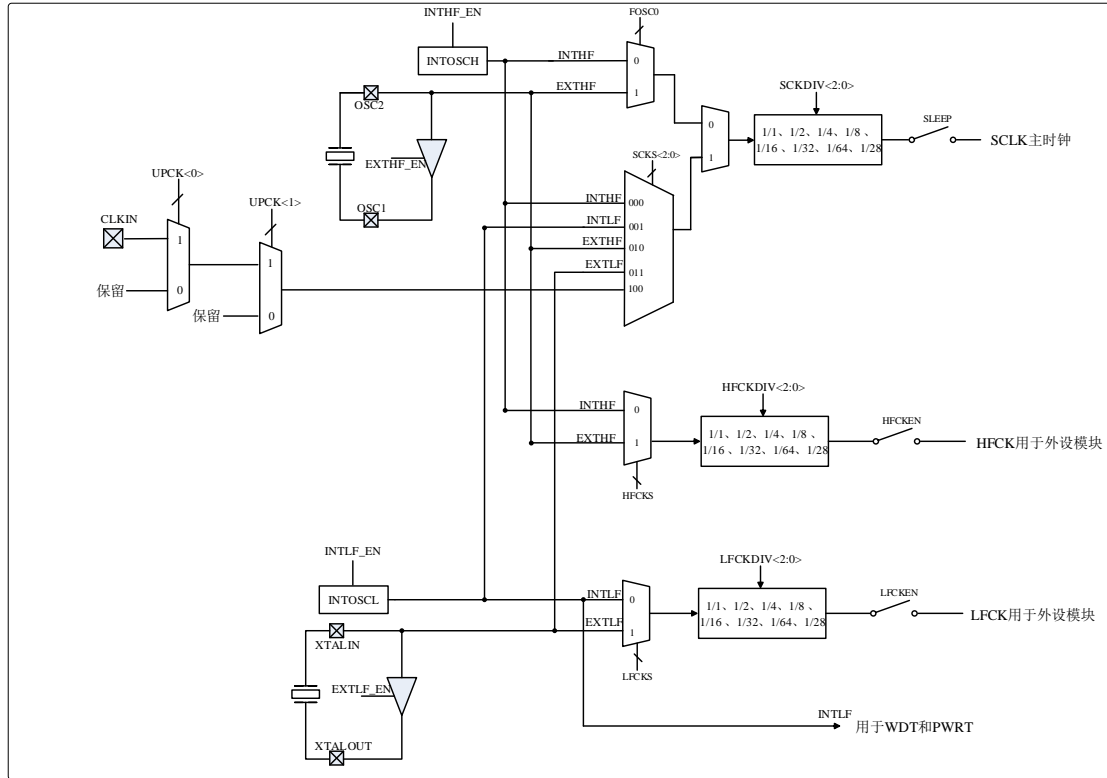


图 3.1 时钟模块原理框图

#### 3.1 概述

KF8A100 单片机包含四个可选时钟源（外部时钟 EXTHF 和 EXTLF 公用 2 个引脚）：

1. 内部高频时钟 INTHF：以内部高频振荡器为时钟源；
2. 内部低频时钟 INTLF：以内部低频振荡器为时钟源；
3. 外部高频时钟 EXTHF：可以用标准晶振、陶瓷谐振器或外接 20M~125KHZ 的高频振荡器作为时钟源；
4. 外部低频时钟 EXTLF：使用外接 32.768KHz 的钟表晶振作为时钟源。

#### 特点：

1. 配备三种主时钟：
  - 系统主时钟 SCLK：
 

SCLK 通过 SCKS<2:0>位进行选择相应的时钟源作为系统时钟 SCLK。SCLK 用于 CPU 和系统外设的时钟。
  - 低频外设时钟 LFCLK：
 

LFCLK 通过 LFCKS 位进行选择相应的低频时钟源作为 LFCLK 时钟。LFCLK 用于各外设模块的时钟。

- 高频外设时钟 HFCLK:  
HFCLK 通过 HFCKS 位进行选择相应的高频时钟源作为 HFCLK 时钟。HFCLK 用于各外设模块的时钟。
2. 系统时钟可选多个时钟源:
- 内部高频时钟 INTHF
  - 内部低频时钟 INTLF
  - 外部高频时钟 EXTHF
  - 外部低频时钟 EXTLF
- 注: WDT 定时器和上电延时定时器 (PWRT) 的时钟为 INTLF。

### 3.2 振荡器相关寄存器

表 3.1 与中断相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
69H	SCLKCTL	SCLKOE	SCS	SCKDIV2	SCKDIV1	SCKDIV0	SCKS2	SCKS1	SCKS0
6AH	HFCKCTL	HFCKEN	UPCK1	UPCK0	-	HFCKDI V2	HFCKDI V1	HFCKDI V0	HFCKS -
6BH	LFCKCTL	LFCKEN	-	-	-	LFCKDI V2	LFCKDI V1	LFCKDI V0	LFCKS
54H	SCLKSTU	OSCFAIL IF	CLKS	OSTS	HTS	LTS	OSTIF	-	-
516H	保留	-	-	-	-	-	-	-	-
131H	OSCCAL0	内部高频晶振校准寄存器 0							
37H	OSCCAL1	内部高频晶振校准寄存器 1							
30H	OSCCAL2	内部高频晶振校准寄存器 2							
132H	OSCCAL3	内部高频晶振校准寄存器 3							
365H	LPRCCA L	内部低频晶振校准寄存器							
366H	LPRCCTL	-	-	-	-	-	LPRC2	LPRC1	LPRC0

#### 3.2.1 系统时钟控制寄存器 (SCLKCTL)

**寄存器:** SCLKCTL 主时钟选择控制寄存器(地址:69H)

复位值	bit7						bit0	
0001 1000	SCLKOE	SCS	SCKDIV2	SCKDIV1	SCKDIV0	SCKS2	SCKS1	SCKS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

注: R=可读 W=可写 U=未用 x=不定 后面与此相同不再复述

- SCLKOE:** 系统时钟输出使能位  
1 = 允许 SCLK 时钟从 CLKOUT 引脚输出  
0 = 禁止 SCLK 时钟从 CLKOUT 引脚输出
- SCS:** 系统时钟选择位  
1 = 由 SCKS<2:0>选择系统时钟

- 0 = 由配置位 FOSC0 选择系统时钟
- SCKDIV<2:0>: 主时钟 SCLK 分频选择位
- 111 = 1/1 分频
- 110 = 1/2 分频
- 101 = 1/4 分频
- 100 = 1/8 分频
- 011 = 1/16 分频(初始值)
- 010 = 1/32 分频
- 001 = 1/64 分频
- 000 = 1/128 分频
- SCKS<2:0>: SCLK 主时钟选择位
- 1xx = 选择 UPCR 时钟作为主时钟
- 011 = 选择 EXTLF 作为主时钟
- 010 = 选择 EXTHF 作为主时钟
- 001 = 选择 INTLF 作为主时钟
- 000 = 选择 INTHF 作为主时钟

注：系统时钟不能低于4MHz。

### 3.2.2 主时钟状态寄存器（SCLKSTU）

寄存器： SCLKSTU:主时钟状态寄存器(地址:54H)

复位值 0000 0---	bit7						bit0	
	OSCFAILIF	CLKS	OSTS	HTS	LTS	OSTIF	-	-
	R/W	R	R	R	R	R	U	U

- OSCFAILIF: 时钟故障标志位
- 1 = 外部高频时钟源出现故障
- 0 = 外部高频时钟源未出现故障
- CLKS: 时钟选择状态位
- 0 = 系统时钟源由 SCKCTL 寄存器的 SCKS<1:0>决定
- 1 = 系统时钟源由 FOSC0 决定
- OSTS: 内外时钟状态位
- 0 = 时钟源为内部时钟
- 1 = 时钟源为外部时钟
- HTS: INTHF 状态位
- 0 = INTHF 不稳定
- 1 = INTHF 稳定
- LTS: INTLF 状态位
- 0 = INTLF 不稳定
- 1 = INTLF 稳定
- OSTIF: OST 计时器溢出标志位



1 = OST 计数器溢出

0 = OST 计数器未溢出

注：任何时钟源切换操作都会清零此位

### 3.2.3 高频外设时钟选择控制寄存器（HFCKCTL）

**寄存器：** HFCKCTL 高频时钟选择控制寄存器(地址:6AH)

		bit7				bit0			
复位值		HFCKEN	UPCK1	UPCK0	-	HFCKDIV2	HFCKDIV1	HFCKDIV0	HFCKS0
0-00 000-		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**HFCKEN:** 高频外设时钟允许位

1 = HFCLK 时钟信号允许

0 = HFCLK 时钟信号禁止

**UPCK<1:0>:** UPCK 时钟源选择位

11 = 选择 CLKIN 引脚作为 UPCK

10 = 保留

01 = 保留

00 = 保留

注：硬件禁止在 SCKS<2:0>=100 时改变 UPCK 的值。

**HFCKDIV<2:0>:** 高频外设时钟源分频选择位

111 = 1/1 分频

110 = 1/2 分频

101 = 1/4 分频

100 = 1/8 分频

011 = 1/16 分频

010 = 1/32 分频

001 = 1/64 分频

000 = 1/128 分频

**HFCKS<0>:** 高频外设时钟源选择位

1 = 选择 EXTHF 为 HFCLK 时钟信号

0 = 选择 INTHF 为 HFCLK 时钟信号

### 3.2.4 低频外设时钟选择控制寄存器（LFCKCTL）

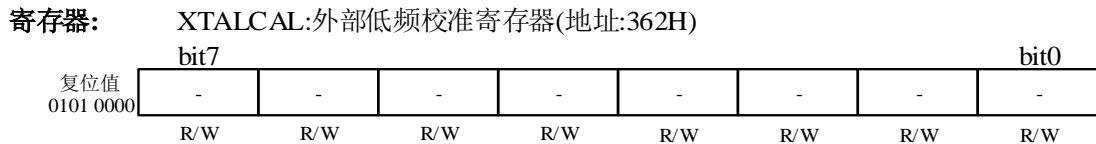
**寄存器：** LFCKCTL 低频时钟选择控制寄存器(地址:6BH)

		bit7				bit0			
复位值		LFCKEN	-	-	-	LFCKDIV2	LFCKDIV1	LFCKDIV0	LFCKS
0---0000		R/W	U	U	U	R/W	R/W	R/W	R/W

**LFCKEN:** 低频外设时钟允许位

- 1=LFCLK 时钟信号允许
- 0=LFCLK 时钟信号禁止
- LKCKDIV<2:0>: 低频外设时钟源分频选择位
  - 111 = 1/1 分频
  - 110 = 1/2 分频
  - 101 = 1/4 分频
  - 100 = 1/8 分频
  - 011 = 1/16 分频
  - 010 = 1/32 分频
  - 001 = 1/64 分频
  - 000 = 1/128 分频
- LFCKS: 低频外设时钟源选择位
  - 0 = 选择内部 INTLF 为 LFCLK 时钟信号
  - 1 = 选择外接低频振荡器产生的 EXTLF 为 LFCLK 时钟信号

### 3.2.5 校准寄存器



- BIT<5:3>: 外部低频的功耗档位选择位
- BIT<5>: 62nA 偏置电流档位
- BIT<4>: 45nA 偏置电流档位
- BIT<3>: 27nA 偏置电流档位
  - 000 = 偏置电流不开, XTAL 不工作
  - ...
  - 111 = 偏置电流 134nA 档位

注: XTALCAL<5:3>档位应选择最大, 即“111”。偏置电流越大, 外部低频时钟越容易起振。

### 3.3 振荡器工作模式

振荡周期又叫时钟周期, 是振荡器振荡频率的倒数。系统时钟是由振荡器时钟分频而来。本芯片中一个机器周期等于四个系统周期(SCLK), 如图 3.2 所示。本芯片除执行部分跳转指令需要两个机器周期外, 其余指令仅需要一个机器周期。

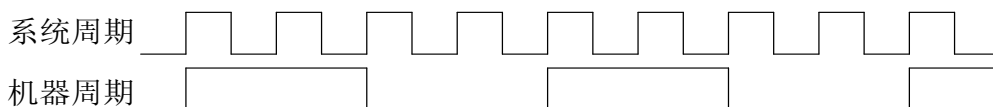


图 3.2 机器周期

### 3.3.1 振荡器详细描述

#### 3.3.1.1 内部高频振荡器

KF8A100 单片机可以通过系统内部高频振荡器提供内部高频时钟源 INTHF，时钟频率为 16Mhz，精度为±1%。

OSCCAL0、OSCCAL1、OSCCAL2 和 OSCCAL3 为 INTHF 振荡器时钟校准寄存器，用来存放系统时钟校准值。用户在编程时，需要在程序初始化部分将存放在程序空间 7FFFH 和 7FFE H 的晶振校准值读出来存放到 OSCCAL0、OSCCAL1、OSCCAL2 和 OSCCAL3，参考例子 1.1，否则会导致系统时钟频率不准。

##### 例1.1 读内部高频晶振校准值

```

MOV P #0X70 ;切换到PAGE7区
CALL 0X7FF
MOV OSCCAL0, R0
CALL 0X7FE
MOV OSCCAL1, R0
CALL 0X7FD
MOV OSCCAL2, R0
CALL 0X7FC
MOV OSCCAL3, R0
MOV P #0X00 ;切换回PAGE0区

```

#### 3.3.1.2 内部低频振荡器

KF8A100 单片机可以通过系统内部低频振荡器提供内部低频时钟源 INTLF。

内部低频时钟 INTLF 为系统内部低频振荡器 INTOSCL 提供，其振荡器频率位 32KHz，它不仅可以为 SCLK 时钟提供时钟源，而且还可以独立作为 LPCLK 时钟信号供外设模块使用。也可以作为内部上电延时定时器和看门狗 WDT 定时器的时钟。

LPRCCAL 寄存器为内部低频振荡器校准寄存器，用来存放校准值。用户在编程时，需要在程序初始化部分，将存放在程序空间 7FF9H 的晶振校准值存放到 LPRCCAL 中。程序示例请参考下例。

##### 例： 读内部低频晶振校准值

```

MOV P #0X70 ;切换到PAGE7区
CALL 0X7F8
MOV LPRCCAL , R0
MOV P #0X00 ;切换回PAGE0区

```

### 3.3.1.3 外部高频振荡器

如图 3.3 所示，引脚 OSC2 和引脚 OSC1 可以接外部标准晶体、陶瓷谐振器或外接 16MHz~125KHz 的时钟为时钟源。EXTOSCH 产生的外部高频时钟 EXTHF 可提供主系统时钟 SCLK。外部晶振所接跨阻推荐阻值为 750KΩ~1MΩ。

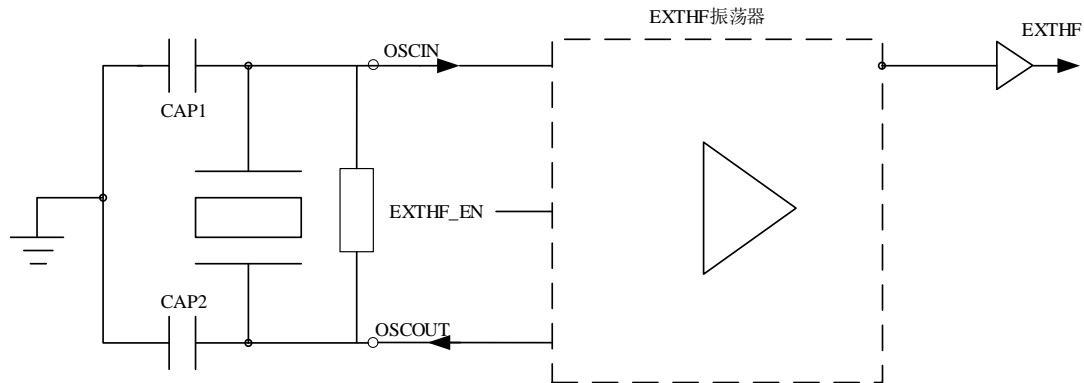


图 3.3 外部高频振荡器原理图

### 3.3.1.4 外部低频振荡器

如图 3.4 所示，引脚 XTALIN 和引脚 XTALOUT 接外部钟表石英晶体，主要是以 32.768KHz 的晶体提供时钟源。EXTLF 时钟源提供主系统时钟 SCLK。

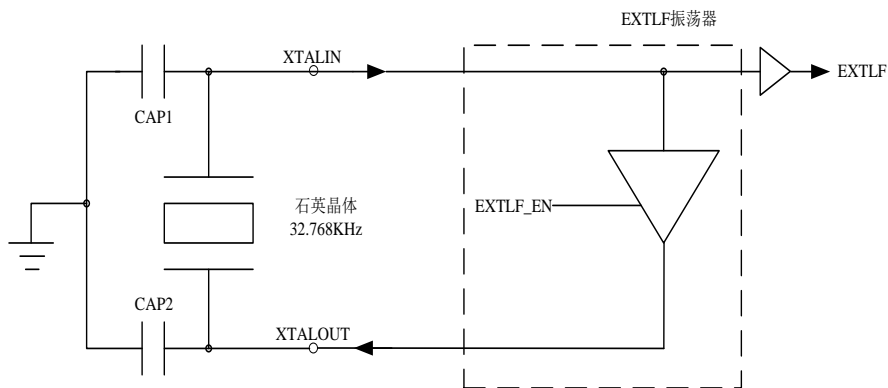


图 3.4 外部低频振荡器原理图

## 3.3.2 系统主时钟 SCLK

KF8A100 的系统主时钟 SCLK 可以由内部高频 INTHF、外接低频 EXTLF、外接高频 EXTHF 和内部低频 INTLF 提供。

SCLKCTL 寄存器的 SCS 位用于选择系统时钟是由 FUSE 的 FOSC0 位或 SCKS<1:0> (SCKCTL<1:0>) 确定。当 SCS=0 时，系统时钟由 FUSE 中的 FOSC0 确定；当 SCS=1 时，系统时钟由 SCKS<1:0> 确定。复位后，SCS 位为 0。

注：双速启动模式或故障保护模式引起的任何自动时钟切换均不会更新 SCS 位，用户可通过 OSTST 时钟状态位及 CLKS 内外时钟选择状态位确定当前的系统时钟源。

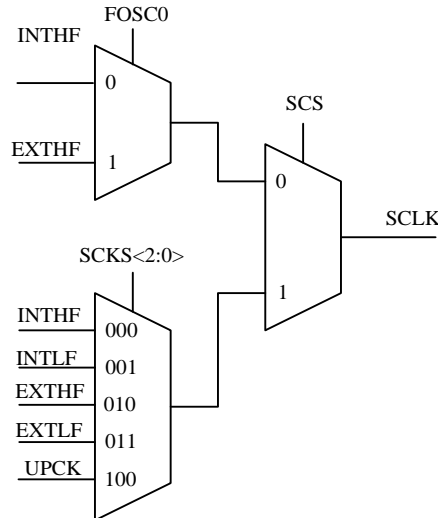


图 3.5 KF8A100 时钟模式选择

### 3.3.2.1 系统时钟的切换与同步

单片机上电或者复位后，配置位 FOSC0 选择系统是内部高频时钟 INTHF 或者外部高频时钟 EXTHF，如果打开双速启动模式，当 FOSC0=1 选择 EXTHF 时，单片机将以 INTHF 作为系统时钟先运行，同时 EXTHF 进入 OST 计数，溢出后，系统时钟切换至 EXTHF 继续工作。

单片机在上电或者复位完成后，用户可以通过 SCS 位置 1 来将系统时钟切换至寄存器位 SCKS<2:0>控制。

当系统时钟由外部时钟源（EXTHF、EXTLF）切换至内部时钟源（INTHF、INTLF）时，系统时钟将在 SCS 位和 SCKS<1:0>位配置后立即对时钟进行切换。

当系统时钟切换至外部时钟源（EXTHF、EXTLF）时，振荡器起振定时器 OST 将启动，并以配置的外部时钟源（EXTHF、EXTLF）为计数时钟开始计数，OST 计数的时间内，系统仍以原来的时钟源作为系统时钟，直到 OST 计数器达到 1024 次计数，系统时钟源切换至配置的外部时钟源（EXTHF、EXTLF）。

SCS 位和 SCKS 位的相关配置(从内部时钟源切换到外部时钟源)将触发 OST 计数器。任何时钟源的切换请在 SCS 位和 SCKS 位上完成。

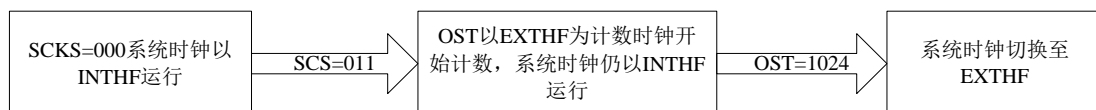


图 3.6 时钟切换流程图

当系统时钟在一个时钟源切换到另一个时钟源时，切换必须同步以避免发生时间竞争。详细描述见[时钟信号同步](#)章节。

### 3.3.2.2 双速时钟启动模式

双速时钟启动模式通过外部振荡器起振到代码执行之间的延时达到最小而进一步降低

功耗。

当系统时钟被配置为外部时钟模式，则使能振荡器起振定时器(OST)。OST 将暂停程序执行直到计数完 1024 次振荡为止。双速启动模式将在 OST 计数时使用内部高频时钟作为系统时钟源，最大限度地缩短代码执行的延时。当 OST 计数达到 1024 且 OST 位置 1 时，系统时钟将切换到外部振荡器充当时钟源。

#### 双速启动模式配置：

1. IESO=1；内外时钟切换使能位（使能双速启动模式）
2. 时钟配置为外部时钟源方法：
  - SCS=0， FOSC0=1；
  - SCS=1， SCKS1=1；

#### 发生下列事件后进入双速启动模式：

1. 上电复位（POR）后；
2. 上电延时定时器（PWRT）超时时；
3. 从休眠状态唤醒后。

#### 双速启动过程：

1. 上电复位或从休眠状态唤醒；
2. 内部振荡器作为时钟源开始执行指令；
3. 使能 OST 计数 1024 个时钟周期；
4. OST 超时，等待内部时钟下降沿；
5. OSTIF 置 1；
6. 系统时钟保持低电平直到新的时钟的下一个下降沿；
7. 系统时钟切换到外部时钟源。

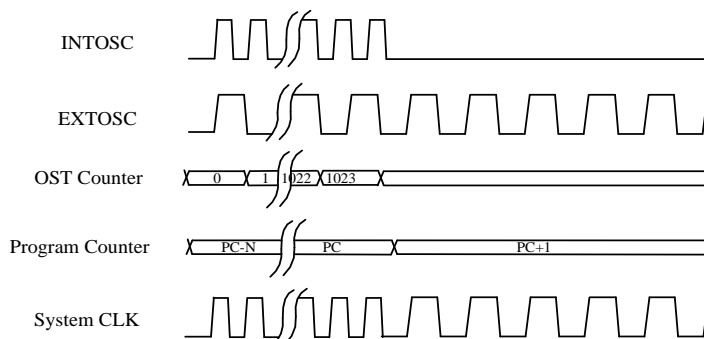


图 3.7 双速启动时序

### 3.3.3 高频外设时钟 HFCLK

高频外设时钟 HFCLK 可以给外设提供高频时钟信号，它可以通过 HFCKS 位选择内部高频振荡器 INTHF 或外部高频振荡器 EXTHF 作为 HFCLK 的时源。通过 HKCKDIV<3:0> 位选择分频，可以进行 1/1~1/128 的分频。HFCKEN (HFCKCTL<6>) 位实现对高频外设时钟 HFCLK 时钟信号的允许和禁止。

### 3.3.4 低频外设时钟 LFCLK

低频外设时钟 LFCLK 可以为外设提供低频时钟信号，它可以通过 LFCKS (LFCKCTL

<0>位选择内部低频 INTLF 或者外部低频 EXTLF 作为 LFCLK 的时源。通过 LKCKDIV<2:0> 位选择分频，可以进行 1~1/128 的分频。

### 3.4 时钟信号同步

SCLK、HFCLK 和 LFCLK 时钟都可以由不同时钟源提供。当一个时钟源切换到另一个时钟源时，切换必须同步以避免发生时间竞争。当选择一个新的时钟源，会发生以下过程：

- (1) 写 SCKCTL 寄存器改变时钟源；
- (2) 时钟切换电路等待当前时钟的下降沿；
- (3) 时钟 CLK 保持为低电平，时钟切换电路等待新的时钟的上升沿；
- (4) CLK 与新的时钟连接，完成时钟切换。

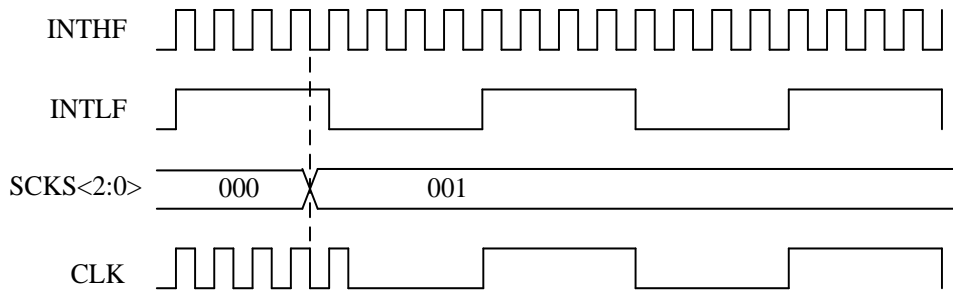


图 3.8 时钟信号同步时序图

### 3.5 休眠模式

在执行指令 IDLE 后，单片机会进入休眠模式。此时 CPU 时钟停止，外设根据所使能的情况选择不同的 LFCLK 时钟，WDT 使能时 INTLF 振荡器使能。

休眠时的电流主要由各自模块的电流组成。

### 3.6 外部时钟故障检测

故障保护时钟监视（FSCM）能使器件在振荡器发生故障时继续运行，其可以检测出振荡器起振定时器（OST）延时结束后的任何时刻发生的振荡器故障。

将配置位的 FSCM 位置 1 使能 FSCM，适用于外部振荡器模式。

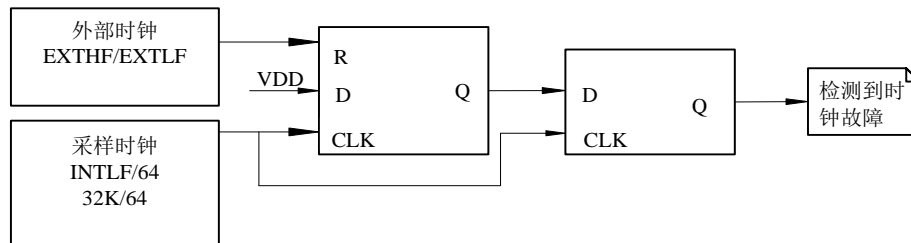


图 3.9 时钟故障检测原理图

### 3.6.1 时钟故障保护检测

FSCM 模块通过比较外部振荡器和采样时钟检测使用的外部振荡器。

FSCM 模块在采样时钟下降沿将第一个寄存器置 1，在外部时钟下降沿将该寄存器复位为 0，由于外部时钟频率远大于采样时钟频率，所以当第一个寄存器刚被采样时钟置 1 不久就被外部时钟复位，经过第二个寄存器(CLK 为采样时钟)后的 Q 值将保持为 0；当发生外部时钟故障时，第一个寄存器由于外部时钟故障而失去复位能力，当采样时钟下降沿到达置 1 后，将一直保持输出为 1，检测到故障。

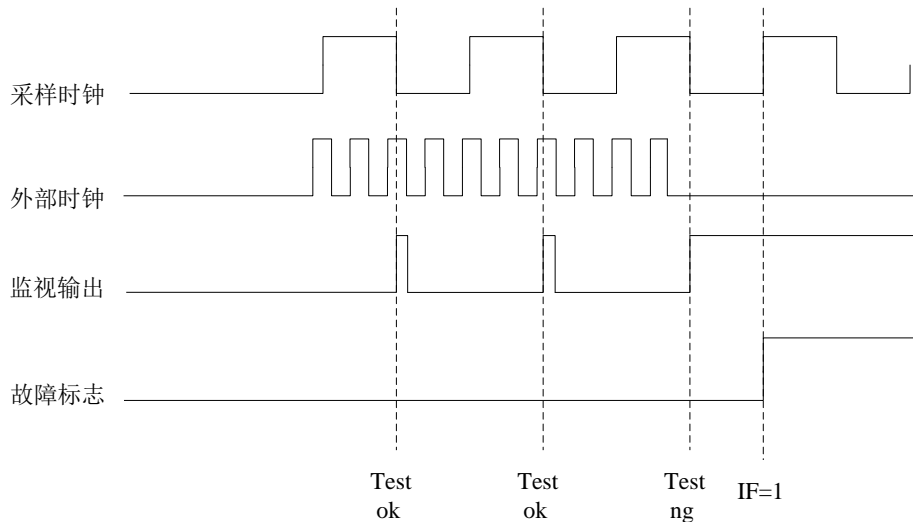


图 3.10 时钟故障检测时序

注：系统时钟频率实际比采样时钟大很多，图示为方便分析起见将频率差异减小。

### 3.6.2 时钟故障保护处理

检测到时钟故障后，系统时钟将自动切换至内部高频时钟源继续工作，分频值由 SCLKCTL 寄存器的 IRCS<2:0>设置，直到器件固件成功重启外部振荡器并使时钟重新切换到外部振荡器为止。

在切换至内部高频时钟源后，时钟故障标志位置 1，如果打开时钟故障中断使能位，程序将进入中断行。



### 4 存储器

如图 4.1 所示, KF8A100 中存储器主要由程序存储器(ROM)和数据存储器(RAM)组成, 程序存储器和数据存储器地址空间相互独立。其中程序存储器为 32K×16 位的 FLASH 存储器; 数据存储器由特殊功能寄存器和通用寄存器组成, 特殊功能寄存器空间为 672×8 位, 通用数据寄存器空间为 (4K+16)×8 位。另外 KF8A100 中还有一些其它存储器, 包括: DATA EEPROM、寄存器组 R0~R7、16 级硬件堆栈、ID 地址单元等。

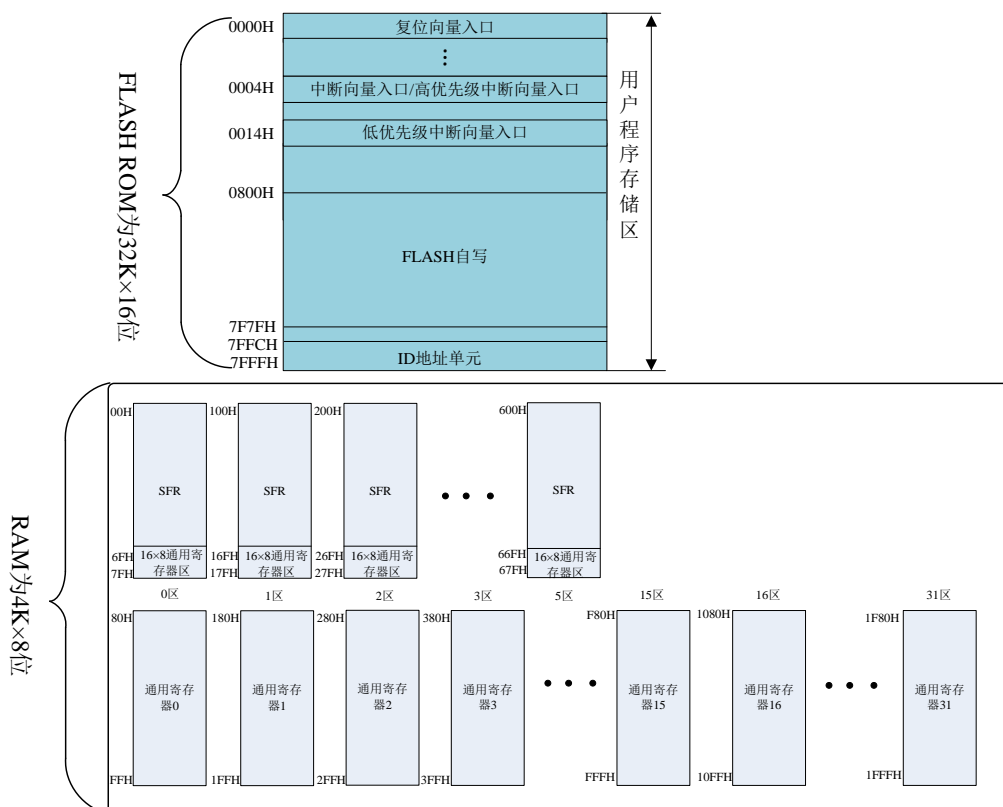


图 4.1 存储器组织图

## 4.1 程序存储器(ROM)区

KF8A100 有一个 15 位的程序计数器，最大可寻址 32K×16 位的程序存储空间，地址为 0000H~7FFFH，复位向量入口地址为 0000H，中断向量有两级入口地址，高为 0004H，低为 0014H。

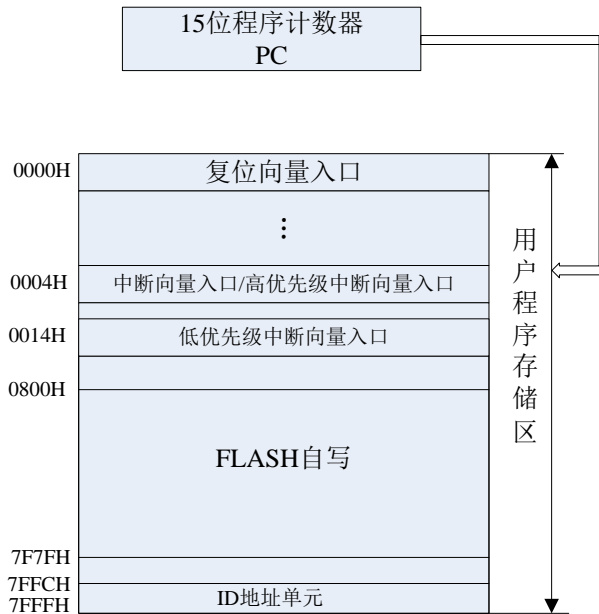


图 4.2 KF8A100 程序存储器映射

### 4.1.1 程序计数器(PC)

KF8A100 的程序计数器(PC)为 15 位宽。程序计数器(PC)的低 8 位(PC<7:0>)来自特殊功能寄存器 PCL，高 7 位(PC<14:8>)来自 PCH 寄存器，PC 不能直接读写，PCH/PCL 寄存器可以读写。当发生复位时，PC 将被清零。在有任何未屏蔽中断发生后 PC 值将指向 0004H 或 0014H 地址。

在用户的程序中，每当执行一条汇编指令 PC 值会自动加 1，指向下一条要执行的指令。当有子程序调用或响应中断时，CPU 会将 PC+1 后的值压入堆栈进行保存，然后将子程序或中断入口地址送到 PC 中，CPU 根据 PC 的值跳转到对应的地址执行命令。

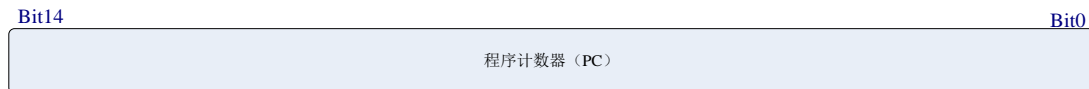


图 4.3 程序计数器 (PC)

#### 4.1.1.1 PCL/PCH 寄存器的使用

当需要改变程序计数器到固定地址（包括跳页）时，可以通过写 PCH 寄存器确定高 7 位地址（写 PCH 寄存器不会改变 PC 的值），当写低 8 位地址数据到 PCL 寄存器时，程序计数器的 15 位地址数据将更新，变为 PCH 寄存器和 PCL 寄存器的数据，如图 4.4 所示。

对 PCL/PCH 寄存器的写操作可以通过 MOV 指令来执行，对 PCH 寄存器的写操作同样可以通过专用写指令 MOVP 来实现（只需 1 条指令即可完成 PCH 的赋值）。

例如当程序计数器从 0000H 开始执行跳转程序到 1F55H 地址，过程如下：

PC	指令	备注
0000	MOVP #0X1F	将 1FH 写入 PCH 寄存器
0001	MOV R0, #0X55	将 55H 赋给 R0
0002	MOV PCL, R0	将 R0 内数据写入 PCL, PC 内容变为 PCH/PCL 寄存器的值
1F55	...	...

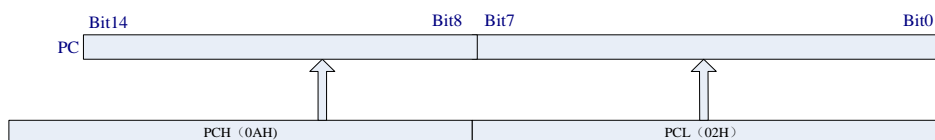


图 4.4 PCL/PCH 寄存器的使用

#### 4.1.1.2 执行 JMP、CALL 指令时的情况

KF8A100 系列单片机的 JMP、CALL 指令编码如下：

JMP #data12	1100_kkkk_kkkk_kkkk
CALL #data12	1101_kkkk_kkkk_kkkk

在执行 JMP 或者 CALL 指令时，程序计数器（PC）的值将变为 PCH\_BIT6、PCH\_BIT5 和 PCH\_BIT4 以及指令所带立即数（#data12），如图 4.5 所示。

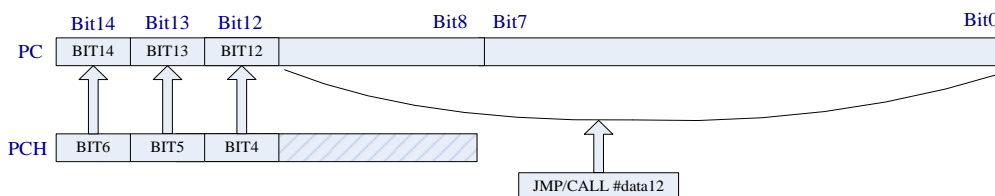


图 4.5 执行 JMP/CALL 指令时 PC 的变化图

执行 JMP 指令时将更新 PC；而执行 CALL 指令在更新 PC 的同时，将 CALL 指令的下一条地址入栈，栈地址加 1；在执行 RETURN 指令（IRET、RRET、CRET）时，将之前入栈的地址数据出栈并更新到 PC，栈地址减 1，PCH 寄存器不受出栈入栈的影响。

#### 4.1.2 堆栈

KF8A100 系列单片机具有 16 级的硬件堆栈。堆栈空间单独编址，不占用任何程序存储区和数据存储区的空间，堆栈指针不可读写。当执行 CALL 指令或者中断导致程序跳转时，PC 值将被压入（PUSH）堆栈。在执行返回指令（IRET、RRET、CRET）时，堆栈中的断点地址将从堆栈中弹出（POP）到 PC 中。PCH 不受 PUSH 或者 POP 操作的影响。

因为是 16 级的硬件堆栈，所以堆栈压栈第 17 次的数值将覆盖第 1 次压栈的所保存的值，堆栈压栈第 18 次的数值将覆盖第 2 次压栈的所保存的值，以此类推。

## 4.2 数据存储器(RAM)区

如图 4.6 所示, KF8A100 中的数据存储器由 37 个区组成, 每个区的空间都是 128 字节, 其中 5 个区用作特殊功能寄存器区(SFR)使用; 另外 32 个存储器区为通用寄存器区, 由用户支配。SFR 地址空间为 00H~6FH、100H~16FH、200H~26FH、300H~36FH 和 400H~46FH; 而 70H~7FH 有 16 个字节为 SRAM 共用区, 即当用户访问其他 BANK 区 70H~7FH 的存储单元时, 均是对 BANK 0 区的 70H~7FH 操作。

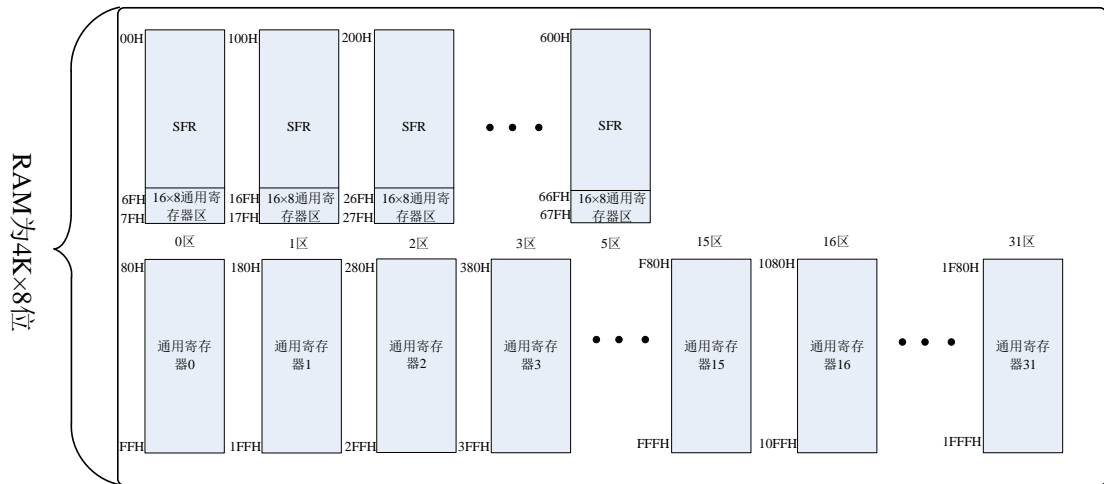


图 4.6 数据存储器地址映射图

### 4.2.1 通用寄存器区

如图 4.6 所示, 通用寄存器的空间为 4096 字节, 0 区至 31 区通过 BANK 寄存器中的 PR4~PR0 位进行选择, 如表 4.1 所示。

**寄存器 4.1:** BANK: 通用寄存器选区寄存器

复位值 ---0 0000	bit7 -	bit6 -	bit5 -	bit4 RP4	bit3 PR3	bit2 PR2	bit1 PR1	bit0 PR0
	U	U	U	R/W	R/W	R/W	R/W	R/W

表 4.1 通用寄存区地址

PR<4:0>	通用寄存器区	地址	PR<4:0>	通用寄存器区	地址
00000	通用寄存器 0 区	80H~17FH	10000	通用寄存器 16 区	1080H~117FH
00001	通用寄存器 1 区	180H~27FH	10001	通用寄存器 17 区	1180H~127FH
00010	通用寄存器 2 区	280H~37FH	10010	通用寄存器 18 区	1280H~137FH
00011	通用寄存器 3 区	380H~47FH	10011	通用寄存器 19 区	1380H~147FH
00100	通用寄存器 4 区	480H~57FH	10100	通用寄存器 20 区	1480H~157FH
00101	通用寄存器 5 区	580H~67FH	10101	通用寄存器 21 区	1580H~167FH
00110	通用寄存器 6 区	680H~77FH	10110	通用寄存器 22 区	1680H~177FH
00111	通用寄存器 7 区	780H~87FH	10111	通用寄存器 23 区	1780H~187FH
01000	通用寄存器 8 区	880H~97FH	11000	通用寄存器 24 区	1880H~197FH
01001	通用寄存器 9 区	980H~107FH	11001	通用寄存器 25 区	1980H~207FH

01010	通用寄存器 10 区	A80H~AFFH	11010	通用寄存器 26 区	1A80H~1AFFH
01011	通用寄存器 11 区	B80H~BFFH	11011	通用寄存器 27 区	1B80H~1BFFH
01100	通用寄存器 12 区	C80H~CFFH	11100	通用寄存器 28 区	1C80H~1CFFH
01101	通用寄存器 13 区	D80H~DFFH	11101	通用寄存器 29 区	1D80H~1DFFH
01110	通用寄存器 14 区	E80H~EFFH	11110	通用寄存器 30 区	1E80H~1EFFH
01111	通用寄存器 15 区	F80H~FFFH	11111	通用寄存器 31 区	1F80H~1FFFH

切换通用寄存区的指令如下例所示：

**例：** 切换BANK寄存器存储区

```

MOV B #0X01 ;切换到存储区1区
MOV B #0X11 ;切换到存储区17区
MOV B #0X0D ;切换到存储区13区

```

注：切换存储区同样可以对 BANK 寄存器进行 MOV/SET/CLR 操作来实现；MOVB 指令可在 1 个指令周期内完成存储区切换，MOV 指令须 2 个指令周期，SET/CLR 指令只能对位操作。

#### 4.2.2 特殊功能寄存器(SFR)区

KF8A100 内部的 I/O 口控制、定时/计数器、PWM、运放、中断等各种控制寄存器和状态寄存器都称为特殊功能寄存器。附录 1 列出 SFR 的地址映射及复位初始值等。

**状态字寄存器(PSW)：**如下寄存器所示，PSW 的低三位是算术运算标志位，在进行加、减等运算时对它们产生影响(具体请参考汇编指令部分)。 $\overline{T0}$  和  $\overline{PD}$  是复位状态位，当单片机有复位或看门狗超时、执行休眠等指令时，会对这两位产生影响。

**寄存器：** PSW: 状态字寄存器(地址: 03H)

bit7								bit0
-	-	-	$\overline{T0}$	$\overline{PD}$	Z	DC	CY	
U	U	U	R/W	R/W	R/W	R/W	R/W	

$\overline{T0}$  : 超时标志位

1 = 在上电复位、CWDT 指令或 IDLE 指令执行之后  
0 = WDT 超时被清 0

$\overline{PD}$  : 上电复位标志位

1 = 上电复位或执行 CWDT 指令后  
0 = 执行 IDLE 指令后被清 0

Z: 零状态标志位

1 = 算术运算或者逻辑运算的运行结果为 0  
0 = 算术运算或者逻辑运算的运行结果不为 0

DC: 辅助进/借位标志位

1 = 执行结果的低 4 位向高 4 位有进位(加指令)或没有借位(减指令)  
0 = 执行结果的低 4 位向高 4 位没有进位(加指令)或有借位(减指令)

CY: 进位/借位标志位

1 = 执行结果(8 位)向高位有进位时(加指令)或没有借位(减指令)

0 = 执行结果(8位)向高位无进位时(加指令)或有借位(减指令)

注：对于借位的情况，当指令执行后，低四位(或高四位)向高位有借位时，DC(或CY)标志为0，当没有借位时其值为1。关于对标志位是否产生影响的指令请参考“汇编指令集”部分。

### 4.3 FLASH 自写

KF8A100 在程序存储区开辟了可自定义的自写区域，只有当目标地址位于未受写保护的存储器段内（由配置位的 WRT<1:0>位定义）时，才能对 Flash 进行写操作。该区域在正常工作期间是可读写的，它并没有直接映射到寄存器空间，而是通过特殊功能寄存器间接寻址。有 6 个特殊功能寄存器用于访问该区域。

表 4.2 与 FLASH 相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
3AH	NVMADDRH	-	地址指针高 7 位						
3BH	NVMADDRL	地址指针低 8 位							
3CH	NVMCTL0	控制寄存器 0							
3DH	NVMCTL1	控制寄存器 1							
39H	NVMDATAL	数据寄存器低 8 位							
38H	NVMDATAH	数据寄存器高 8 位							

如图 4.7 Flash 自写区域地址映射图所示，写 FLASH 时，FLASH 中所有内存单元以连续的 64 个地址为一个数据块，2 个数据块为一页。

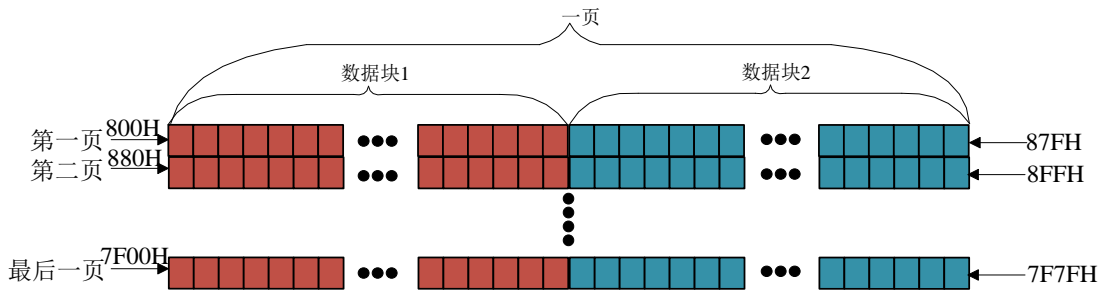


图 4.7 Flash 自写区域地址映射图

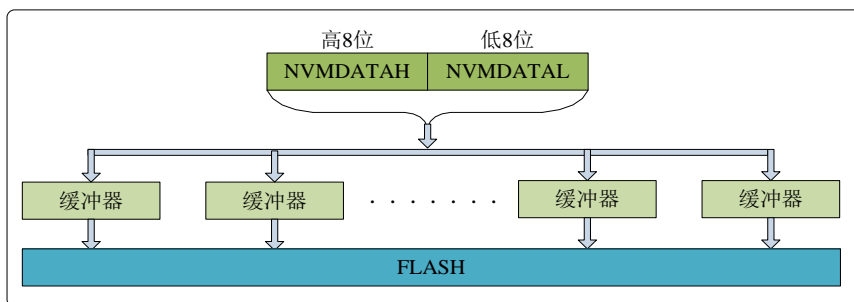


图 4.8 写操作

如图 4.8 写操作所示，在写 Flash 时，有 64 个 16 位的缓冲寄存器，用来临时存放要写入 Flash 中的数据。

注：Flash 自写的有效工作温度范围为 -20~125℃。

### 4.3.1 寄存器 NVMDATAH/L

CPU 读写 Flash 时，用来存放要写入或者读出 Flash 的数据，NVMDATAL 存放数据的低 8 位，NVMDATAH 存放数据的高 8 位。

### 4.3.2 寄存器 NVMADDRH/L

如寄存器所示，NVMADDRH/L 地址位于特殊功能寄存器区的 3AH/3BH。用来存放要写入 Flash 的 15 位的地址信息，NVMADDRH 存放地址的高 7 位，NVMADDRL 存放地址的低 8 位。

**寄存器：** NVMADDRH: 数据指针高5位(地址: 3AH)

复位值 -000 0000	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

NVMADDRL: 数据指针低8位(地址: 3BH)

复位值 0000 0000	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

NVMCTL0/NVMCTL1 为写 Flash 控制寄存器，地址位于特殊功能寄存器区的 3CH/3DH。

用户在写 Flash 时，将 NVMDATAH/L 中送入要写入的数据，NVMADDRH/L 中送入要写入的地址，然后通过向 NVMCTL0 和 NVMCTL1 送入固定的写命令，将要写入块的数据存到对应的缓冲寄存器中。

在读 Flash 时，将要读的地址送到 NVMADDRH/L 中，然后向 NVMCTL0 写入固定的读命令，把要读的数据送到 NVMDATAH/L 中。

### 4.3.3 写 Flash

写 Flash 时，只能对 Flash 成块写入数据，不允许跨区操作。不能单独将一个字节(或字)的数据写入某块的一个字节(或字)中，如果实际上写入 Flash 中的数据没有 64 个字或不能被 64 整除(例如要写入一组 15 个字的数据)，需要将块中不需要写入数据的单元写入 0 或者其它值，否则可能会导致写入的数据出错。如果原来的 Flash 保存有数据，现在需要修改原数据中的一个字或者几个字，其它单元的值不变，则需要先将其对应块中其它数据读出来保存，然后再根据实际情况将需要修改的值和之前读出的值写入即可。

在写 Flash 时，必须先对每个页的第一块进行写操作，以擦除本页的数据，如果没有对第一块进行写操作，直接写后面块则本页的所有数据都不会被擦除。即只有对每个页的第一块进行写操作后才会擦除本页的数据，对其它块写操作不会产生擦除本页数据的命令，可能导致写入数据出错。

**注：** 1.写Flash时，从Flash自写首地址开始处，连续的64个字作为一个数据块，连续的2个数据块作为一个页。

2.写Flash时，不管其存储单元是否有数据，都要先执行一次擦除命令，且擦除命令只有对每页的第一个数据块写完后才会自动执行，将本页所有单元数据擦除。而对每页的其他数据块写操作时不会有擦除命令发出。

3.将各页第一个数据块写完后，CPU将停止6ms执行擦除和写命令，写其他块时，停止3ms执行写命令。

4.需正确配置WRT<1:0>，才能对Flash目标地址区域进行写操作。



在写 FLASH 时，将要写入的数据送到 NVMDATAH/L，地址送到 NVMADDRH/L 后，通过执行以下指令完成写操作：

```

CLR  INTCTL , AIE
JNB  INTCTL , AIE
JMP  $-2
MOV  R5 , #0X84           ;使能Flash的写操作
MOV  NVMCTL0 , R5
MOV  R5 , #0X69
MOV  NVMCTL1 , R5
MOV  R5 , #0X96
MOV  NVMCTL1 , R5
SET  NVMCTL0 , 1
NOP
NOP
MOV  R5 , #0X80           ;关闭Flash的写操作，防止意外写
MOV  NVMCTL0 , R5
SET  INTCTL , 7

```

以上指令中的立即数 0X80, 0X84, 0X69, 0X96 是固定不变的。如果未完全按照上述顺序（先将 0X69 写入 NVMCTL1，再将 0X96 写入 NVMCTL1，最后将 NVMCTL0.1 位置 1）执行指令，将不会启动写操作。

#### 写 FLASH 的步骤为：

1. 将要写入的数据的送到 NVMDATAH/L；
2. 将对应的 FLASH 地址送到 NVMADDRH/L；
3. 执行上面的写命令，此时，CPU 将要写入数据的一个字保存到 FLASH 的数据缓冲器中；
4. 重复执行步骤 1、2、3 六十四次，此时 CPU 自动将要写入第一块的数据分别存入对应的 FLASH 的数据缓冲器中；
5. 当上边第 64 次写命令执行完后，CPU 自动发出擦除本页的命令，将本页原来的数据全部擦除，擦除完毕后，将数据缓冲器中的数据送到对应的地址中。在这个过程中 CPU 停止其它工作 6ms 用来执行擦除和写入数据的命令。
6. 重复执行步骤 1、2、3 六十四次，将数据写入本页的第二块。当执行完第 64 次写命令后，因本次写的不是页的第一块，CPU 不会执行擦除命令，仅将数据缓冲器中的数据写入对应的存储单元，写操作耗时 3ms。

#### 4.3.4 读 Flash

在读 FLASH 时，将要读取的地址送到 NVMADDRH/L 后，通过执行以下操作完成读操作：

```
CLR INTCTL , AIE
JNB INTCTL , AIE
JMP $-2
MOV R5 , #0X81
MOV NVMCTL0 , R5
NOP
NOP
SET INTCTL , AIE
```

上面指令中的立即数 0X81 是固定不变的。此时该地址的数据高 8 位被送 NVMDATAH，低 8 位送到 NVMDATAL。无论配置位 WRT 为何值都不影响读 FLASH。

读 FLASH 是逐字读取的，不要求一块一块的读。读 FLASH 时通过向 NVMCTL0 写入 0X81 来执行读命令。

**读 FLASH 的步骤如下：**

1. 将要读的数据单元的地址送到 NVMADDRH/L 中；
2. 向 NVMCTL0 写入读命令；
3. 两个指令周期后该单元的数据被送到 NVMDATAH/L。

## 4.4 DATA EEPROM

KF8A100 片内的 DATA EEPROM 存储器最大容量为 256×8 位,地址范围 00H~FFH, 在 CPU 正常工作期间是可读写的。DATA EEPROM 是单独编址, 可以通过特殊功能寄存器寻址。DATA EEPROM 与 Flash 自写共用 4 组寄存器。

表 4.3 与 DATA EEPROM 相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
3BH	NVMADDRL	DATA EEPROM 地址指针低 8 位							
3CH	NVMCTL0	控制寄存器 0							
3DH	NVMCTL1	控制寄存器 1							
39H	NVMDATAL	数据寄存器低 8 位							

DATA EEPROM 数据存储器只能以字节为单位进行读写。当对字节写操作时会自动擦除目标存储单元（无论有没有数据）并写入新数据（在写入前擦除）。

当器件被代码保护时, 器件编程器将不再能访问 DATA EEPROM。在代码保护时, CPU 仍可读写 DATA EEPROM 存储器。

### 4.4.1 寄存器 NVMDATAL

使用 DATA EEPROM 时, 寄存器 NVMDATAL 用来存放要写入或者读出 DATA EEPROM 的数据。

### 4.4.2 寄存器 NVMADDRL

DATA EEPROM 最大容量为 256×8 位,地址范围 00H~FFH, 需 8 位地址线参与译码。NVMADDRL 寄存器用来存放要写入 DATA EEPROM 的 8 位的地址信息。

寄存器: NVMADDRL: 数据指针低8位(地址: 3BH)

复位值	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0000 0000	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

### 4.4.1 寄存器 NVMCTL0/NVMCTL1

NVMCTL0/NVMCTL1 为写 DATA EEPROM 控制寄存器, 地址位于特殊功能寄存器区的 3CH/3DH。用户在写 DATA EEPROM 时, 将 NVMDATAL 中送入要写入的数据, NVMADDRL 中送入要写入的地址, 然后通过向 NVMCTL0 和 NVMCTL1 送入固定的写命令, 将数据写入 DATA EEPROM 对应地址指向的单元中。在读 DATA EEPROM 时, 将要读的地址送到 NVMADDRL 中, 然后向 NVMCTL0 写入固定的读命令, 把要读的数据送到 NVMDATAL 中。

### 4.4.2 写 DATA EEPROM

写 DATA EEPROM 时, 一次最多写入一个地址, 写入操作之前自动附加一个擦除操作, 擦除目标存储单元, 然后写入新的数据。

写 DATA EEPROM 时, 将要写入的地址送到 NVMADDRL, 将要写入的数据送到 NVMDATAL。之后通过执行以下操作完成写操作:

```

CLR INTCTL , AIE
JNB INTCTL , AIE
JMP $-2
MOV R5 , #0X04 ;使能DATA EEPROM的写操作
MOV NVMCTL0 , R5
MOV R5 , #0X69
MOV NVMCTL1 , R5
MOV R5 , #0X96
MOV NVMCTL1 , R5
SET NVMCTL0 , 1 ;执行写操作
NOP
NOP
MOV R5 , #0X00 ;关闭DATA EEPROM的写，防止意外写
MOV NVMCTL0 , R5
SET INTCTL , 7

```

以上指令中的立即数 0X04, 0X00, 0X69, 0X96 是固定不变的。如果未完全按照上述顺序（先将 0X69 写入 NVMCTL1，再将 0X96 写入 NVMCTL1，最后置位 NVMCTL0.1）执行指令，将不会启动写操作。写周期完成时，EE 写完成中断标志位（EEIF）置 1，用户可以允许此中断或查询此位。EEIF 必须用软件清零。

**注：**CPU写DATA EEPROM时,不管DATAP设置为何值，都能写入正确的数据

#### 写 DATA EEPROM 的步骤:

1. 将要写入的数据送到 NVMDATAL;
2. 将对应的 DATA EEPROM 地址送到 NVMADDRL;
3. 执行上面的写命令，此时，CPU 发出擦除 DATA EEPROM 目标存储单元的命令，擦除完毕后，将 NVMDATAL 中的数据送到对应地址中。用户需等待 6ms 用来执行擦除和写入数据的命令，此期间 CPU 仍可正常工作；
4. 重复执行步骤 1、2、3，可以执行其他地址的写入。

#### 4.4.3 读 DATA EEPROM

在读 DATA EEPROM 时，将要读取的地址送到 NVMADDRL 后，通过执行以下操作完成读操作：

```

MOV R0 , #0Xxx ; (将目标地址写入NVMADDRL)
MOV NVMADDRL , R0
MOV R5 , #0X01
MOV NVMCTL0 , R5
NOP
MOV R0 , NVMDATAL ;R0=NVMDATAL

```

上面指令中的立即数 0X01 是固定不变的。此时，该地址的数据被送到 NVMDATAL。

读 DATA EEPROM 是逐字读取的。读 DATA EEPROM 时通过向 NVMCTL0 写入 0X01 来执行读命令。NVMDATAL 寄存器保存数据直到下一次读命令覆盖当前值。

注：读DATA EEPROM时，不管DATAP设置为何值，都能读出正确的数据

读 DATA EEPROM 的步骤如下：

1. 将要读的数据单元的地址送到 NVMADDR1 中；
2. 向 NVMCTL0 写入读命令；
3. 一个指令周期后该单元的数据被送到 NVMDATAL。

## 4.5 寄存器组 Rn

KF8A100 芯片中有一个工作寄存器组 R0~R7，可用做间接寻址的中间寄存器，存放操作数的地址；隐含目的操作数的指令中，默认 R0 作为目的操作数(如: RRCR 0X81)；在读晶振校准值和参考电压校准值时，默认将读到的值送到 R0 中。

## 4.6 ID 地址单元

KF8A100 的程序存储器空间的最后 128 个地址单元被指定为 ID 地址单元，地址为 7F80H~7FFFH。用户可在其中存放校准值和或其它信息。正常运行时不能对这些地址单元进行访问，但在编程/校验时这些地址单元是可读写的。

## 5 汇编指令及寻址方式

### 5.1 寻址方式

KF8A100 系列单片机提供 5 种寻址方式，分别为：寄存器寻址、直接寻址、立即数寻址、寄存器间接寻址和位寻址。KF8A100 的指令可以没有操作数、一个操作数、两个操作数。

#### 5.1.1 寄存器寻址

采用这种寻址方式的指令中的操作数为寄存器组 R0-R7 的一个。

例：

CLR R0                   ； R0←0 将寄存器 R0 清 0  
只有一个操作数(R0 的值)，寻址方式为寄存器寻址。

ADD R0, R1  
两个操作数（R0 和 R1），寻址方式为寄存器寻址。

#### 5.1.2 直接寻址

在指令中的操作数为某个寄存器的直接地址，该地址指出其参与运算的数据所在的地址。直接寻址可以是：特殊功能寄存器、通用数据存储器。

例：

MOV R0,0X81           ； R0←(81H) 将 81H 单元的数据送到 R0 中指令中，源操作数寻址方式为直接寻址，目的操作数为寄存器寻址。

INC 0X3B               ； 3BH←(3BH)+1 将地址 3BH 里的值加 1，3BH 即 NVMADDRL。

指令中含有一个操作数，寻址方式为直接寻址。

#### 5.1.3 立即数寻址

在指令中的操作数为立即数。

例：

MOV R0,#0X20           ； R0←0X20 将立即数 0X20 送到寄存器 R0 中  
ADD R0,#0X20           ； R0←(R0)+0X20 寄存器 R0 的值与 0X20 相加结果送到

R0

AND R0,#0X20           ； R0←(R0)&0X20 寄存器 R0 的值与 0X20 相与结果送到

R0

以上三条指令中源操作数都是#0X20，为立即数寻址，目的操作数为寄存器寻址。

### 5.1.4 寄存器间接寻址

这种寻址方式中，寄存器的内容指定操作数的地址，即寄存器中存放的是操作数的地址。间接寻址只有两条指令 LD 和 ST。

例:

LD R0, [R1] ; R0←(R1) 将 R1 的内容所指地址单元的数据送到 R0  
指令中源操作数的寻址方式为寄存器间接寻址，目的操作数为寄存器寻址。

ST [R0], R1 ; (R0)←R1 将 R1 的内容送到 R0 的内容所指向的地址单元  
指令中目的操作数的寻址方式为寄存器间接寻址，源操作数为寄存器寻址。

### 5.1.5 位寻址

指令中的操作数是寄存器的某位，这样的寻址方式称为位寻址。

例:

CLR INTCTL,1 ; 将 INTCTL 的第 1 位清 0  
CLR 0X80,1 ; 将 80H 的第 1 位清 0  
JNB 0X80,1 ; 如果 80H 的第 1 位为 0 则跳过下一条指令执行后面的程序

## 5.2 汇编指令

KF8A100 系列单片机汇编指令共有 73 条，除子程序调用、子程序返回、中断返回、部分跳转指令为双周期指令外，其余指令均为单周期指令。所有指令都占两个字节。

按照指令的功能可将其分为:数据传送指令、算术运算指令、逻辑运算指令、位操作指令和转移指令和特殊指令。具体指令集请参考附录 2。

## 6 中断

KF8A100 单片机的中断源有:

- INT0/1/2
- T0 溢出中断
- T1/2/3/4/5/6/7/8/9/10 中断
- P0 口引脚电平变化中断
- 增强型电平变化中断
- A/D 中断
- PWM1/2 中断
- CMP1/2/3/4 中断
- CCP2/5/8/9/10 中断
- SPI 中断
- I2C 中断
- RTC 中断
- SSCI 模块 I2C 总线冲突(BCL)中断
- 通用串行通讯接口 USART2/3/4 的发送和接收中断
- 外部时钟故障(OSCFAIL)中断
- DATA EEPROM 写操作(EE)中断

在本单片机中有多个中断优先级，其中高优先级向量位于 0X0004H，低优先级向量位于 0X0014H。在中断服务程序里可通过 PCTL 的 IPEN(PCTL<3>)进行中断优先级设置，AIEH 和 AIEL 只有在 IPEN 置一后才可以写入，AIE 和 PUIE 不受影响。在中断服务子程序中通过检测相应的中断标志位来确定具体是哪个中断源触发发生。

KF8A100 中的 INT1/2、T1/2/3/4/5/6/7/8/9/10、ADC、PWM1/2、SPI、I2C、BCL、CMP1/2/3/4、CCP、USART2/3/4、RTC 和 OSCFAIL 等都属于外设，因此对应的中断称作外设中断，其它中断源产生的中断属于内部中断。中断逻辑如图 5.2 所示。在中断逻辑框图中，每个中断源有 3 个位用于控制其操作。这些位的功能分别是：

- 标志位表明发生了中断事件；
- 中断允许位允许程序跳转到中断向量地址处执行；
- 中断优先级位用于选择高优先级及还是低优先级；

通过将 IPEN 位 (PCTL<3>) 置 1，可使能中断优先级功能。

当 IPEN 置 1 时，有两个中断允许位，分别是 AIEH 和 AIEL。将 AIEH (INTCTL<7>) 置 1，可允许所有中断优先级位已置 1 的中断，即高优先级的中断。将 AIEL (INTCTL<6>) 置 1，可允许所有中断优先级位已清 0 的中断，即低优先级的中断。当中断标志位、中断允许位和中断优先级位都被置 1 时，中断将根据设置的中断优先级立即跳转到地址 0x0004H 或者 0x0014H。进低优先级中断时清 0 AIEL，退出中断时置 1 AIEL；进高优先级中断时清 0 AIEH，退出中断时置 1 AIEH。(AIEH=0 时禁止所有中断)。高优先级中断事件可以中断正在处理的低优先级中断事件。

其原理框图如图 6.1 所示：



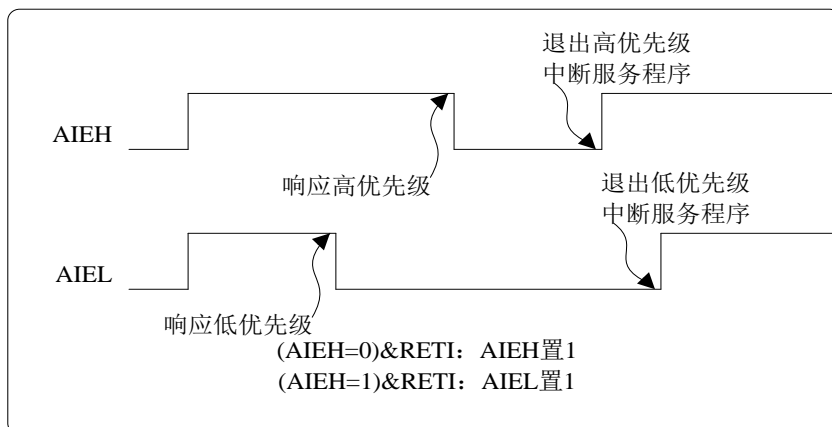


图 6.1 中断优先级工作原理图

当 IPEN 位清 0 时，就会禁止中断优先级，即为普通模式。所有中断都跳转到 0x0004H 开始执行。在普通模式下，没有中断优先级，各个中断源的中断优先级控制寄存器 IP0、IP1 和 IP2 均无效。AIE(INTCTL<7>)为全局中断使能位，PUIE(INTCTL<6>)为外设中断使能位。

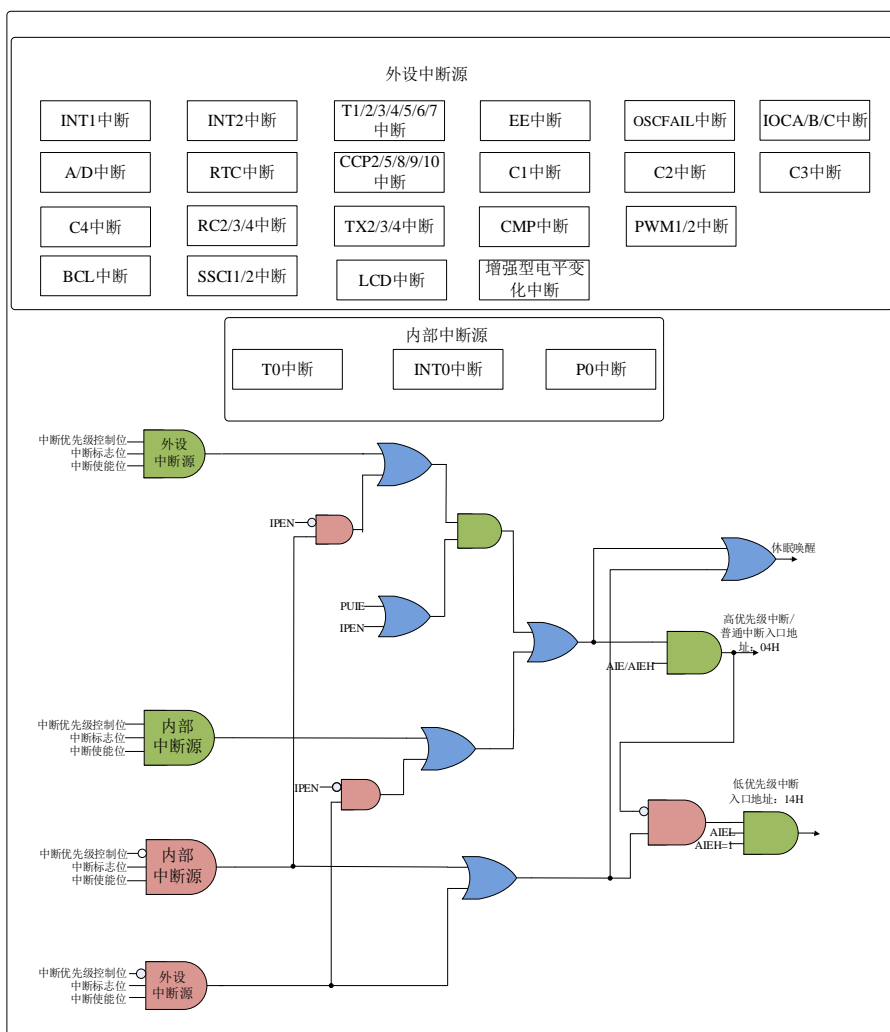


图 6.2 中断逻辑

## 6.1 中断相关的寄存器

表 6.1 与中断相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
0BH	INTCTL	AIE/ AIEH	PUIE/ AIEL	T0IE	INT0IE	P0IE	T0IF	INT0IF	P0IF
02CH	EIE1	T3IE	ADIE	-	INT1IE	C1IE	PWM2IE	T2IE	T1IE
02DH	EIE2	EEIE	C2IE	RC2IE	TX2IE	SSC11IE	CCP2IE	-	-
21BH	EIE3	-	-	TTIE	ALRIE	DAYIE	HOURLIE	MINIE	SECIE
10EH	EIE4	CCP7IE	CCP3IE	T1GIE	-	C4IE	C3IE	T4IE	BCL1IE
110H	EIE5	T7IE	T6IE	T5IE	CCP5IE	RE2IE	TE2IE	-	-
51BH	EIE6	-	WWDTI E	BCL2IE	SSC12IE	RC4IE	TX4IE	RC3IE	TX3IE
25CH	EIE7	-	-	IOCCUI E	IOCBUI E	IOCAUI E	HLVDIE	RTCT1IE	RTCT0IE
631H	EIE8			T10IE	T9IE	T8IE	CCP10IE	CCP9IE	CCP8IE
0CH	EIF1	T3IF	ADIF	-	INT1IF	C1IF	PWM2IF	T2IF	T1IF
0DH	EIF2	EEIF	C2IF	RC2IF	TX2IF	SSC11IF	CCP2IF	-	-
21AH	EIF3	-	-	TTIF	ALRIF	DAYIF	HOURLIF	MINIF	SECIF
10DH	EIF4	CCP7IF	CCP3IF	T1GIF	-	C4IF	C3IF	T4IF	BCL1IF
10FH	EIF5	T7IF	T6IF	T5IF	CCP5IF	RE2IF	TE2IF	-	-
51CH	EIF6	-	WWDTI F	BCL2IF	SSC12IF	RC4IF	TX4IF	RC3IF	TX3IF
25DH	EIF7	-	-	IOCCUIF	IOCBUIF	IOCAUI F	HLVDIF	RTCT1IF	RTCT0IF
630H	EIF8			T10IF	T9IF	T8IF	CCP10IF	CCP9IF	CCP8IF
22H	IP0	-	-	-	-	-	PT0	PINT0	PP0
23H	IP1	PT3	PADC	-	PINT1	PC1	PPWM2	PT2	PT1
24H	IP2	PEE	PC2	PRC2	PTX2	PSSC11	PCCP2	-	-
22AH	IP3	-	-	PTT	PALRM	PDAY	PHOUR	PMIN	PSEC
124H	IP4	PCCP7	PCCP3	PT1G	-	PC4	PC3	PT4	PBCL1
10CH	IP5	PT7	PT6	PT5	PCCP5	PRE2	PTE2	-	-
51DH	IP6	-	PWWDT	PBCL2	PSSC12	PRC4	PTX4	PRC3	PTX3
25EH	IP7	-	-	PIOCC	PIOCB	PIOCA	PHLVD	PRTCT1	PRTCT0
632H	IP8			PT10	PT9	PT8	PCCP10	PCCP9	PCCP8
2EH	PCTL	-	-	-	SLVREN	IPEN	SWDTE N	/POR	/LVR
15H	PWMCTL	-	INTISE	-	-	PWMMODE	PWM16 ON	PWM20 N	PWM10 N

### 6.1.1 中断控制寄存器 INTCTL

普通模式下，AIE 为全局中断使能位，当其被清 0 时，禁止所有中断。PUIE 为外设中断使能位，当其被清 0 时禁止所有外设中断。具体的中断逻辑如图 6.1 所示。

在优先级中断中，AIEH 为全局优先级中断使能位，当其被清 0 时，禁止所有中断。AIEL 为低优先级中断使能位，当其被清 0 时禁止所有低优先级中断。具体的中断逻辑如图 6.1 所示。

注：1. 当中断条件满足时，无论相应的中断使能位或者全局中断使能位AIE的状态如何，中断标志位将被硬件置1。  
2. 中断条件满足时，中断标志位通过硬件置1, 而清零则需要软件完成。

**寄存器：** INTCTL: 中断控制寄存器(地址: 0BH)

	bit7						bit0	
复位值 0000 0000	AIE/AIEH	PUIE/AIEL	TOIE	INTOIE	POIE	TOIF	INTOIF	POIF
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**AIE/AIEH:** 全局中断使能位/优先级中断使能位

当 IPEN=0

1 =使能所有未屏蔽的中断

0=禁止所有中断

当 IPEN=1

1=允许所有高优先级的中断

0=禁止所有中断

**PUIE/AIEL:** 外设中断使能位/低优先级中断使能位

当 IPEN=0

1=使能所有未屏蔽的外设中断

0=禁止所有外设中断

当 IPEN=1

1=允许所有低优先级的外设中断

0=禁止所有低优先级的外设中断

**TOIE:** T0 溢出中断使能位

1=使能 T0 中断

0=禁止 T0 中断

**INTOIE:** INTO 中断使能位

1=使能 INTO 中断

0=禁止 INTO 中断

**POIE:** P0 口电平变化中断使能位

1=使能 P0 口电平变化中断

0=禁止 P0 口电平变化中断

**TOIF:** T0 溢出中断标志位

1=T0 寄存器溢出

0=T0 寄存器未溢出

**INTOIF:** INTO 中断标志位

1=INTO 引脚产生外部中断

0=INT0 引脚未产生外部中断  
 P0IF: P0 口电平变化中断标志位  
 1=引脚 P0.0~P0.7 至少有一个电平状态发生变化  
 0=引脚 P0.0~P0.7 电平状态未发生变化

### 6.1.2 中断使能寄存器 EIE1

**寄存器:** EIE1: 中断使能寄存器(地址: 2CH)

		bit7						bit0	
复位值 0000 0-00		T3IE	ADIE	-	INT1IE	C1IE	PWM2IE	T2IE	T1IE
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

T3IE: T3 中断使能位  
 1 = 使能 T3 中断  
 0 = 禁止 T3 中断  
 ADIE: AD 中断使能位  
 1 = 使能 AD 中断  
 0 = 禁止 AD 中断  
 INT1IE: INT1 中断使能位  
 1 = 使能 INT1 中断  
 0 = 禁止 INT1 中断  
 C1IE: CMP1 中断使能位  
 1 = 使能 CMP1 中断  
 0 = 禁止 CMP1 中断  
 PWM2IE: PWM2 中断使能位  
 1 = 使能 PWM2 中断  
 0 = 禁止 PWM2 中断  
 T2IE: T2 与 PP5 匹配中断允许位  
 1 = 允许 T2 与 PP5 匹配中断  
 0 = 禁止 T2 与 PP5 匹配中断  
 T1IE: T1 中断使能位  
 1 = 使能 T1 中断  
 0 = 禁止 T1 中断

### 6.1.3 中断使能寄存器 EIE2

**寄存器:** EIE2: 中断使能寄存器(地址: 2DH)

		bit7						bit0	
复位值 0000 0000		EEIE	C2IE	RC2IE	TX2IE	SSC1IE	CCP2IE	-	-
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

EEIE: DATA EEPROM 中断使能位  
 1 = 使能 DATA-EE 中断  
 0 = 禁止 DATA-EE 中断

- C2IE: CMP2 中断使能位  
1 = 使能 CMP2 中断  
0 = 禁止 CMP2 中断
- RC2IE: USART2 接收中断使能位  
1 = 允许 RX2 中断  
0 = 禁止 RX2 中断
- TX2IE: USART2 发送中断使能位  
1 = 允许 TX2 中断  
0 = 禁止 TX2 中断
- SSCI1IE: SSCI1 中断使能位  
1 = 允许 SSCI1 中断  
0 = 禁止 SSCI1 中断
- CCP2IE: CCP2 中断使能位  
1 = 允许 CCP2 中断  
0 = 禁止 CCP2 中断

### 6.1.4 中断使能寄存器 EIE3

寄存器: EIE3: 中断使能寄存器(地址: 21BH)

复位值	bit7						bit0	
0000 0000	-	-	TTIE	ALRIE	DAYIE	HOUIE	MINIE	SECIE
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- TTIE: 时间节拍中断使能位  
1 = 使能时间节拍中断  
0 = 禁止时间节拍中断
- ALRIE: 闹钟中断使能位  
1 = 使能闹钟中断  
0 = 禁止闹钟中断
- DAYIE: 日进程中断使能位  
1 = 使能日进程中断  
0 = 禁止日进程中断
- HOUIE: 小时进程中断使能位  
1 = 使能小时进程中断  
0 = 禁止小时进程中断
- MINIE: 分进程中断使能位  
1 = 使能分进程中断  
0 = 禁止分进程中断
- SECIE: 秒进程中断使能位  
1 = 使能秒进程中断  
0 = 禁止秒进程中断

### 6.1.5 中断使能寄存器 EIE4

如寄存器所示，中断使能寄存器 EIE4 包含：

**寄存器：** EIE4: 中断使能寄存器(地址: 10EH)

		bit7						bit0	
复位值	0000 0000	CCP7IE	CCP3IE	T1GIE	-	C4IE	C3IE	T4IE	BCL1IE
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- CCP7IE: CCP7 中断使能位  
1 = 使能 CCP7 中断  
0 = 禁止 CCP7 中断
- CCP3IE: CCP3 中断使能位  
1 = 使能 CCP3 中断  
0 = 禁止 CCP3 中断
- T1GIE: T1G 中断使能位  
1 = 使能 T1G 中断  
0 = 禁止 T1G 中断
- C4IE: 比较器 4 中断使能位  
1 = 使能比较器 4 中断  
0 = 禁止比较器 4 中断
- C3IE: 比较器 3 中断使能位  
1 = 使能比较器 3 中断  
0 = 禁止比较器 3 中断
- T4IE: T4 与 PP6 匹配中断允许位  
1 = 允许 T4 与 PP6 匹配中断  
0 = 禁止 T4 与 PP6 匹配中断
- BCL1IE: 总线冲突中断使能  
1 = 使能 SSCI1 中总线冲突中断  
0 = 禁止总线冲突中断

### 6.1.6 中断使能寄存器 EIE5

如寄存器所示，中断使能寄存器 EIE5 包含：

**寄存器：** EIE5: 中断使能寄存器(地址: 110H)

		bit7						bit0	
复位值	0000 0000	T7IE	T6IE	T5IE	CCP5IE	RE2IE	TE2IE	-	-
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- T7IE: T7 中断使能位  
1 = 使能 T7 中断  
0 = 禁止 T7 中断
- T6IE: T6 中断使能位

- 1 = 使能 T6 中断
- 0 = 禁止 T6 中断
- T5IE: T5 中断使能位
  - 1 = 使能 T5 中断
  - 0 = 禁止 T5 中断
- CCP5IE: CCP5 中断使能位
  - 1 = 允许 CCP5 中断
  - 0 = 禁止 CCP5 中断
- RE2IE: UART2 接收错误中断使能位
  - 1 = 使能 UART2 接收错误中断
  - 0 = 禁止 UART2 接收错误中断
- TE2IE: UART2 发送错误中断使能位
  - 1 = 使能 UART2 发送错误中断
  - 0 = 禁止 UART2 发送错误中断

### 6.1.7 中断使能寄存器 EIE6

如寄存器所示，中断使能寄存器 EIE6 包含：

**寄存器：** EIE6: 中断使能寄存器(地址: 51BH)

		bit7					bit0		
复位值	0000 0000	-	WWDTIE	BCL2IE	SSCI2IE	RC4IE	TX4IE	RC3IE	TX3IE
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- WWDTIE: WWDT 中断使能位
  - 1 = 使能 WWDT 中断
  - 0 = 禁止 WWDT 中断
- BCL2IE: SSCI2 总线冲突中断使能
  - 1 = 使能 SSCI2 中总线冲突中断
  - 0 = 禁止 SSCI2 总线冲突中断
- SSCI2IE: SSCI2 中断使能位
  - 1 = 允许 SSCI2 中断
  - 0 = 禁止 SSCI2 中断
- RC4IE: USART4 接收中断使能位
  - 1 = 允许 RX4 中断
  - 0 = 禁止 RX4 中断
- TX4IE: USART4 发送中断使能位
  - 1 = 允许 TX4 中断
  - 0 = 禁止 TX4 中断
- RC3IE: USART3 接收中断使能位
  - 1 = 允许 RX3 中断
  - 0 = 禁止 RX3 中断
- TX3IE: USART3 发送中断使能位
  - 1 = 允许 TX3 中断
  - 0 = 禁止 TX3 中断



### 6.1.8 中断使能寄存器 EIE7

如寄存器所示，中断使能寄存器 EIE7 包含：

**寄存器：** EIE7: 中断使能寄存器(地址: 25CH)

复位值	bit7						bit0	
---000	-	-	IOCCUIE	IOCBUIE	IOCAUIE	HLVDIE	RTCT1IE	RTCT0IE
	U	U	R/W	R/W	R/W	R/W	R/W	R/W

- IOCCUIE:** IOCC 中断使能位  
1 = 使能 IOCC 中断  
0 = 未使能 IOCC 中断
- IOCBUIE:** IOCB 中断使能位  
1 = 使能 IOCB 中断  
0 = 未使能 IOCB 中断
- IOCAUIE:** IOCA 中断使能位  
1 = 使能 IOCA 中断  
0 = 未使能 IOCA 中断
- HLVDIE:** 保留位
- RTCT1IE:** RTC 定时器 TMR1 中断使能位  
1 = 使能 RTC 定时器 TMR1 中断  
0 = 未使能 RTC 定时器 TMR1 中断
- RTCT0IE:** RTC 定时器 TMR0 中断使能位  
1 = 使能 RTC 定时器 TMR0 中断  
0 = 未使能 RTC 定时器 TMR0 中断

### 6.1.9 中断使能寄存器 EIE8

如寄存器所示，中断使能寄存器 EIE8 包含：

**寄存器：** EIE8: 中断使能寄存器(地址: 631H)

复位值	bit7						bit0	
--00 0000	-	-	T10IE	T9IE	T8IE	CCP10IE	CCP9IE	CCP8IE
	U	U	R/W	R/W	R/W	R/W	R/W	R/W

- T10IE:** T10 中断使能位  
1 = 使能 T10 中断  
0 = 未使能 T10 中断
- T9IE:** T9 中断使能位  
1 = 使能 T9 中断  
0 = 未使能 T9 中断
- T8IE:** T8 中断使能位  
1 = 使能 T8 中断  
0 = 未使能 T8 中断
- CCP10IE:** CCP10 中断使能位  
1 = 使能 CCP10 中断  
0 = 未使能 CCP10 中断



- CCP9IE: CCP9 中断使能位  
 1 = 使能 CCP9 中断  
 0 = 未使能 CCP9 中断
- CCP8IE: CCP8 中断使能位  
 1 = 使能 CCP8 中断  
 0 = 未使能 CCP8 中断

### 6.1.10 中断标志寄存器 EIF1

**寄存器:** EIF1: 外设中断标志寄存器(地址0CH)

	bit7						bit0	
复位值 0000 0000	T3IF	ADIF	-	INT1IF	C1IF	PWM2IF	T2IF	T1IF
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- T3IF: T3 中断标志位  
 1 = T3 产生了中断  
 0 = T3 未产生中断
- ADIF: AD 完成中断标志位  
 1 = AD 转换完成  
 0 = AD 转换没有完成
- INT1IF: INT1 中断标志位  
 1 = INT1 引脚产生外部中断  
 0 = INT1 引脚未产生外部中断
- PWM2IF: PWM2 中断标志位  
 1 = PWM2 使能时, T1H 和 PP2 匹配  
 0 = PWM2 使能时, T1H 和 PP2 不匹配
- C1IF: 模拟比较器 CMP1 中断标志位  
 1 = 模拟比较器 CMP1 输出发生改变(必须软件清 0)  
 0 = 模拟比较器 CMP1 输出未发生改变
- T2IF: T2 与 PP5 匹配中断标志位  
 1 = 发生了 T2 与 PP5 匹配  
 0 = 未发生了 T2 与 PP5 匹配
- T1IF: T1 寄存器溢出标志位  
 1 = T1 寄存器溢出或 PWM1 使能时, T1L 与 PP1 匹配  
 0 = T1 寄存器未溢出或 PWM1 使能时, T1L 与 PP1 不匹配

### 6.1.11 中断标志寄存器 EIF2

**寄存器:** EIF2: 外设中断标志寄存器(地址: 0DH)

	bit7						bit0	
复位值 0000 0000	EEIF	C2IF	RC2IF	TX2IF	SSC1IF	CCP2IF	-	-
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- EEIF: DATA EEPROM 中断标志位



	1 = 发生 DATA-EE 中断
	0 = 未发生 DATA-EE 中断
C2IF:	模拟比较器 2 中断使能位
	1 = CMP2 输出发生改变(必须软件清 0)
	0 = CMP2 输出未发生改变
RC2IF:	USART2 接收中断标志位
	1 = RX2 产生了中断
	0 = RX2 未产生中断
TX2IF:	USART2 发送中断标志位
	1 = TX2 产生了中断
	0 = TX2 未产生中断
SSC1IF:	SSC1 中断标志位
	1 = SSC1 产生了中断
	0 = SSC1 未产生中断
CCP2IF:	CCP2 中断标志位
	1 = CCP2 产生了中断
	0 = CCP2 未产生中断

### 6.1.12 中断标志寄存器 EIF3

**寄存器:** EIF3: 中断使能寄存器(地址: 21AH)

	bit7						bit0	
复位值	-	-	TTIF	ALRIF	DAYIF	HOURIF	MINIF	SECIE
0000 0000	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TTIF:	时间节拍中断标志位
	1 = 产生了时间节拍中断
	0 = 未产生时间节拍中断
ALRIF:	闹钟中断标志位
	1 = 产生了闹钟中断
	0 = 未产生闹钟中断
DAYIF:	日进程中中断标志位
	1 = 产生了日进程中中断
	0 = 未产生日进程中中断
HOURIF:	小时进程中中断标志位
	1 = 产生了小时进程中中断
	0 = 未产生小时进程中中断
MINIF:	分进程中中断标志位
	1 = 产生了分进程中中断
	0 = 未产生分进程中中断
SECIF:	秒进程中中断标志位
	1 = 产生了秒进程中中断
	0 = 未产生秒进程中中断

## 6.1.13 中断标志寄存器 EIF4

寄存器: EIF4: 中断标志寄存器4(地址: 10DH)

		bit7						bit0	
复位值 0000 0000		CCP7IF	CCP3IF	T1GIF	-	C4IF	C3IF	T4IF	BCL1IF
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CCP7IF: CCP7 中断标志位

1 = CCP7 产生中断

0 = CCP7 未产生中断

CCP3IF: CCP3 中断标志位

1 = CCP3 产生中断

0 = CCP3 未产生中断

T1GIF: T1G 中断标志位

1 = T1G 产生中断

0 = T1G 未产生中断

C4IF: 模拟比较器 4 中断使能位

1 = CMP4 输出发生改变(必须软件清 0)

0 = CMP4 输出未发生改变

C3IF: 模拟比较器 3 中断使能位

1 = CMP3 输出发生改变(必须软件清 0)

0 = CMP3 输出未发生改变

T4IF: T4 与 PP6 匹配中断标志位

1 = 发生了 T4 与 PP6 匹配

0 = 未发生了 T4 与 PP6 匹配

BCL1IF: 总线冲突中断标志位

1 = 当配置为 I2C 主控模式时, SSC11 中发生了总线冲突

0 = 未发生总线冲突

## 6.1.14 中断标志寄存器 EIF5

如寄存器所示, 中断标志寄存器 EIF5 包含:

寄存器: EIF5: 中断使能寄存器(地址: 10FH)

		bit7						bit0	
复位值 0000 0000		T7IF	T6IF	T5IF	CCP5IF	RE2IF	TE2IF	-	-
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

T7IF: T7 中断标志位

1=T7 产生了中断

0=T7 未产生中断

T6IF: T6 中断标志位

1=T6 产生了中断

0=T6 未产生中断

T5IF: T5 中断标志位

1=T5 产生了中断

0=T5 未产生中断

- CCP5IF: CCP5 中断标志位  
1 = CCP5 产生了中断  
0 = CCP5 未产生中断
- RE2IF: UART2 接收错误中断标志位  
1 = 发生 UART2 接收错误中断  
0 = 未发生 UART2 接收错误中断
- TE2IF: UART2 发送错误中断标志位  
1 = 发生 UART2 发送错误中断  
0 = 未发生 UART2 发送错误中断

### 6.1.15 中断标志寄存器 EIF6

如寄存器所示，中断标志寄存器 EIF6 包含：

**寄存器：** EIF6: 中断使能寄存器(地址: 51CH)

复位值	bit7						bit0	
0000 0000	-	WWDTIF	BCL2IF	SSCI2IF	RC4IF	TX4IF	RC3IF	TX3IF
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- WWDTIF: WWDT 中断标志位  
1 = 发生 WWDT 中断  
0 = 未发生 WWDT 中断
- BCL2IF: SSCI2 总线冲突中断标志位  
1 = 发生 SSCI2 中总线冲突中断  
0 = 未发生 SSCI2 总线冲突中断
- SSCI2IF: SSCI2 中断标志位  
1 = 发生 SSCI2 中断  
0 = 未发生 SSCI2 中断
- RC4IF: USART4 接收中断标志位  
1 = 发生 RX4 中断  
0 = 未发生 RX4 中断
- TX4IF: USART4 发送中断标志位  
1 = 发生 TX4 中断  
0 = 未发生 TX4 中断
- RC3IF: USART3 接收中断标志位  
1 = 发生 RX3 中断  
0 = 未发生 RX3 中断
- TX3IF: USART3 发送中断标志位  
1 = 发生 TX3 中断  
0 = 未发生 TX3 中断

### 6.1.16 中断标志寄存器 EIF7

如寄存器所示，中断标志寄存器 EIF7 包含：

**寄存器:** EIF7: 中断使能寄存器(地址: 25DH)

复位值 --00 0000	bit7						bit0	
	-	-	IOCCUIF	IOCBUIF	IOCAUIF	HLVDIF	RTCT1IF	RTCT0IF
	U	U	R/W	R/W	R/W	R/W	R/W	

- IOCCUIF:** IOCCUIF 中断标志位  
1 = 检测 IOCC 组引脚有中断  
0 = 未检测 IOCC 组引脚有中断
- IOCBUIF:** IOCBUIF 中断标志位  
1 = 检测 IOCB 组引脚有中断  
0 = 未检测 IOCB 组引脚有中断
- IOCAUIF:** IOCAUIF 中断标志位  
1 = 检测 IOCA 组引脚有中断  
0 = 未检测 IOCA 组引脚有中断
- HLVDIF:** 保留位
- RTCT1IF:** RTC 定时器 TMR1 中断标志位  
1 = RTC 定时器 TMR1 溢出  
0 = RTC 定时器 TMR1 未溢出
- RTCT0IF:** RTC 定时器 TMR0 中断标志位  
1 = RTC 定时器 TMR0 溢出  
0 = RTC 定时器 TMR0 未溢出

### 6.1.17 中断使能寄存器 EIF8

如寄存器所示, 中断标志寄存器 EIF8 包含:

**寄存器:** EIF8: 中断使能寄存器(地址: 630H)

复位值 --00 0000	bit7						bit0	
	-	-	T10IF	T9IF	T8IF	CCP10IF	CCP9IF	CCP8IF
	U	U	R/W	R/W	R/W	R/W	R/W	

- T10IF:** T10 中断标志位  
1 = 定时器 T10 产生中断  
0 = 定时器 T10 未产生中断
- T9IF:** T9 中断标志位  
1 = 定时器 T9 产生中断  
0 = 定时器 T9 未产生中断
- T8IF:** T8 中断标志位  
1 = 定时器 T8 产生中断  
0 = 定时器 T8 未产生中断
- CCP10IF:** CCP10 中断标志位  
1 = CCP10 产生中断  
0 = CCP10 未产生中断
- CCP9IF:** CCP9 中断标志位  
1 = CCP9 产生中断  
0 = CCP9 未产生中断

CCP8IF: CCP8 中断标志位  
 1 = CCP8 产生中断  
 0 = CCP8 未产生中断

### 6.1.18 中断优先级控制寄存器 IP0

当 IPEN=1 时，中断优先级控制寄存器 IP0 才有效。

寄存器: IP0: 中断优先级控制寄存器0(地址: 22H)

	bit7						bit0	
复位值 ---- -000	-	-	-	-	-	PT0	PINT0	PP0
	U	U	U	U	U	R/W	R/W	R/W

PT0: T0 中断优先级控制位  
 1 = T0 中断为高优先级  
 0 = T0 中断为低优先级

PINT0: INT0 中断优先级控制位  
 1 = INT0 中断为高优先级  
 0 = INT0 中断为低优先级

PP0: P0 电平变化中断优先级控制位  
 1 = P0 电平变化中断为高优先级  
 0 = P0 电平变化中断为低优先级

### 6.1.19 中断优先级控制寄存器 IP1

当 IPEN=1 时，中断优先级控制寄存器 IP1 才有效。

寄存器: IP1: 中断优先级控制寄存器1(地址: 23H)

	bit7							bit0
复位值 0000 0000	PT3	PADC	-	PINT1	PC1	PPWM2	PT2	PT1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PT3: T3 中断高优先级控制位  
 1 = T3 中断为高优先级  
 0 = T3 中断为低优先级

PADC: AD 中断高优先级控制位  
 1 = AD 中断为高优先级  
 0 = AD 中断为低优先级

PINT1: INT1 中断高优先级控制位  
 1 = INT1 中断为高优先级  
 0 = INT1 中断为低优先级

PC1: 比较器 CMP1 中断高优先级控制位  
 1 = 比较器 CMP1 中断为高优先级  
 0 = 比较器 CMP1 中断为低优先级

PPWM2: PWM2 中断高优先级控制位

- 1 = PWM2 中断为高优先级  
0 = PWM2 中断为低优先级
- PT2: T2 中断高优先级控制位  
1 = T2 中断为高优先级  
0 = T2 中断为低优先级
- PT1: T1 中断高优先级控制位  
1 = T1 中断为高优先级  
0 = T1 中断为低优先级

### 6.1.20 中断优先级控制寄存器 IP2

当 IPEN=1 时，中断优先级控制寄存器 IP2 才有效。

**寄存器:** IP2: 中断优先级控制寄存器2(地址: 24H)

复位值 0000 0000	bit7						bit0	
	PEE	PC2	PRC2	PTX2	PSSCI1	PCCP2	-	-
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- PEE: DATA EEPROM 中断高优先级控制位  
1 = DATA-EE 中断为高优先级  
0 = DATA-EE 中断为低优先级
- PC2: 比较器 CMP2 中断高优先级控制位  
1 = 比较器 CMP2 中断为高优先级  
0 = 比较器 CMP2 中断为低优先级
- PRC2: RX2 中断高优先级控制位  
1 = RX2 中断为高优先级  
0 = RX2 中断为低优先级
- PTX2: TX2 中断高优先级控制位  
1 = TX2 中断为高优先级  
0 = TX2 中断为低优先级
- PSSCI1: SSCI1 中断高优先级控制位  
1 = SSCI1 中断为高优先级  
0 = SSCI1 中断为低优先级
- PCCP2: CCP2 中断高优先级控制位  
1 = CCP2 中断为高优先级  
0 = CCP2 中断为低优先级

### 6.1.21 中断优先级控制寄存器 IP3

当 IPEN=1 时，中断优先级控制寄存器 IP3 才有效。如寄存器所示:

**寄存器:** IP3: 中断优先级寄存器(地址: 22AH)

复位值 0000 0000	bit7						bit0	
	-	-	PTT	PALRM	PDAY	PHOUR	PMIN	PSEC
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- PTT: 时间节拍中断高优先级控制位  
1 = 时间节拍中断为高优先级  
0 = 时间节拍中断为低优先级
- PALRM: 闹铃中断高优先级控制位  
1 = 闹铃中断为高优先级  
0 = 闹铃中断为低优先级
- PDAY: 日进程中中断高优先级控制位  
1 = 日进程中中断为高优先级  
0 = 日进程中中断为低优先级
- PHOUR: 小时进程中中断高优先级控制位  
1 = 小时进程中中断为高优先级  
0 = 小时进程中中断为低优先级
- PMIN: 分进程中中断高优先级控制位  
1 = 分进程中中断为高优先级  
0 = 分进程中中断为低优先级
- PSEC: 秒进程中中断高优先级控制位  
1 = 秒进程中中断为高优先级  
0 = 秒进程中中断为低优先级

### 6.1.22 中断优先级控制寄存器 IP4

当 IPEN=1 时，中断优先级控制寄存器 IP4 才有效。如寄存器所示：

**寄存器：** IP4: 中断优先级控制寄存器4(地址: 124H)

复位值	bit7				bit0			
0000 0000	PCCP7	PCCP3	PT1G	-	PC4	PC3	PT4	PBCL1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- PCCP7: CCP 中断高优先级控制位  
1 = CCP7 中断为高优先级  
0 = CCP7 中断为低优先级
- PCCP3: CCP3 中断高优先级控制位  
1 = CCP3 中断为高优先级  
0 = CCP3 中断为低优先级
- PT1G: T1G 中断高优先级控制位  
1 = T1G 中断为高优先级  
0 = T1G 中断为低优先级
- PC4: 比较器 4 中断高优先级控制位  
1 = CMP4 中断为高优先级  
0 = CMP4 中断为低优先级
- PC3: 比较器 3 中断高优先级控制位  
1 = CMP3 中断为高优先级  
0 = CMP3 中断为低优先级
- PT4: T4 中断高优先级控制位  
1 = T4 中断为高优先级  
0 = T4 中断为低优先级



PBCL1: 总线冲突中断高优先级控制位  
 1 = 总线冲突中断为高优先级  
 0 = 总线冲突中断为低优先级

### 6.1.23 中断优先级控制寄存器 IP5

当 IPEN=1 时，中断优先级控制寄存器 IP5 才有效。如寄存器所示：

**寄存器:** IP5: 中断优先级控制寄存器(地址: 10CH)

		bit7						bit0	
复位值	0000 0000	PT7	PT6	PT5	PCCP5	PRE2	PTE2	-	-
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PT7: T7 中断高优先级控制位  
 1=T7 中断为高优先级  
 0=T7 中断为低优先级

PT6: T6 中断高优先级控制位  
 1=T6 中断为高优先级  
 0=T6 中断为低优先级

PT5: T5 中断高优先级控制位  
 1=T5 中断为高优先级  
 0=T5 中断为低优先级

PCCP5: CCP5 中断高优先级控制位  
 1 = CCP5 中断为高优先级  
 0 = CCP5 中断为低优先级

PRE2: UART2 接收错误中断优先级控制位  
 1 = UART2 接收错误中断为高优先级  
 0 = UART2 接收错误中断为低优先级

PTE2: UART2 发送错误中断优先级控制位  
 1 = UART2 发送错误中断为高优先级  
 0 = UART2 发送错误中断为低优先级

### 6.1.24 中断优先级控制寄存器 IP6

当 IPEN=1 时，中断优先级控制寄存器 IP6 才有效。如寄存器所示：

**寄存器:** IP6: 中断优先级控制寄存器(地址: 51DH)

		bit7						bit0	
复位值	0000 0000	-	PWWDT	PBCL2	PSSCI2	PRC4	PTX4	PRC3	PTX3
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWWDT: WWDT 中断优先级控制位  
 1 = WWDT 中断为高优先级  
 0 = WWDT 中断为低优先级

PBCL2: BCL2 中断优先级控制位  
 1 = BCL2 中断为高优先级  
 0 = BCL2 中断为低优先级

PSSCI2:	SSCI2 中断优先级控制位 1 = SSCI2 中断为高优先级 0 = SSCI2 中断为低优先级
PRC4:	RX4 中断优先级控制位 1 = RX4 中断为高优先级 0 = RX4 中断为低优先级
PTX4:	TX4 中断优先级控制位 1 = TX4 中断为高优先级 0 = TX4 中断为低优先级
PRC3:	RX3 中断优先级控制位 1 = RX3 中断为高优先级 0 = RX3 中断为低优先级
PTX3:	TX3 中断优先级控制位 1 = TX3 中断为高优先级 0 = TX3 中断为低优先级

### 6.1.25 中断优先级控制寄存器 IP7

当 IPEN=1 时，中断优先级控制寄存器 IP7 才有效。如寄存器所示：

**寄存器：** IP7: 中断高优先级控制寄存器(地址: 25EH)

复位值 --00 0000	bit7						bit0	
-	-	PIOCC	PIOCB	PIOCA	PHLVD	PRTCT1	PRTCT0	
U	U	R/W	R/W	R/W	R/W	R/W	R/W	

PIOCC:	IOCC 中断优先级控制位 1 = IOCC 中断为高优先级 0 = IOCC 中断为低优先级
PIOCB:	IOCB 中断优先级控制位 1 = IOCB 中断为高优先级 0 = IOCB 中断为低优先级
PIOCA:	IOCA 中断优先级控制位 1 = IOCA 中断为高优先级 0 = IOCA 中断为低优先级
PHLVD:	保留位
PRTCT1:	RTC 定时器 TMR1 中断优先级控制位 1 = RTC 定时器 TMR1 中断为高优先级 0 = RTC 定时器 TMR1 中断为低优先级
PRTCT0:	RTC 定时器 TMR0 中断优先级控制位 1 = RTC 定时器 TMR0 中断为高优先级 0 = RTC 定时器 TMR0 中断为低优先级

### 6.1.26 中断优先级控制寄存器 IP8

当 IPEN=1 时，中断优先级控制寄存器 IP8 才有效。如寄存器所示：

**寄存器:** IP8: 中断高优先级控制寄存器(地址: 632H)

		bit7					bit0		
复位值	--00 0000	-	-	PT10	PT9	PT8	PCCP10	PCCP9	PCCP8
		U	U	R/W	R/W	R/W	R/W	R/W	R/W

- PT10:** T10 中断优先级控制位  
1 = T10 中断为高优先级  
0 = T10 中断为低优先级
- PT9:** T9 中断优先级控制位  
1 = T9 中断为高优先级  
0 = T9 中断为低优先级
- PT8:** T8 中断优先级控制位  
1 = T8 中断为高优先级  
0 = T8 中断为低优先级
- PCCP10:** CCP10 中断优先级控制位  
1 = CCP10 中断为高优先级  
0 = CCP10 中断为低优先级
- PCCP9:** CCP9 中断优先级控制位  
1 = CCP9 中断为高优先级  
0 = CCP9 中断为低优先级
- PCCP8:** CCP8 中断优先级控制位  
1 = CCP8 中断为高优先级  
0 = CCP8 中断为低优先级

### 6.1.27 电源控制寄存器 PCTL

**寄存器:** PCTL: 电源控制寄存器(地址:2EH)

		bit7					bit0		
复位值	--1 00xx	-	-	-	SLVREN	IPEN	SWDTEN	$\overline{\text{POR}}$	$\overline{\text{LVR}}$
		U	U	U	R/W	R/W	R/W	R/W	R/W

- SLVREN:** 软件欠压检测使能位  
1 = 使能欠压检测  
0 = 禁止欠压检测
- IPEN:** 中断优先级控制位  
1 = 使能中断优先级功能  
0 = 禁止中断优先级, 即为普通模式
- SWDTEN:** 软件看门狗定时器使能位  
配置字的 WDTEN=0 时  
1 = 软件使能看门狗定时器  
0 = 软件禁止看门狗定时器
- $\overline{\text{POR}}$ :** 上电复位状态位  
1 = 未发生上电复位  
0 = 发生了上电复位

$\overline{\text{LVR}}$ : 欠压复位状态位  
 1 = 未发生欠压复位  
 0 = 已发生欠压复位

### 6.1.28 INT 边沿触发选择

寄存器: PWMCTL: PWM启动控制寄存器(地址: 15H)

复位值	bit7						bit0	
1100 0000	-	INT1SE	保留	保留	PWMMODE	PWM16ON	PWM2ON	PWM1ON
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

INT1SE: INT1 触发脉冲边沿选择位  
 1 = 上升沿触发  
 0 = 下降沿触

PWMMODE: PWM 模式选择位  
 1 = PWM 以 16 位 T1 计数器为时基, 控制 PWM16OUT 输出  
 0 = PWM 被分成 PWM1 和 PWM2, PWM1 以 T1L 为时基控制 PWM1OUT; PWM2 以 T1H 为时基, 控制 PWM2OUT

PWM16ON: PWM16 启动控制位 (PWMMODE=1 时有效)  
 1 = 启动 PWM16  
 0 = 禁止 PWM16

PWM2ON: PWM2 启动控制位 (PWMMODE=0 时有效)  
 1 = 启动 PWM2  
 0 = 禁止 PWM2

PWM1ON: PWM1 启动控制位 (PWMMODE=0 时有效)  
 1 = 启动 PWM1  
 0 = 禁止 PWM1

图注: R=可读 W=可写 --未用 U=未实现位

注: 保留位只能写 0, 不能写 1。

### 6.1.29 中断响应

当 IPEN=1 时, 中断被响应后:

1. 进入优先级中断, 由软件对 AIEH 或 AIEL 位都置 1。
2. 返回地址压入堆栈;
3. 中断入口地址载入 PC;
4. AIEL 位由硬件自动清 0, 执行低优先级中断服务程序, 【如果进入高优先级中断时, AIEH 由硬件自动清 0。执行高优先级中断服务程序, 执行指令 IRET 退出高优先级中断硬件自动置 1 AIEH, 继续执行低优先级中断服务程序】, 执行指令 IRET 退出低优先级中断硬件自动置 1 AIEL。
5. 只置 1 AIEH, 则直接进入高优先级中断, AIEH 由硬件自动清 0;
6. 继续执行中断服务子程序;
7. 执行指令 IRET 退出中断服务子程序并由硬件自动将 AIEH 置 1, 重新使能未屏蔽的中断;

8. 跳转到中断发生处继续执行下面的程序。

当 IPEN=0 时，中断被响应后：

1. AIE 位将被硬件清 0 以禁止其它中断；
2. 返回地址压入堆栈；
3. 中断入口地址载入 PC；
4. 执行该中断服务子程序；
5. 执行指令 IRET 退出中断服务子程序并将 AIE 置 1，重新使能未屏蔽的中断；
6. 跳转到中断发生处继续执行下面的程序。

进入中断服务程序后，首先保存 PSW 和其它寄存器的值，然后通过查询中断标志位确定中断源。在重新使能中断之前，应在软件中将相应的中断标志位清 0，以避免出错。

注 1: 中断条件满足时，无论相应的中断使能位或 AIE 位的状态为何，中断标志位都将被置 1。

2: 当执行一条清除 AIE 位的指令后，任何在下一周期等待响应的中断都将被忽略。当 AIE 位重新置 1 时，被忽略的中断请求将继续等待被响应。

3: 当对中断进行响应，进入中断服务子程序的时候硬件会将 AIE 位清零关闭总中断，当中断程序执行完，中断返回指令跳出中断子程序时，硬件将 AIE 位置 1 打开总中断。

## 6.2 INT 中断

INT 中断有三个中断源: INT0、INT1,都采用边沿触发方式,如果触发边沿选择位(INTxSE)置 1, 则采用上升沿触发; 如果触发边沿选择位清 0, 则采用下降沿触发。

### 6.2.1 INT0 中断

INT0 中断通过寄存器 INTCTL 中的 INT0IE 位置 1 使能 INT0 中断。通过 OPTR 中的 INT0SE 位设置触发边沿，INT0SE 置 1,将 INT0 设置为上升沿触发，清零设置为下降沿触发。INTCTL 中的 INT0IF 为 INT0 的中断标志位。如果 IPEN 和 PINT0 位均置 1, 则 INT0 为高优先级中断。

INT0 引脚有触发脉冲时，INT0IF 被自动置 1, 如果 INT0IE 和 AIE 位为 1, 则响应 INT0 中断。

### 6.2.2 INT1 中断

INT1 中断通过寄存器 EIE1 中的 INT1IE 位置 1 使能 INT1 中断。通过 PWMCTL 中的 INT1SE 位设置触发边沿，INT1SE 置 1,将 INT1 设置为上升沿触发，清零设置为下降沿触发。EIF1 中的 INT1IF 为 INT1 的中断标志位。如果 IPEN 和 PINT1 位均置 1, 则 INT1 为高优先级中断。

INT1 引脚有触发脉冲时，INT1IF 被自动置 1, 如果 INT1IE、PUIE 和 AIE 位为 1, 则响应 INT1 中断。

**使用 INT 中断时的设置：**

1. 将对应的 INTx 引脚设置为数字输入口。
2. 选择触发脉冲边沿时上升沿还是下降沿(INT0/1/2SE 置 1 为上升沿触发)；

3. 将相应的外部中断使能位置 1(INTxIE), 如果为高优先级, 则 IPEN 和 PINTx 均置 1。

注:x=0/1/2

## 6.3 定时器中断

有关定时/计数器模块中断的操作, 请参考定时/计数器相关章节部分。

## 6.4 P0 口中断

P0 口引脚的输入电平变化将使 P0IF(INTCTL.0)位置 1。通过设置/清除 P0IE(INTCTL.3) 位, 可使能/禁止该中断。且该端口各引脚可通过 IOCL 寄存器来对每个引脚进行配置。当 IPEN 和 PP0 均置 1 时, P0 口中断可以配置为高优先级中断。

有关 P0 口的操作, 请参考 P0 口部分。

## 6.5 ADC 中断

先将 ADEN 位置 1, 然后将 START 位(ADCCR0.1)置 1 即可启动 A/D 转换。当转换结束时, A/D 模块将 ADIF 位置 1, 如果将 AIE 与 PUIE 置 1, 且使能 ADIE, 则进入普通中断响应。如果 IPEN 置 1, 且 AIEH 与 AIEL 均置 1, 则进入中断优先级中断, 如果只置 1AIEH, 则直接进入高优先级中断。

## 6.6 PWM1/2 中断

使能 PWM1/2 后, T1L 分配给 PWM1 进行计数, T1H 分配给 PWM2 进行计数, 当 T1L/H 与 PP1/2 匹配时, 会触发相应的中断标志位 T1IF 和 PWM2IF。如果使能 T1IE 或者 PWM2IE, 则会触发中断 (AIE、PUIE 置 1)。当 IPEN 和 PPWM1/2 均置 1 时, PWM 中断可配置为高优先级中断。

在 PWM16 启用后, 当 T1 和<PWM2L:PWM1L>匹配后,其对应的输出引脚变为低电平当 T1 的计数值与<PP2:PP1>的值匹配后, 其对应的输出引脚变为高电平, 同时将 T1 清 0, 将 T1IF 置 1, 如果允许 T1 中断, 将会转入对应的中断子程序中。

详见 PWM1/2 部分。

## 6.7 模拟比较器中断

详见比较器中断章节。

## 6.8 CCP 2/5 中断

捕捉模式下, 当一个捕捉发生时, 中断请求标志位 EIF2 寄存器中的 CCP2IF 置 1; 如果使能 INTCTL 中的 AIE、PUIE 和 EIE2 中的 CCP2IE, 则会响应中断请求; 如果 IPEN 和 PCCP 均置 1, 则为高优先级中断。

比较模式下, 所有比较模式可以产生中断, 当 PWM6H0: PWM6L0 与 T1H 和 T1L 匹配时, 中断请求标志位 EIF2 寄存器中的 CCP2IF 置 1; 如果使能 INTCTL 中的 AIE、PUIE 和 EIE2 中的 CCP2IE, 则会响应中断请求; 如果 IPEN 和 PCCP 均置 1, 则为高优先级中断。

使能 PWM6 后，当 T4L 和 T4H 与 PP61H 和 PP61L 相匹配时，会触发相应的中断标志 T4IF。如果 T4IE 使能，则会触发中断（AIE、PUIE 置 1）。如果 IPEN 和 PT4 位均置 1，则为高优先级中断。

CCP5 的中断机制和 CCP2 完全一致  
详见 CCP2/5 部分。

## 6.9 CCP 8/9/10 中断

详见 [CCP8/9](#) 章节和 CCP10 章节。

## 6.10 USART 中断

通用串行通讯模块 USART 的中断分为接收中断和发送中断。详见通用串行通讯模块部分。

## 6.11 RTC 中断

RTC 中断包括、闹钟中断和节拍中断，详情请见 [RTC 中断](#) 章节。

## 6.12 SSCI 中断

SSCI 模块工作在 SPI 模式时。在发送模式下，当 SSCIBUFR 再写入一个数据后会立刻把数据写到 SSCISR 寄存器中，此时 SSCIIF 为 0。SSCISR 的数据在移位时钟的驱使下，由高到低依次把数据的 Bit7 到 Bit0 发送到 SDO 引脚。当一个字节发送完毕后会使 SSCIIF 置 1。当 AIE 和 PUIE 均置 1，且 SSCIIE 置 1，则相应中断请求。必须用软件清零。如果 IPEN 置 1，且 AIEH 与 AIEL 均置 1，则进入中断优先级中断，如果只置 1 AIEH，则直接进入高优先级中断。

在接收模式下，接收端会在移位时钟的驱动下，由高到低依次把从 SDI 引脚收到的数据移位到 SSCISR 中，当一个字节接收完毕后会把这个字节送到 SSCIBUF 中，并使 SSCIIF 信号由 0 变为 1。当 AIE 和 PUIE 均置 1，且 SSCIIE 置 1，则相应中断请求。必须用软件清零。如果 IPEN 置 1，且 AIEH 与 AIEL 均置 1，则进入中断优先级中断，如果只置 1 AIEH，则直接进入高优先级中断。

在 I2C 模式下，发生了 I2C 特殊事件，如起始信号发送完毕、数据发送完毕、停止信号发送完毕等事件将是 SSCIIF 置 1。当 AIE 和 PUIE 均置 1，且 SSCIIE 置 1，则相应中断请求。必须用软件清零。如果 IPEN 置 1，且 AIEH 与 AIEL 均置 1，则进入中断优先级中断，如果只置 1 AIEH，则直接进入高优先级中断。

## 6.13 中断现场保护

在中断响应时，硬件会把当前 PC 值加 1 入栈保存，中断结束后，硬件在将本次中断入栈时的值弹出载入 PC，继续执行后面的程序。通常，用户可能希望在中断时对一些关键寄存器的内容进行保存(例如，Rn 和 PSW)，这些都需通过软件方式实现。

## 7 电平变化中断

KF8A100 单片机除了 P0 口有简单的电平变化功能外（见 P0 章节），额外提供 3 组可配置的增强型电平变化中断引脚，分别为 IOCA、IOCB 和 IOCC。当检测到具有上升沿或下降沿、高电平或低电平的信号时可产生中断。任何独立的引脚，都可进行配置以产生中断。电平变化中断模块具有以下功能：

- 允许电平变化中断（主开关）
- 独立的引脚配置
- 可以同时检测上升沿和下降沿
- 可以检测高电平和低电平
- 独立的引脚中断标志

### 7.1 电平中断原理框图

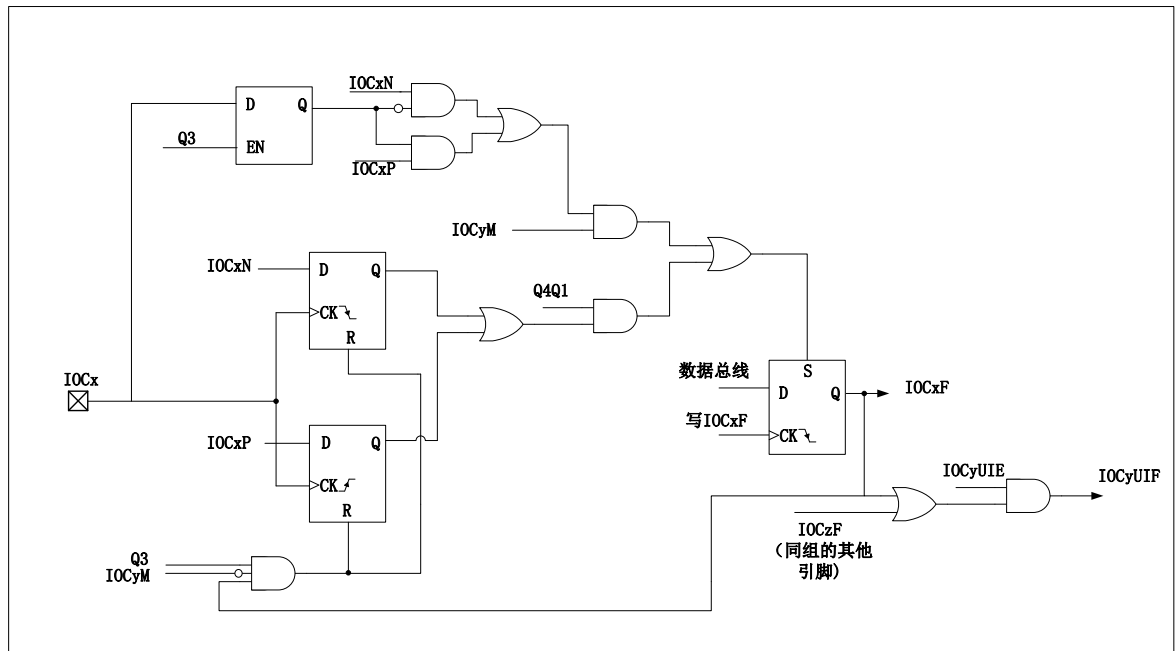


图 7.1 IOC 原理框图

3 组引脚 IOCA、IOCB 和 IOCC 分别对应 IOC0~7、IOC8~15、IOC16~23。

当 IOCyM(y=A/B/C) 被配置为 0 时，该组所有引脚都用与检测边沿信号。组内的每个引脚都带有一个上升沿检测器和一个下降沿检测器。要检测某个引脚的上升沿，必须将 IOCyPOS(y=A/B/C) 寄存器中相应的 IOCxP 位置 1。要检测某个引脚的下降沿，必须将 IOCyNEG(y=A/B/C) 寄存器中相应的 IOCxN 位置 1。通过分别将 IOCyPOS 寄存器中相应的 IOCxP 位和 IOCyNEG 寄存器中相应的 IOCxN 位置 1，可将某个引脚配置为同时检测上升沿和下降沿。

当 IOCyM(y=A/B/C) 被配置为 1 时，该组所有引脚都用与检测电平信号。要检测某个引脚的高电平，必须将 IOCyPOS(y=A/B/C) 寄存器中相应的 IOCxP 位置 1。要检测某个引脚的低电平，必须将 IOCyNEG(y=A/B/C) 寄存器中相应的 IOCxN 位置 1。如果将 IOCyPOS 寄存器



中相应的IOCxP位和IOCyNEG寄存器中相应的IOCxN位置1，则无条件置位IOCxF。

## 7.2 使能模块

要允许独立的引脚产生中断，必须将EIE7寄存器的IOCyUIE位置1。如果IOCyUIE位被禁止，则仍然会对引脚上的信号进行边沿检测，但是不产生中断。

## 7.3 电平变化相关的寄存器

表 7.1 与 T0 相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
413H	IOCAPOS	IOC7P	IOC6P	IOC5P	IOC4P	IOC3P	IOC2P	IOC1P	IOC0P
416H	IOCANEG	IOC7N	IOC6N	IOC5N	IOC4N	IOC3N	IOC2N	IOC1N	IOC0N
410H	IOCAF	IOC7F	IOC6F	IOC5F	IOC4F	IOC3F	IOC2F	IOC1F	IOC0F
414H	IOCBPOS	IOC15P	IOC14P	IOC13P	IOC12P	IOC11P	IOC10P	IOC9P	IOC8P
418H	IOCBNEG	IOC15N	IOC14N	IOC13N	IOC12N	IOC11N	IOC10N	IOC9N	IOC8N
411H	IOCBF	IOC15F	IOC14F	IOC13F	IOC12F	IOC11F	IOC10F	IOC9F	IOC8F
415H	IOCCPOS	IOC23P	IOC22P	IOC21P	IOC20P	IOC19P	IOC18P	IOC17P	IOC16P
419H	IOCCNEG	IOC23N	IOC22N	IOC21N	IOC20N	IOC19N	IOC18N	IOC17N	IOC16N
412H	IOCCF	IOC23F	IOC22F	IOC21F	IOC20F	IOC19F	IOC18F	IOC17F	IOC16F
41AH	IOCMOD						IOCCM	IOCBM	IOCAM
25CH	EIE7	-	-	IOCCUI E	IOCBUI E	IOCAUI E	HLVDIE	RTCT1I E	RTCT0I E
25DH	EIF7	-	-	IOCCUI F	IOCBUI F	IOCAUI F	HLVDIF	RTCT1I F	RTCT0I F
25EH	IP7	-	-	PIOCC	PIOCB	PIOCA	PHLVD	PRTCT1	PRTCT0

### 7.3.1 IOCAPOS 控制寄存器

寄存器: IOCAPOS: 选择寄存器(地址: 403H)

复位值	bit7							bit0
0000 0000	IOC7P	IOC6P	IOC5P	IOC4P	IOC3P	IOC2P	IOC1P	IOC0P
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IOC7P: 正极性电平变化允许位

当 IOCAM=0 时

1 = 允许引脚上的上升沿电平变化中断。检测到边沿时，将中断标志位IOC7F置1。

0 = 禁止相关引脚的电平变化中断。

当 IOCAM=1 时

1 = 允许引脚上的高电平变化中断。检测到高电平时，将中断标志位IOC7F置1。

0 = 禁止相关引脚的电平变化中断。

- IOC6P:** 正极性电平变化允许位  
当 IOCAM=0 时  
1 = 允许引脚上的上升沿电平变化中断。检测到边沿时，将中断标志位IOC6F置1。  
0 = 禁止相关引脚的电平变化中断。  
当 IOCAM=1 时  
1 = 允许引脚上的高电平变化中断。检测到高电平时，将中断标志位IOC6F置1。  
0 = 禁止相关引脚的电平变化中断。
- IOC5P:** 正极性电平变化允许位  
当 IOCAM=0 时  
1 = 允许引脚上的上升沿电平变化中断。检测到边沿时，将中断标志位IOC5F置1。  
0 = 禁止相关引脚的电平变化中断。  
当 IOCAM=1 时  
1 = 允许引脚上的高电平变化中断。检测到高电平时，将中断标志位IOC5F置1。  
0 = 禁止相关引脚的电平变化中断。
- IOC4P:** 正极性电平变化允许位  
当 IOCAM=0 时  
1 = 允许引脚上的上升沿电平变化中断。检测到边沿时，将中断标志位IOC4F置1。  
0 = 禁止相关引脚的电平变化中断。  
当 IOCAM=1 时  
1 = 允许引脚上的高电平变化中断。检测到高电平时，将中断标志位IOC4F置1。  
0 = 禁止相关引脚的电平变化中断。
- IOC3P:** 正极性电平变化允许位  
当 IOCAM=0 时  
1 = 允许引脚上的上升沿电平变化中断。检测到边沿时，将中断标志位IOC3F置1。  
0 = 禁止相关引脚的电平变化中断。  
当 IOCAM=1 时  
1 = 允许引脚上的高电平变化中断。检测到高电平时，将中断标志位IOC3F置1。  
0 = 禁止相关引脚的电平变化中断。
- IOC2P:** 正极性电平变化允许位  
当 IOCAM=0 时  
1 = 允许引脚上的上升沿电平变化中断。检测到边沿时，将中断标志位IOC2F置1。  
0 = 禁止相关引脚的电平变化中断。  
当 IOCAM=1 时  
1 = 允许引脚上的高电平变化中断。检测到高电平时，将中断标志位IOC2F置1。  
0 = 禁止相关引脚的电平变化中断。
- IOC1P:** 正极性电平变化允许位  
当 IOCAM=0 时  
1 = 允许引脚上的上升沿电平变化中断。检测到边沿时，将中断标志位IOC1F置1。

0 = 禁止相关引脚的电平变化中断。

当 IOCAM=1 时

1 = 允许引脚上的高电平变化中断。检测到高电平时，将中断标志位IOC1F置1。

0 = 禁止相关引脚的电平变化中断。

IOC0P: 正极性电平变化允许位

当 IOCAM=0 时

1 = 允许引脚上的上升沿电平变化中断。检测到边沿时，将中断标志位IOC0F置1。

0 = 禁止相关引脚的电平变化中断。

当 IOCAM=1 时

1 = 允许引脚上的高电平变化中断。检测到高电平时，将中断标志位IOC0F置1。

0 = 禁止相关引脚的电平变化中断。

### 7.3.2 IOCANE G 控制寄存器

寄存器: IOCANE G: 控制寄存器(地址: 406H)

复位值	bit7							bit0
0000 0000	IOC7N	IOC6N	IOC5N	IOC4N	IOC3N	IOC2N	IOC1N	IOC0N
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IOC7N: 负极性电平变化允许位

当 IOCAM=0 时

1 = 允许引脚上的下降沿电平变化中断。检测到边沿时，将中断标志位IOC7F置1。

0 = 禁止相关引脚的电平变化中断。

当 IOCAM=1 时

1 = 允许引脚上的低电平变化中断。检测到低电平时，将中断标志位IOC7F置1。

0 = 禁止相关引脚的电平变化中断。

IOC6N: 负极性电平变化允许位

当 IOCAM=0 时

1 = 允许引脚上的下降沿电平变化中断。检测到边沿时，将中断标志位IOC6F置1。

0 = 禁止相关引脚的电平变化中断。

当 IOCAM=1 时

1 = 允许引脚上的低电平变化中断。检测到低电平时，将中断标志位IOC6F置1。

0 = 禁止相关引脚的电平变化中断。

IOC5N: 负极性电平变化允许位

当 IOCAM=0 时

1 = 允许引脚上的下降沿电平变化中断。检测到边沿时，将中断标志位IOC5F置1。

0 = 禁止相关引脚的电平变化中断。

当 IOCAM=1 时

- 1 = 允许引脚上的低电平变化中断。检测到低电平时，将中断标志位IOC5F置1。  
0 = 禁止相关引脚的电平变化中断。
- IOC4N: 负极性电平变化允许位  
当 IOCAM=0 时  
1 = 允许引脚上的下降沿电平变化中断。检测到边沿时，将中断标志位IOC4F置1。  
0 = 禁止相关引脚的电平变化中断。  
当 IOCAM=1 时  
1 = 允许引脚上的低电平变化中断。检测到低电平时，将中断标志位IOC4F置1。  
0 = 禁止相关引脚的电平变化中断。
- IOC3N: 负极性电平变化允许位  
当 IOCAM=0 时  
1 = 允许引脚上的下降沿电平变化中断。检测到边沿时，将中断标志位IOC3F置1。  
0 = 禁止相关引脚的电平变化中断。  
当 IOCAM=1 时  
1 = 允许引脚上的低电平变化中断。检测到低电平时，将中断标志位IOC3F置1。  
0 = 禁止相关引脚的电平变化中断。
- IOC2N: 负极性电平变化允许位  
当 IOCAM=0 时  
1 = 允许引脚上的下降沿电平变化中断。检测到边沿时，将中断标志位IOC2F置1。  
0 = 禁止相关引脚的电平变化中断。  
当 IOCAM=1 时  
1 = 允许引脚上的低电平变化中断。检测到低电平时，将中断标志位IOC2F置1。  
0 = 禁止相关引脚的电平变化中断。
- IOC1N: 负极性电平变化允许位  
当 IOCAM=0 时  
1 = 允许引脚上的下降沿电平变化中断。检测到边沿时，将中断标志位IOC1F置1。  
0 = 禁止相关引脚的电平变化中断。  
当 IOCAM=1 时  
1 = 允许引脚上的低电平变化中断。检测到低电平时，将中断标志位IOC1F置1。  
0 = 禁止相关引脚的电平变化中断。
- IOC0N: 负极性电平变化允许位  
当 IOCAM=0 时  
1 = 允许引脚上的下降沿电平变化中断。检测到边沿时，将中断标志位IOC0F置1。  
0 = 禁止相关引脚的电平变化中断。  
当 IOCAM=1 时  
1 = 允许引脚上的低电平变化中断。检测到低电平时，将中断标志位IOC0F置1。  
0 = 禁止相关引脚的电平变化中断。

## 7.3.3 IOCMOD 控制寄存器

寄存器: IOCMOD: 控制寄存器(地址: 40AH)

复位值	bit7					bit0		
0000 0000	-	-	-	-	-	IOCCM	IOCBM	IOCAM
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IOCCM: C 组引脚电平变化检测选择位

- 1: C 组引脚检测电平敏感信号
- 0: C 组引脚检测边沿敏感信号

IOCBM: B 组引脚电平变化检测选择位

- 1: B 组引脚检测电平敏感信号
- 0: B 组引脚检测边沿敏感信号

IOCAM: A 组引脚电平变化检测选择位

- 1: A 组引脚检测电平敏感信号
- 0: A 组引脚检测边沿敏感信号

## 7.3.4 IOCAF 状态寄存器

寄存器: IOCAF: 控制寄存器(地址: 400H)

复位值	bit7							bit0
0000 0000	IOC7F	IOC6F	IOC5F	IOC4F	IOC3F	IOC2F	IOC1F	IOC0F
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IOC7F: IOC7 引脚电平变化中断标志位

- 1: 检测到 IOC7 引脚上发生允许的中断
- 0: 未检测到 IOC7 引脚上发生允许的中断

IOC6F: IOC6 引脚电平变化中断标志位

- 1: 检测到 IOC6 引脚上发生允许的中断
- 0: 未检测到 IOC6 引脚上发生允许的中断

IOC5F: IOC5 引脚电平变化中断标志位

- 1: 检测到 IOC5 引脚上发生允许的中断
- 0: 未检测到 IOC5 引脚上发生允许的中断

IOC4F: IOC4 引脚电平变化中断标志位

- 1: 检测到 IOC4 引脚上发生允许的中断
- 0: 未检测到 IOC4 引脚上发生允许的中断

IOC3F: IOC3 引脚电平变化中断标志位

- 1: 检测到 IOC3 引脚上发生允许的中断
- 0: 未检测到 IOC3 引脚上发生允许的中断

IOC2F: IOC2 引脚电平变化中断标志位

- 1: 检测到 IOC2 引脚上发生允许的中断

- 0: 未检测到 IOC2 引脚上发生允许的中断
- IOC1F: IOC1 引脚电平变化中断标志位
- 1: 检测到 IOC1 引脚上发生允许的中断
- 0: 未检测到 IOC1 引脚上发生允许的中断
- IOC0F: IOC0 引脚电平变化中断标志位
- 1: 检测到 IOC0 引脚上发生允许的中断
- 0: 未检测到 IOC0 引脚上发生允许的中断

B 组和 C 组的中断机制和 A 组完全一致，这里不赘述 B/C 组寄存器的相关信息，请参照 A 组寄存器说明进行设置，注意引脚是否和寄存器一一对应。

## 7.4 中断标志

IOCyF 寄存器中的 IOCxF 位是与电平变化中断引脚对应的状态标志位。如果在相应的已使能引脚上检测到预期的事件，则该引脚的状态标志位将置 1，且如果 IOCyUIE 位置 1 的话，还会产生中断。EIF7 寄存器的 IOCyUIF 位反映所有该组内 IOCxF 位的状态。

## 7.5 清零中断标志

各个状态标志（IOCxF 位）可通过将其复位为零进行清零。如果在清零操作期间又检测到事件，那么在清零序列结束后相关状态标志位会置 1，而与实际写入的值无关。

## 8 定时器/计数器

KF8A100 单片机提供一个 8 位的定时器/计数器 T0、10 个 16 位的定时器（/计数器）T1/T2/T3/T4/T5/T6/T7/8/9/10。

- 定时器 0：带有 8 位预分频器的 8 位定时器/计数器
- 定时器 1：带重载功能、门控和预分频器的 16 位定时器/计数器
- 定时器 2/4/5/6/8/9/10：带 16 位周期寄存器、预分频器和后分频器 16 位定时器
- 定时器 3/7：带 CCP、重载功能、预分频器及时钟源多选的 16 位定时器/计数器

### 8.1 定时器/计数器 T0

T0 是一个 8 位的定时器/计数器，当 T0 寄存器值加到 255 时，再加 1，则会产生溢出，T0 寄存器的值返回到 0 开始重新计数。

#### 8.1.1 T0 原理框图

图 8.1 为 T0 的原理框图。T0 模块使用一个 8 位计数器作为预分频器，如寄存器 OPTR 所示，通过软件设定 PSA 位(OPTR.3)的状态可对预分频器的分配进行控制，PSA 位清 0 可将预分频器分配给 T0 模块。通过设置 PS<2:0>位可选择预分频器的分频比。预分频器是不可读写的。当预分频器用于 T0 模块时，所有写入 T0 寄存器的指令都会将预分频器清 0。当预分频器用于 WDT 时，CWDT 指令会同时将预分频器和看门狗定时器清 0。

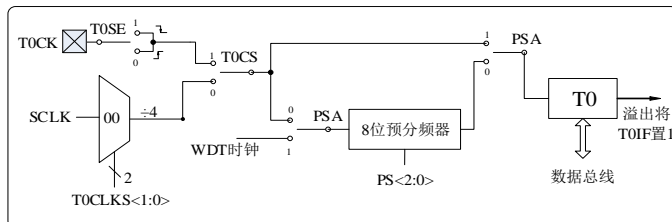


图 8.1 T0 原理框图

#### 8.1.2 T0 相关的寄存器

表 8.1 与 T0 相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
01H	T0	8 位计数器							
21H	OPTR	$\overline{\text{PUPH}}$	INT0SE	T0CS	T0SE	PSA	PS2	PS1	PS0
4FH	T0CTL	-	-	-	-	-	-	T0CLK1	T0CLK0

8.1.2.1 OPTR 选择寄存器

寄存器: OPTR: 选择寄存器(地址: 21H)

复位值	bit7							bit0
1111 1111	PUPH	INT0SE	T0CS	T0SE	PSA	PS2	PS1	PS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- PUPH** IO 口上拉功能总使能位  
1 = 禁止所有 IO 口上拉功能  
0 = 允许所有 IO 口使用上拉功能
- INT0SE** INT0 中断触发脉冲边沿选择位  
1 = INT0 引脚为上升沿触发  
0 = INT0 引脚为下降沿触发
- T0CS:** T0 工作时钟选择位  
1 = T0 的时钟源为外部时钟 T0CK 引脚 (默认) 计数模式  
0 = T0 的时钟源通过 T0CTL 寄存器的 T0CLK<1:0>位选择 (自带 4 分频)
- T0SE:** T0 计数脉冲信号边沿选择位  
1 = 下降沿触发  
0 = 上升沿触发
- PSA:** 预分频器分配控制位  
1 = 预分频器用于 WDT  
0 = 预分频器用于 T0
- PS<2:0>:** 预分频器分频比选择位

PS<2:0>	WDT 分频比	T0 分频比
000	1 : 1	1 : 2
001	1 : 2	1 : 4
010	1 : 4	1 : 8
011	1 : 8	1 : 16
100	1 : 16	1 : 32
101	1 : 32	1 : 64
110	1 : 64	1 : 128
111	1 : 128	1 : 256



### 8.1.2.2 T0CTL 控制寄存器

如下寄存器所示，T0 控制寄存器 T0CTL 包括 T0 工作时钟选择位 T0CTL<1:0>，用于选择 T0 工作时所用到的时钟。

寄存器: T0CTL: T0控制寄存器(地址:4FH)

复位值	bit7						bit0	
---- --00	-	-	-	-	-	-	T0CLK1	T0CLK0
	U	U	U	U	U	U	R/W	R/W

T0CLK<1:0>: T0 工作时钟选择位

- 00 = 选择 SCLK 为 T0 工作时钟
- 01 = 保留
- 10 = 保留
- 11 = 保留

### 8.1.3 定时模式

T0 工作时钟源通过 T0CS(OPTR.5)位和 T0CTL 寄存器的 T0CLK<1:0>位选择。当 T0 工作时钟源为内部时钟时，T0 工作在定时模式。

如果 T0 寄存器被写入初始值，则在接下来的两个指机器周期将不执行递增操作，用户可通过将校正值写入 T0 寄存器进行修正。

### 8.1.4 计数模式

当 T0 工作时钟源为外部时钟时，T0 工作在计数模式。在该模式下，T0 工作时钟源可以为 EXTHF、EXTLF 或者 T0CK。

在计数模式下，要求 T0 的工作时钟周期大于 4 个系统时钟周期，以实现 T0CK 与内部相位时钟的同步。

### 8.1.5 T0 的使用

T0 在使用时通过以下步骤进行设置:

1. 通过将 T0CS 位 (OPTR.5) 和 T0CLK<1:0>位选择 T0 的工作时钟源;
2. 如果需要分频, 则将预分频器分配给 T0, 并设置分频比;
3. 通过 T0SE 位 (OPTR.4) 选择 T0 计数脉冲信号边沿;
4. 给 T0 寄存器设置初始值;
5. 如果使用中断方式则将 T0IE 和 AIE 位置 1。

## 8.2 定时器/计数器 T1

T1 是一个 16 位的定时器/计数器，T1H/T1L 分别为 T1 的高 8 位和低 8 位数据寄存器，当 T1 计数值达到 65535 后，T1 的值再加 1 就会产生溢出，将 T1 中断标志位置 1。T1 属于外部单元，因此在使用 T1 中断时，需将 PUIE 位置 1，使能外设中断。如图 8.2 所示为 T1 的原理框图。

### 8.2.1 T1 原理框图

T1 的原理框图如图 8.2 所示，T1 是一个带有门控和预分频的 16 位定时器/计数器，计数时钟可选择 SCLK、HFCLK 和 LFCLK 或者来自外部时钟 T1CK，当 T1 与内部时钟一起使用时，T1 用作定时器，当 T1 与外部时钟一起使用时，T1 工作在计数器模式，通过对  $\overline{T1SY}$  位(T1CTL<2>)设置可使 T1 工作在异步计数器模式。

该 T1 模块还带有四个预分频器选择项，允许对时钟输入进行 1、2、4 或 8 倍分频。T1CKS 位(T1CTL<5:4>)对预分频计数器进行控制，T1 预分频计数器不能直接进行读写操作，可通过写入 T1H 或 T1L 使预分频计数器清 0。

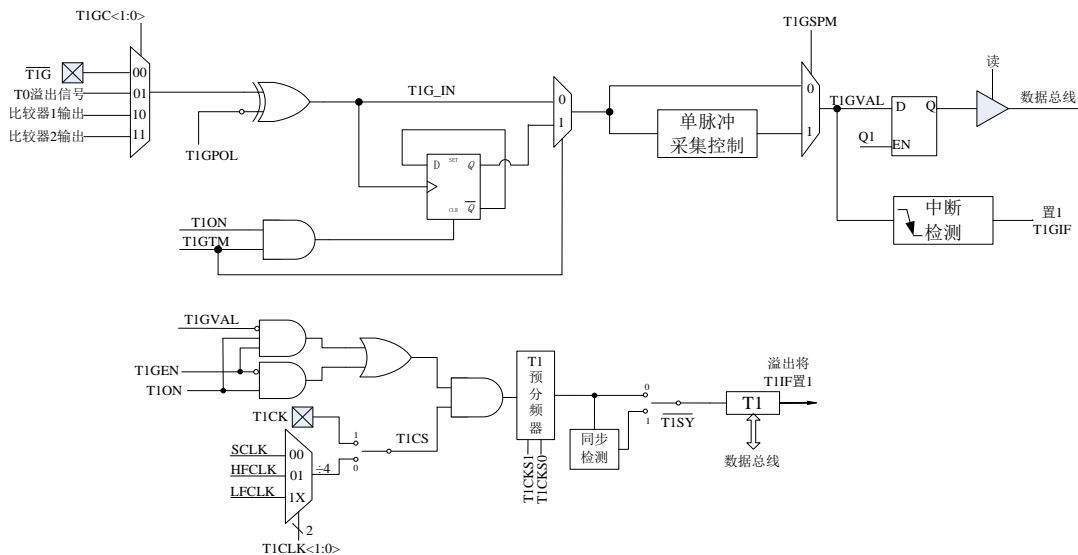


图 8.2 T1 原理框图

### 8.2.2 T1 相关的寄存器

表 8.2 与 T1 相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
0EH	T1L	T1 低 8 位							
0FH	T1H	T1 高 8 位							
10H	T1CTL	-	T1GC1	T1GC0	T1GEN	T1PL	$\overline{T1SY}$	T1CS	T1ON
15H	PWMCTL	-	INTISE	保留	保留	PWMMO DE	PWM160 N	PWM20 N	PWM10 N
118H	T1CTL2	T1GVAL	T1GSPM	T1GTM	T1GGO	T1CKS1	T1CKS0	T1CLK1	T1CLK0

16H	PP1	T1 重载计数值低 8 位
32H	PP2	T1 重载计数值高 8 位

### 8.2.2.1 T1 控制寄存器 (T1CTL)

如下寄存器所示, T1 控制寄存器 (T1CTL) 用于启动/禁止 T1 以及选择 T1 模块的不同功能特性。

**寄存器:** T1CTL: T1控制寄存器(地址: 10H)

	bit7						bit0	
复位值 0000 0000	保留	T1GC1	T1GC0	T1GEN	T1PL	$\overline{T1SY}$	T1CS	T1ON
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**T1GC<1:0>:** T1 门控选择位

00= T1G 引脚

01 = T0 溢出信号 (T0 计数值 FFH 到 00H)

00= 比较器 1 输出

01 = 比较器 2 输出

**T1GEN:** T1 门控使能位

如果 T1ON=0 则该位被忽略

如果 T1ON=1 则:

1 = 使能门控控制

0 = 禁止门控控制

**T1PL:** T1 门控极性选择位

1 = 高电平有效

0 = 低电平有效

**$\overline{T1SY}$ :** T1 计数模式外部触发脉冲输入同步控制位

T1 工作时钟源为外部时钟:

1 = 外部触发脉冲输入不与系统时钟同步

0 = 外部触发脉冲输入与系统时钟同步

T1 工作时钟源为内部时钟:

该位被忽略

**T1CS:** T1 时钟选择位

1 = T1 时钟为外部时钟 T1CK 引脚

0 = T1 时钟为 OSC/4

注: 振荡器时钟 OSC 可以通过 T1CLK<1:0>位选择相应的时钟信号

**T1ON:** T1 启动控制位

1 = 启动 T1

0 = 停止 T1

### 8.2.2.2 T1 控制寄存器 2 (T1CTL2)

寄存器: T1CTL2: T1控制寄存器2(地址: 118H)

复位值 u000 0000	bit7						bit0	
	T1GVAL	T1GSPM	T1GTM	T1GGO	T1CKS1	T1CKS0	T1CLK1	T1CLK0
	R	U	U	U	U	U	R/W	R/W

**T1GVAL:** T1门控当前状态位

指示提供给T1 门控信号的当前状态。不受T1 门控使能位 (T1GEN) 的影响

**T1GSPM:** T1 门控单脉冲模式位

- 1 = 使能T1门控单脉冲模式且正在控制T1 门控信号
- 0 = 禁止T1门控单脉冲模式

**T1GTM:** T1 门控交替计数模式位

- 1 = 使能 Timer1 门控交替计数模式
- 0 = 禁止 Timer1 门控交替计数模式

**T1GGO:** T1门控单脉冲采集状态位

- 1 = T1门控单脉冲采集已就绪, 正在等待边沿
- 0 = T1门控单脉冲采集已完成或还未开始

**T1CKS<1:0>:** T1 输入时钟预分频比选择位

- 11 = 1/8 倍预分频比
- 10 = 1/4 倍预分频比
- 01 = 1/2 倍预分频比
- 00 = 1/1 倍预分频比

**T1CLK<1:0>:** T1 工作时钟选择位

- 00 = 选用 SCLK 时钟为 T1 工作时钟
- 01 = 选用 HFCLK 时钟为 T1 工作时钟
- 1X = 选用 LFCLK 时钟为 T1 工作时钟

## 8.2.3 定时模式

当 T1 工作时钟源选择内部时钟 SCLK 时, T1 工作在定时模式, 对单片机内部时钟进行计数。

## 8.2.4 门控模式

T1 可配置为门控电路来使能和禁止计数功能, T1 的门控可选多个信号源驱动。

### 8.2.4.1 T1 门控使能

通过将 T1CTL 的 T1GEN 位置 1 来使能 T1 门控使能模式。可使用 T1CTL 寄存器的 T1PL 来配置 T1 门控使能模式的极性。

使能 T1 门控使能模式时, T1 在 T1 时钟源的上升沿递增计数。禁止 T1 门控使能模式

时，不进行递增计数且 T1 保持当前计数。时序详情请参见下表

T1CLK	T1PL	T1G	T1 工作状态
↑	0	0	计数
↑	0	1	保持计数
↑	1	0	保持计数
↑	1	1	计数

#### 8.2.4.1 T1 门控信号源选择

可从 4 个门控信号源中选择 1 个作为 T1 门控信号源。门控信号源选择由 T1CTL 的 T1GC<1:0>位控制。也可以选择每个门控信号源的极性。门控信号源的极性选择由 T1CTL 的 T1PL 控制。

T1G 引脚是 T1 门控信号源之一。可用来为 T1 门控电路提供门控信号源。

T0 溢出作为门控信号源，当 T0 从 FFh 递增至 00h 时，将自动生成一个由低到高的脉冲信号并内部提供给 T1 门控电路。

比较器 1 输出 C1OUT 和比较器 2 输出 C2OUT 可用作 T1 门控信号源。

#### 8.2.4.2 T1 门控交替计数模式

当使能了 T1 门控交替计数模式时，可以测量 T1 门控信号的完整周期长度，而不是单电平脉冲信号的持续时间。T1 门控信号源经由一个单稳态触发器输送到 T1，该单稳态触发器在信号的每个递增边沿改变状态。

通过将 T1CTL1 寄存器的 T1GTM 位置 1 来使能 T1 的门控交替计数模式。当 T1GTM 位清零时，单稳态触发器也将清零并保持清零状态。该模式对于控制要计数的边沿是必需的。

注：使能交替计数模式的同时更改门控信号的极性可能会导致操作不确定。

#### 8.2.4.3 T1 门控单脉冲模式

使能了 T1 门控单脉冲模式时，可以捕捉单脉冲门控事件。首先，通过将 T1CTL1 寄存器中的 T1GSPM 位置 1 来使能 Timer1 门控单脉冲模式。其次，必须将 T1CTL1 寄存器中的 T1GGO 位置 1。T1 将在下一个递增沿完全使能。在脉冲信号的后沿，T1GGO 位将自动清零。不允许其他门控事件使 T1 递增计数，直到用软件将 T1GGO 位再次置 1。具体工作时序，参见图示。

如果通过清零 T1CTL1 寄存器中的 T1GSPM 位禁止门控单脉冲模式，T1GGO 位也会清零。同时使能交替计数模式和单脉冲模式将允许这两种模式一起工作。从而允许测量 Timer1 门控信号源上的周期数。

#### 8.2.4.4 T1 门控值状态

采用 T1 门控值状态时，可以读取门控值的当前电平。该值存储在 T1CLT1 寄存器中的 T1GVAL 位中。即使 T1 门控未使能（T1GEN 位清零），T1GVAL 位也是有效的。

### 8.2.4.5 T1 门控中断

允许Timer1 门控事件中断时，可以在门控事件结束时产生中断。出现T1GVAL 的下降沿时，EIF4寄存器中的T1GIF 标志位将置1。如果EIE4 寄存器中的T1GIE 位置1，则将响应中断。

即使T1门控未使能（T1GEN位清零），T1GIF标志位也是有效的。

### 8.2.5 计数模式

当 T1 工作时钟源选择外部时钟时或者选择 HFCLK/LFCLK，T1 工作在计数模式，对单片机外部时钟进行计数。T1 的计数模式又有同步计数和异步计数两种方式：

如果控制位  $\overline{T1SY}$  (T1CTL2)置 1，则 T1 工作在异步计数模式。计数器根据外部时钟的脉冲进行递增计数。在休眠模式下，如果 TICK 作为 T1 的工作时钟，计数器将继续递增并在溢出时产生中断以唤醒处理器。

如果控制位  $\overline{T1SY}$  (T1CTL2)清 0，则 T1 工作在同步计数模式。在内部相位时钟的 Q2 和 Q4 周期对外部时钟电平进行采样，可以实现外部时钟与内部相位时钟的同步。

### 8.2.6 重载模式

置位 PWMCTL 寄存器的 PWM16MODE 位使能 T1 重载功能，只要不使能 PWM16ON 则不会输出 PWM16 波形。然后通过设置 PP2/PP1 寄存器来设置 T1 的重载点。

当设置完 PP2/PP1 寄存器且使能重载功能后，T1 计数器从 0 开始计数，当计数到 PP2/PP1 寄存器设置的值时，T1 计数器清 0 并重新开始计数，T1IF 置 1。

对 PP1 和 PP2 的修改会立即生效。

**注：**1. T1 使能重载前应该配置好 PP2/PP1，在 T1 计数中使能重载可能会直接进入中断。

2. 在重载模式下，程序判断 T1IF 为 1 后，立即执行清零 T1ON 操作可能导致 T1IF 无法清零的情况；可在清零 T1ON 清零操作指令之前，添加两条空指令（NOP）对该情况进行规避。

### 8.2.7 T1 在休眠模式下的运行

当 T1 的工作时钟选择 TICK 或者低频外设时钟 LFCLK 时，T1 可以工作在休眠模式下。

T1 休眠下可选时钟：

- TICK
- LFCLK (INTLF、EXTLF)

如使能 T1 中断，T1 将在溢出或者发生重载时将单片机从休眠模式下唤醒。

T1 无法在关断模式下工作。

### 8.3 定时器 T2/T4

CCP2 模块包含 2 个计数器 T2/T4（原理一样，当 T4 选择 LFCLK 时，通过使能 SLPEN 位，可在休眠模式下继续工作），他们是 16 位的定时器，分别由 2 个 8 位寄存器组成。他们有 2 种计数模式：向上计数和向上-向下计数模式。并且没有外部计数时钟输入脚，其原理框如图 8.3 和图 8.4 所示。

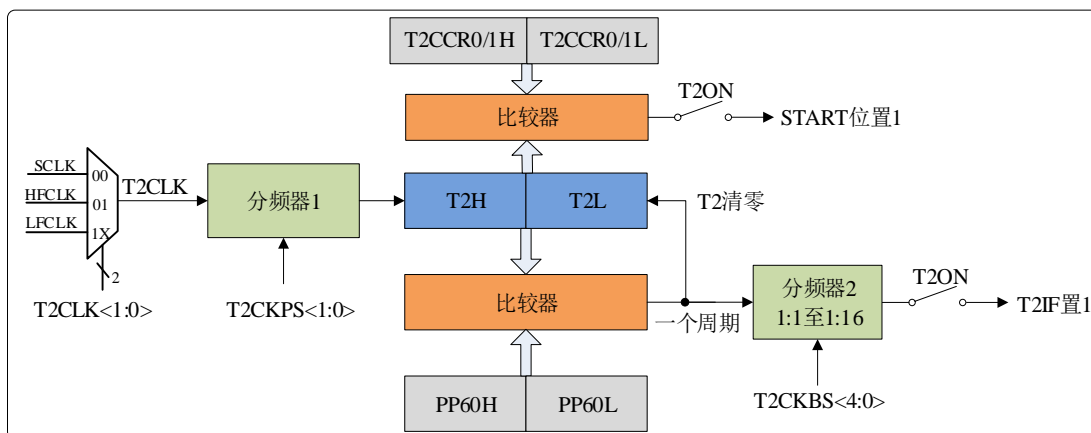


图 8.3 定时器 T2 原理框图

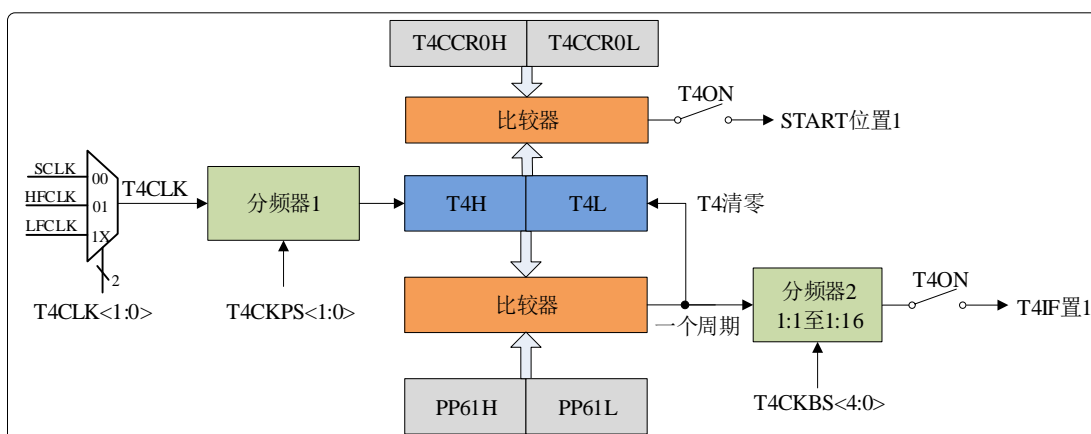


图 8.4 定时器 T4 原理框图

- 注：（1） T2 有 2 组寄存器触发 AD:T2CCR0H/L、T2CCR1H/L；  
T4 有 1 组寄存器触发 AD:T4CCR0H/L；  
（2） T2 与 PP60 进行匹配，T4 与 PP61 进行匹配，用户切忌混淆。

## 8.3.1 T2/T4 相关的寄存器

表 8.3 与 T2/T4 相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
41H	T2H	T2 高 8 位寄存器							
11H	T2L	T2 低 8 位寄存器							
52H	PP60L	PP6 周期 0 低 8 位寄存器							
42H	PP60H	PP6 周期 0 高 8 位寄存器							
12H	T2CTL0	-	T2CKBS3	T2CKBS2	T2CKBS1	T2CKBS0	T2ON	T2CKPS1	T2CKPS0
51H	T2CTL1	T2MOD1	T2MOD0	T2OFS1	T2OFS0	T2TRIGEN	T2CLK1	T2CLK0	T2DIR
141H	T4H	T4 高 8 位寄存器							
111H	T4L	T4 低 8 位寄存器							
142H	PP61H	PP6 周期 1 寄存器高 8 位							
152H	PP61L	PP6 周期 1 寄存器低 8 位							
112H	T4CTL0	T4SLPEN	T4CKBS3	T4CKBS2	T4CKBS1	T4CKBS0	T4ON	T4CKPS1	T4CKPS0
151H	T4CTL1	T4MOD1	T4MOD0	T4OFS1	T4OFS0	T4TRIGEN	T4CLK1	T4CLK0	T4DIR
155H	T2CCR0H	T2 启动 ADC 设置寄存器 0 高 8 位							
154H	T2CCR0L	T2 启动 ADC 设置寄存器 0 低 8 位							
157H	T2CCR1H	T2 启动 ADC 设置寄存器 1 高 8 位							
156H	T2CCR1L	T2 启动 ADC 设置寄存器 1 低 8 位							
15AH	T4CCR0H	T4 启动 ADC 设置寄存器 0 高 8 位							
158H	T4CCR0L	T4 启动 ADC 设置寄存器 0 低 8 位							

### 8.3.1.1 T2 控制寄存器 0(T2CTL0)

**寄存器:** T2CTL0: T2控制寄存器0(地址: 12H)

	bit7							bit0
复位值 -000 0000	-	T2CKBS3	T2CKBS2	T2CKBS1	T2CKBS0	T2ON	T2CKPS1	T2CKPS0
	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

T2CKBS<3:0>:T2 分频器 2 分频比选择位

- 0000: 分频比为 1:1
- 0001: 分频比为 1:2
- 0010: 分频比为 1:3
- 0011: 分频比为 1:4
- 0100: 分频比为 1:5
- 0101: 分频比为 1:6
- 0110: 分频比为 1:7
- 0111: 分频比为 1:8
- 1000: 分频比为 1:9
- 1001: 分频比为 1:10
- 1010: 分频比为 1:11
- 1011: 分频比为 1:12



- 1100: 分频比为 1:13  
 1101: 分频比为 1:14  
 1110: 分频比为 1:15  
 1111: 分频比为 1:16
- T2ON: T2 使能位  
 1 = 使能 T2  
 0 = 禁止 T2
- T2CKPS<1:0>:T2 分频器 1 分频比选择位  
 00=分频比为 1:1  
 01=分频比为 1:4  
 1x=分频比为 1:16

### 8.3.1.2 T4 控制寄存器 0(T4CTL0)

寄存器: T4CTL0: T4控制寄存器0(地址: 112H)

		bit7				bit0			
复位值	0000 0000	T4SLPEN	T4CKBS3	T4CKBS2	T4CKBS1	T4CKBS0	T4ON	T4CKPS1	T4CKPS0
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- T4SLPEN: T4 休眠使能位  
 1 = 允许 T4 在休眠模式下继续工作  
 0 = 禁止 T4 在休眠模式下继续工作

T4CKBS<3:0>:T4 分频器 2 分频比选择位

- 0000: 分频比为 1:1  
 0001: 分频比为 1:2  
 0010: 分频比为 1:3  
 0011: 分频比为 1:4  
 0100: 分频比为 1:5  
 0101: 分频比为 1:6  
 0110: 分频比为 1:7  
 0111: 分频比为 1:8  
 1000: 分频比为 1:9  
 1001: 分频比为 1:10  
 1010: 分频比为 1:11  
 1011: 分频比为 1:12  
 1100: 分频比为 1:13  
 1101: 分频比为 1:14  
 1110: 分频比为 1:15  
 1111: 分频比为 1:16

- T4ON: T4 使能位  
 1 = 使能 T4  
 0 = 禁止 T4

T4CKPS<1:0>:T4 分频器 1 分频比选择位  
 00 = 分频比为 1:1

01 = 分频比为 1:4

1x = 分频比为 1:16

### 8.3.1.3 T2 控制寄存器 1(T2CTL1)

**寄存器:** T2CTL1: T2控制寄存器1(地址:51H)

bit7							bit0	
复位值 0000 0000	T2MOD1	T2MOD0	T2OFS1	T2OFS0	T2TRIGEN	T2CLK1	T2CLK0	T2DIR
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

T2MOD<1:0>:T2 计数模式选择位。

00 = 向上计数, 上溢时产生中断标志 T2IF

01 = 向上-向下计数, 上溢时产生中断标志 T2IF

10 = 向上-向下计数, 下溢时产生中断标志 T2IF

11 = 向上-向下计数, 上溢和下溢时都产生中断标志 T2IF

T2OFS1: T2 溢出中断选择, T2TRIGEN 为 1 时有效

0 = 禁止上溢中断触发 AD

1 = 使能上溢中断触发 AD

T2OFS0: T2 溢出中断选择, T2TRIGEN 为 1 时有效

0 = 禁止下溢中断触发 AD

1 = 使能下溢中断触发 AD

T2TRIGEN: T2 溢出中断自动触发 AD 控制位, 当 AD 转换结束后自动清零。

0 = 禁止 T2 溢出中断自动触发 AD

1 = 使能 T2 溢出中断自动触发 AD

T2CLK<1:0>:T2 工作时钟选择位

00 = 选用外部引脚 T2CKIN 为 T2 工作时钟

01 = 选用 SCLK 时钟为 T2 工作时钟

10 = 选用 HFCLK 时钟为 T2 工作时钟

11 = 选用 LFCLK 时钟为 T2 工作时钟

T2DIR: T2 计数方向标志位 (只读)

0 = 当前计数方向为向下计数

1 = 当前计数方向为向上计数

### 8.3.1.4 T4 控制寄存器 1(T4CTL1)

**寄存器:** T4CTL1: T4控制寄存器1(地址:151H)

bit7							bit0	
复位值 0000 0000	T4MOD1	T4MOD0	T4OFS1	T4OFS0	T4TRIGEN	T4CLK1	T4CLK0	T4DIR
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

T4MOD<1:0>:T4 计数模式选择位。

00 = 向上计数, 上溢时产生中断标志

01 = 向上-向下计数, 上溢时产生中断标志

- 10 = 向上-向下计数，下溢时产生中断标志  
 11 = 向上-向下计数，上溢和下溢时都产生中断标志
- T4OFS1: T4 溢出中断选择，T4TRIGEN 为 1 时有效  
 0 = 禁止上溢中断触发 AD  
 1 = 使能上溢中断触发 AD
- T4OFS0: T4 溢出中断选择，T4TRIGEN 为 1 时有效  
 0 = 禁止下溢中断触发 AD  
 1 = 使能下溢中断触发 AD
- T4TRIGEN: T4 溢出中断自动触发 AD 控制位，当 AD 转换结束后自动清零。  
 0 = 禁止 T4 溢出中断自动触发 AD  
 1 = 使能 T4 溢出中断自动触发 AD
- T4CLK<1:0>: T4 工作时钟选择位  
 00 = 选用外部引脚 T4CKIN 为 T4 工作时钟  
 01 = 选用 SCLK 时钟为 T4 工作时钟  
 10 = 选用 HFCLK 时钟为 T4 工作时钟  
 11 = 选用 LFCLK 时钟为 T4 工作时钟
- T4DIR: T4 计数方向标志位（只读）  
 0 = 当前计数方向为向下计数  
 1 = 当前计数方向为向上计数

### 8.3.2 T2/T4 的工作原理

通过 TxCTL0 (x=2,4) 的 TxON (x=2,4) 启动/禁止 Tx (x=2,4)。Tx (x=2,4) 的原理框图如图 8.3 所示，Tx (x=2,4) 计数时钟 TxCLK (x=2,4) 为 SCLK、HFCLK 和 LFCLK 三种时钟可选，每个计数周期寄存器 Tx (x=2,4) 的值自动加 1。

定时器 Tx (x=2,4) 可以通过 PPyH/L (y=60,61) 来设置周期。周期设置可以应用在 PWM6 的产生。

Tx (x=2,4) 有两个分频器，预分频器（分频器 1）和后分频器（分频器 2）。预分频器用于产生定时器的时钟，后分频器用于控制中断信号产生的时机。

定时器 Tx 有 2 种计数模式，向上计数和向上-向下计数，通过 TxDIR (x=2,4) 的计数方向标志位可以知道当前的计数方向。

当使用 Tx (x=2,4)，可以使用自动触发 AD 的功能。AD 触发功能有两种形式，一种是通过 Tx (x=2,4) 溢出信号触发，一种是通过 TxCCRzH/L (x=2,4;z=0,1) 寄存器与 Tx (x=2,4) 匹配时产生。

分频器 2 的分频比可通过寄存器 TxCTL0 (x=2,4) 中的 TxCKBS<3:0> (x=2,4) 位设置，分频器 2 可以控制 Tx (x=2,4) 中断产生的时机。

### 8.3.3 T2 计数模式

通过控制 TxCTL1 (x=2,4) 的 TxMOD<1:0> (x=2,4) 位可以设置计数器计数模式。T2 可实现两种计数模式，分别是向上计数和向上-向下计数，向上计数模式实现边沿对称 PWM 信号。通过 TxDIR (x=2,4) 的计数方向标志位可以知道当前的计数方向。

在向上计数模式时，当 TxH/L (x=2,4) 寄存器与 PPyH/L (y=60, 61) 相等时，Tx

(x=2,4) 自动清 0，发出上溢信号给分频器 2，分频器 2 递增。向上计数周期为 <PPyH:PPyL>+1 (y=60, 61)。

在向上-向下计数模式时，当 TxH/L (x=2,4) 寄存器与 <PPyH:PPyL>+1 (y=60,61)。寄存器相等时，Tx (x=2,4) 不清零，而是自动从当前值开始向下计数，直到计数值为 0，周期寄存器 <PPyH:PPyL> (y=60,61) 的值为计数周期的一半。这种模式通过 TxMOD<1:0> (x=2,4) 位控制产生中断溢出信号给分频器 2，并进行加一计数。

### 8.3.4 T2/T4 分频器

Tx (x=2,4) 有两个分频器，预分频器（分频器 1）和后分频器（分频器 2）。预分频器用于产生定时器的时钟，后分频器用于控制中断信号产生的时机。

Tx (x=2,4) 计数时钟 TxCLK (x=2,4) 先经过分频器 1 分频后送到 Tx (x=2,4) 寄存器，进行加一计数，分频器 1 通过设置寄存器 TxCTL0 (x=2,4) 中的 TxCKPS<1:0> (x=2,4) 位，可将 TxCLK (x=2,4) 进行 1:1、1:4 和 1:16 分频。

分频器 2 的分频比可通过寄存器 TxCTL0 (x=2,4) 中的 TxCKBS<3:0> (x=2,4) 位设置为 1:1 至 1:16。分频器 2 可以控制 Tx (x=2,4) 中断产生的时机。

Tx (x=2,4) 的预分频器 1 的预分频比由寄存器 TxCTL0 (x=2,4) 的 TxCKPS<1:0> (x=2,4) 位设置。分频器 2 由寄存器 TxCTL0 (x=2,4) 的 TxCKBS<1:0> (x=2,4) 位设置。如果对 Tx (x=2,4) 寄存器执行写操作、对寄存器 TxCTL0 (x=2,4) 执行写操作或发生任何器件复位事件(上电复位、 $\overline{\text{RST}}$  复位、看门狗复位或欠压复位)，分频器 1/2 的计数器将被清 0。

### 8.3.5 T2/T4 中断

如图 8.3 所示，Tx (x=2,4) 预分频器 2 的计数器达到设定值后将会使中断标志位置 1，如果使能位 TxIE (x=2,4) 为 1，且全局中断和外设中断允许位为 1，将会相应 Tx (x=2,4) 中断。如果 IPEN 和 PTx (x=2,4) 位均置 1，则 Tx (x=2,4) 为高优先级中断。

分频器 2 的分频比可通过寄存器 TxCTL0 (x=2,4) 中的 TxCKBS<3:0> (x=2,4) 位设置为 1:1 至 1:16。当其设置为 1:1 时，每次寄存器 TxH/L (x=2,4) 与 PPyH/L (y=60,61) 相等自动清 0 时（向上模式）或者向下计数到 0（向上向下模式），将会使 Tx (x=2,4) 中断标志位 TxIF (x=2,4) 置 1；

当其设置为 1:2 时，寄存器 TxH/L (x=2,4) 与 PPyH/L (y=60,61) 相等者或者向下计数到 0 累计两次才会使 TxIF (x=2,4) 置 1，以此类推。

### 8.3.6 T2/T4 在休眠模式

单片机在休眠模式下：

- T2 停止工作，无法工作在休眠模式
- T4 可在休眠模式下继续工作，须进行如下配置：
  - 将 T4 工作时钟配置为低频外设时钟，即将 T4CTL1 寄存器的 T4CLK1 位置 1；
  - 将 T4CTL0 寄存器的 T4SLPEN 位置 1 以使能休眠下工作。

### 8.3.7 T2/T4 分配给 PWM6

PWM6 在使用时，需要使用 T2/T4 作为时基，其中通道 0/1/2 使用 T2 作为计数时基，通道 3 可选择 T2 或 T4 作为计数时基，详细请参照 PWM6 部分。

### 8.3.8 T2/4 触发 AD

当使用 Tx (x=2,4)，可以使用自动触发 AD 的功能。AD 触发功能有两种形式，一种是通过 Tx (x=2,4) 溢出信号触发，一种是通过 TxCCRzH/L (x=2,4;z=0,1) 寄存器与 Tx (x=2,4) 匹配时产生，这种方法可在周期中的任意位置触发 AD。

以 T2 触发 AD 为例，设置寄存器 T2CTL1，选择合适的计数模式，

将 T2OFS<1:0>选择合适的中断触发时基，同时使能 T2TRIGEN 信号，选择自动触发，可实现 T2 溢出信号自动触发 AD 转换，使 ADSTART 位置 1。T2TRIGEN 信号在完成一次 AD 转换后自动清零。

还可以设置 T2CCRxH/L (x=0,1) 寄存器，当 T2CCRxH/L 与 T2 匹配时，将产生 AD 触发信号将 ADSTART 位置 1 (相应的 T2CCRxON 使能，x=0,1)。添加该项：T2CCRxON 信号在完成一次 AD 转换后也会自动清零

T4 触发 AD 与 T2 相似，不用的是，T4 只有一组任意位置触发 AD 的控制 T4CCR0H/L，T4CCR0H/L 的使能为 T4CCR0ON。

注：T2CCR0ON 位在 ADCCTL0 的第 6 位，T2CCR1ON 位在 ADCINTCTL 的第 7 位，T4CCR0ON 位在 ADCINTCTL 的第 6 位，详见[模数\(A/D\)转换模块](#)。

### 8.3.9 T2/4 读写操作

寄存器 TxH/L (x=2,4)、PPyH/L (y=60,61) 均可读写。任何复位时，寄存器 TxH/L (x=2,4) 均被设置为 00H，PPyH/L (y=60,61) 被设置为 FFFFH。

注意：读 Tx (x=2,4) 时，先读 Tx (x=2,4) 的高八位，后读 Tx (x=2,4) 的低八位 (当执行读 Tx (x=2,4) 高八位指令时，Tx (x=2,4) 的低八位数据存入缓存器中，接着执行读 Tx (x=2,4) 低八位的操作时就把存入缓存器中的数据读出)。若先读低八位，读出的数据是错误的。

## 8.4 定时器 T5/T6

CCP5 模块包含 2 个计数器 T5/T6（原理一样，当 T6 选择 LFCLK 时，通过使能 SLPEN 位，可在休眠模式下继续工作），他们是 16 位的定时器，分别由 2 个 8 位寄存器组成。他们有 2 种计数模式：向上计数和向上-向下计数模式。并且没有外部计数时钟输入脚，其原理框图如图 8.5 和图 8.6 所示。

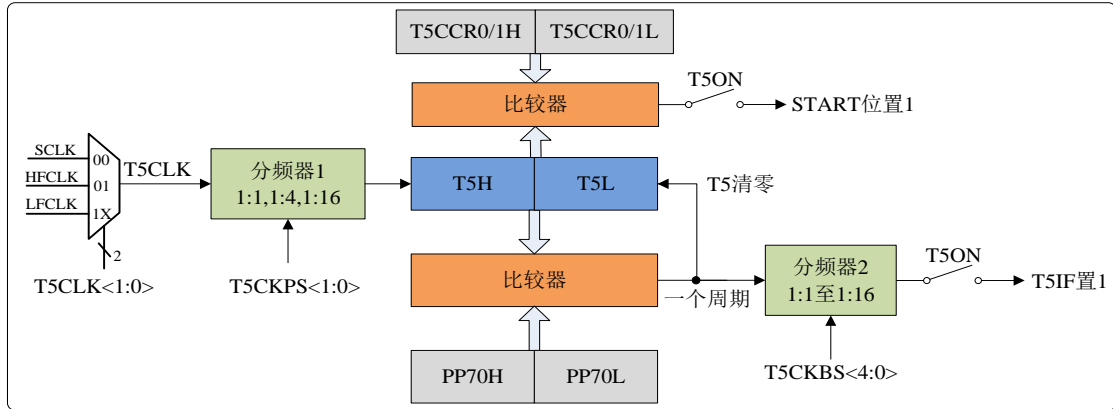


图 8.5 定时器 T5 原理框图

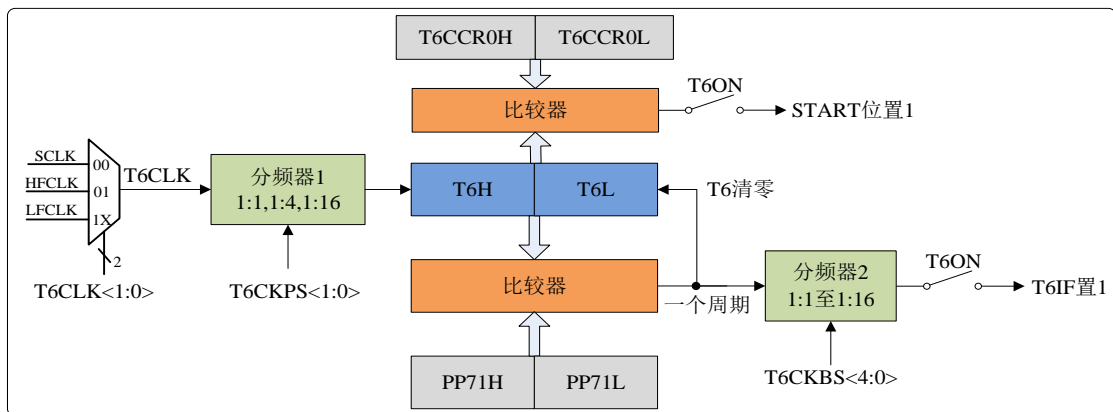


图 8.6 定时器 T6 原理框图

- 注：（1） T5 有 2 组寄存器触发 AD:T5CCR0H/L、T5CCR1H/L；  
T4 有 1 组寄存器触发 AD:T6CCR0H/L；  
（2） T5 与 PP70 进行匹配，T6 与 PP71 进行匹配，用户切忌混淆。

## 8.4.1 T5/T6 相关的寄存器

表 8.4 与 T5/T6 相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
551H	T5H	T5 高 8 位寄存器							
552H	T5L	T5 低 8 位寄存器							
50DH	PP70L	PWM7 周期 0 低 8 位寄存器							
50CH	PP70H	PWM7 周期 0 高 8 位寄存器							
550H	T5CTL0	-	T5CKBS3	T5CKBS2	T5CKBS1	T5CKBS0	T5ON	T5CKPS1	T5CKPS0
515H	T5CTL1	T5MOD1	T5MOD0	T5OFS1	T5OFS0	T5TRIGEN	T5CLK1	T5CLK0	T5DIR
555H	T6H	T6 高 8 位寄存器							
556H	T6L	T6 低 8 位寄存器							
50EH	PP71H	PWM7 周期 1 寄存器高 8 位							
50FH	PP71L	PWM7 周期 1 寄存器低 8 位							
554H	T6CTL0	T6SLPEN	T6CKBS3	T6CKBS2	T6CKBS1	T6CKBS0	T6ON	T6CKPS1	T6CKPS0
553H	T6CTL1	T6MOD1	T6MOD0	T6OFS1	T6OFS0	T6TRIGEN	T6CLK1	T6CLK0	T6DIR
563H	T5CCR0H	T5 启动 ADC 设置寄存器 0 高 8 位							
564H	T5CCR0L	T5 启动 ADC 设置寄存器 0 低 8 位							
565H	T5CCR1H	T5 启动 ADC 设置寄存器 1 高 8 位							
566H	T5CCR1L	T5 启动 ADC 设置寄存器 1 低 8 位							
567H	T6CCR0H	T6 启动 ADC 设置寄存器 0 高 8 位							
568H	T6CCR0L	T6 启动 ADC 设置寄存器 0 低 8 位							

### 8.4.1.1 T5 控制寄存器 0(T5CTL0)

**寄存器:** T5CTL0: T5控制寄存器0(地址: 550H)

	bit7							bit0
复位值	-	T5CKBS3	T5CKBS2	T5CKBS1	T5CKBS0	T5ON	T5CKPS1	T5CKPS0
-000 0000	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

T5CKBS<3:0>:T5 分频器 2 分频比选择位

- 0000: 分频比为 1:1
- 0001: 分频比为 1:2
- 0010: 分频比为 1:3
- 0011: 分频比为 1:4
- 0100: 分频比为 1:5
- 0101: 分频比为 1:6
- 0110: 分频比为 1:7
- 0111: 分频比为 1:8
- 1000: 分频比为 1:9
- 1001: 分频比为 1:10
- 1010: 分频比为 1:11
- 1011: 分频比为 1:12

- 1100: 分频比为 1:13  
 1101: 分频比为 1:14  
 1110: 分频比为 1:15  
 1111: 分频比为 1:16
- T5ON: T5 使能位  
 1 = 使能 T5  
 0 = 禁止 T5
- T5CKPS<1:0>:T5 分频器 1 分频比选择位  
 00=分频比为 1:1  
 01=分频比为 1:4  
 1x=分频比为 1:16

#### 8.4.1.2 T6 控制寄存器 0(T6CTL0)

寄存器: T6CTL0: T6控制寄存器0(地址: 554H)

复位值	bit7						bit0	
0000 0000	T6SLPEN	T6CKBS3	T6CKBS2	T6CKBS1	T6CKBS0	T6ON	T6CKPS1	T6CKPS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- T6SLPEN: T6 休眠使能位  
 1 = 允许 T6 在休眠模式下继续工作  
 0 = 禁止 T6 在休眠模式下继续工作
- T6CKBS<3:0>:T6 分频器 2 分频比选择位  
 0000: 分频比为 1:1  
 0001: 分频比为 1:2  
 0010: 分频比为 1:3  
 0011: 分频比为 1:4  
 0100: 分频比为 1:5  
 0101: 分频比为 1:6  
 0110: 分频比为 1:7  
 0111: 分频比为 1:8  
 1000: 分频比为 1:9  
 1001: 分频比为 1:10  
 1010: 分频比为 1:11  
 1011: 分频比为 1:12  
 1100: 分频比为 1:13  
 1101: 分频比为 1:14  
 1110: 分频比为 1:15  
 1111: 分频比为 1:16
- T6ON: T6 使能位  
 1 = 使能 T6  
 0 = 禁止 T6
- T6CKPS<1:0>:T6 分频器 1 分频比选择位  
 00 = 分频比为 1:1



01 = 分频比为 1:4

1x = 分频比为 1:16

### 8.4.1.3 T5 控制寄存器 1(T5CTL1)

寄存器: T5CTL1: T5控制寄存器1(地址:515H)

	bit7						bit0	
复位值 0000 0000	T5MOD1	T5MOD0	T5OFS1	T5OFS0	T5TRIGEN	T5CLK1	T5CLK0	T5DIR
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

T5MOD<1:0>:T5 计数模式选择位。

00 = 向上计数, 上溢时产生中断标志

01 = 向上-向下计数, 上溢时产生中断标志

10 = 向上-向下计数, 下溢时产生中断标志

11 = 向上-向下计数, 上溢和下溢时都产生中断标志

T5OFS1: T5 溢出中断选择, T5TRIGEN 为 1 时有效

0 = 禁止上溢中断触发 AD

1 = 使能上溢中断触发 AD

T5OFS0: T5 溢出中断选择, T5TRIGEN 为 1 时有效

0 = 禁止下溢中断触发 AD

1 = 使能下溢中断触发 AD

T5TRIGEN: T5 溢出中断自动触发 AD 控制位, 当 AD 转换结束后自动清零。

0 = 禁止 T5 溢出中断自动触发 AD

1 = 使能 T5 溢出中断自动触发 AD

T5CLK<1:0>:T5 工作时钟选择位

00 = 选用外部引脚 T5CKIN 为 T5 工作时钟

01 = 选用 SCLK 时钟为 T5 工作时钟

10 = 选用 HFCLK 时钟为 T5 工作时钟

11 = 选用 LFCLK 时钟为 T5 工作时钟

T5DIR: T5 计数方向标志位 (只读)

0 = 当前计数方向为向下计数

1 = 当前计数方向为向上计数

### 8.4.1.4 T6 控制寄存器 1(T6CTL1)

寄存器: T6CTL1: T6控制寄存器1(地址:553H)

	bit7						bit0	
复位值 0000 0000	T6MOD1	T6MOD0	T6OFS1	T6OFS0	T6TRIGEN	T6CLK1	T6CLK0	T6DIR
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

T6MOD<1:0>:T6 计数模式选择位。

00 = 向上计数, 上溢时产生中断标志

01 = 向上-向下计数, 上溢时产生中断标志

	10 = 向上-向下计数，下溢时产生中断标志
	11 = 向上-向下计数，上溢和下溢时都产生中断标志
T6OFS1:	T6 溢出中断选择，T6TRIGEN 为 1 时有效
	0 = 禁止上溢中断触发 AD
	1 = 使能上溢中断触发 AD
T6OFS0:	T6 溢出中断选择，T6TRIGEN 为 1 时有效
	0 = 禁止下溢中断触发 AD
	1 = 使能下溢中断触发 AD
T6TRIGEN:	T6 溢出中断自动触发 AD 控制位，当 AD 转换结束后自动清零。
	0 = 禁止 T6 溢出中断自动触发 AD
	1 = 使能 T6 溢出中断自动触发 AD
T6CLK<1:0>:	T6 工作时钟选择位
	00 = 选用外部引脚 T6CKIN 为 T6 工作时钟
	01 = 选用 SCLK 时钟为 T6 工作时钟
	10 = 选用 HFCLK 时钟为 T6 工作时钟
	11 = 选用 LFCLK 时钟为 T6 工作时钟
T6DIR:	T6 计数方向标志位（只读）
	0 = 当前计数方向为向下计数
	1 = 当前计数方向为向上计数

### 8.4.2 T5/T6 的工作原理

通过 TxCTL0 (x=5,6) 的 TxON (x=5,6) 启动/禁止 Tx (x=5,6)。Tx (x=5,6) 的原理框图如图 8.5 所示，Tx (x=5,6) 计数时钟 TxCLK (x=5,6) 为 SCLK、HFCLK 和 LFCLK 三种时钟可选，每个计数周期寄存器 Tx (x=5,6) 的值自动加 1。

定时器 Tx (x=5,6) 可以通过 PPyH/L (y=70,71) 来设置周期。周期设置可以应用在 PWM6 的产生。

Tx (x=5,6) 有两个分频器，预分频器（分频器 1）和后分频器（分频器 2）。预分频器用于产生定时器的时钟，后分频器用于控制中断信号产生的时机。

定时器 Tx 有 2 种计数模式，向上计数和向上-向下计数，通过 TxDIR (x=5,6) 的计数方向标志位可以知道当前的计数方向。

当使用 Tx (x=5,6)，可以使用自动触发 AD 的功能。AD 触发功能有两种形式，一种是通过 Tx (x=5,6) 溢出信号触发，一种是通过 TxCCRzH/L (x=5,6;z=0,1) 寄存器与 Tx (x=5,6) 匹配时产生。

分频器 2 的分频比可通过寄存器 TxCTL0 (x=5,6) 中的 TxCKBS<3:0> (x=5,6) 位设置，分频器 2 可以控制 Tx (x=5,6) 中断产生的时机。

### 8.4.3 T5 计数模式

通过控制 TxCTL1 的 TxMOD<1:0>位可以设置计数器计数模式。Tx 可实现两种计数模式，分别是向上计数和向上-向下计数，向上计数实现边沿对称 PWM 信号。通过 TxDIR 的计数方向标志位可以知道当前的计数方向。

在向上计数模式时，当 TxH/L (x=5,6) 寄存器与 PPyH/L (y=70,71) 相等时，Tx

(x=5,6) 自动清 0，发出上溢信号给分频器 2，分频器 2 递增。向上计数周期为  $\langle \text{PPyH:PPyL} \rangle + 1$  (y=70,71)。

在向上-向下计数模式时，当 TxH/L (x=5,6) 寄存器与 PPyH/L (y=70,71) 寄存器相等时，Tx (x=5,6) 不清零，而是自动从当前值开始向下计数，直到计数值为 0，周期寄存器  $\langle \text{PPyH:PPyL} \rangle + 1$  (y=70,71) 的值为计数周期的一半。这种模式通过 TxMOD<1:0>

(x=5,6) 位控制产生中断溢出信号给分频器 2，并进行加一计数。

#### 8.4.4 T5/T6 分频器

Tx (x=5,6) 有两个分频器，预分频器（分频器 1）和后分频器（分频器 2）。预分频器用于产生定时器的时钟，后分频器用于控制中断信号产生的时机。

Tx (x=5,6) 计数时钟 TxCLK (x=5,6) 先经过分频器 1 分频后送到 Tx (x=5,6) 寄存器，进行加一计数，分频器 1 通过设置寄存器 TxCTL0 (x=5,6) 中的 TxCKPS<1:0>

(x=5,6) 位，可将 TxCLK (x=5,6) 进行 1:1、1:4 和 1:16 分频。

分频器 2 的分频比可通过寄存器 TxCTL0 (x=5,6) 中的 TxCKBS<3:0> (x=5,6) 位设置为 1:1 至 1:16。分频器 2 可以控制 Tx (x=5,6) 中断产生的时机。

Tx (x=5,6) 的预分频器 1 的预分频比由寄存器 TxCTL0 (x=5,6) 的 TxCKPS<1:0> (x=5,6) 位设置。分频器 2 由寄存器 TxCTL0 (x=5,6) 的 TxCKBS<1:0> (x=5,6) 位设置。如果对 Tx (x=5,6) 寄存器执行写操作、对寄存器 TxCTL0 (x=5,6) 执行写操作或发生任何器件复位事件(上电复位、 $\overline{\text{RST}}$  复位、看门狗复位或欠压复位)，分频器 1/2 的计数器将被清 0。

#### 8.4.5 T5/T6 中断

如图 8.5 所示，Tx (x=5,6) 预分频器 2 的计数器达到设定值后将会使中断标志位置 1，如果使能位 TxIE (x=5,6) 为 1，且全局中断和外设中断允许位为 1，将会相应 Tx (x=5,6) 中断。如果 IPEN 和 PTx (x=5,6) 位均置 1，则 Tx (x=5,6) 为高优先级中断。

分频器 2 的分频比可通过寄存器 TxCTL0 (x=5,6) 中的 TxCKBS<3:0> (x=5,6) 位设置为 1:1 至 1:16。当其设置为 1:1 时，每次寄存器 TxH/L (x=5,6) 与 PPyH/L (y=70,71) 相等（向上模式）或者向下计数到 0（向上向下模式），将会使 Tx (x=5,6) 中断标志位 TxIF (x=5,6) 置 1；当其设置为 1:2 时，寄存器 TxH/L (x=5,6) 与 PPyH/L (y=70,71) 相等者或者向下计数到 0 累计两次才会使 TxIF (x=5,6) 置 1，以此类推。

#### 8.4.6 T5/T6 在休眠模式

单片机在休眠模式下：

- T5 停止工作，无法工作在休眠模式
- T6 可在休眠模式下继续工作，须进行如下配置：
  - 将 T6 工作时钟配置为低频外设时钟，即将 T6CTL1 寄存器的 T6CLK1 位置 1；
  - 将 T6CTL0 寄存器的 T6SLPEN 位置 1 以使能休眠下工作。

### 8.4.7 T5/T6 分配给 PWM7

PWM7 在使用时，需要使用 T5/T6 作为时基，详细请参照 PWM7 部分。

### 8.4.8 T5/T6 触发 AD

当使用 Tx (x=5,6)，可以使用自动触发 AD 的功能。AD 触发功能有两种形式，一种是通过 Tx (x=5,6) 溢出信号触发，一种是通过 TxCCRzH/L (x=5,6;z=0,1) 寄存器与 Tx (x=5,6) 匹配时产生，这种方法可在周期中的任意位置触发 AD。

以 T5 触发 AD 为例，设置寄存器 T5CTL1，选择合适的计数模式，将 T5OFS<1:0>选择合适的中断触发时基，同时使能 T5TRIGEN 信号，选择自动触发，可实现 T5 溢出信号自动触发 AD 转换，使 ADSTART 位置 1。T5TRIGEN 信号在完成一次 AD 转换后自动清零。还可以设置 T5CCRxH/L (x=0,1) 寄存器，当 T5CCRxH/L 与 T5 匹配时，将产生 AD 触发信号将 ADSTART 位置 1 (相应的 T5CCRxON 使能，x=0,1)。

T6 触发 AD 与 T5 相似，不同的是，T6 只有一组任意位置触发 AD 的控制 T6CCR0H/L，T6CCR0H/L 的使能为 T6CCR0ON。

注：T5CCR0ON 位在 ADCCTL0 的第 6 位，T5CCR1ON 位在 ADCINTCTL 的第 7 位，T6CCR0ON 位在 ADCINTCTL 的第 6 位，详见[模数\(A/D\)转换模块](#)。

### 8.4.9 T5/T6 读写操作

寄存器 TxH/L (x=5,6)、PPyH/L (y=70,71) 均可读写。任何复位时，寄存器 TxH/L (x=5,6) 均被设置为 00H，PPyH/L (y=70,71) 被设置为 FFFFH。

注意：读 Tx (x=5,6) 时，先读 Tx (x=5,6) 的高八位，后读 Tx (x=5,6) 的低八位 (当执行读 Tx (x=5,6) 高八位指令时，Tx (x=5,6) 的低八位数据存入缓存器中，接着执行读 Tx (x=5,6) 低八位的操作时就把存入缓存器中的数据读出)。若先读低八位，读出的数据是错误的。

## 8.5 定时器/计数器 T3/T7

T3 是一个 16 位的定时器/计数器，T3 的低 8 位在寄存器 T3L 中，高 8 位在寄存器 T3H 中，当 T3 计数值达到 65535 后，T3 的值再加 1 就会产生溢出，将 T3 中断标志位置 1。T3 属于外部单元，因此在使用 T3 中断时，需将 PUIE 位置 1，使能外设中断。T3 带有捕捉比较功能。如图 8.7 所示为 T3 的原理框图。

注：KF8A100 系列单片机的 T3 和 T7 完全一致，本章后续内容以定时器 T3 为例，定时器 T7 同理。

### 8.5.1 T3/T7 原理框图

T3/T7 的原理框图如图 8.7 所示，T3/T7 是一个带有预分频的 16 位定时器/计数器，计数时钟可选择 SCLK、HFCLK 和 LFCLK，当 T3 与内部时钟一起使用时，T3 用作定时器，当 T3 与外部时钟一起使用时，T3 工作在计数器模式。T3/T7 模块还带有四个预分频器选择项，允许对时钟输入进行 1、2、4 或 8 倍分频。T3CKS/T7CKS 位对预分频计数器进行控制，T3/T7 预分频计数器不能直接进行读写操作，可通过写入 T3H 或 T3L/T7H 或 T7L 使预分频计数器清 0。

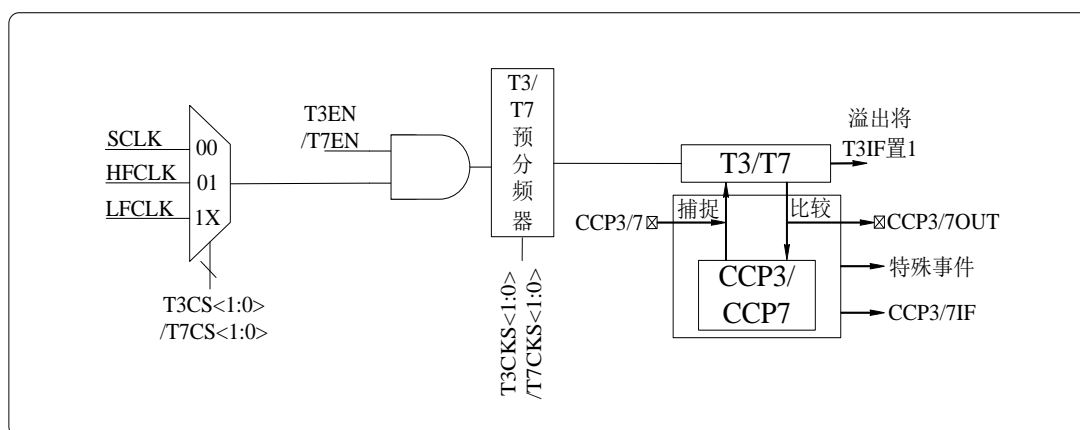


图 8.7 T3/T7 原理框图

### 8.5.2 T3/T7 相关的寄存器

表 8.5 与 T3/T7 相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
119H	T3CTL	T3REN	-	T3CKS1	T3CKS0	-	T3CS1	T3CS0	T3ON
12AH	T7CTL	T7REN	-	T7CKS1	T7CKS0	-	T7CS1	T7CS0	T7ON
11AH	T3H	T3 寄存器高 8 位							
11CH	T3L	T3 寄存器低 8 位							

116H	T3REH	T3 重载寄存器高 8 位							
11BH	T3REL	T3 重载寄存器低 8 位							
12CH	T7H	T7 寄存器高 8 位							
12BH	T7L	T7 寄存器低 8 位							
12EH	T7REH	T7 重载寄存器高 8 位							
12DH	T7REL	T7 重载寄存器低 8 位							
211H	CCP3H	CCP3 寄存器高 8 位							
212H	CCP3L	CCP3 寄存器低 8 位							
214H	CCP7H	CCP7 寄存器高 8 位							
215H	CCP7L	CCP7 寄存器低 8 位							
210H	CCP3CTL	-	-	-	CCP3OE	CCP3MO D3	CCP3MO D2	CCP3MO D1	CCP3MO D0
213H	CCP7CTL	-	-	-	CCP7OE	CCP7MO D3	CCP7MO D2	CCP7MO D1	CCP7MO D0

## 8.5.2.1 T3 控制寄存器

如下寄存器所示，T3 控制寄存器（T3CTL）用于启动/禁止 T3 以及选择 T3 模块的不同功能特性。

寄存器： T3CTL: T3控制寄存器(地址: 119H)

	bit7				bit0			
复位值 0000 0000	T3REN	-	T3CKS1	T3CKS0	-	T3CS1	T3CS0	T3ON
	R/W	U	R/W	R/W	U	R/W	R/W	R/W

T3REN: T3 重载功能使能位

T3REN=0 禁止 T3 重载功能

T3REN=1 使能 T3 重载功能

T3CKS<1:0>: T3 输入时钟预分频比选择位

T3CKS<1:0>=11 1/8 倍预分频比

T3CKS<1:0>=10 1/4 倍预分频比

T3CKS<1:0>=01 1/2 倍预分频比

T3CKS<1:0>=00 1/1 倍预分频比

T3CS<1:0>: T3 定时/计数模式选择

T3CS=00 T3 时钟为系统时钟 SCLK

T3CS=01 T3 时钟为外设高频时钟 HFCLK

T3CS=10 T3 时钟为外设低频时钟 LFCLK

T3ON: T3 启动控制位

T3ON=1 启动 T3

T3ON=0 停止 T3

## 8.5.2.2 T7 控制寄存器

寄存器: T7CTL: T7控制寄存器(地址: 12AH)

	bit7						bit0	
复位值 0-00-000	T7REN	-	T7CKS1	T7CKS0	-	T7CS1	T7CS0	T7ON
	R/W	U	R/W	R/W	U	R/W	R/W	R/W

T7REN: T7 重载功能使能位

T7REN=0 禁止 T7 重载功能

T7REN=1 使能 T7 重载功能

T7CKS&lt;1:0&gt;: T7 输入时钟预分频比选择位

T7CKS&lt;1:0&gt;=11 1/8 倍预分频比

T7CKS&lt;1:0&gt;=10 1/4 倍预分频比

T7CKS&lt;1:0&gt;=01 1/2 倍预分频比

T7CKS&lt;1:0&gt;=00 1/1 倍预分频比

T7CS&lt;1:0&gt;: T7 定时/计数模式选择

T7CS=00 T7 时钟为系统时钟 SCLK

T7CS=01 T7 时钟为外设高频时钟 HFCLK

T7CS=10 T7 时钟为外设低频时钟 LFCLK

T7ON: T7 启动控制位

T7ON=1 启动 T7

T7ON=0 停止 T7

## 8.5.2.3 CCP3 控制寄存器

寄存器: CCP3CTL: CCP3控制寄存器(地址: 210H)

	bit7				bit0			
复位值 ---0 0000	-	-	-	CCP3OE	CCP3MO D3	CCP3MO D2	CCP3MO D1	CCP3MO D0
	U	U	U	R/W	R/W	R/W	R/W	R/W

CCP3OE: CCP3比较模式输出使能位

1 = 比较模式下使能引脚的CCP3输出

0 = 禁止比较模式下引脚的CCP3输出

CCP3MOD&lt;3:0&gt;: CCP3模式选择位

0000 = 捕捉/比较 关闭 (复位CCP3模块)

0001 = 未使用 (保留)

0010 = 比较模式: 匹配时输出电平翻转 (T3IF置1)

0011 = 未使用 (保留)

0100 = 捕捉模式: 在每个下降沿发生捕捉

0101 = 捕捉模式: 在每个上升沿发生捕捉

0110 = 捕捉模式: 每4 个上升沿发生捕捉

0111 = 捕捉模式: 每16 个上升沿发生捕捉

1000 = 比较模式: 比较匹配时输出高电平 (CCP3IF置1)

- 1001 = 比较模式：比较匹配时输出低电平（CCP3IF置1）
- 1010 = 比较模式：比较匹配时产生软件中断（CCP3IF置1，CCP3OUT不受影响）
- 1011 = 比较模式：触发特殊事件（CCP3IF位置1，CCP3复位T3，如果ADC使能将启动AD转换）
- 1100 = 保留
- 1101 = 保留
- 1110 = 保留
- 1111 = 保留

### 8.5.2.4 CCP7 控制寄存器

寄存器: CCP7CTL: CCP7控制寄存器(地址: 213H)

	bit7				bit0			
复位值	-	-	-	CCP7OE	CCP7MO D3	CCP7MO D2	CCP7MO D1	CCP7MO D0
---0 0000	U	U	U	R/W	R/W	R/W	R/W	R/W

CCP7OE: CCP7比较模式输出使能位

- 1 = 比较模式下使能引脚的CCP7输出
- 0 = 禁止比较模式下引脚的CCP7输出

CCP7MOD<3:0>: CCP7模式选择位

- 0000 = 捕捉/比较 关闭（复位CCP7模块）
- 0001 = 未使用（保留）
- 0010 = 比较模式：匹配时输出电平翻转（T7IF置1）
- 0011 = 未使用（保留）
- 0100 = 捕捉模式：在每个下降沿发生捕捉
- 0101 = 捕捉模式：在每个上升沿发生捕捉
- 0110 = 捕捉模式：每4个上升沿发生捕捉
- 0111 = 捕捉模式：每16个上升沿发生捕捉
- 1000 = 比较模式：比较匹配时输出高电平（T7IF置1）
- 1001 = 比较模式：比较匹配时输出低电平（T7IF置1）
- 1010 = 比较模式：比较匹配时产生软件中断（T7IF置1，CCP7OUT不受影响）
- 1011 = 比较模式：触发特殊事件（T7IF位置1，CCP7复位T7，如果ADC使能将启动AD转换）
- 1100 = 保留
- 1101 = 保留
- 1110 = 保留
- 1111 = 保留

### 8.5.3 T3/T7 预分频器

T3/T7 具有四个预分频器选择项，允许对时钟输入进行 1、2、4、或 8 倍分频。T3CKS/T7CKS 位对预分频计数器进行控制。T3/T7 预分频计数器不能直接进行读写操作，



可通过写入 T3H 或 T3L/T7H 或 T7L 使预分频计数器清 0。

#### 8.5.4 T3/T7 计数时钟选择

用户可以通过 T3CTL/T7CTL 寄存器的 T3CS<1:0>/T7CS<1:0>位来选择 T3/T7 的计数时钟：系统时钟 SCLK，外设高频时钟 HFCLK，外设低频时钟 LFCLK。

#### 8.5.5 T3/T7 重载功能

置位 T3CTL/T7CTL 寄存器的 T3REN/T7REN 位使能 T3/T7 重载功能。通过设置 T3REH、T3REL/T7REH、T7REL 寄存器来设置 T3/T7 的重载点，使能重载功能后，T3 计数器从 0 开始计数，当计数到重载点时，T3/T7 计数器清 0 并重新开始计数，T3IF/T7IF 置 1。

T3REN/T7REN 位置 1 时会载入一次 T3REH、T3REL/T7REH、T7REL 寄存器的值；当置位 T3REN/T7REN 且在 T3/T7 计数过程中对其写入新值时，T3/T7 模块会在下一次重载时载入新的重载点。

#### 8.5.6 T3/T7 中断

在两种情况下会使 T3/T7 中断标志位置 1：

1. 当 T3/T7 计数值达到 65535 后，T3 的值再加 1 就会产生溢出，将 T3/T7 中断标志位置 1；
2. 当 T3/T7 发生重载时，T3/T7 中断标志位置 1。

#### 8.5.7 T3/T7 工作在休眠模式

T3/T7 选择低频外设时钟作为工作时钟时，系统进入休眠后，T3/T7 将继续计数；当使能 T3/T7 中断时，系统会在 T3/T7 溢出或者发生重载时唤醒 MCU。

#### 8.5.8 CCP 功能

定时器 T3/T7 带有 CCP 功能：捕捉模式和比较模式；T3 对应 CCP3，T7 对应 CCP7。

##### 8.5.8.1 捕捉模式

在捕捉模式下，当对应的 CCP3/CCP7 引脚发生事件时，寄存器 CCP3H 和 CCP3L/CCP7H 和 CCP7L 捕捉 T3/T7 寄存器的 16 位值，原理如下图所示：

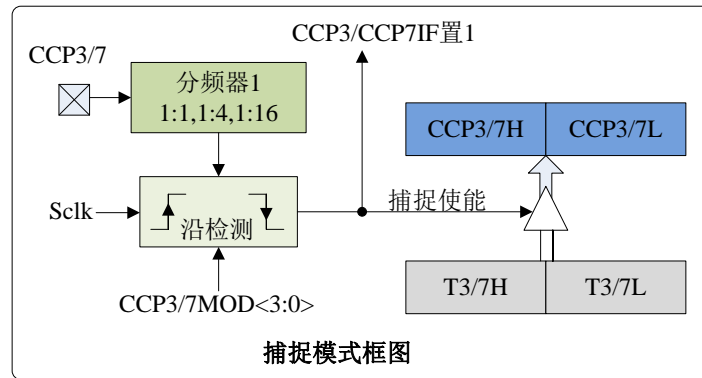


图 8.8 捕捉模式原理框图

触发捕捉的事件可被定义为以下四者之一，并且由 CCP3CTL/CCP6CTL 寄存器中的模式选择位 CCP3MOD<3:0>/ CCP7MOD<3:0>选择事件类型位配置：

- ◆ 0100 = 每个下降沿
- ◆ 0101 = 每个上升沿
- ◆ 0110 = 每4 个上升沿
- ◆ 0111 = 每16个上升沿

在捕捉模式下，应通过将相应的 CCP3/CCP7 引脚配置为数字输入。

当一个捕捉发生时，硬件自动中断请求标志位 CCP3IF/CCP7IF 置1；它必须用软件清零。注意如果在 CCP3H、CCP3L/CCP7H、CCP7L 这对寄存器中的值被读取之前发生另一次捕捉，那么之前捕捉的值将被新捕捉的值覆盖。

当捕捉模式改变时，可能会产生错误的捕捉中断。用户应该在捕捉模式改变之前保持 CCP3IE/CCP7IE 中断允许位清零以避免产生误中断。在捕捉模式发生任何改变之后也应清零中断标志位 CCP3IF/CCP7IF。

CCP3CTL/CCP7CTL 寄存器中的 CCP3/7MOD<3:0>位指定了 4 种预分频器设置。每当关闭 CCP 模块或禁止捕捉模式时，就会清零预分频器计数器。这意味着任何复位都将清零预分频计数器。

虽然从一种捕捉预分频比切换到另一种捕捉预分频比不会将预分频计数器清零，但可能会产生误中断。因此要避免出现这种不期望的操作，应在改变预分频比前通过将 CCP3/7CTL 寄存器清零关闭该模块。

### 8.5.8.2 比较模式

在比较模式下，16 位 CCP3H、CCP3L/CCP7H、CCP7L 寄存器的值将不断与 T3/T7 寄存器的值相比较。当两者匹配时，CCP 模块可能会出现以下几种情况：

- ◆ CCP3/7OUT 引脚的输出电平翻转
- ◆ CCP3/7OUT 引脚输出高电平
- ◆ CCP3/7OUT 引脚输出低电平
- ◆ 产生软件中断触发信号
- ◆ 产生特殊事件触发信号

CCP3/7OUT 引脚的动作取决于 CCP3CTL/CCP7CTL 寄存器中 CCP3/7MOD<3:0> 控制位的值。

所有比较模式都会产生中断。原理图如下图所示：

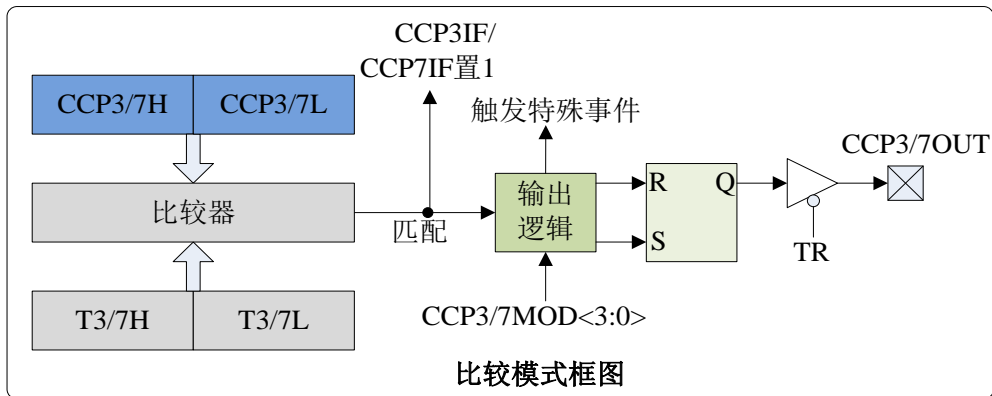


图 8.9 比较模式原理框图

- 1) 当选择输出电平翻转模式时，比较匹配时，CCP3/7OUT 引脚的输出电平翻转，并将 CCP3IF/CCP7IF 置 1。
- 2) 当选择普通比较模式时（1000 或 1001）时，比较匹配时，CCP3/7OUT 引脚输出高电平或者低电平，并且将 CCP3IF/CCP7IF 置 1。
- 3) 当选择了软件中断触发模式时（1010）时，比较匹配时，将 CCP3IF/CCP7IF 置 1，但是 CCP 模块不会控制 CCP3/7OUT 引脚。
- 4) 当选择了特殊事件触发模式（1011）时，比较匹配时，CCP 会立即产生特殊事件触发输出，将 CCP3IF/CCP7IF 置 1，此时如果 ADC 已使能，将启动 AD 转换。但 T3/T7 计数寄存器不会立即复位，直到 T3/T7 计数脉冲的下一个上升沿才复位。从而使 CCP3H、CCP3L/CCP7H、CCP7L 寄存器实际上成为了定时器 T3/T7 的 16 位可编程周期寄存器。

## 8.6 定时器 T8/9

T8 和 T9 是一个带 16 位周期寄存器、预分频器和后分频器 16 位定时器，T8 由 2 个 8 位寄存器 T8H 和 T8L 组成。没有外部计数时钟输入脚。

注：KF8A100 系列单片机的 T8 和 T9 完全一致，本章后续内容以定时器 T8 为例，定时器 T9 同理。

### 8.6.1 T8/T9 的工作原理

如下图所示，为 T8 和 T9 的原理框图，以 T8 为例，T8 的计数时钟源可选 SCLK/HFLCK/LFCLK，每个周期(分频器 1 分频比为 1:1 时)寄存器 T8 的值自动加 1。通过 T8CTL 的 T8ON 启动/禁止 T8。

时钟源先经过分频器 1 分频后送到 T8 寄存器，进行加 1 计数，分频器 1 通过设置寄存器 T8CTL 中的 T8CKPS<1:0>位，可将时钟源进行 1:1、1:4 和 1:16 分频。

当 T8H/L 寄存器与 PP3H/L 相等时，T8 自动清 0，发出相等信号给分频器 2，分频器 2 递增。同时 T8H/L 的值与寄存器 T8CCRH/L 相比较，如果等于 T8CCRH/L，且对应的控制位置 1，则会发出信号，使 ADCATL0 的 START (ADCCTL0<1>) 位置 1。

分频器 2 的分频比可通过寄存器 T8CTL 中的 T8CKBS<3:0>位设置为 1:1 至 1:16。当其设置为 1:1 时，每次寄存器 T8H/L 与 PP3H/L 相等，将会使 T8 中断标志位 T8IF 置 1；

当其设置为 1:2 时，寄存器 T8H/L 与 PP3H/L 相等累计两次才会使 T8IF 置 1，以此类推。

如果对 T8 寄存器执行写操作、对寄存器 T8CTL 执行写操作或发生任何器件复位事件(上电复位、 $\overline{\text{RST}}$  复位、看门狗复位或欠压复位)，分频器 1/2 的计数器将被清 0。

寄存器 T8H/L、PP3 和 T8CCR 均可读写。任何复位时，寄存器 T8H/L 均被设置为 00H，T8CCR<sub>H/L</sub> 被设置为 00H/00H，PP3 被设置为 FFH。

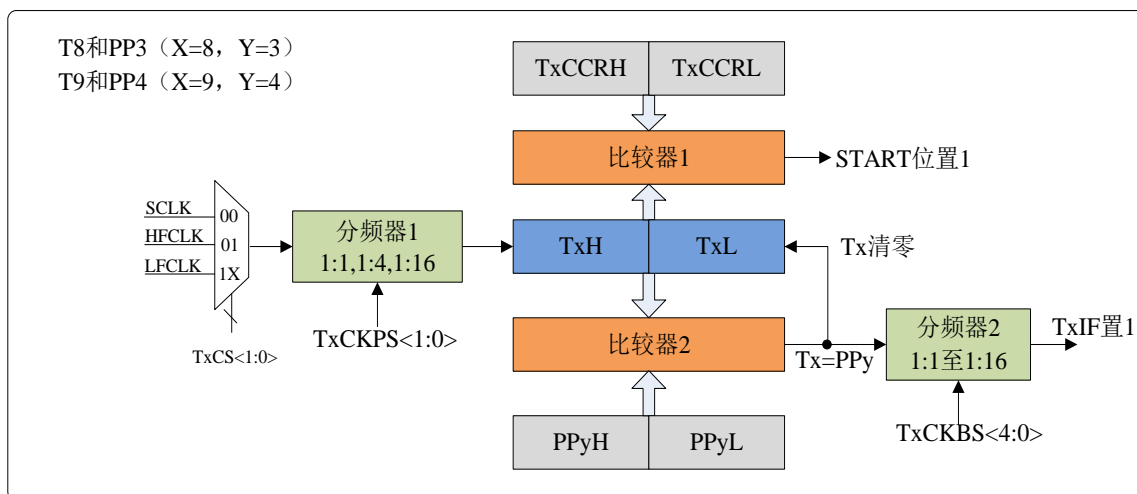


图 8.10 定时器 T8 原理框图

## 8.6.2 T8/9 相关的寄存器

表 8.6 与 T8 相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
44CH	T8CTL	-	T8CKBS3	T8CKBS2	T8CKBS1	T8CKBS0	T8ON	T8CKPS1	T8CKPS0
41DH	T8CTL1							T8CS1	T8CS0
44DH	T8H	T8 高 8 位寄存器							
44EH	T8L	T8 低 8 位寄存器							
442H	PP3H	PWM3 周期寄存器高 8 位							
443H	PP3L	PWM3 周期寄存器低 8 位							
44BH	T8CCRL	T8 启动 ADC 低 8 位设置寄存器							
44AH	T8CCRH	T8 启动 ADC 高 8 位设置寄存器							
45BH	T9CTL	-	T9CKBS3	T9CKBS2	T9CKBS1	T9CKBS0	T9ON	T9CKPS1	T9CKPS0
41EH	T9CTL1							T9CS1	T9CS0
45CH	T9H	T9 高 8 位寄存器							
45DH	T9L	T9 低 8 位寄存器							
451H	PP4H	PWM4 周期寄存器高 8 位							
452H	PP4L	PWM4 周期寄存器低 8 位							
45AH	T9CCRL	T9 启动 ADC 低 8 位设置寄存器							
459H	T9CCRH	T9 启动 ADC 高 8 位设置寄存器							

## 8.6.2.1 T8 控制寄存器 T8CTL

寄存器: T8CTL: T8控制寄存器(地址: 44CH)

复位值	bit7							bit0
-000 0000	-	T8CKBS3	T8CKBS2	T8CKBS1	T8CKBS0	T8ON	T8CKPS1	T8CKPS0
	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

T8CKBS&lt;3:0&gt;: T8 分频器 2 分频比选择位

- 0000 = 分频比为 1:1
- 0001 = 分频比为 1:2
- 0010 = 分频比为 1:3
- 0011 = 分频比为 1:4
- 0100 = 分频比为 1:5
- 0101 = 分频比为 1:6
- 0110 = 分频比为 1:7
- 0111 = 分频比为 1:8
- 1000 = 分频比为 1:9
- 1001 = 分频比为 1:10
- 1010 = 分频比为 1:11
- 1011 = 分频比为 1:12
- 1100 = 分频比为 1:13
- 1101 = 分频比为 1:14

- 1110 = 分频比为 1:15
- 1111 = 分频比为 1:16
- T8ON: T8 使能位
  - 1 = 使能 T8
  - 0 = 禁止 T8
- T8CKPS<1:0>: T8 分频器 1 分频比选择位
  - 00 = 分频比为 1:1
  - 01 = 分频比为 1:4
  - 1x = 分频比为 1:16

图注: R=可读 W=可写 -=未用 U=未实现位

### 8.6.2.2 T8 控制寄存器 T8CTL1

寄存器: T8CTL1: T8控制寄存器(地址: 41DH)

	bit7						bit0	
复位值 ----00	-	-	-	-	-	-	T8CS1	T8CS0
	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- T8CS<1:0>: T8 定时/计数模式选择
  - 00 = 时钟关闭
  - 01 = 选用 SCLK 时钟为 T8 工作时钟
  - 10 = 选用 HFCLK 时钟为 T8 工作时钟
  - 11 = 选用 LFCLK 时钟为 T8 工作时钟

### 8.6.2.3 T9 控制寄存器 T9CTL

寄存器: T9CTL: T9控制寄存器(地址: 45BH)

	bit7						bit0	
复位值 -000 0000	-	T9CKBS3	T9CKBS2	T9CKBS1	T9CKBS0	T9ON	T9CKPS1	T9CKPS0
	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- T9CKBS<3:0>: T9 分频器 2 分频比选择位
  - 0000 = 分频比为 1:1
  - 0001 = 分频比为 1:2
  - 0010 = 分频比为 1:3
  - 0011 = 分频比为 1:4
  - 0100 = 分频比为 1:5
  - 0101 = 分频比为 1:6
  - 0110 = 分频比为 1:7
  - 0111 = 分频比为 1:8



1000 = 分频比为 1:9  
 1001 = 分频比为 1:10  
 1010 = 分频比为 1:11  
 1011 = 分频比为 1:12  
 1100 = 分频比为 1:13  
 1101 = 分频比为 1:14  
 1110 = 分频比为 1:15  
 1111 = 分频比为 1:16

T9ON: T9 使能位  
 1 = 使能 T8  
 0 = 禁止 T8

T9CKPS<1:0>: T9 分频器 1 分频比选择位  
 00 = 分频比为 1:1  
 01 = 分频比为 1:4  
 1x = 分频比为 1:16

图注：R=可读 W=可写 -=未用 U=未实现位

#### 8.6.2.4 T9 控制寄存器 T9CTL1

寄存器: T9CTL1: T9控制寄存器(地址: 41EH)

复位值	bit7						bit0	
----00	-	-	-	-	-	-	T9CS1	T9CS0
	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

T9CS<1:0>: T9 定时/计数模式选择  
 00 = 时钟关闭  
 01 = 选用 SCLK 时钟为 T9 工作时钟  
 10 = 选用 HFCLK 时钟为 T9 工作时钟  
 11 = 选用 LFCLK 时钟为 T9 工作时钟

#### 8.6.2.5 T8CCRH/L 和 T9CCRH/L 寄存器

T8CCRH 和 T8CCRL 以及 T9CCRH 和 T9CCRL 寄存器是 T8 与 T9 分别启动 AD 转换寄存器组。系统不停的将 T8H/L 的值与寄存器 T8CCRH/L 相比较（将 T9H/L 的值与寄存器 T9CCRH/L 相比较），如果相等且对应的控制位置 1，则会发出信号，使 ADCCTL0 的 START（ADCCTL0<1>）位置 1，使能 ADC 采样转换。

### 8.6.3 T8/T9 中断

预分频器 2 的计算器达到设定值后将会使中断标志位置 1，如果使能位 T8IE/T9IE 为

1，且全局中断和外设中断允许位为 1，将会响应 T8/T9 中断。如果 IPEN 和 PT8/PT9 位均置 1，则 T8/T9 为高优先级中断。

### 8.6.4 T8/T9 在休眠模式

单片机进入休眠模式后，如果 T8/9 使用 SCLK，T8/9 将停止工作。

### 8.6.5 T8/T9 分配给 PWM3/4

PWM3 在使用时，需要使用 T8 进行定时，PWM4 在使用时，需要使用 T9 进行定时。PWM3/4 的详细内容请参考 PWM3/4 章节部分。

## 8.7 定时器 T10

T10 是一个带 16 位周期寄存器、预分频器和后分频器 16 位定时器，由 2 个 8 位寄存器 T10H 和 T10L 组成。没有外部计数时钟输入脚。

### 8.7.1 T10 的工作原理

如下图所示，为 T10 的原理框图，T10 的计数时钟为振荡器时钟（Fosc），每个周期（分频器 1 分频比为 1:1 时）寄存器 T10 的值自动加 1。通过 T10CTL0 的 T10ON 启动/禁止 T10。

Fosc 先经过分频器 1 分频后送到 T10 寄存器，进行加 1 计数，分频器 1 通过设置寄存器 T10CTL0 中的 T10CKPS<1:0>位，可将 Fosc 进行 1:1、1:4 和 1:16 分频。

当 T10H/L 寄存器与 PP5H/L 相等时，T10 自动清 0，发出相等信号给分频器 2，分频器 2 递增。同时 T10H/L 的值与寄存器 T10CCRxH/L 相比较，如果等于 T10CCRxH/L，且对应的控制位 T10CCRxON 置 1，则会发出信号，使 ADCATL0 的 START（ADCCTL0<1>）位置 1。

分频器 2 的分频比可通过寄存器 T10CTL0 中的 T10CKBS<3:0>位设置为 1:1 至 1:16。当其设置为 1:1 时，每次寄存器 T10H/L 与 PP5H/L 相等，将会使 T10 中断标志位 T10IF 置 1；当其设置为 1:2 时，寄存器 T10H/L 与 PP5H/L 相等累计两次才会使 T10IF 置 1，以此类推。

如果对 T10 寄存器执行写操作、对寄存器 T10CTL0 执行写操作或发生任何器件复位事件（上电复位、 $\overline{\text{RST}}$  复位、看门狗复位或欠压复位），分频器 1/2 的计数器将被清 0。

寄存器 T10H/L、PP5 和 T10CCRxH/L 均可读写。任何复位时，寄存器 T10H/L 均被设置为 00H，T10CCRxH/L 被设置为 00H/00H，PP5 被设置为 FFH。



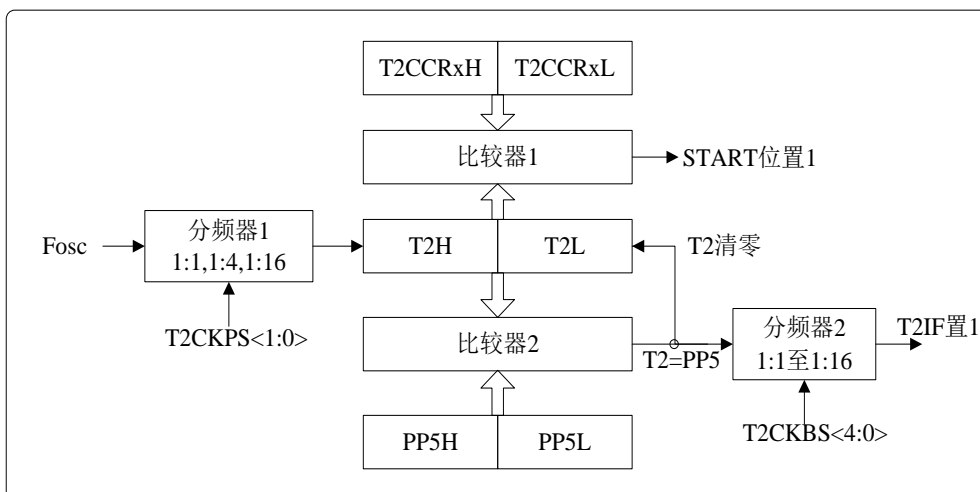


图 8.11 定时器 T10 原理框图

注：Fosc 为振荡器时钟。

### 8.7.2 T10 相关的寄存器

表 8.7 与 T10 相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
62DH	T10CTL0	T10CKBS3	T10CKBS2	T10CKBS1	T10CKBS0	T10CS1	T10CS0	T10CKPS1	T10CKPS0
628H	T10CTL1	T10MOD1	T10MOD0	T10OFS1	T10OFS0	T10TRIGEN	T10ON	-	T10DIR
62FH	T10H	T10 高字节寄存器							
62EH	T10L	T10 低字节寄存器							
638H	PP5H	PWM5 周期寄存器高 8 位							
639H	PP5L	PWM5 周期寄存器低 8 位							
62AH	T10CCR0L	T10 触发 ADC 启动设置寄存器 0 低 8 位							
629H	T10CCR0H	T10 触发 ADC 启动设置寄存器 0 高 8 位							
62CH	T10CCR1L	T10 触发 ADC 启动设置寄存器 1 低 8 位							
62BH	T10CCR1H	T10 触发 ADC 启动设置寄存器 1 高 8 位							

## 8.7.2.1 T10 控制寄存器 T10CTL0

寄存器: T10CTL0: T10控制寄存器1(地址: 62DH)

		bit7						bit0	
复位值		T10CKBS	T10CKBS	T10CKBS	T10CKBS	T10CS1	T10CS0	T10CKPS	T10CKPS
0000 0000		3	2	1	0			1	0
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

T10CKBS&lt;3:0&gt;: T10 分频器 2 (后分频) 分频比选择位

- 0000 = 分频比为 1:1
- 0001 = 分频比为 1:2
- 0010 = 分频比为 1:3
- 0011 = 分频比为 1:4
- 0100 = 分频比为 1:5
- 0101 = 分频比为 1:6
- 0110 = 分频比为 1:7
- 0111 = 分频比为 1:8
- 1000 = 分频比为 1:9
- 1001 = 分频比为 1:10
- 1010 = 分频比为 1:11
- 1011 = 分频比为 1:12
- 1100 = 分频比为 1:13
- 1101 = 分频比为 1:14
- 1110 = 分频比为 1:15
- 1111 = 分频比为 1:16

T10CS&lt;1:0&gt; T10 工作时钟选择

- T10CS=00 时钟关闭
- T10CS=01 T10 时钟为系统时钟 SCLK
- T10CS=10 T10 时钟为外设高频时钟 HFCLK
- T10CS=11 T1 时钟为外设低频时钟 LFCLK

T10CKPS&lt;1:0&gt;: T10 分频器 1 (预分频) 分频比选择位

- 00 = 分频比为 1:1
- 01 = 分频比为 1:4
- 1x = 分频比为 1:16

## 8.7.2.2 T10 控制寄存器 T10CTL1

寄存器: T10CTL1: T10控制寄存器1(地址:628H)

		bit7						bit0	
复位值		T10MOD1	T10MOD0	T10OFS1	T10OFS0	T10TRIGE	T10ON	-	T10DIR
0000 00-u						N			
		R/W	R/W	R/W	R/W	R/W	R/W	U	R

T10MOD&lt;1:0&gt;: T10 计数方式选择位。

- 00 = 向上计数, 上溢时产生中断标志
- 01 = 向上-向下计数, 上溢时产生中断标志

	10 = 向上-向下计数，下溢时产生中断标志
	11 = 向上-向下计数，上溢和下溢时都产生中断标志
T10OFS1:	T10 溢出中断选择，T10TRIGEN 为 1 时有效
	0 = 禁止上溢中断触发 AD
	1 = 使能上溢中断触发 AD
T10OFS0:	T10 溢出中断选择，T10TRIGEN 为 1 时有效
	0 = 禁止下溢中断触发 AD
	1 = 使能下溢中断触发 AD
T10TRIGEN:	T10 溢出中断自动触发 AD 控制位，当 AD 转换结束后自动清零。
	0 = 禁止 T10 溢出中断自动触发 AD
	1 = 使能 T10 溢出中断自动触发 AD
T10ON:	T10 使能位
	1 = 使能 T10
	0 = 禁止 T10
T10DIR:	T10 计数方向标志位（只读）
	0 = 当前计数方向为向下计数
	1 = 当前计数方向为向上计数

### 8.7.2.3 T10CCR1/0H 和 T10CCR1/0L 寄存器

T10CCR1/0H 和 T10CCR1/0L 寄存器是 T10 触发 ADC 启动设置寄存器。通过 T10CCR<sub>x</sub>ON (x=0、1) 位可以控制相应的 T10CCR<sub>x</sub>H/L (x=0、1) 是否工作。当 T10CCR<sub>x</sub>ON (x=0、1) 使能时，系统不停的将 T10H/L 的值与寄存器 T10CCR<sub>x</sub>H/L (x=0、1) 相比较，如果等于 T10CCR<sub>x</sub>H/L (x=0、1)，且对应的控制位 T10CCR1EN (ADCCTL3 寄存器) 置 1，则会发出信号，使 ADCCTL0 的 START (ADCCTL0<1>) 位置 1，使能 ADC 采样转换。

### 8.7.3 T10 中断

预分频器 2 的计算器达到设定值后将会使中断标志位置 1，如果使能位 T10IE 为 1，且全局中断和外设中断允许位为 1，将会响应 T10 中断。如果 IPEN 和 PT10 位均置 1，则 T10 为高优先级中断。

### 8.7.4 T10 在休眠模式

单片机进入休眠模式后，如果 T10 使用 SCLK 时钟源，T10 将停止工作。

### 8.7.5 T10 分配给 PWM5

PWM5 在使用时，需要使用 T10 进行定时，PWM5 的详细内容请参考 PWM5 部分。

## 9 模数(A/D)转换模块

模数(A/D)转换模块可将模拟输入信号转换为 12 位二进制值。KF8A100 拥有 0~44 路模拟输入通道。转换器通过逐次逼近法将模拟输入信号转换为二进制值，并将转换结果存放到 12 位寄存器中。可通过软件方式选择 VDD 或施加在 ADVRIN 引脚上的电压作为转换使用的参考电压。图 9.1 显示了 KF8A100 中 A/D 转换模块的结构框图。

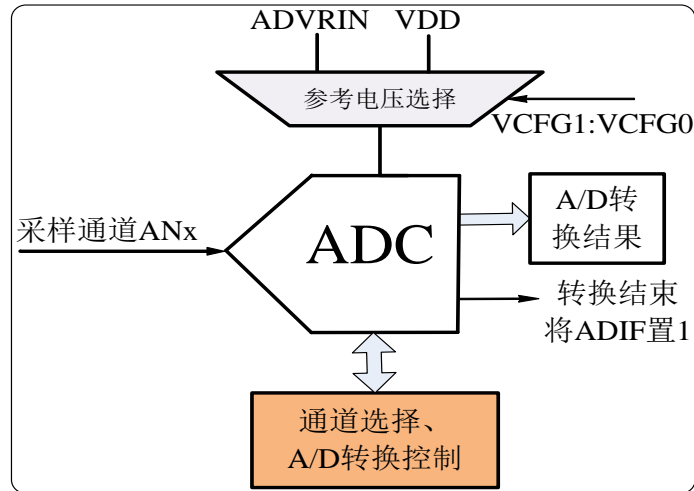


图 9.1 AD 模块结构框图

### 9.1 与 AD 相关的寄存器

表 9.1 与 AD 转换相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
1FH	ADCCTL0	ADLR	T2CCR0 ON	-	-	ADCLK1	ADCLK0	START	ADEN
3FH	ADCCTL1	ADCALE N	ADCS2	ADCS1	ADCS0	VCFG1	VCFG0	保留	保留
50H	ADCCTL2	-	ADCHS6	ADCHS5	ADCHS4	ADCHS3	ADCHS2	ADCHS1	ADCHS0
633H	ADCCTL3	-	-	-	-	T10CCR 1ON	T10CCR 0ON	T9CCRO N	T8CCRO N
505H	ADCOFFSET	-	-	-	-	-	-	-	-
25BH	ADSCANCTL	ADSCANEN	SMOD1	SMOD0	STIM4	STIM3	STIM 2	STIM 1	STIM 0
25AH	ADCINTCTL	T2CCR1 ON	T4CCR0 ON	ADTEST	T6CCR0 ON	T5CCR1 ON	T5CCR0 ON	INTCTL1	INTCTL0
1EH	ADCDATA0H	AD 数据寄存器 0 高 8 位							
3EH	ADCDATA0L	AD 数据寄存器 0 低 8 位							
252H	ADCDATA1H	AD 数据寄存器 1 高 8 位							
251H	ADCDATA1L	AD 数据寄存器 1 低 8 位							

254H	ADCDATA2H	AD 数据寄存器 2 高 8 位							
253H	ADCDATA2L	AD 数据寄存器 2 低 8 位							
256H	ADCDATA3H	AD 数据寄存器 3 高 8 位							
255H	ADCDATA3L	AD 数据寄存器 3 低 8 位							
560H	TEMPSNR	TSEN	-	-	-	-	-	-	-

## 9.1.1 AD 控制寄存器 0(ADCCTL0)

**寄存器:** ADCCTL0: A/D控制寄存器0(地址: 1FH)

	bit7						bit0	
复位值 00--0000	ADLR	T2CCR0ON	-	-	ADCLK1	ADCLK0	START	ADEN
	R/W	R/W	U	U	R/W	R/W	R/W	R/W

**ADLR:** A/D 转换结果输出格式选择位

- 1 = 结果右对齐
- 0 = 结果左对齐

**T2CCR0ON:** T2CCR0H/L 寄存器触发 AD 启动使能位

- 1 = 使能 T2 触发 ADC 启动, , ADC12 采样转换结束后, 将采样的数据保存到 ADCDATA1H 和 ADCDATA1L 中, 并且硬件自动将该位清
- 0 = 禁止 T2 触发 AD 启动

**ADCLK<1:0>:**A/D 工作时钟选择位

- 00 = 选用 SCLK 作为 A/D 工作时钟
- 01 = 选用 HFCLK 作为 A/D 工作时钟
- 1x = 选用 LFCLK 作为 A/D 工作时钟

**START:** A/D 转换状态位

- 1 = A/D 转换正在进行,该位置 1 将启动 A/D 转换, 在转换结束后该位将被硬件自动清 0。
- 0 = A/D 转换结束或者未进行

**ADEN:** A/D 工作使能位

- 1 = 使能 A/D 转换模块工作
- 0 = A/D 转换器关闭且不消耗工作电流

注:如果内部使用参考电压VREOUT时(用作AD参考电压和比较器参考电压),不需要将寄存器VRECTL的VREOE位置1。

## 9.1.2 AD 控制寄存器 1(ADCCTL1)

**寄存器:** ADCCTL1: A/D控制寄存器1(地址: 3FH)

	bit7						bit0	
复位值 0000 0000	ADCALEN	ADCS2	ADCS1	ADCS0	VCFG1	VCFG0	保留	保留
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**ADCALEN:** A/D 失调校准使能位

- 0 = 关闭 A/D 校准

- 1 = 打开 A/D 校准
- ADCS<2:0>: A/D 转换时钟选择位
  - 000 = Fad= SCLK/2
  - 001 = Fad= SCLK/8
  - 010 = Fad= SCLK /32
  - x11 = Fad=内部专用 500KHz 时钟
  - 100 = Fad= SCLK /4
  - 101 = Fad= SCLK /16
  - 110 = Fad= SCLK /64
- VCFG<1:0>: A/D 转换参考电压选择位
  - 00= 断开
  - 01 = 供电电源 VDD 作为 ADC 参考电压
  - 10 = 外部电压输入 ADVRIN 作为 ADC 参考电压
  - 11 = 保留

注： Fad 为 A/D 转换时钟频率。

### 9.1.3 AD 控制寄存器 2(ADCCTL2)

如下寄存器所示，AD 控制寄存器 2 包含 AD 通道选择位 ADCHS<6:0>。

寄存器： ADCCTL2:A/D控制寄存器2（地址：50H）

复位值 -000 0000	bit7 -	ADCHS6	ADCHS5	ADCHS4	ADCHS3	ADCHS2	ADCHS1	ADCHS0 bit0
	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ADCHS<6:0>:模拟通道选择位

CHSx = AD 模拟通道 x

注：ADCHS<6:0>的十进制值代表通道序号，例 ADCHS<6:0> = 00001111，则 AD 采样通道为 AN15。

表 9.2 AD 采样通道选择

ADCHS<6:0>	通道	ADCHS<6:0> >	通道	ADCHS<6 >:0>	通道
0000_0000	AN0/P4.3	0001_1011	AN27/P5.1	0011_0110	
0000_0001	AN1/P4.4	0001_1100	AN28/P5.2	0011_0111	
0000_0010	AN2/P4.5	0001_1101	AN29/P5.3	0011_1000	
0000_0011	AN3/P3.0	0001_1110	AN30/P5.4	0011_1001	
0000_0100	AN4/P3.1	0001_1111	AN31/P5.5	0011_1010	
0000_0101	AN5/P3.2	0010_0000	AN32/P6.1	0011_1011	
0000_0110		0010_0001	AN33/P6.2	0011_1100	
0000_0111	AN7/P0.7	0010_0010	AN34/P6.3	0011_1101	
0000_1000	AN8/P0.6	0010_0011	AN35/P6.4	0011_1110	
0000_1001	AN9/P7.5	0010_0100	AN36/P6.5	0011_1111	
0000_1010		0010_0101	AN37/P6.6	0100_0000	
0000_1011	AN11/P0.5	0010_0110	AN38/P9.5	0100_0001	
0000_1100	AN12/P0.4	0010_0111	AN39/P9.4	0100_0010	

0000_1101	AN13/P0.3	0010_1000	AN40/P9.3	0100_0011	
0000_1110	AN14/P0.2	0010_1001	AN41/P9.2	0100_0100	
0000_1111	AN15/P0.1	0010_1010	AN42/P9.1	0100_0101	
0001_0000		0010_1011	AN43/P9.0	0100_0110	
0001_0001	AN17/P7.3	0010_1100	AN44/TSOUT	0100_0111	
0001_0010	AN18/PA.1	0010_1101		0100_1000	
0001_0011	AN19/PA.0	0010_1110		0100_1001	
0001_0100	AN20/P1.3	0010_1111		0100_1010	
0001_0101	AN21/P1.2	0011_0000		0100_1011	
0001_0110	AN22/P1.0	0011_0001		0100_1100	
0001_0111	AN23/P2.3	0011_0010		0100_1101	
0001_1000	AN24/P2.4	0011_0011		0100_1110	
0001_1001	AN25/P2.5	0011_0100		0100_1111	
0001_1010	AN26/P5.0	0011_0101			

注：表中“-”代表系统保留。

### 9.1.4 AD 中断控制寄存器 (ADCINTCTL)

**寄存器:** ADCINTCTL: A/D中断控制寄存器(地址:25AH)

	bit7						bit0	
复位值	T2CCR1O	T4CCR0O	ADTEST	T6CCR0O	T5CCR1O	T5CCR0O	INTCTL1	INTCTL0
0000 0000	N	N		N	N	N		
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- T2CCR1ON:** T2CCR1H/L 寄存器触发 AD 启动使能位  
 1 = 使能 T2 触发 AD, ADC12 采样转换结束后, 将采样的数据保存到 ADCDATA2H 和 ADCDATA2L 中, 并且硬件自动将该位清 0  
 0 = 禁止 T2 触发 AD
- T4CCR0ON:** T4CCR0H/L 寄存器触发 AD 启动使能位  
 1 = 使能 T4 触发 AD, ADC12 采样转换结束后, 将采样的数据保存到 ADCDATA3H 和 ADCDATA3L 中, 并且硬件自动将该位清 0  
 0 = 禁止 T4 触发 AD
- ADTEST:** A/D 锁存时间选择位 (内部测试位, 暂不对外开放)  
 0 = 锁存时间短  
 1 = 锁存时间长
- T6CCR0ON:** T6CCR0H/L 寄存器触发 AD 启动使能位  
 1 = 使能 T6 触发 AD, ADC12 采样转换结束后, 将采样的数据保存到 ADCDATA3H 和 ADCDATA3L 中, 并且硬件自动将该位清 0  
 0 = 禁止 T6 触发 AD
- T5CCR1ON:** T5CCR1H/L 寄存器触发 AD 启动使能位  
 1 = 使能 T5 触发 AD, ADC12 采样转换结束后, 将采样的数据保存到 ADCDATA2H 和 ADCDATA2L 中, 并且硬件自动将该位清 0  
 0 = 禁止 T5 触发 AD
- T5CCR0ON:** T5CCR0H/L 寄存器触发 AD 启动使能位

1 = 使能 T5 触发 AD, ADC12 采样转换结束后, 将采样的数据保存到 ADCDATA1H 和 ADCDATA1L 中, 并且硬件自动将该位清

0 = 禁止 T5 触发 AD

INTCTL<1:0>:AD 中断控制位

ADSCANEN = 0 时, 此位失效, 始终在完成 1 次转换后即产生中断

ADSCANEN = 1 时,

00 = 完成 1 次转换后产生中断

01 = 完成 2 次转换后产生中断

10 = 完成 3 次转换后产生中断

11 = 完成 4 次转换后产生中断

### 9.1.5 AD 控制寄存器 3 (ADCCTL3)

**寄存器:** ADCCTL3: A/D 中断控制寄存器(地址:633H)

复位值 ---- 0000	bit7				bit0			
	-	-	-	-	T10CCR1 ON	T10CCR0 ON	T9CCRON	T8CCRON
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**T10CCR1ON:** T10CCR1H/L 寄存器触发 AD 启动使能位

1 = 使能 T10 触发 AD, ADC12 采样转换结束后, 将采样的数据保存到 ADCDATA3H 和 ADCDATA3L 中, 并且硬件自动将该位清

0 = 禁止 T10 触发 AD

**T10CCR0ON:** T10CCR0H/L 寄存器触发 AD 启动使能位

1 = 使能 T10 触发 AD, ADC12 采样转换结束后, 将采样的数据保存到 ADCDATA2H 和 ADCDATA2L 中, 并且硬件自动将该位清

0 = 禁止 T10 触发 AD

**T9CCRON:** T9CCRH/L 寄存器触发 AD 启动使能位

1 = 使能 T9 触发 AD, ADC12 采样转换结束后, 将采样的数据保存到 ADCDATA1H 和 ADCDATA1L 中, 并且硬件自动将该位清

0 = 禁止 T9 触发 AD

**T8CCRON:** T8CCRH/L 寄存器触发 AD 启动使能位

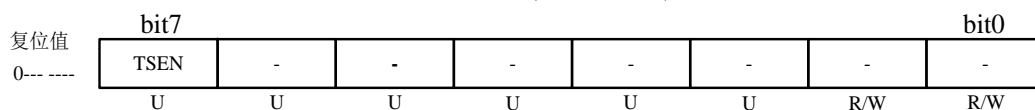
1 = 使能 T8 触发 AD, ADC12 采样转换结束后, 将采样的数据保存到 ADCDATA3H 和 ADCDATA3L 中, 并且硬件自动将该位清

0 = 禁止 T8 触发 AD



### 9.1.6 温度传感器配置寄存器(TEMPSNR)

寄存器: TEMPSNR: 温度传感器配置寄存器(地址:560H)



TSEN: 温度传感器使能位  
0 = 未使能温度传感器  
1 = 使能温度传感器

## 9.2 通道的选择

KF8A100 中的 A/D 转换模块的输入通过寄存器 ADCCTL2 进行通道的选择。当选择内部信号作为 ADC12 转换的输出时, 无需占用任何 I/O 引脚; ADC12 可选的内部信号输入及配置方式如下 (用户如需使用温度传感器, 通过 TEMPSNR 寄存器的 TSEN 位置 1 来使能该功能):

- 通道 44=TSOUT, 温度传感器的输出

## 9.3 模拟输入口的配置

当选择 ANx 作为 A/D 转换的输入时, 需要将对应的引脚配置为模拟输入口。通过将寄存器 ANSEx 的某位置 1 将对应的引脚配置为模拟口, 然后把相应的寄存器 TRx 的对应位置 1 把该引脚配置为输入口, 此时该引脚被设置为模拟输入口。

注: 如果某引脚被配置为模拟输入口, 将会自动禁止有效地数字 I/O、上拉电阻和电平变化中断。

## 9.4 多通道扫描模式

KF8A100 的 A/D 转换模块提供单通道扫描转换功能和多通道扫描转换功能。单通道扫描转换功能可以通过 ADCCTL2 寄存器的 ADCHS<6:0>位来选择模拟通道, 并将转换数据存入 ADCDATA0H/L 中。AN1、AN2、AN3、AN4 这 4 个通道为多通道扫描模式的固定通道。使用多通道扫描模式之前请将 ADCCTL2 设置成模拟通道 1。

用户可以通过设置 ADSCANCTL 寄存器的 ADSCANEN 位来使能多通道扫描模式, 当多通道扫描模式被使能后, 采样将被设置为自动采样模式, 用户可通过设置 ADSCANCTL 寄存器的 STIM<4:0>位来设置自动采样的时间; 通过设置 SMOD<1:0>位来设置多通道扫描的模式。例如 ADSCANEN=1, 且 SMOD<1:0>=11, 将会分别按顺序对 AN1、AN2、AN3、AN4 这 4 个通道进行采样转换, 先对 AN1 进行采样转换, 将转换数据存入 ADCDATA0H/L 中; 再对 AN2 进行采样转换, 将转换数据存入 ADCDATA1H/L 中; 再对 AN3 进行采样转换, 将转换数据存入 ADCDATA2H/L 中; 再对 AN4 进行采样转换, 将转换数据存入 ADC

DATA3H/L 中。

用户还可通过设置 ADCINTCTL 寄存器来设置中断产生的时间。

### 9.4.1 AD 多通道扫描控制寄存器

**寄存器:** ADSCANCTL: A/D多通道扫描控制寄存器(地址:25BH)

复位值	bit7						bit0	
0000 0000	ADSCAN EN	SMOD1	SMOD0	STIM4	STIM3	STIM2	STIM1	STIM0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ADSCANEN: 扫描模式使能位

1 = 使能多通道扫描模式

0 = 禁止多通道扫描模式

SMOD<1:0>: 扫描模式选择位

00= 默认采样转换通道 AN1

01 = 扫描采样通道 AN1, AN2

10 = 扫描采样通道 AN1, AN2, AN3

11 = 扫描采样通道 AN1, AN2, AN3, AN4

STIM<4:0>: 自动采样时间设置位

00001 = 1TAD

00010 = 2TAD

00011 = 3TAD

.....

11110=30TAD

11111=31TAD

## 9.5 A/D 转换参考电压的选择

KF8A100 中 ADC 模块的参考电压可以选择:电源电压(VDD)和外部参考电压(ADVRIN)。通过寄存器 ADCCTL1 的 VCFG<1:0>设置参考电压。

## 9.6 转换时钟的选择

完成一次 A/D 转换所需要的时间为  $13T_{ad}$ 。可通过软件方式设置 ADCS 位 (ADCCTL1<6:4>)选择转换时钟源, 共有 7 种时钟选项。 $T_{ad}$  和  $F_{ad}$  分别为 A/D 转换时钟周期和频率。

## 9.7 输出格式

KF8A100 中 A/D 转换的结果为 12 位二进制数, A/D 转换结果寄存器为两个 8 位的寄存器。用户可以通过 ADLR(ADCCTL0.7)设置转换结果输出格式, ADLR 置 1 输出为右对齐, ADLR 清 0 输出为左对齐。如下图所示。

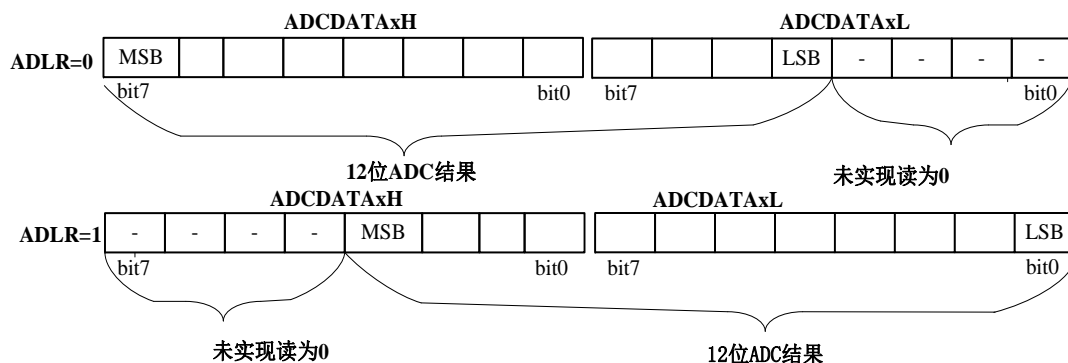


图 9.2 ADC 输出对齐方式

## 9.8 A/D 转换的启动和完成

先将 ADEN 位置 1，然后将 START 位(ADCCR0.1)置 1 即可启动 A/D 转换。ADEN 位置 1 后，AD 模块将开始对 AD 通道进行采样，START 位置 1 则启动 A/D 转换。用户可以根据具体情况增减在 ADEN 置 1 与 START 置 1 之间的时间，即采样时间。

当转换结束时，A/D 模块将：

1. 将 START 位清 0
2. 将 ADIF 位置 1
3. 如果使能 A/D 转换中断，则响应中断

可以采取在程序中将 START 位清 0 的方法中止当前的转换操作。在 A/D 转换采样全部结束之前，ADCDATAxH:ADCDATAxL 寄存器中的内容将不会被更新，而是仍旧保留前一次的转换结果。A/D 转换被中止后，需至少等待  $2T_{ad}$  的延时时间后才能开始下一次数据采集。

## 9.9 A/D 工作在休眠模式

A/D 转换器模块可以在休眠模式下工作。这需把 AD 的时钟源设定为 A/D 专用的内部振荡器（即  $ADCS\langle 2:0 \rangle = x11$ ）。当选择了专用内部时钟源后，A/D 需等待一个指令周期后才能启动转换操作，转换结束后，START 位将被清 0，且转换结果将被载入 ADCDATA0H: ADCDATA0L 寄存器。如果 A/D 中断被使能，器件将从休眠状态唤醒。如果 A/D 中断被禁止，A/D 转换模块在转换完成后被关闭。

如果 A/D 时钟源不是 AD 专用内部振荡器，执行 IDLE 指令将导致当前转换操作中止，并使 A/D 模块关闭。

## 9.10 复位的影响

器件复位将强制所有寄存器进入复位状态。因此，A/D 模块将被关闭，任何进行中的转换操作被中止。ADCDATA0H:ADCDATA0L 寄存器中的值不变。

## 9.11 使用 A/D 转换器的设置

启动 A/D 转换器时的设置:

1. 选择 A/D 采样输入通道, 设置 A/D 转换结果对齐方式;
2. 将对应的 A/D 采样输入通道设置为模拟输入模式;
3. 如果需要使能 AD 的失调校准, 将 ADCCTL1 寄存器的 ADCALEN 位置 1, 清 0 禁止;
4. 选择参考电压和 A/D 采样时钟频率, 打开 A/D 转换;
5. 如果采用中断方式, 使能 A/D 转换中断;
6. 等待 A/D 所需的采集时间;
7. 启动 A/D 转换;
8. 查询 A/D 是否转换完成(START=0)或进入 A/D 中断;
9. 读取 A/D 转换结果。

## 9.12 TxCCR<sub>x</sub> 触发 AD

T2/T4/T5/T6/T8/T9/T10 分别可以通过 T2CCR0H/L、T2CCR1H/L、T4CCR0H/L、T5CCR0H/L、T5CCR1H/L、T6CCR0H/L、T8CCR0H/L、T8CCR1H/L、T9CCR0H/L、T9CCR1H/L、T10CCR0H/L 寄存器触发 AD 启动转换。转换的结果均可在完成转换后通过 ADCDATA0H/L 寄存器读取; 此外, 转换结果还将保存在如下位置: T2CCR0H/L 和 T5CCR0H/L 以及 T10CCR0H/L 和 T8CCR0H/L 触发 AD 的转换结果将存入 ADCDATA1H/L 寄存器中; T2CCR1H/L 和 T5CCR1H/L 以及 T10CCR1H/L 触发 AD 的转换结果将存入 ADCDATA2H/L 寄存器中; T4CCR0H/L 和 T6CCR0H/L 以及 T10CCR1H/L 触发 AD 的转换结果将存入 ADCDATA3H/L 寄存器中。

例如: T2CCR0H/L 触发 AD 转换之后, 转换结果将同时存入 ADCDATA0H/L 和 ADCDATA1H/L 寄存器中, 当其他条件启动 AD 转换后, ADCDATA0H/L 寄存器中的结果将被最近一次转换所覆盖, 但是 ADCDATA1H/L 寄存器中的结果继续保留, 直到下一次 T2CCR0H/L 触发 AD 转换后覆盖或者软件清除。

## 9.13 CCP 触发 AD

CCP2/CCP5/CCP3/CCP7/CCP8/CCP9/CCP10 在特定模式下, 当特殊事件发生时可触发 AD 启动转换。具体描述请参考 CCP 相关章节。

## 10 PWM1/2 模块

脉宽调制（PWM）是一种通过在完全开启和完全关闭状态之间进行快速切换而为负载供电的方案。PWM 信号类似于方波，信号的高电平部分视为开启状态，信号的低电平部分视为关闭状态。高电平部分（也称为脉宽）可以随时间而变，并以步幅为单位进行定义。施加的步幅数量越多（这会增大脉宽），为负载提供的电量就越多。施加的步幅数量降低时（这会缩短脉宽），提供的电量就会下降。PWM 周期定义为一个完整周期的持续时间，或者开启和关闭时间相加的总时间。

PWM1/2 模块具有 2 种模式的 PWM 输出形式，2 路 8 位的 PWM 模式输出 PWM1OUT 和 PWM2OUT，或者 1 路 16 位的 PWM 模式输出 PWM16OUT。

设置 PWMMODE 后，启动相应的 PWMxON(x=1/2/16)后，在对应的 PWMxOUT(x=1/2/16)引脚输出 PWM 脉冲。8 位 T1 时基 PWM 模式下，PWM 脉冲的频率和占空比通过 PP1(或 PP2)和 PWM1L(或 PWM2L)设置。16 位 T1 时基 PWM 模式下，PWM 脉冲的频率和占空比通过 PP1、PP2 以及 PWM1L、PWM2L 设置，且 PP1 和 PP2 分别构成 PWM 周期的低位和高位，PWM1L 和 PWM2L 分别构成 PWM 占空比的低位和高位。

图 10.1 显示了 PWM 逻辑框图。在 8 位 T1 时基 PWM 模式下,其中 PP1 为 PWM1 模块的周期寄存器，PWM1L 为 PWM1 模块占空比设置寄存器，使用 PWM 时需要将定时器 1 配置给 PWM 做定时用，其中 T1L、T1IE 和 T1IF 分配给 PWM1，T1H 分配给 PWM2。启动 PWM1 后，当 T1L 计数值和 PP1 相等时，PWM1OUT 引脚被置 1，此时 T1L 被清 0，重新开始计数，当 T1L 的计数值和 PWM1L 相等时，PWM1OUT 引脚清 0(如图 12.2 所示)。改变 PP1 和 PWM1L 的值可产生不同的 PWM1OUT 周期和 PWM1OUT 占空比。PWM2 的工作原理和 PWM1 模块完全一致。

在 16 位 T1 时基 PWM 模式下,PP1 和 PP2 共同组成周期寄存器低位和高位，PWM1L 和 PWM2L 分别构成 PWM 占空比的低位和高位。使用 PWM16 时需要将定时器 1 配置给 PWM16 做定时用，即将 T1L、T1H、T1IE 和 T1IF 都分配给 PWM。启动 PWM16 后，当  $\langle T1H:T1L \rangle$  和  $\langle PP2: PP1 \rangle$  相等时，PWM16OUT 引脚被置 1，此时 T1 被清 0，重新开始计数，当 T1 的计数值和  $\langle PWM2L: PWM1L \rangle$  相等时，PWM16OUT 引脚清 0。改变  $\langle PP2: PP1 \rangle$  和  $\langle PWM2L: PWM1L \rangle$  的值可产生不同的 PWM16OUT 周期和 PWM16OUT 占空比。

### 10.1 PWM1/2 原理框图

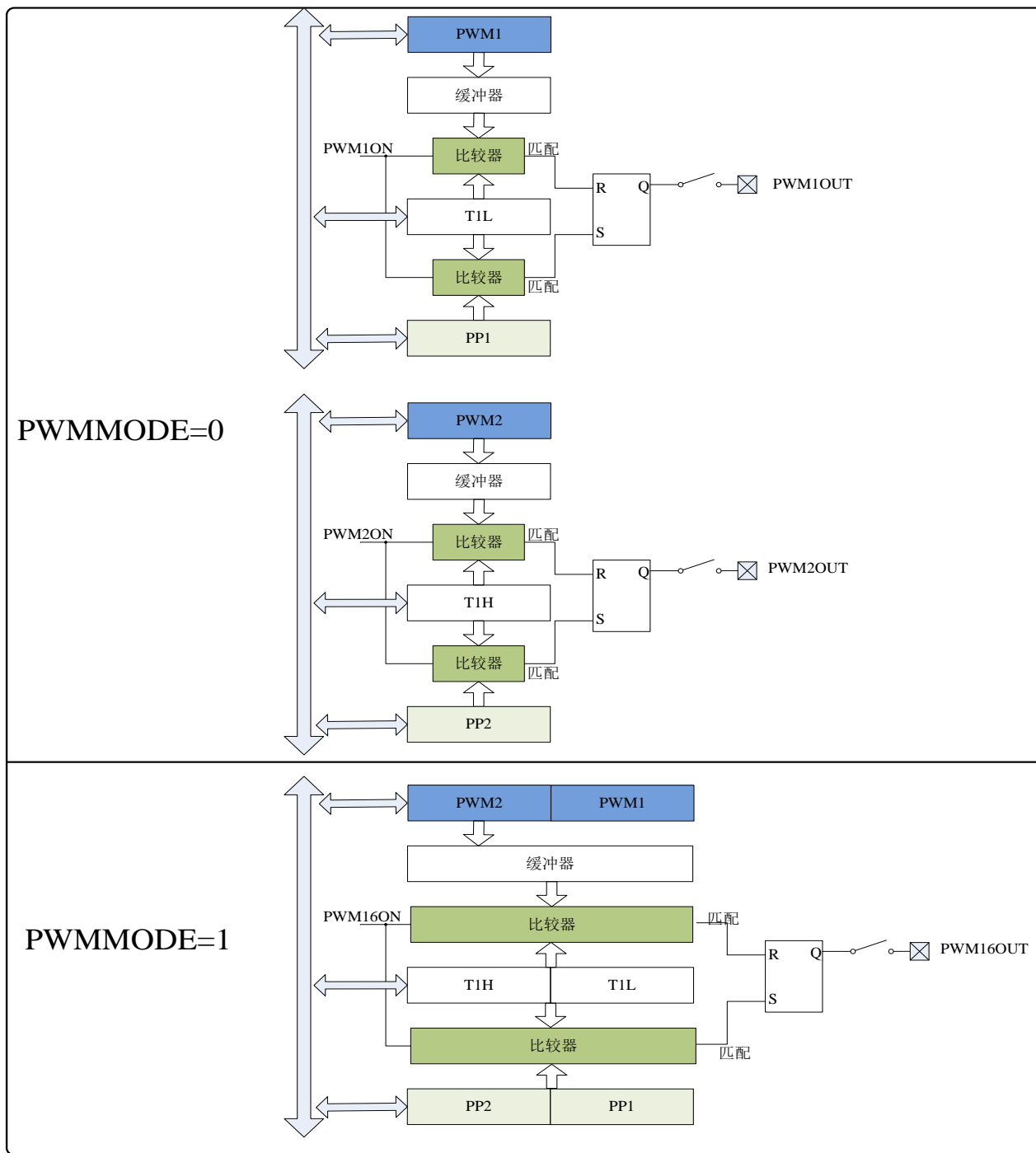


图 10.1 PWM 逻辑框图

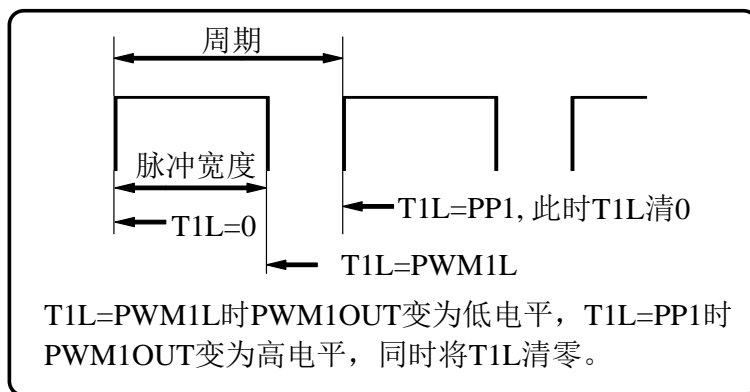


图 10-1 PWM1OUT 输出波形图

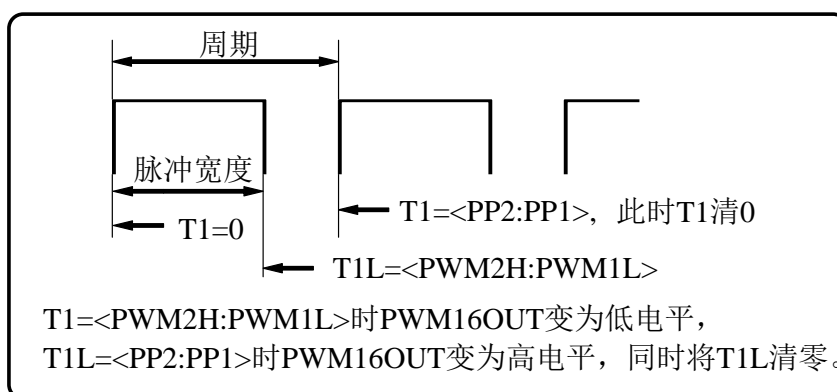


图 10-2 PWM1OUT 输出波形图

## 10.2 PWM 相关的寄存器

表 10.1 与 PWM 相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
13H	PWM1L	PWM 占空比低位寄存器							
16H	PP1	PWM 周期低位寄存器							
32H	PP2	PWM 周期高位寄存器							
33H	PWM2L	PWM 占空比高位寄存器							
15H	PWMCTL	-	INT1SE	保留	保留	PWMMODE	PWM16ON	PWM2ON	PWM1ON

## 10.3 PWM 控制寄存器

寄存器: PWMCTL: PWM启动控制寄存器(地址: 15H)

复位值	bit7							bit0	
1100 0000	-	INT1SE	保留	保留	PWMMODE	PWM16ON	PWM2ON	PWM1ON	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

INT1SE: INT1 触发脉冲边沿选择位

1 = 上升沿触发

0 = 下降沿触

PWMMODE: PWM 模式选择位

1 = PWM 以 16 位 T1 计数器为时基, 控制 PWM16OUT 输出

0 = PWM 被分成 PWM1 和 PWM2, PWM1 以 T1L 为时基控制 PWM1OUT; PWM2 以 T1H 为时基, 控制 PWM2OUT

PWM16ON: PWM16 启动控制位 (PWMMODE=1 时有效)

1 = 启动 PWM16

0 = 禁止 PWM16

PWM2ON: PWM2 启动控制位 (PWMMODE=0 时有效)

1 = 启动 PWM2

0 = 禁止 PWM2

PWM1ON: PWM1 启动控制位 (PWMMODE=0 时有效)

1 = 启动 PWM1

0 = 禁止 PWM1

注: 保留位只能写 0, 不能写 1。

## 10.4 PWM 周期

8 位 PWM 模式下, PWM 周期通过 PP1/2 (地址: 16H/32H) 进行设置, PP1/2 是一个 8 位的寄存器, 其值可设置为 0~255。PWM 周期通过公式 10.1 进行计算。

公式 10.1:

$$\text{PWM周期} = (\text{PP}_x + 1) \cdot 4 \cdot T_{\text{sys}} \cdot (\text{T1 预分频比}) \quad (x=1, 2)$$



16 位 PWM 模式下，只需将 PP<sub>x</sub> 换成<PP2:PP1>进行计算。

## 10.5 PWM 占空比

8 位 PWM 模式下，占空比描述开启时间与关闭时间之间以百分比形式表示的比例，0% 代表完全关闭，100% 代表完全开启。占空比越低，对应的供电量就越低；占空比越高，对应的供电量就越高。

PWM 占空比通过 PWM1<sub>x/2x</sub>(地址: 13H/60H/33H/61H)设置，可写入一个 8 位的值到 PWM1<sub>x/2x</sub> 来设置占空比。脉冲宽度和占空比通过公式 10.2 和公式 10.3 计算:

公式 10.2:

$$\text{脉冲宽度} = \text{PWM}_x \cdot 4 \cdot T_{\text{sys}} \cdot (T1 \text{ 预分频比}) \quad (\text{PWM}_x = \text{PWM1L} / \text{PWM2H})$$

公式 10.3:

$$\text{占空比} = \frac{\text{脉冲宽度}}{\text{PWM 周期}} = \frac{\text{PWM}_x}{\text{PP}_x + 1} \quad (\text{PWM}_x = \text{PWM1L} / \text{PWM2H})$$

16 位 PWM 模式下，只需将 PWM<sub>x</sub> 换成<PWM2L:PWM1L>进行计算。

## 10.6 PWM 分辨率

8 位 PWM 模式下，PWM 分辨率定义可以在单个 PWM 周期中出现的最大步幅数量。分辨率越高，就可以越精确地控制脉宽时间，从而更精确地控制在负载上的供电量。

分辨率决定在给定周期内的占空比数。例如，10 位分辨率将产生 1024 个离散的占空比，8 位分辨率产生 256 个离散的占空比。当 PP1/2 为 255 时，PWM 的最大分辨率为 8 位。分辨率的计算公式如下所示。

公式 10.4:

$$\text{分辨率} = \frac{\text{Log}[(\text{PP}_x + 1)]}{\text{log}2} \text{ 位} \quad (x=1、2)$$

16 位 PWM 模式下，只需将 PP<sub>x</sub> 换成<PP2:PP1>进行计算。

## 10.7 PWM 中断

PWM2 有一个专门的中断使能位 PWM2IE 和中断标志位 PWM2IF，当启动 PWM2 后，会对 PWM2IF 产生影响，而 PWM1/PWM16 和定时器 T1 共用中断使能位 T1IE 和中断标志位 T1IF。

在 PWM1/2 启用后，当 T1L/H 的计数值与 PWM1L、PWM2L 的值匹配后，其对应的输出引脚变为低电平。当 T1L/H 的计数值与 PP1/2 的值匹配后，其对应的输出引脚变为高电平，同时将 T1L/H 清 0，将 T1IF/PWM2IF 置 1，如果允许 T1 或 PWM2 中断，将会转入

对应的中断子程序中。

在 PWM16 启用后，当 T1 和 <PWM2L:PWM1L> 匹配后，其对应的输出引脚变为低电平。当 T1 的计数值与 <PP2:PP1> 的值匹配后，其对应的输出引脚变为高电平，同时将 T1 清 0，将 T1IF 置 1，如果允许 T1 中断，将会转入对应的中断子程序中。

## 10.8 休眠模式下的操作

在休眠模式下，T1 寄存器将不会递增并且模块的状态将保持不变。PWM 输出引脚电平保持不变(如果输出为高电平，则保持高电平，如果为低电平保持低电平)。当器件被唤醒时，T1 将从原来的状态继续工作。

## 10.9 复位的影响

任何复位都会将所有端口强制为输入模式，并强制 PWM 使用的寄存器进入其复位状态。

## 10.10 PWM 使用方法

**PWM 工作的设置应按照以下步骤：**

1. 将 PWM 输出 IO 口状态配置为输入态，禁止 IO 口输出。
2. 赋 PP1 或 PP2 寄存器的初值以设置 PWM<sub>xx</sub> 的 PWM 周期。
3. 赋 PWM<sub>xx</sub> 寄存器的初值以设置 PWM<sub>xx</sub> 的占空比。
4. 配置并启动定时器/计数器 T1:
  - 配置 T1CTL 寄存器的 TICKS1 和 TICKS0 以选择 T1 的预分频比；
  - 将 T1L/H 清 0；
  - 将 T1CTL 寄存器的 T1ON 位置 1 以启动 T1。
1. 设置 PWMCTL 寄存器的 PWM<sub>MODE</sub>。
2. 将 PWMCTL 寄存器的 PWM<sub>x</sub>ON 置 1 以启动 PWM<sub>x</sub> (x=1/2/16)。
6. 将 PWM 输出 IO 口状态配置为输出态，输出 PWM 波形。

## 11 CCP2/5(捕捉/比较/PWM6、7)模块

KF8A100 单片机内部集成有 CCP2/5 模块。CCP2/5 模块是指捕捉/比较/脉宽调制模块，可以提供外部信号捕捉、内部比较输出以及 PWM 输出三种功能。定时器 T2 和 T4 分配给 CCP2/PWM6，T5 和 T6 分配给 CCP5/PWM7。

CCP、PWM 和定时器的对应关系如下图所示：

表 11.1 CCP、PWM 和定时器的对应关系

TMR	CCP	PWM
T2	CCP2	PWM6
T4		
T5	CCP5	PWM7
T6		

### 11.1 捕捉/比较相关寄存器

表 11.2 捕捉/比较模块的相关寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
43H	PWM6L0	捕捉/比较/PWM6 通道 0 占空比寄存器低 8 位							
44H	PWM6H0	捕捉/比较/PWM6 通道 0 占空比寄存器高 8 位							
57H	PWM6CTL0	-	-	-	-	P6MOD3	P6MOD2	P6MOD1	P6MOD0
55AH	PWM7L0	捕捉/比较/PWM7 通道 0 占空比寄存器低 8 位							
55BH	PWM7H0	捕捉/比较/PWM7 通道 0 占空比寄存器高 8 位							
510H	PWM7CTL0	-	-	-	-	P7MOD3	P7MOD2	P7MOD1	P7MOD0

## 11.1.1 PWM6CTL0 寄存器

寄存器: PWM6CTL0: PWM6控制寄存器0(地址: 57H)

	bit7					bit0			
复位值 0000 0000	-	-	-	-	P6MOD3	P6MOD2	P6MOD1	P6MOD0	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P6MOD&lt;3:0&gt;: PWM6模块的模式选择位

- 0000 = 捕捉/比较/PWM 关闭 (复位CCP模块)
- 0001 = 未使用 (保留)
- 0010 = 比较模式: 匹配值加1时输出电平翻转 (CCPxIF 置1)
- 0011 = 未使用 (保留)
- 0100 = 捕捉模式: 在每个下降沿发生捕捉
- 0101 = 捕捉模式: 在每个上升沿发生捕捉
- 0110 = 捕捉模式: 每4 个上升沿发生捕捉
- 0111 = 捕捉模式: 每16 个上升沿发生捕捉
- 1000 = 比较模式: 比较匹配值加1时输出高电平 (CCPx1IF置1)
- 1001 = 比较模式: 比较匹配时输出低电平 (CCPxIF置1)
- 1010 = 比较模式: 比较匹配时产生软件中断 (CCPxIF置1, P6CH0H引脚不受影响)
- 1011 = 比较模式: 触发特殊事件 (CCPxIF 位置1, 复位 T2, 如果ADC使能将启动AD转换)
- 1100 = PWM自由模式, T2作为通道0/1/2的时基, T4作为通道3的时基, T2/T4可独立使能生成PWM输出。
- 1101 = PWM协同模式, T2作为通道0/1/2的时基, T4作为通道3的时基, T2和T4需要全部使能才能生成PWM输出。
- 1110 = PWM单时基 (T2) 模式, T2作为通道0/1/2/3的时基, T2使能控制生成PWM输出。
- 1111 = PWM6/7协同模式, T2和T5分别作为PWM6和PWM7通道0/1/2的时基, T4和T6分别作为PWM6和PWM7通道3的时基, T2和T5需全部使能通道0/1/2才能生成PWM输出, T4和T6需全部使能通道3才能生成PWM输出。

### 11.1.2 PWM7CTL0 寄存器

寄存器: PWM7CTL0: PWM7控制寄存器0(地址: 510H)

	bit7				bit0			
复位值 0000 0000	-	-	-	-	P7MOD3	P7MOD2	P7MOD1	P7MOD0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P7MOD<3:0>: PWM7模块的模式选择位

- 0000 = 捕捉/比较/PWM 关闭 (复位CCP5 模块)
- 0001 = 未使用 (保留)
- 0010 = 比较模式: 匹配时输出电平翻转 (CCP5IF 置1)
- 0011 = 未使用 (保留)
- 0100 = 捕捉模式: 在每个下降沿发生捕捉
- 0101 = 捕捉模式: 在每个上升沿发生捕捉
- 0110 = 捕捉模式: 每4 个上升沿发生捕捉
- 0111 = 捕捉模式: 每16 个上升沿发生捕捉
- 1000 = 比较模式: 比较匹配时输出高电平 (CCP5IF置1)
- 1001 = 比较模式: 比较匹配时输出低电平 (CCP5IF置1)
- 1010 = 比较模式: 比较匹配时产生软件中断 (CCP5IF置1, P7CH0H不受影响)
- 1011 = 比较模式: 触发特殊事件 (CCP5IF 位置1, CCP5 复位 T5, 如果ADC使能将启动AD转换)
- 1100 = PWM自由模式, T5作为通道0/1/2的时基, T6作为通道3的时基, T5/T6可独立使能生成PWM输出。
- 1101 = PWM协同模式, T5作为通道0/1/2的时基, T6作为通道3的时基, T5和T6需要全部使能才能生成PWM输出。
- 1110 = PWM单时基模式, T5作为通道0/1/2/3的时基, T5使能控制生成PWM输出。
- 1111 = PWM6/7协同模式, T2和T5分别作为PWM6和PWM7通道0/1/2的时基, T4和T6分别作为PWM6和PWM7通道3的时基, T2和T5需全部使能通道0/1/2才能生成PWM输出, T4和T6需全部使能通道3才能生成PWM输出。

### 11.1.3 PWM6H0/PWM6L0、 PWM7H0/PWM7L0 寄存器

在捕捉模式下时,当对应的 P<sub>x</sub>CH0H(x=6、7)引脚发生事件时, {PWM<sub>x</sub>H0:PWM<sub>x</sub>L0}(x=6、7)这对寄存器捕捉{T<sub>x</sub>H:T<sub>x</sub>L}(x=2、5)寄存器的 16 位值。

在比较模式下, 16 位 {PWM<sub>x</sub>H0:PWM<sub>x</sub>L0}(x=6、7)寄存器的值将不断与 {T<sub>x</sub>H:T<sub>x</sub>L}(x=2、5)寄存器的值相比较。当两者匹配时, CCP 就会触发相应的事件。

注: {PWM<sub>x</sub>H0:PWM<sub>x</sub>L0}(x=6、7)同时也是 PWM6/PWM7 通道 0 的占空比设置寄存器。

## 11.2 捕捉模式

在捕捉模式下，当捕捉输入引脚发生事件时，{PWMxH0:PWMxL0}(x=6、7)这对寄存器捕捉{T<sub>x</sub>H:T<sub>x</sub>L}(x=2、5)寄存器的16位值，原理框图如下图所示：

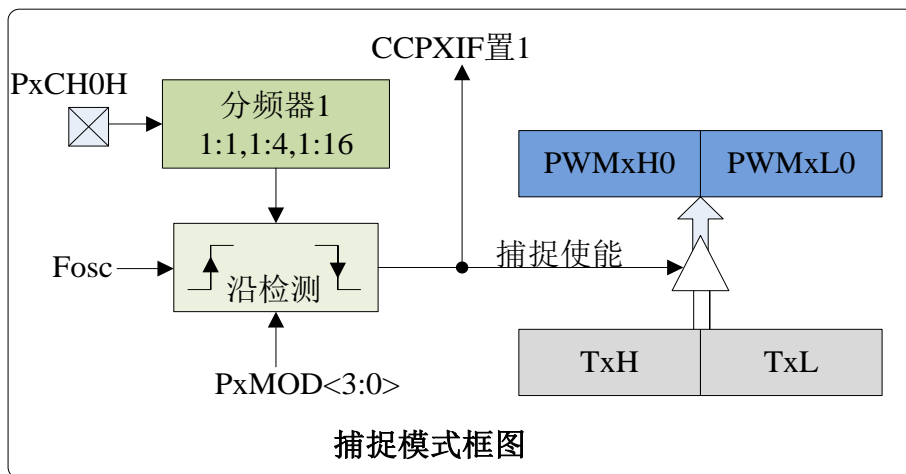


图 11.1 捕捉模式原理框图

注：Tx表示T2/T5，Px表示P6/P7,PWMx表示PWM6/PWM7；  
CCP2捕捉模式输入引脚选择位于APFCTL2寄存器的P7CH0HSEL位；  
CCP5捕捉模式输入引脚选择位于APFCTL2寄存器的P6CH0HSEL位。

触发捕捉的事件可被定义为以下四者之一，并且由PWMxCTL0(x=6、7)寄存器中的模式选择位PxMOD<3:0>(x=6、7)选择事件类型位配置：

- ◆ 0100 = 每个下降沿
- ◆ 0101 = 每个上升沿
- ◆ 0110 = 每4个上升沿
- ◆ 0111 = 每16个上升沿

在捕捉模式下，应通过将对应IO口方向控制寄存器TR位控制位置1，将相应的PxCH0H(x=6、7)引脚配置为输入数字口。

当一个捕捉发生时，硬件自动中断请求标志位EIF2/EIF5寄存器中的CCP2IF/CCP5IF置1；它必须用软件清零。注意如果在{PWMxH0:PWMxL0}(x=6、7)这对寄存器中的值被读取之前发生另一次捕捉，那么之前捕捉的值将被新捕捉的值覆盖。

当捕捉模式改变时，可能会产生错误的捕捉中断。用户应该在捕捉模式改变之前保持EIE2/EIE5寄存器中的CCP2IE/CCP5IE中断允许位清零以避免产生误中断。在捕捉模式发生任何改变之后也应清零EIF2/EIF5寄存器中的中断标志位CCP2IF/CCP5IF。

PWMxCTL0(x=6、7)寄存器中的PxMOD<3:0>(x=6、7)位指定了4种预分频器设置。每当关闭CCP模块或禁止捕捉模式时，就会清零预分频器计数器。这意味着任何复位都将清零预分频计数器。

虽然从一种捕捉预分频比切换到另一种捕捉预分频比不会将预分频计数器清零，但可能会产生误中断。因此要避免出现这种不期望的操作，应在改变预分频比前通过将PWMxCTL0(x=6、7)寄存器清零关闭该模块。

### 11.3 比较模式

在比较模式下，16位{PWMxH0:PWMxL0}(x=6、7)寄存器的值将不断与Tx(x=2,5)寄存器的值相比较。当两者匹配时，CCP模块可能会出现以下几种情况：

- ◆ PxCH0H(x=6、7)引脚的输出电平翻转
- ◆ PxCH0H(x=6、7)引脚输出高电平
- ◆ PxCH0H(x=6、7)引脚输出低电平
- ◆ 产生软件中断触发信号
- ◆ 产生特殊事件触发信号2

PxCH0H(x=6、7)引脚的动作取决于PWMxCTL0寄存器中PxMOD<3:0>控制位的值。

所有比较模式都会产生中断。原理图如下图所示：

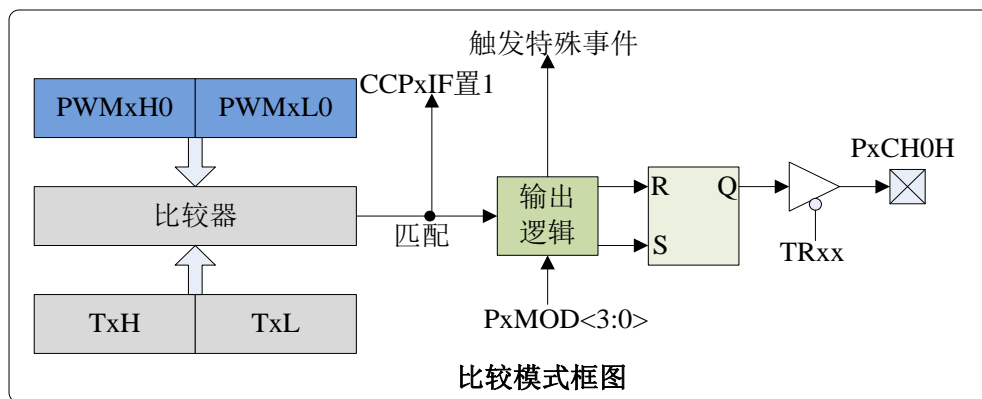


图 11.2 比较模式原理框图

注：Tx表示T2/T5，Px表示P6/P7,PWMx表示PWM6/PWM7,CCPx表示CCP2/5。

用户不需要将PxCH0H(x=6、7)引脚配置为输出，只要比较输出模式使能，硬件自动控制引脚输出。

当选择输出电平翻转模式(PxMOD<3:0>(x=6、7)=0010)时，比较匹配时，PxCH0H(x=6、7)引脚的输出电平翻转，并将CCP2IF/CCP5IF置1。

当选择普通比较模式时(PxMOD<3:0>(x=6、7)=1000或1001)时，比较匹配时，PxCH0H(x=6、7)引脚输出高电平或者低电平，并且将CCP2IF/CCP5IF置1。

当选择了软件中断触发模式时(PxMOD<3:0>(x=6、7)=1010)时，比较匹配时，将CCP2IF/CCP5IF置1，但是CCP模块不会控制PxCH0H(x=6、7)引脚。

当选择了特殊事件触发模式(PxMOD<3:0>(x=6、7)=1011)比较匹配时，CCP会立即产生特殊事件触发输出，将CCP2IF/CCP5IF置1，此时如果ADC已使能，将启动AD转换（这里以T2为例），T2H/T2L寄存器不会立即复位，直到T2计数脉冲的下一个上升沿才复位。从而使PWM6H0/PWM6L0寄存器实际上成为了T2的16位可编程周期寄存器。

### 11.4 PWM6/7 模式

PWM6/7 为增强型 PWM 模块。均有 4 个通道，可在 4 个不同的引脚输出 PWM 信号，分辨率最高 16 位。PWM6/7 的原理框图如下图所示：

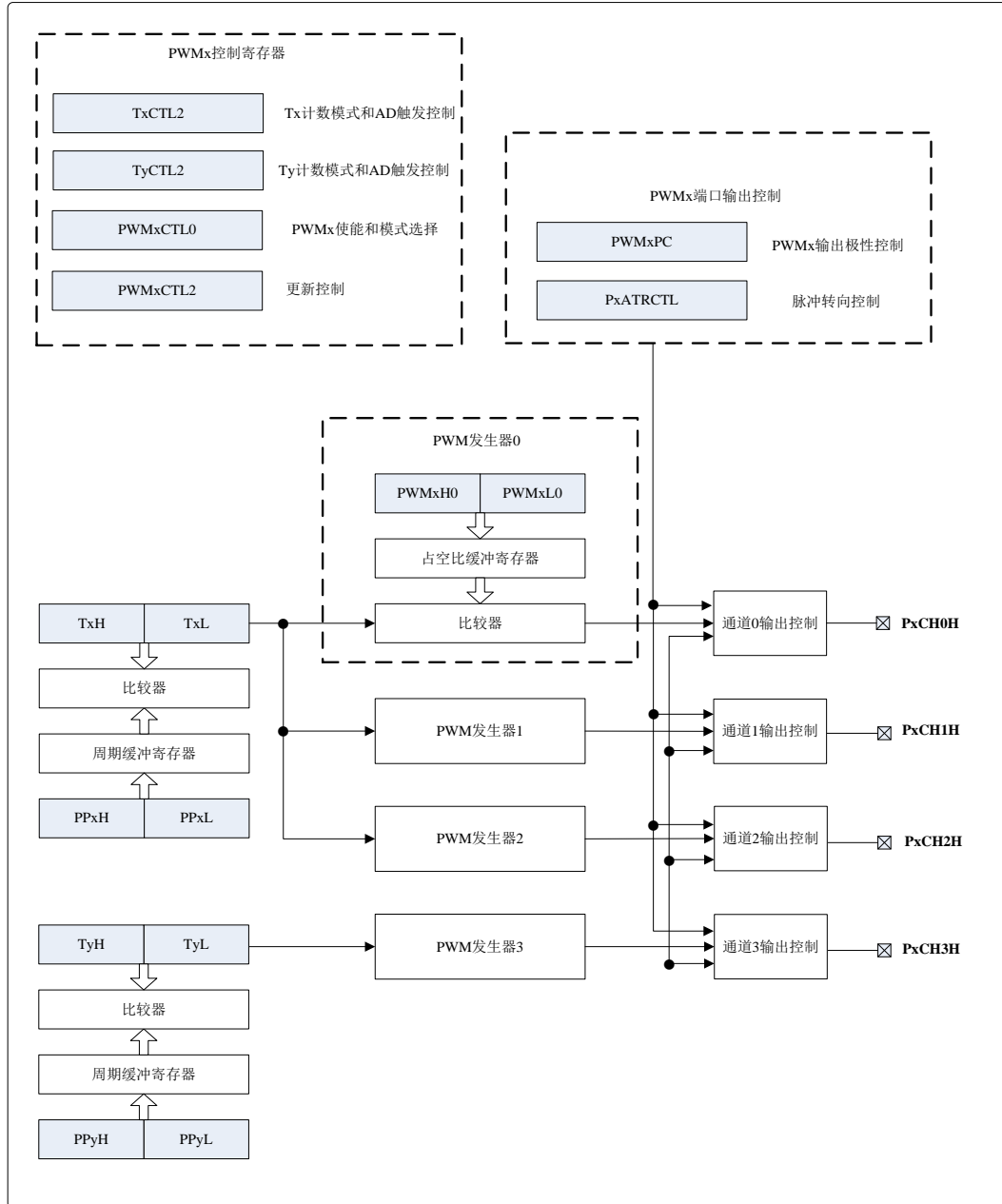


图 11.3 PWM6/7 原理框图

注： Tx表示T2/T5， Ty表示T4/T6， Px表示P6/P7,PWMx表示PWM6/PWM7， PPx表示PP60/PP70， PPy表示PP61/PP71。

通过 PWMxCTL0(x=6、7)控制寄存器的 PxCHyMOD<3:0>(x=6、7， y=0、1、2、3)可以使能或禁止 PWMx(x=6、7)的通道功能。在 PWM 自由模式和 PWM 协同模式下，通道 0/1/2 共用 Tx(x=2、5)作为计数时基，通道 3 用 Tx(x=4、6)作为计数时基；在 PWM 单时基



模式下，通道 0/1/2/3 共用  $T_x(x=2、5)$  作为计数时基。

每个  $PWM_x(x=6、7)$  通道的占空比设置寄存器均为 16 位  $\{PWM_xHy:PWM_xLy\}(x=6、7, y=0、1、2、3)$ 。定时器  $T_x(x=2、5)$  可通过  $\{PP_xH:PP_xL\}(x=60、70)$  设置脉冲周期，定时器  $T_x(x=4、6)$  可通过  $\{PP_xH:PP_xL\}(x=61、71)$  设置脉冲周期。

若输出的是 PWM 信号，可通过  $PWM_xPC(x=6、7)$  极性控制寄存器的相应位控制其输出极性（高有效或是低有效）；通过脉冲转向控制可以控制端口输出为 PWM 输出还是作为通用 I/O 口使用。

注意：在 PWM 输出模式下， $T_x(x=2、5)$  或者  $T_y(y=4、6)$  停止计数时， $PWM_x(x=6、7)$  通道 0/1/2/3 输出由极性控制。以上有关寄存器的说明请参考寄存器描述章节。

表 11.3 PWM6 通道 0/1/2/3 输出模式控制及有效引脚

PWM6/7 输出模式	PxCHxMO Dy	有效引脚
PWM 模式	0	通过 PxATRCTL 寄存器中的 PxSTRyH 各位置 1，分别将各引脚配置为 PWM 输出；

注： $x=6/7, y=0/1/2/3$ ，各种输出模式的详细介绍请参考本节相应部分。

### 11.4.1 PWM6/7 相关寄存器

表 11.4 PWM6/7 模块相关寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
11H	T2L	T2 低字节寄存器							
41H	T2H	T2 高 8 位							
42H	PP60H	PWM6 周期 0 寄存器高 8 位							
43H	PWM6L0	捕捉/比较/PWM6 通道 0 占空比寄存器低 8 位							
44H	PWM6H0	捕捉/比较/PWM6 通道 0 占空比寄存器高 8 位							
52H	PP60L	PWM6 周期 0 寄存器低 8 位							
55H	PWM6H1	PWM6 通道 1 占空比寄存器高 8 位							
56H	PWM6L1	PWM6 通道 1 占空比寄存器低 8 位							
62H	PWM6H2	PWM6 通道 2 占空比寄存器高 8 位							
63H	PWM6L2	PWM6 通道 2 占空比寄存器低 8 位							
64H	PWM6H3	PWM6 通道 3 占空比寄存器高 8 位							
65H	PWM6L3	PWM6 通道 3 占空比寄存器低 8 位							
57H	PWM6CTL0	-	-	-	-	P6MOD3	P6MOD2	P6MOD1	P6MOD0
5EH	PWM6CTL2	P6FUSES	-	P6UDEVT3	P6UDEVT2	P6UDEVT1	P6UDEVT0	P6UDEN1	P6UDEN0
5DH	P6ATRCTL	P6STR3H	-	P6STR2H	-	P6STR1H	-	P6STR0H	-
60H	PWM6PC	P6PC3H	-	P6PC2H	-	P6PC1H	-	P6PC0H	-
51H	T2CTL1	T2MOD1	T2MOD0	T2OFS1	T2OFS0	T2TRIGEN	T2CLK1	T2CLK0	T2DIR
12H	T2CTL0	-	T2CKBS3	T2CKBS2	T2CKBS1	T2CKBS0	T2ON	T2CKPS1	T2CKPS0
151H	T4CTL1	T4MOD1	T4MOD0	T4OFS1	T4OFS0	T4TRIGEN	T4CLK1	T4CLK0	T4DIR
112H	T4CTL0	T4SLPEN	T4CKBS3	T4CKBS2	T4CKBS1	T4CKBS0	T4ON	T4CKPS1	T4CKPS0
111H	T4L	T4 低字节寄存器							
141H	T4H	T4 高 8 位							
142H	PP61H	PWM6 周期 1 寄存器高 8 位							

152H	PP61L	PWM6 周期 1 寄存器低 8 位							
552H	T5L	T5 低字节寄存器							
551H	T5H	T5 高 8 位							
50CH	PP70H	PWM7 周期 0 寄存器高 8 位							
55AH	PWM7L0	捕捉/比较/PWM7 通道 0 占空比寄存器低 8 位							
55BH	PWM7H0	捕捉/比较/PWM7 通道 0 占空比寄存器高 8 位							
50DH	PP70L	PWM7 周期 0 寄存器低 8 位							
55DH	PWM7H1	PWM7 通道 1 占空比寄存器高 8 位							
55CH	PWM7L1	PWM7 通道 1 占空比寄存器低 8 位							
55FH	PWM7H2	PWM7 通道 2 占空比寄存器高 8 位							
55EH	PWM7L2	PWM7 通道 2 占空比寄存器低 8 位							
56BH	PWM7H3	PWM7 通道 3 占空比寄存器高 8 位							
56AH	PWM7L3	PWM7 通道 3 占空比寄存器低 8 位							
510H	PWM7CTL0	-	-	-	-	P7MOD3	P7MOD2	P7MOD1	P7MOD0
512H	PWM7CTL2	P7FUSES	-	P7UDEVT3	P7UDEVT2	P7UDEVT1	P7UDEVT0	P7UDEN1	P7UDEN0
509H	P7ATRCTL	P7STR3H	-	P7STR2H	-	P7STR1H	-	P7STR0H	-
56DH	PWM7PC	P7PC3H	-	P7PC2H	-	P7PC1H	-	P7PC0H	-
515H	T5CTL1	T5MOD1	T5MOD0	T5OFS1	T5OFS0	T5TRIGEN	T5CLK1	T5CLK0	T5DIR
550H	T5CTL0	-	T5CKBS3	T5CKBS2	T5CKBS1	T5CKBS0	T5ON	T5CKPS1	T5CKPS0
553H	T6CTL1	T6MOD1	T6MOD0	T6OFS1	T6OFS0	T6TRIGEN	T6CLK1	T6CLK0	T6DIR
554H	T6CTL0	T6SLPEN	T6CKBS3	T6CKBS2	T6CKBS1	T6CKBS0	T6ON	T6CKPS1	T6CKPS0
556H	T6L	T6 低字节寄存器							
555H	T6H	T6 高 8 位							
50FH	PP71L	PWM7 周期 1 寄存器高 8 位							
50EH	PP71H	PWM7 周期 1 寄存器低 8 位							

### 11.4.1.1 PWM6CTL2 更新控制寄存器

**寄存器:** PWM6CTL2: PWM6更新控制寄存器(地址:5EH)

	bit7						bit0	
复位值 1000 0000	P6FUSES	保留	P6UDEVT3 3	P6UDEVT2 2	P6UDEVT1 1	P6UDEVT0 0	P6UDEN1	P6UDEN0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**P6FUSES:** 由 P6FUSES 位控制 PWM6 和 PWM7 输出引脚

0 = 不由 FUSE 控制 PWM6 和 PWM7 输出引脚

1 = 由 FUSE 控制 PWM6 和 PWM7 输出引脚

**P6UDEVT3:** 更新事件控制 3 (在 P6UDEN1=1 时有效)

0 = 当定时器 T4 为 0 时更新以 T4 为时基的占空比、周期寄存器到缓冲器中

1 = 产生更新事件, 更新 T4 为时基的占空比、周期寄存器到缓冲器中, 并将定时器清零

**P6UDEVT2:** 更新事件控制 2 (在 P6UDEN1=1 时有效)

0 = 当定时器 T4 为 0 时更新 T4 为时基的极性控制寄存器

1 = 立即更新 T4 为时基的极性控制寄存器到缓冲器中

**P6UDEVT1:** 更新事件控制 1 (在 P6UDEN0=1 时有效)

0 = 当定时器 T2 为 0 时更新以 T2 为时基的占空比、周期寄存器到缓冲器中

1 = 产生更新事件，更新以 T2 为时基的占空比、周期控制寄存器到缓冲器中，并将定时器清零

P6UDEVT0: 更新事件控制 0（在 P6UDEN0=1 时有效）

0 = 当定时器 T2 为 0 时更新以 T2 为时基的极性控制寄存器到缓冲器中

1 = 立即更新以 T2 为时基的极性控制到缓冲器中

P6UDEN1: 更新使能 1（控制 T4）

1 = 允许以 T4 为时基的占空比、周期、极性控制寄存器更新

0 = 禁止以 T4 为时基占空比、周期、极性控制寄存器更新

P6UDEN0: 更新使能 0（控制 T2）

1 = 允许以 T2 为时基的占空比、周期、极性控制寄存器更新

0 = 禁止以 T2 为时基的占空比、周期、极性控制寄存器更新

### 11.4.1.2 PWM7CTL2 更新控制寄存器

**寄存器:** PWM7CTL2: PWM7更新控制寄存器(地址:512H)

		bit7				bit0		
复位值	P7FUSES	-	P7UDEVT3	P7UDEVT2	P7UDEVT1	P7UDEVT0	P7UDEN1	P7UDEN0
1000 0000			3	2	1	0		
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P7FUSES: 由 P7FUSES 位控制 PWM7 输出引脚

0 = 不由 FUSE 控制 PWM7 输出引脚

1 = 由 FUSE 控制 PWM7 输出引脚

P7UDEVT3: 更新事件控制 3（在 P7UDEN1=1 时有效）

0 = 当定时器 T6 为 0 时更新以 T6 为时基的占空比、周期寄存器到缓冲器中

1 = 产生更新事件，更新 T6 为时基的占空比、周期寄存器到缓冲器中，并将定时器清零

P7UDEVT2: 更新事件控制 2（在 P7UDEN1=1 时有效）

0 = 当定时器 T6 为 0 时更新 T6 为时基的极性控制寄存器

1 = 立即更新 T6 为时基的极性控制寄存器到缓冲器中

P7UDEVT1: 更新事件控制 1（在 P7UDEN0=1 时有效）

0 = 当定时器 T5 为 0 时更新以 T5 为时基的占空比、周期寄存器到缓冲器中

1 = 产生更新事件，更新以 T5 为时基的占空比、周期寄存器到缓冲器中，并将定时器清零

P7UDEVT0: 更新事件控制 0（在 P7UDEN0=1 时有效）

0 = 当定时器 T5 为 0 时更新以 T5 为时基的极性控制寄存器到缓冲器中

1 = 立即更新以 T5 为时基的极性控制寄存器到缓冲器中

P7UDEN1: 更新使能 1（控制 T6）

1 = 允许以 T6 为时基的占空比、周期、极性控制寄存器更新

0 = 禁止以 T6 为时基的占空比、周期、极性控制寄存器更新

P7UDEN0: 更新使能 0（控制 T5）

1 = 允许以 T5 为时基占空比、周期、极性控制寄存器更新

0 = 禁止以 T5 为时基占空比、周期、极性控制寄存器更新

## 11.4.1.3 PWM6PC 极性控制寄存器

寄存器: PWM6PC: PWM6极性控制寄存器(地址:60H)

	bit7						bit0	
复位值 0000 0000	P6PC3H	-	P6PC2H	-	P6PC1H	-	P6PC0H	-
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P6PC3H: 当P6OC3H为0时, 通道3中P6CH3H端口输出极性控制

0 = 输出为高电平有效

1 = 输出为低电平有效

P6PC2H: 当P6OC2H为0时, 通道2中P6CH2H端口输出极性控制

0 = 输出为高电平有效

1 = 输出为低电平有效

P6PC1H: 当P6OC1H为0时, 通道1中P6CH1H端口输出极性控制

0 = 输出为高电平有效

1 = 输出为低电平有效

P6PC0H: 当P6OC0H为0时, 通道0中P6CH0H端口输出极性控制

0 = 输出为高电平有效

1 = 输出为低电平有效

## 11.4.1.4 PWM7PC 极性控制寄存器

寄存器: PWM7PC: PWM7极性控制寄存器(地址:56DH)

	bit7						bit0	
复位值 0000 0000	P7PC3H	-	P7PC2H	-	P7PC1H	-	P7PC0H	-
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P7PC3H: 当P7OC3H为0时, 通道3中P7CH3H端口输出极性控制

0 = 输出为高电平有效

1 = 输出为低电平有效

P7PC2H: 当P7OC2H为0时, 通道2中P7CH2H端口输出极性控制

0 = 输出为高电平有效

1 = 输出为低电平有效

P7PC1H: 当P7OC1H为0时, 通道1中P7CH1H端口输出极性控制

0 = 输出为高电平有效

1 = 输出为低电平有效

P7PC0H: 当P7OC0H为0时, 通道0中P7CH0H端口输出极性控制

0 = 输出为高电平有效

1 = 输出为低电平有效

## 11.4.1.5 P6ATRCTL 寄存器

寄存器: P6ATRCTL: 脉冲转向控制寄存器(地址: 5DH)

	bit7						bit0	
复位值 0000 0000	P6STR3H	-	P6STR2H	-	P6STR1H	-	P6STR0H	-
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- P6STR3H: P6CH3H 转向使能位  
1 = P6CH3H 引脚输出为 PWM 输出  
0 = P6CH3H 引脚被分配为端口引脚
- P6STR2H: P6CH2H 转向使能位  
1 = P6CH2H 引脚输出为 PWM 输出  
0 = P6CH2H 引脚被分配为端口引脚
- P6STR1H: P6CH1H 转向使能位  
1 = P6CH1H 引脚输出为 PWM 输出  
0 = P6CH1H 引脚被分配为端口引脚
- P6STR0H: P6CH0H 转向使能位  
1 = P6CH0H 引脚输出为 PWM 输出  
0 = P6CH0H 引脚被分配为端口引脚

## 11.4.1.6 P7ATRCTL 寄存器

寄存器: P7ATRCTL: 脉冲转向控制寄存器(地址: 509H)

	bit7						bit0	
复位值 0000 0000	P7STR3H	-	P7STR2H	-	P7STR1H	-	P7STR0H	-
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- P7STR3H: P7CH3H 转向使能位  
1 = P7CH3H 引脚输出为 PWM 输出  
0 = P7CH3H 引脚被分配为端口引脚
- P7STR2H: P7CH2H 转向使能位  
1 = P7CH2H 引脚输出为 PWM 输出  
0 = P7CH2H 引脚被分配为端口引脚
- P7STR1H: P7CH1H 转向使能位  
1 = P7CH1H 引脚输出为 PWM 输出  
0 = P7CH1H 引脚被分配为端口引脚
- P7STR0H: P7CH0H 转向使能位  
1 = P7CH0H 引脚输出为 PWM 输出  
0 = P7CH0H 引脚被分配为端口引脚

## 11.4.2 PWM6/7 的周期、占空比及分辨率

### 11.4.2.1 PWM6/7 周期

PWM6 以 T2/T4 作为计数时基，PWM7 以 T5/T6 作为计数时基。T2/T5 的周期分别通过 16 位的周期寄存器<PP60H:PP60L>和<PP70H:PP70L>进行设置，T4/T6 的周期分别通过 16 位的周期寄存器<PP61H:PP61L>和<PP71H:PP71L>进行设置，其值均可设置为 0~65535。当 PWM6/7 为自由模式或协同模式时，PWM6 通道 0/1/2 使用 T2 作为时基，通道 3 使用 T4 作为时基；PWM7 通道 0/1/2 使用 T5 作为时基，通道 3 使用 T6 作为时基。当 PWM6/7 为单时基模式时，PWM6 通道 0/1/2/3 均使用 T2 作为计数时基，PWM7 通道 0/1/2/3 均使用 T5 作为计数时基。

PWM6/7 边沿对齐的周期通过下式进行计算。

公式 11.1: 计算边沿对齐下周期

$$\text{PWM周期}=(\langle\text{PP60H:PP60L}\rangle+1)\cdot T\cdot(\text{T2预分频比})$$

$$\text{PWM周期}=(\langle\text{PP61H:PP61L}\rangle+1)\cdot T\cdot(\text{T4预分频比})$$

$$\text{PWM周期}=(\langle\text{PP70H:PP70L}\rangle+1)\cdot T\cdot(\text{T5预分频比})$$

$$\text{PWM周期}=(\langle\text{PP71H:PP71L}\rangle+1)\cdot T\cdot(\text{T6预分频比})$$

注：T 为计数时基使用的时钟周期。

周期寄存器为双缓冲模式，只有在 PxUDEN0/1 (x=6,7) 为 1 时才能更新。其中 PxUDEN0 (x=6,7) 控制以 T2/T5 为时基的周期更新，UDEN1 控制以 T4/T6 为时基的周期更新。

当 PxUDEN0/1 (x=6,7) 为 1 时，通过 PyUDEVTx (x=1,3; y=6,7) 可以控制周期更新时机。其中 PxUDEVT1 (x=6,7) 控制以 T2/T5 为时基的周期更新，PxUDEVT3 (x=6,7) 控制以 T4/T6 为时基的周期更新。

若 PxUDEVT1/3 (x=6,7) 为 0，则周期可以在 Tx (x=2,4,5,6) 为 0 时更新到周期缓冲器中；若 PxUDEVT1/3 (x=6,7) 为 1 时，则立即更新周期到周期缓冲器中，且 Tx (x=2,4,5,6) 被清零。在下一次立即更新前，需要软件清零 PxUDEVT1/3 (x=6,7)，再置 1 才能开启新一次的立即更新。

(注意：当 PWM 使能，Tx (x=2,4,5,6) 启动时周期寄存器立即更新。)

### 11.4.2.2 PWM6/7 占空比

PWM6/7 有 4 个 PWM 通道，每个 PWM 通道均有 2 个占空比设置寄存器，每个占空比设置寄存器均为 16 位，通过<PWMxHy:PWMxLy>(x=6/7,y=0/1/2/3)进行设置，PWMxHy 为占空比的高 8 位，PWMxLy 为低 8 位。

在边沿对齐模式下，脉冲宽度和占空比通过计算 12.3~12.4 计算。

公式 11.2 : 计算边沿对齐下脉冲宽度

$$\text{脉冲宽度} = (\text{PWM6Hx}:\text{PWM6Lx}) \cdot T \cdot (\text{T2/T4预分频比})$$

$$\text{脉冲宽度} = (\text{PWM7Hx}:\text{PWM7Lx}) \cdot T \cdot (\text{T5/T6预分频比})$$

公式 11.3 : 计算边沿对齐下占空比

$$\text{占空比} = \frac{\text{脉冲宽度}}{\text{PWM周期}} = \frac{\text{PWM6Hx}:\text{PWM6Lx}}{\langle \text{PPyH}:\text{PPyL} \rangle + 1}$$

$$\text{占空比} = \frac{\text{脉冲宽度}}{\text{PWM周期}} = \frac{\text{PWM7Hx}:\text{PWM7Lx}}{\langle \text{PPzH}:\text{PPzL} \rangle + 1}$$

注: (1) T 为计数时基使用的时钟周期。

(2) x=0,1,2,3

(3) y=60,61

(4) z=70,71

由上述公式可知, 占空比为 0%和 100%满足条件如下:

➤ 边沿对齐模式:

0%: 占空比寄存器为 0

100%: 占空比寄存器为大于等于  $(\langle \text{PPxH}:\text{PPxL} \rangle + 1)$  (x=60/61/70/71)

➤ 在半桥模式下, 占空比为 0%或者 100%时, 只有在边界 PWM 变化时插入死区, 否则不插入死区。

占空比寄存器为双缓冲模式, 只有在  $\text{PxUDEN0/1}(x=6,7)$  为 1 时才能更新。其中  $\text{PxUDEN0}(x=6,7)$  控制以 T2/T5 为时基的占空比更新,  $\text{PxUDEN1}(x=6,7)$  控制以 T4/T6 为时基的占空比更新。

当  $\text{PxUDEN0/1}(x=6,7)$  为 1 时, 通过  $\text{PxUDEVTy}$  (x=6,7;y=0,2) 可以控制占空比更新时机。其中  $\text{PxUDEVT0}(x=6,7)$  控制以 T2/T5 为时基的占空比更新,  $\text{PxUDEVT2}(x=6,7)$  控制以 T4/T6 为时基的占空比更新。

若  $\text{PxUDEVT0/2}(x=6,7)$  为 0, 则占空比可以在  $\text{Tx}(x=2,4,5,6)$  为 0 时更新到占空比缓冲器中; 若  $\text{PxUDEVT0/2}(x=6,7)$  为 1 时, 则立即更新占空比到占空比缓冲器中, 且 T2/T5 被清零。在下次立即更新前, 需要软件清零  $\text{PxUDEVT0/2}(x=6,7)$ , 再置 1 才能开启新一次的立即更新。

(注意: 当 PWM 使能,  $\text{Tx}(x=2,4,5,6)$  启动时占空比寄存器立即更新。)

### 11.4.2.3 PWM6/7 分辨率(最小占空比)

中当  $\langle \text{PPxH}:\text{PPxL} \rangle$  为 65535 时, PWM6/7 的最大分辨率为 16 位。分辨率的计算公式如下式所示。

$$\text{分辨率} = \frac{\text{Log}(\langle \text{PPxH}:\text{PPxL} \rangle + 1)}{\log 2}$$

注: x=60,61,70,71

#### 11.4.2.4 PWM6/7 中断

当 PWM6/7 调制输出满一个周期时, 将中断标志位  $TxIF(x=2,4,5,6)$  置 1, 如果 PWM6/7 中断使能, 且 AIE(全局中断允许位)和 PUIE(外设中断允许位)置 1, 程序将进入相应中断。如果 IPEN 和  $PTx(x=2,4,5,6)$  位均置 1, 则为高优先级中断。

#### 11.4.3 边沿对齐 PWM 信号

当  $TxCTL1(x=2, 4, 5, 6)$  控制寄存器中的  $TxMOD<1:0>(x=2, 4, 5, 6)$  计数模式选择位设置成 00 时, 产生的 PWM 信号为边沿对齐的 PWM 信号。在该模式下, PWM 信号的周期由  $<PPxH:PPxL>(x=60/61/70/71)$  周期寄存器决定, 占空比由 16 位的寄存器  $<PWMxHy:PWMxLy>$  决定 ( $x=6,7/y=0,1,2,3$ )。

图 11.4 所示为边沿对齐模式下的 PWM 输出示意图。当 PWM 输出为高有效时, 周期开始时输出高电平, 占空比与定时器匹配时电平变低; 当 PWM 输出低有效时, 周期开始时输出低电平, 占空比与定时器匹配时电平变高。如图 11.4 所示为  $PxCHyH(x=6,7/y=0,1,2,3)$  引脚为在高有效时输出。

如果占空比寄存器设置成 0, 那么在整个 PWM 周期都输出低电平。如果占空比寄存器设置大于等于周期寄存器  $<PPxH:PPxL>+1(x=60/61/70/71)$  的值, 那么在整个 PWM 周期都输出高电平。

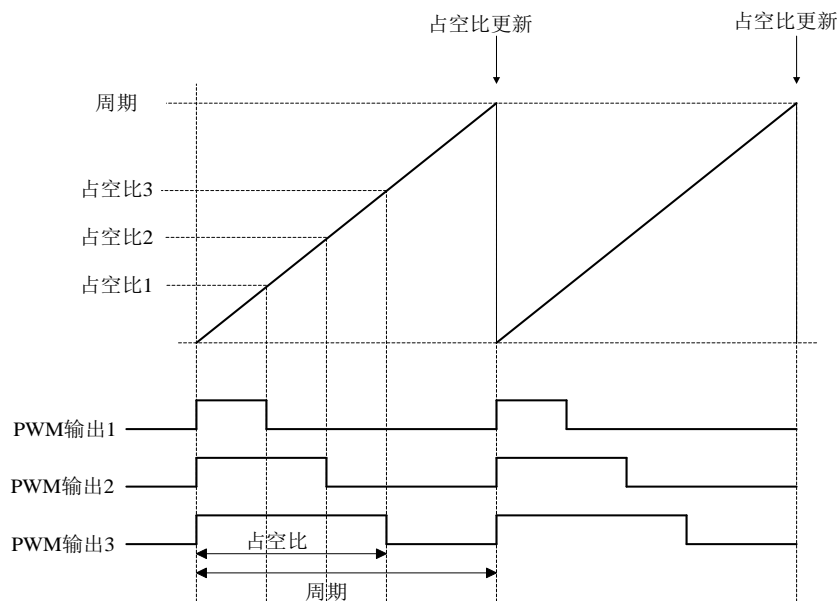


图 11.4 边沿对齐 PWM (高电平有效)

#### 11.4.4 PWM 信号产生和中断

通过设置  $TxCTL1$  控制寄存器中的  $TxMOD<1:0>$  可以选择  $Tx$  的计数方式, 从而产生边沿对齐的 PWM 信号。当  $Tx$  向上计数时, 产生边沿对齐的 PWM 信号。

通过设置  $TxCTL1$  寄存器中的  $TxMOD<1:0>$  位控制选择溢出中断, 则可以选择在不同的时机触发 PWM 中断操作。当  $TxMOD<1:0>=00$  时 (向上计数模式), 是上溢时产生中断



标志，也就是在<PPxH:PPxL>(x=60/61/70/71)和Tx(x=2/4/5/6)相等的时候产生一个中断信号。当TxMOD<1:0>=01时(向上向下计数模式)，是上溢产生中断标志，也就是在计数器Tx向上计数到周期值的时候产生一个中断信号。当TxMOD<1:0>=10时(向上向下计数模式)，是下溢产生中断标志，是计数器Tx向下递减为0的时候产生一个中断信号。当TxMOD<1:0>=11时(向上向下计数模式)，是上溢-下溢中断，在这种状态下可以在<PPxH:PPxL>和计数器Tx相等以及计数器Tx向下计数到0时都产生中断信号。在向上计数模式时，使用上溢中断，在向上-向下计数模式时可以使用3种中断产生方式。

#### 11.4.5 PWM 输出设置

如果PWM相应引脚为模拟数字共用引脚，应将该引脚设置为通用数字I/O口。

通过寄存器脉冲转向寄存器PxATRCTL(x=6/7)脉冲转向控制寄存器可将相应的引脚设置成PWM6/7输出引脚或者通用IO引脚。通过该设置，可以在使用PWM6/7时，将未使用的引脚配置成通用IO口。

#### 11.4.6 PWM 更新锁定

通过设置PWMxCTL2更新控制寄存器的PxUDEN0/1(x=6/7)可以使能或禁止占空比、周期、极性控制的更新。其中PxUDEN0/PxUDEVT0/PxUDEVT1(x=6/7)用于控制以Tx(x=2/5)为时基的通道的占空比、周期、极性控制寄存器，PxUDEN1/PxUDEVT2/PxUDEVT3(x=6/7)用于控制以Tx(x=4/6)为时基的通道的占空比、周期、极性控制寄存器。

当PWMxCTL2(x=6/7)寄存器的PxUDEN0/1(x=6/7)为0时，占空比、周期、极性控制不能更新，这使得用户能够在新值生效前将所需要的值写入到相应的寄存器中。当PxUDEN0/1为1时，可以通过PxUDEVT0/1/2/3来控制占空比、周期、极性控制的更新到缓冲中。PxUDEVT1/3控制周期和占空比的更新，PxUDEVT0/2控制极性控制寄存器的更新。若PxUDEVT1/3为0，则周期和占空比可以在相应的定时器为0时更新到缓冲中；若PxUDEVT1/3为1时，则立即更新周期和占空比到缓冲中，同时定时器被清0。若PxUDEVT0/2为0，则极性控制寄存器可以在定时器为0时更新到缓冲中；若PxUDEVT0/2为1时，则立即更新极性控制寄存器到缓冲中，此时定时器继续计数，不影响占空比和周期的完整性。

注意，当PxUDEVT1/3为1时，会清零定时器，此时若PxUDEVT0/2为0，也会引起极性控制的更新。

PxUDEVT0/1/2/3置1后必须软件清零再置1才能使能下一次的立即更新控制。

#### 11.4.7 PWM 复位控制

PWM6/7的复位控制使得在复位状态下，可以通过将配置字位中的PWMPIN设置成I/O口控制或是选择引脚的有效极性状态。引脚的极性状态可以通过控制配置位中的HPOL输出所需要的极性。配置HPOL可以设置PxCHyH(x=6,7/y=0,1,2,3)的极性

在正常工作时，可以通过PWMxCTL2(x=6/7)更新控制寄存器的PxFUSES位来关闭配置字位对PWM端口的控制，以实现PWM模块对端口控制。默认情况下是使能配置字位控制PWM端口输出。

## 12 CCP8/9(捕捉/比较/PWM3、4)模块

KF8A100 单片机内部集成有 CCP8/9 模块。CCP8/9 模块是指捕捉/比较/脉宽调制模块，可以提供外部信号捕捉、内部比较输出以及 PWM 输出三种功能。

### 12.1 捕捉/比较相关寄存器

表 12.1 捕捉/比较模块的相关寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
448H	PWM3L0	捕捉/比较/PWM3 占空比寄存器低 8 位							
449H	PWM3L1	捕捉/比较/PWM3 占空比寄存器高 8 位							
444H	PWM3CTL0	P3M1	P3M0	-	-	PWM3M3	PWM3M2	PWM3M1	PWM3M0
457H	PWM4L0	捕捉/比较/PWM4 占空比寄存器低 8 位							
458H	PWM4L1	捕捉/比较/PWM4 占空比寄存器高 8 位							
453H	PWM4CTL0	P4M1	P4M0	-	-	PWM4M4	PWM4M2	PWM4M1	PWM4M0

#### 12.1.1 PWM3CTL0 寄存器

寄存器: PWM3CTL0: PWM3控制寄存器0(地址: 444H)

复位值	bit7				bit0			
0000 0000	P3M1	P3M0	保留	保留	PWM3M3	PWM3M2	PWM3M1	PWM3M0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P3M<1:0>: PWM3 输出配置位

00 = 单输出模式; 默认 P3A 配置为 PWM 输出, P3B、P3C 和 P3D 为端口引脚; 可通过将 PATRCTL<3:0>各位置 1, 分别将 P3A、P3B、P3C 和 P3D 配置为 PWM 输出; PWM3 最多可提供 3 路 PWM 输出

01 = 全桥正向输出模式; P3D 配置为 PWM 调制输出, P3A 为有效电平, P3B 和 P3C 为无效电平

10 = 半桥输出模式; P3A 和 P3B 配置为调制输出; P3C 和 P3D 被分配为端口引脚, 此模式带有死区控制功能

11 = 全桥反向输出模式; P3B 配置为调制输出; P3C 为有效电平; P3A 和 P3 为无效电平

PWM3M<3:0>: PWM3 模块的模式选择位

0000 = PWM 关闭

0001 = 未使用 (保留)

0010 = 比较模式, 匹配时输出电平翻转 (CCP9IF 置 1)

0011 = 未使用 (保留)

0100 = 捕捉模式, 在每个下降沿发生捕捉

- 0101 = 捕捉模式，在每个上升沿发生捕捉
- 0110 = 捕捉模式，每 4 个上升沿发生捕捉
- 0111 = 捕捉模式，每 16 个上升沿发生捕捉
- 1000 = 比较模式，比较匹配时输出高电平（CCP9IF 置 1）
- 1001 = 比较模式，比较匹配时输出低电平（CCP9IF 置 1）
- 1010 = 比较模式，比较匹配时产生软件中断（CCP9IF 置 1，P3A 引脚不受影响）

1011 = 比较模式，触发特殊事件（CCP9IF 位置 1， CCP 复位 T1，如果 ADC 使能将启动 AD 转换。）

- 1100 = PWM模式；P3A 和P3C、P3B 和P3D 均为高电平有效
- 1101 = PWM模式；P3A 和P3C 为高电平有效；P3B和P3D为低电平有效
- 1110 = PWM模式；P3A 和P3C 为低电平有效；P3B和P3D为高电平有效
- 1111 = PWM模式；P3A 和P3C、P3B 和P3D 均为低电平有效
- 其他 = 系统保留

### 12.1.2 PWM4CTL0 寄存器

**寄存器:** PWM4CTL0: PWM4控制寄存器0(地址: 453H)

bit7				bit0				
复位值 0000 0000	P4M1	P4M0	保留	保留	PWM4M3	PWM4M2	PWM4M1	PWM4M0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**P4M<1:0>:** PWM4 输出配置位

- 00 = 单输出模式；默认 P4A 配置为 PWM 输出，P4B、P4C 和 P4D 为端口脚；可通过将 PATRCTL<3:0>各位置 1，分别将 P4A、P4B、P4C 和 P4D 配置为 PWM 输出；PWM4 最多可提供 4 路 PWM 输出
- 01 = 全桥正向输出模式；P4D 配置为 PWM 调制输出，P4A 为有效电平，P4B 和 P4C 为无效电平
- 10 = 半桥输出模式；P4A 和 P4B 配置为调制输出；P4C 和 P4D 被分配为端口引脚，此模式带有死区控制功能
- 11 = 全桥反向输出模式；P4B 配置为调制输出；P4C 为有效电平；P4A 和 P4D 为无效电平

**PWM4M<3:0>:** PWM4 模块的模式选择位

- 0000 = PWM 关闭
- 0001 = 未使用（保留）
- 0010 = 比较模式，匹配时输出电平翻转（CCP9IF 置 1）
- 0011 = 未使用（保留）
- 0100 = 捕捉模式，在每个下降沿发生捕捉
- 0101 = 捕捉模式，在每个上升沿发生捕捉
- 0110 = 捕捉模式，每 4 个上升沿发生捕捉
- 0111 = 捕捉模式，每 16 个上升沿发生捕捉
- 1000 = 比较模式，比较匹配时输出高电平（CCP9IF 置 1）
- 1001 = 比较模式，比较匹配时输出低电平（CCP9IF 置 1）
- 1010 = 比较模式，比较匹配时产生软件中断（CCP9IF 置 1，P4A 引脚不受影

响)

1011 = 比较模式，触发特殊事件（CCPxIF 位置 1， CCP 复位 T1， 如果 ADC 使能将启动 AD 转换。）

1100 = PWM模式； P4A 和P4C、 P4B 和P4D 均为高电平有效

1101 = PWM模式； P4A 和P4C 为高电平有效； P4B和P4D为低电平有效

1110 = PWM模式； P4A 和P4C 为低电平有效； P4B和P4D为高电平有效

1111 = PWM模式； P4A 和P4C、 P4B 和P4D 均为低电平有效

其他 = 系统保留

## 12.2 捕捉模式

CCP8和CCP9完全一致，在捕捉模式下，当对应的CCPxIN（x=8/9，下同）引脚发生事件时，PWMMyH0:PWMMyL0（y=3/4，下同）这对寄存器捕捉T1H和T1L寄存器的16位值；原理框图如下图所示：

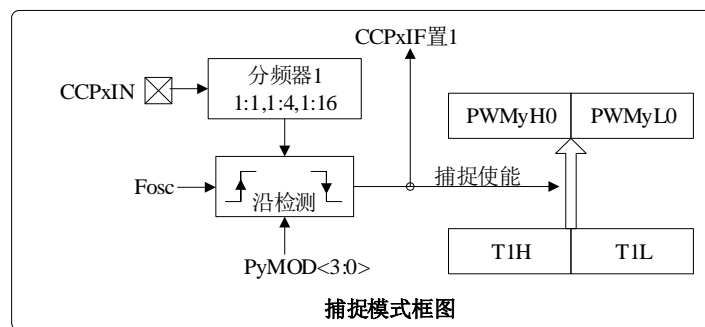


图 12.1 捕捉模式原理框图

触发捕捉的事件可被定义为以下四者之一，并且由 PWMMyCTL0 寄存器中的模式选择位 PyMOD<3:0>选择事件类型位配置：

- ◆ 0100 = 每个下降沿
- ◆ 0101 = 每个上升沿
- ◆ 0110 = 每4 个上升沿
- ◆ 0111 = 每16个上升沿

在捕捉模式下，将 CCPxIN 引脚配置为数字输入。

当一个捕捉发生时，硬件自动中断请求标志位EIF8寄存器中的CCPxIF置1；它必须用软件清零。注意如果在PWMMyH0和PWMMyL0这对寄存器中的值被读取之前发生另一次捕捉，那么之前捕捉的值将被新捕捉的值覆盖。

当捕捉模式改变时，可能会产生错误的捕捉中断。用户应该在捕捉模式改变之前保持 EIE8 寄存器中的 CCPxIE 中断允许位清零以避免产生误中断。在捕捉模式发生任何改变之后也应清零 EIF8 寄存器中的中断标志位 CCPxIF。

PWMMyCTL0 寄存器中的 PyMOD<3:0> 位指定了 4 种预分频器设置。每当关闭 CCP 模块或禁止捕捉模式时，就会清零预分频器计数器。这意味着任何复位都将清零预分频计数器。

从一种捕捉预分频比切换到另一种捕捉预分频比不会将预分频计数器清零，但可能会产生误中断，因此要避免出现这种不期望的操作，应在改变预分频比前通过将 PWMMyCTL0 寄存器清零关闭该模块。

注：T1必须运行在定时模式或同步计数模式下CCP模块才能使用捕捉功能。在异步计数模式下无法进行捕捉操作。

## 12.3 比较模式

在比较模式下，16位的PwMyH0:PwMyL0寄存器的值将不断与T1寄存器的值相比较。当两者匹配时，CCP模块可能会出现以下几种情况：

- ◆ CCPxOUT引脚的输出电平翻转
- ◆ CCPxOUT引脚输出高电平
- ◆ CCPxOUT引脚输出低电平
- ◆ 产生软件中断触发信号
- ◆ 产生特殊事件触发信号

CCPxOUT引脚的动作取决于PwMyCTL0寄存器中PyMOD<3:0>控制位的值。

所有比较模式都会产生中断。原理图如下图所示：

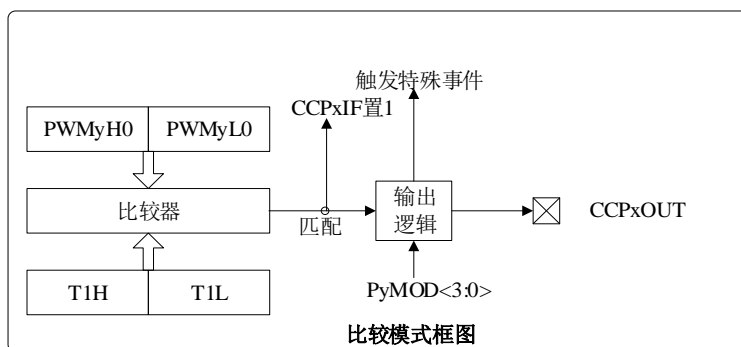


图 12.2 比较模式原理框图

在比较模式下，T1必须运行在定时模式或同步计数模式，计数周期至少大于一个机器周期。在异步计数模式下，可能无法进行比较操作。

- 5) 当选择输出电平翻转模式（PyMOD<3:0> = 0010）时，比较匹配时，CCPxOUT引脚的输出电平翻转，并将CCPxIF置1。
- 6) 当选择普通比较模式时（PyMOD<3:0> = 1000或1001）时，比较匹配时，CCPxOUT引脚输出高电平或者低电平，并且将CCPxIF置1。
- 7) 当选择了软件中断触发模式时（PyMOD<3:0> = 1010）时，比较匹配时，将CCPxIF置1，但是CCP模块不会控制CCPxOUT引脚。
- 8) 当选择了特殊事件触发模式（PyMOD<3:0> = 1011）时，比较匹配时，CCP会立即产生特殊事件触发输出，将CCPxIF置1，此时如果ADC已使能，将启动AD转换。但T1H/T1L寄存器不会立即复位，直到T1计数脉冲的下一个上升沿才复位。从而使PwMyH0/PwMyL0寄存器实际上成为了定时器1(T1)的16位可编程周期寄存器。

## 12.4 PWM3/4 模块

PWM4和PWM3完全一致。PwMy（y=3/4，以下涉及到PWM3和PWM4寄存器或相关内容都用y代替）为可产生最高16位分辨率的PWM模块，16位占空比寄存器<PwMyL1:

PWM<sub>y</sub>L0>组成，16位周期寄存器由{PPyH:PPyL}组成，Tx (x=8/9)计数模式为16位。

PWM<sub>y</sub>为带有死区控制功能的增强型PWM。占空比设置寄存器为16位(PWM<sub>y</sub>L1和PWM<sub>y</sub>L0)，PWM<sub>y</sub>最多可在4个不同的引脚(PyA、PyB、PyC和PyD)输出PWM信号，分辨率最高16位。PWM<sub>y</sub>有4种输出模式：单输出、半桥输出、全桥正向输出和全桥反向输出，通过寄存器PWM<sub>y</sub>CTL0中的PyM<1:0>位选择4种输出模式之一，通过PWM<sub>y</sub>M<1:0>位设置各引脚的有效电平(可设置为高电平有效和低电平有效)。PWM3/4的原理框图如下图所示。

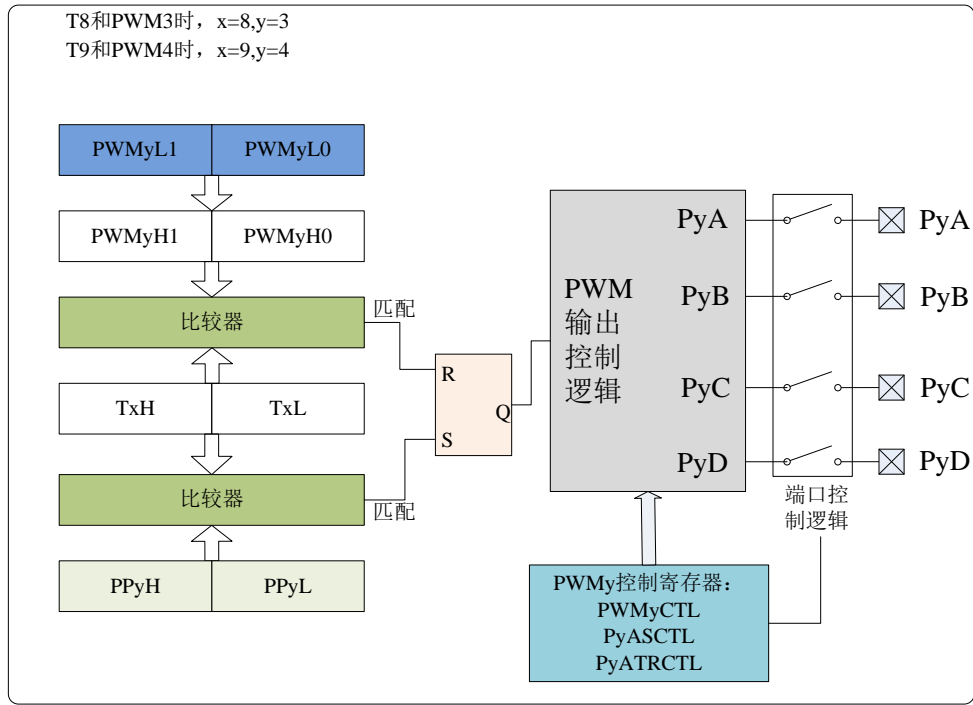


图 12.3 PWM3/4 原理框图

表 12.2 PWM3/4 输出模式控制及有效引脚

PWM <sub>y</sub> 输出模式	PyM<1:0>	有效引脚
单输出模式	00	默认将 PyA 配置为 PWM 输出，PyB、PyC 和 PyD 配置为端口引脚；可通过将 PyATRCTL<3:0>各位置 1，分别将 PyA、PyB、PyC 和 PyD 配置为 PWM 输出；PWM <sub>y</sub> 最多可在 4 个引脚输出 PWM 信号。
半桥输出模式	10	PyA 和 PyB 配置为调制输出；PyC 和 PyD 配置为端口引脚；半桥输出模式带有死区控制功能
全桥正向输出模式	01	PyD 配置为 PWM 调制输出；PyA 为有效电平；PyB 和 PyC 为无效电平
全桥反向输出模式	11	PyB 配置为 PWM 调制输出；PyC 为有效电平，PyA 和 PyD 为无效电平

注：各种输出模式的详细介绍请参考本节相应部分。

## 12.5 PWM3/4 相关寄存器

表 12.3 与 PWM3/4 相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
442H	PP3H	PWM3 周期设置寄存器高 8 位							
443H	PP3L	PWM3 周期设置寄存器低 8 位							
448H	PWM3L0	PWM3 占空比设置寄存器低 8 位							
446H	PWM3H0	PWM3 占空比缓冲寄存器高 8 位							
449H	PWM3L1	PWM3 占空比设置寄存器高 8 位							
447H	PWM3H1	PWM3 占空比缓冲寄存器高 8 位							
444H	PWM3CTL0	P3M1	P3M0	-	-	PWM3M3	PWM3M2	PWM3M1	PWM3M0
445H	PWM3CTL1	P3RSEN	P3DC6	P3DC5	P3DC4	P3DC3	P3DC2	P3DC1	P3DC0
440H	P3ASCTL	P3ASE	P3ASS2	P3ASS1	P3ASS0	P3SSAC1	P3SSAC0	P3SSBD1	P3SSBD0
441H	P3ATRCTL	-	-	-	P3STRSYN C	P3STREND	P3STREN C	P3STRENB	P3STRENA
451H	PP4H	PWM4 周期设置寄存器高 8 位							
452H	PP4L	PWM4 周期设置寄存器低 8 位							
457H	PWM4L0	PWM4 占空比设置寄存器低 8 位							
455H	PWM4H0	PWM4 占空比缓冲寄存器高 8 位							
458H	PWM4L1	PWM4 占空比设置寄存器高 8 位							
456H	PWM4H1	PWM4 占空比缓冲寄存器高 8 位							
453H	PWM4CTL0	P4M1	P4M0	-	-	PWM4M4	PWM4M2	PWM4M1	PWM4M0
454H	PWM4CTL1	P4RSEN	P4DC6	P4DC5	P4DC4	P4DC3	P4DC2	P4DC1	P4DC0
44FH	P4ASCTL	P4ASE	P4ASS2	P4ASS1	P4ASS0	P4SSAC1	P4SSAC0	P4SSBD1	P4SSBD0
450H	P4ATRCTL	-	-	-	P4STRSYN C	P4STREND	P4STREN C	P4STRENB	P4STRENA

### 12.5.1 PWM4CTL0 寄存器

**寄存器:** PWM4CTL0: PWM4控制寄存器0(地址: 453H)

复位值 0000 0000	bit7							bit0
	P4M1	P4M0	保留	保留	PWM4M3	PWM4M2	PWM4M1	PWM4M0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**P4M<1:0>:** PWM4 输出配置位

- 00 = 单输出模式；默认 P4A 配置为 PWM 输出，P4B、P4C 和 P4D 为端口脚；可通过将 PATRCTL<3:0>各位置 1，分别将 P4A、P4B、P4C 和 P4D 配置为 PWM 输出；PWM4 最多可提供 4 路 PWM 输出
- 01 = 全桥正向输出模式；P4D 配置为 PWM 调制输出，P4A 为有效电平，P4B 和 P4C 为无效电平
- 10 = 半桥输出模式；P4A 和 P4B 配置为调制输出；P4C 和 P4D 被分配为端口引脚，此模式带有死区控制功能
- 11 = 全桥反向输出模式；P4B 配置为调制输出；P4C 为有效电平；P4A 和 P4D 为无效电平

PWM4M<3:0>:PWM4 模块的模式选择位

- 0000 = PWM 关闭
- 0001 = 未使用（保留）
- 0010 = 比较模式，匹配时输出电平翻转（CCP9IF 置 1）
- 0011 = 未使用（保留）
- 0100 = 捕捉模式，在每个下降沿发生捕捉
- 0101 = 捕捉模式，在每个上升沿发生捕捉
- 0110 = 捕捉模式，每 4 个上升沿发生捕捉
- 0111 = 捕捉模式，每 16 个上升沿发生捕捉
- 1000 = 比较模式，比较匹配时输出高电平（CCP9IF 置 1）
- 1001 = 比较模式，比较匹配时输出低电平（CCP9IF 置 1）
- 1010 = 比较模式，比较匹配时产生软件中断（CCP9IF 置 1，P4A 引脚不受影响）
- 1011 = 比较模式，触发特殊事件（CCP9IF 位置 1，CCP 复位 T1，如果 ADC 使能将启动 AD 转换。）
- 1100 = PWM模式；P4A 和P4C、P4B 和P4D 均为高电平有效
- 1101 = PWM模式；P4A 和P4C 为高电平有效；P4B和P4D为低电平有效
- 1110 = PWM模式；P4A 和P4C 为低电平有效；P4B和P4D为高电平有效
- 1111 = PWM模式；P4A 和P4C、P4B 和P4D 均为低电平有效
- 其他 = 系统保留

### 12.5.2 PWM4CTL1 寄存器

寄存器：PWM4CTL1: PWM4控制寄存器1(地址：454H)

		bit7						bit0	
复位值	0000 0000	P4RSEN	P4DC6	P4DC5	P4DC4	P4DC3	P4DC2	P4DC1	P4DC0
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P4RSEN: PWM4 重启使能位

1 = 自动关闭时，一旦关闭事件消失，P4ASE 位将自动清零，PWM 自动重启

0 = 自动关闭时，P4ASE 由软件清零，以重启 PWM

P4DC<6:0>: PWM 死区延时时间设置位，用于设置死区延时的时间，死区延时的时间见死区延时章节。

图注：R=可读 W=可写 -=未用 U=未实现位

### 12.5.3 P4ASCTL 寄存器

寄存器：P4ASCTL: PWM4自动关闭控制寄存器(地址：44FH)

		bit7						bit0	
复位值	0000 0000	P4ASE	P4ASS2	P4ASS1	P4ASS0	P4SSAC1	P4SSAC0	P4SSBD1	P4SSBD0
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W



- P4ASE: 自动关闭事件状态位  
 1 = 发生了关闭事件；四路输出为关闭状态  
 0 = 四路输出正常工作
- P4ASS<2:0>: 自动关闭源选择位  
 000 =禁止自动关断  
 001 =比较器 C1 输出高电平  
 100 =INT 引脚上的 V<sub>IL</sub>  
 101 =INT 引脚上的 V<sub>IL</sub> 或比较器 C1 输出高电平  
 其它 = 未使用
- P4SSAC<1:0>:引脚 P4A 和 P4C 关闭状态控制位  
 00 = 驱动引脚 P4A 和 P4C 为 0  
 01 = 驱动引脚 P4A 和 P4C 为 1  
 1x = 引脚 P4A 和 P4C 为三态
- P4SSBD<1:0>:引脚 P4B 和 P4D 关闭状态控制位  
 00 = 驱动引脚 P4B 和 P4D 为 0  
 01 = 驱动引脚 P4B 和 P4D 为 1  
 1x = 引脚 P4B 和 P4D 为三态

图注：R=可读 W=可写 --未用 U=未实现位

### 12.5.4 P4ATRCTL 寄存器

寄存器：P4ATRCTL: 脉冲转向控制寄存器(地址: 450H)

复位值	bit7								bit0
---0 0001	-	-	-	P4STRSY NC	P4STREN D	P4STREN C	P4STREN B	P4STREN A	
	U	U	U	R/W	R/W	R/W	R/W	R/W	

- P4STRSYNC: 转向同步位  
 1 = 在下一个 PWM 周期发生输出转向更新  
 0 = 在指令周期边界的开始发生输出转向更新
- P4STREND: P4D 转向使能位  
 1 = P4D 引脚输出 PWM 波形，其极性由 PWM4M<1:0>控制  
 0 = P4D 引脚被分配为端口引脚
- P4STRENC: P4C 转向使能位  
 1 = P4C 引脚输出 PWM 波形，其极性由 PWM4M<1:0>控制  
 0 = P4C 引脚被分配为端口引脚
- P4STRENB: P4B 转向使能位  
 1 = P4B 引脚输出 PWM 波形，其极性由 PWM4M<1:0>控制  
 0 = P4B 引脚被分配为端口引脚
- P4STRENA: P4A 转向使能位  
 1 = P4A 引脚输出 PWM 波形，其极性由 PWM4M<1:0>控制  
 0 = P4A 引脚被分配为端口引脚

图注：R=可读 W=可写 --未用 U=未实现位

## 12.6 PWM3/4 的周期、占空比及分辨率

### 12.6.1 PWM3/4 周期

PWM<sub>y</sub> 的周期通过 16 位的寄存器<PPyH: PPyL> 进行设置,其值可设置为 0~65535, PWM<sub>y</sub> 的周期通过公式 12.1 进行计算。

公式 12.1:

$$\text{PWM周期}=(\langle\text{PPyH:PPyL}\rangle+1)\cdot T\cdot(\text{Tx预分频比})$$

### 12.6.2 PWM3/4 占空比

PWM<sub>y</sub> 占空比设置寄存器为 16 位, 通过寄存器<PWM<sub>y</sub>L1:PWM<sub>y</sub>L0>进行设置, PWM<sub>y</sub>L1 为占空比的高 8 位, PWM<sub>y</sub>L0 为低 8 位。脉冲宽度和占空比通过公式 12.2 和公式 12.3 计算:

公式 12.2:

$$\text{脉冲宽度}=(\text{PWM}_{yL1}:\text{PWM}_{yL0})\cdot T\cdot(\text{Tx预分频比})$$

公式 12.3:

$$\text{占空比}=\frac{\text{脉冲宽度}}{\text{PWM周期}}=\frac{\text{PWM}_{yL1}:\text{PWM}_{yL0}}{(\langle\text{PPyH:PPyL}\rangle+1)}$$

### 12.6.3 PWM3/4 分辨率

当 PPy 为 65535 时, PWM<sub>y</sub> 的最大分辨率为 16 位。分辨率的计算公式如下式所示。

$$\text{分辨率}=\frac{\text{Log}(\langle\text{PPyH:PPyL}\rangle+1)}{\log 2} \text{ 位}$$

### 12.6.4 PWM3/4 中断

当 PWM<sub>y</sub> 调制输出满一个周期时, 将中断标志位 TxIF (x=8/9) 置 1, 如果 Tx 中断使能, 且 AIE(全局中断允许位)和 PUIE(外设中断允许位)置 1, 程序将响应中断。如果 IPEN 和 PTx (x=8/9) 位均置 1, 则为高优先级中断。

## 12.7 单输出模式

通过将寄存器 PWM<sub>y</sub>CTL0 中的 PyM<1:0>位设置为 00, 选择单输出模式, 在此模式下, 默认从 PyA 引脚输出 PWM 信号, PyB、PyC 和 PyD 引脚为通用端口引脚。

在使能相应的 PWM<sub>y</sub> 引脚时, 将对应的引脚的方向控制位清零, 以将此引脚设置为输出模式; 如果相应引脚为模拟数字共用引脚, 应将该引脚设置为通用数字 I/O 口。

可通过设置 PyATRCTL<3:0>(参考寄存器 8.5)位, 使能或禁止 PWM<sub>y</sub> 的 4 路输出, PWM<sub>y</sub> 使用单输出模式时, 最多可同时在 4 个引脚输出同一个 PWM 信号。如下图所示, 在单输出模式下, 将 PyA 一路设置为 PWM 输出、PyA 和 PyB 两路同时设置为 PWM 输出的示例, 其它设置情况与此类似。

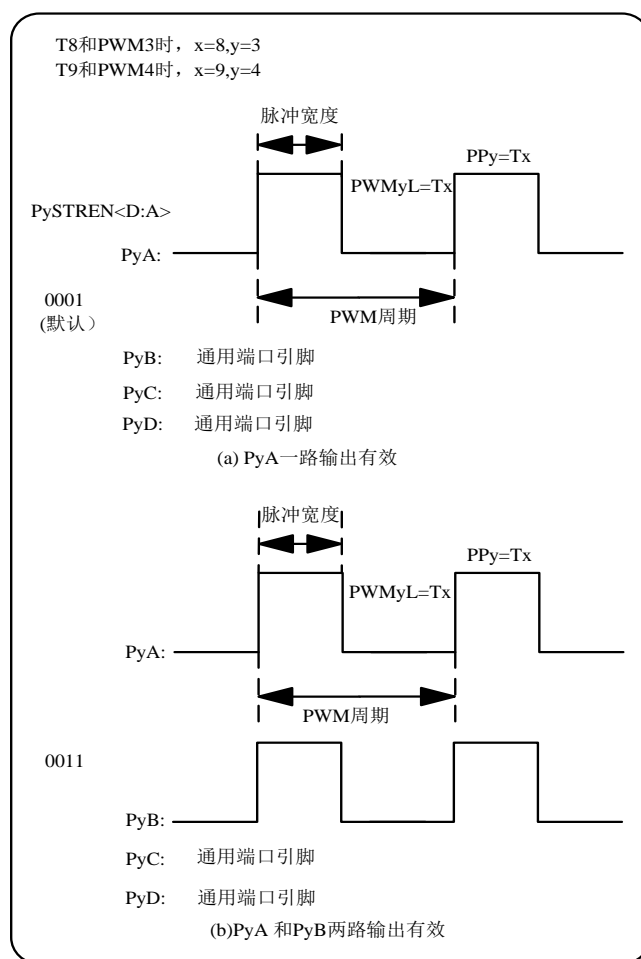


图 12.4 PWM4 的输出示例

在单输出模式过程中, 如果开始时某路输出 PWM 信号无效(为通用端口), 现在要将其设置为有效 PWM 输出, 此时可通过寄存器 PyATRCTL 中的 PySTRSYNC 位进行设置引脚输出切换时是否与指令同步。如下图 12.5 所示, PySTRSYNC=1 时, 对应引脚 Py<sub>z</sub> (y=3/4, z=A/B/C/D, 下同) 输出的 PWM 信号在 PySTREN<sub>z</sub> 置一后的 PWM<sub>y</sub> 输出信号周期结束时输出; PySTRSYNC=0 时, 对应引脚 Py<sub>z</sub> 输出的 PWM 信号在 PySTREN<sub>z</sub> 置一后立即输出。

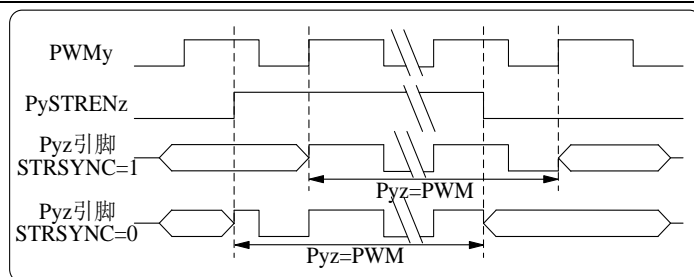


图 12.5 单输出模式 PWM 输出引脚切换

### 12.8 半桥输出模式

通过将寄存器 PWMMyCTL0 的 PyM<1:0>位设置为 10，把 PWMy 设置为半桥输出模式。在此模式下，PyA 和 PyB 被配置为调制输出，来驱动推挽式负载，PyC 和 PyD 被配置为通用端口。PWM 输出信号在 PyA 引脚上输出，而互补的 PWM 输出信号在 PyB 引脚上输出，如图 12.6 所示。

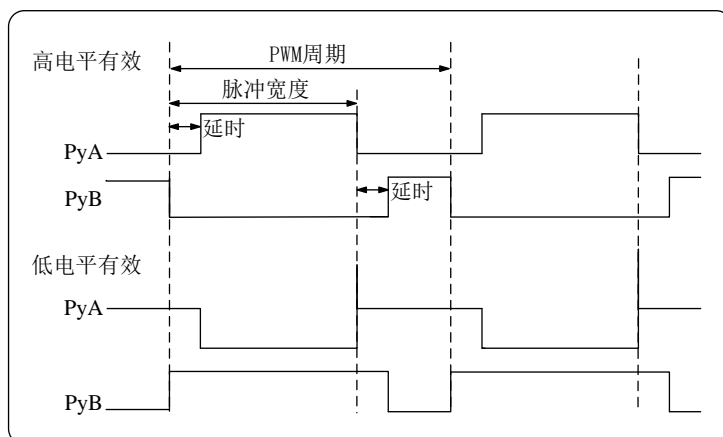


图 12.6 半桥输出模式输出信号示例

半桥输出模式可用于控制半桥和全桥控制电路，如图 12.7 所示，为半桥输出模式应用于半桥和全桥控制电路的示例。半桥输出模式应用于两个开关管的半桥控制电路，或使用 2 个 PWM 信号来控制 4 个开关管的全桥控制电路。

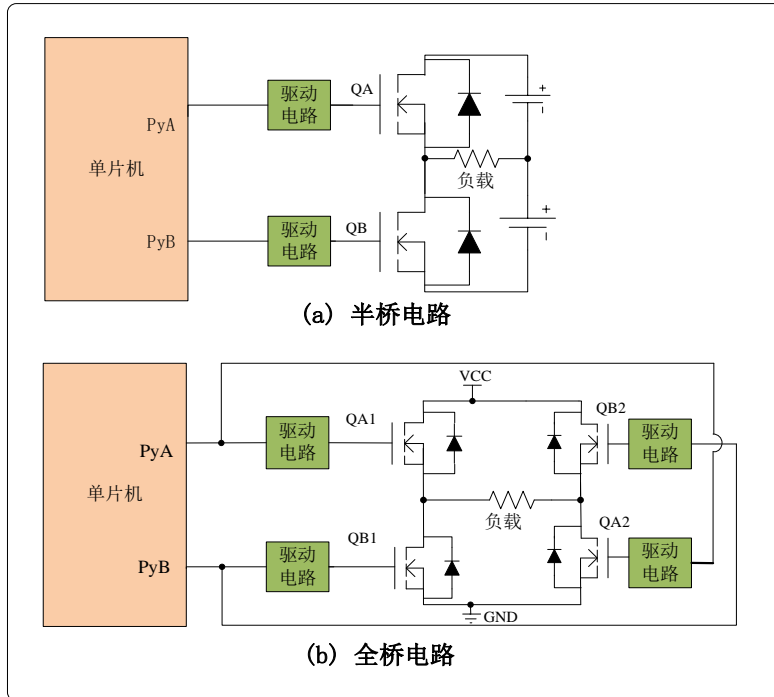


图 12.7 半桥输出模式应用举例

在使用半桥输出模式时，需将 PyA 和 PyB 引脚对应的方向控制位 TRxx 清零，设置为输出，将对应 ANSxx 位清零，设置为数字 I/O 口。

半桥输出模式具有可编程的死区延时功能，由于外部电路中的开关管等元件导通和截止时间存在差异，可用来防止在半桥驱动电路中产生直通电流，损坏相关电路。

PWM<sub>y</sub>CTL1 寄存器中 PyDC<6:0> 位的值用来设置死区延时时间。如果该值大于脉冲宽度，在整个周期内对应的输出将保持无效。

### 12.8.1 死区延时

如图 12.8 (a)、(b)所示，在半桥输出模式应用中，PyA 和 PyB 一直以 PWM 频率调制两个开关管，通常开关管的截止比导通需要更多的时间。如果 QA 和 QB 同时导通，两个管子可能会在一段很短的时间内都处于导通状态，在这很短的时间内，将会产生很大的电流流过两个管子，从而可能导致电路损坏，如图 12.8 (a)中的(1)、(2)、(3)和(4)处所示。为了避免开关期间产生这种具有破坏性的直通电流，可使其中一个管子关闭后再打开另一个管子。在半桥输出模式下，使用一个可编程死区延时模块，来避免产生的直通电流破坏电路。如图 12.8 所示，该延时在 PWM<sub>y</sub> 信号从非有效电平到有效电平转换时发生。延时时间通过寄存器 PWM<sub>y</sub>CTL1 的低 7 位进行设置。延时时间计数公式如下式所示。

$$\text{延时时间} = \text{PyDC}\langle 6:0 \rangle \cdot T$$

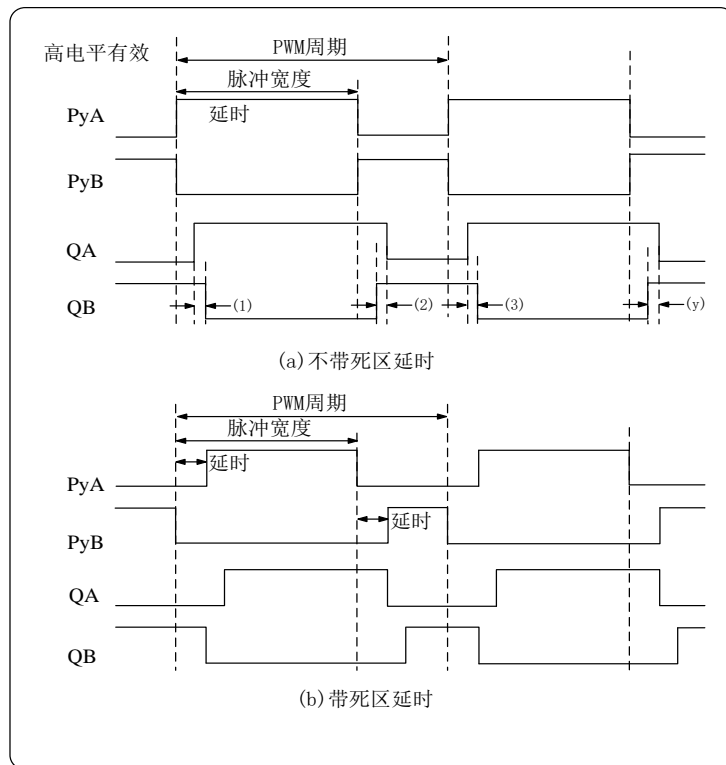


图 12.8 带死区控制和不带死区控制时的信号示例

## 12.9 全桥输出模式

全桥输出模式有全桥正向输出模式和全桥反向输出模式两种。通过将寄存  $PWM_{yCTL0}$  的  $PyM<1:0>$  设置为 01，把  $PWM_y$  设置为全桥正向输出模式；将其设置为 11，把  $PWM_y$  设置为全桥反向输出模式。

在全桥输出模式下， $PyA$ 、 $PyB$ 、 $PyC$  和  $PyD$  四个引脚都用作输出。将其设置为全桥正向模式时，引脚  $PyA$  被设置为有效电平，引脚  $PyD$  为 PWM 调制信号， $PyB$  和  $PyC$  为无效电平，图 12.9 (a)和图 12.10 (a)为全桥正向输出模式引脚信号示例。将其设置为全桥反向输出模式时， $PyC$  被驱动为有效电平，引脚  $PyB$  为 PWM 调制信号，而  $PyA$  和  $PyD$  为无效电平，图 12.9 (b)和图 12.10 (b)为全桥反向输出模式引脚信号示例。图 12.11 给出了全桥输出模式的应用电路示例。

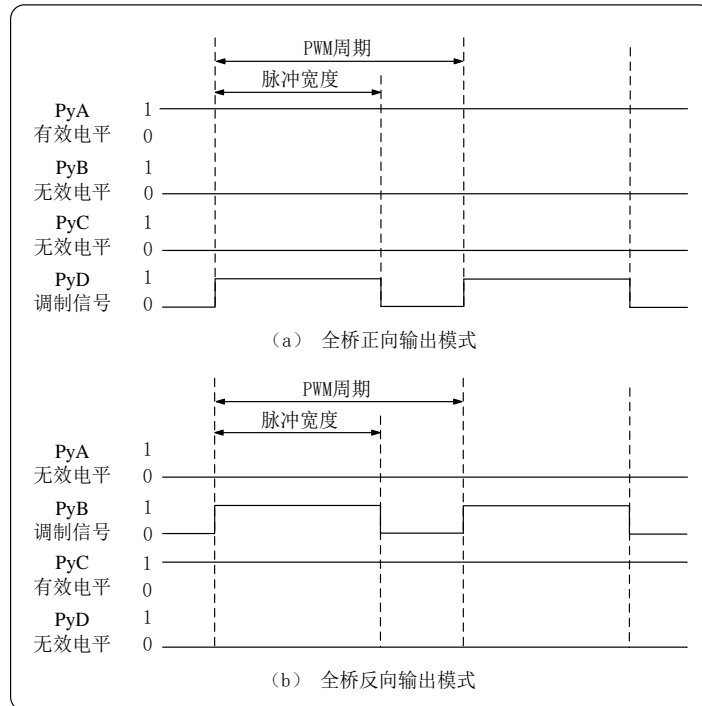


图 12.9 全桥输出模式引脚信号示例(高电平有效)

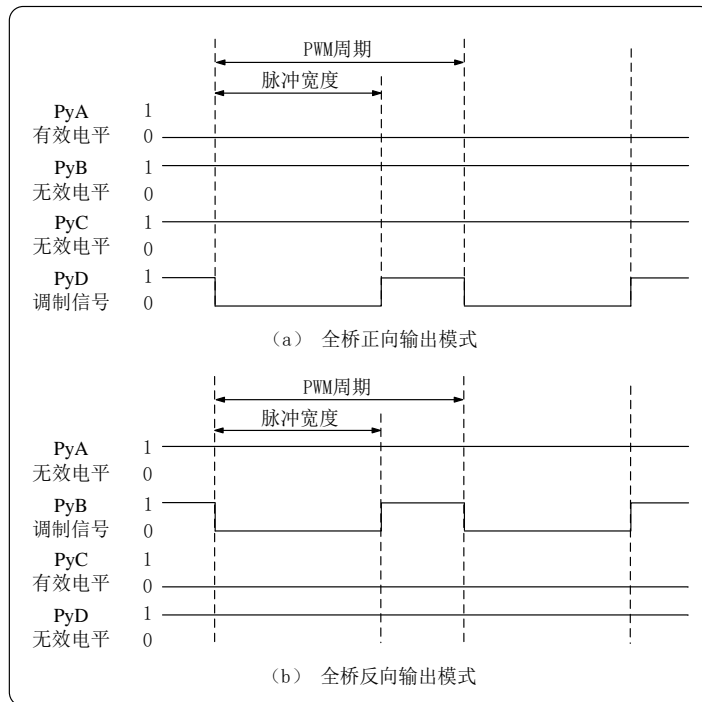


图 12.10 全桥输出模式引脚信号示例(低电平有效)

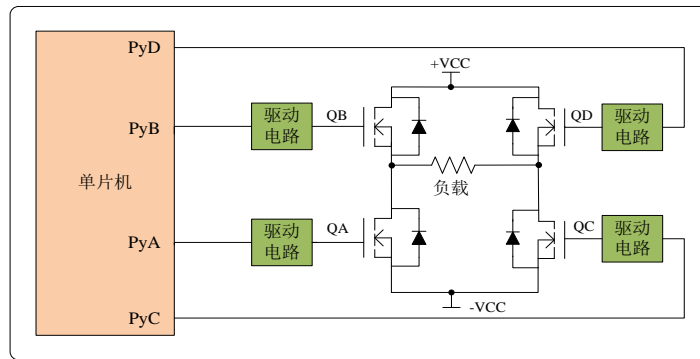


图 12.11 全桥输出模式应用电路示例

在使用全桥输出模式时，需将 PyA、PyB、PyC 和 PyD 引脚对应的方向控制位 TRxx 清零，设置为输出，将对应 ANSxx 位清零，设置为数字 I/O 口。

从全桥输出模式引脚信号波形以及应用电路示例可以看出：正向模式时，调制输出仅能控制一组开关管，假设此时流经负载的电流为正，则反向模式控制另一组开关管，使流经负载的电流为负。因此在使用全桥输出模式时，可将正向模式应与反向模式配合使用，使流经负载的电流方向改变。

通过将寄存器 PWMMyCTL0 的 PyM<1:0>位设置为 01(正向模式)和 11(反向模式)来改变流经负载的电流。如图 12.12 所示，从一种模式切换到另一种模式时，在前一种模式最后一个周期结束之前，PyB(或 PyD)被切换到无效状态，PyA(或 PyC)被切换到相反的状态。图中时间  $T = 62.5\text{ns} \cdot T_2$  预分频值。

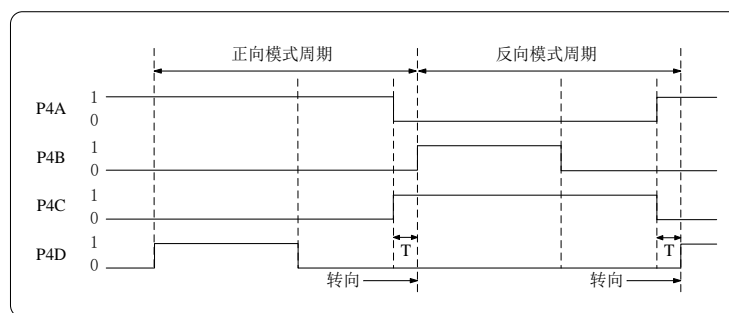


图 12.12 全桥正向、反向输出模式转换示例(高电平有效)

全桥输出模式下没有死区延时功能。通常在此模式中，任何时间只调制一对输出，因此不会导致电路产生直通电流，所以不需要死区延时。然而，当 PWM 的占空比接近百分之百，且开关管导通时间小于截止时间时，将会导致电路产生直通电流。图 12.13 为此情况下各种信号示例。



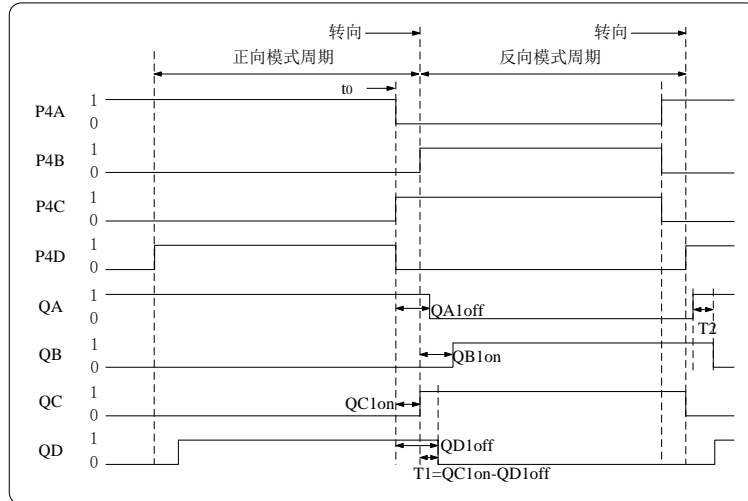


图 12.13 接近满占空比时换向时信号示例

图 12.13 中在  $t_0$  时刻  $P_{yA}$  和  $P_{yD}$  变为无效， $P_{yC}$  为有效。 $Q_A$ 、 $Q_B$ 、 $Q_C$  和  $Q_D$  分别为四路  $PWMy$  输出控制的开关管， $Q_{Aoff}$  和  $Q_{Doff}$  为开关管的截止延时时间， $Q_{Bon}$  和  $Q_{Con}$  为导通延时时间，由于管子导通和截止时间的差异，在第一次换向时，开关管  $Q_C$  和  $Q_D$  产生直通电流，持续时间为一个计数器周期，在后面一次换向时  $Q_A$  和  $Q_B$  产生直通电流，持续时间为一个计数器周期。

为了消除这种问题，可将换向前一个 PWM 信号周期的占空比调小，或者选用开关速度快的驱动电路，或者其它方法。

## 12.10 自动关断和自动重启模式

### 12.10.1 自动关断模式

$PWMy$  模块具有自动关断功能。如图 12.14 所示，为自动关断模式的原理框图。使能自动关断功能后，在外部关断事件发生时，该功能自动禁止 PWM 输出，然后将  $P_{yA}$ 、 $P_{yB}$ 、 $P_{yC}$  和  $P_{yD}$  四个引脚输出电平置于其预定义的状态。此模式用于防止 PWM 破坏应用电路。

自动关断模式具有 3 个关断源： $INT0$  引脚的逻辑低电平、比较器 1 输出高电平和在软件中直接将  $P_{yASE}$  位置 1。关断源触发关断的信号是高电平或低电平，而不是上升沿或下降沿，只要关断源的关断电平存在，自动关断状态将保持。

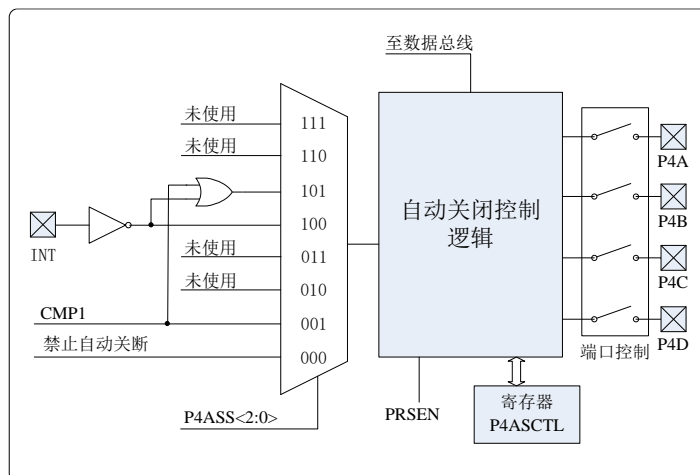


图 12.14 自动关断模式原理框图

通过寄存器 PyASCTL 的 PyASS<2:0>位选择自动关断源。将 PyASS<2:0>位设置为 000 时，关闭自动关断功能。

寄存器 PyASCTL 中的 PyASE 位指示关断的状态。如果该位为 0，表示 PWM<sub>y</sub> 的四个引脚输出正常的 PWM 信号，如果该位为 1，表示 PWM 的四路输出处于关断状态。

发生关断事件时，将会：

- ① PyASE 位被置 1。直到被软件清零或发生自动重启才会将该位清零。
- ② 使能的四个 PWM 引脚将被置于关断电平状态。

关断时，四路输出电平的状态由寄存器 PyASCTL 的 PySSAC<1:0>和 PySSBD<1:0>位决定。通过设定可将输出引脚置为：三态、高电平和低电平三种状态。其中 PyA 和 PyC 的状态由 PySSAC<1:0>设置，PyB 和 PyD 的状态由 PySSBD<1:0>设置。

打开自动关断功能后，如果关断源产生关断事件，则 PyASE 标志位被硬件置 1，四路输出被驱动为关断模式电平；关断源清除关断事件后，PyASE 仍然为 1(如果 PyRSEN=0)，四路输出仍然为关断模式电平，直到将 PyASE 位清零，PWM 重启。

### 12.10.2 自动重启模式

可将 PWM<sub>y</sub> 配置为一旦清除自动关断条件就自动重启 PWM。通过将 PWM<sub>y</sub>CTL1 寄存器中的 PyRSEN 位置一使能自动重启。

如果使能自动重启，只要自动关断条件有效，PyASE 位就将保持置 1。当清除自动关断条件时，将通过硬件将 PyASE 位清零，并且将恢复常规操作。

如图 12.15 所示，在图(a)中，PyRSEN=0，自动重启模式关闭，通过软件将 PyASE 标志位清零，PWM<sub>y</sub> 才会重启。在图(b)中，PyRSEN=1，自动重启模式被打开，关断事件被清除后，PyASE 标志位由硬件自动清零，然后重启 PWM<sub>y</sub>。

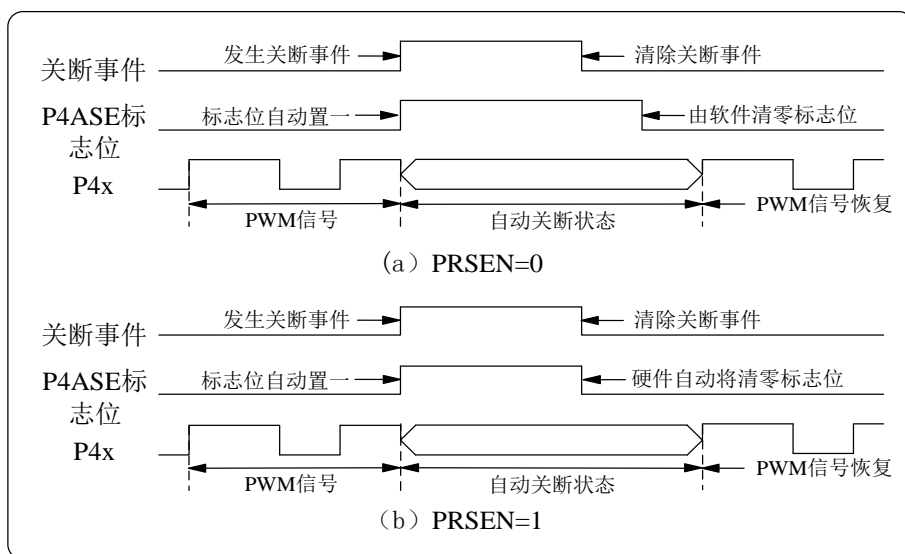


图 12.15 自动重启和软件重启 PWM 示例

## 13 CCP10(捕捉/比较/PWM5)模块

在 CCP 模块中，PWM5 为 16 位模式，T1/T10 计数模式为 16 位。PWM5 的 16 位周期寄存器由 {PP5H,PP5L} 组成。

在捕捉比较模式下，寄存器 PWM5L1 和 PWM5L0 分别作为数据寄存器的高 8 位和低 8 位寄存器；捕捉和比较的 16 位数据寄存器 PWM5L0 和 PWM5L1 与 T1L 和 T1H 进行配合使用。

在 PWM5 模式下，寄存器 PWM5L0 和 PWM5H0 作为 16 位占空比寄存器使用。

### 13.1 CCP 相关寄存器

当 CCP 模块配置为捕捉比较模式时，PWM5L1:PWM5L0 作为 CCP 的数据寄存器使用。

在捕捉模式下时，当 CCP10IN 引脚发生事件时，PWM5L1:PWM5L0 这对寄存器捕捉 T1H 和 T1L 寄存器的 16 位值。T1H 和 T1L 寄存器的值自动传递给 PWM5L1:PWM5L0 这对寄存器。

在比较模式下，16 位 PWM5L1:PWM5L0 寄存器的值将不断与 T1H/L 寄存器的值相比较。当两者匹配时，CCP 就会触发相应的事件。

当 CCP 模块配置为 PWM5 模式时，PWM5H0:PWM5L0 为 CCP 模块的占空比寄存器；PP5H: PP5L 为周期寄存器；PWM5H1: PWM5L1 为 PWM5 占空比缓冲寄存器（与用户无关）。

表 13.1 CCP 模块的相关寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
639H	PP5L	PWM5 周期寄存器低 8 位							
638H	PP5H	PWM5 周期寄存器高 8 位							
622H	PWM5L0	捕捉比较模式时，PWM5L0=CCP 数据寄存器低 8 位 PWM5 模式时，PWM5L0= PWM5 通道 1 占空比寄存器低 8 位							
63EH	PWM5H0	PWM5 通道 1 占空比寄存器高 8 位							
623H	PWM5L1	捕捉比较模式时，PWM5L1=CCP 数据寄存器高 8 位 PWM5 模式时，PWM5L1= PWM5 通道 1 低 8 位占空比缓冲寄存器（与用户无关）							
63FH	PWM5H1	PWM5 通道 1 高 8 位占空比缓冲寄存器（与用户无关）							
624H	PWM5L2	PWM5 通道 2 占空比寄存器低 8 位							
620H	PWM5H2	PWM5 通道 2 占空比寄存器高 8 位							
625H	PWM5L3	PWM5 通道 3 占空比寄存器低 8 位							
621H	PWM5H3	PWM5 通道 3 占空比寄存器高 8 位							
63AH	PWM5CTL0	P5CH1M OD1	P5CH1M OD0	P5CH3 MOD	P5CH2MO D	P5MOD3	P5MOD2	P5MOD1	P5MOD0
63DH	PWM5FC	-	-	FCA3	FCB3	FCA2	FCB2	FCA	FCB
627H	PWM5PC	-	-	PCA3	PCB3	PCA2	PCB2	PCA	PCB
626H	PWM5OC	-	-	OCA3	OCB3	OCA2	OCB2	OCA	OCB
63BH	PWM5CTL1	P5RSEN	P5DC6	P5DC5	P5DC4	P5DC3	P5DC2	P5DC1	P5DC0

634H	P5ASCTL0	P5ASE	P5ASS2	P5ASS1	P5ASS0	P5SSAC1	P5SSAC0	P5SSBD1	P5SSBD0
636H	P5STRCTL0	-	-	-	P5STRSYN C	P5STREN D	P5STREN C	P5STREN B	P5STREN A
635H	P5ASCTL1	P5SSA31	P5SSA30	P5SSB31	P5SSB30	P5SSA21	P5SSA20	P5SSB21	P5SSB20
637H	P5STRCTL1	-	-	-	-	P5STREN A3	P5STREN B3	P5STREN A2	P5STREN B2
63CH	PWM5CTL2	-	-	P5FUSES	-	-	UDEVT1	UDEVT0	UDEN

## 13.2 捕捉模式

在捕捉模式下，当对应的CCP10IN引脚发生事件时，PWM5L1:PWM5L0这对寄存器捕捉T1H和T1L寄存器的16位值，原理框图如下图所示：

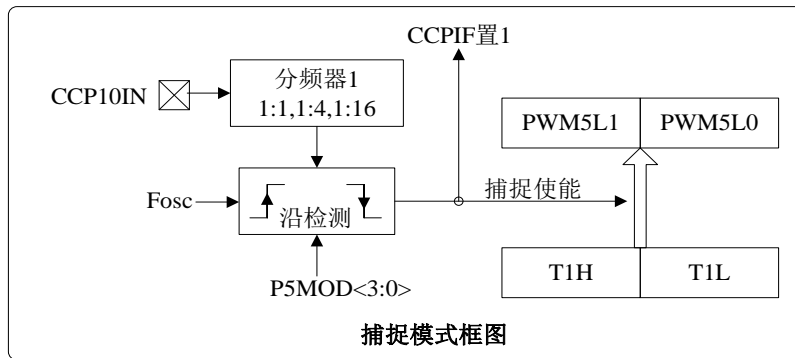


图 13.1 捕捉模式原理框图

触发捕捉的事件可被定义为以下四者之一，并且由 PWM5CTL0 寄存器中的模式选择位 P5MOD<3:0>选择事件类型位配置：

- ◆ 0100 = 每个下降沿
- ◆ 0101 = 每个上升沿
- ◆ 0110 = 每4 个上升沿
- ◆ 0111 = 每16个上升沿

在捕捉模式下，将 CCP10IN 引脚配置为数字输入。

当一个捕捉发生时，硬件自动中断请求标志位EIF8寄存器中的CCP10IF置1；它必须用软件清零。注意如果在PWM5L1和PWM5L0这对寄存器中的值被读取之前发生另一次捕捉，那么之前捕捉的值将被新捕捉的值覆盖。

当捕捉模式改变时，可能会产生错误的捕捉中断。用户应该在捕捉模式改变之前保持 EIE8 寄存器中的 CCP10IE 中断允许位清零以避免产生误中断。在捕捉模式发生任何改变之后也应清零 EIF8 寄存器中的中断标志位 CCP10IF。

PWM5CTL0 寄存器中的 P5MOD<3:0> 位指定了 4 种预分频器设置。每当关闭 CCP 模块或禁止捕捉模式时，就会清零预分频器计数器。这意味着任何复位都将清零预分频计数器。

虽然从一种捕捉预分频比切换到另一种捕捉预分频比不会将预分频计数器清零，但可

能会产生误中断。因此要避免出现这种不期望的操作，应在改变预分频比前通过将PWM5CTL0寄存器清零关闭该模块。

**注：**T1必须运行在定时模式或同步计数模式下CCP模块才能使用捕捉功能。在异步计数模式下无法进行捕捉操作。

### 13.3 比较模式

在比较模式下，16位的PWM5L1:PWM5L0寄存器的值将不断与T1寄存器的值相比较。当两者匹配时，CCP模块可能会出现以下几种情况：

- ◆ CCP10OUT引脚的输出电平翻转
- ◆ CCP10OUT引脚输出高电平
- ◆ CCP10OUT引脚输出低电平
- ◆ 产生软件中断触发信号
- ◆ 产生特殊事件触发信号

CCP10OUT引脚的动作取决于PWM5CTL0寄存器中P5MOD<3:0>控制位的值。

所有比较模式都会产生中断。原理图如下图所示：

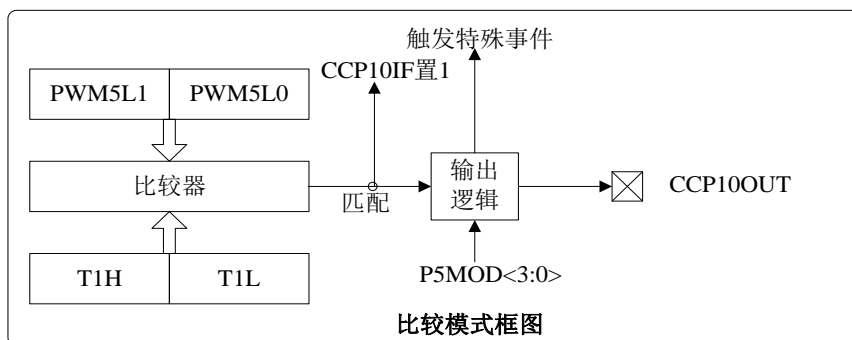


图 13.2 比较模式原理框图

在比较模式下，T1必须运行在定时模式或同步计数模式，计数周期至少大于一个机器周期。在异步计数模式下，可能无法进行比较操作。

- 当选择输出电平翻转模式（P5MOD<3:0>=0010）时，比较匹配时，CCP10OUT引脚的输出电平翻转，并将CCP10IF置1。
- 当选择普通比较模式时（P5MOD<3:0>=1000或1001）时，比较匹配时，CCP10OUT引脚输出高电平或者低电平，并且将CCP10IF置1。
- 当选择了软件中断触发模式时（P5MOD<3:0>=1010）时，比较匹配时，将CCP10IF置1，但是CCP模块不会控制CCP10OUT引脚。
- 当选择了特殊事件触发模式（P5MOD<3:0>=1011）时，比较匹配时，CCP会立即产生特殊事件触发输出，将CCP10IF置1，此时如果ADC已使能，将启动AD转换。但T1H/T1L寄存器不会立即复位，直到T1计数脉冲的下一个上升沿才复位。从而使PWM5L1/PWM5L0寄存器实际上成为了定时器1(T1)的16位可编程周期寄存器。

### 13.4 PWM5 模式

PWM5 为带有死区控制功能的增强型 PWM 模块。PWM5 有 3 个通道，每个通道都有一个 16 位的占空比设置寄存器。PWM5 的原理框图如下图所示。

PWM5不同通道所对应的占空比设置寄存器

通道1占空比寄存器	PWM5H0	PWM5L0
通道2占空比寄存器	PWM5H2	PWM5L2
通道3占空比寄存器	PWM5H3	PWM5L3

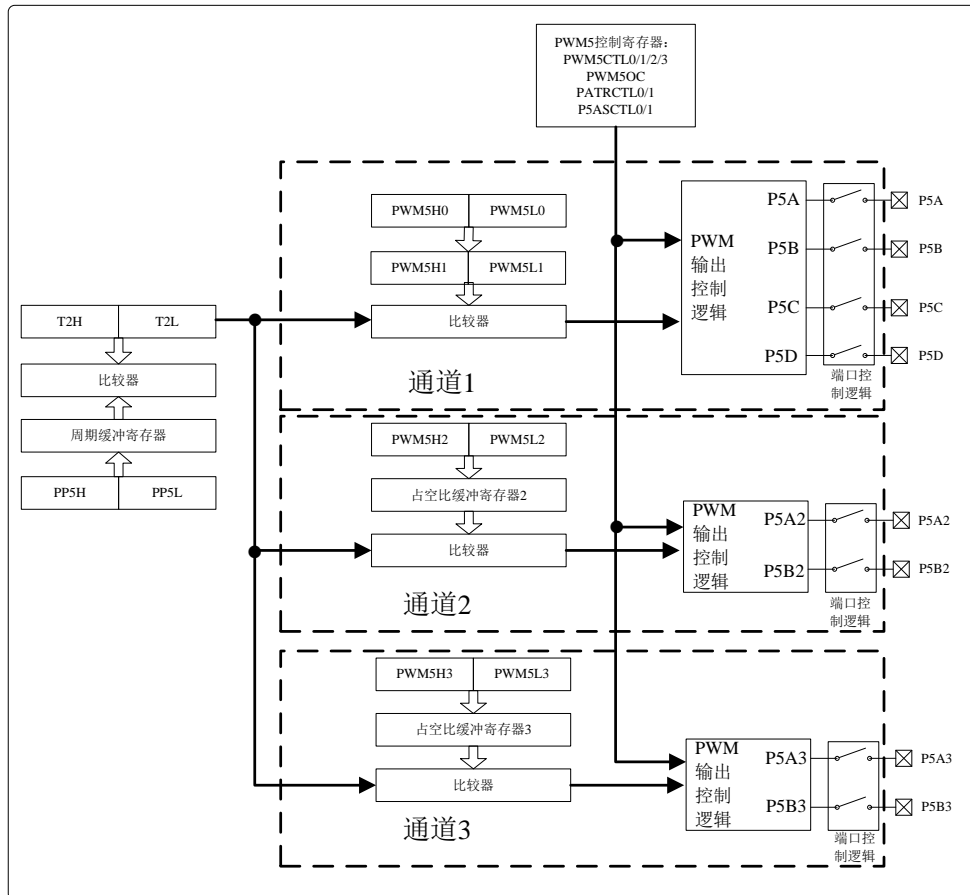


图 13.3 PWM5 原理框图

PWM5 最多可在 8 个不同的引脚输出 PWM 信号，分辨率最高 16 位。其中，通道 1 有 4 个引脚分别为 P5A、P5B、P5C 和 P5D；通道 2 和通道 3 分别有两个输出引脚，分别为 P5A2/P5B3 和 P5A3/P5B3。

PWM5 通道 1 有 4 种输出模式：单输出、半桥输出、全桥正向输出模式和全桥反向输出模式，通过寄存器 PWM5CTL0 中的 P5CHIMOD<1:0>位选择 4 种输出模式之一，通过 P5MOD<1:0>位设置各引脚的有效电平(可设置为高电平有效和低电平有效)。

PWM5 通道 2 和通道 3 一致，都有 2 种输出模式：单输出和半桥输出模式，可以通过 PWM5CTL0 中的 P5CH<3:2>MOD 位选择通道 2 和通道 3 的输出模式。在单输出模式和半

桥输出模式下，均可通过 P5STRCTL1 寄存器控制引脚作为 PWM 引脚还是通用 IO 引脚。在半桥模式下，通道 1/2/3 通均可以获得一个带死区控制的互补 PWM 输出。

P5STRCTL0/1 寄存器控制引脚作为 PWM 引脚还是通用 IO 引脚；通过寄存器 PWM5OC 中的 OCA3/ OCB3/ OCA2/ OCB2 位选择通道 2 和通道 3 作为 PWM 输出还是强制输出；通过 PWM5CTL3 中的 PCA3/ PCB3/ PCA2/ PCB2 位可以设置通道 2 和通道 3 输出引脚的极性；通过 PWM5CTL3 寄存器的 FCA3/ FCB3/ FCA2/ FCB2 位可以设置通道 2 和通道 3 引脚作为强制输出时的电平。

注：通道 1 通过选择单输出模式可以选择指定引脚输出 PWM 波形或作为 IO 口；通过选择半桥模式可以获得一个带死区控制的互补 PWM 输出；通过选择全桥模式可以实现 PWM 输出和强制输出同时使用，通过正向和反向模式还可以控制输出的极性，但对 PWM 的输出有限制，只能从 P5B 或 P5D 中输出 PWM 波形。如表 13.2 所示。

表 13.2 PWM5 通道 1 输出模式控制及有效引脚

PWM5 输出模式	P5CH1MOD <1:0>	有效引脚
单输出模式	00	默认将 P5A 配置为 PWM 输出，P5B、P5C 和 P5D 配置为端口引脚；可通过 P5STRCTL0 寄存器中的 P5STRENA <A:D>各位置 1，分别将 P5A、P5B、P5C 和 P5D 配置为 PWM 输出；PWM5 最多可在 4 个引脚输出 PWM 信号。
半桥输出模式	10	P5A 和 P5B 配置为调制输出；P5C 和 P5D 配置为端口引脚；半桥输出模式带有死区控制功能
全桥正向输出模式	01	P5D 配置为 PWM 调制输出；P5A 为有效电平；P5B 和 P5C 为无效电平
全桥反向输出模式	11	P5B 配置为 PWM 调制输出；P5C 为有效电平，P5A 和 P5D 为无效电平

注：各种输出模式的详细介绍请参考本节相应部分。

表 13.3 PWM5 通道 2 输出模式控制及有效引脚

PWM5 输出模式	P5CH2MOD	有效引脚
单输出模式	0	默认将 P5A2 配置为 PWM 输出，P5B2 配置为端口引脚；可通过 P5STRCTL1 寄存器中的 P5STRENA A2 和 P5STRENA B2 各位置 1，分别将 P5A2、P5B2 配置为 PWM 输出。
半桥输出模式	1	P5A2 和 P5B2 配置为调制输出；半桥输出模式带有死区控制功能

表 13.4 PWM5 通道 3 输出模式控制及有效引脚

PWM5 输出模式	P5CH3MOD	有效引脚
单输出模式	0	默认将 P5A3 配置为 PWM 输出，P5B3 配置为端口引脚；可通过 P5STRCTL1 寄存器中的 P5STRENA A3 和 P5STRENA B3 各位置 1，分别将 P5A3、P5B3 配置为 PWM 输出。
半桥输出模式	1	P5A3 和 P5B3 配置为调制输出；半桥输出模式带有死



表 13.5 PWM5 通道 1 输出极性控制

P5MOD<3:0>	PCA/PCB	PWM 输出极性
高有效	高有效	高有效
高有效	低有效	低有效
低有效	高有效	低有效
低有效	低有效	高有效

注：上述满足同或结果，注意这里的同或与数值 1 和 0 的同或无联系，这里仅从逻辑关系上去考虑，例如当通过 P5MOD<3:0>设置输出极性为高有效和通过 PCA/PCB 设置输出为高有效时，两者同或得实际 PWM 输出极性高有效，当通过 P5MOD<3:0>设置输出极性为低有效和通过 PCA/PCB 设置输出为高有效时，两者同或得实际 PWM 输出极性为低有效。

### 13.4.1 PWM5 相关控制寄存器

#### 13.4.1.1 PWM5CTL0 寄存器

寄存器： PWM5CTL0: PWM5控制寄存器0(地址: 63AH)

bit7				bit0				
复位值 0000 0000	P5CH1MO D1	P5CH1MO D0	P5CH3MO D	P5CH2MO D	P5MOD3	P5MOD2	P5MOD1	P5MOD0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P5CH1MOD<1:0>: PWM5 通道 1 输出配置位

00 = 单输出模式；默认 P5A 配置为 PWM 输出，P5B、P5C 和 P5D 为端口引脚；可通过 PSTRCTL 寄存器中的 P5STRENA<A:D>各位置 1，分别将 P5A、P5B、P5C 和 P5D 配置为 PWM 输出；通道 1 最多可提供 4 路 PWM 输出

01 = 全桥正向输出模式；P5D 配置为 PWM 调制输出,P5A 为有效电平,P5B 和 P5C 为无效电平

10 = 半桥输出模式；P5A 和 P5B 配置为调制输出；P5C 和 P5D 被分配为端口引脚，此模式带有死区控制功能

11 = 全桥反向输出模式；P5B 配置为调制输出；P5C 为有效电平；P5A 和 P5D 为无效电平

P5CH3MOD: PWM5 通道 3 输出配置位

0 = 单输出模式，默认为 P5A3 配置为 PWM 输出，P5B3 为端口引脚；通过 P5STRCTL1 寄存器中的 STRAEN3 和 STRBEN3 两位可以分别设置作为 PWM 引脚或端口引脚。

1 = 半桥输出模式，P5A3 和 P5B3 配置为调制输出；此模式带死区控制。

P5CH2MOD: PWM5 通道 2 输出配置位

0 = 单输出模式，默认为 P5A2 配置为 PWM 输出，P5B2 为端口引脚；通过 P5STRCTL1 寄存器中的 STRAEN2 和 STRBEN2 两位可以分别设置作为 PWM 引脚或端口引脚。

1 = 半桥输出模式，P5A2 和 P5B2 配置为调制输出；此模式带死区控制。

P5MOD<3:0>: PWM5模块的模式选择位

- 0000 = 捕捉/ 比较/PWM5 关闭（复位模块）
- 0001 = 未使用（保留）
- 0010 = 比较模式，匹配时输出电平翻转（CCP10IF 置 1）
- 0011 = 未使用（保留）
- 0100 = 捕捉模式，在每个下降沿发生捕捉
- 0101 = 捕捉模式，在每个上升沿发生捕捉
- 0110 = 捕捉模式，每 4 个上升沿发生捕捉
- 0111 = 捕捉模式，每 16 个上升沿发生捕捉
- 1000 = 比较模式，比较匹配时输出高电平（CCP10IF 置 1）
- 1001 = 比较模式，比较匹配时输出低电平（CCP10IF 置 1）
- 1010 = 比较模式，比较匹配时产生软件中断（CCP10IF 置 1，P5A 引脚不受影响）
- 1011 = 比较模式，触发特殊事件（CCP10IF 位置 1，CCP 复位 T1，如果 ADC 使能将启动 AD 转换。）
- 1100 = PWM 模式；P5A 和 P5C 为高电平有效；P5B 和 P5D 也为高电平有效
- 1101 = PWM 模式；P5A 和 P5C 为高电平有效；P5B 和 P5D 为低电平有效
- 1110 = PWM 模式；P5A 和 P5C 为低电平有效；P5B 和 P5D 为高电平有效
- 1111 = PWM 模式；P5A 和 P5C 为低电平有效；P5B 和 P5D 也为低电平有效

注：通道 1 中 P5A/P5B 极性控制由 PCA/PCB 和 P5MOD<3:0>同或影响（见表 9-5 所示），P5C/P5D 由 P5MOD<3:0>控制。

### 13.4.1.2 PWM5CTL1 寄存器

寄存器: PWM5CTL1: PWM5控制寄存器1(地址: 63BH)

复位值	bit7						bit0	
0000 0000	P5RSEN	P5DC6	P5DC5	P5DC5	P5DC3	P5DC2	P5DC1	P5DC0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P5RSEN: PWM5 重启使能位

- 1 = 自动关闭时，一旦关闭事件消失，P5ASE 位自动清零，PWM5 自动重启
- 0 = 自动关闭时，P5ASE 由软件清零，以重启 PWM5

P5DC<6:0>: PWM5 死区延时时间设置位，用于设置死区延时的时间，见下式

$$\text{延时时间} = \text{P5DC}<6:0> \cdot T$$



## 13.4.1.3 PWM5CTL2 寄存器

寄存器: PWM5CTL2: PWM5更新控制寄存器(地址:63CH)

复位值	bit7						bit0	
--1- -000	-	-	P5FUSES	-	-	UDEVT1	UDEVT0	UDEN
	U	U	R/W	U	U	R/W	R/W	R/W

P5FUSES: 由 P5FUSES 位控制 PWM5 输出

0 = 不由配置字控制 PWM5 输出所对应的 IO 口

1 = 由配置字中的 PWMPIN、HPOL 和 LPOL 位控制输出所对应的 IO 口

UDEVT1: 更新事件控制 1 (在 UDEN=1 时有效)

0 = 当定时器 T10 为 0 时更新占空比、周期寄存器到缓冲器中

1 = 产生更新事件, 更新占空比、周期、输出控制、极性控制、强制控制寄存器到缓冲器中, 并将定时器和死区定时器清零

UDEVT0: 更新事件控制 0 (在 UDEN=1 时有效)

0 = 当定时器 T10 为 0 时更新输出控制、极性控制、强制控制寄存器

1 = 立即更新输出控制、极性控制、强制控制寄存器到缓冲器中

UDEN: 更新使能

1 = 允许占空比、周期、输出控制、极性控制、强制控制寄存器更新

0 = 禁止占空比、周期、输出控制、极性控制、强制控制寄存器更新

## 13.4.1.4 PWM5 极性控制寄存器 (PWM5PC)

寄存器: PWM5PC: PWM5极性控制寄存器(地址:627H)

复位值	bit7						bit0	
--00 0000	-	-	PCA3	PCB3	PCA2	PCB2	PCA	PCB
	U	U	R/W	R/W	R/W	R/W	R/W	R/W

PCA3: 当OCA3为0时, 通道3中P5A3端口输出极性控制

0 = 输出为高电平有效

1 = 输出为低电平有效

PCB3: 当OCB3为0时, 通道3中P5B3端口输出极性控制

0 = 输出为高电平有效

1 = 输出为低电平有效

PCA2: 当OCA2为0时, 通道2中P5A2端口输出极性控制

0 = 输出为高电平有效

1 = 输出为低电平有效

PCB2: 当OCB2为0时, 通道2中P5B2端口输出极性控制

0 = 输出为高电平有效

1 = 输出为低电平有效

PCA: 当OCA为0时, 通道1中P5A端口输出极性控制

0 = 输出为高电平有效

1 = 输出为低电平有效

PCB: 当OCB为0时，通道1中P5B端口输出极性控制  
 0 = 输出为高电平有效  
 1 = 输出为低电平有效

注：通道1中P5A/P5B极性控制由PCA/PCB和P5MOD<3:0>同或影响（见表9-5所示），P5C/P5D由P5MOD<3:0>控制。

### 13.4.1.5 PMW5 强制控制寄存器（PWM5FC）

寄存器: PWM5FC: PWM5强制控制寄存器(地址:63DH)

复位值	bit7						bit0	
--00 0000	-	-	FCA3	FCB3	FCA2	FCB2	FCA	FCB
	U	U	R/W	R/W	R/W	R/W	R/W	R/W

FCA3: 当OCA3为1时，通道3中P5A3端口强制输出控制  
 0 = 强制输出为低电平  
 1 = 强制输出为高电平

FCB3: 当OCB3为1时，通道3中P5B3端口强制输出控制  
 0 = 强制输出为低电平  
 1 = 强制输出为高电平

FCA2: 当OCA2为1时，通道2中P5A2端口强制输出控制  
 0 = 强制输出为低电平  
 1 = 强制输出为高电平

FCB2: 当OCB2为1时，通道2中P5B2端口强制输出控制  
 0 = 强制输出为低电平  
 1 = 强制输出为高电平

FCA: 当OCA为1时，通道1中P5A端口强制输出控制  
 0 = 强制输出为低电平  
 1 = 强制输出为高电平

FCB: 当OCB为1时，通道1中P5B端口强制输出控制  
 0 = 强制输出为低电平  
 1 = 强制输出为高电平

### 13.4.1.6 PWM5 输出控制寄存器（PWM5OC）

寄存器: PWM5OC: PWM5输出控制寄存器(地址:626H)

复位值	bit7						bit0	
--00 0000	-	-	OCA3	OCB3	OCA2	OCB2	OCA	OCB
	U	U	R/W	R/W	R/W	R/W	R/W	R/W

OCA3: 通道3中P5A3端口输出控制  
 0 = PWM输出，输出极性由PCA3控制  
 1 = 强制输出，输出电平由FCA3控制

OCB3: 通道3中P5B2端口输出控制

- 0 = PWM输出, 输出极性由PCB3控制
- 1 = 强制输出, 输出电平由FCB3控制
- OCA2: 通道2中P5A2端口输出控制
  - 0 = PWM输出, 输出极性由PCA2控制
  - 1 = 强制输出, 输出电平由FCA2控制
- OCB2: 通道2中P5B2端口输出控制
  - 0 = PWM输出, 输出极性由PCB2控制
  - 1 = 强制输出, 输出电平由FCB2控制
- OCA: 通道1中P5A端口输出控制
  - 0 = PWM输出, 输出极性由PCA控制
  - 1 = 强制输出, 输出电平由FCA控制
- OCB: 通道1中P5B端口输出控制
  - 0 = PWM输出, 输出极性由PCB控制
  - 1 = 强制输出, 输出电平由FCB控制

### 13.4.1.7 P5ASCTL0 寄存器

寄存器: P5ASCTL0: PWM5自动关闭控制寄存器0(地址: 634H)

复位值 0000 0000	bit7	P5ASE	P5ASS2	P5ASS1	P5ASS0	P5SSAC1	P5SSAC0	P5SSBD1	P5SSBD0	bit0
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- P5ASE: 自动关闭事件状态位
  - 1 = 发生了关闭事件; 四路输出为关闭状态
  - 0 = 四路输出正常工作
- P5ASS<2:0>: 自动关闭源选择位
  - 000 = 禁止自动关断
  - 001 = 比较器 C1 输出高电平
  - 100 = INT 引脚上的逻辑低电平
  - 101 = INT 引脚上的逻辑低电平或比较器 C1 输出高电平
  - 其它 = 未使用
- P5SSAC<1:0>: 引脚 P5A 和 P5C 关闭状态控制位
  - 00 = 驱动引脚 P5A 和 P5C 为 0
  - 01 = 驱动引脚 P5A 和 P5C 为 1
  - 1x = 引脚 P5A 和 P5C 为三态
- P5SSBD<1:0>: 引脚 P5B 和 P5D 关闭状态控制位
  - 00 = 驱动引脚 P5B 和 P5D 为 0
  - 01 = 驱动引脚 P5B 和 P5D 为 1
  - 1x = 引脚 P5B 和 P5D 为三态

## 13.4.1.8 P5ASCTL1 寄存器

寄存器: P5ASCTL1: PWM5自动关闭控制寄存器1(地址: 635H)

	bit7						bit0	
复位值 0000 0000	P5SSA31	P5SSA30	P5SSB31	P5SSB30	P5SSA21	P5SSA20	P5SSB21	P5SSB20
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P5SSA3&lt;1:0&gt;: 引脚 P5A3 关闭状态控制位

00 = 驱动引脚 P5A3 为 0

01 = 驱动引脚 P5A3 为 1

1x = 引脚 P5A3 为三态

P5SSB3&lt;1:0&gt;: 引脚 P5B3 关闭状态控制位

00 = 驱动引脚 P5B3 为 0

01 = 驱动引脚 P5B3 为 1

1x = 引脚 P5B3 为三态

P5SSA2&lt;1:0&gt;: 引脚 P5A2 关闭状态控制位

00 =驱动引脚 P5A2 为 0

01 =驱动引脚 P5A2 为 1

1x =引脚 P5A2 为三态

P5SSB2&lt;1:0&gt;: 引脚 P5B2 关闭状态控制位

00 =驱动引脚 P5B2 为 0

01 =驱动引脚 P5B2 为 1

1x =引脚 P5B2 为三态

## 13.4.1.9 P5STRCTL0 寄存器

寄存器: P5STRCTL0: 脉冲转向控制寄存器(地址: 636H)

	bit7						bit0	
复位值 ---0 0001	-	-	-	P5STRSY NC	P5STREN D	P5STREN C	P5STREN B	P5STREN A
	U	U	U	R/W	R/W	R/W	R/W	R/W

P5STRSYNC: 转向同步位

1 =同步控制输出转向更新, 由 UDEN/UDEVTO 控制更新时机

0 = 在指令周期边界的开始发生输出转向更新

P5STREND: 转向使能位 D

1 = P5D 引脚输出 PWM 波形, 其极性由 P5MOD&lt;1:0&gt;控制

0 = P5D 引脚被分配为端口引脚

P5STRENC: 转向使能位 C

1 = P5C 引脚输出 PWM 波形, 其极性由 P5MOD&lt;1:0&gt;控制

0 = P5C 引脚被分配为端口引脚

P5STRENB: 转向使能位 B

1 = P5B 引脚输出 PWM 波形, 其极性由 P5MOD&lt;1:0&gt;控制

0 = P5B 引脚被分配为端口引脚

P5STRENA: 转向使能位 A

1 = P5A 引脚输出 PWM 波形，其极性由 P5MOD<1:0>控制

0 = P5A 引脚被分配为端口引脚

### 13.4.1.10 P5STRCTL1 寄存器

**寄存器:** P5STRCTL1: 脉冲转向控制寄存器2(地址: 637H)

复位值	bit7				bit0			
---- 1010	-	-	-	-	P5STREN A3	P5STREN B3	P5STREN A2	P5STREN B2
	U	U	U	U	R/W	R/W	R/W	R/W

P5STRENA3: 转向使能位 A3

1 = P5A3 引脚输出 PWM 波形，其极性由 P5MOD<1:0>控制

0 = P5A3 引脚被分配为端口引脚

P5STRENB3: 转向使能位 B3

1 = P5B3 引脚输出 PWM 波形，其极性由 P5MOD<1:0>控制

0 = P5B3 引脚被分配为端口引脚

P5STRENA2: 转向使能位 A2

1 = P5A2 引脚输出 PWM 波形，其极性由 P5MOD<1:0>控制

0 = P5A2 引脚被分配为端口引脚

P5STRENB2: 转向使能位 B2

1 = P5B2 引脚输出 PWM 波形，其极性由 P5MOD<1:0>控制

0 = P5B2 引脚被分配为端口引脚

## 13.4.2 PWM5 的周期、占空比及分辨率

### 13.4.2.1 PWM5 周期

PWM5 的周期通过 16 位的寄存器<PP5H: PP5L> 进行设置,其值可设置为 0~65535。PWM5 边沿对齐的周期通过公式 13.1 进行计算。

公式 13.1:

$$\text{PWM周期}=(\langle\text{PP5H:PP5L}\rangle+1)\cdot T\cdot(\text{T10预分频比})$$

注: T 为 T10 工作时钟周期。

T10 预分频比通过 T10 分频器 1 的配置位 T10CKPS<1:0>位配置。

周期寄存器为带缓冲模式,当 UDEN 位为 0 时,禁止更新占空比寄存器和周期寄存器。当 UDEN 为 1 时,允许更新占空比寄存器和周期寄存器,此时若 UDEVT1 为 0,则周期可以在 T10 为 0 时更新到周期缓冲器中;若 UDEVT1 为 1 时,则立即更新周期到周期缓冲器中,T10 被清零。下一次立即更新前,需要软件清零 UDEVT1,再置 1 才能开启新一次的立即更新。

注意:(1) 当 T10 启动时周期寄存器立即更新;

(2) 当周期寄存器的值更新到周期缓冲器后,才真正完成了周期寄存器的设置。

### 13.4.2.2 PWM5 占空比

PWM5 有 3 个占空比寄存器，占空比设置寄存器全为 16 位，通过寄存器 <PWM5H0:PWM5L0>、<PWM5H2:PWM5L2>和<PWM5H3:PWM5L3>(为后面描述方便，统称为<PWM5Hx:PWM5Lx>)进行设置，PWM5Hx 为占空比的高 8 位，PWM5Lx 为低 8 位。在边沿对齐模式下，脉冲宽度和占空比通过公式 13.2 和公式 13.3 计算。

公式 13.2:

$$\text{脉冲宽度} = (\text{PWM5Hx:PWM5Lx}) \cdot T \cdot (T10\text{预分频比})$$

公式 13.3:

$$\text{占空比} = \frac{\text{脉冲宽度}}{\text{PWM周期}} = \frac{\text{PWM5Hx:PWM5Lx}}{\text{PP5}+1}$$

由上述公式可知，占空比为 0%和 100%满足条件如下:

- 边沿对齐模式:
  - 0%: 占空比寄存器为 0
  - 100%:占空比寄存器为大于等于 (PP5+1) (其中 PP5 为周期寄存器)
- 在半桥模式下，占空比为 0%或者 100%时，只有在边界 PWM 变化时插入死区，否则不插入死区。

占空比寄存器为带缓冲模式，通过 UDEN 位可以使能或禁止占空比和周期寄存器的更新。当 UDEN 位为 0 时，禁止更新占空比寄存器和周期寄存器。当 UDEN 位置 1 时，若 UDEVT1 为 0，则占空比可以在 T10 为 0 时更新到缓冲器中；若 UDEVT1 为 1 时，则立即更新占空比到缓冲器中，T10 被清零。下一次立即更新前，需要软件清零 UDEVT1，再置 1 才能开启新一次的立即更新。

注意: (1) 当 T10 启动时占空比寄存器立即更新;

(2) 占空比寄存器的值更新到占空比缓冲器后，才真正完成了占空比寄存器的设置。

### 13.4.2.3 PWM5 分辨率

当 PP5 为 65535 时，PWM5 的最大分辨率为 16 位。分辨率的计算公式如下式所示。

$$\text{分辨率} = \frac{\text{Log}[4(\text{<PP5H:PP5L>}+1)]}{\text{log}2} \quad \text{位}$$

### 13.4.2.4 PWM5 中断

当 PWM5 调制输出满一个周期时，将中断标志位 T10IF 置 1，如果 PWM5 中断使能，且 AIE(全局中断允许位)和 PUIE(外设中断允许位)置 1，程序将相应中断。如果 IPEN 和 PT10 位均置 1，则为高优先级中断。



### 13.4.3 边沿对齐 PWM 信号

当 T10CTL1 控制寄存器中的 T10MOD<1:0>计数模式选择位设置成 00 时，产生的 PWM 信号为边沿对齐的 PWM 信号。在该模式下，PWM 信号的周期由 <PP5H:PP5L> 周期寄存器决定，占空比由 16 位的寄存器 <PWM5Hx:PWM5Lx> 决定（其中 x=0,2,3）。在周期开始时（即计数器 T10 等于 0 时）PWM 驱动为高电平，当计数器 T10 与 <PWM5Hx:PWM5Lx> 占空比寄存器发生匹配时，PWM 输出低电平。

如果占空比寄存器设置成 0，那么在整个 PWM 周期都输出低电平。如果占空比寄存器设置大于 <PP5H:PP5L> 周期寄存器的值，那么在整个 PWM 周期都输出高电平。

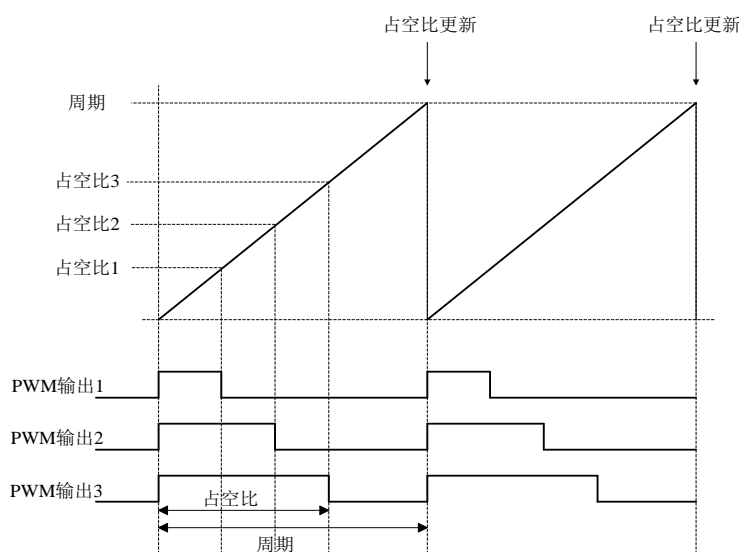


图 13.4 边沿对齐 PWM

### 13.4.4 PWM 信号产生和中断

通过设置 T10CTL1 控制寄存器中的 T10MOD<1:0>可以选择 T10 的计数方式，从而产生边沿对齐的 PWM 信号。当 T10 向上计数时，产生边沿对齐的 PWM 信号。

通过设置 T10CTL1 寄存器中的 T10MOD<1:0>位控制选择溢出中断，则可以选择在不同的时机触发 PWM 中断操作。

当 T10MOD<1:0>=00 或者 01 时，是上溢时产生中断标志，就是在计数器 T10 向上计数到周期值的时候产生一个中断信号。

当 T10MOD<1:0>=10 时，是下溢产生中断标志，也就是计数器 T10 向下递减为 0 的时候产生一个中断信号。

当 T10MOD<1:0>=11 时，是上溢-下溢中断，在这种状态下在 <PP5H:PP5L> 和计数器 T10 相等以及计数器 T10 向下计数到 0 时都产生中断信号。

在向上计数模式时，使用上溢中断，在向上-向下计数模式时可以使用 3 种中断产生方式。

### 13.4.5 单输出模式

通过将寄存器 PWM5CTL0 中的 P5CH1MOD<1:0>位设置为 00，选择通道 1 为单输出模

式，在此模式下，默认从 P5A 引脚输出 PWM 信号，P5B、P5C 和 P5D 引脚为通用端口引脚。将 PWM5CTL0 中的 P5CH3MOD 和 P5CH3MOD 位均设置为 0，选择通道 2 和通道 3 为单输出模式，在此模式下，默认从 P5A2 和 P5A3 引脚输出 PWM 信号，P5B2 和 P5B3 引脚为通用端口引脚。在使能相应的 PWM5 引脚时，应将对应的 TRxx 位清零，以将此引脚设置为输出模式。

可通过设置寄存器 P5STRCTL0/1，使能或禁止 PWM5 的 8 路输出，PWM5 使用单输出模式时，最多可同时在 8 个引脚输出 PWM 信号。当 8 个引脚同时输出 PWM 信号时，通道 1 的四个引脚 P5A,P5B,P5C,P5D 输出的 PWM 信号完全相同；通道 2 的两个引脚 P5A2, P5B2 输出的 PWM 信号完全相同；通道 3 的两个引脚 P5A3, P5B3 输出的 PWM 信号完全相同。通道 1,2,3 有各自独立的占空比寄存器故可输出互不相同的 PWM 信号。如图 13.5 所示，在单输出模式下，只将 P5A 一路设置为 PWM 输出和将 P5A 和 P5B 两路同时设置为 PWM 输出的示例，其它设置情况与此类似。

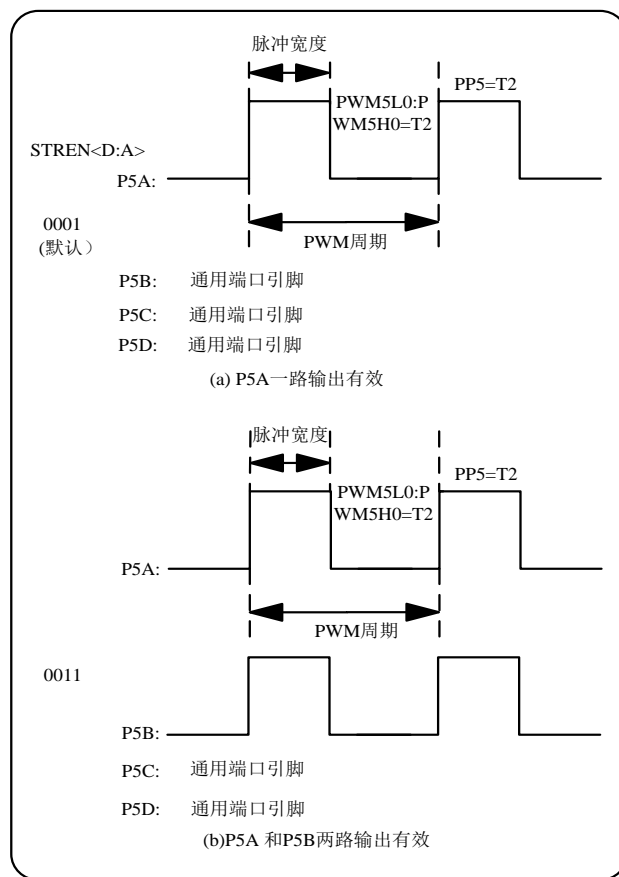


图 13.5 PWM5 的输出示例

在单输出模式过程中，如果开始时某路输出 PWM 信号无效(为通用端口)，现在要将其设置为有效 PWM 输出,此时可通过寄存器 P5STRCTL0 中的 P5STRSYNC 位进行设置引脚输出切换时是否与指令同步。如图 13.5 所示，P5STRSYNC=1 时，脉冲转向控制为同步更新模式。在该模式下，对应引脚的脉冲转向控制受 UDEN/UDEVT0 控制，可以实现同步更新，只有在 UDEN 使能时才能使能所配置的脉冲转向设置，而 UDEVT0 则选择是脉冲转向控制是与周期同步还是立即有效，T10 使能时会产生一次更新操作。P5STRSYNC=0 时，对应引脚 P5x 输出的 PWM 信号在 P5STRENAx 置一后立即输出；

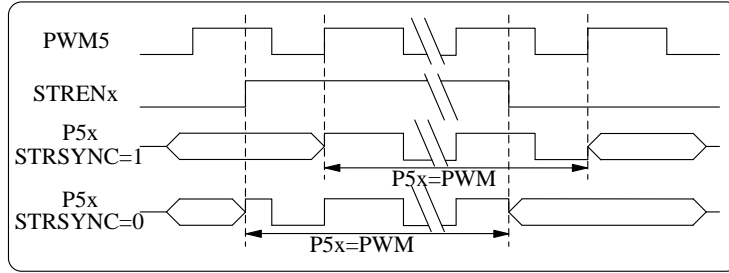


图 13.6 单输出模式 PWM 输出引脚切换

### 13.4.6 半桥输出模式

通过将寄存器 PWM5CTL0 的 P5CH1MOD<1:0>位设置为 10，可将通道 1 设置为半桥输出模式；将 P5CH3MOD 和 P5CH2MOD 置 1，可将通道 2 和 3 设置为半桥输出模式。在此模式下，通道 1 的 P5A 和 P5B 被配置为调制输出，来驱动推挽式负载，P5C 和 P5D 被配置为通用端口。PWM 输出信号在 P5A 引脚上输出，而互补的 PWM 输出信号在 P5B 引脚上输出，如图 13.7 所示，通道 2、3 与此类似。

通道 2 和 3 的在半桥模式下与通道 1 相似，P5A2/P5B2，P5A3/P5B3 被配置为调制输出，PWM 输出信号在 P5A2/3 引脚上输出，而互补的 PWM 输出信号在 P5B2/3 引脚上输出。

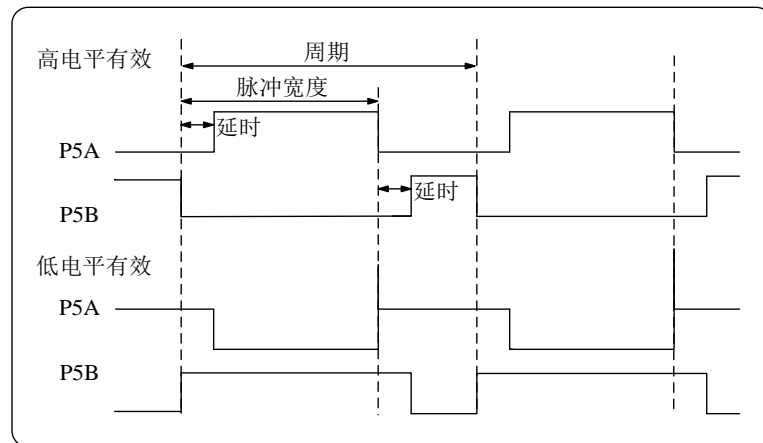


图 13.7 半桥输出模式输出信号示例

半桥输出模式可用于控制半桥和全桥控制电路，如图 13.8 所示，为半桥输出模式应用于半桥桥和全桥控制电路的示例。半桥输出模式应用于两个开关管的半桥控制电路，或使用 2 个 PWM 信号来控制 4 个开关管的全桥控制电路。

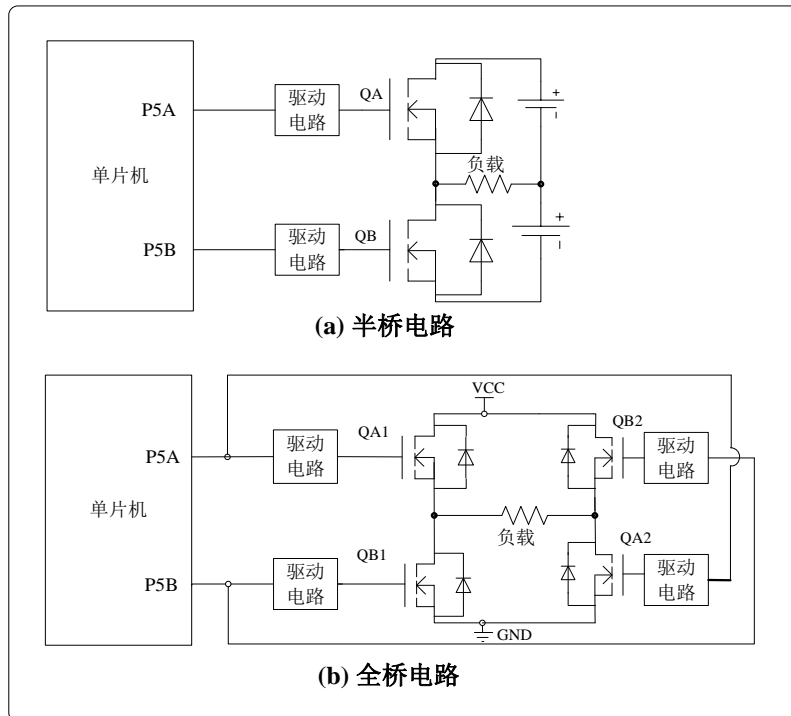


图 13.8 半桥输出模式应用举例

在使用半桥输出模式时，需将 P5A 和 P5B 引脚对应的方向控制位 TRxx 清零，设置为输出。

半桥输出模式具有可编程的死区延时功能，由于外部电路中的开关管等元件导通和截止时间存在差异，可用来防止在半桥驱动电路中产生直通电流，损坏相关电路。PWM5CTL1 寄存器中 PDC<6:0> 位的值用来设置死区延时时间。如果该值大于脉冲宽度，在整个周期内对应的输出将保持无效。

### 13.4.6.1 死区延时

如图 13.9 所示，在半桥输出模式应用中，P5A 和 P5B 一直以 PWM 频率调制两个开关管，通常开关管的截止比导通需要更多的时间。如果 QA 和 QB 两个管子在一小段很短的时间内都处于导通状态，在这很短的时间内，将会产生很大的电流流过两个管子，从而可能导致电路损坏。直到一个管子完全截止才会退出此状态(图 15.9(a)中的(1)、(2)、(3)和(4)处所示)。为了避免开关期间产生这种具有破坏性的直通电流，可使其中一个管子关闭后再打开另一个管子。在半桥输出模式下，使用一个可编程死区延时模块，来避免产生的直通电流破坏电路。3 个通道共用一个死区设置。如图 13.9 (b) 所示，该延时在 PWM5 信号从非有效电平到有效电平转换时发生。延时时间通过寄存器 PWM5CTL1 的低 7 位进行设置。延时时间计数公式如下式所示。

$$\text{延时时间} = \text{P5DC}<6:0> \cdot T$$

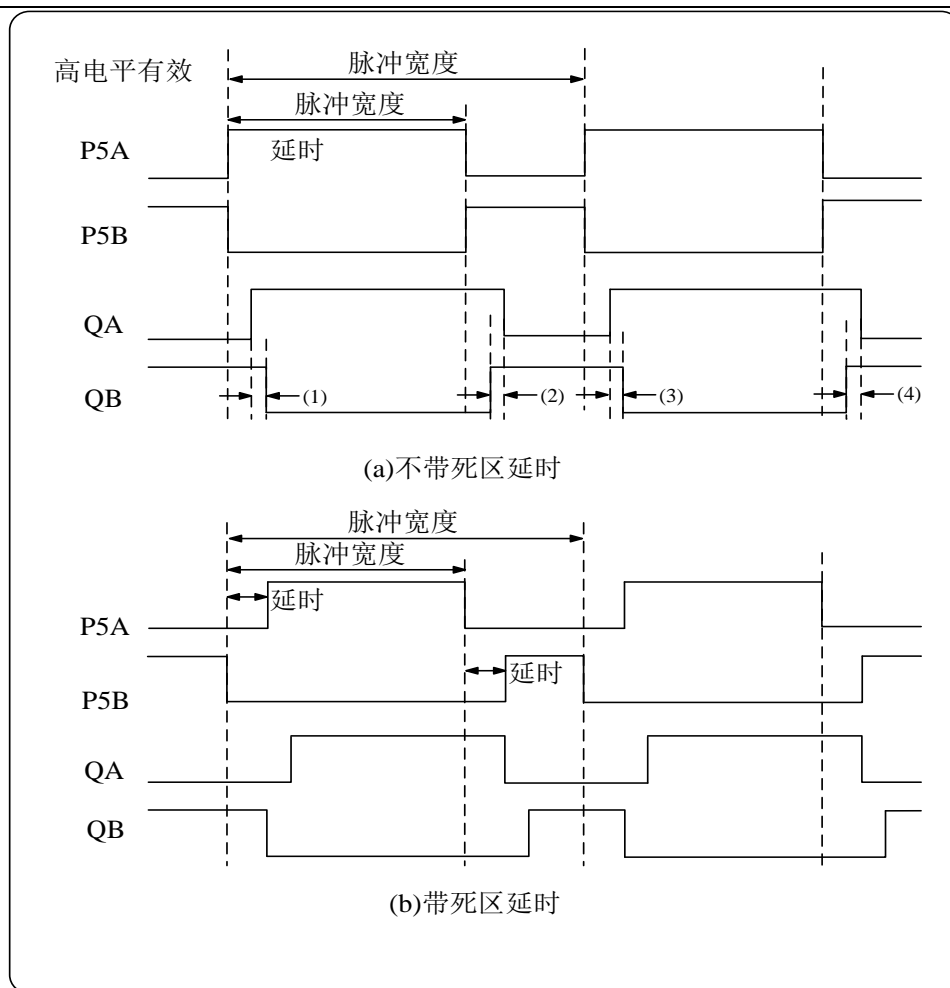


图 13.9 带死区控制和不带死区控制时的信号示例

### 13.4.7 全桥输出模式

通道 1 还可以实现全桥输出模式。全桥输出模式有全桥正向输出模式和全桥反向输出模式两种。通过将寄存器 PWM5CTL0 的 P5CH1MOD<1:0> 设置为 01，把 PWM5 设置为全桥正向输出模式；将其设置为 11，则把 PWM5 设置为全桥反向输出模式。

在全桥输出模式下，P5A、P5B、P5C 和 P5D 四个引脚都用作输出。将其设置为全桥正向输出模式时，引脚 P5A 被设置为有效电平，引脚 P5D 为 PWM 调制信号，P5B 和 P5C 为无效电平，图 13.10 (a) 和图 13.11 (a) 为全桥正向输出模式引脚信号示例。将其设置为全桥反向输出模式时，P5C 被驱动为有效电平，引脚 P5B 为 PWM 调制信号，而 P5A 和 P5D 为无效电平，图 13.10 (b) 和图 13.11 (b) 为全桥反向输出模式引脚信号示例。

图 13.12 给出了全桥输出模式的应用电路示例。

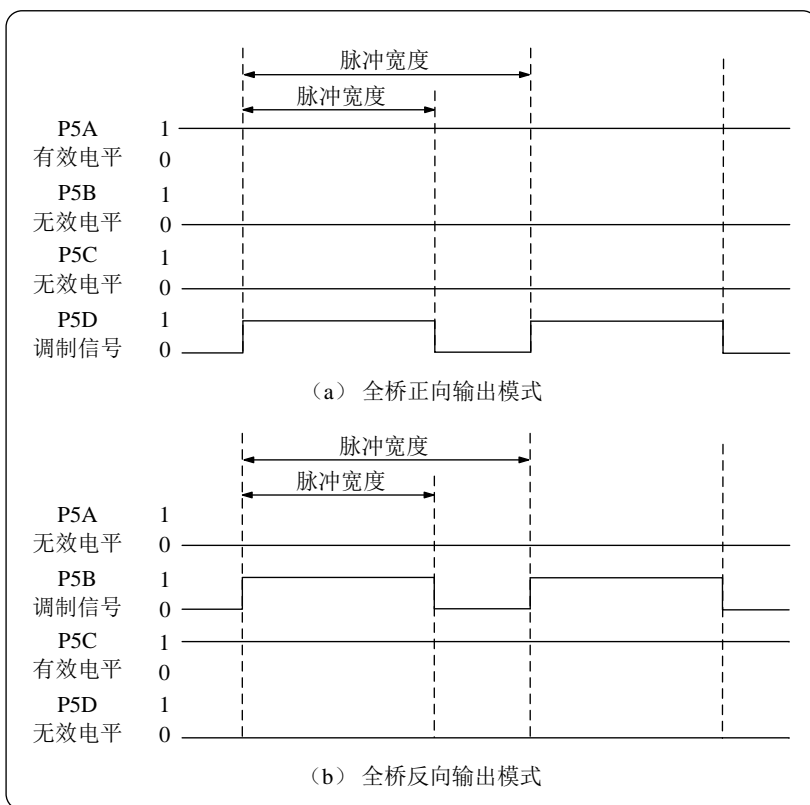


图 13.10 全桥输出模式引脚信号示例(高电平有效)

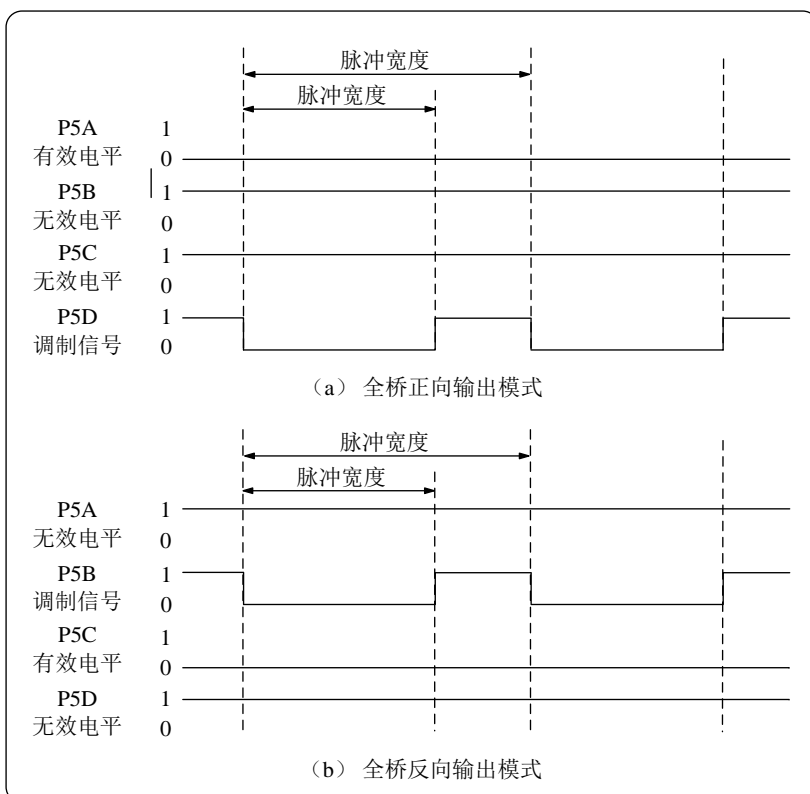


图 13.11 全桥输出模式引脚信号示例(低电平有效)

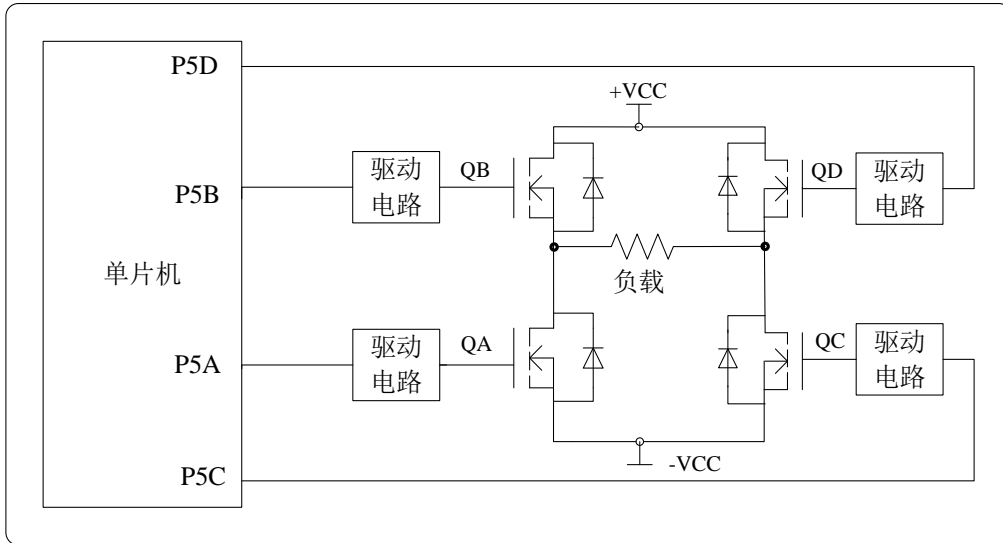


图 13.12 全桥输出模式应用电路示例

在使用全桥输出模式时，需将 P5A、P5B、P5C 和 P5D 引脚对应的方向控制位 TR<sub>xx</sub> 清零，设置为输出。

从全桥输出模式引脚信号波形以及应用电路示例可以看出：正向模式时，调制输出仅能控制一组开关管，假设此时流经负载的电流为正，则反向模式控制另一组开关管，使流经负载的电流为负。因此在使用全桥输出模式时，可将正向模式应与反向模式配合使用，使流经负载的电流方向改变。

通过将寄存器 PWM5CTL0 的 P5CHMOD<1:0>位设置为 01(正向模式)和 11(反向模式)来改变流经负载的电流。如图 15.13 所示，从一种模式模式切换到另一种模式时，在前一种模式最后一个周期结束之前，P5B(或 P5D)被切换到无效状态，P5A(或 P5C)被切换到相反的状态。图中时间  $T = 1/F_{clk} \cdot T_{10}$  预分频值。(F<sub>clk</sub> 为 T10 工作时钟频率)

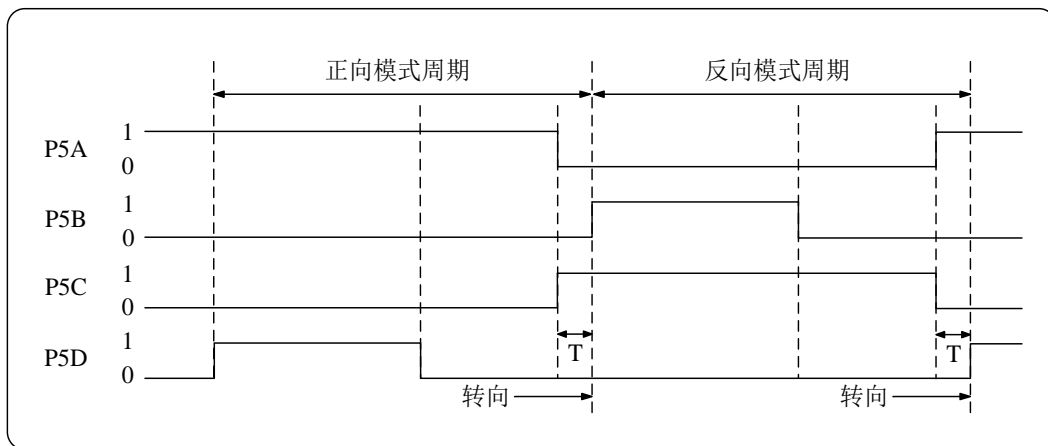


图 13.13 全桥正向、反向输出模式转换示例(高电平有效)

全桥输出模式下没有死区延时功能。通常在此模式中，任何时间只调制一对输出，因此不会导致电路产生直通电流，所以不需要死区延时。然而，当 PWM 的占空比接近百分之百，且开关管导通时间小于截止时间时，将会导致电路产生直通电流。图 13.14 为此

情况下各处信号示例。

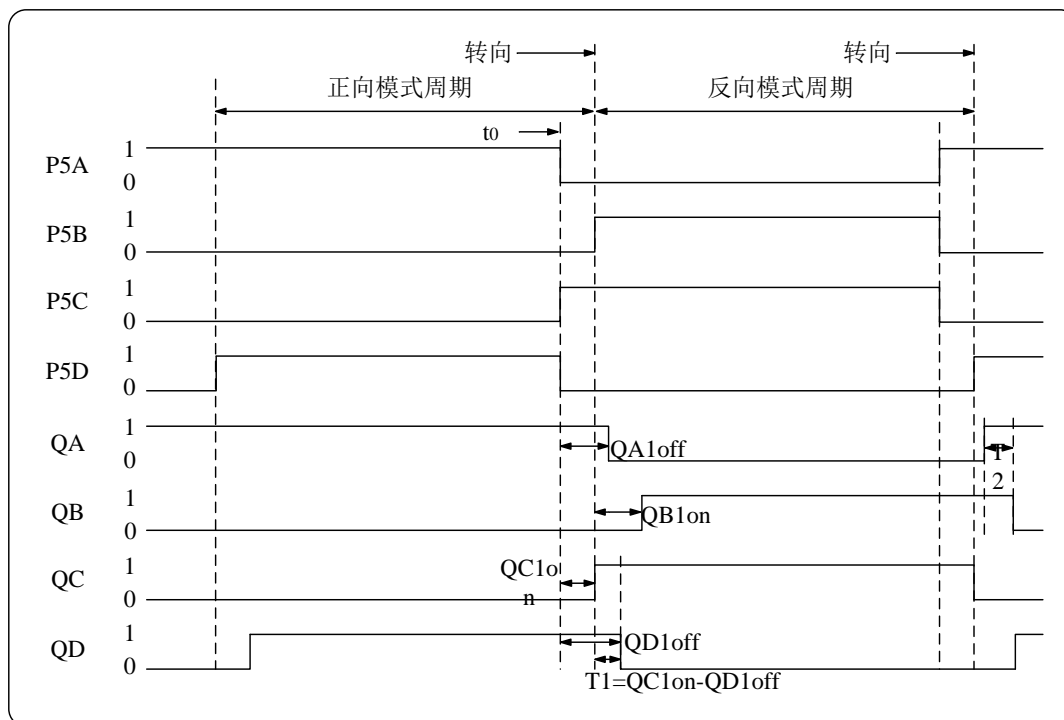


图 13.14 接近满占空比时换向时信号示例

图 13.14 中在  $t_0$  时刻 P5A 和 P5D 变为无效，P5C 为有效。QA、QB、QC 和 QD 分别为四路 PWM5 输出控制的开关管，QAoff 和 QDoff 为开关管的截止延时时间，QBon 和 QCon 为导通延时时间，由于管子导通和截止时间的差异，在第一次换向时，开关管 QC 和 QD 产生直通电流，持续时间为  $T_1$ ，在后面一次换向时 QA 和 QB 产生直通电流，持续时间为  $T_{10}$ 。

为了消除这种问题，可将换向前一个 PWM 信号周期的占空比调小，或者选用开关速度快的驱动电路，或者其它方法。

### 13.4.8 脉冲转向控制

通过寄存器脉冲转向寄存器 P5STRCTL0/P5STRCTL1 可将相应的引脚设置成 PWM5 输出引脚或者通用 IO 引脚。通过寄存器 P5STRCTL0 中的 P5STRSYNC 位可以控制脉冲转向功能切换是与同步模式还是立即有效（与指令周期同步）。通过 P5STRENA0/P5STRENA1/P5STRENA2/P5STRENA3/P5STRENA4/P5STRENA5/P5STRENA6/P5STRENA7 位可以设置相应端口引脚功能切换。默认情况下，通道 1/2/3 的 Ax 输出为 PWM5 引脚输出，Bx(C/D)为 IO 口引脚。

注意，通道 1 的脉冲转向控制只在单输出模式下有效，而通道 2/3 的在任何模式下都有效，当使用 PWM5 输出若不需要太多端口时，可自由屏蔽通道 2/3 的输出而不影响 IO 口功能。

当 P5STRSYNC 为 0 时，对应引脚的输出在 P5STRENAx 置 1 后立即更新；当 P5STRSYNC 为 1 时，对应引脚的输出由 UDEN/UDEVTO 控制。只有当 UDEN 为 1 时，对应引脚的 P5STRENAx 的设置才能生效。当 UDEN 为 1 时，通过 UDEVTO 可以控制同步更新的时机，当 UDEVTO 为 0 时，对应引脚的输出与 PWM 周期同步更新；当 UDEVTO 为 1



时，对应引脚的输出立即更新。通过 UDEN/UDEVTO 可以实现脉冲转向的同步更新操作。

### 13.4.9 输出控制

在半桥模式下，可以通过 PWM5OC 输出控制寄存器控制引脚的输出。当 PWM5OC 寄存器中的 OCx 设置为 0 时，相应的引脚选择 PWM 输出，输出极性由 PWM5PC 极性控制寄存器控制；当 OCx 设置为 1 时，相应的引脚选择强制信号输出，输出电平由 PWM5FC 强制控制寄存器控制。

输出控制为带缓冲模式，通过 PWM5CTL2 更新控制寄存器可以控制输出控制寄存器缓冲的更新。详见 PWM 更新锁定。

### 13.4.10 强制输出模式

通过将 PWM5OC 输出控制寄存器中相应输出引脚设置成 1 可以选择 P5A/ P5B/ P5A2/ P5B2/ P5A3/ P5B3 输出强制信号。在该模式下，通过 PWM5FC 强制控制寄存器可以设置 P5A/ P5B/ P5A2/ P5B2/ P5A3/ P5B3 端口输出高电平或低电平，强制输出不受极性控制的影响。

强制输出控制为带缓冲模式，通过 PWM5CTL2 更新控制寄存器可以控制强制输出的更新。详见 PWM 更新锁定。

### 13.4.11 PWM 输出极性模式

通过将 PWM5OC 输出控制寄存器中相应输出引脚设置成 0 可以选择 P5A/ P5B/ P5A2/ P5B2/ P5A3/ P5B3 输出 PWM 信号。在该模式下，通过 PWM5PC 极性控制寄存器可以设置 P5A/ P5B/ P5A2/ P5B2/ P5A3/ P5B3 端口输出的极性。

当 PWM 输出极性选择高有效时，P5A/ P5A2/ P5A3 输出在周期开始输出为高电平，当占空比与定时器匹配时，输出翻转；而互补的输出 P5B/ P5B2/ P5B3 则在周期开始输出为低电平，当占空比与定时器匹配时，输出翻转。

需要注意的是，通道 1 的 P5A/P5B 有两个极性控制使能，一个是 PWM5CTL0 中的 P5MOD<3:0>，一个是 PWM5PC 中的 PCA/PCB。这两个寄存器相应的控制位共同影响产生最后的极性控制结果。如前述表 15-5 所示。而 P5C/P5D 则只由 P5MOD<3:0>控制。P5A/ P5B/ P5C/ P5B 的极性控制需要在 UDEN 为 1 时才允许更新。

PWM 输出极性控制为带缓冲模式，通过 PWM5CTL2 更新控制寄存器可以控制强制输出的更新。PWM 极性更新有两种形式，一种是立即更新，一种是定时器为 0 时更新。详见 PWM 更新锁定。

### 13.4.12 PWM 更新锁定

通过设置 PWM5CTL2 更新控制寄存器的 UDEN 可以使能或禁止占空比、周期、极性控制、强制控制、输出控制和脉冲转向控制的更新。当 PWM5CTL2 寄存器的 UDEN 为 0 时，占空比、周期、极性控制、强制控制和输出控制不能更新，这使得用户能够在新值生效前将所需要的值写入到相应的寄存器中。当 UDEN 为 1 时，可以通过 UDEVTO 来控制占空比、周期、极性控制、强制控制和输出控制的更新到缓冲中。UDEVTO 控制周期和占空比的更新，UDEVTO 控制输出控制、极性控制和强制控制寄存器的更新。若 UDEVTO 为 0，则周期和占空比可以在 T10 为 0 时更新到缓冲中；若 UDEVTO 为 1 时，则立即更新周期和占空

比到缓冲中，T10 被清。若 UDEVT0 为 0，则输出控制、极性控制和强制控制寄存器可以在定时器为 0 时更新到缓冲中；若 UDEVT0 为 1 时，则立即更新输出控制、极性控制和强制控制寄存器到缓冲中，此时 T10 继续计数，不影响占空比的完整性。

脉冲转向控制中，当 P5STRSYNC=1 时，脉冲转向更新与极性控制、强制控制、输出控制一致，都是由 UDEVT0 控制，且 UDEN 为更新总使能。当 P5STRSYNC=0 时，脉冲转向与 UDEN/UDEVT0 无关，当脉冲转向控制 P5STRENA<sub>x</sub> 设置后输出立即改变。

注意，当 UDEVT1 为 1 时，会清零定时器，并清零死区定时器，此时若 UDEVT0 为 0，也会引起极性控制、强制控制、输出控制和脉冲转向控制的更新。UDEVT0/1 置 1 后必须软件清零再置 1 才能使能下一次的立即更新控制。

### 13.4.13 PWM 复位控制

PWM5 的复位控制使得在复位状态下，可以通过将配置位中的 PWMPIN 设置成 I/O 口控制或是选择引脚的有效极，但是只有和 PWM5 功能相关的 APFCTL<sub>x</sub> 寄存器复位值对应的引脚才会被控制。引脚的极性状态可以通过控制配置位中的 HPOL 和 LPOL 输出所需要的极性。配置 HPOL 可以设置 P5A、P5C、P5A2/3 的极性，配置 LPOL 可以设置 P5B、P5D、P5B2/3 的极性。

### 13.4.14 自动关断和自动重启模式

#### 13.4.14.1 自动关断模式

PWM5 模块具有自动关断功能。如图 13.15 所示，为自动关断模式的原理框图。使能自动关断功能后，在外部关断事件发生时，该功能自动禁止 PWM 输出，然后将 P5A、P5B、P5C、P5D、P5A2、P5B2、P5A3、P5B3 八个引脚输出电平置于其预定义的状态。此模式用于防止 PWM 破坏应用电路。

自动关断模式具有 3 个关断源:INT0 引脚的逻辑低电平、比较器 1 输出高电平和在软件中直接将 P5ASE 位置 1。关断源触发关断的信号是高电平或低电平，而不是上升沿或下降沿，只要关断源的关断电平存在，自动关断状态将保持。

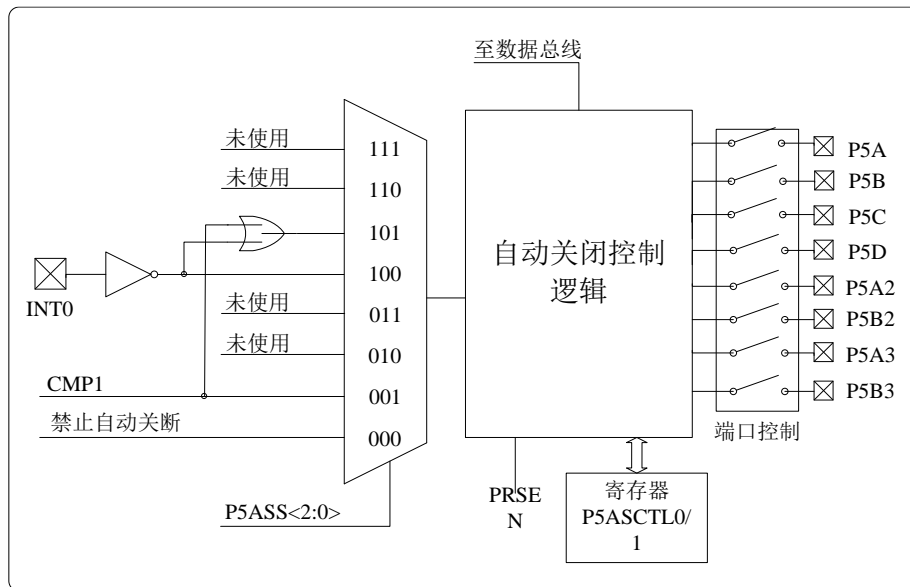


图 13.15 自动关断模式原理框图

通过寄存器 P5ASCTL0 的 P5ASS<2:0>位选择自动关断源。将 P5ASS<2:0>位设置为 000 时，关闭自动关断功能。

寄存器 P5ASCTL0 中的 P5ASE 位指示关断的状态。如果该位为 0，表示 PWM5 的八个引脚输出正常的 PWM 信号，如果该位为 1，表示 PWM 的八路输出处于关断状态。

发生关断事件时，将会：

- P5ASE 位被置 1。直到被软件清零或发生自动重启才会将该位清零。
- 使能的八个 PWM 引脚将被置于关断电平状态。

关断时，通道 1 四路输出电平的状态由寄存器 P5ASCTL0 的 P5SSAC<1:0>和 P5SSBD<1:0>位决定。通道 2 和通道 3 输出电平状态由寄存器 P5ASCTL1 的 P5SSA2<1:0>、P5SSB2<1:0>、P5SSA3<1:0>和 P5SSB3<1:0>位决定。通过设定可将输出引脚置为：三态、高电平和低电平三种状态。其中 P5A 和 P5C 的状态由 P5SSAC<1:0>设置，P5B 和 P5D 的状态由 P5SSBD<1:0>设置，P5A2 状态由 P5SSA2<1:0>设置，P5B2 状态由 P5SSB2<1:0>设置，P5A3 状态由 P5SSA3<1:0>设置，P5B3 状态由 P5SSB3<1:0>设置。

打开自动关断功能后，如果关断源产生关断事件，则 P5ASE 标志位被硬件置 1，四路输出被驱动为关断模式电平；关断源清除关断事件后，P5ASE 仍然为 1(如果 P5RSEN=0)，四路输出仍然为关断模式电平，直到将 P5ASE 位清零，PWM 重启。

#### 13.4.14.2 自动重启模式

可将 PWM5 配置为一旦清除自动关断条件就自动重启 PWM。通过将 PWM5CTL1 寄存器中的 P5RSEN 位置 1 使能自动重启。

如果使能自动重启，只要自动关断条件有效，P5ASE 位就将保持置 1。当清除自动关断条件时，将通过硬件将 P5ASE 位清零，并且将恢复常规操作。

如图 13.16 所示，在图(a)中，P5RSEN=0，自动重启模式关闭，通过软件将 P5ASE 标志位清零，PWM 才会重启。在图(b)中，P5RSEN=1，自动重启模式被打开，关断事件被清除后，P5ASE 标志位由硬件自动清零，然后重启 PWM。

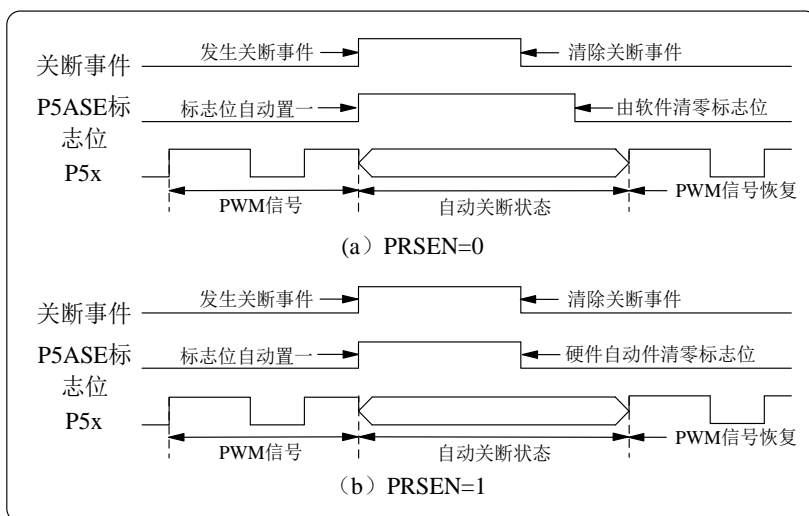


图 13.16 自动重启和软件重启 PWM 示例

## 14 模拟比较器模块

KF8A100 含有 4 路带数字滤波功能和中断功能的模拟比较器，其中模拟比较器的正端输入为 I/O 端口，负端输入可选择 I/O 端口、电阻分压及零点检测电压。当  $INV_x=0$  时，输出极性为正，此时当比较器的正端电压大于负端电压时，比较器输出高电平；当比较器的正端电压小于负端电压时，比较器输出低电平。比较器在不使用时应关闭比较器的使能，此时比较器输出为低电平。

该模块的主要特点如下：

- 多路比较器
- 多输入可选
- 电阻分压模块
- 输出极性可选
- 中断边沿可选
- 数字滤波功能
- 范围控制

### 14.1 模拟比较器原理

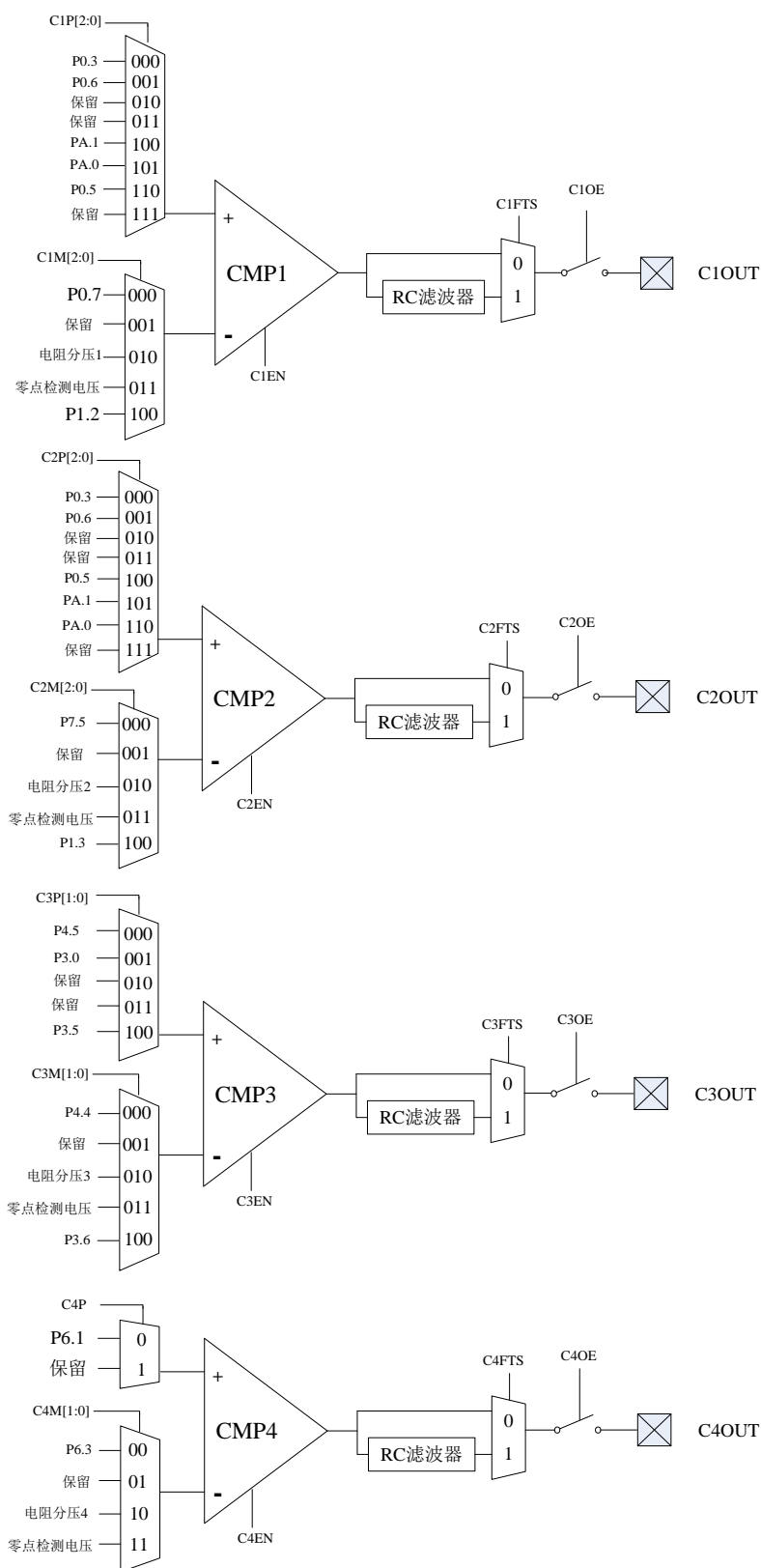


图 14.1 模拟比较器框图

## 14.2 与模拟比较器相关的寄存器

表 14.1 与比较器相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
19H	CMCTL0	-	-	-	-	C4EDG	C3EDG	C2EDG	C1EDG
1AH	C1CTL	C1EN	C1OE	C1P2	C1P1	C1P0	C1M2	C1M1	C1M0
1DH	COUT	-	VCEN	-	-	C4OUT	C3OUT	C2OUT	C1OUT
31H	C2CTL	C2EN	C2OE	C2P2	C2P1	C2P0	C2M2	C2M1	C2M0
239H	C3CTL	C3EN	C3OE	C3P2	C3P1	C3P0	C3M2	C3M1	C3M0
23AH	C4CTL	C4EN	-	-	-	C4P0	-	C4M1	C4M0
12FH	VRES0	VRES23	VRES22	VRES21	VRES20	VRES13	VRES12	VRES11	VRES10
14FH	VRES1	VRES43	VRES42	VRES41	VRES40	VRES33	VRES32	VRES31	VRES30
68H	CCTCTL	CCTEN	CSEL1	CSEL0	CRSET	CFSET	CCT7	CCT3	CCT1

## 14.2.1 比较器中断触发控制寄存器 CMCTL0

寄存器: CMCTL0: CMCTL0控制寄存器(地址: 19H)

复位值	bit7				bit0			
---- 0000	-	-	-	-	C4EDG	C3EDG	C2EDG	C1EDG
	U	U	U	U	R/W	R/W	R/W	R/W

C4EDG : 比较器 4 边沿触发中断选择位

1=下降沿触发中断

0=上升沿触发中断

C3EDG : 比较器 3 边沿触发中断选择位

1=下降沿触发中断

0=上升沿触发中断

C2EDG : 比较器 2 边沿触发中断选择位

1=下降沿触发中断

0=上升沿触发中断

C1EDG : 比较器 1 边沿触发中断选择位

1=下降沿触发中断

0=上升沿触发中断

## 14.2.2 比较器 1 控制寄存器 C1CTL

寄存器: C1CTL: 比较器1控制寄存器(地址:1AH)

复位值 0000 0000	bit7						bit0	
	C1EN	C1OE	C1P2	C1P1	C1P0	C1M2	C1M1	C1M0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

C1EN: 模拟比较器 1 使能位

0 = 关闭比较器 1

1 = 使能比较器 1

C1OE: 模拟比较器 1 输出使能位

0 = 禁止比较器 1 输出到管脚

1 = 使能比较器 1 输出到管脚

C1P[2:0]: 模拟比较器 1 的正输入端选择位

000 = P0.3 作为 CMP1 的正端输入

001 = P0.6 作为 CMP1 的正端输入

010 = 保留

011 = 保留

100 = PA.1 作为 CMP1 的正端输入

101 = PA.0 作为 CMP1 的正端输入

110 = P0.5 作为 CMP1 的正端输入

111 = 保留

C1M[2:0]: 模拟比较器 1 的负输入端选择位

000 = P0.7 作为 CMP1 的负端输入

001 = 保留

010 = 电阻分压作为 CMP1 的负端输入

011 = 零点检测电压作为 CMP1 的负端输入

100 = P1.2 作为 CMP1 的负端输入

注:

电阻分压通过寄存器 VRES 选择分压比;

零点检测分压通过寄存器 ZPDCTL1 和 ZPDCTL2 控制, 见零点检测章节。



## 14.2.3 比较器结果寄存器 COUT

寄存器: COUT: 比较器结果寄存器(地址:1DH)

	bit7				bit0			
复位值 -0-- 0000	-	VCEN	-	-	C4OUT	C3OUT	C2OUT	C1OUT
	U	R/W	U	U	R	R	R	R

VCEN: 保留位

C4OUT: 比较器 4 输出状态位

C3OUT: 比较器 3 输出状态位

C2OUT: 比较器 2 输出状态位

C1OUT: 比较器 1 输出状态位

## 14.2.4 比较器 2 控制寄存器 C2CTL

寄存器: C2CTL: 比较器2控制寄存器(地址:31H)

	bit7							bit0
复位值 0000 0000	C2EN	C2OE	C2P2	C2P1	C2P0	C2M2	C2M1	C2M0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

C2EN: 模拟比较器 2 使能位

0 = 关闭比较器 2

1 = 使能比较器 2

C2OE: 模拟比较器 2 输出使能位

0 = 禁止比较器 2 输出到管脚

1 = 使能比较器 2 输出到管脚

C2P[2:0]: 模拟比较器 2 的正输入端选择位

000 = P0.3 作为 CMP2 的正端输入

001 = P0.6 作为 CMP2 的正端输入

010 = 保留

011 = 保留

100 = PA.1 作为 CMP2 的正端输入

101 = PA.0 作为 CMP2 的正端输入

110 = P0.5 作为 CMP2 的正端输入

111 = 保留

C2M[2:0]: 模拟比较器 2 的负输入端选择位

000 = P7.5 作为 CMP2 的负端输入

001 = 保留

010 = 电阻分压作为 CMP2 的负端输入

011 = 零点检测电压作为 CMP2 的负端输入

100 = P1.3 作为 CMP2 的负端输入

注:

电阻分压通过寄存器 VRES 选择分压比；

零点检测分压通过寄存器 ZPDCTL1 和 ZPDCTL2 控制，见零点检测节。

### 14.2.5 比较器 3 控制寄存器 C3CTL

寄存器: C3CTL: 比较器3控制寄存器(地址:239H)

复位值	bit7						bit0	
0000 0000	C3EN	C3OE	C3P2	C3P1	C3P0	C3M2	C3M1	C3M0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

C3EN: 模拟比较器 3 使能位

0 = 关闭比较器 3

1 = 使能比较器 3

C3OE: 模拟比较器 3 输出使能位

0 = 禁止比较器 3 输出到管脚

1 = 使能比较器 3 输出到管脚

C3P[2:0]: 模拟比较器 3 的正输入端选择位

000 = P4.5 作为 CMP3 的正端输入

001 = P3.0 作为 CMP3 的正端输入

010 = 保留

011 = 保留

100 = P3.5 作为 CMP3 的正端输入

C3M[2:0]: 模拟比较器 3 的负输入端选择位

000 = P4.4 作为 CMP3 的负端输入

001 = 保留

010 = 电阻分压作为 CMP3 的负端输入

011 = 零点检测电压作为 CMP3 的负端输入

100 = P3.6 作为 CMP3 的负端输入

注:

电阻分压通过寄存器 VRES 选择分压比；

零点检测分压通过寄存器 ZPDCTL1 和 ZPDCTL2 控制，见零点检测章节。

### 14.2.6 比较器 4 控制寄存器 C4CTL

寄存器: C4CTL: 比较器4控制寄存器(地址:23AH)

复位值 00000000	bit7				bit0			
	C4EN	-	-	-	C4P0	-	C4M1	C4M0
	R/W	R/W	R/W	R/W	U	R/W	R/W	R/W

C4EN: 模拟比较器 4 使能位

0 = 关闭比较器 4

1 = 使能比较器 4

C4P0: 模拟比较器 4 的正输入端选择位

0 = P6.1 作为 CMP4 的正端输入

1 = 保留

C4M[1:0]: 模拟比较器 4 的负输入端选择位

00 = P6.3 作为 CMP4 的负端输入

01 = 保留

10 = 电阻分压作为 CMP4 的负端输入

11 = 零点检测电压作为 CMP4 的负端输入

注:

电阻分压通过寄存器 VRES 选择分压比;

零点检测分压通过寄存器 ZPDCTL1 和 ZPDCTL2 控制, 见零点检测章节。

### 14.3 零点检测电压

KF8A100 集成的 4 个比较器均可选择零点检测电压作为负端输入, 零点检测电压通过 ZPDCTL1 和 ZPDCTL2 寄存器的 ZPDxs<2:0>位来设置, 相关配置如下表所示:

表 14.2 比较器零点检测电压设置

ZPDxs<2:0>	电阻分压值
000	0.15V
001	0.25V
010	0.35V
011	0.45V
100	0.55V

注: ZPD0S<2:0>分配给比较器 1 电阻分压设置, ZPD1S<2:0>分配给比较器 2 电阻分压设置, ZPD2S<2:0>分配给比较器 3 电阻分压设置, ZPD3S<2:0>分配给比较器 4 电阻分压设置。

### 14.4 电阻分压

电阻分压档可以为比较器负端提供较为宽泛的内部电压档位。

电阻分压的电压值通过如下寄存器 VRES0/VRES1 配置, RESx<3:0>控制比较器 x 的电阻分压, x=1/2/3/4。

## 14.4.1 电阻分压配置寄存器 0

寄存器: VRES0:电阻分压配置寄存器0(地址: 12FH)

复位值	bit7						bit0	
0000 0000	VRES23	VRES22	VRES21	VRES20	VRES13	VRES12	VRES11	VRES10
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

VRES2&lt;3:0&gt; : 比较器 2 电阻分压配置位

0000 = 0.05VDD

0001 = 0.1VDD

0010 = 0.15VDD

0011 = 0.2VDD

0100 = 0.25VDD

0101 = 0.3VDD

0110 = 0.35VDD

0111 = 0.4VDD

1000 = 0.45VDD

1001 = 0.5VDD

1010 = 0.55VDD

1011 = 0.6VDD

1100 = 0.65VDD

1101 = 0.7VDD

1110 = 0.75VDD

1111 = 0.8VDD

VRES1&lt;3:0&gt; : 比较器 1 电阻分压配置位

0000 = 0.05VDD

0001 = 0.1VDD

0010 = 0.15VDD

0011 = 0.2VDD

0100 = 0.25VDD

0101 = 0.3VDD

0110 = 0.35VDD

0111 = 0.4VDD

1000 = 0.45VDD

1001 = 0.5VDD

1010 = 0.55VDD

1011 = 0.6VDD

1100 = 0.65VDD

1101 = 0.7VDD

1110 = 0.75VDD

1111 = 0.8VDD

## 14.4.2 电阻分压配置寄存器 1

寄存器: VRES1:电阻分压配置寄存器1(地址: 14FH)

复位值	bit7						bit0	
0000 0000	VRES43	VRES42	VRES41	VRES40	VRES33	VRES32	VRES31	VRES30
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

VRES4&lt;3:0&gt; : 比较器 4 电阻分压配置位

0000 = 0.05VDD  
 0001 = 0.1VDD  
 0010 = 0.15VDD  
 0011 = 0.2VDD  
 0100 = 0.25VDD  
 0101 = 0.3VDD  
 0110 = 0.35VDD  
 0111 = 0.4VDD  
 1000 = 0.45VDD  
 1001 = 0.5VDD  
 1010 = 0.55VDD  
 1011 = 0.6VDD  
 1100 = 0.65VDD  
 1101 = 0.7VDD  
 1110 = 0.75VDD  
 1111 = 0.8VDD

VRES3&lt;3:0&gt; : 比较器 3 电阻分压配置位

0000 = 0.05VDD  
 0001 = 0.1VDD  
 0010 = 0.15VDD  
 0011 = 0.2VDD  
 0100 = 0.25VDD  
 0101 = 0.3VDD  
 0110 = 0.35VDD  
 0111 = 0.4VDD  
 1000 = 0.45VDD  
 1001 = 0.5VDD  
 1010 = 0.55VDD  
 1011 = 0.6VDD  
 1100 = 0.65VDD  
 1101 = 0.7VDD  
 1110 = 0.75VDD  
 1111 = 0.8VDD

## 14.5 滤波器相关寄存器

表 14.3 与比较器相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
231H	FILT1CTL	INV1	WEN1	FEN1	F1CS1	F1CS0	FILTCN T12	FILTCNT 11	FILTCNT1 0
232H	FILT1PER	FP17	FP16	FP15	FP14	FP13	FP12	FP11	FP10
233H	FILT2CTL	INV2	WEN2	FEN2	F2CS1	F2CS0	FILTCN T22	FILTCNT 21	FILTCNT2 0
234H	FILT2PER	FP27	FP26	FP25	FP24	FP23	FP22	FP21	FP20
235H	FILT3CTL	INV3	WEN3	FEN3	F3CS1	F3CS0	FILTCN T32	FILTCNT 31	FILTCNT3 0
236H	FILT3PER	FP37	FP36	FP35	FP34	FP33	FP32	FP31	FP30
237H	FILT4CTL	INV4	WEN4	FEN4	F4CS1	F4CS0	FILTCN T42	FILTCNT 41	FILTCNT4 0
238H	FILT4PER	FP47	FP46	FP45	FP44	FP43	FP42	FP41	FP40

## 14.5.1 滤波器控制寄存器 FILTxCTL

寄存器: FILTxCTL: 滤波器控制寄存器

复位值	bit7					bit0		
0000 0000	INVx	WENx	FENx	FxCS1	FxCS0	FILTCNT x2	FILTCNT x1	FILTCNT x0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- INVx: 比较器 x 输出极性选择位 comparator1 invert bit  
 0 = 比较器正常输出:  $CIN+ > CIN- = 1$ ,  $CIN+ < CIN- = 0$   
 1 = 比较器输出极性转换:  $CIN+ > CIN- = 0$ ,  $CIN+ < CIN- = 1$
- WENx: 比较器 x 范围控制使能位 window control enable bit  
 0 = 禁止范围控制功能  
 1 = 使能范围控制功能
- FENx: 滤波器 x 使能位 filter block enable bit  
 0 = 禁止滤波功能  
 1 = 使能滤波功能
- FxCS<1:0>: 滤波器x滤波时钟源选择位  
 00 = 滤波时钟源为系统时钟  
 01 = 滤波时钟源为高频外设时钟  
 1x = 滤波时钟源为低频外设时钟
- FILTCNTx<2:0>: 滤波器 x 取样数量选择位 filter sample count bits  
 000 = 滤波器 x 不采样, 禁止滤波功能  
 001 = 滤波器 x 采样 1 次, 并输出结果  
 010 = 滤波器 x 连续 2 次采样结果一致, 输出采样值, 否则保持;  
 011 = 滤波器 x 连续 3 次采样结果一致, 输出采样值, 否则保持;  
 100 = 滤波器 x 连续 4 次采样结果一致, 输出采样值, 否则保持;

101 = 滤波器 x 连续 5 次采样结果一致，输出采样值，否则保持；

110 = 滤波器 x 连续 6 次采样结果一致，输出采样值，否则保持；

111 = 滤波器 x 连续 7 次采样结果一致，输出采样值，否则保持；

注：其中 x = 1/2/3/4。

### 14.5.2 滤波器 x 采样时钟分频寄存器

寄存器: FILTxPER: 滤波器x采样时钟分频寄存器

	bit7							bit0
复位值 0000 0000	FPx7	FPx6	FPx5	FPx4	FPx3	FPx2	FPx1	FPx0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

$$\text{滤波器x采样时钟频率} = \frac{\text{Filter\_Clock}}{\text{FPx}<7:0>+1}$$

注：（1）注：其中 x = 1/2/3/4。

（2）当 FPx<7:0>=0 时，滤波采样时钟即为滤波时钟源 Filter\_Clock。

### 14.6 极性选择

模拟比较器的输出状态可以通过 CxFILTCTL 寄存器的 INVx 位结果选择，以比较器 1 为例，如下表所示。INV1=0, 比较器正常输出: C1IN+>C1IN-=1, C1IN+<C1IN-=0; INV1=1, 比较器输出极性转换: C1IN+>C1IN-=0, C1IN+<C1IN-=1。

表 14.4 比较器 1 输出状态与输入状态

输入条件	极性选择	比较器输出
C1IN+ > C1IN-	INV1 = 0	C1OUT = 1
C1IN+ > C1IN-	INV1 = 1	C1OUT = 0
C1IN+ < C1IN-	INV1 = 0	C1OUT = 0
C1IN+ < C1IN-	INV1 = 1	C1OUT = 1

注：（1）模拟比较器 2/3/4 的输出状态控制同 1。

（2）相应的极性控制位为 INV2/3/4。

### 14.7 范围控制功能

范围控制功能是利用系统时钟作为筛选时钟，当时钟电平为高时，比较器结果输出；当时钟电平为低时，比较器结果保持。通过 FILTxCTL 寄存器的 WENx 位来使能/禁止范围控制功能。

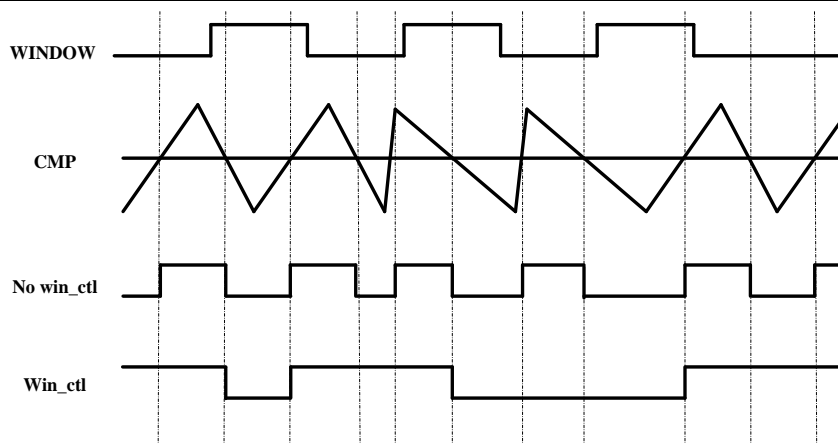


图 14.2 模拟比较器范围控制

## 14.8 滤波功能

当比较器的输入电压的压差很小时，比较器的输出会发生震荡。内部和外部的寄生效应，和信号线、电源线与其它系统部分间的信号耦合造成了比较器的输出震荡。输出震荡降低了比较器的精度，影响了比较输出的结果。通过 FILTxCTL 寄存器的 FENx 位选择将比较器的输出通过滤波器，可以减少比较器震荡带来的误差。

**滤波器使用方法：**

- ◆ 通过 FILTxCTL 寄存器的 FENx 位来使能滤波功能；
- ◆ 通过 FPx<7:0>位选择滤波器的采样时钟；
- ◆ 通过 FxCNT<2:0>位设置滤波器的采样次数。

## 14.9 比较器中断

当比较器的输出电平发生变化时，且变化的方向与 CMCTL0 寄存器的 CxEDG 边沿触发选择位所设置的一致时，比较器就会使相应的中断标志位置 1。CxIF 位是比较器中断标志位，必须由软件清零，因为可以人为向该位写 1，所以也可以模拟中断的产生。

PCTL 的 IPEN 为优先级中断使能位。当 IPEN 为 0 时，即在中断普通模式下，EIE2 寄存器的 CxIE 位和 INTCTL 寄存器的 PUIE 和 AIE 位必须置 1，以允许比较器中断；当 IPEN 为 1 时，即在中断优先级模式下，EIEx 寄存器的 CxIE 位、IPx 寄存器的 PCx 置 1 和 INTCTL 寄存器的 AIEH 位必须置 1，以允许比较器高优先级中断；EIEx 寄存器的 CxIE 位、IPx 寄存器的 PCx 清零和 INTCTL 寄存器的 AIEH 和 AIEL 位置 1，以允许比较器低优先级中断。

在中断服务程序中，用户可以通过读写 CMCTL0 寄存器清零 C1IF 中断标志位，也可以通过软件清零。而 C2IF 中断标志位只能通过软件清零。

## 14.10 比较器使用

**以比较器 1 为例，使用比较器时需要进行下列设置：**

1. 分别通过 C1CTL 寄存器的 C1P<1:0>位和 C1M<2:0>位选择比较器 1 的正负端口，如选择端口为 IO 口，将相应的引脚设置成模拟输入引脚；如负端选择端口为电阻分压，通过 VRES0 寄存器的 VRES1<3:0>位设置电阻分压项；如负端选择为零点检测，详细描述见[零点检测](#)章节；



2. 如需比较器 1 结果输出到 IO 口，请将 C1CTL 寄存器的 C1OE 位置 1 使能输出；
3. 配置 FILT1CTL 寄存器的 INV1 位选择比较器输出极性；
4. 配置 FILT1CTL 寄存器的 WEN 位及 FEN 位自由选择比较器 1 的范围控制功能和滤波功能；
5. 通过 C1CTL 寄存器的 C1EN 位使能比较器 1。

## 14.11 比较器清零定时器

KF8A100 的比较器带有硬件清零定时器（T1/T3/T7）的计数寄存器功能。用户可通过 CCTCTL 寄存器进行相关配置。

### 14.11.1 比较器清零定时器控制寄存器 CCTCTL

**寄存器：** CCTCTL: 比较器清零定时器控制寄存器(地址:68H)

复位值	bit7						bit0	
0000 0000	CCTEN	CSEL1	CSEL0	CRSET	CFSET	CCT7	CCT3	CCT1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- CCTEN:** 比较器清零定时器功能使能位  
 0 = 禁止比较器清零定时器功能  
 1 = 使能比较器清零定时器功能
- CSEL<1:0>:** 比较器选择位  
 00 = 选择比较器 1 的输出 C2OUT  
 01 = 选择比较器 2 的输出 C2OUT  
 10 = 选择比较器 3 的输出 C2OUT  
 11 = 选择比较器 4 的输出 C2OUT
- CRSET:** 比较器输出上升沿清零定时器配置位  
 0 = 禁止比较器输出上升沿清零定时器  
 1 = 使能比较器输出上升沿清零定时器
- CFSET:** 比较器输出下降沿清零定时器配置位  
 0 = 禁止比较器输出下降沿清零定时器  
 1 = 使能比较器输出下降沿清零定时器
- CCT7:** 比较器清零 TMR7 计数寄存器使能位  
 0 = 禁止比较器清零 TMR7 计数寄存器 T7H/T7L  
 1 = 使能比较器清零 TMR7 计数寄存器 T7H/T7L
- CCT3:** 比较器清零 TMR3 计数寄存器使能位  
 0 = 禁止比较器清零 TMR3 计数寄存器 T3H/T3L  
 1 = 使能比较器清零 TMR3 计数寄存器 T3H/T3L
- CCT1:** 比较器清零 TMR1 计数寄存器使能位  
 0 = 禁止比较器清零 TMR1 计数寄存器 T1H/T1L  
 1 = 使能比较器清零 TMR1 计数寄存器 T1H/T1L

### 14.11.2 使用方法

1. 配置 CRSET 位和 CFSET 位选择比较器输出的上升沿、下降沿作为定时器的清零点；CRSET 和 CFSET 位可同时置 1，即比较器输出的上升沿和下降沿均可清零定时计数器；
2. 配置 CSEL<1:0>位选择一个比较器，配置好该比较器并使能开始工作；
3. 配置 CCT1/CCT3/CCT7 位，选择需要被比较器清零的定时器；允许同时使能，即 CCT1、CCT3、CCT7 可同时配置为 1，当比较器翻转时，T1H/T1L、T3H/T3L、T7H/T7L 均被清零；
4. 将 CCTEN 位置 1，使能比较器清零定时器功能。

注：比较器的输出和比较器输出极性选择位相关，请参见 CxFILTCTL 寄存器的 INV<sub>x</sub> 位。

## 15 通用全双工/ 半双工收发器

### 15.1 系统概述

KF8A100 包含 3 个通用全双工/半双工收发器 USART2、UART3 和 UART4。3 个 UART 模块的工作原理完全相同，区别在于 UART2 兼容 7816。本章将以 USARTx (x=2、3、4) 模块来讲述。

USART 是 Universal Synchronous /Asynchronous Receive & Transmit 的缩写，它的中文名称是通用同步/异步收发器，又称通用全双工/半双工收发器。本文称作通用全双工/半双工收发器。这是一个串口通信的 I/O 外设，也可作为串行通信接口。它可被配置为与个人计算机等外设通信的全双工异步系统。也可以被配置为与外设或其它单片机通信的半双工同步系统，与之通信的单片机通常不具有产生波特率的内部时钟，它需要主控同步器件提供外部时钟信号。

USART 模块可以实现如下功能：自动波特率检测和校准、接收每个字符时唤醒和发送 13 位间隔字符，从而保证该模块可以用于局域互连网络（Local Interconnect Network,LIN）总线系统。

#### 15.1.1 相关寄存器

表 15.1 USART 相关寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
168H	RSC2L2	SPEN2	R2X9	SRXEN2	CRXEN2	ADREN2	FRER2	OVFER2	R2X9D
20CH	RSC3L3	SPEN3	R3X9	SRXEN3	CRXEN3	ADREN3	FRER3	OVFER3	R3X9D
430H	RSC4L4	SPEN4	R4X9	SRXEN4	CRXEN4	ADREN4	FRER4	OVFER4	R4X9D
169H	TXSDR2	USART2 发送数据寄存器							
20DH	TXSDR3	USART3 发送数据寄存器							
431H	TXSDR4	USART4 发送数据寄存器							
16AH	RXSDR2	USART2 接收数据寄存器							
20EH	RXSDR3	USART3 接收数据寄存器							
432H	RXSDR4	USART4 接收数据寄存器							
16BH	BRCTL2	ABRDOVF2	RCIDLF2	SCKPS2	B2RG16	BR2CKS1	BR2CKS0	WUEN2	ABRDEN2
20FH	BRCTL3	ABRDOVF3	RCIDLF3	SCKPS3	B3RG16	BR3CKS1	BR3CKS0	WUEN3	ABRDEN3
433H	BRCTL4	ABRDOVF4	RCIDLF4	SCKPS4	B4RG16	BR4CKS1	BR4CKS0	WUEN4	ABRDEN4
16CH	TSCTL2	CSRS2	T2X9	TXEN2	SYNC2	SENDB2	HBRG2	TXSRS2	T2X9D
216H	TSCTL3	CSRS3	T3X9	TXEN3	SYNC3	SENDB3	HBRG3	TXSRS3	T3X9D
434H	TSCTL4	CSRS4	T4X9	TXEN4	SYNC4	SENDB4	HBRG4	TXSRS4	T4X9D
16DH	EUBRGL2	USART2 波特率数据寄存器低字节							
218H	EUBRGL3	USART3 波特率数据寄存器低字节							
435H	EUBRGL4	USART4 波特率数据寄存器低字节							
159H	EUBRGH2	USART2 波特率数据寄存器高字节							
219H	EUBRGH3	USART3 波特率数据寄存器高字节							

436H	EUBRGH4	USART4 波特率数据寄存器高字节							
163H	U7816TXC TL2	U7816EN2	CLKOUT2	TX9DSEL2	STOP2	TPAR2	TINV2	TCONV2	BGTEN2
167H	U7816RXC TL2	ERSW21	ERSW20	RPAR2	RINV2	RCONV2	-	-	PAREF2
166H	U7816CTL2	PSEL2	-	TREPEN2	TREP21	TREP20	RREPEN2	RREP21	RREP20
165H	CLKDIV2	CLKDIV27	CLKDIV26	CLKDIV25	CLKDIV 24	CLKDIV23	CLKDIV22	CLKDIV 21	CLKDIV2 0
164H	EGTCTL2	EGT27	EGT26	EGT25	EGT24	EGT23	EGT22	EGT21	EGT20
123H	UPINSET2	USLM2	-	-	-	-	-	-	-
21EH	UPINSET3	USLM3	-	-	-	-	-	-	-
43AH	UPINSET4	USLM4	-	-	-	-	-	-	-
160H	UADMATC H2	ADM27	ADM26	ADM25	ADM24	ADM23	ADM22	ADM21	ADM20
21CH	UADMATC H3	ADM37	ADM36	ADM35	ADM34	ADM33	ADM32	ADM31	ADM30
438H	UADMATC H4	ADM47	ADM46	ADM45	ADM44	ADM43	ADM42	ADM41	ADM40

## 15.1.2 原理框图

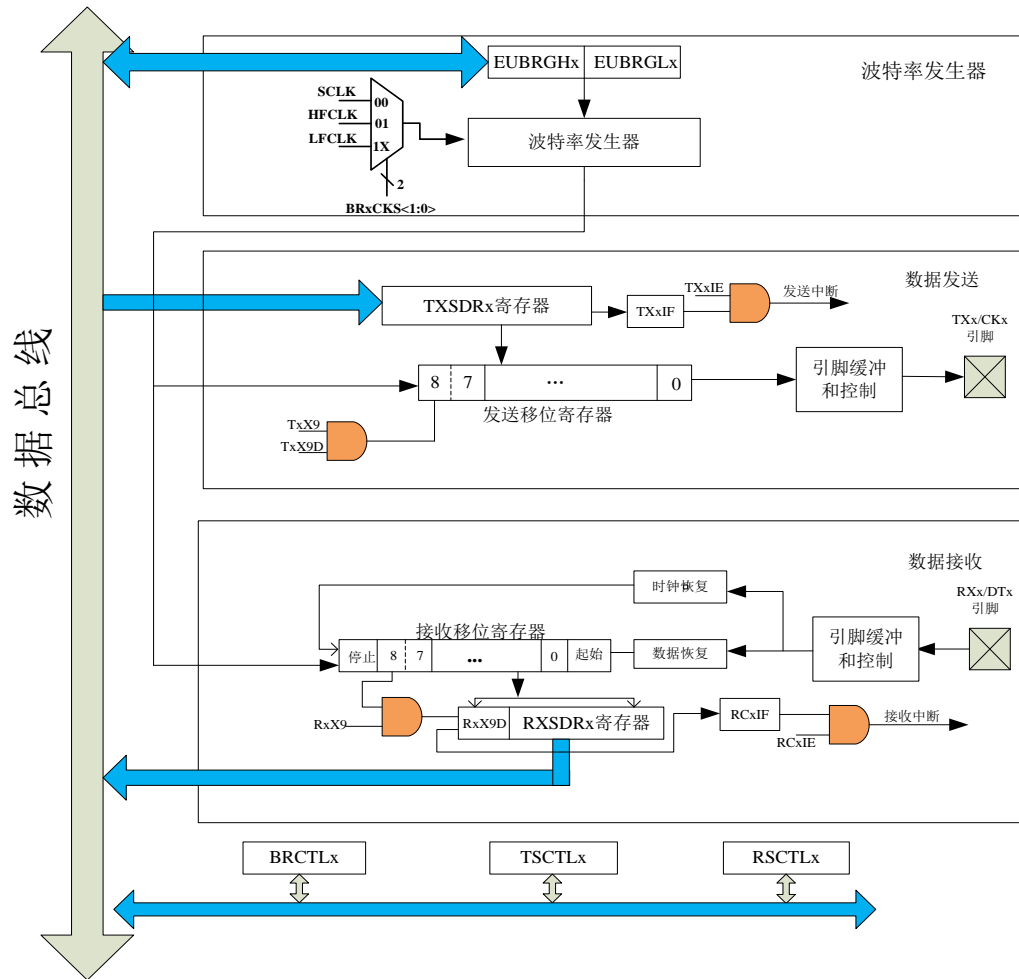


图 15.1 USARTx 模块框图

注：“x”=2/3/4，除特殊说明外，在本章中 x 均代表 2/3/4，用于区别串口的序号。

从图 15.1 中可以看出，通用全双工/半双工收发器（USARTx）模块包括波特率发生器（BRG）、数据发送和数据接收这三部分，每个部分都有相应的寄存器设置，主要包括波特率控制寄存器 BRCTLx，发送状态和控制寄存器 TSCTLx 和接收状态和控制寄存器 RSCTLx。

在波特率发生器这部分，通过设置 SYNCx、HBRGx 和 BxRG16 位来进行选择设置 {EUBRGHx: EUBRGLx} 寄存器的值来配置波特率。详见“波特率发生器”。

在数据发送部分，包括全双工异步发送和半双工同步发送，通过发送状态和控制寄存器 TSCTLx 来控制实现相应的功能，详见“USART 全双工模式及 USART 半双工模式”。

在数据接收部分，包括全双工异步接收和半双工同步接收，通过接收状态和控制寄存器 RSCTLx 来控制实现相应的功能，详见“USART 全双工模式及 USART 半双工模式”。

USARTx 有如下功能特点：全双工异步发送和接收、RS485 检测、双字节输入缓冲器、单字节输出缓冲器、可将字符长度编程为 8 位或 9 位、输入缓冲溢出错误检测、接收到字符的帧错误检测、半双工同步主控/从动模式和半双工同步模式下可编程时钟极性。

USARTx 模块还可实现如下附加功能，从而使其成为局域互网络总线系统的理想选择：自动波特率检测、校准和 13 位间隔字符发送。

## 15.2 波特率发生器

波特率发生器 (BRG) 是 8 位或者 16 位定时器，专用于支持全双工和半双工 USART 操作，默认情况下，BRG 工作在 8 位模式。BRxCKSt<1:0>位决定波特率发生器选用哪一种工作时钟，其中包括 SCLK、HFCLK 和 LFCLK 三种时钟信号。将 BRCTLx 寄存器的 BxRG16 位置 1 可选择 16 位模式。

{EUBRGHx: EUBRGLx} 寄存器决定运行波特率定时器的周期。在全双工模式下，波特率周期的乘数由 TSCTLx 寄存器的 HBRGx 位和 BRCTLx 寄存器的 BxRG16 位决定。在半双工同步模式下，HBRGx 位被忽略。

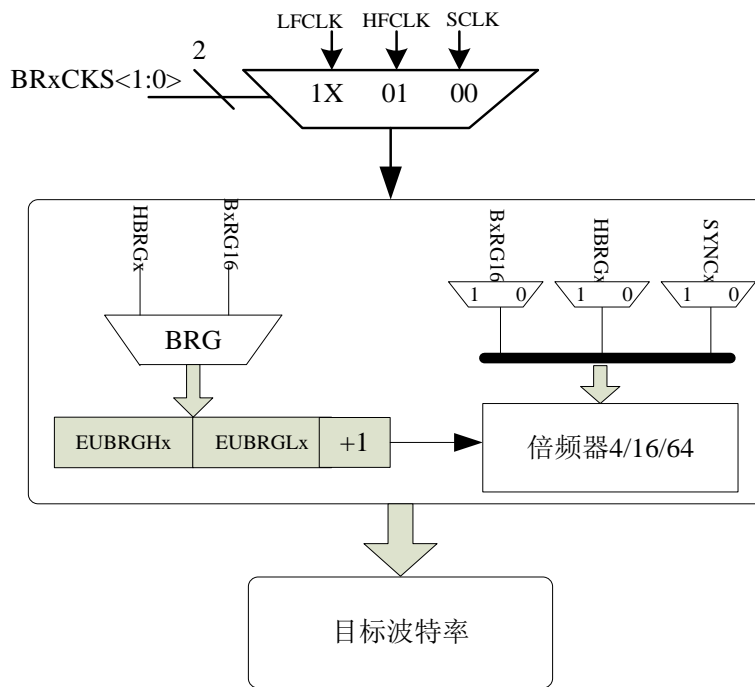


图 15.2 USARTx 目标波特率产生原理框图

### 15.2.1 相关寄存器

#### 15.2.1.1 USARTx 波特率控制寄存器 BRCTLx

寄存器: BRCTLx: 波特率控制寄存器x

复位值	bit7						bit0	
0100 0000	ABRDOV Fx	RCIDLx	SCKPSx	BxRG16	BRxCKSt	BRxCKSt	WUENx	ABRDEN x
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ABRDOVx: 自动波特率检测溢出位

	全双工异步模式: 1 = 自动波特率定时器溢出 0 = 自动波特率定时器没有溢出
	半双工同步模式: 无关位
RCIDLFx:	接收空闲标志位 全双工异步模式: 1 = 接收器空闲 (未接收) 0 = 已接收到起始位且接收器正在接收 半双工同步模式: 无关位
SCKPSx:	半双工同步时钟极性选择位 全双工异步模式: (SYNCx=0) 1 = 传送反相数据到 TXx/CKx 引脚 0 = 传送非反相数据到 TXx/CKx 引脚 半双工同步模式: 1 = 数据在时钟上升沿同步, 时钟空闲状态设置为高电平 0 = 数据在时钟下降沿同步, 空闲时, 时钟空闲状态设置为低电平
BxRG16:	16 位波特率发生器位 1 = 使用 16 位波特率发生器 0 = 使用 8 位波特率发生器
BRxCKS<1:0>:	USARTx 波特率发生器时钟选择位 00 = 选择 SCLK 主时钟作为 USARTx 波特率发生器时钟 01 = 选择 HFCLK 时钟作为 USARTx 波特率发生器时钟 1X = 选择 LFCLK 时钟作为 USARTx 波特率发生器时钟
WUENx:	唤醒使能位 全双工异步模式: 1 = 接收器正在等待下降沿。不会接收字符。RCxIF 将被置 1。RCxIF 置 1 后 WUENx 将被自动清 0。 0 = 接收器正常工作 半双工同步模式: 无关位
ABRDENx:	自动波特率检测使能位 全双工异步模式: 1 = 使能自动波特率模式(完成自动波特率后清 0) 0 = 禁止自动波特率模式 半双工同步模式: 无关位

注: (1) “x”=2/3/4, 除特殊说明外, 在本章中 x 均代表 2/3/4, 用于区别串口的序号。

(2) 这里将 3 个寄存器统一编写, 3 个寄存器分别为 BRCTL2、BRCTL3 和 BRCTL4, 下同。

## 15.2.2 波特率的选择

波特率的选择与状态位 SYNCx、BxRG16 和 HBRGx 有关，还与工作定时器 8 位或 16 位、半双工同步或全双工都有关系。

波特率发生器有三种时钟选择，分别是系统主时钟 SCLK、高频时钟 HFCLK 和低频时钟 LFCLK，通过比特率寄存器 BRCTLx 的 BRxCK<1:0>选用。

波特率的计算公式为：

$$\text{目标波特率} = \frac{\text{工作时钟频率}}{m \times ([\text{EUBRGHx} : \text{EUBRGLx}] + 1)}$$

其中 m 为倍频器选择，分别为 4、16 和 64，它与状态位 SYNCx、BxRG16 和 HBRGx 有关，并且与工作的定时器 8 位还是 16 位，半双工或全双工有一定的关系。表 15.2 列出了 m 选择表。

表 15.2 倍频器 m 选择表

配置位			BRG/USARTx 模式	倍频器 m
SYNCx	BxRG16	HBRGx		
0	0	0	8 位/异步	64
0	0	1	8 位/异步	16
0	1	0	16 位/异步	
0	1	1	16 位/异步	4
1	0	x	8 位/同步	
1	1	x	16 位/同步	

注：1. x 为无关位；

2. 使用内部时钟作为串口的时钟源时，推荐使用 m=16 或 m=64 的配置。内部晶振温漂比较大，若使用 m=4 的配置，则波特率误差会更大。

下例是确定波特率和波特率误差的计算示例。

**例：计算波特率误差**

器件工作在时钟频率 = 16MHz，目标波特率 = 9600，全双工异步模式，8 位 BRG:

$$\text{目标波特率} = \frac{\text{工作时钟频率}}{m \times ([\text{EUBRGHx} : \text{EUBRGLx}] + 1)}, \text{ 求解 } \text{EUBRGHx}:\text{EUBRGLx}.$$

$$X = \frac{\text{Fosc}}{\text{目标波特率}} - 1$$

$$= \frac{16000000}{9600} - 1$$

$$= 25.042$$

$$= 25 \text{ 十进制数}$$

$$= 19\text{H} \text{ 十六进制数}$$

$$\text{计算波特率} = \frac{16000000}{64(25+1)} = 9615$$



$$\begin{aligned} \text{误差} &= \frac{\text{计算波特率} - \text{目标波特率}}{\text{目标波特率}} \\ &= \frac{(9615 - 9600)}{9600} \\ &= 0.16\% \end{aligned}$$

各种全双工异步模式的典型波特率和误差值已经计算出来，请参考附录3。使用高波特率（HBRG<sub>x</sub> = 1）或 16 位 BRG（BxRG16 = 1）有助于降低波特率误差。16 位 BRG 模式用于在高速振荡器频率下取得较缓慢的波特率。

将新值写入 {EUBRGH<sub>x</sub>:EUBRGL<sub>x</sub>} 寄存器将对导致 BRG 定时器复位（或清 0）。这可以确保 BRG 无需等待定时器溢出就可以输出新的波特率。

如果在有效接收操作过程中更改工作时钟，可能会导致接收错误或数据丢失。为避免此问题，应检查 RCIDLF<sub>x</sub> 位的状态，以确保在更改工作时钟前接收操作处于空闲状态。

### 15.2.3 自动波特率检测

USART<sub>x</sub> 模块支持自动波特率检测和校准。

在自动波特率检测（Auto-Baud Rate Detect, ABRD）模式下，BRG 不为 RX<sub>x</sub> 输入提供时钟信号，而是由 RX<sub>x</sub> 为 BRG 定时。波特率发生器用于为接收的 0X55（“U”的 ASCII 码）定时，0X55 是 LIN 总线的同步字符。此字符的特殊之处在于它具有包括停止位边沿在内的 5 个上升沿。

将 BRCTL<sub>x</sub> 寄存器的 ABRDEN<sub>x</sub> 位置 1 将启动自动波特率校验序列（图 15.3）。当发生 ABRD 序列时，USART<sub>x</sub> 状态机保持在空闲状态。在接收线的第一个上升沿（起始位之后），EUBRGL<sub>x</sub> 使用 BRG 计数器时钟递增计数。在第 8 位周期的末尾将在 RX<sub>x</sub> 引脚上出现第 5 个上升沿。此时，对正确的 BRG 周期的累计值被留在 EUBRGH<sub>x</sub> 和 EUBRGL<sub>x</sub> 寄存器中，ABRDEN<sub>x</sub> 位被自动清 0 而 RCxIF 中断标志被置 1。要清除 RCxIF 中断，需要读取 RXSDRx 中的值。RXSDRx 的内容应该被丢弃。校准不使用 EUBRGH<sub>x</sub> 寄存器的模式时，用户可通过查询 EUBRGH<sub>x</sub> 寄存器中的 0X00 验证 EUBRGL<sub>x</sub> 寄存器是否未溢出。

BRG 自动波特率时钟由 BxRG16 和 HBRG<sub>x</sub> 位决定，如表 15.3 所示。在 ABRD 期间，EUBRGH<sub>x</sub> 和 EUBRGL<sub>x</sub> 寄存器共同用作 16 位计数器，这与 BRG16 位的设置无关。在校准波特率周期时，EUBRGH<sub>x</sub> 和 EUBRGL<sub>x</sub> 寄存器的定时频率为 BRG 基时钟频率的 1/8。得到的字节测量结果为全速下的平均位时间。

- 注：1、如果 WUEN<sub>x</sub> 位和 ABRDEN<sub>x</sub> 位都置 1，自动波特率检测将发生在间隔字符之后的字节处。  
 2、需要由用户来判断输入字符的波特率是否处于所选 BRG<sub>x</sub> 时钟源范围内。某些振荡器频率和 USART 波特率组合不可能实现。  
 3、在自动波特率过程中，自动波特率计数器从 1 开始计数。自动波特率序列完成后，为了得到最准确的结果，应从 EUBRGH<sub>x</sub>:EUBRGL<sub>x</sub> 寄存器对的值中减去 1。  
 4、启动自动波特率检测时，存在某一次自动波特率检测结果不正确的情况，不建议每一帧的数据都进行自动波特率检测。

表 15.3 BRG 计数器时钟速率

BxRG16	HBRG <sub>x</sub>	BRG 基时钟	BRG ABRD 时钟
0	0	工作时钟/64	工作时钟/512
0	1	工作时钟/16	工作时钟/128
1	0	工作时钟/16	工作时钟/128

1	1	工作时钟/4	工作时钟/32
---	---	--------	---------

注:在ABRD序列期间, EUBRGLx和EUBRGHx寄存器都用作16位计数器,与BRG16x的设置无关。

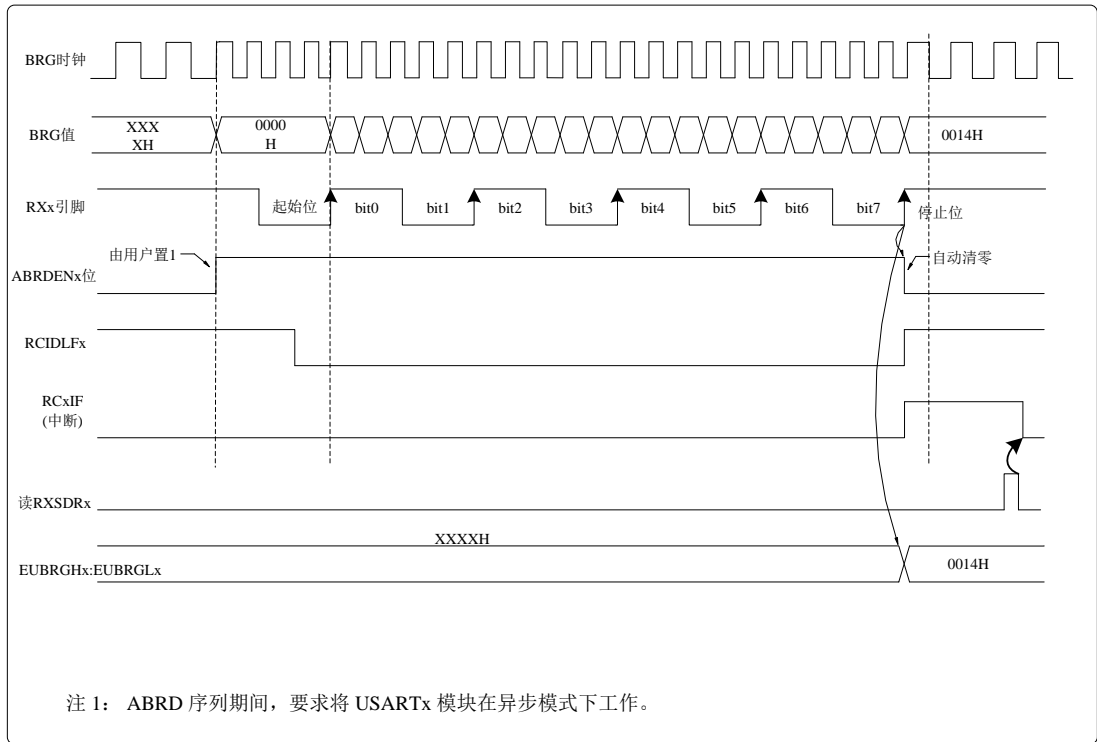


图 15.3 自动波特率检测

### 15.2.4 接收间隔字符时自动唤醒

在休眠模式下, USART 的所有时钟都会停止, 此时波特率发生器处于无效状态, 无法进行正确的字符接收。自动唤醒功能允许在 RXx/DTx 线上活动时唤醒控制器。只有 USARTx 工作在异步模式下时才可以使用该功能。

通过将 BRCTLx 寄存器的 WUENx 位置 1, 使能自动唤醒功能, 置 1 后, 将禁止 RXx/DTx 上的正常序列的接收, 并且 USARTx 保持在空闲状态, 监视唤醒事件, 监视行为与 CPU 的工作模式无关。唤醒事件是指 RXx/DTx 线上从高电平到低电平的转换, 这与同步间隔字符或与 LIN 协议唤醒信号字符的启动条件一致。

在检测到唤醒事件时, 将会产生一个 RCxIF 中断。在 CPU 正常工作的模式下, 中断会与 Q 时钟同步产生; 如果器件处于休眠模式, 则两者不同步。可通过读 RXSDRx 寄存器清除中断条件。

当 RXx 线从低电平向高电平转换时, WUENx 会被清零, 这就意味着“同步间隔”事件结束。此时, USARTx 会处于空闲模式, 等待接收下一字符。

为避免错误, 需要注意以下事项。

(1) 间隔字符

为了避免唤醒事件期间的字符错误或字符分段, 唤醒字符必须为全零。

如果使能唤醒功能, 无论数据的低电平持续时间的长短, 该功能都可以起作用。如果将 WUE 位置 1, 且收到一个有效的非零字符, 则从起始位到第一个上升沿之间的低电平时被解释为唤醒事件, 而后续的字符将会当被接收, 导致帧错误或溢出错误。

因此，使用唤醒功能时，发送的初始字符必须全为 0，且持续时间必须为 10 个或更多数据位的发送时间。针对 LIN 总线建议为 13 个位时间，而对于标准的 RS-232 器件可为任意数量的位时间。

(2) 振荡器的起振时间

在使用自动唤醒时还需要考虑振荡器的起振时间。“同步间隔”（或唤醒信号）字符必须足够长，并且跟由足够长的时间间隔，使选定的振荡器有足够的时间起振并使 USART 正确初始化。

(3) WUEN<sub>x</sub> 位

唤醒事件会产生接收中断，将 RC<sub>x</sub>IF 位置 1。在 RX<sub>x</sub>/DT<sub>x</sub> 的上升沿，由硬件清零 WUEN<sub>x</sub> 位。通过在软件中读 RXSDR<sub>x</sub> 寄存器清除中断条件。

要保证没有丢失数据，在将 WUEN<sub>x</sub> 位置 1 前，应检查 RCIDL<sub>Fx</sub> 位，以验证是否正在进行接收。如果不是在进行接收，则可将 WUEN<sub>x</sub> 位置 1，时器件立即进入休眠模式。(没事干，睡觉)

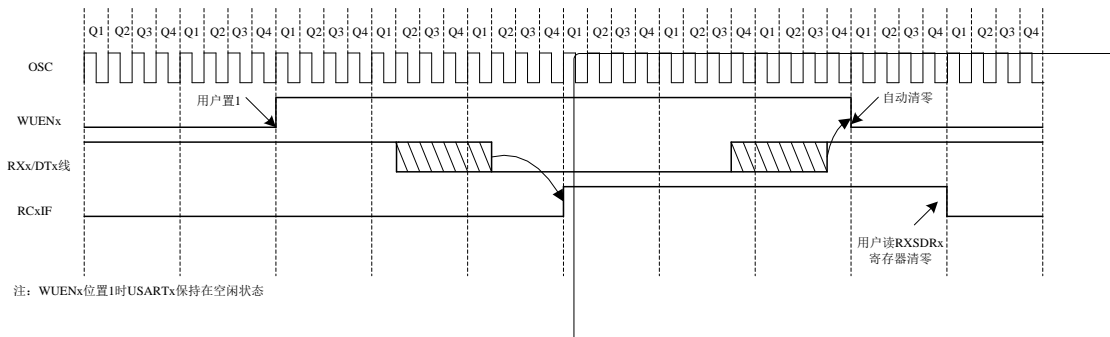


图 15.4 正常工作下的自动唤醒时序

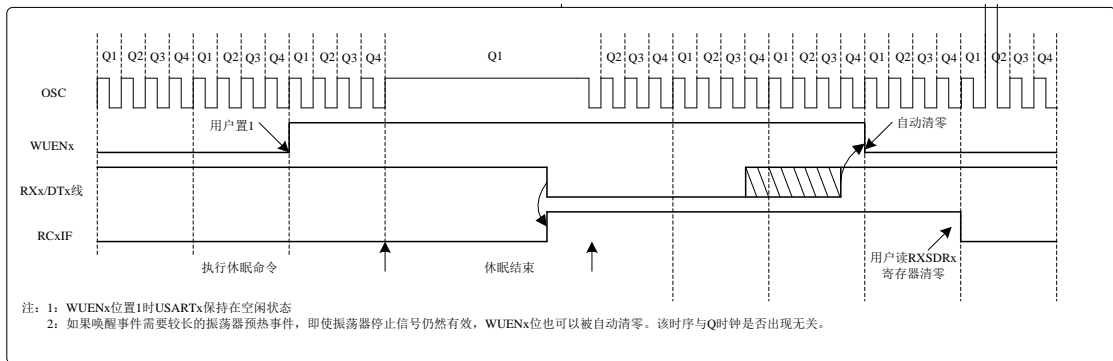


图 15.5 休眠模式下的自动唤醒时序

15.2.5 间隔符时序

USART<sub>x</sub> 模块能够发送符合 LIN 总线标准的特殊间隔字符序列。发送的间隔字符包括 1 个起始位，后面跟着 12 个 0 位和一个停止位。

要发送间隔字符，要先将 TXEN<sub>x</sub> 值 1，然后再将 SENDB<sub>x</sub> 位置 1（SENDB<sub>x</sub> 位需要在 TXEN<sub>x</sub> 位为 1 的时候才能写入），之后对 TXSDR<sub>x</sub> 寄存器的写操作将会启动间隔字符的发送。此时写入 TXSDR<sub>x</sub> 的数据会被忽略，并会发送全 0。

在发送了相应的停止位之后，硬件会自动将 SENDB<sub>x</sub> 位清零。这样用户可以在发送完间隔字符（在 LIN 规范中通常时同步字符）后将下一个要发送的字节预先装入发送 FIFO 中。TXCTL<sub>x</sub> 寄存器中的 TXSR<sub>Sx</sub> 位则表明发送正在进行还是处于空闲状态。

### 15.2.5.1 间隔和同步发送序列

以下序列会发送一个报文帧头，包含一个间隔字符和其后的自动波特率同步字节。此序列适用于典型的 LIN 总线器件。

- 1、将 USARTx 配置为所需的模式
  - 2、将 TXENx 位置 1
  - 3、将 SENDBx 位置 1，设置间隔字符（SENDERx 需要在 TXENx 为 1 的时候才能写入）
  - 4、将无效字符装入 TXSDRx 寄存器，启动发送（该值会被忽略）
  - 5、将 55H 写入 TXSDRx 中，以便把同步字符装入发送 FIFO 缓冲中
  - 6、间隔字符发送后，硬件会将 SENDBx 位复位。开始发送同步字符
- 当 TXxIF 指示 TXSDRx 为空时，下一个数据会写入 TXSDRx 中。

### 15.2.5.2 接收间隔字符

USARTx 接收间隔字符有两种方法。

第一种检测间隔字符的方法是使用 RXCTLx 寄存器的 FERRx 位和 RXSDRx 指示接收的数据。假设已将波特率发生器初始化成预期的波特率。当

- RCxIF 位置 1
- FERRx 位置 1
- RXSDRx 为 00H 时

表明接收到了间隔字符。

第二种方法是使用自动唤醒功能。通过使能此功能，USARTx 将采样 RXx/DTx 上电平的下两个跳变，产生一个 RCxIF 中断，接收下一个数据字节，之后产生另一个中断。

注意，在间隔字符后，用户通常希望使能自动波特率检测功能。无论使用哪种方法，用户都可以在 USARTx 进入休眠模式之前，将 BRCTLx 寄存器的 ABRDENx 位置 1。

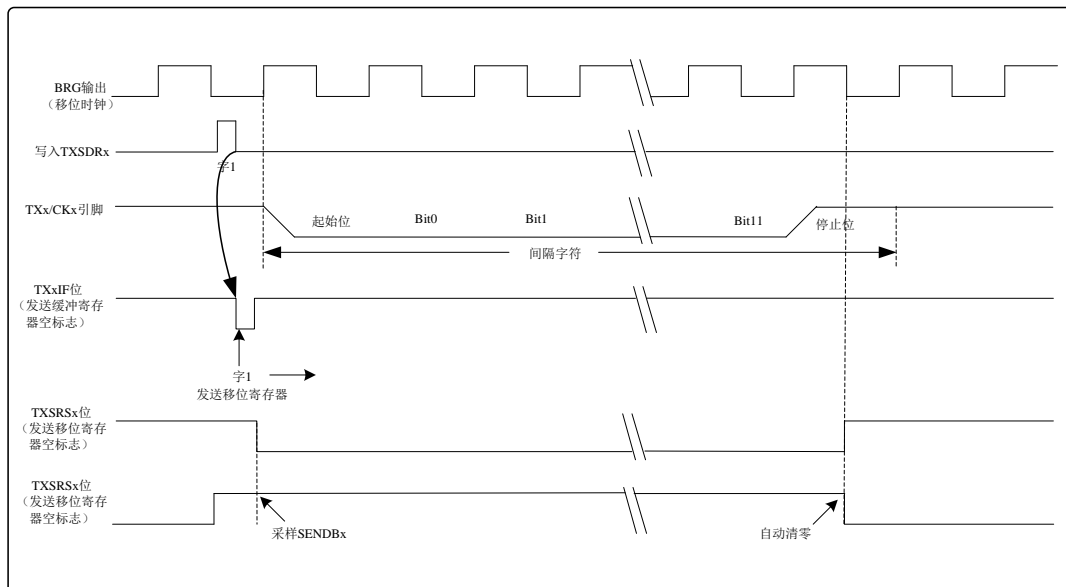


图 15.6 发送间隔字符时序

## 15.3 USART 全双工模式

在全双工异步通信中，数据是一帧一帧传送的，每一帧的数据格式如图 15.7 所示。

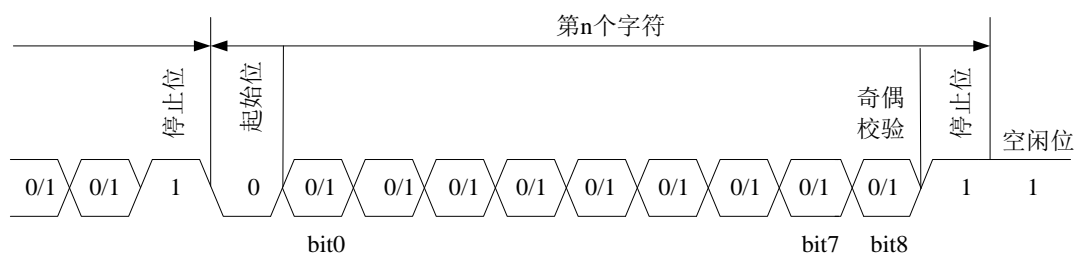


图 15.7 全双工异步通信数据结构图

在帧格式中，一个字符由 4 部分组成：起始位，数据位，奇偶校验位和停止位。

- 起始位：通常情况下是逻辑 0，占用一位，用来通知接收设备一个等待接收字符的开始。
- 数据位：8 位。
- 奇偶校验位：bit8，占用一位，但在字符中可以规定不用奇偶校验位，则这一位可以省去。
- 停止位：一定为逻辑 1，用来表征字符的结束。停止位可以是 1 位、1.5 位或 2 位。接收端收到停止位后，知道上一字符已经传送完毕，同时，也为接收下一字符作好准备，只要在接收到 0，就是新字符的起始位。若停止位以后不再紧接着传送下一个字符，则使线路电平保持为高电平（逻辑 1），处于空闲状态。这也是全双工异步通信的一大特点。

最常用的数据格式为 8 位。每个发送位的持续时间为  $1/(\text{波特率})$ 。片上专用 8 位/16 位波特率发生器可用于三种时钟信号和四种时钟源。请参见表 15.3 了解波特率配置示例。

USARTx 首先发送和接收低位。USARTx 的发送器和接收器在功能上是相互独立的，但采用相同的数据格式和波特率。硬件不支持奇偶校验，但可以用软件实现（奇偶校验位是第 9 个数据位）。

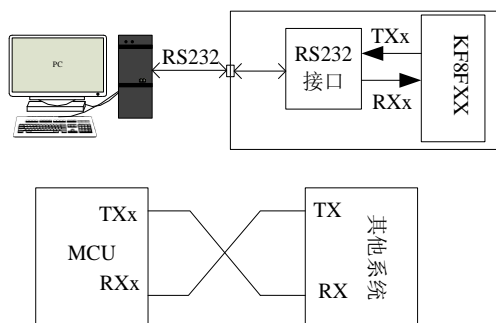


图 15.8 串口通信原理框图

### 15.3.1.1 USARTx 全双工发送操作

USARTx 全双工异步发送操作通过 USARTx 发送器完成。发送器的核心是串行发送移位寄存器（发送移位），该寄存器不能由软件直接访问。发送移位寄存器从 TXSDRx 发送缓冲寄存器获取数据。

通过配置如下三个控制位使能 USARTx 发送器，以用于全双工异步操作：

- 1) TXENx = 1
- 2) SYNCx = 0
- 3) SPENx = 1

假设所有其它 USARTx 控制位处于其默认状态。将 TSCTLx 寄存器的 TXENx 位置 1，使能 USARTx 发送器电路。将 TSCTLx 寄存器的 SYNCx 位清 0，将 USARTx 配置用于全双工异步操作。将 RSCTLx 寄存器的 SPENx 位置 1，使能 USARTx 并自动将 TXx/CKx 的 I/O 引脚配置为输出引脚。如果与模拟外设共用 TXx/CKx 引脚，必须清 0 相应的 ANSEL 位禁止模拟 I/O 功能。

- 注：1. 将 SPENx 位置 1 会自动将 RXx/DTx I/O 引脚配置为输入引脚时，无论相关的 TR 位的状态如何以及 USARTx 接收器使能与否。可以通过普通端口读 RXx/DTx 引脚数据，但却无法使用该端口锁存输出数据。
2. 如果 TXENx 使能位置 1，TXxIF 发送器中断标志位会置 1。

### 15.3.1.2 发送状态和控制寄存器 TSCTLx

**寄存器：** TSCTLx: 发送状态和控制寄存器

复位值	bit7						bit0	
0000 0010	CSRSx	TxX9	TXENx	SYNCx	SENDBx	HBRGx	TXSRSx	TxX9D
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CSRSx: 时钟源选择位

全双工异步模式:

无关位

半双工同步模式:

1 = 主模式(由 BRG 内部产生时钟)

0 = 从模式(时钟源来自外部)

TxX9: RS485 发送使能位

1 = 选择 RS485 发送

0 = 选择 8 位发送

TXENx: 发送使能位(x)

1 = 使能发送

0 = 禁止发送

SYNCx: 串行通信模式选择位

1 = 半双工同步模式

0 = 全双工异步发送模式

SENDBx: 发送间隔字符位

全双工异步模式:

1 = 在下次发送时发送同步间隔字符(完成后由硬件清 0)

0 = 同步间隔字符发送完成

	半双工同步模式:
	无关位
HBRGx:	高波特率选择位
	全双工异步模式:
	1 = 高速
	0 = 低速
	半双工同步模式:
	在此模式下不使用
TXSRx:	发送移位寄存器状态位
	1 = 发送移位空(发完) (满)
	0 = 发送移位满 (空)
Txx9D:	发送数据的第 9 位
	可以是地址/数据位或奇偶校验位。

### 15.3.1.3 发送数据

向 TXSDRx 寄存器写入一个字符，以启动发送。如果这是第一个字符，或者前一个字符已经完全从发送移位寄存器中移出，TXSDRx 中的数据会立即发送给发送移位寄存器。如果发送移位寄存器中仍保存全部或部分前一字符，新的字符数据将保存在 TXSDRx 中，直到发送完前一字符的停止位为止。然后，在停止位发送完毕后经过一个发送周期，TXSDRx 中待处理的数据将被传输到发送移位寄存器。当数据从 TXSDRx 传输至发送移位寄存器后，立即开始进行起始位、数据位和停止位序列的发送。

只要使能 USARTx 发送器且 TXSDRx 中没有待发送数据，就将 EIF2 寄存器的 TXxIF 中断标志位置 1。换句话说，只有当发送移位寄存器忙于处理字符和 TXSDRx 中有排队等待发送的新字符时，TXxIF 位才处于清 0 状态。写 TXSDRx 时，不立即清 0 TXxIF 标志位。TXxIF 在写指令后的第 2 个指令周期清 0。在写 TXSDRx 后立即查询 TXxIF 会返回无效结果。TXxIF 为只读位，不能由软件置 1 或清 0。

可通过将 EIF2 寄存器的 TXxIE 中断允许位置 1 允许 TXxIF 中断。然而，只要 TXSDRx 为空，不管 TXxIE 允许位的状态如何都会将 TXxIF 标志位置 1。

如果要在发送数据时使用中断，只在有待发送数据时，才将 TXxIE 位置 1。当将待发送的最后一个字符写入 TXSDRx 后，将 TXxIE 中断允许位清 0。

TSCTLx 寄存器的 TXSRx 位指示发送移位寄存器的状态。TXSRx 位为只读位。当发送移位寄存器为空时，TXSRx 位被置 1，当有字符从 TXSDRx 传输到发送移位寄存器时，TXSRx 被清 0。TXSRx 位保持清 0 状态，直到所有位从发送移位寄存器移出为止。没有任何中断逻辑与该位有关，所以用户必须查询该位来确定发送移位寄存器的状态。

### 15.3.1.4 全双工发送的设置

1. 设置 BRCTLx 寄存器的 BTxCKs<1:0> 位选择波特率发生器的工作时钟，再根据相应的时钟信号选择相应的时钟源。
2. 初始化 EUBRGHx 和 EUBRGLx 这对寄存器以及 HBRGx 和 BxRG16 位，以获得所需的波特率（见章节“波特率发生器 (BRG)”）。
3. 通过将 SYNCx 位清 0 并将 SPENx 位置 1 使能全双工异步串口。

4. 如果需要9位发送，将TxX9控制位置1。当接收器被设置为进行地址检测时，将数据位的第9位置1，指示8个最低数据位为地址。
5. 将TXEN<sub>x</sub>控制位置1，使能发送；这将导致TXxIF中断标志位置1。
6. 如果需要中断，将TXxIE中断允许位置1；如果INTCTL寄存器的AIE和PUIE位也置1将立即产生中断。
7. 若选择发送9位数据，第9位应该被装入TxX9D数据位。
8. 将8位数据装入TXSDRx寄存器开始发送数据。

例: 发送数据，以USART1为例  
 SET TSCTL1, TXEN1  
 MOV R1, UART\_TEMP  
 MOV TXSDR1, R1  
 NOP  
 NOP  
 NOP  
 JB TSCTL1, TXSRS1; 判断是否发生完毕?  
 JMP \$-3

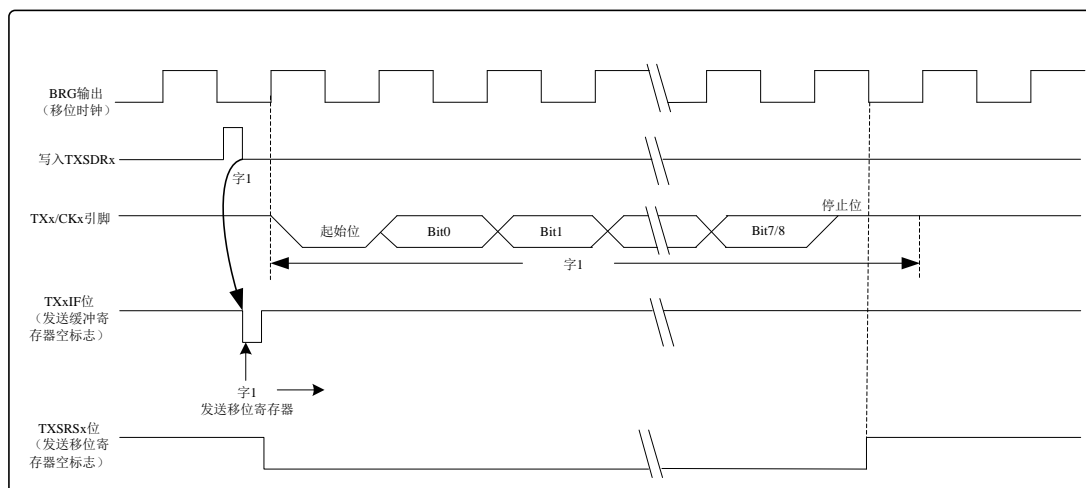


图 15.9 全双工异步发送（两字符间有空闲位）



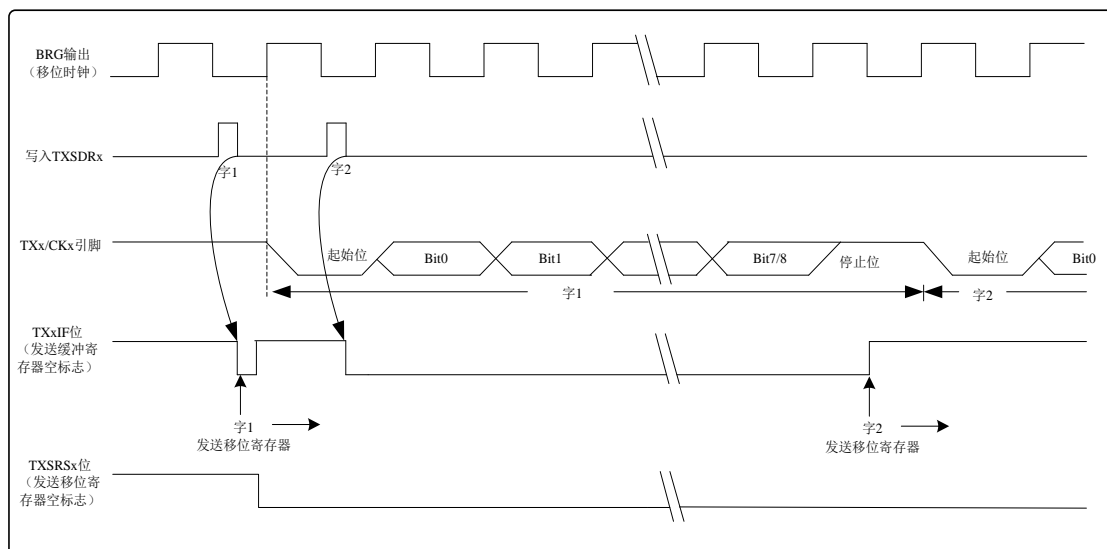


图 15.10 全双工异步发送（一字符接一字符）

### 15.3.2 USART 全双工接收操作

全双工异步接收模式通常用于 RS-232 系统。在  $RXx/DTx$  引脚上接收数据和驱动数据恢复电路。数据恢复电路实际上是一个以 16 倍波特率为工作频率的高速移位器，而串行接收移位寄存器（Receive Shift Register, RSR）则以波特率工作。当字符的全部 8 位或 9 位数据位被移入后，立即将它们传输到一个 2 字符的先入先出（FIFO 缓冲器）缓冲器。FIFO 缓冲器允许接收 2 个完整的字符和第 3 个字符的起始位，然后必须由软件将接收到的数据提供给 USARTx 接收器。FIFO 缓冲器和 RSR 寄存器不能直接由软件访问。通过  $RXSDRx$  寄存器访问接收到的数据。

通过配置如下三个控制位使能 USARTx 接收器，以用于全双工异步接收操作。

- ◆  $CRXEN_x = 1$
- ◆  $SYNC_x = 0$
- ◆  $SPEN_x = 1$

假设所有其它 USARTx 控制位都处于默认状态。将  $RSCTLx$  寄存器的  $CRXEN_x$  位置 1，使能 USARTx 接收器电路。将  $TSCTLx$  寄存器的  $SYNC_x$  位清 0，配置 USARTx 以用于全双工异步操作。将  $RSCTLx$  寄存器的  $SPEN_x$  位置 1，使能 USARTx 并自动将  $RXx/DTx$  引脚配置为输入引脚。如果  $RXx/DTx$  引脚与模拟外设共用，必须清 0 相应的  $ANSEL$  位禁止模拟 I/O 功能。

注：当将  $SPEN_x$  位置 1， $Tx/CKx$  I/O 引脚被自动配置为输出引脚时，无需考虑相应  $TR$  位的状态以及 USART 发送器使能与否。端口锁存器与输出驱动器是断开的，从而不能将  $Tx/CKx$  引脚用作通用输出引脚。

## 15.3.2.1 接收状态和控制寄存器 RSCTLx

寄存器: RSCTLx: 接收状态和控制寄存器x

复位值 0000 000x	bit7						bit0	
	SPENx	RxX9	SRXENx	CRXENx	ADRENx	FRERx	OVFERx	RxX9D
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SPENx: 串行口使能位

- 1 = 使能串行口(将 RXx/DTx 和 TXx/CKx 引脚配置为串行口引脚)
- 0 = 禁止串行口(保持为复位状态)

RxX9: 9 位接收使能位

- 1 = 选择 9 位接收 RS-485 全双工异步模式(RxX9 = 1)
- 0 = 选择 8 位接收

SRXENx: 单字节接收使能位

- 全双工异步模式:
- 无关位
- 半双工同步模式——主:
- 1 = 使能单字节接收
- 0 = 禁止单字节接收
- 此位在接收完成后清 0。
- 半双工同步模式——从:
- 无关位

CRXENx: 连续接收使能位

- 全双工异步模式:
- 1 = 使能接收器
- 0 = 禁止接收器
- 半双工同步模式:
- 1 = 使能连续接收, 直到使能位 CRXENx 清 0(CRXENx 改写 SRXENx)
- 0 = 禁止连续接收

ADRENx: 地址检测使能位

- RS-485 全双工异步模式(RxX9 = 1):
- 1 = 使能地址检测、允许中断, 当 RSR<8>置 1 时装入接收缓冲区
- 0 = 禁止地址检测, 接收所有字节并且第 9 位可作为奇偶校验位
- 8 位异步模式(RxX9=0):
- 无关位

FRERx: 帧错误位

- 1 = 帧错误(可通过读 RXSDRx 寄存器刷新该位, 并接收下一个有效字节)
- 0 = 无帧错误

OVFERx: 溢出错误位

- 1 = 溢出错误(可通过清 0 CRXENx 位来清 0 该位)
- 0 = 无溢出错误

RxX9D: 接收数据的第 9 位

- 该位可以是地址/数据位或奇偶校验位, 并且必须由用户固件计算得到。

### 15.3.2.2 接收数据

接收器数据恢复电路在第一个位的下降沿开始接收字符。第一个位，通常称为起始位，始终为 0。由数据恢复电路计数半个位时间，到起始位的中心位置，校验该位是否仍为零。如果该位不为零，数据恢复电路放弃接收该字符，而不会产生错误，并且继续查找起始位的下降沿。如果起始位零校验通过，则数据恢复电路计数一个完整的位时间，到达下一位的中心位置。由数据检测电路对该位进行采样，将相应的采样结果 0 或 1 移入 RSR。重复该过程，直到完成所有数据位的采样并将其全部移入 RSR 寄存器。测量最后一个位的时间并采样其电平。此位为停止位，总是为 1。如果数据恢复电路在停止位的位置采样到 0，则该字符的帧错误标志将置 1，反之，该字符的帧错误标志会清 0。详见章节“接收帧错误”获得有关帧错误描述。

当接收到所有数据位和停止位后，RSR 中的字符会被立即传输到 USARTx 的接收 FIFO 缓冲器并将 RCxIF 中断标志位置 1。通过读 RXSDRx 寄存器将 FIFO 缓冲器最顶端的字符移出 FIFO 缓冲器。

注：如果接收FIFO溢出，则不能再继续接收其他字符，直到溢出条件被清除。请章节“接收错误”获得有关溢出错误的更多相关信息。

只要使能 USARTx 接收器且在接收 FIFO 缓冲器中没有未读数据，中断标志寄存器中的 RCxIF 中断标志位就会置 1。RCxIF 中断标志位为只读，不能由软件置 1 或清 0。通过将下列所有位均置 1 来允许 RCxIF 中断：

- ◆ 中断使能寄存器的 RCxIE 中断允许位
- ◆ INTCTL 寄存器的 PUIE 外设中断允许位或低优先级中断允许位 AIEL
- ◆ INTCTL 寄存器的 AIE 全局中断允许位或高优先级中断允许位 AIEH

如果 FIFO 缓冲器中有未读数据，无论中断允许位的状态如何，都会将 RCxIF 中断标志位置 1。

### 15.3.2.3 接收错误

接收 FIFO 缓冲器中的每个字符都有一个相应的帧错误状态位。帧错误指示未在预期的时间内接收到停止位。由 RSCTLx 寄存器的 FRERx 位获取帧错误状态。FRERx 位代表接收 FIFO 缓冲器最顶端未读字符的状态。因此，必须在读 RXSDRx 寄存器之前读 FRERx 位。FRERx 位为只读位，且只能用于接收 FIFO 缓冲器的最顶端未读字符。帧错误 (FRERx = 1) 并不会阻止接收更多的字符。无需清 0 FRERx 位。从 FIFO 缓冲器读下一字符会使 FIFO 缓冲器指针前进至下一字符和下一个相应的帧错误。

清 0 RSCTLx 寄存器的 SPENx 位会复位 USARTx，并强制清 0 FRERx 位。清 0 RSCTLx 寄存器的 CRXENx 位不影响 FRERx 位。帧错误本身不会产生中断。

注：如果接收FIFO缓冲器中所有接收到的字符都有帧错误，重复读RCSDRx不会清零FRERx 位。

接收 FIFO 缓冲器可以保存 2 个字符。但如果在访问 FIFO 缓冲器之前，接收到完整的第 3 个字符，则会产生溢出错误。此时，RSCTLx 寄存器的 OVFERx 位会置 1。可以读取 FIFO 缓冲器内的字符，但是在错误清除之前，不能再接收其它字符。可以通过清 0 RSCTLx 寄存器的 CRXENx 位或通过清 0 RSCTLx 寄存器的 SPENx 位使 USARTx 复位来清除错误。

## 15.3.2.4 地址检测

当多个接收器共享同一传输线时（如在RS-485系统中），可使用特殊地址检测模式。将RSCTLx寄存器的ADRENx位置1，使能地址检测模式。地址检测要求接收9位字符。使能地址检测后，只有第9位数据位被置1的字符可以被传输到接收FIFO缓冲器。被传输到接收FIFO缓冲器的字符将与地址匹配设置寄存器中的值进行比较，当匹配时，RXIFx中断标志位置1。所有其它字符将被忽略。

**寄存器:** UADMATCHx: 地址匹配设置寄存器x

		bit7				bit0			
复位值		ADMx7	ADMx6	ADMx5	ADMx4	ADMx3	ADMx2	ADMx1	ADMx0
0000 0000		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ADMx<7:0>: UARTx地址匹配功能设置位

注: UADMATCH1寄存器 地址为260H; UADMATCH2寄存器 地址为160H; UADMATCH3寄存器 地址为21CH; UADMATCH4寄存器 地址为438H;

## 15.3.2.5 全双工接收的设置

1. 设置BRCTLx寄存器的BTxCKS<1:0>位选择波特率发生器的工作时钟，再根据相应的时钟信号选择相应的时钟源。
2. 初始化{EUBRGHx:EUBRGLx}这对寄存器以及HBRGx和BxRG16位，以获得所需的波特率（见章节“波特率发生器（BRG）”）。
3. 将SPENx位置1，使能串行端口。必须清0 SYNCx位以执行全双工异步操作。
4. 如果需要中断，将中断使能寄存器中的RCxIE位和INTCTL寄存器的AIE和PUIE位置1。如果使用中断优先级则将IPEN和PTX均置1
5. 如果需要接收9位数据，将RxX9位置1。
6. 将CRXENx位置1使能接收。
7. 当一个字符从RSR传输到接收缓冲器时，将RCxIF中断标志位置1。如果RCxIE中断允许位也置1还将产生中断。
8. 读RSCTLx寄存器获取错误标志位和第9位数据位（如果使能9位数据接收）。
9. 读RXSDRx寄存器，从接收缓冲器获取接收到的8个低数据位。
10. 如果发生溢出，通过清0 CRXENx接收器使能位清0 OVFERx标志。

例: 接收(USART1为例)

```

LOOP_USART
    NOP
    NOP
    SET    RSCTL1,CRXEN1        ;使能接收器
    JNB    RSCTL1,OVFER1        ;检测溢出错误
    CLR    RSCTL1,CRXEN1
    JB     EIF2,RXIF1           ;接收数据的标志
    JMP    LOOP_USART
    CLR    EIF2,RXIF1
    MOV    R0,RXSDR1
    MOV    UART_TEMP,R0        ;存储接收到的数据

```

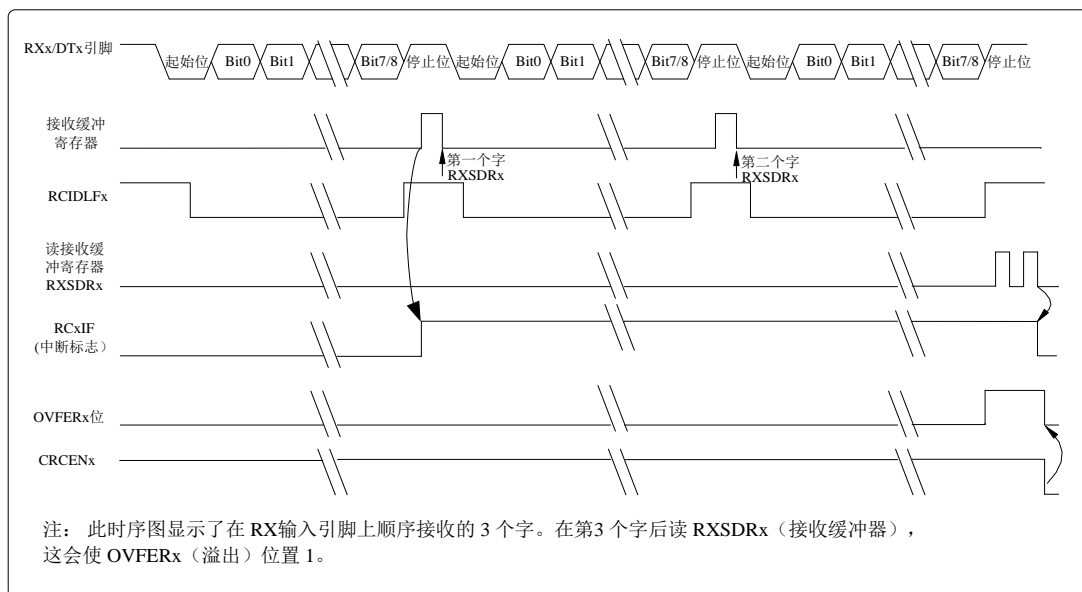


图 15.11 全双工异步接收

### 15.3.3 RS-485 发送/接收

USARTx 支持 RS-485 模式 9 位发送。当 TSCTLx 寄存器的 TxX9 位置 1 时，USARTx 将移出每个待发送字符的 9 位。TSCTLx 寄存器的 TxX9D 位为发送数据第 9 位，即最高数据位。当发送 9 位数据时，必须在将 8 个最低位写入 TXSDRx 之前，写 TxX9D 数据位。在写入 TXSDRx 寄存器后会立即将 9 个数据位传输到发送移位寄存器。

使用多个接收器时可使用特殊的 9 位地址模式。请参见章节“地址检测”获得有关地址模式的更多信息。

USARTx 支持 RS-485 模式 9 位接收。将 RSCTLx 寄存器的 RxX9 位置 1 时，USARTx 将接收到的每个字符的 9 位移入 RSR。RSCTLx 寄存器的 RxX9D 位是接收 FIFO 缓冲器顶端未读字符的第 9 位，同时也是最高数据位。当从接收 FIFO 缓冲器读取 9 位数据时，必须在读 RXSDRx 中的低 8 位之前，读取 RxX9D 数据位。

#### 15.3.3.1 RS-485 9 位地址检测模式设置

要设置使能地址检测的全双工异步接收：

1. 初始化 EUBRGHx 和 EUBRGLx 这对寄存器以及 HBRGx 和 BxRG16 位，以获得所需的波特率（见章节“波特率发生器（BRG）”）。
2. 将 SPENx 位置 1，使能串行端口。必须清 0 SYNCx 位以执行全双工异步操作。
3. 如果需要中断，将中断使能寄存器中的 RCxIE 位和 INTCTL 寄存器的 AIE 和 PUIE 位置 1。如果使用优先级中断，则将 IPEN 和 PRXx 均置 1。
4. 将 RxX9 位置 1，使能 9 位数据接收。
5. 将 ADRENx 位置 1，使能地址检测。
6. 将 CRXENx 位置 1 使能接收。
7. 当一个第 9 位置 1 的字符从 RSR 传输到接收缓冲器时，将 RCxIF 中断标志 位置 1。如果 RCxIE 中断允许位也置 1 还将产生中断。

8. 读 RSCTLx 寄存器获取错误标志位。第 9 个数据位始终置 1。
9. 读 RXSDRx 寄存器，从接收缓冲器获取接收到的 8 个低数据位。由软件判断此地址是否为本地器件的地址。
10. 如果发生溢出，通过清 0 CRXENx 接收器使能位清 0 OVFERx 标志。
11. 如果是对当前器件寻址，将 ADRENx 位清 0 以允许下一个字符所有接收到的数据进入接收缓冲器并产生中断。

### 15.3.4 全双工操作时钟的精确性

内部振荡器在出厂时做了校准。但是，VDD 或温度变化时频率有可能发生漂移，这将直接影响全双工异步波特率。下面的方法可用来调整波特率时钟，但要某种参考时钟源。

这种方法是调整波特率发生器的值。自动波特率检测可自动完成这种调整（见章节“自动波特率检测”）。调整波特率发生器以补偿外设时钟频率的逐渐变化时，可能分辨率精度不够。

## 15.4 USART 半双工模式

半双工同步串行通信通常用在具有一个主控制器件和一个或多个从动器件的系统中。主控制器件包含产生波特率时钟所必需的电路，并为系统中的所有器件提供时钟。从动器件可以使用主控时钟，因此无需内部时钟发生电路。

在半双工同步模式下，有 2 条信号线：双向数据线和时钟线。从动器件使用主控制器件提供的时钟，将数据串行移入或移出相应的接收和发送移位寄存器。半双工是指：主控制器件和从动器件都可以接收和发送数据，但是不能同时进行接收或发送。USARTx 既可以作为主控制器件，也可以作为从动器件。

半双工同步模式发送无需使用起始位和停止位。

### 15.4.1 USARTx 半双工主控模式

下列位用来将 USARTx 配置为半双工同步主控操作：

- SYNCx = 1
- CSRSx = 1 (主模式)      CSRSx = 0 (从模式)
- SRXENx = 0 (关闭单字节接收)；SRXENx = 1 (用于接收)
- CRXENx = 0 (连续接收关闭)；CRXENx = 1 (用于接收)
- SPENx = 1

将 TSCTLx 寄存器的 SYNCx 位置 1，可将 USARTx 配置用于半双工同步操作。将 TSCTLx 寄存器的 CSRSx 位置 1，将器件配置为主控制器件。将 RSCTLx 寄存器的 SRXENx 和 CRXENx 位清 0，以确保器件处于发送模式，否则器件配置为接收模式。将 RSCTLx 寄存器的 SPENx 位置 1，使能 USARTx。如果 RXx/DTx 或 TXx/CKx 引脚与模拟外设共用，必须清 0 相应的 ANSEL 位禁止模拟 I/O 功能。

半双工同步数据传输使用独立的时钟线半双工同步传输数据。配置为主控制器件的器件在 TXx/CKx 引脚发送时钟信号。当 USARTx 被配置为半双工同步发送或接收操作时，TXx/CKx 输出驱动器自动使能。串行数据位在每个时钟的上升沿发生改变，以确保它们在下降沿有效。每个数据位的时间为一个时钟周期，有多少数据位就只能产生多少个时钟周期。

由 BRCTLx 寄存器的 SCKPSx 位选择时钟极性。将 SCKPSx 位置 1 将时钟空闲状态设置为高电平。当 SCKPSx 位置 1 时，数据在每个时钟的下降沿发生改变。清 0 SCKPSx 位，将时钟空闲状态设置为低电平。当清 0 SCKPSx 位时，数据在每个时钟的上升沿发生改变。

### 15.4.1.1 半双工主控发送

由器件的 RXx/DTx 引脚输出数据。当 USARTx 配置为半双工同步主控发送操作时，器件的 RXx/DTx 和 TXx/CKx 输出引脚自动使能。

向 TXSDRx 寄存器写入一个字符开始发送。如果发送移位寄存器中仍保存全部或部分前一字符，新的字符数据保存在 TXSDRx 中，直到发送完前一字符的停止位为止。如果是第一个字符，或者前一个字符已经完全从发送移位中移出，则 TXSDRx 中的数据会被立即传输到发送移位寄存器。当字符从 TXSDRx 传输到发送移位后会立即开始发送数据。

每个数据位在主控时钟的上升沿发生改变，并保持有效，直至下一个时钟的上升沿为止。

注：发送移位寄存器并未映射到数据存储中，因此用户不能直接访问它。

#### 半双工主控发送设置：

1. 初始化 EUBRGHx 和 EUBRGLx 这对寄存器以及 HBRGx 和 BxRG16 位，以获得所需的波特率（见章节“波特率发生器”）。
2. 将 SYNCx、SPENx 和 CSRSx 位置 1，使能半双工同步主控串行端口。
3. 将 SRXENx 和 CRXENx 位清 0，禁止接收模式。
4. 将 TXENx 位置 1 使能发送模式。
5. 如果需要发送 9 位字符，将 TxX9 置 1。
6. 若需要中断，将中断使能寄存器中的 TXxIE 位，以及 INTCTL 寄存器中的 AIE 和 PUIE 位置 1。如果使用优先级中断，则将 IPEN 和 PTXx 均置 1。
7. 如果选择发送 9 位字符，应该将第 9 位数据装入 TxX9D 位。
8. 通过将数据装入 TXSDRx 寄存器启动发送。

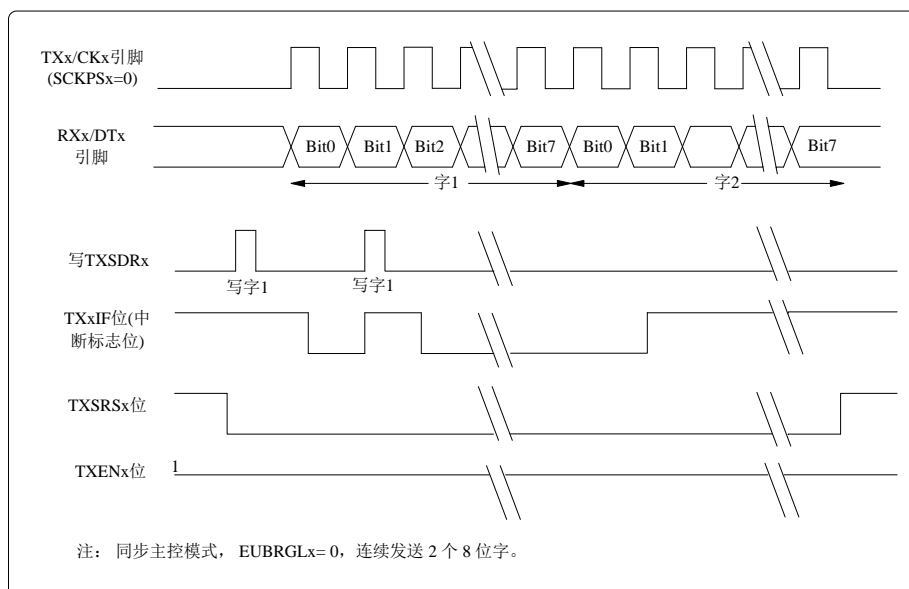


图 15.12 半双工同步发送 (SCKPSx=0)

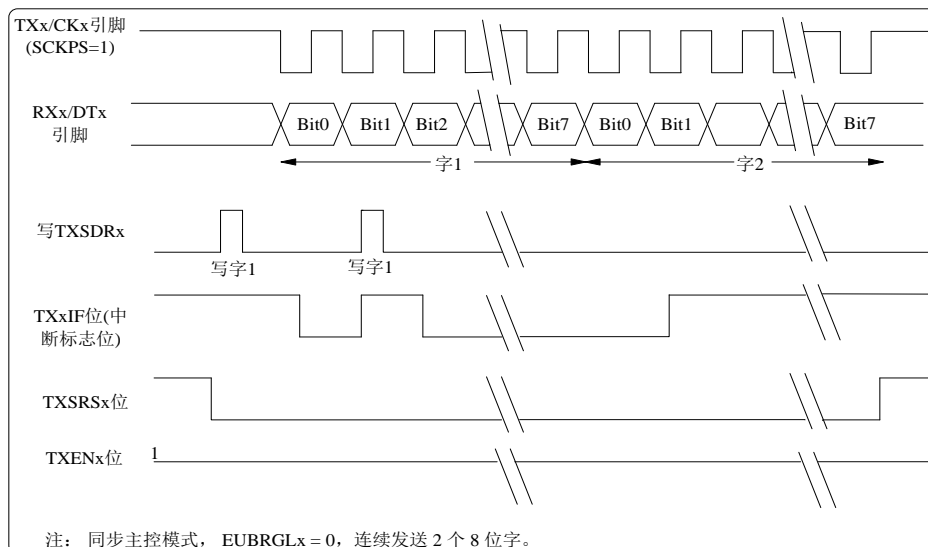


图 15.13 半双工同步发送(SCKPSx=1)

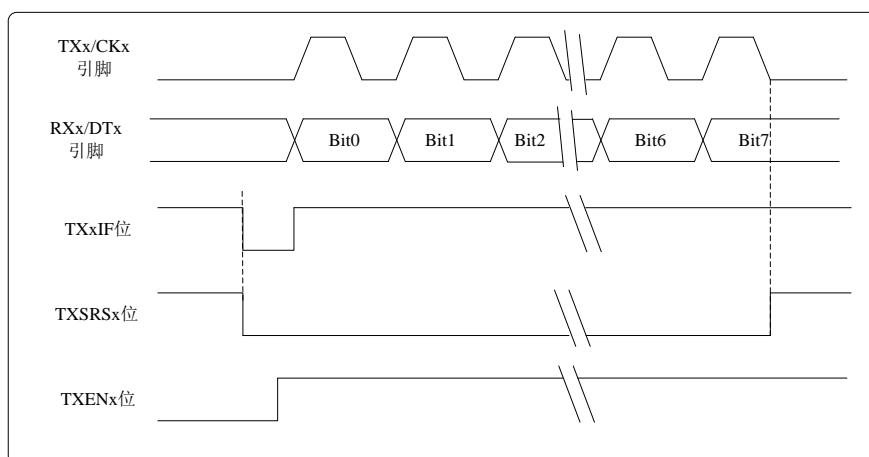


图 15.14 半双工同步发送（通过 TXENx）

### 15.4.1.2 半双工主控接收

在 RXx/DTx 引脚接收数据。当 USARTx 配置为半双工同步主控接收时，自动禁止器件的 RXx/DTx 引脚的输出驱动器。在半双工同步模式下，将单字符接收使能位（RSCTLx 寄存器的 SRXENx 位）或连续接收使能位（RSCTLx 寄存器的 CRXENx 位）置 1 使能接收。

当将 SRXENx 置 1，CRXENx 位清 0 时，一个单字符中有多少数据位就只能产生多少时钟周期。一个字符传输结束后，自动清 0 SRXENx 位。当 CRXENx 置 1 时，将产生连续时钟，直到清 0 CRXENx 为止。如果 CRXENx 在一个字符的传输过程中清 0，则 CK 时钟立即停止，并丢弃该不完整的字符。如果 SRXENx 和 CRXENx 都置 1，则当第一个字符传输完成时，SRXENx 位被清 0，CRXENx 保持。

将 SRXENx 或 CRXENx 位置 1，启动接收。在 TXx/CKx 时钟引脚信号的下降沿采样 RXx/DTx 引脚上的数据，并将采样到的数据移入接收移位寄存器（RSR）。当 RSR 接收到一个完整字符时，将 RCxIF 位置 1，字符自动移入 2 字节接收 FIFO 缓冲器。接收 FIFO 缓冲器中最顶端字符的低 8 位可通过 RXSDRx 读取。只要接收 FIFO 缓冲器中仍有未读字符，则 RCxIF 位就保持置 1 状态。



半双工同步数据传输使用与数据线通读的独立时钟线。配置为从器件的器件接收 TXx/CKx 线上的时钟信号。当器件被配置为半双工同步从发送或接收操作时，TXx/CKx 引脚的输出驱动器自动被禁止。串行数据位在时钟信号的前沿改变，以确保其在每个时钟的后沿有效。每个时钟周期只能传输一位数据，因此有多少数据位要传输就必须接收多少个时钟。

接收 FIFO 缓冲器可以保存 2 个字符。在读 RXSDRx 以访问 FIFO 缓冲器之前，若完整地接收到第 3 个字符，则产生溢出错误。此时，RSCTLx 寄存器的 OVFERx 位会置 1。FIFO 缓冲器中先前的数据不会被改写。可以读取 FIFO 缓冲器内的 2 个字符，但是在错误被清除前，不能再接收其它字符。只能通过清除溢出条件，将 OVFERx 位清 0。如果发生溢出时，SRXENx 位为置 1 状态，CRXENx 位为清 0 状态，则通过读 RXSDRx 寄存器清除错误。如果溢出时，CRXENx 为置 1 状态，则可以清 0 RSCTLx 寄存器的 CRXENx 位或清 0 SPENx 位以复位 USARTx，从而清除错误。

#### 半双工主控接收设置：

1. 用正确的波特率初始化 EUBRGHx:EUBRGLx 寄存器。按需要将 HBRGx 和 BRG16x 位置 1 或清 0，以获得所需的波特率。
2. 将 SYNCx、SPENx 和 CSRSx 位置 1 使能半双工同步主控串行端口。
3. 确保将 CRXENx 和 SRXENx 位清 0。
4. 如果使用中断，将 INTCTL 寄存器的 AIE 和 PUIE 位置 1，并将中断使能寄存器的 RCxIE 位也置 1。如果使用优先级中断，则将 IPEN 和 PRXx 均置 1。
5. 如果需要接收 9 位字符，将 RxX9 位置 1。
6. 将 SRXENx 位置 1，启动接收，或将 CRXENx 位置 1 使能连续接收。
7. 当字符接收完毕后，将 RCxIF 中断标志位置 1。如果允许位 RCxIE 置 1，还会产生一个中断。
8. 读 RSCTLx 寄存器以获取第 9 个数据位（使能 9 位接收时），并判断接收过程中是否产生错误。
9. 读 RXSDRx 寄存器获取接收到的 8 位数据。
10. 如果产生溢出错误，清 0 RSCTLx 寄存器的 CRXENx 位或清 0 SPENx 以复位 USARTx 来清除错误。

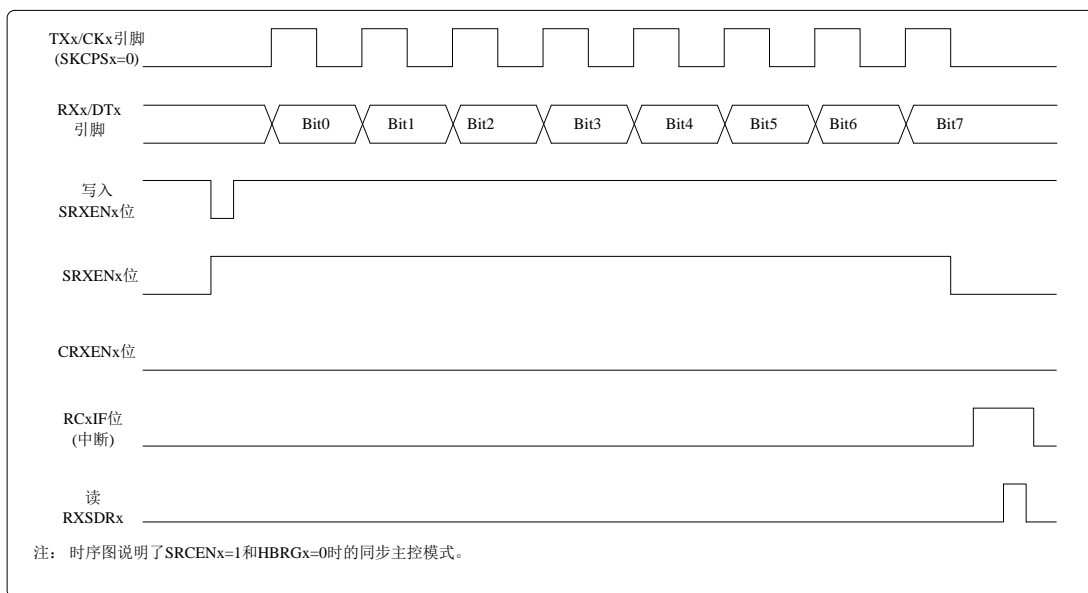


图 15.15 半双工同步接收（主控模式，SRXENx=1，SCKPSx=0）

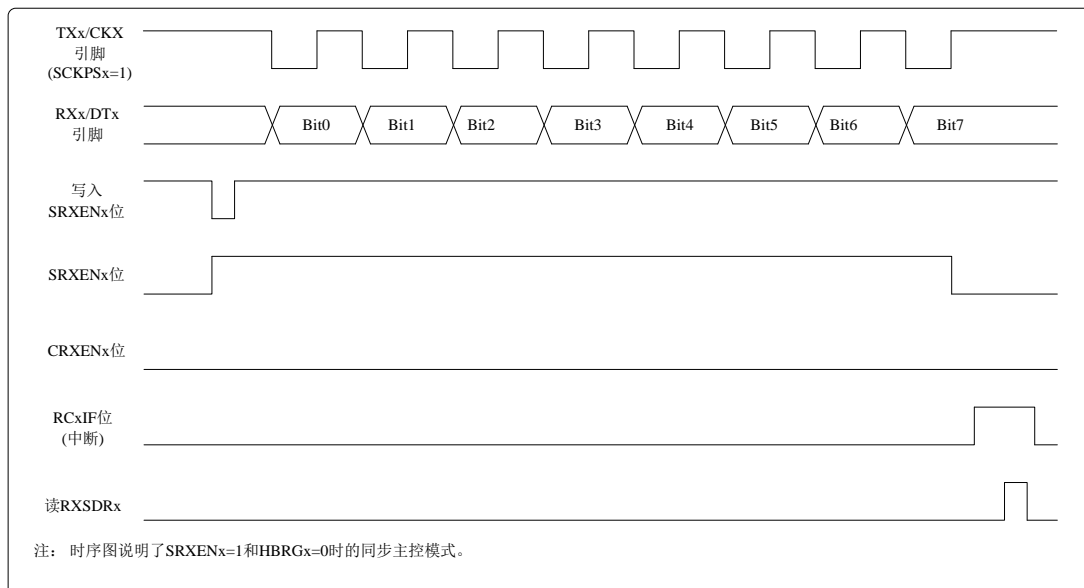


图 15.16 半双工同步接收（主控模式，SRXENx=1，SCKPSx=1）

## 15.4.2 USART 半双工从动模式

下列位用来将 USART<sub>x</sub> 配置为半双工同步从动操作：

- SYNC<sub>x</sub>=1
- CSRS<sub>x</sub>=0
- SRXEN<sub>x</sub>=0（用于发送）；SRXEN<sub>x</sub> = 1（用于接收）
- CRXEN<sub>x</sub>=0（用于发送）；CRXEN<sub>x</sub> = 1（用于接收）
- SPEN<sub>x</sub>=1

将 TSCTL<sub>x</sub> 寄存器的 SYNC<sub>x</sub> 位置 1，可将器件配置用于半双工同步操作。将 TSCTL<sub>x</sub> 寄存器的 CSRS<sub>x</sub> 位置 1，将器件配置为从动器件。将 RSCTL<sub>x</sub> 寄存器的 SRXEN<sub>x</sub> 和 CRXEN<sub>x</sub> 位清 0，以确保器件处于发送模式，否则器件将被配置为接收模式。将 RSCTL<sub>x</sub> 寄存器的 SPEN<sub>x</sub> 位置 1，使能 USART<sub>x</sub>。如果 RX<sub>x</sub>/DT<sub>x</sub> 或 TX<sub>x</sub>/CK<sub>x</sub> 引脚与模拟外设共用，必须清 0 相应的 ANSEL 位禁止模拟 I/O 功能。

### 15.4.2.1 USART 半双工从动发送

除休眠模式外，半双工主控和从动模式的工作原理是相同的（见章节“半双工主控发送”）。

如果向 TXSDRx 写入 2 个字，然后执行 IDLE 指令，则会出现下列情况：

1. 第一个字立即传输到发送移位寄存器并进行发送。
2. 第二个字留在 TXSDRx 寄存器中。
3. TX<sub>x</sub>IF 中断标志位不会置 1。
4. 当第一个字符移出发送移位时，TXSDR 寄存器将把第二个字符传输到发送移位，然后标志位 TX<sub>x</sub>IF 置 1。
5. 如果 PUIE 和 TX<sub>x</sub>IE 位都置 1，则由中断将器件从休眠模式唤醒，然后执行下一条指

令。如果 AIE 位也置 1，程序将调用中断服务程序。

**USART 半双工从动发送设置：**

1. 将 SYNC<sub>x</sub> 和 SPEN<sub>x</sub> 位置 1 并将 CSRS<sub>x</sub> 位清 0。
2. 将 CRXEN<sub>x</sub> 和 SRXEN<sub>x</sub> 位清 0。
3. 如果使用中断，将 INTCTL 寄存器的 AIE 和 PUIE 位置 1，并将中断使能寄存器的 TX<sub>x</sub>IE 位也置 1。如果使用优先级中断，则将 IPEN 和 PTX<sub>x</sub> 均置 1。
4. 如果需要发送 9 位数据，将 TxX9 位置 1。
5. 将 TXEN<sub>x</sub> 位置 1 使能发送。
6. 若选择发送 9 位数据，将最高位写入 TxX9D 位。
7. 将低 8 位数据写入 TXSDRx 寄存器开始传输。

### 15.4.2.2 USART 半双工从动接收

除了以下不同外，半双工主控和从动模式的工作原理相同。（见章节“半双工主控接收”）。

- 休眠模式
- CRXEN<sub>x</sub> 位总是置 1，因此接收器不能进入空闲状态。
- SRXEN<sub>x</sub> 位，在从动模式可为“任意值”。

如果在进入休眠模式之前，已经将 CRXEN<sub>x</sub> 位置 1，则在休眠模式仍可接收字符。RSR 寄存器接收到字后，就会立即将接收到的数据传输到 RXSDRx 寄存器。如果将 RC<sub>x</sub>IE 允许位置 1，则产生的中断将使器件从休眠模式唤醒，然后执行下一条指令。如果 AIE 位也置 1，则程序将跳转到中断向量处执行。

**半双工从动接收设置：**

1. 将 SYNC<sub>x</sub> 和 SPEN<sub>x</sub> 位置 1 并将 CSRS<sub>x</sub> 位清 0。
2. 如果使用中断，将 INTCTL 寄存器的 AIE 和 PUIE 位置 1，并将中断使能寄存器的 RC<sub>x</sub>IE 位也置 1。如果使用优先级中断，则将 IPEN 和 PRX<sub>x</sub> 均置 1。
3. 如果需要接收 9 位字符，将 RxX9 位置 1。
4. 将 CRXEN<sub>x</sub> 位置 1，使能接收。
5. 当接收完成后，将 RC<sub>x</sub>IF 位置 1。如果 RC<sub>x</sub>IE 已置 1，还会产生一个中断。
6. 如果使能 9 位模式，从 RSCTL<sub>x</sub> 寄存器的 RxX9D 位获取最高位。
7. 读 RXSDRx 寄存器，从接收 FIFO 缓冲器获取接收到的 8 个低数据位。
8. 如果产生溢出错误，清 0 RSCTL<sub>x</sub> 寄存器的 CRXEN<sub>x</sub> 位或清 0 SPEN<sub>x</sub> 位以复位 USART<sub>x</sub> 来清除错误。

### 15.4.3 USART 半双工 RS-485 模式

请参考章节 RS-485 发送/接收。

## 15.5 USART 单线通信模式

USART 单线通信模式（半双工异步模式）通过 UPINSETx 寄存器的 USLMx 位置 1 使能；在使能 USART 单线通信模式之前，须确保 TSCTLx 寄存器的 SYNCx 位保持清零状态。

USART 单线通信模式下，USART 相关引脚特性如下：

- RX/DT 引脚不再使用，总是被释放，即可作为通用 I/O 口使用；
- TX/CK 引脚作为数据通信脚；
- 当 USART 配置为单线通信接收时，TX/CK 引脚一直处于接收状态；
- 当 USART 配置为单线通信发送时：
  - 当有数据发送时，TX/CK 引脚处于数据发送状态；
  - 当无数据发送时（空闲状态），TX/CK 引脚被释放，表现为通用 I/O 口，因此在使用前需要将 TX/CK 引脚配置为数字输出口，并输出高电平（根据 USART 通信协议，当 USART 处于空闲状态时，TX/CK 被拉高）。

用户可通过软件来管理线上冲突。

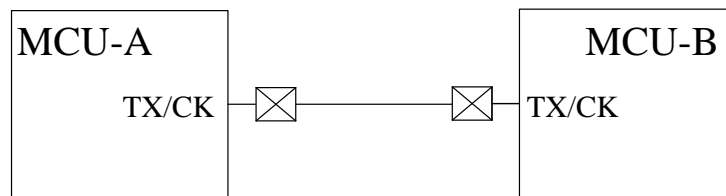


图 15.17 USART 单线通信外接连线示意图

### 15.5.1 USART 引脚配置寄存器 UPINSETx

UPINSETx 寄存器主要用于 USARTx 的单线通信模式使能和相关引脚配置；

寄存器：UPINSETx:USARTx 引脚配置寄存器

复位值	bit7							bit0
0000 0000	USLMx	-	-	-	-	-	-	-
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

USLMx: USARTx 单线通信模式使能位

USLMx = 1 使能串口单线通信模式

USLMx = 0 禁止串口单线通信模式

## 15.6 7816 模式

7816 模式基于 ISO/IEC 7816-3 标准，利用此功能可与其他 7816 设备进行通信。按照 7816 协议标准，7816 基本接口时序如图所示：

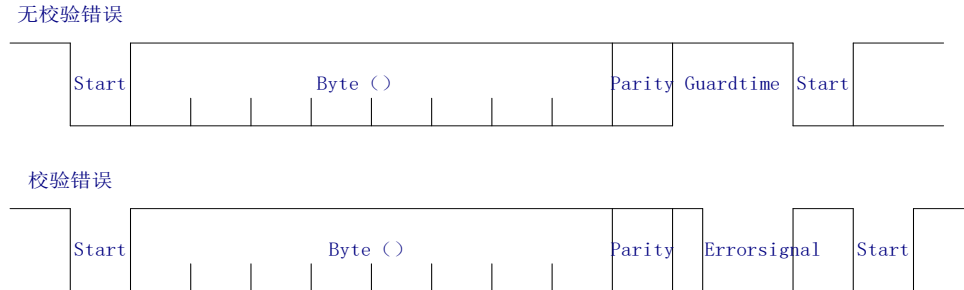


图 15.18 7816 接口时序

- 1, 一个起始位后跟 8 个数据位以及一个校验位，以 2etu 的 guard time 结束。
- 2, 第 10.5 个 etu 接收电路校验接收数据，如果正确，插入 2etu 的 guard time，确保数据长度为 12etu，完成数据发送；若校验错误，则在第 10.5etu 拉低 IO，产生 error signal。error signal 长度可从 1etu、2etu、1.5etu 中选择。
- 3, 第 11 个 etu 时发送电路未采样到 error signal，则说明发送数据正确，数据发送完成。
- 4, 若第 11 个 etu 发送电路采样到 error signal，则说明发送数据错误，等待 2 个 etu 后重发数据。

### 15.6.1 7816 模式发送

#### 15.6.1.1 发送设置

通过配置如下控制位使能 USARTx 发送器，以用于 7816 模式发送操作：

- 1) U7816ENx = 1 使能 7816 模式
- 2) SPENx = 1 使能串口
- 3) SYNCx = 0 选择异步模式
- 4) CLKOUTx = 1 使能 7816 时钟输出
- 5) TxX9 = 1 选择 9 位数据发送
- 6) TX9DSELx = 1 选择自动生成奇偶校验码
- 7) STOPx = 0 选择 2 位停止位
- 8) TXENx = 1 使能发送端
- 9) TPARx、TINVx、TCONVx、BGTENx、PSELx、TREPENx、TREPx、EGTx 等控制寄存器视情况进行选择
- 10) 其他设置请参考章节“全双工发送的设置”

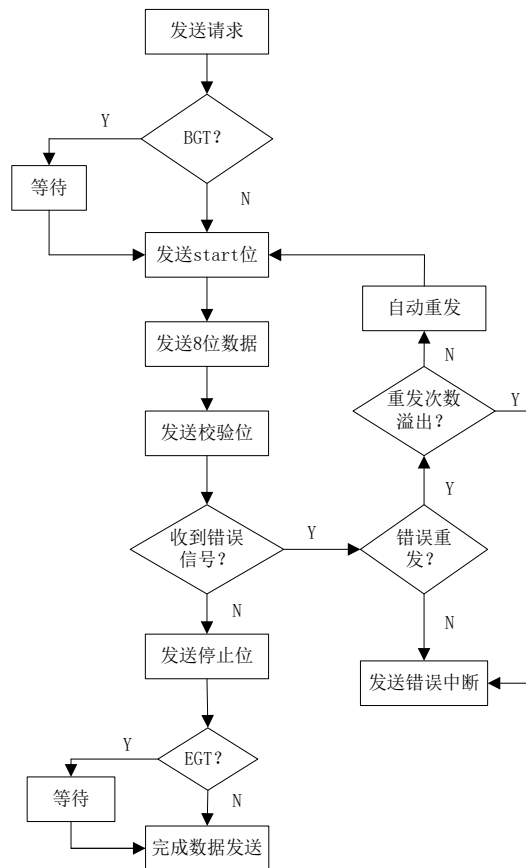


图 15.19 数据发送流程

### 15.6.2 7816 模式接收

通过配置如下控制位使能 USARTx 接收器，以用于 7816 接收操作：

- 1) CRXEN<sub>x</sub> = 1 使能接收端
- 2) U7816EN<sub>x</sub> = 1 使能 7816 模式
- 3) SPEN<sub>x</sub> = 1 使能串口
- 4) SYNC<sub>x</sub> = 0 选择异步模式
- 5) RxX9 = 1 选择 9 位数据接收
- 6) CLKOUT<sub>x</sub> = 1 使能 7816 时钟输出

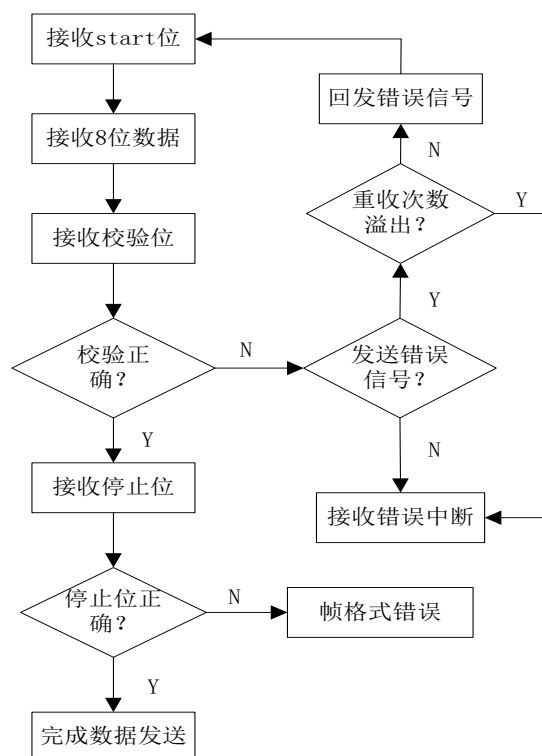


图 15.20 数据接收流程

### 15.6.3 7816 模式寄存器

#### 15.6.3.1 7816 发送控制寄存器 U7816TXCTL2

**寄存器:** U7816TXCTL2: 7816发送控制寄存器2(地址:163H)

bit7							bit0	
复位值 0001 0000	U7816EN2	CLKOUT2	TX9DSEL 2	STOP2	TPAR2	TINV2	TCONV2	BGTEN2
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

U7816EN2: 7816 模式使能控制

1 = 使能 7816 模式

0 = 禁止 7816 模式

CLKOUT2: 7816 时钟输出使能控制

1 = 使能时钟输出

0 = 禁止时钟输出

TX9DSEL2: TX9D 位数据选择

1 = 选择自动生成奇偶校验

0 = 选择软件输入

STOP2: 停止位宽度选择

1 = 停止位宽度为 1etu

- 0 = 停止位宽度为 2etu
- TPAR2: 发送时奇偶校验选择
  - 1 = 奇校验
  - 0 = 偶校验
- TINV2: 7816 数据发送编码方式选择
  - 1 = 发送反相电平
  - 0 = 发送正相电平
- TCONV2: 发送次序选择
  - 1 = 先发送 MSB
  - 0 = 先发送 LSB
- BGTEN2: BGT(block guard time)(块保护时间, 为两个相对方向发送的连续字符的起始沿之间的最短时间, 因此一个已接收块的最后一个字符与一个被传输块的第一个字符之间的延迟至少应为 BGT)控制位, 控制接收到发送之间是否插入 BGT
  - 1 = 插入 BGT, 宽度为 22etu
  - 0 = 不插入 BGT

注: (1) TX9DSEL2、STOP2、TPAR2、TCONV2 在 7816 模式禁止时也可使用;  
 (2) 第九位需要软件输入时, 必须先把 TX9DSEL2 置 0, 再输入 TX9 的值, 否则可能会出错;  
 (3) 要使用 BGTEN2 功能时, 必须保证 TXEN2 同时也使能, 否则 BGT 不能正常工作。

### 15.6.3.2 7816 接收控制寄存器 U7816RXCTL2

**寄存器:** U7816RXCTL2: 7816接收控制寄存器2(地址:167H)

	bit7						bit0	
复位值 0000 0000	ERSW21	ERSW20	RPAR2	RINV2	RCONV2	-	-	PAREF2
	R/W	R/W	R/W	R/W	R/W	U	U	R

- ERSW2[1:0]: error signal 宽度选择
  - 01 =1etu
  - 00 =2etu
  - 1X =1.5etu
- RPAR2: 接收时奇偶校验选择
  - 1 = 奇校验
  - 0 = 偶校验
- RINV2: 数据接收编码方式选择
  - 1 = 接收反相电平
  - 0 = 接收正相电平
- RCONV2: 接收次序选择
  - 1 = 先接收 MSB
  - 0 = 先接收 LSB
- PAREF2: 奇偶校验错误标志位
  - 1 = 发送或接收上以字节数据奇偶校验错误
  - 0 = 发送或接收上以字节数据奇偶校验正确



注：RINV2、RCONV2 在 7816 模式禁止时也可使用。

### 15.6.3.3 7816 控制寄存器 U7816CTL2

寄存器： U7816CTL2: 7816控制寄存器2(地址:166H)

		bit7						bit0	
复位值	0000 0000	PSEL2	-	TREPEN2	TREP21	TREP20	RREPEN2	RREP21	RREP20
		R/W	U	R/W	R/W	R/W	R/W	R/W	R/W

PSEL2: 数据通道选择

1 = 选择通道 1 (RX21/DT21)

0 = 选择通道 0 (RX20/DT20)

TREPEN2: 重发使能控制

1 = 收到校验错误信号后重发，达到最大次数后进入中断

0 = 收到校验错误信号后禁止重发，直接中断

TREP2[1:0]: 重发的最大次数

TREP2=1-3

RREPEN2: 重收使能控制

1= 奇偶校验错误后重收数据，达到最大次数后进入中断

0= 奇偶校验错误后直接进入中断

RREP2[1:0]: 重接收的最大次数

RREP2=1-3

### 15.6.3.4 预分频控制寄存器 CLKDIV2

寄存器： CLKDIV2: 预分频控制寄存器2(地址:165H)

		bit7							bit0
复位值	0000 0000	CLKDIV2	CLKDIV2	CLKDIV2	CLKDIV2	CLKDIV2	CLKDIV2	CLKDIV2	CLKDIV2
		7	6	5	4	3	2	1	0
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CLKDIV2[7:0]:7816 工作时钟和引脚输出时钟控制

当 CLKDIV1 的值不为 0 时，7816 工作时钟  $F_{7816}$  和系统时钟 SCLK 之间的关系为：

$$F_{7816} = SCLK / [2 (CLKDIV2 + 1)]$$

当 CLKDIV2 的值为 0 时， $F_{7816}$  和 SCLK 关系为： $F_{7816} = SCLK$

注：7816 协议规定 7816 工作时钟频率范围为 1-5M。

### 15.6.3.5 EGT 控制寄存器 EGTCTL2

寄存器： EGTCTL2: EGT控制寄存器2(地址:164H)

		bit7							bit0
复位值	0000 0000	EGT27	EGT26	EGT25	EGT24	EGT23	EGT22	EGT21	EGT20
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

EGT2[7:0]: 发送时插入的 EGT(extra guard time)宽度 (单位 etu)

0 = 发送时不插入 EGT

1-255 = 发送时插入 EGT, 宽度为 EGT2 的值

注: EGTCTL2 在发送无校验错误时正常工作, 若有校验错误时无论为何值均不插入 EGT。

## 16 SSCI 模块

### 16.1 概述

KF8A100 包含 2 个 SSCI (Synchronous Serial Communication interface) 同步串行通信接口，它是用于其他外设或单片机进行通信的串行接口。为了方便，本文章叙述为 SSCI1 和 SSCI2，SSCI1 和 SSCI2 完全相同。SSCI1/2 包含两种工作模式：

- ◆ I2C (Inter Intergrated Circuit) 接口模式。
- ◆ 串行外设接口 (Serial Peripheral Interface, SPI) 模式

### 16.2 SSCI 相关寄存器

表 16.1 与 SSCI1/2 相关的寄存器

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
135H	SSCI1CTL0	SSCI1WCFL	SSCI1OV	SSCI1EN	SSCI1CKP	SSCI1MOD3	SSCI1MOD2	SSCI1MOD1	SSCI1MOD0
133H	SSCI1CTL1	-	-	-	-	-	-	SSCI1CKS1	SSCI1CKS0
138H	SSCI1CTL2	SSCI1CALL EN	SSCI1ACK STA	SSCI1ACK DAT	SSCI1ACK EN	SSCI1RC EN	STOP EN1	RESTART EN1	STARTEN1
134H	SSCI1STA	SAMPLE1	CKEGE1	SSCI1DA	SSCI1STOP	SSCI1START	SSCI1RW	SSCI1UA	SSCI1BUF
136H	SSCI1BUFR	SSCI 数据接收缓冲/发送寄存器							
137H	SSCI1ADD	SSCI1ADD7	SSCI1ADD6	SSCI1ADD5	SSCI1ADD4	SSCI1ADD3	SSCI1ADD2	SSCI1ADD1	SSCI1ADD0
	SSCI1MSK	SSCI1MSK7	SSCI1MSK6	SSCI1MSK5	SSCI1MSK4	SSCI1MSK3	SSCI1MSK2	SSCI1MSK1	SSCI1MSK0
205H	SSCI2STA	SAMPLE2	CKEGE2	SSCI2DA	SSCI2STOP	SSCI2START	SSCI2RW	SSCI2UA	SSCI2BUF
206H	SSCI2CTL0	SSCI2WCFL	SSCI2OV	SSCI2EN	SSCI2CKP	SSCI2MOD3	SSCI2MOD2	SSCI2MOD1	SSCI2MOD0
207H	SSCI2BUFR	SSCI2 数据接收缓冲/发送寄存器							
208H	SSCI2ADD	SSCI2ADD7	SSCI2ADD6	SSCI2ADD5	SSCI2ADD4	SSCI2ADD3	SSCI2ADD2	SSCI2ADD1	SSCI2ADD0
	SSCI2MSK	SSCI2MSK7	SSCI2MSK6	SSCI2MSK5	SSCI2MSK4	SSCI2MSK3	SSCI2MSK2	SSCI2MSK1	SSCI2MSK0
209H	SSCI2CTL2	SSCI2CALLEN	SSCI2ACKSTA	SSCI2ACKDAT	SSCI2ACKEN	SSCI2RCEN	STOPEN2	RESTARTEN2	STARTEN2
21FH	SSCI2CTL1	-	-	-	-	-	-	SSCI2CKS1	SSCI2CKS0

注：SSCI1 与 SSCI2 所使用的寄存器，除了序号与地址不一样，其他内容及说明均相同，为了简便，除特殊说明外，本章中“x”= 1/2，用以代表 SSCI 的序号，用户只需区别 1 和 2 即可。

## 16.2.1 SSCIx 控制寄存器 0 (SSCIxCTL0)

寄存器: SSCIxCTL0: SSCI控制寄存器0

	bit7				bit0			
复位值	SSCIxWCF	SSCIxOV	SSCIxEN	SSCIxCKP	SSCIxMO	SSCIxMO	SSCIxMO	SSCIxMO
0000 0000	L				D3	D2	D1	D0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SSCIxWCFL: 写冲突检测位

1 = 正在发送前一个字时,又有数据写入SSCIxBUFR寄存器(必须用软件清零)

0 = 无冲突

SSCIxOV: 接收溢出指示位

在SPI 模式下:

1 = 当SSCIxBUFR 中仍保存前一数据时,又接收到一个新的字节。如果溢出,SSCISR<sub>x</sub> 中的数据会丢失。溢出只会在从动模式下发生。即使只是发送数据,用户也必须读SSCIxBUFR,以避免将溢出标志位置1。在主控模式下,溢出位不会被置1,因为每次接收(和发送)新数据都是通过写入SSCIxBUFR寄存器启动。

0 = 无溢出

在I2C模式下:

1 = SSCIxBUFR中仍保存前一数据时,又接收到一个新的字节。SSCIxOV在发送模式下被忽略。两种模式下都必须用软件将SSCIxOV清零。

0 = 无溢出

SSCIxEN: 同步串行端口使能位

在SPI模式下:

1 = 使能串行端口并将SCK<sub>x</sub>、SDO<sub>x</sub>和SDI<sub>x</sub>配置为串行端口引脚

0 = 禁止串行端口并将这些引脚配置为I/O端口引脚

在I2C模式下:

1 = 使能串行端口并将SDA<sub>x</sub>和SCL<sub>x</sub>引脚配置为串行端口引脚

0 = 禁止串行端口并将这些引脚配置为I/O端口引脚

在两种模式下,当使能时,这些引脚必须被正确配置为输入或输出。

SSCIxCKP: 时钟极性选择位

在SPI 模式下:

1 = 空闲状态时,时钟为高电平

0 = 空闲状态时,时钟为低电平

在I2C模式下:SCK<sub>x</sub> 释放控制

1 = 使能时钟

0 = 保持时钟为低电平(时钟低电平时间延长)。(用于确保数据建立时间。)

SSCIxMOD&lt;3:0&gt;: 同步串行端口模式选择位

0000 = SPI 主控模式,时钟 = SCLK/4

0001 = SPI 主控模式,时钟 = SCLK /16

0010 = SPI 主控模式,时钟 = SCLK /64

0011 = SPI 主控模式,时钟 = T2输出/2

- 0100 = SPI 从动模式，时钟 = SCKx 引脚。使能  $\overline{SS}$  引脚控制。
- 0101 = SPI 从动模式，时钟 = SCKx 引脚。禁止  $\overline{SS}$  引脚控制。 $\overline{SS}$  可作为 I/O 引脚使用。
- 0110 = I2C 从动模式，7 位地址
- 0111 = I2C 从动模式，10 位地址
- 1000 = I2C 主控模式，时钟 = SCLK / (4 \* (SSCIxADD + 1))
- 1001 = 禁止装载功能
- 1010 = 保留
- 1011 = I2C 固件控制主控模式（从动空闲模式）
- 1100 = 保留
- 1101 = 保留
- 1110 = I2C 从动模式，7 位地址，并允许启动位和停止位中断
- 1111 = I2C 从动模式，10 位地址，并允许启动位和停止位中断

### 16.2.2 SSCIx 控制寄存器 1 (SSCIxCTL1)

寄存器: SSCIxCTL1: SSCIx 控制寄存器1

复位值 ----00	bit7						SSCIxCKS	SSCIxCKS	bit0
	-	-	-	-	-	-	1	0	
	U	U	U	U	U	U	R/W	R/W	

SSCIxCKS<1:0> SSCIx 工作时钟选择寄存器  
00 = 选用 SCLK 为 SSP 工作时钟

### 16.2.3 SSCIx 控制寄存器 2 (SSCIxCTL2)

寄存器: SSCIxCTL2: SSCIx 控制寄存器2

复位值 0000 0000	bit7						bit0	
	SSCIxCA LLEN	SSCIxAC KSTA	SSCIxAC KDAT	SSCIxAC KEN	SSCIxRCE N	STOPENx	RESTART SENx	STARTEN x
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SSCIxCALLEN: 广播呼叫使能位（仅限 I2C 从动模式）  
1 = 允许在 SSCIxSRx 中接收到广播呼叫地址（0000h）时产生中断  
0 = 禁止广播呼叫地址

SSCIxACKSTA: 应答状态位（仅限于 I2C 主控模式）  
在主控发送模式下:  
1 = 未接收到来自从动器件的应答。  
0 = 已接收到来自从动器件的应答

SSCIxACKDAT: 应答数据位（仅限于 I2C 主控模式）  
在主控接收模式下: 用户在接收完成后发送的应答序列的值  
1 = 不应答

0 = 应答

SSCIxACKEN: 应答序列使能位 (仅限I2C主控模式)

在主控接收模式下:

1 = 在SDAx 和SCLx 引脚启动应答序列, 发送SSCIxACKDAT数据位。由硬件自动清零。

0 = 应答序列空闲

SSCIxRCEN: 接收使能位 (仅限I2C主控模式)

1 = 使能I2C接收模式

0 = 接收空闲

STOPENx: 停止条件使能位 (仅限 I2C 主控模式)

SCK 释放控制:

1 = 在SDAx和SCLx引脚启动停止条件。由硬件自动清零。

0 = 停止条件空闲

RESTARTENx: 重复启动条件使能位 (仅限I2C主控模式)

1 = 在SDAx和SCLx 引脚启动重复启动条件。由硬件自动清零。

0 = 重复启动条件空闲

STARTENx: 启动条件使能位 (仅限I2C主控模式)

在主控模式下:

1 = 在SDAx 和SCLx引脚启动条件。由硬件自动清零。

0 = 启动条件空闲

在从动模式下:

1 = 从发送和接收都会使能时钟延长 (使能时钟延长)

0 = 禁止时钟延长

注: 对于SSCIxACKEN、SSCIxRCEN、STOPENx、RESTARTENx 和STARTENx 位: 如果I2C模块不处在空闲模式, 此位可能无法被置1 (没有假脱机 (spooling)) 且可能无法对SSCIxBUFR 进行写操作 (禁止写SSCIxBUFR)。

## 16.2.4 SSCIx 状态寄存器 (SSCIxSTA)

寄存器: SSCIxSTA: SSCIx状态寄存器

复位值	bit7						bit0	
0000 0000	SAMPLEx	CKEGEx	SSCIxDA	SSCIxSTO P	SSCIxSTA RT	SSCIxRW	SSCIxUA	SSCIxBUF
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SAMPLEx: SPI数据输入采样相位

SPI 主控模式:

1 = 在数据输出时间结束时采样输入数据

0 = 在数据输出时间中间采样输入数据

SPI 从动模式:

当SPI 用于从动模式时, 必须将SAMPLEx 清零

I<sup>2</sup>C/I2C模式:

此位必须保持清零

- CKEGEx:** SPI 时钟边沿选择位  
 SPI模式, SSCIxCKP = 0:  
 1 = 在SCKx 的下降沿发送数据  
 0 = 在SCKx 的上升沿发送数据  
 SPI 模式, SSCIxCKP = 1:  
 1 = 在SCKx 的上升沿发送数据  
 0 = 在SCKx 的下降沿发送数据  
 I2C<sup>2</sup>C模式:  
 此位必须保持清零
- SSCIxDA:** 数据/ 地址位 (I2C模式)  
 1 = 表示上次接收或发送的字节是数据  
 0 = 表示上次接收或发送的字节是地址
- SSCIxSTOP:** 停止位 (仅I2C模式)  
 当禁止SSCIx 模块或上次检测到启动位时, 该位被清零。  
 SSCIxEN 被清零。  
 1 = 表示上次检测到了停止位 (此位在复位时为0)  
 0 = 表示上次没有检测到停止位
- SSCIxSTART:** 启动位 (仅I2C模式)  
 当禁止SSCIx 模块或上次检测到停止位时, 该位被清零。  
 SSCIxEN 被清零。  
 1 = 表示上次检测到了启动位 (此位在复位时为0)  
 0 = 表示上次没有检测到启动位
- SSCIxRW:** 读/ 写信息位 (仅I2C模式)  
 该位用来保存在上次地址匹配后的SSCIxRW位信息。此位仅在地址匹配与遇到下一个启动位、停止位或SSCIxACK位之间有效。  
 1 = 读  
 0 = 写
- SSCIxUA:**更新地址位 (仅10位I2C模式)  
 1 = 表示用户需要更新SSCIxADD 寄存器中的地址  
 0 = 不需要更新地址
- SSCIxBUF:** 缓冲器满状态位  
 接收 (SPI和I2C模式):  
 1 = 接收完成, SSCIxBUFR满  
 0 = 接收未完成, SSCIxBUFR空  
 发送 (仅I2C模式):  
 1 = 正在发送, SSCIxBUFR满  
 0 = 发送完成, SSCIxBUFR空

### 16.2.5 SSCIx 屏蔽寄存器 (SSCIxMSK)

**寄存器:** SSCIxMSK: SSCIx屏蔽寄存器

复位值	bit7						bit0	
1111 1111	SSCIxMS K7	SSCIxMS K6	SSCIxMS K5	SSCIxMS K4	SSCIxMS K3	SSCIxMS K2	SSCIxMS K1	SSCIxMS K0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SSCIxMSK<7:1>:屏蔽位

1 = 接收到的地址的bit n 与SSCIxADD<n> 比较以检测I<sup>2</sup>C的地址匹配情况

0 = 接收到的地址的bit n 不用于检测I<sup>2</sup>C的地址匹配情况

SSCIxMSK<0>: 在I<sup>2</sup>C从动模式下, 10位地址的屏蔽位

在I<sup>2</sup>C 从动模式, 10位地址 (SSCIxMOD<3:0> = 0111或1111) 条件下:

1 = 将接收到的地址的bit 0位与SSCIxADD<0> 相比较以检测I<sup>2</sup>C的地址匹配情况

0 = 接收到的地址的bit 0位不用于检测I<sup>2</sup>C的地址匹配情况

在I<sup>2</sup>C从动模式, 7位地址条件下, 该位为无关位

注: SSCIxADD与SSCIxMSK共用一个地址, 当SSCIxCTL0位SSCIxMOD<3:0>=1001时, SFR地址对应SSCIxMSK寄存器; SSCIxCTL0位SSCIxMOD<3:0>不为1001时, SFR地址对应SSCIxADD寄存器。

### 16.2.6 SSCIx I2C 地址寄存器 (SSCIxADD)

在10位I<sup>2</sup>C从动模式下, 该地址寄存器是复用的。

寄存器: SSCIxADD: I2C地址寄存器

	bit7						bit0	
复位值	SSCIxAD	SSCIxAD	SSCIxAD	SSCIxAD	SSCIxAD	SSCIxAD	SSCIxAD	SSCIxAD
0000 0000	D7	D6	D5	D4	D3	D2	D1	D0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

#### 10位从动模式下——高地址字节:

SSCIxADD<7:3>: 未使用, SSCIxADD存放高地址字节时, 未使用这5位, 为无关位。主器件发送的位模式由I<sup>2</sup>C规范制定必须等于11110, 但是这些位由硬件进行比较且不受该寄存器中的值的影响

SSCIxADD<2:1>: 保存10位地址的高两位

SSCIxADD0: 未使用, 为无关位, 初始化时写0

#### 10位从动模式下——低地址字节:

SSCIxADD<7:0>: 10位地址的低8位

#### 7位从动模式下:

SSCIxADD<7:1>: 7位地址

SSCIxADD0: 未使用, 为无关位, 初始化时写0

## 16.3 I2C 模式

- ◆ 多主机模式:可用作主设备或者从设备
- ◆ I2C 主设备产生时钟, 起始和停止信号
- ◆ 检测 7 位和 10 位地址

### 16.3.1 工作原理

I<sup>2</sup>C模式下的SSCI 能实现全部从动功能 (除广播呼叫支持外), 且硬件支持启动位和



停止位中断，以便于固件实现主控功能。SSCI模式实现标准模式规范以及7位和10位寻址。有两个引脚用于数据传输：SCK1/SCL1引脚作为时钟线（SCL），而SDI/SDA引脚作为数据线（SDA）。通过将SSCI使能位SSCIEN（SSCICTL0<5>）置1以使能SSCI模块的功能。

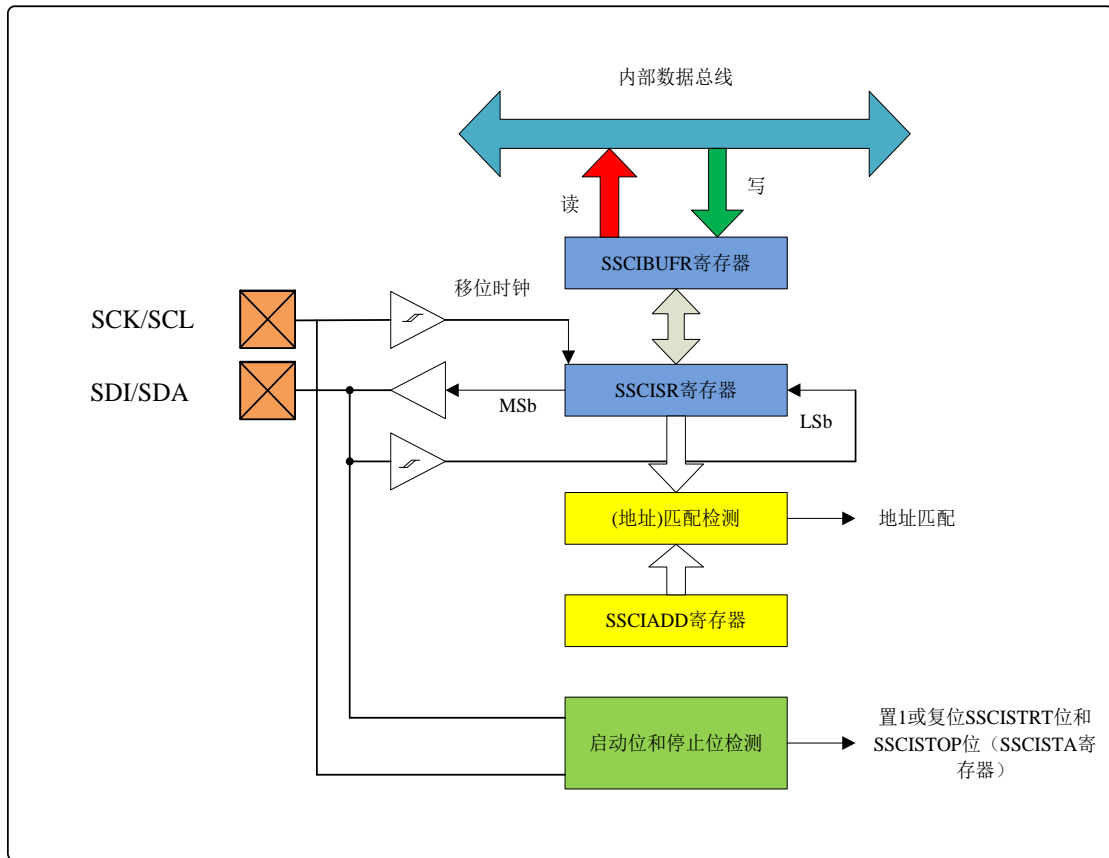


图 16.1 I2C 模式方框图

SSCI模块有7个寄存器用于I2C操作，这7个寄存器是：

- SSCI控制寄存器（SSCICTL0）
- SSCI控制寄存器1（SSCICTL1）
- SSCI控制寄存器1（SSCICTL2）
- SSCI状态寄存器（SSCISTA）
- 串行接收/发送缓冲器（SSCIBUFR）
- SSCI 移位寄存器（SSCISR）——不可直接访问
- SSCI 地址寄存器（SSCIADD）
- SSCI 屏蔽寄存器（SSCIMSK）

SSCICTL0 寄存器用于控制I2C 的工作。可通过设置四个模式选择位（SSCICTL0<3:0>）选择以下I2C 模式之一：

- I2C 从动模式（7 位地址）
- I2C 从动模式（10 位地址）
- I2C 从动模式（7 位地址），允许启动位和停止位中断以支持固件主控模式
- I2C 从动模式（10 位地址），允许启动位和停止位中断以支持固件主控模式
- 允许I2C 启动位和停止位中断以支持固件主控模式而从动模式空闲

任何I2C 模式的选择，在SSCIEN置1后都会强制SCL和SDA引脚为漏极开路（假定通

过编程将相应的TR<sub>xx</sub>位置1，ANS<sub>xx</sub>位置0，使这些引脚成为数字输入引脚）。必须在SCL和SDA引脚上外接上拉电阻，才能使I2C模块正常工作。

### 16.3.2 I2C 从动模式

在从动模式下，SCL 引脚和SDA 引脚必须被配置为输入（TR1<1:0> 置1）。必要时SSCI模块将用输出数据改写输入状态（从发送器）。

当地址匹配或在地址匹配后发送的数据被接收时，硬件会自动产生一个应答（ACK）脉冲，并把当时SSCISR寄存器中接收到的值装入SSCIBUFR寄存器。

某些条件会使SSCI 模块不发出此ACK（低电平有效）脉冲。这些条件包括（之一或全部）：

- 1) 在接收到数据前，缓冲器满标志位SSCIBUF（SSCISTA<0>）置1。
- 2) 在接收到数据前，溢出标志位SSCIOV（SSCICTL0<6>）置1。

在这些情况下，SSCISR寄存器的值不会载入SSCIBUFR，但是SSCIIF位会置1。表 16.2 显示了当已知SSCIBUF位和SSCIOV位的状态时，接收到数据发送字节时产生的结果。阴影单元显示了当用户软件没有正确将溢出状态清零时的情况。当SSCIOV位通过软件清零时，通过读SSCIBUFR寄存器可以将标志位SSCIBUF清零。

表 16.2 接收数据后的动作

接收到传输数据时的状态位		SSCISR 数据存入 SSCIBUFR	产生 ACK 脉冲	SSCIIF 位置 1 (如果允许 SSCI 中断，还将产生 SSCI 中断)
SSCIBUF	SSCIOV			
0	0	有	有	有
1	0	无	无	有
1	1	无	无	有
0	1	无	无	有

注:阴影单元显示了当用户软件没有正确将溢出状态清零时的情况。

#### 16.3.2.1 寻址

一旦SSCI模块被使能，它就会等待启动条件发生。在7位地址模式下，当启动条件发生后，8位数据被移入SSCISR寄存器。在时钟（SCL）线的上升沿采样所有的输入位。在第8个时钟（SCL）脉冲的下降沿寄存器SSCISR<7:1>的值会和SSCIADD地址寄存器的值比较。如果地址匹配，并且SSCIBUF和SSCIOV都被清零，会发生下列事件：

- 1) SSCISR寄存器的值被装入SSCIBUFR寄存器。
- 2) 缓冲器满标志位SSCIBUF被置1。
- 3) 产生ACK脉冲。
- 4) 在第9个SCL脉冲的下降沿，SSCI中断标志位SSCIIF被置1（如果允许中断，则产生中断）。

在10位地址模式下，从控制器需要收到两个地址字节（图 16.3）。第一个地址字节的高5位将指定这是否是一个10位地址。SSCIRW位（SSCISTA<2>）必须指定写操作，这样从控制器才能接收到第二个地址字节。对于10位地址，第一个字节等于“1111 0 A9 A8 0”，其中A9和A8是该地址的两个最高有效位。

10位地址的工作步骤如下，其中7-9步是针对从动发送器而言的：

- 1) 接收地址的第一个（高）字节（SSCIIF位、SSCIBUF位和SSCIUA位置1）。
- 2) 用地址的第二个（低）字节更新SSCIADD寄存器（SSCIUA位清零并释放SCL线）。
- 3) 读SSCIBUFR寄存器（SSCIBUF位清零），并将标志位SSCIIF清零。
- 4) 接收地址的第二个（低）字节（SSCIIF位、SSCIBUF位和SSCIUA位置1）。
- 5) 用地址的第一个（高）字节更新SSCIADD寄存器；如果匹配，则释放SCL线，此时将会清零SSCIUA位。
- 6) 读SSCIBUFR寄存器（SSCIBUF位清零）并将标志位SSCIIF清零。
- 7) 接收重复启动条件。
- 8) 接收地址的第一个（高）字节（SSCIIF位和SSCIBUF位置1）。
- 9) 读SSCIBUFR寄存器（SSCIBUF位清零）并将标志位SSCIIF清零。

### 16.3.2.2 接收

当地址字节的SSCIRW状态位清零并发生地址匹配时，SSCISTA寄存器中的SSCIRW位清零。接收到的地址被装入SSCIBUFR寄存器。

当发生地址字节溢出时，则不会产生应答脉冲（ACK）。溢出条件是指SSCIBUF位置1，或者SSCIOV位（SSCICTL0<6>）置1。这是一个由于用户固件导致的错误状态。

每个数据传输字节都会产生SSCI中断。标志位SSCIIF必须用软件清零。通过SSCISTA寄存器可以确定该字节的状态。

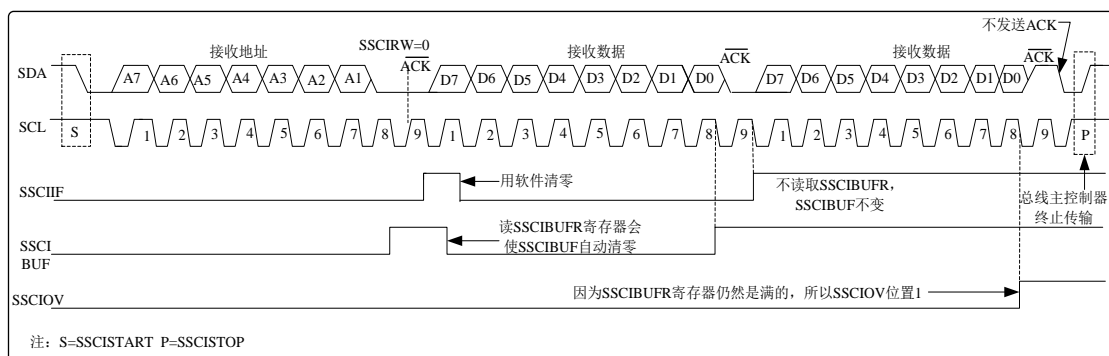


图 16.2 从动模式时序（接收，7 位地址）

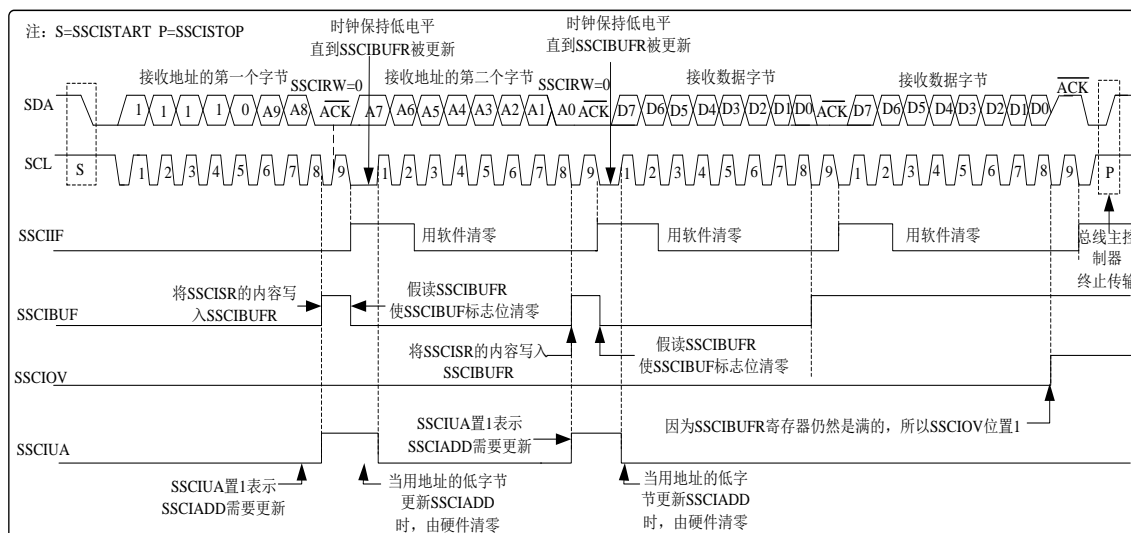


图 16.3 从动模式时序（接收，10 位地址）

## 从动接收设置:

- 通过 SSCIMOD<3:0>位选择 I2C 工作模式
  - 0110 = I2C 从动模式，7 位地址
  - 111 = I2C 从动模式，10 位地址
  - 1011 = I2C 固件控制主控模式（从动空闲模式）
  - 1110 = I2C 从动模式，7 位地址，并允许启动位和停止位中断
  - 1111 = I2C 从动模式，10 位地址，并允许启动位和停止位中断
- 设置SSCIADD寄存器，设置从机地址，仅高七位有效；
- 清零SSCISTA寄存器的各标志，包括SSCIDA、SSCIRW、SSCIBUF等。
- 设置SDA引脚为输入，SCL为输入；
- 清零SSCIIF标志，如果需要中断打开各终端使能位；
- 使能SSCIEN，开始接收数据，等待地址匹配；如果地址匹配，则SSCISTA寄存器的SSCIRW位清零。SSCISR寄存器的值被装入SSCIBUFR寄存器；
- 缓冲器满标志位SSCIBUF被置1；产生ACK脉冲信号；在第9个SCL脉冲的下降沿，SSCI中断标志位SSCIIF被置1，软件清零。

## 16.3.2.3 发送

当输入地址字节的SSCIRW位置1 并发生地址匹配时，SSCISTA寄存器的SSCIRW位被置1。接收到的地址被装入SSCIBUFR寄存器。ACK脉冲在第9位上发送，SCL引脚保持低电平。发送数据必须被装入SSCIBUFR寄存器，同时也装入SSCISR寄存器。然后，应该通过将SSCICKP位（SSCCTL0<4>）置1来使能SCL引脚。主控制器必须在发出另一个时钟脉冲前监视SCL引脚。从控制器可以通过延长时钟低电平时间不与主控制器同步。8个数据位在SCL输入的下沿被移出。这可以确保在SCL为高电平期间SDA信号是有效的。

每个数据传输字节都会产生SSCI中断。标志位SSCIIF必须用软件清零，SSCISTA寄存器用于确定字节的状态。标志位SSCIIF在第9个时钟脉冲的下沿被置1。对于从发送器，来自自主接收器的ACK脉冲将在第9个SCL输入脉冲的上升沿被锁存。若SDA线为高电平（无ACK应答信号），则表示数据传输已完成。在这种情况下，如果从控制器锁存了ACK，将复位从动逻辑（复位SSCISTA寄存器），同时从控制器监视下一个启动位的出现。如果SDA线为低电平（ACK），则必须将下一个要发送的数据装入SSCIBUFR寄存器。然后，通过将SSCICKP位（SSCCTL0<4>）置1使能SCL引脚。

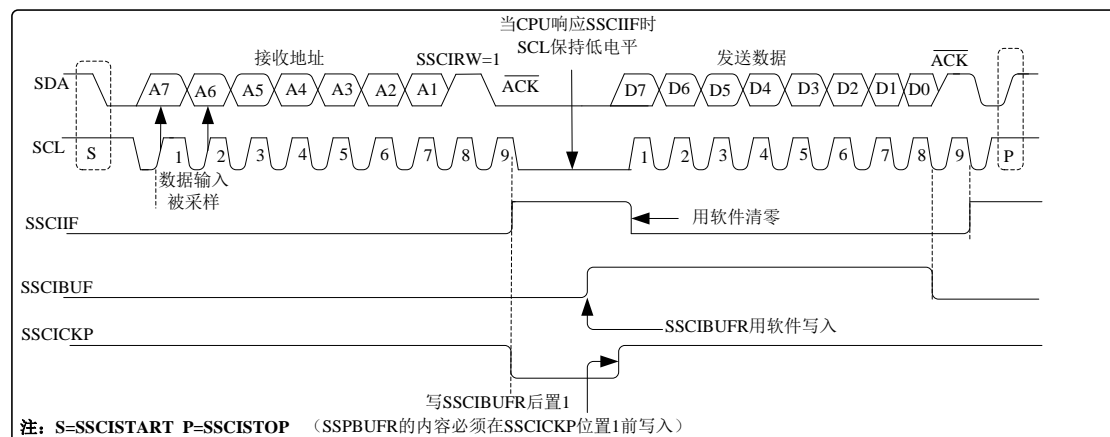


图 16.4 从动模式时序（发送，7 位地址）

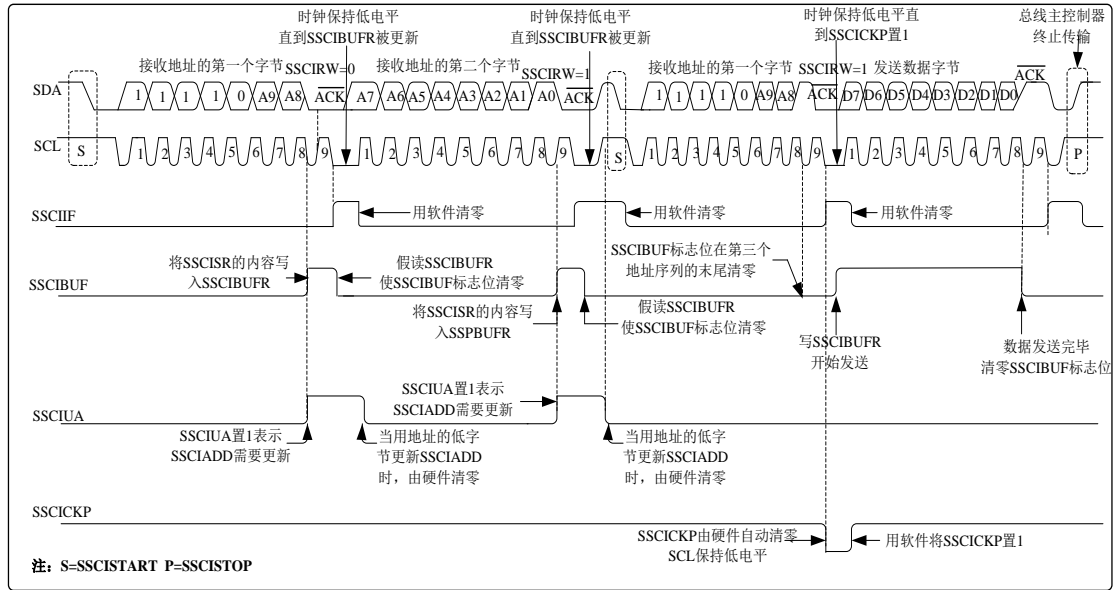


图 16.5 从动模式时序（发送，10 位地址）

从动发送设置:

- 1、 通过 SSCIMOD<3:0>位选择 I2C 工作模式
  - 0110 = I2C 从动模式， 7 位地址
  - 0111 = I2C 从动模式， 10 位地址
  - 1011 = I2C 固件控制主控模式（从动空闲模式）
  - 1110 = I2C 从动模式， 7 位地址， 并允许启动位和停止位中断
  - 1111 = I2C 从动模式， 10 位地址， 并允许启动位和停止位中断
- 2、 设置SSCIADD寄存器， 设置从机地址， 仅高七位有效；
- 3、 清零SSCISTA寄存器的各标志， 包括SSCIDA、 SSCIRW、 SSCIBUF等。
- 4、 设置SDA引脚为输出， SCL为输入；
- 5、 清零SSCIIF标志， 如果需要中断打开各终端使能位；
- 6、 使能 SSCIEN， 当输入地址字节的 SSCIRW 位置 1 并发生地址匹配时， SSCISTA 寄存器的 SSCIRW 位被置 1。接收到的地址被装入 SSCIBUFR 寄存器。
- 7、 ACK 脉冲在第 9 位上发送， SCL 引脚保持低电平。发送的数据装载到 SSCIBUFR 寄存器。
- 8、 将 SSCICKP 位置“1”释放 SCL 引脚， 使能时钟。主控制器必须在发送另一个时钟脉冲前监视 SCL 引脚。从控制器可以通过延长时钟低电平时间不予主控制器同步。
- 9、 标志位 SSCIIF 在第 9 个时钟脉冲的下降沿被置 1。软件清零
- 10、 对于从发送器， 来自主接收器的ACK脉冲将在第9个SCL输入脉冲的上升沿被锁存。若SDA线为高电平（无ACK应答信号）， 则表示数据传输已完成。在这种情况下， 如果从控制器锁存了ACK， 将复位从动逻辑（复位SSCISTA寄存器）， 同时从控制器监视下一个启动位的出现。如果SDA 线为低电平（ACK）， 则必须将下一个要发送的数据装入 SSCIBUFR寄存器。 然后， 通过将SSCICKP位（SSCICTL0<4>）置1使能SCL引脚。

### 16.3.2.4 广播呼叫地址支持

在I2C 总线的寻址过程中,通常由启动条件后的第一个字节决定主器件将寻址哪个从器件。但广播呼叫地址例外,它能寻址所有器件。当使用这个地址时,理论上所有的器件都应该发送一个应答响应。

广播呼叫地址是根据I2C协议为特定目的保留的八个地址之一。它由全0 组成,且SSCIRW = 0。广播呼叫使能位SSCICALLEN (SSCICL2<7>寄存器使能时,即可识别广播呼叫地址。检测到起始位后,8 位数据会移入SSCISR,同时将该地址与SSCIADD 进行比较。它还会与广播呼叫地址进行比较并用硬件设定。

如果与广播呼叫地址匹配,SSCISR的值将传输到SSCIBUFR,SSCIBUF标志位(第8位)置1,并且SSCIIF中断标志位在第9位(ACK位)的下降沿置1。

当响应中断时,可以通过读取SSCIBUFR的内容来判断中断源。该值可以用于判断地址是特定器件的还是一个广播呼叫地址。

在10位模式下,需要更新SSCIADD以使地址的后半部分匹配,同时SSCIUA位(SSCISTA寄存器)置1。如果SSCICALLEN位置1时采样到广播呼叫地址,同时从器件被配置为10位地址模式,则不再需要地址的后半部分,也不会将SSCIUA位置1,从器件将在应答后开始接收数据如下图16.6所示。

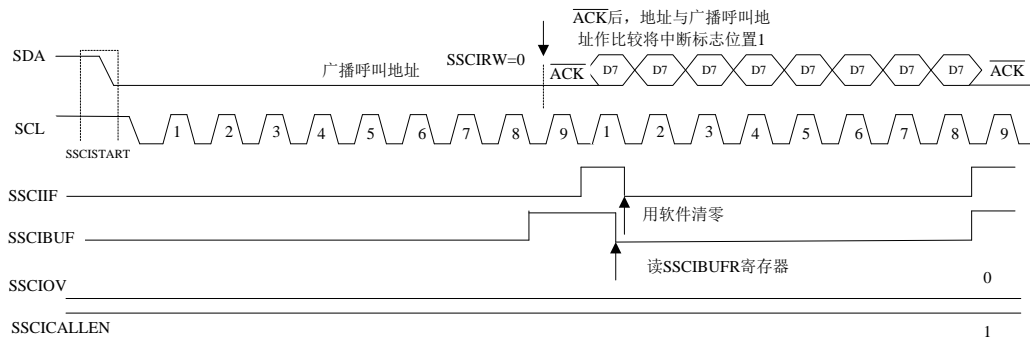


图 16.6 从动模式广播呼叫地址时序(7或10位地址模式)

### 16.3.3 I2C 主控模式

主控模式通过固件在检测到启动条件和停止条件时产生中断来工作。停止(SSCISTOP)位和启动(SSCISTART)位在复位时或禁止SSCI模块时清零。停止(SSCISTOP)位和启动(SSCISTART)位会根据启动和停止条件翻转。当SSCISTOP位置1时,可以获得I2C总线的控制权;否则,停止(SSCISTOP)位和启动(SSCISTART)位都清零,总线处于空闲状态。

在主动模式下,SCL和SDA线不需要清零相应的TRxx位,硬件自动控制,用户只需将相应的ANSxx位清零,使得主机能监控SCL和SDA引脚上的电平。同时SCL和SDA引脚上必须外接上拉电阻,才能使I2C模块正常工作。

下列事件会使SSCI中断标志位SSCIIF置1(如果允许SSCI中断,则产生中断):

启动条件

- 停止条件
- 发送/接收到数据传输字节
- 应答发送

- 重复启动条件

可用从动模式空闲 (SSCIMOD<3:0> = 1011) 或从动模式活动完成主控模式操作。当同时使能主控模式和从动模式时, 需要使用软件区分中断源。

### 16.3.3.1 主控模式支持

通过设置SSCICTL0中的SSCIMOD<3:0>并将SSCIEN位置1可使能主控模式。一旦使能主控模式,

用户即可选择以下6项操作:

- 1) 在SDA 和SCL 上发出一个启动条件。
- 2) 在SDA 和SCL 上发出一个重复启动条件。
- 3) 写入SSCIBUFR寄存器, 开始数据/地址的发送。
- 4) 在SDA 和SCL 上产生停止条件
- 5) 将I2C 端口配置为接收数据。
- 6) 在接收到数据字节后产生应答条件。

注: 当配置为I2C主控模式时, SSCI模块不允许事件排队。例如, 在启动条件结束前, 不允许用户发出另一个启动条件并立即写SSCIBUFR寄存器以发起传输。这种情况下, 将不会写入SSCIBUFR, SSCIWCFL 位将被置1, 这表明没有发生对SSCIBUFR的写操作。图 16.7 为I2C主模式框图。

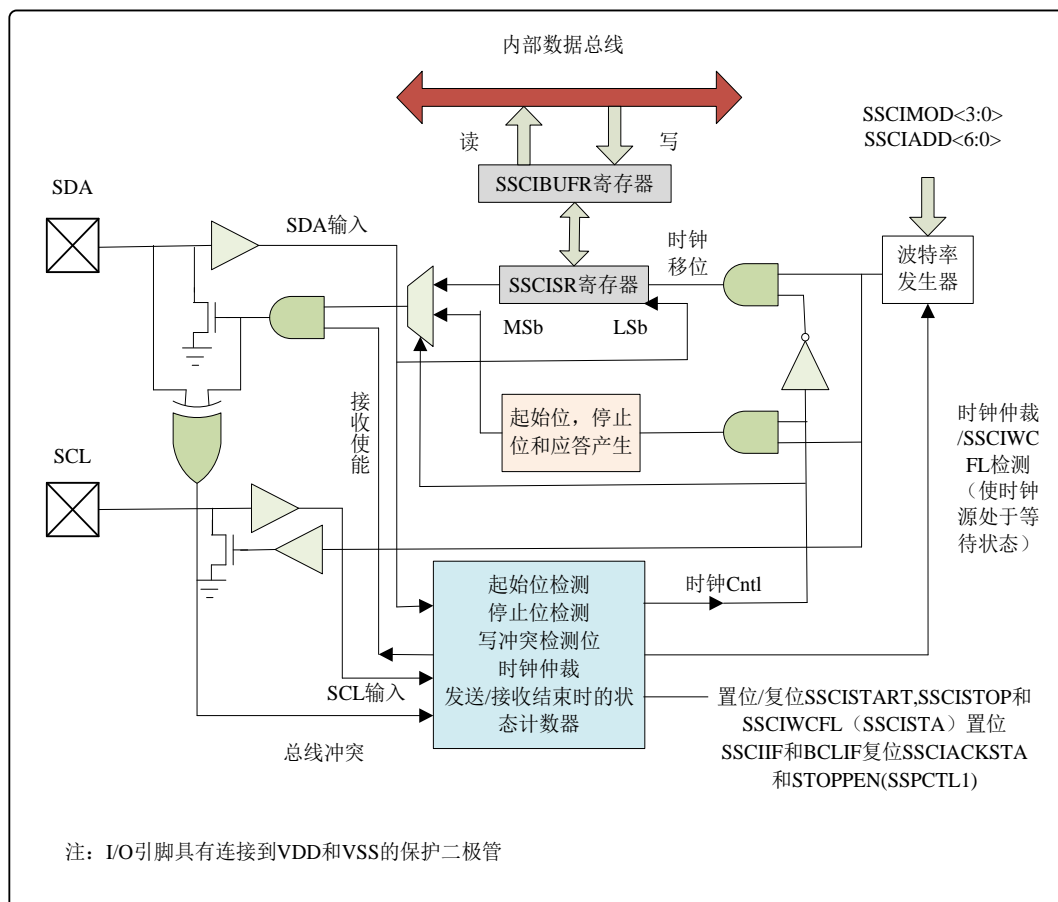


图 16.7 I2C 主模式框图

### 16.3.3.2 I2C 主模式操作

所有串行时钟脉冲和启动/ 停止条件均由主器件产生。停止条件或重复启动条件能结束传输。因为重复启动条件也是下一次串行传输的开始，因此不会释放I2C总线。在主控发送器模式下，串行数据通过SDA 输出，而串行时钟由SCL输出。发送的第一个字节包括接收器件的地址（7 位）和读/ 写（SSCIRW）位。在这种情况下，SSCIRW位将是逻辑0。串行数据每次发送8 位。每发送一个字节，会收到一个应答位。启动和停止条件的输出表明串行传输的开始和结束。

在主控接收模式下，发送的第一个字节包括发送器件的地址（7 位）和SSCIRW位。在这种情况下，SSCIRW位将是逻辑1。因此，发送的第一个字节是一个7 位从器件地址，后面跟1 表示接收。串行数据通过SDA 接收，而串行时钟由SCL 输出。每次接收8 位串行数据。每接收到一个字节，都会发送一个应答位。启动和停止条件分别表明发送的开始和结束。

在I2C模式下，在SPI 模式中使用的波特率发生器被用于将SCL时钟频率设置为100 kHz、400 kHz或1 MHz。波特率发生器的重载值位于SSCIADD寄存器的低7 位。当发生对SSCIBUFR的写操作时，波特率发生器将自动开始计数。如果指定操作完成（即，发送的最后一个数据位后面跟着ACK），内部时钟将自动停止计数，SCL 引脚将保持在其最后的状态。

下面是一个典型的发送事件序列：

- 1) 用户通过将启动使能位STARTEN（SSCICTL2寄存器）置1 产生启动条件。
- 2) SSCIF 位置1。在进行任何其他操作前，SSCI模块将等待所需的启动时间。
- 3) 用户将从器件地址装入SSCIBUFR进行发送。
- 4) 地址从SDA 引脚移出，直到发送完所有8 位为止。
- 5) SSCI模块移入来自从器件的ACK位，并将它的值写入SSCICTL2 寄存器的SSCIACKSTA位。
- 6) SSCI模块在第9 个时钟周期的末尾将SSCIF位置1，产生一个中断。
- 7) 用户将8 位数据装入SSCIBUFR。
- 8) 数据从SDA 引脚移出，直到发送完所有8 位为止。
- 9) SSCI模块移入来自从器件的ACK位，并将它的值写入SSCICTL2 寄存器的SSCIACKSTA 位。
- 10) SSCI 模块在第9 个时钟的末尾将SSCIF 位置1，产生一个中断。
- 11) 用户通过将停止使能位（STOPEN）位（SSCICTL2寄存器）置1产生停止。
- 12) 一旦停止条件完成，将产生一个中断。

### 16.3.3.3 波特率发生器

在I2C 主控模式下，波特率发生器的重载值位于SSCIADD 寄存器的低7 位。当装载了该值后，波特率发生器将自动开始计数并递减至0，然后停止直到下次重载为止。BRG 会在每个指令周期（TCY）中的Q2 和Q4 时钟周期上进行两次减计数。在I2C 主控模式下，会自动重载BRG。例如，在发生时钟仲裁时，BRG 将在SCL 引脚采样到高电平时重载。如图16.8 和图16.9 所示。



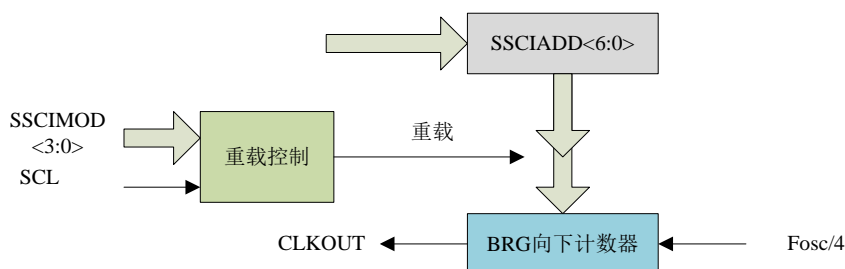


图 16.8 波特率发生器框图

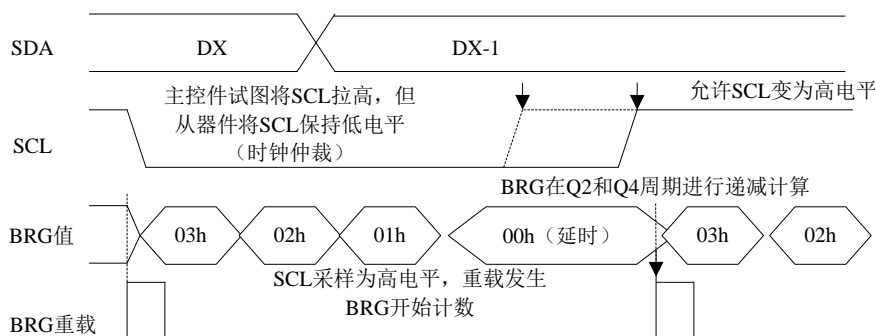


图 16.9 带有时钟仲裁的波特率发生器时序

### 16.3.3.4 I2C 主控模式启动条件时序

要发起启动条件，用户应将SSCICTL2寄存器的启动条件使能位STARTEN置1。当SDA和SCL引脚都采样为高电平时，波特率发生器重新装入SSCIADD<6:0>的内容并开始计数。当波特率发生器发生超时（TBRG）时，如果SCL和SDA都采样为高电平，则SDA引脚被驱动为低电平。当SCL为高电平时，将SDA驱动为低电平就是启动条件，将使SSCISTART位（SSCISTA寄存器）置1。随后波特率发生器重新装入SSCIADD<6:0>的内容并恢复计数。当波特率发生器超时（TBRG）时，SSCICTL2寄存器的STARTEN位将自动被硬件清零。波特率发生器暂停工作，SDA线保持低电平，启动条件结束。

注意：

如果在启动条件开始时，SDA和SCL引脚已经采样为低电平，或者在启动条件期间，SCL在SDA线被驱动为低电平之前已经采样为低电平，则会发生总线冲突。总线冲突中断标志位BCLIF置1，启动条件中止，I2C模块复位到空闲状态。

#### SSCIWCFL状态标志

当启动序列进行时，如果用户写SSCIBUFR，则SSCIWCFL被置1，同时缓冲器内容不变（未发生写操作）。

注：

由于不允许事件排队，在启动条件结束之前，不能对SSCICTL2的低5位进行写操作。

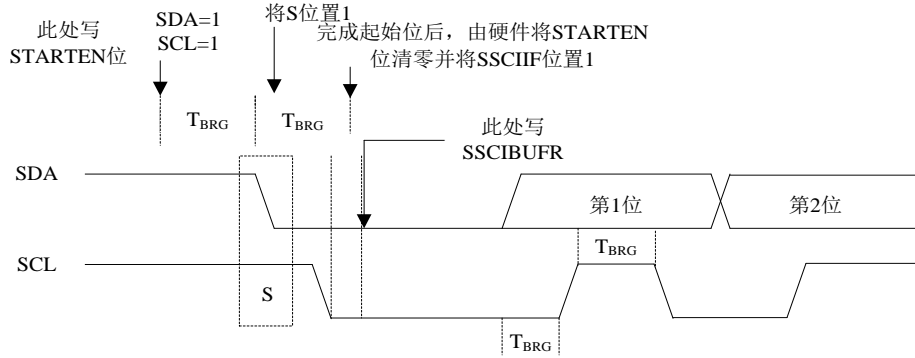


图 16.10 第一个启动位时序

### 16.3.3.5 I2C 主控模式重复启动条件时序

将RESTARTEN位（SSCICTL2寄存器）编程为高电平，并且I2C逻辑模块处于空闲状态时，就会产生重复启动条件。当RESTARTEN位置1时，SCL引脚被拉为低电平。当SCL引脚采样为低电平时，波特率发生器装入SSCIADD<6:0>的内容，并开始计数。在一个波特率发生器计数周期（TBRG）内SDA引脚被释放（其引脚电平被拉高）。当波特率发生器超时，如果SDA 采样为高电平，SCL引脚将被拉高。当SCL引脚采样为高电平时，波特率发生器将被重新装入SSCIADD<6:0>的内容并开始计数。SDA和SCL必须在一个计数周期TBRG内采样为高电平。随后将SDA引脚拉为低电平（SDA=0）并保持一个计数周期TBRG，同时SCL为高电平。然后RESTARTEN位（SSCICTL2 寄存器）将自动清零，波特率发生器不会重载，SDA引脚保持低电平。一旦在SDA和SCL引脚上检测到启动条件，SSCISTART位（SSCISTA寄存器）将被置1。直到波特率发生器超时后，SSCIIF位才会置1。

注:

- 1) 有任何其他事件进行时，对RESTARTEN的编程无效。
- 2) 在重复启动条件期间，下列事件将会导致总线冲突:
  - 当SCL 由低电平变为高电平时，SDA 采样为低电平。
  - 在SDA 被拉低之前， SCL 变为低电平。这表示可能有另一个主器件正尝试发送数据1。

一旦SSCIIF 位被置1，用户便可以在7 位地址模式下将7 位地址写入SSCIBUFR，或者在10 位地址模式下写入默认的第二个地址字节。当发送完第一个8 位并接收到一个ACK 后，用户可以发送另外8 位地址（10 位地址模式下）或8 位数据（7 位地址模式下）。

#### SSCIWCFL 状态标志

当重复启动序列进行时，如果用户写SSCIBUFR，则SSCIWCFL 被置1，同时缓冲器内容不变（未发生写操作）。

注:由于不允许事件排队，在重复启动条件结束之前，不能对SSCICTL2 的低5位进行写操作。

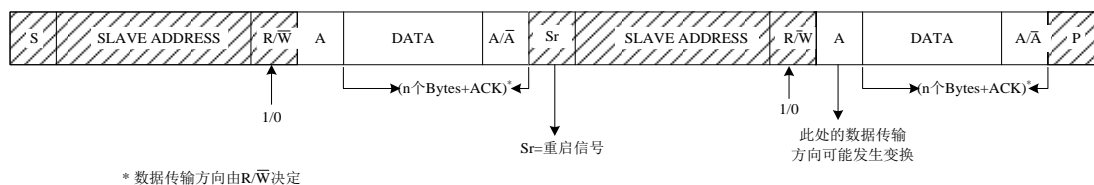


图 16.11 I2C 协议复合数据帧格式

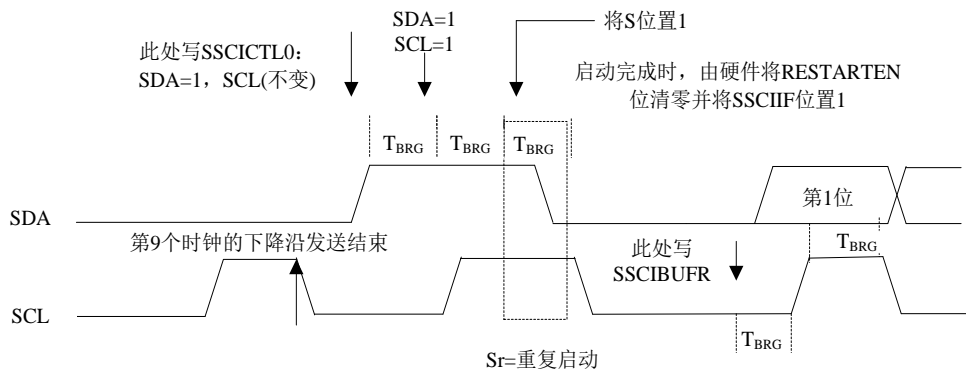


图 16.12 重复启动条件时序波形

### 16.3.3.6 I2C 主控模式发送

I2C 协议中 SDA 引脚上的数据不仅来源于 SSCIBUFR 寄存器，还需要按 I2C 协议的规定发送起始位、停止位、应答位等信号。图 16.13 所示是 I2C 协议典型写数据帧格式。

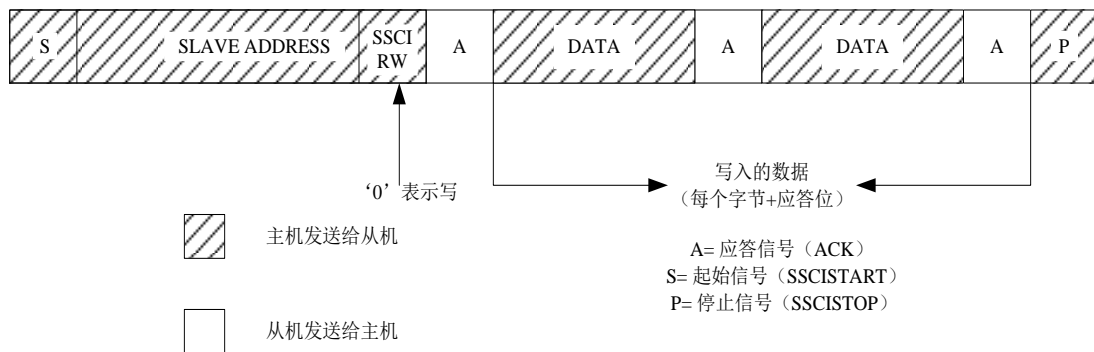


图 16.13 I2C 协议典型写数据帧格式

发送一个数据字节、一 7 位地址或一 10 位地址的另一半，都可以直接通过写一个值到 SSCIBUFR 寄存器来实现。该操作将使缓冲器满标志位 SSCIBUF 置 1，并且波特率发生器开始计数，同时启动下一次发送。

在 SCL 的下降沿有效后，地址/数据的每一位将被移出至 SDA 引脚。在一个波特率发生器计满返回计数周期 (TBRG) 内，SCL 保持低电平。数据应该在 SCL 释放为高电平前保持有效。当 SCL 引脚被释放为高电平时，它将在整个 TBRG 中保持高电平状态。在此期间以及下一个 SCL 下降沿之后的一段时间内，SDA 引脚上的数据必须保持稳定。在第 8 位被移出 (第 8 个时钟周期的下降沿) 之后，SSCIBUF 标志位清零，同时主器件释放 SDA。此时如果发生地址匹配或是数据被正确接收，被寻址的从器件将在第 9 位的时间以一个 ACK 位响应。ACK 的状态在第 9 个时钟周期的下降沿写入 SSCIACKDAT 位。主器件接收到应答之后，应答状态位 SSCIACKSTA 会被清零；如果未收到应答，则该位被置 1。第 9 个时钟之后，SSCIIF 位会置 1，主控时钟 (波特率发生器) 暂停，直到下一个数据字节装入 SSCIBUFR 为止，SCL 引脚保持低电平，SDA 保持不变。

在写 SSCIBUFR 之后，地址的每一位在 SCL 的下降沿被移出，直至地址的所有 7 位和

SSCIRW 位都被移出为止。在第 8 个时钟的下降沿，主器件将 SDA 引脚拉为高电平以允许从器件发出应答响应。在第 9 个时钟的下降沿，主器件通过采样 SDA 引脚来判断地址是否被从器件识别。ACK 位的状态被装入 SSCIACKSTA 状态位 (SSCICTL2 寄存器)。在发送地址的第 9 个时钟下降沿之后，SSCIIF 置 1，SSCIBUF 标志位清零，波特率发生器关闭直到下一次写 SSCIBUFR，且 SCL 引脚保持低电平，允许 SDA 引脚悬空。

**SSCIBUF 状态标志**

在发送模式下，SSCIBUF 位 (SSCISTA 寄存器) 在 CPU 写 SSCIBUFR 时置 1，在所有 8 位数据移出后清零。

**SSCIWCFL 状态标志位**

如果用户在发送过程中 (即，SSCISR 仍在移出数据字节时) 写 SSCIBUFR，则 SSCIWCFL 置 1 且缓冲器的内容保持不变 (未发生写操作) SSCIWCFL 必须由软件清零。

**SSCIACKSTA 状态标志**

在发送模式下，当从器件发送应答响应 (ACK = 0) 时，SSCIACKSTA 位 (SSCICTL2 寄存器) 清零；当从器件没有应答 (ACK = 1) 时，该位置 1。从器件在识别出其地址 (包括广播呼叫地址) 或正确接收数据后，会发送一个应答。

注:若主机发送完一个字节后收到的应答标志位 SSCIACKSTA=1，则应及时停止传输 (通过发送停止信号 STOPEN 位来实现)。

图 16.14 所示为 I2C 协议的典型写数据帧格式在单片机硬件 I2C 上的实现过程时序图。

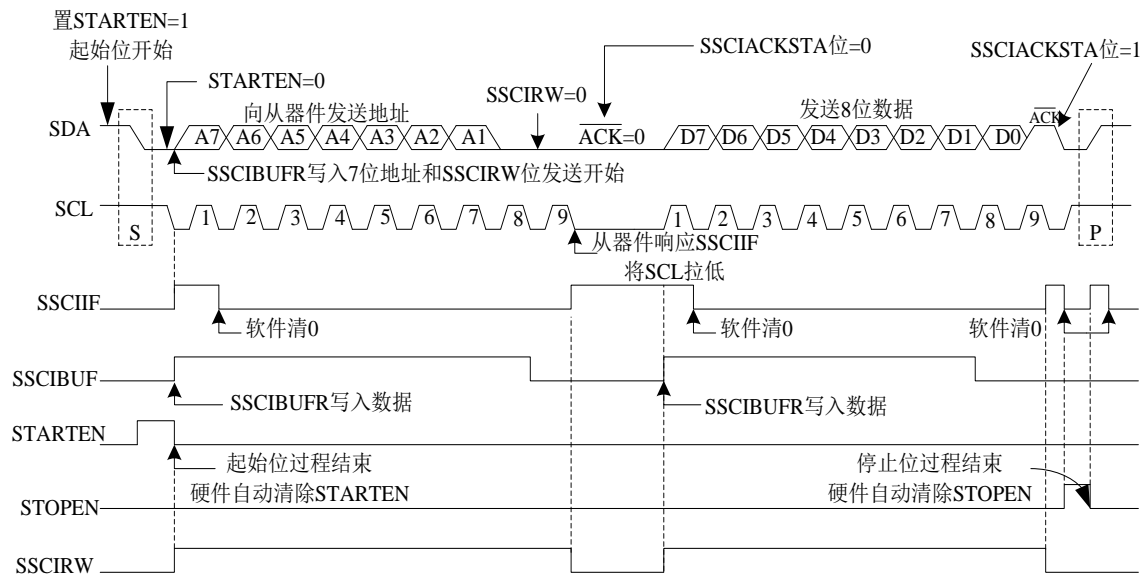
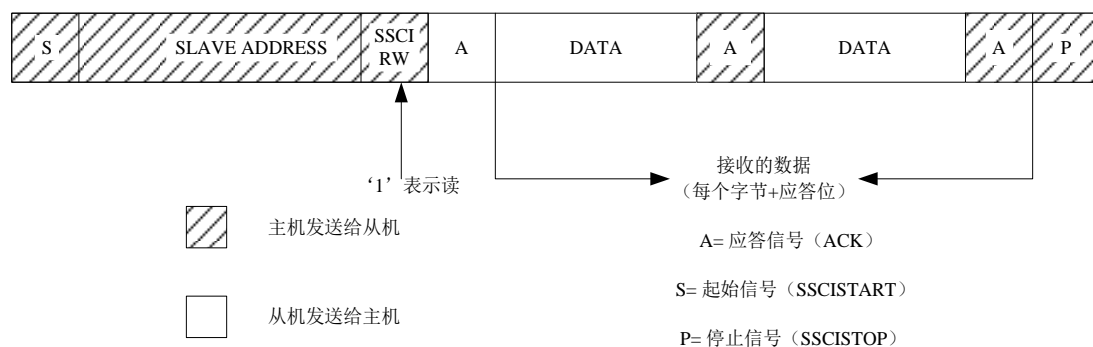


图 16.14 硬件 I2C 主模式发送数据时序图 (7 位地址)

**16.3.3.7 I2C 主控模式接收**

I2C 主模式数据接受的系统结构与主模式数据发送系统结构相同，但是数据接收流程与数据发送流程不同。如图 16.15 为 I2C 协议典型读数据帧格式。



通过编程接收使能位SSCIRCEN（SSCCTL2寄存器）使能主控模式接收。

**注：**SSCIRCEN位被置1前，SSCI模块必须处于空闲状态，否则SSCIRCEN位将被忽略。

波特率发生器开始计数，每次计满返回时，SCL引脚的状态都发生改变（由高变低或由低变高），且数据被移入SSCISR。第8个时钟的下降沿之后，接收使能标志位自动清零，SSCISR的内容装入SSCIBUFR，SSCIBUF标志位置1，SSCIIF标志位置1，波特率发生器暂停计数，SCL保持为低电平。此时SSCI处于空闲状态，等待下一条命令。当CPU读缓冲器时，SSCIBUF标志位将自动清零。通过将应答序列使能位SSCIACKEN（SSCCTL2寄存器）置1，用户可以在接收结束后发送应答位。

#### SSCIBUF 状态标志

接收时，当将地址或数据字节从SSCISR装入SSCIBUFR时，SSCIBUF位置1；在读SSCIBUFR寄存器时SSCIBUF位清零

注:当读操作完成时若SSCIBUF还是1（说明SSCIBUFR上次读到的数据未被读走），会使SSCIIF接收溢出信号SSCIOV自动置位。SSCIOV必须软件清零。

#### SSCIOV 状态标志

接收时，当SSCISR 接收到8位数据时，SSCIOV位置1，SSCIBUF标志位已经在上一次接收时置1。

#### SSCIWCFL 状态标志

如果用户在接收过程中（即，SSCISR仍在移入数据字节时）写SSCIBUFR，则SSCIWCFL位置1，缓冲器内容不变（未发生写操作）

图 16.16为典型读数据帧格式在单片机硬件I2C上的实现接收过程时序图。

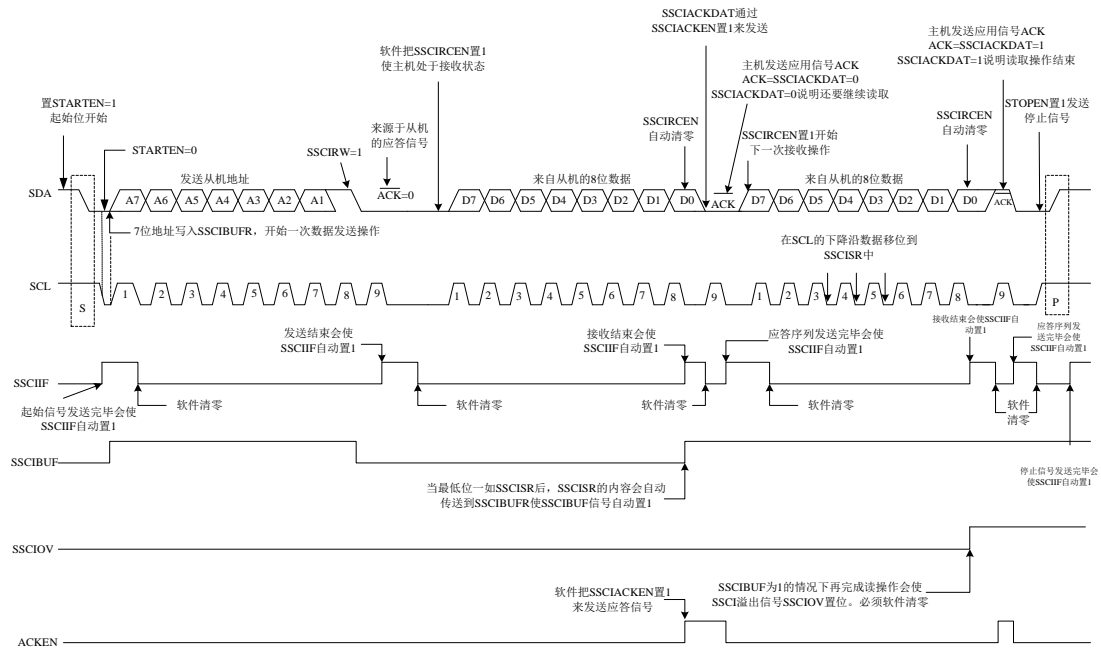


图 16.16 I2C 主模式接收数据时序图（7 位地址）

### 16.3.3.8 应答序列时序

将应答序列使能位SSCIACKEN（SSCICTL2寄存器）置1即可使能应答序列。当该位被置1时，SCL引脚被拉低，应答数据位的内容出现在SDA引脚上。如果用户希望产生一个应答，则应该将SSCIACKDAT位清零；否则，用户应该在应答序列开始前将SSCIACKDAT位置1。然后波特率发生器进行一个计满返回周期（TBRG）的计数，随后SCL引脚电平被拉高。当SCL引脚采样为高电平时（时钟仲裁），波特率发生器再进行一个TBRG周期的计数。然后SCL引脚被拉低。在这之后，SSCIACKEN位自动清零，波特率发生器关闭，SSCI模块进入空闲模式。

#### SSCIWCFL 状态标志位

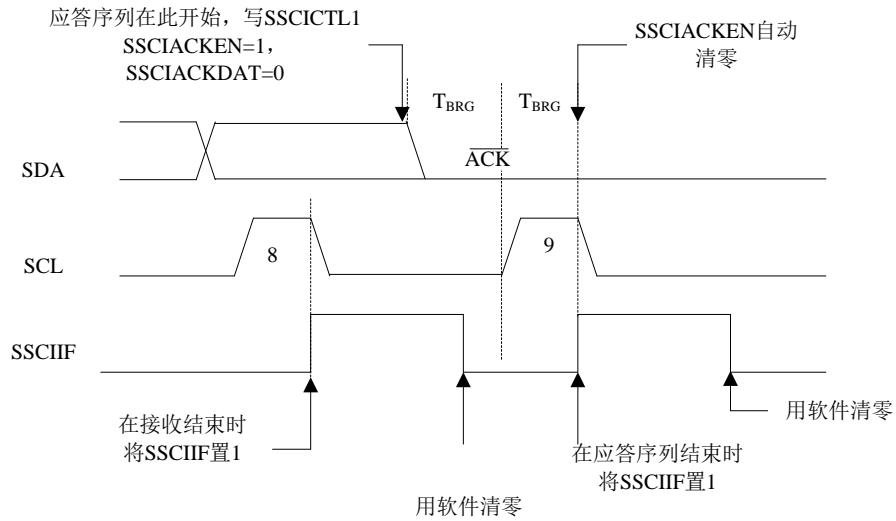
如果用户在应答序列正在进行时写SSCIBUFR，SSCIWCFL 将被置1且缓冲器的内容保持不变（未发生写操作）。

### 16.3.3.9 停止条件序列

在接收/发送结束时，通过置停止序列的使能位，STOPEN（SSCICTL2寄存器），SDA引脚将产生一个停止位。在接收/发送结束时，SCL引脚在第9个时钟的下降沿后保持低电平。当STOPEN位置1时，主控制器将SDA置为低电平。当SDA线采样为低电平时，波特率发生器被重新装入值并递减计数至0。波特率发生器发生超时时，SCL引脚被拉到高电平，且一个TBRG（波特率发生器计满回零）后，SDA引脚被重新拉到高电平。当SDA引脚采样为高电平且SCL也是高电平时，SSCISTOP位（SSCISTA寄存器）置1。一个TBRG周期后，STOPEN位清零且SSCIIF位置1。

#### SSCIWCFL 状态标志

如果用户在停止序列进行过程中试图写SSCIBUFR，则SSCIWCFL 位将置1，缓冲器的内容不会改变（未发生写操作）。



注： $T_{BRG}$  = 一个波特率发生器周期

图 16.17 应答序列时序波形

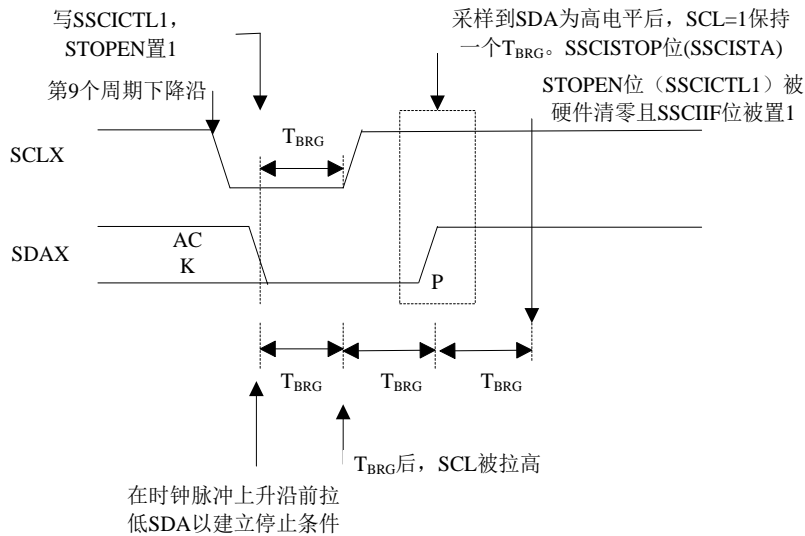


图 16.18 停止条件接收或发送模式

### 16.3.3.10 时钟仲裁

如果在任何接收、发送或重复启动/ 停止条件期间，主器件拉高了 SCL 引脚（允许 SCL 引脚悬空为高电平），就会发生时钟仲裁。如果允许 SCL 引脚悬空为高电平，波特率发生器（BRG）将暂停计数，直到实际采样到 SCL 引脚为高电平为止。当 SCL 引脚采样为高电平时，波特率发生器中将被重新装入 SSCIADD<6:0> 的内容并开始计数。这可以保证当外部器件将时钟拉低时，SCL 始终保持至少一个 BRG 计满返回周期的高电平。

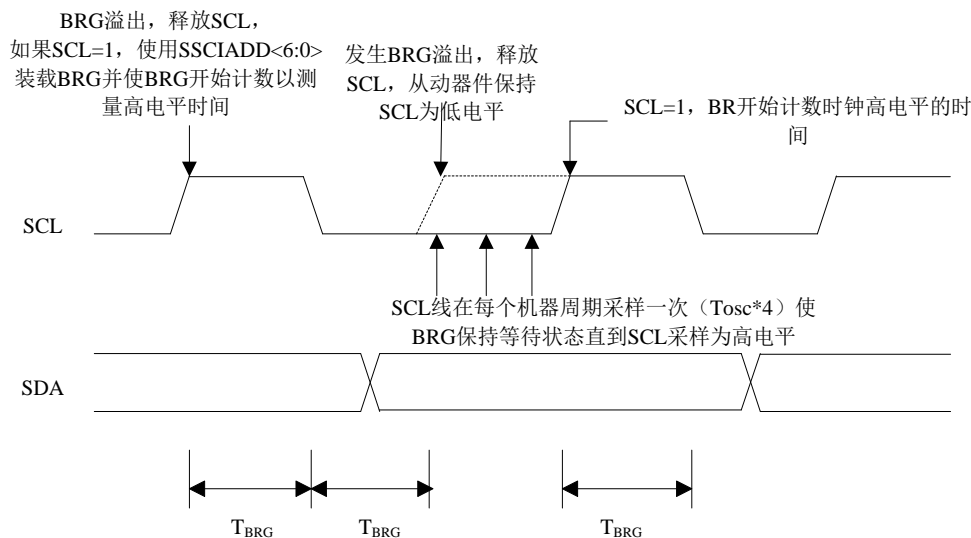


图 16.19 主控发送模式下的时钟仲裁时序

### 16.3.4 多主控器模式

在多主控制器模式下，在检测到启动条件和停止条件时产生的中断可用于判断总线是否空闲。停止（SSCISTOP）位和启动（SSCISTART）位在复位时或禁止SSCI模块时被清零。停止（SSCISTOP）位和启动（SSCISTART）位会根据启动和停止条件翻转。当SSCISTOP位（SSCISTA<4>）置1时，可以获得I2C总线的控制权；否则，SSCISTOP位和SSCISTART位都清零，总线处于空闲状态。当总线处于忙状态且允许SSCI中断时，一旦发生停止条件便产生中断。

在多主控制器操作中，必须监视SDA线以确定信号电平是否为所需的输出电平。此检查仅需在输出为高电平时进行。如果期望输出高电平，但检测到的是低电平，器件就需要释放SDA和SCL线（TR<sub>xx</sub> 位置1）。此仲裁在以下状态可能会失败：

- 地址传输
- 数据传输
- 启动条件
- 重复启动条件
- 应答条件

当使能从动逻辑电路时，从控制器将继续接收数据。如果在地址传输阶段仲裁失败，可能表示与器件的通信正在进行中。如果寻址到器件，则将会产生一个ACK脉冲。如果在数据传输阶段仲裁失败，则器件需要在以后重新传输数据。

#### 16.3.4.1 多主机通信，总线冲突与总线仲裁

多主机模式是通过总线仲裁来支持的。

当主器件将地址/数据位输出到SDA引脚时，如果一个主器件通过将SDA引脚悬空为高电平以在SDA上输出1，而另一个主器件输出0，就会发生总线仲裁。



如果 SDA 引脚上期望的数据是 1，而实际在 SDA 引脚上采样到的数据是 0，则发生了总线冲突。主器件将把总线冲突中断标志位 BCLIF 置 1，并将 I2C 端口复位到空闲状态。如果在发送过程中发生总线冲突，则发送停止，SSCIBUF 标志位清零，SDA 和 SCL 线被拉高，并且允许对 SSCIBUFR 进行写操作。当执行完总线冲突中断服务程序后，如果 I2C 总线空闲，用户可通过发出启动条件恢复通信。

如果在启动、重复启动、停止或应答条件的进行过程中发生总线冲突，则该条件被中止，SDA 和 SCL 线被拉高，SSCICTL2 寄存器中的对应控制位清零。当执行完总线冲突中断服务程序后，如果 I2C 总线空闲，用户可通过发出启动条件恢复通信。主器件将继续监视 SDA 和 SCL 引脚。如果出现停止条件，SSCIIF 位将被置 1。无论发生总线冲突时发送的进度如何，写 SSCIBUFR 都会从第一个数据位开始发送数据。在多主机模式下，通过在检测到启动和停止条件时产生中断可以确定总线何时空闲。SSCISTOP 位置 1 时，可以获取 I2C 总线的控制权，否则总线空闲且 SSCISTART 和 SSCISTOP 位清零。

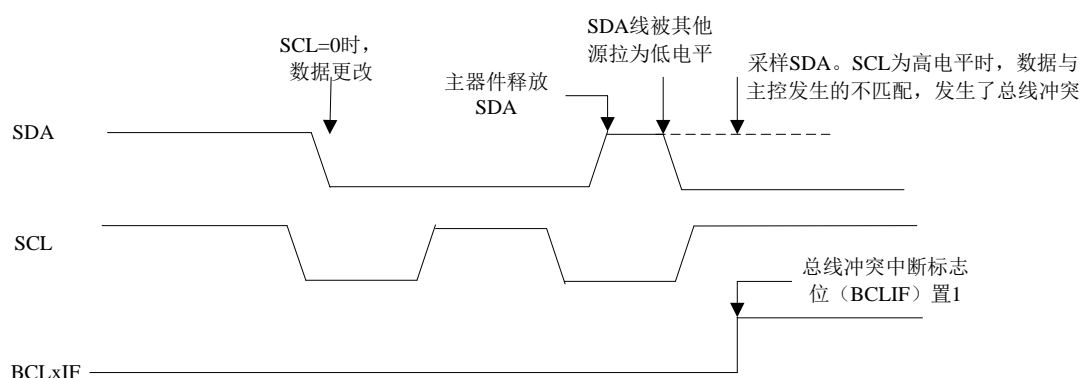


图 16.20 发送和应答时的总线冲突时序

#### 16.3.4.2 启动条件期间的总线冲突

启动条件期间，以下事件将导致总线冲突：

- 1) 在启动条件开始时，SDA 或 SCL 被采样为低电平。
- 2) SDA 被拉低之前，SCL 采样为低电平。

在启动条件期间，SDA 和 SCL 引脚都会被监视。如果 SDA 引脚已经是低电平，或 SCL 引脚已经是低电平，则：

- 中止启动条件，
- BCLIF 标志位置 1，
- 并将 SSCI 模块复位为空闲状态。

启动条件从 SDA 和 SCL 引脚被拉高开始。当 SDA 引脚采样为高电平时，波特率发生器装入 SSCIADD<6:0>的值并递减计数到 0。如果在 SDA 为高电平时，SCL 引脚采样为低电平，则发生总线冲突，因为这表示另一个主器件在启动条件期间试图发送一个数据 1。

如果 SDA 引脚在该计数周期内采样为低电平，则 BRG 复位，同时 SDA 线保持原值。但是，如果 SDA 引脚采样为 1。如果 SDA 引脚将在 BRG 计数结束时被置为低电平。随后波特率发生器被重新装入值并递减计数至 0。在此期间，如果 SCL 引脚采样到 0，则没有发生总线冲突。在 BRG 计数结束时，SCL 引脚被拉为低电平。

注:

在启动条件期间不会发生总线冲突是因为两个总线主器件不可能精确地在同一时刻发出启动条件。因此总是有一个主器件先于另一个主器件将SDA拉低。但是这一情况不会引起总线冲突，因为允许两个主器件对启动条件后的第一个地址进行仲裁。如果地址是相同的，将继续对数据部分、重复启动条件或停止条件进行仲裁。

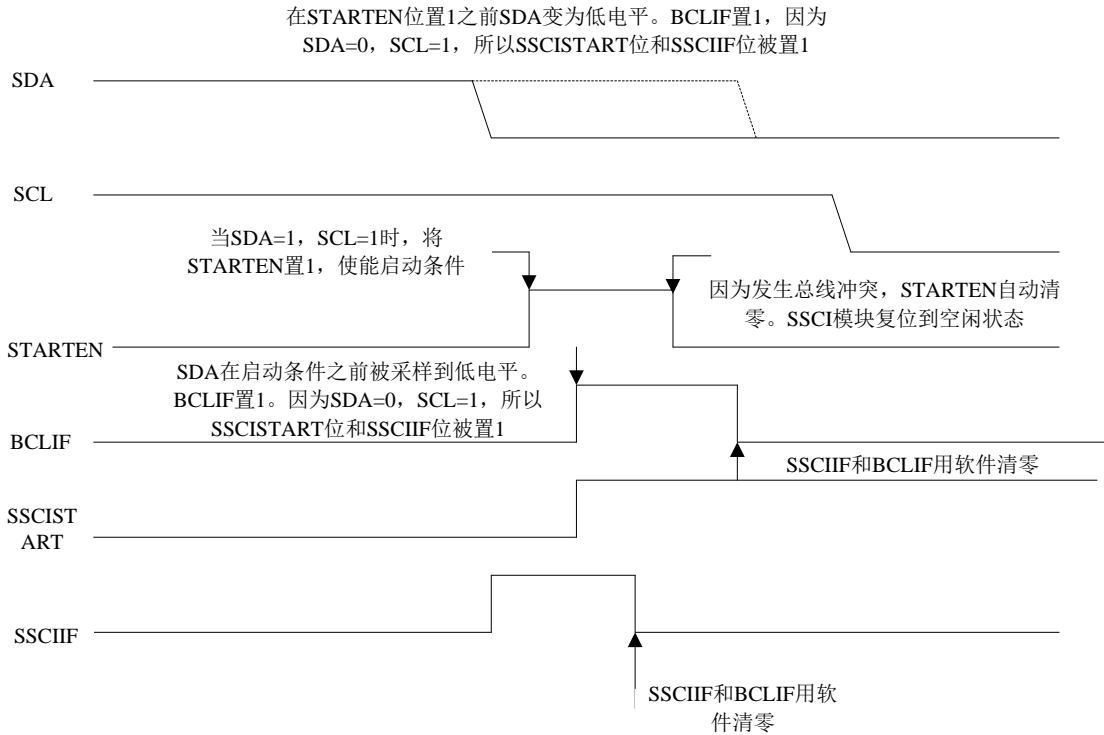


图 16.21 启动条件期间的总线冲突（仅 SDA）

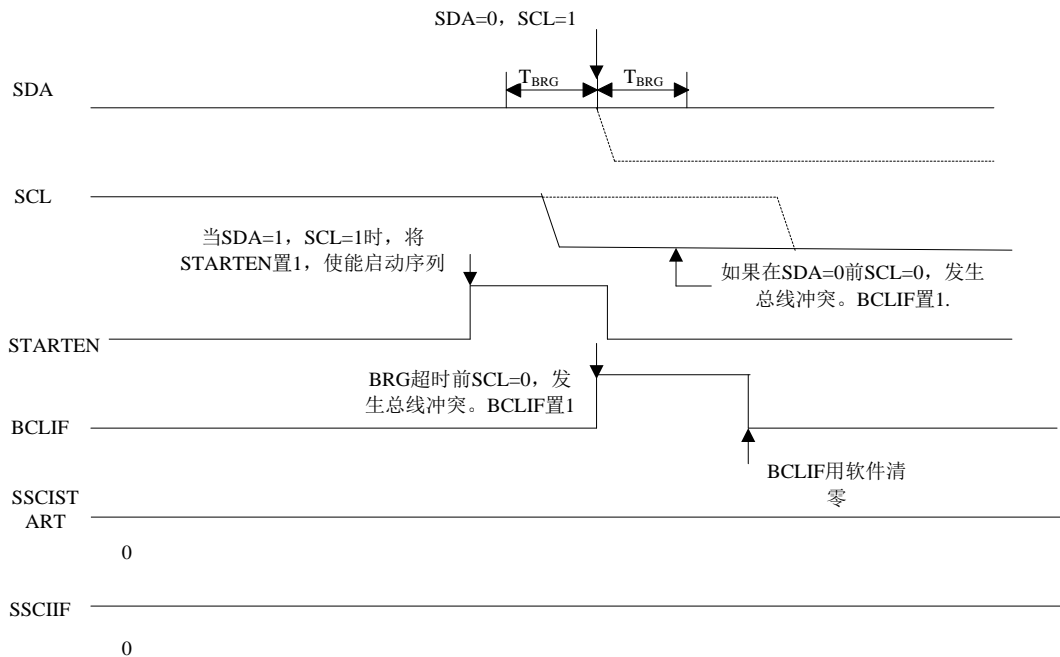


图 16.22 启动条件期间的总线冲突（SCL=0）

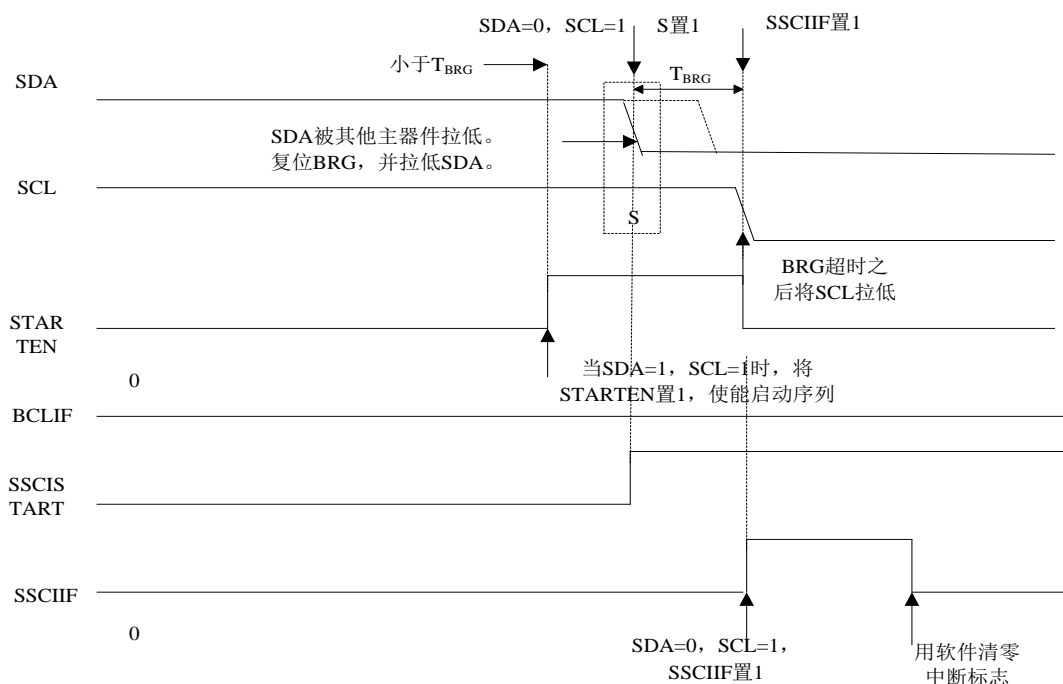


图 16.23 启动条件期间由 SDA 仲裁引起的 BRG 复位

### 16.3.4.3 重复启动条件期间的总线冲突

在下列情况中，重复启动条件期间会发生总线冲突：

- 1) 在 SCL 由低电平变为高电平的过程中，SDA 采样到低电平。
- 2) 在 SDA 被拉为低电平之前，SCL 变为低电平，表示另一个主器件正试图发送一个数据 1。

当用户拉高 SDA 并允许该引脚悬空时，BRG 中装入 SSCIADD<6:0> 中的值并递减计数至 0。接着 SCL 引脚被置为高电平，当 SCL 采样到高电平时，对 SDA 引脚进行采样。

如果 SDA 为低电平，则已发生了总线冲突（即，另一个主器件正试图发送一个数据 0）。如果 SDA 采样为高电平，则 BRG 被重新装入值并开始计数。如果 SDA 在 BRG 超时之前从高电平变为低电平，则没有发生总线冲突，因为两个主器件不可能精确地在同一时刻将 SDA 拉低。

如果 SCL 在 BRG 超时之前从高电平变为低电平，且 SDA 尚未变为低电平，表示发生了总线冲突。在此情况下，在重复启动条件期间另一个主器件正试图发送一个数据 1。

如果在 BRG 超时结束时 SCL 和 SDA 都仍然是高电平，则 SDA 引脚被拉低，BRG 重新装入值并开始计数。在计数结束时，无论 SCL 引脚的状态如何，SCL 引脚都被拉低，重复启动条件结束。

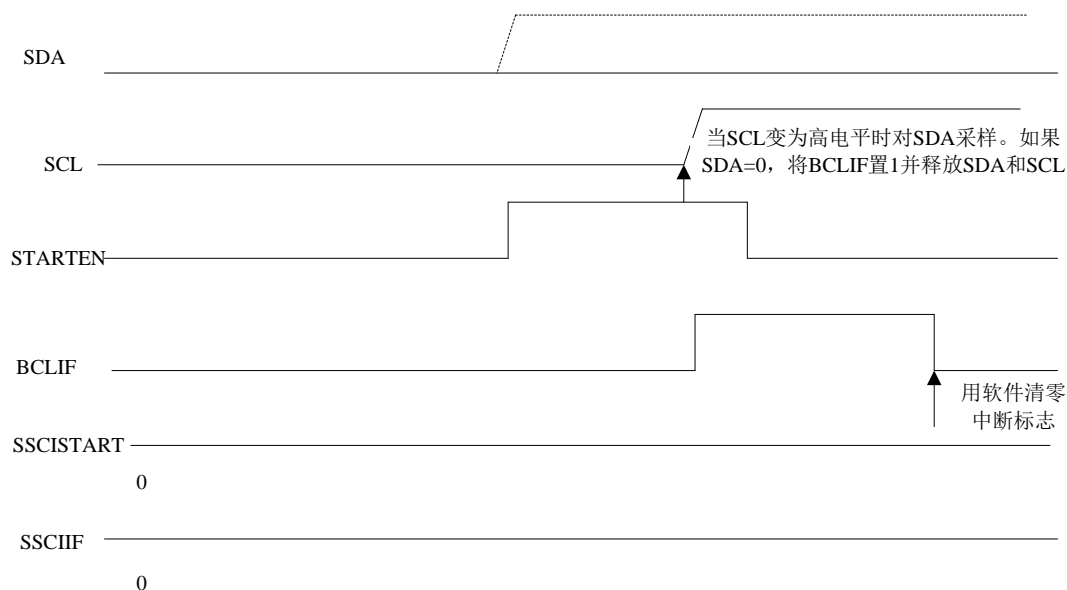


图 16.24 重复启动条件期间的总线冲突（情形 1）

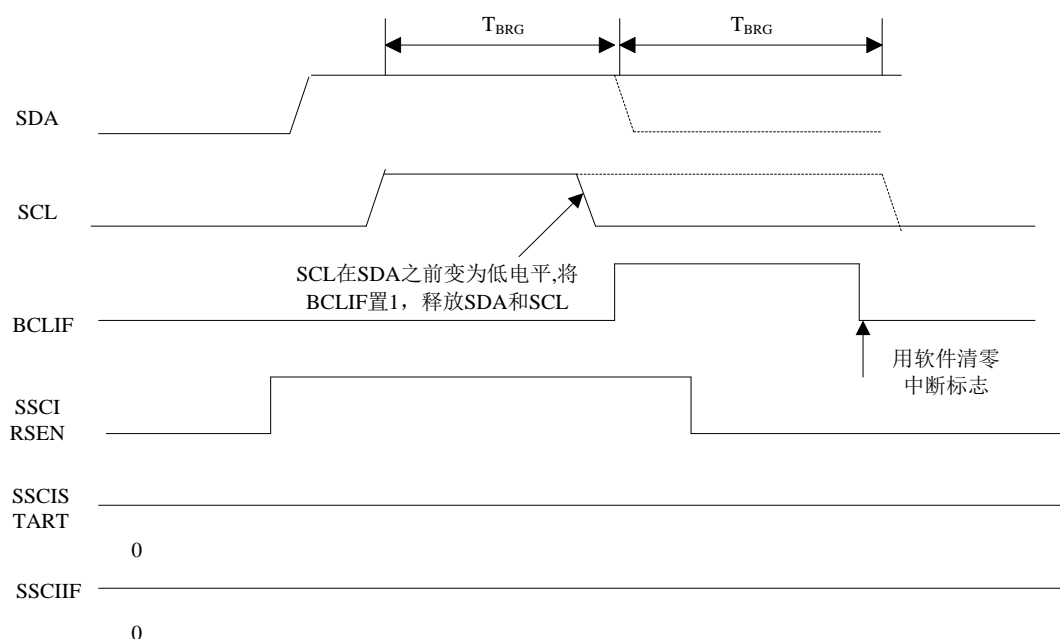


图 16.25 重复启动条件期间的总线冲突（情形 2）

### 16.3.4.4 停止条件期间的总线冲突

以下事件会导致停止条件期间的总线冲突:

- 1) SDA已被拉高并允许悬空为高电平之后, SDA在BRG 超时后被采样到低电平。
- 2) SCL 引脚被拉高之后, SCL 在SDA 变成高电平之前被采样到低电平。

停止条件从SDA被拉低开始。当SDA采样为低电平时, SCL 引脚就可以悬空为高电平。当引脚被采样到高电平时(时钟仲裁), 波特率发生器中装入SSCIADD<6:0>的内容并递减计数到0。BRG 超时后, 采样SDA。如果SDA 采样到低电平, 则已发生总线冲突。这

是因为另一个主器件正试图发送一个数据0。如果SCL 引脚在允许SDA 悬空为高电平前被采样到低电平，也会发生总线冲突。这是另一个主器件正试图发送一个数据0 的又一种情况。

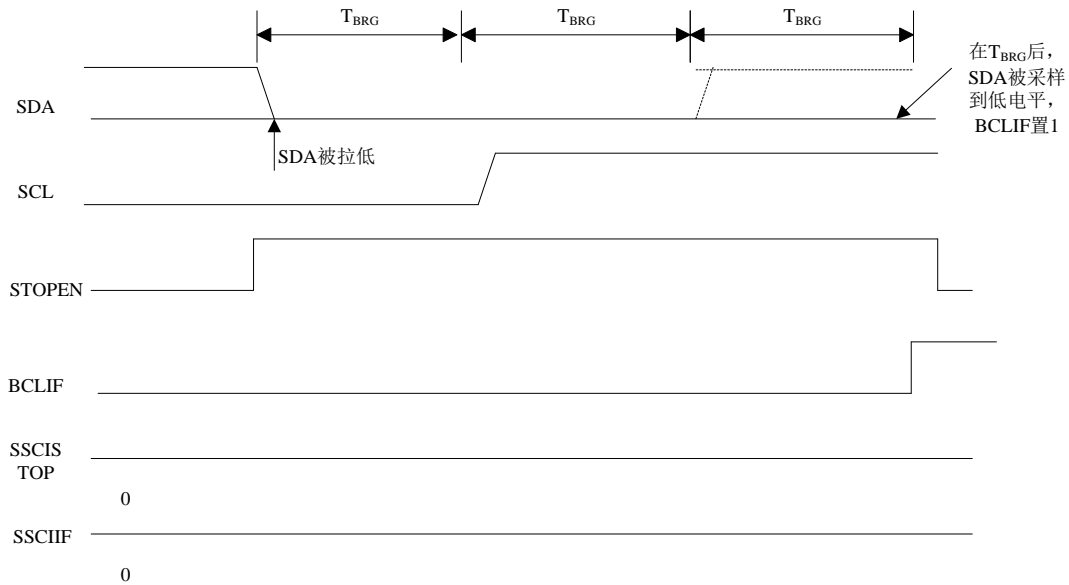


图 16.26 停止条件期间的总线冲突（情形 1）

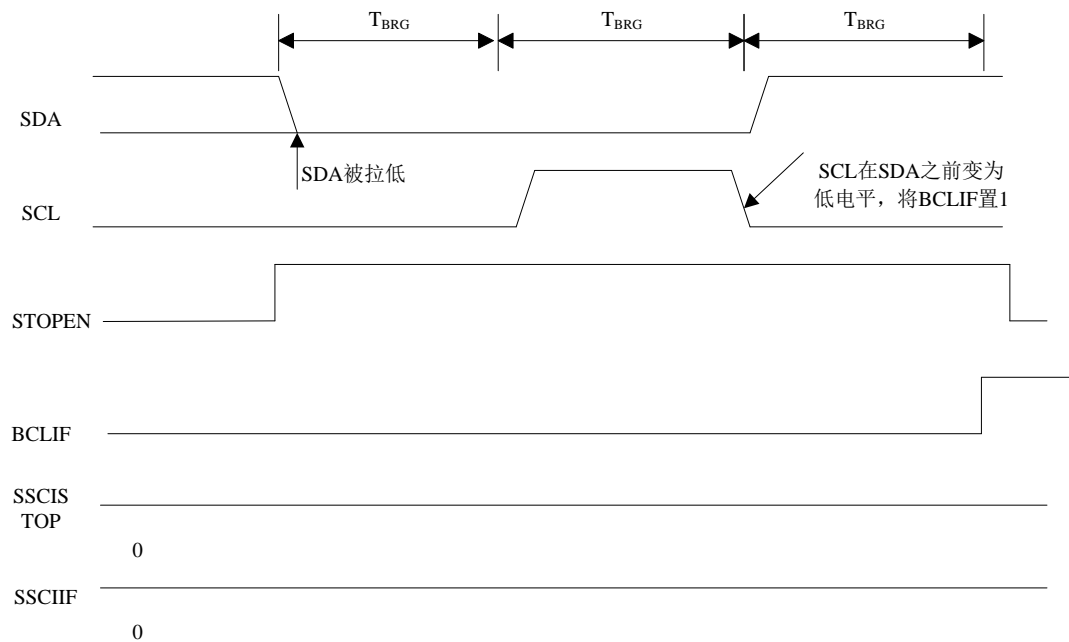


图 16.27 停止条件期间的总线冲突（情形 2）

### 16.3.4.5 SSCI 屏蔽寄存器

在I2C 从动模式下，SSCI 屏蔽（SSCIMSK）寄存器用于在地址比较操作下屏蔽 SSCISR 寄存器中的值。SSCIMSK 寄存器中某位为0 会使SSCISR寄存器中相应的位成为“无关位”。

此寄存器在任何复位条件发生时均复位为全1，因此，在写入屏蔽值前，它对标准

SSCI 操作没有影响。

必须在通过设置SSCIMOD<3:0> 位以选择I2C 从动模式（7位或10 位地址）之前对此寄存器进行初始化。只有通过SSCICTL0 的SSCIMOD<3:0> 位选择了适当的模式后才可访问此寄存器。SSCI 屏蔽寄存器在以下情况下有效:

- 7 位地址模式:与A<7:1> 进行地址比较。
- 10 位地址模式: 仅与 A<7:0> 进行地址比较。SSCI 屏蔽在接收到地址的第一个（高）字节期间无效。

## 16.4 SPI 模式

SPI 是一种应用很广泛的串口总线技术，其特点主要表现在:

- ◆ 3 线或者 4 线数据传输
- ◆ 8 位传输帧格式
- ◆ 主从模式
- ◆ 时钟频率可设
- ◆ 可编程的时钟极性和相位
- ◆ 可触发中断的发送和接收标志

SPI 的原理框图如图 16.28 所示:

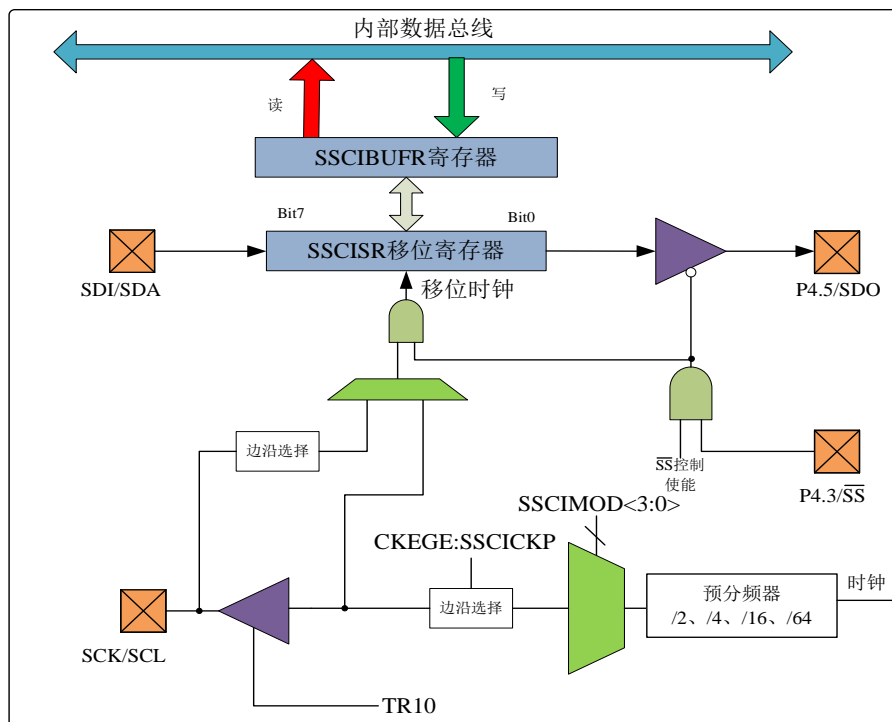


图 16.28 SPI 原理框图

在 SPI 模式下允许同时同步发送和接收 8 位数据。通常使用以下三个引脚来完成通信:

- ◆ 串行数据输出（Serial Data Out，SDO）
- ◆ 串行数据输入（Serial Data In，SDI）
- ◆ 串行时钟（Serial Clock，SCK）
- ◆ 此外，当工作在从动模式下时可以使用第 4 个引脚:
- ◆ 从动选择（ $\overline{SS}$ ）

注意:

- 1) 如果使用SPI 从动模式, 且CKEGE = 1, 则必须使能 $\overline{SS}$  引脚控制 (SSICCTL0<3:0> = 0100)。
- 2) 当SPI 处于从动模式时, 如果 $\overline{SS}$  引脚设为VDD, 那么SPI模块将复位。

### 16.4.1 工作原理

在用户初始化 SPI 时, 需要先通过对相应的控制位 (SSICCTL0<5:0> 和 SSCISTA<7:6>) 编程来指定几个选项。这些控制位用于设置以下选项:

- ◆ 主控模式 (SCK 作为时钟输出)
- ◆ 从动模式 (SCK 作为时钟输入)
- ◆ 时钟极性 (SCK 的空闲状态)
- ◆ 输入数据的采样相位 (数据输出时间的中间或末端)
- ◆ 时钟边沿 (在 SCK 的上升沿/ 下降沿输出数据)
- ◆ 从动选择模式 (仅用于从动模式)

SSCI 模块由一个发送/ 接收移位寄存器 SSCISR (SSCISR 是内部寄存器, 程序无法直接访问) 和数据缓冲寄存器 (SSCIBUFR) 组成。SSCISR 对要发送和接收的数据进行移位, 最高有效位在前。

1) 当在接收数据时, 一旦 8 位数据接收完毕, 该字节就被移入 SSCIBUFR 寄存器。缓冲器满检测位 SSCIBUF (SSCISTA<0>) 和中断标志位 SSCIIF 将会被置 1, 而在新数据接收完毕前, SSCIBUFR 寄存器保存的是上次写入 SSCISR 的数据。

这是一种双重缓冲数据接收方式 (SSCIBUFR), 它允许在 CPU 读取刚接收的数据之前, 就开始接收下一个字节。但在数据发送/接收期间, 任何试图写 SSCIBUFR 寄存器的操作都会被忽略, 并且写冲突检测位 SSCIWCFL (SSICCTL0<7>) 会置 1。此时用户必须用软件将 SSCIWCFL 位清零, 否则无法判别以后对 SSCIBUFR 的写操作是否成功。

2) 当在发送数据时, 为了确保应用软件能有效地接收数据, 应该在要发送的下一数据字节被写入 SSCIBUFR 之前, 读取 SSCIBUFR 中现有的数据。

缓冲器满标志位 SSCIBUF (SSCISTA<0>) 指出将接收到的数据装入 SSCIBUFR (发送完成) 的时间。当 SSCIBUFR 中的数据被读取后, SSCIBUF 位即被清零。如果 SPI 仅作为一个发送器, 则不必理会接收的数据。通常, 可用 SSCI 中断来判断发送和接收完成的时间。必须读取/ 写入 SSCIBUFR。如果不打算使用中断, 用软件查询的方法同样可确保不会发生写冲突。

**注:** 不能直接读写 SSCISR 寄存器, 只能通过寻址 SSCIBUFR 寄存器来访问。此外, SSCI 状态寄存器 (SSCISTA) 指示各种状态条件。

### 16.4.2 使能 SPI/IO 与外部链接

要使能串行端口, SSCI 使能位 SSCIEN (SSICCTL0<5>) 必须置 1。如果要复位或重新配置 SPI 模式, 要先将 SSCIEN 位清零, 重新初始化 SSICCTL0 寄存器, 然后将 SSCIEN 位置 1。这将把 SDI、SDO、SCK 和  $\overline{SS}$  引脚配置为串行端口引脚。要让上述引脚用于串行端口功能, 必须正确设置它们的数据方向位。即:

- ◆ SDI，必须将方向寄存器该位 TR<sub>xx</sub> 置 1（由 SPI 模块自动控制）
- ◆ 对于 SDO，必须将方向寄存器该位 TR<sub>xx</sub> 清零
- ◆ 对于 SCK（主控模式），必须将方向寄存器该位 TR<sub>xx</sub> 清 0
- ◆ 对于 SCK（从动模式），必须将方向寄存器该位 TR<sub>xx</sub> 置 1
- ◆ 对于  $\overline{SS}$ ，必须将方向寄存器该位 TR<sub>xx</sub> 置 1。

对于不需要的任何串行端口功能，可通过将对应的数据方向 TR<sub>x</sub> 寄存器设置为相反值来屏蔽。

### 16.4.3 典型连接

图 16.29给出了两个单片机之间的典型连接。主控制器（处理器1）通过发送SCK信号来启动数据传输。在两个处理器的移位寄存器之间，数据在编程设定的时钟边沿被传送，并在相反的时钟边沿被锁存。必须将两个处理器的时钟极性（SSCICKP）设置为相同，这样两个处理器就可以同时收发数据。数据是否有效，取决于应用软件。这就导致以下三种数据传输情形：

- ◆ 主控制器发送数据 — 从控制器发送无效数据
- ◆ 主控制器发送数据 — 从控制器发送数据
- ◆ 主控制器发送无效数据 — 从控制器发送数据

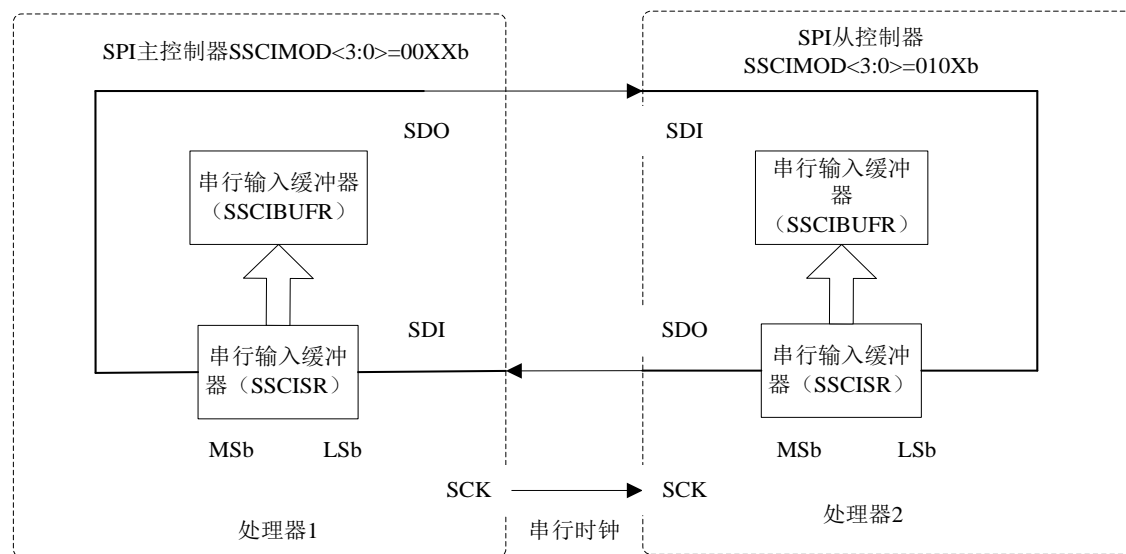


图 16.29 SPI 主从控制器连接

### 16.4.4 主模式

因为由主控制器控制 SCK 信号，所以它可以在任意时刻启动数据传输。主控制器根据软件协议确定从控制器（图 16.29 中的处理器 2）应在何时广播数据。在主控模式下，数据一旦写入 SSCIBUFR 寄存器就开始发送或接收。如果只打算将 SPI 作为接收器，则可以禁止 SDO 输出（将其编程设置为输入）。SSCISR 寄存器按设置的时钟速率，对 SDI 引脚上的信号进行连续移位输入。每收到一个字节，就将其装入 SSCIBUFR 寄存器，就像接收到普通字节一样（中断和状态位相应置 1）。

可通过对 SSCICKP 位（SSCICTL0<4>）进行适当的编程来选择时钟极性。图 16.30、



图 16.31、图 16.32 和图 16.33 将给出 SPI 通信的时序图，其中首先发送的是最高有效位。

在主控模式下，SPI 时钟速率（波特率）可由用户编程设定为下面几种方式之一：

- SCLK/2
- SCLK/4
- SCLK/16
- SCLK/64

图 16.30 给出了主控模式的波形图。当CKEGE位置1 时，SDO数据在SCK出现时钟边沿前一直有效。图中所示的输入采样的变化由SAMPLE（SSCISTA<7>）状态位反映。图中给出了将接收到的数据装入SSCIBUFR的时间。

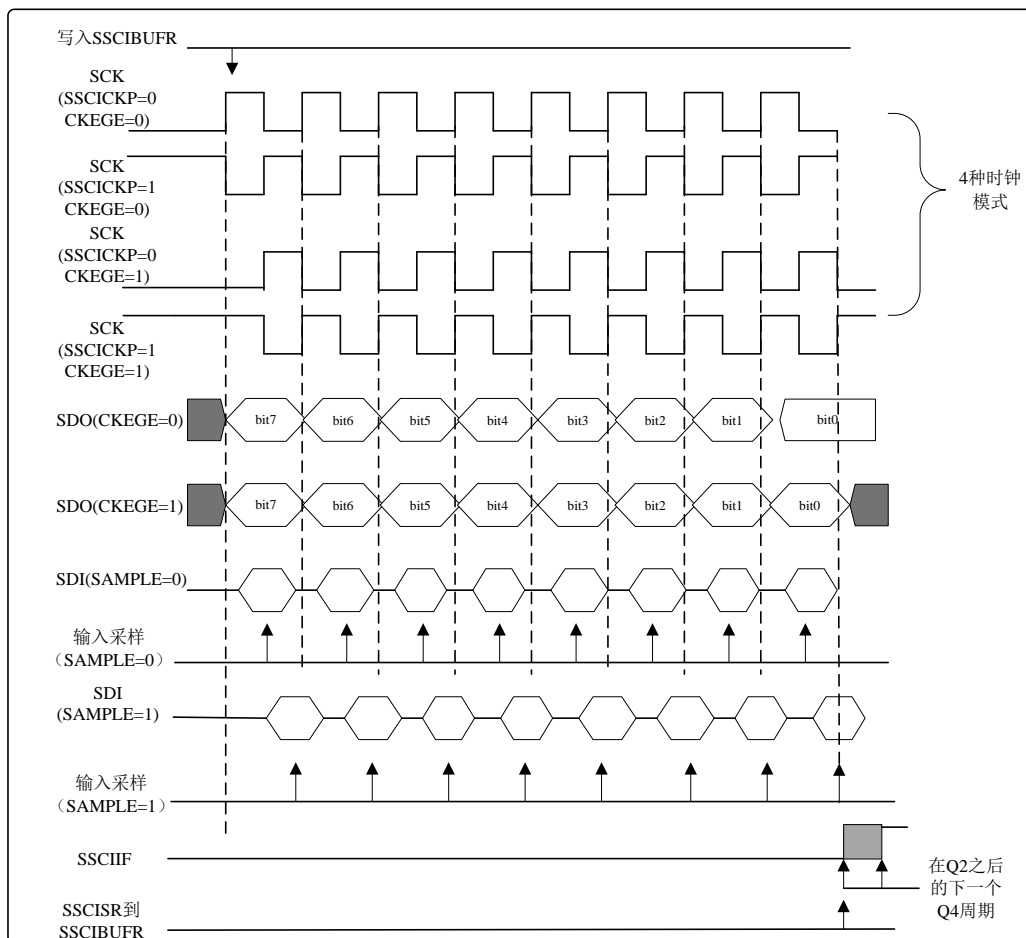


图 16.30 SPI 主控模式时序图

## 16.4.5 从模式

### 16.4.5.1 从动模式

在从动模式下，当SCK引脚上出现外部时钟脉冲时发送和接收数据。当最后一位数据被锁存后，中断标志位SSCIF置1。在从动模式下，外部时钟由SCK引脚上的外部时钟源提供。外部时钟必须满足电气规范中规定的高电平和低电平的最短时间要求。在休眠状态下，从控制器仍可发送/接收数据。当接收到一个字节时，器件从休眠状态唤醒。当有接收数据送到SSCIBUFR后，SSCIBUF信号也会置1。当读取SSCIBUFR后SSCIBUF会自动清零。但有SSCIF信号，一般情况下就不使用SSCIBUF来判断SSCIBUFR是否有数据了。当

SSCIF置1且SSCIBUF置1时，应立即读取SSCIBUFR中的数据，否则会出现数据丢失。

### 16.4.5.2 从动选择同步

$\overline{SS}$  引脚允许器件工作于同步从动模式。SPI必须处于从动模式，并使能 $\overline{SS}$ 引脚控制（SSCICTL0<3:0> = 0100）。要让 $\overline{SS}$ 引脚充当输入端，则不能将此引脚驱动为低电平。数据锁存器必须为高电平。当 $\overline{SS}$ 引脚为低电平时，使能数据的发送和接收，同时驱动SDO引脚。当 $\overline{SS}$ 引脚变为高电平时，即使是在字节的发送过程中，SDO引脚也不再被驱动，而是变成悬空输出状态。根据应用的需要，可在SDO引脚上外接上拉/下拉电阻。

注意：

- 1) 当SPI处于从动模式，并且 $\overline{SS}$ 引脚控制使能（SSCICTL0<3:0> = 0100）时，如果 $\overline{SS}$ 引脚置为VDD电平将使SPI模块复位。
- 2) 如果SPI工作在从动模式下并且CKEGE置1，则必须使能 $\overline{SS}$ 引脚控制。

当SPI模块复位时，位计数器被强制为0。这可以通过强制将 $\overline{SS}$ 引脚拉为高电平或将SSCIEN位清零实现。将SDO引脚和SDI引脚相连，可以仿真二线制通信。当SPI需要作为接收器工作时，SDO引脚可以被配置为输入端。这样就禁止了从SDO发送数据。因为SDI不会引起总线冲突，因而总是可以将其保留为输入（SDI功能）。

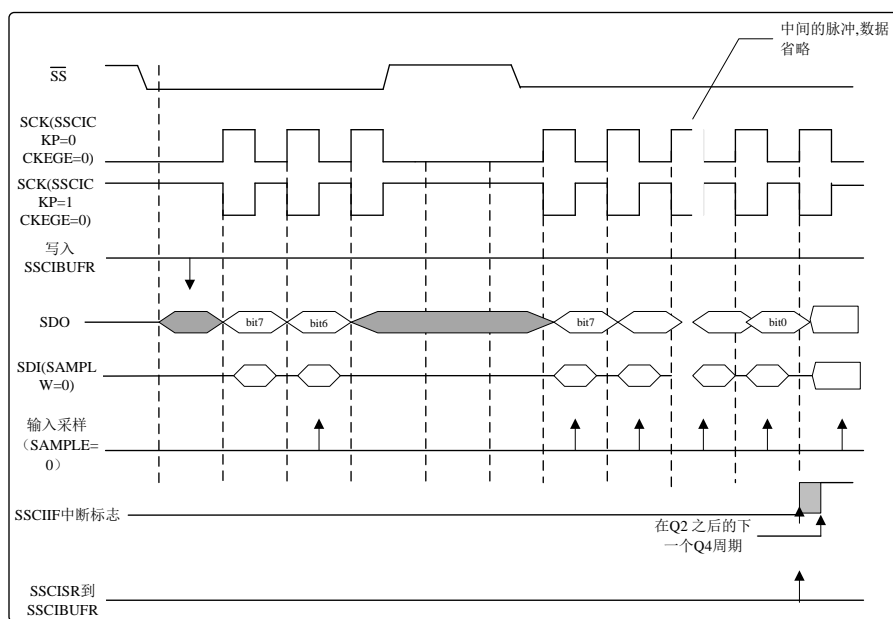


图 16.31 从动时序图

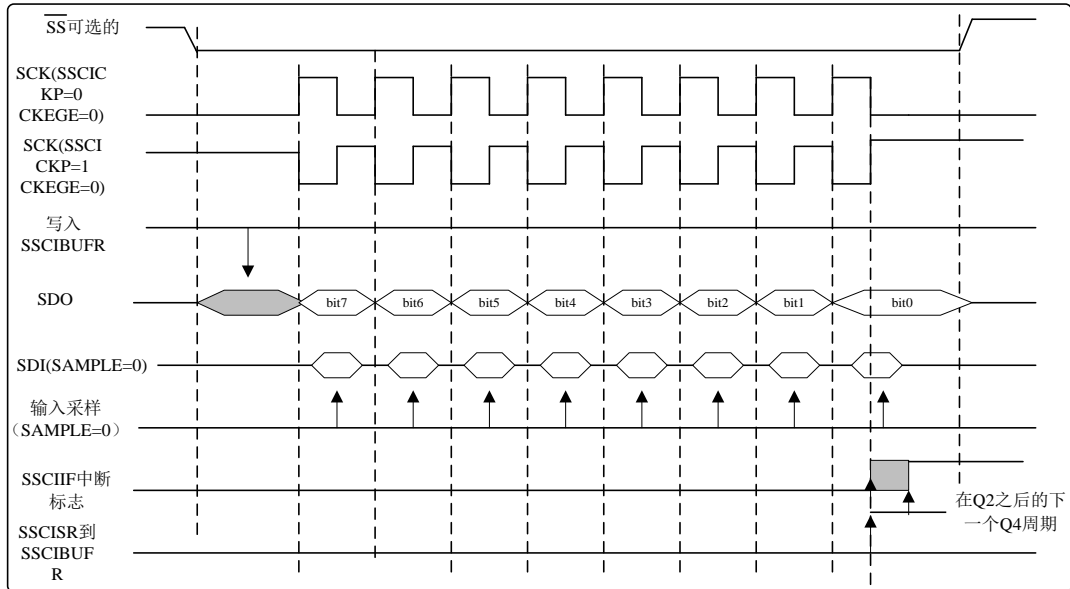


图 16.32 SPI 模式时序图（从动模式且 CKEGE=0）

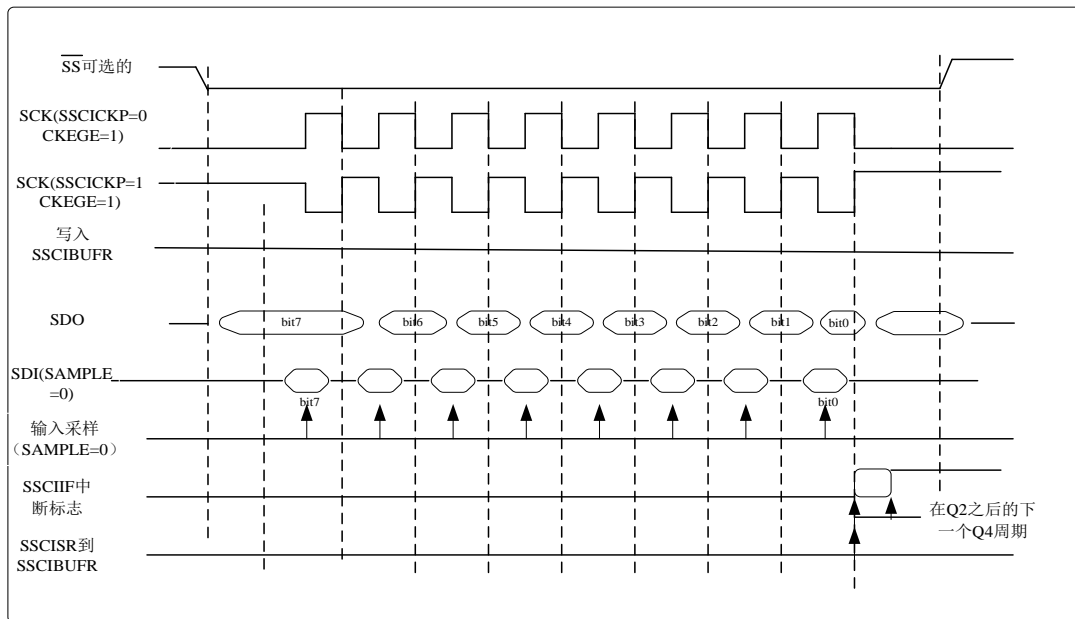


图 16.33 SPI 模式时序图（从动模式且 CKEGE=1）

### 16.4.6 休眠模式和复位

#### 休眠模式下的工作

在主机模式下，进入休眠模式后所有模块的时钟都停振，在器件被唤醒前，发送/接收也将保持原先的状态。在器件恢复正常工作模式后，模块将继续发送/接收数据。

在从动模式下，SPI发送/接收移位寄存器与器件异步工作。这可以使器件在休眠状态时，仍可使数据被移入SPI发送/接收移位寄存器。当接收完8位数据后，SSCI中断标志位将置1，如果此时该中断是允许的，还将唤醒器件。

## 复位的影响

复位会禁止SSCI模块并终止当前的数据传输。

### 16.4.7 SPI 四种工作模式设置

#### 16.4.7.1 主控发送工作流程

主要步骤:

- 1、通过 SSCIMOD<3:0>确定 SPI 主控方式
  - 0000:SPI 主控方式, 时钟=工作时钟/4
  - 0001:SPI 主控方式, 时钟=工作时钟/16
  - 0010:SPI 主控方式, 时钟=工作时钟/64
  - 0011:SPI 主控方式, 时钟= T2 输出/2
- 2、通过寄存器 SSCICTL0 的 SSCICKP 和寄存器 SSCISTA 的 CKEGE 选择时钟与数据的时序关系;
- 3、将相应的引脚 SDO 引脚设置为输出, SCK 引脚设置为输出;
- 4、置 1 SSCIEN 位, 使能 SSCI 模块;
- 5、此时 SSCIIF 清零, 将要发送的数据写到 SSCIBUFR 寄存器;
- 6、当一个字节发送完毕 SSCIIF 自动置 1 (软件清零), 如果需要中断, 则使能相应中断使能位;

#### 16.4.7.2 从动接收工作流程

主要步骤:

- 1、通过 SSCIMOD<3:0>确定 SPI 从动方式:
  - 0100:SPI 从动模式, 时钟由 SCK 引脚输入, 使能  $\overline{SS}$  引脚功能, 此时要设置  $\overline{SS}$  引脚为输入口
  - 0101:SPI 从动模式, 时钟由 SCK 引脚输入, 关闭  $\overline{SS}$  引脚功能,  $\overline{SS}$  被用作普通 I/O 引脚
- 2、通过寄存器 SSCICTL0 的 SSCICKP 和寄存器 SSCISTA 的 CKEGE 选择时钟与数据的时序关系; 与主控方的时钟时序要一致;  
注意:当 CKEGE 为 1 时, 只有信号  $\overline{SS}$  为低电平, 从动端才会接收数据, 否则从动端不工作;  $\overline{SS}$  信号都来源于主控方。  
将相应的引脚 SDI 引脚设置输入, SCK 引脚设置为输入; 使能  $\overline{SS}$  引脚功能, 此时要设置  $\overline{SS}$  引脚为输入口;
- 3、置 1 SSCIEN 位, 使能 SSCI 模块;
- 4、当接受到一个字节的数据后, SSCIIF 自动置 1 (软件清零), SSCIBUF 置 1, 应立即读取 SSCIBUFR 的值;
- 5、如果需要中断, 则使能相应的中断使能位;

### 16.4.7.3 主控接收工作流程

主要步骤:

- 1、通过 SSCIMOD<3:0>确定 SPI 主控方式
  - 0000:SPI 主控方式, 时钟=工作时钟/4
  - 0001:SPI 主控方式, 时钟=工作时钟/16
  - 0010:SPI 主控方式, 时钟=工作时钟/64
  - 0011:SPI 主控方式, 时钟=T2 输出/2
- 2、通过寄存器 SSCICTL0 的 SSCICKP 和寄存器 SSCISTA 的 CKEGE 选择时钟与数据的时序关系;
- 3、将相应的引脚 SDI 引脚设置输入, SCK 引脚设置为输出;
- 4、置 1 SSCIEN 位, 使能 SSCI 模块;
- 5、当接受到一个字节的数据后, SSCIIF 自动置 1 (软件清零), SSCIBUF 置 1, 应立即读取 SSCIBUFR 的值;
- 6、如果需要中断, 则使能相应的中断使能位;

### 16.4.7.4 从动发送工作流程

主要步骤:

- 1、通过 SSCIMOD<3:0>确定 SPI 主控方式:
  - 0100:SPI 从动模式, 时钟由 SCK 引脚输入, 使能  $\overline{SS}$  引脚功能, 此时要设置  $\overline{SS}$  引脚为输入口
  - 0101:SPI 从动模式, 时钟由 SCK 引脚输入, 关闭  $\overline{SS}$  引脚功能,  $\overline{SS}$  被用作普通 I/O 引脚
- 2、通过寄存器 SSCICTL0 的 SSCICKP 和寄存器 SSCISTA 的 CKEGE 选择时钟与数据的时序关系;
- 3、将相应的引脚 SDO 引脚设置为输出, SCK 引脚设置为输入; 如果使能  $\overline{SS}$  引脚功能, 此时要设置  $\overline{SS}$  引脚为输入口;
- 4、置 1 SSCIEN 位, 使能 SSCI 模块;
- 5、此时 SSCIIF 清零, 将要发送的数据写到 SSCIBUFR 寄存器;
- 6、当一个字节发送完毕 SSCIIF 自动置 1 (软件清零), 如果需要中断, 则使能相应中断使能位;

## 17 实时时钟（RTC）模块

### 17.1 概述

实时时钟 (Real Time Counting, RTC) 单元提供给用户实时时间以及日历信息。RTC 的时钟源RTCOSC振荡器外接32.768 kHz 晶振提供。RTC 单元通过时间寄存器提供时间信息(秒、分、时、星期、日、月、年)。数据信息由 BCD 码格式进行表示。修改计数器的值可以重新设置系统当前的时间和日期。

该单元也提供闹钟功能，用户可预先在时间闹钟寄存器中设置闹钟日期进行闹铃设置。

#### RTC特性包含：

- ◆ 提供实时时钟的日历功能，自动解决闰年问题，计时范围 100 年
- ◆ 闰年、平年显示
- ◆ 12/24 小时模式选择功能
- ◆ 高精度的数字时钟校正功能
- ◆ 支持闹钟中断功能
- ◆ 支持时间节拍中断功能，提供 8 个周期选项
- ◆ 支持时分秒进程中中断功能
- ◆ 内置 2 个 8 位定时器 TMR0 和 TMR1

#### 17.1.1 原理框图

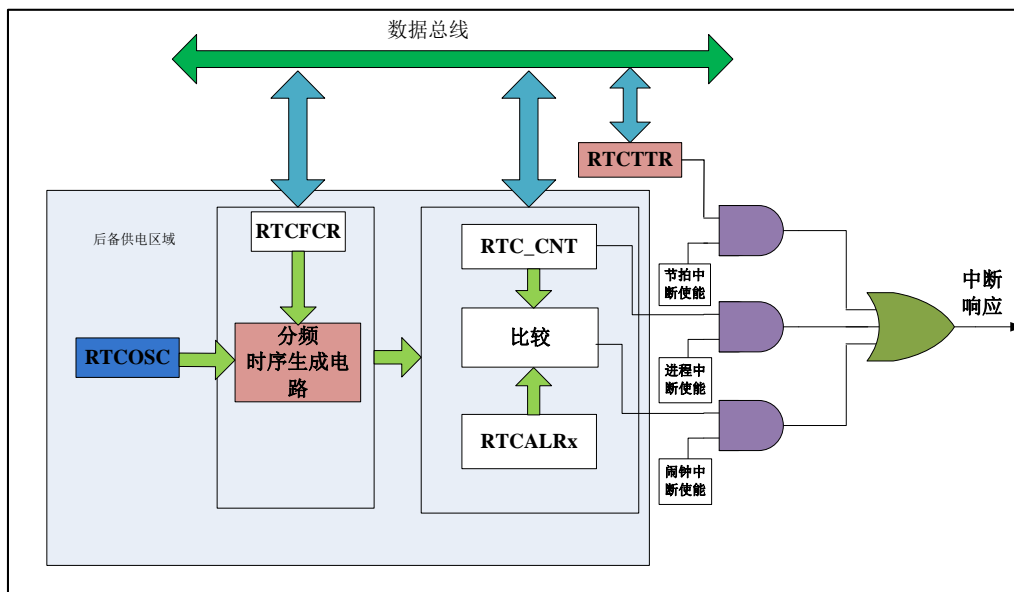


图 17.1 RTC 原理框图

从上图可以看出，RTC 模块由外部提供时钟，可以通过设置时钟校正 RTCFCR 寄存器进行时钟校正，如果不需要校正，可以对校正值写 00H。RTC 提供实时时间和日历进程。详见章节“功能描述”。

RTC 还支持各种中断功能，通过对写入响应的实时时钟寄存器选择响应的进程中中断，也可以与 RTC 闹钟中断寄存器进行比较，当两者值相等时便会产生相应的闹钟中断，使 ALRIF 置 1（用软件清零）。还具有周期时间节拍中断功能，通过设置相应的周期时间节拍进行相应的时间节拍中断。

注：RTC 所有寄存器的读写需要在 BKPCTL 寄存器 bit7=1 时，才能进行读写。

## 17.2 寄存器描述

### 17.2.1 相关寄存器

表 17.1 RTC 相关寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
318H	RTCSRT	SRT7	SRT6	SRT5	SRT4	SRT3	SRT2	SRT1	SRT0
319H	RTCSTU	RTCALREN	-	RESET	LIF	HT	CNF	RTOFF	RTCLD
31CH	RTCALRS	ALRS7	ALRS6	ALRS5	ALRS4	ALRS3	ALRS2	ALRS1	ALRS0
31DH	RTCALRM	ALRM7	ALRM6	ALRM5	ALRM4	ALRM3	ALRM2	ALRM1	ALRM0
31EH	RTCALRH	ALRH7	-	ALRH5	ALRH4	ALRH3	ALRH2	ALRH1	ALRH0
31FH	RTCALRW	ALRW7	-	-	-	-	ALRW2	ALRW1	ALRW0
320H	RTCTTR	RTCOE	RTCOS	-	-	-	RTCTTR2	RTCTTR1	RTCTTR0
321H	RTCFCR	V7	V6	V5	V4	V3	V2	V1	V0
322H	RTCSEC	-	SEC6	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0
323H	RTCMIN	-	MIN6	MIN5	MIN4	MIN3	MIN2	MIN1	MIN0
324H	RTCHOUR	-	AMPM	HUR5	HUR4	HUR3	HUR2	HUR1	HUR0
325H	RTCWEK	-	-	-	-	-	WEK2	WEK1	WEK0
326H	RTCDAY	-	-	DAY5	DAY4	DAY3	DAY2	DAY1	DAY0
327H	RTCMTH	-	-	-	MTH4	MTH3	MTH2	MTH1	MTH0
328H	RTCYEAR	YER7	YER6	YER5	YER4	YER3	YER2	YER1	YER0
32CH	RTCTMRTL	CKST13	CKST12	CKST11	CKST10	CKST03	CKST02	CKST01	CKST00
32DH	RTCTMREN	-	-	-	-	-	-	TMR1EN	TMR0EN
32EH	RTCTMR1	RTC 定时器 1 初值配置寄存器							
32FH	RTCTMR0	RTC 定时器 0 初值配置寄存器							

### 17.2.2 实时时钟启动寄存器 RTCSRT

寄存器: RTCSRT: 实时时钟启动寄存器(地址:318H)

复位值	bit7							bit0	
0000 0000	SRT7	SRT6	SRT5	SRT4	SRT3	SRT2	SRT1	SRT0	SRT0
	W-R0	W-R0	W-R0	W-R0	W-R0	W-R0	W-R0	W-R0	W-R0

SRT<7:0>: 实时时钟启动位  
 写固定值 55H 以使能 RTC 模块;  
 写任何其他值将禁止 RTC 模块;  
 SRT<7:0>读出为全 0。

### 17.2.3 状态显示寄存器 RTCSTU

寄存器: RTCSTU: 状态显示寄存器(地址:319H)

	bit7							bit0
复位值 0-00 0000	RTCALRE N	-	RESET	LIF	HT	CNF	RTOFF	RTCLD
	R/W	U	R/W	R/W	R/W	R/W	R	R

RTCALREN: RTC 闹钟功能总使能位

1 = 使能 RTC 闹钟功能

0 = 禁止 RTC 闹钟功能

RESET: RTC 模块复位位

1 = RTC 模块处于复位状态

0 = RTC 模块退出复位状态

LIF: 闰年指示标志

1 = 当前年份为闰年

0 = 当前年份为平年

HT: 选择小时显示的类型

1 = 小时显示为 12 小时制

0 = 小时显示为 24 小时制

CNF: 配置标志位

此位必须由软件置'1'以进入配置模式,从而允许向实时时钟寄存器、闹钟中断时间寄存器写入数据。只有当此位在被置 1 并重新由软件清 0 后,才会执行写操作。

1 = 进入配置模式。

0 = 退出配置模式(开始更新 RTC 寄存器);

RTOFF: RTC操作关闭

RTC 模块利用这位来指示对其寄存器进行的最后一次操作的状态,指示操作是否完成。若此位为'0',则表示无法对任何的 RTC 寄存器进行写操作。

此位为只读位。

1 = 上一次对 RTC 寄存器的写操作已经完成。

0 = 上一次对 RTC 寄存器的写操作仍在进行;

RTCLD: RTC活动状态 (只读)

1 = RTC 处在正常有效状态

0 = RTC 处在复位状态



## 17.2.4 闹钟中断时间寄存器

闹钟中断时间寄存器分别对4个寄存器进行设置完成对闹钟时间秒，分，时，星期的设定。寄存器的构成与实时数据寄存器的星期、时、分、秒数据寄存器相同，利用BCD码来表现。请不要设定为不存在日期。

另外，设定的闹钟时刻数据需要与状态显示寄存器的HT位中所设定的12小时制或者24小时制显示相匹配。

### 17.2.4.1 闹钟中断秒寄存器 RTCALRS

寄存器: RTCALRS: 闹钟中断秒寄存器(地址:31CH)

	bit7						bit0	
复位值 0000 0000	ALRSEN	ALRS6	ALRS5	ALRS4	ALRS3	ALRS2	ALRS1	ALRS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ALRSEN: 闹钟秒时间使能位

0 = 禁止秒时间 ALRS<6:0>参与闹钟

1 = 使能秒时间 ALRS<6:0>参与闹钟

ALRS<6:0>: 设置闹钟中断秒时间(闹钟秒的数值为 00~59)，采用 BCD 编码

### 17.2.4.2 闹钟中断分寄存器 RTCALRM

寄存器: RTCALRM: 闹钟中断分寄存器(地址:31DH)

	bit7						bit0	
复位值 0000 0000	ALRMEN	ALRM6	ALRM5	ALRM4	ALRM3	ALRM2	ALRM1	ALRM0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ALRMEN: 闹钟分时间使能位

0 = 禁止分时间 ALRM<6:0>参与闹钟

1 = 使能分时间 ALRM<6:0>参与闹钟

ALRM<6:0>: 设置闹钟中断分时间(闹钟分的数值为 00~59)，采用 BCD 编码

### 17.2.4.3 闹钟中断时寄存器 RTCALRH

寄存器: RTCALRH: 闹钟中断时寄存器(地址:31EH)

	bit7						bit0	
复位值 0-00 0000	ALRHEN	-	ALRH5	ALRH4	ALRH3	ALRH2	ALRH1	ALRH0
	R/W	U	R/W	R/W	R/W	R/W	R/W	R/W

ALRHEN: 闹钟小时时间使能位

0 = 禁止小时时间 ALRH<5:0>参与闹钟

1 = 使能小时时间 ALRH<5:0>参与闹钟

ALRH<5:0>: 设置闹钟中断小时时间（闹钟时的数值为00~23），采用BCD编码

#### 17.2.4.4 闹钟中断星期寄存器 RTCALRW

寄存器: RTCALRW: 闹钟中断星期寄存器(地址:31FH)

	bit7					bit0		
复位值 0--- -000	ALRWEN	-	-	-	-	ALRW2	ALRW1	ALRW0
	R/W	U	U	U	U	R/W	R/W	R/W

ALRWEN: 闹钟星期时间使能位

0 = 禁止星期时间 ALRW<2:0>参与闹钟

1 = 使能星期时间 ALRW<2:0>参与闹钟

ALRW<2:0>: 设置闹钟中断星期时间（闹钟星期数值为1~7），采用BCD编码

#### 17.2.5 时间节拍中断寄存器 RTCTTR

周期时间节拍中断有8个选项：1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 以及 1 秒，通过设置 RTCTTR寄存器来进行选择。

寄存器: RTCTTR: 时间节拍中断寄存器(地址:320H)

	bit7					bit0		
复位值 00-- -000	RTCOE	RTCOS	-	-	-	RTCTTR2	RTCTTR1	RTCTTR0
	R/W	R/W	U	U	U	R/W	R/W	R/W

RTCOE: RTC输出使能位

0 = 禁止RTC输出到PA.0口

1 = 使能RTC输出到PA.0口

RTCOS: RTC输出选择位

0 = 选择时间节拍输出到PA.0口

1 = 选择闹钟信号输出到PA.0口

RTCTTR<2:0>:时间节拍设置位

000=时间节拍为1秒

001=时间节拍为1/2秒

010=时间节拍为1/4秒

011=时间节拍为1/8秒

100=时间节拍为1/16秒

101=时间节拍为1/32秒

110=时间节拍为1/64秒

111=时间节拍为1/128秒

注：RTCOE位和RTCOS位为系统保留位。

### 17.2.6 实时时钟校正寄存器 RTCFCR

时钟校正寄存器为1字节的寄存器，是用来校正时钟偏差（提前或滞后）的寄存器。在不使用时钟校正功能时，请设定为"00 h"。

**寄存器：** RTCFCR: 时间校正寄存器(地址:321H)

	bit7						bit0	
复位值 0000 0000	V7	V6	V5	V4	V3	V2	V1	V0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

V<7:0>: 将校正值转化为2进制值置入 B7-B0 中。

### 17.2.7 实时时钟寄存器

实时时钟寄存器主要用于读取或设置 RTC 计数器，具体如下面章节所述。

#### 17.2.7.1 实时时钟秒寄存器 RTCSEC

用户可通过读此寄存器来获得 RTC 计数器当前值的秒时间。秒数值为 00~59，采用BCD 编码。

**寄存器：** RTCSEC: 实时时钟秒寄存器(地址:322H)

	bit7						bit0	
复位值 - 000 0000	-	SEC6	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0
	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

注：要对此寄存器进行写操作，必须先进入配置模式(CNF = 1)。对此寄存器进行读操作时先清零 CNF，再置位 CNF。

#### 17.2.7.2 实时时钟分寄存器 RTCMIN

用户可通过读此寄存器来获得 RTC 计数器当前值的分时间。分钟数值为 00~59，采用BCD 编码。

**寄存器：** RTCMIN: 实时时钟分寄存器(地址:323H)

	bit7						bit0	
复位值 - 000 0000	-	MIN6	MIN5	MIN4	MIN3	MIN2	MIN1	MIN0
	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

注：要对此寄存器进行写操作，必须先进入配置模式(CNF = 1)。对此寄存器进行读操作时先清零CNF，再置位CNF。

### 17.2.7.3 实时时钟时寄存器 RTCHOUR

寄存器: RTCHOUR: 实时时钟时寄存器(地址:324H)

	bit7							bit0
复位值 -000 0000	-	AMPM	HUR5	HUR4	HUR3	HUR2	HUR1	HUR0
	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

AMPM: 当 HT 置 1 选择 24 小时制时, 该位无效, 0,1 均可。

当 HT 置 0 选择 12 小时制时:

1 = PM

0 = AM

HUR<5:0>: 可通过读此寄存器来获得RTC计数器当前值的时时间。小时数值为00~23或00~12, 采用BCD编码

注: 要对此寄存器进行写操作, 必须先进入配置模式(CNF = 1)。对此寄存器进行读操作时先清零CNF, 再置位CNF。

### 17.2.7.4 实时时钟星期寄存器 RTCWEK

用户可通过读此寄存器来获得RTC计数器当前值的星期时间。星期数值为1~7, 采用BCD编码。

寄存器: RTCWEK: 实时时钟星期寄存器(地址:325H)

	bit7							bit0
复位值 ---- -000	-	-	-	-	-	WEK2	WEK1	WEK0
	U	U	U	U	U	R/W	R/W	R/W

注: 要对此寄存器进行写操作, 必须先进入配置模式(CNF = 1)。对此寄存器进行读操作时先清零CNF, 再置位CNF。

### 17.2.7.5 实时时钟日寄存器 RTCDAY

用户可通过读此寄存器来获得RTC计数器当前值的日时间。日的数值为01~31, 采用BCD编码。

寄存器: RTCDAY: 实时时钟日寄存器(地址:326H)

	bit7							bit0
复位值 --00 0000	-	-	DAY5	DAY4	DAY3	DAY2	DAY1	DAY0
	U	U	R/W	R/W	R/W	R/W	R/W	R/W

注: 要对此寄存器进行写操作, 必须先进入配置模式(CNF = 1)。对此寄存器进行读操作时先清零CNF, 再置位CNF。

### 17.2.7.6 实时时钟月寄存器 RTCMTH

用户可通过读此寄存器来获得RTC计数器当前值的月时间。月的数值为01~12, 采用

BCD编码。

**寄存器:** RTCMTH: 实时时钟月寄存器(地址:327H)

		bit7						bit0	
复位值	---0 0000	-	-	-	MTH4	MTH3	MTH2	MTH1	MTH0
		U	U	U	R/W	R/W	R/W	R/W	R/W

注: 要对此寄存器进行写操作, 必须先进入配置模式(CNF = 1)。对此寄存器进行读操作时先清零 CNF, 再置位 CNF。

### 17.2.7.7 实时时钟年寄存器 RTCYEAR

用户可通过读此寄存器来获得RTC计数器当前值的年时间。年的数值为00~99, 采用BCD编码。

**寄存器:** RTCYEAR: 实时时钟年寄存器(地址:328H)

		bit7						bit0	
复位值	0000 0000	YER7	YER6	YER5	YER4	YER3	YER2	YER1	YER0
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

注: 要对此寄存器进行写操作, 必须先进入配置模式(CNF = 1)。对此寄存器进行读操作时先清零CNF, 再置位CNF。

### 17.2.7.8 实时时钟定时器使能寄存器 RTCTMREN

**寄存器:** RTCTMREN: 实时时钟定时器使能寄存器(地址: 32DH)

		bit7						bit0	
复位值	---- --00	-	-	-	-	-	-	TMR1EN	TMR0EN
		U	U	U	U	U	U	R/W	R/W

TMR1EN: 定时器 1 使能位  
1 = 使能定时器 1  
0 = 禁止定时器 1

TMR0EN: 定时器 0 使能位  
1 = 使能定时器 0  
0 = 禁止定时器 0

### 17.2.7.9 实时时钟定时器控制寄存器 RTCTMRCTL

**寄存器:** RTCTMRCTL: 实时时钟定时器控制寄存器(地址: 32CH)

		bit7						bit0	
复位值	0000 0000	CKST13	CKST12	CKST11	CKST10	CKST03	CKST02	CKST01	CKST00
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CKST1<3:0>: 定时器 1 时钟源选择位

0000 = RTC 时钟源/128 约 1/256s  
 0001 = RTC 时钟源/512 约 1/64s  
 0010 = RTC 时钟源/1024 约 1/32s  
 0011 = RTC 时钟源/2048 约 1/16s  
 0100 = RTC 时钟源/8192 约 1/4s  
 0101 = 1s  
 0110 = 1min  
 0111 = 1hour  
 1000 = 计数器 TMR1 和 TMR0 级联

CKST0<3:0>: 定时器 0 时钟源选择位

0000 = RTC 时钟源/128 约 1/256s  
 0001 = RTC 时钟源/512 约 1/64s  
 0010 = RTC 时钟源/1024 约 1/32s  
 0011 = RTC 时钟源/2048 约 1/16s  
 0100 = RTC 时钟源/8192 约 1/4s  
 0101 = 1s  
 0110 = 1min  
 0111 = 1hour

#### 17.2.7.10 实时时钟定时器 0 计数寄存器 RTCTMR0

**寄存器:** RTCTMR0:实时时钟定时器0计数寄存器(地址: 32FH)  
 bit7

复位值 0000 0000	TMR07	TMR06	TMR05	TMR04	TMR03	TMR02	TMR01	TMR00	bit0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TMR0<7:0>: 定时器 0 初值配置位

#### 17.2.7.11 实时时钟定时器 1 计数寄存器 RTCTMR1

**寄存器:** RTCTMR1:实时时钟定时器1计数寄存器(地址: 32EH)  
 bit7

复位值 0000 0000	TMR17	TMR16	TMR15	TMR14	TMR13	TMR12	TMR11	TMR10	bit0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TMR1<7:0>: 定时器 1 初值配置位

## 17.3 功能描述

### 17.3.1 RTC 初始化

备份区外设复位寄存器 PHCLR 中的 VRTCS 位为 RTC 模块的电源控制位，默认为 1，即 RTC 模块处于有电状态。请保持该位为 1，以确保 RTC 模块可以正常工作。

在使能 RTC 模块之前，用户可通过 RTCSTU 寄存器中的 RESET 位对 RTC 模块进行复位以初始化实时时钟寄存器及内部逻辑。

RTCSRT 寄存器用于 RTC 模块的使能。向 RTCSRT 寄存器写固定值 55H 以使能 RTC 模块，写任何其他值将禁止 RTC 模块。RTCSRT 寄存器读出为全 0。

### 17.3.2 RTC 启动

#### RTC 实时时钟启动步骤：

1. RTC 的初始化；配置 RTCSTU 寄存器的 RESET 位置 1 以初始化 RTC 模块内部逻辑及实时时钟寄存器，再将该位清零，退出复位状态。
2. RTC 的初值配置；设置 RTC 实时时钟寄存器，设置年/月/日/星期/时/分/秒等时间寄存器。
3. RTC 的启动；对 RTCSRT 寄存器写固定值 55H，使能 RTC 模块。
4. 检查实时时钟是否成功启动。进行启动操作时，会检查设置的时间数据是否合法。若合法，RTC 状态显示寄存器（RTCSTU）的 LD 位为 1，表示启动成功；若不合法，时间寄存器中的时间不会载入 RTC 中。
5. 实时时钟硬件自动载入新值并开始计时。

### 17.3.3 RTC 时间设置

#### RTC 实时时钟时间设置方法：

1. 查询 RTOFF（RTCSTU<1>）位，直到 RTOFF（RTCSTU<1>）的值变为 1；
2. 置 CNF（RTCSTU<2>）位为 1，进入配置模式；
3. 对一个或多个 RTC 闹钟中断时间寄存器或者实时时钟寄存器进行写操作；
4. 清 0 CNF（RTCSTU<2>）位，退出配置模式；此时写操作才能进行。
5. 重新查询 RTOFF（RTCSTU<1>）位，直至 RTOFF（RTCSTU<1>）位变为 1 以确认写操作已经完成。

注：仅当 CNF 标志位被清 0 时，写操作才能进行。

必须置 CNF（RTCSTU<2>）位为 1，使 RTC 进入配置模式后，才能对实时时钟寄存器，闹钟中断时间寄存器等写入操作。

另外，对 RTC 任何寄存器的写操作，都必须在前一次写操作结束后进行。可以通过查询 RTCSTU 寄存器中的 RTOFF 状态位，判断 RTC 寄存器是否处于更新中。仅当 RTOFF 状态位是 1 时，才可以写入 RTC 时间寄存器。

### 17.3.4 RTC 闹钟功能

RTC 模块提供星期、小时、分和秒的闹钟功能，用户可根据实际需求，通过配置 RTICALRW、RTICALRH、RTICALRM 和 RTICALRS 寄存器进行闹钟时间设定；每个时间进程均提供一位独立的使能位（ALRWEN、ALRHEN、ALRMEN 和 ALRSEN），可满足各种不同的配置需求。

RTCSTU 寄存器的 RTICALREN 位为闹钟功能的总使能位，将该位置 1 后将使能闹钟功能，当实时时钟寄存器中的时间和闹钟中断时间寄存器相同时，将产生一个闹钟中断，ALRIF 位置 1。

闹钟功能原理框图如下图所示。

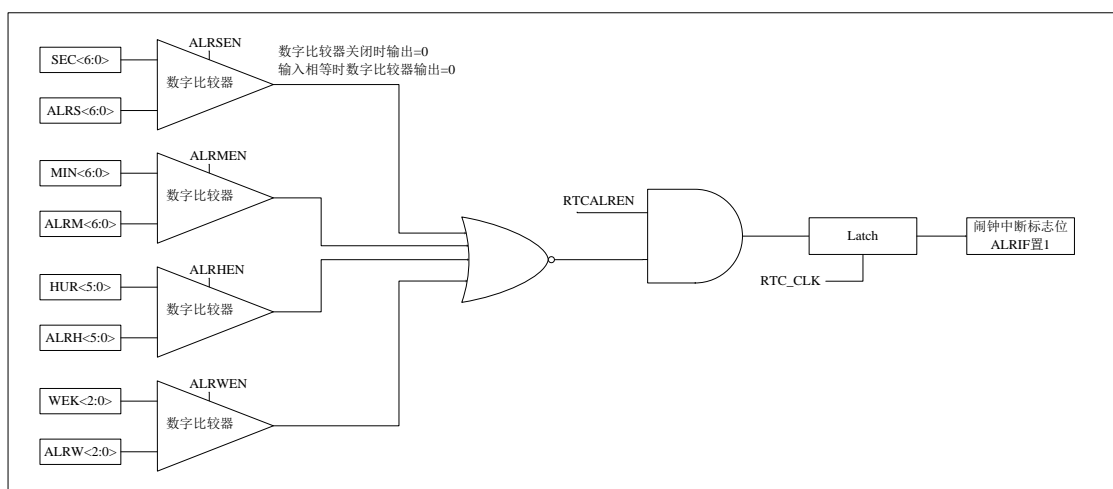


图 17.2 RTC 闹钟功能原理框图



### 17.3.5 时钟校正功能

时钟校正功能是为了实现高精度的时钟功能，校正因振荡频率的偏差而导致的时钟的提前或滞后的功能。进行校正时，并不调节石英振荡器的频率，而是利用分频电路的一部分对时钟脉冲进行调整。校正工作每20秒（或60秒）进行一次。最小分解能约为在大约3 ppm（或约1 ppm）的条件下，可以在-195.3 ppm~+192.2 ppm（或是-65.1 ppm~+64.1 ppm）的范围内进行校正。此设定可在时钟校正寄存器上进行。另外，不使用时钟校正功能时，请务必将其设定为“00 h”。

表 17.2 时间校正参考

项目	B7=0	B7=1
校正工作	每20秒	每60秒
最小分解能	3.052ppm	1.017ppm
校正范围	-195.3ppm~+192.2ppm	-65.1ppm~+64.1ppm

#### 17.3.5.1 计算方法

1. 当前振荡频率 > 目标频率时（时间提前时）

$$\text{校正值} = 128 - \text{整数} \left\lfloor \frac{(\text{当前振荡频率实测值}) - (\text{目标振荡频率})}{(\text{当前振荡频率实测值}) \times (\text{最小分解能})} \right\rfloor$$

注：此计算值的“0~64”范围为可校正范围。

2. 当前振荡频率 < 目标频率时（时间滞后时）

$$\text{校正值} = \text{整数} \left\lceil \frac{(\text{目标振荡频率}) - (\text{当前振荡频率实测值})}{(\text{当前振荡频率实测值}) \times (\text{最小分解能})} \right\rceil + 1$$

注：此计算值的“0~62”范围为可校正范围。

#### 17.3.5.2 校准原理

根据实际测出的频率，利用分频电路的一部分对时钟脉冲进行增加或者减少。

1. 当前振荡频率 > 目标频率时（时间提前时）

$$\text{校正值} = 128 - \text{整数} \left\lfloor \frac{(\text{当前振荡频率实测值}) - (\text{目标振荡频率})}{(\text{当前振荡频率实测值}) \times (\text{最小分解能})} \right\rfloor$$

例：校准寄存器值为 0111 1101（t=1/32768）

表 17.3 校准例图（校准周期为 20s，延迟时间，校准大小为 3）

	校准值为 0111 1101	无校准
分频计数器计数	01 00 0000 0	01 00 0000 0
	00 11 1101 1 +6t	01 00 0000 1
	00 11 1110 0 +6t	01 00 0001 0
	00 11 1110 1 +6t	01 00 0001 1

在 20s 校准周期中增加了 6 个 t (1/32768), 延迟时间。

20 秒校准 1 次增加的 t = 校准寄存器<B6:B0>补码值\*2t

2. 当前振荡频率<目标频率时 (时间滞后时)

$$\text{校正值} = \text{整数} \left( \frac{(\text{目标振荡频率}) - (\text{当前振荡频率实测值})}{(\text{当前振荡频率实测值}) \times (\text{最小分解能})} \right) + 1$$

例：校准寄存器值为 0000 0011 (t=1/32768)

表 17.4 校准例图 (校准周期为 20s, 加速时间, 校准大小为 3)

	校准值为 0000 0011	无校准
分频计数器计数	01 00 0000 0	01 00 0000 0
	01 00 0011 1      -6t	01 00 0000 1
	01 00 0100 0      -6t	01 00 0001 0
	01 00 0100 1      -6t	01 00 0001 1

在 20s 校准周期中减去了 6 个 t (1/32768), 加速时间。

20 秒校准 1 次减去的 t = 校准寄存器值<B6:B0>\*2t

### 17.3.5.3 校准寄存器设定值表

表 17.5 校准寄存器设定值表 1 (最小分解能=3.052ppm)

B7=0, 每20秒校准一次 最小分解能=3.052ppm								校准值 (ppm)	精度 (sec/day)
B7	B6	B5	B4	B3	B2	B1	B0		
0	0	1	1	1	1	1	1	192.3	16.61
0	0	1	1	1	1	1	0	189.2	16.35
0	0	1	1	1	1	0	1	186.2	16.09
...								...	...
0	0	0	0	0	0	1	0	6.1	0.53
0	0	0	0	0	0	0	1	3.1	0.26
0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	-3.1	-0.26
0	1	1	1	1	1	1	0	-6.1	-0.53
...								...	...
0	1	0	0	0	0	1	1	-186.2	-16.09
0	1	0	0	0	0	1	0	-189.2	-16.35
0	1	0	0	0	0	0	1	-192.3	-16.61
0	1	0	0	0	0	0	0	-195.3	-16.88

表 17.6 校准寄存器设定值表 2 (最小分解能=1.017ppm)

B7=1, 每60秒校准一次 最小分解能=1.017ppm								校准值 (ppm)	精度 (sec/day)
B7	B6	B5	B4	B3	B2	B1	B0		
1	0	1	1	1	1	1	1	64.1	5.54

1	0	1	1	1	1	1	0	63.1	5.45
1	0	1	1	1	1	0	1	62.0	5.36
...								...	...
1	0	0	0	0	0	1	0	2.0	0.18
1	0	0	0	0	0	0	1	1.0	0.09
1	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	-1.0	-0.09
1	1	1	1	1	1	1	0	-2.0	-0.18
...								...	...
1	1	0	0	0	0	1	1	-62.0	-5.36
1	1	0	0	0	0	1	0	-63.1	-5.45
1	1	0	0	0	0	0	1	-64.1	-5.54
1	1	0	0	0	0	0	0	-65.1	-5.62

### 17.3.6 12/24 小时模式选择

通过 RTCSTU 寄存器的 HT 位可对 12/24 小时模式进行选择。

HT: 选择小时显示的类型 (hour type)

- 0: 小时显示为 24 小时制
- 1: 小时显示为 12 小时制

### 17.3.7 闰年显示功能

RTCSTU 寄存器的 LIF 位可显示当前年份是否为闰年。

LIF: 闰年指示标志 (Leap indication flag)

- 0: 当前年份为平年
- 1: 当前年份为闰年

### 17.3.8 RTC 定时器

实时时钟模块内置 2 个可级联的 8 位定时器 TMR0 和 TMR1。置位 RTCTMREN 寄存器的 TMRxEN 位来使能定时器 TMRx；通过设置 RTCTMRCTL 寄存器的 CKSTx<3:0>来选择 RTC 定时器 x 的时钟源；寄存器 RTCTMRx 保存定时器的初值。

TMR0 和 TMR1 为向下计数，当定时器发生下溢时（计数值=00H），RTCTxIF 置 1，如果使能 RTCTxIE 位，程序将进入中断行。

TMR0 和 TMR1 带有重载功能，即定时器计数到 00H 后，将会重新载入 RTCTMRx 寄存器的值；如果在计数过程中对 RTCTMRx 寄存器进行修改，不会影响当前计数周期，定时器会在下溢发生后，载入 RTCTMRx 寄存器的新值并开始一个新的计数周期。

将 RTCTMRCTL 寄存器的 CKST1<3:0>配置为 1000H 时，将使能 TMR0 和 TMR1 的级联，TMR0 为低 8 位，TMR1 为高 8 位，组成一个 16 位定时器；通过 CKST0<3:0>位选择计数时钟；TMR0EN 位和 TMR1EN 位需要同时置 1；两个中断标志位 RTCT0IF 和 RTCT1IF 均有效，用户可任选其一。

TMR0 和 TMR1 定时器的实时计数寄存器不可读。

定时器的操作如下：

1. 通过设置 RTCTMRCTL 寄存器的 CKSTx<3:0>选择定时器时钟源及是否级联；
2. 向 RTCTMRx 寄存器写入定时器初值；
3. 置位 RTCTMREN 寄存器的 TMRxEN 位使能定时器；
4. 通过设置相应的定时器中断使能位来使能/禁止定时器中断。

### 17.3.9 RTC 工作在休眠模式

使能 RTC 模块后，单片机进入休眠模式，RTC 模块将继续正常运行。任何 RTC 中断（时间节拍中断、闹钟中断、进程中断和内置定时器中断）在使能后都可将单片机从休眠模式下唤醒。

### 17.3.10 RTC 中断

RTC模块提供如下丰富的中断源供用户使用。

- 闹钟中断
- 节拍中断
- 进程中断：秒中断、分中断、时中断和日中断
- 内置定时器中断：定时器0、定时器1

注1: RTC中断信号持续时间约为16us，在该时间内，RTC中断标志位将无法清零。用户在使用RTC中断时需注意中断清零指令是否已在该时间外；或者在执行清零中断标志位指令操作之后，添加判断是否已将RTC中断标志位清零，以避免重复触发中断条件。

注2: RTC进程中断发生后，如需对RTC实时时钟寄存器进行读出操作，请在读出操作之前增加8ms的延时，否则可能引起RTC实时数据异常的问题。

#### 17.3.10.1 闹钟中断时间设置

闹钟中断时间设置方法：

1. 清 0 闹钟中断使能位 ALRIE (<EIE3.5>)；
2. 设置闹钟中断时间寄存器（星期/时/分/秒）；
3. 将 RTCSTU 寄存器的 RTCALREN 位置 1，使能 RTC 闹钟功能；
4. 清零 ALRIF 位 (<EIF3.5>)；
5. 置 1 闹钟中断使能位 ALRIE (<EIE3.5>)；
6. 达到闹钟时间，产生闹钟中断并进入中断行；
7. 可以通过置 1 IPEN(PCTL<3>)位，如果置 1 闹钟中断优先级控制位 PALR，则闹钟中断为高优先级，此时使 AIEH 置 1 才响应高优先级中断。如果清 0 闹钟中断优先级控制位 PALR，则闹钟中断为低优先级，此时使 AIEH 置 1 且 AIEL 置 1 才响应低优先级中断。

将闹钟中断时间按星期，时，分，秒分别写入 RTCALRW、RTCALRH、RTCALRM 和

RTCALRS 寄存器中，并将对应的 ALRWEN、ALRHEN、ALRMEN 和 ALRSEN 位置 1，用户可根据实际需求使能、禁止对应时间的中断；RTCSTU 寄存器的 RTCALREN 位为闹钟功能的总使能位，在配置好闹钟时间寄存器后，将该位置 1 以使能闹钟功能。

当实时时钟寄存器等于闹钟中断寄存器中的设定值，则闹钟中断标志 ALRIF 被置 1。该位可由软件清零，上电自动清零。闹钟中断使能位置 1，可使能得到想要的闹钟中断。

注：闹钟中断小时寄存器中的 12/24 选择需和实时时钟寄存器设置一致；

### 17.3.10.2 周期时间节拍中断

周期时间节拍中断有 8 个选项：1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 以及 1 秒，通过设置 RTCTTR<2:0>来进行选择。当时间节拍中断使能位 TTIE 置 1 时，可产生周期时间节拍中断。如果使能 IPEN(PCTL<3>)位置 1，如果置 1 TTIP 优先级控制位，则周期节拍中断为高优先级，此时使 AIEH 置 1 才响应高优先级中断。如果清 0 TTIP 优先级控制位，则周期节拍中断为低优先级，此时使 AIEH 置 1 且 AIEL 置 1 才响应低优先级中断。

表 17.7 时间节拍设置

RTCTTR[2:0]	时间节拍(秒)
000	1
001	1/2
010	1/4
011	1/8
100	1/16
101	1/32
110	1/64
111	1/128

### 17.3.10.3 进程中中断

通过 RTCIE 寄存器的 SECIE、MINIE、HURIE 或 DAYIE 4 个使能端分别控制秒、分、时或天的进程中中断。当实时时间达到设置进程中中断时间且进程使能端置 1，则产生相应的进程时间中断。如果使能 IPEN(PCTL<3>)位置 1，如果置 1 优先级控制位，则进程中中断为高优先级，此时使 AIEH 置 1 才响应高优先级中断。如果清 0 优先级控制位，则进程中中断为低优先级，此时使 AIEH 置 1 且 AIEL 置 1 才响应低优先级中断。

### 17.3.10.4 内置定时器中断

RTC 内置的两个定时器均配备中断，中断标志位为 EIF7 寄存器的 RTCT1IF 及 RTCT0IF。当 RTC 内置定时器溢出时，相应的中断标志位置 1。如果外设中断使能且 EIE7 寄存器的 RTCT1IE/RTCT0IE 置 1，则当 RTC 内置定时器溢出时，程序将进入中断行；如果 MCU 在休眠状态下，中断将唤醒 MCU。内置定时器详细描述见 [RTC 定时器章节](#)。

## 18 看门狗定时器

### 18.1 独立看门狗 IWDT

为了防止单片机在正常工作时程序跑飞，KF8A100 提供一个看门狗定时器，单片机正常工作时，当看门狗定时器定时时间达到超时时间后，会使单片机产生复位。

看门狗定时器使用片内看门狗专用 RC 振荡器，因此它无需外接任何器件，在休眠模式仍能正常运行。在正常运行时，WDT 超时事件将使单片机产生一次复位 WDT 超时事件将清零和 T0 公用的 8 位分频器。如果单片机处于休眠模式，WDT 超时事件将唤醒单片机并使其继续执行 IDLE 后面的指令。通过将配置位 WDTE 清 0/置 1，可关闭/打开 WDT。如果配制位 WDTE 清 0，则可以通过寄存器 SWDTE(PCTL<2>)软件清 0/置 1，可关闭/打开 WDT。

#### WDT 周期:

如 WDT 框图所示：WDT 的时钟源为内部低频时钟（带校正功能的 32K），由 WDTCTL 寄存器的 WDTPS<3:0>选择预分频比（1:32 到 1:65536），共 12 档分频比。复位后 WDTPS<3:0>=0100，WDT 时基为 16ms。超时时间最小约为 1ms，最长约为 268s。由于温度、电源电压和工艺等的差异，不同器件之间的超时周期稍有不同。通过软件将 OPCR 寄存器的 PSA 位置 1，还可将 T0 预分频器分配给 WDT。设置 PS<2:0>选择预分频器的分频比,分频比可选择为:1/1、1/2、1/4、1/8、1/16、1/32、1/64、1/128。

为了防止在正常工作时看门狗超时复位，要在固定的时间内对看门狗定时器清 0。执行 CWDT 和 IDLE 指令后会将 WDT 和预分频器清 0，当看门狗定时器出现超时，状态字寄存器 PSW 中的  $\overline{TO}$  位将被清 0。

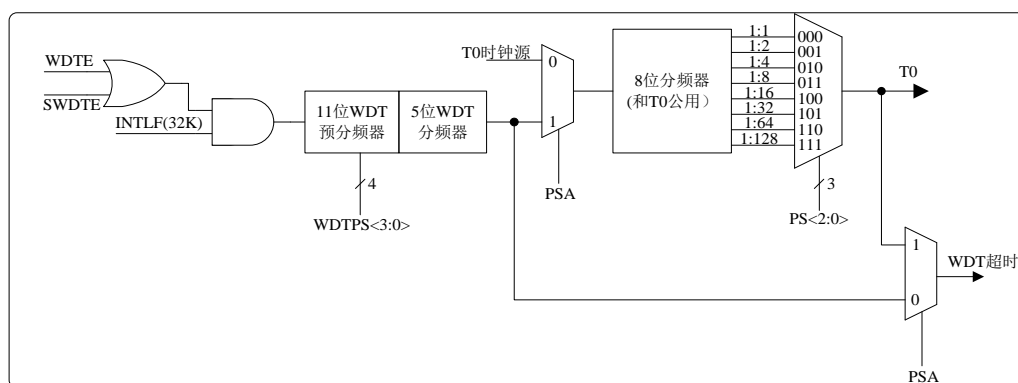


图 18.1 看门狗定时器框图

#### 18.1.1 WDT 预分频选择寄存器 WDTPS

寄存器： WDTPS: WDT预分频选择寄存器(地址:40H)

复位值 ---- 0100	bit7							bit0
-	-	-	-	WDTPS3	WDTPS2	WDTPS1	WDTPS0	
	R/W	U	U	U	R/W	R/W	R/W	R/W

WDTPS<3:0>:看门狗定时器预分频比选择位

0000 = 1: 32
0001 = 1: 64
0010 = 1: 128
0011 = 1: 256
0100 = 1: 512 (默认)
0101 = 1: 1024
0110 = 1: 2048
0111 = 1: 4096
1000 = 1: 8192
1001 = 1: 16384
1010 = 1: 32768
1011 = 1: 65536
11xx = 保留

## 18.2 窗口看门狗模块 WWDT

窗口看门狗通常被用来监测由外部干扰或不可预见的逻辑条件造成的应用程序背离正常的运行序列而产生的软件故障。

WWDG最适合那些要求看门狗在精确计时窗口起作用的应用程序。通过可配置的时间窗口来检测应用程序非正常的过迟或过早的操作。

**特点:**

- 可编程自由运行计数;
- 时钟为内部低频时钟INTLF;
- 可编程预分频;
- 提供中断;
- 条件避免复位: 窗口内写计数器; 利用中断。

### 18.2.1 窗口看门狗相关寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
115H	WCTL	WWDTE N	-	-	WPS4	WPS3	WPS2	WPS1	WPS0
114H	WCNT	-	WT6	WT5	WT4	WT3	WT2	WT1	WT0
113H	WCFR	-	WW6	WW5	WW4	WW3	WW2	WW1	WW0

#### 18.2.1.1 窗口看门狗控制寄存器 WCTL

**寄存器:** WCTL: 窗口看门狗控制寄存器(地址:115 H)

		bit7						bit0	
复位值 0-0 0000	WWDTEN	-	-	WPS4	WPS3	WPS2	WPS1	WPS0	
	R/W	U	U	R/W	R/W	R/W	R/W	R/W	R/W

WWDTEN: 窗口看门狗使能位  
 0 = 禁止窗口看门狗功能  
 1 = 使能窗口看门狗功能

WPS<4:0>: 窗口看门狗预分频位  
 00000/10011-11111 = WWDTCCLK=INTLF  
 00001 = WWDTCCLK=INTLF/2  
 00010 = WWDTCCLK=INTLF/4  
 00011 = WWDTCCLK=INTLF/8  
 00100 = WWDTCCLK=INTLF/16  
 00101 = WWDTCCLK=INTLF/32  
 00110 = WWDTCCLK=INTLF/64  
 00111 = WWDTCCLK=INTLF/128  
 01000 = WWDTCCLK=INTLF/256  
 01001 = WWDTCCLK=INTLF/512  
 01010 = WWDTCCLK=INTLF/1024  
 01011 = WWDTCCLK=INTLF/2048  
 01100 = WWDTCCLK=INTLF/4096  
 01101 = WWDTCCLK=INTLF/8192  
 01110 = WWDTCCLK=INTLF/16384  
 01111 = WWDTCCLK=INTLF/32768  
 10000 = WWDTCCLK=INTLF/65536  
 10001 = WWDTCCLK=INTLF/131072  
 10010 = WWDTCCLK=INTLF/262144

### 18.2.1.2 窗口看门狗计数寄存器 WCNT

寄存器: WCNT: 窗口看门狗计数寄存器(地址:114 H)

复位值 -000 0000	bit7							bit0
	-	WT6	WT5	WT4	WT3	WT2	WT1	WT0
	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

WT<6:0>: 窗口看门狗计数位

### 18.2.1.3 窗口看门狗配置寄存器 WCFR

寄存器: WCFR: 窗口看门狗配置寄存器(地址:113 H)

复位值 -000 0000	bit7							bit0
	-	WW6	WW5	WW4	WW3	WW2	WW1	WW0
	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

WW<6:0>: 窗口看门狗配置位



18.2.2 窗口看门狗原理

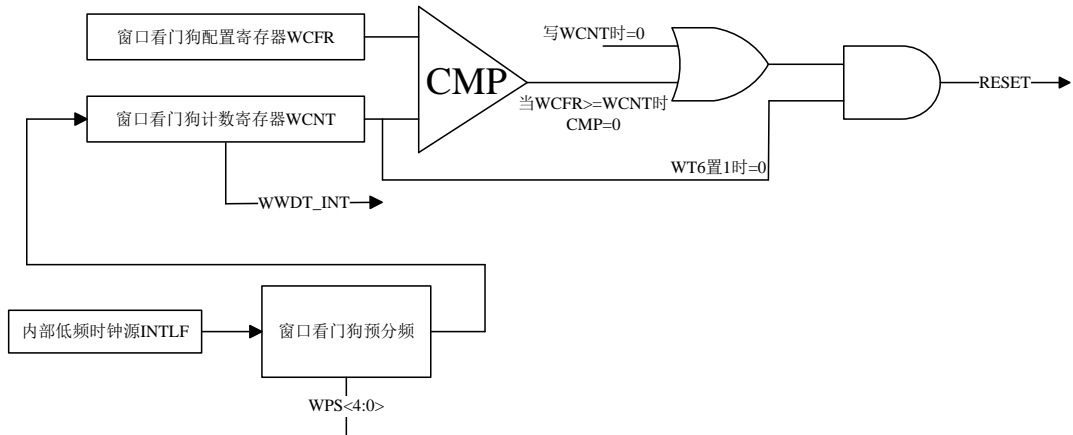


图 18.2 窗口看门狗原理框图

窗口看门狗的7位计数器的时钟源为内部低频振荡器，配备5位预分频位，可以为窗口看门狗较宽的时钟频率选择。在使用窗口看门狗时，需要通过配置LFCKCTL寄存器的LFCKEN位（置1）和LFCKS位（清0）将INTLF打开，否则无法正常工作。

注：LFCKCTL寄存器的LFCKDIV<2:0>位对WWDT无效。

窗口看门狗产生复位信号有两种情况：WCNT寄存器的WT6位置1；当WCFR>WCNT时对WCNT进行写操作。即以配置寄存器中的值为准，当过早得对WCNT寄存器进行操作（WCNT<WCFR）和过迟操作（WT6置1），窗口看门狗WWDT都会产生复位信号。

避免窗口看门狗发生复位，应当在何时的时间段内（WCFR<WCNT<40H）对WCNT寄存器进行清0（写操作）。

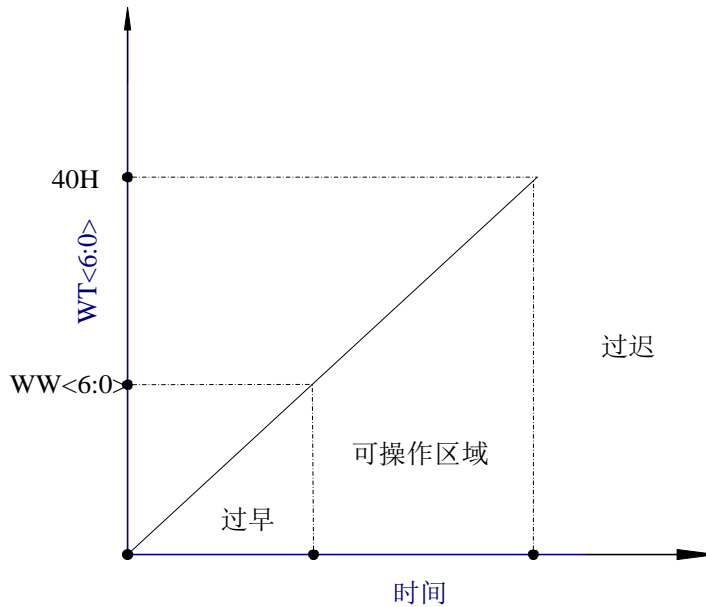


图 18.3 窗口看门狗原理图

### 18.2.3 窗口看门狗中断

窗口看门狗模块还提供一个 WWDT 中断，用户也可以利用此中断来避免窗口看门狗发生复位。窗口看门狗中断会在窗口看门狗计数器 WCNT 计数到 3FH 时产生，用户可以通过使能该中断，在程序的中断行对 CNT 寄存器清 0 以防止复位。（WDT 计数时钟最高为 32KHZ，满足 WDT 从 40H 跳到 39H 即一个计数时钟内执行完中断指令的条件即可）。

## 19 复位

KF8A100 具有:上电复位(POR)、WDT 复位、RST 复位和欠压检测复位(LVR)四种复位方式, KF8A100 带有两个看门狗模块:独立看门狗 IWDT 和窗口看门狗 WWDT。

有些寄存器的状态在任何复位条件下都不会受到影响, 上电复位时它们的状态不定, 而在其它复位发生时其状态将保持不变。其它大多数寄存器在复位事件发生时将被复位成“复位状态”。图 19.1 给出了片内复位电路的简化结构方框图。

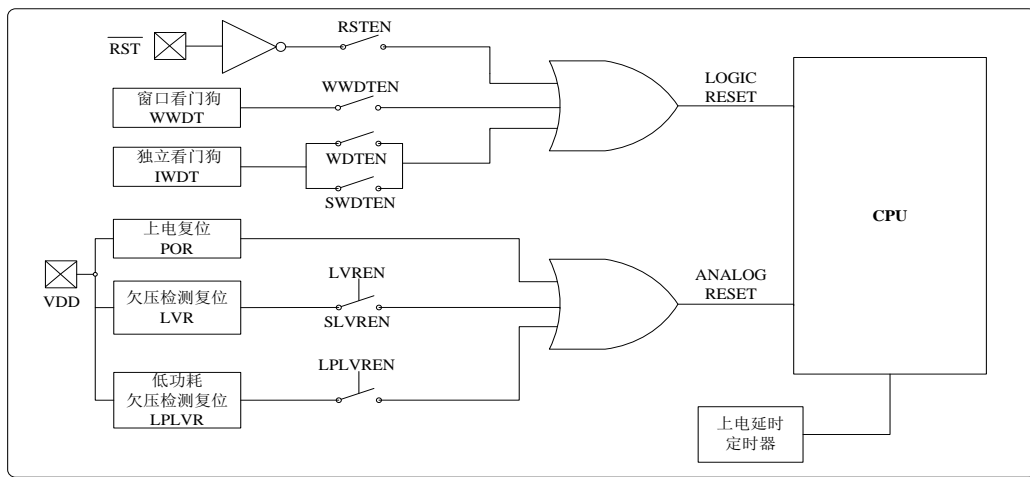


图 19.1 片内复位电路简化框图

注:上电延时定时器只对上电复位(POR)和欠压复位(LVR)有效。

## 19.1 电源控制状态寄存器(PCTL)

如寄存器 PCTL 所示， $\overline{\text{LVR}}$  位的状态在单片机上电复位时是不确定的。如果用户在使用中要用到该位，在程序初始化部分需将其置 1，随后如果有复位发生且  $\overline{\text{LVR}}=0$ ，则表示发生过欠压检测复位。 $\overline{\text{LVR}}$  状态位是“无关”位，如果欠压检测电路被关闭(通过设定配置字中的 LVREN 位和 PCTL 中的 SLVREN 位)， $\overline{\text{LVR}}$  状态位是不可预知的。 $\overline{\text{POR}}$  是上电复位状态位，该位在上电复位时被清 0，在其它情况下不受影响。

**寄存器: PCTL: 电源控制寄存器(地址:2EH)**

	bit7						bit0	
复位值 --01 00xx	-	-	-	SLVREN	IPEN	SWDTEN	$\overline{\text{POR}}$	$\overline{\text{LVR}}$
	U	U	R/W	R/W	R/W	R/W	R/W	R/W

- SLVREN: 软件欠压检测使能位  
1 = 使能欠压检测  
0 = 禁止欠压检测
- IPEN: 中断优先级控制位  
1 = 使能中断优先级功能  
0 = 禁止中断优先级，即为普通模式
- SWDTEN: 软件独立看门狗定时器使能位  
当配置字的 WDTEN=0 时  
1 = 软件使能独立看门狗定时器  
0 = 软件禁止独立看门狗定时器
- $\overline{\text{POR}}$ : 上电复位状态位  
1 = 未发生上电复位  
0 = 发生了上电复位
- $\overline{\text{LVR}}$ : 欠压复位状态位  
1 = 未发生欠压复位  
0 = 已发生欠压复位

## 19.2 上电复位(POR)

在 VDD 达到适合单片机正常工作的电平之前，片内上电复位电路使单片机保持在复位状态，直到 VDD 达到正常工作电平之后，根据配置位 PWRT，单片机将开始正常工作，或者延时 32ms 左右之后再开始工作。

## 19.3 WDT 复位

KF8A100 带有两个看门狗模块：独立看门狗 IWDT 和窗口看门狗 WWDT，详细描述见看门狗定时器章节。

看门狗定时器有一个独立的时钟源，因此单片机在正常工作和休眠模式下都可以正常工作。在单片机正常工作且打开看门狗后，当看门狗计数器计满后产生溢出，将使单片机（RTC 除外）复位。

在休眠模式下，WDT 也可以正常工作，当 WDT 定时器计满溢出后，将会使单片机从休眠模式唤醒转入正常工作模式，在休眠模式不会对各寄存器复位。

## 19.4 RST 复位

使能外部 RST 复位（配置位 RSTEN=1）后，当引脚 P0.0/ $\overline{\text{RST}}$  输入复位信号，不管单片机工作在正常模式还是休眠模式，均会使单片机（RTC 除外）复位。通过在编程时将 P0.0 引脚配置为  $\overline{\text{RST}}$  复位引脚，即可打开  $\overline{\text{RST}}$  复位。复位时，KF8A100 器件有一个噪声滤波器用于检测和滤除小脉冲干扰新哈，图 25.2 是  $\overline{\text{RST}}$  复位推荐外部电路。

注意，当使能外部 RST 复位（配置位 RSTEN=1）后，P0.0 口将被分配为外部复位信号输入引脚；同时，P0.3 口的数字输入功能将被芯片屏蔽。

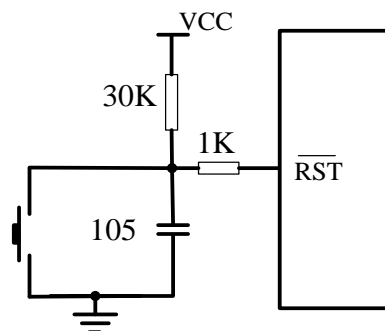


图 19.2 建议  $\overline{\text{RST}}$  复位电路

## 19.5 欠压检测复位(LVR)

KF8A100 系列中的单片机片内配备一个欠压检测复位电路 (LVR) 和一个低功耗欠压检测复位电路 (LPLVR)。通过编程时设定配置位中的 LVREN 位 (LPLVREN 位) 可以禁止/使能(清 0/置 1)欠压检测复位电路 (低功耗欠压检测复位电路), 单片机会按照配置位中的 LVREN 位和 LPLVREN 位的配置, 选择 LVR 和 LPLVR 是否使能, 在程序运行过程中, 如果用户对 PCTL 中的 SLVREN 位进行清零 (或写 0)/置 1 的操作, 则单片机将按照 SLVREN 位的状态来决定 LVR 是否使能。

表 19.1 欠压检测复位 LVR 的配置

配置位 LVREN	PCTL_SLVREN	LVR 是否被使能
0	软件未配置	禁止
0	软件清零	禁止
0	软件置 1	LVR 被软件使能
1	软件未配置	LVR 默认被使能
1	软件清零	禁止
1	软件置 1	LVR 默认被使能

表 19.2 低功耗欠压检测复位 LPLVR 的配置

配置位 LPLVREN	LPLVR 是否被使能
0	禁止
1	使能

LVR (LPLVREN) 的配置方式实际为由配置位 LVREN (LPLVREN) 位写 0/写 1 决定单片机上电后 LVR 是禁止/使能; 程序运行后, 如果不对 PCTL 寄存器的 SLVREN 位进行配置, 则 LVR 继续由配置位决定, 一旦对 PCTL 寄存器的 SLVREN 位进行操作, 则 LVR 将由 PCTL 寄存器的 SLVREN 位写 0/写 1 决定是禁止/使能。

如果  $V_{DD}$  跌落至  $V_{LVR}(V_{LVR}=2.1V)$  以下且持续时间大于  $T_{LVR}$  ( $T_{LVR}$  大于 10us), 欠压检测电路将使单片机复位, 单片机保持复位状态直到  $V_{DD}$  上升到  $V_{LVR}$  以上, 此时上电延时定时器启动, 使器件在随后 70ms 左右的延时时间处于延时复位状态, 过了 70ms 以后单片机开始正常工作。

如果  $V_{DD}$  跌落至  $V_{LVR}$  以下的时间小于规定参数( $T_{LVR}$ ), 将不保证可产生复位。

如果在上电延时定时器运行过程中发生  $V_{DD}$  跌落至  $V_{LVR}$  以下的情况, 器件将返回欠压检测复位状态且上电延时定时器被重新初始化。直到  $V_{DD}$  上升至  $V_{LVR}$  以上时, 上电延时定时器启动一个 70ms 的复位延时, 如在延时器件没有欠压发生, 单片机会退出复位状态开始正常工作。

低功耗欠压检测复位电路 (LPLVR) 原理和欠压检测复位电路(LVR)一致, 详细特性见电气规范章节。

## 19.6 上电延时定时器

KF8A100 配备一个可控上电延时定时器, 用户可以通过配置位 PWRT 选择是否使用该

上电延时定时器。上电延时定时器的时钟源为内部低频振荡器。

当关闭时，在器件上电复位或欠压检测复位发生后，单片机将立即退出复位状态开始正常工作；

当打开时，在器件上电复位或欠压检测复位发生后，上电延时定时器将提供一个长度为 32ms 的固定延时时间，该时间内单片机将保持复位状态，之后才退出复位状态开始正常工作。

由于 VDD、温度、制造工艺、内部振荡器频率等的变化，不同单片机的上电延时时间有所差异。

## 19.7 不同复位条件下对寄存器的影响

表 19.3 不同复位条件下对标志位的影响

$\overline{\text{POR}}$	$\overline{\text{LVR}}$	$\overline{\text{T0}}$	$\overline{\text{PD}}$	复位方式
0	u	1	1	上电复位
1	0	1	1	欠压检测复位
u	u	0	u	WDT 复位
u	u	0	0	WDT 唤醒
u	u	u	u	正常操作中的 $\overline{\text{RST}}$ 复位
u	u	1	0	休眠模式中的 $\overline{\text{RST}}$ 复位

图注: u=未发生变化

## 20 电源管理和功耗模式

### 20.1 电源

KF8A100 电源		
名称	电压范围	注释
VDD	2.7~5.5 V	供电电源
VSS	0	通用地线

VDD 为芯片的主供电电源，VSS 为芯片的通用地线。VDD 为 I/O 口、内部 LDO 供电。使用时，要求所有的电源口 VDD 全部连接到同一供电电源，所有的地口 VSS 连接到同一地线。

KF8A100 的工作电压为 2.7V~5.5V；

#### 20.1.1 内置电压调节器

KF8A100 内置电压调节器，在不同工作模式下，电压调节器的工作状态如下表所示：

表 20.1 内置电压调节器工作状态表

工作模式	电压调节器
运行模式	正常功耗模式运行
普通休眠模式	正常功耗模式运行
深度休眠模式	低功耗模式运行

### 20.2 电源管理

KF8A100 系列单片机有一套 POR 和 LVR 电路，用于主供电电源；详细描述见复位章节相关内容。关于 POR、LVR 的细节请参考数据手册的电气特性相关章节。

#### 20.2.1 被封区读写配置寄存器

用户在对备份区相关寄存器进行操作前，需要先对备份区配置寄存器进行配置，否则无法对备份区进行操作。

**寄存器：BKPCTL：备份区控制寄存器(地址:53H)**

复位值 0000 0000	bit7							bit0
	BKP7	BKP6	BKP5	BKP4	BKP3	BKP2	BKP1	BKP0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**BKP7:** 备份区接口配置位  
 0 = 备份区接口处于默认状态  
 1 = 备份区接口开放，可对备份区内寄存器进行读写操作



BKP<6:0>: 保留位

### 20.2.2 备份区寄存器的读写

Bank3 地址的 SFR 寄存器位于备份区内，在对备份区内 SFR 进行读写操作前，需要将 BKPCTL 寄存器的 BKP7 位置 1，否则将无法进行读写操作。

## 20.3 功耗模式

KF8A100 系列单片机提供丰富的功耗模式供用户选择，以满足用户对功耗的不同需求。KF8A100 提供如下功耗模式：

- 正常运行模式 (Normal run mode, NR)
- 普通休眠模式 (Sleep mode, SLP)
- 深度休眠模式 (Deep sleep mode, DSLP)

### 20.3.1 功耗模式相关寄存器

表 20.2 功耗模式相关寄存器表

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
360H	POWCTL	保留	保留	保留	保留	保留	保留	DSL PEN	保留
364H	PHCLR	保留	保留	保留	保留	保留	保留	EXTLFEN	INTLFEN
368H	BBODCTL	BBODEN	BSCAN	-	-	-	-	-	BBODF

#### 20.3.1.1 功耗模式控制寄存器

寄存器： **POWCTL**：功耗模式控制寄存器(地址:360H)

复位值 0000 0000	bit7							bit0	
	保留	保留	保留	保留	保留	保留	DSL PEN	保留	
	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	

DSL PEN： 深度休眠模式使能位  
0 = 禁止深度休眠模式  
1 = 使能深度休眠模式

注：保留位默认为 0，请勿将该位置 1。

#### 20.3.1.2 备份区外设复位寄存器

寄存器： **PHCLR**：备份区外设复位寄存器(地址:364H)

复位值 1100 0000	bit7						bit0	
	保留	保留	保留	保留	保留	保留	EXTLFEN	INTLFEN
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

EXTLFEN： 外部低频振荡器软件使能位  
0 = 软件未使能外部低频振荡器  
1 = 软件使能外部低频振荡器

INTLFEN： 内部低频振荡器软件使能位  
0 = 软件未使能内部低频振荡器  
1 = 软件使能内部低频振荡器

- 注 1.当外设时钟选择内部低频时钟或者外部低频时钟时，硬件将自动使能对应振荡器，即使振荡器软件使能位未置1。
- 2.保留位默认为0，请勿将该位置1。

### 20.3.2 正常运行模式

单片机退出上电复位后，处于正常运行模式（Normal run mode）下；MCU正常工作（系统时钟运行），所有外设均可使用，内部高频振荡器、内部低频振荡器、外部高频振荡器和外部低频振荡器均被允许使用。用户可以按照功能需求对MCU进行设置以达到最佳性能。

### 20.3.3 普通休眠模式

在如下配置条件下执行IDLE指令，单片机将进入普通休眠模式。

- BKPCTL\_bit0=1
- POWCTL\_DSLPEN=0

普通休眠模式下，系统时钟被禁止，CPU停止工作，程序停止运行；RAM保持；部分外设被禁止使用，部分外设允许使用内部低频时钟或者外部低频时钟继续工作。

表 20.3 普通休眠模式信息表

功耗模式	配置	状态
正常运行模式	BKPCTL_bit0=1 POWCTL_DSLPEN=0 POWCTL_ULPEN=0	系统时钟被禁止 CPU停止工作 程序停止运行 RAM保持

### 20.3.4 深度休眠模式

在如下配置条件下执行IDLE指令，单片机将进入深度休眠模式。

- BKPCTL\_bit0=1
- POWCTL\_DSLPEN=1

深度休眠模式与普通休眠模式在禁止和开放资源上相同，区别在于深度休眠模式有更低的功耗和更长的唤醒时间，详细数据请参考电气特性。

### 20.3.5 普通/深度休眠模式下 I/O 口注意事项

为使休眠状态下（普通休眠模式和深度休眠模式）的电流消耗降至最低，应使所有I/O口状态确定，如果有的端口没有使用，最好设置为输入，接到VDD或VSS上，或者打开弱上拉。如果没用的端口悬空，应设置为输出，以确保I/O引脚没有耗散电流产生，其他在休眠时不用的外设都要关闭。

### 20.3.6 普通/深度休眠模式唤醒方式

单片机进入普通/深度休眠模式后由于工作的需要，要将单片机从休眠状态唤醒，在 KF8A100 中可通过以下方式将单片机从休眠状态唤醒：

1. RST 引脚上输入的外部复位
2. 看门狗定时器唤醒(如果 WDT 已被使能)
3. INT0 内部中断
4. P0 口电平变化中断
5. 外设中断

RST 引脚输入的复位信号在唤醒单片机的同时也将导致单片机复位。其它唤醒时将单片机从休眠状态唤醒，并不会导致复位。可通过状态寄存器中的  $\overline{IO}$  和  $\overline{PD}$  位来确定单片机唤醒的原因。上电时  $\overline{PD}$  位将被置 1，而当器件从休眠状态唤醒时，该位将被清 0。 $\overline{IO}$  位则在 WDT 唤醒发生时被清 0。

在使用中断方式唤醒时，必须使能相应的中断使能位，唤醒与 AIE 位的状态无关。如果 AIE 位被清 0，单片机被唤醒后将继续执行 IDLE 指令后面的指令。如果 AIE 位被置 1，单片机执行 IDLE 指令后面一条指令后进入中断子程序。如果不希望执行 IDLE 指令后面的那条指令直接进入中断子程序，在 IDLE 指令加一条 NOP 指令即可。

## 21 电气规范

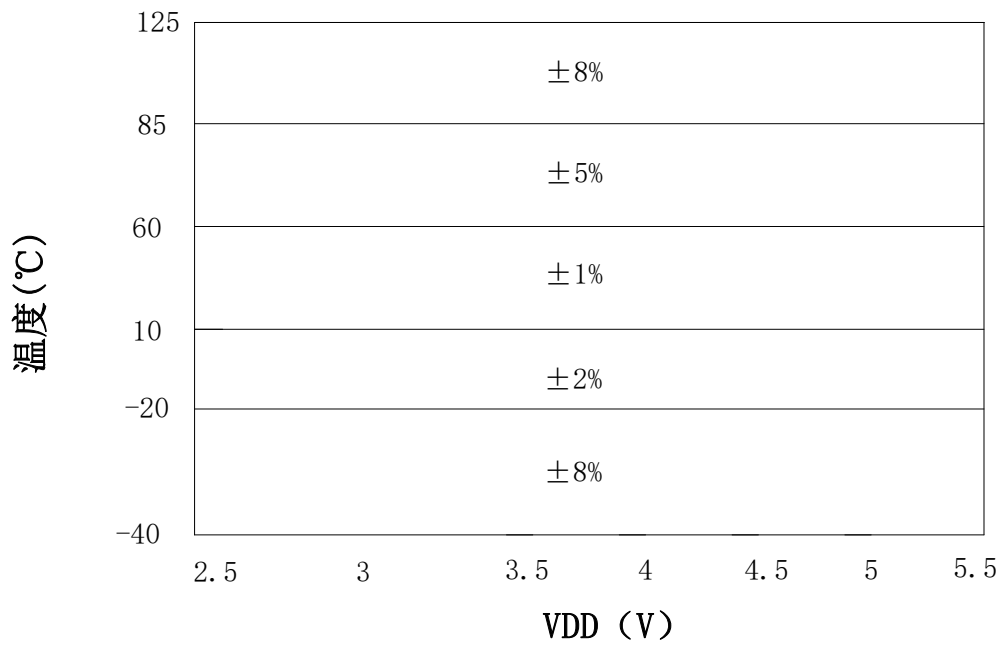
### 21.1 极限参数值

表 21.1 芯片极限参数值

极限参数值		
序号	参数说明	参数范围
1	偏置电压下的环境温度	-40°C~125°C
2	储存温度	-60°C~150°C
3	VDD 相对于VSS 的电压	5.5V
4	V <sub>PP</sub> 相对于VSS 的电压	12.5V
5	其它引脚相对于VSS 的电压	5.5V
6	VSS 引脚的最大输出电流	80mA
7	VDD 引脚的最大输入电流	80mA
8	任一I/O 引脚的最大输出灌电流	15mA
9	任一I/O 引脚的最大输出拉电流	15mA
10	I/O口 的最大灌电流	80mA
11	I/O口 的最大拉电流	80mA

备注:如果器件的工作条件超过“最大值”，可能会对器件造成永久性损坏。上述值仅为运行条件极大值，建议不要使器件在该规范规定的范围以外运行。器件长时间工作在最大值条件下，其稳定性会受到影响。

## 21.2 HFINTOSC 的频率精度与 VDD 和温度之间的关系



## 21.3 静态电流特性

表 21.2 芯片静态电流 ( $I_{DD}$ ) 特性

测试条件:25°C						
序号	测试条件		最小值	典型值	最大值	单位
	振荡频率	VDD(V)				
1	16MHz	5.0	—	1310	—	μA
		3.3	—	1310	—	
2	8MHz	5.0	—	780	—	
		3.3	—	780	—	
3	4MHz	5.0	—	520	—	
		3.3	—	520	—	
4	2MHz	5.0	—	380	—	
		3.3	—	380	—	
5	1MHz	5.0	—	320	—	
		3.3	—	320	—	
6	500kHz	5.0	—	285	—	
		3.3	—	285	—	
7	250kHz	5.0	—	265	—	
		3.3	—	265	—	
8	62.5kHz	5.0	—	255	—	
		3.3	—	250	—	

注 1: 在正常的工作模式下,  $I_{DD}$ 测量的条件为:所有I/O 引脚均设置为输出低,  $RST = V_{SS}$ , 禁止 WDT, 关闭时钟输出。

2: 供电电流主要随工作电压和频率而变化。其它因素, 如I/O 引脚负载和开关速率、内部代码执行模式和温度也会影响电流消耗。

## 21.4 休眠电流特性

表 21.3 芯片休眠电流 ( $I_{DD}$ ) 特性

测试条件:25°C							
序号	休眠模式	测试条件		最小值	典型值	最大值	单位
			VDD(V)				
1	普通休眠模式	WDT、BOR、比较器等外设被禁止	5.0	—	20	30	μA
			3.3	—	20	30	
2	深度休眠模式	WDT、BOR、比较器等外设被禁止	5.0	—	2.8	20	
			3.3	—	2.3	20	

## 21.5 外设电流特性

表 21.4 芯片外设电流特性

测试条件:25°C							
序号	测试参数	测试条件		最小值	典型值	最大值	单位
			VDD(V)				
1	欠电压复位电流 ( $I_{LVR}$ )	WDT、RTC 等外设被禁止	5.0	—	9.5	14.0	μA
			3.3	—	6.0	8.5	
2	比较器电流 ( $I_{CMP}$ )	使能单个比较器	5.0	—	44	62	
			3.3	—	24	35	
3	ADC	使能 ADC, 未转换	5.0	—	895	1350	
			3.3	—	765	1155	

- 注 1: 外设电流是基本 $I_{DD}$  或 $I_{PD}$  电流以及相应外设使能时消耗的额外电流的总和。外设电流可以从此电流中减去基本 $I_{DD}$  或 $I_{PD}$  电流得出。
- 2: 休眠电流与振荡器类型无关。掉电电流是在器件休眠时, 所有I/O 引脚设置为输出低,  $RST = V_{SS}$ ; 禁止WDT, 关闭时钟输出时测得的。
- 3: 外设电流还可能受到温度的影响。



## 21.6 I/O 端口电平和芯片供电电压特性

表 21.5 芯片 IO 端口电平特性

测试条件（特别说明除外）： 工作温度 -40°C~125°C						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
V <sub>IL</sub>	输入低电平 I/O引脚采用TTL缓冲器		V <sub>SS</sub>	—	0.15V <sub>DD</sub>	V
	采用施密特缓冲触发器		V <sub>SS</sub>	—	0.2V <sub>DD</sub>	V
V <sub>IH</sub>	输入高电平 I/O端口采用TTL缓冲器		V <sub>DD</sub> -0.6	—	V <sub>DD</sub>	V
	采用施密特缓冲触发器		0.8V <sub>DD</sub>	—	V <sub>DD</sub>	V
V <sub>OL</sub>	输出低电压		—	—	0.6	V
V <sub>OH</sub>	输出高电压		V <sub>DD</sub> -0.6	—	—	V
I <sub>IL</sub>	输入漏电流	V <sub>SS</sub> <V <sub>I</sub> <V <sub>DD</sub>	-2	—	2	μA

表 21.6 芯片供电电压特性

测试条件（特别说明除外）： 工作温度 -40°C~125°C						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
V <sub>DD</sub>	电源电压		2.7	—	5.5	V
t <sub>vcc</sub>	VCC 上升速率	—	6	—	60000	μs
	VCC 下降速率	—	0	—	60000	

## 21.7 POR

表 21.7 芯片 POR 特性

测试条件（特别声明除外）： 工作温度 -40°C~125°C						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
V <sub>POR</sub>	上电复位电压	校准, temp=25°C	2.0	—	2.7	V
T <sub>emop</sub>	复位延时		—	2.5	—	ms
I <sub>DD</sub>	静态电流	V <sub>DD</sub> =3.3V	—	147	—	nA

## 21.8 LVR

表 21.8 芯片 LVR 特性

测试条件（特别声明除外）： 工作温度 -40°C~125°C						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
V <sub>LVR</sub>	掉电复位电压	temp=25°C	1.7	—	2.1	V
Hysteresis	迟滞		—	40	—	mV
I <sub>DD</sub>	静态电流	V <sub>DD</sub> =3.3V, temp=25°C	—	5	—	μA

## 21.9 INTHF

表 21.9 芯片 INTHF 特性

测试条件（特别说明除外）： 工作温度 -40°C~125°C						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
f <sub>CLKOUT</sub>	HOSC frequency	temp=25°C	15.84	16	16.16	MHz
TRIM	User trimming step	temp=25°C	0.1	0.2	0.35	%
DuCy	Duty Cycle	—	45	—	55	%
t <sub>su</sub>	Oscillator start-up time	—	—	5	—	μs
t <sub>stab</sub>	Oscillator stabilization time	—	—	17	—	μs
I <sub>POWER_ISS</sub>	Oscillator power consumption	temp=25°C	—	133	—	μA

## 21.10 INTLF

表 21.10 INTLF 特性

测试条件（特别说明除外）： 工作温度 -40°C~125°C						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
Freq*	振荡频率	校准后		32		KHz
Iq*	静态电流			330		nA

\*该数据为设计值

## 21.11 EXTLF

表 21.11 EXTLF 特性

测试条件（特别说明除外）： 工作温度 -40°C~125°C						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
Freq	振荡频率			32.768		KHz
Iq*	静态电流			340		nA

\*该数据为设计值

## 21.12 EXTHF

表 21.12 EXTLF 特性

测试条件（特别说明除外）： 工作温度 -40°C~125°C						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
Freq	振荡频率			20		MHz

Iq*	静态电流			4		mA
-----	------	--	--	---	--	----

\*该数据为设计值

## 21.13 ADC12

表 21.13 A/D 转换器 (ADC12) 特性

测试条件 (特别声明除外): 工作温度 -40°C~125°C						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
N <sub>R</sub> *	分辨率		—	—	12	位
E <sub>INL</sub> *	积分误差		—	±2	—	LSB
E <sub>DNL</sub> *	微分误差		—	±1	—	LSB
E <sub>OFF</sub> *	失调误差 <sup>(1)</sup>		—	±2	—	LSB
E <sub>GN</sub> *	增益误差		—	±2	—	LSB
V <sub>AIN</sub> *	满量程范围		V <sub>SS</sub>	—	V <sub>REF</sub>	V
T <sub>CNV</sub> *	AD转换时间		—	11	—	T <sub>AD</sub>
T <sub>sOP</sub> A <sub>le</sub> *	AD转换速率		—	—	400	KSPS
Iq*	AD工作电流		—	550	—	μA
Z	AD通道对地阻抗	特殊AD通道: AN8/AN13/AN22/AN38	7	—	—	MΩ
		其它AD通道	2	—	—	MΩ

注: (1)\*表示该数据为设计值。

(2) 失调误差固定为 5mV 正失调, 针对不同的参考电压对应为不同值, 使用时可以直接减去绝对值。

## 21.14 CMP

表 21.14 模拟比较器模块特性

测试条件 (特别声明除外): 工作温度 -40°C~125°C						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
A <sub>v</sub> *	开环增益			72		dB
V <sub>ICMR</sub> *	输入电压范围		V <sub>SS</sub>		V <sub>DD</sub> -1	V

\*该数据为设计值

## 21.15 ESD 和 Latch up

表 21.15 ESD 和 Latch up 特性

符号	说明	最小值	最大值	单位	附注
V <sub>HBM</sub>	静电放电电压, 人体放电模式	-8000	+8000	V	1
V <sub>CDM</sub>	静电放电电压, 设备充电模式	-1000	+1000	V	2

$I_{LAT}$	125°C温度环境下的闩锁电流	-200	+200	mA	3
-----------	-----------------	------	------	----	---

- 1.测试标准：AEC-Q100-002-E,HUMAN BODY MODEL ELECTROSTATIC DISCHARGE TEST。
- 2.测试标准：AEC-Q100-011-C1,CHARGED DEVICE MODEL (CDM) ELECTROSTATIC DISCHARGE TEST。
- 3.测试标准：AEC-Q100-004-D,IC LATCH-UP TEST。
  - 在 125°C环境温度下进行测试 (II 类)；
  - 电源组 1.5V<sub>ccmax</sub>。

## 21.16 直流特性图表

备注:某些图表中的数据超出了规定的工作范围(即超出了规定的VDD 范围), 这些图表仅供参考, 器件只有在规定的范围下工作才可以确保正常运行。

图 21.1: 不同 VDD 时典型  $I_{DD}$  -  $F_{OSC}$  关系曲线图(HFINTOSC 模式)

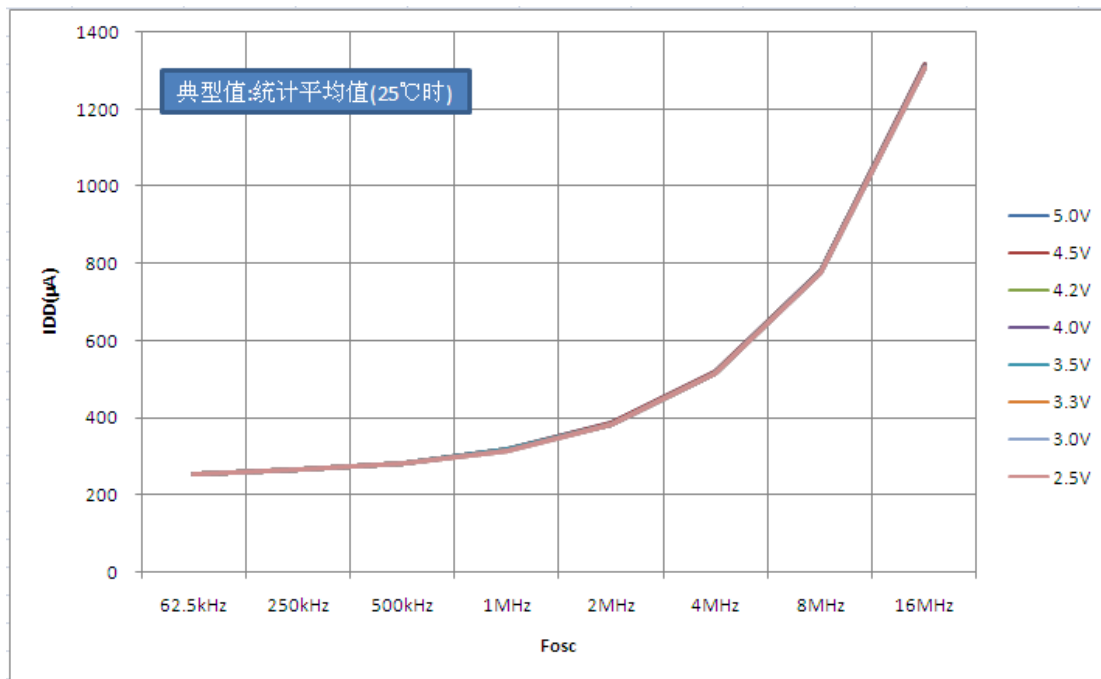


图 21.2: 不同 VDD 时典型  $I_{SLP}$  - VDD 关系曲线图(普通休眠模式, 禁止所有外设)

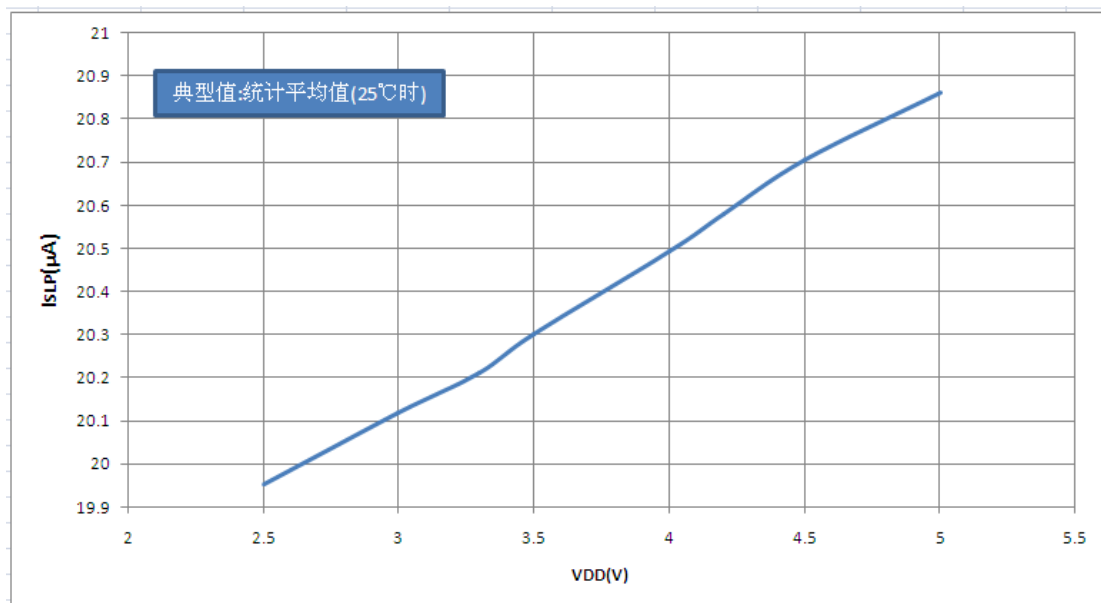


图 21.3: 不同 VDD 时典型  $I_{DSL P}$  — VDD 关系曲线图(深度休眠模式, 禁止所有外设)

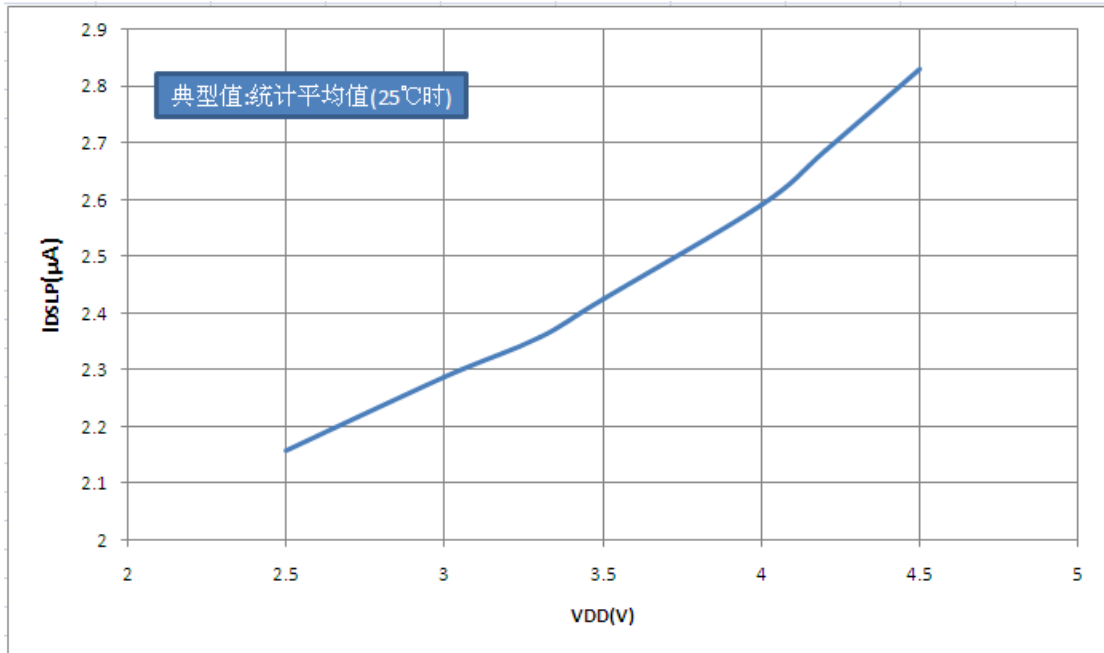
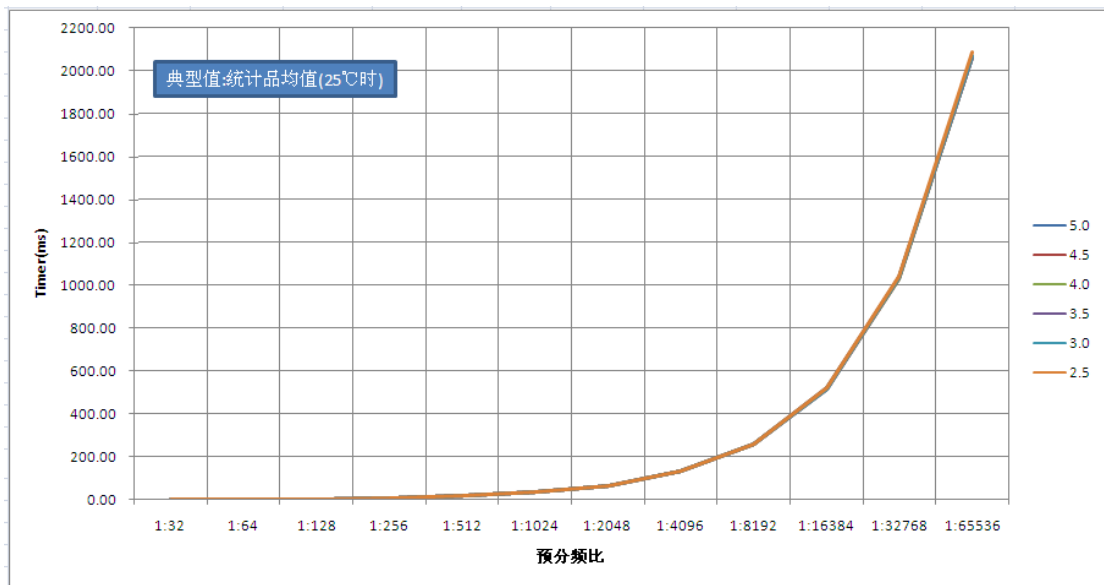
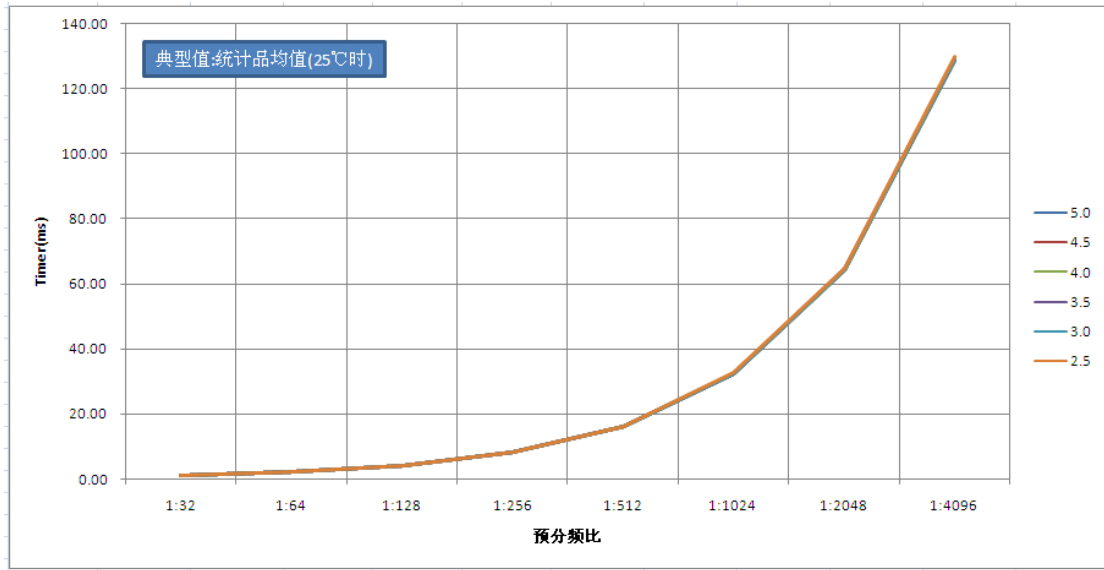


图 21.4 不同 VDD 时看门狗预分频比 WDTPS<3:0> 一周关系曲线图



备注:1/32 分频时看门狗周期为 1ms。

图 21.5 不同 VDD 时看门狗预分频比 PS<2:0> 一周期关系曲线图



备注:1/32 分频时看门狗周期为 1ms。

图 21.6:看门狗电流 I<sub>WDT</sub> — VDD 关系曲线图

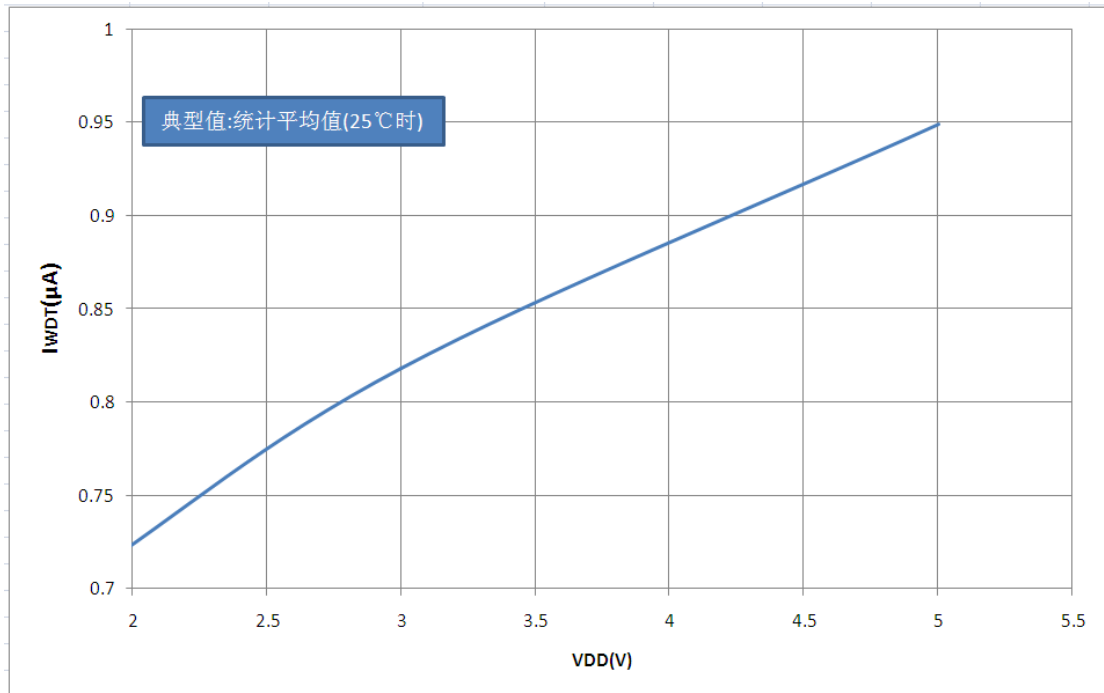




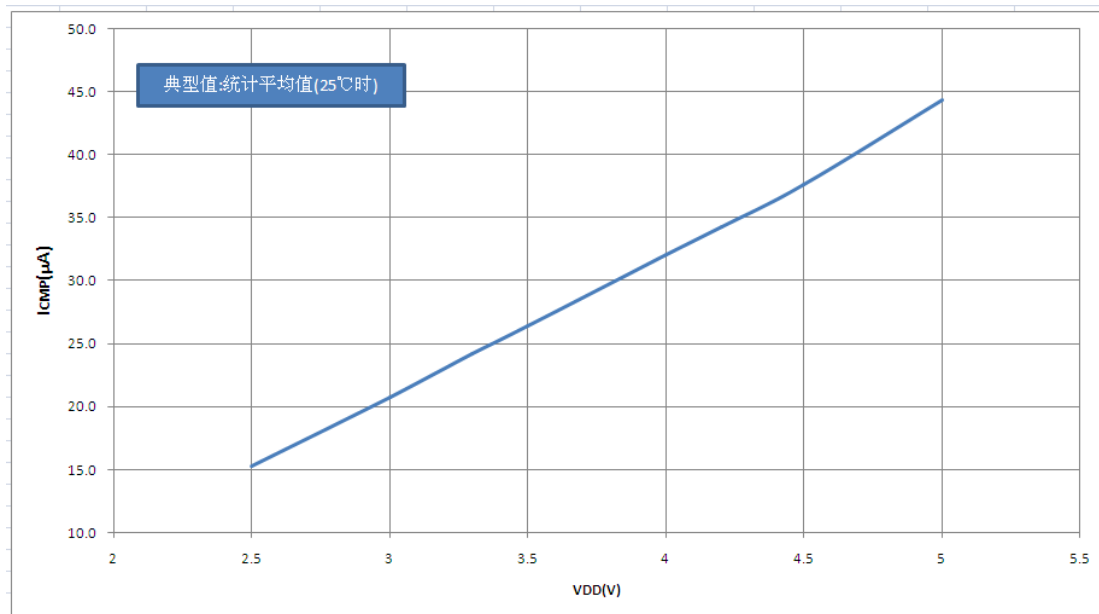
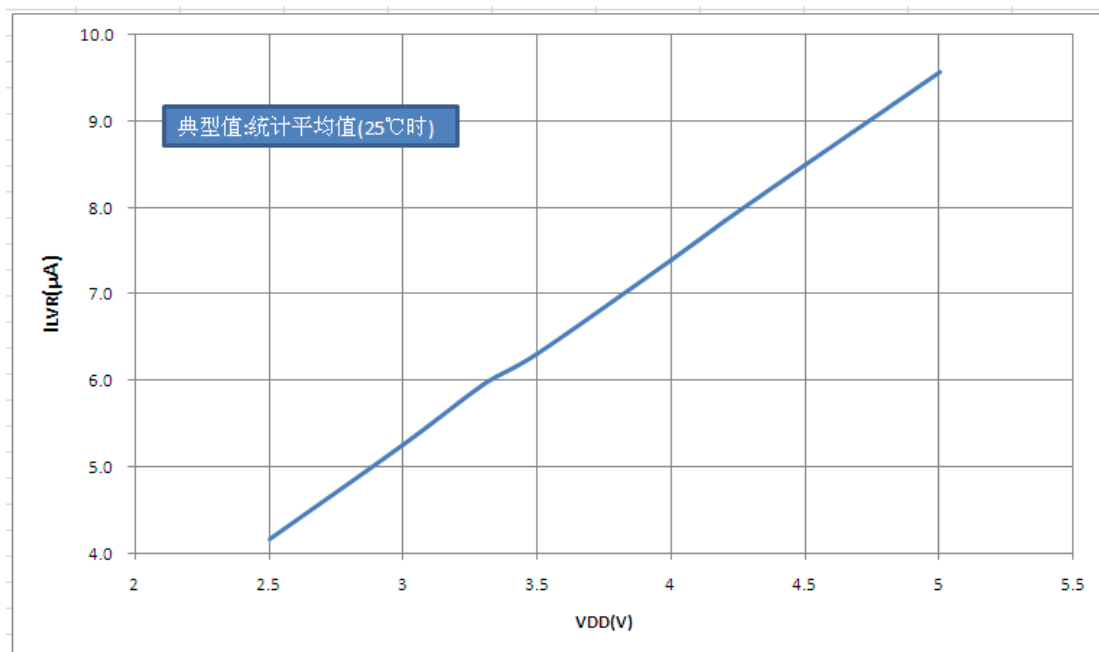
图 21.7:比较器电流  $I_{CMP}$  - VDD 关系曲线图 (使能一路比较器)图 21.8:欠压复位电流  $I_{LVR}$  - VDD 关系曲线图

图 21.9:欠压复位电流  $I_{ADC}$  —  $V_{DD}$  关系曲线图

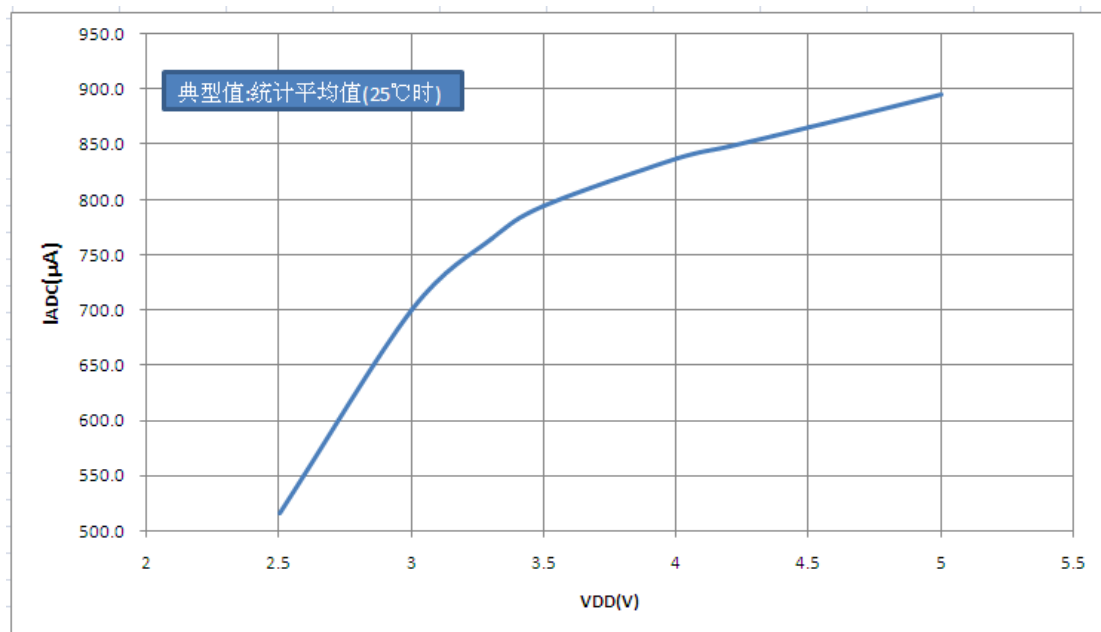


图 21.10:不同温度时  $V_{OH}$  —  $I_{OH}$  关系曲线图 ( $V_{DD} = 5.0V$ )

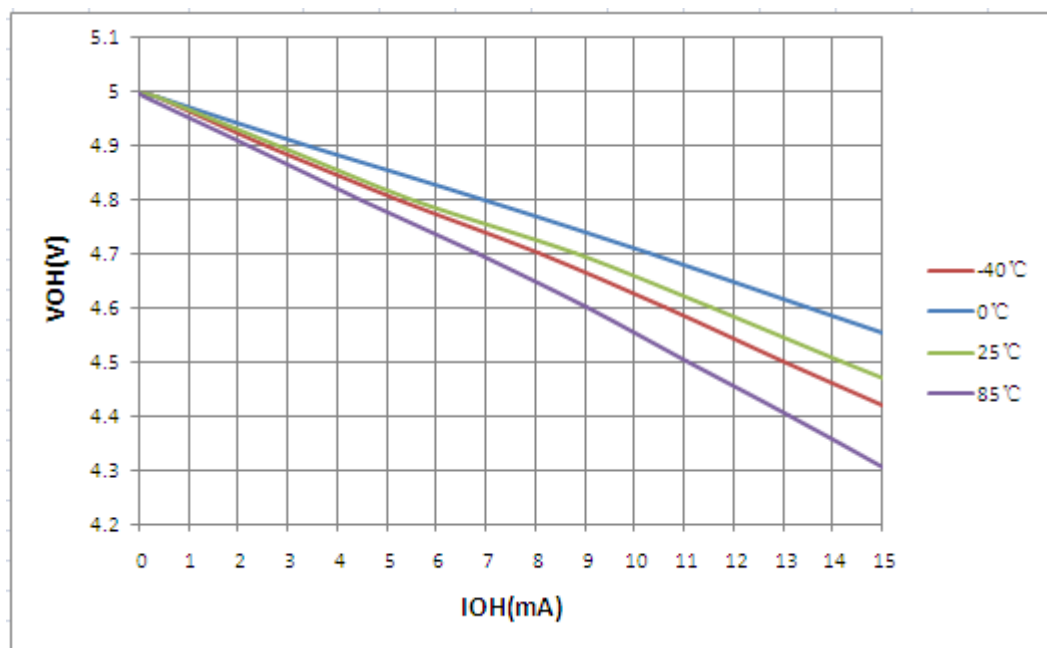


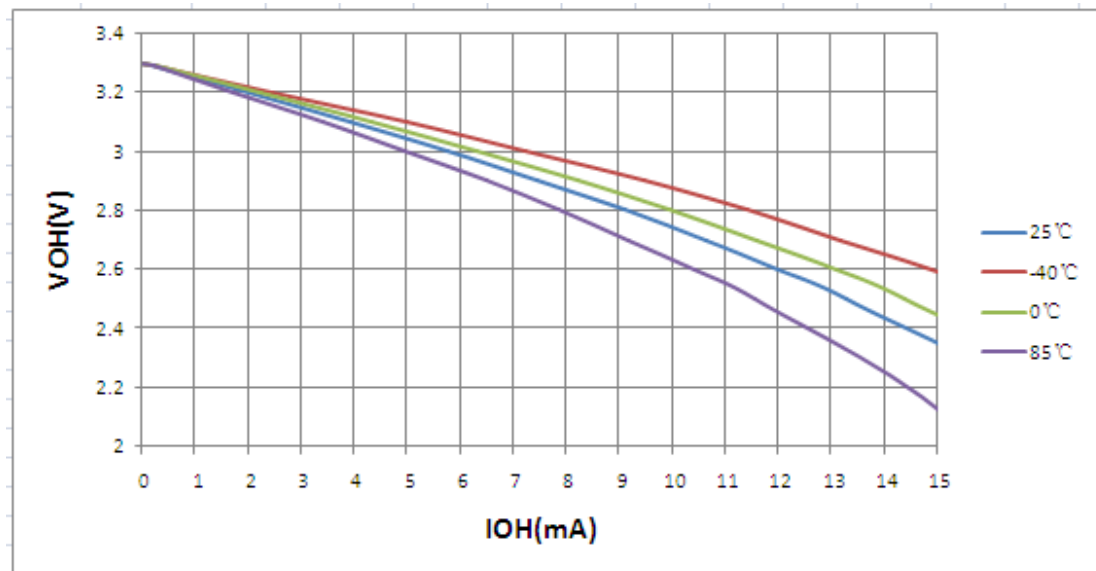
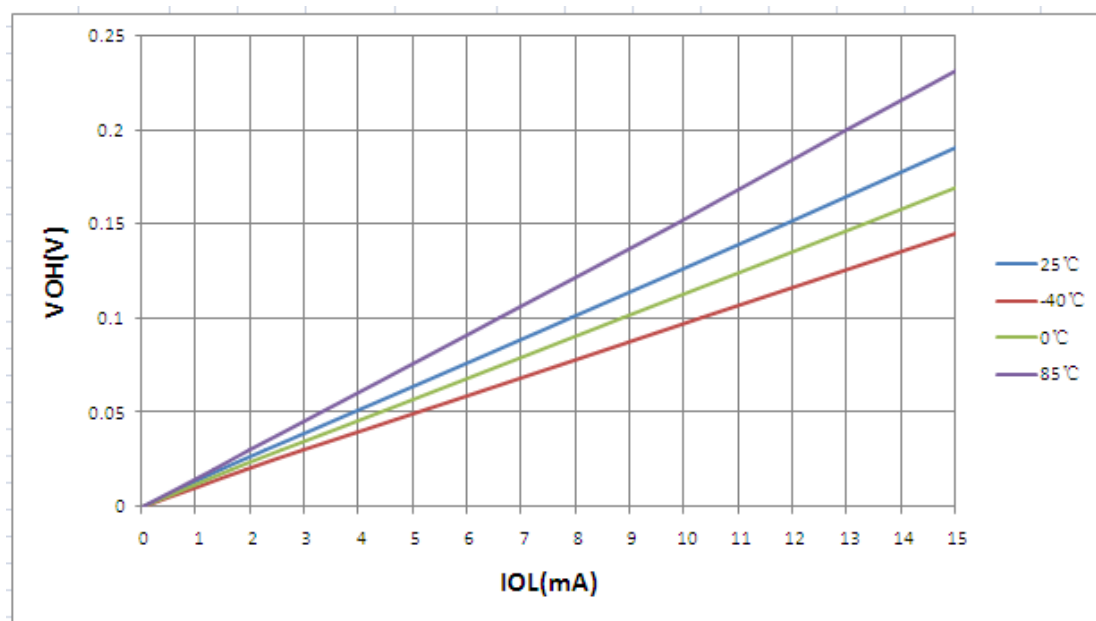
图 21.11:不同温度时  $V_{OH}$  -  $I_{OH}$  关系曲线图 ( $V_{DD} = 3.0V$ )图 21.12:不同温度时  $V_{OH}$  -  $I_{OL}$  关系曲线图 ( $V_{DD} = 5.0V$ )

图 21.13:不同温度时  $V_{OH}$  -  $I_{OL}$  关系曲线图 ( $V_{DD} = 3.0V$ )

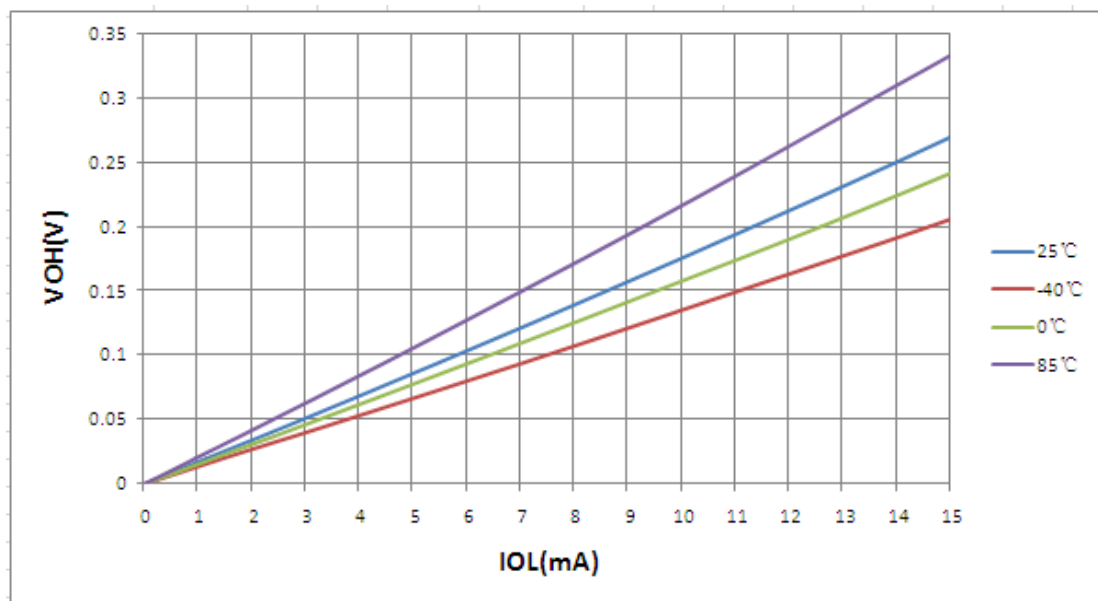


图 21.14:上拉功能打开时端口电压  $V_{OP}$  - 电流  $I_{OP}$  关系曲线图 ( $V_{DD} = 5.0V$ )

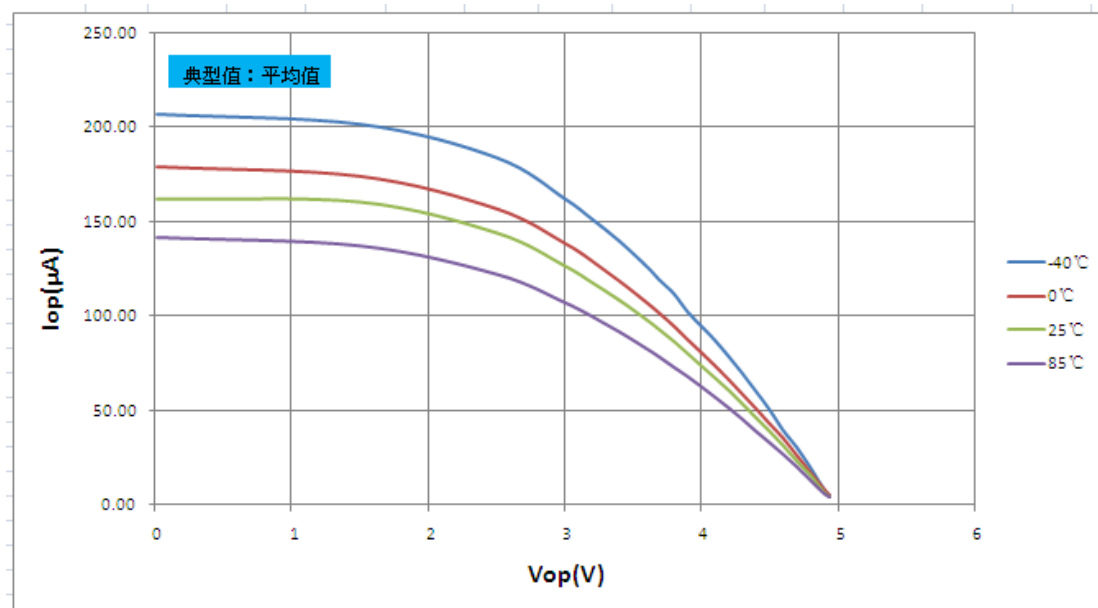
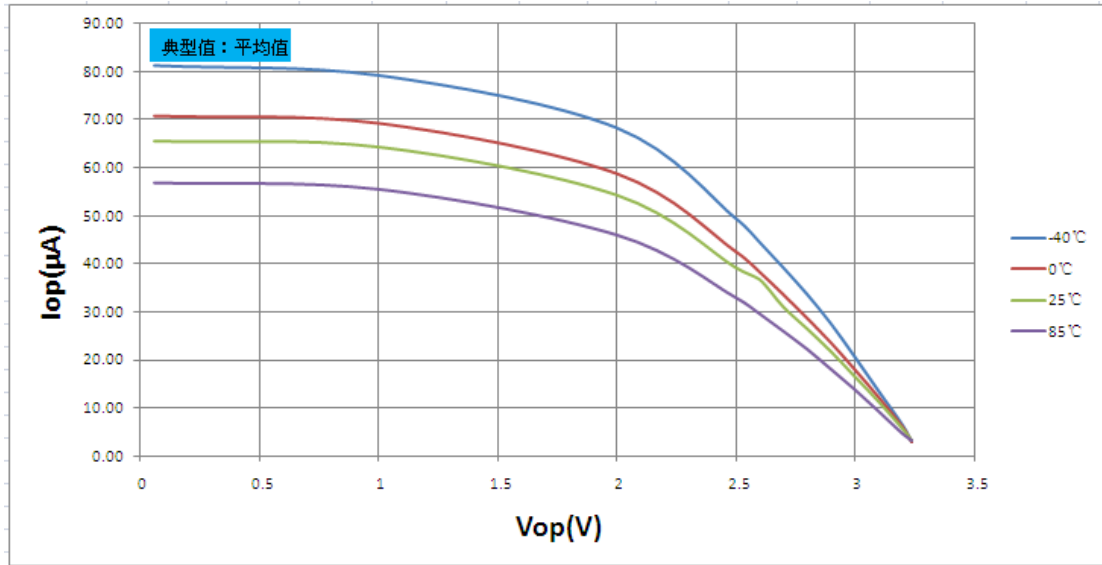
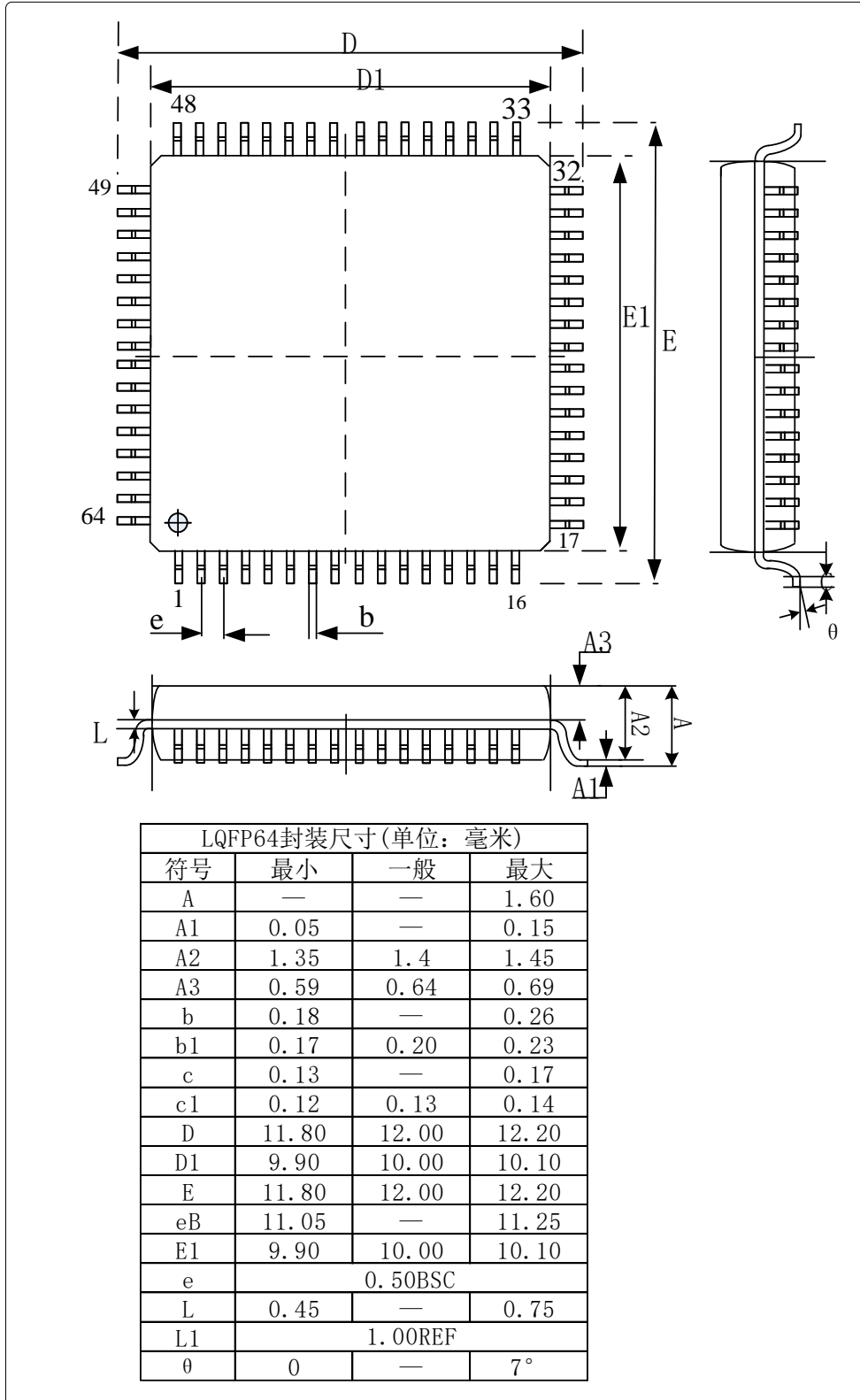


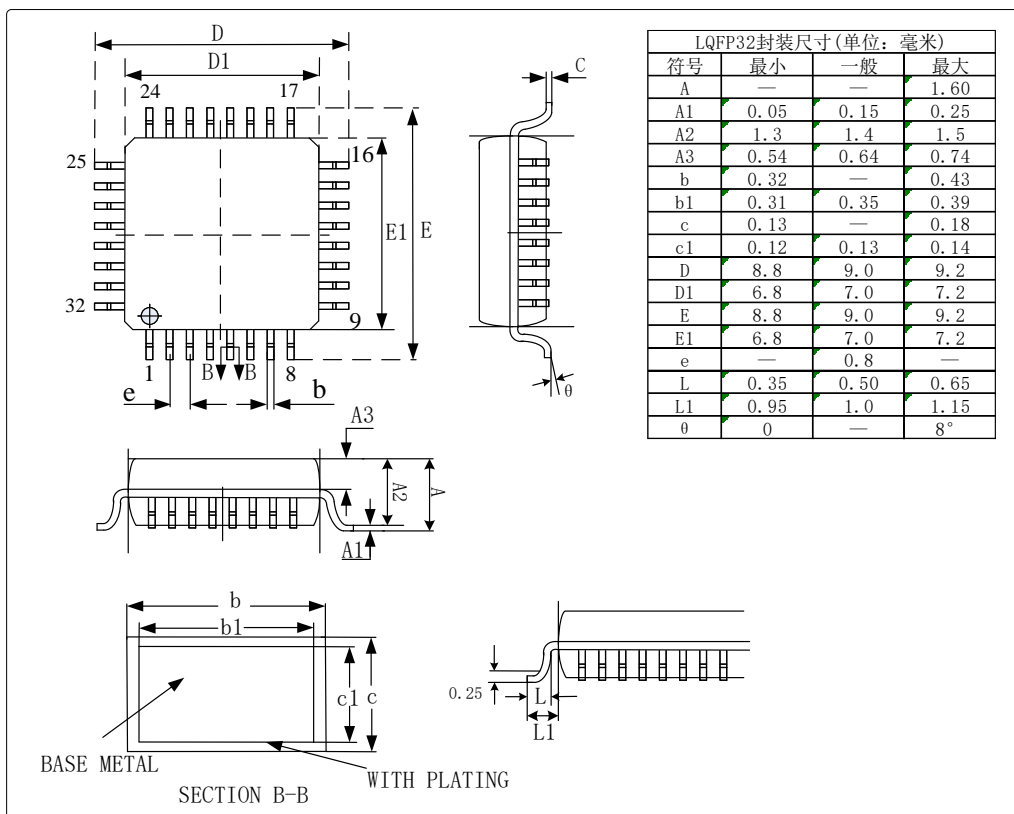
图 21.15:上拉功能打开时端口电压  $V_{OP}$  - 电流  $I_{OP}$  关系曲线图 ( $V_{DD} = 3.3V$ )

22 封装信息

64脚 LQFP 封装:



32脚 LQFP 封装:



## 附录 1 特殊功能寄存器 (SFR) 地址映射及功能汇总

## BANK0

BANK 0										
地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	复位初值
00H	IDLR	非物理寄存器 (用于间接寻址)								xxxx xxxx
01H	T0	定时/计数器 0(T0)寄存器								xxxx xxxx
02H	PCL	程序计数器(PC)低字节								0000 0000
03H	PSW	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	CY	---1 1xxx
04H	IR	用于间接寻址								xxxx xxxx
05H	P0	P07	P06	P05	P04	P03	P02	P01	P00	xx xx xxxx
06H	P2	—	—	P25	P24	P23	P22	P21	P20	xxxx xxxx
07H	P1	P17	P16	P15	P14	P13	P12	P11	P10	xxxx xxxx
08H	P3	P37	P36	P35	P34	—	P32	P31	P30	xxxx xxxx
09H	P4	—	—	P45	P44	P43	—	—	—	xxxx xxxx
0AH	PCH	—	—	—	程序计数器(PC)高字节					---0 0000
0BH	INTCTL	AIE/AIEH	PUIE/AIE L	TOIE	INT0IE	POIE	TOIF	INT0IF	POIF	0000 0000
0CH	EIF1	T3IF	ADIF	—	INT1IF	C1IF	PWM2IF	T2IF	T1IF	0000 0000
0DH	EIF2	EEIF	C2IF	RC2IF	TX2IF	SSC1IF	CCP1IF	RC1IF	TX1IF	0000 0000
0EH	T1L	定时器/计数器 T1 低字节寄存器								xxxx xxxx
0FH	T1H	定时器/计数器 T1 高字节寄存器								xxxx xxxx
10H	T1CTL	—	T1GC1	T1GC0	T1GEN	T1PL	$\overline{T1SY}$	T1CS	T1ON	0000 0000
11H	T2L	定时器 2(T2)低字节寄存器								0000 0000
12H	T2CTL0	—	T2CKBS3	T2CKBS2	T2CKBS1	T2CKBS0	T2ON	T2CKPS1	T2CKPS0	-000 0000
13H	PWM1L	PWM1 占空比设置寄存器								xxxx xxxx
15H	PWMCTL	—	INTISE	—	—	PWMMOD E	PWM16ON	PWM2ON	PWM1ON	11-- 0000
16H	PP1	PWM1 周期寄存器								1111 1111
17H	BANK	—	—	—	PR4	PR3	PR2	PR1	PR0	---0 0000
19H	CMCTL0	—	—	—	—	C4EDG	C3EDG	C2EDG	C1EDG	---- 0000
1AH	C1CTL	C1EN	C1OE	C1P2	C1P1	C1P0	C1M2	C1M1	C1M0	0000 --00
1DH	COU	—	VCEN	—	—	C4OUT	C3OUT	C2OUT	C1OUT	0--- 0000
1EH	ADCDATA0 H	ADC 数据寄存器高字节								xxxx xxxx
1FH	ADCCTL0	ADLR	T2CCR0 ON	—	—	ADCLK1	ADCLK0	START	ADEN	00-- 0000
21H	OPTR	$\overline{PUPH}$	INT0SE	TOCS	TOSE	PSA	PS2	PS1	PS0	1111 1111
22H	IP0	—	—	—	—	—	PT0	PINT0	PP0	---- -000



BANK 0										
地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	复位初值
23H	IP1	PT3	PADC	—	PINT1	PC1	PPWM2	PT2	PT1	0000 0000
24H	IP2	PEE	PC2	PRC2	PTX2	PSSC11	PCCP2	—	—	0000 0000
25H	TR0	TR07	TR06	TR05	TR04	TR03	TR02	TR01	TR00	1111 1111
26H	TR2	—	—	TR25	TR24	TR23	TR22	TR21	TR20	1111 1111
27H	TR1	TR17	TR16	TR15	TR14	TR13	TR12	TR11	TR10	1111 1111
28H	TR3	TR37	TR36	TR35	TR34	—	TR32	TR31	TR30	--11 1111
29H	TR4	—	—	TR45	TR44	TR43	—	—	—	1111 1111
2BH	VRECTL	VRESEL1	VRESEL0	VRECALE N	VREBG	VREOE	P18OE	VREEN	—	0000 0000
2CH	EIE1	T3IE	ADIE	—	INT1IE	C1IE	PWM2IE	T2IE	T1IE	0000 0000
2DH	EIE2	EEIE	C2IE	RC2IE	TX2IE	SSC11IE	CCP2IE	—	—	0000 0000
2EH	PCTL	—	—	—	SLVREN	IPEN	SWDTEN	POR	LVR	0001 00xx
30H	OSCCAL2	晶振校准值寄存器 2 (振荡器频率调节)								1000 0000
31H	C2CTL	C2EN	C2OE	C2P2	C2P1	C2P0	C2M2	C2M1	C2M0	0000 --00
32H	PP2	PWM2 周期寄存器								1111 1111
33H	PWM2L	PWM2 占空比设置寄存器								xxxx xxxx
35H	PUR0	PUR07	PUR06	PUR05	PUR04	PUR03	PUR02	PUR01	—	1111 111-
36H	IOCL0	IOCL07	IOCL06	IOCL05	IOCL04	IOCL03	IOCL02	IOCL01	IOCL00	0000 0000
37H	OSCCAL1	晶振校准值寄存器 1(振荡器模式 8M/16M 等)								---- -000
38H	NVMDATAH	数据高 8 位								0000 0000
39H	NVMDATAL	数据低 8 位								0000 0000
3AH	NVMADDRH	地址指针高 8 位								0000 0000
3BH	NVMADDRL	地址指针低 8 位								0000 0000
3CH	NVMCTL0	控制寄存器 0								--- x000
3DH	NVMCTL1	控制寄存器 1								---- ----
3EH	ADCDATA0 L	ADC 数据寄存器 0 低字节								xxxx xxxx
3FH	ADCCTL1	ADCALEN	ADCS2	ADCS1	ADCS0	VCFG1	VCFG0	保留	保留	0000 0000
40H	WDTPS	—	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	---- 0000
41H	T2H	T2 高 8 位								0000 0000
42H	PP60H	PWM6 周期 0 寄存器高 8 位								xxxx xxxx
43H	PWM6L0	PWM6 通道 0 占空比寄存器低 8 位								0000 0000
44H	PWM6H0	PWM6 通道 0 占空比寄存器高 8 位								0000 0000
45H	P0LR	P0LR7	P0LR6	P0LR5	P0LR4	P0LR3	P0LR2	P0LR1	—	xxxx xxxx
46H	P2LR	—	—	P2LR5	P2LR4	P2LR3	P2LR2	P2LR1	P2LR0	xxxx xxxx
47H	P1LR	P1LR7	P1LR6	P1LR5	P1LR4	P1LR3	P1LR2	P1LR1	P1LR0	xxxx xxxx
48H	P3LR	P3LR7	P3LR6	P3LR5	P3LR4	—	P3LR2	P3LR1	P3LR0	--xx xxxx
49H	P4LR	—	—	P4LR5	P4LR4	P4LR3	—	—	—	xxxx xxxx
4AH	ANS0	ANS07	ANS06	ANS05	ANS04	ANS03	ANS02	ANS01	ANS00	1111 111-
4BH	ANS1	ANS17	ANS16	ANS15	ANS14	ANS13	ANS12	ANS11	ANS10	1111 1111
4CH	ANS2	—	—	ANS25	ANS24	ANS23	ANS22	ANS21	ANS20	1111 1111

BANK 0										
地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	复位初值
4DH	ANS3	ANS37	ANS36	ANS35	ANS34	—	ANS32	ANS31	ANS30	--11 1111
4EH	ANS4	—	—	ANS45	ANS44	ANS43	—	—	—	1111 1111
4FH	T0CTL	—	—	—	—	—	—	T0CLK1	T0CLK0	--- --00
50H	ADCCTL2	—	ADCHS6	ADCHS5	ADCHS4	ADCHS3	ADCHS2	ADCHS1	ADCHS0	-000 0000
51H	T2CTL1	T2MOD1	T2MOD0	T2OFS1	T2OFS0	T2TRIGEN	T2CLK1	T2CLK0	T2DIR	0000 000x
52H	PP60L	PWM6 周期 0 寄存器低 8 位								1111 1111
53H	BKPCTL	BKP7	BKP6	BKP5	BKP4	BKP3	BKP2	BKP1	BKP0	0000 0000
54H	SCLKSTU	OSCFAILIF	CLKS	OSTS	HTS	LTS	OSTIF	—	—	0000 00--
55H	PWM6H1	PWM6 通道 1 占空比寄存器高 8 位								xxxx xxxx
56H	PWM6L1	PWM6 通道 1 占空比寄存器低 8 位								xxxx xxxx
57H	PWM6CTL 0	—	—	—	—	P6MOD3	P6MOD2	P6MOD1	P6MOD0	0000 0000
5DH	P6ATRCCTL	P6STR3H	—	P6STR2H	—	P6STR1H	—	P6STR0H	—	1010 1010
5EH	PWM6CTL 2	P6FUSES	—	P6UDEVT3	P6UDEVT2	P6UDEVT1	P6UDEVT0	P6UDEN1	P6UDEN0	1000 0000
60H	PWM6PC	P6PC3H	—	P6PC2H	—	P6PC1H	—	P6PC0H	—	0000 0000
62H	PWM6H2	PWM 通道 2 占空比寄存器高 8 位								xxxx xxxx
63H	PWM6L2	PWM 通道 2 占空比寄存器低 8 位								xxxx xxxx
64H	PWM6H3	PWM 通道 3 占空比寄存器高 8 位								xxxx xxxx
65H	PWM6L3	PWM 通道 3 占空比寄存器低 8 位								xxxx xxxx
68H	CCTCTL	CCTEN	CSEL1	CSEL0	CRSET	CFSET	CCT7	CCT3	CCT1	0000 0000
69H	SCLKCTL	SCLKOE	SCS	SCKDIV2	SCKDIV1	SCKDIV0	SCKS2	SCKS1	SCKS0	0001 1000
6AH	HFCKCTL	HFCKEN	UPCK1	UPCK0	—	HFCKDIV2	HFCKDIV1	HFCKDIV0	HFCKS	0-00 000-
6BH	LFCKCTL	LFCKEN	—	—	—	LFCKDIV2	LFCKDIV1	LFCKDIV0	LFCKS	0--- 0000

## BANK1

Bank 1										
地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	复位初值
101H	T0	定时/计数器 0(T0)寄存器								xxxx xxxx
102H	PCL	程序计数器(PC)低字节								0000 0000
103H	PSW	—	—	—	$\overline{IO}$	$\overline{PD}$	Z	DC	CY	---1 1xxx
104H	IR	用于间接寻址								xxxx xxxx
105H	P5	—	—	P55	P54	P53	P52	P51	P50	--- xxxx
106H	P6	—	P66	P65	P64	P63	P62	P61	—	--xx xxxx
107H	P7	—	—	P75	—	P73	P72	P71	—	--xx xxxx
109H	P9	—	—	P95	P94	P93	P92	P91	P90	xxxx xxxx
10CH	IP5	PT7	PT6	PT5	PCCP5	PRE2	PTE2	—	—	0000 0000
10DH	EIF4	CCP7IF	CCP3IF	T1GIF	—	C4IF	C3IF	T4IF	BCL1IF	---0 0000
10EH	EIE4	CCP7IE	CCP3IE	T1GIE	—	C4IE	C3IE	T4IE	BCL1IE	---0 0000
10FH	EIF5	T7IF	T6IF	T5IF	CCP5IF	RE2IF	TE2IF	—	—	0000 0000
110H	EIE5	T7IE	T6IE	T5IE	CCP5IE	RE2IE	TE2IE	—	—	0000 0000
111H	T4L	T4 低字节寄存器								0000 0000
112H	T4CTL0	T4SLPEN	T4CKBS3	T4CKBS2	T4CKBS1	T4CKBS0	T4ON	T4CKPS1	T4CKPS0	0000 0000
113H	WCFR	—	WW6	WW5	WW4	WW3	WW2	WW1	WW0	-000 0000
114H	WCNT	—	WT6	WT5	WT4	WT3	WT2	WT1	WT0	-000 0000
115H	WCTL	WWDTEN	—	—	WPS4	WPS3	WPS2	WPS1	WPS0	0--0 0000
116H	T3REH	T3 重载寄存器高 8 位								0000 0000
118H	T1CTL2	T1GVAL	T1GSPM	T1GTM	T1GGO	T1CKS1	T1CKS0	T1CLK1	T1CLK0	0--0 0000
119H	T3CTL	T3REN	—	T3CKS1	T3CKS0	—	T3CS1	T3CS0	T3ON	0-00 -000
11AH	T3H	T3 寄存器高 8 位								0000 0000
11BH	T3REL	T3 重载寄存器低 8 位								0000 0000
11CH	T3L	T3 寄存器低 8 位								0000 0000
123H	UPINSET2	USLM2	—	—	—	—	—	—	—	0000 0000
124H	IP4	PCCP7	PCCP3	PT1G	—	PC4	PC3	PT4	PBCL1	---0 0000
125H	TR5	—	—	TR55	TR54	TR53	TR52	TR51	TR50	1111 1111
126H	TR6	—	TR66	TR65	TR64	TR63	TR62	TR61	—	--11 1111
127H	TR7	—	—	TR75	—	TR73	TR72	TR71	—	1111 1111
129H	TR9	—	—	TR95	TR94	TR93	TR92	TR91	TR90	1111 1111
12AH	T7CTL	T7REN	—	T7CKS1	T7CKS0	—	T7CS1	T7CS0	T7ON	0-00 -000
12BH	T7L	T7 寄存器低 8 位								0000 0000
12CH	T7H	T7 寄存器高 8 位								0000 0000
12DH	T7REL	T7 重载寄存器低 8 位								0000 0000

Bank 1										
地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	复位初值
12EH	T7REH	T7 重载寄存器高 8 位								0000 0000
12FH	VRES0	VRES23	VRES22	VRES21	VRES20	VRES13	VRES12	VRES11	VRES10	0000 0000
131H	OSCCAL0	内部高频晶振校准寄存器 0(振荡器 BG)								0001 0000
132H	OSCCAL3	内部高频晶振校准寄存器 3 (振荡器温漂)								0000 0000
133H	SSCI1CTL1	—	—	—	—	—	—	SSCI1CKS1	SSCI1CKS0	---- --00
134H	SSCI1STA	SAMPLE1	CKEGE1	SSCI1DA	SSCI1ST OP	SSCI1STA RT	SSCI1RW	SSCI1UA	SSCI1BUF	0000 0000
135H	SSCI1CTL0	SSCI1WCFL	SSCI1IOV	SSCI1EN	SSCI1CK P	SSCI1MO D3	SSCI1MOD2	SSCI1MOD 1	SSCI1MOD 0	0000 0000
136H	SSCI1BUFR	SSCI1 数据接收缓冲/发送寄存器								0000 0000
137H	SSCI1ADD	SSCI1ADD7	SSCI1ADD6	SSCI1ADD5	SSCI1ADD4	SSCI1ADD3	SSCI1ADD2	SSCI1ADD1	SSCI1ADD0	1111 1111
	SSCI1MSK	SSCI1MSK 7	SSCI1MSK 6	SSCI1MSK5	SSCI1MSK4	SSCI1MSK3	SSCI1MSK2	SSCI1MSK1	SSCI1MSK0	
138H	SSCI1CTL2	SSCI1CALLEN	SSCI1ACKSTA	SSCI1ACKDA T	SSCI1ACKEN	SSCI1RCEN	STOPEN1	RESTARTEN1	STARTEN1	0000 0000
141H	T4H	T4 寄存器高 8 位								0000 0000
142H	PP61H	PWM6 周期寄存器高 8 位								0000 0000
145H	P5LR	—	—	P5LR5	P5LR4	P5LR3	P5LR2	P5LR1	P5LR0	xxxx xxxx
146H	P6LR	—	P6LR6	P6LR5	P6LR4	P6LR3	P6LR2	P6LR1	—	--xx xxxx
147H	P7LR	—	—	P7LR5	—	P7LR3	P7LR2	P7LR1	—	xxxx xxxx
149H	P9LR	—	—	P9LR5	P9LR4	P9LR3	P9LR2	P9LR1	P9LR0	--xx xxxx
14AH	ANS5	ANS57	ANS56	ANS55	ANS54	ANS53	ANS52	ANS51	ANS50	xxxx xxxx
14BH	ANS6	ANS67	ANS66	ANS65	ANS64	ANS63	ANS62	ANS61	ANS60	--xx xxxx
14CH	ANS7	—	—	ANS75	ANS74	ANS73	ANS72	ANS71	ANS70	xxxx xxxx
14DH	ANS8	ANS87	ANS86	ANS85	ANS84	ANS83	ANS82	ANS81	ANS80	xxxx xxxx
14EH	ANS9	—	—	ANS95	ANS94	ANS93	ANS92	ANS91	ANS90	--xx xxxx
14FH	VRES1	VRES43	VRES42	VRES41	VRES40	VRES33	VRES32	VRES31	VRES30	0000 0000
151H	T4CTL1	T4MOD1	T4MOD0	T4OFS1	T4OFS0	T4TRIGEN	T4CLK1	T4CLK0	T4DIR	0000 0000
152H	PP61L	PWM6 周期寄存器低 8 位								0000 0000
154H	T2CCR0L	T2 启动 ADC 设置寄存器 0 低 8 位								0000 0000
155H	T2CCR0H	T2 启动 ADC 设置寄存器 0 高 8 位								0000 0000
156H	T2CCR1L	T2 启动 ADC 设置寄存器 1 低 8 位								0000 0000
157H	T2CCR1H	T2 启动 ADC 设置寄存器 1 高 8 位								0000 0000
158H	T4CCR0L	T4 启动 ADC 设置寄存器 0 低 8 位								0000 0000
159H	EUBRGH2	USART2 波特率数据寄存器高字节								0000 0000
15AH	T4CCR0H	T4 启动 ADC 设置寄存器 0 高 8 位								0000 0000

Bank 1										
地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	复位初值
160H	UADMATC H2	ADM27	ADM26	ADM25	ADM24	ADM23	ADM22	ADM21	ADM20	0000 0000
163H	U7816TXC TL2	U7816EN2	CLKOUT2	TX9DSE L2	STOP2	TPAR2	TINV2	TCONV2	BGTEN2	0001 0000
164H	EGTCTL2	EGT27	EGT26	EGT25	EGT24	EGT23	EGT22	EGT21	EGT20	0000 0000
165H	CLKDIV2	CLKDIV27	CLKDIV26	CLKDIV2 5	CLKDIV2 4	CLKDIV2 3	CLKDIV22	CLKDIV21	CLKDIV20	0000 0000
166H	U7816CTL2	PSEL2	—	TREPEN2	TREP21	TREP20	RREPEN2	RREP21	RREP20	0-00 0000
167H	U7816RXC TL2	ERSW21	ERSW20	RPAR2	RINV2	RCONV2	—	—	PAREF2	0000 0-0
168H	RSCCTL2	SPEN2	R2X9	SRXEN2	CRXEN2	ADREN2	FRER2	OVFER2	R2X9D	0000 000x
169H	TXSDR2	USART2 发送数据寄存器								xxxx xxxx
16AH	RXSDR2	USART2 接收数据寄存器								xxxx xxxx
16BH	BRCCTL2	ABRDOVF2	RCIDLF2	SCKPS2	B2RG16	BR2CKS1	BR2CKS0	WUEN2	ABRDEN2	0100 0000
16CH	TSCCTL2	CSRS2	T2X9	TXEN2	SYNC2	SENDB2	HBRG2	TXSRS2	T2X9D	0000 0011
16DH	EUBRGL2	USART2 波特率数据寄存器低字节								0000 0000

## BANK2

Bank 2										
地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	复位初值
205H	SSCI2STA	SAMPLE2	CKEGE2	SSCI2DA	SSCI2STO P	SSCI2STAR T	SSCI2RW	SSCI2UA	SSCI2BUF	0000 0000
206H	SSCI2CTL0	SSCI2WC FL	SSCI2OV	SSCI2EN	SSCI2CKP	SSCI2MOD 3	SSCI2MOD 2	SSCI2MOD 1	SSCI2MOD0	---- -000
207H	SSCI2BUFR	SSCI2 数据接收缓冲/发送寄存器								0000 0000
208H	SSCI2ADD	SSCI2ADD7	SSCI2ADD6	SSCI2ADD 5	SSCI2ADD4	SSCI2ADD3	SSCI2ADD2	SSCI2ADD1	SSCI2ADD0	1111 1111
	SSCI2MSK	SSCI2MSK7	SSCI2MSK6	SSCI2MSK 5	SSCI2MSK4	SSCI2MSK3	SSCI2MSK2	SSCI2MSK1	SSCI2MSK0	
209H	SSCI2CTL2	SSCI2CAL LEN	SSCI2AC KSTA	SSCI2AC KDAT	SSCI2AC KEN	SSCI2RCE N	STOPEN2	RESTARTE N2	STARTEN2	0000 0000
20CH	RSCTL3	SPEN3	R3X9	SRXEN3	CRXEN3	ADREN3	FRER3	OVFER3	R3X9D	0000 0000
20DH	TXSDR3	USART3 发送数据寄存器								xxxx xxxx
20EH	RXSDR3	USART3 接收数据寄存器								xxxx xxxx
20FH	BRCTL3	ABRDOV F3	RCIDLF3	SCKPS3	B3RG16	BR3CKS1	BR3CKS0	WUEN3	ABRDEN3	0100 0000
210H	CCP3CTL	—	—	—	CCP3OE	CCP3MOD 3	CCP3MOD 2	CCP3MOD 1	CCP3MOD0	--0 0000
211H	CCP3H	CCP3 寄存器高 8 位								0000 0000
212H	CCP3L	CCP3 寄存器低 8 位								0000 0000
213H	CCP7CTL	—	—	—	CCP7OE	CCP7MOD 3	CCP7MOD 2	CCP7MOD 1	CCP7MOD0	--0 0000
214H	CCP7H	CCP7 寄存器高 8 位								0000 0000
215H	CCP7L	CCP7 寄存器低 8 位								0000 0000
216H	TSCTL3	CSRS3	T3X9	TXEN3	SYNC3	SENB3	HBRG3	TXSRS3	T3X9D	0000 0000
218H	EUBRGL3	USART3 波特率数据寄存器低字节								0000 0000
219H	EUBRGH3	USART3 波特率数据寄存器高字节								0000 0000
21AH	EIF3	—	—	TTIF	ALRIF	DAYIF	HOURLIF	MINIF	SECIF	0000 0000
21BH	EIE3	—	—	TTIE	ALRIE	DAYIE	HOURLIE	MINIE	SECIE	0000 0000
21CH	UADMATC H3	ADM37	ADM36	ADM35	ADM34	ADM33	ADM32	ADM31	ADM30	0000 0000
21EH	UPINSET3	USLM3	—	—	—	—	—	—	—	0000 0000
21FH	SSCI2CTL1	—	—	—	—	—	—	SSCI2CKS1	SSCI2CKS0	---- -00
22AH	IP3	—	—	PTT	PALRM	PDAY	PHOUR	PMIN	PSEC	0000 0000
231H	FILT1CTL	INV1	WEN1	FEN1	F1CS1	F1CS0	FILTCNT12	FILTCNT11	FILTCNT10	000- -000
232H	FILT1PER	FP17	FP16	FP15	FP14	FP13	FP12	FP11	FP10	0000 0000

Bank 2										
地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	复位初值
233H	FILT2CTL	INV2	WEN2	FEN2	F2CS1	F2CS0	FILTCNT22	FILTCNT21	FILTCNT20	000-000
234H	FILT2PER	FP27	FP26	FP25	FP24	FP23	FP22	FP21	FP20	0000 0000
235H	FILT3CTL	INV3	WEN3	FEN3	F3CS1	F3CS0	FILTCNT32	FILTCNT31	FILTCNT30	0000 0000
236H	FILT3PER	FP37	FP36	FP35	FP34	FP33	FP32	FP31	FP30	0000 0000
237H	FILT4CTL	INV4	WEN4	FEN4	F4CS1	F4CS0	FILTCNT42	FILTCNT41	FILTCNT40	0000 0000
238H	FILT4PER	FP47	FP46	FP45	FP44	FP43	FP42	FP41	FP40	0000 0000
239H	C3CTL	C3EN	C3OE	C3P2	C3P1	C3P0	C3M2	C3M1	C3M0	0--0--00
23AH	C4CTL	C4EN	—	—	—	C4P0	—	C4M1	C4M0	0--0--00
23BH	VRECAL2	参考电压校准寄存器 2								0000 0000
23CH	VRECAL3	参考电压校准寄存器 3								1010 0000
240H	PUR9	—	—	PUR95	PUR94	PUR93	PUR92	PUR91	PUR90	1111 1111
241H	PUR1	PUR17	PUR16	PUR15	PUR14	PUR13	PUR12	PUR11	PUR10	1111 1111
242H	PUR2	—	—	PUR25	PUR24	PUR23	PUR22	PUR21	PUR20	1111 1111
243H	PUR3	PUR37	PUR36	PUR35	PUR34	—	PUR32	PUR31	PUR30	--11 1111
244H	PUR4	—	—	PUR45	PUR44	PUR43	—	—	—	1111 1111
245H	PUR5	—	—	PUR55	PUR54	PUR53	PUR52	PUR51	PUR50	1111 1111
246H	PUR6	—	PUR66	PUR65	PUR64	PUR63	PUR62	PUR61	—	--11 1111
247H	PUR7	—	—	PUR75	—	PUR73	PUR72	PUR71	—	1111 1111
251H	ADCDATA 1L	AD 数据寄存器 1 低 8 位								xxxx xxxx
252H	ADCDATA 1H	AD 数据寄存器 1 高 8 位								xxxx xxxx
253H	ADCDATA 2L	AD 数据寄存器 2 低 8 位								xxxx xxxx
254H	ADCDATA 2H	AD 数据寄存器 2 高 8 位								xxxx xxxx
255H	ADCDATA 3L	AD 数据寄存器 3 低 8 位								xxxx xxxx
256H	ADCDATA 3H	AD 数据寄存器 3 高 8 位								xxxx xxxx
25AH	ADCINTCT L	T2CCR10 N	T4CCR00 N	ADTEST	T6CCR00 N	T5CCR10N	T5CCR00N	INTCTL1	INTCTL0	0000 0000
25BH	ADSCANC TL	ADSCAN EN	SMOD1	SMOD0	STIM4	STIM3	STIM2	STIM1	STIM0	0000 0000
25CH	EIE7	—	—	IOCCUIE	IOCBUIE	IOCAUIE	HLVDIE	RTCT1IE	RTCT0IE	----000
25DH	EIF7	—	—	IOCCUIF	IOCBUIF	IOCAUIF	HLVDIF	RTCT1IF	RTCT0IF	----000
25EH	IP7	—	—	PIOCC	PIOCB	PIOCA	PHLVD	PRTCT1	PRTCT0	----000
25FH	HLVDCTL	HLVDEN	VDIR	—	—	VDT3	VDT2	VDT1	VDT0	00--0000

## BANK3

Bank 3										
地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	复位初值
318H	RTCSRT	SRT7	SRT6	SRT5	SRT4	SRT3	SRT2	SRT1	SRT0	0000 0000
319H	RTCSTU	RTCALREN	—	RESET	LIF	HT	CNF	RTOFF	RTCLD	--00 0000
31CH	RTCALRS	ALRS7	ALRS6	ALRS5	ALRS4	ALRS3	ALRS2	ALRS1	ALRS0	-000 0000
31DH	RTCALRM	ALRM7	ALRM6	ALRM5	ALRM4	ALRM3	ALRM2	ALRM1	ALRM0	-000 0000
31EH	RTCALRH	ALRH7	—	ALRH5	ALRH4	ALRH3	ALRH2	ALRH1	ALRH0	--00 0000
31FH	RTCALRW	ALRW7	—	—	—	—	ALRW2	ALRW1	ALRW0	---- -000
320H	RTCTTR	RTCOE	RTCOS	—	—	—	RTCTTR2	RTCTTR1	RTCTTR0	00-- -000
321H	RTCFR	V7	V6	V5	V4	V3	V2	V1	V0	0000 0000
322H	RTCSEC	—	SEC6	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0	-000 0000
323H	RTCMIN	—	MIN6	MIN5	MIN4	MIN3	MIN2	MIN1	MIN0	-000 0000
324H	RTCHOUR	—	AMPM	HUR5	HUR4	HUR3	HUR2	HUR1	HUR0	-000 0000
325H	RTCWEK	—	—	—	—	—	WEK2	WEK1	WEK0	---- -000
326H	RTCDAY	—	—	DAY5	DAY4	DAY3	DAY2	DAY1	DAY0	--00 0000
327H	RTCMTH	—	—	—	MTH4	MTH3	MTH2	MTH1	MTH0	---0 0000
328H	RTCYEAR	YER7	YER6	YER5	YER4	YER3	YER2	YER1	YER0	0000 0000
32CH	RTCTMRCTL	CKST13	CKST12	CKST11	CKST10	CKST03	CKST02	CKST01	CKST00	0000 0000
32DH	RTCTMREN	—	—	—	—	—	—	TMR1EN	TMR0EN	---- --00
32EH	RTCTMR1	RTC 定时器 1 初值配置寄存器								0000 0000
32FH	RTCTMR0	RTC 定时器 0 初值配置寄存器								0000 0000
360H	POWCTL	—	—	—	—	—	—	DSL PEN	—	0000 0000
361H	PCAL	—	—	PCAL5	PCAL4	PCAL3	PCAL2	PCAL1	PCAL0	0001 1111
362H	XTALCAL	XTAL 校准寄存器								0101 0000
364H	PHCLR	—	—	—	—	—	—	EXTLFEN	INTLFEN	0000 0000
365H	LPRCCAL	内部低频晶振校准寄存器								0000 0000
366H	LPRCCTL	—	—	—	—	—	LPRC2	LPRC1	LPRC0	0000 0100
368H	BBODCTL	BBODEN	BSCAN	—	—	—	—	—	BBODF	0000 0000



## BANK4

BANK4										
地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	复位初值
410H	IOCAF	IOC7F	IOC6F	IOC5F	IOC4F	IOC3F	IOC2F	IOC1F	IOC0F	0000 0000
411H	IOCBF	IOC15F	IOC14F	IOC13F	IOC12F	IOC11F	IOC10F	IOC9F	IOC8F	0000 0000
412H	IOCCF	IOC23F	IOC22F	IOC21F	IOC20F	IOC19F	IOC18F	IOC17F	IOC16F	0000 0000
413H	IOCAPOS	IOC7P	IOC6P	IOC5P	IOC4P	IOC3P	IOC2P	IOC1P	IOC0P	0000 0000
414H	IOCBPOS	IOC15P	IOC14P	IOC13P	IOC12P	IOC11P	IOC10P	IOC9P	IOC8P	0000 0000
415H	IOCCPOS	IOC23P	IOC22P	IOC21P	IOC20P	IOC19P	IOC18P	IOC17P	IOC16P	0000 0000
416H	IOCANEG	IOC7N	IOC6N	IOC5N	IOC4N	IOC3N	IOC2N	IOC1N	IOC0N	0000 0000
418H	IOCBNEG	IOC15N	IOC14N	IOC13N	IOC12N	IOC11N	IOC10N	IOC9N	IOC8N	0000 0000
419H	IOCCNEG	IOC23N	IOC22N	IOC21N	IOC20N	IOC19N	IOC18N	IOC17N	IOC16N	0000 0000
41AH	IOCMOD						IOCCM	IOCBM	IOCAM	---- -000
41DH	T8CTL1							T8CS1	T8CS0	---- --00
41EH	T9CTL1							T9CS1	T9CS0	---- --00
424H	PA	—	—	—	—	—	—	PA1	PA0	xxxx xxxx
425H	PALR	—	—	—	—	—	—	PALR1	PALR0	xxxx xxxx
426H	TRA	—	—	—	—	—	—	TRA1	TRA0	1111 1111
427H	PURA	—	—	—	—	—	—	PURA1	PURA0	1111 1111
428H	ANSA	—	—	—	—	—	—	ANSA1	ANSA0	1111 1111
429H	PB	—	—	—	—	—	—	PB1	PB0	xxxx xxxx
42AH	PBLR	—	—	—	—	—	—	PBLR1	PBLR0	xxxx xxxx
42BH	TRB	—	—	—	—	—	—	TRB1	TRB0	1111 1111
42CH	PURB	—	—	—	—	—	—	PURB1	PURB0	1111 1111
42DH	ANSB	—	—	—	—	—	—	ANSB1	ANSB0	1111 1111
430H	RSCTL4	SPEN4	R4X9	SRXEN4	CRXEN4	ADREN4	FRER4	OVFER4	R4X9D	0000 0000
431H	TXSDR4	USART4 发送数据寄存器								xxxx xxxx
432H	RXSDR4	USART4 接收数据寄存器								xxxx xxxx
433H	BRCTL4	ABRDOV F4	RCIDLF4	SCKPS4	B4RG16	BR4CKS1	BR4CKS0	WUEN4	ABRDEN4	0100 0000
434H	TSCTL4	CSRS4	T4X9	TXEN4	SYNC4	SENDB4	HBRG4	TXSRS4	T4X9D	0000 0000
435H	EUBRGL4	USART4 波特率数据寄存器低字节								0000 0000
436H	EUBRGH4	USART4 波特率数据寄存器高字节								0000 0000
438H	UADMATC H4	ADM47	ADM46	ADM45	ADM44	ADM43	ADM42	ADM41	ADM40	0000 0000
43AH	UPINSET4	USLM4	—	—	—	—	—	—	—	0000 0000
440H	P3ASCTL	P3ASE	P3ASS2	P3ASS1	P3ASS0	P3SSAC1	P3SSAC0	P3SSBD1	P3SSBD0	

441H	P3A1RCTL	—	—	—	P3STRSYNC	P3STREND	P3STRENC	P3STRENB	P3STRENA	---0 0001
442H	PP3H	PWM3 周期设置寄存器高 8 位								0000 0000
443H	PP3L	PWM3 周期设置寄存器低 8 位								0000 0000
444H	PWM3CTL 0	P3M1	P3M0	—	—	PWM3M3	PWM3M2	PWM3M1	PWM3M0	0000 0000
445H	PWM3CTL 1	P3RSEN	P3DC6	P3DC5	P3DC4	P3DC3	P3DC2	P3DC1	P3DC0	0000 0000
446H	PWM3H0	PWM3 占空比缓冲寄存器高 8 位								0000 0000
447H	PWM3H1	PWM3 占空比缓冲寄存器高 8 位								0000 0000
448H	PWM3L0	PWM3 占空比设置寄存器低 8 位								0000 0000
449H	PWM3L1	PWM3 占空比设置寄存器高 8 位								0000 0000
44AH	T8CCRH	T8 启动 ADC 高 8 位设置寄存器								0000 0000
44BH	T8CCRL	T8 启动 ADC 低 8 位设置寄存器								0000 0000
44CH	T8CTL	—	T8CKBS3	T8CKBS2	T8CKBS1	T8CKBS0	T8ON	T8CKPS1	T8CKPS0	-000 0000
44DH	T8H	T8 高 8 位寄存器								0000 0000
44EH	T8L	T8 低 8 位寄存器								0000 0000
44FH	P4ASCTL	P4ASE	P4ASS2	P4ASS1	P4ASS0	P4SSAC1	P4SSAC0	P4SSBD1	P4SSBD0	0000 0000
450H	P4A1RCTL	—	—	—	P4STRSYNC	P4STREND	P4STRENC	P4STRENB	P4STRENA	---0 0001
451H	PP4H	PWM4 周期设置寄存器高 8 位								0000 0000
452H	PP4L	PWM4 周期设置寄存器低 8 位								0000 0000
453H	PWM4CTL 0	P4M1	P4M0	—	—	PWM4M4	PWM4M2	PWM4M1	PWM4M0	0000 0000
454H	PWM4CTL 1	P4RSEN	P4DC6	P4DC5	P4DC4	P4DC3	P4DC2	P4DC1	P4DC0	0000 0000
455H	PWM4H0	PWM4 占空比缓冲寄存器高 8 位								0000 0000
456H	PWM4H1	PWM4 占空比缓冲寄存器高 8 位								0000 0000
457H	PWM4L0	PWM4 占空比设置寄存器低 8 位								0000 0000
458H	PWM4L1	PWM4 占空比设置寄存器高 8 位								0000 0000
459H	T9CCRH	T9 启动 ADC 高 8 位设置寄存器								0000 0000
45AH	T9CCRL	T9 启动 ADC 低 8 位设置寄存器								0000 0000
45BH	T9CTL	—	T9CKBS3	T9CKBS2	T9CKBS1	T9CKBS0	T9ON	T9CKPS1	T9CKPS0	-000 0000
45CH	T9H	T9 高 8 位寄存器								0000 0000
45DH	T9L	T9 低 8 位寄存器								0000 0000
45EH	APFCTL2	P7CH3HSEL	P7CH2HSEL	P7CH1HSEL	P7CH0HSEL	P6CH3HSEL	P6CH2HSEL	P6CH1HSEL	P6CH0HSEL	0000 0000
45FH	APFCTL3	P3DSEL	P3CSEL1	P3CSEL0	P3BSEL1	P3BSEL0	P3ASEL	CCP7SEL1	CCP7SEL0	0000 0000
460H	APFCTL4	P4DSEL1	P4DSEL0	P4CSEL1	P4CSEL0	P4BSEL1	P4BSEL0	P4ASEL1	P4ASEL0	0000 0000
461H	APFCTL5	P5CSEL	P5BSEL1	P5BSEL0	P5ASEL1	P5ASEL0	CCP3SEL1	CCP3SEL0	T0CKSEL	0000 0000
462H	APFCTL6	CCP10SEL1	CCP10SEL0	CCP9SEL2	CCP9SEL1	CCP9SEL0	CCP8SEL1	CCP8SEL0	INTCCPSEL	0000 0000
463H	APFCTL7	IOC7SEL	IOC6SEL	IOC5SEL	IOC4SEL	IOC3SEL	IOC2SEL	IOC1SEL	IOC0SEL	0000 0000
464H	APFCTL8	IOC14SEL	IOC13SEL	IOC12SEL	IOC11SEL	IOC10SEL	T1CKSEL	T1GSEL1	T1GSEL0	0000 0000

465H	APFCTL9	SDIDA2S EL1	SDIDA2S EL0	SCKCL2 SEL	SDO2SEL1	SDO2SEL0	SS2SEL1	SS2SEL0	CLKOUTS EL	0000 0000
466H	APFCTL10	SDIDA1S EL1	SDIDA1S EL0	SCKCL1 SEL1	SCKCL1SEL0	SDO1SEL	SS1SEL	PWM2OUT SEL	PWM1OUT SEL	0000 0000
467H	APFCTL11	—	—	ADVRIN SEL1	ADVRINSEL 0	—	—	—	—	0000 0000
468H	APFCTL12	C3OUTS EL	C2OUTS EL1	C2OUTS EL0	C1OUTSEL	CK3TX3SE L	DT3RX3SE L	CK4TX4SE L	DT4RX4SE L	0000 0000

## BANK5

BANK5										
地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	复位初值
505H	ADCOFFSET	—	—	—	—	—	—	—	—	0000 0000
509H	P7ATRCTL	P7STR3H	—	P7STR2H	—	P7STR1H	—	P7STR0H	—	0000 0000
50CH	PP70H	PWM7 周期 0 寄存器高 8 位								0000 0000
50DH	PP70L	PWM7 周期 0 寄存器低 8 位								0000 0000
50EH	PP71H	PWM8 周期 1 寄存器高 8 位								0000 0000
50FH	PP71L	PWM8 周期 1 寄存器低 8 位								0000 0000
510H	PWM7CTL0	—	—	—	—	P7MOD3	P7MOD2	P7MOD1	P7MOD0	0000 0000
512H	PWM7CTL2	P7FUSES	—	P7UDEVT 3	P7UDEVT 2	P7UDEVT 1	P7UDEVT 0	P7UDEN1	P7UDEN0	1000 0000
515H	T5CTL1	T5MOD1	T5MOD0	T5OFS1	T5OFS0	T5TRIGEN	T5CLK1	T5CLK0	T5DIR	0000 0000
516H	保留	—	—	—	—	—	—	—	—	-000 0000
51BH	EIE6	—	WWDTIE	BCL2IE	SSCI2IE	RC4IE	TX4IE	RC3IE	TX3IE	0000 0000
51CH	EIF6	—	WWDTIF	BCL2IF	SSCI2IF	RC4IF	TX4IF	RC3IF	TX3IF	0000 0000
51DH	IP6	—	PWWDT	PBCL2	PSSCI2	PRC4	PTX4	PRC3	PTX3	0000 0000
550H	T5CTL0	—	T5CKBS3	T5CKBS2	T5CKBS1	T5CKBS0	T5ON	T5CKPS1	T5CKPS0	-000 0000
551H	T5H	T5 高 8 位								0000 0000
552H	T5L	T5 低字节寄存器								0000 0000
553H	T6CTL1	T6MOD1	T6MOD0	T6OFS1	T6OFS0	T6TRIGEN	T6CLK1	T6CLK0	T6DIR	0000 0000
554H	T6CTL0	T6SLPEN	T6CKBS3	T6CKBS2	T6CKBS1	T6CKBS0	T6ON	T6CKPS1	T6CKPS0	0000 0000
555H	T6H	T6 高 8 位								0000 0000
556H	T6L	T6 低字节寄存器								0000 0000
55AH	PWM7L0	捕捉/比较/PWM7 通道 0 占空比寄存器低 8 位								0000 0000
55BH	PWM7H0	捕捉/比较/PWM7 通道 0 占空比寄存器高 8 位								0000 0000
55CH	PWM7L1	PWM7 通道 1 占空比寄存器低 8 位								0000 0000
55DH	PWM7H1	PWM7 通道 1 占空比寄存器高 8 位								0000 0000
55EH	PWM7L2	PWM7 通道 2 占空比寄存器低 8 位								0000 0000
55FH	PWM7H2	PWM7 通道 2 占空比寄存器高 8 位								0000 0000
560H	TEMPSNR	TSEN	—	—	—	—	—	—	—	0--- ----
563H	T5CCR0H	T5 启动 ADC 设置寄存器 0 高 8 位								0000 0000
564H	T5CCR0L	T5 启动 ADC 设置寄存器 0 低 8 位								0000 0000
565H	T5CCR1H	T5 启动 ADC 设置寄存器 1 高 8 位								0000 0000
566H	T5CCR1L	T5 启动 ADC 设置寄存器 1 低 8 位								0000 0000
567H	T6CCR0H	T6 启动 ADC 设置寄存器 0 高 8 位								0000 0000
568H	T6CCR0L	T6 启动 ADC 设置寄存器 0 低 8 位								0000 0000

56AH	PWM7L3	PWM7 通道 3 占空比寄存器低 8 位								0000 0000
56BH	PWM7H3	PWM7 通道 3 占空比寄存器高 8 位								0000 0000
56DH	PWM7PC	P7PC3H	—	P7PC2H	—	P7PC1H	—	P7PC0H	—	0000 0000

注：“—”表示未用的存储单元 “x”表示不定

## BANK6

BANK6										
地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	复位初值
620H	PWM5H2	PWM5 通道 2 占空比寄存器高 8 位								0000 0000
621H	PWM5H3	PWM5 通道 3 占空比寄存器高 8 位								0000 0000
622H	PWM5L0	捕捉比较模式时, PWM5L0=CCP 数据寄存器低 8 位 PWM5 模式时, PWM5L0= PWM5 通道 1 占空比寄存器低 8 位								0000 0000
623H	PWM5L1	捕捉比较模式时, PWM5L1=CCP 数据寄存器高 8 位 PWM5 模式时, PWM5L1= PWM5 通道 1 低 8 位占空比缓冲寄存器 (与用户无关)								0000 0000
624H	PWM5L2	PWM5 通道 2 占空比寄存器低 8 位								0000 0000
625H	PWM5L3	PWM5 通道 3 占空比寄存器低 8 位								0000 0000
626H	PWM5OC	—	—	OCA3	OCB3	OCA2	OCB2	OCA	OCB	--00 0000
627H	PWM5PC	—	—	PCA3	PCB3	PCA2	PCB2	PCA	PCB	--00 0000
628H	T10CTL1	T10MOD1	T10MOD0	T10OFS1	T10OFS0	T10TRIGEN	T10ON	—	T10DIR	0000 00—X
629H	T10CCR0H	T10 触发 ADC 启动设置寄存器 0 高 8 位								0000 0000
62AH	T10CCR0L	T10 触发 ADC 启动设置寄存器 0 低 8 位								0000 0000
62BH	T10CCR1H	T10 触发 ADC 启动设置寄存器 1 高 8 位								0000 0000
62CH	T10CCR1L	T10 触发 ADC 启动设置寄存器 1 低 8 位								0000 0000
62DH	T10CTL0	T10CKBS3	T10CKBS2	T10CKBS1	T10CKBS0	T10CS1	T10CS0	T10CKPS1	T10CKPS0	0000 0000
62EH	T10L	T10 低字节寄存器								0000 0000
62FH	T10H	T10 高字节寄存器								0000 0000
630H	EIF8			T10IF	T9IF	T8IF	CCP10IF	CCP9IF	CCP8IF	--00 0000
631H	EIE8			T10IE	T9IE	T8IE	CCP10IE	CCP9IE	CCP8IE	--00 0000
632H	IP8			PT10	PT9	PT8	PCCP10	PCCP9	PCCP8	--00 0000
633H	ADCCTL3					T10CCR1ON	T10CCR0ON	T9CCRON	T8CCRON	---- 0000
634H	P5ASCTL0	P5ASE	P5ASS2	P5ASS1	P5ASS0	P5SSAC1	P5SSAC0	P5SSBD1	P5SSBD0	0000 0000
635H	P5ASCTL1	P5SSA31	P5SSA30	P5SSB31	P5SSB30	P5SSA21	P5SSA20	P5SSB21	P5SSB20	0000 0000
636H	P5STRCTL0	—	—	—	P5STRSYNC	P5STREND	P5STRENC	P5STRENB	P5STRENA	---0 0001
637H	P5STRCTL1	—	—	—	—	P5STRENA3	P5STRENB3	P5STRENA2	P5STRENB2	---- 1010
638H	PP5H	PWM5 周期寄存器高 8 位								0000 0000
639H	PP5L	PWM5 周期寄存器低 8 位								0000 0000
63AH	PWM5CTL0	P5CH1MOD1	P5CH1MOD0	P5CH3MOD	P5CH2MOD	P5MOD3	P5MOD2	P5MOD1	P5MOD0	0000 0000
63BH	PWM5CTL1	P5RSEN	P5DC6	P5DC5	P5DC4	P5DC3	P5DC2	P5DC1	P5DC0	0000 0000
63CH	PWM5CTL2	—	—	P5FUSES	—	—	UDEVT1	UDEVT0	UDEN	--1- -000

63DH	PWM5FC	—	—	FCA3	FCB3	FCA2	FCB2	FCA	FCB	--00 0000
63EH	PWM5H0	PWM5 通道 1 占空比寄存器高 8 位								0000 0000
63FH	PWM5H1	PWM5 通道 1 高 8 位占空比缓冲寄存器（与用户无关）								0000 0000

## 附录 2 汇编指令集

助记符、操作数	指令格式	指令说明	周期	影响标志
NOP	0000_0000_0000_0000	空操作指令	1	
NOPZ	1111_1111_1111_1111	空操作指令	1	
CRET	0000_0000_0000_1000	子程序返回指令	2	
RRET Rn,#data	1011_0rrr_kkkk_kkkk	立即数送到 Rn 中返回	2	
IRET	0000_0000_0000_1001	中断返回指令	2	
CWDT	0000_0000_0110_0100	WDT 清 0	1	
IDLE	0000_0000_0110_0011	进入休眠模式	1	
<b>数据传送指令</b>				
MOV dir	0000_1111_ffff_ffff	dir←(dir)	1	Z
MOV Rn,dir	0101_rrr0_fff_fff ? ?	Rn←(dir)	1	
MOV dir,Rn	0101_rrr1_fff_fff	dir←(Rn)	1	
MOV Rn,#data	1001_1rrr_kkkk_kkkk	Rn←data	1	
MOV Rn,Rs	1111_1000_11ss_srrr	Rn←(Rs)	1	
LD Rn,[Rs]	1111_0111_00ss_srrr	Rn←((Rs))	1	
ST [Rn],Rs	1111_0111_01ss_srrr	(Rn)←(Rs)	1	
SWAPR Rn,dir	0100_rrr0_fff_fff	Rn<7:4>=dir<3:0> Rn<3:0>=dir<7:4>	1	
SWAP dir	0100_rrr1_fff_fff	dir<7:4>=dir<3:0> dir<3:0>=dir<7:4>	1	
MOVB #data	1110_0001_kkkk_kkkk	BANK←data	1	
MOVP #data	1110_0000_kkkk_kkkk	PCH←data	1	
<b>算术运算指令</b>				
ADD Rm,dir	0010_0rr0_fff_fff	Rm←(Rm)+(dir)	1	CY、DC、Z
ADD dir,Rm	0010_0rr1_fff_fff	dir←(Rm)+(dir)	1	CY、DC、Z
ADD Rn,#data	1000_0rrr_kkkk_kkkk	Rn←(Rn)+data	1	CY、DC、Z
ADD Rn,Rs	1111_1000_00ss_srrr	Rn←(Rn)+(Rs)	1	CY、DC、Z
SUB Rm,dir	0011_1rr0_fff_fff	Rm←(dir)-(Rm)	1	CY、DC、Z
SUB dir,Rm	0011_1rr1_fff_fff	dir←(dir)-(Rm)	1	CY、DC、Z
SUB Rn,#data	1010_0rrr_kkkk_kkkk	Rn←data-(Rn)	1	CY、DC、Z
SUB Rn,Rs	1111_1000_01ss_srrr	Rn←(Rs)-(Rn)	1	CY、DC、Z
CMP Rn,#data	1111_0010_1kkk_krrr	-	1	CY、DC、Z
CMP Rn,Rs	1111_0001_10ss_srrr	-	1	CY、DC、Z
INC dir	0000_1011_fff_fff	dir←(dir)+1	1	Z
INCR dir	0000_1010_fff_fff	R0←(dir)+1	1	Z
INC Rn (先读后写)	1111_1111_0001_0rrr	Rn←(Rn)+1	1	Z
DEC dir	0000_0111_fff_fff	dir←(dir)-1	1	Z
DECR dir	0000_0110_fff_fff	R0←(dir)-1	1	Z
DEC Rn	1111_1111_0000_1rrr	Rn←(Rn)-1	1	Z



助记符、操作数	指令格式	指令说明	周期	影响标志
<b>逻辑运算指令</b>				
AND Rm,dir	0010_1rr0_ffff_ffff	Rm←(Rm)∩(dir)	1	Z
AND dir,Rm	0010_1rr1_ffff_ffff	dir←(dir)∩(Rm)	1	Z
AND Rn,#data	1000_1rrr_kkkk_kkkk	Rn←(Rn)∩data	1	Z
AND Rn,Rs	1111_1000_10ss_srrr	Rn←(Rn)∩(Rs)	1	Z
ORL Rm,dir	0011_0rr0_ffff_ffff	Rm←(Rm)∪(dir)	1	Z
ORL dir,Rm	0011_0rr1_ffff_ffff	dir←(dir)∪(Rm)	1	Z
ORL Rn,#data	1001_0rrr_kkkk_kkkk	Rn←(Rn)∪data	1	Z
ORL Rn,Rs	1111_1001_00ss_srrr	Rn←(Rn)∪(Rs)	1	Z
XOR Rm,dir	0001_1rr0_ffff_ffff	Rm←(Rm)⊕(dir)	1	Z
XOR dir,Rm	0001_1rr1_ffff_ffff	dir←(dir)⊕(Rm)	1	Z
XOR Rn,#data	1010_1rrr_kkkk_kkkk	Rn←(Rn)⊕data	1	Z
XOR Rn,Rs	1111_1001_01ss_srrr	Rn←(Rn)⊕(Rs)	1	Z
CLR Rn	0000_0010_xxxx_1rrr	Rn=0	1	Z
CLR dir	0000_0011_ffff_ffff	dir=0	1	Z
CPLR dir	0000_0100_ffff_ffff	R0←!(dir)	1	Z
CPL dir	0000_0101_ffff_ffff	dir←!(dir)	1	Z
CPL Rn	1111_1111_0000_0rrr	Rn←!(Rn)	1	Z
RRCR dir	0001_0000_ffff_ffff	R0←(dir) 带进位 C 循环右移 1 位	1	CY
RRC dir	0001_0001_ffff_ffff	dir←(dir) 带进位 C 循环右移 1 位	1	CY
RRC Rn	1111_1111_0010_0rrr	Rn←(Rn) 带进位 C 循环右移 1 位	1	CY
RLCR dir	0001_0010_ffff_ffff	R0←(dir)带进位 C 循环左移 1 位	1	CY
RLC dir	0001_0011_ffff_ffff	dir←(dir)带进位 C 循环左移 1 位	1	CY
RLC Rn	1111_1111_0001_1rrr	Rn←(Rn) 带进位 C 循环左移 1 位	1	CY
<b>位操作指令</b>				
CLR dir,b	0110_0bbb_ffff_ffff	将 dir 的 b 位清 0	1	
SET dir,b	0110_1bbb_ffff_ffff	将 dir 的 b 位置 1	1	
CLR Rn,b	1111_1110_00bb_brrr	将 Rn 的 b 位清 0	1	
SET Rn,b	1111_1110_01bb_brrr	将 Rn 的 b 位置 1	1	
<b>转移指令</b>				
DECRJZ dir	0000_1000_ffff_ffff	R0←(dir)-1,为 0 跳过下一条指令	1/2	
DECJZ dir	0000_1001_ffff_ffff	dir←(dir)-1,为 0 跳过下一条指令	1/2	
DECJZ Rn	1111_1111_0101_1rrr	Rn←(Rn)-1,为 0 跳过下一条指令	1/2	
INCRJZ dir	0000_1100_ffff_ffff	R0←(dir)+1,为 0 跳过下一条指令	1/2	
INCJZ dir	0000_1101_ffff_ffff	dir←(dir)+1,为 0 跳过下一条指令	1/2	
INCJZ Rn	1111_1111_0101_0rrr	Rn←(Rn)+1,为 0 跳过下一条指令	1/2	
JNB dir,b	0111_0bbb_ffff_ffff	dir 的 b 位为 0 跳过下一条指令	1/2	
JB dir,b	0111_1bbb_ffff_ffff	dir 的 b 位为 1 跳过下一条指令	1/2	
JNB Rn,b	1111_0111_10bb_brrr	Rn 的 b 位为 0 跳过下一条指令	1/2	
JB Rn,b	1111_0111_11bb_brrr	Rn 的 b 位为 1 跳过下一条指令	1/2	
JMP #data12	1100_kkkk_kkkk_kkkk	无条件转移指令	2	
CALL #data12	1101_kkkk_kkkk_kkkk	子程序调用指令	2	

注: dir 为通用寄存器或特殊功能寄存器; Rn、Rs 表示 R0~R7; Rm 表示 R0~R3;  
#data 表示 8 位立即数; #data12 表示 12 位立即数; b 表示寄存器的第 b 位; [Rn]表示 Rn  
中的数值指向的地址中数据 ; ()表示特殊功能寄存器、通用数据寄存器或寄存器组中的数  
据。

## 附录 3 全双工异步模式的典型波特率和误差值

SYNC=0,HBRG=0,BRG16=0									
波特率 时钟频率		300	1200	2400	9600	10417	19.2K	57.6K	116.2K
20.000MHz	实际波特率	—	—	2404	9615	10417	—	—	—
	误差(%)	—	—	0.16	0.16	0.00	—	—	—
	EUBRGL 值(十六进制)	—	—	81H	20H	1DH	—	—	—
16.000MHz	实际波特率	—	1202	2404	9615	10417	19231	—	—
	误差(%)	—	0.16	0.16	0.16	0.00	0.16	—	—
	EUBRGL 值(十六进制)	—	CFH	67H	19H	17H	0CH	—	—
8.000MHz	实际波特率	—	1202	2404	9615	10417	—	—	—
	误差(%)	—	0.16	0.16	0.16	0.00	—	—	—
	EUBRGL 值(十六进制)	—	67H	33H	0CH	0BH	—	—	—
4.000MHz	实际波特率	300	1202	2404	—	10417	—	—	—
	误差(%)	0.16	0.16	0.16	—	0.00	—	—	—
	EUBRGL 值(十六进制)	6BH	33H	19H	—	05H	—	—	—
2.000MHz	实际波特率	300	1202	2404	—	10417	—	—	—
	误差(%)	0.16	0.16	0.16	—	0.00	—	—	—
	EUBRGL 值(十六进制)	67H	19H	0CH	—	02H	—	—	—
1.000MHz	实际波特率	300	1202	—	—	—	—	—	—
	误差(%)	0.16	0.16	—	—	—	—	—	—
	EUBRGL 值(十六进制)	33H	0CH	—	—	—	—	—	—
512.00 KHz	实际波特率	307	—	—	—	—	—	—	—
	误差(%)	2.33	—	—	—	—	—	—	—
	EUBRGL 值(十六进制)	19H	—	—	—	—	—	—	—

SYNC=0,HBRG=1,BRG16=0									
波特率 时钟频率		300	1200	2400	9600	10417	19.2K	57.6K	116.2K
20.000MHz	实际波特率	—	—	—	9541	10417	19230	56818	113.6K
	误差(%)	—	—	—	-0.6	0.00	0.16	-1.35	0.03
	EUBRGL 值(十六进制)	—	—	—	82H	77H	41H	15H	0AH

16.000MHz	实际波特率	—	—	—	9615	10417	19231	58824	111.1K
	误差(%)	—	—	—	0.16	0.00	0.16	2.12	3.55
	EUBRGL 值(十六进制)	—	—	—	67H	5FH	33H	10H	08H
8.000MHz	实际波特率	—	—	2404	9615	10417	19231	55556	—
	误差(%)	—	—	0.16	0.16	0.00	0.16	-3.55	—
	EUBRGL 值(十六进制)	—	—	CFH	33H	2FH	19H	08H	—
4.000MHz	实际波特率	—	1202	2404	9615	10417	19.23K	—	—
	误差(%)	—	0.16	0.16	0.16	0.00	0.16	—	—
	EUBRGL 值(十六进制)	—	CFH	67H	19H	17H	0CH	—	—
2.000MHz	实际波特率	—	1202	2404	9615	10417	—	—	—
	误差(%)	—	0.16	0.16	0.16	0.00	—	—	—
	EUBRGL 值(十六进制)	—	67H	33H	0CH	0BH	—	—	—
1.000MHz	实际波特率	300	1202	2404	—	10417	—	—	—
	误差(%)	0.16	0.16	0.16	—	0.00	—	—	—
	EUBRGL 值(十六进制)	CFH	33H	19H	—	05H	—	—	—
512.000KHz	实际波特率	301	1231	2462	—	—	—	—	—
	误差(%)	0.63	2.58	2.58	—	—	—	—	—
	EUBRGL 值(十六进制)	69H	19H	0CH	—	—	—	—	—

SYNC=0,HBRG=0,BRG16=1									
波特率 时钟频率		300	1200	2400	9600	10417	19.2K	57.6K	116.2K
20.000MHz	实际波特率	—	—	—	9541	10417	19230	56818	113.6K
	误差(%)	—	—	—	-0.6	0.00	0.16	-1.35	0.03
	EUBRGL 值 (十六进制)	—	—	—	82H	77H	41H	15H	0AH
16.000MHz	实际波特率	—	—	—	9615	10417	19231	58824	111.1K
	误差(%)	—	—	—	0.16	0.00	0.16	2.12	3.55
	EUBRGL 值 (十六进制)	—	—	—	67H	5FH	33H	10H	08H
8.000MHz	实际波特率	299.9	1999	2404	9615	10417	19231	55556	—
	误差(%)	-0.02	-0.08	0.16	0.16	0.00	0.16	-3.55	—
	EUBRGL 值 (十六进制)	0682H	01A0H	CFH	33H	3FH	19H	08H	—
4.000MHz	实际波特率	300.1	1202	2404	9615	10417	19.23K	—	—
	误差(%)	0.04	0.16	0.16	0.16	0.00	0.16	—	—
	EUBRGL 值 (十六进制)	0340H	CFH	67H	19H	17H	0CH	—	—
2.000MHz	实际波特率	299.8	1202	2404	9615	10417	—	—	—
	误差(%)	-0.18	0.16	0.16	0.16	0.00	—	—	—
	EUBRGL 值 (十六进制)	01A0H	33H	33H	0BH	0BH	—	—	—
1.000MHz	实际波特率	300	1202	2404	—	10417	—	—	—
	误差(%)	0.16	0.16	0.16	—	0.00	—	—	—
	EUBRGL 值 (十六进制)	CFH	33H	19H	—	05H	—	—	—
512.000KHz	实际波特率	301	1231	2462	—	—	—	—	—
	误差(%)	0.63	2.58	2.58	—	—	—	—	—
	EUBRGL 值 (十六进制)	69H	19H	0CH	—	—	—	—	—

SYNC=0,HBRG=1,BRG16=1 或 SYNC=1, BRG16=1									
波特率 时钟频率		300	1200	2400	9600	10417	19.2K	57.6K	116.2K
20.000MHz	实际波特率	300	1200	2399	9597	10417	19157	57.47K	113.6K
	误差(%)	0.00	0.00	0.03	0.03	0.00	0.22	0.22	0.03
	EUBRGH : EUBRGL	411AH	1046H	0823H	0208H	01DFH	0104H	0056H	002BH
16.000MHz	实际波特率	300	1200	2401	9615	10417	19.23K	57971	117.6K
	误差(%)	0.00	0.01	0.04	0.16	0.00	0.16	0.64	2.12
	EUBRGH : EUBRGL	3414H	0D04H	0681H	019FH	017FH	00CFH	0044H	0021H
8.000MHz	实际波特率	300.0	1200	2401	9615	10417	19.23K	57.14K	117.6K
	误差(%)	0.00	-0.02	0.04	0.16	0.00	0.16	-0.79	2.12
	EUBRGH: EUBRGL	1A0A H	0682H	0340H	00CFH	00BFH	0067H	0022H	0010H
4.000MHz	实际波特率	300.0	1200	2398	9615	10417	19.23K	56.82K	111.1K
	误差(%)	0.01	0.04	0.08	0.16	0.00	0.16	2.12	-3.55
	EUBRGH: EUBRGL	0D04H	0340H	01A0H	0067H	005FH	0033H	0010H	0008H
2.000MHz	实际波特率	299.9	1199	2404	9615	10417	19.23K	55.56K	—
	误差(%)	-0.02	-0.08	0.16	0.16	0.00	0.16	-3.55	—
	EUBRGH: EUBRGL	0682H	01A0H	00CFH	0033H	002FH	0019H	0008H	—
1.000MHz	实际波特率	300.1	1202	2404	9615	10417	19.23K	—	—
	误差(%)	0.04	0.16	0.16	0.16	0.00	0.16	—	—
	EUBRGH: EUBRGL	0340H	00CFH	0067H	0019H	0017H	000CH	—	—
512.000KH z	实际波特率	300.5	1208	2415	9846	10666	—	—	—
	误差(%)	0.16	0.63	0.63	2.56	2.40	—	—	—
	EUBRGH: EUBRGL	01A9H	0069H	0034H	000CH	000BH	—	—	—

注：以上表格中波特率寄存器的值均是十六进制的数值形式

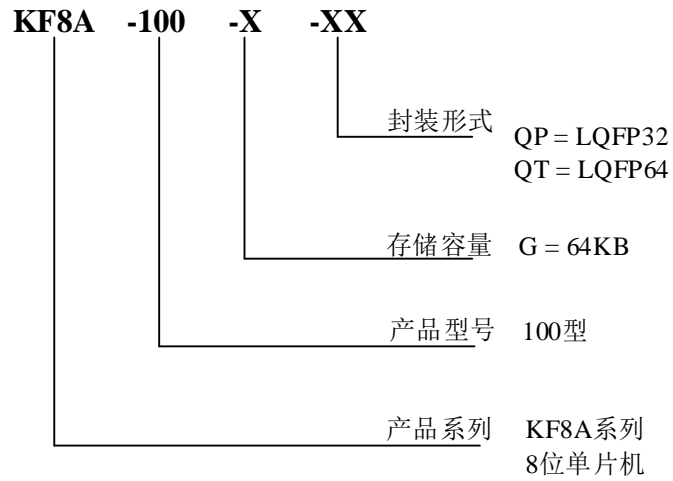
## 附录 4 版本信息

版本号	更新内容	日期
V1.7	<ol style="list-style-type: none"> <li>1. 电气规范章节添加 ESD 和 latch up 小节;</li> <li>2. 更新产品标识体系;</li> <li>3. 添加 P0IF 和 P3IF 清零条件备注说明;</li> <li>4. 更新 CCP2/5 模块章节和 APFCTL2 寄存器关于 CCP 输入引脚配置的说明;</li> <li>5. OPTR 寄存器添加 PSA 位相关备注说明;</li> <li>6. 更新 T0 原理框图;</li> <li>7. T1 重载模式添加关于 T1IF 和 T1ON 关系的备注说明;</li> <li>8. 修改寄存器名称错误, 将部分章节内的 OSCCTL 修改为 SCLKCTL;</li> <li>9. 更新 CAN 总线-发送缓冲器-标准帧格式 SFF 数据字节表描述错误;</li> <li>10. 屏蔽 FVR 和 HLVD 相关信息。</li> </ol>	2019-08-14
V1.8	<ol style="list-style-type: none"> <li>1. 滤波器 x 采样时钟分频寄存器章节中, 寄存器名称“FILT<sub>x</sub>PRE”改为“FILT<sub>x</sub>PER”;</li> <li>2. 附录 1 和复用引脚相关寄存器表中, 修改寄存器 APFCTL2 位名错误;</li> <li>3. 更新页眉信息;</li> <li>4. 封装管脚图上更新订货号全称;</li> </ol>	2019-11-06
V1.9	<ol style="list-style-type: none"> <li>1. 更新封装信息中 LQFP64 的 E 和 E1 位置标注。</li> </ol>	2020-09-04
V2.0	更新 flash 自写工作温度范围	2020-10-19
V2.1	更新储存温度范围	2020-12-30
V2.2	更新引脚描述	2021-03-11
V2.3	更新引脚功能描述	2021-03-16
V2.4	<ol style="list-style-type: none"> <li>1. 更新商标;</li> <li>2. 增加 USART 自动波特率检测注释第 4 条描述;</li> <li>3. 外部晶振框图增加晶振跨阻及其描述;</li> <li>4. 新增外部晶振校准寄存器;</li> <li>5. 新增系统时钟描述: 最高 16MHz;</li> <li>6. 新增 AD 关于小信号 (低于 350mV) 的注释;</li> <li>7. 新增 PWM1/2 组合成 16 位 PWM 的注释;</li> <li>8. 修改 POR 的电气特性;</li> <li>9. 修改工作电压为 2.7~5.5V, 并更新相关电气特性;</li> <li>10. 新增系统时钟寄存器的注释, 频率不能低于 4MHz;</li> <li>11. 修改 CCP8/9 的捕捉/比较对象为 PWM<sub>y</sub>H0:PWM<sub>y</sub>L0。</li> </ol>	2022-07-18
V2.5	<ol style="list-style-type: none"> <li>1. 删除 KF8A100GQS 订货号;</li> <li>2. 新增 AEC-Q100 描述;</li> </ol>	2022-09-09
V2.6	<ol style="list-style-type: none"> <li>删除硬件乘法器模块和硬件除法器模块的描述;</li> <li>删除 PWM5/PWM6/PWM7 中心对齐模式相关描述;</li> </ol>	2022-10-11
V2.7	<ol style="list-style-type: none"> <li>1. 删除 SR 章节和描述;</li> </ol>	2023-06-02

	<ul style="list-style-type: none"> <li>2.删除 CAN 章节和描述;</li> <li>3.修改 CCP2 和 CCP5 描述, 交换位置;</li> <li>4.修改 ADC 章节为 44 路采样通道;</li> <li>5.删除其余没有删掉的超低功耗描述;</li> <li>6.修改 FLASH 写寄存器 NVMADDRH 为高 7 位;</li> <li>7.修改配置位 BOR 和 POR 描述;</li> <li>8.修改深度休眠功耗最大值为 8uA;</li> </ul>	
V2.8	新增 AD 电气特性: AD 通道对地阻抗;	2023-07-10
V2.9	添加电气特性章节的上电/掉电说明。	2023-09-09
V3.0	<ul style="list-style-type: none"> <li>1.删除看门狗与 PSA 寄存器冲突的说明;</li> <li>2.删除不允许占空比实时更新的说明;</li> <li>3.删除不建议用小信号采样的说明;</li> <li>4.删除 PLL 章节和描述;</li> <li>5.修改产品订购信息中内部 HFOSC, 仅保留 16M;</li> <li>6.修改 XTALCAL 寄存器偏置电流档位注解错误;</li> <li>7.修改高频外设时钟 HFCLK 关于“HFCKS”位的描述, 删除有关 PLL 的描述;</li> <li>8.修改振荡器框图, 删除 PLL 模块;</li> <li>9.屏蔽 T0 外设时钟源选择, 及相关描述;</li> <li>10.修改普通比较模式的相关错误描述;</li> <li>11.修改电气参数: 深度休眠模式电流最大值为 20uA;</li> <li>12.修改电气参数: IIL 输入漏电流为 ±2uA;</li> <li>13.修改电气参数: POR 范围为(2.0,2.7)V, 标注测试条件为 25°C;</li> <li>14.修改电气参数: LVR 范围为(1.7,2.1)V, 标注测试条件为 25°C;</li> <li>15.屏蔽 POWER18 及相关描述;</li> <li>16.增加文档说明;</li> <li>17.增加寄存器描述;</li> <li>18.增加串口波特率“倍频器 m 选择表”的注释;</li> <li>19.修改波特率误差参考附录描述错误, 由附录 4 修改为附录 3。</li> </ul>	2024-05-23



## 产品标识体系



以 KF8A100GQT 为例:

KF8A 系列 8 位单片机，100 型，存储容量 64KB，LQFP64 封装。

## **RoHS 认证**

本产品已通过 RoHS 检测。

## 声明及销售网络

### 销售及服务网点

上海 TEL:021-50275927

地址 上海浦东张江集电港龙东大道 3000 号 1 幢 906 室 B1 座