8 位微控制器

KF8F3110/12/20/22/30/32
数据手册

产品订购信息

型号	订货号	封装	FLASH	RAM	FLASH 自写	DATA EEPROM	内部 OSC	外部 HF/LFOSC	8 位定时器	16 位定时器	8 位 PWM	16 位 PWM	CCP	内部参考 (V)	AMP	CMP	12 位 ADC	USART	SSCI (SPI/ I2C)	RTC	工作电压 (V)	内核版本
KF8F3110	KF8F3110SB	SOIC-8	8KB	528B	Y	Y	4MHz	16MHz/32.768kHz	1	2	2	2	/	2/3/4	N	1	3	1	/	Y	2.1~5.5	V2
	KF8F3110SD	SOIC-14	8KB	528B	Y	Y	4MHz	16MHz/32.768kHz	1	3	2	4	1	2/3/4	N	2	10	1	1	Y	2.1~5.5	V2
	KF8F3110TD	TSSOP-14	8KB	528B	Y	Y	4MHz	16MHz/32.768kHz	1	3	2	4	1	2/3/4	N	2	10	1	1	Y	2.1~5.5	V2
KF8F3112	KF8F3112SE	SOIC-16	8KB	528B	Y	128B	4MHz	16MHz/32.768kHz	1	3	2	4	1	2/3/4	1	2	12	1	1	Y	2.1~5.5	V2
	KF8F3112OG	SSOP-20	8KB	528B	Y	128B	4MHz	16MHz/32.768kHz	1	3	2	4	1	2/3/4	1	2	14	1	1	Y	2.1~5.5	V2
	KF8F3112SG	SOIC-20	8KB	528B	Y	128B	4MHz	16MHz/32.768kHz	1	3	2	4	1	2/3/4	1	2	14	1	1	Y	2.1~5.5	V2
KF8F3120	KF8F3120SD	SOIC-14	8KB	528B	Y	128B	8MHz	16MHz/32.768kHz	1	3	2	4	1	2/3/4	N	2	10	1	1	Y	2.1~5.5	V2
KF8F3122	KF8F3122SE	SOIC-16	8KB	528B	Y	128B	8MHz	16MHz/32.768kHz	1	3	2	4	1	2/3/4	1	2	12	1	1	Y	2.1~5.5	V2
	KF8F3122OG	SSOP-20	8KB	528B	Y	128B	8MHz	16MHz/32.768kHz	1	3	2	4	1	2/3/4	1	2	14	1	1	Y	2.1~5.5	V2
	KF8F3122SG	SOIC-20	8KB	528B	Y	128B	8MHz	16MHz/32.768kHz	1	3	2	4	1	2/3/4	1	2	14	1	1	Y	2.1~5.5	V2
	KF8F3122NG	QFN-20	8KB	528B	Y	128B	8MHz	16MHz/32.768kHz	1	3	2	4	1	2/3/4	1	2	14	1	1	Y	2.1~5.5	V2
KF8F3130	KF8F3130SD	SOIC-14	8KB	528B	Y	128B	16MHz	16MHz/32.768kHz	1	3	2	4	1	2/3/4	N	2	10	1	1	Y	2.1~5.5	V2
	KF8F3130TD	TSSOP-14	8KB	528B	Y	128B	16MHz	16MHz/32.768kHz	1	3	2	4	1	2/3/4	N	2	10	1	1	Y	2.1~5.5	V2
KF8F3132	KF8F3132SE	SOIC-16	8KB	528B	Y	128B	16MHz	16MHz/32.768kHz	1	3	2	4	1	2/3/4	1	2	12	1	1	Y	2.1~5.5	V2
	KF8F3132OG	SSOP-20	8KB	528B	Y	128B	16MHz	16MHz/32.768kHz	1	3	2	4	1	2/3/4	1	2	14	1	1	Y	2.1~5.5	V2
	KF8F3132SG	SOIC-20	8KB	528B	Y	128B	16MHz	16MHz/32.768kHz	1	3	2	4	1	2/3/4	1	2	14	1	1	Y	2.1~5.5	V2

注：KF8F3110SB 无 T2 资源。

KF8F3110/12/20/22/30/32 芯片使用注意事项

芯片的 ESD 防护措施

KF8F3110/12/20/22/30/32 芯片提供 8KV 的 ESD 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、抗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或手套，不能用手直接接触芯片等。

芯片的 EFT 防护措施

KF8F3110/12/20/22/30/32 芯片提供 4.2KV 的 EFT 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 之间并接两个电容，大小分别为 105 和 102，电容尽量靠近芯片的 VDD 引脚。

芯片的焊接

KF8F3110/12/20/22/30/32 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

KF8F3110/12/20/22/30/32 芯片提供独立电源管脚。当 KF8F3110/12/20/22/30/32 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性能退化。

芯片的复位

KF8F3110/12/20/22/30/32 芯片提供内部上电复位。对于不同的快速上电/断电或慢速上电/断电系统，内部上电复位电路可能失效，建议用户使用外部复位、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

KF8F3110/12/20/22/30/32 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

KF8F3110/12/20/22/30/32 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

KF8F3110/12/20/22/30/32 芯片提供宽范围的输入管脚电平，用户输入高电平应大于 VIH 的最小值，低电平应小于 VIL 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

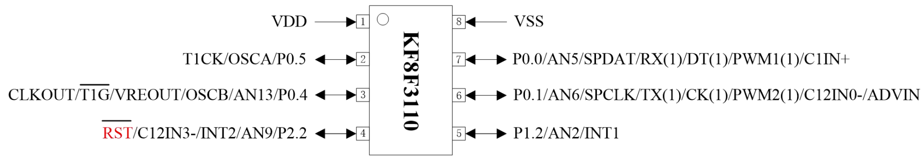
KF8F3110/12/20/22/30/32 芯片提供低功耗设计模式，用户在实际应用中可根据应用系统的要求采用各种不同的低功耗模式，包括系统工作时钟的选择和休眠模式的选择等等。

芯片的开发环境

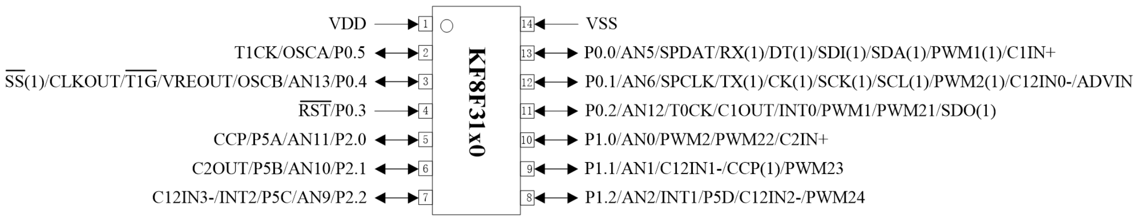
KF8F3110/12/20/22/30/32 芯片提供完整的软/硬件开发环境，并受知识产权保护。选择上海芯旺微电子技术有限公司指定的汇编器、编译器、编程器、硬件仿真器，必须遵循与芯片相关的规定和说明。

引脚示意图

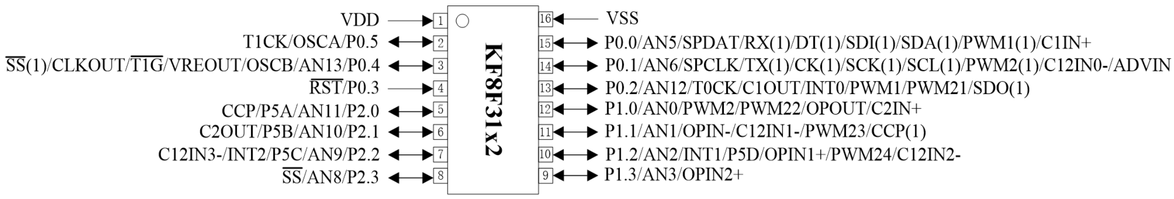
8 引脚示意图：



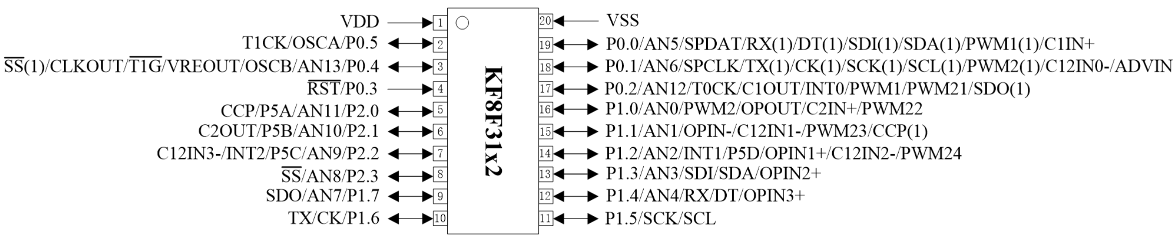
14 引脚示意图：



16 引脚示意图：



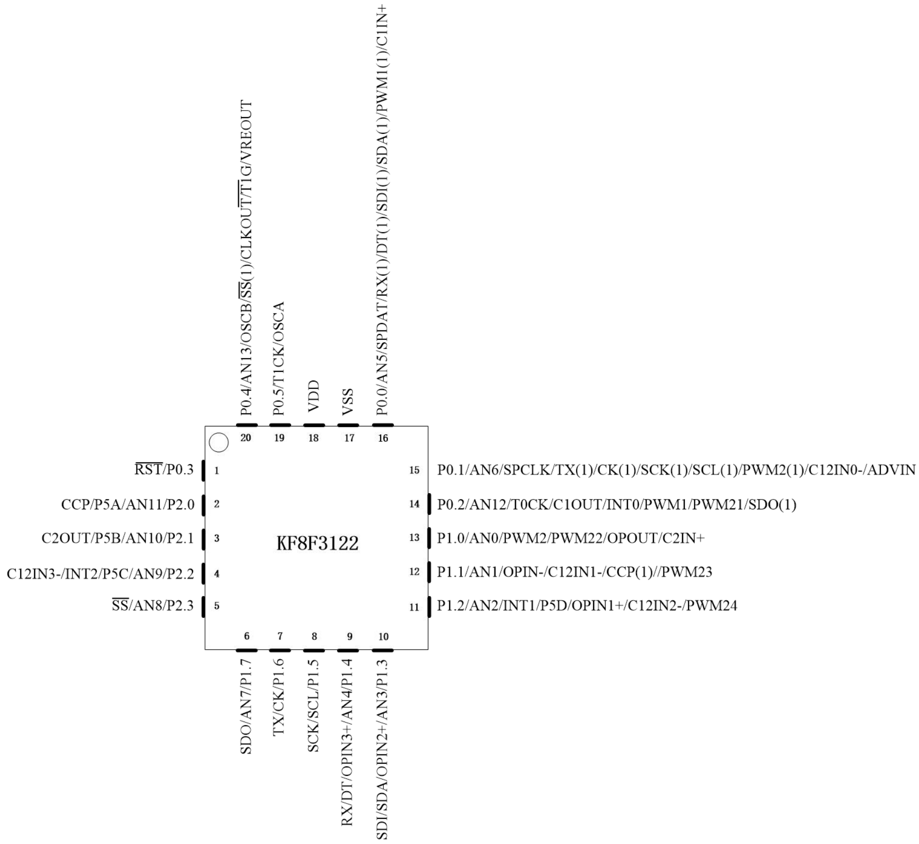
SOIC/SSOP-20 引脚示意图：



注：

1. 管脚图中带（1）标注的外设管脚为可选切换管脚，具体操作在相应章节中描述；
2. 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数字输出模式，如果 P0.3 未用，应外接上拉电阻,并对地下拉一个电容；
3. KF8F31xx 的 8 脚芯片需将 TR2<3>、TR2<1:0>、TR1<7:3>、TR1<1:0>和 TR0<5:2>设置为 0；KF8F31xx 的 14 脚芯片需将 TR2<3>和 TR1<7:3>设置为 0；KF8F31xx 的 16 引脚芯片需将 TR1<7:4>设置为 0；
4. KF8F3110SB(SOIC-8)的外部复位引脚位于 P2.2 口，其他型号位于 P0.3 口。

QFN-20 引脚示意图：



注 1：管脚图中带（1）标注的外设管脚为可选切换管脚，具体操作相应章节中描述。

引脚总述

8 脚型号

IO	引脚序号 8 脚	ADC	PWM	USART	内部 参考	CMP	定时器	中断	上拉	基本
P0.0	7	AN5	PWM1(1)	RX(1) /DT(1)	-	C1IN+	-	IOCL0	Y	SPDAT
P0.1	6	AN6	PWM2(1)	TX(1) /CK(1)	-	C12IN0-	-	IOCL1	Y	SPCLK /ADVRIN
P0.4	3	AN13	-	-	VREOUT	-	T1G	IOCL4	Y	CLKOUT /OSCB
P0.5	2	-	-	-	-	-	T1CK	IOCL5	Y	OSCA
P1.2	5	AN2	-	-	-	C12IN2-	-	INT1	Y	-
P2.2	4	AN9	-	-	-	C12IN3-	-	INT2	Y	RST
-	1	-	-	-	-	-	-	-	-	VDD
-	14	-	-	-	-	-	-	-	-	VSS

14/16/20 脚型号

IO	引脚序号				AD	PWM	SSCI	USART	ECCP	内部参考	AMP	CMP	定时器	中断	上拉	基本
	SOIC-14	SOIC-16	SOIC/SSOP-20	QFN-20												
P0.0	13	15	19	16	AN5	PWM1(1)	SDI(1)/SDA(1)	RX(1)/DT(1)	-	-	-	C1IN+	-	IOCL0	Y	SPDAT
P0.1	12	14	18	15	AN6	PWM2(1)	SCK(1)/SCL(1)	TX(1)/CK(1)	-	-	-	C12IN0-	-	IOCL1	Y	SPCLK/ADVRIN
P0.2	11	13	17	14	AN12	PWM1/PWM21	SDO(1)	-	-	-	-	C1OUT	T0CK	IOCL2/INT0	Y	-
P0.3	4	4	4	1	-	-	-	-	-	-	-	-	-	IOCL3	-	RST
P0.4	3	3	3	20	AN13	-	SS(1)	-	-	VREOUT	-	-	T1G	IOCL4	Y	CLKOUT/OSCB
P0.5	2	2	2	19	-	-	-	-	-	-	-	-	T1CK	IOCL5	Y	OSCA
P1.0	10	12	16	13	AN0	PWM2/PWM22	-	-	-	-	OPOUT	C2IN+	-	-	Y	-
P1.1	9	11	15	12	AN1	PWM23	-	-	CCP(1)	-	OPIN-	C12IN1-	-	-	Y	-
P1.2	8	10	14	11	AN2	PWM24	-	-	P5D	-	OPIN1+	C12IN2-	-	INT1	Y	-
P1.3	-	9	13	10	AN3	-	SDI/SDA	-	-	-	OPIN2+	-	-	-	Y	-
P1.4	-	-	12	9	AN4	-	-	RX/DT	-	-	OPIN3+	-	-	-	Y	-
P1.5	-	-	11	8	-	-	SCK/SCL	-	-	-	-	-	-	-	Y	-
P1.6	-	-	10	7	-	-	-	TX/CK	-	-	-	-	-	-	Y	-
P1.7	-	-	9	6	AN7	-	SDO	-	-	-	-	-	-	-	Y	-
P2.0	5	5	5	2	AN11	-	-	-	P5A/CCP	-	-	-	-	-	Y	-
P2.1	6	6	6	3	AN10	-	-	-	P5B	-	-	C2OUT	-	-	Y	-
P2.2	7	7	7	4	AN9	-	-	-	P5C	-	-	C12IN3-	-	INT2	Y	-
P2.3	-	8	8	5	AN8	-	SS		-	-	-	-	-	-	Y	-
-	1	1	1	18	-	-	-	-	-	-	-	-	-	-	-	VDD
-	14	16	20	17	-	-	-	-	-	-	-	-	-	-	-	VSS

目　录

产品订购信息.. 2

KF8F3110/12/20/22/30/32 芯片使用注意事项.. 3

引脚示意图.. 5

引脚总述.. 7

目　录.. 9

1　系统概述.. 18

1.1　芯片特征.. 19
1.2　系统框图.. 21
1.3　存储器.. 23
1.4　系统时钟.. 24
1.4.1　时钟模块相关寄存器.. 25
1.4.1.1　系统频率控制寄存器 OSCCTL.. 25
1.4.1.2　系统时钟标志寄存器 OSCSTA.. 26
1.4.2　上电延时.. 26
1.4.3　内部高频振荡器.. 26
1.4.4　内部低频振荡器.. 27
1.4.5　外部高频振荡器.. 28
1.4.6　外部低频振荡器.. 28
1.4.7　时钟切换和时钟信号同步.. 28
1.4.8　双速启动模式.. 29
1.4.9　双速启动过程.. 30
1.4.10　外部时钟故障检测.. 30
1.4.11　时钟故障保护处理.. 31
1.5　配置位.. 32
1.6　在线串行编程.. 33

2　I/O 端口介绍.. 34

2.1　I/O 端口的读写.. 34
2.2　I/O 端口的外设功能软件复用.. 35
2.3　P0 口... 36
2.3.1　P0 口相关的寄存器... 37
2.3.1.1　P0 口状态读取寄存器（P0）.. 37
2.3.1.2　P0 口输出锁存寄存器（P0LR）.. 38
2.3.1.3　P0 口方向控制寄存器（TR0）.. 38
2.3.1.4　P0 口上拉功能控制寄存器(PUR0).. 38
2.3.1.5　P0 口电平变化中断控制寄存器(IOCL).. 39
2.3.2　P0 口各引脚内部原理功能框图... 40
2.4　P1 口... 41
2.4.1　P1 口相关的寄存器... 42

2.4.1.1 P1 口状态读取寄存器（P1）......42
2.4.1.2 P1 口输出锁存寄存器（P1LR）......43
2.4.1.3 P1 口方向控制寄存器(TR1)......43
2.4.1.4 P1 口上拉功能控制寄存器(PUR1)......43
2.4.2 P1 口原理功能框图......44
2.5 P2 口......45
2.5.1 P2 口相关的寄存器......46
2.5.1.1 P2 口状态寄存器(P2)......46
2.5.1.2 P2 口输出锁存寄存器（P2LR）......46
2.5.1.3 P2 口方向控制寄存器(TR2)......47
2.5.1.4 P2 口上拉控制寄存器 PUR2......47
2.5.2 P2 口原理功能框图......48
3 存储器......49
3.1 程序存储器(ROM)区......49
3.1.1 JMP、CALL 指令......50
3.2 数据存储器(RAM)区......51
3.2.1 通用寄存器区......51
3.2.2 特殊功能寄存器(SFR)区......52
3.3 FLASH 自写......53
3.3.1 寄存器 NVMDATAH/L......54
3.3.2 寄存器 NVMADDRH/L......54
3.3.3 写 Flash......55
3.3.4 读 Flash......57
3.4 DATA EEPROM......58
3.4.1 寄存器 NVMDATAL......58
3.4.2 寄存器 NVMADDRL......58
3.4.3 寄存器 NVMCTL0/NVMCTL1......58
3.4.4 写 DATA EEPROM......59
3.4.5 读 DATA EEPROM......60
3.5 寄存器组 RN......60
3.6 ID 地址单元......60
4 汇编指令及寻址方式......61
4.1 寻址方式......61
4.1.1 寄存器寻址......61
4.1.2 直接寻址......61
4.1.3 立即数寻址......61
4.1.4 寄存器间接寻址......62
4.1.5 位寻址......62
4.2 汇编指令......62
5 中断......63
5.1 中断相关的寄存器......65
5.1.1 中断控制寄存器(INTCTL)......65

5.1.2 中断使能寄存器 EIE1 67
5.1.3 中断使能寄存器 EIE2 68
5.1.4 中断使能寄存器 EIE3 68
5.1.5 中断标志寄存器 EIF1 69
5.1.6 中断标志寄存器 EIF2 70
5.1.7 中断标志寄存器 EIF3 70
5.1.8 中断优先级控制寄存器 IP0 71
5.1.9 中断优先级控制寄存器 IP1 71
5.1.10 中断优先级控制寄存器 IP2 72
5.1.11 中断优先级控制寄存器 IP3 73
5.1.12 电源控制寄存器 PCTL 73
5.1.13 INT 中断沿选择寄存器 INTEDGCTL 74
5.1.14 中断响应 75
5.2 INT 中断 76
5.2.1 INT0 中断 76
5.2.2 INT1 中断 76
5.2.3 INT2 中断 76
5.3 定时器中断 77
5.4 P0 口中断 77
5.5 PWM 中断 77
5.6 模拟比较器中断 77
5.7 CCP 中断 77
5.8 USART 中断 77
5.9 SSCI 中断 78
5.10 中断现场保护 78

6 定时/计数器 79

6.1 定时器/计数器 T0 79
6.1.1 T0 原理框图 79
6.1.2 T0 相关的寄存器 79
6.1.2.1 OPTR 选择寄存器 80
6.1.3 定时模式 80
6.1.4 计数模式 81
6.1.5 T0 的使用 81
6.2 定时器/计数器 T1 82
6.2.1 T1 原理框图 82
6.2.2 T1 时钟 82
6.2.3 T1 相关的寄存器 83
6.2.3.1 T1 控制寄存器 83
6.2.4 定时模式 84
6.2.5 计数模式 84
6.2.6 T1 重载功能 84
6.2.7 T1 在休眠模式下的运行 85
6.2.8 T1 分配给 PWM1/2 85

6.3 定时器 T2......86
6.3.1 T2 的工作原理......86
6.3.2 T2 相关的寄存器......87
6.3.2.1 T2 控制寄存器 T2CTL0......87
6.3.2.2 T2 控制寄存器 T2CTL1......88
6.3.2.3 T2CCR0H 和 T2CCR0L 寄存器......88
6.3.3 T2 中断......89
6.3.4 T2 在休眠模式......89
6.3.5 T2 分配给 PWM5......89
6.4 定时器 T3......90
6.4.1 T3 原理框图......90
6.4.2 T3 相关寄存器......90
6.4.2.1 T3 控制寄存器 T3CTL......90
6.4.2.2 T3 控制寄存器 T3CTL1......91
6.4.3 T3 中断......91
6.4.4 T3 的使用......91

7 模数(A/D)转换模块......92

7.1 与 AD 相关的寄存器......92
7.1.1 AD 控制寄存器 0(ADCCTL0)......93
7.1.2 AD 控制寄存器 1(ADCCTL1)......94
7.1.3 模拟/数字口选择寄存器(ANSEL/H)......94
7.1.4 A/D 扫描中断控制寄存器（ADCSICM）......95
7.2 通道的选择......95
7.3 模拟输入口的配置......95
7.4 多通道扫描模式......95
7.4.1 A/D 多通道扫描控制寄存器......96
7.5 A/D 转换参考电压的选择......96
7.5.1 VREOUT 参考电压......96
7.5.2 参考电压寄存器（VRECTL）......97
7.6 转换时钟的选择......97
7.7 输出格式......97
7.8 A/D 转换的启动和完成......98
7.9 复位的影响......98
7.10 使用 A/D 转换器的设置......98

8 PWM 模块......99

8.1 工作原理......99
8.1.1 16 位 PWM 模式......99
8.1.2 8 位 PWM 模式......100
8.2 PWM 相关的寄存器......101
8.2.1 PWM1/2 控制寄存器......101
8.2.2 PWM 周期......102
8.2.3 PWM 占空比......102

8.3 PWM 分辨率 103
8.4 PWM 中断 103
8.5 PWM 输出引脚 103
8.6 休眠模式下的操作 104
8.7 复位的影响 104
8.8 PWM 使用方法 104

9 CCP(捕捉/比较/PWM5)模块 105

9.1 CCP 相关寄存器 105
9.1.1 捕捉/比较控制寄存器（CCPCTL） 106
9.2 捕捉模式 107
9.3 比较模式 108
9.4 PWM5 模式 109
9.4.1 PWM5 相关控制寄存器 110
9.4.1.1 PWM5 控制寄存器 0（PWM5CTL0） 110
9.4.1.2 PWM5 极性控制寄存器（PWM5PC） 111
9.4.1.3 PWM5 强制控制寄存器(PWM5FC) 111
9.4.1.4 PWM5 输出控制寄存器（PWM5OC） 111
9.4.1.5 PWM5 控制寄存器 1(PWM5CTL1) 112
9.4.1.6 PWM5 自动关闭控制寄存器（P5ASCTL） 112
9.4.1.7 PWM5 脉冲转向控制寄存器（PSTRCTL） 113
9.4.1.8 PWM5 控制寄存器 2（PWM5CTL2） 113
9.4.2 PWM5 的周期、占空比及分辨率 114
9.4.2.1 PWM5 周期 114
9.4.2.2 PWM5 占空比 114
9.4.2.3 PWM5 分辨率 115
9.4.2.4 PWM5 中断 115
9.4.3 边沿对齐 PWM 信号 115
9.4.4 中心对齐 PWM 信号 116
9.4.5 PWM 信号产生和中断 117
9.4.6 单输出模式 117
9.4.7 半桥输出模式 119
9.4.7.1 死区延时 120
9.4.8 全桥输出模式 122
9.4.9 脉冲转向控制 125
9.4.10 输出控制 125
9.4.11 强制输出模式 125
9.4.12 PWM 输出极性模式 125
9.4.13 PWM 更新锁定 126
9.4.14 PWM 复位控制 126
9.4.15 自动关断和自动重启模式 126
9.4.15.1 自动关断模式 126
9.4.15.2 自动重启模式 127

10 SSCI 模块 129

10.1 概述 129
10.2 SSCI 模块功能引脚 129
10.3 SSCI 相关寄存器 129
10.3.1 SSCI 控制寄存器 0（SSCICTL0） 130
10.3.2 SSCI 控制寄存器 1（SSCICTL1） 131
10.3.3 SSCI 状态寄存器（SSCISTA） 132
10.3.4 SSCI 屏蔽寄存器（SSCIMSK） 133
10.3.5 SSCI I2C 地址寄存器（SSCIADD） 134
10.4 I2C 模式 135
10.4.1 工作原理 135
10.4.2 I2C 从动模式 136
10.4.2.1 寻址 136
10.4.2.2 接收 137
10.4.2.3 发送 138
10.4.2.4 广播呼叫地址支持 140
10.4.3 I2C 主控模式 141
10.4.3.1 主控模式支持 141
10.4.3.2 I2C 主模式操作 142
10.4.3.3 波特率发生器 143
10.4.3.4 I2C 主控模式启动条件时序 144
10.4.3.5 I2C 主控模式重复启动条件时序 144
10.4.3.6 I2C 主控模式发送 145
10.4.3.7 I2C 主控模式接收 147
10.4.3.8 应答序列时序 148
10.4.3.9 停止条件序列 149
10.4.3.10 时钟仲裁 150
10.4.4 多主控器模式 151
10.4.4.1 多主机通信，总线冲突与总线仲裁 151
10.4.4.2 启动条件期间的总线冲突 152
10.4.4.3 重复启动条件期间的总线冲突 154
10.4.4.4 停止条件期间的总线冲突 156
10.4.4.5 SSCI 屏蔽寄存器 157
10.5 SPI 模式 158
10.5.1 工作原理 159
10.5.2 使能 SPI/IO 与外部链接 159
10.5.3 典型连接 160
10.5.4 主控模式 160
10.5.5 从动模式 161
10.5.5.1 从动模式 161
10.5.5.2 从动选择同步 162
10.5.6 休眠模式和复位 163
10.5.7 SPI 四种工作模式设置 164
10.5.7.1 主控发送工作流程 164
10.5.7.2 从动接收工作流程 164

10.5.7.3 主控接收工作流程 165
10.5.7.4 从动发送工作流程 165

11 通用全双工/ 半双工收发器（USART） 166

11.1 系统概述 166
11.2 USART 功能引脚 166
11.3 相关寄存器 166
11.4 原理框图 167
11.5 波特率发生器 168
11.5.1 相关寄存器 168
11.5.1.1 USART 波特率控制寄存器 BRCTL 168
11.5.2 波特率的选择 169
11.5.3 自动波特率检测 171
11.5.4 接收间隔字符时自动唤醒 172
11.5.5 兼容 LIN 总线的间隔符时序 174
11.5.5.1 间隔和同步发送序列 174
11.5.5.2 接收间隔字符 175
11.6 USART 全双工模式 176
11.6.1 USART 全双工发送操作 177
11.6.1.1 发送状态和控制寄存器 TSCTL 177
11.6.1.2 发送数据 178
11.6.1.3 全双工发送的设置 179
11.6.2 USART 全双工接收操作 180
11.6.2.1 接收状态和控制寄存器 RSCTL 181
11.6.2.2 接收数据 182
11.6.2.3 接收错误 182
11.6.2.4 地址检测 183
11.6.2.5 全双工接收的设置 183
11.6.3 RS-485 发送/接收 184
11.6.3.1 RS-485 9 位地址检测模式设置 184
11.6.4 全双工操作时钟的精确性 185
11.7 USART 半双工模式 185
11.7.1 USART 半双工主控模式 185
11.7.1.1 半双工主控发送 186
11.7.1.2 半双工主控接收 187
11.7.2 USART 半双工从动模式 189
11.7.2.1 USART 半双工从动发送 190
11.7.2.2 USART 半双工从动接收 190
11.7.3 USART 半双工 RS-485 模式 191

12 运算放大器模块 192

12.1 与运放有关的寄存器 192
12.1.1 运放控制寄存器（AMPCTL） 192
12.1.2 运放校验寄存器（AMPDT） 193

12.2 运放使用方法......193
12.2.1 校验方法......193
12.2.2 使用方法......194
12.3 复位的影响......194
13 模拟比较器模块......195
13.1 模拟比较器原理......195
13.2 相关寄存器......195
13.3 CMP1 相关寄存器......196
13.3.1 比较器 1 控制寄存器 C1CTL......196
13.3.2 比较器 1 校准寄存器 C1CAL......197
13.3.3 比较器 1 滤波控制寄存器 C1FILTCTL......197
13.3.4 比较器 1 滤波采样时钟分频寄存器 C1FILTPRE......198
13.4 CMP2 相关寄存器......198
13.4.1 比较器 2 控制寄存器 C2CTL......198
13.4.2 比较器 2 校准寄存器 C2CAL......199
13.4.3 比较器 2 滤波控制寄存器 C2FILTCTL......199
13.4.4 比较器 2 滤波采样时钟分频寄存器 C2FILTPRE......200
13.4.5 比较器输出寄存器 COUT......200
13.5 极性选择......201
13.6 范围控制功能......201
13.7 滤波功能......202
13.8 比较器中断......202
13.9 模拟比较器的校准步骤......202
13.10 比较器使用......203
13.11 复位的影响......203
14 硬件乘法器模块......204
14.1 概述......204
14.2 硬件乘法器相关寄存器......205
14.2.1 乘法运算控制寄存器（MULCTL）......205
14.3 硬件乘法器的使用......205
15 硬件除法器模块......206
15.1 概述......206
15.2 硬件除法器相关寄存器......206
15.2.1 除法控制寄存器（DIVCTL）......207
15.2.2 数据寄存器......207
15.3 硬件除法器操作......207
16 复位......208
16.1 电源控制寄存器(PCTL)......209
16.2 上电复位(POR)......210
16.3 WDT 复位......210
16.4 RST 复位......210

16.5 欠压检测复位(LVR)......211
16.6 上电延时定时器......211
16.7 不同复位条件下对寄存器的影响......212

17 休眠模式......215

18 看门狗定时器 WDT......216

18.1 看门狗相关寄存器......216
18.1.1 WDT 预分频选择寄存器 WDTPS......216
18.2 看门狗的开启关闭方式......216
18.3 看门狗的清狗方式......217
18.4 看门狗 WDT 的周期......217

19 电气规范......218

19.1 极限参数特性......218
19.2 HFINT 的频率精度与器件 VDD 和温度之间的关系......219
19.3 静态电流特性......219
19.4 外设电流特性......221
19.5 I/O 端口特性......222
19.6 芯片供电电压特性......222
19.7 A/D 转换器（ADC）特性......223
19.8 内部高频振荡器模块特性......223
19.9 内部低频 LPRC 模块特性......224
19.10 外部低频 LPXTAL 模块特性......224
19.11 模拟比较器模块特性......224
19.12 运算放大器模块特性......225
19.13 内部参考电压模块特性......225
19.14 FLASH 自写和 DATA EEPROM 特性......226

20 直流特性图表......227

21 封装信息......235

附录 1 特殊功能寄存器（SFR）功能汇总......239

附录 2 汇编指令集......242

附录 3 寄存器全称表......244

附录 4 全双工异步模式的典型波特率和误差值......247

附录 5 BUGLIST......251

产品标识体系......252

版本信息......253

ROHS 认证......254

声明及销售网络......255

1 系统概述

KF8F3110/12/20/22/30/32 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

1 个 16 位定时器/计数器 T1

2 个 16 位定时器 T2/T3

1 个 12 位 10/12/14 通道 ADC 模块

2 个 8 位 PWM 模块

1 个 CCP(捕捉/比较/PWM5)模块

2 个模拟比较器模块

1 个运算放大器模块（KF8F3110/20/30 没有运放模块）

1 个 USART 模块

1 个 SSCI(I2C/SPI)模块

一个 2V/3V/4V 可选的参考电压

硬件看门狗（带软件使能）

低电压检测及低电压复位模块等

芯片内集成了 (512+16)×8 位的用户数据存储器 RAM、8K 字节的程序存储器和 128×8 位的 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）
系统时钟源选择内部高频时钟时的工作频率：
KF8F3110/12：15.625kHz～4MHz
KF8F3120/22：31.25kHz～8MHz
KF8F3130/32：62.5kHz～16MHz

- **存储器**

8K 字节的 FLASH 程序存储器
528×8 位的数据存储器
128×8 位的 DATA EEPROM
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
低电压检测及低电压复位
硬件看门狗
内部高频时钟精度：
KF8F3110/12：4MHz±1%（常温）
KF8F3120/22：8MHz±1%（常温）
KF8F3130/32：16MHz±1%（常温）
内部可校正低频 32kHz 时钟
一个 2V/3V/4V 可选的参考电压
支持在线串行编程
低功耗休眠模式

- **I/O 口配置**

输入输出口：除 P0.3 只能作为输入口外其它端口均为双向输入输出口
内置上拉功能：P0/P1/P2 口带有弱上拉功能(P0.3 除外)
电平变化中断：P0 口均有电平变化中断功能
IO 口数字输入类型：P0/P1/P2 口为 SMIT 型

- **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：带门控和预分频器的 16 位定时器/计数器
定时器 2：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：时钟源为系统时钟源或内部高频振荡器源的 16 位定时器

- **其它外设**

 1 个 12 位 10/12/14 通道 ADC 模块
 2 个 8 位脉冲宽度调制 PWM 模块
 1 个 SSCI(SPI/I2C)模块
 1 个 USART 模块
 1 个 CCP(捕捉/比较/增强型 PWM5)模块
 2 个模拟比较器模块
 1 个运算放大器模块（KF8F3110/20/30 没有运放模块）

- **工作条件**

 工作电压：　　2.1V～5.5V
 工作温度范围：-40℃～85℃
 　　　　　　　-40℃～125℃

1.2 系统框图

KF8F3110/20/30 的系统框图如图 1.1a 所示：

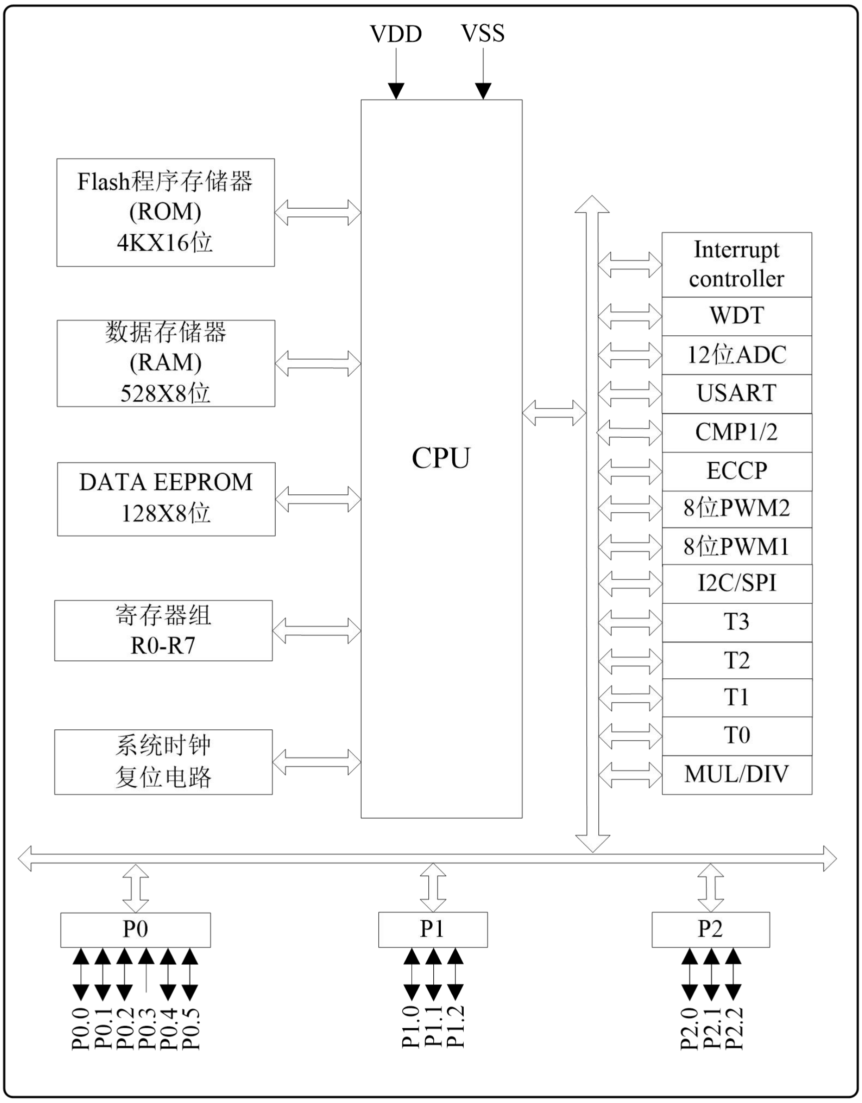


图 1.1a　KF8F3110/20/30 系统框图

KF8F3112/22/32 的系统框图如图 1.1b 所示：

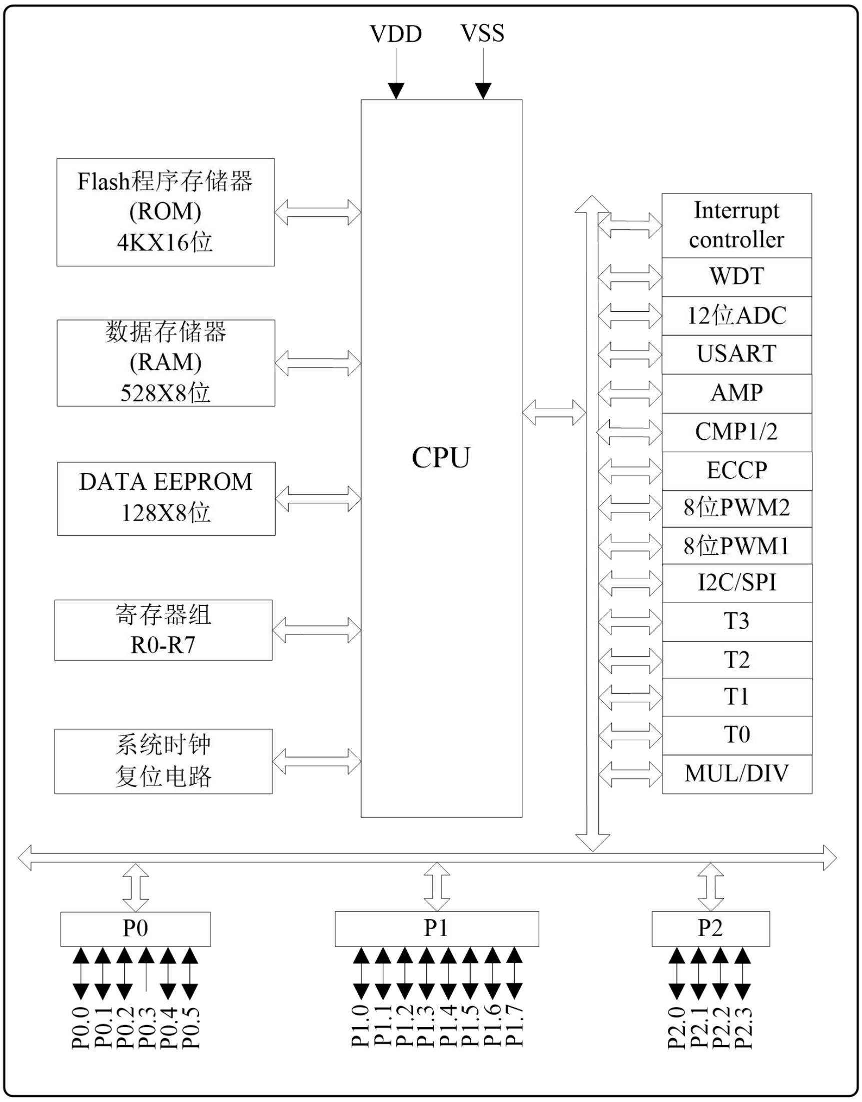


图 1.1b　KF8F3112/22/32 系统框图

1.3 存储器

KF8F3110/12/20/22/30/32 单片机的存储器包含:程序存储器(ROM)、数据存储器(RAM)和 DATA EEPROM。

KF8F3110/12/20/22/30/32 的程序存储器空间为 8K 字节，寻址范围为 0000H～0FFFH，可擦写次数为 10 万次。数据存储器分为特殊寄存器区(SFR)和通用存储器区，其中通用存储器区包括通用存储器区 0 至通用存储器区 3，每个区均有 128×8 位的存储单元，各区的地址请查阅第 3 章。

DATA EEPROM 的地址为独立寻址，地址为 00H-7FH。有关以上各种存储器的具体介绍请参考第 3 章。

1.4 系统时钟

系统时钟是由系统时钟源分频而来。本芯片中一个机器周期等于四个系统时钟周期，如图 1.2 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。

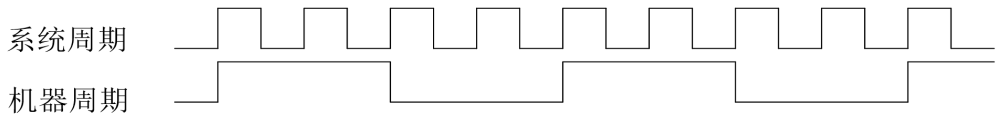


图 1.2 机器周期

KF8F3110/12/20/22/30/32 系列单片机提供 4 个可选时钟源:

- 内部高频时钟 INTHF:以内部高频振荡器为时钟源;
- 内部低频时钟 INTLF:以内部低频振荡器为时钟源;
- 外部高频时钟 EXTHF:标准晶振、陶瓷谐振器或外接 16MHz～125kHz 的时钟源工作;
- 外部低频时钟 EXTLF:外接 32.768kHz 的钟表晶振。

KF8F3110/12/20/22/30/32 系列单片机的系统时钟源可以配置为 EXTHF、EXTLF、INTHF 或者 INTLF。外部高频时钟和外部低频时钟共用外部时钟引脚，不能同时使用两个外部时钟源。

KF8F3110/12/20/22/30/32 系列单片机时钟模块框图

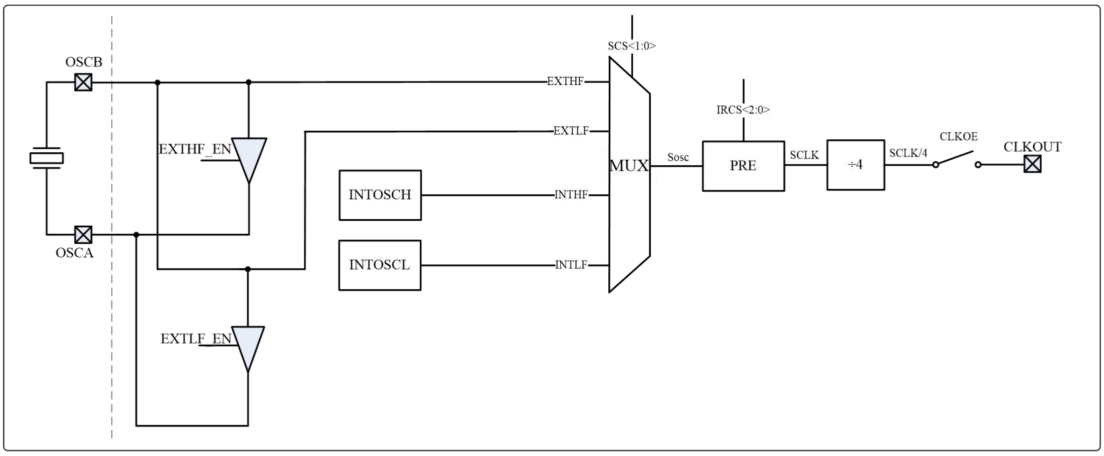


图 1.3 时钟模块原理框图

表 1-1 与时钟有关的名词表述

名称	定义即描述
Sosc	振荡器时钟源
SCLK	定义为系统时钟
SCLK/4	定义为机器时钟
T_{sys} 或 Tsys	定义为系统时钟周期
T_{mc} 或 Tmc	定义为机器周期
INTOSCH(INTHF)	内部高频振荡器（时钟源）
T_{INTHF}	内部高频振荡器周期
INTOSCL(INTLF)	内部低频振荡器（时钟源）
EXTOSCH(EXTHF)	外部高频振荡器（时钟源）
EXTOSCL(EXTLF)	外部低频振荡器（时钟源）
OSC	定义为 INTHF、INTLF、EXTHF 和 EXTLF 的集合

1.4.1 时钟模块相关寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
2FH	OSCCTL	CLKOE	IRCS2	IRCS1	IRCS0	SCS1	SCS0	IESO	FSCM
28H	OSCSTA	-	OSTS	HTS	LTS	-	-	SCF1	SCF0

1.4.1.1 系统频率控制寄存器 OSCCTL

寄存器1.1: OSCCTL系统频率控制寄存器(地址:2FH)

复位值 0010 0000	bit7							bit0
	CLKOE	IRCS2	IRCS1	IRCS0	SCS1	SCS0	IESO	FSCM
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CLKOE: 系统时钟输出使能位
1=使能系统时钟输出
0=禁止系统时钟输出

IRCS<2:0>: 时钟频率选择位
111=1:1
110=1:2
101=1:4
100=1:8
011=1:16
010=1:32（默认）
001=1:64
000=1:256

SCS<1:0>: 系统时钟选择位
11=选择外部高频时钟
10=选择外部低频时钟
01=选择内部低频时钟
00=选择内部高频时钟

IESO: 双速模式使能位
1=启动双速功能
0=禁止双速功能

FSCM: 外部时钟故障检测使能位
1=使能故障检测功能
0=禁止故障检测功能

图注：R=可读 W=可写 -=未用 U=未实现位

注： 1.在使用外部时钟且经常进入休眠模式的应用中，双速模式时唤醒所花费的时间中不包含振荡器的起振时间。
2.在使能故障检测功能后，外部时钟发生故障后会自动切换到内部高频时钟，详见1.4.11。

1.4.1.2 系统时钟标志寄存器 OSCSTA

寄存器1.2: OSCSTA系统时钟标志寄存器(地址:28H)

复位值 0110 --00	bit7							bit0
	-	OSTS	HTS	LTS	-	-	SCF1	SCF0
	R	R	R	R	U	U	R	R

OSTS: 内外时钟标志位
- 1=系统时钟为内部时钟
- 0=系统时钟为外部时钟

HTS: 内部高频时钟稳定标志位
- 1=内部高频时钟稳定
- 0=内部高频时钟未稳定

LTS: 内部低频时钟稳定标志位
- 1=内部低频时钟稳定
- 0=内部低频时钟未稳定

SCF<1:0>: 系统时钟标志位
- 11=当前系统时钟为外部高频时钟
- 10=当前系统时钟为外部低频时钟
- 01=当前系统时钟为内部低频时钟
- 00=当前系统时钟为内部高频时钟

图注：R=可读 W=可写 -=未用 U=未实现位

1.4.2 上电延时

KF8F3110/12/20/22/30/32 单片机的上电延时可以通过配置位 $\overline{\text{PWRT}}$ 设置，上电延时计数时钟 PWRTCLK 为内部低频时钟。

当 $\overline{\text{PWRT}}$=1 时，上电延时关闭；

当 $\overline{\text{PWRT}}$=0 时，上电延时打开，延时时间如下: $Tpwrt = 2^{10} \times Tpwrtclk$ 。

注：$Tpwrtclk$=上电延时计数时钟周期。

1.4.3 内部高频振荡器

KF8F3110/12/20/22/30/32 单片机的内部高频时钟由系统内部高频振荡器提供。KF8F3110/12 的内部高频时钟频率为 4MHz；KF8F3120/22 的内部高频时钟频率为 8MHz；KF8F3130/32 的内部高频时钟频率为 16MHz。精度为±1%（常温）。

OSCCAL0、OSCCAL1、OSCCAL2 和 OSCCAL3 为 INTOSCH 的校准寄存器，用来存放 INTHF 校准值；VRECAL1、VRECAL2 和 VRECAL3 为内核等内部参考电压的校准值。用户在编程时，需要在程序初始化部分将存放在程序空间 0FFFH、0FFEH、0FFDH、0FFCH、0FFBH、0FFAH 和 0FF9H 的校准值读出来分别存放到 OSCCAL0、OSCCAL1、OSCCAL2、OSCCAL3、VRECAL1、VRECAL2 和 VRECAL3 中，否则会导致系统时钟和参考电压不准。

程序示例请参考例 1.1。

例1.1：内部高频振荡器及内部参考电压校准
```
        CALL   0X0FFF              ;读取内部高频振荡器校准值0
        MOV    OSCCAL0, R0
        NOPZ
        NOPZ
        CALL   0X0FFE              ;读取内部高频振荡器校准值1
        MOV    OSCCAL1, R0
        NOPZ
        NOPZ
        CALL   0X0FFD              ;读取内部高频振荡器校准值2
        MOV    OSCCAL2, R0
        NOPZ
        NOPZ
        CALL   0X0FFC              ;读取内部高频振荡器校准值3
        MOV    OSCCAL3, R0
        NOPZ
        NOPZ
        CALL   0X0FFB              ;读取内部参考电压校准值1
        MOV    VRECAL1, R0
        NOPZ
        NOPZ
        MOVB #0X01                 ;切换到存储器1区
        CALL   0X0FFA              ;读取内部参考电压校准值2
        MOV    VRECAL2, R0
        NOPZ
        NOPZ
        CALL   0X0FF9              ;读取内部参考电压校准值3
        MOV    VRECAL3, R0
        NOPZ
        NOPZ
        MOVB #0X00                 ;切换回存储器0区
```

1.4.4 内部低频振荡器

系统内部低频振荡器，其振荡器频率为 32kHz，它不仅可以为 SCLK 提供时钟源，还可以单独作为低频时钟信号供外设模块使用，同时可以作为内部上电延时定时器和看门狗 WDT 定时器的时钟。

RC32KCAL（165H）为内部低频振荡器的校准寄存器，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 FF8H 的晶振校准值存放到 RC32KCAL 中。程序示例请参考例 1.2。

例1.2　　内部低频振荡器校准
```
        CALL     0X0FF8            ;读取内部低频振荡器校准值
        MOVB     #0X01             ;切换到BANK1区
        MOV      RC32KCAL , R0     ;校准值送RC32KCAL
                                   ;切换回原BANK区
```

注：对内部低频精度要求高的使用场合，用户需进行实时校准，以保证精度。

1.4.5 外部高频振荡器

如图 1.4 所示，引脚 OSCA(P0.5)和引脚 OSCB(P0.4)可以接外部标准晶体、陶瓷谐振器或外接 16MHz～125kHz 的时钟为外部高频时钟源。外部高频时钟 EXTHF 可作为系统时钟源。

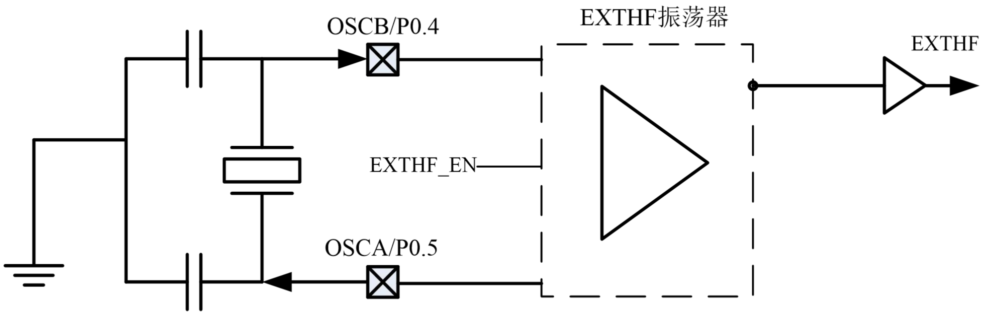


图 1.4 外部高频振荡器原理图

1.4.6 外部低频振荡器

如图 1.5 所示，引脚 OSCA(P0.5)和引脚 OSCB(P0.4)外接 32.768kHz 石英晶振。EXTLF 可以作为系统时钟 SCLK 和其他外设时钟源。

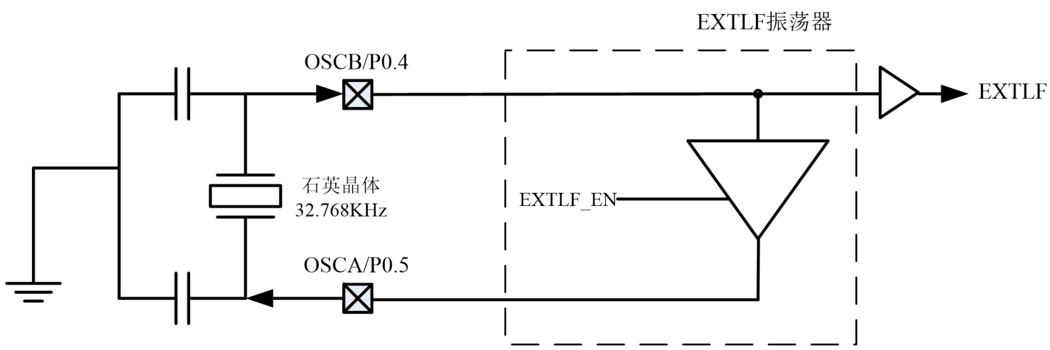


图 1.5 外部低频振荡器原理图

1.4.7 时钟切换和时钟信号同步

通过设置 OSCCTL 寄存器的 SCS<1:0>位可以选择不同的时钟源作为系统时钟。SCS<1:0>位复位为 00，即可选择内部高频时钟源作为单片机的系统时钟。

当系统时钟由外部时钟源切换至内部时钟源时，系统时钟将在 SCS<1:0>配置后立即对时钟进行切换。

当配置 SCS<1:0>位将系统时钟切换至外部时钟源时（包括 EXTHF 和 EXTLF），振荡器起振定时器 OST 将启动，并以 SCS<1:0>位配置的外部时钟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 SCS<1:0>位配置的外部时钟。

图 1.6　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。当选择一个新的时钟源，会发生以下过程。

（1）　写 SCS<1:0>位改变时钟源，如切换至外部时钟源则须经过 OST 计数器 1024 计数；

（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，保持 2 个当前时钟周期，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。

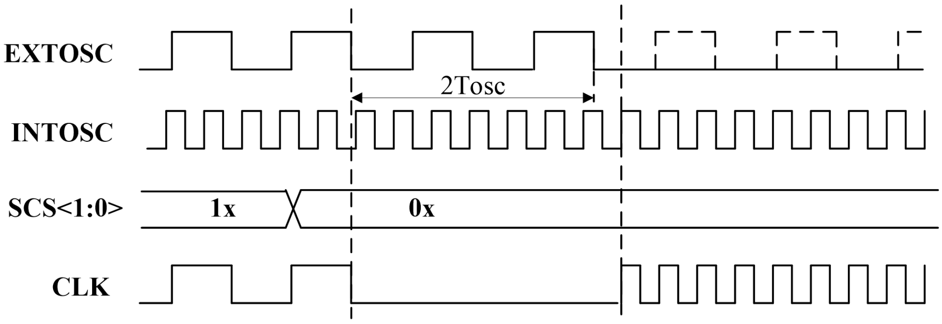


图 1.7　切换至内部时钟源时时钟信号同步时序图

注：图中的 EXTOSC 根据实际功能确定是否使能

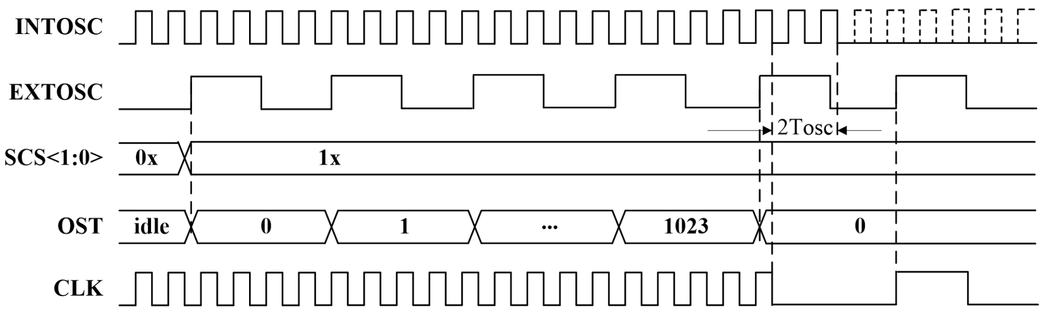


图 1.8　切换至外部时钟源时时钟信号同步时序图

注：图中的 INTOSC 根据实际功能确定是否使能

1.4.8　双速启动模式

当系统时钟选择外部时钟进入休眠后，唤醒时，外部时钟需要经过 OST 计数器完成 1024 次计数后才会恢复系统时钟。

双速启动模式通过寄存器 OSCCTL 的 IESO 位设置，当使能双速模式时，MCU 会在唤醒后，外部时钟进行 OST 计数期间，通过内部高频时钟作为系统时钟运行，当外部时钟完成 OST 计数后，MCU 会自动将系统时钟从内部高频时钟切换至外部时钟。

1.4.9 双速启动过程

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

1.4.10 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 OSCCTL 寄存器的 FSCM 位置 1 使能 FSCM，适用于外部振荡器模式。

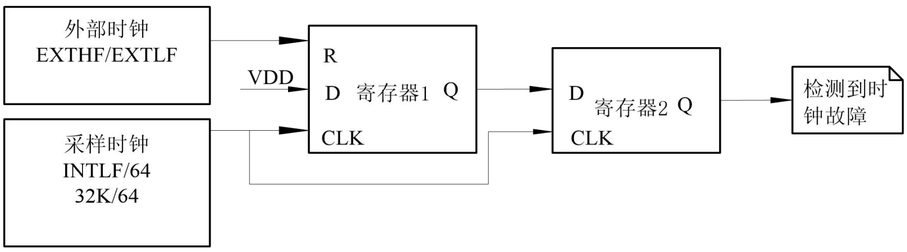


图 1.9 时钟故障检测原理图

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。

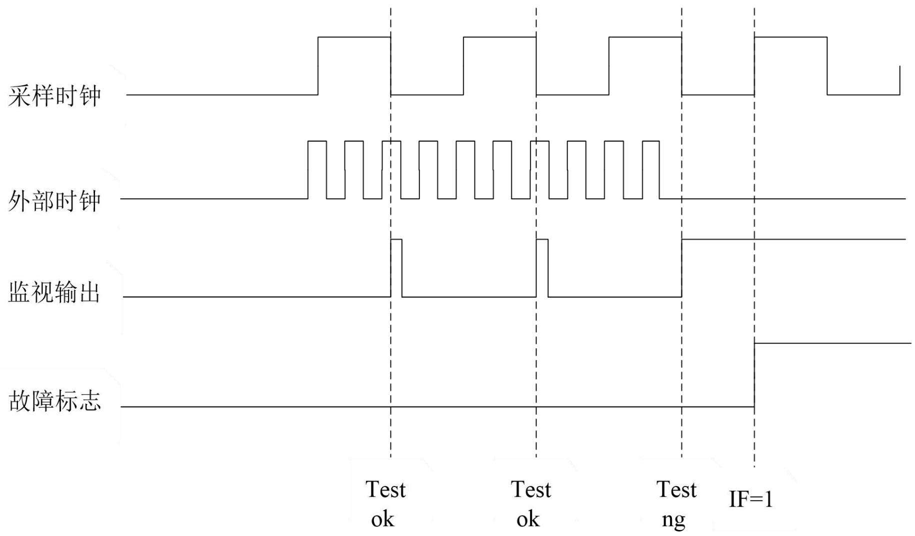


图 1.10 时钟故障检测时序

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。

1.4.11 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 OSCCTL 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。

1.5 配置位

用户在下载程序时，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

R	R	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	R
-	-	PWMPIN	HPOL	LPOL	$\overline{\text{DEBUG}}$	SWRTEN	保留	$\overline{\text{CODEP}}$	LVREN	RSTEN	$\overline{\text{PWRT}}$	WDTEN	$\overline{\text{DATAP}}$	-	-
bit15							bit8								bit0

注：R=编程时可读　P=编程时可擦写

PWMPIN: PWM5 复位控制

1 = PWM5 模块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: PWM5高边驱动器极性选择位

1 = P5A、P5C引脚上的输出信号极性为高电平有效

0 = P5A、P5C引脚上的输出信号极性为低电平有效

LPOL: PWM5低边驱动器极性选择位

1 = P5B、P5D引脚上的输出信号极性为高电平有效

0 = P5B、P5D引脚上的输出信号极性为低电平有效

注：复位默认高阻态

$\overline{\text{DEBUG}}$: 在线调试使能位

1 = 禁止在线调试

0 = 使能在线调试

SWRTEN: FLASH 自写使能位

1 = 使能写保护，此时 FLASH 不能写操作(可读)

0 = 禁止写保护，此时 FLASH 可写(可读)

$\overline{\text{CODEF}}$: 代码加密使能位

1 = 禁止程序存储器代码加密

0 = 使能程序存储器代码加密

LVREN: 欠压检测功能使能位

1 = 使能欠压检测功能

0 = 禁止欠压检测功能

RSTEN: P0.3/$\overline{\text{RST}}$ 引脚功能选择

1 = P0.3/$\overline{\text{RST}}$引脚配置为外部复位输入

0 = P0.3/$\overline{\text{RST}}$引脚功能为数字输入口

$\overline{\text{PWRT}}$: 上电延时使能位

1 = 禁止上电延时

0 = 使能上电延时

WDTEN: 看门狗定时器(WDT)使能位

1 = 使能 WDT

0 = 禁止 WDT

$\overline{\text{DATAP}}$: DATA EEPROM 加密使能位

1 = 禁止 DATA EEPROM 加密

0 = 使能 DATA EEPROM 加密

1.6 在线串行编程

如图 1.11、1.12 所示，在最终应用电路中可对 KF8F3110/12/20/22/30/32 单片机进行在线串行编程、调试。实现串行编程仅需要五根线包括: 时钟线(SPCLK)、数据线(SPDAT)、电源线(VDD)、地线(VSS)和 $\overline{\mathrm{RST}}$ 线。

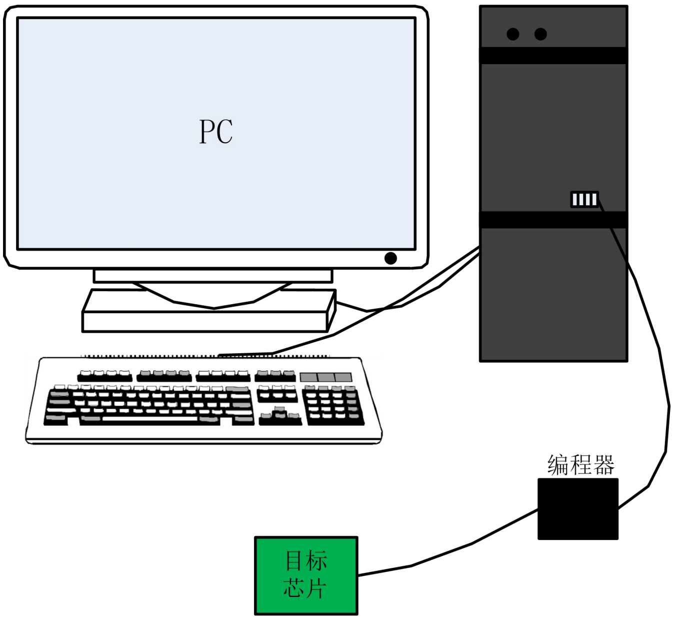


图 1.11 在线编程调试系统示意图

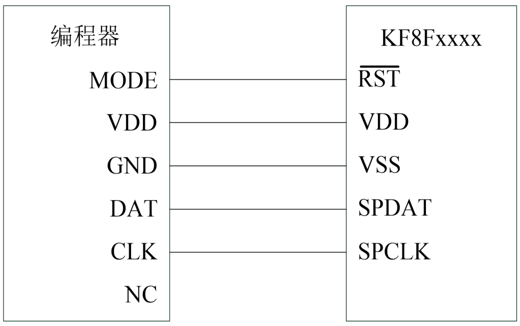


图 1.12 在线串行编程连接图

2 I/O 端口介绍

KF8F3110 提供 8 脚 SOIC-8 和 14 脚 SOIC-14 封装形式；KF8F3120/30 提供 14 脚 SOIC-14 封装形式；KF8F3112/22/32 共有 3 种封装形式，最多有 20 个引脚。除电源正极、电源负极外，其余管脚均为可用 I/O 端口。具体封装对应可用 IO 可以参见前面的“引脚总述”。

注：

1. 管脚图中带（1）标注的外设管脚为可选切换管脚，具体操作在相应章节中描述；
2. 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数字输出模式，如果 P0.3 未用，应外接上拉电阻,并对地下拉一个电容；
3. KF8F31xx 的 14 脚芯片需将 TR2<3>和 TR1<7:3>设置为 0；KF8F31xx 的 16 引脚芯片需将 TR1<7:4>设置为 0；
4. KF8F3110SB 的外部复位引脚位于 P2.2 口，其他型号位于 P0.3 口。

2.1 I/O 端口的读写

读 P0/P1/P2 口时实际为读端口电平，即读 Px(x=0/1/2)，改变 IO 端口的输出状态通过写 PxLR(x=0/1/2)寄存器实现。其原理框图如图 2.1 所示：

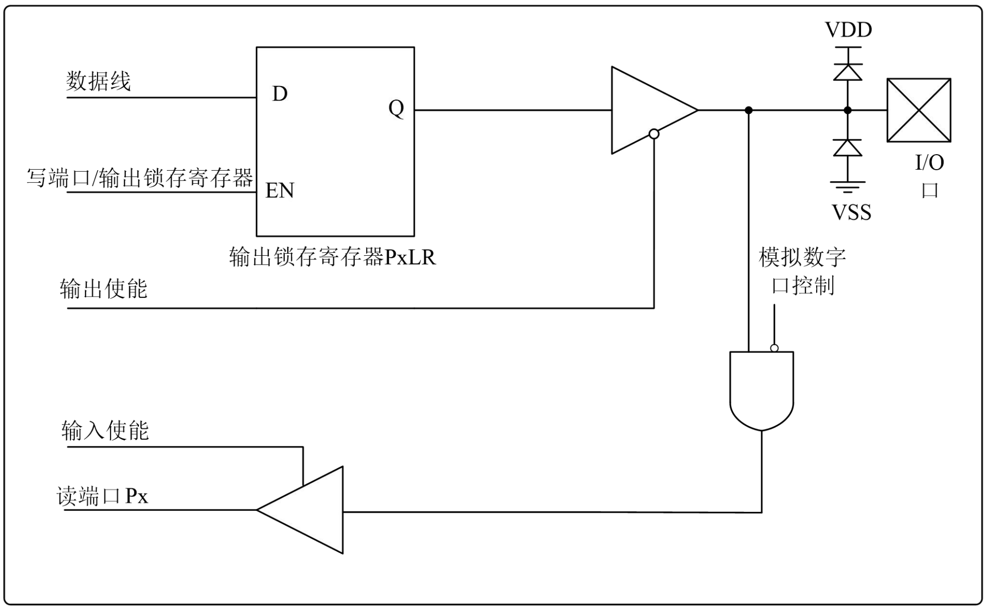


图 2.1 I/O 口读写原理图

注：

1. I/O 口作为 AD 通道时，须将对应的 ANSx 位置 1 将该 I/O 口配置为模拟口。
2. I/O 口无模拟功能时，“模拟数字口控制”为 0。

2.2 I/O 端口的外设功能软件复用

SSCI、USART 和 PWM 模块的功能引脚可以通过 PINSET 寄存器的相关控制位来切换。CCP 模块的功能引脚切换通过 CCPCTL 寄存器的 CCPPIN 位控制。

寄存器**2.1**: PINSET:外设功能引脚复用切换寄存器(地址：156H)

复位值 0000 0000	bit7							bit0
	SSCIPIN	SDOPIN	SSPIN	UARTPIN	PWMPIN	-	-	-
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SSCIPIN：SSCI 模块功能引脚（SDA/SDI/SCK/SCL）切换位

0=SDA/SDI 切换至 P1.3，SCK/SCL 切换至 P1.5

1=SDA/SDI 切换至 P0.0，SCK/SCL 切换至 P0.1

注：14 或 16 引脚芯片使用 SSCI 功能时 SSCIPIN 必须配置为 1。

SDOPIN：功能引脚 SDO 切换位

0=SDO 切换至 P1.7

1=SDO 切换至 P0.2

注：14 或 16 引脚使用 SSCI 功能时 SDOPIN 必须配置为 1。

SSPIN: 功能引脚 $\overline{SS}$ 切换位

0= $\overline{SS}$ 切换至 P2.3

1= $\overline{SS}$ 切换至 P0.4

注：14 引脚芯片使用 SSCI 功能时 SSPIN 必须配置为 1。

UARTPIN:USART 模块功能引脚切换位

0=RX/DT 切换至 P1.4，TX/CK 切换至 P1.6

1=RX/DT 切换至 P0.0，TX/CK 切换至 P0.1

注：14 或 16 引脚芯片使用 USART 功能时 UARTPIN 必须配置为 1。

PWMPIN: PWM1/2 模块输出引脚切换位

0 = PWM1 输出引脚切换至 P0.2，PWM2 输出引脚切换至 P1.0

1 = PWM1 输出引脚切换至 P0.0，PWM2 输出引脚切换至 P0.1

2.3 P0 口

在线编程时 P0 口的 SPCLK、SPDAT 作为编程脚使用。P0 口特点：

- 除 P0.3 口只能作为输入口外，其他 P0 口均可作为普通 I/O；
- 除 P0.3 口外，其他 P0 口均带有上拉功能；
- P0 口所有引脚都有电平变化中断功能。

KF8F3110/12/20/22/30/32 的 P0 口各引脚功能表 2-1 所示。

表 2-1 KF8F3110/12/20/22/30/32 的 P0 口各引脚功能

引脚名	I/O	引脚功能	引脚说明
P0.0/AN5 /SPDAT/RX(1) /DT(1)/SDI(1) /SDA(1) /PWM1(1) /C1IN+	I/O	P0.0	带上拉和电平变化中断功能的双向输入输出端口
		AN5	ADC 输入通道 5
		SPDAT	编程数据输入/输出
		RX(1)	可配置 USART 全双工异步数据接收端
		DT(1)	可配置 USART 半双工同步数据端
		SDI(1)	可配置 SPI 数据输入脚
		SDA(1)	可配置 I2C 数据输入/输出脚
		PWM1(1)	可配置 PWM1 输出通道
		C1IN+	比较器 CMP1 正端输入
P0.1/AN6 /SPCLK/TX(1) /CK(1)/SCK(1) /SCL(1) /PWM2(1) /C12IN0- /ADVIN	I/O	P0.1	带上拉和电平变化中断功能的双向输入输出端口
		AN6	ADC 输入通道 6
		SPCLK	编程时钟输入
		TX(1)	可配置 USART 全双工异步数据发送端
		CK(1)	可配置 USART 半双工同步时钟端
		SCK(1)	可配置 SPI 时钟脚
		SCL(1)	可配置 I2C 时钟脚
		PWM2(1)	PWM2 可配置输出通道
		C12IN0-	比较器 CMP1 或 CMP2 的负端输入
		ADVIN	AD 外部参考电压输入端
P0.2/AN12 /T0CK/C1OUT /INT0/PWM1 /SDO(1)	I/O	P0.2	带上拉和电平变化中断功能的双向输入输出端口
		AN12	ADC 输入通道 12
		T0CK	T0 时钟输入
		C1OUT	比较器 CMP1 输出端
		INT0	外部中断 0 输入
		PWM1	可配置 PWM1 输出通道
		SDO(1)	可配置 SPI 数据输出端

P0.3/$\overline{RST}$	I	P0.3	带电平变化中断的输入端口
		$\overline{RST}$	外部复位信号输入
P0.4/CLKOUT /$\overline{T1G}$/OSCB /AN13/$\overline{SS}$(1)	I/O	P0.4	带上拉和电平变化中断功能的双向输入输出端口
		$\overline{T1G}$	T1 门控信号输入
		OSCB	外部振荡器输入引脚 B
		VREOUT	2V/3V/4V 参考电压输出
		AN13	ADC 输入通道 13
		$\overline{SS}$	SPI 模式从动选择输入
P0.5/T1CK /OSCA	I/O	P0.5	带上拉和电平变化中断功能的双向输入输出端口
		T1CK	T1 时钟输入
		OSCA	外部振荡器输出引脚 A

2.3.1 P0 口相关的寄存器

表 2-2 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	-	-	P05	P04	P03	P02	P01	P00
45H	P0LR	-	-	P0LR5	P0LR4	-	P0LR2	P0LR1	P0LR0
25H	TR0	-	-	TR05	TR04	TR03	TR02	TR01	TR00
36H	IOCL	-	-	IOCL5	IOCL4	IOCL3	IOCL2	IOCL1	IOCL0
35H	PUR0	-	-	PUR05	PUR04	-	PUR02	PUR01	PUR00

2.3.1.1 P0 口状态读取寄存器（P0）

寄存器 P0 各位对应 P0 口相应引脚的状态，如寄存器 2.2 所示：

寄存器2.2: P0：P0口状态输出寄存器(地址：05H)

	bit7							bit0
复位值 --xx xxxx	-	-	P05	P04	P03	P02	P01	P00
	U	U	R/W	R/W	R/W	R/W	R/W	R/W

P0<5:0>: 读 P0 口各引脚电平状态

1 = 对应引脚为逻辑高电平

0 = 对应引脚为逻辑低电平

图注：R=可读 W=可写 -=未用 U=未实现位

注：读 P0 寄存器实际读 P0 引脚的电平状态。

2.3.1.2 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 口作为输出时，通过写 P0LR 寄存器来设置 P0 口的输出状态。

寄存器2.3: P0LR: P0口输出锁存寄存器(地址：45H)

复位值 xxxx xxxx	bit7							bit0
	-	-	P0LR5	P0LR4	-	P0LR2	P0LR1	P0LR0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P0LR<5:4>: P0 口输出状态

P0LR<2:0>: 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

图注：R=可读 W=可写 -=未用 U=未实现位

注：P0.3 不能作输出用。

2.3.1.3 P0 口方向控制寄存器（TR0）

如寄存器 2.4 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。系统复位时，P0 口各引脚默认为输入口。

寄存器**2.4:** TR0: P0口方向控制寄存器(地址: 25H)

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

TR0<5:4>: P0 口各引脚方向控制位

TR0<2:0>: 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

注 1：TR03，P0.3 引脚控制位，始终为 1。

图注：R=可读 W=可写 -=未用 U=未实现位

2.3.1.4 P0 口上拉功能控制寄存器(PUR0)

KF8F3110/12/20/22/30/32 中除了 P0.3 口没有内部上拉功能外，其它引脚均带有上拉功能，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 IO 端口上拉功能打开，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 2.5 为上拉功能控制寄存器。

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

寄存器2.5: PUR0: P0口上拉控制寄存器(地址：35H)

复位值 1111 -111	bit7							bit0
	-	-	PUR05	PUR04	-	PUR02	PUR01	PUR00
	R/W	R/W	R/W	R/W	U	R/W	R/W	R/W

PUR0<5:4>: P0 上拉功能使能位

PUR0<2:0>: P0 上拉功能使能位

1 = 使能对应端口的上拉功能

0 = 禁止对应端口的上拉功能

图注：R=可读 W=可写 -=未用 U=未实现位

2.3.1.5 P0 口电平变化中断控制寄存器(IOCL)

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。如寄存器 2.6 所示，IOCL 为电平变化中断控制寄存器，将 IOCL 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(P0IF)都会置 1，如果全局中断使能位(AIE)和电平变化中断使能位(P0IE)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 P0IF。

注：1. 只有将引脚设置为数字输入口时才可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口时将会自动禁止该引脚的电平变化中断功能。
2. 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。

寄存器2.6: IOCL: P0口电平变化中断控制寄存器(地址:36H)

复位值 0000 0000	bit7							bit0
	-	-	IOCL5	IOCL4	IOCL3	IOCL2	IOCL1	IOCL0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IOCL<5:0>: P0 口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

图注：R=可读 W=可写 -=未用 U=未实现位

注： P0 口电平变化中断是在引脚的当前电平与上次读 P0 寄存器时的电平不匹配时产生的，所以每次中断标志位(P0IF)置 1 后都要更新 P0 寄存器的值。

电平变化中断参考：

```
    JB      INTCTL,P0IF    ;检测是否为P0电平变化中断
    JMP     INT_END        ;退出中断
    MOV     P0             ;锁存P0口状态
    CLR     INTCTL,P0IF    ;清零电平变化中断标志位
```

2.3.2 P0 口各引脚内部原理功能框图

如图 2.2、2.3 所示，为 P0 口引脚内部原理功能框图。

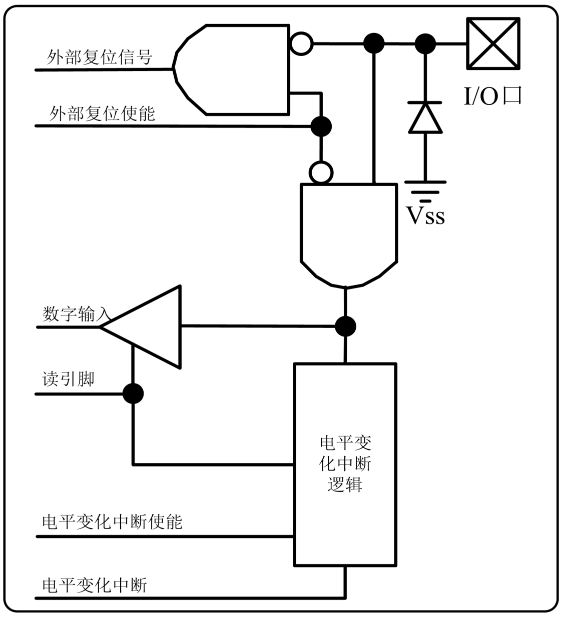


图 2.2 引脚 P0.3 原理功能框图

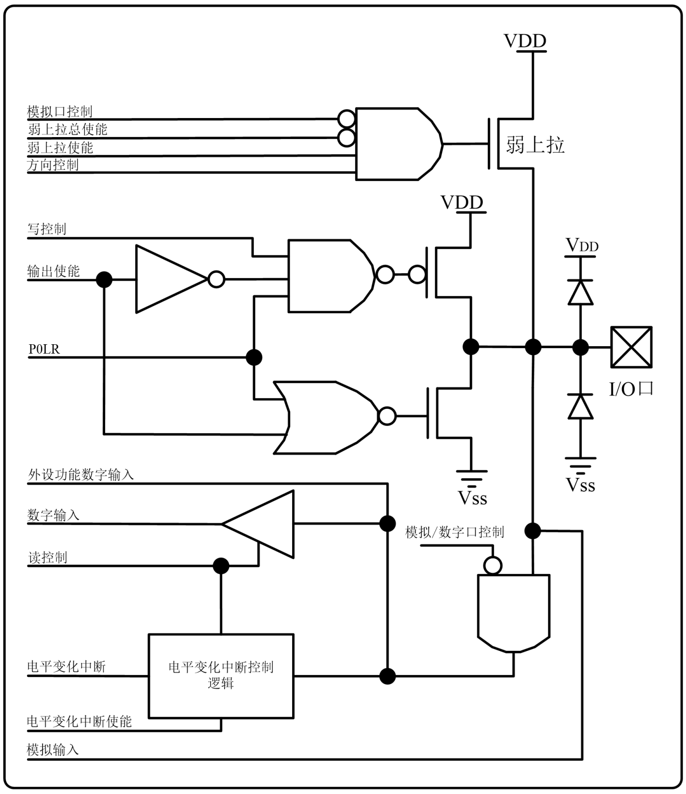


图 2.3 P0.0～P0.2、P0.4～P0.5 口引脚原理框图

2.4 P1 口

KF8F3110/20/30 的 P1 口具有 3 个引脚，引脚功能如表 2-3a 所示。KF8F3112/22/32 单片机 P1 口具有 8 个引脚。引脚功能如表 2-3b 所示。

表 2-3a KF8F3110/20/30 的 P1 口各引脚功能

引脚名	I/O	引脚功能	引脚说明
P1.0/AN0 /PWM2/C2IN+	I/O	P1.0	带上拉的双向输入输出端口
		AN0	ADC 输入通道 0
		PWM2	PWM2 可配置输出通道
		C2IN+	比较器 CMP2 正端输入
P1.1/AN1 /C12IN1- /CCP(1)	I/O	P1.1	带上拉的双向输入输出端口
		AN1	ADC 输入通道 1
		C12IN1-	比较器 CMP1 或 CMP2 的负端输入
		CCP(1)	可配置捕捉输入/比较输出端口
P1.2/AN2/INT1 /P5D/C12IN2-	I/O	P1.2	带上拉的双向输入输出端口
		AN2	ADC 输入通道 2
		INT1	外部中断 1 输入
		P5D	PWM5 输出
		C12IN2-	比较器 CMP1 或 CMP2 的负端输入

表 2-3b KF8F3112/22/32 的 P1 口各引脚功能

引脚名	I/O	引脚功能	引脚说明
P1.0/AN0 /PWM2/C2IN+	I/O	P1.0	带上拉的双向输入输出端口
		AN0	ADC 输入通道 0
		PWM2	PWM2 可配置输出通道
		OPOUT	运放输出端口
		C2IN+	比较器 CMP2 正端输入
P1.1/AN1 /C12IN1- /CCP(1)	I/O	P1.1	带上拉的双向输入输出端口
		AN1	ADC 输入通道 1
		OPIN-	运放输入负端
		C12IN1-	比较器 CMP1 或 CMP2 的负端输入
		CCP(1)	可配置捕捉输入/比较输出端口
P1.2/AN2/INT1 /P5D/C12IN2-	I/O	P1.2	带上拉的双向输入输出端口
		AN2	ADC 输入通道 2
		INT1	外部中断 1 输入
		P5D	PWM5 输出
		OPIN1+	运放正输入端口 1
		C12IN2-	比较器 CMP1 或 CMP2 的负端输入
P1.3/AN3/SDI	I/O	P1.3	带上拉的双向输入输出端口

		AN3	ADC 输入通道 3
		SDI	可配置 SPI 数据输入脚
		SDA	可配置 I2C 数据输入/输出脚
		OPIN2+	运放正输入端口 2
P1.4/AN4/RX/DT	I/O	P1.4	带上拉的双向输入输出端口
		AN4	ADC 输入通道 4
		RX	可配置 USART 全双工异步接收数据输入端
		DT	可配置 USART 半双工同步数据接收端
		OPIN3+	运放正输入端口 3
P1.5/SCK/SCL	I/O	P1.5	带上拉的双向输入输出端口
		SCK	可配置 SPI 时钟脚
		SCL	可配置 I2C 时钟脚
P1.6/TX/CK	I/O	P1.6	带上拉的双向输入输出端口
		TX	可配置 USART 全双工异步发送数据输出端
		CK	可配置 USART 半双工同步时钟发送信号端
P1.7/SDO/AN7	I/O	P1.7	带上拉的双向输入输出端口
		SDO	可配置 SPI 数据输出端
		AN7	ADC 输入通道 7

2.4.1 P1 口相关的寄存器

表 2-4 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
60H	PUR1	PUR17	PUR16	PUR15	PUR14	PUR13	PUR12	PUR11	PUR10

2.4.1.1 P1 口状态读取寄存器（P1）

寄存器 P1 各位对应 P1 口相应引脚的状态，如寄存器 2.7 所示：

寄存器2.7: P1: P1口状态寄存器(地址: 07H)

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

P1<7:0>: 读 P1 口各引脚电平状态

1 = 对应引脚为逻辑高电平

0 = 对应引脚为逻辑低电平

注：读 P1 寄存器实际读 P1 引脚的电平状态。

图注：R=可读 W=可写 -=未用 U=未实现位

2.4.1.2 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，通过写 P1LR 寄存器来设置 P1 口的输出状态。

寄存器2.8: P1LR: P1口输出锁存寄存器(地址: 47H)

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

P1LR<7:0>: 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

图注：R=可读 W=可写 -=未用 U=未实现位

2.4.1.3 P1 口方向控制寄存器(TR1)

如寄存器 2.9 所示，TR1 为 P1 口方向控制寄存器，当 TR1 某位置 1 时，将该引脚设置为输入，此时引脚为三态(悬空)，TR1 某位清 0，对应引脚设置为输出。系统复位时，P1 口各引脚默认为输入口。

寄存器2.9: TR1: P1口方向控制寄存器(地址: 27H)

复位值 1111 1111	bit7							bit0
	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

图注：R=可读 W=可写 -=未用 U=未实现位

2.4.1.4 P1 口上拉功能控制寄存器(PUR1)

KF8F3110/12/20/22/30/32 中 P1 引脚均带有上拉功能，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 IO 端口上拉功能打开，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 2.10 为 P1 上拉功能控制寄存器。

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

寄存器**2.10:** PUR1: P1口弱上拉控制寄存器(地址: 60H)

复位值 1111 1111	bit7							bit0
	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>: P1 上拉功能使能位

1 = 使能对应端口的上拉功能

0 = 禁止对应端口的上拉功能

图注：R=可读 W=可写 -=未用 U=未实现位

2.4.2 P1 口原理功能框图

如图 2.4 所示，为 P1 口内部原理功能框图。

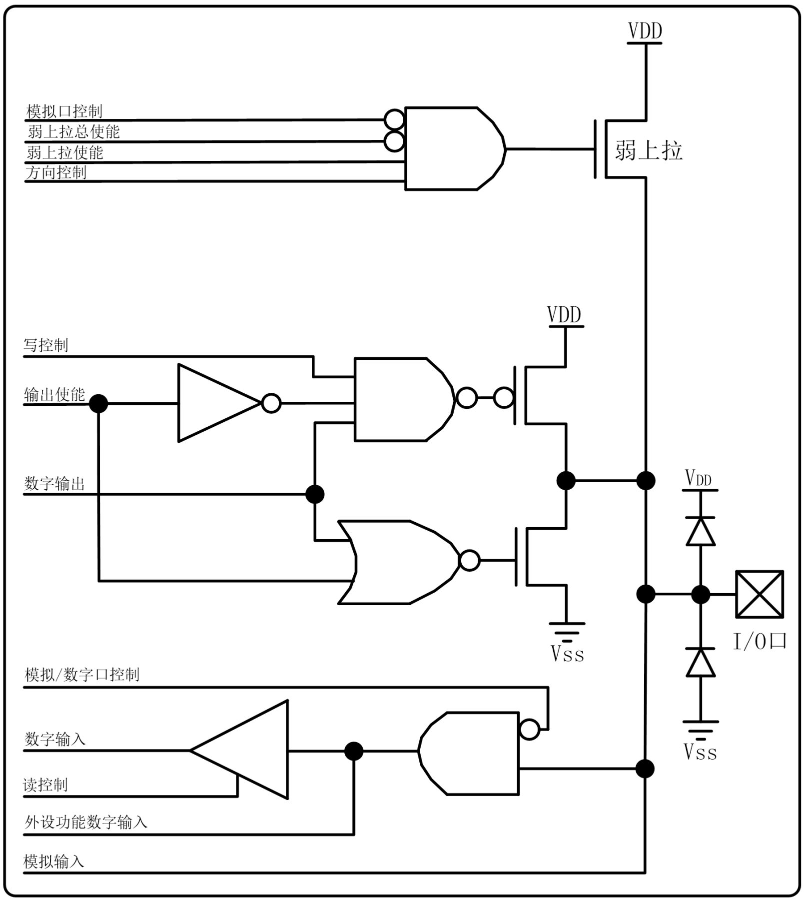


图 2.4 P1 口引脚原理功能框图

2.5 P2 口

KF8F3110/20/30 的 P2 口具有 3 个引脚，如表 2-5a 所示；KF8F3112/22/32 的 P2 口具有 4 个引脚，引脚功能如表 2-5b 所示。

表 2-5a KF8F3110/20/30 的 P2 口各引脚功能

引脚名	I/O	引脚功能	引脚说明
P2.0/CCP/P5A/AN11	I/O	P2.0	带上拉的双向输入输出端口
		CCP	可配置捕捉输入/比较输出端口
		P5A	PWM5 输出
		AN11	ADC 输入通道 11
P2.1/C2OUT/P5B/AN10	I/O	P2.1	带上拉的双向输入输出端口
		C2OUT	比较器 CMP2 输出
		P5B	PWM5 输出
		AN10	ADC 输入通道 10
P2.2/C12IN3-/INT2/P5C/AN9	I/O	P2.2	带上拉的双向输入输出端口
		C12IN3-	比较器 CMP1 或 CMP2 的负端输入
		INT2	外部中断 2 输入
		P5C	PWM5 输出
		AN9	ADC 输入通道 9

表 2-5b KF8F3112/22/32P2 口各引脚功能

引脚名	I/O	引脚功能	引脚说明
P2.0/CCP/P5A/AN11	I/O	P2.0	带上拉的双向输入输出端口
		CCP	可配置捕捉输入/比较输出端口
		P5A	PWM5 输出
		AN11	ADC 输入通道 11
P2.1/C2OUT/P5B/AN10	I/O	P2.1	带上拉的双向输入输出端口
		C2OUT	比较器 CMP2 输出
		P5B	PWM5 输出
		AN10	ADC 输入通道 10
P2.2/C12IN3-/INT2/P5C/AN9	I/O	P2.2	带上拉的双向输入输出端口
		C12IN3-	比较器 CMP1 或 CMP2 的负端输入
		INT2	外部中断 2 输入
		P5C	PWM5 输出
		AN9	ADC 输入通道 9
P2.3/$\overline{SS}$/AN8	I/O	P2.3	带上拉的双向输入输出端口
		$\overline{SS}$	SPI 模式从动选择输入
		AN8	ADC 输入通道 8

2.5.1 P2 口相关的寄存器

表 2-6 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	-	-	-	-	P23	P22	P21	P20
46H	P2LR	-	-	-	-	P2LR3	P2LR2	P2LR1	P2LR0
26H	TR2	-	-	-	-	TR23	TR22	TR21	TR20
61H	PUR2	-	-	-	-	PUR23	PUR22	PUR21	PUR20

2.5.1.1 P2 口状态寄存器(P2)

寄存器 P2 各位对应 P2 口相应引脚的状态，如寄存器 2.11 所示：

寄存器2.11: P2: P2口状态寄存器(地址: 06H)

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

P2<3:0>: 读 P2 口各引脚电平状态
1 = 对应引脚为逻辑高电平
0 = 对应引脚为逻辑低电平

图注：R=可读 W=可写 -=未用 U=未实现位

注：读 P2 寄存器实际读 P2 引脚的电平状态。

2.5.1.2 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，通过写 P2LR 寄存器来设置 P2 口的输出状态。

寄存器2.12: P2LR: P2口输出锁存寄存器(地址: 46H)

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

P2LR<3:0>: 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

图注：R=可读 W=可写 -=未用 U=未实现位

2.5.1.3 P2 口方向控制寄存器(TR2)

如寄存器 2.13 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2.13: TR2: P2口方向控制寄存器(地址: 26H)

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

TR2<3:0>: 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

图注：R=可读 W=可写 -=未用 U=未实现位

2.5.1.4 P2 口上拉控制寄存器 PUR2

KF8F3110/12/20/22/30/32 的 P2 引脚均带有上拉功能，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 IO 端口上拉功能打开，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 2.14 为 P2 上拉功能控制寄存器。

寄存器2.14: PUR2: P2口弱上拉控制寄存器(地址:61H)

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

PUR2<3:0>: P2 上拉功能使能位

1 = 使能对应端口的上拉功能

0 = 禁止对应端口的上拉功能

图注：R=可读 W=可写 -=未用 U=未实现位

2.5.2 P2 口原理功能框图

如图 2.5 所示，为 P2 口内部原理功能框图。

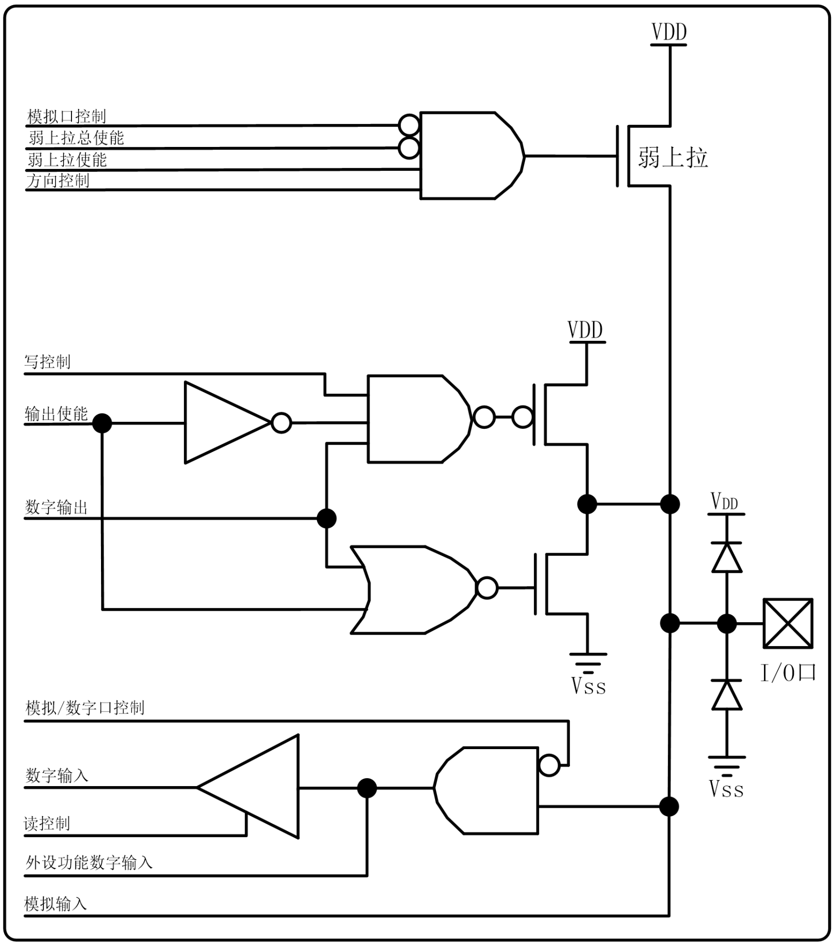


图 2.5 P2 口引脚原理功能框图

3 存储器

如图 3.1 所示，KF8F3110/12/20/22/30/32 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 8K 字节的 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 256×8 位，通用数据寄存器空间为 528(512+16)×8 位。另外 KF8F3110/12/20/22/30/32 中还有一些其它存储器，包括: 工作寄存器组 R0～R7、16 级硬件堆栈、ID 地址单元等。

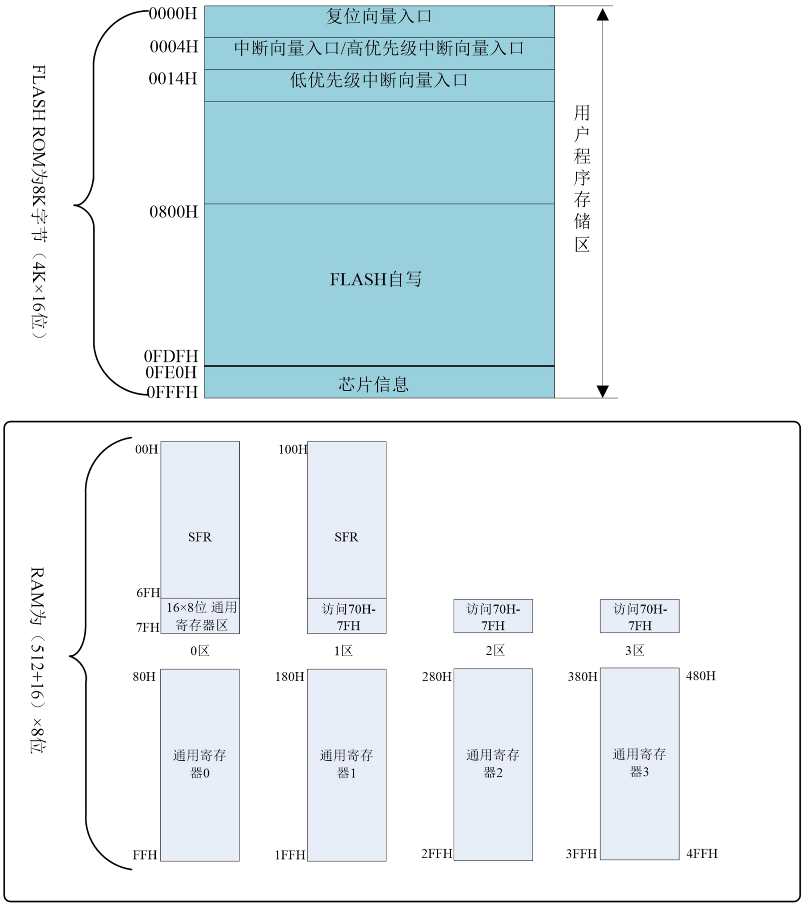


图 3.1 KF8F3110/12/20/22/30/32 存储器组织图

3.1 程序存储器(ROM)区

KF8F3110/12/20/22/30/32 有一个 13 位的程序计数器，实现了 8K 字节的程序存储空间，地址为 0000H～0FFFH，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。

如图 3.2 所示，程序计数器(PC)的低 8 位(PC<7:0>)来自特殊功能寄存器 PCL,高 5 位(PC<12:8>)来自 PCH 寄存器。在任何复位发生后 PC 值将被清 0。在有任何未屏蔽中断发生后 PC 值将指向 0004H 或 0014H 地址。图 3.3 为程序存储器区的地址映射图。

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。

程序计数器

PCH<7:5>	PCH<4:0>	PCL<7:0>

图 3.2　程序计数器（PC）

程序存储器映射

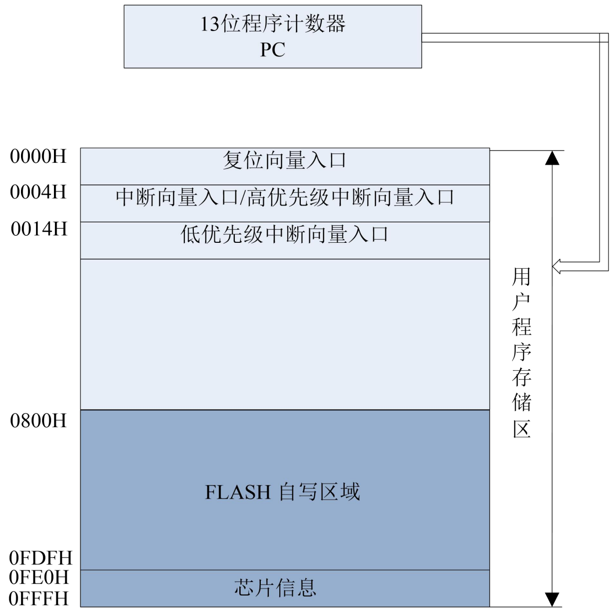


图 3.3　KF8F3110/12/20/22/30/32 程序存储器映射

3.1.1　JMP、CALL 指令

KF8F3110/12/20/22/30/32 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.bit4 以及指令所带立即数（#data12），如图 3.4 所示。

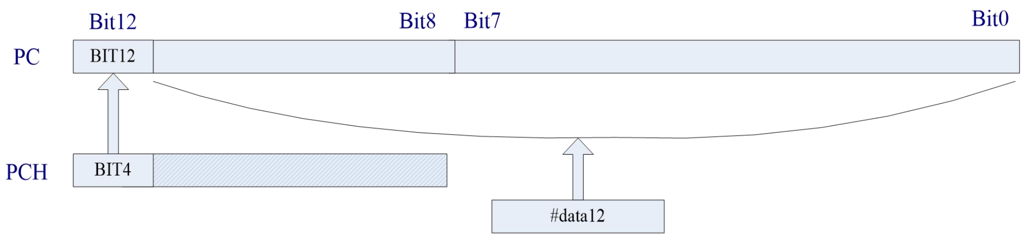


图 3.4 执行 JMP、CALL 指令时 PC 的变化图

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。

3.2　数据存储器(RAM)区

如图 3.5 所示，KF8F3110/12/20/22/30/32 中的数据存储器由 6 个区组成，每个区的空间都是 128 字节，其中 2 个区用作特殊功能寄存器区(SFR)使用；另外 4 个存储器区为通用寄存器区，由用户支配。SFR 地址空间为 00H～7FH、100H～17FH，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。

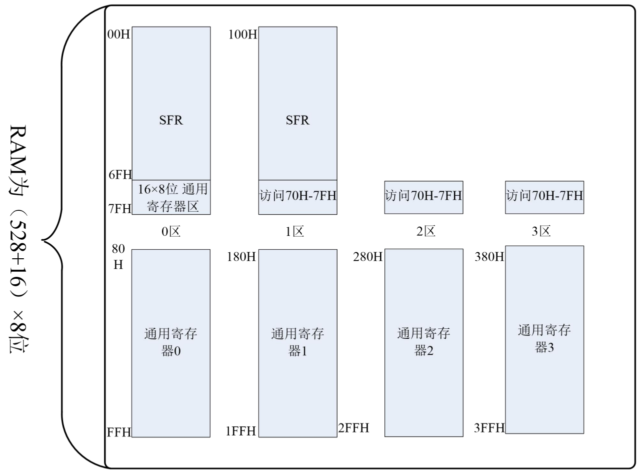


图 3.5 数据存储器地址映射图

3.2.1　通用寄存器区

如图 3.5 所示，通用寄存器的空间为 528 字节，0 区至 3 区通过 BANK 寄存器中的 PR3~PR0 位进行选择，如表 3.1 所示。

寄存器3.1: BANK: 通用寄存器选区寄存器(地址: 17H)

复位值	bit7							bit0
---- 0000	-	-	-	-	PR3	PR2	PR1	PR0
	U	U	U	U	R/W	R/W	R/W	R/W

表 3-1 通用寄存区地址

PR<3:0>	通用寄存器区	地址
0000	通用寄存器 0 区	80H～FFH
0001	通用寄存器 1 区	180H～1FFH
0010	通用寄存器 2 区	280H～2FFH
0011	通用寄存器 3 区	380H～3FFH

注　1：对于未使用到的寄存器值，用作系统保留。

图注：R=可读　W=可写　-=未用　U=未实现位

切换通用寄存区的指令如例 3.2 所示：

例3.2 切换BANK寄存器存储区

```
MOVB   #0X01        ;切换到存储区1区
MOVB   #0X02        ;切换到存储区2区
```

3.2.2 特殊功能寄存器(SFR)区

KF8F3110/12/20/22/30/32 内部的 I/O 口控制、定时/计数器、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信息。

状态字寄存器(PSW): 如寄存器 3.1 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)。$\overline{\text{TO}}$和$\overline{\text{PD}}$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3.2: PSW: 状态字寄存器(地址: 03H)

复位值 0001 1xxx	bit7							bit0
	-	-	-	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	CY
	R/W	R/W	R/W	R	R	R/W	R/W	R/W

$\overline{\text{TO}}$: 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

$\overline{\text{PD}}$: 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)

图注：R=可读　W=可写　-=未用　U=未实现位

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。

3.3 FLASH 自写

KF8F3110/12/20/22/30/32 在程序存储区开辟了一个 2016×16 位的自写区域，地址范围从 800H~0FDFH。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 3-2 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	NVM 地址指针高 8 位							
3BH	NVMADDRL	NVM 地址指针低 8 位							
3CH	NVMCTL0	NVM 控制寄存器 0							
3DH	NVMCTL1	NVM 控制寄存器 1							
39H	NVMDATAL	NVM 数据寄存器低 8 位							
38H	NVMDATAH	NVM 数据寄存器高 8 位							

如图 3.6 所示，写 FLASH 时，FLASH 中所有内存单元以连续的 16 个地址为一个数据块，2 个数据块为一页。

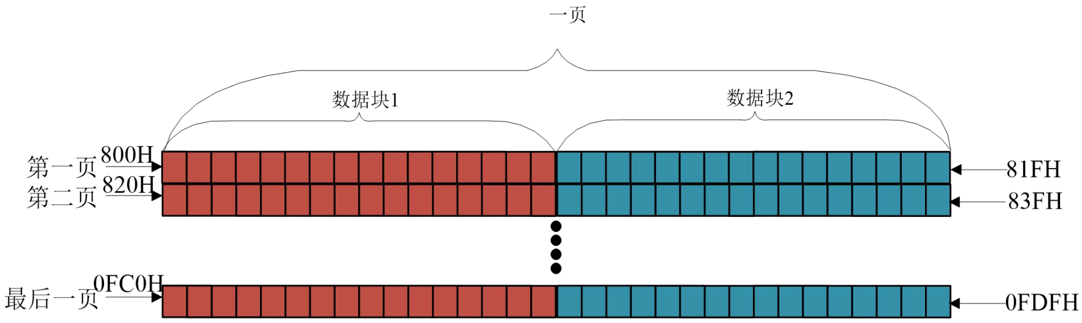


图 3.6 Flash 自写区域地址映射图

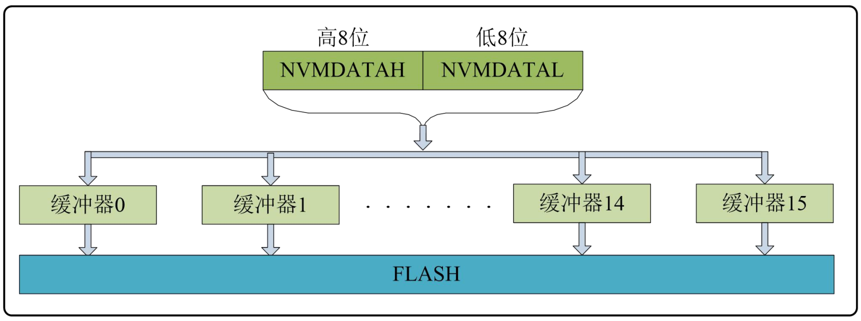


图 3.7 FLASH 写操作

如图 3.7 所示，在写 Flash 时，有 16 个 16 位的缓冲寄存器，用来临时存放要写入 Flash 中的数据。

3.3.1 寄存器 NVMDATAH/L

CPU 读写 Flash 时，寄存器 NVMDATAH/L 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。

寄存器3.2: NVMDATAH: NVM数据高8位(地址: 38H)

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

NVMDATAL: NVM数据低8位(地址: 39H)

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

图注：R=可读 W=可写 -=未用 U=未实现位

3.3.2 寄存器 NVMADDRH/L

如寄存器 3.3 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 12 位的地址信息，NVMADDRH 存放地址的高 4 位，NVMADDRL 存放地址的低 8 位。

寄存器3.3: NVMADDRH: 地址指针高8位(地址: 3AH)

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

NVMADDRL: 地址指针低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

图注：R=可读 W=可写 -=未用 U=未实现位

注：NVMADDRH 的 bit<7:4>必须配置为 0

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，NVMADRRH/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。

3.3.3 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 16 个字或不能被 16 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才会擦除本页的数据，对其它块写操作不会擦除本页数据，可能导致写入数据出错。

注：1.写Flash时，从Flash自写首地址800H开始处，连续的16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操作，且擦除操作只有在写每页的第一个数据块时才会执行，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操作发生。
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.配置位的SWRTEN需配置为0，才能对Flash进行写操作。
5.写Flash工作温度范围为-40℃~105℃。

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

```
MOV  R0 , BANK              ;保存当前寄存器存储区
CLR   BANK                  ;切换到Bank0区
MOV   DATA_BANK , R0        ;该样例要求DATA_BANK在0区,否则添加切区

MOV  R0 , INTCTL            ;保存当前的中断状态
MOV  DATA_INTCTL , R0       ;该样例要求DATA_INTCTL 在0区,否则添加切区
CLR   INTCTL , 7            ;关闭总中断
JNB   INTCTL , 7
JMP   $-2

MOV  R0 , OSCCTL            ;保存当前的时钟状态
MOV  DATA_OSCCTL,R0         ;该样例要求DATA_OSCCTL在0区,否则添加切区
MOV  R0 , #0X20             ;切换到250kHz
MOV  OSCCTL , R0

;;以下时序不可更改
MOV  R0 , #0X84
MOV  NVMCTL0 , R0
MOV  R0 , #0X69
MOV  NVMCTL1 , R0
MOV  R0 , #0X96
MOV  NVMCTL1 , R0
SET   NVMCTL0 ,  1
NOPZ
NOPZ
NOPZ
NOPZ
NOPZ
NOPZ
NOPZ
NOPZ
NOPZ
NOPZ
MOV  R0 , #0X80             ;关闭Flash的写操作，防止意外写
MOV  NVMCTL0 , R0

MOV  R0 , DATA_OSCCTL       ;恢复时钟状态
MOV  OSCCTL,R0

JNB   DATA_INTCTL , 7       ;恢复中断状态
SET   INTCTL , 7

MOV  R0 , DATA_BANK         ;BANK区还原
MOV  BANK , R0
```

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 十六次，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 16 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 十六次，将数据写入本页的第二块。当执行完第 16 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。

3.3.4 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：

```
MOV R0, #0X81
MOV NVMCTL0, R0
NOPZ
NOPZ
```

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 SWRTEN 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. 两个机器周期后该单元的数据被送到 NVMDATAH/L。

3.4 DATA EEPROM

KF8F3110/12/20/22/30/32 片内的 DATA EEPROM 存储器最大容量为 128×8 位，地址范 00H~7FH，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 自写共用同 4 组寄存器。

表 3-3 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	NVM 地址指针低 8 位							
3CH	NVMCTL0	NVM 控制寄存器 0							
3DH	NVMCTL1	NVM 控制寄存器 1							
39H	NVMDATAL	NVM 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。

3.4.1 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。

3.4.2 寄存器 NVMADDRL

DATA EEPROM 最大容量为 128×8 位，地址范围 0~127，只需 7 位地址线参与译码。NVMADDRL 寄存器用来存放要写入 DATA EEPROM 的 7 位的地址信息，最高位不参与译码。

3.4.3 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。

3.4.4 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:

```
MOV  R0 , BANK             ;保存当前寄存器存储区
CLR  BANK                  ;切换到Bank0区
MOV  DATA_BANK , R0        ;该样例要求DATA_BANK在0区,否则添加切区

MOV  R0 , INTCTL           ;保存当前的中断状态
MOV  DATA_INTCTL , R0      ;该样例要求DATA_INTCTL 在0区,否则添加切区
CLR  INTCTL , 7            ;关闭总中断
JNB  INTCTL , 7
JMP  $-2

MOV  R0 , OSCCTL           ;保存当前的时钟状态
MOV  DATA_OSCCTL,R0        ;该样例要求DATA_OSCCTL在0区,否则添加切区
MOV  R0 , #0X20            ;切换到250kHz
MOV  OSCCTL , R0

;;以下时序不可更改
MOV  R0 , #0X04
MOV  NVMCTL0 , R0
MOV  R0 , #0X69
MOV  NVMCTL1 , R0
MOV  R0 , #0X96
MOV  NVMCTL1 , R0
SET  NVMCTL0 , 1
NOPZ
NOPZ
NOPZ
NOPZ
NOPZ
NOPZ
NOPZ
NOPZ
NOPZ
NOPZ
MOV  R0 , #0X00            ;关闭DATA EEPROM的写操作，防止意外写
MOV  NVMCTL0 , R0

MOV  R0 , DATA_OSCCTL      ;恢复时钟状态
MOV  OSCCTL,R0

JNB  DATA_INTCTL , 7       ;恢复中断状态
SET  INTCTL , 7

MOV  R0 , DATA_BANK        ;BANK区还原
MOV  BANK , R0
```

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：
1.CPU写DATA EEPROM时,不管$\overline{\text{DATAP}}$设置为何值，都能写入正确的数据;
2.写DATA EEPROM的工作温度范围为-40℃~105℃。

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。

3.4.5 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 , #0X01
MOV NVMCTL0 , R0
NOPZ
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$\overline{\text{DATAP}}$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 NVMADDRL 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。

3.5 寄存器组 Rn

KF8F3110/12/20/22/30/32 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。

3.6 ID 地址单元

KF8F3110/12/20/22/30/32 的程序存储器空间的最后 32 个地址单元被指定为 ID 地址单元，地址为 0FE0H～0FFFH，用于存放芯片校准信息。

4 汇编指令及寻址方式

4.1 寻址方式

KF8F3110/12/20/22/30/32 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。

4.1.1 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。

4.1.2 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。
指令中含有一个操作数，寻址方式为直接寻址。

4.1.3 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。

4.1.4 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。

4.1.5 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

4.2 汇编指令

KF8F3110/12/20/22/30/32 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。

5 中断

KF8F3110/12/20/22/30/32 单片机的中断源有:

- INT0/1/2 中断
- T0/1/2/3 溢出中断
- P0 口引脚电平变化中断
- A/D 中断
- PWM1/2 中断
- 模拟比较器中断 CMP1/2
- CCP(捕捉/比较/PWM5)中断
- SPI 中断
- I2C 中断
- SSCI 模块 I2C 总线冲突(BCL)中断
- 通用串行通讯接口 USART 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 2 个中断优先级，多个中断源，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>)位进行中断优先级设置。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。

中断逻辑如图 5.2 所示，KF8F3110/12/20/22/30/32 在使用外设中断时需要将外设中断使能位（AIE）置 1 使能外设中断功能。下列中断属于外设中断：

- INT1/2 中断
- T0/1/2/3 溢出中断
- A/D 中断
- PWM1/2 中断
- CMP1/2 中断
- CCP 中断
- SPI 中断
- I2C 中断
- USART 发送/接收中断
- SSCI 模块 I2C 总线冲突(BCL)中断
- 外部时钟故障（OSCFAIL）中断
- DATA EEPROM 写操作(EE)中断

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：

- 中断标志位，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（INTCTL 寄存器相关内容见章节 5.1.1）。只将 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、中断优先级位都被置 1 时，中断将根据设置的中断优先级跳转到地址 0x0004H 或者 0x0014H。进低优先级中断时清 0 AIEL，退出低优先级中断时置 1 AIEL；进高优先级中断时清 AIEH，退出高优先级中断时置 1 AIEH。（AIEH=0 时禁止所有中断）。如果两级中断同时发生，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，等高优先级中断事件结束后再处理低优先级的中断事件。

中断优先级工作原理如图 5.1：

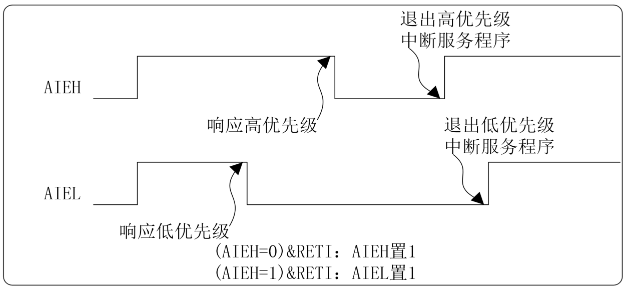


图 5.1 中断优先级工作原理图

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 IPx 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。

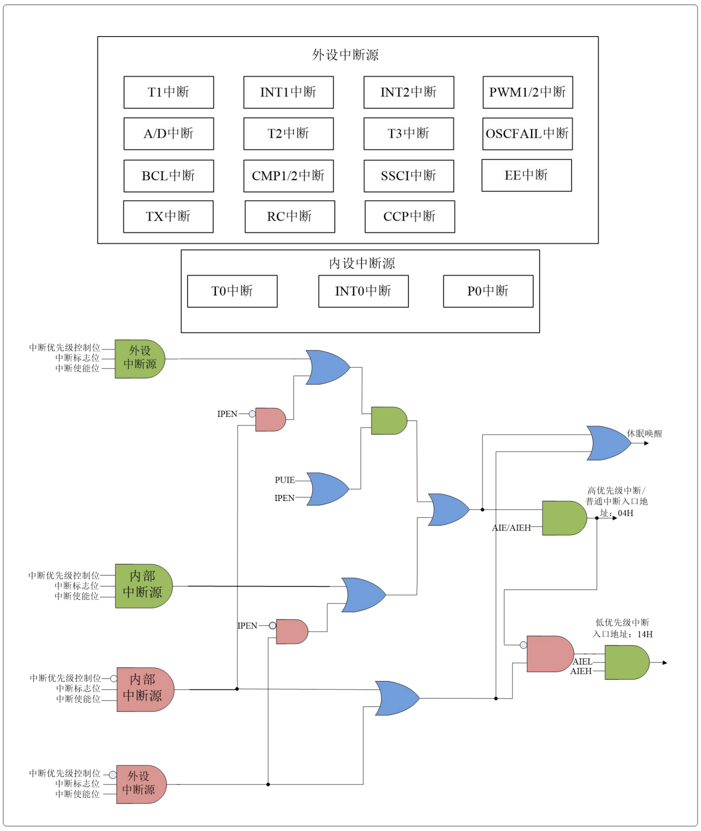


图 5.2 中断逻辑

5.1 中断相关的寄存器

表 5-1 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
2CH	EIE1	EEIE	ADIE	INT2IE	INT1IE	C1IE	PWM2IE	T2IE	T1IE
2DH	EIE2	T3IE	C2IE	RCIE	TXIE	-	CCPIE	BCLIE	SSCIIE
4AH	EIE3	-	-	OSCFAILIE	-	-	-	-	-
0CH	EIF1	EEIF	ADIF	INT2IF	INT1IF	C1IF	PWM2IF	T2IF	T1IF
0DH	EIF2	T3IF	C2IF	RCIF	TXIF	-	CCPIF	BCLIF	SSCIIF
4BH	EIF3	-	-	OSCFAILIF	-	-	-	-	-
22H	IP0	-	-	-	-	-	PT0	PINT0	PP0
23H	IP1	PEE	PADC	PINT2	PINT1	PC1	PPWM2	PT2	PT1
24H	IP2	PT3	PC2	PRC	PTX	-	PCCP	PBCL	PSSCI
29H	IP3	-	-	POSCFAIL	-	-	-	-	-
2EH	PCTL	-	-	-	SLVREN	IPEN	SWDTEN	$\overline{\text{POR}}$	$\overline{\text{LVR}}$
67H	INTEDGCTL	INT2SE	INT1SE	-	-	-	-	-	T1CLKEN

5.1.1 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 5.2 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 5.2 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。
3.AIEL和PUIE是两个地址相同但物理上分开的寄存器，AIEL只有在IPEN=1时才可写，PUIE只有在IPEN=0时才可写；使用时在配置IPEN位之后，再对PUIE（或AIEL）位赋值。

寄存器5.1: INTCTL: 中断控制寄存器(地址: 0BH)

复位值	bit7						bit0
0000 0000	AIE/AIEH	PUIE/AIEL	T0IE	INT0IE	P0IE	T0IF	INT0IF	P0IF
	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中断

0 = 禁止所有低优先级的中断

T0IE: T0 溢出中断使能位

1 = 使能 T0 中断

0 = 禁止 T0 中断

INT0IE: INT0 中断使能位

1 = 使能 INT0 中断

0 = 禁止 INT0 中断

P0IE: 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

T0IF: T0 溢出中断标志位

1 = T0 寄存器溢出

0 = T0 寄存器未溢出

INT0IF: INT0 中断标志位

1 = INT0/P0.2 产生外部中断

0 = INT0/P0.2 未产生外部中断

P0IF: P0 口电平变化中断标志位

1 = 引脚 P0.0～P0.5 至少有一个电平状态发生变化

0 = 引脚 P0.0～P0.5 电平状态未发生变化

图注：R=可读　W=可写　-=未用　U=未实现位

5.1.2 中断使能寄存器 EIE1

如寄存器 5.2 所示，EIE1 是一个可读写的寄存器。

寄存器5.2: EIE1: 中断使能寄存器(地址: 2CH)

复位值 0000 0000	bit7							bit0
	EEIE	ADIE	INT2IE	INT1IE	C1IE	PWM2IE	T2IE	T1IE
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

EEIE: EE 中断使能位
1 = 使能 EE 中断
0 = 禁止 EE 中断

ADIE: AD 中断使能位
1 = 使能 AD 中断
0 = 禁止 AD 中断

INT2IE: INT2 中断使能位
1 = 使能 INT2 中断
0 = 禁止 INT2 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

图注：R=可读 W=可写 -=未用 U=未实现位

5.1.3 中断使能寄存器 EIE2

寄存器**5.3:** EIE2: 中断使能寄存器(地址: 2DH)

复位值 0000 0000	bit7							bit0
	T3IE	C2IE	RCIE	TXIE	-	CCPIE	BCLIE	SSCIIE
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

T3IE: T3 中断使能位
- 1 = 允许 T3 中断
- 0 = 禁止 T3 中断

C2IE: CMP2 中断使能位
- 1 = 使能 CMP2 中断
- 0 = 禁止 CMP2 中断

RCIE: USART 接收中断使能位
- 1 = 允许 USART 接收中断
- 0 = 禁止 USART 接收中断

TXIE: USART 发送中断使能位
- 1 = 允许 USART 发送中断
- 0 = 禁止 USART 发送中断

CCPIE: CCP 中断使能位
- 1 = 允许 CCP 中断
- 0 = 禁止 CCP 中断

BCLIE: BCL 中断使能位
- 1 = 允许 BCL 中断
- 0 = 禁止 BCL 中断

SSCIIE: SSCI 中断使能位
- 1 = 允许 SSCI 中断
- 0 = 禁止 SSCI 中断

图注：R=可读 W=可写 -=未用 U=未实现位

5.1.4 中断使能寄存器 EIE3

寄存器**5.4:** EIE3: 中断使能寄存器(地址: 4AH)

复位值 000- ----	bit7							bit0
	-	-	OSCFAIL IE	-	-	-	-	-
	R/W	R/W	R/W	U	U	U	U	U

OSCFAILIE: 外部时钟故障中断使能位
- 1 = 允许外部时钟故障中断
- 0 = 禁止外部时钟故障中断

图注：R=可读 W=可写 -=未用 U=未实现位

5.1.5 中断标志寄存器 EIF1

寄存器5.5: EIF1: 外设中断标志寄存器(地址0CH)

	bit7							bit0
复位值 0000 0000	EEIF	ADIF	INT2IF	INT1IF	C1IF	PWM2IF	T2IF	T1IF
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

EEIF： EE 中断标志位
1 = 独立 EE 完成操作
0 = 独立 EE 未完成操作

ADIF: AD 完成中断标志位
1 = AD 转换完成
0 = AD 转换没有完成

INT2IF: INT2 中断标志位
1 = INT2 引脚产生外部中断
0 = INT2 引脚未产生外部中断

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

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

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

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
0 = T1 寄存器未溢出

图注：R=可读 W=可写 -=未用 U=未实现位

5.1.6 中断标志寄存器 EIF2

寄存器5.6: EIF2: 外设中断标志寄存器(地址: 0DH)

复位值 0000 0000	bit7						bit0
	T3IF	C2IF	RCIF	TXIF	-	CCPIF	BCLIF	SSCIIF
	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 溢出
0 = T3 未溢出

C2IF： 模拟比较器 CMP2 中断标志位
1 = 模拟比较器 CMP2 输出发生改变
0 = 模拟比较器 CMP2 输出未发生改变

RCIF: USART 接收中断标志位
1 = USART 接收缓冲器满（通过 RXSDR 清 0）
0 = USART 接收缓冲器空

TXIF: USART 发送中断标志位
1 = USART 发送缓冲器满（通过 TXSDR 清 0）
0 = USART 发送缓冲器空

CCPIF： CCP 中断标志位
1 = CCP 产生了中断
0 = CCP 未产生中断

BCLIF: BCL 中断标志位
1 = BCL 产生了中断
0 = BCL 未产生中断

SSCIIF: SSCI 中断标志位
1 = SSCI 产生了中断
0 = SSCI 未产生中断

图注：R=可读 W=可写 -=未用 U=未实现位

5.1.7 中断标志寄存器 EIF3

寄存器5.7: EIF3: 中断标志寄存器(地址: 4BH)

复位值 000- ----	bit7							bit0
	-	-	OSCFAILIF	-	-	-	-	-
	R/W	R/W	R/W	U	U	U	U	U

OSCFAILIF:外部时钟故障中断标志位
1 = 外部时钟发生故障
0 = 外部时钟未发生故障

图注：R=可读 W=可写 -=未用 U=未实现位

5.1.8 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。如寄存器 5.4 所示，中断优先级控制寄存器 IP0 包含：

寄存器5.8: IP0: 中断优先级控制寄存器0(地址: 22H)

复位值 ---- -000	bit7							bit0
	-	-	-	-	-	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

图注：R=可读 W=可写 -=未用 U=未实现位

5.1.9 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。如寄存器 5.5 所示，中断优先级控制寄存器 IP1 包含：

寄存器5.9: IP1: 中断优先级控制寄存器1(地址: 23H)

复位值 0000 0000	bit7							bit0
	PEE	PADC	PINT2	PINT1	PC1	PPWM2	PT2	PT1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PEE: 独立 EE 中断优先级控制位
- 1 = 独立 EE 中断为高优先级
- 0 = 独立 EE 中断为低优先级

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

PINT2: INT2 中断优先级控制位
- 1 = INT2 中断为高优先级
- 0= INT2 中断为低优先级

PINT1: INT1 中断优先级控制位
- 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 中断优先级控制位
1 = T1 中断为高优先级
0 = T1 中断为低优先级

图注：R=可读 W=可写 -=未用 U=未实现位

5.1.10 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。如寄存器 5.8 所示，中断优先级控制寄存器 IP2 包含：

寄存器**5.10:** IP2: 中断优先级控制寄存器2(地址: 24H)

复位值 0000 0000	bit7							bit0
	PT3	PC2	PRC	PTX	-	PCCP	PBCL	PSSCI
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

PC2： 比较器 CMP2 中断优先级控制位
1 = 比较器 CMP2 中断为高优先级
0 = 比较器 CMP2 中断为低优先级

PRC: USART 接收中断优先级控制位
1 = USART 接收中断为高优先级
0 = USART 接收中断为低优先级

PTX: USART 发送中断优先级控制位
1 = USART 发送中断为高优先级
0 = USART 发送中断为低优先级

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

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

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

图注：R=可读 W=可写 -=未用 U=未实现位

5.1.11 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。

寄存器5.11: IP3: 中断优先级控制寄存器3(地址: 29H)

复位值 000- ----	bit7							bit0
	-	-	POSCFAIL	-	-	-	-	-
	R/W	R/W	R/W	U	U	U	U	U

POSCFAIL: OSCFAIL 中断优先级控制位

1 = OSCFAIL 中断为高优先级

0 = OSCFAIL 中断为低优先级

图注：R=可读 W=可写 -=未用 U=未实现位

5.1.12 电源控制寄存器 PCTL

寄存器5.12: PCTL：电源控制寄存器(地址: 2EH)

复位值 ---1 000x	bit7							bit0
	-	-	-	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，即为优先级模式

0 = 禁止中断优先级功能，即为普通模式

SWDTEN: 软件看门狗定时器使能位

当配置字的 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

当配置字的 WDTEN=1 时

为无关位

$\overline{\text{POR}}$: 上电复位状态位

1 = 未发生上电复位

0 = 发生了上电复位

$\overline{\text{LVR}}$: 欠压检测复位状态位

1 = 未发生欠压检测复位

0 = 已发生欠压检测复位

图注：R=可读 W=可写 -=未用 U=未实现位

5.1.13 INT 中断沿选择寄存器 INTEDGCTL

寄存器5.13:INTEDGCTL:INT中断沿选择寄存器（67H）

复位值 11-- ---0	bit7							bit0
	INT2SE	INT1SE	-	-	-	-	-	T1CLKEN
	R/W	R/W	U	U	U	U	U	R/W

INT2SE: INT2 触发脉冲边沿选择位
1 = 上升沿触发
0 = 下降沿触发

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

T1CLKEN： T1 定时模式时钟源选择位
当 T1CS=0 时：
1 = T1 时钟为内部高频振荡器时钟 INTHF
0 = T1 时钟为机器时钟 SCLK/4
当 T1CS=1 时：
无关位

图注：R=可读 W=可写 -=未用 U=未实现位

5.1.14 中断响应

当 IPEN=1 时，为优先级模式，中断被响应后：

1. 返回地址压入堆栈；
2. 中断入口地址载入 PC；
3. 在中断被响应前，AIEH 和 AIEL 位的设置会影响到中断响应的过程：

- 将 AIEH 和 AIEL 均置 1，可进入高优先级或优先级中断。进入的是高或低优先级中断时，AIEH 或 AIEL 由硬件自动清 0，执行高或低优先级中断服务程序，执行指令 IRET 退出高或低优先级中断，硬件自动将 AIEH 或 AIEL 置 1；
- 只将 AIEH 置 1，则直接进入高优先级中断，AIEH 由硬件自动清 0，继续执行中断服务子程序，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；

4. 跳转到中断发生处继续执行下面的程序。

注： 中断的响应有一定的响应时间：

1.如果已经进入了低优先级中断，再有高优先级中断，此时为中断嵌套；

2.如果还没有进入低优先级中断，同时有高优先级中断产生，则直接进入高优先级中断，相当于两个中断同时产生。

当 IPEN=0 时，为普通模式，中断被响应后:

1. 返回地址压入堆栈；
2. 中断入口地址载入 PC；
3. AIE 位将被硬件清 0 以禁止其它中断；
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，同时硬件自动将 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。

5.2 INT 中断

INT 中断有三个中断源: INT0、INT1 和 INT2，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。

5.2.1 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 位来设置触发边沿，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。

5.2.2 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 INTEDGCTL 中的 INT1SE 位来设置触发边沿，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。

5.2.3 INT2 中断

INT2 中断通过寄存器 EIE1 中的 INT2IE 位置 1 使能 INT2 中断。通过 INTEDGCTL 中的 INT2SE 位来设置触发边沿，INT2SE 置 1，将 INT2 设置为上升沿触发，清零设置为下降沿触发。EIF1 中的 INT2IF 为 INT2 的中断标志位。如果 IPEN 和 PINT2 位均置 1，则 INT2 为高优先级中断。

INT2 引脚有触发脉冲时，INT2IF 被自动置 1，如果 INT2IE、PUIE 和 AIE 位为 1，则响应 INT2 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，上升沿触发(INT0/1/2SE=1)还是下降沿触发(INT0/1/2SE=0)；
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；
4. 优先级模式需设置对应的优先级允许位为 1，普通中断模式需设置 AIE 为 1（INT1/2 还需要设置 PUIE 为 1）；

注:（1）x=0/1/2 定时器中断。

5.3 定时器中断

T0/1/3 计数寄存器发生溢出时，T0IF/T1IF/T3IF 位将会被置 1。当 T2 与 PP5 匹配时，T2IF 将被置 1。通过将 T0IE/T1IE/T2IE/T3IE 位置 1/清 0 可使能/禁止该中断。当 IPEN 和 PTx(x=0,1,2,3)置 1 时，定时器中断配置为高优先级中断。

有关定时/计数器模块中断的操作，请参考定时/计数器部分。

5.4 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 口中断配置为高优先级中断。

有关 P0 口的操作，请参考 P0 口部分。

5.5 PWM 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 均置 1 时，PWM1 中断配置为高优先级中断。当 IPEN 和 PPWM2 均置 1 时，PWM2 中断配置为高优先级中断。

详见 PWM 部分。

5.6 模拟比较器中断

当模拟比较器的控制寄存器 C1CTL 的 C1EN 为 1 时，模拟比较器 CMP1 将使能，开始工作。如果 CMP1 的正端输入大于负端输入时，将产生对应的中断标志。

当 IPEN 和 PC1 均置 1 时，模拟比较器中断配置为高优先级中断。模拟比较器 2 的工作方式与比较器 1 相似。

详见模拟比较器部分。

5.7 CCP 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 CCPIF 置 1；如果使能 INTCTL 中的 AIE、PUIE 和 EIE2 中的 CCPIE，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 PWM5H0：PWM5L0 与 T1H 和 TIL 匹配时，中断请求标志位 EIF2 寄存器中的 CCPIF 置 1；如果使能 INTCTL 中的 AIE、PUIE 和 EIE2 中的 CCPIE，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 PWM5 后，当 T2L 和 T2H 与 PP5H 和 PP5L 和匹配时，会触发相应的中断标志 T2IF。如果 T2IE 使能，则会触发中断（AIE、PUIE 置 1）。如果 IPEN 和 PT2 位均置 1，则为高优先级中断。

详见 CCP 部分。

5.8 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。

5.9 SSCI 中断

同步串行端口 SSCI 的中断分为 SSCI 中断和 BCL 中断。详见 SSCI 模块部分。

5.10 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。

6 定时/计数器

KF8F3110/12/20/22/30/32 单片机提供一个 8 位的定时/计数器 T0、1 个 16 位的定时/计数器 T1、1 个 16 位定时器 T2 和 1 个 16 位的定时器 T3。

6.1 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，T0 的值再加 1 则会产生溢出，T0 寄存器的值返回到 0 开始重新计数。

6.1.1 T0 原理框图

图 6.1 为 T0 的结构框图。T0 模块使用一个 8 位计数器作为预分频器，如寄存器 6.1 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。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。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。

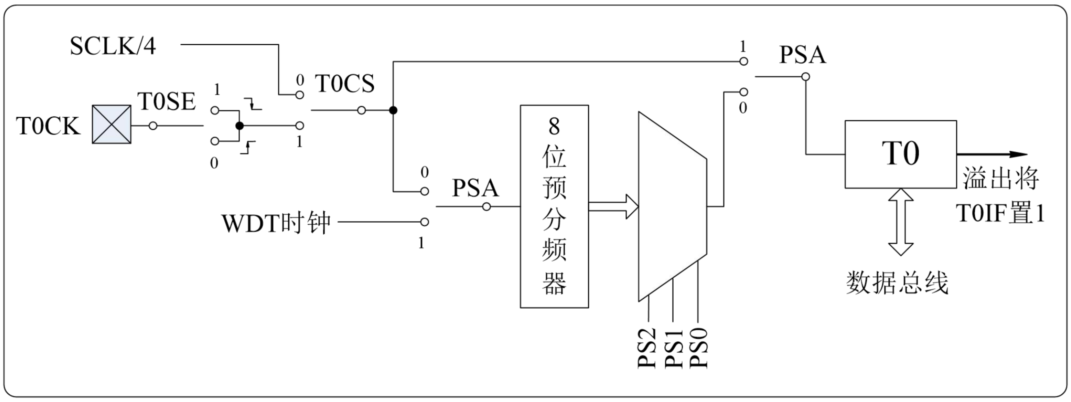


图 6.1 原理框图

6.1.2 T0 相关的寄存器

表 6-1 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	定时/计数器 0 寄存器							
21H	OPTR	$\overline{\text{PUPH}}$	INT0SE	T0CS	T0SE	PSA	PS2	PS1	PS0

6.1.2.1 OPTR 选择寄存器

寄存器6.1: OPTR: 选择寄存器(地址: 21H)

复位值 1111 1111	bit7						bit0
	$\overline{\text{PUPH}}$	INT0SE	T0CS	T0SE	PSA	PS2	PS1	PS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

$\overline{\text{PUPH}}$: IO（P0/P1/P2）端口上拉功能总使能位
 1 = 禁止所有 IO 端口上拉功能
 0 = 允许 IO 端口使用上拉功能

INT0SE: INT0 中断触发脉冲边沿选择位
 1 = INT0/P0.2 为上升沿触发
 0 = INT0/P0.2 为下降沿触发

T0CS: T0 模式选择位
 1 = 计数模式，T0 的时钟为外部时钟 T0CK/P0.2
 0 = 定时模式，T0 的时钟为机器时钟 SCLK/4

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

图注：R=可读 W=可写 -=未用 U=未实现位

6.1.3 定时模式

通过将 T0CS 位(OPTR.5)清 0 可选择定时器模式。在定时模式中，如果不使用预分频器，每一个机器周期 T0 寄存器的值加 1。如果 T0 寄存器被写入初始值，则在接下来的两个指令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。

6.1.4 计数模式

通过将 T0CS 位(OPTR.5)置 1 可选择计数模式。在该模式下，T0 模块在 T0CK 引脚信号的每一次上升沿(T0SE 位清 0)或下降沿(T0SE 位置 1)递增计数。

当不使用预分频器时，要求 T0CK 的高电平状态和低电平状态分别保持至少 $2T_{sys}$ 的时间，以实现 T0CK 与内部相位时钟的同步。

6.1.5 T0 的使用

T0 在使用时通过以下步骤进行设置:

1．通过将 T0CS 位清 0/置 1 选择定时/计数模式(如果是计数模式，再设置 T0SE 选择脉冲触发边沿，将对应的计数脉冲输入脚 T0CK 设置为数字输入);

2．如果需要分频，则将预分频器分配给 T0，并设置分频比;

3．给 T0 寄存器设置初始值;

4．如果使用中断方式则将 T0IE 和 AIE 位置 1。

6.2 定时器/计数器 T1

T1 是一个 16 位的定时器/计数器，T1 的低 8 位在寄存器 T1L 中，高 8 位在寄存器 T1H 中，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 T1IF 置 1。T1 属于外设单元，因此在使用 T1 中断时，需将 PUIE 位置 1，使能外设中断。如图 6.2 所示为 T1 的原理框图。

6.2.1 T1 原理框图

T1 的原理框图如图 6.2 所示，T1 是一个带有门控和预分频的 16 位定时器/计数器，计数时钟可选择外部时钟或者内部时钟，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 $\overline{\text{T1SY}}$位(T1CTL<2>)设置可使 T1 工作在异步计数器模式。

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。

此外，T1 还带有重载功能，重载寄存器利用 PP2/PP1 设置。当使能 T1 重载功能时，T1 计数器计数到 T1 重载寄存器中设置的值时，T1 计数器将清 0 重新开始计数，且将 T1 中断标志位置 1。

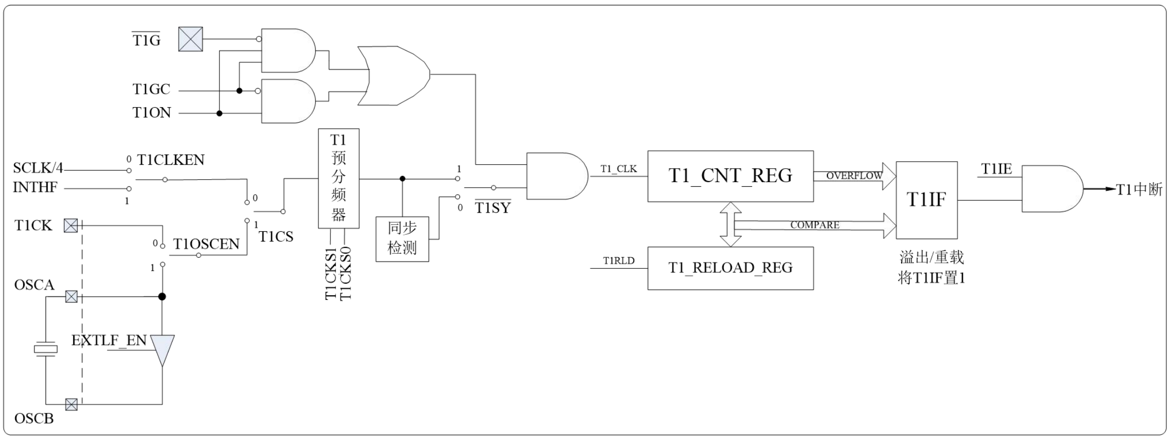


图 6.2 T1 原理框图

6.2.2 T1 时钟

当 T1CTL 寄存器的 T1CS 位置 1 时，T1 工作在计数模式下，T1 的工作时钟源通过 T1OSCEN 位来选择，当 T1OSCEN=1 时，T1 时钟为外部低频时钟，当 T1OSCEN=0 时，T1 时钟为 T1CK。当 T1CTL 寄存器的 T1CS 位清 0 时，T1 工作在定时模式下，T1 的工作时钟源通过 INTEDGCTL 寄存器的 T1CLKEN 位来选择，当 T1CLKEN=1 时，T1 时钟为内部高频振荡器时钟 INTHF，当 T1CLKEN=0 时，T1 时钟为机器时钟 SCLK/4。

6.2.3 T1 相关的寄存器

表 6-2 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	定时/计数器 1 低字节寄存器							
0FH	T1H	定时/计数器 1 高字节寄存器							
10H	T1CTL	T1RLD	T1GC	T1CKS1	T1CKS0	T1OSCEN	$\overline{\text{T1SY}}$	T1CS	T1ON

6.2.3.1 T1 控制寄存器

如寄存器 6.2 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6.2: T1CTL: T1控制寄存器(地址: 10H)

复位值 0000 0000	bit7							bit0
	T1RLD	T1GC	T1CKS1	T1CKS0	T1OSCEN	$\overline{\text{T1SY}}$	T1CS	T1ON
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

T1RLD: T1 重载功能使能位

0 = 禁止 T1 重载功能

1 = 使能 T1 重载功能

T1GC: T1 门控使能位

如果 T1ON=0 则该位被忽略

如果 T1ON=1 则:

1 = 使能 $\overline{\text{T1G}}$ 引脚控制(如果 $\overline{\text{T1G}}$ 引脚为低电平，启动 T1，为高电平，关闭 T1)

0 = 禁止 $\overline{\text{T1G}}$ 引脚控制

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

T1OSCEN: T1 外部低频时钟使能位

1 = 使能外部低频时钟源作为 T1 计数时钟

0 = T1CK 作为 T1 计数时钟

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

T1CS=1 且 T1OSCEN=0:

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

T1CS=0 或 T1OSCEN=1: 该位被忽略

T1CS: T1 定时/计数模式选择

1 = 计数模式，T1 时钟为外部时钟

当 T1OSCEN=1 时，T1 时钟为外部低频时钟

当 T1OSCEN=0 时，T1 时钟为 T1CK

0 = 定时模式，T1 时钟可选择内部高频时钟或机器时钟
当 T1CLKEN=1 时，T1 时钟为内部高频振荡器时钟 INTHF
当 T1CLKEN=0 时，T1 时钟为机器时钟 SCLK/4

T1ON: T1 启动控制位
1 = 启动 T1
0 = 停止 T1

图注：R=可读 W=可写 -=未用 U=未实现位

6.2.4 定时模式

通过将 T1CS 位清 0 将 T1 设定为定时模式。T1 工作在定时模式时，通过设置 T1CLKEN 选择内部时钟源进行计数：

当 T1CLKEN=1 时，T1 时钟为内部高频振荡器时钟 INTHF；

当 T1CLKEN=0 时，T1 时钟为机器时钟 SCLK/4。

T1 溢出，将 T1 中断标志位 T1IF 置 1。

如果使能 T1 门控引脚，且 T1ON=1，则在 $\overline{\text{T1G}}$ 引脚为低电平时，启动 T1，如果 $\overline{\text{T1G}}$ 引脚为高电平，禁止 T1。使用该方式可粗略的对 $\overline{\text{T1G}}$ 引脚的低电平持续时间进行计算。

6.2.5 计数模式

通过将 T1CS 位置 1 将 T1 设定为计数模式，通过设置 T1OSCEN 位选择计数时钟为 T1CK 或者外部低频时钟：

当 T1OSCEN=1 时，T1 时钟为外部低频时钟

当 T1OSCEN=0 时，T1 时钟为 T1CK

当 T1CK 作为 T1 时钟时，有同步计数和异步计数两种方式。如果控制位 $\overline{\text{T1SY}}$(T1CTL.2) 置 1，则 T1 工作在异步计数模式。计数器根据 T1CK(当 T1OSCEN=0 时)引脚的脉冲进行递增计数。在休眠模式下，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 $\overline{\text{T1SY}}$ (T1CTL.2)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 T1CK 引脚电平进行采样，可以实现 T1CK 与内部相位时钟的同步。

6.2.6 T1 重载功能

置位 T1CTL 寄存器的 T1RLD 位使能 T1 重载功能。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。

T1RLD 位置 1 时会载入一次 PP2/PP1 寄存器的值，当置位 T1RLD 且在 T1 计数过程中对 PP2/PP1 寄存器写入新值时，T1 模块会在下一次重载时载入新的重载点。

6.2.7 T1 在休眠模式下的运行

T1 在计数器模式时可在休眠模式下工作。当 T1CK 作为 T1 计数时钟时，须配置为异步模式才可工作在休眠模式。

T1 中断使能方法：

- 将 T1IE 位(EIE1.0)置 1
- 将 PUIE 位(INTCTL.6)置 1

器件将在溢出时被唤醒。如果 AIE 位(INTCTL.7)置 1，器件将被唤醒并跳转至中断服务程序。

6.2.8 T1 分配给 PWM1/2

当使用 PWM1/2 时需要用到 T1，单片机将 T1L，T1IE，T1IF 分配给 PWM1,T1H 分配给 PWM2，具体使用方法参见 PWM1/2 部分。

6.3　　定时器 T2

注意，8 脚型号 KF8F3110SB 无 T2 资源，其余型号均带有定时器 T2。

T2 是一个带 16 位周期寄存器、预分频器和后分频器的 16 位定时器，由 2 个 8 位寄存器 T2H 和 T2L 组成。没有外部计数时钟输入脚，只有内部高频振荡器时钟可用。

6.3.1　　T2 的工作原理

如图 6.3 所示，为 T2 的原理框图，T2 的计数时钟为内部高频振荡器时钟（INTHF），每个计数周期(分频器 1 分频比为 1:1 时)寄存器 T2 的值自动加 1。通过 T2CTL0 的 T2ON 启动/禁止 T2。

INTHF 先经过分频器 1 分频后送到 T2 寄存器，进行加 1 计数，分频器 1 通过设置寄存器 T2CTL0 中的 T2CKPS<1:0>位，可将 INTHF 进行 1:1、1:4 和 1:16 分频。

当 T2H/L 寄存器与 PP5H/L 相等时，T2 自动清 0，发出相等信号给分频器 2，分频器 2 递增。同时 T2H/L 的值与寄存器 T2CCR0H/L 相比较，如果等于 T2CCR0H/L，且对应的控制位 T2CCR0ON 置 1，则会发出信号，使 START（ADCCTL0<1>）位置 1。

分频器 2 的分频比可通过寄存器 T2CTL0 中的 T2CKBS<3:0>位设置为 1:1 至 1:16。当其设置为 1:1 时，每次寄存器 T2H/L 与 PP5H/L 相等，将会使 T2 中断标志位 T2IF 置 1；当其设置为 1:2 时，寄存器 T2H/L 与 PP5H/L 相等累计两次才会使 T2IF 置 1，以此类推。

如果对 T2 寄存器执行写操作、对寄存器 T2CTL0 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、看门狗复位或欠压检测复位)，分频器 1/2 的计数器将被清 0。

寄存器 T2H/L、PP5H/L 和 T2CCR0H/L 均可读写。任何复位时，寄存器 T2H/L 均被设置为 00H，T2CCRxH/L 被设置为 00H/00H，PP5H/L 被设置为 00H。用户可根据需求自行设置 T2H/L 和 PP5H/L 来产生定时配置。

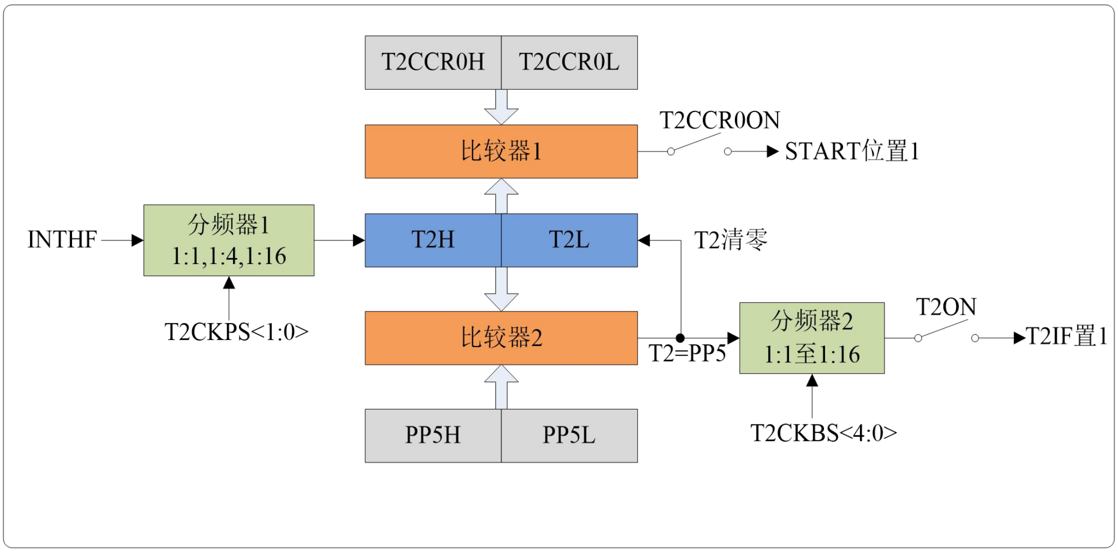


图 6.3　定时器 T2 原理框图

注　1：T2CCR0H/L匹配触发AD信号和T2中断信号需在T2使能后才会产生

6.3.2 T2 相关的寄存器

表 6-3 与 T2 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
12H	T2CTL0	-	T2CKBS3	T2CKBS2	T2CKBS1	T2CKBS0	T2ON	T2CKPS1	T2CKPS0
10FH	T2CTL1	T2MOD1	T2MOD0	T2OFS1	T2OFS0	T2TRIGEN	-	-	T2DIR
41H	T2H	定时器 2 高字节寄存器							
11H	T2L	定时器 2 低字节寄存器							
42H	PP5H	PWM5 周期寄存器高 8 位							
52H	PP5L	PWM5 周期寄存器低 8 位							
54H	T2CCR0L	T2 启动 ADC 设置寄存器 0 低 8 位							
40H	T2CCR0H	T2 启动 ADC 设置寄存器 0 高 8 位							

6.3.2.1 T2 控制寄存器 T2CTL0

寄存器6.3: T2CTL0: T2控制寄存器0(地址: 12H)

复位值 -000 0000	bit7							bit0
	-	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

图注：R=可读 W=可写 -=未用 U=未实现位

6.3.2.2 T2 控制寄存器 T2CTL1

寄存器6.4: T2CTL1: T2控制寄存器1(地址:10FH)

复位值 0000 0--x	bit7							bit0
	T2MOD1	T2MOD0	T2OFS1	T2OFS0	T2TRIGEN	-	-	T2DIR
	R/W	R/W	R/W	R/W	R/W	U	U	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

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

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

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

图注：R=可读 W=可写 -=未用 U=未实现位

6.3.2.3 T2CCR0H 和 T2CCR0L 寄存器

T2CCR0H/L 寄存器是 T2 启动 A/D 转换的寄存器。通过 T2CCR0ON 位可以控制相应的 T2CCR0H/L 是否工作。当 T2CCR0ON=1 时，此时 T2CCR0H/L 将工作，T2H/L 的值与寄存器 T2CCR0H/L 相比较，如果等于 T2CCR0H/L，则会发出信号，使 ADCCTL0 的 START 位置 1，使能 A/D 转换。

6.3.3 T2 中断

如图 6.3 所示，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 T2IE 为 1，且全局中断和外设中断允许位为 1，将会响应 T2 中断。如果 IPEN 和 PT2 位均置 1，则 T2 为高优先级中断。

6.3.4 T2 在休眠模式

单片机进入休眠模式后，因 T2 使用内部高频振荡器时钟源，所以 T2 将停止工作。

6.3.5 T2 分配给 PWM5

PWM5 在使用时，需要使用 T2 进行定时，PWM5 的详细内容请参考 PWM5 部分。

6.4 定时器 T3

T3 为 16 位定时器，时钟源为振荡器时钟源（Sosc）或内部高频振荡器源（INTHF）可选；通过 T3CTL 寄存器的 T3EN 位控制定时器启动。

6.4.1 T3 原理框图

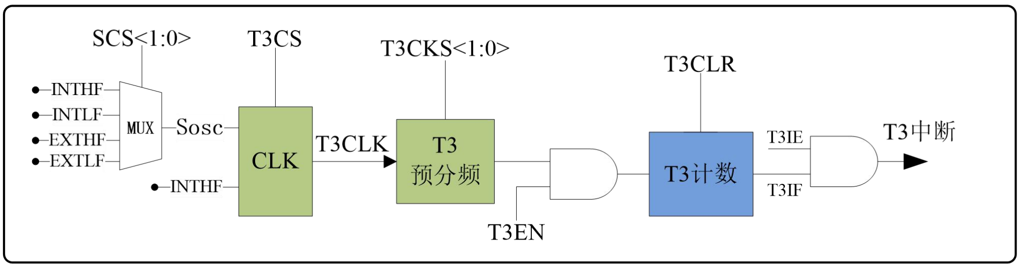


图 6.4 定时器 T3 原理框图

6.4.2 T3 相关寄存器

表 6-4 与 T3 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
2FH	OSCCTL	CLKOE	IRCS2	IRCS1	IRCS0	SCS1	SCS0	IESO	FSCM
4EH	T3CTL	T3EN	T3CKS1	T3CKS0	T3CS	-	-	-	-
53H	T3CTL1	-	-	T3CLR	-	-	-	-	-
4FH	T3L	定时器 3 低字节寄存器							
5FH	T3H	定时器 3 高字节寄存器							

6.4.2.1 T3 控制寄存器 T3CTL

寄存器6.5: T3CTL: T3控制寄存器(地址: 4EH)

复位值 0000 0000	bit7							bit0
	T3EN	T3CKS1	T3CKS0	T3CS	-	-	-	-
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

T3EN: 定时器 T3 使能位

0 = 禁止定时器 T3

1 = 使能定时器 T3

T3CKS<1:0>: 定时器 T3 时钟预分频比

00 = 1/1 倍预分频比

01 = 1/2 倍预分频比

10 = 1/4 倍预分频比

11 = 1/8 倍预分频比

T3CS: 定时器 T3 时钟源选择位

0 = T3 时钟源为振荡器时钟源 Sosc

1 = T3 时钟源为内部高频振荡器时钟源 INTHF

图注：R=可读 W=可写 -=未用 U=未实现位

6.4.2.2 T3 控制寄存器 T3CTL1

寄存器6.6: T3CTL1: T3控制寄存器1(地址: 53H)

复位值	bit7							bit0
0000 0000	保留	保留	T3CLR	-	-	-	-	-
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

T3CLR： T3 计数寄存器复位位

0 = T3 计数寄存器在复位状态

1 = T3 计数寄存器退出复位状态

图注：R=可读 W=可写 -=未用 U=未实现位

注：

1、T3 计数寄存器能够实现正常计数，必须在配置 T3H 和 T3L 寄存器前将 T3CLR 置 1。

2、保留位必须配置为 0。

6.4.3 T3 中断

T3 溢出后会使中断标志位 T3IF 置 1，如果使能位 T3IE 为 1，且全局中断和外设中断允许位为 1，将会响应 T3 中断。如果 IPEN 和 PT3 位均置 1，则 T3 为高优先级中断。

6.4.4 T3 的使用

T3 在使用时通过以下步骤进行设置:

1. 通过 OSCCTL 的 SCS<1:0>和 T3CTL 的 T3CS 位设置定时器 T3 的时钟源，通过 T3CKS<1:0>位设置定时器 T3 的分频比；
2. 将 T3CTL1 的 T3CLR 置 1，T3 计数寄存器 T3H/T3L 退出复位状态；
3. 给 T3 寄存器设置初始值；
4. 将 T3CTL 的 T3EN 位置 1 使能定时器 T3 开始计数；
5. 当 T3 计数溢出时，将会使 T3IF 置 1，如果此时 T3IE 和 AIE 位置 1 允许中断，将响应中断进入相应的中断子程序。

注：1. T3CLR 清零状态下计数器一直处于复位状态，任何的赋值操作都是无效的。

2. T3CLR 置 1 状态下计数器正常计数，此时可以对 T3L/T3H 进行赋初值操作。

1.

7 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。KF8F3110/20/30 拥有 10 路 IO 模拟输入通道；KF8F3112/22/32 最多拥有 14 路 IO 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内部参考电压 VREOUT、VDD 或施加在 ADVRIN 引脚上的电压作为转换使用的参考电压。

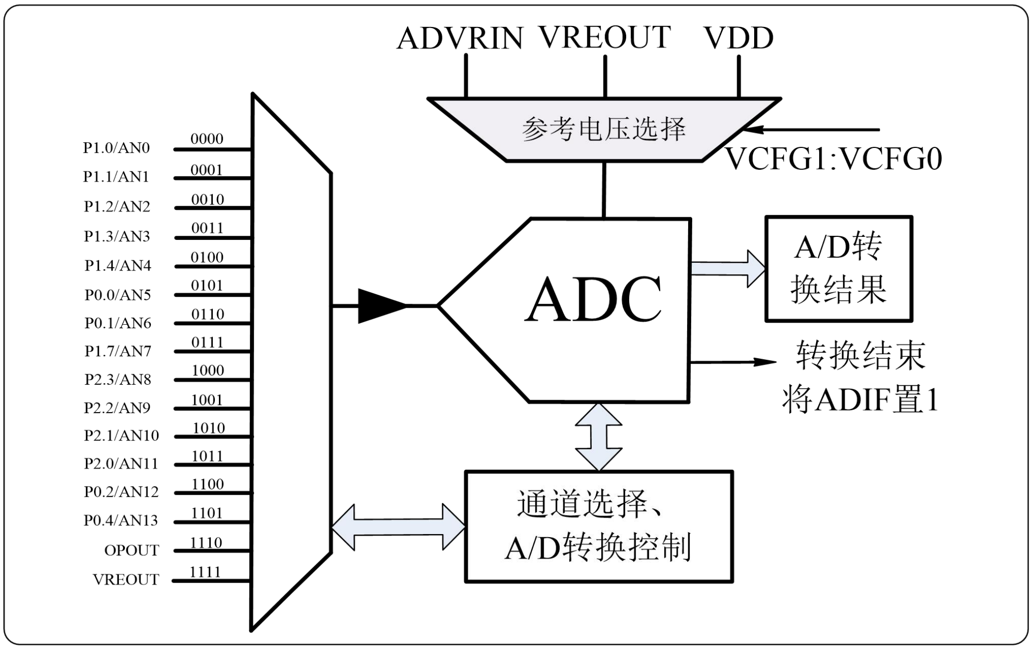


图 7.1 AD 模块结构框图

注：AN3/AN4/AN7/AN8/OPOUT 只对 KF8F3112/22/32 有效。

7.1 与 AD 相关的寄存器

表 7-1 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
0BH	INTCTL	AIE/AIEH	PUIE/AIEL	T0IE	INT0IE	P0IE	T0IF	INT0IF	P0IF
2CH	EIE1	EEIE	ADIE	INT2IE	INT1IE	C1IE	PWM2IE	T2IE	T1IE
0CH	EIF1	EEIF	ADIF	INT2IF	INT1IF	C1IF	PWM2IF	T2IF	T1IF
1FH	ADCCTL0	ADLR	T2CCR0ON	CHS3	CHS2	CHS1	CHS0	START	ADEN
3FH	ADCCTL1	ADCALEN	ADCS2	ADCS1	ADCS0	VCFG1	VCFG0	-	ADCIM
31H	ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
1DH	ANSEH	-	-	ANS13	ANS12	ANS11	ANS10	ANS9	ANS8
18H	ADSCANCTL	ADSCANEN	SMOD1	SMOD0	STIM4	STIM 3	STIM 2	STIM 1	STIM 0
23H	IP1	PEE	PADC	PINT2	PINT1	PC1	PPWM2	PT2	PT1
58H	ADCSICM	-	-	-	-	-	-	ADCSICM1	ADCSICM0
1EH	ADCDATA0H	ADC 数据寄存器 0 高字节							
3EH	ADCDATA0L	ADC 数据寄存器 0 低字节							
50H	ADCDATA1H	ADC 数据寄存器 1 高字节							
51H	ADCDATA1L	ADC 数据寄存器 1 低字节							
59H	ADCDATA2H	ADC 数据寄存器 2 高字节							
5AH	ADCDATA2L	ADC 数据寄存器 2 低字节							
5EH	ADCDATA3H	ADC 数据寄存器 3 高字节							
6AH	ADCDATA3L	ADC 数据寄存器 3 低字节							

7.1.1 AD 控制寄存器 0(ADCCTL0)

寄存器7.1: ADCCTL0: A/D控制寄存器0(地址: 1FH)

复位值 0000 0000	bit7						bit0
	ADLR	T2CCR0ON	CHS3	CHS2	CHS1	CHS0	START	ADEN
	R/W	R/W	R/W	R/W	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，ADC 采样转换结束后，硬件自动将该位清 0

0 = 禁止 T2 触发 AD 启动

CHS<3:0>: 模拟通道选择位

0000 = 通道 00(AN0)

0001 = 通道 01(AN1)

0010 = 通道 02(AN2)

0011 = 通道 03(AN3)

0100 = 通道 04(AN4)

0101 = 通道 05(AN5)

0110 = 通道 06(AN6)

0111 = 通道 07(AN7)

1000 = 通道 08(AN8)

1001 = 通道 09(AN9)

1010 = 通道 10(AN10)

1011 = 通道 11(AN11)

1100 = 通道 12(AN12)

1101 = 通道 13(AN13)

1110 =通道 14（OPOUT）

1111 = 通道 15（VREOUT）

注：AN3/AN4/AN7/AN8/OPOUT 只对 KF8F3112/22/32 有效。

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 模块工作使能位

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

图注：R=可读 W=可写 -=未用 U=未实现位

注 1：改变通道时，在开始下一次转换前需要一段采样延时(至少 1us)(假设输入阻抗为 2.5kΩ)。

7.1.2 AD 控制寄存器 1(ADCCTL1)

寄存器7.2:ADCCTL1: A/D控制寄存器1(地址: 3FH)

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

ADCALEN: AD 失调校准使能位

0 = 关闭 AD 校准

1 = 打开 AD 校准

注：1、自校准时间典型值为 5us。

2、ADEN 清零时需将 ADCALEN 位清零，关闭 AD 失调校准。

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 = 系统保留

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 = VDD 作为 ADC 参考电压

10 = ADVRIN 作为 ADC 参考电压

11 = VREOUT 作为 ADC 参考电压

ADCIM: AD 工作电流选择位

0 = A/D 工作电流较大

1 = A/D 工作电流较小（建议配置为 1，此时工作电流最小）

图注：R=可读 W=可写 -=未用 U=未实现位

注 1：在使用过程中，由于温度变化，导致 AD 采样值会有较小的误差。对 AD 精度要求较高的话，在每次采样前开启 AD 失调校准功能，开启后需要延时 5us 左右。

7.1.3 模拟/数字口选择寄存器(ANSEL/H)

寄存器7.3: ANSEH: 模拟/数字口设置寄存器(地址: 1DH)

复位值	bit7							bit0
--00 0000	-	-	ANS13	ANS12	ANS11	ANS10	ANS9	ANS8
	U	U	R/W	R/W	R/W	R/W	R/W	R/W

寄存器7.4: ANSEL: 模拟/数字口设置寄存器(地址: 31H)

复位值	bit7							bit0
0000 0000	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

A

NS<13:0>: 引脚 AN13～AN0 分别配置为模拟或数字 I/O 口的控制位

1 = 将对应引脚配置为模拟口

0 = 将对应引脚配置为数字 I/O 口或者特殊功能引脚

图注：R=可读　W=可写　-=未用　U=未实现位

7.1.4 A/D 扫描中断控制寄存器（ADCSICM）

寄存器7.5: ADCSICM: A/D扫描中断控制寄存器(地址:58H)

复位值	bit7							bit0
0000 0000	-	-	-	-	-	-	ADCSICM1	ADCSICM0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit<7:2>：系统保留，配置为 0

ADCSICM<1:0>:AD 扫描中断控制位

ADSCANEN = 0 时，此 2 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

图注：R=可读　W=可写　-=未用　U=未实现位

7.2 通道的选择

如图 7.1 所示，KF8F3110/20/30 的 ADC 模拟输入可以选择 10 路外部 IO 输入或内部参考电压 VREOUT；KF8F3112/22/32 最多可以选择 14 路来自外部的模拟信号、运放输出 OPOUT 或内部参考电压 VREOUT。通过寄存器 ADCCTL0(如寄存器 7.1 所示)进行通道的选择。

7.3 模拟输入口的配置

当选择 AN0～AN13 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 ANSEL 的某位置 1 将对应的引脚配置为模拟口，然后把寄存器 TR1/2 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。

7.4 多通道扫描模式

KF8F3110/12/20/22/30/32 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 ADCCTL0 寄存器的 CHS<3:0>位来选择模拟通道，并将转换数据存入 ADCDATA0H/L 中。

AN5、AN6、AN7、AN8 这 4 个通道为扫描模式的固定通道。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 AN5、AN6、AN7、AN8 这 4 个通道进行采样转换，先对 AN5 进行采样转换，将转换数据存入 ADCDATA0H/L 中；再对 AN6 进行采样转换，将转换数据存入 ADCDATA1H/L 中；再对 AN7 进行采样转换，将转换数据存入 ADCDATA2H/L 中；再对 AN8 进行采样转换，将转换数据存入 ADCDATA3H/L 中。

用户还可通过设置 ADCSICM 寄存器来设置中断产生的时间。

7.4.1 A/D 多通道扫描控制寄存器

寄存器7.6: ADSCANCTL: A/D多通道扫描控制寄存器(地址:18H)

复位值	bit7							bit0
0000 0000	ADSCANEN	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 AN5

01 = 扫描采样通道 AN5，AN6

10 = 扫描采样通道 AN5，AN6，AN7

11 = 扫描采样通道 AN5，AN6，AN7，AN8

注：10/11 模式只对 KF8F3112/22/32 有效。

STIM<4:0>: 自动采样时间设置位

00001 = 1TAD

00010 = 2TAD

00011 = 3TAD

……

11110=30TAD

11111=31TAD

图注：R=可读 W=可写 -=未用 U=未实现位

注 1：最小采样时间应保证 3us (假设输入阻抗为 2.5kΩ)。

7.5 A/D 转换参考电压的选择

KF8F3110/12/20/22/30/32 中 ADC 模块的参考电压可以选择 3 种，分别为电源电压(VDD)、内部参考电压 VREOUT 和外部参考电压(ADVRIN)。通过寄存器 ADCCTL1(如寄存器 7.2 所示)的 VCFG<1:0>设置参考电压。

7.5.1 VREOUT 参考电压

KF8F3110/12/20/22/30/32 内部有一个参考电压模块，使能该功能后（VREOE=1），通过引脚 P0.4/VREOUT 可输出稳定的 2V/3V/4V 参考电压。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。

参考电压模块通过参考电压控制寄存器的高两位进行控制(VRECTL<1,3>)，将 VREEN(VRECTL.1)位置 1 将打开参考电压模块，此时的 2V/3V/4V 参考电压可供芯片内部使用，再将 VREOE(VRECTL.3)位置 1 可使能内部 2V/3V/4V 参考电压输出，相应的引脚输出 2V/3V/4V 参考电压。

7.5.2 参考电压寄存器（VRECTL）

寄存器7.7: VRECTL: 参考电压寄存器1(地址:2BH)

	bit7							bit0
复位值 0000 0000	VRESEL1	VRESEL0	保留	VRECKEN	VREOE	保留	VREEN	保留
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

VRESEL<1:0>:内部参考电压 VREOUT 选择位

00=保留

01=2V

10=3V

11=4V

VRECKEN：参考电压工作时钟使能位

1 = 参考电压工作时钟使能

0 = 参考电压工作时钟禁止

注：内部参考电压模块工作时 VRECKEN 必须置 1

VREOE: 参考电压输出使能位

1 = 允许参考电压输出

0 = 禁止参考电压输出

VREEN: 参考电压使能位

1 = 使能参考电压

0 = 关闭参考电压

图注：R=可读 W=可写 -=未用 U=未实现位

注：保留位必须配置为 0。

7.6 转换时钟的选择

完成一次 A/D 转换所需要的时间为 13Tad。如寄存器 7.2 所示，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。Tad 和 Fad 分别为 A/D 转换时钟周期和频率。

为保证 A/D 转换的正确进行，所选择的 A/D 转换时钟周期(Tad)典型值应在 250ns。

7.7 输出格式

KF8F3110/12/20/22/30/32 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。如图 7.2 所示。

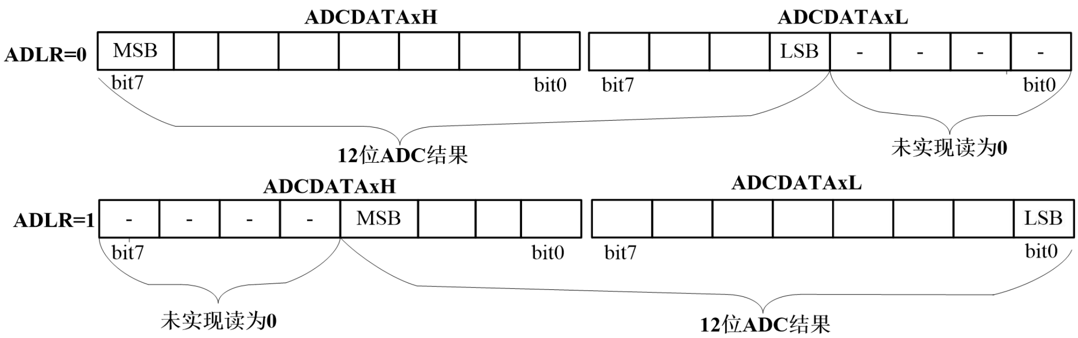


图 7.2　ADC 输出对齐方式

7.8　A/D 转换的启动和完成

先将 ADEN 位置 1，然后将 START 位(ADCCTL0<1>)置 1 即可启动 A/D 转换。当转换结束时，A/D 模块将:

1. 硬件清零 START 位
2. AD 中断标志位 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 采样转换全部结束之前，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 2Tad 的延时时间后才能开始下一次数据采集。

7.9　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。ADCDATAxH:ADCDATAxL 寄存器中的值不变。

7.10　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（ADCCTL0）；
2. 将对应的 A/D 采样输入通道设置为模拟输入模式（ANSEH/ANSEL）；
3. 如果需要使能 AD 的失调校准，将 ADCCTL1 寄存器的 ADCALEN 位置 1，清零禁止；
4. ADCCTL1 的 ADCIM 位置 1，选择工作小电流模式；
5. 选择参考电压和 A/D 采样时钟频率（ADCCTL1），ADEN 置 1 打开 A/D 转换；
6. 如果采用中断方式，使能 A/D 转换中断；
7. 等待 A/D 所需的采集时间；
8. START 置 1 启动 A/D 转换；
9. 查询 A/D 是否转换完成(START=0)或进入 A/D 中断；
10. ADCDATA0H: ADCDATA0L 存放转换结果。

8 PWM 模块

KF8F3110SB 提供 2 路 8 位 PWM，其他型号提供 4 路 16 位的 PWM 模块 PWM21/22/23/24，共用周期寄存器，占空比寄存器独立。

8.1 工作原理

PWM 模块带有 8 位 PWM 模式和 16 位 PWM 模式，通过配置 T1CTL 寄存器的 T1RLD 位，可以实现 PWM 模式的切换。PWM 模式相关信息如下表所示：

表 8-1 PWM 模式相关信息表

PWM 模式	PWM 输出	T1CTL_T1RLD	周期寄存器	占空比寄存器
8 位 PWM	PWM1	T1RLD=0	PP1	PWM1L
	PWM2	T1RLD=0	PP2	PWM2L
16 位 PWM	PWM21	T1RLD=1	<PP2:PP1>	<PWM2L:PWM1L>
	PWM22	T1RLD=1	<PP2:PP1>	<PWM22:PWM12>
	PWM23	T1RLD=1	<PP2:PP1>	<PWM23:PWM13>
	PWM24	T1RLD=1	<PP2:PP1>	<PWM24:PWM14>

8.1.1 16 位 PWM 模式

图 8.1 为 PWM21 的逻辑框图。PWM22、PWM23 和 PWM24 原理和 PWM21 完全一致。

<PP2:PP1>为 PWM21 的周期寄存器，<PWM2L:PWM1L>为占空比设置寄存器；使用 PWM 时需要将定时器 1 分配给 PWM 做定时用；（8 位 PWM 模式下，T1L、T1IE 和 T1IF 分配给 PWM1，T1H 分配给 PWM2）。

启动 PWM21 后，当<T1H:T1L>计数值和<PP2:PP1>相等时，PWM21 输出引脚被置 1，此时<T1H:T1L>被清 0，重新开始计数，当<T1H:T1L>的计数值和<PWM2L:PWM1L>相等时，PWM21 输出引脚清 0 (如图 8.2 所示)。改变<PP2:PP1>和<PWM2L:PWM1L>的值可产生不同的 PWM21 周期和 PWM21 占空比。

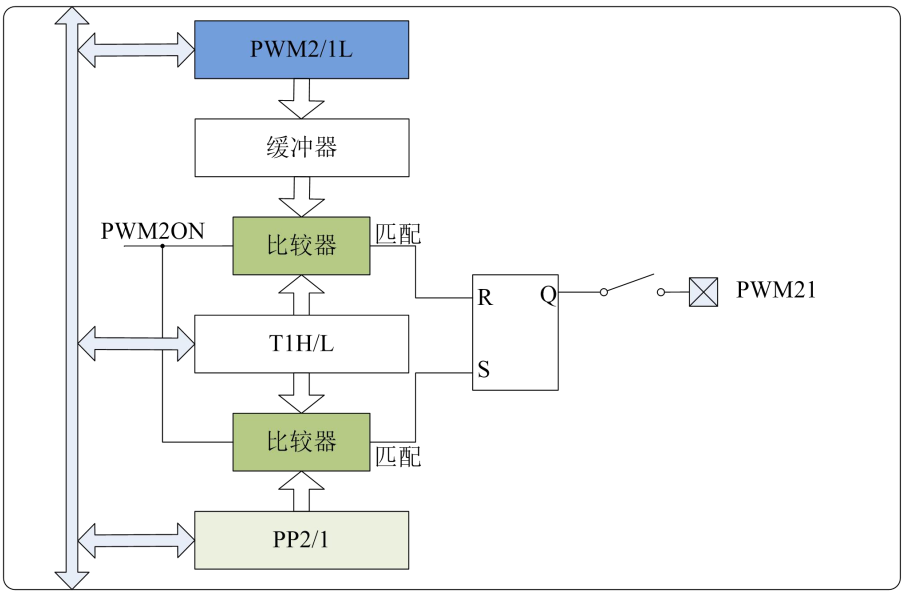


图 8.1 PWM21 逻辑框图

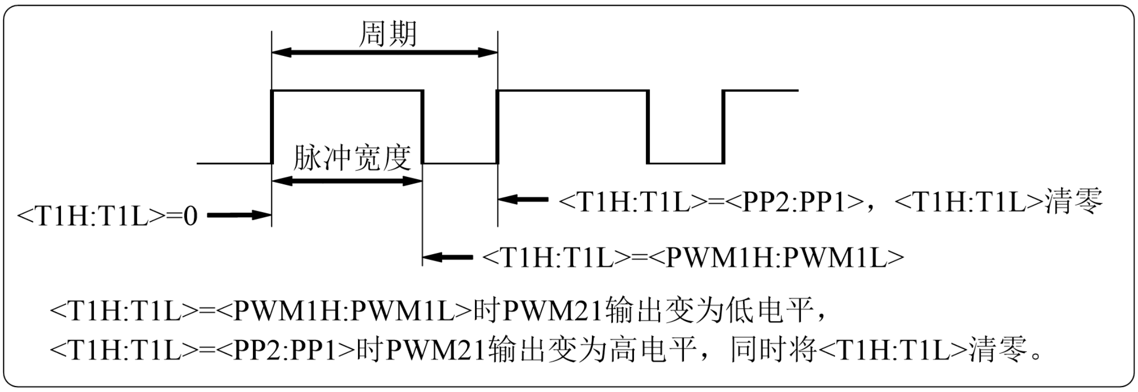


图 8.2 PWM21 输出波形图

8.1.2 8 位 PWM 模式

图 8.3 为 PWM1 的逻辑框图。PWM2 原理和 PWM1 完全一致。

PP1 为 PWM1 的周期寄存器， PWM1L 为占空比设置寄存器；使用 PWM 时需要将定时器 1 分配给 PWM 做定时用； 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，PWM1 输出引脚被置 1，此时 T1L 被清 0，重新开始计数，当 T1L 的计数值和 PWM1L 相等时，PWM1 输出引脚清 0 (如图 8.4 所示)。改变 PP1 和 PWM1L 的值可产生不同的 PWM1 周期和 PWM1 占空比。

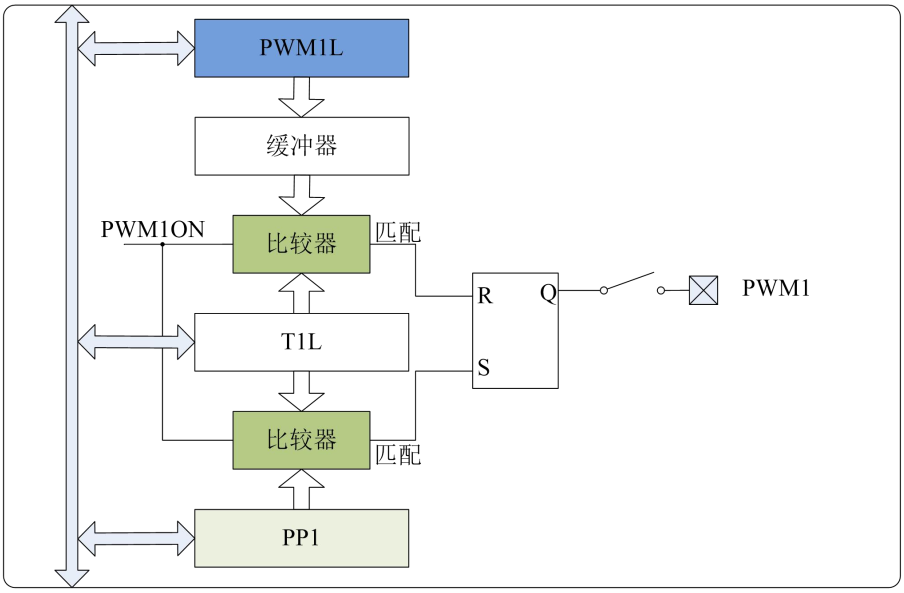
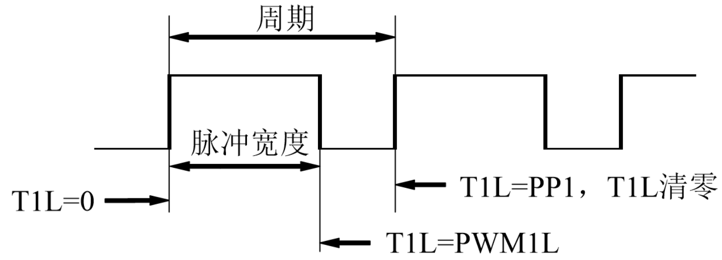


图 8.3 PWM1 逻辑框图



T1L=PWM1L时PWM1输出变为低电平，T1L=PP1时PWM1输出变为高电平，同时将T1L清零。

图 8.4 PWM1 输出波形图

8.2 PWM 相关的寄存器

表 8-2 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	PWM1 占空比设置寄存器							
16H	PP1	PWM1 周期寄存器							
32H	PP2	PWM2 周期寄存器							
33H	PWM2L	PWM2 占空比设置寄存器							
10H	T1CTL	T1RLD	T1GC	T1CKS1	T1CKS0	T1OSCEN	$\overline{T1SY}$	T1CS	T1ON
15H	PWMCTL	PWM24ON	PWM23ON	PWM22ON	PWM2ON	-	-	-	PWM1ON
156H	PINSET	SSCIPIN	SDOPIN	SSPIN	UARTPIN	PWMPIN	-	-	-
65H	PWM22	PWM22 占空比设置寄存器高 8 位							
62H	PWM12	PWM22 占空比设置寄存器低 8 位							
66H	PWM23	PWM23 占空比设置寄存器高 8 位							
63H	PWM13	PWM23 占空比设置寄存器低 8 位							
69H	PWM24	PWM24 占空比设置寄存器高 8 位							
68H	PWM14	PWM24 占空比设置寄存器低 8 位							

8.2.1 PWM1/2 控制寄存器

寄存器8.1：PWMCTL：PWM启动控制寄存器（地址：15H）

复位值 0000 0000	bit7							bit0
	PWM24ON	PWM23ON	PWM22ON	PWM2ON	保留	保留	保留	PWM1ON
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWM24ON: PWM24 启动控制位
1 = 启动 PWM24
0 = 禁止 PWM24

PWM23ON: PWM23 启动控制位
1 = 启动 PWM23
0 = 禁止 PWM23

PWM22ON: PWM22 启动控制位
1 = 启动 PWM22
0 = 禁止 PWM22

PWM2ON: PWM2 或 PWM21 启动控制位
1 = 启动 PWM2 或 PWM21
0 = 禁止 PWM2 或 PWM21

PWM1ON: PWM1 启动控制位
1 = 启动 PWM1
0 = 禁止 PWM1

图注：R=可读 W=可写 -=未用 U=未实现位

注：保留位必须配置为 0

8.2.2 PWM 周期

PWM 周期通过<PP2:PP1>进行设置，<PP2:PP1>由 2 个 8 位的寄存器组成，其值可设置为 0～65535（8 位 PWM 模式下，只是用 PP2 或 PP1 寄存器，其值可设置为 0～255）。

以 PWM21 为例，PWM 周期通过式 8.1 进行计算。

★式 8.1： $\text{PWM周期} = (<PP2:PP1>+1)\cdot Tpwm$

注： 当T1CLKEN=0时 $Tpwm = 4\cdot Tsys\cdot(\text{T1预分频比})$

当T1CLKEN=1时 $Tpwm = T_{INTHF}\cdot(\text{T1预分频比})$

寄存器8.2：PP1：PWM1周期控制寄存器(地址:16H)

复位值 1111 1111	bit7							bit0
	PP17	PP16	PP15	PP14	PP13	PP12	PP11	PP10
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PP2：PWM2周期控制寄存器(地址:32H)

复位值 1111 1111	bit7							bit0
	PP27	PP26	PP25	PP24	PP23	PP22	PP21	PP20
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

图注：R=可读 W=可写 -=未用 U=未实现位

8.2.3 PWM 占空比

PWM21 占空比通过<PWM2L:PWM1L>进行设置；

PWM22 占空比通过<PWM22:PWM12>进行设置；

PWM23 占空比通过<PWM23:PWM13>进行设置；

PWM24 占空比通过<PWM24:PWM14>进行设置；

以 PWM21 为例，脉冲宽度和占空比通过式 8.2 和式 8.3 计算:

★ 式8.2: $$\text{脉冲宽度} = \langle PWM2L:PWM1L\rangle \cdot Tpwm$$

注：当T1CLKEN = 0时，$Tpwm = 4 \cdot Tsys \cdot$(T1预分频比)
　　当T1CLKEN = 1时，$Tpwm = T_{INTHF} \cdot$(T1预分频比)

★ 式8.3: $$\text{PWM占空比}=\frac{\text{脉冲宽度}}{\text{PWM周期}}=\frac{\langle PWM2L:PWM1L\rangle}{\langle PP2:PP1\rangle+1}$$

寄存器**8.3:** PWM1L：PWM1占空比设置寄存器(地址：13H)

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

PWM2L：PWM2占空比设置寄存器(地址：33H)

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

图注：R=可读　W=可写　-=未用　U=未实现位

8.3 PWM 分辨率

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。分辨率的计算公式如式 8.4 所示。

★ 式8.4: $$\text{分辨率}=\frac{Log[(\langle PP2:PP1\rangle+1)]}{log2}\ \text{位}$$

8.4 PWM 中断

8 位 PWM 模式下，PWM2 有一个专门的中断使能位 PWM2IE 和中断标志位 PWM2IF，而 PWM1 和定时器 T1 共用中断使能位 T1IE 和中断标志位 T1IF。

16 位 PWM 模式下，PWM2x 的中断使能位和中断标志位共用，使用 PWM2IE/PWM2IF 或 T1IE/T1IF 均可。

当<T1H:T1L>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<T1H:T1L>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。

8.5 PWM 输出引脚

2 路 8 位 PWM（PWM1 和 PWM2）的输出引脚可以通过 PINSET 寄存器（156H）的 PWMPIN 位选择：PWMPIN=0 时，P0.2 作为 PWM1 的输出引脚，P1.0 作为 PWM2 的输出引脚；PWMPIN=1 时，P0.0 作为 PWM1 的输出引脚，P0.1 作为 PWM2 的输出引脚。

使用 4 路 16 位 PWM（PWM21/22/23/24）时，需要将 PINSET 寄存器的 PWMPIN 位清零，T1CTL 寄存器的 T1RLD 位置 1；PWM21 位于 P0.2 引脚，PWM22 位于 P01.0 引脚，PWM23

位于 P1.1 引脚，PWM24 位于 P01.2 引脚。

表 8-3 16 位 PWM 相关信息

16 位 PWM	引脚位置	使能位	周期寄存器	占空比寄存器
PWM21	P0.2	PWM2ON	PP2:PP1	PWM2L:PWM1L
PWM22	P1.0	PWM22ON	PP2:PP1	PWM22:PWM12
PWM23	P1.1	PWM23ON	PP2:PP1	PWM23:PWM13
PWM24	P1.2	PWM24ON	PP2:PP1	PWM24:PWM14

8.6 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。

8.7 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。

8.8 PWM 使用方法

PWM 工作的设置应按照以下步骤:

1、 通过 PWMPIN 设置选择 PWM 输出引脚。

2、 将 PWM 输出引脚对应的 I/O 口方向控制位 TRx 置 1，禁止 PWM 引脚的输出驱动器。

2、 赋 PP2/PP1 寄存器的初值以设置 PWM 周期。

3、 赋 PWM2L/PWM1L(PWM22/PWM12、PWM23/PWM13、PWM24/PWM14)寄存器的初值以设置 PWM 的占空比。

4、 配置并启动定时器/计数器 T1:

•配置 T1CS 位和 T1CLKEN 位选择 T1 的计数时钟（一般选择内部时钟）；

• 配置 T1CTL 寄存器的 T1CKS1 和 T1CKS0 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。

5、 将 PWMCTL 寄存器的 PWMxON 位置 1 以启动对应 PWM 输出。

6、 将 TRx 位清 0 使能 PWM 引脚的输出驱动器。

9 CCP(捕捉/比较/PWM5)模块

在 CCP 模块中，PWM5 为 16 位模式，T1/T2 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 CCRH 和 CCRL 分别作为数据寄存器的高 8 位和低 8 寄存器；捕捉和比较的 16 位数据寄存器 CCRH 和 CCRL 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。

捕捉/比较模式与 PWM5 模式可同时使用，捕捉/比较模式可通过 CCPCTL 寄存器配置，PWM5 的相关功能可通过 PWM5CTLx(x=0,1,2)等 PWM5 相关控制寄存器配置。

9.1 CCP 相关寄存器

在捕捉模式下时，当对应的 CCP（P1.1 或 P2.0）引脚发生事件时，CCRH:CCRL 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 CCRH:CCRL 这对寄存器。

在比较模式下，16 位 CCRH:CCRL 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。

对于 PWM5 模式，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 9-1 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
52H	PP5L	PWM5 周期寄存器低 8 位							
42H	PP5H	PWM5 周期寄存器高 8 位							
151H	CCRL	捕捉/比较寄存器低 8 位							
154H	CCRH	捕捉/比较寄存器高 8 位							
55H	PWM5L0	PWM5 占空比寄存器低 8 位							
43H	PWM5H0	PWM5 占空比寄存器高 8 位							
56H	PWM5L1	PWM5 低 8 位占空比缓冲寄存器（与用户无关）							
44H	PWM5H1	PWM5 高 8 位占空比缓冲寄存器（与用户无关）							
57H	PWM5CTL0	P5CH1MOD1	P5CH1MOD0	-	-	P5MOD3	P5MOD2	P5MOD1	P5MOD0
5BH	PWM5CTL1	P5RSEN	P5DC6	P5DC5	P5DC4	P5DC3	P5DC2	P5DC1	P5DC0
11CH	PWM5CTL2	-	-	PFUSES			UDEVT1	UDEVT0	UDEN
152H	CCPCTL	-	-	-	CCPPIN	CCMOD3	CCMOD2	CCMOD1	CCMOD0
5CH	P5ASCTL	P5ASE	P5ASS2	P5ASS1	P5ASS0	P5SSAC1	P5SSAC0	P5SSBD1	P5SSBD0
5DH	PSTRCTL	-	-	-	STRSYNC	STREND	STRENC	STRENB	STRENA
11DH	PWM5PC	-	-	-	-	-	-	PCA	PCB
15DH	PWM5FC	-	-	-	-	-	-	FCA	FCB
10EH	PWM5OC	-	-	-	-	-	-	OCA	OCB

9.1.1 捕捉/比较控制寄存器（CCPCTL）

寄存器9.1: CCPCTL: 捕捉/比较控制寄存器(地址: 152H)

复位值 ---0 0000	bit7							bit0
	-	-	-	CCPPIN	CCMOD3	CCMOD2	CCMOD1	CCMOD0
	U	U	U	R/W	R/W	R/W	R/W	R/W

CCPPIN: CCP 通道选择位

0 = P2.0 作为 CCP 通道

1 = P1.1 作为 CCP 通道

CCMOD<3:0>: 捕捉/比较模式选择位

0000 = 捕捉/比较功能关闭

0001 = 未使用（保留）

0010 = 比较模式，匹配时输出电平翻转（CCPIF 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，初始化 CCP/CCP(1)引脚为低电平，比较匹配时 CCP/CCP(1)引脚输出高电平（CCPIF 置 1）

1001 = 比较模式，初始化 CCP/CCP(1)引脚为高电平，比较匹配时 CCP/CCP(1)引脚输出低电平（CCPIF 置 1）

1010 = 比较模式，比较匹配时产生软件中断（CCPIF 置 1，CCP/CCP(1)引脚不受影响）

1011 = 比较模式，比较匹配时触发特殊事件（CCPIF 位置 1，CCP 复位 T1，如果 ADC 使能将启动 AD 转换。）

其它= 未使用（保留）

图注：R=可读 W=可写 -=未用 U=未实现位

9.2 捕捉模式

在捕捉模式下，当对应的CCP/CCP(1)引脚发生事件时，CCRH:CCRL这对寄存器捕捉T1H和T1L 寄存器的16位值，原理框图如下图9.1所示：

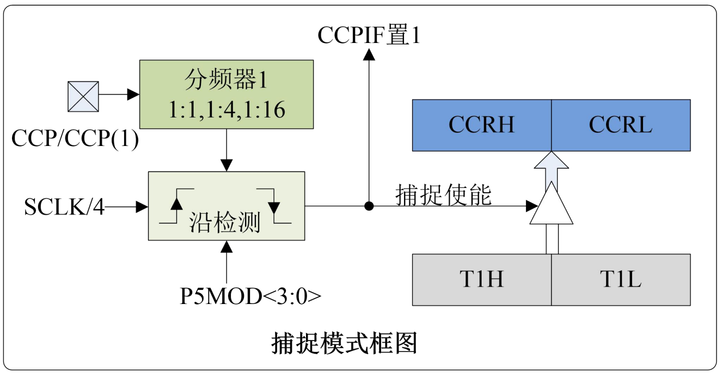


图9.1 捕捉模式原理框图

触发捕捉的事件可被定义为以下四者之一，并且由 CCPCTL 寄存器中的模式选择位CCMOD<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，通过 CCPPIN 选择相应的 IO 口，同时将对应的 IO 口方向控制寄存器控制位置 1，将相应的 CCP/CCP（1）引脚配置为输入。

当一个捕捉发生时，硬件自动将中断请求标志位EIF2寄存器中的CCPIF置1，它必须用软件清零。注意，如果在CCRH:CCRL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 寄存器中的 CCPIE 中断允许位为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 寄存器中的中断标志位 CCPIF。

CCPCTL 寄存器中的 CCMOD<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 CCPCTL 寄存器清零关闭该模块。

注：
1、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。
2、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。
3、T1CLKEN必须配置为0。

9.3 比较模式

在比较模式下，16 位CCRH:CCRL寄存器的值将不断与T1数据寄存器的值相比较。当两者匹配时， CCP模块可能会出现以下几种情况：

- CCP/CCP(1)的输出电平翻转
- CCP/CCP(1)输出高电平
- CCP/CCP(1)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

CCP 引脚的动作取决于 CCPCTL 寄存器中 CCMOD<3:0>控制位的值。所有比较模式都会产生中断。原理图如图 9.2 所示：

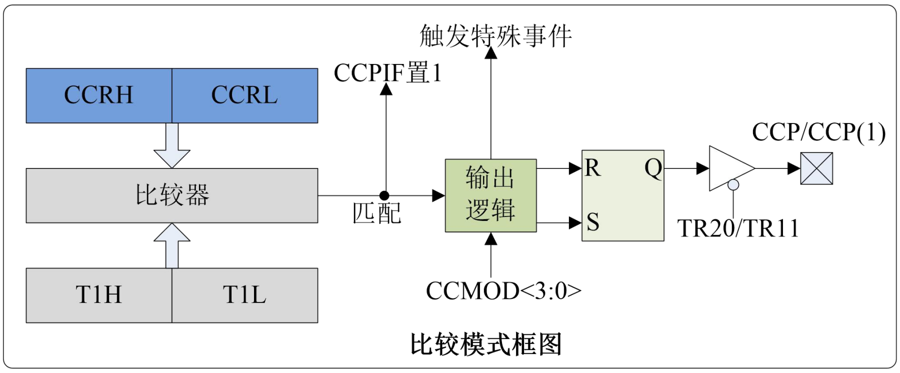


图9.2 比较模式原理框图

通过CCPPIN选择相应的IO口，同时必须将相应的TRx 位清零来将CCP/CCP(1)引脚配置为输出。

在比较模式下，T1必须运行在定时器模式或同步计数器模式。在异步计数器模式下，可能无法进行比较操作。

1) 当选择输出电平翻转模式（CCMOD<3:0> = 0010）时，比较匹配时，CCP/CCP1引脚的输出电平翻转，并将CCPIF置1。
2) 当选择普通比较模式时（CCMOD<3:0> = 1000或1001）时，比较匹配时，CCP/CCP1引脚输出高电平或者低电平，并且将CCPIF置1。
3) 当选择了软件中断触发模式时（CCMOD<3:0> = 1010)时，比较匹配时，将CCPIF置1，但是CCP模块不会控制CCP/CCP1引脚。
4) 当选择了特殊事件触发模式（CCMOD <3:0> = 1011）时，比较匹配时，CCP会立即产生特殊事件触发输出，将CCPIF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CCRH:CCRL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。

注：比较模式T1CLKEN必须配置为0。

9.4 PWM5 模式

PWM5 为带有死区控制功能的增强型 PWM 模块。如图 9.3 所示，PWM5 可在 4 个不同的引脚输出 PWM 信号，分辨率最高 16 位。

PWM5 带有 4 个引脚分别为 P5A、P5B、P5C 和 P5D，可工作在 4 种输出模式下：单输出模式、半桥输出模式、全桥正向输出模式和全桥反向输出模式。用户可通过寄存器 PWM5CTL0 中的 P5CH1MOD<1:0>位选择 4 种输出模式之一，通过 P5MOD<3:0>位设置 PWM5 各引脚的有效电平（可设置为高电平有效和低电平有效，此外 PWM5PC 寄存器也可以控制其极性）。单输出模式下，可通过 PSTRCTL 寄存器选择引脚作为 PWM 引脚还是通用 IO 引脚。

通过寄存器 PWM5OC 中的 OCA/ OCB 位可以选择 PWM5 作为 PWM 输出还是强制输出。作为 PWM 输出时，通过 PWM5PC 中的 PCA/ PCB 位可以设置 PWM5 输出引脚的极性。作为强制输出时，通过 PWM5FC 寄存器的 FCA/ FCB 位可以设置 PWM5 引脚作为强制输出时的电平。

注：极性控制除了 PWM5PC 外，还可以通过 PWM5CTL0 寄存器的 P5MOD<3:0>控制，两者同或共同影响产生最后的输出，如表 9-3 PWM5 输出极性控制所示。

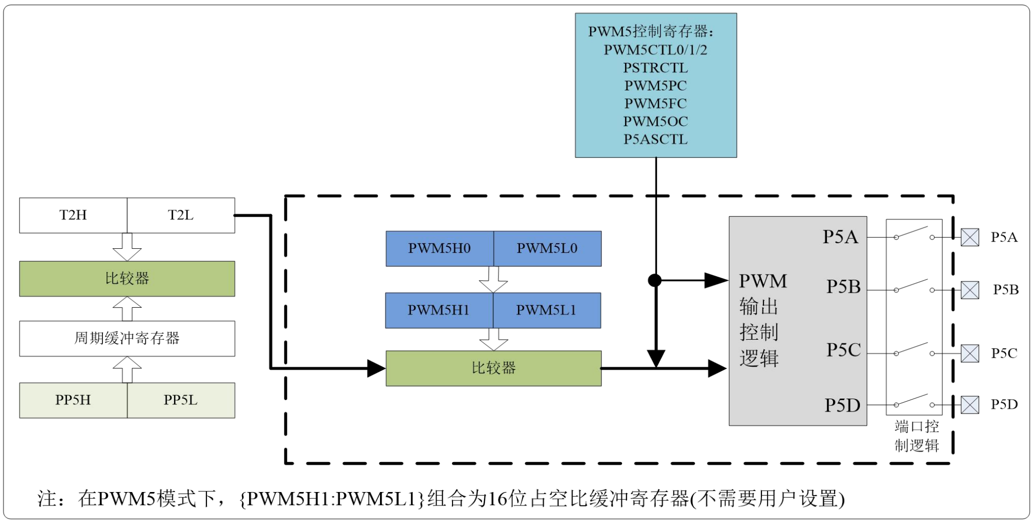


图 9.3 PWM5 原理框图

占空比寄存器	PWM5H0	PWM5L0

图 9.4 PWM5 对应的占空比设置寄存器

表 9-2 PWM5 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 PATRCTL 寄存器中的 STREN<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 1：各种输出模式的详细介绍请参考本节相应部分。

表 9-3　PWM5 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 1：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。

9.4.1　PWM5 相关控制寄存器

9.4.1.1　PWM5 控制寄存器 0（PWM5CTL0）

寄存器9.2: PWM5CTL0: PWM5控制寄存器0(地址: 57H)

	bit7							bit0
复位值 00-- 0000	P5CH1MOD1	P5CH1MOD0	-	-	P5MOD3	P5MOD2	P5MOD1	P5MOD0
	R/W	R/W	U	U	R/W	R/W	R/W	R/W

P5CH1MOD<1:0>: PWM5 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 STREN<A:D>各位置 1，分别将 P5A、P5B、P5C 和 P5D 配置为 PWM 输出，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。

P5MOD<3:0>: PWM5模块的模式选择位

0000 =捕捉/比较/PWM 关闭
1100 = PWM 模式：P5A/P5C 为高电平有效，P5B/P5D 也为高电平有效。
1101 = PWM 模式：P5A/P5C 为高电平有效，P5B/P5D 为低电平有效。
1110 = PWM 模式：P5A/P5C 为低电平有效，P5B/P5D 为高电平有效。
1111 = PWM 模式：P5A/P5C 为低电平有效，P5B/P5D 也为低电平有效 。
其他 = 未使用（保留）

图注：R=可读　W=可写　-=未用　U=未实现位

注 1：极性控制由P5A/P5B和P5MOD<3:0>同或影响，详细见表9-3所示。

9.4.1.2 PWM5 极性控制寄存器（PWM5PC）

寄存器**9.3:** PWM5PC: PWM5极性控制寄存器(地址:11DH)

复位值	bit7							bit0
---- --00	-	-	-	-	-	-	PCA	PCB
	U	U		U	U	U	R/W	R/W

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 1：极性控制由P5A/P5B和P5MOD<3:0>同或影响，详细见表9-3所示。

图注：R=可读 W=可写 -=未用 U=未实现位

9.4.1.3 PWM5 强制控制寄存器(PWM5FC)

寄存器**9.4:** PWM5FC: PWM5强制控制寄存器(地址:15DH)

复位值	bit7							bit0
--00 0000	-	-	-	-	-	-	FCA	FCB
	U	U	R/W	R/W	R/W	R/W	R/W	R/W

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

图注：R=可读 W=可写 -=未用 U=未实现位

9.4.1.4 PWM5 输出控制寄存器（PWM5OC）

寄存器**9.5:** PWM5OC: PWM5输出控制寄存器(地址:10EH)

复位值	bit7							bit0
---- --00	-	-	-	-	-	-	OCA	OCB
	U	U	U	U	U	U	R/W	R/W

OCA: 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: 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

图注：R=可读　W=可写　-=未用　U=未实现位

9.4.1.5 PWM5 控制寄存器 1(PWM5CTL1)

寄存器9.6: PWM5CTL1: PWM5控制寄存器1(地址: 5BH)

复位值 0000 0000	bit7							bit0
	P5RSEN	P5DC6	P5DC5	P5DC4	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。

图注：R=可读　W=可写　-=未用　U=未实现位

9.4.1.6 PWM5 自动关闭控制寄存器（P5ASCTL）

寄存器9.7: P5ASCTL: PWM5自动关闭控制寄存器(地址: 5CH)

复位值 0000 0000	bit7							bit0
	P5ASE	P5ASS2	P5ASS1	P5ASS0	P5SSAC1	P5SSAC0	P5SSBD1	P5SSBD0
	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

图注：R=可读　W=可写　-=未用　U=未实现位

9.4.1.7 PWM5 脉冲转向控制寄存器（PSTRCTL）

寄存器9.8: PSTRCTL: PWM5脉冲转向控制寄存器0(地址: 5DH)

复位值	bit7							bit0
---0 0001	-	-	-	STRSYNC	STREND	STRENC	STRENB	STRENA
	U	U	U	R/W	R/W	R/W	R/W	R/W

STRSYNC: 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

STREND: 转向使能位 D

1 = P5D 引脚输出 PWM 波形，其极性由 P5MOD<3:0>控制

0 = P5D 引脚被分配为端口引脚

STRENC: 转向使能位 C

1 = P5C 引脚输出 PWM 波形，其极性由 P5MOD<3:0>控制

0 = P5C 引脚被分配为端口引脚

STRENB: 转向使能位 B

1 = P5B 引脚输出 PWM 波形，其极性由 P5MOD<3:0>控制

0 = P5B 引脚被分配为端口引脚

STRENA: 转向使能位 A

1 = P5A 引脚输出 PWM 波形，其极性由 P5MOD<3:0>控制

0 = P5A 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

9.4.1.8 PWM5 控制寄存器 2（PWM5CTL2）

寄存器9.9: PWM5CTL2: PWM5控制寄存器2(地址:11CH)

复位值	bit7							bit0
--1- -000	-	-	PFUSES	-	-	UDEVT1	UDEVT0	UDEN
	U	U	R/W	U	U	R/W	R/W	R/W

P

FUSES: 由 PFUSES 位控制 PWM 初始输出状态

0 = 不由配置字控制 PWM 初始输出所对应的 IO 口

1 = 由配置字中的 PWMPIN、HPOL 和 LPOL 位控制初始输出所对应的 IO 口

UDEVT1: 更新事件控制 1（在 UDEN=1 时有效）

0 = 当定时器 T2 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 T2 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

图注：R=可读 W=可写 -=未用 U=未实现位

9.4.2 PWM5 的周期、占空比及分辨率

9.4.2.1 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，在边沿对齐模式下和中心对齐模式下，周期的计算方式不同，中心对齐模式下得到的周期是边沿对齐模式下周期寄存器设置值的两倍。PWM5 边沿对齐和中心对齐的周期分别通过式 9.1 和式 9.2 进行计算。

★ 式9.1: $\text{PWM周期}=(\langle PP5H:PP5L\rangle+1)\cdot T_{INTHF}\cdot(\text{T2预分频比})$

★ 式9.2: $\text{PWM周期}=(\langle PP5H:PP5L\rangle)\cdot T_{INTHF}\cdot(\text{T2预分频比})\cdot 2$

注：T_{INTHF} 内部高频振荡器时钟周期。

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 T2 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，T2 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 1：当T2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。

9.4.2.2 PWM5 占空比

PWM5 模块有 1 个 16 位占空比寄存器，通过寄存器<PWM5H0:PWM5L0>进行设置，PWM5H0 为占空比的高 8 位，PWM5L0 为低 8 位。在不同模式下占空比计算方式不一致。在边沿对齐模式下，脉冲宽度和占空比通过式 9.3 和式 9.4 计算；在中心对齐模式下，脉冲宽度和占空比通过式 9.5 和式 9.6 计算。

★ 式9.3: $\text{脉冲宽度}=(PWM5H0:PWM5L0)\cdot T_{INTHF}\cdot(\text{T2预分频比})$

★ 式9.4: $\text{占空比}=\dfrac{\text{脉冲宽度}}{\text{PWM周期}}=\dfrac{PWM5H0:PWM5L0}{PP5+1}$

★ 式9.5: $\text{脉冲宽度}=(PWM5H0:PWM5L0)\cdot T_{INTHF}\cdot(\text{T2预分频比})\cdot 2$

★ 式9.6: $\text{占空比}=\dfrac{\text{脉冲宽度}}{\text{PWM周期}}=\dfrac{PWM5H0:PWM5L0}{PP5}$

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）

- 中心对齐模式：

  0%: 占空比寄存器为 0

  100%:占空比寄存器为大于等于 PP5（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 T2 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，T2 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 T2 启动时占空比寄存器立即更新；

(2) 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。

9.4.2.3 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式 9.7 所示。

★ 式9.7: $分辨率=\dfrac{Log[4(<PP5H:PP5L>+1)]}{log2}$ 位

9.4.2.4 PWM5 中断

当 PWM5 调制输出满一个周期时，将中断标志位 T2IF 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 PT2 位均置 1，则为高优先级中断。

9.4.3 边沿对齐 PWM 信号

当 T2CTL1 控制寄存器中的 T2MOD<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<PWM5H0:PWM5L0>决定。在周期开始时（即计数器 T2 等于 0 时）PWM 驱动为高电平，当计数器 T2 与<PWM5H0:PWM5L0>占空比寄存器发生匹配时，PWM 输出低电平。

图 9.5 所示为边沿对齐模式下的 PWM 输出示意图。当 PWM（P5A）输出为高有效时，周期开始输出高电平，占空比匹配时电平变低；当 PWM（P5A）输出低有效时，周期开始输出低电平，占空比匹配电平变高。如图 9.5 所示 PWM 输出 1 为 P5A 输出在高有效时输出。而 P5B 输出则相反。

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。

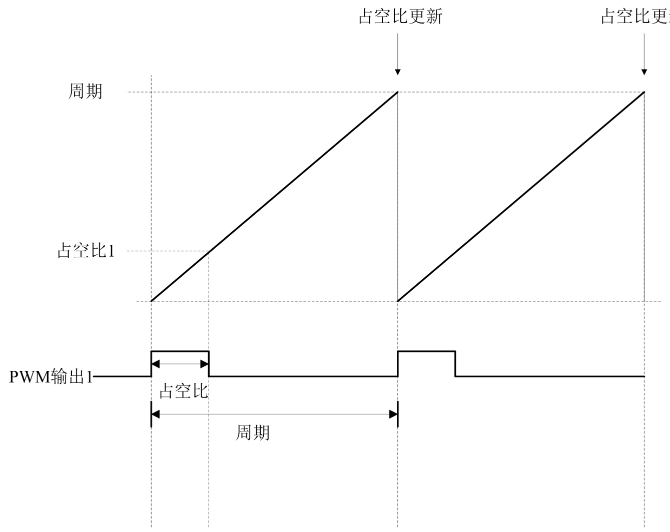


图 9.5 边沿对齐 PWM

9.4.4 中心对齐 PWM 信号

当 T2CTL1 控制寄存器中的 T2MOD<1:0>计数模式选择位设置成 01,10,11 时，产生的 PWM 信号为中心对齐的 PWM 信号。在该模式下，PWM 信号的周期由<PP5H:PP5L>周期寄存器决定，此时，周期寄存器中的值表示的是周期的一半。而占空比是由 16 位的寄存器<PWM5H0:PWM5L0>决定的。在占空比与 T2 寄存器匹配时，PWM 输出翻转。

图 9.6 所示为中心对齐模式下的 PWM 输出示意图。当 PWM 输出（P5A）为高有效时，周期开始输出高电平，占空比匹配时电平翻转；当 PWM（P5A）输出低有效时，周期开始输出低电平，占空比匹配时电平翻转。下图所示 PWM 输出 1 所示为 P5A 输出在高有效时输出。而 P5B 输出则相反。

如果占空比值为 0，则在整个 PWM 输出均为低电平，若占空比值比周期寄存器的值大或者等于周期，则在整个 PWM 输出均为高电平。

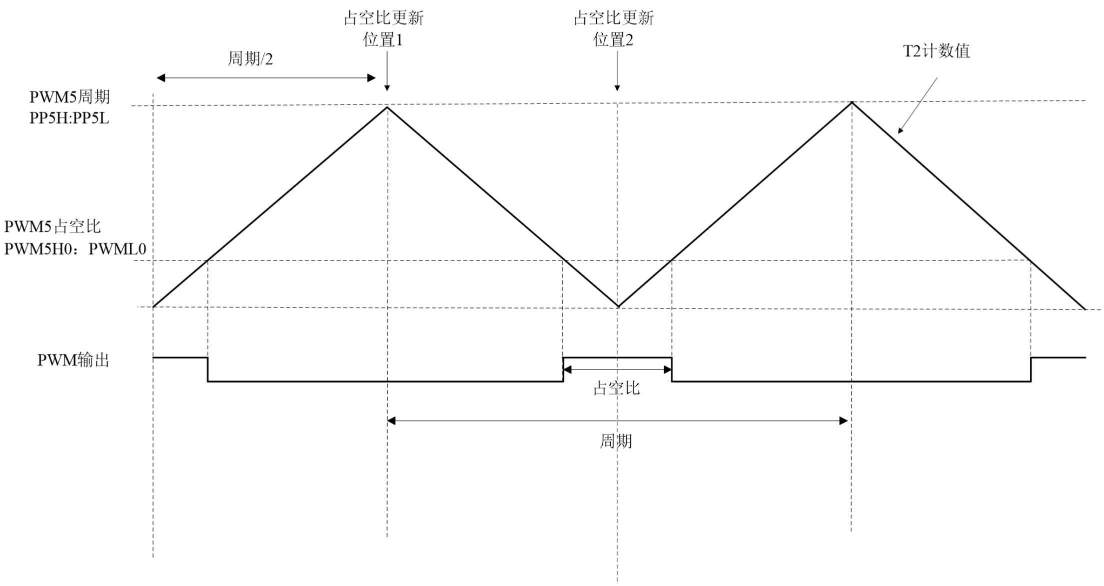


图 9.6 中心对齐 PWM

9.4.5 PWM 信号产生和中断

通过设置 T2CTL1 控制寄存器中的 T2MOD<1:0>可以选择 T2 的计数方式，从而产生边沿对齐和中心对齐的 PWM 信号。当 T2 向上计数时，产生边沿对齐的 PWM 信号；当 T2 向上向下计数时，产生中心对齐的 PWM 信号。

通过设置 T2CTL1 寄存器中的 T2MOD<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 T2MOD<1:0>=00 或者 01 时，是上溢时产生中断标志，就是在计数器 T2 向上计数到周期值的时候产生一个中断信号。

当 T2MOD<1:0>=10 时，是下溢产生中断标志，也就是计数器 T2 向下递减为 0 的时候产生一个中断信号。

当 T2MOD<1:0>=11 时，是上溢-下溢中断，在这种状态下在<PP5H:PP5L>和计数器 T2 相等以及计数器 T2 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。

9.4.6 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。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 PSTRCTL，使能或禁止 PWM5 的 4 路输出或 IO 引脚，PWM5 使用单输出模式时，最多可同时在 4 个引脚输出 PWM 信号。当 4 个引脚同时输出 PWM 信号时，输出的 PWM 信号完全相同；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 输出的示例如图 9.7 所示。

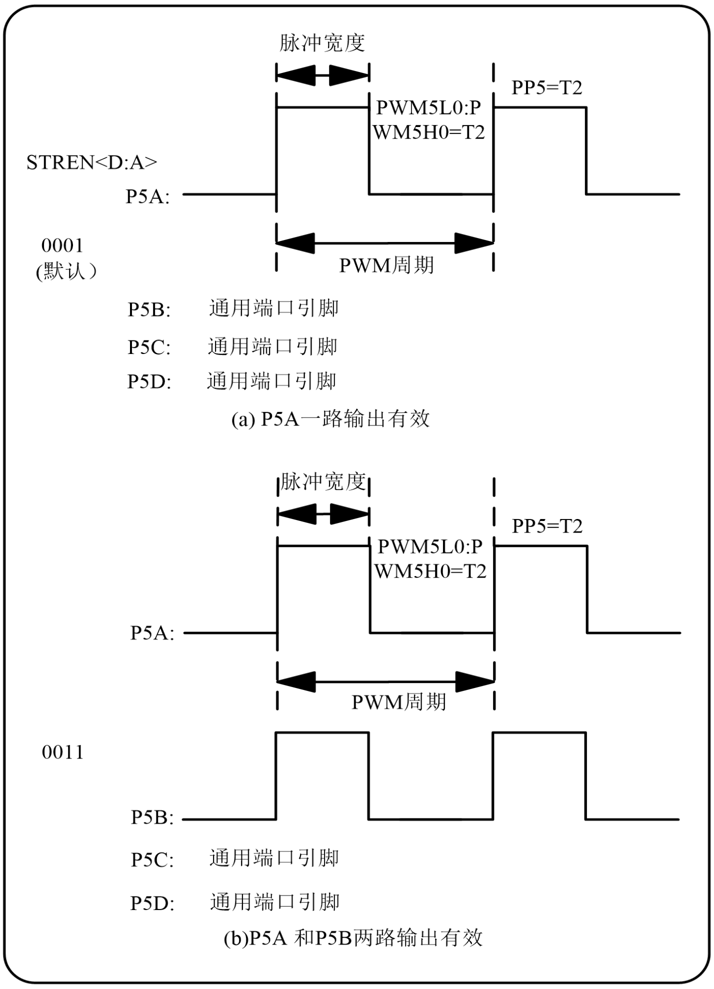


图 9.7 PWM5 的输出示例

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 PSTRCTL 中的 STRSYNC 位进行设置引脚输出切换时是否与指令同步。如图 9.8 所示，STRSYNC=1 时，对应引脚 P5x 输出的 PWM 信号在 STRENx 置一后且的 PWM5 输出信号周期结束时输出；STRSYNC=0 时，对应引脚 P5x 输出的 PWM 信号在 STRENx 置 1 后立即输出。

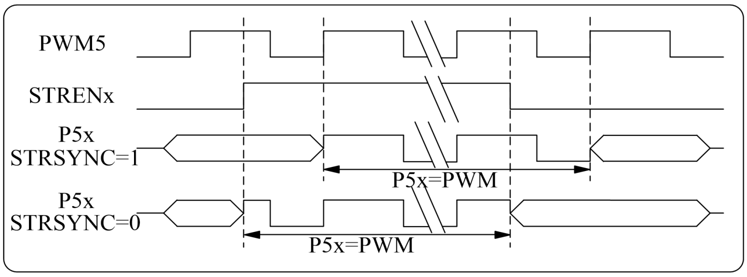


图 9.8 单输出模式 PWM 输出引脚切换

9.4.7 半桥输出模式

通过将寄存器 PWM5CTL0 的 P5CH1MOD<3:0>位设置为 10 可将通道 1 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。

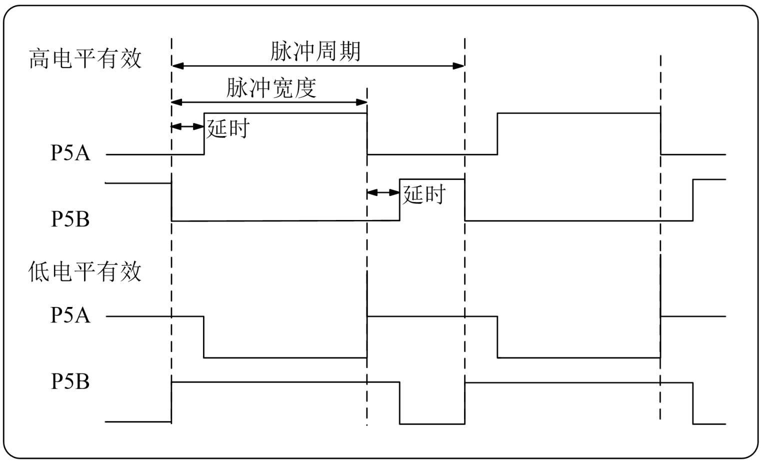


图 9.9 半桥输出模式输出信号示例

半桥输出模式可用于控制半桥和全桥控制电路，如图 9.10 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。

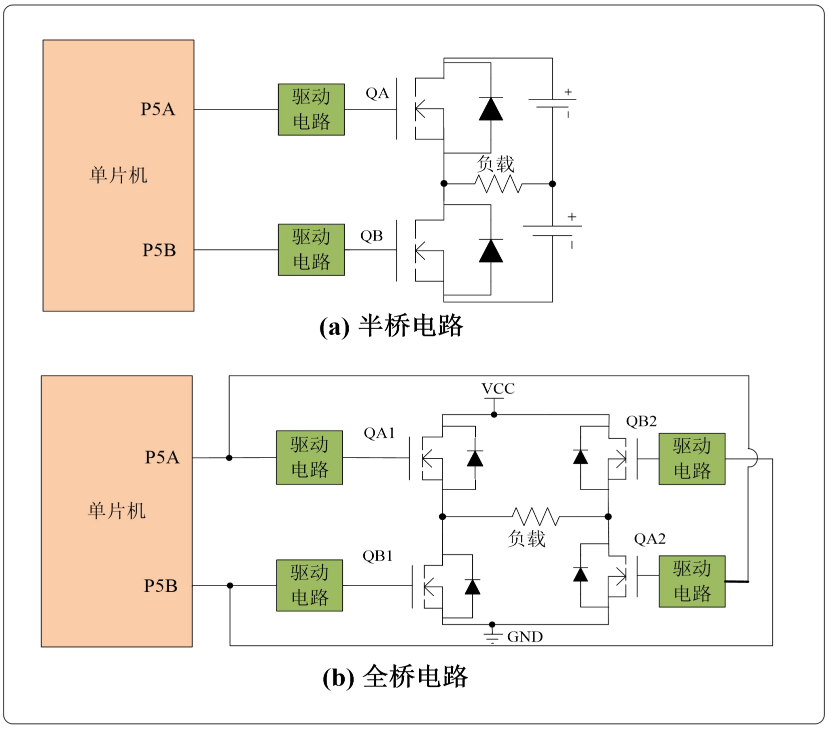


图 9.10 半桥输出模式应用举例

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。

9.4.7.1 死区延时

如图 9.11(a)、(b)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 9.11(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，PWM5 使用一个带有有独立的死区定时器的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 9.11（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式 9.8 所示。

★ 式9.8: 延时时间= $PDC<6:0> \cdot T_{INTHF}$

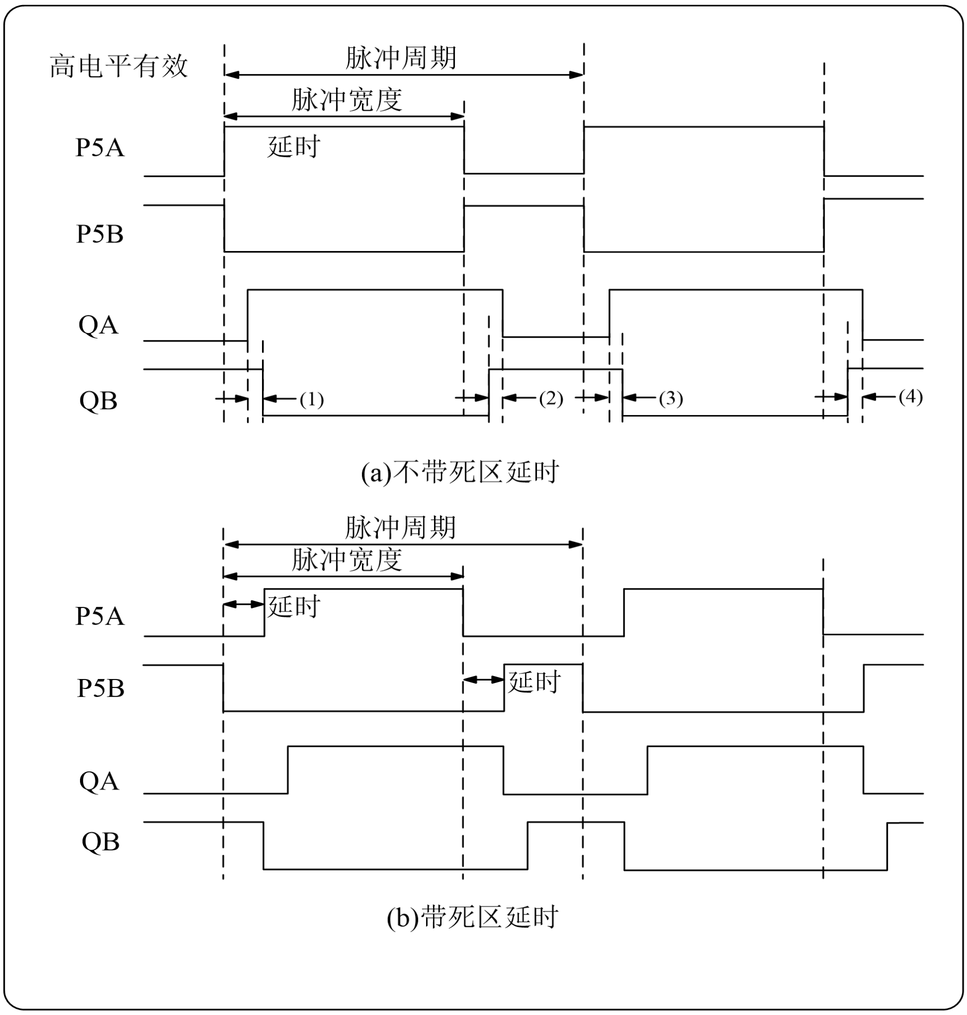


图 9.11　带死区控制和不带死区控制时的信号示例

在半桥模式下，当死区设置为 0 时，死区无效；当占空比为 0%或占 100%时，死区插入与是取决于是否发生了输出变化。若切换前电平和切换后输出电平无变化时不插入死区，若切换前输出电平和切换后输出电平发生变化时，则插入死区，如图 9.12 和图 9.13 所示。

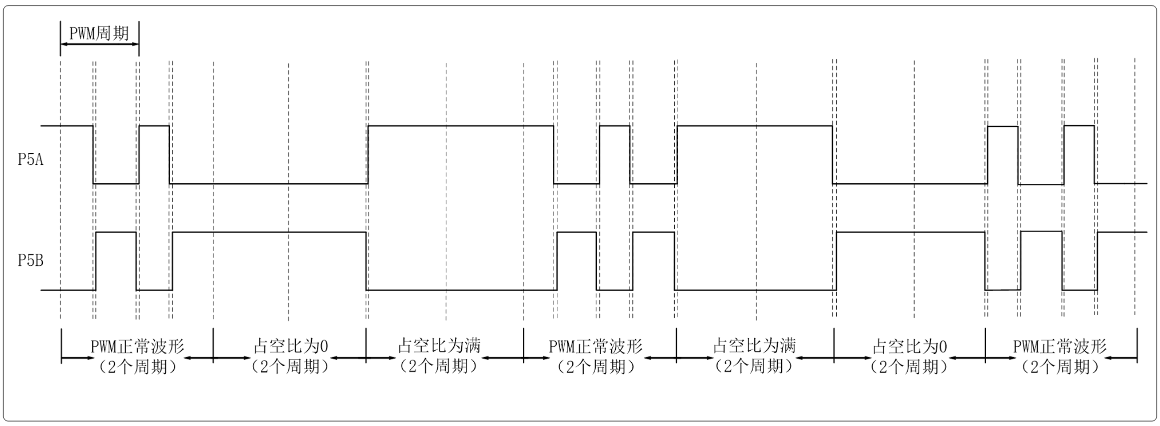


图 9.12 切换电平示意图（边沿对齐模式下）

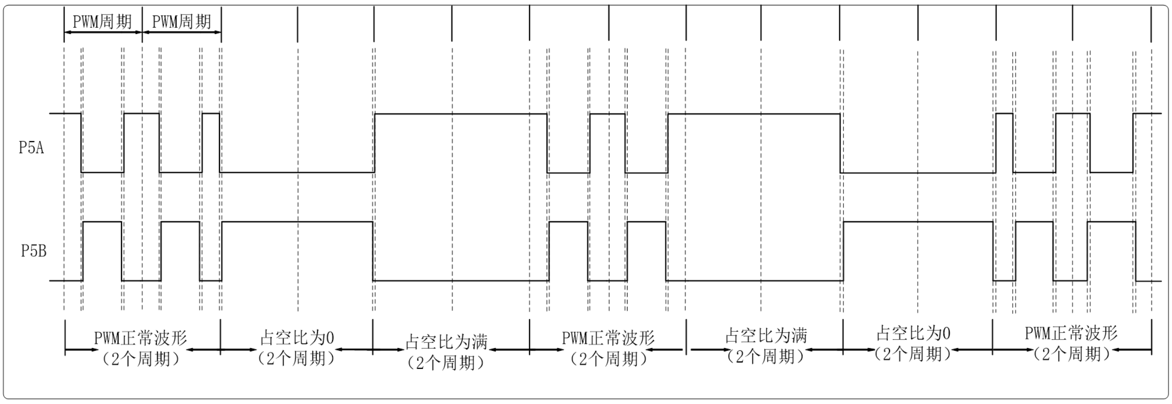


图 9.13 切换电平示意图（中心对齐模式下）

注：(1) 切换前后的电平指默认极性控制下的波形。

(2) 更新设置时，若 UDEN 为 1，且 UDEVT1=1 时，死区定时器被清零。

9.4.8 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<3:0>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 9.14(a)和 9.15(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 9.14(b)和 9.15(b)为全桥反向输出模式引脚信号示例。图 9.16 给出了全桥输出模式的应用电路示例。

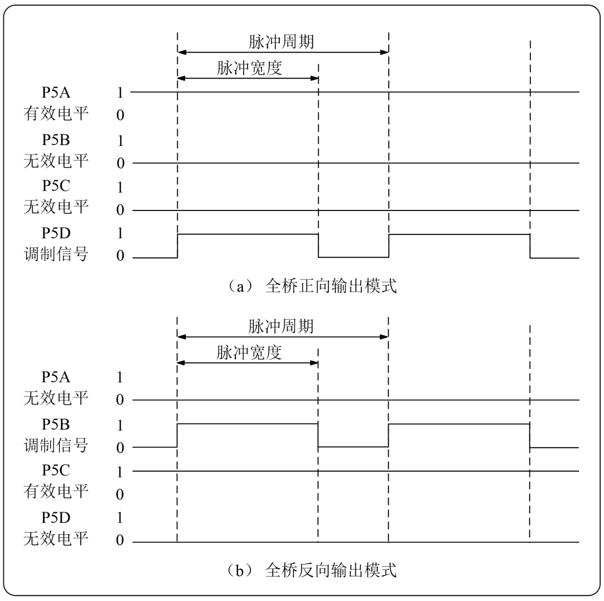


图 9.14 全桥输出模式引脚信号示例(高电平有效)

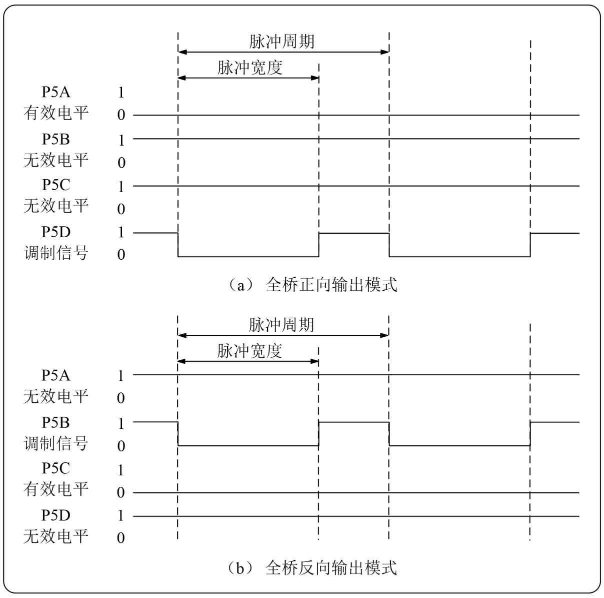


图 9.15 全桥输出模式引脚信号示例(低电平有效)

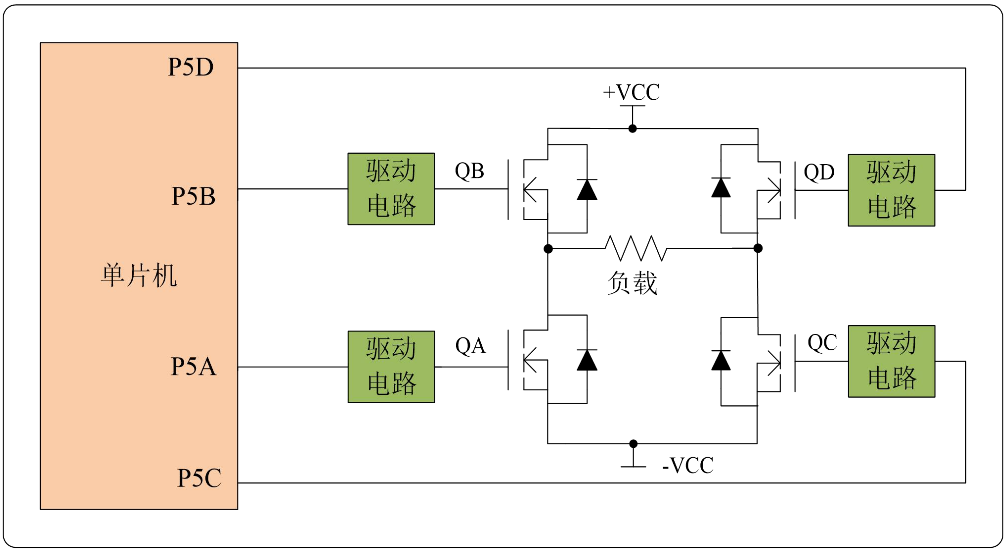


图 9.16 全桥输出模式应用电路示例

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 TRxx 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 9.17 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/INTHF•T2预分频值。

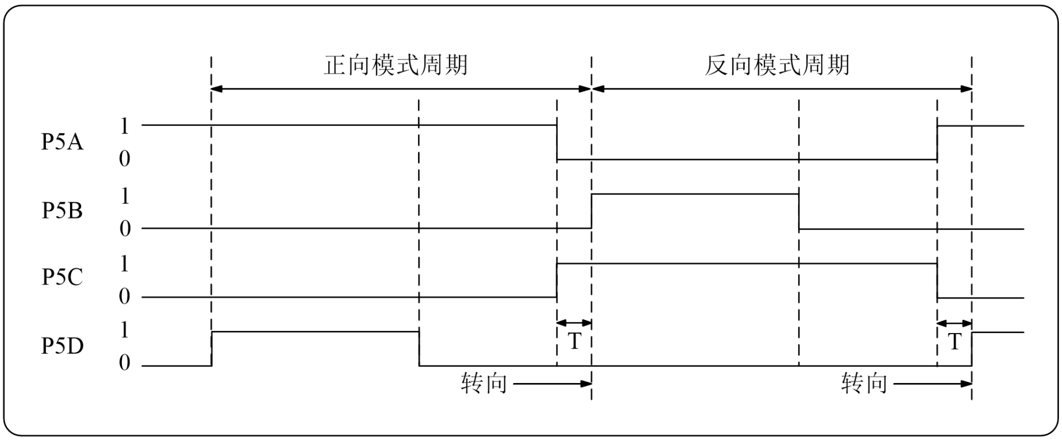


图 9.17 全桥正向、反向输出模式转换示例(高电平有效)

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一百，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 9.18 为此情况下各处信号示例。

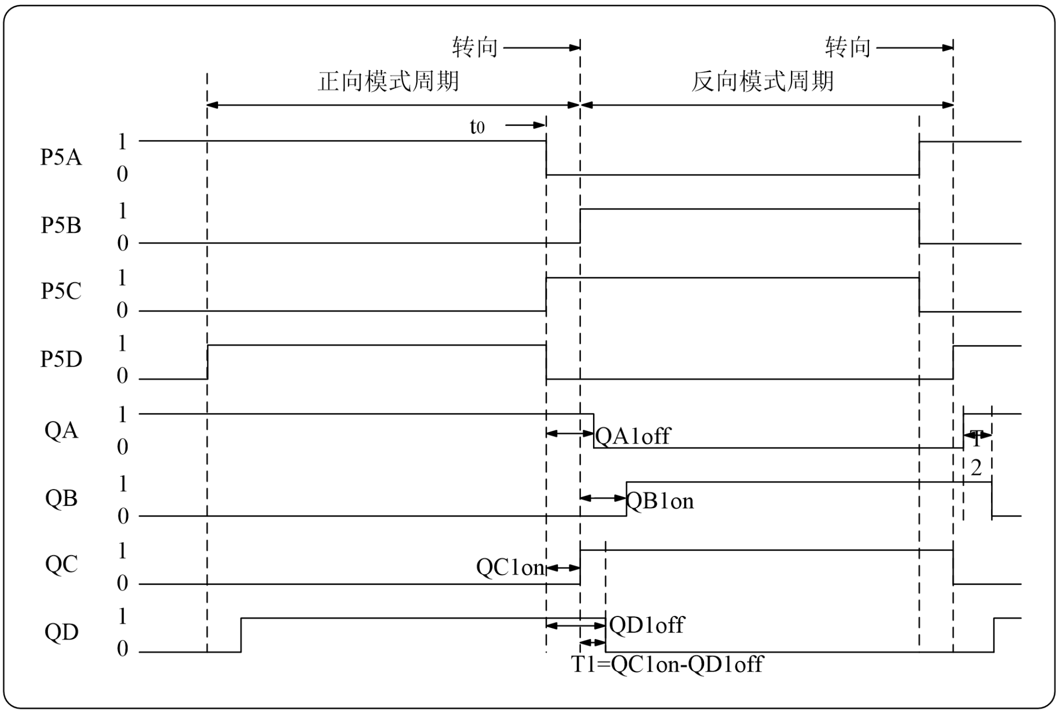


图 9.18 接近满占空比时换向时信号示例

图 9.18 中在 t0 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 T1，在后面一次换向时 QA 和 QB 产生直通电流，持续时间为 T2。

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。

9.4.9 脉冲转向控制

通过寄存器脉冲转向寄存器 PSTRCTL 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 PSTRCTL 中的 STRSYNC 位可以控制脉冲转向功能切换是与 PWM 周期同步还是立即有效（与指令周期同步）。通过 STRENA/ STRENB/ STRENC/ STREND/ 位可以设置相应端口引脚功能切换。默认情况下，P5A 的输出为 PWM5 引脚输出，P5B(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。

9.4.10 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 OCA/B 设置为 1 时，P5A/B 引脚选择 PWM 输出，输出极性由 PWM5PC 极性控制寄存器控制；当 OCA/B 设置为 0 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。

9.4.11 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 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 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。

9.4.12 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 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 端口输出的极性。

当 PWM 输出极性选择高有效时，P5A 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 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 寄存器中的 P5MOD<3:0>位，另一个是 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 9-3 所示。P5A/P5B 的极性控制需要在 UDEN 为 1 时才允许更新，而 P5C/P5B 的极性控制（由 PWM5M<3:0>控制）则与周期同步更新，因此在使用全桥模式时，需将 UDEN 一直打开，才能实现同步更新。

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。

9.4.13 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、强制控制和输出控制的更新。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 UDEVT1/0 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。UDEVT1 控制周期和占空比的更新，UDEVT0 控制输出控制、极性控制和强制控制寄存器的更新。若 UDEVT1 为 0，则周期和占空比可以在 T2 为 0 时更新到缓冲中；若 UDEVT1 为 1 时，则立即更新周期和占空比到缓冲中，T2 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 UDEVT1 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 T2 继续计数，不影响占空比的完整性。

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、强制控制和输出控制的更新。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。

9.4.14 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 和 LPOL 输出所需要的极性。配置 HPOL 可以设置 P5A、P5C 的极性，配置 LPOL 可以设置 P5B、P5D 的极性。

在正常工作时，可以通过 PWM5CTL2 更新控制寄存器的 PFUSES 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。

9.4.15 自动关断和自动重启模式

9.4.15.1 自动关断模式

PWM5 模块具有自动关断功能。如图 9.19 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 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 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。

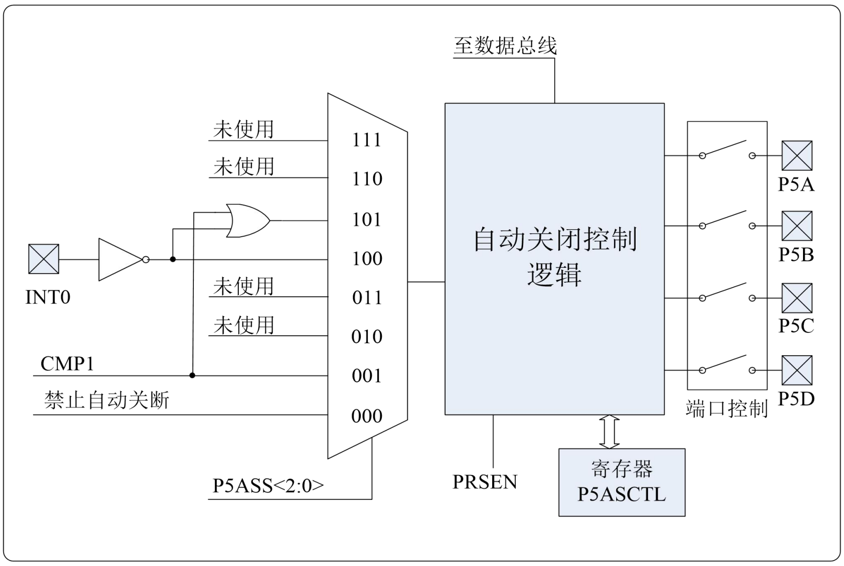


图 9.19 自动关断模式原理框图

通过寄存器 P5ASCTL 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 P5ASCTL 中的 P5ASE 位指示关断的状态。如果该位为 0，表示 PWM5 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 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。

② 使能的四个 PWM 引脚将被置于关断电平状态。

关断时，PWM5 的四路输出电平的状态由寄存器 P5ASCTL 的 P5SSAC<1:0>和 P5SSBD<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。

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 PRSEN=0)，四路输出仍然为关断模式电平，直到将 P5ASE 位清零，PWM 重启。

9.4.15.2 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。当清除自动关闭条件时，将通过硬件将 P5ASE 位清零，并且将恢复常规操作。

如图 9.20 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。

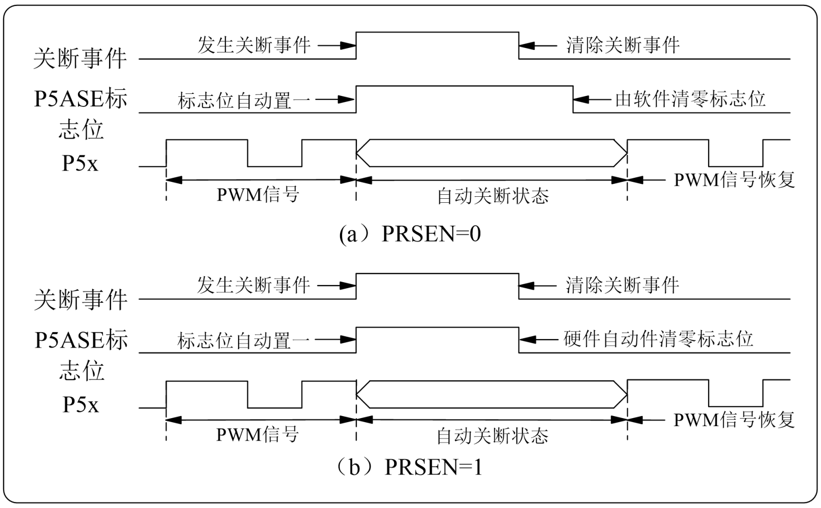


图 9.20 自动重启和软件重启 PWM

10 SSCI 模块

10.1 概述

KF8F3110/12/20/22/30/32 包含一个 SSCI（Synchronous Serial Communication interface）同步串行端口。它是用于和其他外设或单片机进行通信的串行接口。SSCI 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

10.2 SSCI 模块功能引脚

KF8F3110/12/20/22/30/32 的 SSCI 模块可以通过 PINSET 寄存器的 SSCIPIN、SDOPIN 及 SSPIN 位选择 SSCI 模块相关功能引脚位置。

当 SSCIPIN=0 时， SDA/SDI=P1.3, SCK/SCL=P1.5；

当 SSCIPIN=1 时， SDA/SDI=P0.0, SCK/SCL=P0.1。

注：14 或 16 引脚芯片使用 SSCI 功能时 SSCIPIN 必须配置为 1

当 SDOPIN=0 时， SDO=P1.7；

当 SDOPIN=1 时， SDO=P0.2。

注：14 或 16 引脚芯片使用 SSCI 功能时 SDOPIN 必须配置为 1

当 SSPIN=0 时， $\overline{SS}$=P2.3；

当 SSPIN=1 时， $\overline{SS}$=P0.4。

注：14 引脚芯片使用 SSCI 功能时 SSPIN 必须配置为 1

10.3 SSCI 相关寄存器

表 10-1 与 SSCI 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
128H	SSCICTL0	SSCIWCFL	SSCIOV	SSCIEN	SSCICKP	SSCIMOD3	SSCIMOD2	SSCIMOD1	SSCIMOD0
12AH	SSCICTL1	SSCI CALLEN	SSCI ACKSTA	SSCI ACKDAT	SSCI ACKEN	SSCIRCEN	STOPEN	RESTART EN	STARTEN
12BH	SSCISTA	SAMPLE	CKEGE	SSCIDA	SSCISTOP	SSCISTART	SSCIRW	SSCIUA	SSCIBUF
12CH	SSCIBUFR	SSCI 数据接收/发送缓冲寄存器							
12EH	SSCIADD	SSCI 的 I2C 地址寄存器							
	SSCIMSK	SSCIMSK7	SSCIMSK6	SSCIMSK5	SSCIMSK4	SSCIMSK3	SSCIMSK2	SSCIMSK1	SSCIMSK0
156H	PINSET	SSCIPIN	SDOPIN	SSPIN	UARTPIN	PWMPIN	-	-	-

10.3.1 SSCI 控制寄存器 0（SSCICTL0）

寄存器10.1: SSCICTL0: SSCI控制寄存器0(地址:128H)

复位值	bit7							bit0
0000 0000	SSCIWCFL	SSCIOV	SSCIEN	SSCICKP	SSCI MOD3	SSCI MOD2	SSCI MOD1	SSCI MOD0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SSCIWCFL: 写冲突检测位

1 = 正在发送前一个字时，又有数据写入SSCIBUFR 寄存器（必须用软件清零）

0 = 无冲突

SSCIOV: 接收溢出指示位

在SPI 模式下:

1 = 当SSCIBUFR 中仍保存前一数据时，又接收到一个新的字节。如果溢出，SSCISR 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SSCIBUFR，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SSCIBUFR寄存器启动。

0 = 无溢出

在I2C模式下:

1 = SSCIBUFR中仍保存前一数据时，又接收到一个新的字节。SSCIOV 在发送模式下被忽略。两种模式下都必须用软件将SSCIOV 清零。

0 = 无溢出

SSCIEN: 同步串行端口使能位

在SPI模式下:

1 = 使能串行端口并将SCK、SDO 和SDI 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 和SCL 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。

SSCICKP: 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 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。）

SSCIMOD<3:0>: 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 = 系统保留

0100 = SPI从动模式，时钟 = SCK引脚。使能$\overline{SS}$引脚控制。
0101 = SPI从动模式，时钟 = SCK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*(SSCIADD+1))
1001 = 允许SSCIMSK寄存器读写操作
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

图注：R=可读　W=可写　-=未用　U=未实现位

10.3.2 SSCI 控制寄存器 1（SSCICTL1）

寄存器10.2: SSCICTL1: SSCI控制寄存器1(地址:12AH)

复位值	bit7							bit0
0000 0000	SSCI CALLEN	SSCI ACKSTA	SSCI ACKDAT	SSCI ACKEN	SSCI RCEN	STOPEN	RESTART EN	STARTEN
	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

SSCICALLEN:广播呼叫使能位（仅限 I2C 从动模式）
1 =允许在SSCISR 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
SSCIACKSTA: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
SSCIACKDAT: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
SSCIACKEN: 应答序列使能位（仅限I2C主控模式）
在主控接收模式下:
1 = 在SDA 和SCL 引脚启动应答序列，发送SSCIACKDAT数据位。由硬件自动清零。
0 = 应答序列空闲
SSCIRCEN: 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
STOPEN: 停止条件使能位（仅限 I2C 主控模式）

SCK 释放控制:
1 = 在SDA 和SCL 引脚启动停止条件。由硬件自动清零。
0 = 停止条件空闲
RESTARTEN:重复启动条件使能位（仅限I2C主控模式）
1 =在SDA 和SCL 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
STARTEN: 启动条件使能位（仅限I2C主控模式）
在主控模式下:
1=在SDA 和SCL引脚启动条件。由硬件自动清零。
0=启动条件空闲

图注：R=可读 W=可写 -=未用 U=未实现位

注: 对于SSCIACKEN、SSCIRCEN、STOPEN、RESTARTEN 和STARTEN 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SSCIBUFR 进行写操作（禁止写SSCIBUFR）。

10.3.3 SSCI 状态寄存器（SSCISTA）

寄存器10.3: SSCISTA: SSCI状态寄存器(地址:12BH)

复位值 0000 0000	bit7							bit0
	SAMPLE	CKEGE	SSCIDA	SSCISTOP	SSCI START	SSCIRW	SSCIUA	SSCIBUF
	R/W	R/W	R	R	R	R	R	R

SAMPLE: 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SAMPLE 清零
I2C模式:
此位必须保持清零
CKEGE: SPI 时钟边沿选择位
SPI模式，SSCICKP = 0:
1 = 在SCK 的下降沿发送数据
0 = 在SCK 的上升沿发送数据
SPI 模式，SSCICKP = 1:
1 = 在SCK 的上升沿发送数据
0 = 在SCK 的下降沿发送数据
I2C模式:
此位必须保持清零
SSCIDA: 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

SSCISTOP: 停止位（仅I2C模式）

当禁止SSCI 模块或上次检测到启动位时，该位被清零。

SSCIEN 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

SSCISTART: 启动位（仅I2C模式）

当禁止SSCI 模块或上次检测到停止位时，该位被清零。

SSCIEN 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

SSCIRW: 读/ 写信息位（仅I2C模式）

该位用来保存在上次地址匹配后的SSCIRW 位信息。此位仅在地址匹配与遇到下一个启动位、停止位或SSCIACK 位之间有效。

I2C主模式下

1 = 读

0 = 写

I2C从模式下

1 = 发送正在进行

0 = 发送未进行

该位与STARTEN、RESTARTEN、STOPEN、SSCIRCEN或SSCIACKEN位的或运算结果指示SSCI是否处于空闲状态。

SSCIUA: 更新地址位（仅10位I2C模式）

1 = 表示用户需要更新SSCIADD 寄存器中的地址

0 = 不需要更新地址

SSCIBUF: 缓冲器满状态位

接收（SPI和I2C模式）：

1 = 接收完成， SSCIBUFR满

0 = 接收未完成， SSCIBUFR空

发送（仅I2C模式）：

1 = 正在发送， SSCIBUFR满

0 = 发送完成， SSCIBUFR空

图注：R=可读 W=可写 -=未用 U=未实现位

10.3.4 SSCI 屏蔽寄存器（SSCIMSK）

寄存器10.4: SSCIMSK: SSCI屏蔽寄存器(地址:12EH)

	bit7							bit0
复位值 1111 1111	SSCI MSK7	SSCI MSK6	SSCI MSK5	SSCI MSK4	SSCI MSK3	SSCI MSK2	SSCI MSK1	SSCI MSK0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SSCIMSK<7:1>:屏蔽位

1 = 接收到的地址的bit n 与SSCIADD<n> 比较以检测I2C的地址匹配情况

0 = 接收到的地址的bit n 不用于检测I2C的地址匹配情况

SSCIMSK<0>: 在I2C从动模式下，10位地址的屏蔽位

在I2C 从动模式，10位地址（SSCIMOD<3:0> = 0111或1111）条件下:

1 = 将接收到的地址的bit 0位与SSCIADD<0> 相比较以检测I2C的地址匹配情况

0 = 接收到的地址的bit 0位不用于检测I2C的地址匹配情况

在I2C从动模式，7位地址条件下，该位为无关位

图注：R=可读 W=可写 -=未用 U=未实现位

注:当SSCICTL0位SSCIMOD<3:0>=1001时，不能对SSCIADD寄存器进行读和写，任何对SSCIADD寄存器（地址也是12EH）的读和写操作均是对SSCIMSK寄存器操作。

10.3.5 SSCI I2C 地址寄存器（SSCIADD）

在10位I2C从动模式下，该地址寄存器是复用的。

寄存器10.5: SSCIADD: I2C地址寄存器(地址:12EH)

复位值	bit7							bit0
0000 0000	SSCI ADD7	SSCI ADD6	SSCI ADD5	SSCI ADD4	SSCI ADD3	SSCI ADD2	SSCI ADD1	SSCI ADD0
	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：

SSCIADD<7:3>: 未使用，SSCIADD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I2C规范制定必须等于11110，但是这这些位由硬件进行比较且不受该寄存器中的值的影响

SSCIADD<2:1>: 保存10位地址的高两位。

SSCIADD0: 未使用，为无关位。

10位从动模式下——低地址字节：

SSCIADD<7:0>: 10位地址的低8位

7位从动模式下：

SSCIADD<7:1>: 7位地址

SSCIADD0: 未使用，为无关位

图注：R=可读 W=可写 -=未用 U=未实现位

注：在I2C模式下，不支持SSCIADD寄存器的值为0、1或2的情况。

10.4 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

10.4.1 工作原理

I2C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I2C通信:SCL引脚作为时钟线，SDA引脚作为数据线。通过将SSCI使能位SSCIEN（SSCICTL0<5>）置1以使能SSCI模块的功能。

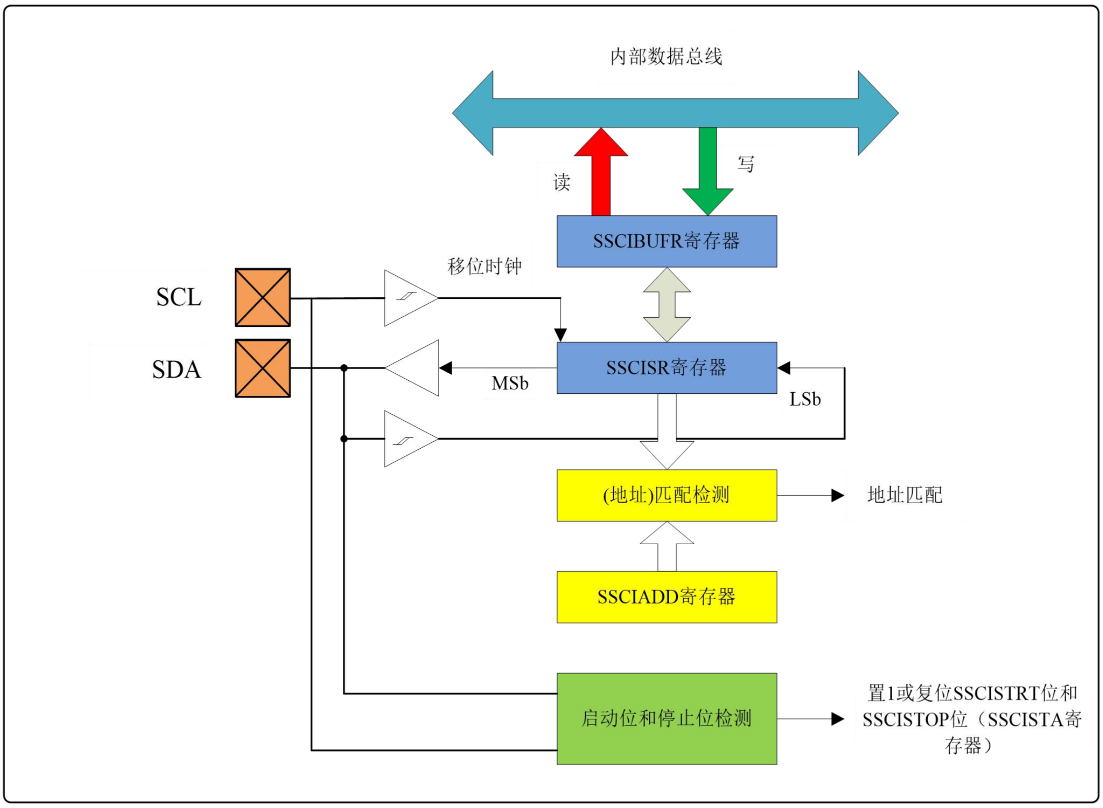


图10.1 **I2C**模式方框图

SSCI模块有8个寄存器用于I2C操作，这7个寄存器是:

- SSCI控制寄存器0（SSCICTL0）
- SSCI控制寄存器1（SSCICTL1）
- 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TR1位置1，使这些引脚成为输入引脚）。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。

10.4.2 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。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10-2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 10-2 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。

10.4.2.1 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10.3）。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。

10.4.2.2 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。

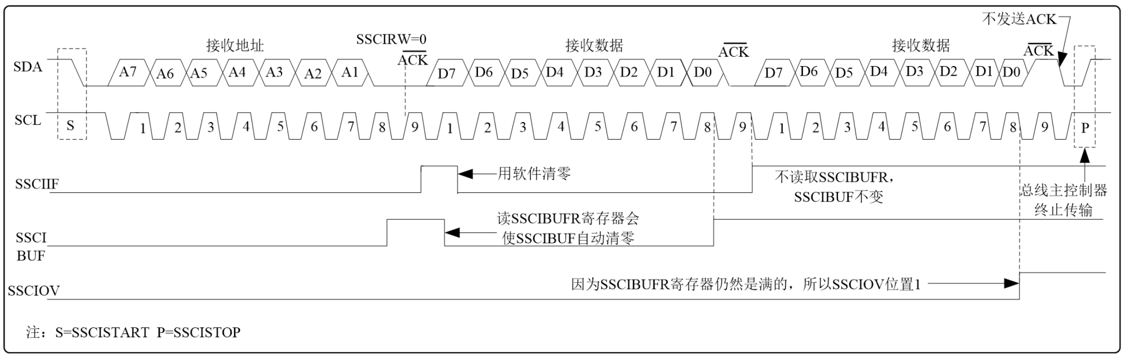


图10.2 从动模式时序（接收，7位地址）

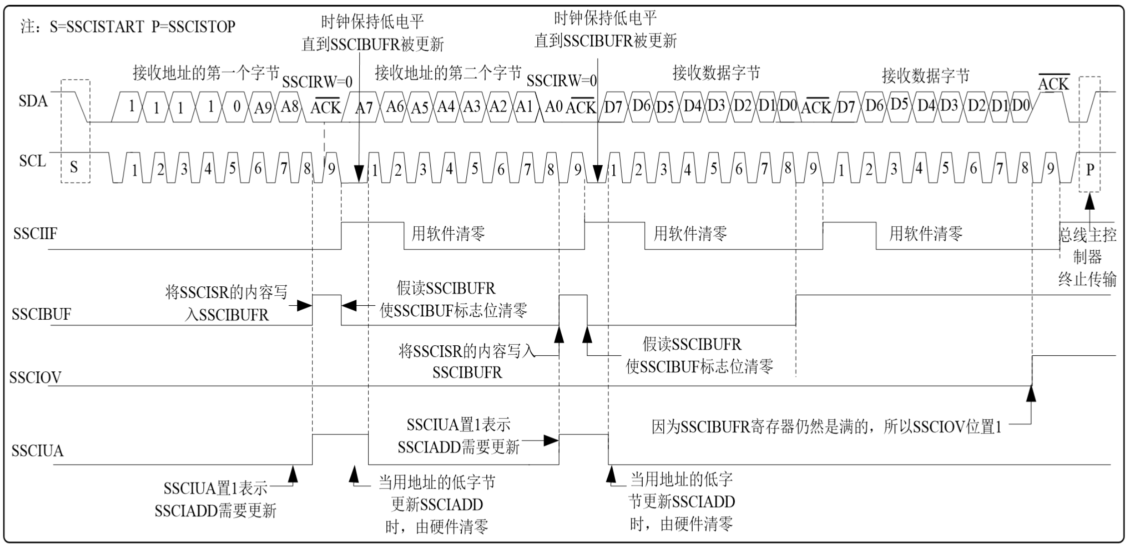


图 10.3　从动模式时序（接收，10 位地址）

从动接收设置:

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引脚为输入，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，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；

7、 缓冲器满标志位SSCIBUF被置1；产生$\overline{\text{ACK}}$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。

10.4.2.3　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（SSCICTL0<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输入的下降沿被移出。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个时钟脉冲的下降沿被置1。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。

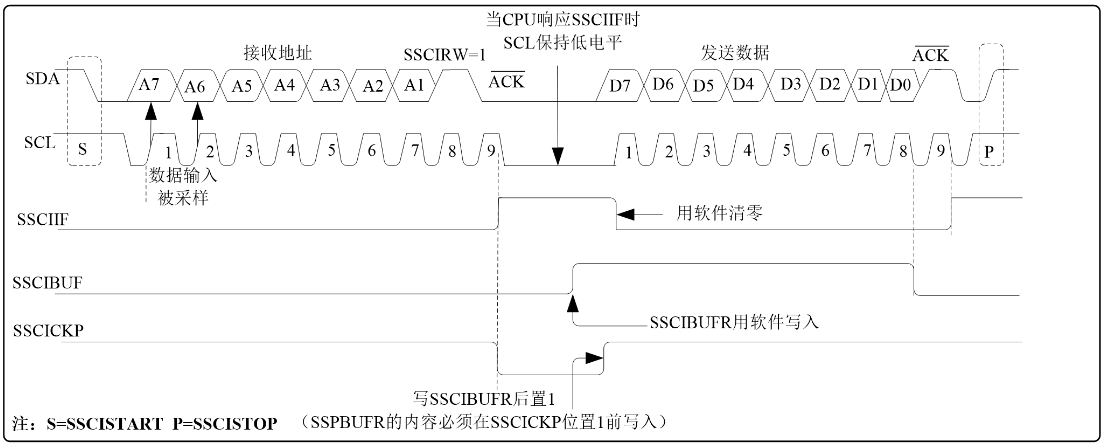


图 10.4　从动模式时序（发送，7 位地址）

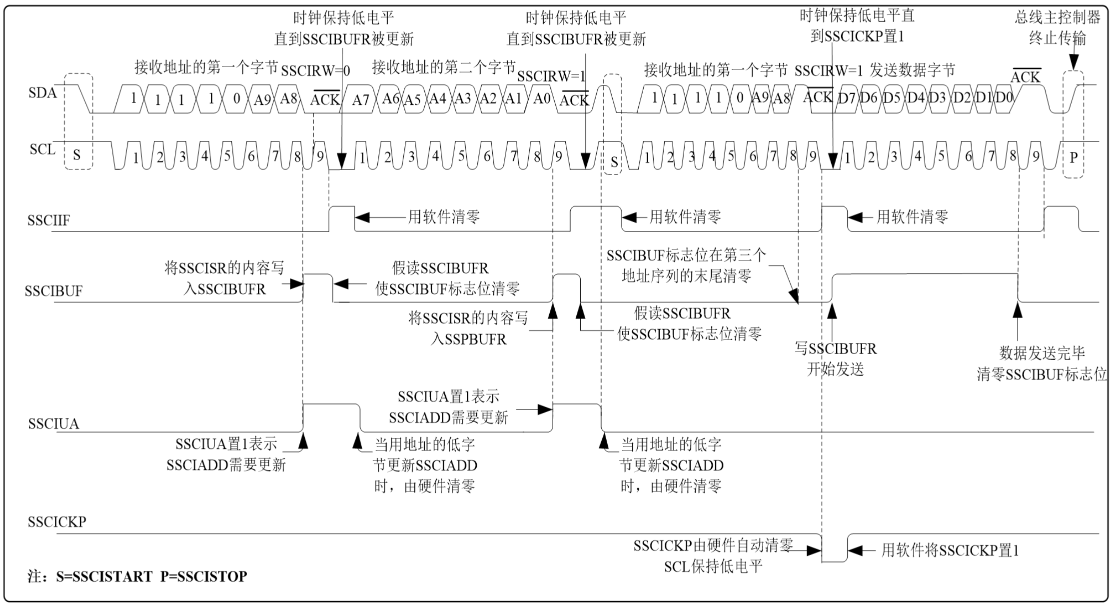


图 10.5　从动模式时序（发送，10 位地址）

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、置 1SSCICKP 位使能 SCL 引脚。主控制器必须再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与主控制器同步。

9、标志位 SSCIIF 在第 9 个时钟脉冲的下降沿被置 1，需软件清零。

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。

10.4.2.4 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（SSCICTL1<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10.6所示。

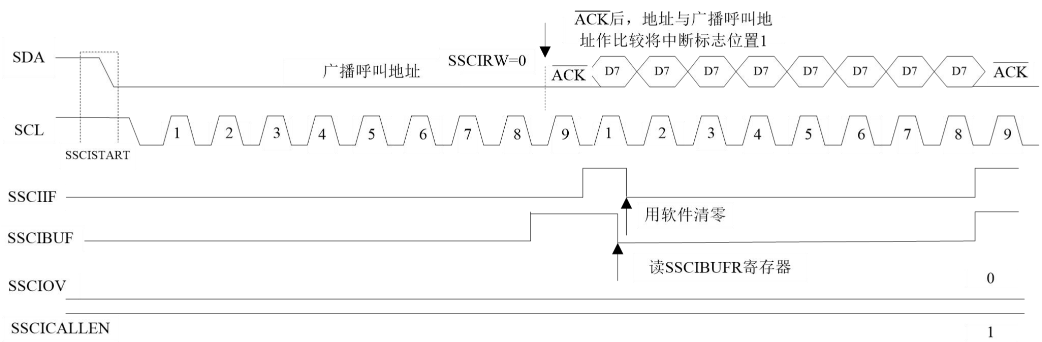


图10.6 从动模式广播呼叫地址时序（7或10位地址模式）

10.4.3 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。

在主控模式下，SCL和SDA线通过清零相应的TR位来控制。输出电平始终为低电平，而与端口输出的值无关。因此当发送数据时，对于SDA线，必须将TR13或TR00置0（输出），对于SCL线，也要将TR15或TR01位置0（输出）。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。

10.4.3.1 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10.7 为I2C主模式框图。

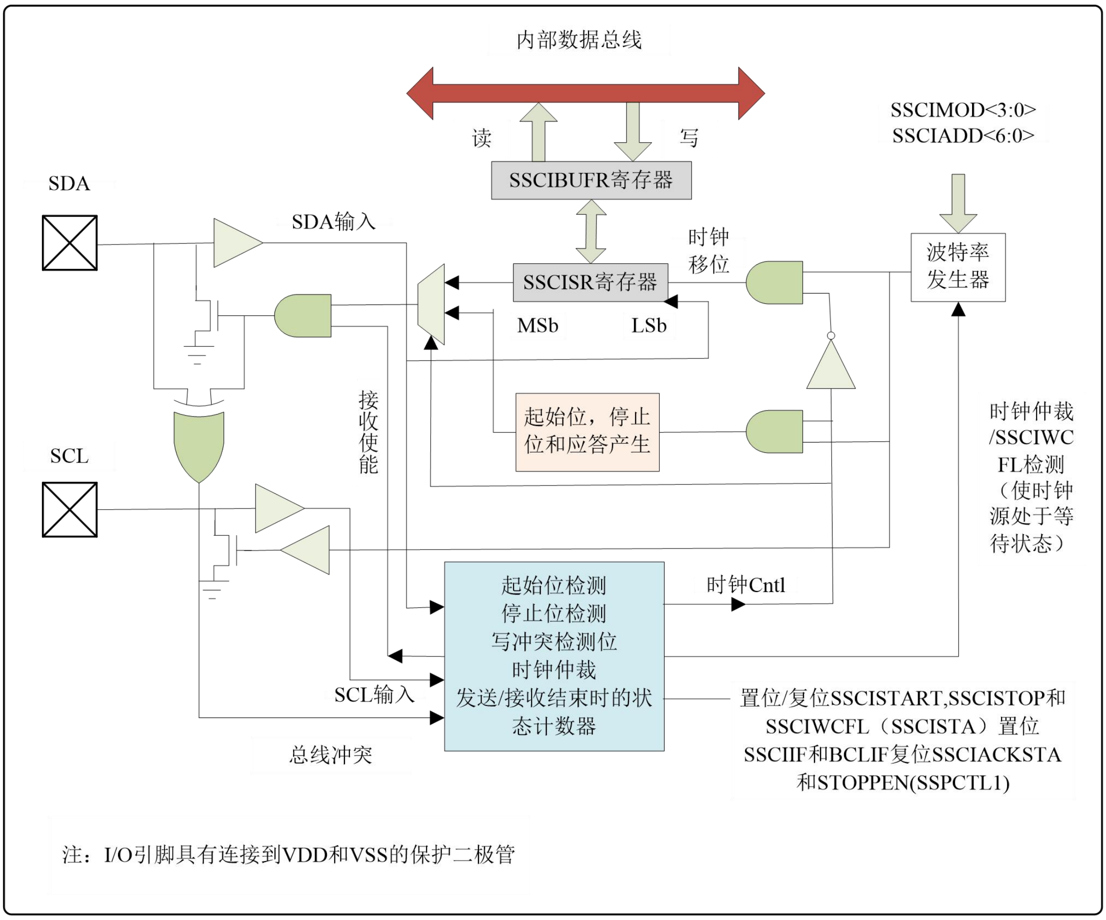


图10.7　I2C主模式框图

10.4.3.2　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 （SSCICTL1寄存器）置1 产生启动条件。
2） SSCIIF 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SSCICTL1 寄存器的SSCIACKSTA位。
6） SSCI模块在第9 个时钟周期的末尾将SSCIIF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SSCICTL1 寄存器的SSCIACKSTA 位。
10） SSCI 模块在第9 个时钟的末尾将SSCIIF 位置1，产生一个中断。
11） 用户通过将停止使能位（STOPEN）位（SSCICTL1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。

10.4.3.3 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10.8和图10.9 所示。

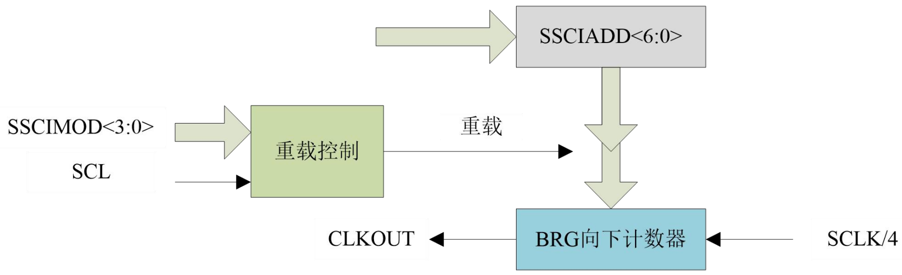


图 10.8 波特率发生器框图

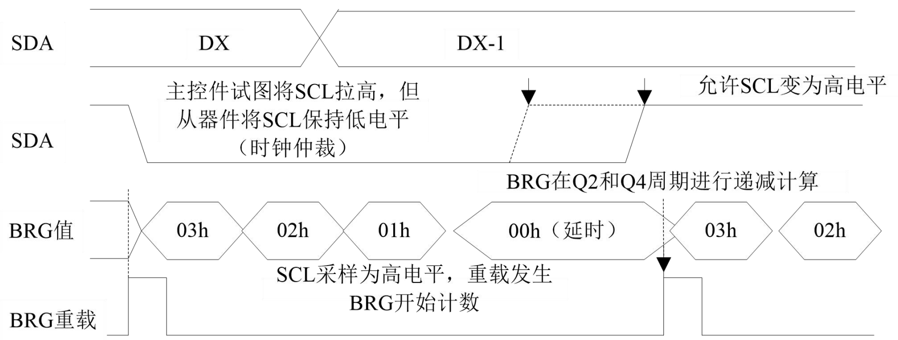


图 10.9 带有时钟仲裁的波特率发生器时序

10.4.3.4 I2C 主控模式启动条件时序

要发起启动条件，用户应将SSCICTL1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，SSCICTL1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SSCICTL1 的低5 位进行写操作。

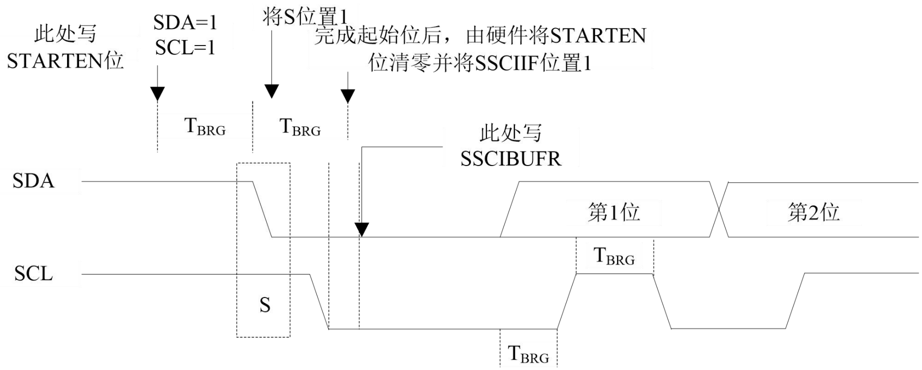


图10.10 第一个启动位时序

10.4.3.5 I2C 主控模式重复启动条件时序

将RESTARTEN位（SSCICTL1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（SSCICTL1 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一个地址字节。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SSCICTL1 的低5位进行写操作。

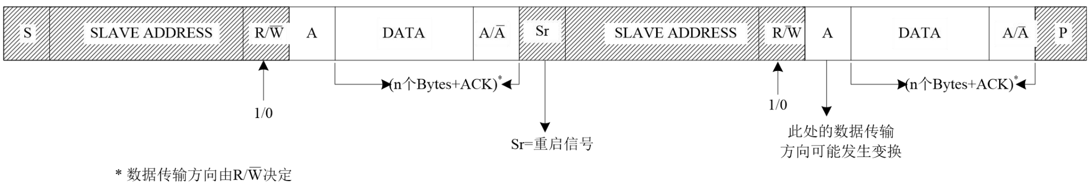


图10.11 I2C协议复合数据帧格式

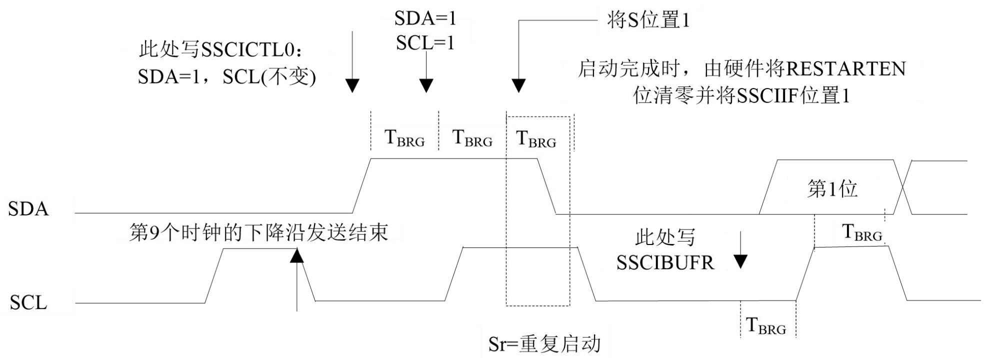


图10.12 重复启动条件时序波形

10.4.3.6 I2C 主控模式发送

I2C 协议中 SDI 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 10.13 所示是 I2C 协议典型写数据帧格式。

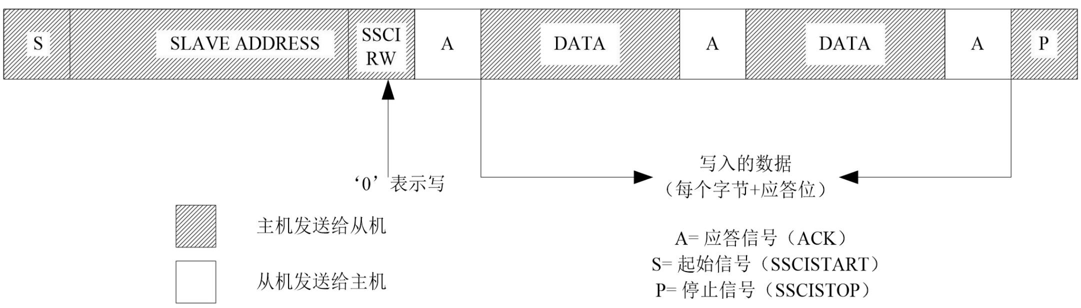


图 10.13 I2C 协议典型写数据帧格式

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（SSCICTL1 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（SSCICTL1 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 10.14 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。

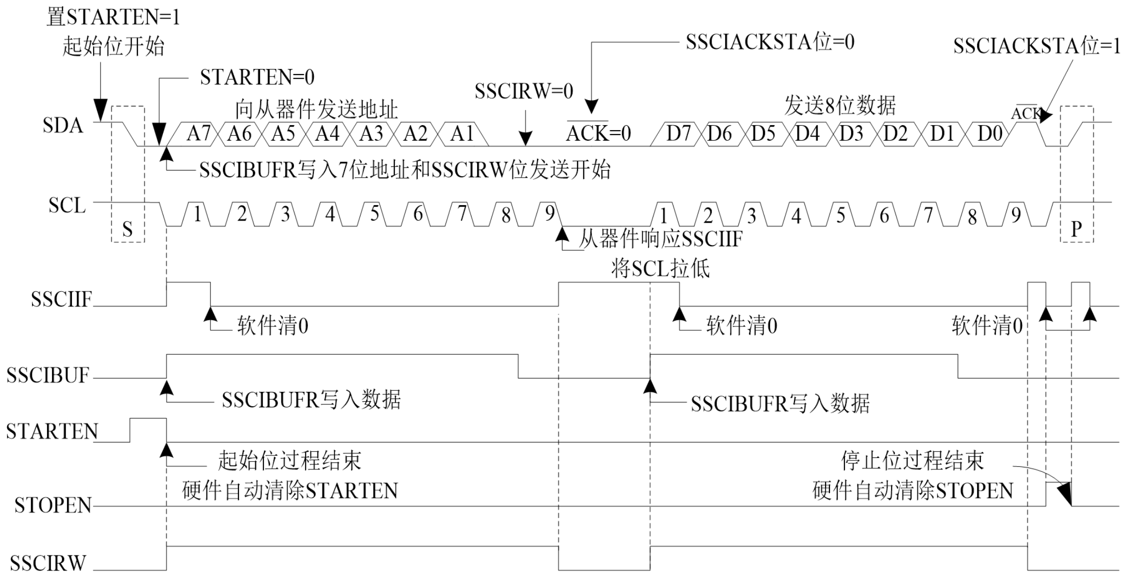


图 10.14　硬件 I2C 主控模式发送数据时序图（7 位地址）

10.4.3.7　I2C 主控模式接收

I2C主控模式数据接收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10.15为I2C协议典型读数据帧格式。

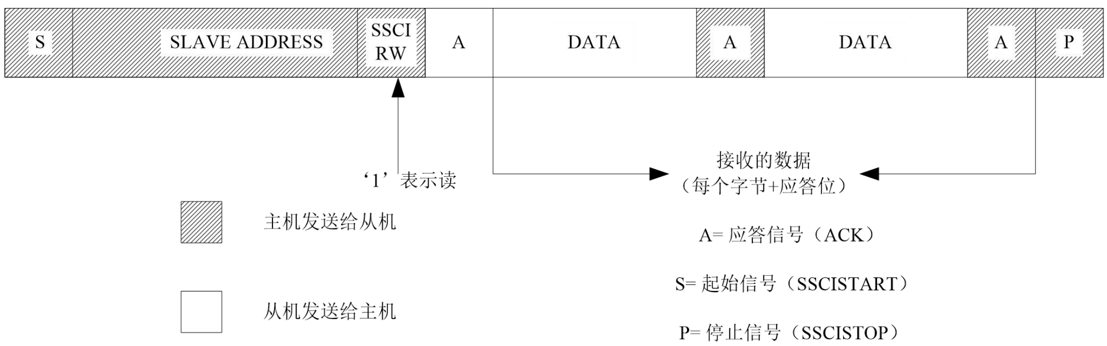


图10.15　I2C协议典型读数据帧格式

通过编程接收使能位SSCIRCEN（SSCICTL1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（SSCICTL1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10.16 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。

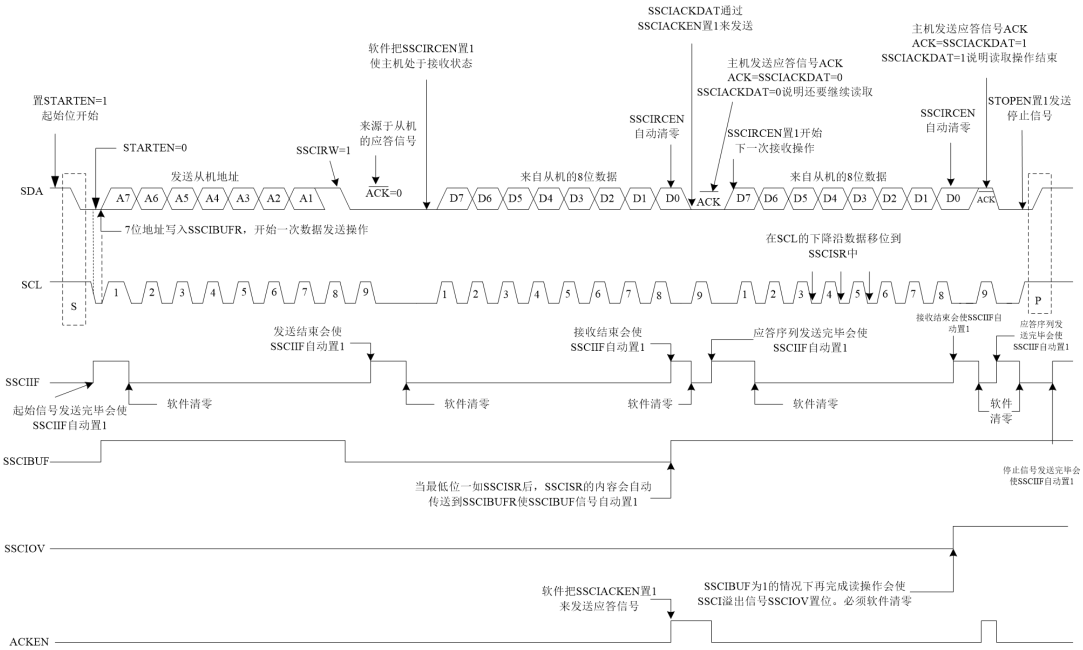


图10.16 I2C主控模式接收数据时序图（7位地址）

10.4.3.8 应答序列时序

将应答序列使能位SSCIACKEN（SSCICTL1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）。

10.4.3.9 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（SSCICTL1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，主控器件将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）。

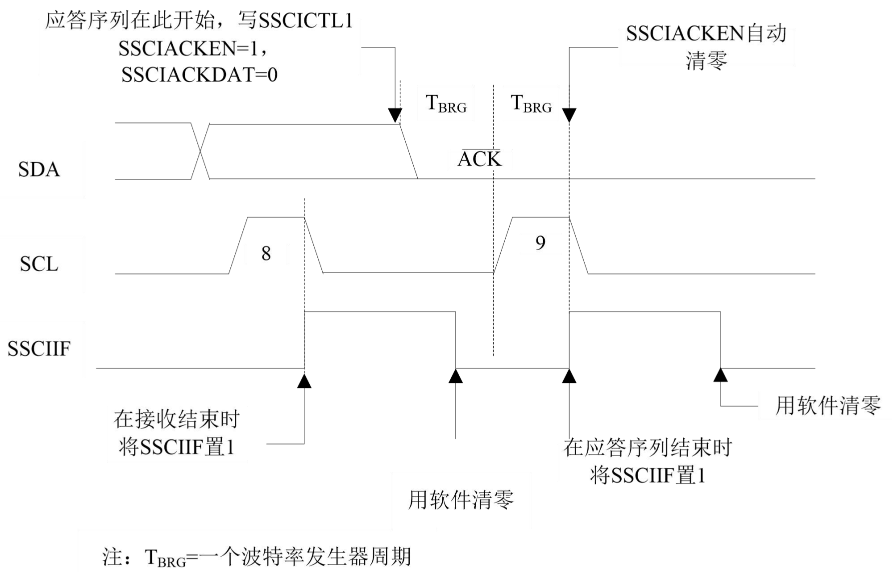


图10.17 应答序列时序波形

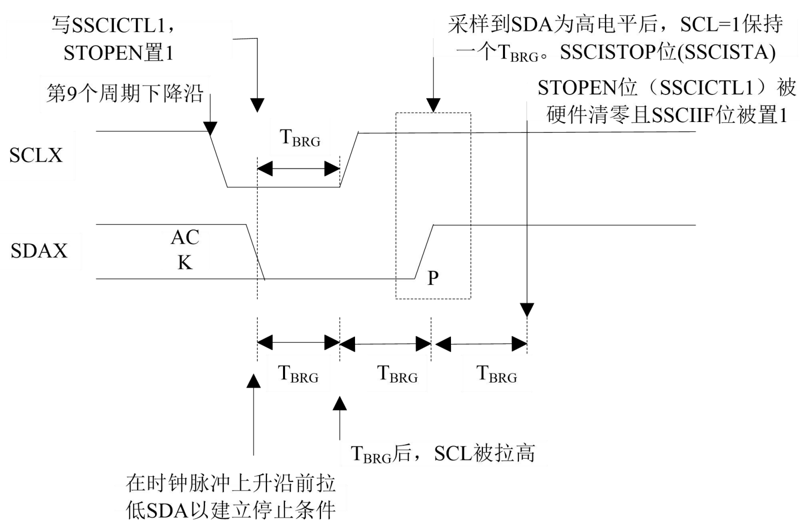


图10.18 停止条件接收或发送模式

10.4.3.10 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。

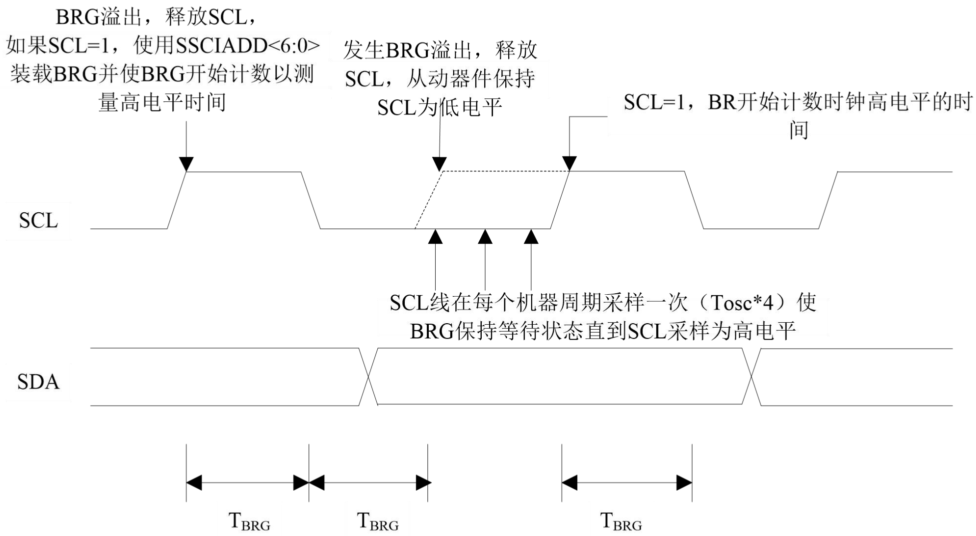


图 10.19　主控发送模式下的时钟仲裁时序

10.4.4 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（对应TR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。

10.4.4.1 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， SSCICTL1 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。

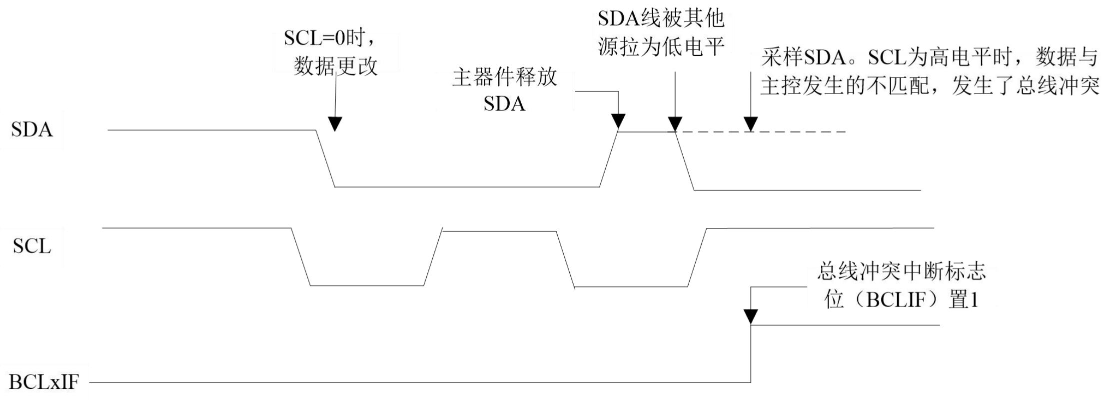


图 10.20 发送和应答时的总线冲突时序

10.4.4.2 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，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。

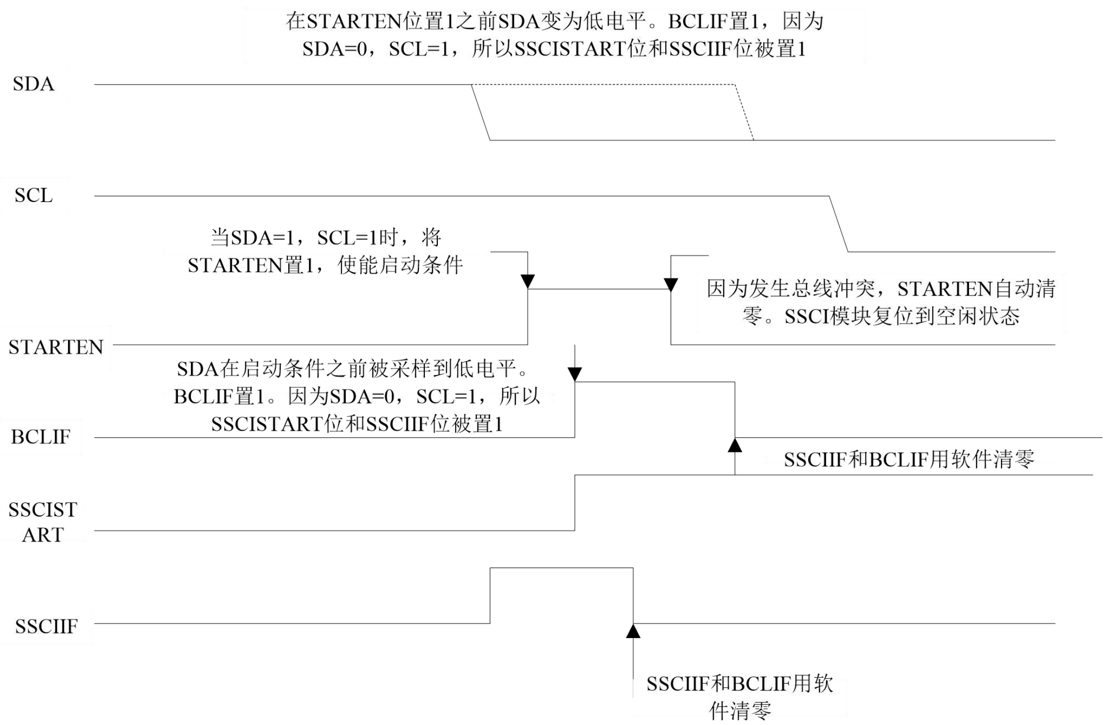
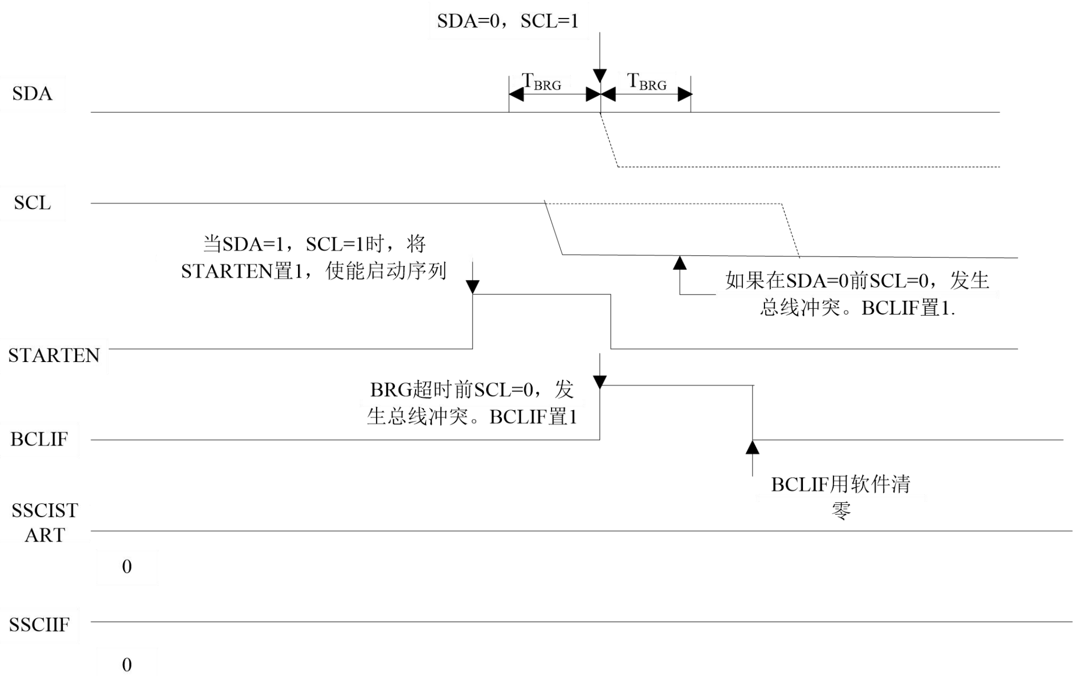


图10.21　启动条件期间的总线冲突（仅SDA）



图

10.22　启动条件期间的总线冲突（SCL=0）

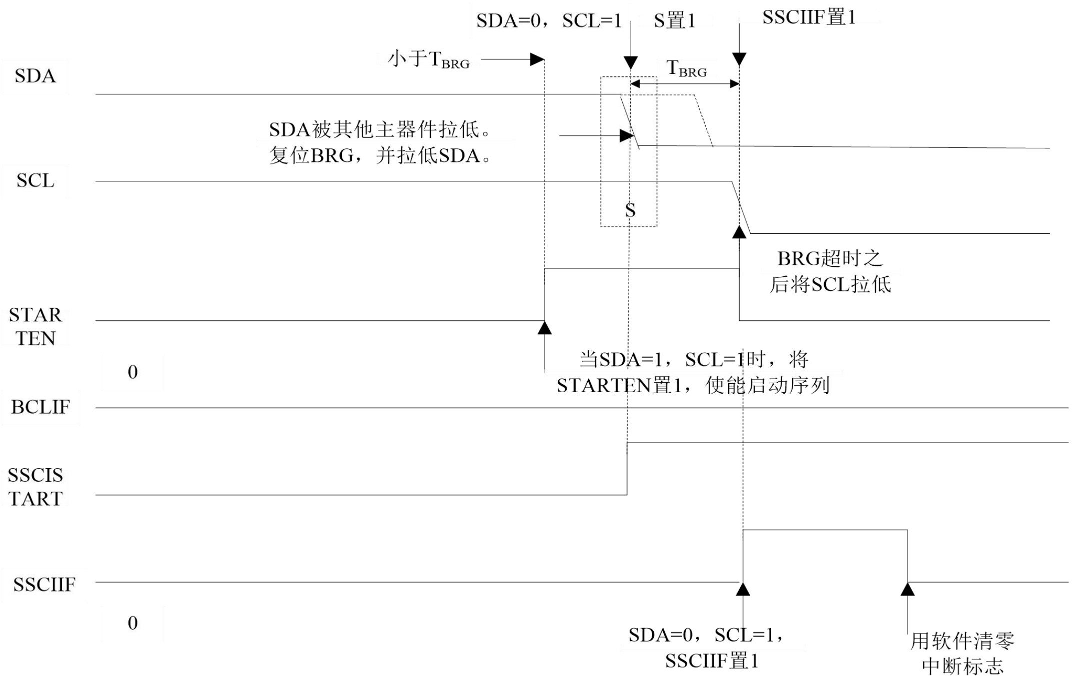


图10.23　启动条件期间由SDA仲裁引起的BRG复位

10.4.4.3　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。

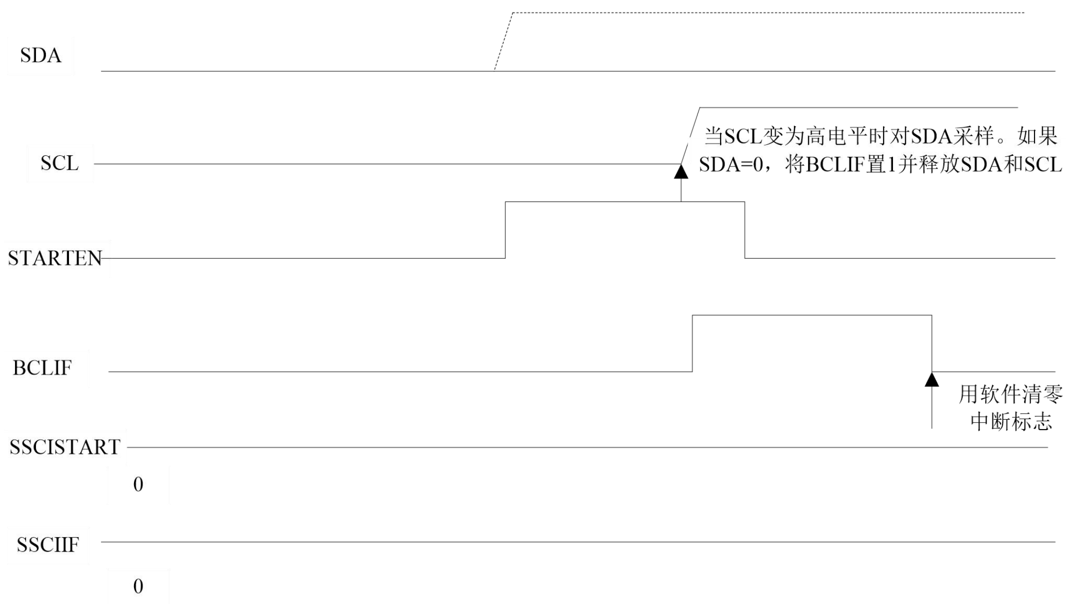


图 10.24 重复启动条件期间的总线冲突（情形 1）

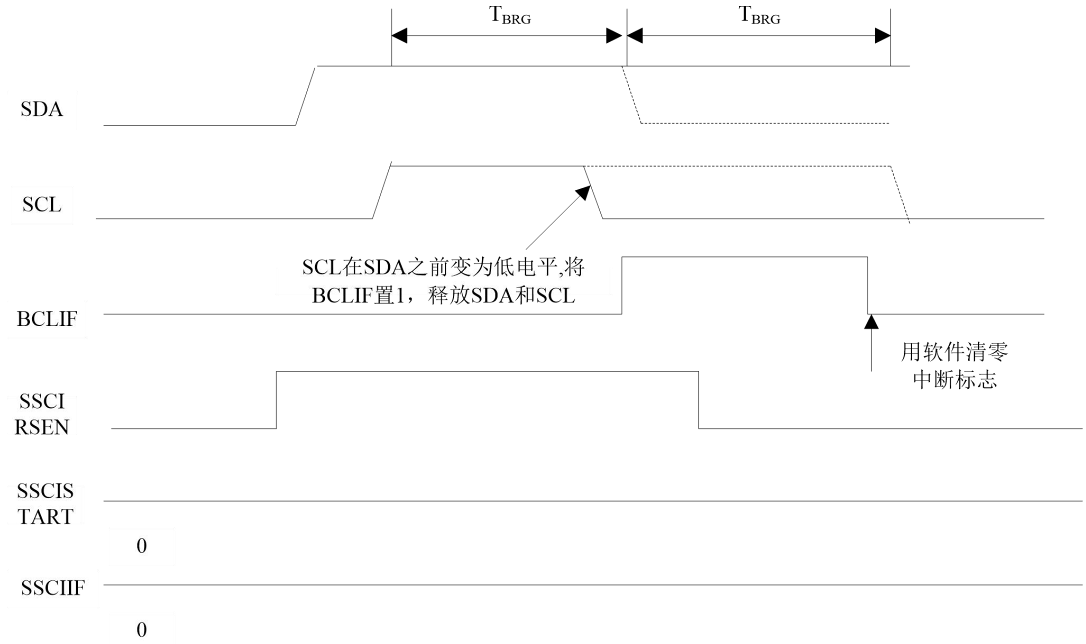


图 10.25 重复启动条件期间的总线冲突（情形 2）

10.4.4.4 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。

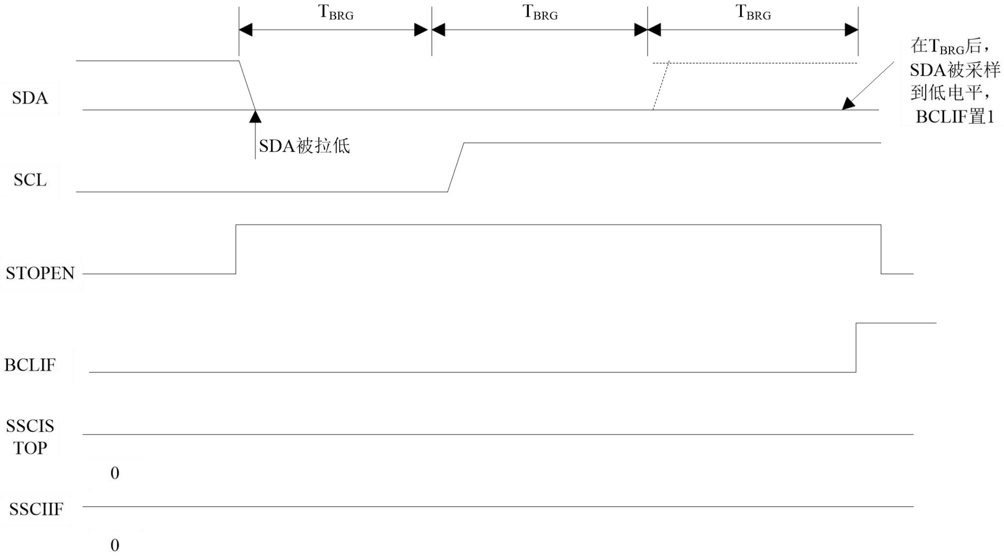


图10.26 停止条件期间的总线冲突（情形1）

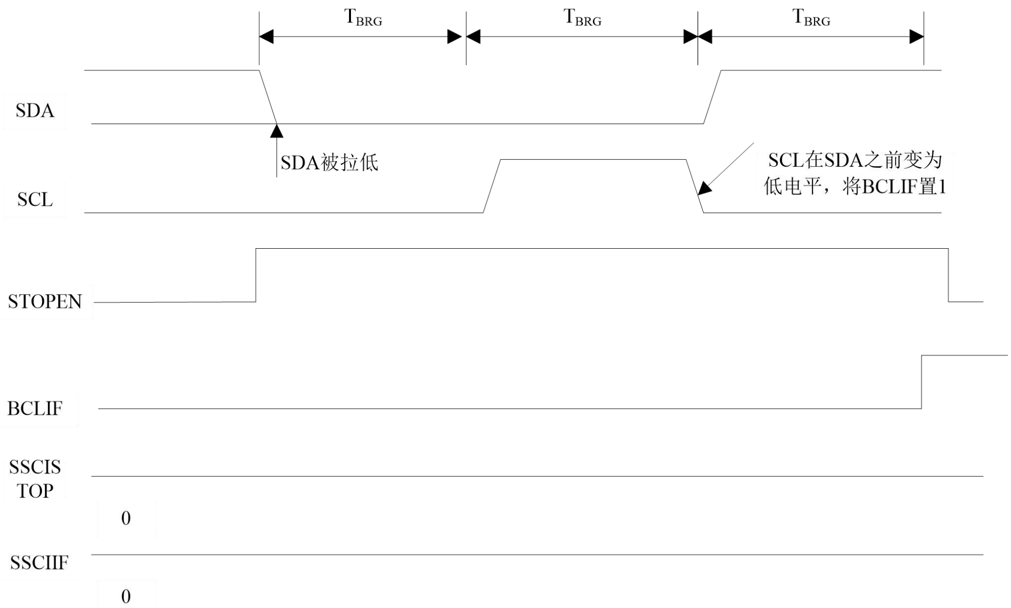


图10.27 停止条件期间的总线冲突（情形2）

10.4.4.5 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SSCIADD <7:1> 进行地址比较。
- 10 位地址模式：仅与SSCIADD <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。

10.5 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 10.28 所示:

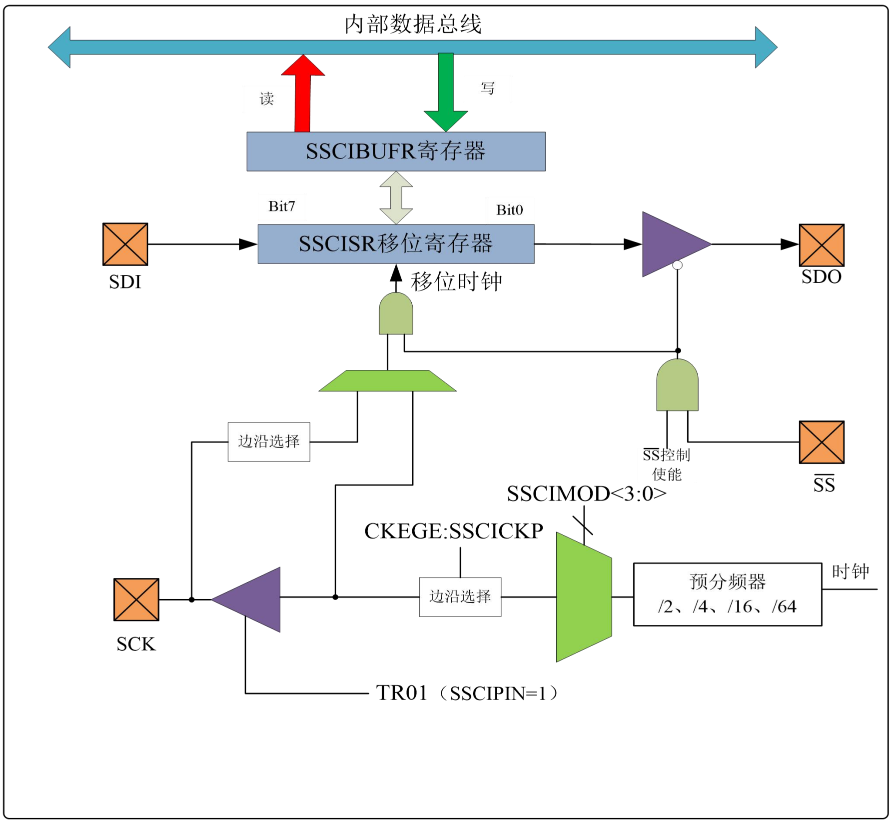


图 10.28　SPI 原理框图

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（SSCICTL0<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。

10.5.1 工作原理

在用户初始化SPI时，需要先通过对相应的控制位（SSCICTL0<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（SSCICTL0<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。

10.5.2 使能 SPI/IO 与外部链接

要使能串行端口， SSCI 使能位 SSCIEN （SSCICTL0<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 SSCICTL0 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（在 TR0 寄存器中）。即:

- ◆ SDI ，必须将 TR00 置 1（由 SPI 模块自动控制）（SSCIPIN=1）
- ◆ 对于SDO，必须将TR02清零（SDOPIN=1）
- ◆ 对于SCK （主控模式），必须将TR01清0（SSCIPIN=1）
- ◆ 对于SCK （从动模式），必须将TR01置1（SSCIPIN=1）

◆ 对于$\overline{SS}$，必须将TR04置1。（SSPIN=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 TRxx 寄存器设置为相反值来屏蔽。

10.5.3 典型连接

下图10.29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

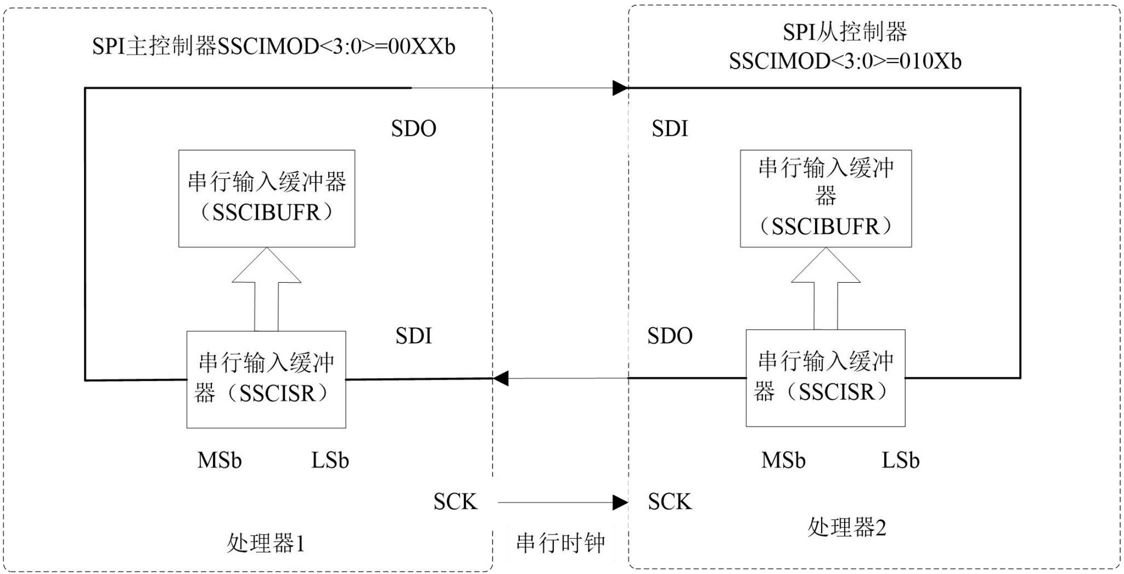


图 10.29 SPI 主从控制器连接

10.5.4 主控模式

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 10.29 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10.30、图10.31、图10.32 和图10.33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/4

SCLK/16

SCLK/64

下图10.30 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。

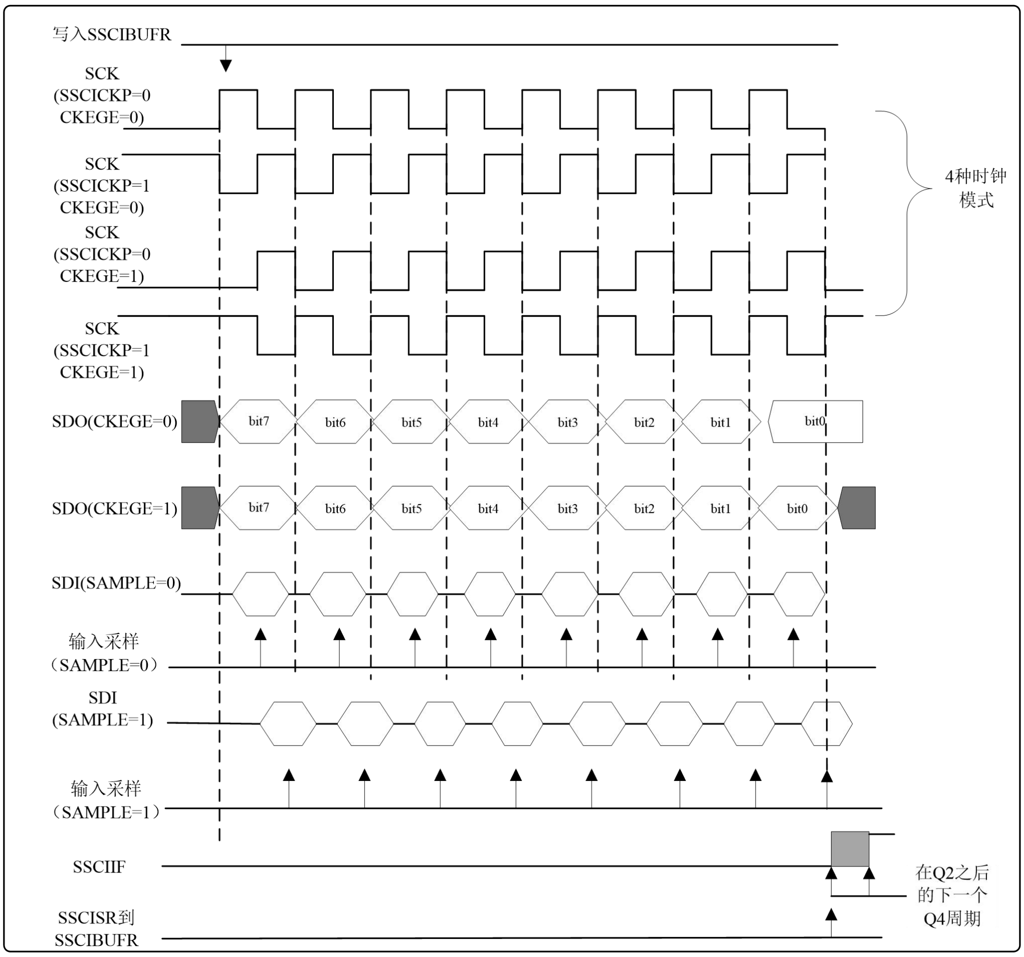


图 10.30 SPI 主控模式时序图

10.5.5 从动模式

10.5.5.1 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SSCIIF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SSCIIF信号，一般情况下就不使用SSCIBUF来判断SSCIBUFR是否有数据了。当SSCIIF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。

10.5.5.2 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）。

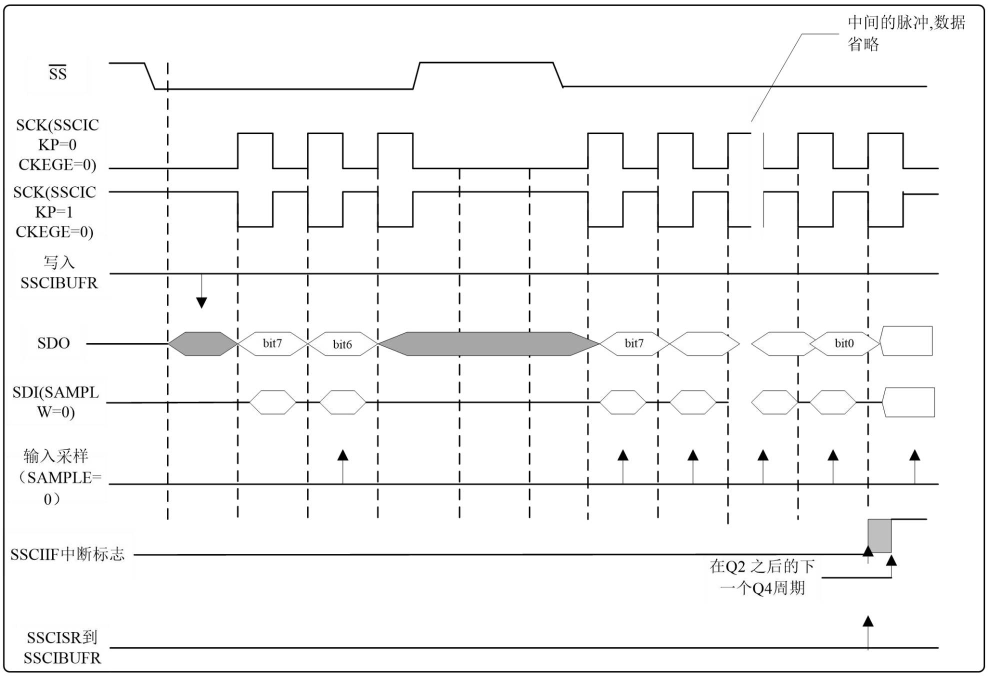


图10.31 从动时序图

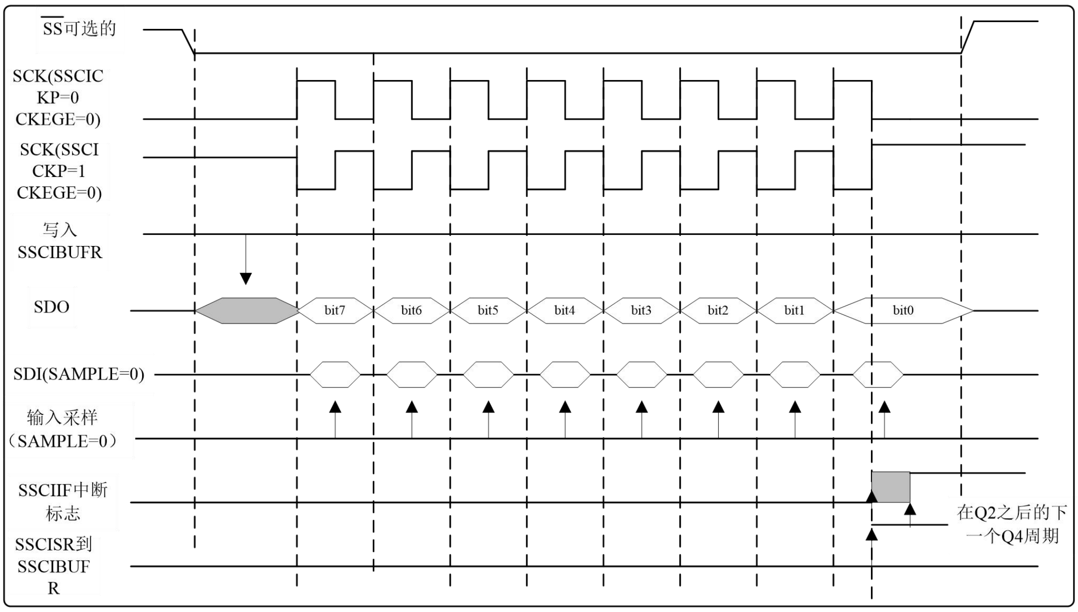


图10.32 SPI模式时序图（从动模式且CKEGE=0）

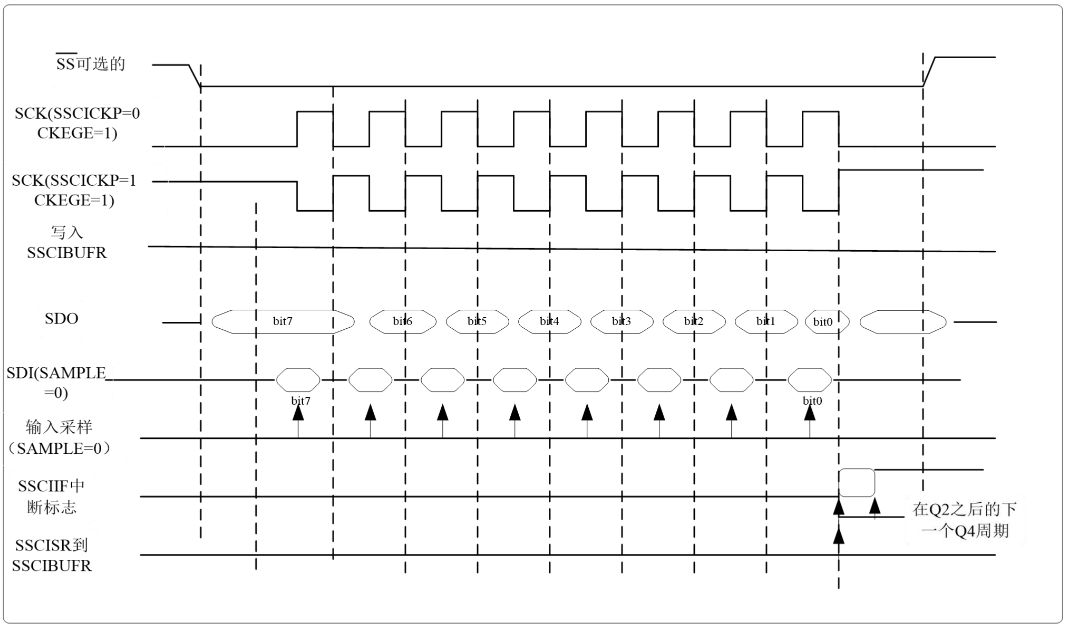


图 10.33 SPI 模式时序图（从动模式且 CKEGE=1）

10.5.6 休眠模式和复位

休眠模式下的工作

在主控模式下，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。

10.5.7 SPI 四种工作模式设置

10.5.7.1 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=SCLK/4
- 0001:SPI 主控方式，时钟=SCLK/16
- 0010:SPI 主控方式，时钟=SCLK/64

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；

10.5.7.2 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；

10.5.7.3 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=SCLK/4
- 0001:SPI 主控方式，时钟=SCLK/16
- 0010:SPI 主控方式，时钟=SCLK/64

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；

10.5.7.4 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；

11 通用全双工/ 半双工收发器（USART）

11.1　系统概述

KF8F3110/12/20/22/30/32 包含 1 个通用全双工/半双工收发器 USART。

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、接收单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部互联网络（Local Interconnect Newxork,LIN）总线系统。

11.2　USART 功能引脚

USART 模块可以通过 PINSET 寄存器的 UARTPIN 位选择相关功能引脚位置。

当 UARTPIN=0 时，RX/DT 在 P1.4 脚，TX/CK 在 P1.6 脚；

当 UARTPIN=1 时，RX/DT 在 P0.0 脚，TX/CK 在 P0.1 脚。

注：14 或 16 引脚使用 USART 功能时 UARTPIN 必须配置为 1。

11.3　相关寄存器

表 11-1 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
156H	PINSET	SSCIPIN	SDOPIN	SSPIN	UARTPIN	PWMPIN	-	-	-
0BH	INTCTL	AIE/ AIEH	PUIE/ AIEL	T0IE	INT0IE	P0IE	T0IF	INT0IF	P0IF
0DH	EIF2	T3IF	C2IF	RCIF	TXIF	-	CCPIF	BCLIF	SSCIIF
2DH	EIE2	T3IE	C2IE	RCIE	TXIE	-	CCPIE	BCLIE	SSCIIE
24H	IP2	PT3	PC2	PRC	PTX	-	PCCP	PBCL	PSSCI
120H	RSCTL	SPEN	RX9	SRXEN	CRXEN	ADREN	FRER	OVFER	RX9D
121H	TXSDR	USART 发送数据寄存器							
122H	RXSDR	USART 接收数据寄存器							
123H	BRCTL	ABRDOVF	RCIDLF	SCKPS	BRG16	-	-	WUEN	ABRDEN
124H	TSCTL	CSRS	TX9	TXEN	SYNC	SENDB	HBRG	TXSRS	TX9D
125H	EUBRGL	USART 波特率数据寄存器低字节							
126H	EUBRGH	USART 波特率数据寄存器高字节							

11.4 原理框图

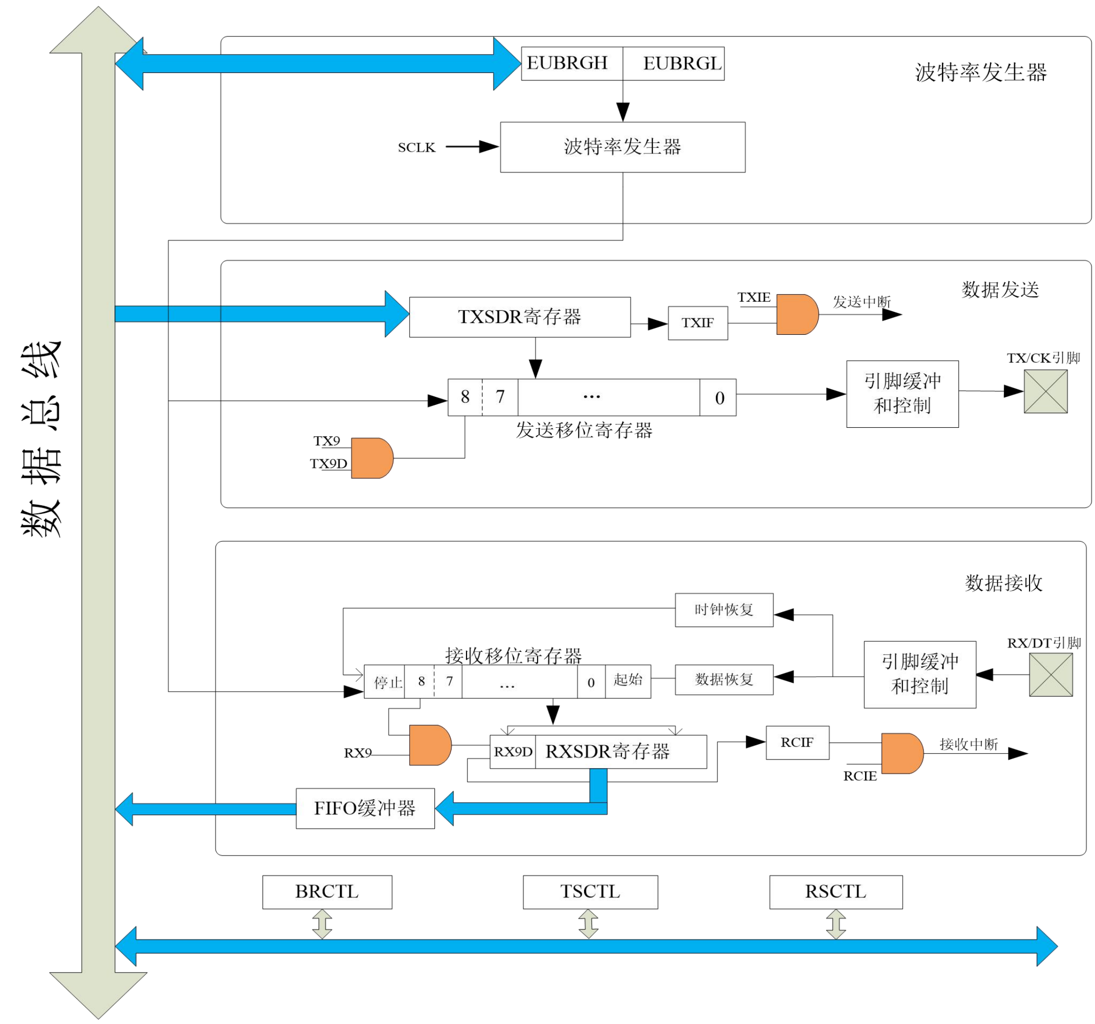


图 11.1 USART 模块框图

从图 11.1 中可以看出，通用全双工/半双工收发器（USART）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 BRCTL、发送状态和控制寄存器 TSCTL、接收状态和控制寄存器 RSCTL。

在波特率发生器这部分，通过设置 SYNC、HBRG 和 BRG16 位来进行选择设置{EUBRGH：EUBRGL}寄存器的值来配置波特率。详细见“11.5 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 TSCTL 来控制实现相应的功能。详见“11.6.1.1”。

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 RSCTL 来控制实现相应的功能。详见“11.6.2.1”。

USART 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。

USART 模块还可实现如下附加功能，从而使其成为局域互联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。

11.5　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。波特率发生器选用 SCLK 作为工作时钟。将 BRCTL 寄存器的 BRG16 位置 1 可选择 16 位模式。

{EUBRGH：EUBRGL}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 TSCTL 寄存器的 HBRG 位和 BRCTL 寄存器的 BRG16 位决定。在半双工同步模式下，HBRG 位被忽略。

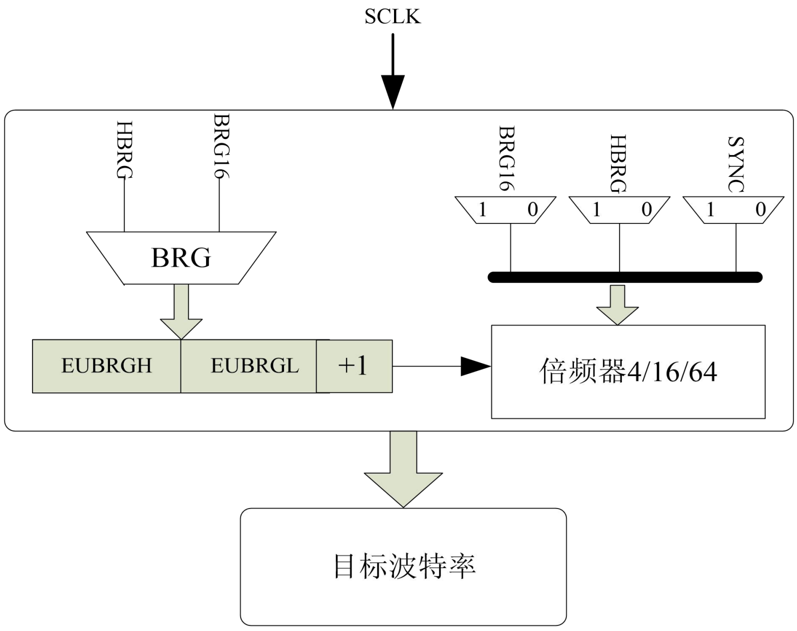


图 11.2　USART 目标波特率产生原理框图

11.5.1　相关寄存器

11.5.1.1　USART 波特率控制寄存器 BRCTL

寄存器**11.1:** BRCTL: 波特率控制寄存器(地址:123H)

复位值 0100 0000	bit7							bit0
	ABRDOVF	RCIDLF	SCKPS	BRG16	-	-	WUEN	ABRDEN
	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

ABRDOVF:　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

RCIDLF:　接收空闲标志位

全双工异步模式:

1 = 接收器空闲
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

SCKPS: 半双工同步时钟极性选择位
全双工异步模式:
1 = 传送反相数据到 TX/CK 引脚
0 = 传送非反相数据到 TX/CK 引脚
半双工同步模式:
1 = 数据在时钟上升沿同步
0 = 数据在时钟下降沿同步

BRG16: 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

WUEN: 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，RCIF 将被置 1，当 RCIF 置 1 后 WUEN 将被自动清 0。
0 = 接收器正常工作
半双工同步模式:
无关位

ABRDEN: 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

图注：R=可读 W=可写 -=未用 U=未实现位

注：1.WUEN、ABDREN 这两位必须在 RSCTL 寄存器的 SPEN 位置 1 后才能写入；
2.在使用 14 或 16 引脚芯片时，由于无 P1.4/P1.6 口，须将 UARTPIN 位置 1，通过 P0.0/P0.1 使用 USART；

11.5.2 波特率的选择

波特率的选择与状态位 SYNC、BRG16 和 HBRG 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。

波特率发生器的时钟是系统主时钟 SCLK，波特率的计算公式为：

$$\text{目标波特率} = \frac{SCLK}{m \times ([\mathrm{EUBRGH} : \mathrm{EUBRGL}] + 1)}$$

其中 m 为倍频器选择，分别为 4、16 和 64，它与状态位 SYNC、BRG16 和 HBRG 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 11-2 列出了 m 选择表。

配置位			BRG/USART 模式	倍频器 m
SYNC	BRG16	HBRG		
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	

表 11-2 倍频器 m 选择表

注：x 为无关位

例 11.1 是确定波特率和波特率误差的计算示例。

例 11.1：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:

$$\text{目标波特率} = \frac{\text{SCLK}}{m\times([\text{EUBRGH}:\text{EUBRGL}]+1)}$$，求解 EUBRGH:EUBRGL。

$$X = \frac{\frac{\text{SCLK}}{\text{目标波特率}}}{64} - 1$$

$$= \frac{\frac{16000000}{9600}}{64} - 1$$

$$= 25.042$$

$$= 25\text{十进制数}$$

$$= 19\text{H 十六进制数}$$

$$\text{计算波特率} = \frac{16000000}{64(25+1)}$$

$$= 9615$$

$$\text{误差} = \frac{\text{计算波特率} - \text{目标波特率}}{\text{目标波特率}}$$

$$= \frac{(9615-9600)}{9600}$$

$$= 0.16\%$$

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，见附录 4 的表格所示。

使用高波特率（HBRG＝1）或 16 位 BRG（BRG16＝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:EUBRGL}寄存器对将导致 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 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。

11.5.3 自动波特率检测

USART 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 输入提供时钟信号，而是由 RX 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 寄存器的 ABRDEN 位置 1 将启动自动波特率校验序列（图 11.3）。当发生 ABRD 序列时，USART 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 使用 BRG 计数器时钟递增计数，如表 11-4 所示。在第 8 位周期的末尾将在 RX 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 和 EUBRGL 寄存器中，ABRDEN 位被自动清 0 而 RCIF 中断标志被置 1。要清除 RCIF 中断，需要读取 RXSDR 中的值。RXSDR 的内容应该被丢弃。校准不使用 EUBRGH 寄存器的模式时，用户可通过查询 EUBRGH 寄存器中的 0X00 验证 EUBRGL 寄存器是否未溢出。

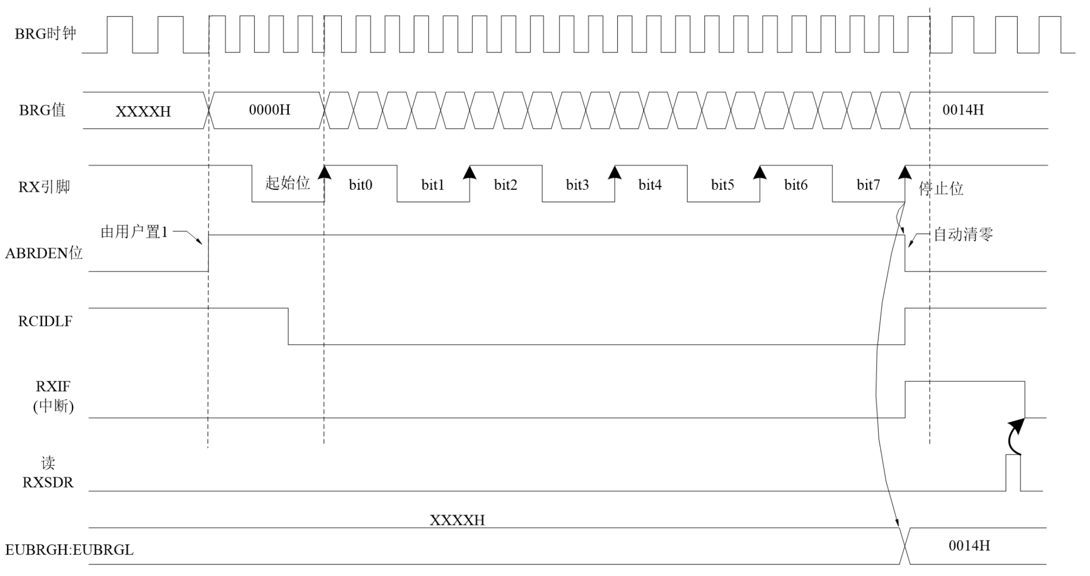
BRG 自动波特率时钟由 BRG16 和 HBRG 位决定，如表 11-4 所示。在 ABRD 期间，EUBRGH 和 EUBRGL 寄存器共同用作 16 位计数器，这与 BRG16 位的设置无关。在校准波特率周期时，EUBRGH 和 EUBRGL 寄存器的定时频率为 BRG 基时钟频率的 1/8。得到的字节测量结果为全速下的平均位时间。

注：1、如果WUEN位和ABRDEN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:EUBRGL 寄存器对的值中减去1。

表 11-4 BRG 计数器时钟速率

BRG16	HBRG	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，EUBRGL和EUBRGH寄存器都用作16位计数器，与BRG16的设置无关。



注 1： ABRD 序列期间，要求将 EUSART 模块在异步模式下工作。

图 11.3　自动波特率检测

11.5.4　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 RX/DT 线上活动时唤醒控制器。只有 USART 工作在异步模式下时才可以使用该功能。

通过将 BRCTL 寄存器的 WUE 位置 1，使能自动唤醒功能，置 1 后，将禁止 RX/DT 上的正常序列的接收，并且 USART 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 RX/DT 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 RCIF 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 RXSDR 寄存器清除中断条件。

当 RX 线从低电平向高电平转换时，WUE 会被清零，这就意味着“同步间隔”事件结束。此时，USART 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间被解释为唤醒事件，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） WUE 位

唤醒事件会产生接收中断，将 RCIF 位置 1。在 RX/DT 的上升沿，由硬件清零 WUE 位。通过在软件中读 RXSDR 寄存器清除中断条件。

要保证没有丢失数据，在将 WUE 位置 1 前，应检查 RCIDL 位，以验证是否正在进行接收。如果不是在进行接收，则可将 WUE 置 1，时器件立即进入休眠模式。

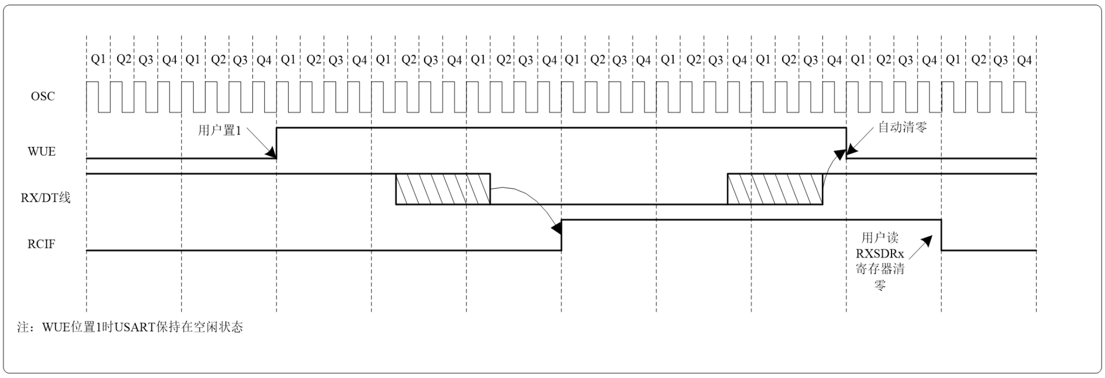


图 11.4 正常工作下的自动唤醒时序

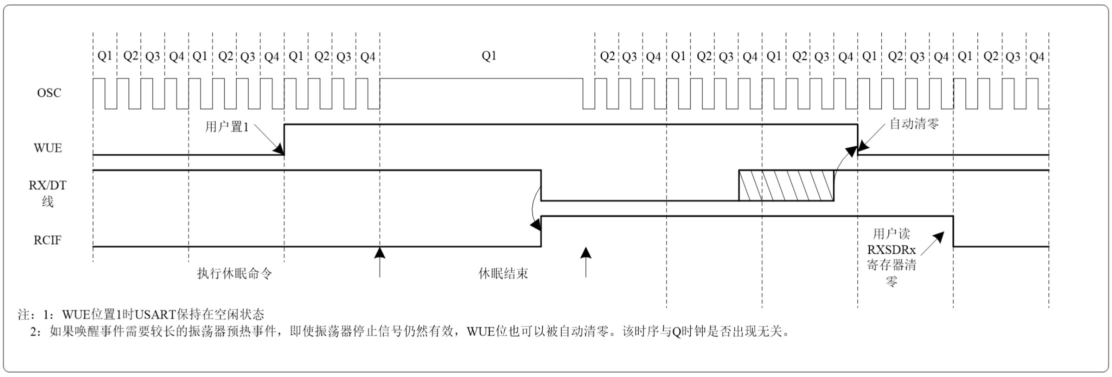


图 11.5 休眠模式下的自动唤醒时序

11.5.5 兼容 LIN 总线的间隔符时序

USART 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 值 1，然后再将 SENDB 位置 1（SENDB 位需要在 TXEN 位为 1 的时候才能写入），之后对 TXSDR 寄存器的写操作将会启动间隔字符的发送。此时写入 TXSDR 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 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 寄存器中的 TXSRS 位则表明发送正在进行还是处于空闲状态。

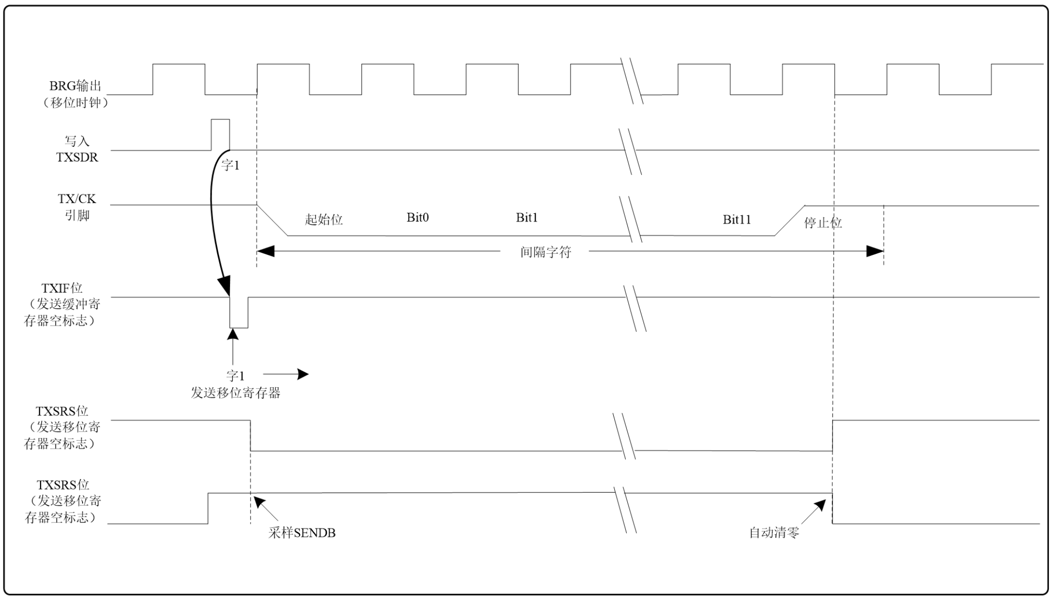


图 11.6 发送间隔字符时序

11.5.5.1 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 USART 配置为所需的模式
2、 将 TXEN 位置 1
3、 将 SENDB 位置 1，设置间隔字符（SENDB 需要在 TXEN 为 1 的时候才能写入）
4、 将无效字符装入 TXSDR 寄存器，启动发送（该值会被忽略）
5、 将 55H 写入 TXSDR 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 SENDB 位复位。开始发送同步字符

当 TXIF 指示 TXSDR 为空时，下一个数据会写入 TXSDR 中。

11.5.5.2 接收间隔字符

USART 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 RSCTL 寄存器的 FRER 位和 RXSDR 指示接收的数据。此方法须将波特率发生器初始化成预期的波特率。当

- RCIF 位置 1
- FRER 位置 1
- RXSDR 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，USART 将采样 RX/DT 上电平的下两个跳变，产生一个 RCIF 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 USART 进入休眠模式之前，将 BRCTL 寄存器的 ABDEN 位置 1.

11.6 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 11.7 所示。

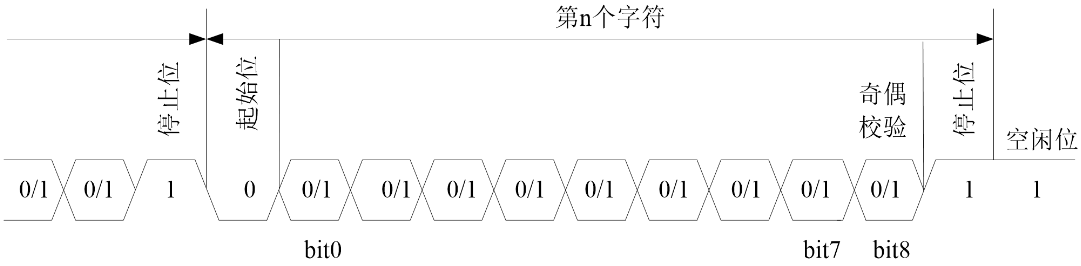


图 11.7 全双工异步通信数据结构图

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。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，只要再接收到 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/(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 11-2 了解波特率配置示例。

USART 首先发送和接收低位。USART 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）。

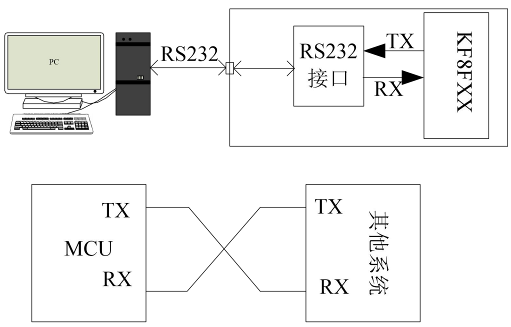


图 11.8 串口通信原理框图

11.6.1 USART 全双工发送操作

USART 全双工异步发送操作通过 USART 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 TXSDR 发送缓冲寄存器获取数据。

通过配置如下三个控制位使能 USART 发送器，以用于全双工异步操作：

1) TXEN= 1

2) SYNC = 0

3) SPEN= 1

假设所有其它 USART 控制位处于其默认状态。将 TSCTL 寄存器的 TXEN 位置 1，使能 USART 发送器电路。将 TSCTL 寄存器的 SYNC 位清 0，将 USART 配置用于全双工异步操作。将 RSCTL 寄存器的 SPEN 位置 1，使能 USART 并自动将 TX/CK 的 I/O 引脚配置为输出引脚。如果与模拟外设共用 TX/CK 引脚，必须清 0 相应的 ANSEL 位禁止模拟 I/O 功能。

注： 1、无论相关的TRI位的状态如何以及USART接收器使能与否，将SPEN位置1会自动将RX/DT I/O引脚配置为输入引脚。可以通过普通端口读RX/DT引脚数据，但却无法使用该端口锁存输出数据。

2、如果TXEN 使能位置1，TXIF 发送器中断标志位会置1。

11.6.1.1 发送状态和控制寄存器 TSCTL

寄存器**11.2:**TSCTL: 发送状态和控制寄存器(地址:124H)

复位值 0000 0010	bit7							bit0
	CSRS	TX9	TXEN	SYNC	SENDB	HBRG	TXSRS	TX9D
	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

CSRS: 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)

TX9: RS485 发送使能位
- 1 = 选择 RS485 发送
- 0 = 选择 8 位发送

TXEN: 发送使能位
- 1 = 使能发送
- 0 = 禁止发送

SYNC: 串行通信模式选择位
- 1 = 半双工同步模式
- 0 = 全双工异步模式

SENDB: 发送间隔字符位
全双工异步模式:
1 = 在下一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

HBRG: 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

TXSRS: 发送移位寄存器状态位
1 = 发送移位空
0 = 发送移位满

TX9D: 发送数据的第 9 位
可以是地址/数据位或奇偶校验位。

图注：R=可读 W=可写 -=未用 U=未实现位

注： 在同步模式下， SRCEN/CRCEN 可改写 TREN。

11.6.1.2 发送数据

向 TXSDR 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，TXSDR 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 TXSDR 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，TXSDR 中待处理的数据将被传输到发送移位寄存器。当数据从 TXSDR 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 USART 发送器且 TXSDR 中没有待发送数据，就将 EIF2 寄存器的 TXIF 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 TXSDR 中有排队等待发送的新字符时，TXIF 位才处于清 0 状态。写 TXSDR 时，不立即清 0 TXIF 标志位。TXIF 在写指令后的第 2 个指令周期清 0。在写 TXSDR 后立即查询 TXIF 会返回无效结果。TXIF 为只读位，不能由软件置 1 或清 0。

可通过将 EIE2 寄存器的 TXIE 中断允许位置 1 允许 TXIF 中断。然而，只要 TXSDR 为空，不管 TXIE 允许位的状态如何都会将 TXIF 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 TXIE 位置 1。当将待发送的最后一个字符写入 TXSDR 后，将 TXIE 中断允许位清 0。

TSCTL 寄存器的 TXSRS 位指示发送移位寄存器的状态。TXSRS 位为只读位。当发送移位寄存器为空时，TXSRS 位被置 1，当有字符从 TXSDR 传输到发送移位寄存器时，TXSRS 被清 0。TXSRS 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。

11.6.1.3　全双工发送的设置

1、　初始化EUBRGH和EUBRGL这对寄存器以及HBRG和BRG16位，以获得所需的波特率（见第11.5节“波特率发生器”）。
2、　通过将SYNC位清0并将SPEN位置1使能全双工异步串口。
3、　如果需要9位发送，将TX9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。
4、　将TXEN控制位置1，使能发送；这将导致TXIF中断标志位置1。
5、　如果需要中断，将EIE2寄存器中的TXIE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。
6、　若选择发送9位数据，第9位应该被装入TX9D数据位。
7、　将8位数据装入TXSDR寄存器开始发送数据。

```
例11.2 发送数据
        SET     TSCTL,TXEN
        MOV     R1, UART_TEMP
        MOV     TXSDR, R1
        NOP
        NOP
        NOP
        JB      TSCTL,TXSRS        ;判断是否发生完毕？
        JMP $-3
```

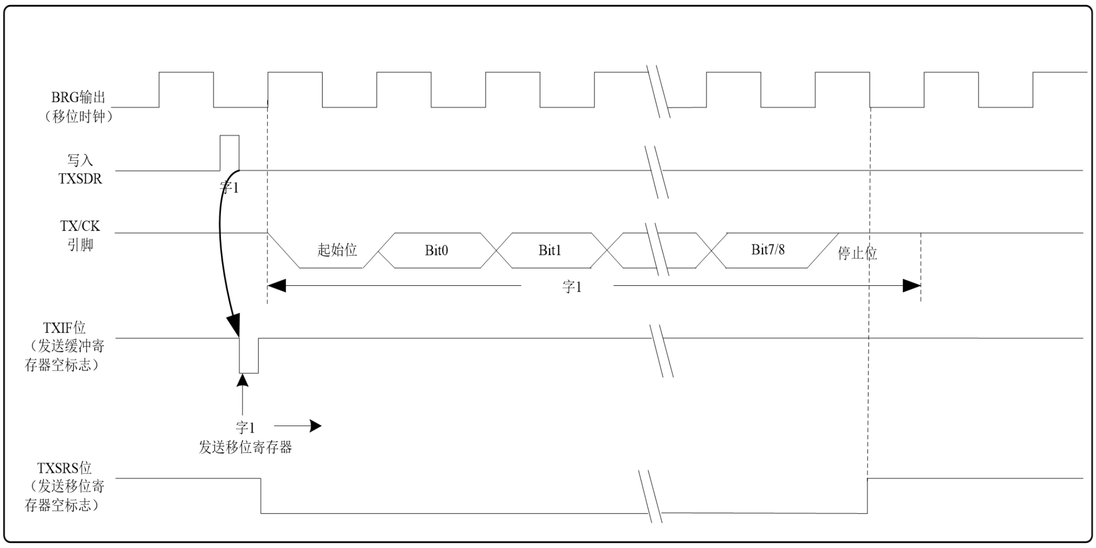


图 11.9 全双工异步发送（两字符间有空闲位）

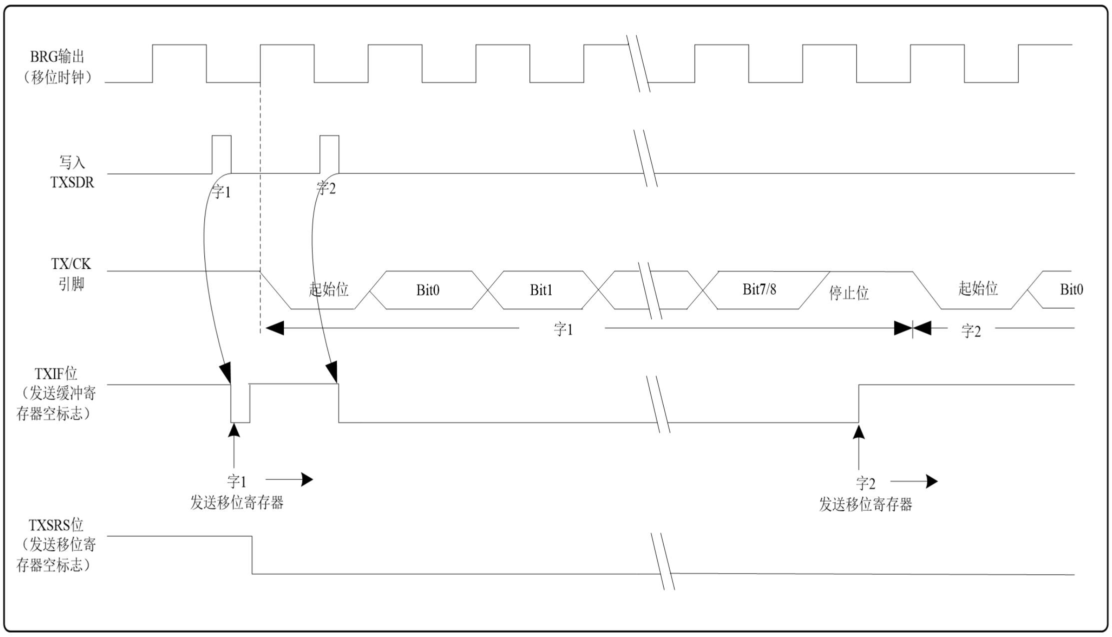


图 11.10 全双工异步发送（一字符接一字符）

11.6.2 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。图 11.1 给出了接收器的框图。在 RX/DT 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（ReceiveShift 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 USART 接收器。FIFO 缓冲器和 RSR 寄存器不能直接由软件访问。通过 RXSDR 寄存器访问接收到的数据。

通过配置如下三个控制位使能 USART 接收器，以用于全双工异步接收操作。

- CRXEN= 1
- SYNC = 0
- SPEN= 1

假设所有其它 USART 控制位都处于默认状态。将 RSCTL 寄存器的 CRXEN 位置 1，使能 USART 接收器电路。将 TSCTL 寄存器的 SYNC 位清 0，配置 USART 以用于全双工异步操作。将 RSCTL 寄存器的 SPEN 位置 1，使能 USART 并自动将 RX/DT 引脚配置为输入引脚。如果 RX/DT 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 位置1，TX/CK 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/CK 引脚用作通用输出引脚。

11.6.2.1 接收状态和控制寄存器 RSCTL

寄存器11.3: RSCTL: 接收状态和控制寄存器(地址:120H)

复位值 0000 000x	bit7							bit0
	SPEN	RX9	SRXEN	CRXEN	ADREN	FRER	OVFER	RX9D
	R/W	R/W	R/W	R/W	R/W	R	R	R

SPEN: 串行口使能位
1 = 使能串行口(将 RX/DT 和 TX/CK 引脚配置为串行口引脚)
0 = 禁止串行口(保持为复位状态)

RX9: 9 位接收使能位
1 = 选择 9 位接收
0 = 选择 8 位接收

SRXEN: 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

CRXEN: 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 CRXEN 清 0(CRXEN 改写 SRXEN)
0 = 禁止连续接收

ADREN: 地址检测使能位
RS-485 全双工异步模式(RX9 = 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(RX9=0):
无关位

FRER: 帧错误位
1 = 帧错误(可通过读 RXSDR 寄存器刷新该位，并接收下一个有效字节)
0 = 无帧错误

OVFER: 溢出错误位
1 = 溢出错误(可通过清 0 CRXEN 位来清 0 该位)
0 = 无溢出错误

RX9D: 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。

图注：R=可读 W=可写 -=未用 U=未实现位

11.6.2.2 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。详见第 11.3.2.3 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 USART 的接收 FIFO 缓冲器并将 EIE2 寄存器的 RCIF 中断标志位置 1。通过读 RXSDR 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。请参见第11.3.2.3 节“接收错误”获得有关溢出错误的更多相关信息。

只要使能 USART 接收器且在接收 FIFO 缓冲器中没有未读数据，EIF2 寄存器中的 RCIF 中断标志位就会清 0。RCIF 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 RCIF 中断：

- ♦ EIE2 寄存器的 RCIE 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 RCIF 中断标志位置 1。

11.6.2.3 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 RSCTL 寄存器的 FRER 位获取帧错误状态。FRER 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 RXSDR 寄存器之前读 FRER 位。
FRER 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（FRER = 1）并不会阻止接收更多的字符。无需清 0 FRER 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 RSCTL 寄存器的 SPEN 位会复位 USART，并强制清 0 FRER 位。清 0 RSCTL 寄存器的 CRXEN 位不影响 FRER 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RCSDR不会清零FRER。

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，RSCTL 寄存器的 OVFER 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 RSCTL 寄存器的 CRXEN 位或通过清 0 RSCTL 寄存器的 SPEN 位使 USART 复位来清除错误。

11.6.2.4 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 RSCTL 寄存器的 ADREN 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，从而使 RCIF 中断标志位置 1。所有其它字符将被忽略。

由用户软件判断接收到的地址字符是否与其匹配。如果匹配，用户软件必须在下一个停止位产生之前，清 0 ADREN 位以禁止地址检测。当用户软件检测到信息的末尾（由所使用的信息协议判断）时，由软件将 ADREN 位置 1，从而使接收器返回地址检测模式。

11.6.2.5 全双工接收的设置

1、 初始化{EUBRGH:EUBRGL}这对寄存器以及 HBRG 和 BRG16 位，以获得所需的波特率（见第 11.5 节“波特率发生器（BRG）”）。
2、 将 SPEN 位置 1，使能串行端口。必须清 0 SYNC 位以执行全双工异步操作。
3、 如果需要中断，将 EIE2 寄存器中的 RCIE 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。
4、 如果需要接收 9 位数据，将 RX9 位置 1。
5、 将 CRXEN 位置 1 使能接收。
6、 当一个字符从 RSR 传输到接收缓冲器时，将 RCIF 中断标志位置 1。如果 RCIE 中断允许位也置 1 还将产生中断。
7、 读 RSCTL 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）。
8、 读 RXSDR 寄存器，从接收缓冲器获取接收到的 8 个低数据位。
9、 如果发生溢出，通过清 0 CRXEN 接收器使能位清 0 OVFER 标志。

```
例11.3 接收
LOOP_USART
          NOP
          NOP
          SET     RSCTL,CRXEN           ;使能接收器
          JNB     RSCTL,OVFER           ;检测溢出错误
          CLR     RSCTL, CRXEN
          JB      EIF2, RCIF            ;接收数据的标志
          JMP     LOOP_USART
          CLR     EIF2, RCIF
          MOV     R0, RXSDR
          MOV     UART_TEMP, R0         ;存储接收到的数据
```

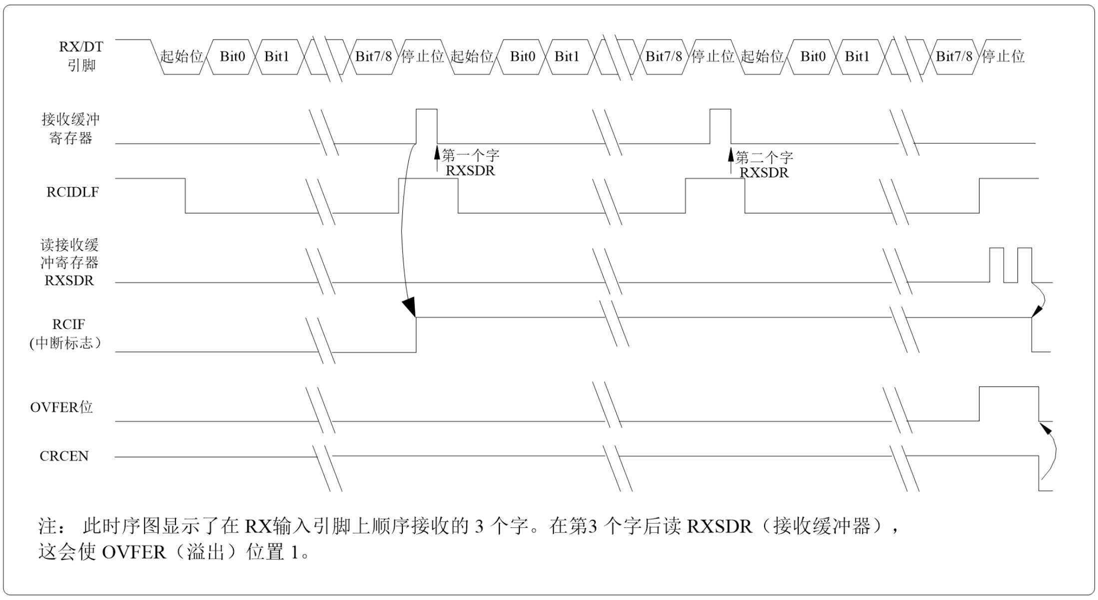


图 11.11 全双工异步接收

11.6.3 RS-485 发送/接收

USART 支持 RS-485 模式 9 位发送。当 TSCTL 寄存器的 TX9 位置 1 时，USART 将移出每个待发送字符的 9 位。TSCTL 寄存器的 TX9D 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 TXSDR 之前，写 TX9D 数据位。在写入 TXSDR 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。请参见第 11.6.2.4 节“地址检测”获得有关地址模式的更多信息。

USART 支持 RS-485 模式 9 位接收。将 RSCTL 寄存器的 RX9 位置 1 时，USART 将接收到的每个字符的 9 位移入 RSR。RSCTL 寄存器的 RX9D 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 RXSDR 中的低 8 位之前，读取 RX9D 数据位。

11.6.3.1 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 EUBRGH 和 EUBRGL 这对寄存器以及 HBRG 和 BRG16 位，以获得所需的波特率（见第 11.5 节“波特率发生器”），设置波特率控制寄存器 BRCTL 的 UPIN 位，选择相应的 I/O 口。

2、 将 SPEN 位置 1，使能串行端口。必须清 0 SYNC 位以执行全双工异步操作。

3、 如果需要中断，将 EIE2 寄存器中的 RCIE 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 PRC 均置 1。

4、 将 RX9 位置 1，使能 9 位数据接收。

5、 将 ADREN 位置 1，使能地址检测。

6、 将 CRXEN 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 RCIF 中断标志位置 1。如果

RCIE 中断允许位也置 1 还将产生中断。

8、 读 RSCTL 寄存器获取错误标志位。第 9 个数据位始终置 1。

9、 读 RXSDR 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 CRXEN 接收器使能位清 0 OVFER 标志。

11、如果是对当前器件寻址，将 ADREN 位清 0 以允许所有接收到的数据进入接收缓冲器并产生中断。

11.6.4 全双工操作时钟的精确性

内部振荡器模块输出（OSCCTL）在出厂时做了校准。但是，VDD 或温度变化时 OSCCTL 频率有可能漂移，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（见第 11.5.3 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。

11.7 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器件和一个或多个从动器件的系统中。主控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。USART 既可以作为主控器件，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。

11.7.1 USART 半双工主控模式

下列位用来将 USART 配置为半双工同步主控操作：

- SYNC = 1
- CSRS = 1
- SRXEN = 0（用于发送）；SRXEN = 1（用于接收）
- CRXEN = 0（用于发送）；CRXEN = 1（用于接收）
- SPEN= 1

将 TSCTL 寄存器的 SYNC 位置 1，可将 USART 配置用于半双工同步操作。将 TSCTL 寄存器的 CSRS 位置 1，将器件配置为主控器件。将 RSCTL 寄存器的 SRXEN 和 CRXEN 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 RSCTL 寄存器的 SPEN 位置 1，使能 USART。如果 RX/DT 或 TX/CK 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。配置为主控器件的器件在 TX/CK 引脚发送时钟信号。当 USART 被配置为半双工同步发送或接收操作时，TX/CK 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 BRCTL 寄存器的 SCKPS 位选择时钟极性。将 SCKPS 位置 1 将时钟空闲状态设置为高电平。当 SCKPS 位置 1 时，数据在每个时钟的下降沿发生改变。清 0 SCKPS 位，将时钟空闲状态设置为低电平。当清 0 SCKPS 位时，数据在每个时钟的上升沿发生改变。

11.7.1.1 半双工主控发送

由器件的 RX/DT 引脚输出数据。当 USART 配置为半双工同步主控发送操作时，器件的 RX/DT 和 TX/CK 输出引脚自动使能。

向 TXSDR 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 TXSDR 中，直到发送完前一字符的停止位为止。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中移出，则 TXSDR 中的数据会被立即传输到发送移位寄存器。当字符从 TXSDR 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器中，因此用户不能直接访问它。

半双工主控发送设置：

1、 初始化 EUBRGH 和 EUBRGL 这对寄存器以及 HBRG 和 BRG16 位，以获得所需的波特率（见第 11.5 节“波特率发生器”）。
2、 将 SYNC、SPEN 和 CSRS 位置 1，使能半双工同步主控串行端口。
3、 将 SRXEN 和 CRXEN 位清 0，禁止接收模式。
4、 将 TXEN 位置 1 使能发送模式。
5、 如果需要发送 9 位字符，将 TX9 置 1。
6、 若需要中断，将 EIE2 寄存器的 TXIE 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 PTX 均置 1。
7、 如果选择发送 9 位字符，应该将第 9 位数据装入 TX9D 位。
8、 通过将数据装入 TXSDR 寄存器启动发送。

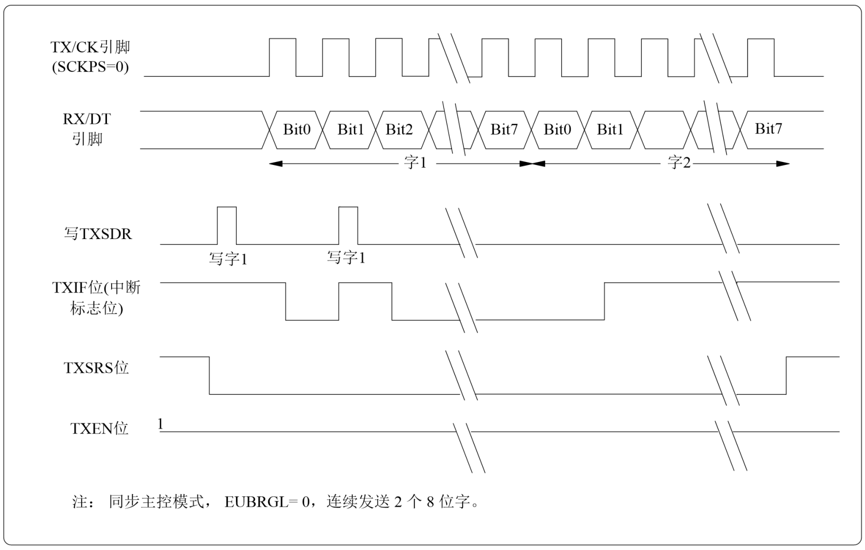


图 11.12 半双工同步发送（SCKPSx=0）

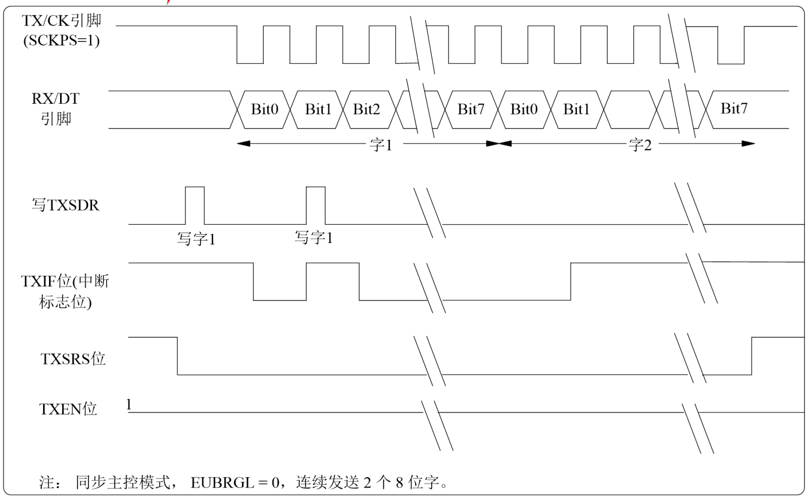


图 11.13 半双工同步发送(SCKPS=1)

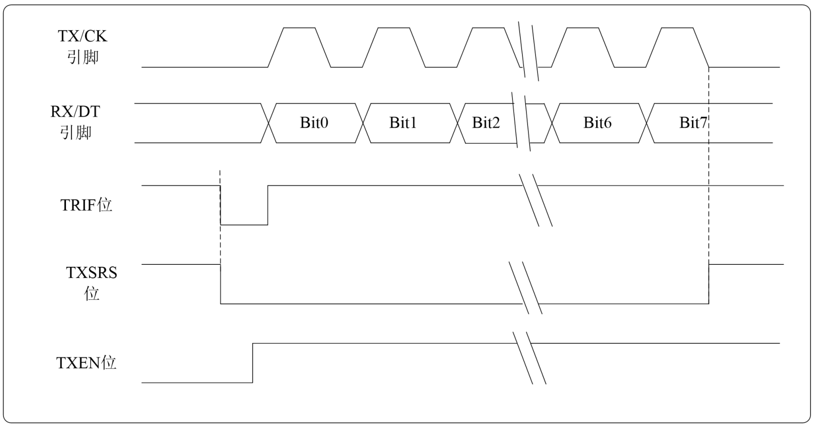


图 11.14 半双工同步发送（通过 TXEN）

11.7.1.2 半双工主控接收

在 RX/DT 引脚接收数据。当 USART 配置为半双工同步主控接收时，自动禁止器件的 RX/DT 引脚的输出驱动器。在半双工同步模式下，将单字接收使能位（RSCTL 寄存器的 SRXEN 位）或连续接收使能位（RSCTL 寄存器的 CRXEN 位）置 1 使能接收。

当将 SRXEN 置 1，CRXEN 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 SRXEN 位。当 CRXEN 置 1 时，将产生连续时钟，直到清 0 CRXEN 为止。如果 CRXEN 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 SRXEN 和 CRXEN 都置 1，则当第一个字符传输完成时，SRXEN 位被清 0，CRXEN 保持。

将 SRXEN 或 CRXEN 位置 1，启动接收。在 TX/CK 时钟引脚信号的下降沿采样 RX/DT 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 RCIF 位置 1，字符自动移入到 2 字节的接收 FIFO 缓冲器。接收 FIFO 缓冲器中最顶端字符的低 8 位可通过 RXSDR 读取。只要接收 FIFO 缓冲器中仍有未读字符，则 RCIF 位就保持置 1 状态。

半双工同步数据传输使用与数据线同步的独立时钟线。配置为从模式的器件接收 TX/CK 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，TX/CK 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 RXSDR 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，RSCTL 寄存器的 OVFER 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 OVFER 位清 0。如果发生溢出时，SRXEN 位为置 1 状态，CRXEN 位为清 0 状态，则通过读 RXSDR 寄存器清除错误。如果溢出时，CRXEN 为置 1 状态，则可以清 0 RSCTL 寄存器的 CRXEN 位或清 0 SPEN 位以复位 USART，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 EUBRGH:EUBRGL 寄存器。按需要将 HBRG 和 BRG16 位置 1 或清 0，以获得所需的波特率。
2、 将 SYNC、SPEN 和 CSRS 位置 1 使能半双工同步主控串行端口。
3、 确保将 CRXEN 和 SRXEN 位清 0。
4、 如果使用中断，将 INTCTL 寄存器的 AIE 和 PUIE 位置 1，并将 EIE2 寄存器的 RCIE 位也置 1。如果使用优先级中断，则将 IPEN 和 PRC 均置 1。
5、 如果需要接收 9 位字符，将 RX9 位置 1。
6、 将 SRXEN 位置 1，启动接收，或将 CRXEN 位置 1 使能连续接收。
7、 当字符接收完毕后，将 RCIF 中断标志位置 1。如果允许位 RCIE 置 1，还会产生一个中断。
8、 读 RSCTL 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 RXSDR 寄存器获取接收到的 8 位数据。
10、如果产生溢出错误，清 0 RSCTL 寄存器的 CRXEN 位或清 0 SPEN 以复位 USART 来清除错误。

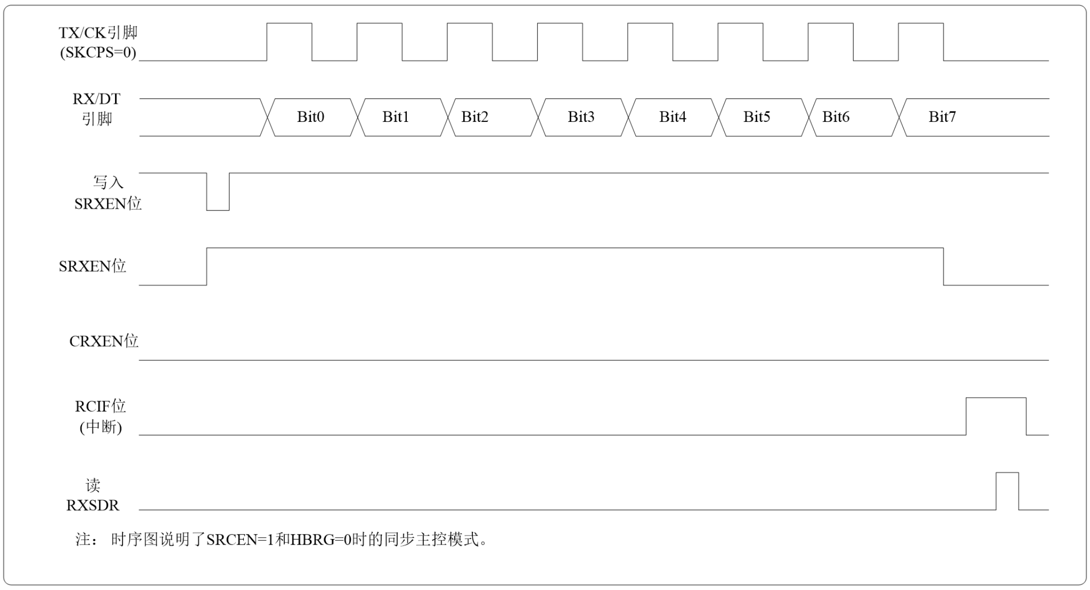


图 11.15 半双工同步接收（主控模式，SRXEN=1，SCKPS=0 ）

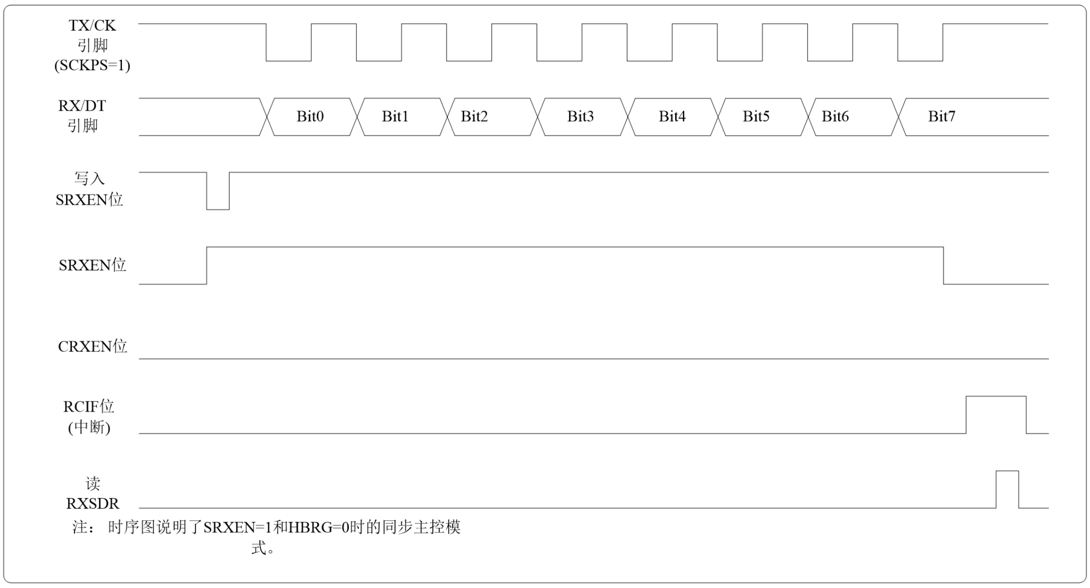


图 11.16 半双工同步接收（主控模式，SRXEN=1，SCKPS=1）

11.7.2 USART 半双工从动模式

下列位用来将 USART 配置为半双工同步从动操作：

- SYNC=1
- CSRS=0
- SRXEN=0（用于发送）； SRXEN = 1（用于接收）
- CRXEN=0（用于发送）； CRXEN = 1（用于接收）
- SPEN=1

将 TSCTL 寄存器的 SYNC 位置 1，可将器件配置用于半双工同步操作。将 TSCTL 寄存器的 CSRS 位置 1，将器件配置为从动器件。将 RSCTL 寄存器的 SRXEN 和 CRXEN 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 寄存器的 SPEN 位置 1，使能 USART。如果 RX/DT 或 TX/CK 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。

11.7.2.1 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（见第 11.7.1.1 节“半双工主控发送”）。

如果向 TXSDR 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 TXSDR 寄存器中。
3. TXIF 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 TXIF 置 1。
5. 如果 PUIE 和 TXIE 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：

2. 将 SYNC 和 SPEN 位置 1 并将 CSRSx 位清 0。
3. 将 CRXEN 和 SRXEN 位清 0。
4. 如果使用中断，将 INTCTL 寄存器的 AIE 和 PUIE 位置 1，并将 EIE2 寄存器的 TXIE 位也置 1。如果使用优先级中断，则将 IPEN 和 PTXx 均置 1。
5. 如果需要发送 9 位数据，将 TX9 位置 1。
6. 将 TXEN 位置 1 使能发送。
7. 若选择发送 9 位数据，将最高位写入 TX9D 位。
8. 将低 8 位数据写入 TXSDR 寄存器开始传输。

11.7.2.2 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（见第 11.7.1.2 节“半双工主控接收”）。

- 休眠模式
- CRXEN 位总是置 1，因此接收器不能进入空闲状态。
- SRXEN 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 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 RXSDR 寄存器。如果将 RCIE 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：

2. 将 SYNC 和 SPEN 位置 1 并将 CSRS 位清 0。

3. 如果使用中断，将 INTCTL 寄存器的 AIE 和 PUIE 位置 1，并将 EIE2 寄存器的 RCIE 位也置 1。如果使用优先级中断，则将 IPEN 和 PRC 均置 1。
4. 如果需要接收 9 位字符，将 RX9 位置 1。
5. 将 CRXEN 位置 1，使能接收。
6. 当接收完成后，将 RCIF 位置 1。如果 RCIE 已置 1，还会产生一个中断。
7. 如果使能 9 位模式，从 RSCTL 寄存器的 RX9D 位获取最高位。
8. 读 RXSDR 寄存器，从接收 FIFO 缓冲器获取接收到的 8 个低数据位。
9. 如果产生溢出错误，清 0 RSCTL 寄存器的 CRXEN 位或清 0 SPEN 位以复位 USART 来清除错误。

11.7.3 USART 半双工 RS-485 模式

请参考第 11.6.3 节 RS-485 发送/接收。

12 运算放大器模块

KF8F3112/22/32 具有 1 个 4 路正向输入端可选的带校准的运算放大器，运放模块通过与外部元件连接实现其放大等作用，内部框图如图 12.1 所示。

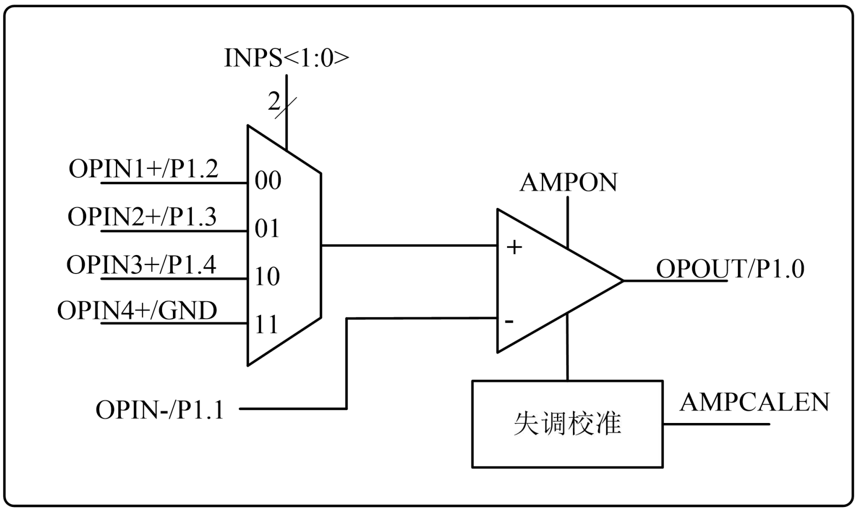


图 12.1 运放内部框图

12.1 与运放有关的寄存器

表 12-1 与运放相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
1BH	AMPCTL	AMPCALD	-	INPS1	INPS0	-	-	AMPCALEN	AMPON
20H	AMPDT	AMPDT7	AMPDT6	AMPDT5	AMPDT4	AMPDT3	AMPDT2	AMPDT1	AMPDT0

12.1.1 运放控制寄存器（AMPCTL）

寄存器12.1: AMPCTL: 运放控制寄存器(1BH)

复位值 0-00 0000	bit7							bit0
	AMPCALD	-	INPS1	INPS0	-	-	AMPCAL EN	AMPON
	R	U	R/W	R/W	R/W	R/W	R/W	R/W

AM

PCALD: 运放输出结果

1 = 运放输出为1

0 = 运放输出为0

INPS<1:0>: 运放正输入端选择位

00 = 将P1.2引脚配置为运放正输入端

01 = 将P1.3引脚配置为运放正输入端

10 = 将P1.4引脚配置为运放正输入端

11 = 运放正输入端接地

AMPCALEN: 运放失调校准使能位

1 = 使能运放失调校准

0 = 禁止运放失调校准

AMPON: 运放使能位
1 = 使能运放
0 = 禁止运放

图注：R=可读 W=可写 -=未用 U=未实现位

12.1.2 运放校验寄存器（AMPDT）

寄存器12.2: AMPDT: 运放校验寄存器(20H)

复位值 1000 0000	bit7							bit0
	AMPDT7	AMPDT6	AMPDT5	AMPDT4	AMPDT3	AMPDT2	AMPDT1	AMPDT0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

AMPDT<7:0>:校验数据位
AMPDTx = 运放校验数据

图注：R=可读 W=可写 -=未用 U=未实现位

12.2 运放使用方法

12.2.1 校验方法

运放失调校准步骤：

1. 设置运放的 IO 端口为模拟端口，OPIN+、OPIN-和 OPOUT 均配置为输入端口；
2. 将 AMPON=1，打开运放，并且延迟 100us；
3. 将 AMPCALEN=1 使能运放失调校准功能，延迟 100us；
4. 读出运放输出结果 AMPCALD（AMPCTL<7>），并写入 AMPDT<7>;
5. 将 AMPDT<6>改写为 1，延迟 100us，读出运放输出结果 AMPCALD;
6. 判断 AMPCALD 和 AMPDT<7>是否相等，相等则 AMPDT<6>保持为 1，不相等则将 AMPDT<6>改写为 0;
7. 同样对 AMPDT<5:3>的每位重复步骤 5 和 6 的操作；
8. AMPDT<2:0>从 000 开始加 1，延迟 100us，读出运放输出结果 AMPCALD;
9. 判断 AMPCALD 与 AMPDT7 是否相等，相等返回步骤 8；不相等，AMPCALEN 清零结束运放失调校准。如果 AMPDT<2:0>加到 111 时还是相等，也结束运放失调校准，开始正常运放工作。

例 12.1 运放失调校准参考程序

```
    ;默认用户已经将运放的OPIN+、OPIN-和OPOUT引脚均配置为输入端口
        SET     AMPCTL, AMPON           ;开启运放,
        CALL    Delay_100US         ;延迟100us，给一个输出缓冲时间
        SET     AMPCTL, AMPCALEN        ;开启自校准
        CALL    Delay_100US         ;延迟
;读出AMPCTL中的AMPCALD位放入AMPDT中的最高位，用于正负失调判断，次高位置1，
其它位设置为全0
        MOV     R0, AMPCTL                  ;读出运放输出的值AMPCALD
        AND     R0,#0X80
        MOV     R5,R0                       ;保存第一次的值
```

```
        MOV     R3,R0
        MOV     R2,#B'01000000'
        MOV     R7,#0X05
AMP_CALIBRATION1
        ORL     R3,R2
        MOV     AMPDT,R3
        CALL    Delay_100US             ;延迟100us
        MOV     R1,AMPCTL
        AND     R1,#0X80
        XOR     R1,R5
        JB      PSW,Z                   ;如果翻转,当前位清零
        XOR     R3,R2
        NOP
        CLR     PSW,CY
        RRC     R2                      ;右移一位
        DECJZ   R7
        JMP     AMP_CALIBRATION1
        MOV     R7,#0x03
AMP_CALIBRATION2
        INC     R3
        MOV     AMPDT,R3
        CALL    Delay_100US             ;延迟100us
        MOV     R1,AMPCTL
        AND     R1,#0X80
        XOR     R1,R5
        JB      PSW,Z
        JMP     AMP_END                 ;校准完成
        DECJZ   R7
        JMP     AMP_CALIBRATION2
AMP_OFFSET_EXIT
        CLR     AMPCTL, AMPCALEN        ;关闭自校准
        NOP
        NOP
        NOP
        CRET
```

12.2.2 使用方法

使用运放时软件的设置：

1. 将运放的输入引脚和输出引脚均设置为输入口；
2. 选则其中一路输入为运放的正向输入端；
3. 打开运放(AMPON =1)。

12.3 复位的影响

当器件复位时，强制将寄存器 AMPCTL 和 AMPT 置为复位状态，运算放大器被“强制”设置为关闭状态。

13 模拟比较器模块

KF8F3110/12/20/22/30/32 含有 2 个模拟比较器 CMP1 和 CMP2；
2 个比较器分别带有电压自校准功能、数字滤波功能和中断；
2 个比较器的正端输入为 IO 端口，负端输入可选择 IO 端口或内部电阻分压电路。

13.1 模拟比较器原理

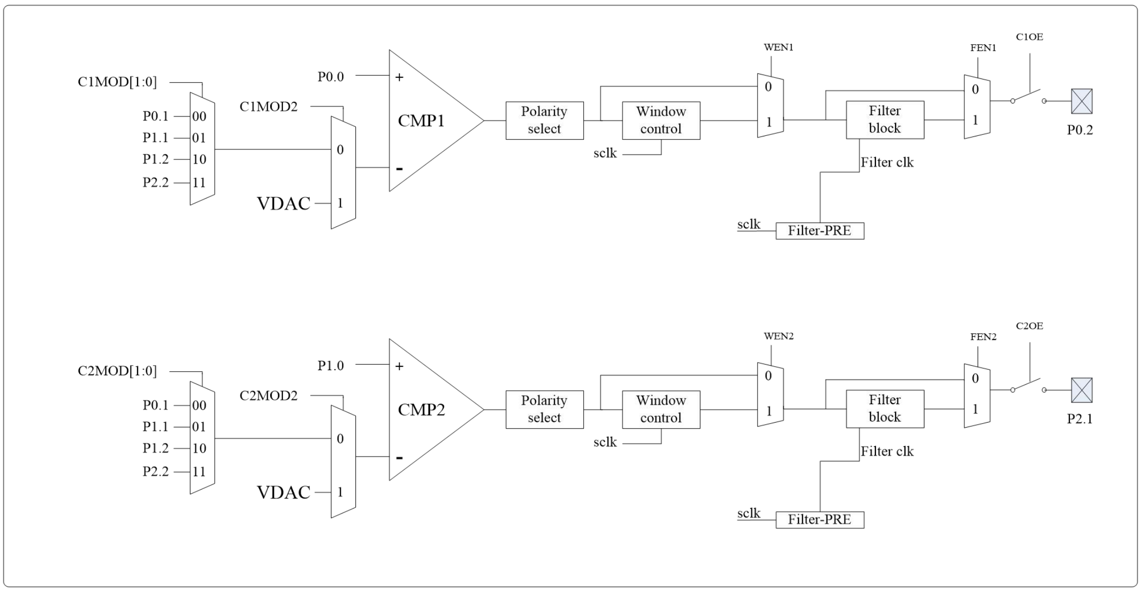


图 13.1 模拟比较器原理框图

13.2 相关寄存器

表 13-1 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
19H	C1CTL	C1OE	C1MOD2	C1MOD1	C1MOD0	C1CALEN	C1CALSET	C1OUT	C1EN
112H	C2CTL	C2M3	C2M2	C2M1	C2M0	C2CALEN	C2CALSET	C2MOD2	C2EN
115H	C1CAL	C1DT7	C1DT6	C1DT5	C1DT4	C1DT3	C1DT2	C1DT1	C1DT0
116H	C2CAL	C2DT7	C2DT6	C2DT5	C2DT4	C2DT3	C2DT2	C2DT1	C2DT0
14AH	C1FILTCTL	INV1	WEN1	FEN1	-	-	F1CNT2	F1CNT1	F1CNT0
14CH	C2FILTCTL	INV2	WEN2	FEN2	-	-	F2CNT2	F2CNT1	F2CNT0
14BH	C1FILTPRE	FP17	FP16	FP15	FP14	FP13	FP12	FP11	FP10
14DH	C2FILTPRE	FP27	FP26	FP25	FP24	FP23	FP22	FP21	FP20
1AH	COUT	-	VCEN	C2OE	C2MOD1	C2MOD0	-	-	C2OUT
2CH	EIE1	EEIE	ADIE	INT2IE	INT1IE	C1IE	PWM2IE	T2IE	T1IE
2DH	EIE2	T3IE	C2IE	RCIE	TXIE	-	CCPIE	BCLIE	SSCIIE
0CH	EIF1	EEIF	ADIF	INT2IF	INT1IF	C1IF	PWM2IF	T2IF	T1IF
0DH	EIF2	T3IF	C2IF	RCIF	TXIF	-	CCPIF	BCLIF	SSCIIF
23H	IP1	PEE	PADC	PINT2	PINT1	PC1	PPWM2	PT2	PT1
24H	IP2	PT3	PC2	PRC	PTX	-	PCCP	PBCL	PSSCI

13.3 CMP1 相关寄存器

13.3.1 比较器 1 控制寄存器 C1CTL

寄存器**13.1:** C1CTL: 比较器1控制寄存器(地址: 19H)

复位值 0000 0000	bit7						bit0
	C1OE	C1MOD2	C1MOD1	C1MOD0	C1CALEN	C1CALSET	C1OUT	C1EN
	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

C1OE: CMP1 输出使能位
- 1 = CMP1 输出到对应引脚，此时需将 P0.2 引脚配置成数字输出端口。
- 0 = 禁止 CMP1 输出

C1MOD2: CMP1 的负输入端选择位
- 1 = 内部电阻分压 VDAC 作为 CMP1 的负输入端
- 0 = IO 口作为 CMP1 的负输入端，具体配置位为 C1MOD<1:0>

注：VDAC 的配置位参考 C2M<3:0>（C2CTL<7:4>）

C1MOD<1:0>: CMP1 的负输入端选择位

C1MOD2=0：
- 00 = P0.1 作为 CMP1 的负输入端
- 01 = P1.1 作为 CMP1 的负输入端
- 10 = P1.2 作为 CMP1 的负输入端
- 11 = P2.2 作为 CMP1 的负输入端

C1MOD2=1：无效

C1CALEN: CMP1 校准使能位
- 1 = 使能比较器 1 的校准功能
- 0 = 禁止比较器 1 的校准功能

C1CALSET: CMP1 校准设置位
- 1 = 将比较器 1 正负端串接 1/2VDD
- 0 = 比较器 1 正负端未串接

C1OUT: CMP1 输出

INV1=0 时
- 1 = C1IN+ > C1IN-
- 0 = C1IN+ < C1IN-

INV1=1 时
- 0 = C1IN+ > C1IN-
- 1 = C1IN+ < C1IN-

C1EN: 比较器 1 使能位
- 1 = 使能比较器 1
- 0 = 关闭比较器 1

图注：R=可读 W=可写 -=未用 U=未实现位

13.3.2 比较器 1 校准寄存器 C1CAL

寄存器13.2: C1CAL: 比较器1校准寄存器 （115H）

复位值 0000 0000	bit7							bit0
	C1DT7	C1DT6	C1DT5	C1DT4	C1DT3	C1DT2	C1DT1	C1DT0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

C1DT<7:0>: 模拟比较器校准数据位

C1DT = 比较器校验数据

图注：R=可读 W=可写 -=未用 U=未实现位

13.3.3 比较器 1 滤波控制寄存器 C1FILTCTL

寄存器13.3: C1FILTCTL: 比较器1滤波控制寄存器 （14AH）

复位值 0000 0000	bit7							bit0
	INV1	WEN1	FEN1	-	-	F1CNT2	F1CNT1	F1CNT0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

INV1: 比较器 1 输出极性选择位

0 = 比较器正常输出：C1IN+>C1IN-=1，C1IN+<C1IN-=0

1 = 比较器输出极性转换：C1IN+>C1IN-=0，C1IN+<C1IN-=1

WEN1: 比较器 1 范围控制使能位

0 = 禁止范围控制功能

1 = 使能范围控制功能

FEN1: 比较器 1 滤波使能位

0 = 禁止比较器 1 滤波功能

1 = 使能比较器 1 滤波功能

F1CNT<2:0>: 比较器 1 滤波采样数量选择位

000 = 不采样，禁止滤波功能

001 = 采样 1 次，并输出结果

010 = 连续 2 次采样结果一致，输出采样值，否则保持；

011 = 连续 3 次采样结果一致，输出采样值，否则保持；

100 = 连续 4 次采样结果一致，输出采样值，否则保持；

101 = 连续 5 次采样结果一致，输出采样值，否则保持；

110 = 连续 6 次采样结果一致，输出采样值，否则保持；

111 = 连续 7 次采样结果一致，输出采样值，否则保持；

图注：R=可读 W=可写 -=未用 U=未实现位

注：每次采样的间隔时间通过采样时钟分频寄存器 C1FILTPRE 设置。

13.3.4 比较器 1 滤波采样时钟分频寄存器 C1FILTPRE

寄存器13.4: C1FILTPRE: 比较器1滤波采样时钟分频寄存器(地址: 14BH)

	bit7							bit0
复位值 0000 0000	FP17	FP16	FP15	FP14	FP13	FP12	FP11	FP10
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

图注：R=可读 W=可写 -=未用 U=未实现位

★ 式13.1: 比较器1滤波采样时钟频率 $= \dfrac{SCLK}{FP1<7:0>+1}$

注：（1）当 FP1<7:0>=0 时，滤波采样时钟即为系统时钟 SCLK。

13.4 CMP2 相关寄存器

13.4.1 比较器 2 控制寄存器 C2CTL

寄存器13.5: C2CTL: 比较器2控制寄存器(地址: 112H)

	bit7							bit0
复位值 0000 0000	C2M3	C2M2	C2M1	C2M0	C2CALEN	C2CALSET	C2MOD2	C2EN
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

C2M<3:0>：比较器电阻分压（VDAC）选择位

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

C2CALEN： 模拟比较器 2 校准使能位
1 = 使能比较器 2 校准功能
0 = 禁止比较器 2 校准功能

C2CALSET： 模拟比较器 2 校准设置位
1 = 比较器 2 正负端串接 1/2VDD

0 = 比较器 2 正负端未串接

C2MOD2: CMP2 的负输入端选择位

1 = 内部参考电压 VDAC 作为 CMP2 的负输入端

0 = IO 口作为 CMP2 的负输入端，具体配置位为 C2MOD[1:0]（COUT）

C2EN: 比较器 2 使能位

1 = 使能比较器 2

0 = 关闭比较器 2

图注：R=可读 W=可写 -=未用 U=未实现位

13.4.2 比较器 2 校准寄存器 C2CAL

寄存器13.6: C2CAL: 比较器2校准寄存器 （116H）

复位值 0000 0000	bit7							bit0
	C2DT7	C2DT6	C2DT5	C2DT4	C2DT3	C2DT2	C2DT1	C2DT0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

C2DT<7:0>: 模拟比较器校准数据位

C2DT = 比较器校验数据

图注：R=可读 W=可写 -=未用 U=未实现位

13.4.3 比较器 2 滤波控制寄存器 C2FILTCTL

寄存器13.7: C2FILTCTL: 比较器2滤波控制寄存器 （14CH）

复位值 0000 0000	bit7							bit0
	INV2	WEN2	FEN2	-	-	F2CNT2	F2CNT1	F2CNT0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

INV2: 比较器 2 输出极性选择位

INV2=0， 比较器正常输出：CIN+>CIN-=1，CIN+<CIN-=0

INV2=1， 比较器输出极性转换：CIN+>CIN-=0，CIN+<CIN-=1

WEN2: 比较器 2 范围控制使能位

WEN2=0，禁止范围控制功能

WEN2=1，使能范围控制功能

FEN2: 比较器 2 滤波使能位

FEN2=0, 禁止比较器 2 滤波功能

FEN2=1, 使能比较器 2 滤波功能

F2CNT<2:0>: 比较器 2 滤波采样数量选择位

F2CNT =000, 不采样，禁止滤波功能

F2CNT =001， 采样 1 次，并输出结果

F2CNT =010， 连续 2 次采样结果一致，输出采样值，否则保持；

F2CNT =011， 连续 3 次采样结果一致，输出采样值，否则保持；

F2CNT =100， 连续 4 次采样结果一致，输出采样值，否则保持；

F2CNT =101， 连续 5 次采样结果一致，输出采样值，否则保持；

F2CNT =110， 连续 6 次采样结果一致，输出采样值，否则保持；

F2CNT =111， 连续 7 次采样结果一致，输出采样值，否则保持；

图注：R=可读 W=可写 -=未用 U=未实现位

13.4.4 比较器 2 滤波采样时钟分频寄存器 C2FILTPRE

寄存器13.8: C2FILTPRE: 比较器2滤波采样时钟分频寄存器(地址: 14DH)

复位值	bit7							bit0
0000 0000	FP27	FP26	FP25	FP24	FP23	FP22	FP21	FP20
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

图注：R=可读 W=可写 -=未用 U=未实现位

★ 式13.2: $$\text{比较器2滤波采样时钟频率} = \frac{SCLK}{FP2<7:0>+1}$$

注：（1）当 FP2<7:0>=0 时，滤波采样时钟即为系统时钟 SCLK。

13.4.5 比较器输出寄存器 COUT

寄存器13.9: COUT: 比较器输出寄存器(地址: 1AH)

复位值	bit7							bit0
-000 0000	-	VCEN	C2OE	C2MOD1	C2MOD0	-	-	C2OUT
	U	R/W	R/W	R/W	R/W	R	R	R

VCEN: 比较器分压电路电压源选择位
- 0 = 电压源为 VDD
- 1 = 电压源为 VREF

C2OE: 比较器 2 输出使能位
- 0 = 禁止比较器 2 结果输出到 IO 口
- 1 = 使能比较器 2 结果输出到 IO 口

C2MOD<1:0>: CMP2 的负输入端选择位
- C2MOD2=0：
  - 00 = P0.1 作为 CMP2 的负输入端
  - 01 = P1.1 作为 CMP2 的负输入端
  - 10 = P1.2 作为 CMP2 的负输入端
  - 11 = P2.2 作为 CMP2 的负输入端
- C2MOD2=1：无效

C2OUT: 比较器 2 输出
- INV2=0 时
  - 1 = C2IN+ > C2IN-
  - 0 = C2IN+ < C2IN-
- INV2=1 时
  - 0 = C2IN+ > C2IN-
  - 1 = C2IN+ < C2IN-

图注：R=可读 W=可写 -=未用 U=未实现位

13.5 极性选择

模拟比较器的输出状态可以通过 CxFILTCTL（x=1,2）寄存器的 INVx（x=1,2）位结果选择，以比较器 1 为例，如表 11-2 所示。 INV1=0，比较器 1 正常输出：C1IN+>C1IN-=1，C1IN+<C1IN-=0；INV1=1， 比较器 1 输出极性转换：C1IN+>C1IN-=0，C1IN+<C1IN-=1。

表 13-2 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 = 0	C1OUT = 0
C1IN+ > C1IN-	INV1 = 1	C1OUT = 0
C1IN+ < C1IN-	INV1 = 1	C1OUT = 1

13.6 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 CxFILTCTL（x=1,2）寄存器的 WENx（x=1,2）位来使能/禁止范围控制功能。

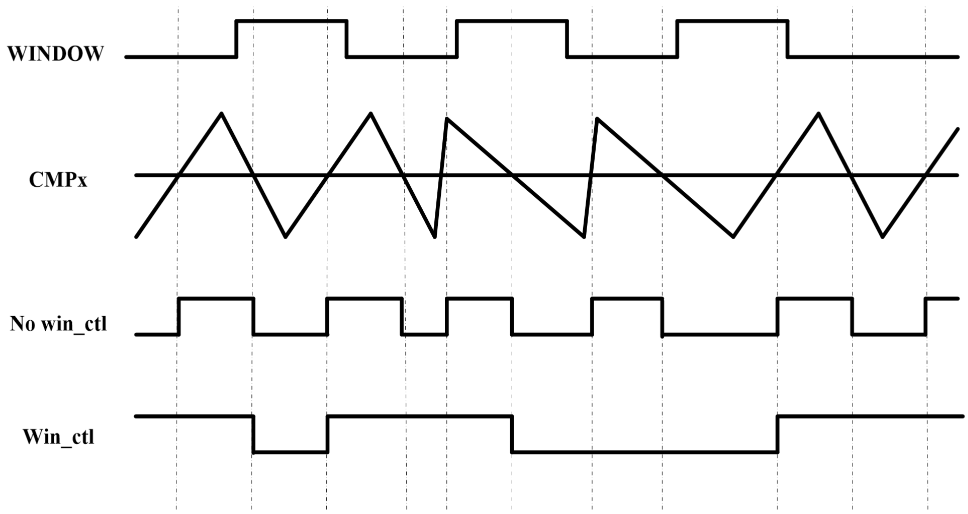


图 13.2 模拟比较器范围控制

13.7 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 CxFILTCTL（x=1,2）寄存器的 FENx（x=1,2）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 CxFILTCTL（x=1,2）寄存器的 FENx（x=1,2）位来使能滤波功能；
- 通过 FPx<7:0>（x=1,2）位选择滤波器采样时钟；
- 通过 FxCNT<2:0>（x=1,2）位设置滤波器的采样次数。

13.8 比较器中断

当比较器的输出值发生变化时，相应的比较器中断标志位就会置 1，此时将产生中断。用户如果使用中断需要打开外设中断使能位 PUIE、全局中断使能位 AIE 以及相应的比较器中断使能位 CxIE（x=1,2）；将 PCx（x=1,2）位置 1，还可将比较器中断设置成高优先级中断。

注：用户在进入比较器中断后，必须先读取相应比较器的输出位 CxOUT（x=1,2），再将中断标志位清 0，如例 13.1 所示。

例 13.1：进入比较器 1 中断后的操作

```
MOV R0 , C1CTL
CLR EIF1 , C1IF
```

13.9 模拟比较器的校准步骤

模拟比较器1校准步骤：

1. 配置校准使能位C1CALEN=1，C1CALSET=1，使能CMP1(C1EN=1);
2. 延迟20us，使C1DT[7:0]=01111111;
3. 延迟20us，判断模拟比较器CMP1的输出C1OUT
   - 比较器CMP1输出C1OUT=0（负失调），将C1DT[7]=0
   - 比较器CMP1输出C1OUT=1（正失调），将C1DT[7]=1
4. 将C1DT[6]清0，延迟20us，判断输出
   - 比较器CMP1输出C1OUT=0，负失调，C1DT[6]=0
   - 比较器CMP1输出C1OUT=1，正失调，C1DT[6]=1
5. 依次再对C1DT<5:0>各位做同样的操作。

注：模拟比较器 2 的校准方法和模拟比较器 1 相同。

例 13.2：模拟比较器 1 校准程序参考

```
        SET     C1CTL,C1CALEN      ;比较器1校准使能位C1CALEN=1,使能比较器1校准功能
        SET     C1CTL,C1CALSET     ;比较器1校准设置位C1CALSET=1, 比较器1正负端串接
1/2VDD
        CLR     C1CTL,C1OUT        ;比较器1输出C1OUT清0
        SET     C1CTL,C1EN         ;比较器 1 使能位 C1EN=1,使能比较器 1
        CALL    DELAY_20US         ;延时20us
        MOV     R0,#0X7F
        SET     BANK,PR0           ;切到SFR的1区
        MOV     C1CAL,R0
        CLR     BANK,PR0           ;切到SFR的0区
        MOV     R3,#B'10000000'
        MOV     R4,#B'10111111'
        MOV     R2,#0X08           ;循环8次
CMP1_Calibration
        CALL    DELAY_20US
        SET     BANK,PR0           ;切到SFR的1区
        MOV     R1,C1CAL
        CLR     BANK,PR0           ;切到SFR的0区
        JNB     C1CTL,C1OUT
        XOR     R1,R3
        AND     R1,R4
        SET     BANK,PR0           ;切到SFR的1区
        MOV     C1CAL,R1
        CLR     BANK,PR0           ;切到SFR的0区
        CALL    DELAY_20US
        CLR     PSW,CY
        RRC     R3                 ;右移一位，进行下一位的置1准备
        SET     PSW,CY
        RRC     R4
        DECJZ   R2
        JMP     CMP1_ Calibration
```

13.10 比较器使用

使用比较器时需要进行下列设置：

1. 如果用户需要使用比较器的失调自校准功能，则需要按照章节 13.9 的校准步骤对比较器进行相应的操作。
2. 配置相应的输入和输出引脚为模拟/数字、输入/输出状态；
3. 设置相应寄存器的控制位来为模拟比较器选择工作模式；
4. 如果需要滤波功能，通过 FENx 位使能滤波器并通过 FPx<7:0>、FxCNT<2:0>位设置滤波器工作模式；
5. 如果需要使用中断需要配置中断相关的寄存器位。

13.11 复位的影响

当器件复位时，强制将寄存器 CxCTL 置为复位状态，这使比较器和电阻分压参考电压被“强制”设置为关闭状态。

14 硬件乘法器模块

14.1 概述

KF8F3110/12/20/22/30/32 包含一个单独的 16×16 的硬件乘法器模块，它是单片机的一个外设。CPU 通过相关的指令操作来对硬件乘法器的寄存器进行读写与运算操作。

乘法器的特性包含：

- 无符号 16 位二进制乘法；
- 乘法通过异步运算过程实现（8 个 T_{mc} 运算出结果）

工作原理框图如图 14.1 所示：

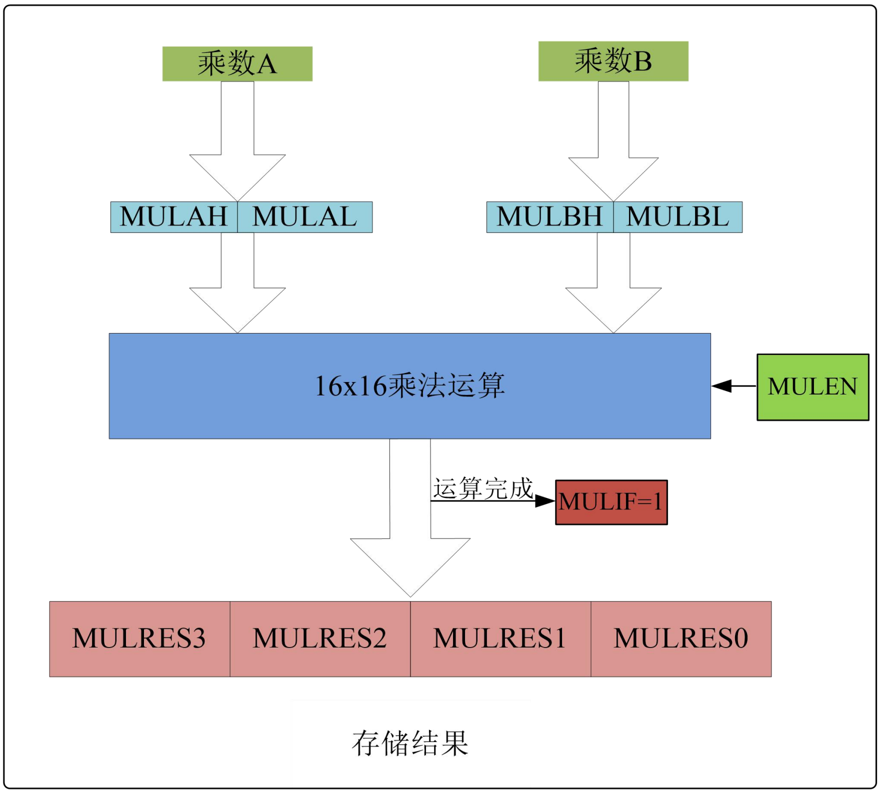


图 14.1　16×16 硬件乘法器原理框图

> 注：结果存入32位结果寄存器MULRES中，即运算结果为：[MULRES3:MULRES2:MULRES1:MULRES0]，MULRES3是最高8位，MULRES0是最低8位。

14.2 硬件乘法器相关寄存器

表 14-1 硬件乘法器相关寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
130H	MULAH	16 位乘法器乘数 A 高 8 位							
131H	MULAL	16 位乘法器乘数 A 低 8 位							
132H	MULBH	16 位乘法器乘数 B 高 8 位							
133H	MULBL	16 位乘法器乘数 B 低 8 位							
134H	MULCTL	-	-	-	-	-	-	MULEN	MULIF
135H	MULRES3	乘法运算结果寄存器 3							
136H	MULRES2	乘法运算结果寄存器 2							
137H	MULRES1	乘法运算结果寄存器 1							
138H	MULRES0	乘法运算结果寄存器 0							

14.2.1 乘法运算控制寄存器（MULCTL）

寄存器**14.1:** MULCTL:乘法运算控制寄存器（地址:134H）

复位值 ---- --00	bit7							bit0
	-	-	-	-	-	-	MULEN	MULIF
	U	U	U	U	U	U	R/W	R

MULEN: 运算使能位

1 = 使能乘法运算

0 = 禁止乘法运算

MULIF: 运算状态标志位

1 = 运算操作已经完成

0 = 没有完成运算

图注：R=可读 W=可写 -=未用 U=未实现位

注：MULIF 不能直接由软件清零。对 MULAH/L 或 MULBH/L 进行赋值操作会使 MULIF 硬件清零，MULEN 置 1 使能乘法运算，当运算操作已经完成，MULIF 被硬件置 1。

14.3 硬件乘法器的使用

运算操作的使用设置如下:

1) 向乘数 A 寄存器和乘数 B 寄存器分别写入一个无符号的 16 位乘数
2) 乘法运算使能位 MULEN 置 1，控制开始乘法运算
3) 等待 MULIF 标志为 1
4) 乘法运算使能位 MULEN 清零，关闭乘法运算
5) [MULRES3:MULRES2:MULRES1:MULRES0]存放计算结果，MULRES3 为最高 8 位，MULRES0 为最低 8 位

15 硬件除法器模块

15.1 概述

KF8F3110/12/20/22/30/32 包含一个硬件除法器，它是单片机的一个外设。CPU 通过相关的指令操作来对硬件除法器的寄存器进行读写与运算操作。

硬件除法器的特性包含:

◆ 无符号除法

◆ 16÷8 位（9 个 T_{mc} 运算）

具体的原理框图如图 15.1 所示:

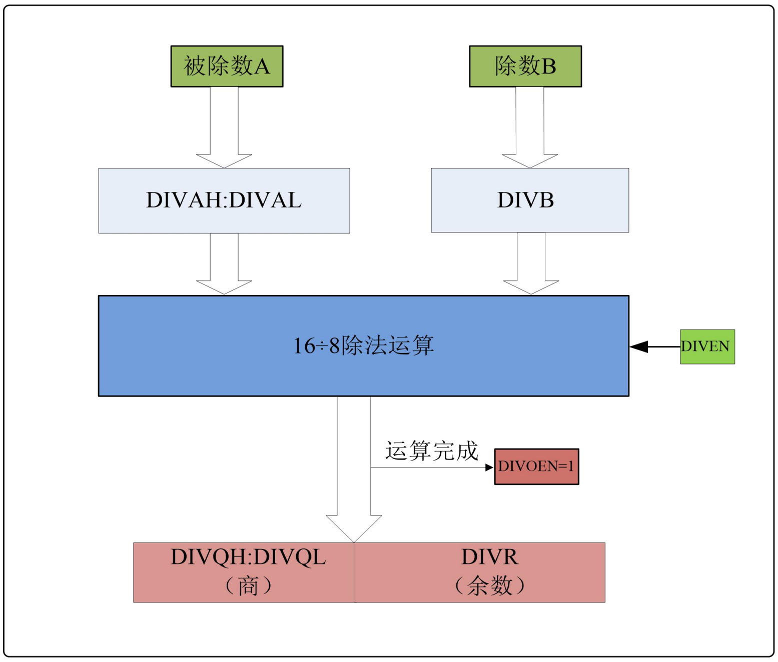


图 15.1 硬件除法器原理框图

15.2 硬件除法器相关寄存器

表 15-1 硬件除法器相关寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
139H	DIVCTL	-	-	-	-	-	-	DIVOEN	DIVEN
13AH	DIVAH	被除数高 8 位寄存器							
13BH	DIVAL	被除数低 8 位寄存器							
13CH	DIVB	除数寄存器							
13DH	DIVQH	商高 8 位寄存器							
13EH	DIVQL	商低 8 位寄存器							
13FH	DIVR	余数寄存器							

15.2.1 除法控制寄存器（DIVCTL）

在除法的过程中，需要对除法模块的使能进行控制，同时需要有一个可以通知 CPU 或者其他模块，除法已经完成的信号。

寄存器**15.1:**DIVCTL: 除法控制寄存器(地址:139H)

复位值	bit7							bit0
------00	-	-	-	-	-	-	DIVOEN	DIVEN
	U	U	U	U	U	U	R	W

DIVOEN: 除法运算完成标志位
1 = 除法完成，商和余数可读
0 = 除法未完成或者未开始，商和余数不可读

DIVEN: 除法运算使能位
1 = 使能除法运算
0 = 禁止除法运算

图注：R=可读 W=可写 -=未用 U=未实现位

注：DIVOEN 不能直接由软件清零。对 DIVAH/DIVAL 或 DIVB 进行赋值操作会使 DIVOEN 硬件清零，DIVEN 软件置 1 使能除法运算，当运算操作已经完成， DIVOEN 被硬件置 1，DIVEN 会被硬件清零。

15.2.2 数据寄存器

对于除法，需要有操作数进行操作，在开始除法的时候，需要被除数和除数，当除法结束后，需要商和余数。

这四种分别使用{DIVAH:DIVAL}，DIVB，{DIVQH:DIVQL}，DIVR 进行表示。

15.3 硬件除法器操作

除法运算器的设置:

1） 分别向被除数寄存器{DIVAH:DIVAL}写入被除数
2） 向除数寄存器 DIVB 写入除数（除数不能为 0）
3） 除法运算使能位 DIVEN 置 1 使能除法运算器
4） 等待除法运算完成标志位 DIVOEN 置 1

{DIVQH:DIVQL}存放商的结果，DIVR 存放余数的结果

16 复位

KF8F3110/12/20/22/30/32 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。

有些寄存器的状态在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 16.1 给出了片内复位电路的简化结构方框图。

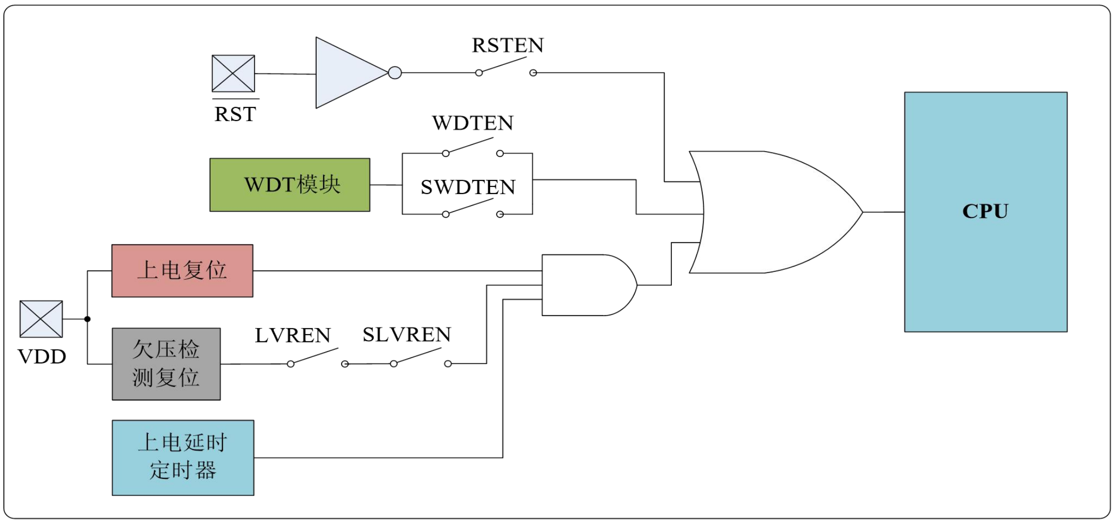


图 16.1 片内复位电路简化框图

注：上电延时定时器只对上电复位(POR)和欠压检测复位(LVR)有效。

16.1 电源控制寄存器(PCTL)

如寄存器 16.1 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16.1: PCTL：电源控制寄存器(地址:2EH)

复位值 ---1 000x	bit7							bit0
	-	-	-	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

当配置字的 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

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

图注：R=可读 W=可写 -=未用 U=未实现位

16.2 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单片机才开始正常工作。KF8F3110/12/20/22/30/32 的上电复位时间为 32ms 左右。

16.3 WDT 复位

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。

16.4 RST 复位

使能外部 RST 复位（配置位 RSTEN=1）后，当引脚 P0.3/$\overline{\text{RST}}$ 输入复位信号，不管单片机工作在正常模式还是休眠模式，均会使单片机复位。通过在编程时将 P0.3 引脚配置为 $\overline{\text{RST}}$ 复位引脚，即可打开 $\overline{\text{RST}}$ 复位。

在 $\overline{\text{RST}}$ 复位时， KF8F3110/12/20/22/30/32 器件有一个噪声滤波器用于检测和滤除 $\overline{\text{RST}}$ 引脚上的噪声干扰，图 16.2 是建议 $\overline{\text{RST}}$ 复位电路。

注：KF8F3110SB 的 RST 引脚位于 P2.2。

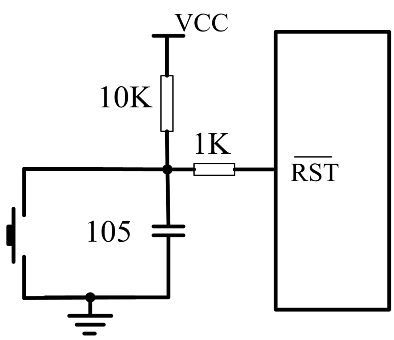


图 16.2 建议 $\overline{\text{RST}}$ 复位电路

16.5 欠压检测复位(LVR)

KF8F3110/12/20/22/30/32 系列中的单片机具有片内欠压检测复位电路。通过编程时设定配置位中的 LVREN 位可以禁止/使能(清 0/置 1)欠压检测复位电路，当配置位中的 LVREN 位被使能后，用户还要在软件中设置 PCTL 中的 SLVREN 位来禁止/使能(清 0/置 1)欠压检测复位电路。

如果 VDD 跌落至 V_{LVR} 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 VDD 上升到 V_{LVR} 以上，如果上电延时使能（$\overline{PWRT}$=0），此时上电延时定时器启动，使器件在随后 32ms 左右的延时时间处于延时复位状态，过了 32ms 以后单片机开始正常工作。

如果 VDD 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 VDD 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 VDD 上升至 V_{LVR} 以上时，上电延时定时器启动一个 32ms 的复位延时，如在延时期间没有欠压发生，单片机会退出复位状态开始正常工作。

16.6 上电延时定时器

上电延时定时器仅在器件上电复位或欠压检测复位发生后提供一个长度为 32ms 的固定延时时间（配置位 $\overline{PWRT}$=0），用户可以通过设置配置位 PWRT 使能上电延时功能。上电延时定时器的定时时钟为系统内部低频振荡器。当使能上电延时时，只要单片机产生上电复位或欠压检测复位，单片机就会在上电复位或欠压检测复位发生后保持复位状态 32ms。上电延时定时器使单片机在 VDD 上升到适当电平后才投入正常运行。

由于 VDD、温度、制造工艺、内部震荡器频率等的变化，不同单片机的上电延时时间有所差异。

16.7 不同复位条件下对寄存器的影响

表 16-1 寄存器在各种复位发生后的状态

地址	名称	上电复位	RST 复位 欠压检测	中断唤醒 WDT 超时唤醒
01H	T0	xxxx xxxx	uuuu uuuu	uuuu uuuu
02H	PCL	0000 0000	0000 0000	PC+1
03H	PSW	---1 1xxx	---q quuu	---q quuu
05H	P0	--xx xxxx	--uu uuuu	--uu uuuu
06H	P2	---- xxxx	---- uuuu	---- uuuu
07H	P1	xxxx xxxx	uuuu uuuu	uuuu uuuu
0AH	PCH	---0 0000	---0 0000	---u uuuu
0BH	INTCTL	0000 0000	0000 0000	uuuu uuuu
0CH	EIF1	0000 0000	0000 0000	uuuu uuuu
0DH	EIF2	0000 0000	0000 0000	uuuu uuuu
0EH	T1L	0000 0000	0000 0000	uuuu uuuu
0FH	T1H	0000 0000	0000 0000	uuuu uuuu
10H	T1CTL	0000 0000	uuuu uuuu	uuuu uuuu
11H	T2L	0000 0000	0000 0000	uuuu uuuu
12H	T2CTL0	-000 0000	-000 0000	-uuu uuuu
13H	PWM1L	xxxx xxxx	uuuu uuuu	uuuu uuuu
14H	PWM1H	xxxx xxxx	uuuu uuuu	uuuu uuuu
15H	PWMCTL	0000 0000	0000 0000	uuuu uuuu
16H	PP1	1111 1111	1111 1111	uuuu uuuu
17H	BANK	---- 0000	---- 0000	---- uuuu
18H	ADSCANCTL	0000 0000	0000 0000	uuuu uuuu
19H	C1CTL	0000 0000	0000 0000	uuuu uuuu
1AH	COUT	-000 0000	-000 0000	-uuu uuuu
1BH	AMPCTL	1-00 0000	1-00 0000	u-uu uuuu
1DH	ANSEH	--00 0000	--00 0000	--uu uuuu
1EH	ADCDATA0H	xxxx xxxx	uuuu uuuu	uuuu uuuu
1FH	ADCCTL0	0000 0000	0000 0000	uuuu uuuu
20H	AMPDT	1000 0000	1000 0000	uuuu uuuu
21H	OPTR	1111 1111	1111 1111	uuuu uuuu
22H	IP0	---- -000	---- -000	--- -uuu
23H	IP1	0000 0000	0000 0000	uuuu uuuu
24H	IP2	0000 0000	0000 0000	uuuu uuuu
25H	TR0	1111 1111	1111 1111	uuuu uuuu
26H	TR2	1111 1111	1111 1111	uuuu uuuu
27H	TR1	1111 1111	1111 1111	uuuu uuuu
28H	OSCSTA	0110 --00	0110 --00	uuuu –uu
29H	IP3	000- ----	000- ----	uuu- ----
2AH	VRECAL1	0111 0111	0111 0111	uuuu uuuu
2BH	VRECTL	0000 0000	0000 0000	uuuu uuuu
2CH	EIE1	0000 0000	0000 0000	uuuu uuuu
2DH	EIE2	0000 0000	0000 0000	uuuu uuuu
2EH	PCTL	---1 000x	---1 00uq	---u uuuu
2FH	OSCCTL	0010 0000	0010 0000	uuuu uuuu
30H	OSCCAL0	1000 0000	1000 0000	uuuu uuuu
31H	ANSEL	0000 0000	0000 0000	uuuu uuuu
32H	PP2	1111 1111	1111 1111	uuuu uuuu
33H	PWM2L	xxxx xxxx	uuuu uuuu	uuuu uuuu
34H	PWM2H	xxxx xxxx	uuuu uuuu	uuuu uuuu
35H	PUR0	1111 -111	1111 -111	uuuu -uuu
36H	IOCL	0000 0000	0000 0000	uuuu uuuu
37H	OSCCAL1	0000 -001	0000 -001	uuuu -uuu
38H	NVMDATAH	0000 0000	0000 0000	uuuu uuuu

地址	名称	上电复位	RST 复位 欠压检测	中断唤醒 WDT 超时唤醒
39H	NVMDATAL	0000 0000	0000 0000	uuuu uuuu
3AH	NVMADDRH	0000 0000	0000 0000	uuuu uuuu
3BH	NVMADDRL	0000 0000	0000 0000	uuuu uuuu
3CH	NVMCTL0	---- x000	---- q000	---- uuuu
3DH	NVMCTL1	---- ----	---- ----	---- ----
3EH	ADCDATA0L	xxxx xxxx	uuuu uuuu	uuuu uuuu
3FH	ADCCTL1	0000 0000	0000 0000	uuuu uuuu
40H	T2CCR0H	0000 0000	0000 0000	uuuu uuuu
41H	T2H	0000 0000	0000 0000	uuuu uuuu
42H	PP5H	0000 0000	0000 0000	uuuu uuuu
43H	PWM5H0	0000 0000	0000 0000	uuuu uuuu
44H	PWM5H1	0000 0000	0000 0000	uuuu uuuu
45H	P0LR	xxxx xxxx	uuuu uuuu	uuuu uuuu
46H	P2LR	xxxx xxxx	uuuu uuuu	uuuu uuuu
47H	P1LR	xxxx xxxx	uuuu uuuu	uuuu uuuu
4AH	EIE3	000- ----	000- ----	uuu- ----
4BH	EIF3	000- ----	000- ----	uuu- ----
4CH	OSCCAL2	0011 1111	0011 1111	uuuu uuuu
4DH	OSCCAL3	0011 1111	0011 1111	uuuu uuuu
4EH	T3CTL	0000 0000	0000 0000	uuuu uuuu
4FH	T3L	0000 0000	0000 0000	uuuu uuuu
50H	ADCDATA1H	xxxx xxxx	uuuu uuuu	uuuu uuuu
51H	ADCDATA1L	xxxx xxxx	uuuu uuuu	uuuu uuuu
52H	PP5L	0000 0000	0000 0000	uuuu uuuu
53H	T3CTL1	0000 0000	0000 0000	uuuu uuuu
54H	T2CCR0L	0000 0000	0000 0000	uuuu uuuu
55H	PWM5L0	0000 0000	0000 0000	uuuu uuuu
56H	PWM5L1	0000 0000	0000 0000	uuuu uuuu
57H	PWM5CTL0	00-- 0000	00-- 0000	uu-- uuuu
58H	ADCSICM	0000 0000	0000 0000	uuuu uuuu
59H	ADCDATA2H	xxxx xxxx	uuuu uuuu	uuuu uuuu
5AH	ADCDATA2L	xxxx xxxx	uuuu uuuu	uuuu uuuu
5BH	PWM5CTL1	0000 0000	0000 0000	uuuu uuuu
5CH	P5ASCTL	0000 0000	0000 0000	uuuu uuuu
5DH	PSTRCTL	---0 0001	---0 0001	---u uuuu
5EH	ADCDATA3H	xxxx xxxx	uuuu uuuu	uuuu uuuu
5FH	T3H	0000 0000	0000 0000	uuuu uuuu
60H	PUR1	1111 1111	1111 1111	uuuu uuuu
61H	PUR2	1111 1111	1111 1111	uuuu uuuu
67H	INTEDGCTL	11-- ---0	11-- ---0	uu-- ---u
6AH	ADCDATA3L	xxxx xxxx	uuuu uuuu	uuuu uuuu
10EH	PWM5OC	--00 0000	--00 0000	--uu uuuu
10FH	T2CTL1	0000 0--x	0000 0--x	uuuu u--u
112H	C2CTL	0000 0000	0000 0000	uuuu uuuu
115H	C1CAL	0000 0000	0000 0000	uuuu uuuu
116H	C2CAL	0000 0000	0000 0000	uuuu uuuu
11CH	PWM5CTL2	--1- -000	--1- -000	--u- -uuu
11DH	PWM5PC	--00 0000	--00 0000	--uu uuuu
120H	RSCTL	0000 0000	0000 0000	uuuu uuuu
121H	TXSDR	0000 0000	0000 0000	uuuu uuuu
122H	RXSDR	0000 0000	0000 0000	uuuu uuuu
123H	BRCTL	0100 0000	0100 0000	uuuu uuuu
124H	TSCTL	0000 0010	0000 0010	uuuu uuuu
125H	EUBRGL	0000 0000	0000 0000	uuuu uuuu
126H	EUBRGH	0000 0000	0000 0000	uuuu uuuu
128H	SSCICTL0	0000 0000	0000 0000	uuuu uuuu

地址	名称	上电复位	RST 复位 欠压检测	中断唤醒 WDT 超时唤醒
12AH	SSCICTL1	0000 0000	0000 0000	uuuu uuuu
12BH	SSCISTA	0000 0000	0000 0000	uuuu uuuu
12CH	SSCIBUFR	xxxx xxxx	uuuu uuuu	uuuu uuuu
12EH	SSCIADD	0000 0000	0000 0000	uuuu uuuu
	SSCIMSK	1111 1111	1111 1111	uuuu uuuu
12FH	WDTPS	---- 0100	---- 0100	---- uuuu
130H	MULAH	0000 0000	0000 0000	uuuu uuuu
131H	MULAL	0000 0000	0000 0000	uuuu uuuu
132H	MULBH	0000 0000	0000 0000	uuuu uuuu
133H	MULBL	0000 0000	0000 0000	uuuu uuuu
134H	MULCTL	---- --00	---- --00	---- --uu
135H	MULRES3	0000 0000	0000 0000	uuuu uuuu
136H	MULRES2	0000 0000	0000 0000	uuuu uuuu
137H	MULRES1	0000 0000	0000 0000	uuuu uuuu
138H	MULRES0	0000 0000	0000 0000	uuuu uuuu
139H	DIVCTL	---- --00	---- --00	---- --uu
13AH	DIVAH	0000 0000	0000 0000	uuuu uuuu
13BH	DIVAL	0000 0000	0000 0000	uuuu uuuu
13CH	DIVB	0000 0000	0000 0000	uuuu uuuu
13DH	DIVQH	0000 0000	0000 0000	uuuu uuuu
13EH	DIVQL	0000 0000	0000 0000	uuuu uuuu
13FH	DIVR	0000 0000	0000 0000	uuuu uuuu
14AH	C1FILTCTL	0000 0000	0000 0000	uuuu uuuu
14BH	C1FILTPRE	0000 0000	0000 0000	uuuu uuuu
14CH	C2FILTCTL	0000 0000	0000 0000	uuuu uuuu
14DH	C2FILTPRE	0000 0000	0000 0000	uuuu uuuu
151H	CCRL	0000 0000	0000 0000	uuuu uuuu
152H	CCPCTL	---0 0000	---0 0000	---u uuuu
154H	CCRH	0000 0000	0000 0000	uuuu uuuu
156H	PINSET	0000 0000	0000 0000	uuuu uuuu
159H	VRECAL2	0000 0000	0000 0000	uuuu uuuu
15AH	VRECAL3	1010 0000	1010 0000	uuuu uuuu
15DH	PWM5FC	--00 0000	--00 0000	--uu uuuu
165H	RC32KCAL	0000 0000	0000 0000	uuuu uuuu

注：u=不变； x=未知值； -=未用，读为 0； q=该位取决于当前条件。

表 16-2　不同复位条件下对标志位的影响

$\overline{\text{POR}}$	$\overline{\text{LVR}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	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

17 休眠模式

当单片机空闲的时候，为使其功耗降到最低，可以将其转入休眠模式。通过执行一条 IDLE 指令即可进入休眠模式。

为使这种方式下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，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。

注：在单片机正常工作时，通常有些引脚用不到，有的用户可能会直接将其悬空。为了减小单片机的功耗，应该将不用的引脚设置为数字输出。如果是P0口的引脚则可打开上拉电阻或者设置为数字输出皆可。

单片机进入休眠模式一段时间后由于工作的需要，要将单片机从休眠模式唤醒，在 KF8F3110/12/20/22/30/32 中可通过以下方式将单片机从休眠模式唤醒:

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/INT1/INT2 外部中断
4. T1 中断(异步计数模式)
5. P0 口电平变化中断

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模式唤醒，并不会导致复位。可通过状态寄存器中的 $\overline{TO}$ 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，而当器件从休眠模式唤醒时，该位将被清 0。$\overline{TO}$ 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 指令之后加一条 NOP 指令即可。

18 看门狗定时器 WDT

为了防止单片机在正常工作时程序跑飞，KF8F3110/12/20/22/30/32 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。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。

18.1 看门狗相关寄存器

表 18-1 看门狗相关的寄存器

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
2EH	PCTL	-	-	-	SLVREN	IPEN	SWDTEN	$\overline{POR}$	$\overline{LVR}$
12FH	WDTPS	-	-	-	-	WDTPS3	WDTPS2	WDTPS1	WDTPS0

18.1.1 WDT 预分频选择寄存器 WDTPS

寄存器18.1: WDTPS：WDT预分频选择寄存器(地址:12FH)

	bit7							bit0
复位值 ---- 0100	-	-	-	-	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） 16ms
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 看门狗的开启关闭方式

看门狗的开启/关闭方式：

- 通过配置位 WDTEN，打开/关闭看门狗；
- 通过寄存器 PCTL 的 SWDTEN 位，打开/关闭看门狗。

上述两种方式任意一种都可启动看门狗；配置位 WDTEN 一旦使能，看门狗将一直开启，软件配置位 SWDTEN 无效；配置位 WDTEN 未使能时，软件配置位 SWDTEN 允许用户在软件上根据实际需求打开/关闭看门狗。

18.3 看门狗的清狗方式

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进行清狗操作。执行 CWDT 指令进行清狗操作或者执行 IDLE 指令进入休眠模式后，将清零整个看门狗定时器（包括看门狗预分频器和后分频器）。当看门狗定时器出现超时时，状态字寄存器 PSW 中的$\overline{\text{TO}}$位将被清 0。

看门狗定时器使用内部低频振荡器作为工作时钟源，因此它无需外接任何器件，在休眠模式仍能正常运行。WDT 超时事件对单片机的动作：

- 在正常运行时，WDT 超时事件将使单片机产生一次复位；
- 在休眠模式下，WDT 超时事件将唤醒单片机并使其继续执行 IDLE 后面的指令。

18.4 看门狗 WDT 的周期

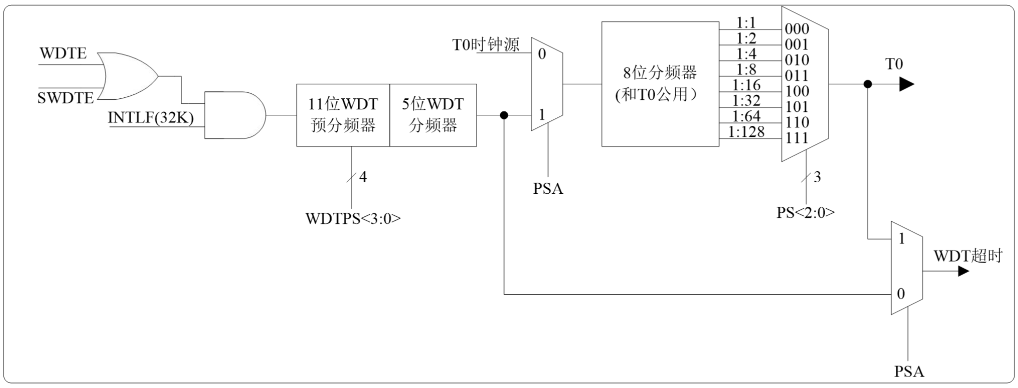


图 18.1 看门狗定时器框图

如看门狗定时器框图所示：看门狗定时器的时钟源为内部低频时钟（带校正功能的 32K），相关内容详见内部低频振荡器章节。

看门狗定时器的周期由两个分频器的配置决定，最短约 1ms，最长约 268s，默认配置时为 16ms。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。

看门狗定时器带有两个分频器：

- 一个 16 位（11 位可编程）预分频器；
- 一个 8 位可编程后分频器（与 T0 共用）。

16 位预分频器中有 11 位可编程，由 WDTCTL 寄存器的 WDTPS<3:0>位选择预分频比（1:32 到 1:65536），共 12 档。

后分频器为看门狗定时器和定时/计数器 T0 共用，通过 OPTR 寄存器的 PSA 位将后分频器分配给 WDT 或者 T0，OPTR 寄存器的 PS<2:0>位选择后分频器的分频比（1/1 到 1/128）。

注：任何对OPTR寄存器的PSA位或者PS<2:0>位操作前和操作后，都须对看门狗定时器进行清狗操作（执行CWDT指令）。否则，可能引起芯片的异常复位。

19 电气规范

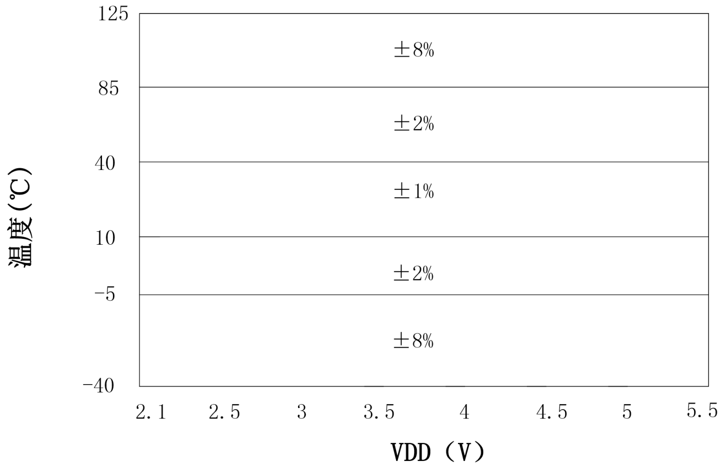
19.1 极限参数特性

表 19-1 芯片极限参数特性

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℃ ~125℃
2	储存温度	-65℃ ~150℃
3	VDD 相对于VSS 的电压	-0.3V~+6.0V
4	其它引脚相对于VSS 的电压	-0.3V~VDD+0.3V
5	VSS 引脚的最大输出电流	80mA
6	VDD 引脚的最大输入电流	80mA
7	任一I/O 引脚的最大输出灌电流	15mA
8	任一I/O 引脚的最大输出拉电流	15mA
19	I/O口 的最大灌电流	80mA
10	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。

19.2 HFINT 的频率精度与器件 VDD 和温度之间的关系



19.3 静态电流特性

KF8F3110/12/20/22/30/32 的静态电流如表 19-2a、表 19-2b、表 19-2c 和表 19-2d 的所示：

表19-2a KF8F3110静态电流(IDD)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	V_{DD}(V)				
1	4MHz	5.0	-	320	455	μA
		3.0	-	320	455	
2	2MHz	5.0	-	240	380	
		3.0	-	240	380	
3	1MHz	5.0	-	180	280	
		3.0	-	180	280	
4	500kHz	5.0	-	150	230	
		3.0	-	150	230	
5	250kHz	5.0	-	135	210	
		3.0	-	135	210	
6	125kHz	5.0	-	130	200	
		3.0	-	130	200	
7	62.5kHz	5.0	-	125	190	
		3.0	-	125	190	
8	15.625kHz	5.0	-	120	185	
		3.0	-	120	185	

表19-2b KF8F3112静态电流（IDD）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	V_{DD}（V）				
1	4MHz	5.0	-	280	390	μA
		3.0	-	270	390	
2	2MHz	5.0	-	215	300	
		3.0	-	210	300	
3	1MHz	5.0	-	170	245	
		3.0	-	160	245	
4	500kHz	5.0	-	145	215	
		3.0	-	140	215	

5	250kHz	5.0	-	135	200	
		3.0	-	130	200	
6	125kHz	5.0	-	130	190	
		3.0	-	120	190	
7	62.5kHz	5.0	-	125	185	
		3.0	-	120	185	
8	15.625kHz	5.0	-	120	180	
		3.0	-	120	180	

表19-2c KF8F3120/22静态电流(IDD)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	V_{DD}（V）				
1	8MHz	5.0	-	585	820	μA
		3.0	-	575	820	
2	4MHz	5.0	-	465	650	
		3.0	-	455	650	
3	2MHz	5.0	-	400	560	
		3.0	-	390	560	
4	1MHz	5.0	-	370	520	
		3.0	-	360	520	
5	500kHz	5.0	-	355	500	
		3.0	-	350	500	
6	250kHz	5.0	-	350	490	
		3.0	-	340	490	
7	125kHz	5.0	-	345	485	
		3.0	-	335	485	
8	31.25kHz	5.0	-	340	480	
		3.0	-	335	480	

表19-2d KF8F3130/32静态电流(IDD)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	-	925	1290	μA
		3.0	-	910	1290	
2	8MHz	5.0	-	690	960	
		3.0	-	680	960	
3	4MHz	5.0	-	570	790	
		3.0	-	560	790	
4	2MHz	5.0	-	505	705	
		3.0	-	500	705	
5	1MHz	5.0	-	475	660	
		3.0	-	465	660	
6	500kHz	5.0	-	460	640	
		3.0	-	450	640	
7	250kHz	5.0	-	455	630	
		3.0	-	445	630	
8	62.5kHz	5.0	-	450	625	
		3.0	-	440	625	

注 1：在正常的工作模式下， IDD 测量的条件为：所有I/O引脚均设置为输出低，RST = VSS ，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。

19.4 外设电流特性

表19-3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	休眠电流（I_{IDLE}）	WDT、LVR 等外设被禁止	5.0	-	1.9	5	μA
			3.0	-	1.1	5	
2	WDT 电流（I_{WDT}）		5.0	-	2.7	4.0	
			3.0	-	1.5	2.5	
3	欠电压复位电流（I_{LVR}）		5.0	-	9.7	14	
			3.0	-	5.0	8	
4	比较器电流(I_{CMP})	使能单个比较器，无负载	3.0	-	35	75	
5	运算放大器电流(I_{OP})	使能运放，无负载	3.0	-	28	60	
6	FVR 电流(I_{FVR})	使能 FVR，无负载	5.0	-	105	115	
			3.0	-	75	80	
7	ADC 电流(I_{ADC})	ADCIM=0，使能 ADC，未转换	5.0	-	790	1200	
			3.0	-	605	925	
		ADCIM=1，使能 ADC，未转换	5.0	-	400	645	
			3.0	-	335	555	

注 1：外设电流是基本I_{DD} 或I_{IDLE} 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_{IDLE} 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 = VSS ；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。

19.5 I/O 端口特性

表 19-4 芯片 IO 端口特性

工作温度 -40℃≤TA≤+125℃						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
V_{IL}	输入低电平 I/O引脚： 采用TTL缓冲器 采用施密特缓冲触发器		VSS VSS VSS	-	0.6 0.15VDD 0.2 VDD	V
V_{IH}	输入高电平 I/O端口： 采用TTL缓冲器 采用施密特缓冲触发器		VDD -0.6 0.8 VDD	-	VDD VDD	V
V_{OL}	输出低电压		-	-	0.6	V
V_{OH}	输出高电压		VDD -0.6	-	-	V
I_{IL}	输入漏电流	VSS<VI<VDD	-1	-	1	µA

19.6 芯片供电电压特性

表 19-5 芯片供电电压特性

工作温度 -40℃≤TA≤+125℃						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
VDD	电源电压	FOSC≤16MHz	2.1	-	5.5	V
V_{LVR}	欠压检测复位电压		1.8		2.0	V
V_{LVR_HYST}	LVR迟滞电压		-	20		mV

19.7 A/D 转换器（ADC）特性

表 19-6 A/D 转换器（ADC）特性

工作温度 -40℃≤T_A≤+125℃						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
N_R*	分辨率		-	-	12	位
E_{INL}*	积分误差		-	±2	-	LSB
E_{DNL}*	微分误差		-	±1	-	LSB
E_{OFF}*	失调误差		-	±2	-	LSB
E_{GN}*	增益误差		-	±2	-	LSB
V_{AIN}*	满量程范围		VSS	-	V_{REF}	V
T_{CNV}*	AD转换时间		-	11	-	TAD
TsOPAle*	AD转换速率		-	-	400	KSPS

*该数据为设计值

19.8 内部高频振荡器模块特性

KF8F3110/12/20/22/30/32 的内部高频振荡器模块的特性如表 19-7a、19-7b、和 19-7c 所示：

表 19-7a KF8F3110/12 内部高频振荡器模块特性

测试条件（特别声明除外）： 工作温度 -40℃≤TA≤＋125℃							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
1	Fosc	振荡频率	3.96	4	4.04	Mhz	2.1V≤VDD≤5.5V @25℃

表 19-7b KF8F3120/22 内部高频振荡器模块特性

测试条件（特别声明除外）： 工作温度 -40℃≤TA≤＋125℃							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
1	Fosc	振荡频率	7.92	8	8.08	MHz	2.1V≤VDD≤5.5V @25℃

表 19-7c KF8F3130/32 内部高频振荡器模块特性

测试条件（特别声明除外）： 工作温度 -40℃≤TA≤＋125℃							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
1	Fosc	振荡频率	15.84	16	16.16	MHz	2.1V≤VDD≤5.5V @25℃

19.9 内部低频 LPRC 模块特性

表 19-7 内部低频 LPRC 模块特性

工作温度 -40℃≤TA≤+125℃						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
Freq*	32kHz		-5		5	%
Iq*	静态电流			330		nA

*该数据为设计值

19.10 外部低频 LPXTAL 模块特性

表 19-8 外部低频 LPXTAL 模块特性

工作温度 -40℃≤TA≤+125℃						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
Freq*	32.768kHz			-		kHz
Iq*	静态电流			340		nA

*该数据为设计值

19.11 模拟比较器模块特性

表 19-9 模拟比较器模块特性

工作温度 -40℃≤TA≤+125℃						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
Av*	开环增益			72		dB
Vicmr*	输入电压范围		VSS		VDD-1	V
Vos*	输入失调电压	校准后			±5	mV

*该数据为设计值

19.12 运算放大器模块特性

表 19-10 运算放大器模块特性

测试条件（特别声明除外）： 工作温度 -40℃ ≤TA≤+125℃							
类型	符号	特性	最小值	典型值	最大值	单位	备注
直流	V_{OS}*	输入失调电压	-	±3	-	mV	校准后
	ICMR*	输入共模电压范围	GND	-	VDD		
	V_{OT}*	输出电压摆幅	GND	-	VDD		
	Iq*	静态电流	-	47		uA	
	Isource*	源电流	-	5		mA	
	Isink*	灌电流	-	8.3		mA	
交流	GBW*	单位增益带宽	-	1		Mhz	Cload=100pF
	A_{OL}*	差模开环直流电压增益	-	115		dB	
	CMRR*	共模抑制比	-	120		dB	
	PSRR*	电源抑制比	-	109	-	dB	
瞬态	SR*	摆率	-1.53		+1.24	V/us	Cload=100pF
	Setting time*		-	4	-	us	Cload=100pF

注：1、*该数据为设计值；

2、KF8F3110/20/30无运算放大器模块。

19.13 内部参考电压模块特性

表 19-11 内部参考电压模块特性

工作温度：-40℃≤T_A≤+125℃							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
1	Vout	输出电压	-	2 3 4	-	V	Vout ≤ VDD ≤ 5.5 V
2	Accuracy	精度	-5		+ 5	%	Vout ≤ VDD ≤ 5.5 V

19.14 FLASH 自写和 DATA EEPROM 特性

表 19-12 内部参考电压模块特性

序号	测试参数	测试条件	最小值	典型值	最大值	单位
1	用于擦/写操作的 VDD	-	2.1	-	5.5	V
2	用于读操作的 VDD	-	2.1	-	5.5	V
3	工作频率 [1]	-		-		kHz
4	Flash 自写 字节耐用性 字节耐用性 擦/写周期	 -40℃≤TA≤+85℃ 85℃≤TA≤+105℃ -	 10k 1k -	 100k 10k 3	 - - -	 Cycles Cycles ms
5	DATA EEPROM 单元耐用性 单元耐用性 擦/写周期	 -40℃≤TA≤+85℃ 85℃≤TA≤+105℃ -	 100k 10k -	 1M 100k 6	 - - -	 Cycles Cycles ms
6	Data retention	假设没有违反其他规范	40	-	-	years

注：

1.工作频率需要通过软件配置.

2.为了保证 FLASH 自写和 DATA EEPROM 工作可靠性，必须使能欠压检测功能。

20 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运行，才可以确保正常工作。

图20.1a： 不同**VDD**时KF8F3110典型**IDD** — **FOSC** 关系曲线图

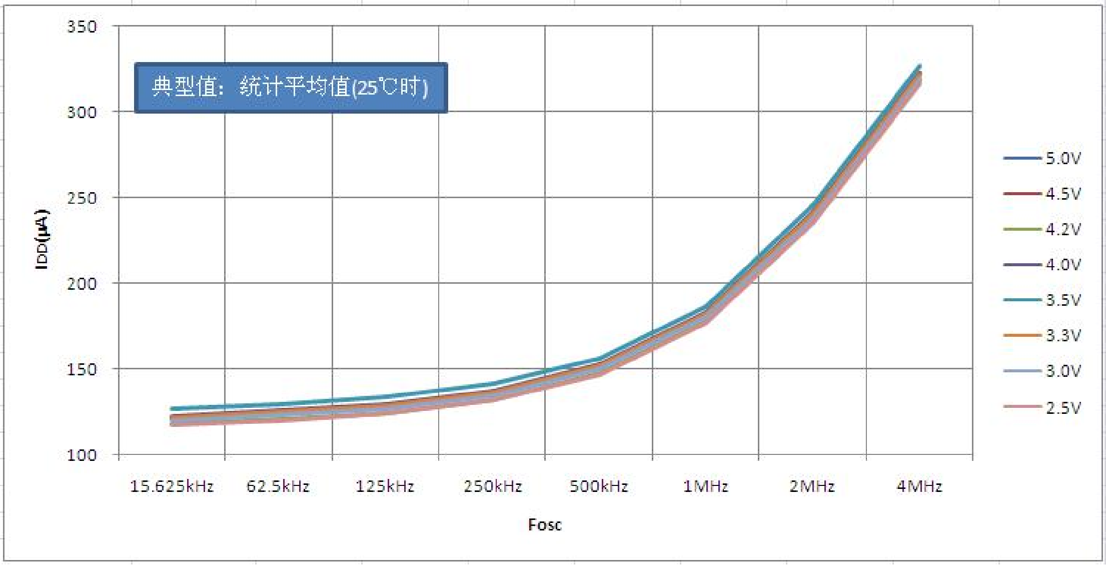


图20.1b： 不同**VDD**时KF8F3112典型**IDD** — **FOSC** 关系曲线图

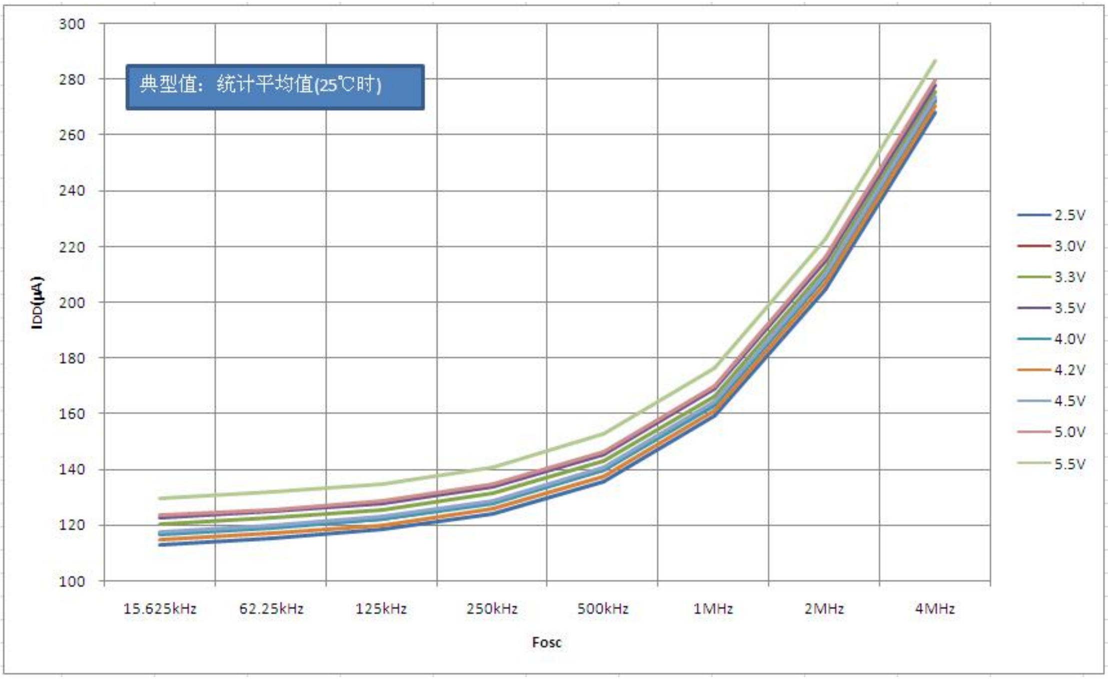


图20.1c: 不同VDD时KF8F3120/22典型IDD — FOSC 关系曲线图

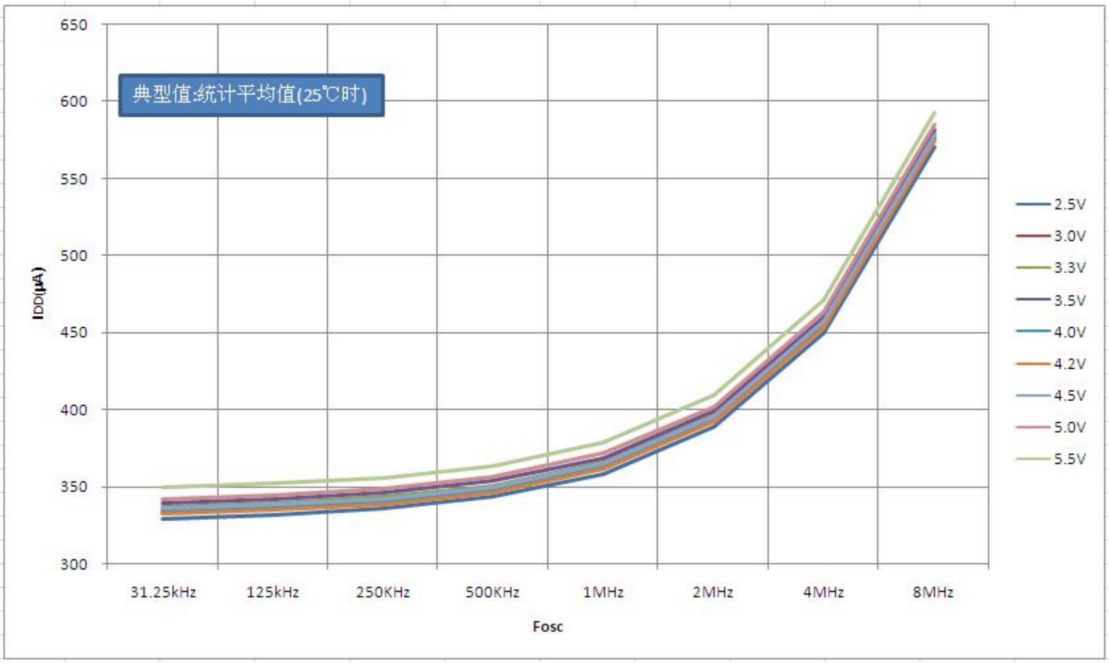


图20.1d: 不同VDD时KF8F3130/32典型IDD — FOSC 关系曲线图

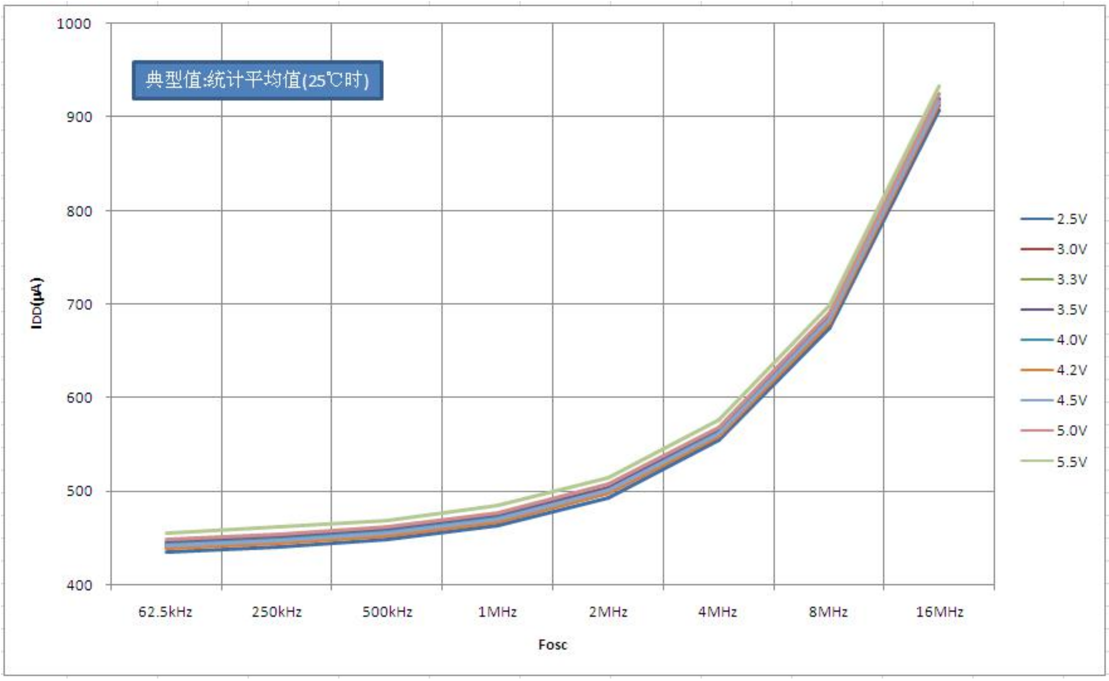


图 20.2： 不同 VDD 时典型 IIDLE － VDD 关系曲线图

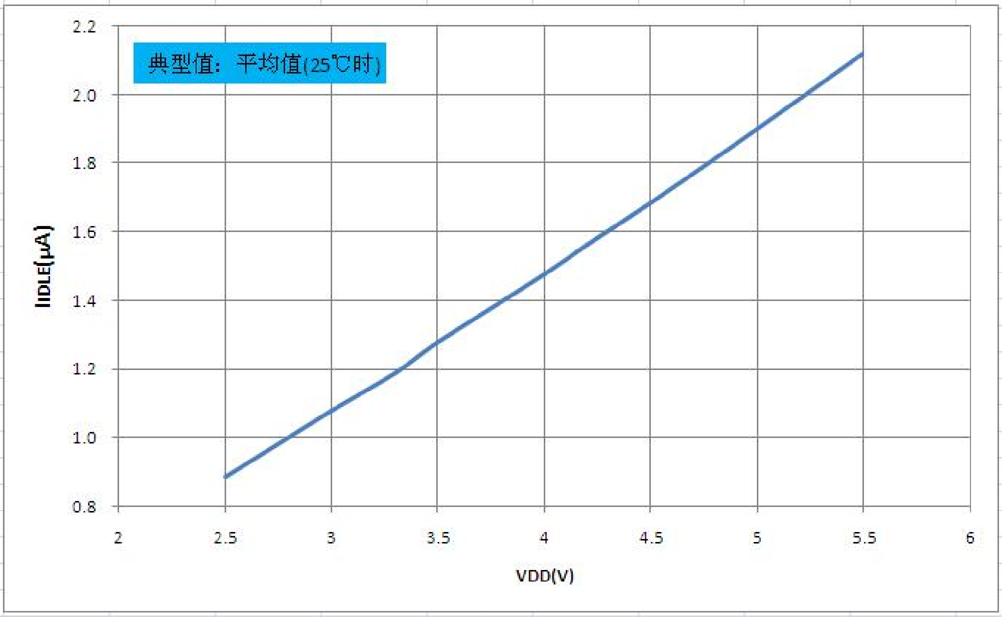
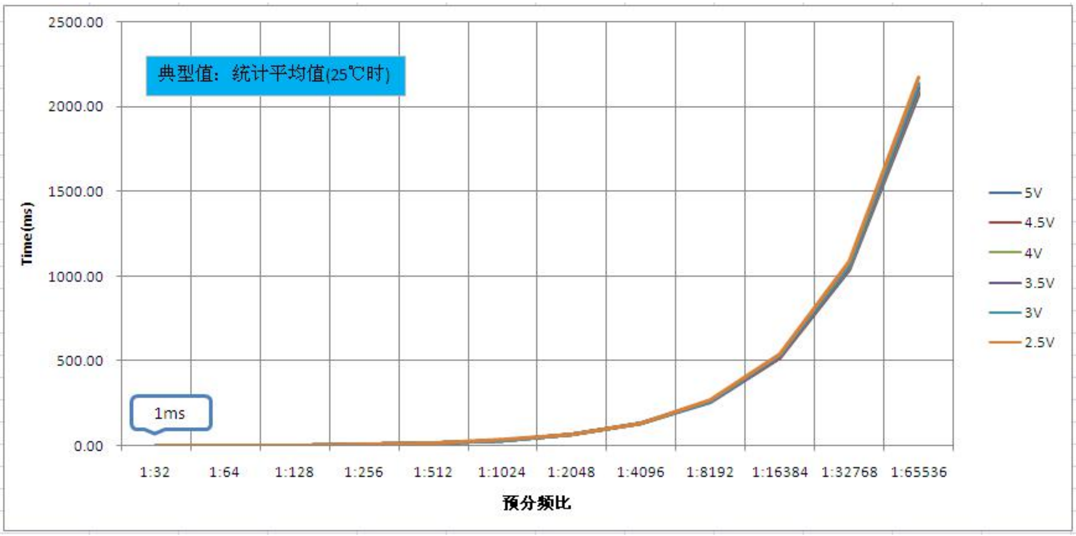


图20.3：　不同**VDD**时看门狗预分频比 －周期关系曲线图



注：1/32 分频时看门狗周期为 1ms。

图20.4：看门狗电流 － **VDD**关系曲线图

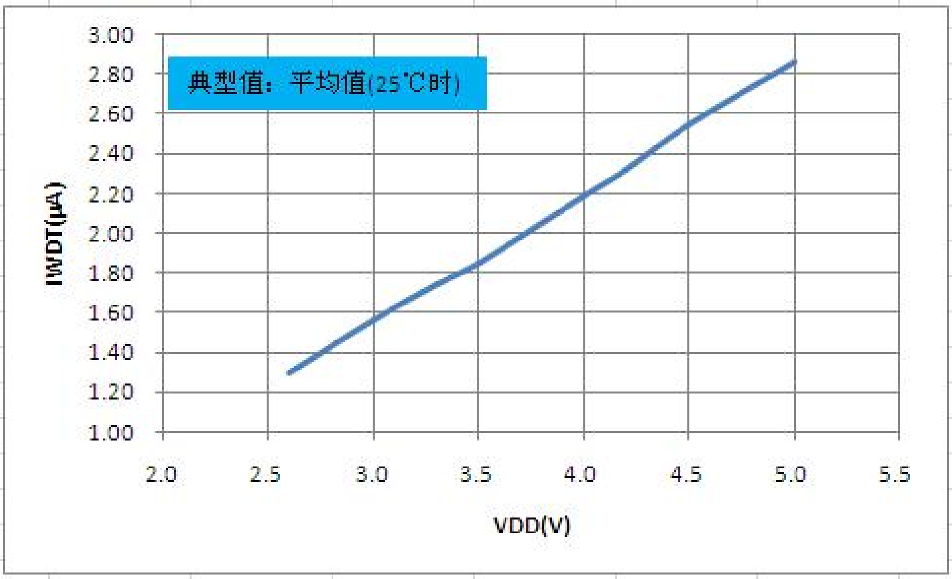


图 20.5：欠压检测复位电流 ILVR － VDD 关系曲线图

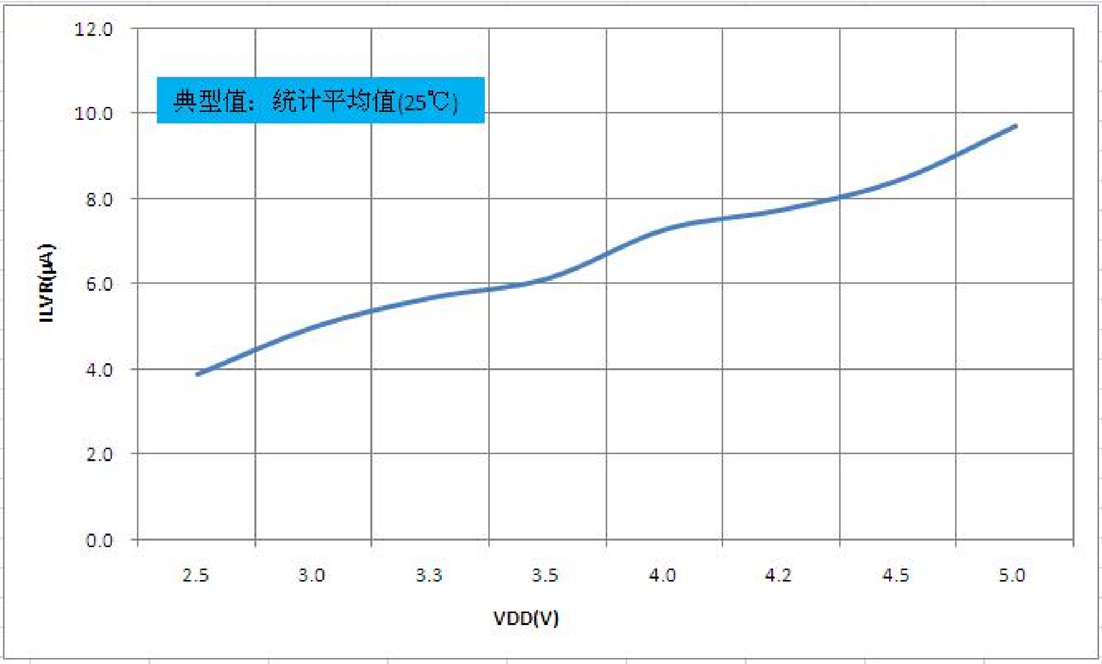


图20.6：不同温度时VOH — IOH 关系曲线图（VDD＝5.0V）

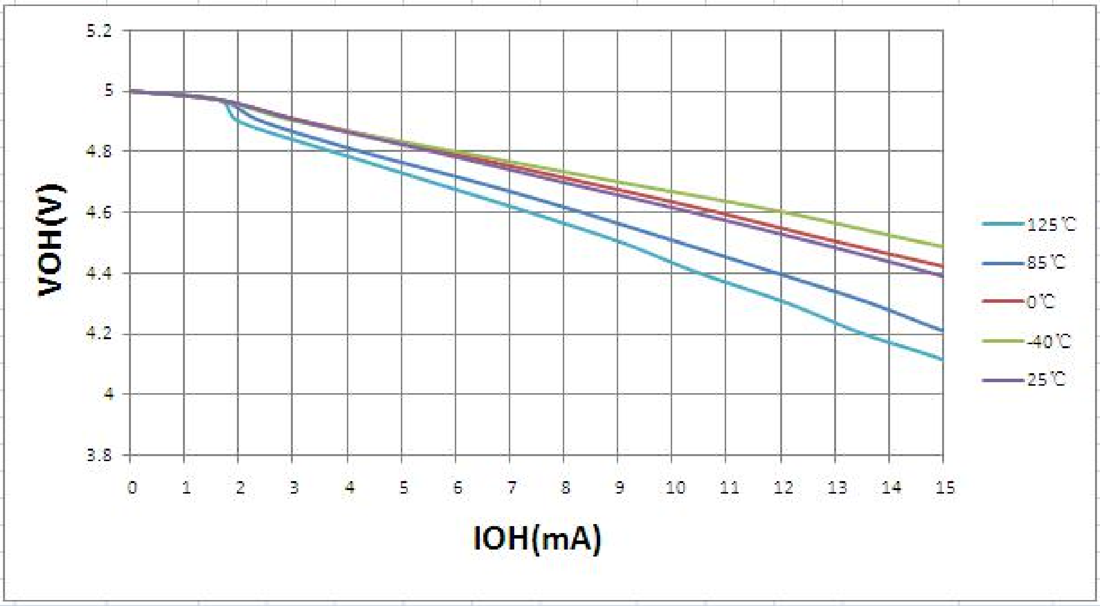


图20.7：不同温度时VOH — IOH 关系曲线图（VDD＝3.0V）

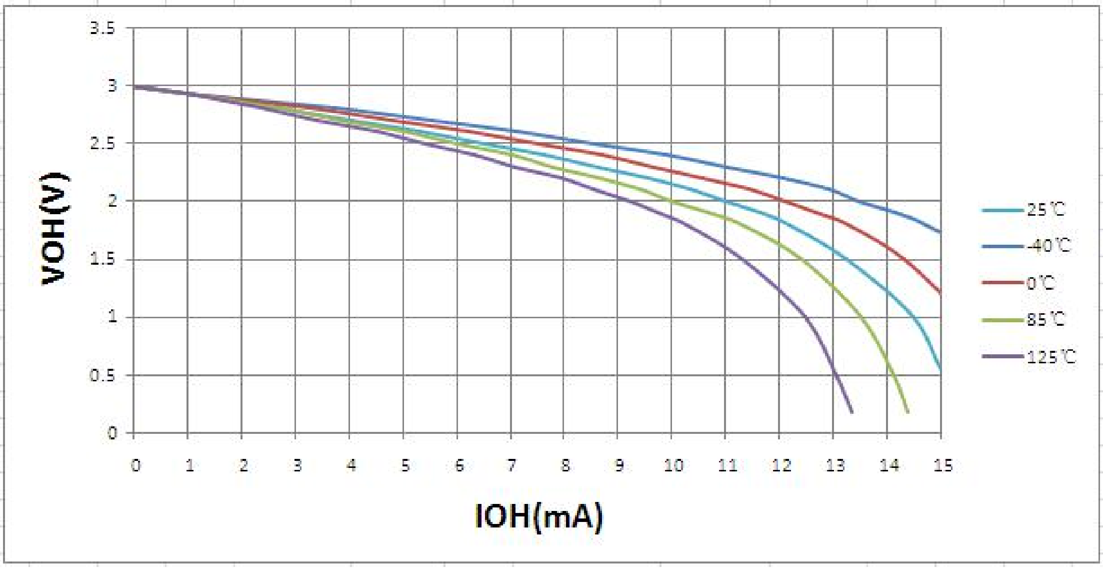


图20.8：不同温度时VOL － IOL 关系曲线图（VDD＝5.0V）

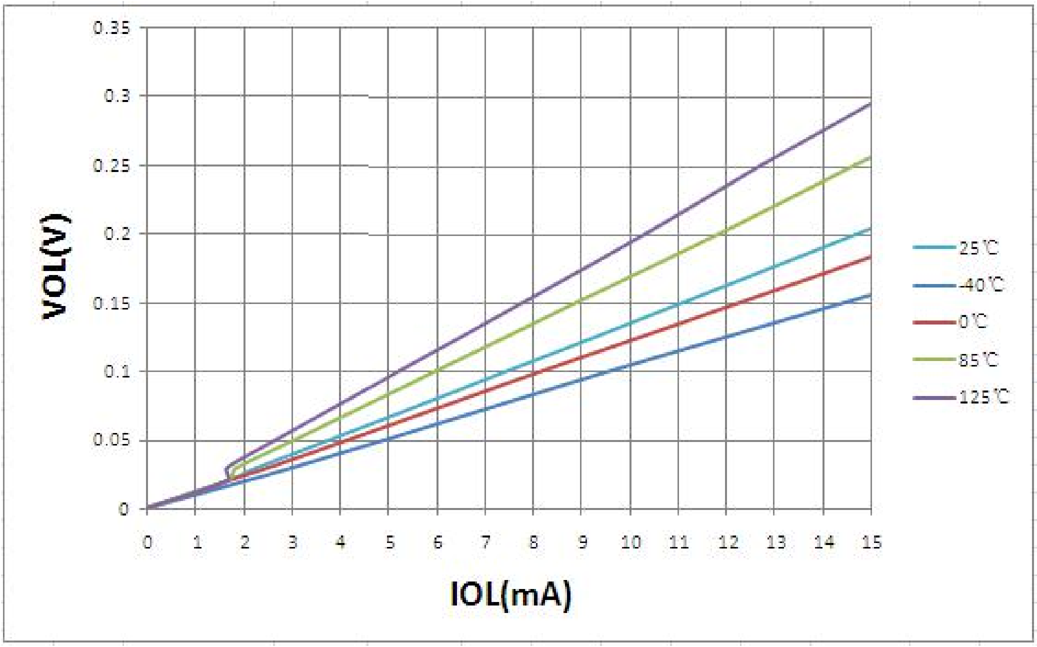


图20.9：不同温度时VOL － IOL 关系曲线图（VDD＝3.0V）

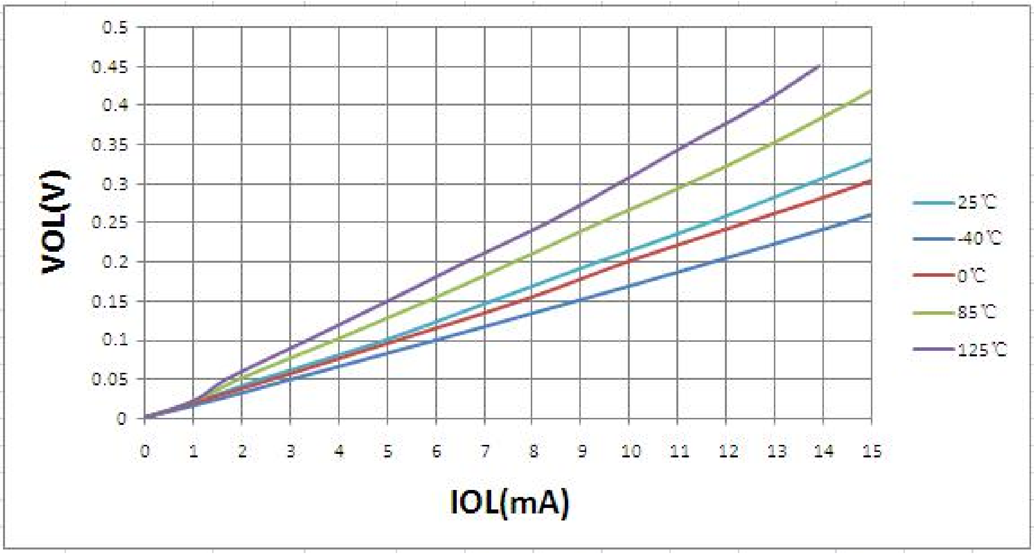


图20.10：不同温度时VOP－IOP 关系曲线图（VDD =5.0V）

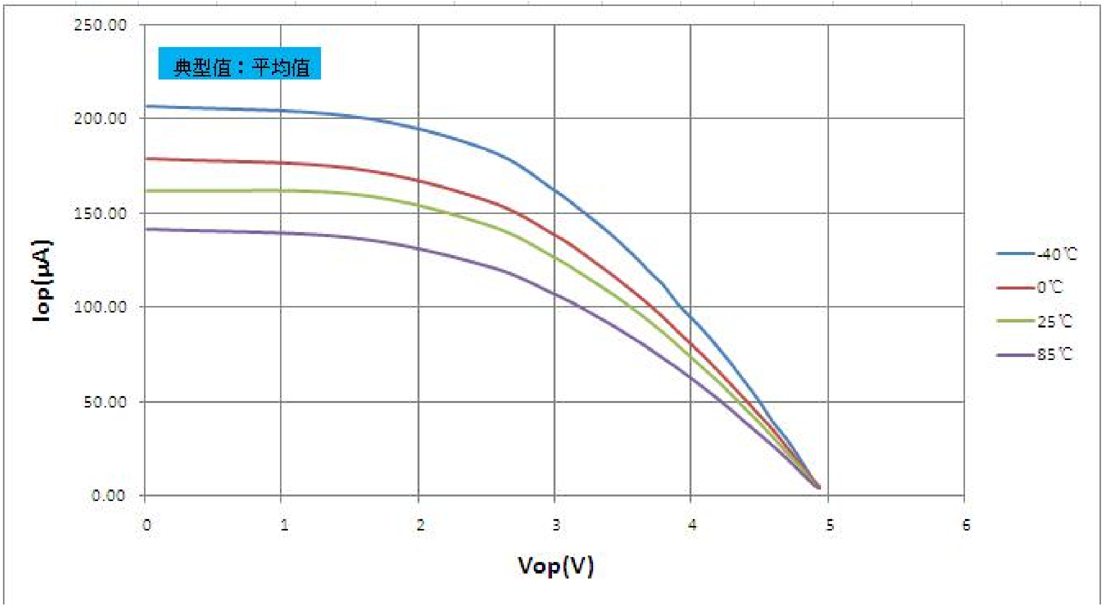
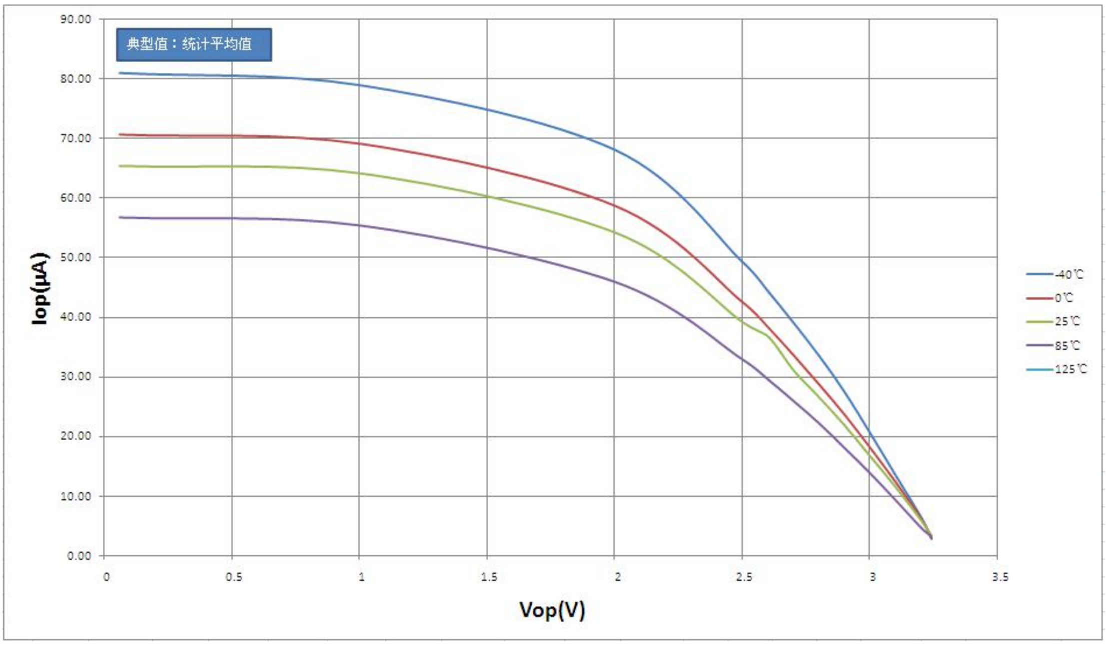
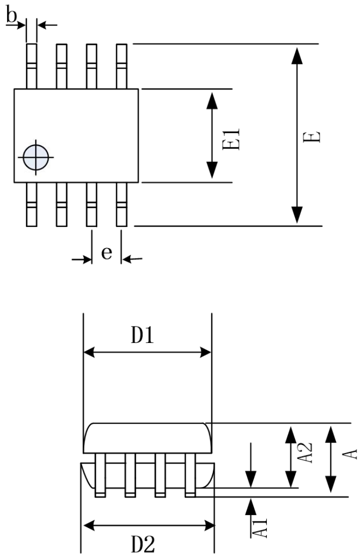


图 20.11：不同温度时 VOP－IOP 关系曲线图（VDD =3.3V）

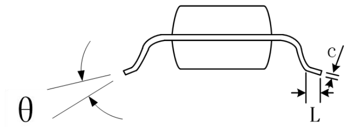


21 封装信息

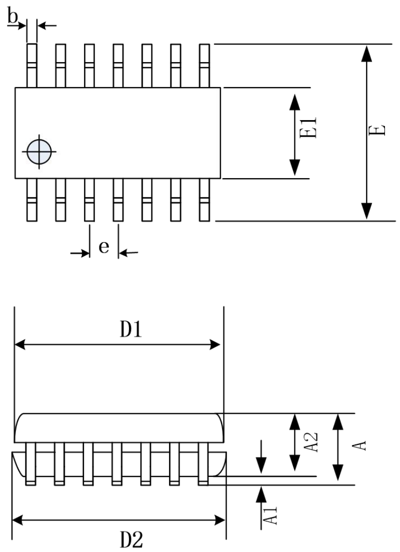
8 脚 SOIC 封装



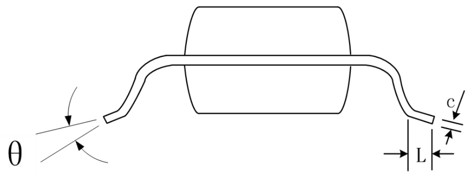
SIZE (Unit: mm)			
TAB	Min	Nom	Max
A	1.500	1.600	1.700
A1	0.050	0.150	0.250
A2		1.450	
b	0.356	0.406	0.5
c		0.220	
D1	4.790	4.840	4.890
D2	4.830	4.880	4.930
E	5.900	6.000	6.100
E1	3.860	3.910	3.960
e		1.270	
L	0.650	0.660	0.670
θ	0	—	10°



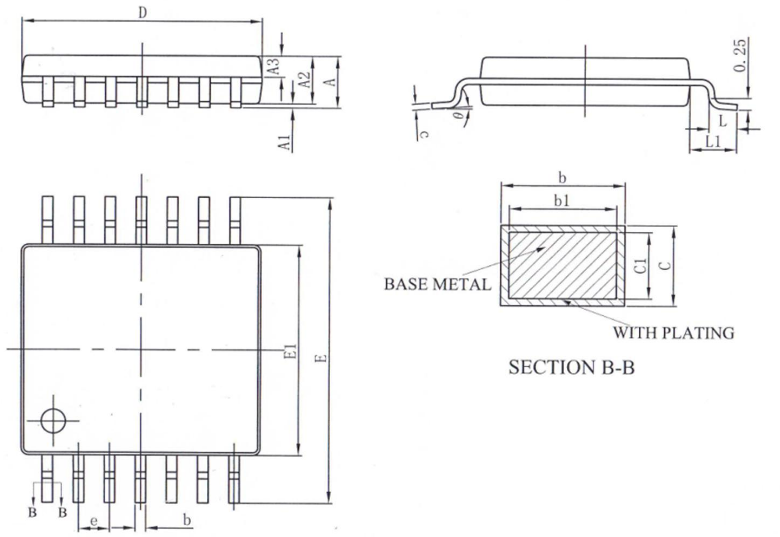
14 脚 SOIC 封装



SOIC14封装尺寸（单位：毫米）			
标号	最小	一般	最大
A	1.500	1.600	1.700
A1	0.100	0.150	0.200
A2	1.400	1.450	1.500
b	0.356	0.406	0.456
c		0.223	
D1	8.600	8.650	8.800
D2	8.610	8.660	8.710
E	5.900	6.000	6.100
E1	3.850	3.950	4.050
e		1.270	
L	0.560	0.660	0.760
θ	0	—	10°

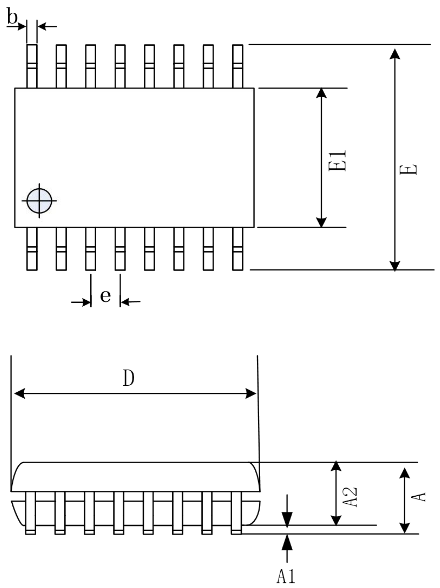


14 脚 TSSOP 封装

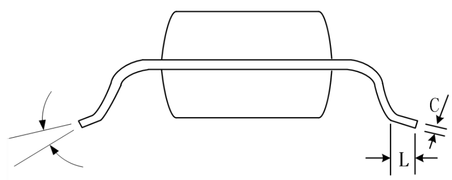


SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.20
A1	0.05	—	0.15
A2	0.90	1.00	1.05
A3	0.39	0.44	0.49
b	0.20	—	0.28
b1	0.19	0.22	0.25
c	0.13	—	0.17
c1	0.12	0.13	0.14
D	4.90	5.00	5.10
E1	4.30	4.40	4.50
E	6.20	6.40	6.60
e	0.65BSC		
L	0.45	0.60	0.75
L1	1.00BSC		
θ	0	—	8°

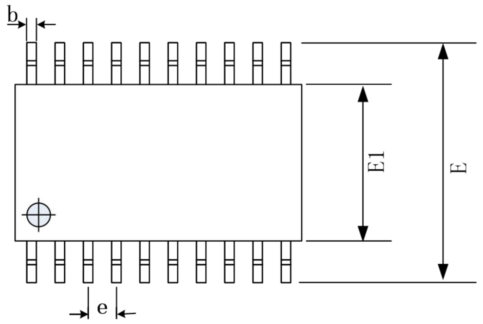
16 脚 SOIC 封装



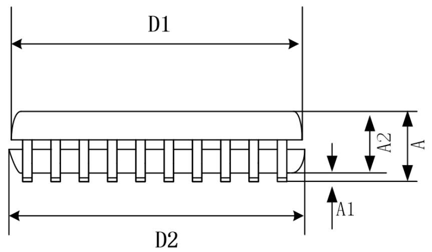
SOIC16封装尺寸（单位：毫米）		
标号	最小	最大
A	1.35	1.75
A1	0.100	0.25
A2	1.25	1.500
b	0.31	0.51
c	0.17	0.25
D	9.80	10.20
E	5.80	6.20
E1	3.80	4.00
e	1.27	
L	0.40	1.27
θ	0	8°

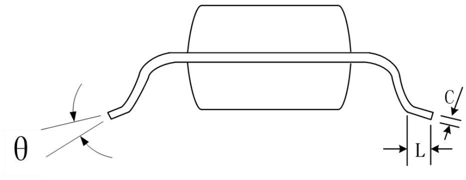


20 脚 SOIC 封装

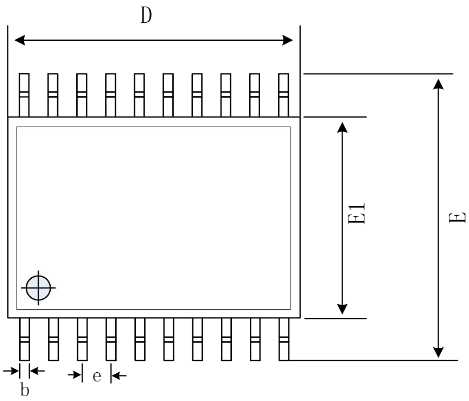


SOIC20封装尺寸(单位：毫米)			
标号	最小	一般	最大
A	2.465	2.515	2.565
A1	0.065	0.215	0.365
A2	2.100	2.300	2.500
b	0.356	0.406	0.456
c		0.274	
D1	12.500	12.700	12.900
D2	12.550	12.750	12.950
E	10.206	10.306	10.406
E1	7.320	7.500	7.550
e		1.270	
L	0.800	0.864	0.900

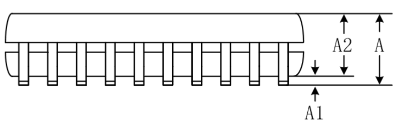


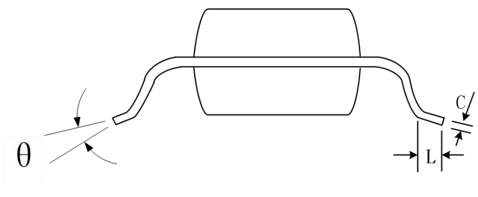


20 脚 SSOP 封装

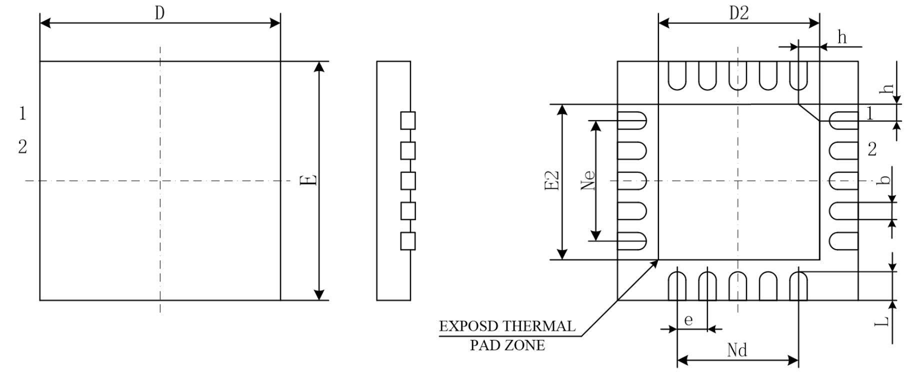


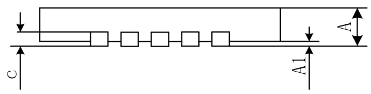
SSOP20封装尺寸（单位：毫米）			
标号	最小	一般	最大
A	1.600	1.650	1.850
A1	0.05	0.150	0.250
A2	1.400	1.500	1.600
b	0.220	0.300	0.380
C	0.090	0.150	0.25
D	7.000	7.200	7.400
E	7.600	7.800	8.000
E1	5.100	5.300	5.500
e		0.650	
L	0.550	0.900	1.050
θ	0	—	10°





20 脚 QFN 封装





符号	封装尺寸（单位：毫米）		
	最小	一般	最大
A	0.70	0.75	0.80
A1	-	0.02	0.05
b	0.18	0.25	0.30
c	0.18	0.20	0.25
D	3.90	4.00	4.10
D2	2.55	2.65	2.75
e	0.50BSC		
Ne	2.00BSC		
Nd	2.00BSC		
E	3.90	4.00	4.10
E2	2.55	2.65	2.75
L	0.35	0.40	0.45
h	0.30	0.35	0.40

附录 1 特殊功能寄存器（SFR）功能汇总

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	复位初值
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	-	-	-	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	CY	0001 1xxx
05H	P0	-	-	P05	P04	P03	P02	P01	P00	-- xx xxxx
06H	P2	-	-	-	-	P23	P22	P21	P20	---- xxxx
07H	P1	P17	P16	P15	P14	P13	P12	P11	P10	xxxx xxxx
0AH	PCH	-	-	-	程序计数器(PC)高字节					---0 0000
0BH	INTCTL	AIE/ AIEH	PUIE /AIEL	T0IE	INT0IE	P0IE	T0IF	INT0IF	P0IF	0000 0000
0CH	EIF1	EEIF	ADIF	INT2IF	INT1IF	C1IF	PWM2IF	T2IF	T1IF	0000 0000
0DH	EIF2	T3IF	C2IF	RCIF	TXIF	-	CCPIF	BCLIF	SSCIIF	0000 0000
0EH	T1L	定时/计数器 T1 低字节寄存器								0000 0000
0FH	T1H	定时/计数器 T1 高字节寄存器								0000 0000
10H	T1CTL	T1RLD	T1GC	T1CKS1	T1CKS0	T1OSCEN	$\overline{\text{T1S}}$	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
14H	PWM1H	PWM1 占空比缓冲寄存器（与用户无关）								xxxx xxxx
15H	PWMCTL	PWM24ON	PWM23ON	PWM22ON	PWM2ON	-	-	-	PWM1ON	0000 0000
16H	PP1	PWM1 周期寄存器								1111 1111
17H	BANK	-	-	-	-	PR3	PR2	PR1	PR0	---- 0000
18H	ADSCANCTL	ADSCANEN	SMOD1	SMOD0	STIM4	STIM3	STIM2	STIM1	STIM0	0000 0000
19H	C1CTL	C1OE	C1MOD2	C1MOD1	C1MOD0	C1CALEN	C1CALSET	C1OUT	C1EN	0000 0000
1AH	COUT	-	VCEN	C2OE	C2MOD1	C2MOD0	-	-	C2OUT	00-- -000
1BH	AMPCTL	AMPCALD	-	INPS1	INPS0	-	-	AMPCALEN	AMPON	1-00 0000
1DH	ANSEH	-	-	ANS13	ANS12	ANS11	ANS10	ANS9	ANS8	--00 0000
1EH	ADCDATA0H	ADC 数据寄存器 0 高字节								xxxx xxxx
1FH	ADCCTL0	ADLR	T2CCR0ON	CHS3	CHS2	CHS1	CHS0	START	ADEN	0000 0000
20H	AMPDT	AMPDT7	AMPDT6	AMPDT5	AMPDT4	AMPDT3	AMPDT2	AMPDT1	AMPDT0	1000 0000
21H	OPTR	$\overline{\text{PUPH}}$	INT0SE	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111
22H	IP0	-	-	-	-	-	PT0	PINT0	PP0	---- -000
23H	IP1	PEE	PADC	PINT2	PINT1	PC1	PPWM2	PT2	PT1	0000 0000
24H	IP2	PT3	PC2	PRC	PTX	-	PCCP	PBCL	PSSCI	0000 0000
25H	TR0	-	-	TR05	TR04	TR03	TR02	TR01	TR00	1111 1111
26H	TR2	-	-	-	-	TR23	TR22	TR21	TR20	1111 1111
27H	TR1	TR17	TR16	TR15	TR14	TR13	TR12	TR11	TR10	1111 1111
28H	OSCSTA	-	OSTS	HTS	LTS	-	-	SCF1	SCF0	0110 --00
29H	IP3	-	-	POSCFAIL	-	-	-	-	-	000- ----
2AH	VRECAL1	内核电压校准寄存器 1								0111 0111
2BH	VRECTL	VRESEL1	VRESEL0	-	VRECKEN	VREOE	-	VREEN	-	0000 0000
2CH	EIE1	EEIE	ADIE	INT2IE	INT1IE	C1IE	PWM2IE	T2IE	T1IE	0000 0000
2DH	EIE2	T3IE	C2IE	RCIE	TXIE	-	CCPIE	BCLIE	SSCIIE	0000 0000
2EH	PCTL	-	-	-	SLVREN	IPEN	SWDTEN	$\overline{\text{POR}}$	$\overline{\text{LVR}}$	---1 000x
2FH	OSCCTL	CLKOE	IRCS2	IRCS1	IRCS0	SCS1	SCS0	IESO	FSCM	0010 0000
30H	OSCCAL0	晶振校准值寄存器 0								1000 0000
31H	ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	0000 0000
32H	PP2	PWM2 周期寄存器								1111 1111
33H	PWM2L	PWM2 占空比设置寄存器								xxxx xxxx
34H	PWM2H	PWM2 占空比缓冲寄存器（与用户无关）								xxxx xxxx
35H	PUR0	-	-	PUR05	PUR04	-	PUR02	PUR01	PUR00	1111 -111
36H	IOCL	-	-	IOCL5	IOCL4	IOCL3	IOCL2	IOCL1	IOCL0	0000 0000
37H	OSCCAL1	晶振校准值寄存器 1								0000 -001
38H	NVMDATAH	NVM 数据高 8 位								0000 0000
39H	NVMDATAL	NVM 数据低 8 位								0000 0000
3AH	NVMADDRH	NVM 地址指针高 8 位								0000 0000
3BH	NVMADDRL	NVM 地址指针低 8 位								0000 0000
3CH	NVMCTL0	NVM 控制寄存器 0								---- x000
3DH	NVMCTL1	NVM 控制寄存器 1								---- ----
3EH	ADCDATA0L	ADC 数据寄存器 0 低字节								xxxx xxxx
3FH	ADCCTL1	ADCALEN	ADCS2	ADCS1	ADCS0	VCFG1	VCFG0	-	ADCIM	0000 0000
40H	T2CCR0H	T2 启动 ADC 设置寄存器 0 高 8 位								0000 0000
41H	T2H	T2 高 8 位								0000 0000
42H	PP5H	PWM5 周期寄存器高 8 位								0000 0000
43H	PWM5H0	PWM5 占空比寄存器高 8 位								0000 0000

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	复位初值
44H	PWM5H1	PWM5 占空比缓冲寄存器高 8 位（与用户无关）								0000 0000
45H	P0LR	-	-	P0LR5	P0LR4	-	P0LR2	P0LR1	P0LR0	xxxx xxxx
46H	P2LR	-	-	-	-	P2LR3	P2LR2	P2LR1	P2LR0	xxxx xxxx
47H	P1LR	P1LR7	P1LR6	P1LR5	P1LR4	P1LR3	P1LR2	P1LR1	P1LR0	xxxx xxxx
4AH	EIE3	-	-	OSCFAILIE	-	-	-	-	-	000- ----
4BH	EIF3	-	-	OSCFAILIF	-	-	-	-	-	000- ----
4CH	OSCCAL2	晶振校准值寄存器 2								0011 1111
4DH	OSCCAL3	晶振校准值寄存器 3								0011 1111
4EH	T3CTL	T3EN	T3CKS1	T3CKS0	T3CS	-	-	-	-	0000 0000
4FH	T3L	定时器 3 低字节寄存器								0000 0000
50H	ADCDATA1H	ADC 数据寄存器 1 高字节								xxxx xxxx
51H	ADCDATA1L	ADC 数据寄存器 1 低字节								xxxx xxxx
52H	PP5L	PWM5 周期寄存器低 8 位								0000 0000
53H	T3CTL1	-	-	T3CLR	-	-	-	-	-	0000 0000
54H	T2CCR0L	T2 启动 ADC 设置寄存器 0 低 8 位								0000 0000
55H	PWM5L0	捕捉比较模式时，PWM5L0=CCP 数据寄存器低 8 位 PWM5 模式时，PWM5L0= PWM5 占空比设置寄存器低 8 位								0000 0000
56H	PWM5L1	捕捉比较模式时，PWM5L1=CCP 数据寄存器高 8 位 PWM5 模式时，PWM5L1= PWM5 低 8 位占空比缓冲寄存器（与用户无关）								0000 0000
57H	PWM5CTL0	P5CH1MOD1	P5CH1MOD0	-	-	P5MOD3	P5MOD2	P5MOD1	P5MOD0	00-- 0000
58H	ADCSICM	-	-	-	-	-	-	ADCSICM1	ADCSICM0	0000 0000
59H	ADCDATA2H	ADC 数据寄存器 2 高字节								xxxx xxxx
5AH	ADCDATA2L	ADC 数据寄存器 2 低字节								xxxx xxxx
5BH	PWM5CTL1	P5RSEN	P5DC6	P5DC5	P5DC4	P5DC3	P5DC2	P5DC1	P5DC0	0000 0000
5CH	P5ASCTL	P5ASE	P5ASS2	P5ASS1	P5ASS0	P5SSAC1	P5SSAC0	P5SSBD1	P5SSBD0	0000 0000
5DH	PSTRCTL	-	-	-	STRSYNC	STREND	STRENC	STRENB	STRENA	---0 0001
5EH	ADCDATA3H	ADC 数据寄存器 3 高字节								xxxx xxxx
5FH	T3H	定时器 3 高字节寄存器								0000 0000
60H	PUR1	PUR17	PUR16	PUR15	PUR14	PUR13	PUR12	PUR11	PUR10	1111 1111
61H	PUR2	-	-	-	-	PUR23	PUR22	PUR21	PUR20	1111 1111
62H	PWM12	PWM22 占空比设置寄存器低 8 位								xxxx xxxx
63H	PWM13	PWM23 占空比设置寄存器低 8 位								xxxx xxxx
65H	PWM22	PWM22 占空比设置寄存器高 8 位								xxxx xxxx
66H	PWM23	PWM23 占空比设置寄存器高 8 位								xxxx xxxx
67H	INTEDGCTL	INT2SE	INT1SE	-	-	-	-	-	T1CLKEN	11-- ---0
68H	PWM14	PWM24 占空比设置寄存器低 8 位								xxxx xxxx
69H	PWM24	PWM24 占空比设置寄存器高 8 位								xxxx xxxx
6AH	ADCDATA3L	ADC 数据寄存器 3 低字节								xxxx xxxx
10EH	PWM5OC	-	-	-	-	-	-	OCA	OCB	--00 0000
10FH	T2CTL1	T2MOD1	T2MOD0	T2OFS1	T2OFS0	T2TRIGEN	-	-	T2DIR	0000 0--x
112H	C2CTL	C2M3	C2M2	C2M1	C2M0	C2CALEN	C2CALSET	C2MOD2	C2EN	0000 0000
115H	C1CAL	C1DT7	C1DT6	C1DT5	C1DT4	C1DT3	C1DT2	C1DT1	C1DT0	0000 0000
116H	C2CAL	C2DT7	C2DT6	C2DT5	C2DT4	C2DT3	C2DT2	C2DT1	C2DT0	0000 0000
11CH	PWM5CTL2	-	-	PFUSES	-	-	UDEVT1	UDEVT0	UDEN	--1- -000
11DH	PWM5PC	-	-	-	-	-	-	PCA	PCB	--00 0000
120H	RSCTL	SPEN	RX9	SRXEN	CRXEN	ADREN	FRER	OVFER	RX9D	0000 0000
121H	TXSDR	USART 发送数据寄存器								0000 0000
122H	RXSDR	USART 接收数据寄存器								0000 0000
123H	BRCTL	ABRDOVF	RCIDLF	SCKPS	BRG16	-	-	WUEN	ABRDEN	0100 0000
124H	TSCTL	CSRS	TX9	TXEN	SYNC	SENDB	HBRG	TXSRS	TX9D	0000 0010
125H	EUBRGL	USART 波特率数据寄存器低字节								0000 0000
126H	EUBRGH	USART 波特率数据寄存器高字节								0000 0000
128H	SSCICTL0	SSCIWCFL	SSCIOV	SSCIEN	SSCICKP	SSCIMOD3	SSCIMOD2	SSCIMOD1	SSCIMOD0	0000 0000
12AH	SSCICTL1	SSCICALLEN	SSCIACKSTA	SSCIACKDAT	SSCIACKEN	SSCIRCEN	STOPEN	RESTARTEN	STARTEN	0000 0000
12BH	SSCISTA	SAMPLE	CKEGE	SSCIDA	SSCISTOP	SSCISTART	SSCIRW	SSCIUA	SSCIBUF	0000 0000
12CH	SSCIBUFR	SSCI 数据接收缓冲/发送寄存器								xxxx xxxx
12EH	SSCIADD	SSCI 的 I2C 地址寄存器								0000 0000
	SSCIMSK	SSCIMSK7	SSCIMSK6	SSCIMSK5	SSCIMSK4	SSCIMSK3	SSCIMSK2	SSCIMSK1	SSCIMSK0	1111 1111
12FH	WDTPS	-	-	-	-	WDTPS3	WDTPS2	WDTPS1	WDTPS0	---- 0100
130H	MULAH	16 位乘法器乘数 A 高 8 位								0000 0000
131H	MULAL	16 位乘法器乘数 A 低 8 位								0000 0000
132H	MULBH	16 位乘法器乘数 B 高 8 位								0000 0000
133H	MULBL	16 位乘法器乘数 B 低 8 位								0000 0000
134H	MULCTL	-	-	-	-	-	-	MULEN	MULIF	---- --00
135H	MULRES3	乘法运算结果寄存器 3								0000 0000
136H	MULRES2	乘法运算结果寄存器 2								0000 0000
137H	MULRES1	乘法运算结果寄存器 1								0000 0000

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	复位初值
138H	MULRES0	乘法运算结果寄存器 0								0000 0000
139H	DIVCTL	-	-	-	-	-	-	DIVOEN	DIVEN	---- --00
13AH	DIVAH	被除数高 8 位寄存器								0000 0000
13BH	DIVAL	被除数低 8 位寄存器								0000 0000
13CH	DIVB	除数寄存器								0000 0000
13DH	DIVQH	商高 8 位寄存器								0000 0000
13EH	DIVQL	商低 8 位寄存器								0000 0000
13FH	DIVR	余数寄存器								0000 0000
14AH	C1FILTCTL	INV1	WEN1	FEN1	-	-	F1CNT2	F1CNT1	F1CNT0	0000 0000
14BH	C1FILTPRE	FP17	FP16	FP15	FP14	FP13	FP12	FP11	FP10	0000 0000
14CH	C2FILTCTL	INV2	WEN2	FEN2	-	-	F2CNT2	F2CNT1	F2CNT0	0000 0000
14DH	C2FILTPRE	FP27	FP26	FP25	FP24	FP23	FP22	FP21	FP20	0000 0000
151H	CCRL	捕捉/比较寄存器低 8 位								0000 0000
152H	CCPCTL	-	-	-	CCPPIN	CCMOD3	CCMOD2	CCMOD1	CCMOD0	---0 0000
154H	CCRH	捕捉/比较寄存器高 8 位								0000 0000
156H	PINSET	SSCIPIN	SDOPIN	SSPIN	UARTPIN	PWMPIN	-	-	-	0000 0000
159H	VRECAL2	内部参考电压校准寄存器 2								0000 0000
15AH	VRECAL3	内部参考电压校准寄存器 3								1010 0000
15DH	PWM5FC	-	-	-	-	-	-	FCA	FCB	--00 0000
165H	RC32KCAL	内部 32K 振荡器频率校验寄存器								0000 0000

注:“-”表示未用的存储单元 “x”表示不定

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	
	数据传送指令			
MOV dir	0000_1111_ffff_ffff	dir←(dir)	1	Z
MOV Rn,dir	0101_rrr0_ffff_ffff	Rn←(dir)	1	
MOV dir,Rn	0101_rrr1_ffff_ffff	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_ffff_ffff	Rn<7:4>=dir<3:0> Rn<3:0>=dir<7:4>	1	
SWAP dir	0100_rrr1_ffff_ffff	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	
	算术运算指令			
ADD Rm,dir	0010_0rr0_ffff_ffff	Rm←(Rm)+(dir)	1	CY、DC、Z
ADD dir,Rm	0010_0rr1_ffff_ffff	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_ffff_ffff	Rm←(dir)-(Rm)	1	CY、DC、Z
SUB dir,Rm	0011_1rr1_ffff_ffff	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_ffff_ffff	dir←(dir)+1	1	Z
INCR dir	0000_1010_ffff_ffff	R0←(dir)+1	1	Z
INC Rn	1111_1111_0001_0rrr	Rn←(Rn)+1	1	Z
DEC dir	0000_0111_ffff_ffff	dir←(dir)-1	1	Z
DECR dir	0000_0110_ffff_ffff	R0←(dir)-1	1	Z
DEC Rn	1111_1111_0000_1rrr	Rn←(Rn)-1	1	Z
	逻辑运算指令			
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

助记符、操作数	指令格式	指令说明	周期	影响标志
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
	位操作指令			
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	
	转移指令			
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 寄存器全称表

BANK0		
地址	名称	全称
01H	T0	Timer 0 Register
02H	PCL	Program Counter Low Register
03H	PSW	Program Status Word Register
05H	P0	Port 0
06H	P2	Port 2
07H	P1	Port 1
0AH	PCH	Program Counter High Register
0BH	INTCTL	Interrupt control Register
0CH	EIF1	Enable Interrupt Flag Register 1
0DH	EIF2	Enable Interrupt Flag Register 2
0EH	T1L	Timer 1 Register Low
0FH	T1H	Timer 1 Register High
10H	T1CTL	Timer 1 Control Register
11H	T2L	Timer 2 Register Low
12H	T2CTL0	Timer 2 Register Control 0
13H	PWM1L	Pulse-Width Modulation Duty Cycle Register 1 Low
14H	PWM1H	Pulse-Width Modulation Duty Cycle Register 1 High
15H	PWMCTL	Pulse-Width Modulation Control Register
16H	PP1	Pulse-Width Modulation Periods Register 1
17H	BANK	Bank
18H	ADSCANCTL	Analog Digital Convert Scan Control Register
19H	C1CTL	Comparer 1 Control Register
1BH	AMPCTL	Amplifier Control Register
1CH	VRECAL0	Core Voltage Calibration Register 0
1DH	ANSEH	Analog channel Selection Register High
1EH	ADCDATA0H	Analog Digital Convert Data 0 High Register
1FH	ADCCTL0	Analog Digital Convert Control Register 0
20H	AMPDT	Amplifier Data Register
21H	OPTR	Option Register
22H	IP0	Interrupt Priority 0 Register
23H	IP1	Interrupt Priority 1 Register
24H	IP2	Interrupt Priority 2 Register
25H	TR0	Tri Register 0
26H	TR2	Tri Register 2
27H	TR1	Tri Register 1
28H	OSCSTA	Oscillator Status Register
29H	IP3	Interrupt Priority 3 Register
2AH	VRECAL1	Reference Voltage Calibration Register 1
2BH	VRECTL	Reference Voltage Control Register
2CH	EIE1	Enable Interrupt Enable Register 1
2DH	EIE2	Enable Interrupt Enable Register 2
2EH	PCTL	Power Control Register
2FH	OSCCTL	Oscillator Control Register
30H	OSCCAL0	Oscillator Calibration Register 0
31H	ANSEL	Analog channel Selection Register Low
32H	PP2	Pulse-Width Modulation Periods Register 2
33H	PWM2L	Pulse-Width Modulation Duty Cycle Register 2 Low
34H	PWM2H	Pulse-Width Modulation Duty Cycle Register 2 High

35H	PUR0	Pull-Up Register 0
36H	IOCL	P0 Interrrupt on Change Register
37H	OSCCAL1	Oscillator Calibration Register 1
38H	NVMDATAH	NVW Buffer Data Register High
39H	NVMDATAL	NVM Buffer Data Register Low
3AH	NVMADDRH	NVM Buffer Address Register High
3BH	NVMADDRL	NVM Buffer Address Register Low
3CH	NVMCTL0	Nonvolatile Memory control Register 1
3DH	NVMCTL1	Nonvolatile Memory control Register 2
3EH	ADCDATA0L	Analog Digital Convert Data 0 Register Low
3FH	ADCCTL1	Analog Digital Convert Control Register 1
40H	T2CCR0H	Timer2 Compare Capture Register 0 High
41H	T2H	Timer2 High
42H	PP5H	Pulse-Width Modulation 5 Periods Register High
43H	PWM5L1	Pulse-Width Modulation 5 Duty Cycle Register Low 1
44H	PWM5H1	Pulse-Width Modulation 5 Duty Cycle Register High 1
45H	P0LR	Port 0 Latch Register
46H	P2LR	Port 2 Latch Register
47H	P1LR	Port 1 Latch Register
4AH	EIE3	Enable Interrupt Register 3
4BH	EIF3	Enable Interrupt Flag Register 3
4CH	OSCCAL2	Oscillator Calibration Register 2
4DH	OSCCAL3	Oscillator Calibration Register 3
4EH	T3CTL	Timer 3 Control Register
4FH	T3L	Timer 3 Register Low
50H	ADCDATA1H	Analog Digital Convert Data Register 1 High
51H	ADCDATA1L	Analog Digital Convert Data Register 1 Low
52H	PP5L	Pulse-Width Modulation Periods 5 Register Low
54H	T2CCR0L	Timer 2 Compare Capture Register 0 Low
55H	PWM5L0	Pulse-Width Modulation 5 Duty Cycle Register Low 0
56H	PWM5H0	Pulse-Width Modulation 5 Duty Cycle Register High 0
57H	PWM5CTL0	Pulse-Width Modulation 5 Control Register 0
58H	ADCSICM	Analog Digital Convert Scan Interrupt control Register
59H	ADCDATA2H	Analog Digital Convert Data Register 2 High
5AH	ADCDATA2L	Analog Digital Convert Data Register 2 Low
5BH	PWM5CTL1	Pulse-Width Modulation 5 Control Register 1
5CH	P5ASCTL	Pulse-Width Modulation 5 Auto Shutdown Control Register
5DH	PSTRCTL	Pulse Auto Steer Control Register
5EH	ADCDATA3H	Analog Digital Convert Data Register 3 High
5FH	T3H	Timer 3 High Register
60H	PUR1	Pull-Up Resister 1
61H	PUR2	Pull-Up Resister 2
67H	INTEDGCTL	Interrupt Edge Control Register
6AH	ADCDATA3L	Analog Digital Convert Data Register 3 Low
6EH		
6FH		
BANK1		
地址	名称	全称
10EH	PWM5OC	Pulse-Width Modulation 5 Output Control Register
10FH	T2CTL1	Timer 2 Control Register 1
115H	C1CAL	Comparer 1 Calibration Register
117H	BANK	Bank
11CH	PWM5CTL2	Pulse-Width Modulation 5 Control Register 2

11DH	PWM5PC	Pulse-Width Modulation Polarity Control Register
120H	RSCTL	Receive Status Control Register
121H	TXSDR	Transmit Date Register
122H	RXSDR	Receive Date Reigster
123H	BRCTL	Baud Rate Control Register
124H	TSCTL	Transmit Control Register
125H	EUBRGL	Enhance Universal Baud Rate Generator Register Low
126H	EUBRGH	Enhance Universal Baud Rate Generator Register High
128H	SSCICTL0	Synchronous Serial Communication Interface Control Register 0
12AH	SSCICTL1	Synchronous Serial Communication Interface Control Register 0
12BH	SSCISTA	Synchronous Serial Communication Interface Status Register
12CH	SSCIBUFR	Synchronous Serial Communication Interface Buffer Register
12EH	SSCIMSK	Synchronous Serial Communication Interface Mask Register
12FH	WDTPS	Watchdog Pre-divider Selection Register
130H	MULAH	Multiplier A Register High
131H	MULAL	Multiplier A Register Low
132H	MULBH	Multiplier B Register High
133H	MULBL	Multiplier B Register Low
134H	MULCTL	Multiplier Control Register
135H	MULRES3	Multiplier Result Register 3
136H	MULRES2	Multiplier Result Register 2
137H	MULRES1	Multiplier Result Register 1
138H	MULRES0	Multiplier Result Register 0
139H	DIVCTL	Divider Control Register
13AH	DIVAH	Divider A Register High
13BH	DIVAL	Divider A Register Low
13CH	DIVB	Divider B Register
13DH	DIVQH	Divider Quotient Register High
13EH	DIVQL	Divider Quotient Register Low
13FH	DIVR	Divider Remainder Register
14AH	C1FILTCTL	Comparer 1 Filter Control Register
14BH	C1FILTPER	Comparer 1 Filter Per-scale Register
14CH	C2FILTCTL	Comparer 2 Filter Control Register
14DH	C2FILTPER	Comparer 2 Filter Per-scale Register
151H	CCRL	Catch Compare Register Low
152H	CCPCTL	Catch Compare PWM Control Register
154H	CCRH	Catch Compare Register High
156H	PINSET	PIN Set Register
159H	VRECAL2	Reference Voltage Calibration 2
15AH	VRECAL3	Reference Voltage Calibration 3
15DH	PWM5FC	Pulse-Width Modulation Force Control Register
165H	RC32KCAL	RC32K Calibration

附录 4 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	115.2K
20.000MHz	实际波特率	-	-	2404	9470	10417	19.5K	—	—
	误差(%)	-	-	0.16	-1.36	0.00	1.73	—	—
	EUBRGL 值(十六进制)	-	-	81H	20H	1DH	0FH	—	—
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)	CFH	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(%)	2.56	-	-	—	—	—	—	—
	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	115.2K
20.000MHz	实际波特率	-	-	-	9615	10417	19230	56818	113.6K
	误差(%)	-	-	-	0.16	0.00	0.16	-1.36	-1.36
	EUBRGL 值(十六进制)	-	-	-	81H	77H	40H	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	299	1185	2462	-	-	—	—	—
	误差(%)	-0.31	-1.23	2.56	-	-	—	—	—
	EUBRGL 值(十六进制)	6AH	1AH	0CH	-	-	—	—	—

SYNC=0,HBRG=0,BRG16=1									
波特率 / 时钟频率		300	1200	2400	9600	10417	19.2K	57.6K	115.2K
20.000MHz	实际波特率	300	1200	2399	9615	10417	19230	56818	113.6K
	误差(%)	-0.01	-0.03	-0.03	0.16	0.00	0.16	-1.36	-1.36
	EUBRGL 值 (十六进制)	1046H	411H	208H	81H	77H	40H	15H	0AH
16.000MHz	实际波特率	300	1200	2398	9615	10417	19231	58824	111.1K
	误差(%)	0.01	0.04	0.08	0.16	0.00	0.16	2.12	-3.55
	EUBRGL 值 (十六进制)	0D04H	340H	1A0H	67H	5FH	33H	10H	08H
8.000MHz	实际波特率	300	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	2FH	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	300	1202	2404	9615	10417	—	—	—
	误差(%)	-0.08	0.16	0.16	0.16	0.00	—	—	—
	EUBRGL 值 (十六进制)	01A0H	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	299	1185	2462	-	-	—	—	—
	误差(%)	-0.31	-1.23	2.56	-	-	—	—	—
	EUBRGL 值 (十六进制)	6AH	1AH	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	115.2K
20.000MHz	实际波特率	300	1200	2400	9597	10417	19230	57.47K	116279
	误差(%)	0.00	0.00	0.02	0.03	0.00	0.16	0.22	0.94
	EUBRGH：EUBRGL	411AH	1046H	0822H	0208H	01DFH	0103H	0056H	002AH
16.000MHz	实际波特率	300	1200	2399	9592	10417	19.23K	57971	114285
	误差(%)	0.00	0.01	-0.02	-0.08	0.00	0.16	0.64	0.79
	EUBRGH：EUBRGL	3414H	0D04H	0682H	01A0H	017FH	00CFH	0044H	0022H
8.000MHz	实际波特率	300	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	1A0AH	0682H	0340H	00CFH	00BFH	0067H	0022H	0010H
4.000MHz	实际波特率	300	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	300	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	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	—	—
512.000kHz	实际波特率	300	1196	2415	9846	10666	—	—	—
	误差(%)	-0.08	-0.31	0.63	2.56	2.40	—	—	—
	EUBRGH: EUBRGL	01AAH	006AH	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 5 BUGLIST

序号	说明	记录时间
1	KF8F3110SB 无上电延时功能； 即配置位 PWRT 置 1 时，也不会插入上电延时时间； 其他型号不受影响，无此限制。	2018-5-16

产品标识体系

XXXX	XXXX	-XX
产品系列	产品型号	封装形式

产品系列： KF8F = KF 系列 8 位单片机

产品型号：

3110 = 3110 型
3112 = 3112 型
3120 = 3120 型
3122 = 3122 型
3130 = 3130 型
3132 = 3132 型

封装形式：

SB = SOIC-8
SD = SOIC-14
TD = TSSOP-14
SE = SOIC-16
SG = SOIC-20
OG = SSOP-20

版本信息

版本号	更新记录	更新时间
V1.0	数据手册	
V1.1	修改文档格式与细节内容	2017-8-1
V1.2	增加多余 IO 端口的第二点说明 增加 AD 校准使能时关闭 ADC，需清零 ADCCALIEN 位说明 上拉功能说明	
V1.3	添加 QFN20 的管脚图和封装图	2017-12-5
V1.4	工作温度范围：-40℃～125℃	2018-3-30
V1.5	更新电气规范（温度范围），添加 8 脚型号 KF8F3110SB	2018-3-31
V1.6	更新静态电流特性	2018-4-9
V1.7	KF8F3110SB 无 DEE 资源 FLASH 自写和 DEE 的写操作可工作温度范围=-40~105℃	2018-4-28
V1.8	3110 添加 TSSOP-14 封装型号	2018-5-11
V1.9	添加附录 5 BUGLIST	2018-5-16
V2.0	3130 添加 TSSOP-14 封装型号	2018-7-13
V2.1	1.引脚示意图章节下备注 4：KF8F3110SB(SOIC-8)的外部复位引脚位于 P2.2 口，而非所有 KF8F3110 2.运放章节校验方法步骤 9：如果 AMPDT<2:0>加到 111 时还是相等，也结束运放失调校准； 3.RST 复位章节增加备注：KF8F3110SB 的 RST 引脚位于 P2.2。	2018-9-10
V2.2	KF8F3110SD 和 KF8F3110TD 开放 DEE	2018-11-28
V2.3	更新 PWM 章节	2019-05-07
V2.4	增加 FLASH 自写和 DEE 特性表格； 更新 SSOP20 封装尺寸信息。	2019-06-20
V2.5	更新 PWM 模块章节。	2019-07-09
V2.6	KF8F3110SB 增加 DEE 资源。	2020-04-30

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