

8 位微控制器

KF8L15Z20XX

数据手册



产品订购信息

芯片型号	订货号	封装	FLASH	RAM (Byte)	DEE (Byte)	内部 HFOSC	外部 HF/LFOSC	8 位 定时器	16 位 定时器	16 位 PWM	12 位 ADC 通道	12 位 DAC	内部 参考 (V)	运放	比较器	触摸	温度 传感器	SPI/I2C	UART	MUL/DIV	RTC
KF8L15Z20	KF8L15Z20QS	LQFP-48	10Kx16	2064	256	16M	20M/ 32.768k	1	4	12	9	1	2/3/4	1	3	18	Y	1/1	2	Y	Y
	KF8L15Z20QP	LQFP-32	10Kx16	2064	256	16M	20M/ 32.768k	1	4	7	8	1	2/3/4	1	3	16	Y	1/1	2	Y	Y
	KF8L15Z20NM	QFN-24	10Kx16	2064	256	32M	20M/ 32.768k	1	4	4	7	1	2/3/4	1	3	N	Y	1/1	2	Y	Y
	KF8L15Z20OG	SSOP-20	10Kx16	2064	256	16M	20M/ 32.768k	1	4	4	6	1	2/3/4	1	3	N	Y	1/1	2	Y	Y

版权所有@
上海芯旺微电子有限公司

本文档为上海芯旺微电子有限公司在现有数据资料基础上慎重且力求准确无误编制而成.确保应用符合技术规范，是您自身应负的责任。上海芯旺微电子有限公司不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。上海芯旺微电子有限公司对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将芯旺微电子有限公司的芯片用于生命维持和或生命安全应用，一切风险由使用方自负。使用方同意在由此引发任何一切伤害、索赔、诉讼或费用时，会维护和保障上海芯旺微电子有限公司免于承担法律责任，并加以赔偿。

本文档中所述的器件应用信息及其他类似内容仅为您提供便利，可能有更新的信息所替代。上海芯旺微电子有限公司会不定期进行更新，恕不另行通知。使用方如需获得最新的产品信息，请及时访问上海芯旺微电子有限公司官网或与上海芯旺微电子有限公司联系。

KF8L15Z20 芯片使用注意事项

芯片的 ESD 防护措施

KF8L15Z20 芯片提供满足工业级 ESD 标准保护电路。建议用户根据芯片存储/应用的环境采取适当静电防护措施。应注意应用环境的湿度；建议避免使用容易产生静电的绝缘体；存放和运输应在抗静电容器、防静电屏蔽袋或导电材料容器中；包括工作台在内的所有测试和测量工具必须保证接地；操作者应该佩戴静电消除手腕环手套，不能用手直接接触芯片等。

芯片的 EFT 防护措施

KF8L15Z20 芯片提供满足工业级 EFT 标准的保护电路。当 MCU 芯片应用在 PCB 系统时，需要遵守 PCB 相关设计要求，包括电源线、地线（包括数字/模拟电源分离，单点/多点接地等）、复位管脚保护电路、电源和地之间的去耦电容、高低频电路单独分别处理以及单/多层板选择等。

芯片的 LATCH-UP 防护措施

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

芯片的焊接

KF8L15Z20 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

KF8L15Z20 芯片提供独立电源管脚。当 KF8L15Z20 芯片应用在多电源供电系统时，应先对 MCU 芯片上电，再对系统其他部件上电；反之，断电时，先对系统其他部件断电，再对 MCU 芯片断电。若操作顺序相反则可能导致芯片内部元件过压或过流，从而导致芯片故障或元件退化。

芯片的复位

KF8L15Z20 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

KF8L15Z20 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

KF8L15Z20 芯片提供各种内部和外部复位。对于不同的应用系统，有必要对芯片寄存器、内存、功能模块等进行初始化，尤其是 I/O 管脚复用功能进行初始化，避免由于芯片上电以后，I/O 管脚状态的不确定情况发生。

芯片的管脚

KF8L15Z20 芯片提供宽范围的输入管脚电平，用户输入高电平应大于 V_{IH} 的最小值，低电平应小于 V_{IL} 的最大值，以免波动噪声进入芯片。对于未使用的输入/输出管脚，建议用户设为输入状态，并通过电阻上拉至电源或下拉至地，或设置为输出管脚，输出固定电平并浮空。对未使用的管脚处理因应用系统而异，具体遵循应用系统的相关规定和说明。

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

芯片的低功耗设计

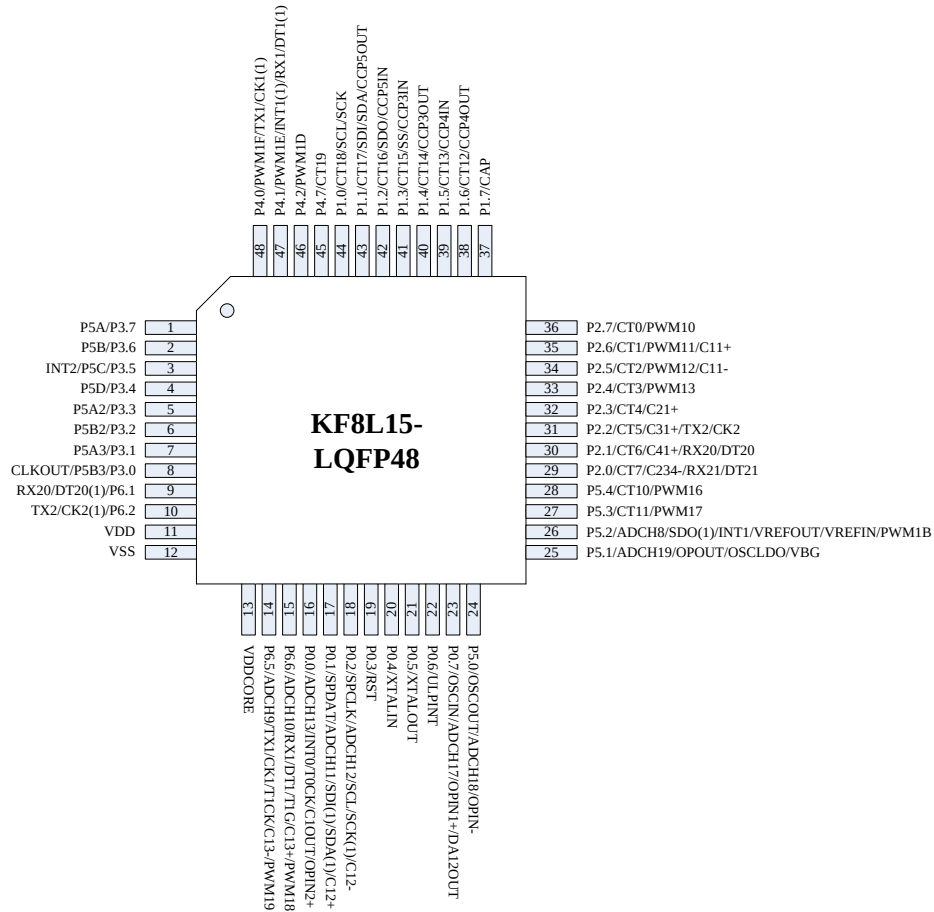
KF8L15Z20 芯片提供低功耗设计模式，用户在实际应用中可根据应用系统的要求采用各种不同的低功耗模式，包括系统工作时钟的选择和休眠模式的选择等等。

芯片的开发环境

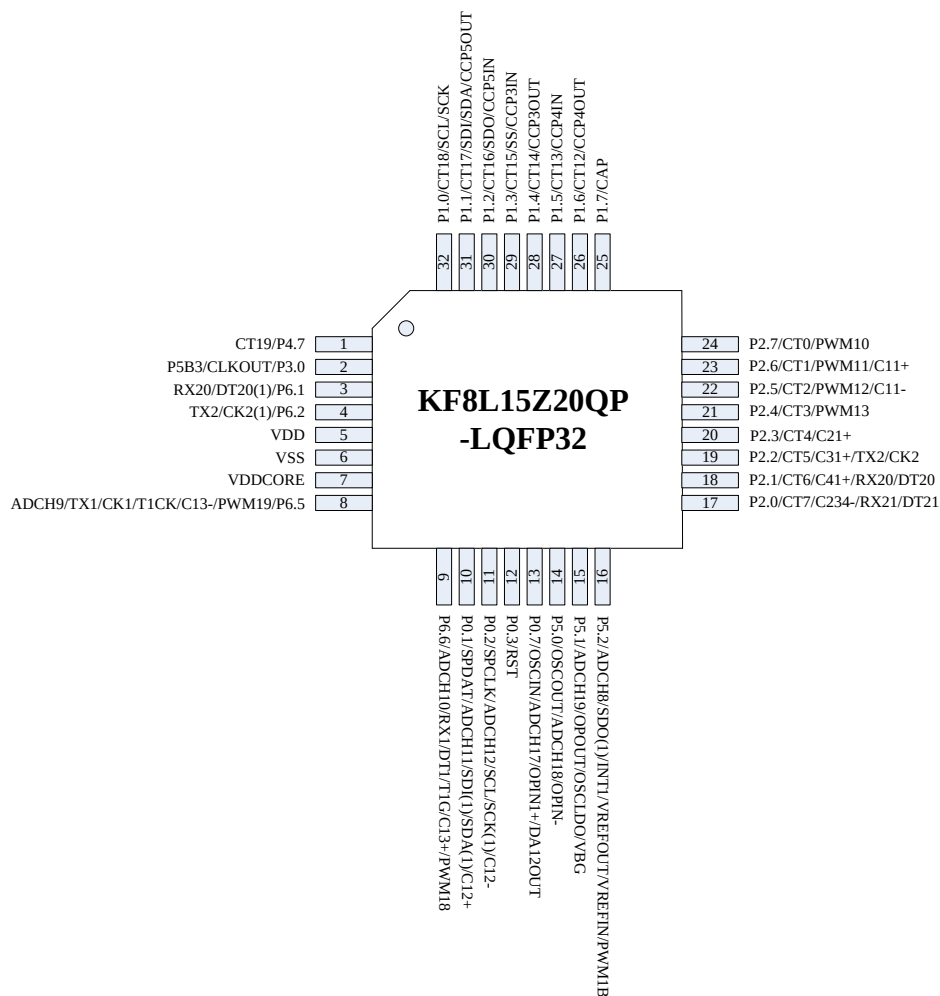
KF8L15Z20 芯片提供完整的软/硬件开发环境，并受知识产权保护。选择上海芯旺微电子有限公司指定的汇编器、编译器、编程器、硬件仿真器开发环境，必须遵循与芯片相关的规定和说明。

引脚示意图

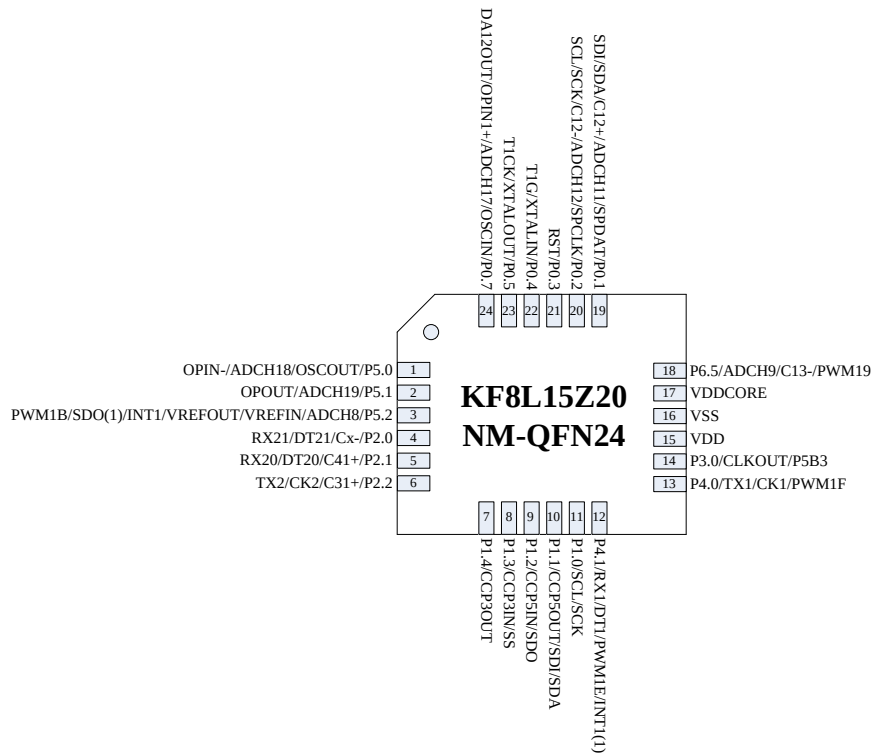
KF8L15Z20QS-LQFP48 引脚示意图:



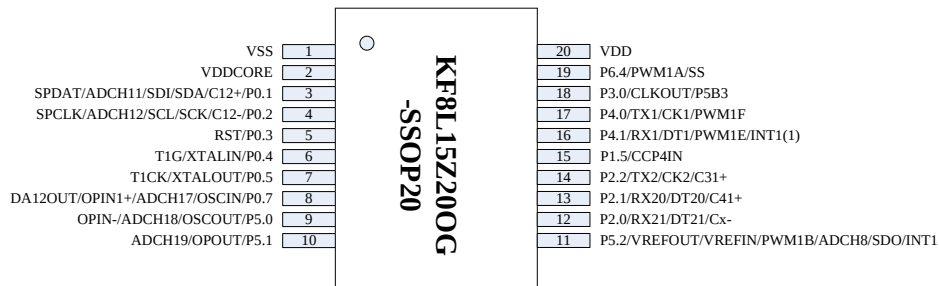
KF8L15Z20QP-LQFP32 引脚示意图:



KF8L15Z20NM-QFN24 引脚示意图:



KF8L15Z20OG-SSOP20 引脚示意图:



引脚功能说明

引脚脚位				I/O	引脚名	引脚说明
LQFP48	LQFP32	QFN24	SSOP20			
1	-	-	-	I/O	P3.7	带上拉和电平变化中断功能的双向输入输出端口
					P5A	PWM5 输出通道
2	-	-	-	I/O	P3.6	带上拉和电平变化中断功能的双向输入输出端口
					P5B	PWM5 输出通道
3	-	-	-	I/O	P3.5	带上拉和电平变化中断功能的双向输入输出端口
					P5C	PWM5 输出通道
					INT2	外部中断 2 输入引脚
4	-	-	-	I/O	P3.4	带上拉和电平变化中断功能的双向输入输出端口
					P5D	PWM5 输出通道
5	-	-	-	I/O	P3.3	带上拉和电平变化中断功能的双向输入输出端口
					P5A2	PWM5 输出通道

6	-	-	-	I/O	P3.2	带上拉和电平变化中断功能的双向输入输出端口
					P5B2	PWM5 输出通道
7	-	-	-	I/O	P3.1	带上拉和电平变化中断功能的双向输入输出端口
					P5A3	PWM5 输出通道
8	2	14	18	I/O	P3.0	带上拉和电平变化中断功能的双向输入输出端口
					CLKOUT	指令时钟输出
					P5B3	PWM5 输出通道
9	3		-	I/O	P6.1	带上拉功能的双向输入输出端口
					DT20(1)	串口 2 同步数据引脚 0
					RX20(1)	串口 2 异步接收引脚 0
10	4		-	I/O	P6.2	带上拉功能的双向输入输出端口
					CK2(1)	串口 2 同步时钟引脚
					TX2(1)	串口 2 异步发送引脚
-	-		19	I/O	P6.4	带上拉功能的双向输入输出端口
					PWM1A	PWM1 输出引脚
					SS(1)	SPI 从动选择引脚
11	5	15	20	P	VDD	电源
12	6	16	1	P	VSS	地线, 0V 参考点
13	7	17	2	P	VDDcore	内部电压外接电容脚, 必须外接 4.7uf (推荐) 电容
14	8	18	-	I/O	P6.5	带上拉功能的双向输入输出端口
					ADCH9	AD 外部通道 9
					TX1	串口 1 异步发送引脚
					CK1	串口 1 同步时钟引脚
					C13-	比较器 1 负端输入 3
					PWM19	PWM1 输出端口
15	9	-	-	I/O	P6.6	带上拉功能的双向输入输出端口

					ADCH10	AD 外部通道 10
					RX1	串口 1 同步数据引脚
					DT1	串口 1 异步接收引脚
					C13+	比较器 1 正端输入 3
					PWM18	PWM1 输出端口
16	-	-	-	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
					ADCH13	AD 外部通道 13
					INT0	外部中断输入 0
					T0CK	T0 外部时钟输入引脚
					C1OUT	比较器 1 输出
					OPIN2+	运算放大器正端输入 2
17	10	19	3	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
					SPDAT	编程数据引脚
					ADCH11	AD 外部通道 11
					SDI(1)	SPI 数据输入脚
					SDA(1)	I2C 数据输入/输出脚
					C12+	比较器 1 正端输入 2
18	11	20	4	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
					SPCLK	编程时钟引脚
					ADCH12	AD 外部通道 12
					SCL(1)	I2C 时钟脚
					SCK(1)	SPI 时钟脚
					C12-	比较器 1 负端输入 2
19	12	21	5	I/O	P0.3	带上拉和电平变化中断功能的双向输入输出端口
					RST	外部复位引脚
20	-	22	6	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
					XTALIN	外接低频晶振输入引脚
21	-	23	7	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
					XTALOUT	外接低频晶振输出引脚
22	-	-	-	I/O	P0.6	带上拉和电平变化中断功能的双向输入输出端口
					ULPINT	超低功耗外部唤醒引脚
23	13	24	8	I/O	P0.7	带上拉和电平变化中断功能的双向输入输出端口
					OSCIN	外部高频晶振输入引脚
					ADCH17	AD 外部通道 17
					OPIN1+	运算放大器正端输入 1
					DA12OUT	12 位数/模转换结果输出引脚

24	14	1	9	I/O	P5.0	带上拉功能的双向输入输出端口
					OSCOOUT	外部高频晶振输出引脚
					ADCH18	AD 外部通道 18
					OPIN-	运算放大器负端输入
25	15	2	10	I/O	P5.1	带上拉功能的双向输入输出端口
					ADCH19	AD 外部通道 19
					OPOUT	运算放大器输出端
26	16	3	11	I/O	P5.2	带上拉功能的双向输入输出端口
					ADCH8	AD 外部通道 8
					SDO(1)	SPI 数据输出引脚
					INT1	外部中断 1 引脚
					VREFOUT	内部参考电压输出
					VREFIN	AD 外部参考电压输入端
27	-	-	-	I/O	PWM1B	PWM1 输出端口
					P5.3	带上拉功能的双向输入输出端口
					PWM17	PWM1 输出端口
28	-	-	-	I/O	CT0	触摸通道 0
					P5.4	带上拉功能的双向输入输出端口
					PWM16	PWM1 输出端口
29	17	4	12	I/O	CT1	触摸通道 1
					P2.0	带上拉功能的双向输入输出端口
					CT4	触摸通道 4
					C234-	比较器 2/3/4 负端输入
					RX21	串口 2 异步接收引脚 1
30	18	5	13	I/O	DT21	串口 2 同步数据引脚 1
					P2.1	带上拉功能的双向输入输出端口
					CT5	触摸通道 5
					C41+	比较器 4 正端输入 1
					RX20	串口 2 异步接收引脚 0
31	19	6	14	I/O	DT20	串口 2 同步数据引脚 0
					P2.2	带上拉功能的双向输入输出端口
					LCDS8	LCD 显示 SEG 端 8
					CT6	触摸通道 6
					C31+	比较器 3 正端输入 1
					TX2	串口 2 异步发送引脚
					CK2	串口 2 同步时钟引脚

32	20	-	-	I/O	P2.3	带上拉功能的双向输入输出端口
					CT7	触摸通道 7
					C21+	比较器 2 正端输入 1
33	21	-	-	I/O	P2.4	带上拉功能的双向输入输出端口
					CT8	触摸通道 8
					PWM13	PWM1 输出通道 3
34	22	-	-	I/O	P2.5	带上拉功能的双向输入输出端口
					CT9	触摸通道 9
					PWM12	PWM1 输出通道 2
					C11-	比较器 1 负端输入 1
35	23	-	-	I/O	P2.6	带上拉功能的双向输入输出端口
					CT10	触摸通道 10
					PWM11	PWM1 输出通道 1
					C11+	比较器 1 正端输入 1
36	24	-	-	I/O	P2.7	带上拉功能的双向输入输出端口
					CT11	触摸通道 11
					PWM10	PWM1 输出通道 0
37	25	-	-	I/O	P1.7	带上拉功能的双向输入输出端口
					CAP	触摸电容脚
38	26	-	-	I/O	P1.6	带上拉功能的双向输入输出端口
					CT12	触摸通道 12
					CCP4OUT	CCP4 比较输出端口
39	27	15	15	I/O	P1.5	带上拉功能的双向输入输出端口
					CT13	触摸通道 13
					CCP4IN	CCP4 输入引脚
40	28	7	-	I/O	P1.4	带上拉功能的双向输入输出端口
					CT14	触摸通道 14
					CCP3OUT	CCP3 输出引脚
41	29	8	-	I/O	P1.3	带上拉功能的双向输入输出端口
					CT15	触摸通道 15
					SS	SPI 模式从动选择输入口
					CCP3IN	CCP3 输入引脚
42	30	9	-	I/O	P1.2	带上拉功能的双向输入输出端口
					CT16	触摸通道 16

					SDO	SPI 数据输出脚
					CCP5IN	CCP5 输入引脚
43	31	10	-	I/O	P1.1	带上拉功能的双向输入输出端口
					CT17	触摸通道 17
					SDI	SPI 数据输入脚
					SDA	I2C 数据输入/输出脚
					CCP5OUT	CCP5 输出引脚
44	32	11	-	I/O	P1.0	带上拉功能的双向输入输出端口
					CT18	触摸通道 18
					SCL	I2C 时钟脚
					SCK	SPI 时钟脚
45	1	-	-	I/O	P4.7	带上拉功能的双向输入输出端口
					CT19	触摸通道 19
46	-	-	-	I/O	P4.2	带上拉功能的双向输入输出端口
					PWM1D	PWM1 输出通道
47	-	12	16	I/O	P4.1	带上拉功能的双向输入输出端口
					PWM1E	PWM1 输出通道
					INT1(1)	外部中断 1 输入引脚
					RX1(1)	串口 1 异步接收引脚
					DT1(1)	串口 1 同步数据引脚
48	-	13	17	I/O	P4.0	带上拉功能的双向输入输出端口
					PWM1F	PWM1 输出通道
					TX1(1)	串口 1 异步发送引脚
					CK1(1)	串口 1 同步时钟引脚

目 录

产品订购信息	2
KF8L15Z20 芯片使用注意事项	3
引脚示意图	5
引脚功能说明	8
目 录	14
1 系统概述	27
1.1 芯片特征	28
1.2 系统框图	30
1.3 存储器	31
1.4 系统时钟	32
1.4.1 时钟模块相关寄存器.....	33
1.4.1.1 系统频率控制寄存器 OSCCTL	33
1.4.1.2 系统时钟标志寄存器 OSCSTA	34
1.4.2 上电延时.....	34
1.4.3 内部高频振荡器.....	34
1.4.3.1 校准寄存器.....	34
1.4.4 内部低频振荡器.....	35
1.4.4.1 校准寄存器.....	35
1.4.5 外部高频振荡器.....	36
1.4.6 外部低频振荡器.....	37
1.4.7 时钟切换和时钟信号同步.....	37
1.4.8 双速启动模式.....	38
1.4.9 双速启动过程.....	38
1.4.10 外部时钟故障检测.....	38
1.4.11 时钟故障保护处理.....	39
1.5 配置位	40
1.6 在线串行编程	42
2 I/O 端口.....	43
2.1 I/O 端口的读写	43
2.2 P0 口.....	44
2.2.1 P0 口相关的寄存器	44
2.2.1.1 P0 口状态寄存器 (P0)	44
2.2.1.2 P0 口输出锁存寄存器 (POLR)	44
2.2.1.3 P0 口方向控制寄存器 (TR0)	45
2.2.1.4 P0 上拉功能控制寄存器 (PUR0)	45
2.2.1.5 P0 口电平变化中断控制寄存器 (IOCL0)	45

2.2.1.6	P0 口模拟/数字口设置寄存器 (ANS0)	46
2.2.2	P0 口各引脚内部原理功能框图	47
2.3	P1 口	48
2.3.1	P1 口相关的寄存器	48
2.3.1.1	P1 口状态寄存器 (P1)	48
2.3.1.2	P1 口输出锁存寄存器 (P1LR)	48
2.3.1.3	P1 口方向控制寄存器 (TR1)	49
2.3.1.4	P1 口上拉功能控制寄存器 (PUR1)	49
2.3.1.5	P1 口模拟/数字口设置寄存器 (ANS1)	49
2.3.2	P1 口原理功能框图	50
2.4	P2 口	51
2.4.1	P2 口相关的寄存器	51
2.4.1.1	P2 口状态寄存器 (P2)	51
2.4.1.2	P2 口输出锁存寄存器 (P2LR)	51
2.4.1.3	P2 口方向控制寄存器 (TR2)	52
2.4.1.4	P2 口上拉功能控制寄存器 (PUR2)	52
2.4.1.5	P2 口模拟/数字口设置寄存器 (ANS2)	52
2.4.2	P2 口原理功能框图	53
2.5	P3 口	54
2.5.1	P3 口相关的寄存器	54
2.5.1.1	P3 口状态寄存器 (P3)	54
2.5.1.2	P3 口输出锁存寄存器 (P3LR)	54
2.5.1.3	P3 口方向控制寄存器 (TR3)	55
2.5.1.4	P3 口上拉功能控制寄存器 (PUR3)	55
2.5.1.5	P3 口电平变化中断控制寄存器 (IOCL3)	55
2.5.1.6	P3 口模拟/数字口设置寄存器 (ANS3)	56
2.5.2	P3 口原理功能框图	57
2.6	P4 口	58
2.6.1	P4 口相关的寄存器	58
2.6.1.1	P4 口状态寄存器 (P4)	58
2.6.1.2	P4 口输出锁存寄存器 (P4LR)	58
2.6.1.3	P4 口方向控制寄存器 (TR4)	59
2.6.1.4	P4 口上拉功能控制寄存器 (PUR4)	59
2.6.1.5	P4 口模拟/数字口设置寄存器 (ANS4)	59
2.6.2	P4 口原理功能框图	60
2.7	P5 口	61
2.7.1	P5 口相关的寄存器	61
2.7.1.1	P5 口状态寄存器 (P5)	61
2.7.1.2	P5 口输出锁存寄存器 (P5LR)	61
2.7.1.3	P5 口方向控制寄存器 (TR5)	62
2.7.1.4	P5 口上拉功能控制寄存器 (PUR5)	62
2.7.1.5	P5 口模拟/数字口设置寄存器 (ANS5)	62
2.7.2	P5 口原理功能框图	63
2.8	P6 口	64

2.8.1	P6 口相关的寄存器	64
2.8.1.1	P6 口状态寄存器(P6)	64
2.8.1.2	P6 口输出锁存寄存器(P6LR)	64
2.8.1.3	P6 口方向控制寄存器(TR6)	65
2.8.1.4	P6 口上拉功能控制寄存器(PUR6)	65
2.8.1.5	P6 口模拟/数字口设置寄存器(ANS6)	65
2.8.2	P6 口原理功能框图	66
3	存储器	67
3.1	程序存储器(ROM)区	67
3.1.1	程序计数器(PC)	67
3.1.1.1	PCL/PCH 寄存器的使用	68
3.1.1.2	执行 JMP、CALL 指令时的情况	68
3.1.2	堆栈	69
3.2	数据存储器(RAM)区	69
3.2.1	通用寄存器区	70
3.2.2	特殊功能寄存器(SFR) 区	70
3.2.3	状态字寄存器(PSW)	70
3.3	FLASH 自写	72
3.3.1	寄存器 NVMDATAH/L	73
3.3.2	寄存器 NVMADDRH/L	73
3.3.3	写 Flash	73
3.3.4	读 Flash	75
3.4	DATA EEPROM	76
3.4.1	寄存器 NVMDATAL	76
3.4.2	寄存器 NVMADDRL	76
3.4.1	寄存器 NVMCTL0/NVMCTL1	76
3.4.2	写 DATA EEPROM	76
3.4.3	读 DATA EEPROM	77
3.5	寄存器组 RN	78
3.6	ID 地址单元	78
4	汇编指令及寻址方式	79
4.1	寻址方式	79
4.1.1	寄存器寻址	79
4.1.2	直接寻址	79
4.1.3	立即数寻址	79
4.1.4	寄存器间接寻址	80
4.1.5	位寻址	80
4.2	汇编指令	80
5	中断	81
5.1	中断相关的寄存器	84
5.1.1	中断控制寄存器 INTCTL	84

5.1.2	中断使能寄存器 EIE1	86
5.1.3	中断使能寄存器 EIE2	86
5.1.4	中断使能寄存器 EIE3	87
5.1.5	中断使能寄存器 EIE4	87
5.1.6	中断使能寄存器 EIE5	88
5.1.7	中断标志寄存器 EIF1	89
5.1.8	中断标志寄存器 EIF2	89
5.1.9	中断标志寄存器 EIF3	90
5.1.10	中断标志寄存器 EIF4	90
5.1.11	中断标志寄存器 EIF5	91
5.1.12	中断优先级控制寄存器 IP0	92
5.1.13	中断优先级控制寄存器 IP1	92
5.1.14	中断优先级控制寄存器 IP2	93
5.1.15	中断优先级控制寄存器 IP3	93
5.1.16	中断优先级控制寄存器 IP4	94
5.1.17	中断优先级控制寄存器 IP5	94
5.1.18	电源控制寄存器 PCTL	95
5.1.19	INT 中断沿选择寄存器 INTEDGCTL.....	96
5.1.20	中断响应.....	96
5.2	INT 中断.....	97
5.2.1	INT1 中断	97
5.3	定时器中断	97
5.4	P0/P3 口中断	97
5.5	PWM 中断	97
5.6	模拟比较器中断	98
5.7	USART 中断	98
5.8	中断现场保护	98
6	定时/计数器.....	99
6.1	定时器 T0.....	99
6.1.1	T0 原理框图	99
6.1.2	T0 相关的寄存器	99
6.1.2.1	OPTR 选择寄存器	100
6.1.3	T0 的使用	100
6.2	定时/计数器 T1	101
6.2.1	T1 原理框图	101
6.2.2	T1 相关的寄存器	101
6.2.2.1	T1 控制寄存器	102
6.2.2.2	INT 中断沿选择寄存器	103
6.2.3	定时模式.....	103
6.2.4	计数模式.....	103
6.2.5	T1 重载功能	103
6.2.6	T1 在休眠模式下的运行	103
6.2.7	T1 分配给 PWM1.....	104

6.2.8 T1 计数寄存器被比较器清零	104
6.3 定时器 T2	105
6.3.1 T2 的工作原理	105
6.3.2 T2 相关的寄存器	106
6.3.2.1 T2 控制寄存器 T2CTL0	106
6.3.2.2 T2 控制寄存器 T2CTL1	107
6.3.2.3 T2CCR1/0H 和 T2CCR1/0L 寄存器	107
6.3.3 T2 中断	107
6.3.4 T2 在休眠模式	107
6.4 定时/计数器 T3	108
6.4.1 T3 原理框图	108
6.4.2 T3 相关寄存器	108
6.4.2.1 T3 控制寄存器	109
6.4.2.2 CCP3 功能寄存器	110
6.4.3 T3 预分频器	110
6.4.4 T3 计数时钟选择	110
6.4.5 T3 重载功能	110
6.4.6 T3 分配给电容触摸	111
6.4.7 CCP3	111
6.4.7.1 捕捉模式	111
6.4.7.2 比较模式	112
6.4.8 T3 中断	113
6.4.9 T3 工作在休眠模式	113
6.4.10 T3 计数寄存器被比较器清零	113
6.5 定时/计数器 T4	114
6.5.1 T4 原理框图	114
6.5.1.1 T4 相关寄存器	114
6.5.1.2 T4 控制寄存器	115
6.5.1.3 CCP4 功能寄存器	116
6.5.2 T4 预分频器	116
6.5.3 T4 计数时钟选择	116
6.5.4 T4 重载功能	116
6.5.5 CCP4	117
6.5.5.1 捕捉模式	117
6.5.5.2 比较模式	117
6.5.6 T4 中断	119
6.5.7 T4 工作在休眠模式	119
6.5.8 T4 计数寄存器被比较器清零	119
7 ADC12 模数转换模块	120
7.1 与 ADC12 相关的寄存器	120
7.1.1 ADC12 控制寄存器 0 (ADCCTL0)	121
7.1.2 ADC12 控制寄存器 1 (ADCCTL1)	121
7.1.3 ADC12 控制寄存器 2 (ADCCTL2)	122

7.1.4	ADC12 中断控制寄存器 (ADCINTCTL)	122
7.1.5	温度传感器配置寄存器 (TEMPSNR)	123
7.2	通道的选择	123
7.3	ADC12 转换参考电压的选择	123
7.4	转换时钟的选择	123
7.5	输出格式	124
7.6	AD 转换的启动和完成	124
7.7	ADC12 工作在休眠模式	124
7.8	复位的影响	124
7.9	使用 ADC12 转换器的设置	125
8	DAC12 数模转换器模块	126
8.1	DAC12 原理框图	126
8.2	DAC12 相关寄存器	126
8.2.1	DAC12 控制寄存器 (DAC12CTL)	127
8.2.2	DAC12 数据寄存器 (DAC12DH/DAC12DL)	127
8.3	DAC12 基准选择	128
8.4	DAC12 失调消除功能	128
8.5	使用 DAC12 的设置	128
9	PWM 模块	129
9.1	工作原理	129
9.2	PWM1 输出端口	130
9.3	PWM1x 相关的寄存器	130
9.3.1	PWM1x 控制寄存器	130
9.3.2	PWM1x 周期	131
9.3.3	PWM1x 占空比	131
9.4	PWM1x 分辨率	131
9.5	PWM1x 中断	132
9.6	休眠模式下的操作	132
9.7	系统时钟频率的改变	132
9.8	复位的影响	132
9.9	PWM1x 使用方法	132
10	CCP(捕捉/比较/PWM5) 模块	133
10.1	CCP 相关寄存器	133
10.2	捕捉模式	134
10.3	比较模式	135
10.4	PWM5 模式	136
10.4.1	PWM5 相关控制寄存器	139
10.4.1.1	PWM5CTL0 寄存器	139
10.4.1.2	PWM5CTL1 寄存器	140
10.4.1.3	PWM5CTL2 寄存器	140
10.4.1.4	PWM5 极性控制寄存器 (PWM5PC)	141

10.4.1.5	PWM5 强制控制寄存器 (PWM5FC)	141
10.4.1.6	PWM5 输出控制寄存器 (PWM5OC)	142
10.4.1.7	PWM5 自动关闭控制寄存器 (P5ASCTL0)	143
10.4.1.8	P5ASCTL1 寄存器	143
10.4.1.9	PSTRCTL0 寄存器	144
10.4.1.10	PSTRCTL1 寄存器	144
10.4.2	PWM5 的周期、占空比及分辨率	146
10.4.2.1	PWM5 周期	146
10.4.2.2	PWM5 占空比	146
10.4.2.3	PWM5 分辨率	147
10.4.2.4	PWM5 中断	147
10.4.3	边沿对齐 PWM 信号	148
10.4.4	中心对齐 PWM 信号	148
10.4.5	PWM 信号产生和中断	149
10.4.6	单输出模式	149
10.4.7	半桥输出模式	151
10.4.7.1	死区延时	152
10.4.8	全桥输出模式	153
10.4.9	脉冲转向控制	156
10.4.10	输出控制	156
10.4.11	强制输出模式	156
10.4.12	PWM 输出极性模式	156
10.4.13	PWM 更新锁定	157
10.4.14	PWM 复位控制	157
10.4.15	自动关断和自动重启模式	158
10.4.15.1	自动关断模式	158
10.4.15.2	自动重启模式	159
11	运算放大器模块	160
11.1	与运放有关的寄存器	160
11.1.1	运放控制寄存器	160
11.1.2	运放校准寄存器	161
11.2	运放使用方法	161
11.2.1	校准方法	161
11.2.2	使用方法	161
12	模拟比较器模块	162
12.1	模拟比较器原理	162
12.2	比较器相关寄存器	164
12.2.1	比较器中断触发控制寄存器 CMCTL0	164
12.2.2	比较器 1 控制寄存器 C1CTL	165
12.2.3	比较器 3 控制寄存器 C3CTL	165
12.2.4	比较器 4 控制寄存器 C4CTL	166
12.2.5	模拟比较器输出寄存器 COUT	166

12.3	滤波器相关寄存器	167
12.3.1	比较器 x 滤波器控制寄存器 CxFILTCTL	167
12.3.2	比较器 x 滤波器采样时钟分频寄存器 CxFILTPRE	168
12.4	极性选择	168
12.5	范围控制功能	168
12.6	滤波功能	169
12.7	比较器使用	169
12.8	比较器清零定时器	169
12.8.1	比较器清零定时器控制寄存器 CCTCTL	169
12.8.2	使用方法.....	170
13	SSCI 模块	171
13.1	概述	171
13.2	SSCI 相关寄存器.....	171
13.2.1	SSCI 引脚配置寄存器 (PINSET)	171
13.2.2	SSCI 控制寄存器 0 (SSCICTL0)	172
13.2.3	SSCI 控制寄存器 1 (SSCICTL1)	173
13.2.4	SSCI 状态寄存器 (SSCISTA)	174
13.2.5	SSCI 屏蔽寄存器 (SSCIMSK)	175
13.3	I2C 模式.....	176
13.3.1	工作原理.....	176
13.3.2	I2C 从动模式	177
13.3.2.1	寻址.....	177
13.3.2.2	接收.....	178
13.3.2.3	发送.....	179
13.3.2.4	广播呼叫地址支持.....	181
13.3.3	I2C 主动模式	182
13.3.3.1	主控模式支持.....	182
13.3.3.2	I2C 主模式操作	183
13.3.3.3	波特率发生器.....	184
13.3.3.4	I2C 主控模式启动条件时序	184
13.3.3.5	I2C 主控模式重复启动条件时序	185
13.3.3.6	I2C 主控模式发送	186
13.3.3.7	I2C 主控模式接收	188
13.3.3.8	应答序列时序.....	189
13.3.3.9	停止条件序列.....	190
13.3.3.10	时钟仲裁.....	191
13.3.4	多主控器模式.....	192
13.3.4.1	多主机通信, 总线冲突与总线仲裁.....	192
13.3.4.2	启动条件期间的总线冲突.....	193
13.3.4.3	重复启动条件期间的总线冲突.....	195
13.3.4.4	停止条件期间的总线冲突.....	197
13.3.4.5	SSCI 屏蔽寄存器	198
13.4	SPI 模式	199

13.4.1	工作原理.....	200
13.4.2	使能 SPI/IO 与外部链接.....	201
13.4.3	典型连接.....	201
13.4.4	主模式.....	202
13.4.5	从模式.....	203
13.4.5.1	从动模式.....	203
13.4.5.2	从动选择同步.....	203
13.4.6	休眠模式和复位.....	205
13.4.7	SPI 四种工作模式设置	205
13.4.7.1	主控发送工作流程.....	205
13.4.7.2	从动接收工作流程.....	205
13.4.7.3	主控接收工作流程.....	206
13.4.7.4	从动发送工作流程.....	206
14	通用全双工/ 半双工收发器 (USART)	207
14.1	系统概述	207
14.1.1	相关引脚.....	207
14.1.2	相关寄存器.....	208
14.1.3	原理框图.....	209
14.2	波特率发生器	210
14.2.1	USARTx 波特率控制寄存器 BRCTLx.....	211
14.2.2	波特率的选择.....	212
14.2.3	自动波特率检测.....	213
14.2.4	接收间隔字符时自动唤醒.....	215
14.2.5	间隔符时序.....	216
14.2.5.1	间隔和同步发送序列.....	216
14.2.5.2	接收间隔字符.....	216
14.3	USART 全双工模式.....	218
14.3.1	USART 全双工发送操作	219
14.3.2	发送和控制状态寄存器 TSCTLx	219
14.3.3	发送数据.....	220
14.3.4	全双工发送的设置.....	220
14.3.5	USART 全双工接收操作	222
14.3.6	接收状态和控制寄存器 RSCTLx	223
14.3.7	接收数据.....	224
14.3.8	接收错误.....	224
14.3.9	地址检测.....	225
14.3.10	全双工接收的设置.....	225
14.3.11	RS-485 发送/接收	226
14.3.11.1	RS-485 9 位地址检测模式设置	227
14.3.12	全双工操作时钟的精确性.....	227
14.4	USART 半双工模式.....	227
14.4.1	USARTx 半双工主控模式	227
14.4.2	半双工主控发送.....	228

14.4.3	半双工主控接收.....	230
14.4.4	USART 半双工从动模式.....	233
14.4.5	USART 半双工从动发送.....	233
14.4.6	USART 半双工从动接收.....	234
14.4.7	USART 半双工 RS-485 模式.....	234
14.5	USART 工作在休眠模式下.....	234
14.5.1	USART 休眠使能寄存器 USLPEN.....	234
14.6	USART 单线通信模式.....	235
14.6.1	USART 引脚配置寄存器 UPINSETx.....	235
14.7	7816 模式.....	236
14.7.1	7816 模式发送.....	236
14.7.1.1	发送设置.....	236
14.7.2	7816 模式接收.....	237
14.7.3	7816 模式寄存器.....	238
14.7.3.1	7816 发送控制寄存器 U7816TXCTL2.....	238
14.7.3.2	7816 接收控制寄存器 U7816RXCTL2.....	239
14.7.3.3	7816 控制寄存器 U7816CTL2.....	240
14.7.3.4	预分频控制寄存器 CLKDIV2.....	240
14.7.3.5	EGT 控制寄存器 EGTCTL2.....	240
15	参考电压模块 FVR.....	241
15.1	参考电压相关寄存器.....	241
15.1.1	参考电压寄存器 (VREFCTL).....	242
16	高/低压监测 HLVD.....	243
16.1	相关寄存器.....	243
16.1.1	高低压监测控制寄存器 HLVDCTL.....	243
16.2	工作原理.....	243
16.3	HLVD 的使用.....	244
16.4	HLVD 工作在休眠模式.....	244
17	硬件乘法器模块.....	245
17.1	概述.....	245
17.2	硬件乘法器相关寄存器.....	246
17.2.1	乘法运算控制寄存器 (MULCTL).....	246
17.3	硬件乘法器的使用.....	246
18	硬件除法器模块.....	248
18.1	概述.....	248
18.2	硬件除法器相关寄存器.....	249
18.2.1	除法控制寄存器 (DIVCTL).....	249
18.2.2	数据寄存器.....	249
18.2.2.1	被除数寄存器 {DIVAH:DIVAL}.....	250
18.2.2.2	除数寄存器 DIVB.....	250
18.2.2.3	结果商寄存器 {DIVQH:DIVQL}.....	250

18.2.2.4	运算余数寄存器 DIVR	251
18.3	硬件除法器操作	251
19	实时时钟 (RTC) 模块	252
19.1	概述	252
19.1.1	原理框图	252
19.2	寄存器描述	253
19.2.1	相关寄存器	253
19.2.2	实时时钟启动寄存器 RTCSRT	253
19.2.3	状态显示寄存器 RTCSTU	254
19.2.4	闹钟中断时间寄存器	255
19.2.4.1	闹钟中断秒寄存器 RTCALRS	255
19.2.4.2	闹钟中断分寄存器 RTCALRM	255
19.2.4.3	闹钟中断时寄存器 RTCALRH	255
19.2.4.4	闹钟中断星期寄存器 RTCALRW	256
19.2.5	时间节拍中断寄存器 RTCTTR	256
19.2.6	实时时钟校正寄存器 RTCFCR	256
19.2.7	实时时钟寄存器	257
19.2.7.1	实时时钟秒寄存器 RTCSEC	257
19.2.7.2	实时时钟分寄存器 RTCMIN	257
19.2.7.3	实时时钟时寄存器 RTCHOUR	257
19.2.7.4	实时时钟星期寄存器 RTCWEK	258
19.2.7.5	实时时钟日寄存器 RTCDAY	258
19.2.7.6	实时时钟月寄存器 RTCMTH	258
19.2.7.7	实时时钟年寄存器 RTCYEAR	258
19.2.7.8	实时时钟定时器使能寄存器 RTCTMREN	258
19.2.7.9	实时时钟定时器控制寄存器 RTCTMRCTL	259
19.2.7.10	实时时钟定时器 0 计数寄存器 RTCTMR0	259
19.2.7.11	实时时钟定时器 1 计数寄存器 RTCTMR1	260
19.3	功能描述	260
19.3.1	RTC 初始化	260
19.3.2	RTC 启动	260
19.3.3	RTC 时间设置	260
19.3.4	RTC 闹钟功能	261
19.3.5	时钟校正功能	261
19.3.5.1	计算方法	261
19.3.5.2	校准原理	262
19.3.5.3	校准寄存器设定值表	262
19.3.6	12/24 小时模式选择	263
19.3.7	闰年显示功能	263
19.3.8	RTC 定时器	264
19.3.9	RTC 工作在休眠模式和超低功耗模式	264
19.3.10	RTC 中断	264
19.3.10.1	闹钟中断时间设置	265

19.3.10.2	周期时间节拍中断.....	265
19.3.10.3	进程中断.....	266
19.3.10.4	内置定时器中断.....	266
20	复位	267
20.1	电源控制状态寄存器(PCTL)	268
20.2	上电复位(POR)	269
20.3	WDT 复位	269
20.4	RST 复位	269
20.5	欠压检测复位(LVR)	270
20.6	上电延时定时器	270
20.7	不同复位条件下对寄存器的影响.....	271
21	电源管理和功耗模式	276
21.1	电源	276
21.1.1	备份区电源 Vb _{kp}	277
21.1.2	内置电压调节器.....	277
21.2	电源管理	277
21.3	备份区	278
21.3.1	备份区数据寄存器.....	278
21.3.2	备份区复位.....	279
21.3.3	备份区配置寄存器 (BK _{PCTL})	279
21.3.4	备份区寄存器的读写.....	279
21.4	功耗模式	280
21.4.1	功耗模式相关寄存器.....	280
21.4.1.1	功耗模式控制寄存器.....	280
21.4.1.2	超低功耗模式解锁寄存器.....	281
21.4.1.3	备份区外设复位寄存器.....	281
21.4.1.4	备份区 BOD 控制寄存器.....	282
21.4.1.5	备份区专用看门狗控制寄存器.....	282
21.4.2	正常运行模式.....	283
21.4.3	普通休眠模式.....	283
21.4.4	深度休眠模式.....	283
21.4.5	普通/深度休眠模式下 I/O 口注意事项.....	283
21.4.6	普通/深度休眠模式唤醒方式.....	283
21.4.7	超低功耗模式.....	284
21.4.7.1	进入超低功耗模式的方法.....	284
21.4.7.2	超低功耗模式的唤醒方式.....	285
21.4.7.3	超低功耗模式下 I/O 口状态.....	285
21.4.7.4	备份区专用看门狗.....	286
22	看门狗定时器	287
22.1	看门狗相关寄存器	287
22.1.1	看门狗预分频选择寄存器 WDTPS	287

22.2	看门狗的开启方式	288
22.3	看门狗的清除方式	288
22.4	看门狗的周期	288
23	电气规范	290
23.1	极限参数值	290
23.2	HFINTOSC 的频率精度与 VDD 和温度之间的关系	291
23.3	静态电流特性	292
23.4	休眠电流特性	293
23.5	外设电流特性	294
23.6	I/O 端口电平和芯片供电电压特性	295
23.7	POR	296
23.8	LVR	296
23.9	POWER18	296
23.10	BACKUP BOD	297
23.11	INTHF	297
23.12	DAC12	297
23.13	ADC12	298
23.14	CMP	298
23.15	OP	299
23.16	FVR	299
23.17	INTLF	300
23.18	EXTLF	300
23.19	EXTHF	300
23.20	直流特性图表	301
24	封装信息	311
	附录 1 特殊功能寄存器 (SFR) 功能汇总	313
	附录 2 汇编指令集	318
	附录 3 寄存器全称表	320
	附录 4 全双工异步模式的典型波特率和误差值	325
	产品标识体系	329
	版本信息	330
	ROSH 认证	331
	声明及销售网络	332

1 系统概述

KF8L15Z20XX 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。

KF8L15Z20XX 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
- 1 个 16 位的定时器 T2
- 1 个 16 位定时器/计数器 T3
- 1 个 16 位定时器/计数器 T4 附带 CCP 功能
- 1 个 12 位 6 路外部（+4 路内部）通道 ADC 模块
- 1 个 12 位 DAC 模块
- 4 路 16 位 PWM
- 1 个运算放大器
- 3 个模拟比较器
- 1 个 I2C/SPI 模块
- 2 个 USART 模块(其中 1 个兼容 7816)

温度传感器硬件看门狗（带软件使能）和低电压检测及低电压复位模块等。

芯片内集成了 $(2048+16) \times 8$ 位的数据存储器 RAM、 $10K \times 16$ 位的程序存储器和 256×8 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，共 39 个中断源

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）

支持 16 级硬件堆栈

工作频率默认内部高频时为 62.5KHz~16MHz，软件可选时钟源及分频

● 存储器

10K×16 位 FLASH 程序存储器

(2048+16)×8 位的数据存储器

256×8 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 16MHz±1%

内部可校正低频 32KHz 时钟

低功耗 RTC 时钟

提供一个 2V/3V/4V 可选的参考电压，精度为±5%

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： 均为双向输入输出口

内置上拉功能：所有 I/O 口均带有弱上拉功能

电平变化中断：P0 和 P3 口均有电平变化中断功能

IO 口数字输入类型：所有 I/O 口均为 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：带重载功能、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：带有重载功能、预分频器及时钟源多选的 16 位定时器/计数器

定时器 4：带有重载功能、预分频器及时钟源多选的 16 位定时器/计数器

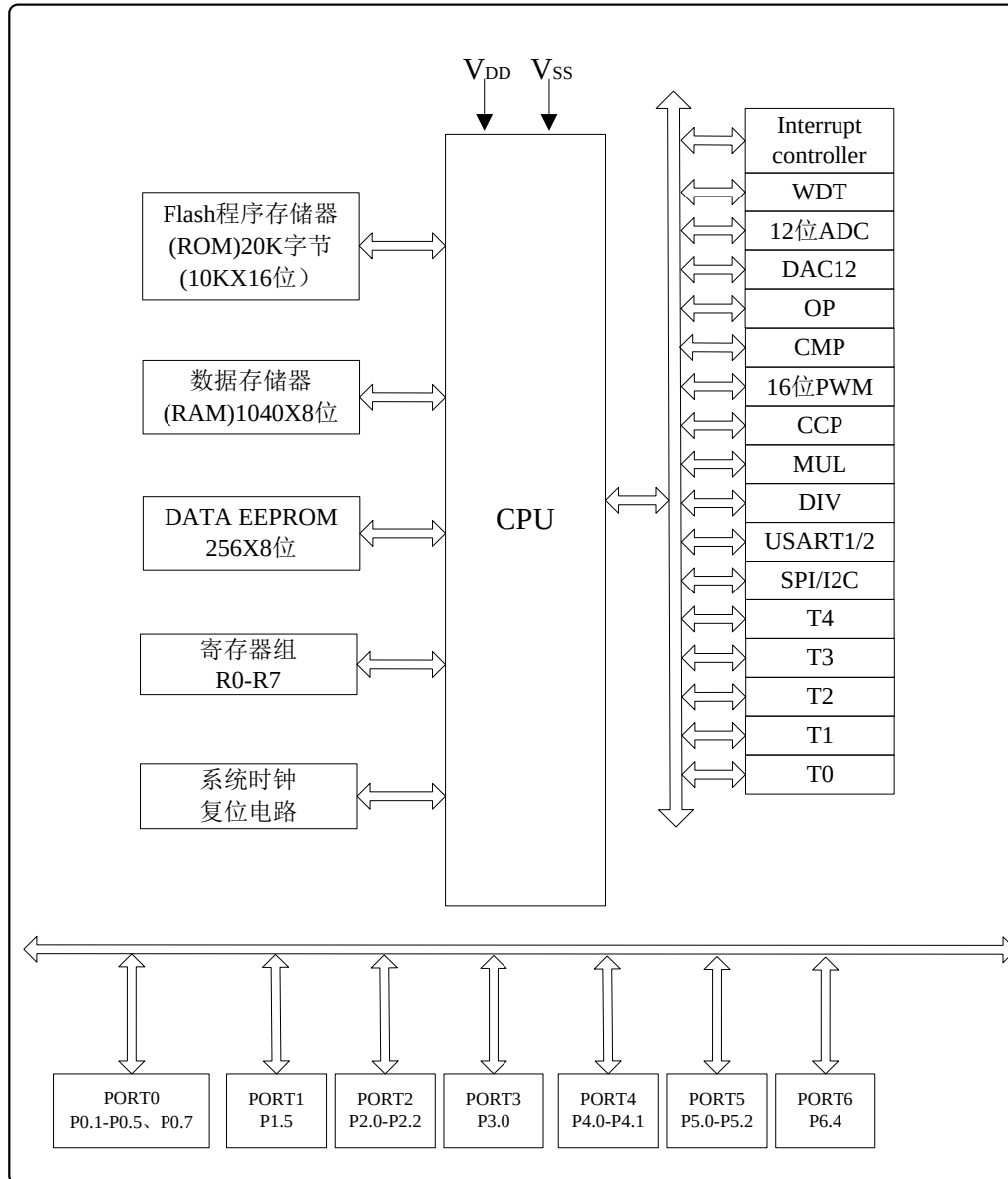
- 其它外设

- 1 个 12 位 6 路外部 (+4 路内部) 通道 ADC 模块
- 1 个 12 位 DAC 模块
- 4 路 16 位脉宽调制 PWM 模块
- 1 个运算放大器模块
- 3 个模拟比较器 (上下沿中断可选)
- 1 个 I2C/SPI 模块
- 2 个 USART 模块
- 1 个温度传感器

- 工作条件

- 工作电压: 1.8V~5.5V
- 工作温度范围: -40~85℃(工业级)

1.2 系统框图



1.3 存储器

KF8L15Z20XX 单片机的存储器包含:程序存储器(ROM)、数据存储器(RAM)和 DATA EEPROM。

KF8L15Z20XX 的程序存储器空间为 20K 字节 (10K×16 位), 寻址范围为 0000H~27FFH, 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 15。通用存储器区 0 至通用存储器区 15 有 128×16 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~FFH。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

振荡周期又叫时钟周期，是振荡器振荡频率的倒数，系统时钟是由振荡器时钟分频而来。本芯片中一个机器周期等于四个系统周期，如图 1.1 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。

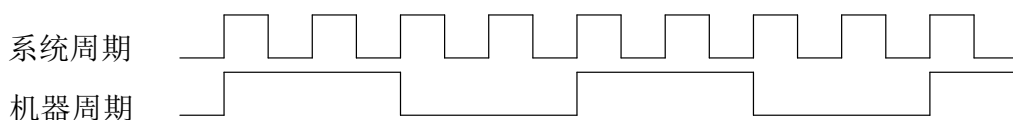


图 1.1 机器周期

KF8L15Z20XX 系列单片机提供 4 个可选时钟源：

内部高频振荡器 INTHF:以内部高频振荡器为时钟源；

内部低频振荡器 INTLF:以内部低频振荡器为时钟源；

外部高频振荡器 EXTHF:标准晶振、陶瓷谐振器或外接 20MHz~125KHz 的时钟源工作；

外部低频振荡器 EXTLF:低电流工作并使用外接 32.768KHz 的钟表晶振。

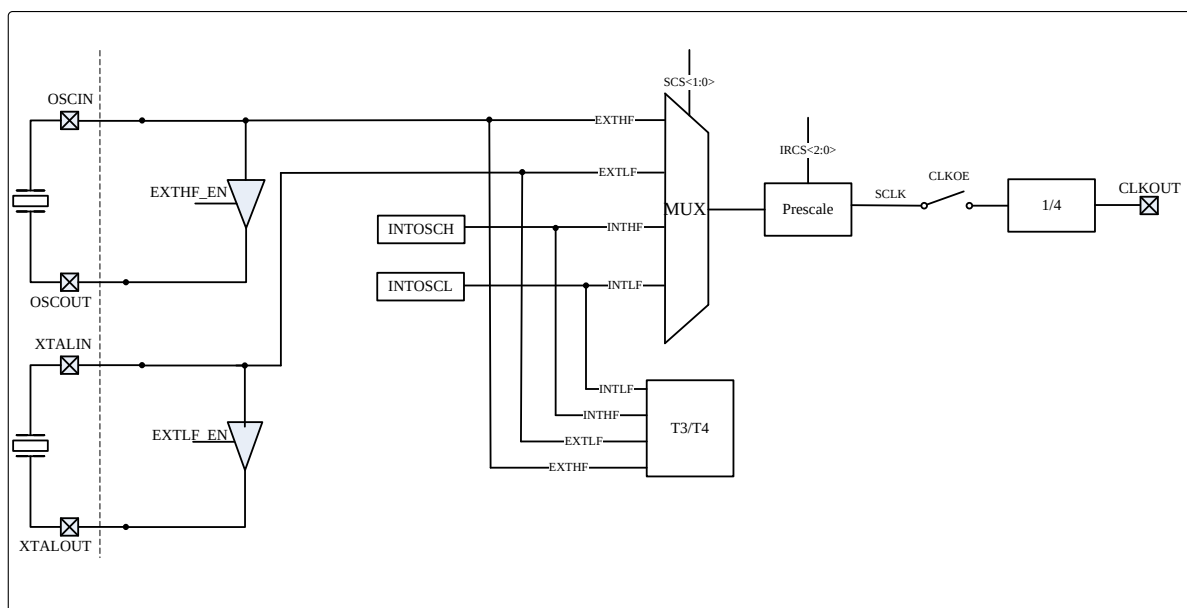


图 1.2 时钟模块原理框图

KF8L15Z20XX 系列单片机的系统时钟源可以配置为 EXTHF、EXTLF、INTHF 或者 INTLF。同时 4 个可选时钟源还可以作为部分外设的工作时钟源。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。

表 1-1 与时钟有关的名词表述

名称	定义即描述
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

INTHF	定义为内部高频振荡器（或时钟源）
INTLF	定义为内部低频振荡器（或时钟源）
EXTHF	定义为外部高频振荡器（或时钟源）
EXTLF	定义为外部低频振荡器（或时钟源）
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1.2: OSCCTL系统频率控制寄存器(地址:2FH)

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

注: R=可读 W=可写 -=未用 x=不定 后面与此相同不再复述

CLKOE: 系统时钟输出使能位
1=使能系统时钟四分频输出
0=禁止系统时钟四分频输出

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>: 系统时钟源选择位

00=选择内部高频时钟
01=选择内部低频时钟
10=选择外部低频时钟
11=选择外部高频时钟

IESO: 双速模式使能位

0=禁止双速功能
1=启动双速功能

FSCM: 外部时钟故障检测使能位

0=禁止故障检测功能
1=使能故障检测功能

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 -110 --00	bit7							bit0
	-	OSTS	HTS	LTS	-	-	SCF1	SCF0
	U	R	R	R	U	U	R	R

注: R=可读 W=可写 -=未用 x=不定 后面与此相同不再复述

- OSTS: 内外时钟标志位
0=系统时钟为外部时钟
1=系统时钟为内部时钟
- HTS: 内部高频时钟稳定位
0=内部高频时钟未稳定
1=内部高频时钟稳定
- LTS: 内部低频时钟稳定位
0=内部低频时钟未稳定
1=内部低频时钟稳定
- SCF<1:0>: 系统时钟源标志位
00=当前系统时钟源为内部高频时钟
01=当前系统时钟源为内部低频时钟
10=当前系统时钟源为外部低频时钟
11=当前系统时钟源为外部高频时钟

1.4.2 上电延时

KF8L15Z20XX 系列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: $T_{\text{pwrt}} = \frac{2^{10}}{\text{PWRTCLK}}$ 。

1.4.3 内部高频振荡器

KF8L15Z20XX 系列单片机的内部高频时钟由系统内部高频振荡器提供, 时钟频率为 16MHz, 精度为 $\pm 1\%$ 。

1.4.3.1 校准寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
4DH	OSCCAL0	内部高频晶振校准寄存器 0							
37H	OSCCAL1	内部高频晶振校准寄存器 1							
2AH	OSCCAL2	内部高频晶振校准寄存器 2							
4CH	OSCCAL3	内部高频晶振校准寄存器 3							

OSCCAL0、OSCCAL1、OSCCAL2 和 OSCCAL3 为内部高频振荡器时钟校准寄存器, 用来存放系统时钟校准值。用户在编程时, 需要在程序初始化部分将存放在程序空间 27FFH、

27FEH、27FDH 和 27FCH 的晶振校准值读出来分别存放到内部高频振荡器的校准寄存器中，否则会导致系统时钟频率不准。

程序示例请参考例 1.1。

例1.1 读内部高频晶振校准值

```

MOV P #0X20 ;切换到PAGE2区
CALL 0X7FF
MOV OSCCAL0, R0
CALL 0X7FE
MOV OSCCAL1, R0
CALL 0X7FD
MOV OSCCAL2, R0
CALL 0X7FC
MOV OSCCAL3, R0
MOV P #0X00 ;切换回PAGE0区
    
```

1.4.4 内部低频振荡器

KF8L15Z20XX 模块内置内部低频振荡器为系统内部低频振荡器，其振荡器频率位 32KHz，它不仅可以为 SCLK 时钟提供时钟源，而且还可以作为独立时钟信号供外设模块使用。也可以作为内部上电延时定时器和看门狗 WDT 定时器的时钟。

LPRCCAL 为内部低频振荡器的时钟校准寄存器，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 2FF8H 的晶振校准值存放到 LPRCCAL 中。程序示例请参考例 1.2。

例1.2 读内部低频晶振校准值

```

MOV P #0X20 ;切换到PAGE2区
CALL 0X7F8
MOV LPRCCAL , R0
MOV P #0X00 ;切换回PAGE0区
    
```

1.4.4.1 校准寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
366H	LPRCCTL	-	-	-	-	-	LPRC2	LPRC1	LPRC0
365H	LPRCCAL	内部低频晶振校准寄存器							

寄存器1.1: LPRCCTL:内部低频RC控制寄存器(地址:366H)

bit7					bit0		
复位值 0000 0000	-	-	-	-	LPRC2	LPRC1	LPRC0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W

注: R=可读 W=可写 -=未用 x=不定 后面与此相同不再复述

LPRC<2:0>: 低功耗内部低频振荡器电流偏置选择位

111=max

...

100=默认值

...

001=min

000=关闭

1.4.5 外部高频振荡器

如图 1.3 所示, 引脚 OSCIN 和引脚 OSCOUT 可以接外部标准晶体、陶瓷谐振器或外接 20MHz~125KHz 的时钟为时钟源。它不仅可以为 SCLK 时钟提供时钟源, 而且还可以作为独立时钟信号供外设模块使用。

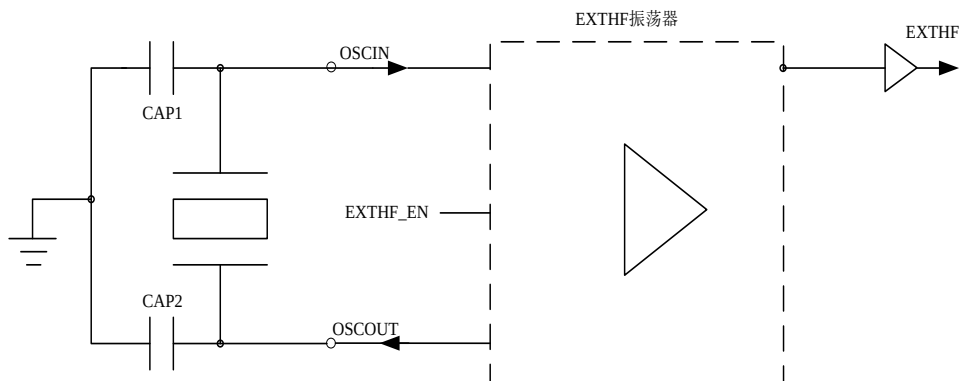


图 1.3 外部高频振荡器原理图

表: 外部高频振荡器相关名称说明

名称	说明
OSCIN	外部高频晶振输入引脚, 位于 P0.7/OSCIN
OSCOUT	外部高频晶振输出引脚, 位于 P5.0/OSCOUT
EXTHF_EN	外部高频晶振软件&硬件使能信号
EXTHF	外部高频时钟信号
CAP1&CAP2	外部高频晶振输入输出引脚外接到地电容, 推荐 10pF

1.4.6 外部低频振荡器

如图 1.4 所示，引脚 XTALIN 和引脚 XTALOUT 接外部 32.768KHz 石英晶振。EXTLF 可以作为主系统时钟 SCLK 的时钟源，还可以作为独立时钟信号供外设模块使用。

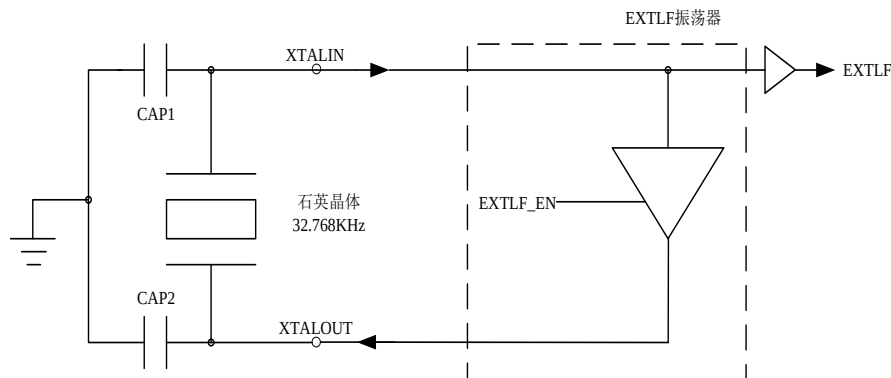


图 1.4 外部低频振荡器原理图

表：外部低频振荡器相关名称说明

名称	说明
XTALIN	外部低频晶振输入引脚，位于 P0.4/XTALIN
XTALOUT	外部低频晶振输出引脚，位于 P0.5/XTALOUT
EXTALF_EN	外部低频晶振软件&硬件使能信号
EXTALF	外部低频时钟信号
CAP1&CAP2	外部低频晶振输入输出引脚外接到地电容，推荐 10pF

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 位复位为 00，即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 配置后立即对时钟进行切换。

当配置 SCS 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 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 位配置的外部时钟。

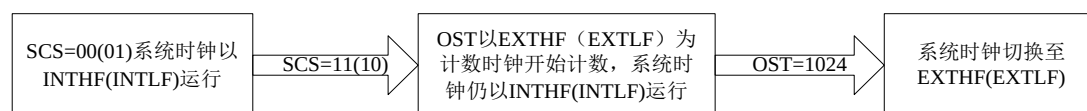


图 1.5 时钟切换流程图

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 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, 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。

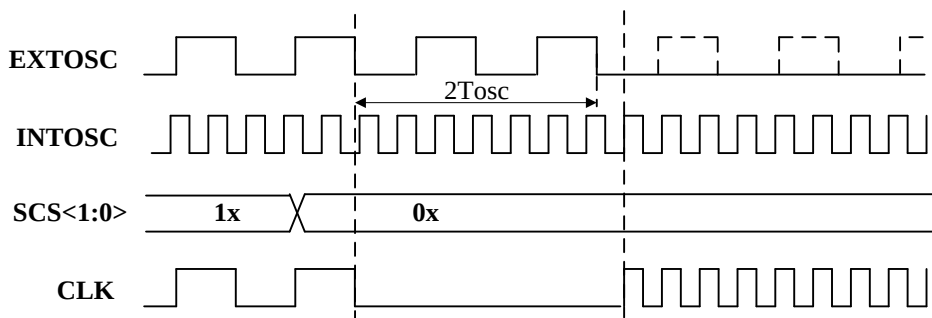


图 1.6 切换至内部时钟源时时钟信号同步时序图

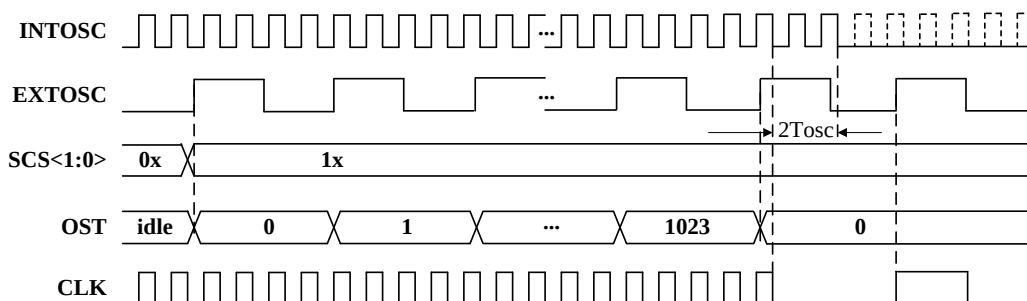


图 1.7 切换至外部时钟源时时钟信号同步时序图

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。

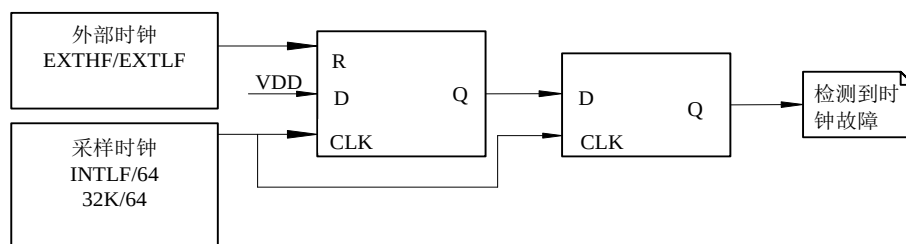


图 1.8 时钟故障检测原理图

FSCM 模块通过比较外部振荡器和采样时钟检测使用的外部振荡器。

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

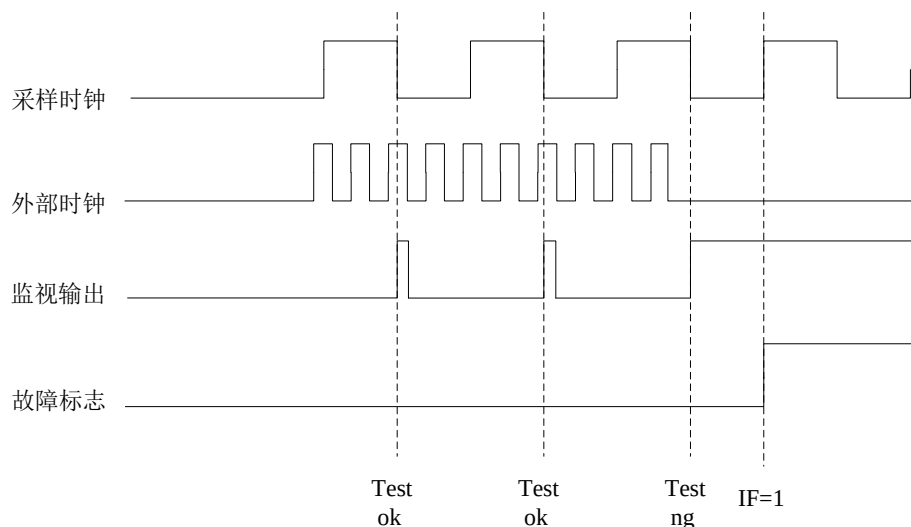


图 1.9 时钟故障检测时序

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

用户在烧写程序时，在编程器中通过对配置位进行设置，使单片机启用诸如看门狗、程序代码保护、欠压检测等功能。KF8L15Z20XX 的配置位映射在 8007H 和 8008H 地址单元。

CONFIG: 配置字(地址:8007H)

R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P
---	---	---	---	---	DEBUG	SWRTEN1	---	CODEP	LVREN	RSTEN	PWRT	WDTEN	DATAP	---	---	SWRTEN0
bit15								bit8				bit0				

注: R=编程器可读 P=编程时可写

- DEBUG:** 在线调试使能位
 DEBUG=1 禁止在线调试
 DEBUG=0 使能在线调试
- CODEP:** 代码保护使能位
 CODEP=1 禁止程序存储器代码保护
 CODEP=0 使能程序存储器代码保护
- LVREN:** 欠压检测功能使能位
 LVREN=1 使能欠压检查功能
 LVREN=0 禁止欠压检查功能
- RSTEN:** P0.3/ $\overline{\text{RST}}$ 引脚功能选择
 RSTEN=1 P0.3/ $\overline{\text{RST}}$ 引脚配置为外部复位输入
 RSTEN=0 P0.3/ $\overline{\text{RST}}$ 引脚功能为数字输入口
- PWRT:** 上电延时使能位
 PWRT=0 使能上电延时
 PWRT=1 禁止上电延时
- WDTEN:** 看门狗定时器(WDT)使能位
 WDTEN=1 使能 WDT
 WDTEN=0 禁止 WDT
- DATAP:** 数据存储区加密使能位
 CPD=1 禁止数据存储区加密
 CPD=0 使能数据存储区加密
- SWRTEN<1:0>:** Flash 自写使能位
 SWRTEN<1:0> =00 写保护关闭，全区域可写
 SWRTEN<1:0> =01 0000H 至 01FFH 受写保护
 SWRTEN<1:0> =10 0000H 至 0FFFH 受写保护
 SWRTEN<1:0> =11 0000H 至 27FFH 受写保护，全区域不可写

CONFIG: 校验字(地址:8008H)

R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P
---	---	---	---	---	---	---	---	POR2	POR1	POR0	BOD4	BOD3	BOD2	BOD1	BOD0

bit15

bit8

bit0

注: R=编程器可读 P=编程时可写

BOR<4:0>: BOD 电压校验位

POR<2:0>: POR 电压校验位

1.6 在线串行编程

如图 1.10、1.11 所示，在最终应用电路中可对 KF8L15Z20XX 单片机进行在线串行编程。实现编程仅需要四根线包括：时钟线(SPCLK)、数据线(SPDAT)、电源线(VDD)、地线(Vss)。

开发人员和用户可以使用未编程的单片机来制造电路板，然后对其在线编程，调试等。只要有电脑、USB 下载线和编程器，即可在任何时候，任何地点，对电路板上的单片机程序进行更新。

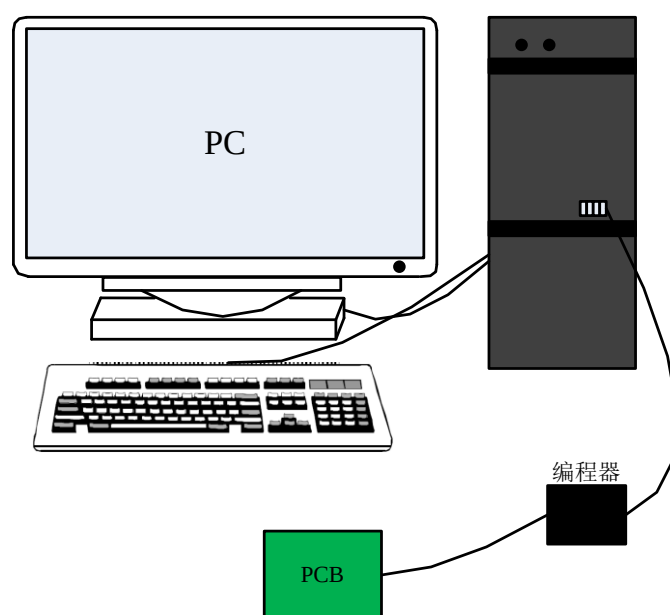


图 1.10 在线调试系统示意图

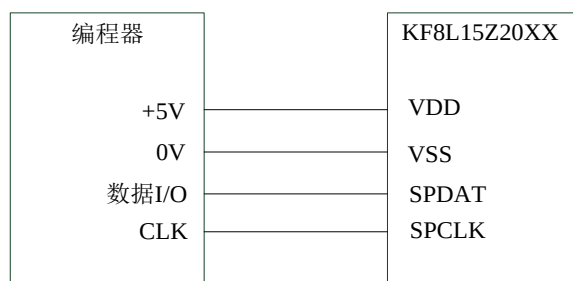


图 1.11 在线串行编程连接图

2 I/O 端口

KF8L15Z20XX 单片机引脚包括：

- 1 组电源地 VDD/VSS
- 一个内部电压外接电容脚 VDDcore
- 其余管脚均为 I/O 端口

2.1 I/O 端口的读写

读 I/O 口时实际为读端口电平，即读 P_x ($x=0\sim6$)，而写 P_x 时实际为写 P_{xLR} ($x=0\sim6$) 寄存器。其原理框图如图 2.1 所示：

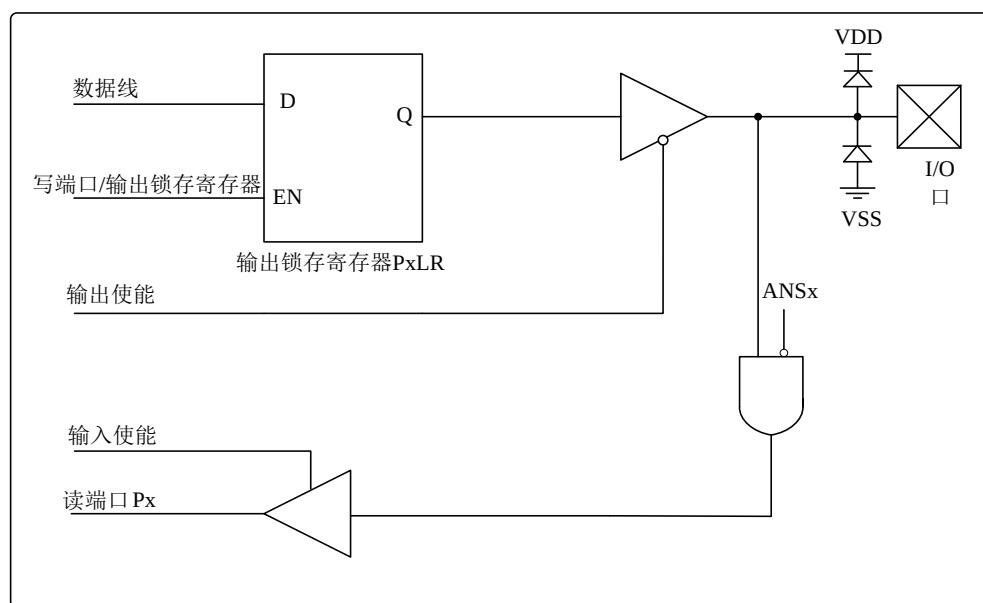


图 2.1 I/O 口读写原理图

IO 端口作为输出时，可对 P_{xLR} 进行赋值(寄存器操作或位操作)，以避免原来的 IO 端口的读-修改-写操作引起的错误。

2.2 P0 口

在线编程时 P0 口的 SPCLK、SPDAT 作为编程脚使用。所有 P0 口均可作为普通 I/O 口且带有上拉功能，P0 口所有引脚都有电平变化中断功能。

特别注意，P0.3、P0.4、P0.5 和 P0.6 四个 I/O 口的供电电源为 Vb_{kp}，因此在 V_{bat} 维持备份区供电、VDD 掉电的条件下，仍旧可以实现保持 I/O 口状态、外部低频晶振继续工作、RTC 时钟信号输出和 RST 复位功能。同时，其内部结构和其他普通 I/O 口有所不同，详见 P0 口各引脚内部原理功能框图章节。P0.3~P0.6 口结构上的差异导致其拉电流能力弱于其他通用 I/O 口；另外，在 P0.3~P0.6 口外接器件时，请考虑在高电平输出驱动管下的 1KΩ 电阻，如直接外接下拉电路，可能导致 P0.3~P0.6 口输出电平被分压。

2.2.1 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	P07	P06	P05	P04	P03	P02	P01	P00
45H	P0LR	P0LR7	P0LR6	P0LR5	P0LR4	P0LR3	P0LR2	P0LR1	P0LR0
25H	TR0	TR07	TR06	TR05	TR04	TR03	TR02	TR01	TR00
36H	IOCL0	IOCL07	IOCL06	IOCL05	IOCL04	IOCL03	IOCL02	IOCL01	IOCL00
35H	PUR0	PUR07	PUR06	PUR05	PUR04	PUR03	PUR02	PUR01	PUR00
31H	ANS0	ANS07	ANS06	ANS05	ANS04	ANS03	ANS02	ANS01	ANS00

2.2.1.1 P0 口状态寄存器（P0）

寄存器 P0 各位对应 P0 口相应引脚当前的状态，如寄存器 2.1 所示：

寄存器 2.1: P0: P0 口状态寄存器(地址: 05H)

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

P0<7><5:1>: 读 P0 口各端口电平
P0_x=1 对应引脚为逻辑高电平
P0_x=0 对应引脚为逻辑低电平
P0<6><0>: 保留位

2.2.1.2 P0 口输出锁存寄存器（P0LR）

寄存器 P0LR 是 P0 口输出锁存寄存器。在 P0 口作为输出时，通过写 P0LR 寄存器来设置输出 P0 口的状态。

寄存器 2.2: P0LR: P0 口输出锁存寄存器(地址: 45H)

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

P0LR<7><5:1>: 写 P0 口输出状态

POLR x=1 对应引脚输出高电平

POLR x=0 对应引脚输出低电平

POLR<6><0>: 保留位

2.2.1.3 P0 口方向控制寄存器(TR0)

如寄存器 2.3 所示, TR0 为 P0 口方向控制寄存器, 当 TR0 某位置 1 时, 将该引脚设置为输入, 此时引脚为三态(悬空), TR0 某位清 0, 对应引脚设置为输出。

寄存器2.3: TR0: P0口方向控制寄存器(地址: 25H)

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

TR0<7><5:1>: P0 口各引脚方向控制位

TR0x=1 对应的引脚设置为输入

TR0x=0 对应的引脚设置为输出

TR0<6><0>: 保留位

2.2.1.4 P0 上拉功能控制寄存器(PUR0)

KF8L15Z20XX 所有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 P0 口上拉功能打开, 然后再将要打开上拉功能的引脚, 所对应的上拉功能控制位置 1 即可。寄存器 2.4 为上拉功能控制寄存器。

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

寄存器2.4: PUR0: P0口弱上拉控制寄存器(地址: 35H)

复位值 1111 1111	bit7							bit0
	PUR07	PUR06	PUR05	PUR04	PUR03	PUR02	PUR01	PUR00
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUR0<7><5:1>: 上拉功能使能位

PUR0x=1 使能对应的端口上拉功能

PUR0x=0 禁止对应的端口上拉功能

PUR0<6><0>: 保留位

2.2.1.5 P0 口电平变化中断控制寄存器(IOCL0)

P0 口每个引脚都具有电平变化中断功能, 当引脚的当前电平与上次读 P0 寄存器时的电平不匹配时将产生电平变化中断。如寄存器 2.5 所示, IOCL0 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2.5: IOCL0: P0口电平变化中断控制寄存器(地址:36H)

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

IOCL0<7><5:1>: P0 端口引脚电平变化中断使能控制位

IOCL0x=1 使能对应引脚的电平变化中断

IOCL0x=0 禁止对应引脚的电平变化中断

IOCL0<6><0>: 保留位

2.2.1.6P0 口模拟/数字口设置寄存器 (ANS0)

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

寄存器2.6: ANS0: P0口模拟/数字口设置寄存器(地址:31H)

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

ANS0<7><5:1>: P0 口各引脚模拟/数字口设置位

1 = 将对应引脚配置为模拟口

0 = 将对应引脚配置为数字口或者特殊功能引脚

ANS0<6><0>: 保留位，请勿清零

2.2.2 P0 口各引脚内部原理功能框图

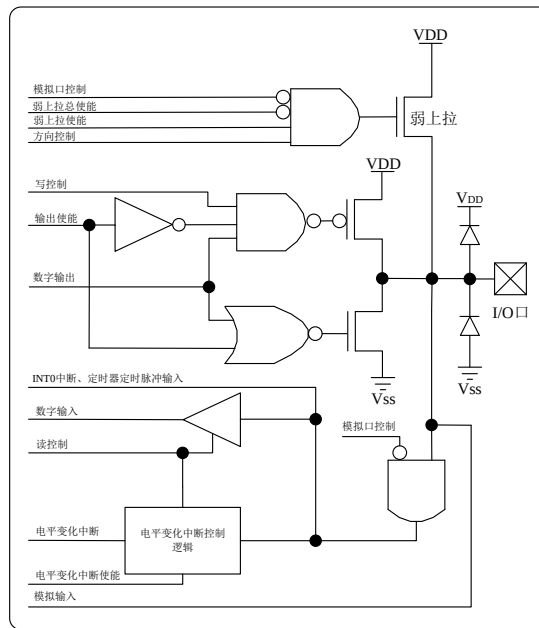


图 2.3 P0 口引脚内部原理框图

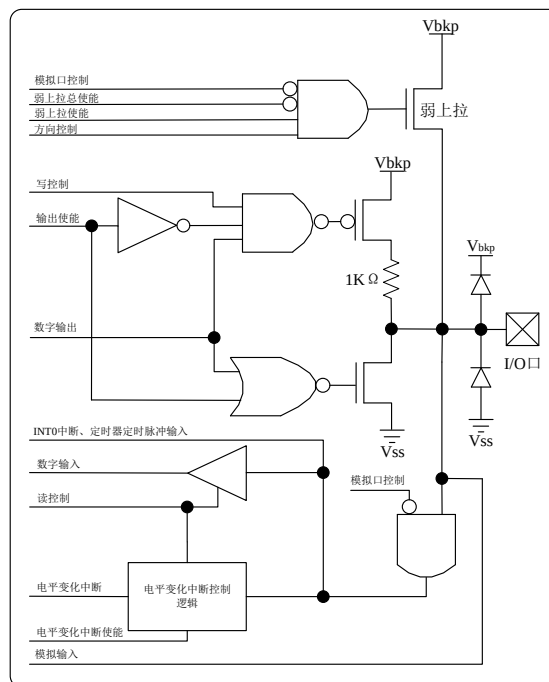


图 2.3.2 P0.3-P0.6 口引脚内部原理框图

注：

- 1.在 P0.3~P0.6 口外接器件时，请考虑在高电平输出驱动管下的 1KΩ 电阻，如直接外接下拉电路，可能导致 P0.3~P0.6 口输出电平被分压。
- 2.P0.3~P0.6 口的最大输出拉电流能力为 370uA。

2.3 P1 口

P1 口所有管脚均可作为普通 I/O 口且带有上拉功能。

2.3.1 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
1DH	ANS1	ANS17	ANS16	ANS15	ANS14	ANS13	ANS12	ANS11	ANS10

2.3.1.1 P1 口状态寄存器(P1)

寄存器 P1 对应端口 P1 引脚作为普通 I/O 口时的状态。如寄存器 2.7 所示

寄存器2.7: P1: P1口状态寄存器(地址: 07H)

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

P15: P1 口各引脚状态位
P1x=1 对应引脚为逻辑高电平
P1x=0 对应引脚为逻辑低电平
P1<7:6><4:0>: 保留位

2.3.1.2 P1 口输出锁存寄存器 (P1LR)

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

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

P1LR5: 写 P1 口输出状态
P1LRx=1 对应引脚输出高电平
P1LRx=0 对应引脚输出低电平
P1 LR <7:6><4:0>: 保留位

2.3.1.3 P1 口方向控制寄存器(TR1)

如寄存器 2.9 所示, 通过将寄存器 TR1 中的某位置 1, 将对应管脚设置为输入口。清 0 设置为输出口, 系统复位时, P1 口各引脚默认为输入口。

寄存器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

TR15: P1 口引脚方向控制位

TR1x=1 P1 口对应引脚被配置为输入端口

TR1x=0 P1 口对应引脚被配置为输出端口

TR1<7:6><4:0>: 保留位

2.3.1.4 P1 口上拉功能控制寄存器(PUR1)

KF8L15Z20XX 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 P1 口上拉功能打开, 然后再将要打开上拉功能的引脚, 所对应的上拉功能控制位置 1 即可。寄存器 2.10 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

PUR15: 上拉功能使能位

PUR1x=1 使能对应的端口上拉功能

PUR1x=0 禁止对应的端口上拉功能

PUR1<7:6><4:0>: 保留位

2.3.1.5 P1 口模拟/数字口设置寄存器 (ANS1)

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

寄存器2.11: ANS1: P1口模拟/数字口设置寄存器(地址: 1DH)

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

ANS15: P1 口各引脚模拟/数字口设置位
 1 = 将对应引脚配置为模拟口
 0 = 将对应引脚配置为数字口或者特殊功能引脚
 ANS1<7:6><4:0>: 保留位, 请勿清零

2.3.2 P1 口原理功能框图

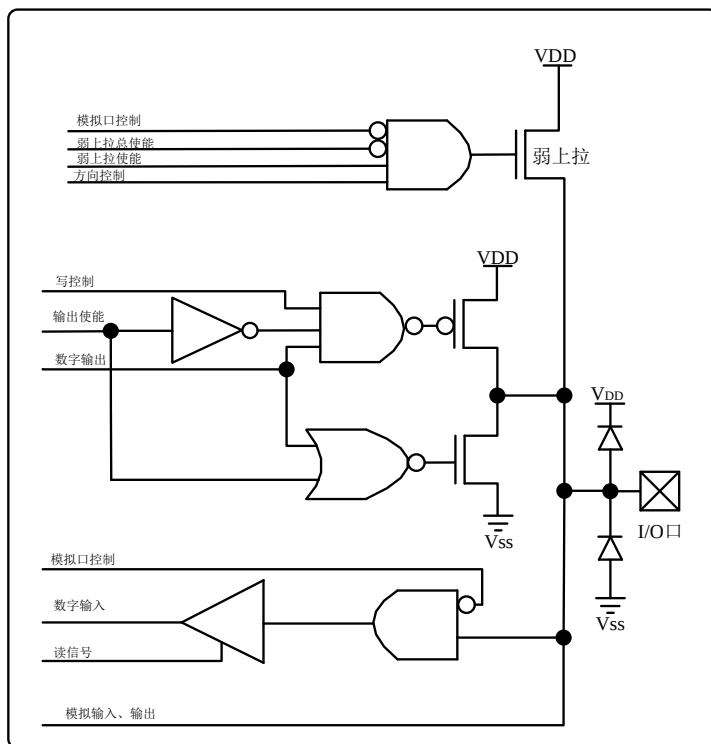


图 2.4 P1 口引脚原理功能框图

2.4 P2 口

P2 口均可作为普通 I/O 口且带有上拉功能。

2.4.1 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	P27	P26	P25	P24	P23	P22	P21	P20
46H	P2LR	P2LR7	P2LR6	P2LR5	P2LR4	P2LR3	P2LR2	P2LR1	P2LR0
26H	TR2	TR27	TR26	TR25	TR24	TR23	TR22	TR21	TR20
61H	PUR2	PUR27	PUR26	PUR25	PUR24	PUR23	PUR22	PUR21	PUR20
32H	ANS2	ANS27	ANS26	ANS25	ANS24	ANS23	ANS22	ANS21	ANS20

2.4.1.1 P2 口状态寄存器(P2)

寄存器 P2 各位对应端口 P2 口各引脚作为普通 I/O 口时的状态。如寄存器 2.12 所示：

寄存器2.12: P2: P2口状态寄存器(地址: 06H)

复位值 xxxx xxxx	bit7							bit0	
	P27	P26	P25	P24	P23	P22	P21	P20	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P2<2:0>: P2 口各引脚状态位
P2x=1 对应引脚为逻辑高电平
P2x=0 对应引脚为逻辑低电平
P2<7:3>: 保留位

2.4.1.2 P2 口输出锁存寄存器 (P2LR)

在 P2 口作为输出时，通过写 P2LR 寄存器来设置输出 P2 口的状态。

寄存器2.13: P2LR: P2口输出锁存寄存器(地址: 46H)

复位值 xxxx xxxx	bit7							bit0	
	P2LR7	P2LR6	P2LR5	P2LR4	P2LR3	P2LR2	P2LR1	P2LR0	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P2LR<2:0>: 写 P2 口输出状态
P2LRx=1 对应引脚输出高电平
P2LRx=0 对应引脚输出低电平
P2LR<7:3>: 保留位

2.4.1.3P2 口方向控制寄存器(TR2)

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2.14: TR2: P2口方向控制寄存器(地址: 26H)

复位值 1111 1111	bit7						bit0	
	TR27	TR26	TR25	TR24	TR23	TR22	TR21	TR20
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TR2<2:0>: P2 口各引脚方向控制位

TR2x=1 P2 口对应引脚被配置为输入端口

TR2x=0 口对应引脚被配置为输出端口

TR2<7:3>: 保留位

2.4.1.4P2 口上拉功能控制寄存器(PUR2)

KF8L15Z20XX 中 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 P2 口上拉功能打开，然后再将要打开上拉功能的引脚，所对应的上拉功能控制位置 1 即可。

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2.15: PUR2: P2口弱上拉控制寄存器(地址: 61H)

复位值 1111 1111	bit7						bit0	
	PUR27	PUR26	PUR25	PUR24	PUR23	PUR22	PUR21	PUR20
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUR2<2:0>: 上拉功能使能位

PUR2x=1 使能对应的端口上拉功能

PUR2x=0 禁止对应的端口上拉功能

PUR2<7:3>: 保留位

2.4.1.5P2 口模拟/数字口设置寄存器 (ANS2)

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

寄存器2.16: ANS2: P2口模拟/数字口设置寄存器(地址: 32H)

复位值 1111 1111	bit7						bit0	
	ANS27	ANS26	ANS25	ANS24	ANS23	ANS22	ANS21	ANS20
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ANS2<2:0>: P2 口各引脚模拟/数字口设置位

1 = 将对应引脚配置为模拟口

0 = 将对应引脚配置为数字口或者特殊功能引脚

ANS2<7:3>: 保留位，请勿清零

2.4.2 P2 口原理功能框图

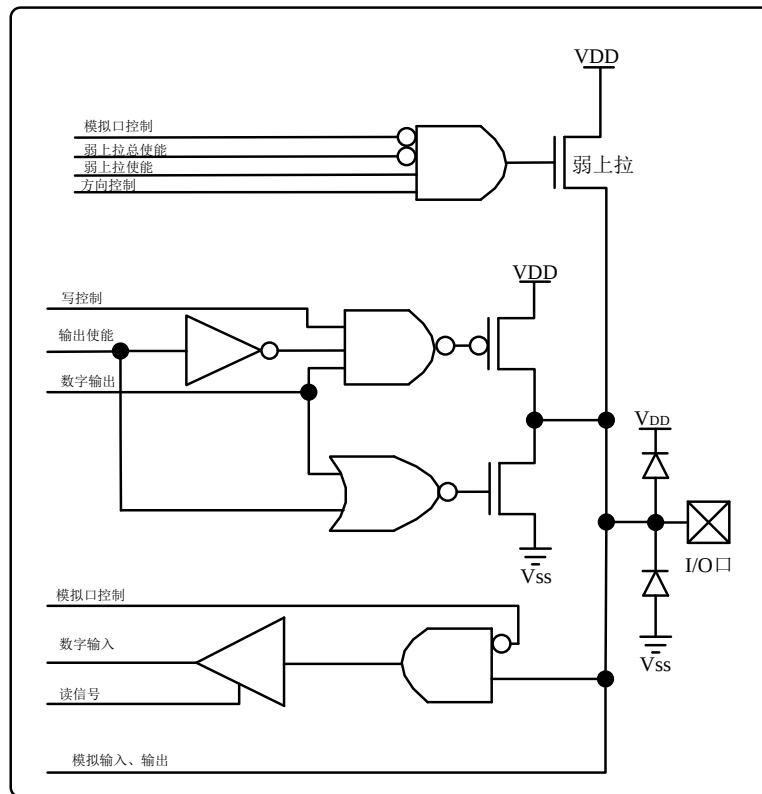


图 2.5 P2 口引脚原理功能框图

2.5 P3 口

P3 口可作为普通 I/O 口且带有上拉功能。P3 口带有电平变化中断功能。

2.5.1 P3 口相关的寄存器

表 2-8 与 P3 口相关的寄存器

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
08H	P3	P37	P36	P35	P34	P33	P32	P31	P30
48H	P3LR	P3LR7	P3LR6	P3LR5	P3LR4	P3LR3	P3LR2	P3LR1	P3LR0
49H	TR3	TR37	TR36	TR35	TR34	TR33	TR32	TR31	TR30
6CH	PUR3	PUR37	PUR36	PUR35	PUR34	PUR33	PUR32	PUR31	PUR30
6BH	IOCL3	IOCL37	IOCL36	IOCL35	IOCL34	IOCL33	IOCL32	IOCL31	IOCL30
33H	ANS3	ANS37	ANS36	ANS35	ANS34	ANS33	ANS32	ANS31	ANS30

2.5.1.1 P3 口状态寄存器(P3)

寄存器 P3 各位对应端口 P3 口各引脚作为普通 I/O 口时的状态。如寄存器 2.17 所示。

寄存器2.17: P3: P3口状态寄存器(地址:08H)

复位值 xxxx xxxx	bit7				bit0			
	P37	P36	P35	P34	P33	P32	P31	P30
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P30: P3 口各引脚状态位

P3x=1 对应引脚为逻辑高电平

P3x=0 对应引脚为逻辑低电平

P3<7:1>: 保留位

2.5.1.2 P3 口输出锁存寄存器 (P3LR)

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

寄存器2.18: P3LR: P3口输出锁存寄存器(地址:48 H)

复位值 xxxx xxxx	bit7				bit0			
	P3LR7	P3LR6	P3LR5	P3LR4	P3LR3	P3LR2	P3LR1	P3LR0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P3LR0: 写 P3 口输出状态

P3LRx=1 对应引脚输出高电平

P3LRx=0 对应引脚输出低电平

P3LR<7:1>: 保留位

2.5.1.3P3 口方向控制寄存器(TR3)

如寄存器 2.19 所示, 通过将寄存器 TR3 中的某位置 1, 将对应管脚设置为输入口。清 0 设置为输出口。

寄存器2.19: TR3: P3口方向控制寄存器(地址: 49H)

复位值 1111 1111	bit7							bit0
	TR37	TR36	TR35	TR34	TR33	TR32	TR31	TR30
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TR30: P3 口各引脚方向控制位
 TR3x=1 P3 口对应引脚被配置为输入端口
 TR3x=0 P3 口对应引脚被配置为输出端口
 P3<7:1>: 保留位

2.5.1.4P3 口上拉功能控制寄存器(PUR3)

KF8L15Z20XX 中 P3 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 P3 口上拉功能打开, 然后再将要打开上拉功能的引脚, 所对应的上拉功能控制位置 1 即可。寄存器 2.20 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2.20: PUR3: P3口弱上拉控制寄存器(地址: 6CH)

复位值 1111 1111	bit7							bit0
	PUR37	PUR36	PUR35	PUR234	PUR33	PUR32	PUR31	PUR30
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUR30: 上拉功能使能位
 PUR3x=1 使能对应的端口上拉功能
 PUR3x=0 禁止对应的端口上拉功能
 PUR3<7:1>: 保留位

2.5.1.5P3 口电平变化中断控制寄存器(IOCL3)

P3 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 P3 寄存器时的电平不匹配时将产生电平变化中断。如寄存器 2.21 所示, IOCL3 为 P3 口电平变化中断控制寄存器, 将 IOCL3 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(P3IF)都会置 1, 如果全局中断使能位(AIE)和电平变化中断使能位(P3IE)都已置 1, 则会响应中断进入中断服务子程序。P3 口所有引脚的电平变化中断共用一个标志位 P3IF。

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. P3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2.21: IOCL3: P3口电平变化中断控制寄存器(地址:6BH)

复位值 0000 0000	bit7							bit0
	IOCL37	IOCL36	IOCL35	IOCL34	IOCL33	IOCL32	IOCL31	IOCL30
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IOCL30: P3 端口引脚电平变化中断使能控制位
 IOCL3=1 使能对应引脚的电平变化中断
 IOCL3=0 禁止对应引脚的电平变化中断
 IOCL3<7:1>: 保留位

2.5.1.6P3 口模拟/数字口设置寄存器 (ANS3)

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

寄存器2.22: ANS3: P3口模拟/数字口设置寄存器(地址:33H)

复位值 1111 1111	bit7							bit0
	ANS37	ANS36	ANS35	ANS34	ANS33	ANS32	ANS31	ANS30
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ANS30: P3 口各引脚模拟/数字口设置位
 1 = 将对应引脚配置为模拟口
 0 = 将对应引脚配置为数字口或者特殊功能引脚
 ANS3<7:1>: 保留位，请勿清零

2.5.2 P3 口原理功能框图

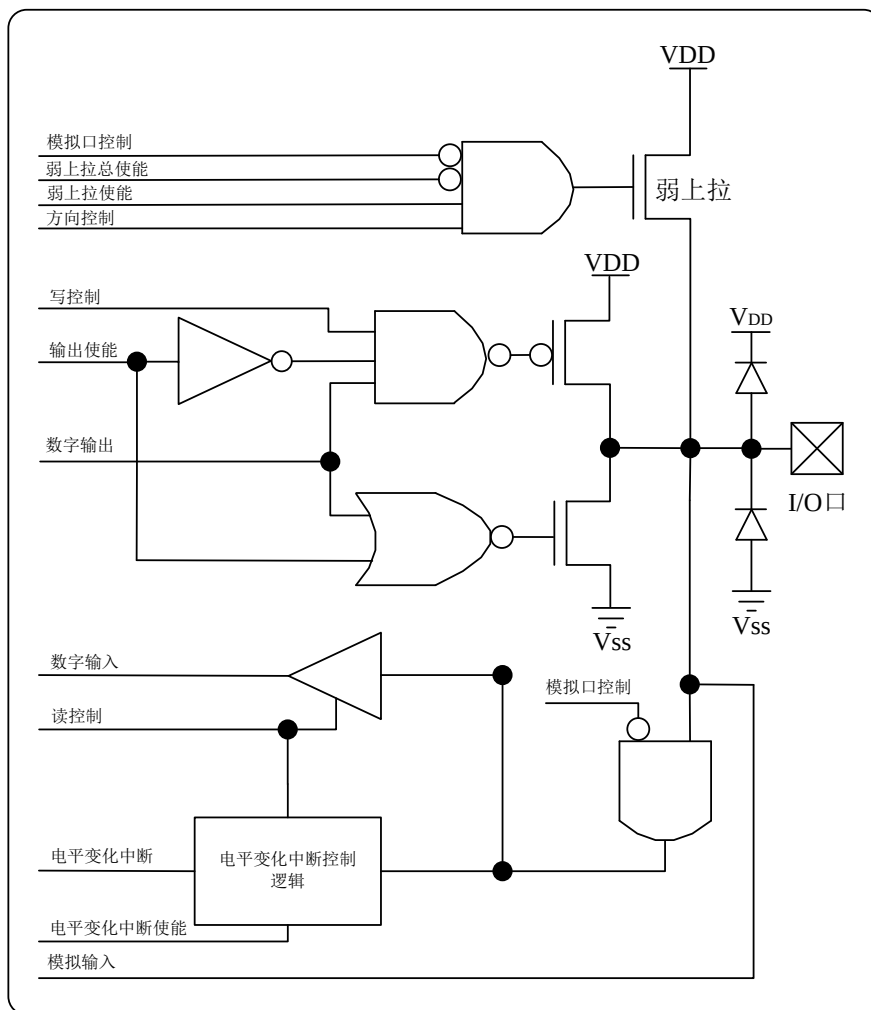


图 2.6 P3 口引脚原理功能框图

2.6 P4 口

P4 口可作为普通 I/O 口且带有上拉功能。

2.6.1 P4 口相关的寄存器

表 2-10 与 P4 口相关的寄存器

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
20CH	P4	P47	P46	P45	P44	P43	P42	P41	P40
20DH	P4LR	P4LR7	P4LR6	P4LR5	P4LR4	P4LR3	P4LR2	P4LR1	P4LR0
20EH	TR4	TR47	TR46	TR45	TR44	TR43	TR42	TR41	TR40
20FH	PUR4	PUR47	PUR46	PUR45	PUR44	PUR43	PUR42	PUR41	PUR40
206H	ANS4	ANS47	ANS46	ANS45	ANS44	ANS43	ANS42	ANS41	ANS40

2.6.1.1 P4 口状态寄存器(P4)

寄存器 P4 各位对应端口 P4 口各引脚作为普通 I/O 口时的状态。如寄存器 2.23 所示：

寄存器 2.23: P4: P4 口状态寄存器(地址: 20CH)

复位值 xxxx xxxx	bit7							bit0
	P47	P46	P45	P44	P43	P42	P41	P40
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P4<1:0>: P4 口各引脚状态位

P4x=1 对应引脚为逻辑高电平

P4x=0 对应引脚为逻辑低电平

P4<7:2>: 保留位

2.6.1.2 P4 口输出锁存寄存器 (P4LR)

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

寄存器 2.24: P4LR: P4 口输出锁存寄存器(地址: 20DH)

复位值 xxxx xxxx	bit7							bit0
	P4LR7	P4LR6	P4LR5	P4LR4	P4LR3	P4LR2	P4LR1	P4LR0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P4LR<1:0>: 写 P4 口输出状态

P4LRx=1 对应引脚输出高电平

P4LRx=0 对应引脚输出低电平

P4LR<7:2>: 保留位

2.6.1.3P4 口方向控制寄存器(TR4)

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

寄存器2.25: TR4: P4口方向控制寄存器(地址: 20EH)

复位值 1111 1111	bit7						bit0	
	TR47	TR46	TR45	TR44	TR43	TR42	TR41	TR40
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TR4<1:0>: P4 口各引脚方向控制位

TR4x=1 P4 口对应引脚被配置为输入端口

TR4x=0 P4 口对应引脚被配置为输出端口

TR4<7:2>: 保留位

2.6.1.4P4 口上拉功能控制寄存器(PUR4)

KF8L15Z20XX 中 P4 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 P4 口上拉功能打开，然后再将要打开上拉功能的引脚，所对应的上拉功能控制位置 1 即可。

注：只有将引脚设置为数字输入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时将会自动禁止该引脚的上拉电阻。

寄存器2.26: PUR4: P4口弱上拉控制寄存器(地址:20FH)

复位值 1111 1111	bit7						bit0	
	PUR47	PUR46	PUR45	PUR44	PUR43	PUR42	PUR41	PUR40
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUR4<1:0>: 上拉功能使能位

PUR4x=1 使能对应的端口上拉功能

PUR4x=0 禁止对应的端口上拉功能

PUR4<7:2>: 保留位

2.6.1.5P4 口模拟/数字口设置寄存器 (ANS4)

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

寄存器2.27: ANS4: P4口模拟/数字口设置寄存器(地址:206H)

复位值 1111 1111	bit7						bit0	
	ANS47	ANS46	ANS45	ANS44	ANS43	ANS42	ANS41	ANS40
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ANS4<1:0>: P4 口各引脚模拟/数字口设置位

1 = 将对应引脚配置为模拟口

0 = 将对应引脚配置为数字口或者特殊功能引脚

ANS4<7:2>: 保留位，请勿清零

2.6.2 P4 口原理功能框图

如引脚示意图所示，P4 口共有 8 个引脚，根据各引脚的作用不同，P4 口引脚原理功能框图中的模拟输入和输出有微小差别。

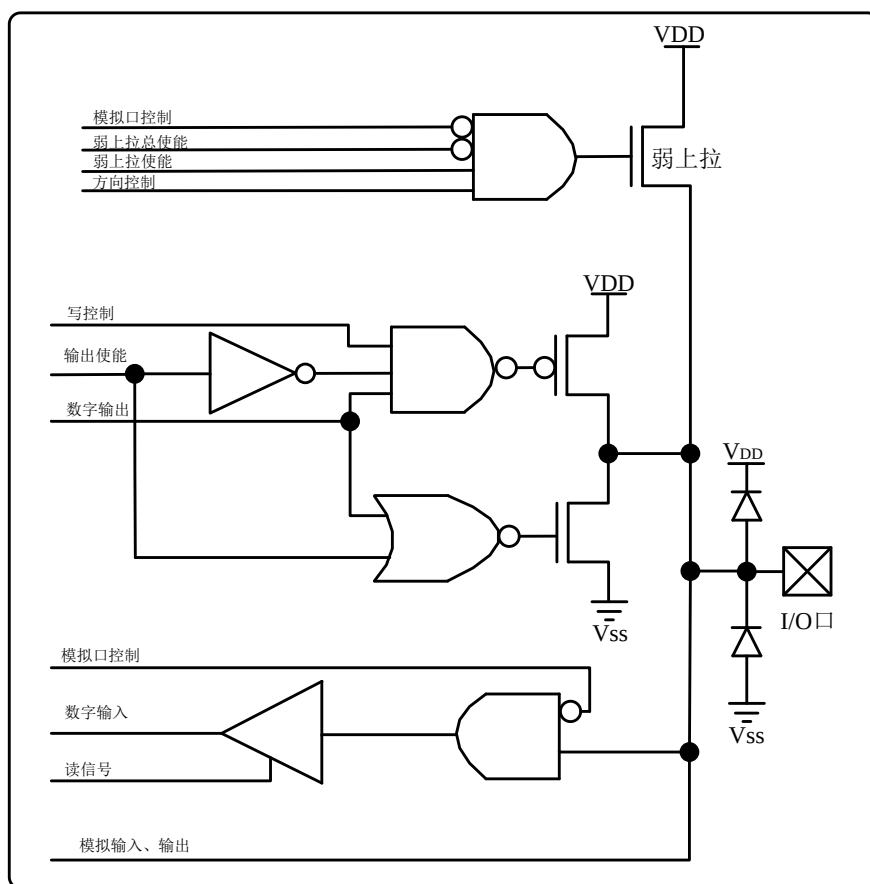


图 2.7 P4 口引脚原理功能框图

2.7 P5 口

P5 口可作为普通 I/O 口且带有上拉功能。

2.7.1 P5 口相关的寄存器

表 2-12 与 P5 口相关的寄存器

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
210H	P5	-	P56	P55	P54	P53	P52	P51	P50
211H	P5LR	-	P5LR6	P5LR5	P5LR4	P5LR3	P5LR2	P5LR1	P5LR0
212H	TR5	-	TR56	TR55	TR54	TR53	TR52	TR51	TR50
213H	PUR5	-	PUR56	PUR55	PUR54	PUR53	PUR52	PUR51	PUR50
207H	ANS5	-	ANS56	ANS55	ANS54	ANS53	ANS52	ANS51	ANS50

2.7.1.1 P5 口状态寄存器(P5)

寄存器 P5 各位对应端口 P5 口各引脚作为普通 I/O 口时的状态。如寄存器 2.28 所示：

寄存器2.28: P5: P5口状态寄存器(地址: 210H)

复位值 -xxx xxxx	bit7							bit0	
	-	P56	P55	P54	P53	P52	P51	P50	-
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P5<2:0>: P5 口各引脚状态位
1 = 对应引脚为逻辑高电平
0 = 对应引脚为逻辑低电平

P5<6:3>: 保留位

2.7.1.2 P5 口输出锁存寄存器 (P5LR)

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

寄存器2.29: P5LR: P5口输出锁存寄存器(地址:211H)

复位值 -xxx xxxx	bit7							bit0	
	-	P5LR6	P5LR5	P5LR4	P5LR3	P5LR2	P5LR1	P5LR0	
	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P5LR<2:0>: 写 P5 口输出状态
1 = 对应引脚输出高电平
0 = 对应引脚输出低电平

P5LR<6:3>: 保留位

2.7.1.3P5 口方向控制寄存器(TR5)

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

寄存器2.30: TR5: P5口方向控制寄存器(地址: 212H)

复位值 -111 1111	bit7						bit0	
	-	TR56	TR55	TR54	TR53	TR52	TR51	TR50
	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TR5<2:0>: P5 口各引脚方向控制位
 1 = P5 口对应引脚被配置为输入端口
 0 = P5 口对应引脚被配置为输出端口

TR<6:3>: 保留位

2.7.1.4P5 口上拉功能控制寄存器(PUR5)

KF8L15Z20XX 中 P5 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 P5 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 2.31 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2.31: PUR5: P5口弱上拉控制寄存器(地址:213H)

复位值 1111 1111	bit7						bit0	
	-	PUR56	PUR55	PUR54	PUR53	PUR52	PUR51	PUR50
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUR5<2:0>: 上拉功能使能位
 1 = 使能对应的端口上拉功能
 0 = 禁止对应的端口上拉功能

PUR5<6:3>: 保留位

2.7.1.5P5 口模拟/数字口设置寄存器 (ANS5)

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

寄存器2.32: ANS5: P5口模拟/数字口设置寄存器(地址:207H)

复位值 1111 1111	bit7						bit0	
	-	ANS56	ANS55	ANS54	ANS53	ANS52	ANS51	ANS50
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ANS5<2:0>: P5 口各引脚模拟/数字口设置位
 1 = 将对应引脚配置为模拟口
 0 = 将对应引脚配置为数字口或者特殊功能引脚

ANS5<6:3>: 保留位，请勿清零

2.7.2 P5 口原理功能框图

如引脚示意图所示，P5 口共有 8 个引脚，根据各引脚的作用不同，P5 口引脚原理功能框图中的模拟输入和输出有微小差别。

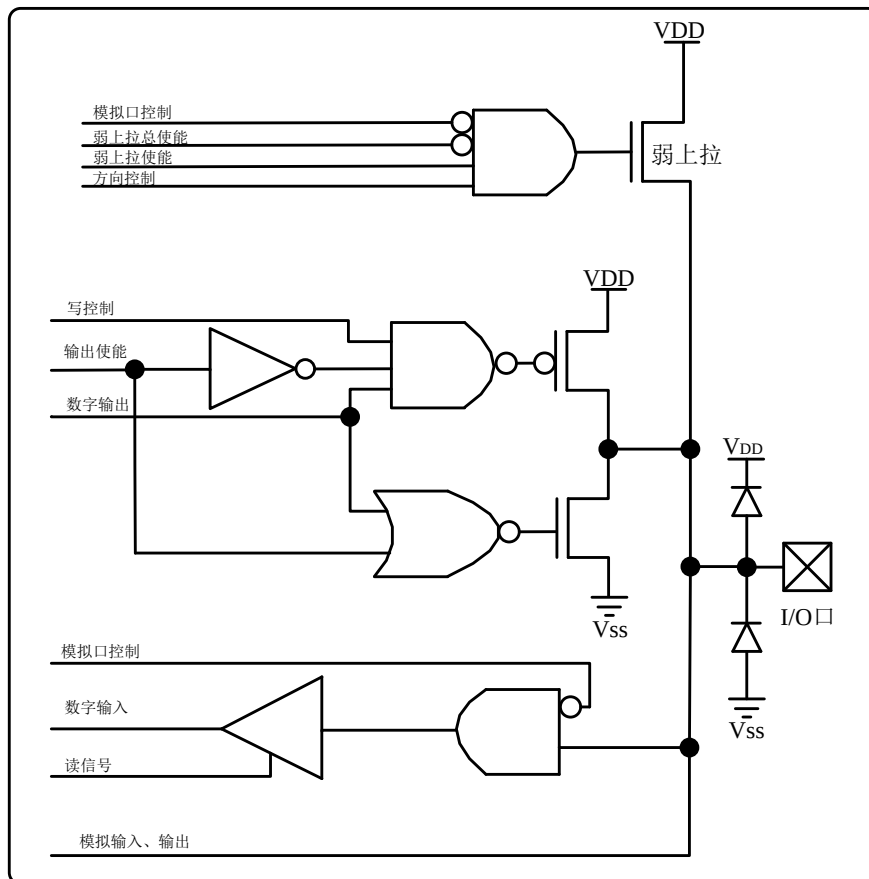


图 2.8 P5 口引脚原理功能框图

2.8 P6 口

P6 口可作为普通 I/O 口且带有上拉功能。

2.8.1 P6 口相关的寄存器

表 2-14 与 P6 口相关的寄存器

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
214H	P6	-	P66	P65	P64	P63	P62	P61	P60
215H	P6LR	-	P6LR6	P6LR5	P6LR4	P6LR3	P6LR2	P6LR1	P6LR0
216H	TR6	-	TR66	TR65	TR64	TR63	TR62	TR61	TR60
218H	PUR6	-	PUR66	PUR65	PUR64	PUR63	PUR62	PUR61	PUR60
208H	ANS6	-	ANS66	ANS65	ANS64	ANS63	ANS62	ANS61	ANS60

2.8.1.1 P6 口状态寄存器(P6)

寄存器 P6 各位对应端口 P6 口各引脚作为普通 I/O 口时的状态。

寄存器2.33: P6: P6口状态寄存器(地址: 214H)

复位值 -xxx xxxx	bit7							bit0	
	-	P66	P65	P64	P63	P62	P61	P60	
	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P64: P6 口各引脚状态位
1 = 对应引脚为逻辑高电平
0 = 对应引脚为逻辑低电平

P6<6:5><3:0>: 保留位

2.8.1.2 P6 口输出锁存寄存器 (P6LR)

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

寄存器2.34: P6LR: P6口输出锁存寄存器(地址:215H)

复位值 -xxx xxxx	bit7							bit0	
	-	P6LR6	P6LR5	P6LR4	P6LR3	P6LR2	P6LR1	P6LR0	
	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P6LR4: 写 P6 口输出状态
1 = 对应引脚输出高电平
0 = 对应引脚输出低电平

P6LR<6:5><3:0>: 保留位

2.8.1.3P6 口方向控制寄存器(TR6)

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

寄存器2.35: TR6: P6口方向控制寄存器(地址: 216H)

复位值 -111 1111	bit7						bit0	
	-	TR66	TR65	TR64	TR63	TR62	TR61	TR60
	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TR64: P6 口各引脚方向控制位
1 = P6 口对应引脚被配置为输入端口
0 = P6 口对应引脚被配置为输出端口

TR6<6:5><3:0>: 保留位

2.8.1.4P6 口上拉功能控制寄存器(PUR6)

KF8L15Z20XX 中 P6 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 P6 口上拉功能打开，然后再将要打开上拉功能的引脚，所对应的上拉功能控制位置 1 即可。寄存器 2.36 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2.36: PUR6: P6口弱上拉控制寄存器(地址:218H)

复位值 1111 1111	bit7						bit0	
	-	PUR66	PUR65	PUR64	PUR63	PUR62	PUR61	PUR60
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUR64: 上拉功能使能位
1 = 使能对应的端口上拉功能
0 = 禁止对应的端口上拉功能

PUR6<6:5><3:0>: 保留位

2.8.1.5P6 口模拟/数字口设置寄存器 (ANS6)

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

寄存器2.37: ANS6: P6口模拟/数字口设置寄存器(地址:208H)

复位值 1111 1111	bit7						bit0	
	-	ANS66	ANS65	ANS64	ANS63	ANS62	ANS61	ANS60
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ANS64: P6 口各引脚模拟/数字口设置位
1 = 将对应引脚配置为模拟口
0 = 将对应引脚配置为数字口或者特殊功能引脚

ANS6<6:5><3:0>: 保留位

2.8.2 P6 口原理功能框图

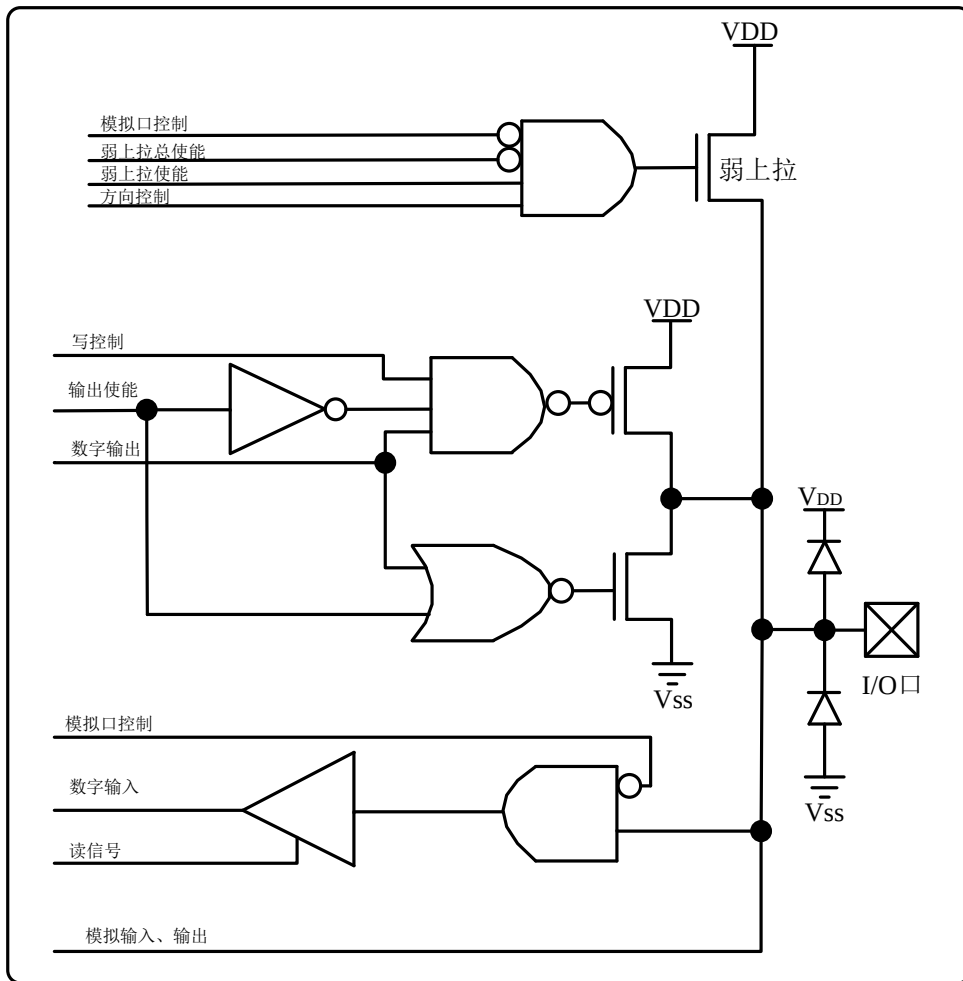


图 2.9 P6 口引脚原理功能框图

3 存储器

如图 3.1 所示, KF8L15Z20XX 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 20K 字节 (10K×16 位) 的 FLASH 存储器; 数据存储器由特殊功能寄存器和通用寄存器组成, 特殊功能寄存器空间为 512×8 位, 通用数据寄存器空间为(2048+16)×8 位。另外 KF8L15Z20XX 中还有一些其它存储器, 包括:寄存器组 R0~R7、16 级硬件堆栈、ID 地址单元等。

3.1 程序存储器(ROM)区

KF8L15Z20XX 有一个 14 位的程序计数器, 最大可寻址 16K×16 位的程序存储空间, 而在 KF8L15Z20XX 中只实现了 10K×16 位的程序存储空间, 地址为 0000H~27FFH, 复位向量入口地址为 0000H, 中断向量有两级入口地址, 高为 0004H, 低为 0014H。图 3.2 为程序存储器区的地址映射图。

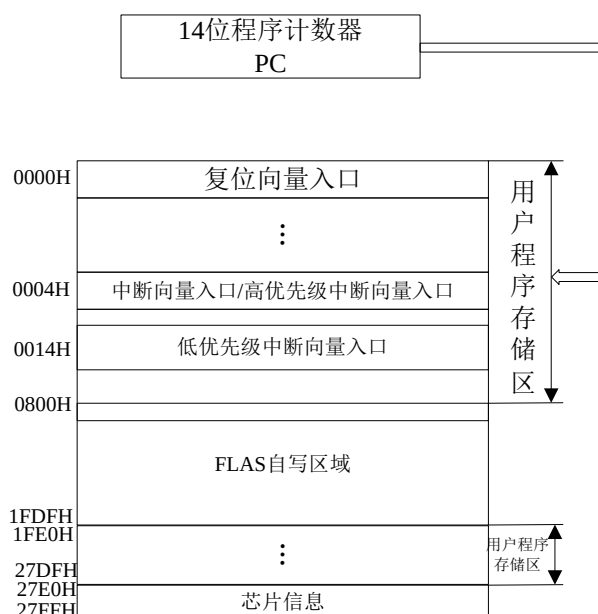


图 3.2 KF8L15Z20XX 程序存储器映射

3.1.1 程序计数器(PC)

KF8L15Z20XX 的程序计数器(PC)为 14 位宽。程序计数器(PC)的低 8 位(PC<7:0>)来自特殊功能寄存器 PCL, 高 6 位(PC<13:8>)来自 PCH 寄存器, PC 不能直接读写, PCH/PCL 寄存器可以读写。当发生复位时, PC 将被清零。在有任何未屏蔽中断发生后 PC 值将指向 0004H 或 0014H 地址。

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

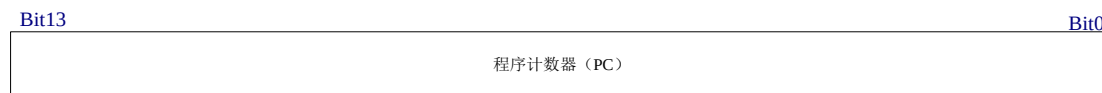


图 3.3 程序计数器 (PC)

3.1.1.1PCL/PCH 寄存器的使用

当需要改变程序计数器到固定地址（包括跳页）时，可以通过写 PCH 寄存器确定高 6 位地址（写 PCH 寄存器不会改变 PC 的值），当写低 8 位地址数据到 PCL 寄存器时，程序计数器的 14 位地址数据将更新，变为 PCH 寄存器和 PCL 寄存器的数据，如图 3.4 所示。

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

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

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

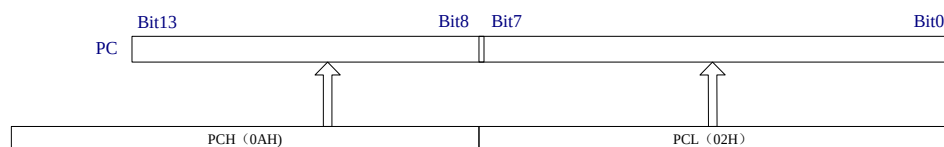


图 3.4 PCL/PCH 寄存器的使用

3.1.1.2执行 JMP、CALL 指令时的情况

KF8L15Z20XX 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_BIT5 和 PCH_BIT4 以及指令所带立即数 (#data12)，如图 3.5 所示。

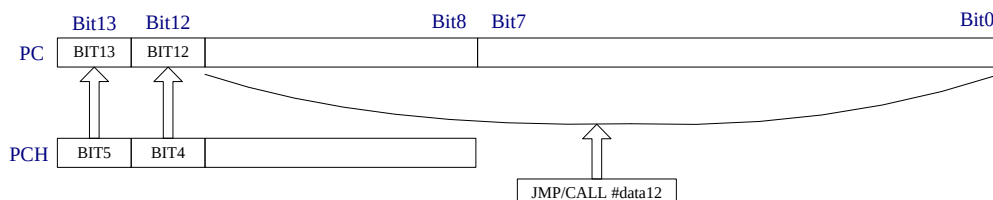


图 3.5 执行 JMP/CALL 指令时 PC 的变化图

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

3.1.2 堆栈

KF8L15Z20XX 单片机具有 16 级的硬件堆栈。堆栈空间单独编址，不占用任何程序存储区和数据存储区的空间，堆栈指针不可读写。当执行 CALL 指令或者中断导致程序跳转时，PC 值将被压入（PUSH）堆栈。在执行返回指令（IRET、RRET、CRET）时，堆栈中的断点地址将从堆栈中弹出（POP）到 PC 中。PCH 不受 PUSH 或者 POP 操作的影响。

KF8L15Z20XX 单片机具有 16 级的硬件堆栈，堆栈压栈第 17 次的数值将覆盖第 1 次压栈的所保存的值，堆栈压栈第 18 次的数值将覆盖第 2 次压栈的所保存的值，以此类推。

3.2 数据存储器(RAM)区

如图 3.6 所示，KF8L15Z20XX 中的数据存储器由 10 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 8 个存储器区为通用寄存器区，由用户支配。SFR 地址空间为 00H~6FH、100H~16FH、200H~26FH 和 300H~36FH；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。

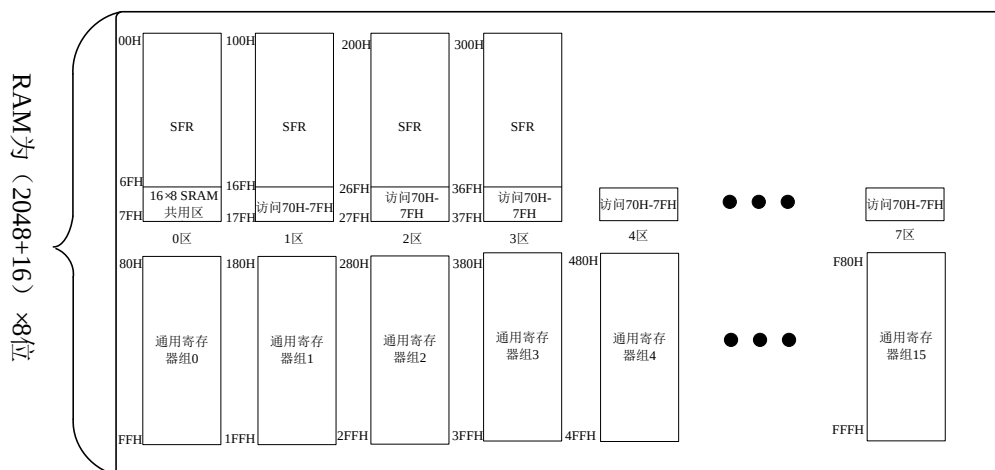


图 3.6 数据存储器地址映射图

3.2.1 通用寄存器区

如图 3.6 所示，通用寄存器的空间为 2048 字节，0 区至 15 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

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
0100	通用寄存器 4 区	480H~4FFH
0101	通用寄存器 5 区	580H~5FFH
0110	通用寄存器 6 区	680H~6FFH
0111	通用寄存器 7 区	780H~7FFH
1000	通用寄存器 8 区	880H~8FFH
1001	通用寄存器 9 区	980H~9FFH
1010	通用寄存器 10 区	A80H~AFFH
1011	通用寄存器 11 区	B80H~BFFH
1100	通用寄存器 12 区	C80H~CFFH
1101	通用寄存器 13 区	D80H~DFFH
1110	通用寄存器 14 区	E80H~EFFH
1111	通用寄存器 15 区	F80H~FFFH

切换通用寄存区的指令如例 3.1 所示：

例3.1 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

KF8L15Z20XX 内部的电源系统、I/O 口控制、定时/计数器、PWM、运放、中断等各种外设的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。

3.2.3 状态字寄存器（PSW）

状态字寄存器(PSW): 如寄存器 3.2 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)。TO 和 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)

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

$\overline{TO}$: 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

$\overline{PD}$: 上电复位标志位

1 = 上电复位或执行 CWDT 指令后

0 = 执行 IDLE 指令后被清 0

Z: 零状态标志位

1 = 算术运算或者逻辑运算的运行结果为 0

0 = 算术运算或者逻辑运算的运行结果不为 0

DC: 辅助进/借位标志位

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

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

CY: 进位/借位标志位

1 = 执行结果(8 位)向高位有进位时(加指令)或没有借位(减指令)

0 = 执行结果(8 位)向高位无进位时(加指令)或有借位(减指令)

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

3.3 FLASH 自写

KF8L15Z20XX 在程序存储区开辟了可自定义的自写区域，只有当目标地址位于未受写保护的存储器段内（由配置位的 SWRTEN<1: 0>位定义）时，才能对 Flash 进行写操作，该区域在正常工作期间是可读写的，它并没有直接映射到寄存器空间，而是通过特殊功能寄存器间接寻址。有 6 个特殊功能寄存器用于访问该区域。

表 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 6 位					
3BH	NVMADDRL	NVM 地址指针低 8 位寄存器							
3CH	NVMCTL0	NVM 控制寄存器 0							
3DH	NVMCTL1	NVM 控制寄存器 1							
38H	NVMDATAH	NVM 数据高 8 位寄存器							
39H	NVMDATAL	NVM 数据低 8 位寄存器							

如图 3.7 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。

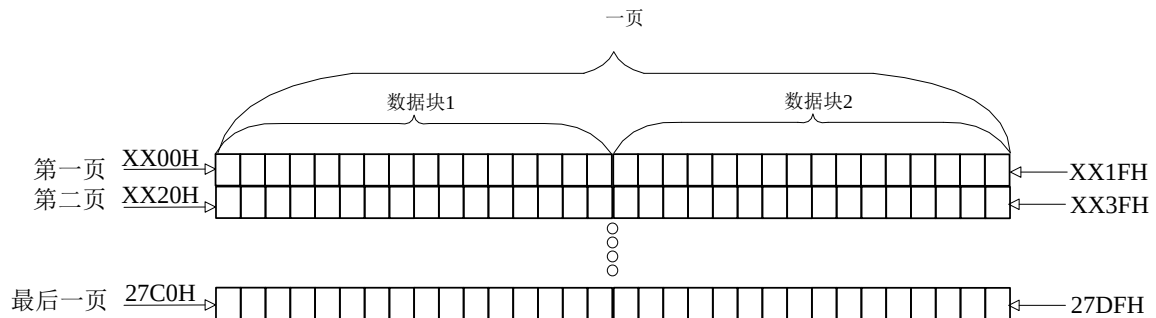


图 3.7 Flash 自写区域地址映射图

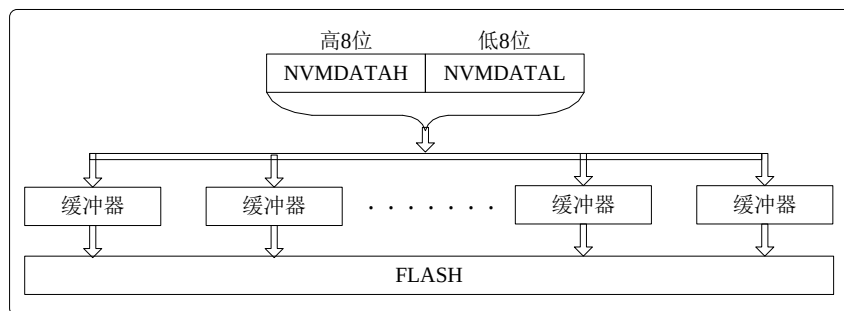


图 3.8 写操作

如图 3.8 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，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。

3.3.2 寄存器 NVMADDRH/L

如寄存器 3.2 和 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 14 位的地址信息，NVMADDRH 存放地址的高 6 位，NVMADDRL 存放地址的低 8 位。

寄存器3.2: NVMADDRH: 数据指针高6位(地址: 3AH)

复位值 ---0 0000	-	-	bit5	bit4	bit3	bit2	bit1	bit0
	U	U	R/W	R/W	R/W	R/W	R/W	R/W

寄存器3.3: NVMADDRL: 数据指针低8位(地址: 3BH)

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

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

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

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

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后才会擦除本页的数据，对其它块写操作不会产生擦除本页数据的命令，可能导致写入数据出错。

注：1.写Flash时，从Flash自写首地址开始处，连续的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命令，且擦除命令只有对每页的第一个数据块写完后才会自动执行，将本页所有单元数据擦除。而对每页的其他数据块写操作时不会有擦除命令发出。

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

4.需正确配置SWRTEN<1:0>，才能对Flash目标地址区域进行写操作。

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

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

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

写 FLASH 的步骤为：

1. 将要写入的数据送到 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：

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

上面指令中的立即数 0X81 是固定不变的。此时该地址的数据高 8 位被送 NVMDATAH，低 8 位送到 NVMDATAL。无论配置位 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. 两个指令周期后该单元的数据被送到 NVMDATAH/L。

3.4 DATA EEPROM

KF8L15Z20XX 片内的 DATA EEPROM 存储器最大容量为 256×8 位，地址范围 00H~FFH，在 CPU 正常工作期间是可读写的。DATA EEPROM 是单独编址，可以通过特殊功能寄存器寻址。DATA EEPROM 与 Flash 自写共用 4 组寄存器。

表 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 位寄存器							
3CH	NVMCTL0	NVM 控制寄存器 0							
3DH	NVMCTL1	NVM 控制寄存器 1							
39H	NVMDATAL	NVM 数据低 8 位寄存器							

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 256×8 位，地址范围 0~255。NVMADDRL 寄存器用来存放要写入 DATA EEPROM 的 8 位的地址信息。

寄存器 3.4 NVMADDRL: 数据指针低 8 位(地址: 3BH)

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

3.4.1 寄存器 NVMCTL0/NVMCTL1

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

3.4.2 写 DATA EEPROM

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

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

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

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

注：

- 1.CPU写DATA EEPROM时,不管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。DATA EEPROM 需要 6ms 来执行擦除和写入数据的命令，此期间 CPU 仍可正常工作，用户可通过中断标志位 EEIF 判断写 DEE 是否完成，期间指令可正常执行；
4. 重复执行步骤 1、2、3，可以执行其他地址的写入。

3.4.3 读 DATA EEPROM

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

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

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

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

注：读 DATA EEPROM 时，不管 $\overline{\text{DATAP}}$ 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

KF8L15Z20XX 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

KF8L15Z20XX 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 27E0H~27FFH。用户可在其中存放校准值或其它信息。

4 汇编指令及寻址方式

4.1 寻址方式

KF8L15Z20XX 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。KF8L15Z20XX 的指令可以没有操作数、一个操作数、两个操作数。

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 1，3BH 即 NVMADDRL。

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

KF8L15Z20XX 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

KF8L15Z20XX 单片机的中断源有:

- INT1
- T0 溢出中断
- T1/2/3/4 中断
- P0/P3 口引脚电平变化中断
- AD 中断
- PWM1x 中断
- CMP1/3/4 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(EF)中断
- 高/低压检测(HLVD)中断

在本单片机中有多个中断优先级, 其中高优先级向量位于 0X0004H, 低优先级向量位于 0X0014H。在中断服务程序里可通过 PCTL 的 IPEN(PCTL<3>)进行中断优先级设置。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。

KF8L15Z20XX 中的 INT1/2、T1、T2、T3、T4、ADC、PWM1x、CMP、HLVD、SPI、I2C、BCL、USART 和 EE 等都属于外设, 因此对应的中断称作外设中断, 其它中断源产生的中断属于内部中断。中断逻辑如图 5.2 所示。在中断逻辑框图中, 每个中断源有 3 个位用于控制其操作。这些位的功能分别是:

☞ 标志位表明发生了中断事件;

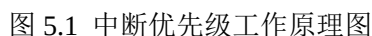
☞ 中断允许位允许程序跳转到中断向量地址处执行;

☞ 中断优先级位用于选择高优先级及还是低优先级;

通过将 IPEN 位 (PCTL<3>) 置 1, 可使能中断优先级功能。

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

其原理框图如图 5.1 下:



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



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	-	P0IE	T0IF	-	P0IF
2CH	EIE1	EEIE	ADIE	-	INT1IE	C1IE	PWMIE	T2IE	T1IE
2DH	EIE2	T3IE	-	RC1IE	TX1IE	C3IE	-	BCLIE	SSCIE
4AH	EIE3	T4IE	C4IE	OSCFAI LIE	-	-	-	P3IE	HLVDIE
251H	EIE4	-	-	-	-	RE2IE	TE2IE	RC2IE	TX2IE
329H	EIE5	ALRIE	RTCT1I E	RTCT0I E	TTIE	DAYIE	HURIE	MINIE	SECIE
0CH	EIF1	EEIF	ADIF	-	INT1IF	C1IF	PWMIF	T2IF	T1IF
0DH	EIF2	T3IF	-	RC1IF	TX1IF	C3IF	-	BCLIF	SSCIF
4BH	EIF3	T4IF	C4IF	OSCFAI LIF	-	-	-	P3IF	HLVDIF
252H	EIF4	-	-	-	-	RE2IF	TE2IF	RC2IF	TX2IF
32AH	EIF5	ALRIF	RTCT1I F	RTCT0I F	TTIF	DAYIF	HURIF	MINIF	SECIF
22H	IP0	-	-	-	-	-	PT0	-	PP0
23H	IP1	PEE	PADC	-	PINT1	PC1	PPWM	PT2	PT1
24H	IP2	PT3	-	PRC1	PTX1	PC3	-	PBCL	PSSCI
29H	IP3	PT4	PC4	POSCF AIL	-	-	-	PP3	PHLVD
253H	IP4	-	-	-	-	PRE2	PTE2	PRC2	PTX2
32BH	IP5	PALR	PRTCT1	PRTCT0	PTT	PDAY	PHUR	PMIN	PSEC
2EH	PCTL	-	-	-	SLVRE N	IPEN	SWDTE N	POR	LVR
67H	INTED GCTL	-	INT1SE	-	-	-	-	-	T1CLK SE

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 5.1 所示。

在优先级中断中，AIEH 为全局优先级中断使能位，当其被清 0 时，禁止所有中断。AIEL 为低优先级中断使能位，当其被清 0 时禁止所有低优先级中断。具体的中断逻辑如图 5.1 所示。

注：

1. 当中断条件满足时，无论相应的中断使能位或者全局中断使能位AIE的状态如何，中断标志位将被硬件置1。
2. 中断条件满足时，中断标志位通过硬件置1，而清零则需要软件完成。
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时才可读，PUIE只有在IPEN=0时才可读；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 0000 0000	bit7						bit0	
	AIE/AIEH	PUIE/AIEL	TOIE	-	POIE	TOIF	-	POIF
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

AIE/AIEH: 全局中断使能位/优先级中断使能位

当 IPEN=0

1=使能所有未屏蔽的中断

0=禁止所有中断

当 IPEN=1

1=允许所有高优先级的中断

0=禁止所有中断

PUIE/AIEL: 外设中断使能位/低优先级中断使能位

当 IPEN=0

1=使能所有未屏蔽的外设中断

0=禁止所有外设中断

当 IPEN=1

1=允许所有低优先级的外设中断

0=禁止所有低优先级的外设中断

TOIE: T0 溢出中断使能位

1=使能 T0 中断

0=禁止 T0 中断

POIE: P0 口电平变化中断使能位

1=使能 P0 口电平变化中断

0=禁止 P0 口电平变化中断

TOIF: T0 溢出中断标志位

1=T0 寄存器溢出

0=T0 寄存器未溢出

POIF: 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

5.1.2 中断使能寄存器 EIE1

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	-	INT1IE	C1IE	PWMIE	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 = 使能独立 EE 中断
0 = 禁止独立 EE 中断
- ADIE:** AD 中断使能位
1 = 使能 AD 中断
0 = 禁止 AD 中断
- INT1IE:** INT1 中断使能位
1 = 使能 INT1 中断
0 = 禁止 INT1 中断
- C1IE:** 模拟比较器 1 中断使能位
1 = 使能 CMP1 中断
0 = 禁止 CMP1 中断
- PWMIE:** PWM 中断使能位
1 = 使能 PWM 中断
0 = 禁止 PWM 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

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	-	RC1IE	TX1IE	C3IE	-	BCLIE	SSCIE
	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
- RC1IE:** USART1 接收中断使能位
1 = 允许 RX1 中断
0 = 禁止 RX1 中断
- TX1IE:** USART1 发送中断使能位
1 = 允许 TX1 中断
0 = 禁止 TX1 中断

- C3IE: 模拟比较器 3 中断使能位
1 = 使能 CMP3 中断
0 = 禁止 CMP3 中断
- BCLIE: BCL 中断使能位
1 = 允许 BCL 中断
0 = 禁止 BCL 中断
- SSCIE: SSCI 中断使能位
1 = 允许 SSCI 中断
0 = 禁止 SSCI 中断

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 0000 0000	bit7						bit0	
	T4IE	C4IE	OSCFAILIE	-	-	-	P3IE	HLVDIE
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- T4IE: T4 中断使能位
1 = 允许 T4 中断
0 = 禁止 T4 中断
- C4IE: 模拟比较器 4 中断使能位
1 = 使能 CMP4 中断
0 = 禁止 CMP4 中断
- 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
- P3IE: P3 口电平变化中断使能位
1 = 允许 P3 口电平变化中断
0 = 禁止 P3 口电平变化中断
- HLVDIE: HLVDIE 中断使能位
1 = 允许 HLVDIE 中断
0 = 禁止 HLVDIE 中断

5.1.5 中断使能寄存器 EIE4

寄存器5.4: EIE4: 中断使能寄存器4(地址: 251H)

复位值 ---- 0000	bit7				bit0			
	-	-	-	-	RE2IE	TE2IE	RC2IE	TX2IE
	U	U	U	U	R/W	R/W	R/W	R/W

- RE2IE: UART2 接收错误中断使能位
1 = 使能 UART2 接收错误中断
0 = 禁止 UART2 接收错误中断
- TE2IE: UART2 发送错误中断使能位

1 = 使能 UART2 发送错误中断
 0 = 禁止 UART2 发送错误中断

RC2IE: USART2 接收中断使能位
 1 = 允许 RX2 中断
 0 = 禁止 RX2 中断

TX2IE: USART2 发送中断使能位
 1 = 允许 TX2 中断
 0 = 禁止 TX2 中断

5.1.6 中断使能寄存器 EIE5

寄存器5.4: EIE5: 中断使能寄存器5(地址:329H)

复位值 0000 0000	bit7						bit0	
	ALRIE	RTCT1IE	RTCT0IE	TTIE	DAYIE	HURIE	MINIE	SECIE
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ALRIE: RTC 闹钟中断
 1 = 使能 RTC 闹钟中断
 0 = 禁止 RTC 闹钟中断

RTCT1IE: RTC 内置定时器 1 溢出中断
 1 = 使能 RTC 内置定时器 1 溢出中断
 0 = 禁止 RTC 内置定时器 1 溢出中断

RTCT0IE: RTC 内置定时器 0 溢出中断
 1 = 使能 RTC 内置定时器 0 溢出中断
 0 = 禁止 RTC 内置定时器 0 溢出中断

TTIE: RTC 时间节拍中断
 1 = 使能 RTC 时间节拍中断
 0 = 禁止 RTC 时间节拍中断

DAYIE: 进程日中断
 1 = 使能 RTC 进程日中断
 0 = 禁止 RTC 进程日中断

HURIE: 进程时中断
 1 = 使能 RTC 进程时中断
 0 = 禁止 RTC 进程时中断

MINIE: 进程分中断
 1 = 使能 RTC 进程分中断
 0 = 禁止 RTC 进程分中断

SECIE: 进程秒中断
 1 = 使能 RTC 进程秒中断
 0 = 禁止 RTC 进程秒中断

5.1.7 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)

复位值 0000 0000	bit7						bit0	
	EEIF	ADIF	-	INT1IF	C1IF	PWMIF	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 完成写操作
0 = 独立 EE 未完成写操作
- ADIF:** AD 完成中断标志位
1 = AD 转换完成
0 = AD 转换没有完成
- INT1IF:** INT1 中断标志位
1 = INT1 产生外部中断
0 = INT1 未产生外部中断
- C1IF:** 模拟比较器 1 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
- PWMIF:** PWM 中断标志位
1 = 发生了 T1H/T1L 与 PP2/PP1 匹配
0 = 未发生 T1H/T1L 与 PP2/PP1 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

5.1.8 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	-	RC1IF	TX1IF	C3IF	-	BCLIF	SSCIF
	R/W	R/W	R	R	R/W	R/W	R/W	R/W

- T3IF:** T3 中断标志位
1 = T3 产生了中断
0 = T3 未产生中断
- RC1IF:** USART1 接收中断标志位
1 = USART1 接收缓冲器满 (通过 RXSDR1 清 0)
0 = USART1 接收缓冲器空
- TX1IF:** USART1 发送中断标志位
0 = USART1 发送缓冲器满
1 = USART1 发送缓冲器空 (通过 TXSDR1 清 0)

- C3IF:** 模拟比较器 CMP3 中断标志位
 1 = 模拟比较器 CMP3 输出发生改变(必须软件清 0)
 0 = 模拟比较器 CMP3 输出未发生改变
- 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

5.1.9 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 0000 0000	bit7						bit0	
	T4IF	C4IF	OSCFAILIF	-	-	-	P3IF	HLVDIF
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- T4IF:** T4 中断标志位
 1 = T4 溢出或发生重载
 0 = T4 不溢出或发生重载
- C4IF:** 模拟比较器 CMP4 中断标志位
 1 = 模拟比较器 CMP4 输出发生改变(必须软件清 0)
 0 = 模拟比较器 CMP4 输出未发生改变
- 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
- P3IF:** P3 口电平变化中断标志位
 1=引脚 P3.0~P3.7 至少有一个电平状态发生变化
 0=引脚 P3.0~P3.7 电平状态未发生变化
- HLVDIF:** HLVD 中断标志位
 1 = 产生了 HLVD 中断
 0 = 未产生 HLVD 中断

5.1.10 中断标志寄存器 EIF4

寄存器5.8:EIF4: 中断使能寄存器4(地址: 252H)

复位值 ---- 0000	bit7						bit0	
	-	-	-	-	RE2IF	TE2IF	RC2IF	TX2IF
	U	U	U	U	R/W	R/W	R	R

- RE2IF:** UART2 接收错误中断标志位
 1 = 发生 UART2 接收错误中断
 0 = 未发生 UART2 接收错误中断
- TE2IF:** UART2 发送错误中断标志位
 1 = 发生 UART2 发送错误中断

- 0 = 未发生 UART2 发送错误中断
- RC2IF: USART2 接收中断标志位
1 = USART2 接收缓冲器满 (通过 RXSDR2 清 0)
- 0 = USART1 接收缓冲器空
- TX2IF: USART2 发送中断标志位
0 = USART2 发送缓冲器满
1 = USART2 发送缓冲器空 (通过 TXSDR2 清 0)

5.1.11 中断标志寄存器 EIF5

寄存器5.4:EIF5: 中断标志寄存器5(地址:32AH)

复位值 0000 0000	bit7						bit0	
	ALRIF	RTCT1IF	RTCT0IF	TTIF	DAYIF	HURIF	MINIF	SECIF
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- ALRIF: RTC 闹钟中断标志位
1 = RTC 发生闹钟事件
0 = RTC 未发生闹钟事件
- RTCT1IF: RTC 内置定时器 1 溢出中断标志位
1 = RTC 内置定时器 1 溢出
0 = RTC 内置定时器 1 未溢出
- RTCT0IF: RTC 内置定时器 0 溢出中断标志位
1 = RTC 内置定时器 0 溢出
0 = RTC 内置定时器 0 未溢出
- TTIF: RTC 时间节拍中断标志位
1 = RTC 产生时间节拍中断
0 = RTC 未产生时间节拍中断
- DAYIF: 进程日中断标志位
1 = RTC 产生进程日中断
0 = RTC 未产生进程日中断
- HURIF: 进程时中断标志位
1 = RTC 产生进程时中断
0 = RTC 未产生进程时中断
- MINIF: 进程分中断标志位
1 = RTC 产生进程分中断
0 = RTC 未产生进程分中断
- SECIF: 进程秒中断标志位
1 = RTC 产生进程秒中断
0 = RTC 未产生进程秒中断

5.1.12 中断优先级控制寄存器 IP0

寄存器5.9:IP0: 中断优先级控制寄存器0(地址: 22H)

复位值 ---- -000	bit7					bit0	
	-	-	-	-	-	PT0	PP0
	U	U	U	U	U	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

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

5.1.13 中断优先级控制寄存器 IP1

寄存器5.10:IP1: 中断优先级控制寄存器1(地址: 23H)

复位值 0000 0000	bit7						bit0
	PEE	PADC	-	PINT1	PC1	PPWM	PT1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

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

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

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

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

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

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

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

5.1.14 中断优先级控制寄存器 IP2

寄存器5.11: IP2: 中断优先级控制寄存器2(地址: 24H)

复位值 0000 0000	bit7						bit0	
	PT3	-	PRC1	PTX1	PC3	-	PBCL	PSSCI
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PT3: T3 中断高优先级控制位

1 = T3 中断为高优先级

0 = T3 中断为低优先级

PRC1: RX1 中断高优先级控制位

1 = RX1 中断为高优先级

0 = RX1 中断为低优先级

PTX1: TX1 中断高优先级控制位

1 = TX1 中断为高优先级

0 = TX1 中断为低优先级

PC3: 模拟比较器 3 中断高优先级控制位

1 = 比较器 CMP3 中断为高优先级

0 = 比较器 CMP3 中断为低优先级

PBCL: BCL 中断高优先级控制位

1 = BCL 中断为高优先级

0 = BCL 中断为低优先级

PSSCI: SSCI 中断高优先级控制位

1 = SSCI 中断为高优先级

0 = SSCI 中断为低优先级

5.1.15 中断优先级控制寄存器 IP3

寄存器5.12: IP3: 中断优先级控制寄存器3(地址: 29H)

bit7							bit0	
复位值 0000 0000	PT4	PC4	POSCFAI L	-	-	-	PP3	PHLVD
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PT4: T4 中断高优先级控制位

1 = T4 中断为高优先级

0 = T4 中断为低优先级

PC4: 模拟比较器 4 中断高优先级控制位

1 = 比较器 CMP4 中断为高优先级

0 = 比较器 CMP4 中断为低优先级

POSCFAIL: OSCFAIL 中断高优先级控制位

1 = OSCFAIL 中断为高优先级

0 = OSCFAIL 中断为低优先级

PP3: P3 口电平变化中断高优先级控制位

1 = P3 电平变化中断为高优先级

0 = P3 电平变化中断为低优先级
 PHLVD: PHLVD 中断高优先级控制位
 1 = PHLVD 中断为高优先级
 0 = PHLVD 中断为低优先级

5.1.16 中断优先级控制寄存器 IP4

寄存器5.13: IP4: 中断优先级控制寄存器4(地址: 253H)

	bit7							bit0
复位值 ---- 0000	-	-	-	-	PRE2	PTE2	PRC2	PTX2
	U	U	U	U	R/W	R/W	R/W	R/W

PRE2: UART2 接收错误中断优先级控制位
 1 = UART2 接收错误中断为高优先级
 0 = UART2 接收错误中断为低优先级
 PTE2: UART2 发送错误中断优先级控制位
 1 = UART2 发送错误中断为高优先级
 0 = UART2 发送错误中断为低优先级
 PRC2: RX2 中断高优先级控制位
 1 = RX2 中断为高优先级
 0 = RX2 中断为低优先级
 PTX2: TX2 中断高优先级控制位
 1 = TX2 中断为高优先级
 0 = TX2 中断为低优先级

5.1.17 中断优先级控制寄存器 IP5

寄存器5.4: IP5: 中断标志寄存器5(地址:32BH)

	bit7							bit0
复位值 0000 0000	PALR	PRTCT1	PRTCT0	PTT	PDAY	PHUR	PMIN	PSEC
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PALR: RTC 闹钟中断高优先级控制位
 1 = RTC 闹钟中断为高优先级
 0 = RTC 闹钟中断为低优先级
 PRTCT1: RTC 内置定时器 1 溢出中断高优先级控制位
 1 = RTC 内置定时器 1 溢出中断为高优先级
 0 = RTC 内置定时器 1 溢出中断为低优先级
 PRTCT0: RTC 内置定时器 0 溢出中断高优先级控制位
 1 = RTC 内置定时器 0 溢出中断为高优先级
 0 = RTC 内置定时器 0 溢出中断为低优先级
 PTT: RTC 时间节拍中断高优先级控制位
 1 = RTC 时间节拍中断为高优先级
 0 = RTC 时间节拍中断为低优先级

- PDAY: RTC 进程日中断高低优先级控制位
 1 = RTC 进程日中断为高优先级
 0 = RTC 进程日中断为低优先级
- PHUR: RTC 进程时中断高低优先级控制位
 1 = RTC 进程时中断为高优先级
 0 = RTC 进程时中断为低优先级
- PMIN: RTC 进程分中断高低优先级控制位
 1 = RTC 进程分中断为高优先级
 0 = RTC 进程分中断为低优先级
- PSEC: RTC 进程秒中断高低优先级控制位
 1 = RTC 进程秒中断为高优先级
 0 = RTC 进程秒中断为低优先级

5.1.18 电源控制寄存器 PCTL

PCTL 寄存器的 $\overline{\text{LVR}}$ 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5.14:PCTL: 电源控制寄存器(地址:2EH)

复位值 --01 00xx	bit7			SLVREN	IPEN	SWDTEN	$\overline{\text{POR}}$	bit0 $\overline{\text{LVR}}$
	-	-	-					
	U	U	R/W	R/W	R/W	R/W	R/W	R/W

- SLVREN: 软件欠压检测使能位
 1 = 使能欠压检测
 0 = 禁止欠压检测
- IPEN: 中断优先级控制位
 1 = 使能中断优先级功能
 0 = 禁止中断优先级，即为普通模式
- SWDTEN: 软件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
- $\overline{\text{LVR}}$: 欠压复位状态位
 1 = 未发生欠压复位
 0 = 已发生欠压复位

5.1.19 INT 中断沿选择寄存器 INTEDGCTL

寄存器5.15:INTEDGCTL:INT中断沿选择寄存器 (67H)

复位值	bit7						bit0
00---1	-	INT1SE	-	-	-	-	T1CLKSE
	R/W	R/W	U	U	U	U	R/W

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

T1CLKSE: T1 定时器时钟选择位
1 = 内部高频时钟
0 = 系统时钟 SCLK/4

5.1.20 中断响应

当 IPEN=1 时，中断被响应后：

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

当 IPEN=0 时，中断被响应后：

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

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

注：

1. 中断条件满足时，无论相应的中断使能位或AIE位的状态为何，中断标志位都将被置1。
2. 当执行一条清除AIE位的指令后，任何在下一周期等待响应的中断都将被忽略。当AIE位重新置1时，被忽略的中断请求将继续等待被响应。
3. 当对中断进行响应，进入中断服务子程序的时候硬件会将AIE位清零关闭总中断，当中断程序执行完，中断返回指令跳出中断子程序时，硬件将AIE位置1打开总中断。

5.2 INT 中断

INT 中断有 1 个中断源: INT1, 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 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 位设置触发边沿, INT1SE 置 1, 将 INT1 设置为上升沿触发, 清零设置为下降沿触发。EIF1 中的 INT1IF 为 INT1 的中断标志位。如果 IPEN 和 PINT1 位均置 1, 则 INT1 为高优先级中断。

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

使用 INT 中断时的设置:

1. 将对应的 INT1 引脚设置为数字输入口。
2. 选择触发脉冲边沿时上升沿还是下降沿(INT1SE 置 1 为上升沿触发);
3. 将相应的外部中断使能位置 1(INT1IE), 如果为高优先级, 则 IPEN 和 PINT1 均置 1。

5.3 定时器中断

定时/计数器产生中断的条件如下:

T0: 计数溢出;

T1: 计数溢出或者与 PP2/PP1 寄存器匹配;

T2: 计数值与 PP5H/PP5L 寄存器匹配;

T3: 计数溢出或者与 T3REH/T3REL 寄存器匹配;

T4: 计数溢出或者与 T4REH/T4REL 寄存器匹配;

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/P3 口中断

P0/P3 口引脚的输入电平变化将使 P0IF/P3IF 位置 1。通过设置/清除 P0IE/P3IE 位, 可使能/禁止该中断。且该端口各引脚可通过 IOCL0/IOCL3 寄存器来对每个引脚进行配置。当 IPEN 和 PP0/PP3 均置 1 时, P0/P3 口中断可以配置为高优先级中断。

有关 P0/P3 口的操作, 请参考 P0/P3 口部分。

5.5 PWM 中断

使能 PWM1 后, T1L/H 分配给 PWM1 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。如果使能 T1IE, 则会触发中断 (AIE、PUIE 置 1)。当 IPEN 和 PT1 均置 1 时, PWM 中断可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 C1CTL1 的 C1MOD[1:0]不为 00 时，模拟比较器 CMP1 将使能，开始工作。（描述有误，可删）

极性 $INV_x=0$ ，比较器正常输出： $CxIN+>CxIN-=1$ ， $CxIN+<CxIN-=0$ ； $INV_x=1$ ，比较器输出极性转换： $CxIN+>CxIN-=0$ ， $CxIN+<CxIN-=1$ 。

设置寄存器 CMCTL0 的 CxEDG 位，用于 CxOUT 的上升沿和下降沿判断位。 $CxEDG=0$ ，上升沿中断； $CxEDG=1$ ，下降沿中断。当 IPEN 和 PCx 均置 1 时，模拟比较器中断可以配置为高优先级中断。其中 $x=1、3、4$ 。

详见模拟比较器部分。

5.7 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。

5.8 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

KF8L15Z20XX 单片机提供一个 8 位的定时器 T0、1 个 16 位的定时/计数器 T1、1 个 16 位定时器 T2 和一个 16 位的定时/计数器 T3 和一个 16 位的定时/计数器 T4。

6.1 定时器 T0

T0 是一个 8 位的定时器，当 T0 寄存器值加到 255 时，再加 1，则会产生溢出，T0 寄存器的值返回到 0 开始重新计数。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，则在接下来的两个机器周期将不执行递增操作，用户可通过将校正值写入 T0 寄存器进行修正。

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。当预分频器用于 T0 模块时，所有写入 T0 寄存器的指令都会将预分频器清 0。当预分频器用于 WDT 时，CWDT 指令会同时将预分频器和看门狗定时器清 0。

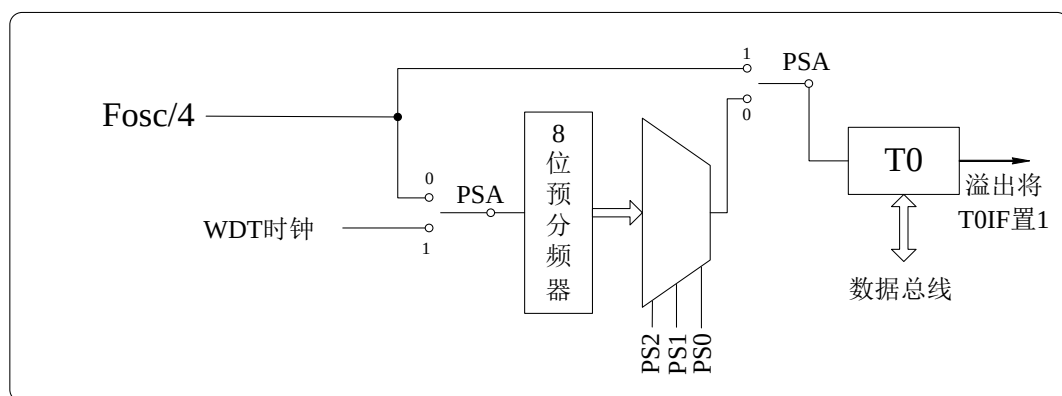


图 6.1 原理框图

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	8 位计数器							
21H	OPTR	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	bit7						bit0	
1111 1111	$\overline{\text{PUPH}}$	INT0SE	T0CS	T0SE	PSA	PS2	PS1	PS0
	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1

$\overline{\text{PUPH}}$ I/O 口上拉功能总使能位

1 = 禁止所有 I/O 口的上拉功能

0 = 允许 I/O 口使用上拉功能

INT0SE 保留位

T0CS: 保留位, 请勿置 1

T0SE: 保留位

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

6.1.3T0 的使用

T0 在使用时通过以下步骤进行设置:

1. 如果需要分频, 则将预分频器分配给 T0, 并设置分频比;
2. 给 T0 寄存器设置初始值;
3. 如果使用中断方式则将 T0IE 和 AIE 位置 1。

6.2 定时/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 中断标志位置 1。T1 属于外部单元, 因此在使用 T1 中断时, 需将 PUIE 位置 1, 使能外设中断。如图 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 $\overline{T1SY}$ 位(T1CTL<2>)设置可使 T1 工作在异步计数器模式。

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

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。

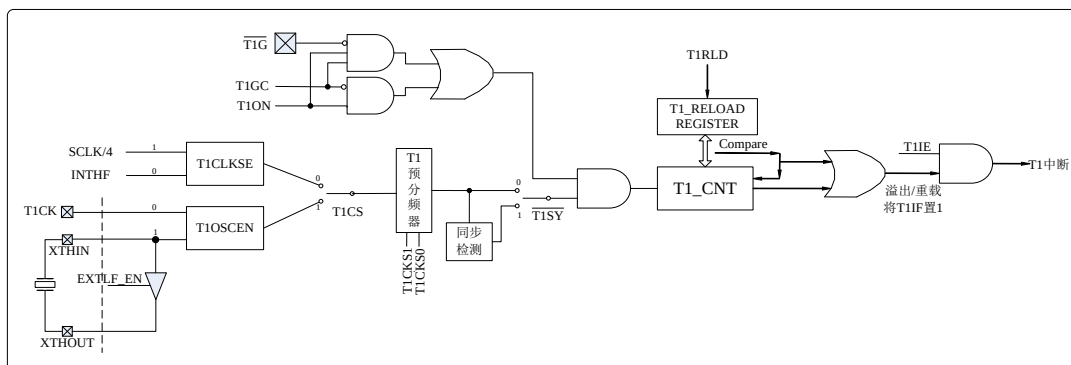


图 6.2 T1 原理框图

6.2.2 T1 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 T1 低字节寄存器							
0FH	T1H	定时/计数器 T1 高字节寄存器							
10H	T1CTL	T1RLD	T1GC	T1CKS1	T1CKS0	T1OSCEN	$\overline{T1SY}$	T1CS	T1ON
67H	INTEDGCTL	-	INT1SE	-	-	-	-	-	T1CLKSE

6.2.2.1 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	$\overline{T1SY}$	T1CS	T1ON
	R/W	R/W	R/W	R/W	U	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 $\overline{T1G}$ 引脚控制: $\overline{T1G}$ 引脚为低电平, 启动 T1, 为高电平, 关闭 T1

0 = 禁止 $\overline{T1G}$ 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 = 禁止外部低频时钟源作为 T1 计数时钟

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

T1CS=1:

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: 该位被忽略, T1 使用内部时钟

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

当 T1CLKSE=1 时 T1 时钟为内部高频时钟

当 T1CLKSE=0 时 T1 时钟为指令时钟 (系统时钟四分频)

T1ON: T1 启动控制位

1 = 启动 T1

0 = 停止 T1

6.2.2.2 INT 中断沿选择寄存器

寄存器6.4: INTEDGCTL:INT中断沿选择寄存器 (67H)

复位值 00---1	bit7						bit0	
	INT2SE	INT1SE	-	-	-	-	T1CLKSE	-
	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

T1CLKSE: T1 定时器时钟选择位
1 = 内部高频时钟
0 = 系统时钟 SCLK/4

6.2.3 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，对单片机内部时钟进行计数，当不使用预分频器时，每个机器周期 T1 寄存器自加 1，加到 0FFFFH 后再加 1，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。

6.2.4 计数模式

通过将 T1CS 位置 1 将 T1 设定为计数模式，在计数模式时，T1 在计数脉冲 T1CK 的上升沿进行递增计数。T1 的计数模式又有同步计数和异步计数两种方式，

如果控制位 $\overline{\text{T1SY}}$ (T1CTL.2)置 1，则 T1 工作在异步计数模式。计数器根据 T1CK 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。

6.2.5 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。

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

6.2.6 T1 在休眠模式下的运行

只有设定在异步计数器模式时，T1 才能在休眠模式下工作。在该模式下，计数脉冲

T1CK 使计数器递增。通过如下步骤设定定时器以唤醒器件:

- 使能 T1(T1ON/T1CTL.0 置 1)
- 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。

6.2.7 T1 分配给 PWM1

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

6.2.8 T1 计数寄存器被比较器清零

T1 计数寄存器 T1H/T1L 可被比较器的输出变化清零, 详见 11.7 比较器清零定时器。

6.3 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 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。

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），每个周期（当分频器 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。

内部高频振荡器时钟（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 T2CCRxH/L 相比较，如果等于 T2CCRxH/L，且对应的控制位 T2CCRxON 置 1，则会发出信号，使 ADCATL0 的 START（ADCCTL0<1>）位置 1。

分频器 2 的分频比可通过寄存器 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、看门狗复位或欠压复位），分频器 1/2 的计数器将被清 0。

寄存器 T2H/L、PP5 和 T2CCRxH/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，PP5 被设置为 FFH。

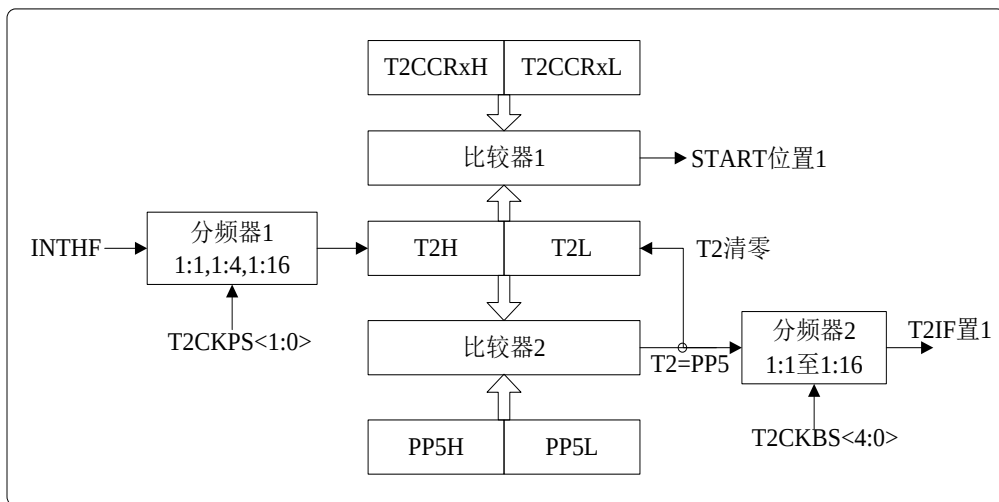


图 6.3 定时器 T2 原理框图

注: 需要在 T2ON 位置 1 的前提下对 T2 计数寄存器 T2H/T2L 进行赋值操作，否则无法写入。

6.3.2T2 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	T2 高字节寄存器							
11H	T2L	T2 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 触发 ADC 启动设置寄存器 0 低 8 位							
40H	T2CCR0H	T2 触发 ADC 启动设置寄存器 0 高 8 位							
116H	T2CCR1L	T2 触发 ADC 启动设置寄存器 1 低 8 位							
118H	T2CCR1H	T2 触发 ADC 启动设置寄存器 1 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1(地址: 12H)

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

T2CKBS<3:0>: T2 分频器 2 (后分频) 分频比选择位

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

T2ON: T2 使能位

- 1 = 使能 T2
- 0 = 禁止 T2

T2CKPS<1:0>: T2 分频器 1 (预分频) 分频比选择位

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

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--0	bit7						bit0
	T2MOD1	T2MOD0	T2OFS1	T2OFS0	T2TRIGEN	-	T2DIR
	R/W	R/W	R/W	R/W	R/W	U	R

T2MOD<1:0>: T2 计数方式选择位。

00 = 向上计数，上溢时产生中断标志
01 = 向上-向下计数，上溢时产生中断标志
10 = 向上-向下计数，下溢时产生中断标志
11 = 向上-向下计数，上溢和下溢时都产生中断标志

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

6.3.2.3 T2CCR1/0H 和 T2CCR1/0L 寄存器

T2CCR1/0H 和 T2CCR1/0L 寄存器是 T2 触发 ADC 启动设置寄存器。通过 T2CCR_xON (x=0、1) 位可以控制相应的 T2CCR_xH/L (x=0、1) 是否工作。当 T2CCR_xON (x=0、1) 使能时，系统不停的将 T2H/L 的值与寄存器 T2CCR_xH/L (x=0、1) 相比较，如果等于 T2CCR_xH/L (x=0、1)，且对应的控制位置 1(见 7.1.1 和 7.1.4 寄存器)，则会发出信号，使 ADCCTL0 的 START (ADCCTL0<1>) 位置 1，使能 ADC 采样转换。

6.3.3 T2 中断

如图 6.3 所示，预分频器 2 的计算器达到设定值后将会使中断标志位置 1，如果使能位 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 使用系统时钟，因此 T2 将停止工作。

6.4 定时/计数器 T3

T3 是一个 16 位的定时/计数器，T3 的低 8 位在寄存器 T3L 中，高 8 位在寄存器 T3H 中，当 T3 计数值达到 65535 后，T3 的值再加 1 就会产生溢出，将 T3 中断标志位置 1。

T3 带有重载功能。当使能 T3 重载功能时，T3 计数器计数到 T3 重载寄存器中设置的值时，T3 计数器将清 0 重新开始计数，且将 T3 中断标志位置 1。

T3 属于外部单元，因此在使用 T3 中断时，需将 PUIE 位置 1，使能外设中断。

6.4.1 T3 原理框图

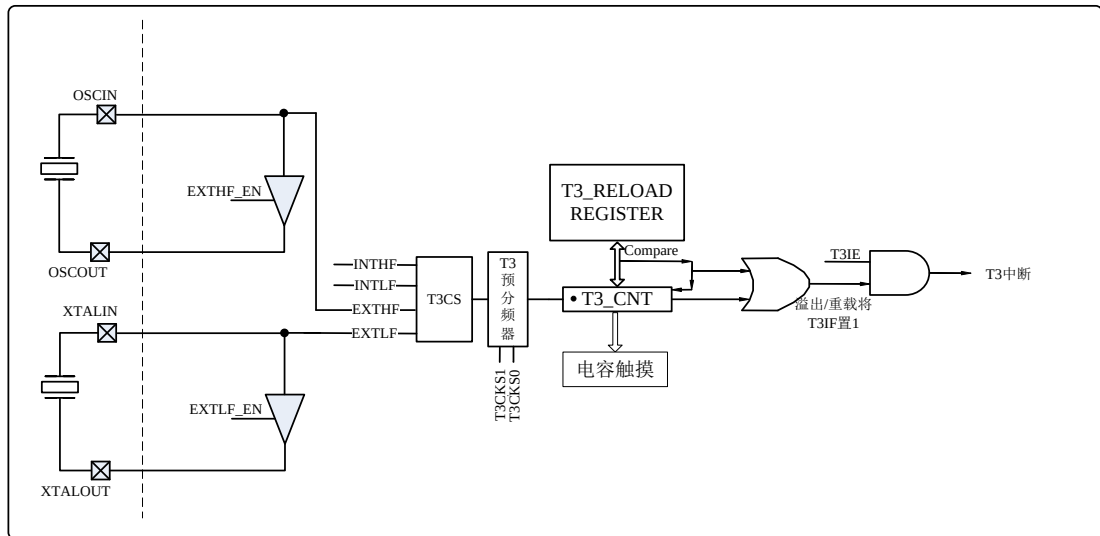


图 6.4 定时器 T3 原理框图

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
4EH	T3CTL	T3REN	-	T3CKS1	T3CKS0	-	T3CS1	T3CS0	T3ON
4FH	T3L	T3 低位寄存器							
5FH	T3H	T3 高位寄存器							
15EH	T3REH	T3 重载设置寄存器高 8 位							
15DH	T3REL	T3 重载设置寄存器低 8 位							
13H	CCP3CTL				CCP3OE	CCP3MOD3	CCP3MOD2	CCP3MOD1	CCP3MOD0
14H	CCP3H	CCP3 寄存器高 8 位							
15H	CCP3L	CCP3 寄存器低 8 位							

6.4.2.1 T3 控制寄存器

如寄存器 6.5 所示，T3 控制寄存器（T3CTL）用于启动/禁止 T3 以及选择 T3 模块的不同功能特性。

寄存器6.5: T3CTL: T3控制寄存器(地址: 4EH)

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

- T3REN:** T3 重载功能使能位
 0 = 禁止 T3 重载功能
 1 = 使能 T3 重载功能
- T3CKS<1:0>:** T3 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
- T3CS<1:0>:** T3 定时/计数模式选择
 00 = T3 时钟为内部高频时钟 INTHF
 01 = T3 时钟为内部低频时钟 INTLF
 10 = T3 时钟为外部高频时钟 EXTHF
 11 = T3 时钟为外部低频时钟 EXTLE
- T3ON:** T3 启动控制位
 1 = 启动 T3
 0 = 停止 T3

6.4.2.2 CCP3 功能寄存器

寄存器6.7: CCP3CTL: CCP3控制寄存器(地址: 13H)

复位值	bit7							bit0
---0 0000	-	-	-	CCP3OE	CCP3 MOD3	CCP3 MOD2	CCP3 MOD1	CCP3 MOD0
	U	U	U	R/W	R/W	R/W	R/W	R/W

CCP3OE: CCP3比较模式输出使能位

1 = 比较模式下使能引脚的CCP3输出

0 = 禁止比较模式下引脚的CCP3输出

CCP3MOD<3:0>: CCP3模式选择位

0000 = 捕捉/比较 关闭 (复位CCP3模块)

0001 = 未使用 (保留)

0010 = 比较模式: 匹配时输出电平翻转 (T3IF置1)

0011 = 未使用 (保留)

0100 = 捕捉模式: 在每个下降沿发生捕捉

0101 = 捕捉模式: 在每个上升沿发生捕捉

0110 = 捕捉模式: 每4 个上升沿发生捕捉

0111 = 捕捉模式: 每16 个上升沿发生捕捉

1000 = 比较模式: 比较匹配时输出高电平 (T3IF置1)

1001 = 比较模式: 比较匹配时输出低电平 (T3IF置1)

1010 = 比较模式: 比较匹配时产生软件中断 (T3IF置1, CCP3OUT不受影响)

1011 = 比较模式: 触发特殊事件 (T3IF位置1, CCP3复位T3, 如果ADC使能将启动AD转换)

1100 = 保留

1101 = 保留

1110 = 保留

1111 = 保留

6.4.3 T3 预分频器

如寄存器 6.5 所示, T3 具有四个预分频器选择项, 允许对时钟输入进行 1、2、4、或 8 倍分频。T3CKS 位(T3CTL<5:4>)对预分频计数器进行控制。T3 预分频计数器不能直接进行读写操作, 可通过写入 T3H 或 T3L 使预分频计数器清 0。

6.4.4 T3 计数时钟选择

用户可以通过 T3CTL 寄存器的 T3CS<1:0>位来选择 T3 的计数时钟, KF8L15Z20XX 系列提供 4 个时钟源, 内部系统高频时钟、内部低频时钟、外部高频时钟和外部低频时钟。

6.4.5 T3 重载功能

置位 T3CTL 寄存器的 T3REN 位使能 T3 重载功能。通过设置 T3REH/T3REL 寄存器来设置 T3 的重载点。当设置 T3REH/T3REL 寄存器且使能重载功能后, T3 计数器从 0 开始计数, 当计数到 T3REH/T3REL 寄存器设置的值时, T3 计数器清 0 并重新开始计数, T3IF

置 1。

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

6.4.6 T3 分配给电容触摸

电容触摸功能使能会将定时器 T3 自动分配给电容触摸模块，T3EN 位失效；此时用户可通过 CTCTL 寄存器的 CTSTART 位判断定时器的状态。

6.4.7 CCP3

定时器 T3 带有 CCP 功能：捕捉模式和比较模式。

6.4.7.1 捕捉模式

在捕捉模式下，当对应的 CCP3IN 引脚发生事件时，寄存器 CCP3H 和 CCP3L 捕捉 T3 寄存器的 16 位值，原理框图如下图 6.5 所示：

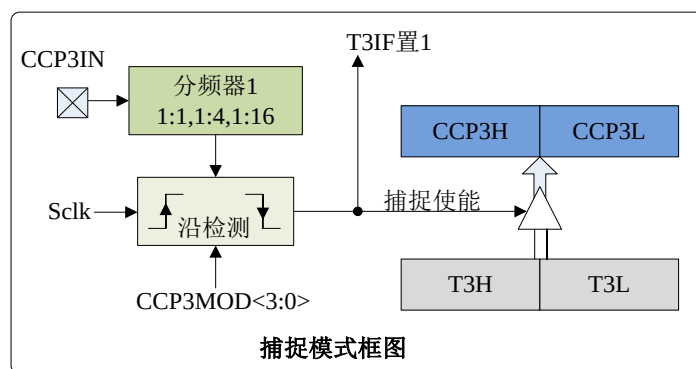


图6.5 捕捉模式原理框图

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

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

在捕捉模式下，应通过将相应的 CCP3 引脚配置为数字输入。

当一个捕捉发生时，硬件自动中断请求标志位 T3IF 置 1；它必须用软件清零。注意如果在 CCP3H、CCP3L 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 T3IE 中断允许位清零以避免产生误中断。在捕捉模式发生任何改变之后也应清零中断标志位 T3IF。

CCP3CTL 寄存器中的 CCP3MOD<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 CCP3CTL 寄存器清零关闭该模块。

6.4.7.2 比较模式

在比较模式下，16 位 CCP3H、CCP3L 寄存器的值将不断与 T3 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：

- ◆ CCP3OUT 引脚的输出电平翻转
- ◆ CCP3OUT 引脚输出高电平
- ◆ CCP3OUT 引脚输出低电平
- ◆ 产生软件中断触发信号
- ◆ 产生特殊事件触发信号

CCP3OUT 引脚的动作取决于 CCP3CTL 寄存器中 CCP3MOD<3:0> 控制位的值。

所有比较模式都会产生中断。原理图如图 6.6 所示：

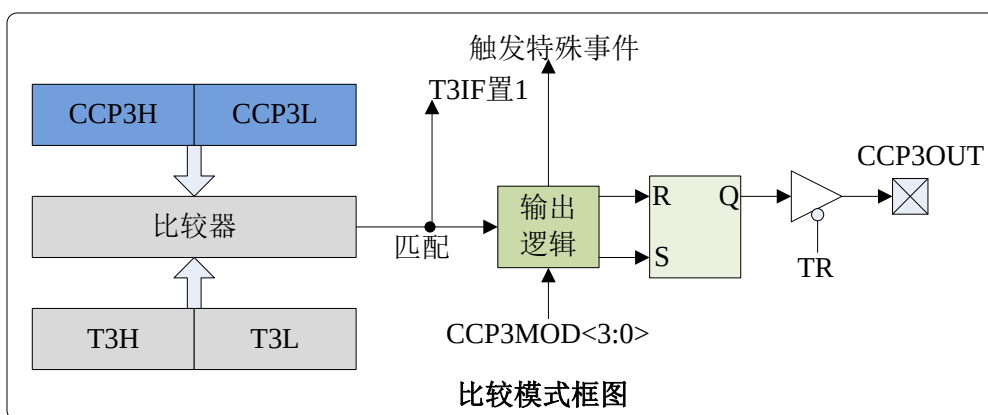


图 6.6 比较模式原理框图

用户必须通过将相应的 CCP3OUT 引脚配置为数字输出。

- 1) 当选择输出电平翻转模式时，比较匹配时，CCP3OUT 引脚的输出电平翻转，并将 T3IF 置 1。
- 2) 当选择普通比较模式时（1010 或 1001）时，比较匹配时，CCP3OUT 引脚输出高电平或者低电平，并且将 T3IF 置 1。
- 3) 当选择了软件中断触发模式时（1010）时，比较匹配时，将 T3IF 置 1，但是 CCP 模块不会控制 CCP3OUT 引脚。
- 4) 当选择了特殊事件触发模式（1011）时，比较匹配时，CCP 会立即产生特殊事件触发输出，将 T3IF 置 1，此时如果 ADC 已使能，将启动 AD 转换。但 T3 计数寄存器不会立即复位，直到 T3 计数脉冲的下一个上升沿才复位。从而使 CCP3H、CCP3L 寄存器实际上成为了定时器 T3 的 16 位可编程周期寄存器。

6.4.8 T3 中断

在两种情况下会使 T3 中断标志位置 1:

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

6.4.9 T3 工作在休眠模式

当选择外部时钟信号作为 T3 的计数时钟时, T3 模块可以工作在休眠模式下。当使能 T3 中断时, 系统会在 T3 溢出或者重载时唤醒。

6.4.10 T3 计数寄存器被比较器清零

T3 计数寄存器 T3H/T3L 可被比较器的输出变化清零, 详见 12.8 比较器清零定时器。

6.5 定时/计数器 T4

T4 是一个 16 位的定时/计数器，T4 的低 8 位在寄存器 T4L 中，高 8 位在寄存器 T4H 中，当 T4 计数值达到 65535 后，T4 的值再加 1 就会产生溢出，将 T4 中断标志位置 1。

T4 带有重载功能。当使能 T4 重载功能时，T4 计数器计数到 T4 重载寄存器中设置的值时，T4 计数器将清 0 重新开始计数，且将 T4 中断标志位置 1。

T4 属于外部单元，因此在使用 T4 中断时，需将 PUIE 位置 1，使能外设中断。

6.5.1 T4 原理框图

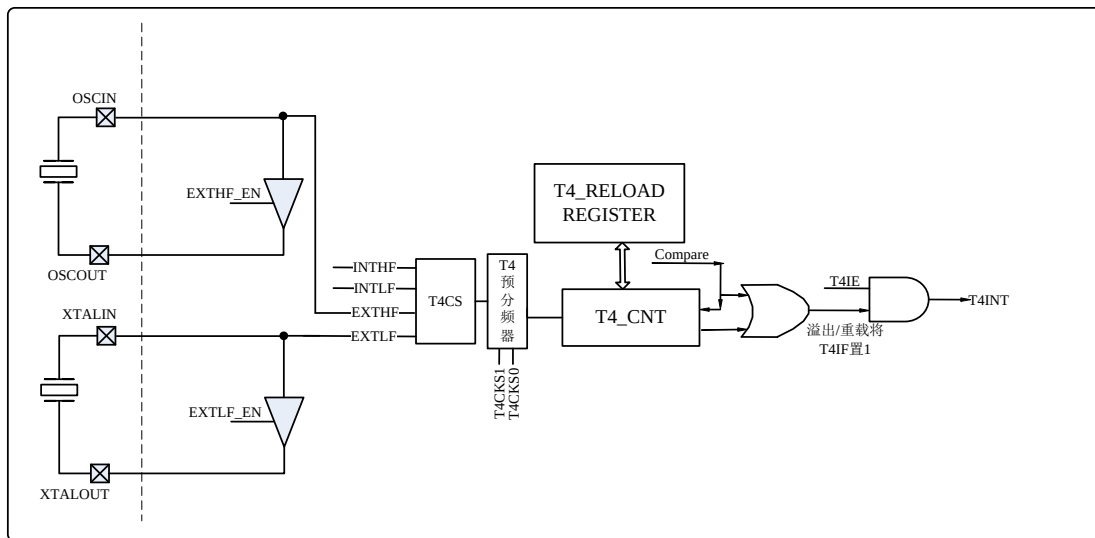


图 6.7 T4 原理框图

6.5.1.1 T4 相关寄存器

表 6-5 与 T4 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
164H	T4CTL	T4REN	T4_BUZ OE	T4CKS1	T4CKS0	-	T4CS1	T4CS0	T4ON
161H	T4H	T4 计数器高 8 位							
160H	T4L	T4 计数器低 8 位							
163H	T4REH	T4 重载设置寄存器高 8 位							
162H	T4REL	T4 重载设置寄存器低 8 位							
109H	CCP4CTL				CCP4OE	CCP4MOD3	CCP4MOD2	CCP4MOD1	CCP4MOD0
106H	CCP4H	CCP4 寄存器高 8 位							
105H	CCP4L	CCP4 寄存器低 8 位							

6.5.1.2 T4 控制寄存器

如寄存器 6.6 所示，T4 控制寄存器（T4CTL）用于启动/禁止 T4 以及选择 T4 模块的不同功能特性。

寄存器6.6: T4CTL: T4控制寄存器(地址: 164H)

	bit7						bit0	
复位值 0000 -000	T4REN	T4BUZOE	T4CKS1	T4CKS0	-	T4CS1	T4CS0	T4ON
	R/W	R/W	R/W	R/W	U	R/W	R/W	R/W

- T4REN:** T4 重载功能使能位
 0 = 禁止 T4 重载功能
 1 = 使能 T4 重载功能
- T4BUZOE:** T4 蜂鸣器控制输出使能位
 0 = 禁止 T4 蜂鸣器控制输出到 P0.7 口
 1 = 使能 T4 蜂鸣器控制输出到 P0.7 口
- T4CKS<1:0>:** T4 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
- T4CS<1:0>:** T4 定时/计数模式选择
 00 = T4 时钟为内部高频时钟 INTHF
 01 = T4 时钟为内部低频时钟 INTLF
 10 = T4 时钟为外部高频时钟 EXTHF
 11 = T4 时钟为外部低频时钟 EXTLE
- T4ON:** T4 启动控制位
 1 = 启动 T4
 0 = 停止 T4

6.5.1.3 CCP4 功能寄存器

寄存器6.7: CCP4CTL: CCP4控制寄存器(地址: 109H)

	bit7				bit0			
复位值	-	-	-	CCP4OE	CCP4MOD3	CCP4MOD2	CCP4MOD1	CCP4MOD0
---0 0000	U	U	U	R/W	R/W	R/W	R/W	R/W

CCP4OE: CCP4比较模式输出使能位

1 = 比较模式下使能引脚的CCP4输出

0 = 禁止比较模式下引脚的CCP4输出

CCP4MOD<3:0>: CCP4模式选择位

0000 = 捕捉/比较 关闭（复位CCP4模块）

0001 = 未使用（保留）

0010 = 比较模式：匹配时输出电平翻转（T4IF置1）

0011 = 未使用（保留）

0100 = 捕捉模式：在每个下降沿发生捕捉

0101 = 捕捉模式：在每个上升沿发生捕捉

0110 = 捕捉模式：每4 个上升沿发生捕捉

0111 = 捕捉模式：每16 个上升沿发生捕捉

1000 = 比较模式：比较匹配时输出高电平（T4IF置1）

1001 = 比较模式：比较匹配时输出低电平（T4IF置1）

1010 = 比较模式：比较匹配时产生软件中断（T4IF置1，CCP4OUT不受影响）

1011 = 比较模式：触发特殊事件（T4IF位置1，CCP4复位T4，如果ADC使能将启动AD转换）

1100 = 保留

1101 = 保留

1110 = 保留

1111 = 保留

6.5.2 T4 预分频器

如寄存器 6.6 所示，T4 具有四个预分频器选择项，允许对时钟输入进行 1、2、4、或 8 倍分频。T4CKS 位(T1CTL<5:4>)对预分频计数器进行控制。T4 预分频计数器不能直接进行读写操作，可通过写入 T4H 或 T4L 使预分频计数器清 0。

6.5.3 T4 计数时钟选择

用户可以通过 T4CTL 寄存器的 T4CS<1:0>位来选择 T4 的计数时钟源，~~KF8F4265-系列?~~提供 4 个时钟源，内部系统高频时钟、内部低频时钟、外部高频时钟和外部低频时钟。

6.5.4 T4 重载功能

置位 T4CTL 寄存器的 T4REN 位使能 T4 重载功能。通过设置 T4REH/T4REL 寄存器来设置 T4 的重载点。当设置 T4REH/T4REL 寄存器且使能重载功能后，T4 计数器从 0 开始计数，当计数到 T4REH/T4REL 寄存器设置的值时，T4 计数器清 0 并重新开始计数，T4IF 置 1。

T4REN 位置 1 时会载入一次 T4REH/T4REL 寄存器的值，当置位 T4REN 且在 T4 计数过程中对 T4REH/T4REL 寄存器写入新值时，T4 模块会在下一次重载时载入新的重载点。

6.5.5 CCP4

定时器 T4 带有 CCP 功能：捕捉模式和比较模式。

6.5.5.1 捕捉模式

在捕捉模式下，当对应的 CCP4IN 引脚发生事件时，寄存器 CCP4H 和 CCP4L 捕捉 T4 寄存器的 16 位值，原理框图如下图 6.6 所示：

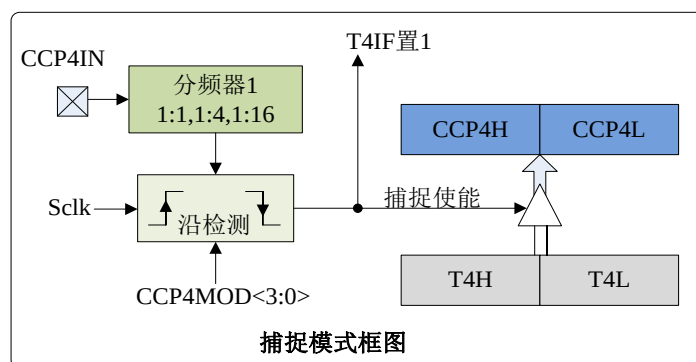


图6.8 捕捉模式原理框图

触发捕捉的事件可被定义为以下四者之一，并且由 CCP4CTL 寄存器中的模式选择位 CCP4MOD<3:0> 选择事件类型位配置：

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

在捕捉模式下，应通过将相应的 CCP4 引脚配置为数字输入。

当一个捕捉发生时，硬件自动中断请求标志位 T4IF 置 1；它必须用软件清零。注意如果在 CCP4H、CCP4L 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 T4IE 中断允许位清零以避免产生误中断。在捕捉模式发生任何改变之后也应清零中断标志位 T4IF。

CCP4CTL 寄存器中的 CCP4MOD<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 CCP4CTL 寄存器清零关闭该模块。

6.5.5.2 比较模式

在比较模式下，16 位 CCP4H、CCP4L 寄存器的值将不断与 T4 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：

- ◆ CCP4OUT引脚的输出电平翻转
- ◆ CCP4OUT引脚输出高电平
- ◆ CCP4OUT引脚输出低电平
- ◆ 产生软件中断触发信号
- ◆ 产生特殊事件触发信号

CCP4OUT引脚的动作取决于CCP4CTL寄存器中CCP4MOD<3:0>控制位的值。

所有比较模式都会产生中断。原理图如图6.7所示：

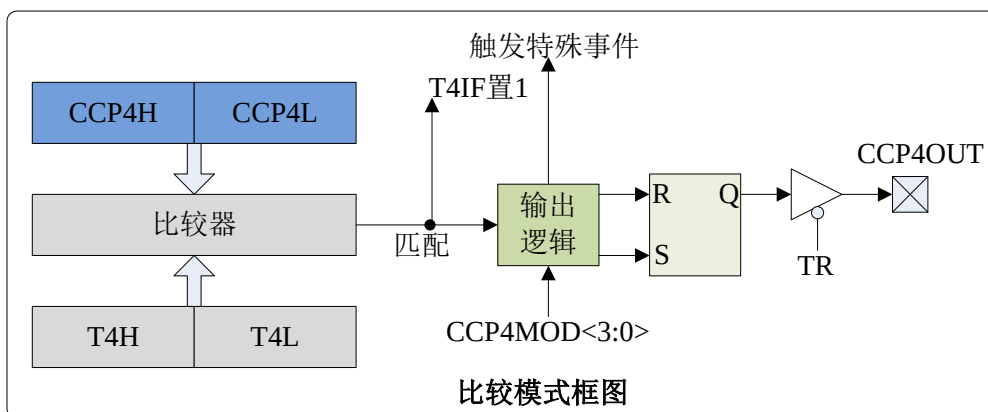


图6.9 比较模式原理框图

用户必须通过将相应的CCP4OUT引脚配置为数字输出。

- 5) 当选择输出电平翻转模式时，比较匹配时，CCP4OUT引脚的输出电平翻转，并将T4IF置1。
- 6) 当选择普通比较模式时（1010或1001）时，比较匹配时，CCP4OUT引脚输出高电平或者低电平，并且将T4IF置1。
- 7) 当选择了软件中断触发模式时（1010）时，比较匹配时，将T4IF置1，但是CCP模块不会控制CCP4OUT引脚。
- 8) 当选择了特殊事件触发模式（1011）时，比较匹配时，CCP会立即产生特殊事件触发输出，将T4IF置1，此时如果ADC已使能，将启动AD转换。但T4计数寄存器不会立即复位，直到T4计数脉冲的下一个上升沿才复位。从而使CCP4H、CCP4L寄存器实际上成为了定时器T4的16位可编程周期寄存器。

6.5.6 T4 中断

在三种情况下会使 T4 中断标志位置 1:

1. 当 T4 计数值达到 65535 后, T4 的值再加 1 就会产生溢出, 将 T4 中断标志位置 1;
2. 当 T4 重载时, T4 中断标志位置 1。
- 3 当用于 CCP 功能时产生中断, T4 中断标志位置 1。

6.5.7 T4 工作在休眠模式

当选择低频时钟信号作为 T4 的计数时钟时, T4 模块可以工作在休眠模式下。当使能 T4 中断时, 系统会在 T4 溢出或者重载时唤醒。

6.5.8 T4 计数寄存器被比较器清零

T4 计数寄存器 T4H/T4L 可被比较器的输出变化清零, 详见 12.8 比较器清零定时器。

7 ADC12 模数转换模块

ADC12 模数转换模块可将模拟输入信号转换为 12 位二进制值。KF8L15Z20XX 拥有 6 路 IO 模拟输入通道和 4 路内部通道。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 VREFOUT、VDD 或施加在 VREFIN 引脚上的电压作为转换使用的参考电压。

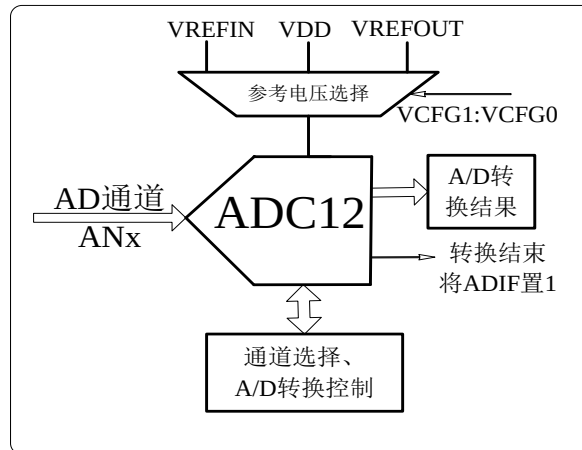


图 7.1 ADC12 模块结构框图

7.1 与 ADC12 相关的寄存器

表 7-1 与 ADC12 转换相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
1FH	ADCCTL0	ADLR	T2CCR0 ON	-	-	-	-	START	ADEN
3FH	ADCCTL1	ADCALEN	ADCS2	ADCS1	ADCS0	VCFG1	VCFG0	-	-
63H	ADCCTL2		ADCHS6	ADCHS5	ADCHS4	ADCHS3	ADCHS2	ADCHS1	ADCHS0
18H	ADSCANCTL	ADSCANEN	SMOD1	SMOD0	STIM4	STIM 3	STIM 2	STIM 1	STIM 0
58H	ADCINTCTL	T2CCR1 ON	-	-	-	-	-	INTCTL1	INTCTL0
236H	TEMPSNR	-	-	-	-	-	-	-	TSEN
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							
159H	VREFCAL0	内部参考电压校准寄存器 0							
15AH	VREFCAL1	内部参考电压校准寄存器 1							

7.1.1 ADC12 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 00-- --00	bit7						bit0	
	ADLR	T2CCR0 ON	-	-	-	-	START	ADEN
	R/W	R/W	U	U	U	U	R/W	R/W

ADLR: ADC12 转换结果输出格式选择位

1 = 结果右对齐

0 = 结果左对齐

T2CCR0ON: T2CRR0H/L 与 T2H/L 匹配时触发 ADC12 启动使能位

1 = 使能 T2 触发 ADC12 启动

ADC12 采样转换结束后, 将采样的数据保存到

ADCDATA0H 和 ADCDATA0L 中, 并且硬件自动将该位清 0

0 = 禁止 T2 触发 ADC12 启动

START: ADC12 转换状态位

1 = ADC12 转换正在进行, 该位置 1 将启动 ADC12 转换, 在转换结束后该位将被硬件自动清 0。

0 = ADC12 转换结束或者未进行

ADEN: ADC12 工作使能位

1 = 使能 ADC12 转换模块工作

0 = ADC12 转换器关闭且不消耗工作电流

注: 如果内部使用参考电压VREFOUT时 (用作ADC12参考电压和比较器参考电压), 不需要将寄存器VRECTL1的VREOE位置1。

7.1.2 ADC12 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 0000 0000	bit7						bit0	
	ADCALE N	ADCS2	ADCS1	ADCS0	VCFG1	VCFG0	保留	保留
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ADCALEN: ADC12 失调校准使能位

0 = 关闭 ADC12 校准

1 = 打开 ADC12 校准

ADCS<2:0>: ADC12 转换时钟选择位

000 = Fad=SCLK/2

001 = Fad=SCLK/8

010 = Fad=SCLK/32

x11 = Fad=内部专用 500KHz 时钟

100 = Fad=SCLK/4

101 = Fad=SCLK/16

110 = Fad=SCLK/64

VCFG<1:0>: ADC12 转换参考电压选择位

00=断开

01 = VDD 作为 ADC12 参考电压

10 = VREFIN 作为 ADC12 参考电压

11 = VREFOUT 作为 ADC12 参考电压

7.1.3 ADC12 控制寄存器 2(ADCCTL2)

如寄存器 7.3 所示, ADC12 控制寄存器 2 包含 ADC12 通道选择位 CHS<6:0>。

寄存器7.3: ADCCTL2:A/D控制寄存器2（地址：63H）

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

ADCHS<6:5>: 保留位, 请勿将该两位置 1, 否则将无法配置 ADC12 通道

ADCHS<4:0>: 模拟通道选择位

ADCHS_x = ADC12 模拟通道 x

ADCHS	通道数	ADCHS	通道数
0 0000	保留	0 1100	通道 12/P0.2
0 0001	保留	0 1101	通道 13/P0.0
0 0010	保留	0 1110	通道 14/OPOUT
0 0011	保留	0 1111	通道 15/TSOUT
0 0100	保留	1 0000	通道 16/VREFOUT
0 0101	保留	1 0001	通道 17/P0.7
0 0110	保留	1 0010	通道 18/P5.0
0 0111	保留	1 0011	通道 19/P5.1
0 1000	通道 8/P5.2	1 0100	保留
0 1001	通道 9/P6.5	1 0101	保留
0 1010	通道 10/P6.6	1 0110	保留
0 1011	通道 11/P0.1	1 0111	保留
-	-	11000	保留

7.1.4 ADC12 中断控制寄存器（ADCINTCTL）

寄存器7.4: ADCINTCTL: ADC12中断控制寄存器(地址:58H)

复位值	bit7						bit0	
0--- --00	T2CCR10N	-	-	-	-	-	INTCTL1	INTCTL0
	R/W	U	U	U	U	U	R/W	R/W

T2CCR10N: T2CCR1H/L 寄存器触发 ADC12 启动使能位

1 = 使能 T2 触发 ADC12, ADC12 采样转换结束后, 将采样的数据保存到 ADCDATA4H 和 ADCDATA4L 中, 并且硬件自动将该位清 0

0 = 禁止 T2 触发 ADC12

INTCTL<1:0>: ADC12 中断控制位

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

ADSCANEN = 1 时，
 00 = 完成 1 次转换后产生中断
 01 = 完成 2 次转换后产生中断
 10 = 完成 3 次转换后产生中断
 11 = 完成 4 次转换后产生中断

7.1.5 温度传感器配置寄存器(TEMPSNR)

寄存器7.5: TEMPSNR: 温度传感器配置寄存器(地址:236H)

复位值 ----00	bit7							bit0
	-	-	-	-	-	-	-	TSEN
	U	U	U	U	U	U	R/W	R/W

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

7.2 通道的选择

如图 7.1 所示，KF8L15Z20XX 中的 ADC12 转换模块的输入可以选择 6 路来自外部的模拟信号和 4 路内部通道。通过寄存器 ADCCTL2(如寄存器 7.3 所示)进行通道的选择。当选择外部的模拟信号作为 ADC12 转换的输入时，需要将对应的引脚配置为模拟输入口；当选择内部信号作为 ADC12 转换的输出时，无需占用任何 I/O 引脚；ADC12 可选的内部信号输入及配置方式如下（用户如需使用温度传感器，通过 TEMPSNR 寄存器的 TSEN 位置 1 来使能该功能）：

- 通道 14=OPOUT，运算放大器的输出
- 通道 15=TSOUT，温度传感器的输出
- 通道 16=VREFOUT，内部参考电压

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。

7.3 ADC12 转换参考电压的选择

KF8L15Z20XX 中 ADC12 模块的参考电压可以选择 3 种分别为：电源电压(VDD)、外部参考电压(VREFIN)和内部参考电压 VREFOUT。通过寄存器 ADCCTL1(如寄存器 7.2 所示)的 VCFG<1:0>设置参考电压。

7.4 转换时钟的选择

完成一次 ADC12 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 ADC12 转换时钟周期和频率。

7.5 输出格式

KF8L15Z20XX 中 ADC12 转换的结果为 12 位二进制数，ADC12 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。

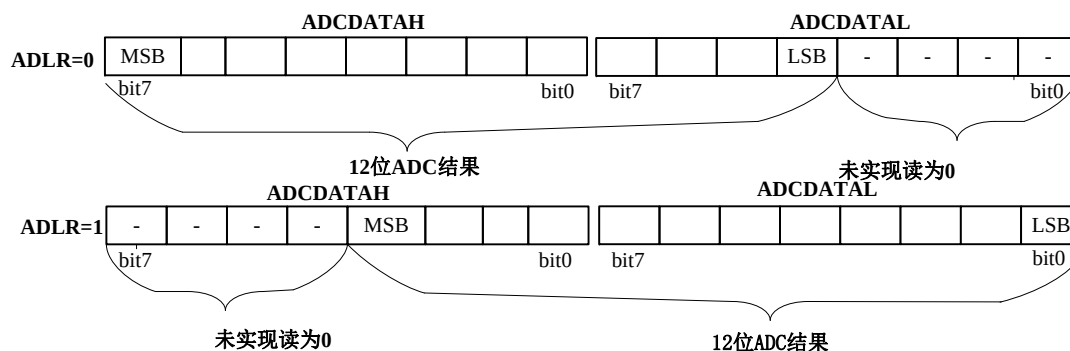


图 7.2 ADC 输出对齐方式

7.6 AD 转换的启动和完成

先将 ADEN 位置 1，然后将 START 位(ADCCR0.1)置 1 即可启动 AD 转换。当转换结束时，ADC12 模块将：

1. 将 START 位清 0
2. 将 ADIF 位置 1
3. 如果使能 AD 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 AD 转换采样全部结束之前，ADCDATA0H:ADCDATA0L 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。AD 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。

7.7 ADC12 工作在休眠模式

ADC12 转换器模块可以在休眠模式下工作。这需把 ADC12 的时钟源设定为 ADC12 专用的内部振荡器。当选择了专用内部时钟源后，ADC12 需等待一个指令周期后才能启动转换操作，转换结束后，START 位将被清 0，且转换结果将被载入 ADCDATA0H:ADCDATA0L 寄存器。如果 ADC12 中断被使能，器件将从休眠状态唤醒。如果 ADC12 中断被禁止，ADC12 模块在转换完成后被关闭。

如果 ADC12 时钟源不是 ADC12 专用的内部振荡器，执行 IDLE 指令将导致当前转换操作中止，并使 ADC12 模块关闭。

7.8 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，ADC12 模块将被关闭，任何进行中的转换操作被中止。ADCDATA0H:ADCDATA0L 寄存器中的值不变。

7.9 使用 ADC12 转换器的设置

启动 ADC12 转换器时的设置:

1. 选择 ADC12 采样输入通道，设置 AD 转换结果对齐方式；
2. 将对应的 ADC12 采样输入通道设置为模拟输入模式；
3. 如果需要使能 ADC12 的失调校准，将 ADCCTL1 寄存器的 ADCALEN 位置 1；
4. 选择参考电压和 ADC12 采样时钟频率，打开 AD 转换；
5. 如果采用中断方式，使能 ADC12 转换中断；
6. 等待 ADC12 所需的采集时间；
7. 启动 AD 转换；
8. 查询 ADC12 是否转换完成(START=0)或进入 AD 中断；
9. 读取 AD 转换结果。

8 DAC12 数模转换器模块

在实际应用中许多元件使用的是模拟信号,处理后的数字信号需要再转换为模拟信号才能在实际电路中使用,所以就需要在输出时进行数字信号向模拟信号的转换,即数模转换器(DAC12),它提供一个可变的参考电压,它与输入源成比例。

KF8L15Z20XX 系列单片机包含一个 12 位的通用数模转换器(DAC12)模块,可以将 12 位的二进制数值转换成模拟信号。DAC12 可以由外部引脚输出。

DAC12 的特性:

- 片上可编程基准信号发生器输出。电压输出范围为 $V_{in} \sim 1/4096V_{in}$
- 三选一 V_{in} 基准源选择(外部参考电压 V_{REFIN} 、 V_{DD} 、 FVR)
- DAC12 带有专用的运算放大器作为输出缓冲器
- DAC12 带有失调消除功能

8.1 DAC12 原理框图

DAC12 的原理框图如图 9.1 所示。

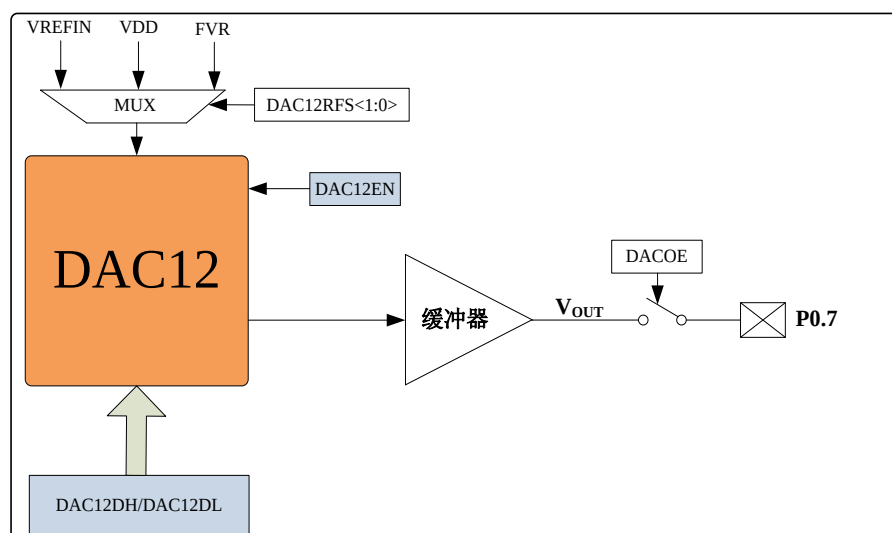


图 9.1 DAC12 原理框图

8.2 DAC12 相关寄存器

表 9-1 DAC12 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
64H	DAC12CTL	DAC12EN	BUF CALI	DAC12OE	-	-	-	DAC12RFS1	DAC12RFS0
65H	DAC12DH	-	-	-	-	DAC12D11	DAC12D10	DAC12D9	DAC12D8
66H	DAC12DL	DAC12D7	DAC12D6	DAC12D5	DAC12D4	DAC12D3	DAC12D2	DAC12D1	DAC12D0

8.2.1 DAC12 控制寄存器 (DAC12CTL)

寄存器8.1: DAC12CTL: DAC12控制寄存器(地址: 64H)

复位值 0000 0000	bit7						bit0
	DAC12EN	BUFCALI	DAC12OE	-	-	-	DAC12RFS1 DAC12RFS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DAC12EN: DAC12 使能位

1 = 使能 DAC12

0 = 禁止 DAC12

BUFCALI: DAC12 失调消除功能配置位

1 = 打开 DAC12 失调消除功能

0 = 关闭 DAC12 失调消除功能

DAC12OE: DAC12 输出使能位

1 = 使能 DAC12 输出到端口 P0.7

0 = 禁止 DAC12 输出到端口 P0.7

DAC12RFS<1:0>: DAC12 参考电压选择位

00 = DAC12 选择供电电源 VDD 作为参考电压

01 = DAC12 选择内部参考电压 FVR 作为参考电压

10 = 保留

11 = DAC12 选择外部参考电压 VREFIN 作为参考电压

8.2.2 DAC12 数据寄存器 (DAC12DH/DAC12DL)

寄存器8.2: DAC12DH: DAC12数据寄存器(地址: 65H)

复位值 0000 0000	bit7						bit0
	-	-	-	-	DAC12D11 DAC12D10	DAC12D9 DAC12D8	DAC12D7 DAC12D6
	R/W	R/W	R/W	R/W	R/W	R/W	R/W

寄存器8.3: DAC12DL: DAC12数据寄存器(地址: 66H)

复位值 0000 0000	bit7						bit0
	DAC12D7 DAC12D6	DAC12D5 DAC12D4	DAC12D3 DAC12D2	DAC12D1 DAC12D0	DAC12D7 DAC12D6	DAC12D5 DAC12D4	DAC12D3 DAC12D2
	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DAC12D<11:0>: DAC12 的 12 位二进制数据

DAC12D<11:0>装载到{DAC12DH:DAC12DL}寄存器中, 然后通过 DAC12 转换器转换成模拟信号输出。

DAC12 输出电压计算公式如下:

$$V_{out} = V_{in} \times \frac{DAC12D[11:0]}{4096}。$$

8.3 DAC12 基准选择

DAC12 模块均可选择 3 种电压参考源输入：供电源 VDD、内部参考电压 FVR 和外部参考电压 VREFIN。通过 DAC12CTL 的 DAC12RFS<1:0>位设置 DAC12 的基准源。

8.4 DAC12 失调消除功能

DAC12CTL 寄存器的 BUFCALI 位置 1，将使能失调消除功能；如果需要使用失调消除功能，须满足如下两个要求：

- 在片外 P0.7 口外加电容（推荐 50pf 以上）
- 系统时钟频率配置为 2MHz 或者 2MHz 以下

8.5 使用 DAC12 的设置

启动 DAC12 时的设置如下：

1. 设置 DA12CTL 的 DAC12RFS<1:0>位选择 D/A 参考电压；
2. 设置 DAC12DH/L，写入转换值；
3. 配置 DAC12CTL 的 BUFCALI 位选择是否使用失调消除功能；
4. 将 DAC12CTL 寄存器的 DAC12EN 位置 1 使能 D/A 转换。

9 PWM 模块

KF8L15Z20XX 单片机具有 4 路的 16 位的 PWM 模块: PWM1A、PWM1B、PWM1E、PWM1F。其中 4 路的 16 位 PWM1x 结构相同。

9.1 工作原理

启动 PWM 后，在对应的 PWM1x 引脚输出 PWM 脉冲。PWM 脉冲的频率和占空比通过 $\langle PP2:PP1 \rangle$ 和 $\langle PWM1xH:PWM1xL \rangle$ 设置。16 路 PWM 共用周期寄存器 PP1 和 PP2。

图 9.1 显示了 PWM1A 逻辑框图。其中 $\langle PP2:PP1 \rangle$ 为 PWM1A 模块的周期寄存器， $\langle PWM1AH:PWM1AL \rangle$ 为 PWM1A 模块占空比设置寄存器，使用 PWM1A 时需要将定时器 1 配置给 PWM1A 做定时用，并且需要使能 T1 重载。启动 PWM1 后，当 $\langle T1H:T1L \rangle$ 计数值和 $\langle PP2:PP1 \rangle$ 相等时，PWM1A 引脚被置 1，此时 $\langle T1H:T1L \rangle$ 被清 0，重新开始计数，当 $\langle T1H:T1L \rangle$ 的计数值和 $\langle PWM1AH:PWM1AL \rangle$ 相等时，PWM1A 引脚清 0(如图 9.2 所示)。改变 $\langle PP2:PP1 \rangle$ 和 $\langle PWM1AH:PWM1AL \rangle$ 的值可产生不同的 PWM1A 周期和 PWM1A 占空比。其他 PWM1x 模块的工作原理和 PWM1A 模块完全一致。

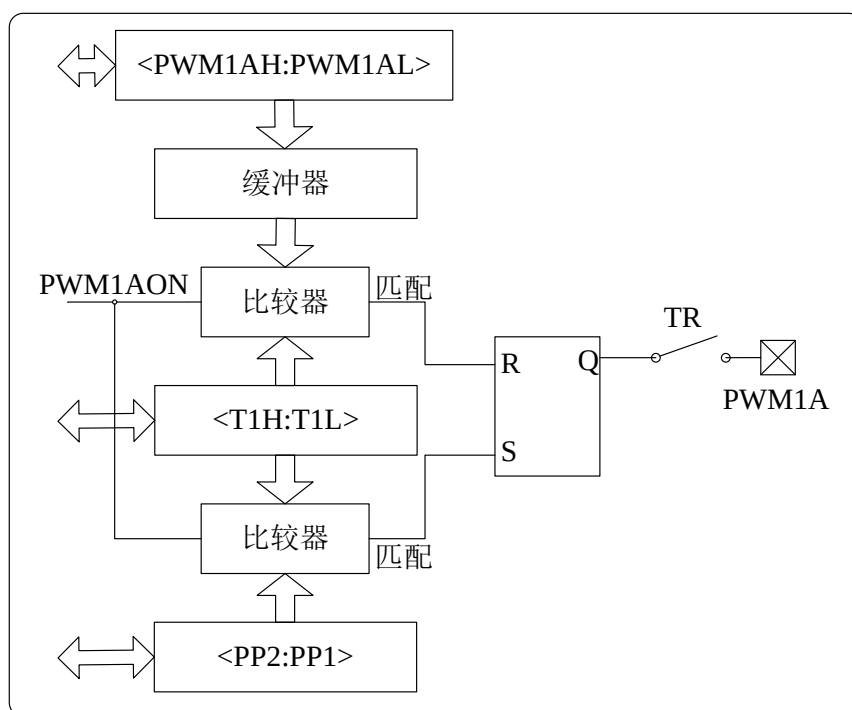


图 9.1 PWM1A 逻辑框图

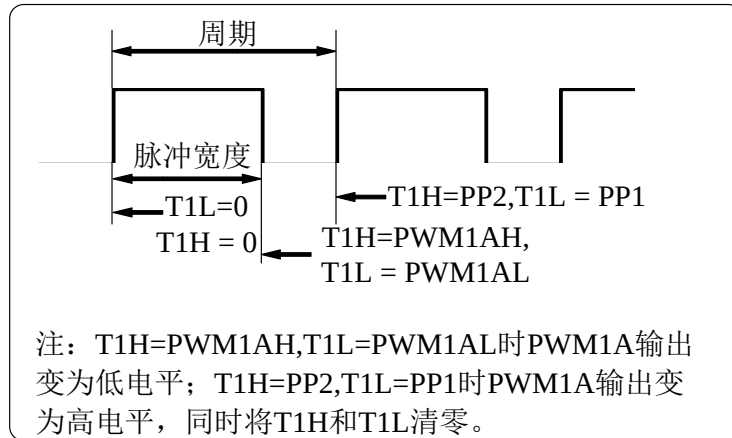


图 9.2 PWM1A 输出波形图

9.2 PWM1 输出端口

表 9-1: PWM1 对应 IO 口

PWM1	对应 I/O 口	备注
PWM1A	P6.4	
PWM1B	P5.2	
PWM1E	P4.1	
PWM1F	P4.0	

9.3 PWM1x 相关的寄存器

表 9-2: 与 PWM1x 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
21EH	PWMCTL0	PWM17O N	PWM16O N	PWM15O N	PWM14O N	PWM13O N	PWM12O N	PWM11O N	PWM10O N
26CH	PWMCTL1	PWM1FO N	PWM1EO N	PWM1DO N	PWM1CO N	PWM1BO N	PWM1AO N	PWM19O N	PWM18O N
21FH	PP1	PWM1x 周期设置寄存器低 8 位							
220H	PP2	PWM1x 周期设置寄存器高 8 位							
235H	PWM1AL	PWM1A 占空比低 8 位寄存器							
260H	PWM1AH	PWM1A 占空比高 8 位寄存器							
261H	PWM1BL	PWM1B 占空比低 8 位寄存器							
262H	PWM1BH	PWM1B 占空比高 8 位寄存器							
267H	PWM1EL	PWM1E 占空比低 8 位寄存器							
268H	PWM1EH	PWM1E 占空比高 8 位寄存器							
269H	PWM1FL	PWM1F 占空比低 8 位寄存器							
26AH	PWM1FH	PWM1F 占空比高 8 位寄存器							

9.3.1 PWM1x 控制寄存器

寄存器9.1: PWMCTL0: PWM启动控制寄存器0(地址:21EH)

复位值 0000 0000	bit7							bit0
	PWM17ON	PWM16ON	PWM15ON	PWM14ON	PWM13ON	PWM12ON	PWM11ON	PWM10ON
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

寄存器9.2: PWMCTL1: PWM启动控制寄存器1(地址:26CH)

复位值 0000 0000	bit7							bit0
	PWM1FON	PWM1EON	PWM1DON	PWM1CON	PWM1BON	PWM1AON	PWM19ON	PWM18ON
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWM1xON: PWM1x 启动控制位

1 = 启动 PWM1x

0 = 禁止 PWM1x

注:

x=A、B、E 和 F; 其他保留。

KF8L15Z20OG 只开放 PWM1A、PWM1B、PWM1E 和 PWM1F 这 4 路 PWM。

9.3.2 PWM1x 周期

PWM1x 周期通过<PP2: PP1>进行设置, PP1 和 PP2 是一个 8 位的寄存器, 其值可设置为 0~255, 这里将 PP1 和 PP2 组合起来使用构成一个 16 位的周期寄存器, 其值可设置为 0~65535。PWM 周期通过如下公式进行计算。

式 9.1:

$$\text{PWM1x周期} = (<\text{PP2:PP1}> + 1) \cdot 4 \cdot T_{\text{osc}} \text{ (T1预分频比)}$$

9.3.3 PWM1x 占空比

PWM1x 占空比通过<PWM1xH: PWM1xL>设置, 可写入一个 16 位的值到<PWM1xH: PWM1xL>来设置占空比。脉冲宽度和占空比通过如下公式计算:

式 9.2:

$$\text{脉冲宽度} = <\text{PWM1xH:PWM1xL}> \cdot 4 \cdot T_{\text{osc}} \text{ (T1预分频比)} \quad (x=0\sim F)$$

式 9.3:

$$\text{占空比} = \frac{\text{脉冲宽度}}{\text{PWM周期}} = \frac{<\text{PWM1xH:PWM1xL}>}{<\text{PP2:PP1}> + 1} \quad (x=0\sim F)$$

9.4 PWM1x 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。KF8L15Z20XX 中当 PP1 和 PP2 均为 255 时, PWM 的最大分辨率为 16 位。分辨率的计算公式如下所示。

式 9.4:

$$\text{分辨率} = \frac{\text{Log}[<\text{PP2:PP1}> + 1]}{\text{log}2} \text{ 位}$$

9.5 PWM1x 中断

PWM 拥有独立的中断使能位 **PWMIE**、中断标志位 **PWMIF** 和高优先级控制位 **PPWM**；同时，PWM 中断产生时，**T1IF** 也会置 1；因此，用户可选用任一组中断位，均可实现。

在 PWM 启用后，当 T1L/H 的计数值与 PWM1L/1H 的值匹配后，其对应的输出引脚变为低电平。当 T1L/H 的计数值与 PP1/2 的值匹配后，其对应的输出引脚变为高电平，同时将 T1L/H 清 0，将 **T1IF**(和 **PWMIF**)置 1，如果允许 T1（或者 PWM）中断，将会转入对应的中断子程序中。

9.6 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。PWM1x 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。

9.7 系统时钟频率的改变

PWM1x 的输出频率是通过 T1L/H 定时产生的，T1 的时钟源改变、T1 所选时钟源的频率发生任何改变都会使 PWM 频率发生变化。

9.8 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 PWM1x 使用的寄存器进入其复位状态。

9.9 PWM1x 使用方法

PWM1x 工作的设置应按照以下步骤：

- 1、将引脚 PWM1x 所对应的方向控制位 **TRxx** 置 1，禁止引脚 PWM1x 的输出驱动器。
- 2、赋 <PP2: PP1>寄存器的初值以设置 PWM1x 的 PWM 周期。
- 3、赋 <PWM1xH: PWM1xL>寄存器的初值以设置 PWM1x 的占空比。
- 4、配置并启动定时器/计数器 T1:
 - 配置 T1CTL 寄存器的 T1CKS1 和 T1CKS0 以选择 T1 的预分频比；
 - 将 T1L/H 清 0；
 - 使能 T1 重载功能
 - 将 T1CTL 寄存器的 T1ON 位置 1 以启动 T1。
- 5、将 PWMCTL 寄存器的 PWM1xON 置 1 以启动 PWM1x。
- 6、将引脚 PWM1x 所对应的方向控制位 **TRxx** 清 0 使能引脚 PWM1x 的输出驱动器。

10 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 PWM5L1 和 PWM5L0 分别作为数据寄存器的高 8 位和低 8 位寄存器; 捕捉和比较的 16 位数据寄存器 PWM5L0 和 PWM5L1 与 T1L 和 T1H 进行配合使用。

在 PWM5 模式下, 寄存器 PWM5L0 和 PWM5H0 作为 16 位占空比寄存器使用。

10.1 CCP 相关寄存器

当 CCP 模块配置为捕捉比较模式时, PWM5L1:PWM5L0 作为 CCP 的数据寄存器使用。

在捕捉模式下时, 当 CCP5IN 引脚发生事件时, PWM5L1:PWM5L0 这对寄存器捕捉 T1H 和 T1L 寄存器的 16 位值。T1H 和 T1L 寄存器的值自动传递给 PWM5L1:PWM5L0 这对寄存器。

在比较模式下, 16 位 PWM5L1:PWM5L0 寄存器的值将不断与 T1H/L 寄存器的值相比较。当两者匹配时, CCP 就会触发相应的事件。

当 CCP 模块配置为 PWM5 模式时, PWM5H0:PWM5L0 为 CCP 模块的占空比寄存器; PP5H: PP5L 为周期寄存器; PWM5H1: PWM5L1 为 PWM5 占空比缓冲寄存器(与用户无关)。

表 10-1 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							
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 通道 1 占空比寄存器低 8 位							
43H	PWM5H0	PWM5 通道 1 占空比寄存器高 8 位							
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 通道 1 低 8 位占空比缓冲寄存器(与用户无关)							
44H	PWM5H1	PWM5 通道 1 高 8 位占空比缓冲寄存器(与用户无关)							
10CH	PWM5L2	PWM5 通道 2 占空比寄存器低 8 位							
11EH	PWM5H2	PWM5 通道 2 占空比寄存器高 8 位							
10DH	PWM5L3	PWM5 通道 3 占空比寄存器低 8 位							
11FH	PWM5H3	PWM5 通道 3 占空比寄存器高 8 位							
57H	PWM5CTL0	P5CH1MOD1	P5CH1MOD0	P5CH3MOD	P5CH2MOD	P5MOD3	P5MOD2	P5MOD1	P5MOD0
119H	PWM5FC	-	-	FCA3	FCB3	FCA2	FCB2	FCA	FCB
11DH	PWM5PC	-	-	PCA3	PCB3	PCA2	PCB2	PCA	PCB
10EH	PWM5OC	-	-	OCA3	OCB3	OCA2	OCB2	OCA	OCB
5BH	PWM5CTL1	P5RSEN	P5DC6	P5DC5	P5DC4	P5DC3	P5DC2	P5DC1	P5DC0

5CH	P5ASCTL0	P5ASE	P5ASS2	P5ASS1	P5ASS0	P5SSAC1	P5SSAC0	P5SSBD1	P5SSBD0
5DH	PSTRCTL0	-	-	-	STRSYNC	STREND	STRENC	STRENB	STRENA
11AH	P5ASCTL1	P5SSA31	P5SSA30	P5SSB31	P5SSB30	P5SSA21	P5SSA20	P5SSB21	P5SSB20
11BH	PSTRCTL1	-	-	-	-	STRENA3	STRENB3	STRENA2	STRENB2
11CH	PWM5CTL2	-	-	PFUSES	-	-	UDEVT1	UDEVT0	UDEN

10.2 捕捉模式

在捕捉模式下，当对应的CCP5IN引脚发生事件时，PWM5L1:PWM5L0这对寄存器捕捉T1H和T1L寄存器的16位值，原理框图如下图所示：

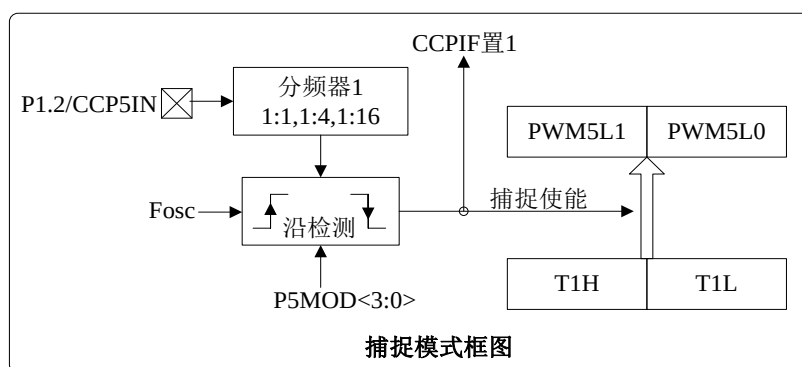


图10.1 捕捉模式原理框图

触发捕捉的事件可被定义为以下四者之一，并且由 PWM5CTL0 寄存器中的模式选择位 P5MOD<3:0>选择事件类型位配置：

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

在捕捉模式下，将 CCP5IN 引脚配置为数字输入。

当一个捕捉发生时，硬件自动中断请求标志位EIF2寄存器中的CCPIF置1；它必须用软件清零。注意如果在PWM5L1和PWM5L0这对寄存器中的值被读取之前发生另一次捕捉，那么之前捕捉的值将被新捕捉的值覆盖。

当捕捉模式改变时，可能会产生错误的捕捉中断。用户应该在捕捉模式改变之前保持 EIE2 寄存器中的 CCPIE 中断允许位清零以避免产生误中断。在捕捉模式发生任何改变之后也应清零 EIF2 寄存器中的中断标志位 CCPIF。

PWM5CTL0 寄存器中的 P5MOD<3:0> 位指定了 4 种预分频器设置。每当关闭 CCP 模块或禁止捕捉模式时，就会清零预分频器计数器。这意味着任何复位都将清零预分频计数器。

虽然从一种捕捉预分频比切换到另一种捕捉预分频比不会将预分频计数器清零，但可能会产生误中断。因此要避免出现这种不期望的操作，应在改变预分频比前通过将 PWM5CTL0 寄存器清零关闭该模块。

注：T1必须运行在定时模式或同步计数模式下CCP模块才能使用捕捉功能。
在异步计数模式下无法进行捕捉操作。

10.3 比较模式

在比较模式下，16位的PWM5L1:PWM5L0寄存器的值将不断与T1寄存器的值相比较。当两者匹配时，CCP模块可能会出现以下几种情况：

- ◆ CCP5OUT引脚的输出电平翻转
- ◆ CCP5OUT引脚输出高电平
- ◆ CCP5OUT引脚输出低电平
- ◆ 产生软件中断触发信号
- ◆ 产生特殊事件触发信号

CCP5OUT引脚的动作取决于PWM5CTL0寄存器中P5MOD<3:0>控制位的值。

所有比较模式都会产生中断。原理图如下图所示：

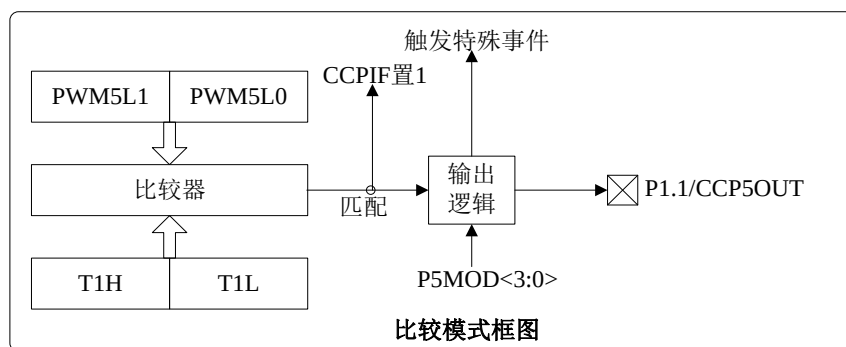


图10.2 比较模式原理框图

用户必须将CCP5OUT引脚配置为数字输出。

在比较模式下，T1必须运行在定时模式或同步计数模式，计数周期至少大于一个机器周期。在异步计数模式下，可能无法进行比较操作。

- 1) 当选择输出电平翻转模式（P5MOD<3:0> = 0010）时，比较匹配时，CCP5OUT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（P5MOD<3:0> = 1010或1001）时，比较匹配时，CCP5OUT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（P5MOD<3:0> = 1010）时，比较匹配时，将CCPIF置1，但是CCP模块不会控制CCP5OUT引脚。
- 4) 当选择了特殊事件触发模式（P5MOD<3:0> = 1011）时，比较匹配时，CCP会立即产生特殊事件触发输出，将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PWM5L1/PWM5L0寄存器实际上成为了定时器1(T1) 的16位可编程周期寄存器。

10.4 PWM5 模式

PWM5 为带有死区控制功能的增强型 PWM 模块。PWM5 有 3 个通道，每个通道都有一个 16 位的占空比设置寄存器。PWM5 的原理框图如下图所示。

PWM5不同通道所对应的占空比设置寄存器

通道1占空比寄存器	PWM5H0	PWM5L0
通道2占空比寄存器	PWM5H2	PWM5L2
通道3占空比寄存器	PWM5H3	PWM5L3

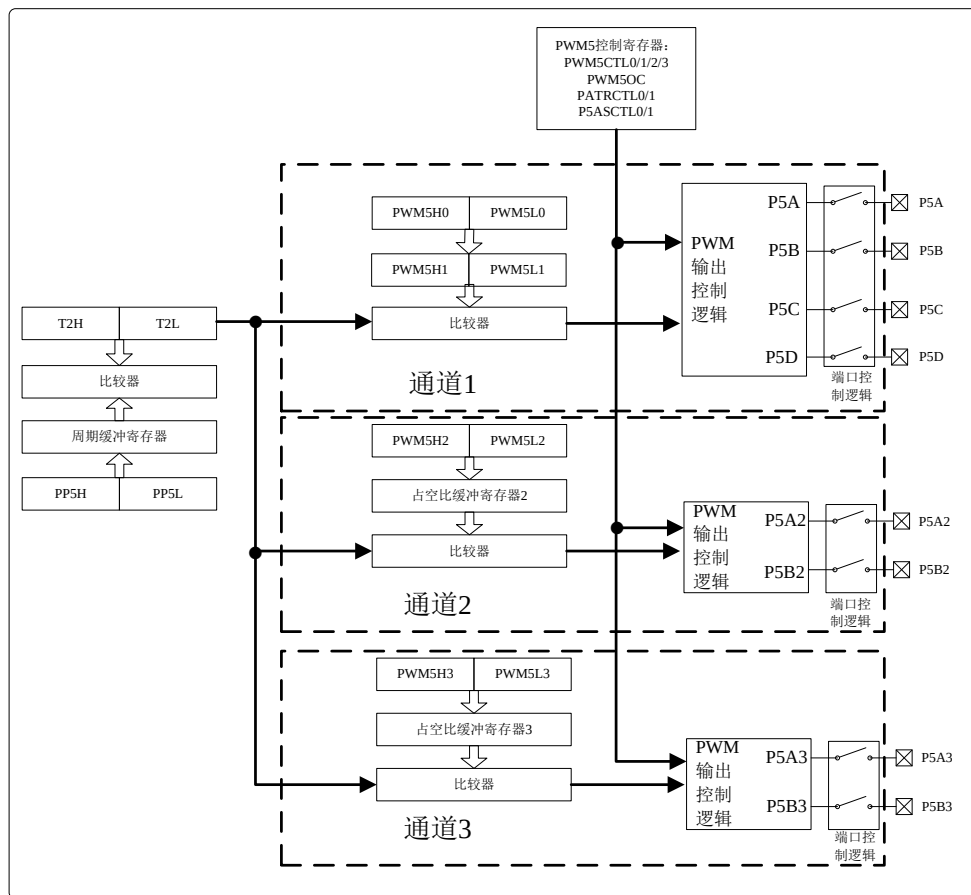


图 10.3 PWM5 原理框图

PWM5 最多可在 8 个不同的引脚输出 PWM 信号，分辨率最高 16 位。其中，通道 1 有 4 个引脚分别为 P5A、P5B、P5C 和 P5D；通道 2 和通道 3 分别有两个输出引脚，分别为 P5A2/P5B3 和 P5A3/P5B3。

PWM5 通道 1 有 4 种输出模式：单输出、半桥输出、全桥正向输出模式和全桥反向输出模式，通过寄存器 PWM5CTL0 中的 P5CH1MOD<1:0>位选择 4 种输出模式之一，通过 P5MOD<1:0>位设置各引脚的有效电平(可设置为高电平有效和低电平有效)。

PWM5 通道 2 和通道 3 一致，都有 2 种输出模式：单输出和半桥输出模式，可以通过 PWM5CTL0 中的 P5CH<3:2>MOD 位选择通道 2 和通道 3 的输出模式。在单输出模式和半桥输出模式下，均可通过 PSTRCTL1 寄存器控制引脚作为 PWM 引脚还是通用 IO 引脚。在半桥模式下，通道 1/2/3 通均可以获得一个带死区控制的互补 PWM 输出。

PSTRCTL0/1 寄存器控制引脚作为 PWM 引脚还是通用 IO 引脚；通过寄存器 PWM5OC 中的 OCA3/ OCB3/ OCA2/ OCB2 位选择通道 2 和通道 3 作为 PWM 输出还是强制输出；通过 PWM5CTL3 中的 PCA3/ PCB3/ PCA2/ PCB2 位可以设置通道 2 和通道 3 输出引脚的极性；通过 PWM5CTL3 寄存器的 FCA3/ FCB3/ FCA2/ FCB2 位可以设置通道 2 和通道 3 引脚作为强制输出时的电平。

注：通道 1 通过选择单输出模式可以选择指定引脚输出 PWM 波形或作为 IO 口；通过选择半桥模式可以获得一个带死区控制的互补 PWM 输出；通过选择全桥模式可以实现 PWM 输出和强制输出同时使用，通过正向和反向模式还可以控制输出的极性，但对 PWM 的输出有限制，只能从 P5B 或 P5D 中输出 PWM 波形。如下表所示。

表 10-2 PWM5 通道 1 输出模式控制及有效引脚

PWM5 输出模式	P5CH1MOD <1:0>	有效引脚
单输出模式	00	默认将 P5A 配置为 PWM 输出，P5B、P5C 和 P5D 配置为端口引脚；可通过 PSTRCTL0 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：各种输出模式的详细介绍请参考本节相应部分。

表 10-3 PWM5 通道 2 输出模式控制及有效引脚

PWM5 输出模式	P5CH2MOD	有效引脚
单输出模式	0	默认将 P5A2 配置为 PWM 输出，P5B2 配置为端口引脚；可通过 PSTRCTL1 寄存器中的 STRENA2 和 STRENB2 各位置 1，分别将 P5A2、P5B2 配置为 PWM 输出。
半桥输出模式	1	P5A2 和 P5B2 配置为调制输出；半桥输出模式带有死区控制功能

表 10-4 PWM5 通道 3 输出模式控制及有效引脚

PWM5 输出模式	P5CH3MOD	有效引脚
单输出模式	0	默认将 P5A3 配置为 PWM 输出, P5B3 配置为端口引脚; 可通过 PSTRCTL1 寄存器中的 STRENA3 和 STRENB3 各位置 1, 分别将 P5A3、P5B3 配置为 PWM 输出。
半桥输出模式	1	P5A3 和 P5B3 配置为调制输出; 半桥输出模式带有死区控制功能

表 10-5 PWM5 通道 1 输出极性控制

P5MOD<3:0>	PCA/PCB	PWM 输出极性
高有效	高有效	高有效
高有效	低有效	低有效
低有效	高有效	低有效
低有效	低有效	高有效

注: 上述满足同或结果, 注意这里的同或与数值 1 和 0 的同或无联系, 这里仅从逻辑关系上去考虑, 例如当通过 P5MOD<3:0>设置输出极性为高有效和通过 PCA/PCB 设置输出为高有效时, 两者同或得实际 PWM 输出极性高有效, 当通过 P5MOD<3:0>设置输出极性为低有效和通过 PCA/PCB 设置输出为高有效时, 两者同或得实际 PWM 输出极性为低有效。

10.4.1 PWM5 相关控制寄存器

10.4.1.1 PWM5CTL0 寄存器

寄存器: PWM5CTL0: PWM5控制寄存器0(地址: 57H)

复位值 0000 0000	bit7				bit0			
	P5CH1MOD D1	P5CH1MOD D0	P5CH3MOD D	P5CH2MOD D	P5MOD3	P5MOD2	P5MOD1	P5MOD0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P5CH1MOD<1:0>: PWM5 通道 1 输出配置位

- 00 = 单输出模式; 默认 P5A 配置为 PWM 输出, P5B、P5C 和 P5D 为端口引脚; 可通过 PSTRCTL 寄存器中的 STREN<A:D>各位置 1, 分别将 P5A、P5B、P5C 和 P5D 配置为 PWM 输出; 通道 1 最多可提供 4 路 PWM 输出
- 01 = 全桥正向输出模式; P5D 配置为 PWM 调制输出, P5A 为有效电平, P5B 和 P5C 为无效电平
- 10 = 半桥输出模式; P5A 和 P5B 配置为调制输出; P5C 和 P5D 被分配为端口引脚, 此模式带有死区控制功能
- 11 = 全桥反向输出模式; P5B 配置为调制输出; P5C 为有效电平; P5A 和 P5D 为无效电平

P5CH3MOD: PWM5 通道 3 输出配置位

- 0 = 单输出模式, 默认为 P5A3 配置为 PWM 输出, P5B3 为端口引脚; 通过 PSTRCTL1 寄存器中的 STRAEN3 和 STRBEN3 两位可以分别设置作为 PWM 引脚或端口引脚。
- 1 = 半桥输出模式, P5A3 和 P5B3 配置为调制输出; 此模式带死区控制。

P5CH2MOD: PWM5 通道 2 输出配置位

- 0 = 单输出模式, 默认为 P5A2 配置为 PWM 输出, P5B2 为端口引脚; 通过 PSTRCTL1 寄存器中的 STRAEN2 和 STRBEN2 两位可以分别设置作为 PWM 引脚或端口引脚。
- 1 = 半桥输出模式, P5A2 和 P5B2 配置为调制输出; 此模式带死区控制。

P5MOD<3:0>: PWM5模块的模式选择位

- 0000 = 捕捉/ 比较/PWM 关闭 (复位 CCP 模块)
- 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, 比较匹配时输出高电平 (CCPIF 置 1)
- 1001 = 比较模式, 比较匹配时输出低电平 (CCPIF 置 1)
- 1010 = 比较模式, 比较匹配时产生软件中断 (CCPIF 置 1, P5A 引脚不受影响)
- 1011 = 比较模式, 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。)
- 1100 = PWM 模式; P5A 和 P5C 为高电平有效; P5B 和 P5D 也为高电平有

效

1101 = PWM 模式； P5A 和 P5C 为高电平有效； P5B 和 P5D 为低电平有效

1110 = PWM 模式； P5A 和 P5C 为低电平有效； P5B 和 P5D 为高电平有效

1111 = PWM 模式； P5A 和 P5C 为低电平有效； P5B 和 P5D 也为低电平有

效

注：通道 1 中 P5A/P5B 极性控制由 PCA/PCB 和 P5MOD<3:0>同或影响（见表:10-5 所示），P5C/P5D 由 P5MOD<3:0>控制。

10.4.1.2 PWM5CTL1 寄存器

寄存器： PWM5CTL1: PWM5控制寄存器1(地址: 5BH)

复位值	bit7						bit0	
0000 0000	P5RSEN	P5DC6	P5DC5	P5DC5	P5DC3	P5DC2	P5DC1	P5DC0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P5RSEN: PWM5 重启使能位

1 = 自动关闭时，一旦关闭事件消失，P5ASE 位自动清零，PWM5 自动重启

0 = 自动关闭时，P5ASE 由软件清零，以重启 PWM5

P5DC<6:0>: PWM5 死区延时时间设置位，用于设置死区延时的时间，见下式：

式 10.1:

$$\text{延时时间} = \text{P5DC} \langle 6:0 \rangle \cdot T_{\text{INTHF}}$$

10.4.1.3 PWM5CTL2 寄存器

寄存器： PWM5CTL2: PWM5更新控制寄存器(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

PFUSES: 由 PFUSES 位控制 PWM 输出

0 = 不由配置字控制 PWM 输出所对应的 IO 口

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

UDEVT1: 更新事件控制 1（在 UDEN=1 时有效）

0 = 当定时器 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

10.4.1.4 PWM5 极性控制寄存器 (PWM5PC)

寄存器: PWM5PC: PWM5极性控制寄存器(地址:11DH)

复位值	bit7						bit0	
--00 0000	-	-	PCA3	PCB3	PCA2	PCB2	PCA	PCB
	U	U	R/W	R/W	R/W	R/W	R/W	R/W

PCA3: 当OCA3为0时, 通道3中P5A3端口输出极性控制

0 = 输出为高电平有效

1 = 输出为低电平有效

PCB3: 当OCB3为0时, 通道3中P5B3端口输出极性控制

0 = 输出为高电平有效

1 = 输出为低电平有效

PCA2: 当OCA2为0时, 通道2中P5A2端口输出极性控制

0 = 输出为高电平有效

1 = 输出为低电平有效

PCB2: 当OCB2为0时, 通道2中P5B2端口输出极性控制

0 = 输出为高电平有效

1 = 输出为低电平有效

PCA: 当OCA为0时, 通道1中P5A端口输出极性控制

0 = 输出为高电平有效

1 = 输出为低电平有效

PCB: 当OCB为0时, 通道1中P5B端口输出极性控制

0 = 输出为高电平有效

1 = 输出为低电平有效

注: 通道 1 中 P5A/P5B 极性控制由 PCA/PCB 和 P5MOD<3:0>同或影响 (见表 10-5 所示), P5C/P5D 由 P5MOD<3:0>控制。

10.4.1.5 PMW5 强制控制寄存器 (PWM5FC)

寄存器: PWM5FC: PWM5强制控制寄存器(地址:119H)

复位值 -00 0000	bit7						bit0	
	-	-	FCA3	FCB3	FCA2	FCB2	FCA	FCB
	U	U	R/W	R/W	R/W	R/W	R/W	R/W

FCA3: 当OCA3为1时, 通道3中P5A3端口强制输出控制

0 = 强制输出为低电平

1 = 强制输出为高电平

FCB3: 当OCB3为1时, 通道3中P5B3端口强制输出控制

0 = 强制输出为低电平

1 = 强制输出为高电平

FCA2: 当OCA2为1时, 通道2中P5A2端口强制输出控制

0 = 强制输出为低电平

1 = 强制输出为高电平

- FCB2: 当OCB2为1时, 通道2中P5B2端口强制输出控制
 0 = 强制输出为低电平
 1 = 强制输出为高电平
- FCA: 当OCA为1时, 通道1中P5A端口强制输出控制
 0 = 强制输出为低电平
 1 = 强制输出为高电平
- FCB: 当OCB为1时, 通道1中P5B端口强制输出控制
 0 = 强制输出为低电平
 1 = 强制输出为高电平

10.4.1.6 PWM5 输出控制寄存器 (PWM5OC)

寄存器: PWM5OC: PWM5输出控制寄存器(地址:10EH)

复位值 --00 0000	bit7						bit0	
	-	-	OCA3	OCB3	OCA2	OCB2	OCA	OCB
	U	U	R/W	R/W	R/W	R/W	R/W	R/W

- OCA3: 通道3中P5A3端口输出控制
 0 = PWM输出, 输出极性由PCA3控制
 1 = 强制输出, 输出电平由FCA3控制
- OCB3: 通道3中P5B2端口输出控制
 0 = PWM输出, 输出极性由PCB3控制
 1 = 强制输出, 输出电平由FCB3控制
- OCA2: 通道2中P5A2端口输出控制
 0 = PWM输出, 输出极性由PCA2控制
 1 = 强制输出, 输出电平由FCA2控制
- OCB2: 通道2中P5B2端口输出控制
 0 = PWM输出, 输出极性由PCB2控制
 1 = 强制输出, 输出电平由FCB2控制
- OCA: 通道1中P5A端口输出控制
 0 = PWM输出, 输出极性由PCA控制
 1 = 强制输出, 输出电平由FCA控制
- OCB: 通道1中P5B端口输出控制
 0 = PWM输出, 输出极性由PCB控制
 1 = 强制输出, 输出电平由FCB控制

10.4.1.7 PWM5 自动关闭控制寄存器 (P5ASCTL0)

寄存器: P5ASCTL0: PWM5自动关闭控制寄存器0(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

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 = INT0 引脚上的逻辑高电平

101 = INT0 引脚上的逻辑高电平或比较器 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

10.4.1.8 P5ASCTL1 寄存器

寄存器: P5ASCTL1: PWM5自动关闭控制寄存器1(地址: 11AH)

复位值 0000 0000	bit7						bit0
	P5SSA31	P5SSA30	P5SSB31	P5SSB30	P5SSA21	P5SSA20	P5SSB21 P5SSB20
	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P5SSA3<1:0>: 引脚 P5A3 关闭状态控制位

00 = 驱动引脚 P5A3 为 0

01 = 驱动引脚 P5A3 为 1

1x = 引脚 P5A3 为三态

P5SSB3<1:0>: 引脚 P5B3 关闭状态控制位

00 = 驱动引脚 P5B3 为 0

01 = 驱动引脚 P5B3 为 1

1x = 引脚 P5B3 为三态

P5SSA2<1:0>: 引脚 P5A2 关闭状态控制位

00 = 驱动引脚 P5A2 为 0

01 = 驱动引脚 P5A2 为 1

1x = 引脚 P5A2 为三态

P5SSB2<1:0>: 引脚 P5B2 关闭状态控制位
 00 =驱动引脚 P5B2 为 0
 01 =驱动引脚 P5B2 为 1
 1x =引脚 P5B2 为三态

10.4.1.9 PSTRCTL0 寄存器

寄存器: PSTRCTL0: 脉冲转向控制寄存器(地址: 5DH)

复位值 ---0 0001	bit7				bit0			
	-	-	-	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 =同步控制输出转向更新, 由 UDEN/UDEVTO 控制更新时机
 0 = 在指令周期边界的开始发生输出转向更新

STREND: 转向使能位 D
 1 = P5D 引脚输出 PWM 波形, 其极性由 P5MOD<1:0>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<1:0>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<1:0>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<1:0>控制
 0 = P5A 引脚被分配为端口引脚

10.4.1.10 PSTRCTL1 寄存器

寄存器: PSTRCTL1: 脉冲转向控制寄存器2(地址: 11BH)

复位值 ---- 1010	bit7				bit0			
	-	-	-	-	STRENA3	STRENB3	STRENA2	STRENB2
	U	U	U	U	R/W	R/W	R/W	R/W

TRENA3: 转向使能位 A3
 1 = P5A3 引脚输出 PWM 波形, 其极性由 P5MOD<1:0>控制
 0 = P5A3 引脚被分配为端口引脚

STRENB3: 转向使能位 B3
 1 = P5B3 引脚输出 PWM 波形, 其极性由 P5MOD<1:0>控制
 0 = P5B3 引脚被分配为端口引脚

STRENA2: 转向使能位 A2
 1 = P5A2 引脚输出 PWM 波形, 其极性由 P5MOD<1:0>控制
 0 = P5A2 引脚被分配为端口引脚

STRENB2: 转向使能位 B2
 1 = P5B2 引脚输出 PWM 波形, 其极性由 P5MOD<1:0>控制



0 = P5B2 引脚被分配为端口引脚

10.4.2 PWM5 的周期、占空比及分辨率

10.4.2.1 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 10.2 和式 10.3 进行计算。

式 10.2:

$$\text{PWM周期}=(\text{<PP5H:PP5L>}+1) T_{\text{INTHF}} (\text{T2预分频比})$$

式 10.3:

$$\text{PWM周期}=(\text{<PP5H:PP5L>}) T_{\text{INTHF}} (\text{T2预分频比}) 2$$

注: T_{INTHF} = 内部高频振荡器时钟周期。

T2 预分频比通过 T2 分频器 1 的配置位 T2CKPS<1:0>位配置。

周期寄存器为带缓冲模式,当 UDEN 位为 0 时,禁止更新占空比寄存器和周期寄存器。当 UDEN 为 1 时,允许更新占空比寄存器和周期寄存器,此时若 UDEV1 为 0,则周期可以在 T2 为 0 时更新到周期缓冲器中;若 UDEV1 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 UDEV1,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。

10.4.2.2 PWM5 占空比

PWM5 有 3 个占空比寄存器,占空比设置寄存器全为 16 位,通过寄存器<PWM5H0:PWM5L0>、<PWM5H2:PWM5L2>和<PWM5H3:PWM5L3>(为后面描述方便,统称为<PWM5Hx:PWM5Lx>)进行设置,PWM5Hx 为占空比的高 8 位,PWM5Lx 为低 8 位。在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 10.4 和式 10.5 计算;在中心对齐模式下,脉冲宽度和占空比通过式 10.6 和式 10.7 计算。

式 10.4:

$$\text{脉冲宽度}=(\text{PWM5Hx:PWM5Lx}) T_{\text{INTHF}} (\text{T2预分频比})$$

式 10.5:

$$\text{占空比} = \frac{\text{脉冲宽度}}{\text{PWM周期}} = \frac{\text{PWM5Hx:PWM5Lx}}{\text{PP5}+1}$$

式 10.6:

$$\text{脉冲宽度} = (\text{PWM5Hx:PWM5Lx}) T_{\text{INTHF}} (T2 \text{ 预分频比}) 2$$

式 10.7:

$$\text{占空比} = \frac{\text{脉冲宽度}}{\text{PWM周期}} = \frac{\text{PWM5Hx:PWM5Lx}}{\text{PP5}}$$

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： (1) 当 T2 启动时占空比寄存器立即更新；
(2) 占空比寄存器的值更新到占空比缓冲器后，才真正完成了占空比寄存器的设置。

10.4.2.3 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 10.8 所示。

式 10.8:

$$\text{分辨率} = \frac{\text{Log}[4(\text{<PP5H:PP5L>}+1)]}{\text{log2}} \text{ 位}$$

10.4.2.4 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。

10.4.3 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<PWM5Hx:PWM5Lx>决定（其中 x=0,2,3）。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<PWM5Hx:PWM5Lx>占空比寄存器发生匹配时，PWM 输出低电平。

如果占空比寄存器设置成 0，那么在整个 PWM 周期都输出低电平。如果占空比寄存器设置大于<PP5H:PP5L>周期寄存器的值，那么在整个 PWM 周期都输出高电平。

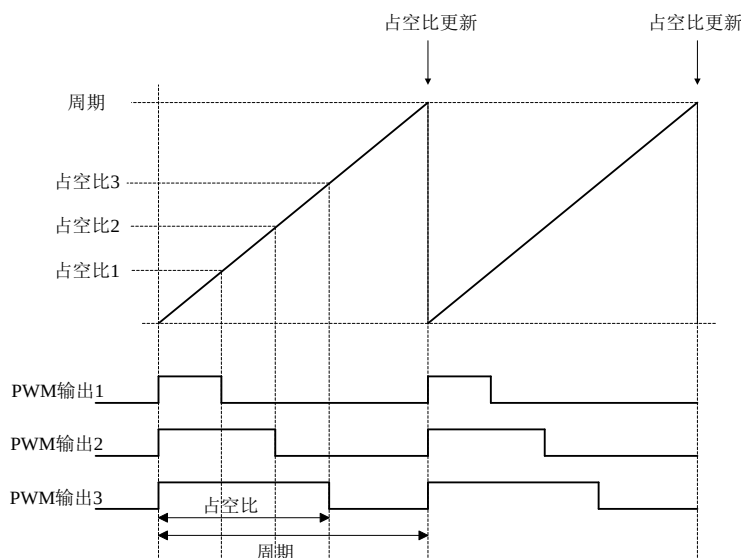


图 10.4 边沿对齐 PWM

10.4.4 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。占空比由 16 位的寄存器<PWM5Hx:PWM5Lx>决定（其中 x=0,2,3）。

在周期开始时，计数器 T2 向上计数，当占空比与 T2 寄存器匹配时，PWM 输出为低电平；当计数器 T2 的值与周期寄存器值的一半匹配时，（应该是周期的一半，或者与周期寄存器的值匹配时）计数器 T2 开始向下计数，而后当占空比与 T2 寄存器匹配时，PWM 输出为高电平。

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。

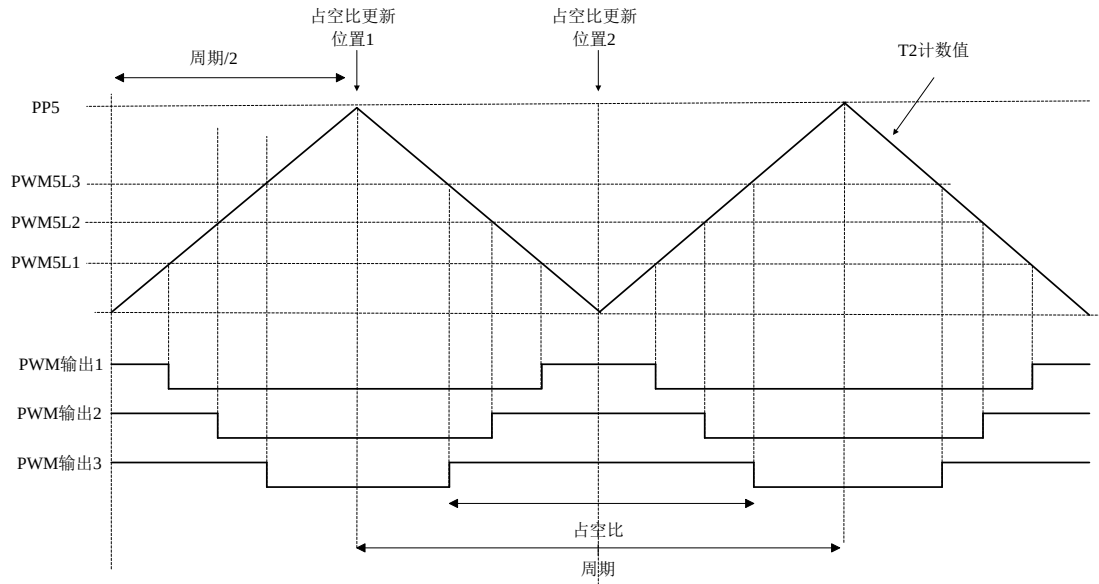


图 10.5 中心对齐 PWM

10.4.5 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。

10.4.6 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。将 PWM5CTL0 中的 P5CH3MOD 和 P5CH3MOD 位均设置为 0，选择通道 2 和通道 3 为单输出模式，在此模式下，默认从 P5A2 和 P5A3 引脚输出 PWM 信号，P5B2 和 P5B3 引脚为通用端口引脚。在使能相应的 PWM5 引脚时，应将对应的 TRxx 位清零，以将此引脚设置为输出模式。

可通过设置寄存器 PSTRCTL0/1，使能或禁止 PWM5 的 8 路输出，PWM5 使用单输出模式时，最多可同时在 8 个引脚输出 PWM 信号。当 8 个引脚同时输出 PWM 信号时，通道

1 的四个引脚 P5A,P5B,P5C,P5D 输出的 PWM 信号完全相同;通道 2 的两个引脚 P5A2,P5B2 输出的 PWM 信号完全相同;通道 3 的两个引脚 P5A3, P5B3 输出的 PWM 信号完全相同。

通道 1,2,3 有各自独立的占空比寄存器故可输出互不相同的 PWM 信号。如图 9.6 所示,在单输出模式下,只将 P5A 一路设置为 PWM 输出和将 P5A 和 P5B 两路同时设置为 PWM 输出的示例,其它设置情况与此类似。

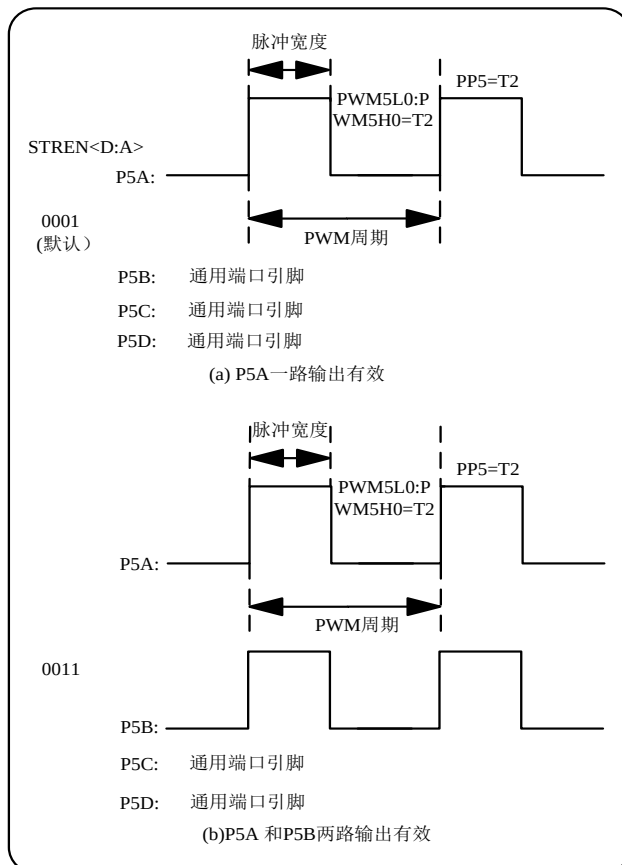


图 10.6 PWM5 的输出示例

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 PSTRCTL0 中的 STRSYNC 位进行设置引脚输出切换时是否与指令同步。如图 9.7 所示,STRSYNC=1 时,脉冲转向控制为同步更新模式。在该模式下,对应引脚的脉冲转向控制受 UDEN/UDEVTO 控制,可以实现同步更新,只有在 UDEN 使能时才能使用所配置的脉冲转向设置,而 UDEVTO 则选择是脉冲转向控制是与周期同步还是立即有效,T2 使能时会产生一次更新操作。STRSYNC=0 时,对应引脚 P5x 输出的 PWM 信号在 STRENx 置一后立即输出;

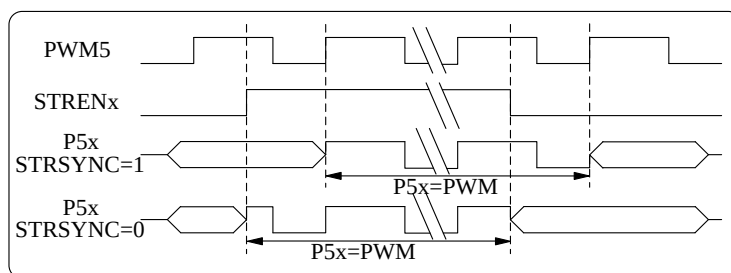


图 10.7 单输出模式 PWM 输出引脚切换

10.4.7 半桥输出模式

通过将寄存器 PWM5CTL0 的 P5CH1MOD<1:0>位设置为 10, 可将通道 1 设置为半桥输出模式; 将 P5CH3MOD 和 P5CH2MOD 置 1, 可将通道 2 和 3 设置为半桥输出模式。在此模式下, 通道 1 的 P5A 和 P5B 被配置为调制输出, 来驱动推挽式负载, P5C 和 P5D 被配置为通用端口。PWM 输出信号在 P5A 引脚上输出, 而互补的 PWM 输出信号在 P5B 引脚上输出, 如图 10.8 所示, 通道 2、3 与此类似。

通道 2 和 3 的在半桥模式下与通道 1 相似, P5A2/P5B2, P5A3/P5B3 被配置为调制输出, PWM 输出信号在 P5A2/3 引脚上输出, 而互补的 PWM 输出信号在 P5B2/3 引脚上输出。

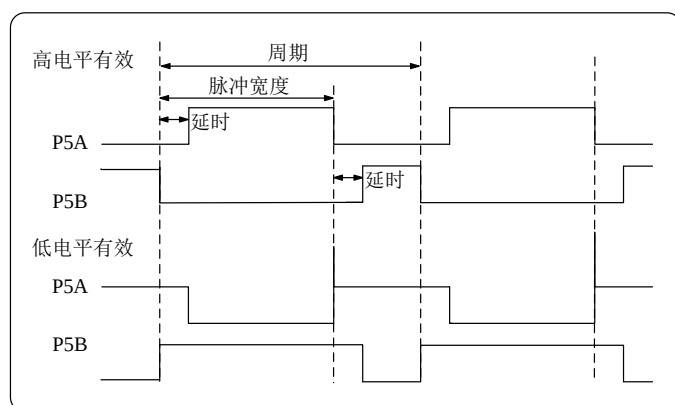


图 10.8 半桥输出模式输出信号示例

半桥输出模式可用于控制半桥和全桥控制电路, 如图 10.9 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。

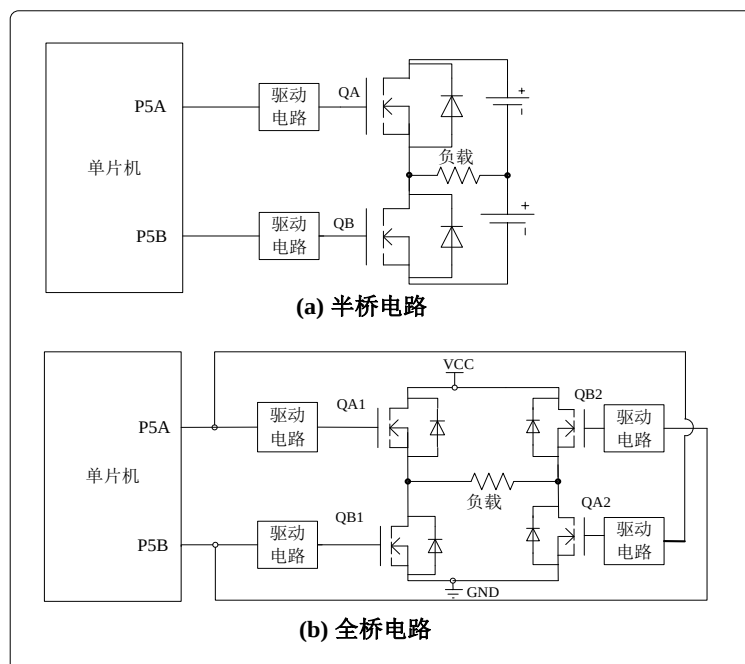


图 10.9 半桥输出模式应用举例

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。

10.4.7.1 死区延时

如图 10.9(a)、10.10 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 10.10(a)中的(1)、(2)、(3)和(4)处所示)。

为了避免开关期间产生这种具有破坏性的直通电流，可使其中一个管子关闭后再打开另一个管子。在半桥输出模式下，使用一个可编程死区延时模块，来避免产生的直通电流破坏电路。3 个通道共用一个死区设置。如图 10.10 (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 10.9 所示。

式 10.9:

$$\text{延时时间} = \text{P5DC} \langle 6:0 \rangle T_{\text{INTHF}}$$

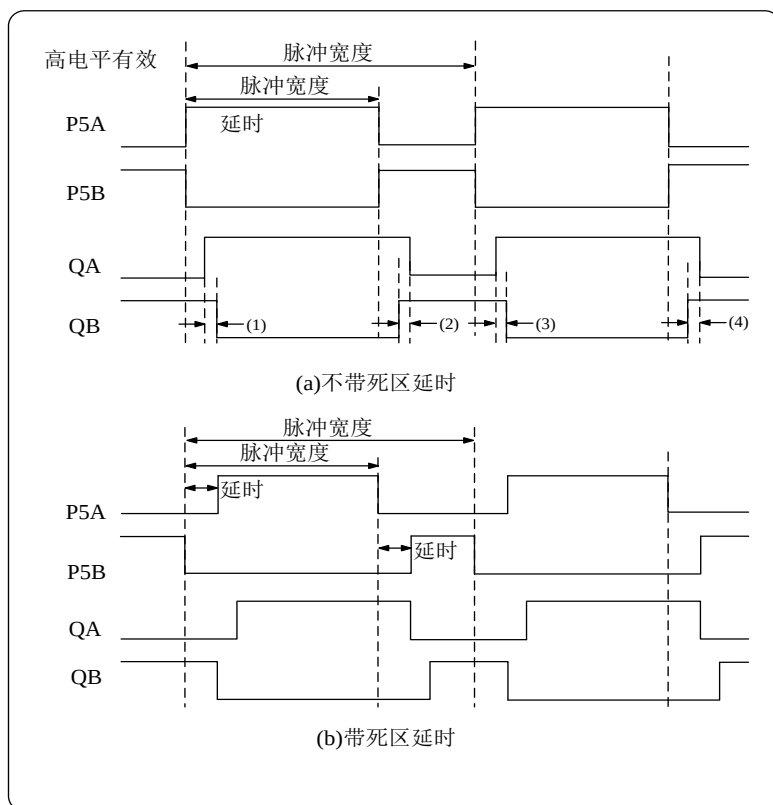


图 9.10 带死区控制和不带死区控制时的信号示例

10.4.8 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<1:0> 设置为 01，把 PWM5 设置为全桥正向输出模式；将其设置为 11，则把 PWM5 设置为全桥反向输出模式。

在全桥输出模式下，P5A、P5B、P5C 和 P5D 四个引脚都用作输出。将其设置为全桥正向输出模式时，引脚 P5A 被设置为有效电平，引脚 P5D 为 PWM 调制信号，P5B 和 P5C 为无效电平，图 10.11(a) 和 10.12(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 10.11(b) 和 10.12(b) 为全桥反向输出模式引脚信号示例。图 10.13 给出了全桥输出模式的应用电路示例。

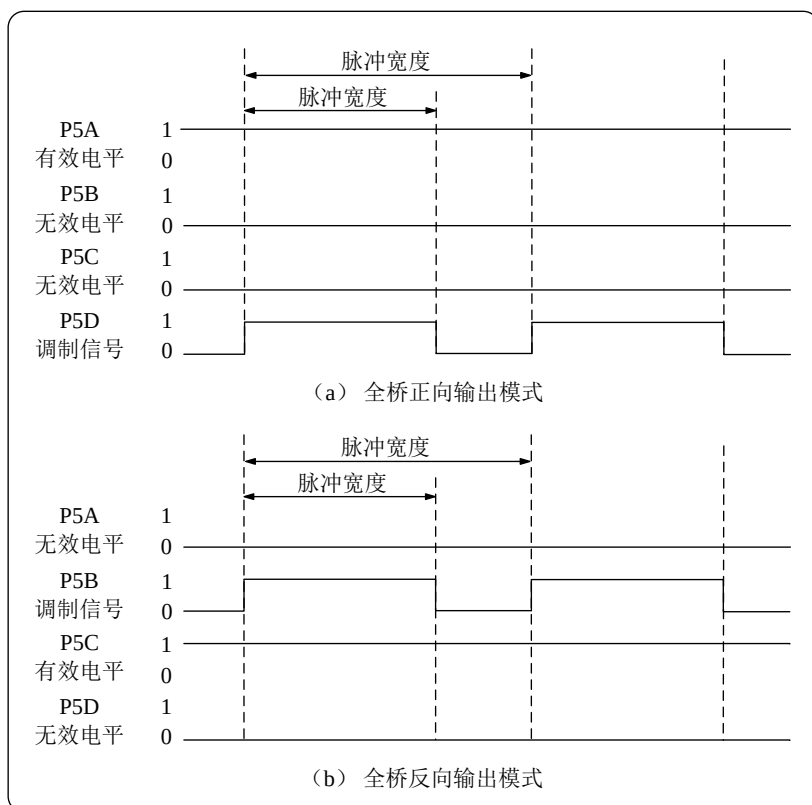


图 10.11 全桥输出模式引脚信号示例(高电平有效)

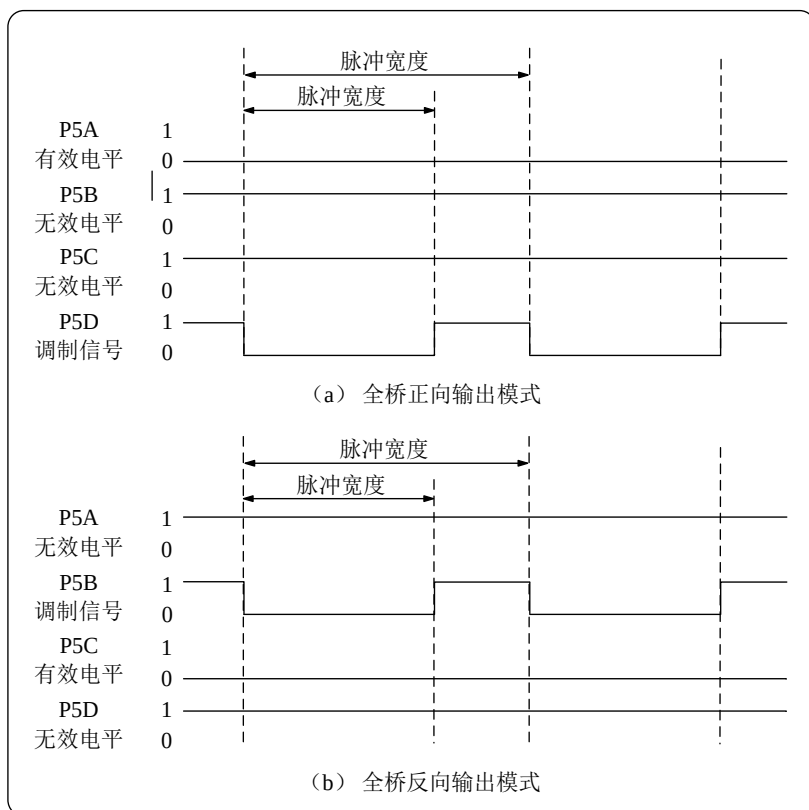


图 10.12 全桥输出模式引脚信号示例(低电平有效)

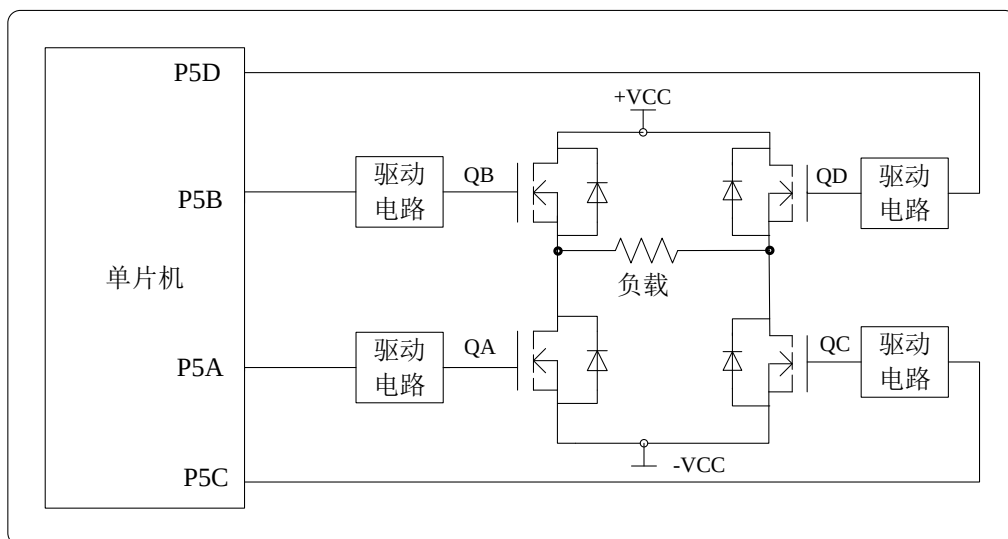


图 10.13 全桥输出模式应用电路示例

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 10.14 所示,从一种模式模式切换到另一种模式时,在前一种模式最后一个周期结束之前,P5B(或 P5D)被切换到无效状态,P5A(或 P5C)被切换到相反的状态。图中时间 $T = 1/F_{osc} \cdot T_2$ 预分频值。

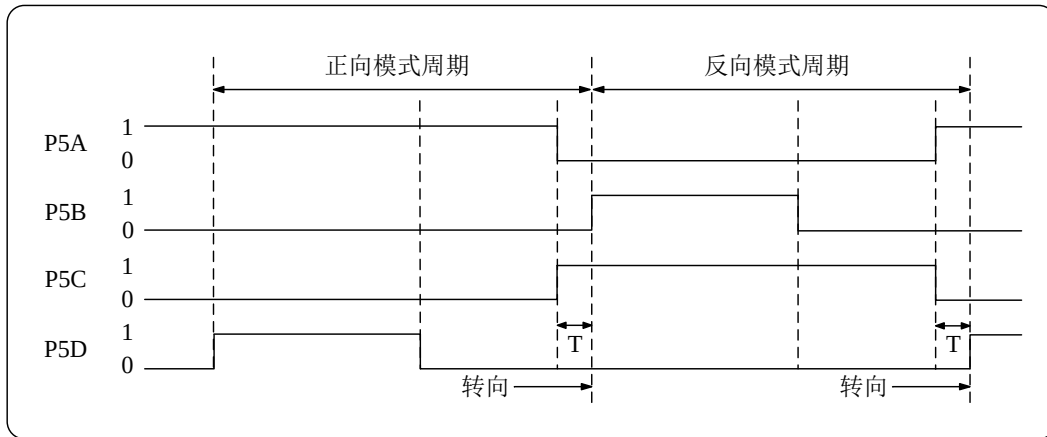


图 10.14 全桥正向、反向输出模式转换示例(高电平有效)

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 10.15 为此情况下各处信号示例。

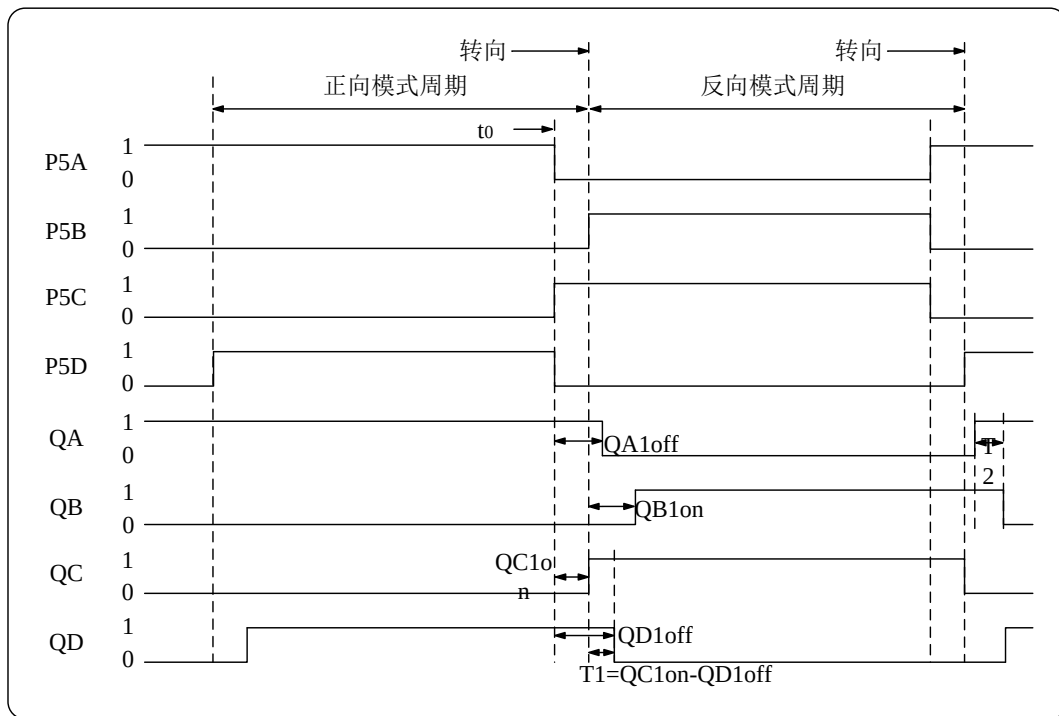


图 10.15 接近满占空比时换向时信号示例

图 10.15 中在 t_0 时刻 P5A 和 P5D 变为无效, P5C 为有效。QA、QB、QC 和 QD 分别为四路 PWM5 输出控制的开关管, QAoff 和 QDoff 为开关管的截止延时时间, QBon 和 QCon 为导通延时时间, 由于管子导通和截止时间的差异, 在第一次换向时, 开关管 QC 和 QD 产生直通电流, 持续时间为 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。

10.4.9 脉冲转向控制

通过寄存器脉冲转向寄存器 PSTRCTL0/PSTRCTL1 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 PSTRCTL0 中的 STRSYNC 位可以控制脉冲转向功能切换是与同步模式还是立即有效 (与指令周期同步)。通过 STRENA/ STRENB/ STRENC/ STREND/ STRENA2/ STRENB2/ STRENA3/ STRENB3 位可以设置相应端口引脚功能切换。默认情况下, 通道 1/2/3 的 Ax 输出为 PWM5 引脚输出, Bx(C/D) 为 IO 口引脚。

注意, 通道 1 的脉冲转向控制只在单输出模式下有效, 而通道 2/3 的在任何模式下都有效, 当使用 PWM5 输出若不需要太多端口时, 可自由屏蔽通道 2/3 的输出而不影响 IO 口功能。

当 STRSYNC 为 0 时, 对应引脚的输出在 STRENx 置 1 后立即更新; 当 STRSYNC 为 1 时, 对应引脚的输出由 UDEN/UDEVTO 控制。只有当 UDEN 为 1 时, 对应引脚的 STRENx 的设置才能生效。当 UDEN 为 1 时, 通过 UDEVTO 可以控制同步更新的时机, 当 UDEVTO 为 0 时, 对应引脚的输出与 PWM 周期同步更新; 当 UDEVTO 为 1 时, 对应引脚的输出立即更新。通过 UDEN/UDEVTO 可以实现脉冲转向的同步更新操作。

10.4.10 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 OCx 设置为 0 时, 相应的引脚选择 PWM 输出, 输出极性由 PWM5PC 极性控制寄存器控制; 当 OCx 设置为 1 时, 相应的引脚选择强制信号输出, 输出电平由 PWM5FC 强制控制寄存器控制。

输出控制为带缓冲模式, 通过 PWM5CTL2 更新控制寄存器可以控制输出控制寄存器缓冲的更新。详见 PWM 更新锁定。

10.4.11 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/ P5A2/ P5B2/ P5A3/ P5B3 输出强制信号。在该模式下, 通过 PWM5FC 强制控制寄存器可以设置 P5A/ P5B/ P5A2/ P5B2/ P5A3/ P5B3 端口输出高电平或低电平, 强制输出不受极性控制的影响。

强制输出控制为带缓冲模式, 通过 PWM5CTL2 更新控制寄存器可以控制强制输出的更新。详见 PWM 更新锁定。

10.4.12 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/ P5A2/ P5B2/ P5A3/ P5B3 输出 PWM 信号。在该模式下, 通过 PWM5PC 极性控制寄存器可以设置 P5A/ P5B/ P5A2/ P5B2/ P5A3/ P5B3 端口输出的极性。

当 PWM 输出极性选择高有效时, P5A/ P5A2/ P5A3 输出在周期开始输出为高电平, 当

占空比与定时器匹配时，输出翻转；而互补的输出 P5B/ P5B2/ P5B3 则在周期开始输出为低电平，当占空比与定时器匹配时，输出翻转。

需要注意的是，通道 1 的 P5A/P5B 有两个极性控制使能，一个是 PWM5CTL0 中的 P5MOD<3:0>，一个是 PWM5PC 中的 PCA/PCB。这两个寄存器相应的控制位共同影响产生最后的极性控制结果。如前述表 9-5 所示。而 P5C/P5D 则只由 P5MOD<3:0>控制。P5A/ P5B/ P5C/ P5B 的极性控制需要在 UDEN 为 1 时才允许更新。

PWM 输出极性控制为带缓冲模式，通过 PWM5CTL2 更新控制寄存器可以控制强制输出的更新。PWM 极性更新有两种形式，一种是立即更新，一种是定时器为 0 时更新。详见 PWM 更新锁定。

10.4.13 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、输出控制和脉冲转向控制的更新。当 PWM5CTL2 寄存器的 UDEN 为 0 时，占空比、周期、极性控制、强制控制和输出控制不能更新，这使得用户能够在新值生效前将所需要的值写入到相应的寄存器中。当 UDEN 为 1 时，可以通过 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 UDEVT0 为 1 时，则立即更新输出控制、极性控制和强制控制寄存器到缓冲中，此时 T2 继续计数，不影响占空比的完整性。

脉冲转向控制中，当 STRSYNC=1 时，脉冲转向更新与极性控制、强制控制、输出控制一致，都是由 UDEVT0 控制，且 UDEN 为更新总使能。当 STRSYNC=0 时，脉冲转向与 UDEN/UDEVT0 无关，当脉冲转向控制 STRENx 设置后输出立即改变。

注意，当 UDEVT1 为 1 时，会清零定时器，并清零死区定时器，此时若 UDEVT0 为 0，也会引起极性控制、强制控制、输出控制和脉冲转向控制的更新。UDEVT0/1 置 1 后必须软件清零再置 1 才能使能下一次的立即更新控制。

10.4.14 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、P5A2/3 的极性，配置 LPOL 可以设置 P5B、P5D、P5B2/3 的极性。

10.4.15 自动关断和自动重启模式

10.4.15.1 自动关断模式

PWM5 模块具有自动关断功能。如图 10.16 所示，为自动关断模式的原理框图。使能自动关断功能后，在外部关断事件发生时，该功能自动禁止 PWM 输出，然后将 P5A、P5B、P5C、P5D、P5A2、P5B2、P5A3、P5B3 八个引脚输出电平置于其预定义的状态。此模式用于防止 PWM 破坏应用电路。

自动关断模式具有 3 个关断源：

- INT0 引脚的逻辑高电平；
- 比较器 1 输出高电平；
- 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。

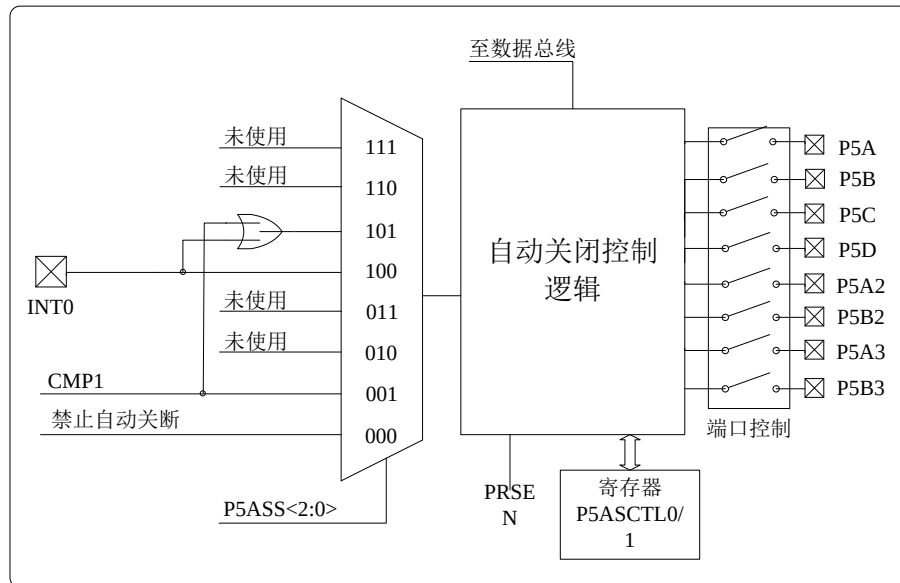


图 10.16 自动关断模式原理框图

通过寄存器 P5ASCTL0 的 P5ASS<2:0>位选择自动关断源。将 P5ASS<2:0>位设置为 000 时，关闭自动关断功能。

寄存器 P5ASCTL0 中的 P5ASE 位指示关断的状态。如果该位为 0，表示 PWM5 的八个引脚输出正常的 PWM 信号，如果该位为 1，表示 PWM 的八路输出处于关断状态。

发生关断事件时，将会：

- 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 使能的八个 PWM 引脚将被置于关断电平状态。

关断时，通道 1 四路输出电平的状态由寄存器 P5ASCTL0 的 P5SSAC<1:0>和 P5SSBD<1:0>位决定。通道 2 和通道 3 输出电平状态由寄存器 P5ASCTL1 的 P5SSA2<1:0>、P5SSB2<1:0>、P5SSA3<1:0>和 P5SSB3<1:0>位决定。通过设定可将输出引脚置为：三态、高电平和低电平三种状态。其中 P5A 和 P5C 的状态由 P5SSAC<1:0>设置，P5B 和 P5D 的状

态由 P5SSBD<1:0>设置, P5A2 状态由 P5SSA2<1:0>设置, P5B2 状态由 P5SSB2<1:0>设置, P5A3 状态由 P5SSA3<1:0>设置, P5B3 状态由 P5SSB3<1:0>设置。

打开自动关断功能后, 如果关断源产生关断事件, 则 P5ASE 标志位被硬件置 1, 四路输出被驱动为关断模式电平; 关断源清除关断事件后, P5ASE 仍然为 1(如果 P5RSEN=0), 四路输出仍然为关断模式电平, 直到将 P5ASE 位清零, PWM 重启。

10.4.15.2 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 10.17 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。

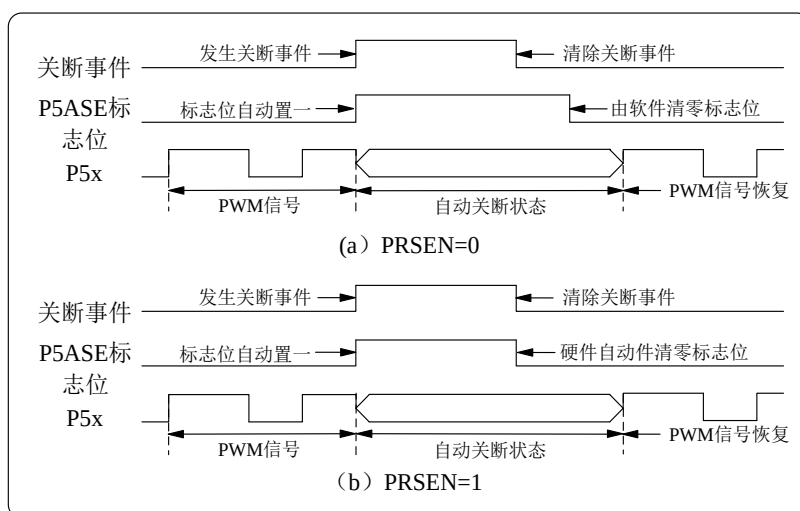


图 10.17 自动重启和软件重启 PWM 示例

11 运算放大器模块

KF8L15Z20XX 具有 1 个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 11.1 所示。

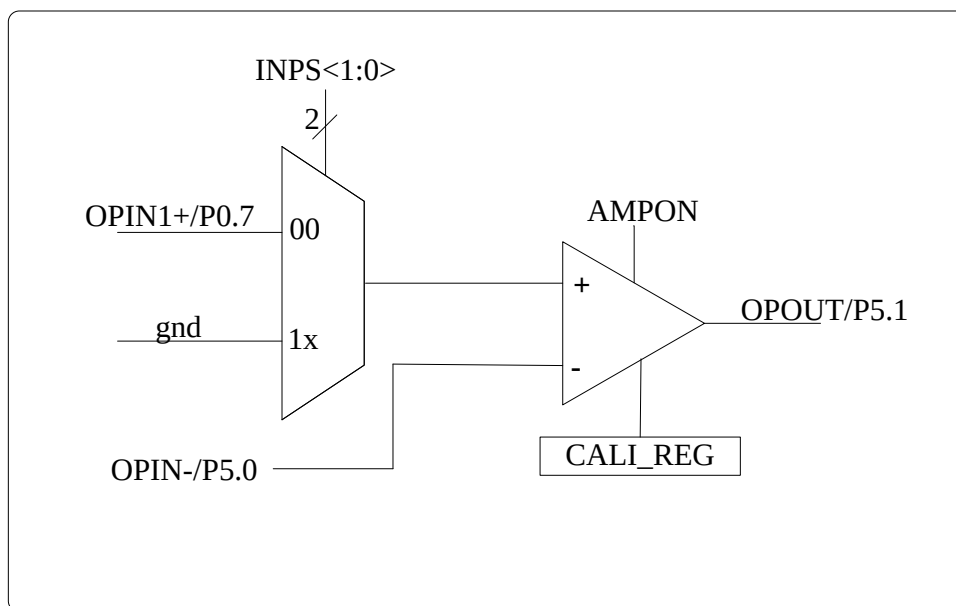


图 11.1 运放内部框图

11.1 与运放有关的寄存器

表 11-1 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	AMPCAL LD	-	INPS1	INPS0	-	-	AMPCAL EN	AMPON
20H	AMPDT	AMPDT 7	AMPDT 6	AMPDT 5	AMPDT 4	AMPDT 3	AMPDT 2	AMPDT 1	AMPDT 0

11.1.1 运放控制寄存器

寄存器： AMPCTL: 运放控制寄存器(1BH)

bit7								bit0	
复位值 1000 0000	AMPCAL D	-	INPS1	INPS0	-	-	AMPCAL EN	AMPON	
	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

AMPCALD: 运放校准输出结果

1 = 运放校准输出为1

0 = 运放校准输出为0

INPS<1:0>: 运放正输入端选择位

00 = P0.7为运放正输入端

01 = 保留

1x = 运放正输入端接地

AMPCALEN: AMP校准使能位

1 = 使能校准

0 = 禁止校准

AMPON: 运放使能位

1 = 运放使能

0 = 运放禁止

11.1.2 运放校准寄存器

寄存器: AMPDT: 运放校准寄存器(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>: 校准数据位

AMPDT = 运放校准数据

11.2 运放使用方法

11.2.1 校准方法

运放失调校准步骤:

1. 设置运放的 IO 端口为模拟端口;
2. AMPON 置 1 打开运放;
3. 延迟 100us, 将 AMPCALEN 置 1 使能运放失调校准功能;
4. 延迟 100us, 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 AMPDT7 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>的每位做步骤 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 0 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。

11.2.2 使用方法

使用运放时软件的设置:

1. 将对应的引脚设置为模拟输入;
2. 选一路为正向输入端
3. 打开运放(AMPON =1)。

注: 1. 运放输出端也要设为模拟输入口

2. 休眠模式下运放被强制关闭不能使用

12 模拟比较器模块

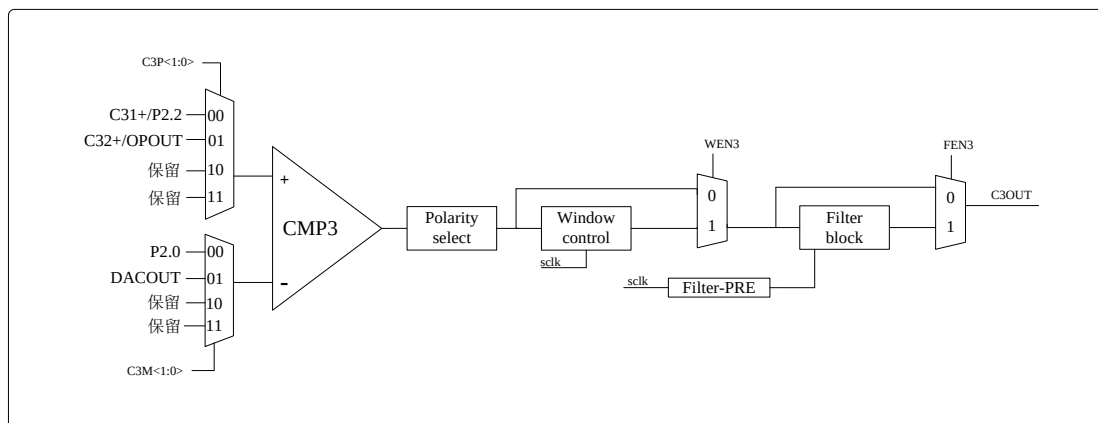
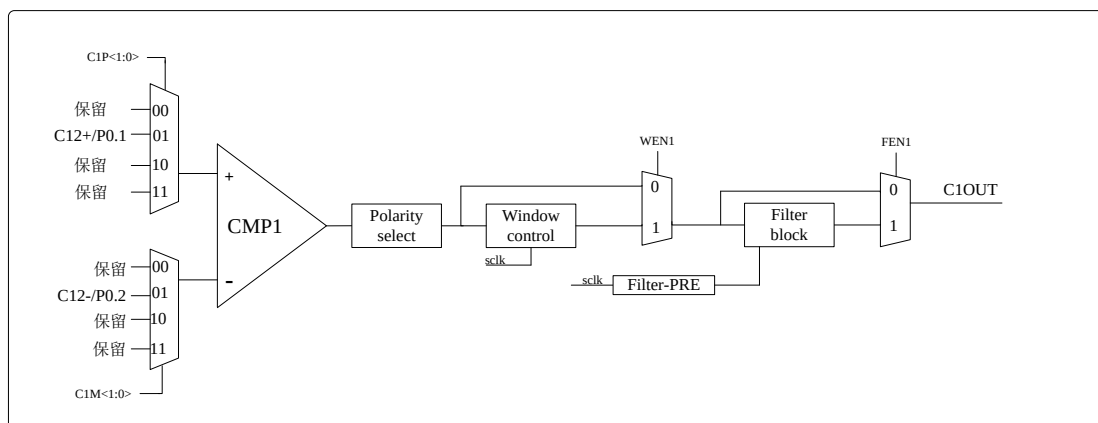
模拟比较器是通过比较两个模拟电压的大小进而输出一个数字量,通常用于模拟电路与数字电路的接口。KF8L15Z20XX 的模拟比较器模块,含有 3 路模拟比较器 CMP1、CMP3 和 CMP4。

该模块包含以下几个特性:

- 带有数字滤波功能
- 多个比较器配置
- 模拟比较器中断
- 可编程输出极性
- 可编程参考电压
- 范围控制功能

12.1 模拟比较器原理

模拟比较器的正端输入为 IO 端口,负端输入可选择 IO 口或者内部电阻分压,用户可通过相应的控制寄存器 CxCTL 进行设置。模拟比较器的框图如图 12.1 所示:



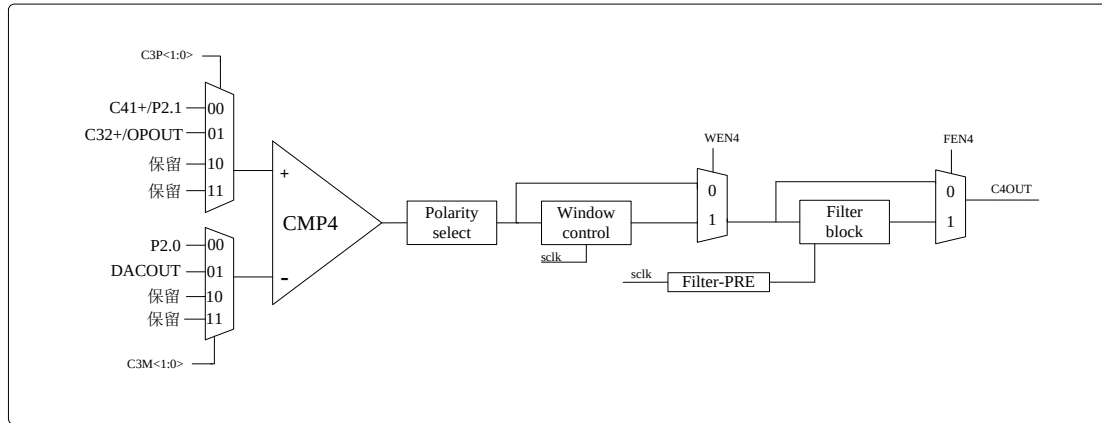


图 12.1 模拟比较器模块框图

12.2 比较器器相关寄存器

表 12-1 与比较器相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
19H	C1CTL	C1EN	C1OE	C1P1	C1P0	-	-	C1M1	C1M0
1AH	COUT	-	VCEN	-	-	C4OUT	C3OUT	C2OUT	C1OUT
113H	C3CTL	C3EN	-	C3P1	C3P0	-	-	C3M1	C3M0
114H	C4CTL	C4EN	-	C4P1	C4P0	-	-	C4M1	C4M0
107H	CMCTL0	C4IMS	C3IMS	C2IMS	C1IMS	C4EDG	C3EDG	C2EDG	C1EDG
16H	CCTCTL	CCTEN	CSEL1	CSEL0	CRSET	CFSET	CCT4	CCT3	CCT1

12.2.1 比较器中断触发控制寄存器 CMCTL0

寄存器: CMCTL0: CMCTL0控制寄存器(地址: 107H)

复位值 0000 0000	bit7						bit0	
	C4IMS	C3IMS	C2IMS	C1IMS	C4EDG	C3EDG	C2EDG	C1EDG
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- C4IMS: 比较器 4 中断触发方式选择位
1=边沿选择触发方式, 通过 C4EDG 位选择触发沿
0=结果变化触发方式, 当比较器 4 输出发生变化时产生中断, C4EDG 位失效
- C3IMS: 比较器 3 中断触发方式选择位
1=边沿选择触发方式, 通过 C3EDG 位选择触发沿
0=结果变化触发方式, 当比较器 3 输出发生变化时产生中断, C3EDG 位失效
- C2IMS: 保留位
- C1IMS: 比较器 1 中断触发方式选择位
1=边沿选择触发方式, 通过 C1EDG 位选择触发沿
0=结果变化触发方式, 当比较器 1 输出发生变化时产生中断, C1EDG 位失效
- C4EDG: 比较器 4 边沿触发中断选择位
1=下降沿触发中断
0=上升沿触发中断
- C3EDG: 比较器 3 边沿触发中断选择位
1=下降沿触发中断
0=上升沿触发中断
- C2EDG: 保留位
- C1EDG: 比较器 1 边沿触发中断选择位
1=下降沿触发中断
0=上升沿触发中断

注: 1.CxEDG 位只在对应的 CxIMS 位为 1 时有效;

2.CxIMS=0 即选择结果变化触发方式时, 对比较器中断标志位的清零操作之前, 都须先对 COUT 寄存器对应的 CxOUT 位进行读操作, 否则无法清零比较器中断标志位。

12.2.2 比较器 1 控制寄存器 C1CTL

寄存器: C1CTL: 比较器1控制寄存器(地址:19H)

复位值 0000 0000	bit7						bit0
	C1EN	C1OE	C1P1	C1P0	-	-	C1M1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W

C1EN: 模拟比较器 1 使能位

0 = 关闭比较器 1

1 = 使能比较器 1

C1OE: 保留位

C1P<1:0>: 比较器 1 的正输入端选择位

00 = 保留

01 = C12+/P0.1 作为 CMP1 的正端输入

10 = 保留

11 = 保留

C1M<1:0>: 模拟比较器 1 的负输入端选择位

00 = 保留

01 = C12-/P0.2 作为 CMP1 的负端输入

10 = 保留

11 = 保留

12.2.3 比较器 3 控制寄存器 C3CTL

寄存器: C3CTL: 比较器3控制寄存器(地址:113H)

复位值 0000 0000	bit7						bit0
	C3EN	-	C3P1	C3P0	-	-	C3M1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W

C3EN: 模拟比较器 3 使能位

0 = 关闭比较器 3

1 = 使能比较器 3

C3P<1:0>: 比较器 3 正输入端选择位

00 = C31+/P2.2 作为 CMP3 的正端输入

01 = C32+/ OPOUT 作为 CMP3 的正端输入

10 = 保留

11 = 保留

C3M<1:0>: 模拟比较器 3 的负输入端选择位

00 = C31-/P2.0 作为 CMP3 的负端输入

01 = C32-/ DACOUT 作为 CMP3 的负端输入

10 = 保留

11 = 保留

12.2.4 比较器 4 控制寄存器 C4CTL

寄存器: C4CTL: 比较器4控制寄存器(地址:114H)

复位值 0000 0000	bit7						bit0
	C4EN	-	C4P1	C4P0	-	-	C4M0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W

C4EN: 模拟比较器 4 使能位

0 = 关闭比较器 4

1 = 使能比较器 4

C4P<1:0>: 比较器 4 正输入端选择位

00 = C41+/P2.1 作为 CMP4 的正端输入

01 = OPOUT 作为 CMP4 的正端输入

10 = 保留

11 = 保留

C4M<1:0>: 模拟比较器 4 的负输入端选择位

00 = C41-/P2.0 作为 CMP4 的负端输入

01 = C42-/ DACOUT 作为 CMP4 的负端输入

10 = 保留

11 = 保留

12.2.5 模拟比较器输出寄存器 COUT

寄存器: COUT: 比较器结果寄存器(地址:1AH)

复位值 0000 0000	bit7						bit0
	-	VCEN	-	-	C4OUT	C3OUT	C1OUT
	R/W	R/W	R/W	R/W	R	R	R

VCEN: 内部电阻分压电路电压源选择位

1=参考电压 VREF

0=供电电压 VDD

C4OUT: 比较器 4 输出

INV4=0 时

1 = C4IN+ > C4IN-

0 = C4IN+ < C4IN-

INV4=1 时

0 = C4IN+ > C4IN-

1 = C4IN+ < C4IN-

C3OUT: 比较器 3 输出

INV3=0 时

1 = C3IN+ > C3IN-

0 = C3IN+ < C3IN-

INV3=1 时

0 = C3IN+ > C3IN-

1 = C3IN+ < C3IN-

C2OUT: 保留位

C1OUT: 比较器 1 输出
 INV1=0 时
 1 = C1IN+ > C1IN-
 0 = C1IN+ < C1IN-
 INV1=1 时
 0 = C1IN+ > C1IN-
 1 = C1IN+ < C1IN-

12.3 滤波器相关寄存器

表 12-2 与比较器相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
14AH	C1FILTCTL	INV1	WEN1	FEN1	-	-	F1CNT2	F1CNT1	F1CNT0
14BH	C1FILTPRE	FP17	FP16	FP15	FP14	FP13	FP12	FP11	FP10
14EH	C3FILTCTL	INV3	WEN3	FEN3	-	-	F3CNT2	F3CNT1	F3CNT0
14FH	C3FILTPRE	FP37	FP36	FP35	FP34	FP33	FP32	FP31	FP30
110H	C4FILTCTL	INV4	WEN4	FEN4	-	-	F4CNT2	F4CNT1	F4CNT0
111H	C4FILTPRE	FP47	FP46	FP45	FP44	FP43	FP42	FP41	FP40

12.3.1 比较器 x 滤波器控制寄存器 CxFILTCTL

寄存器: CxFILTCTL: 比较器x滤波器控制寄存器

复位值 000- -000	bit7					bit0		
	INVx	WENx	FENx	-	-	FxCNT2	FxCNT1	FxCNT0
	R/W	R/W	R/W	U	U	R/W	R/W	R/W

INVx: 比较器 x 输出极性选择位
 0 = CMPx 正常输出: CIN+>CIN-=1, CIN+<CIN-=0
 1 = CMPx 输出极性转换: CIN+>CIN-=0, CIN+<CIN-=1

WENx: 比较器 x 范围控制使能位 window control enable bit
 0 = 禁止范围控制功能
 1 = 使能范围控制功能

FENx: 滤波器 x 使能位
 0 = 禁止滤波功能
 1 = 使能滤波功能

FxCNT<2:0>:滤波器 x 取样数量选择位
 00x = 滤波器 x 采样 1 次, 并输出结果
 010 = 滤波器 x 连续 2 次采样结果一致, 输出采样值, 否则保持;
 011 = 滤波器 x 连续 3 次采样结果一致, 输出采样值, 否则保持;
 100 = 滤波器 x 连续 4 次采样结果一致, 输出采样值, 否则保持;
 101 = 滤波器 x 连续 5 次采样结果一致, 输出采样值, 否则保持;
 110 = 滤波器 x 连续 6 次采样结果一致, 输出采样值, 否则保持;
 111 = 滤波器 x 连续 7 次采样结果一致, 输出采样值, 否则保持;

注: 其中 x = 1/3/4。

12.3.2 比较器 x 滤波器采样时钟分频寄存器 CxFILTPRE

寄存器: CxFILTPRE: 滤波器x采样时钟分频寄存器

复位值 0000 0000	bit7 FPx7	FPx6	FPx5	FPx4	FPx3	FPx2	FPx1	bit0 FPx0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

滤波采样时钟如式 12.1 所示，当 $FP1<7:0>=0$ 时，滤波采样时钟即为滤波时钟源 Filter_Clock。

式 12.1:

$$\text{滤波采样时钟} = \frac{\text{Filter clk}}{FP<7:0>+1}$$

注：其中 $x = 1/3/4$ 。

12.4 极性选择

CxFILTCTL 寄存器的 INV_x 位提供模拟比较器结果的极性选择， $INV_x=0$ ，比较器正常输出： $CxIN+>CxIN-=1$ ， $CxIN+<CxIN-=0$ ； $INV_x=1$ ，比较器输出极性转换： $CxIN+>CxIN-=0$ ， $CxIN+<CxIN-=1$ ，其中 $x = 1、3、4$ 。

12.5 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，使能范围控制功能时，比较器输出可与系统时钟同步，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 寄存器的 WEN_x 位来使能/禁止范围控制功能。

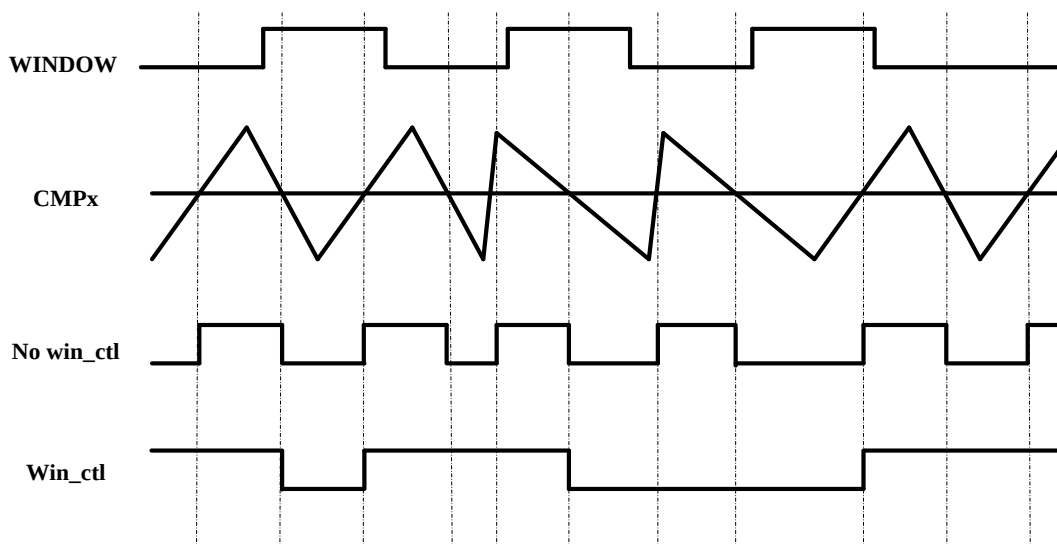


图 12.2 模拟比较器范围控制

12.6 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 寄存器的 FENx 位选择将比较器的输出通过滤波器,可以减少比较器震荡带来的误差。

滤波器使用方法:

- 通过 CxFILTCTL 寄存器的 FENx 位来使能滤波功能;
- 通过 FPx<7:0>位选择滤波器采样时钟;
- 通过 FxCNT1<2:0>位设置滤波器的采样次数。

12.7 比较器使用

使用比较器时需要进行下列设置:

- 1、为相应的引脚设置模拟/数字、输入/输出状态;
- 2、设置相应寄存器的控制位来为模拟比较器选择工作模式;
- 3、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。

12.8 比较器清零定时器

KF8L15Z20XX 的比较器带有硬件清零定时器 (T1/T3/T4) 的计数寄存器功能。用户可通过 CCTCTL 寄存器进行相关配置。

12.8.1 比较器清零定时器控制寄存器 CCTCTL

寄存器: CCTCTL: 比较器清零定时器控制寄存器(地址:16H)

复位值 0000 0000	bit7						bit0	
	CCTEN	CSEL1	CSEL0	CRSET	CFSET	CCT4	CCT3	CCT1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- CCTEN:** 比较器清零定时器功能使能位
 0 = 禁止比较器清零定时器功能
 1 = 使能比较器清零定时器功能
- CSEL<1:0>:** 比较器选择位
 00 = 选择比较器 1
 01 = 保留
 10 = 选择比较器 3
 11 = 选择比较器 4
- CRSET:** 比较器输出上升沿清零定时器配置位
 0 = 禁止比较器输出上升沿清零定时器
 1 = 使能比较器输出上升沿清零定时器
- CFSET:** 比较器输出下降沿清零定时器配置位
 0 = 禁止比较器输出下降沿清零定时器
 1 = 使能比较器输出下降沿清零定时器

CCT4:	比较器清零 TMR4 计数寄存器使能位
	0 = 禁止比较器清零 TMR4 计数寄存器 T4H/T4L
	1 = 使能比较器清零 TMR4 计数寄存器 T4H/T4L
CCT3:	比较器清零 TMR3 计数寄存器使能位
	0 = 禁止比较器清零 TMR3 计数寄存器 T3H/T3L
	1 = 使能比较器清零 TMR3 计数寄存器 T3H/T3L
CCT1:	比较器清零 TMR1 计数寄存器使能位
	0 = 禁止比较器清零 TMR1 计数寄存器 T1H/T1L
	1 = 使能比较器清零 TMR1 计数寄存器 T1H/T1L

12.8.2 使用方法

1. 配置 CRSET 位和 CFSET 位选择比较器输出的上升沿、下降沿作为定时器的清零点；CRSET 和 CFSET 位可同时置 1, 即比较器输出的上升沿和下降沿均可清零定时计数器；
2. 配置 CSEL<1:0>位选择一个比较器，配置好该比较器并使能开始工作；
3. 配置 CCT1/CCT3/CCT4 位，选择需要被比较器清零的定时器；允许同时使能，即 CCT1、CCT3、CCT4 可同时配置为 1，当比较器翻转时，T1H/T1L、T3H/T3L、T4H/T4L 均被清零；
4. 将 CCTEN 位置 1，使能比较器清零定时器功能。

注：比较器的输出和比较器输出极性选择位相关，请参见 CxFILTCTL 寄存器的 INVx 位。

13 SSCI 模块

13.1 概述

KF8L15Z20XX 包含一个 SSCI（Synchronous Serial Communication interface）同步串行端口。它是用于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 Integrated Circuit）接口模式。
- ◆ 串行外设接口（Serial Peripheral Interface，SPI）模式

13.2 SSCI 相关寄存器

表 13-1 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
53H	PINSET	-	INT1PIN	-	-	-	-	-	SSCIPIN
128H	SSCICTL0	SSCIWCFL	SSCIOV	SSCIEN	SSCICKP	SSCIMOD3	SSCIMOD2	SSCIMOD1	SSCIMOD0
12AH	SSCICTL1	SSCICALLEN	SSCIACKSTA	SSCIACKDAT	SSCIACKEN	SSCIRCEN	STOPEN	RESTARTEN	STARTEN
12BH	SSCISTA	SAMPLE	CKEGE	SSCIDA	SSCISTOP	SSCISTART	SSCIRW	SSCIUA	SSCIBUF
12CH	SSCIBUFR	SSCI 数据接收缓冲/发送寄存器							
12EH	SSCIADD	SSCI 的 I2C 地址寄存器							
	SSCIMSK	SSCIMSK7	SSCIMSK6	SSCIMSK5	SSCIMSK4	SSCIMSK3	SSCIMSK2	SSCIMSK1	SSCIMSK0

13.2.1 SSCI 引脚配置寄存器（PINSET）

寄存器： PINSET: SSCI引脚配置寄存器(地址:53H)

	bit7							bit0
复位值 0000 0000	-	INT1PIN	-	-	-	-	-	SSCIPIN
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

INT1PIN: INT1 引脚配置位

1 = INT1 引脚位于 P4.1

0 = INT1 引脚位于 P5.2

SSCIPIN: SSCI 引脚配置位

0 = 引脚未分配给SSCI

1 = SSCI相关功能引脚位置如下：

SCK/SCL = P0.2

SDI/SDA = P0.1

SDO = P5.2

SS = P6.4

注：在使用 SSCI 模块之前，请先将 SSCIPIN 位置 1。

13.2.2 SSCI 控制寄存器 0 (SSICCTL0)

寄存器: SSICCTL0: SSCI控制寄存器0(地址:128H)

	bit7				bit0			
复位值	SSCI WCFL	SSCIOV	SSCIEN	SSCICKP	SSCIMOD 3	SSCIMOD 2	SSCIMOD 1	SSCIMOD 0
0000 0000	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。 在主动模式下, 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 主动模式, 时钟 = 工作时钟/4
- 0001 = SPI 主动模式, 时钟 = 工作时钟/16
- 0010 = SPI 主动模式, 时钟 = 工作时钟/64
- 0011 = SPI 主动模式, 时钟 = T2输出/2
- 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=FOSC/(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

13.2.3 SSCI 控制寄存器 1（SSCICTL1）

寄存器： SSCICTL1: SSCI控制寄存器1(地址:12AH)

复位值 0000 0000	bit7						bit0
	SSPCALL EN	SSCIACK STA	SSCIACK DAT	SSCIACK EN	SSCIRCE N	STOPEN	RESTART EN
	R/W	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ACKDT 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

在从动模式下:

1=从发送和接收都会使能时钟延长（使能时钟延长）

0=禁止时钟延长

注:

对于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）。

13.2.4SSCI 状态寄存器（SSCISTA）

寄存器: SSCISTA: SSCI状态寄存器(地址:12BH)

	bit7				bit0			
复位值 0000 0000	SAMPLE	CKEGE	SSCIDA	SSCISTOP	SSCISTA RT	SSCIRW	SSCIUA	SSCIBUF
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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。

1 = 读

0 = 写

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

13.2.5 SSCI 屏蔽寄存器（SSCIMSK）

寄存器: SSCIMSK: SSCI屏蔽寄存器(地址:12EH)

bit7								bit0
复位值	SSCIMSK	SSCIMSK	SSCIMSK	SSCIMSK	SSCIMSK	SSCIMSK	SSCIMSK	SSCIMSK
1111 1111	7	6	5	4	3	2	1	0
	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

注：SSCIADD与SSCIMSK共用一个地址，当SSCICTL0位SSCIMOD<3:0>=1001时，SFR地址对应SSCIMSK寄存器；当SSCICTL0位SSCIMOD<3:0>不为1001时，SFR地址对应SSCIADD寄存器。

13.3 I2C 模式

- ◆ 多主机模式:可用作主设备或者从设备
- ◆ I2C 主设备产生时钟，起始和停止信号
- ◆ 检测 7 位和 10 位地址

13.3.1 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。有两个引脚用于数据传输: P1.0/SCK/SCL引脚作为时钟线（SCL），而P1.1/SDI/SDA引脚作为数据线（SDA）。通过将SSCI使能位SSCIEN（SSCICTL0<5>）置1以使能SSCI模块的功能。

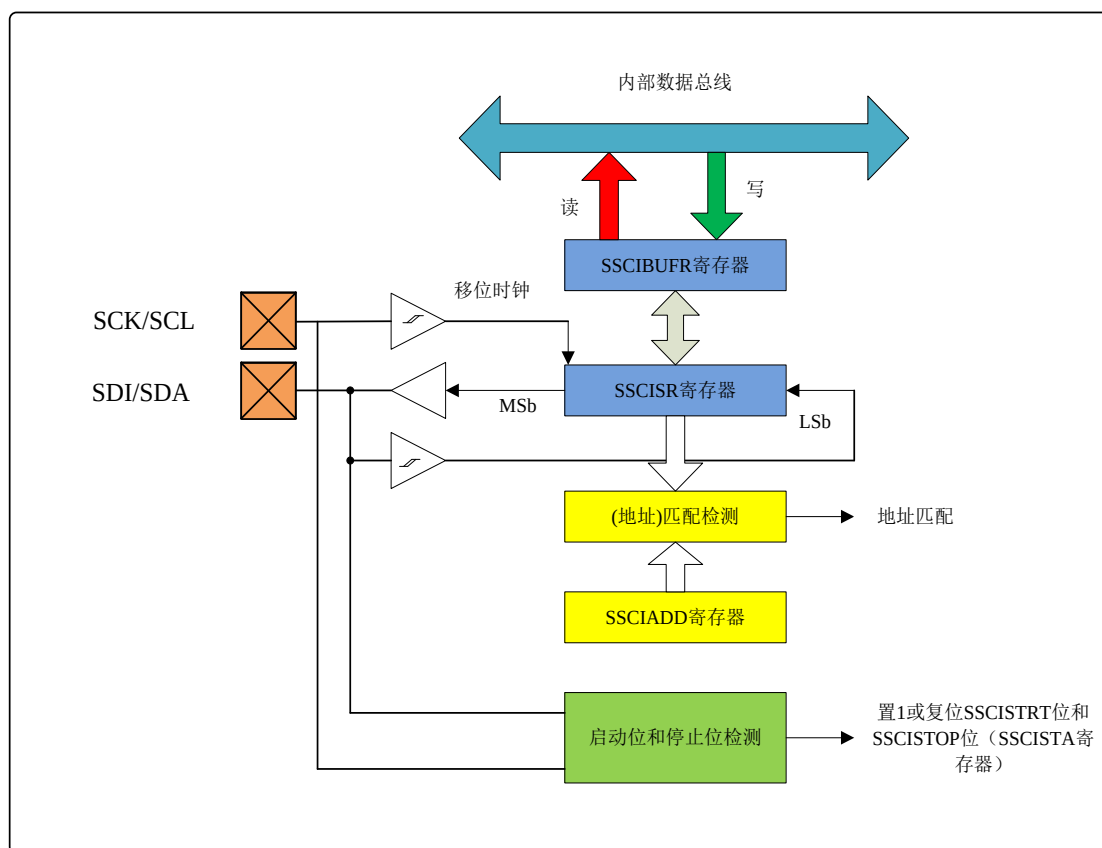


图13.1 I2C模式方框图

SSCI模块有7个寄存器用于I2C操作，这7个寄存器是：

- SSCI控制寄存器（SSCICTL0）
- SSCI控制寄存器1（SSCICTL1）
- SSCI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）

SCICTL0 寄存器用于控制I2C 的工作。可通过设置四个模式选择位（SSCCTL0<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IO口方向寄存器TRx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。

13.3.2 I2C 从动模式

在从动模式下，SCL 引脚和SDA 引脚必须被配置为输入（TR1<1:0> 置1）。必要时SSCI 模块将用输出数据改写输入状态（从发送器）。

当地址匹配或在地址匹配后发送的数据被接收时，硬件会自动产生一个应答（ACK）脉冲，并把当时SSCISR寄存器中接收到的值装入SSCIBUFR寄存器。

某些条件会使SSCI 模块不发出此ACK（低电平有效）脉冲。这些条件包括（之一或全部）：

- 1) 在接收到数据前，缓冲器满标志位SSCIBUF（SSCISTA<0>）置1。
- 2) 在接收到数据前，溢出标志位SSCIOV（SSCCTL0<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12-2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 13-2 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。

13.3.2.1寻址

一旦SSCI模块被使能，它就会等待启动条件发生。在7位地址模式下，当启动条件发生后，8位数据被移入SSCISR寄存器。在时钟（SCL）线的上升沿采样所有的输入位。在第8个时钟（SCL）脉冲的下降沿寄存器SSCISR<7:1>的值会和SSCIADD地址寄存器的值比较。如果地址匹配，并且SSCIBUF和SSCIOV都被清零，会发生下列事件：

- 1) SSCISR寄存器的值被装入SSCIBUFR寄存器。
- 2) 缓冲器满标志位SSCIBUF被置1。
- 3) 产生ACK脉冲。

- 4) 在第9个SCL脉冲的下降沿，SSCI中断标志位SSCIIF被置1（如果允许中断，则产生中断）。

在10位地址模式下，从控制器需要收到两个地址字节（图12.3）。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。

13.3.2.2接收

当地址字节的SSCIRW状态位清零并发生地址匹配时，SSCISTA寄存器中的SSCIRW位清零。接收到的地址被装入SSCIBUFR寄存器。

当发生地址字节溢出时，则不会产生应答脉冲（ACK）。溢出条件是指SSCIBUF位置1，或者SSCIOV位（SSCICTL0<6>）置1。这是一个由于用户固件导致的错误状态。

每个数据传输字节都会产生SSCI中断。标志位SSCIIF必须用软件清零。通过SSCISTA寄存器可以确定该字节的状态。

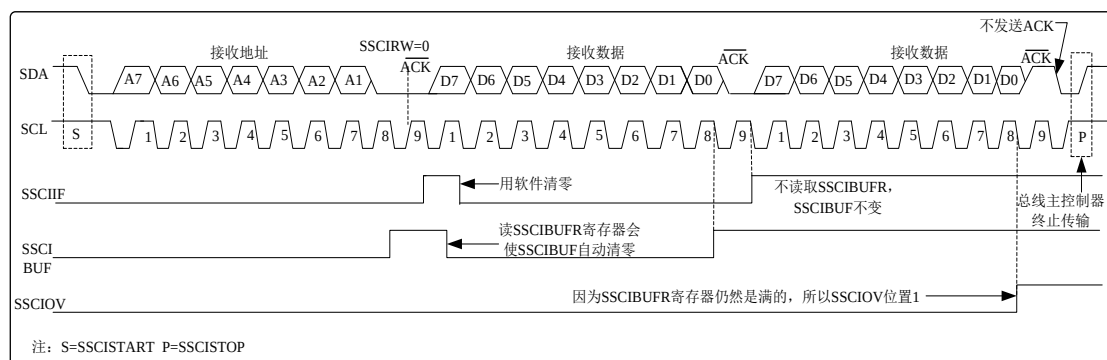


图13.2 从动模式时序（接收，7位地址）

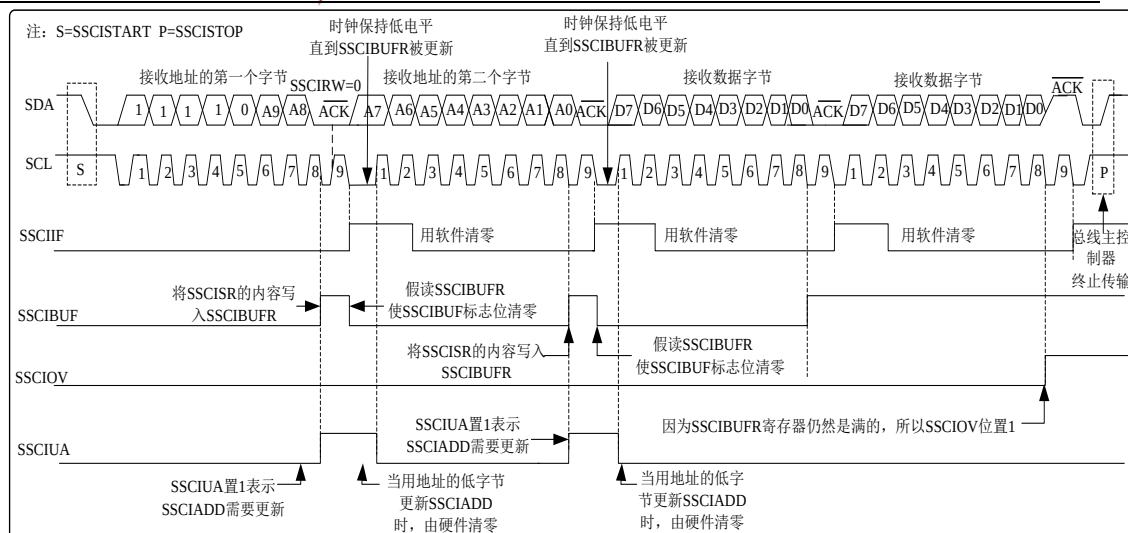


图 13.3 从动模式时序（接收，10 位地址）

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

13.3.2.3 发送

当输入地址字节的 SSCIRW 位置 1 并发生地址匹配时，SSCISTA 寄存器的 SSCIRW 位被置 1。接收到的地址被装入 SSCIBUF 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 SSCIBUF 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 输入的下沿被移出。这可以确保在 SCL 为高电平期间 SDA 信号是有效的。

每个数据传输字节都会产生 SSCI 中断。标志位 SSCIIF 必须用软件清零，SSCISTA 寄存器用于确定字节的状态。标志位 SSCIIF 在第 9 个时钟脉冲的下沿被置 1。对于从发送器，来自主接收器的 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 SSCIBUF 寄存器。然后，通过

将SSCICKP位（SSCICTL0<4>）置1使能SCL引脚。

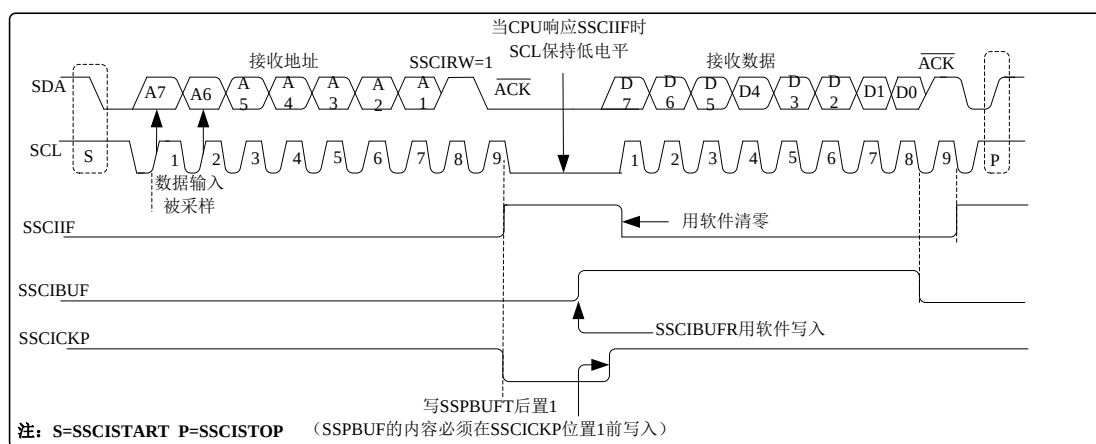


图 13.4 从动模式时序（发送，7 位地址）

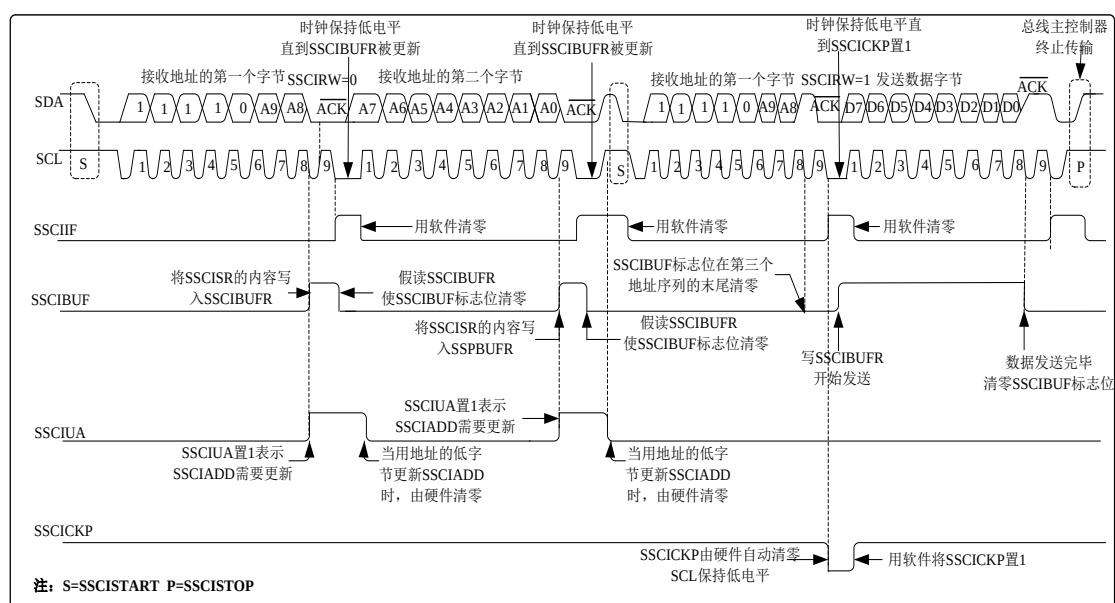


图 13.5 从动模式时序（发送，10 位地址）

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件事 SCL 引脚。从控制器可以通过延长时钟低电平时间不予主控制器同步。

9、标志位 SSCIIF 在第 9 个时钟脉冲的下降沿被置 1。软件清零

10、对于从发送器，来自主接收器的ACK脉冲将在第9个SCL输入脉冲的上升沿被锁存。若SDA线为高电平（无ACK应答信号），则表示数据传输已完成。在这种情况下，如果从控制器锁存了ACK，将复位从动逻辑（复位SSCISTA寄存器），同时从控制器监视下一个启动位的出现。如果SDA 线为低电平（ACK），则必须将下一个要发送的数据装入SSCIBUFR 寄存器。然后，通过将SSCICKP位（SSCICTL0<4>）置1使能SCL引脚。

13.3.2.4广播呼叫地址支持

在I2C 总线的寻址过程中，通常由启动条件后的第一个字节决定主器件将寻址哪个从器件。但广播呼叫地址例外，它能寻址所有器件。当使用这个地址时，理论上所有的器件都应该发送一个应答响应。

广播呼叫地址是根据I2C协议为特定目的保留的八个地址之一。它由全0 组成，且 SSCIRW = 0。广播呼叫使能位SSCICALLEN（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13.6所示。

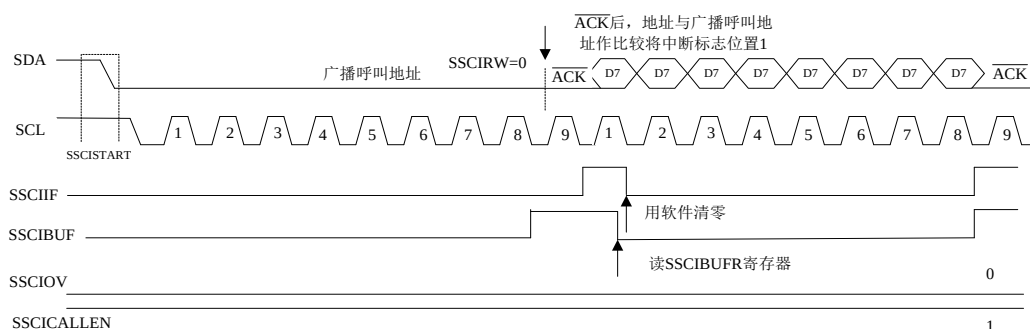


图13.6 从动模式广播呼叫地址时序（7或10位地址模式）

13.3.3 I2C 主动模式

主控模式通过固件在检测到启动条件和停止条件时产生中断来工作。停止（SSCISTOP）位和启动（SSCISTART）位在复位时或禁止SSCI模块时清零。停止（SSCISTOP）位和启动（SSCISTART）位会根据启动和停止条件翻转。当SSCISTOP位置1时，可以获得I2C总线的控制权；否则，停止（SSCISTOP）位和启动（SSCISTART）位都清零，总线处于空闲状态。

在主动模式下，SCL和SDA线通过清零相应的TR1<1:0>位来控制。输出电平始终为低电平，而与P1.<1:0>的值无关。因此当发送数据时，对于SDA线，必须将TR11置0（输出），对于SCL线，也要将TR1<0>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动模式操作。当同时使能主动模式和从动模式时，需要使用软件区分中断源。

13.3.3.1 主动模式支持

通过设置SSCICTL0中的SSCIMOD<3:0>并将SSCIEN位置1可使能主动模式。一旦使能主动模式，

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主动模式时，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13.7 为I2C主动模式框图。

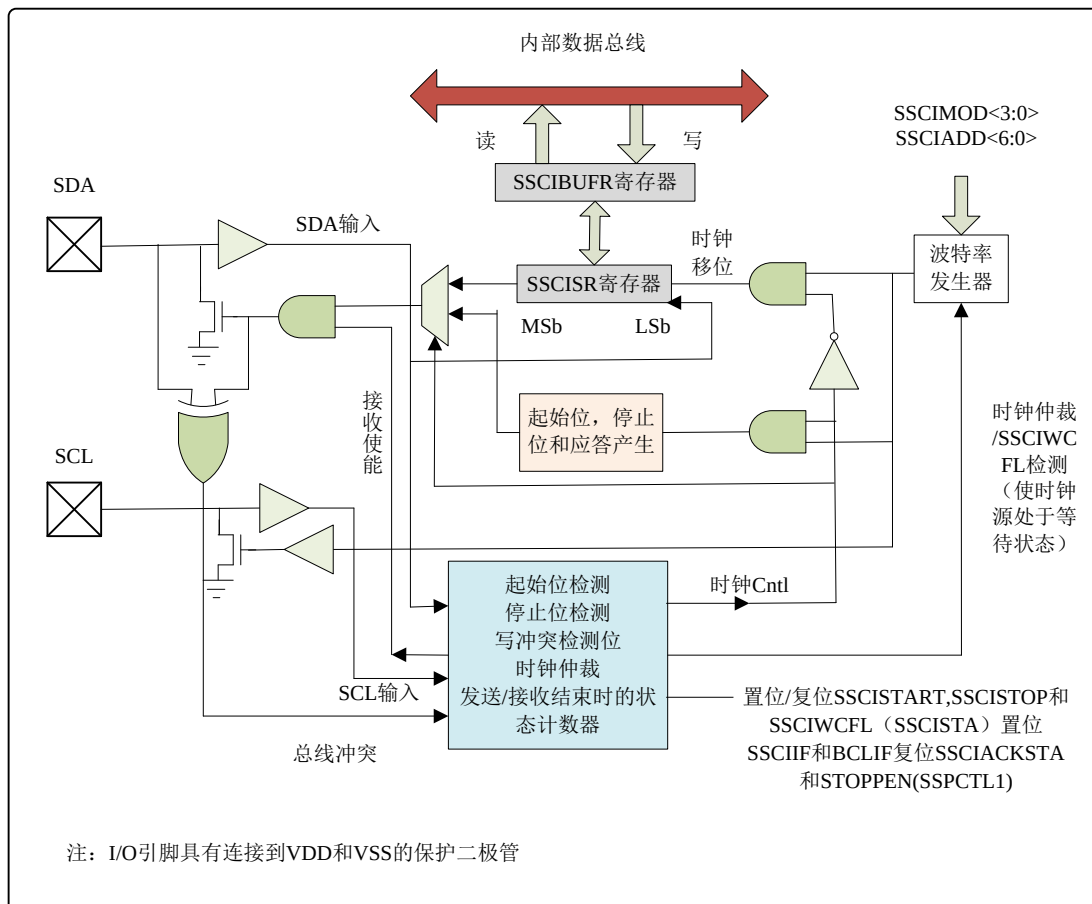


图13.7 I2C主模式框图

13.3.3.2 I2C 主模式操作

所有串行时钟脉冲和启动/ 停止条件均由主器件产生。停止条件或重复启动条件能结束传输。因为重复启动条件也是下一次串行传输的开始，因此不会释放I2C总线。在主机发送器模式下，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。

13.3.3.3波特率发生器

在I2C主控模式下，波特率发生器的重载值位于SSCIADD 寄存器的低7位。当装载了该值后，波特率发生器将自动开始计数并递减至0，然后停止直到下次重载为止。BRG 会在每个指令周期（TCY）中的Q2 和Q4 时钟周期上进行两次减计数。在I2C 主控模式下，会自动重载BRG。例如，在发生时钟仲裁时，BRG 将在SCL 引脚采样到高电平时重载。如图13.8和图13.9 所示。

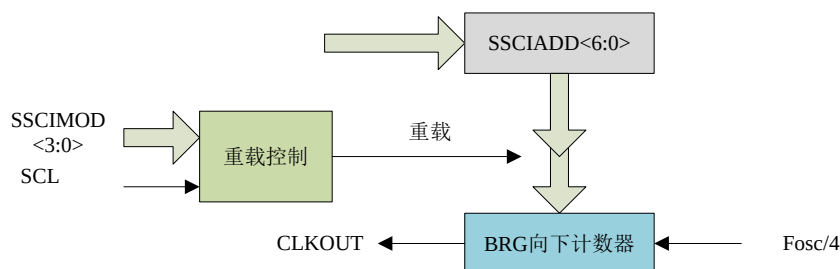


图 13.8 波特率发生器框图

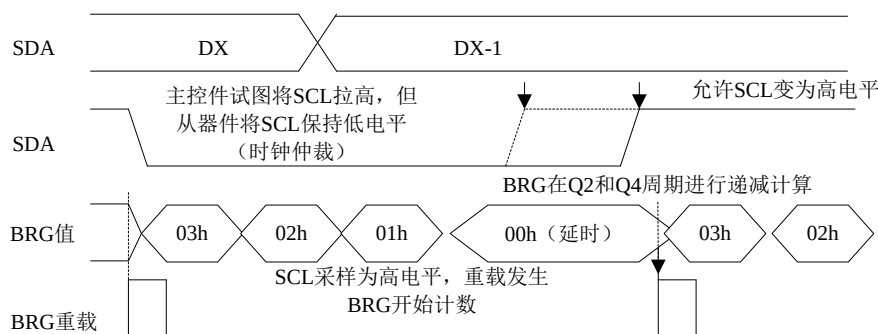


图 13.9 带有时钟仲裁的波特率发生器时序

13.3.3.4I2C 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。

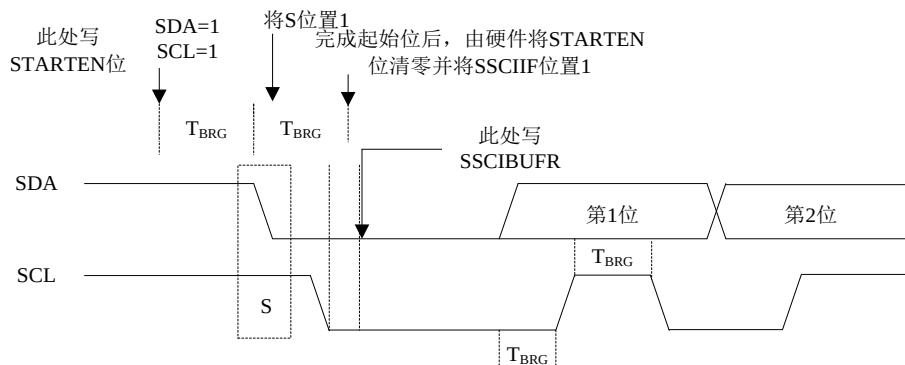


图13.10 第一个启动位时序

13.3.3.5I2C 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，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二个地址字节。当发送完第一个8位并接收到一个ACK后，用户可以发送另外8位地址（10位地址模式下）或8位数据（7位地址模式下）。

SSCIWCFL 状态标志

当重复启动序列进行时，如果用户写SSCIBUFR，则SSCIWCFL 被置1，同时缓冲器内容不变（未发生写操作）。

注:由于不允许事件排队,在重复启动条件结束之前,不能对SSCICTL1 的低5位进行写操作。

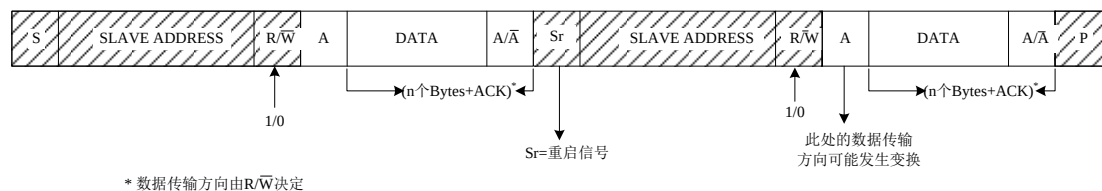


图13.11 I2C协议复合数据帧格式

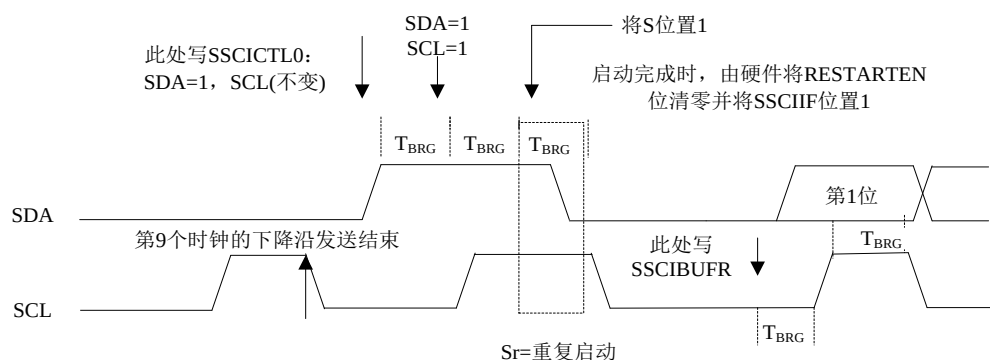


图13.12 重复启动条件时序波形

13.3.3.6 I2C 主控模式发送

I2C 协议中 SDA 引脚上的数据不仅来源于 SSCIBUFR 寄存器，还需要按 I2C 协议的规定发送起始位、停止位、应答位等信号。图 13.13 所示是 I2C 协议典型写数据帧格式。

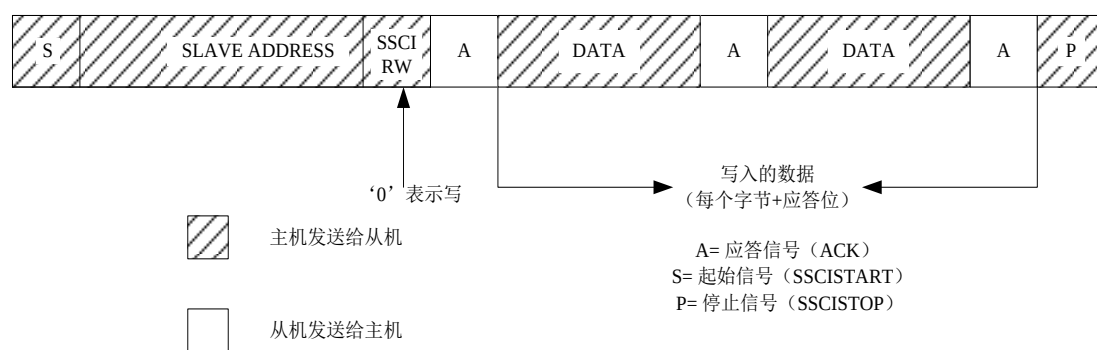


图 13.13 I2C 协议典型写数据帧格式

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 13.14 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。

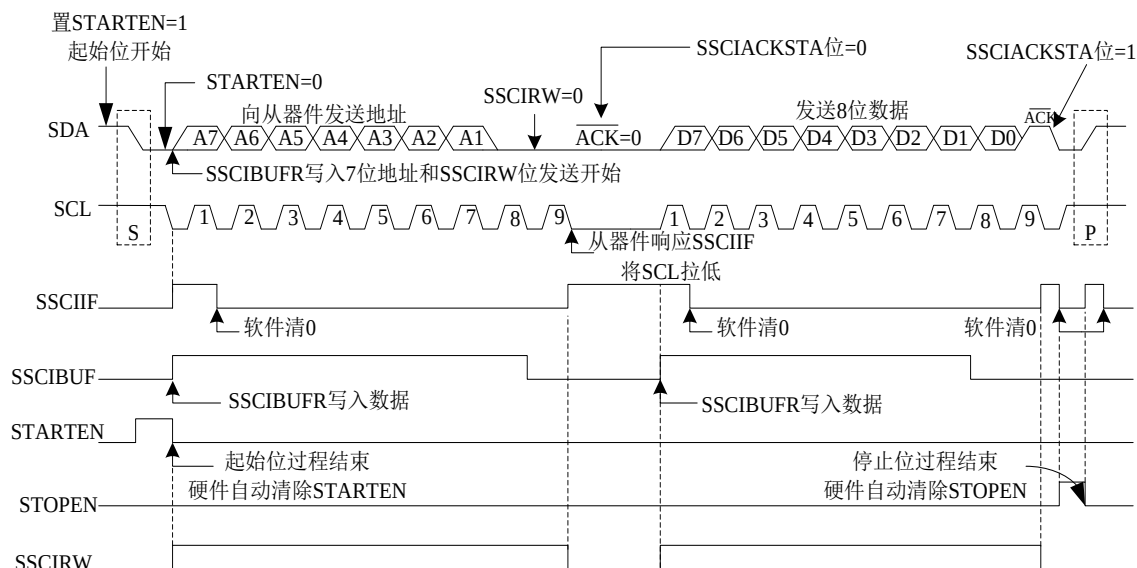


图 13.14 硬件 I2C 主模式发送数据时序图（7 位地址）

13.3.3.7I2C 主控模式接收

I2C主模式数据接受的系统结构与主模式数据发送系统结构相同，但是数据接收流程与数据发送流程不同。如图13.15为I2C协议典型读数据帧格式。

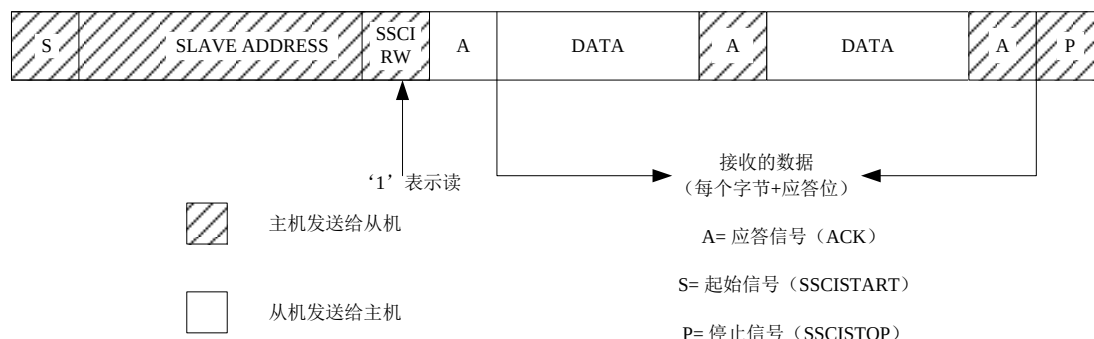


图13.15 I2C协议典型读数据帧格式

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13.16 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。

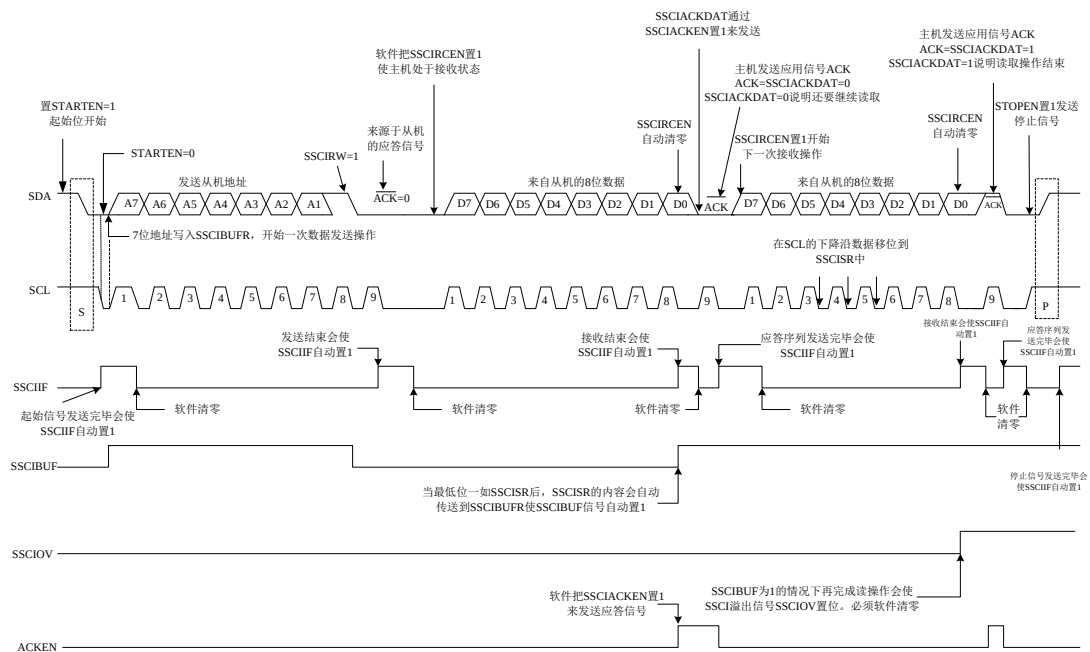


图13.16 I2C主模式接收数据时序图（7位地址）

13.3.3.8 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（TB RG）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TB RG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）。

13.3.3.9 停止条件序列

在接收/发送结束时，通过置停止序列的使能位，STOPEN（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，主控制器将SDA置为低电平。当SDA线采样为低电平时，波特率发生器被重新装入值并递减计数至0。波特率发生器发生超时，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）。

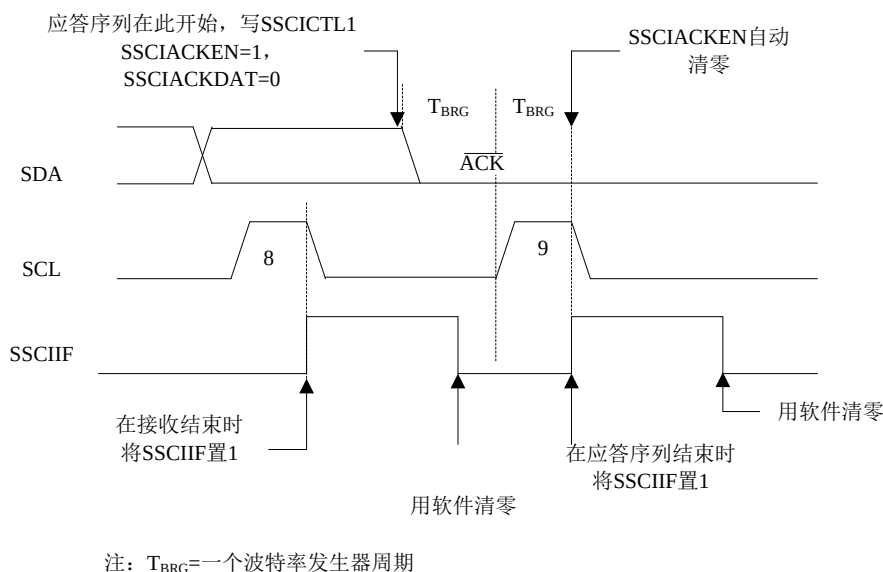


图13.17 应答序列时序波形

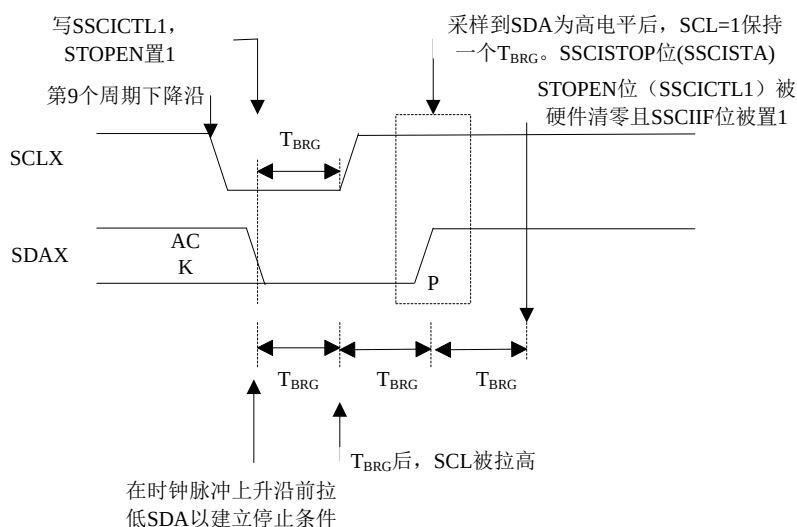


图13.18 停止条件接收或发送模式

13.3.3.10 时钟仲裁

如果在任何接收、发送或重复启动/ 停止条件期间，主器件拉高了 SCL 引脚（允许 SCL 引脚悬空为高电平），就会发生时钟仲裁。如果允许 SCL 引脚悬空为高电平，波特率发生器（BRG）将暂停计数，直到实际采样到 SCL 引脚为高电平为止。当 SCL 引脚采样为高电平时，波特率发生器中将被重新装入 $SSCIADD<6:0>$ 的内容并开始计数。这可以保证当外部器件将时钟拉低时，SCL 始终保持至少一个 BRG 计满返回周期的高电平。

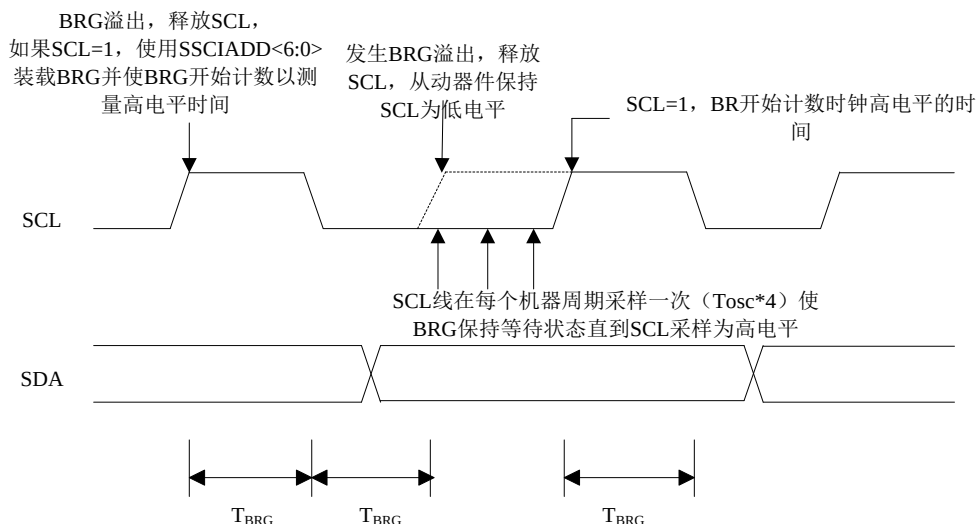


图 13.19 主控发送模式下的时钟仲裁时序

13.3.4 多主控制器模式

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(TR1<1:0>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。

13.3.4.1 多主机通信, 总线冲突与总线仲裁

多主机模式是通过总线仲裁来支持的。

当主器件将地址/数据位输出到SDA引脚时,如果一个主器件通过将SDA引脚悬空为高电平以在SDA上输出1,而另一个主器件输出0,就会发生总线仲裁。

如果SDA引脚上期望的数据是1,而实际在SDA引脚上采样到的数据是0,则发生了总线冲突。主器件将把总线冲突中断标志位BCLIF置1,并将I2C端口复位到空闲状态。如果在发送过程中发生总线冲突,则发送停止,SSCIBUF标志位清零,SDA和SCL线被拉高,并且允许对SSCIBUFR进行写操作。当执行完总线冲突中断服务程序后,如果I2C总线空闲,用户可通过发出启动条件恢复通信。

如果在启动、重复启动、停止或应答条件的进行过程中发生总线冲突,则该条件被中止,SDA和SCL线被拉高,SSCICtrl1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。

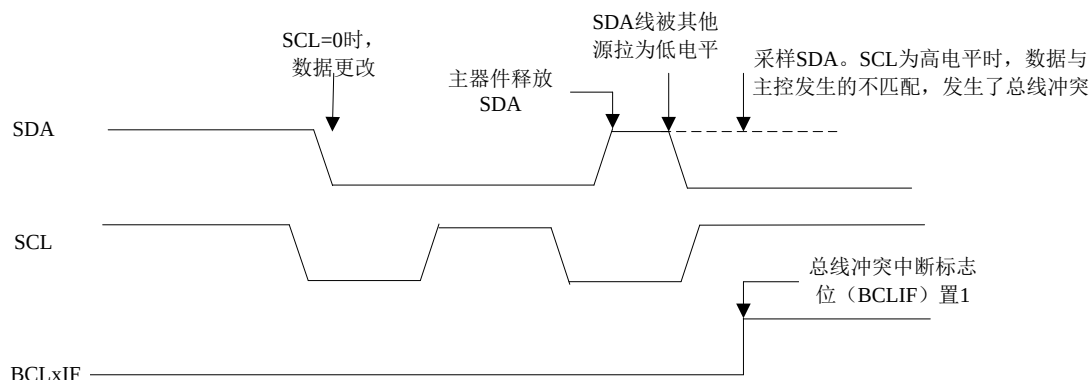


图 13.20 发送和应答时的总线冲突时序

13.3.4.2 启动条件期间的总线冲突

启动条件期间，以下事件将导致总线冲突：

- 1) 在启动条件开始时，SDA 或 SCL 被采样为低电平。
- 2) SDA 被拉低之前，SCL 采样为低电平。

在启动条件期间，SDA 和 SCL 引脚都会被监视。如果 SDA 引脚已经是低电平，或 SCL 引脚已经是低电平，则：

- 中止启动条件，
- BCLIF 标志位置 1，
- 并将 SSCI 模块复位为空闲状态。

启动条件从 SDA 和 SCL 引脚被拉高开始。当 SDA 引脚采样为高电平时，波特率发生器装入 SSCIADD<6:0> 的值并递减计数到 0。如果在 SDA 为高电平时，SCL 引脚采样为低电平，则发生总线冲突，因为这表示另一个主器件在启动条件期间试图发送一个数据 1。

如果 SDA 引脚在该计数周期内采样为低电平，则 BRG 复位，同时 SDA 线保持原值。但是，如果 SDA 引脚采样为 1。如果 SDA 引脚将在 BRG 计数结束时被置为低电平。随后波特率发生器被重新装入值并递减计数至 0。在此期间，如果 SCL 引脚采样到 0，则没有发生总线冲突。在 BRG 计数结束时，SCL 引脚被拉为低电平。

注：

在启动条件期间不会发生总线冲突是因为两个总线主器件不可能精确地在同一时刻发出启动条件。因此总是有一个主器件先于另一个主器件将 SDA 拉低。但是这一情况不会引起总线冲突，因为允许两个主器件对启动条件后的第一个地址进行仲裁。如果地址是相同的，将继续对数据部分、重复启动条件或停止条件进行仲裁。

在STARTEN位置1之前SDA变为低电平。BCLIF置1，因为SDA=0，SCL=1，所以SSCISTART位和SSCIIF位被置1

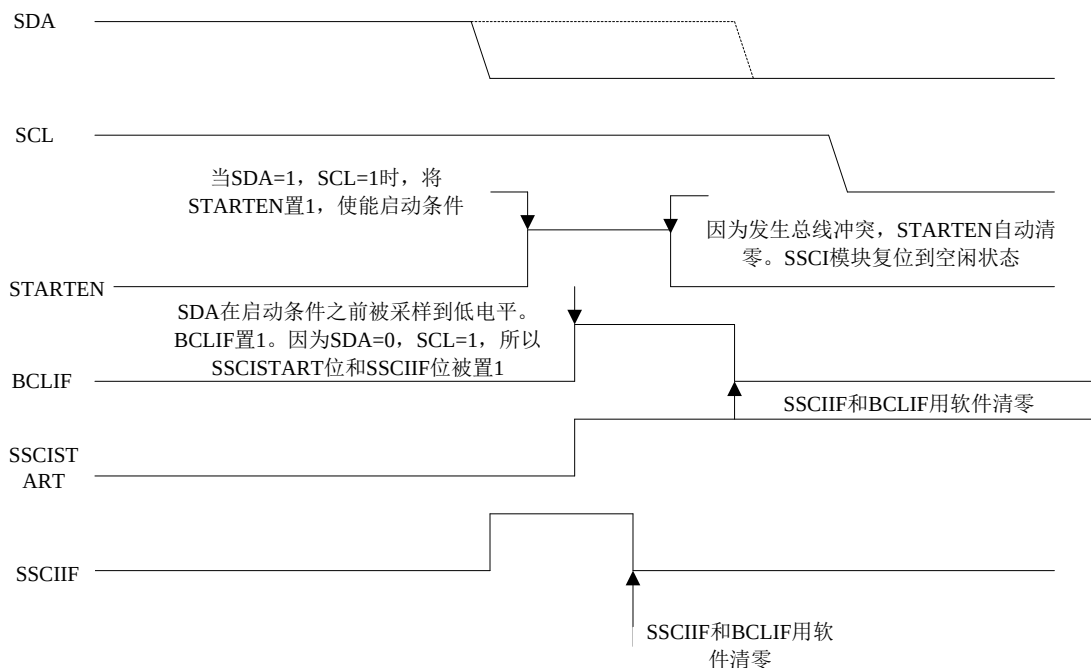
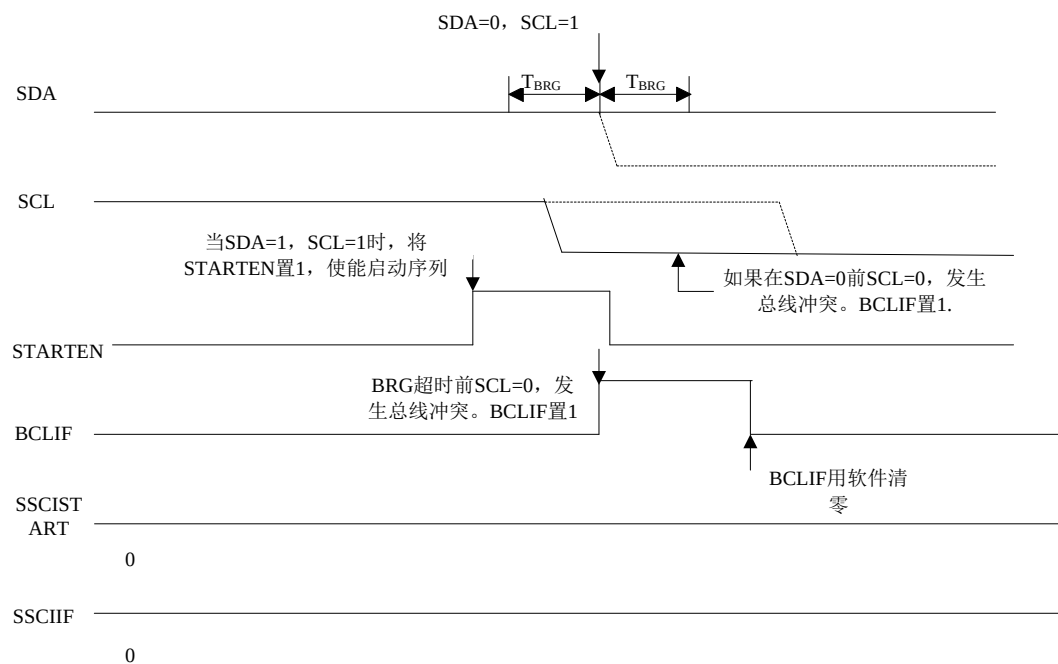


图13.21 启动条件期间的总线冲突（仅SDA）



图

13.22 启动条件期间的总线冲突（SCL=0）

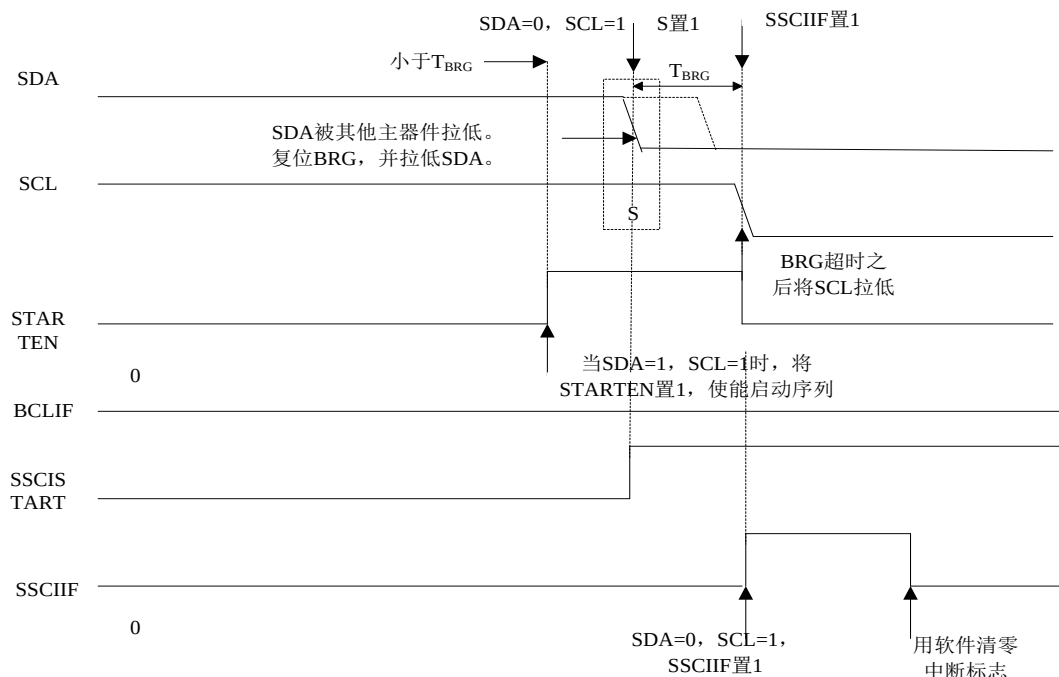


图13.23 启动条件期间由SDA仲裁引起的BRG复位

13.3.4.3 重复启动条件期间的总线冲突

在下列情况中，重复启动条件期间会发生总线冲突：

- 1) 在SCL 由低电平变为高电平的过程中， SDA 采样到低电平。
- 2) 在SDA 被拉为低电平之前， SCL 变为低电平，表示另一个主器件正试图发送一个数据1。

当用户拉高SDA 并允许该引脚悬空时， BRG 中装入SSCIADD<6:0> 中的值并递减计数至0。接着SCL 引脚被置为高电平，当SCL 采样到高电平时，对SDA 引脚进行采样。

如果 SDA 为低电平，则已发生了总线冲突（即，另一个主器件正试图发送一个数据 0）。如果 SDA 采样为高电平，则 BRG 被重新装入值并开始计数。如果 SDA 在 BRG 超时之前从高电平变为低电平，则没有发生总线冲突，因为两个主器件不可能精确地在同一时刻将 SDA 拉低。

如果 SCL 在 BRG 超时之前从高电平变为低电平，且 SDA 尚未变为低电平，表示发生了总线冲突。在此情况下，在重复启动条件期间另一个主器件正试图发送一个数据 1。

如果在 BRG 超时结束时 SCL 和 SDA 都仍然是高电平，则 SDA 引脚被拉低，BRG 重新装入值并开始计数。在计数结束时，无论 SCL 引脚的状态如何， SCL 引脚都被拉低，重复启动条件结束。

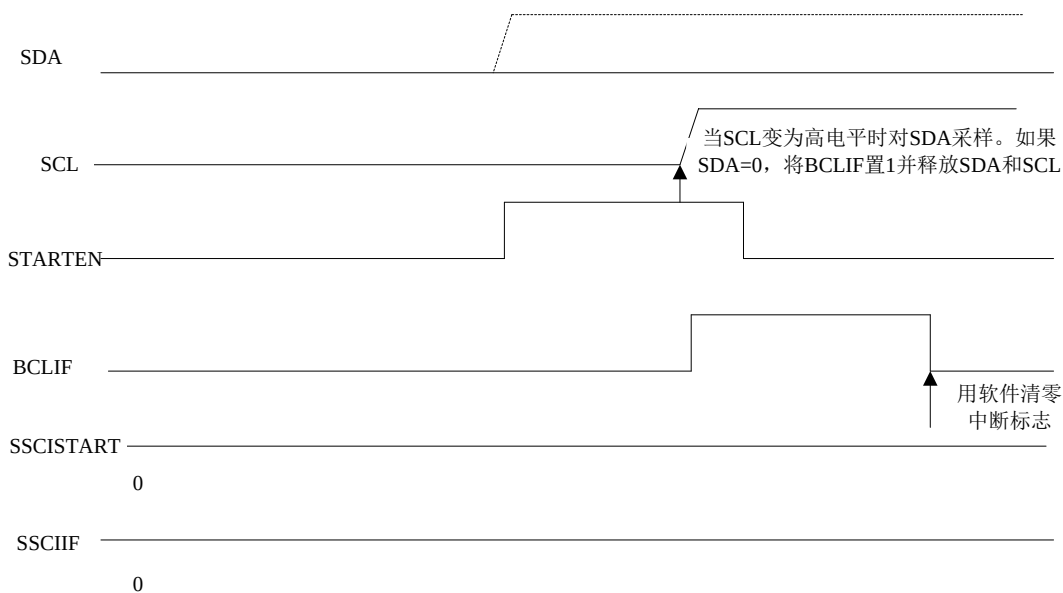


图 13.24 重复启动条件期间的总线冲突（情形 1）

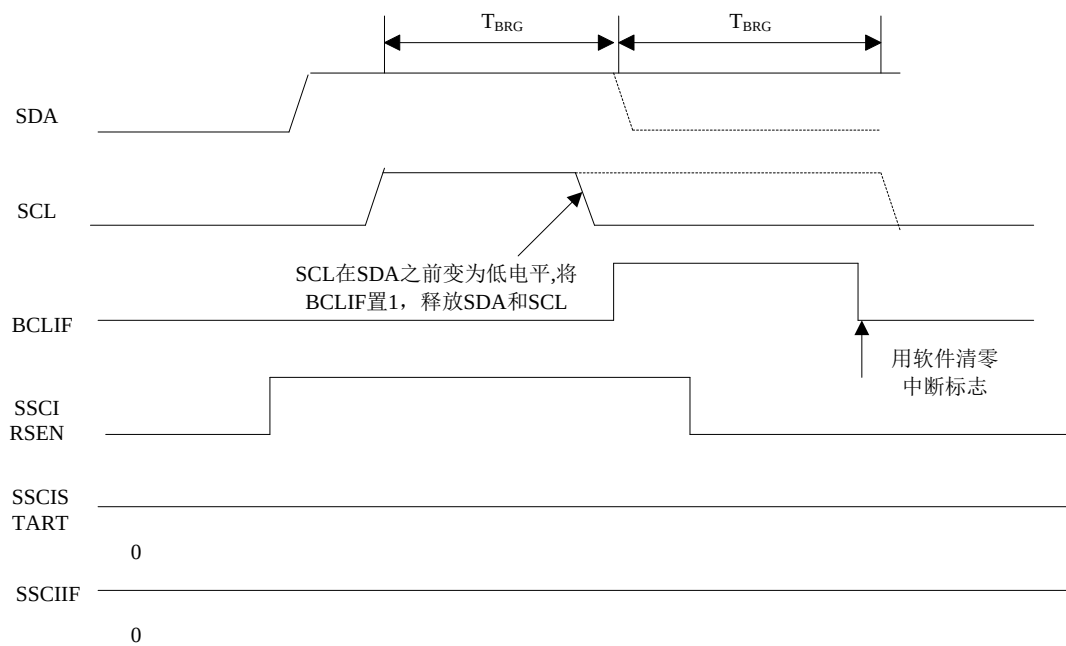


图 13.25 重复启动条件期间的总线冲突（情形 2）

13.3.4.4 停止条件期间的总线冲突

以下事件会导致停止条件期间的总线冲突:

- 1) SDA 已被拉高并允许悬空为高电平之后, SDA 在 BRG 超时后被采样到低电平。
- 2) SCL 引脚被拉高之后, SCL 在 SDA 变成高电平之前被采样到低电平。

停止条件从 SDA 被拉低开始。当 SDA 采样为低电平时, SCL 引脚就可以悬空为高电平。当引脚被采样到高电平时 (时钟仲裁), 波特率发生器中装入 SSCIADD<6:0> 的内容并递减计数到 0。BRG 超时后, 采样 SDA。如果 SDA 采样到低电平, 则已发生总线冲突。这是因为另一个主器件正试图发送一个数据 0。如果 SCL 引脚在允许 SDA 悬空为高电平前被采样到低电平, 也会发生总线冲突。这是另一个主器件正试图发送一个数据 0 的又一种情况。

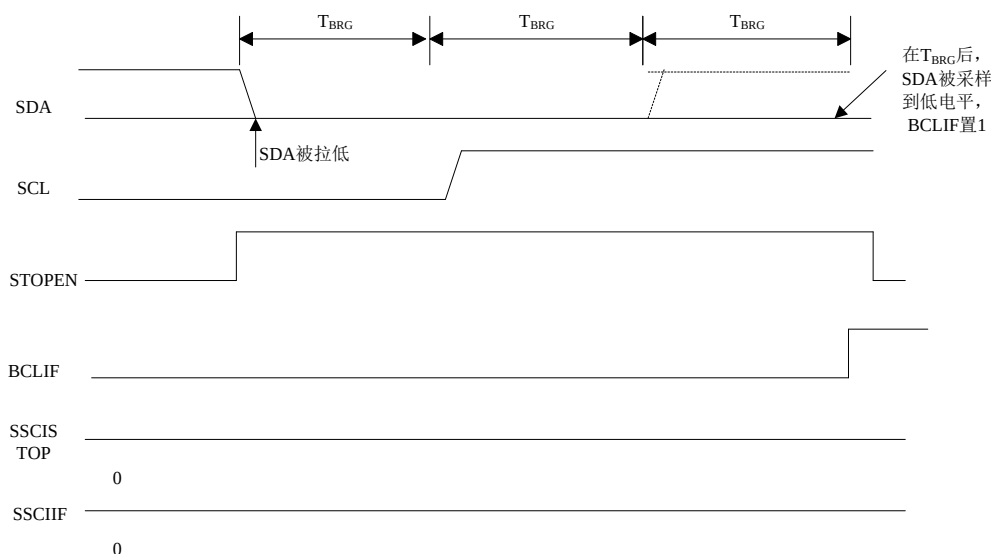


图13.26 停止条件期间的总线冲突 (情形1)

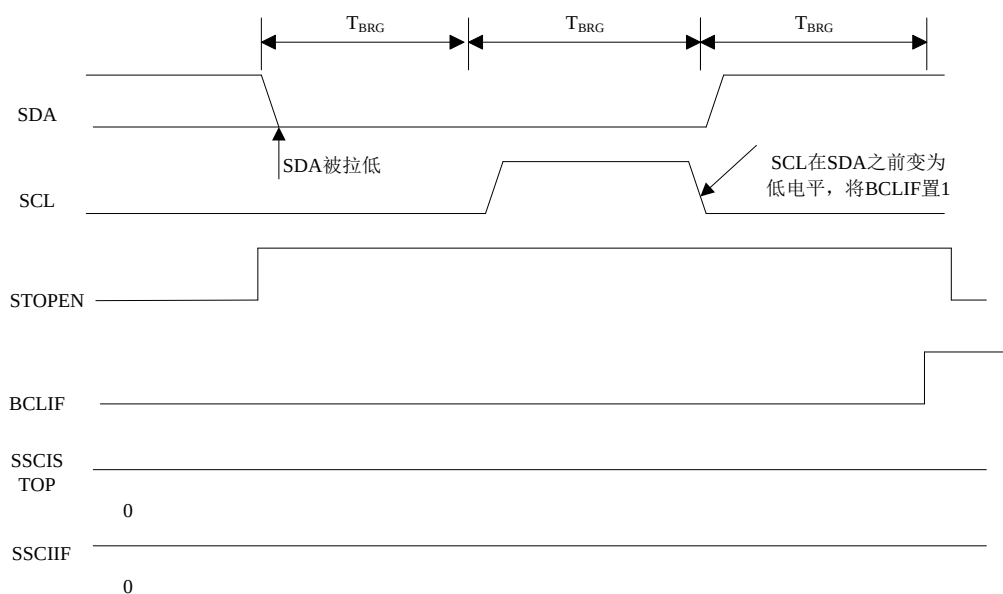


图13.27 停止条件期间的总线冲突 (情形2)

13.3.4.5SSCI 屏蔽寄存器

在I2C 从动模式下，SSCI 屏蔽(SSCIMSK)寄存器用于在地址比较操作下屏蔽SSCISR寄存器中的值。SSCIMSK 寄存器中某位为0 会使SSCISR寄存器中相应的位成为“无关位”。

此寄存器在任何复位条件发生时均复位为全1，因此，在写入屏蔽值前，它对标准SSCI操作没有影响。

必须在通过设置SSCIMOD<3:0> 位以选择I2C 从动模式（7位或10 位地址）之前对此寄存器进行初始化。只有通过SSCICTL0 的SSCIMOD<3:0> 位选择了适当的模式后才可访问此寄存器。SSCI 屏蔽寄存器在以下情况下有效：

7 位地址模式：与A<7:1> 进行地址比较。

10 位地址模式：仅与 A<7:0> 进行地址比较。SSCI 屏蔽在接收到地址的第一个（高）字节期间无效。

13.4 SPI 模式

SPI 是一种应用很广泛的串口总线技术，其特点主要表现在：

- ◆ 3 线或者 4 线数据传输
- ◆ 8 位传输帧格式
- ◆ 主从模式
- ◆ 时钟频率可设
- ◆ 可编程的时钟极性和相位
- ◆ 可触发中断的发送和接收标志

SPI 的原理框图如图 13.28 所示：

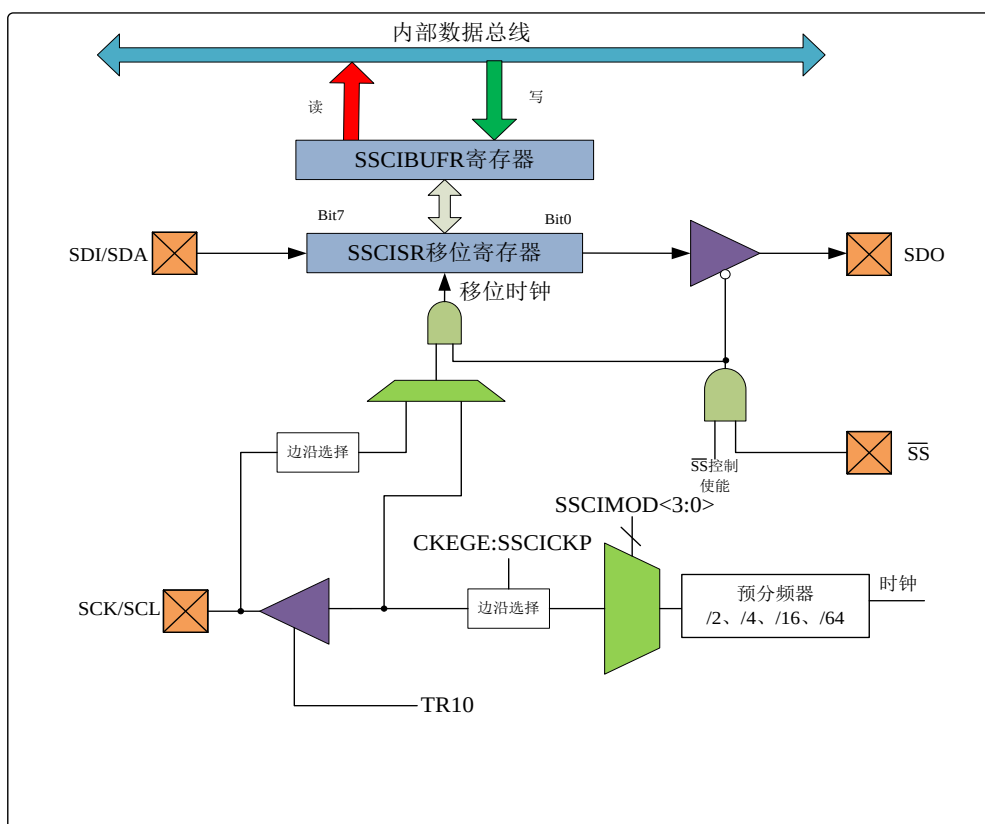


图 13.28 SPI 原理框图

在 SPI 模式下允许同时同步发送和接收 8 位数据。通常使用以下三个引脚来完成通信：

- ◆ 串行数据输出（Serial Data Out，SDO）
- ◆ 串行数据输入（Serial Data In，SDI）
- ◆ 串行时钟（Serial Clock，SCK）

此外，当工作在从动模式下时可以使用第 4 个引脚：

- ◆ 从动选择（ $\overline{SS}$ ）

注意：

- 1) 如果使用 SPI 从动模式，且 CKEGE = 1，则必须使能 $\overline{SS}$ 引脚控制（SSICCTL0<3:0> = 0100）。
- 2) 当 SPI 处于从动模式时，如果 $\overline{SS}$ 引脚设为 VDD，那么 SPI 模块将复位。

13.4.1 工作原理

在用户初始化 SPI 时,需要先通过对相应的控制位(SSCCTL0<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 (SSCCTL0<7>) 会置 1。此时用户必须用软件将 SSCIWCFL 位清零, 否则无法判别以后对 SSCIBUFR 的写操作是否成功。

2) 当在发送数据时, 为了确保应用软件能有效地接收数据, 应该在要发送的下一数据字节被写入 SSCIBUFR 之前, 读取 SSCIBUFR 中现有的数据。

缓冲器满标志位 SSCIBUF (SSCISTA<0>) 指出将接收到的数据装入 SSCIBUFR (发送完成) 的时间。当 SSCIBUFR 中的数据被读取后, SSCIBUF 位即被清零。如果 SPI 仅作为一个发送器, 则不必理会接收的数据。通常, 可用 SSCI 中断来判断发送和接收完成的时间。必须读取/ 写入 SSCIBUFR。如果不打算使用中断, 用软件查询的方法同样可确保不会发生写冲突。

注: 不能直接读写 SSCISR 寄存器, 只能通过寻址 SSCIBUFR 寄存器来访问。此外, SSCI 状态寄存器 (SSCISTA) 指示各种状态条件。

13.4.2 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 TR3 寄存器中）。即：

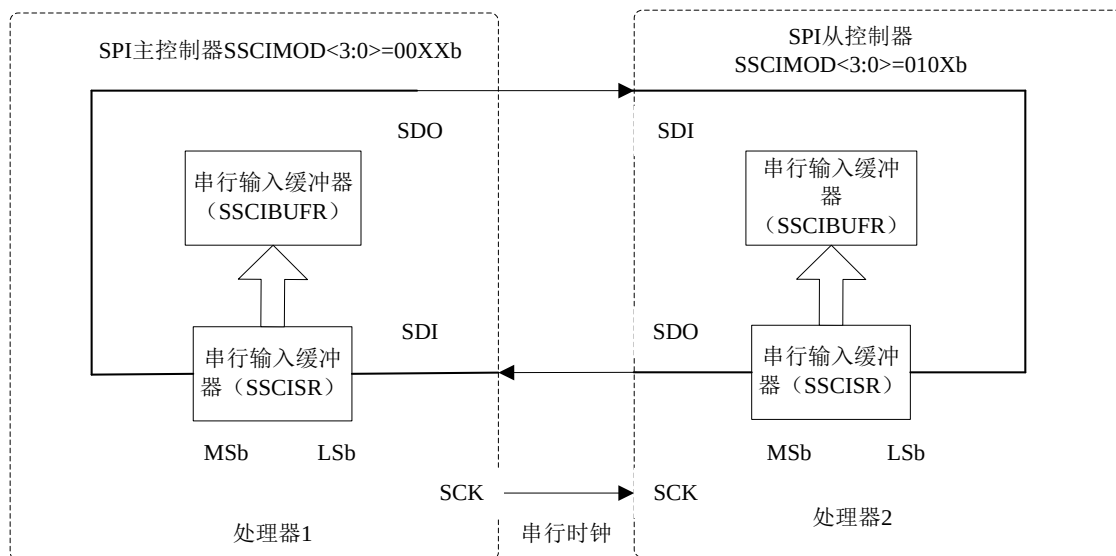
- ◆ SDI，必须将 TR1<1>置 1（由 SPI 模块自动控制）
- ◆ 对于 SDO，必须将 TR3<7>清零
- ◆ 对于 SCK（主控模式），必须将 TR1<0>清 0
- ◆ 对于 SCK（从动模式），必须将 TR1<0>置 1
- ◆ 对于 $\overline{SS}$ ，必须将 TR3<6>置 1。

对于不需要的任何串行端口功能，可通过将对应的数据方向 TRx 寄存器设置为相反值来屏蔽。

13.4.3 典型连接

下图 12.29 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



13.29 SPI 主从控制器连接

13.4.4 主模式

因为由主控制器控制 SCK 信号，所以它可以在任意时刻启动数据传输。主控制器根据软件协议确定从控制器（图 13.29 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 13.30、图 13.31、图 13.32 和图 13.33 将给出 SPI 通信的时序图，其中首先发送的是最高有效位。在主控模式下，SPI 时钟速率（波特率）可由用户编程设定为下面几种方式之一：

- SCLK/2
- SCLK/4
- SCLK/16
- SCLK/64

下图 13.30 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。

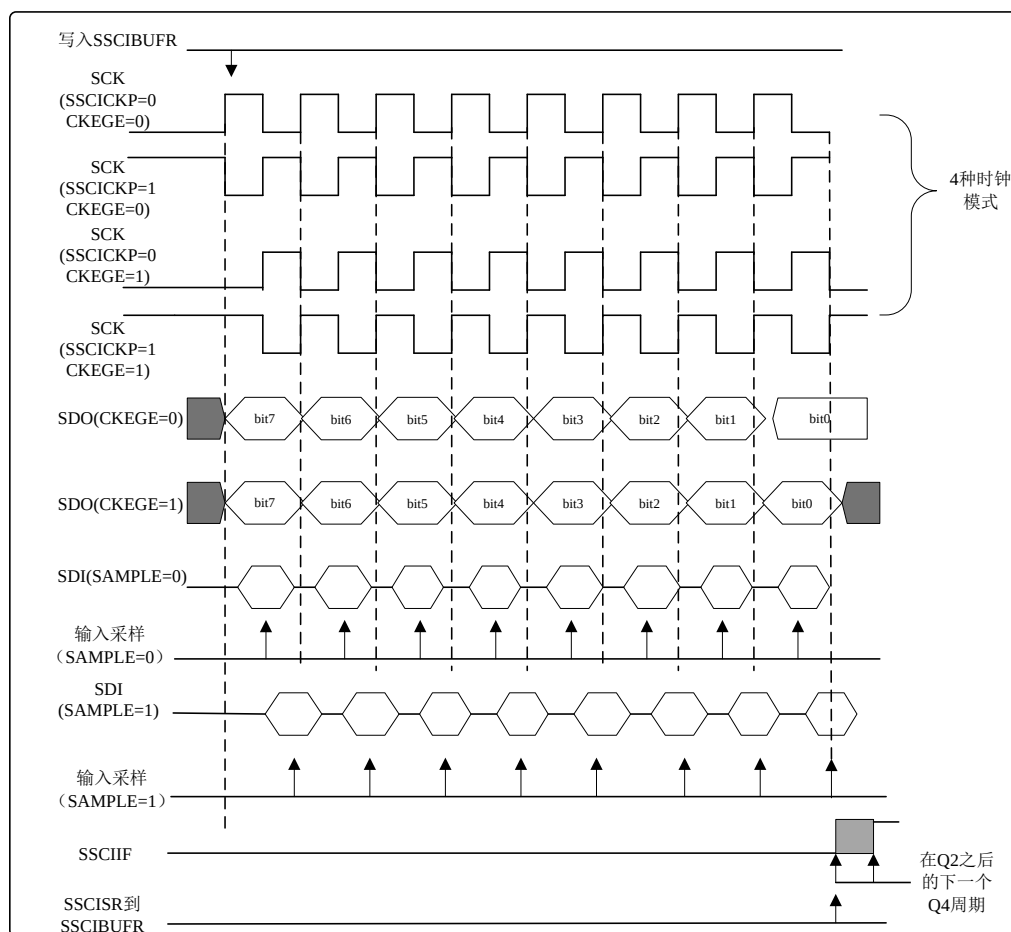


图 13.30 SPI 主控模式时序图

13.4.5 从模式

13.4.5.1 从动模式

在从动模式下，当SCK引脚上出现外部时钟脉冲时发送和接收数据。当最后一位数据被锁存后，中断标志位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。

13.4.5.2 从动选择同步

$\overline{SS}$ 引脚允许器件工作于同步从动模式。SPI必须处于从动模式，并使能 $\overline{SS}$ 引脚控制（SSCICTL0<3:0> = 0100）。要让 $\overline{SS}$ 引脚充当输入端，则不能将此引脚驱动为低电平。数据锁存器必须为高电平。当 $\overline{SS}$ 引脚为低电平时，使能数据的发送和接收，同时驱动SDO引脚。当 $\overline{SS}$ 引脚变为高电平时，即使是在字节的发送过程中，SDO引脚也不再被驱动，而是变成悬空输出状态。根据应用的需要，可在SDO引脚上外接上拉/下拉电阻。

注意: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。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）。

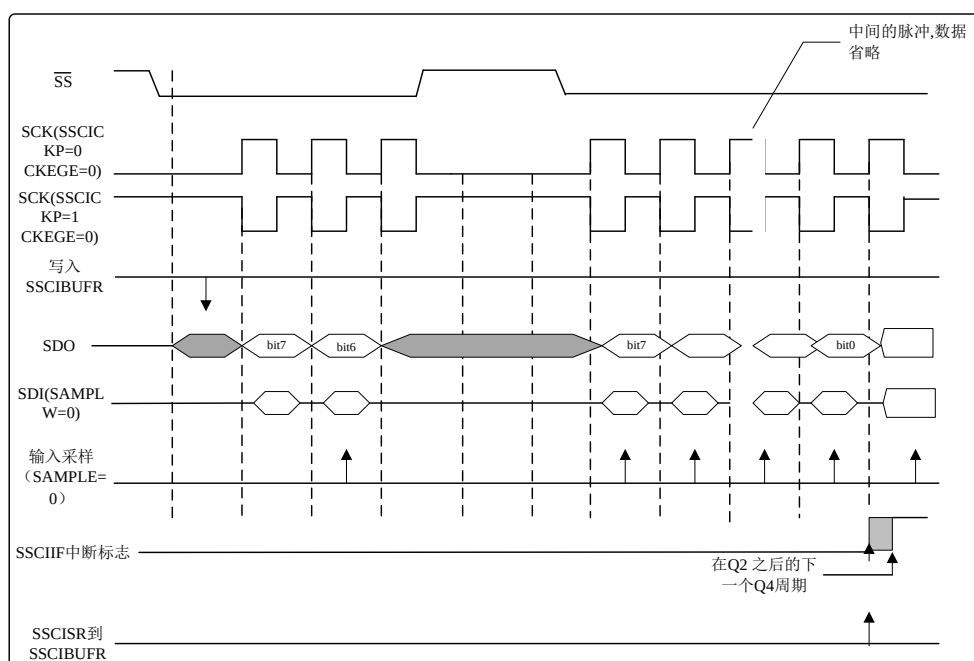


图13.31 从动时序图

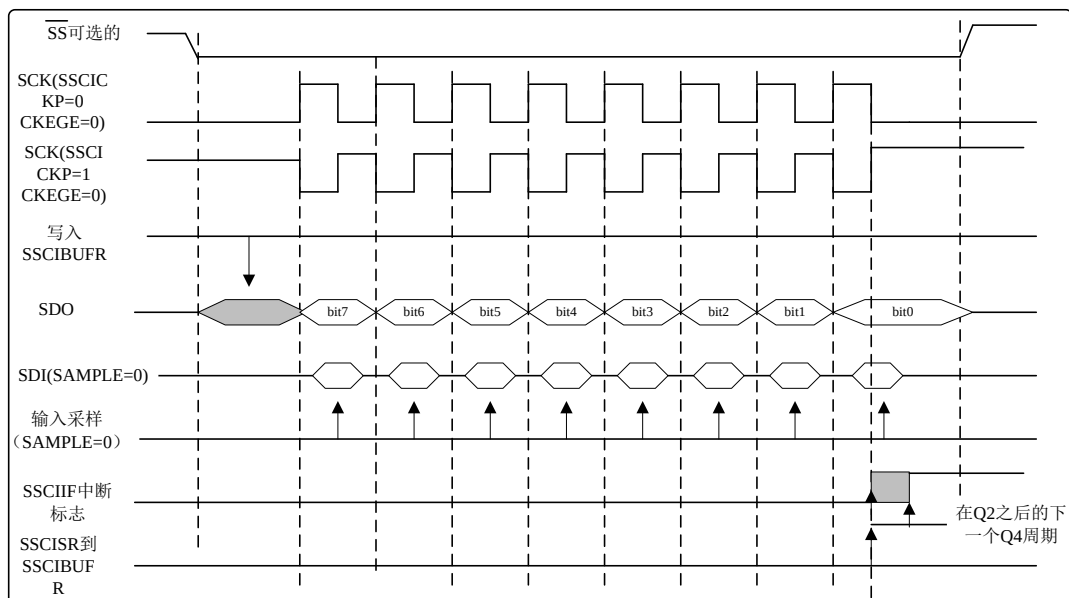


图13.32 SPI模式时序图（从动模式且CKEGE=0）

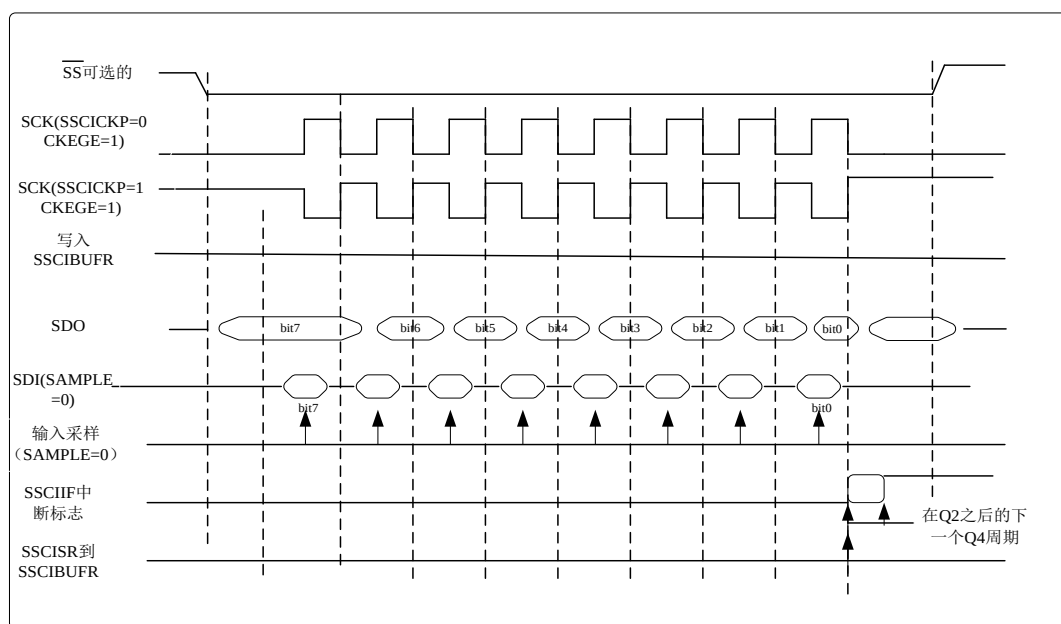


图 13.33 SPI 模式时序图（从动模式且 CKEGE=1）

13.4.6 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。

13.4.7 SPI 四种工作模式设置

13.4.7.1 主控发送工作流程

主要步骤:

- 1、通过 SSCIMOD<3:0>确定 SPI 主控方式
 - 0000:SPI 主控方式，时钟=工作时钟/4
 - 0001:SPI 主控方式，时钟=工作时钟/16
 - 0010:SPI 主控方式，时钟=工作时钟/64
 - 0011:SPI 主控方式，时钟= T2 输出/2
- 2、通过寄存器 SSCICTL0 的 SSCICKP 和寄存器 SSCISTA 的 CKEGE 选择时钟与数据的时序关系；
- 3、将相应的引脚 SDO 引脚设置为输出，SCK 引脚设置为输出；
- 4、置 1 SSCIEN 位，使能 SSCI 模块；
- 5、此时 SSCIIF 清零，将要发送的数据写到 SSCIBUFR 寄存器；
- 6、当一个字节发送完毕 SSCIIF 自动置 1（软件清零），如果需要中断，则使能相应中断使能位；

13.4.7.2 从动接收工作流程

主要步骤:

- 1、通过 SSCIMOD<3:0>确定 SPI 从动方式；
 - 0100:SPI 从动模式，时钟由 SCK 引脚输入，使能 $\overline{SS}$ 引脚功能，此时要设置 $\overline{SS}$ 引脚为输入口
 - 0101:SPI 从动模式，时钟由 SCK 引脚输入，关闭 $\overline{SS}$ 引脚功能， $\overline{SS}$ 被用作普通 I/O 引脚
- 2、通过寄存器 SSCICTL0 的 SSCICKP 和寄存器 SSCISTA 的 CKEGE 选择时钟与数据的时序关系；与主控方的时钟时序要一致；
注意:当 CKEGE 为 1 时，只有信号 $\overline{SS}$ 为低电平，从动端才会接收数据，否则从动端不工作； $\overline{SS}$ 信号都来源于主控方。
将相应的引脚 SDI 引脚设置输入，SCK 引脚设置为输入；使能 $\overline{SS}$ 引脚功能，此时要设置 $\overline{SS}$ 引脚为输入口；

- 3、置 1 SSCIEN 位，使能 SSCI 模块；
- 4、当接受到一个字节的数据后，SSCIIF 自动置 1（软件清零），SSCIBUF 置 1，应立即读取 SSCIBUFR 的值；
- 5、如果需要中断，则使能相应的中断使能位；

13.4.7.3 主控接收工作流程

主要步骤:

- 1、通过 SSCIMOD<3:0>确定 SPI 主控方式
 - 0000:SPI 主控方式，时钟=工作时钟/4
 - 0001:SPI 主控方式，时钟=工作时钟/16
 - 0010:SPI 主控方式，时钟=工作时钟/64
 - 0011:SPI 主控方式，时钟=T2 输出/2
- 2、通过寄存器 SSCICTL0 的 SSCICKP 和寄存器 SSCISTA 的 CKEGE 选择时钟与数据的时序关系；
- 3、将相应的引脚 SDI 引脚设置输入，SCK 引脚设置为输出；
- 4、置 1 SSCIEN 位，使能 SSCI 模块；
- 5、当接受到一个字节的数据后，SSCIIF 自动置 1（软件清零），SSCIBUF 置 1，应立即读取 SSCIBUFR 的值；
- 6、如果需要中断，则使能相应的中断使能位；

13.4.7.4 从动发送工作流程

主要步骤:

- 1、通过 SSCIMOD<3:0>确定 SPI 主控方式；
 - 0100:SPI 从动模式，时钟由 SCK 引脚输入，使能 $\overline{SS}$ 引脚功能，此时要设置 $\overline{SS}$ 引脚为输入口
 - 0101:SPI 从动模式，时钟由 SCK 引脚输入，关闭 $\overline{SS}$ 引脚功能， $\overline{SS}$ 被用作普通 I/O 引脚
- 2、通过寄存器 SSCICTL0 的 SSCICKP 和寄存器 SSCISTA 的 CKEGE 选择时钟与数据的时序关系；
- 3、将相应的引脚 SDO 引脚设置为输出，SCK 引脚设置为输入；如果使能 $\overline{SS}$ 引脚功能，此时要设置 $\overline{SS}$ 引脚为输入口；
- 4、置 1 SSCIEN 位，使能 SSCI 模块；
- 5、此时 SSCIIF 清零，将要发送的数据写到 SSCIBUFR 寄存器；
- 6、当一个字节发送完毕 SSCIIF 自动置 1（软件清零），如果需要中断，则使能相应中断使能位；

14 通用全双工/ 半双工收发器 (USART)

14.1 系统概述

KF8L15Z20XX 包含 2 个通用全双工/半双工收发器 USART1 和 USART2。USART1 与 USART2 是两个完全独立的模块。USART2 与 USART1 工作原理完全相同,区别在于 UART2 兼容 7816(在 13.7 章节介绍)。本章将以 USARTx (x=1、2) 模块来讲述。

USARTx 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。

USART1 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居于互联网络 (Local Interconnect Newxork,LIN) 总线系统。

14.1.1 相关引脚

USART1 和 USART2 的引脚详细信息如表 14-1 所示。用户在使用 USARTx 模块时需要注意:

1. 当使用 USART1 模块前,需要将 UPINSET1 寄存器的 UPSEL1 位置 1,将 P4.1 口和 P4.0 口分配给 USART1 模块;
2. 当使用 USART2 模块时,通信引脚功能无需进行任何额外配置;即,请勿将 UPINSET2 寄存器的 UPSEL2 位置 1,否则将导致 USART2 无法正常使用。

表 14-1 USARTx 相关引脚

名称	位于	说明
RX1/DT1	P4.1	串口 1 同步数据引脚/异步接收引脚
TX1/CK1	P4.0	串口 1 同步时钟引脚/异步发送引脚
RX20/DT20	P2.1	串口 2 同步数据引脚/异步接收引脚 0
RX21/DT21	P2.0	串口 2 同步数据引脚/异步接收引脚 1
TX2/CK2	P2.2	串口 2 同步时钟引脚/异步发送引脚

14.1.2 相关寄存器

表 14-2 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
0BH	INTCTL	AIE/ AIEH	PUIE/ AIEL	T0IE	-	P0IE	T0IF	-	P0IF
0DH	EIF2	T3IF	-	RC1IF	TX1IF	C3IF	-	BCLIF	SSCIF
2DH	EIE2	T3IE	-	RC1IE	TX1IE	C3IE	-	BCLIE	SSCIE
24H	IP2	PT3	-	PRC1	PTX1	PC3	-	PBCL	PSSCI
120H	RSCTL1	SPEN1	R1X9	SRXEN1	CRXEN1	ADREN1	FRER1	OVFER1	RX9D1
24AH	RSCTL2	SPEN2	R2X9	SRXEN2	CRXEN2	ADREN2	FRER2	OVFER2	RX9D2
121H	TXSDR1	USART 发送数据寄存器 1							
24EH	TXSDR2	USART 发送数据寄存器 2							
122H	RXSDR1	USART 接收数据寄存器 1							
24BH	RXSDR2	USART 接收数据寄存器 2							
123H	BRCTL1	ABRDO VF1	RCIDLF1	SCKPS1	B1RG16	BR1CKS1	BR1CKS0	WUEN1	ABRDEN1
249H	BRCTL2	ABRDO VF2	RCIDLF2	SCKPS2	B2RG16	BR2CKS1	BR2CKS0	WUEN2	ABRDEN2
124H	TSCTL1	CSRS1	T1X9	TXEN1	SYNC1	SENDERB1	HBRG1	TXSRS1	TX9D1
24FH	TSCTL2	CSRS2	T2X9	TXEN2	SYNC2	SENDERB2	HBRG2	TXSRS2	TX9D2
150H	UPINSET1	USLM1	UPSEL1	-	-	-	-	-	-
23FH	UPINSET2	USLM2	UPSEL2	-	-	-	-	-	-
125H	EUBRGL1	USART1 波特率数据寄存器低字节							
24DH	EUBRGL2	USART2 波特率数据寄存器低字节							
126H	EUBRGH1	USART1 波特率数据寄存器高字节							
24CH	EUBRGH2	USART2 波特率数据寄存器高字节							
242H	U7816TXCTL2	U7816 EN2	CLKOUT2	TX9DSEL2	STOP2	TPAR2	TINV2	TCONV2	BGTEN2
241H	U7816RXCTL2	ERSW21	ERSW20	RPAR2	RINV2	RCONV2	-	-	PAREF2
240H	U7816CTL2	PSEL2	-	TREPEN2	TREP21	TREP20	RREPEN2	RREP21	RREP20
243H	CLKDIV2	CLKDIV 27	CLKDIV26	CLKDIV25	CLKDIV24	CLKDIV23	CLKDIV22	CLKDIV21	CLKDIV2 0
244H	EGTCTL2	EGT27	EGT26	EGT25	EGT24	EGT23	EGT22	EGT21	EGT20
127H	USLPEN1	SLPEN1	-	-	-	-	-	-	-
248H	USLPEN2	SLPEN2	-	-	-	-	-	-	-
12DH	UARTMTH1	UART1 地址匹配设置寄存器							
247H	UARTMTH2	UART2 地址匹配设置寄存器							

14.1.3 原理框图

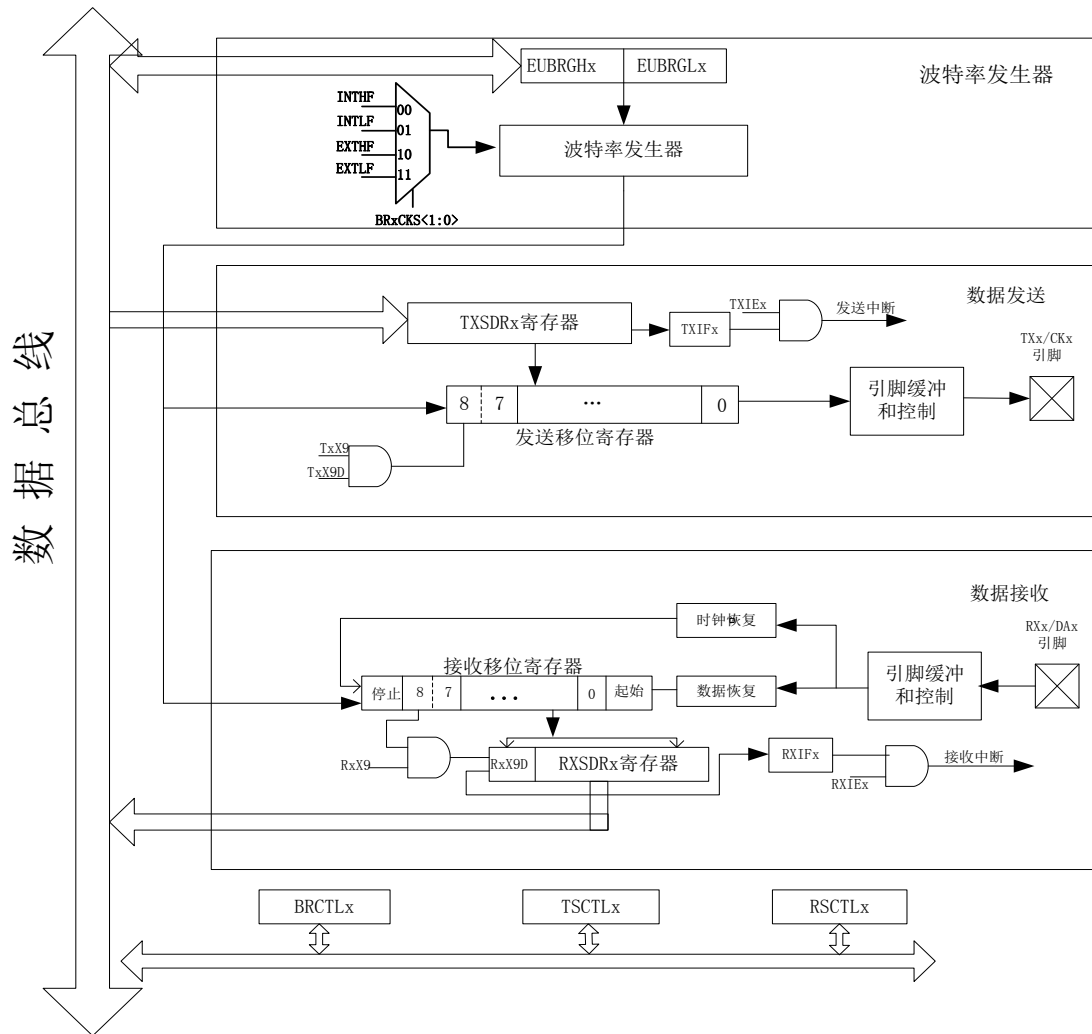


图 14.1 USARTx 模块框图

从图 14.1 中可以看出，通用全双工/半双工收发器（USARTx）模块包括波特率发生器（BRG）、数据发送和数据接收这三部分，每个部分都有相应的寄存器设置，主要包括波特率控制寄存器 BRCTLx、发送状态和控制寄存器 TSCTLx、接收状态和控制寄存器 RSCTLx。

在波特率发生器这部分，通过设置 SYNCx、HBRGx 和 BxRG16 位来进行选择设置 {EUBRGHx: EUBRGLx} 寄存器的值来配置波特率。详细见“[14.2 波特率发生器](#)”。

在数据发送部分，包括全双工异步发送和半双工同步发送，通过发送状态和控制寄存器 TSCTLx 来控制实现相应的功能。详见“[14.3 及 14.4](#)”。

在数据接收部分，包括全双工异步接收和半双工同步接收，通过接收状态和控制寄存器 RSCTLx 来控制实现相应的功能。详见“[14.3 及 14.4](#)”。

USARTx 有如下功能特点：全双工异步发送和接收、RS485 检测、双字节输入缓冲器、单字节输出缓冲器、可将字符长度编程为 8 位或 9 位、输入缓冲溢出错误检测、接收到字符的帧错误检测、半双工同步主控/从动模式和半双工同步模式下可编程时钟极性。

USARTx 模块还可实现如下附加功能，从而使其成为 LIN 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。

14.2 波特率发生器

波特率发生器（BRG）是 8 位或者 16 位定时器，专用于支持全双工和半双工 USART1 操作，默认情况下，BRG 工作在 8 位模式。BRxCKS<1:0>位决定波特率发生器选用哪一种工作时钟，其中包括 INTLF、INTLF、EXTHF 和 EXTFLF 四种时钟信号。将 BRCTLx 寄存器的 BxRG16 位置 1 可选择 16 位模式。

{EUBRGHx: EUBRGLx} 寄存器决定运行波特率定时器的周期。在全双工模式下，波特率周期的乘数由 TSCTLx 寄存器的 HBRGx 位和 BRCTLx 寄存器的 BxRG16 位决定。在半双工同步模式下，HBRGx 位被忽略。

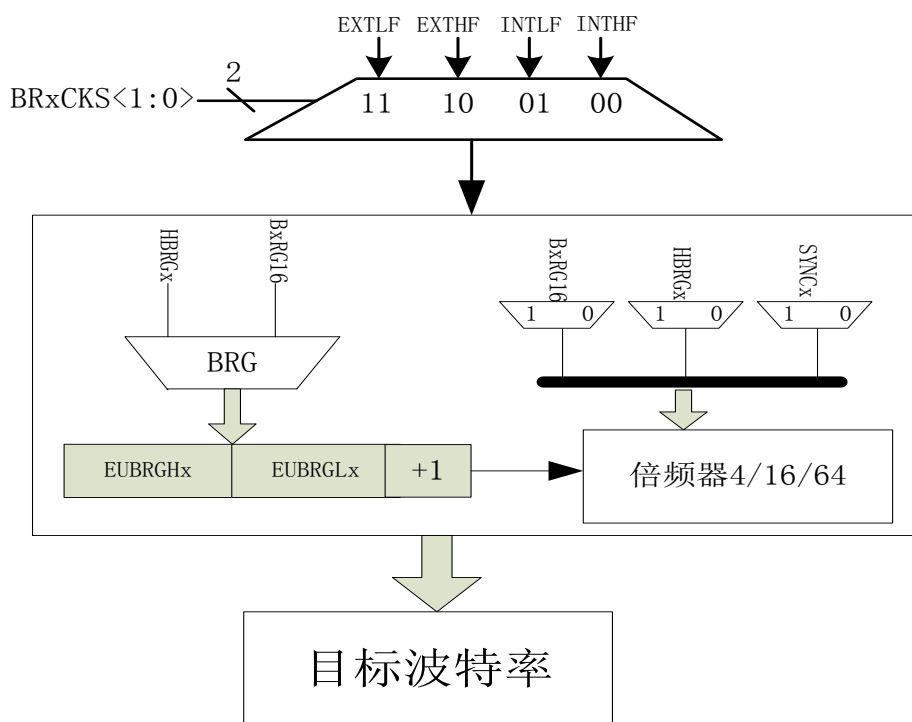


图 14.2 USARTx 目标波特率产生原理框图

14.2.1 USARTx 波特率控制寄存器 BRCTLx

寄存器: BRCTLx: 波特率控制寄存器x

	bit7						bit0	
复位值 0100 0000	ABRDOV Fx	RCIDLFx	SCKPSx	BxRG16	BRxCKS1	BRxCKS0	WUENx	ABRDEN x
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ABRDOVFx: 自动波特率检测溢出位

全双工异步模式:

1 = 自动波特率定时器溢出

0 = 自动波特率定时器没有溢出

半双工同步模式:无关位

RCIDLFx: 接收空闲标志位

全双工异步模式:

1 = 接收器空闲

0 = 已接收到起始位且接收器正在接收

半双工同步模式:无关位

SCKPSx: 半双工同步时钟极性选择位

全双工异步模式:必须为 0

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

BxRG16: 16 位波特率发生器位

1 = 使用 16 位波特率发生器

0 = 使用 8 位波特率发生器

BRxCKS<1:0>:USART 波特率发生器时钟选择位

00 = 选择INTHF时钟作为USARTx波特率发生器时钟

01 = 选择INTLF时钟作为USARTx波特率发生器时钟

10 = 选择EXTHF时钟作为USARTx波特率发生器时钟

11 = 选择EXTLF时钟作为USARTx波特率发生器时钟

WUENx: 唤醒使能位

全双工异步模式:

1 = 接收器正在等待下降沿, 不会接收字符, RCxIF 将被置 1, 当 RCxIF 置 1 后 WUENx 将被自动清 0。

0 = 接收器正常工作

半双工同步模式:无关位

ABRDENx: 自动波特率检测使能位

全双工异步模式:

1 = 使能自动波特率模式(完成自动波特率后清 0)

0 = 禁止自动波特率模式

半双工同步模式:无关位

14.2.2 波特率的选择

波特率的选择与状态位 SYNCx、BxRG16 和 HBRGx 有关，还与工作定时器 8 位或 16 位、半双工同步或全双工都有关系。

波特率发生器有三种时钟选择，分别是系统主时钟 SCLK、高频时钟 HFCLK 和低频时钟 LFCLK，通过比特率寄存器 BRCTLx 的 BRxCK<1:0>选用。

波特率的计算公式为：

$$\text{目标波特率} = \frac{F_{osc}}{m \times ([EUBRGHx : EUBRGLx] + 1)}$$

其中 m 为倍频器选择，分别为 4、16 和 64，它与状态位 SYNCx、BxRG16 和 HBRGx 有关，并且与工作的定时器 8 位还是 16 位，半双工或全双工有一定的关系。表 13-2 列出了 m 选择表。

表 14-3 倍频器 m 选择表

配置位			BRG/USARTx 模式	倍频器 m
SYNCx	BxRG16	HBRGx		
0	0	0	8 位/异步	64
0	0	1	8 位/异步	16
0	1	0	16 位/异步	
0	1	1	16 位/异步	4
1	0	x	8 位/同步	
1	1	x	16 位/同步	

注：x 为无关位

例 14.1 是确定波特率和波特率误差的计算示例。

例 14.1: 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:

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

$$\begin{aligned}
 X &= \frac{\frac{F_{osc}}{\text{目标波特率}}}{64} - 1 \\
 &= \frac{16000000}{9600} - 1 \\
 &= 25.042 \\
 &= 25 \text{ 十进制数} \\
 &= 19H \text{ 十六进制数}
 \end{aligned}$$

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

$$\begin{aligned} &= 9615 \\ \text{误差} &= \frac{\text{计算波特率} - \text{目标波特率}}{\text{目标波特率}} \\ &= \frac{(9615 - 9600)}{9600} \\ &= 0.16\% \end{aligned}$$

各种全双工异步模式的典型波特率和误差值已经计算出来，见附录 5 的表格所示。

使用高波特率（HBRGx = 1）或 16 位 BRG（BxRG16 = 1）有助于降低波特率误差。16 位 BRG 模式用于在高速振荡器频率下取得较缓慢的波特率。

将新值写入 {EUBRGHx:EUBRGLx} 寄存器将导致 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。

14.2.3 自动波特率检测

USARTx 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 RXx 输入提供时钟信号，而是由 RXx 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 BRCTLx 寄存器的 ABRDENx 位置 1 将启动自动波特率校验序列（图 13.3）。当发生 ABRDx 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），EUBRGLx 使用 BRG 计数器时钟递增计数，如图 13.3 所示。在第 8 位周期的末尾将在 RXx 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 EUBRGHx 和 EUBRGLx 寄存器中，ABRDENx 位被自动清 0 而 RCIF 中断标志被置 1。要清除 RCIFx 中断，需要读取 RXSDRx 中的值。RXSDRx 的内容应该被丢弃。校准不使用 EUBRGHx 寄存器的模式时，用户可通过查询 EUBRGHx 寄存器中的 0X00 验证 EUBRGLx 寄存器是否未溢出。

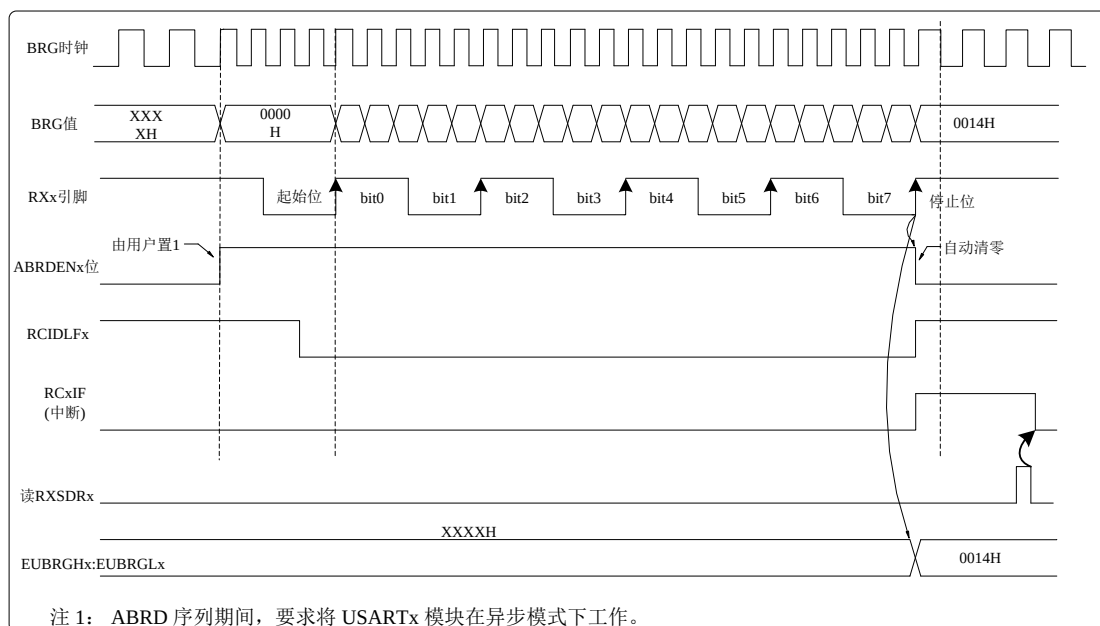


图 14.3 自动波特率检测

BRG 自动波特率时钟由 BxRG16 和 HBRGx 位决定，如表 13-3 所示。在 ABRD 期间，EUBRGHx 和 EUBRGLx 寄存器共同用作 16 位计数器，这与 BxRG16 位的设置无关。在校准波特率周期时，EUBRGHx 和 EUBRGLx 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 WUENx 位和 ABRDENx 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 BRGx 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 EUBRGHx:EUBRGLx 寄存器对的值中减去 1。

表 14-4 BRG 计数器时钟速率

BxRG16	HBRGx	BRG 基时钟	BRG ABRD 时钟
0	0	工作时钟/64	工作时钟/512
0	1	工作时钟/16	工作时钟/128
1	0	工作时钟/16	工作时钟/128
1	1	工作时钟/4	工作时钟/32

注：在 ABRD 序列期间，EUBRGLx 和 EUBRGHx 寄存器都用作 16 位计数器，与 BRG16x 的设置无关。

14.2.4 接收间隔字符时自动唤醒

在休眠模式下，USARTx 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 BRCTLx 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 USARTx 保持在空闲状态，监视唤醒事件，监视行为与 CPU 的工作模式无关。唤醒事件是指 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 RXSDRx 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，并且跟有足够长的时间间隔，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 RXSDRx 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。

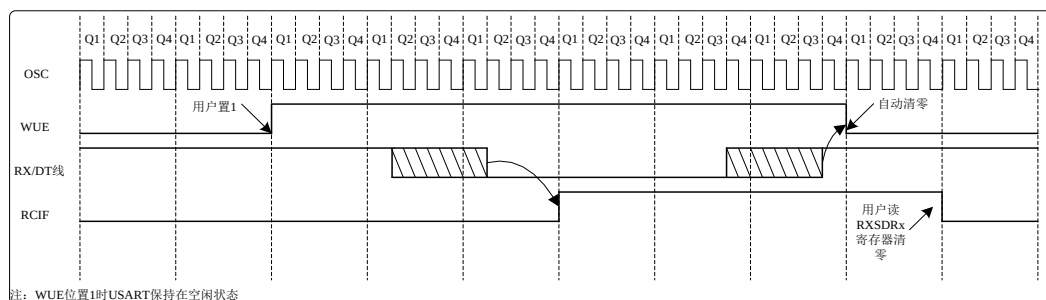


图 14.4 正常工作下的自动唤醒时序

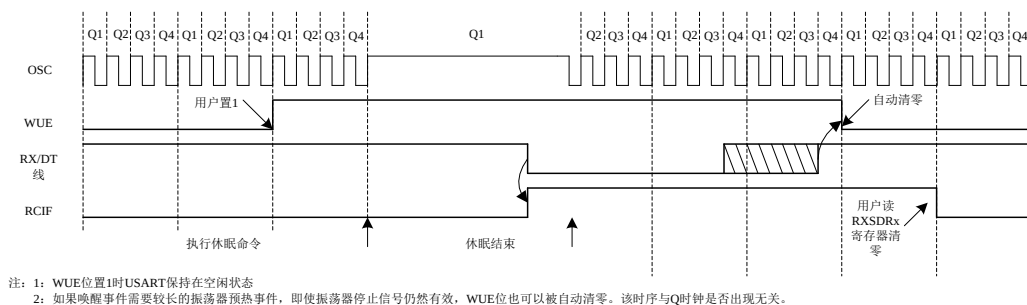


图 14.5 休眠模式下的自动唤醒时序

14.2.5 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 TXSDRx 寄存器的写操作将会启动间隔字符的发送。此时写入 TXSDRx 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 规范中通常同步字符）后将下一个要发送的字节预先装入发送 FIFO 中。TXCTLx 寄存器中的 TXSRx 位则表明发送正在进行还是处于空闲状态。

14.2.5.1 间隔和同步发送序列

以下序列会发送一个报文帧头，包含一个间隔字符和其后的自动波特率同步字节。此序列适用于典型的 LIN 总线器件。

- 1、将 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 TXSDRx 寄存器，启动发送（该值会被忽略）
 - 5、将 55H 写入 TXSDRx 中，以便把同步字符装入发送 FIFO 缓冲中
 - 6、间隔字符发送后，硬件会将 SENDB 位复位。开始发送同步字符
- 当 TXxIF 指示 TXSDRx 为空时，下一个数据会写入 TXSDRx 中。

14.2.5.2 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 RXSDRx 指示接收的数据。假设已将波特率发生器初始化成预期的波特率。当

- RCIF 位置 1
- FRER 位置 1
- RXSDRx 为 00H 时

表明接收到了间隔字符

第二种方法是使用自动唤醒功能。通过使能此功能，USART 将采样 RX/DT 上电平的下两个跳变，产生一个 RCxIF 中断，接收下一个数据字节，之后产生另一个中断。

注意，在间隔字符后，用户通常希望使能自动波特率检测功能。无论使用哪种方法，用

户都可以在 USART 进入休眠模式之前，将 BRCTLx 寄存器的 ABDEN 位置 1.

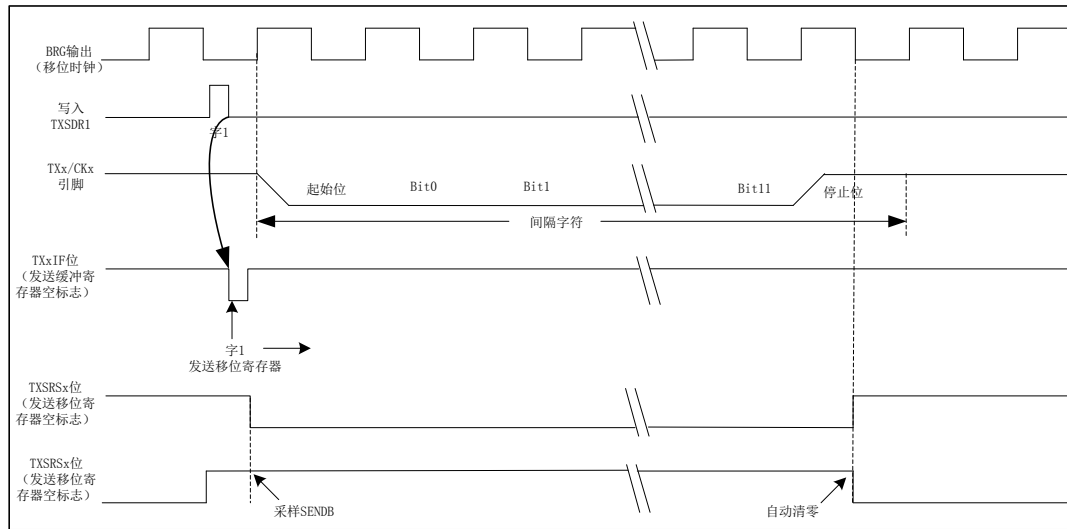


图 14.6 发送间隔字符时序

14.3 USART 全双工模式

在全双工异步通信中，数据是一帧一帧传送的，每一帧的数据格式如图 14.7 所示。

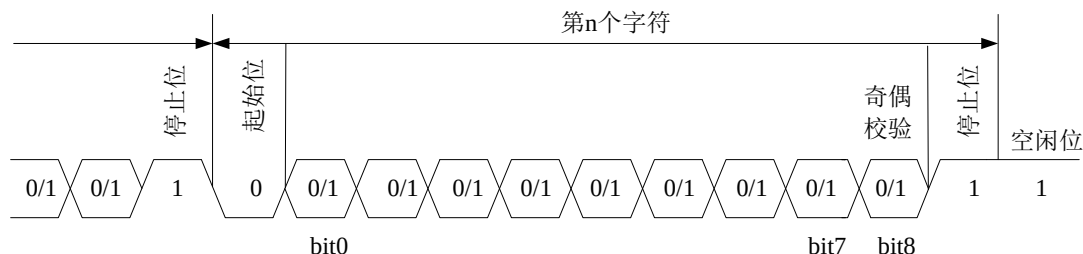


图 14.7 全双工异步通信数据结构图

在帧格式中，一个字符由 4 部分组成：起始位，数据位，奇偶校验位和停止位。

起始位：通常情况下是逻辑 0，占用一位，用来通知接收设备一个等待接收字符的开始。

数据位：8 位。

奇偶校验位：bit8，占用一位，但在字符中可以规定不用奇偶校验位，则这一位可以省去。

停止位：一定为逻辑 1，用来表征字符的结束。停止位可以是 1 位、1.5 位或 2 位。接收端收到停止位后，知道上一字符已经传送完毕，同时，也为接收下一字符作好准备——只要在接收到 0，就是新字符的起始位。若停止位以后不再紧接着传送下一个字符，则使线路电平保持为高电平（逻辑 1），处于空闲状态。这也是全双工异步通信的一大特点。

最常用的数据格式为 8 位。每个发送位的持续时间为 1/(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 14-2 了解波特率配置示例。

USARTx 首先发送和接收低位。USARTx 的发送器和接收器在功能上是相互独立的，但采用相同的数据格式和波特率。硬件不支持奇偶校验，但可以用软件实现（奇偶校验位是第 9 个数据位）。

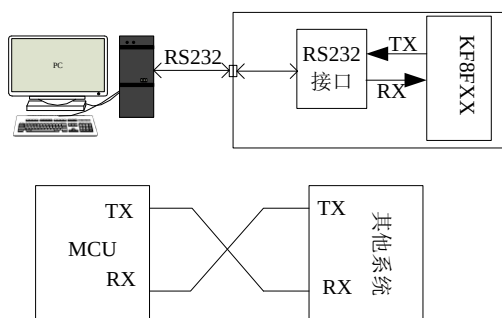


图 14.8 串口通信原理框图

14.3.1 USART 全双工发送操作

USARTx 全双工异步发送操作通过 USARTx 发送器完成。发送器的核心是串行发送移位寄存器（发送移位），该寄存器不能由软件直接访问。发送移位寄存器从 TXSDRx 发送缓冲寄存器获取数据。

通过配置如下三个控制位使能 USARTx 发送器，以用于全双工异步操作：

- 1) TXENx= 1
- 2) SYNCx = 0
- 3) SPENx= 1

假设所有其它 USARTx 控制位处于其默认状态。将 TSCTLx 寄存器的 TXENx 位置 1，使能 USARTx 发送器电路。将 TSCTLx 寄存器的 SYNCx 位清 0，将 USARTx 配置用于全双工异步操作。必须将 TXx/CKx 引脚所在 I/O 口的 ANSx 寄存器对应位清零，即将该口配置为数字口。将 RSCTLx 寄存器的 SPENx 位置 1，使能 USARTx 并自动将 TXx/CKx 的 I/O 引脚配置为输出引脚。

注：

1. 必须将 RXx/DTx 和 TXx/CKx 所在 I/O 口引脚的 ANSx 寄存器对应位清零，将 I/O 口配置为数字口。
2. SPENx 位置 1，会自动将 RXx/DTx 所在 I/O 引脚配置为输入引脚，无论 I/O 口相关 TR 位的状态如何以及 USARTx 接收器使能与否。可以通过普通端口读 RXx/DTx 引脚数据，但无法使用该端口锁存输出数据。
3. 如果 TXENx 使能位置 1，TXIFx 发送器中断标志位会置 1。

14.3.2 发送和控制状态寄存器 TSCTLx

寄存器： TSCTLx: 发送状态和控制寄存器

复位值 0000 0010	bit7						bit0
	CSRSx	TxX9	TXENx	SYNCx	SENDBx	HBRGx	TX9Dx
	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- CSRSx:** 时钟源选择位
全双工异步模式:
无关位
半双工同步模式:
1 = 主模式(由 BRG 内部产生时钟)
0 = 从模式(时钟源来自外部)
- TxX9:** RS485 发送使能位
1 = 选择 RS485 发送
0 = 选择 8 位发送
- TXENx:** 发送使能位
1 = 使能发送
0 = 禁止发送
- SYNCx:** 串行通信模式选择位
1 = 半双工同步模式
0 = 全双工异步模式
- SENDBx:** 发送间隔字符位

- 全双工异步模式:
1 = 在下次发送时发送同步间隔字符(完成后由硬件清 0)
0 = 同步间隔字符发送完成
- 半双工同步模式:
无关位
- HBRGx: 高波特率选择位
全双工异步模式:
1 = 高速
0 = 低速
半双工同步模式:
在此模式下不使用
- TXSRsx: 发送移位寄存器状态位
1 = 发送移位空
0 = 发送移位满
- TX9Dx: 发送数据的第 9 位
可以是地址/数据位或奇偶校验位。

注：在同步模式下，SRCEN/CRCEN 可改写 TREN。

14.3.3 发送数据

向 TXSDRx 寄存器写入一个字符，以启动发送。如果这是第一个字符，或者前一个字符已经完全从发送移位寄存器中移出，TXSDRx 中的数据会立即发送给发送移位寄存器。如果发送移位寄存器中仍保存全部或部分前一字符，新的字符数据将保存在 TXSDRx 中，直到发送完前一字符的停止位为止。然后，在停止位发送完毕后经过一个发送周期，TXSDRx 中待处理的数据将被传输到发送移位寄存器。当数据从 TXSDRx 传输至发送移位寄存器后，立即开始进行起始位、数据位和停止位序列的发送。

只要使能 USARTx 发送器且 TXSDRx 中没有待发送数据，就将 EIF2 寄存器的 TXxIF 中断标志位置 1。换句话说，只有当发送移位寄存器忙于处理字符和 TXSDRx 中有排队等待发送的新字符时，TXxIF 位才处于清 0 状态。写 TXSDRx 时，不立即清 0 TXxIF 标志位。TXxIF 在写指令后的第 2 个指令周期清 0。在写 TXSDRx 后立即查询 TXxIF 会返回无效结果。TXxIF 为只读位，不能由软件置 1 或清 0。

可通过将 EIE2 寄存器的 TXxIE 中断允许位置 1 允许 TXxIF 中断。然而，只要 TXSDRx 为空，不管 TXxIE 允许位的状态如何都会将 TXxIF 标志位置 1。

如果要在发送数据时使用中断，只在有待发送数据时，才将 TXxIE 位置 1。当将待发送的最后一个字符写入 TXSDRx 后，将 TXxIE 中断允许位清 0。

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

14.3.4 全双工发送的设置

1. 设置 BRCTLx 寄存器的 BRxCKs<1:0> 位选择波特率发生器的工作时钟，再根据相应的时钟信号选择相应的时钟源。

2. 初始化EUBRGHx和EUBRGLx这对寄存器以及HBRGx和BxRG16位，以获得所需的波特率（见第14.2节“波特率发生器（BRG）”）。
3. 通过将SYNCx位清0并将SPENx位置1使能全双工异步串口。
4. 如果需要9位发送，将TxX9控制位置1。当接收器被设置为进行地址检测时，将数据位的第9位置1，指示8个最低数据位为地址。
5. 将TXENx控制位置1，使能发送；这将导致TXxIF中断标志位置1。
6. 如果需要中断，将TXxIE中断允许位置1；如果INTCTL寄存器的AIE和PUIE位也置1将立即产生中断。
7. 若选择发送9位数据，第9位应该被装入TX9Dx数据位。
8. 将8位数据装入TXSDRx寄存器开始发送数据。

例14.2 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
    
```

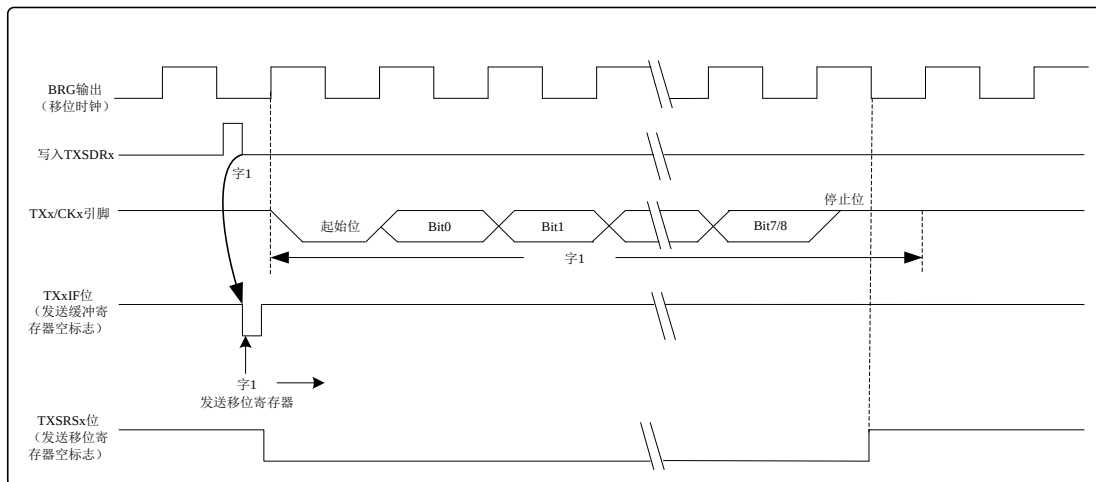


图 14.9 全双工异步发送（两字符间有空闲位）

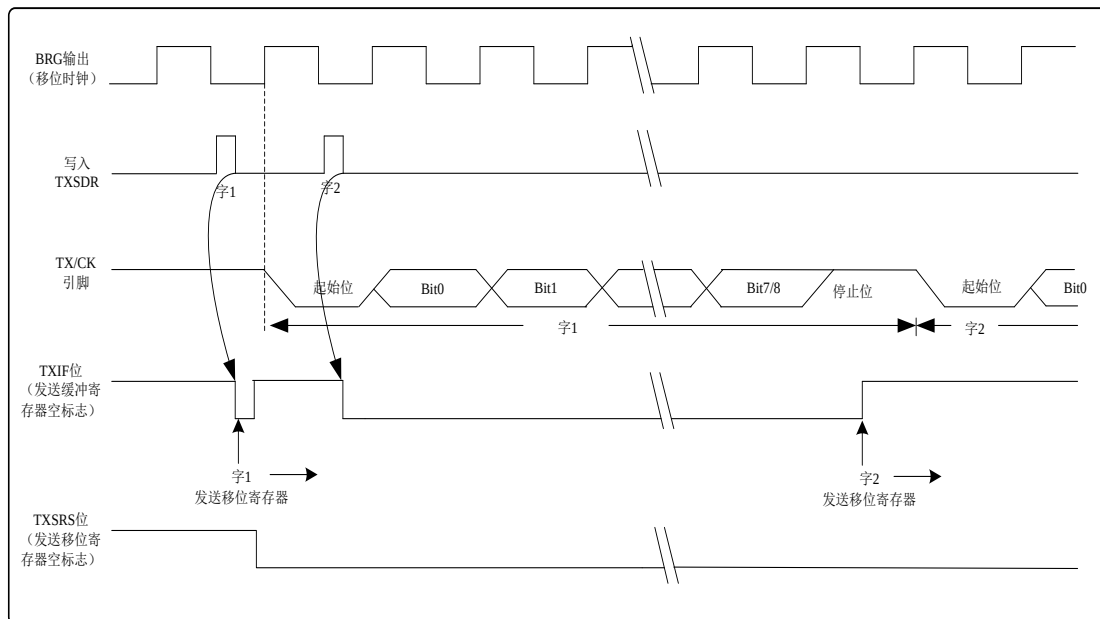


图 14.10 全双工异步发送（一字符接一字符）

14.3.5 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 14.1 给出了接收器的框图。在 RX_x/DT_x 引脚上接收数据和驱动数据恢复电路。数据恢复电路实际上是一个以 16 倍波特率为工作频率的高速移位器，而串行接收移位寄存器（Receive Shift Register, RSR）则以波特率工作。当字符的全部 8 位或 9 位数据位被移入后，立即将它们传输到一个 2 字符的先入先出（FIFO 缓冲器）缓冲器。FIFO 缓冲器允许接收 2 个完整的字符和第 3 个字符的起始位，然后必须由软件将接收到的数据提供给 USART $_x$ 接收器。FIFO 缓冲器和 RSR 寄存器不能直接由软件访问。通过 $RXSDR_x$ 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。

- ◆ $CRXEN_x = 1$
- ◆ $SYNC_x = 0$
- ◆ $SPEN_x = 1$

假设所有其它 USART $_x$ 控制位都处于默认状态。将 $RSCTL_x$ 寄存器的 $CRXEN_x$ 位置 1，使能 USART $_x$ 接收器电路。将 $TSCTL_x$ 寄存器的 $SYNC_x$ 位清 0，配置 USART $_x$ 以用于全双工异步操作。必须将 RX_x/DT_x 引脚所在 I/O 口的 ANS_x 寄存器对应位清零，即将该口配置为数字口。将 $RSCTL_x$ 寄存器的 $SPEN_x$ 位置 1，使能 USART $_x$ 并自动将 RX_x/DT_x 引脚配置为输入引脚。

14.3.6 接收状态和控制寄存器 RSCTLx

寄存器: RSCTLx: 接收状态和控制寄存器x

	bit7						bit0
复位值 0000 000x	SPENx	RxX9	SRXENx	CRXENx	ADRENx	FRERx	OVFERx
	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- SPENx:** 串行口使能位
1 = 使能串行口(将 RXx/DTx 和 TXx/CKx 引脚配置为串行口引脚)
0 = 禁止串行口(保持为复位状态)
- RxX9:** 9 位接收使能位
1 = 选择 9 位接收
0 = 选择 8 位接收
- SRXENx:** 单字节接收使能位
全双工异步模式:
无关位
半双工同步模式——主:
1 = 使能单字节接收
0 = 禁止单字节接收
此位在接收完成后清 0。
半双工同步模式——从:
无关位
- CRXENx:** 连续接收使能位
全双工异步模式:
1 = 使能接收器
0 = 禁止接收器
半双工同步模式:
1 = 使能连续接收, 直到使能位 CRXENx 清 0(CRXENx 改写 SRXENx)
0 = 禁止连续接收
- ADRENx:** 地址检测使能位
RS-485 全双工异步模式(RxX9 = 1):
1 = 使能地址检测、允许中断, 当 RSR<8>置 1 时装入接收缓冲区
0 = 禁止地址检测, 接收所有字节并且第 9 位可作为奇偶校验位
8 位异步模式(RxX9=0):
无关位
- FRERx:** 帧错误位
1 = 帧错误(可通过读 RXSDRx 寄存器刷新该位, 并接收下一个有效字节)
0 = 无帧错误
- OVFERx:** 溢出错误位
1 = 溢出错误(可通过清 0 CRXENx 位来清 0 该位)
0 = 无溢出错误
- RX9Dx:** 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。

14.3.7 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 13.3.2.3 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 USARTx 的接收 FIFO 缓冲器并将 EIE2 寄存器的 RCxIF 中断标志位置 1。通过读 RXSDRx 寄存器将 FIFO 缓冲器最顶端的字符移出 FIFO 缓冲器。

注：如果接收FIFO溢出，则不能再继续接收其他字符，直到溢出条件被清除。请参见第 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 USARTx 接收器且在接收 FIFO 缓冲器中没有未读数据，EIF2 寄存器中的 RCxIF 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 RCxIF 中断：

- ◆ EIE2 寄存器的 RCxIE 中断允许位
- ◆ INTCTL 寄存器的 PUIE 外设中断允许位或低优先级中断允许位 AIEL
- ◆ INTCTL 寄存器的 AIE 全局中断允许位或高优先级中断允许位 AIEH

如果 FIFO 缓冲器中有未读数据，无论中断允许位的状态如何，都会将 RCxIF 中断标志位置 1。

14.3.8 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。帧错误是指未在预期的时间内接收到停止位。由 RSCTLx 寄存器的 FRERx 位获取帧错误状态。FRERx 位代表接收 FIFO 缓冲器最顶端未读字符的状态。因此，必须在读 RXSDRx 寄存器之前读 FRERx 位。FRERx 位为只读位，且只能用于接收 FIFO 缓冲器的最顶端未读字符。帧错误（FRERx = 1）并不会阻止接收更多的字符。无需清 0 FRERx 位。从 FIFO 缓冲器读下一字符会使 FIFO 缓冲器指针前进至下一字符和下一个相应的帧错误。

清 0 RSCTLx 寄存器的 SPENx 位会复位 USARTx，并强制清 0 FRERx 位。清 0 RSCTLx 寄存器的 CRXENx 位不影响 FRERx 位。帧错误本身不会产生中断。

注：如果接收FIFO缓冲器中所有接收到的字符都有帧错误，重复读RCSDRx不会清零FRERx 位。

接收 FIFO 缓冲器可以保存 2 个字符。但如果在访问 FIFO 缓冲器之前，接收到完整的第 3 个字符，则会产生溢出错误。此时，RSCTLx 寄存器的 OVFERx 位会置 1。可以读取 FIFO 缓冲器内的字符，但是在错误清除之前，不能再接收其它字符。可以通过清 0 RSCTLx

寄存器的 CRXENx 位或通过清 0 RSCTLx 寄存器的 SPENx 位使 USARTx 复位来清除错误。

14.3.9 地址检测

当多个接收器共享同一传输线时（如在 RS-485 系统中），可使用特殊地址检测模式。将 RSCTLx 寄存器的 ADRENx 位置 1，使能地址检测模式。地址检测要求接收 9 位字符。使能地址检测后，只有第 9 位数据位被置 1 的字符可以被传输到接收 FIFO 缓冲器，当接收到的地址数据与 UARTMTHx 寄存器内预设的值匹配时，RCxIF 中断标志位将置 1；所有其它字符将被忽略。ADRENx 位不会自动清零，当检测到地址匹配后，如果要进行数据传输，需要将 ADRENx 位手动清零。

寄存器: UARTMTHx: UART地址匹配设置寄存器x

复位值 0000 0000	bit7							bit0
	ADMx7	ADMx6	ADMx5	ADMx4	ADMx3	ADMx2	ADMx1	ADMx0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ADMx<7:0>: UARTx地址匹配功能设置位

注：UARTMTH1寄存器地址为12DH；UARTMTH2寄存器地址为247H。

14.3.10 全双工接收的设置

1. 设置BRCTLx寄存器的BRxCK<1:0>位选择波特率发生器的工作时钟，再根据相应的时钟信号选择相应的时钟源。
2. 初始化{EUBRGHx:EUBRGLx}这对寄存器以及 HBRGx 和 BxRG16 位，以获得所需的波特率（见第 14.2 节“波特率发生器（BRG）”）。
3. 将 SPENx 位置 1，使能串行端口。必须清 0 SYNCx 位以执行全双工异步操作。
4. 如果需要中断，将 EIE2 寄存器中的 RCxIE 位和 INTCTLx 寄存器的 AIE 和 PUIE 位置 1。如果使用中断优先级则将 IPEN 和 PTX 均置 1
5. 如果需要接收 9 位数据，将 RxX9 位置 1。
6. 将 CRXENx 位置 1 使能接收。
7. 当一个字符从 RSR 传输到接收缓冲器时，将 RCxIF 中断标志位置 1。如果 RCxIE 中断允许位也置 1 还将产生中断。
8. 读 RSCTLx 寄存器获取错误标志位和第 9 位数据位（如果使能 9 位数据接收）。
9. 读 RXSDRx 寄存器，从接收缓冲器获取接收到的 8 个低数据位。
10. 如果发生溢出，通过清 0 CRXENx 接收器使能位清 0 OVFERx 标志。

例14.3 接收(USART1为例)

LOOP_USART

```

NOP
NOP
MOVB #0X01                                ;切换至1区
SET   RSCTL1,CRXEN1                       ;使能接收器
JNB   RSCTL1,OVFER1                       ;检测溢出错误
CLR   RSCTL1, CRXEN1
MOVB #0X00                                ;切换至0区
JB    EIF2, RXIF1                         ;接收数据的标志
JMP   LOOP_USART
CLR EIF2, RXIF1
MOVB #0X01                                ;切换至1区
MOV R0, RXSDRx
MOV UART_TEMP, R0                        ;存储接收到的数据

```

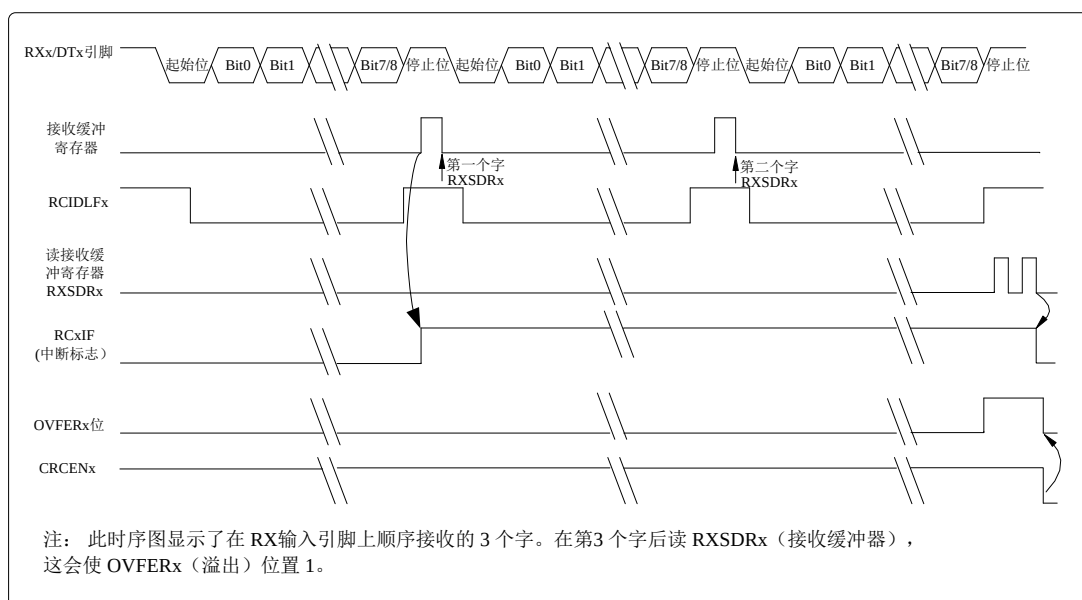


图 14.11 全双工异步接收

14.3.11 RS-485 发送/接收

USARTx 支持 RS-485 模式 9 位发送。当 TSCTLx 寄存器的 TxX9 位置 1 时，USARTx 将移出每个待发送字符的 9 位。TSCTLx 寄存器的 TX9Dx 位为发送数据第 9 位，即最高数据位。当发送 9 位数据时，必须在将 8 个最低位写入 TXSDRx 之前，写 TX9Dx 数据位。在写入 TXSDRx 寄存器后会立即将 9 个数据位传输到发送移位寄存器。

使用多个接收器时可使用特殊的 9 位地址模式。请参见第 13.3.2.4 节“地址检测”获得有关地址模式的更多信息。

USARTx 支持 RS-485 模式 9 位接收。将 RSCTLx 寄存器的 RxX9 位置 1 时，USARTx 将接收到的每个字符的 9 位移入 RSR。RSCTLx 寄存器的 RX9Dx 位是接收 FIFO 缓冲器顶端未读字符的第 9 位，同时也是最高数据位。当从接收 FIFO 缓冲器读取 9 位数据时，必须在读 RXSDRx 中的低 8 位之前，读取 RX9Dx 数据位。

14.3.11.1RS-485 9 位地址检测模式设置

要设置使能地址检测的全双工异步接收:

1. 初始化 EUBRGHx 和 EUBRGLx 这对寄存器以及 HBRGx 和 BxRG16 位, 以获得所需的波特率 (见第 13.2 节“波特率发生器 (BRG)”)。
2. 将 SPENx 位置 1, 使能串行端口。必须清 0 SYNCx 位以执行全双工异步操作。
3. 如果需要中断, 将 EIE2 寄存器中的 RCxIE 位和 INTCTL 寄存器的 AIE 和 PUIE 位置 1。如果使用优先级中断, 则将 IPEN 和 PRX 均置 1。
4. 将 RxX9 位置 1, 使能 9 位数据接收。
5. 将 ADRENx 位置 1, 使能地址检测。
6. 将 CRXENx 位置 1 使能接收。
7. 当一个第 9 位置 1 的字符从 RSR 传输到接收缓冲器时, 将 RCxIF 中断标志位置 1。如果 RCxIE 中断允许位也置 1 还将产生中断。
8. 读 RSCTLx 寄存器获取错误标志位。第 9 个数据位始终置 1。
9. 读 RXSDRx 寄存器, 从接收缓冲器获取接收到的 8 个低数据位。由软件判断此地址是否为本地器件的地址。
11. 如果发生溢出, 通过清 0 CRXENx 接收器使能位清 0 OVFERx 标志。
11. 如果是对当前器件寻址, 将 ADRENx 位清 0 以允许所有接收到的数据进入接收缓冲器并产生中断。

14.3.12 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 13.2.3 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。

14.4 USART 半双工模式

半双工同步串行通信通常用在具有一个主控制器件和一个或多个从动器件的系统中。主控制器件包含产生波特率时钟所必需的电路, 并为系统中的所有器件提供时钟。从动器件可以使用主控时钟, 因此无需内部时钟发生电路。

在半双工同步模式下, 有 2 条信号线: 双向数据线和时钟线。从动器件使用主控制器件提供的时钟, 将数据串行移入或移出相应的接收和发送移位寄存器。半双工是指: 主控制器件和从动器件都可以接收和发送数据, 但是不能同时进行接收或发送。USARTx 既可以作为主控制器件, 也可以作为从动器件。

半双工同步模式发送无需使用起始位和停止位。

14.4.1 USARTx 半双工主控模式

下列位用来将 USARTx 配置为半双工同步主控操作:

- SYNCx = 1
- CSRSx = 1

- SRXEN_x = 0（用于发送）；SRXEN_x = 1（用于接收）
- CRXEN_x = 0（用于发送）；CRXEN_x = 1（用于接收）
- SPEN_x = 1

将 TSCTL_x 寄存器的 SYNC_x 位置 1，可将 USART_x 配置用于半双工同步操作。将 TSCTL_x 寄存器的 CSRS_x 位置 1，将器件配置为主控器件。将 RSCTL_x 寄存器的 SRXEN_x 和 CRXEN_x 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_x 寄存器的 SPEN_x 位置 1，使能 USART_x。注意，配置 USART_x 模块前，必须将 RX_x/DT_x 和 TX_x/CK_x 引脚所在 I/O 口的 ANS_x 寄存器对应位清零，即将该口配置为数字口。

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_x/CK_x 引脚发送时钟信号。当 USART_x 被配置为半双工同步发送或接收操作时，TX_x/CK_x 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_x 寄存器的 SCKPS_x 位选择时钟极性。将 SCKPS_x 位置 1 将时钟空闲状态设置为高电平。当 SCKPS_x 位置 1 时，数据在每个时钟的下降沿发生改变。清 0 SCKPS_x 位，将时钟空闲状态设置为低电平。当清 0 SCKPS_x 位时，数据在每个时钟的上升沿发生改变。

14.4.2 半双工主控发送

由器件的 RX_x/DT_x 引脚输出数据。当 USART_x 配置为半双工同步主控发送操作时，器件的 RX_x/DT_x 和 TX_x/CK_x 输出引脚自动使能。

向 TXSDR_x 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_x 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_x 中的数据会被立即传输到发送移位寄存器。当字符从 TXSDR_x 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寄存器中，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_x 和 EUBRGL_x 这对寄存器以及 HBRG_x 和 BxRG16 位，以获得所需的波特率（见第 14.2 节“波特率发生器”）。
2. 将 SYNC_x、SPEN_x 和 CSRS_x 位置 1，使能半双工同步主控串行端口。
3. 将 SRXEN_x 和 CRXEN_x 位清 0，禁止接收模式。
4. 将 TXEN_x 位置 1 使能发送模式。
5. 如果需要发送 9 位字符，将 TxX9 置 1。
6. 若需要中断，将 EIE2 寄存器中的 TX_xIE 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_x 和 PTX_x 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_x 位。
8. 通过将数据装入 TXSDR_x 寄存器启动发送。（请参考例 14.2）

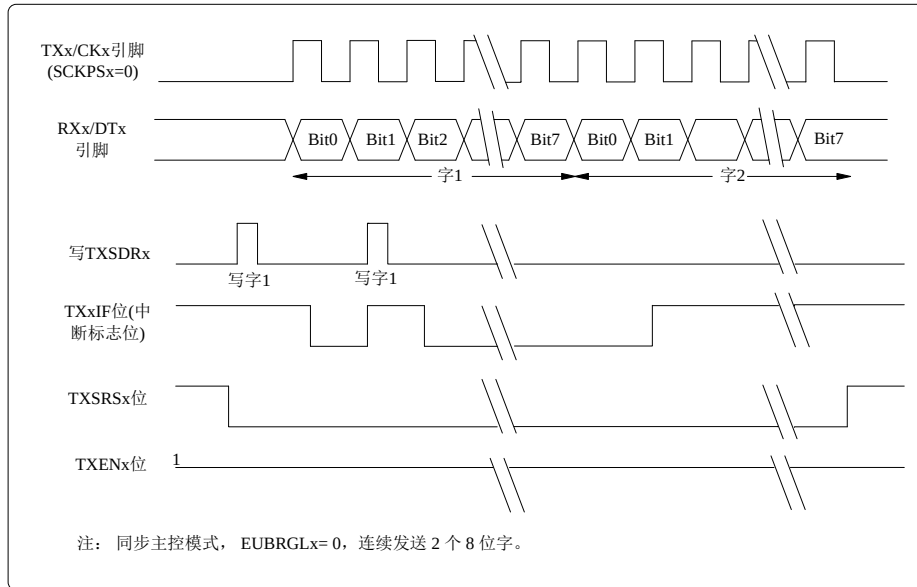


图 14.12 半双工同步发送 (SCKPSx=0)

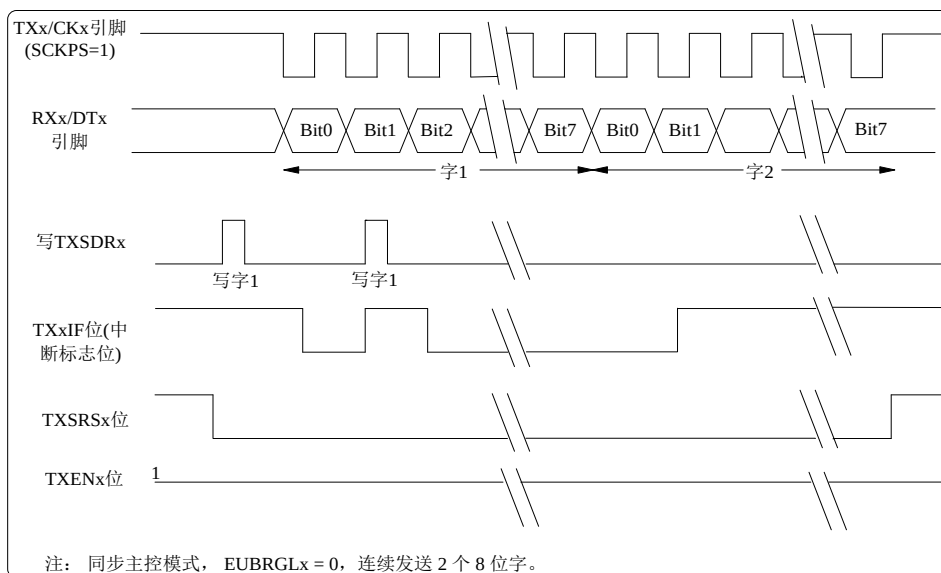


图 14.13 半双工同步发送(SCKPSx=1)

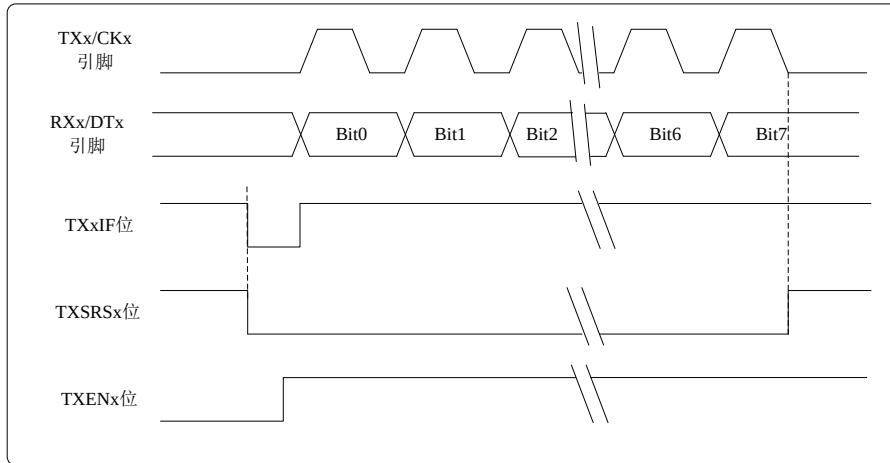


图 14.14 半双工同步发送（通过 TXENx）

14.4.3 半双工主控接收

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

当将 SRXENx 置 1，CRXENx 位清 0 时，一个单字符中有多少数据位就只能产生多少时钟周期。一个字符传输结束后，自动清 0 SRXENx 位。当 CRXENx 置 1 时，将产生连续时钟，直到清 0 CRXENx 为止。如果 CRXENx 在一个字符的传输过程中清 0，则 CK 时钟立即停止，并丢弃该不完整的字符。如果 SRXENx 和 CRXENx 都置 1，则当第一个字符传输完成时，SRXENx 位被清 0，CRXENx 保持。

将 SRXENx 或 CRXENx 位置 1，启动接收。在 TXx/CKx 时钟引脚信号的下降沿采样 RXx/DTx 引脚上的数据，并将采样到的数据移入接收移位寄存器（RSR）。当 RSR 接收到一个完整字符时，将 RCxIF 位置 1，字符自动移入 2 字节接收 FIFO 缓冲器。接收 FIFO 缓冲器中最顶端字符的低 8 位可通过 RXSDRx 读取。只要接收 FIFO 缓冲器中仍有未读字符，则 RCxIF 位就保持置 1 状态。

半双工同步数据传输使用与数据线通读的独立时钟线。配置为从器件的器件接收 TXx/CKx 线上的时钟信号。当器件被配置为半双工同步从发送或接收操作时，TXx/CKx 引脚的输出驱动器自动被禁止。串行数据位在时钟信号的前沿改变，以确保其在每个时钟的后沿有效。每个时钟周期只能传输一位数据，因此有多少数据位要传输就必须接收多少个时钟。

接收 FIFO 缓冲器可以保存 2 个字符。在读 RXSDRx 以访问 FIFO 缓冲器之前，若完整地接收到第 3 个字符，则产生溢出错误。此时，RSCTLx 寄存器的 OVFERx 位会置 1。FIFO 缓冲器中先前的数据不会被改写。可以读取 FIFO 缓冲器内的 2 个字符，但是在错误被清除前，不能再接收其它字符。只能通过清除溢出条件，将 OVFERx 位清 0。如果发生溢出时，SRXENx 位为置 1 状态，CRXENx 位为清 0 状态，则通过读 RXSDRx 寄存器清除错误。如果溢出时，CRXENx 为置 1 状态，则可以清 0 RSCTLx 寄存器的 CRXENx 位或清 0 SPENx 位以复位 USARTx，从而清除错误。

半双工主控接收设置:

1. 用正确的波特率初始化 EUBRGHx:EUBRGLx 寄存器。按需要将 HBRGx 和 BRG16x 位置 1 或清 0，以获得所需的波特率。
2. 将 SYNCx、SPENx 和 CSRSx 位置 1 使能半双工同步主控串行端口。
3. 确保将 CRXENx 和 SRXENx 位清 0。
4. 如果使用中断, 将 INTCTL 寄存器的 AIE 和 PUIE 位置 1, 并将中断使能寄存器的 RCxIE 位也置 1。如果使用优先级中断, 则将 IPEN 和 PRXx 均置 1。
5. 如果需要接收 9 位字符, 将 RxX9 位置 1。
6. 将 SRXENx 位置 1, 启动接收, 或将 CRXENx 位置 1 使能连续接收。
7. 当字符接收完毕后, 将 RCxIF 中断标志位置 1。如果允许位 RCxIE 置 1, 还会产生一个中断。
8. 读 RSCTLx 寄存器以获取第 9 个数据位 (使能 9 位接收时), 并判断接收过程中是否产生错误。
9. 读 RXSDRx 寄存器获取接收到的 8 位数据。

如果产生溢出错误, 清 0 RSCTLx 寄存器的 CRXENx 位或清 0 SPENx 以复位 USARTx 来清除错误。(请参考第 14.3.2.5 节例 14.3)

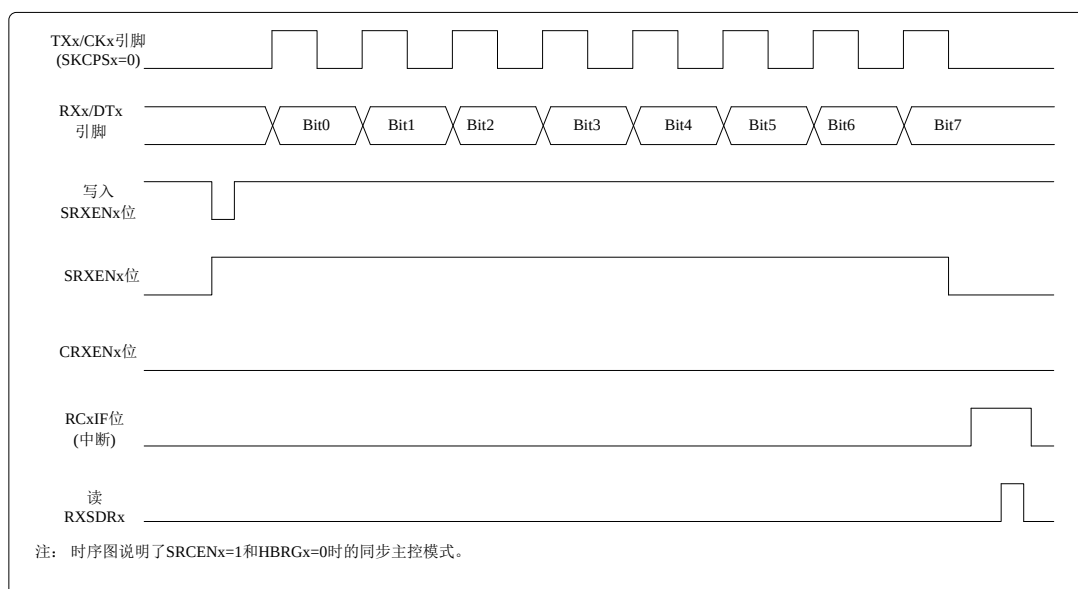


图 14.15 半双工同步接收 (主控模式, SRXENx=1, SCKPSx=0)

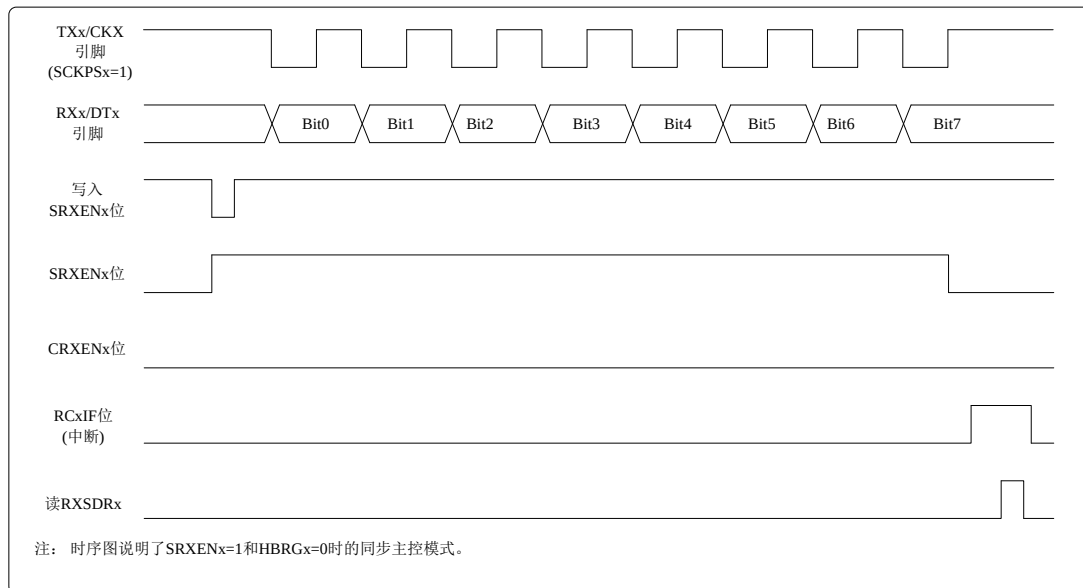


图 14.16 半双工同步接收（主控模式，SRXENx=1，SCKPSx=1）

14.4.4 USART 半双工从动模式

下列位用来将 USARTx 配置为半双工同步从动操作：

- SYNCx=1
- CSRSx=0
- SRXENx=0（用于发送）；SRXENx = 1（用于接收）
- CRXENx=0（用于发送）；CRXENx = 1（用于接收）
- SPENx=1

将 TSCTLx 寄存器的 SYNCx 位置 1，可将器件配置用于半双工同步操作。将 TSCTLx 寄存器的 CSRSx 位置 1，将器件配置为从动器件。将 RSCTLx 寄存器的 SRXENx 和 CRXENx 位清 0，以确保器件处于发送模式，否则器件将被配置为接收模式。将 RSCTLx 寄存器的 SPENx 位置 1，使能 USARTx。注意，配置 USARTx 模块前，必须将 RXx/DTx 和 TXx/CKx 引脚所在 I/O 口的 ANSx 寄存器对应位清零，即将该口配置为数字口。

14.4.5 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 14.4.1.1 节“半双工主控发送”）。

如果向 TXSDRx 写入 2 个字，然后执行 IDLE 指令，则会出现下列情况：

1. 第一个字立即传输到发送移位寄存器并进行发送。
2. 第二个字留在 TXSDRx 寄存器中。
3. TXxIF 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 TXxIF 置 1。
5. 如果 PUIE 和 TXxIE 位都置 1，则由中断将器件从休眠模式唤醒，然后执行下一条指令。如果 AIE 位也置 1，程序将调用中断服务程序。

USART 半双工从动发送设置：

1. 将 SYNCx 和 SPENx 位置 1 并将 CSRSx 位清 0。
 2. 将 CRXENx 和 SRXENx 位清 0。
 3. 如果使用中断，将 INTCTL 寄存器的 AIE 和 PUIE 位置 1，并将中断使能寄存器的 TXxIE 位也置 1。如果使用优先级中断，则将 IPEN 和 PTXx 均置 1。
 4. 如果需要发送 9 位数据，将 TxX9 位置 1。
 5. 将 TXENx 位置 1 使能发送。
 6. 若选择发送 9 位数据，将最高位写入 TX9Dx 位。
- 将低 8 位数据写入 TXSDRx 寄存器开始传输。（请参考例 14.2）

14.4.6 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 14.4.1.2 节“半双工主控接收”）。

- 休眠模式
- CRXENx 位总是置 1，因此接收器不能进入空闲状态。
- SRXENx 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 CRXENx 位置 1，则在休眠模式仍可接收字符。RSR 寄存器接收到字后，就会立即将接收到的数据传输到 RXSDRx 寄存器。如果将 RCxIE 允许位置 1，则产生的中断将使器件从休眠模式唤醒，然后执行下一条指令。如果 AIE 位也置 1，则程序将跳转到中断向量处执行。

半双工从动接收设置：

1. 将 SYNCx 和 SPENx 位置 1 并将 CSRS 位清 0。
2. 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 RCxIE 位也置 1。如果使用优先级中断，则将 IPENx 和 PRXx 均置 1。
3. 如果需要接收 9 位字符，将 RxX9 位置 1。
4. 将 CRXENx 位置 1，使能接收。
5. 当接收完成后，将 RCxIF 位置 1。如果 RCxIE 已置 1，还会产生一个中断。
6. 如果使能 9 位模式，从 RSCTLx 寄存器的 RX9Dx 位获取最高位。
7. 读 RXSDRx 寄存器，从接收 FIFO 缓冲器获取接收到的 8 个低数据位。
8. 如果产生溢出错误，清 0 RSCTLx 寄存器的 CRXENx 位或清 0 SPENx 位以复位 USARTx 来清除错误。（请参考例 14.3）

14.4.7 USART 半双工 RS-485 模式

请参考第 14.3.3 节 RS-485 发送/接收。

14.5 USART 工作在休眠模式下

当 USART 的工作时钟选择为低频时钟且打开 USART 的休眠使能位时，将允许 USART 工作在休眠模式下。

14.5.1 USART 休眠使能寄存器 USLPEN

寄存器： USLPENx:USARTx 休眠时能寄存器x

	bit7						bit0
复位值	SLPENx	-	-	-	-	-	-
0----	R/W	U	U	U	U	U	U

SLPENx: USARTx 休眠使能控制位

SLPENx=1 休眠使能，允许 USARTx 在休眠模式下继续收发数据

SLPENx=0 禁止 USARTx 在休眠模式下收发数据

注：x=1 或 2，USLPEN1 寄存器地址为 127H，USLPEN2 寄存器地址为 248H。

14.6 USART 单线通信模式

USART 单线通信模式（半双工异步模式）通过 UPINSETx 寄存器的 USLMx 位置 1 使能；在使能 USART 单线通信模式之前，须确保 TSCTLx 寄存器的 SYNCx 位保持清零状态。

USART 单线通信模式下，USART 相关引脚特性如下：

- RX/DT 引脚不再使用，总是被释放，即可作为通用 I/O 口使用；
- TX/CK 引脚作为数据通信脚；
- 当 USART 配置为单线通信接收时，TX/CK 引脚一直处于接收状态；
- 当 USART 配置为单线通信发送时：
 - 当有数据发送时，TX/CK 引脚处于数据发送状态；
 - 当无数据发送时（空闲状态），TX/CK 引脚被释放，表现为通用 I/O 口，因此在使用前需要将 TX/CK 引脚配置为数字输出口，并输出高电平（根据 USART 通信协议，当 USART 处于空闲状态时，TX/CK 被拉高）。

用户可通过软件来管理线上冲突。

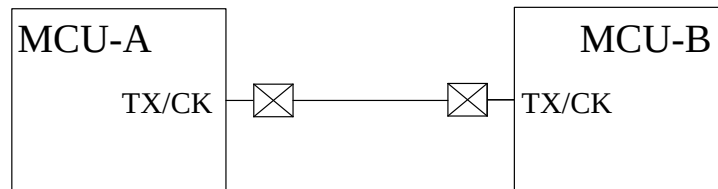


图 14.17: USART 单线通信外接连线示意图

14.6.1 USART 引脚配置寄存器 UPINSETx

UPINSETx 寄存器主要用于 USARTx 的单线通信模式使能和相关引脚配置；为了方便用户的实际使用，USARTx 的 RXx/DTx 引脚和 TXx/CKx 引脚所在位置有两组可选，用户可通过 UPINSETx 寄存器的 UPSELx 位选择。

注：UPSELx 位为 USARTx 模块总控制位，而非单线通信模式下才能使用。

寄存器： UPINSET1:USART1 引脚配置寄存器(地址：150H)

复位值 0000 0000		bit7						bit0	
		USLM1	UPSEL1	-	-	-	-	-	-
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

USLM1: USART1 单线通信模式使能位

USLM1 =1 使能串口单线通信模式

USLM1 =0 禁止串口单线通信模式

UPSEL1: USART1 通信引脚配置位

UPSEL1 =1 USART1 通信引脚使能，RX1/DT1=P4.1 TX1/CK1=P4.0

UPSEL1 =0 USART1 通信引脚未使能

寄存器：UPINSET2:USART2引脚配置寄存器(地址：23FH)

复位值 0000 0000		bit7							bit0
	USLM2	UPSEL2	-	-	-	-	-	-	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

USLM2: USART2 单线通信模式使能位

USLM2 =1 使能串口单线通信模式

USLM2 =0 禁止串口单线通信模式

UPSEL2: 保留位

注：请勿将 UPSEL2 位置 1，否则将导致 USART2 模块无法使用。

14.7 7816 模式

7816 模式基于 ISO/IEC 7816-3 标准，利用此功能可与其他 7816 设备进行通信。按照 7816 协议标准，7816 基本接口时序如图所示：

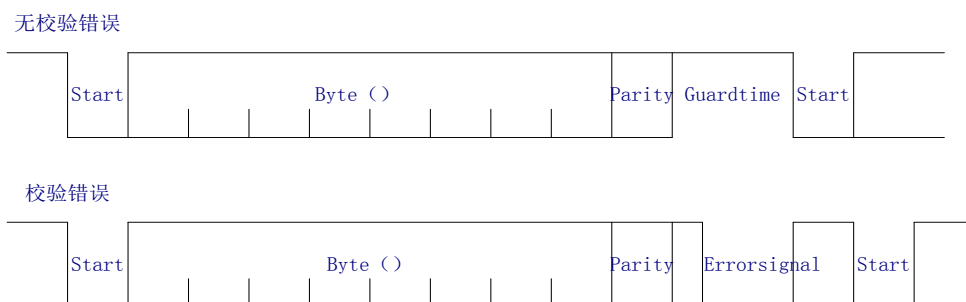


图 14.18 7816 接口时序

- 1, 一个起始位后跟 8 个数据位以及一个校验位，以 2etu 的 guard time 结束。
- 2, 第 10.5 个 etu 接收电路校验接收数据，如果正确，插入 2etu 的 guard time，确保数据长度为 12etu，完成数据发送；若校验错误，则在第 10.5etu 拉低 IO，产生 error signal。error signal 长度可从 1etu、2etu、1.5etu 中选择。
- 3, 第 11 个 etu 时发送电路未采样到 error signal，则说明发送数据正确，数据发送完成。
- 4, 若第 11 个 etu 发送电路采样到 error signal，则说明发送数据错误，等待 2 个 etu 后重发数据。

14.7.1 7816 模式发送

14.7.1.1 发送设置

通过配置如下控制位使能 USART2 发送器，以用于 7816 模式发送操作：

- 1) U7816EN2 = 1 使能 7816 模式
- 2) SPEN2 = 1 使能串口
- 3) SYNC2 = 0 选择异步模式
- 4) CLKOUT2 = 1 使能 7816 时钟输出
- 5) TxX9 = 1 选择 9 位数据发送

- 6) TX9DSEL2 = 1 选择自动生成奇偶校验码
- 7) STOP2 = 0 选择 2 位停止位
- 8) TXEN2 = 1 使能发送端
- 9) TPAR2、TINV2、TCNV2、BGTEN2、PSEL2、TREPEN2、TREP2、EGT2 等控制寄存器视情况进行选择
- 10) 其他设置请参考 14.3.1.3

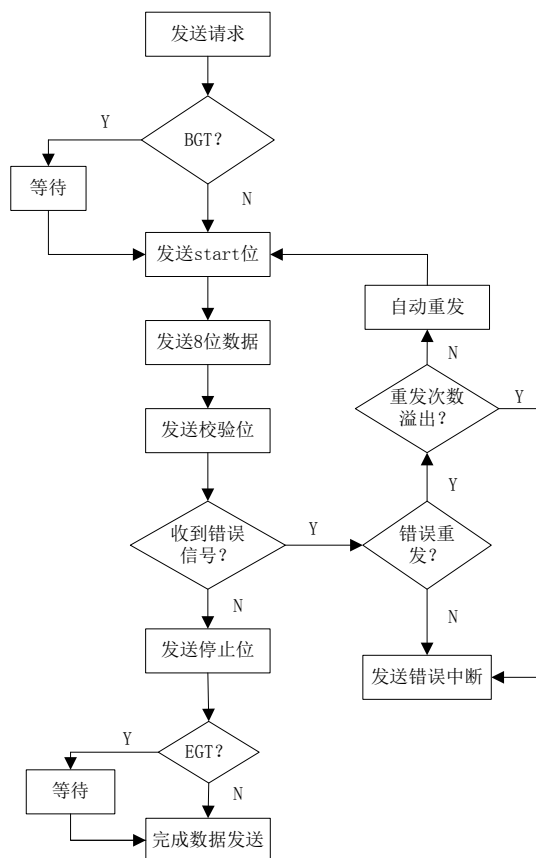


图 14.19 数据发送流程

14.7.2 7816 模式接收

通过配置如下控制位使能 USART2 接收器，以用于 7816 接收操作：

- 1) CRXEN2 = 1 使能接收端
- 2) U7816EN2 = 1 使能 7816 模式
- 3) SPEN2 = 1 使能串口
- 4) SYNC2 = 0 选择异步模式
- 5) R2X9 = 1 选择 9 位数据接收
- 6) CLKOUT2 = 1 使能 7816 时钟输出

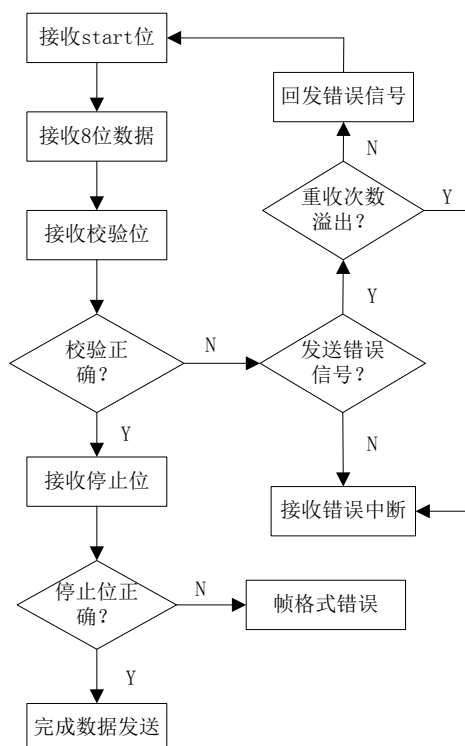


图 14.20 数据接收流程

14.7.3 7816 模式寄存器

14.7.3.1 7816 发送控制寄存器 U7816TXCTL2

寄存器: U7816TXCTL2: 7816发送控制寄存器2(地址:242H)

复位值	bit7						bit0	
0001 0000	U7816EN2	CLKOUT2	TX9DSEL 2	STOP2	TPAR2	TINV2	TCONV2	BGTEN2
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- U7816EN2: 7816 模式使能控制
 1 = 使能 7816 模式
 0 = 禁止 7816 模式
- CLKOUT2: 7816 时钟输出使能控制
 1 = 使能时钟输出
 0 = 禁止时钟输出
- TX9DSEL2: TX9D 位数据选择
 1 = 选择自动生成奇偶校验
 0 = 选择软件输入
- STOP2: 停止位宽度选择
 1 = 停止位宽度为 1etu
 0 = 停止位宽度为 2etu

- TPAR2: 发送时奇偶校验选择
 1 = 奇校验
 0 = 偶校验
- TINV2: 7816 数据发送编码方式选择
 1 = 发送反相电平
 0 = 发送正相电平
- TCONV2: 发送次序选择
 1 = 先发送 MSB
 0 = 先发送 LSB
- BGTEN2: BGT(block guard time)(块保护时间, 为两个相对方向发送的连续字符的起始沿之间的最短时间, 因此一个已接收块的最后一个字符与一个被传输块的第一个字符之间的延迟至少应为 BGT)控制位, 控制接收到发送之间是否插入 BGT
 1 = 插入 BGT, 宽度为 22etu
 0 = 不插入 BGT

注: (1) TX9DSEL2、STOP2、TPAR2、TCONV2 在 7816 模式禁止时也可使用;
 (2) 第九位需要软件输入时, 必须先把 TX9DSEL2 置 0, 再输入 TX9 的值, 否则可能会出错;
 (3) 要使用 BGTEN2 功能时, 必须保证 TXEN2 同时也使能, 否则 BGT 不能正常工作。

14.7.3.2 7816 接收控制寄存器 U7816RXCTL2

寄存器: U7816RXCTL2: 7816接收控制寄存器2(地址:241H)

bit7				bit0			
复位值 0000 0000	ERSW21	ERSW20	RPAR2	RINV2	RCONV2	-	PAREF2
	R/W	R/W	R/W	R/W	R/W	U	R

ERSW2[1:0]: error signal 宽度选择

01 = 1etu

00 = 2etu

1X = 1.5etu

RPAR2: 接收时奇偶校验选择

1 = 奇校验

0 = 偶校验

RINV2: 数据接收编码方式选择

1 = 接收反相电平

0 = 接收正相电平

RCONV2: 接收次序选择

1 = 先接收 MSB

0 = 先接收 LSB

PAREF2: 奇偶校验错误标志位

1 = 发送或接收上以字节数据奇偶校验错误

0 = 发送或接收上以字节数据奇偶校验正确

注: RINV2、RCONV2 在 7816 模式禁止时也可使用。

14.7.3.3 7816 控制寄存器 U7816CTL2

寄存器: U7816CTL2: 7816控制寄存器2(地址:240H)

复位值 0000 0000	bit7						bit0	
	PSEL2	-	TREPEN2	TREP21	TREP20	RREPEN2	RREP21	RREP20
	R/W	U	R/W	R/W	R/W	R/W	R/W	R/W

PSEL2: 数据通道选择

1 = 选择通道 1 (RX21/DT21)

0 = 选择通道 0 (RX20/DT20)

TREPEN2: 重发使能控制

1 = 收到校验错误信号后重发, 达到最大次数后进入中断

0 = 收到校验错误信号后禁止重发, 直接中断

TREP2[1:0]: 重发的最大次数

TREP2=1-3

RREPEN2: 重收使能控制

1= 奇偶校验错误后重收数据, 达到最大次数后进入中断

0= 奇偶校验错误后直接进入中断

RREP2[1:0]: 重接收的最大次数

RREP2=1-3

14.7.3.4 预分频控制寄存器 CLKDIV2

寄存器: CLKDIV2: 预分频控制寄存器2(地址:243H)

复位值 0000 0000	bit7						bit0	
	CLKDIV2	CLKDIV2	CLKDIV2	CLKDIV2	CLKDIV2	CLKDIV2	CLKDIV2	CLKDIV2
	7	6	5	4	3	2	1	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CLKDIV2[7:0]:7816 工作时钟和引脚输出时钟控制

当 CLKDIV1 的值不为 0 时, 7816 工作时钟 F_{7816} 和系统时钟 SCLK 之间的关系为:

$$F_{7816} = SCLK / [2 (CLKDIV2 + 1)]$$

当 CLKDIV2 的值为 0 时, F_{7816} 和 SCLK 关系为: $F_{7816} = SCLK$

注: 7816 协议规定 7816 工作时钟频率范围为 1-5M。

14.7.3.5 EGT 控制寄存器 EGTCTL2

寄存器: EGTCTL2: EGT控制寄存器2(地址:244H)

复位值 0000 0000	bit7						bit0	
	EGT27	EGT26	EGT25	EGT24	EGT23	EGT22	EGT21	EGT20
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

EGT2[7:0]: 发送时插入的 EGT(extra guard time)宽度 (单位 etu)

0 = 发送时不插入 EGT

1-255 = 发送时插入 EGT, 宽度为 EGT2 的值

注: EGTCTL2 在发送无校验错误时正常工作, 若有校验错误时无论为何值均不插入 EGT。

15 参考电压模块 FVR

KF8L15Z20XX 内部有一个参考电压模块，使能该功能后，通过引脚 P5.2/VREFOUT 可输出稳定的 2V/3V/4V 参考电压（VREFOE=1），精度为 5%。如果内部使用参考电压 VREFOUT 时（用作 AD 参考电压和比较器参考电压），不需要将寄存器 VREFCTL 的 VREFOE 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(VREFCTL<1,3>)，将 VREFEN(VREFCTL.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 VREFOE(VREFCTL.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。

用户如果要用到内部 2V/3V/4V 参考电压，需先读出 27FAH 和 27F9H 地址的参考电压校准值(例 7.1)，分别送到 VREFCAL0 和 VREFCAL1 寄存器，然后根据需要设置 VREFCTL 中的 VREFEN 和 VREFOE 位。

例7.1: 读内部参考电压校准值

```
MOVP #0X20                ;切换到PAGE1区
MOVB #0X01                ;切换到存储区1区
CALL 0X7FA
MOV  VREFCAL0, R0
CALL 0X7F9
MOV  VREFCAL1, R0
MOVB #0X00                ;切换回存储器0区
MOVP #0X00                ;切换回PAGE0区
```

15.1 参考电压相关寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
2BH	VREFCTL	VREFSEL1	VREFSEL0	VREFCALEN	VREFCLKEN	VREFOE	P18OE	VREFEN	-
159H	VREFCAL0	内部参考电压校准寄存器 0							
15AH	VREFCAL1	内部参考电压校准寄存器 1							

15.1.1 参考电压寄存器 (VREFCTL)

寄存器7.7: VREFCTL: 参考电压寄存器(地址:2BH)

复位值 0000 0000	bit7						bit0	
	VREFSEL1	VREFSEL0	VREF CALEN	VREF CLKEN	VREFOE	P18OE	VREFEN	-
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

VREFSEL<1:0>:内部参考电压 VREFOUT 选择位

00=保留

01=2V

10=3V

11=4V

VREFCALEN: 参考电压校验使能位

1 = 使能校验

0 = 关闭校验

VREFCLKEN: 参考电压校准时钟使能位

1 = 使能参考电压校准时钟

0 = 关闭参考电压校准时钟

VREFOE: 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

P18OE: 保留位

VREFEN: 参考电压使能位

1 = 使能参考电压

0 = 关闭参考电压

注: 正常使用 FVR 模块时, 请将 VREFCALEN 位和 VREFCLKEN 位置 1 以使能 FVR 的动态校准功能, 以消除环境变化对输出电压偏移的影响。

16 高/低压监测 HLVD

KF8L15Z20XX 系列单片机包含一个高/低压监测（HLVD）模块。

HLVD 模块可以设置电压跳变点和变化方向，当电压按照指定的方向与跳变点发生偏离，将会使高低压监测中断标志位置 1，如果使能了该中断，程序将进入中断行。

16.1 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
15BH	HLVDCTL	HLVDEN	VDIR	-	-	-	VDT2	VDT1	VDT0

16.1.1 高低压监测控制寄存器 HLVDCTL

寄存器18.1: HLVDCTL: 高低压检测控制寄存器(地址: 15BH)

复位值 0000 0000	bit7				bit0			
	HLVDEN	VDIR	-	-	-	VDT2	VDT1	VDT0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

HLVDEN: 高低压监测使能位

0 = 高低压监测关闭

1 = 高低压监测打开

VDIR: 电压监测方向选择位

0 = 当电压等于或者高于跳变点时，事件发生

1 = 当电压等于或者低于跳变点时，事件发生

VDT<2:0>: 电压监测跳变点选择位

111 = 保留

110 = 4.5V

101 = 4.1V

100 = 3.7V

011 = 3.3V

010 = 2.9V

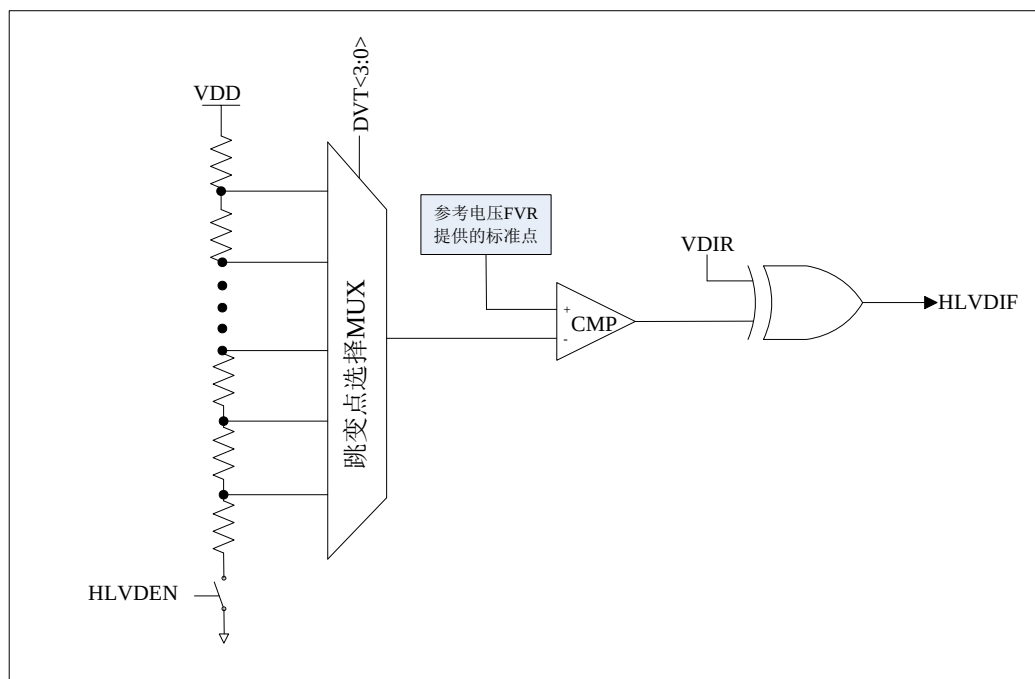
001 = 2.5V

000 = 2.1V

16.2 工作原理

HLVD 模块由内部参考电压 FVR 提供标准点，HLVD 模块不会影响 FVR 模块的工作。用户根据需求通过设置 VDIR 位来选择电压检测的方向，设置 VDT<3:0>位来选择跳变点，跳变点由电阻分压产生。

跳变点跟 FVR 提供的标准点经过比较器比较以判断电压是否在检测范围之内，如果超出则 HLVDIF 置 1。



图：高低压检测原理图

16.3 HLVD 的使用

HLVD 模块通过如下步骤使用：

1. 设置 HLVDCTL 寄存器的 VDT<2:0>位选择 HLVD 跳变点；
2. 设置 VDIR 选择检测的方向；
3. 将 HLVDEN 置 1 以使能 HLVD 模块；
4. 延时 10us，清零 HLVDIF 中断标志位；
5. 如需中断，将 HLVDIE 置 1。

注：

- 1.使能 HLVDEN 后，请延时 10us 后对 HLVDIF 进行清零操作；
- 2.如果需要更改检测条件，则需要将 HLVD 模块关闭后再进行修改，否则可能由于修改设置位引起 HLVDIF 置 1；
- 3.当 VDD 超出检测范围时，HLVDIF 将保持 1 无法清零；直到 VDD 回到检测范围或者禁止 HLVD 模块，HLVDIF 才可被清零操作；
- 4.HLVD 模块的标准点由 FVR 模块提供，无论 FVR 模块是否被使能；FVR 模块的使用不受 HLVD 模块的任何影响。

16.4 HLVD 工作在休眠模式

使能 HLVD 模块后，MCU 进入休眠模式后 HLVD 模块继续工作。当电压越过跳变点时，标志位 HLVDIF 置 1，MCU 从休眠中唤醒；如果 HLVDIE 为 1，则程序进入中断行。

17 硬件乘法器模块

17.1 概述

KF8L15Z20XX 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，运算时间=8 个指令周期

工作原理框图如图 19.1 所示：

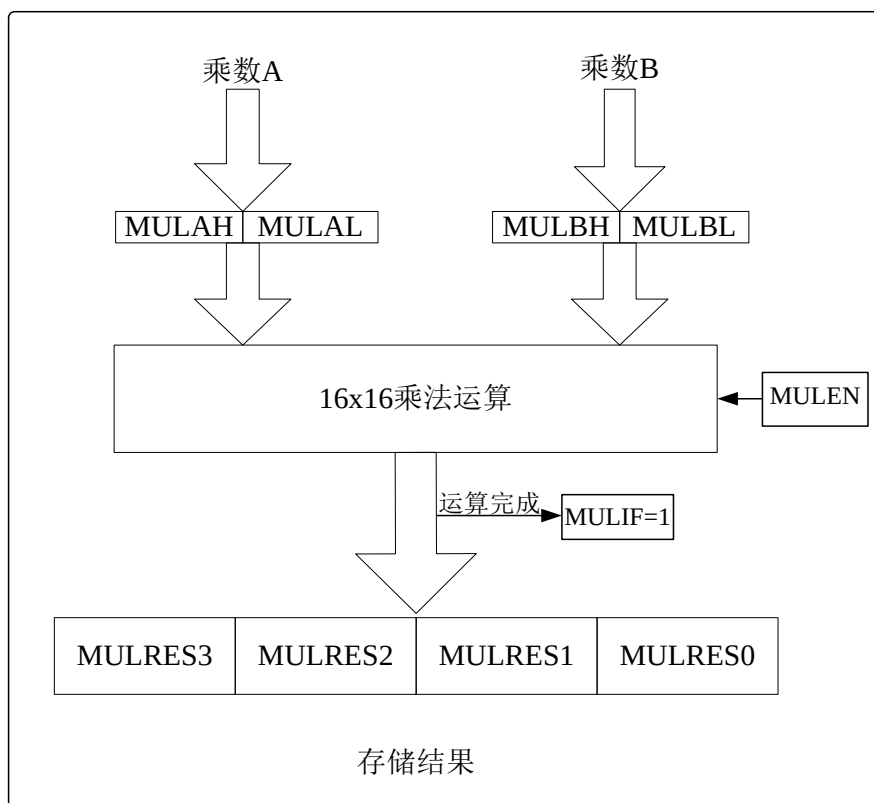


图 19.1 16×16 硬件乘法器原理框图

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。

17.2 硬件乘法器相关寄存器

表 19-1 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	MULRES ₃	乘法运算结果寄存器 3							
136H	MULRES ₂	乘法运算结果寄存器 2							
137H	MULRES ₁	乘法运算结果寄存器 1							
138H	MULRES ₀	乘法运算结果寄存器 0							

17.2.1 乘法运算控制寄存器（MULCTL）

寄存器20.1: MULCTL:乘法运算控制寄存器（地址:134H）

	bit7						bit0	
复位值 ---- --00	-	-	-	-	-	-	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

注：MULIF 不能直接由软件清零。需要清零 MULIF 时，对乘数寄存器（MULAH/L 或 MULBH/L）进行操作即可清零。如：CLR MULAH。

17.3 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

示例程序如例 20.1 所示:

例20.1:

```
MOVB #0X01
MOV R0, #0XxxH
MOV MULAH , R0
MOV R0, #0XxxH
MOV MULAL , R0           ;装载乘数A
MOV R0, #0XxxH
MOV MULBH , R0
MOV R0, #0XxxH
MOV MULBL , R0           ;装载乘数B
SET MULCTL ,MULEN        ;使能乘法运算
```

运算结束后, MUCTL 寄存器的 MULIF 置位, 控制结果存入寄存器 MULRESx (x=0~3) 中, 结果为:[MULRES3:MULRES2:MULRES1:MULRES0]。

18 硬件除法器模块

18.1 概述

KF8L15Z20XX 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÷16 位
- ◆ 运算时间=9 个系统时钟周期

具体的原理框图如图 20.1 所示：

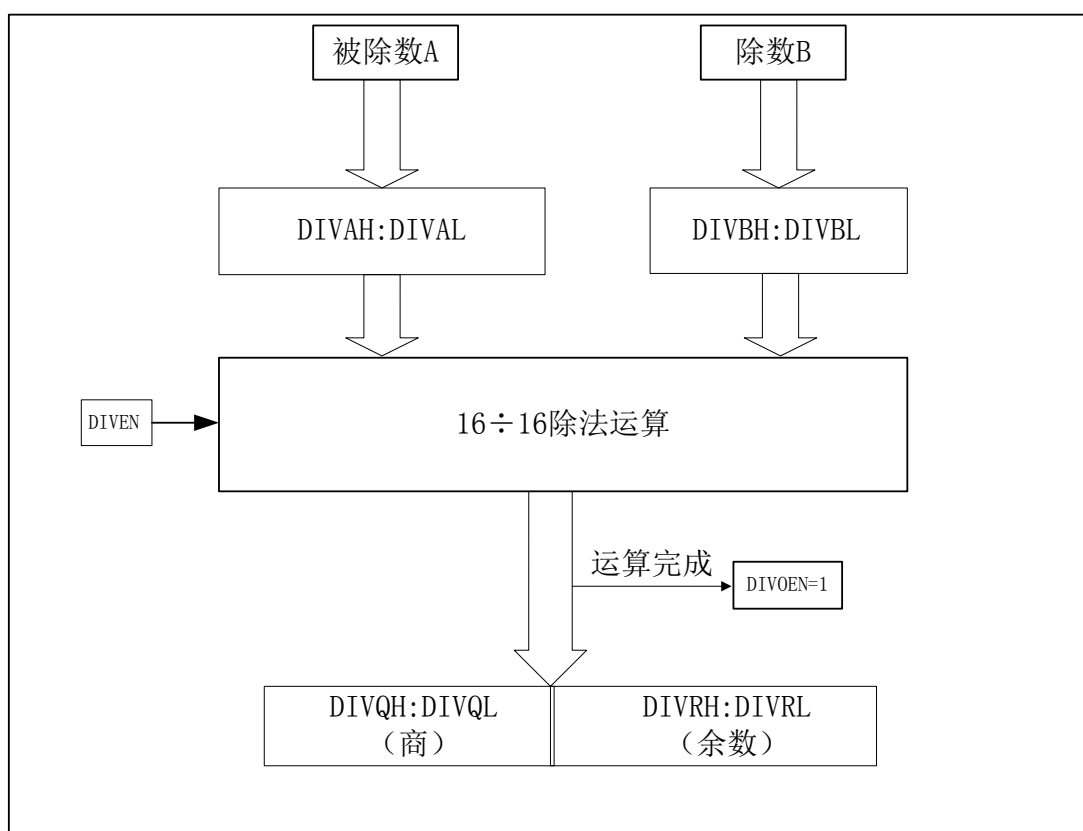


图 20.1 硬件除法器原理框图

18.2 硬件除法器相关寄存器

表 20-1 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							
115H	DIVBH	除数高 8 位寄存器							
13CH	DIVBL	除数低 8 位寄存器							
13DH	DIVQH	商高 8 位寄存器							
13EH	DIVQL	商低 8 位寄存器							
13FH	DIVRL	余数低 8 位寄存器							
108H	DIVRH	余数高 8 位寄存器							

18.2.1 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。

除法控制寄存器 DIVCTL 包含：

- ◆ 除法运算使能位 DIVEN
- ◆ 除法运算完成标志位 DIVOEN

寄存器20.1:DIVCTL: 除法控制寄存器(地址:139H)

	bit7						bit0	
复位值 -----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

18.2.2 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}，{DIVBH:DIVBL}，{DIVQH:DIVQL}，{DIVRH:DIVRL}进行表示。

18.2.2.1 被除数寄存器{DIVAH:DIVAL}

寄存器20.2: DIVAH: 被除数高8位寄存器(地址:13AH)

复位值 00000000	bit7							bit0
	DIVAH7	DIVAH6	DIVAH5	DIVAH4	DIVAH3	DIVAH2	DIVAH1	DIVAH0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

寄存器20.3: DIVAL: 被除数低8位寄存器(地址:13BH)

复位值 00000000	bit7							bit0
	DIVAL7	DIVAL6	DIVAL5	DIVAL4	DIVAL3	DIVAL2	DIVAL1	DIVAL0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DIVAH: 被除数高8位寄存器, 用于存放被除数高8位。

DIVAL: 被除数低8位寄存器, 用于存放被除数低8位。

18.2.2.2 除数寄存器 DIVB

寄存器20.4: DIVBH: 除数高8位寄存器(地址:115H)

复位值 00000000	bit7							bit0
	DIVBH7	DIVBH6	DIVBH5	DIVBH4	DIVBH3	DIVBH2	DIVBH1	DIVBH0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

寄存器20.5: DIVBL: 除数低8位寄存器(地址:13CH)

复位值 00000000	bit7							bit0
	DIVBL7	DIVBL6	DIVBL5	DIVBL4	DIVBL3	DIVBL2	DIVBL1	DIVBL0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DIVBH: 用于存放运算过程中的除数高8位

DIVBL: 用于存放运算过程中的除数低8位

18.2.2.3 结果商寄存器{DIVQH:DIVQL}

寄存器20.6: DIVQH: 结果商高8位寄存器(地址:13DH)

00000000	bit7							bit0
	DIVQH7	DIVQH6	DIVQH5	DIVQH4	DIVQH3	DIVQH2	DIVQH1	DIVQH0
	R	R	R	R	R	R	R	R

寄存器20.7: DIVQL: 结果商低8位寄存器(地址:13EH)

00000000	bit7							bit0
	DIVQL7	DIVQL6	DIVQL5	DIVQL4	DIVQL3	DIVQL2	DIVQL1	DIVQL0
	R	R	R	R	R	R	R	R

DIVQH: 存放运算结果商的高8位; (只读)

DIVQL: 存放运算结果商的低8位; (只读)

18.2.2.4 运算余数寄存器 DIVR

寄存器20.8: DIVRH: 余数高8位寄存器(地址:108H)

复位值 00000000	bit7							bit0
	DIVRH7	DIVRH6	DIVRH5	DIVRH4	DIVRH3	DIVRH2	DIVRH1	DIVRH0
	R	R	R	R	R	R	R	R

寄存器20.9: DIVRL: 余数低8位寄存器(地址:13FH)

复位值 00000000	bit7							bit0
	DIVRL7	DIVRL6	DIVRL5	DIVRL4	DIVRL3	DIVRL2	DIVRL1	DIVRL0
	R	R	R	R	R	R	R	R

DIVRH: 除法完成后存储余数高 8 位（只读）

DIVRL: 除法完成后存储余数低 8 位（只读）

18.3 硬件除法器操作

除法通过使能信号 DIVEN 进行使能，当 DIVEN=1 的时候，除法器开始工作，直到除法完成。

当除法完成之后，除法器模块会自动清除 DIVEN 信号，并将完成标志位 DIVOEN 设为 1。

注意，在除法器使能之前需要将数据输入到 DIVA 和 DIVB 寄存器中，否则的话，除法过程就会出现数据错误。

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） 分别向除数寄存器{DIVBH:DIVBL}写入除数
- （3） 设置 DIVCTL 的 DIVEN (<DIVCTL.0>) 使能除法运算器

示例程序如例 21.1 所示：

```

例21.1:
MOV R0, #0xXX
MOV DIVAH, R0
MOV R0, #0xYY
MOV DIVAL, R0
MOV R0, #0xMM
MOV DIVBH, R0
MOV R0, #0xNN
MOV DIVBL, R0
MOV R0, #0x01
MOV DIVCTL, R0
    
```

这样就可以开始计算 $xyyy \div mmnn$ ，当 DIVOEN=1 的时候，可以从{DIVQH:DIVQL}和{DIVRH:DIVRL}中分别读出商和余数。

19 实时时钟（RTC）模块

19.1 概述

实时时钟 (Real Time Counting, RTC) 单元提供给用户实时时间以及日历信息。RTC 的时钟源 RTCOSC 振荡器外接 32.768 kHz 晶振提供。RTC 单元通过时间寄存器提供时间信息 (秒、分、时、星期、日、月、年)。数据信息由 BCD 码格式进行表示。修改计数器的值可以重新设置系统当前的时间和日期。

该单元也提供闹钟功能，用户可预先在时间闹钟寄存器中设置闹钟日期进行闹铃设置。

RTC 特性包含：

- ◆ 提供实时时钟的日历功能，自动解决闰年问题，计时范围 100 年
- ◆ 闰年、平年显示
- ◆ 12/24 小时模式选择功能
- ◆ 高精度的数字时钟校正功能
- ◆ 支持闹钟中断功能
- ◆ 支持时间节拍中断功能，提供 8 个周期选项
- ◆ 支持时分秒进程中中断功能
- ◆ 内置 2 个 8 位定时器 TMR0 和 TMR1

19.1.1 原理框图

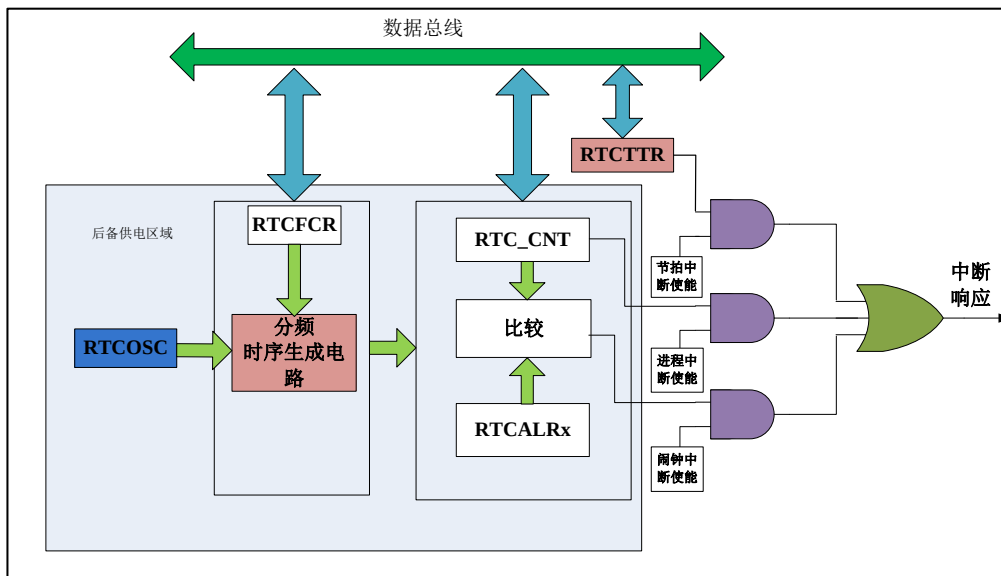


图 22.1 RTC 原理框图

从图 19.1 可以看出，RTC 模块由外部提供时钟，可以通过设置时钟校正 RTCFCR 寄存器进行时钟校正，如果不需要校正，可以对校正值写 00H。RTC 提供实时时间和日历进程。

详见19.3 节功能描述。

RTC 还支持各种中断功能，通过对写入响应的实时时钟寄存器选择响应的进程中中断，也可以与 RTC 闹钟中断寄存器进行比较，当两者值相等时便会产生相应的闹钟中断，使 ALRIF 置 1（用软件清零）。还具有周期时间节拍中断功能，通过设置相应的周期时间节拍进行相应的时间节拍中断。详见19.3.4 中断。

19.2 寄存器描述

19.2.1 相关寄存器

表 19-1 RTC 相关寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
329H	EIE5	ALRIE	RTCT1IE	RTCT0IE	TTIE	DAYIE	HURIE	MINIE	SECIE
32AH	EIF5	ALRIF	RTCT1IF	RTCT0IF	TTIF	DAYIF	HURIF	MINIF	SECIF
32BH	IP5	PALR	PRTCT1	PRTCT0	PTT	PDAY	PHUR	PMIN	PSEC
318H	RTCSRT	SRT7	SRT6	SRT5	SRT4	SRT3	SRT2	SRT1	SRT0
319H	RTCSTU	RTCALREN	-	RESET	LIF	HT	CNF	RTOFF	RTCLD
31CH	RTCALRS	ALRSEN	ALRS6	ALRS5	ALRS4	ALRS3	ALRS2	ALRS1	ALRS0
31DH	RTCALRM	ALRMEN	ALRM6	ALRM5	ALRM4	ALRM3	ALRM2	ALRM1	ALRM0
31EH	RTCALRH	ALRHEN	-	ALRH5	ALRH4	ALRH3	ALRH2	ALRH1	ALRH0
31FH	RTCALRW	ALRWEN	-	-	-	-	ALRW2	ALRW1	ALRW0
320H	RTCTTR	RTCOE	RTCOS	-	-	-	RTCTTR2	RTCTTR1	RTCTTR0
321H	RTCFCR	V7	V6	V5	V4	V3	V2	V1	V0
322H	RTCSEC	-	SEC6	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0
323H	RTCMIN	-	MIN6	MIN5	MIN4	MIN3	MIN2	MIN1	MIN0
324H	RTCHOUR	-	AMPM	HUR5	HUR4	HUR3	HUR2	HUR1	HUR0
325H	RTCWEK	-	-	-	-	-	WEK2	WEK1	WEK0
326H	RTCDAY	-	-	DAY5	DAY4	DAY3	DAY2	DAY1	DAY0
327H	RTCMTH	-	-	-	MTH4	MTH3	MTH2	MTH1	MTH0
328H	RTCYEAR	YER7	YER6	YER5	YER4	YER3	YER2	YER1	YER0
32CH	RTCTMRC TL	CKST13	CKST12	CKST11	CKST10	CKST03	CKST02	CKST01	CKST00
32DH	RTCTMREN	-	-	-	-	-	-	TMR1EN	TMR0EN
32EH	RTCTMR1	RTC 定时器 1 初值配置寄存器							
32FH	RTCTMR0	RTC 定时器 0 初值配置寄存器							

19.2.2 实时时钟启动寄存器 RTCSRT

寄存器: RTCSRT: 实时时钟启动寄存器(地址:318H)

复位值 0000 0000	bit7							bit0
	SRT7	SRT6	SRT5	SRT4	SRT3	SRT2	SRT1	SRT0
	W-R0	W-R0	W-R0	W-R0	W-R0	W-R0	W-R0	W-R0

SRT<7:0>: 实时时钟启动位
写固定值 55H 以使能 RTC 模块;
写任何其他值将禁止 RTC 模块;
SRT<7:0>读出为全 0。

19.2.3 状态显示寄存器 RTCSTU

寄存器19.3: RTCSTU: 状态显示寄存器(地址:319H)

复位值 0-00 0000	bit7						bit0	
	RTCALREN	-	RESET	LIF	HT	CNF	RTOFF	RTCLD
	R/W	U	R/W	R/W	R/W	R/W	R	R

RTCALREN: RTC 闹钟功能总使能位

1 = 使能 RTC 闹钟功能

0 = 禁止 RTC 闹钟功能

RESET: RTC 模块复位位

1 = RTC 模块处于复位状态

0 = RTC 模块退出复位状态

LIF: 闰年指示标志

1 = 当前年份为闰年

0 = 当前年份为平年

HT: 选择小时显示的类型

1 = 小时显示为 12 小时制

0 = 小时显示为 24 小时制

CNF: 配置标志位

此位必须由软件置'1'以进入配置模式，从而允许向实时时钟寄存器、闹钟中断时间寄存器写入数据。只有当此位在被置 1 并重新由软件清 0 后，才会执行写操作。

1 = 进入配置模式。

0 = 退出配置模式(开始更新 RTC 寄存器);

RTOFF: RTC操作关闭

RTC 模块利用这位来指示对其寄存器进行的最后一次操作的状态，指示操作是否完成。若此位为'0'，则表示无法对任何的 RTC 寄存器进行写操作。此位为只读位。

1 = 上一次对 RTC 寄存器的写操作已经完成。

0 = 上一次对 RTC 寄存器的写操作仍在进行;

RTCLD: RTC活动状态 (只读)

1 = RTC 处在正常有效状态

0 = RTC 处在复位状态

19.2.4 闹钟中断时间寄存器

闹钟中断时间寄存器分别对4个寄存器进行设置完成对闹钟时间秒，分，时，星期的设定。寄存器的构成与实时数据寄存器的星期、时、分、秒数据寄存器相同，利用BCD码来表现。请不要设定为不存在日期。

另外，设定的闹钟时刻数据需要与状态显示寄存器的HT位中所设定的12小时制或者24小时制显示相匹配。

19.2.4.1 闹钟中断秒寄存器 RTCALRS

寄存器19.4: RTCALRS: 闹钟中断秒寄存器(地址:31CH)

复位值 0000 0000	bit7						bit0	
	ALRSEN	ALRS6	ALRS5	ALRS4	ALRS3	ALRS2	ALRS1	ALRS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ALRSEN: 闹钟秒时间使能位

0 = 禁止秒时间 ALRS<6:0>参与闹钟

1 = 使能秒时间 ALRS<6:0>参与闹钟

ALRS<6:0>: 设置闹钟中断秒时间(闹钟秒的数值为 00~59)，采用 BCD 编码

19.2.4.2 闹钟中断分寄存器 RTCALRM

寄存器19.5: RTCALRM: 闹钟中断分寄存器(地址:31DH)

复位值 0000 0000	bit7						bit0	
	ALRMEN	ALRM6	ALRM5	ALRM4	ALRM3	ALRM2	ALRM1	ALRM0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ALRMEN: 闹钟分时间使能位

0 = 禁止分时间 ALRM<6:0>参与闹钟

1 = 使能分时间 ALRM<6:0>参与闹钟

ALRM<6:0>: 设置闹钟中断分时间(闹钟分的数值为 00~59)，采用 BCD 编码

19.2.4.3 闹钟中断时寄存器 RTCALRH

寄存器19.6: RTCALRH: 闹钟中断时寄存器(地址:31EH)

复位值 0-00 0000	bit7						bit0	
	ALRHEN	-	ALRH5	ALRH4	ALRH3	ALRH2	ALRH1	ALRH0
	R/W	U	R/W	R/W	R/W	R/W	R/W	R/W

ALRHEN: 闹钟小时时间使能位

0 = 禁止小时时间 ALRH<5:0>参与闹钟

1 = 使能小时时间 ALRH<5:0>参与闹钟

ALRH<5:0>: 设置闹钟中断小时时间（闹钟时的数值为 00~23），采用 BCD 编码

19.2.4.4 闹钟中断星期寄存器 RTCALRW

寄存器19.7: RTCALRW: 闹钟中断星期寄存器(地址:31FH)

复位值 0--- -000	bit7					bit0		
	ALRWEN	-	-	-	-	ALRW2	ALRW1	ALRW0
	R/W	U	U	U	U	R/W	R/W	R/W

ALRWEN: 闹钟星期时间使能位

0 = 禁止星期时间 ALRW<2:0>参与闹钟

1 = 使能星期时间 ALRW<2:0>参与闹钟

ALRW<2:0>: 设置闹钟中断星期时间（闹钟星期数值为 1~7），采用 BCD 编码

19.2.5 时间节拍中断寄存器 RTCTTR

周期时间节拍中断有 8 个选项: 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 以及 1 秒, 通过设置 RTCTTR寄存器来进行选择。

寄存器19.8: RTCTTR: 时间节拍中断寄存器(地址:320H)

复位值 00-- -000	bit7					bit0		
	RTCOE	RTCOS	-	-	-	RTCTTR2	RTCTTR1	RTCTTR0
	R/W	R/W	U	U	U	R/W	R/W	R/W

RTCOE: RTC输出使能位

0 = 禁止RTC输出到P0.6口

1 = 使能RTC输出到P0.6口

RTCOS: RTC输出选择位

0 = 选择时间节拍输出到P0.6口

1 = 选择闹钟信号输出到P0.6口

RTCTTR<2:0>:时间节拍设置位

000=时间节拍为1秒

001=时间节拍为1/2秒

010=时间节拍为1/4秒

011=时间节拍为1/8秒

100=时间节拍为1/16秒

101=时间节拍为1/32秒

110=时间节拍为1/64秒

111=时间节拍为1/128秒

19.2.6 实时时钟校正寄存器 RTCFCR

时钟校正寄存器为1字节的寄存器, 是用来校正时钟偏差 (提前或滞后) 的寄存器。在不使用时钟校正功能时, 请设定为"00 h"。

寄存器19.9: RTCFCR: 时间校正寄存器(地址:321H)

复位值 0000 0000	bit7						bit0	
	V7	V6	V5	V4	V3	V2	V1	V0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

V<7:0>: 将校正值转化为 2 进制值置入 B7-B0 中。

19.2.7 实时时钟寄存器

实时时钟寄存器主要用于读取或设置 RTC 计数器，具体如下面章节所述。

19.2.7.1 实时时钟秒寄存器 RTCSEC

用户可通过读此寄存器来获得 RTC 计数器当前值的秒时间。秒数值为 00~59，采用 BCD 编码。

寄存器19.10: RTCSEC: 实时时钟秒寄存器(地址:322H)

复位值 - 000 0000	bit7						bit0	
	-	SEC6	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0
	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

注：要对此寄存器进行写操作，必须先进入配置模式(CNF = 1)。

19.2.7.2 实时时钟分寄存器 RTCMIN

用户可通过读此寄存器来获得 RTC 计数器当前值的分时间。分钟数值为 00~59，采用 BCD 编码。

寄存器19.11: RTCMIN: 实时时钟分寄存器(地址:323H)

复位值 - 000 0000	bit7						bit0	
	-	MIN6	MIN5	MIN4	MIN3	MIN2	MIN1	MIN0
	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

注：要对此寄存器进行写操作，必须先进入配置模式(CNF = 1)。

19.2.7.3 实时时钟时寄存器 RTCHOUR

寄存器19.12: RTCHOUR: 实时时钟时寄存器(地址:324H)

复位值 - 000 0000	bit7						bit0	
	-	AMPM	HUR5	HUR4	HUR3	HUR2	HUR1	HUR0
	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

AMPM: 当 HT 置 1 选择 24 小时制时，该位无效，0,1 均可。

当 HT 置 0 选择 12 小时制时：

1 = PM

0 = AM

HUR<20:1>: 可通过读此寄存器来获得RTC计数器当前值的时时间。小时数值为00~23或00~12，采用BCD编码

注：要对此寄存器进行写操作，必须先进入配置模式(CNF = 1)。

19.2.7.4 实时时钟星期寄存器 RTCWEK

用户可通过读此寄存器来获得RTC计数器当前值的星期时间。星期数值为1~7，采用BCD编码。

寄存器19.13: RTCWEK: 实时时钟星期寄存器(地址:325H)

	bit7					bit0		
复位值 ----0000	-	-	-	-	-	WEK2	WEK1	WEK0
	U	U	U	U	U	R/W	R/W	R/W

注：要对此寄存器进行写操作，必须先进入配置模式(CNF = 1)。

19.2.7.5 实时时钟日寄存器 RTCDAY

用户可通过读此寄存器来获得RTC计数器当前值的日时间。日的数值为01~31，采用BCD编码。

寄存器19.14: RTCDAY: 实时时钟日寄存器(地址:326H)

	bit7					bit0		
复位值 --00 0000	-	-	DAY5	DAY4	DAY3	DAY2	DAY1	DAY0
	U	U	R/W	R/W	R/W	R/W	R/W	R/W

注：要对此寄存器进行写操作，必须先进入配置模式(CNF = 1)。

19.2.7.6 实时时钟月寄存器 RTCMTH

用户可通过读此寄存器来获得RTC计数器当前值的月时间。月的数值为01~12，采用BCD编码。

寄存器19.15: RTCMTH: 实时时钟月寄存器(地址:327H)

	bit7					bit0		
复位值 ---0 0000	-	-	-	MTH4	MTH3	MTH2	MTH1	MTH0
	U	U	U	R/W	R/W	R/W	R/W	R/W

注：要对此寄存器进行写操作，必须先进入配置模式(CNF = 1)。

19.2.7.7 实时时钟年寄存器 RTCYEAR

用户可通过读此寄存器来获得RTC计数器当前值的年时间。年的数值为00~99，采用BCD编码。

寄存器19.16: RTCYEAR: 实时时钟年寄存器(地址:328H)

	bit7							bit0
复位值 0000 0000	YER7	YER6	YER5	YER4	YER3	YER2	YER1	YER0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

注：要对此寄存器进行写操作，必须先进入配置模式(CNF = 1)。

19.2.7.8 实时时钟定时器使能寄存器 RTCTMREN

寄存器21.18: RTCTMREN: 实时时钟定时器使能寄存器(地址: 32DH)

	bit7						bit0	
复位值 ----00	-	-	-	-	-	-	TMR1EN	TMR0EN
	U	U	U	U	U	U	R/W	R/W

TMR1EN: 定时器 1 使能位

1 = 使能定时器 1
0 = 禁止定时器 1
TMR0EN: 定时器 0 使能位
1 = 使能定时器 0
0 = 禁止定时器 0

19.2.7.9 实时时钟定时器控制寄存器 RTCTMRCTL

寄存器21.19: RTCTMRCTL: 实时时钟定时器控制寄存器(地址: 32CH)

复位值 0---0000	bit7						bit0	
	CKST13	CKST12	CKST11	CKST10	CKST03	CKST02	CKST01	CKST00
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CKST1<3:0>: 定时器 1 时钟源选择位
0000 = RTC 时钟源/128 约 1/256s
0001 = RTC 时钟源/512 约 1/64s
0010 = RTC 时钟源/1024 约 1/32s
0011 = RTC 时钟源/2048 约 1/16s
0100 = RTC 时钟源/8192 约 1/4s
0101 = 1s
0110 = 1min
0111 = 1hour
1000 = 计数器 TMR1 和 TMR0 级联

CKST0<3:0>: 定时器 0 时钟源选择位
0000 = RTC 时钟源/128 约 1/256s
0001 = RTC 时钟源/512 约 1/64s
0010 = RTC 时钟源/1024 约 1/32s
0011 = RTC 时钟源/2048 约 1/16s
0100 = RTC 时钟源/8192 约 1/4s
0101 = 1s
0110 = 1min
0111 = 1hour

19.2.7.10 实时时钟定时器 0 计数寄存器 RTCTMR0

寄存器21.20: RTCTMR0: 实时时钟定时器0计数寄存器(地址: 32FH)

复位值 0000 0000	bit7						bit0	
	TMR07	TMR06	TMR05	TMR04	TMR03	TMR02	TMR01	TMR00
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TMR0<7:0>: 定时器 0 初值配置位

19.2.7.11 实时时钟定时器 1 计数寄存器 RTCTMR1

寄存器21.21: RTCTMR1:实时时钟定时器1计数寄存器(地址: 32EH)

复位值 0000 0000	bit7						bit0	
	TMR17	TMR16	TMR15	TMR14	TMR13	TMR12	TMR11	TMR10
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TMR1<7:0>: 定时器 1 初值配置位

19.3 功能描述

19.3.1 RTC 初始化

备份区外设复位寄存器 PHCLR 中的 VRTCS 位为 RTC 模块的电源控制位, 默认为 1, 即 RTC 模块处于有电状态。如用户需要运行在超低功耗模式下, 且不使用 RTC 模块, 可将 VRTCS 位清零, 否则, 请保持该位为 1, 以确保 RTC 模块可以正常工作。

在使能 RTC 模块之前, 用户可通过 RTCSTU 寄存器中的 RESET 位对 RTC 模块进行复位以初始化实时时钟寄存器及内部逻辑。

RTCSRT 寄存器用于 RTC 模块的使能。向 RTCSRT 寄存器写固定值 55H 以使能 RTC 模块, 写任何其他值将禁止 RTC 模块。RTCSRT 寄存器读出为全 0。

19.3.2 RTC 启动

RTC 实时时钟启动步骤:

1. RTC 的初始化; 配置 RTCSTU 寄存器的 RESET 位置 1 以初始化 RTC 模块内部逻辑及实时时钟寄存器, 再将该位清零, 退出复位状态。
2. RTC 的初值配置; 设置 RTC 实时时钟寄存器, 设置年/月/日/星期/时/分/秒等时间寄存器。
3. RTC 的启动; 对 RTCSRT 寄存器写固定值 55H, 使能 RTC 模块。
4. 实时时钟硬件自动载入新值并开始计时。

19.3.3 RTC 时间设置

RTC 实时时钟时间设置方法:

1. 查询RTOFF (RTCSTU<1>) 位, 直到RTOFF (RTCSTU<1>) 的值变为1;
2. 置CNF (RTCSTU<2>) 位为1, 进入配置模式 ;
3. 对一个或多个RTC闹钟中断时间寄存器或者实时时钟寄存器进行写操作 ;
4. 清0 CNF (RTCSTU<2>) 位, 退出配置模式; 此时写操作才能进行。
5. 重新查询RTOFF (RTCSTU<1>) 位, 直至RTOFF (RTCSTU<1>) 位变为1以确认写操作已经完成。

注: 仅当CNF标志位被清0时, 写操作才能进行。

必须置CNF (RTCSTU<2>) 位为1, 使RTC进入配置模式后, 才能对实时时钟寄存器, 闹钟中断时间寄存器等写入操作。

另外, 对RTC任何寄存器的写操作, 都必须在前一次写操作结束后进行。可以通过查询

RTCSTU寄存器中的RTOFF状态位，判断RTC寄存器是否处于更新中。仅当RTOFF状态位是1时，才可以写入RTC时间寄存器。

19.3.4 RTC 闹钟功能

RTC 模块提供星期、小时、分和秒的闹钟功能，用户可根据实际需求，通过配置RTCALRW、RTCALRH、RTCALRM 和 RTCALRS 寄存器进行闹钟时间设定；每个时间进程均提供一位独立的使能位（ALRWEN、ALRHEN、ALRMEN 和 ALRSEN），可满足各种不同的配置需求。

RTCSTU 寄存器的 RTCALREN 位为闹钟功能的总使能位，将该位置 1 后将使能闹钟功能，当实时时钟寄存器中的时间和闹钟中断时间寄存器相同时，将产生一个闹钟中断，ALRIF 位置 1。

闹钟功能原理框图如下图所示。

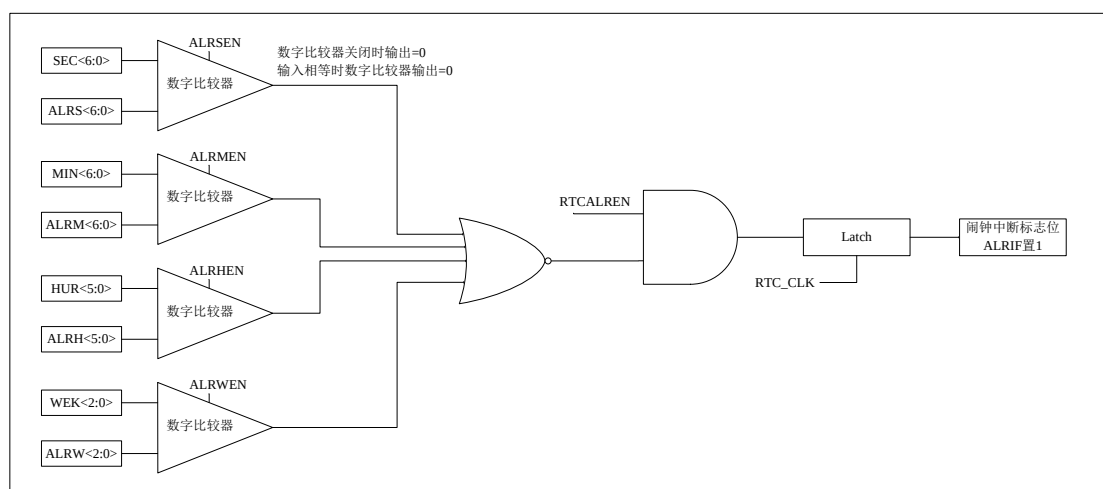


图 22.2 RTC 闹钟功能原理框图

19.3.5 时钟校正功能

时钟校正功能是为了实现高精度的时钟功能，校正因振荡频率的偏差而导致的时钟的提前或滞后的功能。进行校正时，并不调节石英振荡器的频率，而是利用分频电路的一部分对时钟脉冲进行调整。校正工作每20秒（或60秒）进行一次。最小分解能约为在大约3 ppm（或约1 ppm）的条件下，可以在-195.3 ppm ~ +192.2 ppm（或是-65.1 ppm ~ +64.1 ppm）的范围内进行校正。此设定可在时钟校正寄存器上进行。另外，不使用时钟校正功能时，请务必将其设定为 "00 h"。

表19-3 时间校正参考

项目	B7=0	B7=1
校正工作	每20秒	每60秒
最小分解能	3.052ppm	1.017ppm
校正范围	-195.3ppm~+192.2ppm	-65.1ppm~+64.1ppm

19.3.5.1 计算方法

1. 当前振荡频率 > 目标频率时（时间提前时）

$$\text{校正值} = 128 - \text{整数} \left(\frac{(\text{当前振荡频率实测值}) - (\text{目标振荡频率})}{(\text{当前振荡频率实测值}) \times (\text{最小分解能})} \right)$$

注：此计算值的“0~64”范围为可校正范围。

2. 当前振荡频率 < 目标频率时 (时间滞后时)

$$\text{校正值} = \text{整数} \left(\frac{(\text{目标振荡频率}) - (\text{当前振荡频率实测值})}{(\text{当前振荡频率实测值}) \times (\text{最小分解能})} \right) + 1$$

注：此计算值的“0~62”范围为可校正范围。

19.3.5.2 校准原理

根据实际测出的频率，利用分频电路的一部分对时钟脉冲进行增加或者减少。

1. 当前振荡频率 > 目标频率时 (时间提前时)

$$\text{校正值} = 128 - \text{整数} \left(\frac{(\text{当前振荡频率实测值}) - (\text{目标振荡频率})}{(\text{当前振荡频率实测值}) \times (\text{最小分解能})} \right)$$

例：校准寄存器值为 0111 1101 (t=1/32768)

表19-4 校准例图 (校准周期为20s, 延迟时间, 校准大小为3)

	校准值为 0111 1101	无校准
分频计数器计数	01 00 0000 0	01 00 0000 0
	00 11 1101 1 +6t	01 00 0000 1
	00 11 1110 0 +6t	01 00 0001 0
	00 11 1110 1 +6t	01 00 0001 1

在 20s 校准周期中增加了 6 个 t (1/32768), 延迟时间。

20 秒校准 1 次增加的 t = 校准寄存器<B6:B0>补码值*2t

2. 当前振荡频率 < 目标频率时 (时间滞后时)

$$\text{校正值} = \text{整数} \left(\frac{(\text{目标振荡频率}) - (\text{当前振荡频率实测值})}{(\text{当前振荡频率实测值}) \times (\text{最小分解能})} \right) + 1$$

例：校准寄存器值为 0000 0011 (t=1/32768)

表19-5 校准例图 (校准周期为20s, 加速时间, 校准大小为3)

	校准值为 0000 0011	无校准
分频计数器计数	01 00 0000 0	01 00 0000 0
	01 00 0011 1 -6t	01 00 0000 1
	01 00 0100 0 -6t	01 00 0001 0
	01 00 0100 1 -6t	01 00 0001 1

在 20s 校准周期中减去了 6 个 t (1/32768), 加速时间。

20 秒校准 1 次减去的 t = 校准寄存器值<B6:B0>*2t

19.3.5.3 校准寄存器设定值表

表19-6 校准寄存器设定值表1 (最小分解能=3.052ppm)

B7=0, 每20秒校准一次 最小分解能=3.052ppm									
B7	B6	B5	B4	B3	B2	B1	B0	校准值	精度

								(ppm)	(sec/day)
0	0	1	1	1	1	1	1	192.3	16.61
0	0	1	1	1	1	1	0	189.2	16.35
0	0	1	1	1	1	0	1	186.2	16.09
...								...	...
0	0	0	0	0	0	1	0	6.1	0.53
0	0	0	0	0	0	0	1	3.1	0.26
0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	-3.1	-0.26
0	1	1	1	1	1	1	0	-6.1	-0.53
...								...	...
0	1	0	0	0	0	1	1	-186.2	-16.09
0	1	0	0	0	0	1	0	-189.2	-16.35
0	1	0	0	0	0	0	1	-192.3	-16.61
0	1	0	0	0	0	0	0	-195.3	-16.88

表19-7 校准寄存器设定值表2（最小分解能=1.017ppm）

B7=1,每60秒校准一次 最小分解能=1.017ppm									
B7	B6	B5	B4	B3	B2	B1	B0	校准值 (ppm)	精度 (sec/day)
1	0	1	1	1	1	1	1	64.1	5.54
1	0	1	1	1	1	1	0	63.1	5.45
1	0	1	1	1	1	0	1	62.0	5.36
...								...	...
1	0	0	0	0	0	1	0	2.0	0.18
1	0	0	0	0	0	0	1	1.0	0.09
1	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	-1.0	-0.09
1	1	1	1	1	1	1	0	-2.0	-0.18
...								...	...
1	1	0	0	0	0	1	1	-62.0	-5.36
1	1	0	0	0	0	1	0	-63.1	-5.45
1	1	0	0	0	0	0	1	-64.1	-5.54
1	1	0	0	0	0	0	0	-65.1	-5.62

19.3.6 12/24 小时模式选择

通过 RTCSTU 寄存器的 HT 位可对 12/24 小时模式进行选择。

HT: 选择小时显示的类型 (hour type)

- 0: 小时显示为 24 小时制
- 1: 小时显示为 12 小时制

19.3.7 闰年显示功能

RTCSTU 寄存器的 LIF 位可显示当前年份是否为闰年。

LIF: 闰年指示标志 (Leap indication flag)

- 0: 当前年份为平年
- 1: 当前年份为闰年

19.3.8 RTC 定时器

实时时钟模块内置 2 个可级联的 8 位定时器 TMR0 和 TMR1。置位 RTCTMREN 寄存器的 TMRxEN 位来使能定时器 TMRx；通过设置 RTCTMRCTL 寄存器的 CKSTx<3:0>来选择 RTC 定时器 x 的时钟源；寄存器 RTCTMRx 保存定时器的初值。

TMR0 和 TMR1 为向下计数，当定时器发生下溢时（计数值=00H），RTCTxIF 置 1，如果使能 RTCTxIE 位，程序将进入中断行。

TMR0 和 TMR1 带有重载功能，即定时器计数到 00H 后，将会重新载入 RTCTMRx 寄存器的值；如果在计数过程中对 RTCTMRx 寄存器进行修改，不会影响当前计数周期，定时器会在下溢发生后，载入 RTCTMRx 寄存器的新值并开始一个新的计数周期。

将 RTCTMRCTL 寄存器的 CKST1<3:0>位配置为 1000H 时，将使能 TMR0 和 TMR1 的级联，TMR0 为低 8 位，TMR1 为高 8 位，组成一个 16 位定时器；通过 CKST0<3:0>位选择计数时钟；TMR0EN 位和 TMR1EN 位需要同时置 1；两个中断标志位 RTCT0IF 和 RTCT1IF 均有效，用户可任选其一。

TMR0 和 TMR1 定时器的实时计数寄存器不可读。

定时器的操作如下：

1. 通过设置 RTCTMRCTL 寄存器的 CKSTx<3:0>选择定时器时钟源及是否级联；
2. 向 RTCTMRx 寄存器写入定时器初值；
3. 置位 RTCTMREN 寄存器的 TMRxEN 位使能定时器；
4. 通过设置相应的定时器中断使能位来使能/禁止定时器中断。

19.3.9 RTC 工作在休眠模式和超低功耗模式

使能 RTC 模块后，单片机进入休眠模式或者超低功耗模式，RTC 模块将继续正常运行。任何 RTC 中断（时间节拍中断、闹钟中断、进程中断和内置定时器中断）在使能后都可将单片机从休眠模式下唤醒。

RTC 工作在超低功耗模式时，除闹钟中断外，其他中断无法将单片机从关断模式下唤醒；在使能 RTC 模块及闹钟功能后进入超低功耗模式，闹钟中断信号可将单片机从超低功耗模式唤醒。在超低功耗模式下，不需要使能闹钟中断使能位，使能闹钟功能后，闹钟触发信号将自动唤醒单片机。

19.3.10 RTC 中断

RTC 模块提供如下丰富的中断源供用户使用。

- 闹钟中断
- 节拍中断
- 进程中断：秒中断、分中断、时中断和日中断
- 内置定时器中断：定时器0、定时器1

注: RTC中断信号持续时间约为16us，在该时间内，RTC中断标志位将无法清零。用户在使用 RTC中断时需注意中断清零指令是否已在该时间外；或者在执行清零中断标志位指令操作之后，添加判断是否已将RTC中断标志位清零，以避免重复触发中断条件。

19.3.10.1 闹钟中断时间设置

闹钟中断时间设置方法：

1. 清 0 闹钟中断使能位 ALRIE (<EIE5.7>);
2. 设置闹钟中断时间寄存器 (星期/时/分/秒);
3. 将 RTCSTU 寄存器的 RTCALREN 位置 1, 使能 RTC 闹钟功能;
4. 清零 ALRIF 位 (<EIF5.7>);
5. 置 1 闹钟中断使能位 ALRIE (<EIE5.7>);
6. 达到闹钟时间, 产生闹钟中断并进入中断行;
7. 可以通过置 1 IPEN(PCTL<3>)位, 如果置 1 闹钟中断优先级控制位 PALR, 则闹钟中断为高优先级, 此时使 AIEH 置 1 才响应高优先级中断。如果清 0 闹钟中断优先级控制位 PALR, 则闹钟中断为低优先级, 此时使 AIEH 置 1 且 AIEL 置 1 才响应低优先级中断。

将闹钟中断时间按星期, 时, 分, 秒分别写入 RTCALRW、RTCALRH、RTCALRM 和 RTCALRS 寄存器中, 并将对应的 ALRWEN、ALRHEN、ALRMEN 和 ALRSEN 位置 1, 用户可根据实际需求使能、禁止对应时间的中断; RTCSTU 寄存器的 RTCALREN 位为闹钟功能的总使能位, 在配置好闹钟时间寄存器后, 将该位置 1 以使能闹钟功能。

当实时时钟寄存器等于闹钟中断寄存器中的设定值, 则闹钟中断标志 ALRIF 被置 1。该位可由软件清零, 上电自动清零。闹钟中断使能位置 1, 可使能得到想要的闹钟中断。

注: 闹钟中断小时寄存器中的 12/24 选择需和实时时钟寄存器设置一致;

19.3.10.2 周期时间节拍中断

周期时间节拍中断有 8 个选项: 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 以及 1 秒, 通过设置 RTCTTR<2:0>来进行选择。当时间节拍中断使能位 TTIE 置 1 时, 可产生周期时间节拍中断。当 IPEN(PCTL<3>)位置 1, 且周期时间节拍中断的优先级控制位 PTT(IP5<4>)置 1, 则周期节拍中断为高优先级, 此时 AIEH 位置 1 才响应高优先级中断; 如果清 0 周期时间节拍中断的优先级控制位 PTT(IP5<4>), 则周期节拍中断为低优先级, 此时 AIEH 位和 AIEL 位须同时置 1 才响应低优先级中断。

表19-2 时间节拍设置

RTCTTR[2:0]	时间节拍(秒)
000	1
001	1/2
010	1/4
011	1/8
100	1/16
101	1/32
110	1/64
111	1/128

19.3.10.3 进程中断

通过 RTCIE 寄存器的 SECIE、MINIE、HURIE 或 DAYIE 4 个使能端分别控制秒、分、时或天的进程中断。当实时时间达到设置进程中断时间且进程使能端置 1，则产生相应的进程时间中断。如果使能 IPEN(PCTL<3>)位置 1，如果置 1 优先级控制位，则进程中断为高优先级，此时使 AIEH 置 1 才响应高优先级中断。如果清 0 优先级控制位，则进程中断为低优先级，此时使 AIEH 置 1 且 AIEL 置 1 才响应低优先级中断。

19.3.10.4 内置定时器中断

RTC 内置的两个定时器均配备中断，中断标志位为 EIF5 寄存器的 RTCT1IF 及 RTCT0IF。当 RTC 内置定时器溢出时，相应的中断标志位置 1。如果外设中断使能且 EIE5 寄存器的 RTCT1IE/RTCT0IE 置 1，则当 RTC 内置定时器溢出时，程序将进入中断行；如果 MCU 在休眠状态下，中断将唤醒 MCU。内置定时器详细描述见[RTC 定时器章节](#)。

20 复位

KF8L15Z20XX 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任何复位条件下都不会受到影响,上电复位时它们的状态不定,而在其它复位发生时其状态将保持不变。其它大多数寄存器在复位事件发生时将被复位成“复位状态”。图 21.1 给出了片内复位电路的简化结构方框图。

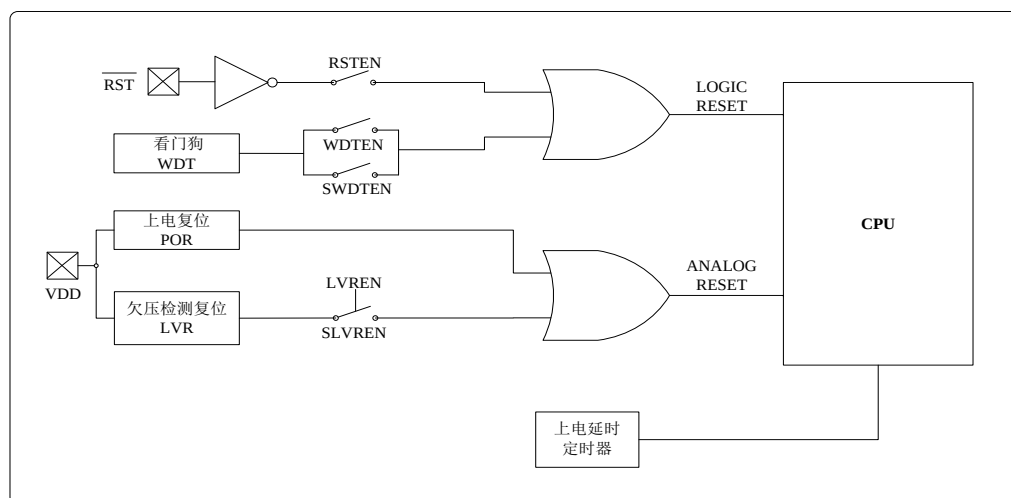


图 20.1 片内复位电路简化框图

注:上电延时定时器只对上电复位(POR)和欠压复位(LVR)有效。

20.1 电源控制状态寄存器(PCTL)

如寄存器 PCTL 所示， $\overline{\text{LVR}}$ 位的状态在单片机上电复位时是不确定的。如果用户在使用中要用到该位，在程序初始化部分需将其置 1，随后如果有复位发生且 $\overline{\text{LVR}}=0$ ，则表示发生过欠压检测复位。 $\overline{\text{LVR}}$ 状态位是“无关”位，如果欠压检测电路被关闭(通过设定配置字中的 LVREN 位和 PCTL 中的 SLVREN 位)， $\overline{\text{LVR}}$ 状态位是不可预知的。 $\overline{\text{POR}}$ 是上电复位状态位，该位在上电复位时被清 0，在其它情况下不受影响。

寄存器： PCTL： 电源控制寄存器(地址:2EH)

复位值 --01 00xx	bit7			SLVREN	IPEN	SWDTEN	$\overline{\text{POR}}$	$\overline{\text{LVR}}$
	-	-	-					
	U	U	R/W	R/W	R/W	R/W	R/W	R/W

- SLVREN:** 软件欠压检测使能位
 1 = 软件使能欠压检测
 0 = 软件禁止欠压检测
- 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
- $\overline{\text{LVR}}$:** 欠压复位状态位
 1 = 未发生欠压复位
 0 = 已发生欠压复位

注：SLVREN 位的配置方法详见 23.5 欠压检测复位（LVR）。

20.2 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。KF8L15Z20XX 的上电复位时间为 70ms 左右。

20.3 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。

20.4 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，KF8L15Z20XX 器件有一个噪声滤波器用于检测和滤除小脉冲，下图是建议 $\overline{\text{RST}}$ 复位电路。

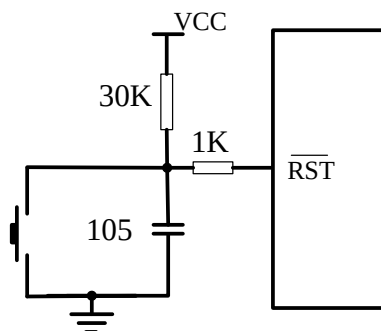


图 20.2 建议 $\overline{\text{RST}}$ 复位电路

20.5 欠压检测复位(LVR)

KF8L15Z20XX 系列中的单片机片内配备一个欠压检测复位电路 (LVR)。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，单片机会按照配置位中的 LVREN 位的配置，选择 LVR 是否使能。LVR 还提供一个软件位 PCTL_SLVREN，当 FUSE_LVREN=0 时，PCTL_SLVREN 位失效；当 FUSE_LVREN=1 时，PCTL_SLVREN 位有效，可以通过软件清零 SLVREN 位关闭 LVR，同样可以通过置位该位以重新使能 LVR。上电时，如果 FUSE_LVREN=1，LVR 将默认工作。

表 20-1 欠压检测复位 LVR 的配置

配置位 LVREN	PCTL_SLVREN	LVR 是否被使能
0	0/1	禁止
1	清零	禁止
1	置 1	使能

如果 VDD 跌落至 VLVR(VLVR=2.1V)以下且持续时间大于 TLVR (TLVR 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 VLVR 以上，再经过 2ms 内置延时后单片机开始正常工作；如果使能上电延时定时器，则在随后 70ms 左右的延时时间处于延时复位状态，过了 70ms 以后单片机开始正常工作。

如果 VDD 跌落至 VLVR 以下的时间小于规定参数(TLVR)，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 VLVR 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 VLVR 以上时，上电延时定时器启动一个 32ms 的复位延时，如在延时器件没有欠压发生，单片机会退出复位状态开始正常工作。

20.6 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(标称值)的固定延时时间。上电延时定时器的定时时钟为系统内部振荡器。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、内部振荡器频率等的变化，不同单片机的上电延时时间有所差异。

20.7 不同复位条件下对寄存器的影响

表 20-2 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	--0q quuu	--uq quuu
05H	P0	xxxx xxxx	xxxx xxxx	uuuu uuuu
06H	P2	xxxx xxxx	xxxx xxxx	uuuu uuuu
07H	P1	xxxx xxxx	xxxx xxxx	uuuu uuuu
08H	P3	xxxx xxxx	xxxx xxxx	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	0-00 -000	0-00 -000	u-uu -uuu
0EH	T1L	xxxx xxxx	xxxx xxxx	uuuu uuuu
0FH	T1H	xxxx xxxx	xxxx xxxx	uuuu uuuu
10H	T1CTL	0000 0000	0000 0000	uuuu uuuu
11H	T2L	0000 0000	0000 0000	uuuu uuuu
12H	T2CTL0	-000 0000	-000 0000	-uuu uuuu
13H	CCP3CTL	0000 0000	0000 0000	uuuu uuuu
14H	CCP3H	0000 0000	0000 0000	uuuu uuuu
15H	CCP3L	0000 0000	0000 0000	uuuu uuuu
16H	CCTCTL	0000 0000	0000 0000	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	0000 0000	0000 0000	uuuu uuuu
1BH	AMPCTL	1000 0000	1000 0000	uuuu uuuu
1DH	ANS1	1111 1111	1111 1111	uuuu uuuu
1EH	ADCDATA0H	xxxx xxxx	xxxx xxxx	uuuu uuuu
1FH	ADCCTL0	00-- --00	00-- --00	uu-- --uu
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	0-00 -000	0-00 -000	u-uu -uuu
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	-110 --00	-110 --00	-uuu --uu
29H	IP3	0000 0000	0000 0000	uuuu uuuu
2AH	OSCCAL2	1000 0000	1000 0000	uuuu uuuu
2BH	VREFCTL	0000 0000	0000 0000	uuuu uuuu
2CH	EIE1	0000 0000	0000 0000	uuuu uuuu
2DH	EIE2	0000 0000	0000 0000	uuuu uuuu
2EH	PCTL	--01 00xx	--01 00xx	--00 00uu
2FH	OSCCTL	0010 0000	0010 0000	uuuu uuuu
31H	ANS0	1111 1111	1111 1111	uuuu uuuu
32H	ANS2	1111 1111	1111 1111	uuuu uuuu
33H	ANS3	1111 1111	1111 1111	uuuu uuuu
34H	OSC0CAL2	0000 0101	0000 0101	uuuu uuuu
35H	PUR0	1111 1111	1111 1111	uuuu uuuu
36H	IOCL0	0000 0000	0000 0000	uuuu uuuu
37H	OSCCAL1	---- -000	---- -000	---- -uuu
38H	NVMDATAH	0000 0000	0000 0000	uuuu uuuu
39H	NVMDATAL	0000 0000	0000 0000	uuuu uuuu
3AH	NVMADDRH	0000 0000	0000 0000	uuuu uuuu
3BH	NVMADDRL	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
3CH	NVMCTL0	----	----	----
3DH	NVMCTL1	----	----	----
3EH	ADCDATA0L	xxxx xxxx	xxxx xxxx	uuuu uuuu
3FH	ADCCTL1	0000 00-0	0000 00-0	uuuu uu-u
40H	T2CCR0H	0000 0000	0000 0000	uuuu uuuu
41H	T2H	xxxx xxxx	xxxx xxxx	uuuu uuuu
42H	PP5H	xxxx xxxx	xxxx xxxx	uuuu uuuu
43H	PWM5H0	xxxx xxxx	xxxx xxxx	uuuu uuuu
44H	PWM5H1	xxxx xxxx	xxxx xxxx	uuuu uuuu
45H	P0LR	xxxx xxxx	xxxx xxxx	uuuu uuuu
46H	P2LR	xxxx xxxx	xxxx xxxx	uuuu uuuu
47H	P1LR	xxxx xxxx	xxxx xxxx	uuuu uuuu
48H	P3LR	xxxx xxxx	xxxx xxxx	uuuu uuuu
49H	TR3	1111 1111	1111 1111	uuuu uuuu
4AH	EIE3	0000 0000	0000 0000	uuuu uuuu
4BH	EIF3	0000 0000	0000 0000	uuuu uuuu
4CH	OSCCAL3	1000 1000	1000 1000	uuuu uuuu
4DH	OSCCAL0	0001 0000	0001 0000	uuuu uuuu
4EH	T3CTL	0000 0000	0000 0000	uuuu uuuu
4FH	T3L	0000 0000	0000 0000	uuuu uuuu
50H	ADCDATA1H	0000 0000	0000 0000	uuuu uuuu
51H	ADCDATA1L	0000 0000	0000 0000	uuuu uuuu
52H	PP5L	1111 1111	1111 1111	uuuu uuuu
53H	PINSET	0000 0000	0000 0000	uuuu uuuu
54H	T2CCR0L	0000 0000	0000 0000	uuuu uuuu
55H	PWM5L0	xxxx xxxx	xxxx xxxx	uuuu uuuu
56H	PWM5L1	xxxx xxxx	xxxx xxxx	uuuu uuuu
57H	PWM5CTL0	0000 0000	0000 0000	uuuu uuuu
58H	ADCINTCTL	0--- --00	0--- --00	u--- --uu
59H	ADCDATA2H	0000 0000	0000 0000	uuuu uuuu
5AH	ADCDATA2L	0000 0000	0000 0000	uuuu uuuu
5BH	PWM5CTL1	0000 0000	0000 0000	uuuu uuuu
5CH	P5ASCTL0	0000 0000	0000 0000	uuuu uuuu
5DH	PSTRCTL0	---0 0001	---0 0001	---u uuuu
5EH	ADCDATA3H	0000 0000	0000 0000	uuuu uuuu
5FH	T3H	0000 0000	0000 0000	uuuu uuuu
60H	PUR1	0000 0000	0000 0000	uuuu uuuu
61H	PUR2	0000 0000	0000 0000	uuuu uuuu
62H	TMRBUZ	---- --00	---- --00	---- --uu
63H	ADCCTL2	---- 0000	---- 0000	---- uuuu
64H	DAC12CTL	0000 0000	0000 0000	uuuu uuuu
65H	DAC12DH	0000 0000	0000 0000	uuuu uuuu
66H	DAC12DL	0000 0000	0000 0000	uuuu uuuu
67H	INTEDGCTL	00-- ---1	00-- ---1	uu-- ---u
6AH	ADCDATA3L	0000 0000	0000 0000	uuuu uuuu
6BH	IOCL3	0000 0000	0000 0000	uuuu uuuu
6CH	PUR3	0000 0000	0000 0000	uuuu uuuu
105H	CCP4L	0000 0000	0000 0000	uuuu uuuu
106H	CCP4H	0000 0000	0000 0000	uuuu uuuu
107H	CMCTL0	0000 0000	0000 0000	uuuu uuuu
108H	DIVRH	0000 0000	0000 0000	uuuu uuuu
109H	CCP4CTL	---0 0000	---0 0000	---u uuuu
10CH	PWM5L2	0000 0000	0000 0000	uuuu uuuu
10DH	PWM5L3	0000 0000	0000 0000	uuuu uuuu
10EH	PWM5OC	--00 0000	--00 0000	--uu uuuu
10FH	T2CTL1	0000 0--0	0000 0--0	uuuu u--u
110H	C4FILTCTL	000- -000	000- -000	uuu- -uuu
111H	C4FILTPRE	0000 0000	0000 0000	uuuu uuuu
112H	C2CTL	0000 0000	0000 0000	uuuu uuuu
113H	C3CTL	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
114H	C4CTL	0000 0000	0000 0000	uuuu uuuu
115H	DIVBH	0000 0000	0000 0000	uuuu uuuu
116H	T2CCR1L	0000 0000	0000 0000	uuuu uuuu
118H	T2CCR1H	0000 0000	0000 0000	uuuu uuuu
119H	PWM5FC	--00 0000	--00 0000	--uu uuuu
11AH	P5ASCTL1	0000 0000	0000 0000	uuuu uuuu
11BH	PSTRCTL1	---- 1010	---- 1010	---- uuuu
11CH	PWM5CTL2	--0- -000	--0- -000	--u- -uuu
11DH	PWM5PC	--00 0000	--00 0000	--uu uuuu
11EH	PWM5H2	0000 0000	0000 0000	uuuu uuuu
11FH	PWM5H3	0000 0000	0000 0000	uuuu uuuu
120H	RSCTL1	0000 000x	0000 000x	uuuu uuuu
121H	TXSDR1	0000 0000	0000 0000	uuuu uuuu
122H	RXSDR1	0000 0000	0000 0000	uuuu uuuu
123H	BRCTL1	0100 0000	0100 0000	uuuu uuuu
124H	TSCTL1	0000 0010	0000 0010	uuuu uuuu
125H	EUBRGL1	0000 0000	0000 0000	uuuu uuuu
126H	EUBRGH1	0000 0000	0000 0000	uuuu uuuu
127H	USLPEN1	0--- ----	0--- ----	u--- ----
128H	SSCICTL0	0000 0000	0000 0000	uuuu uuuu
12AH	SSCICTL1	0000 0000	0000 0000	uuuu uuuu
12BH	SSCISTA	0000 0000	0000 0000	uuuu uuuu
12CH	SSCIBUFR	0000 0000	0000 0000	uuuu uuuu
12DH	UARTMTH1	0000	0000	uuuu
12EH	SSCIADD	1111 1111	1111 1111	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	0000 0000	0000 0000	uuuu uuuu
13AH	DIVAH	0000 0000	0000 0000	uuuu uuuu
13BH	DIVAL	0000 0000	0000 0000	uuuu uuuu
13CH	DIVBL	0000 0000	0000 0000	uuuu uuuu
13DH	DIVQH	0000 0000	0000 0000	uuuu uuuu
13EH	DIVQL	0000 0000	0000 0000	uuuu uuuu
13FH	DIVRL	0000 0000	0000 0000	uuuu uuuu
14AH	C1FILTCTL	000- -000	000- -000	uuu- -uuu
14BH	C1FILTPRE	0000 0000	0000 0000	uuuu uuuu
14CH	C2FILTCTL	000- -000	000- -000	uuu- -uuu
14DH	C2FILTPRE	0000 0000	0000 0000	uuuu uuuu
14EH	C3FILTCTL	000- -000	000- -000	uuu- -uuu
14FH	C3FILTPRE	0000 0000	0000 0000	uuuu uuuu
150H	UPINSET1	0000 0000	0000 0000	uuuu uuuu
159H	VREFCAL0	0000 0000	0000 0000	uuuu uuuu
15AH	VREFCAL1	1010 0000	1010 0000	uuuu uuuu
15BH	HLVDCTL	0000 0000	0000 0000	uuuu uuuu
15DH	T3REL	0000 0000	0000 0000	uuuu uuuu
15EH	T3REH	0000 0000	0000 0000	uuuu uuuu
15FH	VRES0	0000 0000	0000 0000	uuuu uuuu
160H	T4L	0000 0000	0000 0000	uuuu uuuu
161H	T4H	0000 0000	0000 0000	uuuu uuuu
162H	T4REL	0000 0000	0000 0000	uuuu uuuu
163H	T4REH	0000 0000	0000 0000	uuuu uuuu
164H	T4CTL	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
165H	RC32KCAL	0000 0000	0000 0000	uuuu uuuu
206H	ANS4	1111 1111	1111 1111	uuuu uuuu
207H	ANS5	1111 1111	1111 1111	uuuu uuuu
208H	ANS6	1111 1111	1111 1111	uuuu uuuu
209H	ANS7	1111 ----	1111 ----	uuuu ----
20CH	P4	xxxx xxxx	xxxx xxxx	uuuu uuuu
20DH	P4LR	xxxx xxxx	xxxx xxxx	uuuu uuuu
20EH	TR4	1111 1111	1111 1111	uuuu uuuu
20FH	PUR4	1111 1111	1111 1111	uuuu uuuu
210H	P5	-xxx xxxx	-xxx xxxx	-uuu uuuu
211H	P5LR	-xxx xxxx	-xxx xxxx	-uuu uuuu
212H	TR5	-111 1111	-111 1111	-uuu uuuu
213H	PUR5	1111 1111	1111 1111	uuuu uuuu
214H	P6	-xxx xxxx	-xxx xxxx	-uuu uuuu
215H	P6LR	-xxx xxxx	-xxx xxxx	-uuu uuuu
216H	TR6	-111 1111	-111 1111	-uuu uuuu
218H	PUR6	1111 1111	1111 1111	uuuu uuuu
219H	P7	xxxx ----	xxxx ----	uuuu ----
21AH	P7LR	xxxx ----	xxxx ----	uuuu ----
21BH	TR7	1111 ----	1111 ----	uuuu ----
21CH	PUR7	1111 ----	1111 ----	uuuu ----
21EH	PWMCTL0	0000 0000	0000 0000	uuuu uuuu
21FH	PP1	1111 1111	1111 1111	uuuu uuuu
220H	PP2	1111 1111	1111 1111	uuuu uuuu
235H	PWM1AL	xxxx xxxx	xxxx xxxx	uuuu uuuu
236H	TEMPSNR	---- --00	---- --00	---- --uu
23FH	UPINSET2	0000 0000	0000 0000	uuuu uuuu
240H	U7816CTL2	0-00 0000	0-00 0000	u-uu uuuu
241H	U7816RXCTL2	0000 0000	0000 0000	uuuu uuuu
242H	U7816TXCTL2	0001 0000	0001 0000	uuuu uuuu
243H	CLKDIV2	0000 0000	0000 0000	uuuu uuuu
244H	EGTCTL2	0000 0000	0000 0000	uuuu uuuu
245H	TZBRGH2	0000 0000	0000 0000	uuuu uuuu
246H	TZBRGL2	0000 0000	0000 0000	uuuu uuuu
247H	UARTMTH2	0000 0000	0000 0000	uuuu uuuu
248H	USLPEN2	0--- ----	0--- ----	u--- ----
249H	BRCTL2	0100 0000	0100 0000	uuuu uuuu
24AH	RSCTL2	0000 0000	0000 0000	uuuu uuuu
24BH	RXSDR2	xxxx xxxx	xxxx xxxx	uuuu uuuu
24CH	EUBRGH2	0000 0000	0000 0000	uuuu uuuu
24DH	EUBRGL2	0000 0000	0000 0000	uuuu uuuu
24EH	TXSDR2	xxxx xxxx	xxxx xxxx	uuuu uuuu
24FH	TSCTL2	0000 0000	0000 0000	uuuu uuuu
250H	BKPCTL	0000 0000	0000 0000	uuuu uuuu
251H	EIE4	---- 0000	---- 0000	---- uuuu
252H	EIF4	---- 0000	---- 0000	---- uuuu
253H	IP4	---- 0000	---- 0000	---- uuuu
260H	PWM1AH	xxxx xxxx	xxxx xxxx	uuuu uuuu
261H	PWM1BL	xxxx xxxx	xxxx xxxx	uuuu uuuu
262H	PWM1BH	xxxx xxxx	xxxx xxxx	uuuu uuuu
267H	PWM1EL	xxxx xxxx	xxxx xxxx	uuuu uuuu
268H	PWM1EH	xxxx xxxx	xxxx xxxx	uuuu uuuu
269H	PWM1FL	xxxx xxxx	xxxx xxxx	uuuu uuuu
26AH	PWM1FH	xxxx xxxx	xxxx xxxx	uuuu uuuu
26CH	PWMCTL1	0000 0000	0000 0000	uuuu uuuu
318H	RTCSRT	0000 0000	0000 0000	uuuu uuuu
319H	RTCSTU	--00 0000	--00 0000	--uu uuuu
31CH	RTCALRS	-000 0000	-000 0000	-uuu uuuu
31DH	RTCALRM	-000 0000	-000 0000	-uuu uuuu
31EH	RTCALRH	--00 0000	--00 0000	--uu uuuu
31FH	RTCALRW	---- -000	---- -000	---- -uuu

地址	名称	上电复位	RST 复位 欠压检测	中断唤醒 WDT 超时唤醒
320H	RTCTTR	00-- -000	00-- -000	uu-- -uuu
321H	RTCFCR	0000 0000	0000 0000	uuuu uuuu
322H	RTCSEC	-000 0000	-000 0000	-uuu uuuu
323H	RTCMIN	-000 0000	-000 0000	-uuu uuuu
324H	RTCHOUR	-000 0000	-000 0000	-uuu uuuu
325H	RTCWEK	---- -000	---- -000	---- -uuu
326H	RTCDAY	--00 0000	--00 0000	--uu uuuu
327H	RTCMTH	---0 0000	---0 0000	---u uuuu
328H	RTCYEAR	0000 0000	0000 0000	uuuu uuuu
329H	EIE5	0000 0000	0000 0000	uuuu uuuu
32AH	EIF5	0000 0000	0000 0000	uuuu uuuu
32BH	IP5	0000 0000	0000 0000	uuuu uuuu
32CH	RTCTMRCTL	0000 0000	0000 0000	uuuu uuuu
32DH	RTCTMREN	---- --00	---- --00	---- --uu
32EH	RTCTMR1	0000 0000	0000 0000	uuuu uuuu
32FH	RTCTMR0	0000 0000	0000 0000	uuuu uuuu
360H	POWCTL	0000 0000	0000 0000	uuuu uuuu
361H	PCAL	0001 1111	0001 1111	uuuu uuuu
362H	XTALCAL	0101 0000	0101 0000	uuuu uuuu
363H	BWDTCTL	0000 0000	0000 0000	uuuu uuuu
364H	PHCLR	0000 0000	0000 0000	uuuu uuuu
365H	LPRCCAL	0000 0000	0000 0000	uuuu uuuu
366H	LPRCCTL	0000 0100	0000 0100	uuuu uuuu
367H	ULPKEY	0000 0000	0000 0000	uuuu uuuu
368H	BBODCTL	0000 0000	0000 0000	uuuu uuuu
369H	BKPREG0	0000 0000	0000 0000	uuuu uuuu
36AH	BKPREG1	0000 0000	0000 0000	uuuu uuuu
36BH	BKPREG2	0000 0000	0000 0000	uuuu uuuu
36CH	BKPREG3	0000 0000	0000 0000	uuuu uuuu

表 20-3 不同复位条件下对标志位的影响

POR	LVR	TO	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 RST 复位
u	u	1	0	休眠模式中的 RST 复位

图注: u=未发生变化

21 电源管理和功耗模式

KF8L15Z20XX 系列单片机提供备份区，当主电源 VDD 掉电后，通过 VBAT 脚为备份区提供电源，允许实时时钟（RTC）和备份寄存器（BKP-REGISTER）继续工作。

21.1 电源

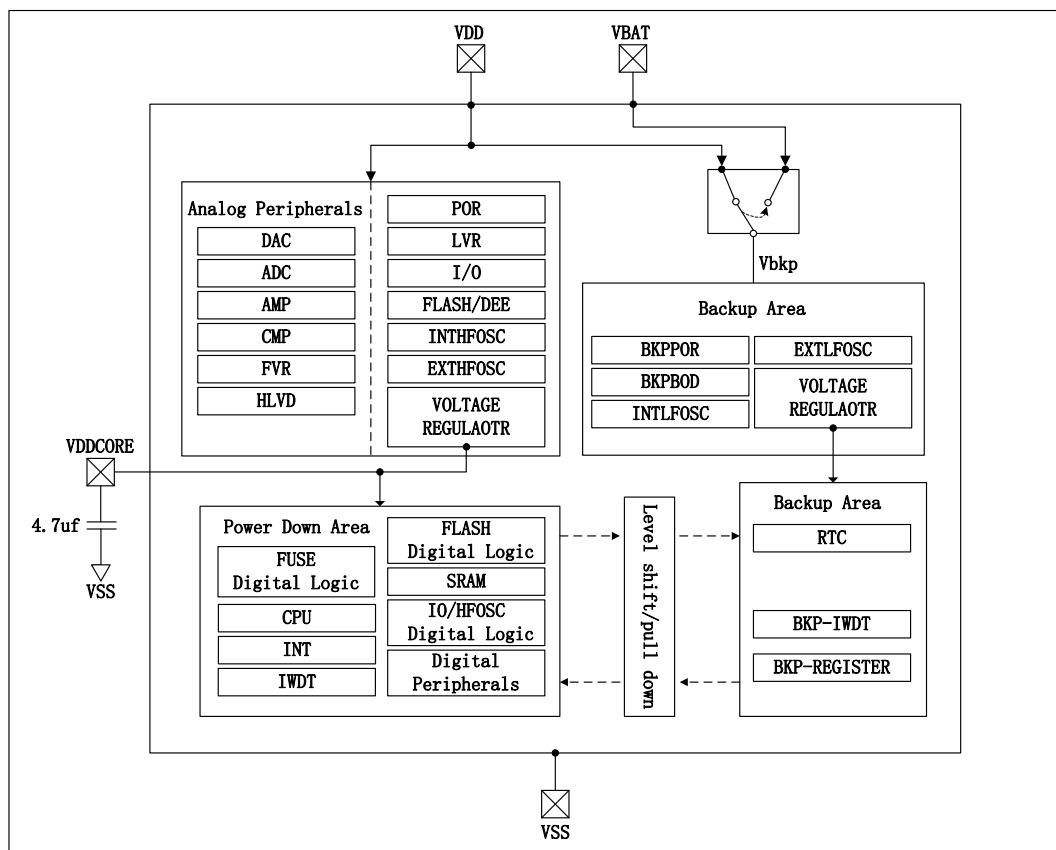


图 23-1: 电源管理系统框图

KF8L15Z20XX 系列单片机有四个和电源地相关的管脚：

- VDD：主供电源
- VBAT：电池脚，在主电源掉电时用于备份区供电
- VDDCORE：内置电压调节器电压，须外接 4.7uf 电容
- VSS：接地

KF8L15Z20XX 的工作电压为 1.8V~5.5V；VDDCORE 为内置电压调节器提供的 1.8V 电源（不同工作模式下，VDDCORE 的电压不同），VDDCORE 脚必须外接 4.7uf 电容，以确保单片机可正常工作；当主电源 VDD 掉电后，VDDCORE 掉电，VBAT 脚为备份区提供电源，实时时钟(RTC)继续运行，备份寄存器的数据不会丢失。

如果 VBAT 脚悬空，主供电源可以为备份区供电，当主供电源掉电后，备份区掉电。

21.1.1 备份区电源 Vb_{kp}

备份区内置电源开关，主 LVR 用于控制备份区电源开关。

当配置字的 LVREN 位为 0 时，LVR 被禁止，备份区电源开关固定选择 VDD 作为备份区的供电电源，即当主电源 VDD 掉电时，即使 VBAT 外接电源，备份区也会掉电。

当配置字的 LVREN 位为 1 时，LVR 被使能，备份区电源开关状态切换：

- 备份区电源开关从 VDD 切换到 VBAT：LVR 检测到低电压并发生复位事件；
- 备份区电源开关从 VBAT 切换到 VDD：LVR 检测到电压上升并发生退出复位事件。

21.1.2 内置电压调节器

KF8L15Z20XX 有两个内置电压调节器：主电压调节器和备份区电压调节器。在不同工作模式下，两个电压调节器的工作状态如下表所示：

表 23-1：内置电压调节器工作状态表

工作模式	主电压调节器	备份区电压调节器
运行模式	正常功耗模式运行	正常功耗模式运行
普通休眠模式	正常功耗模式运行	正常功耗模式运行
深度休眠模式	低功耗模式运行	低功耗模式运行
超低功耗模式	关闭	低功耗模式运行
VDD(供电电源)掉电	关闭	低功耗模式运行

21.2 电源管理

KF8L15Z20XX 系列单片机有一套 POR 和 LVR 电路，用于主供电电源 VDD；详细描述见复位章节相关内容。备份区有一套单独的 BKPPOR 和 BKPBOD 对备份区进行电源检测。

当 V_{b_{kp}} 低于 VBKPPOR 时，备份区保持复位状态；当 V_{b_{kp}} 高于 VBKPPOR 时，备份区退出复位；BKPBOD 通过软件使能，在使能 BKPBOD 条件下，当 V_{b_{kp}} 低于 VBKPBOD 时，BKPBOD 标志位置 1。

关于 POR、LVR、BKPPOR、BKPBOD 的细节请参考数据手册的电气特性相关章节。

21.3 备份区

KF8L15Z20XX 系列单片机备份区内资源如下：

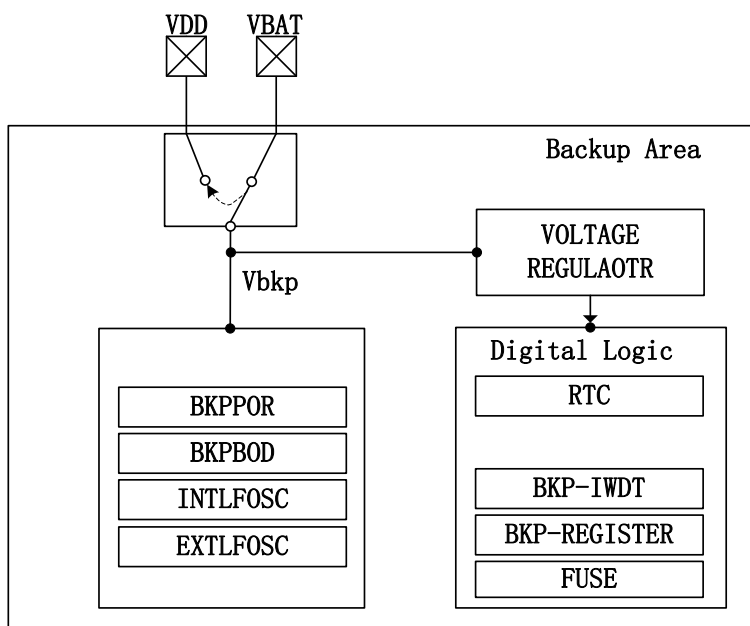


图 23-2: 备份区资源图

备份区内有独立的 BKPPOR、BKPBD 和电压调节器；当 LVR 使能且 VBAT 外接供电源的条件下，允许备份区在主供电电源掉电的情况下继续工作。BKPBD 通过 BBODCTL 寄存器控制，将 BBODEN 位置 1 可使能 BKPBD。BKPBD 带有间隔检测功能，当 BSCAN 位置 1 时，BKPBD 将利用 RTC 的秒时钟信号进行检测，即每秒检测一次。当 RTC 模块未使能时，BKPBD 的间隔检测功能将无法使用。

备份区内置一个专用看门狗模块，用于超低功耗模式的唤醒。

内部低频振荡器和外部低频振荡器也在备份区内，以保证实时时钟(RTC)和备份区专用看门狗（BKP-IWDT）可以正常工作。

21.3.1 备份区数据寄存器

备份区内配有 4 组数据寄存器，可用于超低功耗模式下的数据保存。

表 23-2 备份区数据寄存器表

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
369H	BKPREG0	备份区数据寄存器 0							
36AH	BKPREG1	备份区数据寄存器 1							
36BH	BKPREG2	备份区数据寄存器 2							
36CH	BKPREG3	备份区数据寄存器 3							

21.3.2 备份区复位

如下两种情况会将备份区复位，否则备份区内所有寄存器数据将被保持，被使能的外设将继续工作。

1. 供电源和电池电压均在备份区复位电压以下；
2. 发生外部 RST 复位。

21.3.3 备份区配置寄存器（BKPCTL）

用户在对备份区相关寄存器进行操作前，需要先对备份区配置寄存器进行配置，否则无法对备份区进行操作。备份区配置寄存器位于掉电区域（power down area），即超低功耗模式启动后，该寄存器将被复位。

寄存器24.1: BKPCTL：备份区控制寄存器(地址:250H)

复位值 0000 0000	bit7						bit0	
	BKP7	BKP6	BKP5	BKP4	BKP3	BKP2	BKP1	BKP0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- BKP7:** 备份区接口配置位
0 = 备份区接口处于默认状态
1 = 备份区接口开放，可对备份区内寄存器进行读写操作
- BKP6:** 保留位
- BKP5:** 备份区接口软件解锁位
0 = 关闭软件解锁备份区接口
1 = 软件解锁备份区接口
- BKP<4:0>:** 保留位

注：请在初始化程序中，将 BKP0 位置 1。

21.3.4 备份区寄存器的读写

333H 到 36CH 地址 SFR 位于备份区内，在对备份区内 SFR 进行读写操作前，需要将 BKPCTL 寄存器的 BKP7 位置 1，否则将无法进行读写操作。

```
MOVB #0X02      ; 切换至 2 区
SET BKPCTL,BKP7 ; 开放备份区接口
MOVB #0X00      ; 切换回 0 区
```

对备份区数据寄存器（BKPREG0-3）读写操作时，需要将 BKPCTL 寄存器的 BKP7 位和 PHCLR 寄存器的 REGCLR 位置 1，否则将无法进行读写操作。

```
MOVB #0X02      ; 切换至 2 区
SET BKPCTL,BKP7 ; 开放备份区接口
MOVB #0X03      ; 切换至 3 区
SET PHCLR, REGCLR ; 备份区数据寄存器组退出复位
...
```

21.4 功耗模式

KF8L15Z20XX 系列单片机提供丰富的功耗模式供用户选择，以满足用户对功耗的不同需求。KF8L15Z20XX 提供如下功耗模式：

- 正常运行模式 (Normal run mode, NR)
- 普通休眠模式 (Sleep mode, SLP)
- 深度休眠模式 (Deep sleep mode, DSLP)
- 超低功耗模式 (Ultra-Low power mode, ULP)
- 掉电模式 (Power Down mode, PD)

21.4.1 功耗模式相关寄存器

表 23-2: 功耗模式相关寄存器表

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
360H	POWCTL	IOLATCH	VCORM	FULATCH	PHPDM	WKPEN	WKPF	DSL PEN	ULPEN
364H	PHCLR	VRTCS	VLCDS	BWDTCLR	REGCLR	PDRTC	PDLCD	EXTLFEN	INTLFEN
367H	ULPKEY	UKEY7	UKEY6	UKEY5	UKEY4	UKEY3	UKEY2	UKEY1	UKEY0
368H	BBODCTL	BBODEN	BSCAN	-	-	-	-	-	BBODF

21.4.1.1 功耗模式控制寄存器

寄存器23.1: POWCTL: 功耗模式控制寄存器(地址:360H)

复位值 0000 0000	bit7							bit0	
	IOLATCH	VCORM	FULATCH	PHPDM	WKPEN	WKPF	DSL PEN	ULPEN	
	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	
IOLATCH:	IO 口状态锁存控制位								
	0 =IO 口状态未锁存								
	1 =IO 口状态被锁存								
VCORM:	保留位								
FULATCH:	FUSE 状态锁存控制位								
	0 = 未软件锁存 FUSE 状态								
	1 = 软件锁存 FUSE 状态								
PHPDM:	保留位								
WKPEN:	外部引脚唤醒使能位								
	0 = 禁止外部引脚唤醒								
	1 = 使能外部引脚唤醒，外部唤醒引脚 P0.6 的上升沿将唤醒单片机								
WKPF:	外部唤醒引脚标志位								
	0 = 未发生外部唤醒事件								
	1 = 发生外部唤醒事件								
DSL PEN:	深度休眠模式使能位								
	0 = 禁止深度休眠模式								
	1 = 使能深度休眠模式								
ULPEN:	超低功耗模式使能位								
	0 = 禁止超低功耗模式								
	1 = 使能超低功耗模式								

注：编程时 FULATCH 位将被硬件清零。

VCORM 位为系统保留位，该位默认为 0，请勿将该位置 1。

21.4.1.2 超低功耗模式解锁寄存器

寄存器23.2: ULPKEY：超低功耗模式解锁寄存器(地址:367H)

复位值 0000 0000	bit7							bit0	
	UKEY7	UKEY6	UKEY5	UKEY4	UKEY3	UKEY2	UKEY1	UKEY0	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

UKEY<7:0>：超低功耗模式解锁位

当 ULPEN=0 时，ULPKEY 寄存器保持复位状态，UKEY=00H

当 ULPEN=1 时，ULPKEY 寄存器退出复位状态，对 ULPKEY 寄存器写入 C0H，单片机将进入超低功耗模式

21.4.1.3 备份区外设复位寄存器

寄存器23.3: PHCLR：备份区外设复位寄存器(地址:364H)

复位值 1100 0000	bit7						bit0	
	VRTCS	VLCDS	BWDTCLR	REGCLR	PDRTC	PDLCD	EXTLFEN	INTLFEN
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

VRTCS：保留位，请勿清零

VLCDS：保留位，请勿清零

BWDTCLR：备份区专用看门狗模块复位控制位
0 = 备份区专用看门狗模块处于复位状态
1 = 备份区专用看门狗模块退出复位状态

REGCLR：备份区数据寄存器复位控制位
0 = 备份区数据寄存器处于复位状态
1 = 备份区数据寄存器退出复位状态

PDRTC：保留位

PDLCD：保留位

EXTLFEN：外部低频振荡器软件使能位
0 = 软件未使能外部低频振荡器
1 = 软件使能外部低频振荡器

INTLFEN：内部低频振荡器软件使能位
0 = 软件未使能内部低频振荡器
1 = 软件使能内部低频振荡器

注：1.如需使用备份区专用看门狗，须将复位控制位置 1 以退出复位状态，否则无法使能；
2.当外设时钟选择内部低频时钟或者外部低频时钟时，硬件将自动使能对应振荡器，即使振荡器软件使能位未置 1。

21.4.1.4 备份区 BOD 控制寄存器

寄存器23.4: BBODCTL: 备份区BOD控制寄存器(地址:368H)

复位值 0000 0000	bit7						bit0	
	BBODEN	BSCAN	-	-	-	-	-	BBODF
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BBODEN: 备份区 BOD 使能位

0 = 禁止备份区 BOD

1 = 使能备份区 BOD

BSCAN: 备份区 BOD 检测模式选择位

0 = 常规模式, BOD 持续检测

1 = 间隔检测模式, BOD 每 1 秒内检测一次, 检测时间为 32us

BBODF: 备份区 BOD 标志位

0 = 未检测到 Vb_{kp} 低电压

1 = 检测到 Vb_{kp} 低电压

21.4.1.5 备份区专用看门狗控制寄存器

寄存器24.3: BWDCTL: 备份区专用看门狗控制寄存器(地址:363H)

复位值 0000 0000	bit7						bit0	
	BWDTEN	BPS3	BPS2	BPS1	BPS0	BAPS2	BAPS1	BAPS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BWDTEN: 备份区专用看门狗使能位

0 = 禁止备份区专用看门狗

1 = 使能备份区专用看门狗

BPS<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

0101 = 1: 1024

0110 = 1: 2048

0111 = 1: 4096

1000 = 1: 8192

1001 = 1: 16384

1010 = 1: 32768

1011 = 1: 65536

11xx = 保留

BAPS<2:0>: 看门狗定时器后分频比选择位

000 = 1: 1

001 = 1: 2

010 = 1: 4

011 = 1: 8

100 = 1: 16

101 = 1: 32

110 = 1: 64

111 = 1: 128

21.4.2 正常运行模式

单片机退出上电复位后，处于正常运行模式（Normal run mode）下；MCU 正常工作（系统时钟运行），所有外设均可使用，内部高频振荡器、内部低频振荡器、外部高频振荡器和外部低频振荡器均被允许使用。用户可以按照功能需求对 MCU 进行设置以达到最佳性能。

21.4.3 普通休眠模式

在如下配置条件下执行 IDLE 指令，单片机将进入普通休眠模式。

- **BKPCTL_bit0=1**
- POWCTL_DSLPEN=0
- POWCTL_ULPEN=0

普通休眠模式下，系统时钟被禁止，CPU 停止工作，程序停止运行；RAM 保持；部分外设被禁止使用，部分外设允许使用内部低频时钟或者外部低频时钟继续工作。

表 23-3：普通休眠模式信息表

功耗模式	配置	状态	允许的外设
正常运行模式	BKPCTL_bit0=1 POWCTL_DSLPEN=0 POWCTL_ULPEN=0	系统时钟被禁止 CPU 停止工作 程序停止运行 RAM 保持	T1/T3/T4 UART RTC IWDG

21.4.4 深度休眠模式

在如下配置条件下执行 IDLE 指令，单片机将进入深度休眠模式。

- **BKPCTL_bit0=1**
- POWCTL_DSLPEN=1
- POWCTL_ULPEN=0

深度休眠模式与普通休眠模式在禁止和开放资源上相同，区别在于深度休眠模式有更低的功耗和更长的唤醒时间，详细数据请参考电气特性。

21.4.5 普通/深度休眠模式下 I/O 口注意事项

为使休眠状态下（普通休眠模式和深度休眠模式）的电流消耗降至最低，应使所有 I/O 口状态确定，如果有的端口没有使用，最好设置为输入，接到 VDD 或 VSS 上，或者打开弱上拉。如果没用的端口悬空，应设置为输出，以确保 I/O 引脚没有耗散电流产生，其他在休眠时不用的外设都要关闭。

21.4.6 普通/深度休眠模式唤醒方式

单片机进入普通/深度休眠模式后由于工作的需要，要将单片机从休眠状态唤醒，在 KF8L15Z20XX 中可通过以下方式将单片机从休眠状态唤醒：

1. RST 引脚上输入的外部复位
2. 看门狗定时器唤醒(如果 WDT 已被使能)
3. INT0 内部中断
4. P0 口电平变化中断
5. 外设中断

RST 引脚输入的复位信号在唤醒单片机的同时也将导致单片机复位。其它唤醒时将单片机从休眠状态唤醒,并不会导致复位。可通过状态寄存器中的 $\overline{TO}$ 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,而当器件从休眠状态唤醒时,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 指令加一条 NOP 指令即可。

21.4.7 超低功耗模式

为使芯片有更低的功耗, KF8L15Z20XX 系列单片机提供超低功耗模式。该模式下,内置电压调节器关闭,除备份区以外的数字逻辑部分(系统时钟/CPU/RAM/数字外设等)掉电,最大程度节省功耗。超低功耗模式下,备份区的内置电压调节器以极低的功耗维持备份区正常工作。

超低功耗模式被外部唤醒引脚、闹钟中断或者备份区专用看门狗唤醒后,单片机被复位,系统时钟工作并重新开始执行程序;备份区内仍然继续工作、数据保持不受唤醒操作的影响。如果使能外部复位引脚 P0.0/RST 的外部复位功能,当发生外部复位事件时,备份区将被一起复位。

21.4.7.1 进入超低功耗模式的方法

通过如下操作步骤使芯片进入超低功耗模式:

1. 将 P0.1 口配置为输出口,并输出低电平;将 P0.2 口配置为数字输入口;
2. 将 POWCTL 寄存器的 IOLATCH 位置 1,锁存当前 IO 口状态;
3. 将 POWCTL 寄存器的 ULPEN 位置 1,使能超低功耗模式;
4. 对 ULPKEY 寄存器进行写 C0H 操作,单片机进入超低功耗模式。

MOVB #0X00	;切换到BANK0
*CLR TR0, 1	;P0.1口配置为输出口
*CLR P0LR, 1	;P0.1口输出低电平
SET TR0, 2	;P0.2口配置
MOVB #0X03	;切换到BANK3
SET POWCTL, IOLATCH	;锁存IO口当前状态
SET POWCTL, VCORM	;内置电压调节器工作在低功耗模式
SET POWCTL, ULPEN	;使能超低功耗模式
MOV R0, #0XC0	
MOV ULPKEY, R0	;写寄存器解锁,单片机进入超低功耗模式
NOP	

图 23-3: 超低功耗模式使能操作样例程序

注：进入超低功耗模式后，P0.1 和 P0.2 口将被强制为数字口，且 P0.2 口的弱上拉功能被强制使能，以用于超低功耗模式下可能进行的编程操作；因此在锁存 IO 口状态之前，将 P0.2 口配置为输入口（请勿配置为输出口以防额外电流产生）；如果 P0.1 口悬空，可将 P0.1 口配置为输出口；或者 P0.1 口外接低电平或者高电平。

在执行完最后一步的 ULPKEY 寄存器赋值后，不需要执行 IDLE 指令，芯片将进入超低功耗模式。

21.4.7.2 超低功耗模式的唤醒方式

KF8L15Z20XX 系列单片机在超低功耗模式下可以通过如下几种方式唤醒：

1. 外部复位引脚（P0.3/RST）复位
2. 外部引脚（P0.6/ULPINT）唤醒
3. RTC 闹钟唤醒
4. 备份区专用看门狗唤醒

超低功耗模式被外部唤醒引脚、RTC 闹钟中断或者备份区专用看门狗唤醒后，单片机被复位，系统时钟工作并重新开始执行程序；备份区内仍然继续工作、数据保持不受唤醒操作的影响。如果使能外部复位引脚 P0.0/RST 的外部复位功能，当发生外部复位事件时，备份区将被一起复位。

POWCTL 寄存器的 ULPEN 位和 ULPKEY 寄存器在唤醒后被硬件清零。

注：使用外部引脚唤醒超低功耗模式，软件在使能之前（POWCTL 寄存器的 WKPEN 位置 1 之前），需要对 POWCTL 寄存器的 WKPEN 位进行清零操作；因为单片机从超低功耗模式下被外部引脚唤醒时，备份区不会被复位，因此唤醒信息将被保存，WKPF 位置 1；WKPF 位为只读位，清零 WKPEN 位操作将清零 WKPF 位；如果不清零 WKPF 位，将无法再接收到外部引脚唤醒信号。样例如下：

MOVB #0X03	；切换到BANK3
CLR POWCTL, WKPEN	；禁止外部引脚唤醒功能并清除上次唤醒状态
SET POWCTL, WKPEN	；使能外部引脚唤醒功能

21.4.7.3 超低功耗模式下 I/O 口状态

在使能超低功耗模式之前，用户需要软件锁存 I/O 口的状态。将 POWCTL 寄存器的 IOLATCH 位置 1 后，单片机将 I/O 口的 TRx 寄存器和 PxLR 寄存器的状态进行锁存操作，以防止超低功耗模式下 I/O 口数字逻辑部分掉电后，对 I/O 口状态造成影响。

POWCTL 寄存器的 IOLATCH 位置 1 后，任何对 TRx 寄存器和 PxLR 寄存器进行写操作都不会改变 I/O 口的状态。直到 IOLATCH 位清零，I/O 口状态才会被更新。POWCTL 寄存器在备份区内，因此即使从超低功耗模式唤醒后（非复位），IOLATCH 位仍为 1，用户需要将该位清零。

IOLATCH 位不会对 ANSx 寄存器进行锁存操作，因此从超低功耗模式唤醒后，ANSx 寄存器被复位为 FFH。

MOVB #0X03	; 切换到BANK3
CLR POWCTL, IOLATCH	; IO口状态解锁

图 23-4: 超低功耗模式唤醒后须执行的样例程序

21.4.7.4 备份区专用看门狗

备份区内置一个专用看门狗定时器 BKP-WDT，计数时钟源由内部低频振荡器提供（ $F_{INTLF} \approx 31.25\text{KHZ}$ ）。使用 BKP-WDT 前，需要将 PHCLR 寄存器的 INTLFEN 位置 1 使能内部低频振荡器，否则 BKP-WDT 无法正常使用。

使用时，先将 PHCLR 寄存器的 BWDTCCLR 位置 1 使 BKP-WDT 退出复位状态，再对 BWDTCCTL 寄存器进行配置。备份区专用看门狗的周期：

$$T_{BKP-WDT} = \frac{2^{BPS\langle 30 \rangle + 5} \times 2^{BAPS\langle 2:0 \rangle}}{F_{INTLF}}$$

注： F_{INTLF} 为内部低频振荡器频率，约 31.25KHZ。

备份区专用看门狗溢出时不会对备份区进行复位，可用于超低功耗模式唤醒，详见超低功耗模式的唤醒方式章节。

22 看门狗定时器

为了防止单片机在正常工作时程序跑飞，KF8L15Z20XX 提供一个看门狗定时器。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。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。在休眠模式看门狗仍能正常运行，超时将唤醒单片机并使其继续执行 IDLE 后面的指令。

22.1 看门狗相关寄存器

表 14-1 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	POR	LVR
12FH	WDTPS	-	-	-	-	WDTPS3	WDTPS2	WDTPS1	WDTPS0

22.1.1 看门狗预分频选择寄存器 WDTPS

寄存器14.1: WDTPS: WDT预分频选择寄存器(地址:12FH)

bit7	bit6	bit5	bit4	WDTPS3	WDTPS2	WDTPS1	WDTPS0
-	-	-	-	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
 其他 = 保留

22.2 看门狗的开启方式

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。

22.3 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，状态字寄存器 PSW 中的 $\overline{TO}$ 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。

22.4 看门狗的周期

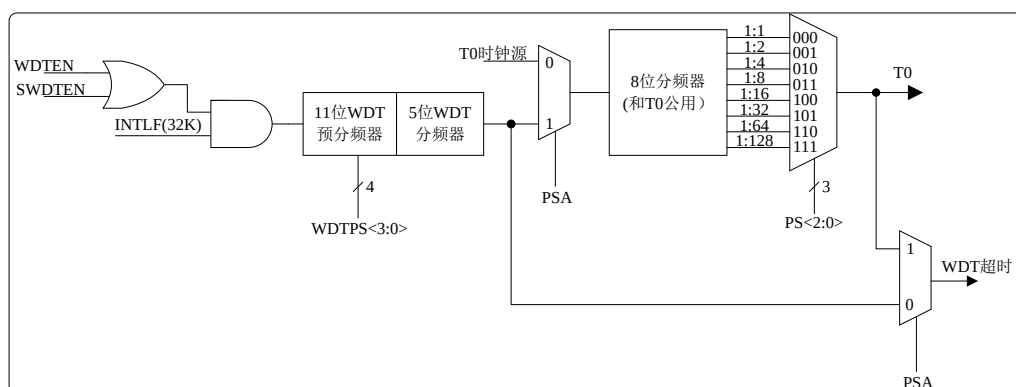


图 14.1 看门狗定时器框图

看门狗周期计算公式如下：

$$T_{WDT} = \frac{2^{WDTPS<30>+5} \times 2^{PS<2:0>}}{F_{INTLF}}$$

注：F_{INTLF} 为内部低频振荡器频率，约 31.25KHZ。

如上框图所示：看门狗定时器的时钟源为内部低频时钟（带校正功能的 31.25KHZ），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，须对看门狗定时器进行清狗操作（执行 CWDTC 指令）。否则，可能引起芯片的异常复位。

23 电气规范

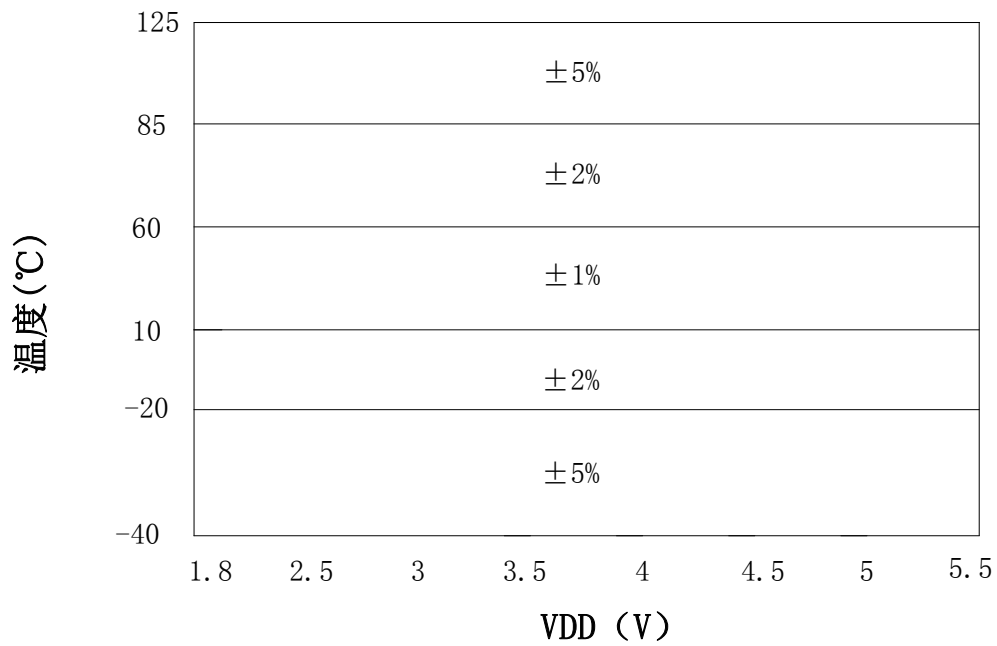
23.1 极限参数值

表 22-1 芯片极限参数值

极限参数值		
序号	参数说明	参数范围
1	偏置电压下的环境温度	-40℃~125℃
2	储存温度	-40℃~150℃
3	VDD 相对于VSS 的电压	5.5V
4	VPP 相对于Vss 的电压	12.5V
5	其它引脚相对于VSS 的电压	5.5V
6	VSS 引脚的最大输出电流	80mA
7	VDD 引脚的最大输入电流	80mA
8	任一I/O 引脚的最大输出灌电流	15mA
9	任一I/O 引脚的最大输出拉电流	15mA
10	I/O口 的最大灌电流	80mA
11	I/O口 的最大拉电流	80mA

备注:如果器件的工作条件超过“最大值”，可能会对器件造成永久性损坏。上述值仅为运行条件极大值，建议不要使器件在该规范规定的范围以外运行。器件长时间工作在最大值条件下，其稳定性会受到影响。

23.2 HFINTOSC 的频率精度与 VDD 和温度之间的关系



23.3 静态电流特性

表22-2芯片静态电流（ I_{DD} ）特性

测试条件:25 ℃						
序号	测试条件		最小值	典型值	最大值	单位
	振荡频率	VDD(V)				
1	16MHz	5.0	-	1310	-	μA
		3.3	-	1310	-	
		2.5	-	1310	-	
2	8MHz	5.0	-	780	-	
		3.3	-	780	-	
		2.5	-	780	-	
3	4MHz	5.0	-	520	-	
		3.3	-	520	-	
		2.5	-	520	-	
4	2MHz	5.0	-	380	-	
		3.3	-	380	-	
		2.5	-	380	-	
5	1MHz	5.0	-	320	-	
		3.3	-	320	-	
		2.5	-	320	-	
6	500kHz	5.0	-	285	-	
		3.3	-	285	-	
		2.5	-	285	-	
7	250kHz	5.0	-	265	-	
		3.3	-	265	-	
		2.5	-	265	-	
8	62.5kHz	5.0	-	255	-	
		3.3	-	250	-	
		2.5	-	250	-	

注 1: 在正常的工作模式下， I_{DD} 测量的条件为:所有I/O 引脚均设置为输出低，RST = V_{SS}，禁止WDT，关闭时钟输出。

2: 供电电流主要随工作电压和频率而变化。其它因素，如I/O 引脚负载和开关速率、内部代码执行模式和温度也会影响电流消耗。

23.4 休眠电流特性

表22-3芯片休眠电流（I_{DD}）特性

测试条件:25℃							
序号	休眠模式	测试条件		最小值	典型值	最大值	单位
			VDD(V)				
1	普通休眠模式	WDT、BOR、比较器等外设被禁止	5.0	-	20	30	μ A
			3.3	-	20	30	
			2.5	-	20	30	
2	深度休眠模式	WDT、BOR、比较器等外设被禁止	5.0	-	2.8	5	
			3.3	-	2.3	5	
			2.5	-	2.1	5	
3	超低功耗模式	WDT、BOR、比较器等外设被禁止	5.0	-	0.8	2	
			3.3	-	0.6	2	
			2.5	-	0.5	2	

23.5 外设电流特性

表22-4芯片外设电流特性

测试条件:25 ℃							
序号	测试参数	测试条件		最小值	典型值	最大值	单位
			VDD(V)				
1	欠电压复位电流 (I_{LVR})	WDT、RTC 等外设被禁止	5.0	-	9.5	14.0	uA
			3.3	-	6.0	8.5	
			2.5	-	4.0	6.0	
2	WDT	在超低功耗模式下	5.0	-	0.9	1.3	
			3.3	-	0.8	1.1	
			2.5	-	0.7	1.0	
3	RTC	在超低功耗模式下	5.0	-	0.9	1.3	
			3.3	-	0.8	1.1	
			2.5	-	0.7	1.0	
4	LCD	在超低功耗模式下	5.0	-	1.8	2.4	
			3.3	-	1.1	1.5	
			2.5	-	0.7	0.9	
5	比较器电流 (ICMP)	使能单个比较器	5.0	-	44	62	
			3.3	-	24	35	
			2.5	-	15	22	
6	运放	使能运放, 无负载	5.0	-	48	88	
			3.3	-	45	83	
			2.5	-	43	80	
7	ADC	使能 ADC, 未转换	5.0	-	895	1350	
			3.3	-	765	1155	
			2.5	-	515	760	
8	DAC	使能 ADC, 未转换	5.0	-	105	180	
			3.3	-	80	140	
			2.5	-	70	120	
9	HLVD	使能 ADC, 未转换	5.0	-	30	42	
			3.3	-	27	30	
			2.5	-	26	38	
10	FVR	使能 FVR, 无负载	5.0	-	70	80	
			3.3	-	70	78	
			2.5	-	68	78	

- 注 1: 外设电流是基本IDD 或IPD 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IDD 或IPD 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。

23.6 I/O 端口电平和芯片供电电压特性

表 22-5 芯片 IO 端口电平特性

测试条件（特别声明除外）： 工作温度 -40℃~85℃(工业级) -40℃~125℃(扩展级)						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
VIL	输入低电平 I/O引脚 采用TTL缓冲器		V _{SS}	—	0.15VDD	V
	采用施密特缓冲触发器		V _{SS}	—	0.2VDD	V
VIH	输入高电平 I/O端口 采用TTL缓冲器		VDD-0.6	—	V _{DD}	V
	采用施密特缓冲触发器		0.8VDD	—	V _{DD}	V
VOL	输出低电压		—	—	0.6	V
VOH	输出高电压		VDD-0.6	—	—	V
IIL	输入漏电流	VSS<VI<VDD	-1	—	1	μA

表 22-6 芯片供电电压特性

测试条件（特别声明除外）： 工作温度 -40℃~85℃(工业级) -40℃~125℃(扩展级)						
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		1.8	—	5.5	V

23.7 POR

表 22-7 芯片 POR 特性

测试条件（特别声明除外）： 工作温度 -40℃~85℃(工业级) -40℃~125℃(扩展级)						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
VPOR	上电复位电压	校准	1.6	1.7	1.8	V
Tempo	复位延时		-	2.5	-	ms
Idd	静态电流	VDD=3.3V	-	147	-	nA

23.8 LVR

表 22-8 芯片 LVR 特性

测试条件（特别声明除外）： 工作温度 -40℃~85℃(工业级) -40℃~125℃(扩展级)						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
VLVR	掉电复位电压		1.8	-	2.1	V
Hysteresis	迟滞		-	40	-	mV
Idd	静态电流	VDD=3.3V	-	5	-	uA

23.9 POWER18

表 22-9 芯片 POWER18 特性

测试条件（特别声明除外）： 工作温度 -40℃~85℃(工业级) -40℃~125℃(扩展级)						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
Vout	内部低压供电电源， 须外接4.7uF	2.8V≤VDD≤5.5V 1.8V≤VDD<2.8V	1.782	1.8	1.818 1.836	V
Idd	静态电流			28		μA

23.10 BACKUP BOD

表 22-10 芯片 BACKUP BOD 特性

测试条件（特别声明除外）： 工作温度 -40℃~85℃(工业级) -40℃~125℃(扩展级)						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
VBOD	电池电压检测			1.8		V
I _{dd}	静态电流	VDD=3.3V, 间隔检测		21*0.000032/T		μA

23.11 INTHF

表 22-11 芯片 INTHF 特性

测试条件（特别声明除外）： 工作温度 -40℃~85℃(工业级) -40℃~125℃(扩展级)						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
f _{CLKOUT}	HOSC frequency	temp=25℃	15.84	16	16.16	MHz
TRIM	User trimming step	temp=25℃	0.1	0.2	0.35	%
DuCy	Duty Cycle	—	45	—	55	%
t _{su}	Oscillator start-up time	—	—	5	—	μs
t _{stab}	Oscillator stabilization time	—	—	17	—	μs
I _{POWER_ISS}	Oscillator power consumption	temp=25℃	—	133	—	μA

23.12 DAC12

表 22-12 芯片 DAC12 特性

测试条件（特别声明除外）： 工作温度 -40℃~85℃(工业级) -40℃~125℃(扩展级)						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
V _o	输出范围		0		VREF	V
INL	积分非线性					
DNL	微分非线性					
I _{load}	驱动电流	VDD=3.3V 0.3V<V _o <VREF-0.3V		3		mA
T _{s_fs}	满摆幅建立时间	VDD=3.3V C _{load} =50pF		10		μs
T _{s_cc}	邻码建立时间	VDD=3.3V C _{load} =50pF		2		μs
I _{dd}	工作电流			220		μA

23.13 ADC12

表 22-13 A/D 转换器（ADC12）特性

测试条件（特别声明除外）： 工作温度 -40℃~85℃(工业级) -40℃~125℃(扩展级)						
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 ⁽¹⁾		-	±2	-	LSB
E_{GN}^*	增益误差		-	±2	-	LSB
V_{AIN}^*	满量程范围		V_{SS}	-	V_{REF}	V
T_{CNV}^*	AD转换时间		-	11	-	T_{AD}
T_{sOP} Ale^*	AD转换速率		-	-	400	KSPS
I_q^*	AD工作电流			550		uA

注：（1）*表示该数据为设计值。

（2）失调误差固定为 5mV 正失调，针对不同的参考电压对应为不同值，使用时可以直接减去绝对值。

23.14 CMP

表 22-14 模拟比较器模块特性

测试条件（特别声明除外）： 工作温度 -40℃~85℃(工业级) -40℃~125℃(扩展级)						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
A_v^*	开环增益			72		dB
V_{icmr}^*	输入电压范围		V_{SS}		$V_{DD}-1$	V

*该数据为设计值

23.15 OP

表 22-15 运算放大器模块直流特性

测试条件（特别声明除外）： 工作温度 -40℃~85℃(工业级) -40℃~125℃(扩展级)							
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 _{OL} *	输出电压摆幅	GND	—	VDD		
	I _Q *	静态电流	—	47		uA	
	I _{source} *	源电流	—	5		mA	
	I _{sink} *	灌电流	—	8.3		mA	
交流	GBW*	单位增益带宽	—	1		MHz	C _{load} =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	C _{load} =100pF
	Setting time*		—	4	—	us	C _{load} =100pF

*该数据为设计值

23.16 FVR

表 22-16 内部参考模块特性

工作温度：25℃							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
1	V _{out}	输出电压	1.9 2.85 3.8	2 3 4	2.1 3.15 4.2	V	2.6V ≤ VDD ≤ 5.5 V
2	LR*	线性调整率	-1	—	1	%/V	ΔV _{ref} /ΔV _{in}
3	TC	温度漂移	-5		+5	%	-40℃~125℃

*该数据为设计值

23.17 INTLF

表 22-17 INTLF 特性

测试条件（特别说明除外）： 工作温度 -40℃~85℃(工业级) -40℃~125℃(扩展级)						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
Freq*	振荡频率	校准后		32		KHz
Iq*	静态电流			330		nA

*该数据为设计值

23.18 EXTLF

表 22-18 EXTLF 特性

测试条件（特别说明除外）： 工作温度 -40℃~85℃(工业级) -40℃~125℃(扩展级)						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
Freq	振荡频率			32.768		KHz
Iq*	静态电流			340		nA

*该数据为设计值

23.19 EXTHF

表 22-19 EXTHF 特性

测试条件（特别说明除外）： 工作温度 -40℃~85℃(工业级) -40℃~125℃(扩展级)						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
Freq	振荡频率			20		MHz
Iq*	静态电流			4		mA

*该数据为设计值

23.20 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工作才可以确保正常运行。

图22.1: 不同VDD 时典型IDD — FOSC 关系曲线图(HFINTOSC模式)

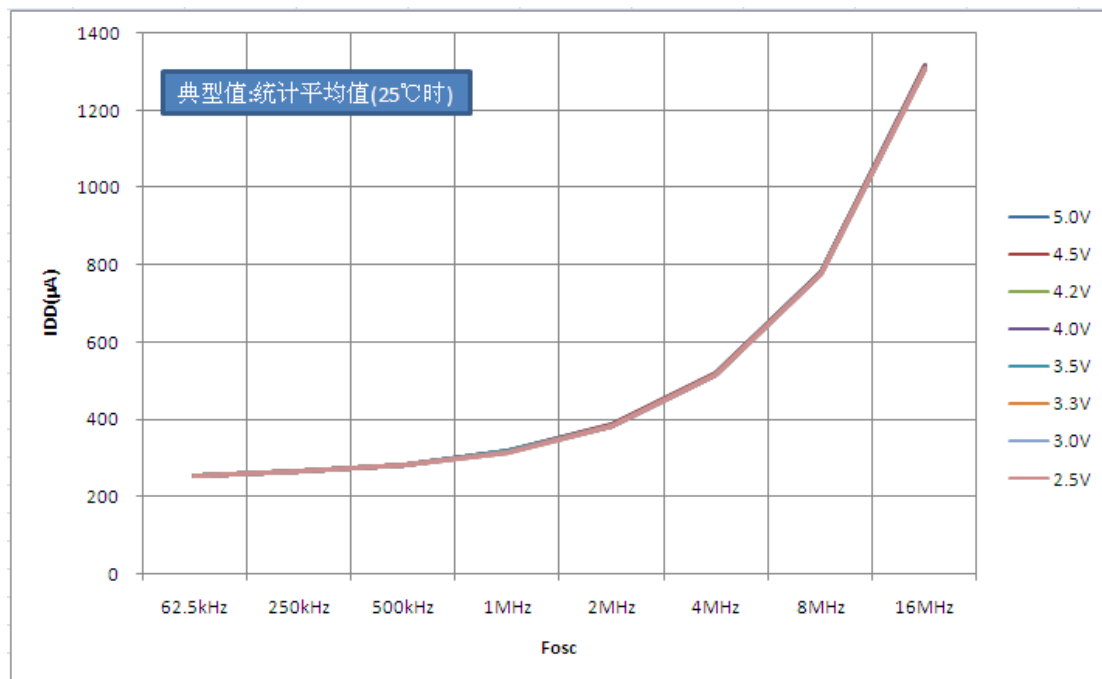


图22.2: 不同VDD 时典型ISLP — VDD 关系曲线图(普通休眠模式, 禁止所有外设)

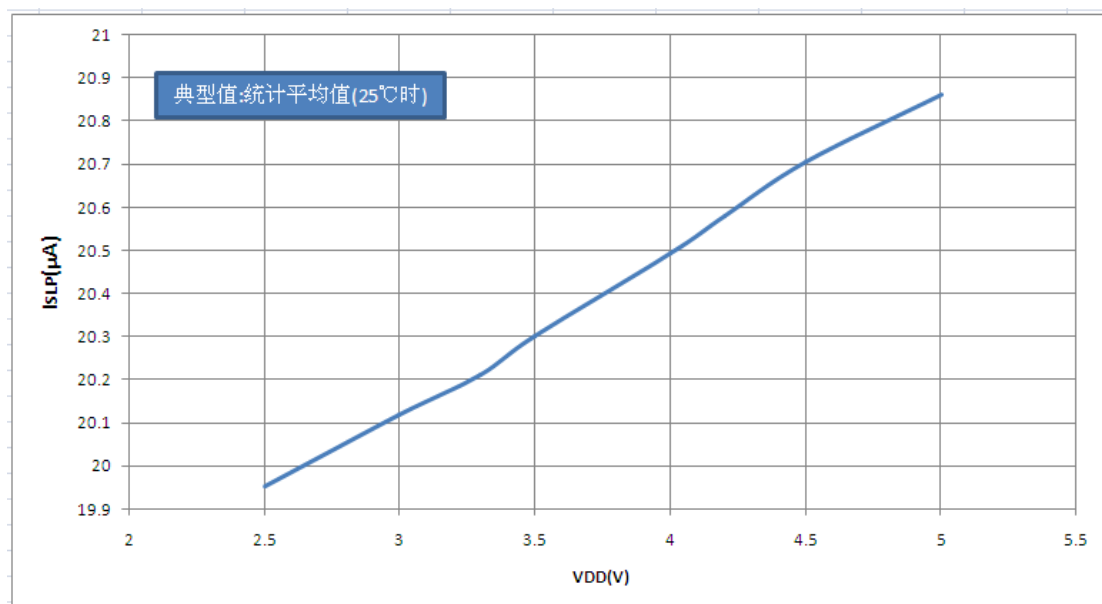


图22.3: 不同VDD 时典型IDSLP — VDD 关系曲线图(深度休眠模式, 禁止所有外设)

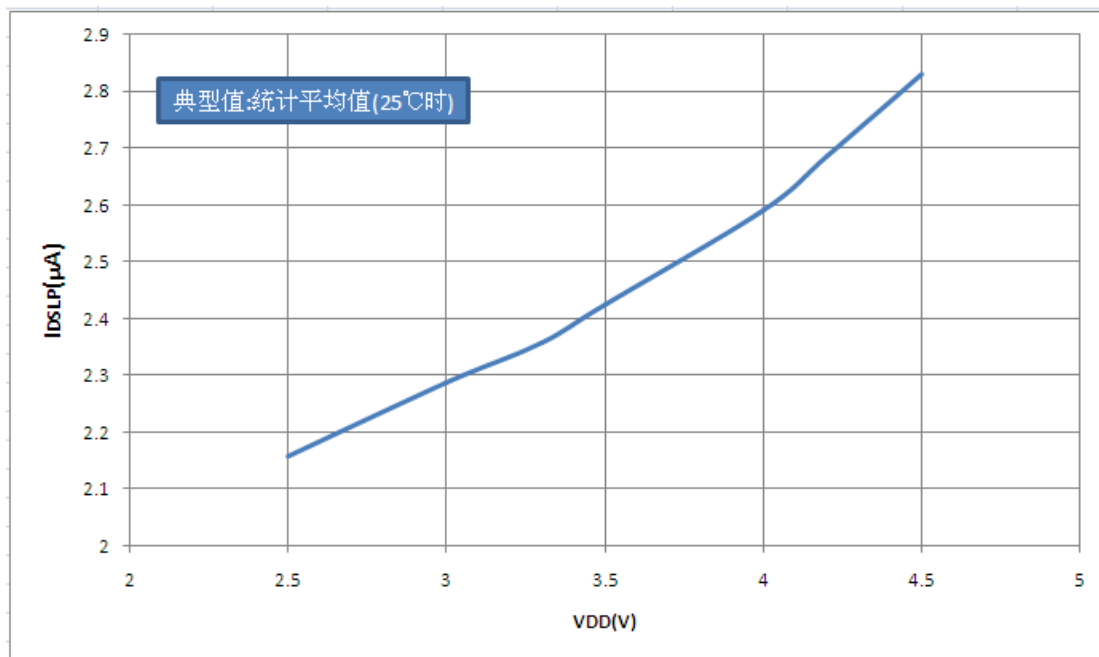


图22.4: 不同VDD 时典型IULP — VDD 关系曲线图(超低功耗模式, 禁止所有外设)

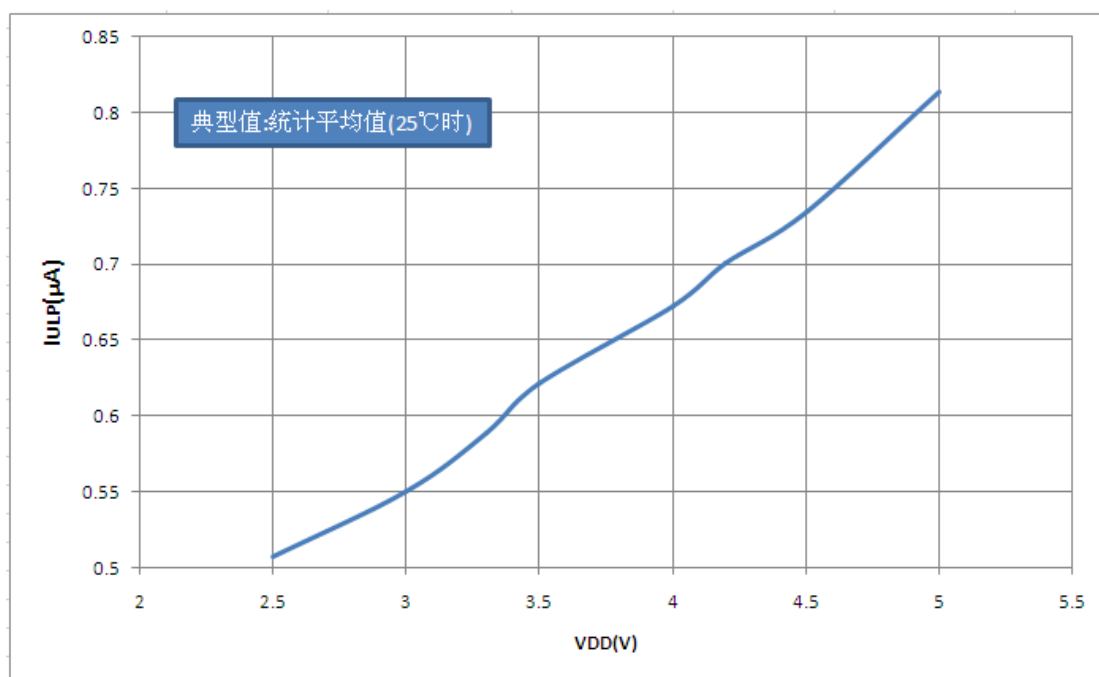
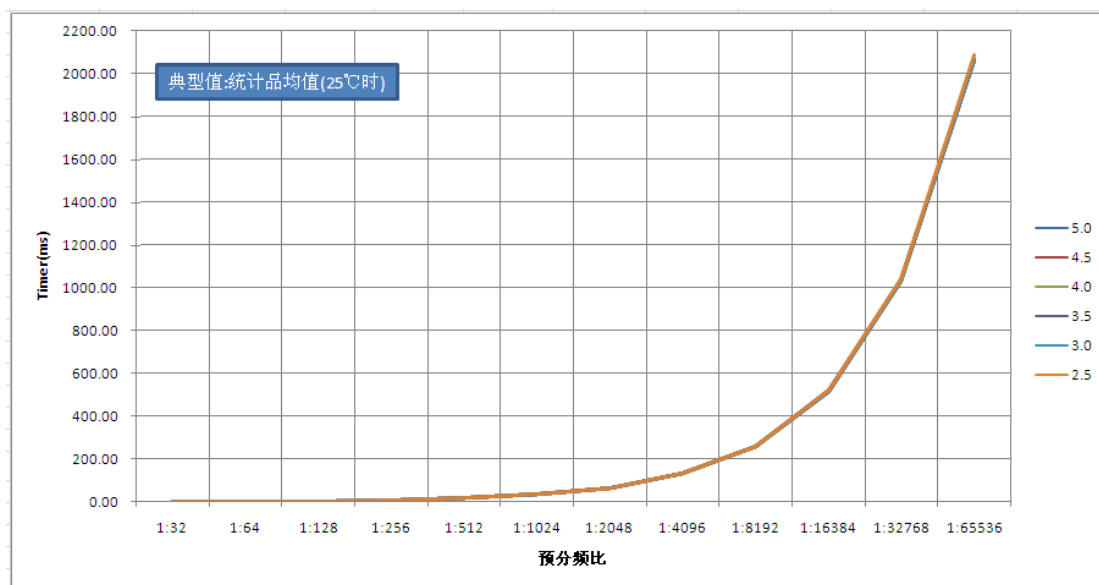
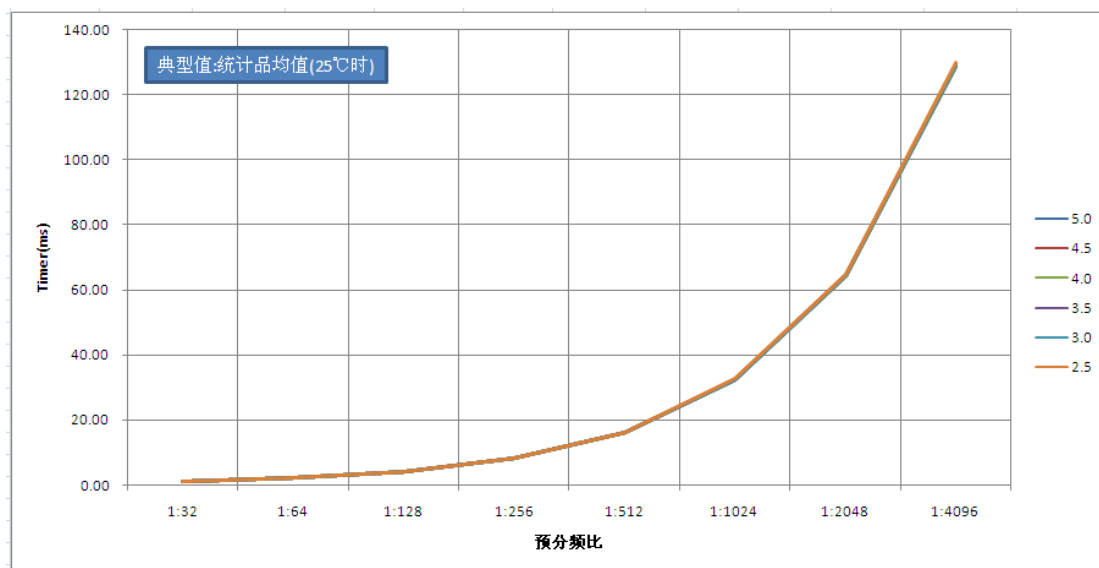


图22.5: 不同VDD时看门狗预分频比WDTPS<3:0> 一周期关系曲线图



备注:1/32 分频时看门狗周期为 1ms。

图22.6: 不同VDD时看门狗预分频比PS<2:0> 一周期关系曲线图



备注:1/32 分频时看门狗周期为 1ms。

图22.7:看门狗电流 I_{WDT} — V_{DD} 关系曲线图

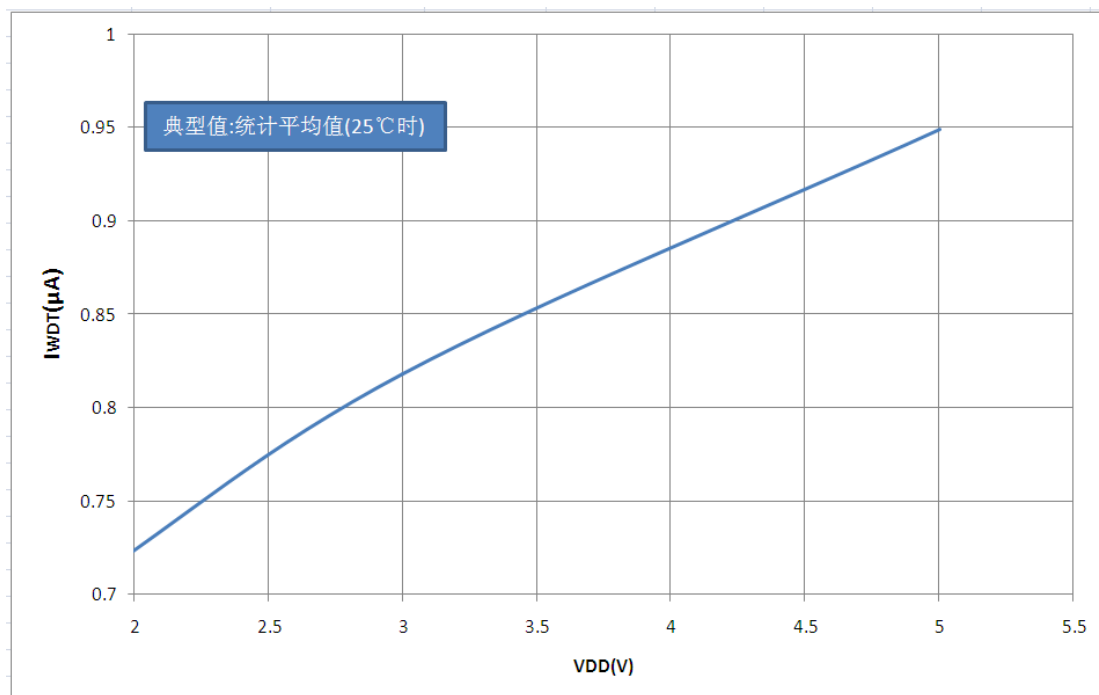


图22.8:比较器电流 I_{CMP} — V_{DD} 关系曲线图（使能一路比较器）

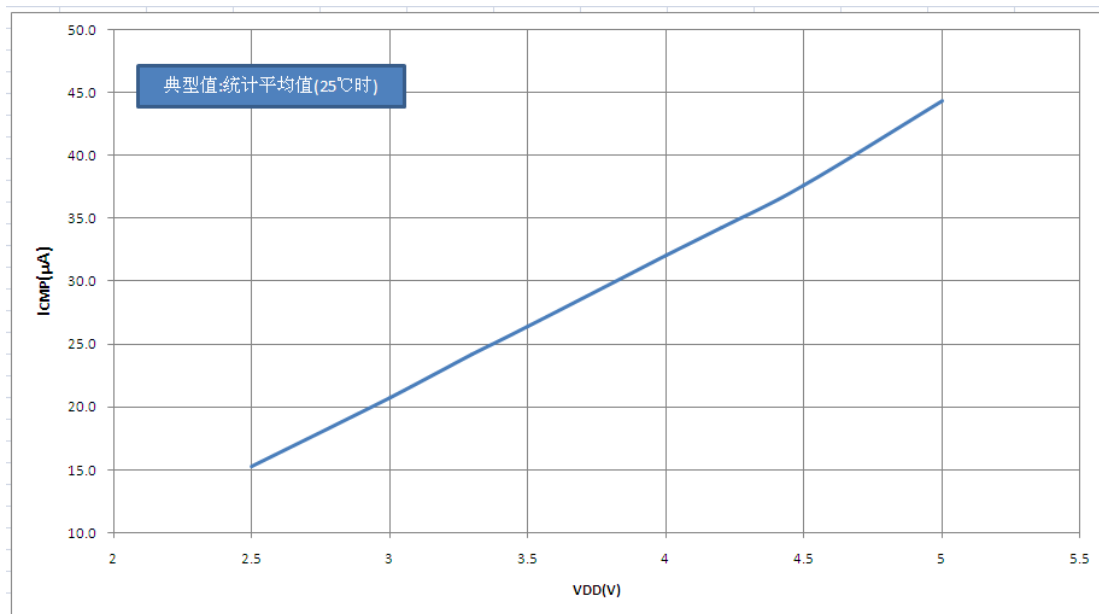


图22.9:运算放大器电流 I_{OP} — V_{DD} 关系曲线图（使能一路比较器）

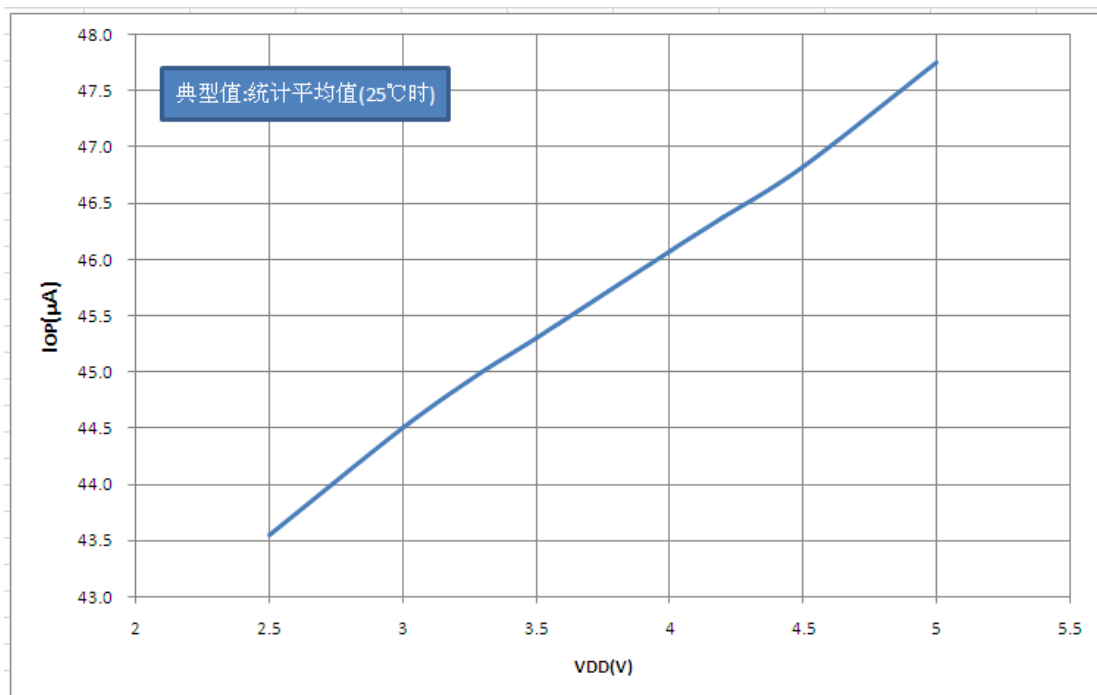


图 22.10:欠压复位电流 I_{LVR} — V_{DD} 关系曲线图

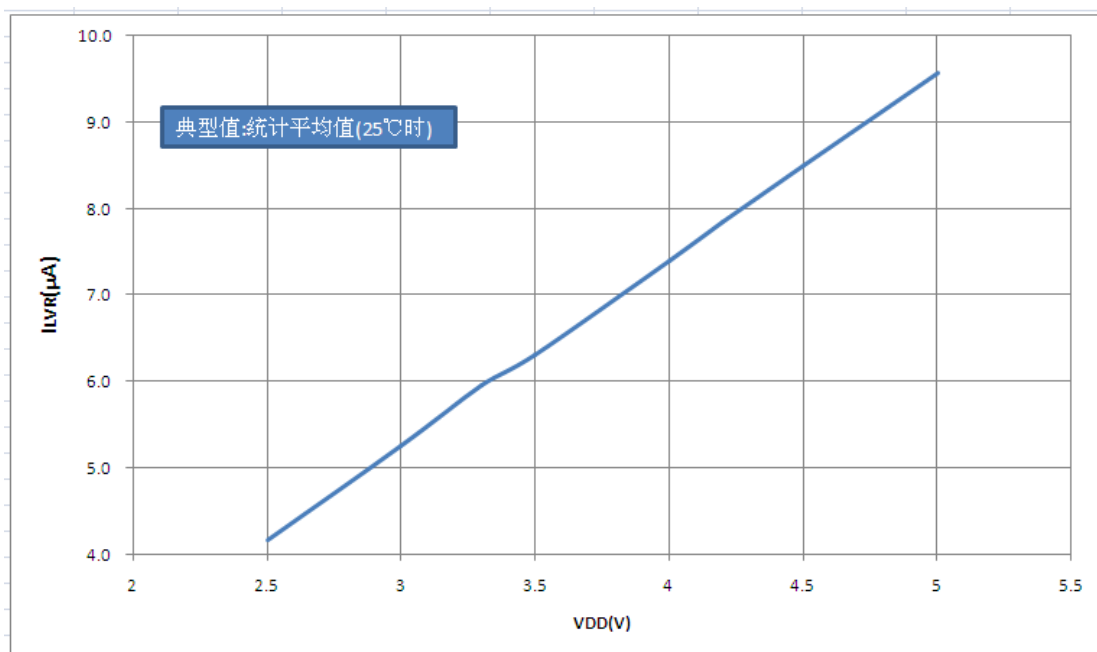


图 22.11: 欠压复位电流 I_{HLVD} — V_{DD} 关系曲线图

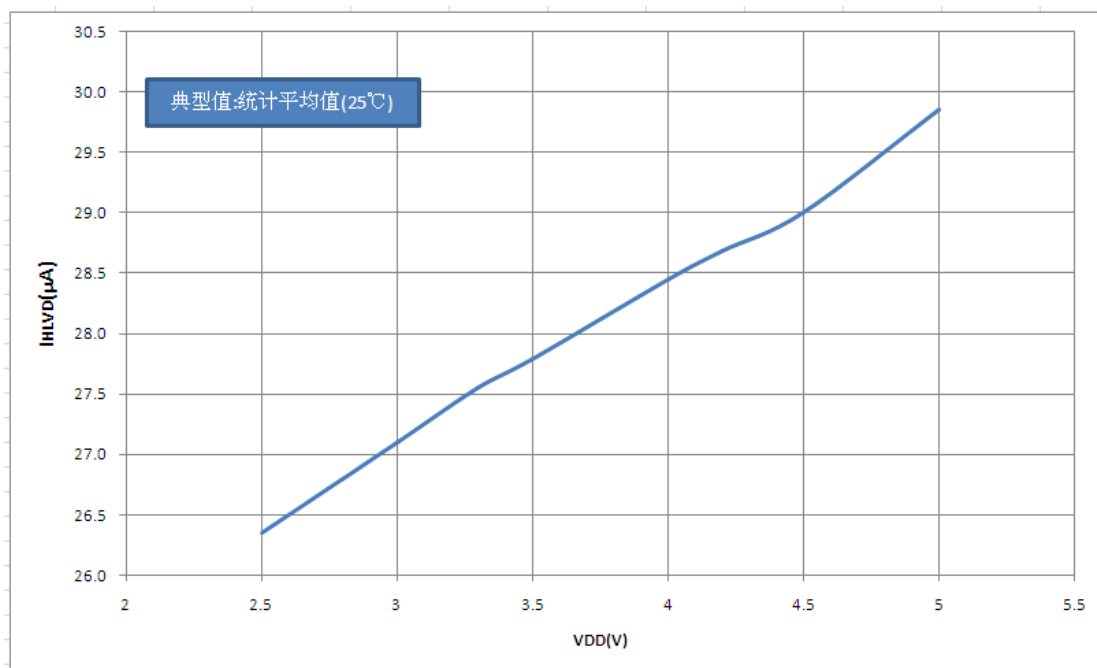


图 22.12: 欠压复位电流 I_{ADC} — V_{DD} 关系曲线图

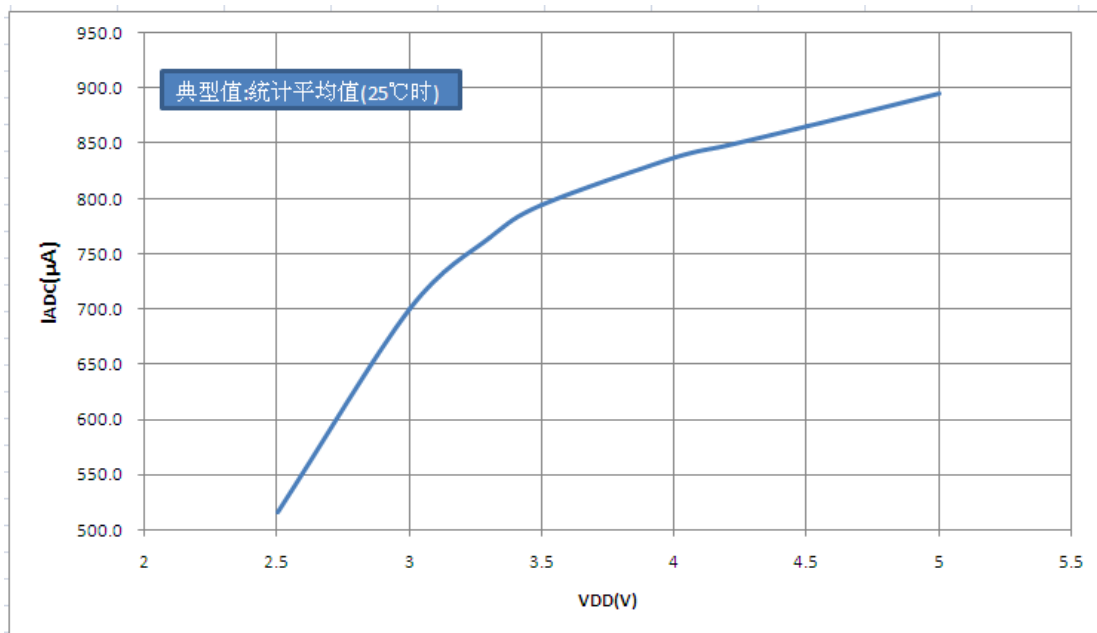


图 22.13: 欠压复位电流 I_{DAC} — V_{DD} 关系曲线图

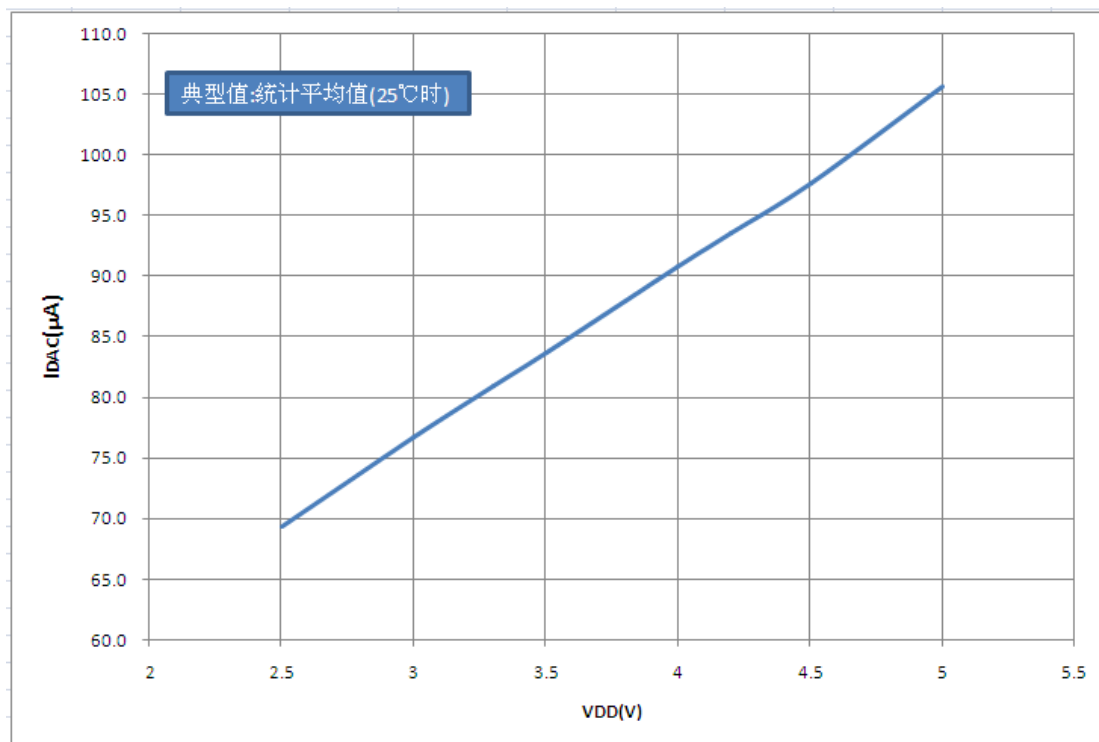


图 22.14: 欠压复位电流 I_{FVR} — V_{DD} 关系曲线图

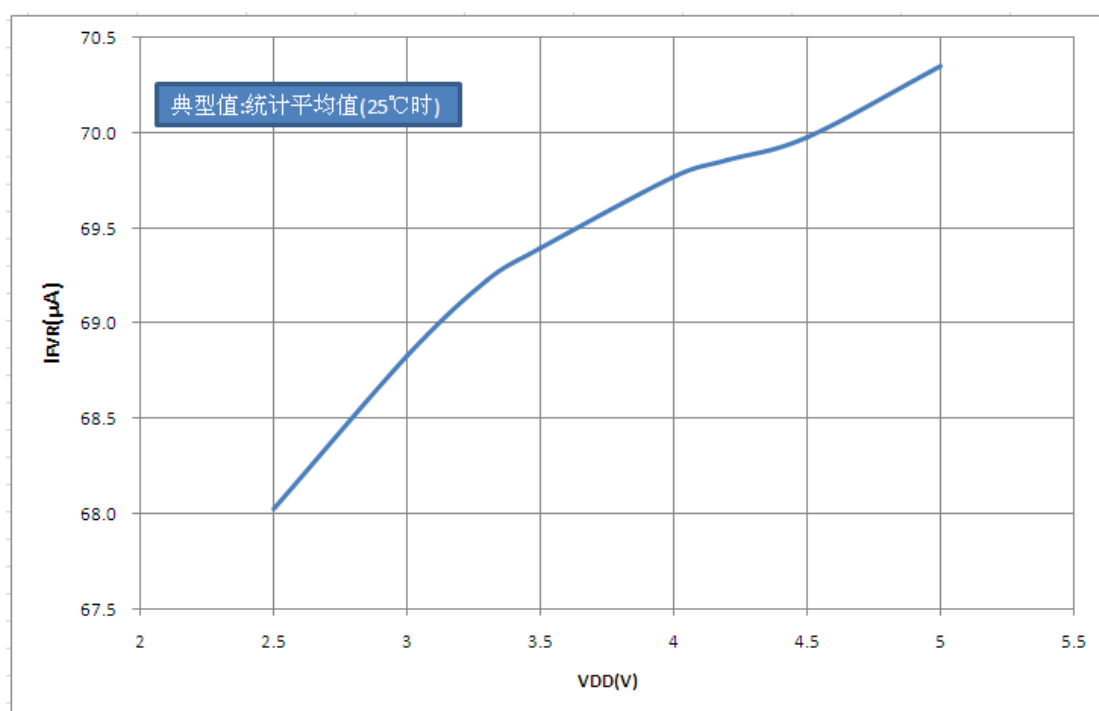


图22.15:不同温度时 V_{OH} — I_{OH} 关系曲线图 ($V_{DD} = 5.0V$)

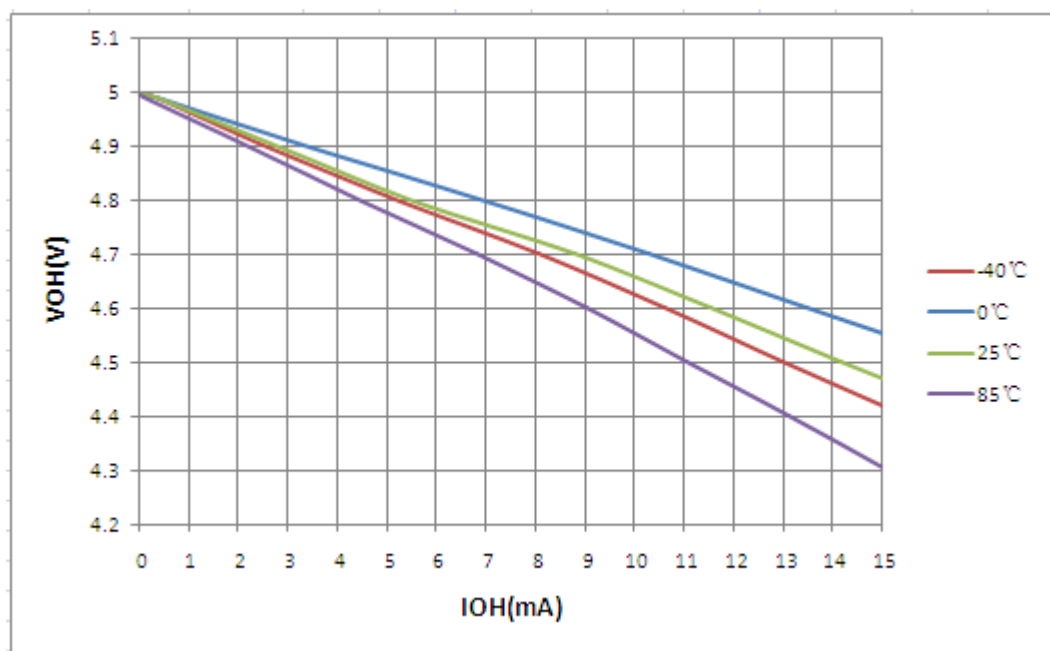


图22.16:不同温度时 V_{OH} — I_{OH} 关系曲线图 ($V_{DD} = 3.0V$)

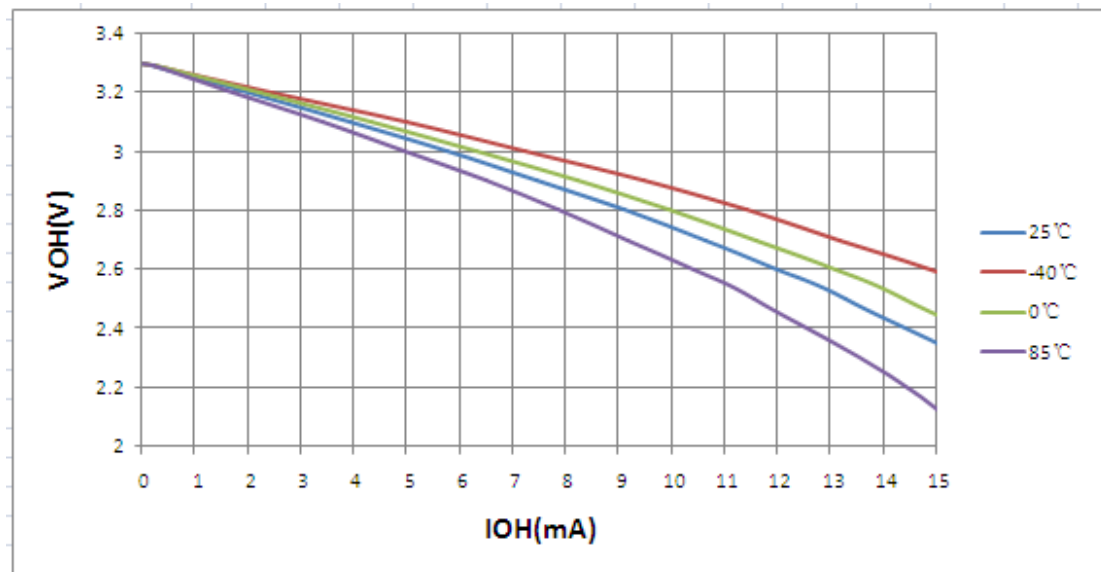


图22.17:不同温度时 V_{OH} — I_{OL} 关系曲线图 ($V_{DD} = 5.0V$)

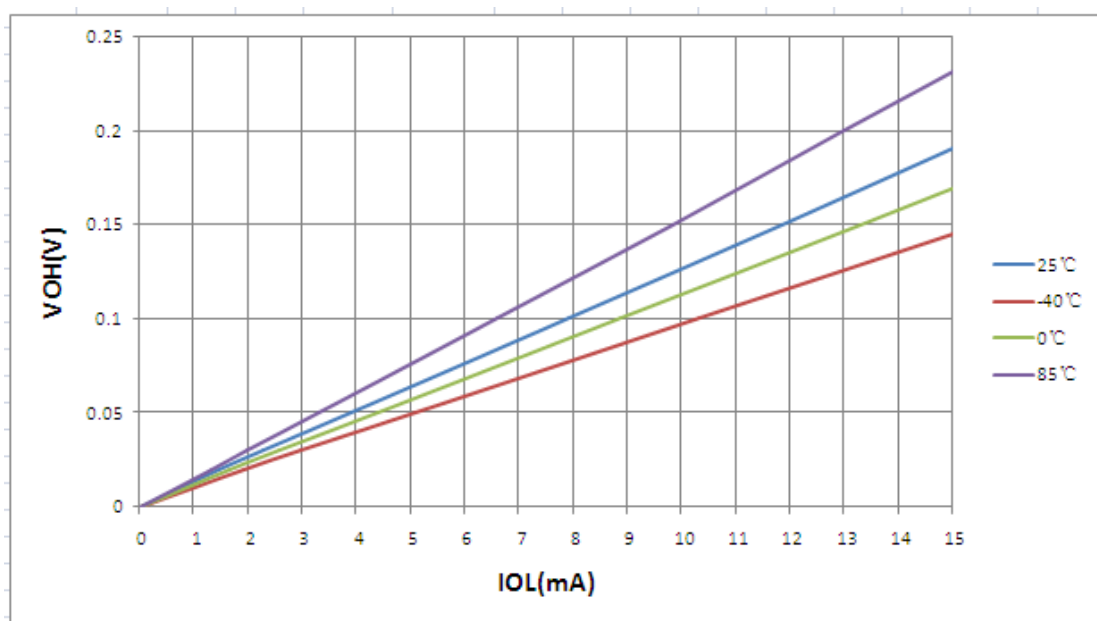


图22.18:不同温度时 V_{OH} — I_{OL} 关系曲线图 ($V_{DD} = 3.0V$)

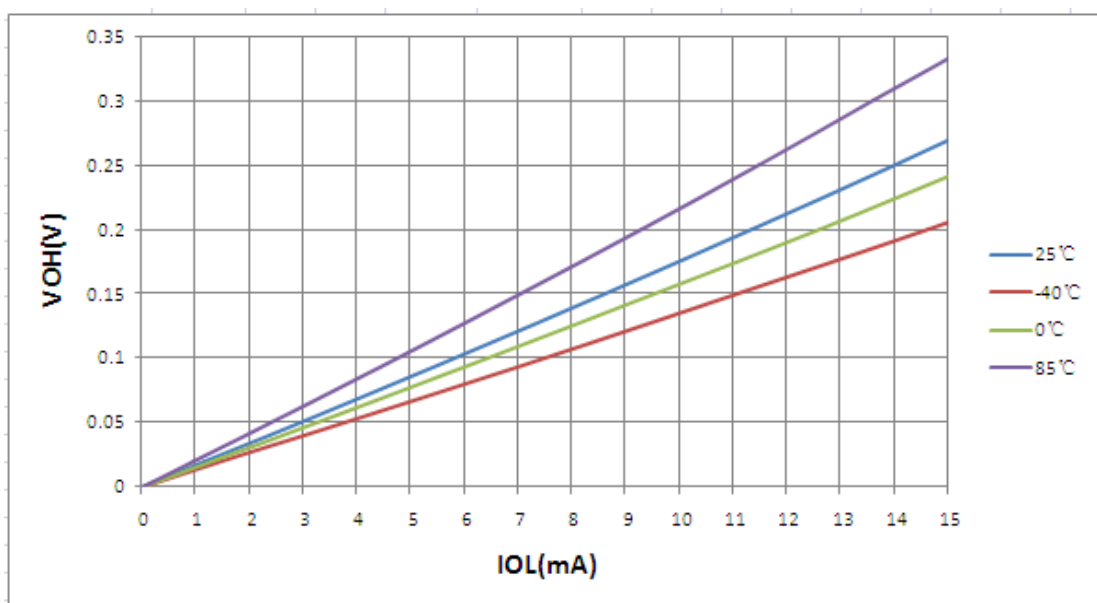


图22.19:上拉功能打开时端口电压 V_{OP} — 电流 I_{OP} 关系曲线图 ($V_{DD} = 5.0V$)

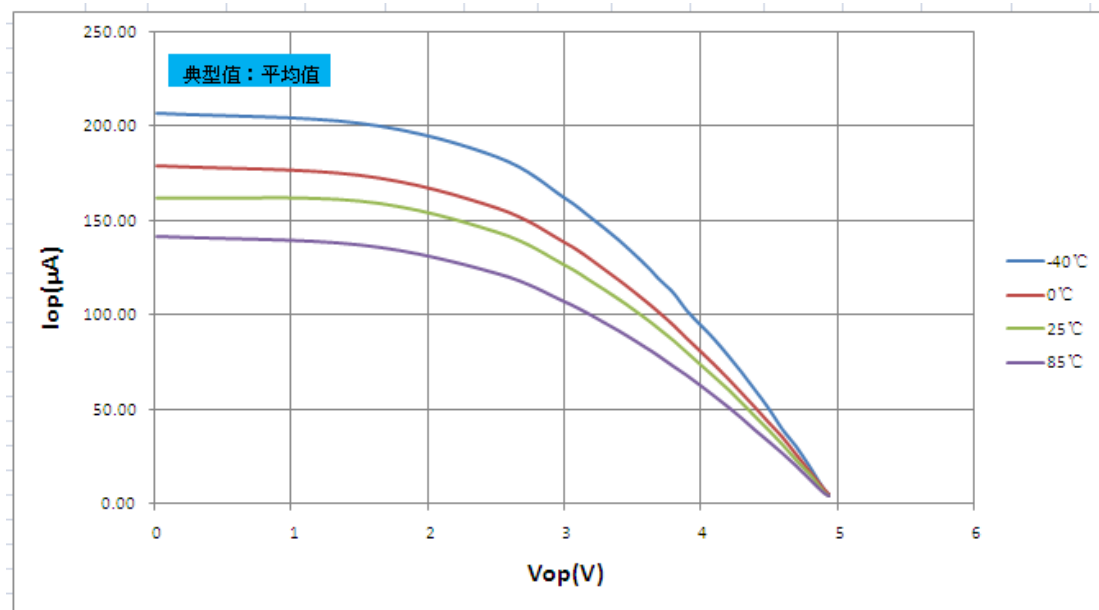
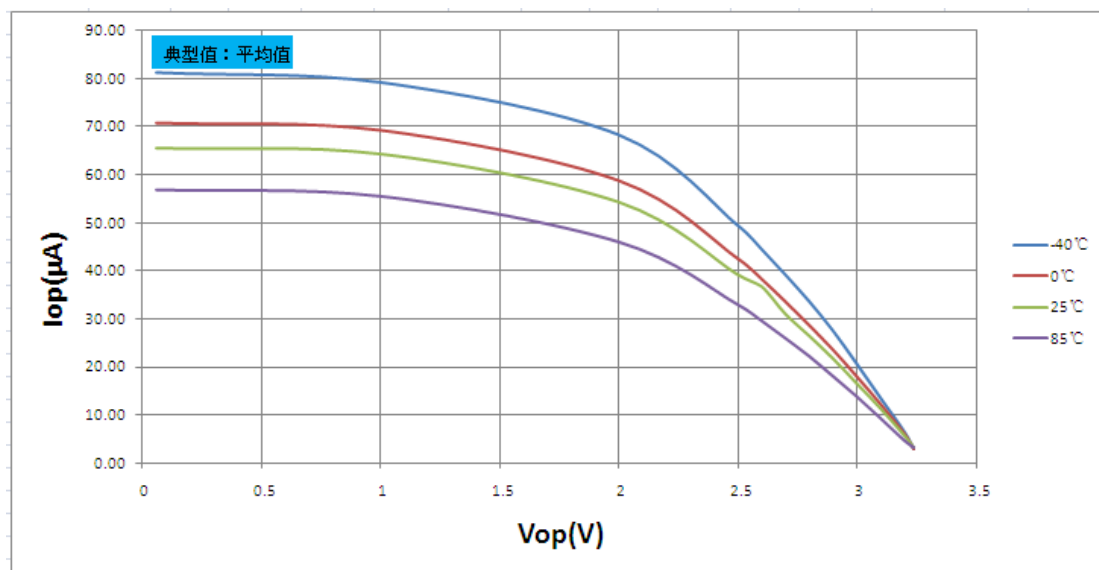
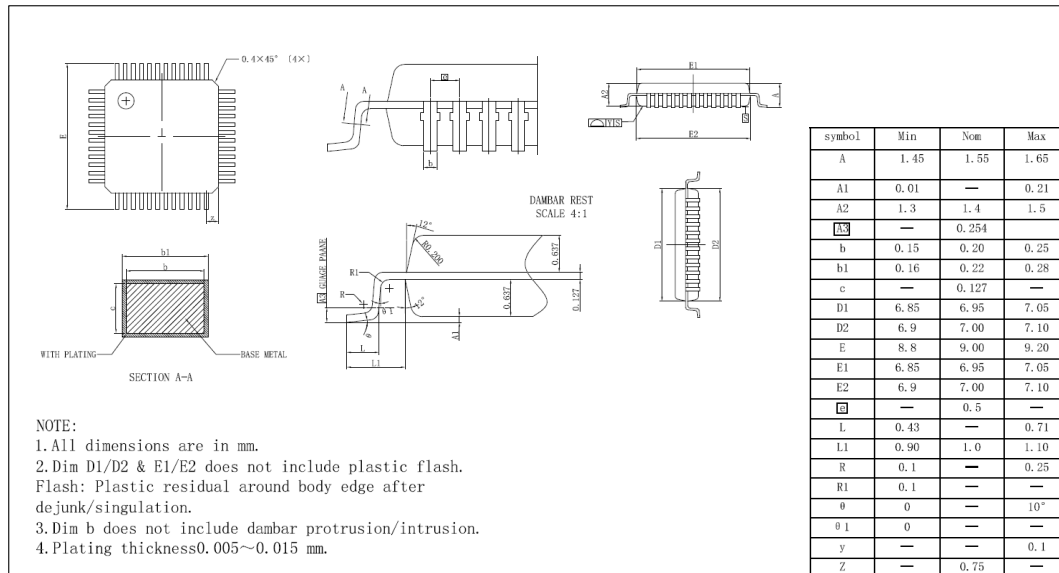


图22.20:上拉功能打开时端口电压 V_{OP} — 电流 I_{OP} 关系曲线图 ($V_{DD} = 3.3V$)

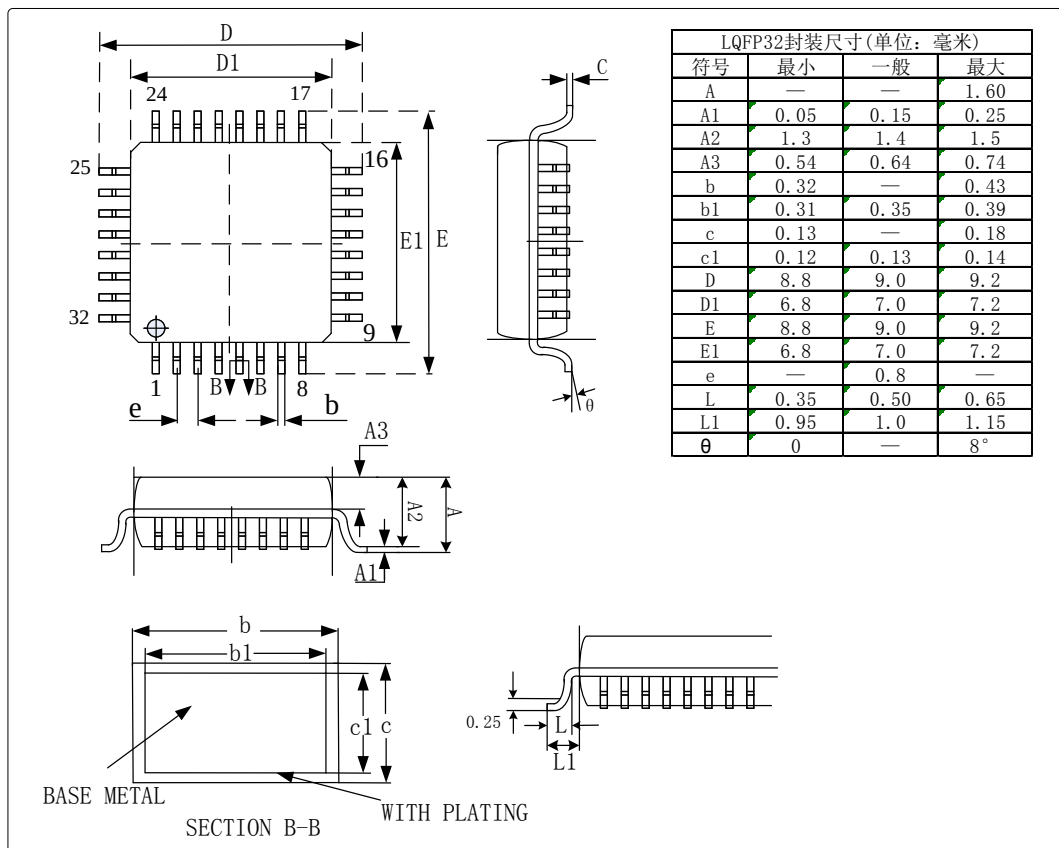


24 封装信息

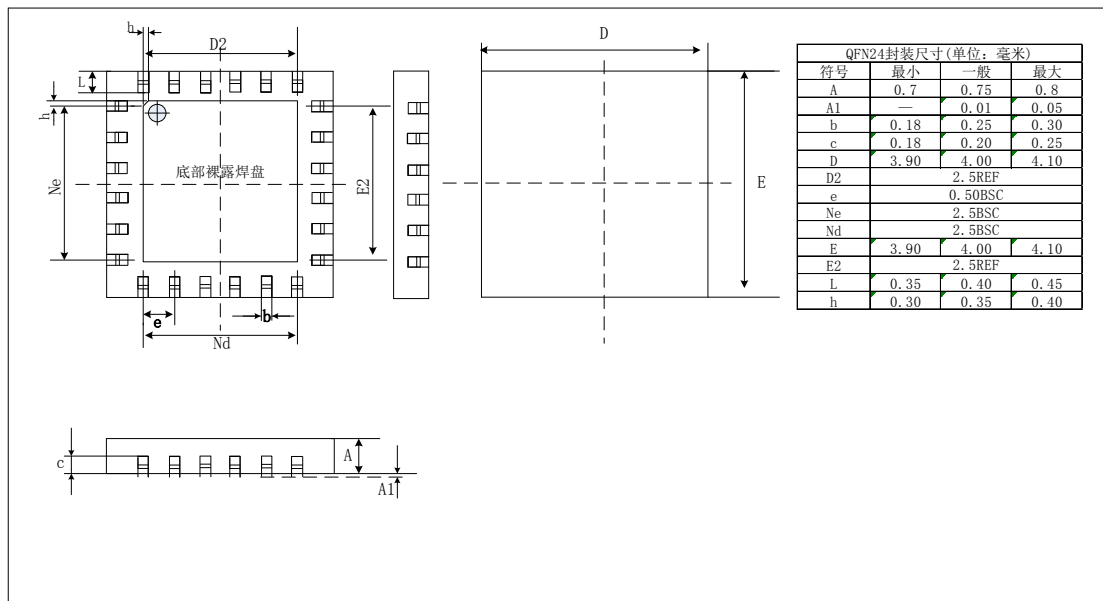
LQFP-48 封装:



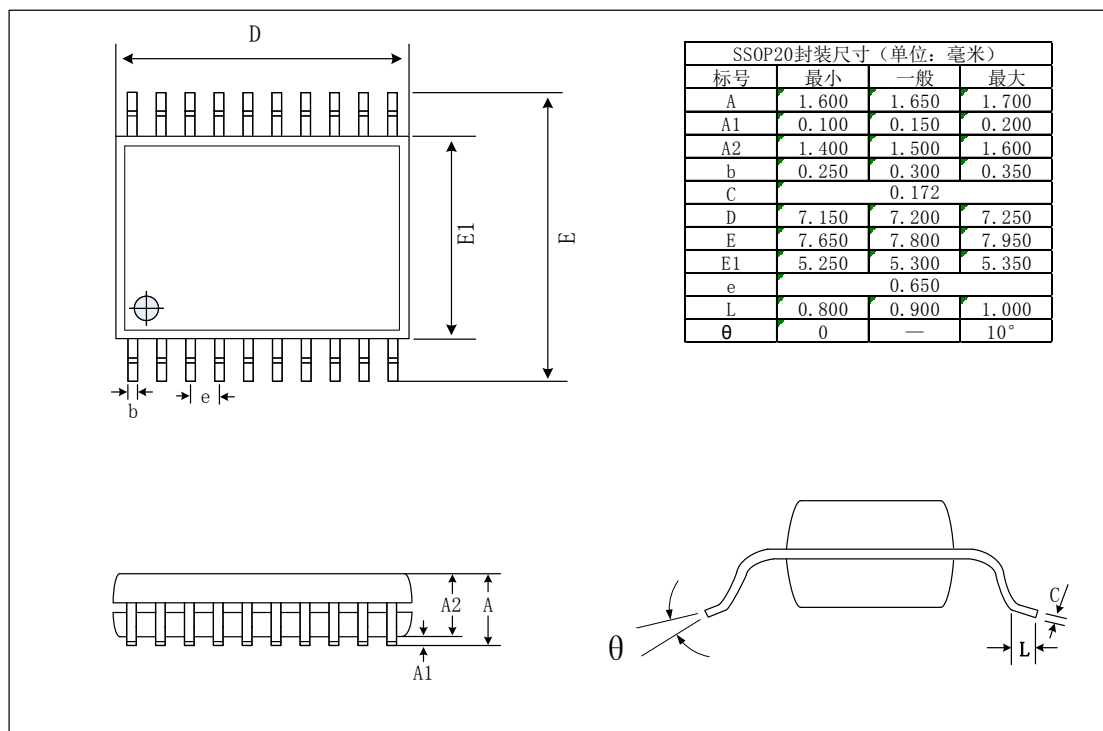
32 脚 LQFP 封装



24 脚 QFN 封装



20 脚 SSOP 封装



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	---1 1xxx
05H	P0	P07	P06	P05	P04	P03	P02	P01	P00	xxxx xxxx
06H	P2	P27	P26	P25	P24	P23	P22	P21	P20	xxxx xxxx
07H	P1	P17	P16	P15	P14	P13	P12	P11	P10	xxxx xxxx
08H	P3	P37	P36	P35	P34	P33	P32	P31	P30	xxxx xxxx
0AH	PCH	-	-	-	程序计数器(PC)高字节					---0 0000
0BH	INTCTL	AIE/AIEH	PUIE/AIEL	T0IE	-	P0IE	T0IF	-	P0IF	0000 0000
0CH	EIF1	EEIF	ADIF	-	INT1IF	C1IF	PWMIF	T2IF	T1IF	0000 0000
0DH	EIF2	T3IF	-	RC1IF	TX1IF	C3IF	-	BCLIF	SSC1IF	0-00 -000
0EH	T1L	定时/计数器 T1 低字节寄存器								xxxx xxxx
0FH	T1H	定时/计数器 T1 高字节寄存器								xxxx xxxx
10H	T1CTL	T1RLD	T1GC	T1CKS1	T1CKS0	T1OSCEN	$\overline{\text{T1SY}}$	T1CS	T1ON	0000 0000
11H	T2L	定时器 2(T2)低字节寄存器								0000 0000
12H	T2CTL0	-	T2CKBS3	T2CKBS2	T2CKBS1	T2CKBS0	T2ON	T2CKPS1	T2CKPS0	0000 0000
13H	CCP3CTL				CCP3OE	CCP3MOD3	CCP3MOD2	CCP3MOD1	CCP3MOD0	0000 0000
14H	CCP3H	CCP3 寄存器高 8 位								0000 0000
15H	CCP3L	CCP3 寄存器低 8 位								0000 0000
16H	CCTCTL	CCTEN	CSEL1	CSEL0	CRSET	CFSET	CCT4	CCT3	CCT1	0000 0000
17H	BANK	-	-	-	-	PR3	PR2	PR1	PR0	---- 0000
18H	ADSCANCTL	ADSCANEN	SMOD1	SMOD0	STIM4	STIM3	STIM2	STIM1	STIM0	0000 0000
19H	C1CTL	C1EN	C1OE	C1P1	C1P0	-	-	C1M1	C1M0	0000 0000
1AH	COUT	-	VCEN	-	-	C4OUT	C3OUT	C2OUT	C1OUT	0000 0000
1BH	AMPCCTL	AMPCALD	-	INPS1	INPS0	-	-	AMPCALEN	AMPON	1000 0000
1DH	ANS1	ANS17	ANS16	ANS15	ANS14	ANS13	ANS12	ANS11	ANS10	1111 1111
1EH	ADCDATA0H	ADC 数据寄存器 0 高字节								xxxx xxxx
1FH	ADCCTL0	ADLR	T2CCR0ON	-	-	-	-	START	ADEN	00-- --00
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	-	PP0	---- -000
23H	IP1	PEE	PADC	-	PINT1	PC1	PPWM	PT2	PT1	0000 0000
24H	IP2	PT3	-	PRC1	PTX1	PC3	-	PBCL	PSSCI	0-00 -000
25H	TR0	TR07	TR06	TR05	TR04	TR03	TR02	TR01	TR00	1111 1111
26H	TR2	TR27	TR26	TR25	TR24	TR23	TR22	TR21	TR20	1111 1111
27H	TR1	TR17	TR16	TR15	TR14	TR13	TR12	TR11	TR10	1111 1111
28H	OSCASTA	-	OSTS	HTS	LTS	-	-	SCF1	SCF0	-110 -00
29H	IP3	PT4	PC4	POSCFAIL	-	-	-	PP3	PHLVD	0000 0000
2AH	OSCCAL2	内部高频晶振校准寄存器 2								1000 0000
2BH	VREFCTL	VREFSEL1	VREFSEL0	VREFCALEN	VREFCLKEN	VREFOE	P18OE	VREFEN	-	0000 0000
2CH	EIE1	EEIE	ADIE	-	INT1IE	C1IE	PWMIE	T2IE	T1IE	0000 0000
2DH	EIE2	T3IE	-	RC1IE	TX1IE	C3IE	-	BCLIE	SSC1IE	0000 0000
2EH	PCTL	-	-	-	SLVREN	IPEN	SWDTEN	$\overline{\text{POR}}$	$\overline{\text{LVR}}$	--01 00xx
2FH	OSCCTL	CLKOE	IRCS2	IRCS1	IRCS0	SCS1	SCS0	IESO	FSCM	0010 0000
31H	ANS0	ANS07	ANS06	ANS05	ANS04	ANS03	ANS02	ANS01	ANS00	1111 1111
32H	ANS2	ANS27	ANS26	ANS25	ANS24	ANS23	ANS22	ANS21	ANS20	1111 1111
33H	ANS3	ANS37	ANS36	ANS35	ANS34	ANS33	ANS32	ANS31	ANS30	1111 1111
34H	OSC0CAL2	-	-	-	-	-	TCAL<2:0>			0000 0101
35H	PUR0	PUR07	PUR06	PUR05	PUR04	PUR03	PUR02	PUR01	PUR00	1111 1111
36H	IOCL0	IOCL07	IOCL06	IOCL05	IOCL04	IOCL03	IOCL02	IOCL01	IOCL00	0000 0000
37H	OSCCAL1	内部高频晶振校准寄存器 1								---- -100
38H	NVMDATAH	NVM 数据高 8 位寄存器								0000 0000
39H	NVMDATAL	NVM 数据低 8 位寄存器								0000 0000
3AH	NVMADDRH	NVM 地址指针高 8 位寄存器								0000 0000
3BH	NVMADDRL	NVM 地址指针低 8 位寄存器								0000 0000
3CH	NVMCTL0	NVM 控制寄存器 0								---- ----
3DH	NVMCTL1	NVM 控制寄存器 1								---- ----



KF8L15Z20XX 数据手册 V0. 9

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	复位初值
3EH	ADCDATA0L	ADC 数据寄存器 0 低字节								xxxx xxxx
3FH	ADCCTL1	ADCALEN	ADCS2	ADCS1	ADCS0	VCFG1	VCFG0	-	-	0000 00-0
40H	T2CCR0H	T2 触发 ADC 启动设置寄存器 0 高 8 位								0000 0000
41H	T2H	定时器 2(T2)高字节寄存器								xxxx xxxx
42H	PP5H	PWM5 周期寄存器高 8 位								xxxx xxxx
43H	PWM5H0	PWM5 通道 1 占空比寄存器高 8 位								xxxx xxxx
44H	PWM5H1	PWM5 通道 1 占空比缓冲寄存器高 8 位（与用户无关）								xxxx xxxx
45H	P0LR	P0LR7	P0LR6	P0LR5	P0LR4	P0LR3	P0LR2	P0LR1	P0LR0	xxxx xxxx
46H	P2LR	P2LR7	P2LR6	P2LR5	P2LR4	P2LR3	P2LR2	P2LR1	P2LR0	xxxx xxxx
47H	P1LR	P1LR7	P1LR6	P1LR5	P1LR4	P1LR3	P1LR2	P1LR1	P1LR0	xxxx xxxx
48H	P3LR	P3LR7	P3LR6	P3LR5	P3LR4	P3LR3	P3LR2	P3LR1	P3LR0	xxxx xxxx
49H	TR3	TR37	TR36	TR35	TR34	TR33	TR32	TR31	TR30	1111 1111
4AH	EIE3	T4IE	C4IE	OSCFAILIE	-	-	-	P3IE	HLVDIE	0000 0000
4BH	EIF3	T4IF	C4IF	OSCFAILIF	-	-	-	P3IF	HLVDIF	0000 0000
4CH	OSCCAL3	内部高频晶振校准寄存器 3								1000 1000
4DH	OSCCAL0	内部高频晶振校准寄存器 0								0001 0000
4EH	T3CTL	T3REN	-	T3CKS1	T3CKS0	-	T3CS1	T3CS0	T3ON	0000 0000
4FH	T3L	T3 低位寄存器								0000 0000
50H	ADCDATA1H	ADC 数据寄存器 1 高字节								0000 0000
51H	ADCDATA1L	ADC 数据寄存器 1 低字节								0000 0000
52H	PP5L	PWM5 周期寄存器低 8 位								1111 1111
53H	PINSET	-	INT1PIN	-	-	-	-	-	SSCIPIN	0000 0000
54H	T2CCR0L	T2 触发 ADC 启动设置寄存器 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 通道 1 占空比设置寄存器低 8 位								xxxx xxxx
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 通道 1 低 8 位占空比缓冲寄存器（与用户无关）								xxxx xxxx
57H	PWM5CTL0	P5CH1MOD1	P5CH1MOD0	P5CH3MOD	P5CH2MOD	P5MOD3	P5MOD2	P5MOD1	P5MOD0	0000 0000
58H	ADCINTCTL	T2CCR1ON	-	-	-	-	-	INTCTL1	INTCTL0	0--- --00
59H	ADCDATA2H	ADC 数据寄存器 2 高字节								0000 0000
5AH	ADCDATA2L	ADC 数据寄存器 2 低字节								0000 0000
5BH	PWM5CTL1	P5RSEN	P5DC6	P5DC5	P5DC4	P5DC3	P5DC2	P5DC1	P5DC0	0000 0000
5CH	P5ASCTL0	P5ASE	P5ASS2	P5ASS1	P5ASS0	P5SSAC1	P5SSAC0	P5SSBD1	P5SSBD0	0000 0000
5DH	PSTRCTL0	-	-	-	STRSYNC	STREND	STRENC	STRENB	STRENA	--0 0001
5EH	ADCDATA3H	ADC 数据寄存器 3 高字节								0000 0000
5FH	T3H	T3 高位寄存器								0000 0000
60H	PUR1	PUR17	PUR16	PUR15	PUR14	PUR13	PUR12	PUR11	PUR10	0000 0000
61H	PUR2	PUR27	PUR26	PUR25	PUR24	PUR23	PUR22	PUR21	PUR20	0000 0000
62H	TMRBUZ	-	-	-	-	-	-	T1H_BUZEN	T1L_BUZEN	---- --00
63H	ADCCCTL2	-	ADCHS6	ADCHS5	ADCHS4	ADCHS3	ADCHS2	ADCHS1	ADCHS0	---- 0000
64H	DAC12CTL	DAC12 EN	BUF CALI	DAC12 OE	-	-	-	DAC12 RFS1	DAC12 RFS0	0000 0000
65H	DAC12DH	-	-	-	-	DAC12 D11	DAC12 D10	DAC12 D9	DAC12 D8	0000 0000
66H	DAC12DL	DAC12 D7	DAC12 D6	DAC12 D5	DAC12 D4	DAC12 D3	DAC12 D2	DAC12 D1	DAC12 D0	0000 0000
67H	INTEDGCTL	-	INT1SE	-	-	-	-	-	T1CLKSE	00-- ---1
6AH	ADCDATA3L	ADC 数据寄存器 3 低字节								0000 0000
6BH	IOCL3	IOCL37	IOCL36	IOCL35	IOCL34	IOCL33	IOCL32	IOCL31	IOCL30	0000 0000
6CH	PUR3	PUR37	PUR36	PUR35	PUR34	PUR33	PUR32	PUR31	PUR30	0000 0000
105H	CCP4L	CCP4 寄存器低 8 位								0000 0000
106H	CCP4H	CCP4 寄存器高 8 位								0000 0000
107H	CMCTL0	C4IMS	C3IMS	C2IMS	C1IMS	C4EDG	C3EDG	C2EDG	C1EDG	0000 0000
108H	DIVRH	余数高 8 位寄存器								0000 0000
109H	CCP4CTL	-	-	-	CCP4OE	CCP4MOD3	CCP4MOD 2	CCP4MOD1	CCP4MOD0	---0 0000
10CH	PWM5L2	PWM5 通道 2 占空比寄存器低 8 位								0000 0000
10DH	PWM5L3	PWM5 通道 3 占空比寄存器低 8 位								0000 0000
10EH	PWM5OC	-	-	OCA3	OCB3	OCA2	OCB2	OCA	OCB	--00 0000
10FH	T2CTL1	T2MOD1	T2MOD0	T2OFS1	T2OFS0	T2TRIGEN	-	-	T2DIR	0000 0--0
110H	C4FILTCTL	INV4	WEN4	FEN4	-	-	F4CNT2	F4CNT1	F4CNT0	000- -000
111H	C4FILTPRE	FP47	FP46	FP45	FP44	FP43	FP42	FP41	FP40	0000 0000
112H	C2CTL	C2EN	-	C2P1	C2P0	-	-	C2M1	C2M0	0000 0000
113H	C3CTL	C3EN	-	C3P1	C3P0	-	-	C3M1	C3M0	0000 0000
114H	C4CTL	C4EN	-	C4P1	C4P0	-	-	C4M1	C4M0	0000 0000
115H	DIVBH	除数高 8 位寄存器								0000 0000
116H	T2CCR1L	T2 触发 ADC 启动设置寄存器 1 低 8 位								0000 0000



KF8L15Z20XX 数据手册 V0. 9

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	复位初值
118H	T2CCR1H	T2 触发 ADC 启动设置寄存器 1 高 8 位								0000 0000
119H	PWM5FC	-	-	FCA3	FCB3	FCA2	FCB2	FCA	FCB	--00 0000
11AH	P5ASCTL1	P5SSA31	P5SSA30	P5SSB31	P5SSB30	P5SSA21	P5SSA20	P5SSB21	P5SSB20	0000 0000
11BH	PSTRCTL1	-	-	-	-	STRENA3	STRENB3	STRENA2	STRENB2	---- 1010
11CH	PWM5CTL2	-	-	PFUSES	-	-	UDEVT1	UDEVT0	UDEN	--0- -000
11DH	PWM5PC	-	-	PCA3	PCB3	PCA2	PCB2	PCA	PCB	--00 0000
11EH	PWM5H2	PWM5 通道 2 占空比寄存器高 8 位								0000 0000
11FH	PWM5H3	PWM5 通道 3 占空比寄存器高 8 位								0000 0000
120H	RSCTL1	SPEN1	R1X9	SRXEN1	CRXEN1	ADREN1	FRER1	OVFER1	RX9D1	0000 000x
121H	TXSDR1	USART 发送数据寄存器 1								0000 0000
122H	RXSDR1	USART 接收数据寄存器 1								0000 0000
123H	BRCTL1	ABRDOVF1	RCIDLF1	SCKPS1	B1RG16	BR1CKS1	BR1CKS0	WUEN1	ABRDEN1	0100 0000
124H	TSCTL1	CSRS1	T1X9	TXEN1	SYNC1	SENDER1	HBRG1	TXSRS1	TX9D1	0000 0010
125H	EUBRGL1	USART1 波特率数据寄存器低字节								0000 0000
126H	EUBRGH1	USART1 波特率数据寄存器高字节								0000 0000
127H	USLPEN1	SLPEN1			-	-	-	-	-	0--- ----
128H	SSICCTL0	SSCIWCFL	SSCIOV	SSCIEN	SSCICKP	SSCIMOD3	SSCIMOD2	SSCIMOD1	SSCIMOD0	0000 0000
12AH	SSICCTL1	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								0000 0000
12DH	UARTMTH1	UART1 地址匹配设置寄存器								0000 0000
12EH	SSCIADD	SSCI 的 I2C 地址寄存器								1111 1111
	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
138H	MULRES0	乘法运算结果寄存器 0								0000 0000
139H	DIVCTL	-	-	-	-	-	-	DIVOEN	DIVEN	0000 0000
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	DIVBL	除数低 8 位寄存器								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	DIVRL	余数低 8 位寄存器								0000 0000
14AH	C1FILTCTL	INV1	WEN1	FEN1	-	-	F1CNT2	F1CNT1	F1CNT0	000- -000
14BH	C1FILTPRE	FP17	FP16	FP15	FP14	FP13	FP12	FP11	FP10	0000 0000
14EH	C3FILTCTL	INV3	WEN3	FEN3	-	-	F3CNT2	F3CNT1	F3CNT0	000- -000
14FH	C3FILTPRE	FP37	FP36	FP35	FP34	FP33	FP32	FP31	FP30	0000 0000
150H	UPINSET1	USLM1	UPSEL1	-	-	-	-	-	-	0000 0000
159H	VREFCAL0	内部参考电压校准寄存器 0								0000 0000
15AH	VREFCAL1	内部参考电压校准寄存器 1								1010 0000
15BH	HLVDCTL	HLVDEN	VDIR	-	-	-	VDT2	VDT1	VDT0	0000 0000
15DH	T3REL	T3 重载设置寄存器低 8 位								0000 0000
15EH	T3REH	T3 重载设置寄存器高 8 位								0000 0000
160H	T4L	T4 计数器低 8 位								0000 0000
161H	T4H	T4 计数器高 8 位								0000 0000
162H	T4REL	T4 重载设置寄存器低 8 位								0000 0000
163H	T4REH	T4 重载设置寄存器高 8 位								0000 0000
164H	T4CTL	T4REN	T4BUZOE	T4CKS1	T4CKS0	-	T4CS1	T4CS0	T4ON	0000 0000
165H	RC32KCAL	内部 32K 振荡器频率校准寄存器								0000 0000
206H	ANS4	ANS47	ANS46	ANS45	ANS44	ANS43	ANS42	ANS41	ANS40	1111 1111
207H	ANS5	-	ANS56	ANS55	ANS54	ANS53	ANS52	ANS51	ANS50	1111 1111
208H	ANS6	-	ANS66	ANS65	ANS64	ANS63	ANS62	ANS61	ANS60	1111 1111
20CH	P4	P47	P46	P45	P44	P43	P42	P41	P40	xxxx xxxx
20DH	P4LR	P4LR7	P4LR6	P4LR5	P4LR4	P4LR3	P4LR2	P4LR1	P4LR0	xxxx xxxx
20EH	TR4	TR47	TR46	TR45	TR44	TR43	TR42	TR41	TR40	1111 1111
20FH	PUR4	PUR47	PUR46	PUR45	PUR44	PUR43	PUR42	PUR41	PUR40	1111 1111
210H	P5	-	P56	P55	P54	P53	P52	P51	P50	-xxx xxxx
211H	P5LR	-	P5LR6	P5LR5	P5LR4	P5LR3	P5LR2	P5LR1	P5LR0	-xxx xxxx
212H	TR5	-	TR56	TR55	TR54	TR53	TR52	TR51	TR50	-111 1111
213H	PUR5	-	PUR56	PUR55	PUR54	PUR53	PUR52	PUR51	PUR50	1111 1111



KF8L15Z20XX 数据手册 V0. 9

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	复位初值
214H	P6	-	P66	P65	P64	P63	P62	P61	P60	-xxx xxxx
215H	P6LR	-	P6LR6	P6LR5	P6LR4	P6LR3	P6LR2	P6LR1	P6LR0	-xxx xxxx
216H	TR6	-	TR66	TR65	TR64	TR63	TR62	TR61	TR60	-111 1111
218H	PUR6	-	PUR66	PUR65	PUR64	PUR63	PUR62	PUR61	PUR60	1111 1111
21EH	PWMCTL0	PWM17ON	PWM16ON	PWM15ON	PWM14ON	PWM13ON	PWM12ON	PWM11ON	PWM10ON	0000 0000
21FH	PP1	PWM1x 周期设置低 8 位寄存器								1111 1111
220H	PP2	PWM1x 周期设置高 8 位寄存器								1111 1111
235H	PWM1AL	PWM1A 占空比低 8 位寄存器								xxxx xxxx
236H	TEMPSNR	-	-	-	-	-	-	-	TSEN	---- --00
23FH	UPINSET2	USLM2	UPSEL2	-	-	-	-	-	-	0000 0000
240H	U7816CTL2	PSEL2	-	TREPEN2	TREP21	TREP20	RREPEN2	RREP21	RREP20	0-00 0000
241H	U7816 RXCTL2	ERSW21	ERSW20	RPAR2	RINV2	RCONV2	-	-	PAREF2	0000 0000
242H	U7816 TXCTL2	U7816EN2	CLKOUT2	TX9DSEL2	STOP2	TPAR2	TINV2	TCONV2	BGTEN2	0001 0000
243H	CLKDIV2	CLKDIV27	CLKDIV26	CLKDIV25	CLKDIV24	CLKDIV23	CLKDIV22	CLKDIV21	CLKDIV20	0000 0000
244H	EGTCTL2	EGT27	EGT26	EGT25	EGT24	EGT23	EGT22	EGT21	EGT20	0000 0000
247H	UARTMTH2	UART2 地址匹配设置寄存器								0000 0000
248H	USLPEN2	SLPEN2	-	-	-	-	-	-	-	0--- ----
249H	BRCTL2	ABRDOVF2	RCIDLF2	SCKPS2	B2RG16	BR2CKS1	BR2CKS0	WUEN2	ABRDEN2	0100 0000
24AH	RSTCTL2	SPEN2	R2X9	SRXEN2	CRXEN2	ADREN2	FRER2	OVFER2	RX9D2	0000 0000
24BH	RXSDR2	USART 接收数据寄存器 2								xxxx xxxx
24CH	EUBRGH2	USART2 波特率数据寄存器高字节								0000 0000
24DH	EUBRGL2	USART2 波特率数据寄存器低字节								0000 0000
24EH	TXSDR2	USART 发送数据寄存器 2								xxxx xxxx
24FH	TSCCTL2	CSRS2	T2X9	TXEN2	SYNC2	SENDB2	HBRG2	TXSR52	TX9D2	0000 0000
250H	BKPCCTL	BKP7	BKP6	BKP5	BKP4	BKP3	BKP2	BKP1	BKP0	0000 0000
251H	EIE4	-	-	-	-	RE2IE	TE2IE	RC2IE	TX2IE	---- 0000
252H	EIF4	-	-	-	-	RE2IF	TE2IF	RC2IF	TX2IF	---- 0000
253H	IP4	-	-	-	-	PRE2	PTE2	PRC2	PTX2	---- 0000
260H	PWM1AH	PWM1A 占空比高 8 位寄存器								xxxx xxxx
261H	PWM1BL	PWM1B 占空比低 8 位寄存器								xxxx xxxx
262H	PWM1BH	PWM1B 占空比高 8 位寄存器								xxxx xxxx
267H	PWM1EL	PWM1E 占空比低 8 位寄存器								xxxx xxxx
268H	PWM1EH	PWM1E 占空比高 8 位寄存器								xxxx xxxx
269H	PWM1FL	PWM1F 占空比低 8 位寄存器								xxxx xxxx
26AH	PWM1FH	PWM1F 占空比高 8 位寄存器								xxxx xxxx
26CH	PWMCTL1	PWM1FON	PWM1EON	PWM1DON	PWM1CON	PWM1BON	PWM1AON	PWM19ON	PWM18ON	0000 0000
318H	RTCSRT	SRT7	SRT6	SRT5	SRT4	SRT3	SRT2	SRT1	SRT0	0000 0000
319H	RTCSTU	RTCALREN	-	RESET	LIF	HT	CNF	RTOFF	RTCLD	0-00 0000
31CH	RTCALRS	ALRSN	ALRS6	ALRS5	ALRS4	ALRS3	ALRS2	ALRS1	ALRS0	-000 0000
31DH	RTCALRM	ALRMEN	ALRM6	ALRM5	ALRM4	ALRM3	ALRM2	ALRM1	ALRM0	-000 0000
31EH	RTCALRH	ALRHEN	-	ALRH5	ALRH4	ALRH3	ALRH2	ALRH1	ALRH0	--00 0000
31FH	RTCALRW	ALRWEN	-	-	-	-	ALRW2	ALRW1	ALRW0	---- -000
320H	RTCTTR	RTCOE	RTCOS	-	-	-	RTCTTR2	RTCTTR1	RTCTTR0	00-- -000
321H	RTCFGR	V7	V6	V5	V4	V3	V2	V1	V0	0000 0000
322H	RTCSEC	-	SEC6	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0	-000 0000
323H	RTCMIN	-	MIN6	MIN5	MIN4	MIN3	MIN2	MIN1	MIN0	-000 0000
324H	RTCHOUR	-	AMPM	HUR5	HUR4	HUR3	HUR2	HUR1	HUR0	-000 0000
325H	RTCWEK	-	-	-	-	-	WEK2	WEK1	WEK0	---- -000
326H	RTCDAY	-	-	DAY5	DAY4	DAY3	DAY2	DAY1	DAY0	--00 0000
327H	RTCMTH	-	-	-	MTH4	MTH3	MTH2	MTH1	MTH0	--0 0000
328H	RTCYEAR	YER7	YER6	YER5	YER4	YER3	YER2	YER1	YER0	0000 0000
329H	EIE5	ALRIE	RTCT1IE	RTCT0IE	TTIE	DAYIE	HURIE	MINIE	SECIE	0000 0000
32AH	EIF5	ALRIF	RTCT1IF	RTCT0IF	TTIF	DAYIF	HURIF	MINIF	SECIF	0000 0000
32BH	IP5	PALR	PRTCT1	PRTCT0	PTT	PDAY	PHUR	PMIN	PSEC	0000 0000
32CH	RTCTMRCTL	CKST13	CKST12	CKST11	CKST10	CKST03	CKST02	CKST01	CKST00	0000 0000
32DH	RTCTMRN	-	-	-	-	-	-	TMR1EN	TMR0EN	---- --00
32EH	RTCTMR1	RTC 定时器 1 初值配置寄存器								0000 0000
32FH	RTCTMR0	RTC 定时器 0 初值配置寄存器								0000 0000
360H	POWCTL	IOLATCH	VCORM	FULATCH	PHPDM	WKPEN	WKPF	DSL PEN	ULPEN	0000 0000
361H	PCAL	-	-	PCAL5	PCAL4	PCAL3	PCAL2	PCAL1	PCAL0	0000 0000
362H	XTALCAL	XTAL 校准寄存器								0101 0000
363H	BWDTCTL	BWDTEN	BPS3	BPS2	BPS1	BPS0	BAPS2	BAPS1	BAPS0	0000 0000
364H	PHCLR	VRTCS	VLCDS	BWDTCLR	REGCLR	PDRTC	PDLCD	EXTLFEN	INTLFEN	0000 0000
365H	LPRCCAL	内部低频晶振校准寄存器								0000 0000
366H	LPRCCTL	-	-	-	-	-	LPRC2	LPRC1	LPRC0	0000 0100
367H	ULPKEY	UKEY7	UKEY6	UKEY5	UKEY4	UKEY3	UKEY2	UKEY1	UKEY0	0000 0000
368H	BBODCTL	BBODEN	BSCAN	-	-	-	-	-	BBODF	0000 0000
369H	BKPREG0	备份区数据寄存器 0								0000 0000
36AH	BKPREG1	备份区数据寄存器 1								0000 0000



KF8L15Z20XX 数据手册 V0. 9

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	复位初值
36BH	BKPREG2	备份区数据寄存器 2								0000 0000
36CH	BKPREG3	备份区数据寄存器 3								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	$dir \leftarrow (dir)$	1	Z
MOV Rn,dir	0101_rrr0_ffff_ffff	$Rn \leftarrow (dir)$	1	
MOV dir,Rn	0101_rrr1_ffff_ffff	$dir \leftarrow (Rn)$	1	
MOV Rn,#data	1001_1rrr_kkkk_kkkk	$Rn \leftarrow data$	1	
MOV Rn,Rs	1111_1000_11ss_srrr	$Rn \leftarrow (Rs)$	1	
LD Rn,[Rs]	1111_0111_00ss_srrr	$Rn \leftarrow ((Rs))$	1	
ST [Rn],Rs	1111_0111_01ss_srrr	$(Rn) \leftarrow (Rs)$	1	
SWAPR Rn,dir	0100_rrr0_ffff_ffff	$Rn < 7:4 > = dir < 3:0 > \quad Rn < 3:0 > = dir < 7:4 >$	1	
SWAP dir	0100_rrr1_ffff_ffff	$dir < 7:4 > = dir < 3:0 > \quad dir < 3:0 > = dir < 7:4 >$	1	
MOVB #data	1110_0001_kkkk_kkkk	$BANK \leftarrow data$	1	
MOVP #data	1110_0000_kkkk_kkkk	$PCH \leftarrow data$	1	
算术运算指令				
ADD Rm,dir	0010_0rr0_ffff_ffff	$Rm \leftarrow (Rm) + (dir)$	1	CY、DC、Z
ADD dir,Rm	0010_0rr1_ffff_ffff	$dir \leftarrow (Rm) + (dir)$	1	CY、DC、Z
ADD Rn,#data	1000_0rrr_kkkk_kkkk	$Rn \leftarrow (Rn) + data$	1	CY、DC、Z
ADD Rn,Rs	1111_1000_00ss_srrr	$Rn \leftarrow (Rn) + (Rs)$	1	CY、DC、Z
SUB Rm,dir	0011_1rr0_ffff_ffff	$Rm \leftarrow (dir) - (Rm)$	1	CY、DC、Z
SUB dir,Rm	0011_1rr1_ffff_ffff	$dir \leftarrow (dir) - (Rm)$	1	CY、DC、Z
SUB Rn,#data	1010_0rrr_kkkk_kkkk	$Rn \leftarrow data - (Rn)$	1	CY、DC、Z
SUB Rn,Rs	1111_1000_01ss_srrr	$Rn \leftarrow (Rs) - (Rn)$	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	$dir \leftarrow (dir) + 1$	1	Z
INCR dir	0000_1010_ffff_ffff	$R0 \leftarrow (dir) + 1$	1	Z
INC Rn	1111_1111_0001_0rrr	$Rn \leftarrow (Rn) + 1$	1	Z
DEC dir	0000_0111_ffff_ffff	$dir \leftarrow (dir) - 1$	1	Z
DECR dir	0000_0110_ffff_ffff	$R0 \leftarrow (dir) - 1$	1	Z
DEC Rn	1111_1111_0000_1rrr	$Rn \leftarrow (Rn) - 1$	1	Z
逻辑运算指令				
AND Rm,dir	0010_1rr0_ffff_ffff	$Rm \leftarrow (Rm) \wedge (dir)$	1	Z
AND dir,Rm	0010_1rr1_ffff_ffff	$dir \leftarrow (dir) \wedge (Rm)$	1	Z
AND Rn,#data	1000_1rrr_kkkk_kkkk	$Rn \leftarrow (Rn) \wedge data$	1	Z
AND Rn,Rs	1111_1000_10ss_srrr	$Rn \leftarrow (Rn) \wedge (Rs)$	1	Z
ORL Rm,dir	0011_0rr0_ffff_ffff	$Rm \leftarrow (Rm) \vee (dir)$	1	Z
ORL dir,Rm	0011_0rr1_ffff_ffff	$dir \leftarrow (dir) \vee (Rm)$	1	Z
ORL Rn,#data	1001_0rrr_kkkk_kkkk	$Rn \leftarrow (Rn) \vee data$	1	Z
ORL Rn,Rs	1111_1001_00ss_srrr	$Rn \leftarrow (Rn) \vee (Rs)$	1	Z

助记符、操作数	指令格式	指令说明	周期	影响标志
XOR Rm,dir	0001_1rr0_ffff_ffff	$Rm \leftarrow (Rm) \oplus (dir)$	1	Z
XOR dir,Rm	0001_1rr1_ffff_ffff	$dir \leftarrow (dir) \oplus (Rm)$	1	Z
XOR Rn,#data	1010_1rrr_kkkk_kkkk	$Rn \leftarrow (Rn) \oplus data$	1	Z
XOR Rn,Rs	1111_1001_01ss_srrr	$Rn \leftarrow (Rn) \oplus (Rs)$	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	$R0 \leftarrow \neg(dir)$	1	Z
CPL dir	0000_0101_ffff_ffff	$dir \leftarrow \neg(dir)$	1	Z
CPL Rn	1111_1111_0000_0rrr	$Rn \leftarrow \neg(Rn)$	1	Z
RRCR dir	0001_0000_ffff_ffff	$R0 \leftarrow (dir)$ 带进位 C 循环右移 1 位	1	CY
RRC dir	0001_0001_ffff_ffff	$dir \leftarrow (dir)$ 带进位 C 循环右移 1 位	1	CY
RRC Rn	1111_1111_0010_0rrr	$Rn \leftarrow (Rn)$ 带进位 C 循环右移 1 位	1	CY
RLCR dir	0001_0010_ffff_ffff	$R0 \leftarrow (dir)$ 带进位 C 循环左移 1 位	1	CY
RLC dir	0001_0011_ffff_ffff	$dir \leftarrow (dir)$ 带进位 C 循环左移 1 位	1	CY
RLC Rn	1111_1111_0001_1rrr	$Rn \leftarrow (Rn)$ 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				
DECJZ dir	0000_1000_ffff_ffff	$R0 \leftarrow (dir) - 1$, 为 0 跳过下一条指令	1/2	
DECJZ dir	0000_1001_ffff_ffff	$dir \leftarrow (dir) - 1$, 为 0 跳过下一条指令	1/2	
DECJZ Rn	1111_1111_0101_1rrr	$Rn \leftarrow (Rn) - 1$, 为 0 跳过下一条指令	1/2	
INCRJZ dir	0000_1100_ffff_ffff	$R0 \leftarrow (dir) + 1$, 为 0 跳过下一条指令	1/2	
INCJZ dir	0000_1101_ffff_ffff	$dir \leftarrow (dir) + 1$, 为 0 跳过下一条指令	1/2	
INCJZ Rn	1111_1111_0101_0rrr	$Rn \leftarrow (Rn) + 1$, 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
08H	P3	Port 3
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
17H	BANK	BANK
18H	ADSCANCTL	Analog Digital Convert Scan control register
19H	C1CTL	Comparer 1 Control register
1AH	C1OUT	Comparer 1 Output register
1BH	AMPCTL	Amplifier Control register
1DH	ANSH	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 Priority3 register
2AH		Power voltage Calibration register
2BH	VREFCTL	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	ANS0	Analog channel Selection register 0
32H	ANS1	Analog channel Selection register 1
33H	ANS2	Analog channel Selection register 2
34H	ANS3	Analog channel Selection register 3
35H	PUR0	Pull-Up Register 0
36H	IOCL0	P0 Interrupt 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 Periods register High
43H	PWM5H0	Pulse-Width Modulation 5 duty cycle register High 0
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
48H	P3LR	Port 3 Latch Register
49H	TR3	Tri Register 3
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
53H	CTCTL0	Capacitance Touch Control register 0
54H	T2CCR0L	Timer 2 Compare Capture Register 0 Low
55H	PWM5L0	Pulse-Width Modulation 5 duty cycle register Low 0
56H	PWM5L1	Pulse-Width Modulation 5 duty cycle register Low 1
57H	PWM5CTL0	Pulse-Width Modulation 5 Control register 0
58H	ADCINTCTL	Analog Digital Convert 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	P5ASCTL0	Pulse-Width Modulation 5Auto Shutdown Control register 0
5DH	PSTRCTL0	Pulse Auto Steer Control register 0
5EH	ADCDATA3H	Analog Digital Convert Data register 3 High
5FH	T3H	Timer 3 High register
62H	BUZCTL	BUZER Control Register
63H	ADCCTL2	Analog Digital Convert Control register 2
64H	CTCTL1	Capacitance Touch Control register 1
65H	MCTCTL	Matrix Capacitance Touch Control register
66H	MCTI	Matrix Capacitance Touch Current selection
67H	INTEDGCTL	Interrupt Edge Control register
6EH	PUR1	Pull-Up Resistor 1
6FH	PUR2	Pull-Up Resistor 2
6AH	ADCDATA3L	Analog Digital Convert Data register 3 Low
6BH	IOCL3	P3 Interrupt on change register
6CH	PUR3	Pull-up control Register 3

BANK1		
108H	DIVRH	Divider Remainder register High
109H		
10CH	PWM5L2	Pulse-Width Modulation 5 duty cycle register 2 low
10DH	PWM5L3	Pulse-Width Modulation 5 duty cycle register 3 low
10EH	PWM5OC	Pulse-Width Modulation 5 Output Control register
10FH	T2CTL1	Timer 2 Control register1
110H	C4FITLCTL	Comparer 4 Filter Control register
111H	C4FILTPRE	Comparer 4 Filter Per-scale register
112H	C2CTL	Comparer 2 Control register
113H	C3CTL	Comparer 3 Control register
114H	C4CTL	Comparer 4 Control register
115H	DIVBH	Divider B register High
116H	T2CCR1L	
118H	T2CCR1H	
119H	PWM5FC	Pulse-Width Modulation 5 Force Control
11AH	P5ASCTL1	Pulse-Width Modulation 5Auto Shutdown Control register 1
11BH	PSTRCTL1	Pulse Auto Steer Control register 1
11CH	PWM5CTL2	Pulse-Width Modulation 5 Control register 2
11DH	PWM5PC	Pulse-Width Modulation 5 Polarity Control register
11EH	PWM5H2	Pulse-Width Modulation 5 duty cycle register High 2
11FH	PWM5H3	Pulse-Width Modulation 5 duty cycle register High 2
120H	RSCTL1	Receive Status Control register
121H	TXSDR1	Transmit Date Register
122H	RXSDR1	Receive Date Reigster
123H	BRCTL1	Baud Rate Control register
124H	TSCTL1	Transmit Control register
125H	EUBRGL1	Enhance Universal Baud Rate Generator register Low
126H	EUBRGH1	Enhance Universal Baud Rate Generator register High
127H	USLPEN1	
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	DIVBL	Divider B register Low
13DH	DIVQH	Divider Quotient register High
13EH	DIVQL	Divider Quotient register Low
13FH	DIVRL	Divider Remainder register Low

14AH	C1FILTCTL	Comarar 1 Filter Control register
14BH	C1FILTPER	Comarar 1 Filter Per-scale register
14CH	C2FILTCTL	Comarar 2 Filter Control register
14DH	C2FILTPRE	Comarar 2 Filter Per-scale register
14EH	C3FILTCTL	Comarar 3 Filter Control register
14FH	C3FILTPRE	Comarar 3 Filter Per-scale register
159H	VREFCAL0	Reference Voltage Calibration 0
15AH	VREFCAL1	Reference Voltage Calibration 1
15BH	HLVDCTL	
15DH	T3REL	Timer 3 Reload register Low
15EH	T3REH	Timer 3 Reload register High
15FH	DACM	
160H	T4L	Timer 4 register Low
161H	T4H	Timer 4 register High
162H	T4REL	Timer 4 Reload register Low
163H	T4REH	Timer 4 Reload register High
164H	T4CTL	Timer 4 Control register
165H	RC32KCAL	RC32K Calibration
BANK 2		
20CH	P4	Port 4
20DH	P4LR	Port 4 Latch Register
20EH	TR4	Tri Register 4
20FH	PUR4	Pull-up Register 4
210H	P5	Port 5
211H	P5LR	Port 5 Latch Register
212H	TR5	Tri Register 5
213H	PUR5	Pull-up Register 5
214H	P6	Port 6
215H	P6LR	Port 6 Latch Register
216H	TR6	Tri Register 6
218H	PUR6	Pull-up Register 6
219H	P7	Port 7
21AH	P7LR	Port 7 Latch Register
21BH	TR7	Tri Register 7
21CH	PUR7	Pull-up Register 7
21DH	PWMCTL0	Pulse-Width-Modulation Control 0
21EH	PWMCTL1	Pulse-Width-Modulation Control 1
21FH	PP1	Pulse-Width Modulation Periods register 1
220H	PP2	Pulse-Width Modulation Periods register 2
221H	PWM10L	Pulse-Width Modulation 10 Low
222H	PWM10H	Pulse-Width Modulation 10 High
223H	PWM11L	Pulse-Width Modulation 11 Low
224H	PWM11H	Pulse-Width Modulation 11 High
225H	PWM12L	Pulse-Width Modulation 12 Low
226H	PWM12H	Pulse-Width Modulation 12 High
227H	PWM13L	Pulse-Width Modulation 13 Low
228H	PWM13H	Pulse-Width Modulation 13 High
229H	PWM14L	Pulse-Width Modulation 14 Low
22AH	PWM14H	Pulse-Width Modulation 14 High
22BH	PWM15L	Pulse-Width Modulation 15 Low
22CH	PWM15H	Pulse-Width Modulation 15 High
22DH	PWM16L	Pulse-Width Modulation 16 Low
22EH	PWM16H	Pulse-Width Modulation 16 High

22FH	PWM17L	Pulse-Width Modulation 17 Low
230H	PWM17H	Pulse-Width Modulation 17 High
231H	PWM18L	Pulse-Width Modulation 18 Low
232H	PWM18H	Pulse-Width Modulation 18 High
233H	PWM19L	Pulse-Width Modulation 19 Low
234H	PWM19H	Pulse-Width Modulation 19 High
235H	PWM1AL	Pulse-Width Modulation 1A Low
260H	PWM1AH	Pulse-Width Modulation 1A High
261H	PWM1BL	Pulse-Width Modulation 1B Low
262H	PWM1BH	Pulse-Width Modulation 1B High
263H	PWM1CL	Pulse-Width Modulation 1C Low
264H	PWM1CH	Pulse-Width Modulation 1C High
265H	PWM1DL	Pulse-Width Modulation 1D Low
266H	PWM1DH	Pulse-Width Modulation 1D High
267H	PWM1EL	Pulse-Width Modulation 1E Low
268H	PWM1EH	Pulse-Width Modulation 1E High
269H	PWM1FL	Pulse-Width Modulation 1F Low
26AH	PWM1FH	Pulse-Width Modulation 1F High
BANK3		
305H	EIE4	Enable Interrupt register 4
306H	EIF4	Enable Interrupt Flag register 4
307H	IP4	Interrupt Priority 4 register
320H	U7816CTL2	Uart 7816 Control register 2
321H	U7816RXCTL2	Uart 7816 Receive Control register 2
322H	U7816TXCTL2	Uart 7816 Transmit Control register 2
323H	CLKDIV2	
324H	EGTCTL2	
325H	TZBRGH2	
326H	TZBRGL2	
327H	UARTMTH2	
328H	USLPEN2	
329H	BRCTL2	Baud Rate Control register 2
32AH	RSCTL2	Receive Status Control register 2
32BH	RXSDR2	Receive Data Register 2
32CH	EUBRGH2	Enhance Universal Baud Rate Generator register High2
32DH	EUBRGL2	Enhance Universal Baud Rate Generator register Low 2
32EH	TXSDR2	Transmit Data Register 2
32FH	TSCTL2	Transmit Control register 2

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	116.2K
20.000MHz	实际波特率	-	-	2404	9615	10417	—	—	—
	误差(%)	-	-	0.16	0.16	0.00	—	—	—
	EUBRGL 值(十六进制)	-	-	81H	20H	1DH	—	—	—
16.000MHz	实际波特率	—	1202	2404	9615	10417	19231	—	—
	误差(%)	—	0.16	0.16	0.16	0.00	0.16	—	—
	EUBRGL 值(十六进制)	—	CFH	67H	19H	17H	0CH	—	—
8.000MHz	实际波特率	—	1202	2404	9615	10417	—	—	—
	误差(%)	—	0.16	0.16	0.16	0.00	—	—	—
	EUBRGL 值(十六进制)	—	67H	33H	0CH	0BH	—	—	—
4.000MHz	实际波特率	300	1202	2404	—	10417	—	—	—
	误差(%)	0.16	0.16	0.16	—	0.00	—	—	—
	EUBRGL 值(十六进制)	6BH	33H	19H	—	05H	—	—	—
2.000MHz	实际波特率	300	1202	2404	—	10417	—	—	—
	误差(%)	0.16	0.16	0.16	—	0.00	—	—	—
	EUBRGL 值(十六进制)	67H	19H	0CH	—	02H	—	—	—
1.000MHz	实际波特率	300	1202	—	—	—	—	—	—
	误差(%)	0.16	0.16	—	—	—	—	—	—
	EUBRGL 值(十六进制)	33H	0CH	—	—	—	—	—	—
512.00 KHz	实际波特率	307	-	-	—	—	—	—	—
	误差(%)	2.33	-	-	—	—	—	—	—
	EUBRGL 值(十六进制)	19H	-	-	—	—	—	—	—

SYNC=0,HBRG=1,BRG16=0									
波特率 时钟频率		300	1200	2400	9600	10417	19.2K	57.6K	116.2K
20.000MHz	实际波特率	-	-	-	9541	10417	19230	56818	113.6K
	误差(%)	-	-	-	-0.6	0.00	0.16	-1.35	0.03
	EUBRGL 值(十六进制)	-	-	-	82H	77H	41H	15H	0AH
16.000MHz	实际波特率	—	—	—	9615	10417	19231	58824	111.1K
	误差(%)	—	—	—	0.16	0.00	0.16	2.12	3.55
	EUBRGL 值(十六进制)	—	—	—	67H	5FH	33H	10H	08H

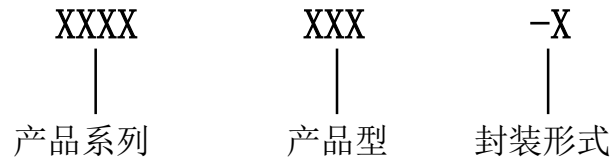
8.000MHz	实际波特率	—	—	2404	9615	10417	19231	55556	—
	误差(%)	—	—	0.16	0.16	0. 00	0.16	-3.55	—
	EUBRGL 值(十六进制)	—	—	CFH	33H	2FH	19H	08H	—
4.000MHz	实际波特率	—	1202	2404	9615	10417	19.23K	—	—
	误差(%)	—	0.16	0.16	0.16	0. 00	0.16	—	—
	EUBRGL 值(十六进制)	—	CFH	67H	19H	17H	0CH	—	—
2.000MHz	实际波特率	—	1202	2404	9615	10417	—	—	—
	误差(%)	—	0.16	0.16	0.16	0. 00	—	—	—
	EUBRGL 值(十六进制)	—	67H	33H	0CH	0BH	—	—	—
1.000MHz	实际波特率	300	1202	2404	—	10417	—	—	—
	误差(%)	0.16	0.16	0.16	—	0.00	—	—	—
	EUBRGL 值(十六进制)	CFH	33H	19H	—	05H	—	—	—
512.000KHz	实际波特率	301	1231	2462	-	-	—	—	—
	误差(%)	0.63	2.58	2.58	-	-	—	—	—
	EUBRGL 值(十六进制)	69H	19H	0CH	-	-	—	—	—

SYNC=0,HBRG=0,BRG16=1									
波特率 时钟频率		300	1200	2400	9600	10417	19.2K	57.6K	116.2K
20.000MHz	实际波特率	-	-	-	9541	10417	19230	56818	113.6K
	误差(%)	-	-	-	-0.6	0.00	0.16	-1.35	0.03
	EUBRGL 值 (十六进制)	-	-	-	82H	77H	41H	15H	0AH
16.000MHz	实际波特率	—	—	—	9615	10417	19231	58824	111.1K
	误差(%)	—	—	—	0.16	0.00	0.16	2.12	3.55
	EUBRGL 值 (十六进制)	—	—	—	67H	5FH	33H	10H	08H
8.000MHz	实际波特率	299.9	1999	2404	9615	10417	19231	55556	—
	误差(%)	-0.02	-0.08	0.16	0.16	0.00	0.16	-3.55	—
	EUBRGL 值 (十六进制)	0682H	01A0H	CFH	33H	3FH	19H	08H	—
4.000MHz	实际波特率	300.1	1202	2404	9615	10417	19.23K	—	—
	误差(%)	0.04	0.16	0.16	0.16	0.00	0.16	—	—
	EUBRGL 值 (十六进制)	0340H	CFH	67H	19H	17H	0CH	—	—
2.000MHz	实际波特率	299.8	1202	2404	9615	10417	—	—	—
	误差(%)	-0.18	0.16	0.16	0.16	0.00	—	—	—
	EUBRGL 值 (十六进制)	01A0H	33H	33H	0BH	0BH	—	—	—
1.000MHz	实际波特率	300	1202	2404	—	10417	—	—	—
	误差(%)	0.16	0.16	0.16	—	0.00	—	—	—
	EUBRGL 值 (十六进制)	CFH	33H	19H	—	05H	—	—	—
512.000KHz	实际波特率	301	1231	2462	-	-	—	—	—
	误差(%)	0.63	2.58	2.58	-	-	—	—	—
	EUBRGL 值 (十六进制)	69H	19H	0CH	-	-	—	—	—

SYNC=0,HBRG=1,BRG16=1 或 SYNC=1, BRG16=1									
波特率 时钟频率		300	1200	2400	9600	10417	19.2K	57.6K	116.2K
20.000MHz	实际波特率	300	1200	2399	9597	10417	19157	57.47K	113.6K
	误差(%)	0.00	0.00	0.03	0.03	0.00	0.22	0.22	0.03
	EUBRGH: EUBRGL	411AH	1046H	0823H	0208H	01DFH	0104H	0056H	002BH
16.000MHz	实际波特率	300	1200	2401	9615	10417	19.23K	57971	117.6K
	误差(%)	0.00	0.01	0.04	0.16	0.00	0.16	0.64	2.12
	EUBRGH: EUBRGL	3414H	0D04H	0681H	019FH	017FH	00CFH	0044H	0021H
8.000MHz	实际波特率	300.0	1200	2401	9615	10417	19.23K	57.14K	117.6K
	误差(%)	0.00	-0.02	0.04	0.16	0.00	0.16	-0.79	2.12
	EUBRGH: EUBRGL	1A0A H	0682H	0340H	00CFH	00BFH	0067H	0022H	0010H
4.000MHz	实际波特率	300.0	1200	2398	9615	10417	19.23K	56.82K	111.1K
	误差(%)	0.01	0.04	0.08	0.16	0.00	0.16	2.12	-3.55
	EUBRGH: EUBRGL	0D04H	0340H	01A0H	0067H	005FH	0033H	0010H	0008H
2.000MHz	实际波特率	299.9	1199	2404	9615	10417	19.23K	55.56K	—
	误差(%)	-0.02	-0.08	0.16	0.16	0.00	0.16	-3.55	—
	EUBRGH: EUBRGL	0682H	01A0H	00CFH	0033H	002FH	0019H	0008H	—
1.000MHz	实际波特率	300.1	1202	2404	9615	10417	19.23K	—	—
	误差(%)	0.04	0.16	0.16	0.16	0.00	0.16	—	—
	EUBRGH: EUBRGL	0340H	00CFH	0067H	0019H	0017H	000CH	—	—
512.000KHz	实际波特率	300.5	1208	2415	9846	10666	—	—	—
	误差(%)	0.16	0.63	0.63	2.56	2.40	—	—	—
	EUBRGH: EUBRGL	01A9H	0069H	0034H	000CH	000BH	—	—	—

注：以上表格中波特率寄存器的值均是十六进制的数值形式

产品标识体系



产品系列: KF8L = KF 系列低功耗 8 位单片机

产品型号: 15Z = 15 系列

Flash 大小: 20 = 20Kbyte

封装形式: OG = SSOP-20
 QP = LQFP-32
 QS = LQFP-48
 NM = QFN-24

版本信息

数据手册版本号	更新内容概述	更新日期
V0.0	新版本	2018-5-15
V0.1	更新电气规范章节	2018-5-26
V0.2	更新电气规范章节表格格式错误	2018-5-29
V0.3	1.改正 UPSEL1 位描述错误; 2.运放校准方法步骤 9 改正: 如果 AMPDT<2:0>加到 111 时还是相等, 也结束运放失调校准; 3.更正 RST 复位章节, 应该为 P0.3/RST,而非 P0.2/RST。	2018-9-10
V0.4	1.改正 USART 章节关于 ANSx 位的不恰当描述; 2.改正附录中 ANSx 寄存器的复位值。	2018-10-17
V0.5	1.改正引脚示意图和引脚功能说明的描述错误; 2.改正 ADC 章节 ADCHSx 位和通道对应表。	2018-10-19
V0.6	更新 CCP 章节;	2018-11-6
V0.7	添加缺失的封装信息;	2019-01-23
V0.8	1.P0 口章节增加对 P0.3~P0.6 口的特别说明和原理框图; 2.备份区寄存器的读写章节增加对 BKPREG0-3 寄存器写入的说明和样例程序; 3.增加 BKPCTL 寄存器的 BKP0 需要在初始化程序中置 1 的备注。	2019-03-01
V0.9	增加订货号 KF8L15Z20NM QFN-24 封装	2019-07-09

ROSH 认证

本产品已通过 ROSE 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