

8 位微控制器

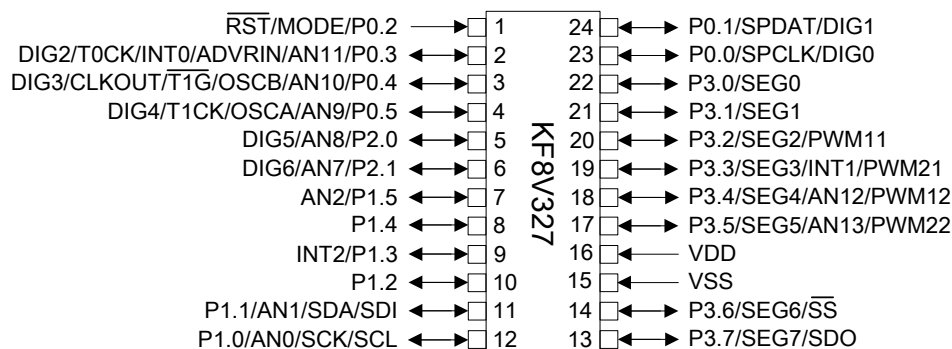
K8V327 数据手册

产品订购信息

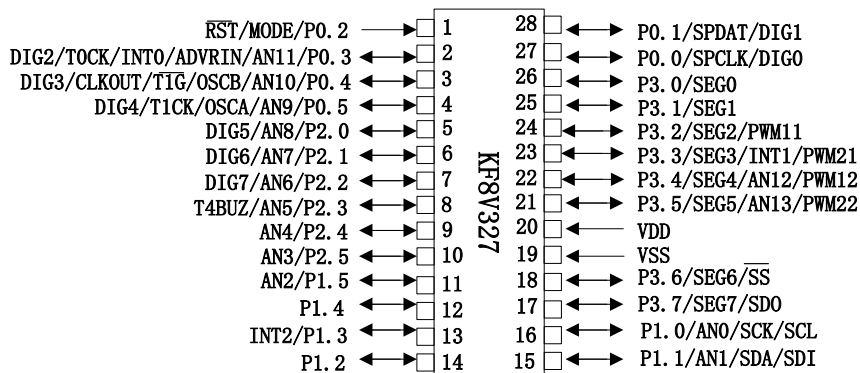
型号	订货号	封装	FLASH	RAM (Byte)	内部 HFOSC(Hz)	外部 HF/LF OSC(Hz)	8 位定时器	16 位定时器	8 位 PWM	LED	12 位 ADC 通道	SSCI(SPI/I2C)	工作电压(V)	内核版本
KF8V327	KF8V327SN	SOIC28	4K×16 位	400	16M	16M/ 32.768k	1	2	4	Y	14	1/1	2.6~5.5	V2
	KF8V327OM	SSOP24									10			

引脚示意图

24 引脚示意图:



28 引脚示意图:



引脚功能说明

引脚名	I/O	引脚功能	引脚说明
1	I/O	P0.2	带电平变化中断功能的输入端口
		$\overline{\text{RST}}$	主复位信号输入
		MODE	编程模式选择
2	I/O	P0.3	带电平变化中断功能和上拉功能的双向输入输出端口
		AN11	ADC 输入通道 11
		ADVRIN	A/D 转换外部参考电压输入
		INT0	外部中断 0 输入
		DIG2	LED 位选端/ Vbias 偏压输出端
		T0CK	T0 用作计数器时的外部计数脉冲输入端
3	I/O	P0.4	带电平变化中断功能和上拉功能的双向输入输出端口
		$\overline{\text{T1G}}$	T1 门控信号输入
		AN10	ADC 输入通道 10
		OSCB	外部振荡器输入引脚 B
		DIG3	LED 位选端/ Vbias 偏压输出端
		CLKOUT	系统时钟输出引脚
4	I/O	P0.5	带电平变化中断功能和上拉功能的双向输入输出端口
		AN9	ADC 输入通道 9
		OSCA	外部振荡器输入引脚 A
		DIG4	LED 位选端/ Vbias 偏压输出端
		T1CK	T1 用作计数器时的外部计数脉冲输入端
5	I/O	P2.0	带上拉功能的双向输入输出端口
		AN8	ADC 输入通道 8
		DIG5	LED 位选端/ Vbias 偏压输出端
6	I/O	P2.1	带上拉功能的双向输入输出端口
		AN7	ADC 输入通道 7
		DIG6	LED 位选端/ Vbias 偏压输出端
7	I/O	P2.2	带上拉功能的双向输入输出端口
		AN6	ADC 输入通道 6
		DIG7	LED 位选端/ Vbias 偏压输出端
8	I/O	P2.3	带上拉功能的双向输入输出端口
		AN5	ADC 输入通道 5
		T4BUZ	T4 蜂鸣器输出
9	I/O	P2.4	带上拉功能的双向输入输出端口
		AN4	ADC 输入通道 4
10	I/O	P2.5	带上拉功能的双向输入输出端口
		AN3	ADC 输入通道 3
11	I/O	P1.5	带上拉功能的双向输入输出端口
		AN2	ADC 输入通道 2
12	I/O	P1.4	带上拉功能的双向输入输出端口
		P1.3	带上拉功能的双向输入输出端口
13	I/O	INT2	外部中断 2 输入
		P1.2	带上拉功能的双向输入输出端口

15	I/O	P1.1	带上拉功能的双向输入输出端口
		AN1	ADC 输入通道 1
		SDA	IIC 数据输入/输出
		SDI	SPI 数据输入
16	I/O	P1.0	带上拉功能的双向输入输出端口
		AN0	ADC 输入通道 0
		SCK	SPI 时钟脚
		SCL	I2C 时钟脚
17	I/O	P3.7	双向输入输出端口
		SEG7	LED 段选端/ Vbias 偏压输出端
		SDO	SPI 数据输出
18	I/O	P3.6	双向输入输出端口
		SEG6	LED 段选端/ Vbias 偏压输出端
		$\overline{SS}$	SPI 从动选择输入
19	P	VSS	电源
20	P	VDD	地线, 0V 参考点
21	I/O	P3.5	双向输入输出端口
		SEG5	LED 段选端/ Vbias 偏压输出端
		AN13	ADC 输入通道 13
		PWM22	PWM22 输出端
22	I/O	P3.4	双向输入输出端口
		SEG4	LED 段选端/ Vbias 偏压输出端
		AN12	ADC 输入通道 12
		PWM12	PWM12 输出端
23	I/O	P3.3	双向输入输出端口
		SEG3	LED 段选端/ Vbias 偏压输出端
		PWM21	PWM21 输出端
		INT1	外部中断 1 输入
24	I/O	P3.2	双向输入输出端口
		SEG2	LED 段选端/ Vbias 偏压输出端
		PWM11	PWM11 输出端
25	I/O	P3.1	双向输入输出端口
		SEG1	LED 段选端/ Vbias 偏压输出端
26	I/O	P3.0	双向输入输出端口
		SEG0	LED 段选端/ Vbias 偏压输出端
27	I/O	P0.0	带电平变化中断功能和上拉功能的双向输入输出端口
		SPCLK	编程时钟输入
		DIG0	LED 位选端/ Vbias 偏压输出端
28	I/O	P0.1	带电平变化中断功能和上拉功能的双向输入输出端口
		SPDAT	编程数据输入/输出
		DIG1	LED 位选端/ Vbias 偏压输出端

目 录

产品订购信息	2
引脚示意图	3
引脚功能说明	4
目 录	6
1 系统概述	13
1.1 芯片特征	14
1.2 系统框图	16
1.3 存储器	17
1.4 系统时钟	18
1.4.1 时钟模块相关寄存器	19
1.4.1.1 系统频率控制寄存器 OSCCTL	19
1.4.1.2 系统时钟标志寄存器 OSCSTA	20
1.4.2 上电延时	20
1.4.3 内部高频时钟 INTHF	20
1.4.4 内部低频振荡器 INTLF	20
1.4.5 外部高频振荡器 EXTHF	21
1.4.6 外部低频振荡器 EXT LF	21
1.4.7 时钟切换和时钟信号同步	21
1.4.8 双速启动模式	22
1.4.9 双速启动过程	22
1.4.10 外部时钟故障检测	23
1.4.11 时钟故障保护处理	23
1.5 芯片初始化	24
1.6 配置位	24
1.7 在线串行编程	26
2 I/O 端口介绍	27
2.1 I/O 端口的读写	27
2.2 I/O 端口数字输入电平类型	27
2.3 P0 口	28
2.3.1 P0 口相关的寄存器	28
2.3.1.1 P0 口状态寄存器 (P0)	29
2.3.1.2 P0 口输出锁存寄存器 (P0LR)	29
2.3.1.3 P0 口方向控制寄存器 (TR0)	29
2.3.1.4 P0 上拉功能控制寄存器 (PUR0)	30
2.3.1.5 电平变化中断控制寄存器 (IOCL)	30
2.3.2 P0 口各引脚内部原理功能框图	31

2.4	P1 口	34
2.4.1	P1 口相关的寄存器	34
2.4.1.1	P1 口状态寄存器(P1)	34
2.4.1.2	P1 口输出锁存寄存器 (P1LR)	35
2.4.1.3	P1 口方向控制寄存器(TR1)	35
2.4.1.4	P1 口上拉功能控制寄存器(PUR1)	35
2.4.2	P1 口原理功能框图	36
2.5	P2 口	37
2.5.1	P2 口相关的寄存器	37
2.5.1.1	P2 口状态寄存器(P2)	37
2.5.1.2	P2 口输出锁存寄存器 (P2LR)	38
2.5.1.3	P2 口方向控制寄存器(TR2)	38
2.5.1.4	P2 口上拉功能控制寄存器(PUR2)	38
2.5.2	P2 口原理功能框图	39
2.1	P3 口	40
2.1.1	P3 口相关的寄存器	40
2.1.1.1	P3 口状态寄存器(P3)	41
2.1.1.2	P3 口输出锁存寄存器 (P3LR)	41
2.1.1.3	P3 口方向控制寄存器(TR3)	41
2.1.2	P3 口原理功能框图	42
3	存储器	43
3.1	程序存储器(ROM)区	43
3.1.1	MOVP 指令	44
3.1.2	JMP、CALL 指令	44
3.2	数据存储器(RAM)区	45
3.2.1	通用寄存器区	45
3.2.2	特殊功能寄存器(SFR)区	46
3.3	FLASH 自写	47
3.3.1	寄存器 NVMDATAH/L	48
3.3.2	寄存器 NVMADDRH/L	48
3.3.3	写 Flash	48
3.3.4	读 Flash	50
3.4	寄存器组 RN	50
3.5	ID 地址单元	50
4	汇编指令及寻址方式	51
4.1	寻址方式	51
4.1.1	寄存器寻址	51
4.1.2	直接寻址	51
4.1.3	立即数寻址	51
4.1.4	寄存器间接寻址	52
4.1.5	位寻址	52
4.2	汇编指令	52

5	中断	53
5.1	中断相关的寄存器	55
5.1.1	中断控制寄存器 INTCTL	55
5.1.2	中断使能寄存器 EIE1	56
5.1.3	中断使能寄存器 EIE2	57
5.1.4	中断使能寄存器 EIE3	57
5.1.5	中断标志寄存器 EIF1	58
5.1.6	中断标志寄存器 EIF2	58
5.1.7	中断标志寄存器 EIF3	59
5.1.8	中断优先级控制寄存器 IP0	59
5.1.9	中断优先级控制寄存器 IP1	59
5.1.10	中断优先级控制寄存器 IP2	60
5.1.11	中断优先级控制寄存器 IP3	60
5.1.12	电源控制寄存器 PCTL	61
5.1.13	INT 中断沿选择寄存器 INTEDGCTL	61
5.1.14	中断响应	63
5.2	INT 中断	64
5.2.1	INT0 中断	64
5.2.2	INT1 中断	64
5.2.3	INT2 中断	64
5.3	定时器中断	65
5.4	P0 口中断	65
5.5	PWM 中断	65
5.6	中断现场保护	65
6	定时/计数器	66
6.1	定时器/计数器 T0	66
6.1.1	T0 原理框图	66
6.1.2	T0 相关的寄存器	66
6.1.2.1	OPTR 选择寄存器	67
6.1.3	定时模式	67
6.1.4	计数模式	67
6.1.5	T0 的使用	68
6.2	定时器/计数器 T1	69
6.2.1	T1 原理框图	69
6.2.2	T1 时钟	69
6.2.3	T1 相关的寄存器	70
6.2.3.1	T1 控制寄存器	70
6.2.4	定时模式	71
6.2.5	计数模式	71
6.2.6	T1 重载功能	71
6.2.7	T1 在休眠模式下的运行	71
6.2.8	T1 分配给 PWM1/2	72
6.3	定时/计数器 T4	73

6.3.1 T4 原理框图	73
6.3.2 T4 相关寄存器	73
6.3.2.1 T4 控制寄存器	73
6.3.2.2 T4 预分频器	74
6.3.2.3 T4 计数时钟选择	74
6.3.3 T4 重载功能	74
6.3.4 T4 蜂鸣器功能	75
6.3.5 T4 中断	75
6.3.6 T4 工作在休眠模式	75
7 模数(A/D)转换模块	76
7.1 与 AD 相关的寄存器	76
7.1.1 AD 控制寄存器 0 (ADCCTL0)	77
7.1.2 AD 控制寄存器 1 (ADCCTL1)	77
7.1.3 模拟/数字口选择寄存器 (ANSEL/H)	78
7.2 通道的选择	78
7.3 模拟输入口的配置	79
7.4 A/D 转换参考电压的选择	79
7.5 转换时钟的选择	79
7.6 输出格式	79
7.7 A/D 转换的启动和完成	79
7.8 A/D 工作在休眠模式	80
7.9 复位的影响	80
7.10 使用 A/D 转换器的设置	80
8 PWM1X/2X 模块	81
8.1 PWM1x/PWM2x 模块	81
8.1.1 PWM1x/2x 相关的寄存器	82
8.1.2 PWM1x/2x 控制寄存器	82
8.1.3 PWM1x/2x 周期	82
8.1.4 PWM1x/2x 占空比	83
8.1.5 PWM1x/2x 分辨率	83
8.1.6 PWM1x/2x 中断	83
8.1.7 休眠模式下的操作	83
8.1.8 复位的影响	83
8.2 PWM1x/2x 使用方法	84
9 SSCI 模块	85
9.1 概述	85
9.2 SSCI 相关寄存器	85
9.2.1 SSCI 控制寄存器 0 (SSCICTL0)	85
9.2.2 SSCI 控制寄存器 1 (SSCICTL1)	86
9.2.3 SSCI 状态寄存器 (SSCISTA)	87
9.2.4 SSCI 屏蔽寄存器 (SSCIMSK)	89

9.2.5 SSCI I2C 地址寄存器 (SSCIADD)	89
9.3 I2C 模式	91
9.3.1 工作原理	91
9.3.2 I2C 从动模式	93
9.3.2.1 寻址	93
9.3.2.2 接收	94
9.3.2.3 发送	95
9.3.2.4 广播呼叫地址支持	97
9.3.3 I2C 主控模式	98
9.3.3.1 主控模式支持	98
9.3.3.2 I2C 主模式操作	99
9.3.3.3 波特率发生器	100
9.3.3.4 I2C 主控模式启动条件时序	100
9.3.3.5 I2C 主控模式重复启动条件时序	101
9.3.3.6 I2C 主控模式发送	102
9.3.3.7 I2C 主控模式接收	104
9.3.3.8 应答序列时序	105
9.3.3.9 停止条件序列	106
9.3.3.10 时钟仲裁	107
9.3.4 多主控器模式	108
9.3.4.1 多主机通信, 总线冲突与总线仲裁	108
9.3.4.2 启动条件期间的总线冲突	109
9.3.4.3 重复启动条件期间的总线冲突	111
9.3.4.4 停止条件期间的总线冲突	112
9.3.4.5 SSCI 屏蔽寄存器	113
9.4 SPI 模式	115
9.4.1 工作原理	116
9.4.2 使能 SPI/I/O 与外部链接	116
9.4.3 典型连接	117
9.4.4 主模式	117
9.4.5 从模式	118
9.4.5.1 从动模式	118
9.4.5.2 从动选择同步	119
9.4.6 休眠模式和复位	120
9.4.7 SPI 四种工作模式设置	121
9.4.7.1 主控发送工作流程	121
9.4.7.2 从动接收工作流程	121
9.4.7.3 主控接收工作流程	122
9.4.7.4 从动发送工作流程	122
10 LED 驱动模块	123
10.1 概述	123
10.2 工作原理	124
10.2.1 扫描时钟产生	124

10.2.2 扫描信号产生.....	124
10.3 LED 相关寄存器.....	125
10.3.1 LED 显示驱动控制寄存器(LEDCTL)	125
10.3.2 LED 显示驱动分频寄存器(LEDPRE)	126
10.3.3 LED 显示驱动数据寄存器(LEDDATAx)	127
10.4 辉度调节	127
10.5 输出模式	128
10.5.1 LED 显示驱动输出模式设置寄存器	128
10.5.2 LED 开漏方式选择寄存器	129
10.6 LED 显示驱动使用方法	129
11 VBIAS 偏压输出.....	130
11.1 相关寄存器	130
11.1.1 偏压输出控制寄存器(VBIASCTL)	130
11.1.2 偏压输出设置寄存器(VBIAS1/2EN)	131
11.2 偏压输出使用方法	133
12 复位.....	134
12.1 电源控制状态寄存器(PCTL)	135
12.2 上电复位(POR)	136
12.3 WDT 复位	136
12.4 RST 复位	136
12.5 欠压检测复位(LVR)	137
12.6 上电延时定时器	137
12.7 不同复位条件下对寄存器的影响.....	138
13 休眠模式	140
14 看门狗定时器	141
14.1 WDT 相关寄存器	142
14.1.1 WDT 预分频选择寄存器 WDTCTL.....	142
15 电气规范	143
15.1 极限参数值	143
15.2 静态电流特性	144
15.3 外设电流特性	145
15.4 I/O 端口电平和芯片供电电压特性	146
15.5 A/D 转换器(ADC)特性	147
16 直流特性图表	148
17 封装信息	153
附录 1 K8V327 SFR 地址映射	154
附录 2 特殊功能寄存器(SFR)功能汇总	157

附录 3 汇编指令集	159
附录 4 寄存器全称表	161
附录 5 全双工异步模式的典型波特率和误差值	164
产品标识体系	168
版本说明	169
ROSH 认证	170
声明及销售网络	171

1 系统概述

K8V327 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。

K8V327 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/计数器 T1、一个 16 位定时/计数器 T4、1 个 12 位 14 通道 AD 模块、4 路 8 位 PWM 模块、1 个 SSCI(I2C/SPI)模块、1 个 8×8 的 LED 驱动模块、1 个 16 路的 Vbias 偏压输出模块、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 $(384+16) \times 8$ 位的数据存储器 RAM、 $4K \times 16$ 位的程序存储器。

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 14 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，内部高频范围 62.5kHz~16MHz，软件可选时钟源及分频

● 存储器

8K 字节（4K×16 位）FLASH 程序存储器

(384+16)×8 位的数据存储器

工作寄存器组 R0~R7

FLASH 可经受 100 000 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%(常温)

内部可校正低频 32kHz 时钟

支持在线串行编程

低功耗休眠模式

● I/O 口配置

输入输出:除 P0.2 只能作为输入口外其它端口均为双向输入输出口

内置上拉功能:P0/P1/P2 口带有弱上拉功能(P0.2 除外)

电平变化中断:P0 口均有电平变化中断功能

IO 口数字输入类型:P0/P1/P2 口为 SMIT 型，P3 口为 TTL 型

● 定时器/计数器

定时器 0:带有 8 位预分频器的 8 位定时器/计数器

定时器 1:带门控和预分频器的 16 位定时器/计数器

定时器 4:带有重载功能、预分频及时钟源多选的 16 位定时器/计数器

● 其它外设

4 路 8 位脉宽调制 PWM 模块

1 个 12 位 14 通道 ADC 模块

1 个 SSCI(SPI/I2C)模块

1 个 8×8LED 驱动模块

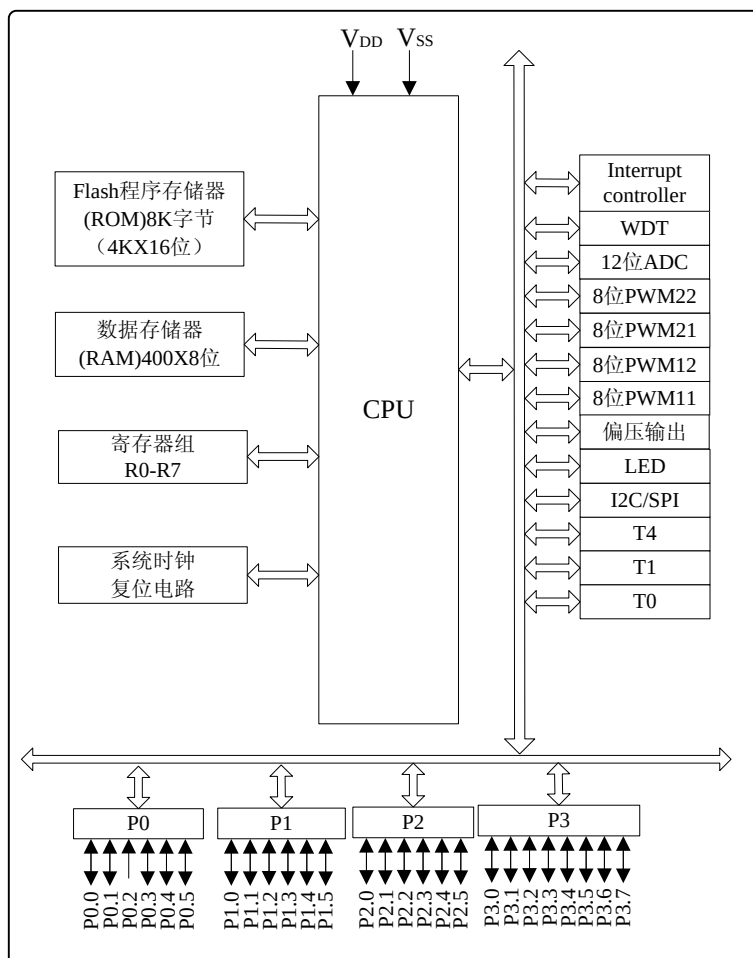
1 个 16 路的 Vbias 偏压输出模块

- 工作条件

工作电压： 2.6V~5.5V

工作温度范围： -30℃~85℃

1.2 系统框图



1.3 存储器

K8V327 单片机的存储器包含:程序存储器(ROM) 和数据存储器(RAM)。

K8V327 的程序存储器空间 8K 字节 (4K×16 位), 寻址范围为 0000H~0FFFH, 可擦写次数为 10 万次。数据存储器分为特殊寄存器区(SFR)和通用存储器区, 其中通用存储器区包括通用存储器区 0 至通用存储器区 2。每个区均有 128×8 位的存储单元, 各区的地址请查阅第 3 章。

有关以上各种存储器的具体介绍请参考第 3 章。

1.4 系统时钟

系统时钟是由系统时钟源分频而来。本芯片中一个机器周期等于四个系统周期，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。

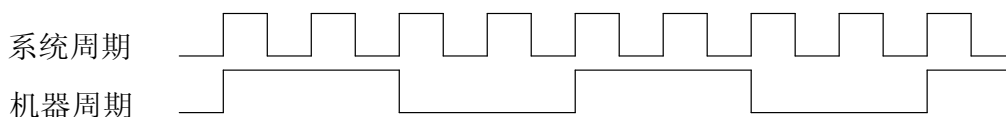


图 1.1 机器周期

K8V327 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 16MHz~125kHz 的时钟源工作；

外部低频振荡器 EXTLF:外接 32.768kHz 的钟表晶振。

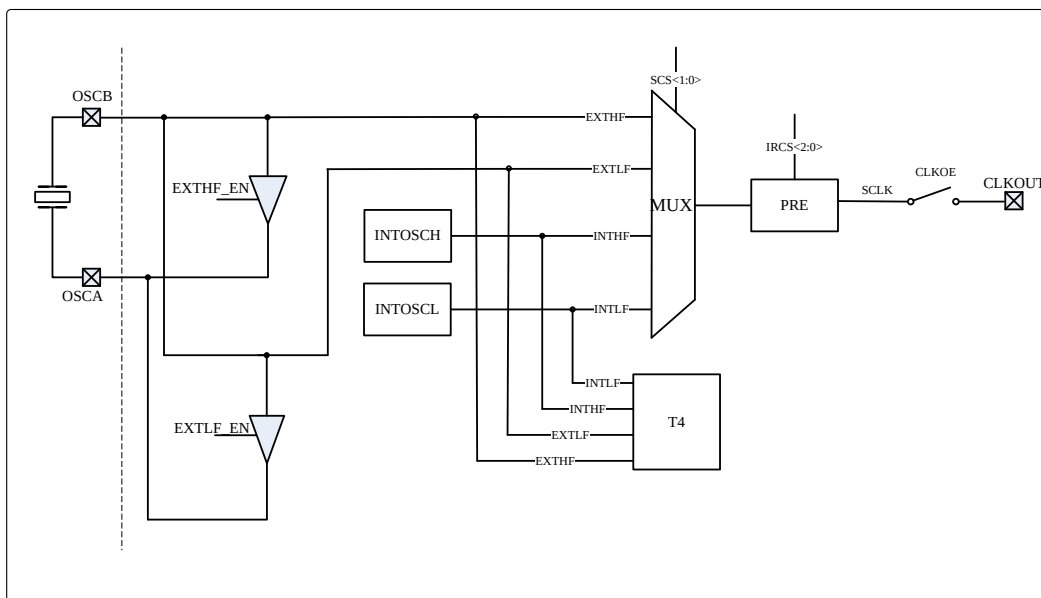


图 1.2 时钟模块原理框图

K8V327 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定时器 T4 的计数时钟。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 EXTLE 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	OSCSA	-	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)

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

CLKOE: 系统时钟输出使能位

1=使能系统时钟输出

0=禁止系统时钟输出

IRCS<2:0>: 时钟频率选择位

111=1:1 (选择内部高频时钟为 16MHz)

110=1:2 (选择内部高频时钟为 8MHz)

101=1:4 (选择内部高频时钟为 4MHz)

100=1:8 (选择内部高频时钟为 2MHz)

011=1:16 (选择内部高频时钟为 1MHz)

010=1:32 (默认, 选择内部高频时钟为 500kHz)

001=1:64 (选择内部高频时钟为 250 kHz)

000=1:256 (选择内部高频时钟为 62.5kHz)

SCS<1:0>: 系统时钟选择位

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

图注: R=可读 W=可写 -=未用 U=未实现位

1.4.1.2 系统时钟标志寄存器 OSCSTA

寄存器1.2: OSCSTA系统时钟标志寄存器(地址:28H)

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

- OSTS: 内外时钟标志位
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

图注: R=可读 W=可写 -=未用 U=未实现位

1.4.2 上电延时

K8V327 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: $T_{pwrt} = \frac{2^{10}}{\text{PWRTCLK}}$ 。

1.4.3 内部高频时钟INTHF

K8V327 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±1%(常温)。

1.4.4 内部低频振荡器 INTLF

LP 振荡器 INTLF 为系统内部低频振荡器, 其振荡器频率为 32kHz, 它不仅可以为 SCLK 时钟提供时钟源, 而且还可以独立作为 LPCLK 时钟信号供外设模块使用。也可以作为内

部上电延时定时器和看门狗 WDT 定时器的时钟。

1.4.5 外部高频振荡器 EXTHF

如图 1.3 所示，引脚 OSCA 和引脚 OSCB 可以接外部标准晶体、陶瓷谐振器或外接 16MHz~125kHz 的时钟为时钟源。EXTHF 时钟源提供主系统时钟 SCLK。

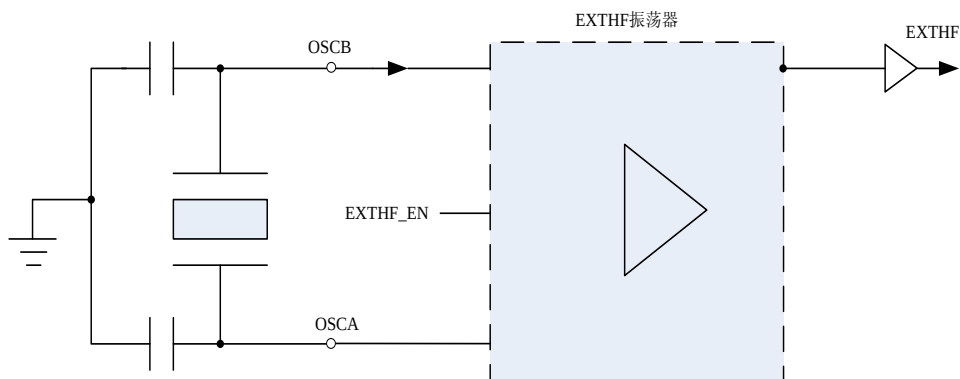


图 1.3 EXTHF 振荡器原理图

1.4.6 外部低频振荡器 EXTLF

如图 1.4 所示，引脚 OSCA 和引脚 OSCB 接外部钟表石英晶体，主要是以 32.768kHz 的晶体提供时钟源。EXTLF 时钟源提供主系统时钟 SCLK。

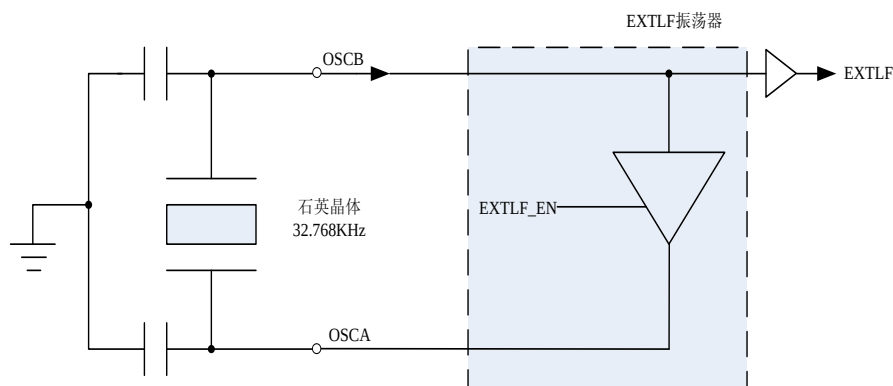


图 1.4 EXTLF 振荡器原理图

1.4.7 时钟切换和时钟信号同步

通过设置 OSCCTL 寄存器的 SCS<1:0>位可以选择不同的时钟源作为系统时钟。SCS 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。

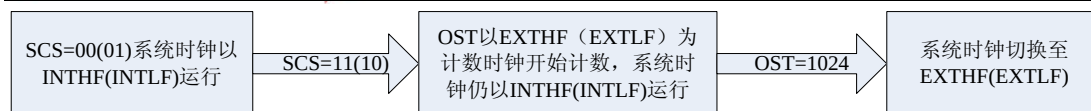


图 1.5 时钟切换流程图

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。

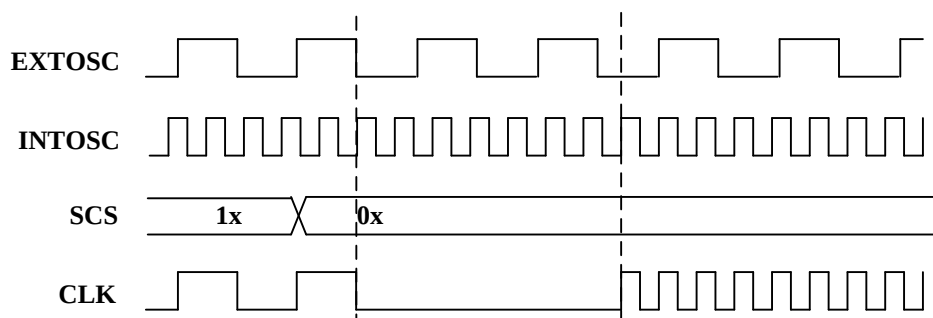


图 1.6 切换至内部时钟源时时钟信号同步时序图

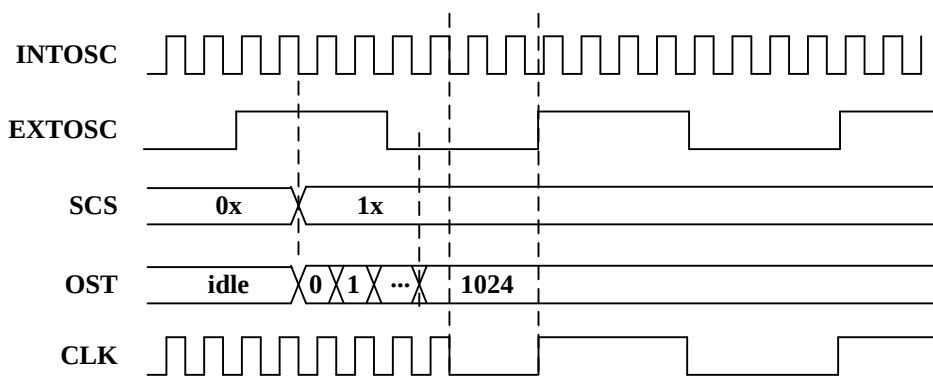


图 1.7 切换至外部时钟源时时钟信号同步时序图

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。

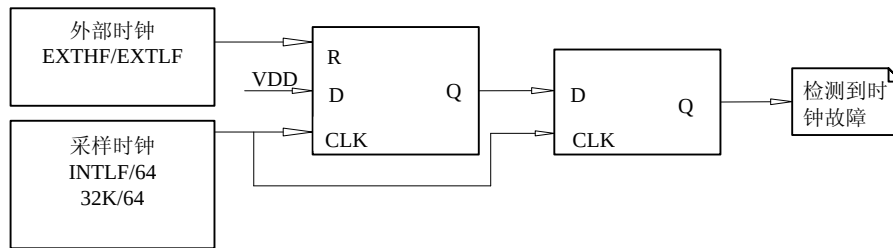


图 1.8 时钟故障检测原理图

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。

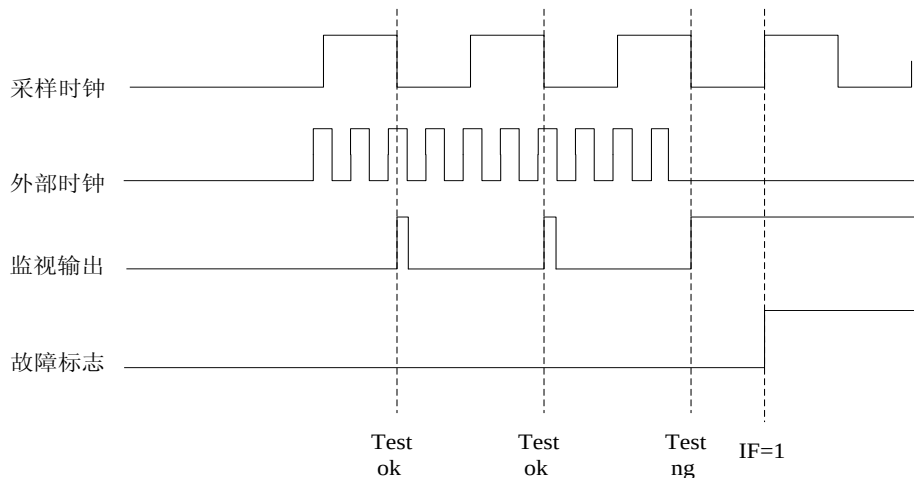


图 1.9 时钟故障检测时序

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，分频值为 OSCCTL 寄存器的 SCLKDIV，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 芯片初始化

用户需在初始化程序中，需要添加相关校准程序，以保证芯片稳定工作。初始化程序如例 1.1 所示。

例1.1 芯片初始化程序

```
CALL    0XFFF
MOV     OSCCAL0, R0
NOPZ
NOPZ
CALL    0XFFE
MOV     OSCCAL1, R0
NOPZ
NOPZ
CALL    0XFFD
MOV     OSCCAL2, R0
NOPZ
NOPZ
CALL    0XFFC
MOV     OSCCAL3, R0
NOPZ
NOPZ
CALL    0XFFB
MOV     VRECAL1, R0
NOPZ
NOPZ
```

1.6 配置位

如寄存器 1.2 所示，用户在烧写程序时，在编程器中通过对配置位进行设置，使单片机启用诸如看门狗、程序代码保护、欠压检测等功能。

CONFIG: 配置字

保留	保留	保留	保留	保留	DEBUG	SWRTEN	保留	CODEP	LVREN	RSTEN	PWRT	WDTEN	保留	保留	保留
bit15						bit8						bit0			

DEBUG: 在线调试使能位
 1 = 禁止在线调试

	0 = 使能在线调试
SWRTEN:	FLASH 自写保护使能位
	1 = 使能写保护, 此时 FLASH 不能写操作,(可读)
	0 = 禁止写保护, 此时 FLASH 可写, (可读)
$\overline{\text{CODEP}}$:	代码保护使能位
	1 = 禁止程序存储器代码保护
	0 = 使能程序存储器代码保护
LVREN:	欠压检测功能使能位
	1 = 使能欠压检查功能
	0 = 禁止欠压检查功能
RSTEN:	P0.2/ $\overline{\text{RST}}$ 引脚功能选择
	1 = P0.2/ $\overline{\text{RST}}$ 引脚配置为外部复位输入
	0 = P0.2/ $\overline{\text{RST}}$ 引脚功能为数字输入口
$\overline{\text{PWRT}}$:	上电延时使能位
	1 = 禁止上电延时
	0 = 使能上电延时
WDTEN:	看门狗定时器(WDT)使能位
	1 = 使能 WDT
	0 = 禁止 WDT

1.7 在线串行编程

如图 1.10、1.11 所示，在最终应用电路中可对 K8V327 单片机进行在线串行编程。实现编程仅需要五根线包括：时钟线(SPCLK)、数据线(SPDAT)、电源线(VDD)、地线(VSS)、编程模式选择线(MODE)。

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。

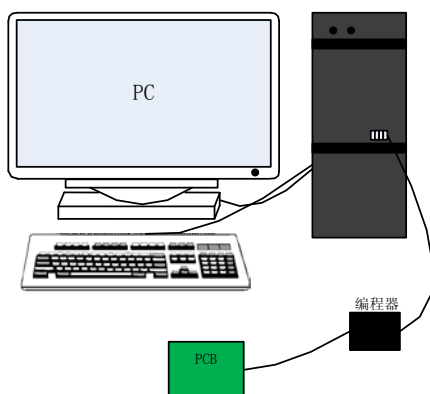


图 1.10 在线调试系统示意图

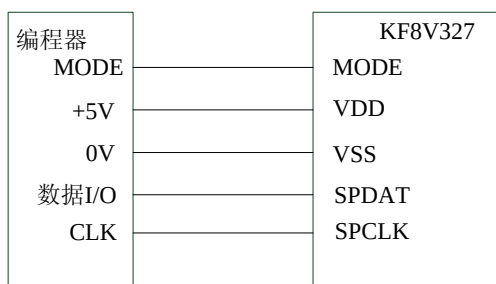


图 1.11 在线串行编程连接图

2 I/O端口介绍

KF8V327 单片机共有 28 个引脚，20 脚接电源正极，19 脚接电源负极，其余管脚均为 I/O 端口，包括 P0 口、P1 口、P2 口和 P3 口。P0 口共有 P0.0~P0.5 六个引脚，P1 口共有 P1.0~P1.5 六个引脚，P2 口共有 P2.0~P2.5 六个引脚，P3 口共有 P3.0~P3.7 八个引脚。

注: 用户在正常使用时，通常会有一些用不到的引脚，如果直接把这些管脚悬空，而不做其他处理可能使单片机功耗增大，因此建议将那些不用的引脚设置为数字输出模式，如果 P0.2 未用，应外接上拉电阻。

2.1 I/O 端口的读写

读 P0/P1/P2/P3 口时实际为读端口电平，即读 P_x ($x=0/1/2/3$)，其原理框图如图 2.1 所示：

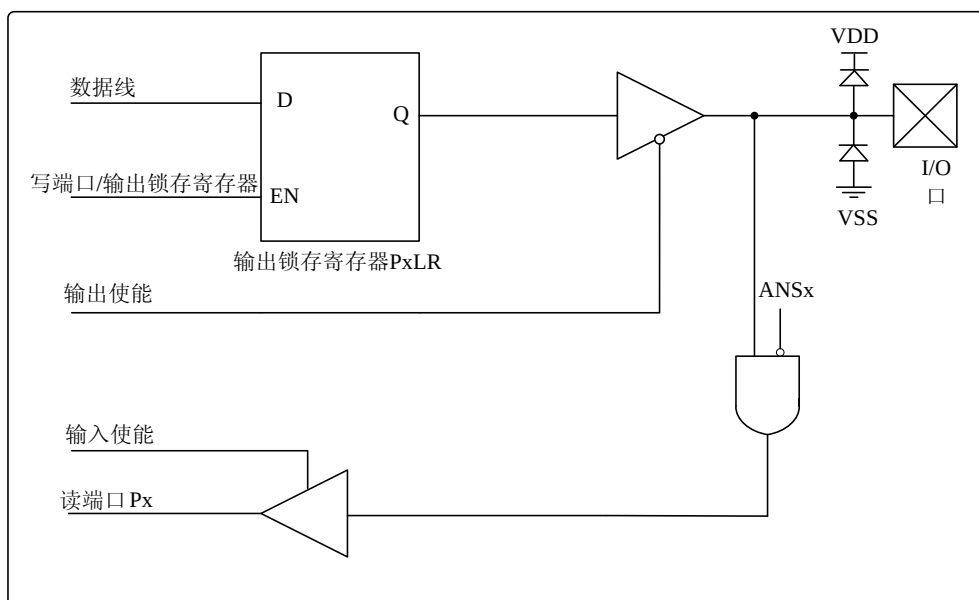


图 2.1 I/O 口读写原理图

IO 端口作为输出时，可对 P_xLR ($x=0/1/2/3$) 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 I/O 端口数字输入电平类型

KF8V327 系列单片机的 I/O 端口数字输入电平类型如下：

I/O 端口	数字输入电平类型
P0	SMIT
P1	SMIT
P2	SMIT
P3	TTL

2.3 P0 口

如引脚示意图所示，P0 口共有 6 个引脚。在线编程时 P0 口的 MODE、SPCLK、SPDAT 作为编程脚使用。P0.2 只能作为输入口且没有上拉功能，其它端口均可作为普通 I/O 口且带有上拉功能，P0 口所有引脚都有电平变化中断功能。各引脚功能表 2-1 所示。

表 2-1 P0 口各引脚功能介绍

1	I/O	P0.2	带电平变化中断功能的输入端口
		$\overline{\text{RST}}$	主复位信号输入
		MODE	编程模式选择
2	I/O	P0.3	带电平变化中断功能和上拉功能的双向输入输出端口
		AN11	ADC 输入通道 11
		ADVRIN	A/D 转换外部参考电压输入
		INT0	外部中断 0 输入
		DIG2	LED 位选端/ Vbias 偏压输出端
		T0CK	T0 用作计数器时的外部计数脉冲输入端
3	I/O	P0.4	带电平变化中断功能和上拉功能的双向输入输出端口
		T1G	T1 门控信号输入
		AN10	ADC 输入通道 10
		OSCB	外部振荡器输入引脚 B
		DIG3	LED 位选端/ Vbias 偏压输出端
		CLKOUT	系统时钟输出引脚
4	I/O	P0.5	带电平变化中断功能和上拉功能的双向输入输出端口
		AN9	ADC 输入通道 9
		OSCA	外部振荡器输入引脚 A
		DIG4	LED 位选端/ Vbias 偏压输出端
		T1CK	T1 用作计数器时的外部计数脉冲输入端
27	I/O	P0.0	带电平变化中断功能和上拉功能的双向输入输出端口
		SPCLK	编程时钟输入
		DIG0	LED 位选端/ Vbias 偏压输出端
28	I/O	P0.1	带电平变化中断功能和上拉功能的双向输入输出端口
		SPDAT	编程数据输入/输出
		DIG1	LED 位选端/ Vbias 偏压输出端

2.3.1 P0 口相关的寄存器

表 2-2 与 P0 端口相关的寄存器

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
05H	P0	-	-	P05	P04	P03	P02	P01	P00
45H	P0LR	-	-	P0LR5	P0LR4	P0LR3	-	P0LR1	P0LR0
25H	TR0	-	-	TR05	TR04	TR03	TR02	TR01	TR00
36H	IOCL	-	-	IOCL5	IOCL4	IOCL3	IOCL2	IOCL1	IOCL0
35H	PUR0	-	-	PUR05	PUR04	PUR03	-	PUR01	PUR00

2.3.1.1 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 --XX XXXX	bit7			bit0				
	-	-	P05	P04	P03	P02	P01	P00
	U	U	R/W	R/W	R/W	R/W	R/W	R/W

P0<5: 0>: 读 P0 口各引脚电平
 1 = 对应引脚为逻辑高电平
 0 = 对应引脚为逻辑低电平

图注: R=可读 W=可写 -=未用 U=未实现位

注: 读 P0 寄存器实际读 P0 引脚的电平状态。

2.3.1.2 P0 口输出锁存寄存器 (POLR)

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

寄存器2.2: POLR: P0口输出锁存寄存器(地址: 45H)

复位值 XXXX XXXX	bit7			bit0				
	-	-	POLR5	POLR4	POLR3	-	POLR1	POLR0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

POLR<5: 3><1: 0>: 写 P0 口输出状态
 1 = 对应引脚输出高电平
 0 = 对应引脚输出低电平

图注: R=可读 W=可写 -=未用 U=未实现位

注: P0.2 不能作输出用。

2.3.1.3 P0 口方向控制寄存器(TR0)

如寄存器 2.3 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 某位清零，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				
	-	-	TR05	TR04	TR03	TR02	TR01	TR00
	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

TR0<5: 4><2: 0>: P0 口各引脚方向控制位
 1 = 对应的引脚设置为输入

0 = 对应的引脚设置为输出

TR02: P0.2 引脚控制位，始终为 1

图注：R=可读 W=可写 -=未用 U=未实现位

2.3.1.4 P0 上拉功能控制寄存器(PUR0)

KF8V327 中除了 P0.2 口没有内部上拉功能外，其它引脚均带有上拉功能，可通过上拉功能控制寄存器和 OPTR 寄存器中的 $\overline{\text{PUPH}}$ 来控制上拉功能是否打开。

如果要将某引脚的上拉功能打开，需要先将 $\overline{\text{PUPH}}$ (上拉功能总使能位)位清零，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2.4: PUR0: P0口弱上拉控制寄存器(地址: 35H)

复位值 1111 1-11	bit7						bit0	
	-	-	PUR05	PUR04	PUR03	-	PUR01	PUR00
	R/W	R/W	R/W	R/W	R/W	U	R/W	R/W

PUR0<5: 3>: 上拉功能使能位

PUR0<1: 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

图注：R=可读 W=可写 -=未用 U=未实现位

2.3.1.5 电平变化中断控制寄存器(IOCL)

P0 口每个引脚都具有电平变化中断功能，当引脚的当前电平与上次读 P0 寄存器时的电平不匹配时将产生电平变化中断。如寄存器 2.5 所示，IOCL 为电平变化中断控制寄存器，将 IOCL 某位置 1 将开启对应引脚的电平变化中断功能，如果该引脚电平发生变化，不管电平变化中断是否使能，电平变化中断标志位(P0IF)都会置 1，如果全局中断使能位(AIE)和电平变化中断使能位(POIE)都已置 1，则会响应中断进入中断服务子程序。P0 口所有引脚的电平变化中断共用一个标志位 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: IOCL: 电平变化中断控制寄存器(地址:36H)

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

IOCL<5: 0>: P0 端口引脚电平变化中断使能控制位

1 = 使能对应引脚的电平变化中断

0 = 禁止对应引脚的电平变化中断

图注: R=可读 W=可写 -=未用 U=未实现位

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

电平变化中断参考:

JNB INTCTL,P0IF ;检测是否为P0电平变化中断

JMP POINT

JMP INT_RET ;退出中断

POINT

MOV P0 ;注意在处理电平变化中断前, 这里一定要更新锁存器锁存的值, 为了下一次电平变化中断处理,

...

;接下来为P0电平变化中断的处理

2.3.2 P0 口各引脚内部原理功能框图

如图 2.2、2.3 所示, 为 P0 口引脚原理功能框图。

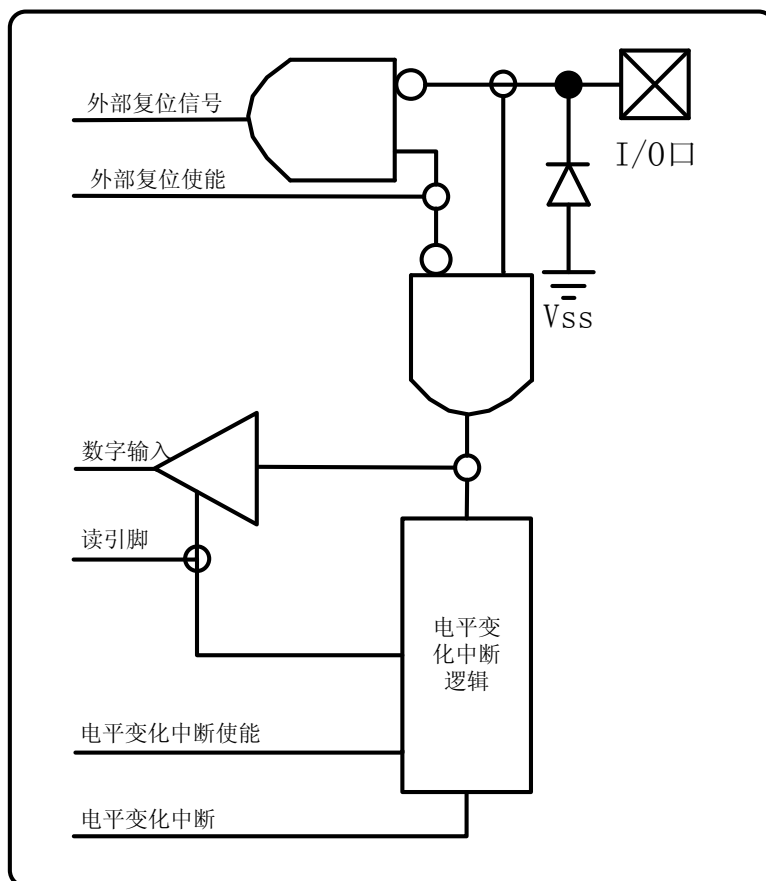


图 2.2 引脚 P0.2 原理功能框图

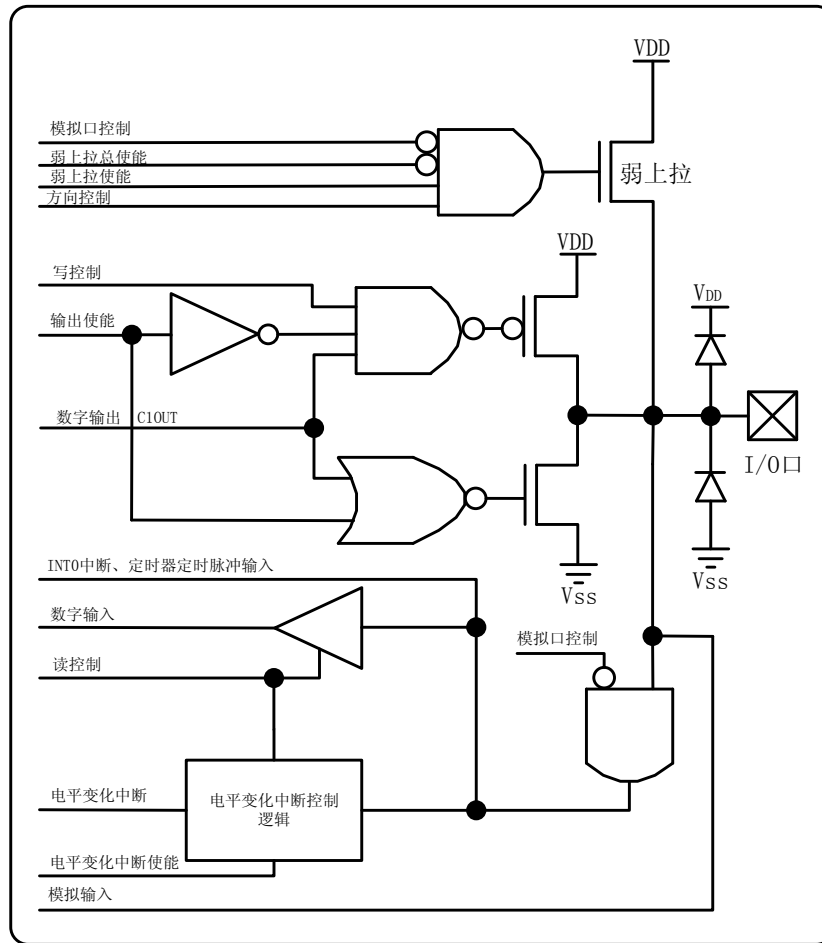


图 2.3 P0.0~P0.1、P0.3~P0.5 口引脚原理框图

2.4 P1 口

如 KF8V327 的 28 引脚示意图所示，P1 口具有 6 个引脚。所有管脚均可作为普通 I/O 口，部分引脚可作为 AD、外部中断 1 等的输入。引脚功能如表 2.3 所示。

表 2-3 P1 口各引脚功能

11	I/O	P1.5	带上拉功能的双向输入输出端口
		AN2	ADC 输入通道 2
12	I/O	P1.4	带上拉功能的双向输入输出端口
13	I/O	P1.3	带上拉功能的双向输入输出端口
		INT2	外部中断 2 输入
14	I/O	P1.2	带上拉功能的双向输入输出端口
15	I/O	P1.1	带上拉功能的双向输入输出端口
		AN1	ADC 输入通道 1
		SDA	SPI 数据输入
		SDI	I2C 数据输入/输出
16	I/O	P1.0	带上拉功能的双向输入输出端口
		AN0	ADC 输入通道 0
		SCK	IIC 数据输入/输出
		SCL	SPI 数据输入

2.4.1 P1 口相关的寄存器

表 2-4 与 P1 口相关的寄存器

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
07H	P1	-	-	P15	P14	P13	P12	P11	P10
47H	P1LR	-	-	P1LR5	P1LR4	P1LR3	P1LR2	P1LR1	P1LR0
27H	TR1	-	-	TR15	TR14	TR13	TR12	TR11	TR10
6EH	PUR1	-	-	PUR15	PUR14	PUR13	PUR12	PUR11	PUR10

2.4.1.1 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 2.6 所示

寄存器2.6: P1: P1口状态寄存器(地址: 07H)

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

P1<5: 0>: 读 P1 口各引脚电平

1 = 对应引脚为逻辑高电平

0 = 对应引脚为逻辑低电平

图注: R=可读 W=可写 -=未用 U=未实现位

注: 读 P1 寄存器实际读 P1 引脚的电平状态。

2.4.1.2 P1 口输出锁存寄存器 (P1LR)

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

寄存器2.7: P1LR: P1口输出锁存寄存器(地址: 47H)

	bit7							bit0
复位值 XXXX XXXX	-	-	P1LR5	P1LR4	P1LR3	P1LR2	P1LR1	P1LR0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P1LR<5: 0>: 写 P1 口输出状态

1 = 对应引脚输出高电平

0 = 对应引脚输出低电平

图注: R=可读 W=可写 -=未用 U=未实现位

2.4.1.3 P1 口方向控制寄存器(TR1)

如寄存器 2.8 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。清零设置为输出口，系统复位时，P1 口各引脚默认为输入口。

寄存器2.8: TR1: P1口方向控制寄存器(地址: 27H)

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

TR1<5: 0>: P1 口引脚方向控制位

1 = P1 口对应引脚被配置为输入端口

0 = P1 口对应引脚被配置为输出端口

图注: R=可读 W=可写 -=未用 U=未实现位

2.4.1.4 P1 口上拉功能控制寄存器(PUR1)

KF8V327 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)位清零，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 2.9 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2.9: PUR1: P1口弱上拉控制寄存器(地址: 60H)

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

PUR1<5: 0>: 上拉功能使能位
 1 = 使能对应的端口上拉功能
 0 = 禁止对应的端口上拉功能

图注: R=可读 W=可写 -=未用 U=未实现位

2.4.2 P1 口原理功能框图

P1 口引脚原理功能如图 2.4 所示:

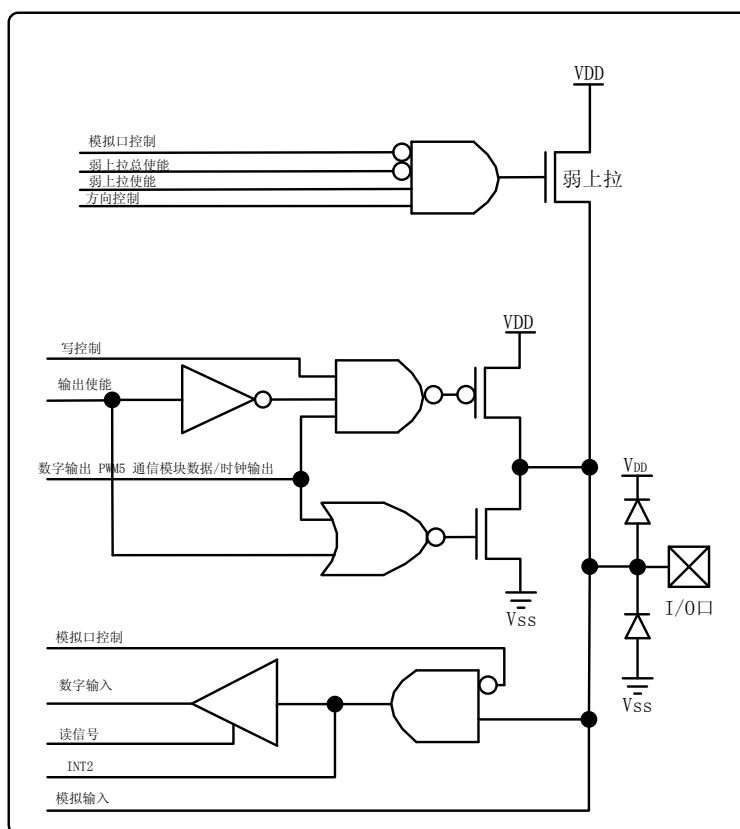


图 2.4 P1 口引脚原理功能框图

2.5 P2 口

如 KV8V327 的 28 引脚示意图所示，P2 口具有 6 个引脚。所有管脚均可作为普通 I/O 口，部分引脚可作为 AD 的输入通道、LED 位选端和 Vbias 偏压输出端。引脚功能如表 2.5 所示。

表 2-5 P2 口各引脚功能

5	I/O	P2.0	带上拉功能的双向输入输出端口
		AN8	ADC 输入通道 8
		DIG5	LED 位选端/ Vbias 偏压输出端
6	I/O	P2.1	带上拉功能的双向输入输出端口
		AN7	ADC 输入通道 7
		DIG6	LED 位选端/ Vbias 偏压输出端
7	I/O	P2.2	带上拉功能的双向输入输出端口
		AN6	ADC 输入通道 6
		DIG7	LED 位选端/ Vbias 偏压输出端
8	I/O	P2.3	带上拉功能的双向输入输出端口
		AN5	ADC 输入通道 5
		T4BUZ	T4 蜂鸣器输出
9	I/O	P2.4	带上拉功能的双向输入输出端口
		AN4	ADC 输入通道 4
10	I/O	P2.5	带上拉功能的双向输入输出端口
		AN3	ADC 输入通道 3

2.5.1 P2 口相关的寄存器

表 2-6 与 P2 口相关的寄存器

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

2.5.1.1 P2 口状态寄存器(P2)

寄存器 P2 对应端口 P2 引脚作为普通 I/O 口时的状态。如寄存器 2.10 所示

寄存器2.10:P2: P2口状态寄存器(地址: 06H)

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

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

图注：R=可读 W=可写 -=未用 U=未实现位

注：读 P2 寄存器实际读 P2 引脚的电平状态。

2.5.1.2 P2 口输出锁存寄存器 (P2LR)

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

寄存器2.11:P2LR: P2口输出锁存寄存器(地址: 46H)

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

P2LR<5: 0>: 写 P2 口输出状态

1 = 对应引脚输出高电平

0 = 对应引脚输出低电平

图注：R=可读 W=可写 -=未用 U=未实现位

2.5.1.3 P2 口方向控制寄存器(TR2)

如寄存器 2.12 所示，通过将寄存器 TR2 中的某位置 1，将对应管脚设置为输入口。清零设置为输出口，系统复位时，P2 口各引脚默认为输入口。

寄存器2.12: TR2: P2口方向控制寄存器(地址: 26H)

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

TR2<5: 0>: P2 口引脚方向控制位

1 = P2 口对应引脚被配置为输入端口

0 = P2 口对应引脚被配置为输出端口

图注：R=可读 W=可写 -=未用 U=未实现位

2.5.1.4 P2 口上拉功能控制寄存器(PUR2)

KF8V327 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)位清零，允许 P2 口上拉功能打开，然后再将要打开上拉功能的引脚，所对应的上拉功能控制位置 1 即可。寄存器 2.13 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2.13: PUR2: P2口弱上拉控制寄存器(地址:61H)

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

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

图注：R=可读 W=可写 -=未用 U=未实现位

2.5.2 P2 口原理功能框图

P2 口引脚原理功能如图 2.5 所示：

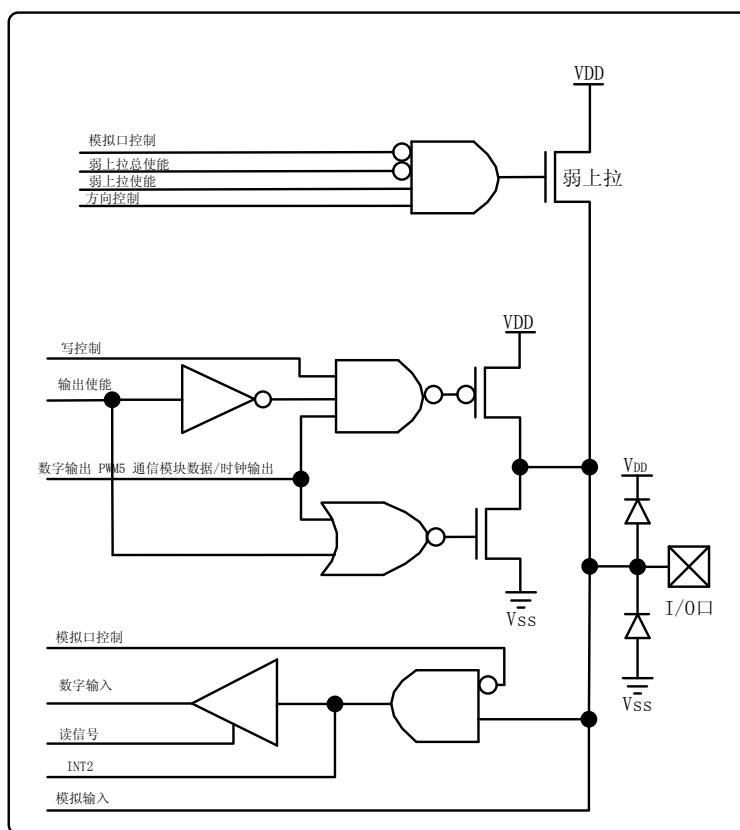


图 2.5 P2 口引脚原理功能框图

2.1 P3 口

如引脚示意图所示，P3 口共有 8 个引脚。所有管脚均可作为通用 I/O 口、部分引脚可作为 ADC 输入通道、PWM 输出口等。引脚功能如表 2-7 所示。

表 2-7 P3 口各引脚功能

17	I/O	P3.7	双向输入输出端口
		SEG7	LED 段选端/ Vbias 偏压输出端
		SDO	SPI 数据输出
18	I/O	P3.6	双向输入输出端口
		SEG6	LED 段选端/ Vbias 偏压输出端
		$\overline{SS}$	SPI 从动选择输入
21	I/O	P3.5	双向输入输出端口
		SEG5	LED 段选端/ Vbias 偏压输出端
		AN13	ADC 输入通道 13
		PWM22	PWM22 输出口
22	I/O	P3.4	双向输入输出端口
		SEG4	LED 段选端/ Vbias 偏压输出端
		AN12	ADC 输入通道 12
		PWM12	PWM12 输出口
23	I/O	P3.3	双向输入输出端口
		SEG3	LED 段选端/ Vbias 偏压输出端
		PWM21	PWM21 输出口
		INT1	外部中断 1 输入
24	I/O	P3.2	双向输入输出端口
		SEG2	LED 段选端/ Vbias 偏压输出端
		PWM11	PWM11 输出口
25	I/O	P3.1	双向输入输出端口
		SEG1	LED 段选端/ Vbias 偏压输出端
26	I/O	P3.0	双向输入输出端口
		SEG0	LED 段选端/ Vbias 偏压输出端

2.1.1 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

2.1.1.1 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 2.14 所示。

寄存器2.14: 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

P3<7: 0>: 读 P3 口各引脚电平
 1 = 对应引脚为逻辑高电平
 0 = 对应引脚为逻辑低电平

图注: R=可读 W=可写 -=未用 U=未实现位

注: 读 P3 寄存器实际读 P3 引脚的电平状态。

2.1.1.2 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, 通过写 P3LR 寄存器来设置输出 P3 口的状态。

寄存器2.15: 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

P3LR<7: 0>: 写 P3 口输出状态
 1 = 对应引脚输出高电平
 0 = 对应引脚输出低电平

图注: R=可读 W=可写 -=未用 U=未实现位

2.1.1.3 P3 口方向控制寄存器(TR3)

如寄存器 2.16 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。清零设置为输出口。

寄存器2.16: 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

TR3<7: 0>: P3 口各引脚方向控制位
 1 = P3 口对应引脚被配置为输入端口
 0 = P3 口对应引脚被配置为输出端口

图注: R=可读 W=可写 -=未用 U=未实现位

2.1.2 P3 口原理功能框图

P3 口共有 8 个引脚，P3 口引脚原理功能如图 2.6 所示：

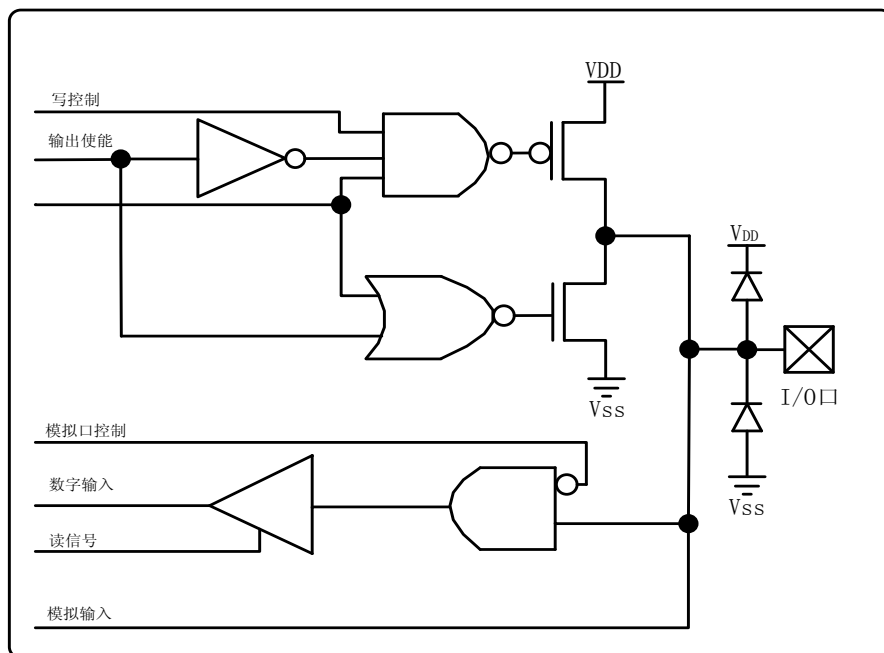


图 2.6 P3 口引脚原理功能框图

3 存储器

如图 3.1 所示，K8V327 中存储器主要由程序存储器(ROM)和数据存储器(RAM)组成，程序存储器和数据存储器地址空间相互独立。其中程序存储器为 8K 字节（4K×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 256×8 位，通用数据寄存器空间为(384+16)×8 位。另外 K8V327 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。

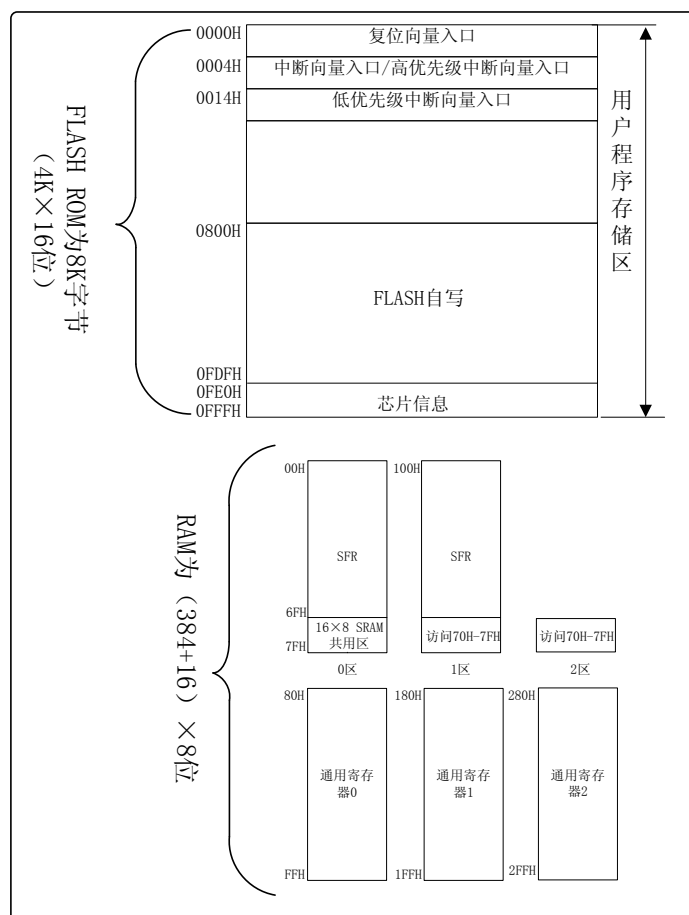


图 3.1 存储器组织图

3.1 程序存储器(ROM)区

K8V327 有一个 13 位的程序计数器，实现了 4K×16 位的程序存储空间，地址为 0000H~0FFFH，复位向量入口地址为 0000H，中断向量有两级入口地址，高为 0004H，低为 0014H。

如图 3.2 所示，程序计数器(PC)的低 8 位(PC<7:0>)来自特殊功能寄存器 PCL，高 5 位(PC<12:8>)来自 PCH 寄存器。在任何复位发生后 PC 值将被清零。在有任何未屏蔽中断发生后 PC 值将指向 0004H 或 0014H 地址。图 3.3 为程序存储器区的地址映射图。

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

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

图 3.2 程序计数器 (PC)

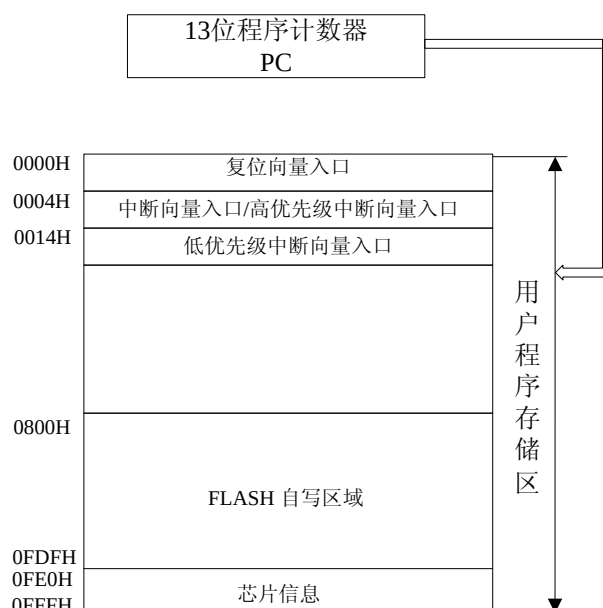


图 3.3 K8V327 程序存储器映射

3.1.1 MOVP 指令

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 5 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 13 位地址数据将更新，变为 PCH 寄存器和 PCL 寄存器的数据，如例 3.1 所示。

例3.2 程序计数器从0000H开始执行跳转程序到1F55H

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	...	...

3.1.2 JMP、CALL 指令

K8V327 系列单片机的 JMP、CALL 指令编码如下：

JMP #data12 1100_kkkk_kkkk_kkkk

CALL #data12 1101_kkkk_kkkk_kkkk

在执行 JMP 或者 CALL 指令时，程序计数器（PC）的值将变为 PCH.bit4 以及指令所带

立即数（#data12），如图 3.4 所示。

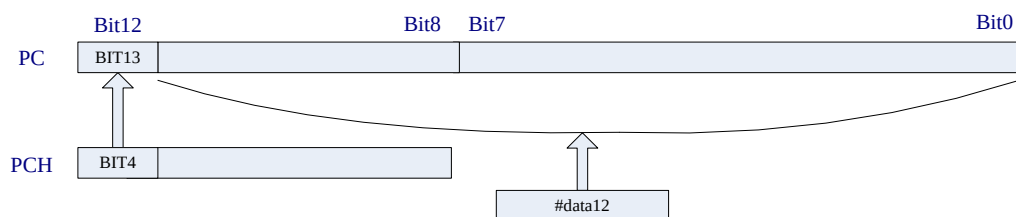


图 3.4 执行 JMP、CALL 指令时 PC 的变化图

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

3.2 数据存储器(RAM)区

如图 3.5 所示，K8V327 中的数据存储器由 5 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 3 个存储器区为通用寄存器区，由用户支配。SFR 地址空间为 00H~7FH、100H~17FH；而 70H~7FH 有 16 个字节为 SRAM 共用区。

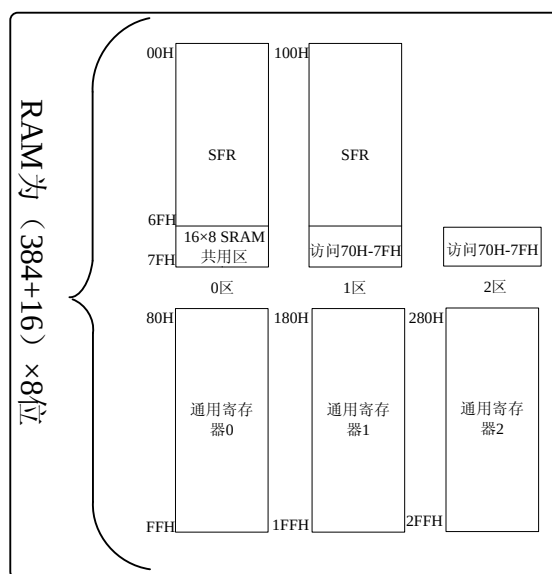


图 3.5 数据存储器地址映射图

3.2.1 通用寄存器区

如图 3.5 所示，通用寄存器的空间为 400 字节，0 区至 2 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-0	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

图注：R=可读 W=可写 -=未用 U=未实现位

切换通用寄存区的指令如例 3.2 所示：

例3.2 切换BANK寄存器存储区

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

3.2.2 特殊功能寄存器(SFR)区

K8V327 内部的 I/O 口控制、定时/计数器、PWM、中断等各种控制寄存器和状态寄存器都称为特殊功能寄存器。附录 1 列出 SFR 的地址映射及复位初始值等。

状态字寄存器(PSW): 如寄存器 3.1 所示, PSW 的低三位是算术运算标志位, 在进行加、减等运算时对它们产生影响(具体请参考汇编指令部分)。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3.1: PSW: 状态字寄存器(地址: 03H)

	bit7							bit0
复位值 0001 1xxx	-	-	-	TO	PD	Z	DC	CY
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 超时被清零

PD: 上电复位标志位

1 = 上电复位或执行 CWDT 指令后
0 = 执行 IDLE 指令后被清零

Z: 零状态标志位

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

DC: 辅助进/借位标志位

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

CY: 进位/借位标志位

1 = 执行结果(8 位)向高位有进位时(加指令)或没有借位(减指令)
0 = 执行结果(8 位)向高位无进位时(加指令)或有借位(减指令)

图注：R=可读 W=可写 -=未用 U=未实现位

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

3.3 FLASH自写

KF8V327 在程序存储区开辟了一个2016×16位的自写区域,地址范围从800H~0FDFH。该区域在正常工作期间是可读写的，它并没有直接映射到寄存器空间，而是通过特殊功能寄存器间接寻址。有6个特殊功能寄存器用于访问该区域。

表 3-2 与 FLASH 相关的寄存器

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

如图 3.5 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。

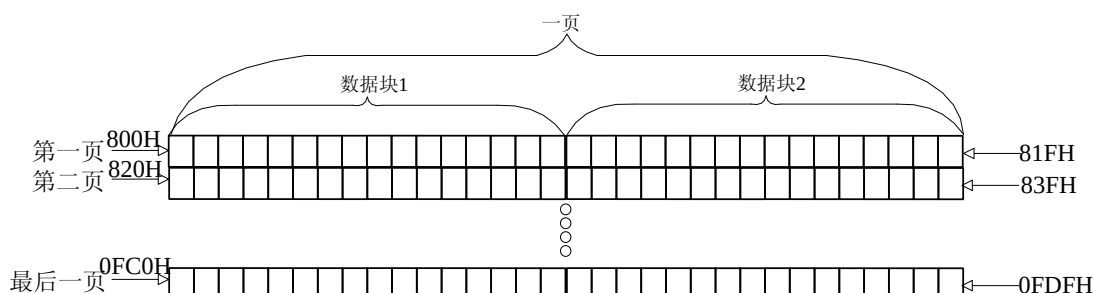


图 3.5 Flash 自写区域地址映射图

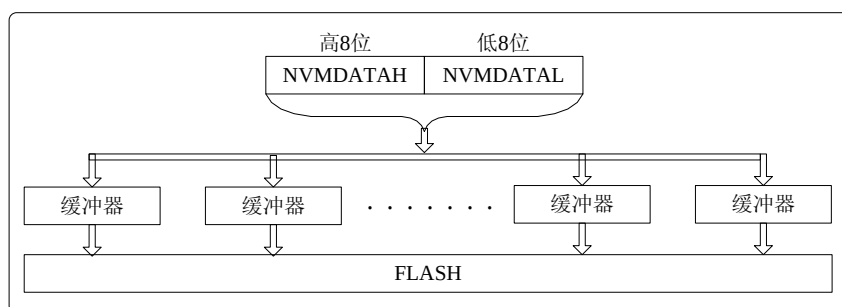


图 3.6 写操作

如图 3.6 所示，在写 Flash 时，有 8 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 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 13 位的地址信息，NVMADDRH 存放地址的高 5 位，NVMADDRL 存放地址的低 8 位。

寄存器3.2: NVMADDRH: 数据指针高5位(地址: 3AH)

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

NVMADDRL: 数据指针低8位(地址: 3BH)								
复位值 0000 0000	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

图注：R=可读 W=可写 -=未用 U=未实现位

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才会擦除本页的数据，对其它块写操作不会擦除本页数据，可能导致写入数据出错。

注：1.写Flash时，从Flash自写首800H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操作，且擦除操作只有在写每页的第一个数据块时才会执行，将本页所有单元数据擦除。而对每页的其他数据块写操作时不会有擦除操作发生。

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

4.配置位的SWRTEN需配置为0，才能对Flash进行写操作。

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

```
MOV R5, BANK           ;保存当前寄存器存储区
CLR BANK               ;切换到Bank0区
MOV DATA_BANK, R5      ;该样例要求DATA_BANK在0区,否则添加切区

MOV R5, INTCTL          ;保存当前的中断状态
MOV DATA_INTCTL, R5     ;该样例要求DATA_INTCTL 在0区,否则添加切区
CLR INTCTL, 7           ;关闭总中断
JNB INTCTL, 7
JMP $-2

MOV R5, OSCCTL          ;保存当前的时钟状态
MOV DATA_OSCCTL, R5     ;该样例要求DATA_OSCCTL在0区,否则添加切区
MOV R5, #0X10           ;切换到250kHz
MOV OSCCTL, R5

;;以下时序不可更改
MOV R5, #0X84
MOV NVMCTL0, R5
MOV R5, #0X69
MOV NVMCTL1, R5
MOV R5, #0X96
MOV NVMCTL1, R5
SET NVMCTL0, 1
NOPZ
NOPZ
NOPZ
NOPZ
NOPZ
NOPZ
NOPZ
NOPZ
NOPZ
NOPZ
MOV R5, #0X80           ;关闭Flash的写操作，防止意外写
MOV NVMCTL0, R5

MOV R5, DATA_OSCCTL    ;恢复时钟状态
MOV OSCCTL, R5

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

MOV R5, DATA_BANK      ;BANK区还原
MOV BANK, R5
```

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

写 FLASH 的步骤为:

1. 将要写入的数据的送到 NVMDATAH/L;
2. 将对应的 FLASH 地址送到 NVMADDRH/L;
3. 执行上面的写命令, 此时, CPU 将要写入数据的一个字保存到 FLASH 的数据缓冲器中;
4. 重复执行步骤 1、2、3 十六次, 此时 CPU 自动将要写入第一块的数据分别存入对应的 FLASH 的数据缓冲器中;
5. 当上边第 16 次写命令执行完后, CPU 自动发出擦除本页的命令, 将本页原来的数据全部擦除, 擦除完毕后, 将数据缓冲器中的数据送到对应的地址中。在这个过程中 CPU 停止其它工作 6ms 用来执行擦除和写入数据的命令。
6. 重复执行步骤 1、2、3 十六次, 将数据写入本页的第二块。当执行完第 16 次写命令后, 因本次写的不是页的第一块, CPU 不会执行擦除命令, 仅将数据缓冲器中的数据写入对应的存储单元, 写操作耗时 3ms。

3.3.4 读 Flash

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

```
MOV R5,#0X81
MOV NVMCTL0,R5
NOPZ
NOPZ
```

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

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

读 FLASH 的步骤如下:

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

3.4 寄存器组 Rn

KF8V327 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。

3.5 ID 地址单元

KF8V327 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, 用于存放芯片校准信息, 地址为 0FE0H~0FFFH。

4 汇编指令及寻址方式

4.1 寻址方式

K8V327 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。K8V327 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 清零
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 位清零
CLR 0X80,1 ; 将 80H 的第 1 位清零
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

K8V327 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

K8V327 单片机的中断源有:

- INT0/1/2
- T0 溢出中断
- T1/4 中断
- P0 口引脚电平变化中断
- A/D 中断
- PWM1/2 中断
- SPI 中断
- I2C 中断
- SSCI 模块 I2C 总线冲突 (BCL) 中断
- 外部时钟故障 (OSCFail) 中断

在本单片机中有两个中断优先级, 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。

K8V327 中的 INT1/2、T1、T4、ADC、PWM1/2、SPI、I2C、BCL 和 OSCFail 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零的中断, 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零 AIEL, 退出中断时置 1 AIEL; 进高优先级中断时清零 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:

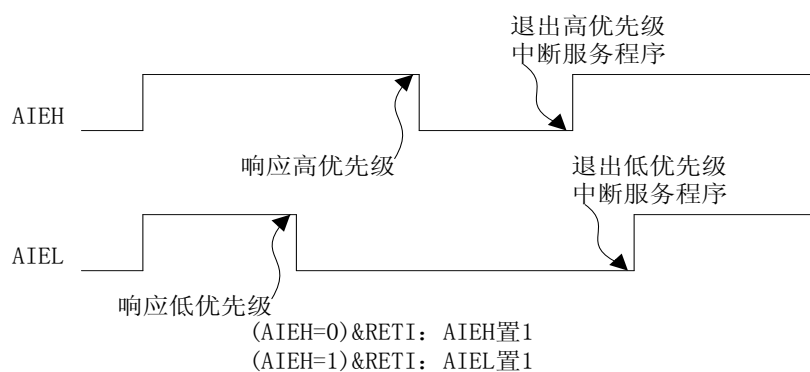


图 5.1 中断优先级工作原理图

当 IPEN 位清零时, 就会禁止中断优先级, 即为普通模式。所有中断都跳转到 0x0004H 开始执行。在普通模式下, 没有中断优先级, 各个中断源的中断优先级控制寄存器 IP0、IP1 和 IP2 均无效。AIE(INTCTL<7>)为全局中断使能位, PUIE(INTCTL<6>)为外设中断使能位。

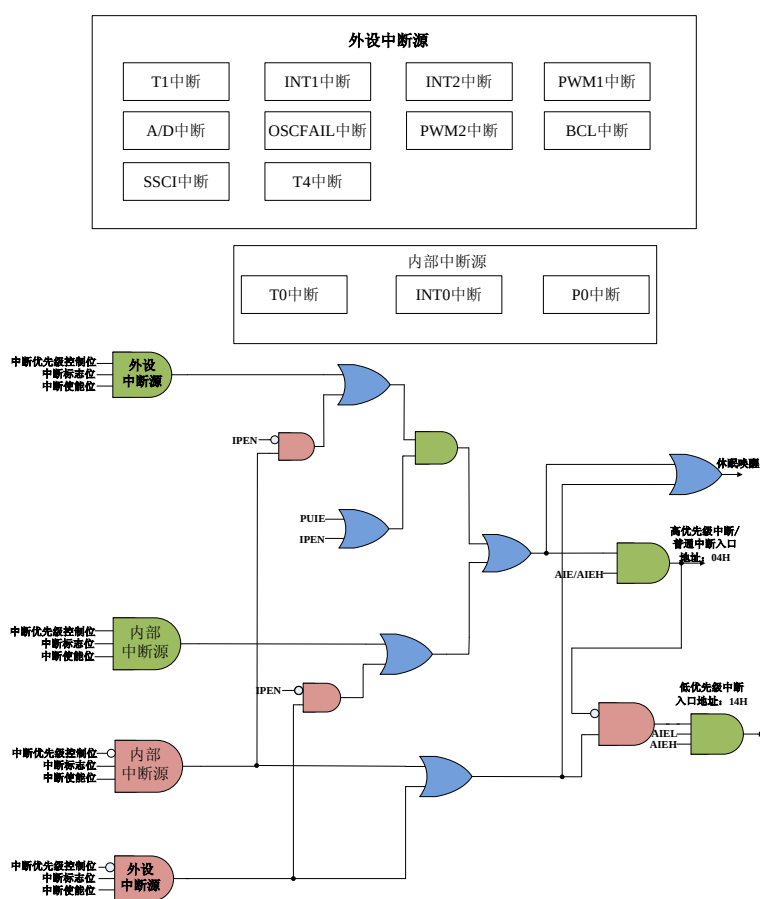


图 5.2 中断逻辑

5.1 中断相关的寄存器

表 5-1 与中断相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
0BH	INTCTL	AIE/AIEH	PUIE/AIEL	T0IE	INT0IE	P0IE	T0IF	INT0IF	P0IF
2CH	EIE1	-	ADIE	INT2IE	INT1IE	-	PWM2IE	-	T1IE
2DH	EIE2	-	-	-	-	-	-	BCLIE	SSCIE
4AH	EIE3	T4IE	-	OSCFAILIE	-	-	-	-	-
0CH	EIF1	-	ADIF	INT2IF	INT1IF	-	PWM2IF	-	T1IF
0DH	EIF2	-	-	-	-	-	-	BCLIF	SSCIF
4BH	EIF3	T4IF	-	OSCFAILIF	-	-	-	-	-
22H	IP0	-	-	-	-	-	PT0	PINT0	PP0
23H	IP1	-	PADC	PINT2	PINT1	-	PPWM2	-	PT1
24H	IP2	-	-	-	-	-	-	PBCL	PSSCI
29H	IP3	PT4	-	POSCFAIL	-	-	-	-	-
2EH	PCTL	-	-	-	SLVREN	IPEN	SWDTEN	POR	BOD
67H	INTEDGCTL	INT2SE	INT1SE	-	-	-	-	-	T1CLKEN

5.1.1 中断控制寄存器 INTCTL

普通模式下，AIE 为全局中断使能位，当其被清零时，禁止所有中断。PUIE 为外设中断使能位，当其被清零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，当其被清零时，禁止所有中断。AIEL 为低优先级中断使能位，当其被清零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。

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	T0IE	INT0IE	P0IE	T0IF	INT0IF	P0IF	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

AIE/AIEH: 全局中断使能位/优先级中断使能位

当 IPEN=0

1 = 使能所有未屏蔽的中断

0 = 禁止所有中断

当 IPEN=1

1 = 允许所有高优先级的中断

0 = 禁止所有中断

PUIE/AIEL: 外设中断使能位/低优先级中断使能位
当 IPEN=0

1 = 使能所有未屏蔽的外设中断

0 = 禁止所有外设中断

当 IPEN=1

1 = 允许所有低优先级的中断

0 = 禁止所有低优先级的中断

TOIE: T0 溢出中断使能位

1 = 使能 T0 中断

0 = 禁止 T0 中断

INT0IE: INT0 中断使能位

1 = 使能 INT0 中断

0 = 禁止 INT0 中断

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

INT0IF: INT0 中断标志位

1 = INT0/P0.3 产生外部中断

0 = INT0/P0.3 未产生外部中断

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

图注: R=可读 W=可写 -=未用 U=未实现位

5.1.2 中断使能寄存器 EIE1

寄存器5.2: EIE1: 中断使能寄存器(地址: 2CH)

复位值 0000 0000	bit7							bit0
	-	ADIE	INT2IE	INT1IE	-	PWM2IE	-	T1IE
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ADIE: AD 中断使能位

1 = 使能 AD 中断

0 = 禁止 AD 中断

INT2IE: INT2 中断使能位

1 = 使能 INT2 中断

0 = 禁止 INT2 中断

INT1IE: INT1 中断使能位

1 = 使能 INT1 中断

0 = 禁止 INT1 中断

PWM2IE: PWM2 中断使能位

1 = 使能 PWM2 中断

0 = 禁止 PWM2 中断

T1IE: T1 中断使能位

1 = 使能 T1 中断

0 = 禁止 T1 中断

图注: R=可读 W=可写 -=未用 U=未实现位

5.1.3 中断使能寄存器 EIE2

寄存器5.3: EIE2: 中断使能寄存器(地址: 2DH)

复位值 0000 0000	bit7							bit0
	-	-	-	-	-	-	BCLIE	SSCIE
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

图注: R=可读 W=可写 -=未用 U=未实现位

5.1.4 中断使能寄存器 EIE3

寄存器5.4: EIE3: 中断使能寄存器(地址: 4AH)

复位值 000- ----	bit7							bit0
	T4IE	-	OSCFAIL IE	-	-	-	-	-
	R/W	R/W	R/W	U	U	U	U	U

T4IE: T4 中断使能位

1 = 使能 T4 中断

0 = 禁止 T4 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

图注: R=可读 W=可写 -=未用 U=未实现位

5.1.5 中断标志寄存器 EIF1

寄存器5.5: EIF1: 外设中断标志寄存器(地址0CH)

复位值 0000 0000	bit7							bit0
	-	ADIF	INT2IF	INT1IF	-	PWM2IF	-	T1IF
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ADIF: AD 完成中断标志位

1 = AD 转换完成

0 = AD 转换没有完成

INT2F: INT2 中断标志位

1 = INT2/P1.3 产生外部中断

0 = INT2/P1.3 未产生外部中断

INT1IF: INT1 中断标志位

1 = INT1/P1.2 产生外部中断

0 = INT1/P1.2 未产生外部中断

PWM2IF: PWM2 中断标志位

1 = PWM2 使能时, T1H 和 PP2 匹配

0 = PWM2 使能时, T1H 和 PP2 不匹配

T1IF: T1 寄存器溢出标志位

1 = T1 寄存器溢出

0 = T1 寄存器未溢出

图注: R=可读 W=可写 -=未用 U=未实现位

5.1.6 中断标志寄存器 EIF2

寄存器5.6: EIF2: 外设中断标志寄存器(地址: 0DH)

复位值 0000 0000	bit7							bit0
	-	-	-	-	-	BCLIF	SSCIIF	
	R/W	R/W	R	R	R/W	R/W	R/W	R/W

BCLIF: BCL 中断标志位

1 = BCL 产生了中断

0 = BCL 未产生中断

SSCIIF: SSCI 中断标志位

1 = SSCI 产生了中断

0 = SSCI 未产生中断

图注: R=可读 W=可写 -=未用 U=未实现位

5.1.7 中断标志寄存器 EIF3

寄存器5.7: EIF3: 中断标志寄存器(地址: 4BH)

复位值 000- ----	bit7						bit0	
	T4IF	-	OSCFAIL IF	-	-	-	-	-
	R/W	R/W	R/W	U	U	U	U	U

T4IF: T4 寄存器溢出标志位

1 = T4 寄存器溢出

0 = T4 寄存器未溢出

OSCFAILIF: 外部时钟故障中断标志位

1 = 外部时钟发生故障

0 = 外部时钟未发生故障

图注: R=可读 W=可写 -=未用 U=未实现位

5.1.8 中断优先级控制寄存器 IP0

当 IPEN=1 时, 中断优先级控制寄存器 IP0 才有效。

寄存器5.8: IP0: 中断优先级控制寄存器0(地址: 22H)

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

PT0: T0 中断优先级控制位

1 = T0 中断为高优先级

0 = T0 中断为低优先级

PINT0: INT0 中断优先级控制位

1 = INT0 中断为高优先级

0 = INT0 中断为低优先级

PP0: P0 电平变化中断优先级控制位

1 = P0 电平变化中断为高优先级

0 = P0 电平变化中断为低优先级

图注: R=可读 W=可写 -=未用 U=未实现位

5.1.9 中断优先级控制寄存器 IP1

当 IPEN=1 时, 中断优先级控制寄存器 IP1 才有效。

寄存器5.9: IP1: 中断优先级控制寄存器1(地址: 23H)

复位值 0000 0000	bit7						bit0	
	-	PADC	PINT2	PINT1	-	PPWM2	-	PT1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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
- PPWM2: PWM2 中断高优先级控制位
1 = PWM2 中断为高优先级
0 = PWM2 中断为低优先级
- 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

图注：R=可读 W=可写 -=未用 U=未实现位

5.1.10 中断优先级控制寄存器 IP2

当 IPEN=1 时，中断优先级控制寄存器 IP2 才有效。

寄存器5.10: IP2: 中断优先级控制寄存器2(地址: 24H)

复位值 0000 0000	bit7						bit0	
	-	-	-	-	-	-	PBCL	PSSCI
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- 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

图注：R=可读 W=可写 -=未用 U=未实现位

5.1.11 中断优先级控制寄存器 IP3

当 IPEN=1 时，中断优先级控制寄存器 IP3 才有效。

寄存器5.11: IP3: 中断优先级控制寄存器3(地址: 29H)

复位值 000- ----	bit7						bit0	
	PT4	-	POSCFAI L	-	-	-	-	-
	R/W	R/W	R/W	U	U	U	U	U

- 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
- 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

图注：R=可读 W=可写 -=未用 U=未实现位

5.1.12 电源控制寄存器 PCTL

寄存器5.12:PCTL: 电源控制寄存器(地址:2EH)

	bit7						bit0	
复位值 ---1 000x	-	-	-	SLVREN	IPEN	SWDTEN	$\overline{\text{POR}}$	$\overline{\text{LVR}}$
	U	U	U	R/W	R/W	R/W	R/W	R/W

SLVREN: 软件欠压检测使能位

1 = 使能欠压检测
0 = 禁止欠压检测

IPEN: 中断优先级控制位

1 = 使能中断优先级功能, 即为优先级模式
0 = 禁止中断优先级, 即为普通模式

SWDTEN: 软件看门狗定时器使能位

当配置字的 WDTEN=0 时
1 = 软件使能看门狗定时器
0 = 软件禁止看门狗定时器
当配置字的 WDTEN=1 时, 为无关位

$\overline{\text{POR}}$: 上电复位状态位

1 = 未发生上电复位
0 = 发生了上电复位

$\overline{\text{LVR}}$: 欠压复位状态位

1 = 未发生欠压复位
0 = 已发生欠压复位

图注：R=可读 W=可写 -=未用 U=未实现位

5.1.13 INT 中断沿选择寄存器 INTEDGCTL

寄存器5.13:INTEDGCTL:INT中断沿选择寄存器 (67H)

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

INT2SE: INT2 触发脉冲边沿选择位

1 = 上升沿触发
0 = 下降沿触发

INT1SE: INT1 触发脉冲边沿选择位

1 = 上升沿触发
0 = 下降沿触

T1CLKEN: T1 定时模式时钟源选择位

当 T1CS=0 时:

1 = T1 时钟为内部高频时钟 INTHF

0 = T1 时钟为系统时钟 4 分频 SCLK/4

当 T1CS=1 时，为计数模式，T1CLKEN 位不起作用，T1 时钟为外部时钟

图注：R=可读 W=可写 -=未用 U=未实现位

5.1.14 中断响应

当 IPEN=1 时，为优先级模式，中断被响应后：

1. 返回地址压入堆栈；
2. 中断入口地址载入 PC；
3. 在中断被响应前，AIEH 和 AIEL 位的设置会影响到中断响应的过程：
 - 将 AIEH 和 AIEL 均置 1，可进入高优先级或优先级中断。进入的是高或低优先级中断时，AIEH 或 AIEL 由硬件自动清零，执行高或低优先级中断服务程序，执行指令 IRET 退出高或低优先级中断，硬件自动将 AIEH 或 AIEL 置 1；
 - 只将 AIEH 置 1，则直接进入高优先级中断，AIEH 由硬件自动清零，继续执行中断服务子程序，执行指令 IRET 退出中断服务子程序并由硬件自动将 AIEH 置 1，重新使能未屏蔽的中断；
4. 跳转到中断发生处继续执行下面的程序。

注： 中断的响应有一定的响应时间：

1. 如果已经进入了低优先级中断，再有高优先级中断，此时为中断嵌套；
2. 如果还没有进入低优先级中断，同时有高优先级中断产生，则直接进入高优先级中断，相当于两个中断同时产生。

当 IPEN=0 时，为普通模式，中断被响应后：

1. 返回地址压入堆栈；
2. 中断入口地址载入 PC；
3. AIE 位将被硬件清零以禁止其它中断；
4. 执行该中断服务子程序；
5. 执行指令 IRET 退出中断服务子程序，同时硬件自动将 AIE 置 1，重新使能未屏蔽的中断；
6. 跳转到中断发生处继续执行下面的程序。

进入中断服务程序后，首先保存 PSW 和其它寄存器的值，然后通过查询中断标志位确定中断源。在重新使能中断之前，应在软件中将相应的中断标志位清零，以避免出错。

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

5.2 INT 中断

INT 中断有三个中断源: INT0、INT1 和 INT2, 都采用边沿触发方式,如果触发边沿选择位(INTxSE)置 1, 则采用上升沿触发; 如果触发边沿选择位清零, 则采用下降沿触发。

5.2.1 INT0 中断

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

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

5.2.2 INT1 中断

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

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

5.2.3 INT2 中断

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

INT2 引脚有触发脉冲时, INT2IF 被自动置 1, 如果 INT2IE、PUIE 和 AIE 位为 1, 则响应 INT2 中断。

使用 INT 中断时的设置:

1. 将对应的 INTx 引脚设置为数字输入口。
2. 选择触发脉冲边沿时上升沿还是下降沿(INT0/1/2SE 置 1 为上升沿触发);
3. 将相应的外部中断使能位置 1(INxIE), 如果为高优先级, 则 IPEN 和 PINTx 均置 1。

注:x=0/1/2。

5.3 定时器中断

T0/1 寄存器发生溢出时，T0IF/T1IF 位将会被置 1。通过将 T0IE/T1IE 位置 1/清零可使得能/禁止该中断。有关定时/计数器模块中断的操作，请参考定时/计数器部分。

5.4 P0 口中断

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

5.5 PWM 中断

使能 PWM1x/2x 后，T1L 分配给 PWM1x 进行计数，T1H 分配给 PWM2x 进行计数，当 T1L/H 与 PP1/2 匹配时，会触发相应的中断标志位 T1IF 和 PWM2IF。如果使能 T1IE 或者 PWM2IE，则会触发中断（AIE、PUIE 置 1）。当 IPEN 和 PT1 均置 1 时，PWM1 中断配置为高优先级中断。当 IPEN 和 PPWM2 均置 1 时，PWM2 中断配置为高优先级中断。

详见 PWM1x/2x 部分。

5.6 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

K8V327 单片机提供一个 8 位的定时/计数器 T0、1 个 16 位的定时/计数器 T1 和 1 个 16 位的定时/计数器 T4。

6.1 定时器/计数器T0

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

6.1.1 T0 原理框图

图 6.1 为 T0 的结构框图。T0 模块使用一个 8 位计数器作为预分频器，如寄存器 6.1 所示，通过软件设定 PSA 位(OPTR.3)的状态可对预分频器的分配进行控制，PSA 位清零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。

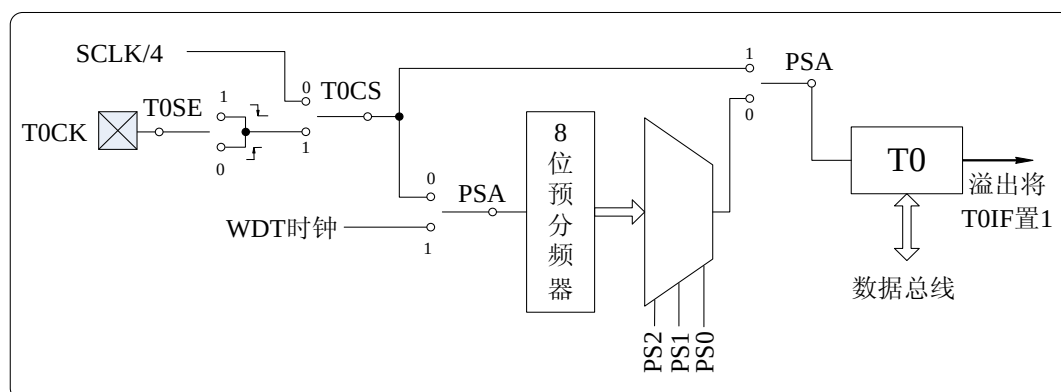


图 6.1 原理框图

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

PUPH P0 口上拉功能总使能位
1 = 禁止 P0 口所有上拉功能
0 = 允许 P0 口使用上拉功能

INT0SE INT0 中断触发脉冲边沿选择位
1 = INT0/P0.3 为上升沿触发
0 = INT0/P0.3 为下降沿触发

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

T0SE: T0 计数脉冲信号边沿选择位
1 = 下降沿触发
0 = 上升沿触发

PSA: 预分频器分配控制位
1 = 预分频器用于 WDT
0 = 预分频器用于 T0

PS<2:0>: 预分频器分频比选择位

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

图注: R=可读 W=可写 -=未用 U=未实现位

6.1.3 定时模式

通过将 T0CS 位(OPTR.5)清零可选择定时器模式。在定时模式中, 如果不使用预分频器, 每一个机器周期 T0 寄存器的值加 1。如果 T0 寄存器被写入初始值, 则在接下来的两个指令周期将不执行递增操作, 用户可通过将校正值写入 T0 寄存器进行修正。

6.1.4 计数模式

通过将 T0CS 位(OPTR.5)置 1 可选择计数模式。在该模式下, T0 模块在 T0CK 引脚信号的每一次上升沿(T0SE 位清零)或下降沿(T0SE 位置 1)递增计数。

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

6.1.5 T0 的使用

T0 在使用时通过以下步骤进行设置：

1. 通过将 T0CS 位清零/置 1 选择定时/计数模式(如果是计数模式，再设置 T0SE 选择脉冲触发边沿，将对应的计数脉冲输入脚 T0CK 设置为输入)；
2. 如果需要分频，则将预分频器分配给 T0，并设置分频比；
3. 给 T0 寄存器设置初始值；
4. 如果使用中断方式则将 T0IE 和 AIE 位置 1。

6.2 定时器/计数器 T1

T1 是一个 16 位的定时器/计数器, T1 的低 8 位在寄存器 T1L 中 高 8 位在寄存器 T1H 中, 当 T1 计数值达到 65535 后, T1 的值再加 1 就会产生溢出, 将 T1 中断标志位置 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 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 使预分频计数器清零。

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 计数器将清零重新开始计数, 且将 T1 中断标志位置 1。

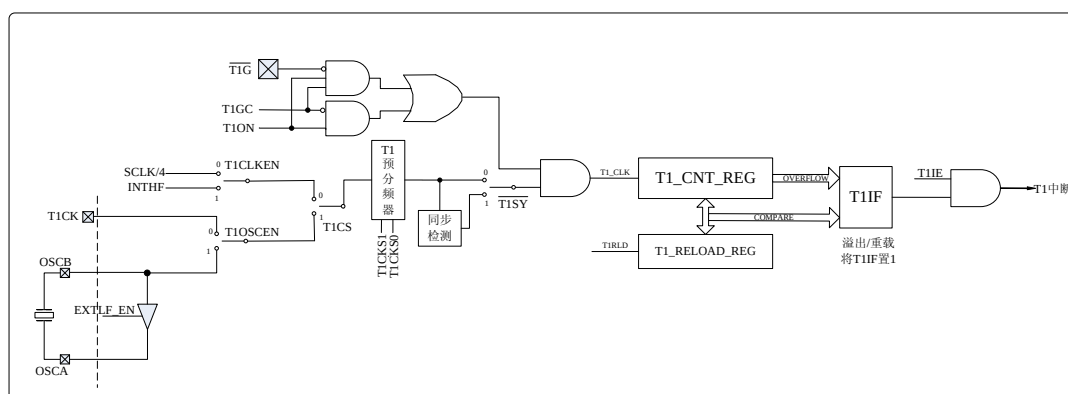


图 6.2 T1 原理框图

6.2.2 T1 时钟

当 T1CTL 寄存器的 T1CS 位置 1 时, T1 工作在计数模式下, T1 的工作时钟源通过 T1OSCEN 位来选择: 当 T1OSCEN=1 时, T1 时钟为外部低频时钟; 当 T1OSCEN=0 时, T1 时钟为 T1CK; 当 T1CTL 寄存器的 T1CS 位清零时, T1 工作在定时模式下, T1 的工作时钟源通过 INTEDGCTL 寄存器的 T1CLKEN 位来选择: 当 T1CLKEN=1 时, T1 时钟为内部高频振荡器时钟 INT1F; 当 T1CLKEN=0 时, T1 时钟为指令时钟 SCLK/4。

6.2.3 T1 相关的寄存器

表 6-2 与 T1 相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
0EH	T1L	T1 低 8 位							
0FH	T1H	T1 高 8 位							
10H	T1CTL	T1RLD	T1GC	T1CKS1	T1CKS0	T1OSCEN	$\overline{T1SY}$	T1CS	T1ON

6.2.3.1 T1 控制寄存器

如寄存器 6.2 所示，T1 控制寄存器（T1CTL）用于启动/禁止 T1 以及选择 T1 模块的不同功能特性。

寄存器6.2: T1CTL: T1控制寄存器(地址: 10H)

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

T1RLD: T1 重载功能使能位

0 = 禁止 T1 重载功能

1 = 使能 T1 重载功能

T1GC: T1 门控使能位

如果 T1ON=0 则该位被忽略

如果 T1ON=1 则:

1 = 使能 $\overline{T1G}$ 引脚控制(如果 $\overline{T1G}$ 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，T1 时钟可选择内部高频时钟或指令时钟

当 T1CLKEN=1 时 T1 时钟为内部高频振荡器时钟

当 T1CLKEN=0 时 T1 时钟为机器时钟 SCLK/4

T1ON: T1 启动控制位

1 = 启动 T1

0 = 停止 T1

图注: R=可读 W=可写 -=未用 U=未实现位

6.2.4 定时模式

通过将 T1CS 位清零将 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, 当计数时钟选择为内部时钟且不使用预分频器时, 每个时钟周期 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。

6.2.5 计数模式

通过将 T1CS 位置 1 将 T1 设定为计数模式, 通过设置 T1OSCEN 位选择计数时钟为 T1CK 或者外部低频时钟。当 T1OSCEN 为 0 时, T1 在计数脉冲 T1CK 的上升沿进行递增计数, 当 T1OSCEN 为 1 时, 每个外部低频时钟周期, T1 寄存器自动加 1。

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(当 T1OSCEN=0 时)引脚的脉冲进行递增计数。在休眠模式下, 计数器将继续递增并在溢出时产生中断以唤醒处理器。

如果控制位 $\overline{\text{T1SY}}$ (T1CTL.2) 清零, 则 T1 工作在同步计数模式。在内部相位时钟的 Q2 和 Q4 周期对 T1CK 引脚电平进行采样, 可以实现 T1CK 与内部相位时钟的同步。

6.2.6 T1 重载功能

置位 T1CTL 寄存器的 T1RLD 位使能 T1 重载功能。通过设置 PP2/PP1 寄存器来设置 T1 的重载点。当设置完 PP2/PP1 寄存器且使能重载功能后, T1 计数器从 0 开始计数, 当计数到 PP2/PP1 寄存器设置的值时, T1 计数器清零并重新开始计数, T1IF 置 1。

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

6.2.7 T1 在休眠模式下的运行

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。

6.2.8 T1 分配给 PWM1/2

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

6.3 定时/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 帶有重載功能。當使能 T4 重載功能時，T4 計數器計數到 T4 重載寄存器中設置的值時，T4 計數器將清零重新開始計數，且將 T4 中斷標志位置 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。

6.3.1 T4 原理框图

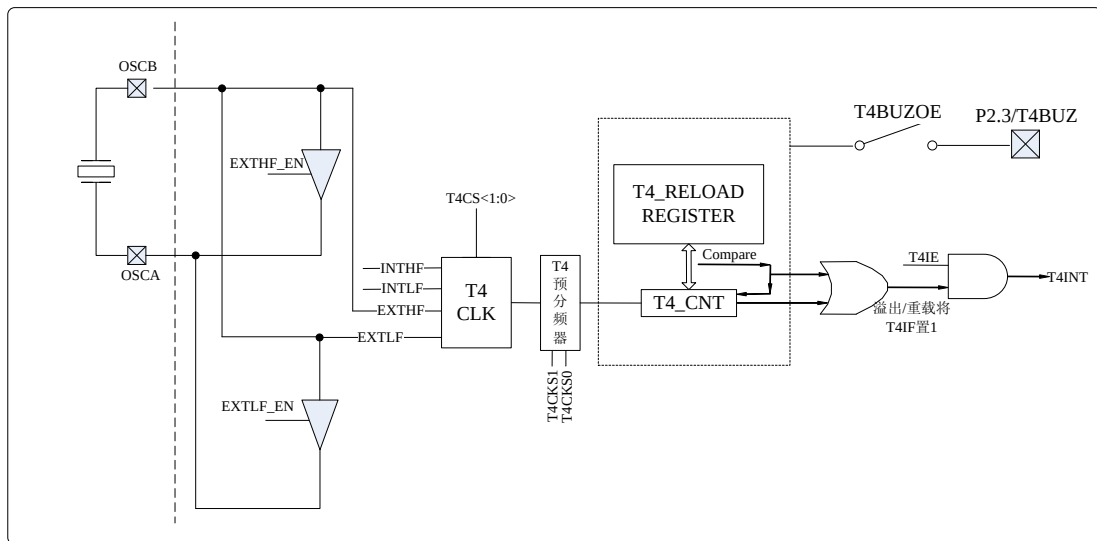


图 6.3 T4 原理框图

6.3.2 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	HSPEN	T4CKS1	T4CKS0	LSPEN	T4CS1	T4CS0	T4ON
161H	T4H	T4 高 8 位							
160H	T4L	T4 低 8 位							
163H	T4REH	T4 重载寄存器高 8 位							
162H	T4REL	T4 重载寄存器低 8 位							

6.3.2.1 T4 控制寄存器

如寄存器 6-1 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 0000	T4REN	HSPEN	T4CKS1	T4CKS0	LSPEN	T4CS1	T4CS0	T4ON
	R/W	R/W	R/W	R/W	R/W	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
HSPEN:	外部高频时钟工作使能位 0 = 禁止 T4 通过外部高频时钟工作 1 = 允许 T4 通过外部高频时钟工作
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
LSPEN:	外部低频时钟工作使能位 0 = 禁止 T4 通过外部低频时钟工作 1 = 允许 T4 通过外部低频时钟工作
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 EXT LF
T4ON:	T4 启动控制位 1 = 启动 T1 0 = 停止 T1

图注：R=可读 W=可写 --未用 U=未实现位

6.3.2.2 T4 预分频器

如寄存器 6.1 所示，T1 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。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 使预分频计数器清零。

6.3.2.3 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，K8V327 系列提供 4 个时钟源，分别为内部高频时钟、内部低频时钟、外部高频时钟以及外部低频时钟。

6.3.3 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 计数器清零并重新开始计数，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。

6.3.4 T4 蜂鸣器功能

定时器 T4 带有蜂鸣器功能。要使用蜂鸣器，需要打开 T4 的重载功能及 BUZCTL（如寄存器 6.7 所示）寄存器的 T4_BUZOE 位。

通过设置 T4REH/T4REL 寄存器设置 T4 的重载点，并打开使能为 T4REN，再将 BUZCTL 寄存器的 T4_BUZOE 位置 1，蜂鸣器将会在 P2.3 口输出蜂鸣器功能波形，如图 6.4 所示。T4 蜂鸣器的周期如式 6.1 所示。

寄存器6.7: BUZCTL: 蜂鸣器控制寄存器(地址: 4EH)

复位值 ---- -0--	bit7							bit0
	-	-	-	-	-	T4_BUZOE	-	-
	U	U	U	U	U	R/W	U	U

T4_BUZOE: T4 蜂鸣器控制输出使能位

0 = 禁止 T4 蜂鸣器控制输出

1 = 使能 T4 蜂鸣器控制输出

图注: R=可读 W=可写 -=未用 U=未实现位

★ 式6.1: $T4\text{蜂鸣器周期} = 2 \cdot (T4REH:T4REL) \cdot T4\text{计数周期}$

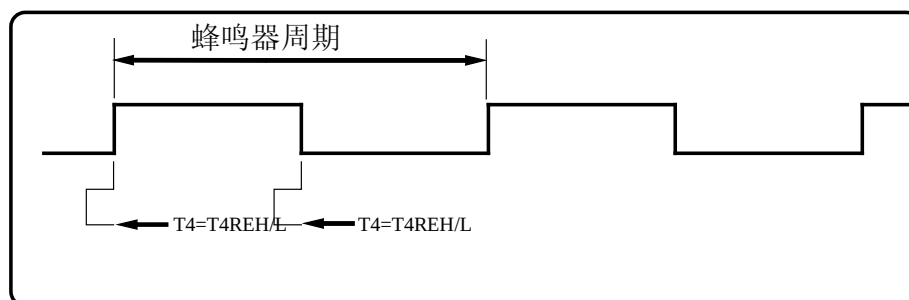


图 6.4 T4 蜂鸣器功能波形

6.3.5 T4 中断

在两种情况下会使 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。

6.3.6 T4 工作在休眠模式

当选择外部时钟信号作为 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。

7 模数(A/D)转换模块

模数(A/D)转换模块可将模拟输入信号转换为 12 位二进制值。K8V327 拥有 14 路 IO 模拟输入通道。转换器通过逐次逼近法将模拟输入信号转换为二进制值，并将转换结果存放到 12 位寄存器中。可通过软件方式选择 VDD 或施加在 ADVRIN 引脚上的电压作为转换使用的参考电压。

图 7.1 显示了 K8V327 中 A/D 转换模块的结构框图。

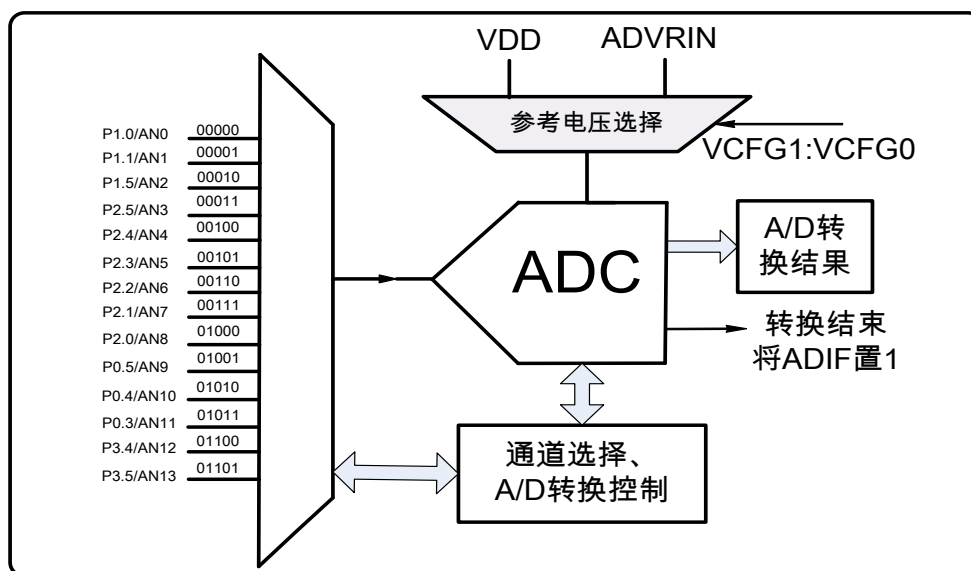


图 7.1 AD 模块结构框图

7.1 与 AD 相关的寄存器

表 7-1 与 AD 转换相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
1FH	ADCCTL0	ADLR	-	CHS3	CHS2	CHS1	CHS0	START	ADEN
3FH	ADCCTL1	ADCALE N	ADCS2	ADCS1	ADCS0	VCFG1	VCFG0	-	ADCIM
31H	ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
1DH	ANSEH	-	-	ANS13	ANS12	ANS11	ANS10	ANS9	ANS8
23H	IP1	-	PADC	PINT2	PINT1	-	PPWM2	-	PT1
58H	ADCINTCTL	-	-	-	-	-	-	INTCTL1	INTCTL0
1EH	ADCDATA0H	AD 数据寄存器 0 高 8 位							
3EH	ADCDATA0L	AD 数据寄存器 0 低 8 位							

7.1.1 AD 控制寄存器 0(ADCCTL0)

寄存器7.1: ADCCTL0: A/D控制寄存器0(地址: 1FH)

复位值 0000 0000	bit7						bit0	
	ADLR	-	CHS3	CHS2	CHS1	CHS0	ADSTART	ADEN
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ADLR: A/D 转换结果输出格式选择位

1 = 结果右对齐

0 = 结果左对齐

CHS<3:0>: 模拟通道选择位

0000 = 通道 00(AN0)

0001 = 通道 01(AN1)

0010 = 通道 02(AN2)

0011 = 通道 03(AN3)

0100 = 通道 04(AN4)

0101 = 通道 05(AN5)

0110 = 通道 06(AN6)

0111 = 通道 07(AN7)

1000 = 通道 08(AN8)

1001 = 通道 09(AN9)

1010 = 通道 10(AN10)

1011 = 通道 11(AN11)

1100 = 通道 12(AN12)

1101 = 通道 13(AN13)

其他 = 系统保留

START: A/D 转换状态位

1 = A/D 转换正在进行,该位置 1 将启动 A/D 转换, 在转换结束后该位将被硬件自动清零。

0 = A/D 转换结束或者未进行

ADEN: A/D 工作使能位

1 = 使能 A/D 转换模块工作

0 = A/D 转换器关闭且不消耗工作电流

图注: R=可读 W=可写 -=未用 U=未实现位

7.1.2 AD 控制寄存器 1(ADCCTL1)

寄存器7.2: ADCCTL1: A/D控制寄存器1(地址: 3FH)

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

ADCALEN: AD 校准使能位

0 = 关闭 AD 校准

1 = 打开 AD 校准

ADCS<2:0>: A/D 转换时钟选择位

000 = $F_{ad} = SCLK/2$

001 = $F_{ad} = SCLK/8$

010 = $F_{ad} = SCLK/32$

x11= 系统保留

100 = $F_{ad} = SCLK/4$

101 = $F_{ad} = SCLK/16$

110 = $F_{ad} = SCLK/64$

VCFG<1:0>: A/D 转换参考电压选择位

00= 断开

01 = VDD 作为 ADC 参考电压

10 = ADV_{RIN} 作为 ADC 参考电压

11 = 保留

ADCI_M: AD 工作电流选择位

0 = A/D 工作电流较大

1 = A/D 工作电流较小（建议配置为 1，此时工作电流最小）

图注：R=可读 W=可写 -=未用 U=未实现位

7.1.3 模拟/数字口选择寄存器(ANSEL/H)

寄存器7.3: ANSEH: 模拟/数字口设置寄存器(地址: 1DH)

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

寄存器7.4: ANSEL: 模拟/数字口设置寄存器(地址: 31H)

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

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

1 = 将对应引脚配置为模拟口

0 = 将对应引脚配置为数字 I/O 口或者特殊功能引脚

图注：R=可读 W=可写 -=未用 U=未实现位

7.2 通道的选择

如图 7.1 所示，KF8V327 中的 A/D 转换模块的输入可以选择 14 路来自外部的模拟信号。通过寄存器 ADCCTL0(如寄存器 7.1 所示)进行通道的选择。

7.3 模拟输入口的配置

当选择 AN0~AN13 作为 A/D 转换的输入时，需要将对应的引脚配置为模拟输入口。通过将寄存器 ANSEL 的某位置 1 将对应的引脚配置为模拟口，然后把寄存器 TR1/2/3 的对应位置 1 把该引脚配置为输入口，此时该引脚被设置为模拟输入口。

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

7.4 A/D 转换参考电压的选择

K8V327 中 ADC 模块的参考电压可以选择 2 种分别为:电源电压(VDD)和外部参考电压(ADVRIN)。通过寄存器 ADCCTL1(如寄存器 2 所示)的 VCFG<1:0>设置参考电压。

7.5 转换时钟的选择

完成一次 A/D 转换所需要的时间为 $13T_{ad}$ 。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，共有 7 种时钟选项。 T_{ad} 和 F_{ad} 分别为 A/D 转换时钟周期和频率。

完成一次 AD 转换所需时间最快选择 SCLK/2，为 2us。

7.6 输出格式

K8V327 中 A/D 转换的结果为 12 位二进制数，A/D 转换结果寄存器为两个 8 位的寄存器。用户可以通过 ADLR(ADCCTL0.7)设置转换结果输出格式，ADLR 置 1 输出为右对齐，ADLR 清零输出为左对齐。如图 7.2 所示。

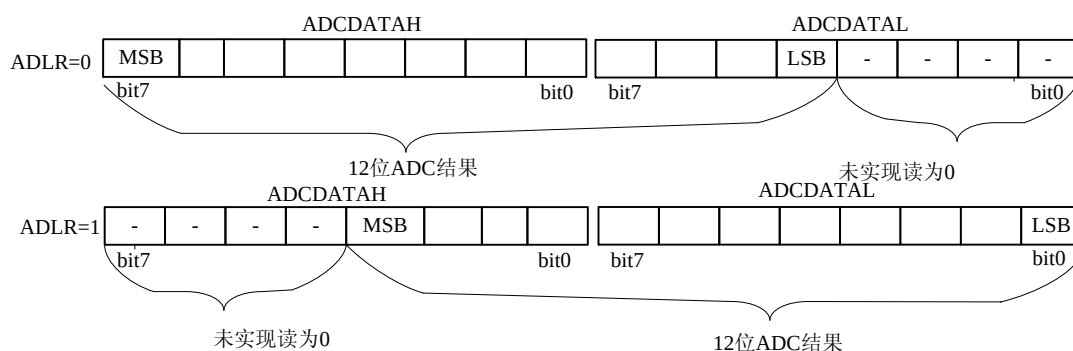


图 7.2 ADC 结构对其方式

7.7 A/D 转换的启动和完成

先将 ADEN 位置 1，然后将 START 位(ADCCR0.1)置 1 即可启动 A/D 转换。当转换结

束时，A/D 模块将：

1. 将 START 位清零
2. 将 ADIF 位置 1
3. 如果使能 A/D 转换中断，则响应中断

可以采取在程序中将 START 位清零的方法中止当前的转换操作。在 A/D 转换采样全部结束之前，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。A/D 转换被中止后，需至少等待 2Tad 的延时时间后才能开始下一次数据采集。

7.8 A/D 工作在休眠模式

A/D 转换器模块可以在休眠模式下工作。这需把 AD 的时钟源设定为 A/D 专用的内部振荡器（即 ADCS<2:0>=x11 时）。当选择了专用内部时钟源后，A/D 需等待一个指令周期后才能启动转换操作，转换结束后，START 位将被清零，且转换结果将被载入 ADCDATA0H: ADCDATA0L 寄存器。如果 A/D 中断被使能，器件将从休眠状态唤醒。如果 A/D 中断被禁止，A/D 转换模块在转换完成后被关闭。

如果 A/D 时钟源不是 AD 专用内部振荡器，执行 IDLE 指令将导致当前转换操作中止，并使 A/D 模块关闭。

7.9 复位的影响

器件复位将强制所有寄存器进入复位状态。因此，A/D 模块将被关闭，任何进行中的转换操作被中止。ADCDATA0H:ADCDATA0L 寄存器中的值不变。

7.10 使用 A/D 转换器的设置

启动 A/D 转换器时的设置：

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

8 PWM1x/2x模块

KF8V327 单片机具有 4 路 8 位的 PWM 模块 PWM11/PWM12/PWM21/PWM22。四路 PWM 结构均相同。

8.1 PWM1x/PWM2x 模块

启动 PWM 后，在对应的 PWM1x(或 PWM2x)引脚输出 PWM 脉冲。PWM 脉冲的频率和占空比通过 PP1(或 PP2)和 PWM1xL(或 PWM2xL)设置。

图 8.1 显示了 PWM 逻辑框图。其中 PP1 为 PWM1x 模块的周期寄存器，PWM1xL 为 PWM1x 模块占空比设置寄存器，使用 PWM 时需要将定时器 T1 配置给 PWM 做定时用，其中 T1L、T1IE 和 T1IF 分配给 PWM1x，T1H 分配给 PWM2x。启动 PWM1x 后，当 T1L 计数值和 PP1 相等时，P3.2 引脚被置 1，此时 T1L 被清零，重新开始计数，当 T1L 的计数值和 PWM1xL 相等时，P3.2 引脚清零(如图 8.2 所示)。改变 PP1 和 PWM1xL 的值可产生不同的 PWM1x 周期和 PWM1x 占空比。PWM2x 模块的工作原理和 PWM1x 模块完全一致。

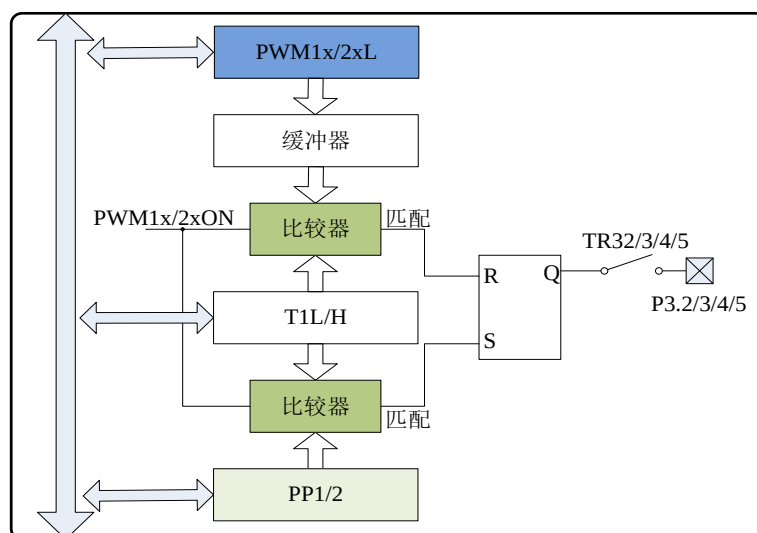


图 8.1 PWM1x/2x 逻辑框图

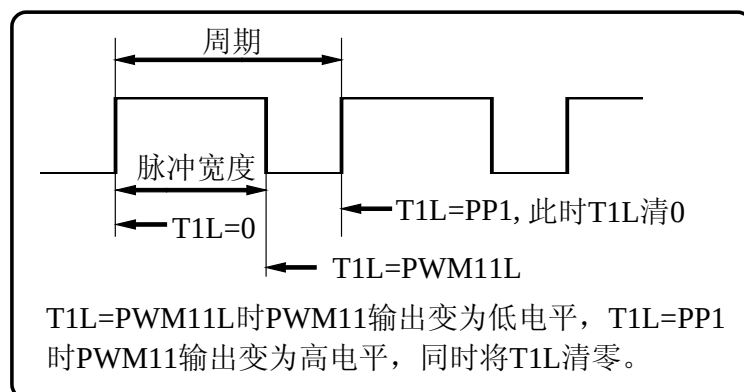


图 8.2 PWM11 输出波形图

8.1.1 PWM1x/2x 相关的寄存器

表 8-1 与 PWM 相关的寄存器

地址	寄存器	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
62H	PWM11L	PWM11 占空比寄存器							
63H	PWM12L	PWM12 占空比寄存器							
16H	PP1	PWM1x 周期寄存器							
32H	PP2	PWM2x 周期寄存器							
65H	PWM21L	PWM21 占空比寄存器							
66H	PWM22L	PWM22 占空比寄存器							
15H	PWMCTL	-	PWM22O N	PWM21O N	-	-	PWM12O N	PWM11ON	-

8.1.2 PWM1x/2x 控制寄存器

寄存器8.1: PWMCTL: PWM启动控制寄存器(地址: 15H)

复位值	bit7				bit0			
0000 0000	保留	PWM22O N	PWM21O N	保留	保留	PWM12O N	PWM11O N	保留
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- PWM22ON: PWM22 启动控制位
1 = 启动 PWM22
0 = 禁止 PWM22
- PWM21ON: PWM21 启动控制位
1 = 启动 PWM21
0 = 禁止 PWM21
- PWM12ON: PWM12 启动控制位
1 = 启动 PWM12
0 = 禁止 PWM12
- PWM11ON: PWM11 启动控制位
1 = 启动 PWM11
0 = 禁止 PWM11

图注: R=可读 W=可写 -=未用 U=未实现位

注: 保留位只能写 0, 不能写 1。

8.1.3 PWM1x/2x 周期

PWM 周期通过 PP1/2 (地址: 16H/32H)进行设置, PP1/2 是一个 8 位的寄存器, 其值可设置为 0~255。PWM 周期通过式 8.1 进行计算。

★ 式8.1:
$$\text{PWM周期} = (\text{PPx} + 1) \cdot 4 \cdot \text{Tsys} \cdot (\text{T1预分频比}) \quad (\text{x}=1、2)$$

8.1.4 PWM1x/2x 占空比

PWM 占空比通过 PWM1x/2xL(地址: 62H/63H/65H/66H)设置, 可写入一个 8 位的值到 PWM1x/2xL 来设置占空比。脉冲宽度和占空比通过式 8.2 和式 8.3 计算:

$$\star \text{ 式8.2: } \text{脉冲宽度} = \text{PWMxL} \cdot 4 \cdot \text{Tsys} \cdot (\text{T1预分频比}) \quad (x=11、12、21、22)$$

$$\star \text{ 式8.3: } \text{占空比} = \frac{\text{脉冲宽度}}{\text{PWM周期}} = \frac{\text{PWMxL}}{\text{PPx}+1} \quad (x=11、12、21、22)$$

8.1.5 PWM1x/2x 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。KF8V327 中当 PP1/2 为 255 时, PWM 的最大分辨率为 8 位。分辨率的计算公式如式 8.4 所示。

$$\star \text{ 式8.4: } \text{分辨率} = \frac{\text{Log}[(\text{PPx}+1)]}{\text{log2}} \text{ 位} \quad (x=1、2)$$

8.1.6 PWM1x/2x 中断

PWM2x 有一个专门的中断使能位 PWM2IE 和中断标志位 PWM2IF, 当启动 PWM2 后, 会对 PWM2IF 产生影响, 而 PWM1x 和定时器 T1 共用中断使能位 T1IE 和中断标志位 T1IF。

在 PWM1x/2x 启用后, 当 T1L/H 的计数值与 PWM1xL/2xL 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 清零, 将 T1IF/PWM2IF 置 1, 如果允许 T1 或 PWM2 中断, 将会转入对应的中断子程序中。

8.1.7 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。PWM1/2 输出引脚电平保持不变(如果输出为高电平, 则保持高电平, 如果为低电平保持低电平)。当器件被唤醒时, T1 将从原来的状态继续工作。

8.1.8 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/2x 使用的寄存器进入其复位状态。

8.2 PWM1x/2x 使用方法

PWM1/2 工作的设置应按照以下步骤:

1. 将 TR32 或 TR33 或 TR34 或 TR35 置 1, 禁止引脚 P3.2/PWM11 或 P3.4/PWM12 或 P3.3/PWM21 或 P3.5/PWM22 的输出驱动器。
2. 赋 PP1 或 PP2 寄存器的初值以设置 PWM1x 或 PWM2x 的 PWM 周期。
3. 赋 PWM1xL 或 PWM2xL 寄存器的初值以设置 PWM1x 或 PWM2x 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 清零;
 - 将 T1CTL 寄存器的 T1ON 位置 1 以启动 T1 (注: 此时需将 T1CTL 寄存器的 T1RLD 位清零, 禁止 T1 重载功能)。
5. 将 PWMCTL 寄存器的 PWM1xON 或 PWM2xON 置 1 以启动 PWM1x 或 PWM2x。
6. 将 TR32 或 TR33 或 TR34 或 TR35 清零使能引脚 P3.2/PWM11 或 P3.4/PWM12 或 P3.3/PWM21 或 P3.5/PWM22 的输出驱动器。

9 SSCI模块

9.1 概述

KF8V327 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

9.2 SSCI相关寄存器

表 9-1 与 SSCI 相关的寄存器

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
128H	SSCICTL0	SSCIWCFL	SSCIOV	SSCIEN	SSCICKP	SSCIMOD3	SSCIMOD2	SSCIMOD1	SSCIMOD0
12AH	SSCICTL1	SSCICALLE N	SSCIACKSTA A	SSCIACKDA T	SSCIACKEN	SSCIRCEN	STOPEN	RESTARTEN N	STARTEN
12BH	SSCISTA	SAMPLE	CKEGE	SSCIDA	SSCIP	SSCISTOP	SSCISTART	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

9.2.1 SSCI 控制寄存器 0（SSCICTL0）

寄存器9.1: SSCICTL0: SSCI控制寄存器0(地址:128H)

复位值 0000 0000	bit7				bit0			
	SSCIWCFL	SSCIOV	SSCIEN	SSCICKP	SSCIMOD3	SSCIMOD2	SSCIMOD1	SSCIMOD0
	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 主控模式, 时钟 = SCLK/4

0001 = SPI 主控模式, 时钟 = SCLK/16

0010 = SPI 主控模式, 时钟 = SCLK/64

0011 = SPI 主控模式, 时钟 = T2输出/2

0100 = SPI 从动模式, 时钟 = SCK 引脚。使能 $\overline{SS}$ 引脚控制。

0101 = SPI 从动模式, 时钟 = SCK 引脚。禁止 $\overline{SS}$ 引脚控制。 $\overline{SS}$ 可作为I/O 引脚使用。

0110 = I2C从动模式, 7 位地址

0111 = I2C从动模式, 10 位地址

1000 = I2C主控模式, 时钟=SCLK/(4*(SSCIADD+1))

1001 = 禁止装载功能

1010 = 保留

1011 = I2C固件控制主控模式(从动空闲模式)

1100 = 保留

1101 = 保留

1110 = I2C从动模式, 7 位地址, 并允许启动位和停止位中断

1111 = I2C 从动模式, 10 位地址, 并允许启动位和停止位中断

图注: R=可读 W=可写 --未用 U=未实现位

9.2.2 SSCI 控制寄存器 1 (SSCICTL1)

寄存器9.2: SSCICTL1: SSCI控制寄存器1(地址:12AH)

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

在发送模式下:

1 = 未接收到来自从器件的应答。

0 = 已接收到来自从器件的应答

SSCIACKDAT: 应答数据位（仅限于I2C主控模式）

在接收模式下: 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）

在接收模式下:

1 = 在SDA 和SCL 引脚启动应答序列，发送SSCIACKDAT 数据位。由硬件自动清零

0 = 应答序列空闲

SSCIRCEN: 接收使能位（仅限I2C主控模式）

1 = 使能I2C接收模式

0 = 接收空闲

STOPEN: 停止条件使能位（仅限 I2C 主控模式）

SCK 释放控制:

1 = 在SDA 和SCL 引脚启动停止条件。由硬件自动清零。

0 = 停止条件空闲

RESTARTEN: 重复启动条件使能位（仅限I2C主控模式）

1 = 在SDA 和SCL 引脚启动重复启动条件。由硬件自动清零。

0 = 重复启动条件空闲

STARTEN: 启动条件使能位（仅限I2C主控模式）

在发送模式下:

1 = 在SDA 和SCL引脚启动条件。由硬件自动清零。

0 = 启动条件空闲

图注: R=可读 W=可写 -=未用 U=未实现位

注: 对于SSCIACKEN、SSCIRCEN、STOPEN、RESTARTEN 和STARTEN 位: 如果I2C模块不处在空闲模式，此位可能无法被置1（没有假脱机（spooling））且可能无法对SSCIBUFR 进行写操作（禁止写SSCIBUFR）。

9.2.3 SSCI 状态寄存器（SSCISTA）

寄存器9.3: 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	R	R	R	R

SAMPLE: SPI数据输入采样相位

SPI 主控模式:

1 = 在数据输出时间结束时采样输入数据

0 = 在数据输出时间中间采样输入数据

SPI 从动模式:

当SPI 用于从动模式时, 必须将SAMPLE 清零

I2C模式:

此位必须保持清零

CKEGE: SPI 时钟边沿选择位

SPI模式, SSCICKP = 0:

1 = 在SCK 的下降沿发送数据

0 = 在SCK 的上升沿发送数据

SPI 模式, SSCICKP = 1:

1 = 在SCK 的上升沿发送数据

0 = 在SCK 的下降沿发送数据

I2C模式:

此位必须保持清零

SSCIDA: 数据/ 地址位 (I2C模式)

1 = 表示上次接收或发送的字节是数据

0 = 表示上次接收或发送的字节是地址

SSCISTOP: 停止位 (仅I2C模式)

当禁止SSCI 模块或上次检测到启动位时, 该位被清零。

SSCIEN 被清零。

1 = 表示上次检测到了停止位 (此位在复位时为0)

0 = 表示上次没有检测到停止位

SSCISTART: 启动位 (仅I2C模式)

当禁止SSCI 模块或上次检测到停止位时, 该位被清零。

SSCIEN 被清零。

1 = 表示上次检测到了启动位 (此位在复位时为0)

0 = 表示上次没有检测到启动位

SSCIRW: 读/ 写信息位 (仅I2C模式)

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

I2C主模式下

1 = 读

0 = 写

I2C从模式下

1 = 发送正在进行

0 = 发送未进行

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

SSCIUA: 更新地址位 (仅10位I2C模式)

1 = 表示用户需要更新SSCIADD 寄存器中的地址

0 = 不需要更新地址

SSCIBUF: 缓冲器满状态位

接收 (SPI和I2C模式):

1 = 接收完成, SSCIBUFR满

0 = 接收未完成, SSCIBUFR空
发送 (仅I2C模式):
1 = 正在发送, SSCIBUFR满
0 = 发送完成, SSCIBUFR空

图注: R=可读 W=可写 -=未用 U=未实现位

9.2.4 SSCI 屏蔽寄存器 (SSCIMSK)

寄存器9.4: SSCIMSK: SSCI屏蔽寄存器(地址:12EH)

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

图注: R=可读 W=可写 -=未用 U=未实现位

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。

9.2.5 SSCI I2C 地址寄存器 (SSCIADD)

在10位I2C从动模式下, 该地址寄存器是复用的。

寄存器9.5: SSCIADD: I2C地址寄存器(地址:12EH)

	bit7							bit0
复位值 0000 0000	SSCIADD 7	SSCIADD 6	SSCIADD 5	SSCIADD 4	SSCIADD 3	SSCIADD 2	SSCIADD 1	SSCIADD 0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

10位从动模式下——高地址字节:

SSCIADD<7:3>: 未使用, SSCIADD存放高地址字节时, 未使用这5位, 为无关位。主器件发送的位模式由I2C规范制定必须等于11110, 但是这这些位由硬件进行比较且不受该寄存器中的值的影响

SSCIADD<2:1>: 保存10位地址的高两位

SSCIADD0: 未使用, 为无关位, 初始化时写0

10位从动模式下——低地址字节:

SSCIADD<7:0>: 10位地址的低8位

7位从动模式下:

SSCIADD<7:1>: 7位地址

SSCIADD0: 未使用，为无关位，初始化时写0

图注：R=可读 W=可写 -=未用 U=未实现位

注：(1)在I2C主控模式下，波特率计算公式= $SCLK/(4*(SSCIADD+1))$;
(2)在I2C模式下，不支持SSCIADD寄存器的值为0、1或2的情况。

9.3 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

9.3.1 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。

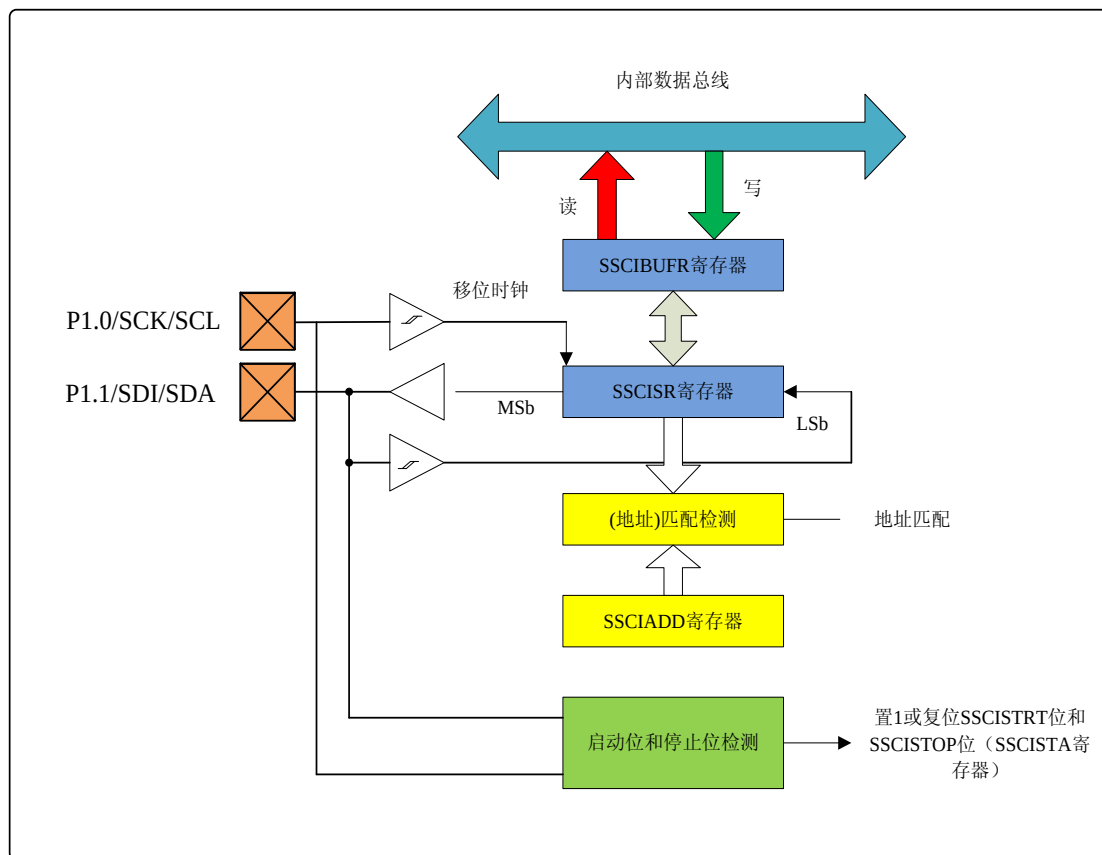


图9.1 I2C模式方框图

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 (SSCICTL0<3:0>) 选择以下I2C 模式之一:

- I2C 从动模式 (7 位地址)
- I2C 从动模式 (10 位地址)
- I2C 从动模式 (7 位地址), 允许启动位和停止位中断以支持固件主控模式
- I2C 从动模式 (10 位地址), 允许启动位和停止位中断以支持固件主控模式
- 允许I2C 启动位和停止位中断以支持固件主控模式而从动模式空闲

任何I2C 模式的选择, 在SSCIEN置1后都会强制SCL和SDA引脚为漏极开路 (假定通过编程将相应的TR1位置1, 使这些引脚成为输入引脚)。必须在SCL和SDA引脚上外接上拉电阻, 才能使I2C模块正常工作。

9.3.2 I2C 从动模式

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

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

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

- 1) 在接收到数据前，缓冲器满标志位SSCIBUF（SSCISTA<0>）置1。
- 2) 在接收到数据前，溢出标志位SSCIOV（SSCICTL0<6>）置1。

在这些情况下，SSCISR寄存器的值不会载入SSCIBUFR，但是SSCIIF位会置1。表9-2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 9-2 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。

9.3.2.1 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9.3）。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。

9.3.2.2 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。

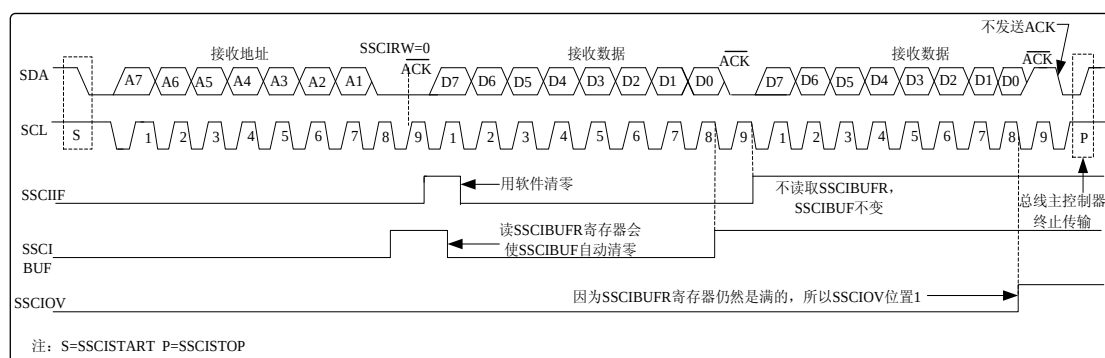


图9.2 从动模式时序（接收，7位地址）

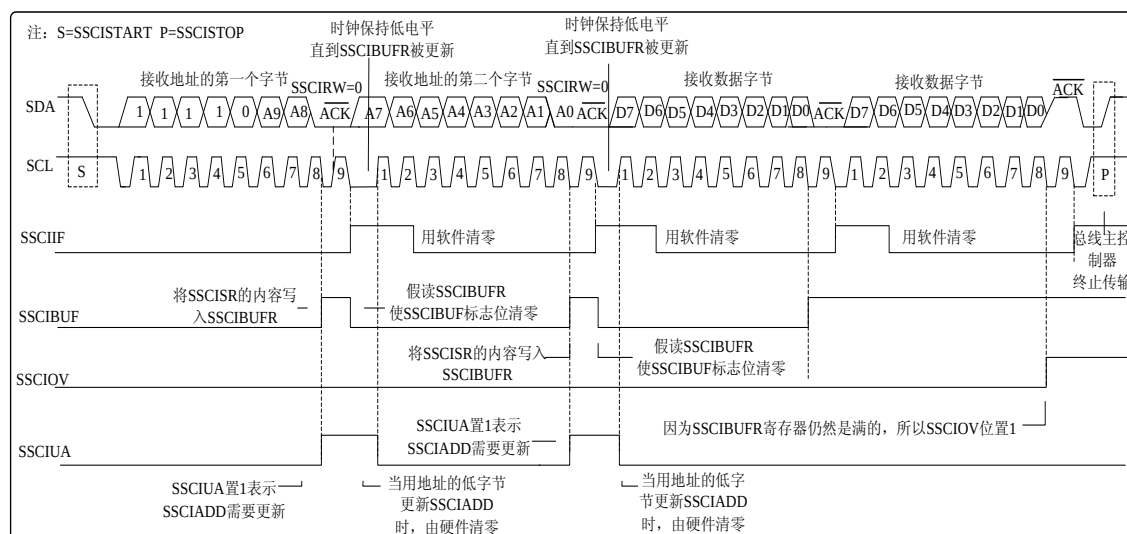


图 9.3 从动模式时序（接收，10 位地址）

从动接收设置:

1. 通过SSCIMOD<3:0>位选择I2C工作模式
 - 0110 = I2C 从动模式，7位地址
 - 0111 = I2C 从动模式，10位地址
 - 1011 = I2C 固件控制主控模式（从动空闲模式）
 - 1110 = I2C 从动模式，7位地址，并允许启动位和停止位中断
 - 1111 = I2C 从动模式，10位地址，并允许启动位和停止位中断
2. 设置SSCIADD寄存器，设置从机地址，仅高七位有效；
3. 清零SSCISTA寄存器的各标志，包括SSCIDA、SSCIRW、SSCIBUF等。
4. 设置SDA引脚为输入，SCL为输入；
5. 清零SSCIIF标志，如果需要中断打开各终端使能位；
6. 使能SSCIEN，开始接收数据，等待地址匹配；如果地址匹配，则SSCISTA寄存器的SSCIRW位清零。SSCISR寄存器的值被装入SSCIBUFR寄存器；
7. 缓冲器满标志位SSCIBUF被置1；产生 $\overline{\text{ACK}}$ 脉冲信号；在第9个SCL脉冲的下降沿，SSCI中断标志位SSCIIF被置1，软件清零。

9.3.2.3 发送

当输入地址字节的SSCIRW位置1 并发生地址匹配时，SSCISTA寄存器的SSCIRW位被置1。接收到的地址被装入SSCIBUFR寄存器。ACK脉冲在第9位上发送，SCL引脚保持低电平。发送数据必须被装入SSCIBUFR寄存器，同时也装入SSCISR寄存器。然后，应该通过将SSCICKP位（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个时钟脉冲的下降沿被置1。对于从发送器，来自主接收器的ACK脉冲将在第9个SCL输入脉冲的上升沿被锁存。若SDA线为高电平（无ACK应答信号），则表示数据传输已完成。在这种情况下，如果从控制器锁存了ACK，将复位从动逻辑（复位SSCISTA寄存器），同时从控制器监视下一个启动位的出现。如果SDA线为低电平（ACK），则必须将下一个要发送的数据装入SSCIBUFR寄存器。然后，通过将SSCICKP位（SSCICTL0<4>）置1使能SCL引脚。

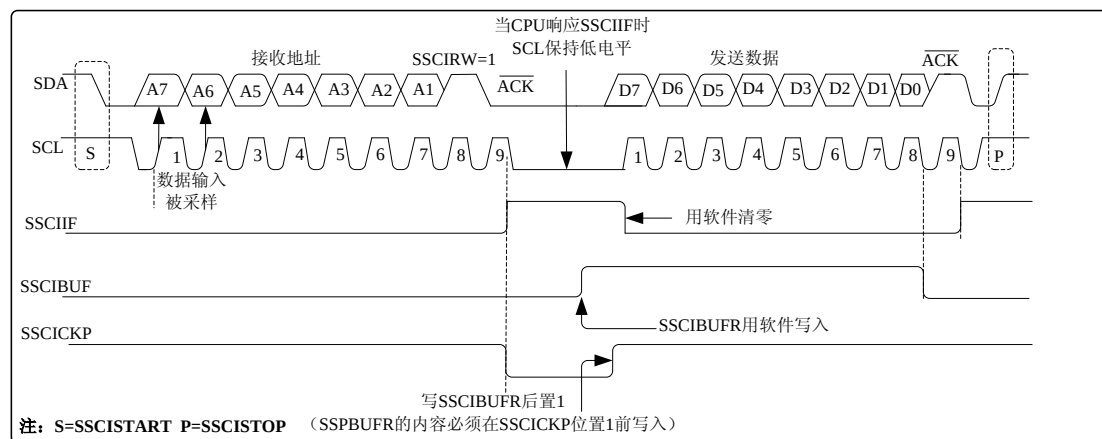


图 9.4 从动模式时序（发送，7 位地址）

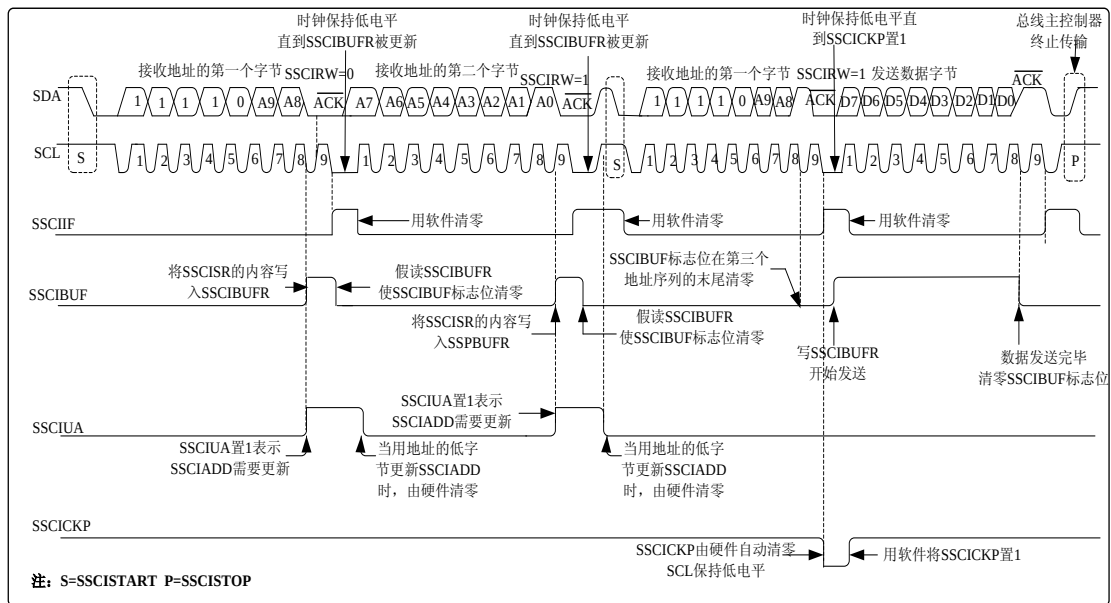


图 9.5 从动模式时序（发送，10 位地址）

从动发送设置:

- 通过 SSCIMOD<3:0>位选择 I2C 工作模式
 - 0110 = I2C 从动模式，7 位地址
 - 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
- 设置SSCIADD寄存器，设置从机地址，仅高七位有效；
- 清零SSCISTA寄存器的各标志，包括SSCIDA、SSCIRW、SSCIBUF等。
- 设置SDA引脚为输出，SCL为输入；
- 清零SSCIIF标志，如果需要中断打开各终端使能位；
- 使能SSCIEN，当输入地址字节的SSCIRW位置1并发生地址匹配时，SSCISTA寄存器的SSCIRW位被置1。接收到的地址被装入SSCIBUFR寄存器。
- ACK脉冲在第9位上发送，SCL引脚保持低电平。发送的数据装载到SSCIBUFR寄存器。
- 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。
- 标志位SSCIIF在第9个时钟脉冲的下降沿被置1。软件清零
- 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。

9.3.2.4 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9.6所示。

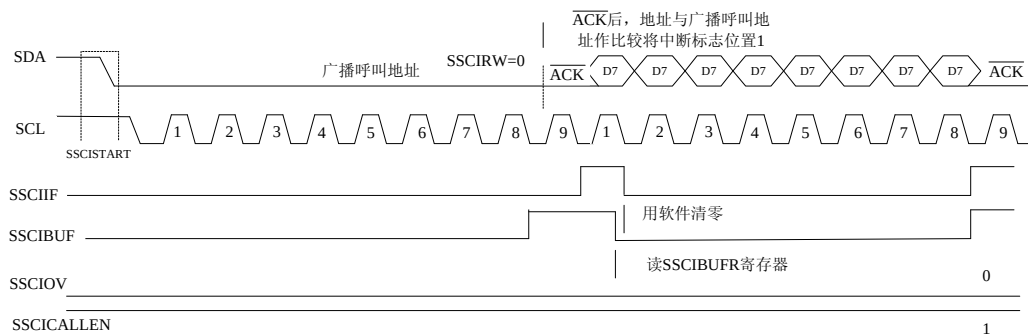


图9.6 从动模式广播呼叫地址时序（7或10位地址模式）

9.3.3 I2C 主控模式

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

在主控模式下，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控模式操作。当同时使能主控模式和从动模式时，需要使用软件区分中断源。

9.3.3.1 主控模式支持

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

用户即可选择以下6 项操作：

- 1) 在SDA 和SCL 上发出一个启动条件。
- 2) 在SDA 和SCL 上发出一个重复启动条件。
- 3) 写入SSCIBUFR寄存器，开始数据/地址的发送。
- 4) 在SDA 和SCL 上产生停止条件
- 5) 将I2C 端口配置为接收数据。
- 6) 在接收到数据字节后产生应答条件。

注：

当配置为I2C主控模式时，SSCI模块不允许事件排队。例如，在启动条件结束前，不允许用户发出另一个启动条件并立即写SSCIBUFR寄存器以发起传输。这种情况下，将不会写入SSCIBUFR，SSCIWCFL 位将被置1，这表明没有发生对SSCIBUFR的写操作。图9.7 为I2C主模式框图。

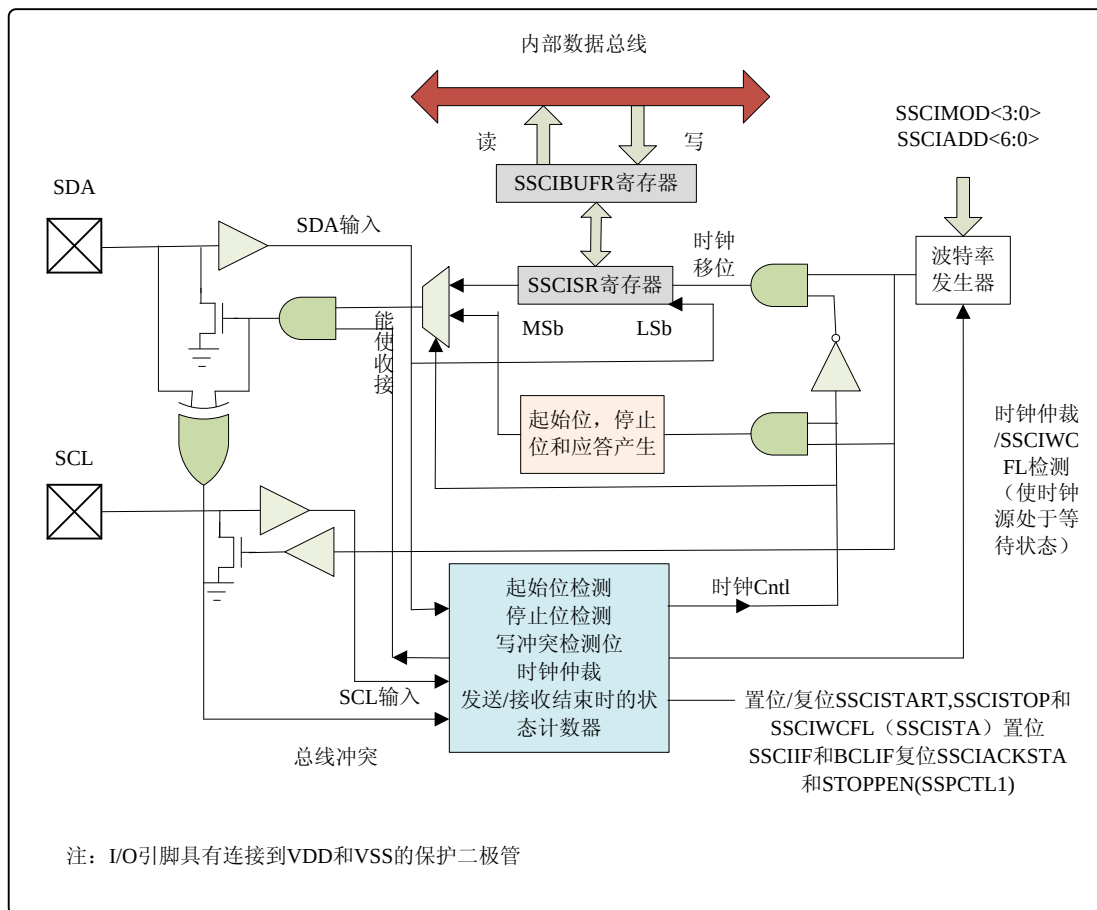


图9.7 I2C主模式框图

9.3.3.2 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 (SSCCTL1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。

9.3.3.3 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9.8和图9.9 所示。

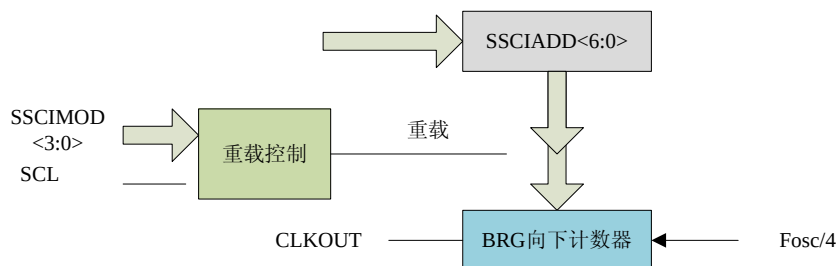


图 9.8 波特率发生器框图

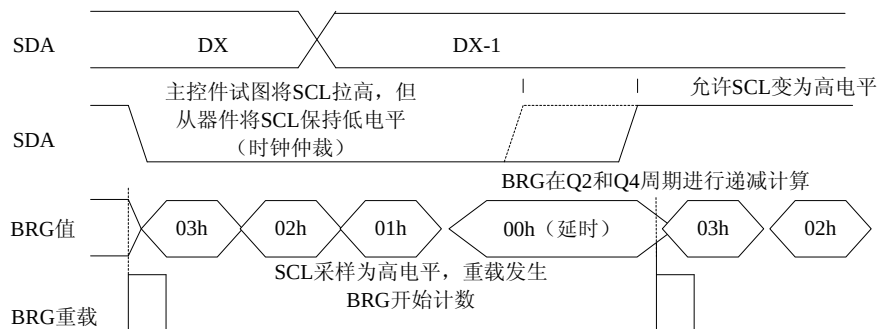


图 9.9 带有时钟仲裁的波特率发生器时序

9.3.3.4 I2C 主控模式启动条件时序

要发起启动条件，用户应将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。

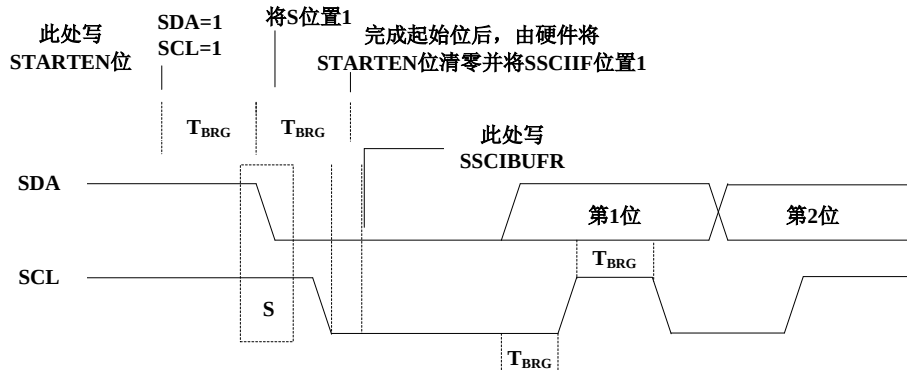


图9.10 第一个启动位时序

9.3.3.5 I2C 主控模式重复启动条件时序

将RESTARTEN位（SSCICTL1寄存器）编程为高电平，并且I2C逻辑模块处于空闲状态时，就会产生重复启动条件。当RESTARTEN位置1时，SCL引脚被拉为低电平。当SCL引脚采样为低电平时，波特率发生器装入SSCIADD<6:0>的内容，并开始计数。在一个波特率发生器计数周期（TBRG）内SDA引脚被释放（其引脚电平被拉高）。当波特率发生器超时时，如果SDA 采样为高电平，SCL引脚将被拉高。当SCL引脚采样为高电平时，波特率发生器将被重新装入SSCIADD<6:0>的内容并开始计数。SDA和SCL必须在一个计数周期TBRG内采样为高电平。随后将SDA引脚拉为低电平（SDA = 0）并保持一个计数周期TBRG，同时SCL为高电平。然后RESTARTEN位（SSCICTL1 寄存器）将自动清零，波特率发生器不会重载，SDA引脚保持低电平。一旦在SDA和SCL引脚上检测到启动条件，SSCISTART位（SSCISTA寄存器）将被置1。直到波特率发生器超时后，SSCIIF位才会置1。

注:

- 1) 有任何其他事件进行时，对RESTARTEN的编程无效。
- 2) 在重复启动条件期间，下列事件将会导致总线冲突:

- 当SCL 由低电平变为高电平时，SDA 采样为低电平。
- 在SDA 被拉低之前， SCL 变为低电平。这表示可能有另一个主器件正尝试发送数据1。

一旦SSCIIF 位被置1，用户便可以在7 位地址模式下将7 位地址写入SSCIBUFR，或者在10 位地址模式下写入默认的第二个地址字节。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。

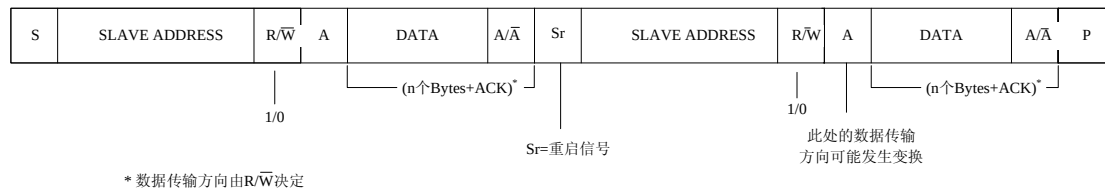


图9.11 I2C协议复合数据帧格式

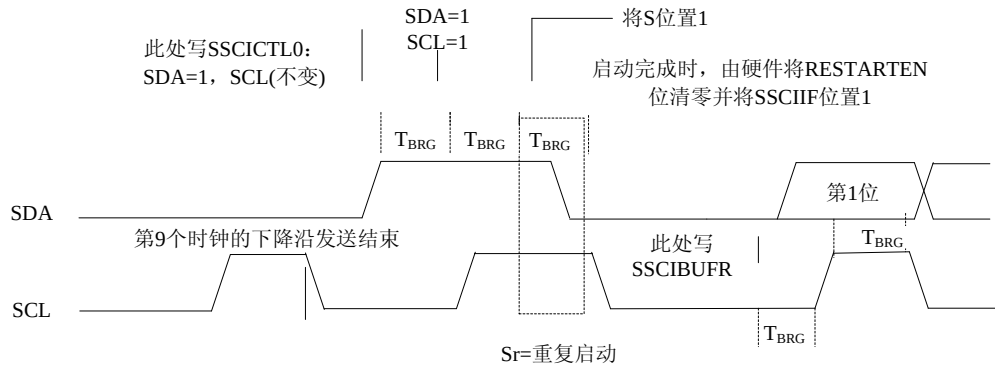


图9.12 重复启动条件时序波形

9.3.3.6 I2C 主控模式发送

I2C 协议中 SDI 引脚上的数据不仅来源于 SSCIBUFR 寄存器，还需要按 I2C 协议的规定发送起始位、停止位、应答位等信号。图 9.13 所示是 I2C 协议典型写数据帧格式。

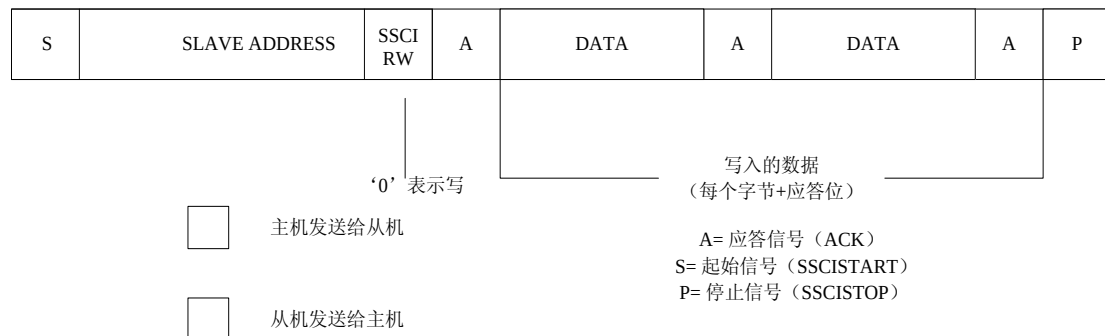


图 9.13 I2C 协议典型写数据帧格式

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 9.14 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。

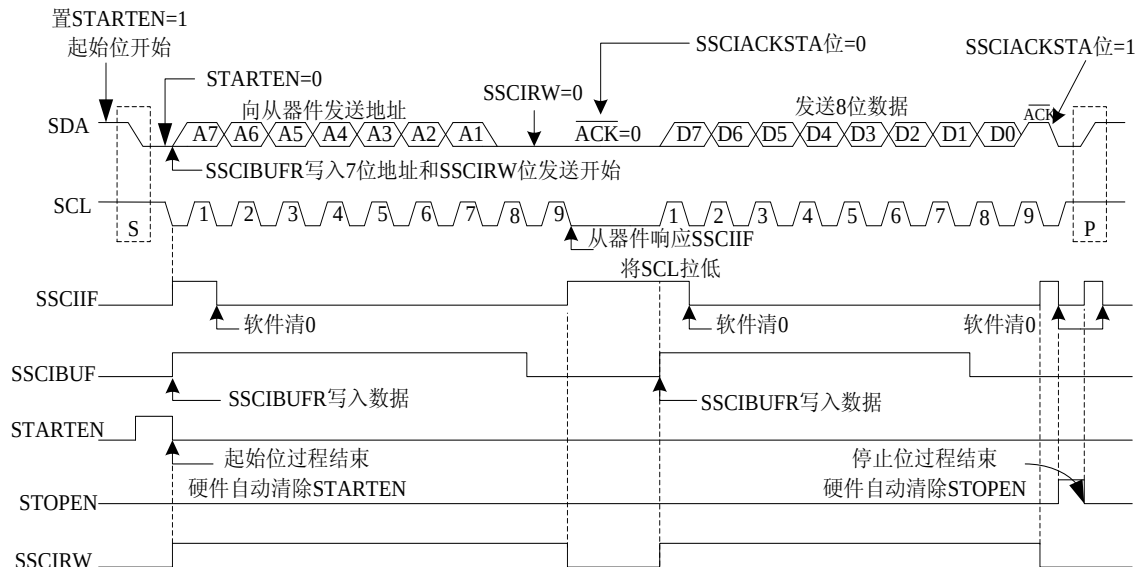


图 9.14 硬件 I2C 主模式发送数据时序图 (7 位地址)

9.3.3.7 I2C 主控模式接收

I2C主模式数据接受的系统结构与主模式数据发送系统结构相同，但是数据接收流程与数据发送流程不同。如图9.15为I2C协议典型读数据帧格式。

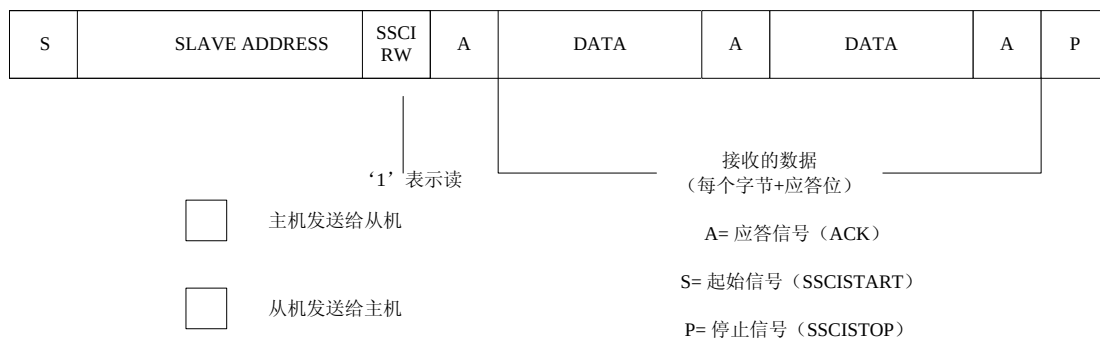


图9.15 I2C协议典型读数据帧格式

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SSCIBF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9.16 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。

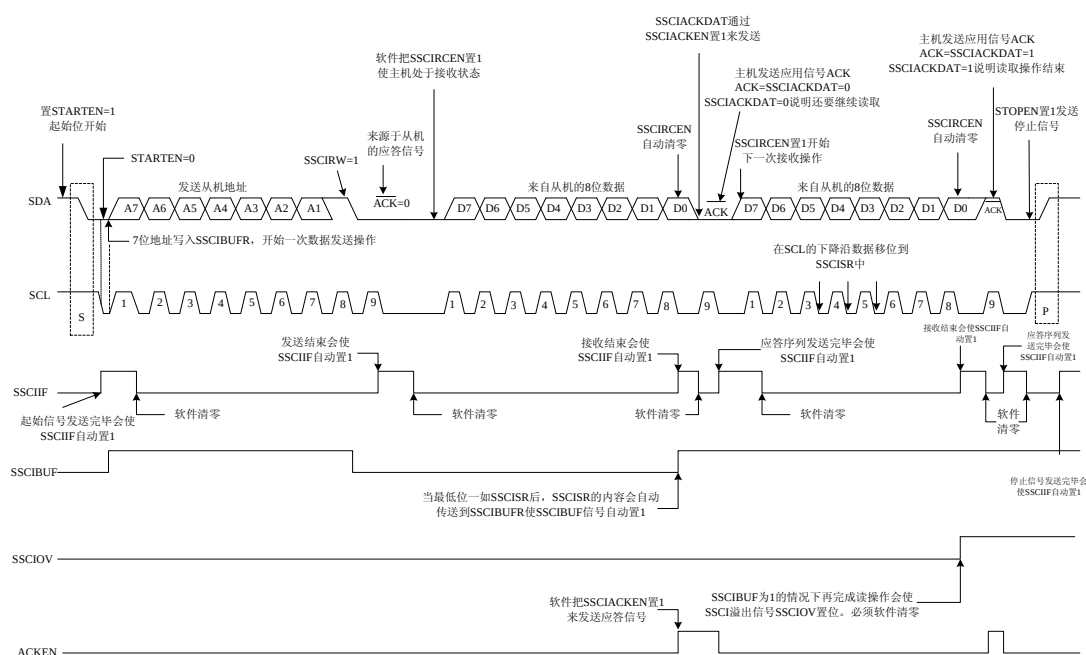


图9.16 I2C主模式接收数据时序图（7位地址）

9.3.3.8 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（TBRG）的计数，随后SCL引脚电平被拉高。当SCL引脚采样为高电平时（时钟仲裁），波特率发生器再进行一个TBRG周期的计数。然后SCL引脚被拉低。在这之后，SSCIACKEN位自动清零，波特率发生器关闭，SSCI模块进入空闲模式。

SSCIWCFL 状态标志位

如果用户在应答序列正在进行时写SSCIBUFR, SSCIWCFL 将被置1 且缓冲器的内容保持不变（未发生写操作）。

9.3.3.9 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。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）。

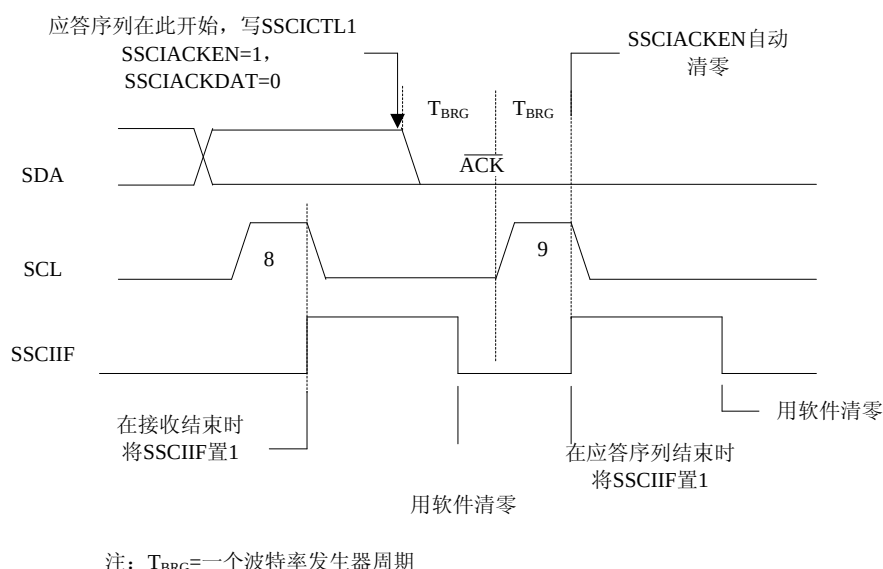


图9.17 应答序列时序波形

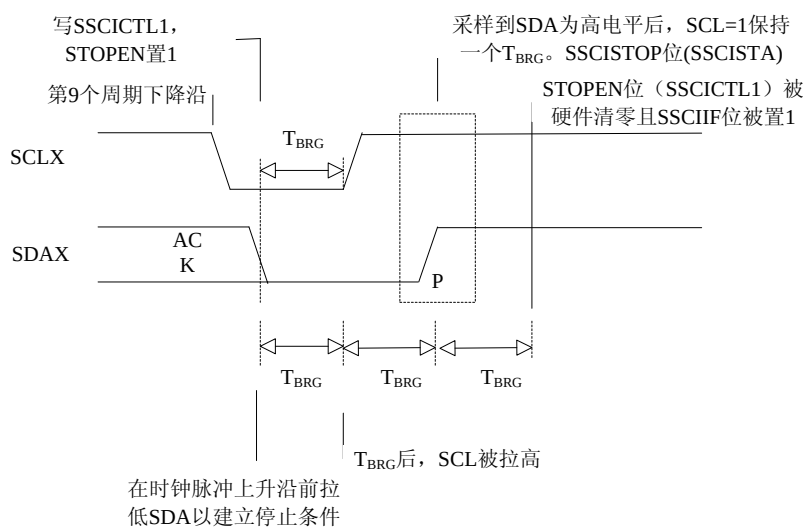


图9.18 停止条件接收或发送模式

9.3.3.10 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。

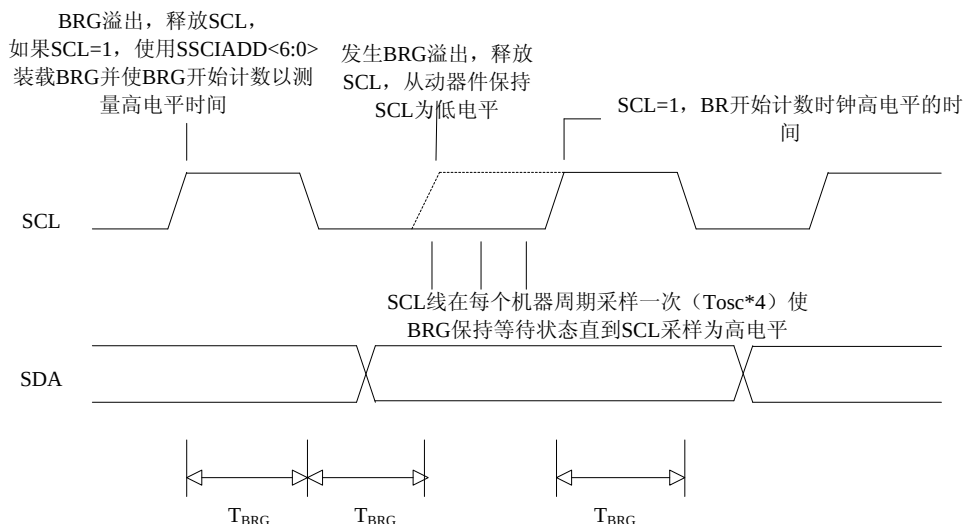


图 9.19 主控发送模式下的时钟仲裁时序

9.3.4 多主控器模式

在多主控器模式下,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。

9.3.4.1 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,SSCICTL1寄存器中的对应控制位清零。当执行完总线冲突中断服务程序后,如果I2C总线空闲,用户可通过发出启动条件恢复通信。主器件将继续监视SDA和SCL引脚。如果出现停止条件,SSCIIF位将被置1。无论发生总线冲突时发送的进度如何,写SSCIBUFR都会从第一个数据位开始发送数据。在多主机模式下,通过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。

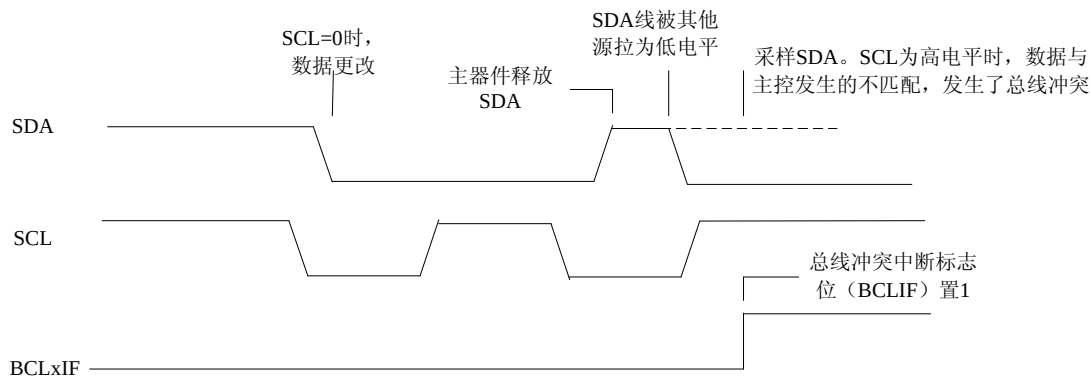


图 9.20 发送和应答时的总线冲突时序

9.3.4.2 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

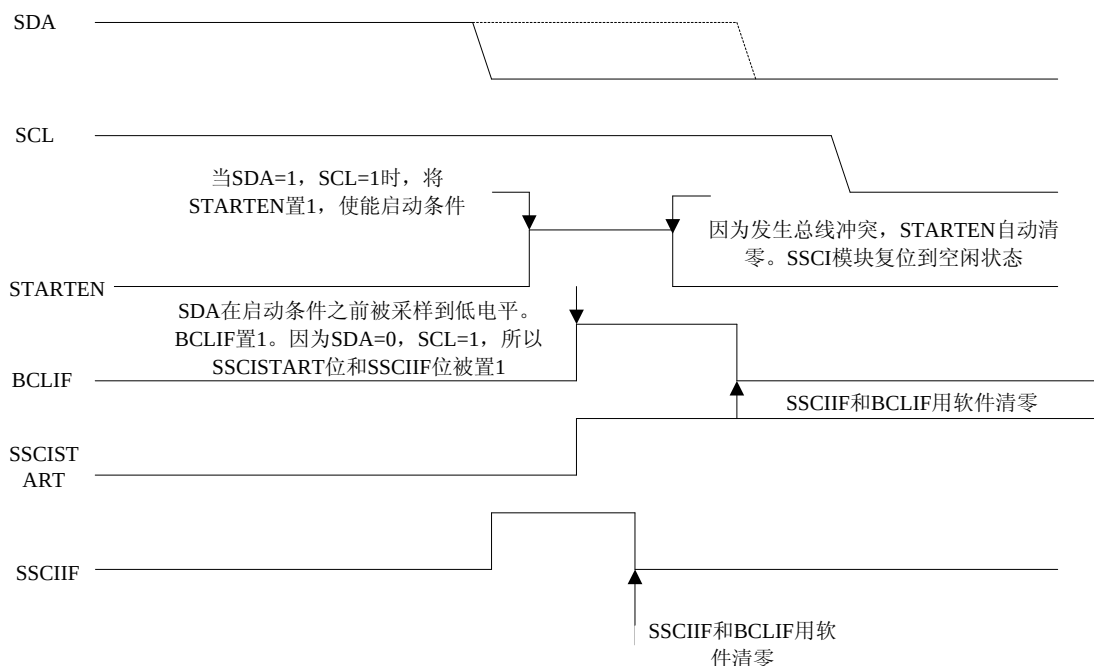
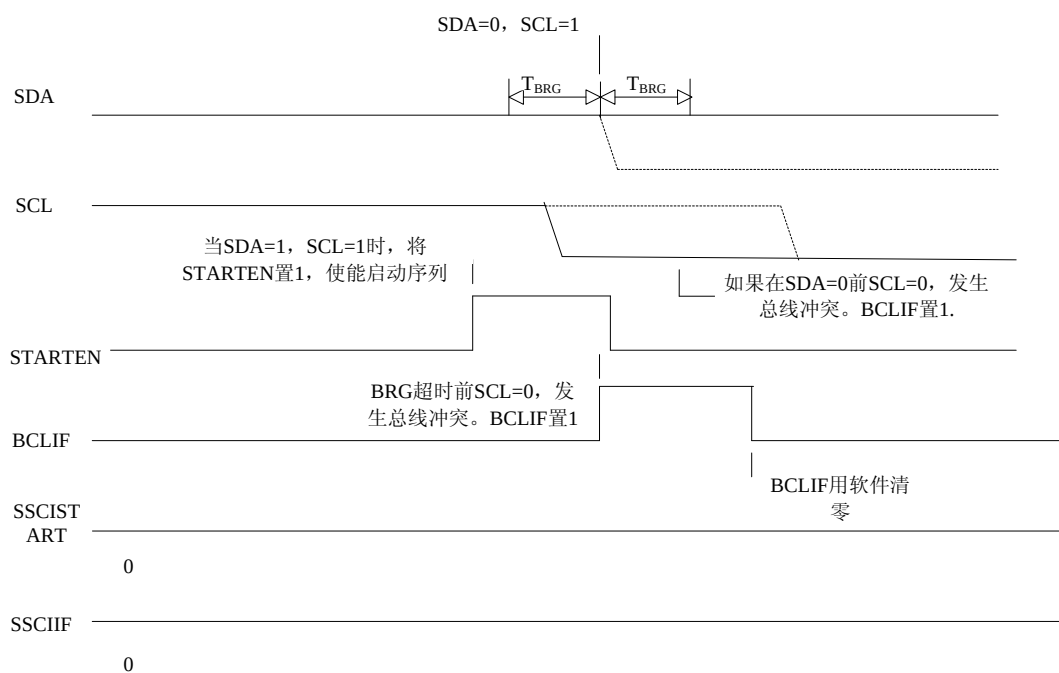


图9.21 启动条件期间的总线冲突（仅SDA）



图

9.22 启动条件期间的总线冲突（SCL=0）

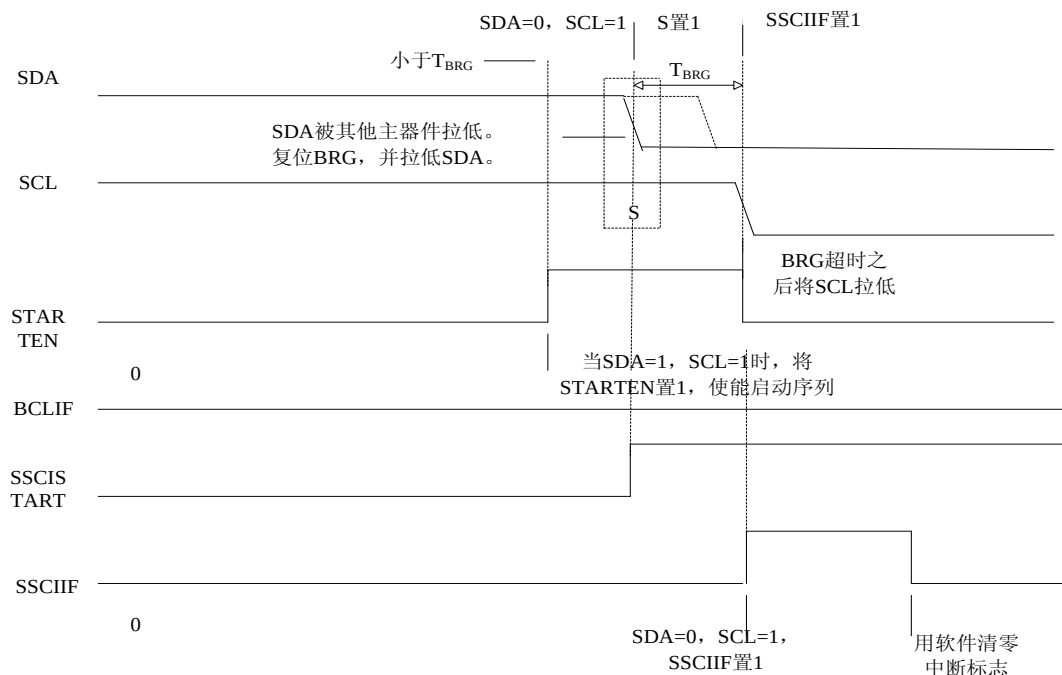


图9.23 启动条件期间由SDA仲裁引起的BRG复位

9.3.4.3 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。

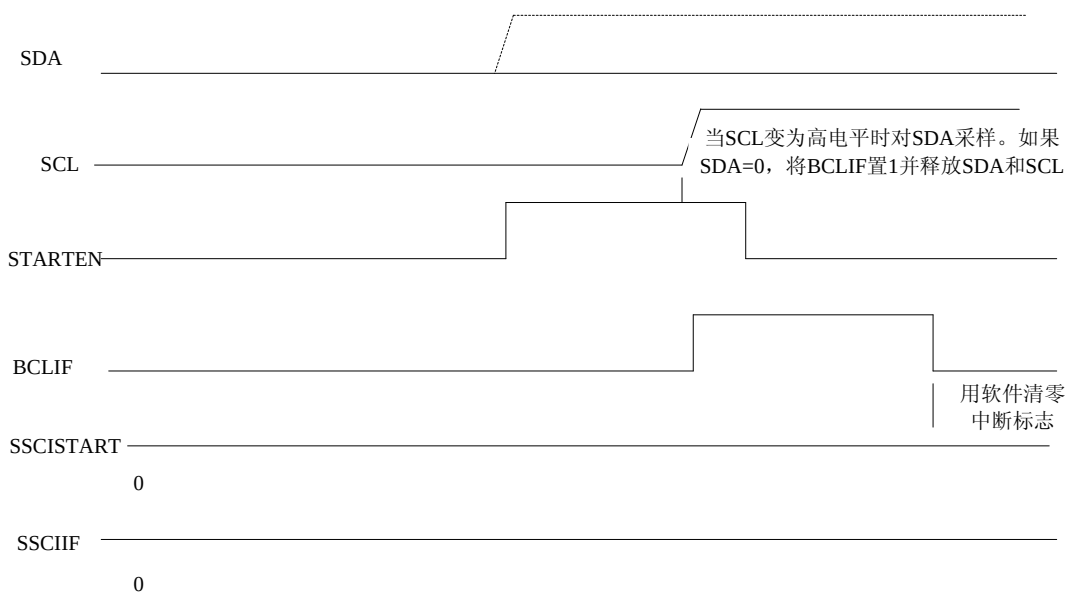


图 9.24 重复启动条件期间的总线冲突（情形 1）

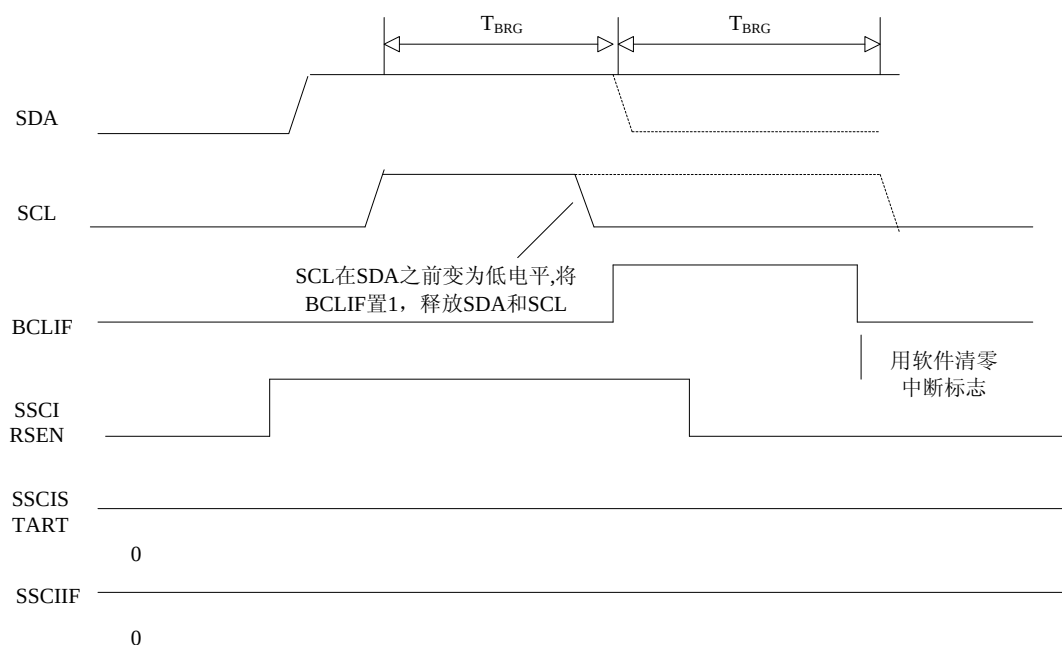


图 9.25 重复启动条件期间的总线冲突（情形 2）

9.3.4.4 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。

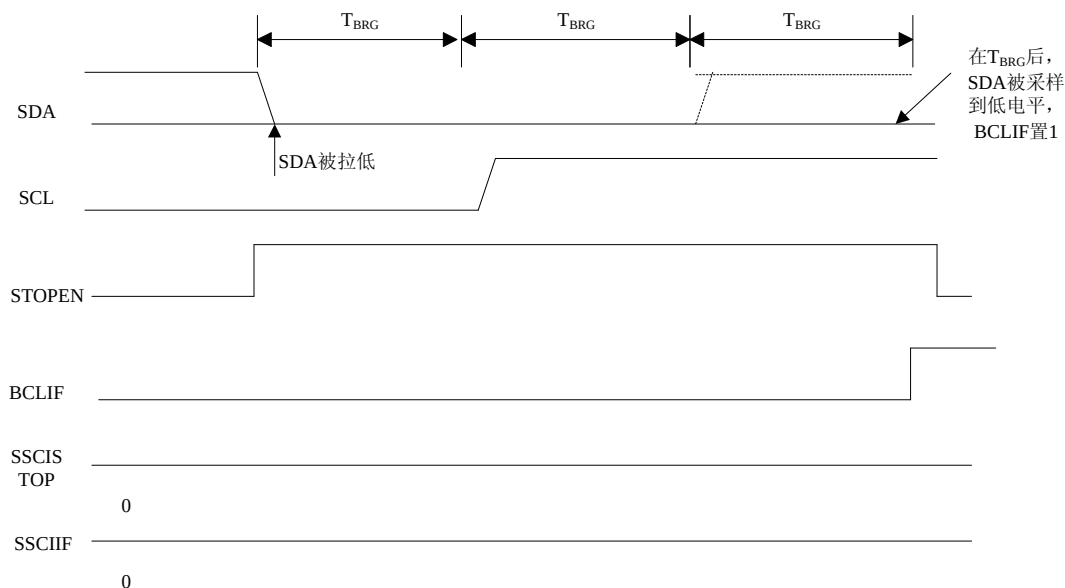


图9.26 停止条件期间的总线冲突（情形1）

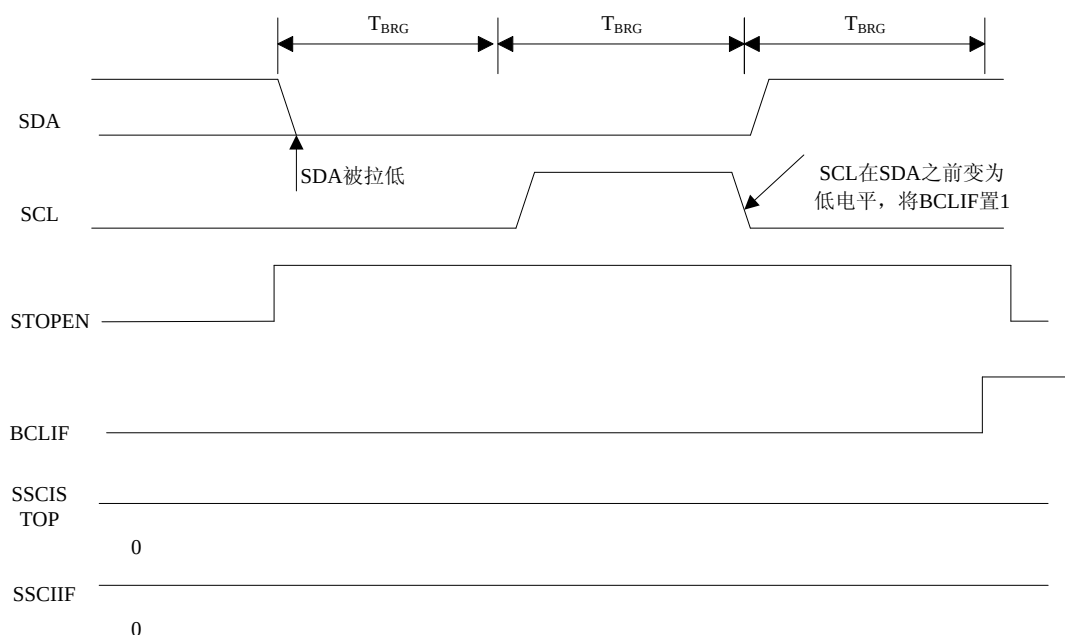


图9.27 停止条件期间的总线冲突（情形2）

9.3.4.5 SSCI 屏蔽寄存器

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

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

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

- 7 位地址模式：与SSCIADD <7:1> 进行地址比较。
- 10 位地址模式：仅与SSCIADD <7:0> 进行地址比较。SSCI 屏蔽在接收到地址的第一个（高）字节期间无效。

9.4 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 9.28 所示：

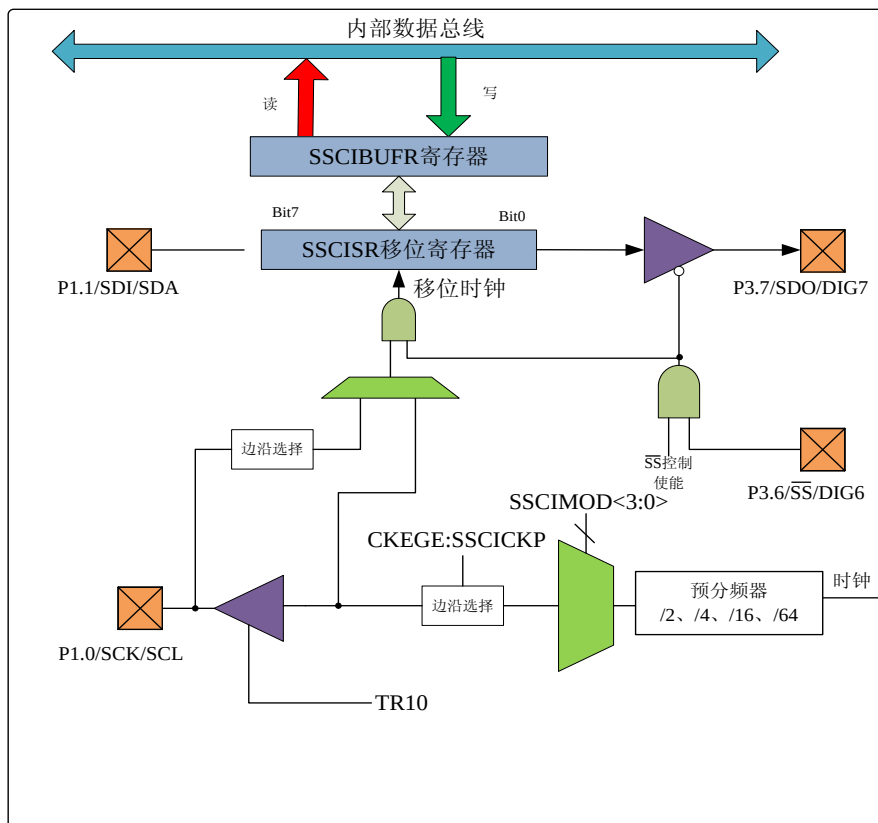


图 9.28 SPI 原理框图

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）——P3.7/SDO/DIG7
- ◆ 串行数据输入（Serial Data In，SDI）——P1.1/SDI/SDA
- ◆ 串行时钟（Serial Clock，SCK）——P1.0/SCK/SCL
- ◆ 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（ $\overline{SS}$ ）——P3.6/ $\overline{SS}$

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。

- 3) 当SPI 处于从动模式时, $\overline{SS}$ 引脚的状态会影响从TR3<7> 位读回的状态。从SSCI 模块向P3口发送的外设输出使能信号控制从TR3<7>位读回的状态。

9.4.1 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 SSCIBUF 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。

9.4.2 使能 SPI/IO 与外部链接

要使能串行端口, SSCI 使能位 SSCIEN (SSCCTL0<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 SSCCTL0 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>清零
- ◆ 对于SCK（从动模式），必须将TR1<0>置1
- ◆ 对于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。

9.4.3 典型连接

下图9.29给出了两个单片机之间的典型连接。主控制器（处理器1）通过发送SCK信号来启动数据传输。在两个处理器的移位寄存器之间，数据在编程设定的时钟边沿被传送，并在相反的时钟边沿被锁存。必须将两个处理器的时钟极性（SSCICKP）设置为相同，这样两个处理器就可以同时收发数据。数据是否有效，取决于应用软件。这就导致以下三种数据传输情形：

- ◆ 主控制器发送数据 — 从控制器发送无效数据
- ◆ 主控制器发送数据 — 从控制器发送数据
- ◆ 主控制器发送无效数据 — 从控制器发送数据

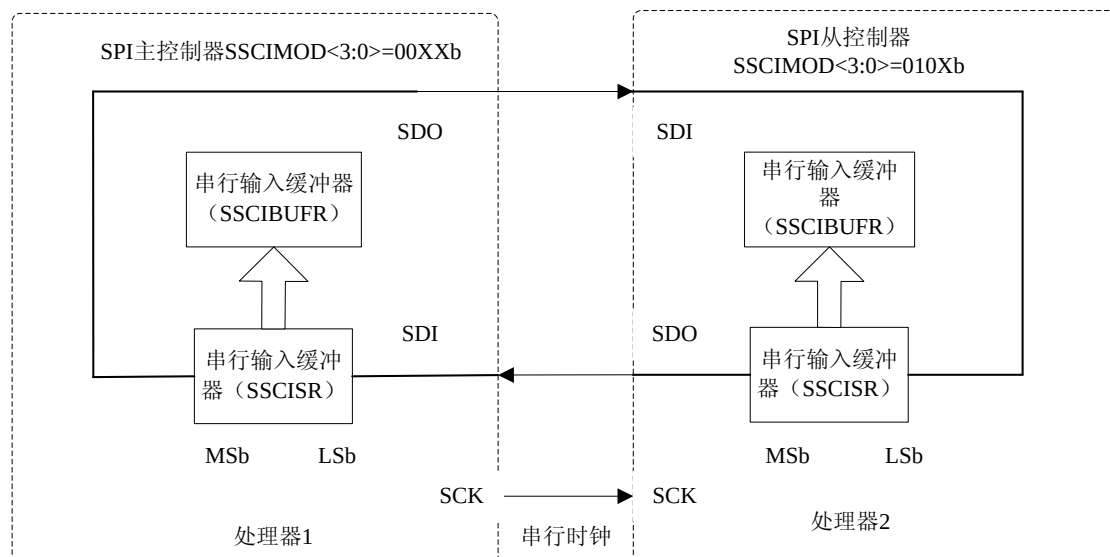


图 12.29 SPI 主从控制器连接

9.4.4 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 9.29 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（SSCCTL0<4>）进行适当的编程来选择时钟极性。图9.30、图9.31、图9.32 和图9.33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9.30 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。

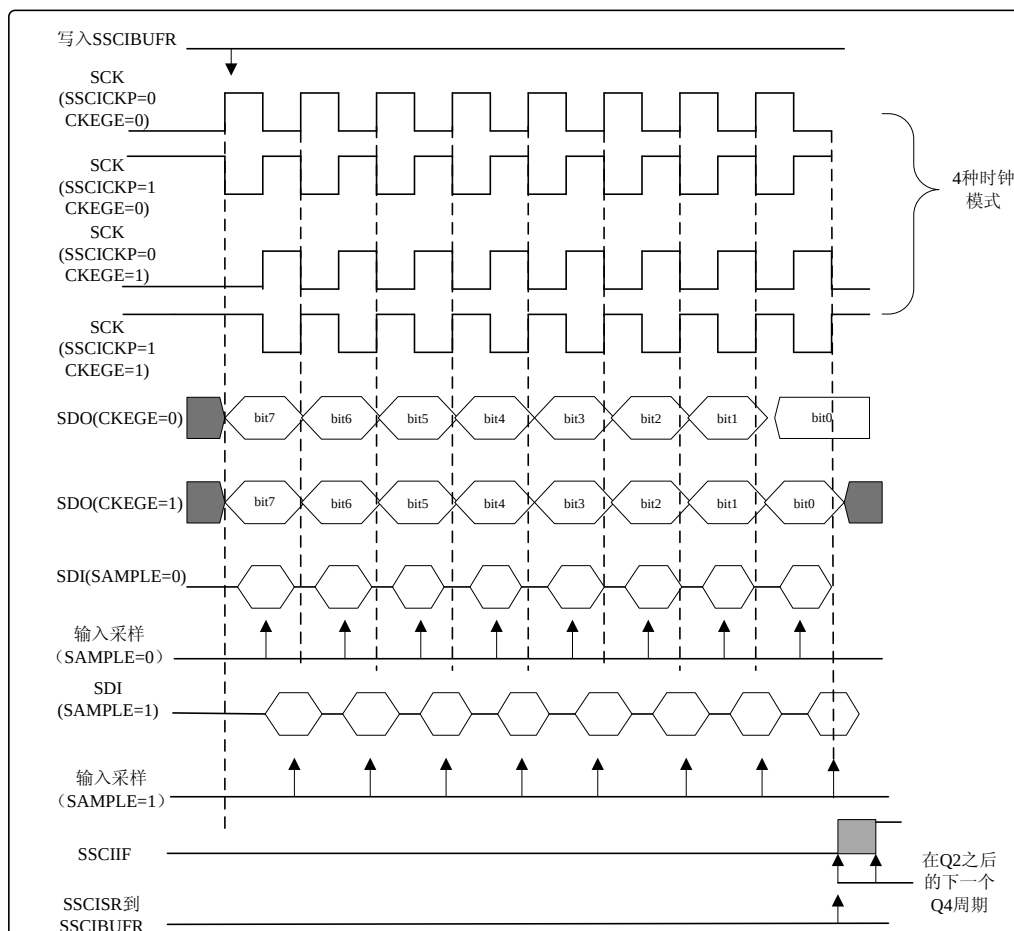


图 9.30 SPI 主控模式时序图

9.4.5 从模式

9.4.5.1 从动模式

在从动模式下，当SCK引脚上出现外部时钟脉冲时发送和接收数据。当最后一位数据被锁存后，中断标志位SSCIF置1。在从动模式下，外部时钟由SCK引脚上的外部时钟源提供。外部时钟必须满足电气规范中规定的高电平和低电平的最短时间要求。在休眠状态下，从控制器仍可发送/接收数据。当接收到一个字节时，器件从休眠状态唤醒。当有接收数据送到SSCIBUFR后，SSCIBUF信号也会置1。当读取SSCIBUFR后SSCIBUF会自动清零。但有SSCIF信号，一般情况下就不使用SSCIBUF来判断SSCIBUFR是否有数据了。当SSCIF置1

且SSCIBUF置1时，应立即读取SSCIBUFR中的数据，否则会出现数据丢失。

9.4.5.2 从动选择同步

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

注意：

- 1) 当SPI处于从动模式，并且 $\overline{SS}$ 引脚控制使能（SSCICTL0<3:0>= 0100）时，如果 $\overline{SS}$ 引脚置为VDD电平将使SPI模块复位。
- 2) 如果SPI工作在从动模式下并且CKEGE置1，则必须使能 $\overline{SS}$ 引脚控制。

当SPI模块复位时，位计数器被强制为0。这可以通过强制将 $\overline{SS}$ 引脚拉为高电平或将SSCIEN位清零实现。将SDO引脚和SDI引脚相连，可以仿真二线制通信。当SPI需要作为接收器工作时，SDO引脚可以被配置为输入端。这样就禁止了从SDO发送数据。因为SDI不会引起总线冲突，因而总是可以将其保留为输入（SDI功能）。

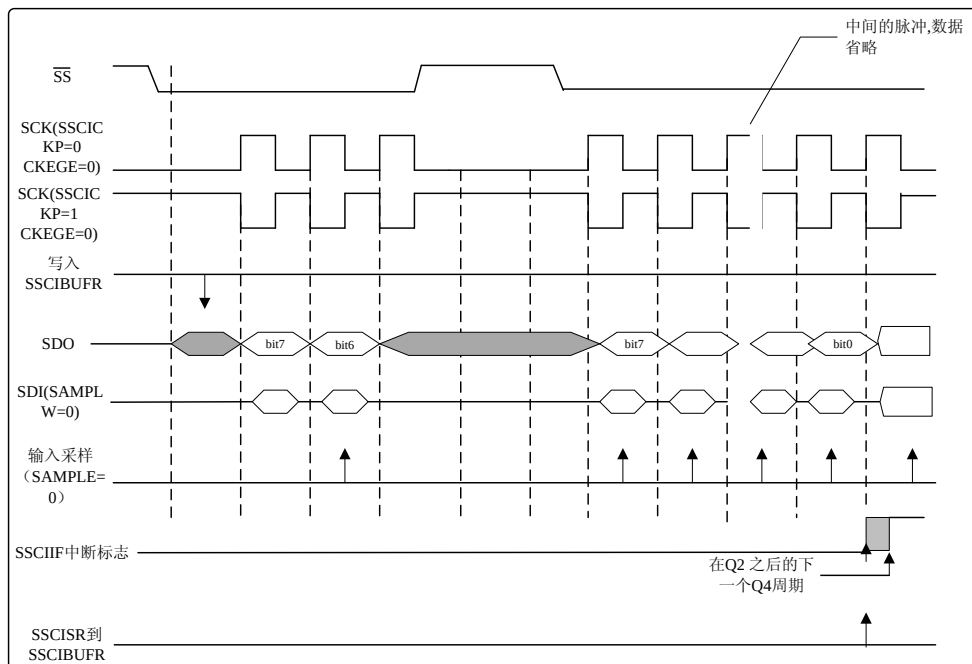


图9.31 从动时序图

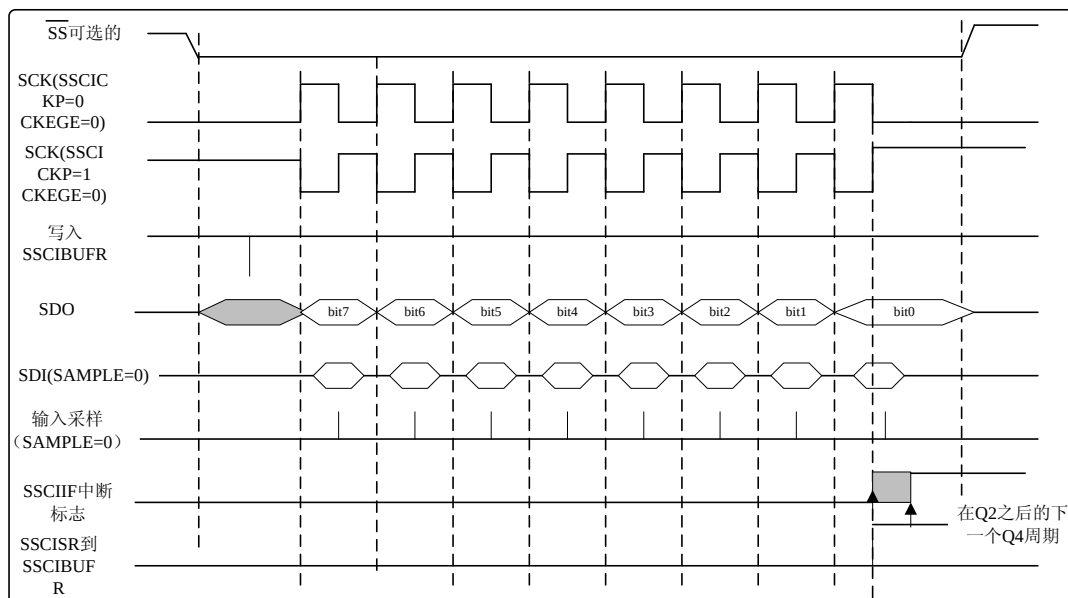


图9.32 SPI模式时序图（从动模式且CKEGE=0）

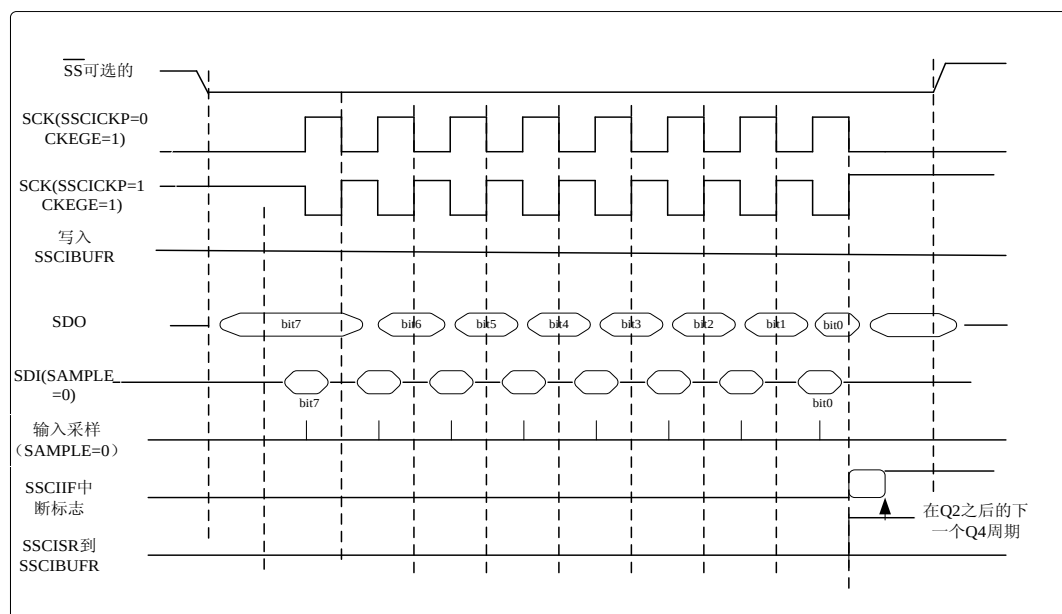


图 9.33 SPI 模式时序图（从动模式且 CKEGE=1）

9.4.6 休眠模式和复位

休眠模式下的工作

在主机模式下，进入休眠模式后所有模块的时钟都停振，在器件被唤醒前，发送/接收也将保持原先的状态。在器件恢复正常工作模式后，模块将继续发送/接收数据。

在从动模式下，SPI发送/接收移位寄存器与器件异步工作。这可以使器件在休眠状态时，仍可使数据被移入SPI发送/接收移位寄存器。当接收完8位数据后，SSCI中断标志位将置1，如果此时该中断是允许的，还将唤醒器件。

复位的影响

复位会禁止SSCI模块并终止当前的数据传输。

9.4.7 SPI 四种工作模式设置

9.4.7.1 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;

9.4.7.2 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;

9.4.7.3 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;

9.4.7.4 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;

10 LED驱动模块

10.1 概述

K8V327 内含一个 8×8 的 LED 驱动模块，最多可驱动 8 位 LED 数码管，共阴共阳可调，且带有辉度调节功能和开漏或推免输出模式可选功能。

模块框图如下：

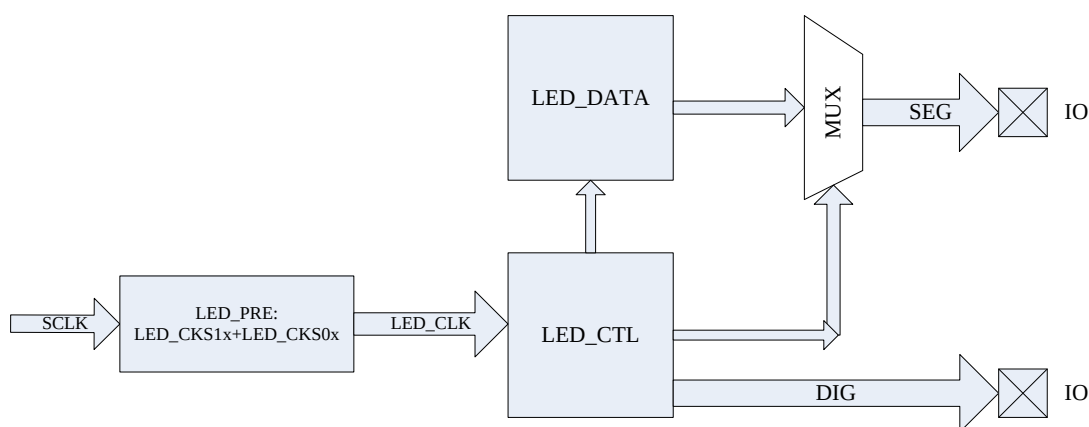


图 10.1 LED 驱动模块框图

当用户配置好 LED 驱动模块时，会在 DIGx 引脚产生一个扫描位选的信号，当 DIGx 为有效电平时，SEGx 引脚将相对应的数据寄存器 LEDDATAx 的内容对应输出，得以产生一个段选信号。图 10.2 为 LED 驱动模块与数码管连接示意图。

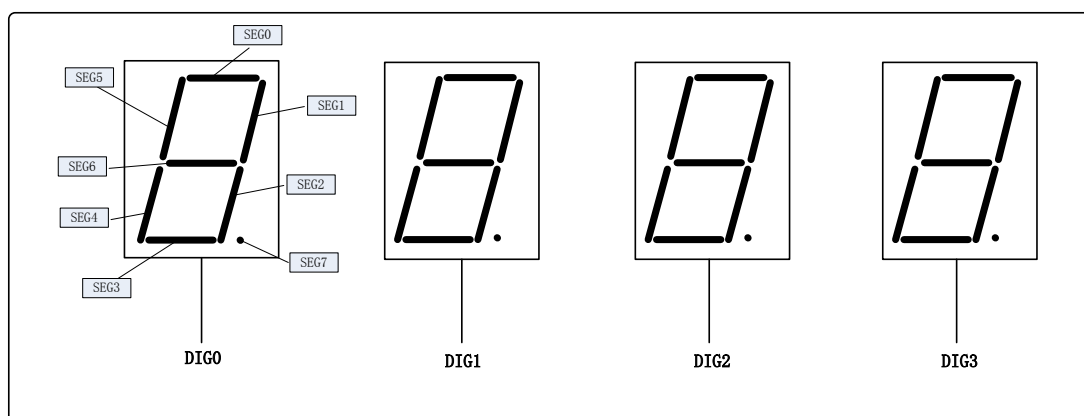


图 10.2 LED 驱动模块与数码管连接示意图

10.2 工作原理

10.2.1 扫描时钟产生

LED 显示电路扫描频率可调，时钟为系统时钟。LED 驱动模块提供两个可调分频，LEDPRE 寄存器的 LEDCKS1<3:0>和 LEDCKS0<3:0>，用户可通过 LEDPRE 寄存器对时钟进行分频得到所需的扫描频率。

LED 扫描频率计算公式如下：

$$CLK_{LED} = \frac{SCLK}{32 \times 2^{LEDCKS1<3:0>} \times (LEDCKS0<3:0> + 1)}。$$

10.2.2 扫描信号产生

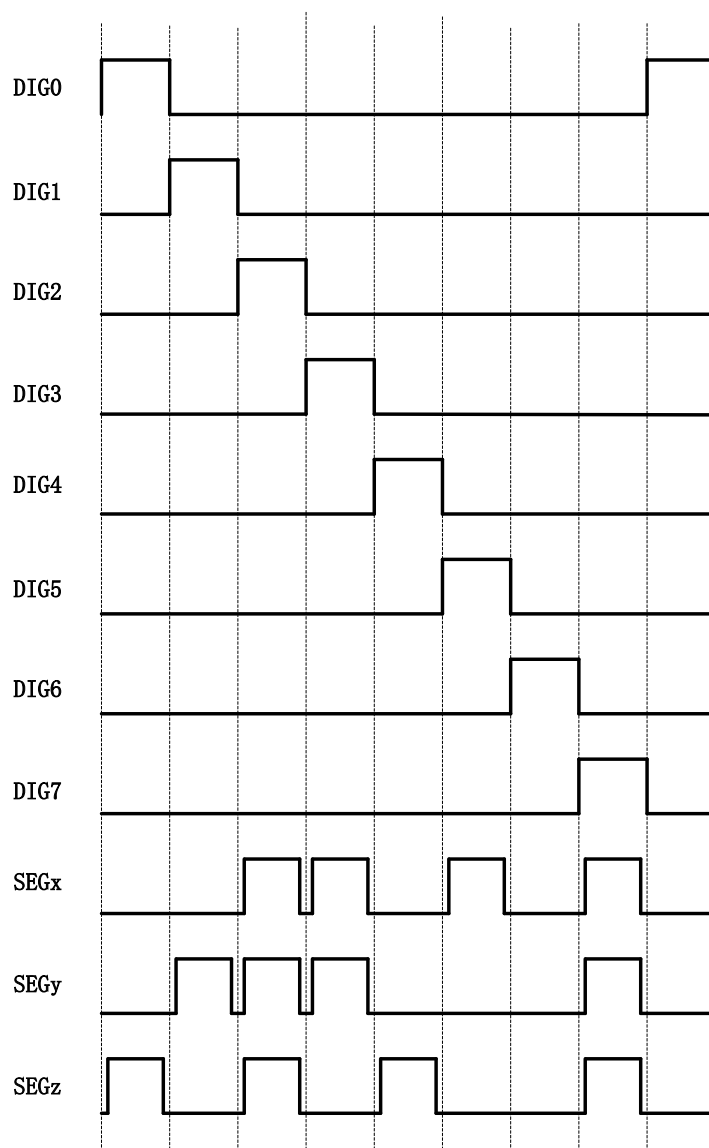


图 10.2 LED 扫描波形例图

10.3 LED 相关寄存器

表 10-1 相关寄存器

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
140H	LEDCTL	LEDEN	-	-	DIGEN2	DIGEN1	DIGEN0	PNS1	PNS0
141H	LEDPRE	LEDCKS13	LEDCKS12	LEDCKS11	LEDCKS10	LEDCKS03	LEDCKS02	LEDCKS01	LEDCKS00
142H	LEDDATA0	DAT07	DAT06	DAT05	DAT04	DAT03	DAT02	DAT01	DAT00
143H	LEDDATA1	DAT17	DAT16	DAT15	DAT14	DAT13	DAT12	DAT11	DAT10
144H	LEDDATA2	DAT27	DAT26	DAT25	DAT24	DAT23	DAT22	DAT21	DAT20
145H	LEDDATA3	DAT37	DAT36	DAT35	DAT34	DAT33	DAT32	DAT31	DAT30
146H	LEDDATA4	DAT47	DAT46	DAT45	DAT44	DAT43	DAT42	DAT41	DAT40
147H	LEDDATA5	DAT57	DAT56	DAT55	DAT54	DAT53	DAT52	DAT51	DAT50
148H	LEDDATA6	DAT67	DAT66	DAT65	DAT64	DAT63	DAT62	DAT61	DAT60
149H	LEDDATA7	DAT77	DAT76	DAT75	DAT74	DAT73	DAT72	DAT71	DAT70
150H	LEDOMS1	LEDOM17	LEDOM16	LEDOM15	LEDOM14	LEDOM13	LEDOM12	LEDOM11	LEDOM10
155H	LEDODS0	LEDOD07	LEDOD06	LEDOD05	LEDOD04	LEDOD03	LEDOD02	LEDOD01	LEDOD00
157H	LEDOMS0	LEDOM07	LEDOM06	LEDOM05	LEDOM04	LEDOM03	LEDOM02	LEDOM01	LEDOM00
158H	LEDODS1	LEDOD17	LEDOD16	LEDOD15	LEDOD14	LEDOD13	LEDOD12	LEDOD11	LEDOD10
15CH	LEDLUM	LEDBR	-	-	-	FRAMEF	LUM2	LUM1	LUM0

10.3.1 LED 显示驱动控制寄存器(LEDCTL)

寄存器10.1: LEDCTL: LED显示驱动控制寄存器 (140H)

	bit7							bit0	
复位值 0000 0000	LEDEN	-	-	DIGEN2	DIGEN1	DIGEN0	PNS1	PNS0	
	R/W	U	U	R/W	R/W	R/W	R/W	R/W	

LEDEN: LED 显示驱动使能位

1 = 使能 LED 模块

0 = 禁止 LED 模块

DIGEN<2:0>:LED 位使能数量选择位

000 = 使能 1 位, LEDDATA0 使能, LEDDATA1-7 禁止

001 = 使能 2 位, LEDDATA0-1 使能, LEDDATA2-7 禁止

010 = 使能 3 位, LEDDATA0-2 使能, LEDDATA3-7 禁止

011 = 使能 4 位, LEDDATA0-3 使能, LEDDATA4-7 禁止

100 = 使能 5 位, LEDDATA0-4 使能, LEDDATA5-7 禁止

101 = 使能 6 位, LEDDATA0-5 使能, LEDDATA6-7 禁止

110 = 使能 7 位, LEDDATA0-6 使能, LEDDATA7 禁止

111 = 使能 8 位, LEDDATA0-7 使能

PNS<1:0>: 共阴共阳选择位
 00 = LED 驱动模块工作在共阴模式
 01 = 系统保留
 10 = LED 驱动模块工作在开漏输出模式
 11 = LED 驱动模块工作在共阳模式

图注: R=可读 W=可写 -=未用 U=未实现位

注: 当 LED 模块选择在开漏输出模式时, 必须使该位设为 01

10.3.2 LED 显示驱动分频寄存器(LEDPRE)

寄存器10.2: LEDPRE: LED显示驱动分频寄存器 (141H)

	bit7				bit0			
复位值 0000 0000	LEDCKS1 3	LEDCKS1 2	LEDCKS1 1	LEDCKS1 0	LEDCKS0 3	LEDCKS0 2	LEDCKS0 1	LEDCKS0 0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LEDCKS1<3:0>: LED 时钟源分频选择位

000x=1:2
 0010=1:4
 0011=1:8
 0100=1:16
 0101=1:32
 0110=1:64
 0111=1:128
 1000=1:256
 1001=1:512
 1010=1:1024
 1011=1:2048
 1100=1:4096
 1101=1:8192
 1110=1:16384
 1111=1:32768

LEDCKS0<3:0>:LED 预分频比选择位

1111 = 1:16
 1110 = 1:15
 1101 = 1:14
 1100 = 1:13
 1011 = 1:12
 1010 = 1:11
 1001 = 1:10
 1000 = 1:9
 0111 = 1:8
 0110 = 1:7

0101 = 1:6
0100 = 1:5
0011 = 1:4
0010 = 1:3
0001 = 1:2
0000 = 1:1

图注：R=可读 W=可写 -=未用 U=未实现位

10.3.3 LED 显示驱动数据寄存器(LED DATAx)

寄存器10.3: LED DATAx: LED显示驱动数字寄存器

复位值 0000 0000	bit7							bit0
	DATx7	DATx6	DATx5	DATx4	DATx3	DATx2	DATx1	DATx0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DATx<7:0>: LED 显示驱动数据位 (x=0~7)

1 = LED 熄灭

0 = LED 点亮

LED 共有 8 个 LED 数据寄存器，地址如下：

表 10-2 LED 数据寄存器地址

LEDDATA0	142H
LEDDATA1	143H
LEDDATA2	144H
LEDDATA3	145H
LEDDATA4	146H
LEDDATA5	147H
LEDDATA6	148H
LEDDATA7	149H

图注：R=可读 W=可写 -=未用 U=未实现位

10.4 辉度调节

LED 驱动模块带有辉度调节功能。用户可以通过 LEDLUM 寄存器的 LUM<2:0>位来设置 LED 的辉度。

寄存器10.4: LEDLUM: LED显示驱动辉度调节设置寄存器 (15CH)

复位值 0--- 0000	bit7							bit0
	LEDBR	-	-	-	FRAMEF	LUM2	LUM1	LUM0
	R/W	U	U	U	R	R/W	R/W	R/W

LEDBR: LED 波形输出控制位

1 = LED 驱动只输出 COM 端波形，SEG 端相应 IO 口为通用 IO 口

0 = LED 驱动波形正常输出

FRAMEF: LED 显示标志位
 1 = LED 完成 1 次周期扫描
 0 = LED 未完成周期扫描

LUM<2:0>: LED 辉度调节设置位
 000 = 15:16
 001 = 14:16
 010 = 12:16
 011 = 10:16
 100 = 8:16
 101 = 6:16
 110 = 4:16
 111 = 2:16

图注：R=可读 W=可写 -=未用 U=未实现位

10.5 输出模式

LED 模块提供 2 种驱动信号输出方式：推挽输出和开漏输出。在选择开漏输出模式同时还可以设置 P 管开漏或者 N 管开漏。

10.5.1 LED 显示驱动输出模式设置寄存器

寄存器10.4: LEDOMS0: LED显示驱动输出模式设置寄存器（157H）

	bit7						bit0	
复位值 0000 0000	LEDOM07	LEDOM06	LEDOM05	LEDOM04	LEDOM03	LEDOM02	LEDOM01	LEDOM00
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LEDOM0x: LED-DIGx 端输出模式设置位
 1 = 输出方式为开漏输出
 0 = 输出方式为推挽输出

图注：R=可读 W=可写 -=未用 U=未实现位

寄存器10.4: LEDOMS1: LED显示驱动输出模式设置寄存器1（150H）

	bit7						bit0	
复位值 0000 0000	LEDOM17	LEDOM16	LEDOM15	LEDOM14	LEDOM13	LEDOM12	LEDOM11	LEDOM10
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LEDOM1x: LED-SEGx 端输出模式设置位
 1 = 输出方式为开漏输出
 0 = 输出方式为推挽输出

图注：R=可读 W=可写 -=未用 U=未实现位

10.5.2 LED 开漏方式选择寄存器

共阳级时，将 LEDODS0 寄存器配置为 P 管开漏，LEDODS1 寄存器配置位 N 管开漏；共阴级时，将 LEDODS0 寄存器配置为 N 管开漏，LEDODS1 寄存器配置位 P 管开漏。

寄存器10.4: LEDODS0: LED显示驱动开漏方式选择寄存器0 (155H)

复位值 0000 0000	bit7							bit0
	LEDOD07	LEDOD06	LEDOD05	LEDOD04	LEDOD03	LEDOD02	LEDOD01	LEDOD00
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LEDOD0x: LED-DIGx 端开漏方式选择位

1 = P 管开漏

0 = N 管开漏

图注: R=可读 W=可写 -=未用 U=未实现位

注: LEDOMS0 寄存器的 LEDOM0x 为 1 时,LEDODS0 寄存器对应的 LEDOD0x 位有效。

寄存器10.4: LEDODS1: LED显示驱动开漏方式选择寄存器1 (158H)

复位值 0000 0000	bit7							bit0
	LEDOD17	LEDOD16	LEDOD15	LEDOD14	LEDOD13	LEDOD12	LEDOD11	LEDOD10
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LEDOD1x: LED-SEGx 端开漏方式选择位

1 = P 管开漏

0 = N 管开漏

图注: R=可读 W=可写 -=未用 U=未实现位

注: LEDOMS1 寄存器的 LEDOM1x 为 1 时,LEDODS1 寄存器对应的 LEDOD1x 位有效。

10.6 LED 显示驱动使用方法

LED 驱动模块使用方法:

1. 通过 LEDPRE 寄存器选择时钟分频得到所需的扫描频率;
2. 设置 LEDCTL 寄存器来设置 DIG 数量;
3. 向相应的 LEDDATAx 寄存器中写入显示数据;
4. 设置 PNS<1:0>位来选择供阴或共阳的输出方式;
5. 设置 LEDOMS0/1 寄存器选择 LED 的输出模式, 如果选择开漏输出模式, 设置 LEDODS1/0 寄存器来选择 P 管开漏或者 N 管开漏;
6. 使能启动 LED 驱动模块。

11 Vbias偏压输出

KF8V327 系列提供 16 个通道的偏压输出。

用户使用 Vbias 偏压输出时，需要将相应 IO 口设置为输入，且必须先将 VBIASCTL 寄存器的 VBIASEN 位置 1 后，再将相应的 VBIAS1xSEN 和 VBIAS2xSEN 寄存器中的设置位打开以实现 1/2VDD 的偏压输出，否则对 VBIAS1xEN 和 VBIAS2xEN 寄存器的操作无效。

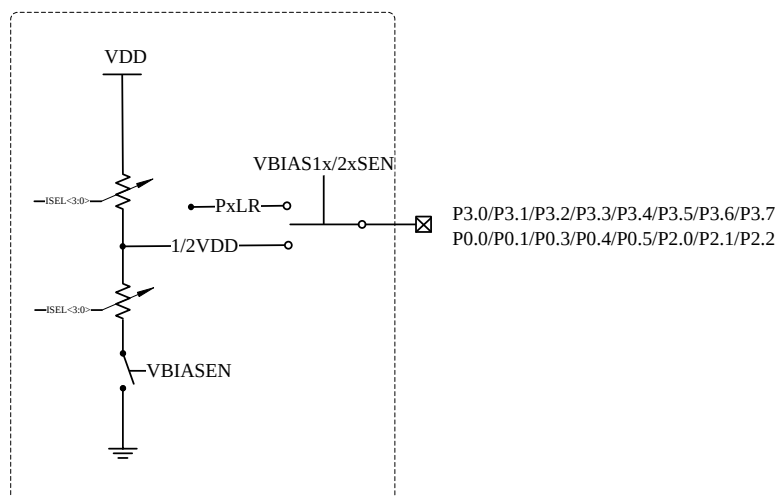


图 12.1 偏压输出电路

11.1 相关寄存器

表 12-1 偏压输出相关寄存器

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
6BH	VBIASCTL	VBIASEN	-	-	-	-	-	ISEL1	ISEL0
6CH	VBIAS1EN	VBS17EN	VBS16EN	VBS15EN	VBS14EN	VBS13EN	VBS12EN	VBS11EN	VBS10EN
6DH	VBIAS2EN	VBS27EN	VBS26EN	VBS25EN	VBS24EN	VBS23EN	VBS22EN	VBS21EN	VBS20EN

11.1.1 偏压输出控制寄存器(VBIASCTL)

寄存器12.1: VBIASCTL: 偏压输出控制寄存器 (6BH)

	bit7							bit0
复位值 00-- --00	VBIASEN	保留	-	-	-	-	ISEL1	ISEL0
	R/W	R/W	U	U	U	U	R/W	R/W

VBIASEN: 偏压输出使能位
1 = 使能偏压输出
0 = 禁止偏压输出

ISEL<1:0>: 偏压电流选择位
00 = 25uA
01 = 50uA

10 = 83uA

11 = 250uA

图注：R=可读 W=可写 -=未用 U=未实现位

注：保留位只能写 0，不能写 1。

11.1.2 偏压输出设置寄存器(VBIAS1/2EN)

寄存器12.2: VBIAS1EN:偏压输出设置寄存器1 (6CH)

复位值 0000 0000	bit7						bit0	
	VBS17EN	VBS16EN	VBS15EN	VBS14EN	VBS13EN	VBS12EN	VBS11EN	VBS10EN
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

VBS1xEN: 偏置输出口状态设置位

0 = 偏置输出口对应IO口为数字口

1 = 偏置输出口对应IO口为模拟口（偏压输出）

图注：R=可读 W=可写 -=未用 U=未实现位

寄存器12.3: VBIAS2EN: 偏压输出设置寄存器1 (6DH)

复位值 0000 0000	bit7						bit0	
	VBS27EN	VBS26EN	VBS25EN	VBS24EN	VBS23EN	VBS22EN	VBS21EN	VBS20EN
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

VBS2xEN: 偏置输出口状态设置位

0 = 偏置输出口对应IO口为数字口

1 = 偏置输出口对应IO口为模拟口（偏压输出）

图注：R=可读 W=可写 -=未用 U=未实现位

注1：必须先将VBIASCTL寄存器的VBIASEN位置1后，再将相应的VBIAS1xSEN和VBIAS2xSEN寄存器中的设置位打开以实现1/2VDD的偏压输出，否则对VBIAS1xEN和VBIAS2xEN寄存器的操作无效。

注2：

表12-2 寄存器设置位与IO口对应表

设置位	对应IO口
VBS10EN	P0.0
VBS11EN	P0.1
VBS12EN	P0.3
VBS13EN	P0.4
VBS14EN	P0.5
VBS15EN	P2.0
VBS16EN	P2.1
VBS17EN	P2.2
VBS20EN	P3.0
VBS21EN	P3.1
VBS22EN	P3.2
VBS23EN	P3.3
VBS24EN	P3.4
VBS25EN	P3.5
VBS26EN	P3.6

11.2 偏压输出使用方法

偏压输出配置方法：

1. 配置ISEL<1:0>位来设置偏压输出电流大小；
2. 配置VBIAS1/2EN寄存器来设置偏压输出端口。

12 复位

K8V327 具有:上电复位(POR)、WDT 复位、RST 复位和欠压检测复位(LVR)四种复位方式。

有些寄存器的状态在上电复位时它们的状态不定，而在其它复位发生时其状态将保持不变；其它大多数寄存器在复位事件发生时将被复位成“复位状态”。图 15.1 给出了片内复位电路的简化结构方框图。

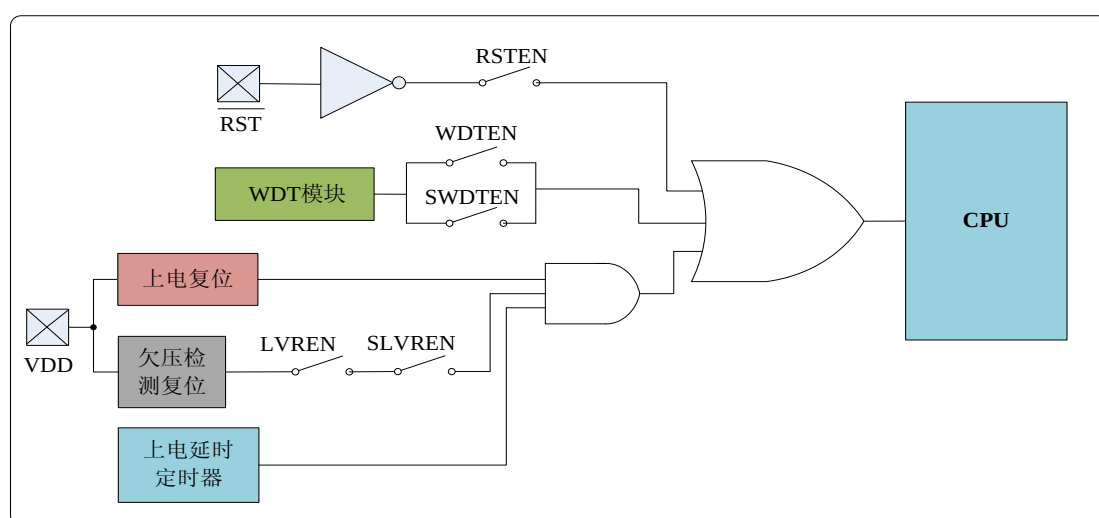


图 12.1 片内复位电路简化框图

注:上电延时定时器只对上电复位(POR)和欠压复位(LVR)有效。

12.1 电源控制状态寄存器(PCTL)

如寄存器 12.1 所示， $\overline{\text{LVR}}$ 位的状态在单片机上电复位时是不确定的。如果用户在使用中要用到该位，在程序初始化部分需将其置 1，随后如果有复位发生且 $\overline{\text{LVR}}=0$ ，则表示发生过欠压检测复位。 $\overline{\text{LVR}}$ 状态位是“无关”位，如果欠压检测电路被关闭(通过设定配置字中的 LVREN 位和 PCTL 中的 SLVREN 位)， $\overline{\text{LVR}}$ 状态位是不可预知的。 $\overline{\text{POR}}$ 是上电复位状态位，该位在上电复位时被清零，在其它情况下不受影响。

寄存器12.1: PCTL: 电源控制寄存器(地址:2EH)

复位值 ---1 000x	bit7			SLVREN	IPEN	SWDTEN	$\overline{\text{POR}}$	bit0 $\overline{\text{LVR}}$
	-	-	-					
	R/W	R/W	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
当配置字的 WDTEN=1 时，为无关位

$\overline{\text{POR}}$: 上电复位状态位
1 = 未发生上电复位
0 = 发生了上电复位

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

图注: R=可读 W=可写 -=未用 U=未实现位

12.2 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。K8V327 的上电复位时间为 30ms 左右（ $\overline{\text{PWRT}}=0$ ，上电延时使能）。

12.3 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。

12.4 RST 复位

使能外部 RST 复位（配置位 RSTEN=1）后，当引脚 P0.2/ $\overline{\text{RST}}$ 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 P0.2 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，K8V327 器件有一个噪声滤波器用于滤除 $\overline{\text{RST}}$ 引脚上的噪声干扰，图 12.2 是建议 $\overline{\text{RST}}$ 复位电路。

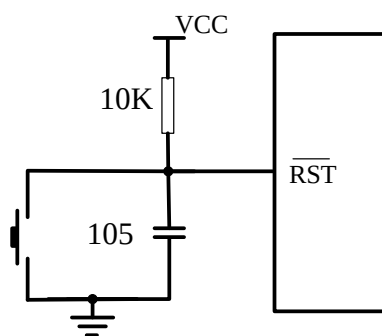


图 12.2 建议 $\overline{\text{RST}}$ 复位电路

12.5 欠压检测复位(LVR)

K8V327 系列中的单片机具有片内欠压检测复位电路。通过编程时设定配置位中的 LVREN 位可以禁止/使能(清零/置 1)欠压检测复位电路,当配置位中的 LVREN 位被使能后,用户还要在软件中设置 PCTL 中的 SLVREN 位来禁止/使能(清零/置 1)欠压检测复位电路。

如果 V_{DD} 跌落至 V_{LVR} 以下且持续时间大于 T_{LVR} (T_{LVR} 大于 10us), 欠压检测电路将使单片机复位,单片机保持复位状态直到 V_{DD} 上升到 V_{LVR} 以上,如果上电延时使能(配置位 $\overline{PWRT}=0$), 此时上电延时定时器启动,使器件在随后约 30ms 的延时时间处于延时复位状态,过了约 30ms 以后单片机开始正常工作。

如果 V_{DD} 跌落至 V_{LVR} 以下的时间小于规定参数(T_{LVR}), 将不保证可产生复位。

如果在上电延时定时器运行过程中发生 V_{DD} 跌落至 V_{LVR} 以下的情况,器件将返回欠压检测复位状态且上电延时定时器被重新初始化。直到 V_{DD} 上升至 V_{LVR} 以上时,上电延时定时器启动一个约 30ms 的复位延时,如在延时期间没有欠压发生,单片机会退出复位状态开始正常工作。

12.6 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约 30ms 的固定延时时间(配置位 $\overline{PWRT}=0$)。上电延时定时器的定时时钟为系统内部低频振荡器。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约 30ms。上电延时定时器使单片机在 V_{DD} 上升到适当电平后才开始正常运行。

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

12.7 不同复位条件下对寄存器的影响

表 15-1 寄存器在各种复位发生后的状态

寄存器	地址	上电复位	RST 复位 欠压检测	中断唤醒 WDT 超时唤醒
T0	01H	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02H	0000 0000	0000 0000	PC+1
PSW	03H	0001 1xxx	000q quuu	uuuq quuu
P0	05H	-- xx xxxx	--uu uuuu	--uu uuuu
P2	06H	xxxx xxxx	uuuu uuuu	uuuu uuuu
P1	07H	xxxx xxxx	uuuu uuuu	uuuu uuuu
P3	08H	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCH	0AH	---0 0000	---0 0000	---u uuuu
INTCTL	0BH	0000 0000	0000 0000	uuuu uuuu
EIF1	0CH	0000 0000	0000 0000	uuuu uuuu
EIF2	0DH	0000 0000	0000 0000	uuuu uuuu
T1L	0EH	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1H	0FH	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CTL	10H	0000 0000	uuuu uuuu	uuuu uuuu
PWMCTL	15H	0000 0000	0000 0000	uuuu uuuu
PP1	16H	1111 1111	1111 1111	uuuu uuuu
BANK	17H	---- 0000	---- 0000	---- uuuu
ANSEH	1DH	0000 0000	0000 0000	uuuu uuuu
ADCDATA0H	1EH	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCCTL0	1FH	0000 0000	0000 0000	uuuu uuuu
OPTR	21H	1111 1111	1111 1111	uuuu uuuu
IP0	22H	---- -000	---- -000	---- -uuu
IP1	23H	0000 0000	0000 0000	uuuu uuuu
IP2	24H	0000 0000	0000 000	uuuu uuuu
TR0	25H	1111 1111	1111 1111	uuuu uuuu
TR2	26H	1111 1111	1111 111	uuuu uuuu
TR1	27H	1111 1111	1111 1111	uuuu uuuu
OSCSTA	28H	0110 --00	0110 --00	uuuu --uu
IP3	29H	000- ----	000- ----	uuu- ----
VRECAL1	2AH	0111 0111	0111 0111	uuuu uuuu
EIE1	2CH	0000 0000	0000 0000	uuuu uuuu
EIE2	2DH	0000 0000	0000 0000	uuuu uuuu
PCTL	2EH	---1 000x	---1 00uq	---u uuuu
OSCCTL	2FH	0010 0000	0010 0000	uuuu uuuu
OSCCAL0	30H	1000 0000	1000 0000	uuuu uuuu
ANSEL	31H	0000 0000	0000 0000	uuuu uuuu
PP2	32H	1111 1111	1111 1111	uuuu uuuu
PUR0	35H	1111 1-11	1111 1-11	uuuu u-uu
IOCL	36H	0000 0000	0000 0000	uuuu uuuu
OSCCAL1	37H	0000 -001	0000 -001	uuuu -uuu
NVMDATAH	38H	0000 0000	0000 0000	uuuu uuuu
NVMDATAL	39H	0000 0000	0000 0000	uuuu uuuu
NVMADDRH	3AH	0000 0000	0000 0000	uuuu uuuu
NVMADDRL	3BH	0000 0000	0000 0000	uuuu uuuu
NVMCTL0	3CH	---- x000	---- q000	---- uuuu
NVMCTL1	3DH	---- ----	---- ----	---- ----
ADCDATA0L	3EH	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCCTL1	3FH	0000 0000	0000 0000	uuuu uuuu
P0LR	45H	xxxx xxxx	uuuu uuuu	uuuu uuuu
P2LR	46H	xxxx xxxx	uuuu uuuu	uuuu uuuu
P1LR	47H	xxxx xxxx	uuuu uuuu	uuuu uuuu
P3LR	48H	xxxx xxxx	uuuu uuuu	uuuu uuuu
TR3	49H	1111 1111	1111 1111	uuuu uuuu
EIE3	4AH	000- ----	000- ----	uuu- ----
EIF3	4BH	000- ----	000- ----	uuu- ----

寄存器	地址	上电复位	RST 复位 欠压检测	中断唤醒 WDT 超时唤醒
OSCCAL2	4CH	0011 1111	0011 1111	uuuu uuuu
OSCCAL3	4DH	0011 1111	0011 1111	uuuu uuuu
BUZCTL	4EH	0000 0000	0000 0000	uuuu uuuu
PUR1	60H	1111 1111	1111 1111	uuuu uuuu
PUR2	61H	1111 1111	1111 1111	uuuu uuuu
PWM12L	62H	xxxx xxxx	uuuu uuuu	uuuu uuuu
PWM13L	63H	xxxx xxxx	uuuu uuuu	uuuu uuuu
PWM22L	65H	xxxx xxxx	uuuu uuuu	uuuu uuuu
PWM23L	66H	xxxx xxxx	uuuu uuuu	uuuu uuuu
INTEDGCTL	67H	11-- --0	11-- --0	uu-- --u
PWM14L	68H	xxxx xxxx	uuuu uuuu	uuuu uuuu
PWM24L	69H	xxxx xxxx	uuuu uuuu	uuuu uuuu
VBIASCTL	6BH	00-- --00	00-- --00	uu-- --uu
VBIAS1EN	6CH	0000 0000	0000 0000	uuuu uuuu
VBIAS2EN	6DH	0000 0000	0000 0000	uuuu uuuu
SSCICTL0	128H	0000 0000	0000 0000	uuuu uuuu
SSCICTL1	12AH	0000 0000	0000 0000	uuuu uuuu
SSCISTA	12BH	0000 0000	0000 0000	uuuu uuuu
SSCIBUFR	12CH	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSCIADD	12EH	0000 0000	0000 0000	uuuu uuuu
SSCIMSK		1111 1111	1111 1111	uuuu uuuu
WDTPS	12FH	---- 0100	---- 0100	---- uuuu
LEDCTL	140H	0000 0000	0000 0000	uuuu uuuu
LEDPRE	141H	0000 0000	0000 0000	uuuu uuuu
LEDDATA0	142H	0000 0000	0000 0000	uuuu uuuu
LEDDATA1	143H	0000 0000	0000 0000	uuuu uuuu
LEDDATA2	144H	0000 0000	0000 0000	uuuu uuuu
LEDDATA3	145H	0000 0000	0000 0000	uuuu uuuu
LEDDATA4	146H	0000 0000	0000 0000	uuuu uuuu
LEDDATA5	147H	0000 0000	0000 0000	uuuu uuuu
LEDDATA6	148H	0000 0000	0000 0000	uuuu uuuu
LEDDATA7	149H	0000 0000	0000 0000	uuuu uuuu
LEDOMS1	150H	0000 0000	0000 0000	uuuu uuuu
LEDODS0	155H	0000 0000	0000 0000	uuuu uuuu
LEDOMS0	157H	0000 0000	0000 0000	uuuu uuuu
LEDODS1	158H	0000 0000	0000 0000	uuuu uuuu
T4L	160H	0000 0000	0000 0000	uuuu uuuu
T4H	161H	0000 0000	0000 0000	uuuu uuuu
T4REL	162H	xxxx xxxx	uuuu uuuu	uuuu uuuu
T4REH	163H	xxxx xxxx	uuuu uuuu	uuuu uuuu
T4CTL	164H	0000 0000	0000 0000	uuuu uuuu
RC32KCAL	165H	0000 0000	0000 0000	uuuu uuuu

注：u=不变，x=未知，-=未实现位，读为 0；q=视具体条件而定。

表 15-2 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

13 休眠模式

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

为使这种方式下的电流消耗降至最低，应使所有 I/O 口状态确定，如果有的端口没有使用，最好设置为输入，接到 VDD 或 VSS 上，如果没用的端口悬空，应设置为输出，以确保 I/O 引脚没有耗散电流产生，其他在休眠时不用的外设都要关闭。

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

单片机进入休眠模式一段时间后由于工作的需要，要将单片机从休眠模式唤醒，在 KF8V327 中可通过以下方式将单片机从休眠模式唤醒：

1. $\overline{\text{RST}}$ 引脚上输入的外部复位
2. 看门狗定时器唤醒(如果 WDT 已被使能)
3. INT0/INT1/INT2 外部中断
4. T1 中断 (T1 异步计数模式)
5. P0 口电平变化中断

$\overline{\text{RST}}$ 引脚输入的复位信号在唤醒单片机的同时也将导致单片机复位。其它唤醒时将单片机从休眠模式唤醒，并不会导致复位。可通过状态寄存器中的 $\overline{\text{TO}}$ 和 $\overline{\text{PD}}$ 位来确定单片机唤醒的原因。上电时 $\overline{\text{PD}}$ 位将被置 1，而当器件从休眠模式唤醒时，该位将被清零。 $\overline{\text{TO}}$ 位则在 WDT 唤醒发生时被清零。

在使用后中断方式唤醒时，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 位被清零，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。

14 看门狗定时器

为了防止单片机在正常工作时程序跑飞，KF8V327 提供一个看门狗定时器，单片机正常工作时，当看门狗定时器定时时间达到超时时间后，会使单片机产生复位。

看门狗定时器使用片内看门狗专用 RC 振荡器，因此它无需外接任何器件，在休眠模式仍能正常运行。在正常运行时，WDT 超时事件将使单片机产生一次复位。如果单片机处于休眠模式，WDT 超时事件将唤醒单片机并使其继续执行 IDLE 后面的指令。通过将配置位 WDTEN 清零/置 1，可关闭/打开 WDT。如果配置位 WDTEN 清零，则可以通过寄存器 SWDTEN(PCTL<2>)软件清零/置 1，可关闭/打开 WDT。

WDT 周期:

如 WDT 框图所示：WDT 的时钟源为内部低频时钟（带校正功能的 32K），由 WDTCTL 寄存器的 WDTPS<3:0>选择预分频比（1:32 到 1:65536），共 12 档分频比。复位后 WDTPS<3:0>=0100，WDT 时基为 16ms。超时时间最小约为 1ms，最长约为 268s。由于温度、电源电压和工艺等的差异，不同器件之间的超时周期稍有不同。通过软件将 OPTR 寄存器的 PSA 位置 1，还可将 T0 预分频器分配给 WDT。设置 PS<2:0>选择预分频器的分频比，分频比可选择为：1/1、1/2、1/4、1/8、1/16、1/32、1/64、1/128。

为了防止在正常工作时看门狗超时复位，要在固定的时间内对看门狗定时器清零。执行 CWDT 和 IDLE 指令后会使看门狗从 0 开始重新计数，当看门狗定时器出现超时，状态寄存器中的 TO 位将被清零。

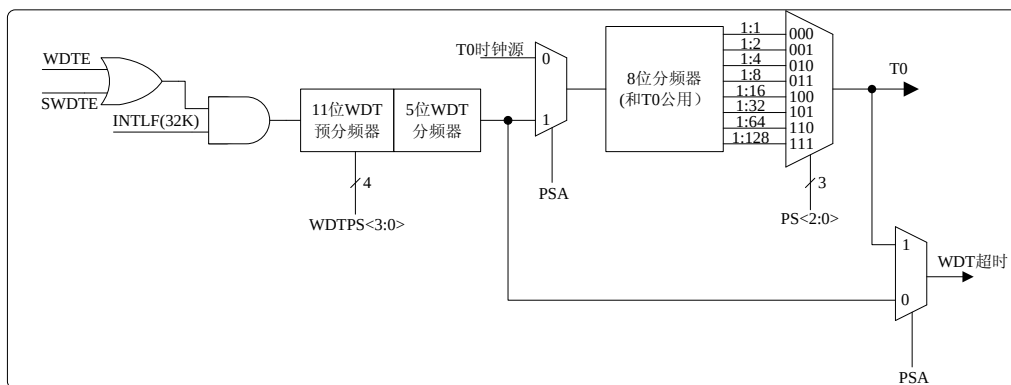


图 14.1 看门狗定时器框图

14.1 WDT 相关寄存器

14.1.1 WDT 预分频选择寄存器 WDTCTL

寄存器14.1: WDTPS: WDT预分频选择寄存器(地址:12FH)

	bit7				bit0			
复位值 ---- 0100	-	-	-	-	WDTPS3	WDTPS2	WDTPS1	WDTPS0
	R/W	U	U	U	R/W	R/W	R/W	R/W

WDTPS<3:0>: 看门狗定时器预分频比选择位

- 0000 = 1: 32
- 0001 = 1: 64
- 0010 = 1: 128
- 0011 = 1: 256
- 0100 = 1: 512 (默认) 16ms
- 0101 = 1: 1024
- 0110 = 1: 2048
- 0111 = 1: 4096
- 1000 = 1: 8192
- 1001 = 1: 16384
- 1010 = 1: 32768
- 1011 = 1: 65536
- 11xx = 保留

图注: R=可读 W=可写 -=未用 U=未实现位

15 电气规范

15.1 极限参数值

15-1 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	-30℃~85℃
2	储存温度	-40℃~150℃
3	VDD 相对于VSS 的电压	5.5V
4	MODE 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。

15.2 静态电流特性

表15-2芯片静态电流（IDD）特性

测试条件:25°C						
序号	测试条件		最小值	典型值	最大值	单位
	振荡频率	VDD (V)				
1	16MHz	5.0	-	1065	1900	μ A
		4.0	-	1060	1900	
		3.0	-	1047	1900	
		2.6	-	1039	1900	
2	8MHz	5.0	-	766	1400	
		4.0	-	763	1400	
		3.0	-	750	1400	
		2.6	-	745	1400	
3	4MHz	5.0	-	611	1100	
		4.0	-	608	1100	
		3.0	-	597	1100	
		2.6	-	591	1100	
4	2MHz	5.0	-	533	1000	
		4.0	-	530	1000	
		3.0	-	519	1000	
		2.6	-	514	1000	
5	1MHz	5.0	-	494	900	
		4.0	-	493	900	
		3.0	-	482	900	
		2.6	-	477	900	
6	500 kHz	5.0	-	475	900	
		4.0	-	472	900	
		3.0	-	461	900	
		2.6	-	456	900	
7	125kHz	5.0	-	465	850	
		4.0	-	462	850	
		3.0	-	452	850	
		2.6	-	447	850	
8	62.5kHz	5.0	-	458	850	
		4.0	-	455	850	
		3.0	-	445	850	
		2.6	-	440	850	

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

2: 供电电流主要随工作电压和频率而变化。其它因素，如I/O 引脚负载和开关速率、内部代码执行模式和温度也会影响电流消耗。

15.3 外设电流特性

表15-3芯片外设电流特性

测试条件:25°C							
序号	测试参数	测试条件		最小值	典型值	最大值	单位
			VDD(V)				
1	休眠电流(IPD)	WDT、BOR 等外设被禁止	5.0	—	2.03	3.12	μ A
			4.0	—	1.46	2.34	
			3.0	—	0.93	1.69	
			2.6	—	0.74	1.43	
2	WDT 电流(IWDT)		5.0	—	2.71	3.9	
			4.0	—	2.05	3.12	
			3.0	—	1.45	2.21	
			2.6	—	1.20	1.95	
3	欠电压复位电流(ILVR)		5.0	—	9.7	13.9	
			4.0	—	7.3	10.7	
			3.0	—	5.0	7.7	
			2.6	—	3.9	6.2	
4	弱上拉电流(IPUR)	单个 IO 口	5.0	—	156	207	
			4.0	—	98	133	
			3.0	—	48	66	
			2.6	—	28	41	

- 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。

15.4 I/O 端口电平和芯片供电电压特性

表 15-4 芯片 IO 端口电平特性

工作温度 $-30^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$						
符号	参数说明	测试条件	最小值	典型值	最大值	单位
V _{IL}	输入低电平 I/O引脚 采用TTL缓冲器		V _{SS}	—	0.15VDD	V
	采用施密特缓冲触 发器		V _{SS}	—	0.2VDD	V
V _{IH}	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

表 15-5 芯片供电电压特性

工作温度 $-30^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$						
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		2.6	—	5.5	V
V _{LVR}	VDD 起始电压确保 能够产生欠压复位信 号		2.2	—	2.6	V
V _{POR}	VDD 起始电压确保 能够产生内部上电复 位信号		2.2	—	2.6	V

15.5 A/D 转换器（ADC）特性

表 15-6 A/D 转换器（ADC）特性

工作温度 $-30^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$						
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_{IL}^*	积分误差	$V_{REF} = 5V, VDD=5.0V$	–2	–	+2	LSB
E_{DL}^*	微分误差	$V_{REF} = 5V, VDD=5.0V$	–1	–	+1	LSB
E_{OFF}^*	失调误差	$V_{REF} = 5V, VDD=5.0V$	–	3	–	LSB
E_{GN}^*	增益误差	$V_{REF} = 5V, VDD=5.0V$	–1.5	0.5	+1.5	LSB
V_{REF}^*	参考电压	保证1个LSb的精度	2	–	VDD	V
V_{AIN}^*	满量程范围		V_{SS}	–	V_{REF}	V
TCNV*	AD转换时间		–	13	–	TAD

*该数据为设计值

16 直流特性图表

备注:某些图表中的数据超出了规定的工作范围(即超出了规定的VDD 范围), 这些图表仅供参考, 器件只有在规定的范围下工作才可以确保正常运行。

图16-1: 不同VDD 时典型IDD — SCLK 关系曲线图

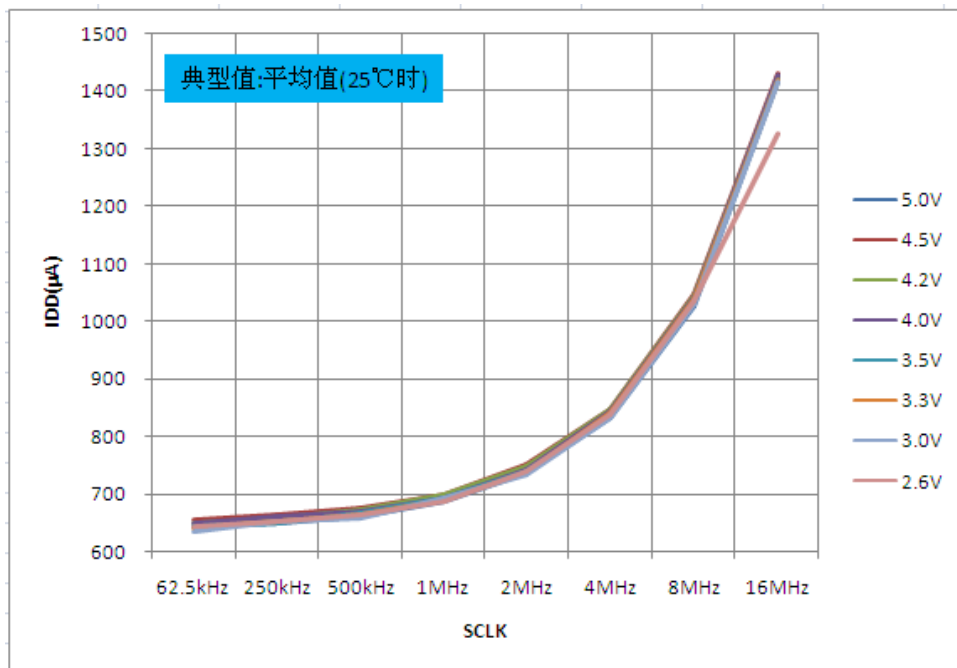


图16-2: 不同VDD 时典型IPD — VDD 关系曲线图

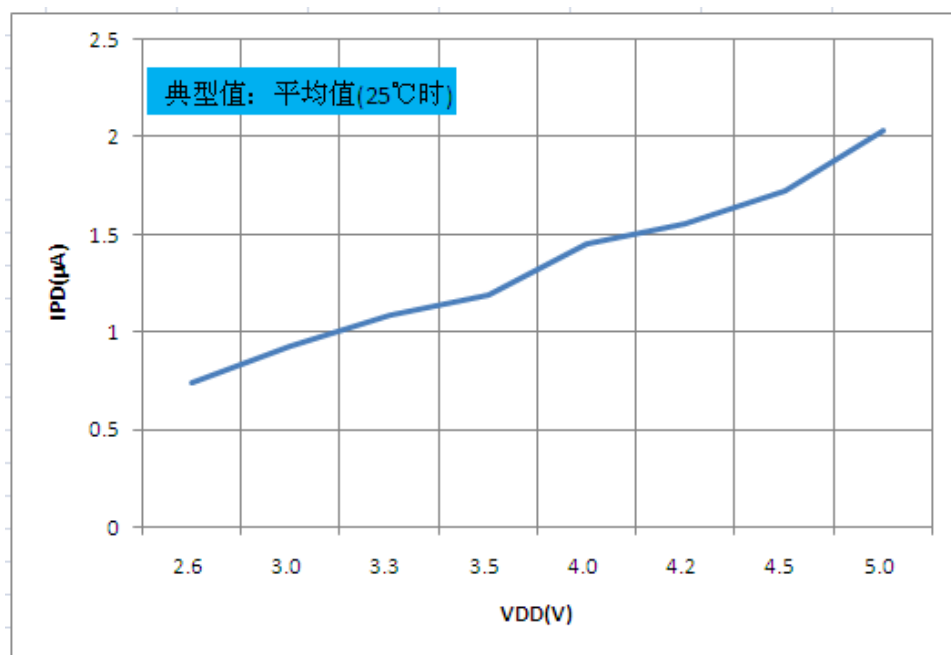


图16-3: 弱上拉电流 I_{PUR} — V_{DD} 关系曲线图

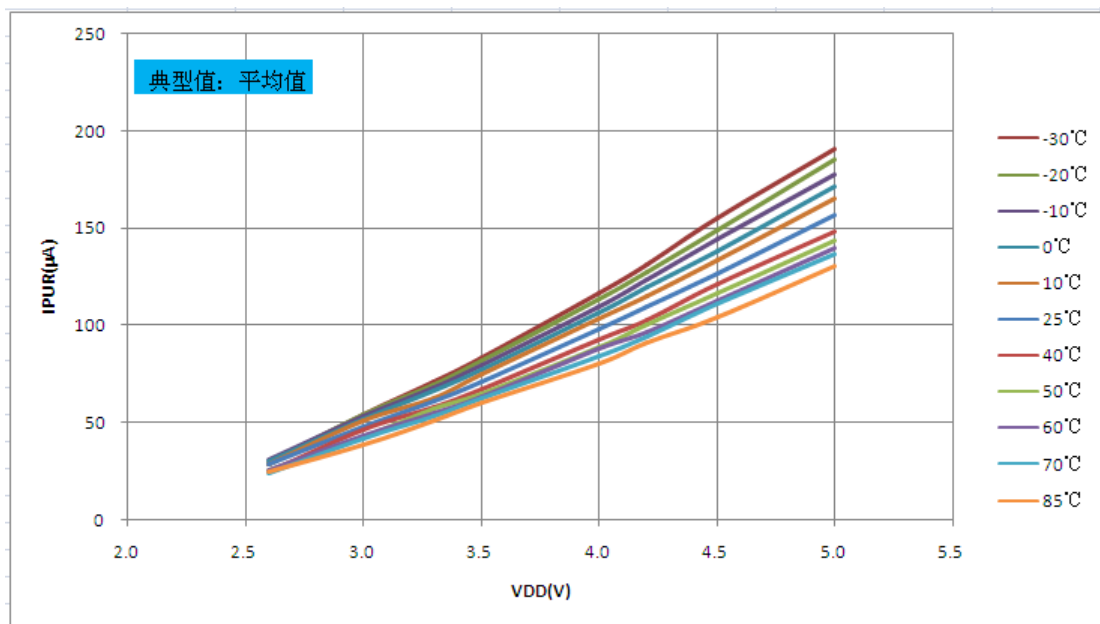
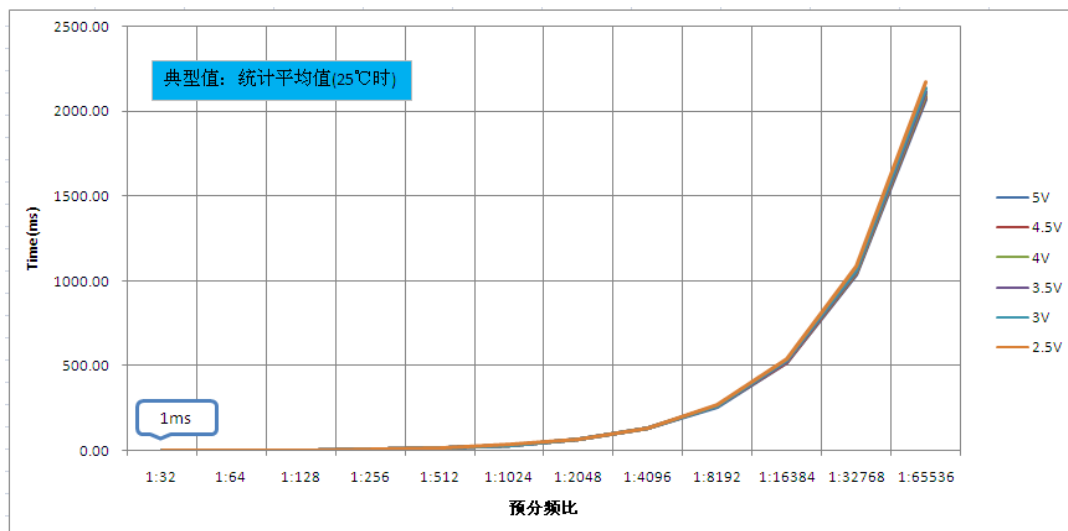


图16-4: 不同 V_{DD} 时看门狗WDTCTL预分频比 — 周期关系曲线图



备注:1/32 分频时看门狗周期为 1ms。

图16-5:看门狗电流 I_{WDT} — VDD关系曲线图

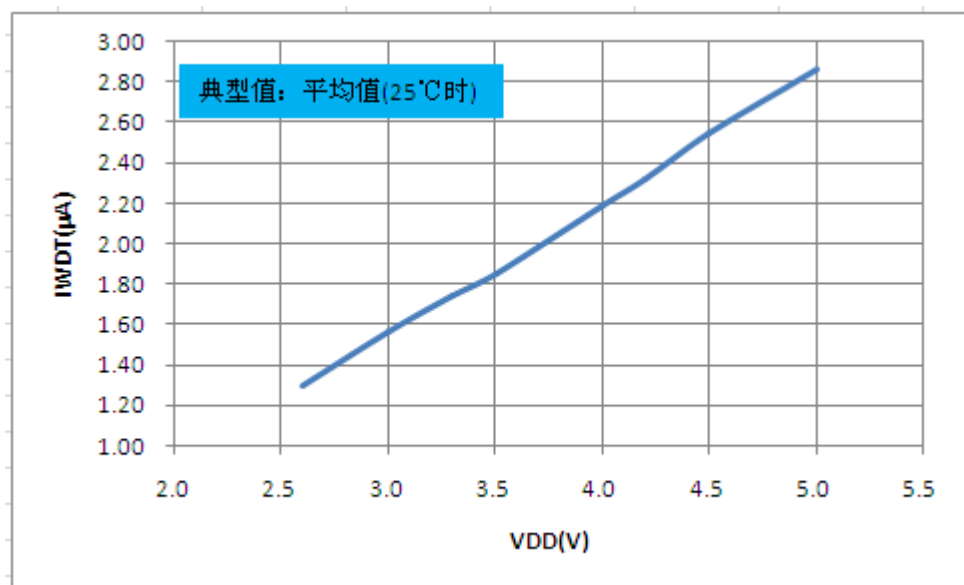


图 16-6:欠压复位电流 I_{LVR} — VDD 关系曲线图

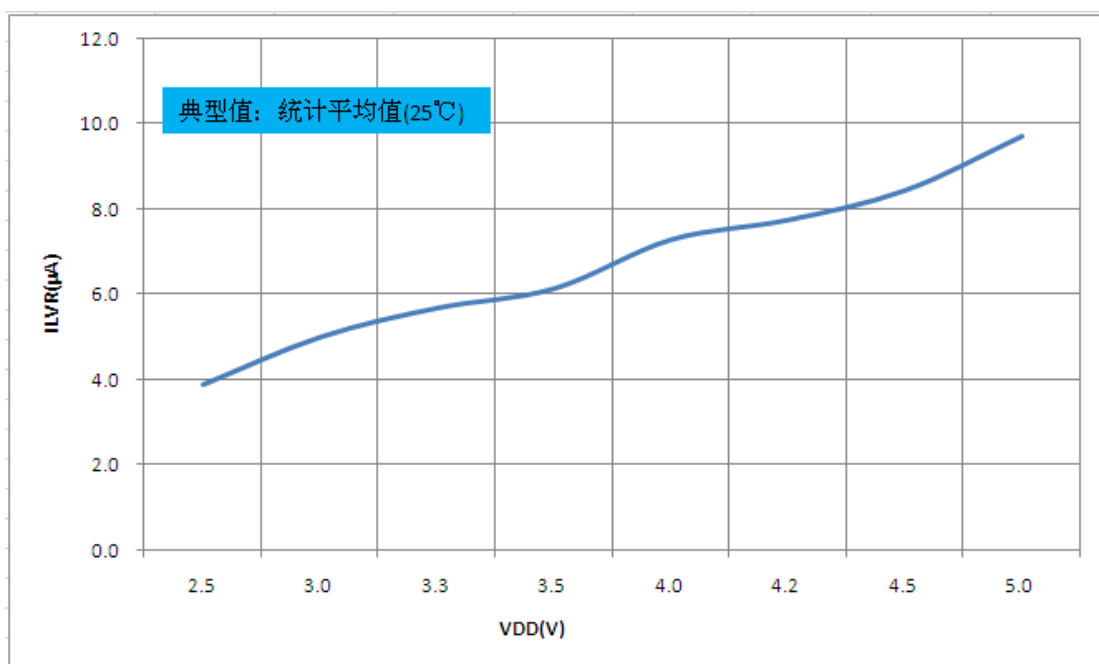


图16-7:不同温度时 V_{OH} — I_{OH} 关系曲线图 ($V_{DD} = 5.0V$)

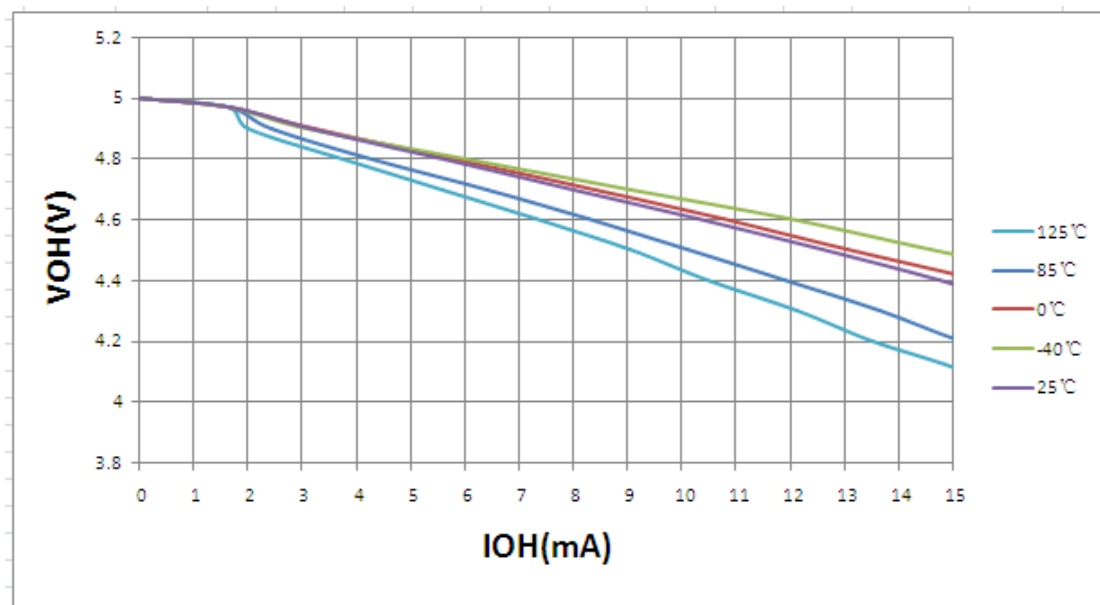


图16-8:不同温度时 V_{OH} — I_{OH} 关系曲线图 ($V_{DD} = 3.0V$)

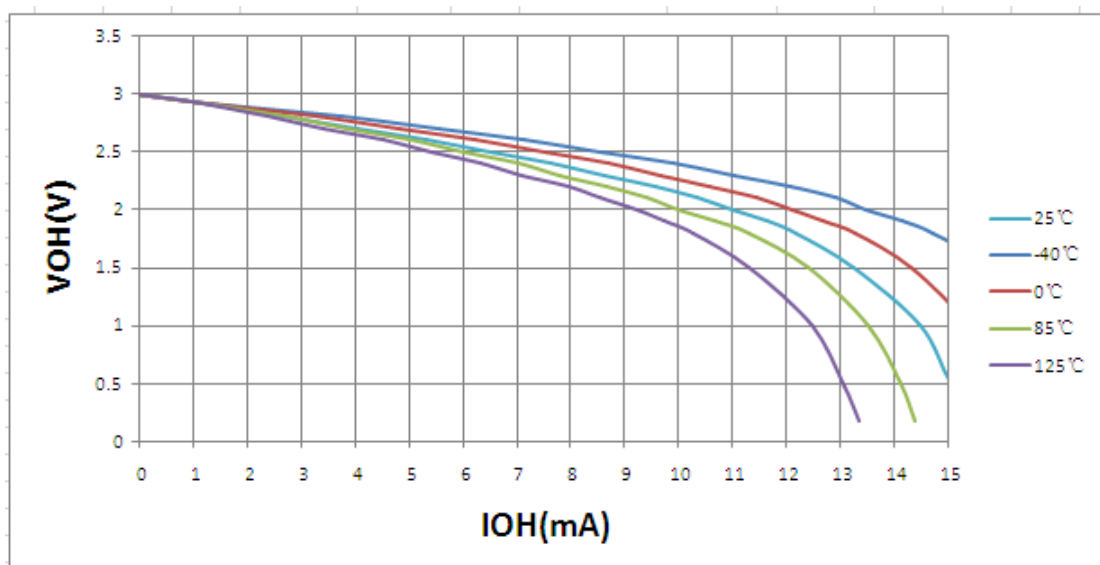


图16-9不同温度时VOL — IOL 关系曲线图 (VDD = 5.0V)

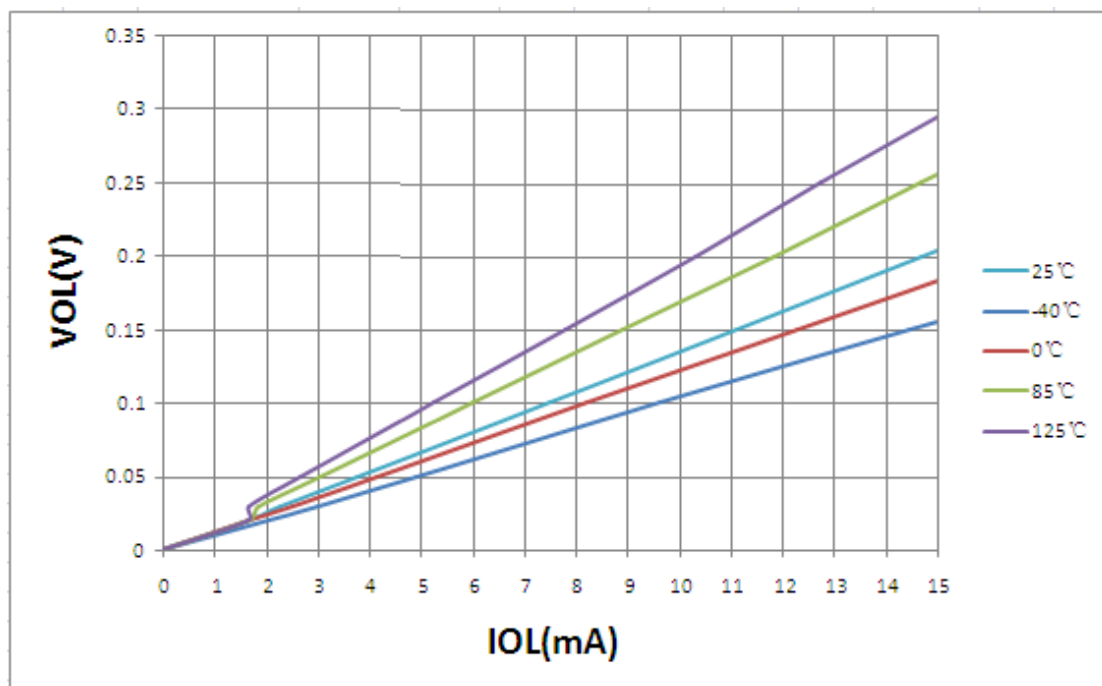
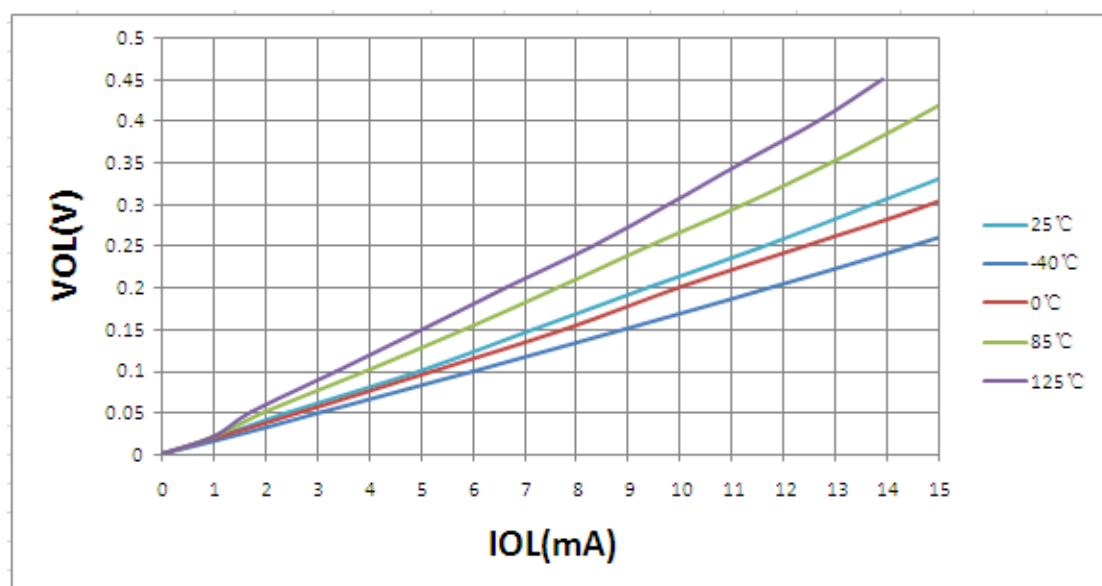
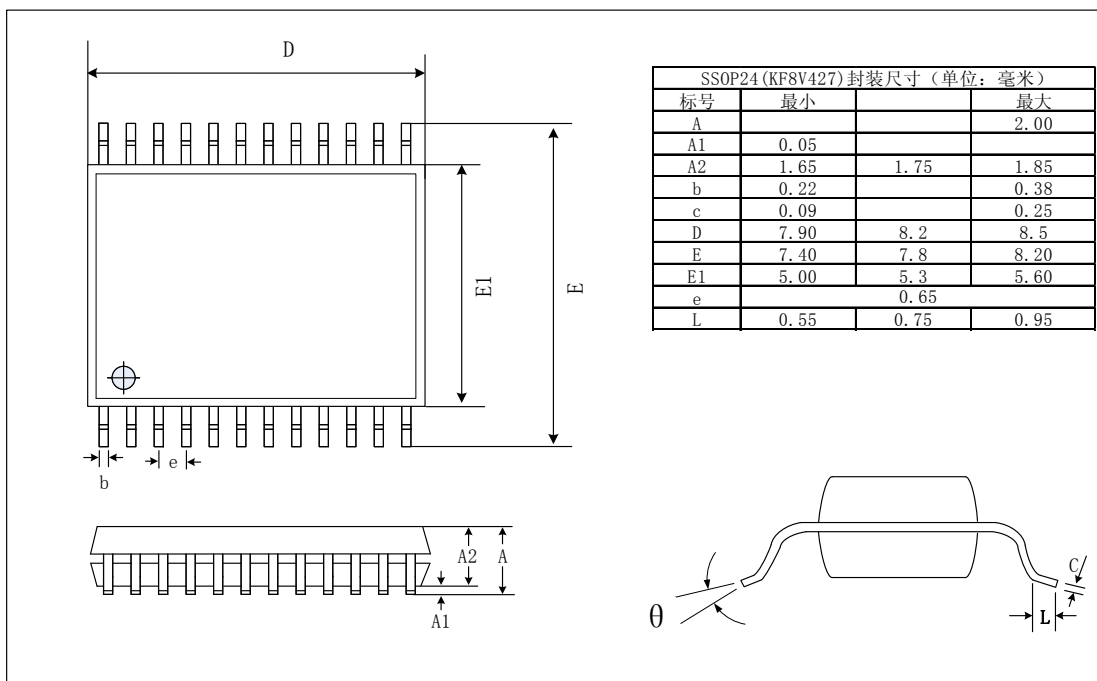


图16-10:不同温度时VOL — IOL 关系曲线图 (VDD = 3.0V)

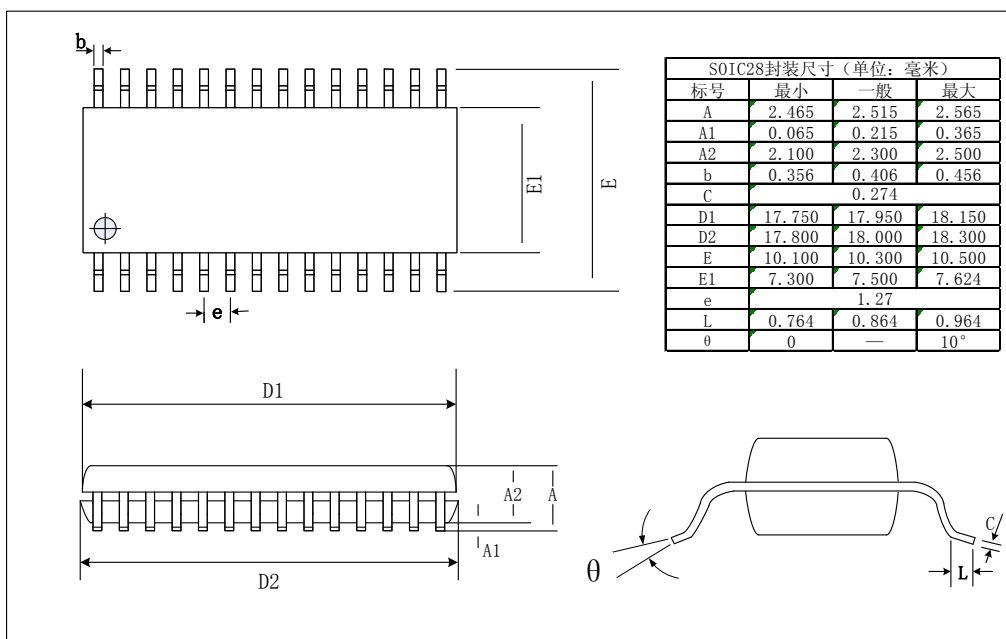


17 封装信息

24 脚 SSOP 封装



28 脚 SOIC 封装



附录 1 K8V327 SFR地址映射

地址	Bank0	地址	Bank1
00H	-	100H	-
01H	T0	101H	T0
02H	PCL	102H	PCL
03H	PSW	103H	PSW
04H	-	104H	-
05H	P0	105H	-
06H	P2	106H	-
07H	P1	107H	-
08H	P3	108H	-
09H	-	109H	-
0AH	PCH	10AH	PCH
0BH	INTCTL	10BH	-
0CH	EIF1	10CH	-
0DH	EIF2	10DH	-
0EH	T1L	10EH	-
0FH	T1H	10FH	-
10H	T1CTL	110H	-
11H	-	111H	-
12H	-	112H	-
13H	-	113H	-
14H	-	114H	-
15H	PWMCTL	115H	-
16H	PP1	116H	-
17H	BANK	117H	BANK
18H	-	118H	-
19H	-	119H	-
1AH	-	11AH	-
1BH	-	11BH	-
1CH	-	11CH	-
1DH	ANSEH	11DH	-
1EH	ADCDATA0H	11EH	-
1FH	ADCCTL0	11FH	-
20H	-	120H	
21H	OPTR	121H	
22H	IP0	122H	
23H	IP1	123H	
24H	IP2	124H	
25H	TR0	125H	
26H	TR2	126H	
27H	TR1	127H	-
28H	OSCSTA	128H	SSCCTL0
29H	IP3	129H	-
2AH	VRECAL1	12AH	SSCCTL02
2BH	-	12BH	SSCISTA
2CH	EIE1	12CH	SSCIBUFR
2DH	EIE2	12DH	-
2EH	PCTL	12EH	SSCIMSK
2FH	OSCCTL	12FH	WDTPS

30H	OSCCAL0	130H	
31H	ANSEL	131H	
32H	PP2	132H	
33H	-	133H	
34H	-	134H	
35H	PUR0	135H	
36H	IOCL	136H	
37H	OSCCAL1	137H	
38H	NVMDATAH	138H	
39H	NVMDATAL	139H	
3AH	NVMADDRH	13AH	
3BH	NVMADDRL	13BH	
3CH	NVMCTL0	13CH	
3DH	NVMCTL1	13DH	
3EH	ADCDATA0L	13EH	
3FH	ADCCTL1	13FH	
40H	-	140H	LEDCTL
41H	-	141H	LEDPRE
42H	-	142H	LEDDATA0
43H	-	143H	LEDDATA1
44H	-	144H	LEDDATA2
45H	P0LR	145H	LEDDATA3
46H	P2LR	146H	LEDDATA4
47H	P1LR	147H	LEDDATA5
48H	P3LR	148H	LEDDATA6
49H	TR3	149H	LEDDATA7
4AH	EIE3	14AH	-
4BH	EIF3	14BH	-
4CH	OSCCAL2	14CH	-
4DH	OSCCAL3	14DH	-
4EH	BUZCTL	14EH	-
4FH	-	14FH	-
50H	ADCDATA1H	150H	LEDOMS1
51H	ADCDATA1L	151H	-
52H	-	152H	-
53H	-	153H	-
54H	-	154H	-
55H	-	155H	LEDODS0
56H	-	156H	-
57H	-	157H	LEDOMS0
58H	ADCINTCTL	158H	LEDODS1
59H	ADDTATA2H	159H	VRECAL2
5AH	ADCDATA2L	15AH	VRECAL3
5BH	-	15BH	-
5CH	-	15CH	LEDLUM
5DH	-	15DH	-
5EH	ADCDATA3H	15EH	-
5FH	-	15FH	-
60H	-	160H	T4L
61H	-	161H	T4H
62H	PWM11L	162H	T4REL
63H	PWM12L	163H	T4REH
64H	-	164H	T4CTL
65H	PWM21L	165H	RC32KCAL

66H	PWM22L	166H	-
67H	INTEDGCTL	167H	-
68H	-	168H	-
69H	-	169H	-
6AH	ADCDATA3L	16AH	-
6BH	VBIASCTL	16BH	-
6CH	VBIAS1EN	16CH	-
6DH	VBIAS2EN	16DH	-
6EH	PUR1	16EH	-
6FH	PUR2	16FH	-

注：“-”的存储单元代表系统保留。

附录 2 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	-	-	-	$\overline{TO}$	$\overline{PD}$	Z	DC	CY	0001 1xxx
05H	P0	-	-	P05	P04	P03	P02	P01	P00	-- xx xxxx
06H	P2	-	-	P25	P24	P23	P22	P21	P20	xxxx xxxx
07H	P1	-	-	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	TOIE	INT0IE	POIE	TOIF	INT0IF	POIF	0000 0000
0CH	EIF1	-	ADIF	INT2IF	INT1IF	-	PWM2IF	-	T1IF	0000 0000
0DH	EIF2	-	-	-	-	-	-	BCLIF	SSCIF	0000 0000
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	$\overline{T1SY}$	T1CS	T1ON	0000 0000
15H	PWMCTL	-	PWM22ON	PWM21ON	-	-	PWM12ON	PWM11ON	-	0000 0000
16H	PP1	PWM1x 周期寄存器								1111 1111
17H	BANK	-	-	-	-	PR3	PR2	PR1	PR0	---- 0000
1DH	ANSEH	-	-	ANS13	ANS12	ANS11	ANS10	ANS9	ANS8	0000 0000
1EH	ADCDATA0H	ADC 数据寄存器 0 高字节								xxxx xxxx
1FH	ADCCTL0	ADLR	-	CHS3	CHS2	CHS1	CHS0	START	ADEN	0000 0000
21H	OPTR	$\overline{PUPH}$	INT0SE	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111
22H	IP0	-	-	-	-	-	PT0	PINT0	PP0	---- -000
23H	IP1	-	PADC	PINT2	PINT1	-	PPWM2	-	PT1	0000 0000
24H	IP2	-	-	-	-	-	-	PBCL	PSSCI	0000 0000
25H	TR0	-	-	TR05	TR04	TR03	TR02	TR01	TR00	0000 0000
26H	TR2	-	-	TR25	TR24	TR23	TR22	TR21	TR20	0000 0000
27H	TR1	-	-	TR15	TR14	TR13	TR12	TR11	TR10	1111 1111
28H	OSCSTA	-	OSTS	HTS	LTS	-	-	SCF1	SCF0	0110 -00
29H	IP3	PT4	-	POSCFAIL	-	-	-	-	-	000- ----
2AH	VRECAL1	内部参考电压校准寄存器 1								0111 0111
2CH	EIE1	-	ADIE	INT2IE	INT1IE	-	PWM2IE	-	T1IE	0000 0000
2DH	EIE2	-	-	-	-	-	-	BCLIE	SSCIE	0000 0000
2EH	PCTL	-	-	-	SLVREN	IPEN	SWDTEN	POR	LVR	---1 000x
2FH	OSCCCTL	CLKOE	IRCS2	IRCS1	IRCS0	SCS1	SCS0	IESO	FSCM	0010 0000
30H	OSCCAL0	晶振校准值寄存器 0								1000 0000
31H	ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	0000 0000
32H	PP2	PWM2x 周期寄存器								1111 1111
35H	PUR0	-	-	PUR05	PUR04	PUR03	-	PUR01	PUR00	1111 1-11
36H	IOCL	-	-	IOCL5	IOCL4	IOCL3	IOCL2	IOCL1	IOCL0	0000 0000
37H	OSCCAL1	晶振校准值寄存器 1								0000 -001
38H	NVMDATAH	NVMDATAH 数据高 8 位								0000 0000
39H	NVMDATAL	NVMDATAL 数据低 8 位								0000 0000
3AH	NVMADDRH	NVMADDRH 地址指针高 8 位								0000 0000
3BH	NVMADDRL	NVMADDRL 地址指针低 8 位								0000 0000
3CH	NVMCTL0	NVM 控制寄存器 0								---- x000
3DH	NVMCTL1	NVM 控制寄存器 1								---- ----
3EH	ADCDATA0L	ADC 数据寄存器 0 低字节								xxxx xxxx
3FH	ADCCTL1	ADCALEN	ADCS2	ADCS1	ADCS0	VCFG1	VCFG0	-	ADCIM	0000 0000
45H	P0LR	-	-	P0LR5	P0LR4	P0LR3	-	P0LR1	P0LR0	xxxx xxxx
46H	P2LR	-	-	P2LR5	P2LR4	P2LR3	P2LR2	P2LR1	P2LR0	xxxx xxxx
47H	P1LR	-	-	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	-	OSCFALIE	-	-	-	-	-	000- ----
4BH	EIF3	T4IF	-	OSCFALIF	-	-	-	-	-	000- ----
4CH	OSCCAL2	晶振校准值寄存器 2								0011 1111
4DH	OSCCAL3	晶振校准值寄存器 3								0011 1111
4EH	BUZCTL	-	-	-	-	-	T4_BUZOE	-	-	0000 0000
60H	PUR1	-	-	PUR15	PUR14	PUR13	PUR12	PUR11	PUR10	1111 1111
61H	PUR2	-	-	PUR25	PUR24	PUR23	PUR22	PUR21	PUR20	1111 1111
62H	PWM11L	PWM11 占空比寄存器								xxxx xxxx



KF8V327 数据手册 V1. 1

地址	名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	复位初值
63H	PWM12L	PWM12 占空比寄存器								xxxx xxxx
65H	PWM21L	PWM21 占空比寄存器								xxxx xxxx
66H	PWM22L	PWM22 占空比寄存器								xxxx xxxx
67H	INTEDGCTL	INT2SE	INT1SE	-	-	-	-	-	T1CLKEN	11-- --0
6AH	ADCDATA3L	ADC 数据寄存器 3 低字节								0000 0000
6BH	VBIASCTL	VBIASEN	-	-	-	-	-	ISEL1	ISEL0	00-- --00
6CH	VBIAS1EN	VBS17EN	VBS16EN	VBS15EN	VBS14EN	VBS13EN	VBS12EN	VBS11EN	VBS10EN	0000 0000
6DH	VBIAS2EN	VBS27EN	VBS26SEN	VBS25 EN	VBS24EN	VBS23EN	VBS22EN	VBS21EN	VBS20EN	0000 0000
128H	SSCICTL0	SSCIWCFL	SSCIOV	SSCIEN	SSCICKP	SSCIMOD3	SSCIMOD2	SSCIMOD1	SSCIMOD0	0000 0000
12AH	SSCICTL1	SSCICALLEN	SSCIACKSTA	SSCIACKDAT	SSCIACKEN	SSCIRCEN	STOPEN	RESTARTEN	STARTEN	0000 0000
12BH	SSCISTA	SAMPLE	CKEGE	SSCIDA	SSCISTOP	SSCISTART	SSCIRW	SSCIUA	SSCIBUF	0000 0000
12CH	SSCIBUFR	SSCI 数据接收缓冲/发送寄存器								xxxx xxxx
12EH	SSCIADD	SSCI 的 I2C 地址寄存器								0000 0000
	SSCIMSK	SSCIMSK7	SSCIMSK6	SSCIMSK5	SSCIMSK4	SSCIMSK3	SSCIMSK2	SSCIMSK1	SSCIMSK0	1111 1111
12FH	WDTPS	-	-	-	-	WDTPS3	WDTPS2	WDTPS1	WDTPS0	---- 0100
140H	LEDCTL	LEDEN	-	-	DIGEN2	DIGEN1	DIGEN0	PNS1	PNS0	0000 0000
141H	LEDPRE	LEDCKS13	LEDCKS12	LEDCKS11	LEDCKS10	LEDCKS03	LEDCKS02	LEDCKS01	LEDCKS00	0000 0000
142H	LEDDATA0	DAT07	DAT06	DAT05	DAT04	DAT03	DAT02	DAT01	DAT00	0000 0000
143H	LEDDATA1	DAT17	DAT16	DAT15	DAT14	DAT13	DAT12	DAT11	DAT10	0000 0000
144H	LEDDATA2	DAT27	DAT26	DAT25	DAT24	DAT23	DAT22	DAT21	DAT20	0000 0000
145H	LEDDATA3	DAT37	DAT36	DAT35	DAT34	DAT33	DAT32	DAT31	DAT30	0000 0000
146H	LEDDATA4	DAT47	DAT46	DAT45	DAT44	DAT43	DAT42	DAT41	DAT40	0000 0000
147H	LEDDATA5	DAT57	DAT56	DAT55	DAT54	DAT53	DAT52	DAT51	DAT50	0000 0000
148H	LEDDATA6	DAT67	DAT66	DAT65	DAT64	DAT63	DAT62	DAT61	DAT60	0000 0000
149H	LEDDATA7	DAT77	DAT76	DAT75	DAT74	DAT73	DAT72	DAT71	DAT70	0000 0000
150H	LEDOMS1	LEDOM17	LEDOM16	LEDOM15	LEDOM14	LEDOM13	LEDOM12	LEDOM11	LEDOM10	0000 0000
155H	LEDODS0	LEDOD07	LEDOD06	LEDOD05	LEDOD04	LEDOD03	LEDOD02	LEDOD01	LEDOD00	0000 0000
157H	LEDOMS0	LEDOM07	LEDOM06	LEDOM05	LEDOM04	LEDOM03	LEDOM02	LEDOM01	LEDOM00	0000 0000
158H	LEDODS1	LEDOD17	LEDOD16	LEDOD15	LEDOD14	LEDOD13	LEDOD12	LEDOD11	LEDOD10	0000 0000
15CH	LEDLUM	LEDBR	-	-	-	FRAMEF	LUM2	LUM1	LUM0	0--- 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								xxxx xxxx
163H	T4REH	T4 重载设置寄存器高 8 位								xxxx xxxx
164H	T4CTL	T4REN	HSPEN	T4CKS1	T4CKS0	LSPEN	T4CS1	T4CS0	T4ON	0000 0000
165H	RC32KCAL	内部 32K 振荡器频率校准寄存器								0000 0000

注:“-”表示未用的存储单元 “x”表示不定

附录 3 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 清零	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
XOR Rm,dir	0001_1rr0_ffff_ffff	$Rm \leftarrow (Rm) \oplus (dir)$	1	Z

助记符、操作数	指令格式	指令说明	周期	影响标志
XOR dir,Rm	0001_1rr1_ffff_ffff	$dir \leftarrow (dir) \oplus (Rm)$	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 位清零	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 位清零	1	
SET Rn,b	1111_1110_01bb_brrr	将 Rn 的 b 位置 1	1	
转移指令				
DECRJZ dir	0000_1000_ffff_ffff	$R0 \leftarrow (dir) - 1$, 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 4 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
04H	IR	Instruction 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
15H	PWMCTL	Pulse-Width Modulation Control register
16H	PP1	Pulse-Width Modulation Periods register 1
17H	BANK	BANK
1DH	ANSEH	Analog channel Selection register High
1EH	ADCDATA0H	Analog Digital Convert Data 0 High register
1FH	ADCCCTL0	Analog Digital Convert Control register 0
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	VRECAL1	Reference Voltage Calibration register 1
2CH	EIE1	Enable Interrupt Enable register 1
2DH	EIE2	Enable Interrupt Enable register 2
2EH	PCTL	Power Control register
2FH	OSCCTL	Oscillator Control register
30H	OSCCAL0	Oscillator Calibration register 0
31H	ANSEL	Analog channel Selection register Low
32H	PP2	Pulse-Width Modulation Periods register 2
35H	PUR0	Pull-Up Register 0
36H	IOCL	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
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	BUZCTL	Buzzer Control register
50H	ADCDATA1H	Analog Digital Convert Data register 1 High
51H	ADCDATA1L	Analog Digital Convert Data register 1 Low
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
5EH	ADCDATA3H	Analog Digital Convert Data register 3 High
62H	PWM11L	Pulse-Width Modulation 11 duty cycle register Low
63H	PWM12L	Pulse-Width Modulation 12 duty cycle register Low
65H	PWM21L	Pulse-Width Modulation 21 duty cycle register Low
66H	PWM22L	Pulse-Width Modulation 22 duty cycle register Low
67H	INTEDGCTL	Interrupt Edge Control register
6AH	ADCDATA3L	Analog Digital Convert Data register 3 Low
6BH	VBIASCTL	Voltage Bias Control register
6CH	VBIAS1EN	Voltage Bias 1 Enable register
6DH	VBIAS2EN	Voltage Bias 2 Enable register
6EH	PUR1	Pull-Up Resistor 1
6FH	PUR2	Pull-Up Resistor 2

BANK1

地址	名称	全称
101H	T0	Timer 0 register
102H	PCL	Program Counter Low register
103H	PSW	Program Status Word register
10AH	PCH	Program Counter High
117H	BANK	Bank
128H	SSICCTL0	Synchronous Serial Communication Interface Control register 0
12AH	SSICCTL1	Synchronous Serial Communication Interface Control register 0
12BH	SSICSTA	Synchronous Serial Communication Interface Status register
12CH	SSICBUFR	Synchronous Serial Communication Interface Buffer Register
12EH	SSICMSK	Synchronous Serial Communication Interface Mask register
12FH	WDTPS	Watchdog Pre-divider Selection register
140H	LEDCTL	Light Emitting Diode Control register
141H	LEDPRE	Light Emitting Diode Control Pre-divider register
142H	LEDDATA0	Light Emitting Diode Data register 0
143H	LEDDATA1	Light Emitting Diode Data register 1
144H	LEDDATA2	Light Emitting Diode Data register 2
145H	LEDDATA3	Light Emitting Diode Data register 3
146H	LEDDATA4	Light Emitting Diode Data register 4
146H	LEDDATA5	Light Emitting Diode Data register 5
147H	LEDDATA6	Light Emitting Diode Data register 6
148H	LEDDATA7	Light Emitting Diode Data register 7
150H	LEDOMS1	Light Emitting Diode Output Model Selection register 1
155H	LEDODS0	Light Emitting Diode Output Drain Selection register 0

157H	LEDOMS0	Light Emitting Diode Output Model Selection register 0
158H	LEDODS1	Light Emitting Diode Output Drain Selection register 1
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

附录 5 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	—

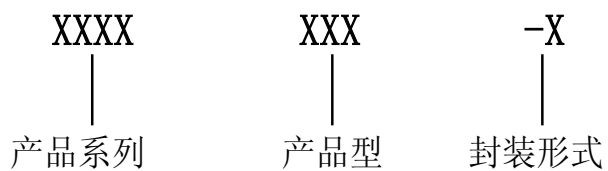
	误差(%)	—	—	0.16	0.16	0. 00	0.16	-3.55	—
	EUBRGL 值(十六进制)	—	—	CFH	33H	2FH	19H	08H	—
4.000MHz	实际波特率	—	1202	2404	9615	10417	19.23K	—	—
	误差(%)	—	0.16	0.16	0.16	0. 00	0.16	—	—
	EUBRGL 值(十六进制)	—	CFH	67H	19H	17H	0CH	—	—
2.000MHz	实际波特率	—	1202	2404	9615	10417	—	—	—
	误差(%)	—	0.16	0.16	0.16	0. 00	—	—	—
	EUBRGL 值(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: KF8V = KF 系列闪存 8 位单片机

产品型号: 327 = 327 型

封装形式:

SN = SOIC28

OM = SSOP24

示例:

- 1) K8V327-SN 表示 KF 系列 SOIC28 封装单片机

版本说明

版本	时间	更新描述	更新页码
1.0		更新数据手册里的一些 bug	
1.1	2016.07.26	修改了版本订购信息和文档中部分叙述	

ROSH认证

本产品已通过 ROSH 检测。

声明及销售网络

销售及服务网点

上海 TEL:021-50275927

地址 上海浦东张江科苑路 201 号