

SPMC65P2708A/2716A

Micro-Controllers (OTP) With ADC

May. 07, 2007

Version 1.1

中文版

目录

	页
MICRO-CONTROLLERS (OTP) WITH ADC	1
1 总述	7
2 特性	7
3 芯片结构概览	9
4 信号描述	10
4.1 管脚说明	10
4.2 管脚分配 (TOP VIEW).....	12
4.2.1 PDIP32 封装.....	12
4.2.2 PDIP42 封装.....	12
4.2.3 SSOP48 封装.....	12
5 功能介绍	13
5.1 CPU	13
5.1.1 简介.....	13
5.1.2 CPU 寄存器.....	13
5.2 存储器	16
5.2.1 简介.....	16
5.2.2 存储空间分配.....	16
5.2.3 硬件控制寄存器.....	17
(1) \$0000~000F: I/O 端口及中断控制寄存器	18
(2) \$0010~\$001E: Timer/PWM 设置 & 数据寄存器	19
(3) \$0026~002D: 中断和 ADC 控制寄存器.....	22
(4) \$0030~\$0036: 特殊功能控制寄存器 (需要连续写两次才能成功)	23
(5) \$0038~\$003C: SPI 控制寄存器	24
(6) \$0040~\$0044: I/O 端口 E 控制寄存器	25
(7) \$0046~\$0049: UART 控制寄存器.....	25
(8) \$0058~\$005D: 端口数据锁存寄存器	25
5.2.4 系统运行控制寄存器.....	26
5.2.5 芯片配置寄存器.....	27
5.2.6 用户信息块.....	27
5.3 系统时钟输入	27
5.4 省电模式	27
5.4.1 简介.....	27
5.4.2 STOP 模式.....	28
5.4.3 HALT 模式.....	29
5.5 中断	31
5.5.1 简述.....	31
5.5.2 外部中断.....	31
5.5.3 非屏蔽中断.....	31
5.5.4 其它中断.....	31
5.5.5 中断寄存器.....	33
(1) 中断标志寄存器 0 (P_INT_Flag0, \$000C).....	33

(2) 中断标志寄存器 1(P_INT_Flag1, \$000E).....	34
(3) 中断标志寄存器 2(P_INT_Flag2, \$0026).....	34
(4) 中断控制寄存器 0 (P_INT_Ctrl0, \$000D).....	35
(5) 中断控制寄存器 1 (P_INT_Ctrl1, \$000F).....	36
(6) 中断控制寄存器 2 (P_INT_Ctrl2, \$0027).....	36
(7) NMI 设置寄存器(P_NMI, \$7FE3).....	36
(8) IRQ 设置寄存器 0 (P_IRQ_Opt0, \$0033).....	37
(9) IRQ 设置寄存器 1(P_IRQ_Opt1, \$0034).....	37
5.6 复位.....	39
5.6.1 简述.....	39
5.6.2 上电复位 (POR).....	39
5.6.3 外部复位 (ERST).....	39
5.6.4 低电压复位 (LVR).....	39
5.6.5 看门狗复位 (WDR).....	39
5.6.6 非法地址复位 (IAR).....	40
5.6.7 软件复位.....	40
(2) 系统控制寄存器(P_SYS_Ctrl, \$0030).....	42
(3) 低电压复位控制寄存器(P_LVR_Opt, \$0036).....	43
5.7 I/O 端口.....	43
5.7.1 简述.....	43
5.7.2 端口 A.....	44
(2) 端口 A 数据寄存器(P_IOA_Data, \$0000).....	45
(3) 端口 A 数据锁存器(P_IOA_Buf, \$0059).....	45
(4) 端口 A 方向寄存器(P_IOA_Dir, \$0004).....	46
(5) 端口 A 属性寄存器(P_IOA_Attrib, \$0008).....	46
5.7.3 端口 B.....	46
(2) 端口 B 数据寄存器(P_IOB_Data, \$0001).....	47
(3) 端口 B 数据锁存器(P_IOB_Buf, \$005A).....	47
(4) 端口 B 方向寄存器(P_IOB_Dir, \$0005).....	48
(5) 端口 B 属性寄存器(P_IOB_Attrib, \$0009).....	48
(6) 慢速输出控制寄存器(P_IO_Opt, \$0035).....	48
5.7.4 端口 C.....	49
(2) 端口 C 数据寄存器(P_IOC_Data, \$0002).....	49
(3) 端口 C 数据锁存器(P_IOC_Buf, \$005B).....	50
(4) 端口 C 方向寄存器(P_IOC_Dir, \$0006).....	50
(5) 端口 C 属性寄存器(P_IOC_Attrib, \$000A).....	50
5.7.5 Port D.....	51
(2) 端口 D 数据寄存器(P_IOD_Data, \$0003).....	51
(3) 端口 D 数据锁存器(P_IOD_Buf, \$005C).....	51
(4) 端口 D 方向寄存器(P_IOD_Dir, \$0007).....	52
(5) 端口 D 属性寄存器(P_IOD_Attrib, \$000B).....	52
5.7.6 Port E.....	52
(2) 端口 E 数据寄存器(P_IOE_Data, \$0040).....	53
(3) 端口 E 数据锁存器(P_IOE_Buf, \$005D).....	53
(4) 端口 E 方向寄存器(P_IOE_Dir, \$0042).....	53
(5) 端口 E 属性寄存器(P_IOE_Attrib, \$0044).....	54
5.8 定时/计数器.....	54
5.8.1 简述.....	54
5.8.2 定时/计数器 0.....	54
(2) 定时/计数器 0-1 控制寄存器 0 (P_TMR0_1_Ctrl0, \$0011).....	57

(3) 定时/计数器 0-1 控制寄存器 1 (P_TMR0_1_Ctrl1, \$0012)	58
(4) 定时/计数器 0 计数寄存器 (P_TMR0_Count, \$0013)	58
(5) 定时/计数器高字节寄存器(P_TMR0_CountHi, \$0014)	59
5.8.3 定时/计数器 1	60
(1) 定时/计数器 1 计数寄存器低字节(P_TMR1_Count, \$0015)	60
(2) 定时/计数器 1 计数寄存器高字节(P_TMR1_CountHi, \$0016)	61
(3) PWM1 占空比的低字节(P_TMR1_PWMduty, \$0017)	62
5.8.4 定时/计数器 2	63
(1) 定时/计数器 2-3 控制器 0 (P_TMR2_3_Ctrl0, \$0018)	63
(2) 定时/计数器 2-3 控制寄存器 1 (P_TMR2_3_Ctrl1, \$0019)	64
(3) 定时/计数器 2 计数寄存器(P_TMR2_Count, \$001A)	64
(4) 定时/计数器 2 计数寄存器高字节(P_TMR2_CountHi, \$001B)	65
5.8.5 定时/计数器 3	66
(1) 定时/计数器 3 计数寄存器低字节(P_TMR3_Count, \$001C)	66
(2) 定时/计数器 3 计数寄存器高字节(P_TMR3_CountHi, \$001D)	67
(3) PWM3 占空比的低字节(P_TMR3_PWMduty, \$001E)	68
5.9 CCP(CAPTURE/COMPARE/PWM)功能	68
5.9.2 8 位比较模式	68
5.9.3 16 位比较模式	69
5.9.4 8 位捕获模式	70
(1) 捕获控制寄存器(P_CAP_Ctrl, \$0058)	70
5.9.5 16 位捕获模式	72
5.9.6 8 位 PWM 模式	72
5.9.7 12 位 PWM 模式	73
5.10 模拟功能	75
5.10.1 A/D 转换	75
1. 简述	75
2. ADC 寄存器	75
(1) ADC 控制寄存器 0(P_AD_Ctrl0, \$0028)	75
(2) ADC 控制寄存器 1(P_AD_Ctrl1, \$0029)	76
(3) ADC 控制寄存器 2 (P_AD_Ctrl2, \$002A)	77
(4) A/D 转换结果的高 8 位(P_AD_DataHi, \$002B)	77
(5) A/D 转换结果的低 2 位(P_AD_DataLo, \$002C)	77
3. ADC 转换流程	78
5.11 通信模块	79
5.11.1 SPI (Serial Peripheral Interface)	79
1. 简述	79
2. SPI 运行模式	80
3. 从模式	81
4. SPI 寄存器	82
(1) SPI 控制寄存器 0 (P_SPI_Ctrl0, \$0038)	82
(2) SPI 控制寄存器 1 (P_SPI_Ctrl1, \$0039)	83
(3) SPI TX/RX 状态寄存器 (P_SPI_Status, \$003A)	83
(4) SPI 发送缓冲器 0 (P_SPI_TxData, \$003B)	84
(5) SPI 接收缓冲器 0 (P_SPI_RxData, \$003C)	84
5.11.2 UART (Universal Asynchronous Receiver/Transceiver)	86
1. 简述	86
2. UART 操作	87
3. UART 寄存器	90
(1) UART 控制寄存器(P_UART_Ctrl, \$0046)	90
(2) UART 波特率分频寄存器(P_UART_Baud, \$0047)	90

(3) UART 状态寄存器(P_UART_Status, \$0048).....	91
(4) UART 数据寄存器(P_UART_Data, \$0049).....	92
5.12 其它外围模块.....	92
5.12.1 看门狗.....	92
(2) 看门狗控制寄存器 (P_WDT_Ctrl, \$0032).....	93
(3) 看门狗清除寄存器 (P_WDT_Clr, \$0010).....	94
5.12.2 时基定时器.....	94
5.12.3 蜂鸣器.....	96
(2) 蜂鸣器控制寄存器(P_BUZ_Ctrl, \$002D).....	97
5.13 芯片配置寄存器.....	98
5.13.1 简述.....	98
5.13.2 芯片配置寄存器.....	98
(1) 芯片配置寄存器 (P_MO, \$7FE0).....	98
(2) 保密设置寄存器 (P_SECU, \$7FE2).....	98
(3) NMI 设置寄存器(P_NMI, \$7FE3).....	99
5.14 指令集.....	99
6 电气特性.....	105
6.1 极限参数 (VSS = 0).....	105
6.2 推荐运行参数.....	105
6.3 DC/AC 电气特性 (VDD = 5.0V, T _A = - 40°C~85°C).....	105
6.3.1 项目符号定义.....	105
6.3.2 管脚说明.....	106
6.4 模拟接口电气特性(VDD = 5.0V, T _A = - 40°C ~ 85°C).....	107
7 封装和管脚焊接位置.....	109
7.1 管脚分配和位置.....	109
7.2 分类.....	109
7.3 封装.....	109
7.3.1 PDIP32 (600mil).....	109
7.3.2 PDIP42 (600mil).....	110
7.3.3 SSOP48 (295mil).....	111
7.4 储存条件与时间.....	112
7.5 建议表面贴装(SMT)温度曲线.....	112

修订记录

版本	日期	作者	备注
V1.1	2007/05/07	SNDC	初始版本

1 总述

SPMC65P2708A 和 SPMC65P2716A 是 SPMC65X 系列的两款芯片，程序存储区均由一次性编程存储器(OTP)构成。SPMC65P2716A 在资源上比 SPMC65P2708A 有所扩展，它们的内核和外围几乎相同，只有一个特性有所不同。SPMC65P2708A 的 ROM 空间比 SPMC65P2716A 小。这两款芯片都具有下面列出的这些特性，它们主要应用于小家电、工业设备控制。下面介绍这两款芯片的主要特性。

2 特性

■ SPMC65 CPU

- 支持 182 条指令
- 支持 11 种寻址方式
- 系统时钟频率最高可达 8MHz
- 支持位操作指令（置 1、清零、取反、测试）

■ 存储空间

- 8K 字节的程序空间（OTP），并且可设置保密功能。（SPMC65P2708A）
- 16K 字节的程序空间（OTP），并且可设置保密功能。（SPMC65P2716A）
- 512 字节的 RAM（含堆栈空间）

■ I/O 端口

- 27 个具有复用功能的双向 I/O 端口（2708A）
- 37 个具有复用功能的双向 I/O 端口（2716A）
- 所有 I/O 端口的输入均由施密特触发器构成
- 可编程设置为带上拉/下拉电阻输入或者悬浮输入
- I/O 具备 LED 驱动能力
- 其中 2 个 I/O 端口能提供 20mA 的吸入电流

■ 中断管理

- 外部中断选项：非屏蔽中断 NMI 或可屏蔽中断 IRQ

- 支持 6 个外部中断，其中的一个可设置为非屏蔽中断 NMI

- 13 个内部中断

■ 复位管理

- 上电复位(POR)
- 低电压复位(LVR)
- 看门狗复位(WDR)
- 外部复位(ERST)
- 非法地址复位(IAR)

■ 时钟管理

- 3 种时钟源：晶体/陶瓷振荡器、RC 振荡器、外部时钟输入
- 在 RC 振荡模式下，能够输出时钟信号

■ 电源管理

- 2 种省电模式：STOP 模式、HALT 模式

■ 2 个外围模拟电路

- 8 通道 10 位精度的 A/D 转换器（100KHz）
- 低电压复位（2.5V/4V）

■ 4 通道 16 位定时/计数器

- 定时、计数功能
- 捕获功能（8 位脉宽/周期测量，或 16 位脉宽测量）
- 16 位比较输出功能
- 8 位 PWM 输出功能(Timer0, Timer2)
- 12 位 PWM 输出功能(Timer1, Timer3)

■ 时基定时器

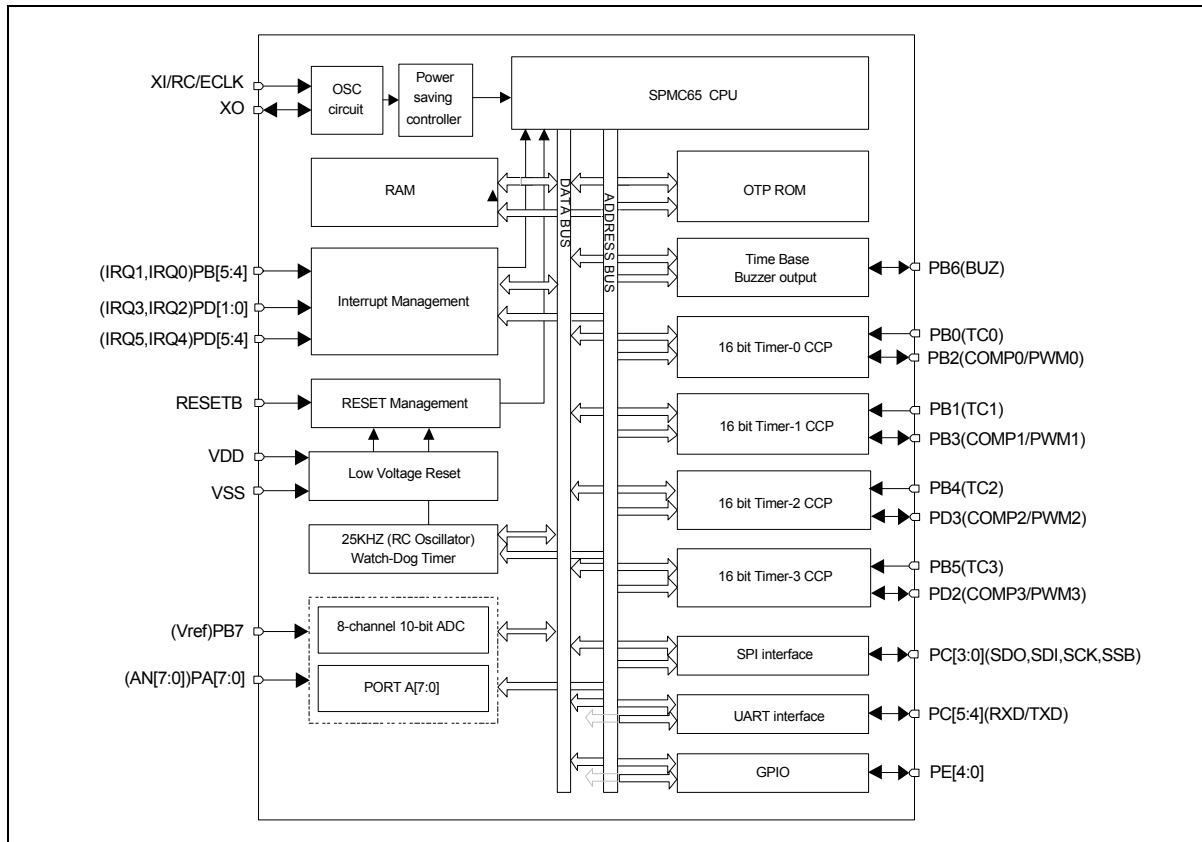
- 频率选择：1KHz ~ 62.5KHz @8MHz
- 15 级分频选择

■ 蜂鸣器输出

- 频率选择: 1KHz ~ 2MHz @8MHz
- 12 级分频选择
- 可编程看门狗定时器
 - 中断频率选择: 1.5Hz ~ 195Hz@8MHz
- 串行总线接口
 - SPI 总线频率: 最高为 2MHz @8MHz
- UART 接口
 - 波特率: 最高为 38400 bps

Device	EPROM	RAM	Max I/O	10 位 A/D(ch)	CCP 功能			SPI	UART	封装类型
					比较	捕获	PWM			
SPMC65P2708A	8K x 8	512 x 8	27	8	4	4	4	Y	Y	PDIP32
	8K x 8	512 x 8	37	8	4	4	4	Y	Y	PDIP42, SSOP48
SPMC65P2716A	16K x 8	512 x 8	27	8	4	4	4	Y	Y	PDIP32
	16K x 8	512 x 8	37	8	4	4	4	Y	Y	PDIP42, SSOP48

3 芯片结构概览



4 信号描述

4.1 管脚说明

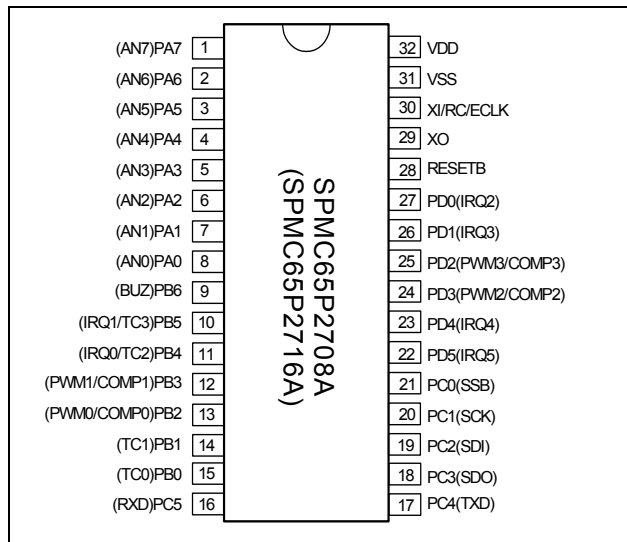
类型：I = 输入；O = 输出；S = 电源

管脚名称	管脚号			类型	主要功能	可选功能
	32Pin	42Pin	48Pin			
AVDD	32	42	48	S	ADC 和 LVR 模拟电源	
VDD	32	42	47	S	电源	
AVSS	31	41	46	S	ADC 和 LVR 模拟接地	
VSS	31	41	45	S	接地	
XI/RC/ECLK	30	40	44	I	晶体/陶瓷振荡器输入端、RC 振荡器输入端或外部时钟输入端：在 RC 振荡模式下，外接一个上拉电阻，可以和芯片内部的 OSC 电路产生内部时钟，作为 CPU 的时钟信号；或者在晶体/陶瓷振荡器模式下，直接和外部晶体/陶瓷振荡器的一个管脚连接，由晶体和芯片内部电路共同产生时钟信号	
XO	29	39	43	O	振荡器频率输出端：在晶体/陶瓷振荡器振荡器模式下，直接和外部晶体/陶瓷振荡器的另一个管脚连接，由晶体和芯片内部电路共同产生时钟信号；也可以做为 RC 振荡器的频率输出管脚	
PA7/AN7	1	1	1	I/O	Port A7	ADC 模拟电压输入
PA6/AN6	2	2	2	I/O	Port A6	ADC 模拟电压输入
PA5/AN5	3	3	3	I/O	Port A5	ADC 模拟电压输入
PA4/AN4	4	4	4	I/O	Port A4	ADC 模拟电压输入
PA3/AN3	5	5	5	I/O	Port A3	ADC 模拟电压输入
PA2/AN2	6	6	6	I/O	Port A2	ADC 模拟电压输入
PA1/AN1	7	7	7	I/O	Port A1	ADC 模拟电压输入
PA0/AN0	8	8	8	I/O	Port A0	ADC 模拟电压输入
PB7/Vref	--	9	9	I/O	Port B7	ADC 参考电压输入
PB6/BUZ	9	10	10	I/O	Port B6	蜂鸣器输出
PB5/IRQ1/TC3	10	11	11	I/O	Port B5	外部中断 1 输入 / Timer3 的捕获事件输入 / Timer3 的外部事件输入
PB4/IRQ0/TC2	11	12	12	I/O	Port B4	外部中断 0 输入 / Timer2 的捕获事件输入 / Timer2 的外部事件输入
PB3/COMP1 /PWM1	12	13	13	I/O	Port B3	Timer1 比较输出 / PWM 输出
PB2/COMP0 /PWM0	13	14	14	I/O	Port B2	Timer0 比较输出 / PWM 输出
PB1/TC1	14	15	15	I/O	Port B1	Timer1 的捕获事件输入 / Timer1 的外部事件输入
PB0/TC0	15	16	18	I/O	Port B0	Timer0 的捕获事件输入 / Timer0 的外部事件输入
PC7	--	17	19	I/O	Port C7	

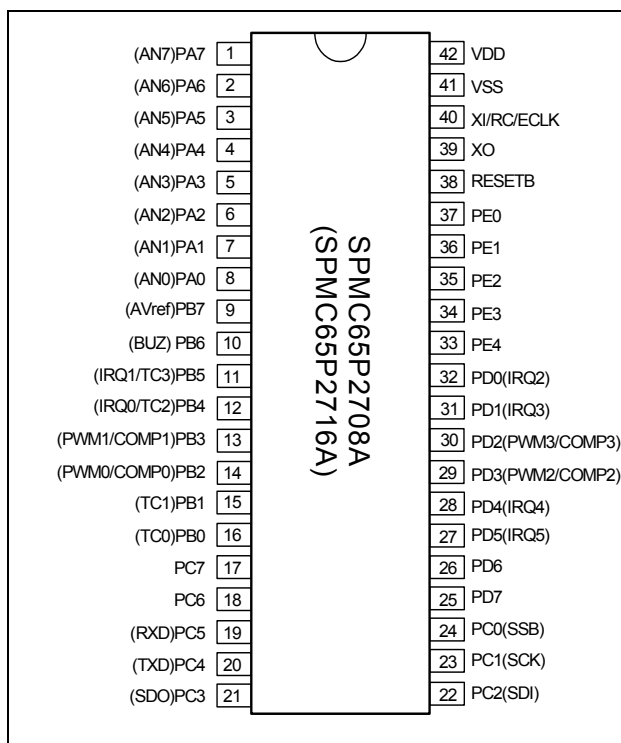
管脚名称	管脚号			类型	主要功能	可选功能
	32Pin	42Pin	48Pin			
PC6	--	18	20	I/O	Port C6	
PC5/RXD	16	19	21	I/O	Port C5	UAR 信号接收
PC4/TXD	17	20	22	I/O	Port C4	UART 信号发送
PC3/SDO	18	21	23	I/O	Port C3	SPI 数据输出
PC2/SDI	19	22	24	I/O	Port C2	SPI 数据输入
PC1/SCK	20	23	27	I/O	Port C1	SPI 时钟输出 / 输入
PC0/SSB	21	24	28	I/O	Port C0	SPI 片选
PD7	--	25	29	I/O	Port D7	
PD6	--	26	30	I/O	Port D6	
PD5/IRQ5	22	27	31	I/O	Port D5	外部中断 5 输入
PD4/IRQ4	23	28	32	I/O	Port D4	外部中断 4 输入
PD3/COMP2 /PWM2	24	29	33	I/O	Port D3	Timer2 比较输出/Timer2 PWM 输出
PD2/COMP3 /PWM3	25	30	34	I/O	Port D2	Timer3 比较输出/Timer3 PWM 输出
PD1/IRQ3	26	31	35	I/O	Port D1	外部中断 3 输入
PD0/IRQ2	27	32	36	I/O	Port D0	外部中断 2 输入
PE4	--	33	37	I/O	Port E4	
PE3	--	34	38	I/O	Port E3	
PE2	--	35	39	I/O	Port E2	
PE1	--	36	40	I/O	Port E1	
PE0	--	37	41	I/O	Port E0	
RESETB	28	38	42	I	Reset pin	外部复位信号

4.2 管脚分配 (Top View)

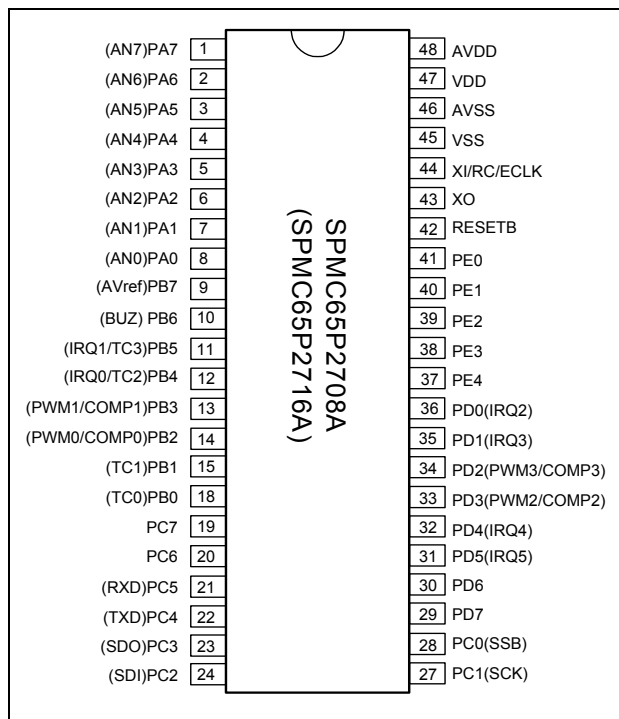
4.2.1 PDIP32 封装



4.2.2 PDIP42 封装



4.2.3 SSOP48 封装



5 功能介绍

5.1 CPU

5.1.1 简介

SPMC65P2716A/2708A 采用高性能微处理 SPMC65 作为内核，全静态 CMOS 工艺设计，它有 6 个内部寄存器：累加器 (A)、程序指针 (PC)、X 寄存器、Y 寄存器、堆栈指针 (SP) 和状态寄存器 (P)。系统最高运行时钟 (FSYS) 可以达到 16MHz。

5.1.2 CPU 寄存器

SPMC65 CPU 有 6 个寄存器：程序指针 (PC)、累加器(A)、X 寄存器、Y 寄存器，堆栈指针(SP)、状态寄存器 (P)。其中程序指针 (PC) 是一个 16 位寄存器，其它都是 8 位寄存器，见下图。

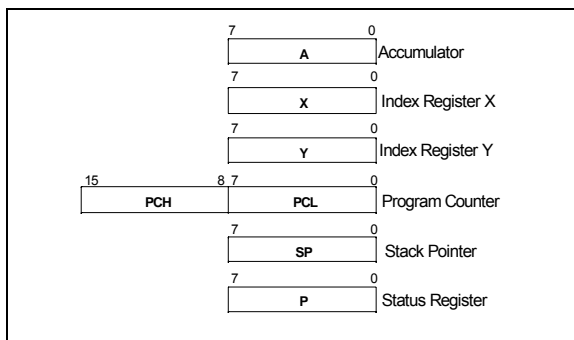


图 5-1 System registers

■ X、Y 寄存器

在寻址模式下，X、Y 寄存器用作地址索引指针，可以很方便的进行数据存取。同时，X、Y 寄存器还可以做为缓存寄存器，进行加 1、减 1、比较和数据传送的操作。

■ 累加器 A

累加器是一个 8 位通用寄存器，可用于数据传送、数据暂存和条件判断等操作。

■ 堆栈指针 (SP)

它是一个 8 位寄存器，当中断发生或者调用子程序时，用于指示堆栈中被操作的字节的地址，也可以表明当前堆栈的使用情况。

当调用子程序或中断发生时，堆栈指针会自动更新。但是如果堆栈指针的值超出了其允许的堆栈空间，系统就

会发生非法地址复位。

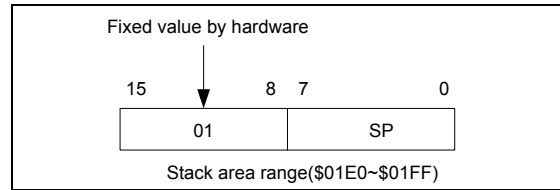


图 5-2 堆栈指针寄存器

【例】5.1.1 堆栈指针初始化

```
ldx  #C_STACK_BOTTOM          ;堆栈指针初始值为$FF
txs                                ;将累加器 A 值送入 X 寄存器中
```

■ 程序计数器(PC)

它是一个 16 位寄存器，存放 CPU 即将执行的下一条指令的 16 位地址。该 16 位地址的高 8 位和低 8 位分别存放在寄存器 PCH 和 PCL 中。复位后，PC 中存储的是程序运行的起始地址\$FFFCH。

■ 状态寄存器(P)

它是一个 8 位寄存器，用于指示 6 个独立的状态标志位，包括一个中断屏蔽位。这些状态标志位可以反映程序执行完上一条指令后的状态信息。该寄存器也可以由指令 PHP 和 PLP 来进行保存和取出。同时，也有相应的特殊指令来对这个寄存器进行位操作。各标志位的详细说明如下。

注：并非所有的指令执行都会影响状态寄存器的值。在 SPMC65 CPU 指令手册中将对每个指令作详细描述。

■ 负标志位 (N)

该位指示一个数据或者运算结果的 bit7 的状态。用户可以通过该位进行条件跳转或者位操作。

■ 溢出标志位(V)

该位指示在算术运算过程是否发生溢出。如果加法运算结果大于 127 或者减法运算结果小于-128，该位则置 1。

■ 十进制模式标志位(D)

SPMC65P2708A/2716A 具备两种算术运算模式：二进

制模式和十进制模式。该标志位表明了当前的运算模式，用户可以通过相应指令切换这两种运算模式。

■ 中断屏蔽位(I)

该位用于使能/禁止除“非屏蔽中断源 (NMI)”以外的所有中断源。将其置 1，CPU 将忽略中断请求；置 0，CPU 将接受中断请求。

■ 零标志位 (Z)

数据和算术运算结果标志位。若数据或算术运算结果为 0，该位被置 1；否则，置 0。

■ 进位标志 (C)

当加法操作中产生进位或减法操作中没有产生借位时，该位被置 1。此外，移位或循环指令也会改变该位的值。

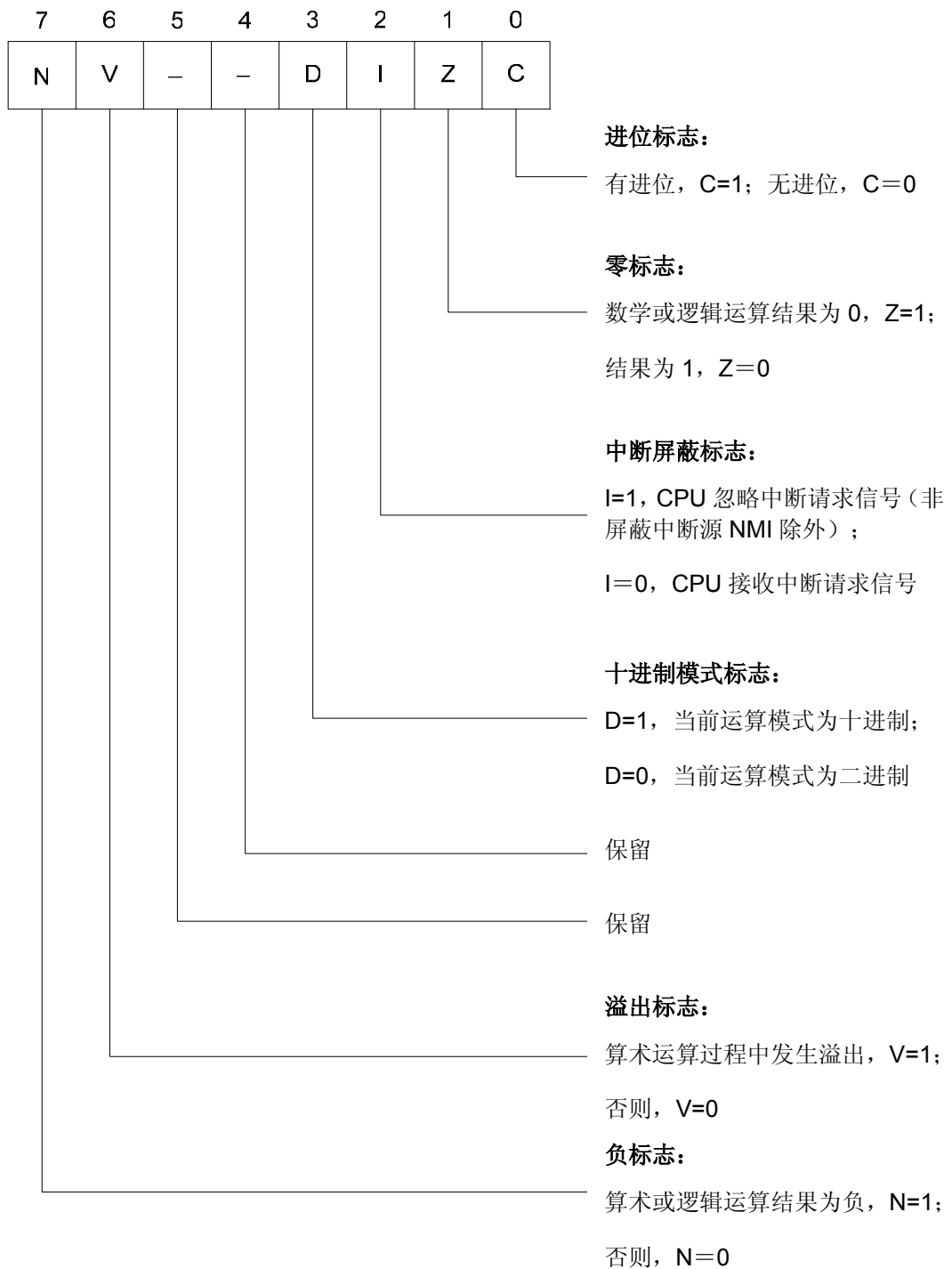


图 5-3 状态寄存器

5.2 存储器

5.2.1 简介

SPMC65P2708A/2716A 芯片有独立的程序存储区和数据存储区。程序存储区为只读，若向其写数据将导致非法地址复位，它的空间大小为 8K/16K 字节。数据存储区可读可写（RAM），空间大小为 512 字节（包括堆栈区）。

5.2.2 存储空间分配

SPMC65P2708A 和 SPMC65P2716A 的存储空间分为几个部分，分别如图 5-5和图 5-6所示。二者的差别仅在于 OTP ROM 的大小。在此仅以 SPMC65P2708A 为例对各个部分进行详细说明，SPMC65P2716A 与之相同。

存储空间的前 96 个字节（\$0000~\$005F）分配给硬件控制寄存器（含功能控制寄存器和 I/O 控制寄存器）。硬件控制寄存器允许用户通过零页指令来设置，同时有助于减少程序大小。

\$0060~\$025F 为数据存储区（RAM）。其中 \$0100~\$01FF（256 个字节）也定义为堆栈区，堆栈指针向下生长（从\$01FF 到\$0100）。一旦堆栈溢出，CPU 复位。

\$7FE0 ~ \$7FE3 这 4 个字节为芯片配置寄存器，用户可以设置一些特殊功能。详情参见 5.13。

SPMC65P2708A 预留了\$7FF0 ~ \$7FFF 作为用户信息块，可以由用户写入芯片序列号或版本控制信息等。

SPMC65P2708A 支持 8K 字节的 OTP ROM（\$E000 ~ \$FFFF）来保存程序，见图 5-5。用户可以设置“保密位”对程序进行保护。当“保密位”设置为 0 时，OTP ROM 中的程序被保护，不可读（但其中的芯片配置寄存器和用户信息仍是可读的）；相反，置 1 时，可读。

中断（NMI、IRQ）及复位（RESET）的向量地址位于 \$FFFA~\$FFFF。这些中断向量的地址应在用户程序中做出正确的定义。

若非法访问未定义的存储空间，CPU 将产生 IAR 复位，重新启动程序。所以芯片中定义的限制访问块（保留区）不能进行读写操作。

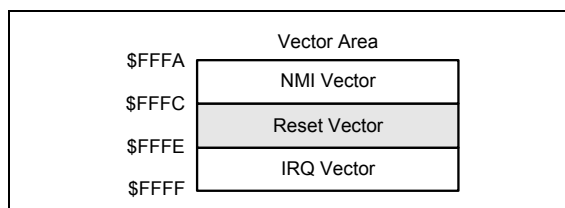


图 5-4 中断和复位向量表

【例】 5.2.1中断和复位向量表的定义

```
VECTOR: .SECTION
        DW   V_NMI
        DW   V_Reset
        DW   V_IRQ
```

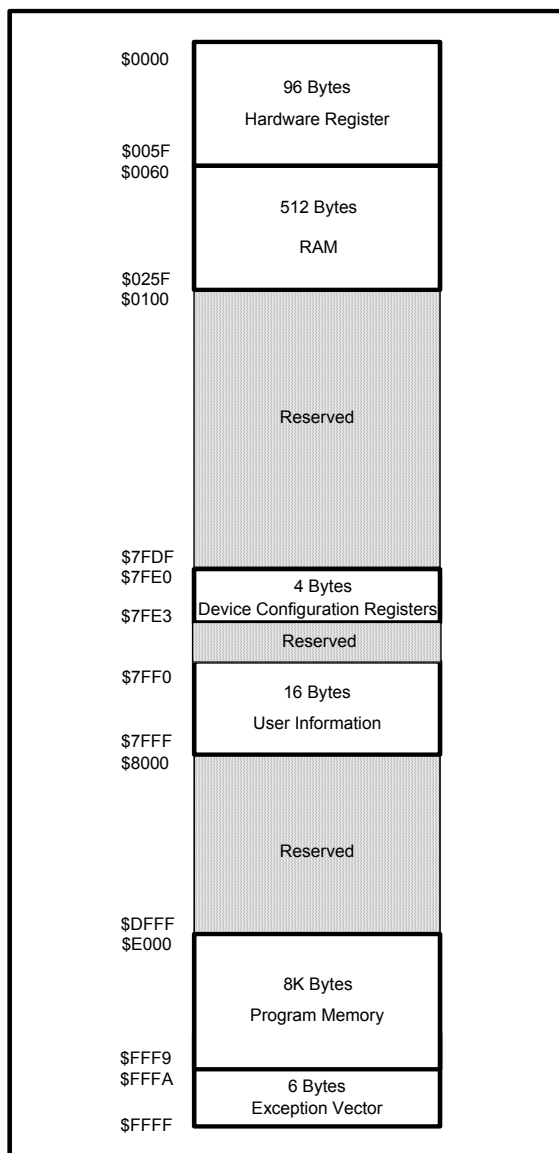


图 5-5 SPMC65P2708A 存储空间分配图

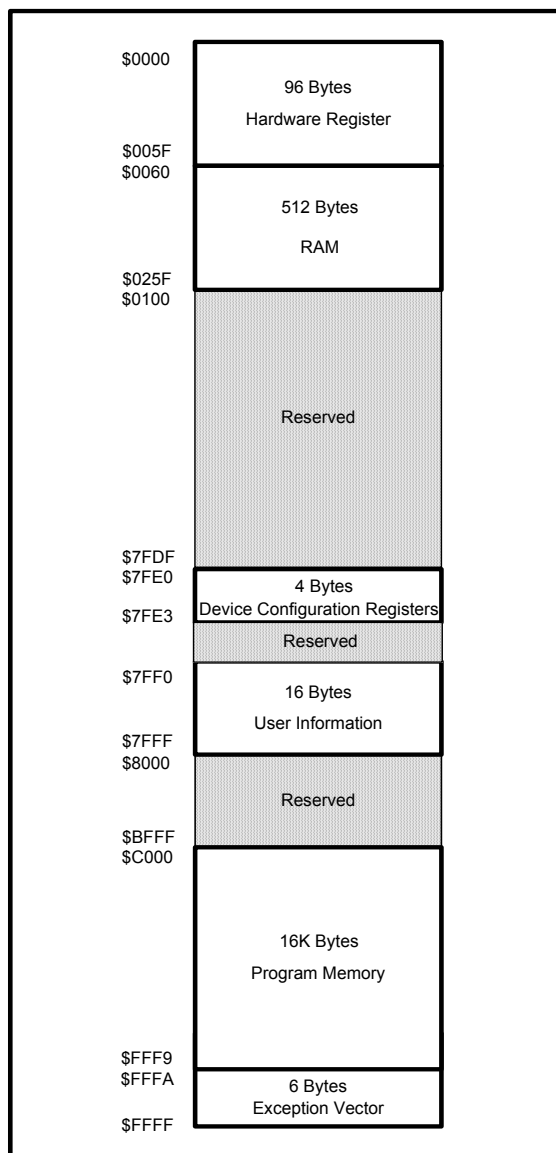


图 5-6 SPMC65P2716A 存储空间分配图

5.2.3 硬件控制寄存器

SPMC65P2708A/2716A 有多达 60 个控制寄存器。通过设置这些寄存器，可以控制芯片及其外围的工作。

某些控制寄存器包含控制位和状态位，例如，可用于对定时/计数器、ADC、SPI、UART 等外围模块进行设置和状态标识。注意控制寄存器无法操作那些标识为“保留”内存空间。

控制寄存器的某些位为只读属性，向这些位写入数据无效。注：保留区的地址不能被操作。下表为控制寄存器的总表，详细介绍请见各章节。

■ 控制寄存器列表

(1) \$0000~000F: I/O 端口及中断控制寄存器

地址	功能	复位值	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
\$0000	端口 A 数据寄存器 (P_IOA_Data)	00h	R	端口 A 的管脚状态							
			W	端口 A 的输出数据(\$0059)							
\$0001	端口 B 数据寄存器 (P_IOB_Data)	00h	R	端口 B 的管脚状态							
			W	端口 B 的输出数据(\$005A)							
\$0002	端口 C 数据寄存器 (P_IOC_Data)	00h	R	端口 C 的管脚状态							
			W	端口 C 的输出数据(\$005B)							
\$0003	端口 D 数据寄存器 (P_IOD_Data)	00h	R	端口 D 的管脚状态							
			W	端口 D 的输出数据(\$005C)							
\$0004	端口 A 方向寄存器 (P_IOA_Dir)	00h	R	0 = 输入 1 = 输出							
			W	0 = 输入 1 = 输出							
\$0005	端口 B 方向寄存器 (P_IOB_Dir)	00h	R	0 = 输入 1 = 输出							
			W	0 = 输入 1 = 输出							
\$0006	端口 C 方向寄存器 (P_IOC_Dir)	00h	R	0 = 输入 1 = 输出							
			W	0 = 输入 1 = 输出							
\$0007	端口 D 方向寄存器 (P_IOD_Dir)	00h	R	0 = 输入 1 = 输出							
			W	0 = 输入 1 = 输出							
\$0008	端口 A 属性寄存器 (P_IOA_Attrib)	00h	R	端口 A 属性							
			W	端口 A 属性							
\$0009	端口 B 属性寄存器 (P_IOB_Attrib)	00h	R	端口 B 属性							
			W	端口 B 属性							
\$000A	端口 C 属性寄存器 (P_IOC_Attrib)	00h	R	端口 C 属性							
			W	端口 C 属性							
\$000B	端口 D 属性寄存器 (P_IOD_Attrib)	00h	R	端口 D 属性							
			W	端口 D 属性							
\$000C	中断标志 0 (P_INT_Flag0)	00h	R	ADIF	WDIF	IRQ5IF	IRQ4IF	IRQ3IF	IRQ2IF	IRQ1IF/ CAP3IF	IRQ0IF/ CAP2IF
			W	写入'1'清除中断标志							
\$000D	中断控制 0 (P_INT_Ctrl0)	00h	R	ADIE	WDIE	IRQ5IE	IRQ4IE	IRQ3IE	IRQ2IE	IRQ1IE / CAP3E	IRQ0IE / CAP2E

			W	ADIE	WDIE	IRQ5IE	IRQ4IE	IRQ3IE	IRQ2IE	IRQ1IE / CAP3IE	IRQ0IE / CAP2IE
\$000E	中断标志 1 (P_INT_Flag1)	00h	R	CAP1IF	CAP0IF	0	0	T3OIF	T2OIF	T1OIF	T0OIF
			W	写入'1'清除中断标志							
\$000F	中断控制 1 (P_INT_Ctrl1)	00h	R	CAP1E	CAP0E	0	0	T3OIE	T2OIE	T1OIE	T0OIE
			W	CAP1E	CAP0E	-	-	T3OIE	T2OIE	T1OIE	T0OIE

注 1：IRQ0、IRQ1 的状态/控制位分别与 CAP2、CAP3 的状态/控制位共用。

注 2：端口 A、端口 B、端口 C、端口 D 复位后的属性取决于 IOINIT(\$7FE2.0)的设置。

(2) \$0010~\$001E: Timer/PWM 设置 & 数据寄存器

地址	功能	复位值	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
\$0010	看门狗清除寄存器 (P_WDT_Clr)	00h	R	0	0	0	0	0	0	0	0
			W	写 #55 清除看门狗(C_WDT_Clr = 55)							
\$0011	定时/计数器 0-1 控制寄存器 0 (P_TMR0_1_Ctrl0)	00h	R	0	定时/计数器 1 功能选择		0	0	定时/计数器 0 功能选择		
			W	-	定时/计数器 1 功能选择		-	-	定时/计数器 0 功能选择		
\$0012	定时/计数器 0-1 控制寄存器 1 (P_TMR0_1_Ctrl1)	00h	R	0	定时/计数器 1 预分频选择		0	定时/计数器 0 预分频选择			
			W	-	定时/计数器 1 预分频选择		-	定时/计数器 0 预分频选择			
\$0013	定时/计数器 0 计数值 (P_TMR0_Count)	00h	R	8 位定时/计数器 0 的计数值							
	定时/计数器 0 重载寄存器 (P_TMR0_Preload)		W	8 位定时/计数器 0 的重载计数值							
	比较器 0 计数值 (P_TMR0_Comp)	00h	R	8 位比较器 0 的计数值							
			W	8 位比较器 0 的比较值							
	捕获器 0 捕获值 (P_TMR0_Cap)	00h	R	8 位捕获器 0 的捕获脉宽值							
			W	8 位捕获器 0 的重载值							
PWM0 周期值 (P_TMR0_PWMPeriod)	00h	R	8 位 PWM0 的周期								
		W	8 位 PWM0 的周期								
\$0014	定时/计数器 0 计数值的高字节 (P_TMR0_CountHi)	00h	R	16 位定时/计数器 0 计数值的高字节							

地址	功能	复位值	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	定时/计数器 0 重载寄存器 (P_TMR0_PreloadHi)		W	16 位定时/计数器 0 重载计数值的高字节							
	比较器 0 计数值的高字节 (P_TMR0_CompHi)	00h	R	16 位定时/计数器 0 计数值的高字节							
			W	16 位比较器 0 比较值的高字节							
	捕获器 0 捕获值的高字节 (P_TMR0_CapHi)	00h	R	16 位捕获器 0 捕获脉宽值的高字节							
			W	16 位捕获器 0 重载值的高字节							
	捕获器 0 周期值 (P_TMR0_CapCycle8)	00h	R	8 位捕获器 0 捕获的周期值							
W			16 位定时/计数器 0 重载值的高字节								
PWM0 占空比/周期值 (P_TMR0_PWMduty)	00h	R	8 位 PWM0 占空比值								
		W	8 位 PWM0 占空比值								
\$0015	定时/计数器 1 计数值 (P_TMR1_Count)	00h	R	8 位或 16 位 (低字节) 定时/计数器 1 的计数值							
	定时/计数器 1 重载寄存器 (P_TMR1_Preload)		W	8 位或 16 位 (低字节) 定时/计数器 1 的重载计数值							
	比较器 1 计数值的低字节 (P_TMR1_Comp)	00h	R	8 位或 16 位 (低字节) 比较器 1 的计数值							
			W	8 位或 16 位 (低字节) 比较器 1 的比较值							
	捕获器 1 捕获值的低字节 (P_TMR1_Cap)	00h	R	8 位或 16 位 (低字节) 捕获器 1 的捕获脉宽值							
			W	8 位或 16 位 (低字节) 捕获器 1 的重载值							
PWM1 低字节周期值 (P_TMR1_PWMPeriod)	00h	R	12 位 PWM 1 低字节的周期值								
		W	12 位 PWM 1 低字节的周期值								
\$0016	定时/计数器 1 计数值的高字节 (P_TMR1_CountHi)	00h	R	16 位定时/计数器 1 计数值的高字节							
	定时/计数器 1 重载寄存器 (P_TMR1_PreloadHi)		W	16 位定时/计数器 1 重载计数值的高字节							
	比较器 1 计数值的高字节 (P_TMR1_CompHi)	00h	R	16 位比较器 1 计数值的高字节							
			W	16 位比较器 1 比较值的高字节							
	捕获器 1 捕获值的高字节 (P_TMR1_CapHi)	00h	R	16 位捕获器 1 捕获脉宽值的高字节							
			W	16 位捕获器 1 重载值的高字节							
捕获器 1 周期值 (P_TMR1_CapCycle8)	00h	R	8 位捕获器 1 捕获的周期值								
		W	16 位定时/计数器 1 重载计数值的高字节								
PWM1 占空比周期 (P_TMR1_DutyPeriod)	00h	R	12 位定时/计数器 PWM1 占空比值(高字节)				12 位定时/计数器 1 PWM1 周期值(高字节)				
		W	12 位定时/计数器 PWM1 占空比值(高字节)				12 位定时/计数器 1 PWM1 周期值(高字节)				

地址	功能	复位值	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
\$0017	PWM1 低字节占空比 (P_TMR1_PWMDuty)	00h	R	12 位定时/计数器 PWM1 的占空比值(低字节)							
			W	12 位定时/计数器 PWM1 的占空比值(低字节)							
\$0018	定时/计数器 2-3 控制器 0 (P_TMR2_3_Ctrl0)	00h	R	0	定时/计数器 3 功能选择		0	定时/计数器 2 功能选择			
			W	-	定时/计数器 3 功能选择		-	定时/计数器 2 功能选择			
\$0019	定时/计数器 2-3 控制器 1 (P_TMR2_3_Ctrl1)	00h	R	0	定时/计数器 3 预分频选择		0	定时/计数器 2 预分频选择			
			W	-	定时/计数器 3 预分频选择		-	定时/计数器 2 预分频选择			
\$001A	定时/计数器 2 计数值 (P_TMR2_Count)	00h	R	8 位定时/计数器 2 的计数值							
	定时/计数器 2 重载寄存器 (P_TMR2_Preload)		W	8 位定时/计数器 2 的重载计数值							
	比较器 2 计数值的低字节 (P_TMR2_Comp)	00h	R	8 位比较器 2 的计数值							
			W	8 位比较器 2 的比较值							
	捕获器 2 捕获值 (P_TMR2_Cap)	00h	R	8 位捕获器 2 的捕获脉宽值							
			W	8 位捕获器 2 的重载值							
PWM2 周期值 (P_TMR2_PWMPeriod)	00h	R	8 位 PWM 2 的周期值								
		W	8 位 PWM2 的周期值								
\$001B	定时/计数器 2 计数值的高字节 (P_TMR2_CountHi)	00h	R	16 位定时/计数器 2 计数值的高字节							
	定时/计数器 2 重载寄存器 (P_TMR2_PreloadHi)		W	16 位定时/计数器 2 重载计数值的高字节							
	比较器 2 计数值高字节 (P_TMR2_CompHi)	00h	R	16 位定时/计数器 2 计数值的高字节							
			W	16 位比较器 2 比较值的高字节							
	捕获器 2 捕获值的高字节 (P_TMR2_CapHi)	00h	R	16 位捕获器 2 捕获脉宽值的高字节							
			W	16 位捕获器 2 重载值的高字节							
	捕获器 2 周期值 (P_TMR2_CapCycle8)	00h	R	8 位捕获器 2 捕获的周期值							
			W	16 位定时/计数器 2 重载值的高字节							
PWM2 占空比值 (P_TMR2_PWMDuty)	00h	R	8 位 PWM2 占空比值								
		W	8 位 PWM2 占空比值								
\$001C	定时/计数器 3 计数值的低字节 (P_TMR3_Count)	00h	R	8 位或 16 位 (低字节) 定时/计数器 3 的计数值							

地址	功能	复位值	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	定时/计数器 3 重载寄存器 (P_TMR3_Preload)		W	8 位或 16 位（低字节）定时/计数器 3 的重载计数值							
	比较器 3 计数值的低字节 (P_TMR3_Comp)	00h	R	8 位或 16 位（低字节）比较器 3 的计数值							
			W	8 位或 16 位（低字节）比较器 3 的比较值							
	捕获器 3 捕获值的低字节 (P_TMR3_Cap)	00h	R	8 位或 16 位（低字节）捕获器 3 的捕获脉宽值							
			W	8 位或 16 位（低字节）捕获器 3 的重载值							
	PWM3 周期值的低字节 (P_TMR3_PWMPeriod)	00h	R	12 位 PWM3 周期值的低字节							
W			12 位 PWM3 周期值的低字节								
\$001D	定时/计数器 3 计数值的高字节 (P_TMR3_CountHi)	00h	R	16 位定时/计数器 3 计数值的高字节							
	定时/计数器 3 重载寄存器 (P_TMR3_PreloadHi)		W	16 位定时/计数器 3 重载计数值的高字节							
	比较器 3 计数值的高字节 (P_TMR3_CompHi)	00h	R	16 位比较器 3 计数值的高字节							
			W	16 位比较器 3 比较值的高字节							
	捕获器 3 捕获值的高字节 (P_TMR3_CapHi)	00h	R	16 位捕获器 3 捕获脉宽值的高字节							
			W	16 位捕获器 3 重载值的高字节							
	捕获器 3 周期值 (P_TMR3_CapCycle8)	00h	R	8 位捕获器 3 捕获的周期值							
W			16 位捕获器 3 的重载值的高字节								
PWM3 占空比/周期值 (P_TMR3_DutyPeriod)	00h	R	12 位 PWM3 占空比高字节值				12 位 PWM3 周期高字节值				
		W	12 位 PWM3 占空比高字节值				12 位 PWM3 周期高字节值				
\$001E	PWM3 低字节占空比 (P_TMR3_PWM Duty)	00h	R	12 位 PWM3 低字节的占空比值							
			W	12 位 PWM3 低字节的占空比值							

(3) \$0026~002D: 中断和 ADC 控制寄存器

地址	功能	复位值	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
\$0026	中断标志 2 (P_INT_Flag2)	00h	R	0	0	ITVALIF	0	UARTIF	SPIIF	0	0
			W	写“1”可清除中断标志，但 SPIIF 标志除外							
\$0027	中断控制 2 (P_INT_Ctrl2)	00h	R	0	0	ITVALIE	0	0	0	0	0
			W	-	-	ITVALIE	-	-	-	-	-

\$0028	A/D 控制 0 (P_AD_Ctrl0)	06h	R	ADEN	ADVRT	0	0	ADCS2	ADCS1	ADCS0	ADRDY
			W	ADEN	ADVRT	-	-	ADCS2	ADCS1	ADCS0	STARTB
\$0029	A/D 输入通道控制 1 (P_AD_Ctrl1)	00h	R	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
			W	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
\$002A	A/D 控制 2 (P_AD_Ctrl2)	00h	R	ADCE	0	ADS2	ADS1	ADS0	0	0	0
			W	ADCE	-	ADS2	ADS1	ADS0	-	-	-
\$002B	A/D 转换值的高字节 (P_AD_DataHi)	00h	R	ADR9	ADR8	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2
			W	-							
\$002C	A/D 转换值的低字节 (P_AD_DataLo)	00h	R	ADR1	ADR0	0	0	0	0	0	0
			W	-							
\$002D	蜂鸣器控制 (P_BUZ_Ctrl)	00h	R	INTIMS3	INTIMS2	INTIMS1	INTIMS0	BZFS3	BZFS2	BZFS1	BZFS0
			W	INTIMS3	INTIMS2	INTIMS1	INTIMS0	BZFS3	BZFS2	BZFS1	BZFS0

(4) \$0030~\$0036: 特殊功能控制寄存器 (需要连续写两次才能成功)

地址	功能	复位值	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
\$0030	系统控制 (P_SYS_Ctrl)	C0h	R	POR	ERST	LVR	-	WDR	IAR	0	0
			W	写“1”清除相应的复位标志							-
\$0031	省电模式控制寄存器 (P_Mode_Ctrl)	00h	R	0	0	0	0	0	0	0	0
			W	写#\$5A 进入 STOP 模式 写#\$A5 进入 HALT 模式 写#\$66 复位除 CPU 以外的所有内部模块							
\$0032	看门狗控制寄存器 (P_WDT_Ctrl)	F0h	R	SCKEN	WDS2	WDS1	WDS0	0	0	0	0
			W	SCKEN	WDS2	WDS1	WDS0	-	-	-	-
\$0033	IRQ 设置寄存器 0 (P_IRQ_Opt0)	00h	R	0	0	0	0	IRQ5ES	IRQM5	IRQ4ES	IRQ4M
			W	-	-	-	-	IRQ5ES	IRQM5	IRQ4ES	IRQ4M
\$0034	IRQ 和捕获设置寄存器 1 (P_IRQ_Opt1)	00h	R	IRQ3ES	IRQM3	IRQ2ES	IRQM2	IRQ1ES / CAP3ES	IRQM1	IRQ0ES / CAP2ES	IRQM0
			W	IRQ3ES	IRQM3	IRQ2ES	IRQM2	IRQ1ES / CAP3ES	IRQM1	IRQ0ES / CAP2ES	IRQM0
\$0035	慢速输出功能	00h	R	0	0	0	0	0	0	0	SLOWE

地址	功能	复位值	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	(P_IO_Opt)		W	-	-	-	-	-	-	-	SLOWE
\$0036	低电压复位选项 (P_LVR_Opt)	00h	R	0	0	0	0	0	0	0	LVRV40
			W	-	-	-	-	-	-	-	LVRV40

注：外部复位不能清除 LVRV40 位。

(5) \$0038~\$003C: SPI 控制寄存器

地址	功能	复位值	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
\$0038	SPI 控制 0 (P_SPI_Ctrl0)	00h	R	SPIEN	MOD	SCKPHA	SCKPOL	SMS	SCKSEL2	SCKSEL1	SCKSEL0
			W	SPIEN	MOD	SCKPHA	SCKPOL	SMS	SCKSEL2	SCKSEL1	SCKSEL0
\$0039	SPI 控制 1 (P_SPI_Ctrl1)	02h	R	SMSSEN	SWRST	0	0	0	0	SPISPLK1	SPISPLK0
			W	SMSSEN	SWRST	-	-	-	-	SPISPLK1	SPISPLK0
\$003A	SPI 发送/接收状态 (P_SPI_Status)	00h	R	SPIIF	SPIIEN	TXBF	0	0	0	0	BUFFULL
			W	'1':清除	SPIIEN	-	-	-	-	-	-
\$003B	SPI 发送数据 (P_SPI_TxData)	00h	R	0	0	0	0	0	0	0	0
			W	SPI 输出的数据							
\$003C	SPI 接收数据 (P_SPI_RxData)	00h	R	SPI 接收的数据							
			W	-	-	-	-	-	-	-	-

(6) \$0040~\$0044: I/O 端口 E 控制寄存器

地址	功能	复位值	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
\$0040	端口 E 电平状态 (P_IOE_Data)	00h	R	0				端口 E 管脚状态			
			W	-				端口 E 输出数据(写入\$005D)			
\$0042	端口 E 方向 (P_IOE_Dir)	00h	R	0				端口 E 数据方向寄存器。 0=IN, 1=OUT.			
			W	-				端口 E 数据方向寄存器。 0=IN, 1=OUT.			
\$0044	端口 E 属性 (P_IOE_Attrib)	00h	R	0				端口 E 属性寄存器			
			W	-				端口 E 属性寄存器			

注：\$0044 复位后的值取决于 IOINIT(\$7FE2.0)的设置。

(7) \$0046~\$0049: UART 控制寄存器

地址	功能	复位值	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
\$0046	UART 控制 (P_UART_Ctrl)	00h	R	RXIE	TXIE	RXEN	TXEN	SOFTRS T	STOPSEL	PSEL	PEN
			W	RXIE	TXIE	RXEN	TXEN	SOFTRS T	STOPSEL	PSEL	PEN
\$0047	UART 波特率分频器 (P_UART_Baud)	00h	R	UARTB AUD7	UARTB AUD6	UARTB AUD5	UARTB AUD4	UARTB AUD3	UARTB AUD2	UARTB AUD1	UARTB AUD0
			W	UARTB AUD7	UARTB AUD6	UARTB AUD5	UARTB AUD4	UARTB AUD3	UARTB AUD2	UARTB AUD1	UARTB AUD0
\$0048	UART 状态 (P_UART_Status)	00h	R	RXIF	TXIF	BUSY	0	0	OERR	PERR	FERR
			W	-	-	-	-	-	OERR	PERR	FERR
\$0049	UART 数据 (P_UART_Data)	00h	R	UART 接收的数据							
			W	UART 发送的数据							

(8) \$0058~\$005D: 端口数据锁存寄存器

地址	功能	复位值 (ERST)	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
\$0058	捕获控制 (P_CAP_Ctrl)	00h	R	CAPOP T	0	CAPIP3	CAPIP2	CAPIP1	CAPIP0	CAP1E S	CAP0E S
			W	CAPOP T	-	CAPIP3	CAPIP2	CAPIP1	CAPIP0	CAP1E S	CAP0E S

地址	功能	复位值 (ERST)	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
\$0059	端口 A 数据锁存器	00h	R	端口 A 数据锁存							
	(P_IOA_Buf)		W	端口 A 数据锁存							
\$005A	端口 B 数据锁存器	00h	R	端口 B 数据锁存							
	(P_IOB_Buf)		W	端口 B 数据锁存							
\$005B	端口 C 数据锁存器	00h	R	端口 C 数据锁存							
	(P_IOC_Buf)		W	端口 C 数据锁存							
\$005C	端口 D 数据锁存器	00h	R	端口 D 数据锁存							
	(P_IOD_Buf)		W	端口 D 数据锁存							
\$005D	端口 D 数据锁存器	00h	R	0				端口 E 数据锁存			
	(P_IOE_Buf)		W	-				端口 E 数据锁存			

5.2.4 系统运行控制寄存器

在 SPMC65P2708A/2716A 芯片中, 位于\$0030~\$0036 的寄存器是与系统运行相关的, 需要慎重配置。为了保证写入这些寄存器的内容正确且防止数据或地址总线上

的噪声干扰, 对它们加强了写保护。用户对它们进行写操作时, 必须连续写两次才能成功。

[例] 5.2.2 设置\$0030 ~\$0036 间的主要寄存器 (两次写入操作)

```

lda    #$FF                                ;清除复位标志
sta    P_SYS_Ctrl
sta    P_SYS_Ctrl
lda    #C_MODE_Reset                       ; 复位所有 IO 端口
sta    P_MODE_Ctrl
sta    P_MODE_Ctrl
lda    #C_WDT_Div_16384 :                   ; WDI= Fslow(25KHz)/16384= 1.5Hz
sta    P_WDT_Ctrl
sta    P_WDT_Ctrl
lda    #C_IRQOpt1_IRQ0ES                   ; 设置 INT0 为上升沿触发, 其它外部中断为下降沿触发
sta    P_IRQ_Opt1
sta    P_IRQ_Opt1
lda    #$00                                ; 设置低电压复位值为 2.5V
sta    P_LVR_Opt
sta    P_LVR_Opt

```

5.2.5 芯片配置寄存器

SPMC65P2708A/2716A 提供 3 个字节的寄存器用于系统配置。在进行芯片烧录的时候，这些寄存器的设置会通过烧录器写入芯片。这样，芯片在上电或外部复位后，能正常启动运转。关于寄存器设置的详情，请参见 5.13。

5.2.6 用户信息块

SPMC65P2708A/2716A 预留了 16 字节（\$7FF0 ~ \$7FFF）作为用户信息块，用户可以写入任意信息。

凌阳创新科技公司将 \$7FF0~\$7FF3 这 4 个字节作为芯片的序列号信息设置位，供用户写入序列号，其它字节可以随意定义。

5.3 系统时钟输入

SPMC65P2708A/2716A 支持三种时钟源：晶体/陶瓷振荡器、RC 振荡器和外部时钟源。可通过设置寄存器（\$7FE0）来选择不同类型的时钟源（详情参见 5.13）。

表 5-1 列出了 RC 振荡器在不同系统时钟频率时的阻容值。用户应根据这些值来设置系统时钟。

注：若选择晶体/陶瓷振荡器或外部时钟作为时钟源，时钟频率将被二分频，然后作为系统时钟（F_{sys}）提供给 CPU，供其正常工作。所以若想得到 8MHz 的系统时钟，晶体/陶瓷振荡器或外部时钟必须设为 16MHz。

表 5-1 RC 振荡器的阻容值与系统时钟频率的对应关系

Frequency (Hz), VDD=5.0V	Resistor (Ω), C=50PF
F _{sys} = 135K	75k
F _{sys} = 195K	51k
F _{sys} = 480K	20k
F _{sys} = 0.9M	10k
F _{sys} = 1.6M	5.1k
F _{sys} = 2.25M	3.3k

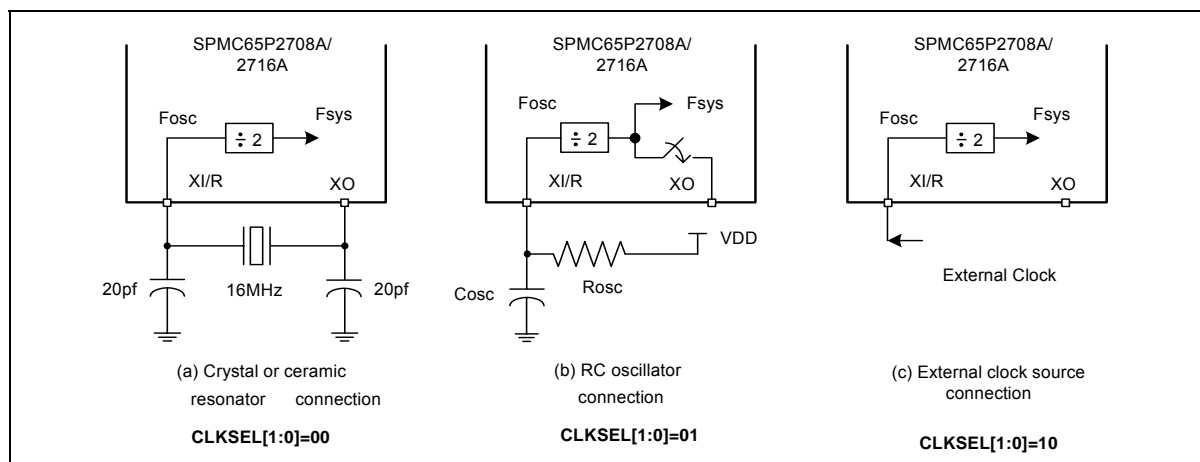


图 5-7 三种时钟源的典型应用电路

5.4 省电模式

5.4.1 简介

SPMC65P2708A/2716A 有三种工作模式：正常 (NORMAL) 模式、STOP 模式和 HALT 模式。为降低芯片的功耗，可以启用省电模式：STOP 模式和 HALT 模式。二者的唤醒时间不同，用户需根据需要进行选择合适的省电模式。注意，须连续向相应的控制寄存器连续写两次值才能进入 STOP 模式或 HALT 模式。关于 STOP 模式和 HALT 模式的详情，请参见随后的章节。

图 5-8 表明了 SPMC65P2708A/2716A 的模式转换关系。芯片上电复位后直接进入 NORMAL 模式工作，可切换到 STOP 模式或 HALT 模式。进入 STOP 模式或 HALT 模式后，系统将只能返回到 NORMAL 模式。例如，若用户想从 HALT 模式切换到 STOP 模式，系统将首先从 HALT 模式切换到 NORMAL 模式，然后再从 NORMAL 模式切换到 STOP 模式。反之亦然。

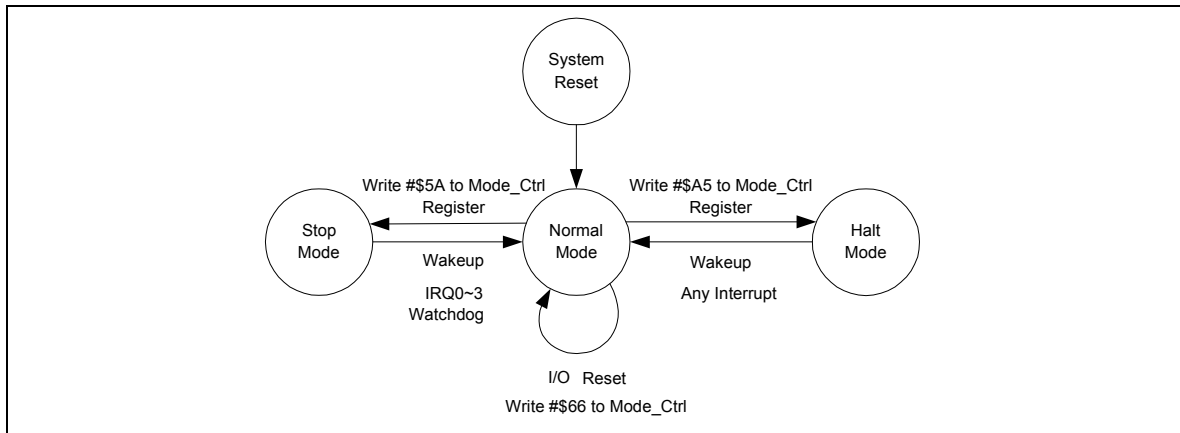


图 5-8 SPMC65P2708A/2716A 的工作模式转换

5.4.2 STOP 模式

一旦进入 STOP 模式，系统时钟将会停止工作（包括时钟发生电路）。只有来自 IO 管脚的外部中断或者看门狗中断才能唤醒系统重新进入 NORMAL 模式。

注意，若采用看门狗中断作为唤醒源，功耗将比采用外部中断唤醒（看门狗关闭）大。

STOP 模式被中断唤醒后，需要一定的时间来稳定振荡，从而获得稳定的时钟源。延迟时间范围为 20ms~60ms，一般设为 40ms。时钟源稳定后，若用户使能了中断（总的中断开关被打开：执行指令 CLI），程序会立即跳转到中断服务子程序去执行。否则，接着执行进入 STOP 模

式指令的下一条指令，不会进入中断。

为确保中断可以唤醒系统，在进入 STOP 模式之前必须打开相应的中断使能位。

若使用看门狗来唤醒，则必须在进入 STOP 模式之前打开看门狗定时器，并且要设置合适的看门狗定时溢出频率，以避免在振荡稳定时间内发生系统复位。参考图 5-9。

向寄存器 P_Mode_Ctrl 连续两次写 #5A，系统进入 STOP 模式。

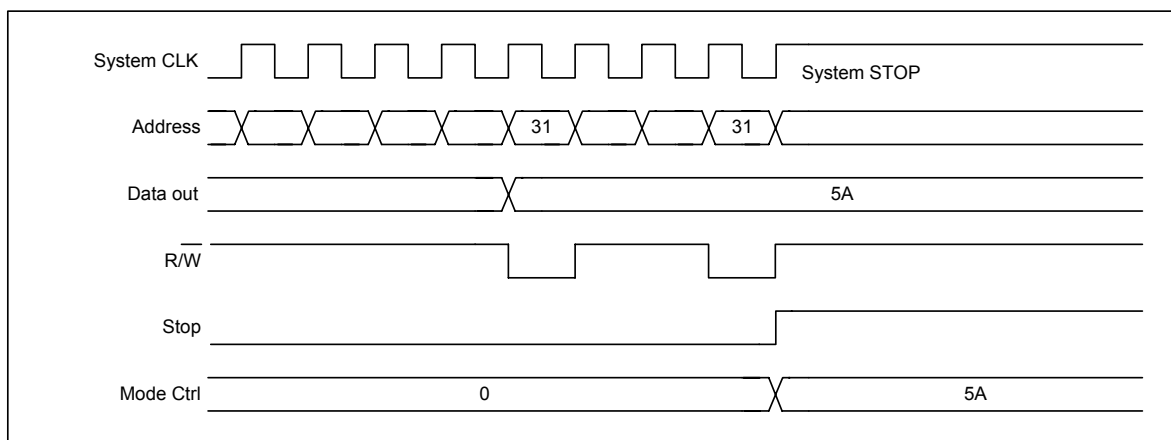


图 5-9 进入 STOP 模式的时序

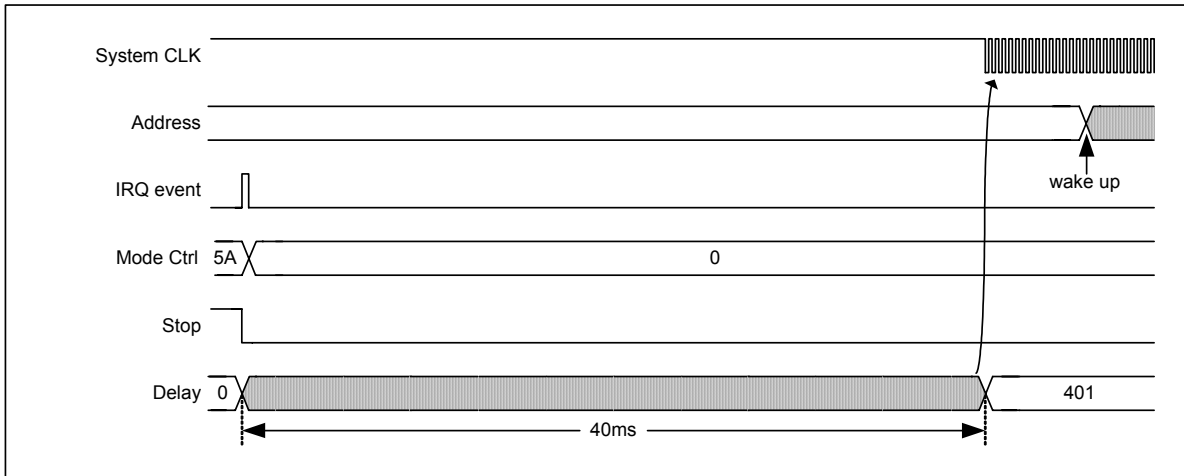


图 5-10 外部中断唤醒 STOP 模式的时序

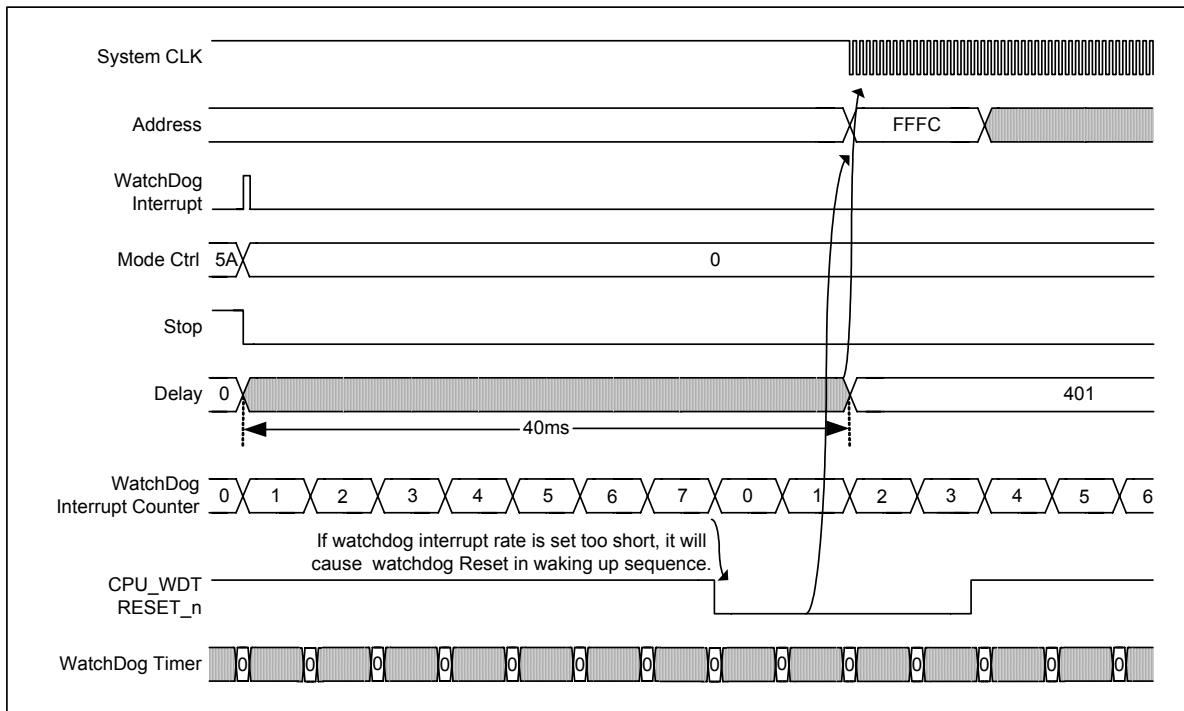


图 5-11 看门狗计数的溢出频率过快，唤醒 STOP 模式失败

5.4.3 HALT 模式

在 HALT 模式下，系统时钟停止，但是其外围正常工作。

HALT 模式下，所有的中断源都可以作为唤醒源。被唤醒后，不需要延迟时间。若用户使能了中断（总的中断开关被打开：执行指令 CLI），程序会立即跳转到中断服务子程序去执行。否则，接着执行进入 HALT 模式指令

的下一条指令，不会进入中断。从 HALT 模式回到 NORMAL 模式无时间延迟。

向寄存器 P_Mode_Ctrl 连续两次写#5A5，系统即进入 HALT 模式。

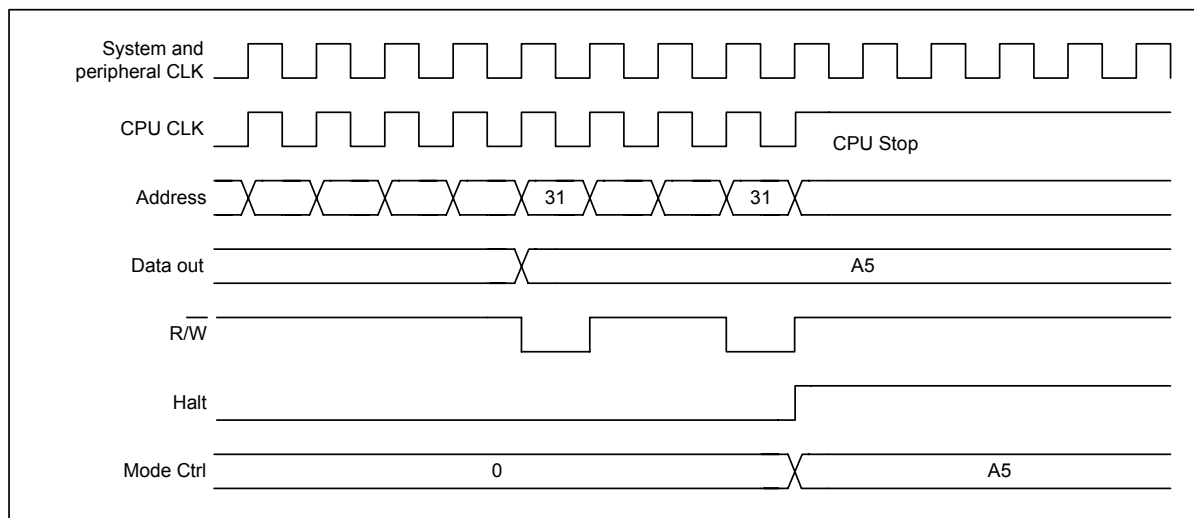


图 5-12 进入 HALT 模式的时序

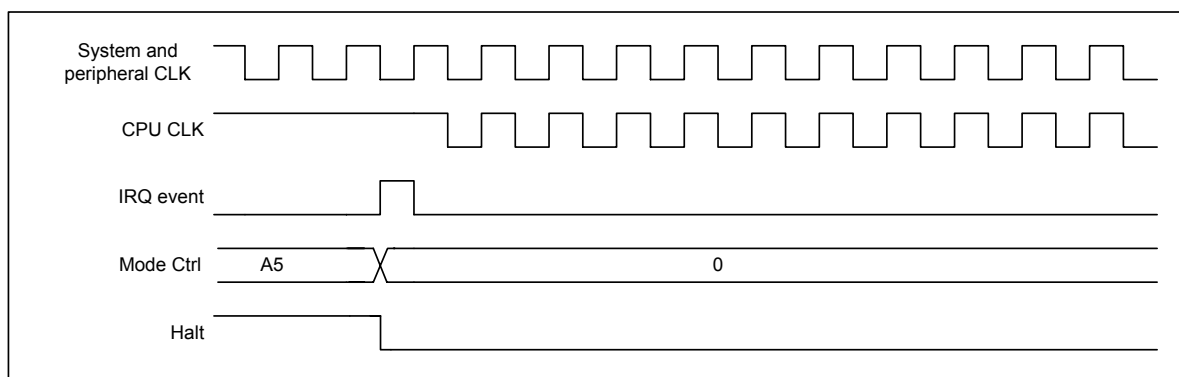


图 5-13 中断

图 5-14 唤醒 HALT 模式的时序

■ 省电模式控制寄存器(P_MODE_Ctrl, \$0031)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	Mode_Ctrl7	Mode_Ctrl6	Mode_Ctrl5	Mode_Ctrl4	Mode_Ctrl3	Mode_Ctrl2	Mode_Ctrl1	Mode_Ctrl0
读 / 写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
默认值	0	0	0	0	0	0	0	0

注：该字节必须连续写两次才能设置生效。

Bit [7:0]: **Mode ctrl** [7:0]: 省电模式选择位

读:

数据总为#\$00

写(两次):

#\$5A =进入 STOP 模式

#\$A5 =进入 HALT 模式

#\$66 =复位除 CPU 外的所有内部模块

【例】5.4.1使 MCU 进入 STOP 模式

lda	#C_MODE_STOP	;进入 STOP 模式命令
sta	P_MODE_Ctrl	
sta	P_MODE_Ctrl	

5.5 中断

5.5.1 简述

SPMC65P2708A/2716A 芯片共有 7 种中断源：外部中断、定时/计数器中断、看门狗中断、ADC 中断、实时中断、通讯中断和捕获中断。外部中断可分为非屏蔽中断和可屏蔽中断 2 种，其中非屏蔽中断的优先级高于可屏蔽中断。下面对它们进行具体描述。

5.5.2 外部中断

SPMC65P2708A/ 2716A 有 6 个外部中断源：INT0、INT1、INT2、INT3、INT4 和 INT5，它们分别对应端口 PB4、PB5、PD0、PD1、PD4 和 PD5。中断信号与中断标志位及控制位一起作用使 CPU 产生外部中断。

对于所有的外部中断通道，每个中断通道都有自己独立的中断控制位和标志位，可以分别设置和清除。一旦外部中断发生，相应中断标志位便会被置 1，直到软件清除。允许外部中断时，除了设置中断控制寄存器外，还需使用 ‘CLI’ 指令清除状态寄存器中的中断屏蔽位。

中断被使能后，才会有中断请求信号产生。每个中断都对应一个寄存器，用来设置中断的触发模式：边沿触发或电平触发。当外部中断被设置为沿触发模式时，一个有效的边沿输入便会触发中断。如果设置为电平触发，一个持续有效的电平会令中断产生，直到该电平被去掉。

5.5.3 非屏蔽中断

6 个外部中断中只能选择 1 个设置为非屏蔽中断(NMI)。非屏蔽中断选择寄存器地址为\$7FE3。非屏蔽中断的设置与可屏蔽中断的设置方法相同。

当非屏蔽中断被使能后，无论 CPU 当前状态如何，它随时可以产生中断。换言之，当 CPU 正在处理 NMI 中断时，另一个 NMI 中断信号仍可以打断当前的 CPU 运行，而去执行新的中断服务子程序，因此，非屏蔽中断 NMI 只能设置为边沿触发而非电平触发，以避免中断频繁产生使堆栈溢出。

5.5.4 其它中断

除外部中断，SPMC65P2708A/2716A 还支持其它 6 种中断源：定时/计数器中断、看门狗中断、ADC 中断、实时中断、通讯中断和捕获中断。这些中断源都有各自的标志位和控制寄存器，分别用于标志中断是否发生和设置中断的禁止/使能。通常，一旦中断事件发生，相应的中断标志位会被置 1。同时，若相应中断控制位被设置为使能，便会产生中断请求信号，CPU 接受中断请求去执行相应中断服务程序。如果相应的中断控制被禁止，则不会产生中断请求信号，从而 CPU 也不会进入中断服务子程序，但是，可以通过查询中断标志位来判断中断事件是否发生。注意在 CPU 进入了中断服务程序进行中断处理时，必须清除中断标志位，否则，CPU 会循环进入中断服务程序。

关于定时/计数器中断、看门狗中断、ADC 中断、实时中断、通讯中断和捕获中断，在后续章节介绍。

表 5-2 中断源列表

中断源		中断标志寄存器	中断控制寄存器	中断源		中断标志寄存器	中断控制寄存器
Timer Overflow	Timer0	T0OIF(\$000E.0)	T0OIE(\$000F.0)	External INT Input	IRQ0	IRQ0IF(\$000C.0)	IRQ0IE(\$000D.0)
	Timer1	T1OIF(\$000E.1)	T1OIE(\$000F.1)		IRQ1	IRQ1IF(\$000C.1)	IRQ1IE(\$000D.1)
	Timer2	T2OIF(\$000E.2)	T2OIE(\$000F.2)		IRQ2	IRQ2IF(\$000C.2)	IRQ2IE(\$000D.2)
	Timer3	T3OIF(\$000E.3)	T3OIE(\$000F.3)		IRQ3	IRQ3IF(\$000C.3)	IRQ3IE(\$000D.3)
					IRQ4	IRQ4IF(\$000C.4)	IRQ4IE(\$000D.4)
IRQ5	IRQ5IF(\$000C.5)	IRQ5IE(\$000D.5)					
Capture	Capture0	CAP0IF(\$000E.6)	CAP0IE(\$000F.6)	Analog	ADC	ADIF(\$000C.7)	ADIE(\$000D.7)
	Capture1	CAP1IF(\$000E.7)	CAP1IE(\$000F.7)				
	Capture2	CAP2IF(\$000C.0)	CAP2IE(\$000D.0)				
	Capture3	CAP3IF(\$000C.1)	CAP3IE(\$000D.1)	Communication	SPI	SPIIF(\$0026.2) R SPIIF(\$003A.7) R/W	SPIIE(\$003A.6)
			UART		RXIF(\$0048.7) TXIF(\$0048.6)	RXIE(\$0046.7) TXIE(\$0046.6)	
Watchdog		WDIF(\$000C.6)	WDIE(\$000D.6)	Interval Timer		ITVALIF(\$0026.5)	ITVALIE(\$0027.5)

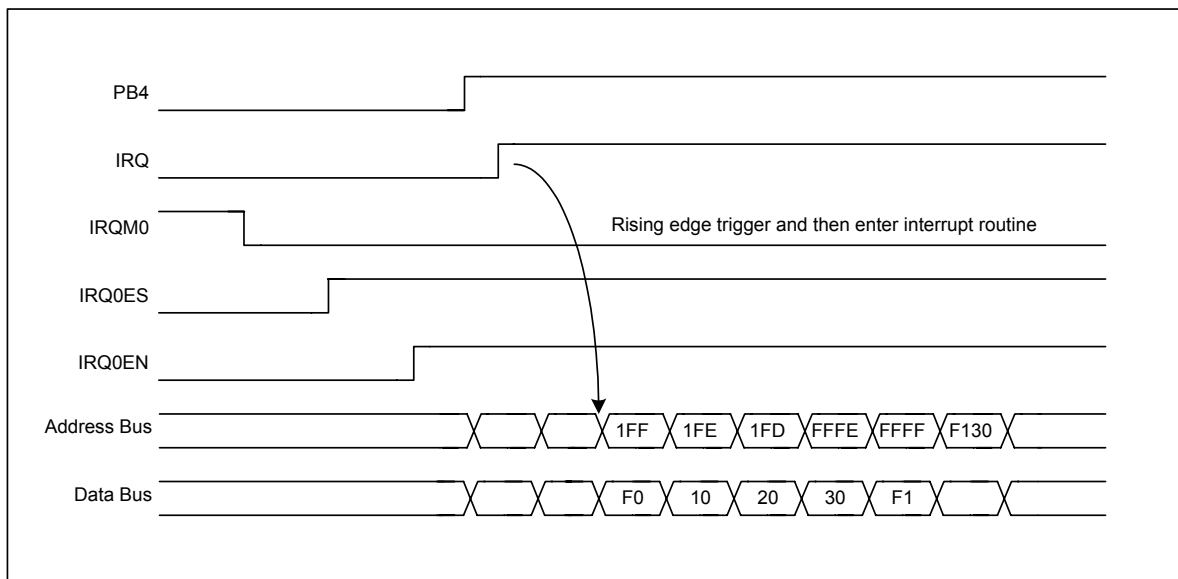


图 5-15 INT0 中断触发时序图(PB4)

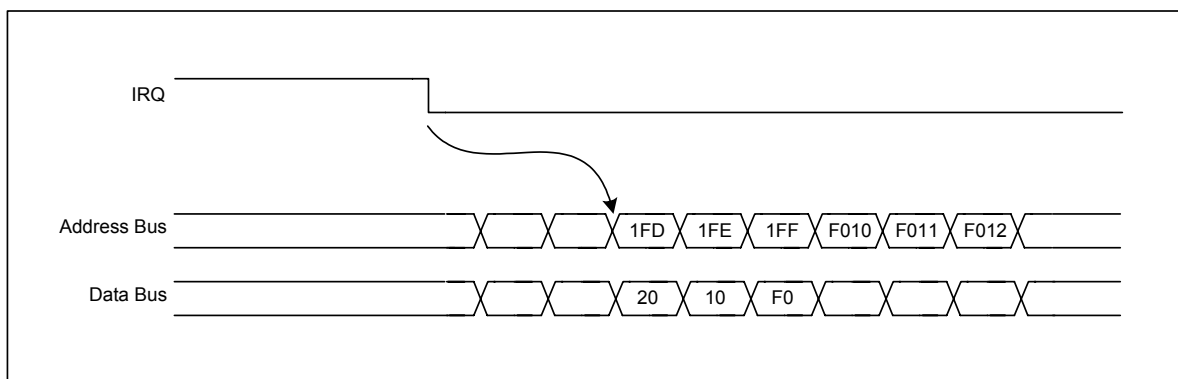


图 5-16 INT0 中断返回时序图

5.5.5 中断寄存器

(1) 中断标志寄存器 0 (P_INT_Flag0, \$000C)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	ADIF	WDIF	IRQ5IF	IR4IF	IRQ3IF	IRQ2IF	IRQ1IF/ CAP3IF	IRQ0IF/ CAP2IF
读 / 写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
默认值	0	0	0	0	0	0	0	0

注：向相应位写“1”，则清除中断标志。

Bit 7 **ADIF**: ADC 中断标志
0 = 中断没有发生

1 = 中断发生
Bit 6 **WDIF**: 看门狗中断标志
0 = 中断没有发生

1 = 中断发生

Bit 5 **IRQ5IF**: IRQ5 中断标志
0 = 中断没有发生
1 = 中断发生

Bit 4 **IRQ4IF**: IRQ4 中断标志
0 = 中断没有发生
1 = 中断发生

Bit 3 **IRQ3IF**: IRQ3 中断标志
0 = 中断没有发生
1 = 中断发生

Bit 2 **IRQ2IF**: IRQ2 中断标志
0 = 中断没有发生
1 = 中断发生

Bit 1 **IRQ1IF/CAP3IF**: IRQ1/CAP3 中断标志
0 = 中断没有发生
1 = 中断发生

当捕获 3 使能时，该位作为捕获中断标志，否则做为 IRQ1 中断标志。

Bit 0 **IRQ0IF/CAP2IF**: IRQ0/CAP2 中断标志
0 = 中断没有发生
1 = 中断发生

当捕获 2 使能时，该位作为捕获中断标志，否则做为 IRQ0 中断标志。

(2) 中断标志寄存器 1(P_INT_Flag1, \$000E)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	CAP1IF	CAP0IF	-	-	T3OIF	T2OIF	T1OIF	T0OIF
读 / 写	R/W	R/W	-	-	R/W	R/W	R/W	R/W
默认值	0	0	0	0	0	0	0	0

注：向相应位写“1”，则清除中断标志。

Bit 7 **CAP1IF**: 捕获 1 中断标志
0 = 中断没有发生
1 = 中断发生

0 = 中断没有发生
1 = 中断发生

Bit 6 **CAP0IF**: 捕获 0 中断标志
0 = 中断没有发生
1 = 中断发生

Bit 1 **T1OIF**: 定时/计数器 1 溢出中断标志
0 = 中断没有发生
1 = 中断发生

Bit [5:4] 保留

Bit 0 **T0OIF**: 定时/计数器 0 溢出中断标志
0 = 中断没有发生
1 = 中断发生

Bit 3 **T3OIF**: 定时/计数器 3 溢出中断标志
0 = 中断没有发生
1 = 中断发生

Bit 2 **T2OIF**: 定时/计数器 2 溢出中断标志

(3) 中断标志寄存器 2(P_INT_Flag2, \$0026)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	-	-	ITVALIF	-	UARTIF	SPIIF	-	-

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
读 / 写	-	-	R/W	-	R	R	-	-
默认值	0	0	0	0	0	0	0	0

注：向相应位写“1”，则清除中断标志。

Bit [7:6]	保留	1 = 中断发生
Bit 5	ITVALIF : 时基中断标志 0 = 中断没有发生 1 = 中断发生	Bit 2 SPIIF : SPI 中断标志 0 = 中断没有发生 1 = 中断发生
Bit 4	保留	SPIIF 标志位不能在本寄存器里直接清除,而必须在 SPI 状态寄存器(P_SPI_Status, \$003A)中清除。
Bit 3	UARTIF : UART 中断标志 0 = 中断没有发生	Bit [1:0] 保留

(4) 中断控制寄存器 0 (P_INT_Ctrl0, \$000D)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	ADIE	WDIE	IRQ5IE	IRQ4IE	IRQ3IE	IRQ2IE	IRQ1IE/ CAP3IE	IRQ0IE/ CAP2IE
读 / 写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
默认值	0	0	0	0	0	0	0	0

Bit 7	ADIE : A/D converter 中断使能位 0 = 禁止 1 = 使能	Bit 2	IRQ2IE : IRQ2 中断使能位 0 = 禁止 1 = 使能
Bit 6	WDIE : Watchdog 中断使能位 0 = 禁止 1 = 使能	Bit 1	IRQ1IE/ CAP3IE : IRQ1/CAP3 中断使能位 0 = 禁止 1 = 使能
Bit 5	IRQ5IE : IRQ5 中断使能位 0 = 禁止 1 = 使能		若捕获 3 使能,则该位作为捕获 3 的中断使能位,否则做为 IRQ1 使能位。
Bit 4	IRQ4IE : IRQ4 中断使能位 0 = 禁止 1 = 使能	Bit 0	IRQ0IE/ CAP2IE : IRQ0/CAP2 中断使能位 0 = 禁止 1 = 使能
Bit 3	IRQ3IE : IRQ3 中断使能位 0 = 禁止 1 = 使能		若捕获 2 使能,则该位作为捕获 2 的中断使能位,否则做为 IRQ10 使能位。

(5) 中断控制寄存器 1 (P_INT_Ctrl1, \$000F)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	CAP1IE	CAP0IE	-	-	T3OIE	T2OIE	T1OIE	T0OIE
读 / 写	R/W	R/W	-	-	R/W	R/W	R/W	R/W
默认值	0	0	0	0	0	0	0	0

- Bit 7 **CAP1IE:** 捕获 1 中断使能位
 0 = 禁止
 1 = 使能
- Bit 6 **CAP0IE:** 捕获 0 中断使能位
 0 = 禁止
 1 = 使能
- Bit [5:4] 保留
- Bit 3 **T3OIE:** 定时/计数器 3 溢出中断使能位
 0 = 禁止
 1 = 使能
- Bit 2 **T2OIE:** 定时/计数器 2 溢出中断使能位
 0 = 禁止
 1 = 使能
- Bit 1 **T1OIE:** 定时/计数器 1 溢出中断使能位
 0 = 禁止
 1 = 使能
- Bit 0 **T0OIE:** 定时/计数器 0 溢出中断使能位
 0 = 禁止
 1 = 使能

(6) 中断控制寄存器 2 (P_INT_Ctrl2, \$0027)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	-	-	ITVALIE	-	-	-	-	-
读 / 写	-	-	R/W	-	-	-	-	-
默认值	0	0	0	0	0	0	0	0

- 0 = 禁止
 1 = 使能
- Bit [7:6] 保留
- Bit 5 **ITVALIE:** 时基中断使能位
- Bit [4:0] 保留

(7) NMI 设置寄存器(P_NMI, \$7FE3)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	-	-	-	-	-	NMIS2	NMIS1	NMIS0
读 / 写	-	-	-	-	-	R	R	R
默认值	1	1	1	1	1	1	1	1

注：可以在 FortisIDE 环境下找到相应的工具栏进行设置。

Bit [7:3] 保留

011 = PD1 (IRQ3)作为 NMI 中断源

Bit [2:0] **NMIS** [2:0]: 非屏蔽中断源控制位

010 = PD0 (IRQ2)作为 NMI 中断源

111 = 禁止

001 = PB5 (IRQ1)作为 NMI 中断源

110 = 保留

000 = PB4 (IRQ0)作为 NMI 中断源

101 = PD5 (IRQ5)作为 NMI 中断源

100 = PD4 (IRQ4)作为 NMI 中断源

(8) IRQ 设置寄存器 0 (P_IRQ_Opt0, \$0033)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	-	-	-	-	IRQ5ES	IRQM5	IRQ4ES	IRQM4
读 / 写	-	-	-	-	R/W	R/W	R/W	R/W
默认值	0	0	0	0	0	0	0	0

注：该寄存器必须连续写两次入才能设置成功。

Bit [7:4] 保留

Bit 1 **IRQ4ES**: IRQ4 极性控制位

Bit 3 **IRQ5ES**: IRQ5 极性控制位

IRQM4="0"

IRQM5="0"

0 = 下降沿触发

0 = 下降沿触发

1 = 上升沿触发

1 = 上升沿触发

IRQM4="1"

IRQM5="1"

0 = 低电平触发

0 = 低电平触发

1 = 高电平触发

1 = 高电平触发

Bit 0 **IRQM4**: IRQ4 触发模式选择位

Bit 2 **IRQM5**: IRQ5 触发模式选择位

0 = 边沿触发

0 = 边沿触发

1 = 电平触发

1 = 电平触发

(9) IRQ 设置寄存器 1(P_IRQ_Opt1, \$0034)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	IRQ3ES	IRQM3	IRQ2ES	IRQM2	IRQ1ES/ CAP3ES	IRQM1	IRQ0ES/ CAP2ES	IRQM0
读 / 写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
默认值	0	0	0	0	0	0	0	0

注：该寄存器必须连续写两次入才能设置成功。

Bit 7 **IRQ3ES**: IRQ3 极性控制位

IRQM3="0"

	0 = 下降沿触发		Bit 2	IRQM1 : IRQ1 触发模式选择位
	1 = 上升沿触发			0 = 边沿触发
	IRQM3="1"			1 = 电平触发
	0 = 低电平触发		Bit 1	IRQ0ES/CAP2ES : IRQ 极性控制位
	1 = 高电平触发			0/CAP2ES
Bit 6	IRQM3 : IRQ3 触发模式选择位			IRQM0="0"
	0 = 边沿触发			0 = 下降沿触发
	1 = 电平触发			1 = 上升沿触发
Bit 5	IRQ2ES : IRQ2 极性控制位			IRQM0="1"
	IRQM2="0"			0 = 低电平触发
	0 = 下降沿触发			1 = 高电平触发
	1 = 上升沿触发			
	IRQM2="1"			
	0 = 低电平触发			
	1 = 高电平触发			
Bit 4	IRQM2 : IRQ2 触发模式选择位			
	0 = 边沿触发			
	1 = 电平触发			
Bit 3	IRQ1ES/CAP3ES : IRQ2 极性控制位或捕获器 3 控制位			
	IRQM1="0"			
	0 = 下降沿触发			
	1 = 上升沿触发			
	IRQM1="1"			
	0 = 低电平触发			
	1 = 高电平触发			

若捕获 2 使能，该位做为捕获 2 开始计数的边沿触发极性设置位，否则为 IRQ0 的极性控制位。

1=下降沿触发计数（下降沿清除计数器）

0=上升沿触发计数（上升沿清除计数器）

Bit 0	IRQM0 : IRQ0 触发模式选择位
	0 = 边沿触发
	1 = 电平触发

若捕获 3 使能，该位做为捕获 3 开始计数的边沿触发极性设置位，否则为 IRQ1 的极性控制位。

1 =下降沿触发计数（下降沿清除计数器）

0 =上升沿触发计数（上升沿清除计数器）

【例】5.5.1使能 IRQ0，上升沿触发

```

set      P_IRQ_Opt1, CB_IRQOpt1_IRQ0ES      ; INT0 上升沿触发
set      P_IRQ_Opt1, CB_IRQOpt1_IRQ0ES
lda      #$FF
sta      P_INT_Flag0                        ; 清除中断标志位
set      P_INT_Ctrl0, CB_INT_IRQ0IE        ; 使能 INT0
    
```

5.6 复位

5.6.1 简述

系统中共有 6 种复位源：上电复位(POR)、外部复位(ERST)、低电压复位(LVR)、看门狗复位(WDR)、非法地址复位(IAR)和软件复位。这些复位源可以分为外部复位和内部复位。外部复位来自电源线或外部触发事件；内部复位来自程序异常或程序设置的软件复位。表 5-3 说明了这些复位发生后对 CPU 和整个系统的影响。

5.6.2 上电复位 (POR)

当芯片的电源电压 VDD 从 0V 上升到 1.45V 左右时，上电复位电路便会产生上电时序，此时，芯片内部的一个计数器开始计数，大约经过约 80ms 的延迟（电源稳定约 40ms，系统时钟稳定约 40ms），系统便以设定的频率开始正常运转。

上电复位将会复位整个芯片和所有的寄存器。通常情况下，RESETB 管脚需要外接 RC 电路，这样，上电复位发生后，经过一段延迟，RESETB 管脚才能变为高电平。对于 SPMC65X 系列芯片，其优点为：RESETB 管脚只需直接拉高（或接一个上拉电阻）到 VDD，就可立即复位，不需外接 RC 电路，不会造成上电延迟。

5.6.3 外部复位 (ERST)

SPMC65P2708A/2716A 提供一个外部复位管脚（RESETB）用于强制系统复位。复位电路如图 5-17 所示，RESETB 管脚连接一个外部 RC 电路。由于 RESETB 为低电平有效，所以当 RESETB 管脚上的电压降至 $0.3 * VDD$ 以下时，系统复位。

外部复位信号脉宽至少要持续 200ns 以上，否则将会被当作干扰信号而滤去，对系统不会产生任何影响。外部复位信号有效后，系统发生复位，当 RESETB 上升为高电平后，再经过 40ms 和 1024 个系统时钟周期，程序重新开始执行。复位时序见图 5-19。

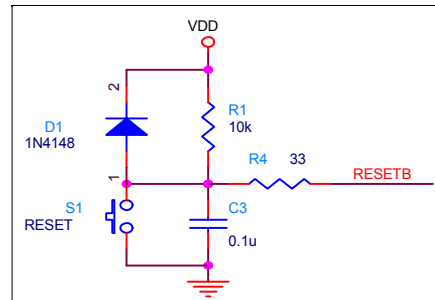


图 5-17 外部复位电路

注：

当 VDD 电源下降时，二极管 D1 帮助电容迅速放电。

推荐 $R1 < 40K\Omega$ 以保证电阻的分压符合芯片的电气指标。

如果 $R4 = 33\Omega \sim 1K\Omega$ ，将会限制流入 RESETB 的电流，从而提高了抗干扰能力 (ESD)。

5.6.4 低电压复位 (LVR)

低电压复位(LVR)即当 MCU 的电源电压低于设定的复位电压后，芯片复位，保证了 MCU 不会在非正常的电压范围内连续工作。

通过芯片配置寄存器 (\$7FE0.2) 可以禁止/使能 LVR 功能。使能 LVR 功能后，LVR 电路便开始对电源电压进行监测，如果监测到电源电压低于设定电压持续 1024 个系统时钟周期，系统便会复位。当电压回升到设定值后，再经过 1024 个系统时钟周期，程序重新开始执行。

用户可通过控制位 LVRV40 (P_LVR_Opt, \$0036.0) 来设置复位电压。若 LVRV40 设置为“0”，则电压低于 2.5V 时复位；若设置为“1”，电压低于 4.0V 时复位。

通常情况下，低电压复位不能清除寄存器 P_SYS_Ctrl 的里面的值。另外，低电压复位在上电复位或外部复位时无效。

另外，在 STOP 模式下，低电压复位会被禁止。如果电源电压过低，将直接触发上电复位。

5.6.5 看门狗复位 (WDR)

看门狗定时器被使能的情况下，当程序运行异常（跑飞或进入死循环等）或没有在规定时间内清除看门狗，看

门狗电路就会产生内部复位信号，将 CPU 复位，以保证 MCU 不会在异常情况下连续工作。看门狗定时器可以通过芯片配置寄存器(\$7FE0)使能或禁止。

看门狗电路内置独立的 RC 振荡器，通过寄存器 P_WDT_Ctrl[6:4]的设置，RC 振荡频率可以经过 8 种分频选择，分频后做为看门狗定时器的时钟源。发生看门狗复位后，CPU 被复位并重新执行程序，但外围电路不会被复位；同时，看门狗定时器的计数值会被清除，但是其复位标志仍然保留。为了防止看门狗定时器计数溢出，用户必须定期向 P_WDT_Clr 寄存器写入“#\$55”。关于看门狗定时器的具体描述，见 5.12.1。

5.6.6 非法地址复位 (IAR)

非法地址复位(IAR)是内部复位，可以防止系统进入非法地址。当程序对某个地址进行读写操作时，这个地址即不在工作区域，也不在堆栈区或者对 OTP ROM 区域写命令时，便会产生非法地址复位信号，从而使 CPU 复位。但外围电路不会复位。

5.6.7 软件复位

除了以上几种复位方式之外，SPMC65P2708A/2716A 还提供了一种软件复位方法。当 CPU 向 P_Mode_Ctrl 寄存器写入立即数\$66 后，所有外围电路复位但 CPU 不会复位。软件复位可用来在危急状况下迅速关闭所有功能。

表 5-3 各种复位的影响范围

	POR	ERST	LVR	WDR	IAR	Software
CPU reset	Yes	Yes	Yes	Yes	Yes	No
Peripheral reset	Yes	Yes	Yes	No	No	Yes
P_IOA_Attrib	vvvvvvvv	vvvvvvvv	vvvvvvvv	xxxxxxxx	xxxxxxxx	vvvvvvvv
P_IOB_Attrib	vvvvvvvv	vvvvvvvv	vvvvvvvv	xxxxxxxx	xxxxxxxx	vvvvvvvv
P_IOC_Attrib	vvvvvvvv	vvvvvvvv	vvvvvvvv	xxxxxxxx	xxxxxxxx	vvvvvvvv
P_IOD_Attrib	vvvvvvvv	vvvvvvvv	vvvvvvvv	xxxxxxxx	xxxxxxxx	vvvvvvvv
P_IOE_Attrib	---vvvvv	---vvvvv	---vvvvv	---xxxxx	---xxxxx	---vvvvv
P_SYS_Ctrl	100-00--	x1x-xx--	xx1-xx--	xxx-1x--	xxx-x1--	xxx-xx--
P_LVR_Opt	-----0	-----x	-----x	-----x	-----x	-----x

注：v：取决于 P_SECU.IOINIT 的设置 x：无影响 -：保留位

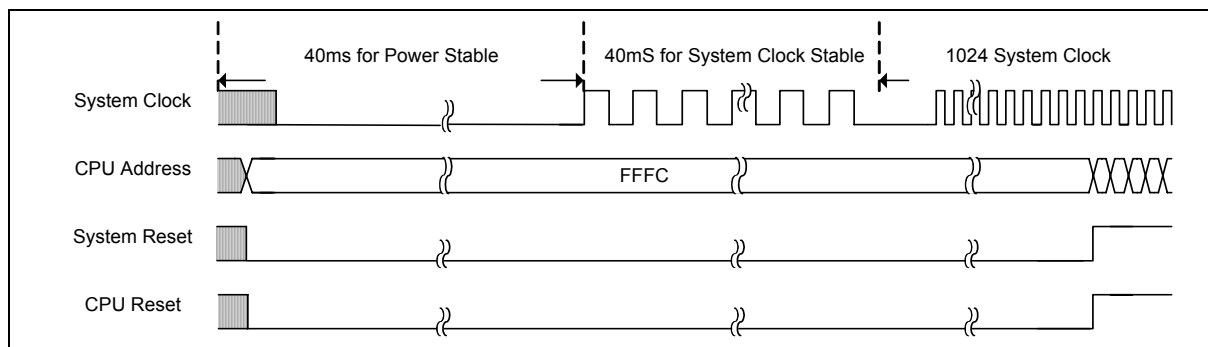


图 5-18 上电复位时序

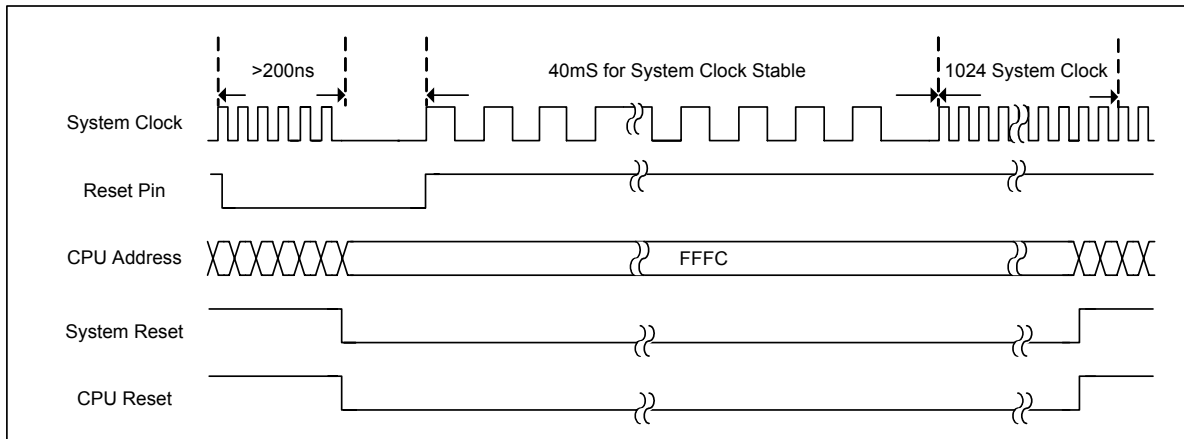


图 5-19 外部复位时序

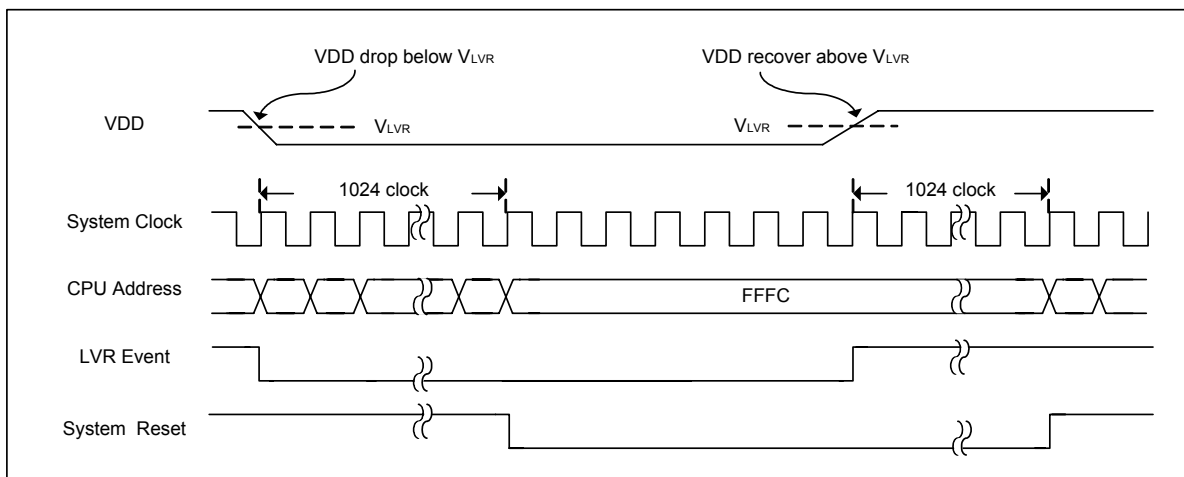


图 5-20 低电压复位时序

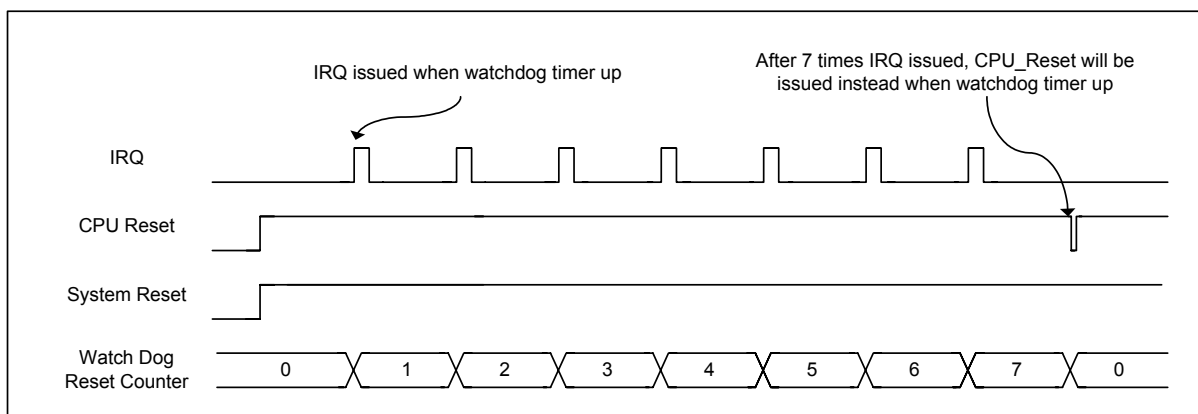


图 5-21 看门狗复位时序

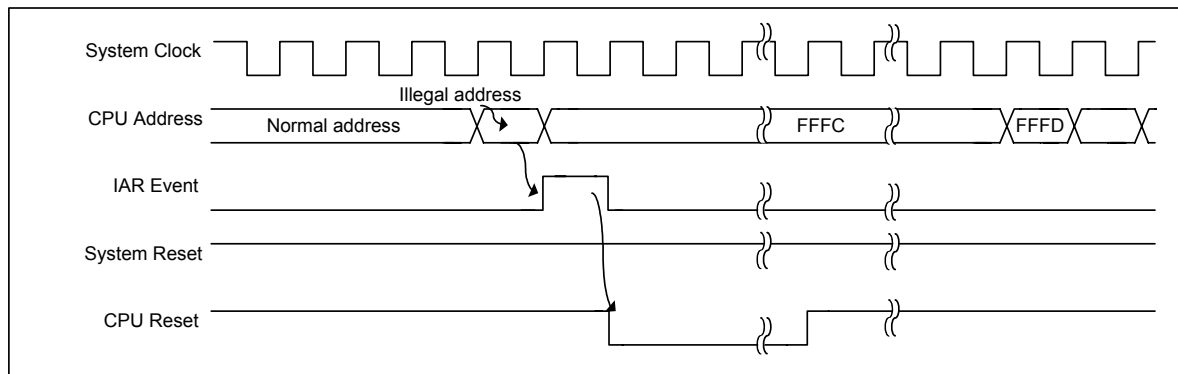


图 5-22 非法地址复位时序

(2) 系统控制寄存器(P_SYS_Ctrl, \$0030)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	POR	ERST	LVR	-	WDR	IAR	-	-
读 / 写	R/W	R/W	R/W	-	R/W	R/W	-	-
默认值	1	0	0	0	0	0	0	0

注：该寄存器必须两次写入才能设置成功。

Bit 7 POR: 上电复位标志

0 = 无上电复位发生

1 = 上电复位发生

用于表示上电复位的发生。上电复位会将芯片所有内部模块复位。

Bit 6 ERST: 外部复位标志

0 = 无外部复位发生

1 = 外部复位发生

用于表示来自 RESETB 管脚上的外部复位的发生。外部复位会将除 P_SYS_Ctrl 寄存器和 P_LVR_Opt 寄存器之外的所有内部模块复位。

Bit 5 LVR: 低电压复位标志

0 = 无低电压复位发生

1 = 低电压复位发生

用于表示在低电压复位使能情况下,低电压复位的发生。

低电压复位会将除 P_SYS_Ctrl 寄存器之外的所有内部模块复位。

Bit 4 保留

Bit 3 WDR: 看门狗复位标志

0 = 无看门狗复位发生

1 = 看门狗复位发生

用于表示在看门狗复位使能情况下,看门狗复位的发生。低电压复位仅复位 CPU, 其他内部模块不变。

Bit 2 IAR: 非法地址复位标志

0 = 无非法地址复位发生

1 = 非法地址复位发生

用于表示系统检测到非法地址时,非法地址复位的发生。非法地址复位仅复位 CPU, 其他内部模块不变。

Bit [1:0] 保留

注：向相应位写“1”清除该标志。

(3) 低电压复位控制寄存器(P_LVR_Opt, \$0036)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	-	-	-	-	-	-	-	LVRV40
读 / 写	-	-	-	-	-	-	-	R/W
默认值	0	0	0	0	0	0	0	0

注：该寄存器必须写两次入才能设置成功。

Bit [7:1] 保留

Bit 0 **LVRV40**: 复位电压选择位

0 = 低于 2.5V 复位

1 = 低于 4.0V 复位

注：芯片工作后，该选项只能设置一次，只有上电复位才能将其清除。

[例] 5.6.1 使能低电压复位(复位电压=4V)

```
set P_LVR_Opt,C_LVR_V40 ;设置复位电压为 4.0V
set P_LVR_Opt,C_LVR_V40
```

[例] 5.6.2 保存寄存器 P_SYS_Ctrl 中的值然后将其清除

```
lda P_SYS_Ctrl ; 读复位标志
sta G_MWorkReg1
lda #$FF ; 清除复位标志
sta P_SYS_Ctrl
sta P_SYS_Ctrl
```

5.7 I/O 端口

5.7.1 简述

SPMC65P2708A/2716A 共有 5 组 IO 端口：端口 A、端口 B、端口 C、和端口 D 和端口 E。这些端口即可以做为普通 IO 口，也可以复用为特殊功能端口。在初始状态时，它们都作为通用输入端口。通常，对端口某一位的设定包括以下 4 个寄存器：数据锁存寄存器 Buffer、数据寄存器 Data、属性寄存器 Attribution 和方向寄存器 Direction。各寄存器的每个对应位组合在一起，形成一个控制字，用来定义相应 I/O 管脚的属性。

下面介绍 I/O 端口设置的规则：

- 方向寄存器 Direction 用来设置端口是输入还是输出。
- 属性寄存器 Attribution 用来设置端口是悬浮还是不悬浮
- 数据锁存寄存器 Buffer 的设置影响各管脚的初始值。当某位作为输入时，该寄存器决定端口是上拉还是下拉；当作为输出时，决定端口输出高电平还是低电平。另外，当使用位指令对 IO 口进行操作时，

必须对数据锁存器 Buffer 来做操作，而不应该对数据寄存器 Data 进行操作，以避免错误。

- 数据寄存器 Data 用来读取各端口的电平状态。若对该寄存器进行写操作，数据会同时被写到数据锁存寄存器 Buffer。

表 5-4 I/O 配置

方向(P_IOX_Dir)	属性(P_IOX_Attrib)	数据(P_IOX_Data)	功能	描述
0	0	0	下拉	带下拉电阻的输入管脚
0	0	1	上拉	带上拉电阻的输入管脚
1	0	1	高电平输出	输出高电平
1	0	0	低电平输出	输出低电平
X	1	X	悬浮	悬浮式输入管脚

注意：

P_IOX_Buf (x=A,B,C,D,E)是一个专门用于存储端口(PX)数据的寄存器，并用来对端口各管脚进行位操作，而不直接用 P_IOX_Data 寄存器来操作。当端口做为输出，向寄存器 P_IOX_Buf 或者 P_IOX_Data 中写数据时，效果相同。但是，当使用位操作指令 SET, TST,CLR 或者 INV 对寄存器 P_IOX_Data 进行位操作时，寄存器 P_IOX_Buf 的值可能会发生错误。因此，强烈建议只对寄存器 P_IOX_Buf (X=A,B,C,D,E)进行位操作。如果要从外部管脚读入数据，必须有专门的 P_IOx_Data。

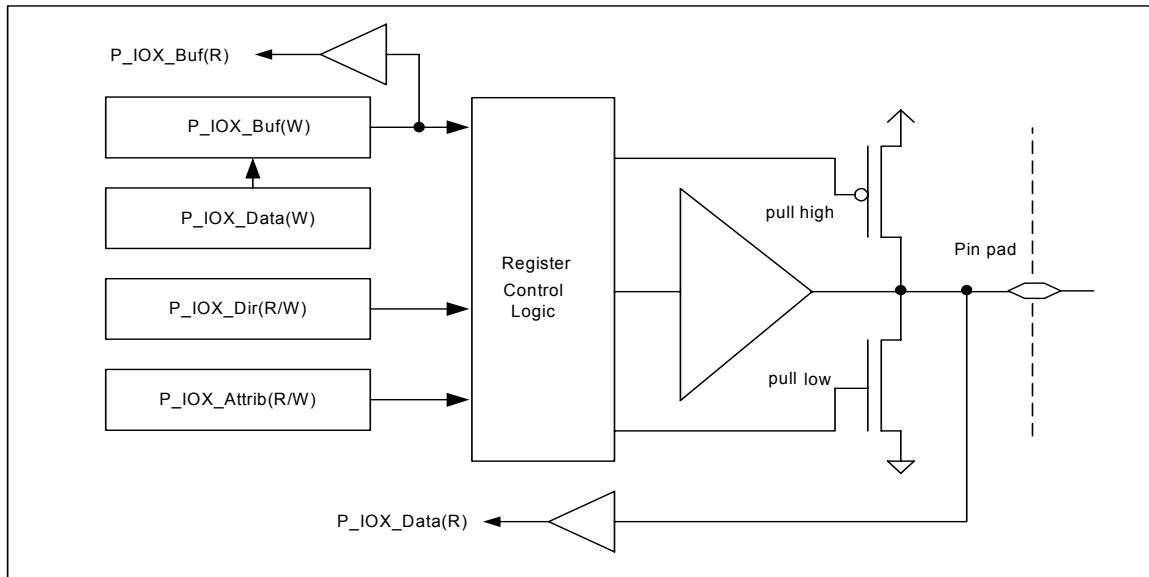


图 5-23 IO 端口结构图

5.7.2 端口 A

端口 A 有 8 个双向可编程 IO 管脚，并且有相应的控制寄存器来进行设置。这些控制寄存器为：数据锁存寄存器 P_IOA_Buf、数据寄存器 P_IOA_Data、方向寄存器

P_IOA_Dir 和属性寄存器 P_IOA_Attrib。

P_IOA_Data 用于访问端口 A。读 P_IOA_Data 将得到

端口 A 各管脚的电平状态，向 P_IOA_Data 写入数据，其值将会存入 P_IOA_Buf 中。P_IOA_Buf 是一个专门用于缓存写入端口 A 数据的寄存器。

复位后，端口 A 默认为通用的 IO 端口，除此之外，端口 A 还可以作为 AD 转换的输入通道。

表 5-5 端口 A 各管脚功能列表

管脚	BIT#	复用功能
PA0	Bit0	通用输入/输出或 AN0 模拟输入
PA1	Bit1	通用输入/输出或 AN1 模拟输入
PA2	Bit2	通用输入/输出或 AN2 模拟输入
PA3	Bit3	通用输入/输出或 AN3 模拟输入
PA4	Bit4	通用输入/输出或 AN4 模拟输入
PA5	Bit5	通用输入/输出或 AN5 模拟输入
PA6	Bit6	通用输入/输出或 AN6 模拟输入
PA7	Bit7	通用输入/输出或 AN7 模拟输入

(2) 端口 A 数据寄存器(P_IOA_Data, \$0000)

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOA_Data							
读/写	R/W							
默认值	00h							

Bit [7:0] **P_IOA_Data**: 端口 A 数据寄存器

读: 读取端口 A 外部管脚上的电平状态值

写: 写入数据会通过相应的管脚输出，数据同时保存到寄存器 P_IOA_Buf 中

(3) 端口 A 数据锁存器(P_IOA_Buf, \$0059)

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOA_Buf							
读/写	R/W							
默认值	00h							

Bit [7:0] **P_IOA_Buf**: 端口 A 数据锁存器

端口 A 设为输入且 P_IOA_Attrib="0"时，

0 = 带下拉电阻的输入

1 = 带上拉电阻的输入

端口 A 设为输出且 P_IOA_Attrib="0"时，

0 = 低电平输出

1 = 高电平输出

(4) 端口 A 方向寄存器(P_IOA_Dir, \$0004)

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOA_Dir							
读/写	R/W							
默认值	00h							

Bit [7:0] **P_IOA_Dir**: 端口 A 方向寄存器

0 = 输入

1 = 输出

(5) 端口 A 属性寄存器(P_IOA_Attrib, \$0008)

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOA_Attrib							
读/写	R/W							
默认值	00h							

Bit [7:0] **P_IOA_Attrib**: 端口 A 属性寄存器

0 =输入/输出

1 =悬浮式输入

【例】5.7.1 设置端口 A[7:0]为低电平输出

```
lda    #$FF                                ; 把$FF 写入累加器
sta    P_IOA_Dir
lda    #$00
sta    P_IOA_Attrib
sta    P_IOA_Data
```

【例】5.7.2 设置端口 A[7:0]为带下拉电阻的输入

```
lda    #$00                                ; 把$00 写入累加器
sta    P_IOA_Dir
sta    P_IOA_Attrib
sta    P_IOA_Data
```

5.7.3 端口 B

端口 B 有 8 个双向可编程 IO 管脚，并且有相应的控制寄存器来进行设置。这些控制寄存器为：数据锁存寄存器 P_IOB_Buf、数据寄存器 P_IOB_Data、方向寄存器 P_IOB_Dir 和属性寄存器 P_IOB_Attrib。

P_IOB_Data 用于访问端口 B。读 P_IOB_Data 将得到

端口 B 各管脚的电平状态，向 P_IOB_Data 写入数据，其值将会存入 P_IOB_Buf 中。P_IOB_Buf 是一个专门用于缓存写入端口 B 数据的寄存器。

复位后，端口 B 默认为通用的 IO 端口，除此之外，端口 B 的每个管脚都有相应的特殊功能（具体情况见表

5-6)。例如：PB6 和 PB7 可以做为慢速输出管脚。将寄存器 P_IO_Opt (\$0035)的 bit0 设置为 1，便会打开 PB[7:6]的慢速输出功能。当 PB[7:6]输出从高电平变为低电平时，将会延迟约 250ns，具体延迟时间由系统时

钟（F_{SYS}）决定。

当 MCU 在进行远距离通信时，慢速输出功能可以防止产生电磁干扰。

表 5-6 端口 B 各管脚功能列表

管脚	BIT	复用功能
PB0	Bit0	通用输入输出、捕获器 0 输入、定时/计数器 0 外部时钟输入
PB1	Bit1	通用输入输出、捕获器 0 输入、定时/计数器 1 外部时钟输入
PB2	Bit2	通用输入输出、Timer0 比较输出、Timer0 PWM 输出
PB3	Bit3	通用输入输出、Timer1 比较输出、Timer1PWM 输出
PB4	Bit4	通用输入输出、外部中断 0 输入、捕获器 2 输入、定时/计数器 2 外部时钟输入
PB5	Bit5	通用输入输出、外部中断 1 输入、捕获器 3 输入、定时/计数器 3 外部时钟输入
PB6	Bit6	通用输入输出、慢速输出、蜂鸣器输出
PB7	Bit7	通用输入输出、慢速输出、ADC 外部参考电压输入

(2) 端口 B 数据寄存器(P_IOB_Data, \$0001)

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOB_Data							
读/写	R/W							
默认值	00h							

Bit [7:0] **P_IOB_Data**: 端口 B 数据寄存器

读：读取端口 B 外部管脚上的电平状态值

写：写入数据会通过相应的管脚输出，数据同时保存到寄存器 P_IOB_Buf 中

(3) 端口 B 数据锁存器(P_IOB_Buf, \$005A)

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOB_Buf							
读/写	R/W							
默认值	00h							

Bit [7:0] **P_IOB_Buf**: 端口 B 数据锁存寄存器

端口 B 设为输入且 P_IOB_Attrib="0"时，

0 = 带下拉电阻的输入

1 = 带上拉电阻的输入

端口 B 设为输出且 P_IOB_Attrib="0"时,

0 = 低电平输出

1 = 高电平输出

(4) 端口 B 方向寄存器(P_IOB_Dir, \$0005)

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOB_Dir							
读/写	R/W							
默认值	00h							

Bit [7:0] **P_IOB_Dir**: 端口 B 方向寄存器

0 = 输入

1 = 输出

(5) 端口 B 属性寄存器(P_IOB_Attrib, \$0009)

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOB_Attrib							
读/写	R/W							
默认值	00h							

Bit [7:0] **P_IOB_Attrib**: 端口 B 属性寄存器

0 =输入/输出

1=悬浮式输入

(6) 慢速输出控制寄存器(P_IO_Opt, \$0035)

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	-	-	-	-	-	-	-	SLOWE
读/写	-	-	-	-	-	-	-	R/W
默认值	0	0	0	0	0	0	0	0

注意：必须连续写入两次，设置才生效。

Bit [7:1] 保留

Bit 0 **SLOWE**: 慢速输出功能使能位

1=开启 PB[7:6]的慢速输出功能

0=关闭 PB[7:6]的慢速输出功能

[例] 5.7.3 设置端口 B[3:0]为低电平输出，端口 B[7:4]为带上拉电阻的输入。

```
lda    #$0F                                ;将$0F 写入累加器
sta    P_IOB_Dir
lda    #$00
sta    P_IOB_Attrib
lda    #$F0
sta    P_IOB_Data
```

[例] 5.7.4 设置 Port B[7:0]为悬浮式输入

```
lda    #$FF                                ;将$FF 写入累加器
sta    P_IOB_Attrib
```

5.7.4 端口 C

端口 C 有 8 个双向可编程 IO 管脚，并且有相应的控制寄存器来进行设置。这些控制寄存器为：数据锁存寄存器 P_IOC_Buf、数据寄存器 P_IOC_Data、方向寄存器 P_IOC_Dir 和属性寄存器 P_IOC_Attrib。

P_IOC_Data 用于访问端口 C。读 P_IOC_Data 将得到端口 C 各管脚的电平状态，向 P_IOC_Data 写入数，其值将会存入 P_IOC_Buf 中。P_IOC_Buf 是一个专门用于缓存写入端口 C 数据的寄存器。复位后，端口 C 默认为通用的 IO 端口，除此之外，端口 C 还具备 SPI 及 UART 功能。

表 5-7 端口 C 管脚功能列表

管脚	BIT	功能
PC0	Bit0	通用输入输出、SPI 从机模式选择
PC1	Bit1	通用输入输出、SPI 时钟输出/输入
PC2	Bit2	通用输入输出、SPI 数据输入
PC3	Bit3	通用输入输出、SPI 数据输出
PC4	Bit4	通用输入输出、UART 数据发送
PC5	Bit5	通用输入输出、UART 数据接收
PC6	Bit6	通用输入输出、无复用功能
PC7	Bit7	通用输入输出、无复用功能

(2) 端口 C 数据寄存器(P_IOC_Data, \$0002)

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOC_Data							
读/写	R/W							
默认值	00h							

Bit [7:0] **P_IOC_Data**: 端口 C 数据寄存器

读：读取端口 C 外部管脚上的电平状态值

写：写入数据会通过相应的管脚输出，数据同时保存到寄存器到寄存器 P_IOC_Buf 中

(3) 端口 C 数据锁存器(P_IOC_Buf, \$005B)

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOC_Buf							
读/写	R/W							
默认值	00h							

Bit [7:0] **P_IOC_Buf**: 端口 C 数据锁存器

端口 C 设为输入且 P_IOC_Attrib="0"时,

0 = 带下拉电阻的输入

1 = 带上拉电阻的输入

端口 C 设为输出且 P_IOC_Attrib="0"时,

0 =低电平输出

1 =高电平输出

(4) 端口 C 方向寄存器(P_IOC_Dir, \$0006)

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOC_Dir							
读/写	R/W							
默认值	00h							

Bit [7:0] **P_IOC_Dir**: 端口 C 方向寄存器

0 = 输入

1 = 输出

(5) 端口 C 属性寄存器(P_IOC_Attrib, \$000A)

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOC_Attrib							
读/写	R/W							
默认值	00h							

Bit [7:0] **P_IOC_Attrib**: 端口 C 属性寄存器

0 =输入/输出

1 =悬浮式输入

[例] 5.7.5 设置端口 C[7:0]为低电平输出

```
lda    #$3F                ; 将$3F 写入累加器
```

```

sta   P_IOC_Dir
lda   #$00
sta   P_IOC_Attrib
sta   P_IOC_Data
    
```

【例】5.7.6 设置端口 C[7:0]为悬浮式输入

```

lda   #$3F           ; 将$3F 写入累加器
sta   P_IOC_Attrib
    
```

5.7.5 Port D

端口 D 有 8 个双向可编程 IO 管脚，并且有相应的控制寄存器来进行设置。这些控制寄存器为：数据锁存寄存器 P_IOD_Buf、数据寄存器 P_IOD_Data、方向寄存器 P_IOD_Dir 和属性寄存器 P_IOD_Attrib。

P_IOD_Data 用于访问端口 D。读 P_IOD_Data 寄存器将得到端口 D 各管脚的电平状态，向 P_IOD_Data 写入数，其值将会存入 P_IOD_Buf 中。P_IOD_Buf 是一个专门用于缓存写入端口 D 数据的寄存器。复位后，端口 D 默认为通用的 IO 端口，除此之外，端口 D 还具备一些特殊功能。

表 5-8 端口 D 管脚功能列表

管脚	BIT	功能
PD0	Bit0	通用输入输出、外部中断 2 输入
PD1	Bit1	通用输入输出、外部中断 3 输入
PD2	Bit2	通用输入输出、Timer3 比较输出、Timer3 PWM 输出
PD3	Bit3	通用输入输出、Timer2 比较输出
PD4	Bit4	通用输入输出、外部中断 4 输入
PD5	Bit5	通用输入输出、外部中断 5 输入
PD6	Bit6	通用输入输出、无复用功能
PD7	Bit7	通用输入输出、无复用功能

(2) 端口 D 数据寄存器(P_IOD_Data, \$0003)

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOD_Data							
读/写	R/W							
默认值	00h							

Bit [7:0] **P_IOD_Data**: 端口 D 数据寄存器

读：读取端口 D 外部管脚上的电平状态值

写：写入数据到 P_IOD_Buf 中

(3) 端口 D 数据锁存器(P_IOD_Buf, \$005C)

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOD_Buf							
读/写	R/W							
默认值	00h							

Bit [7:0] **P_IOD_Buf**: 端口 D 数据锁存寄存器

端口 D 设为输入且 P_IOD_Attrib="0"时,

0 = 带下拉电阻的输入

1 = 带上拉电阻的输入

端口 D 设为输出且 P_IOD_Attrib="0"时,

0 = 低电平输出

1 = 高电平输出

(4) 端口 D 方向寄存器(P_IOD_Dir, \$0007)

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOD_Dir							
读/写	R/W							
默认值	00h							

Bit [7:0] **P_IOD_Dir**: 端口 D 方向寄存器

0 = 输入

1 = 输出

(5) 端口 D 属性寄存器(P_IOD_Attrib, \$000B)

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOD_Attrib							
读/写	R/W							
默认值	00h							

Bit [7:0] **P_IOD_Attrib**: 端口 D 属性寄存器

0 =输入/输出

1=悬浮式输入

【例】5.7.7 设置 Port D[7:0]为高电平输出

```

lda    #$1F                                ; 将$1F 写入累加器 A 中
sta    P_IOD_Dir
lda    #$00
sta    P_IOD_Attrib
lda    #$1F
sta    P_IOD_Data
    
```

5.7.6 Port E

SPMC65P2708A/2716A 端口 E 有 5 个双向可编程 IO 管脚, 并且有相应的控制寄存器来进行设置。端口 E 的控制寄存器为: 数据寄存器 P_IOE_Data、方向寄存器

P_IOE_Dir、属性寄存器 P_IOE_Attrib 和数据锁存寄存器 P_IOE_Buf。P_IOE_Data 用于访问端口 E。读 P_IOE_Data 寄存器将得到端口 E 各管脚的电平状态,

向 P_IOE_Data 写入数，其值将会存入 P_IOE_Buf 中。P_IOE_Buf 是一个专门用于缓存写入端口 E 数据的寄存器。复位后，端口 E 默认为通用的 IO 端口。

表 5-9 端口 E 管脚功能列表

管脚	BIT	复用功能
PE0	Bit0	通用输入输出、无复用功能
PE1	Bit1	通用输入输出、无复用功能
PE2	Bit2	通用输入输出、无复用功能
PE3	Bit3	通用输入输出、无复用功能
PE4	Bit4	通用输入输出、无复用功能

(2) 端口 E 数据寄存器(P_IOE_Data, \$0040)

位	-	-	-	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOE_Data							
读/写	R/W							
默认值	00h							

Bit [4:0] **P_IOE_Data**: 端口 E 数据寄存器

读: 读取端口 E 外部管脚上的电平状态值

写: 写入数据到 P_IOE_Buf 中

(3) 端口 E 数据锁存器(P_IOE_Buf, \$005D)

位	-	-	-	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOE_Buf							
读/写	R/W							
默认值	00h							

Bit [4:0] **P_IOE_Buf**: 端口 E 数据锁存寄存器

(4) 端口 E 方向寄存器(P_IOE_Dir, \$0042)

位	-	-	-	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOE_Dir							
读/写	R/W							
默认值	00h							

Bit [4:0] **P_IOE_Dir**: 端口 E 方向寄存器

0 = 输入

1 = 输出

(5) 端口 E 属性寄存器(P_IOE_Attrib, \$0044)

位	-	-	-	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOE_Attrib							
读/写	R/W							
默认值	00h							

Bit [4:0] **P_IOE_Attrib**: 端口 E 属性寄存器

0 =输入/输出

1=悬浮式输入

[例] 5.7.8 设置 Port E[4:0]为高电平输出

```

lda    #$1F                                ; 将$1F 写入累加器中
sta    P_IOE_Dir
lda    #$00
sta    P_IOE_Attrib
lda    #$1F
sta    P_IOE_Data
    
```

5.8 定时/计数器

5.8.1 简述

SPMC65P2708A 和 SPMC65P2716A 这两款芯片均配有 4 个定时/计数器。其中, Timer0、Timer1、Timer2 和 Timer3 均为 16 位。这 4 个定时/计数器为递增计数, 且具备强大的 CCP(Capture/Compare/PWM)功能, 即

捕获、比较和 PWM 输出功能。表 5-10为这 4 个定时/计数器的功能简介。所有功能可通过相应控制寄存器进行设置。

表 5-10 定时/计数器功能简介

	Timer/Event Counter		Capture		Compare		PWM
	8 bit	16 bit	8 bit	16 bit	8 bit	16 bit	
Timer 0	YES	YES	Width/Cycle	Width	YES	YES	8 bit
Timer 1	YES	YES	Width/Cycle	Width	YES	YES	12 bit
Timer 2	YES	YES	Width/Cycle	Width	YES	YES	8 bit
Timer 3	YES	YES	Width/Cycle	Width	YES	YES	12 bit

5.8.2 定时/计数器 0

定时/计数器 0 是 8 位定时/计数器, 可以采用内部时钟源, 也可以采用外部时钟源。每收到一个内部或外部的

时钟输入, 计数器加一, 当其计数到 255 时, 这时再来一个时钟信号, 计数器便会溢出。此时, 如果定时/计数

器的溢出中断被使能，便会产生溢出中断，Timer0 中断标志被置位。同时计数器重载计数初值。

其内部结构如图 5-24所示；计数溢出及中断时序如图 5-25所示。

设为 16 位工作方式时，计数最大值为 65535。当其计数到 65535 时，这时再来一个时钟信号，计数器便会溢出并触发溢出中断，同时，重新载入计数初值。

注意一个 8 位 CPU，总线宽度为 8 位，通常无法直接访问 16 位的数据。为了克服这样的局限，设计了一个数据寄存器定时/计数器 0 MSB 数据寄存器，用于 16 位数据高 8 位的读写操作，16 位的读写操作如图 5-26.所示，它有专门的读/写缓冲器。在读取 16 位数据时，用户需要先读出低字节，与此同时，高字节被锁存在缓冲器里，低字节读取结束，然后再读取高字节。直到 16 位数据全部读出，缓冲器的值才会改变。相反的，在写 16 位数据时，用户需要先写入高字节并被缓冲器自动保存，然后再写入低字节，直到 16 位数据全部写入，缓冲器的值才会改变，这样缓冲器中的高字节和低字节将会同时载入定时/计数器 0 中去。

下面介绍定时/计数器 0 的特点：

- 可读可写

- 时钟频率可以进行 8 级预分频选择
- 可选外部或内部时钟源
- 支持 8 位或 16 位定时/计数器
- 8 位工作模式，计数值从#\$FF 增到#\$00 时，中断溢出；16 位工作模式，计数值从#\$FFFF 增到#\$0000 时，中断溢出
- 支持 8 位/16 位捕获功能
- 支持 8 位/16 位比较功能
- 支持 8 位 PWM 输出

16 位定时/计数器读写顺序框图见图 5-26；其内部结构如图 5-27所示；计数溢出及中断时序见图 5-28。

【例】5.8.1 8 位定时/计数器 0 溢出频率计算公式

$$f_{T0} = \frac{F_{sys}}{\text{Timer_prescaler} \times (256 - \text{Timer_preload_value})}$$

f_{T0} : Timer overflow frequency

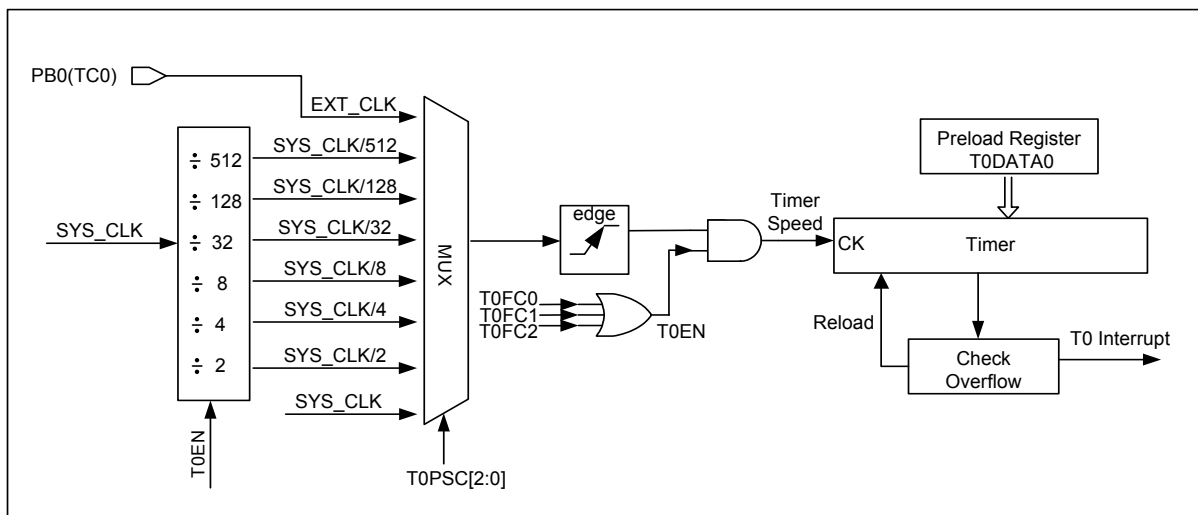


图 5-24 8 位定时/计数器的内部结构

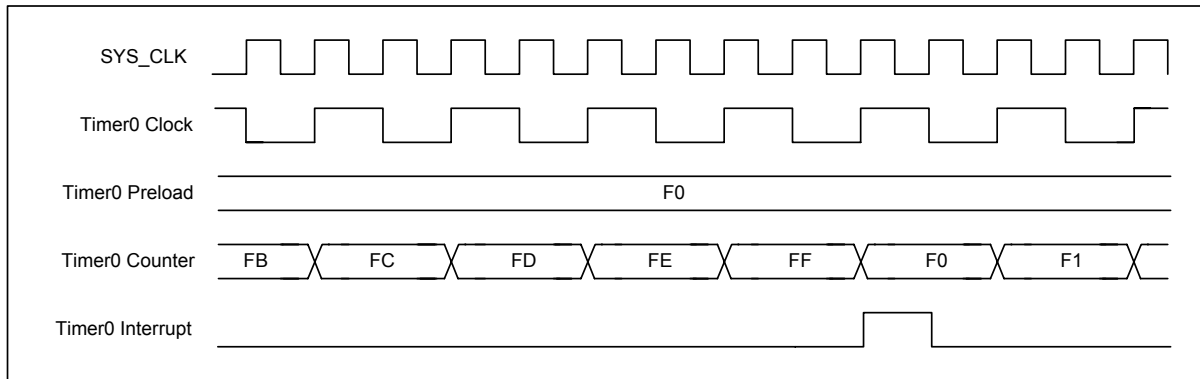


图 5-25 8 位定时/计数器计数溢出及中断时序

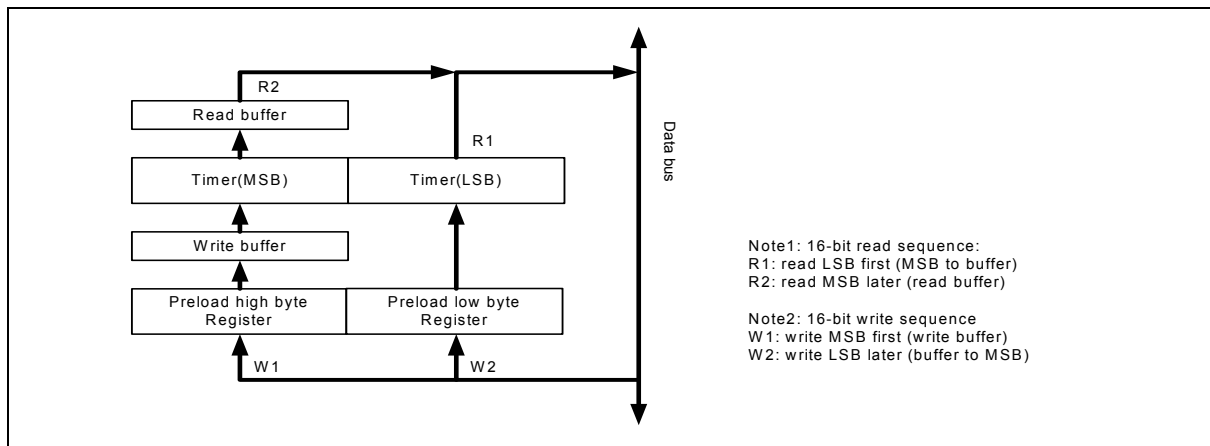


图 5-26 16 位定时/计数器读写顺序框图

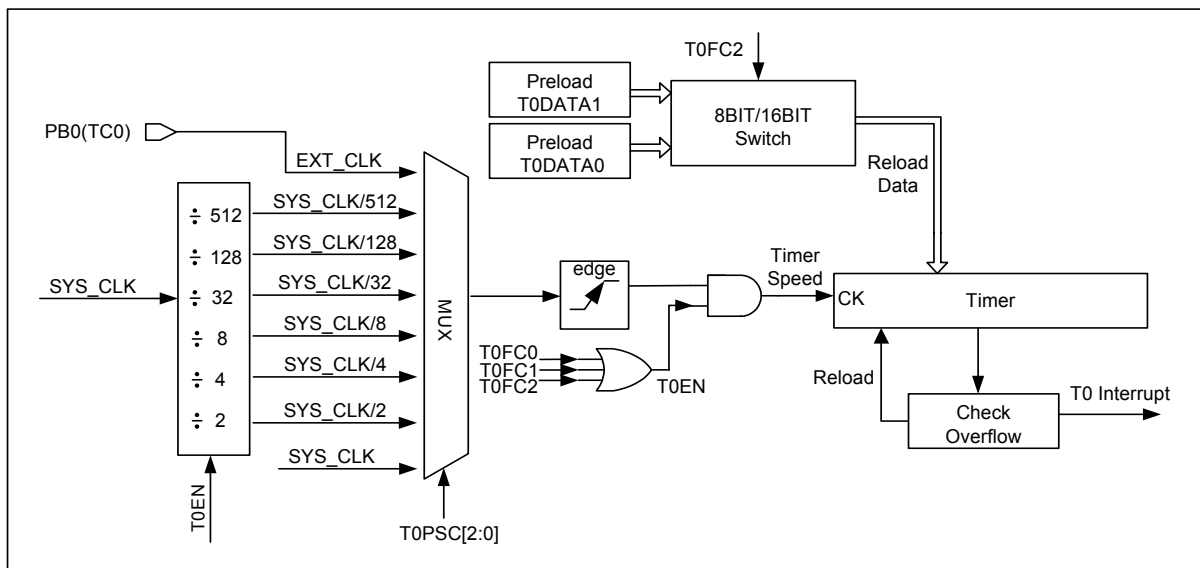


图 5-27 16 位定时/计数器的内部结构

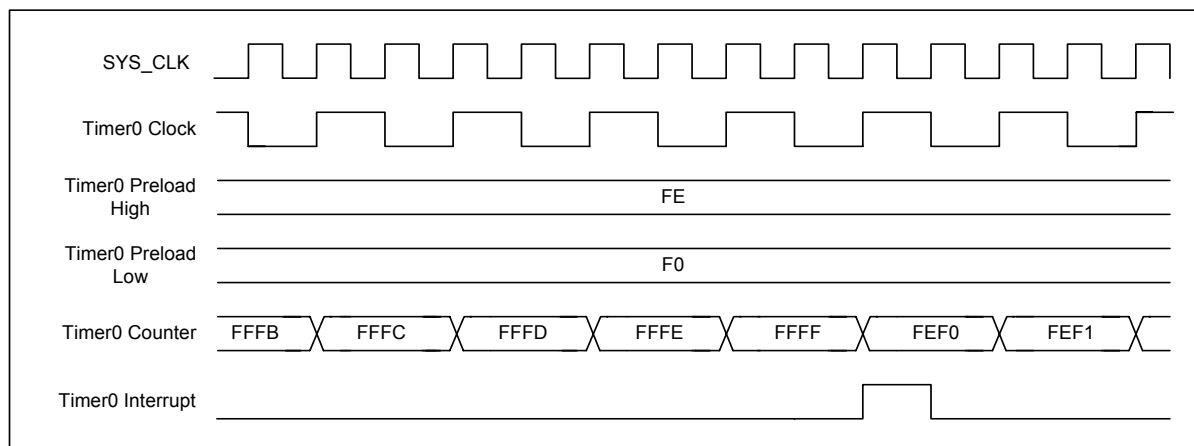


图 5-28 16 位定时/计数器计数溢出及中断时序

(2) 定时/计数器 0-1 控制寄存器 0 (P_TMR0_1_Ctrl0, \$0011)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	-	T1FC2	T1FC1	T1FC0	-	T0FC2	T0FC1	T0FC0
读 / 写	-	R/W	R/W	R/W	-	R/W	R/W	R/W
默认值	0	0	0	0	0	0	0	0

Bit 7 保留

Bit [6:4] **T1FC[2: 0]**: 定时/计数器 1 功能设置位

- 111 = 12 位 PWM
- 110 = 16 位脉宽捕获
- 101 = 16 位比较器
- 100 = 16 位定时/计数器
- 011 = 8 位脉宽/周期捕获
- 010 = 8 位比较器
- 001 = 8 位定时/计数器
- 000 = 禁止

Bit [2:0] **T0FC[2: 0]**: 定时/计数器 0 功能设置位

- 111 = 8 位 PWM 输出
- 110 = 16 位脉宽捕获
- 101 = 16 位比较器
- 100 = 16 位定时/计数器
- 011 = 8 位脉宽/周期捕获
- 010 = 8 位比较器
- 001 = 8 位定时/计数器
- 000 = 禁止

Bit 3 保留

(3) 定时/计数器 0-1 控制寄存器 1 (P_TMR0_1_Ctrl1, \$0012)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	-	T1PSC2	T1PSC1	T1PSC0	-	T0PSC2	T0PSC1	T0PSC0
读 / 写	-	R/W	R/W	R/W	-	R/W	R/W	R/W
默认值	0	0	0	0	0	0	0	0

Bit 7 保留

Bit [6:4] **T1PSC[2:0]**: 定时/计数器 1 预分频设置位

111 = 外部时钟频率输入

110 = $F_{SYS} \div 512$

101 = $F_{SYS} \div 128$

100 = $F_{SYS} \div 32$

011 = $F_{SYS} \div 8$

010 = $F_{SYS} \div 4$

001 = $F_{SYS} \div 2$

000 = F_{SYS}

Bit [2:0] **T0PSC[2:0]**: 定时/计数器 0 预分频设置位

111 = 外部时钟频率输入

110 = $F_{SYS} \div 512$

101 = $F_{SYS} \div 128$

100 = $F_{SYS} \div 32$

011 = $F_{SYS} \div 8$

010 = $F_{SYS} \div 4$

001 = $F_{SYS} \div 2$

000 = F_{SYS}

Bit 3 保留

FSYS: 系统时钟频率

(4) 定时/计数器 0 计数寄存器 (P_TMR0_Count, \$0013)

		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
8/16-bit Timer	W	T0PLV_7	T0PLV_6	T0PLV_5	T0PLV_4	T0PLV_3	T0PLV_2	T0PLV_1	T0PLV_0
	R	T0R_7	T0R_6	T0R_5	T0R_4	T0R_3	T0R_2	T0R_1	T0R_0
8/16-bit Compare	W	T0COV_7	T0COV_6	T0COV_5	T0COV_4	T0COV_3	T0COV_2	T0COV_1	T0COV_0
	R	T0R_7	T0R_6	T0R_5	T0R_4	T0R_3	T0R_2	T0R_1	T0R_0
8/16-bit Capture	W	T0PLV_7	T0PLV_6	T0PLV_5	T0PLV_4	T0PLV_3	T0PLV_2	T0PLV_1	T0PLV_0
	R	T0CWV_7	T0CWV_6	T0CWV_5	T0CWV_4	T0CWV_3	T0CWV_2	T0CWV_1	T0CWV_0
8-bit PWM	W	T0PPV_7	T0PPV_6	T0PPV_5	T0PPV_4	T0PPV_3	T0PPV_2	T0PPV_1	T0PPV_0
	R	T0PPV_7	T0PPV_6	T0PPV_5	T0PPV_4	T0PPV_3	T0PPV_2	T0PPV_1	T0PPV_0
默认值		0	0	0	0	0	0	0	0

Bit [7:0] **P_TMR0_Count[7:0]**: 定时/计数器 0 计数寄存器

读: T0R_[7:0] 定时/计数器 0 计数值

8 位定时/计数模式:

写: T0PLV_[7:0] 定时/计数器 0 重载值

读: T0R_[7:0] 定时/计数器 0 计数值

8 位比较模式:

写: T0COV_[7:0] 定时/计数器 0 比较值

8 位捕获模式:

写: T0PLV_[7: 0] 8 位捕获器 0 的重载值

读: T0CWV_[7: 0] 8 位捕获器 0 的捕获脉宽值

8 位 PWM 输出模式:

写: T0PPV_[7:0] 8 位 PWM0 的周期值

读: T0PPV_[7:0] 8 位 PWM0 的周期值

(5) 定时/计数器高字节寄存器(P_TMR0_CountHi, \$0014)

Bit		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
16-bit Timer	W	T0PLV_15	T0PLV_14	T0PLV_13	T0PLV_12	T0PLV_11	T0PLV_10	T0PLV_9	T0PLV_8
	R	T0R_15	T0R_14	T0R_13	T0R_12	T0R_11	T0R_10	T0R_9	T0R_8
16-bit Compare	W	T0COV_15	T0COV_14	T0COV_13	T0COV_12	T0COV_11	T0COV_10	T0COV_9	T0COV_8
	R	T0R_15	T0R_14	T0R_13	T0R_12	T0R_11	T0R_10	T0R_9	T0R_8
16-bit Capture	W	T0PLV_15	T0PLV_14	T0PLV_13	T0PLV_12	T0PLV_11	T0PLV_10	T0PLV_9	T0PLV_8
	R	T0CWV_15	T0CWV_14	T0CWV_13	T0CWV_12	T0CWV_11	T0CWV_10	T0CWV_9	T0CWV_8
8-bit Capture	W	T0PLV_15	T0PLV_14	T0PLV_13	T0PLV_12	T0PLV_11	T0PLV_10	T0PLV_9	T0PLV_8
	R	T0CCV_7	T0CCV_6	T0CCV_5	T0CCV_4	T0CCV_3	T0CCV_2	T0CCV_1	T0CCV_0
8-bit PWM	W	T0PDV_7	T0PDV_6	T0PDV_5	T0PDV_4	T0PDV_3	T0PDV_2	T0PDV_1	T0PDV_0
	R	T0PDV_7	T0PDV_6	T0PDV_5	T0PDV_4	T0PDV_3	T0PDV_2	T0PDV_1	T0PDV_0
默认值		0	0	0	0	0	0	0	0

Bit [7:0] **P_TMR0_CountHi [7:0]**: 定时/计数器 0 计数值的高字节

字节

16 位定时/计数器模式:

写: T0PLV_[15:8] 16 位定时/计数器 0 重载计数值的高字节

读: T0R_[15:8] 定时/计数器 0 计数值的高字节

16 位比较模式:

写: T0COV_[15:8] 16 位比较器 0 比较值的高字节

读: T0R_[15:8] 16 位定时/计数器 0 计数值的高

16 位捕获模式:

写: T0PLV_[15:8] 16 位捕获器 0 重载值的高字节

读: T0CWV_[15:8] 16 位捕获器 0 捕获脉宽值

8 位捕获模式:

写: T0PLV_[15:8] 16 位定时/计数器 0 重载值的高字节

读: T0CCV_[7:0] 8 位捕获器 0 捕获的周期值

8 位 PWM 输出模式:

读: TOPDV_[7:0] 8 位 PWM0 占空比值

写: TOPDV_[7:0] 8 位 PWM0 占空比值

[例] 5.8.2 设定时/计数器 0 为 8 位定时方式, 定时 1ms

```

lda    #6                ; 启动 Timer0 之前, 首先设置计数初值 (重载值)
sta    P_TMR0_Preload
lda    #C_T0FCS_Div_32   ; 设定 Timer0 时钟频率为 Fsys/32
sta    P_TMR0_1_Ctrl1
lda    #C_T08B_Timer     ; 将 Timer0 设置为 8 位定时/计数器工作方式
sta    P_TMR0_1_Ctrl0
lda    #6                ; 设定 Timer0 计数重载值为 256-6= 250
sta    P_TMR0_Preload    ; 设定 Fsys(8MHz)/32/250= 1KHz(1ms)
    
```

[例] 5.8.3 设定时/计数器 0 为 8 位比较模式

```

lda    #156              ; 启动 Timer0 之前, 首先设置计数初值 (重载值)
sta    P_TMR0_Preload
lda    #C_T0FCS_Div_128  ; 设定 Timer0 时钟频率为 Fsys/128
sta    P_TMR0_1_Ctrl1
lda    #C_T08B_COMP      ; 将 Timer0 设置为 8 位比较模式
sta    P_TMR0_1_Ctrl0
lda    #156              ; 设定 Timer0 计数重载值为 256-156= 100
sta    P_TMR0_Preload    ; PB2 脚输出频率:Fsys(8MHz)/128/100= 625Hz
    
```

5.8.3 定时/计数器 1

定时/计数器 1 与定时/计数器 0 功能完全相同, 因此关于定时/计数器 1 的控制, 请参见定时/计数器 0 章节中对控制寄存器的相关描述。

下面介绍定时/计数器 1 的特点:

- 可读可写
- 时钟频率可以进行 8 级预分频选择
- 可选外部或内部时钟源
- 可选 8 位/16 位工作方式
- 8 位工作方式下, 计数从#\$FF 增到#\$00 时溢出并产生溢出中断; 16 位工作方式下, 计数从#\$FFFF 增到#\$0000 时溢出并产生溢出中断
- 支持 8 位/16 位捕获功能
- 支持 8 位/16 位比较功能
- 支持 12 位 PWM 输出功能

(1) 定时/计数器 1 计数寄存器低字节(P_TMR1_Count, \$0015)

Bit		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
8/16-bit Timer	W	T1PLV_7	T1PLV_6	T1PLV_5	T1PLV_4	T1PLV_3	T1PLV_2	T1PLV_1	T1PLV_0
	R	T1R_7	T1R_6	T1R_5	T1R_4	T1R_3	T1R_2	T1R_1	T1R_0
8/16-bit	W	T1COV_7	T1COV_6	T1COV_5	T1COV_4	T1COV_3	T1COV_2	T1COV_1	T1COV_0

Bit		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Compare	R	T1R_7	T1R_6	T1R_5	T1R_4	T1R_3	T1R_2	T1R_1	T1R_0
8/16-bit Capture	W	T1PLV_7	T1PLV_6	T1PLV_5	T1PLV_4	T1PLV_3	T1PLV_2	T1PLV_1	T1PLV_0
	R	T1CWV_7	T1CWV_6	T1CWV_5	T1CWV_4	T1CWV_3	T1CWV_2	T1CWV_1	T1CWV_0
12-bit PWM	W	T1PPV_7	T1PPV_6	T1PPV_5	T1PPV_4	T1PPV_3	T1PPV_2	T1PPV_1	T1PPV_0
	R	T1PPV_7	T1PPV_6	T1PPV_5	T1PPV_4	T1PPV_3	T1PPV_2	T1PPV_1	T1PPV_0
默认值		0	0	0	0	0	0	0	0

Bit [7:0] **P_TMR1_Count** [7:0]: 定时/计数器 1 计数寄存器低字节

8 位/16 位定时/计数模式:

写: T1PLV_[7:0] 定时/计数器 1 重载值的低字节

读: T1R_[7:0] 定时/计数器 1 计数值的低字节

8 位/16 位比较模式:

写: T1COV_[7:0] 定时/计数器 1 比较值的低字节

读: T1R_[7:0] 定时/计数器 1 计数值的低字节

8 位/16 位捕获模式:

写: T1PLV_[7:0] 定时/计数器 1 重载值的低字节

读: T1CWV_[7:0] 定时/计数器 1 脉宽捕获值的低字节

12 位 PWM 模式:

写: T1PPV_[7:0] 定时/计数器 1 PWM 周期值的低 8 位

读: T1PPV_[7:0] 定时/计数器 1 PWM 周期值的低 8 位

(2) 定时/计数器 1 计数寄存器高字节(P_TMR1_CountHi, \$0016)

Bit		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
16-bit Timer	W	T1PLV_15	T1PLV_14	T1PLV_13	T1PLV_12	T1PLV_11	T1PLV_10	T1PLV_9	T1PLV_8
	R	T1R_15	T1R_14	T1R_13	T1R_12	T1R_11	T1R_10	T1R_9	T1R_8
16-bit Compare	W	T1COV_15	T1COV_14	T1COV_13	T1COV_12	T1COV_11	T1COV_10	T1COV_9	T1COV_8
	R	T1R_15	T1R_14	T1R_13	T1R_12	T1R_11	T1R_10	T1R_9	T1R_8
16-bit Capture	W	T1PLV_15	T1PLV_14	T1PLV_13	T1PLV_12	T1PLV_11	T1PLV_10	T1PLV_9	T1PLV_8
	R	T1CWV_5	T1CWV_4	T1CWV_3	T1CWV_2	T1CWV_1	T1CWV_0	T1CWV_9	T1CWV_8
8-bit Capture	W	T1PLV_15	T1PLV_14	T1PLV_13	T1PLV_12	T1PLV_11	T1PLV_10	T1PLV_9	T1PLV_8
	R	T1CCV_7	T1CCV_6	T1CCV_5	T1CCV_4	T1CCV_3	T1CCV_2	T1CCV_1	T1CCV_0
12-bit PWM	W	T1PDV_11	T1PDV_10	T1PDV_9	T1PDV_8	T1PPV_11	T1PPV_10	T1PPV_9	T1PPV_8
	R	T1PDV_11	T1PDV_10	T1PDV_9	T1PDV_8	T1PPV_11	T1PPV_10	T1PPV_9	T1PPV_8
默认值		0	0	0	0	0	0	0	0

Bit [7:4] **P_TMR1_CountHi** [7:4]: 定时/计数器 1 计数寄存器高字节

16 位定时/计数模式:

写: T1PLV_[15:12] 定时/计数器 1 重载值

数寄存器

读: T1R_[15:12] 定时/计数器 1 计数值

16 位定时/计数模式:

16 位比较模式:

写: T1PLV_[11:8] 定时/计数器 1 重载值

写: T1COV_[15:12] 定时/计数器 1 比较值

读: T1R_[11:8] 定时/计数器 1 计数值

读: T1R_[15:12] 定时/计数器 1 计数值

16 位比较模式:

16 位捕获模式:

写: T1COV_[11:8] 定时/计数器 1 比较值

写: T1PLV_[15:12] 定时/计数器 1 重载值

读: T1R_[11:8] 定时/计数器 1 计数值

读: T1CWV_[15:12] 定时/计数器 1 脉宽捕获值

16 位捕获模式:

8 位捕获模式:

写: T1PLV_[11:8] 定时/计数器 1 重载值

写: T1PLV_[15:12] 定时/计数器 1 重载值的高字节

读: T1CWV_[11:8] 定时/计数器 1 脉宽捕获值

读: T1CCV_[7:4] 定时/计数器 1 周期捕获值

8 位捕获模式:

12 位 PWM 模式:

写: T1PLV_[11:8] 定时/计数器 1 重载值

写: T1PDV_[11:8] 定时/计数器 1 的 PWM 占空比高四位

读: T1CWV_[3:0] 定时/计数器 1 周期捕获值

读: T1PDV_[11:8] 定时/计数器 1 的 PWM 占空比高四位

12 位 PWM 模式:

Bit [3:0] **P_TMR1_CountHi** [3:0]: 定时/计数器 1 计

写: T1PPV_[11:8] 定时/计数器 1 PWM 周期值的高 4 位

读: T1PPV_[11:8] 定时/计数器 1 PWM 周期值的高 4 位

(3) PWM1 占空比的低字节(P_TMR1_PWMDuty, \$0017)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	T1PDV_7	T1PDV_6	T1PDV_5	T1PDV_4	T1PDV_3	T1PDV_2	T1PDV_1	T1PDV_0
读 / 写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
默认值	0	0	0	0	0	0	0	0

Bit [7:0] **P_TMR1_PWMDuty** [7:0]: PWM1 占空比的低字节

12 位 PWM 模式:

写: T1PDV_[7:0] 定时/计数器 1 PWM 占空比的低字节

读: T1PDV_[7:0] 定时/计数器 1 PWM 占空比的低字节

[例] 5.8.4 设定定时/计数器 1 为 16 位定时方式, 定时 10ms

```
lda #253 ; 启动 Timer1 之前, 首先设置计数初值 (重载值)
```

```

sta P_TMR1_PreloadHi      ; 设置高字节初始值
lda #143
sta P_TMR1_Preload       ; 设置地字节初始值
lda #C_T1FCS_Div_128     ; 设定 Timer1 时钟频率为 Fsys/128
sta P_TMR0_1_Ctrl1
lda #C_T116B_Timer       ; 将 Timer1 设置为 16 位定时/计数器模式
sta P_TMR0_1_Ctrl0
lda #253                 ; 首先设定 Timer1 计数重载值高字节为 2x 256= 512
sta P_TMR1_PreloadHi
ld #143                 ; 然后设定 Timer1 计数重载值低字节为 256-143= 113
sta P_TMR1_Preload       ; 设定 Fsys(8Mhz)/128/625= 100Hz(10ms)

```

5.8.4 定时/计数器 2

定时/计数器 2 与定时/计数器 0 功能完全相同，因此关于定时/计数器 2 的控制，请参见定时/计数器 0 章节中对控制寄存器的相关描述。

下面介绍定时/计数器 2 的特点：

- 可读可写
- 时钟频率可以进行 8 级预分频选择
- 可选外部或内部时钟源
- 可选 8 位/16 位工作方式
- 8 位工作方式下，计数从#\$FF 增到#\$00 时溢出并产生溢出中断；16 位工作方式下，计数从#\$FFFF 增到#\$0000 时溢出并产生溢出中断
- 支持 8 位/16 位捕获功能
- 支持 8 位/16 位比较功能
- 支持 8 位 PWM 输出功能

(1) 定时/计数器 2-3 控制器 0 (P_TMR2_3_Ctrl0, \$0018)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	-	T3FC2	T3FC1	T3FC0	-	T2FC2	T2FC1	T2FC0
读 / 写	-	R/W	R/W	R/W	-	R/W	R/W	R/W
默认值	0	0	0	0	0	0	0	0

Bit 7 保留

001 = 8 位定时/计数器

Bit [6:4] T3FC [2:0]: 定时/计数器 3 功能设置位

000 = 禁止

111 = 12 位 PWM

Bit 3 保留

110 = 16 位脉宽捕获

Bit [2:0] T2FC[2: 0]: 定时/计数器 2 功能设置位

101 = 16 位比较器

111 = 8 位 PWM

100 = 16 位定时/计数器

110 = 16 脉宽捕获

011 = 8 位脉宽/周期捕获

101 = 16 位比较器

010 = 8 位比较器

100 = 16 位定时/计数器

011 = 8 位脉宽/周期捕获

000 = 禁止

010 = 8 位比较器

001 = 8 位定时/计数器

(2) 定时/计数器 2-3 控制寄存器 1 (P_TMR2_3_Ctrl1, \$0019)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	-	T3PSC2	T3PSC1	T3PSC0	-	T2PSC2	T2PSC1	T2PSC0
读 / 写	-	R/W	R/W	R/W	-	R/W	R/W	R/W
默认值	0	0	0	0	0	0	0	0

Bit 7 保留

Bit 3 保留

Bit 7 保留

Bit [2:0]

T2PSC[2: 0]: 定时/计数器 2 预分频设置位

Bit [6:4] **T3PSC[2: 0]:** 定时/计数器 3 预分频设置位

111 = 外部时钟频率输入

111 = 外部时钟频率输入

110 = $F_{SYS} \div 512$

110 = $F_{SYS} \div 512$

101 = $F_{SYS} \div 128$

101 = $F_{SYS} \div 128$

100 = $F_{SYS} \div 32$

100 = $F_{SYS} \div 32$

011 = $F_{SYS} \div 8$

011 = $F_{SYS} \div 8$

010 = $F_{SYS} \div 4$

010 = $F_{SYS} \div 4$

001 = $F_{SYS} \div 2$

001 = $F_{SYS} \div 2$

000 = F_{SYS}

000 = F_{SYS}

F_{SYS} : 系统时钟频率

(3) 定时/计数器 2 计数寄存器(P_TMR2_Count, \$001A)

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
8/16-bit Timer	W	T2PLV_7	T2PLV_6	T21PLV_5	T2PLV_4	T2PLV_3	T2PLV_2	T2PLV_1	T2PLV_0
	R	T2R_7	T2R_6	T2R_5	T2R_4	T2R_3	T2R_2	T2R_1	T2R_0
8/16-bit Compare	W	T2COV_7	T2COV_6	T2COV_5	T2COV_4	T2COV_3	T2COV_2	T2COV_1	T2COV_0
	R	T2R_7	T2R_6	T2R_5	T2R_4	T2R_3	T2R_2	T2R_1	T2R_0
8/16-bit Capture	W	T2PLV_7	T2PLV_6	T2PLV_5	T2PLV_4	T2PLV_3	T2PLV_2	T2PLV_1	T2PLV_0
	R	T2CWV_7	T2CWV_6	T2CWV_5	T2CWV_4	T2CWV_3	T2CWV_2	T2CWV_1	T2CWV_0
8-bit PWM	W	T2PPV_7	T2PPV_6	T2PPV_5	T2PPV_4	T2PPV_3	T2PPV_2	T2PPV_1	T2PPV_0
	R	T2PPV_7	T2PPV_6	T2PPV_5	T2PPV_4	T2PPV_3	T2PPV_2	T2PPV_1	T2PPV_0
默认值	0	0	0	0	0	0	0	0	

Bit [7:0] **P_TMR2_Count** [7:0]: 定时/计数器 2 计数寄存器

8 位定时/计数模式:

写: T2PLV_[7:0] 定时/计数器 2 重载值

读: T2R_[7:0] 定时/计数器 2 计数值

8 位比较模式:

写: T2COV_[7:0] 定时/计数器 2 比较值

读: T2R_[7:0] 定时/计数器 2 计数值

8 位捕获模式:

写: T2PLV_[7:0] 8 位捕获器 2 的重载值

读: T2CWV_[7:0] 8 位捕获器 2 的捕获脉宽值

8 位 PWM 模式:

写: T2PPV_[7:0] 8 位 PWM 2 的周期值

读: T2PPV_[7:0] 8 位 PWM 2 的周期值

(4) 定时/计数器 2 计数寄存器高字节(P_TMR2_CountHi, \$001B)

Bit		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
16-bit Timer	W	T2PLV_15	T2PLV_14	T2PLV_13	T2PLV_12	T2PLV_11	T2PLV_10	T2PLV_9	T2PLV_8
	R	T2R_15	T2R_14	T2R_13	T2R_12	T2R_11	T2R_10	T2R_9	T2R_8
16-bit Compare	W	T2COV_15	T2COV_14	T2COV_13	T2COV_12	T2COV_11	T2COV_10	T2COV_9	T2COV_8
	R	T2R_15	T2R_14	T2R_13	T2R_12	T2R_11	T2R_10	T2R_9	T2R_8
16-bit Capture	W	T2PLV_15	T2PLV_14	T2PLV_13	T2PLV_12	T2PLV_11	T2PLV_10	T2PLV_9	T2PLV_8
	R	T2CWV_15	T2CWV_14	T2CWV_13	T2CWV_12	T2CWV_11	T2CWV_10	T2CWV_9	T2CWV_8
8-bit Capture	W	T2PLV_15	T2PLV_14	T2PLV_13	T2PLV_12	T2PLV_11	T2PLV_10	T2PLV_9	T2PLV_8
	R	T2CCV_7	T2CCV_6	T2CCV_5	T2CCV_4	T2CCV_3	T2CCV_2	T2CCV_1	T2CCV_0
8-bit PWM	W	T2PDV_7	T2PDV_6	T2PDV_5	T2PDV_4	T2PDV_3	T2PDV_2	T2PDV_1	T2PDV_0
	R	T2PDV_7	T2PDV_6	T2PDV_5	T2PDV_4	T2PDV_3	T2PDV_2	T2PDV_1	T2PDV_0
默认值		0	0	0	0	0	0	0	0

Bit [7:0] **P_TMR1_CountHi** [7:0]: 定时/计数器 2 计数寄存器高字节

16 位定时/计数模式:

写: T2PLV_[15:8] 定时/计数器 2 重载值

读: T2R_[15:8] 定时/计数器 2 计数值

16 位比较模式:

写: T2COV_[15:8] 定时/计数器 2 比较值

读: T2R_[15:8] 定时/计数器 2 计数值

16 位捕获模式:

写: T2PLV_[15:8] 定时/计数器 2 重载值

读: T2CWV_[15:8] 定时/计数器 2 脉宽捕获值

8 位捕获模式:

写: T2PLV_[15:8] 定时/计数器 1 重载值的高字节

读: T2CCV_[7:0] 定时/计数器 1 周期捕获值

8 位 PWM 模式:

写: T2PDV_[7:0] 定时/计数器 2 的 PWM 占空比

读: T2PDV_[7:0] 定时/计数器 2 的 PWM 占空比

5.8.5 定时/计数器 3

定时/计数器 3 与定时/计数器 0 功能完全相同, 因此关于定时/计数器 3 的控制, 请参见定时/计数器 0 章节中对控制寄存器的相关描述。

下面介绍定时/计数器 3 的特点:

- 可读可写
- 时钟频率可以进行 8 级预分频选择
- 可选外部或内部时钟源
- 可选 8 位/16 位工作方式
- 8 位工作方式下, 计数从#\$FF 增到#\$00 时溢出并产生溢出中断; 16 位工作方式下, 计数从#\$FFFF 增到#\$0000 时溢出并产生溢出中断
- 支持 8 位/16 位捕获功能
- 支持 8 位/16 位比较功能
- 支持 12 位 PWM 输出功能

(1) 定时/计数器 3 计数寄存器低字节(P_TMR3_Count, \$001C)

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
8/16-bit Timer	W	T3PLV_7	T3PLV_6	T3PLV_5	T3PLV_4	T3PLV_3	T3PLV_2	T3PLV_1	T3PLV_0
	R	T3R_7	T3R_6	T3R_5	T3R_4	T3R_3	T3R_2	T3R_1	T3R_0
8/16-bit Compare	W	T3COV_7	T3COV_6	T3COV_5	T3COV_4	T3COV_3	T3COV_2	T3COV_1	T3COV_0
	R	T3R_7	T3R_6	T3R_5	T3R_4	T3R_3	T3R_2	T3R_1	T3R_0
8/16-bit Capture	W	T3PLV_7	T3PLV_6	T3PLV_5	T3PLV_4	T3PLV_3	T3PLV_2	T3PLV_1	T3PLV_0
	R	T3CWV_7	T3CWV_6	T3CWV_5	T3CWV_4	T3CWV_3	T3CWV_2	T3CWV_1	T3CWV_0
12-bit PWM	W	T3PPV_7	T3PPV_6	T3PPV_5	T3PPV_4	T3PPV_3	T3PPV_2	T3PPV_1	T3PPV_0
	R	T3PPV_7	T3PPV_6	T3PPV_5	T3PPV_4	T3PPV_3	T3PPV_2	T3PPV_1	T3PPV_0
默认值		0	0	0	0	0	0	0	

P_TMR3_Count [7:0]: 定时/计数器 3 计数寄存器低字节

8 位/16 位定时/计数模式:

写: T3PLV_[7:0] 定时/计数器 3 重载值的低字节

读: T3R_[7:0] 定时/计数器 3 计数值

8 位/16 位比较模式:

写: T3COV_[7:0] 定时/计数器 3 比较值

读: T3R_[7:0] 定时/计数器 3 计数值

8 位/16 位捕获模式:

写: T3PLV_[7:0] 定时/计数器 3 重载值

读: T3CWV_[7:0] 定时/计数器 3 脉宽捕获值

12 位 PWM 模式:

写: T3PPV_[7:0] 定时/计数器 3 PWM 周期的低 8 位

读: T3PPV_[7:0] 定时/计数器 3 PWM 周期的低 8 位

(2) 定时/计数器 3 计数寄存器高字节(P_TMR3_CountHi, \$001D)

Bit		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
16-bit Timer	W	T3PLV_15	T3PLV_14	T3PLV_13	T3PLV_12	T3PLV_11	T3PLV_10	T3PLV_9	T3PLV_8
	R	T3R_15	T3R_14	T3R_13	T3R_12	T3R_11	T3R_10	T3R_9	T3R_8
16-bit Compare	W	T3COV_15	T3COV_14	T3COV_13	T3COV_12	T3COV_11	T3COV_10	T3COV_9	T3COV_8
	R	T3R_15	T3R_14	T3R_13	T3R_12	T3R_11	T3R_10	T3R_9	T3R_8
16-bit Capture	W	T3PLV_15	T3PLV_14	T3PLV_13	T3PLV_12	T3PLV_11	T3PLV_10	T3PLV_9	T3PLV_8
	R	T3CWV_15	T3CWV_14	T3CWV_13	T3CWV_12	T3CWV_11	T3CWV_10	T3CWV_9	T3CWV_8
8-bit Capture	W	T3PLV_15	T3PLV_14	T3PLV_13	T3PLV_12	T3PLV_11	T3PLV_10	T3PLV_9	T3PLV_8
	R	T3CCV_7	T3CCV_6	T3CCV_5	T3CCV_4	T3CCV_3	T3CCV_2	T3CCV_1	T3CCV_0
12-bit PWM	W	T3PDV_11	T3PDV_10	T3PDV_9	T3PDV_8	T3PPV_11	T3PPV_10	T3PPV_9	T3PPV_8
	R	T3PDV_11	T3PDV_10	T3PDV_9	T3PDV_8	T3PPV_11	T3PPV_10	T3PPV_9	T3PPV_8
默认值		0	0	0	0	0	0	0	0

Bit [7:4] **P_TMR3_CountHi** [7:4]: 定时/计数器 3 计数寄存器

Bit [3:0] **P_TMR3_CountHi** [3:0]: 定时/计数器 3 计数寄存器

16 位定时/计数模式:

写: T3PLV_[15:12] 定时/计数器 3 重载值

读: T3R_[15:12] 定时/计数器 3 计数值

16 位比较模式:

写: T3COV_[15:12] 定时/计数器 3 比较值

读: T3R_[15:12] 定时/计数器 3 计数值

16 位捕获模式:

写: T3PLV_[15:12] 定时/计数器 3 重载值

读: T3CWV_[15:12] 定时/计数器 3 脉宽捕获值

8 位捕获模式:

写: T3PLV_[15:12] 定时/计数器 3 重载值

读: T3CCV_[15:12] 定时/计数器 3 周期捕获值

12 位 PWM 模式:

写: T3PDV_[11:8] 定时/计数器 3 的 PWM 占空比高四位

读: T3PDV_[11:8] 定时/计数器 3 的 PWM 占空比高四位

16 位定时/计数模式:

写: T3PLV_[11:8] 定时/计数器 3 重载值

读: T3R_[11:8] 定时/计数器 3 计数值

16 位比较模式:

写: T3COV_[11:8] 定时/计数器 3 比较值

读: T3R_[11:8] 定时/计数器 3 计数值

16 位捕获模式:

写: T3PLV_[11:8] 定时/计数器 3 重载值

读: T3CWV_[11:8] 定时/计数器 3 脉宽捕获值

8 位捕获模式:

写: T3PLV_[11:8] 定时/计数器 3 重载值

读: T3CCV_[11:8] 定时/计数器 3 周期捕获值

12 位 PWM 模式:

写: T3PPV_[11:8] 定时/计数器 3 PWM 周期的高 4 位

读: T3PPV_[11:8] 定时/计数器 3 PWM 周期的高 4 位

(3) PWM3 占空比的低字节(P_TMR3_PWMDuty, \$001E)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	T3PDV_7	T3PDV_6	T3PDV_5	T3PDV_4	T3PDV_3	3PDV_2	T3PDV_1	T3PDV_0
读 / 写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
默认值	0	0	0	0	0	0	0	0

Bit [7:0] P_TMR3_PWMDuty [7:0]: PWM3 占空比的低字节

比的低字节

读: T3PDV_ [7:0] 定时/计数器 3 的 PWM 占空比的低字节

12 位 PWM 模式:

写: T3PDV_ [7:0] 定时/计数器 3 的 PWM 占空

5.9 CCP(Capture/Compare/PWM)功能

SPMC65P2708A/2716A 两款芯片具有强大的 CCP 功能 (捕获、比较、PWM 输出)。由于这两款芯片的定时/计数器有 8 位/16 位两种, 所以 CCP 功能也有所不同。通常, 有 6 种 CCP 功能模式: 8 位 比较模式、16 位比较模式、8 位捕获模式、16 位捕获模式、8 位 PWM 模式和 12 位 PWM 模式。

5.9.2 8 位比较模式

所有定时/计数器 (Timer0, Timer1, Timer2, Timer3) 均支持 8 位比较输出。下面以定时/计数器 0 为例介绍 8 位比较输出功能: 定时/计数器 0 为 8 位比较模式时, 将比较数据写入寄存器 P_TMR0_Comp(\$0013)中。计数器将以设置的比较数据作为计数初始值开始计数, 当计数器发生溢出中断时, PB2 管脚上输出电平翻转。如果定时/计数器 0 的溢出中断使能, 此时会产生溢出中断。然后载入初始值重新开始计数依此不断循环。也就是说, PB2 管脚会一直输出占空比为 50% 的方波 (即比较输出的波形), 直到定时/计数器停止计数, 比较输出频率为计数器溢出频率的一半。关于 8 位比较输出的时序, 见图 5-29。

表 5-11 CCP 模式 – 在各定时/计数器中应用情况

CCP 功能	Timer Resource 定时/计数器
8 位比较	Timer0, Timer1, Timer2, Timer3
16 位比较	Timer0, Timer1, Timer2, Timer3
8 位捕获	Timer0, Timer1, Timer2, Timer3
16 位捕获	Timer0, Timer1, Timer2, Timer3
8 位 PWM	Timer0, Timer2
12 位 PWM	Timer1, Timer3

[例] 5.9.1 比较输出频率计算见公式

$$f_{\text{Comp}} = \frac{F_{\text{sys}}}{\text{Timer_prescaler} \times (256 - \text{Timer_Compare_value})} \times \frac{1}{2}$$

f_{Comp} : 比较输出频率

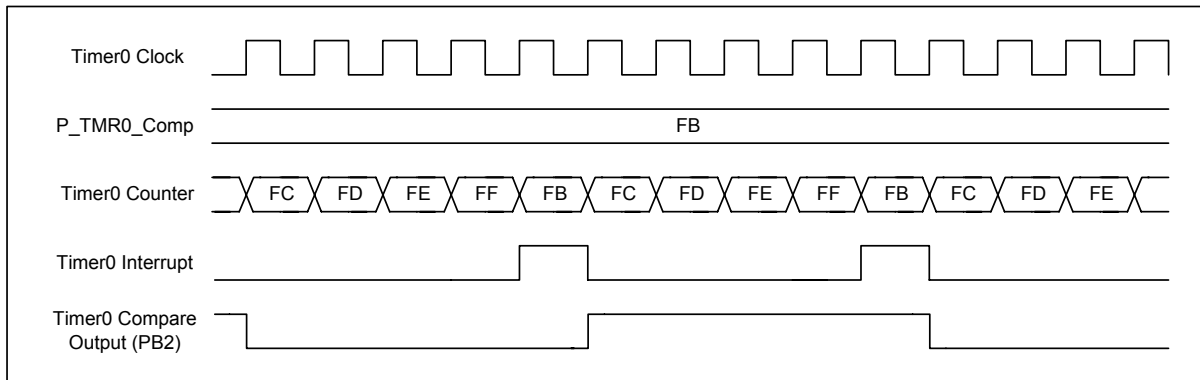


图 5-29 8 位比较输出的时序

5.9.3 16 位比较模式

所有定时/计数器 (Timer0, Timer1, Timer2, Timer3) 均支持 16 位比较输出。下面以定时/计数器 1 为例介绍 16 位比较输出功能：定时/计数器 1 为 16 位比较模式时，将比较数据的低字节、高字节分别写入寄存器 P_TMR1_Comp(\$0015)、P_TMR1_CompHi (\$0016) 中。计数器将以设置的比较数据作为计数初始值开始计数，设置的初始值开始计数，当计数器发生溢出时，PB3

管脚上输出电平翻转。如果定时/计数器 1 的溢出中断使能，此时会产生溢出中断。然后载入初始值重新开始计数，依此不断循环。也就是说，PB3 管脚会一直输出占空比为 50% 的方波（即比较输出的波形），直到定时/计数器停止计数，比较输出频率为计数器溢出频率的一半。关于 16 位比较输出的时序，见图 5-30。

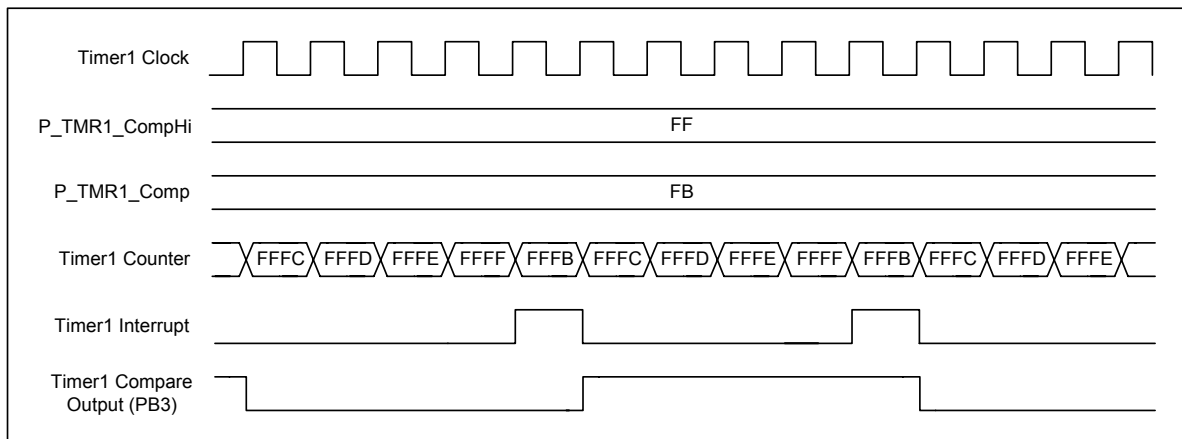


图 5-30 16 位比较输出的时序

[例] 5.9.2 设置定时/计数器 1 为 16 位比较输出模式

```

lda    #253                ;启动定时/计数器 1 之前，首先设置计数初值
sta    P_TMR1_PreloadHi
lda    #143
sta    P_TMR1_Preload      ;设置初值
lda    #C_T1FCS_Div_128    ;设置定时/计数器 1 时钟源为 Fsys/128
    
```

```

sta    P_TMR0_1_Ctrl
lda    #C_T116B_COMP      ;设置定时/计数器 1 为 16 位比较输出方式
sta    P_TMR0_1_Ctrl0

lda    #253                ;首先, 设置定时/计数器 1 计数值高字节为 2x 256= 512
sta    P_TMR1_PreloadHi
lda    #143                ;然后, 设置定时/计数器 1 计数值低字节为 256-143= 113
sta    P_TMR1_Preload     ;PB3 脚输出频率: Fsys(8Mhz)/128/625= 100Hz(10ms)9
    
```

5.9.4 8 位捕获模式

SPMC65P2708A/2716A 两款芯片的所有定时/计数器都支持 8 位捕获模式。8 位捕获模式下，捕获输入管脚必须设置为输入。捕获器计数清除及中断触发的极性通捕获器模块还具有捕获数据保持功能。将 P_CAP_Ctrl.CAPOPT 位设置为 1 就可以进行捕获数据保持，直到将该值读出。然后再进行捕获数据的更新、保持、读出，如此循环。可以利用该功能进行单个脉冲波形的脉宽和周期测量。

【例】5.9.3 定时/计数器 0 的 8 位捕获计算公式

$$\text{Capture Width} = (\text{P_TMR0_Cap} + 1) \times \text{Timer Pre-scale}$$

过捕获控制寄存器(P_CAP_Ctrl, \$0058)来设置。所有的定时 / 计数器均为 16 位的，可以用于捕获脉宽及周期。8 位捕获的工作时序如图 5-31 所示。

【例】5.9.4 定时/计数器 1 的 8 位捕获计算公式

$$\text{Capture Width} = (\text{P_TMR1_Cap} + 1) \times \text{Timer Pre-scale}$$

$$\text{Capture Cycle} = (\text{P_TMR1_CapCycle8} + 1) \times \text{Timer Pre-scale}$$

(1) 捕获控制寄存器(P_CAP_Ctrl, \$0058)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	CAPOPT	-	CAPIP3	CAP1P2	CAPIP1	CAP1P0	CAP1ES	CAP0ES
读 / 写	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
默认值	0	0	0	0	0	0	0	0

Bit 7: CAPOPT: 捕获数据保持选择位
1= 即使接收到新的捕获数据，也不更新，除非对捕获数据寄存器进行读操作。

0=若接收到新数据，立即更新捕获数据

Bit 6: 保留

Bit 5: CAPIP3: 捕获器 3 中断触发极性选择位

1=与捕获器 3 计数触发沿极性相同

0=与捕获器 3 计数触发沿极性相反

Bit 4: CAPIP2: 捕获器 2 中断触发极性选择位

1=与捕获器 2 计数触发沿极性相同

0=与捕获器 2 计数触发沿极性相反

Bit 3: CAPIP1: 捕获器 1 中断触发极性选择位

1=与捕获器 1 计数触发沿极性相同

0=与捕获器 1 计数触发沿极性相反

Bit 2: CAPIP0: 捕获器 0 中断触发极性选择位

1=与捕获器 0 计数触发沿极性相同

0=与捕获器 0 计数触发沿极性相反

Bit 1: **CAP1ES:** 捕获器 1 的计数触发沿极性选择位

1=下降沿触发计数（下降沿清除计数器）

0=上升沿触发计数（上升沿清除计数器）

Bit 0: **CAP0ES:** 捕获器 0 的计数触发沿极性选择位

1=下降沿触发计数（下降沿清除计数器）

0=上升沿触发计数（上升沿清除计数器）

注：捕获器 3 和捕获器 2 分别与 IRQ1、IRQ0 共用管脚，在 IRQ 设置寄存器 1 (\$0034) 中对 CAP3ES、CAP2ES 进行了介绍，详情见 5.5 节。

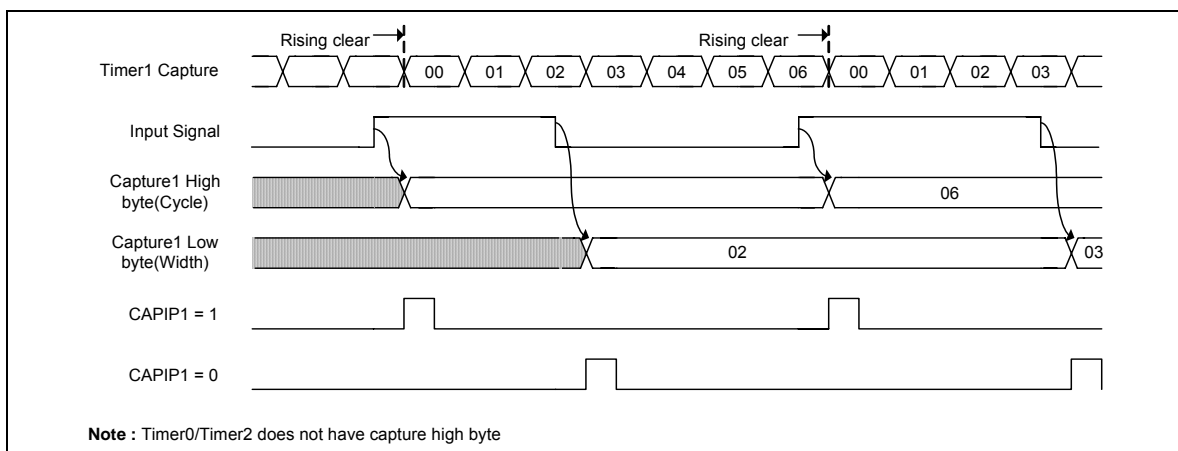
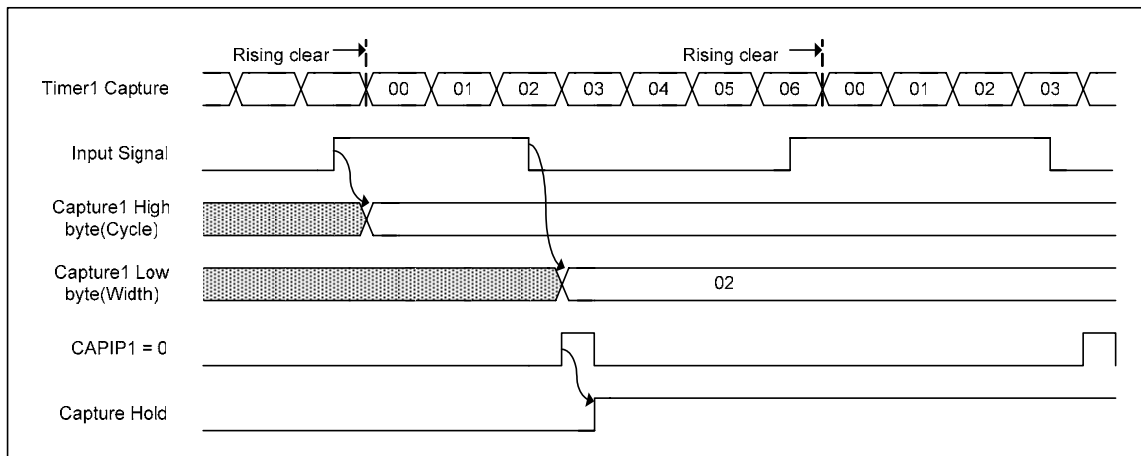


图 5-31 8 位捕获时序



注：Timer0和Timer2无高字节的捕获

图 5-32 8 位捕获及数据保持的时序

【例】5.9.5 设置定时/计数器 0 为 8 位捕获模式

```
lda    #C_T0FCS_Div_512          ; 设置定时/计数器 0 的时钟源为 Fsys/512 (测量范围：
                                   ; 15.6KHz~61Hz)
sta    P_TMR0_1_Ctrl1
```

```

lda    #C_T08B_CAP          ; 设置定时/计数器 0 为 8 位捕获方式
sta    P_TMR0_1_Ctrl0
lda    #(C_CAP_IP0 + C_CAP0_ES) ; 设置为下降沿采数据，同时下降沿产生捕获中断
sta    P_CAP_Ctrl          ; 测量 PB0 管脚输入的低电平脉宽
    
```

5.9.5 16 位捕获模式

SPMC65P2708A/2716A 两款芯片的所有定时/计数器都支持 16 位脉宽捕获输出。16 位脉宽捕获与 8 位脉宽捕获的操作相同。16 位脉宽捕获不能进行周期捕获。

[例] 5.9.6 定时/计数器 1 的 16 位捕获计算公式

$$\text{Capture Width} = ((P_TMR1_CapHi, P_TMR1_Cap) + 1) \times \text{Timer Pre-scale}$$

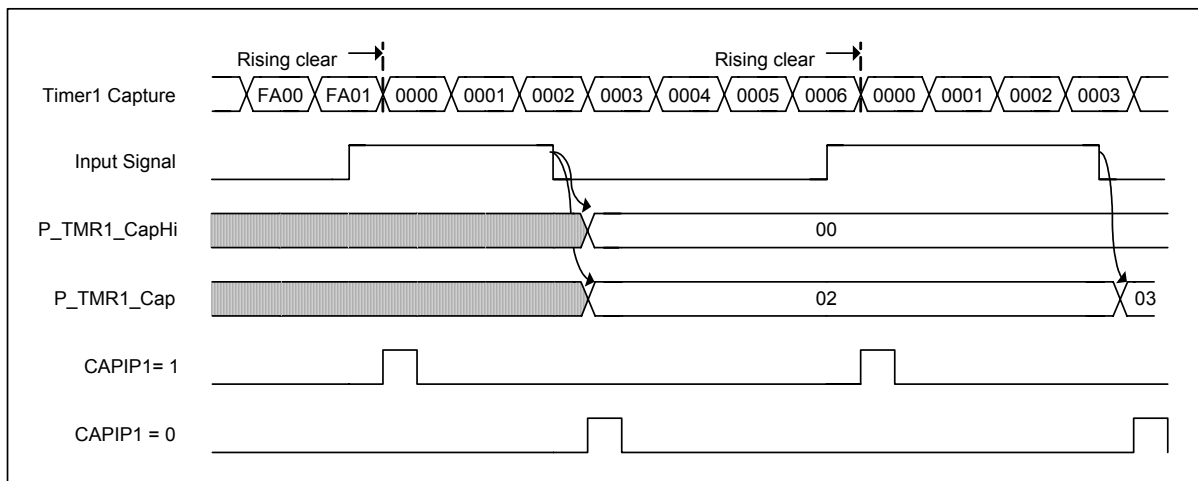


图 5-33 16 位捕获时序

[例] 5.9.7 将定时/计数器 1 设置为 16 位捕获模式

```

lda    #C_T1FCS_Div_128     ; 设定定时/计数器 1 的时钟源为 Fsys/128 (测量范围: 62.5KHz~1Hz)
sta    P_TMR0_1_Ctrl1
lda    #C_T116B_CAP        ; 设置定时/计数器 1 为 16 位捕获方式
sta    P_TMR0_1_Ctrl0
lda    (C_CAP_IP1 + C_CAP1_ES) ; 设置为下降沿采数据，同时下降沿产生捕获中断
sta    P_CAP_Ctrl          ; 测量 PB1 管脚输入的低电平脉宽
    
```

5.9.6 8 位 PWM 模式

SPMC65P2708A/2716A 这两款芯片的定时/计数器 0 和定时/计数器 2 均可设置为 PWM 输出模式。下面

以定时/计数器 0 为例，说明在 PWM 模式下如何设置寄存器。

当定时/计数器 0 被设为 PWM 模式后, PB2 管脚会自动成为 8 位 PWM 输出管脚。PWM 周期在寄存器 P_TMR0_PWM Period [7:0]中设置; 占空比在寄存器

P_TMR0_PWMDuty[7:0]中设置。详细内容请参看后面章节。

5.9.7 12 位 PWM 模式

SPMC65P2708A/2716A 这两款芯片的定时/计数器 1 和定时/计数器 3 均可设置为 PWM 输出模式。下面以定时/计数器 1 为例, 说明在 PWM 模式下如何设置寄存器。

当定时/计数器 1 被设为 PWM 模式后, PB3 管脚会自动成为 12 位 PWM 输出管脚。PWM 周期在寄存器 P_TMR1_DutyPeriod[3:0 和 P_TMR1_PWMPeriod[7:0] 中设置; 占空比在寄存器 P_TMR1_DutyPeriod [7:4]和 P_TMR1_PWMDuty[7:0]中设置。

设置周期值时, 先写高四位的周期值 P_TMR1_DutyPeriod [3:0], 然后再写低 8 位周期值 P_TMR1_PWMPeriod[7:0].; 设置占空比时, 应先设置高四位的占空比值 P_TMR1_DutyPeriod [7:4], 再设置低四位的占空比值 P_TMR1_PWMDuty [7:0]。

下表列出了 PWM 精度与输出频率的关系, 供用户参考。

表 5-12 不同精度和分频系数下 PWM 的几种输出频率

Resolution	PWM 输出频率		
	Pre-Scale= $= F_{sys} \div 1$	Pre-Scale= $= F_{sys} \div 2$	Pre-Scale= $= F_{sys} \div 4$
12 位	1.95KHZ	975HZ	487HZ
11 位	3.9 KHZ	1.95KHZ	975HZ
10 位	7.8 KHZ	3.9 KHZ	1.95KHZ
9 位	15.6 KHZ	7.8 KHZ	3.9 KHZ
8 位	31.25 KHZ	15.63 KHZ	7.8 KHZ

注: 系统时钟频率 F_{sys} 为 8MHz。

【例】5.9.8 定时/计数器 1 的 12 位 PWM 计算公式

$$\text{PWM 周期} = (\text{P_TMR1_DutyPeriod [3:0]} : \text{P_TMR1_PWMPeriod [7:0]}) \times \text{Timer Pre-scale}$$

$$\text{PWM 占空比} = (\text{P_TMR1_DutyPeriod [7:4]} : \text{P_TMR1_PWMDuty [7:0]}) \times \text{Timer Pre-scale}$$

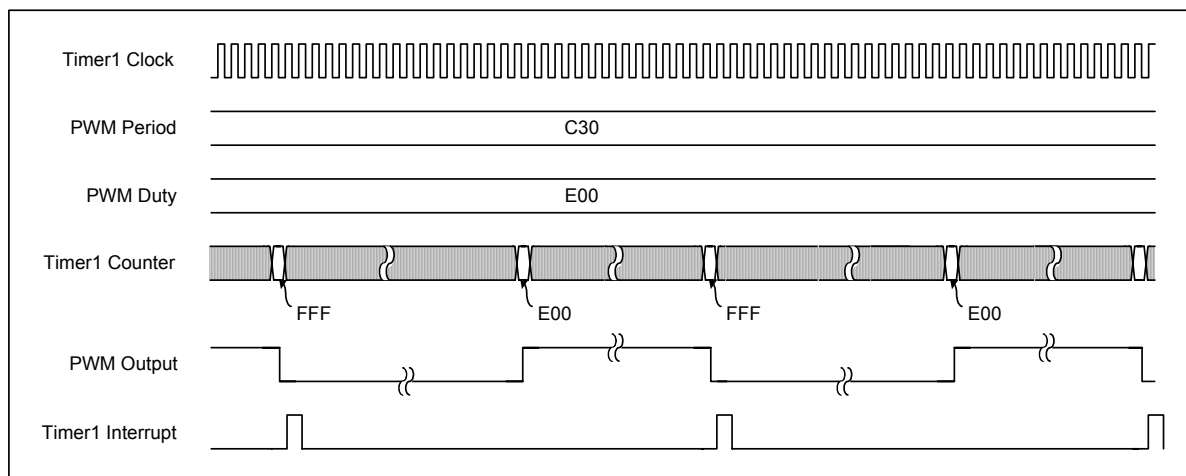


图 5-34 12 位 PWM 模式的工作时序

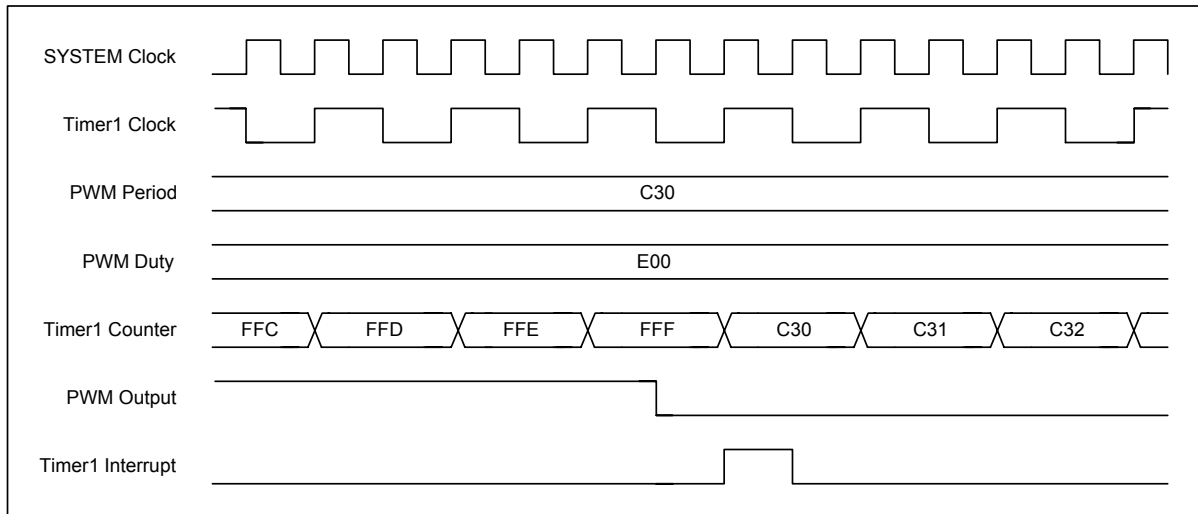


图 5-35 12 位 PWM 模式下计数初值的重载时序

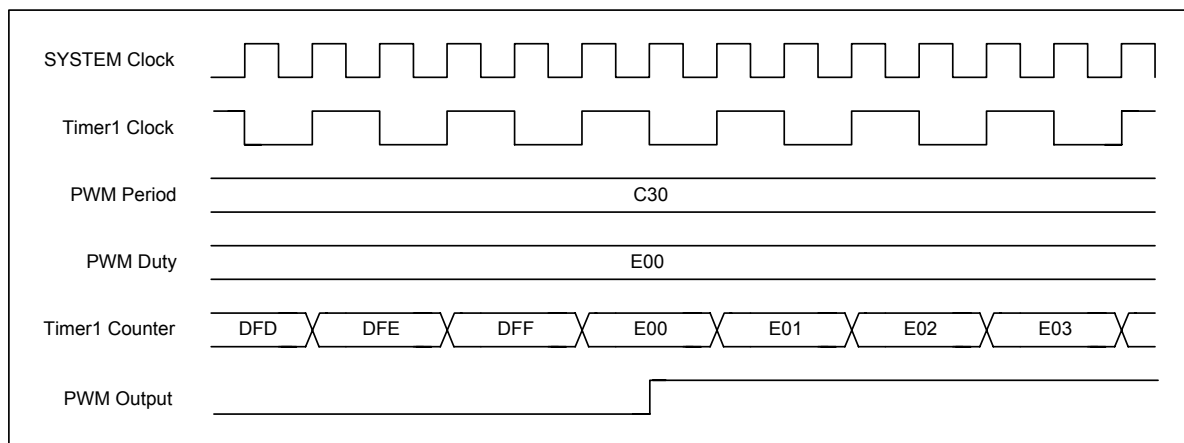


图 5-36 12 位 PWM 输出波形的翻转时序

[例] 5.9.9 将定时/计数器 1 设置为 12 位 PWM 模式，并从 PB3 输出 PWM 波形

```

lda  #$70                ; 启动定时/计数器 1 之前，首先设置计数初值
sta  P_TMR1_DutyPeriod   ; 设置占空比
lda  #$00
sta  P_TMR1_PWMPeriod    ; 设置周期
lda  #$FF
sta  P_TMR1_PWMDuty
lda  #C_T1FCS_Div_8      ; 设置定时/计数器 1 时钟源为 Fsys/8
sta  P_TMR0_1_Ctrl1
lda  #C_T112B_PWM        ; 设置定时/计数器 1 为 12 位 PWM 模式
sta  P_TMR0_1_Ctrl0

lda  #$70
sta  P_TMR1_DutyPeriod   ; PWM 占空比值为$7FF, PWM 周期值为$1000
    
```

```

lda #00
sta P_TMR1_PWMPeriod
lda #0FF
sta P_TMR1_PWMDuty ; 在 PB3 管脚产生 244Hz 50%占空比的 PWM 波形
    
```

5.10 模拟功能

5.10.1 A/D 转换

1. 简述

SPMC65P2708A/2716A 两款芯片均内置了一个 10 位 8 通道 ADC。ADC 可以应用于触摸屏、电池电量检测等很多场合。若应用于语音录制方面，需外置自动增益控制 (AGC)。ADC 输入通道和端口 PA[7:0] 复用，外部参考电压的输入和 PB7 管脚复用。

A/D 转换有三个控制寄存器：P_AD_Ctrl0、P_AD_Ctrl1 和 P_AD_Ctrl2。

寄存器 P_AD_Ctrl0 用于 ADC 模块的各种设置，如 A/D 转换的使能、参考电压选择（通过 P_AD_Ctrl0 的 ADVRT 位选择 ADC 参考电压为 PB7 管脚的外部输入

电压或者内部电源 VDD）、转换时钟频率的选择等。为满足 A/D 硬件速度的需要，A/D 转换的时钟频率应该小于 1.4MHz。

寄存器 P_AD_Ctrl1 负责将端口 PA[7:0] 设置为 A/D 转换的模拟量输入通道。用户在将 I/O 端口作为 A/D 输入通道前，必须先设置该寄存器，否则 A/D 转换值不正确。

P_AD_Ctrl2 用于选择当前的 A/D 转换通道。

当 P_AD_Ctrl0 中的 ADRDY 位被置为“0”时，AD 转换开始。转换完成后，该位被自动置“1”。

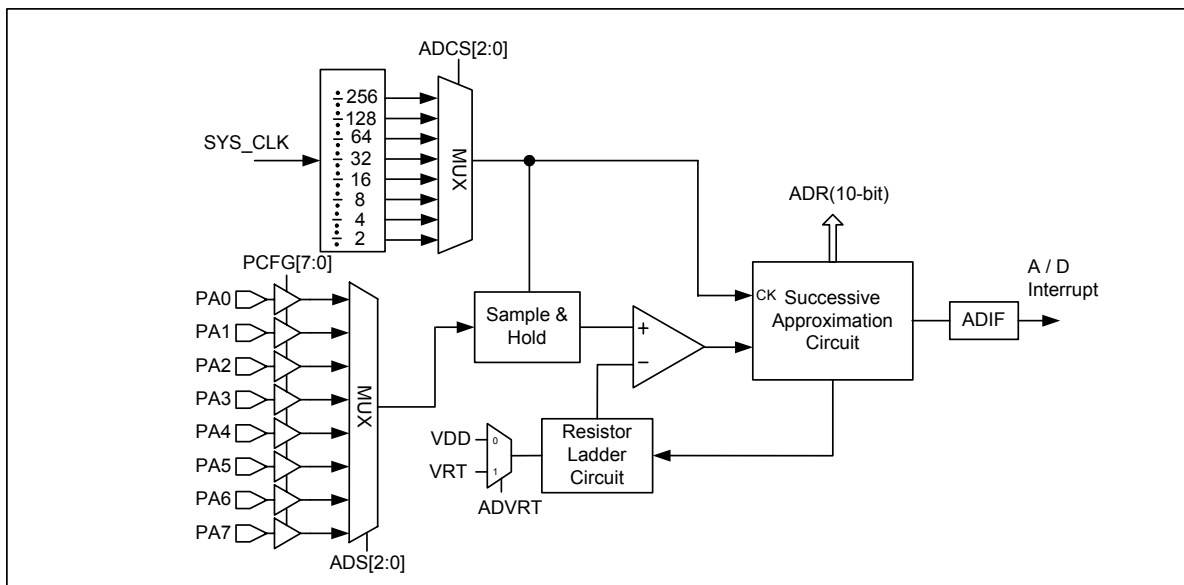


图 5-37 AD 转换原理框图

2. ADC 寄存器

(1) ADC 控制寄存器 0 (P_AD_Ctrl0, \$0028)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	ADEN	ADVRT	0	0	ADCS2	ADCS1	ADCS0	ADRDY/ STARTB
读 / 写	R/W	R/W	-	-	R/W	R/W	R/W	R/W
默认值	0	0	0	0	0	1	1	0

Bit 7 **ADEN**: ADC 使能位

0 = 禁止 AD 转换功能

1 = 允许 AD 转换功能

Bit 6 **ADVRT**: ADC 参考电压选择位

1 = 外部电压(PB7)作为参考电压

0 = 内部电源电压(Vdd)作为参考电压

Bit [5:4] 保留

Bit [3:1] **ADCS**[2:0]: ADC 时钟选择位

111 = $F_{SYS} \div 256$

110 = $F_{SYS} \div 128$

101 = $F_{SYS} \div 64$

100 = $F_{SYS} \div 32$

011 = $F_{SYS} \div 16$

010 = $F_{SYS} \div 8$

001 = $F_{SYS} \div 4$

000 = $F_{SYS} \div 2$

F_{SYS} : 系统时钟频率

Bit 0 **ADRDY/STARTB**: AD 转换状态标志或启动位

读操作: AD 转换的状态

1: AD 转换结束, 等待下次转换开始

0: 正在进行数据转换

写操作: AD 转换启动位

1: 无效

0: 启动 AD 转换

(2) ADC 控制寄存器 1(P_AD_Ctrl1, \$0029)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
读 / 写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
默认值	0	0	0	0	0	0	0	0

Bit [7:0] **PCFG**: ADC 通道设置位

这些位用于将数字端口 A 设置为 A/D 转换的模拟量输入通道。

PCFG7:

1 = 模拟输入(AN7)

0 = 数字输入(PA7)

PCFG6:

1 = 模拟输入(AN6)

0 = 数字输入(PA6)

PCFG5:

1 = 模拟输入(AN5)

0= 数字输入(PA5)

1= 模拟输入(AN2)

PCFG4:

0= 数字输入(PA2)

1= 模拟输入(AN4)

PCFG1:

0= 数字输入(PA4)

1= 模拟输入(AN1)

PCFG3:

0= 数字输入(PA1)

1= 模拟输入(AN3)

PCFG0:

0= 数字输入(PA3)

1= 模拟输入(AN0)

PCFG2:

0= 数字输入(PA0)

(3) ADC 控制寄存器 2 (P_AD_Ctrl2, \$002A)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	ADCE	0	ADS2	ADS1	ADS0	0	0	0
读 / 写	R/W	-	R/W	R/W	R/W	-	-	-
默认值	0	0	0	0	0	0	0	0

Bit 7 **ADCE**: A/D 转换电源控制位。

1= 打开 A/D 转换电源

0= 关闭 A/D 转换电源

010= 选择通道 2 (AN2)

011= 选择通道 3 (AN3)

100= 选择通道 4 (AN4)

Bit 6: 保留

101= 选择通道 5 (AN5)

Bit [5:3] ADC 当前通道选择位

110= 选择通道 6 (AN6)

000= 选择通道 0(AN0)

111= 选择通道 7 (AN7)

001= 选择通道 1 (AN1)

Bit [2:0]: 保留

(4) A/D 转换结果的高 8 位(P_AD_DataHi, \$002B)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_AD_DataHi							
读 / 写	R/W							
默认值	00h							

Bit [7:0] **P_AD_DataHi**: 10 位 AD 转换结果的高 8 位数据。

(5) A/D 转换结果的低 2 位(P_AD_DataLo, \$002C)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_AD_DataLo		-	-	-	-	-	-
读 / 写	R/W		-	-	-	-	-	-
默认值	0	0	-	-	-	-	-	-

Bit [7:6] **P_AD_DataLo**: 10 位 AD 转换结果的低 2 位数据。

Bit [5:0] 保留

3. ADC 转换流程

AD 转换流程如下图所示。

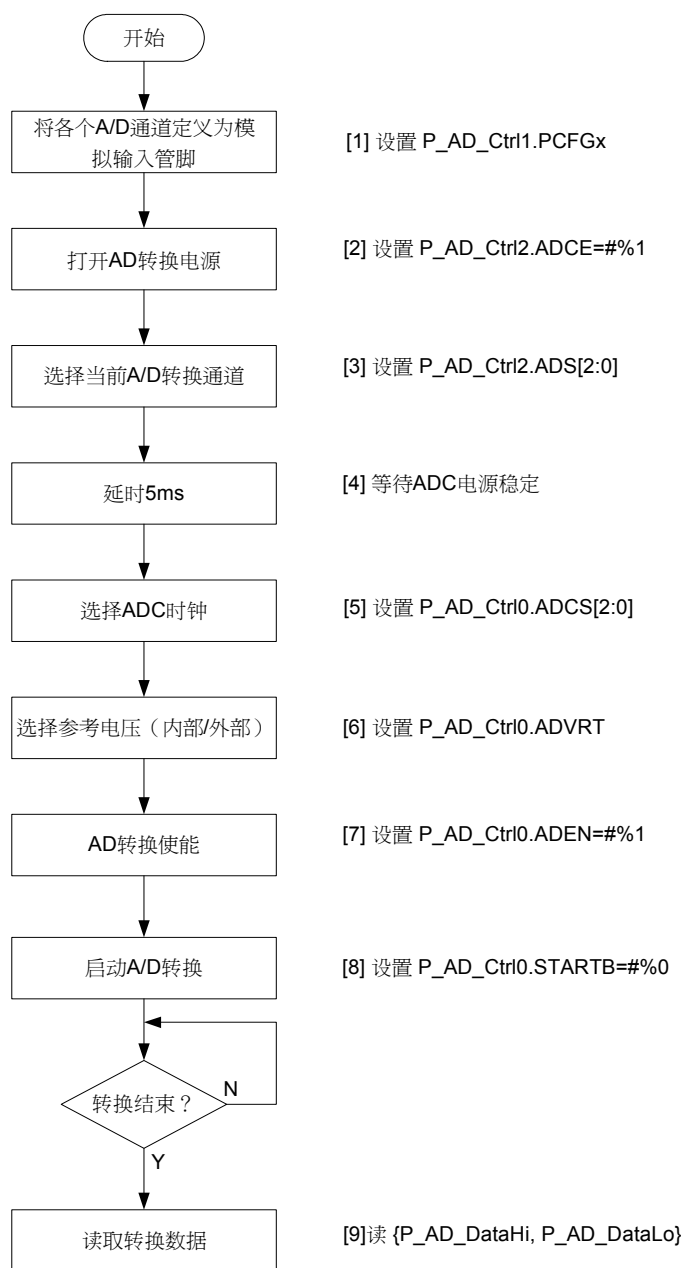


图 5-38 A/D 转换流程

[例] 5.10.1 使能 AD 转换功能

```

lda #C_AD_Pin0                ; 将 PA0 定义为模拟输入口
sta P_AD_Ctrl1
lda #(C_AD_CE+ C_AD_Ch0)      ; 使能 ADC 电源，选择通道 0 ( AN0 )
sta P_AD_Ctrl2                ; 选择通道 0
jsr F_Delay5ms                ; 延时 5.0ms
lda #(C_AD_EN + C_AD_CS_8)
sta P_AD_Ctrl0                ; ADC 使能，ADC 时钟频率 = Fsys / 8
lda P_AD_Ctrl0
and #11111110B                ; 启动转换
sta P_AD_Ctrl0

```

5.11 通信模块

5.11.1 SPI (Serial Peripheral Interface)

1. 简述

SPI(Serial Peripheral Interface)即串行外设接口，它是一种高速同步的串行接口，可以串行接收或发送数据，支持主从机之间的全双工同步传输，传输速率可编程设定。在 SPMC65P2708A 和 SPMC65P2716A 中，SPI 模块有 4 个管脚，支持主模式和从模式。SPI 的相关参数都可编程设定，如运行模式、工作时钟频率、时钟相位和极性等。

SPI 模块具有以下特性：

- 四个接口管脚
 - SDO: 数据输出管脚 (与 PC3 复用)
 - SDI: 数据输入管脚 (与 PC2 复用)
 - SCK: 时钟输入/输出管脚 (与 PC1 复用)
- SSB: 从机片选管脚 (与 PC0 复用)
- 支持全双工同步传输
- 两种工作模式：主模式、从模式
- 波特率：可编程传输速率。CPU 时钟在 8 MHz 时，最大传输速率可达 2Mbps
- 每次发送或接收的数据长度：8 位
- 时钟相位和极性的可编程设定
- 数据采样时刻选择：可在数据输出过程中或数据输出末尾进行采样
- SPI 接收/发送缓冲器大小为 1 个字节

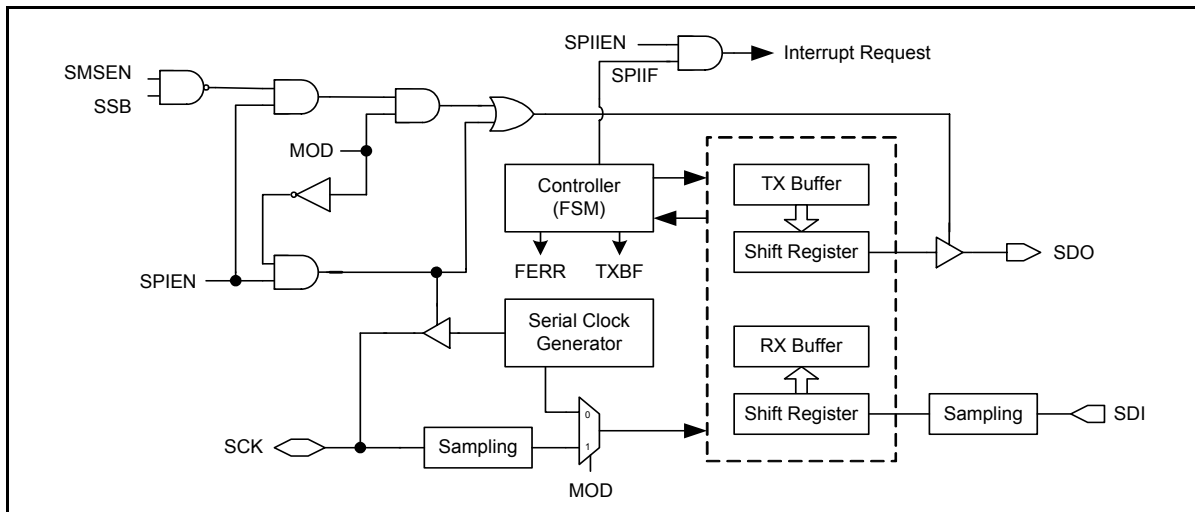


图 5-39 SPI 模块的结构框图

2. SPI 运行模式

主模式下，移位时钟(SCK)由 SPI 模块产生。在寄存器 P_SPI_Ctrl0 中有两个控制位用于控制时钟相位 (SCKPHA) 和极性 (SCKPOL)。当向寄存器 P_SPI_TxData 写入数据时，发送立即开始。另外如果用户想要接收从机数据，SDI 管脚 (PC2)可以设定为悬浮输入。SPMC65P2708A/2716A 支持 SPI 模块的接收和发送中断。在寄存器 P_SPI_Status 中将 SPIOEN 位置“1”可同时使能收/发中断。与其它中断处理一样，在收/发中断服务程序结束后需要清除中断标志位 (SPIIF)。

在向寄存器 P_SPI_TxData 写入一个字节之后，数据被锁存到寄存器的内部发送缓冲中。如果此时移位寄存器是空的，该数据将被载入到移位寄存器中并在下一个 SCLK 相位时开始传输。相反，如果移位寄存器不空，正在进行数据移位 (P_SPI_Status 寄存器中的 TXBF 标志位为“1”)，新数据会等待当前的数据移出后才可以载入进行移位。

SPI 通过 SDO 管脚将移位寄存器中的待发送的数据从最高有效位(MSB)开始移出。8 个 SCK 周期后，8 位发送数据将全部被移出，同时，接收数据也通过 SDI 管脚移入。当每组 8 位数据发送完成后，寄存器 P_SPI_Status 中的 SPIIF 被设置为“1”；此外，如果寄存器 P_SPI_Status 中的 SPIOEN 位被设置为“1”，会产生一个 SPI 发送中断。

同样，当 SPI 接口成功地接收了一组 8 位数据时，接收到的数据将被锁存到接收缓冲器中。此时，寄存器 P_SPI_Status 中的 SPIIF 位被设置为“1”，如果 P_SPI_Status 寄存器中的 SPIOEN 位被设置为“1”，会产生一个 SPI 接收中断。

下图给出了在 SPI 主机模式下，不同设置的运行时序(极性控制位等于“1”或“0”，相位控制位等于“1”或“0”，取样控制位等于“1”或“0”)。

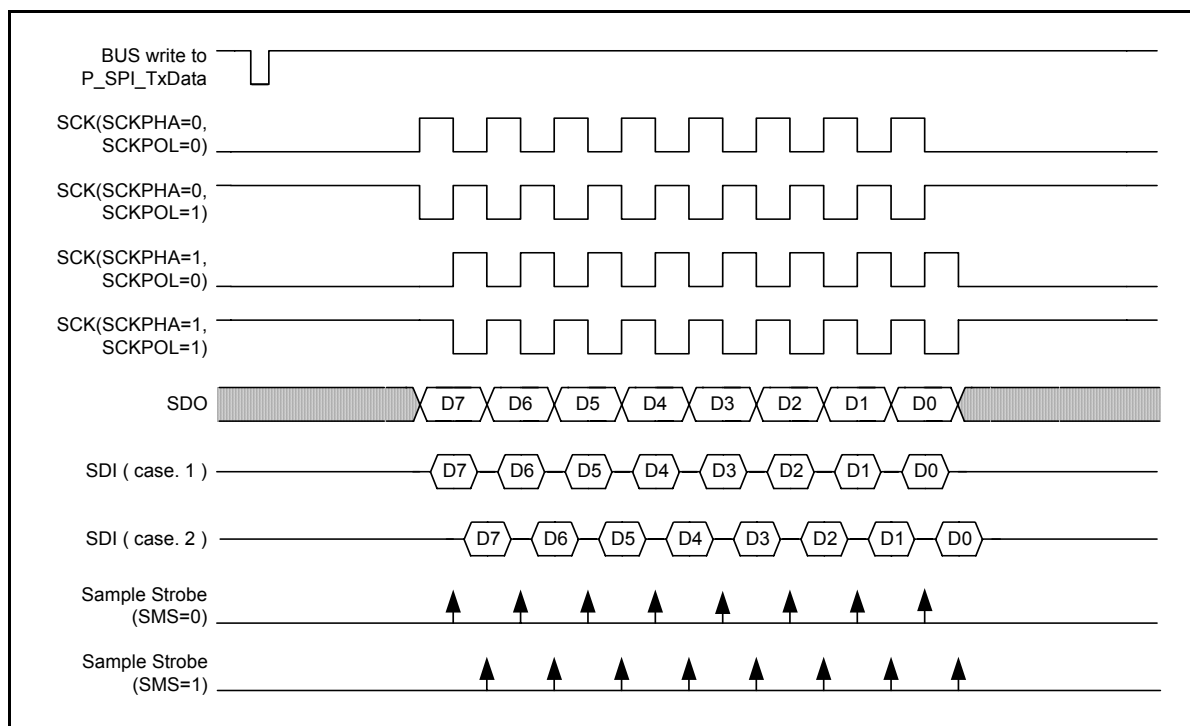


图 5-40 SPI 运行模式时序图（主模式）

3. 从模式

在从模式下，移位时钟 SCK 由主机产生，所以从第一个外部时钟周期开始传输数据。发送前，应在第一个 SCK 周期来临之前将数据写入发送缓冲器。注意，主机与从机都必须设置按相同的 SCK 相位和极性运行，以进行数据的发送与接收。

如果时钟相位 (SCKPHA) 为“1”，只要向寄存器 P_SPI_TxData 写入数据，就开始移出第一个数据位。如果时钟相位(SCKPHA)为“0”，则在第一个 SCK 边沿后才开始移出第一个数据位。

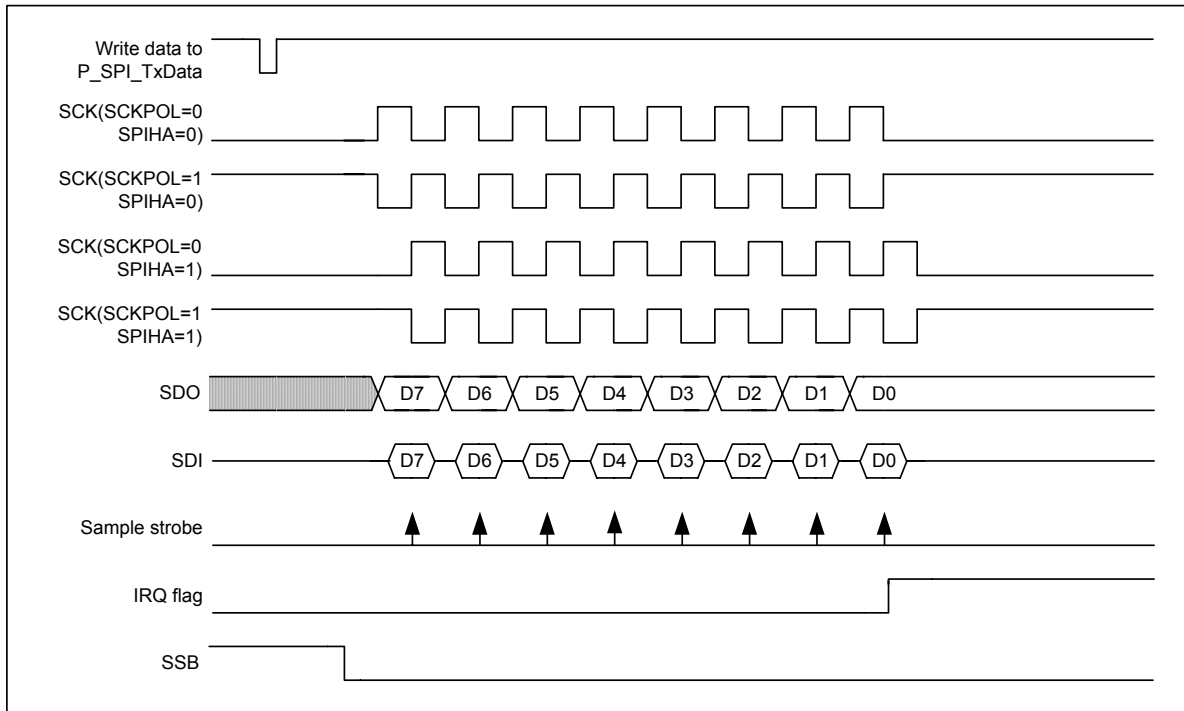


图 5-41 SPI 运行模式时序图（从模式）

4. SPI 寄存器

(1) SPI 控制寄存器 0 (P_SPI_Ctrl0, \$0038)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	SPIEN	MOD	SCKPHA	SCKPOL	SMS	SCKSEL2	SCKSEL1	SCKSEL0
读 / 写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
默认值	0	0	0	0	0	0	0	0

Bit 7 **SPIEN**: SPI 使能位。该位置 1 后 PC[3:0] 即作为 SPI 通讯接口。

1= 使能 SPI 功能

0= 禁止 SPI 功能

Bit 6 **MOD**: SPI 运行模式选择位

1= 从机模式

0= 主机模式

Bit 5 **SCKPHA**: SPI 时钟相位选择位, 参见 SPI 主模式时序。

Bit 4 **SCKPOL**: SPI 时钟极性选择位, 见 SPI 主模式时序。

Bit 3 **SMS**: 主机采样模式选择位

1=在接收数据的输出末尾进行采样

0=在接收数据的输出过程中进行采样

Bit [2:0] **CKSEL [2:0]**: 主模式时钟选择位

111= $FSYS \div 128$

110= $FSYS \div 128$

101= $FSYS \div 128$

100= $FSYS \div 64$

011= $FSYS \div 32$

010= $FSYS \div 16$

001= FSYS ÷ 8

F_{sys}: 系统时钟频率

000= FSYS ÷ 4

(2) SPI 控制寄存器 1 (P_SPI_Ctrl1, \$0039)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	SMSSEN	SWRST	-	-	-	-	SPISPCLK1	SPISPCLK0
读 / 写	R/W	R/W	-	-	-	-	R/W	R/W
默认值	0	0	0	0	0	0	1	0

Bit 7 **SMSSEN**: SPI 从模式选择输入 Bit [1:0] **SPISPCLK** [1:0]: 采样时钟选择位

1= PC0 作为从模式选择的输入管脚, 低有效

11= FSYS ÷ 4

0= PC0 作为通用 IO 管脚

10= FSYS ÷ 2

Bit 6 **SWRST**: SPI 软件复位

01= FSYS

写:

00=不采样

1=产生脉冲复位 SPI 模块(寄存器设置除外)

F_{sys}: 系统时钟频率

0=无效

注: 采样时钟的目的是为了防止在接收数据时尖脉冲的干扰, 但是低的采样时钟会影响通信速度。

读: 总为 0

Bit [5:2] 保留

(3) SPI TX/RX 状态寄存器 (P_SPI_Status, \$003A)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	SPIIF	SPIIEN	TXBF	-	-	-	-	BUFFull
读 / 写	R/W	R/W	R	-	-	-	-	R
默认值	0	0	0	0	0	0	0	0

Bit 7 **SPIIF**: SPI 中断标志位

Bit 6 **SPIIEN**: SPI 中断使能位

读:

1=使能

1=SPI 中断标志为 1

0=禁止

0= SPI 中断标志为 0

Bit 5 **TXBF**: 发送缓冲器满标志位

写:

1= 发送缓冲器满

1= 清除标志

0= 发送缓冲器空

0=无效

Bit [4:1] 保留

Bit 0 **BUFFull**: 缓冲器满并覆盖

0= 缓冲器工作正常

1= 覆盖

(4) SPI 发送缓冲器 0 (P_SPI_TxData, \$003B)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	SPITXDATA 7	SPITXDATA 6	SPITXDATA 5	SPITXDATA 4	SPITXDATA 3	SPITXDATA 2	SPITXDATA 1	SPITXDATA 0
读 / 写	W	W	W	W	W	W	W	W
默认值	0	0	0	0	0	0	0	0

Bit [7:0] **SPITXDATA**: SPI 发送数据

读: 总为#0

写: 发送数据

(5) SPI 接收缓冲器 0 (P_SPI_RxData, \$003C)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	SPIRXDAT A7	SPIRXDAT A6	SPIRXDAT A5	SPIRXDAT A4	SPIRXDATA3	SPIRXDAT A2	SPIRXDAT A1	SPIRXDAT A0
读 / 写	R	R	R	R	R	R	R	R
默认值	0	0	0	0	0	0	0	0

Bit [7:0] **SPIRXDATA**: SPI 接收数据

读: SPI 接收数据

写: 无效果

[例] 5.11.1 SPI 采样时钟频率计算公式和 IO 设置

SPI 采样时钟频率计算公式

用下面的公式计算采样时钟，以保证获得正确的数据。

Sampling clock > 4 x SPI clock (主模式)

System clock > 4 x SPI clock x SPISPCLK (从模式)

例：

SPMC65P2708A 作为主机

SPI 时钟频率=1MHz (PC1)，FSYS=8MHz

采样时钟频率>4x1MHz = 4MHz

采样时钟频率必须大于 4MHz 才能保证数据接收正确。因此，采样时钟选择位 (SPISPCLK [1:0]) 宜设置为 1

采样时钟频率=8MHz/1=8MHz

SPI 运行在主模式下端口 C 的设置

PC0(SS): 输出

PC1(SCK): 输出

PC2(SDI): 悬浮输入

PC3(SDO): 输出

SPI 运行在从模式下端口 C 的设置

PC0(SS): 悬浮输入

PC1(SCK): 悬浮输入

PC2(SDI): 悬浮输入

PC3(SDO): 悬浮输入

【例】5.11.2 将 SPI 初始化为主机模式

```

lda    #00000100B                ; PC2 设置为悬浮的输入作为 SDI
sta    P_IOC_Attrib
lda    #(C_SPI_INTIF+C_SPI_INTEN) ; 使能 SPI 中断，清除 SPI 中断标志
sta    P_SPI_Status
lda    #(C_SPI_SWRST+C_SPI_SPC_Div_4) ; 软件复位 SPI，采样时钟频率=Fssys/4 (must >4* SPI
                                        clock)
sta    P_SPI_Ctrl1
lda    #(C_SPI_EN+C_SPI_CS_Div_128) ; 使能 SPI，主机模式，SPI 时钟频率
                                        =Fssys (8MHz) / 128 = 62.5KHz
sta    P_SPI_Ctrl0
    
```

【例】5.11.3 SPI 主机模式下发送数据

```

L_TestSPIL7:
set    P_SPI_Status, CB_SPI_INTIF ; 清除 SPI 中断标志
lda    #$55
sta    P_SPI_TxData                ; 发送数据
L_TestSPIL71:
lda    P_SPI_Status
and    #C_SPI_INTIF                ; 发送结束？
beq    L_TestSPIL71                ; 否，继续发送
    
```

【例】5.11.4 SPI 主机模式下接收数据

```

L_TestSPIL5:
lda    P_SPI_Status
and    #C_SPI_INTIF                ; 产生 SPI 发送中断？
beq    L_TestSPIL5                ; 否，继续检测
nop
lda    P_SPI_RxData                ; 读取数据
sta    G_MWorkReg1
    
```

[例] 5.11.5 将 SPI 初始化为从机模式

```

lda    #00001111B                ; 设 PC0~3 为悬浮输入
sta    P_IOC_Attrib
lda    #(C_SPI_EN+C_SPI_MOD)      ; 使能 SPI, 从机模式, SPI 时钟频率
                                     =Fsys(8MHz)/128= 62.5KHz
sta    P_SPI_Ctrl0
lda    #(C_SPI_SSMSEN+C_SPI_SWRST+C_SPI_SPC_Div_4) ; 软件复位 SPI, 采样时钟频率
                                     =Fsys/4(sample clock >4* SPI clock)
sta    P_SPI_Ctrl1
lda    #(C_SPI_INTIF+C_SPI_INTEN) ; 使能 SPI 中断, 清除 SPI 中断标志
sta    P_SPI_Status
    
```

5.11.2 UART (Universal Asynchronous Receiver/Transceiver)

1. 简述

SPMC65P2708A/2716A 芯片内置了 UART 模块, 用于 CPU 和外围设备进行异步通讯。通讯格式为标准的非归零格式(NRZ)。UART 的接收器和发送器均为双缓冲结构, 可以分别使能、设置中断, 可同时或分别在全双工模式下工作。UART 的波特率的范围为 2400 ~ 38400 bps。用户可以通过一个 8 位的波特率选择寄存器来选择波特率。

为确保传输的数据正确, UART 模块设有一个状态寄存器。用户可以通过读状态寄存器中的标志位(奇偶校验位、溢出错误标志位、帧校验位)来检查数据传输是否正确。奇偶校验可检测传输前后的数据的奇偶是否匹配。帧校验用于检测数据位置是否正确, 即停止位是否丢失。如果传输速度太快, 接收到的数据不能在下一组数据到来前被及时读出, 从而造成数据丢失, 发生溢出错误。

UART 模块特点如下:

- 两个接口管脚
 - RXD: 数据接收管脚 (与 PC5 复用)
 - TXD: 数据发送管脚 (与 PC4 复用)
- 提供标准的异步全双工通讯

- 可编程的收发波特率
- 可进行偶校验、奇校验或禁止校验
- 停止位宽度可设置为 1 位或 2 位
- 支持发送中断
- 支持接收中断
- 高的抗噪声能力的接收(接收中对数据连续进行 3 次采样, 并对结果进行多数决策)
- 在接收中进行帧校验和奇偶校验
- 溢出侦测
- CPU 工作频率为 8MHz 时, 波特率可在 2400 bps~38400 bps@ 8MHz 之间编程设定

UART 内部结构见图 5-42, 数据格式图 5-43, 数据采样方式见图 5-45。

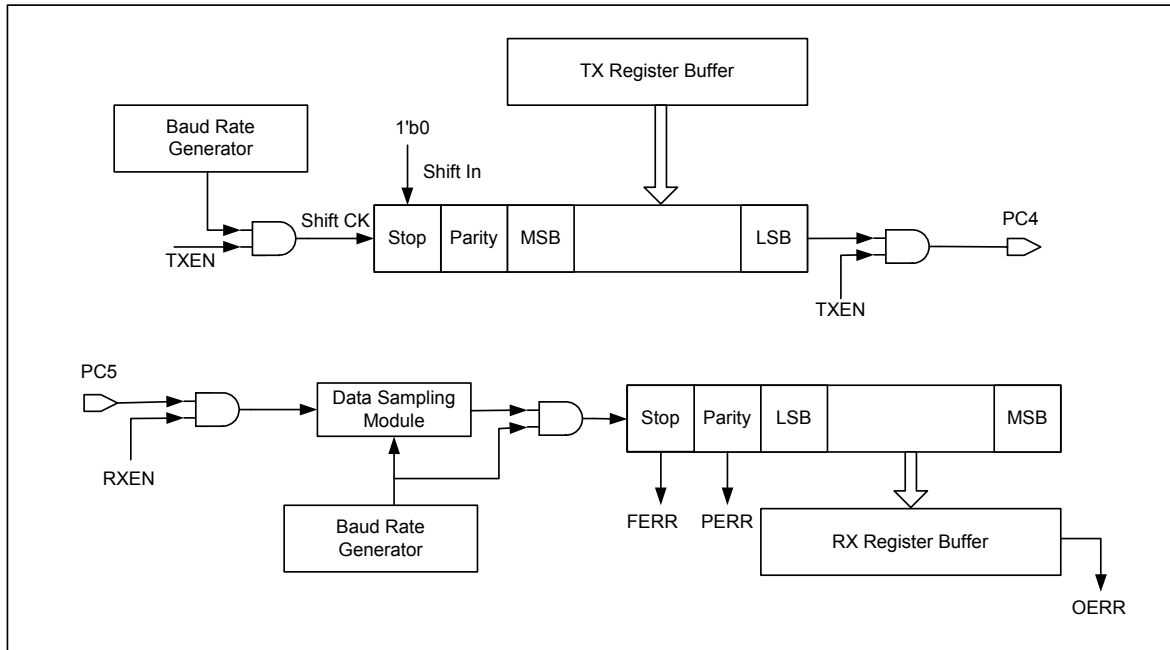


图 5-42 UART 内部结构图

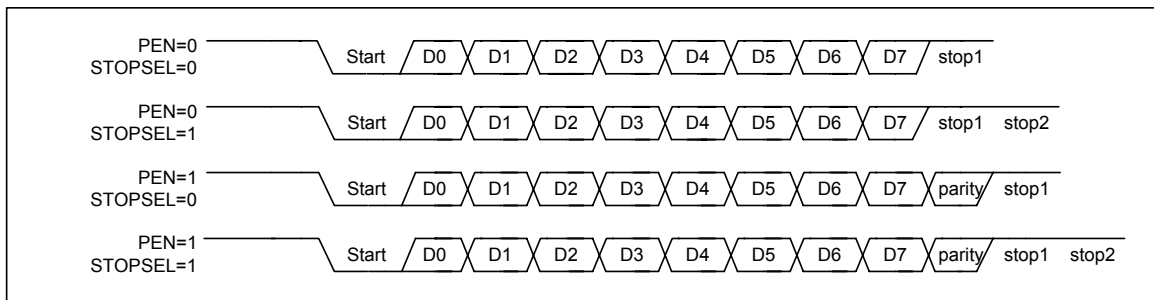


图 5-43 UART 的数据格式

2. UART 操作

波特率是由一个波特率寄存器和一个 8 位定时/计数产生的。每次定时/计数器计到最大计数值(255)后,再加 1 时,一个时钟信号就会被发送到波特率产生电路,在该电路中,时钟信号会通过一个 16 分频的计数器,然后产生波特率。定时/计数器溢出后,会自动重新载入波特率寄存器中的值。

$$\text{波特率} = F_{\text{sys}} / [16 \times (256 - P_UART_Baud)]$$

波特率寄存器中的内容为 8 位的无符号数。使用下面的公式可从一个已知的波特率导出所需的波特率寄存器

值:

$$P_UART_Baud = 256 - F_{\text{sys}} / (16 \times \text{Baud Rate})$$

在向寄存器 P_UART_Data 写入数据后, UART 开始发送。UART 在 TXD 引脚上发送数据按照以下顺序:起始位、8 位数据(低位在前)、奇偶校验位(奇偶校验使能的情况下有效)、停止位。发送停止位后再经过两个 CPUCLK 周期,寄存器 P_UART_Status 中的 TXIF 位被置 1。

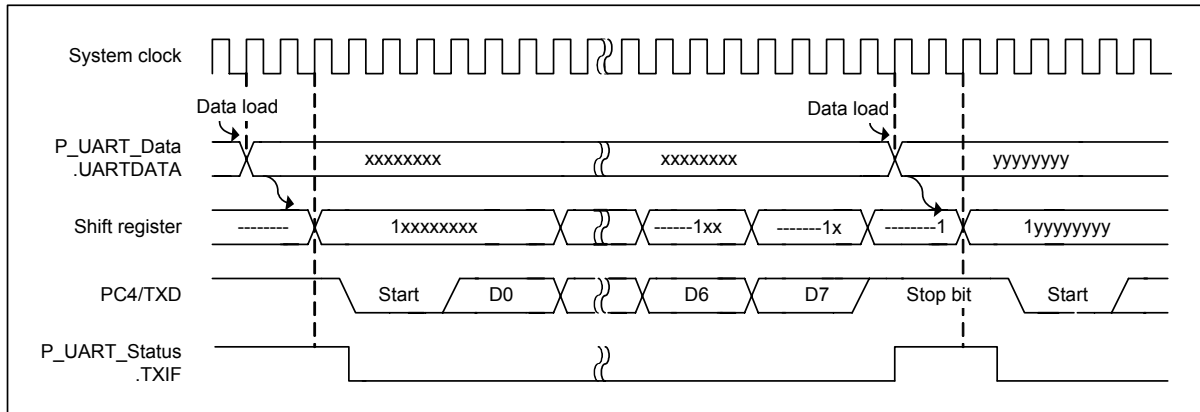


图 5-44 数据发送示意图

当寄存器 P_UART_Ctrl 中的 RXEN 位使能接收功能后，如果 RXD 引脚接收到一个开始位的下降沿，标志着接收数据开始。在任何波特率下 RXD 的每一位都要采样 16 次。当检测到开始位的下降沿时，产生波特率的 16 分频计数器会被复位，复位后与接收位边沿对齐，开始接收数据。

为抑制噪声，串口通过对每位接收期间进行 3 次连续采样，并通过多数决议来确定每个接收位的值，此方法尤其适用于起始位。如果 RXD 上的下降沿没有通过连续 3 次采样的逻辑低电平多数决议校验，串口则停止接收并等待 RXD 管脚上下一个下降沿的产生。

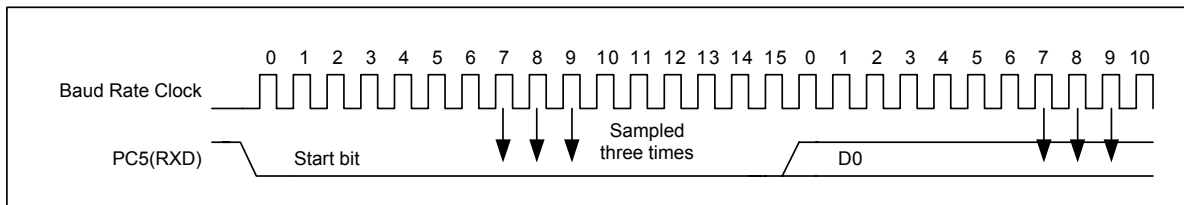


图 5-45 UART 的数据采样方式

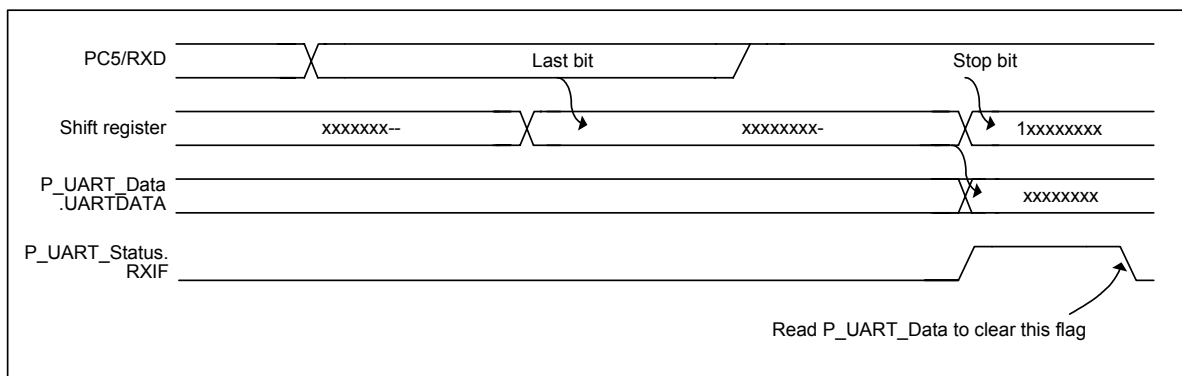


图 5-46 数据接收完毕示意图

接收到停止位后，UART 模块便向寄存器 P_UART_Data 中写入接收到的数据，并且将 RXIF 和 RXBF 置 1。然后，RXD 脚开始等待下一个下降沿。在下次接收完成后，如果上次接收的数据还没有被读取，那么将会被新数据覆盖。在每次接收到停止位后，CPU

会检查 RXBF 标志，如果 RXBF 标志为 1，OE 标志将会被置 1，表示发生越界错误。如果在下次接收期间，检查到 RXBF 标志为 0，OE 标志将会被清除。

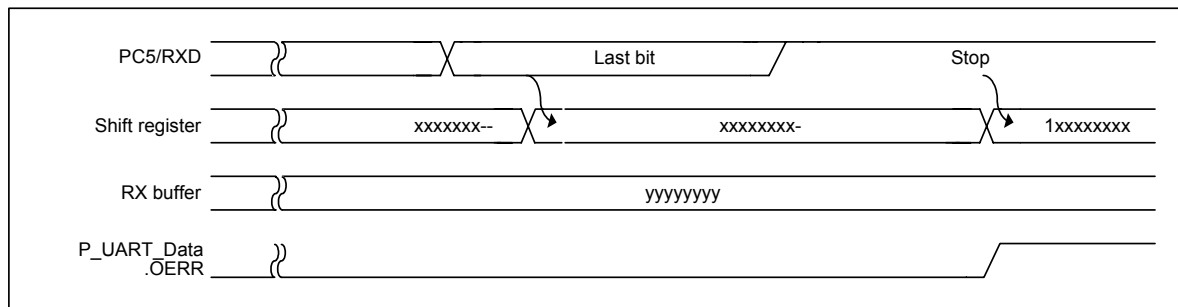


图 5-47 发生越界错误时序

UART 具有奇偶校验和帧检测功能，这些功能有利于提高接收数据的可靠性。通过 UART 控制寄存器 P_UART_Ctrl 中的 PSEL 项，可以将校验功能设定为奇校验或偶校验。如果奇偶校验功能被允许，当接收到数据帧中的奇偶校验位后，接收器进行奇偶校验。若校验的结果错误，UART 状态寄存器 P_UART_Status 中的 PE 被置 1，如图 5-48 所示。

停止位是 UART 数据格式中的其中一位，若在数据传输过程中接收器没有接收到停止位，说明一组完整的数据帧格式被破坏，此时 UART 状态寄存器 P_UART_Status 中的 FE 被置 1，如图 5-49 所示。

需注意的是，当奇偶校验和帧检测结果正确时，PE 和 FE 标志位会被相应的自动清除。

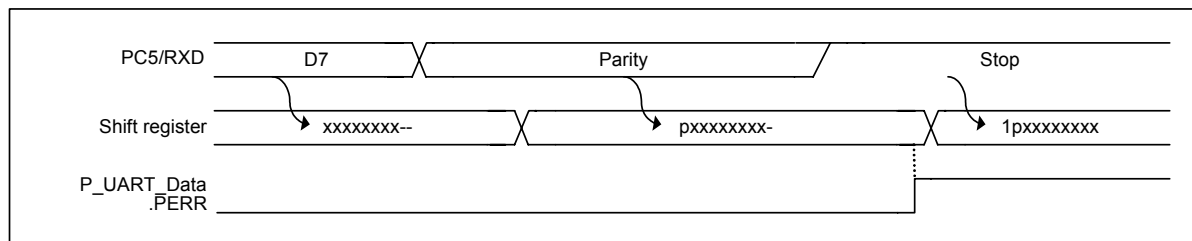


图 5-48 发生奇偶校验错误时序

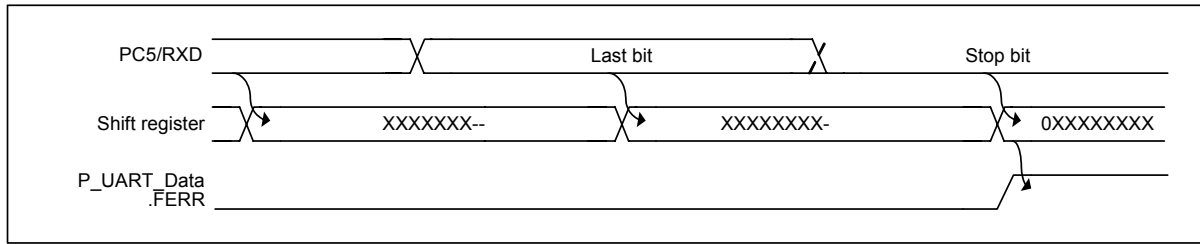


图 5-49 发生帧检测错误时序

3. UART 寄存器

(1) UART 控制寄存器(P_UART_Ctrl, \$0046)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	RXIE	TXIE	RXEN	TXEN	SOFRST	STOPSEL	PSEL	PEN
读 / 写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
默认值	0	0	0	0	0	0	0	0

Bit 7 **RXIE**: 接收中断使能位

1=使能

0=禁止

Bit 6 **TXIE**: 发送中断使能位

1=使能

0=禁止

Bit 5 **RXEN**: UART 接收功能使能位

1=使能

0=禁止

Bit 4 **TXEN**: UART 发送功能使能位

1=使能

0=禁止

Bit 3 **SOFRST**: 软件复位

写:

1=复位所有 UART 模块

0= 无效

Bit 2 **STOPSEL**:停止位宽度选择位

1= 2 位停止位

0= 1 位停止位

Bit 1 **PSEL**: 校验类型选择位

1=偶校验

0=奇校验

Bit 0 **PEN**: 校验使能位

1=使能

0=禁止

(2) UART 波特率分频寄存器(P_UART_Baud, \$0047)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	UARTBAUD 7	UARTBAUD 6	UARTBAUD 5	UARTBAUD 4	UARTBAUD 3	UARTBAUD 2	UARTBAUD 1	UARTBAUD 0
读 / 写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
默认值	0	0	0	0	0	0	0	0

表 5-13 波特率与寄存器 P_UART_Baud 的值的对应关系表

波特率(bps)	Fsys=8MHz
2400	#\$30
4800	#\$98
9600	#\$CC
19200	#\$E6
38400	#\$F3

Bit [7:0] UARTBAUD[7:0]: UART 波特率分频寄存器

按下面的公式推导出寄存器 P_UART_Baud 的值:

$$\text{Baud Rate} = \text{FSYS} / [16 \times (256 - \text{P_UART_Baud})]$$

$$\text{P_UART_Baud} = 256 - \text{FSYS} / (16 \times \text{Baud Rate})$$

(3) UART 状态寄存器(P_UART_Status, \$0048)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	RXIF	TXIF	BUSY	-	-	OERR	PERR	FERR
读 / 写	R	R	R	-	-	R	R	R
默认值	0	0	0	0	0	0	0	0

Bit 7 **RXIF**: 接收中断标志

1=接收数据准备好(满)

0=接收数据未准备好

Bit 6 **TXIF**: 发送中断标志

1=发送数据准备好(空)

0=发送数据未准备好

Bit 5 **BUSY**: UART 发送标志位

1=发送正在进行

0=发送结束, 等待下一次发送

Bit [4:3] 保留

Bit 2 **OERR**: 越界错误标志

读:

1=发生越界错误

0=未发生越界错误

Bit 1 **PERR**: 奇偶校验错误标志

读:

1=奇偶校验错误发生

0= 无奇偶校验错误

Bit 0 **FERR**: 帧错误标志

读:

1=帧错误发生

0=无帧错误

(4) UART 数据寄存器(P_UART_Data, \$0049)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	UARTDATA 7	UARTDATA 6	UARTDATA 5	UARTDATA 4	UARTDATA 3	UARTDATA 2	UARTDATA 1	UARTDATA 0
读 / 写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
默认值	0	0	0	0	0	0	0	0

Bit [7:0] **UARTDATA** [7:0]: UART 数据寄存器

读: 接收到的数据

写: 将发送数据写入寄存器

[例] 5.11.6 初始化 UART

```

lda    #00000000B
sta    P_IOC_Data
sta    P_IOC_Attrib
lda    #11011111B           ; 设 PC5 (Rx) 为输入, PC4 (Tx) 为输出
sta    P_IOC_Dir
lda    #C_UART_SOFTTRST    ; UART 软件复位
sta    P_UART_Ctrl
lda    #(C_UART_RXEN+C_UART_TXEN
      +C_UART_PSEL+C_UART_PEN) ; 使能偶校验
sta    P_UART_Ctrl
lda    #$CC                ; 设置波特率= 9600 @8MHz
sta    P_UART_Baud
lda    #(C_UART_OERR+C_UART_PERR+C_UART_FERR) ; 清除错误标志
sta    P_UART_Status
    
```

5.12 其它外围模块

5.12.1 看门狗

看门狗定时/计数器用于防止系统进入软件死循环。软件死循环通常是由程序逻辑错误或外部干扰引起的。

看门狗定时器电路采用内部独立的 RC 振荡器做为时钟源进行计数, 振荡器的频率为 25KHz, 即 SLOWCK。为避免产生看门狗中断发生, 用户程序中必须定期清除看门狗计数器。若没有在规定的时间内清除看门狗计数器, 看门狗电路会向 CPU 发出中断请求信号。若 CPU 没有回应, 看门狗电路继续运行并间隔地发出中断请求信号。连续发出 8 次后, 则发生看门狗复位: CPU 将复位程序计数器 PC 和状态寄存器并从头开始执行程序。这样, 在系统执行错误的代码后进入死循环, 可以通过复位来跳出。

SPMC65P2708A/2716A 芯片的看门狗功能可以通过特殊功能寄存器(\$7FE0)来使能或者禁止。看门狗定时/计数器原理见图 5-50。

看门狗定时器也可用来唤醒 STOP 模式。系统进入 STOP 模式前, 先对看门狗控制寄存器(P_WDT_Ctrl, \$0032)进行相应设置, 使看门狗在系统进入 STOP 模式后仍工作。当发生看门狗中断后, 系统即被唤醒。但是, 如 5.4.2 节中所述, 看门狗中断的频率必须设置适当, 以避免在系统时钟稳定的过程中发生看门狗复位。

看门狗定时器的特点如下:

- 8 级可选中断频率 (1.5Hz~195Hz)

WDS[2:0]	看门狗复位频率(Hz)	看门狗中断频率 Clock (Hz)
000	195/8	195
001	97/8	97
010	48/8	48
011	24/8	24
100	12/8	12
101	6/8	6
110	3/8	3
111	1.5/8	1.5

(3) 看门狗清除寄存器 (P_WDT_Clr, \$0010)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_WDT_Clr							
读 / 写	W							
默认值	00h							

Bit [7:0] **P_WDT_Clr**: 看门狗清除寄存器，向此寄存器写入“#\$55”可以清除看门狗计数器值。

[例] 5.12.1 使能看门狗定时器功能

```

lda  #C_WDT_Div_16384          ; WDI= Fslow(25KHz)/16384= 1.5Hz
sta  P_WDT_Ctrl
sta  P_WDT_Ctrl
lda  #$FF
sta  P_INT_Flag0              ; 清除中断标志
set  P_INT_Ctrl0,CB_INT_WDIE  ; 使能看门狗中断
    
```

5.12.2 时基定时器

SPMC65P2708A/2716A 芯片配置了一个 23 位的时基定时器，通过寄存器 P_BUZ_Ctrl 的设置，可以产生 15 种不同的时基（见表 5-14），以供编程需要。关于寄存

器 P_BUZ_Ctrl 设置的详细描述，见下面的章节。若时基中断使能，每个时基周期都可产生时基中断。时基定时器可代替通常的定时/计数器作为固定的时钟源。

表 5-14 不同的时基周期列表（表中，系统时钟 FSYS = 8.0MHz）

INTIMS [3:0]	时基周期	
	分频	$F_{T0} = F_{SYS}$
0000	-	该功能禁止
0001	$2^7=128$	16us
0010	$2^8=256$	32us
0011	$2^9=512$	64us
0100	$2^{10}=1024$	128us
0101	$2^{11}=2048$	256us
0110	$2^{12}=4096$	512us
0111	$2^{13}=8192$	1.024ms
1000	$2^{14}=16384$	2.048ms
1001	$2^{15}=32768$	4.096ms
1010	$2^{16}=65535$	8.192ms
1011	$2^{17}=131072$	16.384ms
1100	$2^{18}=262144$	32.768ms
1101	$2^{19}=524288$	65.535ms
1110	2^{21}	262.144ms
1111	2^{23}	1s

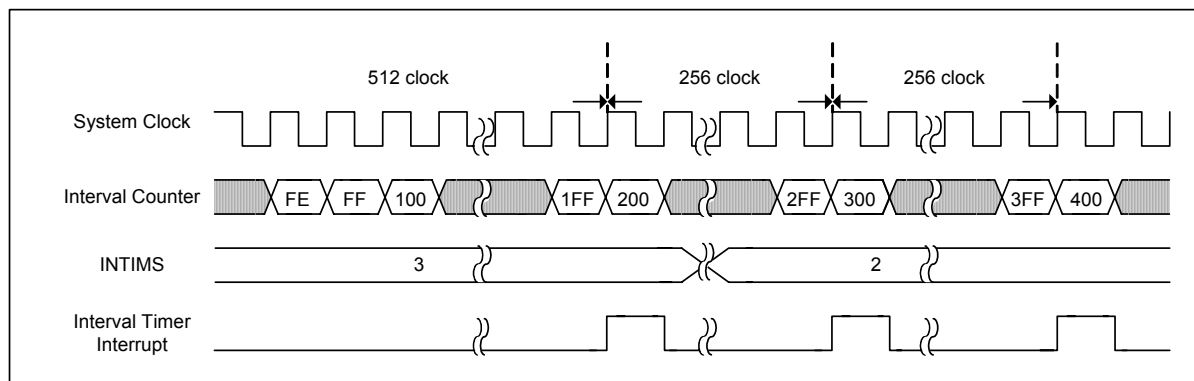


图 5-51 时基定时器工作时序

【例】5.12.2 利用时基定时器产生 512us 中断

```

lda    #C_TBASE_Div_4k           ; 写$60 到 P_BUZ_Ctrl 寄存器
sta    P_BUZ_Ctrl
set    P_INT_Ctrl2, CB_INT_ITVALIE ; 使能时基中断
    
```

5.12.3 蜂鸣器

当蜂鸣器输出功能被使能后，可以输出频率可调、占空比为 50% 的方波来驱动蜂鸣器。蜂鸣器的频率可以通过

寄存器 P_BUZ_Ctrl(\$002D)的 BZFS[3:0]设置。频率范围如下表所列。

表 5-15 蜂鸣器输出频率列表（系统时钟 F_{SYS} = 8.0MHz）

BZFS[3:0]	蜂鸣器输出频率	
	分频	$F_{T0} = F_{SYS} / \text{Divisor}$
0000	-	蜂鸣器输出禁止
0001	64	125k
0010	128	62.5k
0011	256	31.25k
0100	512	15.625k
0101	1024	7.8125k
0110	2048	3.906k
0111	4096	1.953k
1000	8192	0.976k
1001	4	2M
1010	8	1M
1011	16	500k
1100	32	250k
1101	32	250k
1110	32	250k
1111	32	250k

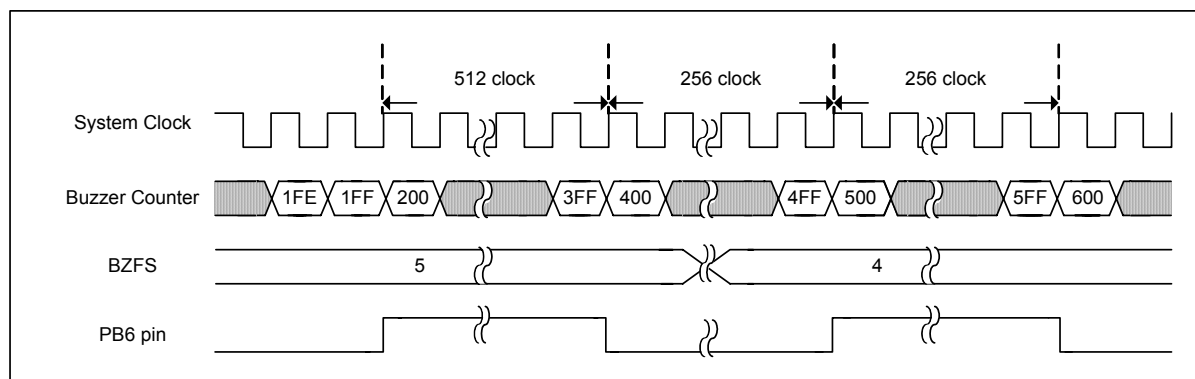


图 5-52 蜂鸣器运行时序

(2) 蜂鸣器控制寄存器(P_BUZ_Ctrl, \$002D)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	INTIMS3	INTIMS2	INTIMS1	INTIMS0	BZFS3	BZFS2	BZFS1	BZFS0
读 / 取	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
默认值	0	0	0	0	0	0	0	0

$$1111 = F_{sys} \div 2^{23}$$

Bit 7 保留

Bit [3:0] **BZFS**[3:0]: 蜂鸣器频率选择位

Bit [6:4] **INTIMS**[3:0]: 时基频率选择位

0000 = 功能禁止

0000 = 功能禁止

0001 = $F_{SYS} \div 64$

0001 = $F_{sys} \div 128$

0010 = $F_{SYS} \div 128$

0010 = $F_{sys} \div 256$

0011 = $F_{SYS} \div 256$

0011 = $F_{sys} \div 512$

0100 = $F_{SYS} \div 512$

0100 = $F_{sys} \div 1024$

0101 = $F_{SYS} \div 1024$

0101 = $F_{sys} \div 2048$

0110 = $F_{SYS} \div 2048$

0110 = $F_{sys} \div 4096$

0111 = $F_{SYS} \div 4096$

0111 = $F_{sys} \div 8192$

1000 = $F_{SYS} \div 8192$

1000 = $F_{sys} \div 16384$

1001 = $F_{SYS} \div 4$

1001 = $F_{sys} \div 32768$

1010 = $F_{SYS} \div 8$

1010 = $F_{sys} \div 65535$

1011 = $F_{SYS} \div 16$

1011 = $F_{sys} \div 131072$

1100 = $F_{SYS} \div 32$

1100 = $F_{sys} \div 262144$

1101 = $F_{SYS} \div 32$

1101 = $F_{sys} \div 524288$

1110 = $F_{SYS} \div 32$

1110 = $F_{sys} \div 2^{21}$

1111 = $F_{SYS} \div 32$

【例】 5.12.3 利用蜂鸣器功能输出方波，周期 128us，占空比 50%

```
lda    #C_BUZ_Div_1k          ;将立即数$05 写入累加器
sta    P_BUZ_Ctrl
```

5.13 芯片配置寄存器

5.13.1 简述

芯片配置寄存器用于设置芯片工作条件，这些设置在烧录时同代码一起被写入芯片。该寄存器的地址分别为 \$7FE0、\$7FE2、\$7FE3。\$7FE0 用于选择系统的时钟源、使能低电压复位 (LVR) 及使能看门狗功能。\$7FE2

用于 RC 振荡器信号输出使能和 IO 口初始化选择。
\$7FE3 用于选择非屏蔽中断 (NMI) 的中断源。

5.13.2 芯片配置寄存器

(1) 芯片配置寄存器 (P_MO, \$7FE0)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	-	-	-	CLKSEL1	CLKSEL0	LVREN	WDTEN	-
读 / 写	-	-	-	R	R	R	R	-
默认值	1	1	1	0	0	0	0	1

Bit [7:5] 保留

Bit 2 **LVREN**: 低电压复位使能位

Bit [4:3] **CLKSEL** [1:0] : 系统时钟源选择位

0 = 禁止 LVR

11 = 保留

1 = 使能 LVR

10 = 外部时钟 External clock

Bit 1 **WDTEN**: 看门狗功能使能位

01 = 内部 RC 振荡器 RC oscillator

0 = 禁止 WDT

00 = 晶体或陶瓷振荡器

1 = 使能 WDT

Bit 0 保留

(2) 保密设置寄存器 (P_SECU, \$7FE2)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	-	-	-	-	-	-	RCOUT	IOINIT
读 / 写	-	-	-	-	-	-	R	R
默认值	1	1	1	1	1	1	1	1

Bit [7:2] 保留

Bit 0 **IOINIT**: IO 口初始化选择位

Bit 1 **RCOUT**: RC 振荡器信号输出使能位

1=所有端口均初始化为悬浮

1=使能 (从 XO 管脚输出时钟信号)

0=所有端口均初始化为下拉输入

0=禁止 (无输出)

(3) NMI 设置寄存器(P_NMI, \$7FE3)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	-	-	-	-	-	NMIS2	NMIS1	NMIS0
读 / 写	-	-	-	-	-	R	R	R
默认值	1	1	1	1	1	1	1	1

Bit [7:3] 保留 100 = PD4 (INT4)作为 NMI 中断源

Bit [2:0] **NMIS** [2:0]: 非屏蔽中断源控制位 011 = PD1 (INT3)作为 NMI 中断源

111 = 禁止 010 = PD0 (INT2)作为 NMI 中断源

110 = 保留 001 = PB5 (INT1)作为 NMI 中断源

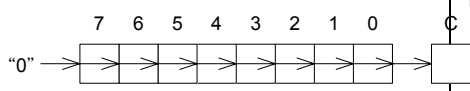
101 = PD5 (INT5)作为 NMI 中断源 000 = PB4 (INT0)作为 NMI 中断源

5.14 指令集

序号	助记符	操作码	字节数	指令周期 (Cycles)	操作	标志位 NV-BDIZC
1.	ADC #dd	69	2	2	累加器 A 带进位加法 $A \leftarrow (A) + (M) + C$ 如果 D 标志为 1, 表示选择十进制加法模式	NV--D-ZC
2.	ADC aa	65	2	3		
3.	ADC aa, X	75	2	4		
4.	ADC aaaa	6D	3	4		
5.	ADC aaaa,X	7D	3	4		
6.	ADC aaaa,Y	79	3	4		
7.	ADC (aa,X)	61	2	6		
8.	ADC (aa), Y	71	2	5		
9.	AND #dd	29	2	2	累加器 A 与数据进行“与”操作 $A \leftarrow (A) \wedge (M)$	N----Z-
10.	AND aa	25	2	3		
11.	AND aa, X	35	2	4		
12.	AND aaaa	2D	3	4		
13.	AND aaaa,X	3D	3	4		
14.	AND aaaa,Y	39	3	4		
15.	AND (aa,X)	21	2	6		
16.	AND (aa), Y	31	2	5		
17.	ASLA	0A	1	2	算术左移	N----ZC
18.	ASL aa	06	2	5		
19.	ASL aa,X	16	2	6		
20.	ASL aaaa	0E	3	6		

序号	助记符	操作码	字节数	指令周期 (Cycles)	操作	标志位 NV-BDIZC
21.	ASL aaaa,X	1E	3	7		
22.	BCC ??	90	2	2*	若(C) = 0, 跳转, $pc \leftarrow (pc) + ??$	-----
23.	BCS ??	B0	2	2*	C = 1, 跳转, $pc \leftarrow (pc) + ??$	-----
24.	BEQ ??	F0	2	2*	Z = 1, 相等, 跳转, $pc \leftarrow (pc) + ??$	-----
25.	BIT aa	24	2	3	用累加器 A 进行位测试	
26.	BIT aaaa	2C	3	4	$Z \leftarrow (A) \wedge (M)$, $N \leftarrow (M7)$, $V \leftarrow (M6)$ N = 1, 数据为负, 跳转, $pc \leftarrow (pc) + ??$	NV----Z-
27.	BMI ??	30	2	2*		-----
28.	BNE ??	D0	2	2*	Z = 0, 不相等, 跳转, $pc \leftarrow (pc) + ??$	-----
29.	BPL ??	10	2	2*	N = 0, 数据不为负, 跳转, $pc \leftarrow (pc) + ??$	-----
30.	BVC ??	50	2	2*	V = 0, 无溢出, 跳转, $pc \leftarrow (pc) + ??$	-----
31.	BVS ??	70	2	2*	V = 1, 有溢出, 跳转, $pc \leftarrow (pc) + ??$	-----
32.	CLC	18	1	2	C \leftarrow "0" 清除进位标志	-----0
33.	CLD	D8	1	2	D \leftarrow "0" 退出十进制运算模式	---0---
34.	CLI	58	1	2	I \leftarrow "0" 清除中断屏蔽位	----0--
35.	CLR aa,0	0F	2	5	位清除 (M.bit) \leftarrow "0"	-----
36.	CLR aa,1	1F	2	5		
37.	CLR aa,2	2F	2	5		
38.	CLR aa,3	3F	2	5		
39.	CLR aa,4	4F	2	5		
40.	CLR aa,5	5F	2	5		
41.	CLR aa,6	6F	2	5		
42.	CLR aa,7	7F	2	5		
43.	CLV	B8	1	2	清除溢出标志: V \leftarrow "0"	-0-----
44.	CMP #dd	C9	2	2	将数据与累加器 A 的值进行比较 (A) - (M)	N----ZC
45.	CMP aa	C5	2	3		
46.	CMP aa, X	D5	2	4		
47.	CMP aaaa	CD	3	4		
48.	CMP aaaa,X	DD	3	4		
49.	CMP aaaa,Y	D9	3	4		
50.	CMP (aa,X)	C1	2	6		
51.	CMP (aa), Y	D1	2	5		
52.	CPX #dd	E0	2	2	将数据与 X 寄存器的值比较 (X) - (M)	N----ZC
53.	CPX aa	E4	2	3		
54.	CPX aaaa	EC	3	4		

序号	助记符	操作码	字节数	指令周期 (Cycles)	操作	标志位 NV-BDIZC
55.	CPY #dd	C0	2	2	将数据与 Y 寄存器的值比较 (Y) - (M)	N----ZC
56.	CPY aa	C4	2	3		
57.	CPY aaaa	CC	3	4		
58.	DEC aa	C6	2	5	减 1 $M \leftarrow (M) - 1$	N----Z-
59.	DEC aa, X	D6	2	6		
60.	DEC aaaa	CE	3	6		
61.	DEC aaaa, X	DE	3	7		
62.	DEX	CA	1	2		
63.	DEY	88	1	2		
64.	EOR #dd	49	2	2	异或 $A \leftarrow (A) \oplus (M)$	N----Z-
65.	EOR aa	45	2	3		
66.	EOR aa, X	55	2	4		
67.	EOR aaaa	4D	3	4		
68.	EOR aaaa, X	5D	3	4		
69.	EOR aaaa, Y	59	3	4		
70.	EOR (aa, X)	41	2	6		
71.	EOR (aa), Y	51	2	5		
72.	INC aa	E6	2	5	加 1 $M \leftarrow (M) + 1$	N----Z-
73.	INC aa, X	F6	2	6		
74.	INC aaaa	EE	3	6		
75.	INC aaaa, X	FE	3	7		
76.	INV aa, 0	87	2	5	按位取反 $(M.bit) \leftarrow \sim(M.bit)$	-----
77.	INV aa, 1	97	2	5		
78.	INV aa, 2	A7	2	5		
79.	INV aa, 3	B7	2	5		
80.	INV aa, 4	C7	2	5		
81.	INV aa, 5	D7	2	5		
82.	INV aa, 6	E7	2	5		
83.	INV aa, 7	F7	2	5		
84.	INX	E8	1	2	$X \leftarrow X + 1$	N----Z-
85.	INY	C8	1	2	$Y \leftarrow Y + 1$	N----Z-
86.	JMP aaaa	4C	3	3	无条件跳转	-----
87.	JMP (aaaa)	6C	3	5	$Pc \leftarrow \text{jump address}$	-----
88.	JSR aaaa	20	3	6	调用子程序 $(sp) \leftarrow (PcH), sp \leftarrow sp - 1, (sp) \leftarrow (PcL),$	-----

序号	助记符	操作码	字节数	指令周期 (Cycles)	操作	标志位 NV-BDIZC		
					$sp \leftarrow sp - 1, pc \leftarrow aaaa$			
89.	LDA #dd	A9	2	2	将数据送入累加器 A 中 $A \leftarrow (M)$	N----Z-		
90.	LDA aa	A5	2	3				
91.	LDA aa, X	B5	2	4				
92.	LDA aaaa	AD	3	4				
93.	LDA aaaa,X	BD	3	4				
94.	LDA aaaa,Y	B9	3	4				
95.	LDA (aa,X)	A1	2	6				
96.	LDA (aa), Y	B1	2	5				
97.	LDX #dd	A2	2	2	将数据送入 X 寄存器中 $X \leftarrow (M)$	N----Z-		
98.	LDX aa	A6	2	3				
99.	LDX aa, Y	B6	2	4				
100.	LDX aaaa	AE	3	4				
101.	LDX aaaa,Y	BE	3	4				
102.	LDY #dd	A0	2	2	将数据送入 Y 寄存器中 $Y \leftarrow (M)$	N----Z-		
103.	LDY aa	A4	2	3				
104.	LDY aa, X	B4	2	4				
105.	LDY aaaa	AC	3	4				
106.	LDY aaaa,X	BC	3	4				
107.	LSR A	4A	1	2	逻辑右移 “0” → 	N----ZC		
108.	LSR aa	46	2	5				
109.	LSR aa, X	56	2	6				
110.	LSR aaaa	4E	3	6				
111.	LSR aaaa,X	5E	3	7				
112.	NOP	EA	1	2	空操作	-----		
113.	ORA #dd	09	2	2	逻辑或 $A \leftarrow (A) \vee (M)$ 	N----Z-		
114.	ORA aa	05	2	3				
115.	ORA aa, X	15	2	4				
116.	ORA aaaa	0D	3	4				
117.	ORA aaaa,X	1D	3	4				
118.	ORA aaaa,Y	19	3	4				
119.	ORA (aa,X)	01	2	6				
120.	ORA (aa), Y	11	2	5				
121.	PHA	48	1	3			$(sp) \leftarrow A, sp \leftarrow sp - 1$	-----

序号	助记符	操作码	字节数	指令周期 (Cycles)	操作	标志位 NV-BDIZC
122.	PHP	08	1	3	$(sp) \leftarrow P \text{ status}, sp \leftarrow sp - 1$	
123.	PLA	68	1	4	$sp \leftarrow sp + 1, A \leftarrow (sp)$	-----
124.	PLP	28	1	4	$Sp \leftarrow sp + 1, P \text{ status} \leftarrow (sp)$	restored
125.	ROL A	2A	1	2	循环左移	N----ZC
126.	ROL aa	26	2	5		
127.	ROL aa, X	36	2	6		
128.	ROL aaaa	2E	3	6		
129.	ROL aaaa,X	3E	3	7		
130.	ROR A	6A	1	2	循环右移 	N----ZC
131.	ROR aa	66	2	5		
132.	ROR aa, X	76	2	6		
133.	ROR aaaa	6E	3	6		
134.	ROR aaaa,X	7E	3	7		
135.	RTI	40	1	6	中断返回 $Sp \leftarrow sp + 1, P \text{ status} \leftarrow (sp), sp \leftarrow sp + 1,$ $pc_L \leftarrow (sp), sp \leftarrow sp + 1, pc_H \leftarrow (sp)$	restored
136.	RTS	60	1	6	子程序返回 $Sp \leftarrow sp + 1, pc_L \leftarrow (sp), sp \leftarrow sp + 1,$ $pc_H \leftarrow (sp)$	-----
137.	SBC #dd	E9	2	2	带借位减法 $A \leftarrow (A) - (M) - \sim(C)$	NV----ZC
138.	SBC aa	E5	2	3		
139.	SBC aa, X	F5	2	4		
140.	SBC aaaa	ED	3	4		
141.	SBC aaaa,X	FD	3	4		
142.	SBC aaaa,Y	F9	3	4		
143.	SBC (aa,X)	E1	2	6		
144.	SBC (aa), Y	F1	2	5		
145.	SEC	38	1	2	C 标志置 “1” : $C \leftarrow "1"$	-----1
146.	SED	F8	1	2	设置十进制运算模式 : $D \leftarrow "1"$	----1--
147.	SEI	78	1	2	中断标志位置 “1”，屏蔽所有的中断 : $I \leftarrow "1"$	----1--
148.	SET aa,0	8F	2	5	位置 “1” $(M.bit) \leftarrow "1"$	-----
149.	SET aa,1	9F	2	5		
150.	SET aa,2	AF	2	5		
151.	SET aa,3	BF	2	5		
152.	SET aa,4	CF	2	5		

序号	助记符	操作码	字节数	指令周期 (Cycles)	操作	标志位 NV-BDIZC
153.	SET aa,5	DF	2	5		
154.	SET aa,6	EF	2	5		
155.	SET aa,7	FF	2	5		
156.	STA aa	85	2	3	将累加器 A 的值赋给存储器 M $(M) \leftarrow A$	-----
157.	STA aa, X	95	2	4		
158.	STA aaaa	8D	3	4		
159.	STA aaaa,X	9D	3	5		
160.	STA aaaa,Y	99	3	5		
161.	STA (aa,X)	81	2	6		
162.	STA (aa), Y	91	2	6		
163.	STX aa	86	2	3	将寄存器 X 的值赋给存储器 M $(M) \leftarrow X$	-----
164.	STX aa, Y	96	2	4		
165.	STX aaaa	8E	3	4		
166.	STY aa	84	2	3	将寄存器 Y 的值赋给存储器 M $(M) \leftarrow Y$	-----
167.	STY aa, X	94	2	4		
168.	STY aaaa	8C	3	4		
169.	TAX	AA	1	2	将累加器 A 值送入 X 寄存器中: $X \leftarrow A$	N----Z-
170.	TAY	A8	1	2	将累加器 A 值送入 Y 寄存器中: $Y \leftarrow A$	N----Z-
171.	TST aa,0	07	2	3	位测试 $Z \leftarrow \sim(M.bit)$	----Z-
172.	TST aa,1	17	2	3		
173.	TST aa,2	27	2	3		
174.	TST aa,3	37	2	3		
175.	TST aa,4	47	2	3		
176.	TST aa,5	57	2	3		
177.	TST aa,6	67	2	3		
178.	TST aa,7	77	2	3		
179.	TSX	BA	1	2	将堆栈指针(SP)的值送入 X 寄存器: $X \leftarrow sp$	N----Z-
180.	TXA	8A	1	2	将 X 寄存器的值送入累加器 A: $A \leftarrow X$	N----Z-
181.	TXS	9A	1	2	将 X 寄存器的值送入堆栈指针 (SP): $sp \leftarrow X$	N----Z-
182.	TYA	98	1	2	将 Y 寄存器的值送入累加器 A: $A \leftarrow Y$	N----Z-

* : 若跳转发生，需要再加 1 个指令周期。

6 电气特性

6.1 极限参数 (VSS = 0)

特性	符号	最小值	典型值	最大值	单位	条件
额定电压 VDD	VDD	-0.3	-	6.0	V	
额定输入电压	V _{IN}	-0.3	-	VDD + 0.3	V	
Vdd 管脚电流	I _{VDD}	-	-	100	mA	
Vss 管脚电流	I _{VSS}	-	-	120	mA	
各 I/O 端口源电流	I _{OHR}	-	-	15	mA	
各 I/O 端口吸入电流	I _{OLR}	-	-	20	mA	
功耗	PD	-	-	350	mW	Ta=85°C
储存温度	T _{STR}	-55	-	125	°C	

注：若超过上表列出的极限值，将引起运行错误甚至毁坏芯片。请参看 DC/AC 电气特性，了解芯片的正常运行条件。

6.2 推荐运行参数

特性	符号	最小值	典型值	最大值	单位	条件
工作电压	VDD	2.4	-	5.5	V	禁止 LVR 功能
	VDD	LVR	-	5.5	V	使能 LVR 功能
系统时钟	F _{sys}	200K	-	8.0M	Hz	VDD = 3.0~5.5V
		200K	-	4.0M		VDD = 2.4~5.5V
环境温度	T _{OPR}	-40	-	85	°C	

注：F_{sys} = 1/2 X Fosc，其中 Fosc 为晶体频率。

6.3 DC/AC 电气特性 (VDD = 5.0V, T_A = -40°C~85°C)

6.3.1 项目符号定义

符号	定义	符号	定义
V _{IH}	输入高电平电压	I _{OH}	输出高电平电流 (源电流)
V _{IL}	输入低电平电压	I _{OL}	输出低电平电流 (吸入电流)
V _{TH}	输入阈值电压	I _Z	输入漏电流
I _P	上拉/下拉电流		

6.3.2 管脚说明

管脚	描述	符号	最小值	典型值	最大值	单位	条件
VDD,VSS	Normal 模式的电源电流	I_{DD}	-	12	16	mA	FOSC = 16.0MHz @ VDD = 5.5V
	Halt 模式的电源电流	I_{HALT}		8	11	mA	VDD = 5.5V
	Stop 模式的电源电流	I_{PD}	-	8	15	μA	VDD = 5.5V
XI,XO	振荡稳定时间	t_{ST}	-	1024	-	F _{sys}	晶体模式
	RC 时钟频率偏差	DFV	-15	-	+15	%	RC 振荡器模式
	外部电容	C_{RC}	20	-	100	pf	RC 振荡器模式
	外部电阻	R_{RC}	2.0	-	-	k Ω	RC 振荡器模式
	高电平时钟脉宽	t_H	31.25	-	-	ns	外部时钟源模式
	低电平时钟脉宽	t_L	31.25	-	-	ns	外部时钟源模式
PA[7:0], PB[5:0], PC[7:0], PD[7:0], PE[4:0]	输入: a. 施密特触发器触发 b. 上拉/下拉/悬浮	V_{IH}	3.5	-	-	V	
		V_{IL}	-	-	1.4	V	
		V_{TH}	-	0.5	-	V	
	输出: a. 有 LED 驱动能力 b. 4mA/10mA 驱动输出	I_{OH}	4.0	-	-	mA	VOH = 4.5V
		I_{OL}	10.0	-	-	mA	VOL = 0.5V
		I_P	-	50	-	μA	VIN = VDD 或 VSS
PB[7:6]	输入: a. 施密特触发器触发 b. 上拉/下拉/悬浮	V_{IH}	3.5	-	-	V	
		V_{IL}	-	-	1.4	V	
		V_{TH}	-	0.5	-	V	
	输出: a. 有 LED 驱动能力 b. 4mA/20mA 驱动输出	I_{OH}	4.0	-	-	mA	VOH = 4.5V
		I_{OL}	20.0	-	-	mA	VOL = 0.5V
		I_P	-	50	-	μA	VIN = VDD 或 VSS
所有端口	I/O 端口高阻态输入泄漏电流	I_Z	-	-	10	μA	
RESETB	输入: a. 施密特触发器触发 b. 上拉	V_{IH}	3.5	-	-	V	
		V_{IL}	-	-	1.4	V	
		V_{TH}	-	0.5	-	V	
	输出: 无	I_P	-	125	-	μA	VIN = VSS
	t_W	1.0	-	-	μS	输入最小宽度	

6.4 模拟接口电气特性(VDD = 5.0V, T_A = - 40°C ~ 85°C)

功能	描述	符号	最小值	典型值	最大值	单位	条件	
10 位 A/D 转换器	ADC 时钟频率	F _{CK_AD}	-	-	1.4	MHz		
	精度	N _{R_AD}	-	-	10	Bit		
	参考电压范围	V _{RT}	2.4	-	VDD	V	PB7 作为参考电压输入 V _{RT}	
	参考电压输入电流	I _{RT}	-	-	800	uA	PB7 作为参考电压输入 V _{RT}	
	输入电压	V _{AIN}	0	-	VDD/V _{RT}	V		
	转换速率	F _{AD}	-	-	100	KHz	VDD=5.0V (Note 1)	
	输入阻抗	R _{AIN}	-	-	-		(Note 2)	
	误差	积分线性误差	E _{INL_AD}	-	-	±2.0	LSB	
		微分线性误差	E _{DNL_AD}	-	-	±1.0	LSB	
		零点飘移	E _{ZOE_AD}	-	-	±1.5	LSB	
满量程误差		E _{FSE_AD}	-	-	±2.5	LSB		
累计误差		E _{ALL_AD}	-	-	±2.5	LSB		
8 位 A/D 转换器	ADC 时钟频率	F _{CK_AD}	-	-	1.4	MHz		
	精度	N _{R_AD}	-	-	8.0	Bit		
	参考电压范围	V _{RT}	2.4	-	VDD	V	PB7 作为参考电压输入 V _{RT}	
	参考电压输入电流	I _{RT}	-	-	800	uA	PB7 作为参考电压输入 V _{RT}	
	输入电压	V _{AIN}	0	-	VDD/V _{RT}	V		
	转换速度	F _{AD}	-	-	100	KHz	VDD=5.0V (注 1)	
	输入阻抗	R _{AIN}	-	-	-		(注 2)	
	误差	积分线性误差	E _{INL_AD}	-	-	±1.0	LSB	
		微分线性误差	E _{DNL_AD}	-	-	±1.0	LSB	
		零点调整	E _{ZOE_AD}	-	-	±1.0	LSB	
满量程误差		E _{FSE_AD}	-	-	±1.0	LSB		
积累误差		E _{ALL_AD}	-	-	±1.0	LSB		
低电压复位	LV40=0	V _{LVR}	2.3	2.5	2.7	V		
	LV40=1	V _{LVR}	3.68	4.0	4.32	V		

注 1: 由于受 ADC 采样电容、输入阻抗及外部电路影响，采样电容充电直至与输入电压相等需要一定的时间。这段时间的最小值为 1.5us，但通常情况下会大于这个值。

注 2: (1).SPMC65P2708A/2716A 的 ADC 有一个采样保持电路，如下所示，允许 A/D 转换功能后，模拟电压输入到内部采样保持电容上。(图 6-1为 ADC 采样简化结构图)

(2).如果外部电路的阻抗比较大，可能会导致输入到 ADC 的电压在采样周期内（1.5us）未达到稳定状态。因

此，建议外部电路的阻抗不宜过大。

(3).如果外部输入阻抗较大，建议在模拟电压输入端接一个电容，电容的容值为 0.01 uF ~0.1 uF。

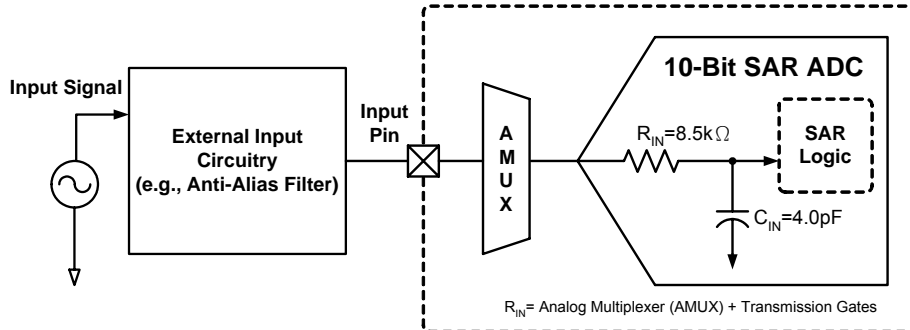


Fig. 1. Equivalent ADC Circuit for Estimating Settling Time

图 6-1 ADC 采样简化结构图

7 封装和管脚焊接位置

7.1 管脚分配和位置

请联系凌阳创新科技销售代表，以获得更多信息。

7.2 分类

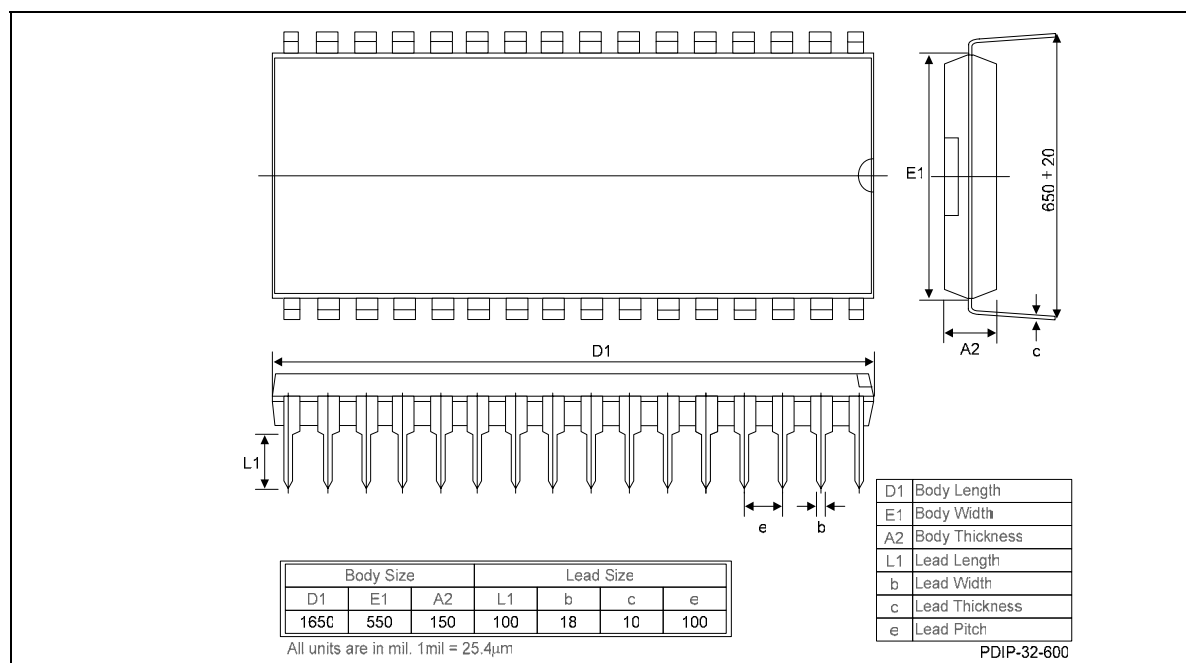
芯片系列号	封装类型
SPMC65P2708A-NnnV-HD112	封装类型 – PDIP 32 (600mil)
SPMC65P2708A-NnnV-HD132	封装类型 – PDIP 42 (600mil)
SPMC65P2708A-NnnV-HG062	封装类型 – SSOP 48 (295mil)
SPMC65P2716A-NnnV-PD111	封装类型 – PDIP 32 (600mil)
SPMC65P2716A-NnnV-PD131	封装类型 – PDIP 42 (600mil)
SPMC65P2716A-NnnV-HG061	封装类型 – SSOP 48 (295mil)

注 1：代码号 (Nnn) 用于用户区分不同的芯片。

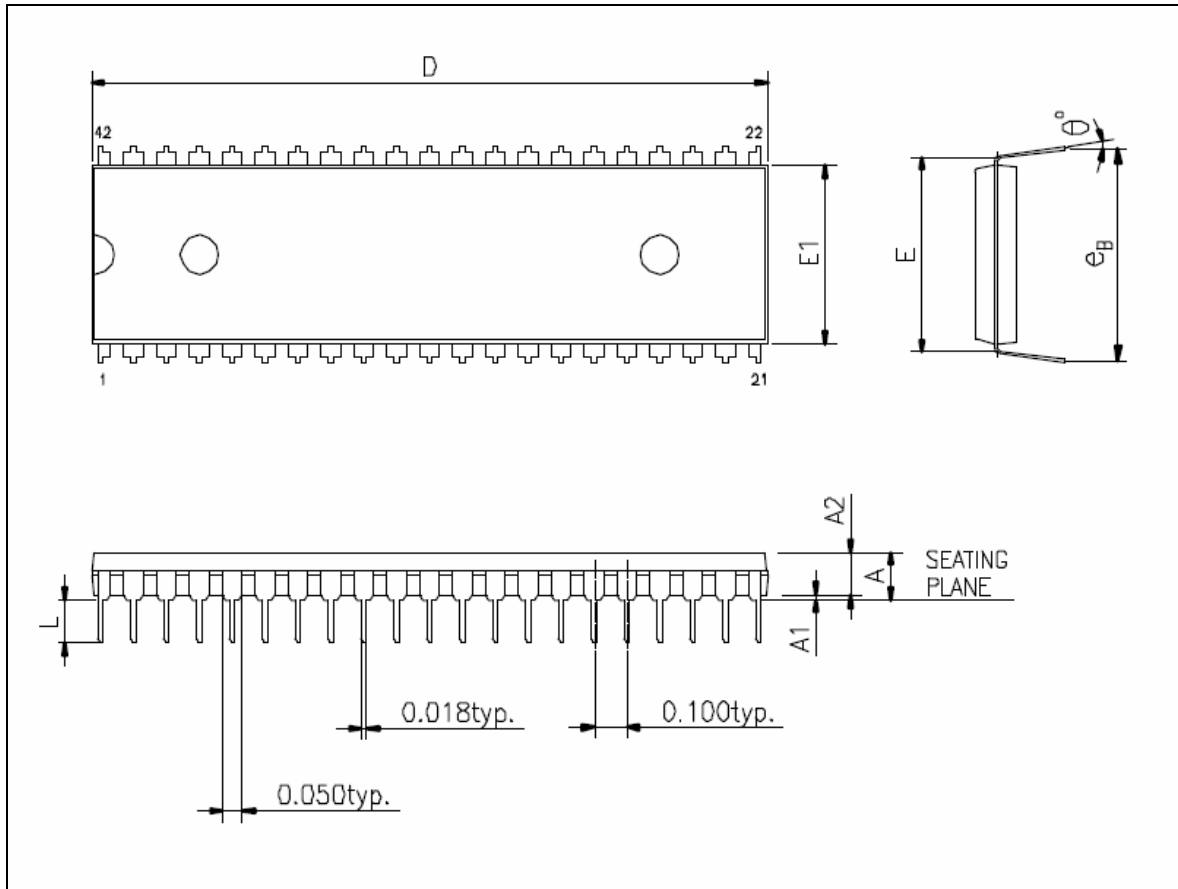
注 2：Nnn (代码号 , N = A - Z 或 0 - 9 , nn = 00 - 99) ; V (版本号 , V = A - Z) 。

7.3 封装

7.3.1 PDIP32 (600mil)



7.3.2 PDIP42 (600mil)

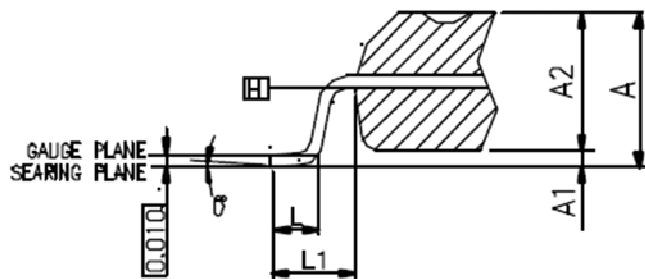
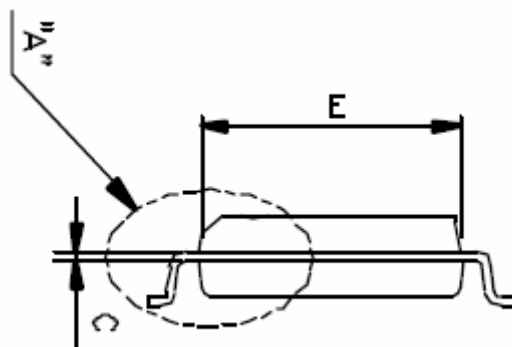
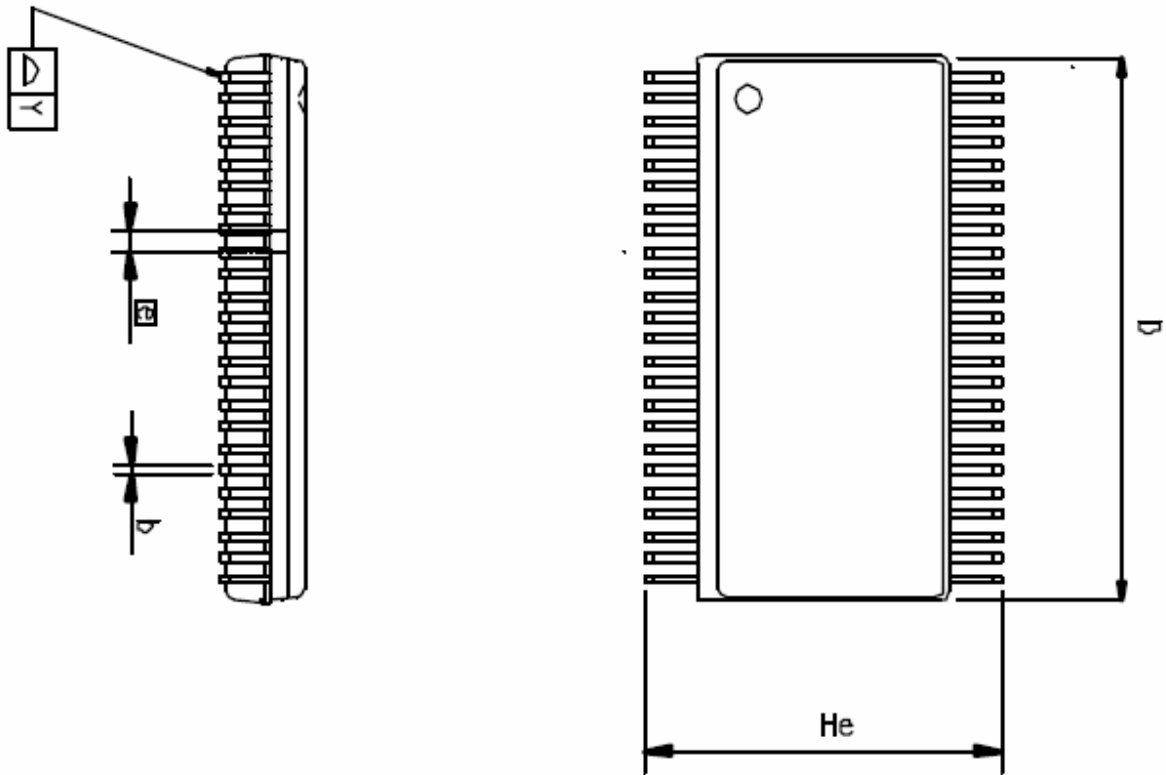


SYMBOLS	MIN.	NOR.	MAX.
A	—	—	0.220
A1	0.015	—	—
A2	0.150	0.155	0.160
D	2.055	2.060	2.070
E	0.600 BSC		
E1	0.540	0.545	0.550
L	0.115	0.130	0.150
e_B	0.630	0.650	0.670
θ	0	7	15

UNIT : INCH

NOTE:
1. JEDEC OUTLINE: N/A

7.3.3 SSOP48 (295mil)



DETAIL : A

SYMBOLS	MIN.	NOM.	MAX.
A	0.095	0.102	0.110
A1	0.008	0.012	0.016
A2	0.089	0.094	0.099
b	0.008	0.010	0.013
c	—	0.008	—
D	0.620	0.625	0.630
E	0.291	0.295	0.299
e	—	0.025	—
He	0.396	0.406	0.416
L	0.020	0.030	0.040
L1	—	0.056	—
Y	—	—	0.003
θ°	0°	—	8°

UNIT : INCH

NOTES:

1. DATUM PLANE \square IS LOCATED AT THE BOTTOM OF THE MOLD PARTING LINE COINCIDENT WITH WHERE THE LEAD EXITS THE BODY.
2. DIMENSIONS E AND D DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 6 MIL PER SIDE. DIMENSIONS E AND D DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE \square
3. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.

7.4 储存条件与时间

封装	湿度灵敏性级别	最大回流焊温度	储存条件	干燥包装
SSOP	3 级	255 +5/-0°C	N/A	否
PDIP	3 级 (参考)	220 +5/-0°C (参考)	N/A	否
PDIP	3 级 (参考)	255 +5/-0°C (参考)	N/A	否

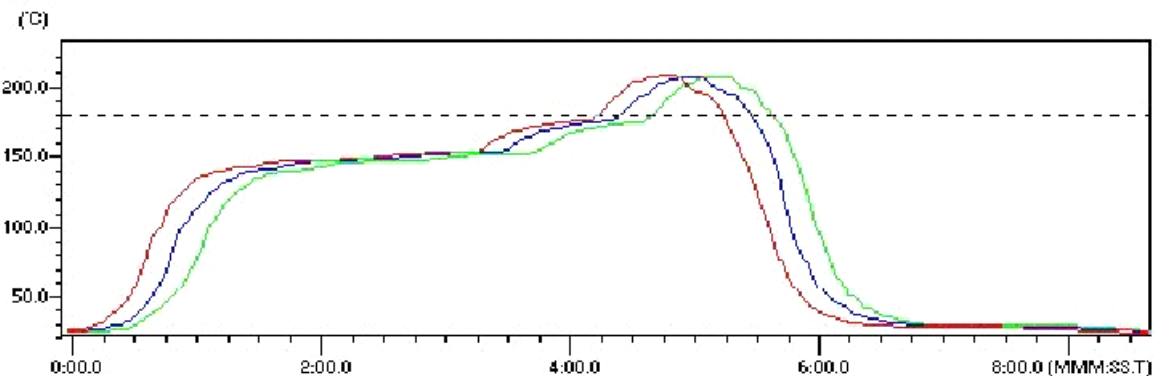
注 1：请参考 IPC/JEDEC 规范 J-STD-020A 及 EIA JEDEC 规范 JFSD22-A112。

注 2：参看干燥包装上的“注意事项”。

7.5 建议表面贴装(SMT)温度曲线

此温度曲线建议是提供表面粘着(SMT)制程的一份大致的指导方针。大部分之凌阳创新科技导线架(Lead-Frame Base)产品皆采用纯锡电镀或锡铋电镀。关于预先电镀导线架产品而搭配 63/37 锡铅比之锡膏时，我们所建议回焊炉所需之最高温度为摄氏 240 度至 245 度。

Recommended Reflow Profile for 63/37 Solder Paste or Cu lead frame



此温度曲线适用于锡含量63%或62%之产品或锡膏, 可作为初期建立回焊炉温度曲线之参考.

回焊炉温度曲线摘要:

- (1) 升温, 以每秒升温1~3度加热至摄氏140度
- (2) 预热, 在摄氏140~150度间停留120~160秒
- (3) 焊接, 再以每秒升温2~3度加热至尖峰温度摄氏约220~225度, 而温度在摄氏183度以上之停留时间为45~75秒
- (4) 降温, 再以每秒4~2度降温至原始温度(室温)

Recommended Reflow Profile for Lead-free Solder Paste or PPF lead frame

