

DATA SHEET

SunplusIT

SPMC65P1504A/1502A

I/O 型微控制器(OTP)

V1.1 – JUL. 19, 2006

中文版

3F, 19 Innovation First Road • Science Park • Hsin-Chu • Taiwan 300 • R.O.C.

Tel: 886-3-563-2822 Fax: 886-3-668-7799 Web: www.sunplusit.com

版权声明

凌阳创新股份有限公司保留对此文件修改之权利且不另行通知。凌阳创新股份有限公司所提供之信息相信为正确且可靠之信息，但并不保证本文件中绝无错误。请于向凌阳创新股份有限公司提出订单前，自行确定所使用之相关技术文件及规格为最新之版本。若因贵公司使用本公司之文件或产品，而涉及第三人之专利或著作权等智能财产权之应用及配合时，则应由贵公司负责取得同意及授权，本公司仅单纯贩售产品，上述关于同意及授权，非属本公司应为保证之责任。又未经凌阳创新股份有限公司之正式书面许可，本公司之所有产品不得使用于医疗器材，维持生命系统及飞航等相关设备。

修订记录

版本	日期	译者	修订内容	对应页码
V1.1	2006/07/19	朱萌	翻译 “SPMC65P1504A/1502A Data Sheet V1.1”	

目录

	<u>页</u>
1 概述	6
2 芯片特性	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	11
4.2.1 28 PIN 封装 (SPMC65P1504A)	11
4.2.2 24 PIN 封装(SPMC65P1504A/SPMC65P1502A)	11
4.2.3 18 PIN 封装(SPMC65P1502A)	11
5 功能描述	12
5.1 CPU	12
5.1.1 CPU 简介	12
5.1.2 CPU 寄存器	12
5.2 存储器	14
5.2.1 简介	14
5.2.2 存储空间分配	14
5.2.3 硬件控制寄存器	15
5.2.4 系统运行控制寄存器	19
5.2.5 芯片配置寄存器	19
5.3 时钟源	20
5.4 省电模式	20
5.4.1 简介	20
5.4.2 STOP 模式	21
5.4.3 HALT 模式	23
5.5 中断	24
5.5.1 简介	24
5.5.2 外部中断	24
5.5.3 其它中断	25
5.5.4 中断寄存器	26
5.6 复位	29
5.6.1 简介	29
5.6.2 上电复位 (POR)	29
5.6.3 外部复位 (ERST)	29
5.6.4 低电压复位 (LVR)	29
5.6.5 看门狗复位(WDR)	30
5.6.6 非法地址复位(IAR)	30
5.6.7 软件复位 (Software)	30

5.7 I/O 端口.....	33
5.7.1 简介.....	33
5.7.2 端口 A.....	34
5.7.3 端口 B.....	36
5.7.4 端口 C.....	38
5.7.5 端口 D.....	40
5.8 定时/计数器.....	42
5.8.1 简介.....	42
5.8.2 定时/计数器 0.....	42
5.9 Capture/Compare/PWM (CCP).....	47
5.9.1 8 位比较模式.....	47
5.9.2 16 位比较模式.....	47
5.9.3 8 位捕获模式.....	48
5.9.4 16 位模式.....	50
5.9.5 8 位 PWM 模式.....	50
5.10 其它外围模块.....	53
5.10.1 看门狗定时器.....	53
5.10.2 时基定时器.....	55
5.10.3 蜂鸣器控制寄存器.....	56
5.11 芯片配置寄存器.....	58
5.11.1 简介.....	58
5.11.2 芯片配置寄存器.....	58
5.12 指令集.....	60
6 电气特性.....	66
6.1 极限参数(VSS = 0).....	66
6.2 推荐运行参数.....	66
6.3 DC/AC 电气特性(VDD = 5.0V, T _A = -40°C~85°C).....	66
6.3.1 符号定义.....	66
6.3.2 管脚属性描述.....	67
6.4 模拟接口电气特性 (VDD = 5.0V, T _A = -40°C~85°C).....	68
7 封装和管脚焊接位置.....	69
7.1 管脚分配和位置.....	69
7.2 订货信息.....	69
7.3 封装.....	70
7.3.1 PDIP 18 (300mil).....	70
7.3.2 PDIP 24 (300mil).....	70
7.3.3 PDIP 28 (300mil).....	71
7.3.4 SOP 18 (300mil).....	71
7.3.5 SOP 24 (300mil).....	72
7.3.6 SOP 28 (300mil).....	73
7.4 储存条件及时间.....	73

7.5 建议表面贴装(SMT)温度曲线..... 73

1 概述

SPMC65P1504A 和 SPMC65P1502A 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。它们的内核和外围几乎相同，只是 SPMC65P1502A 的 ROM 空间比 SPMC65P1504A 小，

存储器的详细信息请参考后续介绍。这两款芯片主要应用于小家电、工业设备控制、电池充电器。下面介绍这两款芯片的主要特性。

2 芯片特性

■ 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 8 MHz
- 支持位操作指令（置 1、清零、取反、测试）

■ 存储空间

- 4K 字节的程序空间 (OTP)，并且可设置保密功能 (SPMC65P1504A)
- 2K 字节的程序空间 (OTP)，并且可设置保密功能 (SPMC65P1502A)
- 128 字节的 RAM，其中包括堆栈空间

■ I/O 端口

- 多达 23/19 /13 个具有复用功能的双向 I/O 端口
- 所有 I/O 端口的输入均由施密特触发器构成
- 可设置为带上拉/下拉电阻输入或者悬浮输入
- 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
- 1 个外部中断，可设置为非屏蔽中断 NMI
- 4 个内部中断

■ 复位管理

- 增强的复位系统
- 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

■ 外围模拟电路

- 低电压复位 (2.5V/4V)

■ 一个 16 位定时/计数器(定时/计数器 0)

- 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 位和 8 位比较输出功能
- 8 位 PWM 输出功能

■ 时基

- 频率选择：1Hz ~ 62.5kHz @8MHz(系统时钟)
- 15 级分频选择

■ 蜂鸣器输出

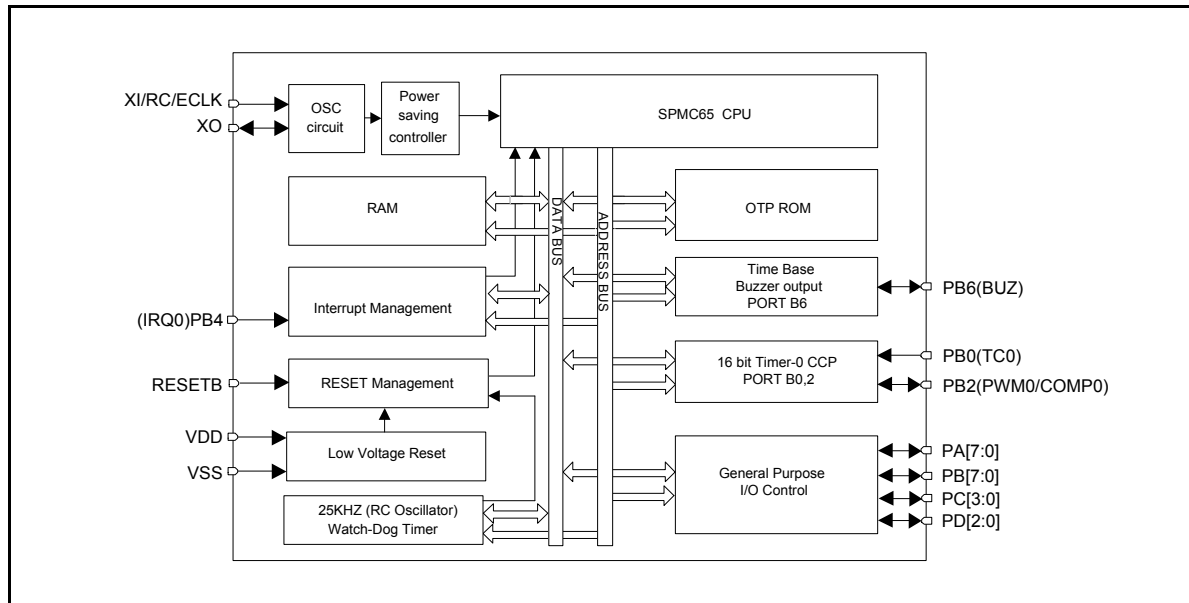
- 频率选择：1kHz ~ 2MHz @8MHz(系统时钟)
- 12 级分频选择

■ 可编程看门狗定时器

- 中断频率选择：1.5Hz ~ 195Hz @8MHz(系统时钟)

芯片	EPROM	RAM	Max I/O	CCP 功能			SPI	UART	封装类型
				比较	捕获	PWM			
SPMC65P1504A	4K x 8	128 x 8	23	1	1	1	N	N	PDIP28, SOP28
	4K x 8	128 x 8	19	1	1	1	N	N	PDIP24, SOP24
SPMC65P1502A	2K x 8	128 x 8	19	1	1	1	N	N	PDIP24, SOP24
	2K x 8	128 x 8	13	1	1	1	N	N	PDIP18, SOP18

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
	28PIN (1504)	24PIN (1504/1502)	18PIN (1502)			
VDD	28	24	18	S	电源	
VSS	27	23	17	S	接地	
XI/RC/ECLK	26	22	16	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； 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	25	21	15	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	1	--	--	I/O	Port A7	
PA6	2	--	--	I/O	Port A6	
PA5	3	1	--	I/O	Port A5	
PA4	4	2	--	I/O	Port A4	
PA3	5	3	1	I/O	Port A3	
PA2	6	4	2	I/O	Port A2	
PA1	7	5	3	I/O	Port A1	
PA0	8	6	4	I/O	Port A0	
PB7	9	7	5	I/O	Port B7	
PB6/BUZ	10	8	6	I/O	Port B6	蜂鸣器输出
PB5	11	9	7	I/O	Port B5	
PB4/IRQ0	12	10	8	I/O	Port B4	外部中断 0 输入
PB3	13	11	--	I/O	Port B3	
PB2/PWM0/COMP0	14	12	9	I/O	Port B2	定时/计数器 0 的 PWM 输出、定时/计数器 0 比较输出
PB1	15	13	10	I/O	Port B1	
PB0/TC0	16	14	11	I/O	Port B0	定时/计数器 0 的捕获输入/定时/计数器 0 的外部事件输入
PC3	17	--	--	I/O	Port C3	
PC2	18	--	--	I/O	Port C2	

管脚名称	管脚号			类型	主要功能	可选功能
	28PIN (1504)	24PIN (1504/1502)	18PIN (1502)			
PC1	19	15	--	I/O	Port C1	
PC0	20	16	--	I/O	Port C0	
PD2	21	17	12	I/O	Port D2	
PD1	22	18	13	I/O	Port D1	
PD0	23	19	--	I/O	Port D0	
RESETB	24	20	14	I	Reset pin	外部复位输入

4.2 管脚分配

4.2.1 28 PIN 封装 (SPMC65P1504A)

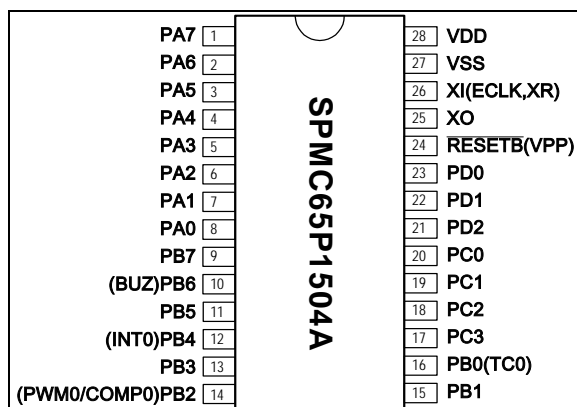


图 4-1

4.2.3 18 PIN 封装(SPMC65P1502A)

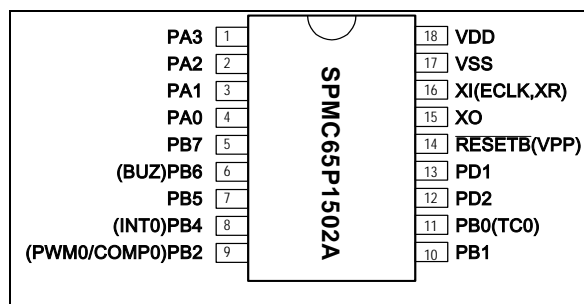
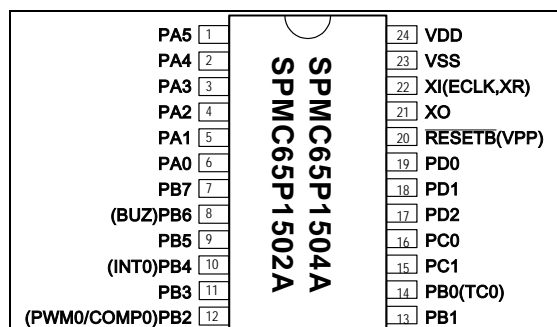


图 4-2

4.2.2 24 PIN 封装

(SPMC65P1504A/SPMC65P1502A)



5 功能描述

5.1 CPU

5.1.1 CPU 简介

SPMC65P1504A/1502A 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)。系统运行时钟 (F_{sys}) 最高可以达到 8.0MHz。

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，见图 5-1。

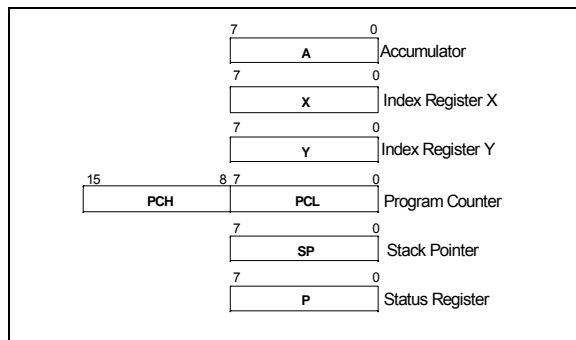


图 5-1 System registers

■ 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

累加器 A 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)

堆栈指针 (SP) 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。

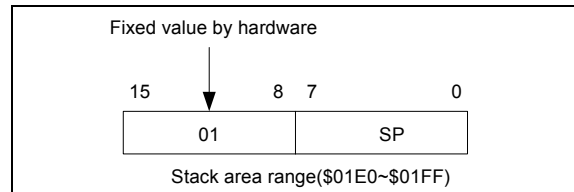


图 5-2 堆栈指针寄存器

[例] 1.2.1 堆栈指针初始化

```
ldx #C_STACK_BOTTOM ;将堆栈指针初始化为$FF
txs
```

■ 程序计数器(PC)

程序计数器(PC)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\$FFFC。

■ 状态寄存器(P)

状态寄存器(P)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所述。

注意：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中是否发生溢出。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)

SPMC65P1504A/1502A 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。

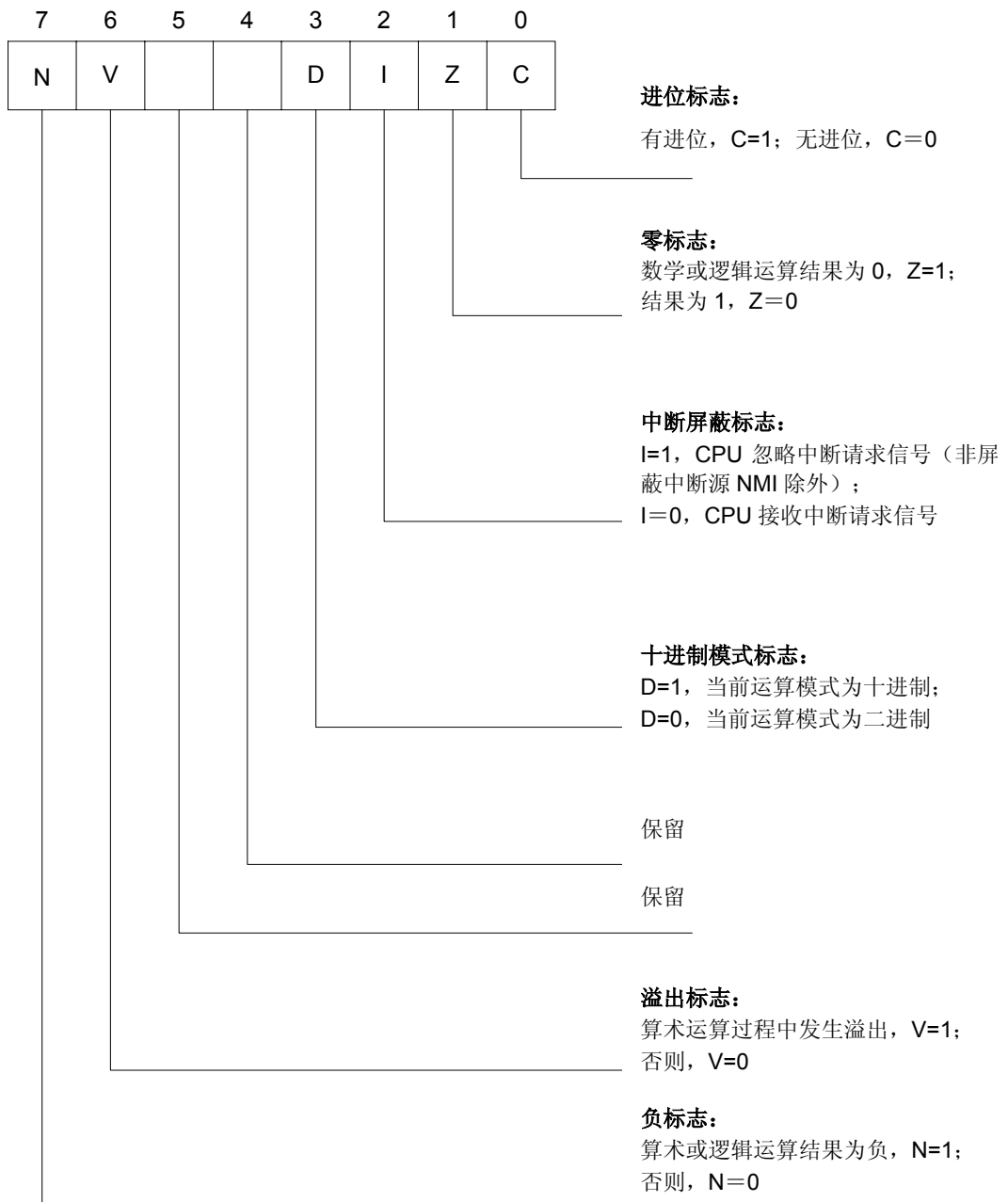


图 5-3 状态寄存器 (P)

注意：并非所有的指令执行都会影响状态寄存器的值。在 25 章中将每个指令作详细描述。

5.2 存储器

5.2.1 简介

SPMC65P1504A/1502A 这两款芯片均有独立的程序存储区和数据存储区。程序存储区为只读，当执行程序时，若向其写入数据，将导致非法地址复位，SPMC65P1504A/1502A 的程序存储区大小分别为 4K 和 2K 字节。数据存储区可读可写（RAM），SPMC65P1504A/1502A 的数据存储区大小为 128 字节（包括堆栈区）。

5.2.2 存储空间分配

SPMC65P1504A 和 SPMC65P1502A 的存储空间分为几个部分，分别如图 5-5和图 5-6所示。二者的差别在于 OTP ROM 的大小。在此仅以 SPMC65P1504A 为例对各个部分进行详细说明，SPMC65P1502A 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。\$00A0 ~ \$00FF 和 \$01E0 ~ \$01FF 为数据存储区（RAM），其中 \$01E0~\$01FF（32 个字节）又定义为堆栈区。堆栈指针向下生长（从 \$01FF 到 \$01E0）。一旦堆栈溢出，CPU 复位。

\$7FE0 ~ \$7FE3 这 4 个字节为芯片配置寄存器，用户可以通过它设置一些特殊功能。详情参见 5.11。

SPMC65P1504A 支持 4K 字节的 OTP ROM（\$F000 ~ \$FFFF）来存储程序，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，程序可以读出。

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。

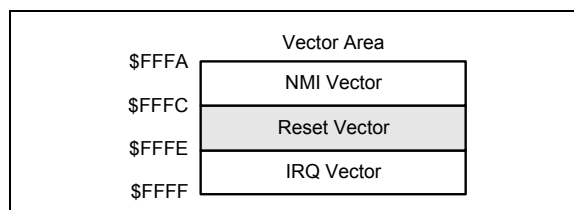


图 5-4 中断和复位向量表

[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
```

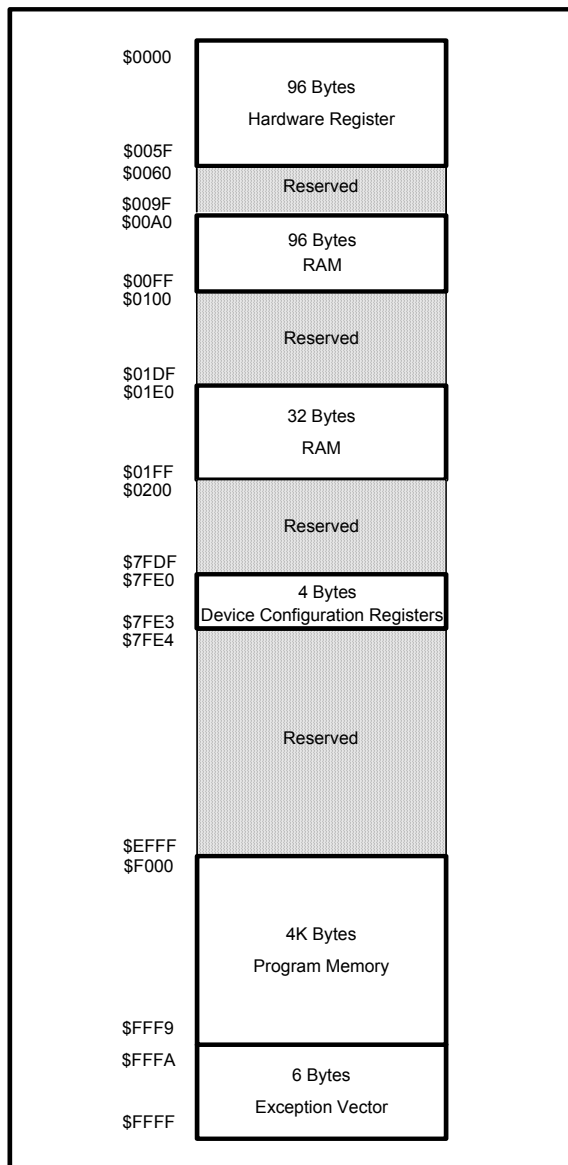


图 5-5 SPMC65P1504A 存储空间分配图

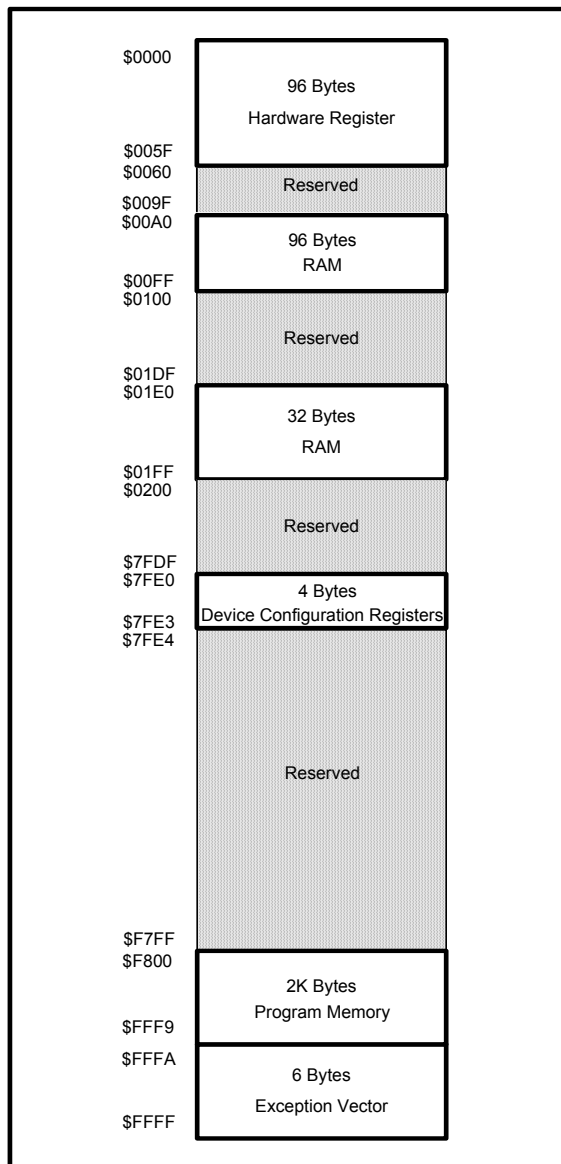


图 5-6 SPMC65P1502A 存储空间分配图

5.2.3 硬件控制寄存器

SPMC65P1504A/1502A 有多达 32 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

■ 控制寄存器列表

(1) \$0000~000F: I/O 端口及中断控制寄存器

定时/计数器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。注意：保留区的地址不能被操作。下面是控制寄存器的列表，详细介绍请见各章节。

地址	功能	(ERST) 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	0			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	0			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	端口 A 的数据方向 0=输入 1=输出.								
			W	端口 A 的数据方向 0=输入 1=输出.								
\$0005	端口 B 方向寄存器 (P_IOB_Dir)	00h	R	端口 B 的数据方向 0=输入 1=输出.								
			W	端口 B 的数据方向 0=输入 1=输出.								
\$0006	端口 C 方向寄存器 (P_IOC_Dir)	00h	R	0			端口 C 的数据方向. 0=输入 1=输出					
			W	-			端口 C 的数据方向. 0=输入 1=输出					
\$0007	端口 D 方向寄存器 (P_IOD_Dir)	00h	R	0			端口 D 的数据方向. 0=输入 1=输出					
			W	-			端口 D 的数据方向. 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	0			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	0			端口 D 属性					
			W	-			端口 D 属性					
\$000C	中断标志 0 (P_INT_Flag0)	00h	R	0	WDIF	0	0	0	0	0	0	IRQ0IF
			W	写入'1'清除中断标志								
\$000D	中断控制 0 (P_INT_Ctrl0)	00h	R	0	WDIE	0	0	0	0	0	0	IRQ0IE
			W	-	WDIE	-	-	-	-	-	-	-
\$000E	中断标志 1 (P_INT_Flag1)	00h	R	0	CAP0IF	0	0	0	0	0	0	T00IF
			W	写入'1'清除中断标志								
\$000F	中断控制 1 (P_INT_Ctrl1)	00h	R	0	CAP0IE	0	0	0	0	0	0	T00IE
			W	-	CAP0IE	-	-	-	-	-	-	-

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

(2) \$0010~\$0014: Timer/PWM 设置 & 数据寄存器

地址	功能	复位值 (ERST)	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
\$0010	看门狗清除 (P_WDT_Clr)	00h	R	0	0	0	0	0	0	0	0
			W	写 #55 清除看门狗计数(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 0 功能选择				
			W	-			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 0 预分频选择				
			W	-			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 位或 16 位（低字节）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 位或 16 位（低字节）定时/计数器 0 重载计数值							
	比较器 0 计数值的低字节 (P_TMR0_Comp)	00h	R	8 位或 16 位（低字节）定时/计数器 0 计数值							
			W	8 位或 16 位（低字节）比较器 0 计数值							
	捕获器 0 捕获值的低字节 (P_TMR0_Cap)	00h	R	8 位或 16 位（低字节）捕获器 0 捕获脉宽值							
			W	8 位或 16 位（低字节）捕获器 0 重载值							
PWM0 低字节周期值 (P_TMR0_PWMPeriod)	00h	R	8 位 PWM 0 的周期值								
		W	8 位 PWM 0 的周期值								
\$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							
	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_PWM Duty)	00h	R	8 位定时/计数器 PWM0 占空比值								
		W	8 位定时/计数器 0 PWM0 周期值								

(3) \$0026~002D: 中断控制寄存器

地址	功能	(ERST) 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	0	0	0	0
			W	写“1”清除相应中断标志。							
\$0027	中断控制 2 (P_INT_Ctrl2)	00h	R	0	0	ITVALI E	0	0	0	0	0
			W	-	-	ITVALI E	-	-	-	-	-
\$002D	蜂鸣器控制 (P_BUZ_Ctrl)	00h	R	INTIMS 3	INTIMS 2	INTIMS 1	INTIMS 0	BZFS3	BZFS2	BZFS1	BZFS0
			W	INTIMS 3	INTIMS 2	INTIMS 1	INTIMS 0	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	(ERST) 复位值	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
\$0030	系统控制寄存器 (P_SYS_Ctrl)	C0h	R	POR	ERST	LVR	0	WDR	IAR	0	0
			W	写“1”清除相应复位标志							
\$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	-	-	-	-
\$0034	IRQ 设置寄存器 (P_IRQ_Opt1)	00h	R	0	0	0	0	0	0	IRQ0ES	IRQM0
			W	-	-	-	-	-	-	IRQ0ES	IRQM0
\$0035	慢速输出控制 (P_IO_Opt)	00h	R	0	0	0	0	0	0	0	SLOWE
			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) \$0058~\$005F: 端口数据锁存寄存器

地址	功能	(ERST) 复位值	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
\$0058	捕获控制寄存器	00h	R	CAPOP T	0	0	0	0	CAPIP0	0	CAPOE S

	(P_CAP_Ctrl)		W	CAPOP T	-	-	-	-	CAPIPO	-	CAPOE S
\$0059	端口 A 数据锁存寄存器	00h	R	端口 A 数据锁存							
	(P_IOA_Buf)		W	端口 A 数据锁存							
\$005A	端口 B 数据锁存寄存器	00h	R	端口 B 数据锁存							
	(P_IOB_Buf)		W	端口 B 数据锁存							
\$005B	端口 C 数据锁存寄存器	00h	R	0			端口 C 数据锁存				
	(P_IOC_Buf)		W	-			端口 C 数据锁存				
\$005C	端口 D 数据锁存寄存器	00h	R	0			端口 D 数据锁存				
	(P_IOD_Buf)		W	-			端口 D 数据锁存				

5.2.4 系统运行控制寄存器

在 SPMC65P1504A/1502A 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 INTO 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  #$01                                ; LVR= 4V
sta  P_LVR_Opt
sta  P_LVR_Opt

```

5.2.5 芯片配置寄存器

SPMC65P1504A/1502A 提供 3 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被同时写入芯片。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 5.11。

5.3 时钟源

SPMC65P1504A/1502A 支持三种时钟源：晶体/陶瓷振荡器、RC 振荡器和外部时钟源（见图 5-7），可通过设置寄存器 P_MO (\$7FE0) 来选择（详情参见 5.11）。

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

注意：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(FSYS)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

系统时钟频率(Hz), VDD=5.0V	电阻(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

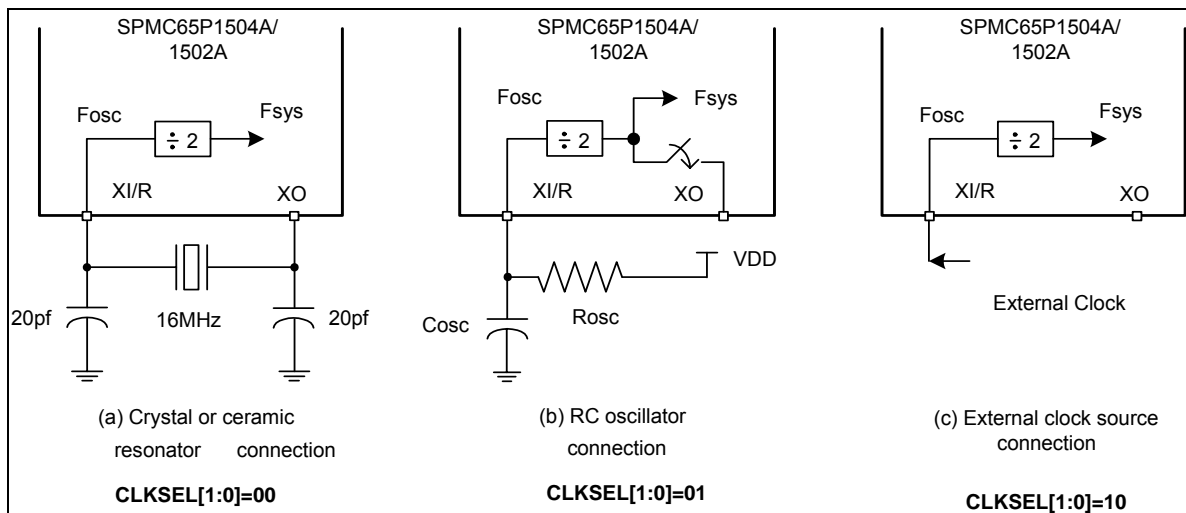


图 5-7 三种时钟源的典型应用电路

5.4 省电模式

5.4.1 简介

SPMC65P1504/1502A 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 SPMC65P1504A /1502A 的模式转换关系。芯片上电复位后直接进入 NORMAL 模式工作，然后可切换到 STOP 模式或 HALT 模式。注意，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。

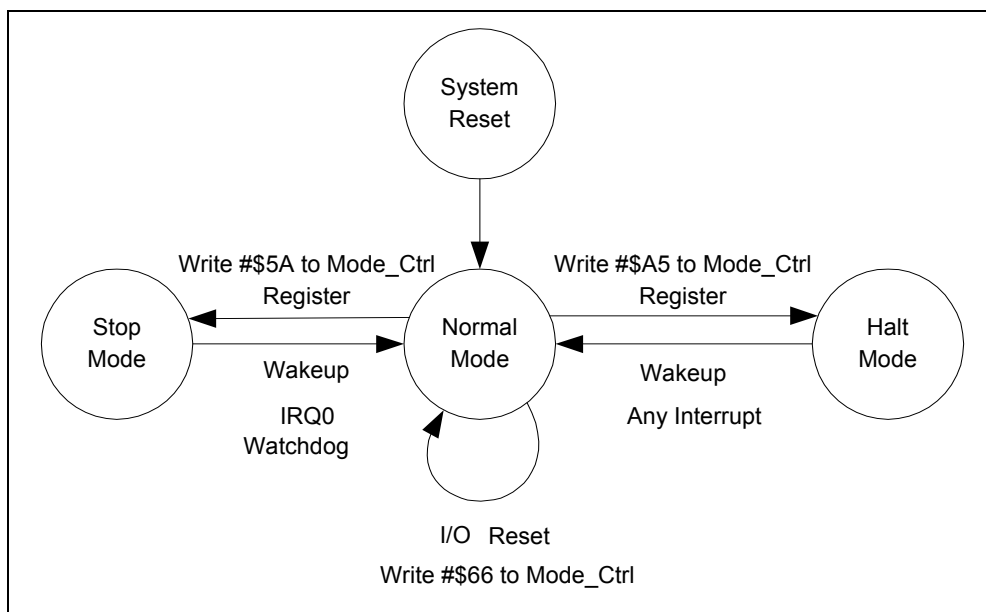


图 5-8 SPMC65P1504A/1502A 的工作模式转换

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（总的中断开关已经被打开：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。

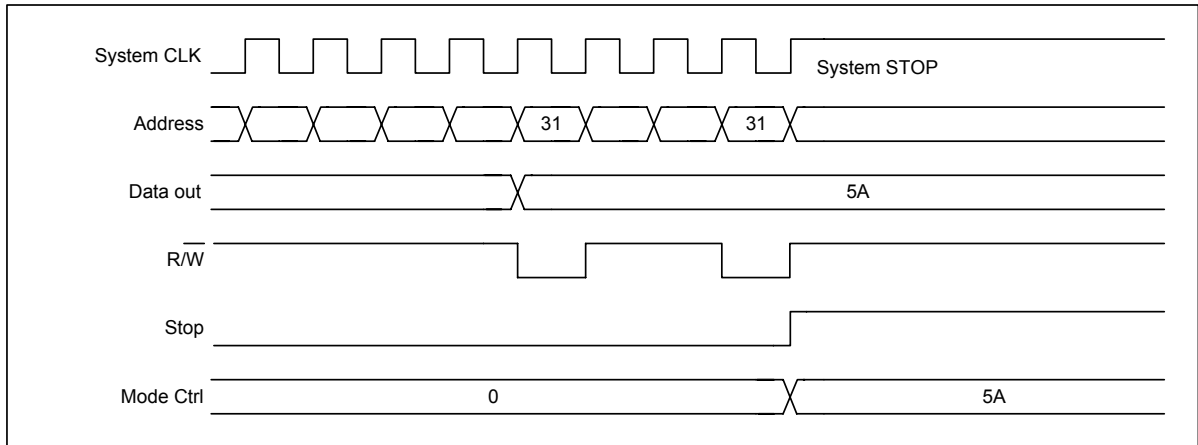


图 5-9 进入 STOP 模式的时序

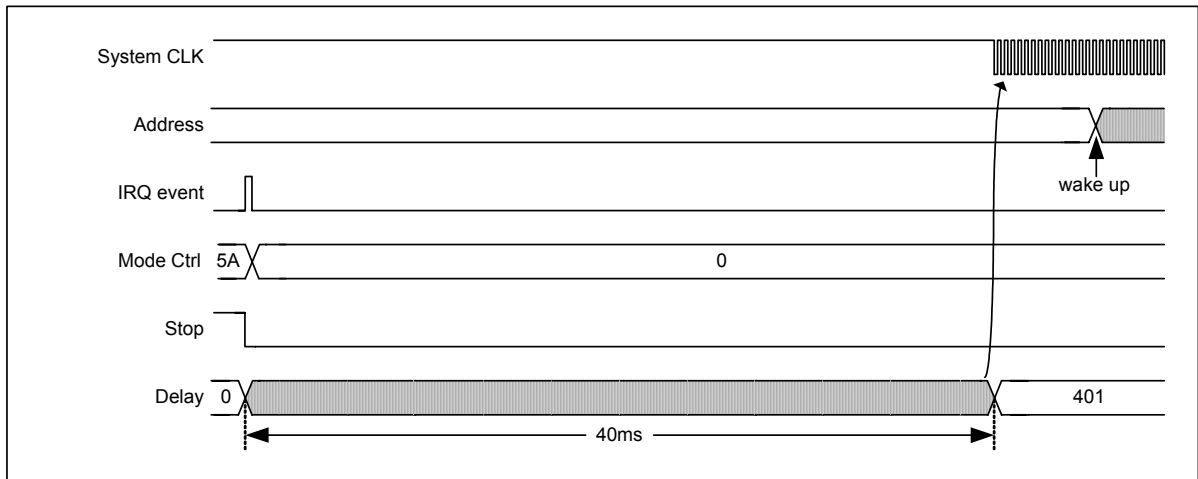


图 5-10 外部中断唤醒 STOP 模式的时序

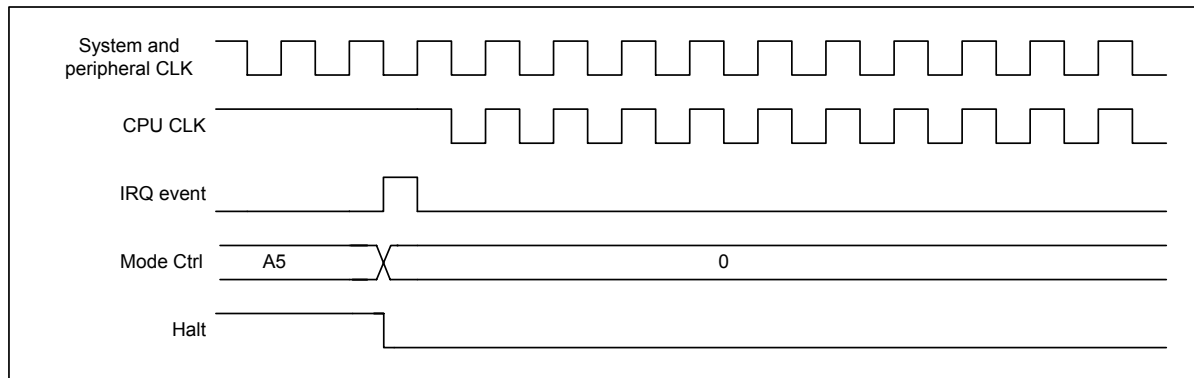


图 5-13 中断唤醒 HALT 模式的时序

■ 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
ACCESS	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	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 简介

SPMC65P1504A/1502A 芯片共有 5 种中断源：外部中断、定时/计数器中断、看门狗中断、时基中断和捕获中断。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

外部中断中可以设置为非屏蔽中断(NMI)。非屏蔽中断源的选择可以通过寄存器\$7FE3 进行设置，除此之外，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

只能设置为边沿触发而非电平触发，以避免中断的频繁产生导致堆栈溢出。

5.5.3 其它中断

除外部中断外，SPMC65P1504A/1502A 还支持其它 4 种中断源：定时/计数器中断、看门狗中断、时基中断和捕获中断。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、时基中断和捕获中断，在后续章节介绍。

表 5-2 中断源列表

中断源		中断状态寄存器	中断控制寄存器	中断源		中断状态寄存器	中断控制寄存器
定时器中断	定时/计数器 0	T0OIF(\$000E.0)	T0OIE(\$000F.0)	外部中断	IRQ0	IRQ0IF(\$000C.0)	IRQ0IE(\$000D.0)
捕获中断	Capture0	CAP0IF(\$000E.6)	CAP0IE(\$000F.6)				
看门狗中断		WDIF(\$000C.6)	WDIE(\$000D.6)	时基中断		ITVALIF(\$0026.5)	ITVALIE(\$0027.5)

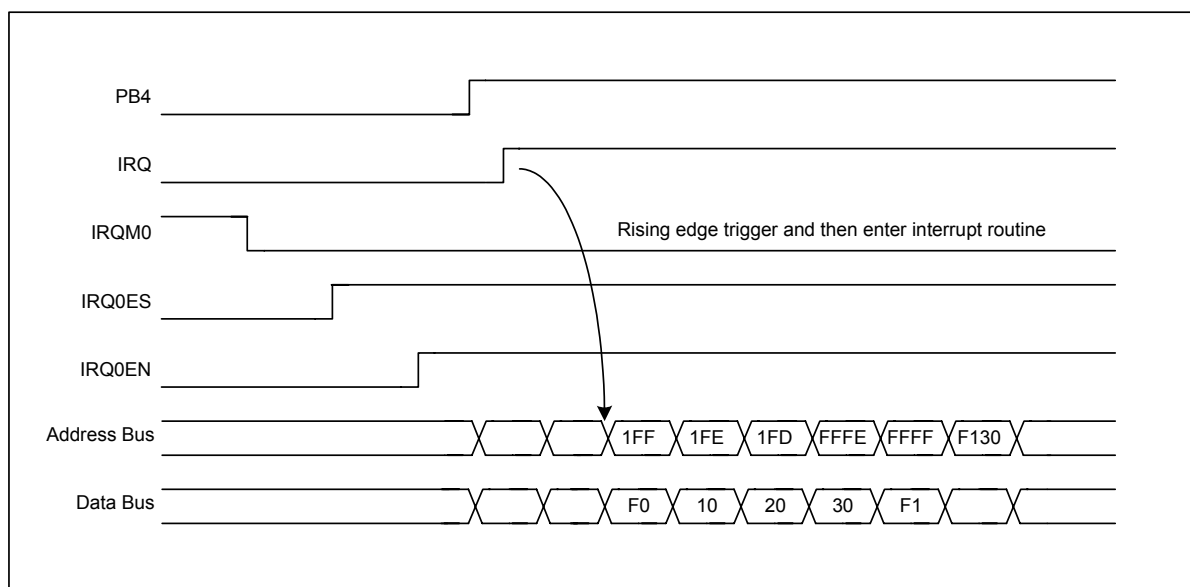


图 5-14 IRQ0 中断触发时序图 (PB4)

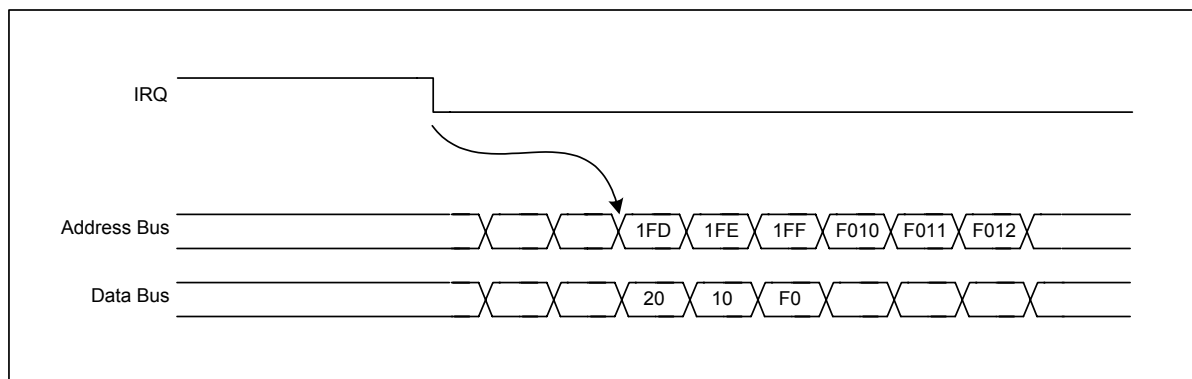


图 5-15 中断返回时序图

5.5.4 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	-	WDIF	-	-	-	-	-	IRQ0IF
ACCESS	-	R/W	-	-	-	-	-	R/W
DEFAULT	0	0	0	0	0	0	0	0

注：向相应位写“1”，则清除中断标志。

Bit 7 保留

Bit [5:1] 保留

Bit 6 **WDIF**: 看门狗中断标志

Bit 0 **IRQ0IF**: IRQ0 中断标志

0 = 中断没有发生

0 = 中断没有发生

1 = 中断发生

1 = 中断发生

(2) 中断标志寄存器 1 (P_INT_Flag1, \$000E)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	-	CAP0IF	-	-	-	-	-	T00IF
ACCESS	-	R/W	-	-	-	-	-	R/W
DEFAULT	0	0	0	0	0	0	0	0

注：向相应位写“1”，则清除中断标志。

Bit 7 保留

Bit [5:1] 保留

Bit 6 **CAP0IF**: 捕获 0 中断标志

Bit 0 **T00IF**: 定时/计数器 0 溢出中断标志

0 = 中断没有发生

0 = 中断没有发生

1 = 中断发生

1 = 中断发生

(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	-	-	-	-	-
ACCESS	-	-	R/W	-	-	-	-	-
DEFAULT	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

0 = 中断没有发生

Bit 5 **ITVALIF**: 时基中断标志

1 = 中断发生

Bit [4:0] 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	-	WDIE	-	-	-	-	-	IRQ0IE
ACCESS	-	R/W	-	-	-	-	-	R/W
DEFAULT	0	0	0	0	0	0	0	0

Bit 7 保留

Bit [5:1] 保留

Bit 6 **WDIE**: 看门狗中断使能位

Bit 0 **IRQ0IE**: IRQ0 中断使能位

0 = 中断禁止

0 = 中断禁止

1 = 中断使能

1 = 中断使能

(5) 中断控制寄存器 1 (P_INT_Ctrl1, \$000F)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	-	CAPOIE	-	-	-	-	-	T0OIE
ACCESS	-	R/W	-	-	-	-	-	R/W
DEFAULT	0	0	0	0	0	0	0	0

Bit 7 保留

Bit [5:1] 保留

Bit 6 **CAPOIE**: 捕获 0 中断使能位

Bit 0 **T0OIE**: 定时/计数器 0 溢出中断使能位

0 = 中断禁止

0 = 中断禁止

1 = 中断使能

1 = 中断使能

(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	-	-	-	-	-
ACCESS	-	-	R/W	-	-	-	-	-
DEFAULT	0	0	0	0	0	0	0	0

Bit [7:6] 保留 0 = 中断禁止

Bit 5 **ITVALIE**: 时基中断使能位 1 = 中断使能

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
ACCESS	-	-	-	-	-	R	R	R
DEFAULT	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 100 = 保留

Bit [2:0] **NMIS [2:0]**: 非屏蔽中断源控制位 011 = 保留

111 = 禁止 010 = 保留

110 = 保留 001 = 保留

101 = 保留 000 = PB4 (Iirq0)作为 NMI 中断源

(8) IRQ 设置寄存器 1 (P_IRQ_Opt1, \$0034)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	-	-	-	-	-	-	IRQ0ES	IRQM0
ACCESS	-	-	-	-	-	-	R/W	R/W
DEFAULT	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 [7:2] 保留 0 = 低电平触发

Bit 1 **IRQ0ES**: IRQ0 极性控制位 1 = 高电平触发

IRQM0="0" Bit 0 **IRQM0**: RQ0 触发模式选择位

0 = 下降沿触发 0 = 边沿触发

1 = 上升沿触发 1 = 电平触发

IRQM0="1"

[例] 5.5.1 使能 IRQ0 中断，上升沿触发

```

set    P_IRQ_Opt1, CB_IRQOpt1_IRQ0ES           ; 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 IRQ0
    
```

5.6 复位

5.6.1 简介

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。而对于 SPMC65X 系列芯片，则只需将 RESETB 管脚直接拉高（或接一个上拉电阻）到 VDD，不需外接 RC 电路，从而不会造成上电延迟。

5.6.3 外部复位 (ERST)

SPMC65P1504A/1502A 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 5-16 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 $0.3 \times VDD$ 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滤去，从而系统不会被复位。外部复位信

号有效后，系统发生复位。当 RESETB 上升为高电平后，并再经过 40ms 和 1024 个系统时钟周期，程序才重新开始执行，即复位结束。复位时序见图 5-18。

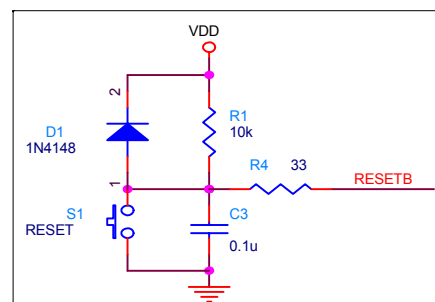


图 5-16 复位电路

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。这样，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并持续 1024

一个系统时钟周期，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，程序将重新开始执行。

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 中的值。另外，低电压复位信号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。这种情况下，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(P_MO, \$7FE0)使能或禁止。

看门狗电路内置独立的 RC 振荡器。可通过寄存器 P_WDT_Ctrl[6:4]对看门狗中断频率进行设置。复位信

号是由看门狗中断频率经过 8 分频后产生的。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 (Software)

除了以上几种复位方式之外，SPMC65P1504A/1502A 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 复位	Yes	Yes	Yes	Yes	Yes	No
外围复位	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	----vvvv	----vvvv	----vvvv	----xxxx	----xxxx	----vvvv
P_IOD_Attrib	-----vvv	-----vvv	-----vvv	-----xxx	-----xxx	-----vvv
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\$7FE2 的设置；x无影响；-表示保留位

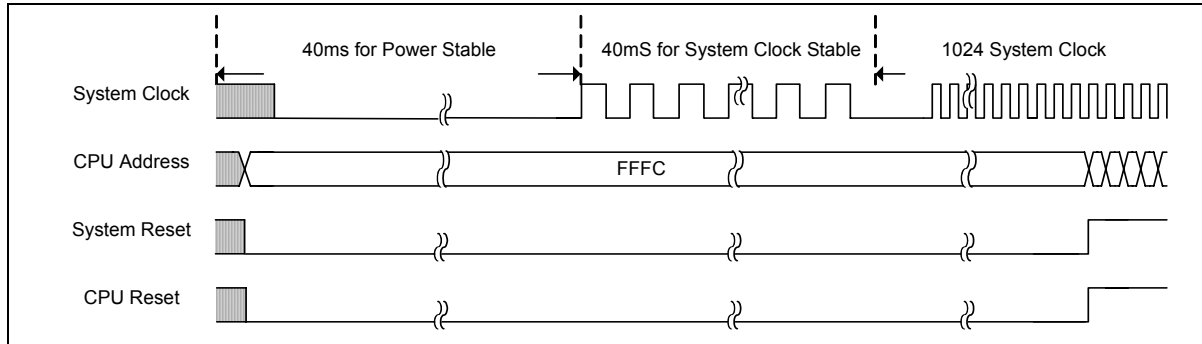


图 5-17 上电复位时序

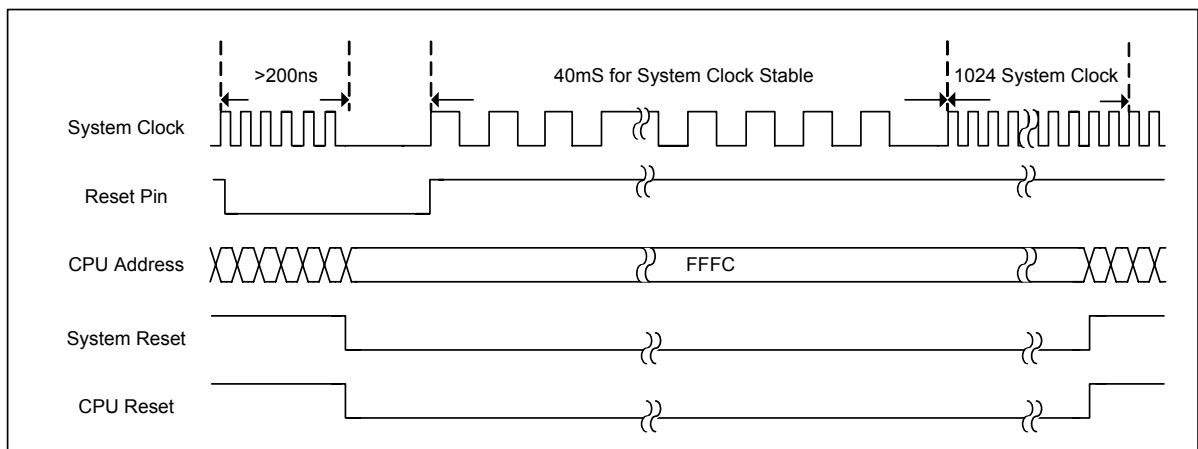


图 5-18 外部复位时序

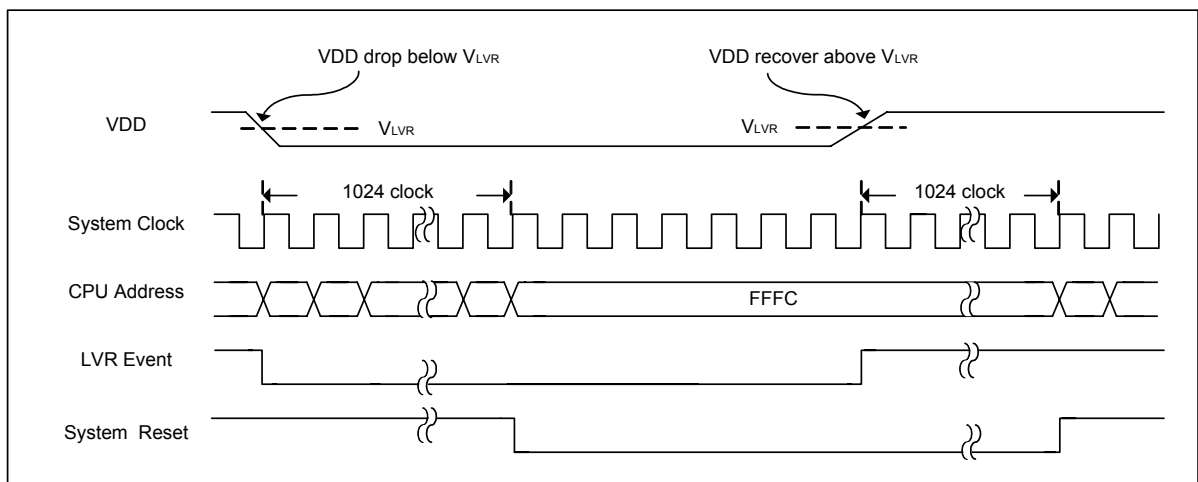


图 5-19 低电压复位时序

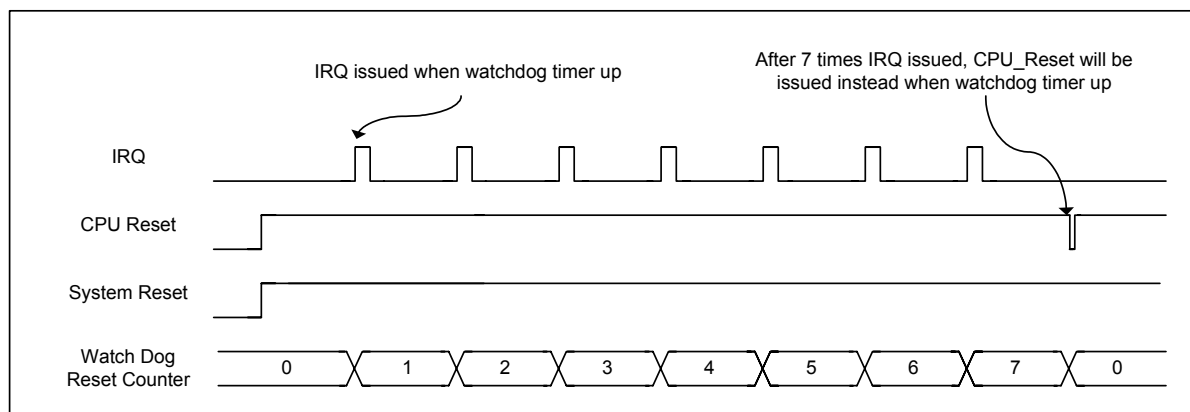


图 5-20 看门狗复位时序

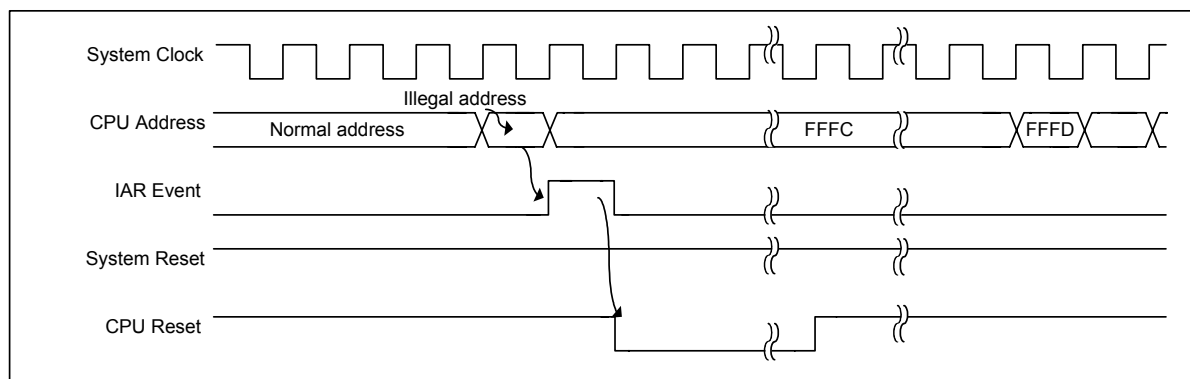


图 5-21 非法地址复位时序

(1) 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	-	-
ACCESS	R/W	R/W	R/W	-	R/W	R/W	-	-
DEFAULT	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

用于表示系统侦测到非法地址时，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

注意：向相应位写“1”清除该标志。

(2) 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
ACCESS	-	-	-	-	-	-	-	R/W
DEFAULT	0	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 [7:1] 保留

1 = 低于 4.0V 复位

Bit 0 **LVRV40**: 复位电压选择位

0 = 低于 2.5V 复位

注意：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 简介

SPMC65P1504A /1502A 共有 4 组 IO 端口：端口 A、端口 B、端口 C、端口 D。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, 必须对数据锁存寄存器 **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。设置不同的输入属性(带上拉电阻输入和带下拉电阻输入)时, 端口的电平状态会不同。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 端口设置

方向(P_IOX_Dir)	属性(P_IOX_Attrib)	数据(P_IOX_Data)	功能	描述
0	0	0	下拉 (WPD)	带下拉电阻的输入管脚
0	0	1	上拉 (WPU)	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 是一个专门用于存储端口(PX)数据的寄存器, 向寄存器 P_IOX_Buf 或者 P_IOX_Data 中写数据时, 效果相同, 而读 P_IOX_Data 寄存器将得到管脚的电平状态。因此, 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。所以强烈建议只对寄存器 P_IOX_Buf 进行位操作。如果要获得外部管脚的状态, 只能读取 P_IOX_Data 获得。

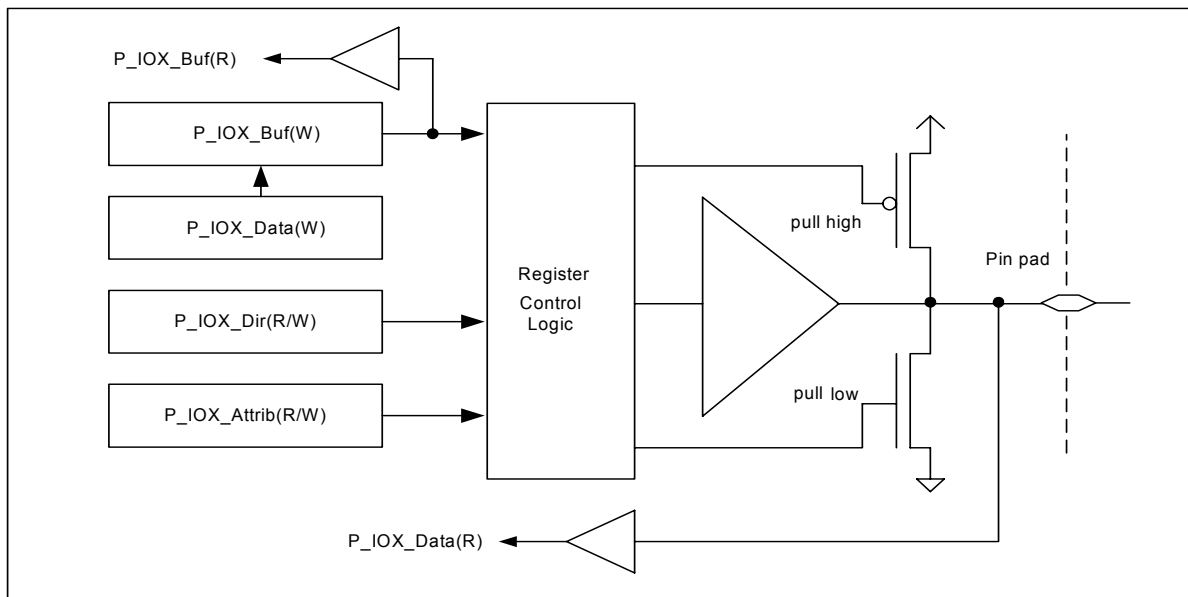


图 5-22 典型的 IO 内部结构图

5.7.2 端口 A

端口 A 有 8 个双向可编程 IO 管脚，并且各管脚均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。

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/输出管脚
PA1	Bit1	通用输入/输出管脚
PA2	Bit2	通用输入/输出管脚
PA3	Bit3	通用输入/输出管脚
PA4	Bit4	通用输入/输出管脚
PA5	Bit5	通用输入/输出管脚
PA6	Bit6	通用输入/输出管脚
PA7	Bit7	通用输入/输出管脚

(1) 端口 A 数据寄存器(P_IOA_Data, \$0000)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOA_Data							
地址	R/W							
默认值	00h							

Bit [7:0] **P_IOA_Data**: 端口 A 数据寄存器。向该寄存器写入数据，会同时保存到寄存器 P_IOA_Buf 中

(2) 端口 A 数据锁存器(P_IOA_Buf, \$0059)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOA_Buf							
地址	R/W							
默认值	00h							

Bit [7:0] **P_IOA_Buf**: 端口 A 数据锁存器

(3) 端口 A 方向寄存器(P_IOA_Dir, \$0004)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOA_Dir							
地址	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

(4) 端口 A 属性寄存器(P_IOA_Attrib, \$0008)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOA_Attrib							
地址	R/W							
默认值	00h							

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

0 = 非悬浮式输入/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    #$FF                                ;存$FF 到累加器
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                                ;存$00 到累加器
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，并且各管脚均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。

PB6 和 PB7 可以作为慢速输出管脚。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]的输出信号从高电平变低电平时，将会有约 250ns 的延迟（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

端口 B	BIT	复用功能
PB0	Bit0	通用输入/输出管脚、捕获器 0 输入、定时/计数器 0 外部时钟输入
PB1	Bit1	通用输入/输出管脚

端口 B	BIT	复用功能
PB2	Bit2	通用输入/输出管脚、定时/计数器 0 比较输出、PWM 输出
PB3	Bit3	通用输入/输出管脚
PB4	Bit4	通用输入/输出管脚、外部中断 0
PB5	Bit5	通用输入/输出管脚
PB6	Bit6	通用输入/输出管脚(慢速输出)、蜂鸣器输出
PB7	Bit7	通用输入/输出管脚(慢速输出)

(1) 端口 B 数据寄存器 (P_IOB_Data, \$0001)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOB_Data							
地址	R/W							
默认值	00h							

Bit [7:0] **P_IOB_Data**: 端口 B 数据寄存器。向该寄存器写入数据，会同时保存到寄存器 P_IOB_Buf 中。

(2) 端口 B 数据锁存器(P_IOB_Buf, \$005A)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOB_Buf							
地址	R/W							
默认值	00h							

Bit [7:0] **P_IOB_Buf**: 端口 B 数据锁存器

(3) 端口 B 方向寄存器(P_IOB_Dir, \$0005)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOB_Dir							
地址	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

(4) 端口 B 属性寄存器(P_IOB_Attrib, \$0009)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	P_IOB_Attrib							
地址	R/W							
默认值	00h							

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

0 = 非悬浮式输入/输出

1 = 悬浮式输入

(5) 慢速输出控制寄存器((P_IO_Opt, \$0035)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	-	-	-	-	-	-	-	SLOWE
地址	-	-	-	-	-	-	-	R/W
默认值	0	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 [7:1] 保留

0 = 关闭 PB[7:6]的慢速输出功能

Bit 0 **SLOWE**: 慢速输出功能使能

1 = 开启 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                                ;存$0F 到累加器
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    #$FF                                ;存$FF 到累加器
sta    P_IOB_Attrib
```

5.7.4 端口 C

端口 C 是一个通用 IO 端口，有 4 个双向可编程 IO 管脚，并且各管脚均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 写入数据，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。

表 5-7 端口 C 各管脚功能列表

端口 C	BIT	复用功能
PC0	Bit0	通用输入/输出管脚
PC1	Bit1	通用输入/输出管脚
PC2	Bit2	通用输入/输出管脚
PC3	Bit3	通用输入/输出管脚

(1) 端口 C 数据寄存器 (P_IOC_Data, \$0002)

BIT	-	-	-	-	Bit3	Bit2	Bit1	Bit0
名称	P_IOC_Data							
地址	R/W							
默认值	00h							

Bit [3:0] **P_IOC_Data**: 端口 C 数据寄存器。向该寄存器写入数据，会同时保存到寄存器 P_IOC_Buf 中。

(2) 端口 C 数据锁存器(P_IOC_Buf, \$005B)

BIT	-	-	-	-	Bit3	Bit2	Bit1	Bit0
名称	P_IOC_Buf							
地址	R/W							
默认值	00h							

Bit [3:0] **P_IOC_Buf**: 端口 C 数据锁存器

(3) 端口 C 方向寄存器(P_IOC_Dir, \$0006)

BIT	-	-	-	-	Bit3	Bit2	Bit1	Bit0
名称	P_IOC_Dir							
地址	R/W							
默认值	00h							

Bit [3:0] **P_IOC_Dir**: 端口 C 方向寄存器

0 = 输入

1 = 输出

(4) 端口 C 属性寄存器(P_IOC_Attrib, \$000A)

BIT	-	-	-	-	Bit3	Bit2	Bit1	Bit0
名称	P_IOC_Attrib							
地址	R/W							
默认值	00h							

Bit [3:0] **P_IOC_Attrib**: 端口 C 属性寄存器

0 = 非悬浮式输入/输出

1 = 悬浮式输入

[例] 5.7.5 设置端口 C[3:0] 为低电平输出。

```
lda    #$0F                                ; 存$0F 到累加器
sta    P_IOC_Dir
lda    #$00
sta    P_IOC_Attrib
sta    P_IOC_Data
```

[例] 5.7.6 设置端口 C[3:0] 为悬浮式输入。

```
lda    #$0F                                ; 存$0F 到累加器
sta    P_IOC_Attrib
```

5.7.5 端口 D

端口 D 是一个通用 IO 端口, 有 3 个双向可编程 IO 管脚, 并且各管脚均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 将得到端口 D 各管脚的电平状态, 向 P_IOD_Data 写入数据, 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。

表 5-8 端口 D 各管脚功能列表

端口 D	BIT	复用功能
PD0	Bit0	通用输入/输出管脚
PD1	Bit1	通用输入/输出管脚
PD2	Bit2	通用输入/输出管脚

(1) 端口 D 数据寄存器 (P_IOD_Data, \$0003)

BIT	-	-	-	-	-	Bit2	Bit1	Bit0
名称	P_IOD_Data							
地址	R/W							
默认值	00h							

Bit [2:0] **P_IOD_Data**: 端口 D 数据寄存器。向该寄存器写入数据，会同时保存到寄存器 P_IOD_Buf 中。

(2) 端口 D 数据锁存器(P_IOD_Buf, \$005C)

BIT	-	-	-	-	-	Bit2	Bit1	Bit0
名称	P_IOD_Buf							
地址	R/W							
默认值	00h							

Bit [2:0] **P_IOD_Buf**: 端口 D 数据锁存器

(3) 端口 D 方向寄存器(P_IOD_Dir, \$0007)

BIT	-	-	-	-	-	Bit2	Bit1	Bit0
名称	P_IOD_Dir							
地址	R/W							
默认值	00h							

Bit [2:0] **P_IOD_Dir**: 端口 D 方向寄存器

0 = 输入

1 = 输出

(4) 端口 D 属性寄存器(P_IOD_Attrib, \$000B)

BIT	-	-	-	-	-	Bit2	Bit1	Bit0
名称	P_IOD_Attrib							
地址	R/W							
默认值	00h							

Bit [2:0] **P_IOD_Attrib**: 端口 D 属性寄存器

0 = 非悬浮式输入/输出

1 = 悬浮式输入

[例] 5.7.7 设置端口 D[2:0] 为高电平输出

```

lda    #$07                                ; 存$07 到累加器
sta    P_IOD_Dir
lda    #$00
sta    P_IOD_Attrib
lda    #$07
sta    P_IOD_Data
    
```

5.8 定时/计数器

5.8.1 简介

SPMC65P1504A 和 SPMC65P1502A 这两款芯片均配有 1 个 16 位递增定时/计数器，定时/计数器 0。定时/计数器 0 具备强大的 CCP(Capture/Compare/PWM)功能，即捕获、比较和 PWM 输出功能。表 5-9 为定时/

计数器 0 的功能简介。这些功能可通过相应控制寄存器进行设置。

表 5-9 SPMC65P1504A/1502A 定时/计数器 0 功能简介

	定时/事件计数器		捕获器		比较器		PWM
	8 位	16 位	8 位	16 位	8 位	16 位	
Timer 0	YES	YES	脉宽/周期	脉宽	YES	YES	8 位

5.8.2 定时/计数器 0

定时/计数器 0 可设为 16 位和 8 位两种工作方式。设为 8 位时，计数最大值为 255。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，定时/计数器 0 中断标志被置位。同时计数器重载计数初值。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。设为 16 位时，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。

一个 8 位 CPU，其总线宽度为 8 位，通常无法直接访问 16 位的数据。为了克服这样的局限，在芯片中设计了一个有专门的读/写缓冲器的数据寄存器 Timer (MSB)，用于 16 位数据高 8 位的读写操作（如图 5-23 所示）。在读取 16 位数据时，用户需要先读出低字节（与此同时，高字节被自动锁存在缓冲器里），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
- 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

16 位定时/计数器的读写操作、内部结构、计数溢出及中断时序，分别如图 5-23、图 5-24、图 5-25、图 5-26、所示。

【例】5.8.1 8 位定时/计数器 0 溢出频率计算公式

$$f_{T0} = \frac{F_{sys}}{Timer_prescaler \times (65536 - Timer_preload_value)}$$

f_{T0} : 定时 / 计数器溢出频率

【例】5.8.2 16 位定时/计数器 0 溢出频率计算公式

$$f_{T0} = \frac{F_{sys}}{Timer_prescaler \times (65536 - Timer_preload_value)}$$

f_{T0} : 定时 / 计数器溢出频率

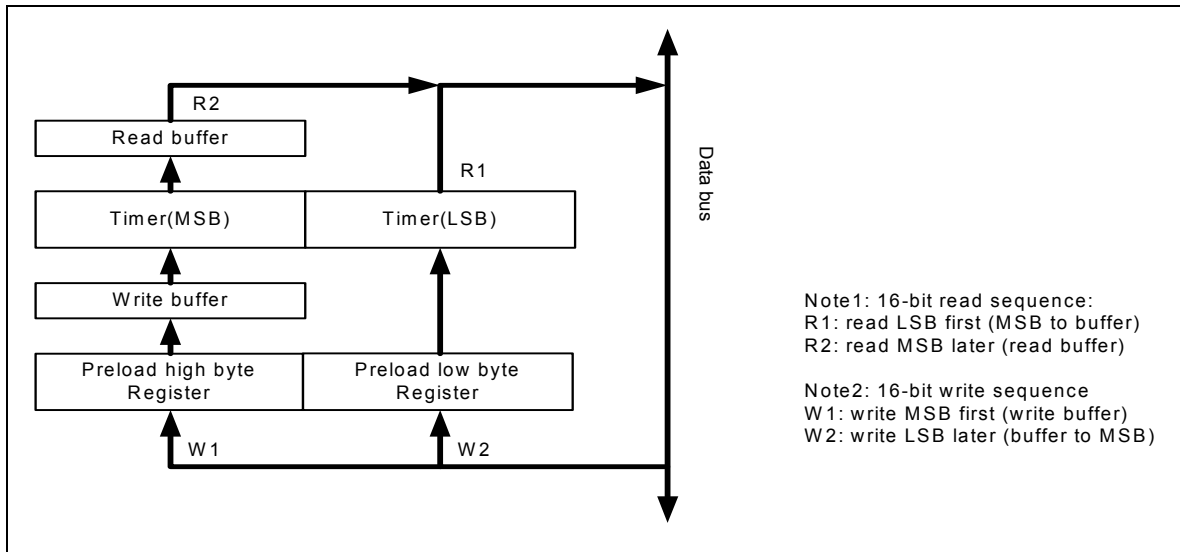


图 5-23 16 位定时/计数器的读写操作

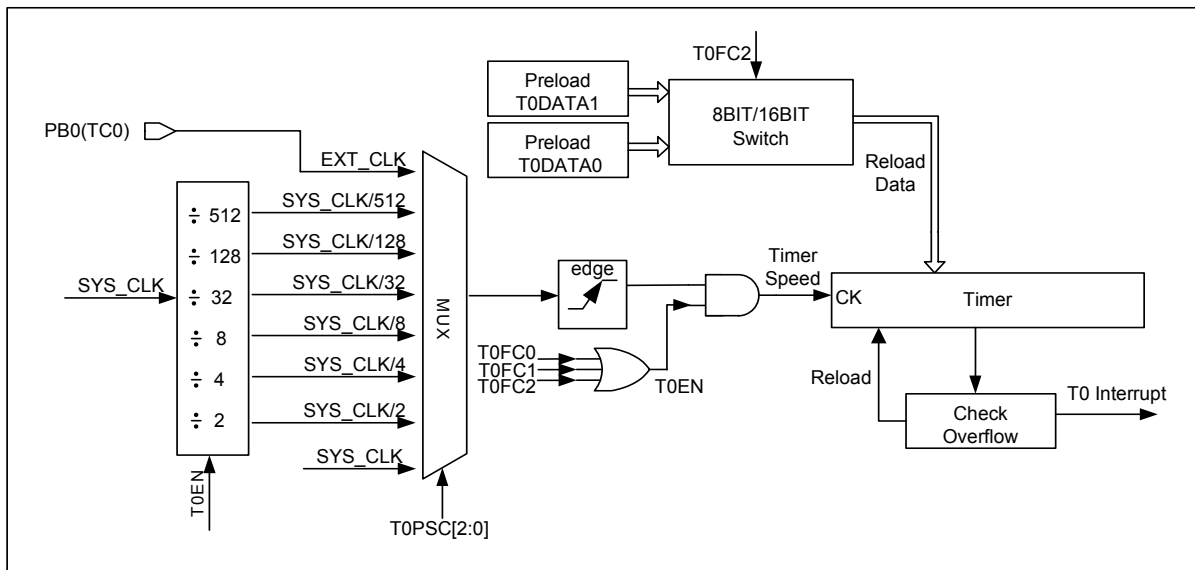


图 5-24 16 位定时/计数器 0 的内部结构

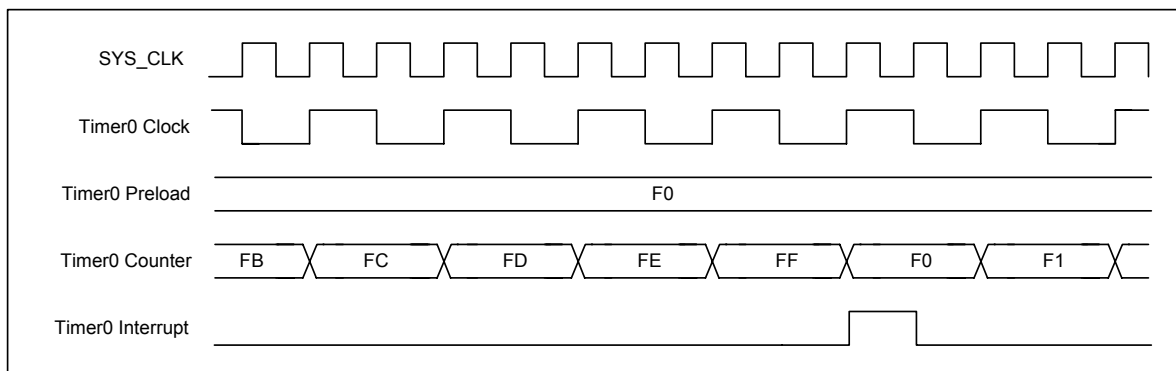


图 5-25 8 位定时/计数器 0 计数溢出及中断时序

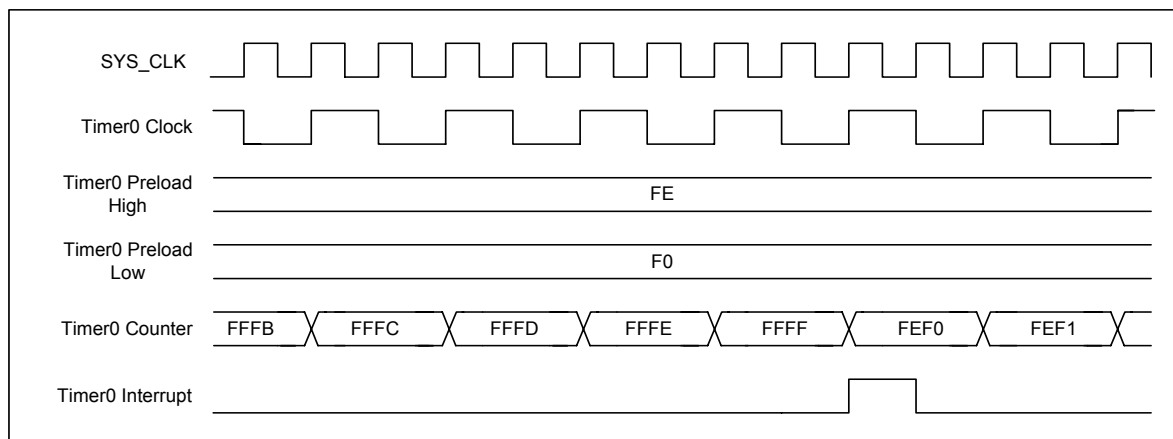


图 5-26 16 位定时/计数器 0 计数溢出及中断时序

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

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	-	-	-	-	-	T0FC2	T0FC1	T0FC0
ACCESS	-	-	-	-	-	R/W	R/W	R/W
DEFAULT	0	0	0	0	0	0	0	0

Bit [7:3] 保留 100 = 16 位定时/计数器

Bit [2:0] **T0FC**[2:0]: 定时/计数器 0 功能设置位 011 = 8 位脉宽/周期捕获

111 = 8 位 PWM 010 = 8 位比较器

110 = 16 位脉宽捕获 001 = 8 位定时/计数器

101 = 16 位比较器 000 = 禁止

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

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	-	-	-	-	-	T0PSC2	T0PSC1	T0PSC0
ACCESS	-	-	-	-	-	R/W	R/W	R/W
DEFAULT	0	0	0	0	0	0	0	0

Bit [7:3] 保留 100 = $F_{SYS} \div 32$

Bit [2:0] **T0PSC**[2:0]: 定时/计数器 0 预分频设置位 011 = $F_{SYS} \div 8$

111 = 外部时钟频率输入 010 = $F_{SYS} \div 4$

110 = $F_{SYS} \div 512$ 001 = $F_{SYS} \div 2$

101 = $F_{SYS} \div 128$ 000 = F_{SYS}

FSYS: 系统时钟频率

(3) 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 位 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 位比较器	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 位捕获模式	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 位 PWM 模式	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
初始值		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

写: T0PLV_[7:0] 定时/计数器 0 重载值

8/16 位定时/计数模式:

读: T0CWV_[7:0] 定时/计数器 0 脉宽捕获值

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

8/16 位比较器模式

8 位 PWM 模式

写: T0COV_[7:0] 定时/计数器 0 比较值

写: T0PPV_[7:0] 定时/计数器 0 PWM 周期值

读: T0R_[7:0] 定时/计数器 0 计数值

读: T0PPV_[7:0] 定时/计数器 0 PWM 周期值

8/16 位捕获模式

(4) 定时/计数器 0 计数寄存器(P_TMR0_CountHi, \$0014)

Bit		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
16 位定时/计数模式	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 位比较器模式	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 位捕获模式	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 位捕获模式	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

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
8 位 PWM 模式	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
DEFAULT	0	0	0	0	0	0	0	0	

Bit [7:0] **P_TMR0_CountHi** [7:0]: 定时/计数器 0 计数寄存器的高字节

16 位定时/计数模式

写: T0PLV_[15:8] 定时/计数器 0 重载值

读: T0R_[15:8] 定时/计数器 0 计数值

16 位比较器位模式

写: T0COV_[15:8] 定时/计数器 0 比较值

读: T0R_[15:8] 定时/计数器 0 计数值

16 位捕获模式

写: T0PLV_[15:8] 定时/计数器 0 重载值

读: T0CWV_[15:8] 定时/计数器 0 脉宽捕获

值

8 位捕获模式

写: T0PLV_[15:8] 定时/计数器 0 重载值

读: T0CCV_[7:0] 定时/计数器 0 周期捕获值

8 位 PWM 位模式

写: T0PDV_[7:0] 定时/计数器 0 PWM 占空比值

读: T0PDV_[7:0] 定时/计数器 0 PWM 占空比值

[例 5.8.3] 设定定时/计数器 0 为 8 位定时方式，定时 1ms。

```

lda    #6                ; 启动定时/计数器 0 之前，首先设置计数初值
sta    P_TMR0_Preload
lda    #C_T0FCS_Div_32   ; 设置定时/计数器 0 时钟源为 Fsys/32
sta    P_TMR0_1_Ctrl1
lda    #C_T08B_Timer     ; 设置定时/计数器 0 为 8 位定时方式
sta    P_TMR0_1_Ctrl0
lda    #6                ; 设置计数重载值= 256-6= 250
sta    P_TMR0_Preload    ; Fsys(8MHz)/32/250= 1KHz(1ms)

```

[例 5.8.4] 设定定时/计数器 0 为 8 位比较模式

```

lda    #156              ; 启动定时/计数器 0 之前，首先设置计数初值
sta    P_TMR0_Preload
lda    #C_T0FCS_Div_128  ; 设置定时/计数器 0 时钟源为 Fsys/128
sta    P_TMR0_1_Ctrl1
lda    #C_T08B_COMP      ; 设置定时/计数器 0 为 8 位比较输出模式
sta    P_TMR0_1_Ctrl0
lda    #156              ; 设置计数重载值 = 256-156= 100
sta    P_TMR0_Preload    ; 设置管脚 PB2 的比较频率为 Fsys(8MHz)/128/100= 625Hz

```

[例] 5.8.5 将定时/计数器 0 设置为 16 位工作方式，定时 10ms。

```

lda    #253                ; 启动定时/计数器 0 之前，首先设置计数初值
sta    P_TMR0_PreloadHi   ; 首先设置高字节初始值（重载值）
lda    #143
sta    P_TMR0_Preload     ; 然后再设置低字节初始值（重载值）
lda    #C_T0FCS_Div_128   ; 设置定时/计数器 0 时钟源为 Fsys/128
sta    P_TMR0_1_Ctrl1
lda    #C_T016B_Timer     ; 将定时/计数器 0 设置为 16 位工作方式
sta    P_TMR0_1_Ctrl0
lda    #253                ; 首先设定计数重载值高字节为 2x256= 512
sta    P_TMR0_PreloadHi   ; 然后设定计数重载值低字节为 256-143= 113
lda    #143
sta    P_TMR0_Preload     ; 设定 Fsys(8Mhz)/128/625= 100Hz(10ms)

```

5.9 Capture/Compare/PWM (CCP)

SPMC65P1504A/1502A 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 5 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。

表 5-10 CCP 模式 – 在各定时/计数器中应用情况

CCP 功能	定时/计数器
8 位比较模式	定时/计数器 0
16 位比较模式	定时/计数器 0
8 位捕获模式	定时/计数器 0
16 位捕获模式	定时/计数器 0
8 位 PWM 模式	定时/计数器 0

5.9.1 8 位比较模式

定时/计数器 0 支持 8 位比较模式。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，当计数器发生溢出时，PB2 管脚输出的电平翻转。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，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。

[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

5.9.2 16 位比较模式

定时/计数器 0 支持 16 位比较输出。定时/计数器 0 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_TMR0_Comp (\$0013)、P_TMR0_CompHi (\$0014) 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，PB2 管脚输出的电平翻转。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 16 位比较输出的时序，见图 5-27。

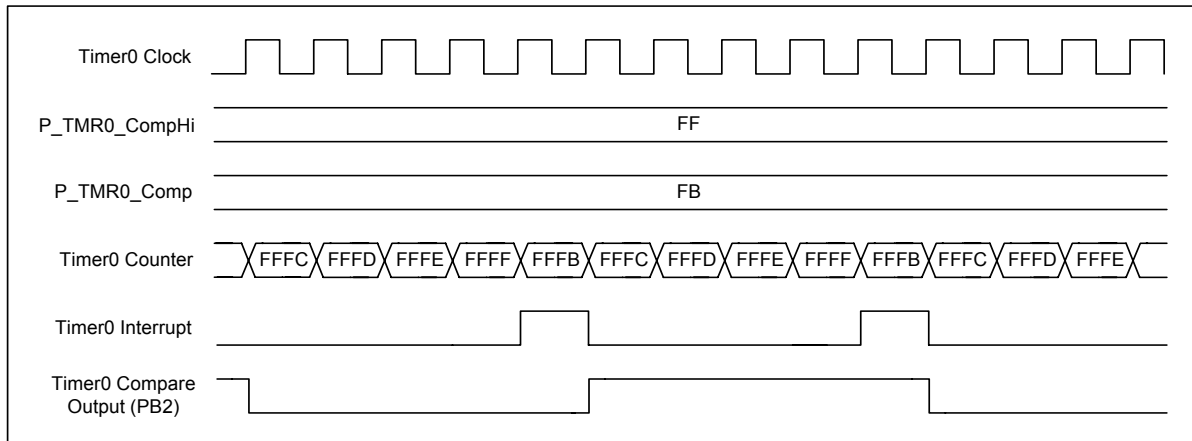


图 5-27 16 位比较输出的时序

[例] 5.9.2 设置定时/计数器 0 为 16 位比较输出模式

```

lda    #253                ; 启动定时/计数器 0 之前，首先设置计数初值
sta    P_TMR0_PreloadHi   ; 设置计数初值的高字节
lda    #143
sta    P_TMR0_Preload     ; 设置计数初值的低字节
lda    #C_T0FCS_Div_128   ; 设置定时/计数器 0 时钟源为 Fsys/128
sta    P_TMR0_1_Ctrl
lda    #C_T016B_COMP      ; 设置定时/计数器 0 为 16 位比较模式
sta    P_TMR0_1_Ctrl0
lda    #253                ; 设置计数重载值的高字节 = 2x 256 = 512
sta    P_TMR0_PreloadHi
lda    #143                ; 设置计数重载值的低字节 = 256 - 143 = 113
sta    P_TMR0_Preload     ; 设置管脚 PB2 的比较频率为
                                ; Fsys(8Mhz)/128/625 = 100Hz(10ms)

```

5.9.3 8 位捕获模式

SPMC65P1504A/1502A 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过捕获控制寄存器(P_CAP_Ctrl, \$0058)来设置。

定时/计数器 0 是 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 5-28 所示。

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

```

捕获脉宽值 = (P_TMR0_Cap + 1) × 计数器分频系数
捕获周期值 = (P_TMR0_CapCycle8 + 1) × 计数器分频系数

```

■ 捕获控制寄存器 (P_CAP_Ctrl, \$0058)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	CAPOPT	-	-	-	-	CAPIP0	-	CAP0ES
ACCESS	R/W	-	-	-	-	R/W	-	R/W
DEFAULT	0	0	0	0	0	0	0	0

- Bit 7: **CAPOPT**: 捕获数据保持选择位
 0 =与捕获器 0 计数触发沿极性相反
 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 [6:3] 保留
- 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
- Bit 1: 保留
- 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)

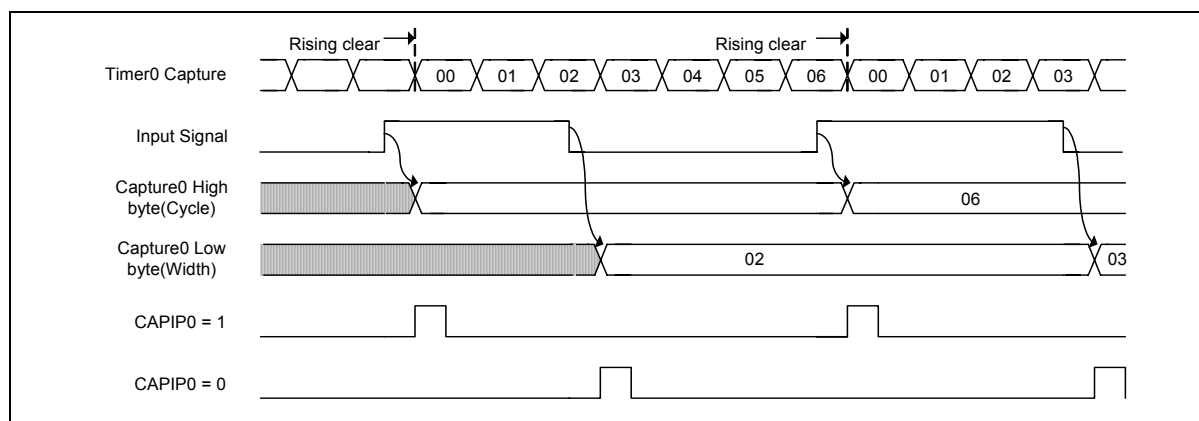


图 5-28 8 位捕获的工作时序

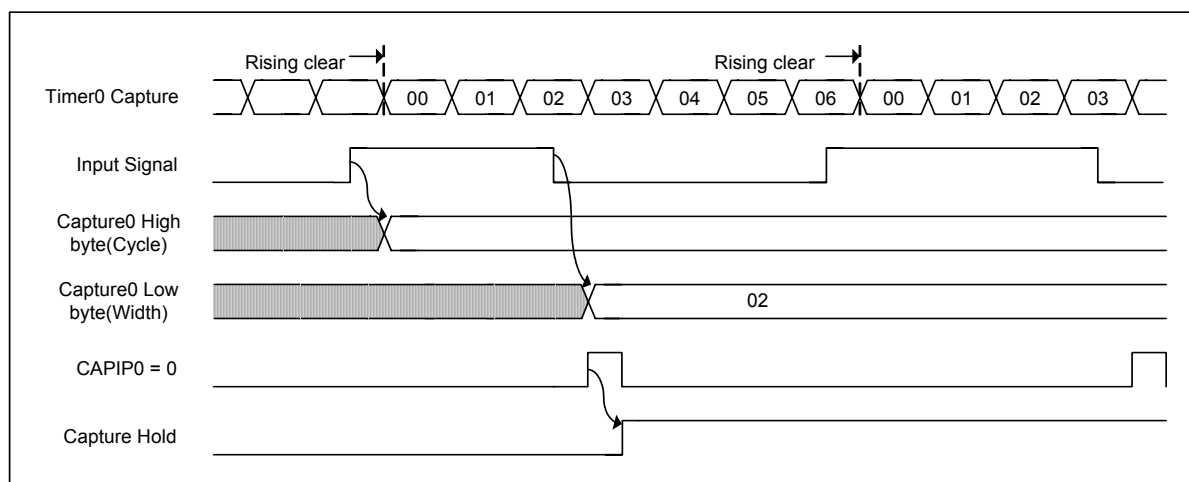


图 5-29 8 位捕获及数据保持的工作时序

[例] 5.9.4 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 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 CAP0 捕获中断
sta P_CAP_Ctrl           ; 测量 PB0 管脚输入的低电平脉宽
    
```

5.9.4 16 位模式

定时/计数器 0 支持 16 位脉宽捕获输出, 不能进行周期捕获。16 位脉宽捕获与 8 位脉宽捕获的操作相同。

捕获脉宽值 = ((P_TMR0_CapHi, P_TMR0_Cap) + 1) × 计数器分频系数

[例] 5.9.5 定时/计数器 0 的 16 位捕获计算公式

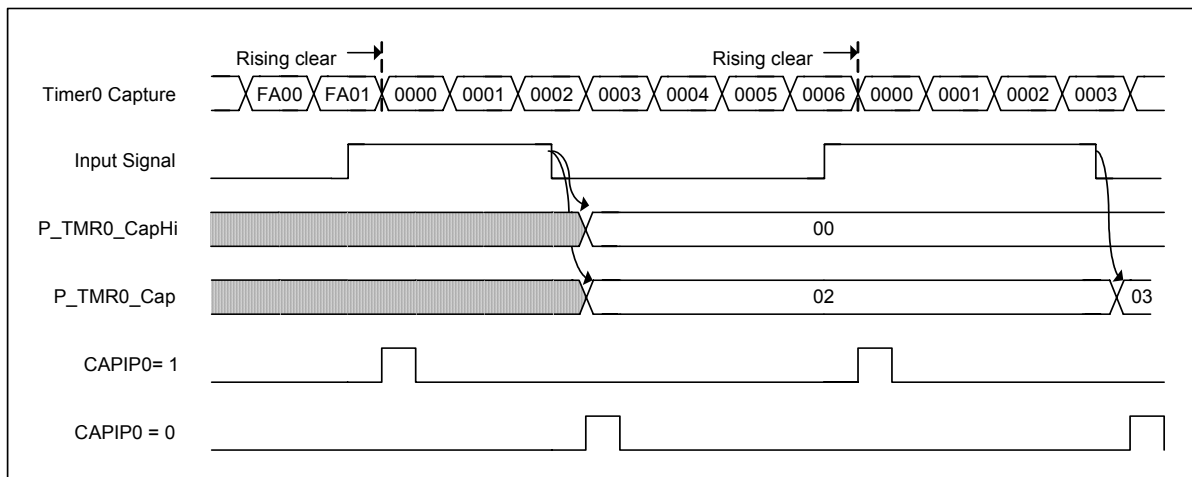


图 5-30 16 位捕获的工作时序

[例] 5.9.6 将定时/计数器 0 设置为 16 位捕获模式

```

lda #C_T0FCS_Div_128     ; 设置定时/计数器 0 时钟源为 Fsys/128 (测量范围 : 62.5KHz~1Hz)
sta P_TMR0_1_Ctrl1
lda #C_T016B_CAP         ; 设置定时/计数器 0 为 16 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 8 位 PWM 模式

SPMC65P1504A/1502A 这两款芯片的定时/计数器可

设置为 PWM 输出模式。下面说明在 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 波形的输出管脚。PWM 周期在寄存器 P_TMR0_PWMPeriod[7:0]中设置; PWM 占空比在寄存器 P_TMR0_PWMduty[7:0]中设置。

[例] 5.9.7 定时/计数器 0 的 8 位 PWM 模式计算公式

$$\text{PWM 周期值} = \text{P_TMR0_PWMPeriod [7:0]} \times \text{计数器分频系数}$$

$$\text{PWM Duty} = \text{P_TMR0_PWMduty [7:0]} \times \text{计数器分频系数}$$

表 1-1列出了 PWM 精度与输出频率的关系, 供用户参考。

表 5-11 不同精度和分频系数下 PWM 的几种输出频率

PWM 精度	PWM 输出频率		
	分频= Fsys ÷1	分频= Fsys ÷2	分频= Fsys ÷4
8 位 PWM	31.25 KHZ	15.6 KHZ	7.8 KHZ
7 位 PWM	62.5 KHZ	31.25 KHZ	15.6 KHZ
6 位 PWM	125 KHZ	62.5 KHZ	31.25 KHZ
5 位 PWM	250KHZ	125 KHZ	62.5 KHZ

注：系统时钟 8MHz

如下图所示为 8 位 PWM 模式的工作时序。

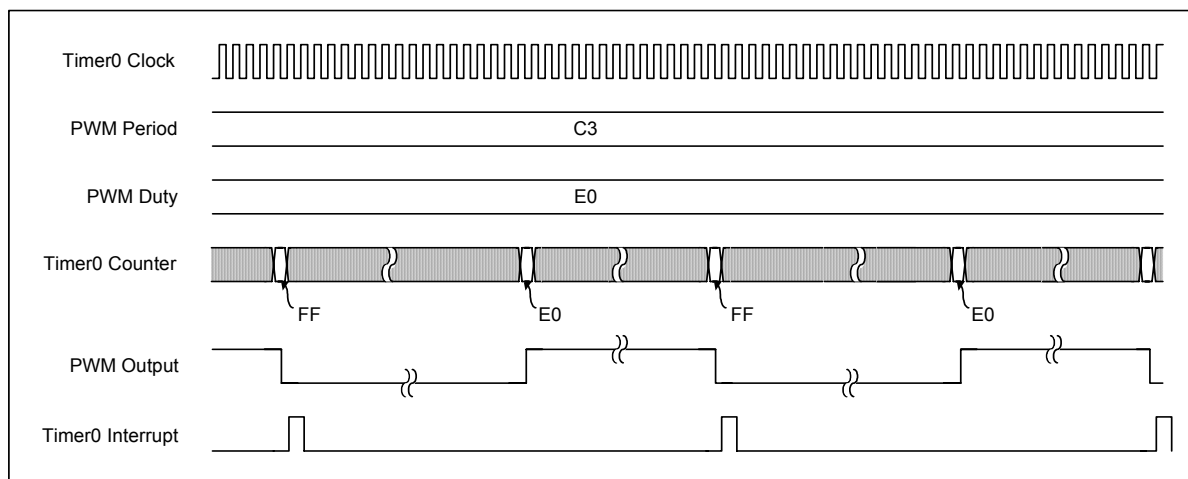


图 5-31 8 位 PWM 模式的工作时序

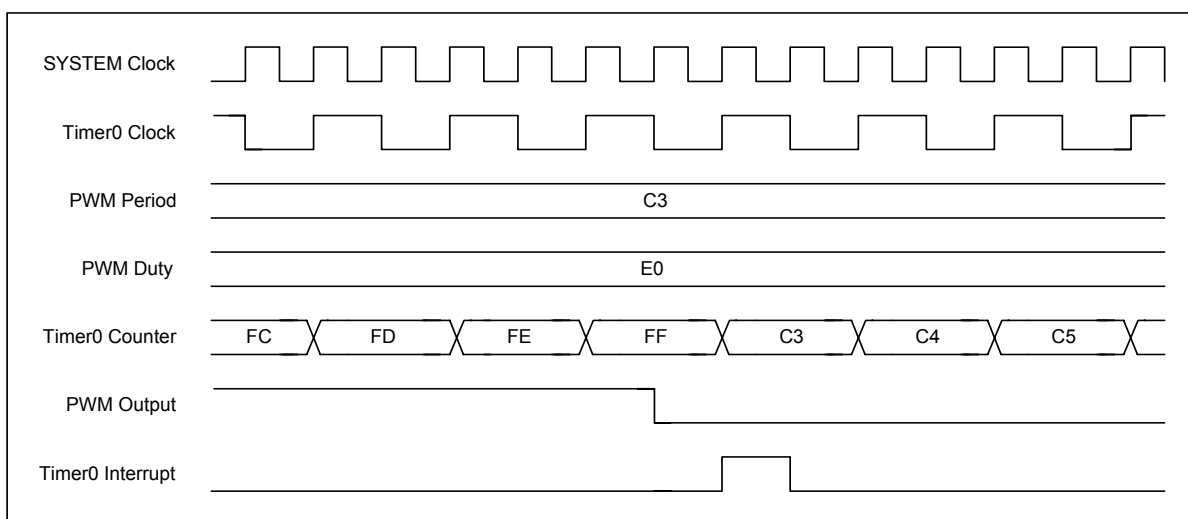


图 5-32 8 位 PWM 模式下计数初值的重载时序

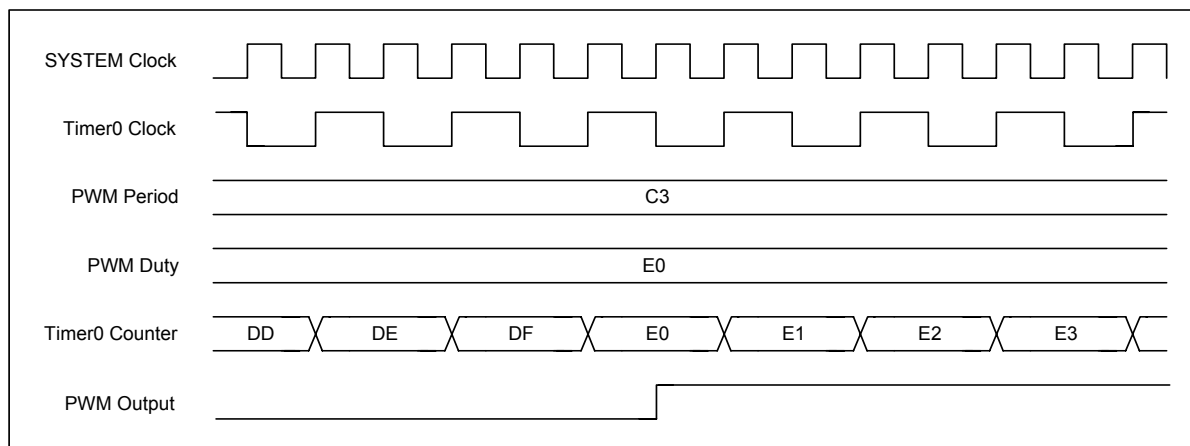


图 5-33 8 位 PWM 输出波形的翻转时序

【例 5.9.8】 将定时/计数器 0 设置为 8 位 PWM 模式，并从 PB2 输出 PWM 波形。

```

lda    #$7F                ; 启动定时/计数器 0 之前，首先设置计数初值
sta    P_TMR0_PWMduty     ; 设置占空比
lda    #$00
sta    P_TMR0_PWMPeriod   ; 设置周期值
lda    #C_T0FCS_Div_8     ; 设置定时/计数器 0 时钟源为 Fsys/8
sta    P_TMR0_1_Ctrl1
lda    #C_T08B_PWM        ; PWM 将定时/计数器 0 设置为 8 位 PWM 模式
sta    P_TMR0_1_Ctrl0
    
```

5.10 其它外围模块

5.10.1 看门狗定时器

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/计数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，可以通过看门狗复位来跳出。

SPMC65P1504A/1502A 芯片的看门狗定时/计数器功能可以通过芯片配置寄存器(\$7FE0)来使能或者禁止。看门狗定时/计数器原理见图 5-34。

看门狗定时/计数器也可用来唤醒 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/计数器的特点如下：

- 8 级可选中断频率（1.5Hz~195Hz）
- 8 个中断请求信号后即复位 CPU
- 可作为 STOP 模式的唤醒源
- 通过设置芯片配置寄存器(P_MO, \$7FE0)来使能或者禁止

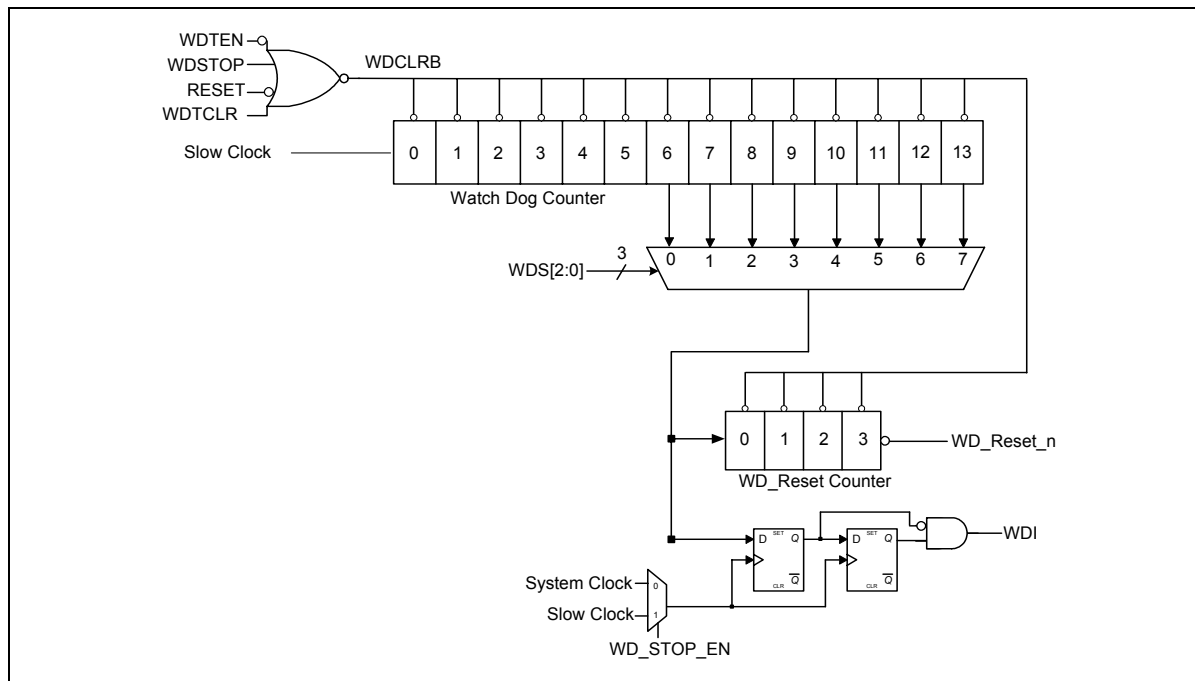


图 5-34 看门狗定时器原理框图

(1) 看门狗控制寄存器(P_WDT_Ctrl, \$0032)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	SCKEN	WDS2	WDS1	WDS0	-	-	-	-
ACCESS	R/W	R/W	R/W	R/W				
DEFAULT	1	1	1	1	0	0	0	0

Bit 7 **SCKEN**: STOP 模式下的看门狗内置 RC 振荡器使能位

- 0 = 禁止
- 1 = 使能

$$011 = F_{SLOW} \div 1024$$

$$100 = F_{SLOW} \div 2048$$

$$101 = F_{SLOW} \div 4096$$

$$110 = F_{SLOW} \div 8192$$

$$111 = F_{SLOW} \div 16384$$

Bit [6:4] **WDS** [2:0]: 看门狗中断频率选择位

$$000 = F_{SLOW} \div 128$$

$$001 = F_{SLOW} \div 256$$

$$010 = F_{SLOW} \div 512$$

F_{SLOW} : 内置 RC 振荡器振荡频率, 典型值为 25kHz

Bit [3:0] 保留

WDS[2:0]	看门狗复位频率 (Hz)	看门狗中断频率 (Hz)
000	195/8	195
001	97/8	97

WDS[2:0]	看门狗复位频率 (Hz)	看门狗中断频率 (Hz)
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

(2) 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							
ACCESS	W							
DEFAULT	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] 5.10.1 使能看门狗定时/计数器功能

```

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
    
```

5.10.2 时基定时器

SPMC65P1504A/1502A 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 5-12），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 5-12 时基周期列表（系统时钟 F_{sys} = 8.0MHz）

INTIMS [3:0]	时基周期	
	分频	F _{T0} = F _{sys}
0000	-	Function disable
0001	2 ⁷ =128	16us
0010	2 ⁸ =256	32us
0011	2 ⁹ =512	64us

INTIMS [3:0]	时基周期	
	分频	$F_{T0} = F_{SYS}$
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

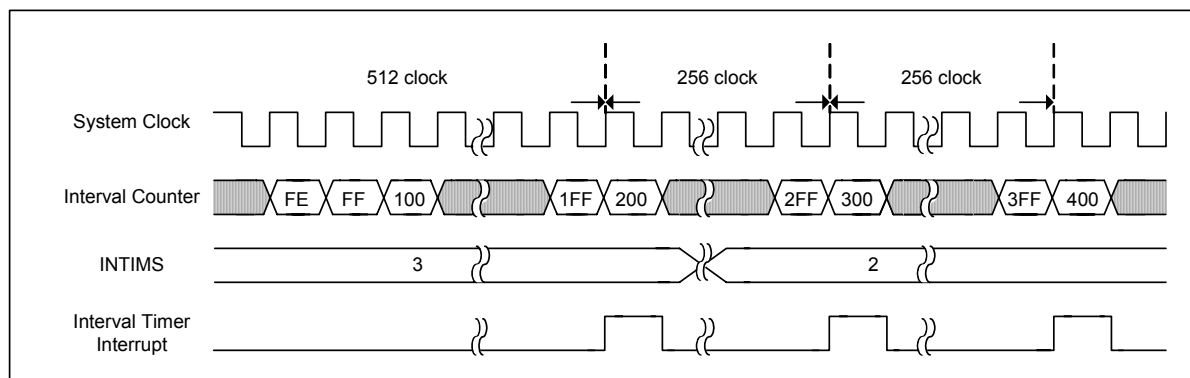


图 5-35 时基定时器工作时序

[例] 5.10.2 利用时基定时器产生 512us 中断

```

lda    #C_TBSE_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
    
```

5.10.3 蜂鸣器控制寄存器

当蜂鸣器输出功能使能时，可以通过 PB6 管脚输出频率可调的、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，频率范围如表 5-13所列。

表 5-13 蜂鸣器输出频率列表（系统时钟 $F_{SYS} = 8.0\text{MHz}$ ）

BZFS[3:0]	蜂鸣器输出频率	
	分频	$F_{T0} = F_{SYS} / \text{分频}$
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

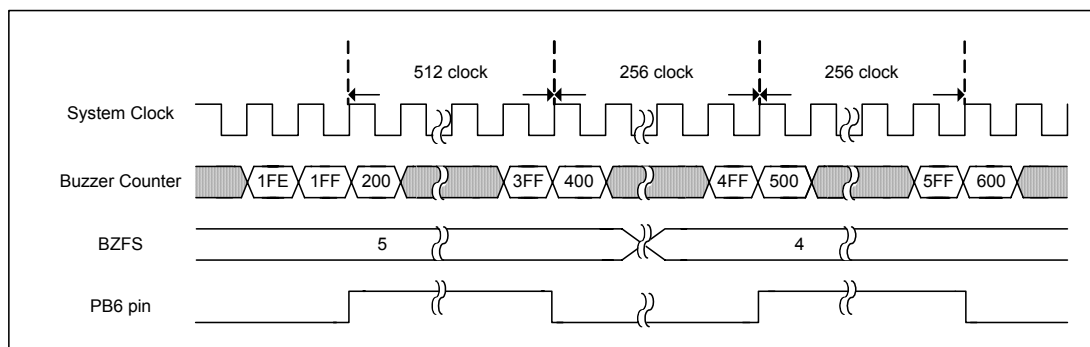


图 5-36 蜂鸣器运行时序

■ 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
ACCESS	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	0	0	0	0	0	0	0	0

Bit [7:4] INTIMS[3:0]: 时基频率选择位

0000 = 功能禁止
 0001 = $F_{SYS} \div 128$
 0010 = $F_{SYS} \div 256$
 0011 = $F_{SYS} \div 512$
 0100 = $F_{SYS} \div 1024$
 0101 = $F_{SYS} \div 2048$
 0110 = $F_{SYS} \div 4096$
 0111 = $F_{SYS} \div 8192$
 1000 = $F_{SYS} \div 16384$
 1001 = $F_{SYS} \div 32768$
 1010 = $F_{SYS} \div 65535$
 1011 = $F_{SYS} \div 131072$
 1100 = $F_{SYS} \div 262144$
 1101 = $F_{SYS} \div 524288$
 1110 = $F_{SYS} \div 2^{21}$
 1111 = $F_{SYS} \div 2^{23}$

Bit [3:0] BZFS[3:0]: 蜂鸣器频率选择位

0000 = 功能禁止
 0001 = $F_{SYS} \div 64$
 0010 = $F_{SYS} \div 128$
 0011 = $F_{SYS} \div 256$
 0100 = $F_{SYS} \div 512$
 0101 = $F_{SYS} \div 1024$
 0110 = $F_{SYS} \div 2048$
 0111 = $F_{SYS} \div 4096$
 1000 = $F_{SYS} \div 8192$
 1001 = $F_{SYS} \div 4$
 1010 = $F_{SYS} \div 8$
 1011 = $F_{SYS} \div 16$
 1100 = $F_{SYS} \div 32$
 1101 = $F_{SYS} \div 32$
 1110 = $F_{SYS} \div 32$
 1111 = $F_{SYS} \div 32$

【例】5.10.3 利用蜂鸣器功能输出方波，周期为 128us，占空比为 50%

```
lda    #C_BUZ_Div_lk          ; 将立即数$05 放入累加器
sta    P_BUZ_Ctrl
```

5.11 芯片配置寄存器

5.11.1 简介

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。其中，\$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 用于设置 RC 振荡器频率输出及 I/O 端口初始状态。
 \$7FE3 用于选择非屏蔽中断源（NMI）。

5.11.2 芯片配置寄存器

(1) \$7FE0 寄存器

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	-	-	-	CLKSEL1	CLKSEL0	LVREN	WDTEN	-
ACCESS	-	-	-	R	R	R	R	-
DEFAULT	1	0	1	0	0	0	0	1

Bit [7:5] 保留

0 = 禁止 LVR

Bit [4:3] **CLKSEL** [1:0]: 系统时钟源选择位

1 = 使能 LVR

11 = 保留

Bit 1 **WDTEN**: 看门狗功能使能位

10 = 外部时钟

0 = 禁止 WDT

01 = 内部 RC 振荡器

1 = 使能 WDT

00 = 晶体或陶瓷振荡器

Bit 0 保留

Bit 2 **LVREN**: 低电压复位使能位

(2) \$7FE2 寄存器

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	-	-	-	-	-	-	RCOUT	IOINIT
ACCESS	-	-	-	-	-	-	R	R
DEFAULT	1	1	1	1	1	1	1	1

Bit [7:2] 保留

Bit 0 **IOINIT**: IO 端口初始化选择位

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输入

1 = 使能 (从 XO 管脚输出时钟信号)

0 = 所有端口均初始化为带下拉电阻输入

0 = 禁止 (无输出)

(3) \$7FE3 寄存器

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
名称	-	-	-	-	-	NMIS2	NMIS1	NMIS0
ACCESS	-	-	-	-	-	R	R	R
DEFAULT	1	1	1	1	1	1	1	1

Bit [7:3] 保留

100 = 保留

Bit [2:0] **NMIS** [2:0]: 非屏蔽中断源控制位

011 = 保留

111 = 禁止

010 = 保留

110 = 保留

001 = 保留

101 = 保留

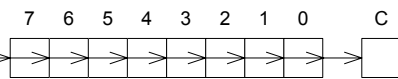
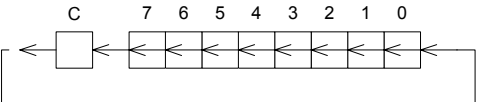
000 = PB4 (INT0)作为 NMI 中断源。

5.12 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	数据与累加器 A 中的值进行“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.	ASL A	0A	1	2	算术左移 C 7 6 5 4 3 2 1 0 □ ← □ ← □ ← □ ← □ ← □ ← □ ← □ ← “0”	N----ZC
18.	ASL aa	06	2	5		
19.	ASL aa,X	16	2	6		
20.	ASL aaaa	0E	3	6		
21.	ASL aaaa,X	1E	3	7		
22.	BCC ??	90	2	2*	若无进位 (C = 0), 跳转: $pc \leftarrow (pc) + ??$	-----
23.	BCS ??	B0	2	2*	若有进位 (C = 1), 跳转: $pc \leftarrow (pc) + ??$	-----
24.	BEQ ??	F0	2	2*	若相等 (Z = 1), 跳转: $pc \leftarrow (pc) + ??$	-----
25.	Bit aa	24	2	3	用累加器 A 进行位测试	NV----Z-
26.	Bit aaaa	2C	3	4	$Z \leftarrow (A) \wedge (M), N \leftarrow (M_7), V \leftarrow (M_6)$	
27.	BMI ??	30	2	2*	若结果为负 (N = 1), 跳转: $pc \leftarrow (pc) + ??$	-----
28.	BNE ??	D0	2	2*	若不相等 (Z = 0), 跳转: $pc \leftarrow (pc) + ??$	-----
29.	BPL ??	10	2	2*	若结果不为负 (N = 0), 跳转: $pc \leftarrow (pc) + ??$	-----
30.	BVC ??	50	2	2*	若无溢出 (V = 0), 跳转: $pc \leftarrow (pc) + ??$	-----
31.	BVS ??	70	2	2*	若有溢出 (V = 1), 跳转: $pc \leftarrow (pc) + ??$	-----
32.	CLC	18	1	2	清除进位标志: $C \leftarrow "0"$	-----0
33.	CLD	D8	1	2	退出十进制运算模式: $D \leftarrow "0"$	----0---

序号	助记符	操作码	字节数	指令周期 (Cycles)	操作	标志位 NV-BDIZC
34.	CLI	58	1	2	清除中断屏蔽位: $I \leftarrow "0"$	----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		
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	异或: $A \leftarrow (A) \oplus (M)$	N----Z-
64.	EOR #dd	49	2	2		
65.	EOR aa	45	2	3		
66.	EOR aa, X	55	2	4		
67.	EOR aaaa	4D	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
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: $PC \leftarrow$ 跳转地址	-----
87.	JMP (aaaa)	6C	3	5		
88.	JSR aaaa	20	3	6	调用子程序: $(sp) \leftarrow (pc_H)$, $sp \leftarrow sp - 1$, $(sp) \leftarrow (pc_L)$, $sp \leftarrow sp - 1$, $pc \leftarrow aaaa$	-----
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		

序号	助记符	操作码	字节数	指令周期 (Cycles)	操作	标志位 NV-BDIZC
101.	LDX aaaa,Y	BE	3	4	将数据送入 Y 寄存器中： $Y \leftarrow (M)$	N----Z-
102.	LDY #dd	A0	2	2		
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	$(sp) \leftarrow A, sp \leftarrow sp - 1$	-----
122.	PHP	08	1	3	$(sp) \leftarrow \text{状态寄存器 P}, sp \leftarrow sp - 1$	-----
123.	PLA	68	1	4	$sp \leftarrow sp + 1, A \leftarrow (sp)$	-----
124.	PLP	28	1	4	$sp \leftarrow sp + 1, \text{状态寄存器 P} \leftarrow (sp)$	恢复
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	循环右移： 	N----ZC
130.	ROR A	6A	1	2		
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		

序号	助记符	操作码	字节数	指令周期 (Cycles)	操作	标志位 NV-BDIZC
135.	RTI	40	1	6	中断返回: $sp \leftarrow sp + 1$, 状态寄存器 $P \leftarrow (sp)$, $sp \leftarrow sp + 1$, $pc_L \leftarrow (sp)$, $sp \leftarrow sp + 1$, $pc_H \leftarrow (sp)$	恢复
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	设置为十进制运算模式: $D \leftarrow "1"$	---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		
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		
164.	STX aa, Y	96	2	4		
165.	STX aaaa	8E	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
166.	STY aa	84	2	3	将寄存器 Y 的值赋给存储器 M (M) ← Y	-----
167.	STY aa, X	94	2	4		
168.	STY aaaa	8C	3	4		
169.	TAX	AA	1	2	将累加器 A 的值送入 X 寄存器中: X ← A	N----Z-
170.	TAY	A8	1	2	将累加器 A 的值送入 Y 寄存器中: Y ← A	N----Z-
171.	TST aa,0	07	2	3	位测试: Z ← ~(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 ← sp	N----Z-
180.	TXA	8A	1	2	将 X 寄存器的值送入累加器 A: A ← X	N----Z-
181.	TXS	9A	1	2	将 X 寄存器的值送入堆栈指针 (SP): sp ← X	N----Z-
182.	TYA	98	1	2	将 Y 寄存器的值送入累加器 A: A ← 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	T _a =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	4.0	-	5.5	V	
系统时钟	F _{SYS}	200K	-	8.0M	Hz	VDD = 4.0V~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 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 管脚属性描述

管脚	描述	符号	最小值	典型值	最大值	单位	条件
VDD,VSS	HALT 模式的电源电流	I_{DD}	-	10.5	14	mA	$F_{OSC} = 16.0\text{MHz} @$ $V_{DD} = 5.5\text{V}$
	STOP 模式的电源电流	I_{HALT}	-	6.5	8.5	mA	$V_{DD} = 5.5\text{V}$
	振荡稳定时间	I_{PD}	-	8	15	μA	$V_{DD} = 5.5\text{V}$
XI,XO	RC 时钟频率偏差	t_{ST}	-	1024	-	F_{SYS}	晶体振荡器模式
	外部电容	DFV	-15	-	+15	%	RC 振荡器模式
	外部电阻	C_{RC}	20	-	100	pf	RC 振荡器模式
	高电平时钟脉宽	R_{RC}	2.0	-	-	$k\Omega$	RC 振荡器模式
	低电平时钟脉宽	t_H	31.25	-	-	ns	外部时钟源模式
	输入： a. 施密特触发器触发 b. 上拉/下拉/悬浮 输出： a. 有 LED 驱动能力 b. 4mA/10mA 驱动输出	t_L	31.25	-	-	ns	外部时钟源模式
PA[7:0], PB[5:0], PC[3:0], PD[2:0]	输入： a. 施密特触发器触发 b. 上拉/下拉/悬浮 输出： a. 有 LED 驱动能力 b. 4mA/20mA 驱动输出	V_{IH}	3.5	-	-	V	
		V_{IL}	-	-	1.4	V	
		V_{TH}	-	0.5	-	V	
		I_{OH}	4.0	-	-	mA	$V_{OH} = 4.5\text{V}$
		I_{OL}	10.0	-	-	mA	$V_{OL} = 0.5\text{V}$
		I_P	-	50	-	μA	$V_{IN} = V_{DD} \text{ 或 } V_{SS}$
PB[7:6]	端口高阻态输入泄漏电流	V_{IH}	3.5	-	-	V	
		V_{IL}	-	-	1.4	V	
		V_{TH}	-	0.5	-	V	
		I_{OH}	4.0	-	-	mA	$V_{OH} = 4.5\text{V}$
		I_{OL}	20.0	-	-	mA	$V_{OL} = 0.5\text{V}$
		I_P	-	50	-	μA	$V_{IN} = V_{DD} \text{ 或 } V_{SS}$
All Port	输入： a. 施密特触发器触发 b. 上拉 输出：无	I_Z	-	-	10	μA	
RESETB		V_{IH}	3.5	-	-	V	
		V_{IL}	-	-	1.4	V	
		V_{TH}	-	0.5	-	V	

管脚	描述	符号	最小值	典型值	最大值	单位	条件
		I_P	-	125	-	μA	$V_{IN} = V_{SS}$
		t_W	1.0	-	-	μS	输入最小宽度

6.4 模拟接口电气特性 ($V_{DD} = 5.0V$, $T_A = -40^{\circ}C \sim 85^{\circ}C$)

功能	描述	符号	最小值	典型值	最大值	单位	条件
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	

7 封装和管脚焊接位置

7.1 管脚分配和位置

请联系凌阳销售代表，以获得更多信息。

7.2 订货信息

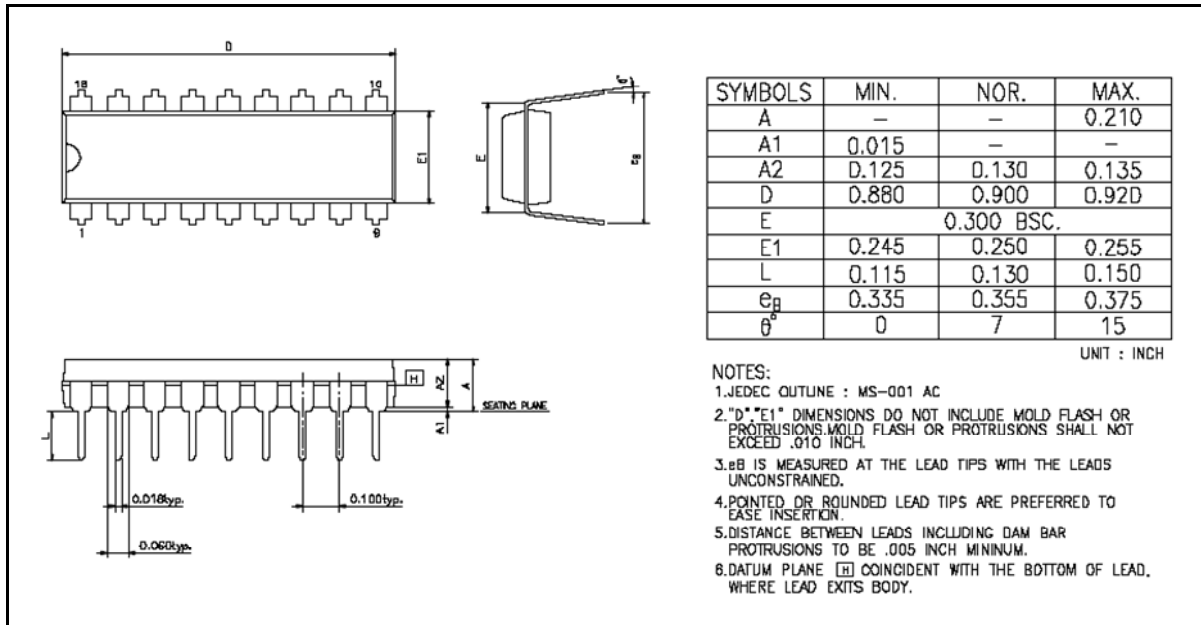
芯片系列号	封装类型
SPMC65P1504A-NnnV-PS07	封装类型 – SOP 24 (300mil)
SPMC65P1504A-NnnV-PS08	封装类型 – SOP 28 (300mil)
SPMC65P1504A-NnnV-PD05	封装类型 – PDIP 24 (300mil)
SPMC65P1504A-NnnV-PD08	封装类型 – PDIP 28 (300mil)
SPMC65P1502A-NnnV-PS05	封装类型 – SOP 18 (300mil)
SPMC65P1502A-NnnV-PS08	封装类型 – SOP 24 (300mil)
SPMC65P1502A-NnnV-PD08	封装类型 – PDIP 18 (300mil)
SPMC65P1502A-NnnV-PD10	封装类型 – PDIP 24 (300mil)

注 1：代码号 (Nnn) 用于用户区分不同的芯片。

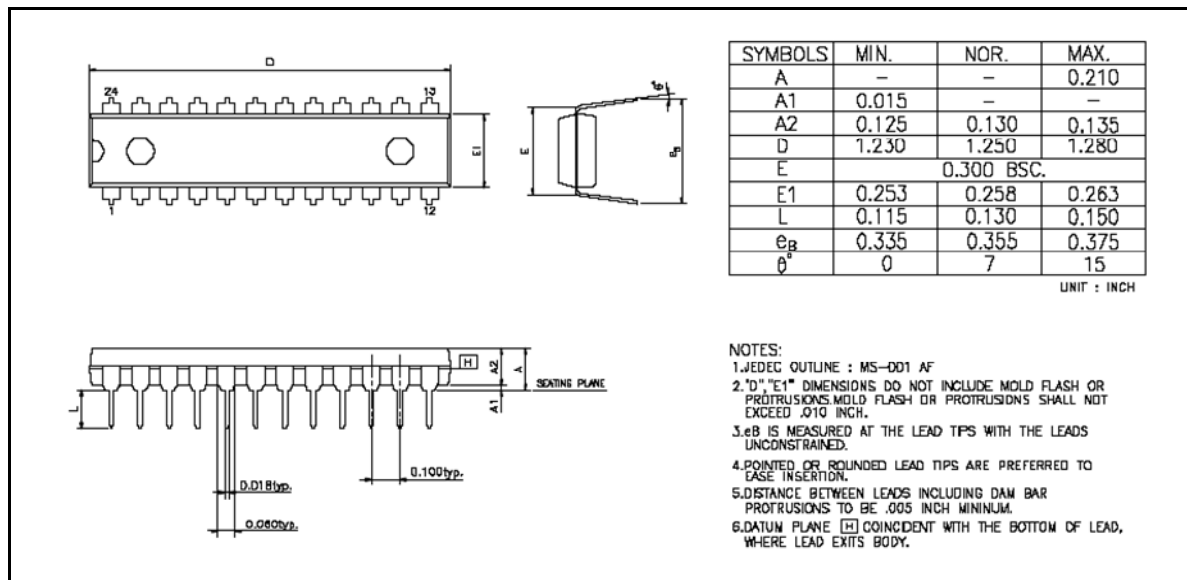
注 2：Nnn (代码号，N = A - Z 或 0 - 9，nn = 00 - 99)；V (版本号，V = A - Z)。

7.3 封装

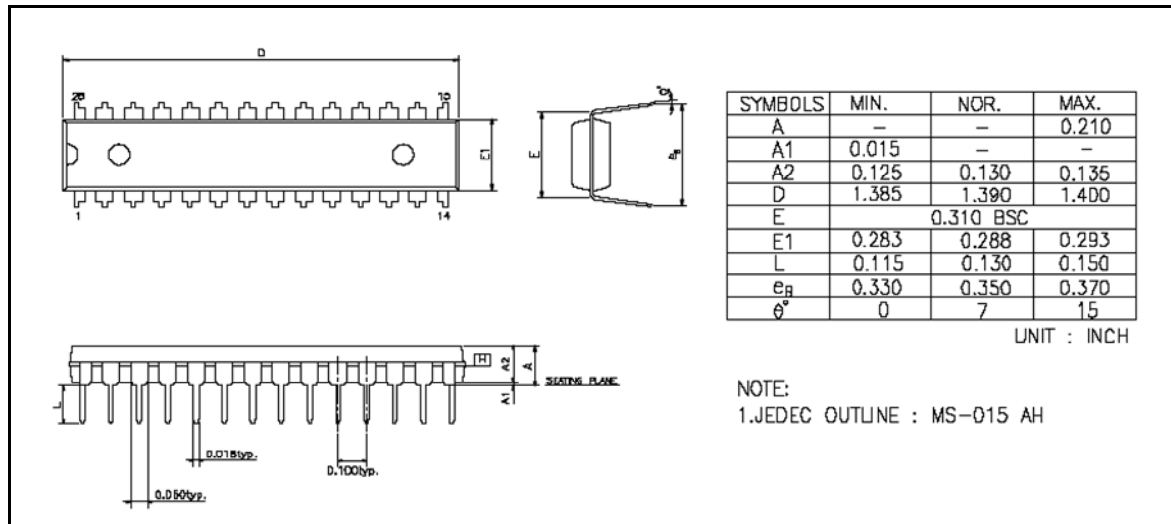
7.3.1 PDIP 18 (300mil)



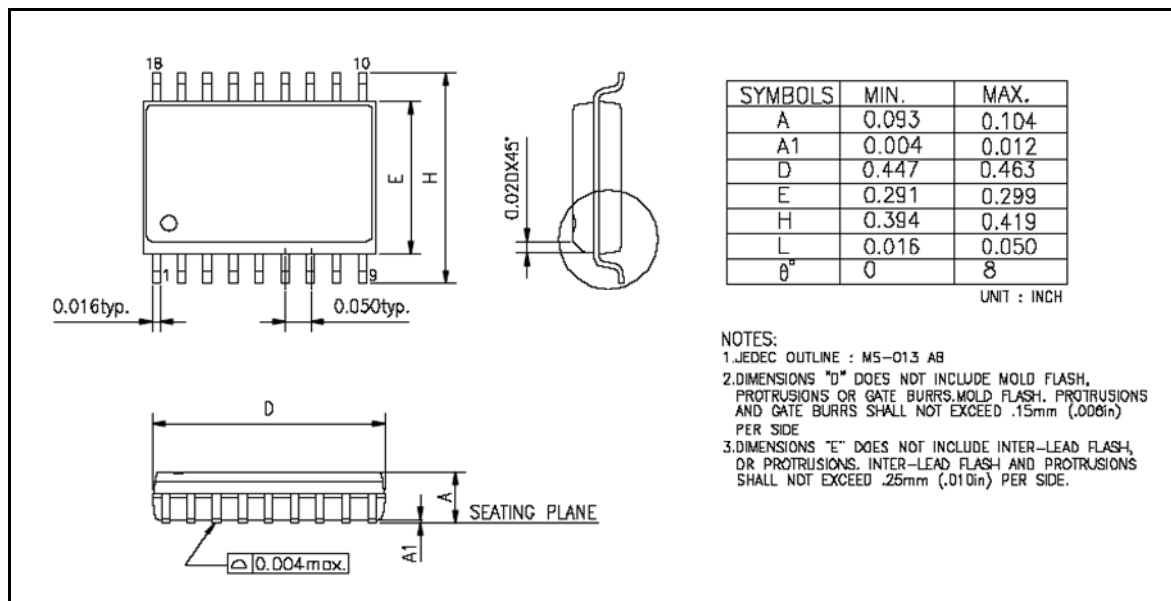
7.3.2 PDIP 24 (300mil)



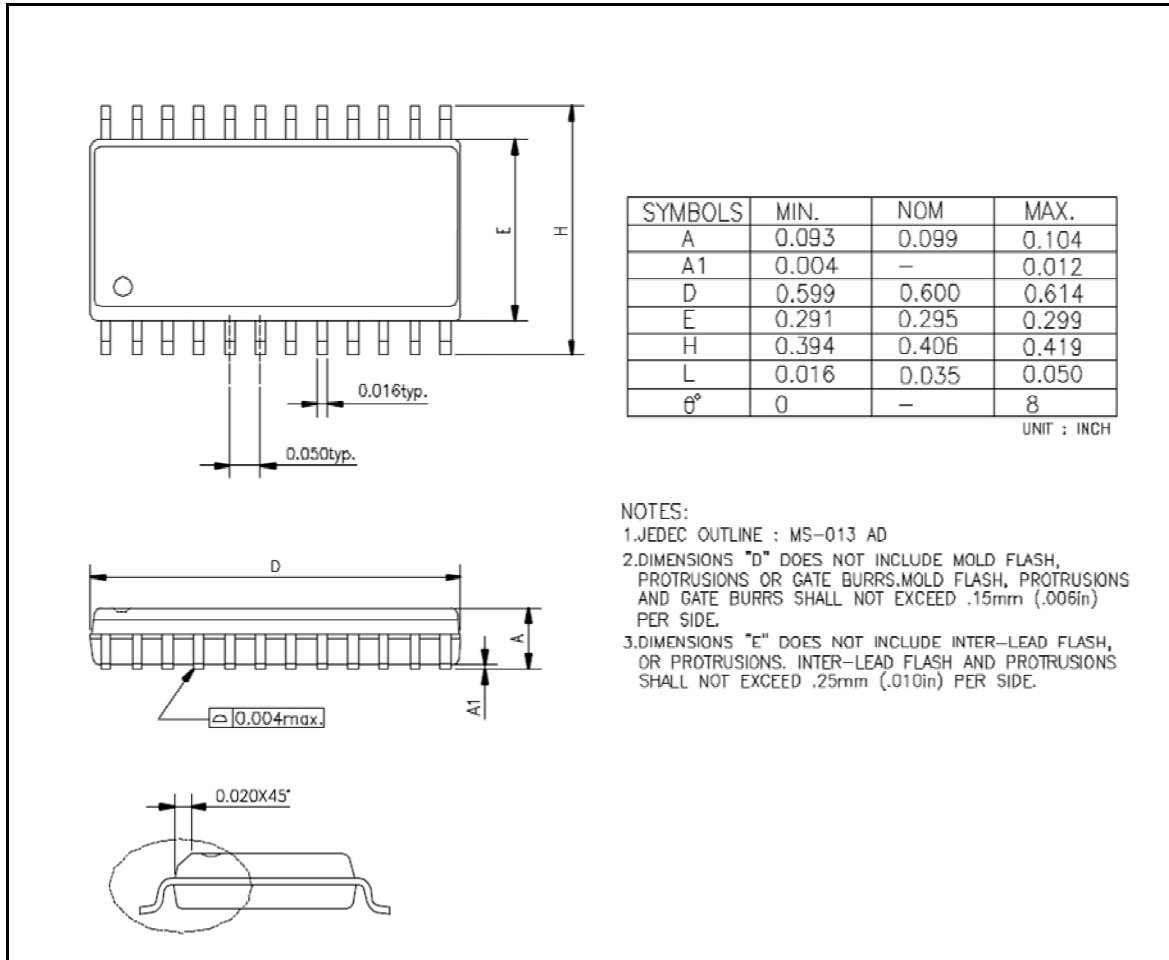
7.3.3 PDIP 28 (300mil)



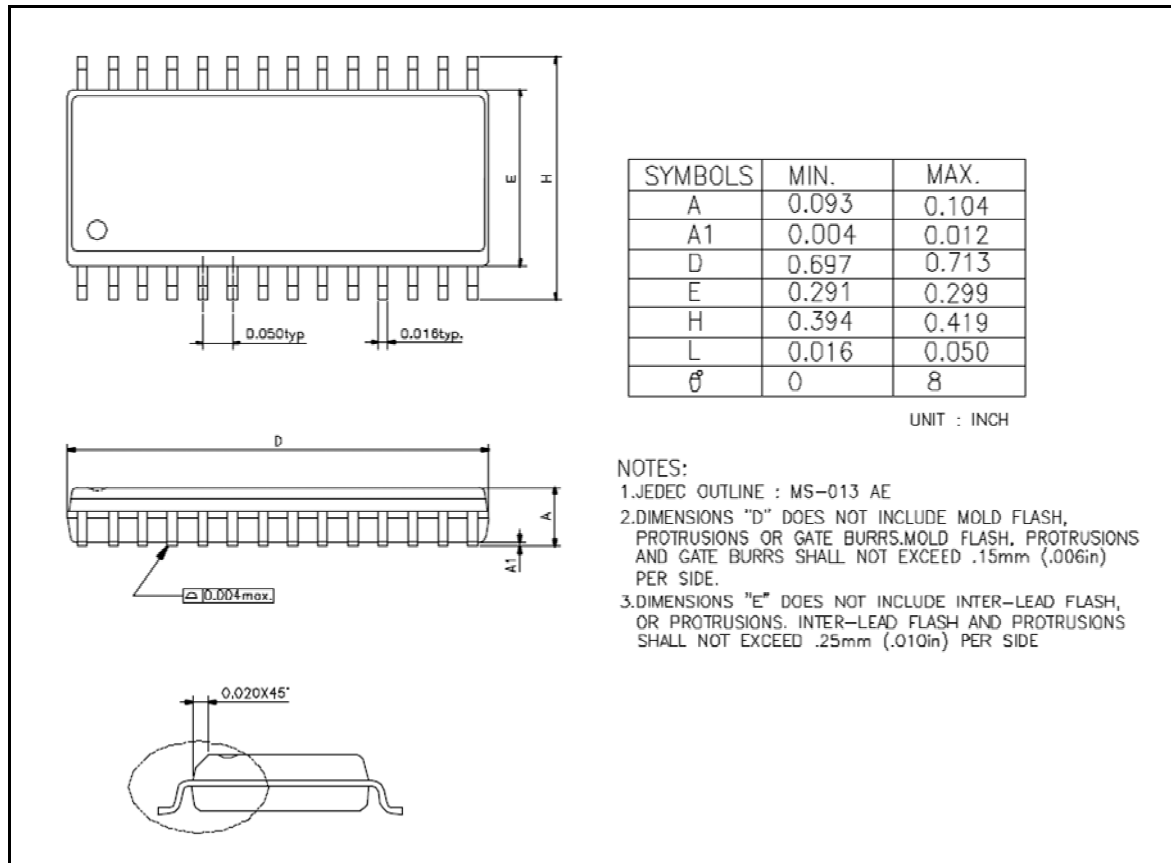
7.3.4 SOP 18 (300mil)



7.3.5 SOP 24 (300mil)



7.3.6 SOP 28 (300mil)



7.4 储存条件及时间

封装	湿度灵敏性级别	最大回流焊温度	曝露储存条件	干燥包装
SOP	3级	220 +5/-0°C	168Hrs @ ≤30°C/ 60% R.H.	否
PDIP	3级 (参考)	220 +5/-0°C (参考)	N/A	否

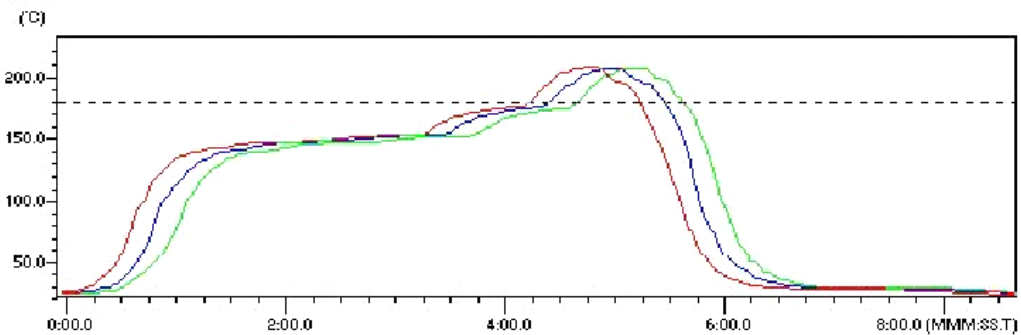
注 1: 请参考 IPC/JEDEC 规范 J-STD-020A 及 EIA JEDEC 规范 JFSD22-A112。

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)