



TM57PA20/

TM57PA40

8 位 单片机

使用手册

**Tenx reserves the right to change or
discontinue this product without notice.**

tenx technology inc.

CONTENTS

1. 基本功能.....	3
2. 系统结构图	3
3. 管脚分配图	4
4. 管脚描述.....	5
5. 功能描述.....	6
5.1 CPU芯片	6
5.1.1 时钟配置和指令周期.....	6
5.1.2 寻址模式	6
5.1.3 程序计数器和堆栈.....	7
5.1.4 ALU和工作寄存器.....	7
5.1.5 状态寄存器	7
5.1.6 中断	9
5.2 芯片工作模式.....	10
5.2.1 复位	10
5.2.2 系统配置寄存器（SYSCFG）	10
5.2.3 可重复编程	11
5.2.4 省电模式	12
5.3 外围功能图	12
5.3.1 WDT/WKT定时器	12
5.3.2 定时器 0：可扩展的 8 位定时计数器（PSC）	13
5.3.3 定时器 1：可扩展的 8 位定时器（PSC）	14
5.3.4 8+2 位PWM	15
5.3.5 12 位数模转换器	17
5.3.6 系统时钟振荡器	18
5.3.7 蜂鸣输出.....	19
5.4 I/O口	21
5.4.1 PA0-2	21
5.4.2 PA3-6，PB0-1，PD0-7	22
5.4.3 PA7	23
6. 内存功能图	24
F-Plane.....	24
R-Plane	25
7. 指令表	27
8. 电气特性.....	39
9. 包装说明.....	42
20-DIP Package Dimension	42

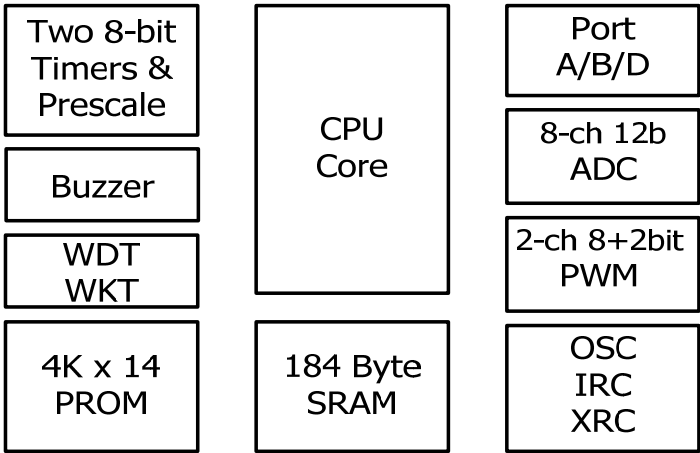
20-SOP Package Dimension.....	43
20-SSOP Package Dimension	44
16-DIP Package Dimension	45
16-SOP Package Dimension.....	46
16-SSOP Package Dimension	47
8-DIP Package Dimension	48
8-SOP Package Dimension.....	49

1. 基本功能

- 1. ROM：4K x 14 OTP 或者 2K x 14 位 TTP™（ROM 可编程两次）
- 2. PAM：184 x 8 位
- 3. 堆栈：6 级
- 4. I/O 口：可编程三次(三组可编程端口)（最大 18 脚）
- 5. 定时器 0/控制器：带有 1-256 预分频的 8 位定时控制器
- 6. 定时器 1：带有 1-256 预分频的 8 位可自动重新加载定时器
- 7. 两个含 1024 工作决议的 8+2 PWM 通道
- 8. 含 8 个管脚输入口的 12 位数模转换
- 9. 可蜂鸣输出
- 10. 看门狗/唤醒器：基于内部 RC 振荡，唤醒时间为 13-140 毫秒的芯片定时器
- 11. 复位：电源复位，看门狗复位，低电压复位，外部复位
- 12. 系统时钟模式：
 - 慢速晶体：32KHZ
 - 快速晶体：455KHZ – 24MHZ
 - 内部 RC：4MHZ
 - 外部 RC
- 13. 工作电压：LV 复位电压至 5.5V
- 14. 指令：36 个
- 15. 中断：可利用三个 IO 端口或定时器 0/1 以及唤醒定时器发生中断
- 16. 支持省电模式
- 17. 封装：8DIP/SOP，16DIP/SOP/SSOP，20DIP/SOP/SSOP

DEVICE	OTP ROM	TTP™ function	Note
TM57PA40	4K x 14 bits	Available	
TM57PA20	2K x 14 bits	Not Available	

2. 系统结构图



3. 管脚分配图

VSS	1	U	20	VDD
Xrc/Xin/PA4	2		19	PA6/ADC0/INT0
Xout/PA3	3		18	PA1/ADC1/INT1
VPP/nRESET/INT2/PA7	4	TM57PA20	17	PA2/ADC2/T0I
T1OUT/PD0	5	TM57PA40	16	PB1/ADC3
BUZZER/PD1	6	20 SOP	15	PD7/ADC4
PD2	7	20 DIP	14	PA5/ADC5
PD3	8	20 SSOP	13	PA0/PWM0
PD4	9		12	PB0/ADC7/PWM1
PD5	10		11	PD6/ADC6/TCOUT

VSS	1	U	16	VDD
Xrc/Xin/PA4	2	TM57PA20	15	PA6/ADC0/INT0
Xout/PA3	3	TM57PA40	14	PA1/ADC1/INT1
VPP/nRESET/ INT2/PA7	4	16 SOP	13	PA2/ADC2/T0I
T1OUT/PD0	5	16 DIP	12	PB1/ADC3
BUZZER/PD1	6	16 SSOP	11	PD7/ADC4
PD2	7		10	PA5/ADC5
PD3	8		9	PA0/PWM0

VSS	1	U	8	VDD
Xrc/Xin/PA4	2	TM57PA20	7	PA1/ADC1/INT1
Xout/PA3	3	TM57PA40	6	PA2/ADC2/T0I
VPP/nRESET/ INT2/PA7	4	8 SOP/DIP	5	PA0/PWM0

4. 管脚描述

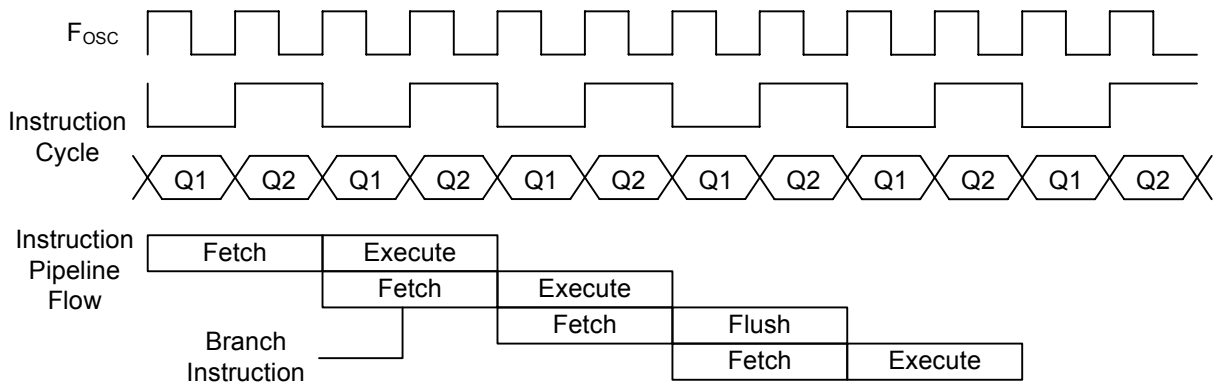
管脚名称	输入/ 输出	管脚描述
PA2-PA0	I/O	位编程 I / O 端口，可施密特触发输入或推挽输出或者伪开漏式输出。上拉电阻由软件分配
PA6-PA3 PB1-PB0 PD7-PD0	I/O	位编程 I / O 端口，可施密特触发输入，推挽输出或者开漏式输出。上拉电阻由软件分配
PA7	I	施密特触发输入
nRESET	I	外部能动低复位 (外部低电平复位)
Xin, Xout	-	晶体/陶瓷振荡用于系统时钟
Xrc		外部 RC 振荡用于系统时钟
VDD, VSS	P	电源输入管脚和接地脚
VPP	I	PROM 编程高电压输出 (输入)
INT0~2	I	外部中断输入
T0I	I	在计数模式中定时器 0 输入
T1OUT	O	定时器 1 匹配输出，当定时器 1 发生溢出时 T1OUT 切换 (脚翻转输出)
BUZZER	O	蜂鸣输出
TCOUT	O	N 输出预分频指令周期时钟。当 N 是 1, 2, 4, 8 时，指令时钟频率是预分频两次的系统时钟频率 (指令周期为系统时钟的两倍)
PWM0/PWM1	O	10 位 PWM 输出
ADC7~0	I	A/D 转换输出(输入)

5. 功能描述

5.1 CPU 芯片

5.1.1 时钟配置和指令周期

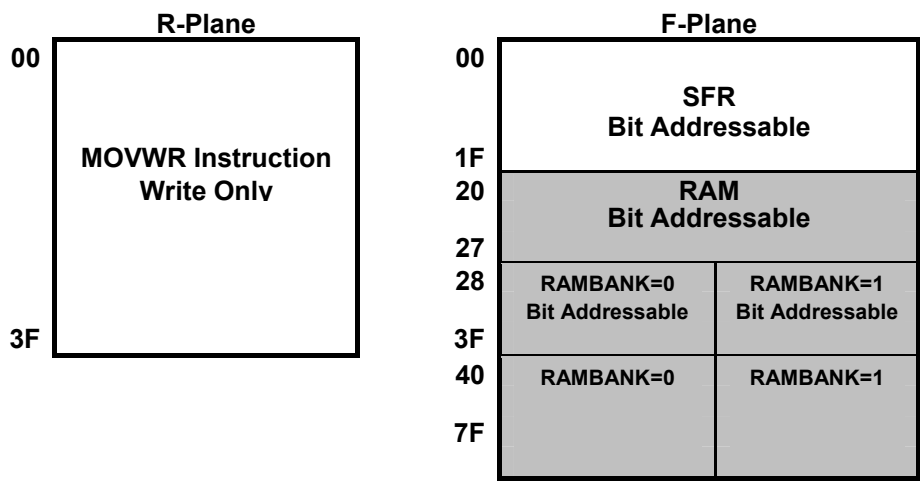
CPU 时钟输入（X_{IN}）在内部被分成用于指令周期的 Q1 和 Q2 两个状态.程序计数器在 Q1 状态更新且指令从程序寄存器获得，在状态 Q2 锁于指令寄存器。在接下来的 Q1-Q2 周期，程序被解码并执行。分支指令占用两个周期，因为从传输信道获取指令的同时新指令被获取并执行。



5.1.2 寻址模式

CPU 有两个数据存储盘：R-Plane 和 F-Plane。R-Plane 里的寄存器只能写入。“MOBWR”指令通过直接寻址方式把 W 寄存器里的常量放在 R-Plane 寄存器中。

F-Plane 的低位保留用于 SFR 寄存器。SFR 寄存器以上的寄存器是一般目的的数据存储器，作为静态 RAM 使用。F-Plane 可被直接或间接寻址。通过 INDF 寄存器可间接寻址，INDF 不是物理寄存器。当对 INDF 进行存取时，它会根据 FSR 寄存器内的值作为地址，并指向该地址的寄存器。F-Plane 的前半部分可位寻址，后半部分不可位寻址。



5.1.3 程序计数器和堆栈

程序计数器是一个 12-bit 宽，可寻址 4K*14 的程序只读存储器。当一个程序指令被执行时，PC 里就存放着下一条将要被执行程序地址。PC 值都会自动加一，以下情况除外：复位向量（000h）和中断向量（001h）被提供用于 PC 初始化和中断。对于 CALL/GOTO 指令，PC 从指令中加载 12 位地址。对于 RET/RETI/RETLW 指令，PC 从堆栈顶取回其地址值。对于更新 PC[7:0] 的其他指令，PC[11:8] 不变。堆栈有 12 位宽，深 6 级。CALL 指令和硬件中断将依顺序进入堆栈，RET/RETI/RETLW 指令将按顺序弹出堆栈。

5.1.4 ALU 和工作寄存器

ALU 是个 8 位的算术逻辑单元，ALU 可以对 W 寄存器和其它寄存器中的数据进行加、减、移位以及逻辑运算。在两个操作数的指令中，主操作数是一个 8 位不可寻址的用于 ALU 操作的 W 寄存器。另一个操作数是一个文件寄存器或者一个立即常量。在单个操作数的指令中，操作数是 W 寄存器或者文件寄存器。指令执行时，ALU 可能影响状态寄存器中的标志位 C，DC 和 Z。C 和 DC 标志运用于借位和十进制借位，特别是在减法中。

5.1.5 状态寄存器

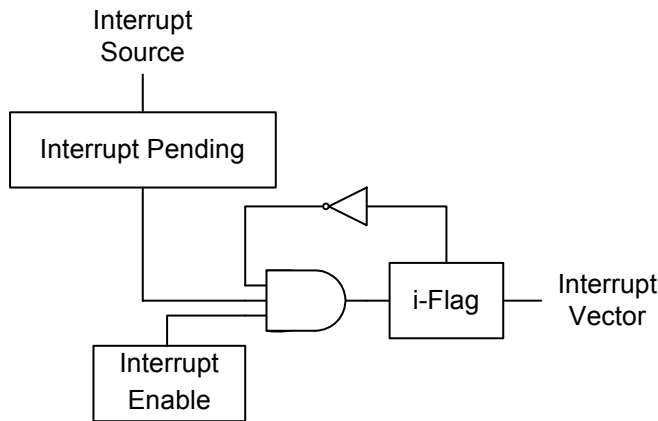
状态寄存器包含 ALU 计算结果的状态和复位状态。状态寄存器和其它的寄存器一样可以是任何寄存器的目的单元格。如果是影响 Z、DC 或 C 标志位的指令以状态寄存器作为目的单元格，那么就不可以写进这三个标志位了。这些位的置位或清零是来自 ALU 运算的逻辑值。所以，只有 BCF、BSF 和 MOVWF 指令被推荐选择状态寄存器用，因为这些指令不影响标志值。

状态	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
复位值	—	—	0	0	0	0	0	0
R/W	—	—	R/W	R	R	R/W	R/W	R/W
位	描述							
7-6	不使用							
5	RAM 区： RAM 区的选择 0: RAM0 区 1: RAM1 区							
4	TO: 时间输出 0: 上电复位， LVR 复位或执行 CLRWDT/SLEEP 指令以后 1: WDT 时间输出出现后 (WDT 计时溢出后)							
3	PD: 省电 0: 上电复位， LVR 复位或执行 CLRWDT 指令以后 1: SLEEP 指令后							
2	Zero Flag (Z) 0: 逻辑操作的结果不是0 1: 逻辑操作的结果是0							
1	十进制进位标志和借位标志(DC)							
	加法指令				减法指令			
	1: 低四位有进位 0: 没进位				1: 没借位 0: 低四位有借位			
0	进借位标志							
	加法指令				减法指令			
	1: MSB 有进位 0: 没进位				1:没借位 0: MSB有借位			

5.1.6 中断

TM57PA40 有六个中断源，共一个向量,没有中断优先级,每个中断源都有它自己的使能控制位,每个中断事件都可以触发它自己的中断标志位。因为没有中断优先权 ,所以 TM57PA40 中断的优先处理取决于 F/W 。

若相应的中断使能位被置 1(INTCON),它会触发 CPU 在最近执行指令周期的结尾处,接受并处理中断。同时,“CALL 0001”指令被插进 CPU,中断标志位被置 1 来防止中断嵌套。防止中断嵌套标志位在执行“RETI”指令后即被清零。也就是说,在中断触发时至少有一条指令在主程序中执行着。中断事件是边沿触发的。在处理中断程序后,F/W 必须清除中断事件的中断标志位。



5.2 芯片工作模式

5.2.1 复位

TM57PA40 可用如下四种方式复位：

- 上电复位
- 低电压复位（LVR）
- 通过 RESET 管脚来实现硬件复位
- 看门狗复位

上电复位以后，所有的系统和外围控制寄存器都将恢复它们的默认硬件复位值。时钟源，LVR 和芯片工作模式在 SYSCFG 寄存器中设定。

当电源电压低于阈值水平时，低电压复位功能静态重置。电源电压有两个阈值水平可以选择。LVR 的工作模式由 SYSCFG 寄存器定义。

外部管脚复位和看门狗复位是否能工作由 SYSCFG 寄存器设定，其所有控制寄存器在上电复位以后也都恢复默认复位值。TO/PD 标志位不受这些复位的影响。

5.2.2 系统配置寄存器（SYSCFG）

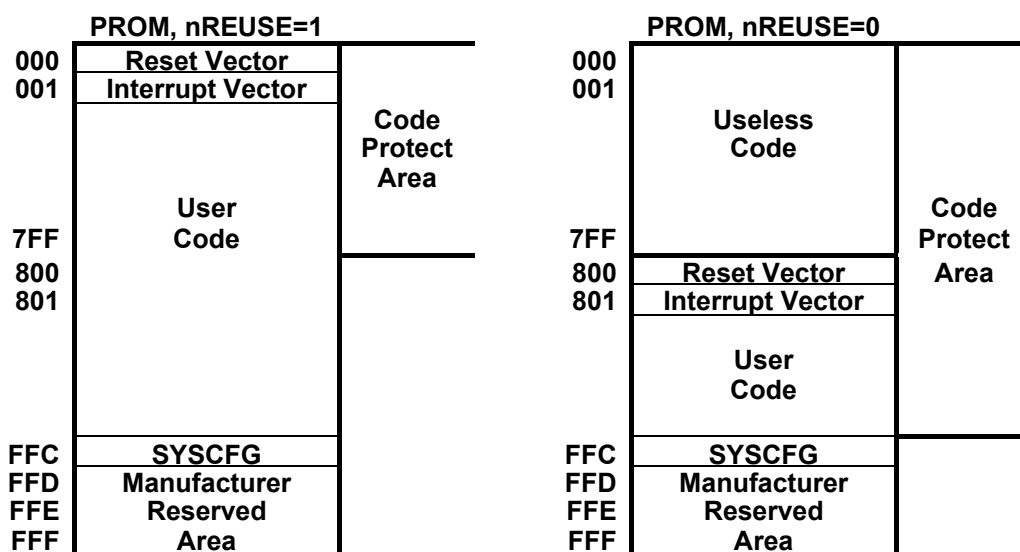
系统配置寄存器（SYSCFG）位于 ROM 的地址是 0FFCh，其功能选项由 MCU 初始条件确定。用户只能通过写入 SYSCFG 值来选择时钟源，LVR 阈值电压和芯片工作模式。SYSCFG 的第 13 位是代码保护选择位，如果此位为 0，当用户读取程序时，程序中的数据将被保护。

位	13~0	
初始值	11_1111_111x_xxxx	
位	描述	
13	nPROTECT : 代码保护选择位	
	1	不保护
	0	保护
12	nREUSE : 程序重新写入控制位	
	1	非重新写入
	0	重新写入 (TM57PA20 不支援)
11-10	LVR:LV 复位模式	
	11	LVR 阈值是 2.1V, 总是处于使能状态
	10	LVR 阈值是 2.1V, 在省电模式 LVR 处于非使能状态
	01	LVR 阈值是 3.1V, 总是处于使能状态
	00	LVR 非使能
9-8	CLKS : 时钟源选择位	
	11	Fast Xtal (455kHz~24MHz)
	10	Slow Xtal (32KHz)
	01	内部 RC (4MHz)
	00	外部 RC
7	XRESETE : 外部管脚复位选择位	

	1	外部管脚复位使能
	0	外部管脚复位非使能
	WDTE: 看门狗复位选择位	
6	1	看门狗复位使能, WKT 定时器非使能
	0	看门狗复位非使能, WKT 定时器使能
4-0	IRCF: 内部 RC 频率调整控制位	

5.2.3 可重复编程

TM57PA40 的程序存储器有 4K 字，比一些少于 2K 字的 F/W 程序存储器空间要大很多。为了充分利用程序存储器，TM57PA40 允许编程者重复编程，故称为可两次编程存储器。当程序存储器的前一部分被无用的程序代码占用着，后一部分空间仍然是空白的，编程者就可在程序存储器的后半部分重写程序。在第二次编程的情况下，复位向量和中断向量会通过编译在程序的后半部分开头重新分配。编程者只需在 ICE 工具界面选择“再利用”选项，然后编译器将目标代码放在适当的位置。也就是说，程序的复位值仍然在地址 000h，但汇编对象代码的复位值在 800h。在 SYSCFG 寄存器中，如果 **nPROTECT** = 0 和 **nREUSE** = 1，该代码保护区就在程序存储器的上半部分。这就允许在第二次编程时，编程者重新写入然后检验。把重新写入的程序代码放在程序存储器的后半部分以后，“**nREUSE**”写入“0”，代码保护区域就变成除保留位的整个程序存储区了。



TM57PA20 的程序存储器有 2K 字 (0x000~0x7ff)，其中 位址 0xffc~fff 是保留區。
TM57PA20 没有支援重新写入 功能 (系统配置寄存器 (SYSCFG) 的 bit 12 不支援)。

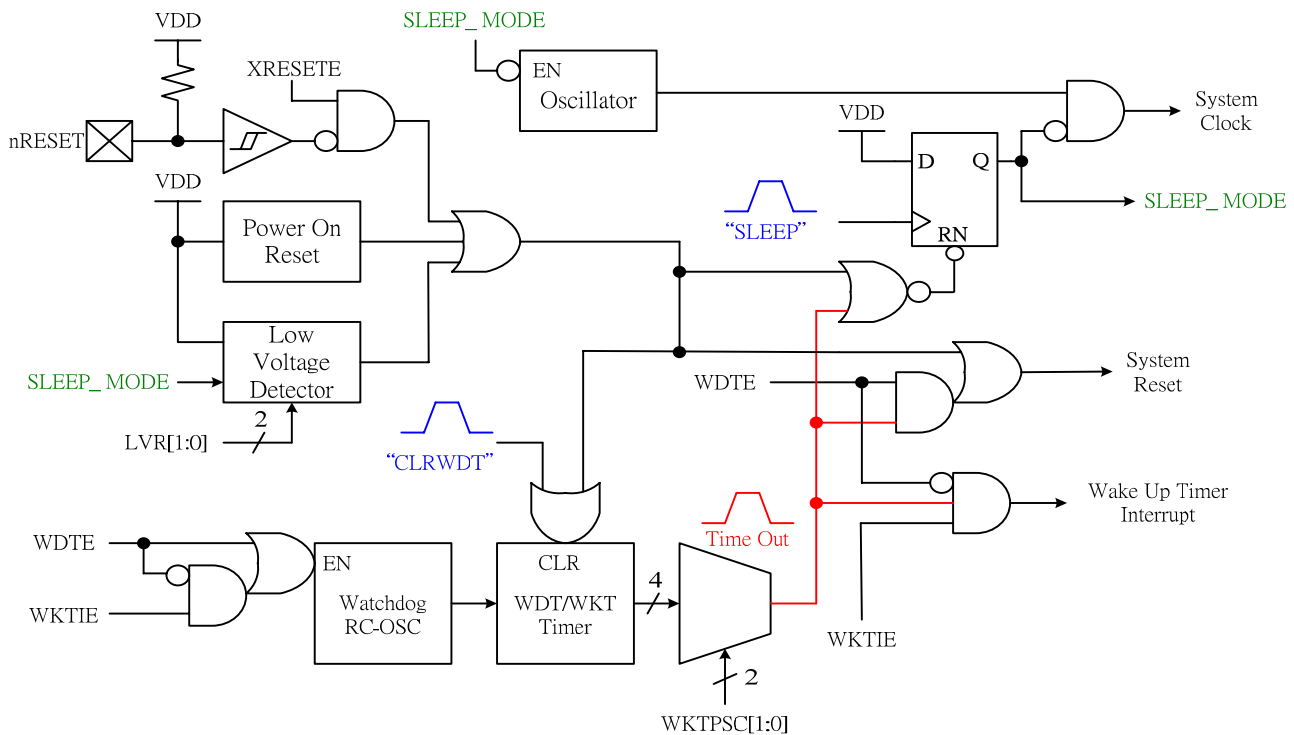
5.2.4 省电模式

省电模式由 SLEEP 指令激活。在省电模式中，系统时钟和所有外围电路都停止工作以最小化电源消耗，当然 WDT/WKT 定时器是否工作由 F/W 设置来决定。省电模式可因复位或使能的中断（外部管脚引起的中断和 WKT 中断）而结束。

5.3 外围功能图

5.3.1 WDT/WKT 定时器

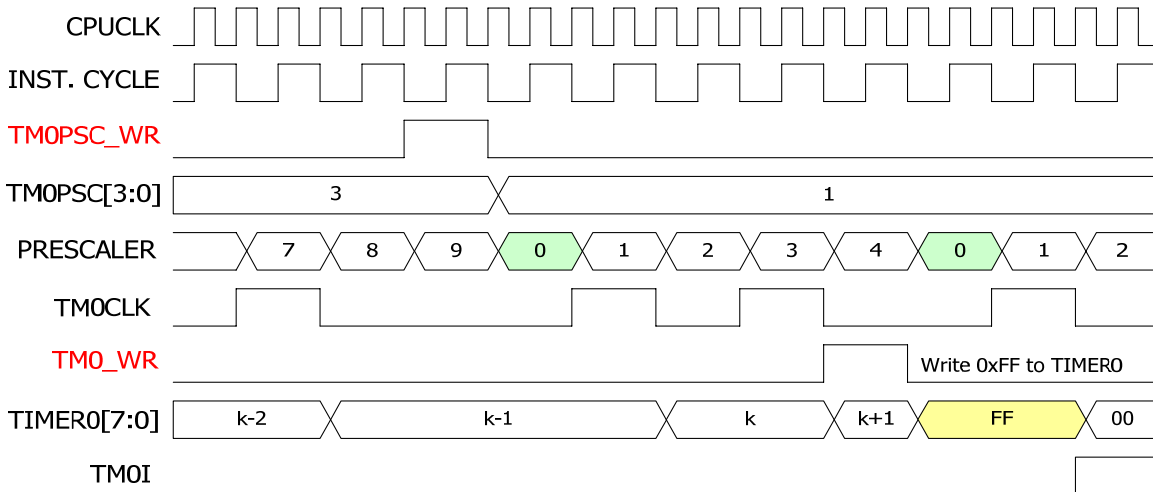
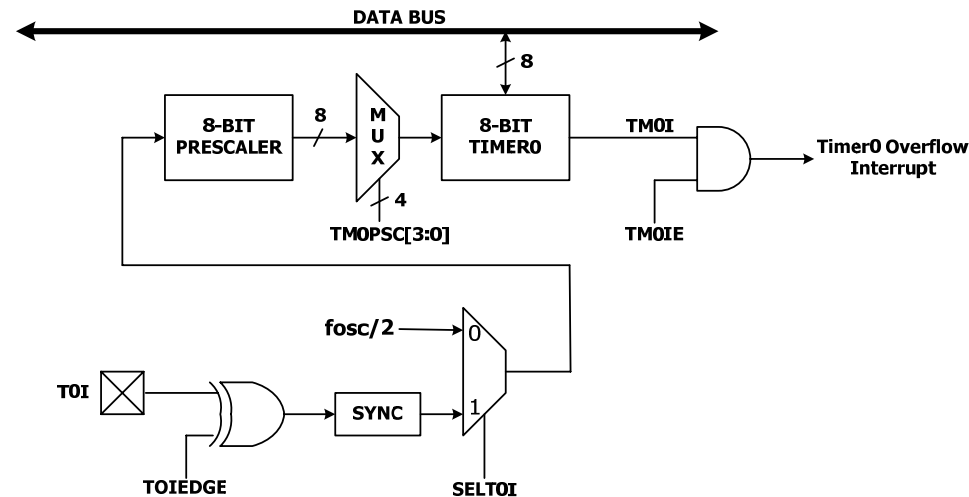
WDT/WKT 是同一个内部 RC 定时器，其溢出周期可在 13 毫秒到 140 毫秒之间选择。WDT/WKT 位由 CLRWDT 指令来清零。若 WDT 复位使能（即 WDTE=1），WDT 就产生芯片复位信号，即使已进入省电模式。否则 WKT 仅产生溢出超时中断。若 WDTE=0，WKTIE=0（唤醒中断非使能），内部 RC 定时器停止工作。



5.3.2 定时器 0：可扩展的 8 位定时计数器（PSC）

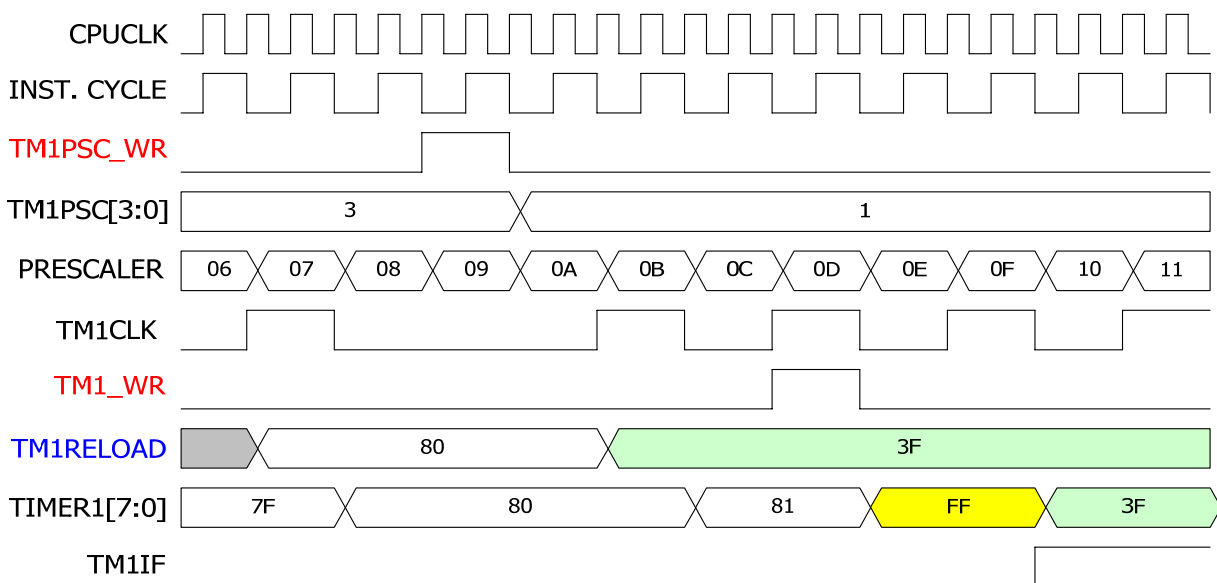
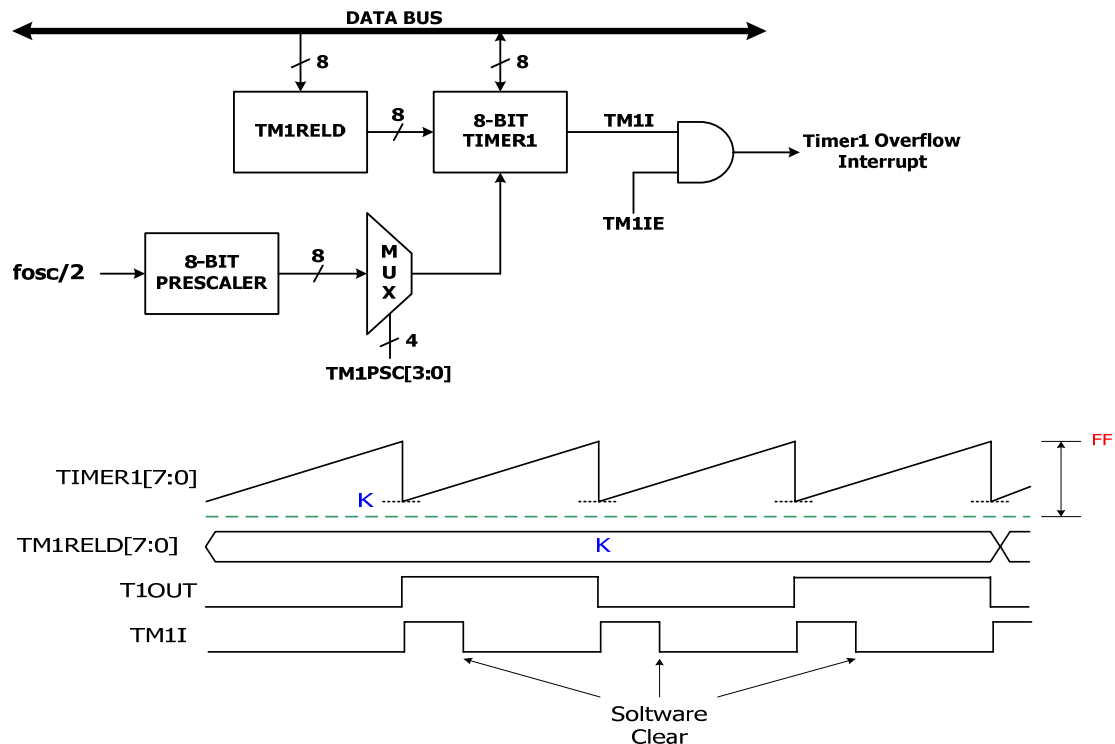
可扩展的 8 位定时计数器（PSC）

定时器 0 是 F-Plane 的 8 位宽寄存器，其可作为 F-Plane 的任何寄存器读写。另外，定时器 0 增加其定期和自动工作卷，是基于可扩展的时钟源，可扩展的时钟源可能是指令周期或 TOI 输入。定时器 0 增加工作卷是由 R-Plane 的“定时器 0 预扩展”寄存器决定的。当定时器 0 的扩展工作卷使用完了，定时器 0 就会产生一个中断（TMOI）。



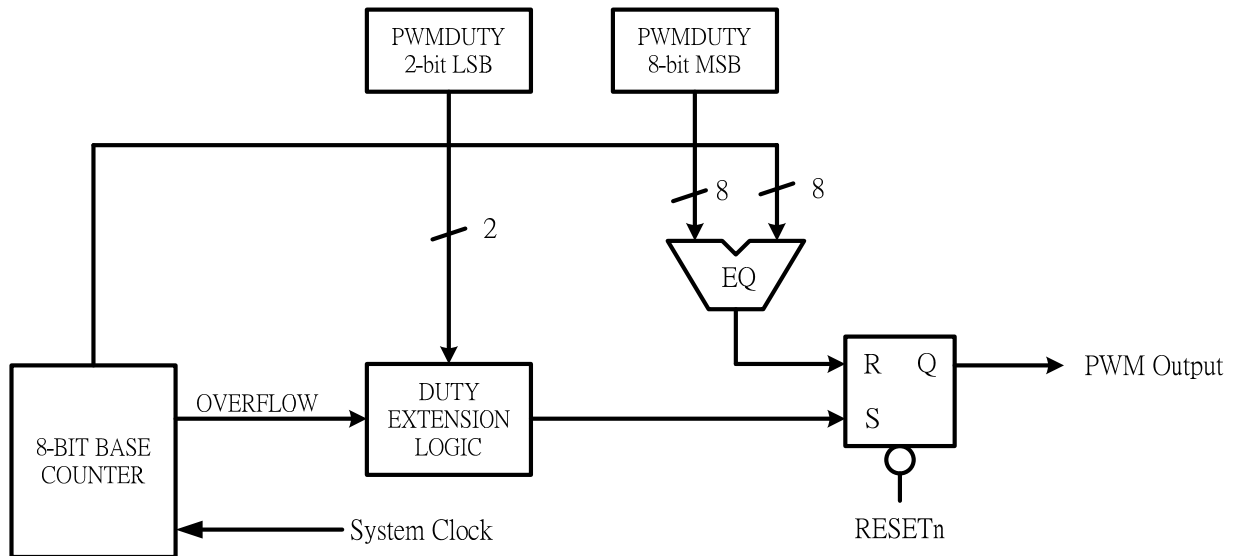
5.3.3 定时器 1：可扩展的 8 位定时器（PSC）

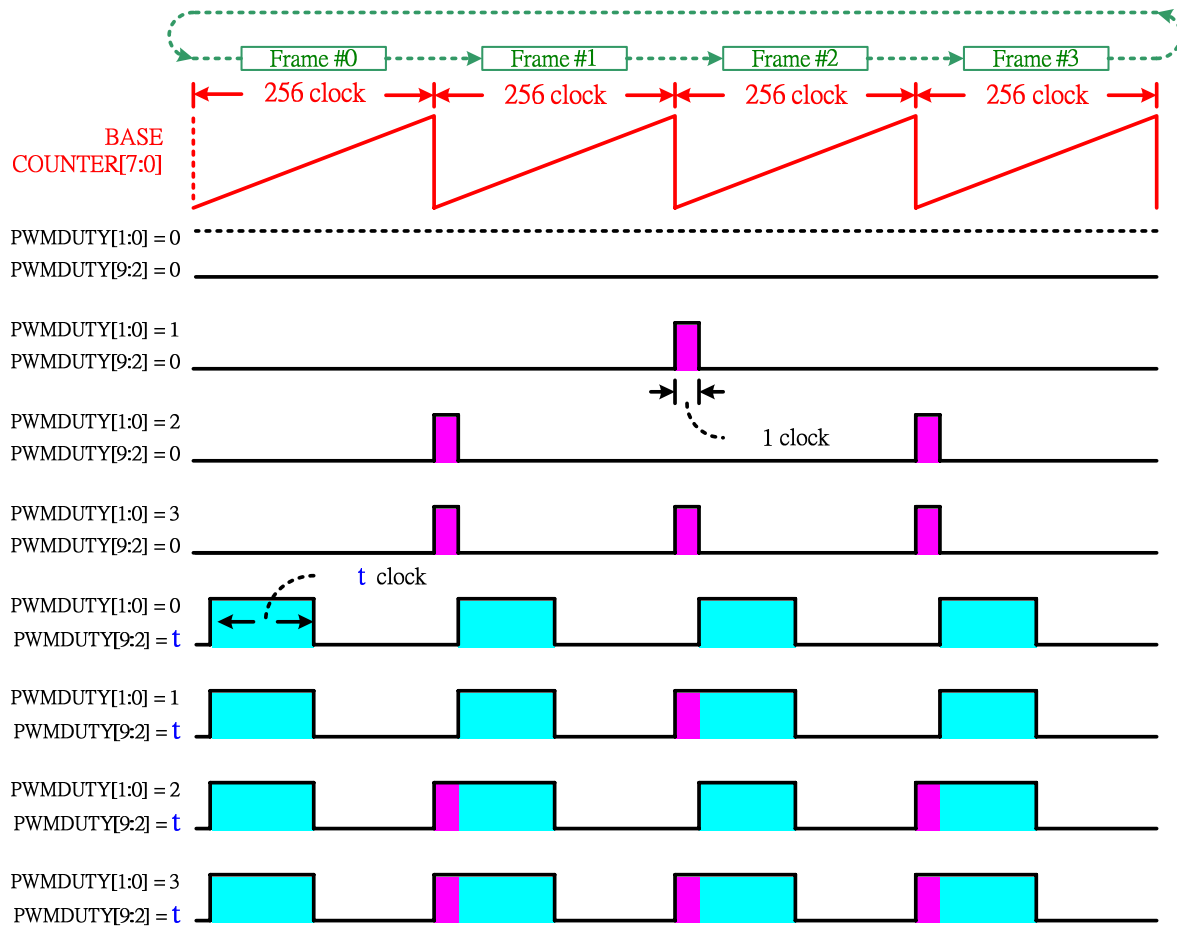
定时器 1 是 F-Plane 的 8 位宽寄存器，其可作为 F-Plane 的任何寄存器读写。另外，定时器 1 可增加其周期时间，并自动加载一个新的“计数值”（TM1RELD），其工作卷的增是基于可扩展的指令时钟。定时器 0 增加工作卷是由 R-Plane 的“定时器 1 预扩展”寄存器（TM1PSC）决定的。当定时器 1 的扩展工作卷使用完了，定时器 1 就会产生 TM1I 中断和 T1OUT 触发信号。



5.3.4 8+2 位 PWM

PWM0 和 PWM1 是一样的电路结构。PWM 基于系统时钟产生固定的 1024 占空比决议频率波形。LSB 扩展技术允许 PWM 以“系统时钟预分频 256”而不是“系统时钟预分频 1024”，这就意味着 PWM 运行的速度是正常的。高一些的 PWM 频率的优点是后 RC 滤波器可改变 PWM 信号以更加稳定直流电压值。无论 8 位基计数器和 PWM 占空比寄存器 (PWMDUTY) 的 8 位 MSB 是否匹配，PWM 输出信号都可重置低位。当基计数器工作卷使用完毕，PWM 占空比寄存器的 2 位 LSB 决定是否立即还是延迟一个时钟周期来设置 PWM 输出信号为高。





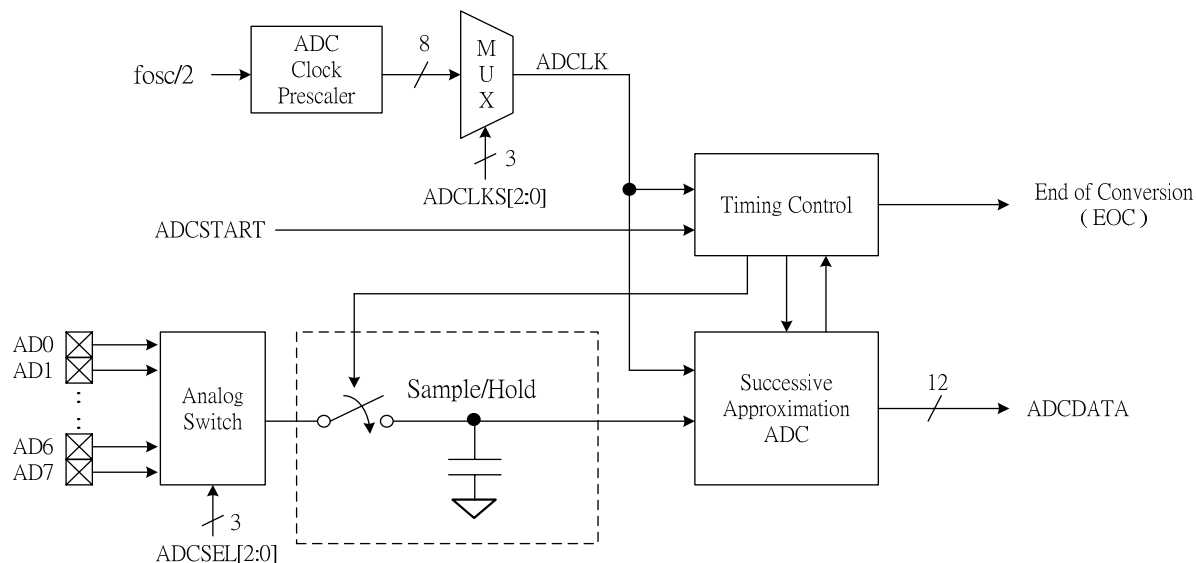
PWM 程序代码举例 如下：

```

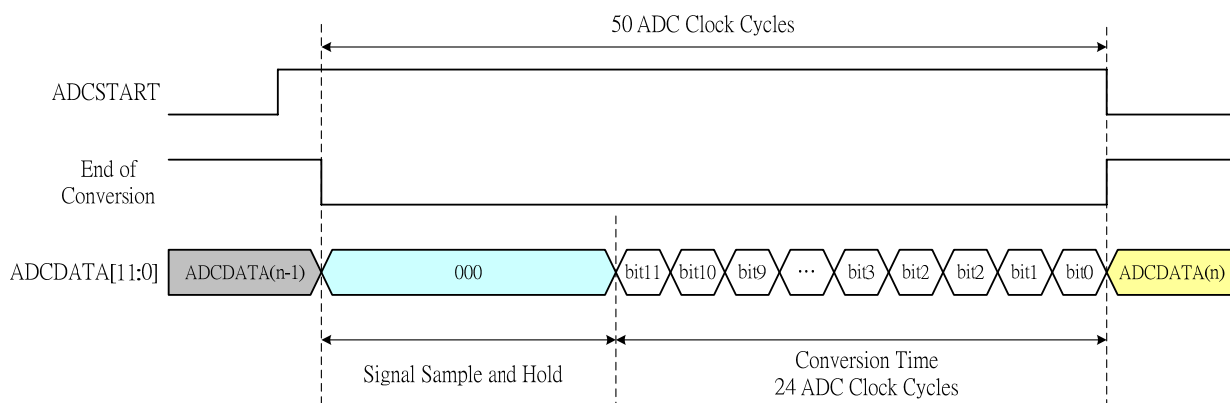
movlw    01111111b
movwf    0ch          ;set PWM0DUTY[9:2]=8'b01111111
movlw    11000000b
movwf    0dh          ;set PWM0DUTY[1:0]=2'b11
movlw    01000000b
movwr    0bh          ;enable PWM0 output to PA0 (PWM0_OUT)
:
:
movlw    00h
movwr    0bh          ;disable PWM0 (PWM0_OUT)

```

5.3.5 12 位数模转换器



12-bit ADC (模数转换器) 由一个 8 通道输入多路复用器，控制寄存器，时钟产生器，12-bit 逐次逼近 AD 转换器和输出寄存器组成。使用数模转换器，用户需要设置 ADCLKS 选择适当的 ADC 时钟频率，它必须小于 2MHz，然后通过设置 ADCSTART 控制位来启动该 ADC 转换。转换结束后，H/ W 自动清零 ADCSTAT 位。用户可以查看这个位单元来了解转换状态。nADC_IE 控制寄存器用于模数转换器引脚类型设置，当相应的引脚被用作 ADC 输入时，用户写入相应位为“0”即可。该设置可以使引脚逻辑输入路径非使能以节省电力消耗。



ADC 程序举例：

```
movlw    00000111b
movwf    11h          ;ADC channel select,ADC7(PB0) (ADCSEL)
```

```

movlw      00000001h
movwr      09h          ;disable PB0 pull up resistor  (nPBPUP)

movlw      01111111b
movwr      12h          ;set ADC7 input enable (nADC_IE)

movlw      00010000b
movwr      0ch          ;set ADC clock is instruction cycle / 64 (ADCCLKS)

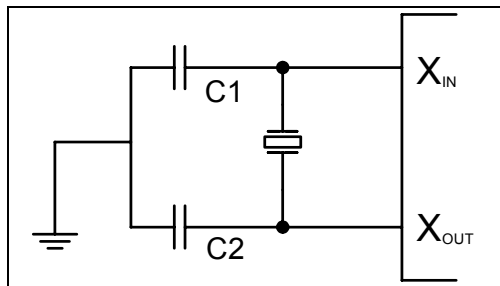
ADC_LOOP:  bsf          11h,3      ;start  ADC conversion (ADCSTART)
           btfsc        11h,3
           goto         ADC_LOOP  ;wait ADCSTART go LOW

           :                  ;read ADCQ[11:0] (ADCQDQ)
           :

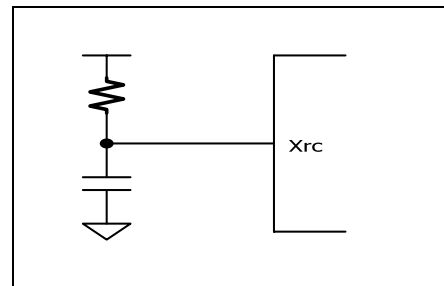
```

5.3.6 系统时钟振荡器

系统时钟振荡器可工作在四种不同的振荡模式中，应用哪种振荡模式通过设置 SYSCFG 寄存器的 CLKS 位来选择。在快/慢晶体模式中，一个晶体或陶瓷电阻被连接在 Xin 和 Xout 管脚来用做振荡器。在外部 RC 模式中，外部电阻和电容决定了振荡频率。在内部 RC 模式中，内置振荡器产生 4MHZ 的振荡时钟。



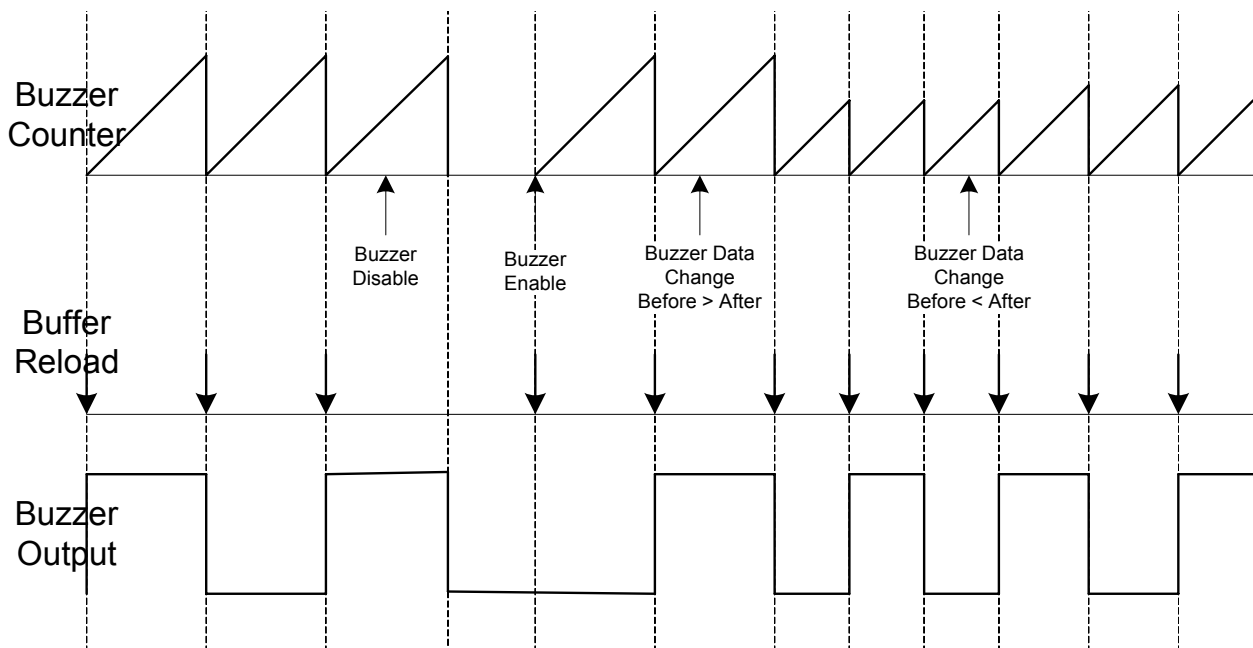
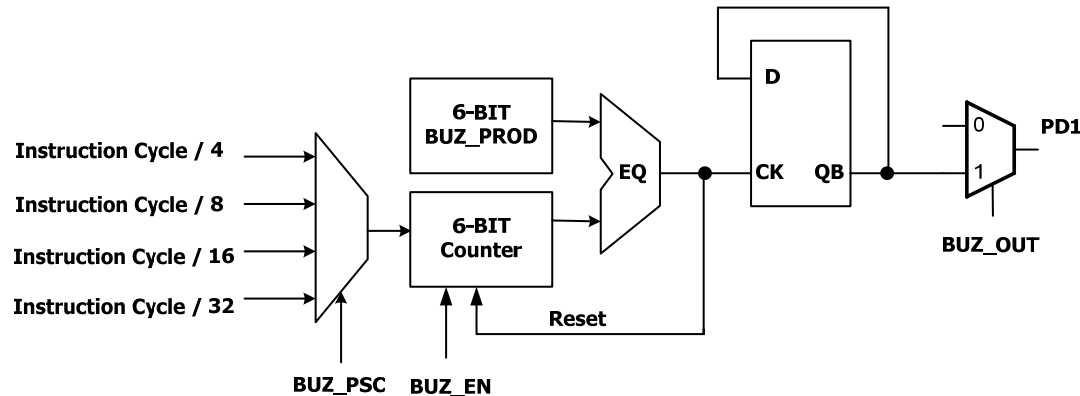
External Oscillator Circuit
(Crystal or Ceramic)



External RC Oscillator

5.3.7 蜂鸣输出

TM59PA80 (TM57PA40)的蜂鸣驱动器包含有 6-BIT 的计数器和时钟分频器。它可产生 50%(占空比) 的工作方波，并且其频率覆盖一个较宽的范围。使用蜂鸣输出功能，需要设置蜂鸣使能控制位（BUZ-EN）和蜂鸣输出使能控制位（BUZ-OUT）。



BUZ-PROD [5 : 0] 决定输出频率。频率计算如下：

$$F_{BZ} = (f_{osc}/2) / (\text{Instruction Cycle Divider}) / (\text{BUZ_PROD} + 1)$$

计算输出频率：

CPU Clock (f_{osc}) = 8192KHz

Instruction Cycle = $f_{osc}/2 = 8192\text{KHz}/2 = 4096\text{KHz}$

Prescaler Ratio (BUZ_PSC) = 2'b11 (Instruction Cycle Divider = 32)

Period Data (BUZ_PROD) = 9

$$F_{BZ} = (8192\text{KHz}/2) / 32 / (9+1) = 12.8 \text{ (KHz)}$$

BUZZER example code:

```
movlw      10000000b
movwr      0bh      ; enable BUZZER output to PD1 (BUZ_OUT)

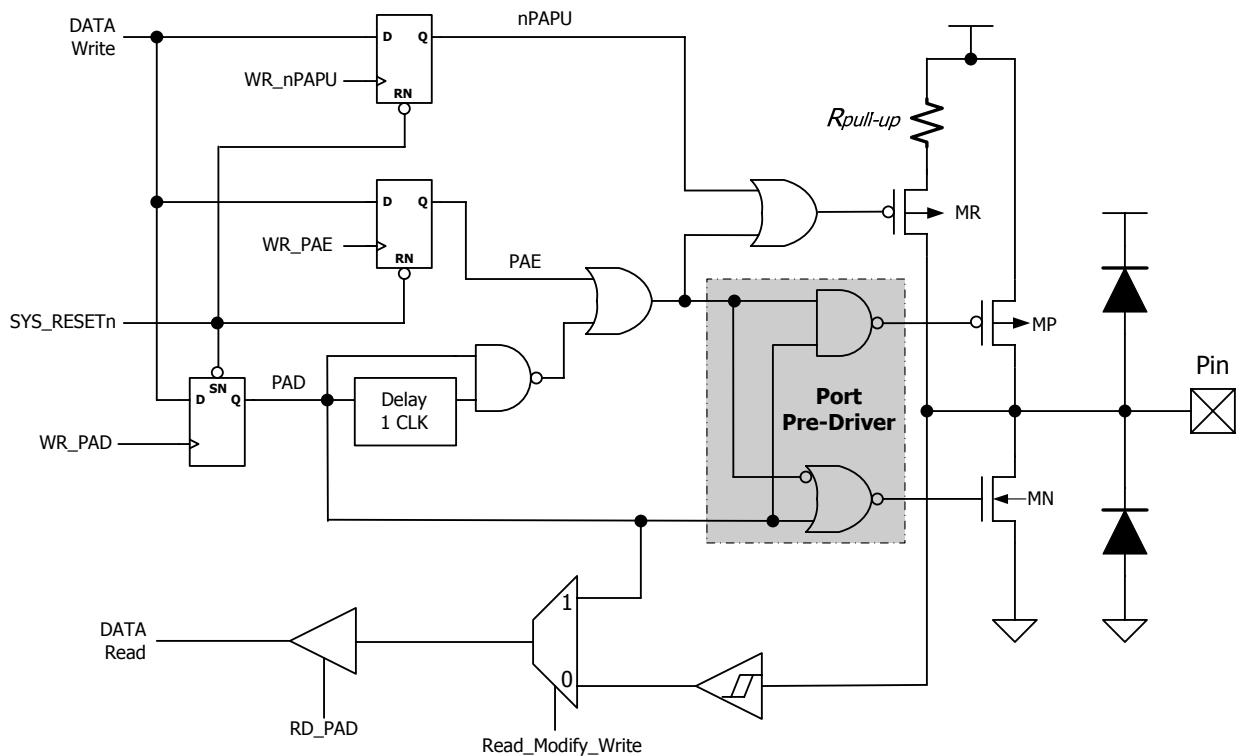
movlw      11001001b ; (fosc/2)/32 (BUZ_PSC)
movwr      10h      ; Period=9 (BUZ_PROD)

movlw      80h
movwr      0ch      ; enable BUZZER counting (BUZ_EN)
:
:
movlw      00h
movwr      0ch      ; disable BUZZER counting (BUZ_EN)
```

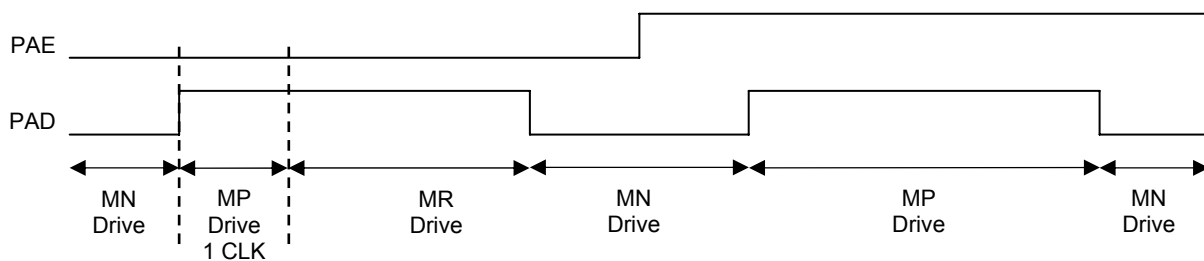
5.4 I/O □

5.4.1 PA0-2

PA0-2 这三个管脚可被用作施密特触发输入，CMOS 推挽输出或“伪漏式”输出。分配给每个引脚的上拉电阻是由 S / W 设置。在施密特触发输入模式，S / W 需要设置 PAE= 0, PAD= 1；在伪漏式输出模式，S / W 设置 PAE= 0。伪漏式结构的优点是其输出上升时间可以大大快于纯漏式结构。S/W 设置 PAE=1 可使用 CMOS 推挽输出模式。读引脚数据（PAD）具有不同的含义。在“读-修改-写”指令中，CPU 实际上读取输出数据寄存器。在别的指令中，CPU 读取引脚的状态。所谓的“读-修改-写”指令包括 BSF，BCF 以及所有把 F -Plane 做目的地的指令。

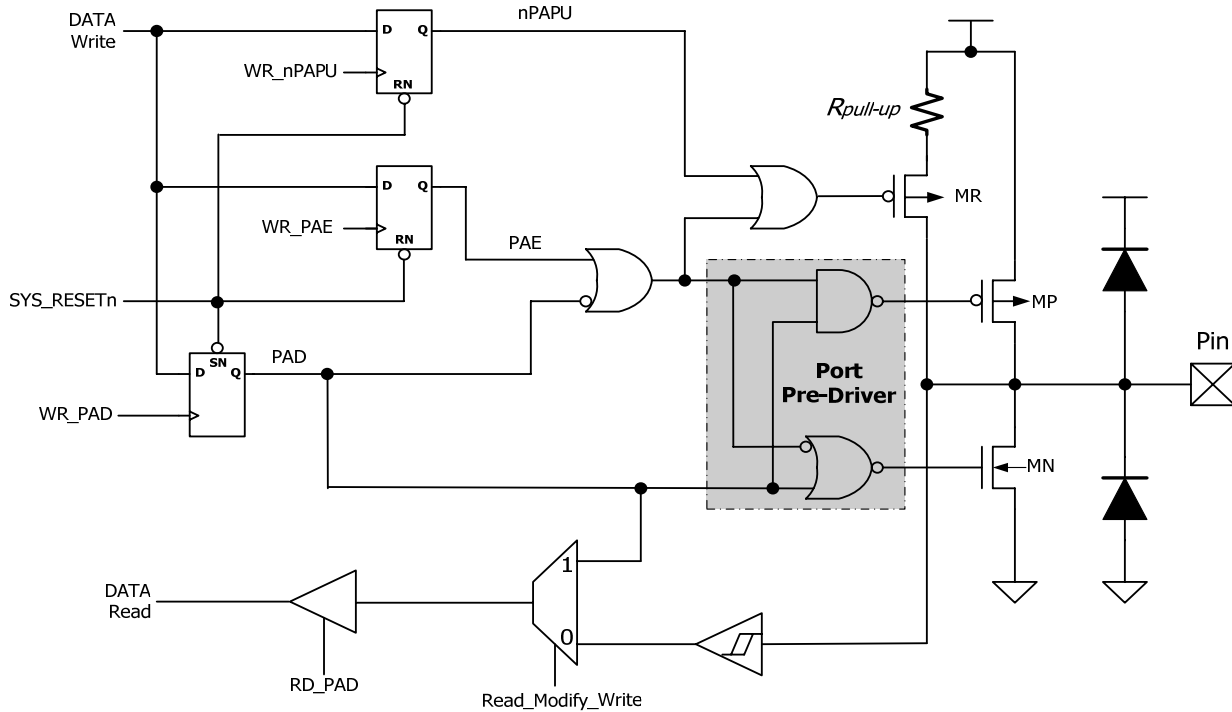


PA0-2, nPAPU=0

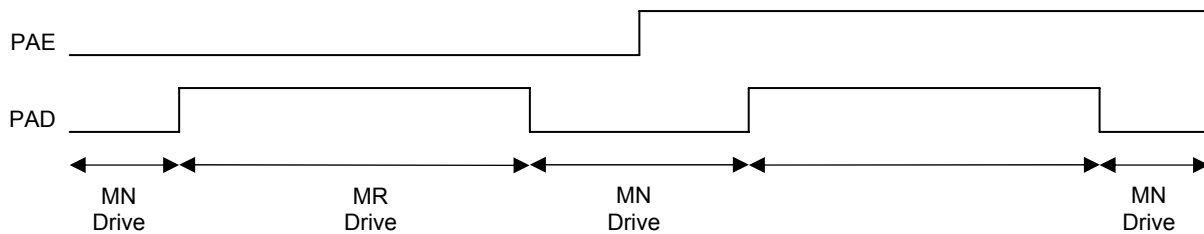


5.4.2 PA3-6, PB0-1, PD0-7

除开不支持伪开漏式模式外，PA3-6, PB0-1, PD0-7 这些管脚和 PA0-2 可被用作施密特触发输入，CMOS 推挽输出。不过，这些管脚可用于纯开漏模式中。

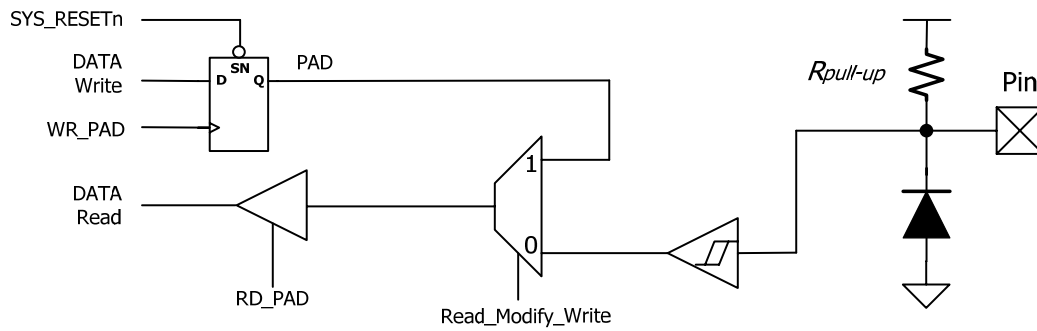


PA3-6, nPAPU=0



5.4.3 PA7

PA7 只可用于施密特触发输入模式中，这个管脚总是和上拉电阻连接。



6. 内存功能图

F-Plane

名称	地址	R/W	复位值	描述
INDF	00.7~0	R/W	-	INDF 寄存器不是一个物理寄存器，寻址 INDF 实际上就是寻址放在 FSR 寄存器里面的值的寄存器。
TIMER0	01.7~0	R/W	0	定时器 0 的值
PC	02.7~0	R/W	0	程序计数器[7~0]
RAMBANK	03.5	R/W	0	RAM 区选择位
TO	03.4	R	0	WDT 时间溢出标志，用'SLEEP'/'CLRWDI'指令可清零
PD	03.3	R	0	睡眠模式标志，用'CLRWDI'指令可清零
ZFLAG	03.2	R/W	0	零标志位
DCFLAG	03.1	R/W	0	十进制加法标志位
CFLAG	03.0	R/W	0	加法标志位
FSR	04.6~0	R/W	-	文件选择寄存器
PA[7]	05.7	R	-	PA[7]管脚状态位
PAD[6 : 0]	05.6~0	R	-	PA 引脚或“数据寄存器”状态
		W	7F	PA 输出数据寄存器
PBD[1:0]	06.1~0	R	-	PB 引脚或“数据寄存器”状态
		W	3	PB 输出数据寄存器
PDD[7:0]	07.7~0	R	-	PD 引脚或“数据寄存器”状态
		W	FF	PD 输出数据寄存器
TM1I	09.5	R	-	定时器 1 中断标志位，定时器 1 溢出时 H/W 置 1
		W	0	置 0：该标志位清零；置 1：无效
TM0I	09.4	R	-	定时器 0 中断标志位，定时器 0 溢出时 H/W 置 1
		W	0	置 0：该标志位清零；置 1：无效
WKT1	09.3	R	-	唤醒定时器中断标志位，定时器溢出时 H/W 置 1
		W	0	置 0：该标志位清零；置 1：无效
XINT2	09.2	R	-	PA7 中断标志位，PA7 下降沿时 H/W 置 1
		W	0	置 0：该标志位清零；置 1：无效
XINT1	09.1	R	-	PA1 中断标志位，PA1 下降沿时 H/W 置 1
		W	0	置 0：该标志位清零；置 1：无效
XINT0	09.0	R	-	PA6 中断标志位，PA6 下降沿时 H/W 置 1
		W	0	置 0：该标志位清零；置 1：无效
TIMER1	0a.7~0	R/W	0	定时器 1 的值
PWM0DUTY	0c.7~0	R/W	0	PWM0 占空比 8 位 MSB
	0d.7~6	R/W	0	PWM0 占空比 2 位 LSB
PWM1DUTY	0e.7~0	R/W	0	PWM1 占空比 8 位 MSB
	0f.7~6	R/W	0	PWM1 占空比 2 位 LSB
ADCDQ	10.7~0	R	-	AD 转换结果 AFCQ[11: 4]
	11.7~4	R	-	AD 转换结果 AFCQ[3: 0]
ADCSTART	11.3	R	-	AD 转换结束后，H/W 清零该位
		W	0	AD 转换开始时，S/W 置 1 该位
ADCSEL	11.2~0	R/W	0	AD 转换通道选择; 0:ADC0, 1:ADC1,...,7:ADC7
RAM	20~27	R/W	-	内部 RAM-普通区域
	28~7F	R/W	-	内部 RAM-RAM0 区
	28~7F	R/W	-	内部 RAM-RAM1 区

R-Plane

名称	地址	R/W	复位值	描述
TCOUT_PSC	02.7~6	W	0	TCOUT 预扩展 00: TCOUT 输出时钟是“指令周期”/1 01: TCOUT 输出时钟是“指令周期”/2 10: TCOUT 输出时钟是“指令周期”/4 11: TCOUT 输出时钟是“指令周期”/8
T0IEDGE	02.5	W	0	0: 在 T0I (PA2) 上升沿, 定时器 0/PSC 值加 1 1: 在 T0I (PA2) 下降沿, 定时器 0/PSC 值加 1
SELT0I	02.4	W	0	0: 定时器 0/PSC 时钟根据指令周期而来 1: 定时器 0/PSC 时钟根据 T0I 管脚而来
TM0PSC	02.3~0	W	0	定时器 0 预扩展 0000: 定时器 0 输入时钟是“指令周期”/1 0001: 定时器 0 输入时钟是“指令周期”/2 0111: 定时器 0 输入时钟是“指令周期”/128 1000: 定时器 0 输入时钟是“指令周期”/256
PWRDOWN	03	W		对此寄存器进行写操作时进入省电模式
CLRWDT	04	W		对此寄存器进行写操作时会清除 WDT/WKT
PAE	05.6~3	W	0	每个位控制其相应的管脚, 若这个位是 0: 开漏式输出或施密特触发输入 1: CMOS 推挽式输出
	05.2~0	W	0	每个位控制其相应的管脚, 若这个位是 0: 伪开漏式输出或施密特触发输入 1: CMOS 推挽式输出
PBE	06.1~0	W	0	每个位控制其相应的管脚, 若这个位是 0: 开漏式输出或施密特触发输入 1: CMOS 推挽式输出
PDE	07.7~0	W	0	每个位控制其相应的管脚, 若这个位是 0: 开漏式输出或施密特触发输入 1: CMOS 推挽式输出
nPAPU	08.6~0	W	0	每个位控制其相应的管脚, 若这个位是 0: 管脚上拉电阻使能, 除非以下情况: a.管脚输出数据寄存器 (PAD) 是 0 b.管脚选择 CMOS 推挽式 (PAE=1) c.管脚是为晶体或外部 RC 振荡服务 d.管脚是为 PWM0 输出服务 1: 管脚上拉电阻非使能
nPBPU	09.1~0	W	0	每个位控制其相应的管脚, 若这个位是 0: 管脚上拉电阻使能, 除非以下情况: a.管脚输出数据寄存器 (PBD) 是 0 b.管脚选择 CMOS 上拉模式 (PBE=1) c.管脚是为 PWM0 输出服务 1: 管脚上拉电阻非使能
nPDPU	0a.7~0	W	0	每个位控制其相应的管脚, 若这个位是 0: 管脚上拉电阻使能, 除非以下情况: a.管脚输出数据寄存器 (PDD) 是 0 b.管脚选择 CMOS 推挽式 (PDE=1)

				c.管脚是为 T1OUT/BUZZER/TCOUT 输出服务 1: 管脚上拉电阻非使能
BUZ_OUT	0b.7	W	0	0: 蜂鸣输出到 PD1 脚非使能 1: 蜂鸣输出到 PD1 脚非使能
PWM0_OUT	0b.6	W	0	0: PWM0 输出到 PA0 脚非使能 1: PWM0 输出到 PA0 脚非使能
PWM1_OUT	0b.5	W	0	0: PWM0 输出到 PB0 脚非使能 1: PWM0 输出到 PB0 脚非使能
INT1EDGE	0b.4	W	0	0: PA1 下降沿触发中断 1: PA1 上升沿触发中断
TC_OUT	0b.3	W	0	0: 指令时钟驱动器输出到 PD6 管脚非使能 1: 指令时钟驱动器输出到 PD6 管脚使能
TM1_OUT	0b.2	W	0	0: 定时器 1 匹配输出到 PD0 非使能 1: 定时器 1 匹配输出到 PD0 使能
WKT_PSC	0b.1~0	W	11	WDT/WKT 周期 (VDD=5V) 00: 13mS 01: 25mS 10: 50mS 11: 100mS
BUZ_EN	0c.7	W	0	0: 蜂鸣定时器计数非使能 1: 蜂鸣定时器计数使能
ADCLKS	0c.6~4	W	0	000: AD 转换时钟是“指令周期”/128 001: AD 转换时钟是“指令周期”/64 111: AD 转换时钟是“指令周期”/1
TM1_PSC	0c.3~0	W	0	0000: 定时器 1 输入时钟是“指令周期”/1 0001: 定时器 1 输入时钟是“指令周期”/2 0111: 定时器 1 输入时钟是“指令周期”/128 1000: 定时器 1 输入时钟是“指令周期”/256
TM1_RELD	0d.7~0	W	0	当定时器 1 工作卷使用完毕, 定时器 1 重置初始值
TM1IE	0e.5	W	0	定时器 1 中断使能: 1=使能, 0=非使能
TM0IE	0e.4	W	0	定时器 0 中断使能: 1=使能, 0=非使能
WKTIE	0e.3	W	0	唤醒定时器中断使能: 1=使能, 0=非使能
XINT2E	0e.2	W	0	PA7 中断使能: 1=使能, 0=非使能
XINT1E	0e.1	W	0	PA1 中断使能: 1=使能, 0=非使能
XINT0E	0e.0	W	0	PA6 中断使能: 1=使能, 0=非使能
TESTREG	0f.1~0	W	0	测试用寄存器, 只可被生产商使用, 用户不能写入
BUZ_PSC	10.7~6	W	0	蜂鸣时钟预扩展: 00: 蜂鸣时钟是“指令周期”/4 01: 蜂鸣时钟是“指令周期”/8 10: 蜂鸣时钟是“指令周期”/16 11: 蜂鸣时钟是“指令周期”/32
BUZ_PROD	10.5~0	W	0	蜂鸣周期值。蜂鸣输出是蜂鸣时钟/蜂鸣周期值
ADC_TRIM	11.2~0	W	0	测试模式寄存器, 只可被生产商使用, 用户不可用
nADC_IE	12.7~0	W	ff	每个位控制其相应的 ADC7-0 使能管脚。若这个位是 0: 相应的管脚是 ADC 输入口中 1: 相应的管脚是数字输入逻辑输入口

7. 指令表

每个指令都是一个 14 位字被分开成一个 OPCODE，它详细的阐述了指令类型，一个或更多的能进一步详解指令运行的操作数，指令在下面的表格中被分类为字节向导，位向导和文字操作列表。

对字节指令来说，“f”或“r”代表指示地址，“d”代表指示目的单元格。地址指示被用来指定程序存储器的哪位地址被指令在使用。目的单元格指示出操作结果被放置的位置。如果“D”是 0，结果被放在 W 寄存器，否则结果被放在指令指示的地址中。

对位指示指令来说，当“f”代表地址指示时，“b”代表位指示，它用来选择受操作影响的位。对于文字操作，“k”代表文字或者常量。

Field	描述
f	寄存器文件地址
b	位地址
k	文字，常量或标签
d	目的选择项： 0: 工作寄存器 1: 寄存器档案
W	工作寄存器
Z	零标记
C	进位标志
DC	辅助进位标志
PC	程序计数器
TOS	堆栈顶
GIE	Global 中断使能标志 (i-Flag)
()	内容
.	比特项
B	之前
A	之后
←	赋值指示方向

助记符		运算代码	周期	受影响的标志位	描述
面向文件寄存器的字节指令					
<u>ADDWF</u>	f,d	00 0111 dfff ffff	1	C,DC,Z	W 和 “f” 相加
<u>ANDWF</u>	f,d	00 0101 dfff ffff	1	Z	W 和 “f”相与
<u>CLRF</u>	f	00 0001 1fff ffff	1	Z	“f”清零
<u>CLRW</u>		00 0001 0100 0000	1	Z	W 清零
<u>COMF</u>	f,d	00 1001 dfff ffff	1	Z	“f”取反
<u>DECF</u>	f,d	00 0011 dfff ffff	1	Z	“f”减一
<u>DECFSZ</u>	f,d	00 1011 dfff ffff	1 or 2	-	“f”减一, 结果是 0 就跳转
<u>INCF</u>	f,d	00 1010 dfff ffff	1	Z	“f”加一
<u>INCFSZ</u>	f,d	00 1111 dfff ffff	1 or 2	-	“f”加一, 结果是 0 就跳转
<u>IORWF</u>	f,d	00 0100 dfff ffff	1	Z	W 和”f”相或
<u>MOVFW</u>	f	00 1000 0fff ffff	1	-	“f” → “w”
<u>MOVWF</u>	f	00 0000 1fff ffff	1	-	W → “ f”
<u>RLF</u>	f,d	00 1101 dfff ffff	1	C	“f”带进位左移
<u>RRF</u>	f,d	00 1100 dfff ffff	1	C	“f”带进位右移
<u>SUBWF</u>	f,d	00 0010 dfff ffff	1	C,DC,Z	W - “f”
<u>SWAPF</u>	f,d	00 1110 dfff ffff	1	-	“f”的高低半字节互换
<u>TESTZ</u>	f	00 1000 1fff ffff	1	Z	测试“f” 是否为 0,影响 z 标志
<u>XORWF</u>	f,d	00 0110 dfff ffff	1	Z	W 和 “f”相异或
面向文件寄存器的位指令					
<u>BCF</u>	f,b	01 000b bbff ffff	1	-	“f”的“b” bit 清零
<u>BSF</u>	f,b	01 001b bbff ffff	1	-	“f”的“b” bit 置 1
<u>BTFSC</u>	f,b	01 010b bbff ffff	1 or 2	-	“f”的 B 位为 0 跳转
<u>BTFSS</u>	f,b	01 011b bbff ffff	1 or 2	-	“f”的 B 位为 1 跳转
文字和控制指令					
<u>ADDLW</u>	k	01 1100 kkkk kkkk	1	C,DC,Z	“k” + W →W
<u>ANDLW</u>	k	01 1011 kkkk kkkk	1	Z	“k” 和 W 相与→W
<u>CALL</u>	k	10 kkkk kkkk kkkk	2	-	调用子程序 “k”
<u>CLRWDI</u>		00 0000 0000 0100	1	TO,PD	定时监视器清零
<u>GOTO</u>	k	11 kkkk kkkk kkkk	2	-	跳到分支“K”
<u>IORLW</u>	k	01 1010 kkkk kkkk	1	Z	“k” 和 W 相或→W
<u>MOVLW</u>	k	01 1001 kkkk kkkk	1	-	“k” → W
<u>NOP</u>		00 0000 0000 0000	1	-	空指令
<u>RET</u>		00 0000 0100 0000	2	-	从子程序返回
<u>RETI</u>		00 0000 0110 0000	2	-	从中断返回
<u>RETLW</u>	k	01 1000 kkkk kkkk	2	-	W 带 K 值返回
<u>SLEEP</u>		00 0000 0000 0011	1	TO,PD	进入待机模式, 时钟振荡停止
<u>XORLW</u>	k	01 1111 kkkk kkkk	1	Z	“k” 和 W 相异→w

Preliminary

BCF		Clear "b" bit of "f"	
句法	BCF f[,b]		
操作数	f : 00h ~ 3Fh b : 0 ~ 7		
运行	(f.b) ← 0		
影响的状态位	-		
OP-Code	01 000b bbff ffff		
描述	寄存器 'f' 的 'b'位被清零。		
周期	1		
举例	BCF FLAG_REG, 7	B : FLAG_REG = 0xC7	A : FLAG_REG = 0x47
BSF		Set "b" bit of "f"	
句法	BSF f[,b]		
操作数	f : 00h ~ 3Fh b : 0 ~ 7		
运行	(f.b) ← 1		
影响的状态位	-		
OP-Code	01 001b bbff ffff		
描述	寄存器 'f' 的 'b'位被置 1。		
周期	1		
举例	BSF FLAG_REG, 7	B : FLAG_REG = 0x0A	A : FLAG_REG = 0x8A
BTFSC		Test 'b' bit of 'f', skip if clear(0)	
句法	BTFSC f[,b]		
操作数	f : 00h ~ 3Fh b : 0 ~ 7		
运行	若 (f.b) = 0 跳转到下一条指令		
影响的状态位	-		
OP-Code	01 010b bbff ffff		
描述	若寄存器 f 的位 b 是 1，下一条指令被执行。否则下一条指令被放弃，一个 NOP 指令被执行，使这条指令要两个指令周期。		
周期	1 or 2		
举例	LABEL1 BTFSC FLAG, 1	B : PC = LABEL1	
	TRUE GOTO SUB1	A : if FLAG.1 = 0, PC = FALSE	
	FALSE ...	if FLAG.1 = 1, PC = TRUE	
BTFSS		Test "b" bit of "f", skip if set(1)	
句法	BTFSS f[,b]		
操作数	f : 00h ~ 3Fh b : 0 ~ 7		
运行	Skip next instruction if (f.b) = 1		
影响的状态位	-		
OP-Code	01 011b bbff ffff		
描述	若寄存器 F 的 b 位是 0，下一条指令被执行。否则下一条指令被放弃，一个 NOP 指令被执行，使这条指令要两个指令周期。		
周期			
举例	LABEL1 BTFSS FLAG, 1	B : PC = LABEL1	
	TRUE GOTO SUB1	A : if FLAG.1 = 0, PC = TRUE	
	FALSE ...	if FLAG.1 = 1, PC = FALSE	

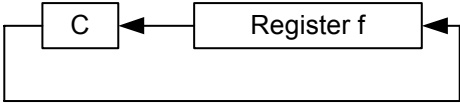
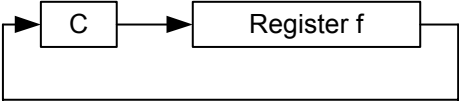
CALL	Call subroutine "k"
句法	CALL k
操作数	K : 00h ~ 7FFh
运行	Operation: TOS $\leftarrow$ (PC)+ 1, PC.10~0 $\leftarrow$ k
影响的状态位	-
OP-Code	10 0kkk kkkk kkkk
描述	调用子程序。首先返回的地址，(PC+1) 被压入堆栈的最高位置。11-bit 的 k 值会立刻装入 PC<10:0>的位置中，PC<12:11..的值则从 PCLATH<4:3>加载。
周期	2
举例	LABEL1 CALL SUB1 B : PC = LABEL1 A : PC = SUB1, TOS = LABEL1+1
CLRF	Clear f
句法	CLRF f
操作数	f : 00h ~ 7Fh
运行	(f) $\leftarrow$ 00h, Z $\leftarrow$ 1
影响的状态位	Z
OP-Code	00 0001 1fff ffff
描述	寄存器 f 被清零，Z 标志被置 1。
周期	1
举例	CLRF FLAG_REG B : FLAG_REG = 0x5A A : FLAG_REG = 0x00, Z = 1
CLRW	Clear W
句法	CLRW
操作数	-
运行	(W) $\leftarrow$ 00h, Z $\leftarrow$ 1
影响的状态位	Z
OP-Code	00 0001 0100 0000
描述	寄存器 W 被清零，Z 标志被置 1。
周期	1
举例	CLRW B : W = 0x5A A : W = 0x00, Z = 1
CLRWD	Clear Watchdog Timer
句法	CLRWD
操作数	-
运行	WDT/WKT Timer $\leftarrow$ 00h
影响的状态位	TO, PD
OP-Code	00 0000 0000 0100
描述	CLRWD 指令会清零看门狗定时器/唤醒定时器。
周期	1
举例	CLRWD B : WDT counter = ? A : WDT counter = 0x00

COMF	Complement f
句法	COMF f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运行	(destination) $\leftarrow$ ($\bar{f}$)
影响的状态位	Z
OP-Code	00 1001 dfff ffff
描述	寄存器 f 的值做补码运算后, 运算结果放在目标寄存器中。
周期	1
举例	COMF REG1,0 B : REG1 = 0x13 A : REG1 = 0x13, W = 0xEC
DECF	Decrement f
句法	DECF f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运行	(destination) $\leftarrow$ (f) - 1
影响的状态位	Z
OP-Code	00 0011 dfff ffff
描述	寄存器 f 的值递减 1。运算结果放在目标寄存器中。
周期	1
举例	DECF CNT, 1 B : CNT = 0x01, Z = 0 A : CNT = 0x00, Z = 1
DECFSZ	Decrement f, Skip if 0
句法	DECFSZ f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运行	(destination) $\leftarrow$ (f) - 1, skip next instruction if result is 0
影响的状态位	-
OP-Code	00 1011 dfff ffff
描述	寄存器 f 的值递减 1。操作结果放在目标寄存器中; 若操作结果不是 0, 下一条指令就会继续执行。若操作结果是 0, 下一条指令被忽略, 一个 NOP 指令被取代执行, 此时会需要两个指令周期的时间。
周期	1 or 2
举例	LABEL1 DECFSZ CNT, 1 GOTO LOOP CONTINUE B : PC = LABEL1 A : CNT = CNT - 1 if CNT=0, PC = CONTINUE if CNT $\neq$ 0, PC = LABEL1+1
GOTO	Unconditional Branch
句法	GOTO k
操作数	k : 00h ~ 7FFh
运行	PC.10~0 $\leftarrow$ k
影响的状态位	-
OP-Code	11 0kkk kkkk kkkk
描述	GOTO 是无条件转移指令。11 个位的 k 值会立即被放在 Pc<11:0>的位置中。PC<12:11>的值则从 PCLATH<4:3>加载。
周期	2
举例	LABEL1 GOTO SUB1 B : PC = LABEL1 A : PC = SUB1

INCF	Increment f
句法	INCF f [,d]
操作数	f : 00h ~ 7Fh
运行	(destination) $\leftarrow$ (f) + 1
影响的状态位	Z
OP-Code	00 1010 dfff ffff
描述	寄存器 f 的值递增 1。操作结果放在目标寄存器中；若操作结果不是 0，下一条指令就会继续执行。若操作结果是 0，下一条指令被忽略，一个 NOP 指令被取代执行，此时会需要两个指令周期的时间。
周期	1 or 2
举例	INCF CNT, 1 B : CNT = 0xFF, Z = 0 A : CNT = 0x00, Z = 1
INCFSZ	Increment f, Skip if 0
句法	INCFSZ f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运行	(destination) $\leftarrow$ (f) + 1, skip next instruction if result is 0
影响的状态位	-
OP-Code	00 1111 dfff ffff
描述	增量寄存器 f。运算结果放在目标寄存器中。若结果是 1，下一条指令被执行。若结果是 0，一个 NOP 指令被执行，使这条指令变成一个二周期指令。
周期	1 or 2
举例	LABEL1 INCFSZ CNT, 1 B : PC = LABEL1 GOTO LOOP A : CNT = CNT + 1 CONTINUE if CNT=0, PC = CONTINUE if CNT≠0, PC = LABEL1+1
IORLW	Inclusive OR Literal with W
句法	IORLW k
操作数	k : 00h ~ FFh
运行	(W) $\leftarrow$ (W) OR k
影响的状态位	Z
OP-Code	01 1010 kkkk kkkk
描述	W 寄存器的值和 K 的值做 or 运算。结果放在 W 寄存器中。
周期	1
举例	IORLW 0x35 B : W = 0x9A A : W = 0xBF, Z = 0
IORWF	Inclusive OR W with f
句法	IORWF f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运行	(destination) $\leftarrow$ (W) OR k
影响的状态位	Z
OP-Code	00 0100 dfff ffff
描述	包含 W 寄存器和 F 寄存器值相或。操作结果存在目标寄存器中。
周期	1
举例	IORWF RESULT, 0 B : RESULT = 0x13, W = 0x91 A : RESULT = 0x13, W = 0x93, Z = 0

MOVF	Move f
句法	MOVF f,d
操作数	f : 00h ~ 7Fh d:0,1
运行	(d) ← (f)
影响的状态位	-
OP-Code	00 1000 dfff ffff
描述	把 f 寄存器中的值移到目标寄存器中，当目标寄存器为 f 寄存器自己时，可以用来测试 f 寄存器的内容是否为 0。
周期	1
举例	MOVF FSR, 0 B : FSR=0X3E A : W =0X3E
MOVFW	Move f to W
句法	MOVFW f
操作数	f : 00h ~ 7Fh
运行	(W) ← (f)
影响的状态位	-
OP-Code	00 1000 0fff ffff
描述	把 f 寄存器中的值移到 W 寄存器中。
周期	1
举例	MOVF FSR, 0 B : W = ? A : W ← f
MOVLW	Move Literal to W
句法	MOVLW k
操作数	k : 00h ~ FFh
运行	(W) ← k
影响的状态位	-
OP-Code	01 1001 kkkk kkkk
描述	K 值装进 W 寄存器中。
周期	1
举例	MOVLW 0x5A B : W = ? A : W = 0x5A
MOVWF	Move W to f
句法	MOVWF f
操作数	F : 00h ~ 7Fh
运行	(f) ← (W)
影响的状态位	-
OP-Code	00 0000 1fff ffff
描述	W 寄存器值移到 f 寄存器中。
周期	1
举例	MOVWF REG1 B : REG1 = 0xFF, W = 0x4F A : REG1 = 0x4F, W = 0x4F

NOP	No Operation
句法	NOP
操作数	-
运行	空操作
影响的状态位	Z
OP-Code	00 0000 0000 0000
描述	空操作
周期	1
举例	NOP -
RETFIE	Return from Interrupt
句法	RETI
操作数	-
运行	$PC \leftarrow TOS, GIE \leftarrow 1$
影响的状态位	-
OP-Code	00 0000 0110 0000
描述	从中断中返回。堆栈被弹出，堆栈顶端的值会被弹进放在 PC 中。全局中断允许位会被设为 1。
周期	2
举例	RETFIE A : PC = TOS, GIE = 1
RETLW	Return with Literal in W
句法	RETLW k
操作数	k : 00h ~ FFh
运行	$PC \leftarrow TOS, (W) \leftarrow k$
影响的状态位	-
OP-Code	01 1000 kkkk kkkk
描述	W 寄存器装进“K”值。堆栈中弹出的值放入 PC 中。
周期	2
举例	CALL TABLE B : W = 0x07 : A : W = value of k8 TABLE ADDWF PCL,1 RETLW k1 RETLW k2 : RETLW kn
RETURN	Return from Subroutine
句法	RET
操作数	-
运行	$PC \leftarrow TOS$
影响的状态位	-
OP-Code	00 0000 0100 0000
描述	从子程序中返回。堆栈顶端的值会被弹出放在 PC 中。
周期	2
举例	RETURN A : PC = TOS

RLF	Rotate Left f through Carry
句法	RLF f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运行	
影响的状态位	C
OP-Code	00 1101 dfff ffff
描述	C 位放在 F 值的 MSB 之前当作位 9，然后每个位向左边移动，C 位则位移到位 0，操作结果存于目标寄存器中。
周期	1
举例	RLF REG1,0 B : REG1 = 1110 0110, C = 0 A : REG1 = 1110 0110 W = 1100 1100, C = 1
RRF	Rotate Right f through Carry
句法	RRF f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运行	
影响的状态位	C
OP-Code	00 1100 dfff ffff
描述	C 位放在 F 值的 MSB 之前当作位 9，然后每个位向右移动，0 位则位移到 c 位，操作结果存于目标寄存器中。
周期	1
举例	RRF REG1,0 B : REG1 = 1110 0110, C = 0 A : REG1 = 1110 0110 W = 0111 0011, C = 0
SLEEP	Go into standby mode, Clock oscillation stops
句法	SLEEP
操作数	-
运行	-
影响的状态位	TO, PD
OP-Code	00 0000 0000 0011
描述	单片机会进入休眠模式，振荡器也会停止输出。
周期	1
举例	SLEEP -

SUBWF	Subtract W from f
句法	SUBWF f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运行	$(W) \leftarrow (f) - (W)$
影响的状态位	C, DC, Z
OP-Code	00 0010 dfff ffff
描述	寄存器 'f' 的值减去 W 寄存器中的值(2 的 补码运算)。运算结果存于目标寄存器中。
周期	1
举例	SUBWF REG1,1 B : REG1 = 3, W = 2, C = ?, Z = ? A : REG1 = 1, W = 2, C = 1, Z = 0 SUBWF REG1,1 B : REG1 = 2, W = 2, C = ?, Z = ? A : REG1 = 0, W = 2, C = 1, Z = 1 SUBWF REG1,1 B : REG1 = 1, W = 2, C = ?, Z = ? A : REG1 = FFh, W = 2, C = 0, Z = 0
SWAPF	Swap Nibbles in f
句法	SWAPF f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运行	$(\text{destination}, 7 \sim 4) \leftarrow (f.3 \sim 0), (\text{destination}.3 \sim 0) \leftarrow (f.7 \sim 4)$
影响的状态位	-
OP-Code	00 1110 dfff ffff
描述	寄存器 f 的高低半位值互换。运算结果放在目标寄存器中。
周期	1
举例	SWAPF REG1, 0 B : REG1 = 0xA5 A : REG1 = 0xA5, W = 0x5A
TESTZ	Test if 'f' is zero
句法	TESTZ f
操作数	f : 00h ~ 7Fh
运行	Set Z flag if (f) is 0
影响的状态位	Z
OP-Code	00 1000 1fff ffff
描述	若寄存器 f 的内容是 0，零标志被设置为 1。
周期	1
举例	TESTZ REG1 B : REG1 = 0, Z = ? A : REG1 = 0, Z = 1
XORLW	Exclusive OR Literal with W
句法	XORLW k
操作数	k : 00h ~ FFh
运行	$(W) \leftarrow (W) \text{ XOR } k$
影响的状态位	Z
OP-Code	01 1111 kkkk kkkk
描述	'K' 的值和 W 寄存器的值做 XOR 运算。运算结果放进 W 寄存器中。
周期	1
举例	XORLW 0xAF B : W = 0xB5 A : W = 0x1A

XORWF	Exclusive OR W with f
句法	XORWF f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运行	(destination) ← (W) XOR (f)
影响的状态位	Z
OP-Code	00 0110 dfff ffff
描述	f 寄存器的值和 W 寄存器的值做 XOR 运算。运算结果存于目标寄存器中。
周期	1
举例	XORWF REG 1 B : REG = 0xAF, W = 0xB5 A : REG = 0x1A, W = 0xB5

8. 电气特性

1. 最大绝对额定值 (TA = 25°C)

参数	范围	单位
电源电压	- 0.3 to + 5.5	V
输入电压	- 0.3 to $V_{DD} + 0.3$	
输出电压	- 0.3 to $V_{DD} + 0.3$	
每个 PIN 的最大拉电流	- 25	mA
所有 PIN 的最大拉电流	- 80	
每个 PIN 的最大灌电流	+ 30	
所有 PIN 的最大灌电流	+ 150	
最大工作电压	5.5	V
工作温度	- 40 to + 85	°C
存储温度	- 65 to + 150	

2. DC 特性 (TA = 25°C, VDD = 2.2 V to 5.5 V; unless otherwise specified)

量	符号	条件	最小	典型值	最大	单位
输入高电压	V_{IH}	All Input, except PA7	$V_{DD} = 5\text{ V}$	$0.44V_{DD}$		V
			$V_{DD} = 3\text{ V}$	$0.32V_{DD}$		V
		PA7	$V_{DD} = 5\text{ V}$	$0.58V_{DD}$		V
			$V_{DD} = 3\text{ V}$	$0.63V_{DD}$		V
输入低电压	V_{IL}	All Input, except PA7	$V_{DD} = 5\text{ V}$		$0.28V_{DD}$	V
			$V_{DD} = 3\text{ V}$		$0.27V_{DD}$	V
		PA7	$V_{DD} = 5\text{ V}$		$0.32V_{DD}$	V
			$V_{DD} = 3\text{ V}$		$0.3V_{DD}$	V
输出高电压	V_{OH}	All Output	$V_{DD} = 5\text{ V}, I_{OH}=7\text{mA}$	4.5		V
			$V_{DD} = 3\text{ V}, I_{OH}=4\text{mA}$	2.7		V
输出低电压	V_{OL}	All Output	$V_{DD} = 5\text{ V}, I_{OL}=20\text{mA}$		0.5	V
			$V_{DD} = 3\text{ V}, I_{OL}=10\text{mA}$		0.3	V
输入漏电流(pin high)	I_{ILH}	All Input	$V_{IN} = V_{DD}$	—	—	1 uA
输入漏电流(pin low)	I_{ILL}	All Input	$V_{IN} = 0\text{ V}$	—	—	-1 uA
电源电流	I_{DD}	Run 10 MHz, No Load	$V_{DD} = 4.5\text{ to }5.5\text{ V}$	—	5	mA
		Run 4 MHz, No Load	$V_{DD} = 2.0\text{ V}$		1	
		Stop mode, No Load	$V_{DD} = 4.5\text{ to }5.5\text{ V}$ (LVR disable)	—	0.5	uA
			$V_{DD} = 4.5\text{ to }5.5\text{ V}$ (LVR enable)		1	
			$V_{DD} = 3.0\text{ V}$ (LVR enable)		1	
系统时钟频率	f_{OSC}	$V_{DD} > \text{LVR}_{th}$	$V_{DD} = 5\text{ V}$	—	24	MHz
			$V_{DD} = 3\text{ V}$		18	
			$V_{DD} = 2.2\text{ V}$		10	
上拉寄存器	R_P	$V_{IN} = 0\text{ V}$ Ports A/B/D	$V_{DD} = 5\text{ V}$		90	k
			$V_{DD} = 3\text{ V}$		200	
		$V_{IN} = 0\text{ V}$ PA7	$V_{DD} = 5\text{ V}$		15	k
			$V_{DD} = 3\text{ V}$		20	

3. 时钟常量 ($T_A = 25^\circ\text{C}$; unless otherwise specified)

荡器	条件		最小	典型值	最大	单位
外部 RC	$V_{DD} = 3\text{V}$	$R = 5\text{K}$	$C = 33\text{pF}$	—	2.3	MHz
		$R = 10\text{K}$	$C = 100\text{pF}$	—	0.85	
		$R = 100\text{K}$	$C = 300\text{pF}$	—	0.046	
	$V_{DD} = 5\text{V}$	$R = 5\text{K}$	$C = 33\text{pF}$	—	2.8	
		$R = 10\text{K}$	$C = 100\text{pF}$	—	0.74	
		$R = 100\text{K}$	$C = 300\text{pF}$	—	0.032	
内部 RC	$V_{DD} = 4.75 \text{ to } 5.25 \text{ V}$		Typ-2%	3.9	Typ+2%	
	$V_{DD} = 2.8 \text{ to } 3.2 \text{ V}$		Typ-4%	3.83	Typ+4%	

4. 复位定时特性 ($T_A = 25^\circ\text{C}$, $V_{DD} = 2.0 \text{ V to } 5.5 \text{ V}$)

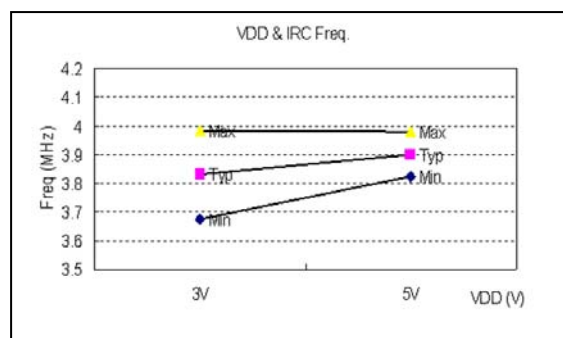
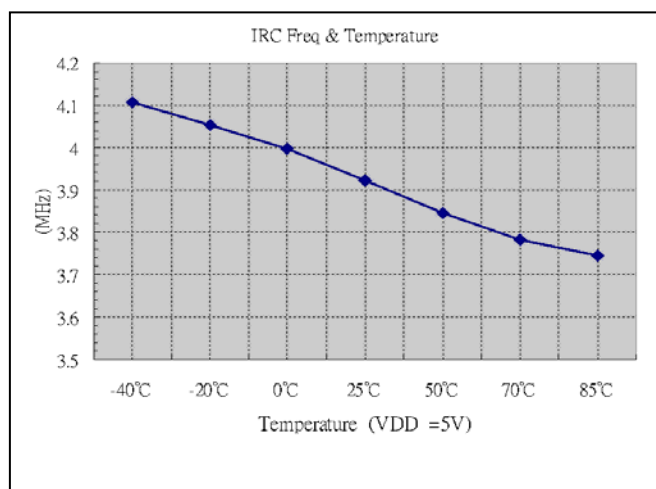
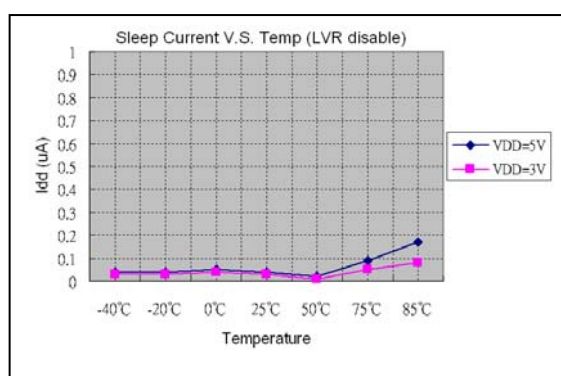
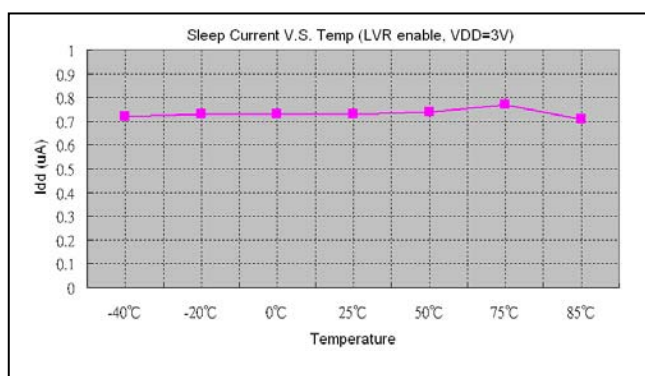
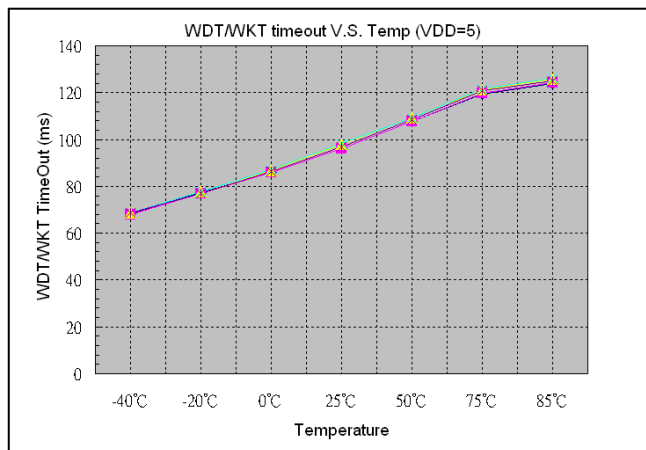
参量	条件	最小值	参考值	最大值	单位
复位输入低脉宽时间	Input $V_{DD} = 5 \text{ V} \pm 10 \%$	3	—	—	μs
WDT 唤醒时间	$V_{DD} = 5 \text{ V}$, WKTPSC = 11	Typ-15%	100	Typ+15%	mS
	$V_{DD} = 3 \text{ V}$, WKTPSC = 11	Typ-15%	128	Typ+15%	
CPU 启动时间	$V_{DD} = 5 \text{ V}$	—	3.5	-	mS

5. LVR 电路特性($T_A = 25^\circ\text{C}$, $V_{DD} = 5.0 \text{ V}$)

数	条件	最小	典型值	最大	单位
LVR 相关电压	V_{LVR}	1.8	2.1	2.3	V
		2.9	3.1	3.5	
LVR 滞后电压	V_{HYST}	—	± 0.1	—	V
低电压探测时间	t_{LVR}	10	—	—	μs

6. ADC 电气特性($T_A = 25^\circ\text{C}$, $V_{DD} = 2.0 \text{ V to } 5.5 \text{ V}$, $V_{SS} = 0 \text{ V}$)

	条件	最小	典型值	最大	单位
总正确性	$V_{DD} = 5.12 \text{ V}$, $V_{SS} = 0 \text{ V}$	—	± 2.5	± 4	LSB
积分非线性		—	± 3.2	± 5	
最大输入时钟	—	—	—	2	MHz
转换时间	$f_{ADC} = 2 \text{ MHz}$	—	25	—	μs
输入电压	—	V_{SS}	—	V_{DD}	V

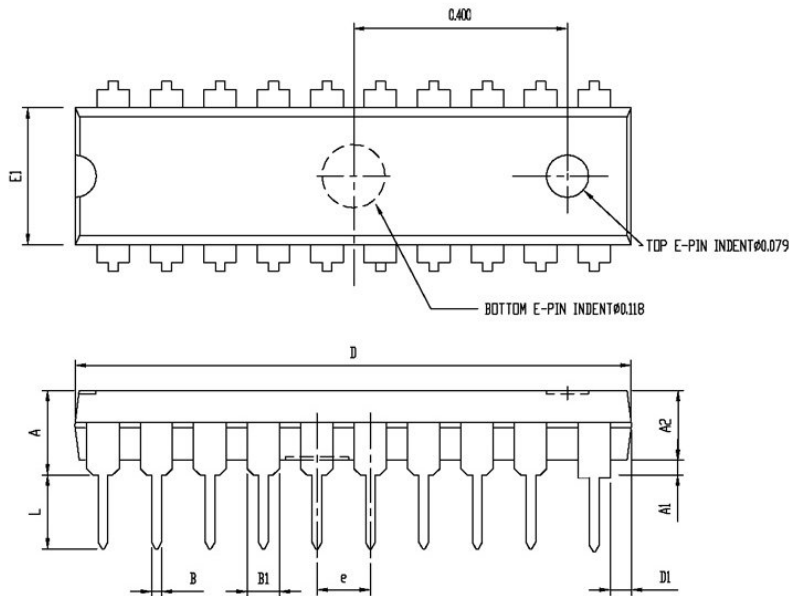


9. 包装说明

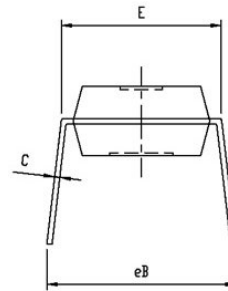
TM57PA40 封装命名：“IC Type” “XX” “YY” “C” “Z”

1. “IC 类型”：TM57PA20 /TM57PA40
2. “XX”：封装类型
 1. DIP 代号：D
 2. SOP 代号：S
 3. SSOP 代号：SS
3. “YY”：IC 插脚数
 1. 插脚数：8 代号：8
 2. 插脚数：16 代号：16
 3. 插脚数：20 代号：20
4. “C”：保留位 (必须写成 “C”)
5. “Z”：包装材料
 1. 包装材料：无铅封装 代号：W
 2. 包装材料：绿色封装 代号：G

● 20-DIP Package Dimension

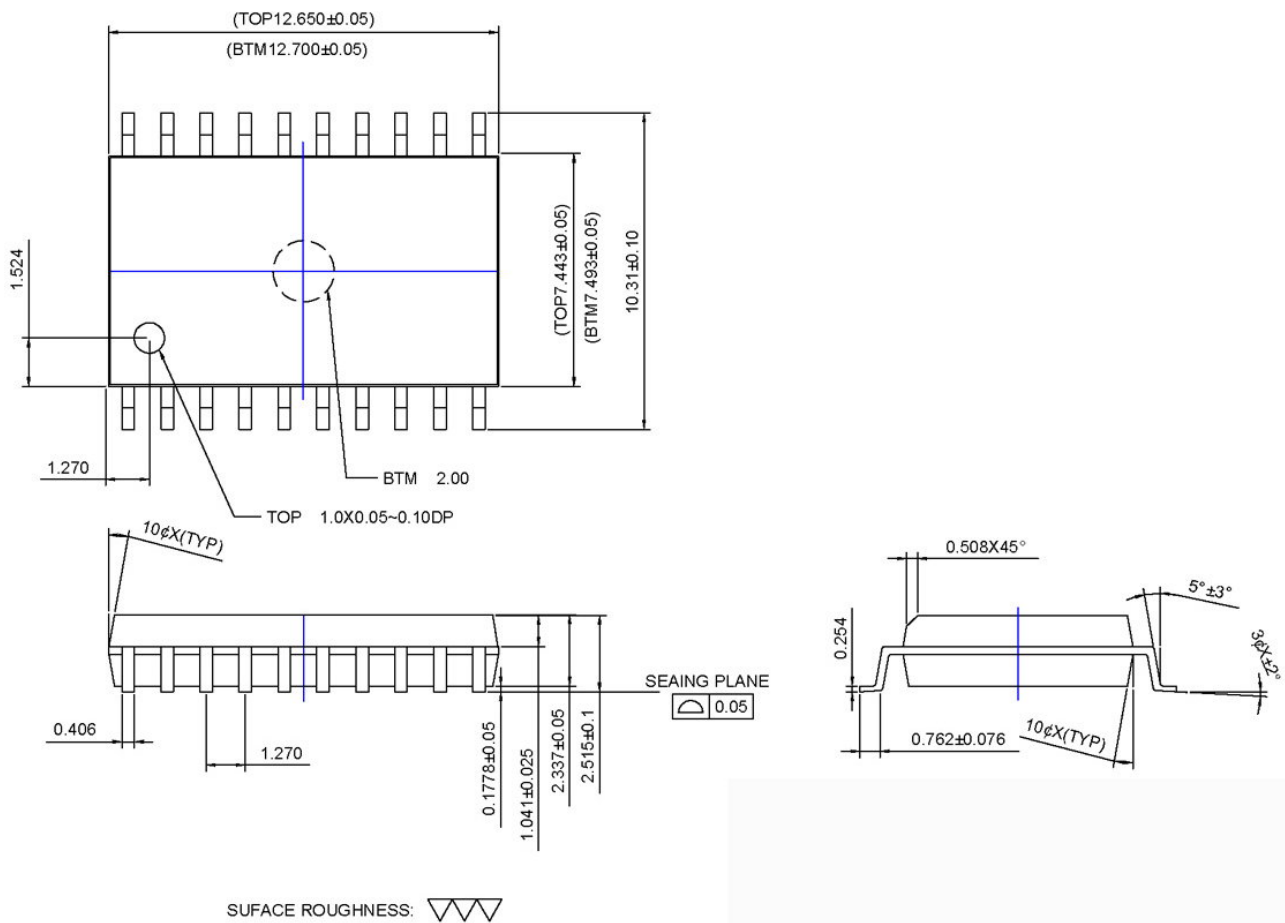


NOTE:
 1. CONTROLLING DIMENSION : INCH
 2. LEAD FRAME MATERIAL : A194
 3. PACKAGE DIMENSION EXCLUDE MOLD FLASH OR PROTRUSION.
 4. ALLOWABLE MOLD FLASH OR PROTRUSION SHALL NOT EXCEED 0.010"
 5. TOLERANCE : 0.010" UNLESS OTHERWISE SPECIFIED.
 6. AFTER SOLDER DIPPING LEAD THICKNESS WILL BE 0.020" MAX.

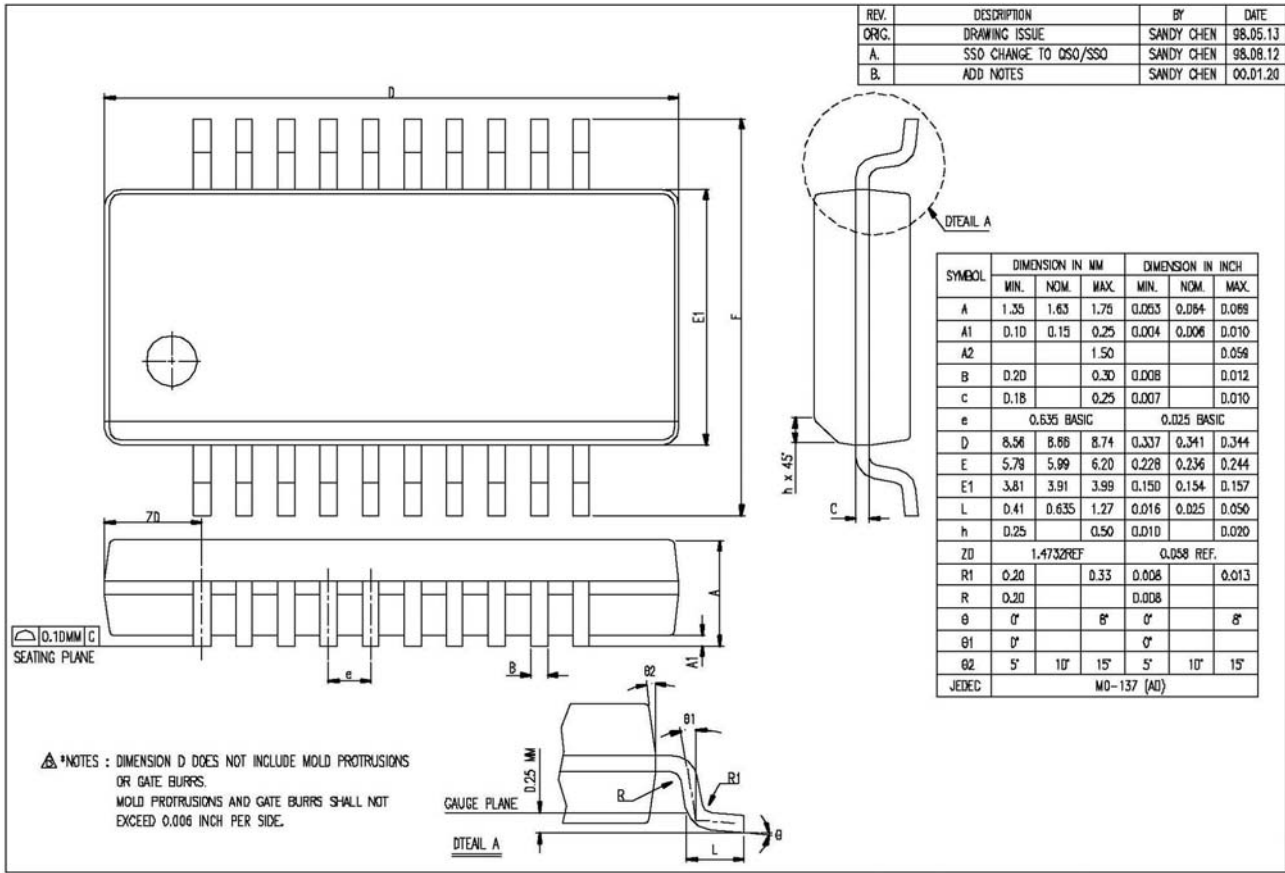


SYMBOLS	DIMENSIONS IN MILLIMETERS			DIMENSIONS IN INCHES		
	MIN	NOM	MAX	MIN	NOM	MAX
A	—	—	4.57	—	—	0.180
A1	0.38	—	—	0.015	—	—
A2	—	3.30	3.56	—	0.130	0.140
B	0.36	0.46	0.56	0.014	0.018	0.022
B1	1.27	1.52	1.78	0.050	0.060	0.070
C	0.20	0.25	0.33	0.008	0.010	0.013
D	26.32	26.416	26.52	1.036	1.040	1.044
D1	0.43	0.56	0.69	0.017	0.022	0.027
E	7.62	—	8.26	0.300	—	0.325
E1	6.40	6.50	6.65	0.252	0.256	0.262
e	—	2.54	—	—	0.100	—
L	3.18	—	—	0.125	—	—
eB	8.38	—	9.65	0.330	—	0.380

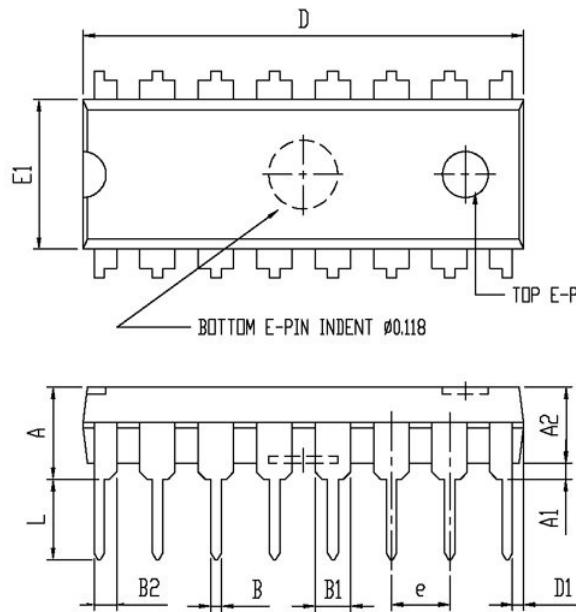
● 20-SOP Package Dimension



20-SSOP Package Dimension

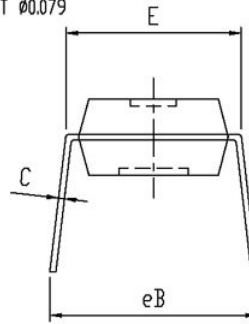


● 16-DIP Package Dimension



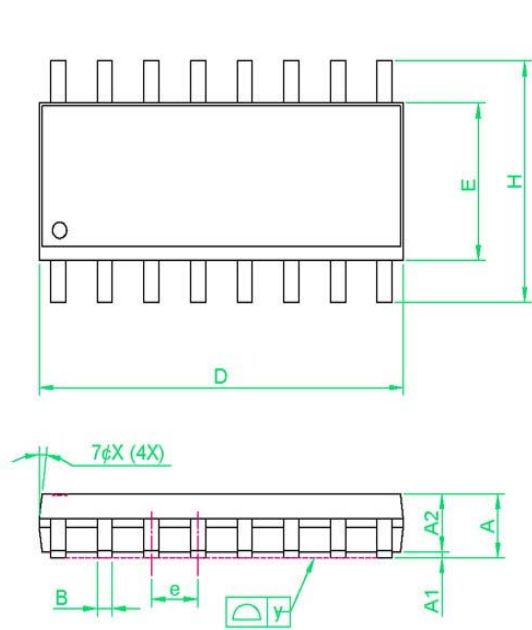
NOTE:

1. CONTROLLING DIMENSION : INCH
2. LEAD FRAME MATERIAL : C194
3. DIMENSION D AND E1 DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS. MOLD FLASH OR PROTRUSIONS SHALL NOT EXCEED 0.010"(0.25mm)
4. DIMENSION "B1" DO NOT INCLUDE DAMBAR PROTRUSION. DAMBAR PROTRUSIONS SHALL NOT EXCEED 0.010"(0.25mm). DISTANCE BETWEEN LEADS INCLUDING DAMBAR PROTRUSIONS TO BE 0.005"(0.13mm) INCH MINIMUM.
5. TOLERANCE : ± 0.010 "(0.25mm) UNLESS OTHERWISE SPECIFIED.
6. OTHERWISE DIMENSION FOLLOW ACCEPTABLE SPEC.
7. REFERENCE DOCUMENT: JEDEC SPEC MS-001-A2



SYMBOLS	DIMENSIONS IN MILLIMETERS			DIMENSIONS IN INCHES		
	MIN	NOM	MAX	MIN	NOM	MAX
A	—	—	4.57	—	—	0.180
A1	0.38	—	—	0.015	—	—
A2	3.25	3.30	3.45	0.128	0.130	0.136
B	0.36	0.46	0.56	0.014	0.018	0.022
B1	1.27	1.52	1.78	0.050	0.060	0.070
B2	0.81	0.99	1.17	0.032	0.039	0.046
C	0.20	0.25	0.33	0.008	0.010	0.013
D	18.90	19.15	19.30	0.744	0.754	0.760
D1	0.33	0.46	0.58	0.013	0.018	0.023
E	7.62	—	8.26	0.300	—	0.325
E1	6.35	6.50	6.65	0.250	0.256	0.262
e	—	2.54	—	—	0.100	—
L	3.18	—	—	0.125	—	—
eB	8.64	—	9.65	0.340	—	0.380

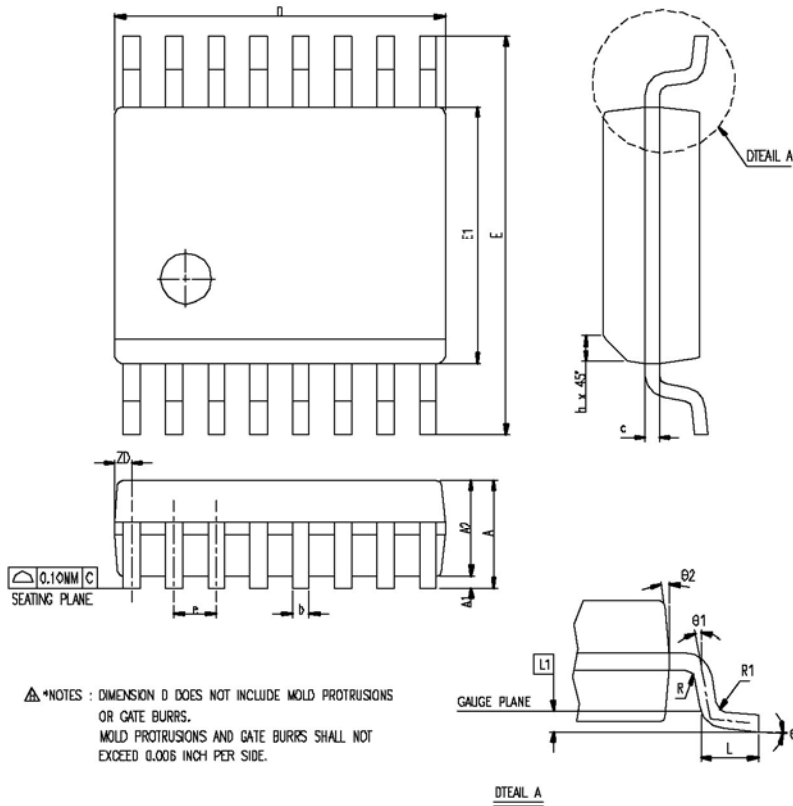
● 16-SOP Package Dimension



SYMBOLS	DIMENSIONS IN MILLIMETERS			DIMENSIONS IN INCHES		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.35	1.60	1.75	0.053	0.063	0.069
A1	0.10	—	0.25	0.004	—	0.010
A2	—	1.45	—	—	0.057	—
B	0.33	—	0.51	0.013	—	0.020
C	0.19	—	0.25	0.007	—	0.010
D	9.80	—	10.00	0.386	—	0.394
E	3.80	—	4.00	0.150	—	0.157
e	—	1.27	—	—	0.050	—
H	5.80	—	6.20	0.228	—	0.244
L	0.40	—	1.27	0.016	—	0.050
y	—	—	0.10	—	—	0.004
θ	0°	—	8°	0°	—	8°

- NOTE
- 1. PACKAGE BODY SIZES EXCLUDE MOLD FLASH AND GATE BURRS
 - 2. DIMENSION L IS MEASURED IN GAGE PLANE
 - 3. TOLERANCE 0.10 mm UNLESS OTHERWISE SPECIFIED
 - 4. CONTROLLING DIMENSION IS MILLIMETER. CONVERTED INCH DIMENSIONS ARE NOT NECESSARILY EXACT.
 - 5. FOLLOWED FROM JEDEC MS-012

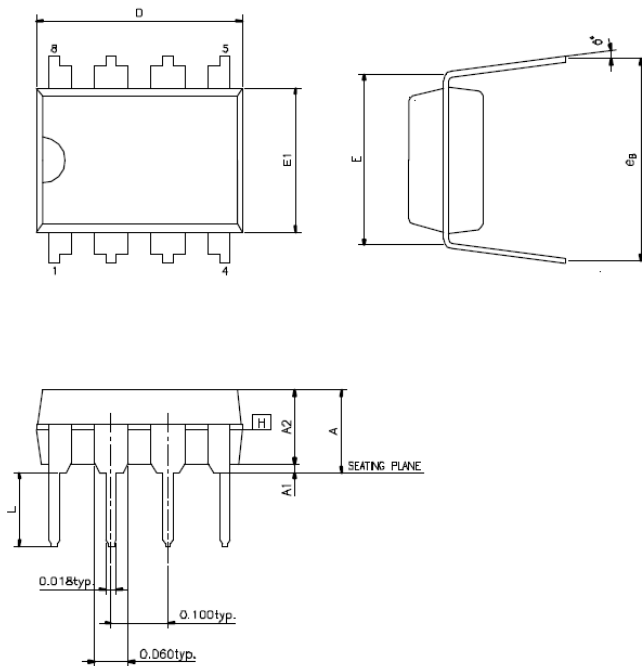
16-SSOP Package Dimension



REV.	DESCRIPTION	BY	DATE
ORIG.	DRAWING ISSUE	SANDY CHEN	98.01.16
A.	SSD CHANGE TO QSO/SSO	SANDY CHEN	98.08.12
B.	ADD NOTES	SANDY CHEN	00.01.20
C.	DIMENSION IN MM CHANGE TO IN INCH	SANDY CHEN	01.01.15

SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	1.35	1.63	1.75	0.053	0.064	0.069
A1	0.10	0.15	0.25	0.004	0.006	0.010
A2			1.50			0.059
b	0.20		0.30	0.008		0.012
c	0.18		0.25	0.007		0.010
e	0.635 BASIC			0.025 BASIC		
D	4.80	4.90	5.00	0.189	0.193	0.197
E	5.79	5.99	6.20	0.228	0.236	0.244
E1	3.81	3.91	3.99	0.150	0.154	0.157
L	0.41	0.635	1.27	0.016	0.025	0.050
h	0.25		0.50	0.010		0.020
L1	0.254 BASIC			0.010 BASIC		
ZD	0.229 REF			0.009 REF		
R1	0.20		0.33	0.008		0.013
R	0.20			0.008		
θ	0°		8°	0°		8°
$\theta 1$	0°			0°		
$\theta 2$	5°	10°	15°	5°	10°	15°
JEDEC	MO-137 (AB)					

8-DIP Package Dimension



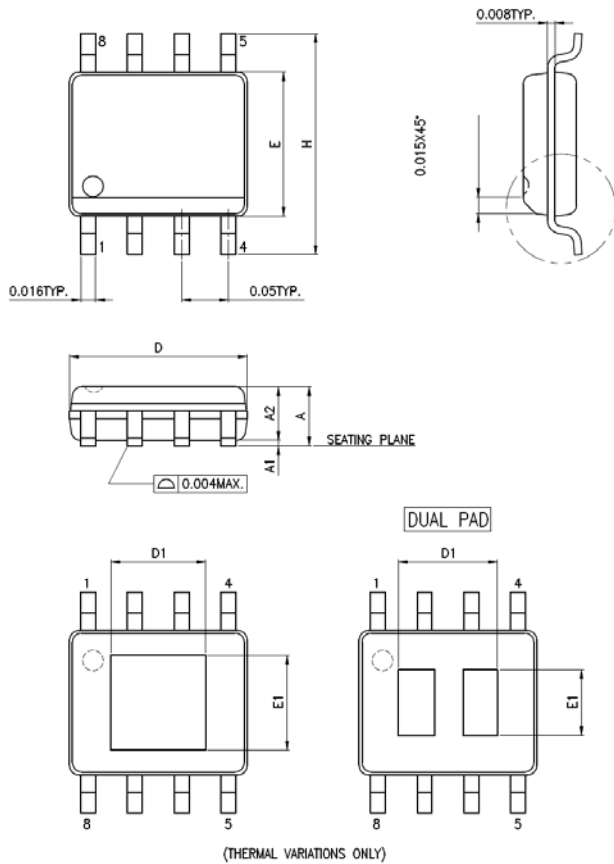
SYMBOLS	MIN.	NOR.	MAX.
A	—	—	0.210
A1	0.015	—	—
A2	0.125	0.130	0.135
D	0.355	0.365	0.400
E	0.300 BSC.		
E1	0.245	0.250	0.255
L	0.115	0.130	0.150
eB	0.335	0.355	0.375
ϕ	0	7	15

UNIT : INCH

NOTES:

1. JEDEC OUTLINE : MS-001 BA
2. "D", "E1" DIMENSIONS DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS. MOLD FLASH OR PROTRUSIONS SHALL NOT EXCEED .010 INCH.
3. eB IS MEASURED AT THE LEAD TIPS WITH THE LEADS UNCONSTRAINED.
4. POINTED OR ROUNDED LEAD TIPS ARE PREFERRED TO EASE INSERTION.
5. DISTANCE BETWEEN LEADS INCLUDING DAM BAR PROTRUSIONS TO BE .005 INCH MINIMUM.
6. DATUM PLANE $[H]$ COINCIDENT WITH THE BOTTOM OF LEAD, WHERE LEAD EXITS BODY.

● 8-SOP Package Dimension



SYMBOLS	STANDARD		THERMAL	
	MIN.	MAX.	MIN.	MAX.
A	0.053	0.069	0.053	0.067
A1	0.004	0.010	0.000	0.006
A2	0.053	0.059	0.049	0.059
D	0.189	0.196	0.189	0.196
E	0.150	0.157	0.150	0.157
H	0.228	0.244	0.228	0.244
L	0.016	0.050	0.016	0.050
θ°	0	8	0	8

UNIT : INCH

THERMALLY ENHANCED DIMENSIONS

PAD SIZE	E1		D1	
	MIN.	MAX.	MIN.	MAX.
90X90E	0.077	0.090	0.077	0.090
95X13E	0.081	0.095	0.111	0.130
96X65E(DUAL PAD)	0.070	0.096	0.114	0.140

UNIT : INCH

NOTES:

1. JEDEC OUTLINE : MS-012 AA(STANDARD)
MS-012 BA(THERMAL)
2. DIMENSIONS "D" DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS. MOLD FLASH, PROTRUSIONS AND GATE BURRS SHALL NOT EXCEED .15mm (.006in) PER SIDE.
3. DIMENSIONS "E" DOES NOT INCLUDE INTER-LEAD FLASH, OR PROTRUSIONS. INTER-LEAD FLASH AND PROTRUSIONS SHALL NOT EXCEED .25mm (.010in) PER SIDE.