



十速

TM57PE11A

规格书

Rev V1.1

tenx reserves the right to change or discontinue the manual and online documentation to this product herein to improve reliability, function or design without further notice. **tenx** does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. **tenx** products are not designed, intended, or authorized for use in life support appliances, devices, or systems. If Buyer purchases or uses tenx products for any such unintended or unauthorized application, Buyer shall indemnify and hold tenx and its officers, employees, subsidiaries, affiliates and distributors harmless against all claims, cost, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use even if such claim alleges that tenx was negligent regarding the design or manufacture of the part.

修改记录

版本	日期	描述
V1.0	Apr, 2013	新颁
V1.1	Jul, 2013	1. 新增支援的 EV board 于 ICE 2. 新增引脚摘要

CONTENTS

修改记录.....	2
基本功能.....	5
系统结构图.....	7
引脚分配图.....	8
引脚描述.....	8
引脚摘要.....	9
功能描述.....	10
1. CPU 核心.....	10
1.1 时钟配置和指令周期	10
1.2 寻址模式	11
1.3 程序计数器 (PC) 和堆叠	11
1.4 ALU 和工作 (W)寄存器.....	12
1.5 状态寄存器	12
1.6 中断	13
2. 芯片工作模式.....	14
2.1 复位	14
2.2 外部复位线路	15
2.3 系统配置寄存器(SYSCFG)	16
2.4 可重复编程	17
2.5 省电模式	17
3. 外围功能图.....	18
3.1 看门狗(WDT) / 唤醒(WKT)计时器.....	18
3.2 Timer0: 8 位计时器/有预分频(PSC)的计数器.....	20
3.3 系统时钟振荡器	21
4. I/O 端口	22
4.1 PA0-1.....	22
4.2 PA2-4.....	23
4.3 PA7	24
内存功能图.....	25
F-Plane	25
R-Plane	26

指令集.....	27
电气特性.....	39
1. 绝对最大值.....	39
2. DC 特性.....	40
3. 时钟计时.....	41
4. 复位定时特性.....	41
5. 特性曲线图.....	42
封装说明.....	44

基本功能

1. ROM: 1K x 14 位 OTP or 512 x 14 位 TTP™ (可两次编程的 ROM)
2. RAM: 48 x 8 位
3. 堆叠: 5 级
4. I/O 端口: 1 位可编程的 I/O 端口 (最多 6 个引脚)
5. 计时器/计数器: 一个带有 1~256 预分频的 8 位定时器/计数器
6. 看门狗定时器
 - 以内部 RC 震荡器四种可调的复位/中断时间进行计时
(103ms/52 ms/26 ms/13 ms, @5V; 131ms/66 ms/33 ms/16 ms, @3V)
 - 看门狗计时器可于停止模式下被禁止/使能
7. 复位
 - 上电复位
 - 看门狗复位
 - 低电压复位
 - 外部引脚复位
8. 系统时钟模式
 - 内部 RC: 4/8 MHz. 当 IRC 为 8 MHz, LVR 只能设置为 2.9V (不能使用 2V).
 - 外部 RC
9. 2 阶低电压复位: 2.0V/2.9V (可以禁止)

Freq	LVR		
	2.0V	2.9V	禁止
4 MHz	☑	☑	☑
8 MHz	☒	☑	☑

10. 操作电压: 低电压复位阶至 5.5V
 - fosc = 4 MHz, 2.2V ~ 5.5V
 - fosc = 8 MHz, 2.5V ~ 5.5V
11. I/O 端口
 - CMOS 输出
 - 伪开漏或开漏输出
 - 施密特触发输入搭配/无上拉电阻

12. 指令集: 36 个指令

13. 中断

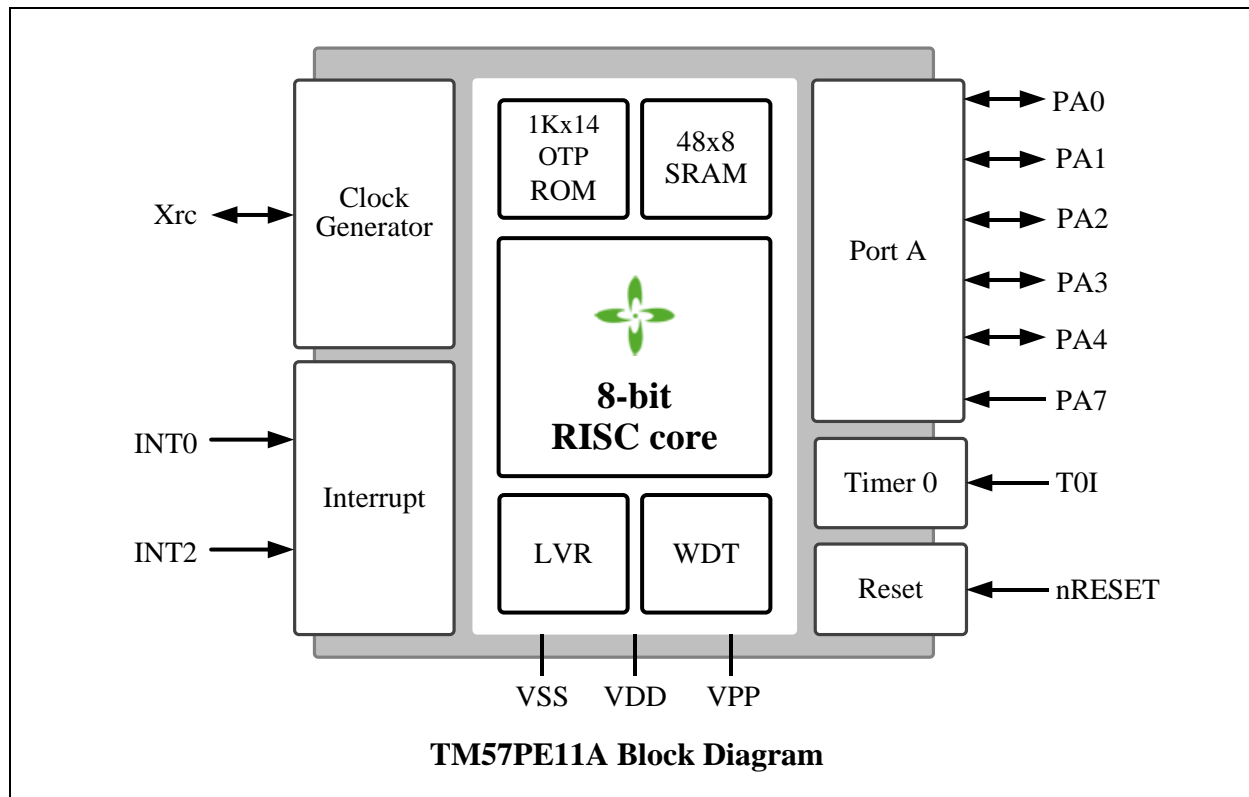
- 二个外部中断引脚:
 - 一个为下边沿唤醒触发
 - 一个为上或下边沿唤醒触发
- Timer0, 唤醒定时器中断

14. 支持省电模式

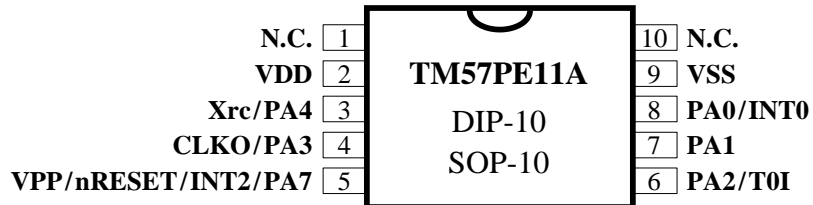
15. 支援的 EV Board 于 ICE

EV Board: EV2793

系统结构图



引脚分配图



引脚描述

名称	输入/输出	引脚描述
PA0-PA1	I/O	位编程 I/O 端口，可施密特触发输入或 CMOS 推挽输出或伪开漏式输出。上拉电阻由软件分配。
PA2-PA4	I/O	位编程 I/O 端口，可施密特触发输入或 CMOS 推挽输出或开漏式输出。上拉电阻由软件分配。
PA7	I	施密特触发输入
nRESET	I	外部低电平复位引脚
Xrc	-	外部 RC 震荡器连线用于系统时钟
CLKO	O	CPU 指令时钟输出用于外部/内部 RC 模式
VDD, VSS	P	电源电压输入引脚和接地引脚
VPP	I	PROM 编程高电压输入引脚
INT0, INT2	I	外部中断输入引脚
TOI	I	时钟输入至 Timer0

引脚摘要

引脚数量	引脚名称	类别	GPIO					复位后功能	替代功能				
			输入		输出				PWM	Touch Key	ADC	MISC	
			上拉	外部中断	O.D	P.O.D	P.P						
1	VDD	P											
2	Xrc/PA4	I/O	○		○		○	PA4					
3	CLKO/PA3	I/O	○		○		○	PA3					CLKO
4	VPP/nRESET/INT2/PA7	I	○	○				PA7					nRESET
5	NC	-											
6	NC	-											
7	PA2/T0I	I/O	○		○		○	PA2					T0I
8	PA1	I/O	○				○	PA1					
9	PA0/INT0	I/O	○	○			○	PA0					
10	VSS	P											

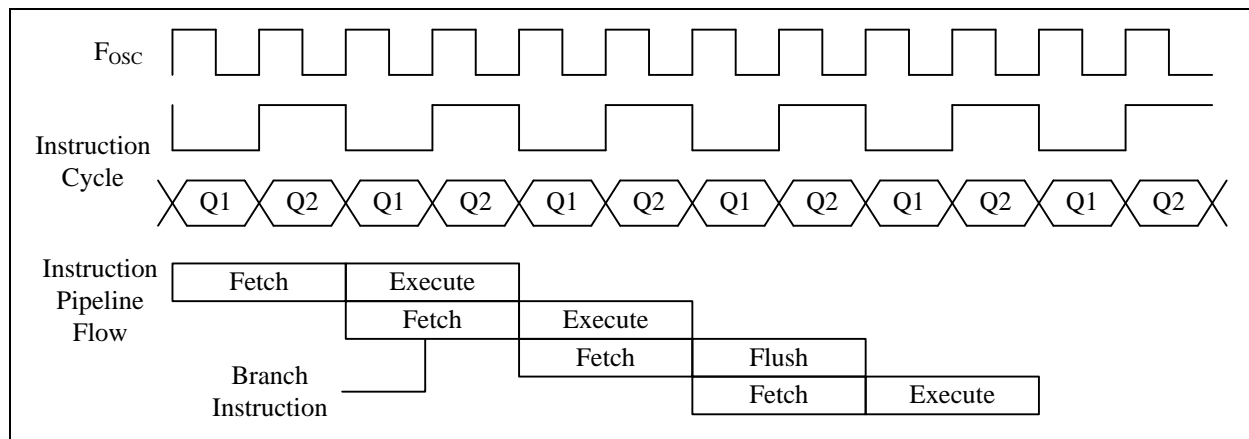
符号： P.P. = 推挽输出
 P.O.D. = 伪开漏式
 O.D. = 开漏式

功能描述

1. CPU 核心

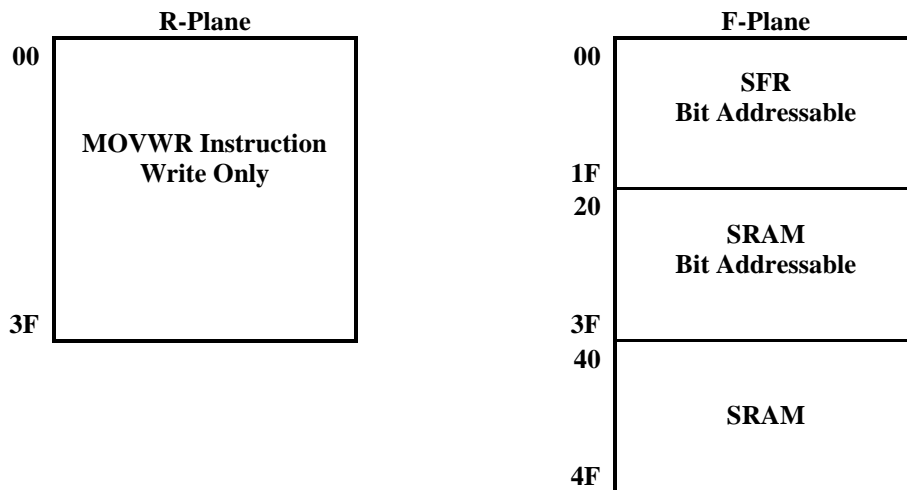
1.1 时钟配置和指令周期

系统时钟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 (PC) 在 Q1 更新且指令从程序寄存器获得并且在状态 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，因为从传输管道获得指令的同时新指令被获取并执行。



1.2 寻址模式

CPU 有两个数据储存器：R-Plane 和 F-Plane。R-Plane 里的寄存器只能写入。“MOVWR”指令通过直接寻址方式把 W 寄存器里的常数放在 R-Plane 寄存器中。F-Plane 的低位保留用于 SFR。SFR 寄存器以上的寄存器是一般目的数据储存器，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 寄存器不是物理寄存器。当对 INDF 进行存取时，它会根据 FSR 寄存器内的值作为地址（FSR 是指标）。F-Plane 的前半部是可寻址之位，后半部是不可寻址之位。



1.3 程序计数器 (PC) 和堆叠

程序计数器是一个 10 位宽，可寻址 $1K \times 14$ 的程序只读储存器。当一个程序指令被执行时，PC 里就存放着下一个将要被执行程序的地址。PC 值都会自动加 1，以下情况除外：复位向量 (000h) 和中断向量 (001h) 用来做 PC 初始化和中断。对于 CALL/GOTO 指令，PC 从指令中加载 10 位地址。对于 RET/RETI/RETLW 指令，PC 从堆叠顶取回其他地址值。对于更新 PC[7:0] 的其他指令，PC[9:8] 不变。堆叠有 10 位宽，深 5 级。CALL 指令和硬件中断将依顺序进入堆叠。RET/RETI/RETLW 指令将按顺序弹出堆叠。

1.4 ALU 和工作 (W)寄存器

ALU 有 8 位宽，可进行加，减，移位和逻辑运算的操作。在两个操作数的指令中，主操作数是一个 8 位不可寻址的 W 寄存器用于 ALU 运算。另一个操作数是一个文件寄存器或一个立即常数。在单个操作数的指令中，操作数即可是 W 寄存器，也可能是文件寄存器。决定于执行的指令，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。

备注：借位表示借位寄存器的反转

十进制借位表示十进制借位寄存器的反转

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。状态寄存器和其他寄存器一样可以是任何指令的目的单元。如果状态寄存器是影响 Z，DC，或 C 标志位的指令目的地，那么就不可以写进这三个标志位了。这些位的设定或清除是根据设备的逻辑运算。建议，只有 BCF，BSF，和 MOVWF 指令用来改变状态寄存器因为这些指令不会影响那些标志位。

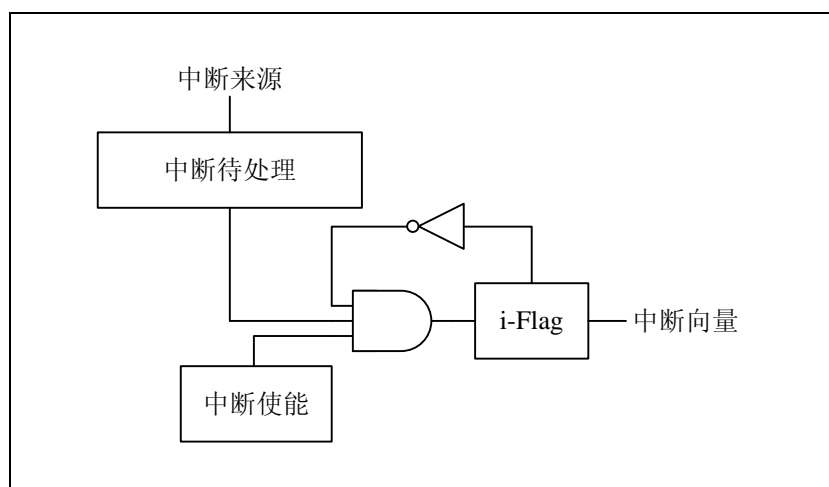
状态	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
复位值	-	-	-	0	0	0	0	0
R/W	-	-	-	R	R	R/W	R/W	R/W
位	描述							
7~5	未使用							
4	TO: 时间溢出 0: 上电复位后, LVR 复位或执行 CLRWDT/SLEEP 指令后 1: WDT 计时发生溢出							
3	PD: 省电模式标志 0: 上电复位后, LVR 复位或执行 CLRWDT 指令后 1: SLEEP 指令后							
2	Z: 零标志位 0: 逻辑操作的结果不是 0 1: 逻辑操作的结果是 0							
1	DC: 十进制进位标志或十进制借位标志							
	加法指令				减法指令			
	1: 低四位有进位 0: 无进位				1: 无借位 0: 低四位有借位			
0	C: 进位标志或借位标志							
	加法指令				减法指令			
	1: 高四位有进位 0: 无进位				1: 无借位 0: 高四位有借位			

1.6 中断

TM57PE11A 有一级，一个向量，四个中断源。每个中断源有它自己启用控制位。每个中断事件都可以触发它自己的中断标志位，不论它的中断启用控制位是 0 或 1。因为 TM57PE11A 只有一个向量，所以没有中断优先寄存器。中断优先权由 F/W 决定。

若相对的中断启用控制位已被设定（INTE），它将触发 CPU 服务此中断。CPU 接受目前执行的指令周期结尾的中断。同时，“CALL 001”指令被插进 CPU，并且 i-flag 被设定来防止中断嵌套。

i-flag 在执行“RETI”指令后即被清零。所以，待处理中断时，至少一个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 必须清除中断事件寄存器。



2. 芯片工作模式

2.1 复位

TM57PE11A 可用以下四种方式复位：

- 上电复位
- 低电压复位(LVR)
- 外部引脚复位(PA7)
- 看门狗复位(WDT)

上电复位后，所有系统和外围控制寄存器都将恢复他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。当低电压低于阈值水平时，低电压复位功能将能启动。复位电压阈值有二组可以选择。LVR 工作模式是通过 SYSCFG 寄存器定义。

低电复位阈值水平有二种选择，比较高的 LVR 电压适合应用在 VDD 电源电压大于 3.3V 以上的应用，另一个是适合应用在 VDD 电源电压小于 3.3V。参考下列 LVR 低电复位选项表；使用者必须考虑到操作频率的最低工作电压。

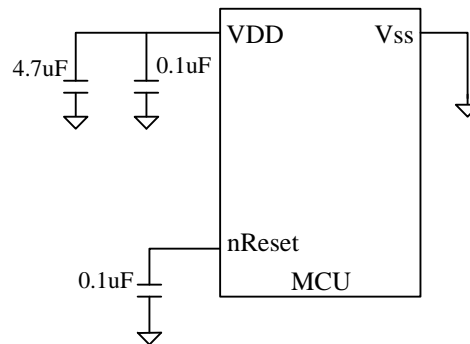
LVR 低电复位选项表：

LVR 阈值水平	考虑操作电压来选LVR
LVR2.9	$5.5V > V_{DD} > 3.3V$
LVR2.0	较宽的 V_{DD} 电源电压

外部引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复位以后所有设置也都恢复默认复位值。TO/PD 标志不会受到这些复位所影响。

2.2 外部复位线路

外部复位电路是低电平有效。系统运作时，复位引脚为高电平输入电压。复位引脚低电平时系统复位。外部复位可以在上电过程中进行系统复位，且好的外部复位电路可以保护系统避免在不寻常的功率状态下工作。



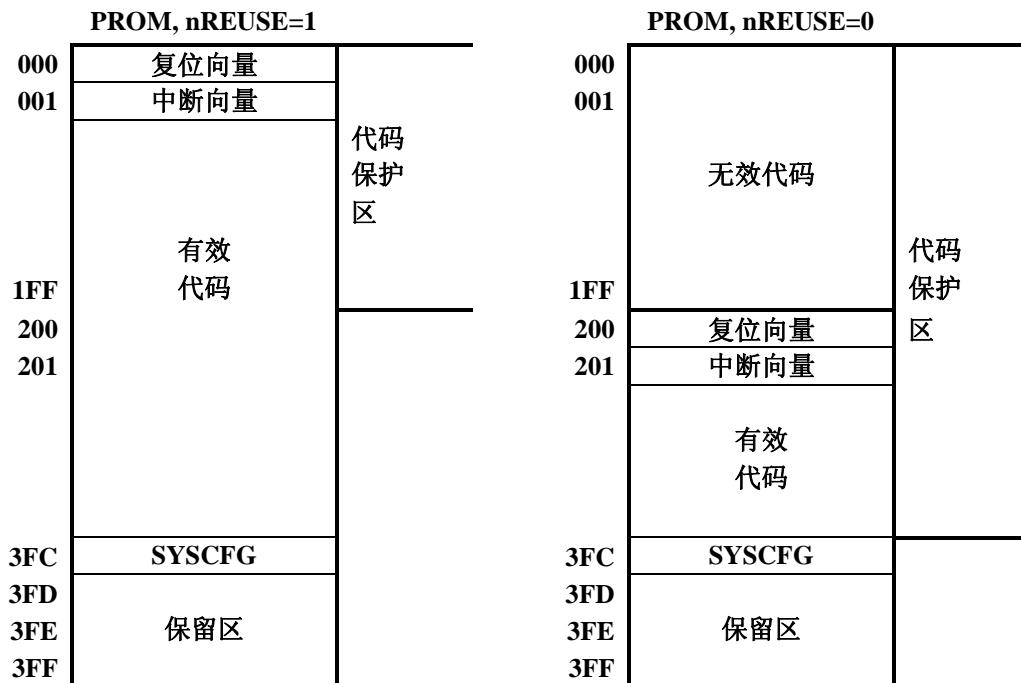
2.3 系统配置寄存器(SYSCFG)

系统配置寄存器 (SYSCFG)位于 ROM 的地址是 3FCh。其功能决定 MCU 初始条件选项。其只能由 PROM 烧录器进行写入。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 的预设值为 3FFFh。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	
预设值	11111111111111	
位	描述	
13	nPROTECT: 代码保护选择位	
	1	禁止
	0	使能
12	nREUSE: 程序重写控制位	
	1	禁止
	0	使能
11-10	LVR: 低电压复位模式	
	11	LVR 阈值为 2.0V, 总是使能
	10	LVR 阈值为 2.0V, 终止模式下禁止
	01	LVR 阈值为 2.9V, 总是使能
	00	LVR 禁止
9-8	CLKS: 时钟源选择位	
	1X	无效
	01	内部 RC (4/8 MHz)
	00	外部 RC
7	XRESETE: 外部引脚复位使能	
	1	使能外部引脚复位
	0	禁止外部引脚复位为输入 I/O 引脚
6	WDTE: WDT 复位使能	
	1	使能
	0	禁止
5	IRC: 1:IRCCLK=4 MHz 0: IRCCLK=8 MHz 当 IRC 为 8 MHz, LVR 只能设置为 2.9V 或禁止(不能使用 2V).	
4-0	IRCF: 内部 RC 频率调整控制	

2.4 可重复编程

TM57PE11A 程序存储器有 1K 字，比一些少于 512K 字的 F/W 程序来说要大很多。为了充分利用程序存储器，TM57PE11A 允许编程者分区编程，故称为可分区编程存储器。此特性称为两次可编程（TTP）ROM。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“重新写入（REUSE）”选项，然后编译器将目标代码放在相应的位置。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 200h。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，“REUSE”写入“0”，代码保护区就变成除保留区的整个程序存储区了。



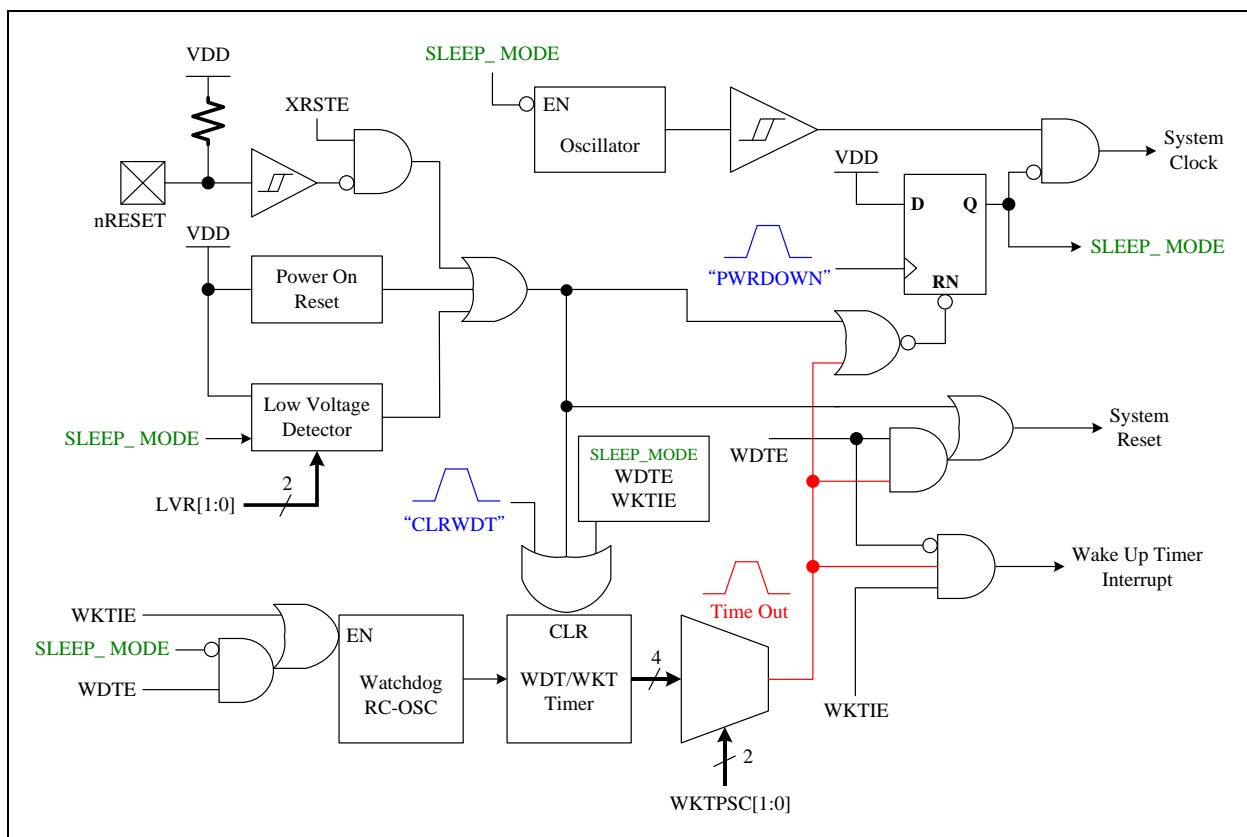
2.5 省电模式

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（外部引脚引起的中断和 WKT 中断）而结束。

3. 外围功能图

3.1 看门狗(WDT)/唤醒(WKT)计时器

WDT/WKT 是同一个内部 RC 定时器，其溢出周期可在 13 毫秒到 131 毫秒间选择。WDT/WKT 位由 CLRWDT 指令来清零。若 WDT 复位启用（即 WDTE=1），WDT 就产生芯片复位信号，否则，WKT 仅产生溢出中断。WDT/WKT 在正常工作模式和睡眠模式都可以独立工作。省电模式中，用户可利用"WKTIE"选择使能或禁止 WDT/WKT。若 WKTIE=0 在省电模式（WDTE 为 1 或 0），内部 RC 定时器停止工作来减少电源消耗。换言之，使用者让可设定 WKTIE=1 在省电模式让 WDT/WKT 持续有效。若 WDTE=1 且 WKTIE=0，WDT/WKT 计时器将被清除且停止工作来减少电源消耗。若 WDTE=1 且 WKTIE=1，WDT/WKT 将在省电/正常模式持续计数。请参考以下表格和图。

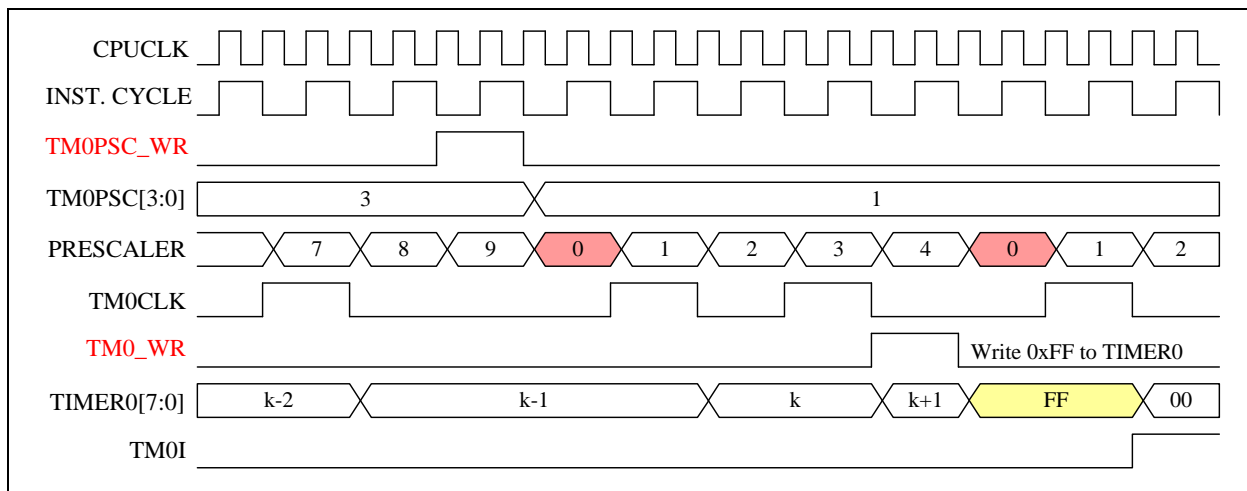
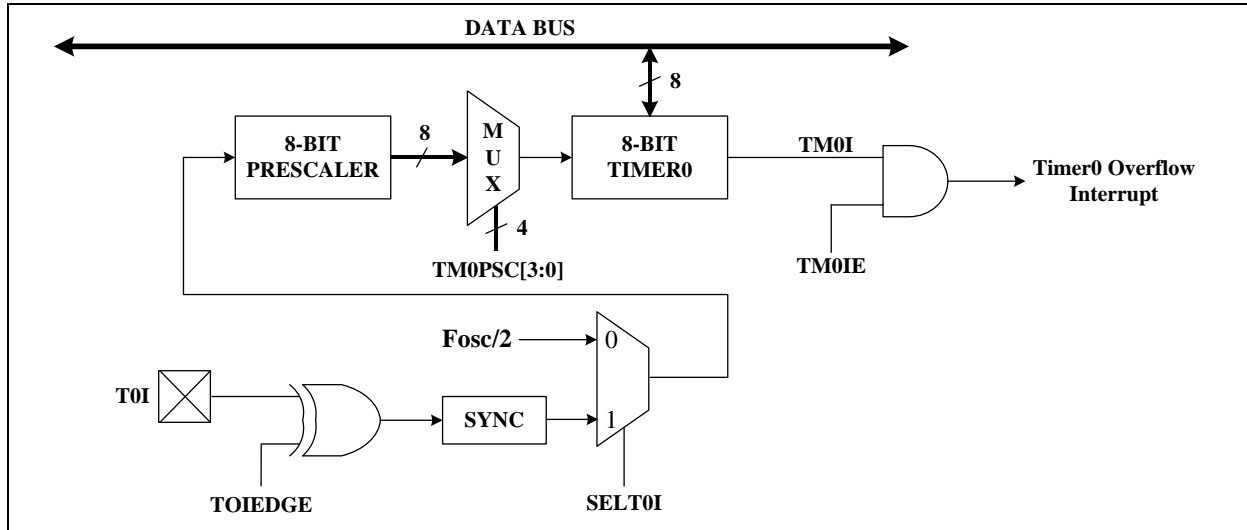


若用户程序需要 MCU 在省电模式下完全关闭电源，可以依照以下设定控制位。

模式	WDTE	WKTIE	看门狗RC震荡器
正常模式	0	0	停止
	0	1	运行
	1	0	
	1	1	
省电模式	0	0	停止
	0	1	运行
	1	0	停止
	1	1	运行

3.2 Timer0: 8 位计时器/有预分频(PSC)的计数器

Timer0 是位于 F-Plane 的 8 位寄存器。如其他 F-Plane 寄存器一样可以读和写。另外, Timer0 在预分频的时钟源下, 周期性的增加并可以自动溢出, 这即是指令周期或 TOI 输入。Timer0 的增加比例由 R-Plane 中的 Timer0 预分频 (TM0PSC) 决定。在定时器溢出后 Timer0 产生中断标志 (TM0I)。



Timer0 中断频率由指令周期: $(F_{osc} / 2) / \text{div} / 256$

备注: 变量 **div** 代表预分频因子由 TM0PSC [3:0]选择的值(1, 2, ~ 128, 256)

当 $F_{osc} = 4 \text{ MHz}$, $\text{div} = \text{TM0PSC} [3:0]$ 当选择 $4'b0000 = 1$

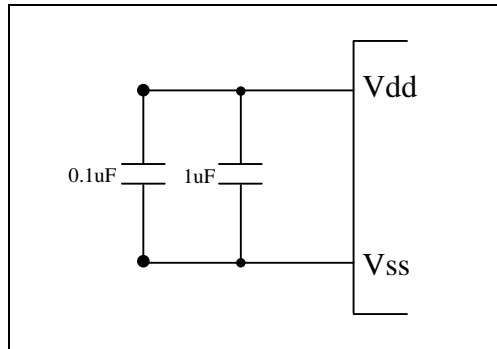
$(4\text{M}/2)/1/256 = 2 \text{ M}/256 \text{ Hz} = 7.8125 \text{ KHz}$

Timer0 中断频率由 TOI: $(\text{TOI}) / \text{div} / 256$

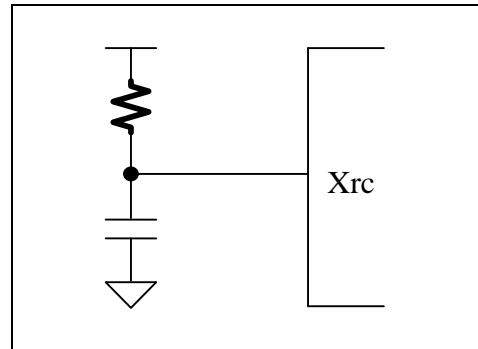
备注: **TOI 频率** $\leq F_{osc} / 4$

3.3 系统时钟振荡器

系统时钟源可以通过 SYSCFG 寄存器的 CLKS 设置 2 种不同的振荡模式。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 4/8 MHz 的振荡时钟。当内部 RC 为 8 MHz 时，LVR 只能设定为 2.9V（不能用 2V）。在此种模式下，PCB 布局可能会严重影响内部时钟的稳定性。电源噪声可降低内部时钟性能，因此，在紧靠 VDD/VSS 引脚的位置放 1 uF 和 0.1 uF 的电源旁路电容可提高内部时钟和整个系统的稳定性。



内部RC模式

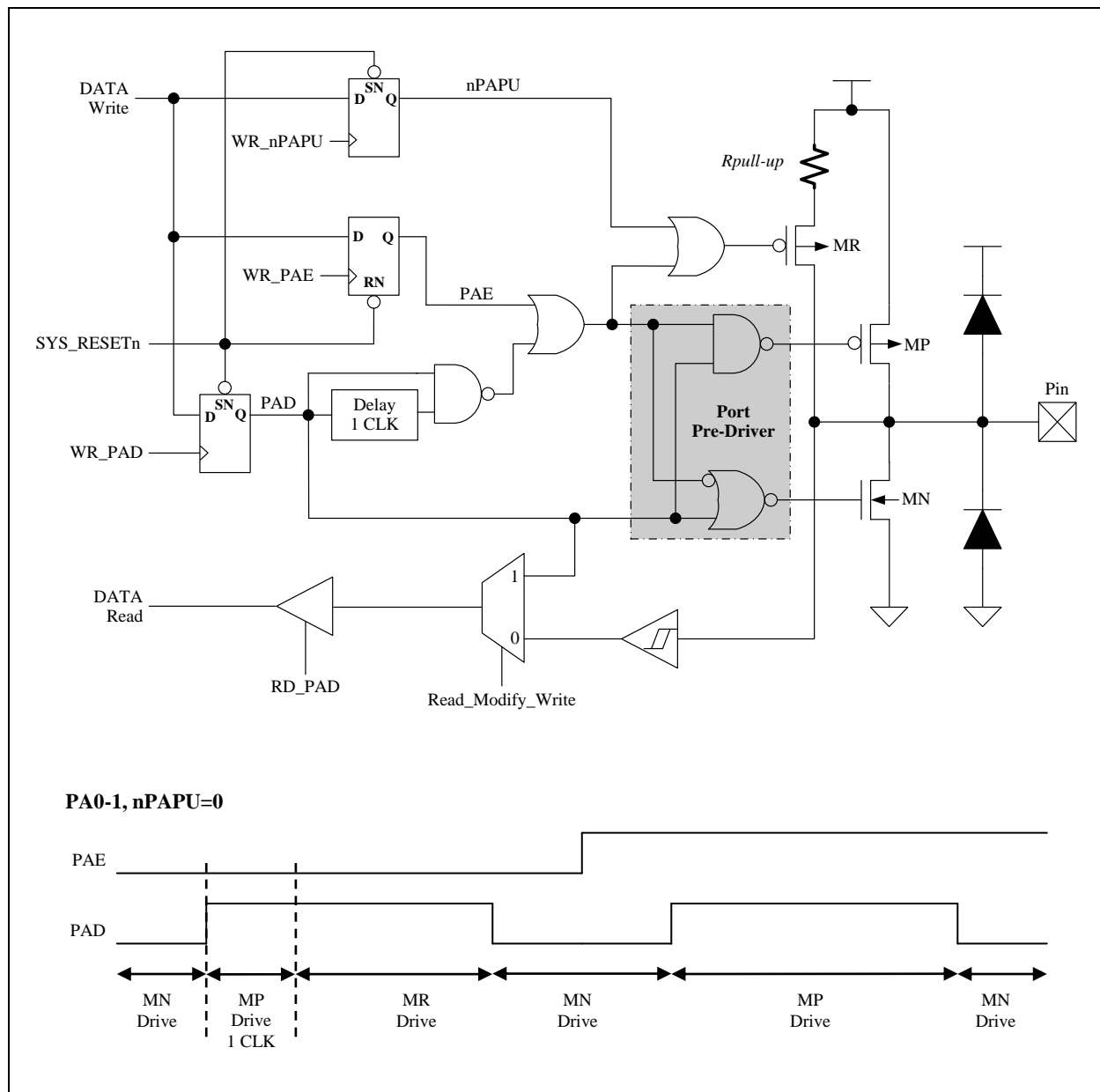


外部RC震荡器

4. I/O 端口

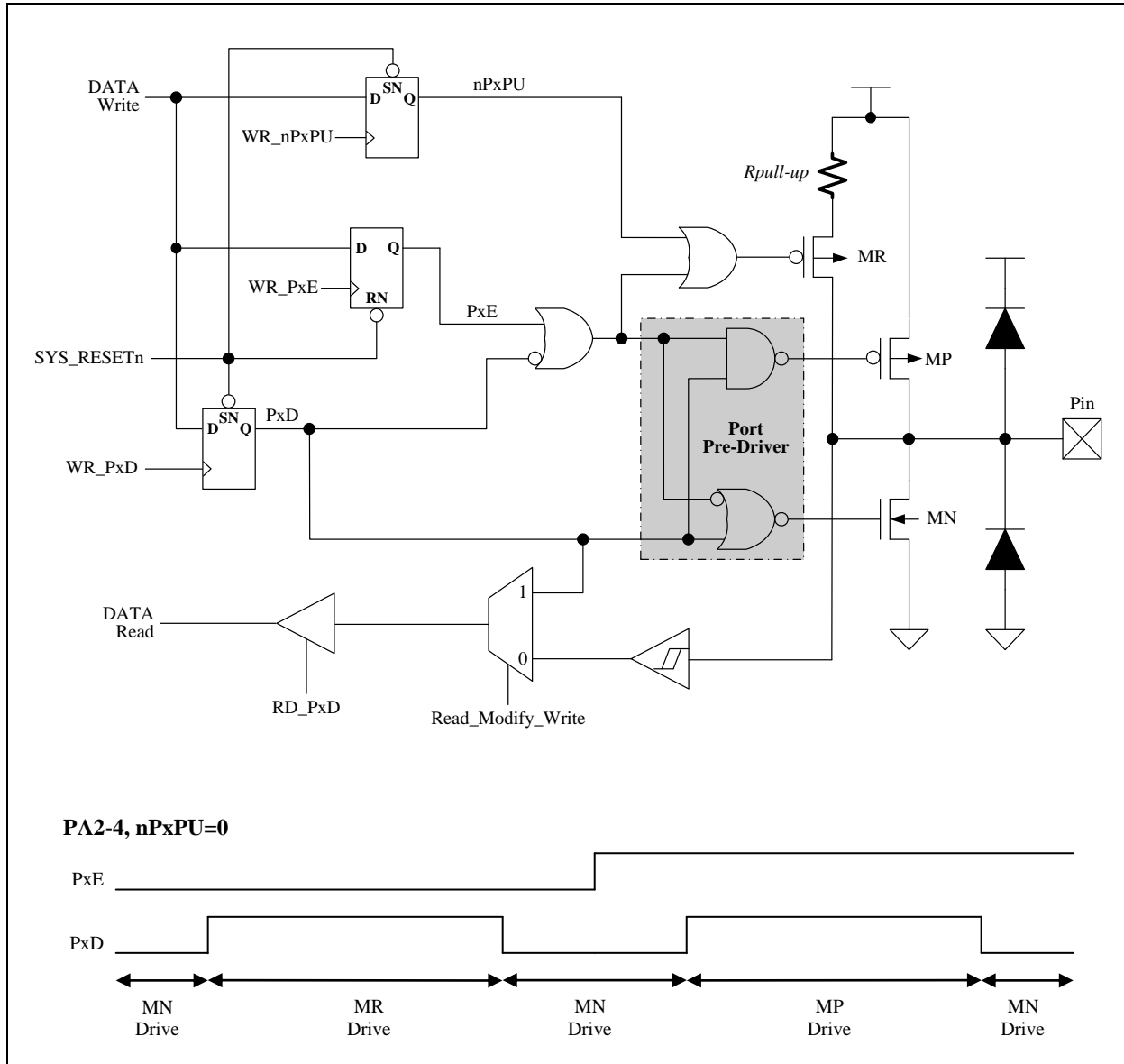
4.1 PA0-1

PA0-1 这两个引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。



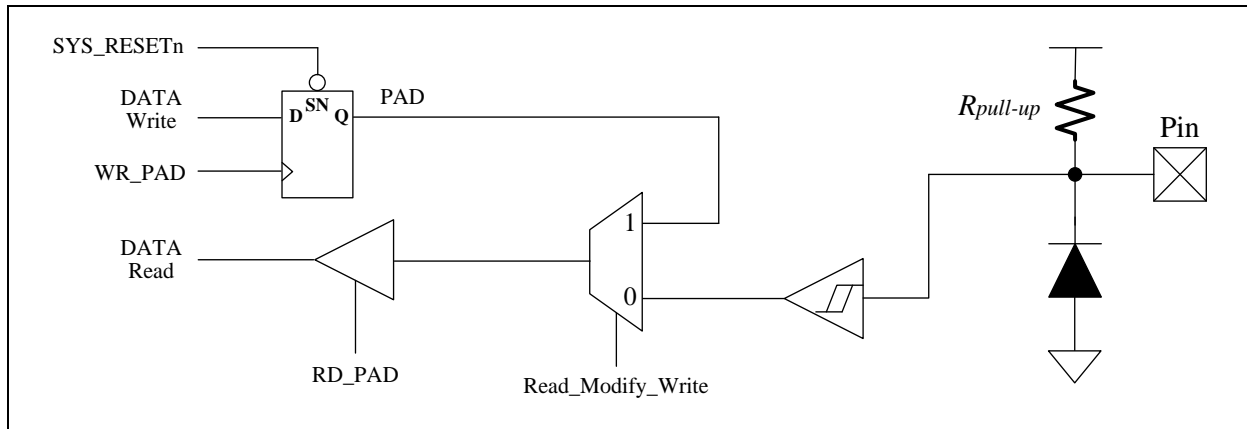
4.2 PA2-4

这些引脚特性与 PA0-1 基本相同，除了他们不支持伪开漏式模式，只能用于纯开漏式模式。



4.3 PA7

PA7 可用于施密特触发或纯开漏式输出。上拉电阻预设为连结到此引脚。



内存功能图

F-Plane

名称	地址	R/W	复位值	描述
INDF	00.7~0	R/W	-	非物理寄存器，INDF 寻址实际上指向 FSR 寄存器内的地址的寄存器
TIMER0	01.7~0	R/W	0	Timer0 内容
PC	02.7~0	R/W	0	Programming Counter [7~0]
TO	03.4	R	0	WDT 溢出标志
PD	03.3	R	0	省电模式标志
ZFLAG	03.2	R/W	0	零标志
DCFLAG	03.1	R/W	0	十进制进位标志
CFLAG	03.0	R/W	0	进位标志
FSR	04.6~0	R/W	-	档案选择寄存器，间接寻址模式指标
PAD7	05.7	R	-	PA7 引脚状态
PAD	05.4~0	R	-	Port A 引脚或“数据寄存器”状态
		W	1F	Port A 输出数据寄存器
TM0IE	08.4	R/W	0	Timer0 中断使能, 1=使能, 0=禁止
WKTIE	08.3	R/W	0	唤醒计时器中断使能, 1=使能, 0=禁止
XINT2E	08.2	R/W	0	INT2 引脚中断使能, 1=使能, 0=禁止
XINT0E	08.0	R/W	0	INT0 引脚中断使能, 1=使能, 0=禁止
TM0I	09.4	R	-	Timer0 中断事件延迟标志, 当 Timer0 溢出由 H/W 设定
		W	0	0: 清除此标志 1: 无变化
WKTI	09.3	R	-	WKT 中断事件延迟标志, 当 WKT 溢出由 H/W 设定
		W	0	0: 清除此标志 1: 无变化
XINT2	09.2	R	-	INT2 中断事件延迟标志, 当 INT2 引脚下降沿由 H/W 设定
		W	0	0: 清除此标志 1: 无变化
XINT0	09.0	R	-	INT0 上升/下降中断事件延迟标志, 当 INT0 引脚上升/下降沿由 H/W 设定
		W	0	0: 清除此标志 1: 无变化
SRAM	20~4F	R/W	-	内部 RAM

R-Plane

名称	地址	R/W	复位值	描述
T0IEDGE	02.5	W	0	0: Timer0/预分频计数沿用于 TOI (PA2)上升引脚 1: Timer0/预分频计数沿用于 TOI (PA2)下降引脚
SELTOI	02.4	W	0	0: Timer0/预分频时钟源为“指令周期” 1: Timer0/预分频时钟源为 TOI 引脚
TM0PSC	02.3~0	W	0	0000: Timer0 输入时钟除以 1 0001: Timer0 输入时钟除以 2 ~ 0111: Timer0 输入时钟除以 128 1000: Timer0 输入时钟除以 256
PWRDOWN	03	W	-	写入此寄存器将进入省电模式
CLRWDT	04	W	-	写入此寄存器将清除 WDT/WKT
PAE	05.4~2	W	0	0: 引脚为开漏式输出或施密特触发输入 1: 引脚为 CMOS 推挽输出
	05.1~0	W	0	0: 引脚为伪开漏式输出或施密特触发输入 1: 引脚为 CMOS 推挽输出
nPAPU	08.4~0	W	1F	0: 使能上拉电阻, 除了 a. 当输出数据寄存器 (PAD) 为 0 b. 当 CMOS 拉挽模式被选定 (PAE=1) c. 此运行在外部 RC 震荡 1: 上拉电阻被禁止
INT0EDGE	0b.4	W	0	0: INT0 下降沿触发中断事件 1: INT0 上升沿触发中断事件
CLK2PIN	0b.3	W	0	0: 无指令时钟输出至 PA3 引脚 1: 指令时钟输出至 PA3 引脚用于外部/内部 RC 模式
WKT PSC	0b.1~0	W	11	WDT/WKT 预分频选项选择 00: WDT/WKT 区间为 13 ms, @5V; 16 ms, @3V 01: WDT/WKT 区间为 26 ms, @5V; 33 ms, @3V 10: WDT/WKT 区间为 52 ms, @5V; 66 ms, @3V 11: WDT/WKT 区间为 103 ms, @5V; 131 ms, @3V

指令集

每个指令都是一个 14 位字被分开成一个 Op Code，他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导向，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，“b”代表位指示，他用来选择受操作影响的位，而“f”代表地址指示。对于文字操作，“k”代表文字或常数。

简记符号	描述
f	F-Plane寄存器
r	R-Plane 寄存器
b	位地址
k	文字，常数或标签
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	零标志
C	进位标志
DC	十进制进位标志
PC	程序计数器
TOS	堆叠项
GIE	全局中断使能标志 (i-Flag)
[]	可选项
()	内容
.	比特项
B	之前
A	之后
←	赋值方向

助记符		操作码	指令周期	影响标志	说明
字节数据寄存器操作指令					
<u>ADDWF</u>	f,d	00 0111 dfff ffff	1	C,DC,Z	Add W and "f"
<u>ANDWF</u>	f,d	00 0101 dfff ffff	1	Z	AND W with "f"
<u>CLRF</u>	f	00 0001 1fff ffff	1	Z	Clear "f"
<u>CLRWF</u>		00 0001 0100 0000	1	Z	Clear W
<u>COMF</u>	f,d	00 1001 dfff ffff	1	Z	Complement "f"
<u>DECF</u>	f,d	00 0011 dfff ffff	1	Z	Decrement "f"
<u>DECFSZ</u>	f,d	00 1011 dfff ffff	1 or 2	-	Decrement "f", skip if zero
<u>INCF</u>	f,d	00 1010 dfff ffff	1	Z	Increment "f"
<u>INCFSZ</u>	f,d	00 1111 dfff ffff	1 or 2	-	Increment "f", skip if zero
<u>IORWF</u>	f,d	00 0100 dfff ffff	1	Z	OR W with "f"
<u>MOVFW</u>	f	00 1000 0fff ffff	1	-	Move "f" to W
<u>MOVWF</u>	f	00 0000 1fff ffff	1	-	Move W to "f"
<u>MOVWR</u>	r	00 0000 00rr rrrr	1	-	Move W to "r"
<u>RLF</u>	f,d	00 1101 dfff ffff	1	C	Rotate left "f" through carry
<u>RRF</u>	f,d	00 1100 dfff ffff	1	C	Rotate right "f" through carry
<u>SUBWF</u>	f,d	00 0010 dfff ffff	1	C,DC,Z	Subtract W from "f"
<u>SWAPF</u>	f,d	00 1110 dfff ffff	1	-	Swap nibbles in "f"
<u>TESTZ</u>	f	00 1000 1fff ffff	1	Z	Test if "f" is zero
<u>XORWF</u>	f,d	00 0110 dfff ffff	1	Z	XOR W with "f"
位数据寄存器操作指令					
<u>BCF</u>	f,b	01 000b bbff ffff	1	-	Clear "b" bit of "f"
<u>BSF</u>	f,b	01 001b bbff ffff	1	-	Set "b" bit of "f"
<u>BTFSC</u>	f,b	01 010b bbff ffff	1 or 2	-	Test "b" bit of "f", skip if clear
<u>BTFSS</u>	f,b	01 011b bbff ffff	1 or 2	-	Test "b" bit of "f", skip if set
文字和控制指令					
<u>ADDLW</u>	k	01 1100 kkkk kkkk	1	C,DC,Z	Add Literal "k" and W
<u>ANDLW</u>	k	01 1011 kkkk kkkk	1	Z	AND Literal "k" with W
<u>CALL</u>	k	10 kkkk kkkk kkkk	2	-	Call subroutine "k"
<u>CLRWDI</u>		00 0000 0000 0100	1	TO,PD	Clear Watch Dog Timer
<u>GOTO</u>	k	11 kkkk kkkk kkkk	2	-	Jump to branch "k"
<u>IORLW</u>	k	01 1010 kkkk kkkk	1	Z	OR Literal "k" with W
<u>MOVLW</u>	k	01 1001 kkkk kkkk	1	-	Move Literal "k" to W
<u>NOP</u>		00 0000 0000 0000	1	-	No operation
<u>RET</u>		00 0000 0100 0000	2	-	Return from subroutine
<u>RETI</u>		00 0000 0110 0000	2	-	Return from interrupt
<u>RETLW</u>	k	01 1000 kkkk kkkk	2	-	Return with Literal in W
<u>SLEEP</u>		00 0000 0000 0011	1	TO,PD	Go into standby mode, Clock oscillation stops
<u>XORLW</u>	k	01 1111 kkkk kkkk	1	Z	XOR Literal "k" with W

ADDLW "k"和W相加

语法	ADDLW k	
操作数	k : 00h ~ FFh	
运算	$(W) \leftarrow (W) + k$	
影响的状态位	C, DC, Z	
OP-Code	01 1100 kkkk kkkk	
描述	W寄存器中的内容和8位'k'值相加，结果放在W寄存器中。	
周期	1	
范例	ADDLW 0x15	B : W = 0x10 A : W = 0x25

ADDWF W和"f"相与

语法	ADDWF f [,d]	
操作数	f : 00h ~ 7Fh, d : 0, 1	
运算	$(\text{destination}) \leftarrow (W) + (f)$	
影响的状态位	C, DC, Z	
OP-Code	00 0111 dfff ffff	
描述	W寄存器的内容和寄存器的内容和'f'相加。结果放在目标寄存器中。若'd'为0，结果放在W中，若'd'为1，结果放在'f'中。	
周期	1	
范例	ADDWF FSR, 0	B : W = 0x17, FSR = 0xC2 A : W = 0xD9, FSR = 0xC2

ANDLW "k"和 W 相与

语法	ANDLW k	
操作数	k : 00h ~ FFh	
运算	$(W) \leftarrow (W) \text{ AND } k$	
影响的状态位	Z	
OP-Code	01 1011 kkkk kkkk	
描述	W寄存器中的内容和8位'k'值相加。结果放在W寄存器中。	
周期	1	
范例	ANDLW 0x5F	B : W = 0xA3 A : W = 0x03

ANDWF W和"f"相与

语法	ANDWF f [,d]	
操作数	f : 00h ~ 7Fh, d : 0, 1	
运算	$(\text{destination}) \leftarrow (W) \text{ AND } (f)$	
影响的状态位	Z	
OP-Code	00 0101 dfff ffff	
描述	W寄存器中的内容和'f'寄存器相与。若'd'为0，结果放在W寄存器中。若'd'为1，结果放在'f'寄存器中。	
周期	1	
范例	ANDWF FSR, 1	B : W = 0x17, FSR = 0xC2 A : W = 0x17, FSR = 0x02

BCF "f"的"b"位清零

语法	BCF f [,b]	
操作数	f : 00h ~ 3Fh, b : 0 ~ 7	
运算	(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	
范例	BCF FLAG_REG, 7	B : FLAG_REG = 0xC7 A : FLAG_REG = 0x47

BSF "f"的"b"位设置 1

语法	BSF f [,b]	
操作数	f : 00h ~ 3Fh, b : 0 ~ 7	
运算	(f.b) ← 1	
影响的状态位	-	
OP-Code	01 001b bbff ffff	
描述	寄存器'f'的'b'位被设置 1	
周期	1	
范例	BSF FLAG_REG, 7	B : FLAG_REG = 0x0A A : FLAG_REG = 0x8A

BTFSC "f"的"b"位若为 0, 间跳

语法	BTFSC f [,b]	
操作数	f : 00h ~ 3Fh, b : 0 ~ 7	
运算	若(f.b) = 0, 忽略下一个指令	
影响的状态位	-	
OP-Code	01 010b bbff ffff	
描述	若寄存器'f'的'b'位为 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	
范例	LABEL1 BTFSC FLAG, 1 TRUE GOTO SUB1 FALSE ...	B : PC = LABEL1 A : if FLAG.1 = 0, PC = FALSE if FLAG.1 = 1, PC = TRUE

BTFSS "f"的"b"位若为 1, 间跳

语法	BTFSS f [,b]	
操作数	f : 00h ~ 3Fh, b : 0 ~ 7	
运算	若(f.b) = 1, 忽略下一个指令	
影响的状态位	-	
OP-Code	01 011b bbff ffff	
描述	若寄存器'f'的'b'位为 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	1 or 2	
范例	LABEL1 BTFSS FLAG, 1 TRUE GOTO SUB1 FALSE ...	B : PC = LABEL1 A : if FLAG.1 = 0, PC = TRUE if FLAG.1 = 1, PC = FALSE

CALL **调用 "k" 子程序**

语法	CALL k
操作数	k : 000h ~ FFFh
运算	Operation: TOS ← (PC) + 1, PC.11~0 ← k
影响的状态位	-
OP-Code	10 kkkk kkkk kkkk
描述	调用子程序。首先返回的地址，(PC+1) 被推进堆叠的最高位置。12 位的 k 值会立刻载入 PC <11:0> 的位置中。这条指令要两个指令周期。
周期	2
范例	LABEL1 CALL SUB1 B : PC = LABEL1 A : PC = SUB1, TOS = LABEL1 + 1

CLRF **Clear "f"**

语法	CLRF f
操作数	f : 00h ~ 7Fh
运算	(f) ← 00h, Z ← 1
影响的状态位	Z
OP-Code	00 0001 1fff ffff
描述	寄存器 f 被清零，Z 标志被设置为 1。
周期	1
范例	CLRF FLAG_REG B : FLAG_REG = 0x5A A : FLAG_REG = 0x00, Z = 1

CLRW **W 清零**

语法	CLRW
操作数	-
运算	(W) ← 00h, Z ← 1
影响的状态位	Z
OP-Code	00 0001 0100 0000
描述	寄存器 W 被清零，Z 标志被设置为 1。
周期	1
范例	CLRW B : W = 0x5A A : W = 0x00, Z = 1

CLRWDT **看门狗清零**

语法	CLRWDT
操作数	-
运算	WDT/WKT Timer ← 00h
影响的状态位	TO, PD
OP-Code	00 0000 0000 0100
描述	CLRWDT 指令清看门狗定时器
周期	1
范例	CLRWDT B : WDT counter = ? A : WDT counter = 0x00

COMF
"f"取反

语法	COMF f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运算	(destination) ← (\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。若"d"为 0, 结果放在 W 中。若"d"为 1, 结果放在'f'中。
周期	1
范例	COMF REG1, 0 B : REG1 = 0x13 A : REG1 = 0x13, W = 0xEC

DECF
"f"递减 1

语法	DECF f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运算	(destination) ← (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。若"d"为 0, 结果放在 W 中。若"d"为 1, 结果放在'f'中。
周期	1
范例	DECF CNT, 1 B : CNT = 0x01, Z = 0 A : CNT = 0x00, Z = 1

DECFSZ
"f"递减 1, 结果为 0 则跳

语法	DECFSZ f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运算	(destination) ← (f) - 1, 若结果为 0 时, 忽略下一个指令
影响的状态位	-
OP-Code	00 1011 dfff ffff
描述	寄存器'f'的值递减 1。若'd'为 0, 结果放在 W 寄存器。若'd'为 1, 结果放在'f'寄存器中。若结果为 1, 下一个指令会继续执行。若结果为 0, 一个 NOP 指令会被取代执行, 此时会需要两个指令周期的时间。
周期	1 or 2
范例	LABEL1 DECFSZ CNT, 1 GOTO LOOP CONTINUE B : PC = LABEL1 A : CNT = CNT - 1 if CNT = 0, PC = CONTINUE if CNT ≠ 0, PC = LABEL1 + 1

IORLW	W 和立即数逻辑或
语法	IORLW k
操作数	k : 00h ~ FFh
运算	(W) ← (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
范例	IORLW 0x35 B : W = 0x9A A : W = 0xBF, Z = 0
IORWF	W 和'f'逻辑或
语法	IORWF f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运算	(destination) ← (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。若'd'为 0, 结果放在 W 寄存器。若'd'为 1, 结果放在'f'寄存器。
周期	1
范例	IORWF RESULT, 0 B : RESULT = 0x13, W = 0x91 A : RESULT = 0x13, W = 0x93, Z = 0
MOVFW	'f'内容取出送到 W
语法	MOVFW f
操作数	f : 00h ~ 7Fh
运算	(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
范例	MOVFW FSR B : FSR = 0xC2, W = ? A : FSR = 0xC2, W = 0xC2
MOVLW	W 置立即数
语法	MOVLW k
操作数	k : 00h ~ FFh
运算	(W) ← k
影响的状态位	-
OP-Code	01 1001 kkkk kkkk
描述	8 位'k'载入 W 寄存器中。忽略的值汇集为 0。
周期	1
范例	MOVLW 0x5A B : W = ? A : W = 0x5A

MOVWF **把 W 内容移动到"f"**

语法	MOVWF f	
操作数	f : 00h ~ 7Fh	
运算	(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	
范例	MOVWF REG1	B : REG1 = 0xFF, W = 0x4F A : REG1 = 0x4F, W = 0x4F

MOVWR **把 W 内容移动到"r"**

语法	MOVWR r	
操作数	r : 00h ~ 3Fh	
运算	(r) ← (W)	
影响的状态位	-	
OP-Code	00 0000 00rr rrrr	
描述	W 寄存器值移到r'寄存器中	
周期	1	
范例	MOVWR REG1	B : REG1 = 0xFF, W = 0x4F A : REG1 = 0x4F, W = 0x4F

NOP **空操作**

语法	NOP	
操作数	-	
运算	空操作	
影响的状态位	-	
OP-Code	00 0000 0000 0000	
描述	空操作	
周期	1	
范例	NOP	-

RET **子程序返回**

语法	RET	
操作数	-	
运算	PC ← TOS	
影响的状态位	-	
OP-Code	00 0000 0100 0000	
描述	从子程序返回。堆叠顶端的值会被弹出放在 PC。这是两个周期的指令。	
周期	2	
范例	RET	A : PC = TOS

RRF 带进位循环右移"f"

语法	RRF f [,d]	
操作数	f : 00h ~ 7Fh, d : 0, 1	
运算		
影响的状态位	C	
OP-Code	00 1100 dfff ffff	
描述	C 位放在'f'值的 MSB 之前当做位 9，然后每个位项右边移动，C 位则移到位 0，操作结果存于目标寄存器中。	
周期	1	
范例	RRF REG1, 0	B : REG1 = 1110 0110, C = 0 A : REG1 = 1110 0110 W = 0111 0011, C = 0

SLEEP 进入睡眠模式，时钟震荡停止

语法	SLEEP
操作数	-
运算	-
影响的状态位	TO, PD
OP-Code	00 0000 0000 0011
描述	进入睡眠模式，震荡器停止
周期	1
范例	SLEEP -

SUBWF "f"减去 W

语法	SUBWF f [,d]	
操作数	f : 00h ~ 7Fh, d : 0, 1	
运算	(destination) ← (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。若"d"为 0，结果放在 W 中。若"d"为 1，结果放在'f'中。	
周期	1	
范例	SUBWF REG1, 1	B : REG1 = 0x03, W = 0x02, C = ?, Z = ? A : REG1 = 0x01, W = 0x02, C = 1, Z = 0
	SUBWF REG1, 1	B : REG1 = 0x02, W = 0x02, C = ?, Z = ? A : REG1 = 0x00, W = 0x02, C = 1, Z = 1
	SUBWF REG1, 1	B : REG1 = 0x01, W = 0x02, C = ?, Z = ? A : REG1 = 0xFF, W = 0x02, C = 0, Z = 0

SWAPF	"f"高低 4 位内容互换
语法	SWAPF f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运算	(destination,7~4) ← (f.3~0), (destination.3~0) ← (f.7~4)
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。若"d"为 0, 结果放在 W 中。若"d"为 1, 结果放在'f'中。
周期	1
范例	SWAPF REG, 0 B : REG1 = 0xA5 A : REG1 = 0xA5, W = 0x5A
TESTZ	检测"f"是否为零
语法	TESTZ f
操作数	f : 00h ~ 7Fh
运算	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
范例	TESTZ REG1 B : REG1 = 0, Z = ? A : REG1 = 0, Z = 1
XORLW	W 和立即数逻辑导或
语法	XORLW k
操作数	k : 00h ~ FFh
运算	(W) ← (W) 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
范例	XORLW 0xAF B : W = 0xB5 A : W = 0x1A
XORWF	W 和"f"逻辑导或
语法	XORWF f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运算	(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。若"d"为 0, 结果放在 W 中。若"d"为 1, 结果放在'f'中。
周期	1
范例	XORWF REG, 1 B : REG = 0xAF, W = 0xB5 A : REG = 0x1A, W = 0xB5

电气特性

1. 绝对最大值 ($T_A = 25^\circ\text{C}$)

参数	范围	单位
电源电压	$V_{SS} - 0.3$ to $V_{SS} + 6.5$	V
输入电压	$V_{SS} - 0.3$ to $V_{DD} + 0.3$	
输出电压	$V_{SS} - 0.3$ to $V_{DD} + 0.3$	
每个引脚的输出拉电流	-25	mA
所有引脚的输出拉电流	-80	
每个引脚的输出灌电流	+30	
所有引脚的输出灌电流	+150	
最大工作电压	5.5	V
工作温度	-40 to +85	°C
储存温度	-65 to +150	

2. DC 特性 ($T_A = 25^\circ\text{C}$, $V_{DD} = 2.0\text{V to } 5.5\text{V}$)

参数	代号	条件	最小	典型值	最大	单位	
输入高电平	V_{IH}	所有输入, 除了 PA7	$V_{DD} = 5\text{V}$	$0.7V_{DD}$	-	-	V
			$V_{DD} = 3\text{V}$	$0.7V_{DD}$	-	-	V
		PA7	$V_{DD} = 5\text{V}$	$0.8V_{DD}$	-	-	V
			$V_{DD} = 3\text{V}$	$0.8V_{DD}$	-	-	V
输入低电平	V_{IL}	所有输入, 除了 PA7	$V_{DD} = 5\text{V}$	-	-	$0.2V_D$	V
			$V_{DD} = 3\text{V}$	-	-	$0.2V_D$	V
		PA7	$V_{DD} = 5\text{V}$	-	-	$0.2V_D$	V
			$V_{DD} = 3\text{V}$	-	-	$0.2V_D$	V
输出高电平	V_{OH}	所有输出	$V_{DD} = 5\text{V}, I_{OH} = 8\text{ mA}$	4.4	-	-	V
			$V_{DD} = 3\text{V}, I_{OH} = 4\text{ mA}$	2.6	-	-	
输出低电平	V_{OL}	所有输出	$V_{DD} = 5\text{V}, I_{OL} = 20\text{ mA}$	-	-	0.5	V
			$V_{DD} = 3\text{V}, I_{OL} = 10\text{ mA}$	-	-	0.3	
输入漏电流 (高输入)	I_{ILH}	所有输入	$V_{IN} = V_{DD}$	-	-	1	μA
输入漏电流 (低输入)	I_{ILL}	所有输入	$V_{IN} = 0\text{V}$	-	-	-1	μA
电源电流	I_{DD}	运行 8 MHz, 无负载	$V_{DD} = 5.0\text{V}$	-	2.1	-	mA
		运行 4 MHz, 无负载	$V_{DD} = 3.0\text{V}$	-	0.6	-	
		终止模式, LVR 禁止	$V_{DD} = 5\text{V}$	-	0.1	-	μA
			$V_{DD} = 3\text{V}$	-	0.1	-	
终止模式, LVR 使能	$V_{DD} = 5\text{V}$	-	0.9	-	μA		
	$V_{DD} = 3\text{V}$	-	0.6	-			
系统时钟频率	F_{OSC}	$V_{DD} > LVR_{th}$	$V_{DD} = 2.9\text{V}$	-	-	8	MHz
			$V_{DD} = 2.0\text{V}$	-	-	4	
LVR 参考电压			V_{LVR}	-	2.0	-	V
				-	2.9	-	V
LVR 滞后电压			V_{HYST}	-	± 0.1	-	V
低电压侦测时间			t_{LVR}	100	-	-	μs
上拉电阻	R_P	$V_{IN} = 0\text{ V}$ Port A	$V_{DD} = 5\text{V}$	-	65	-	$\text{K}\Omega$
			$V_{DD} = 3\text{V}$	-	130	-	
		$V_{IN} = 0\text{ V}$ PA7	$V_{DD} = 5\text{V}$	-	20	-	$\text{K}\Omega$
			$V_{DD} = 3\text{V}$	-	20	-	

3. 时钟计时 ($T_A = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$)

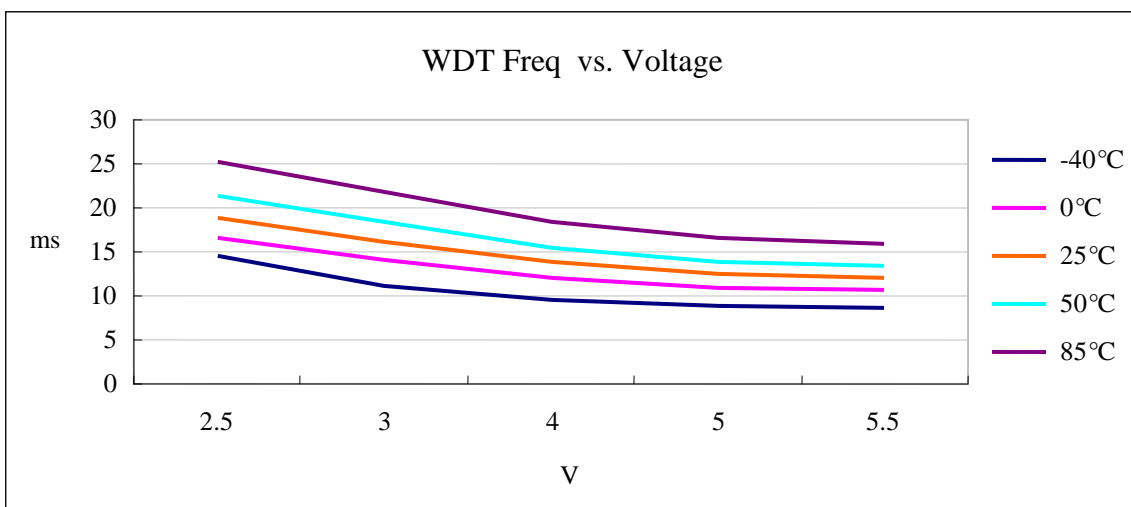
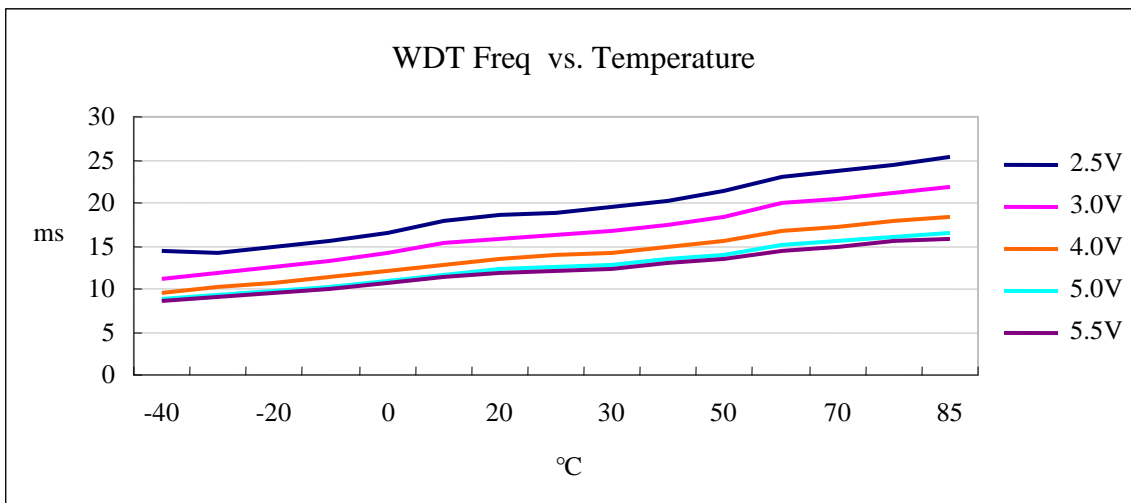
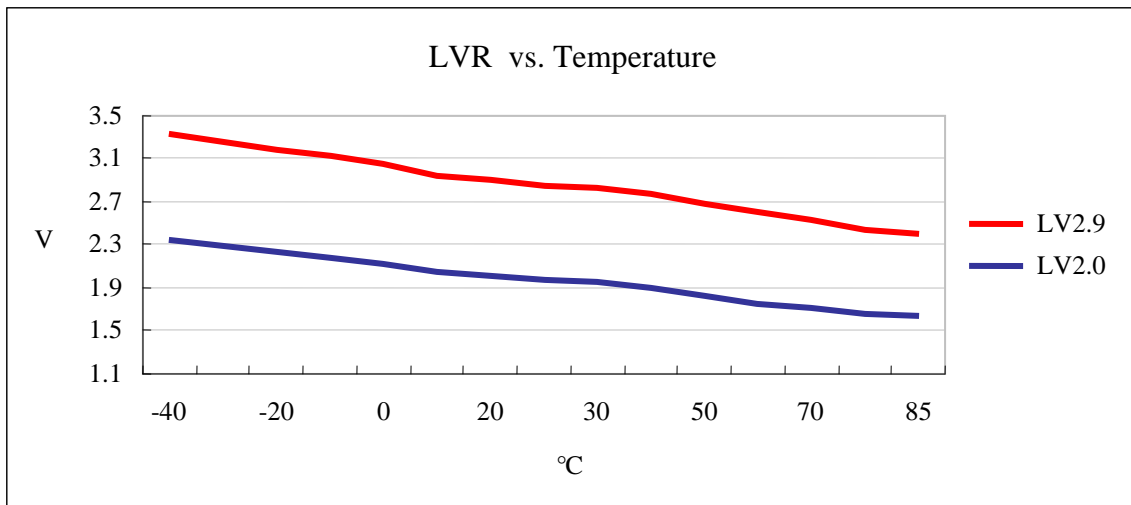
参数	条件		最小	典型值	最大	单位
外部 RC 频率	$V_{DD} = 3\text{V}$	R = 4.7K C = 20 pF	–	2.76	–	MHz
		R = 10K C = 100 pF	–	0.78	–	
		R = 100K C = 300 pF	–	0.05	–	
	$V_{DD} = 5\text{V}$	R = 4.7K C = 20 pF	–	3.55	–	
		R = 10K C = 100 pF	–	0.68	–	
		R = 100K C = 300 pF	–	0.03	–	
内部 RC 频率	$25^{\circ}\text{C}, V_{DD} = 2.5 \sim 5.5\text{V}$		3.9	4	4.1	
	$-40^{\circ}\text{C} \sim 85^{\circ}\text{C}, V_{DD} = 2.5 \sim 5.5\text{V}$		3.75	4	4.25	

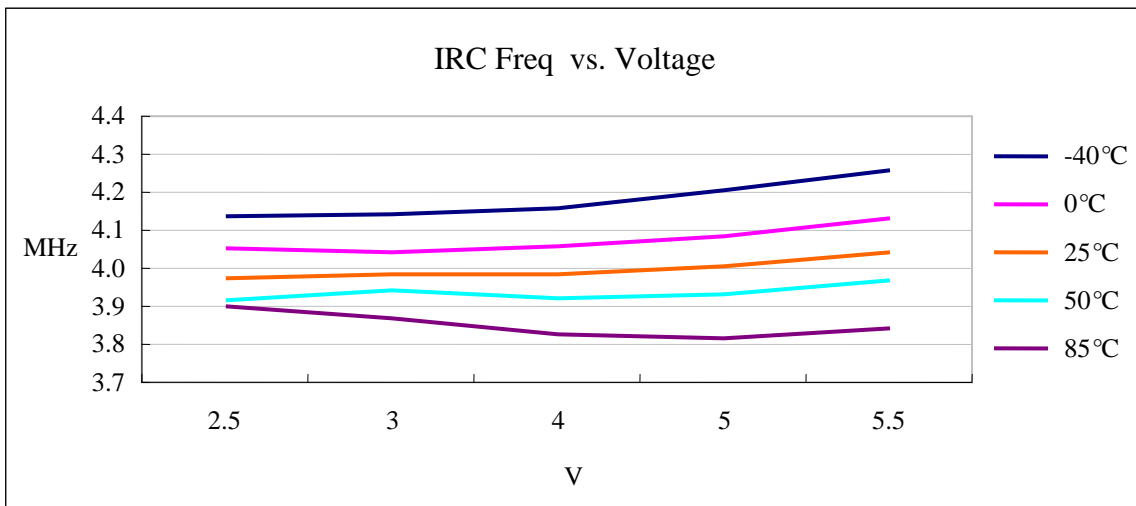
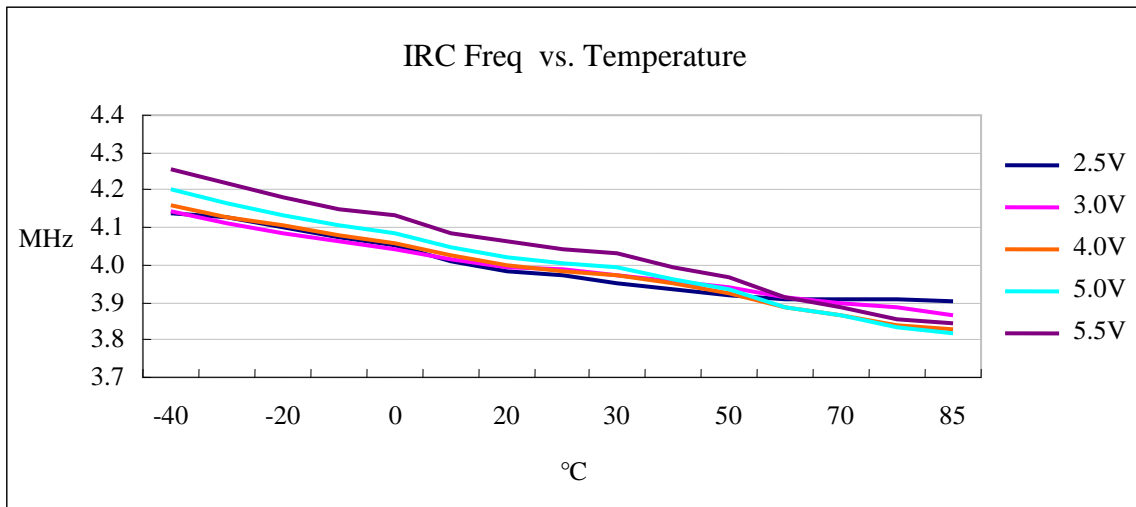
备注: 内部 RC 频率在晶元时已经调较完成。封装或 COB 后, 其频率变化在 10~20%。

4. 复位定时特性 ($T_A = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$, $V_{DD} = 2.0\text{V}$ to 5.5V)

参数	条件	最小	典型值	最大	单位
复位输入低脉宽时间	输入 $V_{DD} = 5\text{V} \pm 10\%$	3	–	–	μs
WDT 时间	$V_{DD} = 5\text{V}, \text{WKTPSC} = 11$	–	103	–	ms
	$V_{DD} = 3\text{V}, \text{WKTPSC} = 11$	–	131	–	
CPU 启动时间	$V_{DD} = 5\text{V}$	–	3.5	–	ms

5. 特性曲线图





封装说明

订购须知:

订购代码	封装
TM57PE11A-OTP	Wafer / Dice 空片
TM57PE11A-COD	Wafer / Dice 代烧code