

# TM8101-X2-MCU 规格书



## 目录

<b>1 概述</b> .....	<b>4</b>
<b>2 框架图</b> .....	<b>5</b>
<b>3 引脚</b> .....	<b>6</b>
<b>4 电气特性</b> .....	<b>8</b>
<b>5 存储器结构</b> .....	<b>10</b>
5.1 程序存储器.....	10
5.2 数据存储器.....	10
5.2.1 数据存储器结构.....	10
5.2.2 特殊功能寄存器描述.....	11
5.3 EEPROM 数据存储器.....	13
<b>6 程序计数器</b> .....	<b>14</b>
6.1 概述.....	14
6.2 程序计数器低字节寄存器.....	14
6.3 堆栈.....	14
<b>7 算术逻辑单元</b> .....	<b>15</b>
<b>8 看门狗定时器</b> .....	<b>16</b>
8.1 概述.....	16
8.2 相关特殊功能寄存器.....	16
<b>9 系统时钟与工作模式</b> .....	<b>17</b>
9.1 振荡器.....	17
9.2 相关特殊功能寄存器.....	18
9.3 系统工作模式.....	18
<b>10 复位与初始化</b> .....	<b>20</b>
10.1 复位.....	20
10.2 初始化.....	21
<b>11 输入/输出端口</b> .....	<b>22</b>
11.1 概述.....	22
11.2 相关特殊功能寄存器.....	22
<b>12 定时计数器 TIMER0</b> .....	<b>24</b>
12.1 概述.....	24
12.2 工作原理.....	24
12.3 相关特殊功能寄存器.....	24
<b>13 定时计数器 TIMER1</b> .....	<b>26</b>
13.1 概述.....	26
13.2 相关特殊功能寄存器.....	26
<b>14 中断</b> .....	<b>27</b>
14.1 概述.....	27
14.2 相关特殊功能寄存器.....	27
14.3 工作原理.....	28
14.4 中断类型介绍.....	29
<b>15 脉宽调制模块</b> .....	<b>30</b>
15.1 概述.....	30
15.2 相关特殊功能寄存器.....	30

<b>16 SRA 8 位 ADC.....</b>	<b>32</b>
16.1 概述.....	32
16.2 相关特殊功能寄存器.....	32
<b>17 SIGMA-DELTA 24 位 ADC.....</b>	<b>33</b>
17.1 概述.....	33
17.2 相关特殊功能寄存器.....	33
<b>18 温度传感器.....</b>	<b>35</b>
18.1 概述.....	35
18.2 相关特殊功能寄存器.....	35
<b>19 18X4 的 LCD 驱动器.....</b>	<b>36</b>
19.1 概述.....	36
19.2 LCD 结构图.....	36
19.3 结构说明.....	37
19.4 命令格式.....	38
19.5 时序图.....	38
19.6 命令概述.....	39
<b>20 指令集(总计 61 条).....</b>	<b>40</b>
<b>21 封装(SSOP48).....</b>	<b>50</b>

# 1 概述

该芯片为 8 位 RISC MCU

主要特征:

工作电压: 2.5V ~ 3.6V

内部集成高、低速度振荡器

    低速 -- 32KHz

    高速 -- 4.9MHz

多种工作模式: 正常模式、休眠模式、空闲模式

提供休眠模式、空闲模式 0、空闲模式 1 和唤醒功能, 以降低功耗

所有指令都可在一个或两个指令周期内完成

OTP: 8K \* 16bit 空间

SRAM: 256 byte

EEPROM: 内建 EEPROM, 芯片内部 EEPROM 与 PORT 端口封装在一起, EEPROM 大小为 256 byte

主要外设:

-- 1 通道 8 位逐次逼近 ADC

-- 2 通道 24 位 SIGMA-DELTA ADC

-- 温度传感器

-- 18x4 的 LCD 驱动器

其它

一个 8 位可编程定时/计数器

一个时基功能, 用于产生固定时间的中断信号

5 个中断源

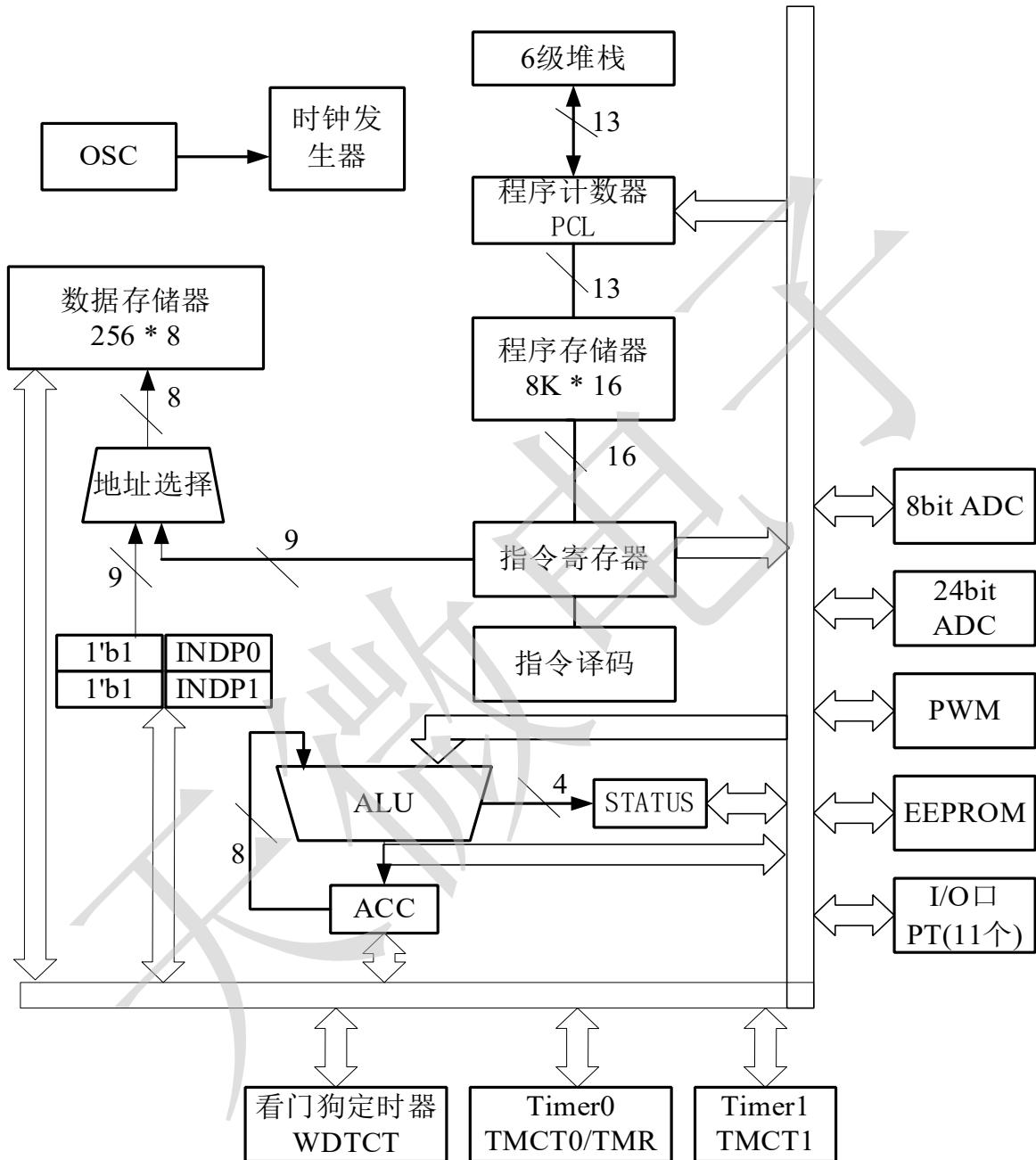
两路 8 位脉宽调制 (PWM) 输出

11 个双向 I/O 口

与 I/O 口复用外部中断输入

封装形式: SSOP48

## 2 框架图



### 3 引脚

ICPSCL/PA1	1	48	PA3
VPP	2	47	PA0/ICPSDA
VDD	3	46	PB1
VIN1P	4	45	PB0
VIN1N	5	44	PB3
AGND	6	43	PB4
ADC/PB2	7	42	SEG14
PWM1/PA5	8	41	SEG15
PWM0/PA4	9	40	SEG16
DGND	10	39	SEG17
PA7	11	38	SEG18
PA6	12	37	SEG19
INT/PA2	13	36	SEG20
$\overline{CS}$	14	35	SEG21
$\overline{WR}$	15	34	SEG22
DATA	16	33	SEG23
VSS	17	32	SEG24
VLCD	18	31	SEG25
VDD	19	30	SEG26
BZ	20	29	SEG27
COM0	21	28	SEG28
COM1	22	27	SEG29
COM2	23	26	SEG30
COM3	24	25	SEG31

**引脚功能定义:**

引脚名称	引脚类型	引脚序号	引脚描述
PA0/ICPSDA	I/O	47	IO; 在线烧录数据线
PA1/ICPSCL	I/O	1	IO; 在线烧录时钟线
PA2/INT	I/O	13	IO; 外部中断输入
PA3	I/O	48	IO
PA4/PWM0	I/O	9	IO; PWM0 输出
PA5/PWM1	I/O	8	IO; PWM1 输出
PA6	I/O	12	IO
PA7	I/O	11	IO
PB0	I/O	45	IO
PB1	I/O	46	IO
PB2/ADC	I/O	7	8 位 ADC 输入, 该管脚复用了 PB2 端口, 使用 ADC 输入时, 应配置端口为输入模式; 该管脚用作普通 IO 时, 只能使用输出功能
PB3	I/O	44	IO
PB4	I/O	43	IO
VIN1P	I	4	24bitADC 1 通道正端信号输入
VIN1N	I	5	24bitADC 1 通道负端信号输入
VPP	P	2	在线烧录供电脚, 正常应用接 3 脚 VDD
VDD	P	3	电源, 应用时可与 19 脚 VDD 接一起再接电源正
DGND	P	10	数字地, 应用时可与 VSS (17 脚) 接一起再接地
AGND	P	6	模拟地
/CS	I	14	当 /CS 为高电平时, 读写数据和命令无效, 串行接口电路复位; 当 /CS 为低电平和作为输入时, 读写数据和命令有效
/WR	I	15	在 /WR 信号的上升沿, DATA 线上的数据写入
DATA	I	16	外接上拉电阻的串行数据逻辑输入
VSS	P	17	系统接地, 应用时可与 DGND (10 脚) 接一起再接地
VLCD	P	18	LCD 电源输入
VDD	P	19	系统接电源正, 应用时可与 3 脚 VDD 接一起再接电源正
BZ	O	20	声音频率输出
COM0~COM3	O	21~24	LCD 共用端 (common) 驱动口
SEG14~SEG31	O	42~25	LCD 段 (segment) 驱动口

## 4 电气特性

### 极限参数

参数名称	参数符号	极限值	单位
电源电压	VDD	-0.3~VDD+0.5	V
输入电压	Vin	-0.3~VDD+0.3	V
输出电压	Vout	-0.3~VDD+0.5	V
工作温度范围	Topr	-25~+75	°C
储存温度范围	Tstg	-50~+125	°C
静电 ESD	人体模式 (HBM)	4000	V

### 电气特性

在 VDD=2.5~3.6V 及工作温度-25~+75°C 下测试，除非另有说明			TM8101-X2-MCU			单位
参数名称	参数符号	测试条件	最小值	典型值	最大值	
电源电压	VDD		2.5	3.3	3.6	V
正常工作电流	IDD1	24 位 ADC 开启、不带负载 片内 RC 振荡器		2.7		mA
待机电流	IDD2	高速 RC 振荡器关闭、不带负载 省电模式		85.2		uA
高电平输入电压	VIH	VDD=3.3V	2.8		VDD	V
低电平输入电压	VIL	VDD=3.3V	0		0.6	V

### MCU 电气参数

测试条件 (如无特殊说明均采用此条件): VDD=3.3V; 工作温度 25°C						
参数名称	参数符号	条件及说明	测试数据			
			最小值	典型值	最大值	单位
正常工作电流	IDD3	24 位 ADC 开启		2.5		mA
休眠模式电流	IDD4	高速 RC 振荡器关闭		85		uA
高速振荡器频率	HF		3.4	4.9	7.4	MHz
低速振荡器频率	LF		21.4	32	55	KHz
上电复位电压		VDD 上电		2	2.2	V
掉电复位电压		VDD 掉电	1.8	1.90	2.1	V
24 位 ADC 参数						
差模电压输入范围	VINP-VINN	Gain=128	±0.64534 (VDD/Gain)			mV
共模电压输入范围	Vin		AGND+1		VDD-1.3	V
增益误差				0.01		%of FS
积分非线性	INL	Gain=2		0.0002	0.001	%of FS
		Gain=128		0.0005	0.001	
有效位数 (Effective-Number-of-Bits)	EONB	Gain=128, 输出速率=10Hz	19.2			Bits
不动位数			17			
8 位 ADC 参数						
分辨率	LSB	ADC 能分辨的最小输入电压	12.89			mV
输入电压范围	Vin		0		VDD	V



**LCD 驱动器电气参数**

测试条件（如无特殊说明均采用此条件）：VDD=3.3V；工作温度 25℃						
参数名称	参数符号	条件及说明	测试数据			
			最小值	典型值	最大值	单位
正常工作电流	IDD5	不带负载 片内 RC 振荡器		200	350	uA
待机电流	IDD6	不带负载 省电模式		0.2	8	uA
DATA、BZ	IOL1	VOL=0.3V	0.5	1.2		mA
	IOH1	VOH=2.7V	-0.4	-0.8		mA
LCD 公共口源电流	IOL2	VOL=0.3V	80	150		uA
	IOH2	VOH=2.7V	-80	-120		uA
LCD 段管脚漏电流	IOL3	VOL=0.3V	60	120		uA
	IOH3	VOH=2.7V	-40	-70		uA
上拉电阻	RPH	DATA、/WR、/CS	40	80	150	KΩ
交流电气特性						
系统时钟	fSYS	片内 RC 振荡器		256		KHz
LCD 时钟	fLCD	片内 RC 振荡器		fSYS/ 1024		Hz
COM 时钟周期	fCOM	n: COM 个数		n/fLCD		S
串行数据时钟	fCLK				300	KHz
声音频率	fTONE	片内 RC 振荡器		2.0 或 4.0		KHz
DATA 管脚输出延迟 时间	fDD				2	uS
串行接口复位脉冲 宽度	fCS	/RS	4			uS

## 5 存储器结构

本单片机主要包括两种存储器模块：程序存储器和数据存储器。每一个模块都有自己的总线，在同一个周期内可对两种存储器模块同时进行访问。

### 5.1 程序存储器

程序存储器用来存放用户代码即存储程序。本单片机有一个 13 位宽的程序计数器，最大可寻址 8K x 16 的程序存储空间。本单片机具有 8K x 16 的程序存储器。



程序存储器结构

程序存储器中某些地址保留用作诸如复位和中断等特殊用途。0000H 是保留用做单片机复位后的程序起始地址。在芯片初始化或发生复位时，程序将会跳转到这个地址并开始执行。0004H~0018H 为中断向量，用以执行中断服务程序。1DFFH~1FFFH 为加密区不可用。

### 5.2 数据存储器

#### 5.2.1 数据存储器结构

数据存储器由特殊功能寄存器（SFR）和通用寄存器（GPR）组成。SFR 控制器件的操作，这些寄存器有特定的地址且与单片机的正确操作密切相关。大多特殊功能寄存器都可在程序控制下直接读取和写入，而有些是被加以保护而不对用户开放。GPR 是数据存储和改写的通用区域，所有地址都可在程序的控制下进行读取和写入。

单片机的数据存储器的起始地址都是“000H”，地址范围为 000H~1FFFH。

#### 数据存储列表

000H	INDF0	020H	INTCT1	040H	100H - 1FFFH 通用寄存器
001H	INDP0	021H	SMOD	041H	
002H	INDF1	022H		042H	
003H	INDP1	023H		043H	
004H		024H	WDTCT	044H	
005H	STATUS	025H	TMCT0	045H	
006H	ACC	026H	TMR	046H	
007H	PCL	027H	TMCT1	047H	
008H		028H		048H	
009H		029H		049H	
00AH		02AH		04AH	

00BH	PTA	02BH		04BH	
00CH	PTAPU	02CH	PWM0	04CH	
00DH	PTAIO	02DH	PWM1	04DH	
00EH		02EH	PWMCT	04EH	
00FH	PTB	02FH		04FH	
010H	PTBPU	030H	SARDATA	050H	
011H	PTBIO	031H	SARCT	051H	
012H		032H		052H	
013H		033H		053H	
014H		034H	SGADCCON	054H	
015H		035H	SGADCCT	055H	
016H		036H	SGADC2	056H	
017H		037H	SGADC1	057H	
018H		038H	SGADCO	058H	
019H		039H		059H	
01AH		03AH		05AH	
01BH		03BH		05BH	
01CH		03CH		05CH	
01DH		03DH		05DH	
01EH	INTEG	03EH		05EH	
01FH	INTCT0	03FH		05FH	

**特殊功能寄存器汇总**

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
000H	INDFO	寻址地址为 INDPO 的数据存储器 (非物理存储器)							
001H	INDPO	间接寻址 0, 数据存储器指针							
002H	INDF1	寻址地址为 INDP1 的数据存储器 (非物理存储器)							
003H	INDP1	间接寻址 1, 数据存储器指针							
005H	STATUS	—	—	TO	PDF	OV	Z	AC	C
006H	ACC	工作寄存器 (累加器)							
007H	PCL	程序计数器低字节							
00BH	PTA	端口 A 数据							
00CH	PTAPU	端口 A 上拉电阻控制							
00DH	PTAIO	端口 A 输入/输出状态控制寄存器							
00FH	PTB	—	—	—	—	—	端口 B 数据		
010H	PTBPU	—	—	—	—	—	端口 B 上拉电阻控制		
011H	PTBIO	—	—	—	—	—	端口 B 输入/输出状态控制寄存器		
01EH	INTEG	—	—	—	—	—	—	INTS1	INTS0
01FH	INTCT0	—	TM1F	TMOF	INTF	TM1E	TM0E	INTE	GIE
020H	INTCT1	—	—	SRAF	SGF	—	—	SRAE	SGE
021H	SMOD	—	—	—	—	CP1	CP0	IDLEC	FSYSEN
024H	WDTCT	—	—	—	—	—	WTS2	WTS1	WTS0
025H	TMCT0	—	—	TSEL	TON	—	TMFD2	TMFD1	TMFDO
026H	TMR	定时器计数器 Timer0 计数值							
027H	TMCT1	—	—	—	—	—	TNBON	TBFD1	TBFD0
02CH	PWM0	PWM 通道 0 的门限							
02DH	PWM1	PWM 通道 1 的门限							
02EH	PWMCT	—	—	—	—	PWM1 EN	PWM0 EN	PWM1 INV	PWM0 INV
030H	SARDATA	ADC 转换结果数据寄存器							
031H	SARCT	—	—	—	—	—	—	SAREN	SAR STA
034H	SGADCCON	—	—	—	—	—	SG CHA	SG SP1	SG SPO
035H	SGADCCT	—	—	—	—	—	SG EN	SG LOCK	TEMP EN
036H	SGADC2	24bitADC 转换数据 DATA[23:16]							
037H	SGADC1	24bitADC 转换数据 DATA[15:8]							
038H	SGADCO	24bitADC 转换数据 DATA[7:0]							

## 5.2.2 特殊功能寄存器描述

大部分特殊功能寄存器将在相关功能中详细描述，但有几个寄存器在此章节单独描述。

### 间接寻址寄存器 - INDF0, INDF1

间接寻址寄存器 INDF0 和 INDF1 的地址虽位于数据存储区，但其并没有实际的物理地址。间接寻址的方式是使用间接寻址寄存器和间接寻址指针对数据操作，以取代定义实际存储器地址的直接存储器寻址方法。在间接寻址寄存器 (INDF0 和 INDF1) 上的任何动作，将对间接寻址指针 (INDP0 和 INDP1) 所指定的存储器地址产生对应的读/写操作。它们总是成对出现，INDF0 和 INDP0 可以访问 Bank 0，而 INDF1 和 INDP1 可以访问所有 Bank (本款单片机只有 Bank0)。

### 间接寻址指针 - INDP0, INDP1

此系列单片机提供两个间接寻址指针，即 INDP0 和 INDP1。由于这些指针在数据存储区中能像普通的寄存器一般被操作，因此提供了一个寻址和数据追踪的有效方法。当对间接寻址寄存器进行任何操作时，单片机指向的实际地址是由间接寻址指针所指定的地址。INDP0, INDF0 用于访问 Bank 0，而 INDP1 和 INDF1 可通过 RBS 寄存器访问所有的 Bank。间接寻址只访问通用寄存器，即对于 9 位地址的最高位默认为高。

### 状态寄存器 - STATUS

这 8 位寄存器包括零标志位 (Z)、进位标志位 (C)、辅助进位标志位 (AC)、溢出标志位 (OV)，暂停标志位 (PDF)、和看门狗溢出标志位 (TO)。这些标志位同时记录单片机的状态数据和算术/逻辑运算。

C: 当加法运算的结果产生进位，或减法运算的结果没有产生借位时，则 C 被置位，否则 C 被清零，同时 C 也会被带进位的移位指令所影响。

AC: 当低半字节加法运算的结果产生进位，或低半字节减法运算的结果没有产生借位时，AC 被置位，否则 AC 被清零。

Z: 当算术或逻辑运算结果是零时，Z 被置位，否则 Z 被清零。

OV: 当运算结果高两位的进位状态异或结果为 1 时，OV 被置位，否则 OV 被清零。

PDF: 系统上电或执行“CLRWDT”指令会清零 PDF，而执行“STOP”指令则会置位 PDF。

TO: 系统上电或执行“CLRWDT”或“STOP”指令会清零 TO，而当 WDT 溢出则会置位 TO。

另外，当进入一个中断程序或执行子程序调用时状态寄存器将不会自动压入到堆栈中保存。假如状态寄存器的内容很重要，且中断子程序会改变状态寄存器的内容，则需要保存备份以备恢复。

Bit	7	6	5	4	3	2	1	0
Name	—	—	TO	PDF	OV	Z	AC	C
R/W	—	—	R	R	R/W	R/W	R/W	R/W

Bit 7~6 未使用，读为“0”

Bit 5 TO: 看门狗溢出标志位

0: 系统上电或执行“CLRWDT”或“STOP”指令

1: WDT 溢出

Bit 4 PDF: 暂停标志位

0: 系统上电或执行“CLRWDT”指令

1: 执行“STOP”指令将会置位 PDF 位。

Bit 3 OV: 溢出标志位

0: 不发生溢出时

1: 当运算结果高两位的进位状态异或结果为 1 时

Bit 2 Z: 零标志位

0: 算数运算或逻辑运算的结果不为零时

- 1: 算数运算或逻辑运算的结果为零时
- Bit 1 AC: 辅助进位标志位
- 0: 没有辅助进位时
- 1: 当低字节的加法造成进位或减法没有造成借位时
- Bit 0 C: 进位标志位 (用于借位时, 极性相反)
- 0: 没有进位时
- 1: 当加法造成进位或减法没有造成借位时, 同时移位指令也会影响 C 标志位 C 也受循环移位指令的影响。

### 5.3 EEPROM 数据存储器

该单片机的一个特性是内建 EEPROM 数据存储器。“Electrically Erasable Programmable Read Only Memory”为电可擦可编程只读存储器, 由于其非易失的存储结构, 即使在电源掉电的情况下存储器内的数据仍然保存完好。这种存储区扩展了 ROM 空间对设计者来说增加了许多新的应用机会。EEPROM 可以用来存储产品编号、校准值、用户特定数据、系统配置参数或其它产品信息等。

在整个 VDD 范围内的正常运行期间, EEPROM 数据存储器是可以读写的。该存储器并不直接映射寄存器到文件空间, 而是通过端口以 I2C 的方式访问 (具体端口连接见 11.1 端口概述)。

EEPROM 数据存储器可按字节进行读和写。一个字节的写操作将自动擦除并写入新的值 (即先擦除后写入)。EEPROM 是一种具有高擦/写周期的存储器。写入的时间由片内定时器控制, 它随着电压、温度以及器件的不同而不同。

#### EEPROM 数据存储器大小

EEPROM 数据存储器大小	地址范围
256byte	00H - FFH

## 6 程序计数器

### 6.1 概述

程序计数器(PC)控制程序存储器中的指令执行顺序，它可以寻址整个程序存储器，取得指令码后，程序计数器(PC)会自动加一，指向下一个指令码的地址。但如果执行跳转、向PCL赋值、子程序调用、中断、中断返回、子程序返回等操作时，PC会加载与指令相关的地址而不是下一条指令的地址。当遇到条件跳转指令且符合跳转条件时，当前指令执行过程中读取的下一条指令将会被丢弃，且会插入一个空指令操作周期，随后才能取得正确的指令。反之，就会顺序执行下一条指令。

程序计数器(PC)是13位宽度，低8位可以通过PCL寄存器访问，高5位不能访问。可容纳8Kx16位程序地址。对PCL赋值将会产生一个跳转动作，跳转范围为当前页的256个地址。

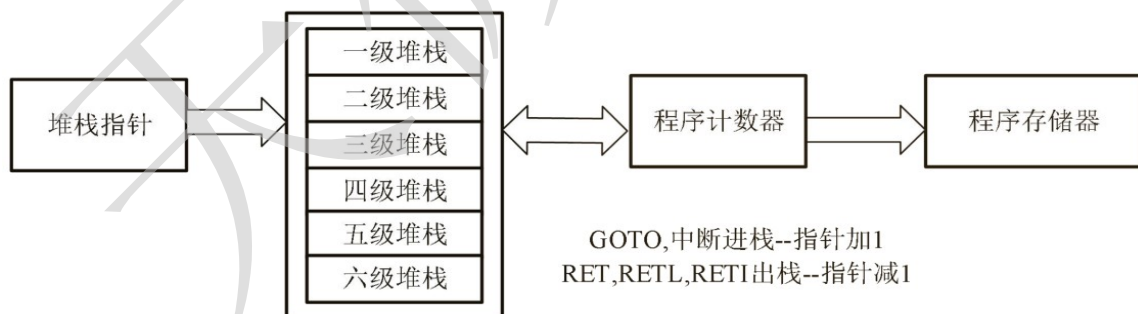
### 6.2 程序计数器低字节寄存器

为了提供额外的程序控制功能，程序计数器的低字节被设置在数据存储器的特殊功能区域，程序员可对此寄存器进行操作，很容易直接跳转到其它程序地址。直接给PCL寄存器赋值将导致程序直接跳转到专用程序存储器某一地址，然而，由于寄存器只有8位的长度，因此只允许在本页的程序存储器中跳转。注意，使用这种运算时，会插入一个空指令周期。

### 6.3 堆栈

本单片机的堆栈缓存器共6层，位宽为13，堆栈存储程序执行分支的返回地址。堆栈既不占用程序存储空间也不占用数据存储空间，堆栈数据与栈指针均不能读写。

当系统复位后堆栈指针会指向堆栈顶部。当发生子程序调用及中断时，程序计数器(PC)值会被压入堆栈缓存器，当从中断或子程序返回时将数值返回给程序计数器(PC)，下图说明其工作原理。



堆栈缓存器的使用将遵循“先进后出”的原则。

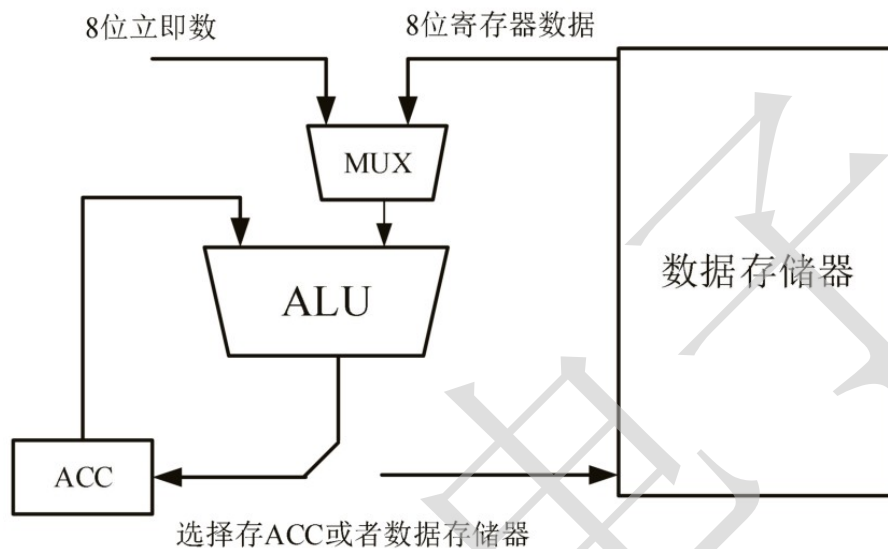
如果堆栈已满，且有非屏蔽的中断发生，则只有中断请求标志位会被置位，而中断响应会被禁止，直到堆栈指针发生递减(执行RET、RETL或RETIN指令)，中断才会被响应。然而即使堆栈已满，GOTO指令仍然可以执行，从而造成堆栈溢出。使用时应避免堆栈溢出的情况发生，因为这样会造成不可预期的程序分支指令的执行错误。如果堆栈溢出，第一个保存在堆栈中的PC会丢失。



## 7 算术逻辑单元

单片机包含一个 8 位 ALU 和一个 8 位工作寄存器 ACC。ALU 是一个通用的算术逻辑单元，它对工作寄存器和数据寄存器中的数据进行算术和逻辑运算。ACC 是一个 8 位宽、用于 ALU 运算的工作寄存器。

### ALU 与 ACC 的操作



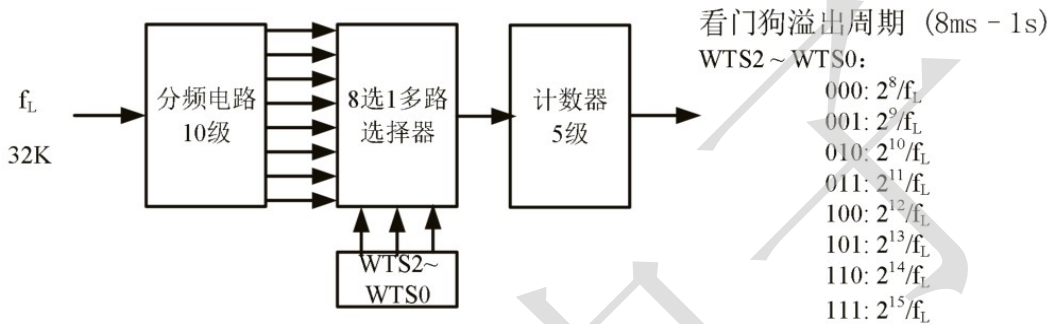
ALU 具有 8 位宽，能够进行加、减、移位和逻辑操作。除非特别指明，算术运算一般是以二进制补码形式进行。在 2 个操作数的指令中，典型情况下，其中一个操作数是在 ACC 中，另一个操作数来自于数据寄存器或是一个立即数。在单操作数指令中，操作数放在 ACC 中或某个数据寄存器中。

## 8 看门狗定时器

### 8.1 概述

看门狗定时器 (Watch Dog Timer) 时钟源来自于内部低速振荡器  $f_L$ ，是一个片内自振式的 RC 振荡定时器，无需任何外围组件，即使芯片的主时钟停止工作，WDT 也能保持计时。WDT 计时溢出将产生复位。

#### 看门狗定时器结构框图



在正常操作期间，一次 WDT 超时溢出将产生一次器件复位。如果器件处于休眠状态，一次 WDT 超时溢出将唤醒器件，使其继续正常操作（即称作 WDT 唤醒）。正常工作期间，用户需在看门狗定时器溢出前将看门狗定时器清零以防止其产生复位，有两种方法可以用来清零 WDT 的内容。第一种是通过清除看门狗指令；而第二种是通过“STOP”指令。

### 8.2 相关特殊功能寄存器

#### 看门狗定时器控制寄存器

WDTCT 寄存器用于控制 WDT 溢出周期选择。任何单片机复位，WDTCT 都为 011B，且这个值在休眠模式、空闲模式 0 和空闲模式 1 下 WDT 溢出也不会改变。

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	WTS2	WTS1	WTS0
R/W	—	—	—	—	—	R/W	R/W	R/W

Bit 7~3 未使用，读为“0”

Bit 2~0 WTS2 ~ WTS0: 选择看门狗溢出周期

- 000:  $2^8/f_L$
- 001:  $2^9/f_L$
- 010:  $2^{10}/f_L$
- 011:  $2^{11}/f_L$  (默认)
- 100:  $2^{12}/f_L$
- 101:  $2^{13}/f_L$
- 110:  $2^{14}/f_L$
- 111:  $2^{15}/f_L$

这三位控制 WDT 时钟源的分频比，从而实现了对 WDT 溢出周期的控制。



## 9 系统时钟与工作模式

### 9.1 振荡器

该单片机有两个内部全集成的振荡器，高速振荡器为内部 4.9MHz RC 振荡器，低速振荡器为内部 32kHz RC 振荡器。

振荡类型	名称	频率范围
内部高速 RC	HIRC	4.9MHz
内部低速 RC	LIRC	32kHz

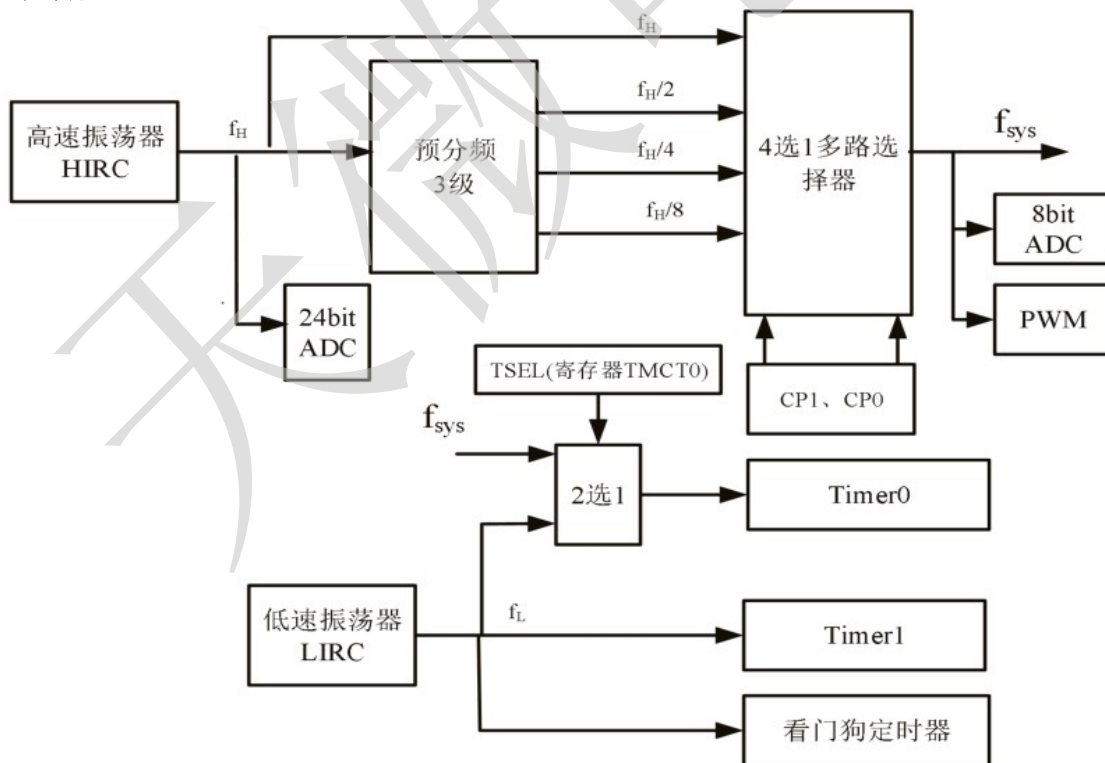
#### 内部高速 RC 振荡器 - HIRC

内部高速 RC 振荡器是一个全集成的系统振荡器，不需其它外部器件。芯片在制造时进行调整且内部含有频率补偿电路，使得振荡频率因 VDD、温度以及芯片制成工艺不同的影响减至最低程度。

#### 内部低速 RC 振荡器 - LIRC

内部 32kHz 系统振荡器为低速振荡器。LIRC 是一个全集成的 RC 振荡器，无需外接器件在常温 5V 条件下，振荡频率值为 32kHz。芯片在制造时进行调整且内部含有频率补偿电路使得振荡频率因 VDD、温度以及芯片制成工艺不同的影响减至最低程度。系统上电，LIRC 振荡器就使能，不存在将该振荡器除能的寄存器位。

#### 系统时钟配置



## 9.2 相关特殊功能寄存器

SMOD 寄存器 - 用于控制单片机内部时钟。

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	CP1	CP0	IDLEC	FSYSEN
R/W	-	-	-	-	R/W	R/W	R/W	R/W

Bit 7~4 未使用，读为“0”

Bit 3~2 CP1~CP0: 系统时钟选择位，用于选择系统时钟源。

- 00:  $f_H$
- 01:  $f_H/2$
- 10:  $f_H/4$
- 11:  $f_H/8$

Bit 1 IDLEC: 空闲模式控制位

- 0: 除能 (STOP 指令执行后进入休眠模式)
- 1: 使能

此位为空闲模式控制位，用于决定 STOP 指令执行后发生的动作。若此位为高，当指令 STOP 执行后，单片机进入空闲模式，至于进入空闲模式 0 还是空闲模式 1 取决于 FSYSEN 位，若 FSYSEN 位为高，在空闲模式 1 中 CPU 停止运行，系统时钟将继续工作以保持外围功能继续工作；若 FSYSEN 为低，在空闲模式 0 中 CPU 和系统时钟都将停止运行。若此位为低，单片机将在 STOP 指令执行后进入休眠模式。

Bit 0 FSYSEN: IDLE 模式时， $f_{SYS}$  控制位

- 0: 除能
- 1: 使能

## 9.3 系统工作模式

该系列单片机有四种不同的工作模式，每种有它自身的特性，根据应用中不同的性能和功耗要求可选择不同的工作模式。单片机正常工作有正常模式。剩余的 3 种工作模式：休眠模式、空闲模式 0 和空闲模式 1 用于单片机 CPU 关闭时以节省耗电。

工作模式	CPU	$f_{SYS}$	$f_L$	$f_H$
正常模式	On	$f_H \sim f_H/64$	On	On
空闲模式 0	Off	Off	On	On
空闲模式 1	Off	On	On	On
休眠模式	Off	Off	On	Off

### 正常模式

顾名思义，这是主要的工作模式，单片机的所有功能均可在此模式中实现且系统时钟由一个高速振荡器提供。该模式下单片机正常工作的时钟源来自 HIRC 振荡器。高速振荡器频率可被分为 1~8 的不等比率，实际的比率由 SMOD 寄存器中的 CP1~CP0 选择的。单片机使用高速振荡器分频作为系统时钟可减小工作电流。

### 休眠模式

在 STOP 指令执行后且 SMOD 寄存器中 IDLEC 位为低时，系统进入休眠模式。在休眠模式中，CPU 停止运行。看门狗定时器功能使能， $f_L$  继续运行。

### 空闲模式 0

执行 STOP 指令后且 SMOD 寄存器中 IDLEC 位为高，SMOD 寄存器中 FSYSEN 位为低时，系统进入空闲模式 0。在空闲模式 0 中，系统时钟停止，CPU 停止工作，但一些外围功能如看门狗定时器和定时/计数器将继续工作。

**空闲模式 1**

执行 STOP 指令后且 SMOD 寄存器中 IDLEC 位为高，SMOD 寄存器中 FSYSSEN 位为高时，系统进入空闲模式 1。在空闲模式 1 中，CPU 停止，但会提供一个时钟源给一些外围功能如看门狗定时器和定时/计数器。在空闲模式 1 中，系统时钟继续运行。

MCU 核的省电工作模式，主要通过关闭相关部分的时钟来实现：

模式	状态描述	进入方式
休眠模式	MCU 停止读取指令，进入等待 若 TIMERO 选择系统时钟，将关闭；若选择 $f_L$ ，将继续打开 关闭 8bit ADC 时钟 关闭 PWM 时钟 关闭 24bit ADC 使能 无视其他配置强行关闭（24bit ADC）	IDLEC 清 0 STOP 指令
空闲模式 0	MCU 停止读取指令，进入等待 若 TIMERO 选择系统时钟，将关闭；若选择 $f_L$ ，将继续打开 关闭 8bit ADC 时钟 关闭 PWM 时钟	IDLEC 置 1 FSYSSEN 清 0 STOP 指令
空闲模式 1	MCU 停止读取指令，进入等待 相关时钟不关闭	IDLE 置 1 FSYSSEN 置 1 STOP 指令

PWM、8bit ADC 时钟即为系统时钟

系统进入休眠或空闲模式之后，可以通过以下几种方式唤醒：

- 系统中断
- WDT 溢出

若由 WDT 溢出唤醒，则会发生看门狗定时器复位。这种唤醒方式都会使系统复位，可以通过状态寄存器中 TO 和 PDF 位来判断它的唤醒源。系统上电或执行清除看门狗的指令，会清零 PDF；执行 STOP 指令，PDF 将被置位。看门狗计数器溢出将会置位 TO 标志并唤醒系统，这种复位会重置程序计数器和堆栈指针，其它标志保持原有状态。

如果系统是通过中断唤醒，则有两种可能发生。第一种情况是：相关中断除能或是中断使能且堆栈已满，则程序会在“STOP”指令之后继续执行。这种情况下，唤醒系统的中断会等到相关中断使能或有堆栈层可以使用之后才执行。第二种情况是：相关中断使能且堆栈未满，则中断可以马上执行。如果在进入休眠或空闲模式之前中断标志位已经被设置为“1”，则相关中断的唤醒功能将无效。

## 10 复位与初始化

### 10.1 复位

复位发生时，所有的系统寄存器将恢复默认状态，程序停止运行，同时程序计数器 PC 清零，复位结束后程序从复位向量 0000H 开始运行。

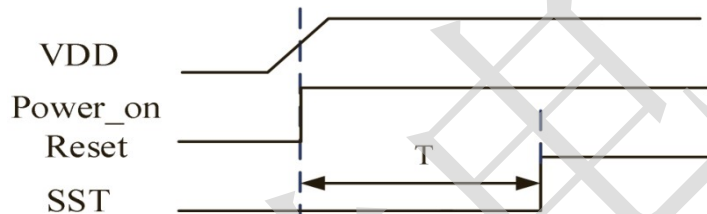
任何一种复位情况都需要一定的响应时间，系统提供完善的复位流程以保证复位动作的顺利进行。

本单片机包括内部和外部事件触发复位，共有三种复位方式：

#### 上电复位

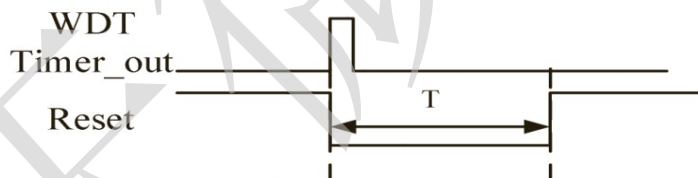
这是最基本且不可避免的复位，发生在单片机上电后。除了保证程序存储器从开始地址执行，上电复位也使得其它寄存器被设定在预设条件，所有的输入/输出端口控制寄存器在上电复位时会保持高电平，以确保上电后所有引脚被设定为输入状态。

由于上电条件不稳定，单片机具有内部复位功能。RC 电路产生一段延迟时间，能确保电源电压稳定时使得 POR 较长时间处于低电平。在此期间，单片机无法正常工作。当 POR 达到一定的电压值后，再经过一段复位延迟时间 T（典型值为 50ms）后，单片机才开始正常工作。



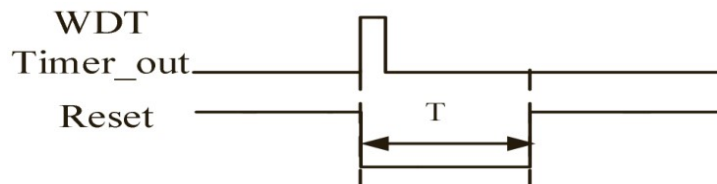
#### 正常运行时看门狗溢出复位

除了看门狗溢出标志位 T0 将被设为“1”之外，正常运行时看门狗溢出复位和外部硬件上电复位相同。复位延时典型值： $T=16.7\text{ms}$ 。



#### 空闲或休眠时看门狗溢出复位

空闲或休眠时看门狗溢出复位和其它种类的复位有些不同，除了程序计数器与堆栈指针将被清“0”及 T0 位被设为“1”外，绝大部分的条件保持不变。复位延时为 1 个低速振荡器周期。



## 10.2 初始化

在单片机上电复位之后，各功能单元初始化的情形，列于下表。

单元	复位后情况
程序计数器	清零
中断	所有中断被除能
看门狗定时器	WDT 清零并重新计时
Timer0	Timer0 停止, timer1 停止
输入/ 输出口	所有 I/O 设为输入模式
堆栈指针	堆栈指针指向堆栈顶端

复位方式对 FSR 的影响。

‘n’ 表示未定义

‘x’ 表示未知

‘k’ 表示保持

寄存器	上电复位 & LVR		WDT 溢出 正常工作模式		WDT 溢出 空闲或休眠模式	
INDFO	nnnn	nnnn	nnnn	nnnn	nnnn	nnnn
INDPO	0000	0000	kkkk	kkkk	kkkk	kkkk
INDF1	nnnn	nnnn	nnnn	nnnn	nnnn	nnnn
INDP1	0000	0000	kkkk	kkkk	kkkk	kkkk
STATUS	nn00	0000	nn1k	kkkk	nn11	kkkk
ACC	0000	0000	kkkk	kkkk	kkkk	kkkk
PCL	0000	0000	0000	0000	0000	0000
PTA	1111	1111	1111	1111	kkkk	kkkk
PTAPU	0000	0000	0000	0000	kkkk	kkkk
PTAIO	1111	1111	1111	1111	kkkk	kkkk
PTB	nnn1	1111	nnn1	1111	nnnk	kkkk
PTBPU	nnn0	0000	nnn0	0000	nnnk	kkkk
PTBIO	nnn1	1111	nnn1	1111	nnnk	kkkk
INTEG	nnnn	nn00	nnnn	nn00	nnnn	nnkk
INTCT0	n000	0000	n000	0000	nkkk	kkkk
INTCT1	nn00	nn00	nn00	nn00	nnkk	nnkk
SMOD	nnnn	0010	nnnn	0010	nnnn	kkkk
WDTCT	nnnn	n011	nnnn	n011	nnnn	nkkk
TMCT0	nn00	n000	nn00	n000	nnkk	nkkk
TMR	0000	0000	0000	0000	kkkk	kkkk
TMCT1	nnnn	n000	nnnn	n000	nnnn	nkkk
PWM0	1111	1111	1111	1111	kkkk	kkkk
PWM1	1111	1111	1111	1111	kkkk	kkkk
PWMCT	nnnn	0000	nnnn	0000	nnnn	kkkk
SARDATA	0000	0000	0000	0000	kkkk	kkkk
SARCT	nnnn	nn00	nnnn	nn00	nnnn	nnkk
SGADCCON	nnnn	n000	nnnn	n000	nnnn	nkkk
SGADCCT	nnnn	n100	nnnn	n100	nnnn	nkkk
SGADC2	0000	0000	0000	0000	kkkk	kkkk
SGADC1	0000	0000	0000	0000	kkkk	kkkk
SGADC0	0000	0000	0000	0000	kkkk	kkkk

# 11 输入/输出端口

## 11.1 概述

通用 I/O 引脚可看作是最简单的外设，单片机通过 I/O 端口监视和控制其它设备。为了增强器件的灵活性和功能，一些引脚被定义为多功能复用引脚。这些功能由器件上相应外设的特点决定。一般来说，当相应的外设使能时，其对应的引脚不能作为通用 I/O 引脚使用。

本单片机有两组 I/O 端口：PORTA、PORTB（13 个 I/O，其中 11 个为芯片引脚，PORTB[4:3]用于内部与 EEPROM 通信，PB3 连接 SCL，PB4 连接 SDA），可通过端口数据寄存器直接存取这些端口。

## 11.2 相关特殊功能寄存器

### 端口 A

端口 A 是 8 位宽的双向端口，它所对应的输入输出状态控制寄存器是 PTAIO。将 PTAIO 的一个位置 1 可以将相应的引脚配置为输入。清零 PTAIO 的一个位可将相应的端口 A 引脚配置为输出。

#### PTA - 端口 A 数据

Bit	7	6	5	4	3	2	1	0
Name	P7	P6	P5	P4	P3	P2	P1	P0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

#### PTAIO - 端口 A 输入/输出状态控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	P7	P6	P5	P4	P3	P2	P1	P0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0 PTA 口 bit 7 ~ bit 0 输入/输出控制

0: 输出

1: 输入

#### PTAPU - 端口 A 上拉电阻

每个端口 A 引脚都有可单独配置的内部弱上拉。由特殊功能寄存器 PTAPU 使能或禁止每个弱上拉。当将端口引脚配置为输出时，其弱上拉会自动切断。

Bit	7	6	5	4	3	2	1	0
Name	P7	P6	P5	P4	P3	P2	P1	P0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0 PTA 口 bit 7 ~ bit 0 上拉电阻控制

0: 除能

1: 使能

### 端口 B

端口 B 是 5 位宽的双向端口，它所对应的输入输出状态控制寄存器为 PTBIO，对应的上拉电阻控制寄存器为 PTBPU。

**PTB - 端口 B 数据**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	P4	P3	P2	P1	P0
R/W	—	—	—	R/W	R/W	R/W	R/W	R/W

**PTBIO - 端口 B 输入/输出状态控制寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	P4	P3	P2	P1	P0
R/W	—	—	—	R/W	R/W	R/W	R/W	R/W

Bit 7~5 未使用，读为“0”

Bit 4~0 PTB 口 bit 4 ~ bit 0 输入/输出控制

0: 输出

1: 输入

**PTBPU - 端口 B 上拉电阻**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	P4	P3	P2	P1	P0
R/W	—	—	—	R/W	R/W	R/W	R/W	R/W

Bit 7~5 未使用，读为“0”

Bit 4~0 PTB 口 bit 4 ~ bit 0 上拉电阻控制

0: 除能

1: 使能



## 12 定时计数器 Timer0

### 12.1 概述

Timer0 有以下特性：

8 位定时计数器寄存器 TMR

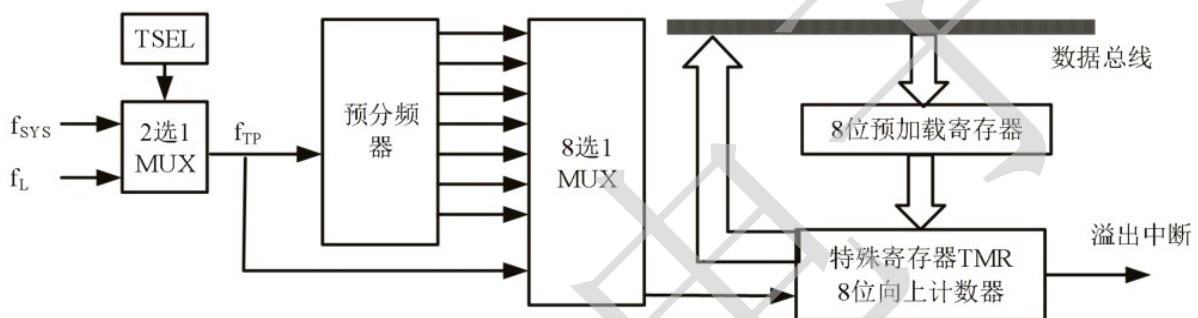
可读写

8 位预加载寄存器

二选一内部时钟信号

溢出中断

#### Timer0 结构图



定时计数器的时钟源可以来自系统时钟  $f_{SYS}$  或低速振荡器  $f_{LIRC}$ ，由 TMCT0 寄存器的 TSEL 位选择使用哪种时钟源。内部时钟首先由预分频器分频，分频比由定时器控制寄存器的位 TMFD2 ~ TMFD0 来确定。

### 12.2 工作原理

定时/计数器可以用来测量固定时间间隔，当定时器发生溢出时，就会产生一个内部中断信号。 $sys\_clk$  或  $f_{LIRC}$  被用来当定时器的输入时钟源。然而，该定时器时钟源被预分频器进一步分频，分频比是由定时器控制寄存器的 TMFD2 ~ TMFD0 位来确定。定时器使能位，即 TON 位需要设为逻辑高，才能令定时器工作。每次内部时钟由高到低的电平转换都会使定时器值增加一。当定时器计数已满，即溢出时，会产生中断信号且定时器会重新载入预加载寄存器的值，然后继续计数。定时器溢出以及相应的内部中断产生也是唤醒暂停模式的一种方法，通过设置中断寄存器中的定时器中断使能位为 0，可以禁止计数器中断。

注意，如果定时/计数器在关闭条件下，写数据到预置寄存器，会立即写入实际的定时器 TMR。而如果定时/计数器已经打开且正在计数，在这个周期内写入到预加载寄存器的任何新数据将保留在预加载寄存器，直到溢出发生时才被写入实际定时器。

### 12.3 相关特殊功能寄存器

定时/计数寄存器 - TMR

定时/计数寄存器 TMR，是特殊功能寄存器，用于储存定时器的当前值。当收到一个内部计数脉冲，此寄存器的值将会加一。定时器将从预加载寄存器所载入的值开始计数，到 FFH 时定时器溢出且会产生一个内部中断信号。定时器的值随后被预加载寄存器的值重新载入并继续计数。



**定时/计数控制寄存器 - TMCT0**

定时/计数控制寄存器为 TMCT0，配合相应的 TMR 寄存器控制定时/计数器的全部操作。在使用定时器之前，需要先正确地设定定时/计数控制寄存器，以保证定时器能正确操作，而这个过程通常在程序初始化期间完成。

定时/计数控制寄存器的第 4 位即 TON，用于定时器开关控制，设定为逻辑高时，计数器开始计数，而清零时则停止计数。定时/计数控制寄存器的第 0~2 位用来控制输入时钟预分频器。TSEL 位用来选择内部时钟源。

Bit	7	6	5	4	3	2	1	0
Name	—	—	TSEL	TON	—	TMFD2	TMFD1	TMFD0
R/W	—	—	R/W	R/W	—	R/W	R/W	R/W

Bit 7~6 未使用，读为“0”

Bit 5 TSEL: 定时器时钟源选择位，同时作用于时基模块

0:  $f_{SYS}$

1:  $f_L$

Bit 4 TON: Timer0 控制位

0: 除能

1: 使能

Bit 3 未使用，读为“0”

Bit 2~0 TMFD2 ~ TMFD0: 定时器预分频比选择位

定时器内部时钟 =

000:  $f_{TP}$

001:  $f_{TP}/2$

010:  $f_{TP}/4$

011:  $f_{TP}/8$

100:  $f_{TP}/16$

101:  $f_{TP}/32$

110:  $f_{TP}/64$

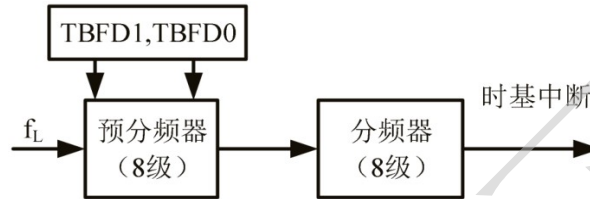
111:  $f_{TP}/128$

## 13 定时计数器 Timer1

### 13.1 概述

定时计数器 Timer1 为固定周期计数器，时钟源来自内部时钟源  $f_L$ 。由 TMCT1 寄存器相关位对时钟进行分频，以提供更长的时基计数周期。当计数器溢出时，将产生时基中断。

Timer1 结构图



### 13.2 相关特殊功能寄存器

时基计数器控制器 - TMCT1

时基计数器为固定周期计数器，时钟源来自内部时钟源  $f_L$ 。输入时钟首先经过分频器，分频率由程序设置 TMCT1 寄存器相关位获取合适的分频值以提供更长的时基计数周期。当计数器溢出时，将产生时基中断。

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	TBON	TBFD1	TBFD0
R/W	—	—	—	—	—	R/W	R/W	R/W

Bit 7~3 未使用，读为“0”

Bit 2 TBON: TIMER1 控制位

0: 除能

1: 使能

Bit 1~0 TBFD1, TBFD0: 预分频选择位

00:  $f_L/32$  中断周期 256ms

01:  $f_L/64$  中断周期 512ms

10:  $f_L/128$  中断周期 1024ms

11:  $f_L/256$  中断周期 2048ms

## 14 中断

### 14.1 概述

单片机提供以下中断（按优先级从高到低排列）

中断类型	优先级别
外部中断	1
Timer0 溢出中断	2
Timer1 溢出中断	3
24 位 ADC 中断	4
8 位 ADC 中断	5

中断是单片机一个重要功能。当发生外部中断或内部中断时，系统会中止当前的程序，而转到相对应的中断服务程序中。该系列单片机提供一个外部中断和多个内部中断。外部中断由 INT 引脚信号触发，而内部中断由 ADC 中断、定时/计数器控制。

### 14.2 相关特殊功能寄存器

寄存器中含有中断控制位和中断请求标志位。中断控制位用于使能或除能各种中断，中断请求标志位用于存放当前中断请求的状态。它们都按照特定的模式命名，前面表示中断类型的缩写，紧接着的字母“E”代表使能/除能位，“F”代表请求标志位。

中断控制寄存器 0 - INTCT0

Bit	7	6	5	4	3	2	1	0
Name	—	TM1F	TMOF	INTF	TM1E	TMOE	INTE	GIE
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bit 7 未使用，读为“0”
- Bit 6 TM1F: Timer1 中断请求标志位
  - 0: 无请求
  - 1: 中断请求
- Bit 5 TMOF: Timer0 中断请求标志位
  - 0: 无请求
  - 1: 中断请求
- Bit 4 INTF: 外部中断请求标志位
  - 0: 无请求
  - 1: 中断请求
- Bit 3 TM1E: Timer1 中断控制位
  - 0: 除能
  - 1: 使能
- Bit 2 TMOE: Timer0 中断控制位
  - 0: 除能
  - 1: 使能
- Bit 1 INTE: 外部中断控制位
  - 0: 除能
  - 1: 使能

Bit 0 GIE: 总中断控制位  
 0: 除能  
 1: 使能

#### 中断控制寄存器1 - INTCT1

Bit	7	6	5	4	3	2	1	0
Name	—	—	SARF	SGF	—	—	SARE	SGE
R/W	—	—	R/W	R/W	—	—	R/W	R/W

Bit 7~6 未使用, 读为“0”  
 Bit 5 SARF: 8位ADC中断请求标志位  
 0: 无请求  
 1: 中断请求  
 Bit 4 SGF: 24位ADC中断请求标志位  
 0: 无请求  
 1: 中断请求  
 Bit 3~2 未使用, 读为“0”  
 Bit 1 SARE: 8位ADC中断控制位  
 0: 除能  
 1: 使能  
 Bit 0 SGE: 24位ADC中断控制位  
 0: 除能  
 1: 使能

#### 外部中断边沿控制 - INTEG

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	INTS1	INTSO
R/W	—	—	—	—	—	—	R/W	R/W

Bit 7~2 未使用, 读为“0”  
 Bit 1~0 INTS1, INTSO: INT脚中断边沿控制位  
 00: 除能  
 01: 上升沿  
 10: 下降沿  
 11: 双沿

### 14.3 工作原理

若中断事件条件产生, 如Timer0溢出, 相关中断请求标志将置起。中断标志产生后程序是否会跳转至相关中断向量执行将由中断使能位决定。若使能位为“1”, 程序将跳至相关中断向量中执行; 若使能位为“0”, 即使中断请求标志置起中断也不会发生, 程序也不会跳转至相关中断向量执行。若总中断使能位为“0”, 所有中断都将除能。

当中断发生时, 下条指令的地址将被压入堆栈。相应的中断向量地址加载至PC中。系统将从此向量取下条指令。中断向量处通常为跳转指令, 以跳转到相应的中断服务程序。中断服务程序必须以“RETIN”指令返回至主程序, 以继续执行原来的程序。

各个中断优先级的次序为: 外部中断 - Timer0中断 - Timer1中断 - 24位ADC中断 - 8位ADC中断。各个中断有自己的中断向量, 依次为: 04h, 08h, 0Ch, 10h, 14h。

一旦中断子程序被响应, 系统将自动清除GIE位, 所有其它的中断将被屏蔽, 这个方式可以防止任何进一步的中断嵌套。其它中断请求可能发生在此期间, 虽然中断不会立即响应, 但是中断请求标志位会被记录。

当某个中断服务子程序正在执行时，有另一个中断要求立即响应，那么 GIE 位应在程序进入中断子程序后置位，以允许此中断嵌套。如果堆栈已满，即使此中断使能，中断请求也不会被响应，直到堆栈指针减少为止。如果要求立刻动作，则堆栈必须避免成为储满状态。请求同时发生时，按中断优先级执行。所有被置起的中断请求标志都可把单片机从休眠或空闲模式中唤醒，若要防止唤醒动作发生，在单片机进入休眠或空闲模式前应将相应的请求标志位置起。

## 14.4 中断类型介绍

### 外部中断

通过 INT 引脚上的信号变化可控制外部中断。当触发沿选择位设置好触发类型，INT 引脚的状态发生变化，外部中断请求标志 INTF 被置位时外部中断请求产生。若要跳转到相应中断向量地址，总中断控制位 GIE 和相应中断使能位 INTE 需先被置位。此外，必须使用 INTEG 寄存器使能外部中断功能并选择触发沿类型。外部中断引脚和普通 I/O 口共用，如果相应寄存器中的中断使能位被置位，此引脚将被作为外部中断脚使用。此时该引脚必须通过设置控制寄存器，将该引脚设置为输入口。当中断使能，堆栈未满并且外部中断脚状态改变，将调用外部中断向量子程序。当响应外部中断服务子程序时，中断请求标志位 INTF 会自动复位且 GIE 位会被清零以除能其它中断。注意，即使此引脚被用作外部中断输入，其上拉电阻仍保持有效。

寄存器 INTEG 被用来选择有效的边沿类型来触发外部中断。可以选择上升沿还是下降沿或双沿触发都产生外部中断。INTEG 也可以用来除能外部中断功能。

### Timer0 中断

要使定时/计数器 Timer0 中断发生，总中断控制位 GIE 和相应的内部中断使能位 TMOE 必须先被置位。当 Timer0 溢出，相应的中断请求标志位 TMOF 将置位并触发 Timer0 中断，若中断使能，堆栈未满，将调用位于计数器 timer0 中断向量处的子程序。当响应中断服务子程序时，中断请求标志位 TMOF 会被自动复位且 GIE 位会被清零以除能其它中断。

### Timer1 中断

定时/计数器 Timer1 中断提供一个固定周期的中断信号，由定时器功能产生溢出信号控制。当中断请求标志 TM1F 被置位时，中断请求发生。当总中断使能位 GIE 和 Timer1 使能位 TM1E 被置位，允许程序跳转到各自的中断向量地址。当中断使能，堆栈未满且 Timer1 溢出时，将调用它的中断向量子程序。当响应中断服务子程序时，相应的中断请求标志位 TM1F 会自动复位且 GIE 位会被清零以除能其它中断。

### 24 位 ADC 转换中断

24 位 ADC 转换器中断由 A/D 转换动作的结束来控制。当 A/D 转换器中断请求标志被置位，即 A/D 转换过程完成时，中断请求发生。当总中断使能位 GIE 和 24 位 ADC 中断使能位 SGE 被置位，允许程序跳转到各自的中断向量地址。当中断使能，堆栈未满且 A/D 转换动作结束时，将调用它的中断向量子程序。当响应中断服务子程序时，相应的中断请求标志位 SGF 会自动清零。GIE 位也会被清零以除能其它中断。

### 8 位 ADC 转换中断

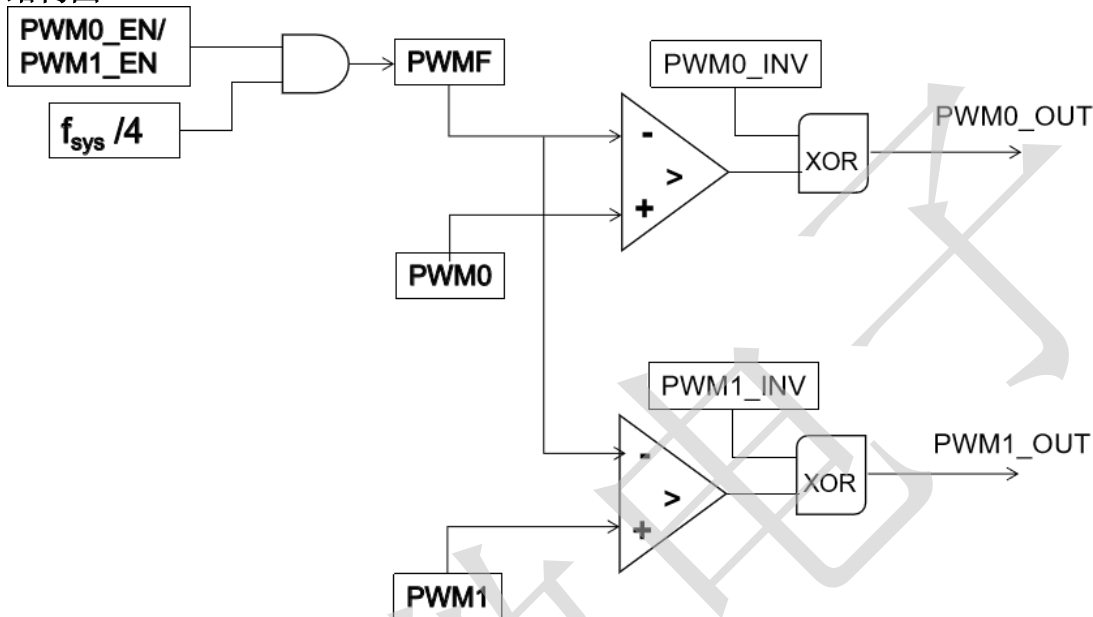
8 位 ADC 转换器中断由 A/D 转换动作的结束来控制。当 A/D 转换器中断请求标志被置位，即 A/D 转换过程完成时，中断请求发生。当总中断使能位 GIE 和 8 位 ADC 中断使能位 SRAE 被置位，允许程序跳转到各自的中断向量地址。当中断使能，堆栈未满且 A/D 转换动作结束时，将调用它的中断向量子程序。当响应中断服务子程序时，相应的中断请求标志位 SRAF 会自动清零。GIE 位也会被清零以除能其它中断。

## 15 脉宽调制模块

### 15.1 概述

本模块主要用于输出特定占空比的 PWM 波形。

结构图



PWMF 计数器为 9 位。  
PWM0 和 PWM1 分别为门限计数器，分别为 8 位。  
PWM0\_EN 与 PWM1\_EN 输出到端口模块，用于切换到 PWM 输出

PWM 开启，PWM0\_EN = 1，PWM1\_EN = 1  
PWM0\_INV = 0:  
当 PWMF <= PWM0 时，PWM0\_OUT 输出 1  
PWM0\_INV = 1:  
当 PWMF <= PWM0 时，PWM0\_OUT 输出 0

PWM1\_INV = 0:  
当 PWMF <= PWM1 时，PWM1\_OUT 输出 1  
PWM1\_INV = 1:  
当 PWMF <= PWM1 时，PWM1\_OUT 输出 0  
(可以通过 PWM1\_INV、PWM0\_INV 控制输出反相)

### 15.2 相关特殊功能寄存器

- 1、PWM0[7:0]: PWM 通道 0 的门限
- 2、PWM1[7:0]: PWM 通道 1 的门限

3、PWMCT (0x2E): PWM 控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	PWM1_EN	PWM0_EN	PWM1_IN	PWM0_INV

							V	
R/W	—	—	—	—	R/W	R/W	R/W	R/W

Bit 7~4 未使用，读为“0”

Bit 3 PWM1\_EN: PWM1 使能

1: 开启

0: 关闭

Bit 2 PWM0\_EN: PWM0 使能

1: 开启

0: 关闭

Bit 1 PWM1\_INV: PWM1 的输出配置位

1: ( PWM1\_EN=1) 当 PWMF <= PWM1 时, PWM1\_OUT 输出 0

0: ( PWM1\_EN=1) 当 PWMF <= PWM1 时, PWM1\_OUT 输出 1

Bit 0 PWM0\_INV: PWM0 的输出配置位

1: ( PWM0\_EN=1) 当 PWMF <= PWM0 时, PWM0\_OUT 输出 0

0: ( PWM0\_EN=1) 当 PWMF <= PWM0 时, PWM0\_OUT 输出 1



## 16 SRA 8 位 ADC

### 16.1 概述

对于大多数电子系统而言，处理现实世界的模拟信号是共同的需求。为了完全由单片机来处理这些信号，首先需要通过 A/D 转换器将模拟信号转换成数字信号。将 ADC 电路集成到单片机，可有效的减少外部器件，随之而来，具有降低成本和减少器件空间需求的优势。

本芯片提供了 1 通道的 8bit ADC，当相关 IO 口作为 ADC 通道使用时，该 IO 口必须设置为仅输入模式（高阻）。ADC 能将一个模拟输入信号转换成相应的 8 位数字信号。当转换完成时，MCU 中断。转换周期为 32 个机器周期。

### 16.2 相关特殊功能寄存器

1、SARDATA(0x30)： ADC 转换结果数据寄存器

2、SARCT (0x31)： ADC 转换控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	SAREN	SAR_STA
R/W	—	—	—	—	—	—	R/W	R/W

Bit 7~2 未使用，读为“0”

Bit 1 SAREN: SARADC 使能

0: 关闭 SARADC 模块的电源

1: 打开 SARADC 模块的电源

Bit 0 SAR\_STA: 启动转换位

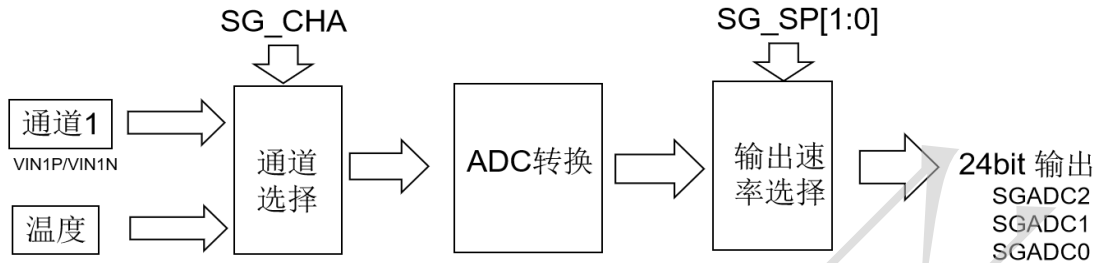
置 1 后，SARADC 开始转换，转换完成后自动清零。



## 17 SIGMA-DELTA 24 位 ADC

### 17.1 概述

#### 结构图



#### Sigma-Delta ADC

高精度模数转换器的核心部分为采用 Sigma-Delta 调制器结构，集成可编程增益 (PGA =2 或 128)。该调制器对输入模拟差分信号的采样频率为 CLK\_OSC (4.9MHz/ 64，远高于模拟信号的最大带宽，因而简化了应用时输入通道的前置防混叠滤波器。

#### 通道选择 (MUX)

高精度模数转换器提供两个通道数据测量通道，通过 SG\_CHA 来控制。其中，通道 1 是普通的可外接的差分输入，通道 2 连接内部的温度传感器。

#### 内部增益 (PGA)

通道 1 增益为 128 倍，温度通道增益为 2 倍。

#### ADC 输出速率选择 (SP)

通过 SG\_SP1、SG\_SP0 可以调节 ADC 的输出速率：10Hz、20Hz、40Hz、80Hz

#### ADC 输出值

参考电压：VREF=VCC

差分输入电压范围：±0.64534 (VREF/Gain)

采用二进制补码输出

输入为正时输出数据范围：000001h ~ 7FFFFFFh

输入为负时输出数据范围：FFFFFFh ~ 800000h

### 17.2 相关特殊功能寄存器

#### 1、SGADCCON: SIGMA 配置寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	SG_CHA	SG_SP1	SG_SP0
R/W	—	—	—	—	—	R/W	R/W	R/W

Bit 7~3 未使用，读为“0”

Bit 2 SG\_CHA: 通道选择

0: 通道 1 (VIN1P/VIN1N-芯片引脚)

1: 温度通道

Bit 1~0 SG\_SP1/ SG\_SP0: 输出速率选择

00: 10Hz

01: 20Hz  
10: 40Hz  
11: 80Hz

## 2、SGADCCT: SIGMA 控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	SG_EN	SG_LOCK	TEMP_EN
R/W	—	—	—	—	—	R/W	R/W	R/W

Bit 7~3 未使用，读为“0”

Bit 2 SG\_EN: 使能

1: 开启

0: 关闭

Bit 1 SG\_LOCK: 24 位 ADC 核心寄存器锁定

1: 锁定寄存器, SGADCCON

0: 解锁

Bit 0 TEMP\_EN: 温度模块使能

1: 开启温度模块

0: 关闭温度模块

3、24 位转换数据: SGADC2[7:0], SGADC1[7:0], SGADC0[7:0]  
可读不可写

## 18 温度传感器

### 18.1 概述

将 24 位 ADC 切换到温度通道，打开温度模块，可以通过 ADC 的转换值方便的测得温度信息。

### 18.2 相关特殊功能寄存器

#### 1、SGADCCON: SIGMA 配置寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	SG_CHA	SG_SP1	SG_SPO
R/W	—	—	—	—	—	R/W	R/W	R/W

Bit 2 SG\_CHA: 通道选择

0: 通道 1 (VIN1P/VIN1N-芯片引脚)

1: 温度通道

#### 2、SGADCCT: SIGMA 控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	SG_EN	SG_LOCK	TEMP_EN
R/W	—	—	—	—	—	R/W	R/W	R/W

Bit 0 TEMP\_EN: 温度模块使能

1: 开启温度模块

0: 关闭温度模块

#### 3、温度传感器的转换数值

注意: 本表格的 ADC 输出值均右移了 2 位后的结果	
参考电压	3.30V
25 度输出值	224700
温度每增加 1 度的输出变化	740

## 19 18x4 的 LCD 驱动器

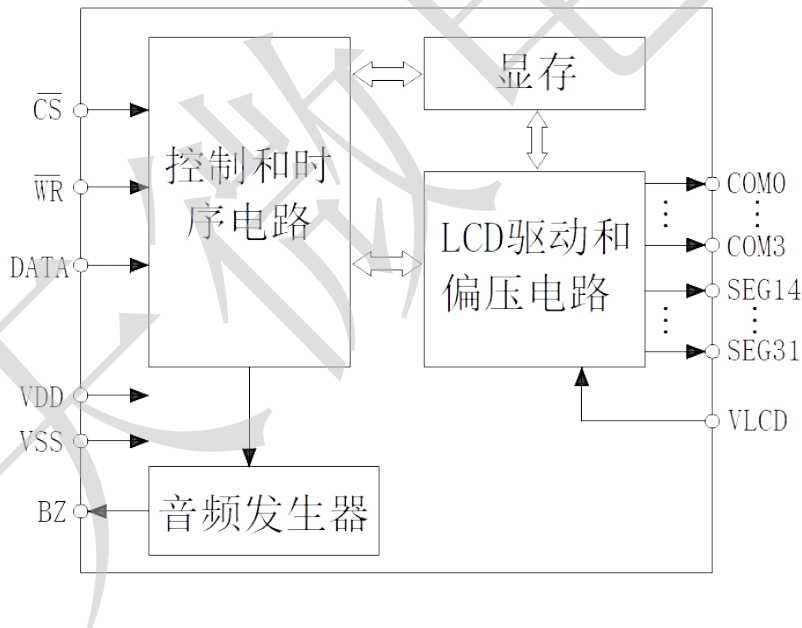
### 19.1 概述

此 18x4 的 LCD 驱动器可由软件配置成 1/2 或 1/3 的 LCD 驱动器偏压和 2、3 或 4 个公共端口，这一特性使其适用于多种 LCD 应用场合。LCD 驱动时钟由系统时钟分频产生，LCD 驱动时钟的频率值保持为 256KHz，由片内 RC 振荡器产生。LCD 驱动器相关命令参见下表。

名称	命令代码	功能描述
LCD OFF	<b>100</b> 00000010X	关闭LCD 输出
LCD ON	<b>100</b> 00000011X	打开LCD 输出
BIAS&COM	<b>100</b> 0010abXcX	c=0: 可选1/2偏压 c=1: 可选1/3偏压 ab=00: 可选2 个公共口 ab=01: 可选3 个公共口 ab=10: 可选4 个公共口

粗体 100 即“**100**”，表示命令模式类型。如果执行连续的命令，除了第一个命令，其它命令的模式类型码将被忽略。LCD OFF 命令使 LCD 偏压发生器失效，从而关闭 LCD 显示；LCD ON 命令使 LCD 偏压发生器有效，从而打开 LCD 显示。BIAS&COM 是 LCD 模块相关命令，可以使其与大多数 LCD 模块相兼容。

### 19.2 LCD 结构图



其中， /CS : 为片选信号线  
/WR、DATA: 串行接口  
BZ: 声音输出  
COM0~COM3, SEG14~SEG31: LCD 输出

## 19.3 结构说明

### 1、显示内存 (RAM)

静态显示内存 (RAM) 以18x4位的格式储存所显示的数据。RAM的数据直接映象到LCD驱动器，可以用WRITE命令访问。

ADDR \ Data	Bit0	Bit1	Bit2	Bit3	
14					SEG14
15					SEG15
16					SEG16
17					SEG17
⋮					⋮
31					SEG31

COM0      COM1      COM2      COM3

RAM 映象图

### 2、系统振荡器

系统振荡器用于产生LCD驱动时钟和声音频率。片内RC振荡器 (256KHz) 可以产生时钟源。系统振荡器配置图参见下图。执行SYS DIS命令可以停止系统时钟和LCD偏压发生器工作，SYS DIS命令只适用于片内RC振荡器或晶振，当系统时钟停止工作时，LCD将显示空白。



系统振荡器配置图

LCD OFF命令用于关闭LCD 偏压发生器，当LCD 偏压发生器关闭后，可用SYS DIS命令降低系统功耗，这时SYS DIS 为节电命令。系统开始上电时，处于SYS DIS状态。

### 3、声音输出

产品内嵌一个简单的声音发生器，可以在管脚BZ上输出驱动信号用于产生一个单音。执行TONE 4K或TONE 2K命令可以输出两种频率的声音 (4KHz和2KHz)。TONE ON或TONE OFF命令用于打开或关闭声音输出。管脚BZ是声音输出管脚，用于驱动蜂鸣器。当系统失效或禁止声音输出时，管脚BZ将保持低电平的输出。

### 4、接口

该驱动器只有三根管脚用于接口。管脚/CS用于初始化串行接口电路和结束主控制器与驱动器之间的通讯。管脚/CS设置为“1”时，主控制器和驱动器之间的数据和命令无效并初始化。在产生命令模式或模式转换之前，必须用一个高电平脉冲初始化驱动器的串行接口。管脚DATA是串行数据输入管脚，写数据和写命令通过管脚DATA进行。管脚/WR是写时钟输入管脚，在/WR信号上升沿时，管脚DATA上的数据、地址和命令被写入驱动器。

## 19.4 命令格式

该驱动器可以用软件设置。两种模式的命令可以配置此LCD驱动器和传送到LCD所显示的数据。该驱动器的配置模式称为命令模式，类型码为100。命令模式包括一个系统配置命令，一个系统频率选择命令，一个LCD配置命令，一个声音频率选择命令和一个操作命令。数据模式有WRITE操作，下表是数据

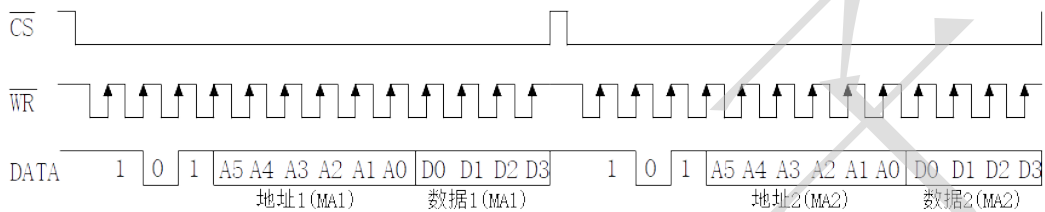
和命令模式类型码表。

操作	模式	类型码
WRITE	数据	101
COMMAND	命令	100

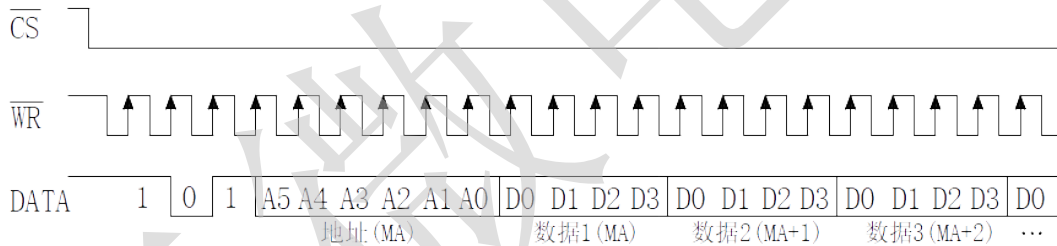
模式命令应在数据或命令传送前运行，如果执行连续的命令，命令模式代码即**100**，将被忽略。当系统在不连续命令模式或不连续地址数据模式下，管脚/CS应设为“1”，而且先前的操作模式将复位。当管脚/CS返回“0”时，新的操作模式类型码应先运行。

## 19.5 时序图

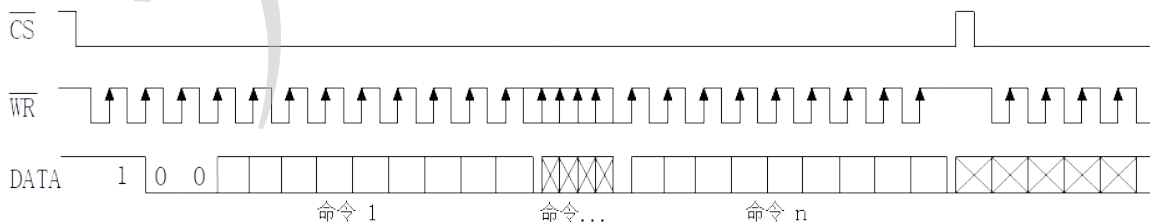
WRITE模式（命令代码101）

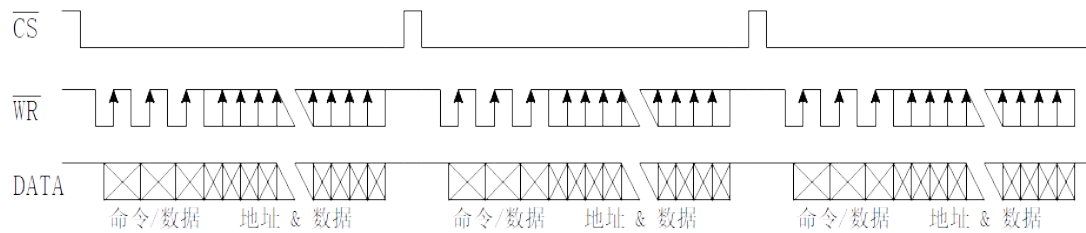


WRITE模式（写连续地址）



命令模式（命令代码100）





## 19.6 命令概述

命令名称	命令代码	D/C	功能描述	上电时复位缺省
WRITE	<b>101</b> a5a4a3a2a1a0d0d1d2d3	D	写数据到RAM	
SYS DIS	<b>100</b> 00000000X	C	关闭系统振荡器和LCD 偏压发生器	Y
SYS EN	<b>100</b> 00000001X	C	打开系统振荡器	
LCD OFF	<b>100</b> 00000010X	C	关闭LCD 偏压发生器	Y
LCD ON	<b>100</b> 00000011X	C	打开LCD 偏压发生器	
TONE OFF	<b>100</b> 00001000X	C	关闭声音输出	Y
TONE ON	<b>100</b> 00001001X	C	打开声音输出	
RC 256K	<b>100</b> 000110XXX	C	系统时钟源片内RC振荡器	Y
BIAS1/2	<b>100</b> 0010abX0X	C	LCD 1/2偏压选项 ab=00:2 个公共口 ab=01:3 个公共口 ab=10:4 个公共口	
BIAS 1/3	<b>100</b> 0010abX1X	C	LCD 1/3偏压选项 ab=00:2 个公共口 ab=01:3 个公共口 ab=10:4 个公共口	
TONE 4K	<b>100</b> 010XXXXXX	C	声音频率4KHz	
TONE2K	<b>100</b> 011XXXXXX	C	声音频率2KHz	
TOPT	<b>100</b> 11100000X	C	测试模式	
TNORMAL	<b>100</b> 11100011X	C	普通模式	Y

注： X: 0或1; a5~a0: RAM 地址; d3~d0: RAM 数据; D/C: 数据/命令模式

所有的粗体数字即**110**, **101**和**100** 为模式命令码。**100**为命令模式类型码, 如果运行连续的命令, 除了第一个命令, 其它命令的模式类型码将被忽略。声音频率源由片内256KHz RC 振荡器产生, 频率的计算情况如前文所述。建议在上电复位后, 用主控制器初始化该驱动器, 因为如果上电复位失败, 驱动器将不能正常工作。

## 20 指令集(总计 61 条)

### 指令描述约定

m 数据寄存器地址

R m 对应数据存储寄存器

ACC 累加器(工作寄存器)

b 八位数据寄存器内的位地址

k 立即数

### 指令一览表

助记符	描述	周期	影响标注位	
算术运算				
1	ADDAM m, 1	R 与 ACC 相加, 结果存入 ACC	1	Z, C, AC, OV
2	ADDAM m, 0	R 与 ACC 相加, 结果存入 R	1	Z, C, AC, OV
3	ADDAL k	立即数与 ACC 相加, 结果存入 ACC	1	Z, C, AC, OV
4	ADDAMC m, 1	R 与 ACC、进位标志相加, 结果存入 ACC	1	Z, C, AC, OV
5	ADDAMC m, 0	R 与 ACC、进位标志相加, 结果存入 R	1	Z, C, AC, OV
6	SUBAM m, 1	R 与 ACC 相减, 结果存入 ACC	1	Z, C, AC, OV
7	SUBAM m, 0	R 与 ACC 相减, 结果存入 R	1	Z, C, AC, OV
8	SUBAL k	立即数与 ACC 相减, 结果存入 ACC	1	Z, C, AC, OV
9	SUBAMC m, 1	R 与 ACC、进位标志的反相减, 结果存入 ACC	1	Z, C, AC, OV
10	SUBAMC m, 0	R 与 ACC、进位标志的反相减, 结果存入 R	1	Z, C, AC, OV
11	BTD m	将累加器中的数据调整为十进制数, 结果存入 R	1	C
逻辑运算				
1	ANDAM m, 1	R 与 ACC 做“与”运算, 结果存入 ACC	1	Z
2	ORAM m, 1	R 与 ACC 做“或”运算, 结果存入 ACC	1	Z
3	XORAM m, 1	R 与 ACC 做“异或”运算, 结果存入 ACC	1	Z
4	ANDAM m, 0	R 与 ACC 做“与”运算, 结果存入 R	1	Z
5	ORAM m, 0	R 与 ACC 做“或”运算, 结果存入 R	1	Z
6	XORAM m, 0	R 与 ACC 做“异或”运算, 结果存入 R	1	Z
7	ANDAL k	立即数与 ACC 做“与”运算, 结果存入 ACC	1	Z
8	ORAL k	立即数与 ACC 做“或”运算, 结果存入 ACC	1	Z
9	XORAL k	立即数与 ACC 做“异或”运算, 结果存入 ACC	1	Z
10	CPL m, 1	R 取反, 结果存入 ACC	1	Z
11	CPL m, 0	R 取反, 结果存入 R	1	Z
递增和递减				
1	INC m, 1	R 递增, 结果存入 ACC	1	Z
2	INC m, 0	R 递增, 结果存入 R	1	Z
3	DEC m, 1	R 递减, 结果存入 ACC	1	Z
4	DEC m, 0	R 递减, 结果存入 R	1	Z
移位				
1	RR m, 1	R 右移一位, 结果存入 ACC	1	无
2	RR m, 0	R 右移一位, 结果存入 R	1	无
3	RRC m, 1	R 带进位右移一位, 结果存入 ACC	1	C
4	RRC m, 0	R 带进位右移一位, 结果存入 R	1	C
5	RL m, 1	R 左移一位, 结果存入 ACC	1	无
6	RL m, 0	R 左移一位, 结果存入 R	1	无
7	RLC m, 1	R 带进位左移一位, 结果存入 ACC	1	C
8	RLC m, 0	R 带进位左移一位, 结果存入 R	1	C
数据传送				
1	MOVMA m	将 R 送至 ACC	1	无
2	MOVAM m	将 ACC 送至 R	1	无
3	MOVLA k	将立即数送至 ACC	1	无
位运算				



1	CLRB m, b	清零 R 的第 b 位	1	无
2	SETB m, b	置位 R 的第 b 位	1	无
条件转移				
1	SZ m, 0	如果 R 为零, 则跳过下一条指令	1/2	无
2	SZ m, 1	将 R 送至 ACC, 若内容为零, 则跳过下一条指令	1/2	无
3	SZB m, b	如果 R 的第 b 位为零, 则跳过下一条指令	1/2	无
4	SZBN m, b	如果 R 的第 b 位不为零, 则跳过下一条指令	1/2	无
5	SZINC m, 0	R 递增, 若结果为零, 则跳过下一条指令	1/2	无
6	SZDEC m, 0	R 递减, 若结果为零, 则跳过下一条指令	1/2	无
7	SZINC m, 1	R 递增, 结果存入 ACC, 若结果为零, 则跳过下一条指令	1/2	无
8	SZDEC m, 1	R 递减, 结果存入 ACC, 若结果为零, 则跳过下一条指令	1/2	无
无条件转移				
1	JMP k	无条件跳转	2	无
2	GOTO k	调用子程序	2	无
3	RET	从子程序返回	2	无
4	RETL k	从子程序返回, 并将立即数存入 ACC	2	无
5	RETIN	从中断返回	2	无
其他指令				
1	NOP	空指令	1	无
2	CLR A	清除 ACC	1	无
3	CLR m	清除 R	1	无
4	SET m	置位 R	1	无
5	CLRWDT	清零看门狗定时器	1	TO, PDF
6	CLRWDT1	预清零看门狗定时器	1	无
7	SWOP m, 0	R 的高低字节互换, 结果存入 R	1	无
8	SWOP m, 1	R 的高低字节互换, 结果存入 ACC	1	无
9	STOP	进入暂停模式	1	TO, PDF

## 指令定义惯例

k: 立即数

m: 数据存储器地址

ACC: 累加器

b: 第 0~7 位

### 一、算术运算

#### 1、ADDAM m, 1

指令说明 将指定的数据存储器内容和累加器内容相加, 结果存放到累加器。

功能表示  $ACC \leftarrow m + ACC$

影响标志位 OV、Z、AC、C

#### 2、ADDAM m, 0

指令说明 将指定的数据存储器内容和累加器内容相加, 结果存放到指定的数据存储器。

功能表示  $m \leftarrow m + ACC$

影响标志位 OV、Z、AC、C

#### 3、ADDAL k

指令说明 将累加器和立即数相加, 结果存放到累加器。

功能表示  $ACC \leftarrow k + ACC$

影响标志位 OV、Z、AC、C

**4、ADDAMC m, 1**

指令说明 将指定的数据存储器、累加器内容以及进位标志相加，结果存放到累加器。

功能表示  $ACC \leftarrow m + ACC + C$

影响标志位 OV、Z、AC、C

**5、ADDAMC m, 0**

指令说明 将指定的数据存储器、累加器内容和进位标志位相加，结果存放到指定的数据存储器。

功能表示  $m \leftarrow m + ACC + C$

影响标志位 OV、Z、AC、C

**6、SUBAL k**

指令说明 将立即数去累加器的内容减，结果存放到累加器。如果结果为负，C 标志位清除为0，反之结果为正或0，C 标志位设置为1。

功能表示  $ACC \leftarrow k - ACC$

影响标志位 OV、Z、AC、C

**7、SUBAM m, 1**

指令说明 将指定数据存储器的数据减去累加器的内容，把结果存放到累加器。如果结果为负，C 标志位清除为0，反之结果为正或0，C 标志位设置为1。

功能表示  $ACC \leftarrow m - ACC$

影响标志位 OV、Z、AC、C

**8、SUBAM m, 0**

指令说明 将指定数据存储器的数据减去累加器的内容，结果存放到指定的数据存储器。如果结果为负，C 标志位清除为0，反之结果为正或0，C 标志位设置为1。

功能表示  $m \leftarrow m - ACC$

影响标志位 OV、Z、AC、C

**9、SUBAMC m, 1**

指令说明 将指定数据存储器的数据减去累加器的内容减去进位标志的反，结果存放到累加器。如果结果为负，C 标志位清除为0，反之结果为正或0，C 标志位设置为1。

功能表示  $ACC \leftarrow m - ACC - (!C)$

**10、SUBAMC m, 0**

指令说明 将指定数据存储器的数据减去累加器的内容减去进位标志的反，结果存放到数据存储器。如果结果为负，C 标志位清除为0，反之结果为正或0，C 标志位设置为1。

功能表示  $m \leftarrow m - ACC - (!C)$

影响标志位 OV、Z、AC、C

**11、BTD m**

指令说明 一般与加法一起使用，作用是把上一次的加法中的加数和被加数直接看成十进制数进行运算，把十进制的和存到 m 中。如果低四位的值大于“9”或 AC=1，那么 BCD 调整就执行对原值加“6”，否则原值保持不变；如果高四位的值大于“9”或 C=1，那么 BCD 调整就执行对原值加“6”。BCD 转换实质上是根据累加器和标志位执行 00H, 06H, 60H 或 66H 的加法运算，结果存放到数据存储器。只有进位标志位 C 受影响，用来指示原始 BCD 的和是否大于 100，并可以进行双精度十进制数的加法运算。

功能表示  $m \leftarrow ACC + 00H$  或

$m \leftarrow ACC + 06H$  或

$m \leftarrow ACC + 60H$  或

$m \leftarrow ACC + 66H$

影响标志位 C

例如上一次加法中,

29H+35H=5EH

直接看成

29D+35D=64D → m

## 二、逻辑运算

### 1、ANDAM m, 1

指令说明 将指定数据存储器内容和累加器中的数据做逻辑与, 结果存放到累加器。

功能表示  $ACC \leftarrow m \text{ "AND" } ACC$

影响标志位 Z

### 2、ANDAM m, 0

指令说明 将指定数据存储器内容和累加器中的数据做逻辑与, 结果存放到数据存储器。

功能表示  $m \leftarrow m \text{ "AND" } ACC$

影响标志位 Z

### 3、ANDAL k

指令说明 将累加器中的数据和立即数做逻辑与, 结果存放到累加器。

功能表示  $ACC \leftarrow k \text{ "AND" } ACC$

影响标志位 Z

### 4、ORAM m, 1

指令说明 将指定数据存储器中的数据和累加器逻辑或, 结果存放到累加器。

功能表示  $ACC \leftarrow m \text{ "OR" } ACC$

影响标志位 Z

### 5、ORAM m, 0

指令说明 将指定数据存储器中的数据和累加器逻辑或, 结果放到数据存储器。

功能表示  $m \leftarrow m \text{ "OR" } ACC$

影响标志位 Z

### 6、ORAL k

指令说明 将累加器中的数据和立即数逻辑或, 结果存放到累加器。

功能表示  $ACC \leftarrow k \text{ "OR" } ACC$

影响标志位 Z

### 7、XORAM m, 1

指令说明 将指定的数据存储器内容和累加器的数据逻辑异或, 结果存放到累加器。

功能表示  $ACC \leftarrow m \text{ "XOR" } ACC$

影响标志位 Z

### 8、XORAM m, 0

指令说明 将指定的数据存储器内容和累加器的数据逻辑异或, 结果放到数据存储器。

功能表示  $m \leftarrow m \text{ "XOR" } ACC$

影响标志位 Z

### 9、XORAL k

指令说明 将累加器的数据与立即数逻辑异或, 结果存放到累加器。

功能表示  $ACC \leftarrow k \text{ "XOR" } ACC$

影响标志位 Z

## 10、CPL m, 0

指令说明 将指定数据存储器中的每一位取逻辑反，相当于从 1 变 0 或 0 变 1。

功能表示  $m \leftarrow \sim m$

影响标志位 Z

## 11、CPL m, 1

指令说明 将指定数据存储器中的每一位取逻辑反，相当于从 1 变 0 或 0 变 1，而结果被储存回累加器且数据存储器中的内容不变。

功能表示  $ACC \leftarrow \sim m$

影响标志位 Z

## 三、递增和递减

## 1、INC m, 0

指令说明 将指定数据存储器的内容加 1。

功能表示  $m \leftarrow m + 1$

影响标志位 Z

## 2、INC m, 1

指令说明 将指定数据存储器的内容加 1，结果存放回累加器并保持指定的数据存储器内容不变。

功能表示  $ACC \leftarrow m + 1$

影响标志位 Z

## 3、DEC m, 0

指令说明 将指定数据存储器内容减 1。

功能表示  $m \leftarrow m - 1$

影响标志位 Z

## 4、DEC m, 1

指令说明 将指定数据存储器的内容减 1，把结果存放回累加器并保持指定数据存储器内容不变。

功能表示  $ACC \leftarrow m - 1$

影响标志位 Z

## 四、移位

## 1、RR m, 0

指令说明 将指定数据存储器的内容循环右移 1 位且第 0 位移到第 7 位。

功能表示  $m.b \leftarrow m.(b+1) (b=0\sim 6)$

$m.7 \leftarrow m.0$

影响标志位 无

## 2、RR m, 1

指令说明 将指定数据存储器的内容循环右移 1 位，第 0 位移到第 7 位，移位结果存放到累加器，而指定数据存储器的内容保持不变。

功能表示  $ACC.b \leftarrow m.(b+1) (b=0\sim 6)$

$ACC.7 \leftarrow m.0$

影响标志位 无

## 3、RRC m, 0

指令说明 将指定数据存储器的内容连同进位标志右移 1 位，第 0 位取代进位标志且原本的进位标志移到第 7 位。

功能表示  $m.b \leftarrow m.(b+1) (b=0\sim6)$   
 $m.7 \leftarrow C$   
 $C \leftarrow m.0$   
影响标志位 C

#### 4、RRC m, 1

指令说明 将指定数据存储器的内容连同进位标志右移 1 位，第 0 位取代进位标志且原本的进位标志移到第 7 位，移位结果送回累加器，但是指定数据寄存器的内容保持不变。

功能表示  $ACC.b \leftarrow m.(b+1) (b=0\sim6)$   
 $ACC.7 \leftarrow C$   
 $C \leftarrow m.0$   
影响标志位 C

#### 5、RL m, 0

指令说明 将指定数据存储器的内容左移 1 位，且第 7 位移到第 0 位。

功能表示  $m.(b+1) \leftarrow m.b (b=0\sim6)$   
 $m.0 \leftarrow m.7$   
影响标志位 无

#### 6、RL m, 1

指令说明 将指定数据存储器的内容左移 1 位，且第 7 位移到第 0 位，结果送到累加器，而指定数据存储器的内容保持不变。

功能表示  $ACC.(b+1) \leftarrow m.b (b=0\sim6)$   
 $ACC.0 \leftarrow m.7$   
影响标志位 无

#### 7、RLC m, 0

指令说明 将指定数据存储器的内容连同进位标志左移 1 位，第 7 位取代进位标志且原本的进位标志移到第 0 位。

功能表示  $m.(b+1) \leftarrow m.b (b=0\sim6)$   
 $m.0 \leftarrow C$   
 $C \leftarrow m.7$   
影响标志位 C

#### 8、RLC m, 1

指令说明 将指定数据存储器的内容连同进位标志左移 1 位，第 7 位取代进位标志且原本的进位标志移到第 0 位，移位结果送回累加器，但是指定数据寄存器的内容保持不变。

功能表示  $ACC.(b+1) \leftarrow m.b (b=0\sim6)$   
 $ACC.0 \leftarrow C$   
 $C \leftarrow m.7$   
影响标志位 C

### 五、数据传送

#### 1、MOVMA m

指令说明 将指定数据存储器的内容复制到累加器。

功能表示  $ACC \leftarrow m$   
影响标志位 无

#### 2、MOVLA k

指令说明 将 8 位立即数载入累加器。

功能表示  $ACC \leftarrow k$   
影响标志位 无

### 3、MOVAM m

指令说明 将累加器的内容复制到指定的数据存储器。

功能表示  $m \leftarrow ACC$

影响标志位 无

## 六、位运算

### 1、CLRB m, b

指令说明 将指定数据存储器的 b 位内容清零。

功能表示  $m, b \leftarrow 0$

影响标志位 无

### 2、SETB m, b

指令说明 将指定数据存储器的第 b 位置位为 1。

功能表示  $m, b \leftarrow 1$

影响标志位 无

## 七、条件转移

### 1、SZ m, 0

指令说明 判断指定数据存储器的内容是否为 0，若为 0，则程序跳过下一条指令执行。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。

功能表示 如果  $m=0$ ，跳过下一条指令执行

影响标志位 无

### 2、SZ m, 1

指令说明 将指定数据存储器内容复制到累加器，并判断指定数据存储器的内容是否为 0，若为 0 则跳过下一条指令。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。

功能表示  $ACC \leftarrow m$ ，如果  $m=0$ ，跳过下一条指令执行

影响标志位 无

### 3、SZB m, b

指令说明 判断指定数据存储器的第 b 位是否为 0，若为 0，则跳过下一条指令。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。

功能表示 如果  $m, b=0$ ，跳过下一条指令执行

影响标志位 无

### 4、SZBN m, b

指令说明 判断指定数据存储器的第 i 位，若不为 0，则程序跳过下一条指令执行。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果为 0，则程序继续执行下一条指令。

功能表示 如果  $m, i \neq 0$ ，跳过下一条指令执行

影响标志位 无

指令说明 判断指定数据存储器的第 b 位是否为 0，若为 0，则跳过下一条指令。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。

功能表示 如果  $m, b = 0$ ，跳过下一条指令执行

影响标志位 无



## 5、SZINC m, 0

指令说明 将指定的数据存储器的内容加1，判断是否为0，若为0则跳过下一条指令。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为2个周期的指令。如果结果不为0，则程序继续执行下一条指令。

功能表示  $m \leftarrow m+1$ ，如果  $m=0$  跳过下一条指令执行

影响标志位 无

## 6、SZDEC m, 0

指令说明 将指定的数据存储器的内容减1，判断是否为0，若为0则跳过下一条指令，由于取得下一个指令时会要求插入一个空指令周期，所以此指令为2个周期的指令。如果结果不为0，则程序继续执行下一条指令。

功能表示  $m \leftarrow m-1$ ，如果  $m=0$  跳过下一条指令执行

影响标志位 无

## 7、SZINC m, 1

指令说明 将指定数据存储器的内容加1，判断是否为0，如果为0则跳过下一条指令，此结果会被存放到累加器，但是指定数据存储器的内容不变。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为2个周期的指令。如果结果不为0，则程序继续执行下一条指令。

功能表示  $ACC \leftarrow m+1$ ，如果  $ACC=0$  跳过下一条指令执行

影响标志位 无

## 8、SZDEC m, 1

指令说明 将指定数据存储器内容减1，判断是否为0，如果为0则跳过下一条指令，此结果将存放到累加器，但指定数据存储器内容不变。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为2个周期的指令。如果结果不为0，则程序继续执行下一条指令。

功能表示  $ACC \leftarrow m-1$ ，如果  $ACC=0$  跳过下一条指令执行

影响标志位 无

## 八、无条件转移

## 1、JMP k (k —12bit)

指令说明 程序计数器的内容无条件地由被指定的地址取代，程序由新的地址继续执行。当新的地址被加载时，必须插入一个空指令周期，所以此指令为2个周期的指令。

功能表示  $Program\ Counter \leftarrow k$

影响标志位 无

## 2、GOTO k

指令说明 无条件地调用指定地址的子程序，此时程序计数器先加1获得下一个要执行的指令地址并压入堆栈，接着载入指定地址并从新地址继续执行程序，由于此指令需要额外的运算，所以为一个2周期的指令。

功能表示  $Stack \leftarrow Program\ Counter + 1$

$Program\ Counter \leftarrow k$

影响标志位 无

## 3、RET

指令说明 将堆栈寄存器中的程序计数器值恢复，程序由取回的地址继续执行。

功能表示  $Program\ Counter \leftarrow Stack$

影响标志位 无

## 4、RETL k

指令说明 将堆栈寄存器中的程序计数器值恢复且累加器载入指定的立即数，程序由取回的地址继续执行。

功能表示 Program Counter  $\leftarrow$  Stack

ACC  $\leftarrow$  k

影响标志位 无

## 5、RETIN

指令说明 将堆栈寄存器中的程序计数器值恢复且中断功能通过设置 GIE 位重新使能。GIE 是控制中断使能的主控制位。如果在执行 RETIN 指令之前还有中断未被相应，则这个中断将在返回主程序之前被相应。

功能表示 Program Counter  $\leftarrow$  Stack

EMI  $\leftarrow$  1

影响标志位 无

## 九、其他指令

## 1、NOP

指令说明 空操作，接下来顺序执行下一条指令。

功能表示 PC  $\leftarrow$  PC+1

影响标志位 无

## 2、CLR A

指令说明 将工作寄存器 ACC 的内容清零。

功能表示 m  $\leftarrow$  00H

影响标志位 无

## 3、CLR m

指令说明 将指定数据存储器的内容清零。

功能表示 m  $\leftarrow$  00H

影响标志位 无

## 4、SET m

指令说明 将指定数据存储器的每一位设置为 1。

功能表示 m  $\leftarrow$  FFH

影响标志位 无

## 5、CLRWDT

指令说明 WDT 计数器、暂停标志位 PDF 和看门狗溢出标志位 TO 清零。

功能表示 WDT  $\leftarrow$  00H

TO & PDF  $\leftarrow$  0

影响标志位 TO、PDF

## 6、CLRWDT1

指令说明 清零 WDT 计时器，PDF 与 TO 保留原状态不变。

功能表示 WDT  $\leftarrow$  00H

## 7、SWOP m, 0

指令说明 将指定数据存储器的低 4 位和高 4 位互相交换。

功能表示 m. 3 $\sim$ m. 0  $\leftrightarrow$  m. 7 $\sim$ m. 4

影响标志位 无



8、SWOP m, 1

指令说明 将指定数据存储器的低 4 位与高 4 位互相交换，再将结果存放到累加器且指定数据寄存器的数据保持不变。

功能表示  $ACC.3 \sim ACC.0 \leftarrow m.7 \sim m.4$

$ACC.7 \sim ACC.4 \leftarrow m.3 \sim m.0$

影响标志位 无

9、STOP

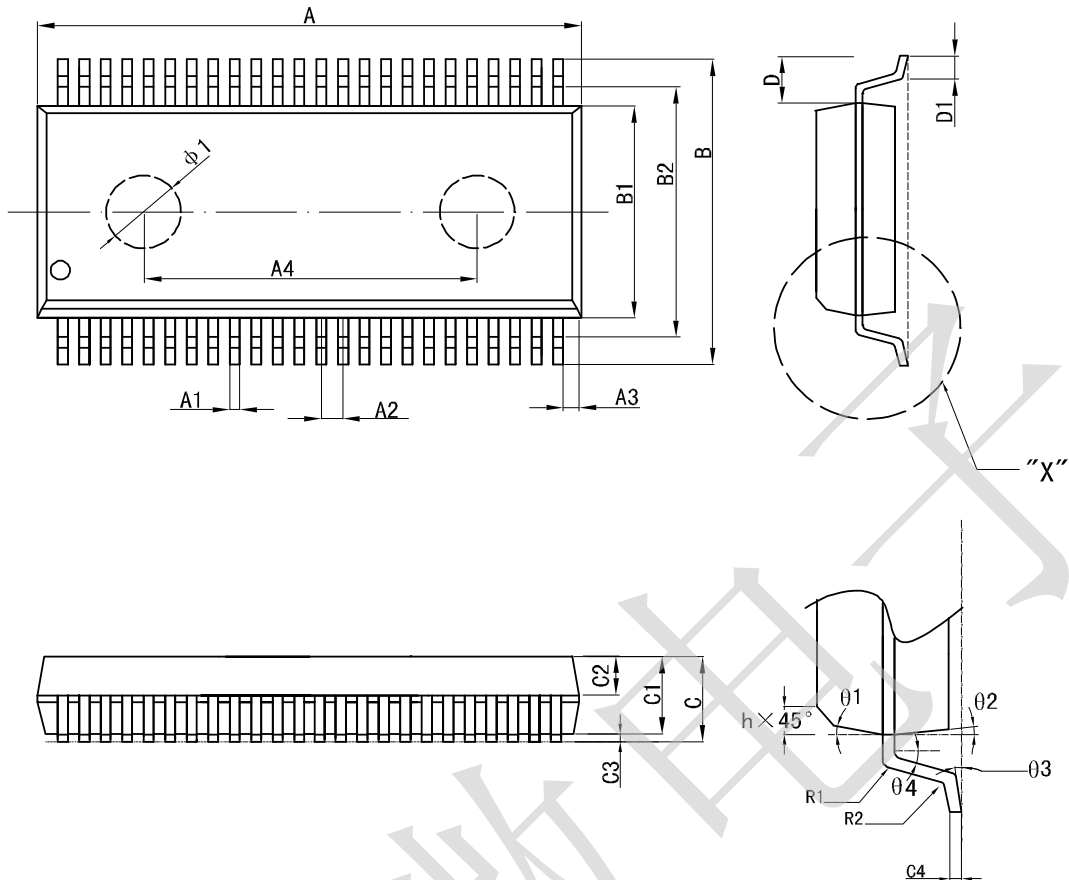
指令说明 此指令终止程序执行并关掉系统时钟，RAM 和寄存器的内容保持原状态，WDT 计数器和分频器被清“0”，暂停标志位 PDF 被置位 1，WDT 溢出标志位 TO 被清 0。

功能表示  $TO \leftarrow 0$

$PDF \leftarrow 1$

影响标志位 TO、PDF

## 21 封装 (SSOP48)



尺寸 标注	最小 (mm)	最大 (mm)	尺寸 标注	最小 (mm)	最大 (mm)
A	15.77	15.97	C3	0.2	0.4
A1	0.20	0.35	C4	0.12	0.25
A2	0.635TYP		D	1.41TYP	
A3	0.5TYP		D1	0.61	0.91
A4	10.2TYP		h	0.381	0.635
B	10.01	10.61	phi1	2.2TYP	
B1	7.39	7.59	theta1	15° TYP	
B2	8.6TYP		theta2	15° TYP	
C	2.41	2.78	theta3	4° TYP	
C1	2.18	2.38	theta4	8° TYP	
C2	1.067TYP				

All specs and applications shown above subject to change without prior notice.  
(以上电路及规格仅供参考, 如本公司进行修正, 恕不另行通知。)