

SinoMCU 8 位单片机

MC32F7342

用户手册

V0.0



上海晟矽微电子股份有限公司

Shanghai SinoMCU Microelectronics Co., Ltd.

目录

1	产品概要	4
1.1	产品特性	4
1.2	订购信息	5
1.3	引脚排列	5
1.4	端口说明	7
2	电气特性	8
2.1	极限参数	8
2.2	直流电气特性	8
2.3	ADC 特性参数	9
2.4	EERPOM 特性参数	9
2.5	交流电气特性	10
3	CPU 及存储器	11
3.1	指令集	11
3.2	程序存储器	13
3.3	数据存储器	14
3.4	在线编程	15
3.5	堆栈	16
3.6	控制寄存器	16
3.7	用户配置字	19
4	系统时钟	20
4.1	内部高频 RC 振荡器	20
4.2	内部低频 RC 振荡器	20
4.3	工作模式	21
4.4	低功耗模式	22
5	复位	23
5.1	复位条件	23
5.2	上电复位	23
5.3	外部复位	24
5.4	低电压复位	24
5.5	看门狗复位	24
6	I/O 端口	25
6.1	I/O 工作模式	25
6.2	内部上/下拉电阻	26
6.3	端口模式控制	27
7	定时器 TIMER	29
7.1	看门狗定时器 WDT	29
7.2	定时器 T0	29
7.3	定时器 T1	31
7.4	定时器 T2	34
7.5	定时器 T3	36
8	模数转换器 ADC	38
8.1	ADC 概述	38

8.2	ADC 操作步骤.....	38
8.3	ADC 相关寄存器.....	39
8.4	ADC 零点偏移修调流程.....	42
9	EEPROM 数据存储器.....	44
9.1	EEPROM 概述.....	44
9.2	EEPROM 相关寄存器.....	44
9.3	EEPROM 操作示例.....	45
10	中断.....	47
10.1	外部中断.....	47
10.2	定时器中断.....	47
10.3	ADC 中断.....	47
10.4	键盘中断.....	48
10.5	中断相关寄存器.....	48
11	特性曲线.....	51
11.1	I/O 特性.....	51
11.2	功耗特性.....	51
11.3	模拟电路特性.....	51
12	封装尺寸.....	52
12.1	SOP24.....	52
12.2	SOP20.....	52
12.3	DIP20.....	52
12.4	SOP16.....	52
12.5	DIP16.....	52
12.6	TSSOP20.....	52
13	修订记录.....	53

1 产品概要

1.1 产品特性

- 8 位 CPU 内核
 - ◇ 精简指令集, 8 级深度硬件堆栈
 - ◇ CPU 为双时钟, 可在系统高/低频时钟之间切换
 - ◇ 高频时钟下 F_{CPU} 可配置为 2T/4T/8T/16T/32T/64T, 低频时钟下 F_{CPU} 固定为 2T
- 程序存储器
 - ◇ 4K×16 位 FLASH 型程序存储器, 可通过间接寻址读取程序存储器内容
 - ◇ 支持在线烧录, 擦写次数至少 1000 次
- 数据存储器
 - ◇ 384 字节 SRAM 通用数据存储器, 支持直接寻址、间接寻址等多种寻址方式
 - ◇ 64 字节 EEPROM 型数据存储器, 支持单独烧录和软件读写, 擦写次数至少 10000 次
- 3 组共 22 个 I/O
 - ◇ P0 (P00~P07), P1 (P10~P17), P2 (P20~P25)
 - ◇ 所有端口均内置上拉和下拉电阻, 均可单独使能/禁用
 - ◇ P1 支持键盘中断唤醒功能
- 系统时钟
 - ◇ 内置高频 RC 振荡器 (16MHz), 可用作系统高频时钟源
 - ◇ 内置低频 RC 振荡器 (32KHz), 可用作系统低频时钟源
- 多种系统工作模式
 - ◇ 高速模式: CPU 在高频时钟下运行, 低频时钟源工作
 - ◇ 低速模式: CPU 在低频时钟下运行, 高频时钟源可选停止或工作
 - ◇ HOLD1 模式: CPU 停止运行, 高频时钟源工作
 - ◇ HOLD2 模式: CPU 停止运行, 高频时钟源停止工作, 低频时钟源工作
 - ◇ 休眠模式: CPU 停止运行, 所有时钟源停止工作
- 内部自振式看门狗计数器 (WDT)
 - ◇ 溢出时间可配置: 64ms/2048ms
 - ◇ 工作模式可配置: 始终开启、始终关闭、低功耗模式下停止
- 4 个定时器
 - ◇ 8 位定时器 T0, 可实现外部计数功能、BUZ 功能、PWM 功能
 - ◇ 16 位定时器 T1, 可实现外部计数、16 位 PWM 功能
 - ◇ 16 位定时器 T2
 - ◇ 8 位定时器 T3
- 1 个 12 位高精度 ADC
 - ◇ 14 路外部通道: AN0~AN13; 2 路内部通道: GND、VDD/4
 - ◇ 参考电压可选: VDD、内部参考电压 V_{IR} (2V)
 - ◇ ADC 时钟: F_{HIRC} 的 8/16/32/64 分频
 - ◇ 支持零点校准
- 中断
 - ◇ 外部中断 (INT0~INT3), 键盘中断 (P10~P17)
 - ◇ 定时器中断 (T0~T3), ADC 中断

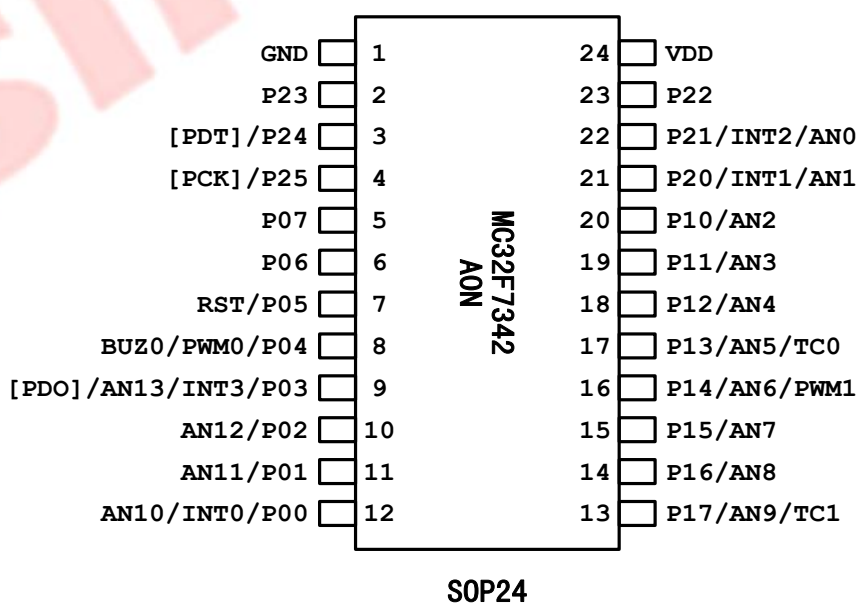
- 低电压复位 LVR: 2.0V/2.4V/2.8V/3.6V
- 工作电压
 - ◇ $V_{LVR28} \sim 5.5V @ F_{cpu} = 0 \sim 8MHz$
 - ◇ $V_{LVR24} \sim 5.5V @ F_{cpu} = 0 \sim 4MHz$
 - ◇ $V_{LVR20} \sim 5.5V @ F_{cpu} = 0 \sim 1MHz$
- 封装形式
 - ◇ SOP24/SOP20/DIP20/TSSOP20/SOP16/DIP16

1.2 订购信息

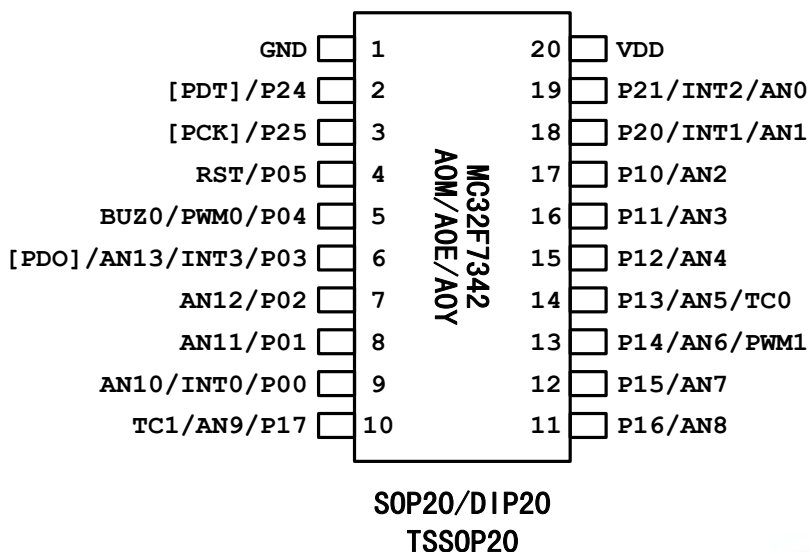
产品名称	封装形式	备注
MC32F7342A0N	SOP24	
MC32F7342A0M	SOP20	
MC32F7342A0E	DIP20	
MC32F7342A0Y	TSSOP20	
MC32F7342A1M	SOP20	
MC32F7342A1E	DIP20	
MC32F7342A0K	SOP16	
MC32F7342A0C	DIP16	

1.3 引脚排列

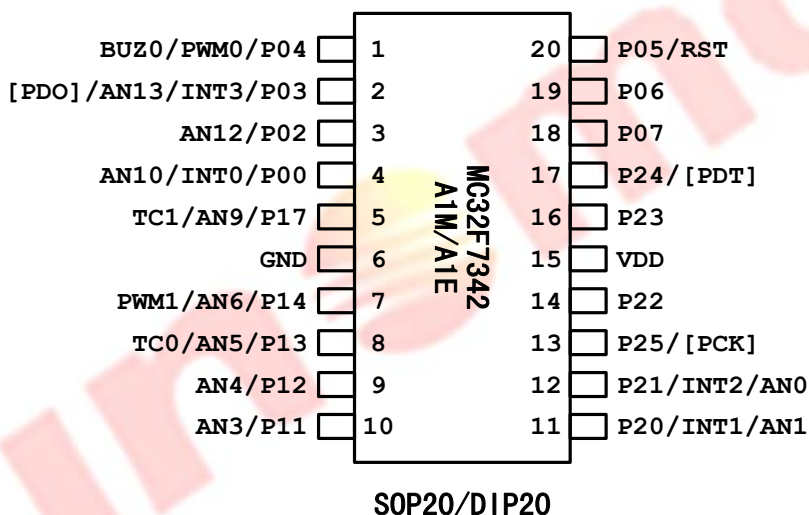
MC32F7342A0N



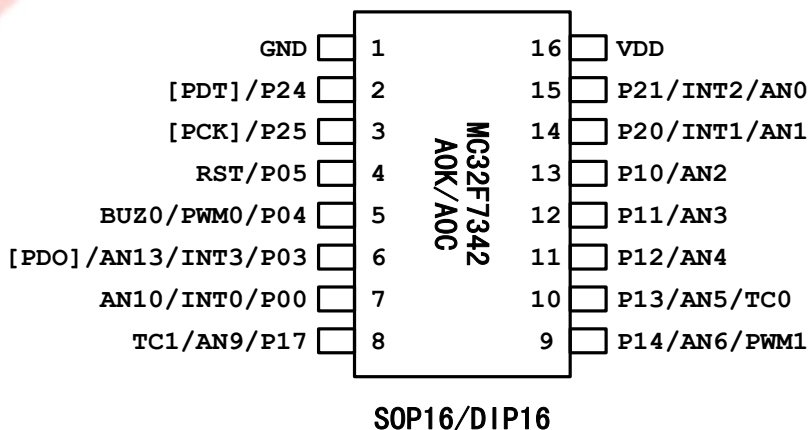
MC32F7342A0M/A0E/A0Y



MC32F7342A1M/A1E



MC32F7342A0K/A0C



1.4 端口说明

端口名称	类型	功能说明
VDD	P	电源
GND	P	地
P0, P1, P2	D	GPIO, 内部上/下拉
INT0~INT3	DI	外部中断输入
TC0~TC1	DI	定时器 T0~T1 的外部计数输入
PWM0/BUZ0	DO	定时器 T0 的 PWM/BUZ 输出
PWM1	DO	定时器 T1 的 PWM 输出
AN0~AN13	AI	ADC 模拟输入通道
RST	DI	外部复位输入
PCK, PDT, PDO	D	编程时钟/数据接口

注：P-电源；D-数字输入输出，DI-数字输入，DO-数字输出；A-模拟输入输出，AI-模拟输入，AO-模拟输出。

2 电气特性

2.1 极限参数

参数	符号	值	单位
电源电压	VDD	-0.3~6.0	V
输入电压	Vin	-0.3~VDD+0.3	V
工作温度	Ta	-40~85	°C
储存温度	Tstg	-65~150	°C
流入 VDD 最大电流	IVDDmax	100	mA
流出 GND 最大电流	IGNDmax	100	mA

注：若芯片工作条件超过极限值，将会造成永久性损坏；若芯片长时间工作在极限条件下，将会影响其可靠性。

2.2 直流电气特性

VDD=5V, T=25°C

特性	符号	端口	条件	最小	典型	最大	单位
工作电压	VDD	VDD	Fcpu=8MHz@FHIRC/2	VLVR28		5.5	V
			Fcpu=4MHz@FHIRC/4	VLVR24		5.5	
			Fcpu=2MHz@FHIRC/8	VLVR24		5.5	
			Fcpu=1MHz@FHIRC/16	VLVR20		5.5	
			Fcpu=16KHz@FLIRC/2	VLVR20		5.5	
输入漏电流	Ileak	所有输入脚	VDD=5V	-1		1	uA
输入高电平	Vih	所有输入脚		0.8VDD			V
输入低电平	Vil	所有输入脚				0.2VDD	V
输出拉电流	Ioh1	P0	Voh=0.9VDD		10		mA
	Ioh2	P1, P2	Voh=0.9VDD		20		mA
输出灌电流	Iol1	P0	Vol=0.1VDD		15		mA
	Iol2	P1, P2	Vol=0.1VDD		30		mA
上拉电阻	Rpu	P0, P1, P2	VDD=5V, Vin=0		30		KΩ
下拉电阻	Rpd	P0, P1, P2	Vin=VDD=5V		30		KΩ
动态功耗	Idd	VDD	VDD=5V, Fcpu=8MHz@HIRC		3.8		mA
			VDD=5V, Fcpu=4MHz@HIRC		2.5		mA
			VDD=5V, Fcpu=16KHz@LIRC		20		uA
HOLD1 功耗	Ihold1	VDD	VDD=5V, CPU 停, HIRC/LIRC 开		500		uA
HOLD2 功耗	Ihold2	VDD	VDD=5V, CPU 停, HIRC 关, LIRC 开		3	6	uA
休眠模式功耗	Istop	VDD	休眠模式, WDT/LVR 关		0.1	1	uA
			休眠模式, WDT 开, LVR 关		1	2	uA
			休眠模式, WDT 关, LVR 开		45	60	uA

低压复位电压	VLVR20	VDD		-15%	2.0	+15%	V
	VLVR24			-15%	2.4	+15%	V
	VLVR28			-15%	2.8	+15%	V
	VLVR36			-15%	3.6	+15%	V

注：功耗特性参数的条件说明中，诸如 HIRC/LIRC/WDT/LVR/LVD/ADC 等未注明模块，默认其为关闭状态。

2.3 ADC 特性参数

VDD=5V, T=25°C

特性	符号	条件	最小	典型	最大	单位
ADC 有效工作电压	V _{ADC}	T=-40°C~85°C	2.6		5.5	V
积分线性误差	INL	V _{REF} =VDD, F _{ADC} =1MHz, T _{con} =20us			±4	LSB
微分线性误差	DNL	V _{REF} =VDD, F _{ADC} =1MHz, T _{con} =20us			±2	LSB
零点偏移误差	EZ	V _{REF} =VDD, F _{ADC} =1MHz, T _{con} =20us			±4	LSB
增益误差	ET	V _{REF} =VDD, F _{ADC} =1MHz, T _{con} =20us			±4	LSB
转换时钟	F _{ADC}	VDD=5V	0.25		2	MHz
转换时间	T _{con}	F _{ADC} =1MHz	16		27	us
ADC 输入电压	V _{AIN}		GND		V _{REF}	V
ADC 输入阻抗	R _{AIN}		2			MΩ
ADC 输入电流	I _{AIN}				2	uA
ADC 动态电流	I _{ADD}	VDD=5V, ADC 转换中		1	3	mA
ADC 静态电流	I _{ADS}	VDD=5V, ADC 关闭		0.1	1	uA
模拟信号源推荐阻抗	Z _{AIN}				10	KΩ
ADC 参考电压	V _{REF}	选择 VDD		VDD		V
		选择内部参考电压 V _{IR} , T=25°C	-1%	2.0	+1%	
		选择内部参考电压 V _{IR} , T=-40°C~85°C	-2%	2.0	+2%	
V _{IR} 有效工作电压	V _{VIR}	选择内部参考电压 V _{IR}	V _{IR} +0.5		VDD	V

2.4 EERPOM 特性参数

特性	符号	条件	最小	典型	最大	单位
EEPROM 读操作电压	V _{EERD}	T=-40°C~85°C	1.8		5.5	V
EEPROM 写操作电压	V _{EEWR}	T=-40°C~85°C	2.6		5.5	V
EEPROM 操作时钟	F _{EE}	T=-40°C~85°C	250			KHz
EEPROM 字节写入时间	T _{EEWR}	VDD=5V, T=25°C			1	ms
		VDD=2.6V~5.5V, T=-40°C~85°C			20	ms
擦写次数		VDD=5V, T=25°C	10000			cycle

2.5 交流电气特性

特性	符号	条件	最小	典型	最大	单位
HIRC 振荡频率	F _{HIRC}	VDD=5V, T=25°C	-2%	16	+2%	MHz
		VDD=2.2V~5.5V, T=-40°C~85°C	-4%	16	+4%	MHz
LIRC 振荡频率	F _{LIRC}	VDD=5V, T=25°C	-50%	32	+50%	KHz

3 CPU 及存储器

3.1 指令集

本芯片指令集为精简指令集。除程序跳转类指令，其余指令均为单周期指令，即执行时间为 1 个指令周期；所有指令均为单字指令，即指令码只占用 1 个程序存储器地址空间。

指令汇总表

助记符	说明	操作	周期	长度	标志
ADDAR R	R 和 ACC 相加，结果存到 ACC	$R + ACC \rightarrow ACC$	1	1	C,DC,Z
ADDRA R	R 和 ACC 相加，结果存到 R	$R + ACC \rightarrow R$	1	1	C,DC,Z
ADCAR R	R 和 ACC 相加（带 C 标志），结果存到 ACC	$R + ACC + C \rightarrow ACC$	1	1	C,DC,Z
ADCRA R	R 和 ACC 相加（带 C 标志），结果存到 R	$R + ACC + C \rightarrow R$	1	1	C,DC,Z
RSUBAR R	R 和 ACC 相减，结果存到 ACC	$R - ACC \rightarrow ACC$	1	1	C,DC,Z
RSUBRA R	R 和 ACC 相减，结果存到 R	$R - ACC \rightarrow R$	1	1	C,DC,Z
RSBCAR R	R 和 ACC 相减（带 C 标志），结果存到 ACC	$R - ACC - C \rightarrow ACC$	1	1	C,DC,Z
RSBCRA R	R 和 ACC 相减（带 C 标志），结果存到 R	$R - ACC - C \rightarrow R$	1	1	C,DC,Z
ASUBAR R	ACC 和 R 相减，结果存到 ACC	$ACC - R \rightarrow ACC$	1	1	C,DC,Z
ASUBRA R	ACC 和 R 相减，结果存到 R	$ACC - R \rightarrow R$	1	1	C,DC,Z
ASBCAR R	ACC 和 R 相减（带 C 标志），结果存到 ACC	$ACC - R - C \rightarrow ACC$	1	1	C,DC,Z
ASBCRA R	ACC 和 R 相减（带 C 标志），结果存到 R	$ACC - R - C \rightarrow R$	1	1	C,DC,Z
ANDAR R	R 和 ACC 与操作，结果存到 ACC	$R \text{ and } ACC \rightarrow ACC$	1	1	Z
ANDRA R	R 和 ACC 与操作，结果存到 R	$R \text{ and } ACC \rightarrow R$	1	1	Z
ORAR R	R 和 ACC 或操作，结果存到 ACC	$R \text{ or } ACC \rightarrow ACC$	1	1	Z
ORRA R	R 和 ACC 或操作，结果存到 R	$R \text{ or } ACC \rightarrow R$	1	1	Z
XORAR R	R 和 ACC 异或操作，结果存到 ACC	$R \text{ xor } ACC \rightarrow ACC$	1	1	Z
XORRA R	R 和 ACC 异或操作，结果存到 R	$R \text{ xor } ACC \rightarrow R$	1	1	Z
COMAR R	对 R 取反，结果存到 ACC	$R \text{ 取反} \rightarrow ACC$	1	1	Z
COMR R	对 R 取反，结果存到 R	$R \text{ 取反} \rightarrow R$	1	1	Z
RLA	ACC 循环左移（带 C 标志）	$ACC[7] \rightarrow C$ $ACC[6:0] \rightarrow ACC[7:1]$ $C \rightarrow ACC[0]$	1	1	C
RLAR R	R 循环左移（带 C 标志），结果存到 ACC	$R[7] \rightarrow C$ $R[6:0] \rightarrow ACC[7:1]$ $C \rightarrow ACC[0]$	1	1	C
RLR R	R 循环左移（带 C 标志），结果存到 R	$R[7] \rightarrow C$ $R[6:0] \rightarrow R[7:1]$ $C \rightarrow R[0]$	1	1	C
RRA	ACC 循环右移（带 C 标志）	$C \rightarrow ACC[7]$ $ACC[7:1] \rightarrow ACC[6:0]$ $ACC[0] \rightarrow C$	1	1	C

RRAR	R	R 循环右移 (带 C 标志), 结果存到 ACC	C→ACC[7] R[7:1]→ACC[6:0] R[0]→C	1	1	C
RRR	R	R 循环右移 (带 C 标志), 结果存到 R	C→R[7] R[7:1]→R[6:0] R[0]→C	1	1	C
SWAPAR	R	交换 R 的高低半字节, 结果存到 ACC	R[7:4]→ACC[3:0] R[3:0]→ACC[7:4]	1	1	-
SWAPR	R	交换 R 的高低半字节, 结果存到 R	R[7:4]→R[3:0] R[3:0]→R[7:4]	1	1	-
MOVRA	R	将 ACC 存到 R	ACC→R	1	1	-
MOVAR	R	将 R 存到 ACC	R→ACC	1	1	Z
MOVR	R	将 R 存到 R	R→R	1	1	Z
CLRA		对 ACC 清零	0→ACC	1	1	Z
CLRR	R	对 R 清零	0→R	1	1	Z
INCA		ACC 自加 1	ACC+1→ACC	1	1	-
INCAR	R	R 加 1, 结果存到 ACC	R+1→ACC	1	1	Z
INCR	R	R 加 1, 结果存到 R	R+1→R	1	1	Z
DECA		ACC 自减 1	ACC-1→ACC	1	1	-
DECAR	R	R 减 1, 结果存到 ACC	R-1→ACC	1	1	Z
DECR	R	R 减 1, 结果存到 R	R-1→R	1	1	Z
JZA		ACC 自加 1; 结果为 0 则跳过下一条指令	ACC+1→ACC; 结果为 0 则 PC+2→PC	1/2	1	-
JZAR	R	R 加 1, 结果存到 ACC; 结果为 0 则跳过下一条指令	R+1→ACC; 结果为 0 则 PC+2→PC	1/2	1	-
JZR	R	R 加 1, 结果存到 R; 结果为 0 则跳过下一条指令	R+1→R; 结果为 0 则 PC+2→PC	1/2	1	-
DJZA		ACC 自减 1; 结果为 0 则跳过下一条指令	ACC-1→ACC; 结果为 0 则 PC+2→PC	1/2	1	-
DJZAR	R	R 减 1, 结果存到 ACC; 结果为 0 则跳过下一条指令	R-1→ACC; 结果为 0 则 PC+2→PC	1/2	1	-
DJZR	R	R 减 1, 结果存到 R; 结果为 0 则跳过下一条指令	R-1→R; 结果为 0 则 PC+2→PC	1/2	1	-
BCLR	R, b	对 R 的第 b 位清 0	0→R[b]	1	1	-
BSET	R, b	对 R 的第 b 位置 1	1→R[b]	1	1	-
JBCLR	R, b	若 R 的第 b 位为 0 则跳过下一条指令	若 R[b]=0, 则 PC+2→PC	1/2	1	-
JBSET	R, b	若 R 的第 b 位为 1 则跳过下一条指令	若 R[b]=1, 则 PC+2→PC	1/2	1	-
ADDAI	K	K 和 ACC 相加, 结果存到 ACC	K+ACC→ACC	1	1	C,DC,Z
ADCAI	K	K 和 ACC 相加 (带 C 标志), 结果存到 ACC	K+ACC+C→ACC	1	1	C,DC,Z
ISUBAI	K	K 和 ACC 相减, 结果存到 ACC	K-ACC→ACC	1	1	C,DC,Z
ISBCAI	K	K 和 ACC 相减 (带 C 标志), 结果存到 ACC	K-ACC-/C→ACC	1	1	C,DC,Z
ASUBAI	K	ACC 和 K 相减, 结果存到 ACC	ACC-K→ACC	1	1	C,DC,Z
ASBCAI	K	ACC 和 K 相减 (带 C 标志), 结果存到 ACC	ACC-K-/C→ACC	1	1	C,DC,Z
ANDAI	K	K 和 ACC 与操作, 结果存到 ACC	K and ACC→ACC	1	1	Z
ORAI	K	K 和 ACC 或操作, 结果存到 ACC	K or ACC→ACC	1	1	Z
XORAI	K	K 和 ACC 异或操作, 结果存到 ACC	K xor ACC→ACC	1	1	Z
MOVAI	K	将 K 存到 ACC	K→ACC	1	1	-
CALL	K	子程序调用	PC+1→TOS K→PC[12:0]	2	1	-

GOTO K	无条件跳转	K→PC[12:0]	2	1	-
RETURN	从子程序返回	TOS→PC	2	1	-
RETAI K	从子程序返回，并将 K 存到 ACC	TOS→PC K→ACC	2	1	-
RETIE	从中断返回	TOS→PC 1→GIE	2	1	-
NOP	空操作	空操作	1	1	-
DAA	BCD 码加法后，将 ACC 的值调整为 BCD 码	ACC(十六进制)→ACC(十进制)	1	1	C
DSA	BCD 码减法后，将 ACC 的值调整为 BCD 码	ACC(十六进制)→ACC(十进制)	1	1	-
CLRWDT	清看门狗定时器	0→WDT	1	1	TO,PD
STOP	进入低功耗模式	0→WDT；进入低功耗模式	1	1	TO,PD

注：

- 1、ACC-算数逻辑单元累加器，R-数据存储器，K-立即数；
- 2、对于条件跳转类指令，若跳转条件成立，则指令需 2 个周期，否则只需 1 个周期；

3.2 程序存储器

本芯片的程序存储器为 FLASH 型存储器，4K×16 位的地址为 0000H~0FFFH。

程序存储器地址分配如下图所示：

复位向量（0000H）
通用程序区 （0000H - 0007H）
中断向量（0008H）
通用程序区 （0000H - 0FFFH）

程序存储器支持间接寻址，可通过寄存器 INDF3 访问 FSR1×256+FSR0 指向的程序存储器地址中的内容。例：通过间接寻址读取程序存储器 0155H 地址中的内容，高 8 位存入数据存储器 11H 地址，低 8 位存入数据存储器 10H 地址

```

MOVAI    01H
MOVRA    FSR1          ; 将 01H 写入 FSR1
MOVAI    55H
MOVRA    FSR0          ; 将 55H 写入 FSR0
MOVAR    INDF3         ; 读取 FSR1×256+FSR0 指向的程序存储器地址（0155H）
                        ; 中的内容，高 8 位存入 HIBYTE，低 8 位存入 A 寄存器
MOVRA    10H           ; 低 8 位存入数据存储器 10H 地址
MOVAR    HIBYTE        ; 从 HIBYTE 读取高 8 位
MOVRA    11H           ; 高 8 位存入数据存储器 11H 地址

```

3.3 数据存储器

数据存储器包括通用数据存储器 GPR 和特殊功能寄存器 SFR，具体地址分配参照下表。GPR0 可直接寻址或通过 INDF0/INDF2 间接寻址，GPR1 和 SFR 可直接寻址或通过 INDF1/INDF2 间接寻址。

数据存储器还包括 EEPROM 存储器，需通过特殊寄存器 SFR 进行读写操作。

数据存储器区地址映射表

地址	类型	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
000H-0FFH	GPR0	通用数据存储器区 0							
100H-17FH	GPR1	通用数据存储器区 1							
180H-187H	SFR	INDF0	INDF1	INDF2	HIBYTE	FSR0	FSR1	PCL	PFLAG
188H-18FH		MCR	INDF3	INTE	INTF	OSCMR	INTE1	INTF1	P1KBCR
190H-197H		IOP0	OEP0	PUP0	PDP0	IOP1	OEP1	PUP1	PDP1
198H-19FH		IOP2	OEP2	PUP2	PDP2				
1A0H-1A7H		T0CR	T0CNT	T0LOAD	T0DATA				
1A8H-1AFH		T1CR	T1CNTH	T1CNTL	T1LOADH	T1LOADL	T1DATAH	T1DATAL	
1B0H-1B7H						EECR	EEMASK	EEAR	EEDR
1B8H-1BFH		ADCR0	ADCR1	ADRH	ADRL	ADIOS0	ADIOS1	OSADJCR	
1C0H-1C7H		T2CR	T2CNTH	T2CNTL	T2LOADH	T2LOADL	T3CR	T3CNT	T3LOAD
1C8H-1FFH	保留								

注：上表中灰色部分地址为系统保留区，读出数据不确定，写入操作可能会影响芯片正常工作。

数据存储器地址组成

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	寻址模式
0	0	0	0	0	0	0	来自指令的 9 位地址									直接寻址模式
0	0	0	0	0	0	0	0	FSR0								间接寻址模式 0
0	0	0	0	0	0	0	1	FSR1								间接寻址模式 1
FSR1								FSR0								间接寻址模式 2

直接寻址模式，是以指令的低 9 位为数据存储器地址，通过指令访问，寻址范围为 000H~1FFH。

例：通过直接寻址模式将数据 55H 写入数据存储器 010H 地址

MOVAI 55H

MOVRA 10H

：将数据 55H 写入数据存储器 10H 地址

间接寻址模式 0，是以 FSR0 为数据存储器地址指针，通过 INDF0 访问，寻址范围为 000H~0FFH。

例：通过间接寻址模式 0 将数据 55H 写入数据存储器 010H 地址

MOVAI 10H

MOVRA FSR0

MOVAI 55H

MOVRA INDF0

：将数据 55H 写入 FSR0 指向的数据存储器中

间接寻址模式 1，是以 FSR1 为数据存储器地址指针，通过 INDF1 访问，寻址范围为 100H~1FFH。

例：通过间接寻址模式 1 将数据 55H 写入数据存储器 110H 地址

```
MOVAI    10H
MOVRA    FSR1
MOVAI    55H
MOVRA    INDF1          : 将数据 55H 写入 FSR1 指向的数据存储器中
```

间接寻址模式 2，是以 [FSR1:FSR0] 为数据存储器地址指针，通过 INDF2 访问，寻址范围为 0000H ~ FFFFH。例：通过间接寻址模式 2 将数据 55H 写入数据存储器 0110H 地址

```
MOVAI    01H
MOVRA    FSR1
MOVAI    10H
MOVRA    FSR0
MOVAI    55H
MOVRA    INDF2          : 将数据 55H 写入 FSR1×256+FSR0 指向的数据存储器中
```

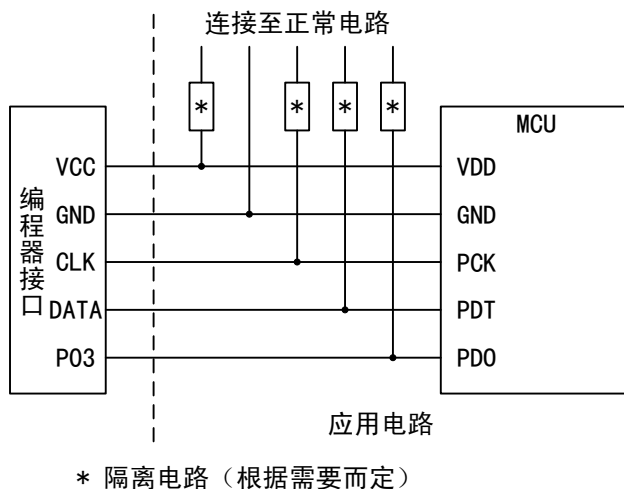
注：间接寻址模式 2 可寻址 0~FFFFH，但访问本芯片数据存储器中未定义的地址时，读出数据不确定，写入数据可能会更改其他地址中的数据。

3.4 在线编程

本芯片支持对存储器在线编程，可在最终应用电路中通过芯片的串行编程接口将用户程序代码或数据初值烧录进存储器中。在线编程功能，可让用户先采用未编程的空芯片制造电路板而仅在产品交付前才将代码或数据烧录进芯片，也方便用户直接在电路板上更新芯片中的代码或数据。

本芯片的在线编程通过引脚 VDD、GND、PCK、PDT、PDO 实现，编程引脚的外围电路需进行针对性设计，以保证外围电路不会影响在线编程时端口上的电压/电流/时序等特性。

下图是典型的在线编程连接图：



3.5 堆栈

8 级堆栈深度，当程序响应中断或执行子程序调用指令时 CPU 会将 PC 自动压栈保存；当执行中断返回指令或子程序返回指令时，栈顶数据自动出栈载入 PC。

3.6 控制寄存器

间接寻址寄存器 0

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INDF0	INDF07	INDF06	INDF05	INDF04	INDF03	INDF02	INDF01	INDF00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **INDF0[7:0]** – 间接寻址寄存器 0

INDF0: INDF0 不是物理寄存器，对 INDF0 寻址实际是对 FSR0 指向的数据存储器地址进行访问，从而实现间接寻址模式。

间接寻址寄存器 1

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INDF1	INDF17	INDF16	INDF15	INDF14	INDF13	INDF12	INDF11	INDF10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **INDF1[7:0]** – 间接寻址寄存器 1

INDF1: INDF1 不是物理寄存器，对 INDF1 寻址实际是对 FSR1+256 指向的数据存储器地址进行访问，从而实现间接寻址模式。

间接寻址寄存器 2

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INDF2	INDF27	INDF26	INDF25	INDF24	INDF23	INDF22	INDF21	INDF20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **INDF2[7:0]** – 间接寻址寄存器 2

INDF2: INDF2 不是物理寄存器，对 INDF2 寻址实际是对 FSR1×256+FSR0 指向的数据存储器地址进行访问，从而实现间接寻址模式。

间接寻址寄存器 3

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INDF3	INDF37	INDF36	INDF35	INDF34	INDF33	INDF32	INDF31	INDF30

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **INDF3[7:0]** – 间接寻址寄存器 3

INDF3: INDF3 不是物理寄存器，对 INDF3 寻址实际是对 FSR1×256+FSR0 指向的程序存储器地址进行访问，从而实现间接寻址模式。

注：寄存器 INDF3 仅可使用读取指令 (MOVAR INDF3) 进行读取操作，读取内容高 8 位存入 HIBYTE 寄存器，低 8 位存入 A 寄存器。

字操作高字节缓存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
HIBYTE	HIBYTE7	HIBYTE6	HIBYTE5	HIBYTE4	HIBYTE3	HIBYTE2	HIBYTE1	HIBYTE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **HIBYTE[7:0]** – 字操作高字节缓存器

HIBYTE: 用于读取 INDF3 时存放 FSR1×256+FSR0 指向的程序存储器内容高 8 位数据。

数据指针寄存器 0

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FSR0	FSR07	FSR06	FSR05	FSR04	FSR03	FSR02	FSR01	FSR00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **FSR0[7:0]** – 数据指针寄存器 0

FSR0: 间接寻址模式 0 的指针，或间接寻址模式 2、3 的指针低 8 位。

数据指针寄存器 1

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FSR1	FSR17	FSR16	FSR15	FSR14	FSR13	FSR12	FSR11	FSR10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **FSR1[7:0]** – 数据指针寄存器 1

FSR1: 间接寻址模式 1 的指针，或间接寻址模式 2、3 的指针高 8 位。

程序指针计数器低字节

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCL	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **PC[7:0]** – 程序指针计数器低 8 位

程序指针计数器（PC）有以下几种操作模式：

- ✧ 顺序运行指令：PC = PC + 1；
- ✧ 分支指令 GOTO/CALL：PC = 指令码低 13 位；
- ✧ 返回指令 RETIE/RETURN/RETAI：PC = 堆栈栈顶（TOS）；

对 PCL 操作指令：

- ✧ 对 PCL 操作的加法指令：PC = (PC[15:0]+ALU[7:0])；
- ✧ 对 PCL 操作的其它指令：PC = {PC[15:8]:ALU[7:0](ALU 运算结果)}；

CPU 状态寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PFLAG	-	-	-	-	-	Z	DC	C
R/W	-	-	-	-	-	R/W	R/W	R/W
初始值	-	-	-	-	-	X	X	X

BIT[2] **Z** – 零标志位

- 0: 算术或逻辑运算的结果不为零；
- 1: 算术或逻辑运算的结果为零；

BIT[1] **DC** – 半字节进/借位标志位

- 0: 加法运算时半字节无进位；减法运算时半字节有借位；
- 1: 加法运算时半字节有进位；减法运算时半字节无借位；

BIT[0] **C** – 进/借位标志位

- 0: 加法运算时无进位；减法运算时有借位；移位后移出逻辑 0；
- 1: 加法运算时有进位；减法运算时无借位；移位后移出逻辑 1；

杂项控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MCR	GIE	-	TO	PD	MINT11	MINT10	MINT01	MINT00
R/W	R/W	-	R	R	R/W	R/W	R/W	R/W
初始值	0	-	0	0	0	0	0	0

BIT[7] **GIE** – 中断总使能位

- 0: 屏蔽所有中断；
- 1: 由对应的中断使能位决定是否响应中断源触发的中断；

BIT[5] **TO** – 看门狗溢出标志位

- 0: 上电复位；执行 CLRWDT 或 STOP 指令；
- 1: 发生 WDT 溢出；

BIT[4] **PD** – 进入低功耗模式标志位

- 0: 上电复位；执行 CLRWDT 指令；
- 1: 执行 STOP 指令；

BIT[3:2] MINT1[1:0] – 外部中断 INT1 触发方式选择位

MINT1[1:0]	INT1 触发方式
00	上升沿触发
01	下降沿触发
1X	电平变化触发

BIT[1:0] MINT0[1:0] – 外部中断 INT0 触发方式选择位

MINT0[1:0]	INT0 触发方式
00	上升沿触发
01	下降沿触发
1X	电平变化触发

3.7 用户配置字

芯片为保证系统正常工作，会将关键模块的配置信息预先存储于单独的存储器区域中，在上电或其他复位发生后将配置信息载入寄存器中，通过寄存器控制关键模块的工作状态。该部分存储器中用户可选的内容即为用户配置字，可在烧录用户程序代码时进行配置与烧录。

本芯片的用户配置字，定义如下：

符号	功能说明
FCPUS	CPU 高速模式时钟频率选择： FCPU= F _{HOSC} /2 ; FCPU= F _{HOSC} /4 ; FCPU= F _{HOSC} /8 ; FCPU= F _{HOSC} /16 ; FCPU= F _{HOSC} /32 ; FCPU= F _{HOSC} /64 ;
VLVRS	LVR 复位电压选择： 2.0V ; 2.4V ; 2.8V ; 3.6V ;
LVRSLP	LVR 低功耗模式设置： LVR 低功耗模式下使能； LVR 低功耗模式下关闭；
WDTM	WDT 模式设置： WDT 始终关闭； WDT 低功耗模式下关闭 WDT 始终开启；
WDTT	WDT 溢出时间选择： TWDT=64ms ; TWDT=2048ms ;
RSTEN	RST 外部复位设置： P05 用作复位脚； P05 用作 I/O 脚；
ENCR	代码加密设置： 程序代码加密； 程序代码不加密；

4 系统时钟

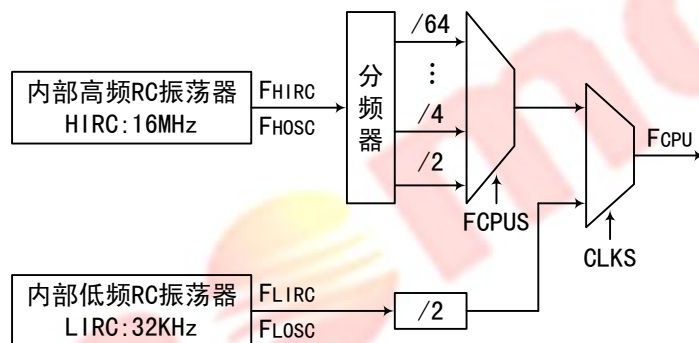
本芯片为双时钟系统，内部电路均在系统高频时钟 F_{HOSC} 或系统低频时钟 F_{LOSC} 下工作，部分模块的时钟还可在 F_{HOSC} 和 F_{LOSC} 之间切换。

本芯片系统高频时钟固定为内部高频 RC 振荡器 HIRC（16MHz），系统低频时钟固定为内部低频 RC 振荡器 LIRC（32KHz）。

CPU 的时钟源可在系统高频时钟 F_{HOSC} 和系统低频时钟 F_{LOSC} 之间切换。选择 F_{HOSC} 时，CPU 的工作时钟 F_{CPU} 通过用户配置字 FCPUS 选择；选择 F_{LOSC} 时，F_{CPU} 固定为 F_{LOSC} 的 2 分频。

WDT（看门狗）电路的时钟源固定为内部低频 RC 振荡器。

系统时钟示意图



4.1 内部高频 RC 振荡器

本芯片内置一个振荡频率为 16MHz 的高精度 HIRC 振荡器，可用作系统高频时钟源。

4.2 内部低频 RC 振荡器

本芯片内置一个振荡频率典型值为 32KHz 的 LIRC 振荡器，可用作系统低频时钟源，同时用于上电延时定时器、WDT 定时器等电路。

4.3 工作模式

本芯片支持高速模式、低速模式、HOLD1 模式、HOLD2 模式和休眠模式等多种系统工作模式。

工作模式	转入条件	系统状态
高速模式	复位重启； 低速模式下，CLKS 位清 0； HOLD1/HOLD2/休眠模式下，CPU 唤醒；	CPU 高速运行，高/低频时钟源均工作
低速模式	高速模式下，CLKS 位置 1； HOLD1/HOLD2/休眠模式下，CPU 唤醒；	CPU 低速运行，高频时钟源由 HFEN 决定
HOLD1	高/低速模式下，HFEN 置 1，执行 STOP；	CPU 暂停，高频时钟源工作，低频时钟源由 LFEN 决定
HOLD2	高/低速模式下，HFEN 清 0，LFEN 置 1，执行 STOP；	CPU 暂停，高频时钟源停止，低频时钟源工作
休眠模式	高/低速模式下，HFEN 清 0，LFEN 清 0，执行 STOP；	CPU 暂停，高/低频时钟源均停止

注：WDT 时钟源为 LIRC，WDT 开启时 LIRC 将一直工作而不受系统工作模式影响。

工作模式寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OSCMR	-	-	STBL	STBH	-	CLKS	LFEN	HFEN
R/W	-	-	R	R	-	R/W	R/W	R/W
初始值	-	-	X	1	-	0	0	0

BIT[5] **STBL** – 低频时钟源振荡稳定标志位

0: 低频时钟源停振或未稳定；

1: 低频时钟源已稳定振荡；

BIT[4] **STBH** – 高频时钟源振荡稳定标志位

0: 高频时钟源停振或未稳定；

1: 高频时钟源已稳定振荡；

BIT[2] **CLKS** – CPU 时钟源选择位

0: 系统高频时钟作为 CPU 时钟源；

1: 系统低频时钟作为 CPU 时钟源；

BIT[1] **LFEN** – 低频时钟源使能位

0: 在休眠/HOLD 模式下，低频时钟源停止工作；

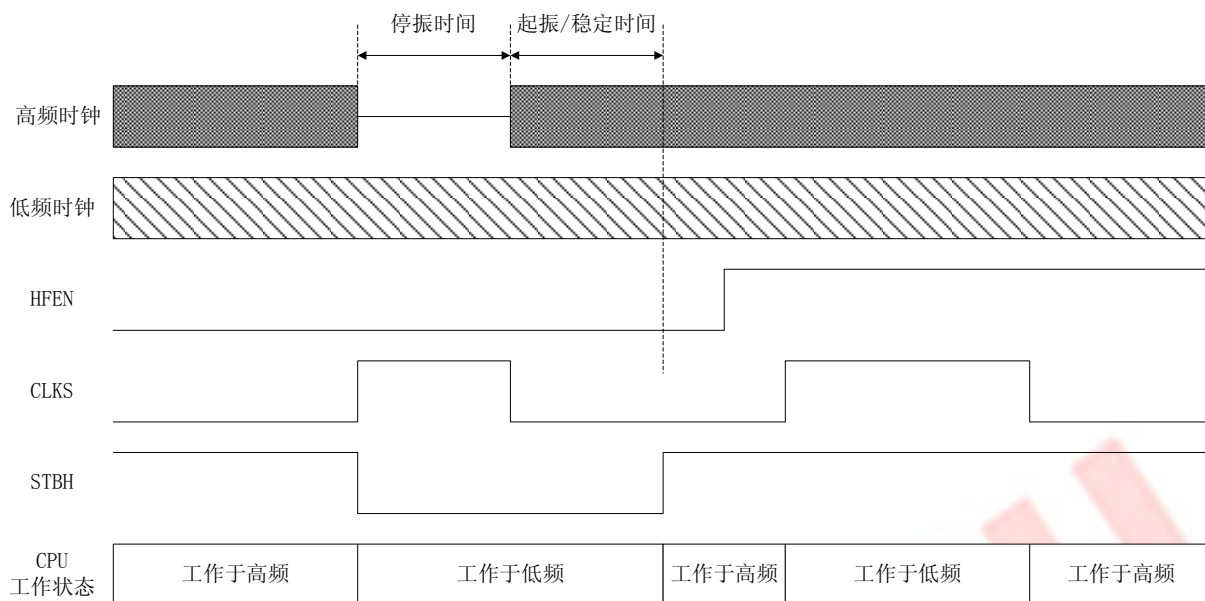
1: 低频时钟源始终工作；

BIT[0] **HFEN** – 高频时钟源使能位

0: 在低速/休眠/HOLD 模式下，高频时钟源停止工作；

1: 高频时钟源始终工作；

高低频时钟切换时序图



4.4 低功耗模式

本芯片的低功耗模式包括休眠模式、HOLD1 模式、HOLD2 模式。

执行 STOP 指令可使系统进入低功耗模式，同时对系统会产生以下影响：

- ✧ CPU 停止运行；
- ✧ 根据不同模式停止相应时钟源的振荡；
- ✧ RAM 内容保持不变；
- ✧ 所有的输入输出端口保持原态不变；
- ✧ 定时器若其时钟源未停止，则可以保持继续工作；

以下情况可使系统退出低功耗模式：

- ✧ 有外部中断请求发生（若有外部中断功能）；
- ✧ 定时器溢出中断发生（若低功耗模式下定时器保持继续工作）；
- ✧ 有键盘中断请求发生（若有键盘中断功能）；
- ✧ 有 WDT 溢出（若低功耗模式下 WDT 保持继续工作）；
- ✧ 外部复位（若有外部复位功能）；
- ✧ 上电复位；

注：低功耗模式下触发中断请求时，若对应的中断使能位关闭，则不会退出低功耗模式；若对应的中断使能位开启而中断总使能位关闭，则仅唤醒 CPU 执行下一条指令；若对应的中断使能位和中断总使能位均开启，则唤醒 CPU 后执行中断服务程序。

5 复位

5.1 复位条件

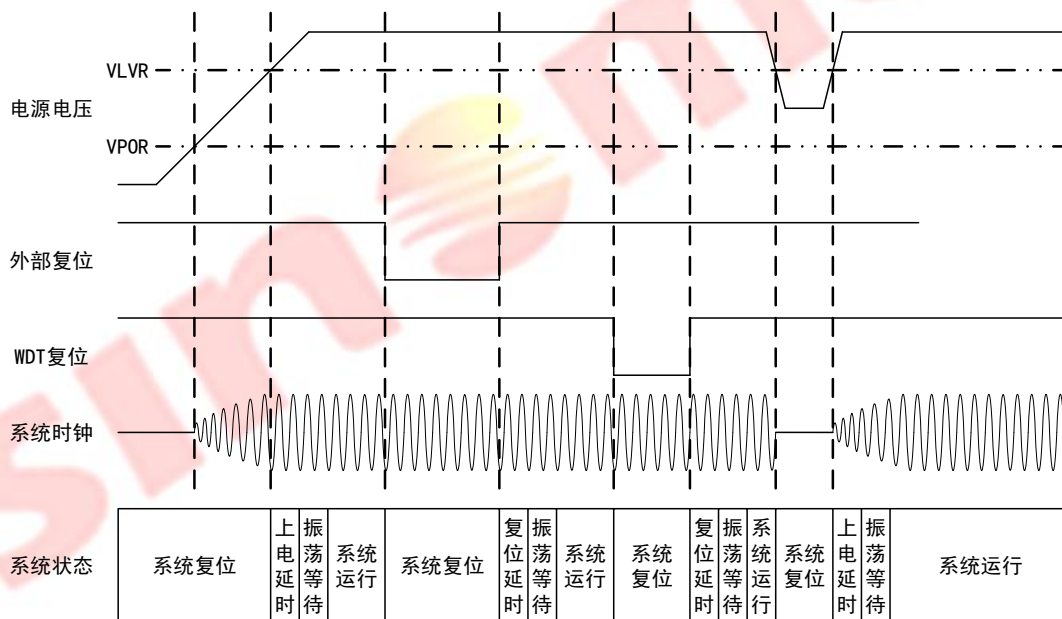
本芯片有如下几种复位方式：

- ✧ 上电复位 POR；
- ✧ 外部复位；
- ✧ 低电压复位 LVR；
- ✧ WDT 看门狗复位；

任何一种复位发生时，系统将会重新从 0000H 地址处开始执行指令；另外系统还会将所有特殊功能寄存器 SFR 重置为复位初始值。

上电复位和 LVR 复位会关闭系统主时钟振荡器，复位解除后才重新打开振荡器，由于振荡器起振和稳定需要一定的时间，所以系统会保持一定时间的上电延时和振荡等待后才开始工作；外部复位和 WDT 复位不会关闭系统主时钟振荡器，复位解除时系统会在较短的复位延时和振荡等待后即开始工作。

下图是复位产生和系统工作状态之间的关系示意图：



5.2 上电复位

本芯片的上电复位电路可以适应快速、慢速上电的情况，并且当芯片上电过程中出现电源电压抖动时都能保证系统可靠的复位。

上电复位过程可以概括为以下几个步骤：

- (1) 检测系统工作电压，等待电压高于上电复位电压 V_{POR} 并保持稳定；

- (2) 若有 LVR 功能，则需等待电压高于 V_{LVR} 并保持稳定；
- (3) 若有外部复位功能，则需等待复位引脚电压高于 V_{ih} ；
- (4) 初始化所有寄存器；
- (5) 开启主时钟振荡器，并等待一段时间以待振荡器稳定；
- (6) 上电结束，系统开始执行指令。

5.3 外部复位

本芯片的外部复位功能可通过用户配置字 **RSTEN** 开启，详见用户配置字，选择引脚用作复位脚即为开启外部复位功能，引脚的内部上拉电阻自动有效。外部复位输入端口 **RST** 为施密特结构，低电平有效。当外部复位输入为高电平时，系统正常运行；输入为低电平时，系统复位。

5.4 低电压复位

本芯片的 LVR 电压点可通过用户配置字 **VLVRS** 选择，详见用户配置字。电压检测电路有一定的回滞特性，通常回滞电压为 0.1V 左右，当电源电压下降到 LVR 电压时 LVR 复位有效，而电压需要上升到 LVR 电压+0.1V 时 LVR 复位才会解除。

5.5 看门狗复位

看门狗（WDT）复位是一种对程序正常运行的保护机制。正常情况下，用户软件需要按时对 WDT 定时器进行清零操作，保证 WDT 不溢出。若出现异常状况，程序未按时对 WDT 定时器清零，WDT 会溢出从而产生看门狗复位，系统重新初始化，返回受控状态。

注：低功耗模式下 CPU 暂停工作，若此时有 WDT 溢出，则仅唤醒 CPU 而不产生复位。

6 I/O 端口

6.1 I/O 工作模式

芯片共有两组 8 位端口 P0、P1 和一组 6 位端口 P2。除用作通用输入/输出端口外，部分端口还可复用为外部中断输入端口、PWM/BUZ 输出端口、或 ADC 模拟输入端口。

端口数据寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IOP0	P07D	P06D	P05D	P04D	P03D	P02D	P01D	P00D
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **P0nD** – P0n 口数据位 (n=7-0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IOP1	P17D	P16D	P15D	P14D	P13D	P12D	P11D	P10D
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **P1nD** – P1n 口数据位 (n=7-0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IOP2	-	-	P25D	P24D	P23D	P22D	P21D	P20D
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	-	X	X	X	X	X	X

BIT[5:0] **P2nD** – P2n 口数据位 (n=5-0)

端口方向寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OEP0	P07OE	P06OE	P05OE	P04OE	P03OE	P02OE	P01OE	P00OE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **P0nOE** – P0n 口输出使能位 (n=7-0)

0: 端口作为输入口，读端口操作将读取端口状态；

1: 端口作为输出口，读端口操作将读取数据寄存器值；

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OEP1	P17OE	P16OE	P15OE	P14OE	P13OE	P12OE	P11OE	P10OE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **P1nOE** – P1n 口输出使能位 (n=7-0)

- 0: 端口作为输入口, 读端口操作将读取端口状态;
1: 端口作为输出口, 读端口操作将读取数据寄存器值;

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OEP2	-	-	P25OE	P24OE	P23OE	P22OE	P21OE	P20OE
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	-	0	0	0	0	0	0

BIT[5:0] **P2nOE** – P2n 口输出使能位 (n=5-0)

- 0: 端口作为输入口, 读端口操作将读取端口状态;
1: 端口作为输出口, 读端口操作将读取数据寄存器值;

6.2 内部上/下拉电阻

所有端口均有内部上拉和下拉电阻, 均有独立的上/下拉电阻寄存器控制位, 控制其上/下拉电阻在端口处于输入状态时是否有效。端口处于输出状态时, 上/下拉电阻及其控制位无效。

上拉电阻控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PUP0	P07PU	P06PU	P05PU	P04PU	P03PU	P02PU	P01PU	P00PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **P0nPU** – P0n 口上拉电阻控制位 (n=7-0)

- 0: 端口内部上拉电阻无效;
1: 端口内部上拉电阻有效;

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PUP1	P17PU	P16PU	P15PU	P14PU	P13PU	P12PU	P11PU	P10PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **P1nPU** – P1n 口上拉电阻控制位 (n=7-0)

- 0: 端口内部上拉电阻无效;
1: 端口内部上拉电阻有效;

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PUP2	-	-	P25PU	P24PU	P23PU	P22PU	P21PU	P20PU
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	-	0	0	0	0	0	0

BIT[5:0] **P2nPU** – P2n 口上拉电阻控制位 (n=5-0)

- 0: 端口内部上拉电阻无效;
1: 端口内部上拉电阻有效;

下拉电阻控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PDP0	P07PD	P06PD	P05PD	P04PD	P03PD	P02PD	P01PD	P00PD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **P0nPD** – P0n 口下拉电阻控制位 (n=7-0)

0: 端口内部下拉电阻无效;

1: 端口内部下拉电阻有效;

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PDP1	P17PD	P16PD	P15PD	P14PD	P13PD	P12PD	P11PD	P10PD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **P1nPD** – P1n 口下拉电阻控制位 (n=7-0)

0: 端口内部下拉电阻无效;

1: 端口内部下拉电阻有效;

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PDP2	-	-	P25PD	P24PD	P23PD	P22PD	P21PD	P20PD
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	-	0	0	0	0	0	0

BIT[5:0] **P2nPD** – P2n 口下拉电阻控制位 (n=5-0)

0: 端口内部下拉电阻无效;

1: 端口内部下拉电阻有效;

6.3 端口模式控制

部分端口除可作为数字 I/O 端口外, 还可复用为模拟 I/O 端口。端口输入或输出模拟信号时, 若数字 I/O 功能同时开启, 则会产生漏电流, 可通过端口数模控制寄存器关闭端口的数字 I/O 功能。

端口数模控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADIOS0	AN7DM	AN6DM	AN5DM	AN4DM	AN3DM	AN2DM	AN1DM	AN0DM
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **ANnDM** – 模拟端口 ANn 数字功能屏蔽位 (n=7-0)

0: 开启端口的数字 I/O 功能;

1: 关闭端口的数字 I/O 功能;

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADIOS1	-	-	AN13DM	AN12DM	AN11DM	AN10DM	AN9DM	AN8DM
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	-	0	0	0	0	0	0

BIT[5:0] **ANnDM** – 模拟端口 ANn 数字功能屏蔽位 (n=13-8)

0: 开启端口的数字 I/O 功能;

1: 关闭端口的数字 I/O 功能;

7 定时器 TIMER

7.1 看门狗定时器 WDT

看门狗定时器 WDT 的时钟源为内部低频 RC 振荡器，WDT 计数器溢出将复位芯片或唤醒 CPU。

可通过用户配置字 WDTM 设置 WDT 工作模式。若选择始终开启，则 WDT 一直工作，高速/低速模式下 WDT 溢出将复位芯片，休眠/HOLD 模式下 WDT 溢出将唤醒 CPU；若选择低功耗模式下关闭，则 WDT 在休眠/HOLD 模式下自动停止、在其他方式唤醒 CPU 后恢复工作。

执行 CLRWDT 指令将清零 WDT 计数器。

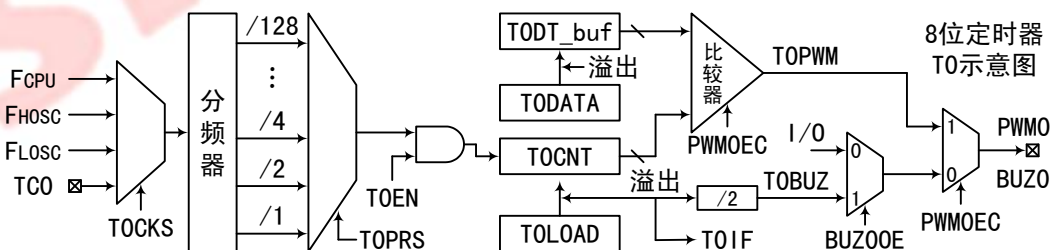
WDT 溢出时间可配置为 64ms/2048ms。

注：WDT 溢出时间为典型值，实际值偏差大，必须保证清 WDT 时间小于典型值的 1/4。

7.2 定时器 T0

定时器 T1 为 8 位定时/计数器，包含 1 个 8 位递减计数器、可编程预分频器、控制寄存器、8 位重载寄存器及比较寄存器。

- ✧ 可通过预分频比设置计数频率，可通过重载寄存器控制计数周期；
- ✧ 支持 PWM 输出，可通过比较寄存器设置 PWM 占空比；
- ✧ 支持 BUZ 输出；
- ✧ 支持溢出中断和溢出唤醒功能；



定时器 T0，可通过寄存器位 T0CKS 选择时钟源，通过 T0PRS 选择预分频比，所选时钟源通过预分频器后产生 T0 计数器 TOCNT 的计数时钟。预分频比可选择 1~128 分频，对 TOCNT 的写操作将清零预分频计数器，而预分频比保持不变。

当 T0EN=0 时，TOCNT 保持不变，写重载寄存器 TOLOAD 将立即载入 TOCNT 中；当 T0EN=1 时，TOCNT 递减计数，计数到 0 时产生溢出信号并触发中断，中断标志 T0IF 将被置 1，在下一个时钟 T0 自动将当前 TOLOAD 值载入 TOCNT 中重新开始计数。

如图所示，定时器 T0 可实现 BUZ 功能，当 BUZ0OE=1 且 PWM0EC=0 时，端口输出蜂鸣器驱动信号，频率为 T0 溢出频率的 2 分频。

如图所示，定时器 T0 可实现 PWM 功能，PWM0EC 置 1 将使能 PWM0 功能并允许端口输出 PWM 波形。每个 PWM 周期内，计数器 T0CNT 从重载值开始递减计数：当计数到与比较寄存器 T0DATA 相等时，T0PWM 信号变为高电平；当计数溢出时，T0PWM 信号变为低电平。

T0DATA 配有 1 个 8 位缓冲器用于与 T0CNT 比较，PWM0 功能关闭时写 T0DATA 将立即载入缓冲器中，而 PWM0 功能使能后写 T0DATA 则将在 T0 溢出时才载入缓冲器中。若要首个 PWM 周期和占空比准确，需先写重载寄存器和比较寄存器，再开启定时器和 PWM 功能。

T0PWM 信号的占空比计算如下：

- ✧ 高电平时间 = (T0DATA) × T0CNT 计数时钟周期
- ✧ 周期 (T0 的溢出周期) = (T0LOAD+1) × T0CNT 计数时钟周期
- ✧ 占空比 = 高电平时间 / 周期 = (T0DATA) / (T0LOAD+1)

定时器 T0 控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0CR	TOEN	PWM0EC	BUZ0OE	T0CKS1	T0CKS0	T0PRS2	T0PRS1	T0PRS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] **TOEN** – 定时器 T0 使能位

- 0: 关闭定时器 T0;
- 1: 开启定时器 T0;

BIT[6] **PWM0EC** – PWM0 使能及输出控制位

- 0: 关闭 PWM0 功能，并禁止端口输出 PWM 波形;
- 1: 使能 PWM0 功能，并允许端口输出 PWM 波形;

BIT[5] **BUZ0OE** – BUZ0 输出使能位

- 0: 禁止端口输出 BUZ 波形;
- 1: 允许端口输出 BUZ 波形 (仅 PWM0EC=0 时有效);

BIT[4:3] **T0CKS[1:0]** – T0 时钟源选择位

T0CKS[1:0]	T0 时钟源
00	FCPU
01	FHOSC
10	FLOSC
11	TC0 上升沿

BIT[2:0] **T0PRS[2:0]** – T0 预分频比选择位

T0PRS[2:0]	T0 时钟预分频比
000	1:1
001	1:2
010	1:4

011	1:8
100	1:16
101	1:32
110	1:64
111	1:128

定时器 T0 计数器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0CNT	T0CNT7	T0CNT6	T0CNT5	T0CNT4	T0CNT3	T0CNT2	T0CNT1	T0CNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] T0CNT[7:0] – T0 计数器，为可读写的递减计数器

定时器 T0 重载寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0LOAD	T0LOAD7	T0LOAD6	T0LOAD5	T0LOAD4	T0LOAD3	T0LOAD2	T0LOAD1	T0LOAD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] T0LOAD[7:0] – T0 重载寄存器，用于设置 T0 的计数周期

注：定时器重载寄存器的值禁止为 0，否则定时器将无法正常工作。

定时器 T0 比较寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0DATA	T0DATA7	T0DATA6	T0DATA5	T0DATA4	T0DATA3	T0DATA2	T0DATA1	T0DATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] T0DATA[7:0] – T0 比较寄存器，用于设置 PWM0 的占空比

7.3 定时器 T1

定时器 T1 为 16 位定时/计数器，包含 1 个 16 位递减计数器、可编程预分频器、控制寄存器、16 位重载寄存器及比较寄存器。

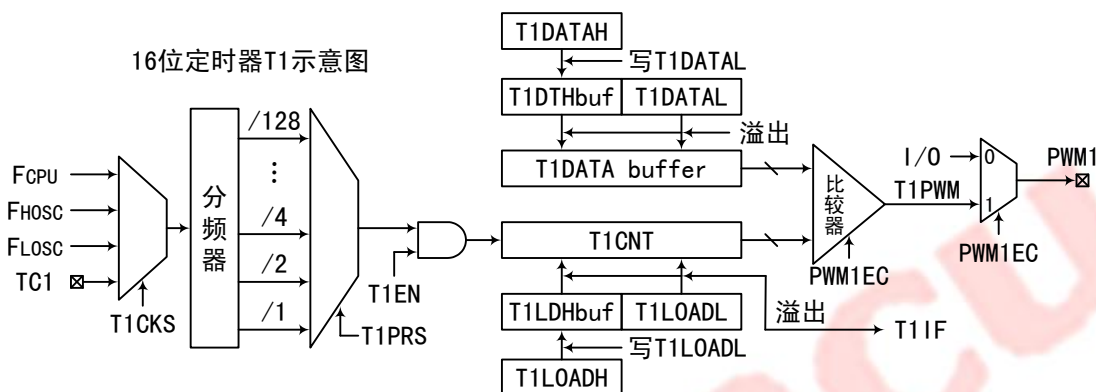
- ✧ 可通过预分频比设置计数频率，可通过重载寄存器控制计数周期；
- ✧ 支持 16 位 PWM 输出，可通过比较寄存器设置 PWM 占空比；
- ✧ 支持溢出中断和溢出唤醒功能；

定时器 T1，可通过寄存器位 T1CKS 选择时钟源，通过 T1PRS 选择预分频比，所选时钟源通过预分频器后产生 T1 计数器 T1CNT 的计数时钟。预分频比可选择 1~128 分频，对 T1CNT 的写操作将清零预

分频计数器，而预分频比保持不变。

当 T1EN=0 时，T1CNT 保持不变，写重载寄存器 T1LOAD 将立即载入 T1CNT 中；当 T1EN=1 时，T1CNT 递减计数，计数到 0 时产生溢出信号并触发中断，中断标志 T1IF 将被置 1，在下一个时钟 T1 自动将当前 T1LOAD 值载入 T1CNT 中重新开始计数。

16 位 T1LOAD 配有 1 个 8 位的高位缓冲器，写 T1LOADL 时会同时将 T1LOADH 值载入该缓冲器中。此时若 T1EN=0，则会同时再将[缓冲器:T1LOADL]载入 T1CNT；若 T1EN=1，则需在 T1 溢出后才将[缓冲器:T1LOADL]的值载入 T1CNT。调整 T1LOAD 值时需先写 T1LOADH，再写 T1LOADL。



如图所示，定时器 T1 可实现 PWM 功能，PWM1EC 置 1 将使能 PWM1 功能并允许端口输出 PWM 波形。每个 PWM 周期内，计数器 T1CNT 从重载值开始递减计数：当计数到与比较寄存器 T1DATA 相等时，T1PWM 信号变为高电平；当计数溢出时，T1PWM 信号变为低电平。

16 位 T1DATA 配有 1 个 8 位的高位缓冲器和 1 个 16 位的比较缓冲器，写 T1DATAH 时会同时将 T1DATAH 值载入高位缓冲器中，此时若 PWM1 功能关闭，则会同时再将[高位缓冲器:T1DATAH]载入比较缓冲器中，若 PWM1 功能已使能，则将在 T1 溢出后才将[高位缓冲器:T1DATAH]载入比较缓冲器中。若要首个 PWM 周期和占空比准确，需先写重载寄存器和比较寄存器，再开启定时器和 PWM 功能。

T1PWM 信号的占空比计算如下：

- ✧ 高电平时间 = (T1DATA) × T1CNT 计数时钟周期
- ✧ 周期 (T1 的溢出周期) = (T1LOAD+1) × T1CNT 计数时钟周期
- ✧ 占空比 = 高电平时间 / 周期 = (T1DATA) / (T1LOAD+1)

定时器 T1 控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1CR	T1EN	PWM1EC	-	T1CKS1	T1CKS0	T1PRS2	T1PRS1	T1PRS0
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
初始值	0	0	-	0	0	0	0	0

BIT[7] **T1EN** – 定时器 T1 使能位

- 0: 关闭定时器 T1;
- 1: 开启定时器 T1;

BIT[6] **PWM1EC** – PWM1 使能及输出控制位

- 0: 关闭 PWM1 功能，并禁止端口输出 PWM 波形;
- 1: 使能 PWM1 功能，并允许端口输出 PWM 波形;

BIT[4:3] T1CKS[1:0] – T1 时钟源选择位

T1CKS[1:0]	T1 时钟源
00	FCPU
01	FHOSC
10	FLOSC
11	TC1 上升沿

BIT[2:0] T1PRS[2:0] – T1 预分频比选择位

T1PRS[2:0]	T1 时钟预分频比
000	1:1
001	1:2
010	1:4
011	1:8
100	1:16
101	1:32
110	1:64
111	1:128

定时器 T1 计数器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1CNTH	T1CNT15	T1CNT14	T1CNT13	T1CNT12	T1CNT11	T1CNT10	T1CNT9	T1CNT8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] T1CNT[15:8] – T1 计数器高 8 位，为可读写的递减计数器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1CNTL	T1CNT7	T1CNT6	T1CNT5	T1CNT4	T1CNT3	T1CNT2	T1CNT1	T1CNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] T1CNT[7:0] – T1 计数器低 8 位，为可读写的递减计数器

定时器 T1 重载寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1LOADH	T1LOAD15	T1LOAD14	T1LOAD13	T1LOAD12	T1LOAD11	T1LOAD10	T1LOAD9	T1LOAD8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] T1LOAD[15:8] – T1 重载寄存器高 8 位，用于设置 T1 的计数周期

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1LOADL	T1LOAD7	T1LOAD6	T1LOAD5	T1LOAD4	T1LOAD3	T1LOAD2	T1LOAD1	T1LOAD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

初始值	1	1	1	1	1	1	1	1
-----	---	---	---	---	---	---	---	---

BIT[7:0] **T1LOAD[7:0]** – T1 重载寄存器低 8 位，用于设置 T1 的计数周期

注：定时器重载寄存器的值禁止为 0，否则定时器将无法正常工作。

定时器 T1 比较寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1DATAH	T1DATA15	T1DATA14	T1DATA13	T1DATA12	T1DATA11	T1DATA10	T1DATA9	T1DATA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **T1DATA[15:8]** – T1 比较寄存器高 8 位，用于设置 PWM1 的占空比

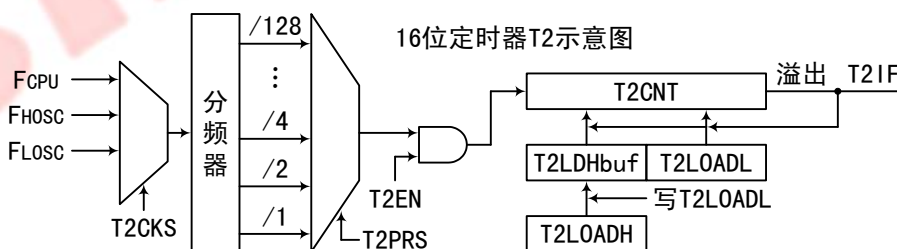
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1DATAL	T1DATA7	T1DATA6	T1DATA5	T1DATA4	T1DATA3	T1DATA2	T1DATA1	T1DATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **T1DATA[7:0]** – T1 比较寄存器低 8 位，用于设置 PWM1 的占空比

7.4 定时器 T2

定时器 T2 为 16 位定时器，包含 1 个 16 位递减计数器、可编程预分频器、控制寄存器、16 位重载寄存器。

- ◇ 可通过预分频比设置计数频率，可通过重载寄存器控制计数周期；
- ◇ 支持溢出中断和溢出唤醒功能；



定时器 T2，可通过寄存器位 T2CKS 选择时钟源，通过 T2PRS 选择预分频比，所选时钟源通过预分频器后产生 T2 计数器 T2CNT 的计数时钟。预分频比可选择 1~128 分频，对 T2CNT 的写操作将清零预分频计数器，而预分频比保持不变。

当 T2EN=0 时，T2CNT 保持不变，写重载寄存器 T2LOAD 将立即载入 T2CNT 中；当 T2EN=1 时，T2CNT 递减计数，计数到 0 时产生溢出信号并触发中断，中断标志 T2IF 将被置 1，在下一个时钟 T2 自动将当前 T2LOAD 值载入 T2CNT 中重新开始计数。

16 位 T2LOAD 配有 1 个 8 位的高位缓冲器，写 T2LOADL 时会同时将 T2LOADH 值载入该缓冲器

中。此时若 T2EN=0，则会同时再将[缓冲器:T2LOADL]载入 T2CNT；若 T2EN=1，则需在 T2 溢出后才将[缓冲器:T2LOADL]的值载入 T2CNT。调整 T2LOAD 值时需先写 T2LOADH，再写 T2LOADL。

定时器 T2 控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2CR	T2EN	-	-	T2CKS1	T2CKS0	T2PRS2	T2PRS1	T2PRS0
R/W	R/W	-	-	R/W	R/W	R/W	R/W	R/W
初始值	0	-	-	0	0	0	0	0

BIT[7] T2EN – 定时器 T2 使能位

0: 关闭定时器 T2;

1: 开启定时器 T2;

BIT[4:3] T2CKS[1:0] – T2 时钟源选择位

T2CKS[1:0]	T2 时钟源
00	FCPU
01	FHOSC
10	FLOSC
11	-

BIT[2:0] T2PRS[2:0] – T2 预分频比选择位

T2PRS[2:0]	T2 时钟预分频比
000	1:1
001	1:2
010	1:4
011	1:8
100	1:16
101	1:32
110	1:64
111	1:128

定时器 T2 计数器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2CNTH	T2CNT15	T2CNT14	T2CNT13	T2CNT12	T2CNT11	T2CNT10	T2CNT9	T2CNT8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] T2CNT[15:8] – T2 计数器高 8 位，为可读写的递减计数器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2CNTL	T2CNT7	T2CNT6	T2CNT5	T2CNT4	T2CNT3	T2CNT2	T2CNT1	T2CNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] T2CNT[7:0] – T2 计数器低 8 位，为可读写的递减计数器

定时器 T2 重载寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2LOADH	T2LOAD15	T2LOAD14	T2LOAD13	T2LOAD12	T2LOAD11	T2LOAD10	T2LOAD9	T2LOAD8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] **T2LOAD[15:8]** – T2 重载寄存器高 8 位，用于设置 T2 的计数周期

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2LOADL	T2LOAD7	T2LOAD6	T2LOAD5	T2LOAD4	T2LOAD3	T2LOAD2	T2LOAD1	T2LOAD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

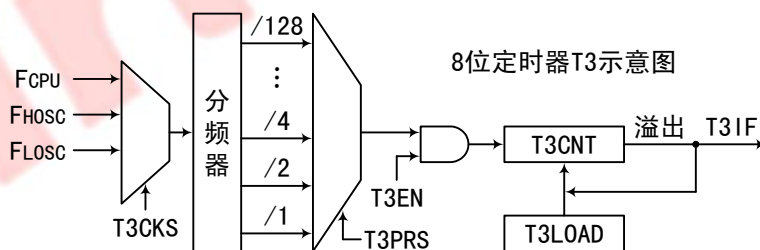
BIT[7:0] **T2LOAD[7:0]** – T2 重载寄存器低 8 位，用于设置 T2 的计数周期

注：定时器重载寄存器的值禁止为 0，否则定时器将无法正常工作。

7.5 定时器 T3

定时器 T3 为 8 位定时器，包含 1 个 8 位递减计数器、可编程预分频器、控制寄存器、8 位重载寄存器。

- ◇ 可通过预分频比设置计数频率，可通过重载寄存器控制计数周期；
- ◇ 支持溢出中断和溢出唤醒功能；



定时器 T3，可通过寄存器位 T3CKS 选择时钟源，通过 T3PRS 选择预分频比，所选时钟源通过预分频器后产生 T3 计数器 T3CNT 的计数时钟。预分频比可选择 1~128 分频，对 T3CNT 的写操作将清零预分频计数器，而预分频比保持不变。

当 T3EN=0 时，T3CNT 保持不变，写重载寄存器 T3LOAD 将立即载入 T3CNT 中；当 T3EN=1 时，T3CNT 递减计数，计数到 0 时产生溢出信号并触发中断，中断标志 T3IF 将被置 1，在下一个时钟 T3 自动将当前 T3LOAD 值载入 T3CNT 中重新开始计数。

定时器 T3 控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	-------	-------	-------	-------	-------	-------	-------	-------

T3CR	T3EN	-	-	T3CKS1	T3CKS0	T3PRS2	T3PRS1	T3PRS0
R/W	R/W	-	-	R/W	R/W	R/W	R/W	R/W
初始值	0	-	-	0	0	0	0	0

BIT[7] **T3EN** – 定时器 T3 使能位

0: 关闭定时器 T3;

1: 开启定时器 T3;

BIT[4:3] **T3CKS[1:0]** – T3 时钟源选择位

T3CKS[1:0]	T3 时钟源
00	FCPU
01	FHOSC
10	FLOSC
11	-

BIT[2:0] **T3PRS[2:0]** – T3 预分频比选择位

T3PRS[2:0]	T3 时钟预分频比
000	1:1
001	1:2
010	1:4
011	1:8
100	1:16
101	1:32
110	1:64
111	1:128

定时器 T3 计数器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T3CNT	T3CNT7	T3CNT6	T3CNT5	T3CNT4	T3CNT3	T3CNT2	T3CNT1	T3CNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] **T3CNT[7:0]** – T3 计数器，为可读写的递减计数器

定时器 T3 重载寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T3LOAD	T3LOAD7	T3LOAD6	T3LOAD5	T3LOAD4	T3LOAD3	T3LOAD2	T3LOAD1	T3LOAD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] **T3LOAD[7:0]** – T3 重载寄存器，用于设置 T3 的计数周期

注：定时器重载寄存器的值禁止为 0，否则定时器将无法正常工作。

8 模数转换器 ADC

8.1 ADC 概述

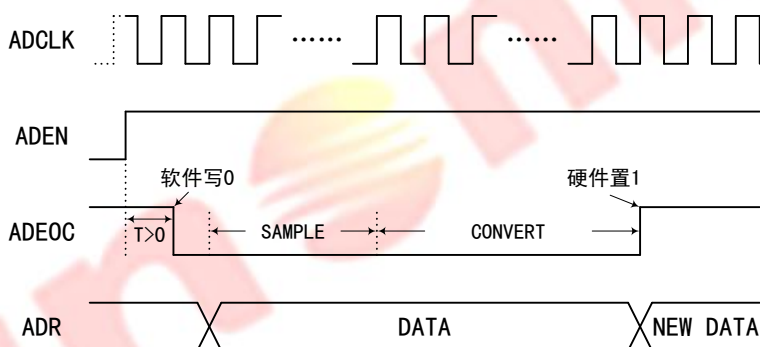
芯片内置的 ADC 模块为 12 位高精度逐次逼近型 ADC。

- ◇ 14 路外部通道：AN0~AN13；2 路内部通道：GND、VDD/4；
- ◇ 参考电压可选：VDD、内部参考电压 V_{IR}（2V）；
- ◇ ADC 时钟：F_{HIRC} 的 8/16/32/64 分频；
- ◇ 支持零点校准；

ADC 模块可通过寄存器位 ADEN 开启，通过 ADCKS 选择转换时钟，通过 ADCHS 选择转换的模拟通道，通过 ADEOC 启动并标识 ADC 的转换状态。当 ADEOC 为 1 时，对其写 0 将启动模数转换，转换完成后结果存入 ADRH/ADRL 中，ADEOC 将自动置 1，同时中断标志 ADIF 置 1 触发 ADC 中断。

采样（SAMPLE）时间可选择 4/8/15 个 ADCLK（即 ADC 时钟周期），转换（CONVERT）时间固定为 12 个 ADCLK，一次 ADC 转换为 16/20/27 个 ADCLK。

ADC 转换时序如下图所示：



注：

- 1、AD 转换过程中或者 ADEN 未使能时，ADRH/ADRL 中的数据未知，应在 AD 转换结束且 ADEN 使能的情况下读取 AD 转换数据；
- 2、若选择内部参考电压 V_{IR}，则需保证 $VDD > (V_{IR} + 0.5V)$ ，否则 V_{IR} 将随之下降；
- 3、使能 ADC 模块、切换参考电压等操作后，需待电路稳定（时间 > 200us）后才能启动 AD 转换；切换输入通道后，受外部输入影响，前两次转换的结果会有误差，建议舍弃；
- 4、AD 转换精度受参考电压精度的影响，且内部参考电压下的转换精度，比外部参考电压略低 2 个 LSB 左右；
- 5、转换时钟越慢、采样时间越长，则越能过滤外部输入的波动，越能保证 AD 转换的精度；

8.2 ADC 操作步骤

模数转换操作步骤：

- (1) 设置相应端口为输入端口，关闭上下拉电阻；
- (2) 通过端口数模控制寄存器，关闭相应端口的数字 I/O 功能；

- (3) 若转换时钟可选，则设置 ADCKS，选取适当的 ADC 转换时钟；
- (4) 若采样时间可选，则设置 ADSPS，选取适当的 ADC 采样时间；
- (5) 若参考电压可选，则设置 ADVRS，选择适当的参考电压；
- (6) 若数据格式可选，则设置 ADRSEL，选择 ADC 结果的数据格式；
- (7) ADEN 置 1，使能 ADC 模块；
- (8) 设置 ADCHS，选择 ADC 转换通道；
- (9) ADEOC 写 0，启动 AD 转换；
- (10) 等待 ADEOC 硬件置 1（或利用 ADC 中断）；
- (11) 读取 ADC 转换结果（ADRH、ADRL）；
- (12) 重复（8）~（11），对不同的通道进行转换或对同一通道进行多次转换；

8.3 ADC 相关寄存器

ADC 控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCR0	ADCHS3	ADCHS2	ADCHS1	ADCHS0	ADVRS1	ADVRS0	ADEOC	ADEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	0	0	1	0

BIT[7:4] ADCHS[3:0] – ADC 模拟通道选择位

ADCHS[3:0]	ADC 模拟通道选择
0000	AN0
0001	AN1
0010	AN2
0011	AN3
0100	AN4
0101	AN5
0110	AN6
0111	AN7
1000	AN8
1001	AN9
1010	AN10
1011	AN11
1100	AN12
1101	AN13
1110	GND
1111	VDD/4

BIT[3:2] ADVRS[1:0] – ADC 参考电压选择位

ADVRS[1:0]	ADC 参考电压选择
00	内部 2.0V

01	-
10	-
11	VDD

BIT[1] ADEOC – ADC 启动位及转换结束标志位
 0: AD 转换过程中, 转换结束后自动置 1;
 1: AD 转换结束, 对 ADEOC 写 0 启动 AD 转换;

BIT[0] ADEN – ADC 使能位
 0: 关闭 ADC;
 1: 开启 ADC;

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCR1	ADRSEL	-	ADCKS1	ADCKS0	-	-	ADSPS1	ADSPS0
R/W	R/W	-	R/W	R/W	-	-	R/W	R/W
初始值	0	-	0	0	-	-	1	1

BIT[7] ADRSEL – ADC 转换数据格式选择位
 0: ADC 转换结果 12 位数据, 高 8 位存入 ADRH[7:0]、低 4 位存入 ADRL[3:0];
 1: ADC 转换结果 12 位数据, 高 4 位存入 ADRH[3:0]、低 8 位存入 ADRL[7:0];

BIT[5:4] ADCKS[1:0] – ADC 转换时钟选择位

ADCKS[1:0]	ADC 转换时钟 F _{ADC}
00	F _{HIRC} /8
01	F _{HIRC} /16
10	F _{HIRC} /32
11	F _{HIRC} /64

BIT[1:0] ADSPS[1:0] – ADC 采样时间选择位

ADSPS[1:0]	ADC 采样时间选择
00	禁用
01	4 个 ADCLK
10	8 个 ADCLK
11	15 个 ADCLK

ADC 转换结果寄存器

ADRSEL=0 时:

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADRH	ADR11	ADR10	ADR9	ADR8	ADR7	ADR6	ADR5	ADR4
R/W	R	R	R	R	R	R	R	R
初始值	X	X	X	X	X	X	X	X

BIT[7:0] ADR[11:4] – ADC 转换结果高 8 位

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADRL	-	-	-	-	ADR3	ADR2	ADR1	ADR0
R/W	-	-	-	-	R	R	R	R
初始值	-	-	-	-	X	X	X	X

BIT[3:0] ADR[3:0] – ADC 转换结果低 4 位

ADRSEL=1 时:

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADRH	-	-	-	-	ADR11	ADR10	ADR9	ADR8
R/W	-	-	-	-	R	R	R	R
初始值	-	-	-	-	X	X	X	X

BIT[3:0] ADR[11:8] – ADC 转换结果高 4 位

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADRL	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
R/W	R	R	R	R	R	R	R	R
初始值	X	X	X	X	X	X	X	X

BIT[7:0] ADR[7:0] – ADC 转换结果低 8 位

ADC 零点偏移修调控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OSADJCR	OSADJEN	-	OSADJTD	OSADJT4	OSADJT3	OSADJT2	OSADJT1	OSADJT0
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	U	-	U	U	U	U	U	U

BIT[7] OSADJEN – ADC 零点偏移修调使能位（复位初始值为出厂设定值）

- 0: 关闭 ADC 零点偏移修调功能;
1: 使能 ADC 零点偏移修调功能;

BIT[5] OSADJTD – ADC 零点偏移修调方向选择位（复位初始值为出厂设定值）

- 0: 负向修调，即根据修调电压减小转换值（转换结果大于理论值时应选择负向修调）;
1: 正向修调，即根据修调电压增加转换值（转换结果小于理论值时应选择正向修调）;

BIT[4:0] OSADJT[4:0] – ADC 零点偏移修调电压选择位（复位初始值为出厂设定值）

OSADJT[4:0]	修调电压 (典型值)
00000	0mV
00001	0.5mV
00010	1.0mV
00011	1.5mV
00100	2.0mV
00101	2.5mV
00110	3.0mV
00111	3.5mV

01000	4.0mV
01001	4.5mV
01010	5.0mV
01011	5.5mV
01100	6.0mV
01101	6.5mV
01110	7.0mV
01111	7.5mV
10000	8.0mV
10001	8.5mV
10010	9.0mV
10011	9.5mV
10100	10.0mV
10101	10.5mV
10110	11.0mV
10111	11.5mV
11000	12.0mV
11001	12.5mV
11010	13.0mV
11011	13.5mV
11100	14.0mV
11101	14.5mV
11110	15.0mV
11111	15.5mV

8.4 ADC 零点偏移修调流程

- (1) 将 ADC 的输入通道选择为内部接地，设置 OSADJEN=1；设置 ADC 时钟、采样时间等；
- (2) 设置 OSADJTD=0、OSADJT[4:0]=00H；进行 ADC 转换；
- (3) 读取 ADC 转换结果：
 - ✧ 如果 ADC 结果为 0，则执行（6）；
 - ✧ 如果 ADC 结果不为 0，则执行（4）；
- (4) OSADJT[4:0]加 1 后进行 ADC 转换；
- (5) 读取 ADC 转换结果：
 - ✧ 如果 ADC 结果为 0，则跳至（10）；
 - ✧ 如果 ADC 结果不为 0，则执行（4），直到 ADC 结果为 0 或 OSADJT[4:0]=1FH 后跳至（10）；
- (6) 设置 OSADJTD=1、OSADJT[4:0]=3FH，进行 ADC 转换；
- (7) 读取 ADC 转换结果：
 - ✧ 如果 ADC 结果为 0，则跳至（10）；
 - ✧ 如果 ADC 结果不为 0，则执行（8）；
- (8) OSADJT[4:0]减 1 后进行 ADC 转换；

-
- (9) 读取 ADC 转换结果:
- ✧ 如果 ADC 结果为 0, 则跳至 (10);
 - ✧ 如果 ADC 结果不为 0, 则执行(8), 直到 ADC 结果为 0 或 OSADJT[4:0]=00H 后跳至(10);
- (10) OSADJTD 及 OSADJT[4:0]的值即为零点偏移最佳修调结果, 修调流程结束, 后续 ADC 工作时直接使用, 不需要再次修调。

9 EEPROM 数据存储器

9.1 EEPROM 概述

芯片内置 64 字节的 EEPROM 数据存储器，支持用户程序在带电工作中实时地读出或写入数据。读写 EEPROM 中的数据时需通过控制寄存器 EECR、数据寄存器 EEDR、地址寄存器 EEAR 和保护寄存器 EEMASK 进行。

EECR 中 EETRIG 位为操作启动标志位，置 1 启动读写操作，完成后自动清 0；EERW 位为读写命令位，为 0 表示读数据、为 1 表示写数据：读数据操作将从 EEAR 对应的 EEPROM 地址中读出数据，保存在 EEDR 中，写数据操作将 EEDR 中的数据写入 EEAR 对应的 EEPROM 地址中。

为防止误触发 EEPROM 读写操作，寄存器 EEMASK 需先写入 5AH 再立即写入 A5H，并在随后 2 个指令周期内将 EETRIG 位置 1，才能执行读写操作。否则 2 个指令周期后 EEMASK 将自动清零，此时 EETRIG 位置 1 不会执行读写操作，也不会自动清 0。

芯片仅支持单字节数据的读写操作，不支持连续地址读写功能，每次都必须通过 EEAR 设置将要访问的 EEPROM 数据的 6 位地址后，才能进行读写操作。

当启动 EEPROM 读写操作后，CPU 将暂停在当前指令，只有等 EEPROM 读写操作完成后，才能继续执行下一条指令。在读写 EEPROM 前需屏蔽中断并清 WDT 计数器，以免系统响应中断或复位而导致读写错误。

注：

- 1、EEPROM 的操作时钟为 CPU 时钟，若 EE 的操作时钟频率过低将导致读写故障，所以 F_{CPU} 低于 EE 操作时钟最低频率时禁止读写 EEPROM（EE 操作时钟最低频率参见特性参数表）；
- 2、与 EEPROM 相关的寄存器 EECR、EEDR、EEAR 和 EEMASK，仅能通过 MOVAR 和 MOVRA 指令进行读写，其他指令的执行结果不确定；

9.2 EEPROM 相关寄存器

EEPROM 控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EECR	EETRIG	EERW	-	-	-	-	-	-
R/W	R/W	R/W	-	-	-	-	-	-
初始值	0	0	-	-	-	-	-	-

BIT[7] EETRIG – EEPROM 读写启动控制位

0：未进行 EEPROM 读写操作，或 EEPROM 读写操作完成后硬件自动清 0；

1：启动 EEPROM 读写操作；

BIT[6] **EERW** – EEPROM 读写命令位

0: 从 EEPROM 中读出数据;

1: 向 EEPROM 中写入数据;

EEPROM 保护寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EEMASK	EEMASK7	EEMASK6	EEMASK5	EEMASK4	EEMASK3	EEMASK2	EEMASK1	EEMASK0
R/W	W	W	W	W	W	W	W	W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **EEMASK[7:0]** – EEPROM 操作保护位, 需先写 5AH 再立即写 A5H, EETRIG 才能置 1

EEPROM 地址寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EEAR	-	-	EEA5	EEA4	EEA3	EEA2	EEA1	EEA0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	-	0	0	0	0	0	0

BIT[5:0] **EEA[5:0]** – EEPROM 读写操作的 6 位地址

EEPROM 数据寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EEDR	EED7	EED6	EED5	EED4	EED3	EED2	EED1	EED0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **EED[7:0]** – EEPROM 读写操作的 8 位数据

9.3 EEPROM 操作示例

例: 向地址为 10H 的 EEPROM 单元写入数据 55H

```

CLRR      EECR          ; 清 EETRIG 位
MOVAI     10H
MOVRA     EEAR          ; 将地址 10H 写入 EEAR
MOVAI     55H
MOVRA     EEDR          ; 将数据 55H 写入 EEDR
BCLR      GIE           ; 屏蔽中断
CLRWDI    WDT           ; 清 WDT
MOVAI     5AH
MOVRA     EEMASK        ; 使能 EE 操作, 第 1 步
MOVAI     A5H
    
```

MOVRA	EEMASK	；使能 EE 操作，第 2 步
MOVAI	COH	
MOVRA	EECR	；启动 EE 写操作，将数据 55H 写入 EEPROM 地址 10H 中
NOP		；防止时序错误，CPU 必须冗余 1 个指令周期
BSET	GIE	；允许中断

例：从地址为 10H 的 EEPROM 单元读出数据（数据存放在 EEDR 中）

CLRR	EECR	；清 EETRIG 位
MOVAI	10H	
MOVRA	EEAR	；将地址 10H 写入 EEAR
BCLR	GIE	；屏蔽中断
CLRWDT		；清 WDT
MOVAI	5AH	
MOVRA	EEMASK	；使能 EE 操作，第 1 步
MOVAI	A5H	
MOVRA	EEMASK	；使能 EE 操作，第 2 步
MOVAI	80H	
MOVRA	EECR	；启动 EE 写操作，从 EEPROM 地址 10H 中读出数据
NOP		；防止时序错误，CPU 必须冗余 1 个指令周期
BSET	GIE	；允许中断

10 中断

本芯片中断源有外部中断（INT0~INT3）、定时器中断（T0~T3）、ADC 转换中断和键盘中断等。可通过中断总使能位 GIE 屏蔽所有中断。

CPU 响应中断的过程如下：

- ✧ CPU 响应中断源触发的中断请求时，自动将当前指令的下一条要执行指令的地址压栈保存，自动清 0 中断总使能位 GIE 以暂停响应后续中断。与复位不同，硬件中断不停止当前指令的执行，而是暂时挂起中断直到当前指令执行完成。
- ✧ CPU 响应中断后，程序跳到中断向量地址开始执行中断服务程序，中断服务程序应先保存累加器 A 和状态寄存器 PFLAG，然后处理被触发的中断。
- ✧ 中断服务程序处理完中断后，应先恢复累加器 A 和状态寄存器 PFLAG，然后执行 RETIE 返回主程序。此时芯片将自动恢复 GIE 为 1，然后从堆栈取出 PC 值，从中断产生时当前指令的下一条指令继续执行。

本芯片的中断向量地址是 0008H。

10.1 外部中断

本芯片有 4 路外部中断源，INT0/INT1 可选择上升沿、下降沿或电平变化等触发方式，INT2/INT3 固定为下降沿触发。外部中断触发时，中断标志（INT0IF、INT1IF、INT2IF、INT3IF）将被置 1，若中断总使能位 GIE 为 1 且外部中断使能位（INT0IE、INT1IE、INT2IE、INT3IE）为 1，则产生外部中断。

10.2 定时器中断

定时器 T0、T1、T2、T3 在计数溢出时触发定时器中断，中断标志（T0IF、T1IF、T2IF、T3IF）将被置 1，若中断总使能位 GIE 为 1 且定时器中断使能位（T0IE、T1IE、T2IE、T3IE）为 1，则产生定时器中断。

10.3 ADC 中断

ADC 转换完成后触发 ADC 中断，中断标志（ADIF）将被置 1，若中断总使能位 GIE 为 1 且 ADC 中断使能位（ADIE）为 1，则产生 ADC 中断。

10.4 键盘中断

本芯片有 8 路键盘中断源，可通过寄存器控制位单独使能或屏蔽，任意一路使能的中断源的输入电平发生变化时，触发键盘中断，中断标志（KBIF）将被置 1，若中断总使能位 GIE 为 1 且键盘中断使能位（KBIE）为 1，则产生键盘中断。

键盘中断控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1KBCR	P17KE	P16KE	P15KE	P14KE	P13KE	P12KE	P11KE	P10KE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **P1nKE** – P1n 口键盘中断使能位（n=7-0）

- 0: 屏蔽端口键盘中断功能；
- 1: 使能端口键盘中断功能；

10.5 中断相关寄存器

中断使能寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTE	-	ADIE	-	KBIE	INT1IE	INT0IE	T1IE	T0IE
R/W	-	R/W	-	R/W	R/W	R/W	R/W	R/W
初始值	-	0	-	0	0	0	0	0

BIT[6] **ADIE** – ADC 中断使能位

- 0: 屏蔽 ADC 中断；
- 1: 使能 ADC 中断；

BIT[4] **KBIE** – 键盘中断使能位

- 0: 屏蔽键盘中断；
- 1: 使能键盘中断；

BIT[3] **INT1IE** – INT1 中断使能位

- 0: 屏蔽 INT1 中断；
- 1: 使能 INT1 中断；

BIT[2] **INT0IE** – INT0 中断使能位

- 0: 屏蔽 INT0 中断；
- 1: 使能 INT0 中断；

BIT[1] **T1IE** – 定时器 T1 中断使能位

- 0: 屏蔽定时器 T1 中断；
- 1: 使能定时器 T1 中断；

BIT[0] **T0IE** – 定时器 T0 中断使能位

0: 屏蔽定时器 T0 中断;

1: 使能定时器 T0 中断;

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTE1	-	-	-	-	INT3IE	INT2IE	T3IE	T2IE
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	0	0	0	0

BIT[3] **INT3IE** – INT3 中断使能位

0: 屏蔽 INT3 中断;

1: 使能 INT3 中断;

BIT[2] **INT2IE** – INT2 中断使能位

0: 屏蔽 INT2 中断;

1: 使能 INT2 中断;

BIT[1] **T3IE** – 定时器 T3 中断使能位

0: 屏蔽定时器 T3 中断;

1: 使能定时器 T3 中断;

BIT[0] **T2IE** – 定时器 T2 中断使能位

0: 屏蔽定时器 T2 中断;

1: 使能定时器 T2 中断;

中断标志寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTF	-	ADIF	-	KBIF	INT1IF	INT0IF	T1IF	T0IF
R/W	-	R/W	-	R/W	R/W	R/W	R/W	R/W
初始值	-	0	-	0	0	0	0	0

BIT[6] **ADIF** – ADC 中断标志位

0: 未触发 ADC 中断;

1: 已触发 ADC 中断, 需软件清零;

BIT[4] **KBIF** – 键盘中断标志位

0: 未触发键盘中断;

1: 已触发键盘中断, 需软件清零;

BIT[3] **INT1IF** – INT1 中断标志位

0: 未触发 INT1 中断;

1: 已触发 INT1 中断, 需软件清零;

- BIT[2] **INT0IF** – INT0 中断标志位
0: 未触发 INT0 中断;
1: 已触发 INT0 中断, 需软件清零;
- BIT[1] **T1IF** – 定时器 T1 中断标志位
0: 未触发定时器 T1 中断;
1: 已触发定时器 T1 中断, 需软件清零;
- BIT[0] **T0IF** – 定时器 T0 中断标志位
0: 未触发定时器 T0 中断;
1: 已触发定时器 T0 中断, 需软件清零;

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTF1	-	-	-	-	INT3IF	INT2IF	T3IF	T2IF
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	0	0	0	0

- BIT[3] **INT3IF** – INT3 中断标志位
0: 未触发 INT3 中断;
1: 已触发 INT3 中断, 需软件清零;
- BIT[2] **INT2IF** – INT2 中断标志位
0: 未触发 INT2 中断;
1: 已触发 INT2 中断, 需软件清零;
- BIT[1] **T3IF** – 定时器 T3 中断标志位
0: 未触发定时器 T3 中断;
1: 已触发定时器 T3 中断, 需软件清零;
- BIT[0] **T2IF** – 定时器 T2 中断标志位
0: 未触发定时器 T2 中断;
1: 已触发定时器 T2 中断, 需软件清零;

11 特性曲线

注：本章节所列特性曲线图为抽样实测数据，仅作为应用参考，部分数据因生产工艺偏差，可能与实际芯片不符；为保证芯片能正常工作，请确保其工作条件符合电气特性参数说明。

11.1 I/O 特性

11.2 功耗特性

11.3 模拟电路特性

12 封装尺寸

12.1 SOP24

12.2 SOP20

12.3 DIP20

12.4 SOP16

12.5 DIP16

12.6 TSSOP20

13 修订记录

版本	修订日期	修订内容
V0.0	2018-10-11	讨论初稿
V0.0	2018-10-17	增加 TSSOP20 封装