



苏州华芯微电子股份有限公司



HuaXin Micro-electronics Co., Ltd

HS26F2728 单片机
用户手册
V2.1



目 录

1. 概述.....	1
2. 特征.....	1
3. 管脚分配.....	2
3.1 管脚图.....	2
3.2 管脚说明.....	2
4. 系统框图及存储结构.....	6
4.1 系统框图.....	6
4.2 ROM 程序存储器.....	6
4.3 数据存储器.....	7
5. 功能描述.....	12
5.1 操作寄存器.....	12
5.1.1 H/L（间接寻址寄存器）.....	12
5.1.2 Y/Z（间接寻址寄存器）.....	12
5.1.3 R 寄存器.....	13
5.1.4 X（X 寄存器）.....	13
5.1.5 W（W 寄存器）.....	13
5.1.6 PC（程序计数器）与堆栈.....	14
5.1.7 PFLAG（状态寄存器）.....	15
5.1.8 RBANK（RAM 存储区选择寄存器）.....	15
5.1.9 WDTC（看门狗控制寄存器）.....	16
5.1.10 WDTR 寄存器.....	16
5.1.11 LVDC（控制寄存器）.....	17
5.1.12 OSCM 寄存器.....	17
5.1.13 REMC.....	18
5.1.14 LFSEL.....	18
5.1.15 SMTS.....	18
5.1.16 TIRCF.....	19
5.2 TC0（16 定时/计数器寄存器）.....	19
5.2.1 概述.....	20
5.2.2 TC0CL.....	20
5.2.3 TC0M.....	20
5.2.4 PWM 高电平占空比寄存器.....	21
5.2.5 TC0R.....	21
5.2.6 TC0M1.....	22
5.2.7 PWM 波形.....	22
5.3 T0（定时/计数器寄存器）.....	22
5.3.1 概述.....	23
5.3.2 T0M 寄存器.....	23
5.3.3 T0C 寄存器.....	23
5.3.4 T0M1 寄存器.....	24
5.4 T2（定时/计数器寄存器）.....	24
5.4.1 概述.....	24
5.4.2 TL0C 寄存器.....	24
5.4.3 TL0M.....	25
5.5 带死区互补的 12 位 PWM（TM1）.....	25
5.5.1 TM1 特性.....	25



5.5.2	TM1 输出模式.....	25
5.5.3	TM1 相关寄存器.....	26
5.6	带死区互补的 16 位 PWM (TM2)	29
5.6.1	TM2 特性.....	29
5.6.2	TM2 输出模式.....	29
5.6.3	TM2 相关寄存器.....	29
5.7	中断.....	33
5.7.1	概述.....	33
5.7.2	中断控制寄存器 0.....	33
5.7.3	中断控制寄存器 1.....	34
5.7.3	INTRQ0.....	34
5.7.4	INTRQ1.....	35
5.7.5	PEDGE.....	35
5.7.6	STKP.....	36
5.8	EEPROM.....	37
5.8.1	EECON1 (EE 控制寄存器)	37
5.8.2	EADRL (EE 地址低位)	37
5.8.3	EDATL (EE 数据低位)	37
5.8.4	EEPEOM 操作示例.....	38
6.	I/O 口.....	40
6.1	端口模式控制寄存器.....	40
6.2	端口数据寄存器.....	40
6.3	端口上拉控制寄存器.....	40
6.4	端口下拉控制寄存器.....	40
6.5	端口唤醒控制寄存器.....	41
6.6	端口开漏输出控制寄存器.....	41
6.7	端口扫描模式控制寄存器.....	41
6.8	端口驱动控制寄存器.....	41
7.	14 通道 ADC.....	43
7.1	概述.....	43
7.2	ADM (ADC 模式寄存器)	43
7.3	ADB (ADC 数据寄存器)	44
7.4	VREFH (ADC 参考电压寄存器)	45
7.5	ADC 操作说明和注意事项.....	46
7.5.1	ADC 信号格式.....	46
7.5.2	AD 转换时间.....	46
7.5.3	ADC 引脚配置.....	47
7.6	P4 口/P1 口 ADC 共用引脚.....	48
7.7	ADT (ADC 结果微调寄存器)	48
7.8	ADC 应用电路.....	49
8.	端口比较器、运算放大器	51
8.1	端口比较器相关寄存器.....	51
8.1.1	CMPCON(比较器控制寄存器).....	51
8.1.2	CMPM(比较器负端输入选择寄存器).....	51
8.2	运算放大器相关寄存器.....	52
8.2.1	OPAM.....	52
8.2.2	OPAM1.....	53
8.2.3	OPAMC.....	53



8.2.4	OPAMC1	54
8.3	应用原理与应用举例	55
8.3.1	应用原理	55
8.3.2	应用举例	56
9.	通讯接口	57
9.1	增强型异步收发 UART	57
9.1.1	TXSTA: 发送状态和控制寄存器	57
9.1.2	RCSTA: 接收状态和控制寄存器	58
9.1.3	BAUDCTL: UART 中断寄存器	58
9.1.4	UART 异步发送器	59
9.1.5	UART 异步接收器	61
9.1.6	UART 波特率发生器 (BRG)	63
9.2	串行外部设别接口 SPI	64
9.2.1	SIM (SIM 控制器)	64
9.2.2	SIMD (SIM 的缓冲器)	65
9.2.3	SIMC0 (SIM 控制器 0)	65
9.2.4	SIMC1 (SIM 控制器 1)	65
9.2.5	SPI 通信	66
9.3	INTFACES (通信管脚控制器)	67
10.	系统时钟	69
10.1	概述	69
10.2	指令周期 Fcpu	69
11.	系统工作模式	70
11.1	概述	70
11.2	普通模式	71
11.3	低速模式	71
11.4	睡眠模式	71
11.5	绿色模式	72
11.6	系统唤醒	72
11.6.1	概述	72
11.6.2	唤醒时间	72
12.	复位	74
12.1	概述	74
12.2	上电复位	74
12.3	看门狗复位	75
12.4	REST 脚复位 (P04)	75
13.	指令集	76
14.	电气参数	78
14.1	极限参数	78
14.2	DC 特性 (TEMP=25℃、VDD=5V、全指令周期)	78
15.	封装及尺寸	79
15.1	TSSOP20 封装图及尺寸	79
15.2	SOP16 封装图及尺寸	80
16.	修正记录	81



HS26F2728

用户手册

Ver2.1

1. 概述

HS26F2728 是一款采用低耗高速 CMOS 工艺制造的全集成 AD 型芯片，它包含一个 8K Word ROM，256 Byte 的 EEPROM，内部集成了一个 12 位 ADC，且有着丰富的通讯接口。IO 端口既可以配置为普通的 IO 口，同时可通过软件配置为 AD 采样口，在不增加更多 IO 口的基础上实现更加丰富的功能。

2. 特征

- 工作电压 2.0~5.5v(14M 主频)
- 工作温度范围：-20℃~80℃
- 8K Word ROM
- 256 Byte EEPROM
- 384 Byte RAM
- 最大 18 个 IO 口
- 最大 11 通道的 12 位 AD
- 所有端口驱动可选
- 6 路 12 位 PWM，可配置为 2 组带死区控制的 PWM
- 1 个 UART 通讯口
- 1 个 SPI 通讯口
- 集成端口比较器，可比较端口电压
- 集成一个增益可调运放，可放大端口信号
- 可配置内部高速，内部低速，外部高速，外部低速的系统时钟，
- 内部高速最大 28Mhz
- 内置 WDT 定时器
- 包含端口变化唤醒及定时器溢出唤醒
- 包括高速，低速，绿色，休眠四种模式
- 包含上电复位，掉电复位（可选）



3. 管脚分配

3.1 管脚图

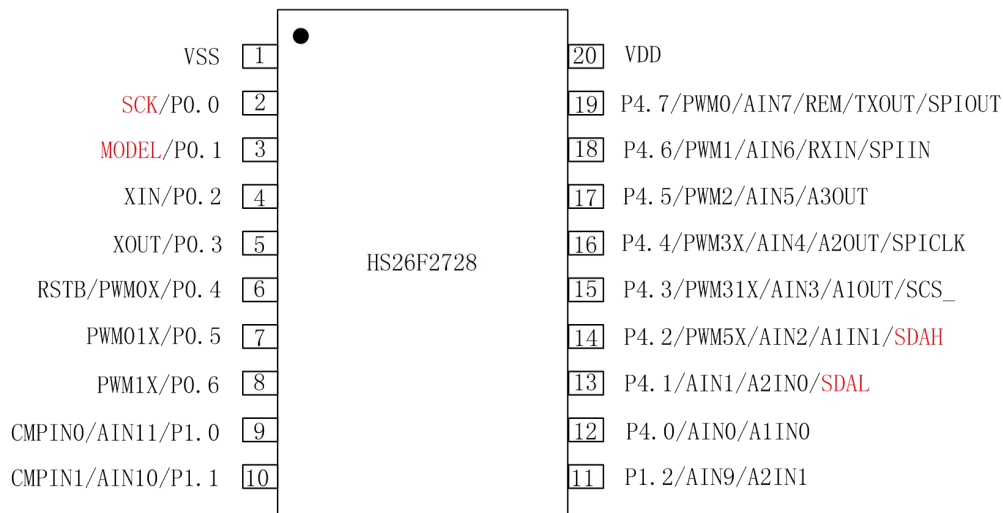


图 3.1 TSSOP20 管脚分配

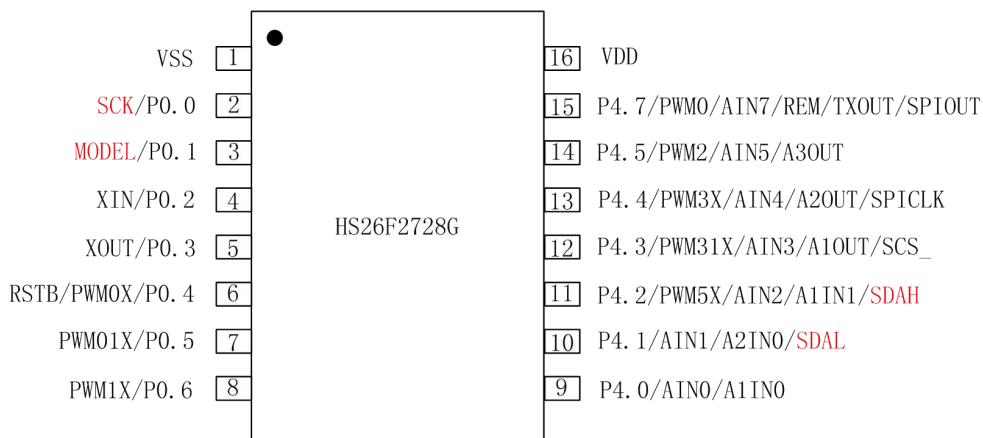


图 3.2 SOP16 管脚分配

3.2 管脚说明

表 3.1 管脚说明

管脚名称	I/O	说 明
VDD	I	电源正端（+）输入端；
VSS	I	电源负端（-）输入端；
P4.7	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 3 级可选 编码输出端（可通过寄存器设置） 可配置 AD 采样脚



		可配置 8 位 PWM 输出口 可配置 UART 数据输出口 可配置 SPI 数据输出口
P4.6	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 3 级可选 可配置 AD 采样脚 可配置 8 位 PWM 输出口 可配置 UART 数据输入口 可配置 SPI 数据输入口
P4.5	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 3 级可选 可配置 AD 采样脚 可配置运放 A3 输出脚 可配置 8 位 PWM 输出口
P4.4	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 3 级可选 可配置 AD 采样脚 可配置运放 A2 输出脚 可配置 16 位 PWM 输出口 可配置 SPI 时钟脚
P4.3	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 3 级可选 可配置 AD 采样脚 可配置运放 A1 输出脚 可配置 16 位 PWM 输出口 可配置 SCS_信号
P4.2	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 3 级可选 可配置 AD 采样脚 可配置运放 A1 负端输入脚 可配置 16 位 PWM 输出口 在线烧写 SDAH 脚
P4.1	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出



		驱动 3 级可选 可配置 AD 采样脚 可配置运放 A2 正端输入脚 在线烧写 SDAL 脚
P4.0	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 3 级可选 可配置 AD 采样脚 可配置运放 A1 正端输入脚
P1.2	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 3 级可选 可配置 AD 采样脚 可配置运放 A2 负端输入脚
P1.1	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 3 级可选 可配置 AD 采样脚 可配置端口比较器负端输入脚
P1.0	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 3 级可选 可配置 AD 采样脚 可配置端口比较器正端输入脚
P0.6~P0.5	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 3 级可选 可配置 12 位 PWM 输出口
P0.4	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 3 级可选 外部复位脚 可配置 12 位 PWM 输出口
P0.3~P0.2	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 3 级可选 晶振输出、输入脚

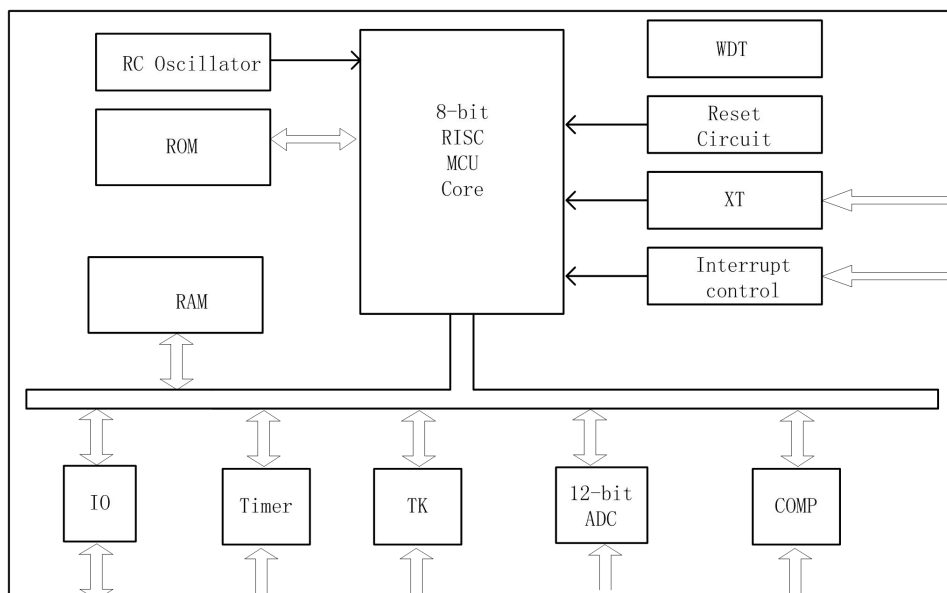


P0.1~P0.0	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 3 级可选 在线烧写 MODEL/SCK 脚
-----------	-----	--



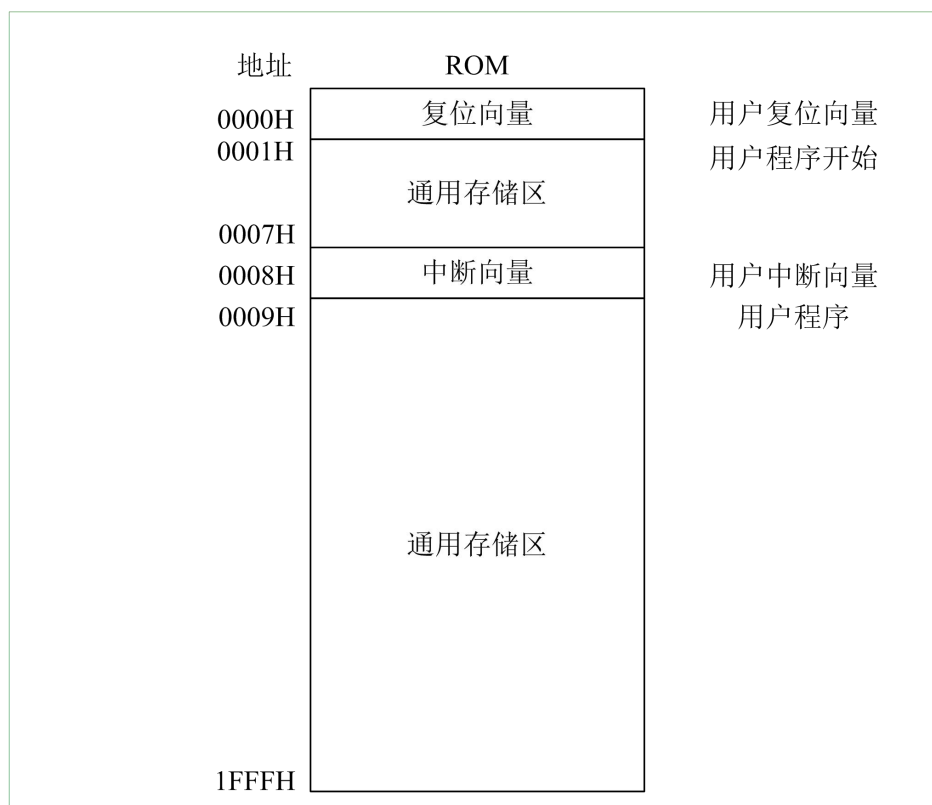
4. 系统框图及存储结构

4.1 系统框图



4.2 ROM 程序存储器

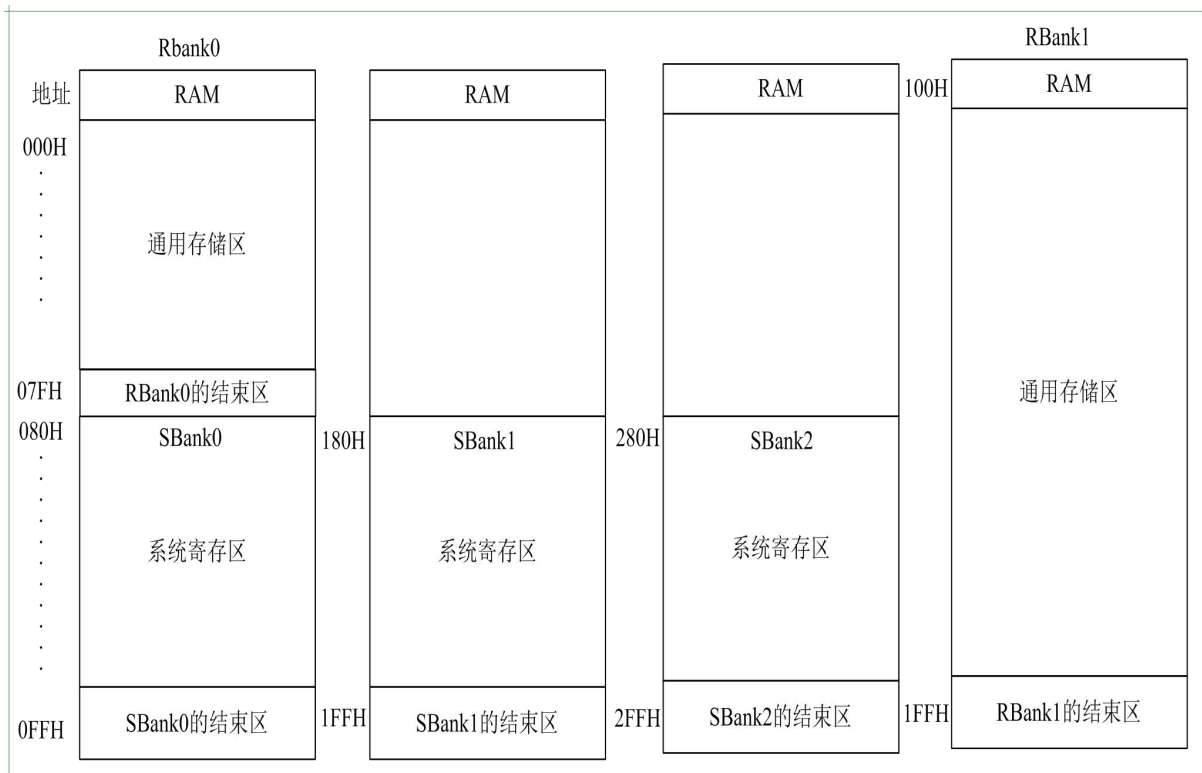
PC 为 13 位,支持 JMP 和 CALL 全地址跳转, 复位地址 0000H,硬件中断入口 0008H。



程序存储器结构图



4.3 数据存储



特殊寄存器列表:

地址8X/9X/AX/FX/EF不受BANK影响																
8	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
L	H	R	Z	Y	X	PFLAG	RBANK	W0	W1	W2	W3	W4	W5	W6	W7	
@HL	@YZ		PCL	PCH	OSCM	WDTR	INTRQ0	INTRQ1	INTEN0	INTEN1	P0W			P4W	PEDGE	
P0M	P1M		P4M		P0	P1	P4		P0UR				P4UR		REMC	
					PWM0DL	PWM0DH	PWM0PL	PWM0PH	PWM0EN	PWM0C	PWM0TL	PWM0TH	PWM0DT	PWM0DTH	PWMPSC	
							ADPCON	ADM	ADB	ADR	ADT	WDT		VREFH	LVDC	
P0PD			P4PD													
P0OD			P4OD		SMTS	TC0D2	TC0M1	PWM2DL	PWM2DH	TC0M	TC0R	TC0CL	TC0CH		STKP	
STK7L	STK7H	STK6L	STK6H	STK5L	STK5H	STK4L	STK4H	STK3L	STK3H	STK2L	STK2H	STK1L	STK1H	STK0L	STK0H	
BANK1																
18																
19																
1A																
1B				PWM5DL	PWM5DH	PWM3DL	PWM3DH	PWM3PL	PWM3PH	PWM3EN	PWM3C	PWM3TL	PWM3TH	PWM3DT	PWM3DTH	PWM3PSC
P0SC	P4SC	P0IS			P4IS		TL0M	TL0C	T0M	T0L	T0H	T0M1			TIRCF	
1D																
1E														ITFACES		
1F																
BANK2																
28																
29																
2A																
2B			LFSEL													
2C																
2D																
2E	EADRL	EECON1	EDATL	CMPC	CMPM	OPAM	OPAM1	OPAMC	OPAMC1							
2F																



特殊功能寄存器地址:

BANK0:

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
80H(r/w)	L	间接寻址寄存器							
81H(r/w)	H	间接寻址寄存器							
82H(r/w)	R	ACC 高8位缓存							
83H(r/w)	Z	间接寻址寄存器							
84H(r/w)	Y	间接寻址寄存器							
85H(r/w)	X	工作寄存器							
86H(r/w)	PFLAG	/T0	/PD	-	-	-	C	DC	Z
87H(r/w)	RBANK	SBANKS2	SBANKS1	-	-	-	-	-	RBANKS
88H(r/w)	W0	通用工作寄存器使用							
89H(r/w)	W1	通用工作寄存器使用							
8AH(r/w)	W2	通用工作寄存器使用							
8BH(r/w)	W3	通用工作寄存器使用							
8CH(r/w)	W4	通用工作寄存器使用							
8DH(r/w)	W5	通用工作寄存器使用							
8EH(r/w)	W6	通用工作寄存器使用							
8FH(r/w)	W7	通用工作寄存器使用							
90H(r/w)	@HL	间接寻址寄存器							
91H(r/w)	@YZ	间接寻址寄存器							
93H(r/w)	PCL	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
94H(r/w)	PCH	-	-	-	PC12	PC11	PC10	PC9	PC8
95H(r/w)	OSCM	LRC32K	LRC3XT	-	CPUM1	CPUM0	CLKMD	STPHX	14M32KSEL
96H(r/w)	WDTR	看门狗清零寄存器							
97H(r/w)	INTRQ0	PBIF	T2IF	TC0IF	T0IF	ADCIF	INT2IF	INT1IF	INT0IF
98H(r/w)	INTRQ1	SPIIF	-	-	CMPIF	-	-	TM2IF	TM1IF
99H(r/w)	INTEN0	PBIE	T2IE	TC0IE	T0IE	ADCIE	INT2IE	INT1IE	INT0IE
9AH(r/w)	INTEN1	SPIIE	-	-	CMPIE	-	-	TM2IE	TM1IE
9BH(r/w)	P0W	-	P06W	P05W	P04W	P03W	P02W	P01W	P00W
9CH(r/w)	P1W	-	-	-	-	-	P12W	P11W	P10W
9EH(r/w)	P4W	P47W	P46W	P45W	P44W	P43W	P42W	P41W	P40W
9FH(r/w)	PEDGE	-	-	-	-	-	-	PG1	PG0
0A0H(r/w)	P0M	-	P06M	P05M	P04M	P03M	P02M	P01M	P00M
0A1H(r/w)	P1M	-	-	-	-	-	P12M	P11M	P10M
0A3H(r/w)	P4M	P47M	P46M	P45M	P44M	P43M	P42M	P41M	P40M
0A5H(r/w)	P0	-	P06	P05	P04	P03	P02	P01	P00
0A6H(r/w)	P1	-	-	-	-	-	P12	P11	P10
0A8H(r/w)	P4	P47	P46	P45	P44	P43	P42	P41	P40
0AAH(r/w)	P0UR	-	P06UR	P05UR	P04UR	P03UR	P02UR	P01UR	P00UR
0ABH(r/w)	P1UR	-	-	-	-	-	P12UR	P11UR	P10UR
0ADH(r/w)	P4UR	P47UR	P46UR	P45UR	P44UR	P43UR	P42UR	P41UR	P40UR



0AFH(r/w)	REMC	IE_sel	D_sel	ELE_en	Rem_sel4	Rem_sel3	Rem_sel2	Rem_sel1	Rem_sel0
0B5H(r/w)	PWM0DL	PWM0DL[7:0]							
0B6H(r/w)	PWM0DH	-	-	-	-	PWM0DH[3:0]			
0B7H(r/w)	PWM0PL	PWM0PL[7:0]							
0B8H(r/w)	PWM0PH	-	-	-	-	PWM0PH[3:0]			
0B9H(r/w)	PWM0EN	TM1EN	-	TM1SEL8	PWM1_O EN	PWM0M	PWM01_OE N	PWM0_O EN	PWM0_E N
0BAH(r/w)	PWM0C	CK0[3]	CK0[2]	-	-	PWM0SL[1:0]		CK0[1]	CK0[0]
0BBH(r/w)	PWM0TL	PWM0TL[7:0]							
0BCH(r/w)	PWM0TH	-	-	-	-	PWM0TH[3:0]			
0BDH(r/w)	PWM0DTL	PWM0DTL[7:0]							
0BEH(r/w)	PWM0DTH	-	-	-	-	PWM0DTH[3:0]			
0BFH(r/w)	PWMPSC	-	-	-	-	-	PWM1S0	PWM01S0	PWM0S0
0C5H(r/w)	ADPCON1	-	-	-	-	ADPC13	ADPC12	ADPC11	-
0C7H(r/w)	ADPCON	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0
0C8H(r/w)	ADM	ADENB	ADS	EOC	GCHS	CHS3	CHS2	CHS1	CHS0
0C9H(r/w)	ADB	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4
0CAH(r/w)	ADR	-	ADCKS1	-	ADCKS0	ADB3	ADB2	ADB1	ADB0
0CBH(r/w)	ADT	-	-	-	ADTS<0>	ADT<3>	ADT<2>	ADT<1>	ADT<0>
0CCH(r/w)	WDTC	WDT_EN	WDT_pre2	WDT_pre1	WDT_pre0	-	-	-	-
0CEH(r/w)	VREFH	EVHENB	-	-	-	-	-	VHS1	VHS0
0CFH(r/w)	LVDC	LVDTE	-	-	-	-	-	-	-
0D0H(r/w)	P0PD	-	P06PD	P05PD	P04PD	P03PD	P02PD	P01PD	P00PD
0D1H(r/w)	P1PD	-	-	-	-	-	P12PD	P11PD	P10PD
0D3H(r/w)	P4PD	P47PD	P46PD	P45PD	P44PD	P43PD	P42PD	P41PD	P40PD
0E0H(r/w)	P0OD	-	P06OD	P05OD	P04OD	P03OD	P02OD	P01OD	P00OD
0E1H(r/w)	P1OD	-	-	-	-	-	P12OD	P11OD	P10OD
0E3H(r/w)	P4OD	P47OD	P46OD	P45OD	P44OD	P43OD	P42OD	P41OD	P40OD
0E5H(r/w)	SMTS	-	-	-	-	SMTS4	-	SMTS1	SMTS0
0E6H(r/w)	TC0D2	PWM2 高电平占空比控制寄存器							
0E7H(r/w)	TC0M1	PWM2S1	PWM1S1	PWM0S1	-	-	-	PWM2EN	PWM1EN
0E8H(r/w)	PWM2DL	PWM2DL[7:0]							
0E9H(r/w)	PWM2DH	-	-	-	-	PWM2DH[3:0]			
0EAH(r/w)	TC0M	TC0TR	TC0S2	TC0S1	TC0S0	TCCKS	INT0CK	-	PWM0
0EBH(r/w)	TC0R	自动重装寄存器							
0ECH(r/w)	TC0CL	TC0 的定时寄存器的低 8 位							
0EDH(r/w)	TC0CH	PWM0 高电平占空比控制寄存器/TC0 定时寄存器的高 8 位							
0EEH(r/w)	TC0D1	PWM1 高电平占空比控制寄存器							
0EFH(r/w)	STKP	GIE	-	-	-	-	STKPB2	STKPB1	STKPB0
0F0H(r/w)	STKP7L	S7PC7	S7PC6	S7PC5	S7PC4	S7PC3	S7PC2	S7PC1	S7PC0
0F1H(r/w)	STKP7H	-	-	-	S7PC12	S7PC11	S7PC10	S7PC9	S7PC8



0F2H(r/w)	STKP6L	S6PC7	S6PC6	S6PC5	S6PC4	S6PC3	S6PC2	S6PC1	S6PC0
0F3H(r/w)	STKP6H	-	-	-	S6PC12	S6PC11	S6PC10	S6PC9	S6PC8
0F4H(r/w)	STKP5L	S5PC7	S5PC6	S5PC5	S5PC4	S5PC3	S5PC2	S5PC1	S5PC0
0F5H(r/w)	STKP5H	-	-	-	S5PC12	S5PC11	S5PC10	S5PC9	S5PC8
0F6H(r/w)	STKP4L	S4PC7	S4PC6	S4PC5	S4PC4	S4PC3	S4PC2	S4PC1	S4PC0
0F7H(r/w)	STKP4H	-	-	-	S4PC12	S4PC11	S4PC10	S4PC9	S4PC8
0F8H(r/w)	STKP3L	S3PC7	S3PC6	S3PC5	S3PC4	S3PC3	S3PC2	S3PC1	S3PC0
0F9H(r/w)	STKP3H	-	-	-	S3PC12	S3PC11	S3PC10	S3PC9	S3PC8
0FAH(r/w)	STKP2L	S2PC7	S2PC6	S2PC5	S2PC4	S2PC3	S2PC2	S2PC1	S2PC0
0FBH(r/w)	STKP2H	-	-	-	S2PC12	S2PC11	S2PC10	S2PC9	S2PC8
0FCH(r/w)	STKP1L	S1PC7	S1PC6	S1PC5	S1PC4	S1PC3	S1PC2	S1PC1	S1PC0
0FDH(r/w)	STKP1H	-	-	-	S1PC12	S1PC11	S1PC10	S1PC9	S1PC8
0FEH(r/w)	STKP0L	S0PC7	S0PC6	S0PC5	S0PC4	S0PC3	S0PC2	S0PC1	S0PC0
0FFH(r/w)	STKP0H	-	-	-	S0PC12	S0PC11	S0PC10	S0PC9	S0PC8

BANK1:

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
1B3H(r/w)	PWM5DL	PWM5DL[7:0]							
1B4H(r/w)	PWM5DH	PWM5DH[7:0]							
1B5H(r/w)	PWM3DL	PWM3DL[7:0]							
1B6H(r/w)	PWM3DH	PWM3DH[7:0]							
1B7H(r/w)	PWM3PL	PWM3PL[7:0]							
1B8H(r/w)	PWM3PH	PWM3PH[7:0]							
1B9H(r/w)	PWM3EN	TM2EN	TM2SEL 16	TM2SEL8	PWM5_O EN	PWM3M	PWM31_OE N	PWM3_OE N	PWM3_E N
1BAH(r/w)	PWM3C	CK3[3]	CK3[2]	-	-	PWM3SL[1:0]		CK3[1]	CK3[0]
1BBH(r/w)	PWM3TL	PWM3TL[7:0]							
1BCH(r/w)	PWM3TH	PWM3TH[7:0]							
1BDH(r/w)	PWM3DTL	PWM3DTL[7:0]							
1BEH(r/w)	PWM3DTH	PWM3DTH[7:0]							
1BFH(r/w)	PWM3PSC	-	-	-	-	-	PWM5S	PWM31S	PWM3S
1C0H(r/w)	P0SC	-	P06SC	P05SC	P04SC	P03SC	P02SC	P01SC	P00SC
1C1H(r/w)	P4SC	P47SC	P46SC	P45SC	P44SC	P43SC	P42SC	P41SC	P40SC
1C2H(r/w)	P0IS	-	-	-	-	P03IS	P02IS	P01IS	P00IS
1C3H(r/w)	P1IS	-	-	-	-	P13IS	P12IS	P11IS	P10IS
1C5H(r/w)	P4IS	-	-	-	-	P43IS	P42IS	P41IS	P40IS
1C7H(r/w)	TL0M	T2TR	T2WC	TP2S2	TP2S1	TP2S0	-	-	-
1C8H(r/w)	TL0C	T2 的定时寄存器的低 8 位							
1C9H(r/w)	T0M	T0TR	TP0S2	TP0S1	TP0S0	TP0SEL	T0CK	-	TC0GN
1CAH(r/w)	T0L	T0 的定时寄存器的低 8 位							
1CBH(r/w)	T0H	T0 的定时寄存器的高 8 位							
1CCH(r/w)	T0M1	-	-	-	PWMW2	PWMW1	INT0W	T0OVS	T0GN



1CEH(r/w)	TIRCF	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
1E0H(r/w)	RCSTA	SPEN	RX9	SREN	CREN	-	FERR	OERR	RX9D
1E1H(r/w)	TXREG	USRT 发送数据寄存器							
1E2H(r/w)	RCREG	UART 接收数据寄存器							
1E3H(r/w)	TXSTA	-	TX9	TXEN	SYNC	-	BRGH	TRMT	TX9D
1E4H(r/w)	SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0
1E5H(r/w)	SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8
1E6H(r/w)	BAUDCTL	-	-	IRXEN	ITXEN	-	-	IFRX	IFTX
1E7H(r/w)	SIMC0	SIMM2	SIMM1	SIMM0	-	-	-	SPIEN	-
1E8H(r/w)	SIMC1	-	-	CKEG	CKP	MLS	SSEN	-	TRF
1EAH(r/w)	SIMD	SPI 缓冲器							
1EDH(r/w)	INTFACES	-	SPIIN	SCS_	SPICLK	-	-	TXOUT	RXIN

BANK2:

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
2B2H(r/w)	LFSEL	-	-	-	-	-	LFSEL2	LFSEL1	LFSEL0
2E0H(r/w)	EADRL	EE 地址低位							
2E1H(r/w)	EECON1	-	TIMEEE	-	-	ERSALLEE	ERASEEE	PROGEE	READEE
2E2H(r/w)	EDATL	EE 数据低位							
2E3H(r/w)	CMPCON	CMPEN	CMPIF	FTCKS	CODEB	CPEN+	CVEN+	COPOL	-
2E4H(r/w)	CMPM	COMSEL	CPINS1	CPINS0	XRVSEL	VRS3	VRS2	VRS1	VRS0
2E5H(r/w)	OPAM	OPAM7	OPAM6	OPAM5	OPAM4	OPAM3	OPAM2	OPAM1	OPAM0
2E6H(r/w)	OPAM1	OPAM17	-	-	-	OPAM13	OPAM12	OPAM11	OPAM10
2E7H(r/w)	OPAMC	OPAMC7	LVHSEN	AP1PS	AP1PS0	AP1OSP	AP1OS	AP1NS0	AP1NS
2E8H(r/w)	OPAMC1	OPAMC17	LVHSEN1	AP2PS	AP2PS0	AP2OSP	AP2OS	AP2NS0	AP2NS



5. 功能描述

5.1 操作寄存器

5.1.1 H/L（间接寻址寄存器）

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
80H(R/W)	L	1、通用工作寄存器； 2、RAM 数据寻址指针@HL（90H）。							
81H(R/W)	H								

例：用 H、L 作为数据指针，访问bank0 中020H 处的内容。

B0MOV H, #00H

B0MOV L, #20H

B0MOV A, @HL

例：对bank0中的数据进行清零处理。

CLR H ; H = 0, 指向bank 0。

B0MOV L, #7FH ; L = 7FH。

CLR_HL_BUF:

CLR @HL ; @HL 清零。

DECMS L ; L - 1, 如果 L = 0, 程序结束。

JMP CLR_HL_BUF

CLR @HL

END_CLR:

...

...

5.1.2 Y/Z（间接寻址寄存器）

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
83H(R/W)	Z	1、通用工作寄存器； 2、RAM 数据寻址指针@HL（91H）。 3、配合指令 MOVC 对 ROM 数据进行查表。							
84H(R/W)	Y								

例：用 Y、Z 作为数据指针，访问bank0中 020H 处的内容。

B0MOV Y, #00H

B0MOV Z, #20H

B0MOV A, @YZ

例：对bank0中的数据进行清零处理。

CLR Y ; Y = 0, 指向bank 0。

B0MOV Z, #7FH ; Z = 7FH。

CLR_YZ_BUF:

CLR @YZ ; @YZ 清零。



```

DECMS Z                      ; Z - 1, 如果 Z = 0, 程序结束。
JMP CLR_YZ_BUF
CLR @YZ
END_CLR:
...
...

```

5.1.3 R 寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
82H(r/w)	R	1、作为工作寄存器使用； 2、存储执行查表指令后的高字节数据。（执行MOVC 指令，指定ROM 单元的高字节数据会被存入R 寄存器而低字节数据则存入ACC。） 3、存储乘法指令的高8位数据（执行MUL 指令，运算结果的高字节数据会被存入R 寄存器而低字节数据则存入ACC。）							

例：查找 ROM 地址为“TABLE1”的值。

```

B0MOV Y, #TABLE1$M          ; 设置 TABLE1 地址高字节。
B0MOV Z, #TABLE1$L          ; 设置 TABLE1 地址低字节。
MOVC                         ; 查表，R = 00H，ACC = 35H。
INCMS Z                      ; 查找下一地址。
JMP @F                       ; Z 没有溢出。
INCMS Y                      ; Z 溢出（FFH -> 00），-> Y=Y+1
NOP
@@: MOVC                     ; 查表，R = 51H，ACC = 05H。
... ;

```

TABLE1:

```

DW 0035H                    ; 定义数据表（16 位）数据。
DW 5105H
DW 2012H
...

```

5.1.4 X（X 寄存器）

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
85H(r/w)	X	通用工作寄存器使用							

5.1.5 W（W 寄存器）

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
88H(r/w)~ 8FH(r/w)	W0~ W7	通用工作寄存器使用							



5.1.6 PC（程序计数器）与堆栈

程序计数器PC是一个13位二进制程序地址寄存器，分高5位和低8位。专门用来存放下一条需要执行指令的内存地址。通常程序计数器会随程序中指令的执行自动增加。

若程序执行CALL和JMP 指令时，PC 指向特定的地址。下图描述了其结构。

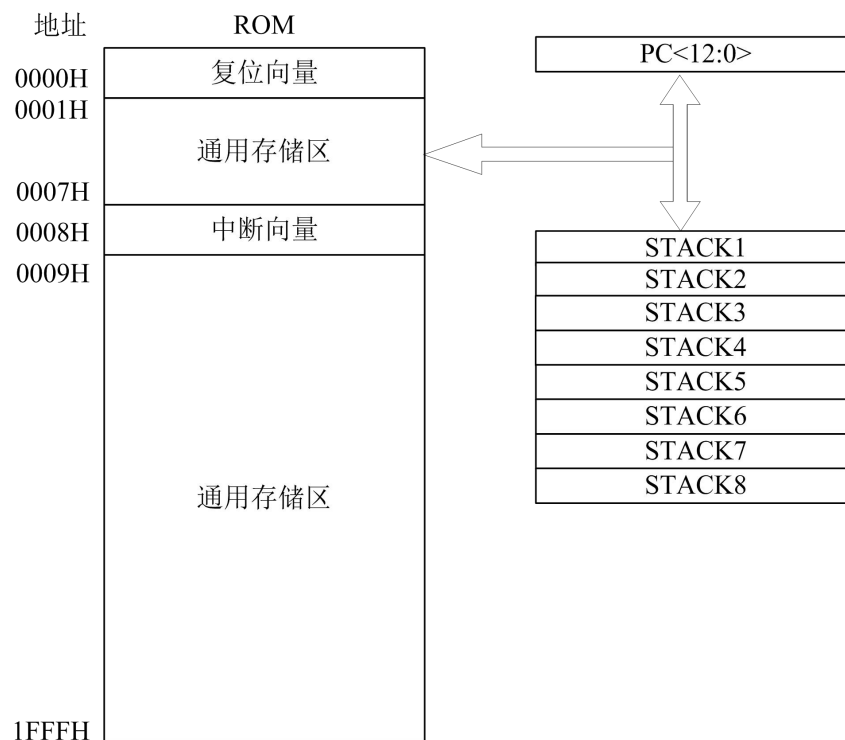


图 5-3 程序计数器结构

5.1.6.1 PCH 寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
94H(r/w)	PCH	-	-	-	PC12	PC11	PC10	PC9	PC8

5.1.6.2 PCL 寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
93H(r/w)	PCL	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0

产生 8K Word ROM 地址以获取对应程序指令代码。

复位后 PCL 的所有位均清零

JMP 指令直接装载 PC 的 13 位。因此，“JMP”指令跳转范围为 8K 程序区间

CALL 指令加载 PC 的 13 位，然后 PC+1 进入堆栈。

RET、RETI、RETLW 指令将栈顶数据装入 PC。

ADD PCL,A（ADC PCL,A）允许“A”的值加到当前 PC，PC 的高 5 位将自然进位。

MOV PCL,A 允许将寄存器“A”的值装入 PC 的低 8 位，同时 PC 的高 5 位保持不变。



SUB PCL,A (SBC PCL,A) 允许“A”的值被减到当前 PC，但 PC 的高 5 位保持不变。

改变 PCL 内容的指令需要 2 个指令周期，除此之外，所有的指令均只需 1 个指令周期。

5.1.7 PFLAG（状态寄存器）

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
86H(r/w)	PFLAG	/TO	/PD	-	-	-	C	DC	Z

寄存器包含 ALU 的算术状态，RESET 状态。

C：进位/借位（加减法指令）

=1，有进位/无借位

=0，无进位/有借位

DC：半进位/半借位（加减法指令）

=1，有低 4 位进位/无低 4 位借位

=0，无低 4 位进位/有低 4 位借位

Z：零标志位

=1，逻辑操作结果为 0

=0，逻辑操作结果不为 0

/PD：掉电标志位

=1，系统上电或执行 CLRWDWT 指令后

=0，执行 SLEEP 指令后

/TO：时间溢出标志位

=1，系统上电或执行 CLRWDWT 或执行 SLEEP 指令后

=0，看门狗溢出后

5.1.8 RBANK（RAM 存储区选择寄存器）

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
87H(r/w)	RBANK	SBANKS2	SBANKS1	-	-	-	-	-	RBANKS

RBANKS：通用寄存器 bank0，bank1 选择

=1，选择 bank1

=0，选择 bank0

SBANKS<2:1>：特殊寄存器 bank0、bank1、bank2 选择

=00，选择 bank0

=01，选择 bank1

=10，选择 bank2

=11，选择 bank1



5.1.9 WDTC (看门狗控制寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0CCH(r/w)	WDTC	WDT_EN	WDT_pre2	WDT_pre1	WDT_pre0	-	-	-	-

WDT_EN: 看门狗使能位

=0: 有效

=1: 无效

WDT_pre<2:0>: WDT 溢出时间选择位

WDT_pre2	WDT_pre1	WDT_pre0	WDT 的分频比
0	0	0	1:1
0	0	1	1:2
0	1	0	1:4
0	1	1	1:8
1	0	0	1:16
1	0	1	1:32
1	1	0	1:64
1	1	1	1:128

5.1.10 WDTR 寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
96H(w)	WDTR	看门狗清零寄存器							

例：如下是对看门狗定时器的操作，在主程序开头对看门狗清零。

```

MOV      A,#5AH
B0MOV    WDTR,A
. . .
CALL     SUB1
CALL     SUB2
. . .
JMP      MAIN

```

清门狗定时器应注意：

- 1、对看门狗定时器清零之前，检查 I/O 口的状态和 RAM 的内容可增强程序的可靠性；
- 2、不能在中断中对看门狗清零，否则无法检测到主程序跑飞的情况。
- 3、程序中应该只在主程序中有一次清看门狗的动作，这种架构能够最大限度地发挥看门狗的保护功能。
- 4、WDT 的复位时间是可配置的。



5.1.11 LVDC（控制寄存器）

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0CFH(r/w)	LVDC	LVDTE	-	-	-	-	-	-	-

LVDTE: LVD 使能位

=0: 标志无效

=1: 标志有效

5.1.12 OSCM 寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
95H(r/w)	OSCM	LRCS32K	LRCSXT	-	CPUM1	CPUM0	CLKMD	STPHX	14M32KSEL

LRCS32K:

=0: 选择内部低速 32K RC 作为低速系统时钟

=1: 选择内部低速 RC（7.8K~250K）作为低速系统时钟；

LRCSXT:

=1: RTC 模式使能

=0: RTC 模式禁止；

CPUM<1:0>: CPU 工作模式控制位

00: 普通模式 01: 睡眠模式 10: 绿色模式 11: 系统保留

CLKMD: 系统时钟模式控制位

=0: 普通（双时钟）模式，高速时钟作为系统时钟

=1: 低速模式，低速时钟作为系统时钟

STPHX: 高速振荡器控制位

=0: 运行

=1: 停止，内部低速 RC 振荡器仍然运行

例：停止高速振荡器

B0BSET FSTPHX ; 停止外部高速振荡器

注：STPHX 为内部高速 RC 振荡器和外部高速振荡器的控制位。当 STPHX=0 时内部高速 RC 振荡器和外部高速振荡器正常运行；当 STPHX=1 时内部高速 RC 振荡器和外部高速振荡器停止运行。不同的高速时钟选项决定不同的 STPHX 功能。

14M32KSEL:

=1: 选择内部低速 RC 作为时钟

=0: 选择高速 RC 作为时钟；

注：此位对 TC0、T0、UART、12bitPWM、16bitPWM 有效，即选择 Fosc 作为时钟时，此位可控制时钟为高低速。



5.1.13 REMC

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0AFH(r/w)	REMC	IE_sel	D_sel	ELE_en	Rem_sel <4>	Rem_sel <3>	Rem_sel <2>	Rem_sel <1>	Rem_sel <0>

IE_sel : 内置三极管选择

=0 无效

=1 有效

D_sel : 内置三极管时, 发码斜率选择

=0: 不变

=1: 加强

ELE_en : 内置三极管时, 电源是否加电解的控制位

=0: 省电解

=1: 必须加电解

REM_sel<4:0> : REM 端内置三极管的驱动能力选择。

=00000: 默认大小

=00011: 最小

=11100: 最大

5.1.14 LFSEL

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
2B2H(r/w)	LFSEL	-	-	-	-	-	LFSEL2	LFSEL1	LFSEL0

LFSEL<2: 0>: 低速模式时钟频率选择

=000: 250K

=001: 125K

=010: 62.5K

=011: 31.25K

=100: 15.6K

=101~111: 7.8K

5.1.15 SMTS

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0E5H(r/w)	SMTS	-	-	-	-	SMTS4	-	SMTS1	SMTS0

SMTS4: P4 口施密特使能位

=0: 使能

=1: 禁止

SMTS1: P1 口施密特使能位



=0: 使能

=1: 禁止

SMTS0: P0 口施密特使能位

=0: 使能

=1: 禁止

5.1.16 TIRCF

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
1CEH(r/w)	TIRCF	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0

TIRCF<7:0>: 系统频率调节位，对应位为 1 则降低系统频率。

共分 163 档，当系统频率选择 14MHZ 时，每档 35KHZ 左右，Bit7 占用 80 档，Bit6 占用 40 档，Bit5 占用 20 档，Bit4 占用 10 档，Bit3 占用 6 档，Bit2 占用 4 档，Bit1 占用 2 档，Bit0 占用 1 档。

例 1: 默认 TIRCF<7:0>=00h，系统频率为 14MHZ，若寄存器 TIRCF 写值为 01h，则系统频率变为 13.965MHZ。

例 2: 默认 TIRCF<7:0>=00h，系统频率为 14MHZ，若想系统频率调节至 10MHZ，则需加 $\frac{4\text{MHZ}}{35\text{KHZ}} = 114$ 档，即寄存器 TIRCF 写值为 B4h。

例 3: 默认 TIRCF<7:0>=00h，系统频率为 14MHZ，若 TIRCF 写值为 FFh，则系统频率调节至最小 8.295MHZ。

当系统频率选择 28MHZ 时，每档 50KHZ 左右，Bit7 占用 80 档，Bit6 占用 40 档，Bit5 占用 20 档，Bit4 占用 10 档，Bit3 占用 6 档，Bit2 占用 4 档，Bit1 占用 2 档，Bit0 占用 1 档。

例 4: 默认 TIRCF<7:0>=00h，系统频率为 28MHZ，若寄存器 TIRCF 写值为 01h，则系统频率变为 27.950MHZ。

例 5: 默认 TIRCF<7:0>=00h，系统频率为 28MHZ，若想系统频率调节至 24MHZ，则需加 $\frac{4\text{MHZ}}{50\text{KHZ}} = 80$ 档，即寄存器 TIRCF 写值为 80h 或者 7Ch。

例 6: 默认 TIRCF<7:0>=00h，系统频率为 28MHZ，若 TIRCF 写值为 FFh，则系统频率调节至最小 19.850MHZ。

5.2 TC0（16 定时/计数器寄存器）

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0ECH(w/r)	TC0CL	定时器 TC0 低 8 位							



0EAH(w/r)	TC0M	模式选择
0EBH(w/r)	TC0R	自动重装载寄存器
0EDH(w/r)	TC0CH	PWM0 高电平占空比选择/定时器高 8 位
0EEH(w/r)	TC0D1	PWM1 高电平占空比选择
0E6H(w/r)	TC0D2	PWM2 高电平占空比选择
0E7H(w/r)	TC0M1	PWM 控制

5.2.1 概述

定时/计数器 TC0 具有多时钟源，可根据实际需要选择 Fcpu 或者 Fosc，寄存器 TC0M 控制时钟源的选择。当 TC0C 从 FFFFH 溢出到 0000H 时，TC0C 在继续计数的同时产生一个溢出信号 TC0IF，TC0IE=1，触发 TC0 中断请求。

TC0 的主要作用如下：

- 16 位可编程定时器：根据选定的时钟频率在特定时间产生中断信号(非 PWM 模式)；
- PWM 输出（TC0 为 8 位）。

5.2.2 TC0CL

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0ECH(r/w)	TC0CL	TC0 的定时寄存器的低 8 位							

5.2.3 TC0M

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0EAH(r/w)	TC0M	TC0TR	TC0S2	TC0S1	TC0S0	TCCKS	INT0CK	-	PWM0

TC0TR: TC0 启动控制位。

=0 : 禁止 TC0 定时器；

=1 : 开启 TC0 定时器。

TC0S[2:0]: TC0 分频选择位。

TC0S[2:0]	TCCKS=0	TCCKS=1
000	Fcpu / 256	Fosc / 128
001	Fcpu / 128	Fosc / 64
010	Fcpu / 64	Fosc / 32
011	Fcpu / 32	Fosc / 16
100	Fcpu / 16	Fosc / 8
101	Fcpu / 8	Fosc / 4
110	Fcpu / 4	Fosc / 2
111	Fcpu / 2	Fosc / 1

TCCKS: TC0 时钟信号控制位。

=0 : Fcpu



=1 : Fosc

INT0CK: TC0 外部时钟信号控制位。

=0 : 选内部时钟

=1 : 选择 P0.0 作为时钟信号

PWM0:

=0: 禁止 PWM0 输出

=1: 使能 PWM0 输出

注：当禁止 PWMOUT 时，TC0 为 16 位定时/计数器。TC0CH 为高 8 位，TC0CL 为低 8 位。

5.2.4 PWM 高电平占空比寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0EDH(r/w)	TC0CH	PWM0 高电平占空比控制寄存器/TC0 定时寄存器的高 8 位							
0EEH(r/w)	TC0D1	PWM1 高电平占空比控制寄存器							
0E6H(r/w)	TC0D2	PWM2 高电平占空比控制寄存器							

TC0R 控制 PWM 的周期及分辨率，TC0H 控制 PWM 高电平占空比，当 TC0C=TC0H 时，PWM 翻转为低，完成高电平的占空比。

TC0H 初始值计算公式如下：

$$\text{TC0H 初始值} = \text{TC0R} + (\text{高电平周期} * \text{输入时钟})$$

5.2.5 TC0R

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0EBH(r/w)	TC0R	自动重装载寄存器							

当 TC0 溢出时，TC0R 的值自动装入 TC0 中。这样，用户在使用的过程中就不需要在中断中重新赋值。TC 为双重缓存器结构。若程序对 TC0R 进行了修改，那么修改后的 TC0R 值先被暂存在 TC0R 的第一个缓存器中，直到当前 TC0 溢出后，才被真正存入 TC0R 缓存器中，从而避免 TC 中断时间出错以及 PWM 和蜂鸣器误动作。

TC0R 初始值计算公式如下：

$$\text{TC0R 初始值} = N - (\text{TC 中断间隔时间} * \text{输入时钟})$$

上式中，N 为 TC 的最大计数范围 256。

例：TC 中断间隔时间设置为 10ms，时钟源选 Fcpu (TnCKS = 0)，使能 PWM 输出 (PWMEN = 1)，高速时钟为 4MHz，Fcpu = Fosc/4，TPS = 010 (Fcpu/64)。

$$\begin{aligned} \text{TC0R 初始值} &= N - (\text{TC 中断间隔时间} * \text{输入时钟源}) \\ &= 256 - (10\text{ms} * 4\text{MHz} / 4 / 64) \\ &= 256 - (10^{-2} * 4 * 10^6 / 4 / 64) = 100 = 64\text{H} \end{aligned}$$

注：在 PWM 模式下，系统自动开启自动装载功能。



5.2.6 TC0M1

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0E7H(r/w)	TC0M1	PWM2S1	PWM1S1	PWM0S1	-	-	-	PWM2EN	PWM1EN

PWM1EN: PWM1 使能位

=1 : 允许 PWM1 输出

=0 : 禁止 PWM1 输出, 相应端口变成 IO 口

PWM2EN: PWM2 使能位

=1 : 允许 PWM2 输出

=0 : 禁止 PWM2 输出, 相应端口变成 IO 口

PWM0S1: PWM0 输出端口选择

=0: 无输出

=1: P4.7 输出 PWM0

PWM1S1: PWM1 输出端口选择

=0: 无输出

=1: P4.6 输出 PWM1

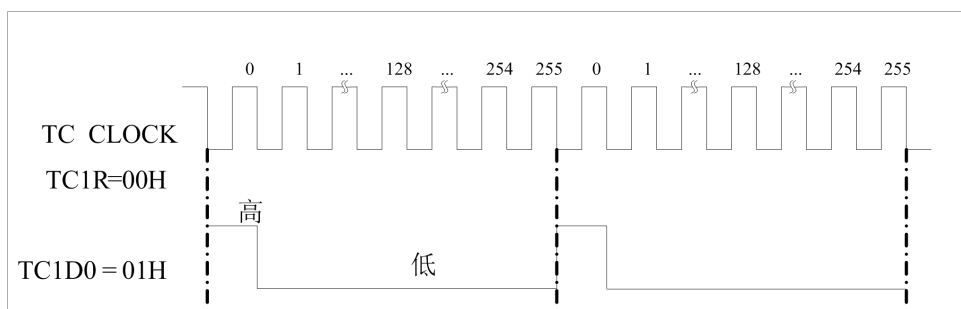
PWM2S1: PWM2 输出端口选择

=0: 无输出

=1: P4.5 输出 PWM2

5.2.7 PWM 波形

PWM信号靠TC0C, TC0R, TC0H (TC0D1、TC0D2) 的比较产生, 当TC0溢出时, PWM输出高电平, 为PWM的初始态, 此时TC0C重新载入TC0R寄存器的值, TC0R决定PWM的周期及分辨率。TC0开始计数后, 当TC0C等于TC0D时, PWM翻转为低电平, TC0继续计数, 当TC0溢出时, 一个PWM周期完成, PWM重新自动载入TC0R的值并且输出高电平, 开始下一个周期。TC0R决定了PWM的分辨率, 如果TC0R为0x00, 则分辨率为1/256, 如果TC0R为0x80, 则分辨率为1/128, 可参看下图示。



注: 如果 TC0D0 小于 TC0R, 则 PWM 一直输出高电平。

5.3 T0 (定时/计数器寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
1CAH(w/r)	T0L	定时器 T0 低 8 位							



1C9H(w/r)	T0M	T0 模式控制寄存器
1CBH(w/r)	T0H	定时器 T0 高 8 位
1CCH(w/r)	T0M1	T0 模式控制寄存器 1

5.3.1 概述

16位普通定时/计数器，内部时钟来自Fcpu或低速时钟，当T0从FFFFH 溢出到0000H时，T0在继续计数的同时产生一个溢出信号T0IF，中断使能T0IE有效时，触发T0中断请求。

5.3.2 T0M 寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
1C9H(r/w)	T0M	T0TR	TP0S2	TP0S1	TP0S0	TP0SEL	T0CK	-	TC0GN

T0TR: T0 使能控制。

=0 : 禁止;

=1 : 使能。

TP0S[2:0]: T0 分频选择位。

TP0S[2:0]	T0CK=0	T0CK=1
000	Fcpu / 256	Fosc / 128
001	Fcpu / 128	Fosc / 64
010	Fcpu / 64	Fosc / 32
011	Fcpu / 32	Fosc / 16
100	Fcpu / 16	Fosc / 8
101	Fcpu / 8	Fosc / 4
110	Fcpu / 4	Fosc / 2
111	Fcpu / 2	Fosc / 1

TP0SEL: T0 预分频是否使用;

=0 : 不使用。T1 的时钟源为 1 个 Fcpu，预分频位不再有效。

=1 : 使用

T0CK: T0 时钟源;

=0 : T0 的时钟源为 Fcpu。

=1 : T0 的时钟源为低速 RC 振荡器

TC0GN: TC0 绿色模式溢出唤醒使能控制。

=0 : 禁止;

=1 : 使能。

5.3.3 T0C 寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
----	----	----	----	----	----	----	----	----	----



1CAH(r/w)	T0L	T0 的定时寄存器的低 8 位
1CBH(r/w)	T0H	T0 的定时寄存器的高 8 位

5.3.4 T0M1 寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
1CCH(r/w)	T0M1	-	-	-	PWMW2	PWMW1	INT0W	T0OVS	T0GN

PWMW<2:1>PWM2 上升沿唤醒延时。

=00: 2us

=01: 4us

=10: 8us

=11: 12us

INT0W: INT 绿色模式唤醒选择控制位。

=0 : 禁止 PWM 上升沿在绿色模式唤醒

=1 : 允许 PWM 上升沿在绿色模式唤醒

T0OVS: T0 绿色模式溢出唤醒选择位。

=0 : 选择 T0 的 16 位为溢出位

=1 : 选择 T0 的低 8 位为溢出位

T0GN: T1 绿色模式溢出唤醒使能控制。

=0 : 禁止

=1 : 使能

5.4 T2（定时/计数器寄存器）

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
1C8H(w/r)	TL0C	定时器 T2 低 8 位							
1C7H(w/r)	TL0M	T2 模式控制寄存器							

5.4.1 概述

8位普通定时/计数器，内部时钟来自低速时钟，ILRC/2或者ILRC1/2，在RTC模式下，外部32.768K晶振作为其时钟，时钟源为XT_CLK/2。当T2从FFH 溢出到00H时，T2在继续计数的同时产生一个溢出信号T2IF，中断使能T2IE有效时，触发T2中断请求。

5.4.2 TL0C 寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
1C8H(r/w)	TL0C	T2 的定时寄存器的低 8 位							



5.4.3 TL0M

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
1C7H(r/w)	TL0M	T2TR	T2WC	TP2S2	TP2S1	TP2S0	-	-	-

TR2: T2 使能控制。

=0 : 禁止;

=1 : 使能。

T2WC: 低速唤醒睡眠选择位

=0: 低速唤醒睡眠进入普通模式

=1: 低速唤醒睡眠进入低速模式

TP2S[2:0]: T2 溢出频率选择。

TP2S[2:0]	分频
000	2
001	4
010	8
011	16
100	32
101	64
110	128
111	256

5.5 带死区互补的 12 位 PWM (TM1)

5.5.1 TM1 特性

- 1 组带死区互补 PWM 或 3 路独立 PWM 输出
- 提供每个 PWM 周期溢出中断，但中断共用同一个向量入口
- 输出极性可选择
- PWM 工作时钟源可设定时钟分频比
- TM1 可做定时器/计数器使用

5.5.2 TM1 输出模式

TM1 模块包含 1 个独立的波形发生模块，对应的 3 个 PWM 输出为 PWM0X、PWM01X、PWM1X，通过控制相关寄存器可对 PWM 输出配置成互补模式或独立输出模式。

5.5.2.1 TM1 互补输出模式

当 PWM0M 置 0: PWM 将工作在互补输出模式，互补模式时，通常使能 PWM0X&PWM01X 输出，此时可以控制对应的周期寄存器、占空比寄存器及死区时间寄存器，从而控制互补波形的输出。互补输出时可选择 PWM0 X&PWM01X 输出极性，方便用户各种电平驱动需求。



5.5.2.2 TM1 独立输出模式

当 PWM0M 置 1：PWM 将工作在独立输出模式，独立输出模式时，可以控制相关寄存器使能对应 PWM 端口单一输出或同时输出，同时让 PWM0X&PWM01X 输出时，其周期相同但占空比可单独设置。当为独立模式时占空比寄存器将控制 PWM0X 的占空比，死区时间控制寄存器将控制 PWM01X 的占空比，独立输出时也可控制 PWM0X&PWM01X 输出极性，方便用户各种电平驱动需求。

5.5.3 TM1 相关寄存器

5.5.3.1 TM1 使能寄存器 PWM0EN

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0B9H(r/w)	PWM0EN	TM1EN	-	TM1SEL 8	PWM1_O EN	PWM0M	PWM01 _OEN	PWM0_ OEN	PWM0_ EN

TM1EN：TM1 定时器总使能

=0：禁止

=1：使能

TM1EL8：8/12 位定时器选择位（定时器时 PWM0_EN 必须无效）

=0：8 位

=1：12 位

PWM1_OEN：PWM1X 输出控制位

=0：禁止 PWM1X 输出

=1：允许 PWM1X 输出

PWM0M：PWM0 工作模式选择位

=0：PWM0X&PWM01X 工作于互补输出模式

=1：PWM0X&PWM01X 工作于独立输出模式

注：修改 PWM0 工作模式时建议先关闭 PWM0 模块。

PWM01_OEN：PWM01X 输出控制位

=0：禁止 PWM01X 输出

=1：允许 PWM01X 输出

PWM0_OEN：PWM0X 输出控制位

=0：禁止 PWM0X 输出

=1：允许 PWM0X 输出

注：PWM0 允许输出，必须在 PWM0_EN 置 1 下才有效，否则为 PWM0 输出关闭状态（输出时对应端口必须设为输出模式）；即使都禁止输出，只有 PWM0_EN 位被使能，PWM0 都可以溢出中断，即此时 PWM0 可以作为定时器使用，此控制位修改立即生效。

PWM0_EN：PWM0 模块使能控制位

=0：关闭 PWM0 模块

=1：打开 PWM0 模块（重新计数）

注：关闭时，PWM0 计数停止，输出立即关闭。打开时，PWM0 计数器都重新从 1 开始计数，输出受 PWM0_OEN 和 PWM01_OEN 控制。修改 PWM0 工作模式时建议先关闭 PWM0 模块。



5.5.3.2 PWM0 控制寄存器 PWM0C

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0BAH(r/w)	PWM0C	CK0[3]	CK0[2]	-	-	PWM0SL[1:0]		CK0[1]	CK0[0]

PWM0SL[1:0]: PWM0X 和 PWM01X 输出模式选择位

=00: PWM0X 和 PWM01X 均为低有效

=01: PWM0X 为低有效, PWM01X 为高有效

=10: PWM0X 为高有效, PWM01X 为低有效

=11: PWM0X 和 PWM01X 均为高有效

注: 对于独立模式, 输出模式选择位同样有效, 但与互补模式不同的是: 有效期间为占空比期间; 而互补模式中对于 PWM0X 的有效期间为占空比期间, PWM01X 的有效期间为占空比的互补期间。

CK0[3:0]: PWM0 时钟源选择位

CK0[2:0]	CK0[3]=0	CK0[3]=1
111	Fcpu/2	Fosc/1
110	Fcpu/4	Fosc/2
101	Fcpu/8	Fosc/4
100	Fcpu/16	Fosc/8
011	Fcpu/32	Fosc/16
010	Fcpu/64	Fosc/32
001	Fcpu/128	Fosc/64
000	Fcpu/256	Fosc/128

5.5.3.3 PWM0 周期寄存器 PWM0PL、PWM0PH

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0B7H(r/w)	PWM0PL	PWM0PL[7:0]							
0B8H(r/w)	PWM0PH	-	-	-	-	PWM0PH[3:0]			

PWM0 周期=[0XFFF-PWM0PH:PWM0PL]*PWM0 工作时钟源周期。

注: 修改 PWM0 周期寄存器, 须先修改高位后修改低位, 且修改都在下一个周期才有效。

5.5.3.4 PWM0 占空比寄存器 PWM0DL、PWM0DH

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0B5H(r/w)	PWM0DL	PWM0DL[7:0]							
0B6H(r/w)	PWM0DH	-	-	-	-	PWM0DH[3:0]			

注: 修改 PWM0 占空比寄存器, 操作类似修改 PWM0 周期寄存器, 都是必须先修改高位后修改



低位，且修改都在下一个周期才有效。

$PWM0X \text{ 占空比} = [0xFFFF - PWM0DH : PWM0DL] * PWM0 \text{ 工作时钟周期}$

5.5.3.5 PWM0 死区时间寄存器 PWM0DTL、PWM0DTH

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0BDH(r/w)	PWM0DTL	PWM0DTL[7:0]							
0BEH(r/w)	PWM0DTH	-	-	-	-	PWM0DTH[3:0]			

当 PWM0M=1 时，PWM0 工作在 2 路独立模式，此时的死区时间寄存器被用来那工作 PWM01X 的占空比寄存器，即独立模式的 PWM0 可以产生 2 路周期相同，但占空比可以不同的 PWM 波形。

互补模式下：PWM0X 死区时间= $[PWM0DTH : PWM0DTL] * PWM0 \text{ 工作时钟周期}$ 。

互补模式下：死区时间必须小于占空比时间，死区时间与占空比时间的和必须小于 PWM0 周期。

独立模式下：PWM01X 占空比时间= $[0xFFFF - PWM0DTH : PWM0DTL] * PWM0 \text{ 工作时钟周期}$ 。

5.5.3.6 PWM1 占空比寄存器 PWM2DL、PWM2DH

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0E8H(r/w)	PWM2DL	PWM2DL[7:0]							
0E9H(r/w)	PWM2DH	-	-	-	-	PWM2DH[3:0]			

注：修改 PWM1 占空比寄存器，操作类似修改 PWM0 周期寄存器，都是必须先修改高位后修改低位，且修改都在下一个周期才有效。

$PWM1X \text{ 占空比} = [0xFFFF - PWM2DH : PWM2DL] * PWM0 \text{ 工作时钟周期}$

5.5.3.7 PWM0 计数器/定时器时间寄存器 PWM0TL、PWM0TH

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0BBH(r/w)	PWM0TL	PWM0TL[7:0]							
0BCH(r/w)	PWM0TH	-	-	-	-	PWM0TH[3:0]			

PWM0 模块也可以做定时器，无自动装载功能，配置 TM1SEL8 (PWM0EN.5) 位：清 0 可选择 8 位计数器模式；置 1 可选 12 位计数器模式。做计数器模式下，需要把 PWM0 模块使能控制位 PWM0_EN (PWM0EN.0) 清零。

5.5.3.8 PWM0 输出端口选择寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0BFH(r/w)	PWMPSC	-	-	-	-	-	PWM1S0	PWM01S0	PWM0S0

PWM0S0: PWM0X 输出端口选择

=0: 无输出

=1: P0.4 输出 PWM0X

PWM01S0: PWM01X 输出端口选择



=0: 无输出

=1: P0.5 输出 PWM01X

PWM1S0: PWM1X 输出端口选择

=0: 无输出

=1: P0.6 输出 PWM1X

5.6 带死区互补的 16 位 PWM (TM2)

5.6.1 TM2 特性

- 1 组带死区互补 PWM 或 3 路独立 PWM 输出
- 提供每个 PWM 周期溢出中断，但中断共用同一个向量入口
- 输出极性可选择
- PWM 工作时钟源可设定时钟分频比
- TM2 可做定时器/计数器使用

5.6.2 TM2 输出模式

TM2 模块包含 1 个独立的波形发生模块，对应的 3 个 PWM 输出为 PWM3X、PWM31X、PWM5X，通过控制相关寄存器可对 PWM 输出配置成互补模式或独立输出模式。

5.6.2.1 TM2 互补输出模式

当 PWM3M 置 0: PWM 将工作在互补输出模式，互补模式时，通常使能 PWM3X&PWM31X 输出，此时可以控制对应的周期寄存器、占空比寄存器及死区时间寄存器，从而控制互补波形的输出。互补输出时可选择 PWM3X & PWM31X 输出极性，方便用户各种电平驱动需求。

5.6.2.2 TM2 独立输出模式

当 PWM3M 置 1: PWM 将工作在独立输出模式，独立输出模式时，可以控制相关寄存器使能对应 PWM 端口单一输出或同时输出，同时让 PWM3X&PWM31X 输出时，其周期相同但占空比可单独设置。当为独立模式时占空比寄存器将控制 PWM3X 的占空比，死区时间控制寄存器将控制 PWM31X 的占空比，独立输出时也可控制 PWM&PWMx1 输出极性，方便用户各种电平驱动需求。

5.6.3 TM2 相关寄存器

5.6.3.1 TM2 使能寄存器 PWM3EN

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
1B9H(r/w)	PWM3EN	TM2EN	TM2SE L16	TM2SE L8	PWM5_O EN	PWM3M	PWM31 _OEN	PWM3_ OEN	PWM3_ EN

TM2EN: TM2 定时器总使能

=0: 禁止



=1: 使能

TM2SEL16: 12/ 16 位定时器选择位 (TM2SEL8=1)

=0: 12 位

=1: 16 位

TM2SEL8: 8/ 12 位定时器选择位 (定时器时 PWM3_EN 必须无效)

=0: 8 位

=1: 12 位

PWM5_OEN: PWM5X 输出控制位

=0: 禁止 PWM5X 输出

=1: 允许 PWM5X 输出

PWM3M: PWM3X 工作模式选择位

=0: PWM3X&PWM31X 工作于互补输出模式

=1: PWM3X&PWM31X 工作于独立输出模式

注: 修改 PWM3 工作模式时建议先关闭 PWM3 模块。

PWM31_OEN: PWM31X 输出控制位

=0: 禁止 PWM31X 输出

=1: 允许 PWM31X 输出

PWM3_OEN: PWM3X 输出控制位

=0: 禁止 PWM3X 输出

=1: 允许 PWM3X 输出

注: PWM3 允许输出, 必须在 PWM3_EN 置 1 下才有效, 否则为 PWM3 输出关闭状态 (输出时对应端口必须设为输出模式); 即使都禁止输出, 只有 PWM3_EN 位被使能, PWM3 都可以溢出中断, 即此时 PWM3 可以作为定时器使用, 此控制位修改立即生效。

PWM3_EN: PWM3 模块使能控制位

=0: 关闭 PWM3 模块

=1: 打开 PWM3 模块 (重新计数)

注: 关闭时, PWM3 计数停止, 输出立即关闭。打开时, PWM3 计数器都重新从 1 开始计数, 输出受 PWM3_OEN 和 PWM31_OEN 控制。修改 PWM3 工作模式时建议先关闭 PWM3 模块。

5.6.3.2 PWM3 控制寄存器 PWM3C

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
1BAH(r/w)	PWM3C	CK3[3]	CK3[2]	-	-	PWM3SL[1:0]		CK3[1]	CK3[0]

PWM3SL[1:0]: PWM3X 和 PWM31X 输出模式选择位

=00: PWM3X 和 PWM31X 均为低有效

=01: PWM3X 为低有效, PWM31X 为高有效

=10: PWM3X 为高有效, PWM31X 为低有效

=11: PWM3X 和 PWM31X 均为高有效

注: 对于独立模式, 输出模式选择位同样有效, 但与互补模式不同的是: 有效期间为占空比期间; 而互补模式中对于 PWM3X 的有效期间为占空比期间, PWM31X 的有效期间为占空比的互补期间。



CK3[3:0]: PWM3 时钟源选择位

CK3[2:0]	CK3[3]=0	CK3[3]=1
111	Fcpu/2	Fosc/1
110	Fcpu/4	Fosc/2
101	Fcpu/8	Fosc/4
100	Fcpu/16	Fosc/8
011	Fcpu/32	Fosc/16
010	Fcpu/64	Fosc/32
001	Fcpu/128	Fosc/64
000	Fcpu/256	Fosc/128

5.6.3.3 PWM3 周期寄存器 PWM3PL、PWM3PH

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
1B7H(r/w)	PWM3PL	PWM3PL[7:0]							
1B8H(r/w)	PWM3PH	PWM3PH[7:0]							

PWM3 周期=[FFFF-PWM0PH:PWM0PL]*PWM3 工作时钟源周期。

注：修改 PWM3 周期寄存器，须先修改高位后修改低位，且修改都在下一个周期才有效。

5.6.3.4 PWM3 占空比寄存器 PWM3DL、PWM3DH

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
1B5H(r/w)	PWM3DL	PWM3DL[7:0]							
1B6H(r/w)	PWM3DH	PWM3DH[7:0]							

注：修改 PWM3 占空比寄存器，操作类似修改 PWM3 周期寄存器，都是必须先修改高位后修改低位，且修改都在下一个周期才有效。

PWM3X 占空比=[FFFF-PWM3DH:PWM3DL]*PWM3 工作时钟周期

5.6.3.5 PWM3 死区时间寄存器 PWM3DTL、PWM3DTH

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
1BDH(r/w)	PWM3DTL	PWM3DTL[7:0]							
1BEH(r/w)	PWM3DTH	PWM3DTH[7:0]							

当 PWM3M=1 时，PWM3 工作在 2 路独立模式，此时的死区时间寄存器被用来那工作 PWM31 的占空比寄存器，即独立模式的 PWM3 可以产生 2 路周期相同，但占空比可以不同的 PWM 波形。

互补模式下：PWM3X 死区时间=[PWM3DTH:PWM3DTL]*PWM3 工作时钟周期。

互补模式下：死区时间必须小于占空比时间，死区时间与占空比时间的和必须小于 PWM3 周期。

独立模式下：PWM31X 占空比时间=[FFFF-PWM3DTH:PWM3DTL]*PWM3 工作时钟周期。



5.6.3.6 PWM5 占空比寄存器 PWM5DL、PWM5DH

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
1B3H(r/w)	PWM5DL	PWM5DL[7:0]							
1B4H(r/w)	PWM5DH	PWM5DH[7:0]							

注：修改 PWM5X 占空比寄存器，操作类似修改 PWM3 周期寄存器，都是必须先修改高位后修改低位，且修改都在下一个周期才有效。

PWM5X 占空比=[FFFF-PWM5DH:PWM5DL]*PWM3 工作时钟周期

5.6.3.7 PWM3 计数器/定时器时间寄存器 PWM3TL、PWM3TH

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
1BBH(r/w)	PWM3TL	PWM3TL[7:0]							
1BCH(r/w)	PWM3TH	PWM3TH[7:0]							

PWM3 模块也可以做定时器，无自动装载功能，配置 TM2SEL8（PWM3EN.5）位：清 0 可选择 8 位计数器模式；置 1 可选 12 位计数器模式。配置 TM2SEL16（PWM3EN.6）位：清 0 可选择 12 位计数器模式；置 1 可选 16 位计数器模式。计数器模式下，需要把 PWM3 模块使能控制位 PWM3_EN（PWM3EN.0）清零。

5.6.3.8 PWM3 输出端口选择寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
1BFH(r/w)	PWM3PSC	-	-	-	-	-	PWM5S	PWM31S	PWM3S

PWM3S: PWM3X 输出端口选择

=0: 无输出

=1: P4.4 输出 PWM3X

PWM31S: PWM31X 输出端口选择

=0: 无输出

=1: P4.3 输出 PWM31X

PWM5S: PWM5X 输出端口选择

=0: 无输出

=1: P4.2 输出 PWM5X



5.7 中断

5.7.1 概述

HS26F2728提供14个中断源：TC0、T0、T2、端口变化、ADC中断、UART收发、SPI、TM1、TM2、COMP和外部中断INT0、INT1、INT2。外部中断等可以将系统从睡眠模式中唤醒进入高速模式，在返回到高速模式前，中断请求被锁定。一旦程序进入中断，寄存器STKP的位GIE被硬件自动清零以避免响应其它中断。系统退出中断后，硬件自动将GIE置“1”，以响应下一个中断。中断请求存放在寄存器INTRQ中。其中在进入中断时，8X系列特殊寄存器被自动保存，在退出中断时(RETI),自动将保存的值写入寄存器。

注：程序响应中断时，必须开启全局中断控制位GIE。

5.7.2 中断控制寄存器 0

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
99H(r/w)	INTEN0	PBIE	T2IE	TC0IE	T0IE	ADCIE	INT2IE	INT1IE	INT0IE

PBIE：端口变化中断控制位

=0：无效

=1：有效

T2IE：T2 中断控制位

=0：无效

=1：有效

TC0IE：TC0 中断控制位

=0：无效

=1：有效

T0IE：T0 中断控制位

=0：无效

=1：有效

ADCIE：12 bit ADC 中断使能

=0：无效

=1：有效

INT2IE： INT2 外部中断（P1.2）控制位

=0：无效

=1：有效

INT1IE： INT1 外部中断（P0.1）控制位

=0：无效

=1：有效

INT0IE： INT0 外部中断（P0.0）控制位

=0：无效

=1：有效



中断请求寄存器 INTRQ 中存放各中断请求标志。一旦有中断请求发生，则 INTRQ 中对应位将被置“1”，该请求被响应后，程序应将该标志位清零。根据 INTRQ 的状态，程序判断是否有中断发生，并执行相应的中断服务。

5.7.3 中断控制寄存器 1

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
9AH(r/w)	INTEN1	SPIIE	-	-	CMPIE	-	-	TM2IE	TM1IE

SPIIE: SPI 中断控制位

=0: 无效

=1: 有效

CMPIE: 比较器中断控制位

=0: 无效

=1: 有效

TM2IE: 16bitPWM 定时器中断控制位

=0: 无效

=1: 有效

TM1IE: 12bitPWM 定时器中断控制位

=0: 无效

=1: 有效

5.7.3 INTRQ0

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
97H(r/w)	INTRQ0	PBIF	T2IF	TC0IF	T0IF	ADCIF	INT2IF	INT1IF	INT0IF

PBIF: 端口变化中断请求标志

=0: 无中断请求

=1: 有中断请求

T2IF: T2 中断请求标志

=0: T2 无中断请求

=1: T2 有中断请求

TC0IF: TC0 中断请求标志

=0: TC0 无中断请求

=1: TC0 有中断请求

T0IF: T0 中断请求标志

=0: T0 无中断请求

=1: T0 有中断请求

ADCIF: 12bitADC 数据传送完中断请求标志



=0: 无中断请求

=1: 有中断请求

INT2IF: INT2 外部中断请求标志

=0: INT2 无中断请求

=1: INT2 有中断请求

INT1IF: INT1 外部中断请求标志

=0: INT1 无中断请求

=1: INT1 有中断请求

INT0IF: INT0 外部中断请求标志

=0: INT0 无中断请求

=1: INT0 有中断请求

5.7.4 INTRQ1

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
98H(r/w)	INTRQ1	SPIIF	-	-	CMPIF	-	-	TM2IF	TM1IF

SPIIF: SPI 中断请求标志

=0: 无中断请求

=1: 有中断请求

CMPIF: 比较器中断请求标志

=0: 无中断请求

=1: 有中断请求

TM2IF: 16bitPWM 定时器中断请求标志

=0: 无中断请求

=1: 有中断请求

TM1IF: 12bitPWM 定时器中断请求标志

=0: 无中断请求

=1: 有中断请求

5.7.5 PEDGE

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
9FH(r/w)	PEDGE	-	-	-	-	-	-	PG1	PG0

PG<1: 0>: 外中断边沿控制位

=00: 保留

=01: 上升沿

=10: 下降沿

=11: 上升/下降沿



5.7.6 STKP

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0EFH(r/w)	STKP	GIE	-	-	-	-	STKPB2	STKPB1	STKPB0

只有当全局中断控制位 GIE 置 1 时，程序才能响应中断。一旦发生中断，PC 指向中断向量地址 008H，堆栈加 1。

GIE：全局中断控制位

=0：禁止全局中断

=1：允许全局中断

STKPB<2: 0>：堆栈指针

入栈时，将 PC 存入当前指针的堆栈寄存器，然后 STKP-1(初始为全 1)；出栈时 STKP+1，然后将堆栈寄存器的内容写入 PC。

执行程序调用指令 CALL 和响应中断服务时，堆栈指针 STKP 的值减 1，指针指向下一个堆栈缓存器。同时，对程序计数器 PC 的内容进行入堆栈保存。

堆栈层数	STKP 寄存器			堆栈缓存器		注释
	STKPB2	STKPB1	STKPB0	高字节	低字节	
0	1	1	1	Free	Free	——
1	1	1	0	STK0H	STK0L	——
2	1	0	1	STK1H	STK1L	——
3	1	0	0	STK2H	STK2L	——
4	0	1	1	STK3H	STK3L	——
5	0	1	0	STK4H	STK4L	——
6	0	0	1	STK5H	STK5L	——
7	0	0	0	STK6H	STK6L	——
8	1	1	1	STK7H	STK7L	——
>8	1	1	0	——	——	堆栈溢出，出错

对应每个入栈操作，都有一个出栈操作来恢复程序计数器 PC 的值。RETI 指令用于中断服务程序中，RET 用于子程序调用。出栈时，STKP 加 1 并指向下一个空闲堆栈缓冲器。堆栈恢复操作如下表所示：

堆栈层数	STKP 寄存器			堆栈缓存器		注释
	STKPB2	STKPB1	STKPB0	高字节	低字节	
8	1	1	1	STK7H	STK7L	——
7	0	0	0	STK6H	STK6L	——
6	0	0	1	STK5H	STK5L	——
5	0	1	0	STK4H	STK4L	——
4	0	1	1	STK3H	STK3L	——
3	1	0	0	STK2H	STK2L	——



2	1	0	1	STK1H	STK1L	——
1	1	1	0	STK0H	STK0L	——
0	1	1	1	Free	Free	——

5.8 EEPROM

5.8.1 EECON1 (EE 控制寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
2E1H(r/w)	EECON1	-	TIMEEE	-	-	ERSALLEE	ERASEEE	PROGEE	READEE

TIMEEE: EEPROM 操作时长选择位。

=0 : EEPROM 操作为默认时长

=1 : EEPROM 操作为默认时长 2 倍

ERSALLEE: ERSALL (全擦) 操作使能。

=0 : 禁止 ERSALL 操作

=1 : 使能 ERSALL 操作

ERASEEE: ERASE (位擦) 操作使能。

=0 : 禁止 ERASE 操作

=1 : 使能 ERASE 操作

PROGEE: PROG 操作使能。

=0 : 禁止 PROG 操作

=1 : 使能 PROG 操作

READEE: READ 操作使能。

=0 : 禁止 READ 操作

=1 : 使能 READ 操作

注：每次执行完读、写、擦操作后，使能位会自动清零，无需软件清零，若需继续执行读、写、擦操作需重新使能。执行全擦操作时，需先使能 ERSALLEE，再使能 ERASEEE。

5.8.2 EADRL (EE 地址低位)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
2E0H(r/w)	EADRL	EE 地址低位							

5.8.3 EDATL (EE 数据低位)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
2E2H(r/w)	EDATL	EE 数据低位							



5.8.4 EEPEOM 操作示例

5.8.4.1 两次写、两次读：

```

B0BSET    RBANK.7      ;BANK2 ENABLE
;=====test write EE
CLR        EECON1
BSET       EECON1.6     ;DOUBLE
MOV        A, #000H
MOV        EEADRL,A
MOV        A, #012H
MOV        EEDATL,A
BSET       EECON1.1     ;WRITE
LOOPEE2:
BTS0       EECON1.1
JMP        LOOPEE2
;=====write again
MOV        A, #001H
MOV        EEADRL,A
;MOV       A, #0F1H
MOV        A, #027H
MOV        EEDATL,A
BSET       EECON1.1     ;WRITE
LOOPEE2_1:
BTS0       EECON1.1
JMP        LOOPEE2_1
;=====READ
MOV        A, #000H
MOV        EEADRL,A
BSET       EECON1.0     ;READ
NOP
MOV        A,EEDATL
MOV        A, #001H
MOV        EEADRL,A
BSET       EECON1.0     ;READ
MOV        A,EEDATL

```

5.8.4.2 全擦：

```

;=====test eraseALL
CLR        EECON1
BSET       EECON1.6     ;DOUBLE
MOV        A, #000H
MOV        EEADRL,A
BSET       EECON1.3     ;ERASE
BSET       EECON1.2     ;ERASE
LOOPEE4:
BTS0       EECON1.3
JMP        LOOPEE4

```



=====read erase era

```
MOV    A, #000H
MOV    EEADRL, A
BSET   EECON1.0    ;READ
NOP
MOV    A, EEDATL
MOV    A, #001H
MOV    EEADRL, A
BSET   EECON1.0    ;READ
MOV    A, EEDATL
MOV    A, #03FH
MOV    EEADRL, A
BSET   EECON1.0    ;READ
MOV    A, EEDATL
```



6. I/O 口

6.1 端口模式控制寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0A0H(r/w)	P0M	-	P06M	P05M	P04M	P03M	P02M	P01M	P00M
0A1H(r/w)	P1M	-	-	-	-	-	P12M	P11M	P10M
0A3H(r/w)	P4M	P47M	P46M	P45M	P44M	P43M	P42M	P41M	P40M

PnM[7:0]: Pn 模式控制位 (n=0、1、4)

0 = 输入模式;

1 = 输出模式。

6.2 端口数据寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0A5H(r/w)	P0	-	P06	P05	P04	P03	P02	P01	P00
0A6H(r/w)	P1	-	-	-	-	-	P12	P11	P10
0A8H(r/w)	P4	P47	P46	P45	P44	P43	P42	P41	P40

注：上电默认是不定态，需赋初值。

6.3 端口上拉控制寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0AAH(r/w)	P0UR	-	P06UR	P05UR	P04UR	P03UR	P02UR	P01UR	P00UR
0ABH(r/w)	P1UR	-	-	-	-	-	P12UR	P11UR	P10UR
0ADH(r/w)	P4UR	P47UR	P46UR	P45UR	P44UR	P43UR	P42UR	P41UR	P40UR

PnUR[7:0]: Pn 上拉控制位 (n=0、1、4)

0 = 上拉无效;

1 = 上拉有效。

6.4 端口下拉控制寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0D0H(r/w)	P0PD	-	P06PD	P05PD	P04PD	P03PD	P02PD	P01PD	P00PD
0D1H(r/w)	P1PD	-	-	-	-	-	P12PD	P11PD	P10PD
0D3H(r/w)	P4PD	P47PD	P46PD	P45PD	P44PD	P43PD	P42PD	P41PD	P40PD

PnPD[7:0]: Pn 下拉控制位 (n=0、1、4)

0 = 下拉无效;

1 = 下拉有效。



6.5 端口唤醒控制寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
9BH(r/w)	P0W	-	P06W	P05W	P04W	P03W	P02W	P01W	P00W
9CH(r/w)	P1W	-	-	-	-	-	P12W	P11W	P10W
9EH(r/w)	P4W	P47W	P46W	P45W	P44W	P43W	P42W	P41W	P40W

PnW[7:0]: Pn 唤醒控制位 (n=0、1、4)

0 = 唤醒无效;

1 = 唤醒有效。

6.6 端口开漏输出控制寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0E0H(r/w)	P0OD	-	P06OD	P05OD	P04OD	P03OD	P02OD	P01OD	P00OD
0E1H(r/w)	P1OD	-	-	-	-	-	P12OD	P11OD	P10OD
0E3H(r/w)	P4OD	P47OD	P46OD	P45OD	P44OD	P43OD	P42OD	P41OD	P40OD

PnOD[7:0]: Pn 开漏输出控制位 (n=0、1、4)

0 = 开漏无效;

1 = 开漏有效。

6.7 端口扫描模式控制寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
1C0H(r/w)	P0SC	-	P06SC	P05SC	P04SC	P03SC	P02SC	P01SC	P00SC
1C1H(r/w)	P4SC	P47SC	P46SC	P45SC	P44SC	P43SC	P42SC	P41SC	P40SC

PnSC[7:0]: Pn 扫描模式控制位 (n=0、4)

0 = 扫描模式无效;

1 = 扫描模式有效。

6.8 端口驱动控制寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
1C2H(r/w)	P0IS	-	-	-	-	P03IS	P02IS	P01IS	P00IS
1C3H(r/w)	P1IS	-	-	-	-	P13IS	P12IS	P11IS	P10IS
1C5H(r/w)	P4IS	-	-	-	-	P43IS	P42IS	P41IS	P40IS

PnIS[1:0]: Pn 输出强 0 驱动控制位 (n=0、1、2、4、5)

00= 默认强度

01= 加强



10= 加强

11= 最强

PnIS[3:2]: Pn 输出强 1 驱动控制位 (n=0、1、2、4、5)

0 0= 默认强度

01= 加强

10= 加强

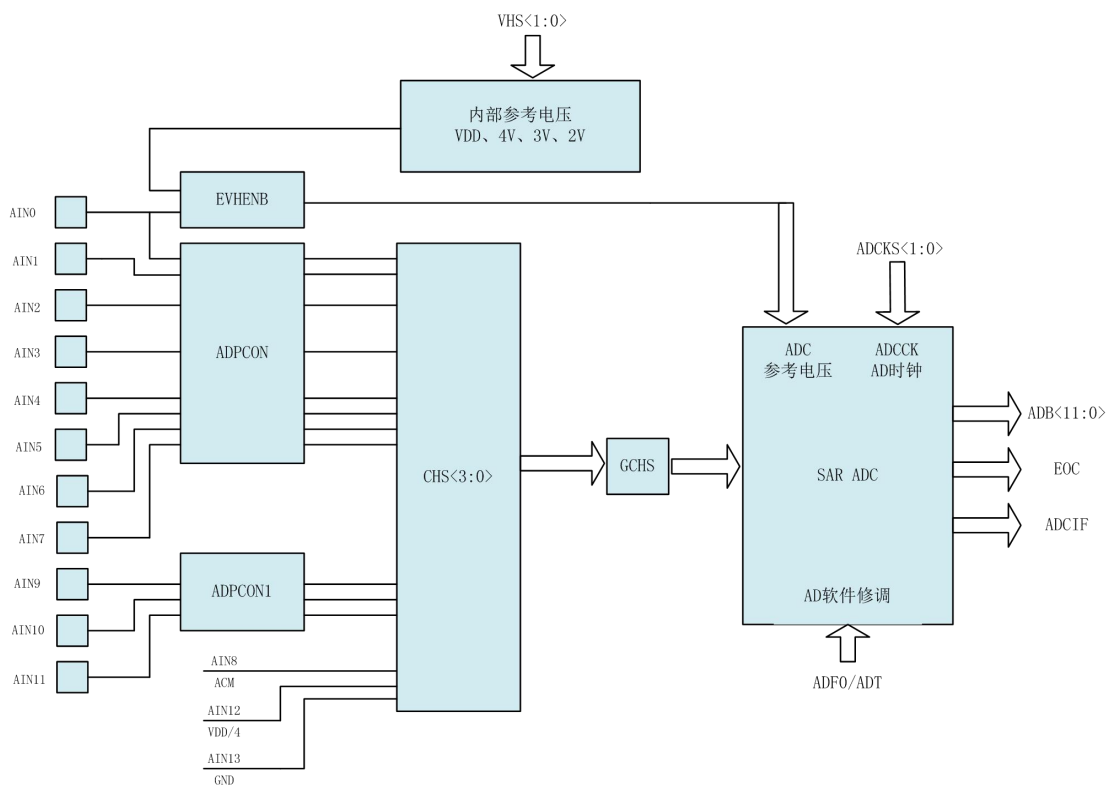
11= 最强



7. 14 通道 ADC

7.1 概述

模拟数字转换，内置 14 个模拟通道，高达 4096 阶的分辨率，能将一个模拟信号转换成相应的 12 位数字信号。通过 CHS[3:0]选择模拟信号输入引脚（AIN 引脚），内部 $1/4 \cdot V_{dd}$ 电压源，GCHS 位使能全部 ADC 通道，模拟信号输入至 SAR ADC。ADC 的分辨率为 12 位；可以通过 ADCKS[1:0]位选择 ADC 的转换速率以决定 ADC 的转换时间。ADC 参考电压的高电平包括 2 种，内部参考源，包括 VDD、4V、3V、2V（EVHENB=0），外部参考源，由 P4.0 提供（EVHENB=1）。ADC 内置 ADPCON、ADPCON1 寄存器来设置模拟输入引脚，必须由程序将 ADC 输入引脚设为不带上拉电阻的输入引脚。设置好 ADENB 和 ADS 位后，ADC 开始转换，转换结束时，ADC 电路将 EOC 和 ADCIF 置 1，并将转换结果存入 ADB 和 ADR 寄存器中。若 ADCIEN=1，ADC 请求中断，AD 转换完成后，ADCIF=1 时，程序计数器跳转中断向量地址（ORG 0008H）执行中断服务程序。



7.2 ADM（ADC 模式寄存器）

ADC模式寄存器ADM设置ADC的相关配置：包括ADC启动，ADC通道选择，ADC的参考源选择和ADC处理状态显示等。必须在AD开始转换前将这些配置设置完毕。

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0C8H(r/w)	ADM	ADENB	ADS	EOC	GCHS	CHS3	CHS2	CHS1	CHS0

ADENB: ADC控制位。睡眠模式下，禁止ADC以省电。

=0 : 禁止;



=1：使能。

ADS：ADC启动位。ADC处理完成后，ADS位自动清零。

=0：停止；

=1：开始。

EOC：ADC状态控制位。

=0：转换过程中；

=1：转换结束，ADS复位。

GCHS：通道选择位。

=0：禁止AIN通道；

=1：使能AIN通道。

CHS[3:0]：ADC输入通道选择位。

0000 = AIN0, 0001 = AIN1, 0010 = AIN2, 0011 = AIN3, 0100 = AIN4, 0101 = AIN5, 0110 = AIN6, 0111 = AIN7, 1000 = AIN8, 1001 = AIN9, 1010 = AIN10, 1011 = AIN11, 1100 = AIN12, 1101 = AIN13。

AIN8 通道没有连接外部管脚，连接的是 ACM（1.238V），AIN12 通道没有连接外部管脚，连接的是 $1/4 \times VDD$ 。这种处理方式可以使 ADC 测得 VDD 的电压。这对于 VDD 变化的，特别是手持设备非常有用。AIN13 通道没有连接外部管脚，连接的是 GND。

ADR 寄存器包括 ADC 模式控制和 ADC 低字节数据缓存器，ADC 配置包括 ADC 时钟速率和 ADC 分辨率。必须在启动 ADC 之前设置好这些配置。

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0CAH(r/w)	ADR	-	ADCKS1	-	ADCKS0	ADB3	ADB2	ADB1	ADB0

ADCKS1	ADCKS0	ADC 时钟源
0	0	Fcpu/16
0	1	Fcpu/8
1	0	Fcpu
1	1	Fcpu/2

ADB[3:0]：ADC12 位分辨率的低字节数据缓存器。

7.3 ADB（ADC 数据寄存器）

ADC 数据缓存器共 12 位，用来存储 AD 转换结果，8 位只读寄存器 ADB 存放结果的高字节（bit11~bit4），ADR（ADR[3:0]）存放低字节（bit3~bit0）。ADC 数据缓存器是只读寄存器，系统复位后处于未知状态。

ADB[11:4]：8 位 ADC 模式下，ADC 数据存放于 ADB 寄存器中。

ADB[3:0]：12 位 ADC 模式下，ADC 数据存放于 ADB 和 ADR 寄存器中。

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0C9H(r)	ADB	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4

ADB[11:4]：ADC12 位分辨率的高字节数据缓存器。



AIN _n	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4	ADB3	ADB2	ADB1	ADB0
0/4096*VREFH	0	0	0	0	0	0	0	0	0	0	0	0
1/4096*VREFH	0	0	0	0	0	0	0	0	0	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...
4094/4096*VREFH	1	1	1	1	1	1	1	1	1	1	1	0
4095/4096*VREFH	1	1	1	1	1	1	1	1	1	1	1	1

针对不同的应用，用户可能需要精度介于 8 位到 12 位之间的 AD 转换器。对于这种情况，可以通过对保存在 ADR 和 ADB 中的转换结果进行处理得到。首先，用户必须选择 12 位分辨率的模式，进行 AD 转换，然后在转换结果中去掉最低的几位得到需要的结果。如下表所示：

ADC 分辨率	ADB								ADR			
	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4	ADB3	ADB2	ADB1	ADB0
8-bit	0	0	0	0	0	0	0	0	X	X	X	X
9-bit	0	0	0	0	0	0	0	0	0	X	X	X
10-bit	0	0	0	0	0	0	0	0	0	0	X	X
11-bit	0	0	0	0	0	0	0	0	0	0	0	X
12-bit	0	0	0	0	0	0	0	0	0	0	0	0
0=可选位，X=未使用的位												

7.4 VREFH（ADC 参考电压寄存器）

ADC 内置 5 种参考电压，由 VREFH 寄存器控制：包括一个外部参考电压和 4 个内部参考源（VDD、4V、3V、2V）。EVHENB=1 时，ADC 参考电压由外部参考源提供（P4.0），必须输入一个电压作为 ADC 参考电压的高电平，且不能低于 2V。EVHENB=0 时，ADC 参考电压由内部参考源提供，并由 VHS[1:0]选择控制。VHS[1:0]=11 时，ADC 参考源选择 VDD；VHS[1:0]=10 时，ADC 参考源选择 4V；VHS[1:0]=01 时，ADC 参考源选择 3V；VHS[1:0]=00 时，ADC 参考源选择 2V。外部参考源的限制条件为，最高为 VDD，最低为内部最低电平，否则默认为 VDD。

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0CEH(r)	VREFH	EVHENB	-	-	-	-	-	VHS1	VHS0



VHS[1:0]: ADC 内部参考电压选择位。

VHS1	VHS0	内部 VREFH 电压
1	1	VDD
1	0	4.0V
0	1	3.0V
0	0	2.0V

EVHENB: ADC 内部参考电压控制位。

= 0: 允许 ADC 内部 VREFH 功能, VREFH 引脚是 P4.0/AIN0 引脚;

= 1: 禁止 ADC 内部 VREFH 功能, P4.0/AIN0/VREFH 引脚来自外部 VREFH 输入引脚。

7.5 ADC 操作说明和注意事项

7.5.1 ADC 信号格式

ADC 采样电压范围为参考电压高/低电平之间, ADC 参考低电压为 VSS, 高电压包括 VDD/4V/3V/2V, 外部参考电压由 P4.0/AVREFH 引脚提供 (由 EVHENB 控制)。EVHENB = 0 时, ADC 参考电压选择内部参考源; EVHENB = 1 时, ADC 参考电压选择外部参考源 (P4.0/AVREFH)。ADC 参考电压的范围为: (ADC 参考高电压-ADC 参考低电压) $\geq$ 2V, ADC 参考低电压为 VSS=0V, 故 ADC 参考高电压范围为 2V~VDD, 外部参考电压需在此范围之内。

ADC 内部参考低电压=0V。

ADC 内部端口低电压=VDD/4V/3V/2V。(EVHENB = 0)

ADC 外部参考电压=2V~VDD。(EVHENB = 1)

ADC 采样输入信号电压必须在 ADC 参考低电压和 ADC 参考高电压之间, 若 ADC 输入信号的电压不在此范围内, 则 ADC 的转换结果会出错 (满量程或者为 0)。

ADC 参考低电压 $\leq$ ADC 采用输入信号电压 $\leq$ ADC 参考高电压

7.5.2 AD 转换时间

ADC 转换时间是指从 ADS=1 (开始 ADC) 到 EOC=1 (ADC 结束) 所用的时间, 由 ADC 分辨率和 ADC 时钟 Rate 控制, 12 位 ADC 的转换时间为 $1/(ADC \text{ 时钟}/4) * 16 \text{ S}$; 8 位 ADC 的转换时间为 $1/(ADC \text{ 时钟}/4) * 12 \text{ S}$ 。ADC 的时钟源为 Fcpu, 包括 Fcpu/1, Fcpu/2, Fcpu/8, Fcpu/16, 由 ADCKS[1:0] 位控制。

ADC 的转换时间会影响 ADC 的性能, 如果输入高 Rate 的模拟信号, 必须要选择一个高 Rate 的 ADC 转换 Rate。如果 ADC 的转换时间比模拟信号的转换 Rate 慢, 则 ADC 的结果出错。故选择合适的 ADC 时钟 Rate 和 ADC 分辨率才能得到合适的 ADC 转换 Rate。

12 位 ADC 转换时间 = $1/(ADC \text{ 时钟 Rate}/4) * 16 \text{ sec}$

示例如下:

ADL EN	ADCKS1 ADCKS0	ADC 时 钟 rate	Fcpu=4MHz		Fcpu=16MHz	
			AD 转换时间	AD 转换 Rate	AD 转换时间	AD 转换 Rate



1(12-bit)	00	Fcpu/16	$1/(4\text{MHz}/16/4)*16=256\mu\text{s}$	3.906KHz	$1/(16\text{MHz}/16/4)*16=64\mu\text{s}$	15.625KHz
	01	Fcpu/8	$1/(4\text{MHz}/8/4)*16=128\mu\text{s}$	7.813KHz	$1/(16\text{MHz}/8/4)*16=32\mu\text{s}$	31.25KHz
	10	Fcpu	$1/(4\text{MHz}/4)*16=64\mu\text{s}$	62.5KHz	$1/(16\text{MHz}/4)*16=4\mu\text{s}$	250KHz
	11	Fcpu/2	$1/(4\text{MHz}/2/4)*16=32\mu\text{s}$	31.25KHz	$1/(16\text{MHz}/2/4)*16=8\mu\text{s}$	125KHz

8 位 ADC 转换时间 = $1/(\text{ADC 时钟 Rate}/4)*12\text{sec}$

ADLEN	ADCKS1 ADCKS0	ADC 时钟 rate	Fcpu=4MHz		Fcpu=16MHz	
			AD 转换时间	AD 转换 Rate	AD 转换时间	AD 转换 Rate
0(8-bit)	00	Fcpu/16	$1/(4\text{MHz}/16/4)*16=192\mu\text{s}$	5.208KHz	$1/(16\text{MHz}/16/4)*16=48\mu\text{s}$	20.833KHz
	01	Fcpu/8	$1/(4\text{MHz}/8/4)*16=96\mu\text{s}$	10.416KHz	$1/(16\text{MHz}/8/4)*16=24\mu\text{s}$	41.667KHz
	10	Fcpu	$1/(4\text{MHz}/4)*16=48\mu\text{s}$	83.333KHz	$1/(16\text{MHz}/4)*16=12\mu\text{s}$	333.333KHz
	11	Fcpu/2	$1/(4\text{MHz}/2/4)*16=24\mu\text{s}$	41.667KHz	$1/(16\text{MHz}/2/4)*16=6\mu\text{s}$	166.667KHz

7.5.3 ADC 引脚配置

ADC输入引脚与P4口共用，ADC输入通道的选择由ADCHS[2:0]控制，ADCCHS[2:0]=000时选择AIN0，ADCCHS[2:0]=001时选择AIN1.....同一时间设置P4口的一个引脚作为ADC的输入引脚，该引脚必须设置为输入引脚，禁止内部上拉，并首先由程序使能P4CON寄存器。通过ADCHS[2:0]选择好ADC输入通道后，GCHS置1以使用ADC功能。

ADC输入引脚为GPIO引脚时必须设为输入模式。

必须禁止ADC输入引脚的内部上拉电阻。

ADC输入通道的P4CON位必须置1。

EVHENB = 1时，P4.0/AIN0为ADC外部参考源的输入引脚，此时，P4.0必须设为输入模式，并禁止其上拉电阻。

ADC外部参考源输入引脚为GPIO引脚时必须设为输入模式。

必须禁止ADC外部参考源输入引脚的内部上拉电阻。

ADC 输入引脚与普通 I/O 引脚共用。当输入一个模拟信号到 CMOS 结构端口时，尤其当模拟信号为 $1/2 VDD$ 时，可能产生额外的漏电流。当 P4 输入多个模拟信号时，也会产生额外的漏电流。睡眠模式下，上述漏电流会严重影响到系统的整体功耗。P4CON 为 P4 口的配置寄存器，将 P4CON[4:0]置 1，其对应的 P4 引脚将被设为纯模拟信号输入引脚，从而避免上述漏电流的产生。



地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0C7H(r)	ADPCON	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0

ADPCON[7: 0]: P4.n 配置控制位。

=0 : P4.n 可以作为模拟输入 (ADC 输入) 引脚或者 GPIO 引脚;

=1 : P4.n 只能作为模拟输入引脚, 不能作为 GPIO 引脚。

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0C5H(r)	ADPCON1	-	-	-	-	ADPC13	ADPC12	ADPC11	-

ADPCON1[3: 1]: P1.0~P1.2 配置控制位。

=0 : P1.0~P1.2 可以作为模拟输入 (ADC 输入) 引脚或者 GPIO 引脚;

=1 : P1.0~P1.2 只能作为模拟输入引脚, 不能作为 GPIO 引脚。

注: 当 P4.n/P1.0~P1.2 作为普通 I/O 口而不是 ADC 输入引脚时, ADPCON.n 必须置为 0, 否则 P4.n/P1.0~P1.2 的普通 I/O 信号会被隔离开来。

7.6 P4 口/P1 口 ADC 共用引脚

P4 口、P1 口和 ADC 的输入口共用, 施密特触发。同一时间只能设置 P4 口、P1 口的一个引脚作为 ADC 的测量信号输入口 (通过 ADM 寄存器来设置), 其它引脚则作为普通 I/O 使用。具体应用中, 当输入一个模拟信号到 CMOS 结构端口, 尤其当模拟信号为 $1/2 V_{DD}$ 时, 将可能产生额外的漏电流。同样, 当 P4 口、P1 口外接多个模拟信号时, 也会产生额外的漏电流。在睡眠模式下, 上述漏电流会严重影响到系统的整体功耗。ADPCON 为 P4 口、P1 口的配置寄存器。将 ADPCON 置“1”, 其对应的 P4 口、P1 口将被设置为纯模拟信号输入口, 从而避免上述漏电流的情况。

7.7 ADT (ADC 结果微调寄存器)

ADC 模式寄存器 ADM 设置 ADC 的相关配置: 包括 ADC 启动, ADC 通道选择, ADC 的参考源选择和 ADC 处理状态显示等。必须在 AD 开始转换前将这些配置设置完毕。

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0CBH(r/w)	ADT	-	-	-	ADTS<0 >	ADT<3 >	ADT<2 >	ADT<1 >	ADT<0 >

ADTS<0>: 修调方向控制。

=0 : 负方向(ADB 转换制变小);

=1 : 正方向(ADB 转换制变大)。

ADT<3:0>: ADT 修调位

=0000 : 修调 0mv

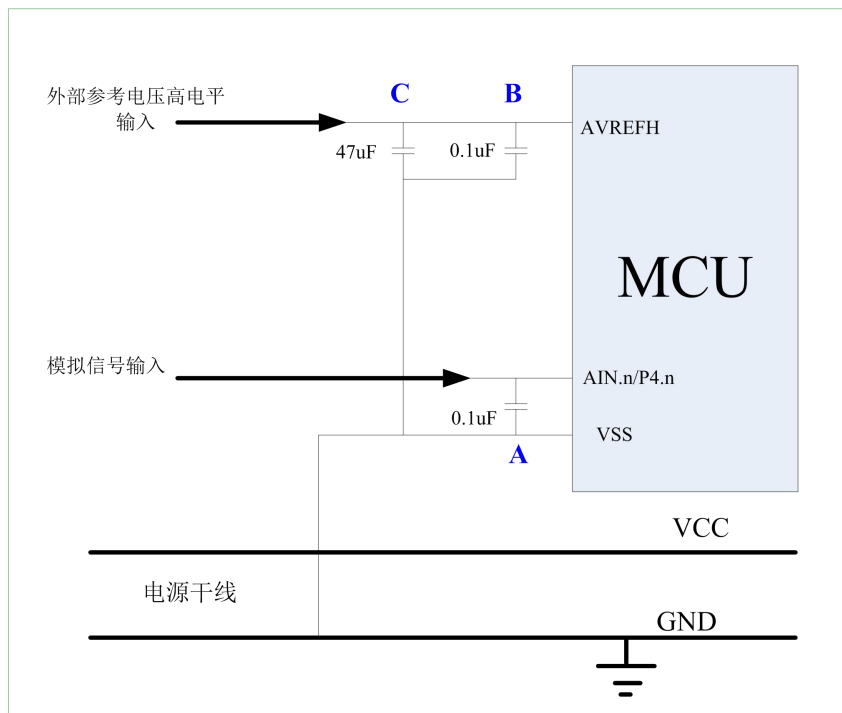
=0001 : 修调 1mv



=0010 : 修调2mv
=0011 : 修调3mv
=0100 : 修调4mv
=0101 : 修调5mv
=0110 : 修调6mv
=0111 : 修调7mv
=1000 : 修调8mv
=1001 : 修调9mv
=1010 : 修调10mv
=1011 : 修调11mv
=1100 : 修调12mv
=1101 : 修调13mv
=1110 : 修调14mv
=1111 : 修调15mv

注：如果得到的 AD 转换值偏小，可以使 ADTS<0>=1,并适当增加 ADT<3:0>的值做调整，反之亦然。

7.8 ADC 应用电路



模拟信号从 ADC 输入引脚 AINn/P4.n 输入。在 ADC 输入引脚和 VSS 之间(A)必须连接一个 0.1uF 的电容，且要尽可能的 ADC 输入引脚。不能将电容的 GND 直接连接到电源干线的 GND，必须通过 VSS 引脚。该电容可以减小电源干扰对模拟信号的影响。

ADC 参考电压高电平由外部参考源提供，外部参考源连接到 AVREFH 引脚 (P4.0)。在 AVREFH 引脚和 VSS 之间连接电容，首先在图中 C 处连接一个 47uF 的电解电容，再在 B 处连接一个 0.1uF 的



电容，且要尽可能的靠近 AVREFH 引脚。不能将电容的 GND 直接连接到电源干线上的 GND，必须通过 VSS 引脚。



8. 端口比较器、运算放大器

8.1 端口比较器相关寄存器

8.1.1 CMPCON(比较器控制寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
2E3H(r/w)	CMPCON	CMPEN	COMIF	FTCKS	CODEB	CPEN+	CVEN+	COPOL	-

CMPEN: CMP 使能位

=1 : 使能

=0 : 禁止

COMIF: 比较器转换标志

=1 : 比较输出 1

=0 : 比较输出 0

FTCKS: 输出使能滤波时钟选择

=1 : 低速时钟

=0 : 系统时钟

CODEB: 输出滤波

=1 : 使能, 3 个时钟延迟

=0 : 禁止

CPEN+: 比较器+端

=1 : 使能, 接 P1.0 端口

=0 : 禁止

CVEN+: 比较器+端

=1 : 使能, 接内部基准 1.25v (已校准)

=0 : 禁止

COPOL: 比较器输出极性选择

=1 : 取反

=0 : 不取反

8.1.2 CMPM(比较器负端输入选择寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
2E4H(r/w)	CMPM	COMSEL	CPINS1	CPINS0	XRVSEL	VRS3	VRS2	VRS1	VRS0

COMSEL: 运算放大器 A3 输出端 P4.5 作为负端比较电压使能位

=1: 使能

=0: 禁止

CPIINS[1:0]: 比较器负端输入选择



00	VDD
01	P1.1
10	VDD
11	VDD

输入电压的选择

电压选择	比较的电压点	
VRS<3:0>	XRVSEL=0	XRVSEL=1
0	5.63	4.69
1	5.29	4.41
2	5.00	4.17
3	4.74	3.95
4	4.50	3.75
5	4.29	3.57
6	4.09	3.41
7	3.91	3.26
8	3.75	3.13
9	3.46	2.88
10	3.00	2.50
11	2.65	2.21
12	2.37	1.97
13	1.96	1.63
14	1.73	1.44
15	1.50	CPIINS[1:0]选择的管脚

8.2 运算放大器相关寄存器

8.2.1 OPAM

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
2E5H(r/w)	OPAM	OPAM7	OPAM6	OPAM5	OPAM4	OPAM3	OPAM2	OPAM1	OPAM0

OPAM7: 运放 A1 使能控制

=0: A1 使能关闭

=1: A1 使能开启

OPAM6: RG 电阻连接开关

=0: 无连接

=1: 连接

OPAM<5:4>: RG 电阻阻值控制

=00: RG=R

=01: RG=1/3R

=10: RG=1/3R

=11: RG=1/5R



OPAM3: 仪表放大器后级连接开关与 A3 使能开关

=0: 使能关闭

=1: 使能开启

OPAM2: 仪表放大器运放 A3 输出控制开关, 输出到端口 P4.5

=0: 无输出

=1: 放大器 A3 输出到端口 P4.5

OPAM<1: 0>: 电阻 RSA1 阻值选择

=00: 4R

=01: 2R

=10: 1R

=11: 6R

8.2.2 OPAM1

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
2E6H(r/w)	OPAM1	OPAM17	-	-	-	OPAM13	OPAM12	OPAM11	OPAM10

OPAM1<7>: 运放 A2 使能控制

=0: A2 使能关闭

=1: A2 使能开启

OPAM1<3>: 输出 VAD2 控制, 运放 A2 输出到 ADC 控制

=0: 输出运放 A2 正端输入信号

=1: 输出运放 A2 的输出信号

结果输出到 12 位 ADC 的 Ain1 口。

OPAM1<2>: 输出 VAD1 控制, 运放 A1 或 A3 输出到 ADC 控制

OPAM<7>=0, OPAM1<2>=0, 输出运放 A1 的正端输入信号;

OPAM<7>=1, OPAM1<2>=0, 输出运放 A1 的输出信号;

OPAM<7>=1, OPAM1<2>=1, 输出仪表放大器的后级输出信号;

结果输出到 12 位 ADC 的 Ain0 口。

OPAM1<1: 0>: 电阻 RSA2 阻值选择

=00: 4R

=01: 2R

=10: 1R

=11: 6R

8.2.3 OPAMC

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
2E7H(r/w)	OPAMC	OPAMC7	LVHSEN	AP1PS	AP1PS0	AP1OSP	AP1OS	AP1NS0	AP1NS

OPAMC<7>: 运放 A1 正端信号输入选择;

=0, 直接输入端口 P4.0 口信号;



=1, 端口 P4.0 小信号抬高 Vth 输入到运放, 或 VSS 抬高 Vth 输入到运放 A1 正端

LVHSEN: 小信号电压平移转换选择

=0: 转换 P4.0 口值=VSS 时平移的结果;

=1: 转换 P4.0 口当前值平移时的结果;

注: 如果要得到 P4.0 口小信号输入 (500mv 以下) 比较精确的结果, 可以 $P4.0LVHSEN=1-P4.0LVHSEN=0$, 差值就是端口 P4.0 输入信号的实际值, 这样就能得到 P4.0 口的小信号比较准确的结果。

AP1PS: 运放 A1 正端信号选择, 端口 P4.0 处理后的信号

=0: 无信号

=1: 运放 A1 正端信号选择端口 P4.0 处理后的信号

AP1PS0: 运放 A1 正端信号选择, VSS

=0: 无信号

=1: 放 A1 正端信号选择 VSS

AP1OSP: 运放 A1 输出信号选择, 运放 A1 结果输出到 P4.3 口

=0: 无输出

=1: 运放 A1 结果输出到 P4.3 口

AP1OS: 运放 A1 输出信号选择, 运放 A1 结果反馈到 A1 运放的负端输入开关

=0: 无输出

=1: 运放 A1 结果反馈到 A1 运放的负端输入开关

AP1NS0: 运放 A1 负端信号选择, VSS

=0: 无信号

=1: 运放 A1 负端信号选择 VSS

AP1NS: 运放 A1 负端信号选择, P4.2 口

=0: 无信号

=1: 运放 A1 负端信号选择 P4.2 口

8.2.4 OPAMC1

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
2E8H(r/w)	OPAMC1	OPAMC17	LVHSEN1	AP2PS	AP2PS0	AP20SP	AP20S	AP2NS0	AP2NS

OPAMC1<7>: 运放 A2 正端信号输入选择;

=0, 直接输入端口 P4.1 口信号;

=1, 端口 P4.1 小信号抬高 Vth 输入到运放, 或 VSS 抬高 Vth 输入到运放 A1 正端

LVHSEN1: 小信号电压平移转换选择

=0: 转换 P4.1 口值=VSS 时平移的结果;

=1: 转换 P4.1 口当前值平移时的结果;

注: 如果要得到 P4.1 口小信号输入 (500mv 以下) 比较精确的结果, 可以 $P4.1LVHSEN=1-P4.1LVHSEN=0$, 差值就是端口 P4.1 输入信号的实际值, 这样就能得到 P4.1 口的小信号比较准确的结果。

AP2PS: 运放 A2 正端信号选择, 端口 P4.1 处理后的信号

=0: 无信号



=1: 运放 A2 正端信号选择端口 P4.1 处理后的信号

AP2PS0: 运放 A2 正端信号选择, VREF

=0: 无信号

=1: 放 A1 正端信号选择 VREF

AP2OPS: 运放 A2 输出信号选择, 运放 A2 结果输出到 P4.4 口

=0: 无输出

=1: 运放 A2 结果输出到 P4.4 口

AP2OS: 运放 A2 输出信号选择, 运放 A2 结果反馈到 A2 运放的负端输入开关

=0: 无输出

=1: 运放 A2 结果反馈到 A2 运放的负端输入开关

AP2NS0: 运放 A2 负端信号选择, VSS

=0: 无信号

=1: 运放 A2 负端信号选择 VSS

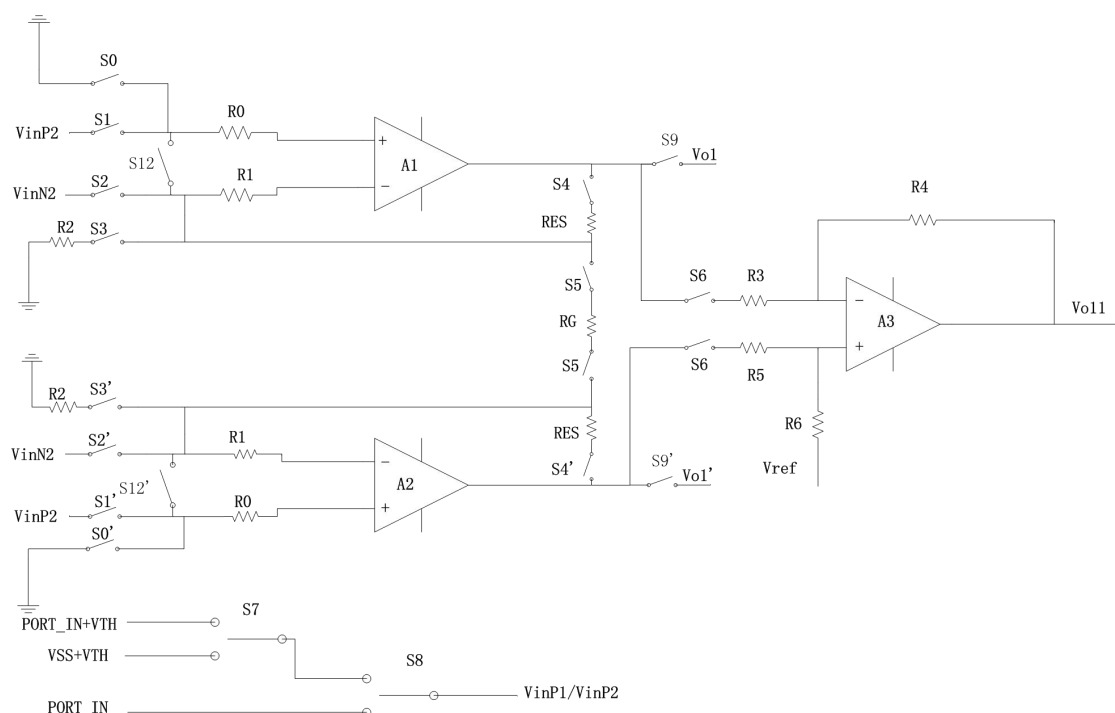
AP2NS: 运放 A2 负端信号选择, P1.2

=0: 无信号

=1: 运放 A2 负端信号选择 P1.2 口

8.3 应用原理与应用举例

8.3.1 应用原理



图中输入、输出、开关对应寄存器:



OPAMC<0>:S2

OPAMC<1>:S3

OPAMC<2>:S4

OPAMC<3>:S9, Vo1输出

OPAMC<4>:S0

OPAMC<5>:S1

OPAMC<6>:1>S7接PORT, 0>接VSS

OPAMC<7>:1>S8接VTH, 0>接PORT

S12:S1与S2同时断开时开启

OPAM<1:0>: RES电阻大小选择

OPAM<2>: Vo11输出

OPAM<3>: S6/A3使能

OPAM<5:4>: RG电阻大小选择

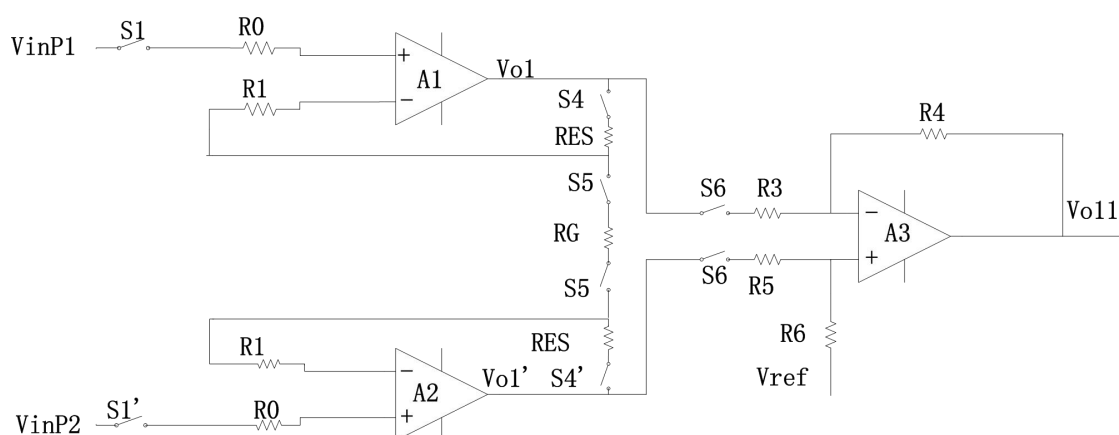
OPAM<6>: S5

OPAM<7>: A1使能

OPAM1<1:0>:VAD1与VAD2输出

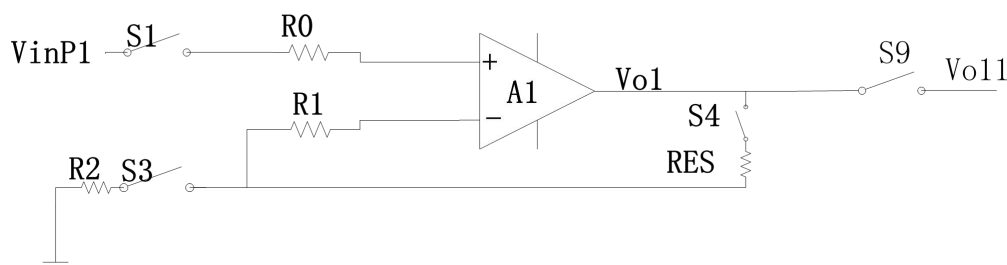
8.3.2 应用举例

8.3.2.1 仪表放大器



当电路作为仪表放大器使用时，需将开关 S1，S4，S5，S6 闭合，其他开关打开，通过调节 RES 与 RG 电阻的大小，来调节输出电压增益，理论最大差模输入增益可达到 59 倍，最小差模输入增益为 3.67 倍，可把 Vo11 输出到端口 P4.5 或者 12 位 ADC 的 Ain0 口。

8.3.2.2 小信号放大器



运放 A1 作为小信号放大器使用时，需将开关 S1，S3，S4，S9 闭合，其他开关断开，并将后级放大器断开，让其不使能。作为小信号放大器使用时，主要是用于放大 100mV 以下小信号，可把 Vo11 输出到端口 P4.3 或者 12 位 ADC 的 Ain0 口。运放 A2 类似。



9. 通讯接口

9.1 增强型异步收发 UART

UART 模块具备以下功能：

- 全双工异步收发
- 双字符输入缓冲区
- 单字符输出缓冲区
- 可编程 8 位或 9 位字符长度
- 9 位模式下的地址检测
- 输入缓冲区溢出错误检测
- 接收字符帧错误检测

UART 的发送器和接收器在功能上时相互独立的，但它们的数据格式和波特率相同，UART 先发送和接收 LSB。硬件不支持奇偶校验，但可通过软件实现并作为第 9 位存储。

UART 相关的寄存器：

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
1E6H(r/w)	BAUDCTL			IRXEN	ITXEN			IFRX	IFTX
1E2H(r/w)	RCREG	UART 接收数据寄存器							
1E0H(r/w)	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
1E4H(r/w)	SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0
1E5H(r/w)	SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8
1E1H(r/w)	TXREG	USRT 发送数据寄存器							
1E3H(r/w)	TXSTA	-	TX9	TXEN	SYNC	-	BRGH	TRMT	TX9D

9.1.1 TXSTA：发送状态和控制寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1 (R)	B0
1E3H(r/w)	TXSTA	-	TX9	TXEN	BRGS	-	BRGH	TRMT	TX9D

TX9：9 位发送使能位

=0：选择 8 位发送

=1：选择 9 位发送

TXEN：发送使能位

=0：禁止发送

=1：使能发送

SYNC：异步选择位

=0：异步

=1：无效

BRGH：高波特率选择位

=0：低速

=1：高速



TRMT: 发送移位寄存器状态位

=0: TSR 满

=1: TSR 空

TX9D: 发送数据的第 9 位

可以是地址/数据位或者奇偶校验位

9.1.2 RCSTA: 接收状态和控制寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
1E0H(r/w)	RCSTA	SPEN	RX9	SREN	CREN	-	FERR	OERR	RX9D

SPEN: 串行口使能位

=0: 禁止串行口（保持为复位状态）

=1: 使能串行口（将 RXD 和 TXD 引脚配置为串行口引脚）

RX9: 9 位接收使能位

=0: 选择 8 位接收

=1: 选择 9 位接收

SREN: 单字节接收使能位（异步模式无关位）

CREN: 连续接收使能位

=0: 禁止接收器

=1: 使能接收器

FERR: 帧错误位

=0: 无帧错误

=1: 帧错误（可通过 RCREG 寄存器刷新该位，并接收下一个有效字节）

OERR: 溢出错误位

=0: 无溢出错误

=1: 溢出错误（可通过清零 CREN 位来清零该位）

RX9D: 接收数据的第 9 位

该位可以是地址/数据位或者奇偶校验位，并且必须由用户固件计算得到。

9.1.3 BAUDCTL: UART 中断寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
1E6H(r/w)	BAUDCTL	-	-	IRXEN	ITXEN	-	-	IFRX	IFTX

IRXEN: RX 中断使能

=0: 无效

=1: 有效

ITXEN: TX 中断使能

=0: 无效

=1: 有效

IFRX: RX 中断标志

=0: 无效

=1: 有效



IFTX: TX 中断标志

=0: 无效

=1: 有效

9.1.4 UART 异步发送器

发送器的核心是串行发送移位寄存器（Transmit Shift Register, TRS），该寄存器不可用软件直接访问。TSR 从发送缓冲区 TXREG 取得数据。

9.1.4.1 使能发送器

UART 发送器可通过配置以下 3 个控制位使能为异步操作：

- TXEN=1
- SYNC=0
- SPEN=1

假定所有其他 UART 控制位均处于默认状态。将 TXSTA 寄存器的 TXEN 位置 1 使能 UART 的发送器电路。清零 TXSTA 寄存器的 SYNC 位将 UART 配置为异步操作。将 RCSTA 寄存器的 SPEN 位置 1 使能 UART 并自动将 TXD: I/O 引脚配置为输出。

注：

1.当 SPEN 位置 1 时，RXD: I/O 引脚被自动配置为输入，无论相应 TRIS 位的状态如何以及 UART 接收器是否被使能。通过正常的 PORT 读取操作可读取 RXD 引脚数据，但不包括 PORT 锁存器的数据输出值。

2.TXEN 允许位置 1 时，TXIF 发送器中断标志置 1。

9.1.4.2 发送数据

向 TXREG 寄存器写入一个字符时启动发送。如果这是首字符，或前一个字符被完全从 TSR 中送出，TXREG 中的数据就立即被传送到 TSR 寄存器。如果 TSR 中仍保存前一个字符的全部或部分，则新字符被保存在 TXREG 中，直到前一个字符的停止位被发送。之后，在 TXREG 中等待的字符在停止位发送后 1 个 T_{CY} 后被传送到 TSR 中。TXREG 中的数据被传送到 TSR 后，启动位、数据位和停止位的发送序列立即开始。

9.1.4.3 发送中断标志

只要 UART 发送器被使能，而且 TXREG 中没有等待发送的字符，PIR1 寄存器的 TXIF 中断标志位就被置 1。换句话说，只有在 TSR 中有字符，并且 TXREG 中还有一个排队等待发送的新字符时，TXIF 位才被清零。写入 TXREG 后并不立即清零 TXIF 标志位。执行写操作后的第二个指令周期 TXIF 才有效。写入 TXREG 后立即查询 TXIF 位将返回无效结果。TXIF 位是只读的，不能用软件置 1 或清零。

将 TXEN 中断允许位置 1 可允许 TXIF 中断。但是，只要 TXREG 为空，TXIF 标志位就被置 1，无论 TXEN 允许位的状态如何。

要是发送数据时使用中断，应只在没有数据可发送时才将 TXIE 位置 1。在将发送的最后一个字符写入 TXREG 后应清零 TXEN 中断允许位。

9.1.4.4 TSR 状态

TXSTA 寄存器的 TRMT 位指示 TSR 寄存器的状态。该位是只读位。TSR 寄存器为空时，TRMT 位置 1，而当一个字符从 TXREG 寄存器传送到 TSR 寄存器中时，该位清零。TRMT 位将保持清零，直到所有位移出 TSR 寄存器。该位不与任何终端逻辑相连，因此用户必须查询该位以确定 TSR 的状



态。

注：TSR 寄存器不映射到数据寄存器中，因此用户无法使用。

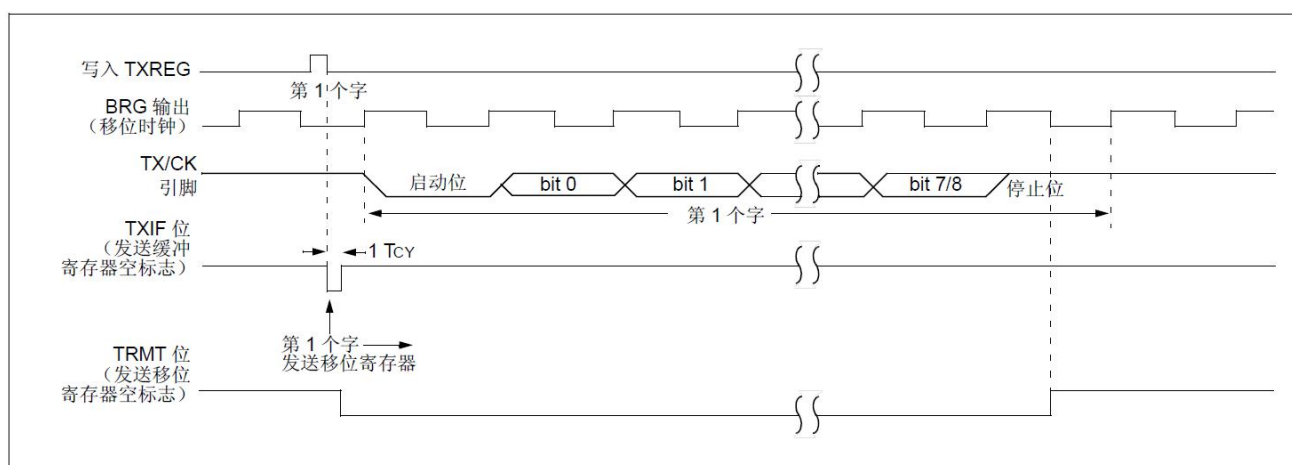
9.1.4.5 发送 9 位字符

UART 支持 9 位字符发送。当 TXSTA 寄存器的 TX9 位置 1 时，UART 将在发送每个字符时移出 9 位。TXSTA 寄存器的 TX9D 位是第 9 个数据位，也是最高有效位。发送 9 位数据时，TX9D 数据位必须先于 8 个最低有效位写入 TXREG。写入 TXREG 后，所有 9 个位将被立即传送到 TSR 移位寄存器中。

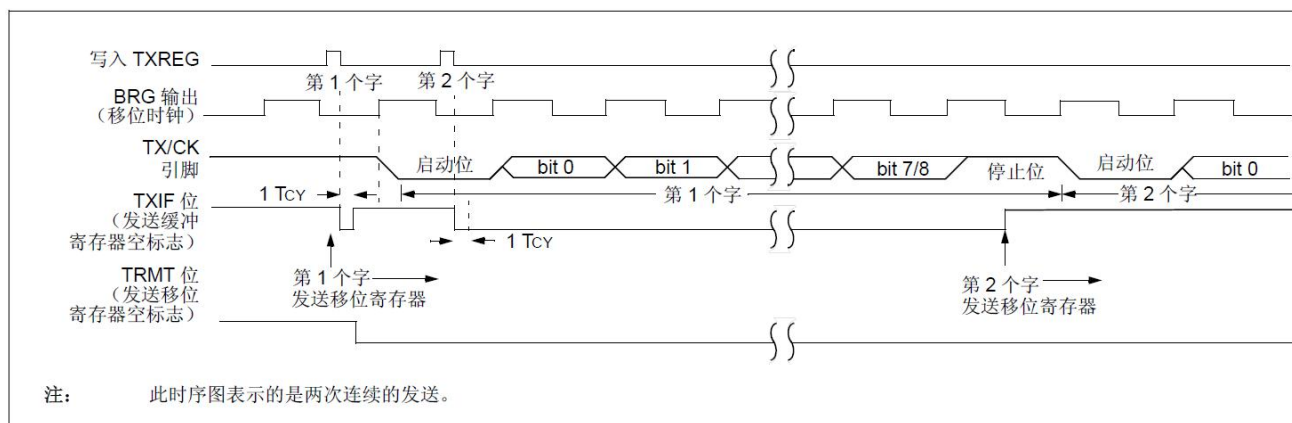
9.1.4.6 异步发送设置：

1. 初始化 SPBRGH、SPBRG 寄存器对以及 BRGH 和 BRG16 位，获得所需的波特率。
2. 清零 SYNC 位并将 SPEN 位置 1，使能异步串行口。
3. 如果需要 9 位发送，将 TX9 控制位值 1。如果第 9 个数据位置 1，则表示发送器置于地址检测时，8 个最低有效位为地址。
4. 将 TXEN 控制位置 1 使能发送。这将导致 TXIF 中断位置 1。
5. 如果需要中断，将 PIE1 寄存器中的 TXIE 中断允许位置 1。如果 INTCON 寄存器的 GIE 和 PEIE 位也置 1，则立即产生中断。
6. 如果选择了 9 位发送，应将第 9 位装入 TX9D 数据位。
7. 将 8 位数据装入 TXREG 寄存器。这将启动发送。

9.1.4.7 UART 异步发送：



9.1.4.8 UART 异步发送->背对背：



注：此时序图表示的是两次连续的发送。



9.1.5 UART 异步接收器

异步模式通常用于 RS-232 系统中。数据在 RX/TD 引脚上接收并驱动数据恢复模块。数据恢复模块实际上是一个高速移位器，工作频率为 16 倍波特率，而串行接收移位器（Receive Shift Register, RSR）的工作频率为波特率。所有 8 位或 9 位字符移入后就立即传送到双字符的先进先出（First-In-First-Out, FIFO）存储器中。FIFO 缓冲区允许先接收两个完整字符和第三个字符的开始部分后，再启动软件服务 UART 接收器。FIFO 和 RSR 寄存器不能直接用软件访问。通过 RCREG 寄存器访问接收数据。

9.1.5.1 使能接收器

UART 接收器可通过配置以下 3 个控制位使能为异步操作：

- CREN=1
- SYNC=0
- SPEN=1

假定所有其他 UART 控制位均处于其默认状态。将 RCSTA 寄存器的 CREN 位置 1 使能 UART 的接收器电路。清零 TXSTA 寄存器的 SYNC 位将 UART 配置为异步操作。将 RCSTA 寄存器的 SPEN 位置 1 使能 UART 并自动将 RXD I/O 引脚配置为输入。如果 RXD 引脚与模拟外设共用，那必须通过清零相应的 ANSEL 位禁止模拟 I/O 功能。

注：当 SPEN 位置 1 时，TXD I/O 引脚被自动配置为输出，无论相应 TRIS 位的状态如何以及 UART 发送器是否被使能。PORT 锁存器与输出驱动器断开，因此不可能将 TXD 引脚用作通用输出。

9.1.5.2 接收数据

接收器的数据恢复电路在第一位的下降沿启动字符接收。第一位也称启动（Start）位，始终为零。数据恢复电路计数一个半位时间至启动位的中点并验证该位是否仍为零。如果该位非零则数据恢复电路中止字符接收，不产生错误，并恢复寻找启动位的下降沿。如果启动位被验证为零，则数据恢复电路计数一整个位时间至下一个位的中点。该位被一个择多检测电路采样，其结果（0 或 1）被移入 RSR。重复此过程直到所有数据位均被采样并移入 RSR。最后一个位时间被测量且其电平被采样。此位停止（Stop）位，始终为 1。如果数据恢复电路在停止位处采样到 0，则产生此字符的帧错误，否则此字符的帧错误被清零。

所有数据位和停止位被接收后，RSR 中的字符就被立即传送到 UART 接收 FIFO，且 PIR1 寄存器的 RCIF 中断标志位被置 1。读取 RCREG 寄存器时，FIFO 中顶部的字符被送出 FIFO。

注：如果接收 FIFO 溢出，在溢出条件被清楚前不会接收更多字符。

9.1.5.3 接收中断

只要 UART 接收器被使能且接收 FIFO 中存在未被读取的字符，PIR1 寄存器的 RCIF 中断标志位就会被置 1。RCIF 中断标志位是只读位，不能用软件 1 或清零。

将以下所有位置 1 可允许 RCIF 中断：

- PIE 寄存器的 RCIE 中断允许位
- INTCON 寄存器的 PEIE 外设中断允许位
- INTCON 寄存器的 GIE 全局中断允许位

当 FIFO 中存在未被读取的字符时，无论中断允许位的状态如何，RCIF 中断标志位均会被置 1。

9.1.5.4 接收帧错误

接收 FIFO 缓冲区中的每个字符都有相应的帧错误状态位。帧错误表明在预期时间内未见到停止



位。通过 RCSTA 寄存器的 FERR 位可访问帧错误状态。FERR 位表示接收 FIFO 中顶部的未读字符的状态，因此，在读取 RCREG 前必须读出 FERR 位。

FERR 位是只读位，只用于接收 FIFO 中顶部的未读字符。帧错误（FERR=1）不排除接收额外字符。此时不必将 FERR 位清零。从 FIFO 缓冲区读出下一个字符。将使 FIFO 进入下一个字符和下一个相应的帧错误。

将 RCSTA 寄存器的 SPEN 位清零可复位 UART，这样就可将 FERR 位强制清零。将 RCSTA 寄存器的 CREN 位清零不影响 FERR 位。自身产生的帧错误不会产生中断。

注：如果接收 FIFO 中的所有接收字符均有帧错误，反复读取 RCREG 不会将 FERR 位清零。

9.1.5.5 接收溢出错误

接收 FIFO 缓冲区可容纳两个字符。在访问 FIFO 前接收到完整的第三个字符时会产生溢出错误。此时，RCSTA 寄存器的 OERR 位置 1。FIFO 缓冲区已有的字符可被读出，但溢出错误被清除前不能再接收其他字符。将 RCSTA 寄存器的 CREN 位清零或通过将 RCSTA 寄存器的 SPEN 位清零复位 UART，可清除该错误。

9.1.5.6 接收 9 位字符

UART 支持 9 位字符接收。当 RCSTA 寄存器 RX9 位置 1 时，UART 将在接收每个字符时将 9 个位移入 RSR。RCSTA 寄存器的 RX9D 位是第 9 位，也是接收 FIFO 顶部未读字符的最高有效位。从接收 FIFO 缓冲区读取 9 位数据时，在读取 RCREG 的 8 个最低有效位前必须先读取 RX9D 数据位。

9.1.5.7 地址检测

当多个接收器共用同一条发送线时，将 RCSTA 寄存器的 ADDEN 位置 1 可使能地址检测。

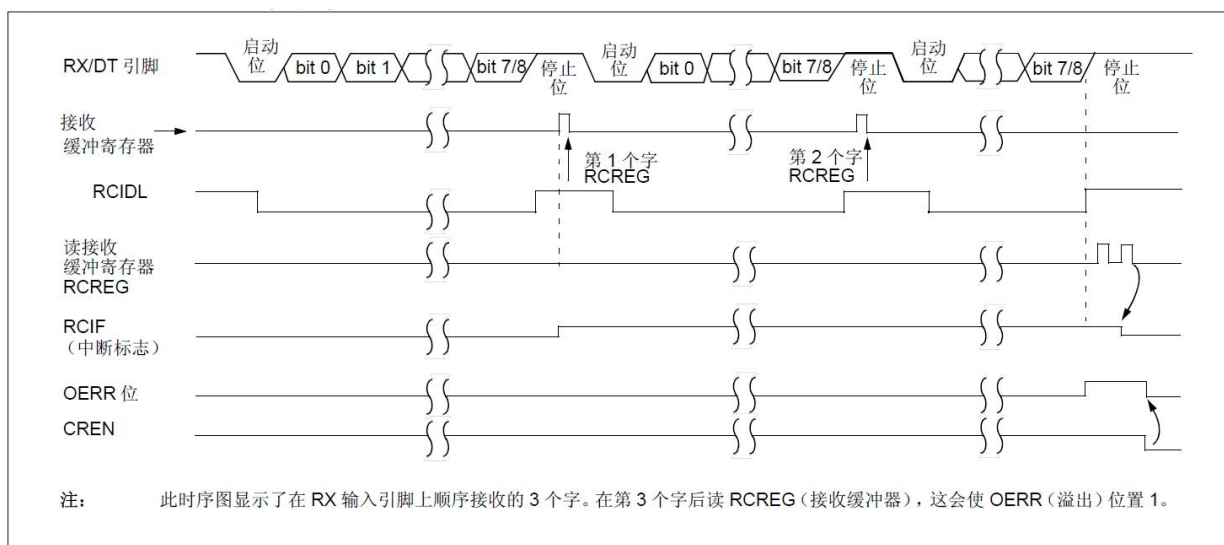
地址检测要求接收 9 位字符。使能地址检测时，只有第 9 个数据位置 1 的字符会被传送到接收 FIFO 缓冲区，并将 RCIF 中断位置 1。所有其他字符均被忽略。

接收到地址字符后，用户软件可决定地址是否与自身匹配。地址匹配时。发生下一个停止位前，用户软件必须用过清零 ADDEN 位禁止地址检测。

9.1.5.8 异步接收设置：

1. 初始化 SPBRGH 和 SPBRG 寄存器对以及 BRGH 和 BRG16 位，获得所需的波特率。
2. 将 SPEN 位置 1 使能串行口。SYNC 位必须清零才能进行异步操作。
3. 如果需要中断，将 PPIE1 寄存器中的 RCIE 位置 1，并将 INTCON 寄存器的 GIE 和 PEIE 位置 1。
4. 如果需要接收 9 位，将 RX9 位置 1。
5. 将 CREN 位置 1 使能接收。
6. 当字符从 RSR 被移入接收缓冲区时，RCIF 中断标志位将被置 1。如果 RCIE 中断允许位也置 1，则产生中断。
7. 读取 RCSTA 寄存器取得错误标志，以及第 9 个数据位（9 位诗句接收使能时）。
8. 读取 RCREG 寄存器从接收缓冲区取得接收的 8 个最低有效位。
9. 发生溢出时，通过清零 CREN 接收器使能位清零 OERR 标志位。

9.1.5.9 UART 异步接收：



9.1.6 UART 波特率发生器（BRG）

波特率发生器（BRG）是8位或16位定时器，专用于支持异步UART操作。默认情况下，BRG工作在8位模式下，将BAUDCTL寄存器的BRG16位置1可选择16位模式。

SPBRGH和SPBRG寄存器对决定自动运行波特率定时器的周期。在异步模式下，波特率周期的乘数由TXSTA寄存器的BRGH位和BAUDCTL寄存器的BRG16位决定。

波特率公式：

配置位			BRG/UART 模式	波特率公式
SYNC	BRG16	BRGH		
0	0	0	8位/异步	$F_{osc}/[64(n+1)]$
0	0	1	8位/异步	$F_{osc}/[16(n+1)]$
0	1	0	16位/异步	
0	1	1	16位/异步	$F_{osc}/[4(n+1)]$

图注：n=SPBRGH和SPBRG寄存器对的值：

确定波特率和波特率误差的计算示例：

器件工作在 $F_{osc} = 16\text{MHz}$ ，目标波特率 = 9600，异步模式,8位BRG：

$$\text{目标波特率} = \frac{F_{osc}}{64([SPBRGH : SPBRG] + 1)}$$

求解 SPBRGH 和 SPBRG：

$$X = \frac{F_{osc}}{64 \times \text{目标波特率}} - 1 = \frac{16000000}{64 \times 9600} - 1 = [25.042] = 25 (\text{十进制整数})$$

$$\text{计算波特率} = \frac{16000000}{64(25 + 1)} = 9615$$



$$\text{误差} = \frac{\text{计算波特率} - \text{目标波特率}}{\text{目标波特率}} = \frac{(9615 - 9600)}{9600} = 0.16\%$$

为了便于您使用，各种异步模式的典型波特率和误差值已经计算出来。使用高波特率（BRGH=1）或 16 位 BRG（BRG16=1）有助于降低波特率误差。16 位 BRG 模式用于在高速振荡器频率下取得较缓慢的波特率。

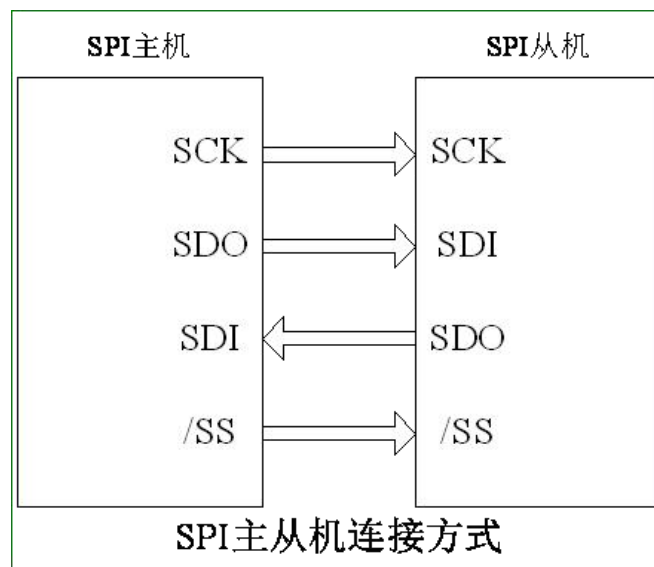
将新值写入 SPBRGH 和 SPBRG 寄存器将导致 BRG 定时器复位（或清零）。这可以确保 BRG 无需等待定时器溢出就可以输出新的波特率。

如果在有效接收操作过程中更改系统时钟，可能会导致接收错误或数据丢失。为避免此问题，应检查 RCIDL 位的状态，以确保在更改系统时钟前接收操作处于空闲状态。

9.2 串行外部设别接口 SPI

9.2.1 SIM（SIM 控制器）

SPI 接口是一个全双工串行数据传输器。SPI 接口的四线为：SDI、SDO、SCK 和 /SS。SDI 和 SDO 是数据的输入和输出线。SCK 是串行时钟线，/SS 是从机的选择线。SPI 的接口引脚与普通 I/O 口。通过设定 SIMC0、SIMC1 寄存器的对应位，来使能 SPI 接口。连接到 SPI 接口的单片机以主/从模式进行通信，且主机完成所有的数据传输初始化，并控制时钟信号。由于单片机只有一个 /SS 引脚，所以只能拥有一个从机设备。可通过软件控制 /SS 引脚使能，设置 SSEN 位为“1”使能 /SS 引脚功能，设置 SSEN 位为“0”/SS 引脚将作为普通输入/输出引脚。



SPI 功能具有以下特点：

全双工同步数据传输

主从模式

最低有效位先传或最高有效位先传的数据传输模式

传输完成标志位

时钟源上升沿或下降沿有效



SPI 相关的寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
1E7H(r/w)	SIMC0	SIMM2	SIMM1	SIMM0	-	-	-	SPIEN	-
1E8H(r/w)	SIMC1			CKEG	CKP	MLS	SSEN	-	TRF
1EAH(r/w)	SIMD	SPI 缓冲器							

9.2.2 SIMD (SIM 的缓冲器)

SIM 的缓冲器，用以存放接收和发送的数据。在单片机尚未将数据写入到 SPI 总线中时，要传输的数据应先存在 SIMBUF 寄存器中。SPI 总线接收到数据之后，单片机就可以从 SIMBUF 寄存器中读取。所有通过 SPI 总线传输或接收的数据都必须通过 SIMBUF 寄存器实现。

9.2.3 SIMC0 (SIM 控制器 0)

SIMM2~SIPM0: SIM 工作模式控制位

- =000: SPI 主机模式; SPI 时钟为 Fsys/4
- =001: SPI 主机模式; SPI 时钟为 Fsys/16
- =010: SPI 主机模式; SPI 时钟为 Fsys/64
- =011: SPI 主机模式; SPI 时钟为 TC0/2
- =100: 未使用模式
- =101: SPI 从机模式
- =110: 未使用模式
- =111: 未使用模式

这几位用于设置 SPI 功能的工作模式，用于选择 SPI 的主从模式和 SPI 的主机时钟频率。SPI 时钟源可来自于系统时钟也可以选择来自 TMR0。若选择的是作为 SPI 从机，则其时钟源从外部主机而得。

SPIEN: SIM 控制位。

- =0: 无效
- =1: 有效

此位为 SIM 接口的开/关控制位。选择 SPI 模式时，此位为“0”时，SPI 接口除能，SDI、SDO、SCK 和 SS 脚用于输入/输出功能，SPI 工作电流减小到最小值。此位为“1”时，SPI 接口使能。配置选项中首先将 SPI 接口使能才能使此位有效。若 SPI 经由 SIMM2~SIMM0 位设置为工作在 SPI 接口，当 SIMEN 位由低到高转变时，SPI 控制寄存器中的设置不会发生变化，其首先应在应用程序中初始化。

9.2.4 SIMC1 (SIM 控制器 1)

CKEG: SPI 的 SCK 有效时钟边沿类型位

CKP=0 :

- =0: SCK 为高电平且在 SCK 上升沿抓取数据
- =1: SCK 为高电平且在 SCK 下降沿抓取数据

CKP=1 :

- =0: SCK 为低电平且在 SCK 下降沿抓取数据



=1: SCK 为低电平且在 SCK 上升沿抓取数据

CKEG 和 CKP 位用于设置 SPI 总线上时钟信号输入和输出方式。在执行数据传输前，这两位必须被设置，否则将产生错误的时钟边沿信号。CKP 位决定时钟线的基本状态，若时钟无效且此位为高，则 SCK 为低电平，若时钟无效且此位为低，则 SCK 为高电平。CKEG 位决定有效时钟边沿类型，取决于 CKP 的状态。

CKP: 时钟线的基础状态位

=0: 当时钟无效时，SCK 引脚为高电平

=1: 当时钟无效时，SCK 引脚为低电平

此位决定了时钟线的基础状态，当时钟无效时，若此位为高，SCK 为低电平，若此位为低，SCK 为高电平。

MLS: SPI 数据移位命令位

=0: LSB

=1: MSB

数据移位选择位，用于选择数据传输时高位优先传输还是低位优先传输。此位设置为高时高位优先传输，为低时低位优先传输。

SSEN: SPI /SS 引脚控制位

=0: 除能

=1: 使能

SSEN 位用于/SS 引脚的使能/除能控制。此位为低时，/SS 除能并处于浮空状态。此位为高时，/SS 作为选择脚。

注:在主机模式下，SSEN=0，则/SS 脚做 I/O 口使用。SSEN=1，则/SS 脚在 SPIEN=1 时，变低。

在从机模式下，SSEN=0，则 SPI 模块不受控与/SS 脚，不论/SS 脚为高还是为低都可以接收数据。此时/SS 端可做 I/O 口使用。在 SSEN=1 时，SPI 模块必须在/SS 脚为低时才会接收数据。

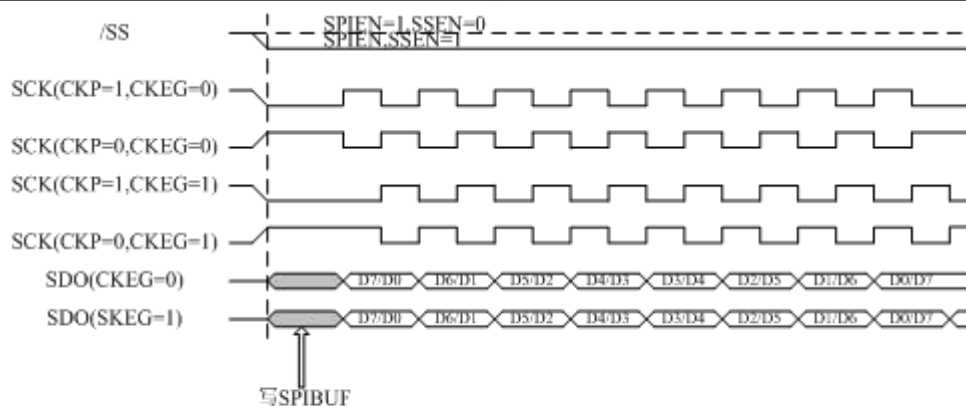
TRF: SPI 发送/接收数据结束标志位

=0: 正在发送

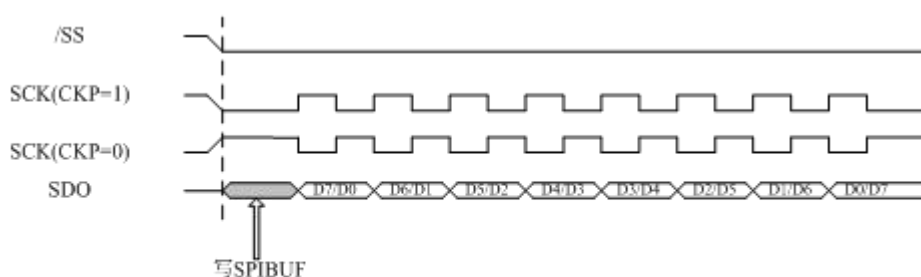
=1: 发送结束

9.2.5 SPI 通信

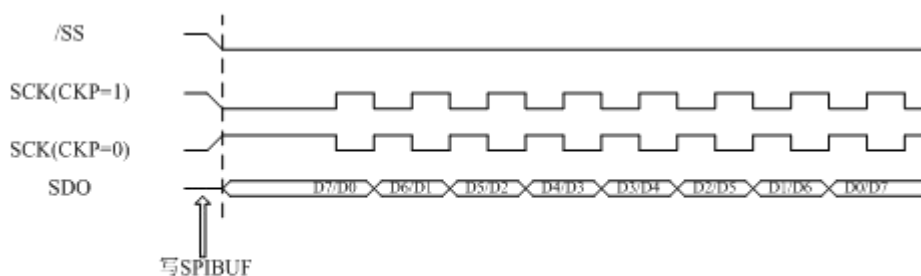
将 SPIEN 设置为高，使能 SPI 功能之后，单片机处于主机模式，当数据写入到寄存器 SIMBUF 的同时传输/接收开始进行。数据传输完成时，TRF 位将自动被置位但清除只能通过应用程序完成。单片机处于从机模式时，收到主机发来的信号之后，会传输 SIMBUF 中的数据，而且在 SDI 引脚上的数据也会被移到 SIMBUF 寄存器中。主机应在输出时钟信号之前先输出一个/SS 信号以使能从机，从机的数据传输功能也应在与/SS 信号相关的适当时候准备就绪，这由 CKP 和 CKEG 位决定。所附时序图表明了 CKP 和 CKEG 位各种设置情况下从机数据与/SS 信号的关系。



SPI主机模式时序



SPI从机模式时序 (CKEG=0)



SPI从机模式时序 (CKEG=1)

注：使用 SPI 前，应先定义 CKEG,CKP，然后才使能 SPIEN 信号。

9.3 INTFACES（通信管脚控制器）

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
1EDH(r/w)	INTFACES	-	SPIIN	SCS_	SPICLK	-	-	TXOUT	RXIN

RXIN: UART 数据输入控制位

=0: 输入高电平

=1: P46 作为 UART 输入脚

TXOUT: UART 数据输出控制位

=0: P47 为普通 IO 口



=1: P47 作为 UART 输出脚

SPICLK: SPI 时钟控制信号

=0: P44 为 SPI 时钟信号, 同时 P47 为 SPI 数据输出脚

=1: SPI 从机接收时钟为高

SCS_: SPI 的/SS 从机的选择线控制信号

=0: P43 作为/SS 信号

=1: /SS 信号为高

SPIIN: SPI 数据控制信号

=0: P46 为 SPI 数据输入信号

=1: SPI 数据输入信号为高



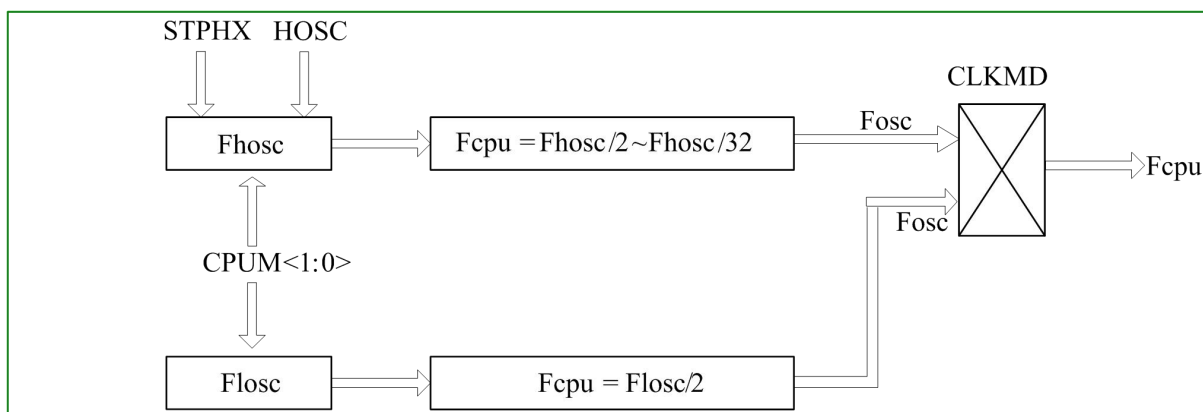
10. 系统时钟

10.1 概述

HS26F2728内置双时钟系统：高速时钟和低速时钟。低速时钟由内部低速振荡器提供，由OSCM寄存器的CLKMD位控制，高、低速时钟都可以作为系统时钟源。HS26F2728同时还配置了外部晶振，可选择32.768K、4M、16M外置晶振模式作为系统时钟，并且增加了RTC模式。

内部高速振荡器：Config配置可选14/28MHz，称为IHRC；

内部低速振荡器：一个32KHz（ILRC）以及一个可变（7.8~250K），称为ILRC1。



HOSC: High_Clk 编译选项。

Fhosc: 外部高速/内部 RC 振荡器时钟频率。

Fosc: 内部低速 RC 时钟频率。

Fosc: 系统时钟频率。

Fcpu: 指令执行频率。

10.2 指令周期 Fcpu

系统时钟速率，即指令周期（Fcpu），从系统时钟源分离出来，决定系统的工作速率。Fcpu的速率由Fcpu编译选项决定，正常模式下， $Fcpu = Fhosc/2 \sim Fhosc/32$ 。若高速时钟源为内部14MHz振荡器，则Fcpu编译选项选择Fhosc/4，则Fcpu频率为14MHz/4=3.5MHz。低速模式下， $Fcpu = Fosc/2$ ，即32KHz/2=16KHz。

高干扰环境下，强烈建议设置 code option 的 Fcpu 选项在 Fhosc/4 以下，以减少干扰的影响。



11. 系统工作模式

11.1 概述

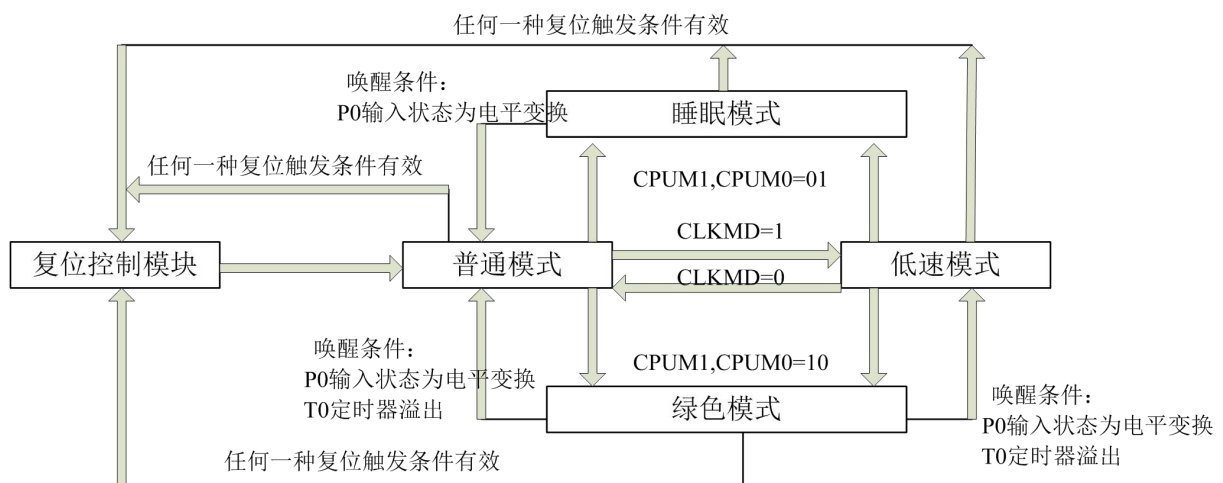
HS26F2728可以在4种工作模式下以不同的时钟频率工作，这些模式可以控制振荡器的工作、程序的执行以及模拟电路的功耗。

普通模式：系统高速工作模式；

低速模式：系统低速工作模式；

省电模式：系统省电模式（睡眠模式）；

绿色模式：系统理想模式。



工作模式	普通模式	低速模式	绿色模式	睡眠模式
IHRC	运行	STPHX 控制	STPHX 控制	停止
ILRC	运行	运行	运行	运行
CPU 指令	执行	执行	停止	停止
TC0	TC0TR 控制	TC0TR 控制	TC0TR 控制	停止
T0	T0TR 控制	T0TR 控制	T0TR 控制	停止
T2	T2TR 控制	T2TR 控制	T2TR 控制	T2TR 控制
看门狗定时器	WDT 控制	WDT 控制	WDT 控制	WDT 控制
中断	全部有效	全部有效	全部有效	全部有效
唤醒功能	——	——	TC0/T0/T2/端口变化/外中断/复位/按键扫描/COMP	T2/端口变化/外中断/复位/按键扫描/COMP

IHRC：内部高速时钟（RC 振荡器）。

ILRC：内部低速时钟（RC 振荡器）。



11.2 普通模式

普通模式是系统高速时钟正常工作模式，系统时钟源由高速振荡器提供，程序执行。上电复位或任意一种复位触发后，系统进入普通模式执行程序。当系统从睡眠模式被唤醒后进入普通模式。普通模式下，高速振荡器正常工作，功耗最大。

程序被执行，所有的功能都可控制。

系统速率为高速。

高速振荡器和内部低速RC振荡器都正常工作。

通过OSCM寄存器，系统可以从普通模式切换到其它任何一种工作模式。

系统从睡眠模式唤醒后进入普通模式。

低速模式可以切换到普通模式。

从普通模式切换到绿色模式，唤醒后返回到普通模式。

11.3 低速模式

低速模式为系统低速时钟正常工作模式。系统时钟源由内部低速RC振荡器提供。低速模式由OSCM寄存器的CLKMD位控制。当CLKMD=0时，系统为普通模式；当CLKMD=1时，系统进入低速模式。切换进入低速模式后，不能自动禁止高速振荡器，必须通过SPTHX位来禁止以减少功耗。低速模式下，系统时钟默认选用FLOSC（32K）。

程序被执行，所有的功能都可控制。

系统速率为低速，可选分频系数。

内部低速RC振荡器正常工作，高速振荡器由STPHX=1控制。低速模式下，强烈建议停止高速振荡器，但必须先进入低速模式后，再STPHX=1。

通过OSCM寄存器，低速模式可以切换进入其它的工作模式。

从低速模式切换到睡眠模式，唤醒后返回到低速模式或者正常模式。

普通模式可以切换进入低速模式。

从低速模式切换到绿色模式，唤醒后返回到低速模式。

11.4 睡眠模式

睡眠模式是系统的理想状态，不执行程序，振荡器也停止工作。整个芯片的功耗低于1uA（ADC关闭）。睡眠模式的唤醒源为P0/P1/P4电平变换触发，T2溢出，ADC转换完成、ADC滤波条件满足、外中断、按键扫描、比较器、REST或者WDT溢出复位。

由OSCM寄存器的CPUM0位控制是否进入睡眠模式，当CPUM0=1，系统进入睡眠模式。当系统从睡眠模式被唤醒后，CPUM0被自动禁止（0状态）。

程序停止执行。

外部高速振荡器、内部高速振荡器都停止工作。

功耗低于1uA。

如果需要，内部低速振荡器可工作

系统从睡眠模式被唤醒后进入普通模式或低速模式。



11.5 绿色模式

绿色模式是另外的一种理想状态。在睡眠模式下，所有的功能和硬件设备都被禁止，但在绿色模式下，系统时钟保持工作，绿色模式下的功耗大于睡眠模式下的功耗。绿色模式下，不执行程序，但具有唤醒功能的定时器仍正常工作，定时器的时钟源为仍在工作的系统时钟。绿色模式的唤醒源和睡眠模式一样。

由OSCM寄存器CPUM1位决定是否进入绿色模式，当CPUM1=1，系统进入绿色模式。当系统从绿色模式下被唤醒后，自动禁止CPUM1（0状态）。

程序停止执行，所有的功能被禁止。

具有唤醒功能的定时器正常工作。

作为系统时钟源的振荡器正常工作，其它的振荡器工作状态取决于系统工作模式的配置。

由普通模式切换到绿色模式，被唤醒后返回到普通模式。

由低速模式切换到绿色模式，被唤醒后返回到低速模式。

绿色模式下的唤醒方式和睡眠模式唤醒一样。

绿色模式下PWM和Buzzer功能仍然有效，但是定时器溢出时不能唤醒系统。

11.6 系统唤醒

11.6.1 概述

睡眠模式和绿色模式下，系统并不执行程序。唤醒触发信号可以将系统唤醒进入普通模式或低速模式。

从睡眠模式唤醒后返回到上一个工作模式（普通模式或低速模式）；唤醒触发信号包括：

- 1、外部触发信号（P0/P1/P4的电平变换，对应的PnW需要置1）
- 2、内部触发（T2定时器溢出）
- 3、INT0/INT1/INT2（INT0IE，INT1IE需开启）
- 4、按键扫描唤醒。
- 5、比较器
- 6、REST脚复位或者WDT溢出复位唤醒

从绿色模式唤醒后返回到上一个工作模式（普通模式或低速模式）；唤醒触发信号包括：

- 1、外部触发信号（P0/P1/P4的电平变换，对应的PnW需要置1）
- 2、内部触发（TC0、T0、T2定时器溢出）
- 3、INT0/INT1/INT2（INT0IE，INT1IE需开启）
- 4、按键扫描唤醒。
- 5、比较器
- 6、REST脚复位或者WDT溢出复位唤醒

11.6.2 唤醒时间

系统进入睡眠模式后，高速时钟振荡器停止运行。把系统从睡眠模式唤醒时，单片机需要等待一段时间以等待振荡电路稳定工作，等待的这段时间就称为唤醒时间。唤醒时间结束后，系统进入普



通模式。

注：从绿色模式下唤醒系统不需要唤醒时间，因为系统时钟在绿色模式下仍然正常工作。

内部高速 RC 振荡器的唤醒时间的计算如下：

唤醒时间 = $1/F_{\text{Hosc}} * 2048(\text{sec})$ + 高速时钟启动时间

例：睡眠模式下，系统被唤醒进入普通模式。唤醒时间的计算如下：

唤醒时间 = $1/F_{\text{Hosc}} * 2048$ ($F_{\text{Hosc}} = 14\text{MHz}$)

若睡眠之前的状态为低速模式，并且 TL0M.6=1，则唤醒计算时间大约 2ms

注：高速时钟的启动时间与 VDD 和振荡器类型有关。



12. 复位

12.1 概述

HS26F2728有以下几种复位方式：

上电复位；

看门狗复位；

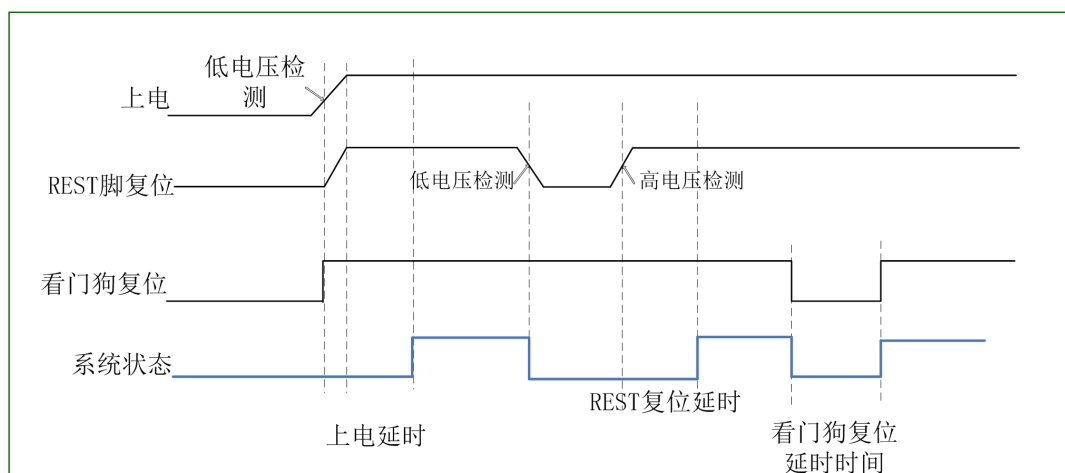
掉电复位；

REST脚复位；

上述任一种复位发生时，所有的系统寄存器恢复默认状态，程序停止运行，同时程序计数器PC清零。复位结束后，系统从向量0000H处重新开始运行。PFLAG寄存器的/T0和/PD两个标志位能够给出系统复位状态的信息。用户可以编程控制/T0和/PD，从而控制系统的运行路径。

/T0	/PD	复位情况	说明
0	0	看门狗复位	看门狗溢出
0	1	保留	——
1	0	上电及 LVD 复位	电源电压低于 LVD 检测值
1	1	REST 管脚复位	P0.4 管脚复位

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。对于不同类型的振荡器，完成复位所需要的时间也不同。因为，VDD 的上升速度和不同晶振的起振时间都不固定。RC 振荡器的起振时间最短，晶体振荡器的起振时间则较长。在用户终端使用的过程中，应注意考虑主机对上电复位时间的要求。



12.2 上电复位

上电复位与LVD操作密切相关。系统上电的过程呈逐渐上升的曲线形式，需要一定时间才能达到正常电平值。下面给出上电复位的正常时序：

上电：系统检测到电源电压上升并等待其稳定；

系统初始化：所有的系统寄存器被置为初始值；

振荡器开始工作：振荡器开始提供系统时钟；



执行程序：上电结束，程序开始运行。

12.3 看门狗复位

看门狗复位是系统的一种保护设置。在正常状态下，由程序将看门狗定时器清零。若出错，系统处于未知状态，看门狗定时器溢出，此时系统复位。看门狗复位后，系统重启进入正常状态。看门狗复位的时序如下：

看门狗定时器状态：系统检测看门狗定时器是否溢出，若溢出，则系统复位；

系统初始化：所有的系统寄存器被置为默认状态；

振荡器开始工作：振荡器开始提供系统时钟；

执行程序：上电结束，程序开始运行。

看门狗定时器应用注意事项：

对看门狗清零之前，检查I/O口的状态和RAM的内容可增强程序的可靠性；

不能在中断中对看门狗清零，否则无法检测到主程序跑飞的状况；

程序中应该只在主程序中有一次清看门狗的动作，这种架构能够最大限度的发挥看门狗的保护功能。

不论系统是否处于何种模式（普通模式、睡眠模式等），一旦WDT溢出，则系统会复位。

12.4 REST 脚复位（P04）

外部复位（仅限于外部复位引脚使能状态）：系统检测外部复位引脚状态。如果不为高电平，系统保持复位状态直到外部复位引脚释放。



13. 指令集

序号	助记符号	描述	C	DC	Z	CYC
1	MOV A, M	$A \leftarrow M$	—	—	√	1
2	MOV M, A	$M \leftarrow A$	—	—	—	1
3	B0MOV A, M	$A \leftarrow M$ (bank0)	—	—	√	1
4	B0MOV M, A	M (bank0) $\leftarrow A$	—	—	—	1
5	MOV A, I	$A \leftarrow I$	—	—	—	1
6	B0MOV M, I	$M \leftarrow I$, (M= R,X,Y,Z,H,L,RBANK,PFLAG) M 地址位 0x80--0x87 对应 0x30--0x37	—	—	—	1
7	XCH A, M	$A \leftrightarrow M$	—	—	—	1+N
8	B0XCH A, M	$A \leftrightarrow M$ (bank0), M 不能是 80--87H 的系统寄存器	—	—	—	1+N
9	MOVC	$R, A \leftarrow ROM[Y, Z]$	—	—	—	2

10	ADC A, M	$A \leftarrow M + A + C$, 如果发生进位, 则 C=1, 否则 C=0	√	√	√	1
11	ADC M, A	$M \leftarrow M + A + C$, 如果发生进位, 则 C=1, 否则 C=0	√	√	√	1+N
12	ADD A, M	$A \leftarrow M + A$ 如果发生进位, 则 C=1, 否则 C=0	√	√	√	1
13	ADD M, A	$M \leftarrow M + A$ 如果发生进位, 则 C=1, 否则 C=0	√	√	√	1+N
14	B0ADD M, A	M (bank0) $\leftarrow M$ (bank0) + A if 进位 C=1, else C=0	√	√	√	1+N
15	ADD A, I	$A \leftarrow A + I$, 如果发生进位, 则 C=1, 否则 C=0	√	√	√	1
16	SBC A, M	$A \leftarrow A - M / C$, 如果发生进位, 则 C=0, 否则 C=1	√	√	√	1
17	SBC M, A	$M \leftarrow A - M / C$, 如果发生进位, 则 C=0, 否则 C=1	√	√	√	1+N
18	SUB A, M	$A \leftarrow A - M$, 如果发生进位, 则 C=0, 否则 C=1	√	√	√	1
19	SUB M, A	$M \leftarrow A - M$, 如果发生进位, 则 C=0, 否则 C=1	√	√	√	1+N
20	SUB A, I	$A \leftarrow A - I$, 如果发生进位, 则 C=0, 否则 C=1	√	√	√	1
21	DAA	调整 ACC 十六进制为十进制	√	—	—	1
22	MUL A, M	$R, A \leftarrow A * M$ 计算结果低位存入 ACC, 高位存入 R 寄存器	—	—	√	2
23	AND A, M	$A \leftarrow A \& M$	—	—	√	1
24	AND M, A	$M \leftarrow A \& M$	—	—	√	1+N
25	AND A, I	$A \leftarrow A \& I$	—	—	√	1
26	OR A, M	$A \leftarrow A M$	—	—	√	1
27	OR M, A	$M \leftarrow A M$	—	—	√	1+N
28	OR A, I	$A \leftarrow A I$	—	—	√	1
29	XOR A, M	$A \leftarrow A \oplus M$	—	—	√	1
30	XOR M, A	$M \leftarrow A \oplus M$	—	—	√	1+N
31	XOR A, I	$A \leftarrow A \oplus I$	—	—	√	1
32	COM M	$A \leftarrow M$ 取反	—	—	√	1
33	COMM M	$M \leftarrow M$ 取反	—	—	√	1
34	CLR M	$M \leftarrow 0$	—	—	—	1
35	SWAP M	$A(b3 \sim b0, b7 \sim b4) \leftrightarrow M(b7 \sim b4, b3 \sim b0)$	—	—	—	1
36	SWAPM M	$M(b3 \sim b0, b7 \sim b4) \leftrightarrow M(b7 \sim b4, b3 \sim b0)$	—	—	—	1+N
37	RRC M	$A \leftarrow RRC M$	√	—	—	1
38	RRCM M	$M \leftarrow RRC M$	√	—	—	1+N
39	RLC M	$A \leftarrow RLC M$	√	—	—	1



40	RLCM	M	$M \leftarrow \text{RLC } M$	√	—	—	1+N
41	BCLR	M.b	$M.b \leftarrow 0$	—	—	—	1+N
42	BSET	M.b	$M.b \leftarrow 1$	—	—	—	1+N
43	B0BCLR	M.b	$M(\text{bank0}).b \leftarrow 0$	—	—	—	1+N
44	B0BSET	M.b	$M(\text{bank0}).b \leftarrow 1$	—	—	—	1+N
45	CMPRS	A, I	$ZF, C \leftarrow A-I$, if $A=I$, 则跳过下一条程序	√	—	√	1+S
46	CMPRS	A, M	$ZF, C \leftarrow A-M$, if $A=M$, 则跳过下一条程序	√	—	√	1+S
47	INCS	M	$A \leftarrow M+1$, if $A=0$, 则跳过下一条程序	—	—	—	1+S
48	INCMS	M	$M \leftarrow M+1$, if $M=0$, 则跳过下一条程序	—	—	—	1+S+N
49	INC	M	$A \leftarrow M+1$	—	—	√	1
50	INCM	M	$M \leftarrow M+1$	—	—	√	1+N
51	DECS	M	$A \leftarrow M-1$, if $A=0$, 则跳过下一条程序	—	—	—	1+S
52	DECMS	M	$M \leftarrow M-1$, if $M=0$, 则跳过下一条程序	—	—	—	1+S+N
53	DEC	M	$A \leftarrow M-1$	—	—	√	1
54	DECM	M	$M \leftarrow M-1$	—	—	√	1+N
55	BTS1	M.b	if $M.b=1$, 则跳过下一条程序	—	—	—	1+S
56	B0BTS0	M.b	If $M(\text{bank0}).b=0$, 则跳过下一条程序	—	—	—	1+S
57	B0BTS1	M.b	If $M(\text{bank0}).b=1$, 则跳过下一条程序	—	—	—	1+S
58	TS0M	M	If $M=0, Z=1$ 否则 $Z=0$	-	-	√	1
59	JMP	d	$PC\ 15/14 \leftarrow \text{ROMPAGES}1/0, PC13 \sim PC0 \leftarrow d$	—	—	—	2
60	CALL	d	$\text{Stack} \leftarrow PC15 \sim PC0$ $15/14 \leftarrow \text{RomPAGES}1/0, PC13 \sim PC0 \leftarrow d$	—	—	—	2
61	CALLHL		$\text{Stack} \leftarrow PC15 \sim PC0, PC\ 15 \sim 8 \leftarrow H, PC7 \sim PC0 \leftarrow L$	—	—	—	2
62	CALLYZ		$\text{Stack} \leftarrow PC15 \sim PC0, PC\ 15 \sim 8 \leftarrow Y, PC7 \sim PC0 \leftarrow Z$	—	—	—	2
63	RET		$PC \leftarrow \text{Stack}$	—	—	—	2
64	RETI		$PC \leftarrow \text{Stack}$, 退出中断, 并打开全局中断	—	—	—	2
65	RETLW	I	$PC \leftarrow \text{Stack}$, 写 I 到 ACC	-	-	-	2
66	NOP		空操作	—	—	—	1

注:

- 1、在条件跳转指令的条件为真, 则 $S=1$, 否则 $S=0$ 。
- 2、在“B0MOV M, I”中, M 的范围是 $0x80 \sim 0x87$
- 3、执行 DAA 指令的前提是操作数的高 4 位和低 4 位都必须小于 A。即它们都是 BCD 码。否则, 结果会出错。

DAA 应用实例: $A=1001\ 1001=0x99$,
 $[0x20H]=1001\ 1001=0x99$;
 ADDAR $[0x20H], 0$; 结果= $0x132$, $C=1, DC=1$
 DAA ; 结果= $0x198$, 98 正是需要的 BCD 码值。

操作原理:

当加的结果的低 4bit 大于 9 或者有进位 $DC=1$, 则低 4 位+6, 高四位也是大于 9 或者 $C=1$, 则高 4 位+6。即根据结果判断是加上 06h, 60h 还是 66h。



14. 电气参数

14.1 极限参数

参数	符号	参数范围	单位
电源电压	V _{CC}	-0.3~5.5	V
输入电压	V _I	-0.3~V _{CC} +0.3	V
输出电压	V _O	-0.3~V _{CC} +0.3	V
最大功耗 (V _{CC} =5V,空载)	P _a	5	mW
工作温度	T _{opr}	-20~+80	°C
贮存温度	T _{stg}	-40~+125	°C

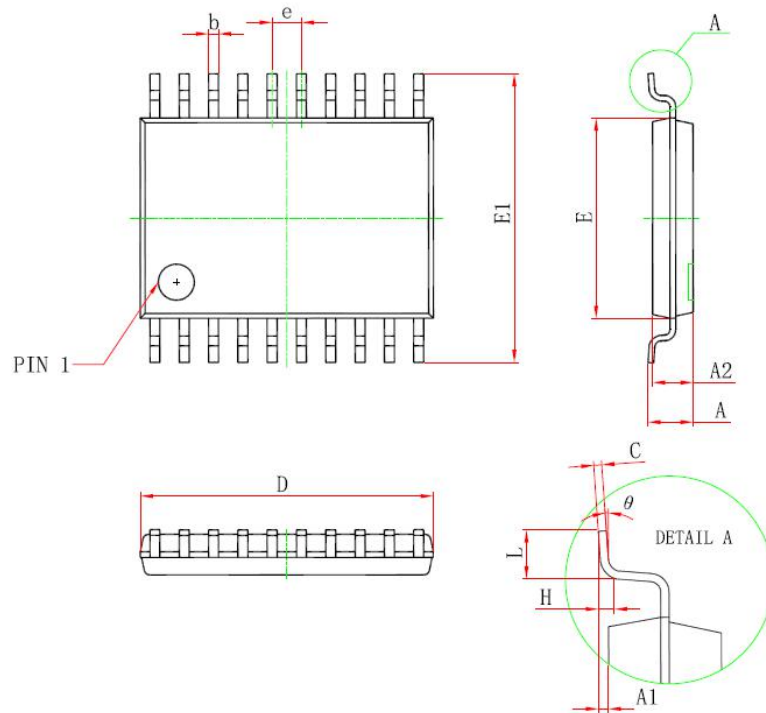
14.2 DC 特性 (TEMP=25°C、VDD=5V、全指令周期)

参数	符号	测试条件	最小值	典型值	最大值	单位
电源电压	V _{CC}	14MHz, Fosc/2	2.0		5.5	V
		28MHz, Fosc/2	2.8		5.5	V
静态电流	I _{DD}	振荡器停振,休眠	1	2.5	3	μA
工作电流	I _{CC}	14MHz/4 分频, 空载		1.5		mA
	I _{CC}	28MHz/4 分频, 空载		2.5		mA
	I _{CC1}	32KHz 低速,VDD=5V		14		uA
	I _{CC2}	32KHz 低速,VDD=3V		7.5		uA
	I _{CC3}	500KHz 低速,VDD=5V		80		uA
	I _{CC4}	500KHz 低速,VDD=3V		60		uA
内置高频 RC	F _{HRC}	温度: -20~80°C 电压: 2~5V	13.5	14	14.5	MHz
内置低频 RC	F _{LRC}	VDD=5v		32		KHz
		VDD=3V		16		
I/O 管脚输入高电压	V _{IH}		4.0			V
I/O 管脚输入低电压	V _{IL}				1.1	V
RESET	V _{IH}		4.0			V
	V _{IL}				1.1	V
I/O 管脚 (1 级驱动)	I _{OL}	V _{OL} =0.2VDD		12.6		mA
	I _{OH}	V _{OH} =0.8VDD		5.5		mA
I/O 管脚 (2 级驱动)	I _{OL}	V _{OL} =0.2VDD		44.5		mA
	I _{OH}	V _{OH} =0.8VDD		18.5		mA
I/O 管脚 (3 级驱动)	I _{OL}	V _{OL} =0.2VDD		64.5		mA
	I _{OH}	V _{OH} =0.8VDD		30		mA
端口上/下拉电阻	R	0.5VDD		60		KΩ
BOR disable	V	温度: -20~80°C	1.8	1.9	2.0	V
BOR 1.8V	V	温度: -20~80°C	1.8	1.9	2.0	V
BOR 2.2V	V	温度: -20~80°C	2.1	2.2	2.3	V



15. 封装及尺寸

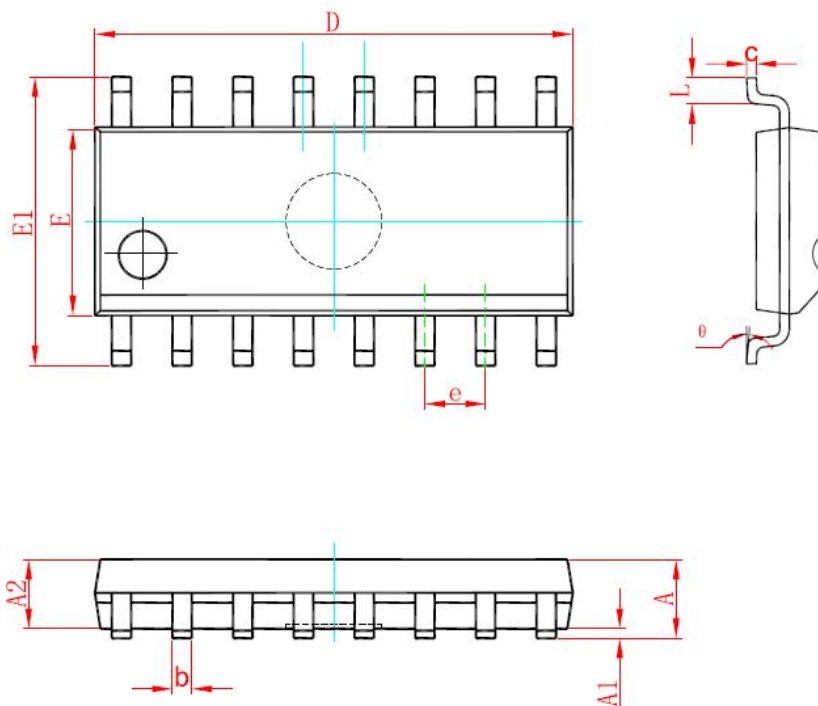
15.1 TSSOP20 封装图及尺寸



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
D	6.400	6.600	0.252	0.259
E	4.300	4.500	0.169	0.177
b	0.190	0.300	0.007	0.012
e	0.090	0.200	0.004	0.008
E1	6.250	6.550	0.246	0.258
A		1.200		0.047
A2	0.800	1.000	0.031	0.039
A1	0.050	0.150	0.002	0.006
c	0.65 (BSC)		0.026 (BSC)	
L	0.500	0.700	0.020	0.028
H	0.25(TYP)		0.01(TYP)	
θ	1°	7°	1°	7°



15.2 SOP16 封装图及尺寸



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	1.350	1.750	0.053	0.069
A1	0.100	0.250	0.004	0.010
A2	1.350	1.550	0.053	0.061
b	0.330	0.510	0.013	0.020
c	0.170	0.250	0.007	0.010
D	9.800	10.200	0.386	0.402
E	3.800	4.000	0.150	0.157
E1	5.800	6.200	0.228	0.244
e	1.270 (BSC)		0.050 (BSC)	
L	0.400	1.270	0.016	0.050
θ	0°	8°	0°	8°



16. 修正记录

版本	时间	内容
V1.0	2020.04.22	初版
V1.1	2020.06.16	添加 SOP16 管脚信息，修改部分说明
V2.0	2020.10.14	添加 TIRCF 寄存器信息
V2.1	2020.11.23	完善部分内容

Information furnished is believed to be accurate and reliable. However, HuaXin Micro-electronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. The actual part delivered may not completely agree with the description written here and it is user's responsibility to make wise judgment on the performance. HuaXin Micro-electronics assumes no responsibility for the mismatch occurred. No license is granted by implication or otherwise under any patent or patent rights of HuaXin Micro-electronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. HuaXin Micro-electronics products are not authorized for use as critical components in life support devices or systems without express written approval of HuaXin Micro-electronics.