



8-Bit 带 A/D 触摸按键式 Flash 单片机 BS45F16



版本：V06 日期：2013/10/15

www.greenmcu.com

目录

第 1 章 简述.....	1
1.1 特性.....	1
1.1.1 CPU 特性.....	1
1.1.2 周边特性.....	1
1.2 概述.....	2
1.3 方框图.....	3
1.3.1 引脚图.....	3
1.3.2 引脚说明.....	5
1.3.3 极限参数.....	8
1.3.4 直流电气特性.....	9
1.3.5 交流电气特性.....	10
1.3.6 ADC 特性.....	11
1.3.7 放大器电气特性.....	13
1.3.8 比较器电气特性.....	13
1.3.9 上电复位特性.....	13
第 2 章 系统.....	15
2.1 系统结构.....	15
2.1.1 时序和流水线结构.....	15
2.1.2 程序计数器.....	16
2.1.3 堆栈.....	16
2.1.4 算术逻辑单元 — ALU.....	17
2.2 FLASH 程序存储器.....	18
2.2.1 结构.....	18
2.2.2 特殊向量.....	18
2.2.3 Flash 存储器部分锁保护.....	18
2.2.4 查表.....	19
2.2.5 查表范例.....	19
2.2.6 表格读取程序范例.....	20
2.3 数据存储器.....	21
2.3.1 结构.....	21
2.4 特殊功能寄存器描述.....	22
2.4.1 间接寻址寄存器 — IAR0, IAR1.....	22
2.4.2 间接寻址指针 — MPO, MP1.....	22
2.4.3 间接寻址程序范例.....	23
2.4.4 存储区指针 — BP.....	24

2.4.5	存储区指针寄存器.....	24
2.4.6	累加器 — ACC.....	24
2.4.7	程序计数器低字节寄存器 — PCL.....	24
2.4.8	表格寄存器 — TBLP, TBHP, TBLH.....	24
2.4.9	状态寄存器 — STATUS.....	24
2.5	EEPROM 数据存储器.....	26
2.5.1	EEPROM 数据存储器结构.....	26
2.5.2	EEPROM 寄存器.....	26
2.5.3	EEPROM 寄存器列表.....	26
2.5.4	从 EEPROM 中读取数据.....	27
2.5.5	写数据到 EEPROM.....	27
2.5.6	写保护.....	27
2.5.7	EEPROM 中断.....	27
2.5.8	编程注意事项.....	28
2.5.9	程序举例.....	28
2.6	振荡器.....	29
2.6.1	振荡器概述.....	29
2.6.2	系统时钟配置.....	29
2.6.3	内部高速 RC 振荡器 — HIRC.....	29
2.6.4	内部低速 RC 振荡器 — LIRC.....	30
2.7	工作模式和系统时钟.....	31
2.7.1	系统时钟.....	31
2.7.2	控制寄存器.....	31
2.7.3	系统工作模式.....	32
2.7.4	工作模式切换.....	33
2.7.5	静态电流的注意事项.....	35
2.7.6	唤醒.....	35
2.7.7	编程注意事项.....	35
2.8	看门狗定时器.....	37
2.8.1	看门狗定时器时钟源.....	37
2.8.2	看门狗定时器控制寄存器.....	37
2.8.3	看门狗定时器操作.....	37
2.8.4	复位和初始化.....	38
2.9	输入/输出端口.....	43
2.9.1	输入/输出寄存器列表.....	43
2.9.2	上拉电阻.....	43
2.9.3	PA 口唤醒.....	44

2.9.4	输入/输出端口控制寄存器.....	44
2.9.5	输入/输出引脚结构.....	45
2.9.6	编程注意事项.....	45
2.10	定时器模块 - TM.....	47
2.10.1	简介.....	47
2.10.2	TM 操作.....	47
2.10.3	TM 时钟源.....	47
2.10.4	TM 中断.....	47
2.10.5	TM 外部引脚.....	47
2.10.6	TM 输入/输出引脚控制寄存器.....	48
2.10.7	编程注意事项.....	51
2.11	简易型 TM - CTM (TMO 和 TM2)	52
2.11.1	简易型 TM 操作.....	52
2.11.2	简易型 TM 寄存器介绍.....	52
2.11.3	简易型 TM 工作模式.....	55
2.11.4	PWM 输出模式.....	57
2.12	增强型 TM - ETM (TM1)	60
2.12.1	增强型 TM 操作.....	60
2.12.2	增强型 TM 寄存器介绍.....	61
2.12.3	增强型 TM 工作模式.....	65
2.13	运算放大器 OPA.....	78
2.13.1	运算放大器寄存器.....	78
2.14	比较器.....	80
2.14.1	比较器操作.....	80
2.14.2	比较器寄存器.....	80
2.14.3	DAC 寄存器.....	81
2.14.4	运算放大器&比较器&DAC 结构框图.....	82
2.15	A/D 转换器.....	83
2.15.1	A/D 简介.....	83
2.15.2	A/D 转换寄存器介绍.....	83
2.15.3	A/D 操作.....	86
2.15.4	A/D 输入引脚.....	86
2.15.5	A/D 转换步骤.....	87
2.15.6	编程注意事项.....	88
2.15.7	A/D 转换功能.....	88
2.15.8	A/D 转换应用范例.....	88
2.16	中断.....	91

2.16.1	中断寄存器.....	91
2.16.2	中断寄存器内容.....	91
2.16.3	中断操作.....	95
2.16.4	外部中断.....	96
2.16.5	A/D 转换器中断.....	96
2.16.6	触摸按键中断.....	97
2.16.7	多功能中断.....	97
2.16.8	时基中断.....	97
2.16.9	LVD 中断.....	98
2.16.10	TM 中断.....	98
2.16.11	EEPROM 中断.....	98
2.16.12	运算放大器中断.....	98
2.16.13	比较器中断.....	98
2.16.14	中断唤醒功能.....	99
2.16.15	编程注意事项.....	99
2.17	低电压检测-LVD.....	100
2.17.1	LVD 寄存器.....	100
2.17.2	LVDC 寄存器.....	100
2.17.3	LVD 操作.....	100
2.18	带 SCOM 和 SSEG 功能的 LCD 驱动器.....	102
2.18.1	LCD 操作.....	102
2.18.2	LCD 偏压控制.....	103
2.18.3	LCD 驱动寄存器.....	103
2.18.4	LED 驱动器.....	104
2.18.5	LED 驱动寄存器.....	105
2.19	应用电路.....	106
第3章	指令集.....	107
3.1	简介.....	107
3.2	指令周期.....	107
3.3	数据的传送.....	107
3.4	算术运算.....	107
3.5	逻辑和移位运算.....	107
3.6	分支和控制的转换.....	107
3.7	位运算.....	107
3.8	查表运算.....	108
3.9	其它运算.....	108
3.10	指令设定一览表.....	108

3.11	指令定义.....	110
3.11.1	ADC A, [m] Add Data Memory to ACC with Carry.....	110
3.11.2	ADCM A, [m] Add ACC to Data Memory with Carry.....	110
3.11.3	ADD A, [m] Add Data Memory to ACC.....	110
3.11.4	ADD A, x Add immediate data to ACC.....	110
3.11.5	ADDM A, [m] Add ACC to Data Memory.....	110
3.11.6	AND A, [m] Logical AND Data Memory to ACC.....	110
3.11.7	AND A, x Logical AND immediate data to ACC.....	110
3.11.8	ANDM A, [m] Logical AND ACC to Data Memory.....	110
3.11.9	CALL addr Subroutine call.....	110
3.11.10	CLR [m] Clear Data Memory.....	111
3.11.11	CLR [m].i Clear bit of Data Memory.....	111
3.11.12	CLR WDT Clear Watchdog Timer.....	111
3.11.13	CLR WDT1 Pre-clear Watchdog Timer.....	111
3.11.14	CLR WDT2 Pre-clear Watchdog Timer.....	111
3.11.15	CPL [m] Complement Data Memory.....	111
3.11.16	CPLA [m] Complement Data Memory with result in ACC.....	111
3.11.17	DAA [m] Decimal-Adjust ACC for addition with result in Data Memory	112
3.11.18	DEC [m] Decrement Data Memory.....	112
3.11.19	DECA [m] Decrement Data Memory with result in ACC.....	112
3.11.20	HALT Enter power down mode.....	112
3.11.21	INC [m] Increment Data Memory.....	112
3.11.22	INCA [m] Increment Data Memory with result in ACC.....	112
3.11.23	JMP addr Jump unconditionally.....	112
3.11.24	MOV A, [m] Move Data Memory to ACC.....	113
3.11.25	MOV A, x Move immediate data to ACC.....	113
3.11.26	MOV [m], A Move ACC to Data Memory.....	113
3.11.27	NOP No operation.....	113
3.11.28	OR A, [m] Logical OR Data Memory to ACC.....	113
3.11.29	OR A, x Logical OR immediate data to ACC.....	113
3.11.30	ORM A, [m] Logical OR ACC to Data Memory.....	113
3.11.31	RET Return from subroutine.....	113
3.11.32	RET A, x Return from subroutine and load immediate data to ACC.....	113
3.11.33	RETI Return from interrupt.....	113
3.11.34	RL [m] Rotate Data Memory left.....	114
3.11.35	RLA [m] Rotate Data Memory left with result in ACC.....	114

3. 11. 36	RLC [m]	Rotate Data Memory Left through Carry	114
3. 11. 37	RLCA [m]	Rotate Data Memory left through Carry with result in ACC.....	114
3. 11. 38	RR [m]	Rotate Data Memory right	114
3. 11. 39	RRA [m]	Rotate Data Memory right with result in ACC	114
3. 11. 40	RRC [m]	Rotate Data Memory right through Carry	115
3. 11. 41	RRCA [m]	Rotate Data Memory right through Carry with result in ACC...	115
3. 11. 42	SBC A, [m]	Subtract Data Memory from ACC with Carry	115
3. 11. 43	SBCM A, [m]	Subtract Data Memory from ACC with Carry and result in Data Memory	115
3. 11. 44	SDZ [m]	Skip if Decrement Data Memory is 0	115
3. 11. 45	SDZA [m]	Skip if decrement Data Memory is zero with result in ACC.....	115
3. 11. 46	SET [m]	Set Data Memory	116
3. 11. 47	SET [m]. i	Set bit of Data Memory	116
3. 11. 48	SIZ [m]	Skip if increment Data Memory is 0	116
3. 11. 49	SIZA [m]	Skip if increment Data Memory is zero with result in ACC...	116
3. 11. 50	SNZ [m]. i	Skip if bit i of Data Memory is not 0	116
3. 11. 51	SUB A, [m]	Subtract Data Memory from ACC	116
3. 11. 52	SUBM A, [m]	Subtract Data Memory from ACC with result in Data Memory...	116
3. 11. 53	SUB A, x	Subtract immediate Data from ACC	117
3. 11. 54	SWAP [m]	Swap nibbles of Data Memory	117
3. 11. 55	SWAPA [m]	Swap nibbles of Data Memory with result in ACC	117
3. 11. 56	SZ [m]	Skip if Data Memory is 0	117
3. 11. 57	SZA [m]	Skip if Data Memory is 0 with data movement to ACC.....	117
3. 11. 58	SZ [m]. i	Skip if bit i of Data Memory is 0	117
3. 11. 59	TABRDC [m]	Read table (current page) to TBLH and Data Memory.....	117
3. 11. 60	TABRDL [m]	Read table (last page) to TBLH and Data Memory.....	117
3. 11. 61	XOR A, [m]	Logical XOR Data Memory to ACC	118
3. 11. 62	XORM A, [m]	Logical XOR ACC to Data Memory	118
3. 11. 63	XOR A, x	Logical XOR immediate data to ACC	118
第 4 章 封装			119
4. 1	20-PIN DIP (300MIL)外形尺寸.....		119
4. 1. 1	MS-001d (见 fig1)		119
4. 1. 2	MO-095a (见 fig2)		120
4. 2	20-pin SOP (300mil)外形尺寸		121
4. 2. 1	MS-013.....		121
4. 3	24-PIN SKDIP(300MIL)外形尺寸		121
4. 3. 1	MS-001d (见 Fig1)		122

4.3.2 MS-001d (见 Fig2)	122
4.3.3 MO-095a (见 Fig2)	123
4.4 24-PIN SOP(300MIL)外形尺寸	124
4.5 28-PIN SKDIP(300MIL)外形尺寸	125
4.6 28-PIN SOP(300MIL)外形尺寸	126
4.6.1 MS-013.....	126
4.7 包装带和卷轴规格.....	127
4.7.1 卷轴尺寸.....	127
4.8 运输带尺寸	128
4.8.1 SOP 20W.....	128
4.8.2 SOP 24W.....	129
4.8.3 SOP 28W(300mil)	129
第5章 订购信息	130
5.1 订购信息	130
第6章 产品命名规则	131
6.1 产品命名规则	131

第 1 章 简述

1.1 特性

1.1.1 CPU 特性

- 工作电压：
f_{SYS} = 8MHz: 2.2V~5.5V
f_{SYS} = 12MHz: 2.7V~5.5V
- 集成 16 个触摸按键功能 — 不需要增加外接元件
- 暂停和唤醒功能，以降低功耗
- 集成高/低速振荡器
低速 — 32kHz
高速 — 8MHz, 12MHz
- 多种工作模式：正常模式，低速模式，空闲模式和休眠模式
- 所有指令都可在 1 或 2 个指令周期内完成
- 查表指令
- 63 条功能强大的指令系统
- 多达 8 层硬件堆栈
- 位操作指令

1.1.2 周边特性

- Flash 程序存储器：4K×16
- 数据存储器：384×8
- EEPROM 存储器：64×8
- 看门狗定时器功能
- 多达 26 个双向 I/O 口
- 三个定时器模块
- 双时基功能，用于产生固定时间中断信号
- 软件控制的 1/3 bias 4×16 LCD 驱动器
- 两个与 I/O 口复用的外部中断输入
- 内置比较器和运算放大器
- 6 通道 12-bit 的 A/D 转换器
- 低电压复位功能
- 低电压检测功能
- 提供多种封装形式

1.2 概述

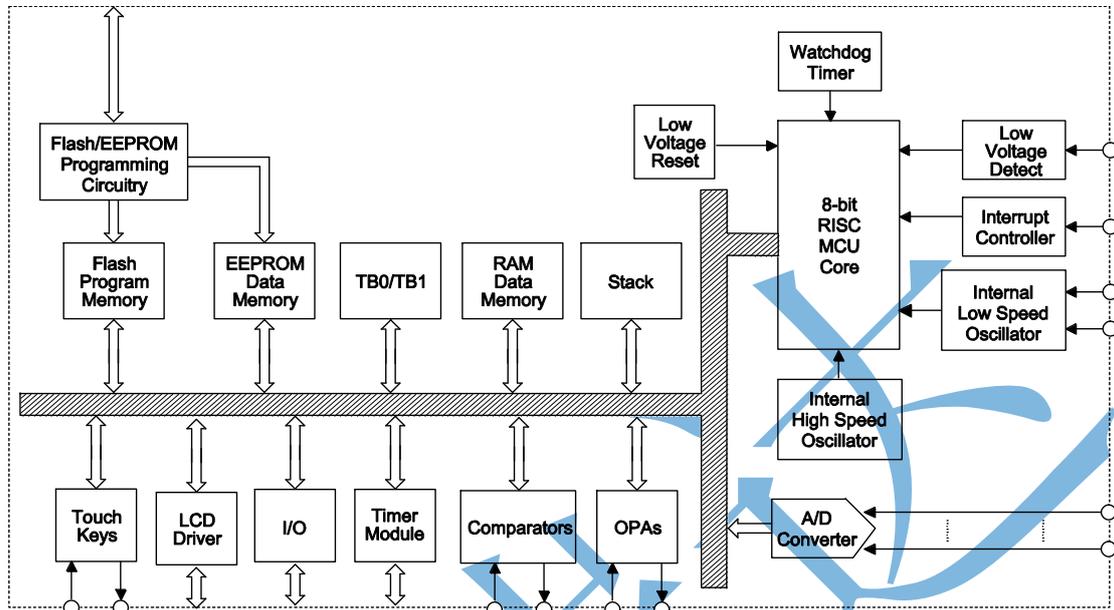
该单片机是一款 A/D 型具有 8 位高性能精简指令集且完全集成触摸按键功能的 Flash 单片机。此单片机具有一系列功能和特性，其 Flash 存储器可多次编程的特性给用户提供了极大的方便。存储器方面，还包含了一个 RAM 数据存储器和一个可用于存储序号、校准数据等非易失性数据的 EEPROM 存储器。

在模拟特性方面，该单片机包含一个多通道 12-bit A/D 转换器、16 通道触摸按键输入、比较器和运算放大器。还带有多个使用灵活的定时器模块，可提供定时功能、脉冲产生功能及 PWM 产生功能。内部看门狗定时器、低电压复位和低电压检测等内部保护特性，外加优秀的抗干扰和 ESD 保护性能，确保单片机在恶劣的电磁干扰环境下可靠地运行。

这款单片机提供了 HIRC 和 LIRC 振荡器功能选项，且内建完整的系统振荡器，无需外围元器件。其在不同工作模式之间动态切换的能力，为用户提供了一个优化单片机操作和减少功耗的手段。

外加时基功能、I/O 使用灵活等其它特性，使这款单片机可以广泛应用于各种产品中，例如电子测量仪器、环境监控、手持式测量工具、家庭应用、电子控制工具、马达控制等方面。

1. 3 方框图



1. 3. 1 引脚图

PC0/TCK0/INT0/SSEG0/KEY1	1	20	PB0/TP0_0/SCOM0/KEY13
PC1/TCK1/SSEG1/KEY2	2	19	PB1/TP1A/SCOM1/KEY14
PC2/TCK2/SSEG2/KEY3	3	18	PB2/TP2_0/SCOM2/KEY15
PC3/SSEG3/KEY4	4	17	PB3/TP1B_0/SCOM3/KEY16
PD0/KEY9/AN0	5	16	PA1/INT1/CX/SSEG12
PD1/KEY10/AN1	6	15	PA4/SSEG13/CN
PD2/KEY11/AN2	7	14	PA3/SSEG14/CP
PD3/KEY12/AN3	8	13	PA0/SSEG15/AX
VSS	9	12	PA2/AN/AN5/VREF
VDD	10	11	PA7/AP/AN4

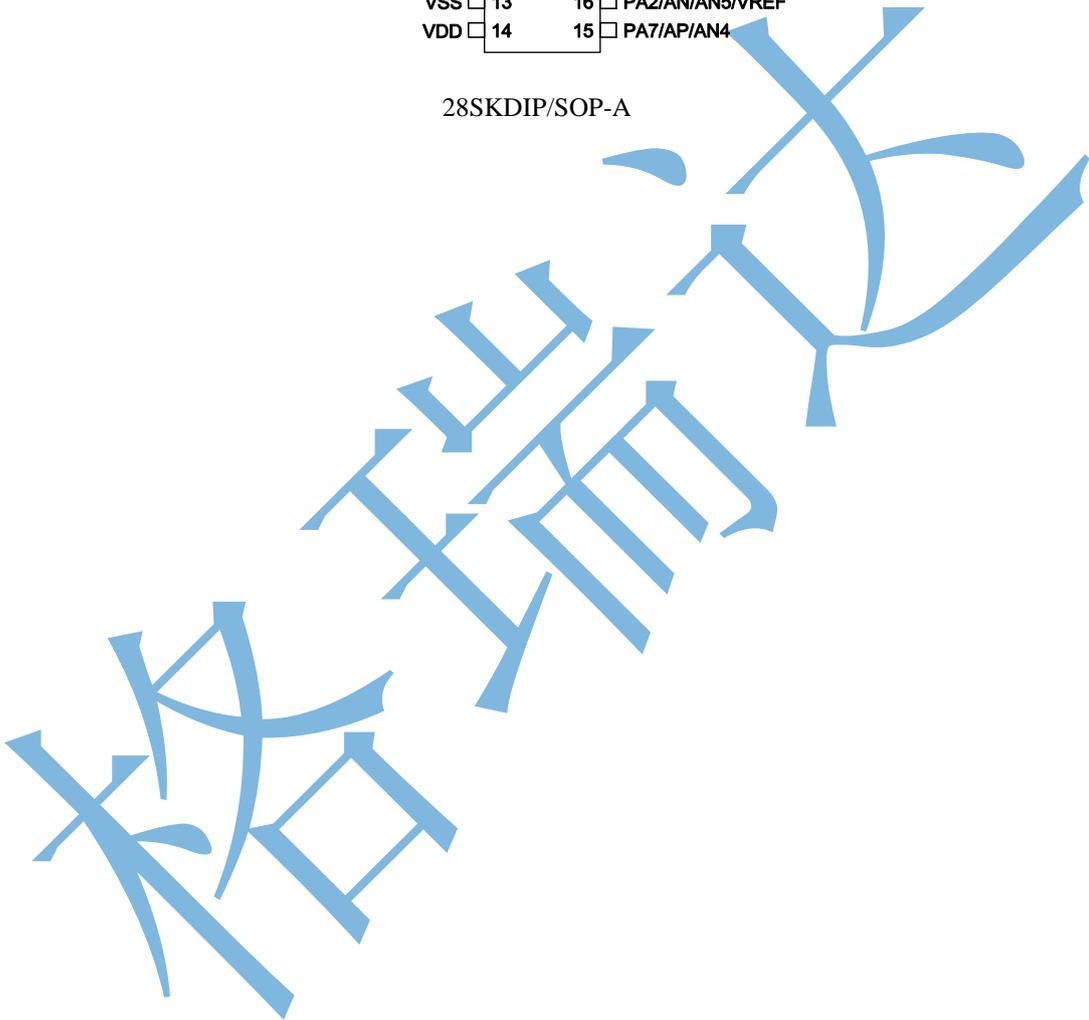
20DIP/SOP-A

PC1/TCK1/SSEG1/KEY2	1	24	PC0/TCK0/INT0/SSEG0/KEY1
PC2/TCK2/SSEG2/KEY3	2	23	PB0/TP0_0/SCOM0/KEY13
PC3/SSEG3/KEY4	3	22	PB1/TP1A/SCOM1/KEY14
PC4/SSEG4/KEY5	4	21	PB2/TP2_0/SCOM2/KEY15
PC5/SSEG5/KEY6	5	20	PB3/TP1B_0/SCOM3/KEY16
PC6/SSEG6/KEY7	6	19	PA1/INT1/CX/SSEG12
PC7/SSEG7/KEY8	7	18	PA4/SSEG13/CN
PD0/KEY9/AN0	8	17	PA3/SSEG14/CP
PD1/KEY10/AN1	9	16	PA0/SSEG15/AX
PD2/KEY11/AN2	10	15	PA2/AN/AN5/VREF
PD3/KEY12/AN3	11	14	PA7/AP/AN4
VSS	12	13	VDD

24SKDIP/SOP-A

PC0/TCK0/INT0/SSEG0/KEY1	1	28	PB0/TP0_0/SCOM0/KEY13
PC1/TCK1/SSEG1/KEY2	2	27	PB1/TP1A/SCOM1/KEY14
PC2/TCK2/SSEG2/KEY3	3	26	PB2/TP2_0/SCOM2/KEY15
PC3/SSEG3/KEY4	4	25	PB3/TP1B_0/SCOM3/KEY16
PC4/SSEG4/KEY5	5	24	PB4/TP0_1/SSEG8
PC5/SSEG5/KEY6	6	23	PB5/TP1B_1/SSEG9
PC6/SSEG6/KEY7	7	22	PB6/TP2_1/SSEG10
PC7/SSEG7/KEY8	8	21	PB7/TP1B_2/SSEG11
PD0/KEY9/AN0	9	20	PA1/INT1/CX/SSEG12
PD1/KEY10/AN1	10	19	PA4/SSEG13/CN
PD2/KEY11/AN2	11	18	PA3/SSEG14/CP
PD3/KEY12/AN3	12	17	PA0/SSEG15/AX
VSS	13	16	PA2/AN/AN5/VREF
VDD	14	15	PA7/AP/AN4

28SKDIP/SOP-A



1.3.2 引脚说明

下表中列出了每个引脚的功能，而每个引脚功能的细节将在文中其它章节有详细的描述。

引脚名称	功能	寄存器选择	I/T	O/T	说明
PA0/SSEG15/AX	PA0	PAWU PAPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻和唤醒功能
	SSEG15	SLCDCn	—	LCD	软件控制的 LCD SEG
	AX	OPAC0	—	OPAO	OPA 输出脚
PA1/INT1/ CX/SSEG12	PA1	PAWU PAPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻和唤醒功能
	INT1	INTC0	ST	—	外部中断 1 输入
	CX	CPC0	—	CMPO	比较器输出脚
	SSEG12	SLCDCn	—	LCD	软件控制的 LCD SEG
PA2/AN/AN5/VREF	PA2	PAWU PAPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻和唤醒功能
	AN	OPAC1	OPAI	—	OPA 反相输入
	AN5	ADCR0	AN	—	A/D 通道 5
	VREF	ADCR1	AN	—	A/D 转换器参考电压输入
PA3/SSEG14/CP	PA3	PAWU PAPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻和唤醒功能
	SSEG14	SLCDCn	—	LCD	软件控制的 LCD SEG
	CP	CPC1	OPAI	—	比较器同相输入脚
PA4/ SSEG13/CN	PA4	PAWU PAPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻和唤醒功能
	SSEG13	SLCDCn	—	LCD	软件控制的 LCD SEG
	CN	CPC1	CMPI	—	比较器反相输入脚
PA7/AP/AN4	PA7	PAWU PAPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻和唤醒功能
	AP	OPAC1	OPAI	—	OPA 同相输入
	AN4	ADCR0	AN	—	A/D 通道 4
PB0/TP0_0/ SCOM0/KEY13	PB0	PBPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻
	TP0_0	TMPCn	ST	CMOS	TM0 I/O 口
	SCOM0	SLCDCn	—	LCD	软件控制的 LCD COM
	KEY13	TKMnC1	NS	—	触摸按键输入口
PB1/TP1A/ SCOM1/KEY14	PB1	PBPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻
	TP1A	TMPCn	ST	CMOS	TM1 I/O 口
	SCOM1	SLCDCn	—	LCD	软件控制的 LCD COM
	KEY14	TKMnC1	NS	—	触摸按键输入口
PB2/TP2_0/ SCOM2/KEY15	PB2	PBPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻
	TP2_0	TMPCn	ST	CMOS	TM2 I/O 口
	SCOM2	SLCDCn	—	LCD	软件控制的 LCD COM
	KEY15	TKMnC1	NS	—	触摸按键输入口
PB3/TP1B_0/ SCOM3/KEY16	PB3	PBPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻
	TP1B_0	TMPCn	ST	CMOS	TM1 I/O 口

引脚名称	功能	寄存器选择	I/T	O/T	说明
	SCOM3	SLCDCn	—	LCD	软件控制的 LCD COM
	KEY16	TKMnC1	NS	—	触摸按键输入口
PB4/TP0_1/ SSEG8	PB4	PBPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻
	TP0_1	TMPCn	ST	CMOS	TM0 I/O 口
	SSEG8	SLCDCn	—	LCD	软件控制的 LCD SEG
PB5/TP1B_1/ SSEG9	PB5	PBPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻
	TP1B_1	TMPCn	ST	CMOS	TM1 I/O 口
	SSEG9	SLCDCn	—	LCD	软件控制的 LCD SEG
PB6/TP2_1/SSEG10	PB6	PBPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻
	TP2_1	TMPCn	ST	CMOS	TM2 I/O 口
	SSEG10	SLCDCn	—	LCD	软件控制的 LCD SEG
PB7/TP1B_2/SSEG11	PB7	PBPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻
	TP1B_2	TMPCn	ST	CMOS	TM1 I/O 口
	SSEG11	SLCDCn	—	LCD	软件控制的 LCD SEG
PC0/TCK0/INT0/ SSEG0/KEY1	PC0	PCPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻
	TCK0	TM0C0	ST	—	定时器模块 0 输入口
	INT0	INTC0	ST	—	外部中断 0 输入口
	SSEG0	SLCDCn	—	LCD	软件控制的 LCD SEG
	KEY1	TKMnC1	NS	—	触摸按键输入口
PC1/TCK1/ SSEG1/KEY2	PC1	PCPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻
	TCK1	TM1C0	ST	—	定时器模块 1 输入口
	SSEG1	SLCDCn	—	LCD	软件控制的 LCD SEG
	KEY2	TKMnC1	NS	—	触摸按键输入口
PC2/TCK2/ SSEG2/KEY3	PC2	PCPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻
	TCK2	TM2C0	ST	—	定时器模块 2 输入口
	SSEG2	SLCDCn	—	LCD	软件控制的 LCD SEG
	KEY3	TKMnC1	NS	—	触摸按键输入口
PC3/SSEG3/KEY4	PC3	PCPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻
	SSEG3	SLCDCn	—	LCD	软件控制的 LCD SEG
	KEY4	TKMnC1	NS	—	触摸按键输入口
PC4/SSEG4/KEY5	PC4	PCPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻
	SSEG4	SLCDCn	—	LCD	软件控制的 LCD SEG
	KEY5	TKMnC1	NS	—	触摸按键输入口
PC5/SSEG5/KEY6	PC5	PCPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻

引脚名称	功能	寄存器选择	I/T	O/T	说明
	SSEG5	SLCDCn	—	LCD	软件控制的 LCD SEG
	KEY6	TKMnC1	NS	—	触摸按键输入口
PC6/SSEG6/KEY7	PC6	PCPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻
	SSEG6	SLCDCn	—	LCD	软件控制的 LCD SEG
	KEY7	TKMnC1	NS	—	触摸按键输入口
PC7/SSEG7/KEY8	PC7	PCPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻
	SSEG7	SLCDCn	—	LCD	软件控制的 LCD SEG
	KEY8	TKMnC1	NS	—	触摸按键输入口
PD0/KEY9/AN0	PD0	PDPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻
	KEY9	TKMnC1	NS	—	触摸按键输入口
	AN0	ADCR0	AN	—	A/D 通道 0
PD1/KEY10/AN1	PD1	PDPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻
	KEY10	TKMnC1	NS	—	触摸按键输入口
	AN1	ADCR0	AN	—	A/D 通道 1
PD2/KEY11/AN2	PD2	PDPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻
	KEY11	TKMnC1	NS	—	触摸按键输入口
	AN2	ADCR0	AN	—	A/D 通道 2
PD3/KEY12/AN3	PD3	PDPU	ST	CMOS	通用 I/O 口, 可通过寄存器设置上拉电阻
	KEY12	TKMnC1	NS	—	触摸按键输入口
	AN3	ADCR0	AN	—	A/D 通道 3
VDD	VDD	—	PWR	—	电源电压
VSS	VSS	—	PWR	—	地

注: I/T: 输入类型
 O/T: 输出类型
 寄存器选择: 通过设置寄存器来选择功能
 PWR: 电源
 ST: 斯密特触发输入
 CMOS: CMOS 输出
 NMOS: NMOS 输出
 LCD: LCD COM 或 SEG Vbias 输出
 NS: 无标准输入或输出
 此引脚功能表是针对大封装芯片而言的, 对于小封装的芯片可能不具有上述引脚和功能。

1.3.3 极限参数

电源供应电压 $V_{SS}-0.3V$ 至 $V_{SS}+6.0V$
端口输入电压 $V_{SS}-0.3V$ 至 $V_{DD}+0.3V$
 I_{OL} 总电流 100mA
总功耗 500mW

储存温度 $-50^{\circ}C$ 至 $125^{\circ}C$
工作温度..... $-40^{\circ}C$ 至 $85^{\circ}C$
 I_{OH} 总电流 -100mA

注：这里只强调额定功率，超过极限参数所规定的范围将对芯片造成损害，无法预期芯片在上述标示范围外的工作状态，而且若长期在标示范围外的条件下工作，可能影响芯片的可靠性。



1.3.4 直流电气特性

 $T_a=25^{\circ}\text{C}$

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
V_{DD}	工作电压(HIRC)	—	$f_{SYS} = 8\text{MHz}$	2.2	—	5.5	V
			$f_{SYS} = 12\text{MHz}$	2.7	—	5.5	V
I_{DD1}	工作电流 (HIRC) ($f_{SYS}=f_H$)	3V	无负载, $f_H = 8\text{MHz}$,	—	1.2	1.8	mA
		5V	ADC 关闭,WDT 使能	—	2.7	4.1	mA
		3V	无负载, $f_H=12\text{MHz}$	—	1.9	2.9	mA
		5V	ADC 关闭,WDT 使能	—	4.2	6.3	mA
I_{DD2}	工作电流 (LIRC) ($f_{SYS}=f_L$)	3V	无负载, $f_L = 32\text{kHz}$	—	10	20	μA
		5V	ADC 关闭,WDT 使能	—	20	40	μA
I_{DD3}	工作电流,正常模式, $f_H=8\text{MHz}$	3V	无负载, $f_{SYS}=f_H/2$,	—	0.9	1.5	mA
		5V	ADC 关闭,WDT 使能	—	2.5	3.75	mA
		3V	无负载, $f_{SYS}=f_H/4$,	—	0.7	1.0	mA
		5V	ADC 关闭,WDT 使能	—	2.0	3.0	mA
		3V	无负载, $f_{SYS}=f_H/8$,	—	0.6	0.9	mA
		5V	ADC 关闭,WDT 使能	—	1.6	2.4	mA
		3V	无负载, $f_{SYS}=f_H/16$,	—	0.5	0.75	mA
		5V	ADC 关闭,WDT 使能	—	1.5	2.25	mA
		3V	无负载, $f_{SYS}=f_H/32$,	—	0.49	0.74	mA
		5V	ADC 关闭,WDT 使能	—	1.45	2.18	mA
		3V	无负载, $f_{SYS}=f_H/64$,	—	0.47	0.71	mA
		5V	ADC 关闭,WDT 使能	—	1.4	2.1	mA
I_{DD4}	工作电流,正常模式, $f_H=12\text{MHz}$	3V	无负载, $f_{SYS}=f_H/2$,	—	0.9	1.5	mA
		5V	ADC 关闭,WDT 使能	—	2.5	3.75	mA
		3V	无负载, $f_{SYS}=f_H/4$,	—	0.7	1.0	mA
		5V	ADC 关闭,WDT 使能	—	2.0	3.0	mA
		3V	无负载, $f_{SYS}=f_H/8$,	—	0.6	0.9	mA
		5V	ADC 关闭,WDT 使能	—	1.6	2.4	mA
		3V	无负载, $f_{SYS}=f_H/16$,	—	0.5	0.75	mA
		5V	ADC 关闭,WDT 使能	—	1.5	2.25	mA
		3V	无负载, $f_{SYS}=f_H/32$,	—	0.49	0.74	mA
		5V	ADC 关闭,WDT 使能	—	1.45	2.18	mA
		3V	无负载, $f_{SYS}=f_H/64$,	—	0.47	0.71	mA
		5V	ADC 关闭,WDT 使能	—	1.4	2.1	mA
I_{IDLE0}	IDLE0 模式静态电流 (LIRC)	3V	无负载,ADC 关闭,	—	1.5	3.0	μA
		5V	WDT 使能	—	3.0	6.0	μA
I_{IDLE1}	IDLE1 模式静态电流 (HIRC)	3V	无负载,ADC 关闭,	—	0.5	3.0	mA
		5V	WDT 使能, $f_{SYS}=8\text{MHz}$	—	1.0	6.0	mA
I_{IDLE2}	IDLE1 模式静态电流 (HIRC)	3V	无负载,ADC 关闭,	—	0.6	3.0	mA
		5V	WDT 使能, $f_{SYS}=12\text{MHz}$	—	1.2	6.0	mA
I_{SLEEP0}	SLEEP0 模式静态电流 (LIRC 关闭)	3V	无负载,ADC 关闭,	—	0.2	1.0	μA
		5V	WDT 除能	—	0.4	2.0	μA
I_{SLEEP1}	SLEEP1 模式静态电流 (LIRC 开启)	3V	无负载,ADC 关闭,	—	1.5	3.0	μA
		5V	WDT 使能	—	2.5	5.0	μA
V_{IL1}	输入/输出口或输入引脚 低电平输入电压	—	—	0	—	$0.3V_{DD}$	V

符号	参数	测试条件		最小	典型	最大	单位	
		V _{DD}	条件					
V _{IHI}	输入/输出或输入引脚高电平输入电压	—	—	0.7V _{DD}	—	V _{DD}	V	
I _{OL1}	输入/输出灌电流 (除了 PB)	3V	V _{OL} =0.1V _{DD}	4	8	—	mA	
		5V		10	20	—	mA	
I _{OL2}	输入/输出灌电流 (PB,加大2倍电流驱动)	3V	V _{OL} =0.1V _{DD}	8	16	—	mA	
		5V		20	40	—	mA	
I _{OH1}	输入/输出源电流	3V	V _{OH} = 0.9V _{DD}	-2	-4	—	mA	
		5V		-5	-10	—	mA	
V _{LVR}	低电压复位电压	—	LVR 使能, V _{LVR} =2.1V	-5%	2.1	+5%	V	
				LVR 使能, V _{LVR} =2.55V	-5%	2.55	-5%	V
				LVR 使能, V _{LVR} =3.15V	-5%	3.15	-5%	V
				LVR 使能, V _{LVR} =3.8V	-5%	3.8	-5%	V
V _{LVD}	低电压检测电压	—	LVDEN=1, V _{LVD} =2.0V	-5%	2.0	+5%	V	
			LVDEN=1, V _{LVD} =2.2V	-5%	2.2	-5%	V	
			LVDEN=1, V _{LVD} =2.4V	-5%	2.4	+5%	V	
			LVDEN=1, V _{LVD} =2.7V	-5%	2.7	+5%	V	
			LVDEN=1, V _{LVD} =3.0V	-5%	3.0	+5%	V	
			LVDEN=1, V _{LVD} =3.3V	-5%	3.3	+5%	V	
			LVDEN=1, V _{LVD} =3.6V	-5%	3.6	-5%	V	
LVDEN=1, V _{LVD} =4.0V	-5%	4.0	-5%	V				
V _{BG}	IC 内部参考电压	—	—	-3%	1.19	+3%	V	
I _{BG}	使用 V _{BG} 的额外功耗	—	—	—	200	300	μA	
I _{LVR}	使用 LVR 的额外功耗	3V	LVR 除能→LVR 使能	—	10	20	μA	
		5V		—	15	30	μA	
I _{LVD}	使用 LVD 的额外功耗	3V	LVD 除能→LVD 使能 (LVR 除能)	—	10	20	μA	
		5V		—	15	30	μA	
		3V	LVD 除能→LVD 使能 (LVR 使能)	—	1	2	μA	
		5V		—	2	4	μA	
I _{SCOM}	SCOM 工作电流	5V	ISEL[1:0]=00	17.5	25.0	32.5	μA	
			ISEL[1:0]=01	35	50	65	μA	
			ISEL[1:0]=10	70	100	130	μA	
			ISEL[1:0]=11	140	200	260	μA	
I _{SSEG}	SSEG 工作电流	5V	ISEL[1:0]=00	17.5	25.0	32.5	μA	
			ISEL[1:0]=01	35	50	65	μA	
			ISEL[1:0]=10	70	100	130	μA	
			ISEL[1:0]=11	140	200	260	μA	
V _{SCOM}	用于 LCD SCOM 的电压	5V	1/3 V _{DD}	-3%	0.33	+3%	V _{DD}	
			2/3 V _{DD}	-3%	0.67	+3%	V _{DD}	
V _{SSEG}	用于 LCD SSEG 的电压	5V	1/3 V _{DD}	-3%	0.33	+3%	V _{DD}	
			2/3 V _{DD}	-3%	0.67	+3%	V _{DD}	
R _{PH}	输入/输出上拉电阻	3V	—	20	60	100	kΩ	
		5V		10	30	50	kΩ	

1.3.5 交流电气特性

Ta=25°C

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
f _{CPU}	系统时钟	—	2.2V ~ 5.5V	DC	—	8	MHz

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
			2.7V ~ 5.5V	DC	—	12	MHz
f _{HIRC}	系统时钟(HIRC)	3V/5V	Ta=25°C	-2%	8	+2%	MHz
		3V/5V	Ta=25°C	-2%	12	+2%	MHz
		3V/5V	Ta=0~70°C	-4%	8	+3%	MHz
		3V/5V	Ta=0~70°C	-4%	12	+3%	MHz
		3V/5V	Ta=-40°C~85°C	-7%	8	+7%	MHz
		3V/5V	Ta=-40°C~85°C	-7%	12	+7%	MHz
		2.5V~4.0V	Ta=0~70°C	-9%	8	+6%	MHz
		3.0V~5.5V	Ta=0~70°C	-5%	8	+12%	MHz
		2.7V~4.0V	Ta=0~70°C	-9%	12	+5%	MHz
		3.0V~5.5V	Ta=0~70°C	-5%	12	+11%	MHz
		2.5V~4.0V	Ta=-40°C~85°C	-12%	8	+6%	MHz
		3.0V~5.5V	Ta=-40°C~85°C	-8%	8	+12%	MHz
		2.7V~4.0V	Ta=-40°C~85°C	-13%	12	+5%	MHz
		3.0V~5.5V	Ta=-40°C~85°C	-8%	12	+11%	MHz
		2.2V~5.5V	Ta=-40°C~85°C	-15%	8	+15%	MHz
		2.7V~5.5V	Ta=-40°C~85°C	-15%	12	+15%	MHz
f _{LIRC}	系统时钟(LIRC)	5V	Ta=25°C	-10%	32	+10%	kHz
		2.2V~5.5V	Ta=-40°C~85°C	-50%	32	+60%	kHz
t _{TIMER}	定时器输入最小脉宽	—	—	0.3	—	—	μs
t _{INT}	中断脉冲宽度	—	—	10	—	—	μs
t _{LVR}	低电压复位时间	—	—	120	240	480	μs
t _{LVD}	低电压中断时间	—	—	20	45	90	μs
t _{SRESET}	软件复位时间	—	—	45	90	120	μs
t _{LVDS}	LVDO 稳定的时间	—	—	15	—	—	μs
t _{EERD}	EEPROM 读时间	—	—	1	2	4	t _{sys}
t _{EEWR}	EEPROM 写时间	—	—	—	2	4	ms
t _{SST}	系统启动延时周期 (从 HALT 模式下唤醒, HALT 时 f _{sys} 关闭,低速 模式与正常模式之间进 行切换时)	—	f _{sys} =HIRC	16	—	—	t _{sys}
		—	f _{sys} =LIRC	2	—	—	
	系统启动延时周期 (从 HALT 模式下唤醒, HALT 时 f _{sys} 开启)	—	—	2	—	—	t _{sys}
t _{RSTD}	系统复位延时时间 (上电复位)	—	—	25	50	100	ms
	系统复位延时时间(上电 复位以外的其它复位)	—	—	8.3	16.7	33.3	ms

注: 1. t_{sys} = 1/f_{sys}

2. 为了保持内部 HIRC 振荡器频率的准确性, 需要在 VDD 和 VSS 之间接入一个 0.1μF 的去耦电容, 并且尽可能地靠近单片机。

1.3.6 ADC 特性

符号	参数	测试条件		最小值	典型值	最大值	单位
		V _{DD}	条件				
AV _{DD}	ADC 工作电压	—	—	2.2	—	5.5	V

V _{ADI}	ADC 输入电压	—	—	0	—	V _{REF}	V
V _{REF}	ADC 参考电压	—	—	2	—	AV _{DD}	V
V _{BG}	IC 内部参考电压	—	—	-3%	1.19	+3%	V
DNL	A/D 非线性微分误差	5V	t _{ADCK} =1.0μs	—	±1	±2	LSB
INL	A/D 非线性积分误差	5V	t _{ADCK} =1.0μs	—	±2	±4	LSB
I _{ADC}	打开 A/D 增加的功耗	3V	无负载, t _{ADCK} =0.5μs	—	0.90	1.35	mA
		5V	无负载, t _{ADCK} =0.5μs	—	1.20	1.80	mA
t _{ADCK}	A/D 时钟周期	—	—	0.5	—	100	μs
t _{ADC}	A/D 转换时间(包括 A/D 采样和保持时间)	—	12-bit ADC	—	16	—	t _{ADCK}
t _{ADS}	A/D 采样时间	—	—	—	4	—	t _{ADCK}
t _{ON2ST}	A/D On-to-Start 时间	—	—	2	—	—	μs
t _{BGS}	V _{BG} 稳定时间	—	—	200	—	—	μs

1.3.7 放大器电气特性

 $T_a=25^{\circ}\text{C}$

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
DC 电气特性							
V_{DDO}	工作电压	—	—	2.2	—	5.5	V
I_{DDO}	静态电流	5V	无负载	—	200	350	μA
V_{OPOS1}	输入失调电压	5V	$A_{xOF4\sim 0}=(10000)$	-15	—	15	mV
V_{OPOS2}	输入失调电压	5V	校准后	-4	—	4	mV
I_{OPOS}	输入失调电流	—	$V_{DD}=5\text{V}, V_{CM}=1/2V_{DD}$	—	TBD	—	—
V_{CM}	共模电压范围	—	—	V_{SS}	—	$V_{DD}-1.4\text{V}$	V
PSRR	电源电压抑制比	—	—	58	80	—	dB
CMRR	共模抑制比	—	$V_{DD}=5\text{V},$ $V_{CM}=V_{DD}-1.4\text{V}$	58	80	—	dB
PGAG	PGA 增益	5V	—	-3%	—	+3%	—
R_{10K}	PGA 10K 电阻	—	—	-20%	10K	+20%	Ω
AC 电气特性							
A_{OL}	开环增益	—	—	60	80	—	dB
SR	摆率+, 摆率-	—	无负载	—	0.1	—	$\text{V}/\mu\text{s}$
GBW	增益带宽	—	$R_L=1\text{M}\Omega, C_L=100\text{pF}$	100K	2M	—	Hz

1.3.8 比较器电气特性

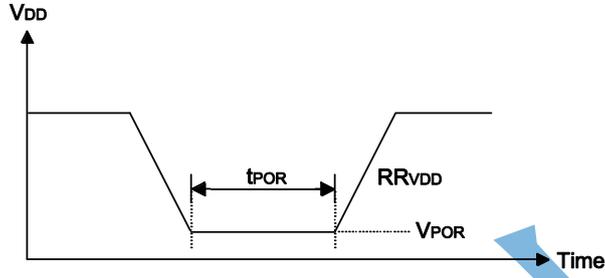
 $T_a=25^{\circ}\text{C}$

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
V_{DDC}	比较器工作电压	—	—	2.2	—	5.5	V
I_{DDC}	比较器工作电流	3.3V	—	—	20	40	μA
		5V	—	—	30	60	μA
V_{OPOS1}	比较器输入失调电压	5V	$C_{xOF4\sim 0}=(10000)$	-10	—	10	mV
V_{OPOS2}	比较器输入失调电压	5V	校准后	-4	—	4	mV
V_{CM}	比较器共模电压范围	—	—	V_{SS}	—	$V_{DD}-1.4\text{V}$	V
A_{OL}	比较器开环增益	—	—	60	80	—	dB

1.3.9 上电复位特性

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
V_{POR}	上电复位电压	—	—	—	—	100	mV

RR _{VDD}	上电复位电压速率	—	—	0.035	—	—	V/ms
t _{POR}	V _{DD} 保持为 V _{POR} 的最小时	—	—	1	—	—	ms



格瑞达

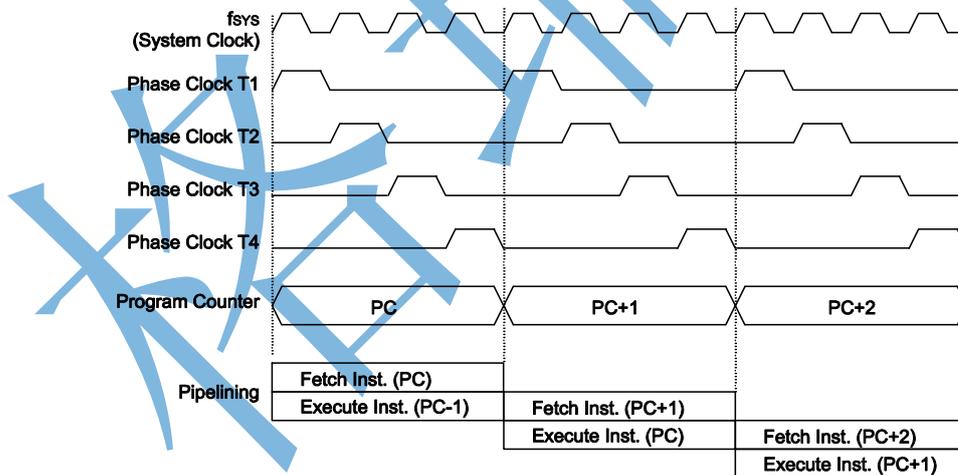
第 2 章 系统

2.1 系统结构

内部系统结构是格瑞达单片机具有良好性能的主要因素。由于采用 RISC 结构，该单片机具有高运算速度和高性能的特点。通过流水线的方式，指令的取得和执行同时进行，此举使得除了跳转和调用指令外，其它指令都能在一个指令周期内完成。8 位 ALU 参与指令集中所有的运算，它可完成算术运算、逻辑运算、移位、递增、递减和分支等功能，而内部的数据路径则是以通过累加器或 ALU 的方式加以简化。有些寄存器在数据存储中被实现，且可以直接或间接寻址。简单的寄存器寻址方式和结构特性，确保了在提供具有最大可靠性和灵活性的实用性 I/O 和 A/D 控制系统时，仅需要少数的外部器件。这些特性使得该单片机适用于低成本，大批量生产的控制应用。

2.1.1 时序和流水线结构

主系统时钟由 HIRC 或 LIRC 振荡器提供，它被细分为 T1~T4 四个内部产生的非重叠时序。在 T1 时间，程序计数器自动加一并抓取一条新的指令。剩下的时间 T2~T4 完成译码和执行功能，因此，一个 T1~T4 时间周期构成一个指令周期。虽然指令的抓取和执行发生在连续的指令周期，但单片机流水线结构会保证指令在一个指令周期内被有效执行。除非程序计数器的内容被改变，如子程序的调用或跳转，在这种情况下指令将需要多一个指令周期的时间去执行。



系统时序和流水线

如果指令牵涉到分支，例如跳转或调用等指令，则需要两个指令周期才能完成指令执行。需要

一个额外周期的原因是程序先用一个周期取出实际要跳转或调用的地址，再用另一个周期去实际执行分支动作，因此用户需要特别考虑额外周期的问题，尤其是在执行时间要求较严格的时候。



指令捕捉

2.1.2 程序计数器

在程序执行期间，程序计数器用来指向下一个要执行的指令地址。除了“JMP”和“CALL”指令需要跳转到一个非连续的程序存储器地址之外，它会在每条指令执行完成以后自动加一。然而只有较低的 8 位，即所谓的程序低字节寄存器 PCL，可以被用户直接读写。

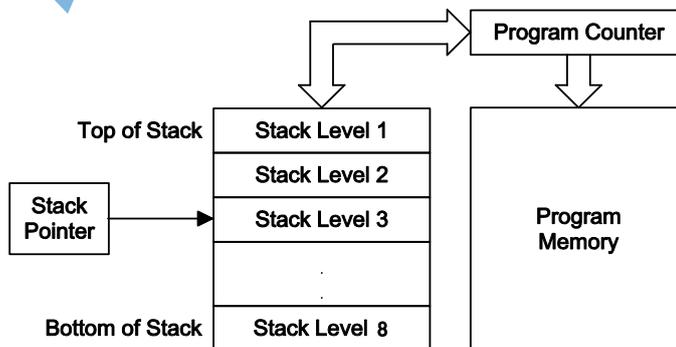
当执行的指令要求跳转到不连续的地址时，如跳转指令、子程序调用、中断或复位等，单片机通过加载所需要的位址到程序寄存器来控制程序，对于条件跳转指令，一旦条件符合，在当前指令执行时取得的下一条指令将会被舍弃，而由一个空指令周期来取代。

程序计数器	
程序计数器高字节	PCL 寄存器
PC11~PC8	PCL7~PCL0

程序计数器的低字节，即程序计数器的低字节寄存器 PCL，可以通过程序控制，且它是可以读取和写入的寄存器。通过直接写入数据到这个寄存器，一个程序短跳转可直接执行，然而只有低字节的操作是有效的，跳转被限制在存储器的当前页中，即 256 个存储器地址范围内，当这样一个程序跳转要执行时，会插入一个空指令周期。PCL 的使用可能引起程序跳转，因此需要额外的指令周期。

2.1.3 堆栈

堆栈是一个特殊的存储器空间，用来保存程序计数器中的值。该单片机含有多层堆栈。堆栈寄存器既不是数据存储器的一部分，也不是程序存储器的一部分，而且它既不能读出，也不能写入。堆栈的使用是通过堆栈指针 SP 来指示的，堆栈指针也不能读出或写入。当发生子程序调用或中断响应时，程序计数器中的内容会被压入堆栈；在子程序调用结束或中断响应结束时，执行指令 RET 或 RETI，堆栈将原先压入堆栈的内容弹出，重新装入程序计数器中。在系统复位后，堆栈指针会指向堆栈顶部。



如果堆栈已满，且有非屏蔽的中断发生，则只有中断请求标志位会被置位，而中断响应会被禁止，直到堆栈指针发生递减(执行 RET 或 RETI 指令)，中断才会被响应。这个特性提供程序设计者简单的方法来预防堆栈溢出。然而即使堆栈已满，CALL 指令仍然可以执行，从而造成堆栈溢出。使用时应避免堆栈溢出的情况发生，因为这样会造成不可预期的程序分支指令的执行错误。

如果堆栈溢出，第一个保存在堆栈中的 PC 会丢失。

2.1.4 算术逻辑单元 — ALU

算术逻辑单元是单片机中很重要的部分，执行指令集中的算术和逻辑运算。ALU 连接到单片机的数据总线，在接收相关的指令码后执行需要的算术与逻辑运算，并将结果储存在指定的寄存器，当 ALU 计算或操作时，可能导致进位、借位或其它状态的改变，而相关的状态寄存器会因此更新内容以显示这些改变，ALU 所提供的功能如下：

- 算术运算：ADD、ADDM、ADC、ADCM、SUB、SUBM、SBC、SBCM、DAA
- 逻辑运算：AND、OR、XOR、ANDM、ORM、XORM、CPL、CPLA
- 移位运算：RRA、RR、RRCA、RRC、RLA、RL、RLCA、RLC
- 递增和递减：INCA、INC、DECA、DEC
- 分支判断：JMP、SZ、SZA、SNZ、SIZ、SDZ、SIZA、SDZA、CALL、RET、RETI

2.2 Flash 程序存储器

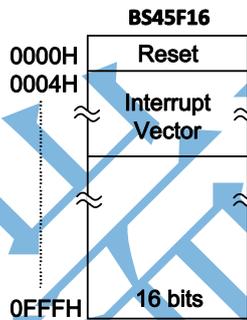
程序存储器用来存放用户代码即存储程序。该单片机提供可多次编程的存储器(Flash)，用户可以很方便的在同一个芯片中修改它们的应用代码。通过使用合适的编程工具，该 Flash 型单片机提供用户以灵活的方式自由开发它们的应用，这对于需要除错或需要经常升级和改变程序的产品是很有帮助的。

2.2.1 结构

程序存储器的容量为 4K×16 位。程序存储器用程序计数器来寻址，其中也包含数据、表格和中断入口，数据表格可以设定在程序存储器的任何地址，由表格指针来寻址。

2.2.2 特殊向量

程序存储器中某些地址保留用作诸如复位和中断的入口等特殊用途。0000H 是保留用做单片机复位后的程序起始地址。在芯片初始化后，程序将会跳转到这个地址并开始执行。



Flash 程序存储器结构

2.2.3 Flash 存储器部分锁保护

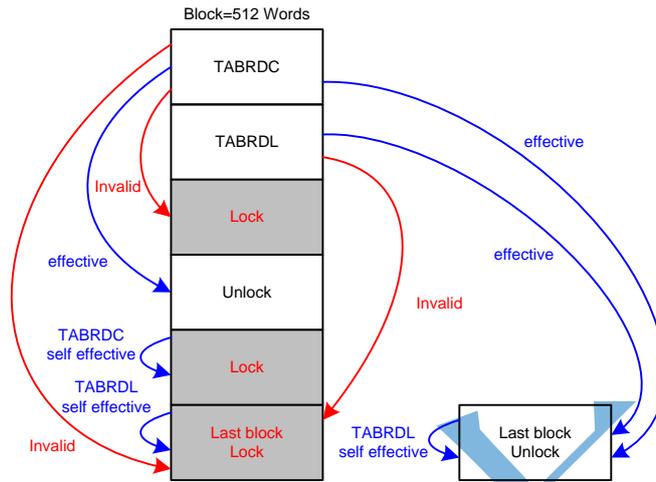
用于 Flash 程序存储器的部分锁保护功能可通过配置选项选择。部分锁保护控制位可保护 512 字的区域。如果部分锁保护控制位被置“1”，该区域被锁保护，使用者只能读到“00H”。

当 TBLP 和 TBHP 寄存器指针指向非锁保护区域并执行查表指令“TABRDC”或“TABRDL”时，读表指令有效。

当 TBLP 和 TBHP 寄存器指针指向锁保护区域时，出现以下两种状态：

1. 读表指令和表格数据在同一区域时，执行“TABRDC [m]”可将表格数据高字节传送到 TBLH，表格数据低字节传送到数据存储器[m]。
2. 读表指令和表格数据在不同区域时，执行“TABRDC [m]”无效。

如果最后一个区域被锁保护，在其他区域，设置 TBLP 和 TBHP 寄存器指向最后一页时，读表指令“TABRDC”和“TABRDL”无效且读到的数据为“00H”。但是，如果在最后一个区域设置 TBLP 寄存器指向最后一页时，读表指令“TABRDL”有效。



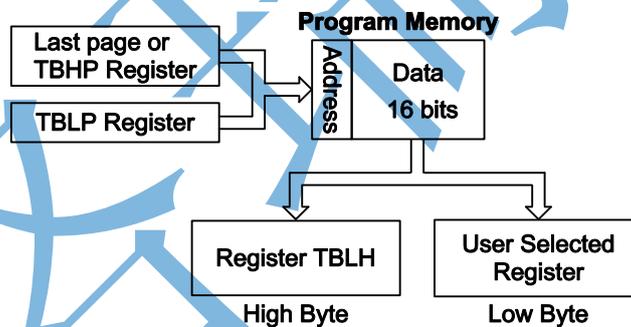
部分锁保护示意图

2.2.4 查表

程序存储器中的任何地址都可以定义为一个表格，以便存储固定的数据。使用查表指令时，查表指针需要先行设定，其方式是将表格的地址放在表格指针寄存器 TBLP 和 TBHP 中。这两个寄存器定义的是表格总的地址。

在设定完表格指针后，表格数据可以使用“TABRDC [m]”或“TABRDL [m]”指令从程序存储器中查表来读取。当这些指令执行时，程序存储器的表格的低字节，将会传输到用户所指定的数据存储器[m]中。程序存储器表格的高字节，将会传输到特殊寄存器 TBLH 中。传输数据中任何未定义的字节将会读取为“0”。

下图为查表寻址/数据流程图：



2.2.5 查表范例

以下范例说明在芯片中表格指针和表格数据如何被定义和执行。这个实例使用的表格数据用 ORG 伪指令储存在存储器的最后一页，在此 ORG 伪指令中的值为“F00H”，即 4K 程序存储器最后一页存储器的起始地址，而表格指针的初始值则为“06H”，这可保证从数据表格读取的第一笔数据位于程序存储器地址“F06H”，即最后一页起始地址后的第 6 个地址。注意，假如“TABRDC [m]”指令被使用，则表格指针指向当前页。在这个例子中，表格数据的高字节等于零，而当“TABRDL [m]”指令被执行时，此值将会自动的被传送到 TBLH 寄存器。

因为 TBLH 寄存器是只读寄存器，不能重新储存，若主程序和中断服务程序都使用表格读取指令，应该注意它的保护。使用表格读取指令，中断服务程序可能会改变 TBLH 的值，若随后在主程序中再次使用这个值，则会发生错误。因此建议避免同时使用表格读取指令。然而在某些情况下，如果同时使用表格读取指令是不可避免的，则在执行任何主程序的表格读取指令前，中断应该先除能，另外要注意，所有与表格相关的指令，都需要两个指令周期去完成操作。

指令	表格地址											
	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TABRDC[m]	P11	P10	P9	P8	@7	@6	@5	@4	@3	@2	@1	@0
TABRDL[m]	1	1	1	1	@7	@6	@5	@4	@3	@2	@1	@0

表格地址

注: b11~b0: 表格地址
 @7~@0: 表格指针TBLP 位
 P11~P8: 当前程序计数器位

2.2.6 表格读取程序范例

```

tempreg1 db ? ; temporary register #1
tempreg2 db ? ; temporary register #2
:
:
mov a,06h ; initialise low table pointer - note that this address
mov tblp,a ; is referenced
mov a,0Fh ; initialise high table pointer
tblp,a
:
:
tabrd tempreg1 ; transfers value in table referenced by table pointer data at program
; memory address "F06H" transferred to tempreg1 and TBLH
dec tblp ; reduce value of table pointer by one
tabrd tempreg2 ; transfers value in table referenced by table pointer data at program
; memory address "F05H" transferred to tempreg2 and TBLH in this
; example the data "1AH" is transferred to tempreg1 and data "0FH" to register tempreg2
:
:
org 0F00h ; sets initial address of program memory
dc 00Ah, 00Bh, 00Ch, 00Dh, 00Eh, 00Fh, 01Ah, 01Bh
    
```

2.3 数据存储器

数据存储器是内容可以更改的 8 位 RAM 内部存储器，用来存储临时数据。

2.3.1 结构

数据存储器分为两个部分，第一部分是特殊功能寄存器，这些寄存器有特定的地址且与单片机的正确操作密切相关。大多特殊功能寄存器都可在程序控制下直接读取和写入，而有些是被加以保护而不对用户开放。

第二部分是通用数据存储器，所有地址都可在程序的控制下进行读取和写入。

该单片机总的存储器被分为三个区。大部分特殊功能数据寄存器均可在所有 Bank 被访问，处于“40H”地址的 EEC 寄存器却只能在 Bank 1 中被访问到。切换不同区域可通过设置区域指针(BP)实现。该单片机的存储器的起始地址都是“00H”。

2.4 特殊功能寄存器描述

大部分特殊功能寄存器的细节将在相关功能中描述，但有几个寄存器在此章节单独描述。

2.4.1 间接寻址寄存器 — IAR0, IAR1

间接寻址寄存器 IAR0 和 IAR1，位于数据存储区，并没有实际的物理地址。间接寻址方式是使用间接寻址寄存器和存储器指针对数据操作，以取代定义在实际存储器地址的直接存储器寻址方式。在间接寻址寄存器上的任何动作，将对间接寻址指针(MP0 或 MP1)所指定的存储器地址产生对应的读/写操作。IAR0 和 MP0, IAR1 和 MP1 对数据存储区中数据的操作是成对出现的，MP0 和 IAR0 用于访问 Bank 0，而 MP1 和 IAR1 可访问所有的 Bank。间接寻址寄存器不是实际存在的，直接读取 IAR 寄存器将会返回 00H 的结果，而直接写入此寄存器则不做任何操作。

2.4.2 间接寻址指针 — MP0, MP1

该单片机提供两个间接寻址指针，即 MP0 和 MP1。由于这些指针在数据存储区中能像普通的寄存器一样被写入和操作，因此提供了一个有效的间接寻址和数据追踪的方法。当对间接寻址寄存器进行任何操作时，单片机所指向的实际地址是由间接寻址指针所指定的地址。MP0 和 IAR0 用于访问 Bank 0，而 MP1 和 IAR1 可访问所有的 Bank。注意，对于该单片机，间接寻址指针 MP0 和 MP1 为 8 位寄存器，常与 IAR0、IAR1 搭配一起对数据存储区寻址。

Bank0,2 Bank1		Bank0,2 Bank1	
00H	IAR0	40H	Unused
01H	MP0	41H	EEA
02H	IAR1	42H	EED
03H	MP1	43H	TMPC0
04H	BP	44H	TMPC1
05H	ACC	45H	Unused
06H	PCL	46H	Unused
07H	TBLP	47H	Unused
08H	TBLH	48H	TM1C0
09H	TBHP	49H	TM1C1
0AH	STATUS	4AH	TM1C2
0BH	SMOD	4BH	TM1DL
0CH	LVDC	4CH	TM1DH
0DH	INTEG	4DH	TM1AL
0EH	WDTC	4EH	TM1AH
0FH	TBC	4FH	TM1BL
10H	INTC0	50H	TM1BH
11H	INTC1	51H	TM2C0
12H	INTC2	52H	TM2C1
13H	MFI4	53H	TM2DL
14H	MFI0	54H	TM2DH
15H	MFI1	55H	TM2AL
16H	MFI2	56H	TM2AH
17H	MFI3	57H	CTRL
18H	PAWU	58H	Unused
19H	PAPU	59H	Unused
1AH	PA	5AH	Unused
1BH	PAC	5BH	Unused
1CH	PBPU	5CH	Unused
1DH	PB	5DH	SLCDC0
1EH	PBC	5EH	SLCDC1
1FH	PCPU	5FH	SLCDC2
20H	PC	60H	Unused
21H	PCC	61H	Unused

22H	PDPU	62H	Unused
23H	PD	63H	Unused
24H	PDC	64H	Unused
25H	CPC0	65H	Unused
26H	CPC1	66H	Unused
27H	OPAC0	67H	Unused
28H	OPAC1	68H	Unused
29H	OPAC2	69H	Unused
2AH	Unused	6AH	Unused
2BH	ADRL	6BH	Unused
2CH	ADRH	6CH	Unused
2DH	ADCR0	6DH	Unused
2EH	ADCR1	6EH	Unused
2FH	ACERL	6FH	Unused
30H	DAC0	70H	Unused
31H	DAC1	71H	Unused
32H	DAC2		Unused
33H	Unused		Unused
34H	SLEDC		Unused
35H	LVRC		Unused
36H	Unused		Unused
37H	Unused		Unused
38H	Unused	7FH	Unused
39H	Unused	80H	General Purpose Date mermory (128Bytes)
3AH	TM0C0		
3BH	TM0C1		
3CH	TM0DL		
3DH	TM0DH		
3EH	TM0AL		
3FH	TM0AH	FFH	

特殊功能数据存储

注：表中“Unused”字节不能由用户修改。

以下范例说明如何清除一个具有 4 个 RAM 地址的区块，它们已经事先被定义成地址 adres1 到 adres4。

2.4.3 间接寻址程序范例

```
data . section 'data'
adres1 db ?
adres2 db ?
adres3 db ?
adres4 db ?
block db ?
code . section at 0 'code'
org 00h
```

```
start:
    mov a,04h ;setup size of block
    mov block,a
    mov a,offset adres1 ; Accumulator loaded with first RAM address
    mov mp0,a ; setup memory pointer with first RAM address

loop:
    clr IAR0 ; clear the data at address defined by MP0
    inc mp0 ; increment memory pointer
    sdz block ; check if last memory location has been cleared
    jmp loop
continue:
```

在以上的例子中，没有提及具体的数据存储地址。

2.4.4 存储区指针 — BP

该单片机数据存储区被分为三个部分。可以通过设置存储区指针（Bank Pointer）值来访问不同的数据存储区。位 0 和 1 用来选择数据存储区 Bank 0~2。

复位后，数据存储区会初始化到 Bank 0，但是在暂停模式下的 WDT 溢出复位，不会改变通用数据存储区的存储区号。应该注意的是特殊功能数据存储区不受存储区的影响，也就是说，不论是在哪一个存储区，都能对特殊功能寄存器进行读写操作。数据存储区的直接寻址总是访问 Bank 0，不影响存储区指针的值。要访问除 Bank 0 外的其它 Bank，则必须要使用间接寻址方式。

2.4.5 存储区指针寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	DMBP1	DMBP0
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2

未定义，读为“0”

Bit 1~0

DMBP1, DMBP0: 数据存储区选择

00: Bank 0

01: Bank 1

10: Bank 2

11: 未定义

2.4.6 累加器 — ACC

对于任何单片机来说，累加器是相当重要的，且与 ALU 所完成的运算有密切关系，所有的 ALU 得到的运算结果都将暂存在累加器中，如果没有累加器，ALU 必须在每次进行如加法、减法和移位等运算时，将结果写入数据存储区中，这样会造成程序编写和时间的负担。另外，数据传输通常涉及到累加器的临时储存功能，如在用户定义的寄存器和另一个寄存器之间，由于两者之间的不能直接传送数据，因此需要通过累加器来传送数据。

2.4.7 程序计数器低字节寄存器 — PCL

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

2.4.8 表格寄存器 — TBLP, TBHP, TBLH

这三个特殊功能寄存器对存储在程序存储器中的表格进行操作。TBLP 和 TBHP 为表格指针，指向表格的地址。它的值必须在任何表格读取指令执行前加以设定。由于它的值可以被如 INC 或 DEC 的指令所改变，这就提供了一种简单的方法对表格数据进行读取。表格读取数据指令执行之后，表格数据高字节存储在 TBLH 中。其中要注意的是，表格数据低字节会被传送到用户指定的地址。

2.4.9 状态寄存器 — STATUS

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

除了 TO 和 PDF 标志位以外，状态寄存器的其它位像其它大多数寄存器一样可以被改变。但是任何数据写入状态寄存器将不会改变 TO 和 PDF 标志位。另外，执行不同指令操作后，与状态寄存器相关的运算将会得到不同的结果。TO 标志位只会受系统上电、看门狗溢出、或执行“CLR WDT”或“HALT”指令的影响。PDF 指令只会受执行“HALT”或“CLR WDT”指令或系统上电的影响。

Z、OV、AC 和 C 标志位通常反映最近的运算操作的状态。

- ◆ **C**: 当加法运算的结果产生进位, 或减法运算的结果没有产生借位时, 则 C 被置位, 否则 C 被清零, 同时 C 也会被带进位的移位指令所影响。
- ◆ **AC**: 当低半字节加法运算的结果产生进位, 或高半字节减法运算的结果没有产生借位时, AC 被置位, 否则 AC 被清零。
- ◆ **Z**: 当算术或逻辑运算结果是零时, Z 被置位, 否则 Z 被清零。
- ◆ **OV**: 当运算结果高两位的进位状态异或结果为 1 时, OV 被置位, 否则 OV 被清零。
- ◆ **PDF**: 系统上电或执行“CLR WDT”指令会清零 PDF, 而执行“HALT”指令则会置位 PDF。
- ◆ **TO**: 系统上电或执行“CLR WDT”或“HALT”指令会清零 TO, 而当 WDT 溢出则会置位 TO。

另外, 当进入一个中断程序或执行子程序调用时状态寄存器将不会自动压入到堆栈中保存。假如状态寄存器的内容很重要, 且中断子程序会改变状态寄存器的内容, 则需要保存备份以备恢复。注意, 状态寄存器的 0~3 位可以读取和写入。

• 状态寄存器

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

“x”表示未知

- Bit 7, 6 未定义, 读为“0”
- Bit 5 **TO**: 看门狗溢出标志位
0: 系统上电或执行“CLR WDT”或“HALT”指令
1: WDT 溢出
- Bit 4 **PDF**: 暂停标志位
0: 系统上电或执行“CLR WDT”指令
1: 执行“HALT”指令将会置位 PDF 位。
- Bit 3 **OV**: 溢出标志位
0: 不发生溢出时
1: 当运算结果高两位的进位状态异或结果为 1 时
- Bit 2 **Z**: 零标志位
0: 算数运算或逻辑运算的结果不为零时
1: 算数运算或逻辑运算的结果为零时
- Bit 1 **AC**: 辅助进位标志位
0: 没有辅助进位时
1: 当低字节的加法造成进位或高字节的减法没有造成借位时
- Bit 0 **C**: 进位标志位
0: 没有进位时
1: 当加法造成进位或减法没有造成借位时, 同时移位指令也会影响 C 标志位
C 也受循环移位指令的影响。

2.5 EEPROM 数据存储

该单片机内建 EEPROM 数据存储。 “Electrically Erasable Programmable Read Only Memory” 为电可擦可编程只读存储器，由于其非易失性的存储结构，即使在电源掉电的情况下存储器内的数据仍然保存完好。通过 EEPROM 数据存储，可以给设计者提供一个全新的主机应用。应用 EEPROM 的存储功能使得该单片机可以用来存储产品编号、校准值、用户特定数据、系统配置参数或其它产品信息等。读取和写数据到 EEPROM 存储器的过程已经变得很容易。

2.5.1 EEPROM 数据存储结构

该单片机内部 EEPROM 数据寄存器容量为 64x8 位。由于映射方式与程序存储器和数据存储器不同，因此不能像其它类型的存储器一样寻址。使用 Bank 0 中的一个地址和数据寄存器以及 Bank 1 中的一个控制寄存器，可以实现对 EEPROM 的单字节读写操作。

2.5.2 EEPROM 寄存器

有三个寄存器控制内部 EEPROM 数据存储总的操作。地址寄存器 EEA、数据寄存器 EED 及控制寄存器 EEC。EEA 和 EED 位于 Bank 0 中，它们能像其它特殊功能寄存器一样直接被访问。EEC 位于 Bank 1 中，不能被直接访问，仅能通过 MP1 和 IAR1 进行间接读取或写入。由于 EEC 控制寄存器位于 Bank 1 中的“40H”，在 EEC 寄存器上的任何操作被执行前，MP1 必须先设为“40H”，BP 被设为“01H”。

2.5.3 EEPROM 寄存器列表

Name	7	6	5	4	3	2	1	0
EEA	—	—	D5	D4	D3	D2	D1	D0
EED	D7	D6	D5	D4	D3	D2	D1	D0
EEC	—	—	—	—	WREN	WR	RDEN	RD

• EEA 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	D5	D4	D3	D2	D1	D0
R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	—	×	×	×	×	×	×

“×”为未知

Bit 7~6 未使用，读为“0”
 Bit 5~0 数据 EEPROM 地址
 数据 EEPROM 地址 Bit 5~Bit 0

• EEC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	WREN	WR	RDEN	RD
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

Bit 7~4 未定义，读为“0”
 Bit 3 **WREN**: 数据 EEPROM 写使能位
 0: 除能
 1: 使能

此位为数据 EEPROM 写使能位，向数据 EEPROM 写操作之前需将此位置高。将此位清零时，则禁止向数据 EEPROM 写操作。

Bit 2 **WR**: EEPROM 写控制位
 0: 写周期结束

- 1: 写周期有效
 此位为数据 EEPROM 写控制位, 由应用程序将此位置高将激活写周期。写周期结束后, 硬件自动将此位清零。当 WREN 未先置高时, 此位置高无效。
- Bit 1 **RDEN:** 数据 EEPROM 读使能位
 0: 除能
 1: 使能
 此位为数据 EEPROM 读使能位, 向数据 EEPROM 读操作之前需将此位置高。将此位清零时, 则禁止向数据 EEPROM 读操作。
- Bit 0 **RD:** EEPROM 读控制位
 0: 读周期结束
 1: 读周期有效
 此位为数据 EEPROM 读控制位, 由应用程序将此位置高将激活读周期。读周期结束后, 硬件自动将此位清零。当 RDEN 未首先置高时, 此位置高无效。
- 注: 在同一条指令中 WREN、WR、RDEN 和 RD 不能同时置为“1”。WR 和 RD 不能同时置为“1”。

• EED 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	×	×	×	×	×	×	×	×

“x”为未知

Bit 7~0 数据 EEPROM 数据
 数据 EEPROM 数据 Bit 7~Bit 0

2.5.4 从 EEPROM 中读取数据

从 EEPROM 中读取数据, EEC 寄存器中的读使能位 RDEN 先置为高以使能读功能, EEPROM 中读取数据的地址要先放入 EEA 寄存器中。若 EEC 寄存器中的 RD 位被置高, 一个读周期将开始。若 RD 位已置为高而 RDEN 位还未被设置则不能开始读操作。若读周期结束, RD 位将自动清除为“0”, 数据可以从 EED 寄存器中读取。数据在其它读或写操作执行前将一直保留在 EED 寄存器中。应用程序将轮询 RD 位以确定数据可以有效地被读取。

2.5.5 写数据到 EEPROM

EEPROM 中写入数据的地址要先放入 EEA 寄存器中, 写数据至 EEPROM, EEC 寄存器中的写使能位 WREN 先置为高以使能写功能。之后, EEC 寄存器中的 WR 位必须立即置为高以开始写周期。这两个指令必须连续执行。总中断控制位 EMI 必须在写周期执行前清零, 并在写周期开始后置为高。注意, 若 WR 位已置为高而 WREN 位还未被设置则不能开始写操作。由于控制 EEPROM 写周期是一个内部时钟, 与单片机的系统时钟异步, 所以数据写入 EEPROM 的时间将有所延迟。可通过轮询 EEC 寄存器中的 WR 位或判断 EEPROM 中断以侦测写周期是否完成。若写周期完成, WR 位将自动清除为“0”, 通知用户数据已写入 EEPROM。因此, 应用程序将轮询 WR 位以确定写周期是否结束。

2.5.6 写保护

防止误写入的写保护有以下几种。单片机上电后控制寄存器中的写使能位将被清除以杜绝任何写入操作。上电后 BP 将重置为“0”, 这意味着数据存储区 Bank 0 被选中。由于 EEPROM 控制寄存器位于 Bank 1 中, 这增加了对写操作的保护措施。在正常程序操作中确保控制寄存器中的写使能位被清除将能防止不正确的写操作。

2.5.7 EEPROM 中断

EEPROM 写周期结束后将产生 EEPROM 写中断, 需先通过设置相关中断寄存器的 DEE 位使能 EEPROM 中断。由于 EEPROM 中断包含在多功能中断中, 相应的多功能中断使能位需被设置。当 EEPROM 写周期结束, DEF 请求标志位及其相关多功能中断请求标志位将被置位。若 EEPROM 和多功能中断使能且堆栈未满的情况下将跳转到相应的多功能中断向量中执行。当中断被响应, 只有多功能中断标志位将自动复位, 而 EEPROM 中断标志将通过应用程序手动复位。更多细节将在中

断章节讲述。

2.5.8 编程注意事项

必须注意的是数据不会无意写入 EEPROM。在没有写动作时写使能位被正常清零可以增强保护功能。BP 指针也可以正常清零以阻止进入 EEPROM 控制寄存器存在的 Bank 1。尽管没有必要，写一个简单的读回程序以检查新写入的数据是否正确还是应该考虑的。

写数据时，为了确保写周期正确执行，WR 位必须在 WREN 位置为高之后立即置为高。总中断控制位 EMI 必须在写周期执行前清零，并在写周期开始后置为高。

2.5.9 程序举例

◆ 从 EEPROM 中读取数据—轮询法

```

MOV A, EEPROM_ADRES      ; user defined address
MOV EEA, A
MOV A, 040H              ; setup memory pointer MP1
MOV MP1, A               ; MP1 points to EEC register
MOV A, 01H               ; setup Bank Pointer
MOV BP, A
SET IAR1.1               ; set RDEN bit, enable read operations
SET IAR1.0               ; start Read Cycle - set RD bit
BACK:
SZ IAR1.0                ; check for read cycle end
JMP BACK
CLR IAR1                  ; disable EEPROM read/write
CLR BP
MOV A, EED                ; move read data to register
MOV READ_DATA, A
    
```

◆ 写数据到 EEPROM—轮询法

```

CLR EMI
MOV A, EEPROM_ADRES     ; user defined address
MOV EEA, A
MOV A, EEPROM_DATA      ; user defined data
MOV EED, A
MOV A, 040H              ; setup memory pointer MP1
MOV MP1, A               ; MP1 points to EEC register
MOV A, 01H               ; setup Bank Pointer
MOV BP, A
SET IAR1.3               ; set WREN bit, enable write operations
SET IAR1.2               ; Start Write Cycle - set WR bit - executed immediately
                        ; after set WREN bit
SET EMI
BACK:
SZ IAR1.2                ; check for write cycle end
JMP BACK
CLR IAR1                  ; disable EEPROM read/write
CLR BP
    
```

2.6 振荡器

不同的振荡器选择可以让使用者在不同的应用需求中实现更大范围的功能。振荡器的灵活性使得在速度和功耗之间可以达到最优化。振荡器选项是通过寄存器来完成的。

2.6.1 振荡器概述

该单片机有两个内部振荡器，一个低速振荡器和一个高速振荡器。它们都可以作为系统时钟源，低速振荡器还可以作为看门狗定时器，时基功能和定时/计数器的时钟源。集成的两个内部振荡器不需要任何外接器件。所有振荡器选项通过寄存器设置。较高频率的振荡器提供更高的性能，但要求有更高的功率，反之亦然。动态切换快慢系统时钟的能力使单片机具有灵活而优化的性能/功耗比，此特性对功耗敏感的应用领域尤为重要。

振荡类型	名称	频率范围
内部高速振荡器	HIRC	8,12MHz
内部低速振荡器	LIRC	32kHz

振荡器类型

2.6.2 系统时钟配置

该单片机有两个方式产生系统时钟，一个高速内部时钟源和一个低速内部时钟源。高速振荡器为内部 8MHz 或 12MHz RC 振荡器，低速振荡器为内部 32 kHz RC 振荡器。这两个振荡器都是内部全集成的振荡器，无需外接器件。选择高速或低速振荡器作为系统振荡器，是通过 SMOD 寄存器中的 HLCLK 位及 CKS2~CKS0 位进行选择。

2.6.3 内部高速 RC 振荡器 — HIRC

内部 RC 振荡器是一个集成的系统振荡器，不需其它外部器件。内部 RC 振荡器具有两种固定的频率：8MHz 和 12MHz。芯片在制造时进行调整且内部含有频率补偿电路，使得振荡频率因 V_{DD} 、温度以及芯片制成工艺不同的影响减至最低程度。在电源电压为 3V 或 5V 及温度为 25°C 的条件下，8MHz 和 12MHz 这两个固定频率的容差为 2%。

• CTRL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	FSYSON	—	HIRCS1	HIRCS0	—	LVRF	LRF	WRF
R/W	R/W	—	R/W	R/W	—	R/W	R/W	R/W
POR	0	—	0	0	—	×	0	0

“×”未知

- Bit 7 **FSYSON**: 空闲模式中 f_{SYS} 控制位
 0: 除能
 1: 使能
- Bit 6 未使用，读为“0”
- Bit 5,4 **HIRCS1,HIRCS0**: 高频时钟选择位
 00: 8 MHz
 01: 8 MHz
 10: 12 MHz
 11: 12 MHz
- Bit 3 未使用，读为“0”
- Bit 2 **LVRF**: LVR 复位标志位
 0: 无效
 1: 有效
- Bit 1 **LRF**: LVR 控制的复位标志位
 0: 无效

1: 有效
Bit 0 **WRF**: WE[4:0]控制的复位标志位
0: 无效
1: 有效
此位只能被清零，不能被置“1”。

2.6.4 内部低速 RC 振荡器 — LIRC

内部 32kHz 系统振荡器为低频振荡器。这种单片机有一个完全集成 RC 振荡器，它在 5V 电压下运行的典型频率值为 32kHz 且无需外部元件。芯片在制造时进行调整且内部含有频率补偿电路，使得振荡器因电源电压、温度及芯片制成工艺不同的影响减至最低。因此，内部 32kHz 振荡器频率在 25°C 温度 5V 电压下的精度保持在 10% 以内。

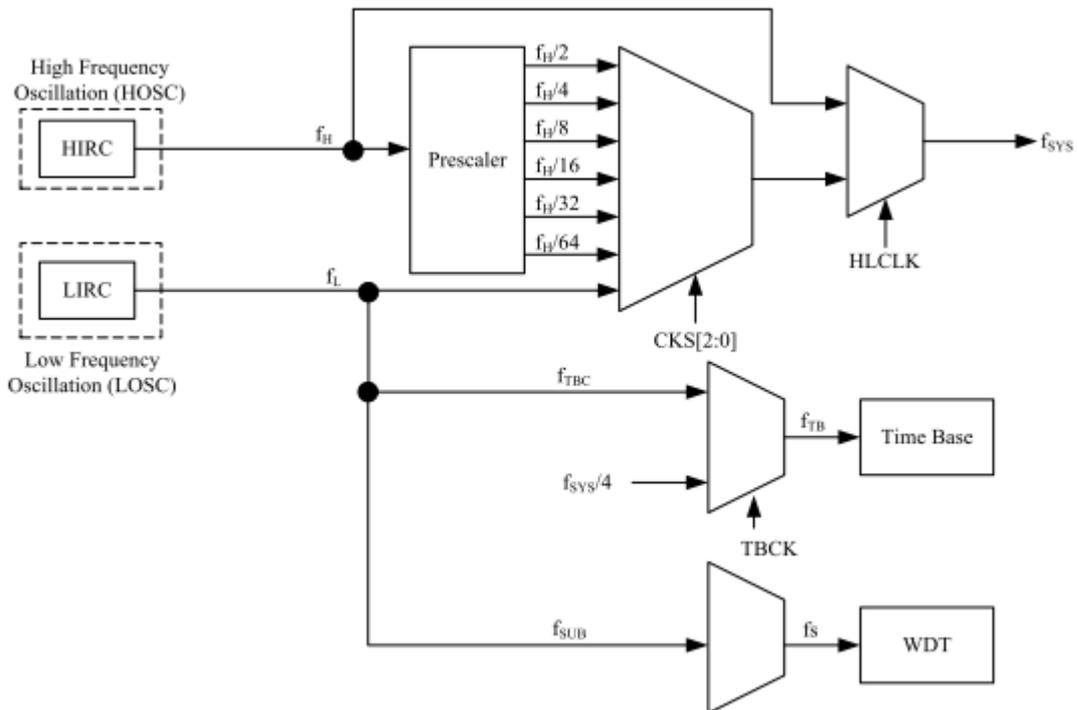


2.7 工作模式和系统时钟

现今的应用要求单片机具有较高的性能及尽可能低的功耗，这种矛盾的要求在便携式电池供电的应用领域尤为明显。高性能所需要的高速时钟将增加功耗，反之亦然。此单片机提供高、低速两种时钟源，它们之间可以动态切换，用户可通过优化单片机操作来获得最佳性能/功耗比。

2.7.1 系统时钟

主系统时钟可来自高频时钟源 f_H 或低频时钟源 f_L ，通过 SMOD 寄存器中的 HLCLK 位及 CKS2~CKS0 位进行选择。高频时钟和低频时钟都来自内部 RC 振荡器。



系统时钟配置

2.7.2 控制寄存器

寄存器 SMOD 用于控制单片机内部时钟。

• SMOD 寄存器

Bit	7	6	5	4	3	2	1	0
Name	CKS2	CKS1	CKS0	—	LTO	HTO	IDLEN	HLCLK
R/W	R/W	R/W	R/W	—	R	R	R/W	R/W
POR	0	0	0	—	0	0	1	1

Bit 7~5 **CKS2~CKS0**: 当 HLCLK 为“0”时系统时钟选择位

- 000: f_L (f_{LIRC})
- 001: f_L (f_{LIRC})
- 010: $f_H/64$
- 011: $f_H/32$
- 100: $f_H/16$
- 101: $f_H/8$
- 110: $f_H/4$
- 111: $f_H/2$

这三位用于选择系统时钟源。除了 LIRC 振荡器提供的系统时钟源外，也可使用高频振荡器的分频作为系统时钟。

- Bit 4 未使用，读为“0”
- Bit 3 **LTO**: 低速振荡器就绪标志位
 0: 未就绪
 1: 就绪
 此位为低速系统振荡器就绪标志位，用于表明低速系统振荡器在系统上电复位或经唤醒后何时稳定下来。该标志由休眠模式 0 中唤醒后会处于低电平状态，若系统时钟来自 LIRC 振荡器，该位转换为高需 1~2 个时钟周期。
- Bit 2 **HTO**: 高速振荡器就绪标志位
 0: 未就绪
 1: 就绪
 此位为高速系统振荡器就绪标志位，用于表明高速系统振荡器何时稳定下来。此标志在系统上电后经硬件清零，高速系统振荡器稳定后变为高电平。因此，此位在单片机上电后由应用程序读取的总为“1”。该标志由休眠模式或空闲模式 0 中唤醒后会处于低电平状态，若使用 HIRC 振荡器，15~16 个时钟周期后该标志会处于高电平状态。
- Bit 1 **IDLEN**: 空闲模式控制位
 0: 除能
 1: 使能
 此位为空闲模式控制位，用于决定 HALT 指令执行后发生的动作。若此位为高，当指令 HALT 执行后，单片机进入空闲模式。若 FSYSON 位为高，在空闲模式 1 中 CPU 停止运行，系统时钟将继续工作以保持外围功能继续工作；若 FSYSON 为低，在空闲模式 0 中 CPU 和系统时钟都将停止运行。若此位为低，单片机将在 HALT 指令执行后进入休眠模式。
- Bit 0 **HLCLK**: 系统时钟选择位
 0: $f_H/2 \sim f_H/64$ 或 f_L
 1: f_H
 此位用于选择 f_H 或 $f_H/2 \sim f_H/64$ 还是 f_L 作为系统时钟。该位为高时选择 f_H 作为系统时钟，为低时则选择 $f_H/2 \sim f_H/64$ 或 f_L 作为系统时钟。当系统时钟由 f_H 时钟向 f_L 时钟转换时， f_H 将自动关闭以降低功耗。

2.7.3 系统工作模式

该单片机有 6 种不同的工作模式，每种有它自身的特性，根据应用中不同的性能和功耗要求可选择不同的工作模式。单片机正常工作有两种模式：正常模式和低速模式。剩余的 4 种工作模式：休眠模式 0、休眠模式 1、空闲模式 0 和空闲模式 1 用于单片机 CPU 关闭时以节省耗电。

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

- 正常模式

顾名思义，这是主要的工作模式之一，单片机的所有功能均可在此模式中实现且系统时钟由一个高速振荡器提供。该模式下单片机正常工作的时钟源来自 HIRC 振荡器。高速振荡器频率可被分为 1~64 的不等比率，实际的比率由 SMOD 寄存器中的 CKS2~CKS0 位及 HLCLK 位选择的。单片机使用高速振荡器分频作为系统时钟可减少工作电流。

- 低速模式

此模式的系统时钟虽为较低速时钟源，但单片机仍能正常工作。该低速时钟源可来自 LIRC 振荡器。单片机在此模式中运行所耗工作电流较低。在低速模式下， f_H 关闭。

- 休眠模式 0

在 HALT 指令执行后且 SMOD 寄存器中 IDLEN 位为低时，系统进入休眠模式 0。在休眠模式 0

中，CPU、 f_{SUB} 及 f_S 停止运行，看门狗定时器功能除能。在该模式中 LV DEN 位需置为“0”，否则将不能进入休眠模式 0 中。

• 休眠模式 1

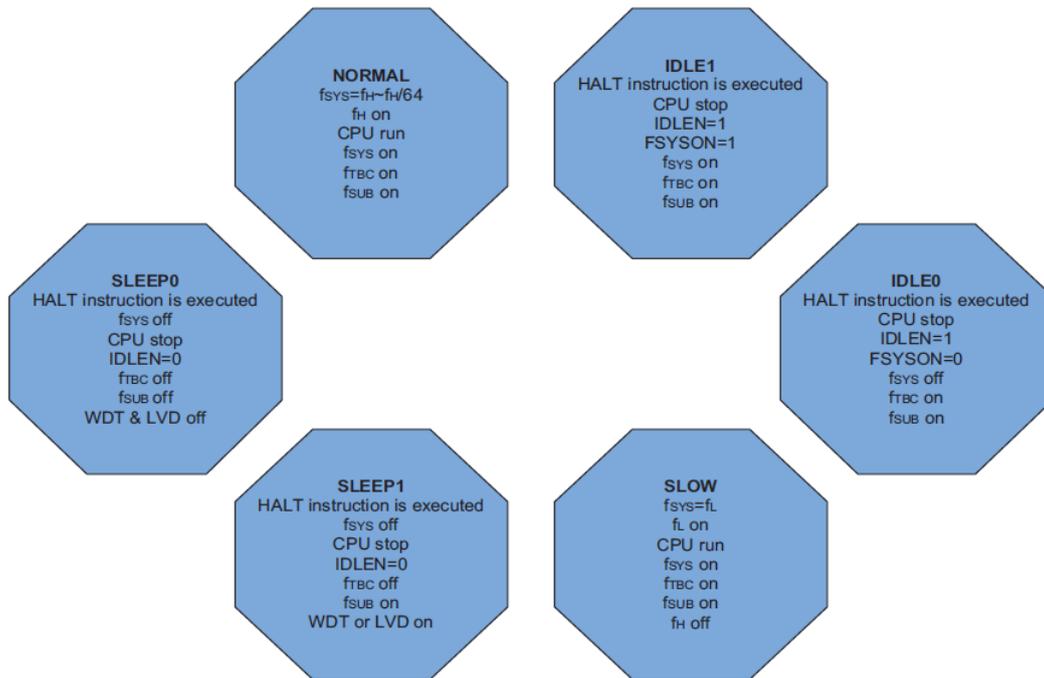
在 HALT 指令执行后且 SMOD 寄存器中 IDLEN 位为低时，系统进入休眠模式 1。在休眠模式 1 中，CPU 停止运行。然而当其时钟源经配置选项选择为 f_{SUB} 时，若 LV DEN 位为“1”或看门狗定时器功能使能， f_{SUB} 及 f_S 继续运行。

• 空闲模式 0

执行 HALT 指令后且 SMOD 寄存器中 IDLEN 位为高，CTRL 寄存器中 FSYSON 位为低时，系统进入空闲模式 0。在空闲模式 0 中，CPU 停止，但一些外围功能如看门狗定时器、TMs 将继续工作。在空闲模式 0 中，系统振荡器停止。

• 空闲模式 1

执行 HALT 指令后且 SMOD 寄存器中 IDLEN 位为高，CTRL 寄存器中 FSYSON 位为高时，系统进入空闲模式 1。在空闲模式 1 中，CPU 停止，但会提供一个时钟源给一些外围功能如看门狗定时器、TMs。在空闲模式 1 中，系统振荡器继续运行，该系统振荡器可以为高速或低速系统振荡器。

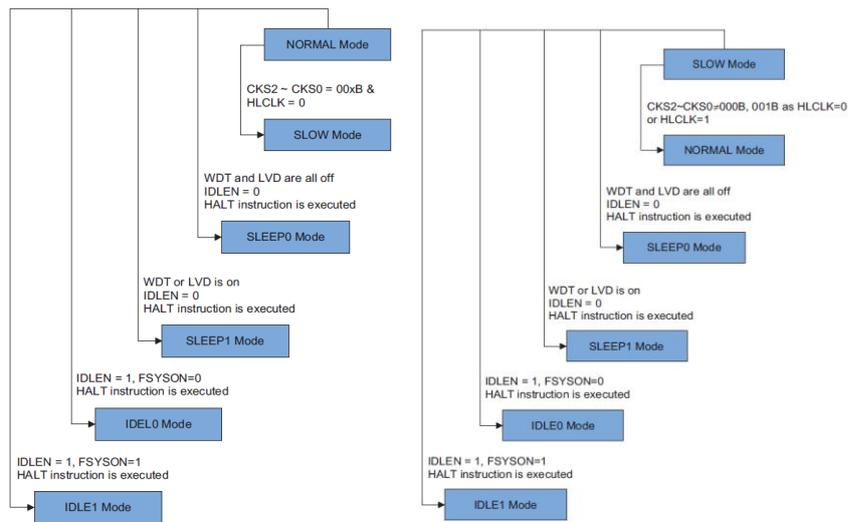


2.7.4 工作模式切换

单片机可在各个工作模式间自由切换，使得用户可根据所需选择最佳的性能/功耗比。用此方式，对单片机工作的性能要求不高的情况下，可使用较低频时钟以减少工作电流，在便携式应用上延长电池的使用寿命。

简单来说，正常模式和低速模式间的切换仅需设置 SMOD 寄存器中的 HLCLK 位及 CKS2~CKS0 位即可实现，而正常模式/低速模式与休眠模式/空闲模式间的切换经由 HALT 指令实现。当 HALT 指令执行后，单片机是否进入空闲模式或休眠模式由 SMOD 寄存器中的 IDLEN 位和 CTRL 寄存器中的 FSYSON 位决定的。

当 HLCLK 位变为低电平时，时钟源将由高速时钟源 f_H 转换成时钟源 $f_H/2 \sim f_H/64$ 或 f_L 。若时钟源来自 f_L ，高速时钟源将停止运行以节省耗电。此时须注意， $f_H/16$ 和 $f_H/64$ 内部时钟源也将停止运行。所附流程图显示了单片机在不同工作模式间切换时的变化。



正常模式切换到低速模式

系统运行在正常模式时使用高速系统振荡器，因此较为耗电。可通过设置 SMOD 寄存器中的 HLCLK 位为“0”及 CKS2~CKS0 位为“000”或“001”使系统时钟切换至运行在低速模式下。此时将使用低速系统振荡器以节省耗电。用户可在对性能要求不高的操作中使用此方法以减少耗电。

低速模式的时钟源来自 LIRC 振荡器。因此要求这些振荡器在所有模式切换动作发生前稳定下来。该动作由 SMOD 寄存器中 LTO 位控制。

低速模式切换到正常模式

在低速模式系统使用 LIRC 低速振荡器。切换到使用高速系统时钟振荡器的正常模式需设置 HLCLK 位为“1”，也可设置 HLCLK 位为“0”但 CKS2~CKS0 需设为“010”、“011”、“100”、“101”、“110”或“111”。高频时钟需要一定的稳定时间，通过检测 HTO 位的状态可进行判断。高速振荡器的稳定时间由所使用高速系统振荡器的类型决定。

进入休眠模式 0

进入休眠模式 0 的方法仅有一种——应用程序中执行“HALT”指令前需设置寄存器 SMOD 中 IDLEN 位为“0”且 WDT 和 LVD 功能除能。在上述条件下执行该指令后，将发生的情况如下：

- 系统时钟、WDT 时钟和时基时钟停止运行，应用程序停止在“HALT”指令处。
- 数据存储器和寄存器的内容将保持当前值。
- WDT 都将被清除并停止运行。
- 输入/输出将保持当前值。
- 状态寄存器中暂停标志 PDF 将被置起，看门狗溢出标志 TO 将被清除。

进入休眠模式 1

进入休眠模式 1 的方法仅有一种——应用程序中执行“HALT”指令前需设置寄存器 SMOD 中 IDLEN 位为“0”且 WDT 或 LVD 功能使能。在上述条件下执行该指令后，将发生的情况如下：

- 系统时钟和时基时钟停止运行，应用程序停止在“HALT”指令处。WDT 或 LVD 继续运行，其时钟源来自 f_{SUB} 。
- 数据存储器和寄存器的内容将保持当前值。
- 若 WDT 使能，则 WDT 将被清零并重新开始计数。
- 输入/输出将保持当前值。
- 状态寄存器中暂停标志 PDF 将被置起，看门狗溢出标志 TO 将被清除。

进入空闲模式 0

进入空闲模式 0 的方法仅有一种，应用程序中执行“HALT”指令前需设置寄存器 SMOD 中 IDLEN 位为“1”且 CTRL 寄存器中的 FSYSON 位为“0”。在上述条件下执行该指令后，将发生的情况如下：

- 系统时钟停止运行，应用程序停止在“HALT”指令处，时基时钟和 f_{LIRC} 时钟将继续运行。
- 数据存储器和寄存器的内容将保持当前值。

- 若 WDT 使能，WDT 将被清零并重新开始计数
- 输入/输出口将保持当前值。
- 状态寄存器中暂停标志 PDF 将被置起，看门狗溢出标志 TO 将被清除。

进入空闲模式 1

进入空闲模式 1 的方法仅有一种，应用程序中执行“HALT”指令前需设置寄存器 SMOD 中 IDLEN 位为“1”且 CTRL 寄存器中的 FSYSON 位为“1”。在上述条件下执行该指令后，将发生的情况如下：

- 系统时钟和 f_{LIRC} 开启，应用程序停止在“HALT”指令处。
- 数据存储器和寄存器将保持当前值。
- 若 WDT 使能，WDT 将被清零并重新开始计数。
- 输入/输出口将保持当前值。
- 状态寄存器中暂停标志 PDF 将被置起，看门狗溢出标志 TO 将被清除。

2.7.5 静态电流的注意事项

由于单片机进入休眠或空闲模式的主要原因是将 MCU 的电流降低到尽可能低，可能到只有几个微安的级别（空闲模式 1 除外），所以如果要降低电路的电流，电路设计者还应有其它的考虑。应该特别注意的是单片机的输入/输出引脚。所有高阻抗输入脚都必须连接到固定的高或低电平，因为引脚浮空会造成内部振荡并导致耗电增加。这也应用于有不同封装的单片机，因为它们可能含有未引出的引脚，这些引脚也必须设为输出或带有上拉电阻的输入。

另外还需注意单片机设为输出的 I/O 引脚上的负载。应将它们设置在有最小拉电流的状态或将它们和其它的 CMOS 输入一样接到没有拉电流的外部电路上。还应注意的是，如果使能配置选项中的 LIRC 振荡器，会导致耗电增加。

在空闲模式 1 中，系统时钟开启。若系统时钟来自高速系统振荡器，额外的静态电流也可能会有几百微安。

2.7.6 唤醒

系统进入休眠或空闲模式之后，可以通过以下几种方式唤醒：

- 外部复位
- PA 口下降沿
- 系统中断
- WDT 溢出

若由外部唤醒，系统会经过完全复位的过程；若由 WDT 溢出唤醒，则会发生看门狗定时器复位。这两种唤醒方式都会使系统复位，可以通过状态寄存器中 TO 和 PDF 位来判断它的唤醒源。系统上电或执行清除看门狗的指令，会清零 PDF；执行 HALT 指令，PDF 将被置位。看门狗计数器溢出将会置位 TO 标志并唤醒系统，这种复位会重置程序计数器和堆栈指针，其它标志保持原有状态。

PA 口中的每个引脚都可以通过 PAWU 寄存器使能下降沿唤醒功能。PA 端口唤醒后，程序将在“HALT”指令后继续执行。如果系统是通过中断唤醒，则有两种可能发生。第一种情况是：相关中断除能或是中断使能且堆栈已满，则程序会在“HALT”指令之后继续执行。这种情况下，唤醒系统的中断会等到相关中断使能或有堆栈层可以使用之后才执行。第二种情况是：相关中断使能且堆栈未滿，则中断可以马上执行。如果在进入休眠或空闲模式之前中断标志位已经被设置为“1”，则相关中断的唤醒功能将无效。

系统振荡器	唤醒时间 (休眠模式 0)	唤醒时间 (休眠模式 1)	唤醒时间 (空闲模式 0)	唤醒时间 (空闲模式 1)
HIRC	15~16 HIRC 周期		1~2 HIRC 周期	
LIRC	1~2 LIRC 周期		1~2 LIRC 周期	

唤醒时间

2.7.7 编程注意事项

高速和低速振荡器都使用 SST 计数器。例如，如果系统从休眠模式下唤醒，HIRC 振荡器起振需要一定的延迟时间。如果单片机从休眠模式唤醒到正常模式，则高速振荡器需要 SST 的系统延迟。

HTO 为高后，单片机会执行第一条指令。



2.8 看门狗定时器

看门狗定时器的功能在于防止如电磁的干扰等外部不可控制事件，所造成的程序不正常动作或跳转到未知的地址。

2.8.1 看门狗定时器时钟源

WDT 定时器时钟源来自于内部时钟 f_S ，而 f_S 时钟源来自内部 LIRC 振荡器。看门狗定时器的时钟源可分频为 $2^8 \sim 2^{18}$ 以提供更大的溢出周期，分频比由 WDT 寄存器中的 WS2~WS0 位来决定。电压为 5V 时内部振荡器 LIRC 的频率大约为 32kHz。

需要注意的是，这个特殊的内部时钟周期随 V_{DD} 、温度和制成的不同而变化。

2.8.2 看门狗定时器控制寄存器

WDT 寄存器用于控制 WDT 功能的使能/除能及选择溢出周期。寄存器结合配置选项控制看门狗定时器的操作。

• WDT 寄存器

Bit	7	6	5	4	3	2	1	0
Name	WE4	WE3	WE2	WE1	WE0	WS2	WS1	WS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	1	0	1	0	0	1	1

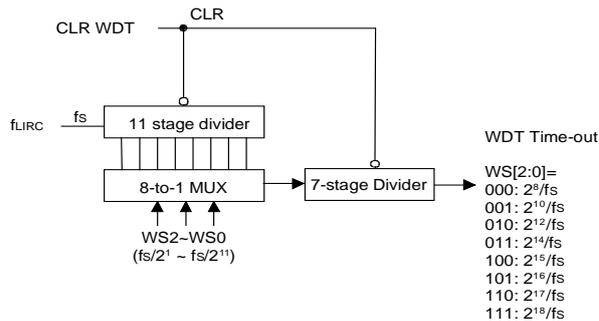
Bit 7~3 **WE4~WE0:** WDT 软件控制位
 10101B: 除能
 01010B: 使能 (默认值)
 其它: MCU 复位 (2~3 个 LIRC 时钟响应时间后复位才有效)

Bit 2~0 **WS2 ~ WS0:** 选择看门狗溢出周期
 000: $2^8/f_S$
 001: $2^{10}/f_S$
 010: $2^{12}/f_S$
 011: $2^{14}/f_S$ (默认值)
 100: $2^{15}/f_S$
 101: $2^{16}/f_S$
 110: $2^{17}/f_S$
 111: $2^{18}/f_S$
 这三位控制看门狗时钟的分频比，进而控制看门狗的溢出周期

2.8.3 看门狗定时器操作

当 WDT 溢出时，它产生一个芯片复位的动作。这也就意味着正常工作期间，用户需在应用程序中看门狗定时器溢出前将看门狗定时器清零以防止其产生复位，可使用清看门狗指令实现。

程序正常运行时，WDT 溢出将导致芯片复位，并置位状态标志位 TO。若系统处于休眠或空闲模式，当 WDT 发生溢出时，状态寄存器中的 TO 标志位会被置位，且只有程序计数器 PC 和堆栈指针 SP 会被复位。有三种方法可以用来清除 WDT 的内容。第一种是外部硬件复位，第二种是通过软件清除指令，而第三种是通过“HALT”指令。该单片机使用单一“CLR WDT”指令来清除 WDT。



看门狗定时器

2.8.4 复位和初始化

复位功能是整个单片机中基本的部分，使得单片机可以设定一些与外部参数无关的先置条件。最重要的复位条件是在单片机首次上电以后，经过短暂的延迟，内部硬件电路使得单片机处于预期的稳定状态并开始执行第一条程序指令。上电复位以后，在程序执行之前，部分重要的内部寄存器将会被设定为预先设定的状态。程序计数器就是其中之一，它会被清除为零，使得单片机从最低的程序存储器地址开始执行程序。

另一种复位为看门狗溢出单片机复位，不同方式的复位操作会对寄存器产生不同的影响。还有一种复位为低电压复位即 LVR 复位，在电源供应电压低于 LVR 设定值时，系统会产生 LVR 复位。

复位功能

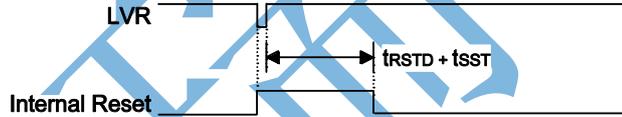
包括内部和外部事件触发复位，单片机共有以下几种复位方式：

• 上电复位

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

• 低电压复位-LVR

单片机具有低电压复位电路，用来监测它的电源电压，可通过配置选项进行选择。例如在更换电池的情况下，单片机供应的电压可能会落在 $0.9V \sim V_{LVR}$ 的范围内，这时 LVR 将会自动复位单片机。LVR 包含以下的规格：有效的 LVR 信号，即在 $0.9V \sim V_{LVR}$ 的低电压状态的时间，必须超过交流电气特性中 t_{LVR} 参数的值。如果低电压存在不超过 t_{LVR} 参数的值，则 LVR 将会忽略它且不会执行复位功能。 V_{LVR} 参数值可通过配置选项进行设定。



注： t_{RSTD} 为上电延迟时间，典型值为50ms。

低电压复位时序图

• LVRC Register

Bit	7	6	5	4	3	2	1	0
Name	LVS7	LVS6	LVS5	LVS4	LVS3	LVS2	LVS1	LVS0
R/W								
POR	0	1	0	1	0	1	0	1

Bit 7~0 **LVS[7:0]: LVR 电压选择位**
 01010101B: 2.1V
 00110011B: 2.55V
 10011001B: 3.15V
 10101010B: 3.8V

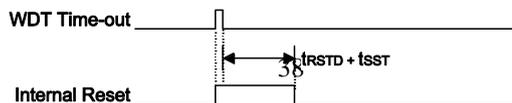
其它位: MCU 复位 (需要 2~3 个 LIRC 周期响应复位)

注: 可以通过 S/W 写 00H-FFH 来控制 LVR 电压, 也可以复位单片机, 如果单片机复位且由 LVRC 寄存器引起, 则在复位后 CTRL 寄存器中的 LRF 标志位会被置位。

除了 LVR 复位, 其他任何复位后 LVRC 都为 01010101, 且在 LVR 复位或者暂停模式下的 WDT。

• 正常运行时看门狗溢出复位

除了看门狗溢出标志位 TO 将被设为 1 之外, 正常运行时看门狗溢出复位和上电复位相同。

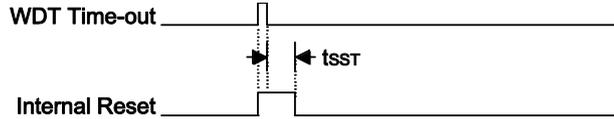


注: t_{RSTD} 为上电延迟时间, 典型值为50ms。

正常运行时看门狗溢出时序图

- 空闲/休眠时看门狗溢出复位

空闲/休眠时看门狗溢出复位和其它种类的复位有些不同, 除了程序计数器与堆栈指针将被清 0 及 TO 位被设为 1 外, 绝大部分的条件保持不变。图中 t_{SST} 的详细说明请参考交流电气特性。



注: 如果系统时钟源为HIRC时, t_{SST} 为15~16个时钟周期。
如果系统时钟源为LIRC, 则 t_{SST} 为1~2个时钟周期。

空闲/休眠时看门狗溢出复位时序图

复位初始状态

不同的复位形式以不同的途径影响复位标志位。这些标志位, 即 PDF 和 TO 位存放在状态寄存器中, 由空闲/休眠功能或看门狗计数器等几种控制器操作控制。复位标志位如下所示:

TO	PDF	复位条件
0	0	上电复位
u	u	正常模式或低速模式时的 LVR 复位
1	u	正常模式或低速模式时的 WDT 溢出复位
1	1	空闲模式或休眠模式时的 WDT 溢出复位

注: “u”代表不改变

在单片机上电复位之后, 各功能单元初始化的情形, 列于下表。

项目	复位后情况
程序计数器	清除为零
中断	所有中断被除能
看门狗定时器	WDT 清除并重新计时
定时/计数器	定时/计数器停止
输入/输出口	所有 I/O 设为输入模式
堆栈指针	堆栈指针指向堆栈顶端

不同的复位形式对单片机内部寄存器的影响是不同的。为保证复位后程序能正常执行, 了解寄存器在特定条件复位后的设置是非常重要的, 下表即为不同方式复位后内部寄存器的状况。注意若芯片有多种封装类型, 表格反应较大的封装的情况。

寄存器	上电复位	低电压复位	WDT 溢出 (正常模式)	WDT 溢出 (空闲或休眠模式)
IAR0	0000 0000	0000 0000	0000 0000	uuuu uuuu
MP0	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
IAR1	0000 0000	0000 0000	0000 0000	uuuu uuuu
MP1	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu

寄存器	上电复位	低电压复位	WDT 溢出 (正常模式)	WDT 溢出 (空闲或 休眠模式)
BP	---- --00	---- --00	---- --00	---- --uu
ACC	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
PCL	0000 0000	0000 0000	0000 0000	0000 0000
TBLP	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLH	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBHP	---- xxxx	---- uuuu	---- uuuu	---- uuuu
STATUS	--00 xxxx	--uu uuuu	--1u uuuu	--11 uuuu
SMOD	000- 0011	000- 0011	000- 0011	uuu- uuuu
LVDC	--00 -000	--00 -000	--00 -000	--uu -uuu
INTEG	---- 0000	---- 0000	---- 0000	---- uuuu
WDTC	0101 0011	0101 0011	0101 0011	uuuu uuuu
TBC	0011 -111	0011 -111	0011 -111	uuuu -uuu
INTC0	-000 0000	-000 0000	-000 0000	-uuu uuuu
INTC1	-000 -000	-000 -000	-000 -000	-uuu -uuu
INTC2	0000 0000	0000 0000	0000 0000	uuuu uuuu
MFI4	--00 --00	--00 --00	--00 --00	--uu --uu
MFI0	--00 --00	--00 --00	--00 --00	--uu --uu
MFI1	--00 --00	--00 --00	--00 --00	--uu --uu
MFI2	-000 -000	-000 -000	-000 -000	-uuu -uuu
MFI3	00-0 00-0	00-0 00-0	00-0 00-0	uu-u uu-u
PAWU	0-0 0000	0-0 0000	0-0 0000	u--u uuuu
PAPU	0-0 0000	0-0 0000	0-0 0000	u--u uuuu
PA	1--1 1111	1--1 1111	1--1 1111	u--u uuuu
PAC	1--1 1111	1--1 1111	1--1 1111	u--u uuuu
PBPU	0000 0000	0000 0000	0000 0000	uuuu uuuu
PB	1111 1111	1111 1111	1111 1111	uuuu uuuu
PBC	1111 1111	1111 1111	1111 1111	uuuu uuuu
PCPU	0000 0000	0000 0000	0000 0000	uuuu uuuu
PC	1111 1111	1111 1111	1111 1111	uuuu uuuu
PCC	1111 1111	1111 1111	1111 1111	uuuu uuuu
PDPU	---- 0000	---- 0000	---- 0000	---- uuuu
PD	---- 1111	---- 1111	---- 1111	---- uuuu
PDC	---- 1111	---- 1111	---- 1111	---- uuuu
CPC0	0000 0000	0000 0000	0000 0000	uuuu uuuu
CPC1	0001 0000	0001 0000	0001 0000	uuuu uuuu
OPAC0	0000 0000	0000 0000	0000 0000	uuuu uuuu
OPAC1	-001 0000	-001 0000	-001 0000	-uuu uuuu
OPAC2	0000 0-00	0000 0-00	0000 0-00	uuuu u-uu
ADRL(ADRF5=0)	xxxx ----	xxxx ----	xxxx ----	uuuu ----

寄存器	上电复位	低电压复位	WDT 溢出 (正常模式)	WDT 溢出 (空闲或 休眠模式)
ADRL(ADRFS=1)	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
ADRH(ADRFS=1)	---- xxxx	---- xxxx	---- xxxx	---- uuuu
ADRH(ADRFS=0)	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
ADCR0	0110 0000	0110 0000	0110 0000	uuuu uuuu
ADCR1	00-0 -000	00-0 -000	00-0 -000	uu-u -uuu
ACERL	--11 1111	--11 1111	--11 1111	--uu uuuu
DAC0	0000 0000	0000 0000	0000 0000	uuuu uuuu
DAC1	0000 0000	0000 0000	0000 0000	uuuu uuuu
DAC2	---- 0000	---- 0000	---- 0000	---- uuuu
SLEDC	0000 0000	0000 0000	0000 0000	uuuu uuuu
LVRC	0101 0101	0101 0101	0101 0101	uuuu uuuu
TM0C0	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM0C1	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM0DL	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM0DH	--- -00	--- -00	--- -00	--- -uu
TM0AL	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM0AH	---- -00	---- -00	---- -00	---- -uu
EEA	--00 0000	--00 0000	--00 0000	--uu uuuu
EED	0000 0000	0000 0000	0000 0000	uuuu uuuu
TMPC0	1001 --01	1001 --01	1001 --01	uuuu --uu
TMPC1	---- --01	---- --01	---- --01	---- -uu
TM1C0	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1C1	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1C2	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1DL	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1DH	---- -00	---- -00	---- -00	---- -uu
TM1AL	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1AH	---- -00	---- -00	---- -00	---- -uu
TM1BL	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1BH	---- -00	---- -00	---- -00	---- -uu
TM2C0	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM2C1	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM2DL	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM2DH	---- -00	---- -00	---- -00	---- -uu
TM2AL	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM2AH	---- -00	---- -00	---- -00	---- -uu
CTRL	0-00 -x00	0-00 - 100	0-00 - u00	u-uu - uuu
SLCDC0	0000 0000	0000 0000	0000 0000	uuuu uuuu
SLCDC1	0000 0000	0000 0000	0000 0000	uuuu uuuu

寄存器	上电复位	低电压复位	WDT 溢出 (正常模式)	WDT 溢出 (空闲或 休眠模式)
SLCDC2	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKC	0000 0---	0000 0---	0000 0---	uuuu u---
TKM016DH	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM016DL	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM0C0	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM0C1	---0 0000	---0 0000	---0 0000	---u uuuu
TKM116DH	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM116DL	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM1C0	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM1C1	---0 0000	---0 0000	---0 0000	---u uuuu
TKM216DH	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM216DL	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM2C0	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM2C1	---0 0000	---0 0000	---0 0000	---u uuuu
TKM316DH	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM316DL	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM3C0	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM3C1	---0 0000	---0 0000	---0 0000	---u uuuu
EEC	---- 0000	---- 0000	---- 0000	---- uuuu

注： “-”表示未定义
 “x”表示未知
 “u”表示不改变

2.9 输入/输出端口

格瑞达单片机的输入/输出控制具有很大的灵活性。大部分引脚都可在用户程序控制下被设定为输入或输出，所有引脚的上拉电阻设置以及指定引脚的唤醒设置也都由软件控制，这些特性也使得此类单片机在广泛应用上都能符合开发的需求。

此单片机提供 PA~PD 双向输入/输出口。这些寄存器在数据存储寄存器有特定的地址。所有 I/O 口用于输入输出操作。作为输入操作，输入引脚无锁存功能，也就是说输入数据必须在执行“MOV A, [m]”，T2 的上升沿准备好，m 为端口地址。对于输出操作，所有数据都是被锁存的，且保持不变直到输出锁存被重写。

2.9.1 输入/输出寄存器列表

寄存器名称	Bit							
	7	6	5	4	3	2	1	0
PAWU	PAWU7	—	—	PAWU4	PAWU3	PAWU2	PAWU1	PAWU0
PAPU	PAPU7	—	—	PAPU4	PAPU3	PAPU2	PAPU1	PAPU0
PA	PA7	—	—	PA4	PA3	PA2	PA1	PA0
PAC	PAC7	—	—	PAC4	PAC3	PAC2	PAC1	PAC0
PBPU	PBPU7	PBPU6	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0
PB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
PBC	PBC7	PBC6	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
PCPU	PCPU7	PCPU6	PCPU5	PCPU4	PCPU3	PCPU2	PCPU1	PCPU0
PC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
PCC	PCC7	PCC6	PCC5	PCC4	PCC3	PCC2	PCC1	PCC0
PDPU	—	—	—	—	PDPU3	PDPU2	PDPU1	PDPU0
PD	—	—	—	—	PD3	PD2	PD1	PD0
PDC	—	—	—	—	PDC3	PDC2	PDC1	PDC0

2.9.2 上拉电阻

许多产品应用在端口处于输入状态时需要外加一个上拉电阻来实现上拉的功能。为了免去外部上拉电阻，当引脚规划为输入时，可由内部连接到一个上拉电阻，这些上拉电阻可通过寄存器 PAPU~PDPU 来设置，它用一个 PMOS 晶体管来实现上拉电阻功能。

PAPU 寄存器

Bit	7	6	5	4	3	2	1	0
Name	PA7	—	—	PA4	PA3	PA2	PA1	PA0
R/W	R/W	—	—	R/W	R/W	R/W	R/W	R/W
POR	1	—	—	1	1	1	1	1

Bit 7,4-0 **PAPU:** PA 口 bit7,bit 4~bit 0 上拉电阻控制

0: 除能

1: 使能

Bit 6~5

未使用，读为“0”

PBPU 寄存器

Bit	7	6	5	4	3	2	1	0
Name	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

Bit 7~0 **PBPU:** PB 口 bit 7~bit 0 上拉电阻控制

0: 除能

1: 使能

PCPU 寄存器

Bit	7	6	5	4	3	2	1	0
Name	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

POR	1	1	1	1	1	1	1	1
-----	---	---	---	---	---	---	---	---

Bit 7~0 **PCPU**: PC 口 bit 7~bit 0 上拉电阻控制
 0: 除能
 1: 使能

PDPU 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	PD3	PD2	PD1	PD0
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	1	1	1	1

Bit 7~4 未使用, 读为“0”
 Bit 3~0 **PDPU**: PD 口 bit 3~bit 0 上拉电阻控制
 0: 除能
 1: 使能

2.9.3 PA 口唤醒

当使用暂停指令“HALT”迫使单片机进入空闲/休眠模式状态, 单片机的系统时钟将会停止以降低功耗, 此功能对于电池及低功耗应用很重要。唤醒单片机有很多种方法, 其中之一就是使 PA 口的其中一个引脚从高电平转为低电平。这个功能特别适合于通过外部开关来唤醒的应用。PA 口上的每个引脚是可以设置 PAWU 寄存器来单独选择是否具有唤醒功能。

• PAWU 寄存器

Bit	7	6	5	4	3	2	1	0
Name	PAWU7	—	—	PAWU4	PAWU3	PAWU2	PAWU1	PAWU0
R/W	R/W	—	—	R/W	R/W	R/W	R/W	R/W
POR	0	—	—	0	0	0	0	0

Bit 7,4~0 **PAWU**: PA 口 bit 7,bit 4~bit 0 唤醒控制
 0: 除能
 1: 使能
 Bit 6~5 未使用, 读为“0”

2.9.4 输入/输出端口控制寄存器

每一个输入/输出都具有各自的控制寄存器(PAC~PDC)用来控制输入/输出状态。通过这些控制寄存器, 每个 CMOS 输出或输入都可以通过软件动态控制。所有的 I/O 端口的引脚都各自对应于 I/O 端口控制寄存器的某一位。若 I/O 引脚要实现输入功能, 则对应的控制寄存器的位需要设置为“1”, 这时程序指令可以直接读取输入脚的逻辑状态。若控制寄存器相应的位被设定为“0”, 则此引脚被设置为 CMOS 输出。当引脚设置为输出状态时, 程序指令读取的是输出端口寄存器的内容。注意, 如果对输出口做读取动作时, 程序读取到的是内部输出数据锁存器中的状态, 而不是输出引脚上实际的逻辑状态。

PAC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	PAC7	—	—	PAC4	PAC3	PAC2	PAC1	PAC0
R/W	R/W	—	—	R/W	R/W	R/W	R/W	R/W
POR	1	—	—	1	1	1	1	1

Bit 7,4~0 **PAC**: PA 口 bit 7,bit 4~bit 0 输入/输出控制
 0: 输出
 1: 输入
 Bit 6~5 未使用, 读为“0”

PBC 寄存器

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

Name	PBC7	PBC6	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
R/W								
POR	1	1	1	1	1	1	1	1

Bit 7~0 **PBC:** PB 口 bit 7~bit 0 输入/输出控制
 0: 输出
 1: 输入

PCC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	PCC7	PCC6	PCC5	PCC4	PCC3	PCC2	PCC1	PCC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

Bit 7~0 **PCPU:** PC 口 bit 7~bit 0 输入/输出控制
 0: 输出
 1: 输入

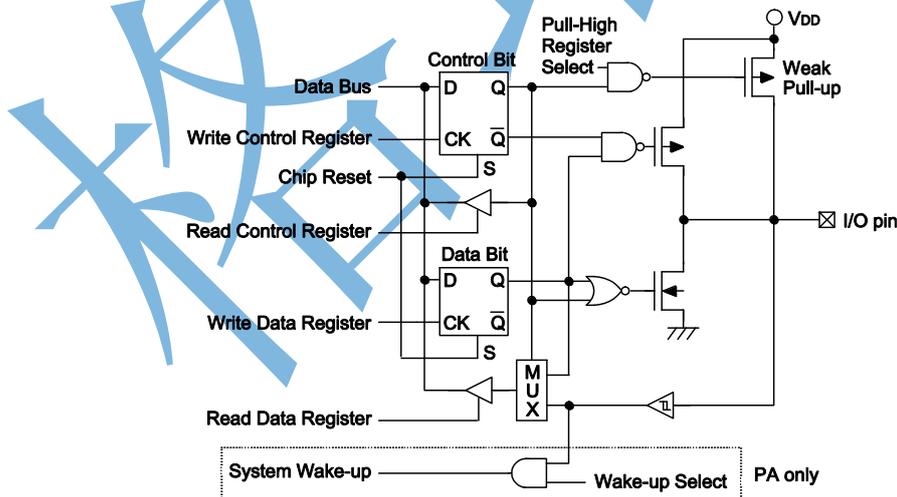
PDC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	PDC3	PDC2	PDC1	PDC0
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	1	1	1	1

Bit 7~4 未使用，读为“0”
 Bit 3~0 **PDC:** PD 口 bit 3~bit 0 输入/输出控制
 0: 输出
 1: 输入

2.9.5 输入/输出引脚结构

下图为输入/输出引脚的内部结构图。输入/输出引脚的准确逻辑结构图可能与此图不同，这里只是为了方便对功能的理解提供的一个参考。图中引脚共用并非针对所有单片机。



通用输入/输出端口结构

2.9.6 编程注意事项

在编程中，最先要考虑的是端口的初始化。复位之后，所有的输入/输出数据及端口控制寄存器都将被设为逻辑高。所有输入/输出引脚默认为输入状态，而其电平则取决于其它相连接电路以及是否选择了上拉电阻。如果端口控制寄存器 PAC~PDC，某些引脚位被设定输出状态，这些输出引脚会有初始高电平输出，除非数据寄存器端口 PA~PD 在程序中被预先设定。设置哪些引脚是输入及哪些引脚是输出，可通过设置正确的值到适当的端口控制寄存器，或使用指令“SET [m].i”及“CLR [m].i”

来设定端口控制寄存器中个别的位。注意，当使用这些位控制指令时，系统即将产生一个读-修改-写的操作。单片机需要先读入整个端口上的数据，修改个别的位，然后重新把这些数据写入到输出端口。

PA 口的每个引脚都带唤醒功能。单片机处于休眠或空闲模式时，有很多方法可以唤醒单片机，其中之一就是通过 PA 口任一引脚电平从高到低转换的方式，可以设置 PA 口一个或多个引脚具有唤醒功能。



2.10 定时器模块 - TM

控制和测量时间在任何单片机中都是一个很重要的部分。每个单片机提供几个定时器模块（简称 TM），来实现和时间有关的功能。定时器模块是包括多种操作的定时单元，提供的操作有：定时/事件计数器，捕捉输入，比较匹配输出，单脉冲输出以及 PWM 输出等功能。每个定时器模块有两个或三个独立中断。每个 TM 外加的输入输出引脚，扩大了定时器的灵活性，便于用户使用。

这里只介绍各种 TM 的共性，更多详细资料请参考简易型，增强型定时器章节。

2.10.1 简介

该单片机包含 3 个 TM。每个 TM 可被划分为一个特定的类型，即两个简易型 TM 和一个增强型 TM。虽然性质相似，但不同 TM 特性复杂度不同。本章介绍简易型和增强型 TM 的共性，更多详细资料分别见后面各章。二种类型 TM 的特性和区别见下表。

功能	CTM	ETM
定时/计数器	√	√
捕捉输入	—	√
比较匹配输出	√	√
PWM 通道数	1	2
单脉冲输出	—	1
PWM 对齐方式	边沿对齐	边沿&中心对齐
PWM 调节周期&占空比	占空比或周期	占空比或周期

TM 功能概要

该单片机包括一定数目的定时器单元，其中有两个简易型和一个增强型 TM，见下表（此单片机没有标准型 TM）。

TM0	TM1	TM2
10-bit CTM	10-bit ETM	10-bit CTM（用于触摸按键）

TM 名称/类型参考

2.10.2 TM 操作

不同类型的 TM 提供从简单的定时操作到 PWM 信号产生等多种功能。理解 TM 操作的关键是比较 TM 内独立运行的计数器的值与内部比较器的预置值。当计数器的值与比较器的预置值相同时，则比较匹配，TM 中断信号产生，清零计数器并改变 TM 输出引脚的状态。用户选择内部时钟或外部引脚来驱动内部 TM 计数器。

2.10.3 TM 时钟源

驱动 TM 计数器的时钟源很多。通过设置 TM 控制寄存器的 TnCK2~TnCK0 位，选择所需的时钟源。该时钟源来自系统时钟 f_{SYS} 或内部高速时钟 f_H 或 f_{TBC} 时钟源或外部 TCKn 引脚时钟的分频比。注意：设置 TnCK2~TnCK0 为 101，将选择 TM 预设时钟输入，有效切断 TM 时钟源。TCKn 引脚时钟源用于允许外部信号作为 TM 时钟源或用于事件计数。

2.10.4 TM 中断

简易型 TM 拥有两个内部中断，分别是内部比较器 A 或比较器 P，当比较匹配发生时产生 TM 中断。增强型 TM 有三个内部比较器，即比较器 A 或比较器 B 或比较器 P，相应的有三个内部中断。当 TM 中断产生时，计数器清零并改变 TM 输出引脚的状态。

2.10.5 TM 外部引脚

无论哪种类型的 TM，都有一个 TM 输入引脚 TCKn。通过设置 TMnCO 寄存器中的 TnCK2~TnCK0 位，选择 TM 功能并将该引脚作为 TM 时钟源输入脚。外部时钟源可通过该引脚来驱动内部 TM。外部 TM 输入脚也与其它功能共用，但是，如果设置适当值给 TnCK2~TnCK0，该引脚会连接到内

部 TM。TM 引脚可选择上升沿有效或下降沿有效。

每个 TM 有一个或多个输出引脚 TPn。当 TM 工作在比较匹配输出模式且比较匹配发生时，这些引脚会由 TM 控制切换到高电平或低电平或翻转。外部 TPn 输出引脚也被 TM 用来产生 PWM 输出波形。当 TM 输出引脚与其它功能共用时，TM 输出功能需要通过寄存器先被设置。寄存器中的一个单独位用于决定其相关引脚用于外部 TM 输出还是用于其它功能。每个单片机和不同类型 TM 中输出引脚的个数是不同的，详见下表。

所有 TM 输出引脚名称都以“_n”为后缀名。后缀是“_1”或“_2”的引脚名称表示它们是一个 TM 的多个输出引脚。此类 TM 通过输入/输出寄存器数据位可以产生互补输出。

CTM	ETM	寄存器
TPn_0,TPn_1	TPnA,TPnB_0,TPnB_1,TPnB_2	TMPC0,TMPC1

TM 输出引脚

2.10.6 TM 输入/输出引脚控制寄存器

通过设置一或两个与 TM 输入/输出引脚相关的寄存器的一位，选择作为 TM 输入/输出功能或其它共用功能。设定为高时，相关引脚用作 TM 输入/输出，清零时将保持原来的功能。

寄存器	位							
	7	6	5	4	3	2	1	0
TMPC0	T1ACP0	T1BCP2	T1BCP1	T1BCP0	—	—	T0CP1	T0CP0
TMPC1	—	—	—	—	—	—	T2CP1	T2CP0

TM 输入/输出引脚控制寄存器列表

• TMPC0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	T1ACP0	T1BCP2	T1BCP1	T1BCP0	—	—	T0CP1	T0CP0
R/W	R/W	R/W	R/W	R/W	—	—	R/W	R/W
POR	1	0	0	1	—	—	0	1

Bit 7 **T1ACP0**: TP1A 引脚控制位

0: 除能
1: 使能

Bit 6 **T1BCP2**: TP1B_2 引脚控制位

0: 除能
1: 使能

Bit 5 **T1BCP1**: TP1B_1 引脚控制位

0: 除能
1: 使能

Bit 4 **T1BCP0**: TP1B_0 引脚控制位

0: 除能
1: 使能

Bit 3,2 未定义，读为“0”

Bit 1 **T0CP1**: TP0_1 引脚控制位

0: 除能
1: 使能

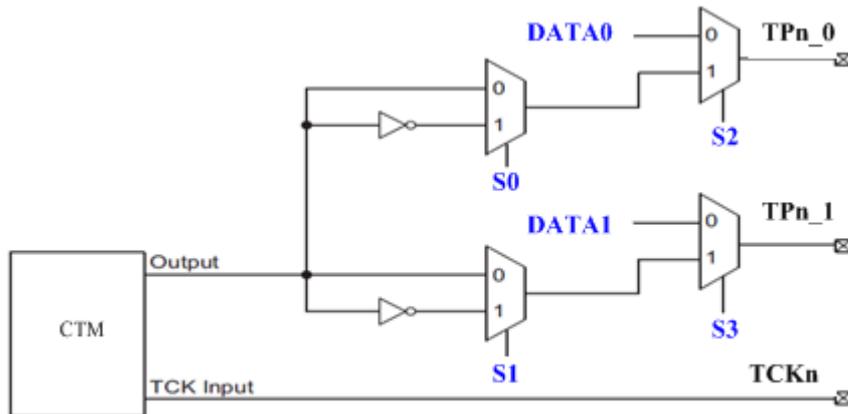
Bit 0 **T0CP0**: TP0_0 引脚控制位

0: 除能
1: 使能

• TMPC1 寄存器

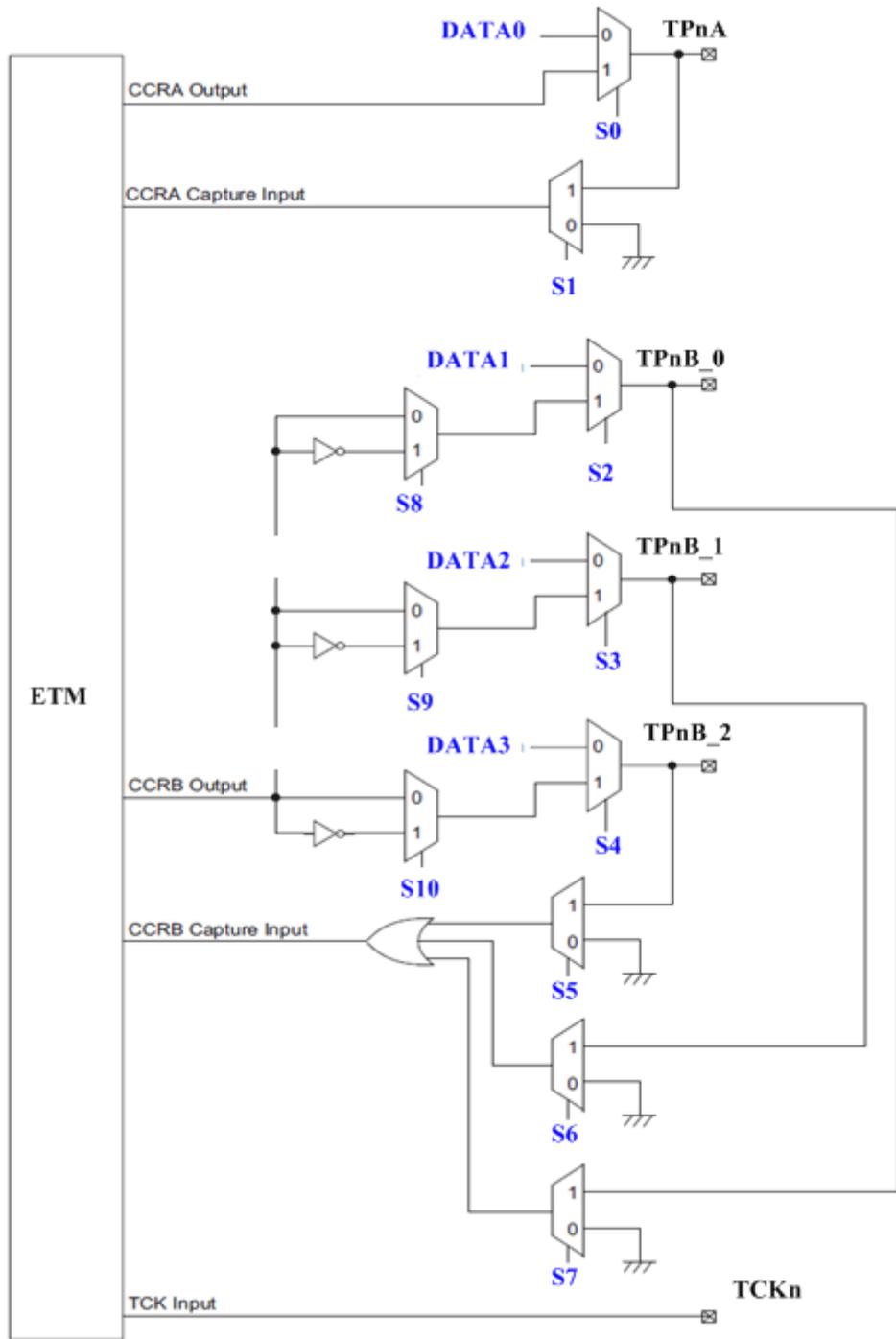
Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	T2CP1	T2CP0
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	1

- Bit 7~2 未定义，读为“0”
- Bit 1 **T2CP1**: TP2_1 引脚控制位
 - 0: 除能
 - 1: 使能
- Bit 0 **T2CP0**: TP2_0 引脚控制位
 - 0: 除能
 - 1: 使能



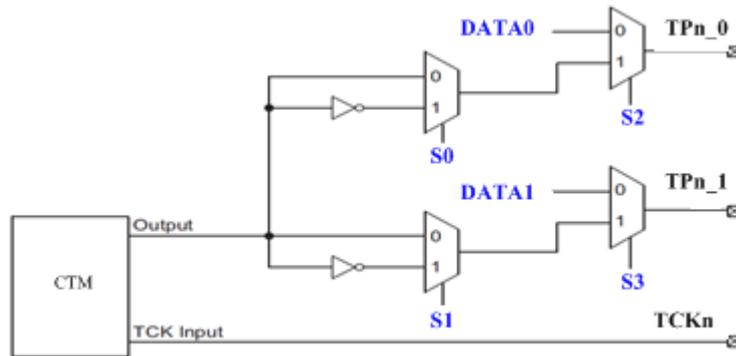
TM0 功能引脚控制方框图

- 注：1.S0=PB0,S1=PB4,S2=T0CP0,S3=T0CP1,DATA0=PB0,DATA1=PB4
 2.上图所示输入/输出寄存器数据位用于 TM 输出反相控制。
 3.捕捉输入模式中，TM 引脚控制寄存器最多只能使能一个 TM 引脚输入。



TM1 功能引脚控制方框图

- 注: 1. S0=T1ACP0, S1=T1ACP0, S2=T1BCP0, S3=T1BCP1, S4=T1BCP2, S5=T1BCP2, S6=T1BCP1, S7=T1BCP0, S8=PB3, S9=PB5, S10=PB7, DATA0=PB1, DATA1=PB3, DATA2=PB5, DATA3=PB7
2. 上图所示输入/输出寄存器数据位用于 TM 输出反相控制。
3. 捕捉输入模式中, TM 引脚控制寄存器最多只能使能一个 TM 引脚输入。



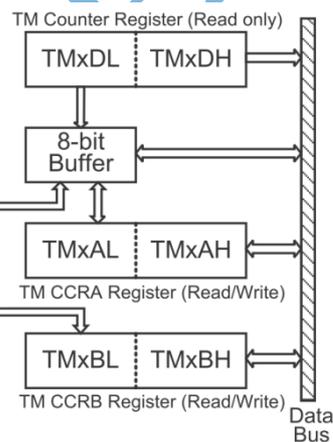
TM2 功能引脚控制方框图

注：1. S0=PB2, S1=PB6, S2=T2CP0, S3=T2CP1, DATA0=PB2, DATA1=PB6

1. 上图所示输入/输出寄存器数据位用于 TM 输出反相控制。
2. 捕捉输入模式中，TM 引脚控制寄存器最多只能使能一个 TM 引脚输入。

2.10.7 编程注意事项

TM 计数寄存器和捕捉/比较寄存器 CCRA、CCRB 为 10-bit 的寄存器，含有低字节和高字节结构。高字节可直接访问，低字节则仅能通过一个内部 8-bit 的缓存器进行访问。读写这些成对的寄存器需通过特殊的方式。值得注意的是 8-bit 缓存器的存取数据及相关低字节的读写操作仅在其相应的高字节读取操作执行时发生。



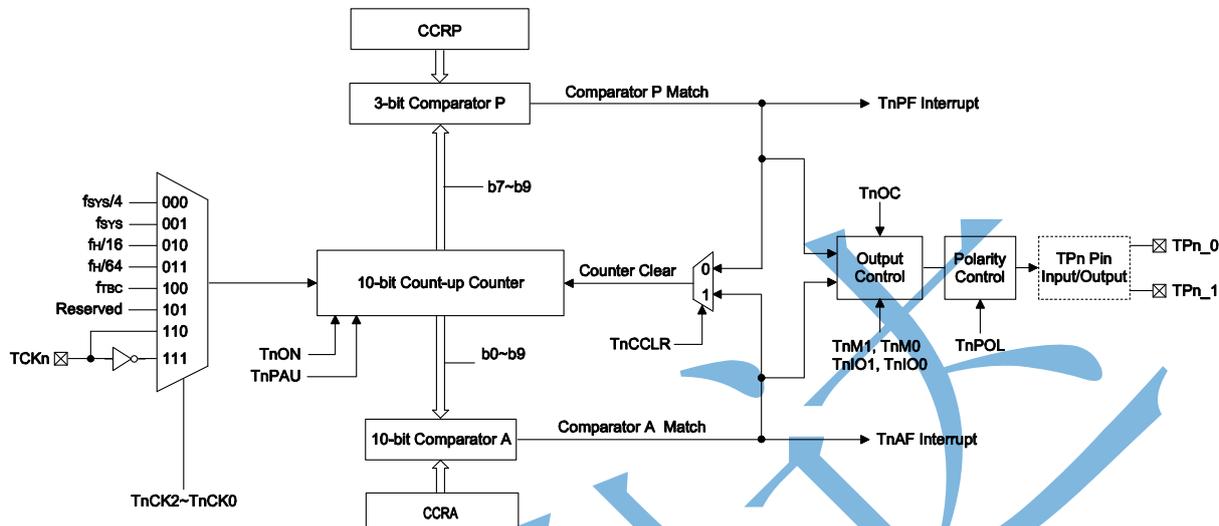
CCRA、CCRB 寄存器的应用如图所示。如上所述，读写这些成对的寄存器需通过特殊的方式。使用“MOV”指令并按照下面的流程来读写 CCRA 和 CCRB 低字节寄存器，即：TMxAL 寄存器和 TMxBL 寄存器。否则，将会产生不可预测的值。

读写流程如下步骤所示：

- 写数据至 CCRB 或 CCRA
 - ◆ 步骤 1. 写数据至低字节寄存器 TMxAL 或 TMxBL
 - 注意，此时数据仅写入 8-bit 缓存器。
 - ◆ 步骤 2. 写数据至高字节寄存器 TMxAH 或 TMxBH
 - 注意，此时数据直接写入高字节寄存器，同时锁存在 8-bit 缓存器中的数据写入低字节寄存器。
- 由计数器寄存器和 CCRB 或 CCRA 中读取数据
 - ◆ 步骤 1. 由高字节寄存器 TMxDH, TMxAH 或 TMxBH 读取数据
 - 注意，此时高字节寄存器中的数据直接读取，同时由低字节寄存器读取的数据锁存至 8-bit 缓存器中。
 - ◆ 步骤 2. 由低字节寄存器 TMxDL、TMxAL 或 TMxBL 读取数据
 - 注意，此时读取 8-bit 缓存器中的数据。

2.11 简易型 TM - CTM (TMO 和 TM2)

虽然简易型 TM 是三种 TM 类型中最简单的形式,但仍然包括三种工作模式,即比较匹配输出,定时/事件计数器和 PWM 输出模式。简易型 TM 也由外部输入脚控制并驱动一或两个外部输出脚。两个外部输出脚信号可以相同也可以相反。



简易型 TM 方框图

2.11.1 简易型 TM 操作

简易型 TM 核心是一个由用户选择的内部或外部时钟源驱动的 10 位向上计数器,它还包括两个内部比较器即比较器 A 和比较器 P。这两个比较器将计数器的值与 CCRP 和 CCRA 寄存器中的值进行比较。CCRP 是 3 位的,与计数器的高 3 位比较;而 CCRA 是 10 位的,与计数器的所有位比较。

通过应用程序改变 10 位计数器值的唯一方法是使 TnON 位发生上升沿跳变清除计数器。此外,计数器溢出或比较匹配也会自动清除计数器。上述条件发生时,通常情况会产生 TM 中断信号。简易型 TM 可工作在不同的模式,可由包括来自输入脚的不同时钟源驱动,也可以控制输出脚。所有工作模式的设定都是通过设置相关寄存器来实现的。

2.11.2 简易型 TM 寄存器介绍

简易型 TM 的所有操作由六个寄存器控制。包含一对只读寄存器用来存放 10 位计数器的值,一对读/写寄存器存放 10 位 CCRA 的值,剩下两个控制寄存器设置不同的操作和控制模式以及 CCRP 的 3 个位。

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TMnCO	TnPAU	TnCK2	TnCK1	TnCK0	TnON	TnRP2	TnRP1	TnRP0
TMnC1	TnM1	TnM0	TnIO1	TnIO0	TnOC	TnPOL	TnDPX	TnCCLR
TMnDL	D7	D6	D5	D4	D3	D2	D1	D0
TMnDH	—	—	—	—	—	—	D9	D8
TMnAL	D7	D6	D5	D4	D3	D2	D1	D0
TMnAH	—	—	—	—	—	—	D9	D8

简易型 TM 寄存器列表(n=0 或 2)

• TMnDL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~0 **TMnDL**: TMn 计数器低字节寄存器 bit 7~bit 0
TMn 10-bit 计数器 bit 7~bit 0

• TMnDH 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R	R
POR	—	—	—	—	—	—	0	0

Bit 7~2 未定义，读为“0”

 Bit 1~0 **TMnDH:** TMn 计数器高字节寄存器 bit 1~bit 0
 TMn 10-bit 计数器 bit 9~bit 8

• TMnAL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

 Bit 7~0 **TMnAL:** TMn CCRA 低字节寄存器 bit 7~bit 0
 TMn 10-bit CCRA bit 7~bit 0

• TMnAH 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 未定义，读为“0”

 Bit 1~0 **TMnAH:** TMn CCRA 高字节寄存器 bit 1~bit 0
 TMn 10-bit CCRA bit 9~bit 8

• TMnC0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	TnPAU	TnCK2	TnCK1	TnCK0	TnON	TnRP2	TnRP1	TnRP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

 Bit 7 **TnPAU:** TMn 计数器暂停控制位
 0: 运行
 1: 暂停

通过设置此位为高可使计数器暂停，清零此位恢复正常计数器操作。当处于暂停条件时，TM 保持上电状态并继续耗电。当此位由低到高转换时，计数器将保留其剩余值，直到此位再次改变为低电平，从此值开始继续计数。

 Bit 6~4 **TnCK2~TnCK0:** 选择 TMn 计数时钟位

 000: $f_{SYS}/4$
 001: f_{SYS}
 010: $f_H/16$
 011: $f_H/64$
 100: f_{TBC}
 101: 保留位
 110: TCKn 上升沿时钟
 111: TCKn 下降沿时钟

此三位用于选择 TM 的时钟源。选择保留时钟输入将有效地除能内部计数器。外部引脚时钟源能被选择在上沿或下降沿有效。 f_{SYS} 是系统时钟， f_H 和 f_{TBC} 是其它的内部时钟源，细节方面请参考振荡器章节。

 Bit 3 **TnON:** TMn 计数器 On/Off 控制位
 0: Off

1: On

此位控制 TM 的总开关功能。设置此位为高则使能计数器使其运行，清零此位则除能 TM。清零此位将停止计数器并关闭 TM 减少耗电。当此位经由低到高转换时，内部计数器将保持其剩余值。

若 TM 处于比较匹配输出模式时(通过 TnOC 位指定)，当 TnON 位经由低到高的转换时，TM 输出脚将重置其初始值。

Bit 2~0 TnRP2~TnRP0: TMn CCRP 3-bit 寄存器，对应于 TMn 计数器 bit 9~bit 7 比较器 P 匹配周期

- 000: 1024 个 TMn 时钟周期
- 001: 128 个 TMn 时钟周期
- 010: 256 个 TMn 时钟周期
- 011: 384 个 TMn 时钟周期
- 100: 512 个 TMn 时钟周期
- 101: 640 个 TMn 时钟周期
- 110: 768 个 TMn 时钟周期
- 111: 896 个 TMn 时钟周期

此三位设定内部 CCRP 3-bit 寄存器的值，然后与内部计数器的高三位进行比较。如果 TnCCLR 位设定为 0 时，比较结果为 0 并清除内部计数器。TnCCLR 位设为低，内部计数器在比较器 P 比较匹配发生时被重置；由于 CCRP 只与计数器高三位比较，比较结果是 128 时钟周期的倍数。CCRP 被清零时，实际上会使得计数器在最大值溢出。

• TMnC1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	TnM1	TnM0	TnIO1	TnIO0	TnOC	TnPOL	TnDPX	TnCCLR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 TnM1~TnM0: 选择 TMn 工作模式位

- 00: 比较匹配输出模式
- 01: 未定义模式
- 10: PWM 模式
- 11: 定时/计数器模式

这两位设置 TM 需要的工作模式。为了确保操作可靠，TM 应在 TnM1 和 TnM0 位有任何改变前先关掉。在定时/计数器模式，TM 输出脚控制必须除能。

Bit 5~4 TnIO1~TnIO0: 选择 TPn_0, TPn_1 输出功能位

比较匹配输出模式

- 00: 无变化
- 01: 输出低
- 10: 输出高
- 11: 输出翻转

PWM 模式

- 00: 强制无效状态
- 01: 强制有效状态
- 10: PWM 输出
- 11: 未定义

定时/计数器模式

未使用

此两位用于决定在一定条件达到时 TM 输出脚如何改变状态。这两位值的选择决定 TM 运行在哪种模式下。

在比较匹配输出模式下，TnIO1 和 TnIO0 位决定当比较器 A 比较匹配输出发生时 TM 输出脚如何改变状态。当比较器 A 比较匹配输出发生时 TM 输出脚能设为切换高、切换低或翻转当前状态。若此两位同时为 0 时，这个输出将不会改变。TM 输出脚的初始值通过 TMnC1 寄存器的 TnOC 位设置取得。注意，由 TnIO1 和 TnIO0 位得到的输出电平必须与通过 TnOC 位设置的初始值不同，否则当比较匹配发生时，TM 输出脚将不会发生变化。在 TM 输出脚改变状态后，通过 TnON 位由低到高电平的转换复位至初始值。

在 PWM 模式，TnIO1 和 TnIO0 用于决定比较匹配条件发生时怎样改变 TM 输出脚的状态。PWM 输出功能通过这两位的变化进行更新。仅在 TMn 关闭时改变 TnIO1 和 TnIO0 位的值是很有必要的。若在 TM 运行时改变 TnIO1 和 TnIO0 的值，PWM 输出的值是无法预料的。

Bit 3 TnOC: TPn_0, TPn_1 输出控制位

比较匹配输出模式

- 0: 初始低
- 1: 初始高

PWM 模式

- 0: 低有效
- 1: 高有效

这是 TM 输出脚输出控制位。它取决于 TM 此时正运行于比较匹配输出模式还是 PWM 模式。若 TM 处于定时/计数器模式，则其不受影响。在比较匹配输出模式时，比较匹配发生前其决定 TM 输出脚的逻辑电平值。在 PWM 模式时，其决定 PWM 信号是高有效还是低有效。

Bit 2 **TnPOL:** TPn_0, TPn_1 输出极性控制位

- 0: 同相
- 1: 反相

此位控制 TPn_0 或 TPn_1 输出脚的极性。此位为高时 TM 输出脚反相，为低时 TM 输出脚同相。若 TM 处于定时/计数器模式时其不受影响。

Bit 1 **TnDPX:** TMn PWM 周期/占空比控制位

- 0: CCRP - 周期; CCRA - 占空比
- 1: CCRP - 占空比; CCRA - 周期

此位决定 CCRA 与 CCRP 寄存器哪个被用于 PWM 波形的周期和占空比控制。

Bit 0 **TnCCLR:** 选择 TMn 计数器清零条件位

- 0: TM0 比较器 P 匹配
- 1: TM0 比较器 A 匹配

此位用于选择清除计数器的方法。简易型 TM 包括两个比较器-比较器 A 和比较器 P。这两个比较器每个都可以用作清除内部计数器。TnCCLR 位设为高，计数器在比较器 A 比较匹配发生时被清除；此位设为低，计数器在比较器 P 比较匹配发生或计数器溢出时被清除。计数器溢出清除的方法仅在 CCRP 被清除为 0 时才能生效。TnCCLR 位在 PWM 模式时未使用。

2.11.3 简易型 TM 工作模式

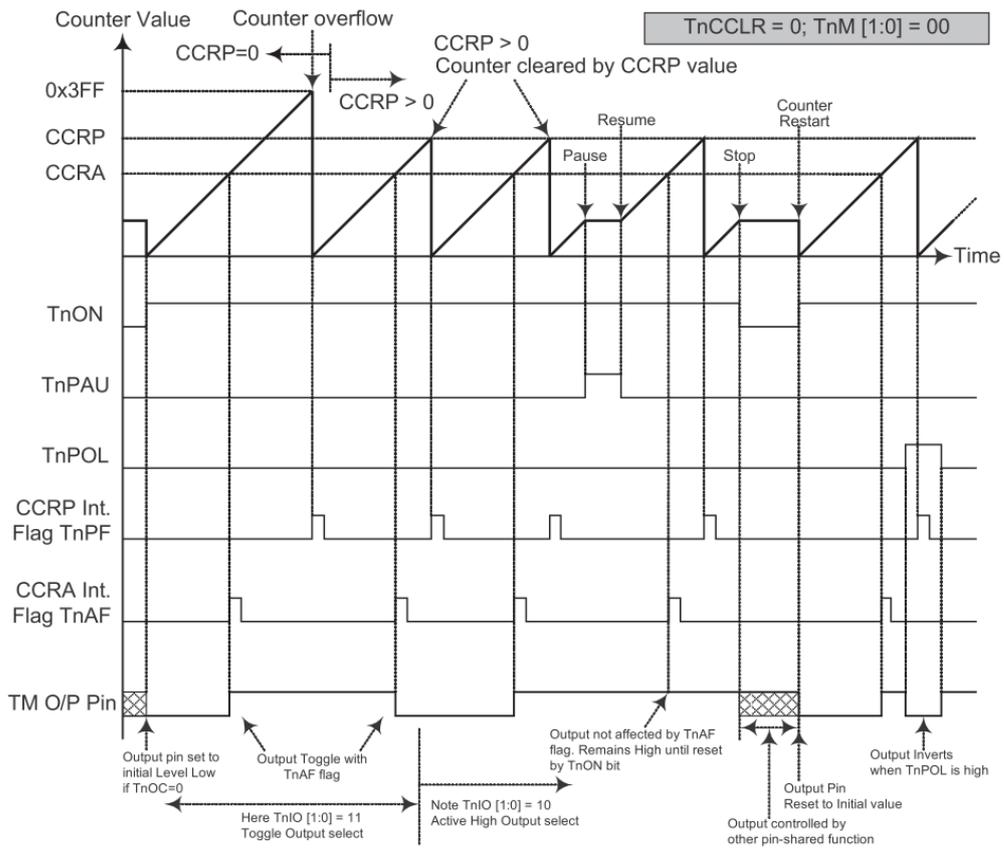
简易型 TM 有三种工作模式，即比较匹配输出模式，PWM 模式或定时/计数器模式。通过设置 TMnC1 寄存器的 TnM1 和 TnM0 位选择任意工作模式。

比较匹配输出模式

为使 TM 工作在此模式，TMnC1 寄存器中的 TnM1 和 TnM0 位需要设置为“00”。当工作在该模式，一旦计数器使能并开始计数，有三种方法来清零，分别是：计数器溢出，比较器 A 比较匹配发生和比较器 P 比较匹配发生。当 TnCCLR 位为低，有两种方法清除计数器。一种是比较器 P 比较匹配发生，另一种是 CCRP 所有位设置为零并使得计数器溢出。此时，比较器 A 和比较器 P 的请求标志位 TnAF 和 TnPF 将分别置起。

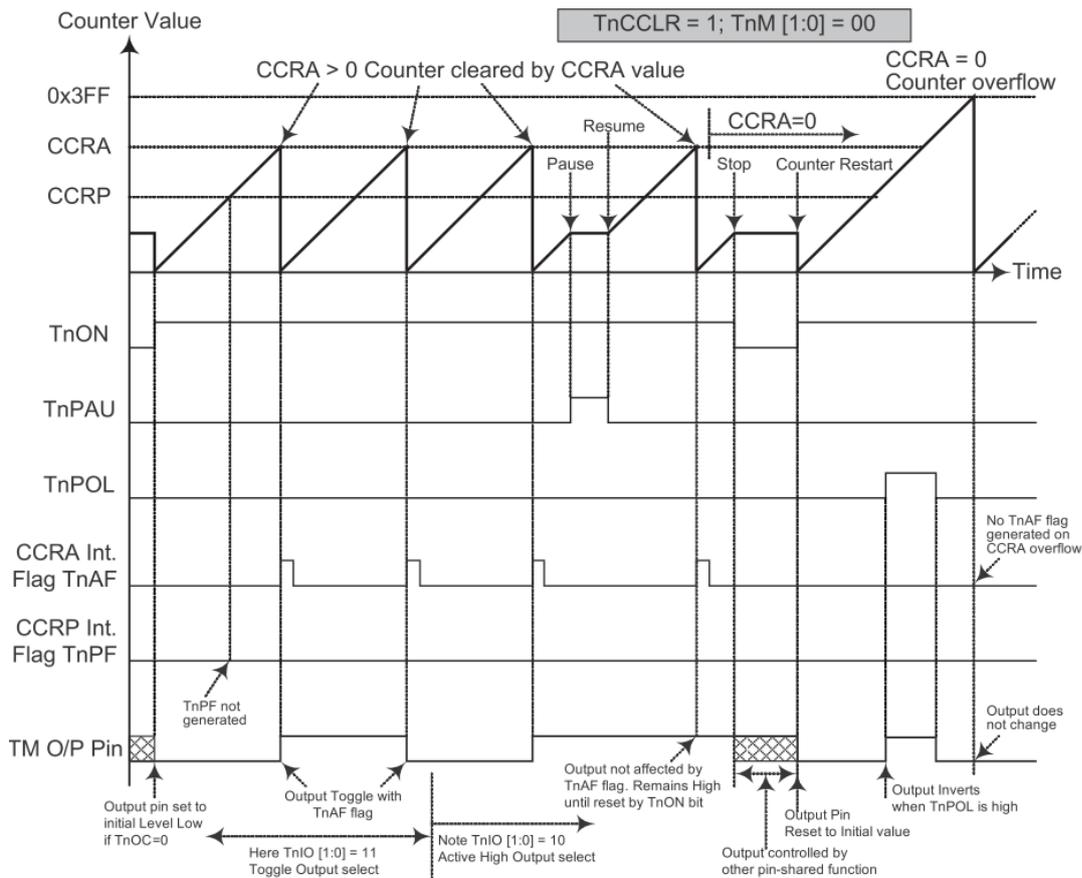
如果 TMnC1 寄存器的 TnCCLR 位设置为高，当比较器 A 比较匹配发生时计数器被清零。此时，即使 CCRP 寄存器的值小于 CCRA 寄存器的值，仅 TnAF 中断请求标志产生。所以当 TnCCLR 为高时，不产生 TnPF 中断请求标志。如果 CCRA 被清零，当计数达到最大值 3FFH 时，计数器溢出，而此时不产生 TnAF 请求标志。

正如该模式名所言，当比较匹配发生后，TM 输出脚状态改变。当比较器 A 比较匹配发生后 TnAF 标志产生时，TM 输出脚状态改变。比较器 P 比较匹配发生时产生的 TnPF 标志不影响 TM 输出脚。TM 输出脚状态改变方式由 TMnC1 寄存器中 TnIO1 和 TnIO0 位决定。当比较器 A 比较匹配发生时，TnIO1 和 TnIO0 位决定 TM 输出脚输出高、低或翻转当前状态。TM 输出脚初始值，既可以通过 TnON 位由低到高电平的变化设置，也可以由 TnOC 位设置。注意，若 TnIO1 和 TnIO0 位同时为 0 时，引脚输出不变。



比较匹配输出模式 -- TnCCLR=0

- 注:
1. TnCCLR=0, 比较器 P 匹配将清除计数器
 2. TM 输出脚仅由 TnAF 标志位控制
 3. 在 TnON 上升沿 TM 输出脚复位至初始值



比较匹配输出模式 -- TnCCLR=1

- 注： 1.TnCCLR=1，比较器 A 匹配将清除计数器
 2.TM 输出脚仅由 TnAF 标志位控制
 3.在 TnON 上升沿 TM 输出脚复位至初始值
 4.当 TnCCLR=1 时，TnPF 标志位不会产生

• **CTM, PWM 模式，边沿对齐模式，TnDPX=0**

CCRP	001b	010b	011b	100b	101b	110b	111b	000b
Period	128	256	384	512	640	768	896	1024
Duty	CCRA							

若 $f_{SYS}=8MHz$ ，TM 时钟源选择 $f_{SYS}/4$ ，CCRP=010b，CCRA=128，
 CTM PWM 输出频率= $(f_{SYS}/4)/256=f_{SYS}/1024=7.8125kHz$ ，duty=128/256=50%
 若由 CCRA 寄存器定义的 Duty 值等于或大于 Period 值，PWM 输出占空比为 100%

• **CTM, PWM 模式，边沿对齐模式，TnDPX=1**

CCRP	001b	010b	011b	100b	101b	110b	111b	000b
Period	CCRA							
Duty	128	256	384	512	640	768	896	1024

PWM 的输出周期由 CCRA 寄存器的值与 TM 的时钟共同决定，PWM 的占空比由 CCRP 寄存器的值决定。

定时/计数器模式

为使 TM 工作在此模式，TMnC1 寄存器中的 TnM1 和 TnM0 位需要设置为“11”。定时/计数器模式与比较输出模式操作方式相同，并产生同样的中断请求标志。不同的是，在定时/计数器模式下 TM 输出脚未使用。因此，比较匹配输出模式中的描述和时序图可以适用于此功能。该模式中未使用的 TM 输出脚用作普通 I/O 脚或其它功能。

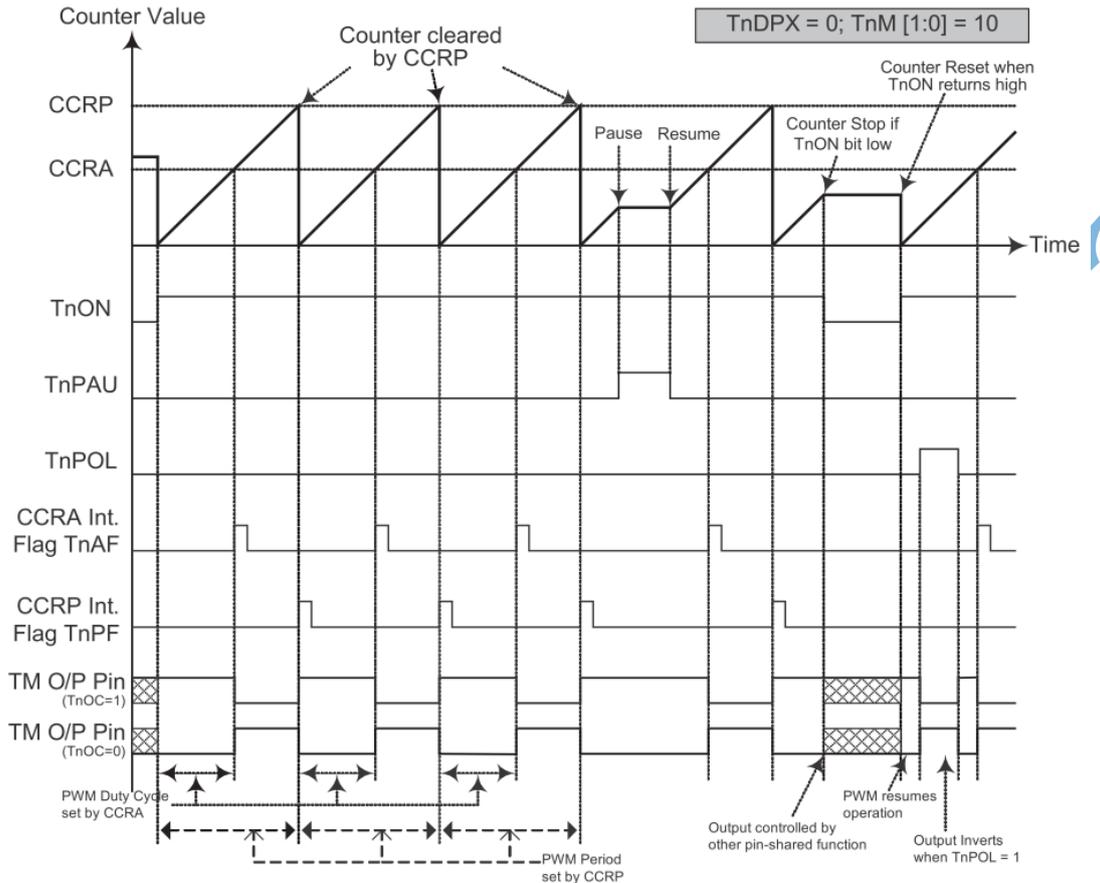
2.11.4 PWM 输出模式

为使 TM 工作在此模式，TMnC1 寄存器中的 TnM1 和 TnM0 位需要设置为“10”。TM 的 PWM

功能在马达控制，加热控制，照明控制等方面十分有用。给 TM 输出脚提供一个频率固定但占空比可调的信号，将产生一个有效值等于 DC 均方根的 AC 方波。

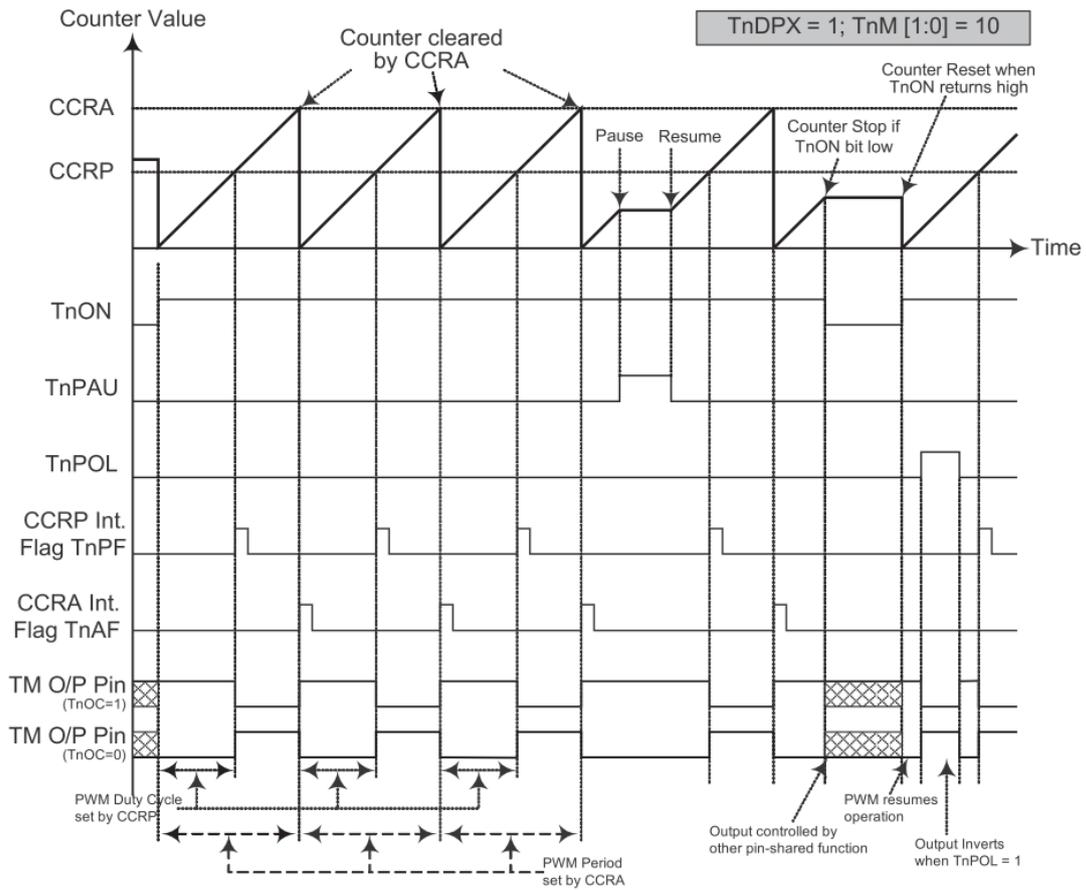
由于 PWM 波形的周期和占空比可调，其波形的选择就极其灵活。在 PWM 模式中，TnCCLR 位不影响 PWM 操作。CCRA 和 CCRP 寄存器决定 PWM 波形，一个用来清除内部计数器并控制 PWM 波形的频率，另一个用来控制占空比。哪个寄存器控制频率或占空比取决于 TMnC1 寄存器的 TnDPX 位。所以 PWM 波形频率和占空比由 CCRA 和 CCRP 寄存器共同决定。

当比较器 A 或比较器 P 比较匹配发生时，将产生 CCRA 或 CCRP 中断标志。TMnC1 寄存器中的 TnOC 位决定 PWM 波形的极性，TnIO1 和 TnIO0 位使能 PWM 输出或将 TM 输出脚置为逻辑高或逻辑低。TnPOL 位对 PWM 输出波形的极性取反。



PWM 模式 -- TnDPX=0

- 注：
1. TnDPX=0, CCRP 清除计数器
 2. 计数器清零并设置 PWM 周期
 3. 当 TnIO1, TnIO0=00 或 01, PWM 功能不变
 4. TnCCLR 位不影响 PWM 操作



PWM 模式 -- TnDPX=1

- 注:
1. TnDPX=1, CCRA 清除计数器
 2. 计数器清零并设置 PWM 周期
 3. 当 TnIO1, TnIO0=00 或 01, PWM 功能不变
 4. TnCCLR 位不影响 PWM 操作

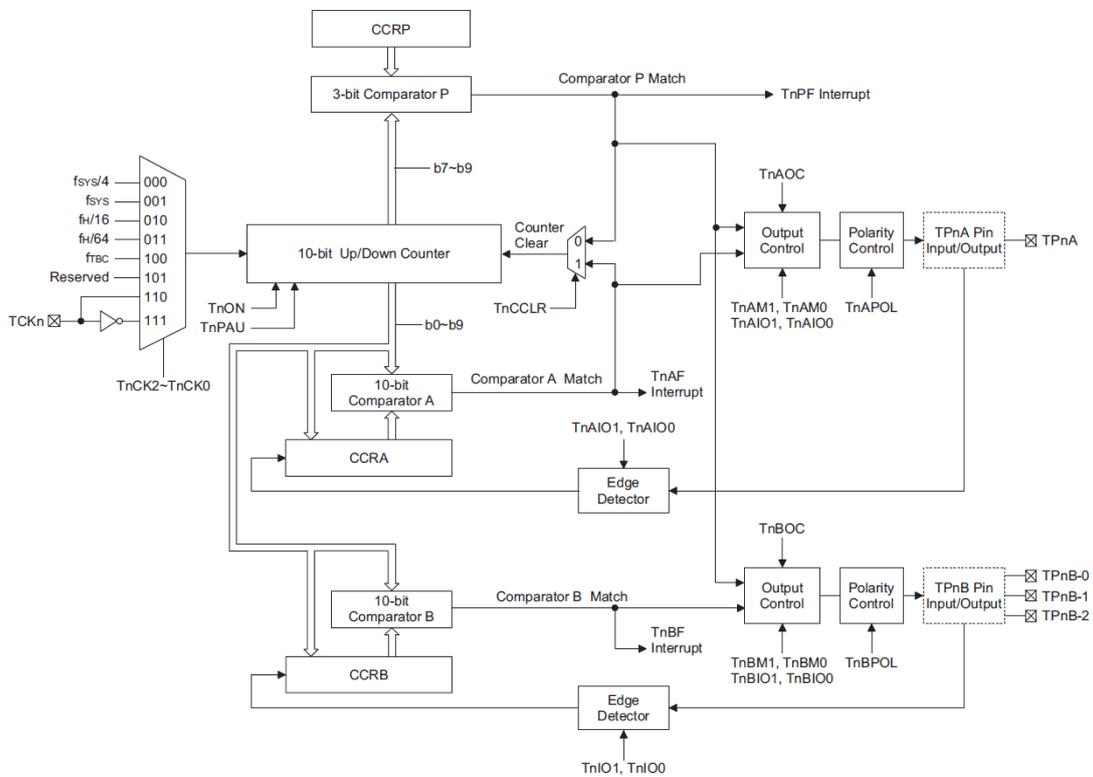
2.12 增强型 TM - ETM (TM1)

增强型 TM 包括 5 种工作模式，即比较匹配输出，定时/事件计数器，捕捉输入，单脉冲输出和 PWM 输出模式。增强型 TM 也由外部输入脚控制并驱动四个外部输出脚。

2.12.1 增强型 TM 操作

增强型 TM 核心是一个由用户选择的内部或外部时钟源驱动的 10 位向上/向下计数器，它还包括三个内部比较器即比较器 A，比较器 B 和比较器 P。这三个比较器将计数器的值与 CCRA，CCRB 和 CCRP 寄存器中的值进行比较。CCRP 是 3 位的，与计数器的高 3 位比较；而 CCRA 和 CCRB 是 10 位的，与计数器的所有位比较。

通过应用程序改变 10 位计数器值的唯一方法是使 TnON 位发生上升沿跳变清除计数器。此外，计数器溢出或比较匹配也会自动清除计数器。上述条件发生时，通常情况会产生 TM 中断信号。增强型 TM 可工作在不同的模式，可由包括来自输入脚的不同时钟源驱动，也可以控制输出脚。所有工作模式的设定都是通过设置相关寄存器来实现的。



Enhanced Type TM Block Diagram

增强型 TM 方框图

2.12.2 增强型 TM 寄存器介绍

增强型 TM 的所有操作由一系列寄存器控制。一对只读寄存器用来存放 10 位计数器的值，两对读/写寄存器存放 10 位 CCRA 和 CCRB 的值。剩下三个控制寄存器用来设置不同的操作和控制模式，以及 CCRP 的 3 个位。

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1C0	T1PAU	T1CK2	T1CK1	T1CK0	T1ON	T1RP2	T1RP1	T1RP0
TM1C1	T1AM1	T1AM0	T1AIO1	T1AIO0	T1AOC	T1APOL	T1CDN	T1CCLR
TM1C2	T1BM1	T1BM0	T1BIO1	T1BIO0	T1BOC	T1BPOL	T1PWM1	T1PWM0
TM1DL	D7	D6	D5	D4	D3	D2	D1	D0
TM1DH	—	—	—	—	—	—	D9	D8
TM1AL	D7	D6	D5	D4	D3	D2	D1	D0
TM1AH	—	—	—	—	—	—	D9	D8
TM1BL	D7	D6	D5	D4	D3	D2	D1	D0
TM1BH	—	—	—	—	—	—	D9	D8

10-bit 增强型 TM 寄存器列表(ETM 为 TM1)

10-bit 增强型 TM 寄存器列表

• TM1C0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	T1PAU	T1CK2	T1CK1	T1CK0	T1ON	T1RP2	T1RP1	T1RP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **T1PAU**: TM1 计数器暂停控制位

0: 运行
1: 暂停

通过设置此位为高可使计数器暂停，清零此位恢复正常计数器操作。当处于暂停条件时，TM 保持上电状态并继续耗电。当此位由低到高转变时，计数器将保留其剩余值，直到此位再次改变为低电平，并从此值开始继续计数。

Bit 6~4 **T1CK2~T1CK0**: 选择 TM1 计数时钟位

000: $f_{SYS}/4$
001: f_{SYS}
010: $f_H/16$
011: $f_H/64$
100: f_{TBC}
101: 保留位

110: TCK1 上升沿时钟
111: TCK1 下降沿时钟

此三位用于选择 TM 的时钟源。选择保留时钟输入将有效地除能内部计数器。外部引脚时钟源能被选择在上升沿或下降沿有效。 f_{SYS} 是系统时钟， f_H 和 f_{TBC} 是其它的内部时钟源，细节方面请参考振荡器章节。

Bit 3 **T1ON**: TM1 计数器 On/Off 控制位

0: Off
1: On

此位控制 TM 的总开关功能。设置此位为高则使能计数器使其运行，清零此位则除能 TM。清零此位将停止计数器并关闭 TM 减少耗电。当此位经由低到高转变时，内部计数器将保持其剩余值，直到此位再次改变为高电平。

若 TM 处于比较匹配输出模式时(通过 TIOC 位指定)，当 TION 位经由低到高的转变时，TM 输出脚将重置其初始值。

Bit 2~0 **T1RP2~T1RP0**: TM1 CCRP 3-bit 寄存器，对应于 TM1 计数器 bit 9~bit 7 比较器 P 匹配周期

000: 1024 个 TM1 时钟周期
001: 128 个 TM1 时钟周期
010: 256 个 TM1 时钟周期
011: 384 个 TM1 时钟周期
100: 512 个 TM1 时钟周期

- 101: 640 个 TM1 时钟周期
- 110: 768 个 TM1 时钟周期
- 111: 896 个 TM1 时钟周期

此三位设定内部 CCRP 3-bit 寄存器的值，然后与内部计数器的高三位进行比较。如果 TICCLR 位设定为 0 时，比较结果为 0 并清除内部计数器。TICCLR 位设为低，内部计数器在比较器 P 比较匹配发生时被重置；由于 CCRP 只与计数器高三位比较，比较结果是 128 时钟周期的倍数。CCRP 被清零时，实际上会使得计数器在最大值溢出。

• TM1C1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	T1AM1	T1AM0	T1AIO1	T1AIO0	T1AOC	T1APOL	T1CDN	T1CCLR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **T1AM1~T1AM0**: 选择 TM1 CCRA 工作模式位

- 00: 比较匹配输出模式
- 01: 捕捉输入模式
- 10: PWM 模式或单脉冲输出模式
- 11: 定时/计数器模式

这两位设置 TM 需要的工作模式。为了确保操作可靠，TM 应在 T1AM1 和 T1AM0 位有任何改变前先关掉。在定时/计数器模式，TM 输出脚控制必须除能。

Bit 5~4 **T1AIO1~T1AIO0**: 选择 TP1A 输出功能位

- 比较匹配输出模式
 - 00: 无变化
 - 01: 输出低
 - 10: 输出高
 - 11: 输出翻转
- PWM 模式/单脉冲输出模式
 - 00: 强制无效状态
 - 01: 强制有效状态
 - 10: PWM 输出
 - 11: 单脉冲输出
- 捕捉输入模式
 - 00: 在 TP1A 上升沿输入捕捉
 - 01: 在 TP1A 下降沿输入捕捉
 - 10: 在 TP1A 双沿输入捕捉
 - 11: 输入捕捉除能

定时/计数器模式

未使用

此两位用于决定在一定条件达到时 TM 输出脚如何改变状态。这两位值的选择决定 TM 运行在哪种模式下。

在比较匹配输出模式下，T1AIO1 和 T1AIO0 位决定当从比较器 A 比较匹配输出发生时 TM 输出脚如何改变状态。当从比较器 A 比较匹配输出发生时 TM 输出脚能设为切换高、切换低或翻转当前状态。若此两位同时为 0 时，这个输出将不会改变。TM 输出脚的初始值通过 TM1C1 寄存器的 T1AOC 位设置取得。注意，由 T1AIO1 和 T1AIO0 位得到的输出电平必须与通过 T1AOC 位设置的初始值不同，否则当比较匹配发生时，TM 输出脚将不会发生变化。在 TM 输出脚改变状态后，通过 T1ION 位由低到高电平的转换复位至初始值。

在 PWM 模式，T1IO1 和 T1IO0 用于决定比较匹配条件发生时怎样改变 TM 输出脚的状态。PWM 输出功能通过这两位的变化进行更新。仅在 TM1 关闭时改变 T1IO1 和 T1IO0 位的值是很有必要的。若在 TM 运行时改变 T1IO1 和 T1IO0 的值，PWM 输出的值是无法预料的。

Bit 3 **T1AOC**: TP1A 输出控制位

- 比较匹配输出模式
 - 0: 初始低
 - 1: 初始高
- PWM 模式/单脉冲输出模式
 - 0: 低有效
 - 1: 高有效

这是 TM 输出脚输出控制位。它取决于 TM 此时正运行于比较匹配输出模式还是 PWM 模式/单脉冲

输出模式。若 TM 处于定时/计数器模式，则其不受影响。在比较匹配输出模式时，比较匹配发生前其决定 TM 输出脚的逻辑电平值。在 PWM 模式时，其决定 PWM 信号是高有效还是低有效。

Bit 2 **T1APOL**: TP1A 输出极性控制位

- 0: 同相
- 1: 反相

此位控制 TP1A 输出脚的极性。此位为高时 TM 输出脚反相，为低时 TM 输出脚同相。若 TM 处于定时/计数器模式时其不受影响。

Bit 1 **T1CDN**: TM1 计数器向上/向下计数标志位

- 0: 向上计数
- 1: 向下计数

Bit 0 **T1CCLR**: 选择 TM1 计数器清零条件位

- 0: TM1 比较器 P 匹配
- 1: TM1 比较器 A 匹配

此位用于选择清除计数器的方法。增强型 TM 包括三个比较器--比较器 P、比较器 A 和比较器 B，其中比较器 P 和比较器 A 都可以用作清除内部计数器。T1CCLR 位设为高，计数器在比较器 A 比较匹配发生时被清除；此位设为低，计数器在比较器 P 比较匹配发生或计数器溢出时被清除。计数器溢出清除的方法仅在 CCRP 被清除为 0 时才能生效。T1CCLR 位在单脉冲或输入捕捉模式时未使用。

• TM1C2 寄存器

Bit	7	6	5	4	3	2	1	0
Name	T1BM1	T1BM0	T1BIO1	T1BIO0	T1BOC	T1BPOL	T1PWM1	T1PWM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **T1BM1~T1BM0**: 选择 TM1 CCRB 工作模式位

- 00: 比较匹配输出模式
- 01: 捕捉输入模式
- 10: PWM 模式或单脉冲输出模式
- 11: 定时/计数器模式

这两位设置 TM 需要的工作模式。为了确保操作可靠，TM 应在 T1BM1 和 T1BM0 位有任何改变前先关掉。在定时/计数器模式，TM 输出脚控制必须除能。

Bit 5~4 **T1BIO1~T1BIO0**: 选择 TP1B_0, TP1B_1, TP1B_2 输出功能位

比较匹配输出模式

- 00: 无变化
- 01: 输出低
- 10: 输出高
- 11: 输出翻转

PWM 模式/单脉冲输出模式

- 00: 强制无效状态
- 01: 强制有效状态
- 10: PWM 输出
- 11: 单脉冲输出

捕捉输入模式

- 00: 在 TP1B_0, TP1B_1, TP1B_2 上升沿输入捕捉
- 01: 在 TP1B_0, TP1B_1, TP1B_2 下降沿输入捕捉
- 10: 在 TP1B_0, TP1B_1, TP1B_2 双沿输入捕捉
- 11: 输入捕捉除能

定时/计数器模式

未使用

此两位用于决定在一定条件达到时 TM 输出脚如何改变状态。这两位值的选择决定 TM 运行在何种模式下。

在比较匹配输出模式下，T1BIO1 和 T1BIO0 位决定当比较器 A 比较匹配输出发生时 TM 输出脚如何改变状态。当比较器 A 比较匹配输出发生时 TM 输出脚能设为切换高、切换低或翻转当前状态。若此两位同时为 0 时，这个输出将不会改变。TM 输出脚的初始值通过 TM1C2 寄存器的 T1BOC 位设置取得。注意，由 T1BIO1 和 T1BIO0 位得到的输出电平必须与通过 T1BOC 位设置的初始值不同，否则当比较匹配发生时，TM 输出脚将不会发生变化。在 TM 输出脚改变状态后，通过 T1ON 位由低到高电平的转换复位至初始值。

在 PWM 模式下, T1BIO1 和 T1BIO0 位决定当发生比较器匹配输出时 TM 输出脚如何改变状态, PWM 输出功能的切换可通过改变这两位。只有 TM 关闭后才需要改变 T1BIO1 和 T1BIO0 的值。若 TM 运行时改变 T1BIO1 和 T1BIO0 的值, 将产生不可预见的 PWM 输出。

Bit 3 **T1BOC:** TP1B_0, TP1B_1, TP1B_2 输出控制位

比较匹配输出模式

0: 初始低

1: 初始高

PWM 模式/单脉冲输出模式

0: 低有效

1: 高有效

这是 TM 输出脚输出控制位。它取决于 TM 此时正运行于比较匹配输出模式还是 PWM 模式/单脉冲输出模式。若 TM 处于定时/计数器模式, 则其不受影响。在比较匹配输出模式时, 比较匹配发生前其决定 TM 输出脚的逻辑电平值。在 PWM 模式时, 其决定 PWM 信号是高有效还是低有效。

Bit 2 **T1BPOL:** TP1B_0, TP1B_1, TP1B_2 输出极性控制位

0: 同相

1: 反相

此位控制 TP1B_0, TP1B_1, TP1B_2 输出脚的极性。此位为高时 TM 输出脚反相, 为低时 TM 输出脚同相。若 TM 处于定时/计数器模式时其不受影响。

Bit 1~0 **T1PWM1~T1PWM0:** 选择 PWM 模式位

00: 边沿对齐

01: 中心对齐, 向上计数比较匹配

10: 中心对齐, 向下计数比较匹配

11: 中心对齐, 向上/下计数比较匹配

• TM1DL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~0 **TM1DL:** TM1 计数器低字节寄存器 bit 7~bit 0
TM1 10-bit 计数器 bit 7~bit 0

• TM1DH 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R	R
POR	—	—	—	—	—	—	0	0

Bit 7~2 未定义, 读为“0”

Bit 1~0 **TM1DH:** TM1 计数器高字节寄存器 bit 1~bit 0
TM1 10-bit 计数器 bit 9~bit 8

• TM1AL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **TM1AL:** TM1 CCRA 低字节寄存器 bit 7~bit 0
TM1 10-bit CCRA bit 7~bit 0

• TM1AH 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 未定义，读为“0”

 Bit 1~0 **TM1AH:** TM1 CCRA 高字节寄存器 bit 1~bit 0
 TM1 10-bit CCRA bit 9~bit 8

• TM1BL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

 Bit 7~0 **TM1BL:** TM1 CCRB 低字节寄存器 bit 7~bit 0
 TM1 10-bit CCRB bit 7~bit 0

• TM1BH 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 未定义，读为“0”

 Bit 1~0 **TM1BH:** TM1 CCRB 高字节寄存器 bit 1~bit 0
 TM1 10-bit CCRB bit 9~bit 8

2.12.3 增强型 TM 工作模式

增强型 TM 有五种工作模式，即比较匹配输出模式，PWM 输出模式，单脉冲输出模式，捕捉输入模式或定时/计数器模式。通过设置 TMnC1 寄存器的 TnAM1 和 TnAM0 位和 TMnC2 寄存器的 TnBM1 和 TnBM0 位选择任意模式。

ETM 工作模式	CCRA 比较匹配输出模式	CCRA 定时/计数器模式	CCRA PWM 输出模式	CCRA 单脉冲输出模式	CCRA 输入捕捉模式
CCRB 比较匹配输出模式	√	—	—	—	—
CCRB 定时/计数器模式	—	√	—	—	—
CCRB PWM 输出模式	—	—	√	—	—
CCRB 单脉冲输出模式	—	—	—	√	—
CCRB 输入捕捉模式	—	—	—	—	√

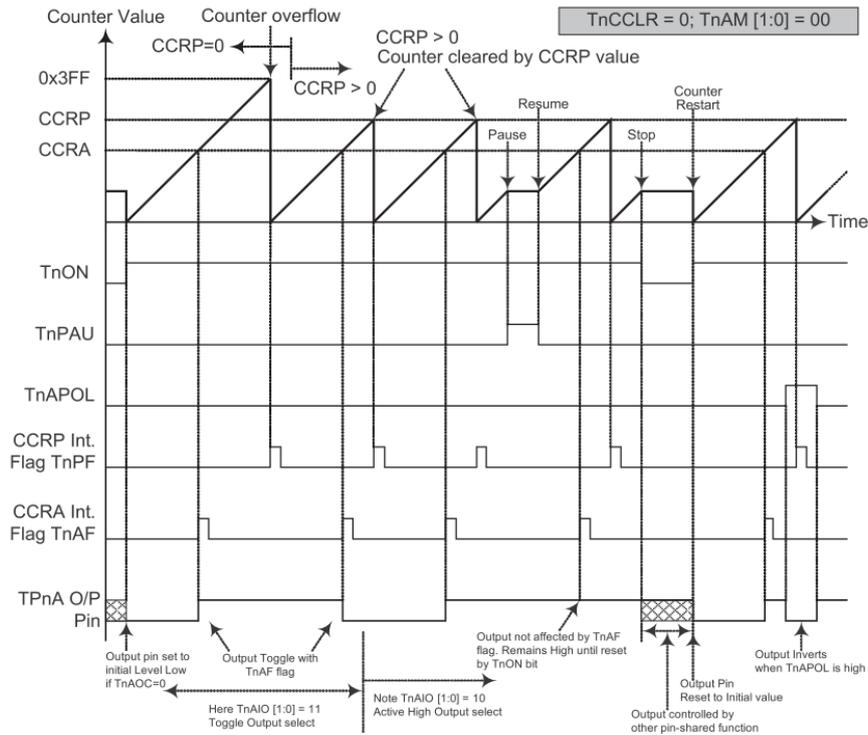
“√”: 允许, “—”: 不允许

比较匹配输出模式

为使 TM 工作在此模式，TMnC1 寄存器的 TnAM1, TnAM0 位和 TMnC2 寄存器的 TnBM1, TnBM0 位需要全部清零。当工作在该模式，一旦计数器使能并开始计数，有三种方法来清零，分别是：计数器溢出，比较器 A 比较匹配发生和比较器 P 比较匹配发生。当 TnCCLR 位为低，有两种方法清除计数器。一种是比较器 P 比较匹配发生，另一种是 CCRP 所有位设置为零并使得计数器溢出。此时，比较器 A 和比较器 P 的请求标志位 TnAF 和 TnPF 将分别置起。

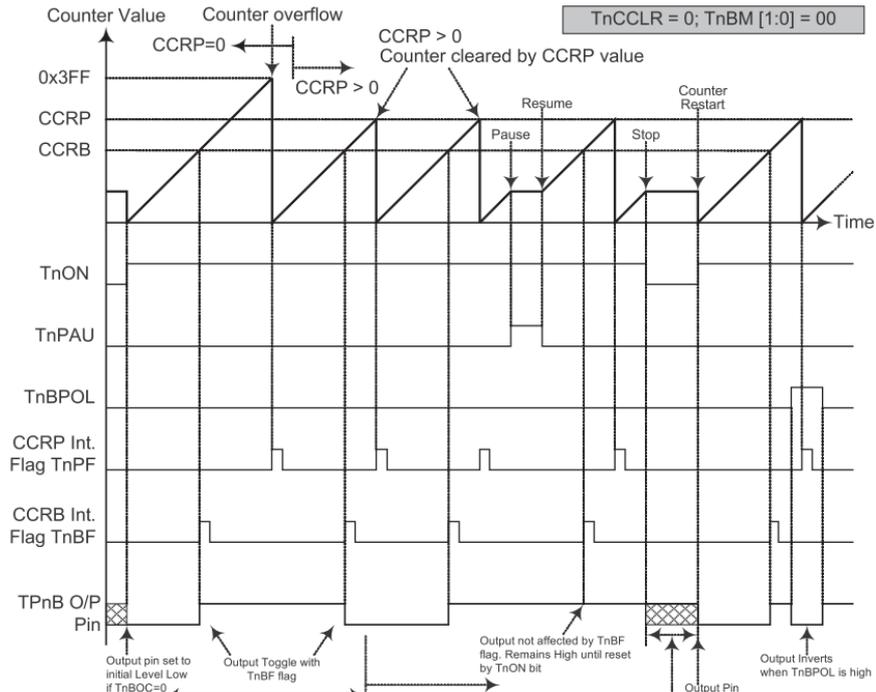
如果 TMnC1 寄存器的 TnCCLR 位设置为高，当比较器 A 比较匹配发生时计数器被清零。此时，即使 CCRP 寄存器的值小于 CCRA 寄存器的值，仅 TnAF 中断请求标志产生。所以当 TnCCLR 为高时，不会产生 TnPF 中断请求标志。

正如该模式名所言，当比较匹配发生后，TM 输出脚状态改变。当比较器 A 或比较器 B 比较匹配发生后 TnAF 或 TnBF 中断请求标志产生时，TM 输出脚状态改变。比较器 P 比较匹配发生时产生的 TnPF 标志不影响 TM 输出脚。TM 输出脚状态改变方式由 ETM CCRA 的 TMnC1 寄存器中 TnAIO1 和 TnAIO0 位，ETM CCRB 的 TMnC2 寄存器中的 TnBIO1 和 TnBIO0 位决定。当比较器 A 或比较器 B 比较匹配发生时，TnAIO1, TnAIO0 位(对于 TPnA 引脚)和 TnBIO1, TnBIO0 位(对于 TPnB_0, TPnB_1, TPnB_2 引脚)决定 TM 输出脚输出高，低或翻转当前状态。TM 输出脚初始值，既可以通过 TnON 位由低到高电平的变化设置，也可以由 TnAOC 或 TnBOC 位设置。注意，若 TnAIO1, TnAIO0 和 TnBIO1, TnBIO0 位同时为 0 时，引脚输出不变。



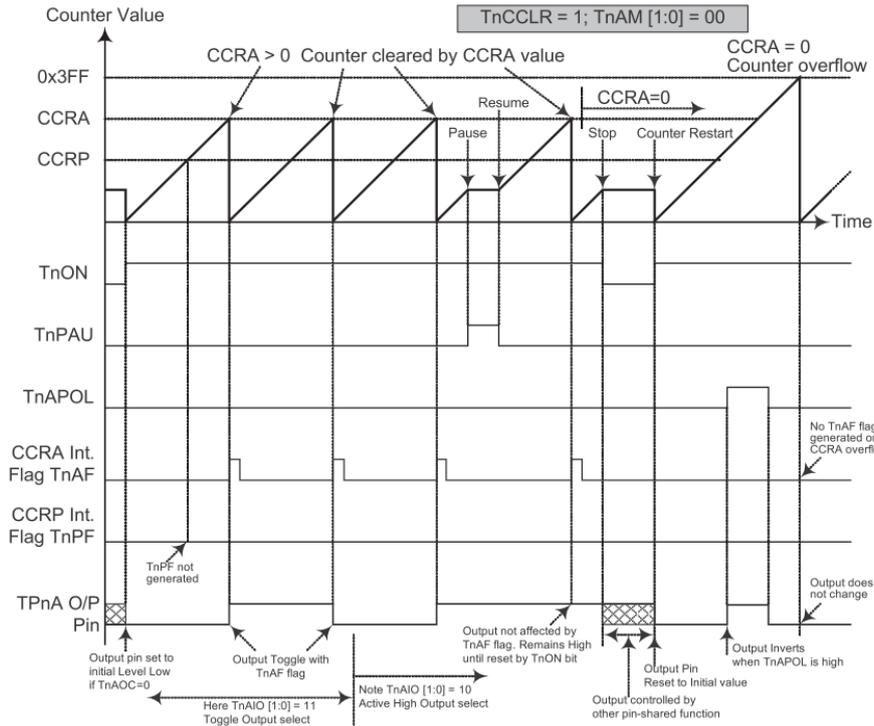
ETM CCRA 比较匹配输出模式 -- TnCCLR=0

- 注：
1. TnCCLR=0，比较器 P 匹配将清除计数器
 2. TPnA 输出脚仅由 TnAF 标志位控制
 3. 在 TnON 上升沿 TM 输出脚复位至初始值



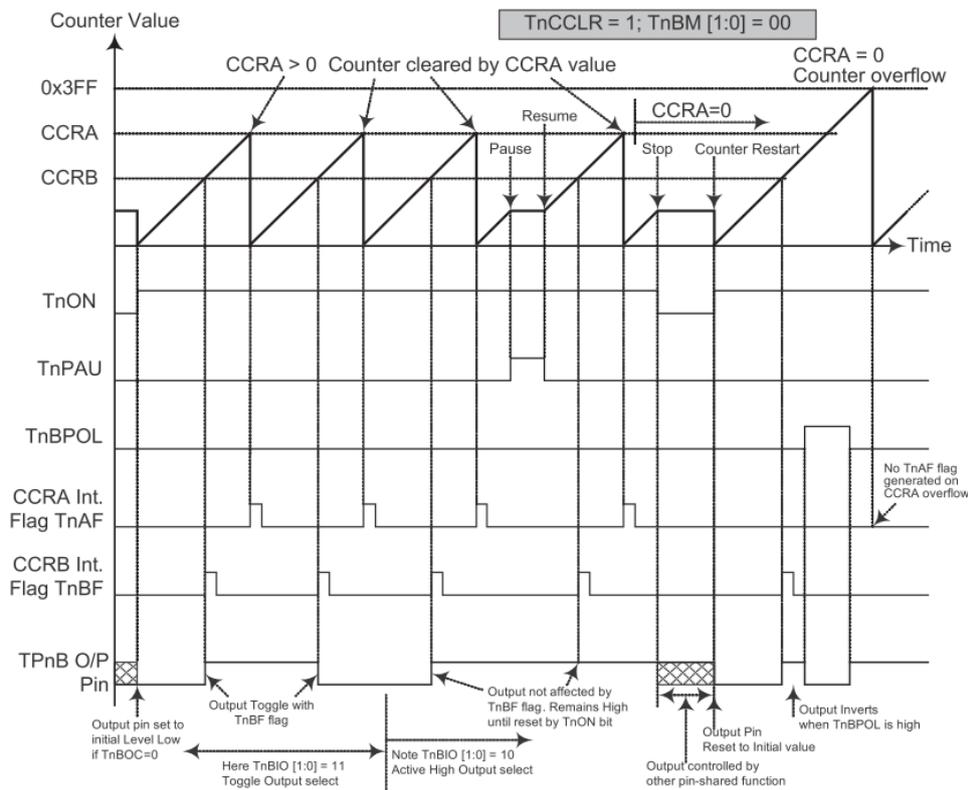
ETM CCRB 比较匹配输出模式 -- TnCCLR=0

- 注: 1. TnCCLR=0, 比较器 P 匹配将清除计数器
- 2. TPnB 输出脚仅由 TnBF 标志位控制
- 3. 在 TnON 上升沿 TM 输出脚复位至初始值



ETM CCRA 比较匹配输出模式 -- TnCCR=1

- 注:
1. $TnCCR=1$, 比较器 A 匹配将清除计数器
 2. TPnA 输出脚仅由 TnAF 标志位控制
 3. 在 TnON 上升沿 TPnA 输出脚复位至初始值
 4. 当 $TnCCR=1$ 时, 不会产生 TnPF 标志



ETM CCRB 比较匹配输出模式 -- TnCCLR=1

- 注：
1. TnCCLR=1, 比较器 A 匹配将清除计数器
 2. TPnB 输出脚仅由 TnBF 标志位控制
 3. 在 TnON 上升沿 TPnB 输出脚复位至初始值
 4. 当 TnCCLR=1 时, 不会产生 TnPF 标志

定时/计数器模式

为使 TM 工作在此模式, TMnC1 寄存器的 TnAM1, TnAM0 位和 TMnC2 寄存器的 TnBM1, TnBM0 位需要全部设为高。定时/计数器模式与比较输出模式操作方式相同, 并产生同样的中断请求标志。不同的是, 在定时/计数器模式下 TM 输出脚未使用。因此, 比较匹配输出模式中的描述和时序图可以适用于此功能。该模式中未使用的 TM 输出脚用作普通 I/O 脚或其它功能。

PWM 输出模式

为使 TM 工作在此模式, TnAM1, TnAM0 和 TnBM1, TnBM0 位需要分别设置为“10”, 且 TnAIO1, TnAIO0 和 TnBIO1, TnBIO0 位也需要分别设置为“10”。TM 的 PWM 功能在马达控制, 加热控制, 照明控制等方面十分有用。给 TM 输出脚提供一个频率固定但占空比可调的信号, 将产生一个有效值等于 DC 均方根的 AC 方波。

由于 PWM 波形的周期和占空比可调, 其波形的选择就极其灵活。在 PWM 模式中, TnCCLR 位决定 PWM 周期控制方式。当 TnCCLR 设为高, CCRA 寄存器控制 PWM 周期。在这种情况下, CCRB 寄存器设置 PWM 的占空比(针对 TPnB 输出脚)。CCRP 寄存器和 TPnA 输出脚未使用。PWM 输出只在 TPnB 输出脚产生。当 TnCCLR 清零时, PWM 周期通过 CCRP 三位中八个值之一设置, 并且是 128 的倍数。此时, CCRA 和 CCRB 寄存器设置不同占空比, 在 TPnA 和 TPnB 引脚输出两个 PWM 波形。

TnPWM1 和 TnPWM0 位决定 PWM 的对齐方式, 即边沿或中心对齐方式。在边沿对齐方式中, 当计数器清零时, 产生 PWM 前沿信号。与此同时电流发生跳变, 这在高功耗应用中会出现问题。在中心对齐方式中, PWM 波形中心持续产生有效信号, 因此可以减少电流跳变引起的功耗问题。

当比较器 A, 比较器 B 或比较器 P 比较匹配发生时, CCRA, CCRB 和 CCRP 中断标志位分别产生。TMnC1 寄存器的 TnAOC 位和 TMnC2 寄存器的 TnBOC 位选择 PWM 波形的极性, TnAIO1,

TnAIO0 和 TnBIO1, TnBIO0 位使能 PWM 输出或迫使 TM 输出脚为高电平或低电平。TnAPOL 和 TnBPOL 位用来取反 PWM 输出波形的极性。

• ETM, PWM 模式, 边沿对齐模式, TnCCLR=0

CCRP	001b	010b	011b	100b	101b	110b	111b	000b
Period	128	256	384	512	640	768	896	1024
A Duty	CCRA							
B Duty	CCRB							

• ETM, PWM 模式, 边沿对齐模式, TnCCLR=1

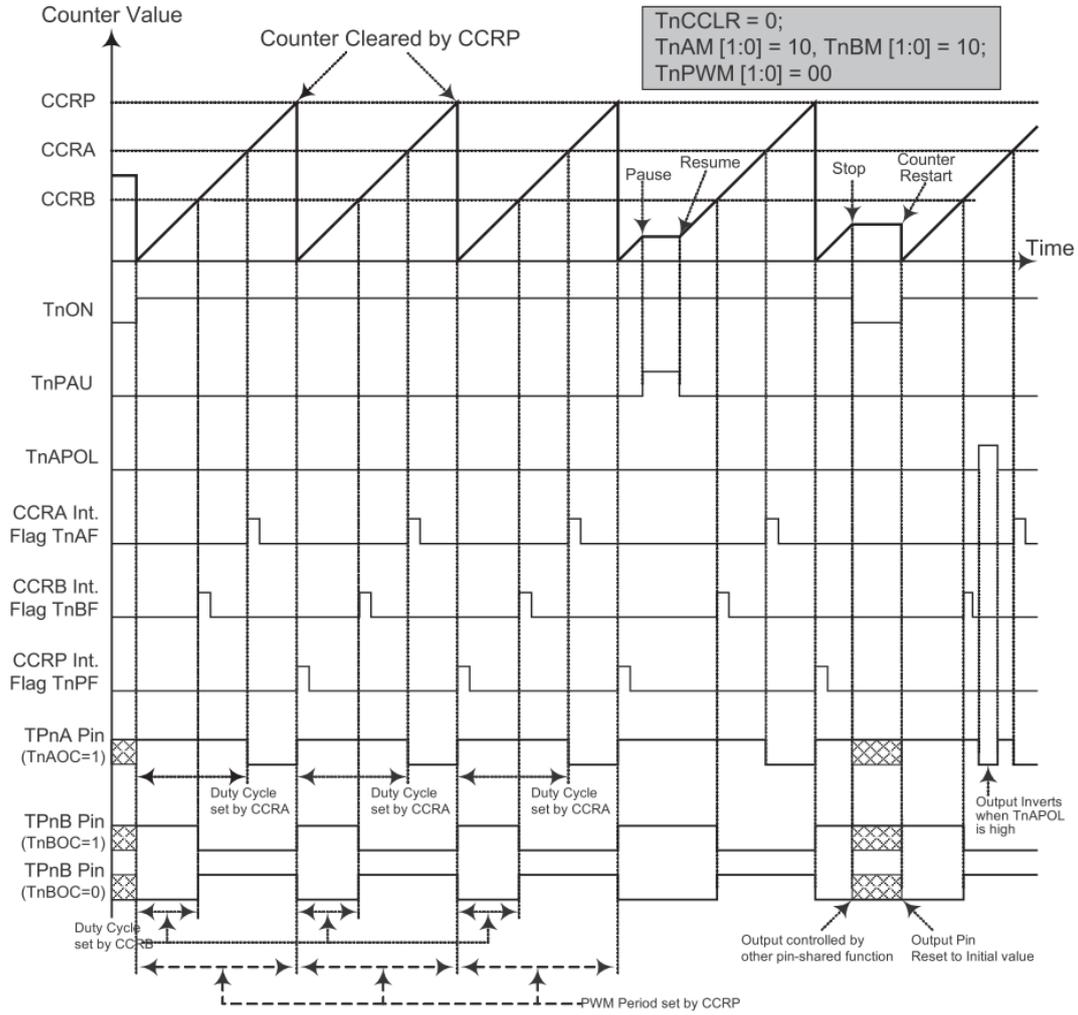
CCRA	1	2	3	511	512	1021	1022	1023
Period	1	2	3	511	512	1021	1022	1023
B Duty	CCRB							

• ETM, PWM 模式, 中心对齐模式, TnCCLR=0

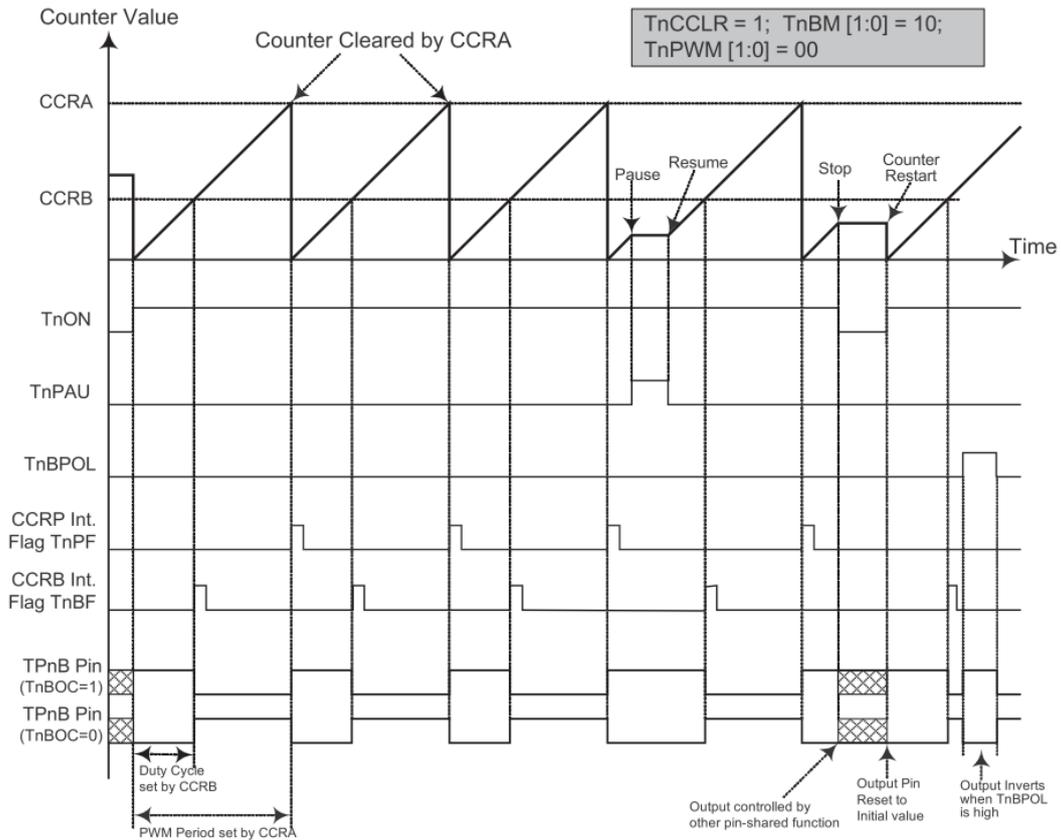
CCRP	001b	010b	011b	100b	101b	110b	111b	000b
Period	256	512	768	1024	1280	1536	1792	2046
A Duty	$(CCRA \times 2) - 1$							
B Duty	$(CCRB \times 2) - 1$							

• ETM, PWM 模式, 中心对齐模式, TnCCLR=1

CCRA	1	2	3	511	512	1021	1022	1023
Period	2	4	6	1022	1024	2042	2044	2046
B Duty	$(CCRB \times 2) - 1$							

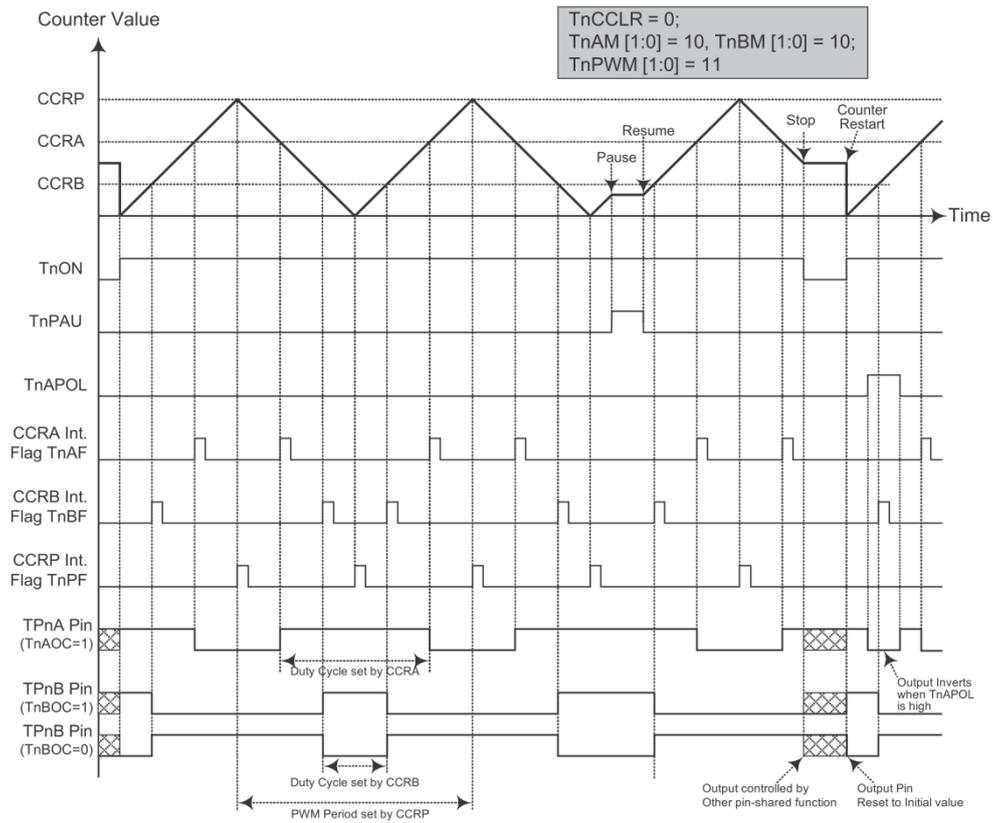


- 注:
1. TnCCLR=0, CCRP 清除计数器并决定 PWM 周期
 2. 当 TnAIO1, TnAIO0(或 TnBIO1, TnBIO0)=00 或 01, PWM 功能不变
 3. CCRA 控制 TPnA PWM 占空比, CCRB 控制 TPnB PWM 占空比



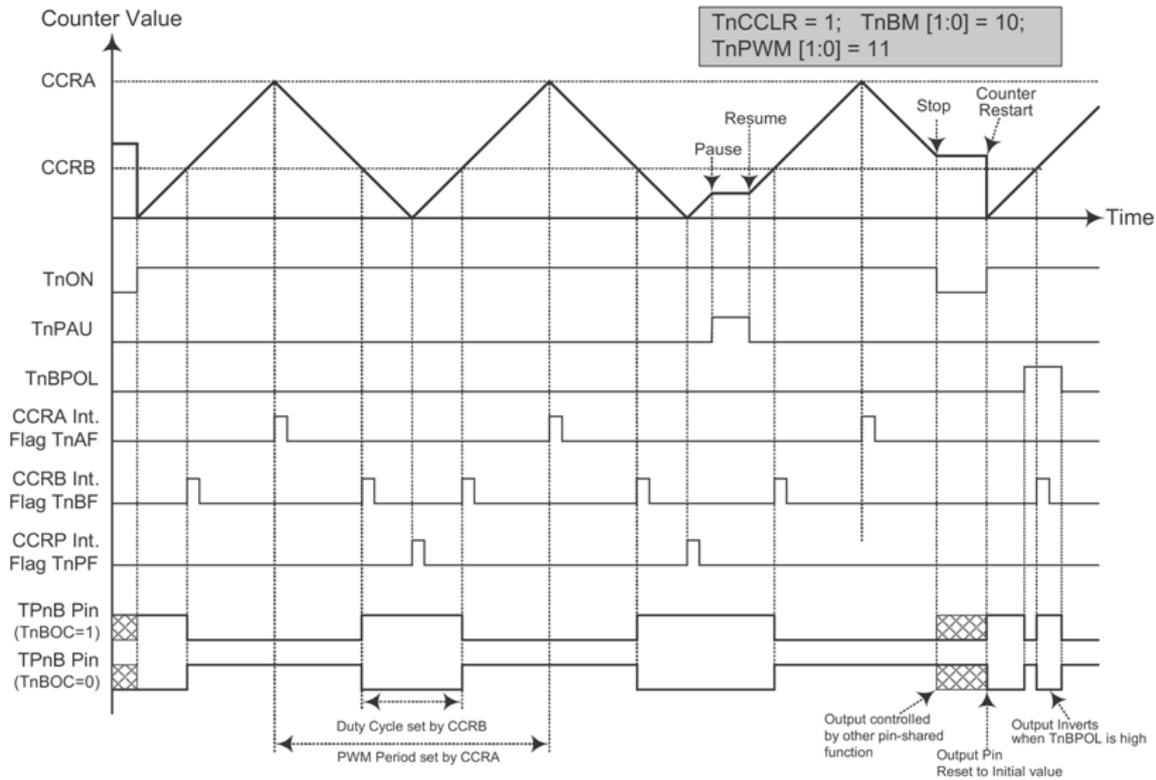
ETM PWM 模式 -- 边沿对齐

- 注:
1. TnCCLR=1, CCRA 清除计数器并决定 PWM 周期
 2. 当 TnBIO1, TnBIO0=00 或 01, PWM 功能不变
 3. CCRA 控制 TPnB PWM 周期, CCRB 控制 TPnB PWM 占空比



ETM PWM 模式 -- 中心对齐

- 注:
1. $TnCCLR=0$, CCRP 清除计数器并决定 PWM 周期
 2. $TnPWM1/TnPWM0=11$, PWM 为中心对齐
 3. 当 $TnAIO1, TnAIO0$ (或 $TnBIO1, TnBIO0$)=00 或 01, PWM 功能不变
 4. CCRA 控制 TPnA PWM 占空比, CCRB 控制 TPnB PWM 占空比
 5. 计数器值递减至“0”时 CCRP 将产生中断请求



ETM PWM 模式 -- 中心对齐

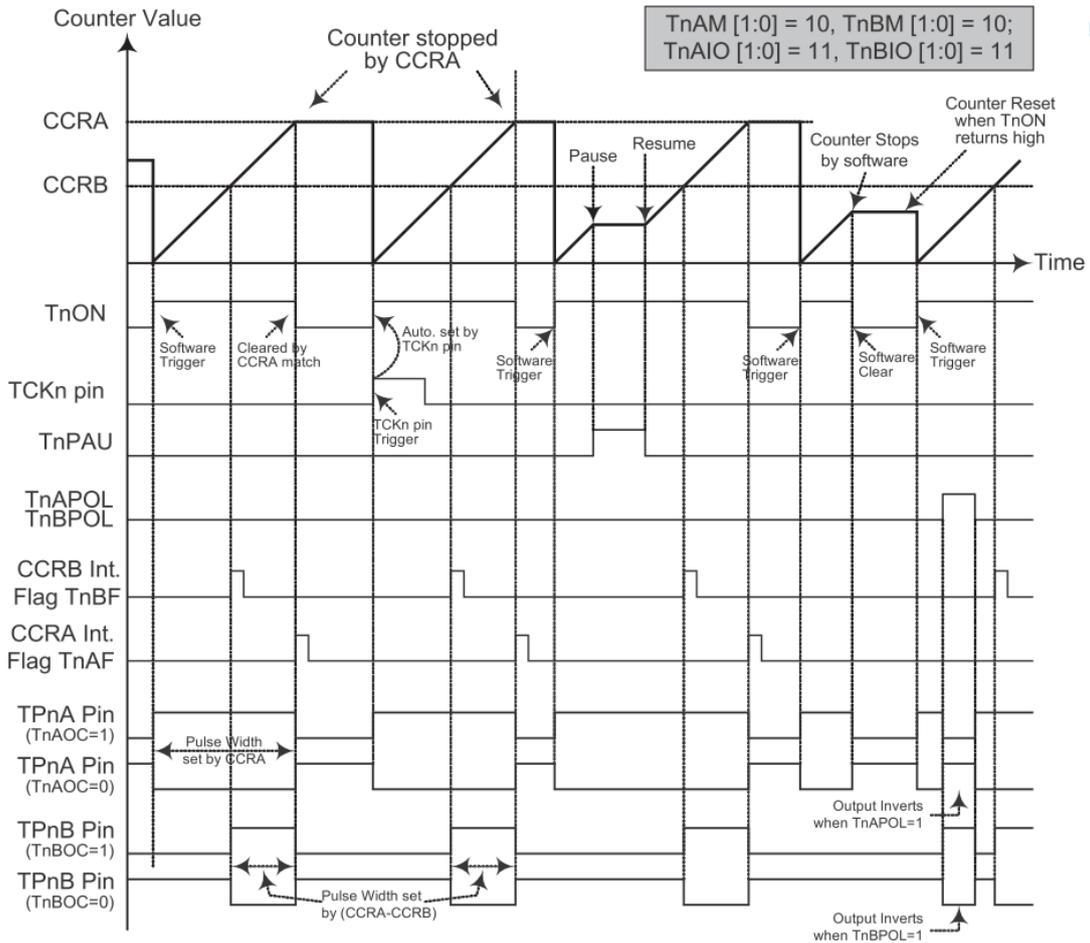
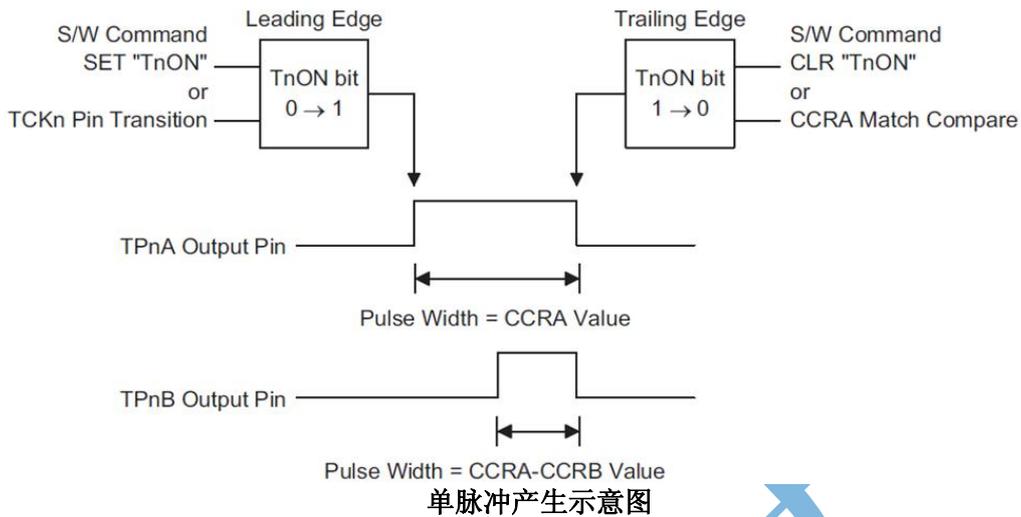
- 注：
1. TnCCLR=1, CCRA 清除计数器并决定 PWM 周期
 2. TnPWM1/TnPWM0=11, PWM 为中心对齐
 3. 当 TnBIO1, TnBIO0=00 或 01 时, PWM 功能不变
 4. CCRA 控制 TPnB PWM 周期, CCRB 控制 TPnB PWM 占空比
 5. 计数器值递减至“0”时 CCRP 将产生中断请求

单脉冲输出模式

为使 TM 工作在此模式, TnAM1, TnAM0 和 TnBM1, TnBM0 位需要分别设置为“10”, 并且相应的 TnAIO1, TnAIO0 和 TnBIO1, TnBIO0 需要分别设置为“11”。正如模式名所言, 单脉冲输出模式, 在 TM 输出脚将产生一个脉冲输出。

通过应用程序控制 TnON 位由低到高的转变来触发 TPnA 脉冲前沿输出。通过应用程序产生比较器 B 的比较匹配来触发 TPnB 脉冲前沿输出。而处于单脉冲模式时, TnON 位可由 TCKn 脚自动由低转变为高, 进而依次初始化 TPnA 的单脉冲输出。当 TnON 位转变为高电平时, 计数器将开始运行, 并产生 TPnA 脉冲前沿。当脉冲有效时 TnON 位保持高电平。通过应用程序使 TnON 位清零或比较器 A 比较匹配发生时, 产生 TPnA 和 TPnB 的脉冲下降沿。

而比较器 A 比较匹配发生时, 会自动清除 TnON 位并产生 TPnA 和 TPnB 单脉冲输出下降沿。CCRA 的值通过这种方式控制 TPnA 的脉冲宽度, CCRA-CCRB 的值控制 TPnB 的脉冲宽度。比较器 A 和比较器 B 比较匹配发生时, 也会产生 TM 中断。TnON 位在计数器重启时会发生由低到高的转变, 此时计数器才复位至零。在单脉冲模式中, CCRP 寄存器和 TnCCLR 位未使用。



ETM -- 单脉冲模式

- 注：
- 1.通过 CCRA 匹配停止计数器
 - 2.CCRP 未使用
 - 3.通过 TCKn 脚或设置 TnON 位为高来触发脉冲
 - 4.TCKn 脚有效沿会自动置位 TnON
 - 5.单脉冲模式中，TnAIO[1:0]和 TnBIO[1:0]需置位“11”，且不能更改。

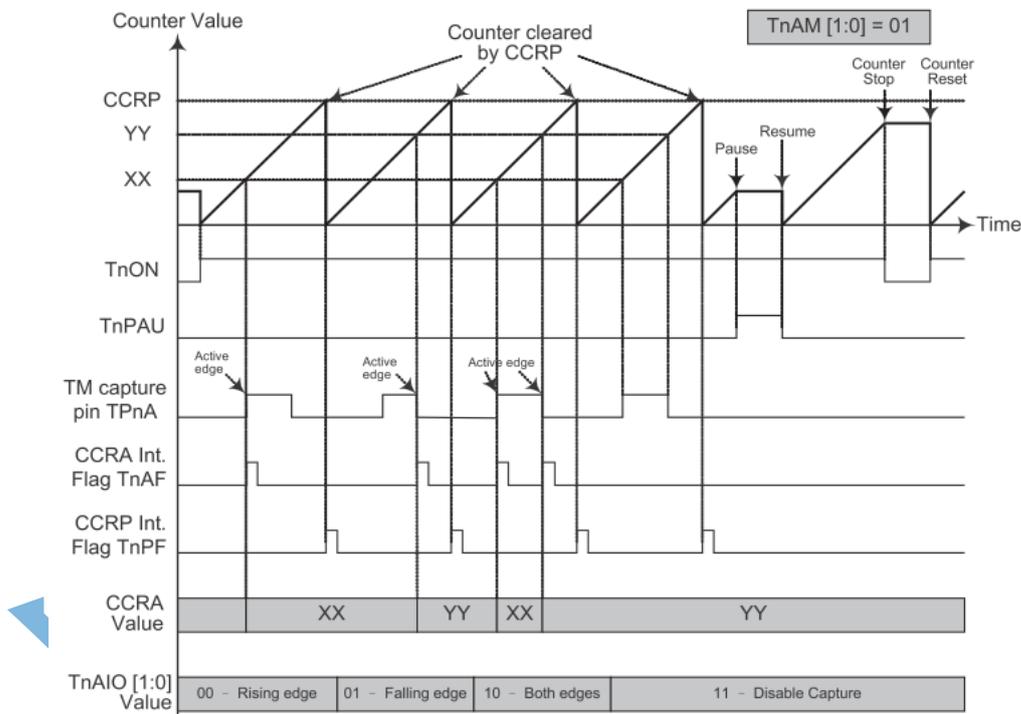
捕捉输入模式

为使 TM 工作在此模式，TMnC1 寄存器的 TnAM1, TnAM0 位和 TMnC2 寄存器的 TnBM1, TnBM0 位需要分别设置为“01”。此模式使能外部信号捕捉并保存内部计数器当前值，因此被用于诸如脉冲

宽度测量的应用中。TPnA 和 TPnB_0, TPnB_1, TPnB_2 引脚上的外部信号, 通过设置 TMnC1 寄存器的 TnAIO1, TnAIO0 位和 TMnC2 寄存器的 TnBIO1, TnBIO0 位选择有效边沿类型, 即上升沿, 下降沿或双沿有效。计数器在 TnON 位由低到高转变时启动并通过应用程序初始化。

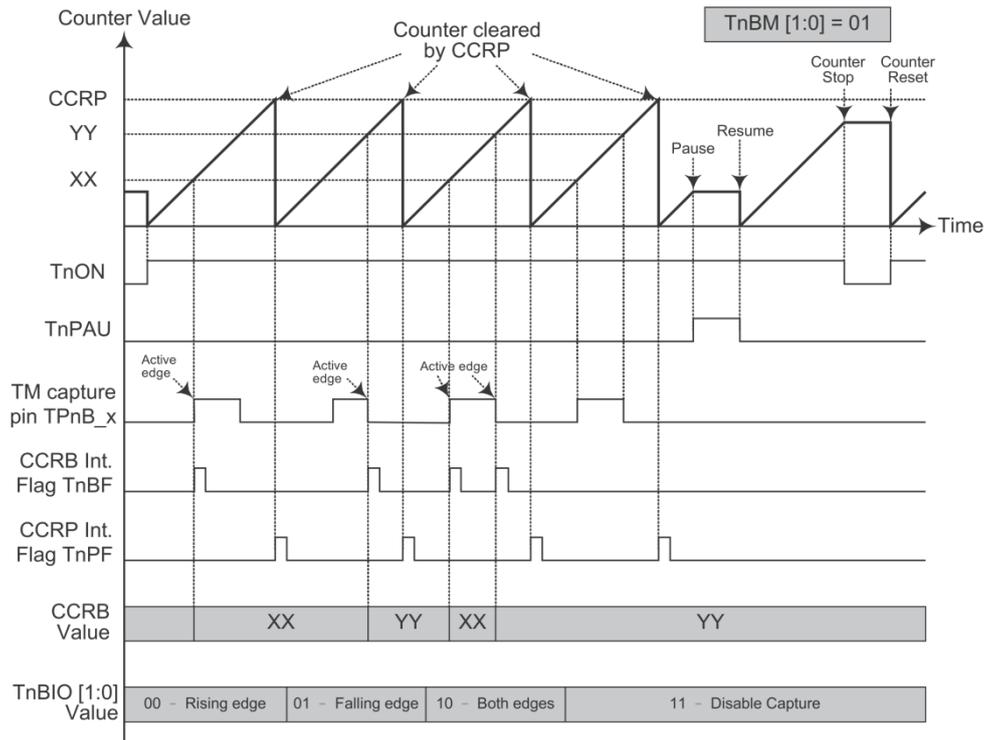
当 TPnA 和 TPnB_0, TPnB_1, TPnB_2 引脚出现有效边沿转换时, 计数器当前值被锁存到 CCRA 和 CCRB 寄存器, 并产生 TM 中断。不考虑 TPnA 和 TPnB_0, TPnB_1, TPnB_2 引脚事件, 计数器继续工作直到 TnON 位发生下降沿跳变。当 CCRP 比较匹配发生时计数器复位至零; CCRP 的值通过这种方式控制计数器的最大值。当比较器 P CCRP 比较匹配发生时, 也会产生 TM 中断。记录 CCRP 溢出中断信号的值可以测量长脉宽。通过设置 TnAIO1, TnAIO0 位和 TnBIO1, TnBIO0 位选择 TPnA 和 TPnB_0, TPnB_1, TPnB_2 引脚为上升沿, 下降沿或双沿有效。不考虑 TPnA 和 TPnB_0, TPnB_1, TPnB_2 引脚事件, 如果 TnAIO1, TnAIO0 位和 TnBIO1, TnBIO0 位都设为高, 不会产生捕捉操作, 但计数器继续运行。

当 TPnA 和 TPnB_0, TPnB_1, TPnB_2 引脚与其它功能共用, TM 工作在输入捕捉模式时需多加注意。这是因为如果引脚被设为输出, 那么该引脚上的任何电平转变都可能执行输入捕捉操作。TnCCLR, TnAOC, TnBOC, TnAPOL 和 TnBPOL 位在此模式中未使用。



ETM CCRA 捕捉输入模式

- 注:
1. TnAM1, TnAM0=01 并通过 TnAIO1 和 TnAIO0 位设置有效边沿
 2. TM 捕捉输入脚的有效边沿将计数器的值转移到 CCRA 中
 3. TnCCLR 位未使用
 4. 无输出功能-TnAOC 和 TnAPOL 位未使用
 5. 计数器值由 CCRP 决定, 在 CCRP 为“0”时, 计数器计数值可达最大



ETM CCRB 捕捉输入模式

- 注:
1. TnBM1, TnBM0=01 并通过 TnBIO1 和 TnBIO0 位设置有效边沿
 2. TM 捕捉输入脚的有效边沿将计数器的值转移到 CCRB 中
 3. TnCCLR 位未使用
 4. 无输出功能-TnBOC 和 TnBPOL 位未使用
 5. 计数器值由 CCRP 决定, 在 CCRP 为“0”时, 计数器计数值可达最大

2.13 运算放大器 OPA

该单片机内部集成个运算放大器，可用于用户特定的模拟信号处理。它的使能或除能只能通过软件设置来实现。通过控制特殊寄存器，OPA 相关的应用可以很容易的实现，例如单位增益缓冲器，同相放大器，反相放大器和各种各样的滤波器等。此外，该单片机提供了校准功能，用以校准 OPA 的失调电压。

2.13.1 运算放大器寄存器

内部运算放大器的控制寄存器有：OPAC0，OPAC1 和 OPAC2。这些寄存器用来控制运算放大器的使能/除能功能、输入路径选择、增益控制、极性和校准等功能。

• OPAC0 控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	OPAEN	OPAO	OPAPOL	STEN	SA8EN	SA7EN	SA6EN	SA1EN
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **OPAEN**: OPA 控制位

0: 除能
1: 使能

Bit 6 **OPAO**: OPA 输出位

OPAPOL=0: 如果 $V_+ < V_-$, OPAO=0
如果 $V_+ > V_-$, OPAO=1
OPAPOL=1: 如果 $V_+ < V_-$, OPAO=1
如果 $V_+ > V_-$, OPAO=0

V_+ :运算放大器同相输入, V_- :运算放大器反相输入

Bit 5 **OPAPOL**: OPA 极性选择位

0: 同相
1: 反相

Bit 4 **STEN**: OPA 施密特触发控制位

0: 除能
1: 使能

当施密特触发被除能时，OPA 被用作运算放大器，否则被用作比较器。

Bit 3 **SA8EN**: 开关 SA8 控制位

0: 关闭
1: 开启

Bit 2 **SA7EN**: 开关 SA7 控制位

0: 关闭
1: 开启

Bit 1 **SA6EN**: 开关 SA6 控制位

0: 关闭
1: 开启

Bit 0 **SA1EN**: 开关 S1 控制位

0: 关闭
1: 开启

• OPAC1 控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	AOFM	ARS	AOF4	AOF3	AOF2	AOF1	AOF0
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	0	0	1	0	0	0	0

- Bit 7 未定义，读为“0”
- Bit 6 **AOFM**: 输入失调电压校准模式/运算放大器模式选择位
 0: 运算放大器模式
 1: 输入失调电压校准模式
- Bit 5 **ARS**: 运算放大器输入失调电压校准参考选择位
 0: 选择 AN 作为参考输入脚
 1: 选择 AP 作为参考输入脚
- Bit 4~0 **AOF4-AOF0**: 运算放大器输入失调电压校准控制位

• OPAC2 控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	SA5EN	SA4EN	SA3EN	SA2EN	SA12EN	—	PGAG1	PGAG0
R/W	R/W	R/W	R/W	R/W	R/W	—	R/W	R/W
POR	0	0	0	0	0	—	0	0

- Bit 7 **SA5EN**: 开关 SA5 控制位
 0: 关闭
 1: 开启
- Bit 6 **SA4EN**: 开关 SA4 控制位
 0: 关闭
 1: 开启
- Bit 5 **SA3EN**: 开关 SA3 控制位
 0: 关闭
 1: 开启
- Bit 4 **SA2EN**: 开关 SA2 控制位
 0: 关闭
 1: 开启
- Bit 3 **SA12EN**: 开关 SA12 控制位
 0: 关闭
 1: 开启
- Bit 2 未定义，读为“0”
- Bit 1~0 **PGAG1-PGAG0**: PGA 增益控制位
 00: 8
 01: 16
 10: 32
 11: 64

2.14 比较器

该单片机中含一个模拟比较器。它具有暂停、中断等功能，可通过寄存器进行灵活配置。比较器的引脚与普通 I/O 引脚共用，当比较器功能未使用时，此引脚可做普通引脚使用而不浪费 I/O 资源。除此之外，该单片机还提供了比较器失调电压校准功能。

2.14.1 比较器操作

此单片机包含一个比较器功能，用于比较两个模拟电压，基于它们的差值上提供一个输出。控制寄存器 CPC0、CPC1 可分别控制相应的内部比较器。比较器的输出可由各自寄存器的一位记录，并且在共用的 I/O 口上输出。

2.14.2 比较器寄存器

比较器由控制寄存器 CPC0 和 CPC1 控制。这些寄存器用于控制比较器的使能/除能、选择输入路径和失调电压校准等功能。

• CPC0 控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	CEN	COU	CPOL	SC5EN	SC4EN	SC3EN	SC2EN	SC1EN
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **CEN**: 比较器控制位

0: 除能
1: 使能

Bit 6 **COU**: 比较器输出位

CPOL=0: 如果 $V_+ < V_-$, COU=0
如果 $V_+ > V_-$, COU=1
CPOL=1: 如果 $V_+ < V_-$, COU=1
如果 $V_+ > V_-$, COU=0

V_+ : 比较器同相输入, V_- : 比较器反相输入

Bit 5 **CPOL**: 比较器极性选择位

0: 同相
1: 反相

Bit 4 **SC5EN**: 开关 SC5 控制位

0: 关闭
1: 开启

Bit 3 **SC4EN**: 开关 SC4 控制位

0: 关闭
1: 开启

Bit 2 **SC3EN**: 开关 SC3 控制位

0: 关闭
1: 开启

Bit 1 **SC2EN**: 开关 SC2 控制位

0: 关闭
1: 开启

Bit 0 **SC1EN**: 比较器输出路径选择位

0: 比较器输出到 CX 引脚 (必须在 CEN=1 时, CX 引脚才会有效)
1: 比较器输出不到 CX 引脚且仅作单片机内部使用。CX 复用脚用作 I/O 引脚。

• CPC1 控制寄存器

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

Name	HYSON	COFM	CRS	COF4	COF3	COF2	COF1	COF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	1	0	0	0	0

- Bit 7 **HYSON**: 比较器迟滞控制位
 0: 除能
 1: 使能
- Bit 6 **COFM**: 输入失调电压校准模式/比较器模式选择位
 0: 比较器模式
 1: 输入失调电压校准模式
- Bit 5 **CRS**: 比较器输入失调电压校准参考选择位
 0: 选择 CN 作为参考输入脚
 1: 选择 CP 作为参考输入脚
- Bit 4~0 **COF4~COF0**: 比较器输入失调电压校准控制位

2. 14. 3 DAC 寄存器

• DAC0 控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	DA0EN	DA0REF	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

- Bit 7 **DA0EN**: 用于比较器同相输入控制位
 0: 除能
 1: 使能
- Bit 6 **DA0REF**: DAC 参考电压控制位
 0: 来自 Bandgap
 1: 来自 AVDD
- Bit 5~0 **D5~D0**: DACn 数据位

• DAC1 控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	DA1EN	DA1REF	D5	D4	D3	D2	D1	D0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

- Bit 7 **DA1EN**: 用于运算放大器同相输入控制位
 0: 除能
 1: 使能
- Bit 6 **DA1REF**: DAC 参考电压控制位
 0: 来自 Bandgap
 1: 来自 AVDD
- Bit 5~0 **D5~D0**: DACn 数据位

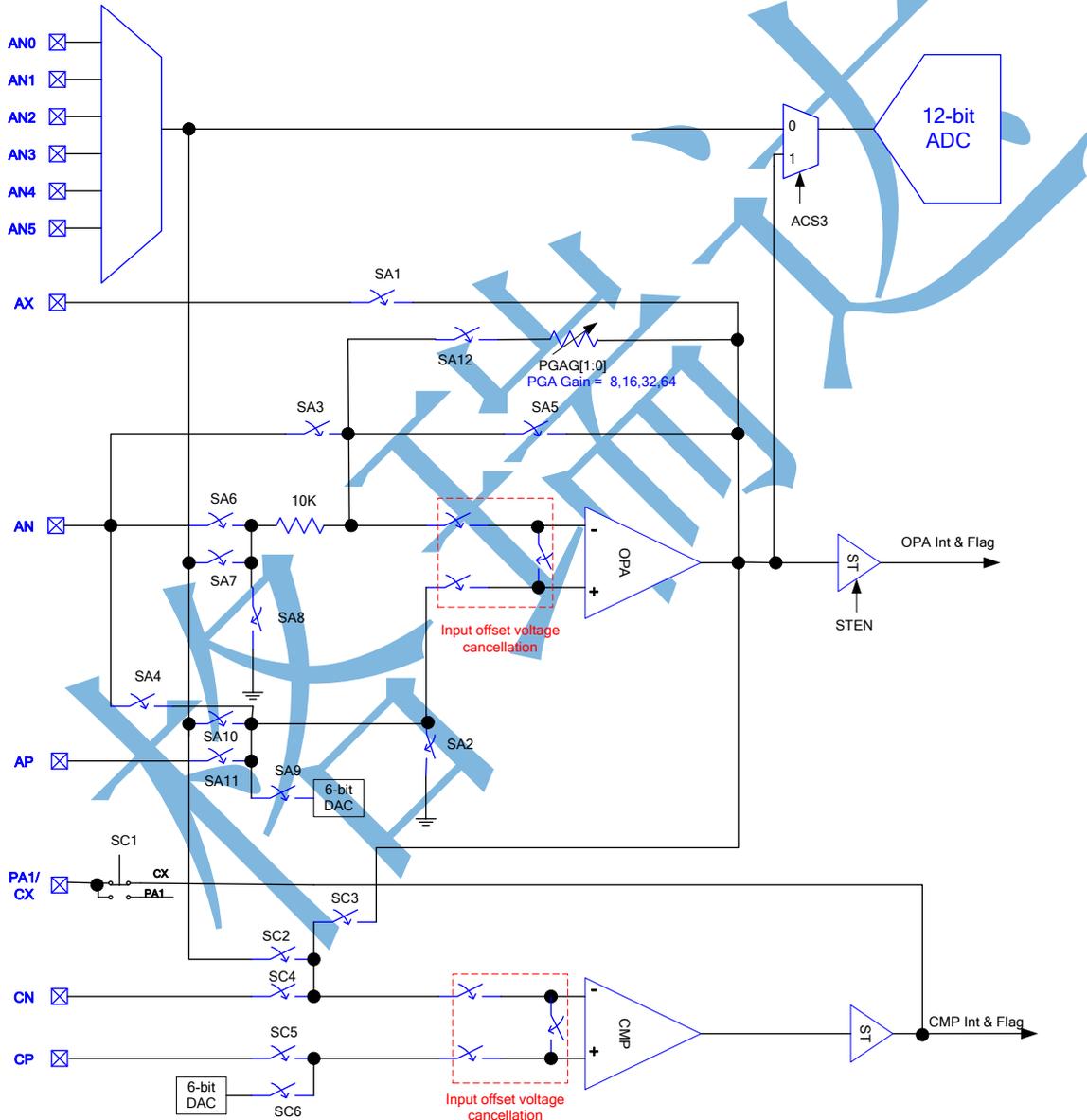
• DAC2 控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	SA11EN	SA10EN	SA9EN	SC6EN
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

Bit 7~4 未定义, 读为“0”

- Bit 3 **SA11EN**: 开关 SA11 控制位
 0: 关闭
 1: 开启
- Bit 2 **SA10EN**: 开关 SA10 控制位
 0: 关闭
 1: 开启
- Bit 1 **SA9EN**: 开关 SA9 控制位
 0: 关闭
 1: 开启
- Bit 0 **SC6EN**: 开关 SC6 控制位
 0: 关闭
 1: 开启

2.14.4 运算放大器&比较器&DAC 结构框图



运算放大器&比较器&DAC 结构框图

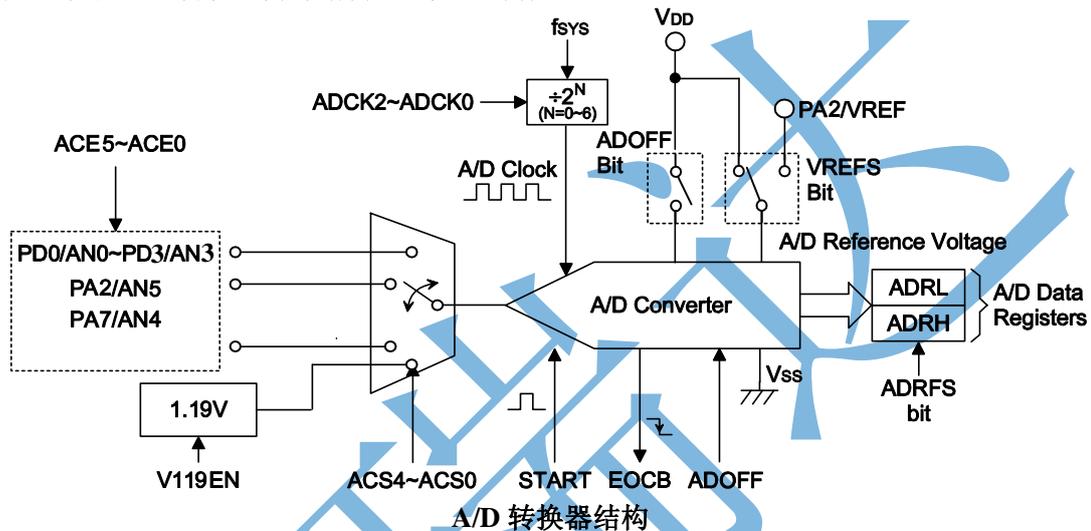
2.15 A/D 转换器

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

2.15.1 A/D 简介

此单片机都包含一个多通道的 A/D 转换器，它们可以直接接入外部模拟信号（来自传感器或其它控制信号）并直接将这此信号转换成 12 位的数字量。

下图显示了 A/D 转换器内部结构和相关的寄存器。



2.15.2 A/D 转换寄存器介绍

A/D 转换器的所有工作由五个寄存器控制。一对只读寄存器来存放 12 位 ADC 数据的值。剩下三个控制寄存器设置 A/D 转换器的操作和控制功能。

寄存器名称	位							
	7	6	5	4	3	2	1	0
ADRL(ADRFs=0)	D3	D2	D1	D0	—	—	—	—
ADRL(ADRFs=1)	D7	D6	D5	D4	D3	D2	D1	D0
ADRH(ADRFs=0)	D11	D10	D9	D8	D7	D6	D5	D4
ADRH(ADRFs=1)	—	—	—	—	D11	D10	D9	D8
ADCR0	START	EOCB	ADOFF	ADRFs	ACS3	ACS2	ACS1	ACS0
ADCR1	ACS4	V119EN	—	VREFS	—	ADCK2	ADCK1	ADCK0
ACERL	—	—	ACE5	ACE4	ACE3	ACE2	ACE1	ACE0

A/D 转换寄存器列表

A/D 转换器数据寄存器 – ADRL，ADRH

对于具有 12 位 A/D 转换器的芯片，需要两个数据寄存器存放转换结果，一个高字节寄存器 ADRH 和一个低字节寄存器 ADRL。在 A/D 转换完毕后，单片机可以直接读取这些寄存器以获得转换结果。由于寄存器只使用了 16 位中的 12 位，其数据存储格式由 ADCR0 寄存器的 ADRFs 位控制，如下表所示。D0~D11 是 A/D 转换数据结果位。未使用的位读为“0”。

ADRFs	ADRH								ADRL							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0

1	0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
---	---	---	---	---	-----	-----	----	----	----	----	----	----	----	----	----	----

A/D 数据寄存器

A/D 转换控制寄存器 – ADCR0, ADCR1, ACERL

寄存器 ADCR0, ADCR1 和 ACERL 用来控制 A/D 转换器的功能和操作。这些 8 位的寄存器定义包括选择连接至内部 A/D 转换器的模拟通道, 数字化数据格式, A/D 时钟源, 并控制和监视 A/D 转换器的开始和转换结束状态。寄存器 ADCR0 的 ACS3~ACS0 位和 ADCR1 的 ACS4 位定义 ADC 输入通道编号。由于每个单片机只包含一个实际的模数转换电路, 因此这 8 个模拟输入中的每一个都需要分别被发送到转换器。ACS4~ACS0 位的功能决定选择哪个模拟输入通道或内部 1.19V 电路是否被连接到内部 A/D 转换器。

ACERL 控制寄存器中的 ACE5~ACE0 位, 用来定义 PA 口中的哪些引脚为 A/D 转换器的模拟输入, 哪些引脚不作为 A/D 转换输入。相应位设为高将选择 A/D 输入功能, 清零将选择 I/O 或其它引脚共用功能。当引脚作为 A/D 输入时, 其原来的 I/O 或其它引脚共用功能消失, 此外, 其内部上拉电阻也将自动断开。

• ADCR0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	START	EOCB	ADOFF	ADRF5	ACS3	ACS2	ACS1	ACS0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	1	1	0	0	0	0	0

- Bit 7 **START:** 启动 A/D 转换位
 0→1→0 : 启动
 0→1 : 重置 A/D 转换, 并且设置 EOCB 为“1”
 此位用于初始化 A/D 转换过程。通常此位为低, 但如果设为高再被清零, 将初始化 A/D 转换过程。当此位为高, 将重置 A/D 转换器。
- Bit 6 **EOCB:** A/D 转换结束标志
 0: A/D 转换结束
 1: A/D 转换中
 此位用于表明 A/D 转换过程的完成。当转换正在进行时, 此位为高。
- Bit 5 **ADOFF:** ADC 模块电源开/关控制位
 0: ADC 模块电源开
 1: ADC 模块电源关
 此位控制 A/D 内部功能的电源。该位被清零将使能 A/D 转换器。如果该位设为高将关闭 A/D 转换器以降低功耗。由于 A/D 转换器在不执行转换动作时都会产生一定的功耗, 所以这在电源敏感的电池应用中需要多加注意。
 注: 1. 建议在进入空闲/休眠模式前, 设置 ADOFF=1 以减少功耗。
 2. ADOFF=1 将关闭 ADC 模块的电源。
- Bit 4 **ADRF5:** ADC 数据格式控制位
 0: ADC 数据高字节是 ADRH 的 bit 7~bit 0, 低字节是 ADRL 的 bit 7~bit 4
 1: ADC 数据高字节是 ADRH 的 bit 3~bit 0, 低字节是 ADRL 的 bit 7~bit 0
 此位控制存放在两个 A/D 数据寄存器中的 12 位 A/D 转换结果的格式。细节方面请参考 A/D 数据寄存器章节。
- Bit 3~0 **ACS3~ACS0:** 选择 A/D 通道(ACS4 为“0”)位
 0000: AN0
 0001: AN1
 0010: AN2
 0011: AN3
 0100: AN4
 0101: AN5
 0110: 保留位
 0111: 保留位
 1xxx: 来自 OPA 输出信号
 这三位是 A/D 通道选择控制位。由于只包含一个内部 A/D 转换电路, 因此通过这些位将 6 个 A/D 输入连接到转换器。如果 ADCR1 寄存器中的 ACS4 设为高, 内部 1.19V 电路将被连接到内部 A/D

转换器。

• **ADCR1 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	ACS4	V119EN	—	VREFS	—	ADCK2	ADCK1	ADCK0
R/W	R/W	R/W	—	R/W	—	R/W	R/W	R/W
POR	0	0	—	0	—	0	0	0

- Bit 7 **ACS4:** 选择内部 1.19V 作为 ADC 输入控制位
 0: 除能
 1: 使能
 此位使能 1.19V 连接到 A/D 转换器。V119EN 位必须先被置位使能 1.19V 电压带隙电路被用于 A/D 转换器。当 ACS4 设为高, 1.19V 带隙电压将连接到 A/D 转换器, 其它 A/D 输入通道断开。
- Bit 6 **V119EN:** 内部 1.19V 控制位
 0: 除能
 1: 使能
 此位控制连接到 A/D 转换器的内部带隙电路开/关功能。当此位设为高, 带隙电压 1.19V 连接至 A/D 转换器。如果 1.19V 未连接至 A/D 转换器且 LVR/LVD 除能, 带隙参考电压电路自动关闭以减少功耗。当 1.19V 打开连接至 A/D 转换器, 在 A/D 转换动作执行前, 带隙电路稳定需一段时间 t_{BG} 。
- Bit 5 未定义, 读为“0”
- Bit 4 **VREFS:** 选择 ADC 参考电压
 0: 内部 ADC 电源
 1: VREF 引脚
 此位用于选择 A/D 转换器的参考电压。如果该位设为高, A/D 转换器参考电压来源于外部 VREF 引脚。如果该位设为低, 内部参考电压来源于电源电压 VDD。
- Bit 3 未定义, 读为“0”
- Bit 2~0 **ADCK2~ADCK0:** 选择 ADC 时钟源
 000: f_{SYS}
 001: $f_{SYS}/2$
 010: $f_{SYS}/4$
 011: $f_{SYS}/8$
 100: $f_{SYS}/16$
 101: $f_{SYS}/32$
 110: $f_{SYS}/64$
 111: 未定义
 这三位用于选择 A/D 转换器的时钟源。

• **ACERL 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	ACE5	ACE4	ACE3	ACE2	ACE1	ACE0
R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	—	1	1	1	1	1	1

- Bit 7~6 未定义, 读为“0”
- Bit 5 **ACE5:** 定义 PA2 是否为 A/D 输入
 0: 不是 A/D 输入
 1: A/D 输入, AN5
- Bit 4 **ACE4:** 定义 PA7 是否为 A/D 输入
 0: 不是 A/D 输入
 1: A/D 输入, AN4
- Bit 3 **ACE3:** 定义 PD3 是否为 A/D 输入
 0: 不是 A/D 输入
 1: A/D 输入, AN3
- Bit 2 **ACE2:** 定义 PD2 是否为 A/D 输入
 0: 不是 A/D 输入
 1: A/D 输入, AN2

- Bit 1 **ACE1**: 定义 PD1 是否为 A/D 输入
 0: 不是 A/D 输入
 1: A/D 输入, AN1
- Bit 0 **ACE0**: 定义 PD0 是否为 A/D 输入
 0: 不是 A/D 输入
 1: A/D 输入, AN0

2. 15.3 A/D 操作

ADCR0 寄存器中的 START 位, 用于打开和复位 A/D 转换器。当单片机设定此位从逻辑低到逻辑高, 然后再到逻辑低, 就会开始一个模数转换周期。当 START 位从逻辑低到逻辑高, 但不再回到逻辑低时, ADCR0 寄存器中的 EOCB 位置“1”, 复位模数转换器。START 位用于控制内部模数转换器的开启动作。

ADCR0 寄存器中的 EOCB 位用于表明模数转换过程的完成。在转换周期结束后, EOCB 位会被单片机自动地置为“0”。此外, 也会置位中断控制寄存器内相应的 A/D 中断请求标志位, 如果中断使能, 就会产生对应的内部中断信号。A/D 内部中断信号将引导程序到相应的 A/D 内部中断入口。如果 A/D 内部中断被禁止, 可以让单片机轮询 ADCR0 寄存器中的 EOCB 位, 检查此位是否被清除, 以作为另一种侦测 A/D 转换周期结束的方法。

A/D 转换器的时钟源为系统时钟 f_{SYS} 分频, 而分频系数由 ADCR1 寄存器中的 ADCK2~ADCK0 位决定。虽然 A/D 时钟源是由系统时钟 f_{SYS} , ADCK2~ADCK0 位决定, 但可选择的最大 A/D 时钟源则有一些限制。允许的 A/D 时钟周期 t_{AD} 的最小值为 $0.5\mu s$, 当系统时钟速度等于或超过 4MHz 时必须小心。如果系统时钟速度为 4MHz 时, ADCK2~ADCK0 位不能设为“000”。必须保证设定的 A/D 转换时钟周期不小于时钟周期的最小值, 否则将会产生不准确的 A/D 转换值。使用者可以参考下面的表格, 被标上星号*的数值是不允许的, 因为它们的 A/D 转换时钟周期小于规定的最小值。

f_{SYS}	A/D 时钟周期 (t_{ADCK})							
	ADCK2, ADCK1, ADCK0 =000 (f_{SYS})	ADCK2, ADCK1, ADCK0 =001 ($f_{SYS}/2$)	ADCK2, ADCK1, ADCK0 =010 ($f_{SYS}/4$)	ADCK2, ADCK1, ADCK0 =011 ($f_{SYS}/8$)	ADCK2, ADCK1, ADCK0 =100 ($f_{SYS}/16$)	ADCK2, ADCK1, ADCK0 =101 ($f_{SYS}/32$)	ADCK2, ADCK1, ADCK0 =110 ($f_{SYS}/64$)	ADCK2, ADCK1, ADCK0 =111
1MHz	1 μs	2 μs	4 μs	8 μs	16 μs	32 μs	64 μs	未定义
2MHz	500ns	1 μs	2 μs	4 μs	8 μs	16 μs	32 μs	未定义
4MHz	250ns*	500ns	1 μs	2 μs	4 μs	8 μs	16 μs	未定义
8MHz	125ns*	250ns*	500ns	1 μs	2 μs	4 μs	8 μs	未定义
12MHz	83ns*	167ns*	333ns*	667ns	1.33 μs	2.67 μs	5.33 μs	未定义

A/D 时钟周期范例

ADCR0 寄存器的 ADOFF 位用于控制 A/D 转换电路电源的开/关。该位必须清零以开启 A/D 转换器电源。即使通过清除 ACERL 寄存器的 ACE5~ACE0 位, 选择无引脚作为 A/D 输入, 如果 ADOFF 设为“0”, 那么仍然会产生功耗。因此当未使用 A/D 转换器功能时, 在功耗敏感的应用中建议设置 ADOFF 为高以减少功耗。

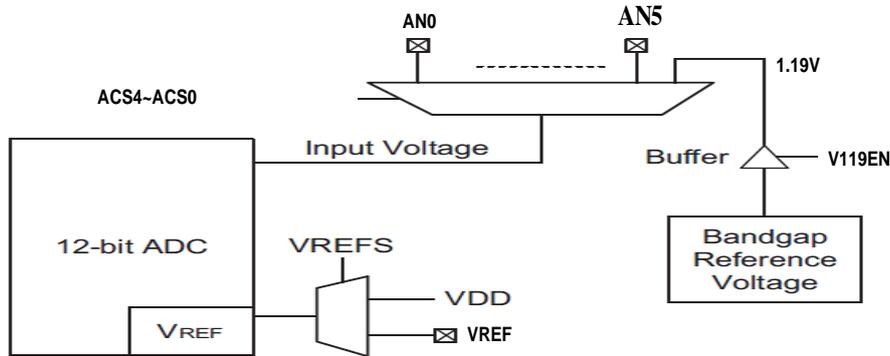
A/D 转换器参考电压来自正电源电压 VDD 或外部参考源引脚 VREF, 可通过 VREFS 位来选择。由于 VREF 引脚与其它功能共用, 当 VREFS 设为高, 选择 VREF 引脚功能且其它引脚功能将自动除能。

2. 15.4 A/D 输入引脚

所有的 A/D 模拟输入引脚都与 PD0~PD3, PA7 及 PA2 端口的 I/O 引脚及其它功能共用。使用 ACERL 寄存器中的 ACE5~ACE0 位, 可以将它们设置为 A/D 转换器模拟输入脚或具有其它功能。如果引脚的对应位 ACE5~ACE0 设为高, 那么该引脚作为 A/D 转换输入且原引脚功能除能。通过这种方式, 引脚的功能可由程序来控制, 灵活地切换引脚功能。如果将引脚设为 A/D 输入, 则通过寄存器编程设置的所有上拉电阻会自动断开。请注意, PAC 端口控制寄存器不需要为使能 A/D 输入而

先设定为输入模式，当 ACE5~ACE0 位使能 A/D 输入时，端口控制寄存器的状态将被重置。

A/D 转换器有自己的参考电压引脚 VREF，而通过设置 ADCR1 寄存器的 VREFS 位，参考电压也可以选择来自电源电压引脚。模拟输入值一定不能超过 VREF 值。



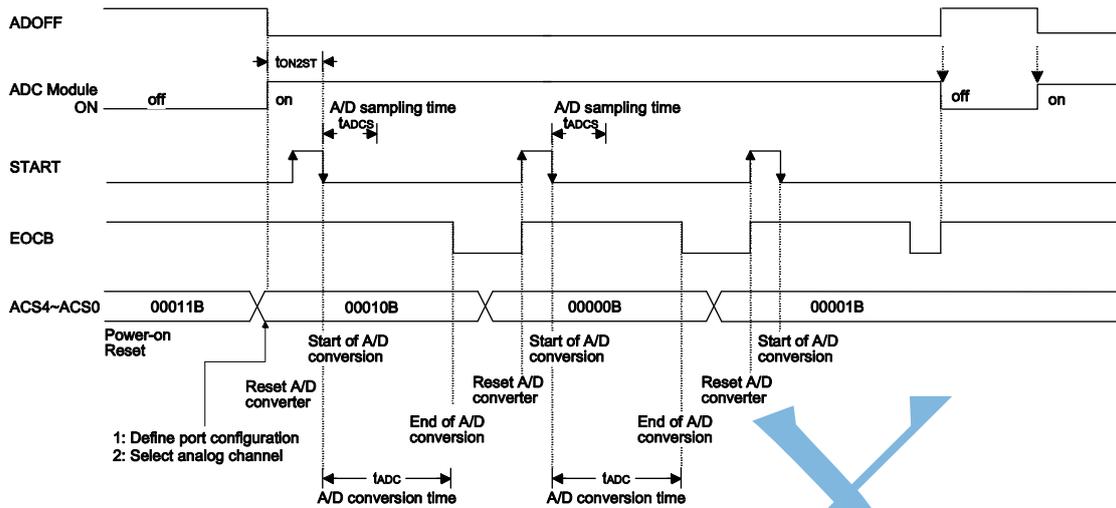
A/D 输入结构

2.15.5 A/D 转换步骤

下面概述实现 A/D 转换过程的各个步骤。

- 步骤 1
通过 ADCR1 寄存器中的 ADCK2~ADCK0 位，选择所需的 A/D 转换时钟。
- 步骤 2
清零 ADCR0 寄存器中的 ADOFF 位使能 A/D。
- 步骤 3
通过 ADCR1 和 ADCR0 寄存器中的 ACS4~ACS0 位，选择连接至内部 A/D 转换器的通道。
- 步骤 4
通过 ACERL 寄存器中的 ACE5~ACE0 位，选择哪些引脚规划为 A/D 输入引脚。
- 步骤 5
如果要使用中断，则中断控制寄存器需要正确地设置，以确保 A/D 转换功能是激活的。总中断控制位 EMI 需要置位为“1”，以及 A/D 转换器中断控制位 ADE 也需要置位为“1”。
- 步骤 6
现在可以通过设定 ADCR0 寄存器中的 START 位从“0”到“1”再回到“0”，开始模数转换的过程。注意，该位需初始化为“0”。
- 步骤 7
可以轮询 ADCR0 寄存器中的 EOCB 位，检查模数转换过程是否完成。当此位成为逻辑低时，表示转换过程已经完成。转换完成后，可读取 A/D 数据寄存器 ADRL 和 ADRH 获得转换后的值。另一种方法是，若中断使能且堆栈未滿，则程序等待 A/D 中断发生。
注：若使用轮询 ADCR0 寄存器中 EOCB 位的状态的方法来检查转换过程是否结束时，则中断使能的步骤可以省略。

下列时序图表示模数转换过程中不同阶段的图形与时序。由应用程序控制开始 A/D 转换过程后，单片机的内部硬件就会开始进行转换，在这个过程中，程序可以继续其它功能。A/D 转换时间为 $16t_{ADCK}$ ， t_{ADCK} 为 A/D 时钟周期。



A/D 转换时序图

2. 15. 6 编程注意事项

在编程时，如果 A/D 转换器未使用，通过设置 ADCR0 寄存器中的 ADOFF 为高，关闭 A/D 内部电路以减少电源功耗。此时，不考虑输入脚的模拟电压，内部 A/D 转换器电路不产生功耗。如果 A/D 转换器输入脚用作普通 I/O 脚，必须特别注意，输入电压为无效逻辑电平也可能增加功耗。

2. 15. 7 A/D 转换功能

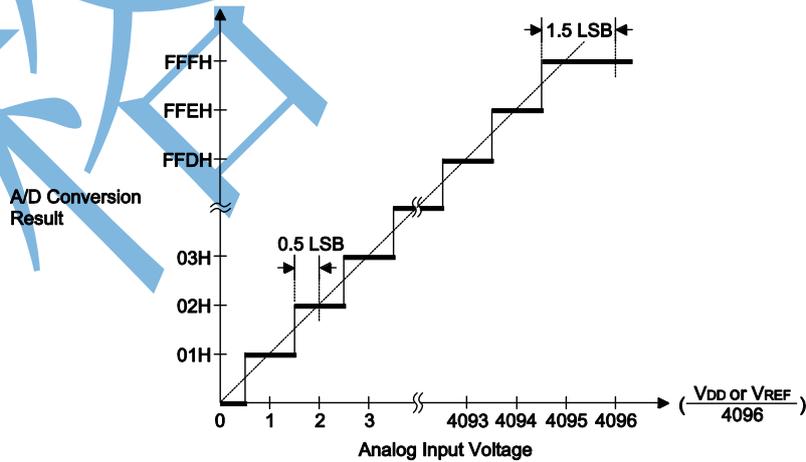
单片机含有一组 12 位的 A/D 转换器，它们转换的最大值可达 FFFH。由于模拟输入最大值等于 V_{DD} 或 V_{REF} 的电压值，因此每一位可表示 V_{DD} 或 $V_{REF}/4096$ 的模拟输入值。

$$1 \text{ LSB} = (V_{DD} \text{ 或 } V_{REF}) \div 4096$$

通过下面的等式可估算 A/D 转换器输入电压值：

$$\text{A/D 输入电压} = \text{A/D 数字输出值} \times (V_{DD} \text{ 或 } V_{REF}) \div 4096$$

下图显示 A/D 转换器模拟输入值和数字输出值之间理想的转换功能。除了数字化数值 0，其后的数字化数值会在精确点之前的 0.5 LSB 处改变，而数字化数值的最大值将在 V_{DD} 或 V_{REF} 之前的 1.5 LSB 处改变。



理想的 A/D 转换功能

2. 15. 8 A/D 转换应用范例

下面两个范例程序用来说明怎样使用 A/D 转换。第一个范例是轮询 ADCR0 寄存器中的 EOCB 来判断 A/D 转换是否完成；第二个范例则使用中断的方式判断。

范例：使用查询 EOCB 的方式来检测转换结束

```

clr  ADE                ; disable ADC interrupt
mov  a,03H
mov  ADCR1,a            ; select fsys/8 as A/D clock and switch off 1.19V
clr  ADOFF
mov  a,0Fh              ; setup ACERL to configure pins AN0~AN5
mov  ACERL,a
mov  a,00h
mov  ADCR0,a            ; enable and connect AN0 channel to A/D converter
:
start_conversion:
clr  START              ; high pulse on start bit to initiate conversion
set  START              ; reset A/D
clr  START              ; start A/D
polling_EOC:
sz   EOCB               ; poll the ADCR0 register EOCB bit to detect end
                        ; of A/D conversion
jmp  polling_EOC       ; continue polling
mov  a,ADRL             ; read low byte conversion result value
mov  ADRL_buffer,a     ; save result to user defined register
mov  a,ADRH            ; read high byte conversion result value
mov  ADRH_buffer,a     ; save result to user defined register
:
:
jmp  start_conversion  ; start next A/D conversion
    
```

范例：使用中断的方式来检测转换结束

```

clr  ADE                ; disable ADC interrupt
mov  a,03H
mov  ADCR1,a            ; select fsys/8 as A/D clock and switch off 1.19V
clr  ADOFF
mov  a,0Fh              ; setup ACERL to configure pins AN0~AN5
mov  ACERL,a
mov  a,00h
mov  ADCR0,a            ; enable and connect AN0 channel to A/D converter
Start_conversion:
clr  START              ; high pulse on start bit to initiate conversion
set  START              ; reset A/D
clr  START              ; start A/D
clr  ADF                ; clear ADC interrupt request flag
set  ADE                ; enable ADC interrupt
set  EMI                ; enable global interrupt
:
:
;ADC interrupt service routine
; ADC interrupt service routine
ADC_ISR:
mov  acc_stack,a       ; save ACC to user defined memory
mov  a,STATUS
mov  status_stack,a   ; save STATUS to user defined memory
:
:
mov  a,ADRL            ; read low byte conversion result value
mov  adr_l_buffer,a   ; save result to user defined register
mov  a,ADRH            ; read high byte conversion result value
mov  adr_h_buffer,a   ; save result to user defined register
    
```

```
:  
:  
EXIT_INT_ISR:  
  mov  a,status_stack  
  mov  STATUS,a          ; restore STATUS from user defined memory  
  mov  a, acc_stack     ; restore ACC from user defined memory  
  reti
```



2.16 中断

中断是单片机一个重要功能。当发生外部中断或内部中断(如触摸动作或定时/计数器溢出), 系统会中止当前的程序, 而转到相对应的中断服务程序中。该单片机提供多个外部中断和内部中断。外部中断由 INT0 和 INT1 引脚信号触发, 而内部中断由定时器模块, 时基, LVD, EEPROM, 触摸按键和 A/D 转换器等控制。

2.16.1 中断寄存器

中断控制基本上是在一定单片机条件发生时设置请求标志位, 应用程序中中断使能位的设置是通过位于专用数据存储器中的一系列寄存器控制的。寄存器的数量由所选单片机的型号决定, 但总的分为三类。第一类是 INTC0~INTC2 寄存器, 用于设置基本的中断; 第二类是 MFIO~MFI4 寄存器, 用于设置多功能中断; 最后一种有 INTEG 寄存器, 用于设置外部中断边沿触发类型。

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

功能	使能位	请求标志	注释
总中断	EMI	—	—
INT 脚	INTnE	INTnF	n=0 或 1
触摸按键模块中断	TKME	TKMF	—
EEPROM 中断	DEE	DEF	—
多功能中断	MFnE	MFnF	n=0~4
时基中断	TBnE	TBnF	n=0 或 1
LVD	LVE	LVF	—
TM	TnPE	TnPF	n=0 或 1 或 2
	TnAE	TnAF	n=0 或 1 或 2
	TnBE	TnBF	n=1
运算放大器中断	OPAE	OPAF	—
比较器中断	CMPE	CMPF	—

中断寄存器位命名模式

2.16.2 中断寄存器内容

Name	Bit							
	7	6	5	4	3	2	1	0
INTEG	—	—	—	—	INT1S1	INT1S0	INT0S1	INT0S0
INTC0	—	INT1F	INT0F	MF4F	INT1E	INT0E	MF4E	EMI
INTC1	—	TB0F	ADF	TKMF	—	TB0E	ADE	TKME
INTC2	MF3F	MF2F	MF1F	MF0F	MF3E	MF2E	MF1E	MF0E
MFIO	—	—	T2AF	T2PF	—	—	T2AE	T2PE
MF11	—	—	T0AF	T0PF	—	—	T0AE	TOPE
MF12	—	T1BF	T1AF	T1PF	—	T1BE	T1AE	T1PE
MF13	DEF	LVF	—	TB1F	DEE	LVE	—	TB1E
MF14	—	—	OPAF	CMPF	—	—	OPAE	CMPE

• INTEG 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	INT1S1	INT1S0	INT0S1	INT0S0
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

- Bit 7~4 未定义，读为“0”
- Bit 3, 2 **INT1S1, INT1S0**: INT1 脚中断边沿控制位
 00: 除能
 01: 上升沿
 10: 下降沿
 11: 双沿
- Bit 1, 0 **INT0S1, INT0S0**: INT0 脚中断边沿控制位
 00: 除能
 01: 上升沿
 10: 下降沿
 11: 双沿

• INTC0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	INT1F	INT0F	MF4F	INT1E	INT0E	MF4E	EMI
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	0	0	0	0	0	0	0

- Bit 7 未定义，读为“0”
- Bit 6 **INT1F**: INT1 脚中断请求标志位
 0: 无请求
 1: 中断请求
- Bit 5 **INT0F**: INT0 中断请求标志位
 0: 无请求
 1: 中断请求
- Bit 4 **MF4F**: 多功能中断 4 请求标志位
 0: 无请求
 1: 中断请求
- Bit 3 **INT1E**: INT1 脚中断控制位
 0: 除能
 1: 使能
- Bit 2 **INT0E**: INT0 脚中断控制位
 0: 除能
 1: 使能
- Bit 1 **MF4E**: 多功能中断 4 控制位
 0: 除能
 1: 使能
- Bit 0 **EMI**: 总中断控制位
 0: 除能
 1: 使能

• INTC1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	TB0F	ADF	TKMF	—	TB0E	ADE	TKME
R/W	—	R/W	R/W	R/W	—	R/W	R/W	R/W
POR	—	0	0	0	—	0	0	0

- Bit 7 未定义，读为“0”
- Bit 6 **TB0F**: 时基 0 中断请求标志位
 0: 无请求
 1: 中断请求
- Bit 5 **ADF**: A/D 转换器中断请求标志位
 0: 无请求
 1: 中断请求
- Bit 4 **TKMF**: 触摸按键模块中断请求标志位
 0: 无请求
 1: 中断请求
- Bit 3 未定义，读为“0”

- Bit 2 **TB0E**: 时基 0 中断控制位
 0: 除能
 1: 使能
- Bit 1 **ADE**: A/D 转换器中断控制位
 0: 除能
 1: 使能
- Bit 0 **TKME**: 触摸按键模块中断控制位
 0: 除能
 1: 使能

• INTC2 寄存器

Bit	7	6	5	4	3	2	1	0
Name	MF3F	MF2F	MF1F	MF0F	MF3E	MF2E	MF1E	MF0E
R/W								
POR	0	0	0	0	0	0	0	0

- Bit 7 **MF3F**: 多功能中断 3 请求标志位
 0: 无请求
 1: 中断请求
- Bit 6 **MF2F**: 多功能中断 2 请求标志位
 0: 无请求
 1: 中断请求
- Bit 5 **MF1F**: 多功能中断 1 请求标志位
 0: 无请求
 1: 中断请求
- Bit 4 **MF0F**: 多功能中断 0 请求标志位
 0: 无请求
 1: 中断请求
- Bit 3 **MF3E**: 多功能中断 3 控制位
 0: 除能
 1: 使能
- Bit 2 **MF2E**: 多功能中断 2 控制位
 0: 除能
 1: 使能
- Bit 1 **MF1E**: 多功能中断 1 控制位
 0: 除能
 1: 使能
- Bit 0 **MF0E**: 多功能中断 0 控制位
 0: 除能
 1: 使能

• MF10 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	T2AF	T2PF	—	—	T2AE	T2PE
R/W	—	—	R/W	R/W	—	—	R/W	R/W
POR	—	—	0	0	—	—	0	0

- Bit 7~6 未定义, 读为“0”
- Bit 5 **T2AF**: TM2 比较器 A 匹配中断请求标志位
 0: 无请求
 1: 中断请求
- Bit 4 **T2PF**: TM2 比较器 P 匹配中断请求标志位
 0: 无请求
 1: 中断请求
- Bit 3~2 未定义, 读为“0”
- Bit 1 **T2AE**: TM2 比较器 A 匹配中断控制位
 0: 除能
 1: 使能
- Bit 0 **T2PE**: TM2 比较器 P 匹配中断控制位

0: 除能
1: 使能

• MF11 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	TOAF	TOPF	—	—	TOAE	TOPE
R/W	—	—	R/W	R/W	—	—	R/W	R/W
POR	—	—	0	0	—	—	0	0

Bit 7~6 未定义，读为“0”
 Bit 5 **TOAF**: TM0 比较器 A 匹配中断请求标志位
 0: 无请求
 1: 中断请求
 Bit 4 **TOPF**: TM0 比较器 P 匹配中断请求标志位
 0: 无请求
 1: 中断请求
 Bit 3~2 未定义，读为“0”
 Bit 1 **TOAE**: TM0 比较器 A 匹配中断控制位
 0: 除能
 1: 使能
 Bit 0 **TOPE**: TM0 比较器 P 匹配中断控制位
 0: 除能
 1: 使能

• MF12 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	T1BF	T1AF	T1PF	—	T1BE	T1AE	T1PE
R/W	—	R/W	R/W	R/W	—	R/W	R/W	R/W
POR	—	0	0	0	—	0	0	0

Bit 7 未定义，读为“0”
 Bit 6 **T1BF**: TM1 比较器 B 匹配中断请求标志位
 0: 无请求
 1: 中断请求
 Bit 5 **T1AF**: TM1 比较器 A 匹配中断请求标志位
 0: 无请求
 1: 中断请求
 Bit 4 **T1PF**: TM1 比较器 P 匹配中断请求标志位
 0: 无请求
 1: 中断请求
 Bit 3 未定义，读为“0”
 Bit 2 **T1BE**: TM1 比较器 B 匹配中断控制位
 0: 除能
 1: 使能
 Bit 1 **T1AE**: TM1 比较器 A 匹配中断控制位
 0: 除能
 1: 使能
 Bit 0 **T1PE**: TM1 比较器 P 匹配中断控制位
 0: 除能
 1: 使能

• MF13 寄存器

Bit	7	6	5	4	3	2	1	0
Name	DEF	LVF	—	TB1F	DEE	LVE	—	TB1E
R/W	R/W	R/W	—	R/W	R/W	R/W	—	R/W
POR	0	0	—	0	0	0	—	0

Bit 7 **DEF**: 数据 EEPROM 中断请求标志位
 0: 无请求

- 1: 中断请求
- Bit 6 **LVF**: LVD 中断请求标志位
0: 无请求
1: 中断请求
- Bit 5 未定义, 读为“0”
- Bit 4 **TB1F**: 时基 1 中断请求标志位
0: 无请求
1: 中断请求
- Bit 3 **DEE**: 数据 EEPROM 中断控制位
0: 除能
1: 使能
- Bit 2 **LVE**: LVD 中断控制位
0: 除能
1: 使能
- Bit 1 未定义, 读为“0”
- Bit 0 **TB1E**: 时基 1 中断控制位
0: 除能
1: 使能

• MF14 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	OPAF	CMPF	—	—	OPAE	CMPE
R/W	—	—	R/W	R/W	—	—	R/W	R/W
POR	—	—	0	0	—	—	0	0

- Bit 7~6 未定义, 读为“0”
- Bit 5 **OPAF**: 运算放大器中断请求标志位
0: 无请求
1: 中断请求
- Bit 4 **CMPF**: 比较器中断请求标志位
0: 无请求
1: 中断请求
- Bit 3~2 未定义, 读为“0”
- Bit 1 **OPAE**: 运算放大器中断控制位
0: 除能
1: 使能
- Bit 1 **CMPE**: 比较器中断控制位
0: 除能
1: 使能

2.16.3 中断操作

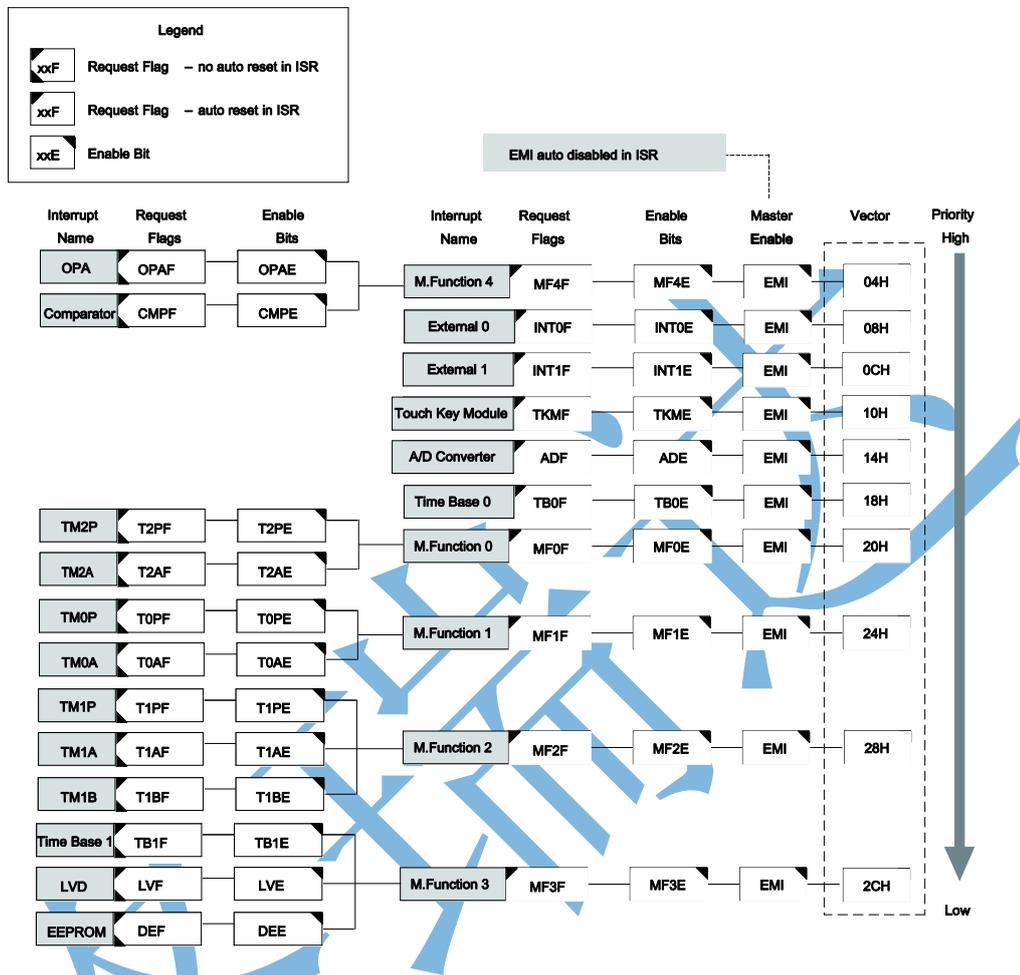
若中断事件条件产生, 如触摸按键计数器溢出、定时/计数器溢出等等, 相关中断请求标志将置起。中断标志产生后程序是否会跳转至相关中断向量执行是由中断使能位的条件决定的。若使能位为“1”, 程序将跳至相关中断向量中执行; 若使能位为“0”, 即使中断请求标志置起中断也不会发生, 程序也不会跳转至相关中断向量执行。若总中断使能位为“0”, 所有中断都将除能。

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

各个中断使能位以及相应的请求标志位, 以优先级的次序显示在下图。一些中断源有自己的向量, 但是有些中断却共用多功能中断向量。一旦中断子程序被响应, 系统将自动清除 EMI 位, 所有其它的中断将被屏蔽, 这个方式可以防止任何进一步的中断嵌套。其它中断请求可能发生在此期间, 虽然中断不会立即响应, 但是中断请求标志位会被记录。

如果某个中断服务子程序正在执行时, 有另一个中断要求立即响应, 那么 EMI 位应在程序进入中断子程序后置位, 以允许此中断嵌套。如果堆栈已满, 即使此中断使能, 中断请求也不会被响应,

直到 SP 减少为止。如果要求立刻动作，则堆栈必须避免成为储满状态。请求同时发生时，执行优先级如下流程图所示。所有被置起的中断请求标志都可把单片机从休眠或空闲模式中唤醒，若要防止唤醒动作发生，在单片机进入休眠或空闲模式前应将相应的标志置起。



中断结构

2.16.4 外部中断

通过 INT0 和 INT1 引脚上的信号变化可控制外部中断。当触发沿选择位设置好触发类型，INT0 和 INT1 引脚的状态发生变化，外部中断请求标志 INT0F 和 INT1F 被置位时外部中断请求产生。若要跳转到相应中断向量地址，总中断控制位 EMI 和相应中断使能位 INT0E 和 INT1E 需先被置位。此外，必须使用 INTEG 寄存器使能外部中断功能并选择触发沿类型。外部中断引脚和普通 I/O 口共用，如果相应寄存器中的中断使能位被置位，此引脚将被作为外部中断脚使用。此时该引脚必须通过设置控制寄存器，将该引脚设置为输入口。当中断使能，堆栈未满并且外部中断脚状态改变，将调用外部中断向量子程序。当响应外部中断服务子程序时，中断请求标志位 INT0F 和 INT1F 会自动复位且 EMI 位会被清零以除能其它中断。注意，即使此引脚被用作外部中断输入，其配置选项中的上拉电阻仍保持有效。

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

2.16.5 A/D 转换器中断

A/D 转换器中断由 A/D 转换动作的结束来控制。当 A/D 转换器中断请求标志被置位，即 A/D

转换过程完成时，中断请求发生。当总中断使能位 EMI 和 A/D 中断使能位 ADE 被置位，允许程序跳转到各自的中断向量地址。当中断使能，堆栈未滿且 A/D 转换动作结束时，将调用它们各自的中断向量子程序。当响应中断服务子程序时，相应的中断请求标志位 ADF 会自动清零。EMI 位也会被清零以除能其它中断。

2.16.6 触摸按键中断

要使触摸按键中断发生，总中断控制位 EMI 和相应的内部中断使能位 TKME 必须先被置位。当所有触摸按键中的时隙计数器溢出（或触摸按键 CCRA 比较匹配中断），相应的中断请求标志位 TKMF 将置位并触发触摸按键中断。中断使能，堆栈未滿，当触摸按键时隙计数器溢出发生中断时，将调用位于触摸按键中断向量处的子程序。当响应中断服务子程序时，中断请求标志位 TKMF 会被自动复位且 EMI 位会被清零以除能其它中断。

2.16.7 多功能中断

此单片机中有五个多功能中断，与其它中断不同，它没有独立源，但由其它现有的中断源构成，即触运算放大器模块，比较器模块，定时器模块，低电压检测，EEPROM 和时基中断源。

当多功能中断中任何一个中断请求标志 MF_nF 被置位，多功能中断请求产生。当中断使能，堆栈未滿，包括在多功能中断中的任意一个中断发生时，将调用多功能中断向量中的一个子程序。当响应中断服务子程序时，相关的多功能请求标志位会自动复位且 EMI 位会自动清零以除能其它中断。

但必须注意的是，在中断响应时，虽然多功能中断标志会自动复位，但多功能中断源的请求标志位不会自动复位，必须由应用程序清零。

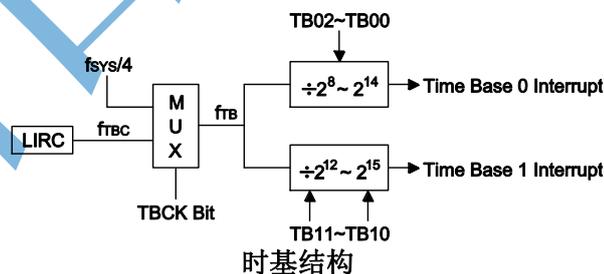
2.16.8 时基中断

时基中断提供一个固定周期的中断信号，由定时器功能产生溢出信号控制。该单片机包含两个时基，即 TB0 和 TB1。TB0 中断是独立中断，TB1 中断包含在多功能中断中。

对于 TB0 中断，当中断请求标志 TB0F 被置位时，中断请求发生。当中断使能位 EMI 和时基中断使能位 TB0E 被置位，堆栈未滿且 TB0 溢出时，将调用其中断向量子程序。当响应中断服务子程序时，相应中断请求标志位 TB0F 将自动复位且 EMI 位会被清零以除能其它中断。

对于 TB1 中断，当中断请求标志 TB1F 被置位时，中断请求发生。当中断使能标志位 EMI、相应多功能中断使能位 MF_nE 和时基中断使能位 TB1E 被置位，堆栈未滿且 TB1 溢出，将调用相应多功能中断向量中 TB2 中断子程序。当响应中断服务子程序时，相应多功能中断请求标志 MF_nF 将自动复位且 EMI 被清零以除能其它中断。注意，TB1F 需通过手动复位。

时基中断的目的是提供一个固定周期的中断信号，时钟源来自内部时钟源 f_{TB}。f_{TB} 输入时钟首先经过分频器，分频率由程序设置 TBC 寄存器相关位获取合适的分频值以提供更长的时基中断周期。控制时基中断频率 f_{TB} 的时钟源有几种，如在系统工作模式章节所示。



• TBC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	TBON	TBCK	TB11	TB10	—	TB02	TB01	TB00
R/W	R/W	R/W	R/W	R/W	—	R/W	R/W	R/W
POR	0	0	1	1	—	1	1	1

Bit 7 **TBON:** TB0 和 TB1 控制位
0: 除能

	1: 使能
Bit 6	TBCK : 选择 f_{TB} 时钟位
	0: f_{TBC}
	1: $f_{SYS}/4$
Bit 5, 4	TB11, TB10 : 选择时基 1 溢出周期位
	00: $4096/f_{TB}$
	01: $8192/f_{TB}$
	10: $16384/f_{TB}$
	11: $32768/f_{TB}$
Bit 3	未定义, 读为 “0”
Bit 2~0	TB02~TB00 : 选择时基 0 溢出周期位
	000: $128/f_{TB}$
	001: $256/f_{TB}$
	010: $512/f_{TB}$
	011: $1024/f_{TB}$
	100: $2048/f_{TB}$
	101: $4096/f_{TB}$
	110: $8192/f_{TB}$
	111: $16384/f_{TB}$

2.16.9 LVD 中断

LVD 中断也属于多功能中断。当低电压检测功能检测到一个低电压时, LVD 中断请求标志 LVF 被置位, LVD 中断请求产生。若要程序跳转到相应中断向量地址, 总中断控制位 EMI、低电压中断使能位 LVE 和相应多功能中断使能位需先被置位。当中断使能, 堆栈未满足且低电压条件发生时, 可跳转至相关多功能中断向量子程序中执行。当低电压中断响应, EMI 将被自动清零以除能其它中断, 多功能中断请求标志也可自动清除, 但 LVF 标志需在应用程序中手动清除。

2.16.10 TM 中断

简易型 TM 有两个中断, 增强型 TM 有三个中断。所有的 TM 中断也属于多功能中断。简易型 TM 有两个中断请求标志位 TnPF、TnAF 及两个使能位 TnPE、TnAE。增强型 TM 有三个中断请求标志 TnPF、TnAF、TnBF 及三个使能位 TnPE、TnAE、TnBE。当 TM 比较器 P、A、B 匹配情况发生时, 任意 TM 中断请求标志被置位, TM 中断请求产生。

若要程序跳转到相应中断向量地址, 总中断控制位 EMI、相应 TM 中断使能位和相关多功能中断使能位 MFnE 需先被置位。当中断使能, 堆栈未满足且 TM 比较器匹配情况发生时, 可跳转至相关多功能中断向量子程序中执行。当 TM 中断响应, EMI 将被自动清零以除能其它中断, 相关 MFnF 标志也可自动清除, 但 TM 中断请求标志需在应用程序中手动清除。

2.16.11 EEPROM 中断

EEPROM 中断也属于多功能中断。当写周期结束, EEPROM 中断请求标志 DEF 被置位, EEPROM 中断请求产生。若要程序跳转到相应中断向量地址, 总中断控制位 EMI、MF3E、EEPROM 中断使能位 DEE 需先被置位。当中断使能, 堆栈未满足且 EEPROM 写周期结束时, 可跳转至相关中断向量子程序中执行。当 EEPROM 中断响应, EMI 将被自动清零以除能其它中断, 但 DEF 标志需在应用程序中手动清零。

2.16.12 运算放大器中断

运算放大器中断也属于多功能中断。此运算放大器也可通过开关的切换而当成比较器, 当其输出状态改变, 运算放大器中断请求标志 OPAF 被置位, 运算放大器中断请求产生。若要跳转到相应中断向量地址, 总中断控制位 EMI、多功能中断使能位 MFnE 和运算放大器中断使能位 OPAE 需先被置位。当中断使能, 堆栈未满足并且比较器输入产生一个运算放大器输出变化时, 将调用相应多功能中断向量中运算放大器中断子程序。当响应中断服务子程序时, 多功能中断请求标志位 MFnF 会自动复位且 EMI 位会被清零以除能其它中断, 运算放大器中断请求标志位需手动复位。

2.16.13 比较器中断

比较器中断也属于多功能。当比较器输出状态改变, 比较器中断请求标志 CMPF 被置位, 比较器中断请求产生。若要跳转到相应中断向量地址, 总中断控制位 EMI、多功能中断使能位 MFnE 和比较器中断使能位 CMPE 需先被置位。当中断使能, 堆栈未满足并且比较器输入产生一个比较器输出

变化时，将调用相应多功能中断向量中运算放大器中断子程序。当响应中断服务子程序时，多功能中断请求标志位 MF_nF 会自动复位且 EMI 位会被清零以除能其它中断，运算放大器中断请求标志位需手动复位。

2.16.14 中断唤醒功能

每个中断都具有将处于休眠或空闲模式的单片机唤醒的能力。当中断请求标志由低到高转换时唤醒动作产生，其与中断是否使能无关。因此，尽管单片机处于休眠或空闲模式且系统振荡器停止工作，如有外部中断脚上产生外部边沿跳变，低电压改变都可能导致其相应的中断标志被置位，由此产生中断，因此必须注意避免伪唤醒情况的发生。若中断唤醒功能被除能，单片机进入休眠或空闲模式前相应中断请求标志应被置起。中断唤醒功能不受中断使能位的影响。

2.16.15 编程注意事项

通过禁止相关中断使能位，可以屏蔽中断请求，然而，一旦中断请求标志位被设定，它们会被保留在中断控制寄存器内，直到相应的中断服务子程序执行或请求标志位被软件指令清除。

多功能中断中所含中断相应程序执行时，多功能中断请求标志 MF_nF 可以自动清零，但各自的请求标志需在应用程序中手动清除。

建议在中断服务子程序中不要使用“CALL 子程序”指令。中断通常发生在不可预料的情况或是需要立刻执行的某些应用。假如只剩下一层堆栈且没有控制好中断，当“CALL 子程序”在中断服务子程序中执行时，将破坏原来的控制序列。

所有中断在休眠或空闲模式下都具有唤醒功能，当中断请求标志发生由高到低的转变时都可产生唤醒功能。若要避免相应中断产生唤醒动作，在单片机进入休眠或空闲模式前需先将相应请求标志置为高。

当进入中断服务程序，系统仅将程序计数器的内容压入堆栈，如果中断服务程序会改变状态寄存器或其它的寄存器的内容而破坏控制流程，应事先将这些数据保存起来。

若从中断子程序中返回可执行 RET 或 RETI 指令。除了能返回至主程序外，RETI 指令还能自动设置 EMI 位为高，允许进一步中断。RET 指令只能返回至主程序，清除 EMI 位，除能进一步中断。

2.17 低电压检测-LVD

此单片机都具有低电压检测功能，即 LVD。该功能使能用于监测电源电压 V_{DD} ，若电源电压低于一定值可提供一个警告信号。此功能在电池类产品中非常有用，在电池电压较低时产生警告信号。低电压检测也可产生中断信号。

2.17.1 LVD 寄存器

低电压检测功能由 LVDC 寄存器控制。VLVD2~VLVD0 位用于选择 8 个固定的电压参考点。LVDO 位被置位时低电压情况发生，若 LVDO 位为低，表明 V_{DD} 电压工作在当前所设置低电压水平值之上。LVDEN 位用于控制低电压检测功能的开启/关闭，设置此位为高使能此功能，反之，关闭内部低电压检测电路。低电压检测会有一些的功耗，在不使用时可考虑关闭此功能，此举在功耗要求严格的电池供电应用中值得考虑。

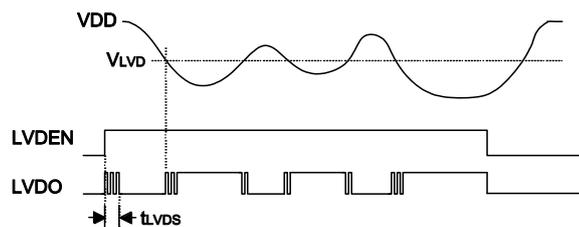
2.17.2 LVDC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	LVDO	LVDEN	—	VLVD2	VLVD1	VLVD0
R/W	—	—	R	R/W	—	R/W	R/W	R/W
POR	—	—	0	0	—	0	0	0

- Bit 7, 6 未定义，读为“0”
- Bit 5 **LVDO**: LVD 输出标志位
0: 未检测到低电压
1: 检测到低电压
- Bit 4 **LVDEN**: 低电压检测控制位
0: 除能
1: 使能
- Bit 3 未定义，读为“0”
- Bit 2~0 **VLVD2~VLVD0**: 选择 LVD 电压位
000: 2.0V
001: 2.2V
010: 2.4V
011: 2.7V
100: 3.0V
101: 3.3V
110: 3.6V
111: 4.0V

2.17.3 LVD 操作

通过比较电源电压 V_{DD} 与存储在 LVDC 寄存器中的预置电压值的结果，低电压检测功能工作。其设置的范围为 2.0V~4.0V。当电源电压 V_{DD} 低于预置电压值时，LVDO 位被置为高，表明低电压产生。低电压检测功能由一个自动使能的参考电压提供。若 LVDEN 位为高，当单片机掉电时低电压检测器保持有效状态。低电压检测器使能后，读取 LVDO 位前，电路稳定需要一定的延时 t_{LVDS} 。注意， V_{DD} 电压可能上升或下降比较缓慢，在 V_{LVD} 电压值附近时，LVDO 位可能有多种变化。



LVD 操作

低电压检测器也有自己的中断功能，也是属于多功能中断的一种，它是除了轮询 LVDO 位之外

的另一种检测低电压的方法。中断条件产生置位 LVDO 并延时 t_{LVD} 后，中断产生。若 LVDEN 位为高，当单片机掉电时低电压检测器保持有效状态。此种情况下，若 V_{DD} 降至小于 LVD 预置电压值时，中断请求标志位 LVF 将被置位，中断产生，单片机将被从休眠或空闲模式中唤醒。若不要求低电压检测的唤醒功能使能，在单片机进入休眠或空闲模式前应将 LVF 标志置为高。



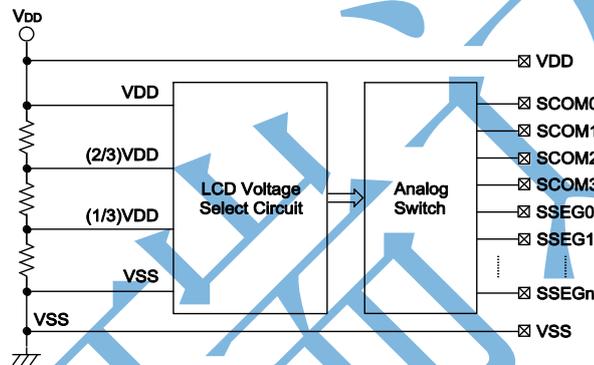
2.18 带 SCOM 和 SSEG 功能的 LCD 驱动器

单片机具有驱动外部 LCD 面板的能力。LCD 控制信号(COM & SEG)可以在 I/O 引脚产生，由软件编程实现。

2.18.1 LCD 操作

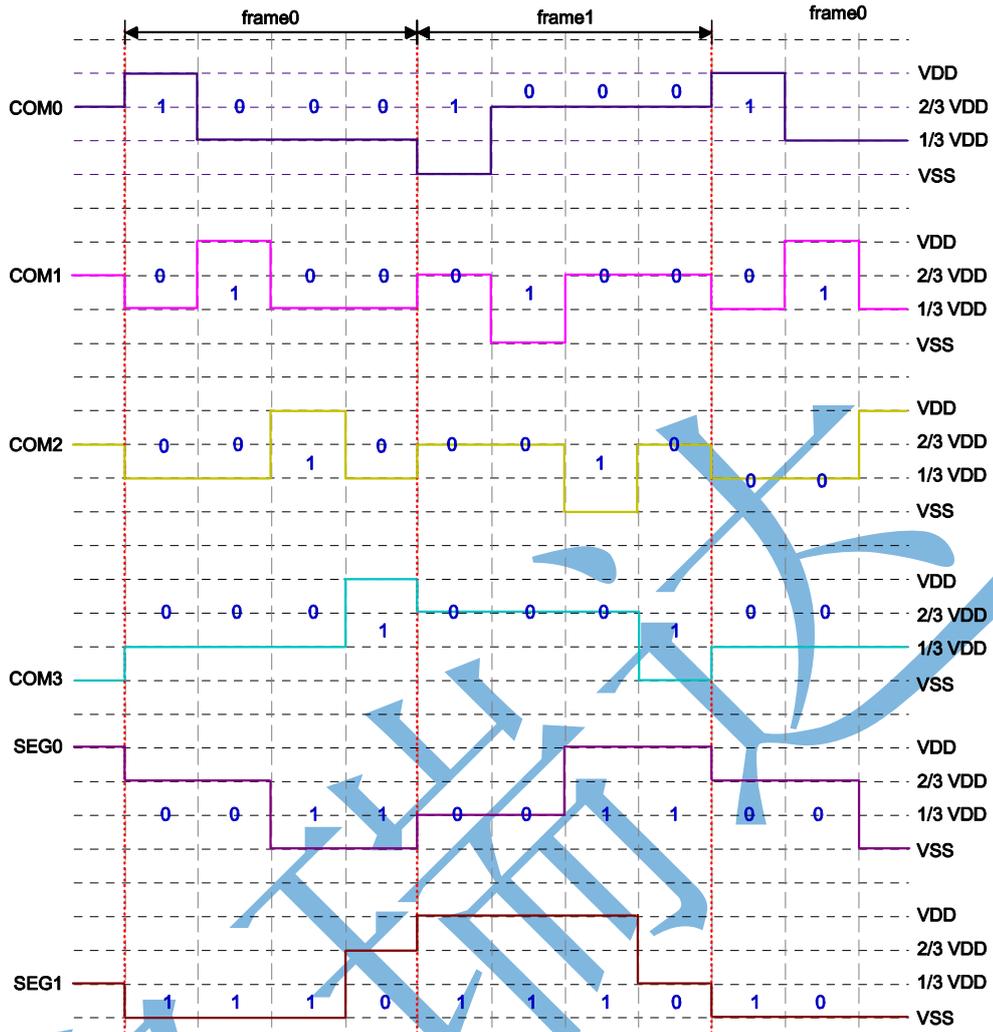
LCD 驱动的 COM 脚 SCOM0~SCOM3 和 SEG 脚 SSEG0~SSEGN 与 I/O 口共用。LCD 驱动功能由一系列的 LCD 控制寄存器来控制，另外，该寄存器可设置 LCD 的开启和关闭以及输出偏压值等功能，使得 COM 和 SEG 驱动器输出 V_{SS} ， $(1/3)V_{DD}$ ， $(2/3)V_{DD}$ 和 V_{DD} 电压，从而实现 1/3 bias LCD 的显示。

LCD 控制寄存器中的 SLCDEN 位是 LCD 驱动的主控制位，它与 COMnEN 和 SEGNEN 位搭配共同选择哪些 I/O 端口用于 LCD 驱动。需注意的是，端口控制寄存器不需要设置为输出以使能 LCD 驱动操作。



LCD 驱动器结构图

下图显示的是通过应用程序来产生的 1/3 bias LCD 波形图。需注意的是，下图用“1”来描述照明 LCD 的像素。LCD 信号极性由 I/O 口寄存器 PB 的 PB0~PB3 位来控制，在引脚 SCOM0~SCOM3 上输出为“0”或“1”。



注：图中的逻辑值为 I/O 口寄存器 PB 的值，即 PB0~PB3。

1/3 bias LCD 波形图

一个周期的 LCD 波形包括两个帧，即帧 0 和帧 1，其功能如下。

帧 0

选择帧 0，FRAME 位清零。

在帧 0，COM 信号可以输出 V_{DD} 或 1/3 bias V_{DD}。SEG 信号可以输出 V_{SS} 或 2/3 bias V_{DD}。

帧 1

在帧 1，COM 信号可以输出 V_{SS} 或 2/3 bias V_{DD}。SEG 信号可以输出 V_{DD} 或 1/3 bias V_{DD}。

COM0~COMn 的波形图通过应用程序由 FRAME 位控制，每个 COM 脚由 I/O 口寄存器决定输出 V_{DD}，V_{SS} 或者偏压。SEG0~SEGm 的波形图也是由 FRAME 位控制，每个 SEG 脚由 I/O 口寄存器决定输出 V_{DD}，V_{SS} 或者偏压。

2.18.2 LCD 偏压控制

LCD COM 和 SEG 驱动器可以提供多种驱动电流选择以适应不同 LCD 面板的需求。通过设置 LCD 控制寄存器中 ISEL0 位和 ISEL1 位可以配置不同的偏压电阻。

2.18.3 LCD 驱动寄存器

• **SLCDC0 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	FRAME	ISEL1	ISEL0	SLCDEN	COM3EN	COM2EN	COM1EN	COM0EN

R/W								
POR	0	0	0	0	0	0	0	0

- Bit 7 **FRAME**: 选择当前帧（帧 0 或帧 1）输出位
 0: 帧 0
 1: 帧 1
- Bit 6, 5 **ISEL1, ISEL0**: 选择 SCOM 和 SSEG 工作电流位($V_{DD}=5V$)
 00: 25 μ A
 01: 50 μ A
 10: 100 μ A
 11: 200 μ A
- Bit 4 **SLCDEN**: SCOM 和 SSEG 模块开/关控制位
 0: 除能
 1: 使能
 当 SLCDEN=1 时, SCOMn 和 SSEGm 口可以通过 COMnEN 和 SEGmEN 位使能。当 SLCDEN=0 时, SCOMn 和 SSEGm 输出电平将保持在 V_{DD} 。
- Bit 3 **COM3EN**: 选择 SCOM3 或其它功能位
 0: 其它功能
 1: SCOM3
- Bit 2 **COM2EN**: 选择 SCOM2 或其它功能位
 0: 其它功能
 1: SCOM2
- Bit 1 **COM1EN**: 选择 SCOM1 或其它功能位
 0: 其它功能
 1: SCOM1
- Bit 0 **COM0EN**: 选择 SCOM0 或其它功能位
 0: 其它功能
 1: SCOM0

• SLCDC1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	SEG7EN	SEG6EN	SEG5EN	SEG4EN	SEG3EN	SEG2EN	SEG1EN	SEG0EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

- Bit 7~0 **SEG7EN~SEG0EN**: 选择 SSEG7~SSEG0 或其它功能位
 0: 其它功能
 1: SSEG7~SSEG0

• SLCDC2 寄存器

Bit	7	6	5	4	3	2	1	0
Name	SEG15EN	SEG14EN	SEG13EN	SEG12EN	SEG11EN	SEG10EN	SEG9EN	SEG8EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

- Bit 7~0 **SEG15EN~SEG8EN**: 选择 SSEG15~SSEG8 或其它功能位
 0: 其它功能
 1: SSEG15~SSEG8

2.18.4 LED 驱动器

该单片机具有 LED 驱动功能，通过输出大电流驱动外部 LED 设备。

LED 驱动操作

根据不同型号的单片 LED 大电流输出驱动引脚不同。

单片机型号	LED 驱动引脚
-------	----------

BS45F16	PB0~PB7 (大灌电流)
---------	----------------

通过 SLEDC 寄存器来选择为普通灌电流或大灌电流驱动。

2.18.5 LED 驱动寄存器

• SLEDC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0

D7~D0: 选择 PB7~PB0 引脚输出灌电流位

0: 正常输出灌电流

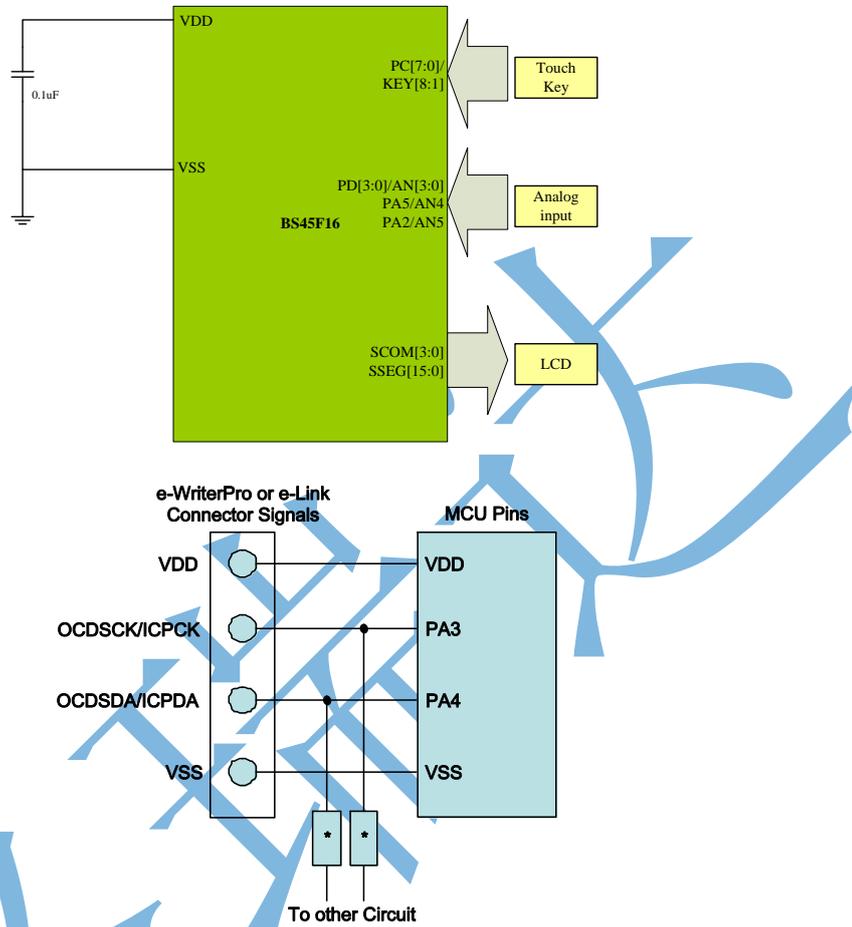
1: 2 倍输出灌电流

配置选项

No.	Options
1	HIRC frequency selection: 1. 8 MHz 2. 12 MHz
2	WDT: Always Enable or By S/W Control

Note: 1. HIRC 除了在配置选项选择 HIRC 频率之外, 也必须在寄存器 (HIRCS1 和 HIRCS0) 中选择相同的频率。
 2. WDT 在配置选项选择 Always Enable, 建议 WE [4:0] 也设成 01010B。

2.19 应用电路



注：“*”表示电阻或电容，且电阻值必须大于1kΩ，电容值必须小于1nF。

第 3 章 指令集

3.1 简介

任何单片机成功运作的核心在于它的指令集,此指令为一组程序指令码,用来指导单片机如何去执行指定的工作。在格瑞达单片机中,提供了丰富且灵活的指令集,共超过 60 条,程序设计师可以事半功倍地实现它们的应用。

为了更容易的了解各式各样的指令码,接下来按功能分组介绍它们。

3.2 指令周期

大部分的操作均只需要一个指令周期来执行。分支、调用或查表则需要两个指令周期。一个指令周期相当于四个系统时钟周期,因此如果在 8MHz 的系统时钟振荡器下,大部分的操作将在 0.5 μ s 中执行完成,而分支或调用操作则将在 1 μ s 中执行完成。虽然需要两个指令周期的指令通常指的是 JMP、CALL、RET、RETI 和查表指令,但如果牵涉到程序计数器低字节寄存器 PCL 也将多花费一个周期去加以执行。即指令改变 PCL 的内容进而导致直接跳转至新地址时,需要多一个周期去执行。例如“CLR PCL”或“MOV PCL, A”。对于跳转命令必须注意,如果比较的结果牵涉到跳转动作将多花费一个周期,如果没有则需一个周期即可。

3.3 数据的传送

单片机程序的数据传送是使用最为频繁的操作之一。使用三种 MOV 的指令,数据不但可以从寄存器转移至累加器(反之亦然),而且能够直接移动立即数到累加器。数据传送最重要的应用之一是从接收端口接收数据或传送数据到输出端口。

3.4 算术运算

算术运算和数据处理是大部分单片机应用所需具备的能力,在格瑞达单片机内部的指令集中,可直接实现加与减的运算。当加法的结果超出 255 或减法的结果少于 0 时,要注意正确的处理进位和借位的问题。INC、INCA、DEC 和 DECA 指令提供了对一个指定地址的值加一或减一的功能。

3.5 逻辑和移位运算

标准逻辑运算例如 AND、OR、XOR 和 CPL 全都包含在格瑞达单片机内部的指令集中,如同大多数牵涉到数据运算的指令,数据的传送必须通过累加器。在所有逻辑数据运算中,如果运算结果为零,则零标志位将被置位。另外逻辑数据运用形式还有移位指令,例如 RR、RL、RRC 和 RLC 提供了向左或向右移动一位的方法。移位指令常用于串行端口的程序应用,数据可从内部寄存器转移至进位标志位,而此位则可被检验。移位运算还可应用在乘法与除法的运算组成中。

3.6 分支和控制的转换

程序分支是采取使用 JMP 指令跳转到指定地址或使用 CALL 指令调用子程序的形式。两者之不同在于当子程序被执行完毕后,程序必须马上返回原来的地址。这个动作是由放置在子程序里的返回指令 RET 来实现,它可使程序跳回 CALL 指令之后的地址。在 JMP 指令中,程序则只是跳到一个指定的地址而已,并不需如 CALL 指令跳回。一个非常有用的分支指令是条件跳转,条件是由数据存储器或指定位来加以决定。遵循跳转条件,程序将继续执行下一条指令或略过且跳转至接下来的指令。这些分支指令是程序走向的关键,跳转条件可能是外部开关输入,或是内部数据位的值。

3.7 位运算

提供数据存储器单一位的运算指令是格瑞达单片机的特性之一,这特性对于输出端口位的规划尤其有用,其中个别的位或端口的引脚可以使用“SET [m].i”或“CLR [m].i”指令来设定其为高位或低位。如果没有这特性,程序设计师必须先读入输入输出的 8 位数据,处理这些数据,然后再输出正确的

新数据。这种读入-修改-写出的程序现在则由位指令所取代。

3.8 查表运算

数据的储存通常由寄存器完成，然而当处理大量的数据时，其庞大与复杂的内容常造成对指定存储器储存上的不便，为了改善此问题，格瑞达单片机允许在程序存储器中设定一块数据可直接存取的区域，只需要一组简明的指令即可对数据进行查表。

3.9 其它运算

除了上述功能指令外，其它指令还包括用于省电的“HALT”指令和使程序在极端电压或电磁环境下仍能正常工作的看门狗定时器控制指令。这些指令的使用则请查阅相关的章节。

3.10 指令设定一览表

下表中说明了按功能分类的指令集，用户可以将该表作为基本的指令参考。

惯例

x: 立即数

m: 数据存储器地址

A: 累加器

i: 第0~7 位

addr: 程序存储器地址

助记符	说明	周期	影响标志位
算术运算			
ADD A,[m]	ACC 与数据存储器相加，结果放入 ACC	1	
ADDM A,[m]	ACC 与数据存储器相加，结果放入数据存储器	1 ^注	Z,C,AC,OV
ADD A,x	ACC 与立即数相加，结果放入 ACC	1	Z,C,AC,OV
ADC A,[m]	ACC 与数据存储器、进位标志相加，结果放入 ACC	1	Z,C,AC,OV
ADCM A,[m]	ACC 与数据存储器、进位标志相加，结果放入数据存储器	1 ^注	Z,C,AC,OV
SUB A,x	ACC 与立即数相减，结果放入 ACC	1	Z,C,AC,OV
SUB A,[m]	ACC 与数据存储器相减，结果放入 ACC	1	Z,C,AC,OV
SUBM A,[m]	ACC 与数据存储器相减，结果放入数据存储器	1 ^注	Z,C,AC,OV
SBC A,[m]	ACC 与数据存储器、进位标志相减，结果放入 ACC	1	Z,C,AC,OV
SBCM A,[m]	ACC 与数据存储器、进位标志相减，结果放入数据存储器	1 ^注	Z,C,AC,OV
DAA [m]	将加法运算中放入 ACC 的值调整为十进制数，并将结果放入数据存储器	1 ^注	Z,C,AC,OV C
逻辑运算			
AND A,[m]	ACC 与数据存储器做“与”运算，结果放入 ACC	1	Z
OR A,[m]	ACC 与数据存储器做“或”运算，结果放入 ACC	1	Z
XOR A,[m]	ACC 与数据存储器做“异或”运算，结果放入 ACC	1	Z
ANDM A,[m]	ACC 与数据存储器做“与”运算，结果放入数据存储器	1 ^注	Z
ORM A,[m]	ACC 与数据存储器做“或”运算，结果放入数据存储器	1 ^注	Z
XORM A,[m]	ACC 与数据存储器做“异或”运算，结果放入数据存储器	1 ^注	Z
AND A,x	ACC 与立即数做“与”运算，结果放入 ACC	1	Z
OR A,x	ACC 与立即数做“或”运算，结果放入 ACC	1	Z
XOR A,x	ACC 与立即数做“异或”运算，结果放入 ACC	1	Z
CPL [m]	对数据存储器取反，结果放入数据存储器	1 ^注	Z
CPLA [m]	对数据存储器取反，结果放入 ACC	1	Z
递增和递减			
INCA [m]	递增数据存储器，结果放入 ACC	1	Z
INC [m]	递增数据存储器，结果放入数据存储器	1 ^注	Z
DECA [m]	递减数据存储器，结果放入 ACC	1	Z
DEC [m]	递减数据存储器，结果放入数据存储器	1 ^注	Z
移位			

RRA	[m]	数据存储器右移一位, 结果放入 ACC	1	无
RR	[m]	数据存储器右移一位, 结果放入数据存储器	1 ^注	无
RRCA	[m]	带进位将数据存储器右移一位, 结果放入 ACC	1	C
RRC	[m]	带进位将数据存储器右移一位, 结果放入数据存储器	1 ^注	C
RLA	[m]	数据存储器左移一位, 结果放入 ACC	1	无
RL	[m]	数据存储器左移一位, 结果放入数据存储器	1 ^注	无
RLCA	[m]	带进位将数据存储器左移一位, 结果放入 ACC	1	C
RLC	[m]	带进位将数据存储器左移一位, 结果放入数据存储器	1 ^注	C

助记符		说明	周期	影响标志位
数据传送				
MOV	A,[m]	将数据存储器送至 ACC	1	无
MOV	[m],A	将 ACC 送至数据存储器	1 ^注	无
MOV	A,x	将立即数送至 ACC	1	无
位运算				
CLR	[m].i	清除数据存储器的位	1 ^注	无
SET	[m].i	置位数据存储器的位	1 ^注	无
转移				
JMP	addr	无条件跳转	2	无
SZ	[m]	如果数据存储器为零, 则跳过下一条指令	1 ^注	无
SZA	[m]	数据存储器送至 ACC, 如果内容为零, 则跳过下一条指令	1 ^注	无
SZ	[m].i	如果数据存储器的第 i 位为零, 则跳过下一条指令	1 ^注	无
SNZ	[m].i	如果数据存储器的第 i 位不为零, 则跳过下一条指令	1 ^注	无
SIZ	[m]	递增数据存储器, 如果结果为零, 则跳过下一条指令	1 ^注	无
SDZ	[m]	递减数据存储器, 如果结果为零, 则跳过下一条指令	1 ^注	无
SIZA	[m]	递增数据存储器, 将结果放入 ACC, 如果结果为零, 则跳过下一条指令	1 ^注	无
SDZA	[m]	递减数据存储器, 将结果放入 ACC, 如果结果为零, 则跳过下一条指令	1 ^注	无
CALL	addr	子程序调用	2	无
RET		从子程序返回	2	无
RET	A,x	从子程序返回, 并将立即数放入 ACC	2	无
RETI		从中断返回	2	无
查表				
TABRDC	[m]	读取当前页的 ROM 内容, 并送至数据存储器 and TBLH	2 ^注	无
TABRDL	[m]	读取最后页的 ROM 内容, 并送至数据存储器 and TBLH	2 ^注	无
其它指令				
NOP		空指令	1	无
CLR	[m]	清除数据存储器	1 ^注	无
SET	[m]	置位数据存储器	1 ^注	无
CLR	WDT	清除看门狗定时器	1	TO, PDF
CLR	WDT1	预清看门狗定时器	1	TO, PDF
CLR	WDT2	预清看门狗定时器	1	TO, PDF
SWAP	[m]	交换数据存储器的高低字节, 结果放入数据存储器	1 ^注	无
SWAPA	[m]	交换数据存储器的高低字节, 结果放入 ACC	1	无
HALT		进入空闲/休眠模式	1	TO, PDF

- 注:
1. 对跳转指令而言, 如果比较的结果牵涉到跳转即需2个周期, 如果没有跳转发生, 则只需一个周期即可。
 2. 任何指令若要改变PCL的内容将需要2个周期来执行。
 3. 对于“CLR WDT1”和“CLR WDT2”指令而言, TO和PDF标志位也许会受执行结果影响, “CLR WDT1”和“CLR WDT2”被连续执行后, TO和PDF标志位会被清零, 除此外TO和PDF标志位保持不变。

3.11 指令定义

3.11.1 ADC A, [m] Add Data Memory to ACC with Carry

指令说明 将指定数据存储器、累加器和进位标志位的内容相加后，把结果储存回累加器。
 功能表示 $ACC \leftarrow ACC + [m] + C$
 影响标志位 OV, Z, AC, C

3.11.2 ADCM A, [m] Add ACC to Data Memory with Carry

指令说明 将指定数据存储器、累加器和进位标志位的内容相加后，把结果储存回指定数据存储器。
 功能表示 $[m] \leftarrow ACC + [m] + C$
 影响标志位 OV, Z, AC, C

3.11.3 ADD A, [m] Add Data Memory to ACC

指令说明 将指定数据存储器和累加器的内容相加后，把结果储存回累加器。
 功能表示 $ACC \leftarrow ACC + [m]$
 影响标志位 OV, Z, AC, C

3.11.4 ADD A, x Add immediate data to ACC

指令说明 将累加器和立即数的内容相加后，把结果储存回累加器。
 功能表示 $ACC \leftarrow ACC + x$
 影响标志位 OV, Z, AC, C

3.11.5 ADDM A, [m] Add ACC to Data Memory

指令说明 将指定数据存储器和累加器的内容相加后，把结果储存回指定数据存储器。
 功能表示 $[m] \leftarrow ACC + [m]$
 影响标志位 OV, Z, AC, C

3.11.6 AND A, [m] Logical AND Data Memory to ACC

指令说明 将存在累加器和指定数据存储器中的数据作AND的运算，然后把结果储存回累加器。
 功能表示 $ACC \leftarrow ACC \text{ "AND" } [m]$
 影响标志位 Z

3.11.7 AND A, x Logical AND immediate data to ACC

指令说明 将存在累加器中的数据和立即数作AND的运算，然后把结果储存回累加器。
 功能表示 $ACC \leftarrow ACC \text{ "AND" } x$
 影响标志位 Z

3.11.8 ANDM A, [m] Logical AND ACC to Data Memory

指令说明 将存在指定数据存储器和累加器中的数据作AND的运算，然后把结果储存回数据存储器。
 功能表示 $[m] \leftarrow ACC \text{ "AND" } [m]$
 影响标志位 Z

3.11.9 CALL addr Subroutine call

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

影响标志位 None

3.11.10 CLR [m] Clear Data Memory

指令说明 指定数据存储器中的每一位均清除为0。

功能表示 [m] ← 00H

影响标志位 None

3.11.11 CLR [m].i Clear bit of Data Memory

指令说明 指定数据存储器中的i位清除为0。

功能表示 [m].i ← 0

影响标志位 None

3.11.12 CLR WDT Clear Watchdog Timer

指令说明 将TO、PDF 标志位和WDT全都清零。

功能表示 WDT cleared

TO ← 0

PDF ← 0

影响标志位 TO, PDF

3.11.13 CLR WDT1 Pre-clear Watchdog Timer

指令说明 将TO、PDF 标志位和WDT全都清零，请注意此指令要结合CLR WDT2一起动作且必须交替执行才有作用，重复执行此项指令而没有与CLR WDT2交替执行将无任何作用。

功能表示 WDT cleared

TO ← 0

PDF ← 0

影响标志位 TO, PDF

3.11.14 CLR WDT2 Pre-clear Watchdog Timer

指令说明 将TO、PDF标志位和WDT全都清零，请注意此指令要结合CLR WDT1一起动作且必须交替执行才有作用，重复执行此项指令而没有与CLR WDT1交替执行将无任何作用。

功能表示 WDT cleared

TO ← 0

PDF ← 0

影响标志位 TO, PDF

3.11.15 CPL [m] Complement Data Memory

指令说明 将指定数据存储器中的每一位取逻辑反，相当于从1变0或0变1。

功能表示 [m] ← $\overline{[m]}$

影响标志位 Z

3.11.16 CPLA [m] Complement Data Memory with result in ACC

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

功能表示 ACC ← $\overline{[m]}$

影响标志位 Z

3.11.17	DAA [m]	Decimal-Adjust ACC for addition with result in Data Memory
指令说明	将存在累加器中的内容数值转换为BCD(二进制转成十进制)数值, 如果低4位大于9或AC标志位被置位, 则在低4位加上一个6, 不然低4位的内容不变, 如果高4位大于9或C标志位被置位, 则在高4位加上一个6, 十进制的转换主要是依照累加器和标志位状况, 分别加上00H、06H、60H或66H, 只有C标志位也许会被此指令影响, 它会指出原始BCD数是否大于100, 并可以进行双精度十进制数相加。	
功能表示	[m] ← ACC + 00H 或 [m] ← ACC + 06H 或 [m] ← ACC + 60H 或 [m] ← ACC + 66H	
影响标志位	C	
3.11.18	DEC [m]	Decrement Data Memory
指令说明	将在指定数据存储单元内的数据减1。	
功能表示	[m] ← [m] - 1	
影响标志位	Z	
3.11.19	DECA [m]	Decrement Data Memory with result in ACC
指令说明	将在指定数据存储单元内的数据减1, 把结果储存回累加器且数据存储单元中的内容不变。	
功能表示	ACC ← [m] - 1	
影响标志位	Z	
3.11.20	HALT	Enter power down mode
指令说明	此指令停止程序的执行并且关闭系统时钟, 但数据存储器和寄存器的内容仍被保留, WDT和预分频器(Prescaler)被清零, 暂停标志位PDF被置位且WDT溢出标志位TO 被清零。	
功能表示	TO ← 0 PDF ← 1	
影响标志位	TO, PDF	
3.11.21	INC [m]	Increment Data Memory
指令说明	将指定数据存储单元内的数据加1。	
功能表示	[m] ← [m] + 1	
影响标志位	Z	
3.11.22	INCA [m]	Increment Data Memory with result in ACC
指令说明	将指定数据存储单元内的数据加1, 把结果储存回累加器且数据存储单元中的内容不变。	
功能表示	ACC ← [m] + 1	
影响标志位	Z	
3.11.23	JMP addr	Jump unconditionally
指令说明	程序计数器的内容被指定地址所取代, 程序由新地址继续执行, 当新地址被加载入时, 必须插入一个空指令周期, 所以此指令为2个周期的指令	
功能表示	Program Counter ← addr	
影响标志位	None	

- 3.11.24 MOV A, [m] Move Data Memory to ACC**
 指令说明 将指定数据存储器的内容复制到累加器中。
 功能表示 $ACC \leftarrow [m]$
 影响标志位 None
- 3.11.25 MOV A, x Move immediate data to ACC**
 指令说明 将立即数载入至累加器中。
 功能表示 $ACC \leftarrow x$
 影响标志位 None
- 3.11.26 MOV [m], A Move ACC to Data Memory**
 指令说明 将累加器的内容复制到指定数据存储器。
 功能表示 $[m] \leftarrow ACC$
 影响标志位 None
- 3.11.27 NOP No operation**
 指令说明 空操作，接下来顺序执行下一条指令。
 功能表示 No operation
 影响标志位 None
- 3.11.28 OR A, [m] Logical OR Data Memory to ACC**
 指令说明 将存在累加器和指定数据存储器中的数据作OR的运算，然后把结果储存回累加器。
 功能表示 $ACC \leftarrow ACC \text{ "OR" } [m]$
 影响标志位 Z
- 3.11.29 OR A, x Logical OR immediate data to ACC**
 指令说明 将存在累加器中的数据和立即数作OR的运算，然后把结果储存回累加器。
 功能表示 $ACC \leftarrow ACC \text{ "OR" } x$
 影响标志位 Z
- 3.11.30 ORM A, [m] Logical OR ACC to Data Memory**
 指令说明 将存在指定数据存储器 and 累加器中的数据作OR的运算，然后把结果储存回数据存储器。
 功能表示 $[m] \leftarrow ACC \text{ "OR" } [m]$
 影响标志位 Z
- 3.11.31 RET Return from subroutine**
 指令说明 将堆栈区的数据取回至程序计数器，程序由取回的地址继续执行。
 功能表示 $Program\ Counter \leftarrow Stack$
 影响标志位 None
- 3.11.32 RET A, x Return from subroutine and load immediate data to ACC**
 指令说明 将堆栈区的数据取回至程序计数器且累加器载入立即数，程序由取回的地址继续执行。
 功能表示 $Program\ Counter \leftarrow Stack$
 $ACC \leftarrow x$
 影响标志位 None
- 3.11.33 RETI Return from interrupt**
 指令说明 将堆栈区的数据取回至程序计数器且中断功能通过EMI位重新被使能，EMI是控制中

断使能的主中断位(寄存器INTC的第0位), 如果在执行RETI指令之前还有中断未被响应, 则这个中断将在返回主程序之前被响应。

功能表示 Program Counter ← Stack
EMI ← 1
影响标志位 None

3.11.34 RL [m] Rotate Data Memory left

指令说明 将指定数据存储器的内容向左移1个位, 且第7位移回第0位。
功能表示 $[m].(i+1) \leftarrow [m].i ; (i = 0\sim 6)$
 $[m].0 \leftarrow [m].7$
影响标志位 None

3.11.35 RLA [m] Rotate Data Memory left with result in ACC

指令说明 将指定数据存储器的内容向左移1个位, 且第7位移回第0位, 而移位的结果储存回累加器且数据存储器中的内容不变。
功能表示 $ACC.(i+1) \leftarrow [m].i ; (i = 0\sim 6)$
 $ACC.0 \leftarrow [m].7$
影响标志位 None

3.11.36 RLC [m] Rotate Data Memory Left through Carry

指令说明 将指定数据存储器的内容连同进位标志位向左移1个位, 第7位取代进位位且原本的进位标志位移至第0位。
功能表示 $[m].(i+1) \leftarrow [m].i ; (i = 0\sim 6)$
 $[m].0 \leftarrow C$
 $C \leftarrow [m].7$
影响标志位 C

3.11.37 RLCA [m] Rotate Data Memory left through Carry with result in ACC

指令说明 将指定数据存储器的内容连同进位标志位向左移1个位, 第7位取代进位位且原本的进位标志位移至第0位, 而移位的结果储存回累加器且数据存储器中的内容不变。
功能表示 $ACC.(i+1) \leftarrow [m].i ; (i = 0\sim 6)$
 $ACC.0 \leftarrow C$
 $C \leftarrow [m].7$
影响标志位 C

3.11.38 RR [m] Rotate Data Memory right

指令说明 将指定数据存储器的内容向右移1个位, 且第0位移回第7位。
功能表示 $[m].i \leftarrow [m].(i+1) ; (i = 0\sim 6)$
 $[m].7 \leftarrow [m].0$
影响标志位 None

3.11.39 RRA [m] Rotate Data Memory right with result in ACC

指令说明 将指定数据存储器的内容向右移1个位, 且第0位移回第7位, 而移位的结果储存回累加器且数据存储器中的内容不变。
功能表示 $ACC.i \leftarrow [m].(i+1) ; (i = 0\sim 6)$
 $ACC.7 \leftarrow [m].0$
影响标志位 None

- 3.11.40 RRC [m] Rotate Data Memory right through Carry**
 指令说明 将指定数据存储器的内容连同进位标志位向右移1个位，第0位取代进位位且原本的进位标志位移至第7位。
 功能表示 $[m].i \leftarrow [m].(i+1)$; (i = 0~6)
 $[m].7 \leftarrow C$
 $C \leftarrow [m].0$
 影响标志位 C
- 3.11.41 RRCA [m] Rotate Data Memory right through Carry with result in ACC**
 指令说明 将指定数据存储器的内容连同进位标志位向右移1个位，第0位取代进位位且原本的进位标志位移至第7位，而移位的结果储存回累加器且数据存储器中的内容不变。
 功能表示 $ACC.i \leftarrow [m].(i+1)$; (i = 0~6)
 $ACC.7 \leftarrow C$
 $C \leftarrow [m].0$
 影响标志位 C
- 3.11.42 SBC A, [m] Subtract Data Memory from ACC with Carry**
 指令说明 将累加器中的数据与指定数据存储器内容和进位标志位的反相减，把结果储存回累加器。如果结果为负，C标志位清除为0，反之结果为正或0，C标志位设置为1。
 功能表示 $ACC \leftarrow ACC - [m] - \bar{C}$
 影响标志位 OV, Z, AC, C
- 3.11.43 SBCM A, [m] Subtract Data Memory from ACC with Carry and result in Data Memory**
 指令说明 将累加器中的数据与指定数据存储器内容和进位标志位的反相减，把结果储存回数据存储器。如果结果为负，C标志位清除为0，反之结果为正或0，C标志位设置为1。
 功能表示 $[m] \leftarrow ACC - [m] - \bar{C}$
 影响标志位 OV, Z, AC, C
- 3.11.44 SDZ [m] Skip if Decrement Data Memory is 0**
 指令说明 将指定数据存储器的内容先减去1后，如果结果为0，则程序计数器再加1跳过下一条指令，由于取得下一指令时会要求插入一个空指令周期，所以此指令为2个周期的指令。如果结果不为0，则程序继续执行下面的指令。
 功能表示 $[m] \leftarrow [m] - 1$
 Skip if [m] = 0
 影响标志位 None
- 3.11.45 SDZA [m] Skip if decrement Data Memory is zero with result in ACC**
 指令说明 将指定数据存储器的内容先减去1后，如果结果为0，则程序计数器再加1 跳过下一条指令，此结果会被储存回累加器且指定数据存储器中的内容不变，由于取得下一指令时会要求插入一个空指令周期，所以此指令为2个周期的指令。如果结果不为0，则程序继续执行下面的指令。
 功能表示 $ACC \leftarrow [m] - 1$
 Skip if ACC = 0
 影响标志位 None

- 3.11.46 SET [m] Set Data Memory**
 指令说明 将指定数据存储器的每一个位置位为1。
 功能表示 $[m] \leftarrow FFH$
 影响标志位 None
- 3.11.47 SET [m].i Set bit of Data Memory**
 指令说明 将指定数据存储器的第i位置位为1。
 功能表示 $[m].i \leftarrow 1$
 影响标志位 None
- 3.11.48 SIZ [m] Skip if increment Data Memory is 0**
 指令说明 将指定数据存储器的内容先加上1后, 如果结果为0, 则程序计数器再加1 跳过下一条指令, 由于取得下一指令时会要求插入一个空指令周期, 所以此指令为2个周期的指令。如果结果不为0, 则程序继续执行下面的指令。
 功能表示 $[m] \leftarrow [m] + 1$
 Skip if [m] = 0
 影响标志位 None
- 3.11.49 SIZA [m] Skip if increment Data Memory is zero with result in ACC**
 指令说明 将指定数据存储器的内容先加上1后, 如果结果为0, 则程序计数器再加1跳过下一条指令, 此结果会被储存回累加器且指定数据存储器中的内容不变, 由于取得下一指令时会要求插入一个空指令周期, 所以此指令为2个周期的指令。如果结果不为0, 则程序继续执行下面的指令。
 功能表示 $ACC \leftarrow [m] + 1$
 Skip if ACC = 0
 影响标志位 None
- 3.11.50 SNZ [m].i Skip if bit i of Data Memory is not 0**
 指令说明 如果指定数据存储器的第i位不为0, 则程序计数器再加1跳过下一条指令, 由于取得下一指令时会要求插入一个空指令周期, 所以此指令为2个周期的指令。如果结果不为0, 程序继续执行下面的指令。
 功能表示 Skip if [m].i $\neq 0$
 影响标志位 None
- 3.11.51 SUB A, [m] Subtract Data Memory from ACC**
 指令说明 将累加器中内容减去指定数据存储器的数据, 把结果储存回累加器。如果结果为负, C标志位清除为0, 反之结果为正或0, C标志位设置为1。
 功能表示 $ACC \leftarrow ACC - [m]$
 影响标志位 OV, Z, AC, C
- 3.11.52 SUBM A, [m] Subtract Data Memory from ACC with result in Data Memory**
 指令说明 将累加器中内容减去指定数据存储器的数据, 把结果储存回数据存储器。如果结果为负, C标志位清除为0, 反之结果为正或0, C标志位设置为1。
 功能表示 $[m] \leftarrow ACC - [m]$
 影响标志位 OV, Z, AC, C

- 3.11.53 SUB A, x Subtract immediate Data from ACC**
 指令说明 将累加器中内容减去立即数，把结果储存回累加器。如果结果为负，C标志位清除为0，反之结果为正或0，C标志位设置为1。
 功能表示 $ACC \leftarrow ACC - x$
 影响标志位 OV, Z, AC, C
- 3.11.54 SWAP [m] Swap nibbles of Data Memory**
 指令说明 将指定数据存储器的低4位与高4位互相交换。
 功能表示 $[m].3 \sim [m].0 \leftrightarrow [m].7 \sim [m].4$
 影响标志位 None
- 3.11.55 SWAPA [m] Swap nibbles of Data Memory with result in ACC**
 指令说明 将指定数据存储器的低4位与高4位互相交换，然后把结果储存回累加器且数据存储器的内容不变。
 功能表示 $ACC.3 \sim ACC.0 \leftarrow [m].7 \sim [m].4$
 $ACC.7 \sim ACC.4 \leftarrow [m].3 \sim [m].0$
 影响标志位 None
- 3.11.56 SZ [m] Skip if Data Memory is 0**
 指令说明 如果指定数据存储器的内容为0，则程序计数器再加1跳过下一条指令，由于取得下一指令时会要求插入一个空指令周期，所以此指令为2个周期的指令。如果结果不为0，程序继续执行下面的指令。
 功能表示 Skip if $[m] = 0$
 影响标志位 None
- 3.11.57 SZA [m] Skip if Data Memory is 0 with data movement to ACC**
 指令说明 将指定数据存储器的内容复制到累加器，如果值为0，则程序计数器再加1跳过下一条指令，由于取得下一指令时会要求插入一个空指令周期，所以此指令为2个周期的指令。如果结果不为0，程序继续执行下面的指令。
 功能表示 $ACC \leftarrow [m]$
 Skip if $[m] = 0$
 影响标志位 None
- 3.11.58 SZ [m].i Skip if bit i of Data Memory is 0**
 指令说明 如果指定存储器第i位为0，则程序计数器再加1跳过下一条指令，由于取得下一指令时会要求插入一个空指令周期，所以此指令为2个周期的指令。如果结果不为0，程序继续执行下面的指令。
 功能表示 Skip if $[m].i = 0$
 影响标志位 None
- 3.11.59 TABRDC [m] Read table (current page) to TBLH and Data Memory**
 指令说明 将表格指针TBLP所指的程序代码低字节(当前页)移至指定存储器且将高字节移至TBLH。
 功能表示 $[m] \leftarrow$ 程序代码(低字节)
 $TBLH \leftarrow$ 程序代码(高字节)
 影响标志位 None
- 3.11.60 TABRDL [m] Read table (last page) to TBLH and Data Memory**
 指令说明 将表格指针TBLP所指的程序代码低字节(最后一页)移至指定存储器且将高字节移至TBLH。

功能表示 $[m] \leftarrow$ 程序代码(低字节)
 $TBLH \leftarrow$ 程序代码(高字节)
 影响标志位 None

3.11.61 XOR A, [m] Logical XOR Data Memory to ACC

指令说明 将存在累加器和指定数据存储器中的数据作XOR的运算，然后把结果储存回累加器。

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

影响标志位 Z

3.11.62 XORM A, [m] Logical XOR ACC to Data Memory

指令说明 将存在指定数据存储器 and 累加器中的数据作XOR的运算，然后把结果储存回数据存储器。

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

影响标志位 Z

3.11.63 XOR A, x Logical XOR immediate data to ACC

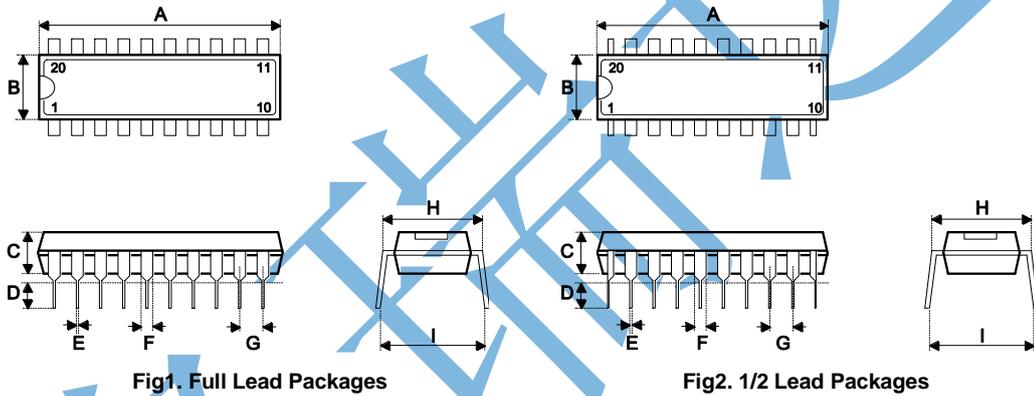
指令说明 将存在累加器中的数据和立即数作XOR的运算，然后把结果储存回累加器。

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

影响标志位 Z

第 4 章 封装

4.1 20-pin DIP (300mil)外形尺寸



4.1.1 MS-001d (见 fig1)

符号	尺寸(单位: inch)		
	最小	正常	最大
A	0.980	—	1.060
B	0.240	—	0.280
C	0.115	—	0.195
D	0.115	—	0.150
E	0.014	—	0.022
F	0.045	—	0.070
G	—	0.100	—
H	0.300	—	0.325
I	—	0.430	—

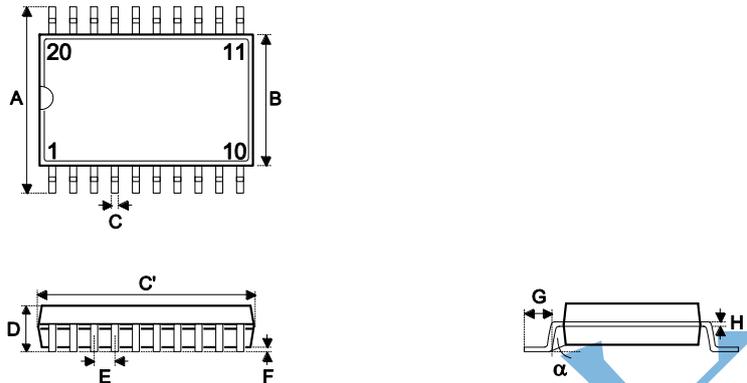
符号	尺寸(单位: mm)		
	最小	正常	最大
A	24.89	—	26.92
B	6.10	—	7.11
C	2.92	—	4.95
D	2.92	—	3.81
E	0.36	—	0.56
F	1.14	—	1.78
G	—	2.54	—
H	7.62	—	8.26
I	—	10.92	—

4.1.2 M0-095a (见 fig2)

符号	尺寸(单位: inch)		
	最小	正常	最大
A	0.945	—	0.985
B	0.275	—	0.295
C	0.120	—	0.150
D	0.110	—	0.150
E	0.014	—	0.022
F	0.045	—	0.060
G	—	0.100	—
H	0.300	—	0.325
I	—	0.430	—

符号	尺寸(单位: mm)		
	最小	正常	最大
A	24.00	—	25.02
B	6.99	—	7.49
C	3.05	—	3.81
D	2.79	—	3.81
E	0.36	—	0.56
F	1.14	—	1.52
G	—	2.54	—
H	7.62	—	8.26
I	—	10.92	—

4.2 20-pin SOP (300mil) 外形尺寸

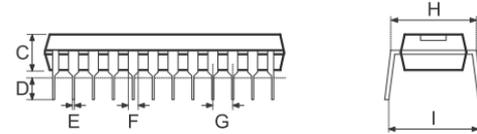
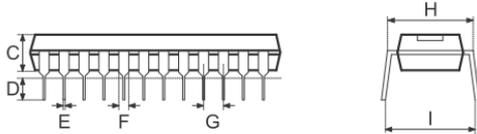
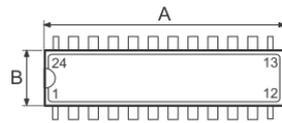
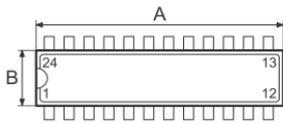


4.2.1 MS-013

符号	尺寸(单位: inch)		
	最小	正常	最大
A	0.393	—	0.419
B	0.256	—	0.300
C	0.012	—	0.020
C'	0.496	—	0.512
D	—	—	0.104
E	—	0.050	—
F	0.004	—	0.012
G	0.016	—	0.050
H	0.008	—	0.013
α	0°	—	8°

符号	尺寸(单位: mm)		
	最小	正常	最大
A	9.98	—	10.64
B	6.50	—	7.62
C	0.30	—	0.51
C'	12.60	—	13.00
D	—	—	2.64
E	—	1.27	—
F	0.10	—	0.30
G	0.41	—	1.27
H	0.20	—	0.33
α	0°	—	8°

4.3 24-pin SKDIP(300mil) 外形尺寸


Fig1. Full Lead Packages
Fig2. 1/2 Lead Packages
4.3.1 MS-001d (见 Fig1)

符号	尺寸(单位: inch)		
	最小	正常	最大
A	1.230	—	1.280
B	0.240	—	0.280
C	0.115	—	0.195
D	0.115	—	0.150
E	0.014	—	0.022
F	0.045	—	0.070
G	—	0.100	—
H	0.300	—	0.325
I	—	0.430	—

符号	尺寸(单位: mm)		
	最小	正常	最大
A	31.24	—	32.51
B	6.10	—	7.11
C	2.92	—	4.95
D	2.92	—	3.81
E	0.36	—	0.56
F	1.14	—	1.78
G	—	2.54	—
H	7.62	—	8.26
I	—	10.92	—

4.3.2 MS-001d (见 Fig2)

符号	尺寸(单位: inch)		
	最小	正常	最大
A	1.160	—	1.195
B	0.240	—	0.280
C	0.115	—	0.195
D	0.115	—	0.150
E	0.014	—	0.022

F	0.045	—	0.070
G	—	0.100	—
H	0.300	—	0.325
I	—	0.430	—

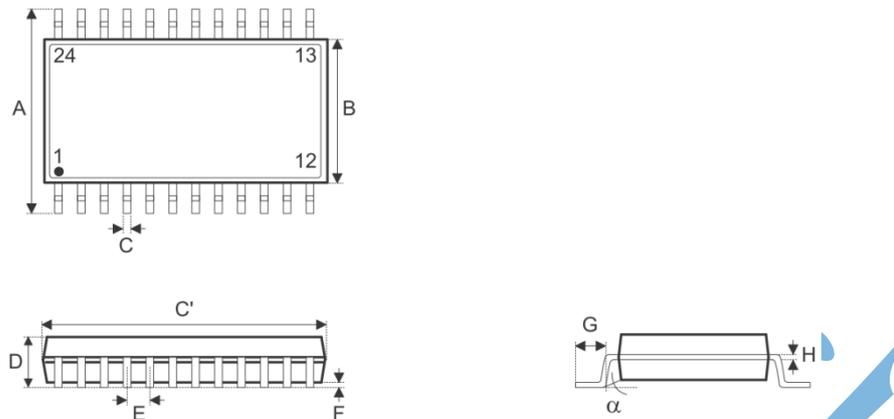
符号	尺寸(单位: mm)		
	最小	正常	最大
A	29.46	—	30.35
B	6.10	—	7.11
C	2.92	—	4.95
D	2.92	—	3.81
E	0.36	—	0.56
F	1.14	—	1.78
G	—	2.54	—
H	7.62	—	8.26
I	—	10.92	—

4.3.3 M0-095a (见 Fig2)

符号	尺寸(单位: inch)		
	最小	正常	最大
A	1.145	—	1.185
B	0.275	—	0.295
C	0.120	—	0.150
D	0.110	—	0.150
E	0.014	—	0.022
F	0.045	—	0.060
G	—	0.100	—
H	0.300	—	0.325
I	—	0.430	—

符号	尺寸(单位: mm)		
	最小	正常	最大
A	29.08	—	30.10
B	6.99	—	7.49
C	3.05	—	3.81
D	2.79	—	3.81
E	0.36	—	0.56
F	1.14	—	1.52
G	—	2.54	—
H	7.62	—	8.26
I	—	10.92	—

4.4 24-pin SOP(300mil)外形尺寸

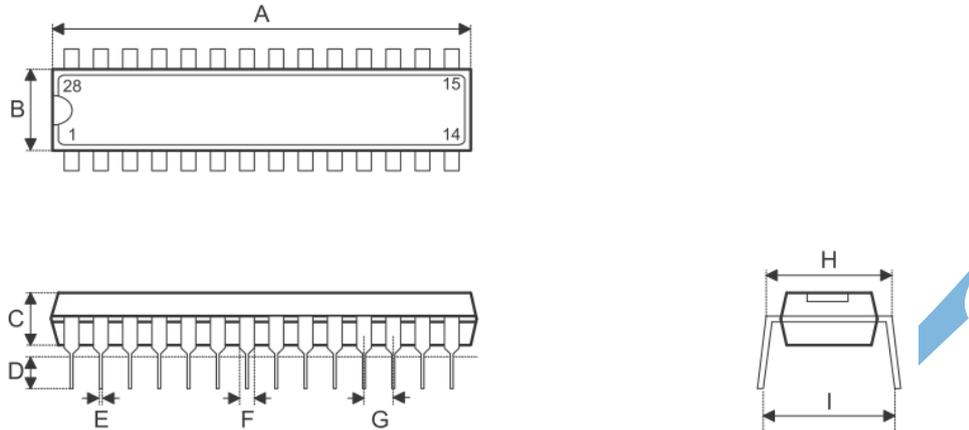


4.4.1 MS-013

符号	尺寸(单位: inch)		
	最小	正常	最大
A	0.393	—	0.419
B	0.256	—	0.300
C	0.012	—	0.020
C'	0.598	—	0.613
D	—	—	0.104
E	—	0.050	—
F	0.004	—	0.012
G	0.016	—	0.050
H	0.008	—	0.013
α	0°	—	8°

符号	尺寸(单位: mm)		
	最小	正常	最大
A	9.98	—	10.64
B	6.50	—	7.62
C	0.30	—	0.51
C'	15.19	—	15.57
D	—	—	2.64
E	—	1.27	—
F	0.10	—	0.30
G	0.41	—	1.27
H	0.20	—	0.33
α	0°	—	8°

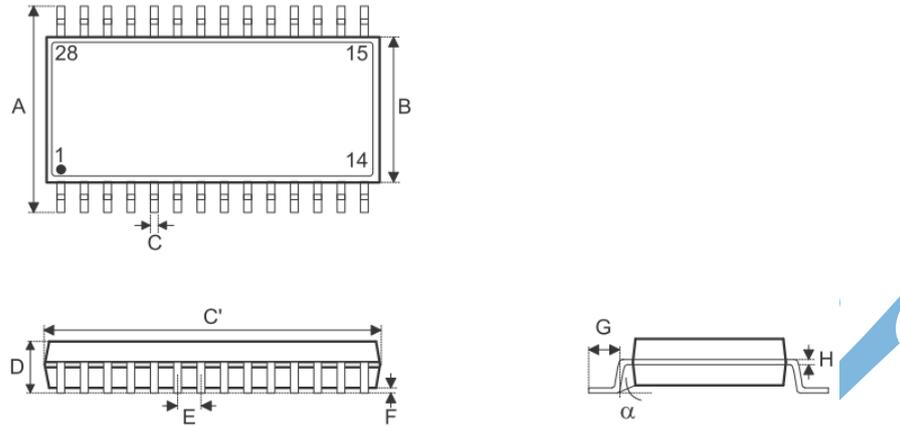
4.5 28-pin SKDIP(300mil)外形尺寸



符号	尺寸(单位: inch)		
	最小	正常	最大
A	1.375	—	1.395
B	0.278	—	0.298
C	0.125	—	0.135
D	0.125	—	0.145
E	0.016	—	0.020
F	0.050	—	0.070
G	—	0.100	—
H	0.295	—	0.315
I	—	0.375	—

符号	尺寸(单位: mm)		
	最小	正常	最大
A	34.93	—	35.43
B	7.06	—	7.57
C	3.18	—	3.43
D	3.18	—	3.68
E	0.41	—	0.51
F	1.27	—	1.78
G	—	2.54	—
H	7.49	—	8.00
I	—	9.53	—

4.6 28-pin SOP (300mil) 外形尺寸



4.6.1 MS-013

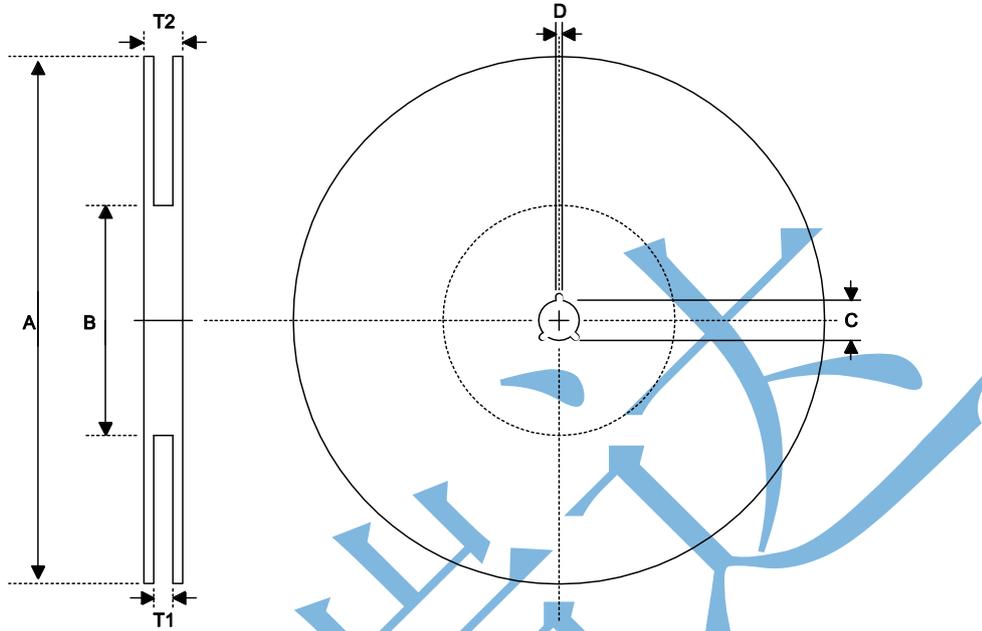
符号	尺寸(单位: inch)		
	最小	正常	最大
A	0.393	—	0.419
B	0.256	—	0.300
C	0.012	—	0.020
C'	0.697	—	0.713
D	—	—	0.104
E	—	0.050	—
F	0.004	—	0.012
G	0.016	—	0.050
H	0.008	—	0.013
α	0°	—	8°

符号	尺寸(单位: mm)		
	最小	正常	最大
A	9.98	—	10.64
B	6.50	—	7.62
C	0.30	—	0.51
C'	17.70	—	18.11
D	—	—	2.64
E	—	1.27	—
F	0.10	—	0.30
G	0.41	—	1.27
H	0.20	—	0.33

α	0°	—	8°
----------	-----------	---	-----------

4.7 包装带和卷轴规格

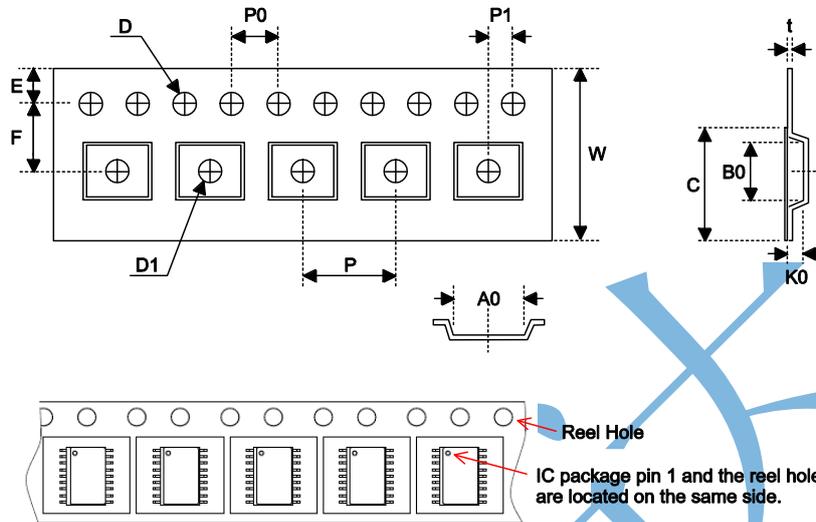
4.7.1 卷轴尺寸



SOP 20W, SOP 24W(300mil), SOP 28W(300mil)

符号	说明	尺寸 (单位: mm)
A	卷轴外圈直径	330.0 ± 1.0
B	卷轴内圈直径	100.0 ± 1.5
C	轴心直径	$13.0^{+0.5/-0.2}$
D	缝宽	2.0 ± 0.5
T1	轮缘宽	$24.8^{+0.3/-0.2}$
T2	卷轴宽	30.2 ± 0.2

4.8 运输带尺寸



4.8.1 SOP 20W

符号	说明	尺寸 (单位: mm)
W	运输带宽	24.0 ^{+0.3/-0.1}
P	空穴间距	12.0±0.1
E	穿孔位置	1.75±0.1
F	空穴至穿孔距离(宽度)	11.5±0.1
D	穿孔直径	1.5 ^{+0.1/-0.0}
D1	空穴中之小孔直径	1.5 ^{+0.25/-0.00}
P0	穿孔间距	4.0±0.1
P1	空穴至穿孔距离(长度)	2.0±0.1
A0	空穴长	10.8±0.1
B0	空穴宽	13.3±0.1
K0	空穴深	3.2±0.1
T	传输带厚度	0.30±0.05
C	覆盖带宽度	21.3±0.1

4.8.2 SOP 24W

符号	说明	尺寸 (单位: mm)
W	运输带宽	24.0±0.3
P	空穴间距	12.0±0.1
E	穿孔位置	1.75±0.1
F	空穴至穿孔距离(宽度)	11.5±0.1
D	穿孔直径	1.55 ^{+0.10/-0.00}
D1	空穴中之小孔直径	1.50 ^{+0.25/-0.00}
P0	穿孔间距	4.0±0.1
P1	空穴至穿孔距离(长度)	2.0±0.1
A0	空穴长	10.9±0.1
B0	空穴宽	15.9±0.1
K0	空穴深	3.1±0.1
t	传输带厚度	0.35±0.05
C	覆盖带宽度	21.3±0.1

4.8.3 SOP 28W(300mil)

符号	说明	尺寸 (单位: mm)
W	运输带宽	24.0±0.3
P	空穴间距	12.0±0.1
E	穿孔位置	1.75±0.10
F	空穴至穿孔距离(宽度)	11.5±0.1
D	穿孔直径	1.5 ^{+0.1/-0.0}
D1	空穴中之小孔直径	1.50 ^{+0.25/-0.00}
P0	穿孔间距	4.0±0.1
P1	空穴至穿孔距离(长度)	2.0±0.1
A0	空穴长	10.85±0.10
B0	空穴宽	18.34±0.10
K0	空穴深	2.97±0.10
t	传输带厚度	0.35±0.01
C	覆盖带宽度	21.3±0.1

第 5 章 订购信息

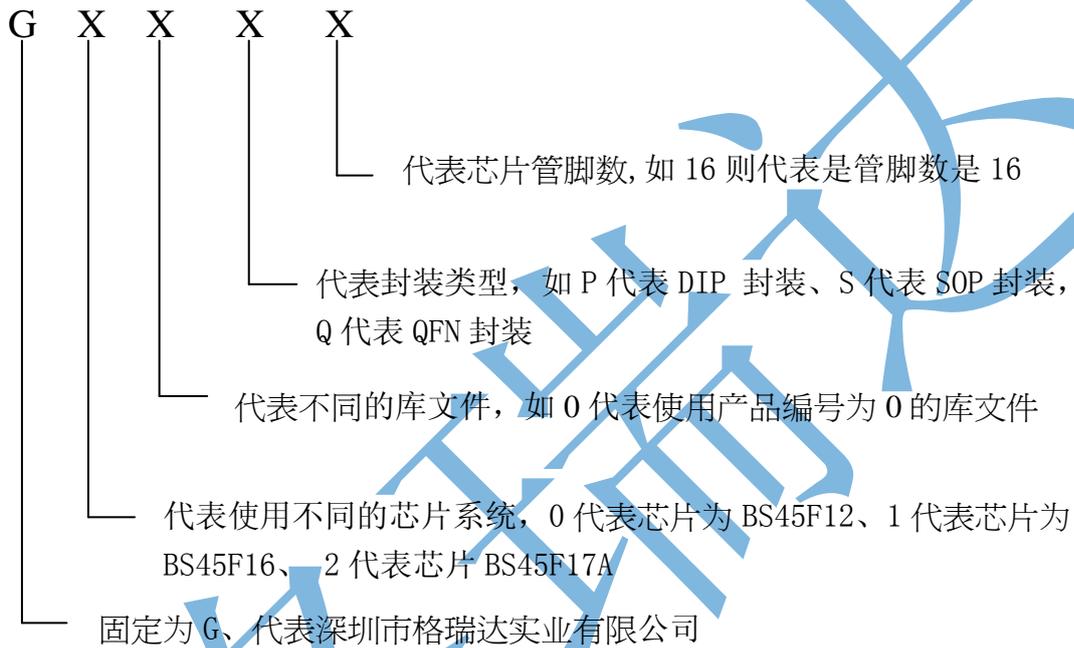
5.1 订购信息

下单规格	功能简述	空片名称	芯片型号	封装
G10P20	8Bit 带 A/D 触摸按键式 Flash 单片机	L20	BS45F16	DIP20
G10S20	8Bit 带 A/D 触摸按键式 Flash 单片机	L20S	BS45F16	SOP20
G10P24	8Bit 带 A/D 触摸按键式 Flash 单片机	L24	BS45F16	SKDIP24
G10S24	8Bit 带 A/D 触摸按键式 Flash 单片机	L24S	BS45F16	SOP24
G10P28	8Bit 带 A/D 触摸按键式 Flash 单片机	L28	BS45F16	SKDIP28
G10S28	8Bit 带 A/D 触摸按键式 Flash 单片机	L28S	BS45F16	SOP28

第 6 章 产品命名规则

6.1 产品命名规则

客户可二次开发的产品名称构成：



深圳市格瑞达实业有限公司（总公司）

SHENZHEN GREENMCU TECHNOLOGY CO.,LTD.

地址：深圳市福田区彩田南路海鹰大厦 20B

电话：(86) 755-83051793 82913392

(86) 755-82914749 82913502

传真：(86) 755-82971356

网址：www.greenmcu.com

深圳市格瑞达实业有限公司（顺德办事处）

地址：顺德区容桂镇文海西路保利百合花园 10 栋 B 单元 1901

电话：(86) 757-28302691 22909432

传真：(86) 757-28302691