



九齐科技股份有限公司
Nyquest Technology Co., Ltd.

DATA SHEET

NY8BE62D

14 I/O + 13-通道 ADC 8 位带 EEPROM 的
MTP-Based 单片机

Version 1.0

Sept. 30, 2021

NYQUEST TECHNOLOGY CO. reserves the right to change this document without prior notice. Information provided by NYQUEST is believed to be accurate and reliable. However, NYQUEST makes no warranty for any errors which may appear in this document. Contact NYQUEST to obtain the latest version of device specifications before placing your orders. No responsibility is assumed by NYQUEST for any infringement of patent or other rights of third parties which may result from its use. In addition, NYQUEST products are not authorized for use as critical components in life support devices/systems or aviation devices/systems, where a malfunction or failure of the product may reasonably be expected to result in significant injury to the user, without the express written approval of NYQUEST.

改版记录

版本	日期	内容描述	修正页
0.1	2021/03/25	预发布版本。	-
1.0	2021/09/30	<ol style="list-style-type: none"> 1. 修正 R-page SFR 0x17 名称 2. 修正 EEPROM 写入超时设置与注意事项 3. 修正 3.1.8 PCON Bit1, Bit0 名称与 Bit6 的初值 4. 修正 3.1.9 BWUCON (PortB 唤醒控制寄存器) 5. 移除 3.1.10 PCHBUF 的最高位说明 6. 修正 3.1.20 AWUCON PortA 唤醒控制寄存器的初值 7. 修正 3.1.23 INTEDG Bit4 “EIS0” 的说明 	<p style="text-align: right;">16</p> <p style="text-align: right;">19, 20</p> <p style="text-align: right;">23</p> <p style="text-align: right;">24</p> <p style="text-align: right;">24</p> <p style="text-align: right;">29</p> <p style="text-align: right;">30</p>

目 录

1. 概述	8
1.1 功能	9
1.2 系统框图	11
1.3 引脚图	11
1.4 引脚说明	12
2. 内存结构	14
2.1 Program Memory	14
2.2 寄存器存储器	14
2.3 EEPROM Memory	18
2.3.1 EEA (EEPROM Address Register) (EEPROM地址寄存器)	19
2.3.2 EED (EEPROM Data Register) (EEPROM数据寄存器)	19
2.3.3 EEPL (EEPROM write protect Register) (EEPROM写保护寄存器)	19
2.3.4 EETO (EEPROM Time-Out Register) (EEPROM超时寄存器)	19
3. 功能概述	21
3.1 R-page特殊功能寄存器	21
3.1.1 INDF (间接寻址寄存器)	21
3.1.2 TMR0 (定时器0寄存器)	21
3.1.3 PCL (程序计数器低字节)	21
3.1.4 STATUS (状态寄存器)	21
3.1.5 FSR (数据指针寄存器)	22
3.1.6 PortA (PortA数据寄存器)	22
3.1.7 PortB (PortB数据寄存器)	23
3.1.8 PCON (Power寄存器)	23
3.1.9 BWUCON (PortB唤醒控制寄存器)	24
3.1.10 PCHBUF (程序计数器高字节)	24
3.1.11 ABPLCON (PortA/PortB下拉电阻控制寄存器)	24
3.1.12 BPHCON (PortB上拉电阻控制寄存器)	24
3.1.13 INTE (中断使能寄存器)	25
3.1.14 INTF (中断标志寄存器)	26
3.1.15 ADMD (ADC模式寄存器)	26
3.1.16 ADR (ADC时钟, ADC中断标志位与ADC转换结果低四位数据寄存器)	27

3.1.17	ADD (ADC转换结果高八位数据寄存器)	28
3.1.18	ADVREFH (ADC参考电压寄存器)	28
3.1.19	ADCR (ADC采样时间与ADC位数寄存器)	28
3.1.20	AWUCON (PortA唤醒控制寄存器)	29
3.1.21	PACON (ADC引脚数模控制寄存器)	29
3.1.22	ADJMD (ADC offset Register) (ADC补偿寄存器)	29
3.1.23	INTEDG (外部中断控制寄存器)	30
3.1.24	TMRH (定时器 1/2 高字节寄存器)	30
3.1.25	ANAEN (比较器使能寄存器)	31
3.1.26	RFC (电阻频率转换控制寄存器)	31
3.1.27	TM4RH (定时器 4 高字节寄存器)	32
3.1.28	OSCCAL (内部高频振荡微调寄存器)	32
3.1.29	INTE2 (第 2 中断屏蔽寄存器)	32
3.2	T0MD定时器 0 控制寄存器	33
3.3	F-page 殊功能寄存器	34
3.3.1	IOSTA (PortA I/O控制寄存器)	34
3.3.2	IOSTB (PortB I/O控制寄存器)	34
3.3.3	APHCON (PortA上拉电阻控制寄存器)	34
3.3.4	PS0CV (预分频器 0 寄存器)	35
3.3.5	BODCON (PortB开漏控制寄存器)	35
3.3.6	CMPCR (比较器控制寄存器)	35
3.3.7	PCON1 (Power控制寄存器 1)	36
3.4	S-page 特殊功能寄存器	37
3.4.1	TMR1 (定时器 1 寄存器)	37
3.4.2	T1CR1 (定时器 1 控制寄存器 1)	37
3.4.3	T1CR2 (定时器 1 控制寄存器 2)	38
3.4.4	PWM1DUTY (PWM1 占空比寄存器)	39
3.4.5	PS1CV (预分频器 1 寄存器)	39
3.4.6	BZ1CR (蜂鸣器 1 控制寄存器)	39
3.4.7	IRCR (IR控制寄存器)	40
3.4.8	TBHP (表格指针高字节寄存器)	41
3.4.9	TBHD (表格数据高字节寄存器)	41
3.4.10	P2CR1 (PWM2 控制寄存器 1)	42
3.4.11	PWM2DUTY (PWM2 占空比寄存器)	42
3.4.12	OSCCR (振荡器控制寄存器)	42
3.4.13	P3CR1 (PWM3 控制寄存器 1)	43

3.4.14	PWM3DUTY (PWM3 占空比寄存器)	43
3.4.15	TMR4 (定时器4 寄存器)	43
3.4.16	T4CR1 (定时器4 控制寄存器1)	44
3.4.17	T4CR2 (定时器4 控制寄存器2)	45
3.4.18	PWM4DUTY (PWM4 占空比寄存器)	45
3.4.19	PS4CV (预分频器4 寄存器)	46
3.4.20	TMR5 (定时器5 寄存器)	46
3.4.21	T5CR1 (Timer5 Control Register1) (定时器5 控制寄存器1)	46
3.4.22	T5CR2 (定时器5 控制寄存器2)	47
3.4.23	PWM5DUTY (PWM5 占空比寄存器)	48
3.4.24	PS5CV (预分频器5 寄存器)	48
3.4.25	TM5RH (定时器5 高字节寄存器)	48
3.5	T-page 特殊功能寄存器	49
3.5.1	INTE3 (中断使能寄存器3)	49
3.5.2	INTF3 (中断标志寄存器3)	49
3.5.3	CCPCON (CCP控制寄存器)	49
3.5.4	PWMDB (PWM死区控制寄存器)	51
3.6	I/O Port	51
3.6.1	IO引脚结构框图	53
3.7	定时器0	62
3.8	定时器1/PWM1/Buzzer1	63
3.9	PWM2	65
3.10	PWM3	66
3.11	Timer4 / PWM4	67
3.12	Timer5 / PWM5	69
3.13	CCP模式	71
3.13.1	捕捉模式	71
3.13.2	比较模式	72
3.13.3	CCP PWM Mode	72
3.14	电阻/频率转换器模式 (RFC)	74
3.15	IR Carrier	75
3.16	低电压侦测 (LVD)	75
3.17	电压比较器	77
3.17.1	比较器参考电压 (Vref)	78
3.18	ADC模数转换器	80

3.18.1	ADC参考电压.....	80
3.18.2	ADC模拟输入通道.....	81
3.18.3	ADC时钟 (ADCLK), 采样时钟 (SHCLK) 与位数选择.....	82
3.18.4	ADC偏移误差校准.....	83
3.18.5	ADC操作过程.....	83
3.19	看门狗定时器 (WDT).....	83
3.20	中断.....	83
3.20.1	Timer0 上溢中断.....	84
3.20.2	Timer1 下溢中断.....	84
3.20.3	Timer4 下溢中断.....	84
3.20.4	Timer5 下溢中断/CCP中断.....	85
3.20.5	看门狗超时中断.....	85
3.20.6	PA/PB输入引脚状态改变中断.....	85
3.20.7	外部中断0.....	85
3.20.8	外部中断1.....	85
3.20.9	External 2 Interrupt外部中断2.....	85
3.20.10	低电压侦测中断/比较器输出翻转中断.....	85
3.20.11	ADC模数转换完成中断.....	85
3.20.12	EEPROM写入完成中断.....	85
3.21	振荡器配置.....	86
3.22	工作模式.....	88
3.22.1	正常模式.....	89
3.22.2	慢速模式.....	89
3.22.3	待机模式.....	89
3.22.4	睡眠模式.....	90
3.22.5	唤醒稳定时间.....	90
3.22.6	工作模式概述.....	90
3.23	复位.....	91
3.24	片上仿真 (OCD).....	92
3.24.1	功能描述.....	92
3.24.2	OCD限制.....	93
4.	指令设置.....	94
5.	配置字节表.....	110
6.	电气特性.....	112

6.1	最大绝对值	112
6.2	直流电气特性	112
6.3	OSC 特性	115
6.4	比较器/LVD电气特性	115
6.5	ADC电气特性	115
6.6	特性图	116
6.6.1	高速RC振荡频率 (I_HRC) 与电源电压(VDD) 曲线图	116
6.6.2	高速RC振荡频率 (I_HRC) 与温度曲线图	116
6.6.3	低速RC振荡频率 (I_LRC) 与电源电压(VDD) 曲线图	117
6.6.4	低速RC振荡频率 (I_LRC) 与温度曲线图	117
6.6.5	内部参考电压LDO与温度曲线图	118
6.6.6	内部上拉电阻与电源电压(VDD) 曲线图	118
6.6.7	VIH/VIL 与电源电压(VDD) 曲线图	119
6.7	建议工作电压	120
6.8	LVR电压与温度曲线图	121
6.9	LVD电压与温度曲线图	121
7.	封装尺寸	122
7.1	8 引脚SOP (150 毫寸)	122
7.2	14 引脚SOP (150 毫寸)	122
7.3	16 引脚SOP (150 毫寸)	123
8.	订购信息	123

1. 概述

NY8BE62D 是以MTP作为程式记忆体，并以EEPROM作为非挥发性资料记忆体的 8 位微控制器，特别适合断电后还需保持资料内容的应用，例如遥控器、风扇/灯光控制或是游乐器周边等等。除此之外内建类比数位转换器也适用于家电或量测等等的的应用设计，除此之外更针对雾化器产品做优化调适。采用CMOS制程并同时提供客户低成本、高性能、及高抗电磁干扰等显着优势。而MTP作为程式记忆体能让使用者更方便且有效率地开发产品。NY8BE62D 核心建立在精简指令集架构，可以很容易地做编辑和控制，共有 55 条指令。除了少数指令需要 2 个时序，大多数指令都是 1 个时序即能完成，可以让使用者轻松地以程式控制完成不同的应用。因此非常适合各种中低记忆容量但又复杂的应用。

NY8BE62D内建除错仿真电路，利用两个脚位与很少的外接硬体，就能实现在线仿真器的大多数功能，例如设定程式执行条件与中断条件，片上单步执行，以及查看及设定各种寄存器的内容。仿真的执行效果将比一般的仿真器更接近实际IC运作。

NY8BE62D内建 12 位高精度 11+2 通道类比数位转换器与高精度电压比较器，足以应付各种类比界面的侦测与量测。

NY8BE62D内建的高频振荡器，能弹性选择输出频率区段，以适用于多样的雾化器元件，更可以程式细调输出频率，细度足以应付雾化器的元件偏差。

在I/O的资源方面，NY8BE62D 有 14 根弹性的双向I/O脚，每个I/O脚都有单独的寄存器控制为输入或输出脚。而且每一个I/O脚位都有附加的程式控制功能如上拉或下拉电阻或开漏极(Open-Drain) 输出。此外针对红外线遥控的产品方面，NY8BE62D内建了可选择频率的红外载波发射口。

NY8BE62D 有四组计时器，可用系统频率当作一般的计时的应用或者从外部讯号触发来计数。另外NY8BE62D 提供 5 组 10 位解析度的PWM输出，1 组蜂鸣器输出可用来驱动马达、LED、或蜂鸣器等等。NY8BE62D的计时器同时具有增强型捕捉/比较/可编程死区时间的PWM模块(ECCP)。

NY8BE62D 采用双时钟机制，高速振荡或者低速振荡都可以分别选择内部RC振荡或外部晶振输入。在双时钟机制下，NY8BE62D 可选择多种工作模式如正常模式(Normal)、慢速模式(Slow mode)、待机模式(Standby mode) 与睡眠模式(Halt mode)可节省电力消耗延长电池寿命。并且微控制器在使用内部RC高速振荡时，低速振荡可以同时使用外部精准的晶振计时，可以维持高速处理同时又能精准计算真实时间。

在省电的模式下如待机模式(Standby mode) 与睡眠模式(Halt mode)中，有多种事件可以触发中断唤醒NY8BE62D 进入正常操作模式(Normal) 或 慢速模式(Slow mode) 来处理突发事件。

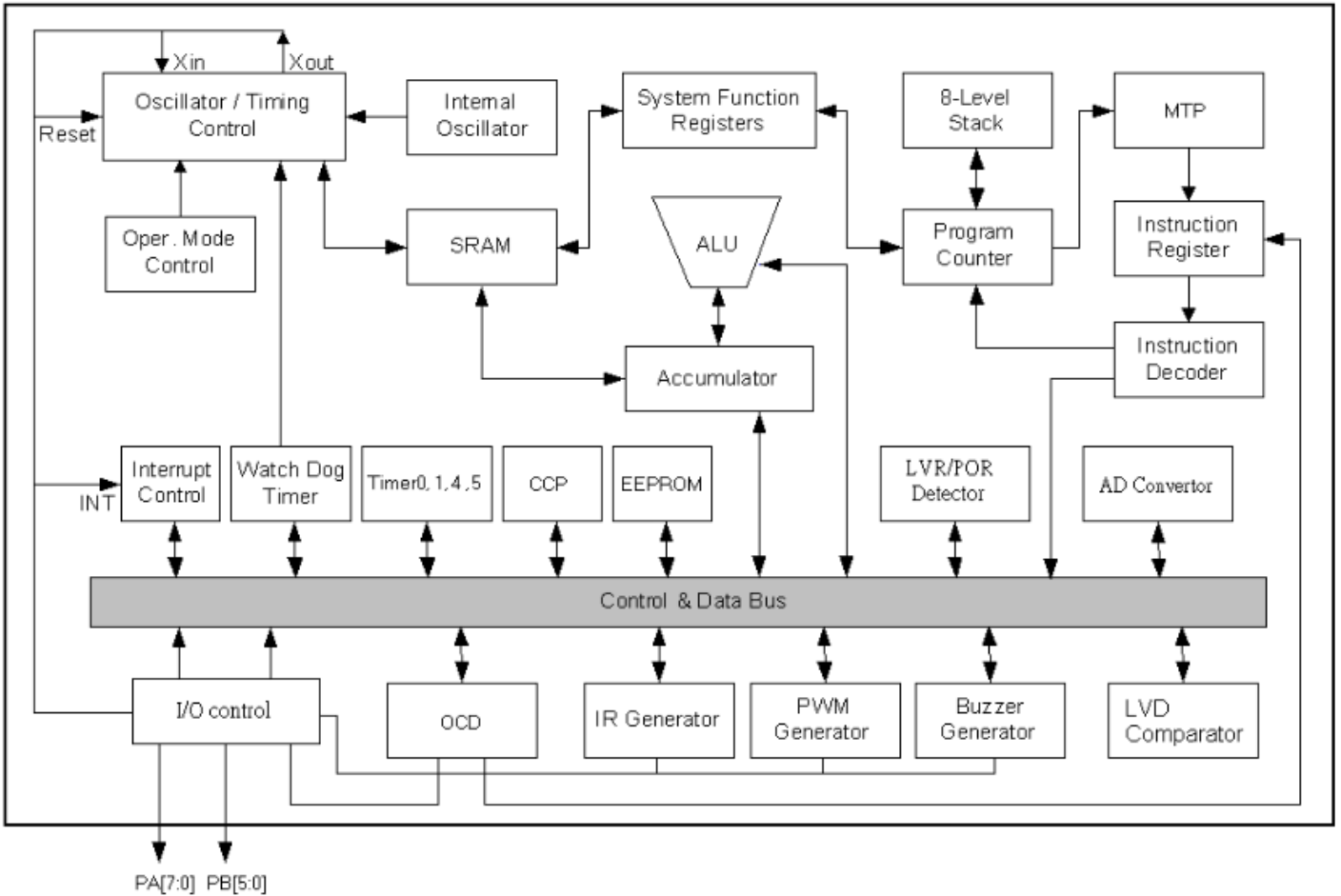
NY8BE62D能通过初始设定实现与NY8B060E相容的功能，可使用NY8BE62D内建除错仿真电路来模拟实际IC动作。

1.1 功能

- 系统主频提供 20.8M/19.2M/16M/14.4M/13.6M 五种选择。
- Timer 1/4/5 可选择系统主频直接输入。
- 高频率振荡器(F_{HOSC})可微调, 微调精度至 $\pm 0.1\%$ 。
- 内建二线控制的除错仿真电路(On Chip Debug)。
- 杂讯过滤功能(Noise Filter)打开时可容忍超过 $\pm 4KV$ 的EFT。(操作电压@5V)
- 宽广的工作电压:
 - 2.0V ~ 5.5V @系统频率 $\leq 8MHz$ 。
 - 2.2V ~ 5.5V @系统频率 $> 8MHz$ 。
- 宽广的工作温度: $-20^{\circ}C \sim 85^{\circ}C$ 。
- 2Kx14 bits MTP。
- 256 bytes EEPROM。
- 128 bytes SRAM。
- 14 根可分别单独控制输入输出方向的I/O脚(GPIO)、PA[7:0]、PB[5:0] 。
- PA[5:0]及PB[3:0]可选择输入时使用内建下拉电阻。
- PA[7:0]及PB[5:0]可选择输入时使用上拉电阻, 上拉电阻为 100K Ω 。
- PB[5:0]可选择开漏极输出(Open-Drain)。
- 所有I/O脚输出灌电流(Sink Current)有三档可选: 小(Small)、一般(Normal)或大(Large)。
- 所有I/O脚输出推电流(Drive Current)可选择小(Small)或一般(Normal)。
- 8 层程式堆栈(Stack)。
- 存取资料有直接或间接定址模式。
- 一组 8 位上数计时器(Timer0)包含可程式化的频率预除线路。
- 三组 10 位下数计时器(Timer1, 4, 5)可选重复载入或连续下数计时。
- 五个 10 位脉冲宽度调变(PWM1, 2, 3, 4, 5)。
- Timer4, 5 可组成 16 位捕捉/比较功能, Timer5/PWM5 可转变成具死区控制的PWM输出。
- 一个蜂鸣器输出(BZ1)。
- 38/57KHz红外线载波频率可供选择, 同时载波之极性也可以根据数据作选择。
- 内建准确的低电压侦测电路(LVD)。
- 内建 11+2 通道 12 位类比数位转换器(Analog to Digital Converter)。
- 内建准确的电压比较器(Voltage Comparator)。
- 内建上电复位电路(POR)。

- 内建低压复位功能(LVR)。
- 内建看门狗计时(WDT)，可由程式韧体控制开关。
- 内建电阻频率转换器(RFC)功能。
- 双时钟机制，系统可以随时切换高速振荡或者低速振荡。
 - 高速振荡：E_HXT (超过 6MHz外部高速石英振荡)
E_XT (455K~6MHz外部石英振荡)
I_HRC (1~20.8MHz内部高速RC振荡)
 - 低速振荡：E_LXT (32KHz外部低速石英振荡)
I_LRC (内部 32KHz低速RC振荡)
- 四种工作模式可随系统需求调整电流消耗：正常模式(Normal)、慢速模式(Slow mode)、待机模式(Standby mode) 与 睡眠模式(Halt mode)。
- 十二种硬体中断：
 - Timer0 溢位中断。
 - Timer1 借位中断。
 - Timer4 借位中断。
 - Timer5 借位中断 或 CCP中断。
 - WDT 中断。
 - PA/PB 输入状态改变中断。
 - 三组外部中断输入。
 - 低电压侦测/比较器输出转态中断。
 - 类比数位转换完成中断°
 - EEPROM写入完成中断°
- NY8BE62D在待机模式(Standby mode)下的十一种唤醒中断：
 - Timer0 溢位中断。
 - Timer1 借位中断。
 - Timer4 借位中断。
 - Timer5 借位中断 或 CCP中断。
 - WDT 中断。
 - PA/PB 输入状态改变中断。
 - 三组外部中断输入。
 - 低电压侦测/比较器输出转态中断。
 - 类比数位转换完成中断。
- NY8BE62D在睡眠模式(Halt mode)下的五种唤醒中断：
 - WDT 中断。
 - PA/PB 输入状态改变中断。
 - 三组外部中断输入。

1.2 系统框图



1.3 引脚图

NY8BE62D提供三种封装类型：SOP16、SOP14 及SOP8。

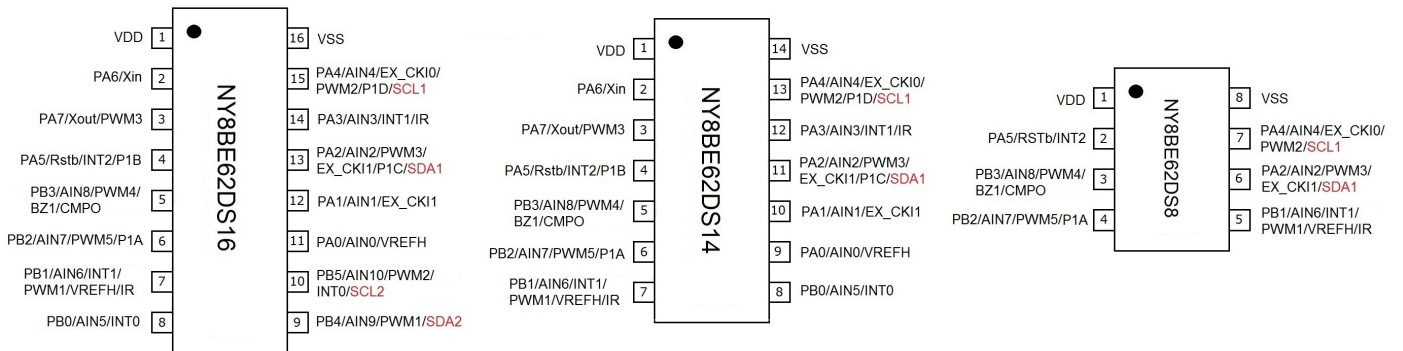


图 1 封装引脚图

1.4 引脚说明

引脚名	I/O	描述
PA0 AIN0 VREFH	I/O	PA0是一个双向I/O引脚,也可当作比较器输入引脚。 PA0可作ADC的模拟输入引脚AIN0。 PA0可当作ADC外部参考电压输入引脚VREFH。
PA1 AIN1 EX_CK11	I/O	PA1是一个双向I/O引脚,也可当作比较器输入引脚。 PA1可作ADC的模拟输入引脚。 PA1可作定时器4/5外部时钟来源EX_CK11。
PA2 AIN2 PWM3 EX_CK11 P1C SDA1	I/O	PA2是一个双向I/O引脚,也可当作比较器输入引脚。 PA2可作ADC的模拟输入引脚AIN2。 PA2可输出PWM3。 PA2可作定时器4/5外部时钟来源EX_CK11。 PA2可作CCP模式的P1C输出脚。 PA2也是编程数据输入SDA1。
PA3 AIN3 INT1 IR	I/O	PA3是一个双向I/O引脚,也可当作比较器输入引脚。 PA3可作为ADC的模拟输入引脚AIN3。 PA3可作为外部中断INT1的输入引脚。 PA3也是IR输出口。
PA4 AIN4 PWM2 EX_CK10 P1D SCL1	I/O	PA4是一个双向I/O引脚。 PA4可作为ADC的模拟输入引脚AIN4。 PA4可当作定时器0 / 1外部时钟来源EX_CK10。 PA4可当作CCP模式的P1D输出脚。 PA4也是编程输入SCL1。
PA5 Rstb INT2 P1B	I/O	PA5是一个双向I/O引脚。 PA5可当作复位引脚RSTb。 PA5可作为外部中断INT2的输入引脚。 PA5可当作CCP模式的P1B输出脚。
PA6 Xin	I/O	PA6是一个双向I/O引脚。 PA6可当作晶振输入引脚Xin。
PA7 Xout PWM3	I/O	PA7是一个双向I/O引脚。 PA7可当作晶振输出引脚Xout。 PA7也可以当成指令时钟输出。
PB0 AIN5 INT0	I/O	PB0是一个双向I/O引脚。 PB0可作为ADC的模拟输入引脚AIN5。 PB0也可当作外部中断0的输入引脚INT0。
PB1 AIN6 INT1 PWM1 VREFH IR	I/O	PB1 是一个双向I/O引脚。 PB1 可作为ADC的模拟输入引脚AIN6。 PB1 可当作外部中断 1 的输入引脚INT1。 PB1 可输出PWM1。 PB1 可当作ADC外部参考电压输入引脚VREFH。 PB1 可作为红外载波输出引脚。

引脚名	I/O	描述
PB2 AIN7 PWM5 P1A	I/O	PB2 是一个双向I/O引脚。 PB2 可作为ADC的模拟输入引脚AIN7。 PB2 可输出PWM5。 PB2 可当作CCP模式的P1A输出脚。
PB3 AIN8 PWM4 BZ1 CMPO	I/O	PB3 是一个双向I/O引脚。 PB3 可作为ADC的模拟输入引脚AIN8。 PB3 可作为PWM4 输出。 PB3 可作为BZ1 输出。 PB3 可作为比较器输出。
PB4 AIN9 PWM1 SDA2	I/O	PB4 是一个双向I/O引脚。 PB4 可作为ADC的模拟输入引脚AIN9。 PB4 可作为PWM1 输出。 PB4 可作为编程数据输入SDA2。
PB5 AIN10 PWM2 INT0 SCL2	I/O	PB5 是一个双向I/O引脚。 PB5 可作为ADC的模拟输入引脚AIN10。 PB5 可作为PWM1 输出。 PB5 可当作外部中断 0 的输入引脚INT0。 PB5 可作为编程数据输入SCL2。
VDD	-	电源正端。
VSS	-	电源负端。

2. 内存结构

NY8BE62D存储器分为两类：分别是程序存储器和数据存储器。数据存储器又分为寄存器存储器和EEPROM存储器。

2.1 Program Memory

NY8BE62D程序存储器空间是 2Kx14 位。因此，11 位的程序计数器（PC）可以访问程序存储器的任何地址。

复位地址位于 0x000，软件中断地址位于 0x001，内部和外部硬件中断地址位于 0x008。

NY8BE62D提供GOTOA和CALLA等指令去访问程序空间的 256 个地址，LCALL和LGOTO指令访问程序空间的任何地址。

当发生子程序调用或中断情况时，下一个ROM地址写入堆栈的顶部。而当执行RET、RETIA或RETIE指令，堆栈顶部的数据会被读取并加载到程序计数器。

NY8BE62D程序存储器地址 0x7FE~0x7FF是保留地址。如果用户在这些地址写入程序可能会发生无法预期的程序执行错误。

NY8BE62D程序存储器地址 0x00E~0x00F是Preset Rolling Code地址。如果用户在不设置滚码时可当作程序区使用。

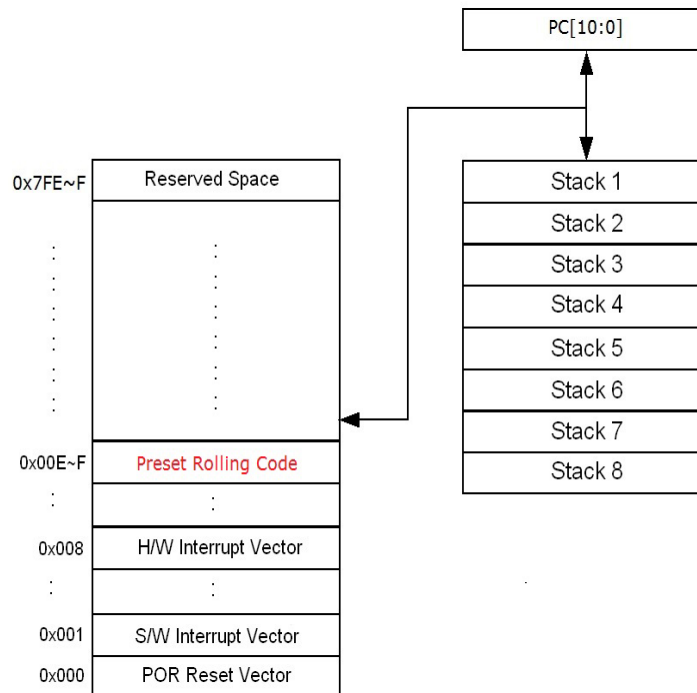


图 2 程序存储器对应地址

2.2 寄存器存储器

根据用于存取数据存储器的指令，存储器可分为三类：R-page特殊功能寄存器（SFR）和通用寄存器（GPR）、F-page特殊功能寄存器、S-page特殊功能寄存器和T-page特殊功能寄存器。R-page特殊功能寄存器又是由特殊功能寄存器（SFR）和通用寄存器（GPR）组成。GPR是由SRAM组成，用户可以使用它们来存储变量或计算结果。

R-page特殊功能寄存器和数据存储器分为四组Bank，可透过数据指针寄存器（FSR）来切换Bank。寄存器BK[1:0]为STATUS[7:6]，可从四个Bank中选择其中一个。

R-page特殊功能寄存器和数据存储器可用直接寻址方式和间接寻址方式来进行存取。

数据存储器使用间接寻址方式如下图所描述，这种间接寻址方式包含使用INDF寄存器。Bank选择是由STATUS[7:6]决定，地址选择则是由FSR[6:0]而定。

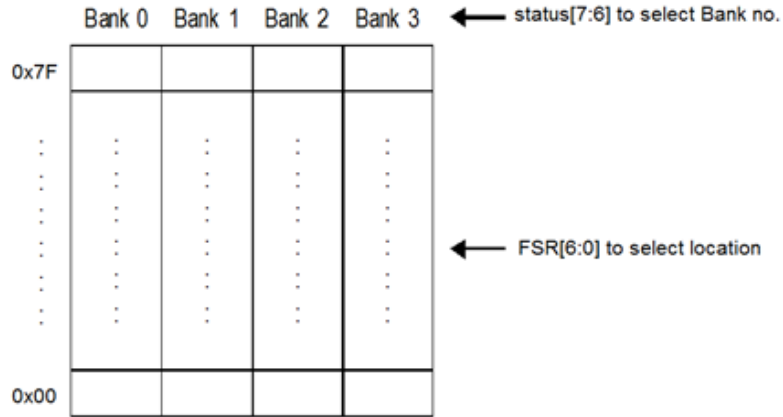


图3 间接寻址方式存取数据存储器

下面描述了数据存储器使用的直接寻址方式。Bank选择是由寄存器STATUS[7:6]决定，而地址选择则是由指令码OP-Code[6:0]直接决定。

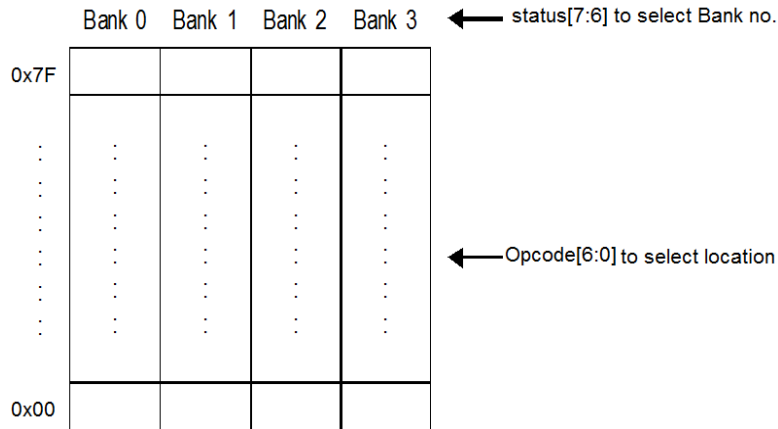


图4 直接寻址方式存取数据存储器

R-page特殊功能寄存器可以通过一般的指令存取，如算术指令和数据搬移指令。R-page特殊功能寄存器占用了从Bank 0的0x0到0x1F。然而，Bank 1、Bank 2和Bank 3的相同地址会映像到Bank 0。换句话说，R-page特殊功能寄存器只存在于Bank 0。GPR占用了Bank 0数据存储器的0x20到0x7F与Bank 1数据存储器的0x20到0x3F如表1所示。

NY8BE62D寄存器名称和R-page特殊功能寄存器的映像地址说明如下表。

Status [7:6] 地址	00 (Bank 0)	01 (Bank 1)	10 (Bank 2)	11 (Bank 3)
0x0	INDF	映射至Bank 0		
0x1	TMR0			
0x2	PCL			
0x3	STATUS			
0x4	FSR			
0x5	PORTA			
0x6	PORTB			
0x7	-			
0x8	PCON			
0x9	BWUCON			
0xA	PCHBUF			
0xB	ABPLCON			
0xC	BPHCON			
0xD	-			
0xE	INTE			
0xF	INTF			
0x10	ADMD			
0x11	ADR			
0x12	ADD			
0x13	ADVREFH			
0x14	ADCR			
0x15	AWUCON			
0x16	PACON			
0x17	ADJMD			
0x18	INTEDG			
0x19	TMRH			
0x1A	ANAEN			
0x1B	RFC			
0x1C	TM4RH			
0x1D	OSCCALH			
0x1E	OSCCALL			
0x1F	INTE2			
0x20 ~ 0x3F	通用寄存器	通用寄存器	映射至Bank 0	映射至Bank 1
0x40 ~ 0x7F	通用寄存器	映射至Bank 0	映射至Bank 0	映射至Bank 0

表 1 R-page特殊功能寄存器地址映像表

F-page特殊功能寄存器只能被指令IOST和IOSTR存取，S-page特殊功能寄存器只能被指令SFUN和SFUNR存取，T-page特殊功能寄存器只能被指令TFUN和TFUNR存取。当F-page，S-page和T-page寄存器被存取时，STATUS[7:6]选择位会被忽略。寄存器名称和F-page、S-page、T-page的地址说明如下表。

SFR Category Address	F-page SFR	S-page SFR	T-page SFR
0x0	-	TMR1	
0x1	-	T1CR1	
0x2	-	T1CR2	
0x3	-	PWM1DUTY	
0x4	-	PS1CV	
0x5	IOSTA	BZ1CR	
0x6	IOSTB	IRCR	
0x7	-	TBHP	INTE3
0x8	-	TBHD	INTF3
0x9	APHCON	-	
0xA	PS0CV	P2CR1	
0xB	-	-	
0xC	BODCON	PWM2DUTY	
0xD	-	-	
0xE	CMPCR	-	
0xF	PCON1	OSCCR	
0X10	-	-	
0X11	-	P3CR1	
0X12	-	-	
0X13	-	PWM3DUTY	
0X14	-	-	
0X15	-	TMR4	
0X16		T4CR1	
0X17		T4CR2	
0X18		PWM4DUTY	
0X19		PS4CV	
0X1A		TMR5	
0X1B		T5CR1	
0X1C		T5CR2	
0X1D		PWM5DUTY	
0X1E		PS5CV	CCPCON
0X1F		TM5RH	PWMDB

表 2 F-page特殊功能寄存器、S-page特殊功能寄存器和T-page特殊功能寄存器地址表

2.3 EEPROM Memory

读和写存取到EEPROM存储器是通过EEA，EED，EEP和EETO 4个特殊功能寄存器间接产生。EEA寄存器保存EEPROM的存取地址。EED寄存器保存EEA地址中的读和写的的数据。EEP保存写入EEPROM的未锁定的数据。当EEPROM写入超时的EEWR_TO启用时，EETO寄存器保存EEPROM写入的超时信息。以下图表显示EEPROM的运行方式。

在这4个特殊寄存器和256位EEPROM之间，提供7个指令来控制数据流。EEAR/EEAW是用来读和写EEA寄存器的指令。EEDR/EEDW是用来读和写EED寄存器的指令。EERD/EEWR/EEPL指令，另一方面，是通过EEA提供的EEPROM地址来控制EEPROM与EED寄存器之间的数据流。当发出EEWR指令时，EETO寄存器将会被设定。如果EEPROM写入时间比EETO设置的超时时间还长，则会停止EEPROM的写入，并设置一个超时标志（PCON[1]）。

EEPROM指令的描述如下表所示。

Mnemonic Operands	Description	Status affected
EEAR	从EEA写入ACC。	-
EEAW	从ACC写入EEA。	-
EEDR	从EED写入ACC。	-
EEDW	从ACC写入EED。	-
EERD	带EEA地址读EEPROM，并写入EED寄存器。	-
EEWR	带EEA地址和EED数据写EEPROM，从ACC写入EETO。	-
EEPL	写特殊代码到解锁/锁住EE写保护。	-

当EEWR指令写入EEPROM数据之前，必须要有带EEPL指令的3个连续的代码写入EEP寄存器中。这3个代码是C9H，3AH和D3H。当这3个代码被写入到EEP寄存器后，EEPROM写入就被解锁或锁住。EEPROM写入在上电后被锁住。配置字节EEPL_MODE选择EEPL指令是一次型态或切换型态。只有当EEPROM写入是解锁时，EEPROM才可以被申请的EEWR指令写入。以下范例是EEPROM写入解锁的过程。

```

MOVIA  C9H
EEPL
MOVIA  3AH
EEPL
MOVIA  D3H
EEPL      ;解锁过程完成
MOVIA  01H ;设置写入超时时间。
EEWR      ;写入 EEPROM
    
```

当EEWR被成功执行和完成时，如果EEWIE被设置为1且GIE启用时，EEWIF中断将会触发。

注意：

1. 在运行程序中，如果EEPROM写入失败，为阻止不想要的停止，建议启用看门狗复位或EEPROM写入超时功能。
2. 当用户写数据到EEPROM时，建议开启LVD=2.4V功能来监控VDD。
3. 当VDD低于2.4V时，数据不能写入到EEPROM，用户可以使用LVD和LVR功能来防止写入失败。

2.3.1 EEA (EEPROM Address Register) (EEPROM 地址寄存器)

名称	SFR 类型	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EEA	EE	X	EEA[7:0]						
读/写属性		X	读/写						
初始值		X	XXXXXXXX						

EEA[6:0]: 指向EEPROM地址。EEA[6]必须是 0。该指令是由EEAR/EEAW指令来读/写。

2.3.2 EED (EEPROM Data Register) (EEPROM 数据寄存器)

名称	SFR类型	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EED	EE	EED[7:0]							
读/写属性		读/写							
初始值		XXXXXXXX							

EED[7:0]: EEPROM数据寄存器。该指令是由EEDR/EEDW指令来读/写。

2.3.3 EEPL (EEPROM write protect Register) (EEPROM 写保护寄存器)

名称	SFR 类型	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EEPL	EE	PL[7:0]							
读/写属性		写							
初始值		XXXXXXXX							

PL[7:0]: EEPROM锁住/解锁代码。EEPL指令执行写入动作。

2.3.4 EETO (EEPROM Time-Out Register) (EEPROM 超时寄存器)

名称	SFR类型	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EETO	EE	-	-	-	-	-	TO2	TO1	TO0
读/写属性		-	-	-	-	-	写	写	写
初始值		X	X	X	X	X	X	X	X

TO[2:0]: EEPROM写入超时寄存器,当发出EEWR指令开始执行计时。

超时时间如下表所示:

TO[2:0]	Time-out period
000	1ms
001	4ms
010	2ms
011	8ms
100	16ms
101	32ms
110	16ms
111	32ms

范例:

```
MOVIA    0x01    ; 设置 EE 写入超时时限为 4ms  
EEWR     ; 写入 EEPROM
```

注意:

1. 在写入 **EEPROM** 之前, 必须要先设置好 **EETO[2:0]** 的初值.
2. **EEPROM** 写入超时设定参考 **time-out reference**:
 - 如果 $V_{OPL} = 2.1V \sim 2.2V$, 设定超时时间为 **4ms**,
 - 如果 $V_{OPL} = 2.3V \sim 2.6V$, 设定超时时间为 **2ms**,
 - 如果 $V_{OPL} \geq 2.7V$, 设定超时时间为 **1ms**,

3. 功能概述

本章节将详细描述NY8BE62D的操作方式。

3.1 R-page特殊功能寄存器

3.1.1 INDF（间接寻址寄存器）

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INDF	R	0x0	INDF[7:0]							
读/写属性			读/写							
初始值			XXXXXXXX							

间接寻址寄存器并不是真的存在，而是以间接寻址模式来使用。任何指令访问间接寻址寄存器时，实际上是访问数据指针寄存器FSR所选择的寄存器。

3.1.2 TMR0（定时器 0 寄存器）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMR0	R	0x1	TMR0[7:0]							
读/写属性			读/写							
初始值			XXXXXXXX							

当读取TMR0 寄存器时，会得到定时器 0 目前计数数值。

当写入TMR0 寄存器时，会更新定时器 0 目前计数数值。

藉由设置T0MD与配置字节（Configuration Word），定时器 0 时钟源可以从指令时钟F_{INST}、外部时钟EX_CK10或低频振荡器I_LRC/E_LXT中择一。

3.1.3 PCL（程序计数器低字节）

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCL	R	0x2	PCL[7:0]							
读/写属性			读/写							
初始值			0x00							

程序计数器（PC）是一个 11 位寄存器，分高 3 位和低 8 位。当程序执行了一个指令，同时PC数值会增加，除了某些指令会直接更改PC数值。PCL寄存器可存取PC低字节（PC[7:0]），PC高字节（PC[10:8]）并不能直接存取，必须藉由PCHBUF寄存器完成存取。LGOTO指令的PC[10:0]是从指令码取得。

LCALL指令的PC[10:0]是从指令码取得，下一个PC地址（PC+1），将被存到堆栈的顶部。

3.1.4 STATUS（状态寄存器）

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
STATUS	R	0x3	BK[1]	BK[0]	GP5	/TO	/PD	Z	DC	C
读/写属性			读/写	读/写	读/写	读/写(*2)	读/写(*1)	读/写	读/写	读/写
初始值			0	0	0	1	1	X	X	X

状态寄存器包含算术/逻辑指令的结果和是否发生看门狗超时复位。

C: 进位/借位标志位。

C=1 时，加法运算有进位或减法运算无借位。

C=0 时，加法运算无进位或减法运算有借位。

DC: 半进位/半借位标志位。

DC=1 时，加法运算低四位有进位或减法运算时没有向高四位借位。

DC=0 时，加法运算低四位无进位或减法运算时有向高四位借位。

Z: 零位。

Z=1 时，算术或逻辑运算的结果是零。

Z=0 时，算术或逻辑运算的结果不为零。

/PD: 睡眠模式标志位。

/PD=1 时，上电或执行CLRWDWT指令后。

/PD=0 时，执行SLEEP指令后。

/TO: 看门狗超时标志位。

/TO=1 时，上电或执行CLRWDWT或SLEEP指令后。

/TO=0 时，发生WDT上溢。

GP5: 通用寄存器数据位。

BK[1:0]: Bank 选择位，BK[1:0]=00b 选择Bank0，BK[1:0]=01b 选择Bank1，BK[1:0]=10b 选择Bank2。

BK[1:0]=11b，选择Bank3。

(*1) 可以被SLEEP指令清除。

(*2) 可以由CLRWDWT指令设定。

3.1.5 FSR (数据指针寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FSR	R	0x4	GP7	FSR[6:0]						
读/写属性			读/写							
初始值			0	X	X	X	X	X	X	X

FSR[6:0]: 从指定Bank数据存储器的 128 个寄存器中选择一个。

GP7: 通用寄存器数据位。

3.1.6 PortA (PortA 数据寄存器)

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PortA	R	0x5	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
读/写属性			读/写							
初始值			数据锁存器值是xxxxxxx，读取值则是xxxxxxx端口值(PA7~PA0)							

读取PortA时，若特定引脚被配置为输入引脚，将得到该引脚输入状态。然而，若该引脚被配置为输出引脚，依据配置字节，得到该引脚的状态或相对应的输出数据锁存值。当写入PortA时，数据是被写入PortA的输出数据锁存器中。

3.1.7 PortB (PortB 数据寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PortB	R	0x6	-	-	PB5	PB4	PB3	PB2	PB1	PB0
读/写属性			-	-	读/写					
初始值			数据锁存器值是xxxxxx，读取值则是xxxxxx端口值 (PB5~PB0)							

读取PortB时，若特定引脚被配置为输入引脚，将得到该引脚输入状态。然而，若该引脚被配置为输出引脚，依据配置字节，得到该引脚的状态或相对应的输出数据锁存值。当写入PortB时，数据是被写入PortB的输出数据锁存器中。

3.1.8 PCON (Power 寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	R	0x8	WDTEN	/PLPA4	LV DEN	/PHPA5	LVREN	GP2	EEW_ERR	EELOCK
读/写属性			读/写						R	R
初始值			1	1	0	1	1	0	0	1

GP2: 通用寄存器数据位。

EELOCK: EEPROM写入锁住标志。

LOCK=1, EEPROM写入锁住状态。

LOCK=0, EEPROM写入解锁状态。

EEW_ERR: EEPROM写入超时标志。

TO=1, EEPROM写入超时。

TO=0, EEPROM写入未超时。

LVREN: 开启/关闭 LVR。

LVREN=1 时, 开启LVR。

LVREN=0 时, 关闭LVR。

/PHPA5: 关闭/开启PA5 上拉电阻。

/PHPA5=1 时, 关闭PA5 上拉电阻。

/PHPA5=0 时, 开启PA5 上拉电阻。

LV DEN: 开启/关闭 LVD。

LV DEN=1 时, 开启LVD。

LV DEN=0 时, 关闭LVD。

/PLPA4: 关闭/开启PA4 下拉电阻。

/PLPA4=1 时, 关闭PA4 下拉电阻。

/PLPA4=0 时, 开启PA4 下拉电阻。

WDTEN: 开启/关闭 WDT。
 WDTEN=1 时, 开启WDT。
 WDTEN=0 时, 关闭WDT。

3.1.9 BWUCON (PortB 唤醒控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BWUCON	R	0x9	-	-	WUPB5	WUPB4	WUPB3	WUPB2	WUPB1	WUPB0
读/写属性			-	-	读/写	读/写	读/写	读/写	读/写	读/写
初始值			X	X	0	0	0	0	0	0

WUPBx: 开启/关闭PBx唤醒功能, $0 \leq x \leq 5$ 。
 WUPBx=1 时, 开启PBx唤醒功能。
 WUPBx=0 时, 关闭PBx唤醒功能。

3.1.10 PCHBUF (程序计数器高字节)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCHBUF	R	0xA	-	XSPD_STP	-	-	-	PCHBUF[2:0]		
读/写属性			-	写	-	-	-	写		
初始值			X	0	X	X	X	0		

PCHBUF[2:0]: 程序计数器PC的第十个位到第八个位。
XSPD_STP: 写 1 停止外部晶振 32.768KHz起振强化功能。

3.1.11 ABPLCON (PortA/PortB 下拉电阻控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ABPLCON	R	0xB	/PLPB3	/PLPB2	/PLPB1	/PLPB0	/PLPA3	/PLPA2	/PLPA1	/PLPA0
读/写属性			读/写							
初始值			1	1	1	1	1	1	1	1

/PLPAx: 关闭/开启PAx下拉电阻, $0 \leq x \leq 3$ 。
 /PLPAx=1, 关闭PAx下拉电阻。
 /PLPAx=0, 开启PAx下拉电阻。
/PLPBx: 关闭/开启PBx下拉电阻, $0 \leq x \leq 3$ 。
 /PLPBx=1, 关闭PBx下拉电阻。
 /PLPBx=0, 开启PBx下拉电阻。

3.1.12 BPHCON (PortB 上拉电阻控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
----	--------	----	------	------	------	------	------	------	------	------

BPHCON	R	0xC	-	-	/PHPB5	/PHPB4	/PHPB3	/PHPB2	/PHPB1	/PHPB0
读/写属性			-	-	读/写	读/写	读/写	读/写	读/写	读/写
初始值			X	X	1	1	1	1	1	1

/PHPBx: 关闭/开启PBx上拉电阻, $0 \leq x \leq 5$ 。

/PHPBx=1 时, 关闭PBx上拉电阻。

/PHPBx=0 时, 开启PBx上拉电阻。

3.1.13 INTE (中断使能寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTE	R	0xE	INT1IE	WDTIE	-	LVD CMPIE	T1IE	INT0IE	PABIE	T0IE
读/写属性			读/写	读/写	-	读/写	读/写	读/写	读/写	读/写
初始值			0	0	X	0	0	0	0	0

T0IE: 定时器 0 上溢 (overflow) 中断使能位。

T0IE=1 时, 开启定时器 0 上溢中断。

T0IE=0 时, 关闭定时器 0 上溢中断。

PABIE: PortA / PortB输入状态变化中断使能位。

PABIE=1 时, 开启PortA/ PortB输入状态变化中断。

PABIE=0 时, 关闭PortA/ PortB输入状态变化中断。

INT0IE: 外部中断 0 使能位。

INT0IE=1 时, 开启外部中断 0。

INT0IE=0 时, 关闭外部中断 0。

T1IE: 定时器 1 下溢 (underflow) 中断使能位。

T1IE=1 时, 开启定时器 1 下溢中断。

T1IE=0 时, 关闭定时器 1 下溢中断。

LVD CMPIE: 低电压侦测中断使能位。

LVD CMPIE =1 时, 开启低电压侦测中断。

LVD CMPIE =0 时, 关闭低电压侦测中断。

WDTIE: WDT上溢中断使能位。

WDTIE=1 时, 开启WDT上溢中断。

WDTIE=0 时, 关闭WDT上溢中断。

INT1IE: 外部中断 1 使能位。

INT1IE=1 时, 开启外部中断 1。

INT1IE=0 时, 关闭外部中断 1。

3.1.14 INTF（中断标志寄存器）

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF	R	0xF	INT1IF	WDTIF	-	LVDIF	T1IF	INT0IF	PABIF	T0IF
读/写属性			读/写	读/写	-	读/写	读/写	读/写	读/写	读/写
初始值（Note*）			0	0	X	0	0	0	0	0

T0IF: 定时器 0 上溢中断标志位。

T0IF=1 时，发生定时器 0 上溢中断。

T0IF必须由程序清零。

PABIF: PortA / PortB输入状态变化中断标志位。

PABIF=1 时，发生PortA / PortB输入状态变化中断。

PABIF必须由程序清零。

INT0IF: 外部中断 0 标志位。

INT0IF=1 时，发生外部 0 中断。

INT0IF必须由程序清零。

T1IF: 定时器 1 下溢中断标志位。

T1IF=1 时，发生定时器 1 下溢中断。

T1IF必须由程序清零。

LVDIF: 低电压侦测中断标志位。

LVDIF=1，发生低电压侦测中断。

LVDIF必须由程序清零。

WDTIF: WDT超时上溢标志位。

WDTIF=1 时，发生WDT上溢中断。

WDTIF必须由程序清零。

INT1IF: 外部中断 1 标志位。

INT1IF=1 时，发生外部 1 中断。

INT1IF必须由程序清零。

注意: 当对应的INTE寄存器控制位未使能，读取中断标志是0。

3.1.15 ADMD（ADC 模式寄存器）

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADMD	R	0x10	ADEN	START	EOC	GCHS	CHS3	CHS2	CHS1	CHS0
读/写属性			读/写	写	读	读/写	读/写	读/写	读/写	读/写
初始值			0	0	1	0	0	0	0	0

ADEN: 开启/关闭ADC功能。

ADEN=1 时，开启ADC功能。

START: ADC转换启动位。

写 1 开始执行ADC转换。读取此位将得到 0。

EOC: ADC转换结束标志位。

EOC=1: ADC转换完成。可由ADR与ADD读取转换结果数据。

EOC=0: ADC转换中。

GCHS: 开启/关闭ADC总通道。

GCHS=0: 关闭所有ADC模拟输入通道。

GCHS=1: 开启所有ADC模拟输入通道。

CHS3~0: ADC模拟输入通道选择位。

0000: 选择PA0 引脚为模拟输入通道。

0001: 选择PA1 引脚为模拟输入通道。

0010: 选择PA2 引脚为模拟输入通道。

0011: 选择PA3 引脚为模拟输入通道。

0100: 选择PA4 引脚为模拟输入通道。

0101: 选择PB0 引脚为模拟输入通道。

0110: 选择PB1 引脚为模拟输入通道。

0111: 选择PB2 引脚为模拟输入通道。

1000: 选择PB3 引脚为模拟输入通道。

1001: 选择PB4 引脚为模拟输入通道。

1010: 选择PB5 引脚为模拟输入通道。

1011: 选择内部 1/4 VDD为模拟输入通道。

1100: 选择内部VSS为模拟输入通道。

3.1.16 ADR (ADC 时钟, ADC 中断标志位与 ADC 转换结果低四位数据寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADR	R	0x11	ADIF	ADIE	ADCK1	ADCK0	AD3	AD2	AD1	AD0
读/写属性			读/写	读/写	读/写	读/写	读	读	读	读
初始值			0	0	0	0	X	X	X	X

ADIF: ADC中断标志位。

ADIF=1 时, 发生ADC转换完成中断。

ADIF必须由程序清零。

ADIE: ADC中断使能位。

ADIE=1 时, 开启ADC中断。

ADIE=0 时, 关闭ADC中断。

ADCK1~0: ADC时钟选择位。

00: ADC时钟=Fcpu/16, 01: ADC时钟=Fcpu/8, 10: ADC时钟=Fcpu/4, 11: ADC时钟=Fcpu/2。

AD3~0: ADC转换结果低四位数据。

3.1.17 ADD (ADC 转换结果高八位数据寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADD	R	0x12	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4
读/写属性			读	读	读	读	读	读	读	读
初始值			1	0	0	0	0	0	0	0

AD11~4: ADC转换结果高八位数据。

3.1.18 ADVREFH (ADC 参考电压寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADVREFH	R	0x13	EVHENB	-	-	-	-	-	VHS1	VHS0
读/写属性			读/写	-	-	-	-	-	读/写	读/写
初始值			0	X	X	X	X	X	1	1

EVHENB: ADC参考电压 (VREFH) 选择控制位。

EVHENB=0: ADC参考电压由内部产生, 参考电压水平由VHS1~0 决定。

EVHENB=1: ADC参考电压由引脚PA0 提供。

VHS1~0: ADC内部参考电压选择位。

11: VREFH=VDD, 10: VREFH=4V, 01: VREFH=3V, 00: VREFH=2V。

3.1.19 ADCR (ADC 采样时间与 ADC 位数寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCR	R	0x14	-	PBCON5	PBCON4	PBCON3	SHCK1	SHCK0	ADCR1	ADCR0
读/写属性			-	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值			X	0	0	0	1	0	1	0

SHCK1~0: ADC采样时间选择位。

00: 1 个ADC时钟, 01: 2 个ADC时钟, 10: 4 个ADC时钟, 11: 8 个ADC时钟。

ADCR1~0: ADC位数选择位。

00: 8 位, 01: 10 位, 1x: 12 位。

PBCONx: PB引脚选择位, $3 \leq x \leq 5$ 。

0=PBx 作为ADC模拟输入引脚或数字I/O引脚。

1=PBx 仅作为ADC模拟输入引脚。

3.1.20 AWUCON (PortA 唤醒控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
AWUCON	R	0x15	WUPA7	WUPA6	WUPA5	WUPA4	WUPA3	WUPA2	WUPA1	WUPA0
读/写属性			读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值			0	0	0	0	0	0	0	0

WUPAx: 开启/关闭PAx唤醒功能, $0 \leq x \leq 7$ 。

WUPAx=1 时, 开启PAx唤醒功能。

WUPAx=0 时, 关闭PAx唤醒功能。

3.1.21 PACON (ADC 引脚数模控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PACON	R	0x16	PBCON2	PBCON1	PBCON0	PACON4	PACON3	PACON2	PACON1	PACON0
读/写属性			读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值			0	0	0	0	0	0	0	0

PACONx: PA引脚选择位, $0 \leq x \leq 4$ 。

0=PAx 作为ADC模拟输入引脚或数字I/O引脚。

1=PAx 仅作为ADC模拟输入引脚。

PBCONx: PA引脚选择位, $0 \leq x \leq 2$ 。

0=PBx 作为ADC模拟输入引脚或数字IO引脚。

1=PBx 仅作为ADC模拟输入引脚。

3.1.22 ADJMD (ADC offset Register) (ADC 补偿寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADJMD	R	0x17	-	-	ADJ_SIGN	ADJ[4]	ADJ[3]	ADJ[2]	ADJ[1]	ADJ[0]
读/写属性			-	-	读/写	读/写	读/写	读/写	读/写	读/写
初始值			X	X	0	0	0	0	0	0

ADJ[x]: 调校位选择, $0 \leq x \leq 4$

00000 = 补偿 0mV

11111 = 补偿 11mV

ADJ_SIGN: 调校记号位。

0 = ADC数据减小。

1 = ADC数据增加。

注意: 在应用中, 请参考NYIDE范例“ADC_Interrupt_AutoK”

3.1.23 INTEDG (外部中断控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTEDG	R	0x18	INT2DEG	EIS2	EIS1	EIS0	INT1G1	INT1G0	INT0G1	INT0G0
读/写属性			读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值			0	0	0	0	0	1	0	1

EIS2: 外部中断 2 引脚选择位。

EIS2=1 时, PA5 选择为外部中断 2 引脚。

EIS2=0 时, PA5 选择为GPIO。

EIS1: 外部中断 1 引脚选择位。

EIS1=1 时, PB1/PA3 选择为外部中断 1 引脚。

EIS1=0 时, PB1/PA3 选择为GPIO。

EIS0: 外部中断 0 引脚选择位。

EIS0=1 时, PB0/PA5 选择为外部中断 0 引脚。

EIS0=0 时, PB0/PA5 选择为GPIO。

INT1G1~0: INT1 沿触发选择位。

00: 保留, 01: 上升沿触发, 10: 下降沿触发, 11: 上升/下降沿触发。

INT0G1~0: INT0 沿触发选择位。

00: 保留, 01: 上升沿触发, 10: 下降沿触发, 11: 上升/下降沿触发。

INT2DEG: INT2 沿触发选择位。

0: 上升沿触发, 1: 下降沿触发

3.1.24 TMRH (定时器 1/2 高字节寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMRH	R	0x19	-	-	TMR19	TMR18	PWM2 DUTY9	PWM2 DUTY8	PWM1 DUTY9	PWM1 DUTY8
读/写属性			-	-	读/写	读/写	读/写	读/写	读/写	读/写
初始值			X	X	X	X	X	X	X	X

TMR19~8: 定时器 1 高 2 位。写这 2 位将覆写定时器 1 第 9 位与第 8 位重载值。

读取这 2 位将得到定时器 1 第 9 位与第 8 位目前计数值。

PWM2DUTY9~8: PWM2 占空比高 2 位。

PWM1DUTY9~8: PWM1 占空比高 2 位。

3.1.25 ANAEN (比较器使能寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ANAEN	R	0x1A	COMPEN	-	-	-	-	-	-	-
读/写属性			读/写	-	-	-	-	-	-	-
初始值			0	X	X	X	X	X	X	X

COMPEN: 开启/关闭电压比较器。

COMPEN=1 时, 开启电压比较器。

COMPEN=0 时, 关闭电压比较器。

3.1.26 RFC (电阻频率转换控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RFC	R	0x1B	RFCEN	-	-	-	PSEL[3:0]			
读/写属性			读/写	-	-	-	读/写			
初始值			0	X	X	X	0			

RFCEN: 关闭/开启RFC。

RFCEN=1, 开启RFC。

RFCEN=0, 关闭RFC。

PSEL[3:0]: 选择RFC输入引脚。

PSEL[3:0]	RFC PAD
0000	PA0
0001	PA1
0010	PA2
0011	PA3
0100	PA4
0101	PA5
0110	PA6
0111	PA7
1000	PB0
1001	PB1
1010	PB2
1011	PB3
1100	PB4
1101	PB5
1110	-
1111	-

表 3 选择RFC输入引脚

3.1.27 TM4RH (定时器 4 高字节寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TM4RH	R	0x1C	TMR49	TMR48	-	-	PWM4D9	PWM4D8	PWM3D9	PWM3D8
读/写属性			读/写	读/写	-	-	读/写	读/写	读/写	读/写
初始值			X	X	X	X	X	X	X	X

TMR49, TMR48: 定时器 4 高 2 位。写这 2 位将覆写定时器 4 第 9 位与第 8 位重载值。

读取这 2 位将得到定时器 4 第 9 位与第 8 位目前计数值。

PWM4DUTY9~8: PWM4 占空比高 2 位。

PWM3DUTY9~8: PWM3 占空比高 2 位。

3.1.28 OSCCAL (内部高频振荡微调寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSCCALH	R	0x1D	-	-	-	-	-	OSC CAL10	OSC CAL9	OSC CAL8
读/写属性			-	-	-	-	-	读/写	读/写	读/写
初始值			X	X	X	X	X	0	0	0

I_HRC 的 11 位 trim 值 bit10~bit8。

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSCCALL	R	0x1E	OSC CAL7	OSC CAL6	OSC CAL5	OSC CAL4	OSC CAL3	OSC CAL2	OSC CAL1	OSC CAL0
读/写属性			读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值			0	0	0	0	0	0	0	0

I_HRC 的 11 位 trim 值 bit7~bit0。

OSCCAL10~0: I_HRC 的 11 位 trim 值载入 OSCCAL10~OSCCAL0, 用户可在慢速模式下微调 OSCCAL10~0。

3.1.29 INTE2 (第 2 中断屏蔽寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTE2	R	0x1F	INT2IF	T4IF	-	-	INT2IE	T4IE	-	-
读/写属性			读/写	读/写	-	-	读/写	读/写	-	-
初始值			0	0	X	X	0	0	X	X

INT2IF: 外部中断 2 标志位。

INT2IF=1, 产生外部中断 2。

INT2IF 必须软件清零。

T4IF: 定时器 4 下溢中断标志位。

T4IF=1 时, 发生定时器 4 下溢中断。

T4IF必须由程序清零。

INT2IE: 外部中断 2 开启位。

INT2IE=1, 外部中断 2 开启。

INT2IE=0, 外部中断 2 关闭。

T4IE: 定时器 4 下溢 (underflow) 中断使能位。

T4IE=1 时, 开启定时器 4 下溢中断。

T4IE=0 时, 关闭定时器 4 下溢中断。

3.2 T0MD定时器 0 控制寄存器

T0MD是可读写寄存器, 但只能由指令T0MD / T0MDR存取。

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T0MD	-	-	LCKTM0	GP6	T0CS	T0CE	PS0WDT	PS0SEL[2:0]		
读/写属性			读/写							
初始值(注意*)			0	0	1	1	1	111		

PS0SEL[2:0]: 选择预分频器 0 的预分频比 (Dividing Rate)。预分频器 0 根据PS0WDT控制位决定分配给定时器 0 或WDT。当预分频器 0 被分配给WDT, 预分频比取决于选择哪种计数机制 (WDT复位或WDT中断)。

PS0SEL[2:0]	预分频比选项		
	PS0WDT=0 (定时器 0)	PS0WDT=1 (WDT 复位)	PS0WDT=1 (WDT中断)
000	1:2	1:1	1:2
001	1:4	1:2	1:4
010	1:8	1:4	1:8
011	1:16	1:8	1:16
100	1:32	1:16	1:32
101	1:64	1:32	1:64
110	1:128	1:64	1:128
111	1:256	1:128	1:256

表 4 预分频器 0 的预分频比选项

PS0WDT: 预分频器 0 分配选择。

PS0WDT=1 时, 预分频器 0 被分配到WDT。

PS0WDT=0 时, 预分频器 0 被分配到定时器 0。

注意: 在使能看门狗或定时器中断前, 要先设定PS0WDT和PS0SEL[2:0], 否则复位或中断可能导致错误触发。

T0CE: 定时器 0 外部时钟源触发沿选择。

T0CE=1 时, EX_CK10 发生上升沿信号时定时器 0 加一。

T0CE=0 时, EX_CK10 发生下降沿信号时定时器 0 加一。

注意: T0CE应用在外部 EX_CK10 引脚作为定时器 0 时钟源。

T0CS: 定时器 0 时钟源选择。

T0CS=1 时，选择EX_CKIO 引脚或低频振荡I_LRC / E_LXT。

T0CS=0 时，选择指令时钟F_{INST}。

GP6: 通用读写器寄存器位。

LCKTM0: T0CS=0 时，指令时钟F_{INST}被选作定时器 0 时钟源。

T0CS=1 时，LCKTM0=0 时，外部EX_CKIO 引脚被选择当作定时器 0 时钟源。

T0CS=1 时，LCKTM0=1 时，低频振荡I_LRC / E_LXT为定时器 0 时钟源。

注意：有关定时器 0 时钟源选择的详细说明，请参考定时器 0 章节。

3.3 F-page 殊功能寄存器

3.3.1 IOSTA (PortA I/O 控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOSTA	F	0x5	IOPA7	IOPA6	IOPA5	IOPA4	IOPA3	IOPA2	IOPA1	IOPA0
读/写属性			读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值			1	1	1	1	1	1	1	1

IOPAx: PAx I/O模式选择， $0 \leq x \leq 7$ 。

IOPAx=1 时，PAx设为输入口。

IOPAx=0 时，PAx设为输出口。

3.3.2 IOSTB (PortB I/O 控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOSTB	F	0x6	-	-	IOPB5	IOPB4	IOPB3	IOPB2	IOPB1	IOPB0
读/写属性			-	-	读/写	读/写	读/写	读/写	读/写	读/写
初始值			X	X	1	1	1	1	1	1

IOPBx: PBx I/O模式选择， $0 \leq x \leq 5$ 。

IOPBx=1 时，PBx设为输入口。

IOPBx=0 时，PBx设为输出口。

3.3.3 APHCON (PortA 上拉电阻控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
APHCON	F	0x9	/PHPA7	/PHPA6	/PLPA5	/PHPA4	/PHPA3	/PHPA2	/PHPA1	/PHPA0
读/写属性			读/写							
初始值			1	1	1	1	1	1	1	1

/PHPAx: 关闭/开启 PAx上拉电阻， $x=0\sim4, 6\sim7$ 。

/PHPAx=1 时，关闭 PAx上拉电阻。

/PHPAx=0 时，开启 PAx上拉电阻。

/PLPA5: 关闭/开启下拉电阻PA5。

/PLPA5=1 时，关闭 PA5 下拉电阻。

/PLPA5=0 时，开启 PA5 下拉电阻。

注意: PA6 与 PA7 作为晶振引脚时应关闭内置上拉电阻，否则振荡失败。

3.3.4 PS0CV (预分频器 0 寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS0CV	F	0xA	PS0CV[7:0]							
读/写属性			R							
初始值			1	1	1	1	1	1	1	1

读取PS0CV时，会得到预分频器 0 寄存器的目前计数值。

3.3.5 BODCON (PortB 开漏控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BODCON	F	0xC	-	-	ODPB5	ODPB4	ODPB3	ODPB2	ODPB1	ODPB0
读/写属性			-	-	读/写	读/写	读/写	读/写	读/写	读/写
初始值			X	X	0	0	0	0	0	0

ODPBx: 开启/关闭PBx的开漏， $0 \leq x \leq 5$ 。

ODPBx=1 时，开启PBx的开漏。

ODPBx=0 时，关闭PBx的开漏。

3.3.6 CMPCR (比较器控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPCR	F	0xE	GP7	RBIAS_H	RBIAS_L	CMPF_INV	PS1	PS0	NS1	NS0
读/写属性			读/写							
初始值			0	0	0	0	1	1	0	0

NS[1:0]: 比较器负输入源选择。

NS[1:0]	负输入源
00	PA1
01	PA3
10	Bandgap (0.6V)
11	Vref

表 5 比较器负输入源选择

PS[1:0]: 比较器正输入源选择。

PS[1:0]	正输入源
00	PA0
01	PA2
10	Vref
11	---

表 6 比较器正输入源选择

CMPF_INV: 比较器输出反相控制位。

CMPF_INV = 1, 反相比较器输出。

CMPF_INV = 0, 正相比较器输出。

RBIAS_L, RBIAS_H: 设置对应的参考电压。

(请参考 3.16.1 章节)

注意: 在睡眠模式和待机模式下, **RBIAS_H** 和 **RBIAS_L** 必须要设置为 0 防止耗电。

3.3.7 PCON1 (Power 控制寄存器 1)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON1	F	0xF	GIE	LVDOUT	LVDS3	LVDS2	LVDS1	LVDS0	GP1	T0EN
读/写属性			读/写(1*)	R	读/写	读/写	读/写	读/写	读/写	读/写
初始值			0	X	1	1	1	1	0	1

T0EN: 开启/关闭定时器 0。

T0EN=1 时, 开启定时器 0。

T0EN=0 时, 关闭定时器 0。

LVDS3~0: 选择LVD电压。

LVDS[3:0]	Voltage
0000	1.9V
0001	2.0V
0010	2.2V
0011	2.4V
0100	2.6V
0101	2.8V
0110	2.9V
0111	3.0V
1000	3.15V
1001	3.30V
1010	3.45V
1011	3.60V
1100	3.75V

LVDS[3:0]	Voltage
1101	3.90V
1110	4.05V
1111	4.15V

表 7 LVD电压选择

LVDOUT: LVD输出位，只读。

GIE: 开启/关闭总中断屏蔽位。

GIE=1 时，开启总中断。

GIE=0 时，关闭总中断。

GP5, GP1: 通用寄存器数据位。

(1*): 由指令ENI设置 1、指令DISI清除、指令IOSTR所读取。

3.4 S-page 特殊功能寄存器

3.4.1 TMR1 (定时器 1 寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMR1	S	0x0	TMR1[7:0]							
读/写属性			读/写							
初始值			XXXXXXXX							

当读取TMR1 寄存器时，会得到 10 位定时器 1 中的低字节目前计数值。写TMR1 时，会将TMRH[5:4]和TMR1[7:0]一起写到定时器 1 重载寄存器中。

3.4.2 T1CR1 (定时器 1 控制寄存器 1)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T1CR1	S	0x1	PWM1OEN	PWM1OAL	TM1OE	-	TM1_HRC	T1OS	T1RL	T1EN
读/写属性			读/写	读/写	读/写	-	读/写	读/写	读/写	读/写
初始值			0	0	0	X	0	0	0	0

此寄存器用于配置定时器 1 功能。

T1EN: 开启/关闭定时器 1。

T1EN=1 时，开启定时器 1。

T1EN=0 时，关闭定时器 1。

T1RL: 当连续模式被选择 (T1OS=0)，选择定时器 1 下数方式。

T1RL=1 时，当下溢发生，定时器 1 初始值从TMR1[9:0]寄存器被重新加载。

T1RL=0 时，当下溢发生，定时器 1 继续从 0x3FF下数。

T1OS: 当下溢发生，设置定时器 1 操作模式。

T1OS=1 时，单次计数模式 (One-Shot mode)。定时器 1 会从初始值到 0x00 计数一次。

T1OS=0 时，连续计数模式（Non-Stop mode）。下溢后，定时器 1 会持续下数。

T1OS	T1RL	定时器 1 下数功能
0	0	定时器 1 从重载值下数到 0x00。 当下溢发生，0x3FF 被重载至定时器 1 并继续下数。
0	1	定时器 1 从重载的数值下数到 0x00。 当下溢发生，定时器 1 从 TMR1[9:0] 重新载入数值并继续下数。
1	x	定时器 1 从初始值下数到 0x00。 当下溢发生，定时器 1 停止下数。

表 8 定时器 1 功能

TM1_HRC: 定时器 1 时钟源选择。

TM1_HRC =1, PWM1,2,3 & Timer 1 时钟源是内部高频振荡。

TM1_HRC =0, PWM1,2,3 & Timer 1 时钟源依 T1CS 寄存器来决定。

PWM1OAL: 定义 PWM1 输出有效状态。

PWM1OAL=1 时，PWM1 为低电平有效位输出。

PWM1OAL=0 时，PWM1 为高电平有效位输出。

PWM1OEN: 开启/关闭 PWM1 输出。

PWM1OEN=1, PB1 / PB4 输出 PWM1。

PWM1OEN=0, PB1 / PB4 为 GPIO。

TM1OE: 开启/关闭定时器 1 匹配输出，当定时器 1 发生下溢时，T1OUT 切换输出。

TM1OE=1, T1OUT 输出至 PB4。

TM1OE=0, PB4 为 GPIO。

3.4.3 T1CR2 (定时器 1 控制寄存器 2)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T1CR2	S	0x2	-	-	T1CS	T1CE	/PS1EN	PS1SEL[2:0]		
读/写属性			-	-	读/写	读/写	读/写	读/写	读/写	读/写
初始值			X	X	1	1	1	1	1	1

该寄存器用于配置定时器 1 功能。

PS1SEL[2:0]: 预分频器 1 预分频比选项。

PS1SEL[2:0]	预分频比选项
000	1:2
001	1:4
010	1:8
011	1:16
100	1:32
101	1:64
110	1:128
111	1:256

表 9 预分频器 1 预分频比选项

注意：在/PS1EN=1 前须先设定PS1SEL[2:0]，否则可能会误发生中断。

/PS1EN: 关闭/开启预分频器 1。

/PS1EN=1 时，关闭预分频器 1。

/PS1EN=0 时，开启预分频器 1。

注意：当选择高频振荡器作为时钟源时，必须要设置/PS1EN=1（关闭预分频器 1）。

T1CE: 定时器 1 外部时钟触发沿选项。

T1CE=1 时，EX_CKIO 引脚下降沿时定时器 1 减一。

T1CE=0 时，EX_CKIO 引脚上升沿时定时器 1 减一。

T1CS: 定时器 1 时钟源选项。

T1CS=1 时，选择EX_CKIO 引脚作为外部时钟输入。

T1CS=0 时，选择指令时钟F_{INST}或内部高频振荡。

3.4.4 PWM1DUTY（PWM1 占空比寄存器）

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1DUTY	S	0x3	PWM1DUTY[7:0]							
读/写属性			写							
初始值			XXXXXXXX							

定时器 1 重新加载的数值储存在TMRH[5:4]与TMR1[7:0]寄存器，用来定义PWM1 帧率，TMRH[1:0]与PWM1DUTY[7:0]寄存器用于定义PWM1 的占空比。

3.4.5 PS1CV（预分频器 1 寄存器）

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS1CV	S	0x4	PS1CV[7:0]							
读/写属性			读							
初始值			1	1	1	1	1	1	1	1

读取PS1CV时，将会得到预分频器 1 的目前数值。

3.4.6 BZ1CR（蜂鸣器 1 控制寄存器）

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BZ1CR	S	0x5	BZ1EN	-	-	-	BZ1FSEL[3:0]			
读/写属性			W	-	-	-	写			
初始值			0	X	X	X	1	1	1	1

BZ1FSEL[3:0]: BZ1 输出频率选项。

BZ1FSEL[3:0]	BZ1 频率选项	
	时钟源	预分频比
0000	预分频器 1 输出	1:2
0001		1:4
0010		1:8
0011		1:16
0100		1:32
0101		1:64
0110		1:128
0111		1:256
1000		定时器 1 输出
1001	定时器 1 bit 1	
1010	定时器 1 bit 2	
1011	定时器 1 bit 3	
1100	定时器 1 bit 4	
1101	定时器 1 bit 5	
1110	定时器 1 bit 6	
1111	定时器 1 bit 7	

表 10 蜂鸣器BZ1 输出频率选项

BZ1EN: 开启/关闭蜂鸣器 1 输出。

BZ1EN=1 时，开启蜂鸣器 1。

BZ1EN=0 时，关闭蜂鸣器 1。

3.4.7 IRCR (IR 控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IRCR	S	0x6	IROSC358M	-	-	-	-	IRCSEL	IRF57K	IREN
读/写属性			写	-	-	-	-	写	写	写
初始值			0	X	X	X	X	0	0	0

IREN: 开启/关闭IR载波输出。

IREN=1 时，开启IR载波输出。

IREN=0 时，关闭IR载波输出。

IRF57K: IR载波频率选择。

IRF57K=1 时，IR载波频率是 57KHz。

IRF57K=0 时，IR载波频率是 38KHz。

IRCSEL: IR载波极性选择。

IRCSEL=0 且I/O引脚数据是 1 时，IR载波会被产生。

IRCSEL=1 且I/O引脚数据是 0 时，IR载波会被产生。

IROSC358M: 选择使用的外部晶振频率类型。若选择 I_HRC此位将被忽略。

IROSC358M=1，外部晶振频率请用 3.58MHz。

IROSC358M=0，外部晶振频率请用 455KHz。

注意:

1. 仅有高速振荡时钟 F_{Hosc} (详见章节 3.17) 可以当作 IR 时钟源。

2. 不同振荡类型的分频比。

OSC. 类型	57KHz	38KHz	条件
I_HRC(13.6MHz)	64	96	IR 模块的输入时钟为 13.6M/4=3.4MHz 3.4M/64 =53KHz , 3.4M/96 =35.4KHz
I_HRC(14.4MHz)	64	96	IR 模块的输入时钟为 14.4M/4=3.6MHz 3.6M/64 =56.3KHz , 3.6M/96 =37.5KHz
I_HRC(16M/8M/4M/2M/1MHz)	64	96	IR 模块的输入时钟为 16M/4=4MHz 4M/64 =62.5KHz , 4M/96 =41.67KHz
I_HRC (19.2MHz)	64	96	IR 模块的输入时钟为 19.2M/5=3.84MHz 3.84M/64 =60KHz , 3.84M/96 =40KHz
I_HRC(20.8MHz)	64	96	IR 模块的输入时钟为 20.8M/5=4.16MHz 4.16M/64 =65KHz , 4.16M/96 =43.3KHz
Xtal 3.58MHz	64	96	E_XT 模式 & IROSC358M=1
Xtal 455KHz	8	12	E_XT 模式 & IROSC358M=0

表 11 不同振荡类型的分频比

3.4.8 TBHP (表格指针高字节寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TBHP	S	0x7	-	-		-	-	TBHP2	TBHP1	TBHP0
读/写属性			-	-		-	-	读/写	读/写	读/写
初始值			X	X	X	X	X	X	X	X

当指令CALLA、GOTOA或TABLEA被执行时，程序计数寄存器会指向欲寻址的 11 位ROM地址，此目标地址是由TBHP[2:0]与ACC组成。ACC是PC[10:0]的低字节，TBHP[2:0]是PC[10:0]的高字节。

3.4.9 TBHD (表格数据高字节寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TBHD	S	0x8	-	-	TBHD5	TBHD4	TBHD3	TBHD2	TBHD1	TBHD0
读/写属性			-	-	读	读	读	读	读	读
初始值			X	X	X	X	X	X	X	X

当指令TABLEA被执行后，会得到ROM表格的 14 位数据内容，其中ROM表格的数据高字节内容被加载到TBHD[5:0]寄存器，ROM表格的数据低字节内容则被加载到ACC。

3.4.10 P2CR1 (PWM2 控制寄存器 1)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2CR1	S	0xA	PWM2OEN	PWM2OAL	-	-	-	-	-	-
读/写属性			读/写	读/写	-	-	-	-	-	-
初始值			0	0	X	X	X	X	X	X

PWM2OAL: 定义PWM2 输出有效状态。

PWM2OAL=1 时，PWM2 为低电平有效位输出。

PWM2OAL=0 时，PWM2 为高电平有效位输出。

PWM2OEN: 开启/关闭PWM2 输出。

PWM2OEN=1，PA4 / PB5 输出PWM2。

PWM2OEN=0，PA4 / PB5 为GPIO。

3.4.11 PWM2DUTY (PWM2 占空比寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2DUTY	S	0xC	PWM2DUTY[7:0]							
读/写属性			写							
初始值			XXXXXXXX							

定时器 1 重新加载的数值储存在TMRH[5:4]与TMR1[7:0]寄存器，以用来定义PWM2 帧率，TMRH[3:2]与PWM2DUTY[7:0]寄存器用于定义PWM2 的占空比。

3.4.12 OSCCR (振荡器控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSCCR	S	0xF	-	CMPOE	-	-	OPMD[1:0]	STPHOSC	SELHOSC	
读/写属性			-	读/写	-	-	读/写	读/写	读/写	
初始值			X	0	X	X	00	0	1	

SELHOSC: 系统振荡器选择 (F_{osc})。

SELHOSC=1 时， F_{osc} 是高频振荡器 (F_{HOSC})。

SELHOSC=0 时， F_{osc} 是低频振荡器 (F_{LOSC})。

STPHOSC: 关闭/开启高频率振荡器 (F_{HOSC})。

STPHOSC=1 时， F_{HOSC} 会停止振荡并被关闭。

STPHOSC=0 时， F_{HOSC} 保持振荡。

OPMD[1:0]: 选择操作模式。

OPMD[1:0]	操作模式
00	正常模式
01	睡眠模式
10	待机模式
11	保留

表 12 选择OPMD[1:0]的操作模式

CMPOE: 开启/关闭比较器输出到PB3 引脚。

CMPOE=1 时, 开启比较器输出到PB3 引脚。

CMPOE=0 时, 关闭比较器输出到PB3 引脚。

注意: 比较器输出到 PB3 引脚优先于 PWM1/BUZZER1。

注意: STPHOSC不能与SELHOSC或OPMD同时更改。在SELHOSC=1 时, STPHOSC不能与OPMD同时更改。

3.4.13 P3CR1 (PWM3 控制寄存器 1)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3CR1	S	0x11	PWM3OEN	PWM3OAL	-	-	-	-	-	-
读/写属性			读/写	读/写	-	-	-	-	-	-
初始值			0	0	X	X	X	X	X	X

PWM3OAL: 定义PWM3 输出有效状态。

PWM3OAL=1 时, PWM3 为低电平有效位输出。

PWM3OAL=0 时, PWM3 为高电平有效位输出。

PWM3OEN: 开启/关闭PWM3 输出。

PWM3OEN=1, PA2/PA7 输出PWM3。

PWM3OEN=0, PA2/PA7 为GPIO。

3.4.14 PWM3DUTY (PWM3 占空比寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM3DUTY	S	0x13	PWM3DUTY[7:0]							
读/写属性			写							
初始值			XXXXXXXX							

定时器 1 重新加载的数值储存在TMRH[5:4]与TMR1[7:0]寄存器, 以用来定义PWM3 帧率, TM1RH[1:0]与PWM3DUTY[7:0]寄存器用于定义PWM3 的占空比。

3.4.15 TMR4 (定时器 4 寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
----	--------	----	------	------	------	------	------	------	------	------

TMR4	S	0x15	TMR4[7:0]
读/写属性			读/写
初始值			XXXXXXXX

当读取TMR4 寄存器时，会得到 10 位定时器 4 中的低字节目前计数值。写TMR4 时，会将TMR4[7:0]写到定时器 4 重载寄存器中。

3.4.16 T4CR1（定时器 4 控制寄存器 1）

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T4CR1	S	0x16	PWM4OEN	PWM4OAL	-	-	TM4_HRC	T4OS	T4RL	T4EN
读/写属性			读/写	读/写	-	-	读/写	读/写	读/写	读/写
初始值			0	0	X	X	0	0	0	0

此寄存器用于配置定时器 4 功能。

T4EN: 开启/关闭定时器 4。

T4EN=1 时，开启定时器 4。

T4EN=0 时，关闭定时器 4。

T4RL: 当连续模式被选择（T4OS=0），选择定时器 4 下数方式。

T4RL=1 时，当下溢发生，定时器 4 初始值从TMR4[9:0]寄存器被重新加载。

T4RL=0 时，当下溢发生，定时器 4 继续从 0x3FF下数。

T4OS: 当下溢发生，设置定时器 4 操作模式。

T4OS=1 时，单次计数模式（One-Shot mode）。定时器 4 会从初始值到 0x00 计数一次。

T4OS=0 时，连续计数模式（Non-Stop mode）。下溢后，定时器 4 会持续下数。

T4OS	T4RL	定时器 4 下数功能
0	0	定时器 4 从重载值下数到 0x00。 当下溢发生，0x3FF被重载至定时器 4 并继续下数。
0	1	定时器 4 从重载的数值下数到 0x00。 当下溢发生，定时器 4 从TMR4[9:0]重新载入数值并继续下数。
1	x	定时器 4 从初始值下数到 0x00。 当下溢发生，定时器 4 停止下数。

表 13 定时器 4 功能

TM4_HRC: 定时器 4 时钟源选择。

TM4_HRC =1，PWM4 & Timer 4 时钟源是内部高频振荡。

TM4_HRC =0， PWM4 & Timer 4 时钟源依T4CS寄存器来决定。

PWM4OAL: 定义PWM4 输出有效状态。

PWM4OAL=1 时，PWM4 为低电平有效位输出。

PWM4OAL=0 时，PWM4 为高电平有效位输出。

PWM4OEN: 开启/关闭PWM4 输出。

PWM4OEN=1, PB3 输出PWM4。

PWM4OEN=0, PB3 为GPIO。

3.4.17 T4CR2 (定时器 4 控制寄存器 2)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T4CR2	S	0x17	-	-	T4CS	T4CE	/PS4EN	PS4SEL[2:0]		
读/写属性			-	-	读/写	读/写	读/写	读/写	读/写	读/写
初始值			X	X	1	1	1	1	1	1

该寄存器用于配置定时器 4 功能。

PS4SEL[2:0]: 预分频器 4 预分频比选项。

PS4SEL[2:0]	预分频比选项
000	1:2
001	1:4
010	1:8
011	1:16
100	1:32
101	1:64
110	1:128
111	1:256

表 14 预分频器 4 预分频比选项

注意: 在/PS4EN=1 前须先设定PS4SEL[2:0], 否则可能会误发生中断。

/PS4EN: 关闭/开启预分频器 4。

/PS4EN=1 时, 关闭预分频器 4。

/PS4EN=0 时, 开启预分频器 4。

注意: 当选择高频振荡器作为时钟源时, 必须要设置/PS4EN=1 (关闭预分频器 4)。

T4CE: 定时器 4 外部时钟触发沿选项。

T4CE=1 时, EX_CK11 引脚下降沿时定时器 4 减一。

T4CE=0 时, EX_CK11 引脚上升沿时定时器 4 减一。

T4CS: 定时器 4 时钟源选项。

T4CS=1 时, 选择EX_CK11 引脚作为外部时钟输入。

T4CS=0 时, 选择指令时钟F_{INST}或内部高频振荡。

3.4.18 PWM4DUTY (PWM4 占空比寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM4DUTY	S	0x18	PWM3DUTY[7:0]							
读/写属性			W							
初始值			XXXXXXXX							

定时器 4 重新加载的数值储存在TMR4[7:0]寄存器，以用来定义PWM4 帧率，PWM4DUTY[7:0]寄存器用于定义PWM4 的占空比。

3.4.19 PS4CV (预分频器 4 寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS4CV	S	0x19	PS4CV[7:0]							
读/写属性			R							
初始值			1	1	1	1	1	1	1	1

读取PS4CV时，将会得到预分频器 4 的目前数值。

3.4.20 TMR5 (定时器 5 寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMR5	S	0x1A	TMR5[7:0]							
读/写属性			读/写							
初始值			XXXXXXXX							

当读取TMR5 寄存器时，会得到 10 位定时器 1 中的低字节目前计数值。写TMR5 时，会将TM5RH[5:4]和TMR5[7:0]一起写到定时器 5 重载寄存器中。

3.4.21 T5CR1 (Timer5 Control Register1) (定时器 5 控制寄存器 1)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T5CR1	S	0x1B	PWM5OEN	PWM5OAL	-	-	TM5_HRC	T5OS	T5RL	T5EN
读/写属性			读/写	读/写	-	-	读/写	读/写	读/写	读/写
初始值			0	0	X	X	0	0	0	0

此寄存器用于配置定时器 5 功能。

T5EN: 开启/关闭定时器 5。

T5EN=1 时，开启定时器 5。

T5EN=0 时，关闭定时器 5。

T5RL: 当连续模式被选择 (T5OS=0)，选择定时器 5 下数方式。

T5RL=1 时，当下溢发生，定时器 5 初始值从TMR5[9:0]寄存器被重新加载。

T5RL=0 时，当下溢发生，定时器 5 继续从 0x3FF下数。

T5OS: 当下溢发生，设置定时器 5 操作模式。

T5OS=1 时，单次计数模式 (One-Shot mode)。定时器 5 会从初始值到 0x00 计数一次。

T5OS=0 时，连续计数模式 (Non-Stop mode)。下溢后，定时器 5 会持续下数。

T5OS	T5RL	定时器 5 下数功能
0	0	定时器 5 从重载值下数到 0x00。 当下溢发生，0x3FF被重载至定时器 5 并继续下数。

T5OS	T5RL	定时器 5 下数功能
0	1	定时器 5 从重载的数值下数到 0x00。 当下溢发生，定时器 5 从TMR5[9:0]重新载入数值并继续下数。
1	x	定时器 5 从初始值下数到 0x00。 当下溢发生，定时器 5 停止下数。

表 16 定时器 5 功能

TM5_HRC: 定时器 5 时钟源选择。

TM5_HRC = 1, PWM5 & Timer 5 时钟源是内部高频振荡。

TM5_HRC = 0, PWM5 & Timer 5 时钟源依T5CS寄存器来决定。

PWM5OAL: 定义PWM5 输出有效状态。

PWM5OAL=1 时, PWM5 为低电平有效位输出。

PWM5OAL=0 时, PWM5 为高电平有效位输出。

PWM5OEN: 开启/关闭PWM5 输出。

PWM5OEN=1, PB2 输出PWM5。

PWM5OEN=0, PB2 为GPIO。

3.4.22 T5CR2 (定时器 5 控制寄存器 2)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T5CR2	S	0x1C	-	-	T5CS	T5CE	/PS5EN	PS5SEL[2:0]		
读/写属性			-	-	读/写	读/写	读/写	读/写	读/写	读/写
初始值			X	X	1	1	1	1	1	1

该寄存器用于配置定时器 5 功能。

PS5SEL[2:0]: 预分频器 5 预分频比选项。

PS5SEL[2:0]	预分频比选项
000	1:2
001	1:4
010	1:8
011	1:16
100	1:32
101	1:64
110	1:128
111	1:256

表 15 预分频器 5 预分频比选项

注意: 在/PS5EN=1 前须先设定PS5SEL[2:0], 否则可能会误发生中断。

/PS1EN: 关闭/开启预分频器 5。

/PS5EN=1 时, 关闭预分频器 5。

/PS5EN=0 时, 开启预分频器 5。

注意：当选择高频振荡器作为时钟源时，必须要设置/PS5EN=1（关闭预分频器5）。

T5CE: 定时器 5 外部时钟触发沿选项。

T5CE=1 时，EX_CK11 引脚下降沿时定时器 1 减一。

T5CE=0 时，EX_CK11 引脚上升沿时定时器 1 减一。

T5CS: 定时器 5 时钟源选项。

T5CS=1 时，选择EX_CK11 引脚作为外部时钟输入。

T5CS=0 时，选择指令时钟F_{INST}或内部高频振荡。

3.4.23 PWM5DUTY（PWM5 占空比寄存器）

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM5DUTY	S	0x1D	PWM5DUTY[7:0]							
读/写属性			写							
初始值			XXXXXXXX							

定时器 5 重新加载的数值储存在TM5RH[5:4]与TMR5[7:0]寄存器，以用来定义PWM5 帧率，TM5RH[1:0]与PWM5DUTY[7:0]寄存器用于定义PWM5 的占空比。

3.4.24 PS5CV（预分频器 5 寄存器）

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS5CV	S	0x1E	PS5CV[7:0]							
读/写属性			读							
初始值			1	1	1	1	1	1	1	1

读取PS5CV时，将会得到预分频器 5 的目前数值。

3.4.25 TM5RH（定时器 5 高字节寄存器）

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TM5RH	S	0x1F	-	-	TMR59	TMR58	-	-	PWM5DUTY9	PWM5DUTY8
读/写属性			-	-	读/写	读/写	-	-	读/写	读/写
初始值			X	X	X	X	X	X	X	X

TMR59~8: 定时器 5 高 2 位。写这 2 位将覆写定时器 5 第 9 位与第 8 位重载值。

读取这 2 位将得到定时器 5 第 9 位与第 8 位目前计数值。

PWM5DUTY9~8: PWM5 占空比高 2 位。

3.5 T-page 特殊功能寄存器

3.5.1 INTE3 (中断使能寄存器 3)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTE3	T	0x7	-	EEIE	T5IE/CCPIE	-	-	-	-	-
读/写属性			-	读/写	读/写	-	-	-	-	-
初始值			X	0	0	X	X	X	X	X

T5IE/CCPIE: 当CCP捕捉或比较模式开启时, 该中断被用作CCP中断使能位, 否则被用作T5 下溢中断使能位。

T5IE/CCPIE=1, 开启中断。

T5IE/CCPIE=0, 关闭中断。

EEIE: EEPROM结束写入中断使能位。

EEIE=1, 开启EEPROM结束写入中断。

EEIE=0, 关闭EEPROM结束写入中断。

3.5.2 INTF3 (中断标志寄存器 3)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF3	T	0x8	-	EEIF	T5IF/CCPIF	-	-	-	-	-
读/写属性			-	读/写	读/写	-	-	-	-	-
初始值(注意*)			X	0	0	X	X	X	X	X

T5IF/CCPIF: 当CCP捕捉或比较模式开启时, 该中断被用作CCP中断标志位, 否则被用作T5 下溢中断标志位。

T5IF/CCPIF=1, 中断发生。

T5IF/CCPIF必须由程序清零。

EEIF: EEPROM结束写入中断标志位。

EEIF=1, EEPROM结束写入中断发生。

EEIF必须由程序清零。

3.5.3 CCPCON (CCP 控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCPCON	T	0x1E	PWM5M1	PWM5M0	FBCH1	FBCH0	CCPM3	CCPM2	CCPM1	CCPM0
读/写属性			读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值(note*)			0	0	0	0	0	0	0	0

CCPM[3:2]=00/01/10: 捕捉或比较模式。PB2 用作捕捉输入或比较输出。

CCPM[3:2]=11:

PWM5M1~0 = 00 → PWM单个输出。

PWM5M1~0 = 01 → PWM全桥向前输出。

PWM5M1~0 = 10 → PWM半桥输出。

PWM5M1~0 = 11 → PWM全桥反向输出。

T5IF/CCPIF=1, 中断发生。

注意: T5IF/CCPIF必须由程序清零。

CCPM[3:0]:

0000 = CCP 关闭。

0010 = 比较模式, 切换输出。

0100 = 捕捉模式, 每一个下降沿捕捉一次。

0101 = 捕捉模式, 每一个上升沿捕捉一次。

0110 = 捕捉模式, 每 4 个上升沿捕捉一次。

0111 = 捕捉模式, 每 16 个上升沿捕捉一次。

1000 = 比较模式, 匹配时设置输出中断。

1001 = 比较模式, 匹配时清除输出和中断。

1010 = 比较模式, 匹配时只有中断。

1011 = 比较模式, 匹配时触发ADC和中断。

1100 = PWM模式, P1A/P1C高有效, P1D/P1B高有效。

1101 = PWM模式, P1A/P1C高有效, P1D/P1B低有效。

1110 = PWM模式, P1A/P1C低有效, P1D/P1B高有效。

1111 = PWM模式, P1A/P1C低有效, P1D/P1B低有效。

注意: P1A是 PB2, P1B是PA5, P1C是PA2, P1D是PA4。

FBCH1~0: 全带改变方向间隙

00 = 1 CPU 循环

01 = 4 CPU 循环

1x = 16 CPU 循环

注意: 比较/捕捉模式步骤

- a. 操作模式设置定时器 5: 设置T1OS/T1RL, 设置T5EN = PWM5OEN = 0。
- b. 设置定时器 4 时钟源, 定时器 4 分频器。
- c. 设置定时器 4/定时器 5 的重载/初始值。
- d. 设置PWM4 占空比/PWM5 占空比。
- e. 开启比较/捕捉模式。
- f. 在更新PWM和定时器数据之前关闭比较/捕捉模式。

3.5.4 PWMDB (PWM 死区控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDB	T	0x1F	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
读/写属性			读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值(note*)			0	0	0	0	0	0	0	0

DB[7:0]: 定义CCP PWM模式的死区宽度。

3.6 I/O Port

NY8BE62D 提供 14 个 I/O 口 (PA[7:0]和PB[5:0])，用户可以由寄存器PORTA和PORTB读写这些引脚。每个 I/O 引脚都有一个对应的寄存器控制位以定义该引脚是输入或输出口，寄存器IOSTA[7:0]定义PA[7:0]为输入或输出口，寄存器IOSTB[5:0]定义PB[5:0]为输入或输出口。

当一个 I/O 引脚被配置为输入口，它可以由寄存器开启或关闭内部上拉/下拉电阻。寄存器APHCON[7:6]，PCON[4]和 APHCON[4:0]用于开启或关闭PA[7:0]的内部上拉电阻。寄存器APHCON[5]，PCON[6]和ABPLCON[3:0]用来开启或关闭PA[5:0]的下拉电阻。寄存器BPHCON[5:0]用于开启或关闭PB[5:0]的内部上拉电阻。寄存器ABPLCON[7:4]则是用于开启或关闭PB[3:0]的内部下拉电阻。

当一个PB 的 I/O 引脚被配置为输出口，可由寄存器开启或关闭开漏。寄存器BODCON[5:0]决定PB[5:0]是否为开漏输出引脚。

I/O口功能摘要如下表：

Feature		PA[5:0]	PA[7:6]	PB[3:0]	PB[5:4]
Input	Pull-High Resistor	V	V	V	V
	Pull-Low Resistor	V	X	V	X
Output	Open-Drain	X	X	V	V

表 16 I/O端口功能摘要

在PA和PB的每个 I/O 引脚都有输入状态改变产生中断功能。寄存器AWUCON[7:0]和BWUCON[5:0]会使能或禁止任一PA和PB引脚的唤醒功能。只要AWUCON和BWUCON对应到的任一PA和PB引脚被置为 1 时，且在此输入引脚有状态改变时，寄存器PABIF (INTF[1]) 就会被设为 1。如果寄存器PABIE (INTE[1]) 与GIE (PCON1[7]) 同时设定为 1，将发生中断要求并执行中断服务程序。

NY8BE62D提供 3 个外部中断，当寄存器EIS0 (INTEDG[4]) 设定为 1，PB0/PB5 则被当作外部中断 0 的输入引脚。当寄存器EIS1 (INTEDG[5]) 设定为 1，PB1/PA3 则被当作外部中断 1 的输入引脚。当寄存器EIS2 (INTEDG[6]) 设定为 1，PA5 则被当作外部中断 2 的输入引脚。

注意：当PB0/PB1/PA3/PB5/PA5 同时设定输入变化中断引脚与外部中断引脚，外部中断有较高的优先级，而PB0/PB1/PA3/PB5/PA5 输入变化中断引脚则会被关闭，但其它输入状态改变触发引脚不会被影响。

NY8BE62D提供红外线IR载波生成器。如下表 20 所示，IR载波生成器是由寄存器IREN（IRCR[0]）开启，并由配置字节设定IR_PAD。当IREN=1 且配置字节设定IR_PAD=1 时，IR载波由PB1 引脚输出。当配置字节设定IR电流为Large，IR输出电流约为 340mA并由PA3 引脚输出红外线载波。当IREN=0，则不产生红外线载波。当IREN=1 且IR_PAD=0 时，IR载波由PA3 引脚输出。

由配置字节决定PA5 可否当作外部复位输入RSTb。当PA5 为低电平时将导致NY8BE62D发生复位。

当配置字节设置外置晶振（E_HXT, E_XT or E_LXT）用于高速振荡时钟或低速振荡时钟时，PA6 作为晶振输入引脚（Xin），PA7 作为晶振输出引脚（Xout）。

当配置字节设置I_HRC 或 I_LRC用于高速振荡时钟或低速振荡时钟时，用户可以在PA7 输出指令时钟F_{INST}。

如果寄存器T0CS为 1 和LCK_TM0 为 0，EX_CKIO(PA4)可以当作定时器 0 外部时钟源。如果寄存器T1CS为 1，EX_CKIO(PA4)引脚可以当作定时器 1 外部时钟源。如果寄存器T4CS / T5CS为 1，EX_CKIO(PA1 或PA2)引脚可以当作定时器 4/5 外部时钟源。

如果寄存器PWM1OEN（P1CR1[7]）为 1，PB1 或PB4（依照PWM1 输出的配置字节）可以当作脉冲宽度调制PWM1 输出。如果T1CR1[5] TM1OE=1，PB4 可作T1OUT输出。PB4 的输出优先级为T1OUT > PWM1。

如果寄存器PWM2OEN（P2CR1[7]）为 1，PB4 或PB5（依照PWM2 输出的配置字节）可以当作脉冲宽度调制PWM2 输出。若寄存器BZ2EN（BZ2CR[7]）为 1，PB2 也可以当作蜂鸣器 2 输出。

如果寄存器PWM3OEN（T3CR1[7]）为 1，PA2 或PA7（依照PWM3 输出的配置字节）可以当作脉冲宽度调制PWM3 输出。若寄存器BZ3EN（BZ3CR[7]）为 1，PA2 也可以当作蜂鸣器 3 输出。

如果寄存器PWM4OEN（T4CR1[7]）为 1，PB3 可以当作脉冲宽度调制PWM4 输出。若寄存器BZ1EN（BZ1CR[7]）为 1，PB3 也可以当作蜂鸣器 1 输出。

如果寄存器PWM5OEN（T5CR1[7]）为 1，PB2 可以当作脉冲宽度调制PWM5 输出。

当IO配置为输出口，每个引脚可在配置字节设置为小灌电流（4mA@VDD=3V），一般灌电流（16mA@VDD=3V），大灌电流（23mA@VDD=3V）。每个引脚可在配置字节设置为小推电流（1.0mA@VDD=3V），一般推电流（10mA@VDD=3V），如下表 21 所示。

配置字节	小灌电流	一般灌电流	大灌电流
PXcurrent	X	0	1
PXcsc	X	0	0
Small Sink	0	1	1

配置字节	小推电流	一般推电流
Small Drive & Sink	0	1

表 17 I/O 口 推/灌电流模式选择（X=A,B）

3.6.1 IO 引脚结构框图

IO_SEL: 设定引脚为输入或输出口。

WRITE_EN: 将数据写入引脚。

READ_EN: 读取引脚状态。

PULLUP_ENB: 开启内部上拉电阻。

PULLDOWN_EN: 开启内部下拉电阻。

VPEN: 开启为比较器正输入引脚。

CMPVP: 比较器输入引脚。

RD_TYPE: 选择读取引脚或数据锁存器。

WUA: 开启PA口唤醒功能。

SET_PBIF: PB口唤醒标志。

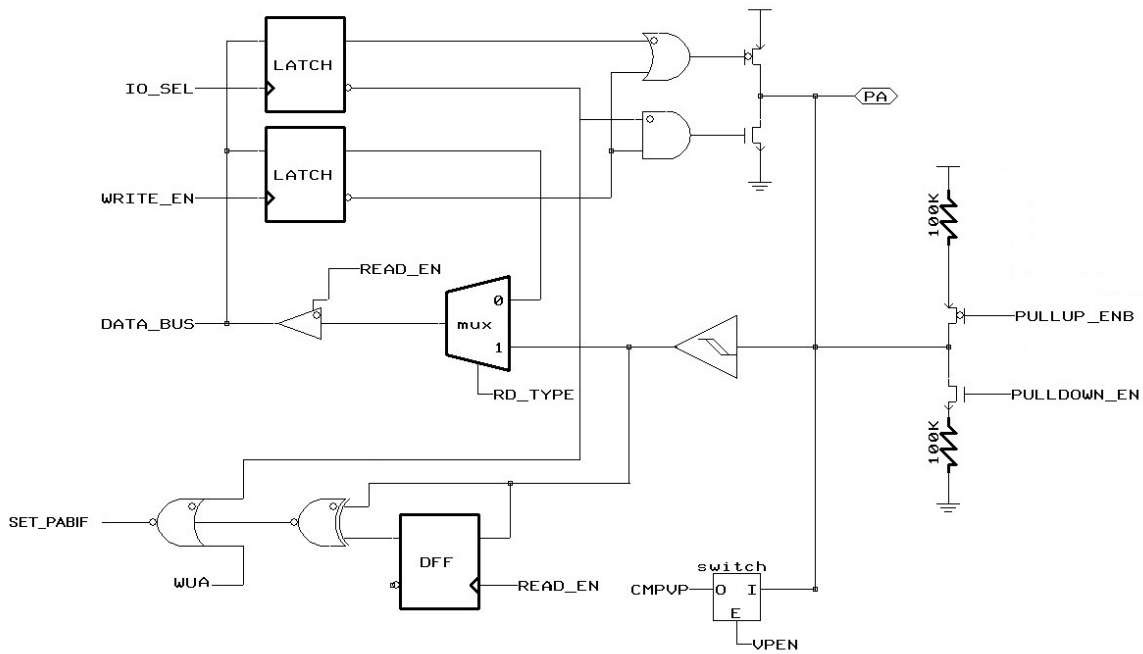


图 5 PA0, PA1, PA2 引脚结构框图

IO_SEL: 设定引脚为输入或输出口。

WRITE_EN: 将数据写入引脚。

READ_EN: 读取引脚状态。

PULLUP_ENB: 开启内部上拉电阻。

PULLDOWN_EN: 开启内部下拉电阻。

VPEN: 开启为比较器正输入引脚。

CMPVP: 比较器正输入引脚。

RD_TYPE: 选择读取引脚或数据锁存器。

WUA: PA口唤醒功能开启。

SET_PBIF: PB口唤醒标志。

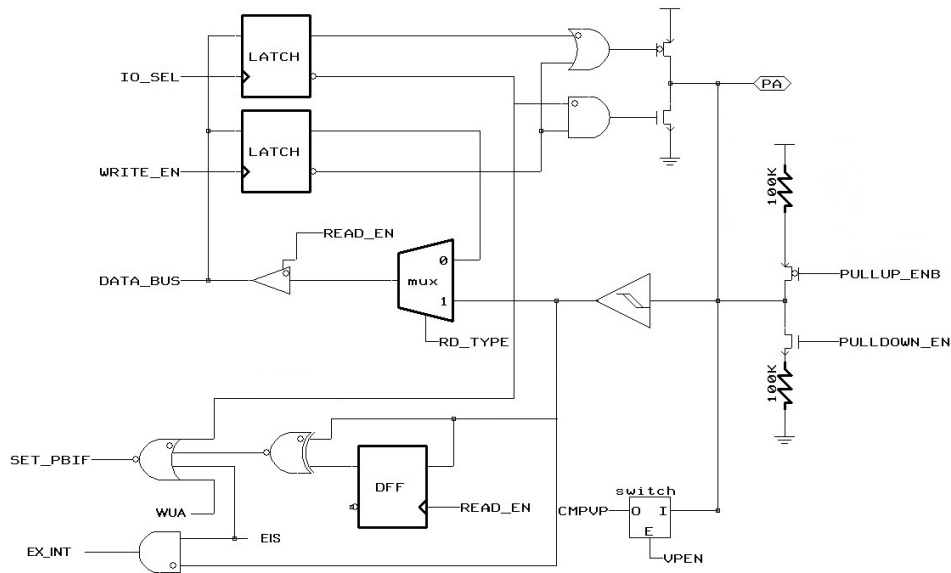


图 6 PA3 引脚结构框图

IO_SEL: 设定引脚为输入或输出口。

WRITE_EN: 将数据写入引脚。

READ_EN: 读取引脚状态。

PULLUP_ENB: 开启内部上拉电阻。

RD_TYPE: 选择读取引脚或数据锁存器。

WUA: PA口唤醒功能开启。

SET_PBIF: PB口唤醒标志。

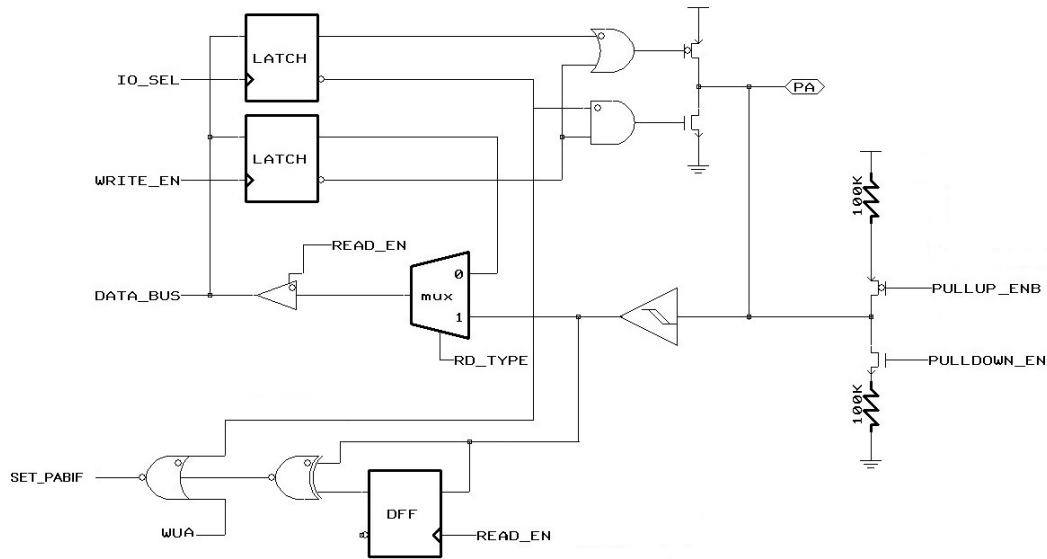


图 7 PA4 引脚结构框图

RSTPAD_EN: 开启PA5 作为复位引脚。

RSTB_IN: 位信号输入。

IO_SEL: 设定引脚为输入或输出口。

WRITE_EN: 将数据写入引脚。

READ_EN: 读取引脚状态。

PULLUP_ENB: 开启内部上拉电阻。

PULLDOWN_EN: 开启内部下拉电阻。

EIS: 开启外部中断功能。

INTEDG[7]: 外部中断边沿选择。

EX_INT: 外部中断信号。

RD_TYPE: 选择读取引脚或数据锁存器。

WUA: 开启PA口唤醒功能。

SET_PBIF: PB口唤醒标志。

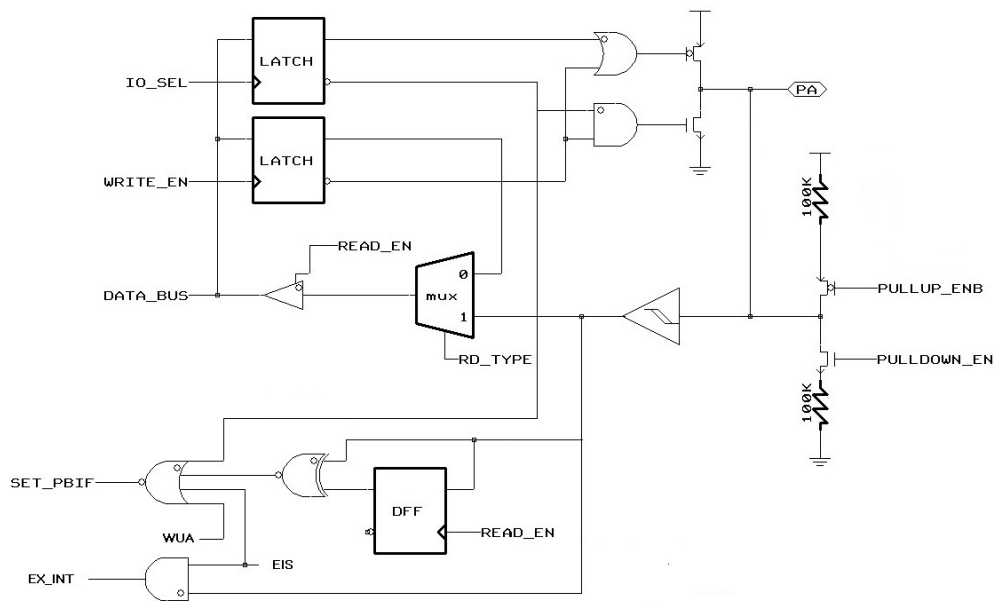
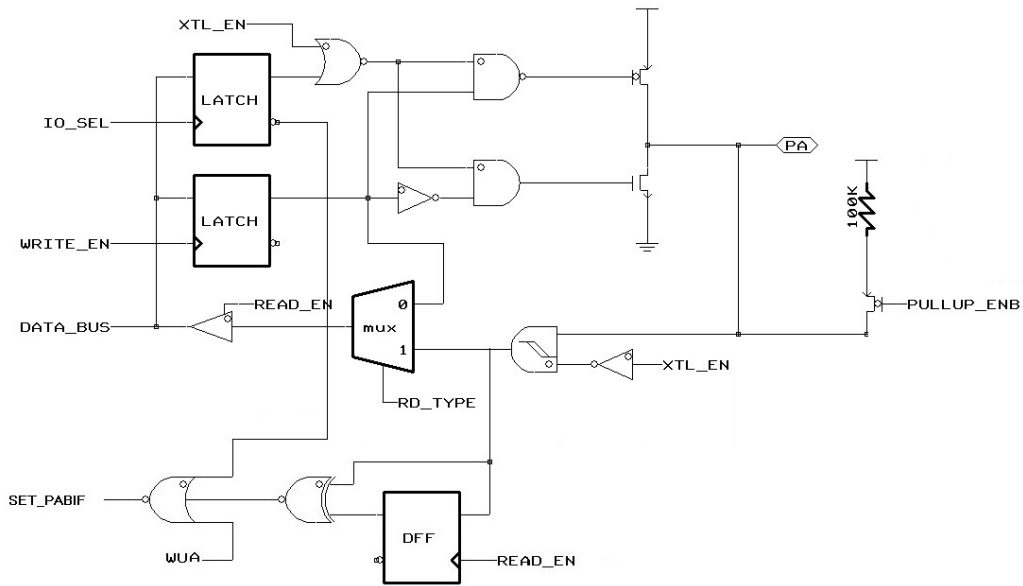


图 8 PA5 引脚结构框图

- XTL_EN: 开启晶振模式。
- IO_SEL: 设定引脚为输入或输出口。
- WRITE_EN: 将数据写入引脚。
- READ_EN: 读取引脚状态。
- PULLUP_ENB: 开启内部上拉电阻。
- RD_TYPE: 选择读取引脚或数据锁存器。
- WUA: 开启PA口唤醒功能。



SET_PBIF: PB口唤醒标志。

图9 PA6, PA7 引脚结构框图

IO_SEL: 设定引脚为输入或输出口。

WRITE_EN: 将数据写入引脚。

READ_EN: 读取引脚状态。

OD_EN: 开启开漏汲。

PULLUP_ENB: 开启内部上拉电阻。

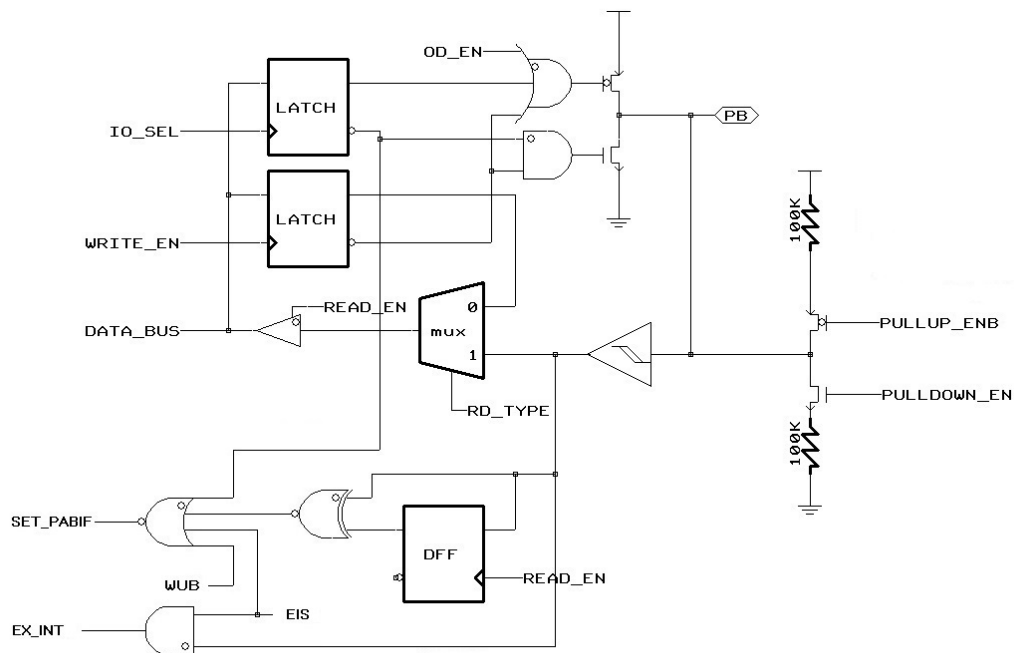
PULLDOWN_EN: 开启内部下拉电阻。

RD_TYPE: 选择读取引脚或数据锁存器。

WUB: 开启PB口唤醒功能。

SET_PBIF: PB口唤醒标志。

EIS: 开启外部中断功能。



EX_INT: 外部中断信号。

图 10 PB0, PB1 引脚结构框图

IO_SEL: 设定引脚为输入或输出口。

WRITE_EN: 将数据写入引脚。

READ_EN: 读取引脚状态。

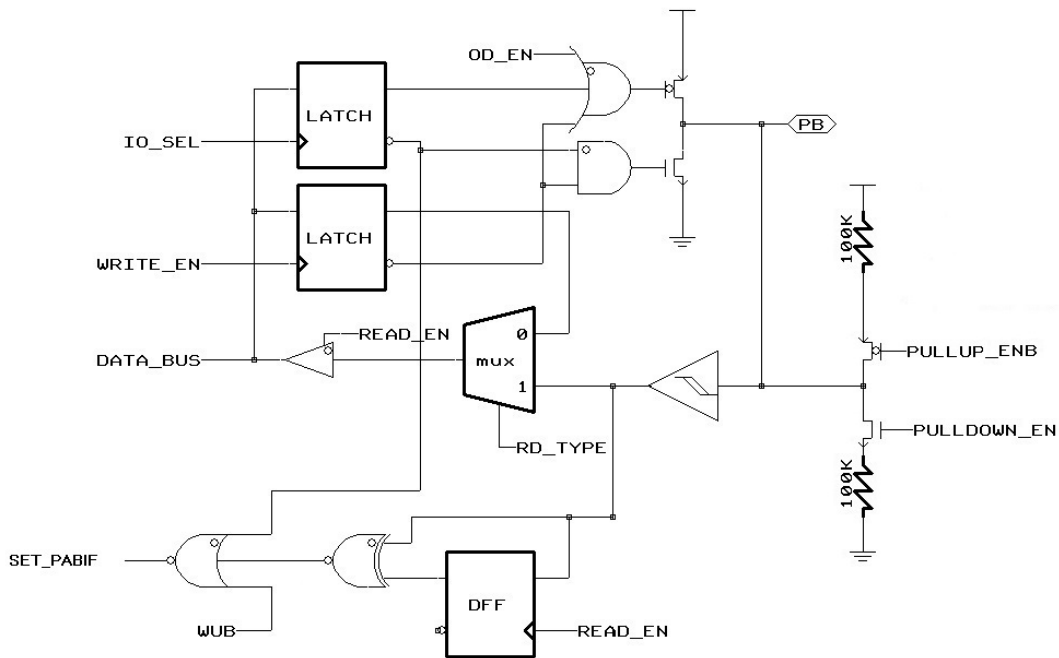
OD_EN: 开启开漏汲。

PULLUP_ENB: enable Pull-High.

PULLDOWN_EN: 开启内部上拉电阻。

RD_TYPE: 选择读取引脚或数据锁存器。

WUB: 开启PB口唤醒功能。



SET_PBIF: PB口唤醒标志。

图 11 PB2, PB3 引脚结构框图

IO_SEL: 设定引脚为输入或输出。

WRITE_EN: 将数据写入引脚。

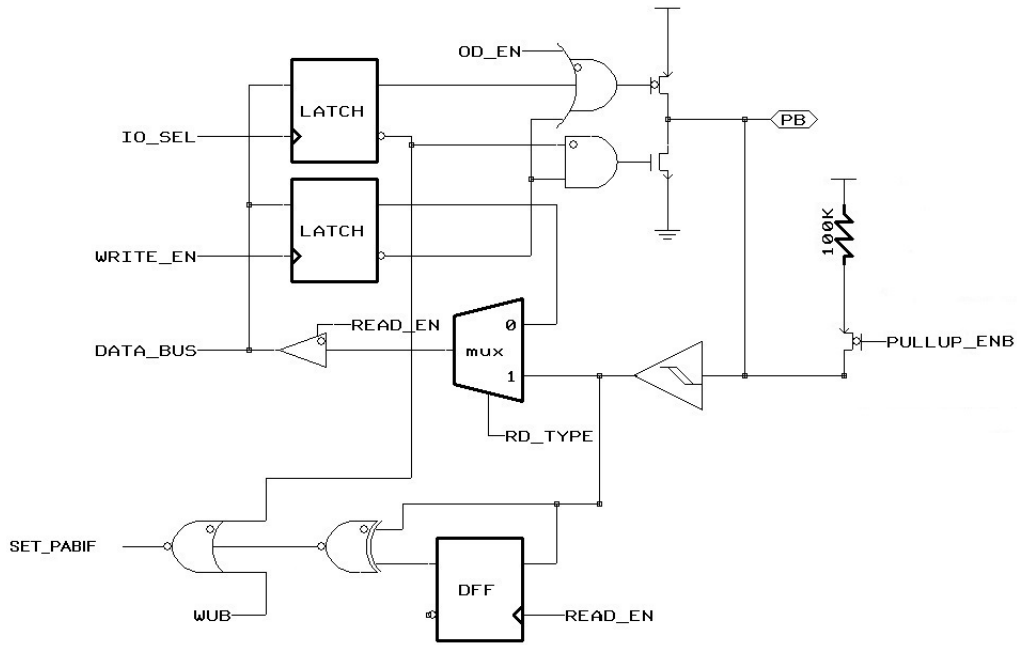
READ_EN: 读取引脚状态。

OD_EN: 开启开漏汲。

PULLUP100K_ENB: 开启 100K上拉电阻。

RD_TYPE: 选择读取引脚或数据锁存器。

WUB: 开启PB口唤醒功能。



SET_PBIF: PB口唤醒标志。

图 12 PB4 引脚结构框图

IO_SEL: 设定引脚为输入或输出口。

WRITE_EN: 将数据写入引脚。

READ_EN: 读取引脚状态。

OD_EN: 开启开漏汲。

PULLUP_ENB: 开启内部上拉电阻。

RD_TYPE: 选择读取引脚或数据锁存器。

WUB: 开启PB口唤醒功能。

SET_PBIF: PB口唤醒标志。

EIS: 开启外部中断功能。

EX_INT: 外部中断信号。

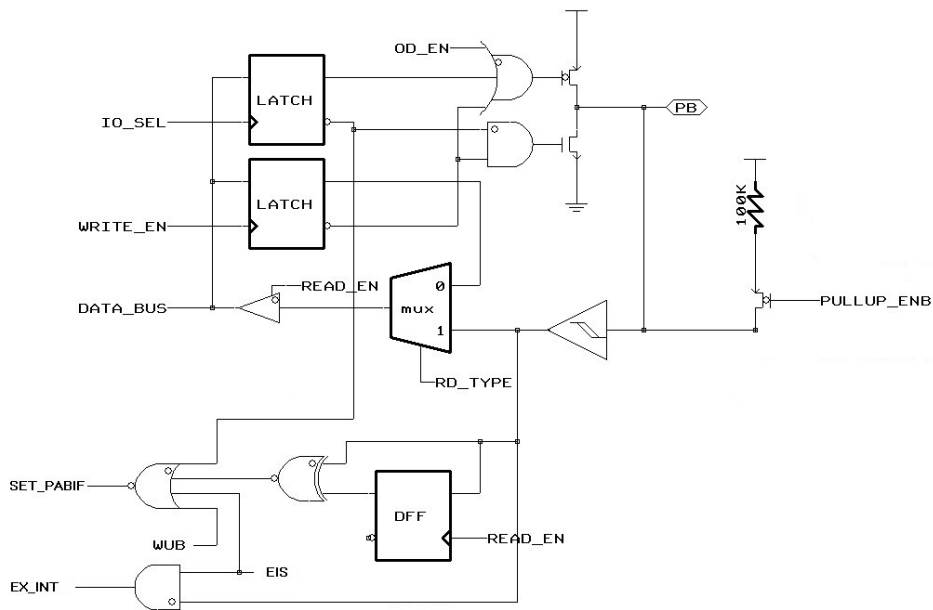


图 13 PB5 引脚结构框图

3.7 定时器 0

定时器 0 是 8 位上数定时器，由寄存器 T0EN (PCON1[0]) 开启/关闭。写入定时器 0 将会设定其初始值，读取定时器 0 时则会显示目前的计数数值。

定时器 0 的时钟源可由寄存器 T0CS (T0MD[5]) 与 LCK_TM0 (T0MD[7]) 所决定，可以从指令时钟 F_{INST}、外部时钟输入引脚 EX_CKIO 或低频振荡 I_LRC / E_LXT 中择一。当 T0CS 为 0，指令时钟会被选择当作定时器 0 时钟源。当 T0CS 为 1 且 LCK_TM0 为 0，EX_CKIO 会被当作定时器 0 时钟源。当 T0CS 是 1 且 LCK_TM0 为 1，会选择低频振荡 I_LRC / E_LXT 当作定时器 0 时钟源。汇总成表格如下。(也请参考图 18)

定时器 0 时钟源	T0CS	LCKTM0	定时器 0 来源	低频振荡
指令时钟	0	X	X	X
EX_CKIO	1	0	X	X
		X	0	
E_LXT	1	1	1	1
I_LRC	1	1	1	0

表 18 定时器 0 时钟源摘要

寄存器 T0CE (T0MD[4]) 可决定 EX_CKIO 引脚或 I_LRC / E_LXT 的时钟触发沿选择。当 T0CE 是 1，EX_CKIO 引脚或 I_LRC / E_LXT 的上升沿将让定时器 0 计数加一。当 T0CE 是 0，EX_CKIO 引脚或 I_LRC / E_LXT 的下降沿将让定时器 0 计数加一。

如果寄存器 PS0WDT (T0MD[3]) 为 0，定时器 0 时钟源可以由预分频器 0 所分频，预分频器 0 会被指定到定时器 0，且会在 PS0WDT 设为 0 时清除 Timer0 与预分频器 0。寄存器 PS0SEL[2:0] (T0MD[2:0]) 决定预分频器 0 的预分频比，其数值从 1:2 到 1:256。

当定时器 0 上溢，寄存器 T0IF (INTF[0]) 将会设定为 1，以标明定时器 0 发生上溢中断。如果寄存器 T0IE (INTE[0]) 与 GIE 都设定为 1，会发生中断的请求并执行中断服务程序。直到程序写入 0 到 T0IF，T0IF 才会被清除。

定时器 0 与 WDT 的结构框图如下图：

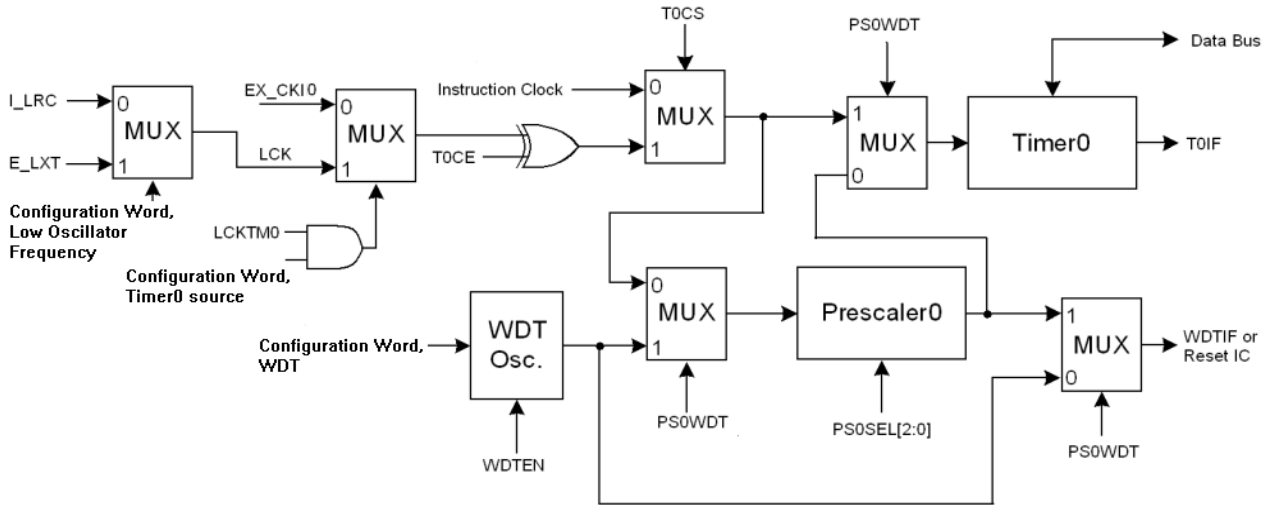


图 14 定时器 0 与WDT结构框图

3.8 定时器 1/PWM1/Buzzer1

定时器 1 是具有预分频器 1 的 10 位下数定时器，其预分频比是可编程的。定时器 1 的输出可以被用于产生 PWM1 输出与蜂鸣器 1 输出。写入定时器 1 高 2 位 (TMRH[5:4]) 再写入 TMR1 时，就可更新定时器 1 重载缓存寄存器。当 T1EN=0 时，定时器 1 重载缓存寄存器会立即写入定时器 1。当 T1EN=1 时，会等到定时器 1 下溢后，定时器 1 重载缓存寄存器才会写入定时器 1。读取寄存器 TMR1 会显示定时器 1 目前计数数值的内容。

定时器 1 的结构框图如下图所示：

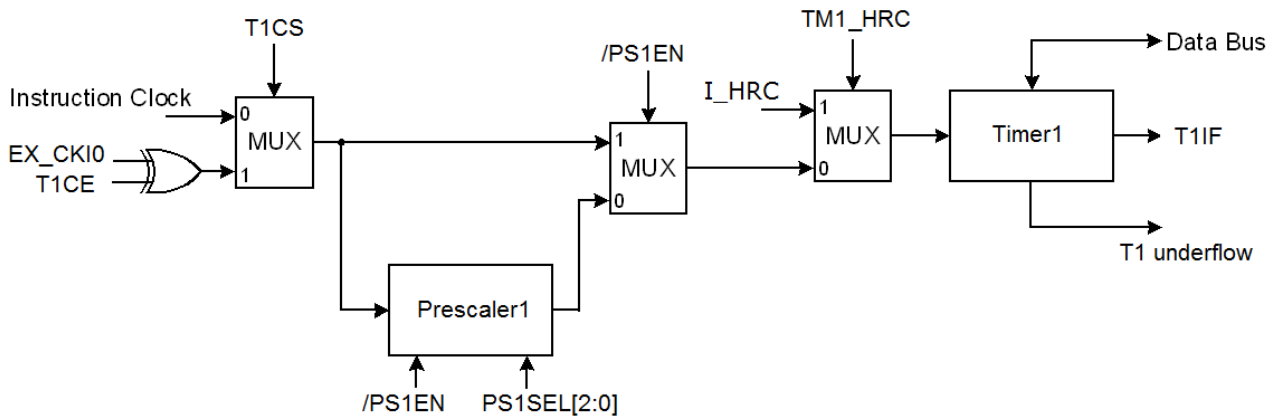


图 15 定时器 1 结构框图

定时器 1 的操作可以由寄存器 T1EN (T1CR1[0]) 开启或关闭。开启定时器 1 后，寄存器 T1CS (T1CR2[5]) 和 TM1_HRC (T1CR1[3]) 可决定时钟源是指令时钟 F_{INST}、I_HRC 或外部时钟引脚 EX_CK10。当 T1CS 为 1 和 TM1_HRC 为 0 时，EX_CK10 引脚当做时钟源。当 T1CS 为 0 和 TM1_HRC 为 0，指令时钟会被选择当做时钟源。当 TM1_HRC 为 1，I_HRC 会被选择当做时钟源。当 EX_CK10 引脚被选取，寄存器控制位 T1CE (T1CR2[4]) 可决定 EX_CK10 引脚的时钟触发沿。当 T1CE 是 1，EX_CK10 引脚的下降沿将让定时器 1 计数减一。当 T1CE 是 0，

EX_CK10 引脚的上升沿将让定时器 1 计数减一。定时器 1 时钟源可以由预分频器 1 所分频。寄存器/PS1EN (T1CR2[3]) 为 0, 可开启预分频器 1。寄存器PS1SEL[2:0] (T1CR2[2:0]) 可以决定其预分频比从 1:2 到 1:256。预分频器 1 的目前数值可以由读取寄存器PS1CV取得。

定时器 1 提供两种计数模式：单次计数与连续计数。当寄存器T1OS (T1CR1[2]) 为 1, 即为单次计数模式。定时器 1 从储存在寄存器TMR1[9:0]的初始值下数到 0x00, 当下溢发生时, 定时器 1 停止计数。当寄存器T1OS (T1CR1[2]) 为 0, 即为连续计数模式。当下溢发生, 寄存器T1RL (T1CR1[1]) 决定计数的下数方式有两种。当T1RL为 1, 定时器 1 从寄存器TMR1[9:0]重新载入数值作为初始值并继续下数。当T1RL为 0, 定时器 1 以 0x3FF 作为初始值并继续下数。

当定时器 1 下溢, 寄存器T1IF (INTF[3]) 会被设定为 1, 表明定时器 1 发生下溢中断。如果寄存器T1IE (INTE[3]) 与GIE同时设定为 1, 会发生中断请求且执行中断服务程序。直到程序写入 0 到T1IF, T1IF才会被清除。

定时器 1 时序图如下图所示：

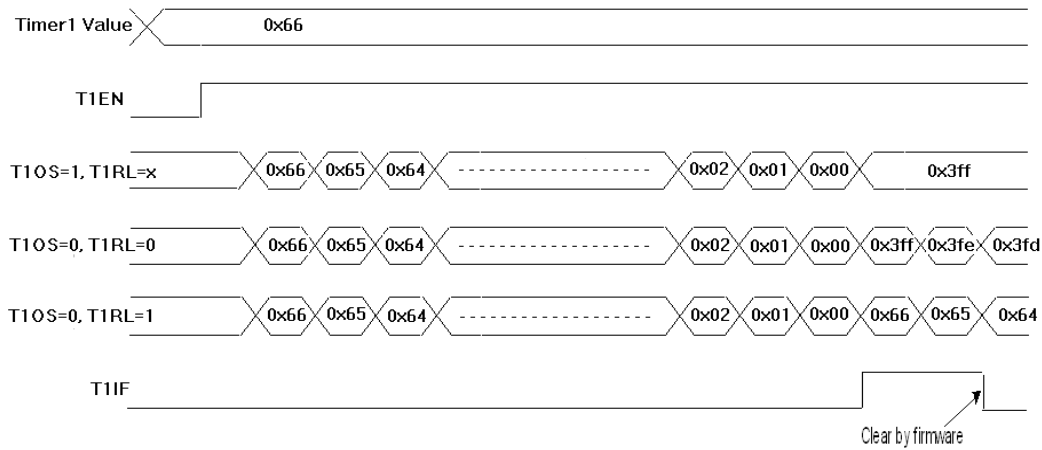


图 16 定时器 1 时序图

PWM1 可由PB1 或PB4 引脚输出。当寄存器PWM1OEN (T1CR1[7]) 设定为 1, PB3 / PB4 为PWM1 输出。当 PWM1OEN为 1, PB3 / PB4 会自动成为输出引脚。PWM1 输出的有效状态是由寄存器PWM1OAL (T1CR1[6]) 决定。当PWM1OAL为 1, PWM1 为低电平有效输出；PWM1OAL为 0, PWM1 为高电平有效输出。此外, PWM1 的占空比与帧率皆是可编程的。占空比是由寄存器TMRH[1:0]和PWM1DUTY[7:0]决定。当PWM1DUTY为 0, PWM1 无法输出占空比。当PWM1DUTY为 0x3FF, PWM1 将输出 1023/1024 的占空比 (当PWM1OAL为 0)。帧率是由TMRH[5:4] +TMR1[7:0]初始值所决定。因此, PWM1DUTY数值必须小于或等于TMRH[5:4] + TMR1[7:0]。当写入PWM1DUTY时, 先写入PWM1DUTY[9:8]高 2 位 (TMRH[1:0]) 再写入PWM1DUTY[7:0], 等到定时器 1 下溢后, 就可更新PWM1DUTY重载缓存寄存器。

PWM1 的结构框图如下：

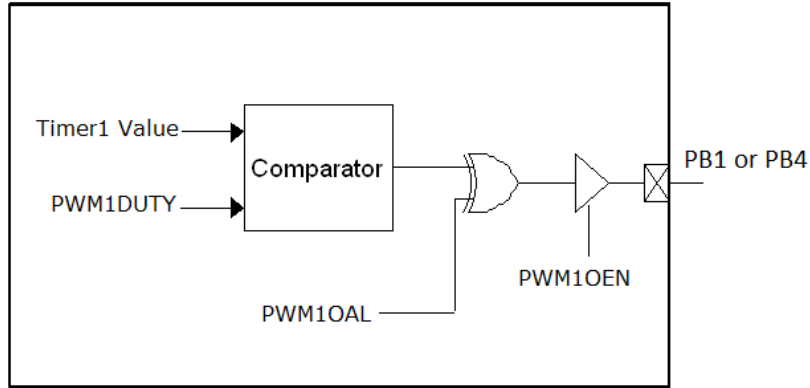


图 17 PWM1 结构框图

当寄存器BZ1EN (BZ1CR1[7]) 设定成 1, PB3 引脚为蜂鸣器 1 输出。当BZ1EN设定为 1, PB3 会自动成为输出引脚。BZ1 的频率是由寄存器BZ1FSEL[3:0] (BZ1CR[3:0]) 决定, 可以选择从定时器 1 输出或预分频器 1 输出。当BZ1FSEL[3]为 0, 预分频器 1 输出被选择来产生BZ1 输出。当BZ1FSEL[3]为 1, 定时器 1 输出被选择来产生 BZ1 输出。预分频比的范围是 1:2 到 1:256。蜂鸣器 1 结构框图如下所示:

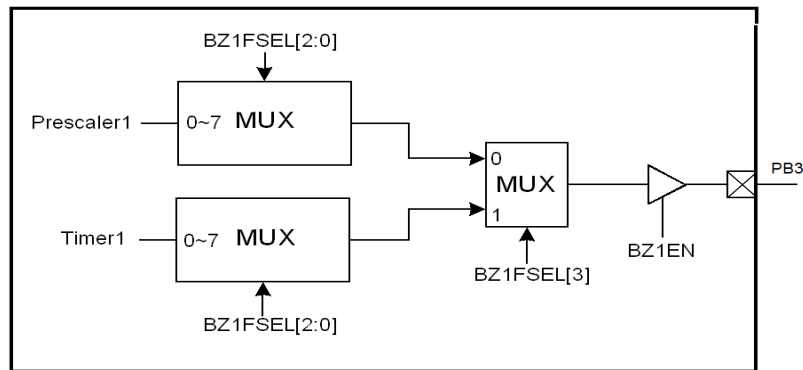


图 18 蜂鸣器 1 结构框图

注意: PB3 同时作为PWM1 和Buzzer1 输出时, 优先级为 PWM1 输出 > 蜂鸣器 1 输出。

3.9 PWM2

PWM2 可由PA4 或PB5 引脚输出。当寄存器PWM2OEN (P2CR1[7]) 设定为 1, PWM2 自动成为输出引脚。PWM2 输出的有效状态是由寄存器PWM2OAL (P2CR1[6]) 决定。当PWM2OAL为 1, PWM2 为低电平有效输出; PWM2OAL为 0, PWM2 为高电平有效输出。

此外, PWM2 的占空比与帧率皆是可编程的。占空比是由寄存器TMRH[3:2]和PWM2DUTY[7:0]决定。当 PWM2DUTY为 0, PWM2 无法输出占空比。当PWM2DUTY为 0x3FF, PWM2 将输出 1023/1024 的占空比 (当 PWM2OAL为 0)。帧率是由TMRH[5:4] +TMR1[7:0]初始值所决定。因此, PWM2DUTY数值必须小于或等于 TMR1[9:0]。当用户在写PWM2DUTY时, 先写入PWM2 DUTY[9:8]高 2 位 (TMRH[3:2]) 再写入PWM2DUTY[7:0], 等到定时器 1 下溢后, 就可更新PWM2DUTY重载缓存寄存器。

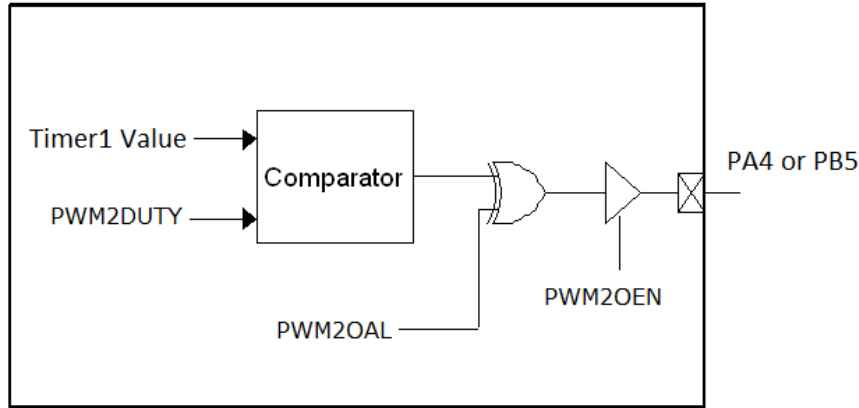
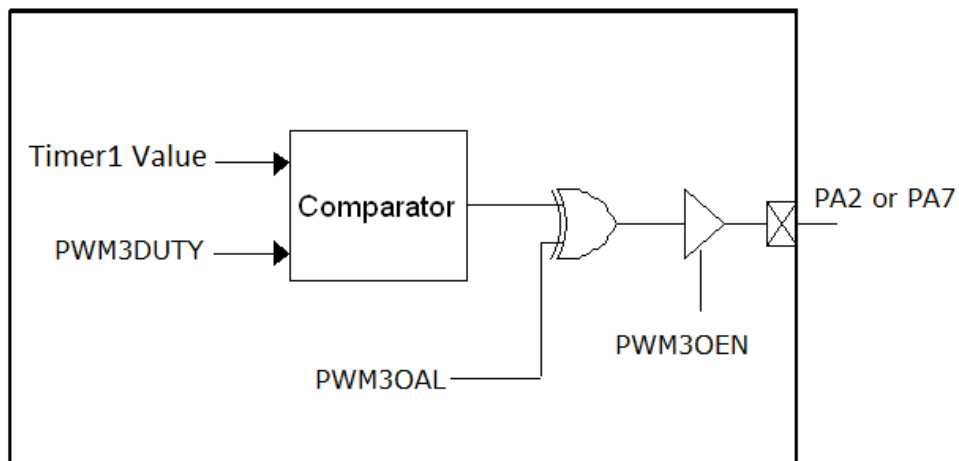


图 19 PWM2 结构框图

3.10 PWM3

PWM3 可由PA2 或PA7 引脚输出。当寄存器PWM3OEN (P3CR1[7]) 设定为 1, PWM3 会自动成为输出引脚。PWM3 输出的有效状态是由寄存器PWM3OAL (P3CR1[6]) 决定。当PWM3OAL为 1, PWM3 为低电平有效输出; PWM3OAL为 0, PWM3 为高电平有效输出。

此外, PWM3 的占空比与帧率皆可是编程的。占空比是由寄存器TM1RH[1:0]和PWM3DUTY[7:0]决定。当 PWM3DUTY为 0, PWM3 无法输出占空比。当PWM3DUTY为 0x3FF, PWM3 将输出 1023/1024 的占空比(当 PWM3OAL为 0)。帧率是由TMRH[5:4] +TMR1[7:0]初始值所决定。因此, PWM3DUTY数值必须小于或等于 TMR1[9:0]。当用户在写PWM3DUTY时,先写入PWM3DUTY[9:8]高 2 位(TM1RH[1:0])再写入PWM3DUTY[7:0],

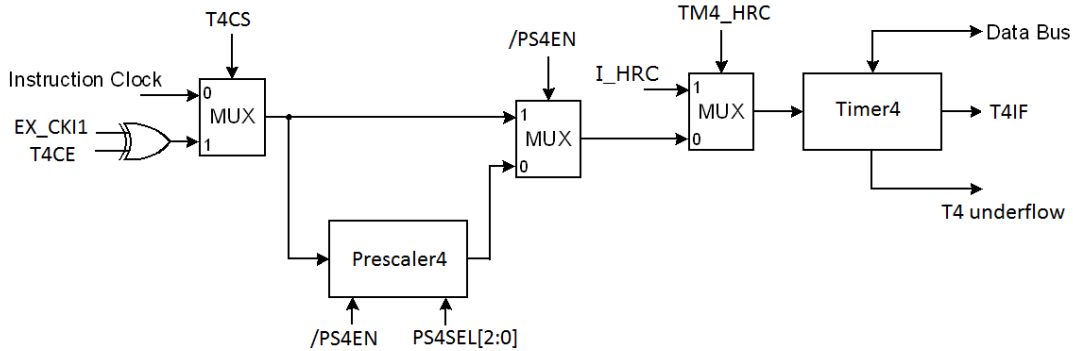


等到定时器 1 下溢后,就可更新PWM3DUTY重载缓存寄存器。PWM3 的结构框图如下:

图 20 PWM3 结构框图

3.11 Timer4 / PWM4

定时器 4 是具有预分频器 4 的 10 位下数定时器，其预分频比是可编程的。定时器 4 的输出可以被用于产生 PWM4 输出。定时器 4 创建自动重载功能和定时器 4 重载寄存器储存双重命令。当用户写定时器重载寄存器时，先写入定时器 4 高 2 位（TM4RH[7:6]）再写入 TMR4，如果 T4EN=1，定时器 4 重载寄存器会等到定时器 4 下溢后，再更新到定时器 4 的计数器中。如果 T4EN=0，定时器 4 重载寄存器将会在 TMR4 写入立即值后更新到定时器 4 的计数器中。读取定时器 4 会显示定时器 4 目前计数数值的内容。



定时器 4 的结构框图如下图所示：

图 21 定时器 4 结构框图

定时器 4 的操作可以由寄存器 T4EN (T4CR1[0]) 开启或关闭。开启定时器 4 后，寄存器 T4CS (T4CR2[5]) 和 TM4_HRC (T4CR1[3]) 可决定时钟源是指令时钟 F_{INST} 、I_HRC 或外部时钟引脚 EX_CK11。当 T4CS 为 1 和 TM4_HRC 为 0，EX_CK11 引脚当作时钟源。当 T4CS 为 0 和 TM4_HRC 为 0，指令时钟会被选择当作时钟源。当 T4CS 为 0 和 TM4_HRC 为 1，I_HRC 会被选择当作时钟源。当 EX_CK11 引脚被选取，寄存器控制位 T4CE (T4CR2[4]) 可决定 EX_CK11 引脚的时钟触发沿。当 T4CE 是 1，EX_CK11 引脚的上升沿将让定时器 4 计数减一。当 T4CE 是 0，EX_CK11 引脚的下降沿将让定时器 4 计数减一。定时器 4 时钟源可以由预分频器 4 所分频。寄存器 /PS4EN (T4CR2[3]) 为 0，可开启预分频器 4。寄存器 PS4SEL[2:0] (T4CR2[2:0]) 可以决定其预分频比从 1:2 到 1:256。预分频器 4 的目前数值可以由读取寄存器 PS4CV 取得。

定时器 4 提供两种计数模式：单次计数与连续计数。当寄存器 T4OS (T4CR1[2]) 为 1，即为单次计数模式。定时器 4 从储存在寄存器 TMR4[9:0] 的初始值下数到 0x00，当下溢发生时，定时器 4 停止计数。当寄存器 T4OS (T4CR1[2]) 为 0，即为连续计数模式。当下溢发生，寄存器 T4RL (T4CR1[1]) 会决定计数的初始值。当 T4RL 为 1，定时器 4 从寄存器 TMR4[9:0] 重新载入数值作为初始值并继续下数。当 T4RL 为 0，定时器 4 以 0x3FF 作为初始值并继续下数。

当定时器 4 下溢，寄存器 T4IF (INTE2[6]) 会被设定为 1，标明定时器 4 发生下溢中断。如果寄存器 T4IE (INTE2[2]) 与 GIE 同时设定为 1，会发生中断请求且执行中断服务程序。直到程序写入 0 到 T4IF，T4IF 才会被清除。

定时器 4 时序图如下图所示：

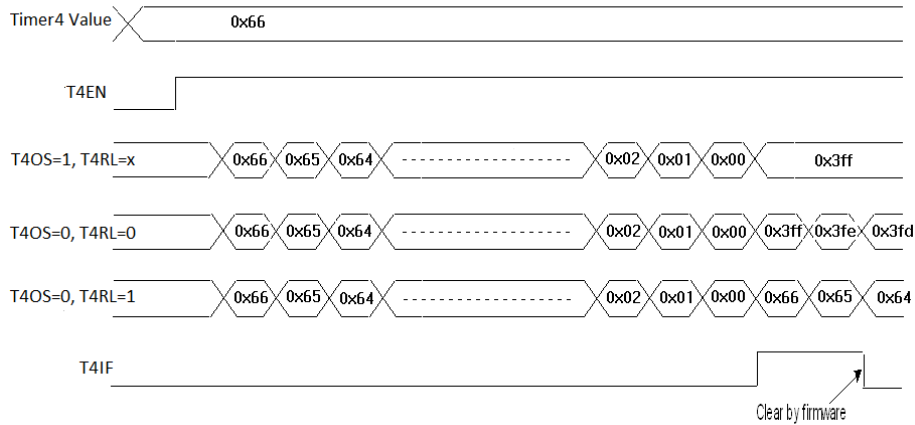


图 22 定时器 4 时序图

PWM4 可由PB3 引脚输出。当寄存器PWM4OEN (T4CR1[7]) 设定为 1, PWM4 会自动成为输出引脚。PWM4 输出的有效状态是由寄存器PWM4OAL (T4CR1[6]) 决定。当PWM4OAL为 1, PWM4 为低电平有效输出; PWM4OAL为 0, PWM4 为高电平有效输出。

此外, PWM4 的占空比与帧率皆是可编程的。占空比是由寄存器PWM4DUTY[7:0]决定。当PWM4DUTY为 0, PWM4 无法输出占空比。当PWM4DUTY为 0x3FF, PWM4 将输出 1023/1024 的占空比 (当PWM4OAL为 0)。帧率是由 TM4RH[7:6] +TMR4[7:0] 初始值所决定。因此, PWM4DUTY 数值必须小于或等于 TM4RH[7:6] +TMR4[7:0]。当用户在写PWM4DUTY时, 先写入PWM4DUTY[9:8]高 2 位再写入PWM4DUTY[7:0], 等到定时器 4 下溢后, 就可更新PWM4DUTY重载缓存寄存器。PWM4 的结构框图如下:

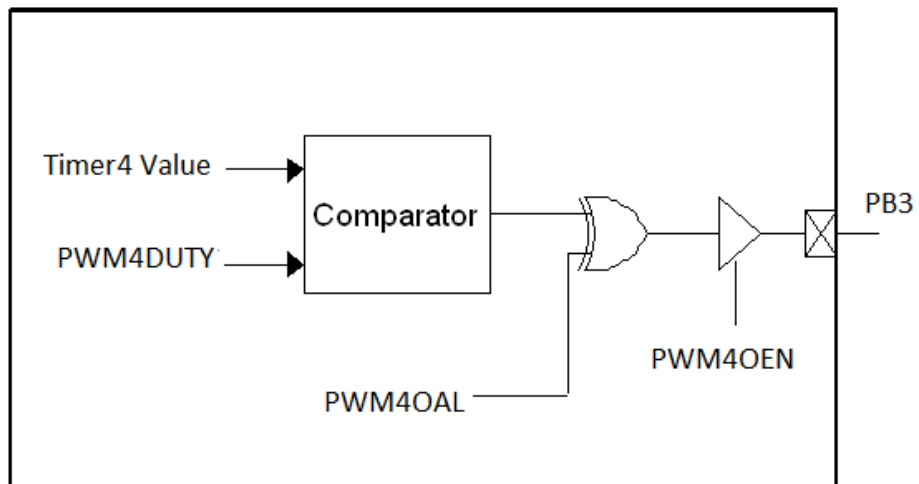


图 23 PWM4 结构框图

3.12 Timer5 / PWM5

定时器 5 是具有预分频器 5 的 10 位下数定时器，其预分频比是可编程的。定时器 5 的输出可以被用于产生 PWM5 输出。定时器 5 创建自动重载功能和定时器 5 重载寄存器存储双重命令。当用户写定时器重载寄存器时，先写入定时器 5 高 2 位（TM5RH[5:4]）再写入 TMR5，如果 T5EN=1，定时器 5 重载寄存器会等到定时器 5 下溢后，再更新到定时器 5 的计数器中。如果 T5EN=0，定时器 5 重载寄存器将会在 TMR5 写入立即值后更新到定时器 5 的计数器中。读取定时器 5 会显示定时器 5 目前计数数值的内容。

定时器 5 的结构框图如下图所示：

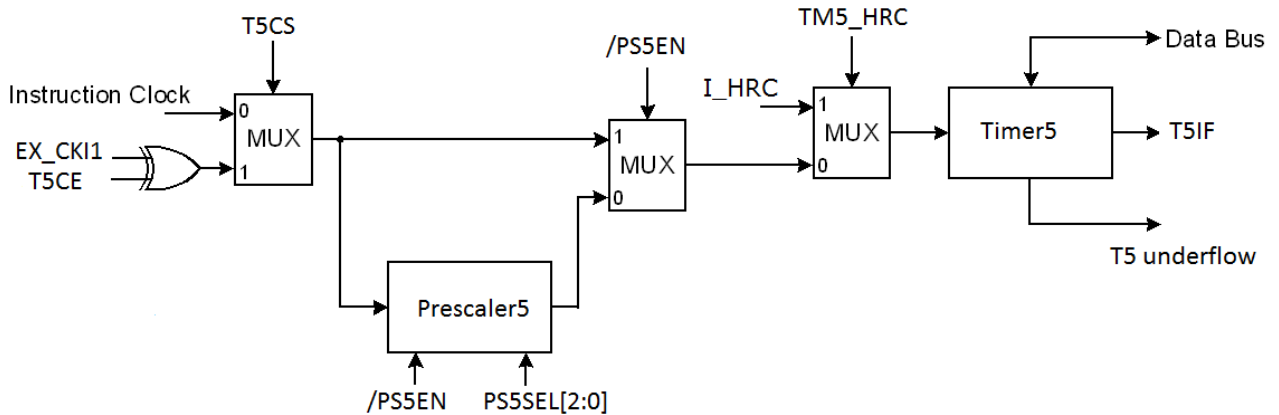


图 24 定时器 5 结构框图

定时器 5 的操作可以由寄存器 T5EN (T5CR1[0]) 开启或关闭。开启定时器 5 后，寄存器 T5CS (T5CR2[5]) 和 TM5_HRC (T5CR1[3]) 可决定时钟源是指令时钟 F_{INST}、I_HRC 或外部时钟引脚 EX_CK11。当 T5CS 为 1 和 TM5_HRC 为 0，EX_CK11 引脚当作时钟源。当 T5CS 为 0 和 TM5_HRC 为 0，指令时钟会被选择当作时钟源。当 T5CS 为 0 和 TM5_HRC 为 1，I_HRC 会被选择当作时钟源。当 EX_CK11 引脚被选取，寄存器控制位 T5CE (T5CR2[4]) 可决定 EX_CK11 引脚的时钟触发沿。当 T5CE 是 1，EX_CK11 引脚的上升沿将让定时器 5 计数减一。当 T5CE 是 0，EX_CK11 引脚的下降沿将让定时器 5 计数减一。定时器 5 时钟源可以由预分频器 5 所分频。寄存器 /PS5EN (T5CR2[3]) 为 0，可开启预分频器 5。寄存器 PS5SEL[2:0] (T5CR2[2:0]) 可以决定其预分频比从 1:2 到 1:256。预分频器 5 的目前数值可以由读取寄存器 PS5CV 取得。

定时器 5 提供两种计数模式：单次计数与连续计数。当寄存器 T5OS (T5CR1[2]) 为 1，即为单次计数模式。定时器 5 从储存在寄存器 TMR5[9:0] 的初始值下数到 0x00，当下溢发生时，定时器 5 停止计数。当寄存器 T5OS (T5CR1[2]) 为 0，即为连续计数模式。当下溢发生，寄存器 T5RL (T5CR1[1]) 会决定计数的初始值。当 T5RL 为 1，定时器 5 从寄存器 TMR5[9:0] 重新载入数值作为初始值并继续下数。当 T5RL 为 0，定时器 5 以 0x3FF 作为初始值并继续下数。

当定时器 5 下溢，寄存器 T5IF (INTF3[5]) 会被设定为 1，标明定时器 5 发生下溢中断。如果寄存器 T5IE (INTE3[5]) 与 GIE 同时设定为 1，会发生中断请求且执行中断服务程序。直到程序写入 0 到 T5IF，T5IF 才会被清除。

定时器 5 时序图如下图所示：

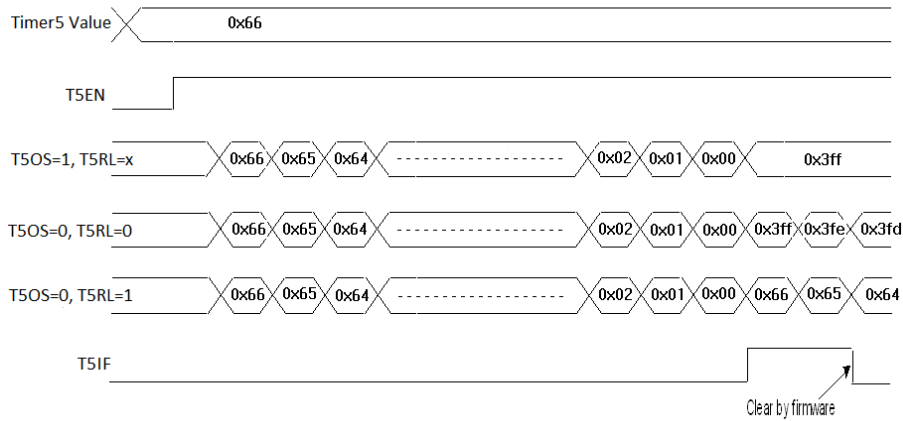


图 25 定时器 5 时序图

PWM5 可由PB2 引脚输出。当寄存器PWM5OEN (T5CR1[7]) 设定为 1, PWM5 会自动成为输出引脚。PWM5 输出的有效状态是由寄存器PWM5OAL (T5CR1[6]) 决定。当PWM5OAL为 1, PWM5 为低电平有效输出; PWM5OAL为 0, PWM5 为高电平有效输出。

此外, PWM5 的占空比与帧率皆是可编程的。占空比是由寄存器TM5RH[1:0]和PWM5DUTY[7:0]决定。当 PWM5DUTY为 0, PWM5 无法输出占空比。当PWM5DUTY为 0x3FF, PWM5 将输出 1023/1024 的占空比(当 PWM5OAL为 0)。帧率是由TM5RH[5:4] +TMR5[7:0]初始值所决定。因此, PWM5DUTY数值必须小于或等于 TM5RH[5:4] +TMR5[7:0]。当用户在写PWM5DUTY时, 先写入PWM5DUTY[9:8]高 2 位(TM5RH[1:0])再写入 PWM5DUTY[7:0], 等到定时器 5 下溢后, 就可更新PWM5DUTY重载缓存寄存器。PWM5 的结构框图如下:

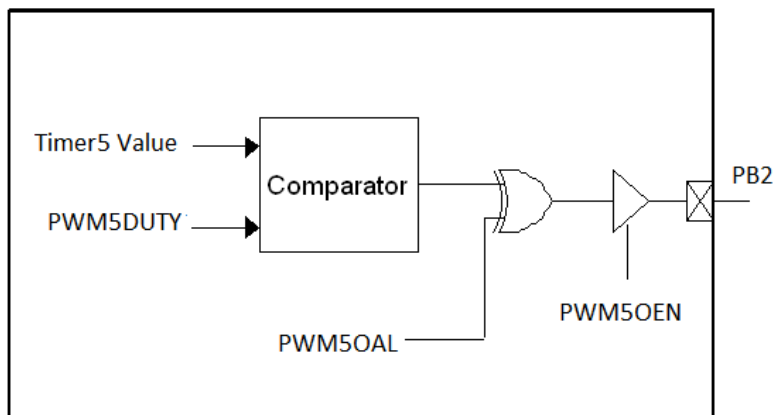


图 26 PWM5 结构框图

3.13 CCP模式

CCP（捕捉/比较/脉宽调变）寄存器（CCPR）由 2 个 8-bit寄存器组成。CCPRL（低字节）和CCPRH（高字节）。CCPCON和PWMDB寄存器控制CCP操作。捕捉和比较模式使用 16 位定时器，PWM模式使用 10 位定时器。CCP定时器和寄存器利用现有的定时器和寄存器。以下表格显示的是与CCP共享的定时器和寄存器。

注意：当NY8BE62D在CCP模式时，相关的定时器/PWM功能关闭。

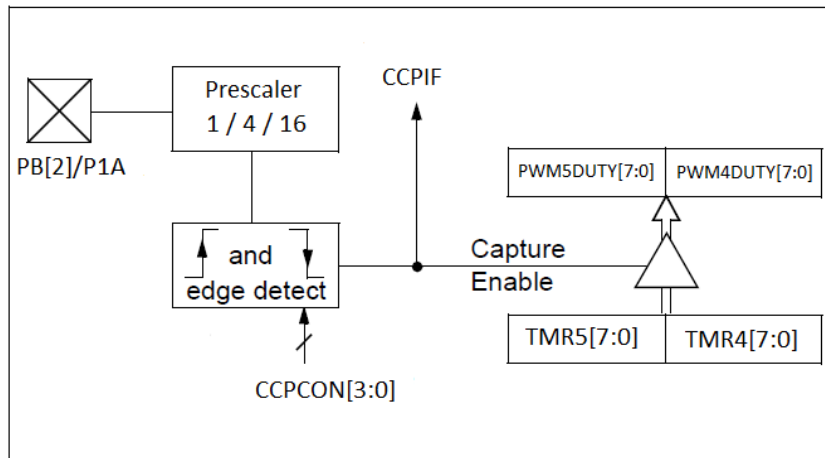
CCP 模式	CCP 寄存器/定时器	共享定时器	共享寄存器
捕捉/比较	CCPL	-	PWM4DUTY[7:0]
捕捉/比较	CCPH	-	PWM5DUTY[7:0]
PWM	PWMDUTY	-	PWM5DUTY[9:0]
PWM	PWMDDB	-	-
捕捉/比较	捕捉/比较定时器	Timer5[7:0] (MSB) + Timer4[7:0] (LSB)	-
PWM	PWM定时器	Timer5[9:0]	-

3.13.1 捕捉模式

在捕捉模式下，当P1A（PB2）发生事件时，CCPRH:CCPRL (PWM5DUTY[7:0]:PWM4DUTY[7:0])捕捉 16 位定时寄存器事件定义：

- 每一次的下降沿
- 每一次的上升沿
- 每 4 次的上升沿
- 每 16 次的上升沿

捕捉模式的结构框图如下：



一次事件由CCPM[3:0]控制。当捕捉一次时，设置中断请求标志位CCPIF。

在捕捉模式下，P1A(PB2)应该被配置为输入脚，如果它是输出脚，要写入IO口来创造捕捉条件。

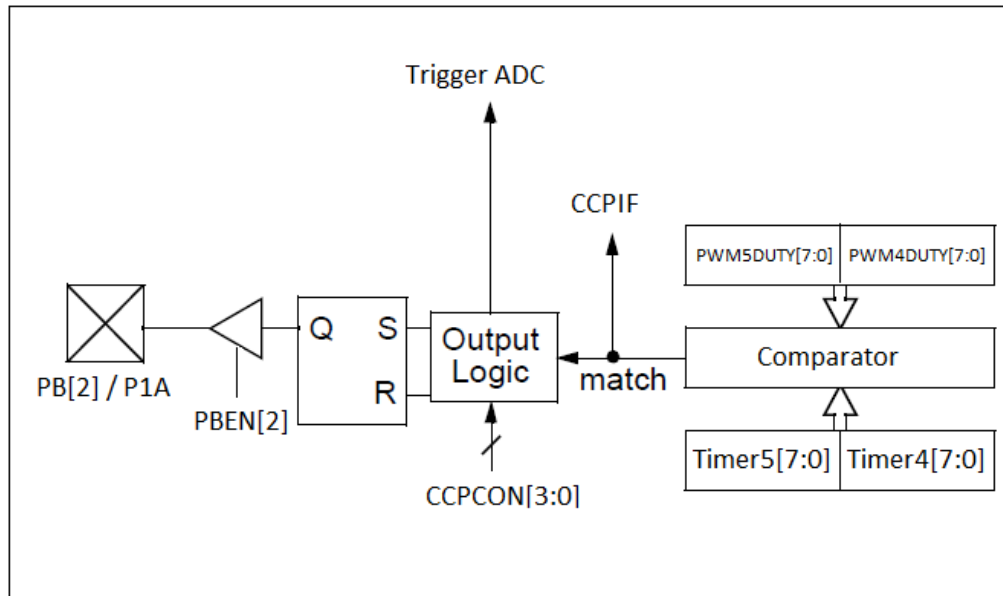
在捕捉模式下，捕捉定时器必须与CPU时钟同步，否则捕捉操作不能进行。

3.13.2 比较模式

在比较模式下，16 位（PWM5DUTY[7:0]:PWM4DUTY[7:0]）寄存器的值不断与比较寄存器的值作比较。当匹配时，CCP1（PB2）引脚为：

- 高驱动
- 低驱动
- 切换输出
- 保持不变（仅中断）
- 如果ADC开启则开触发ADC

比较模式的结构框图如下：



引脚状态取决于控制位CCPM[3:0]的值。当比较匹配发生时，置起中断标志位CCPIF。

在比较模式下，用户必须把CCP(PB2)配置为输出口。

在比较模式下，比较定时器必须与CPU时钟同步，否则比较操作不能进行。

3.13.3 CCP PWM Mode

在CCP PWM模式，CCP模式块上升为一个 10 位分辨率的PWM输出。PWM引脚为P1A(PB2)，P1B(PA5)，P1C(PA2)，P1D(PA4)。PWM的周期和占空比由寄存器Timer5[9:0]和PWM5DUTY[9:0]指定。

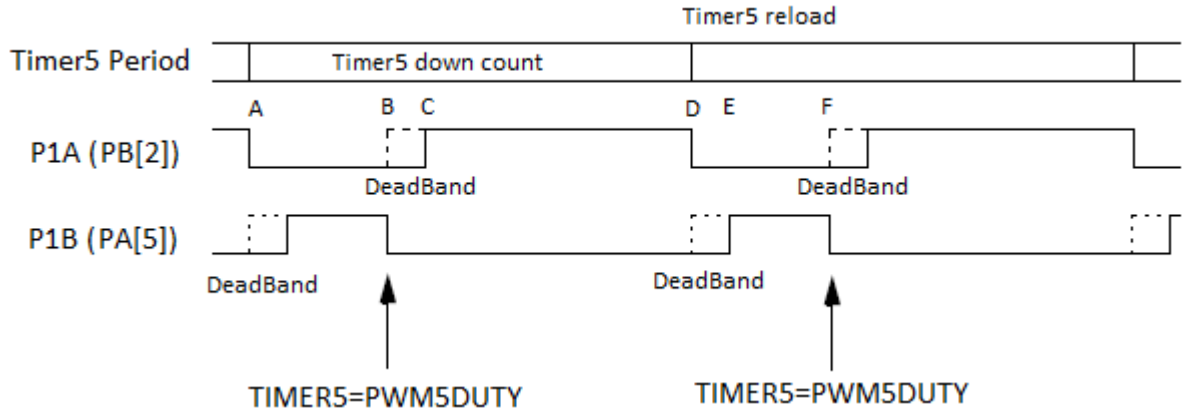
寄存器PWM5M[1:0]在CCPCON寄存器中允许以下其中一种配置：

- 单个输出：P1A输出，P1B，P1C，P1D由一般I/O指定。
- 半桥输出：P1A，P1B由死区控制调节，P1C，P1D由一般I/O指定。
- 全桥输出，向前模式：P1D 被调节，P1A有效，P1B，P1C无效。

- 全桥输出，反向模式：P1B 被调节，P1C有效，P1A，P1D无效。

在单个输出模式下，P1A (PB2)被当作PWM输出。PB2 必须设置为输出口。

在半桥模式下，P1A (PB2)有PWM输出信号，P1B (PA5)有互补的PWM输出信号。在这个模式下，PB2 和PA5 必须设置为输出口。

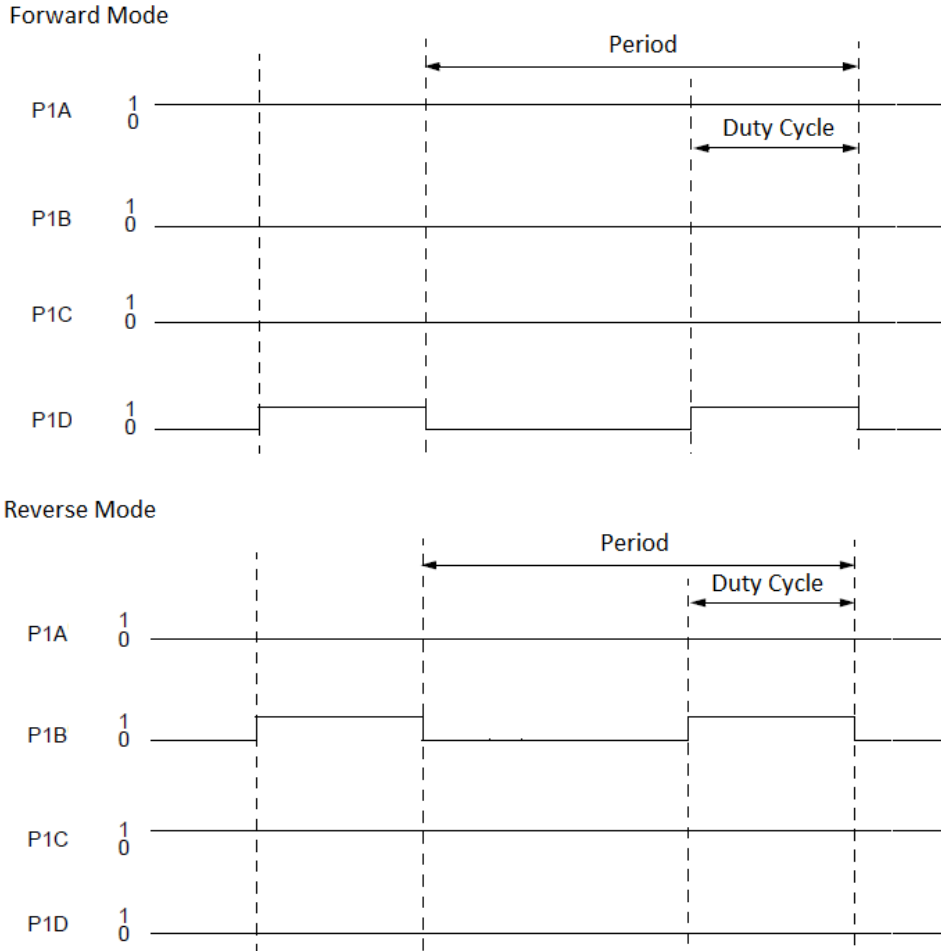


在半桥输出模式下，一个数字化可编程死区延时可避免桥电开关毁坏时被击穿。PWMDDB[7:0]是半桥模式的死区延时。

从上面的时序图可以看出，计时器 5 是一个 10 位的下数计时器，P1A是PWM5 输出，P1B是PWM5 互补输出。在B点处，计时器 5 的值等于PWM5DUTY，在D点处，计时器 5 计数为 0 并重新回载值。如果没有死区控制，P1A输出会从B点一直工作到D点。P1B输出会从D点一直工作到F点。一个非零的死区将会延迟P1A的上升沿从B点到C点，延迟P1B的上升沿从D点到E点。死区区域时间是从B到C和从D到E。

在全桥模式下，4 个引脚都用作输出口，但是只有 2 个输出口同时工作。在向前模式下，P1A(PB2)是连续工作的，P1D(PA4)是PWM调制输出。在反向模式下，P1C(PA2)是连续工作的，P1B(PA5)是PWM调制输出。在这个模式下，PB2, PA5, PA2 和 PA4 必须设置为输出口。

下列时序图显示的是全桥向前和反向时序。



3.14 电阻/频率转换器模式 (RFC)

NY8BE62D内置RFC功能，当开启RFC功能 (RFCEN=1)，选择的RFC输入引脚的状态将会控制定时器 1 的计数行为。当选择的引脚状态为 0 (RFC输入引脚电压低于 V_{IL})，定时器 1 将会持续计数，当选择的引脚状态为 1 (RFC输入引脚电压高于 V_{IH})，定时器 1 停止计数。如下图显示RFC模式如何工作：PSEL3~0 用来从 14 个NY8BE62D的 14 个引脚中选择 1 个RFC输入引脚。RFCEN是用来在一般使能信号T1EN和选择的RFC输入引脚之前切换定时器 1 的使能信号。

RFC的一个应用是用来测量RC充电时间，如下图所示，当PSEL3~0=0x01，PA1 为RFC输入引脚。首先设置PA1 输出 0 (低于 V_{IL})，接着，清除计时器的内容，将PA1 设置为输入引脚并开启RFC模式，然后，定时器 1 会开始下数，这时RC电路开始对PA1 引脚充电。当PA1 引脚电压高于 V_{IH} 时，定时器 1 会停止计数。定时器 1 将会记录RC电路充电时间。(注意：计时器 1 是下数。)

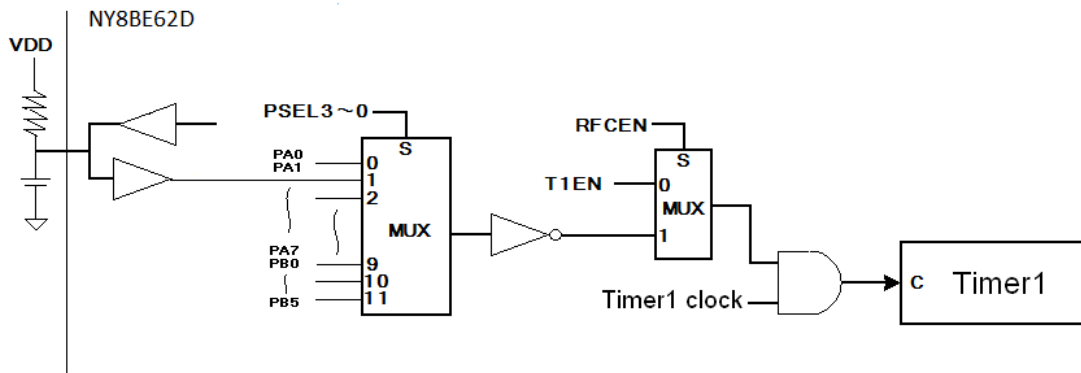


图 27 RFC结构框图

3.15 IR Carrier

根据IR口的配置，NY8BE62D的IR输出口可以是PB1 或PA3。当寄存器IREN（IRCR[0]）被设定为 1 后，IR载波将会产生，并且PB1 / PA3 会自动成为输出引脚。当IREN清零，PB1 / PA3 将会成为一般I/O引脚。

红外线载波频率是由寄存器IRF57K（IRCR[1]）所选择。当IRF57K为 1，红外线载波频率是 57KHz。当IRF57K为 0，频率是 38KHz。由于红外线载波输出是以高速振荡时钟除频得来，当有使用外部晶振时，有必要指定哪个频率用作系统振荡频率。寄存器IROSC358M（IRCR[7]）被用来提供这个信息。当IROSC358M 为 1 时，外部晶振频率是 3.58MHz，当IROSC358M 为 0 时，外部晶振频率是 455KHz。当选择L_HRC当作高速振荡时钟来源时，IROSC358M（IRCR[7]）将会被忽略。

红外线载波的极性是可选择的。当寄存器IRCSEL（IRCR[2]）为 1 且PB1 / PA3 输出数据为 0，红外线载波将由PB1 / PA3 引脚输出。当寄存器IRCSEL（IRCR[2]）为 0 且PB1 / PA3 输出数据为 1，红外线载波将由PB1 / PA3 引脚输出。红外线载波的极性如下图所示：

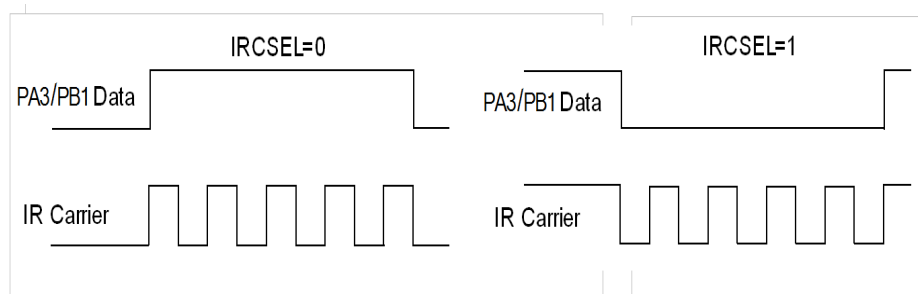


图 28 红外线载波的极性 vs. 输出口数据

3.16 低电压侦测（LVD）

NY8BE62D集成了低压侦测电路和比较器功能。在LVD模式下，NY8BE62D设置PS[1:0]（寄存器CMPPCR[3:2]）= 10 和 NS[1:0]（寄存器CMPPCR[1:0]）= 10。NY8BE62D内置准确的低电压侦测电路来侦测VDD电压水平。如果LVDEN

(寄存器PCON[5]) 设为 1, 当VDD电压低于下表LVDS[3:0]选择的电压值时, 读取LVDOOUT会得到 0。如果开启LVD中断使能位且GIE=1 时, LVD中断标志位将会被设置为 1, 程序将跳入中断子程序。LVD的实时输出可以通过寄存器PCON1[6]查询。以下是LVD结构框图:

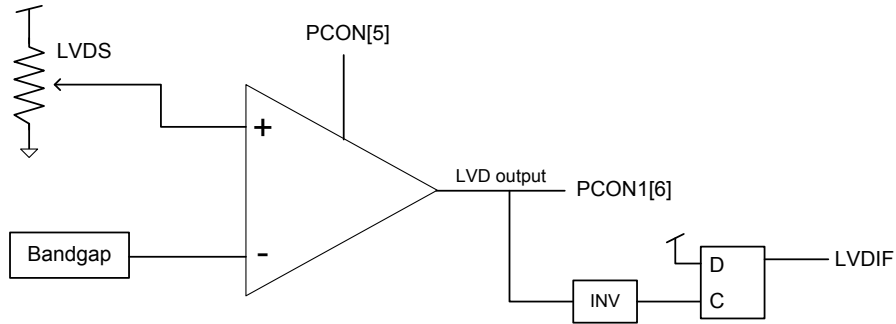


图 29 LVD结构框图

下表为LVD电压选择表格:

LVDS[3:0]	Voltage
0000	1.9V
0001	2.0V
0010	2.2V
0011	2.4V
0100	2.6V
0101	2.8V
0110	2.9V
0111	3.0V
1000	3.15V
1001	3.30V
1010	3.45V
1011	3.60V
1100	3.75V
1101	3.90V
1110	4.05V
1111	4.15V

表 19 LVD 电压选择

注意: LVD 的迟滞电压 (从低到高) 大约为 0.1V, 在电压充电应用中 (检测电压是从低到高), LVD 选择电压的表格如下:

LVDS[3:0]	Voltage
0000	(1.9+0.1) V
0001	(2.0+0.1) V
0010	(2.2+0.1) V
0011	(2.4+0.1) V

LVDS[3:0]	Voltage
0100	(2.6+0.1) V
0101	(2.8+0.1) V
0110	(2.9+0.1) V
0111	(3.0+0.1) V
1000	(3.15+0.1) V
1001	(3.30+0.1) V
1010	(3.45+0.1) V
1011	(3.60+0.1) V
1100	(3.75+0.1) V
1101	(3.90+0.1) V
1110	(4.05+0.1) V
1111	(4.15+0.1) V

LVD 的控制步骤如下：

步骤 1：根据 LVDS[3:0] 选择 LVD 电压

步骤 2：设置 CMPCR = 0x0A

步骤 3：设置 PCON[5]=1（开启 LVD）

步骤 4：用 PCON1[6] 检查 LVD 状态

注意：如果 LVD 电压 LVDS[3:0] 发生改变，用户必须等待至少 50us (@F_{Hosc}=1MHz) 再通过 PCON1[6] 得到正确的 LVD 状态。

3.17 电压比较器

NY8BE62D 提供各种模拟比较模式的电压比较器和内部参考电压。比较器的正输入源与负输入源和 GPIO 口复用。

CMPEN（寄存器 PCON[2]）用来开启或关闭比较器，当 CMPEN=0（默认）时，比较器关闭，当 CMPEN=1 时，比较器开启。在睡眠模式（Halt mode）中比较器将自动关闭。

比较器的结构框图如下图所示：

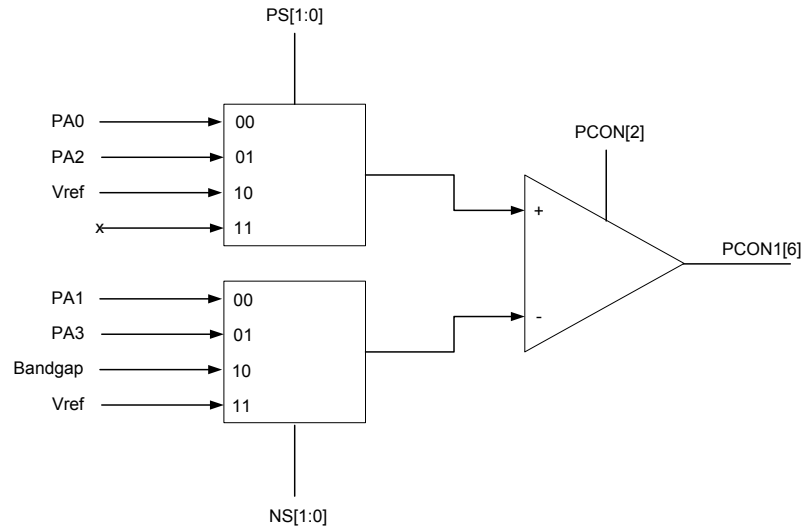
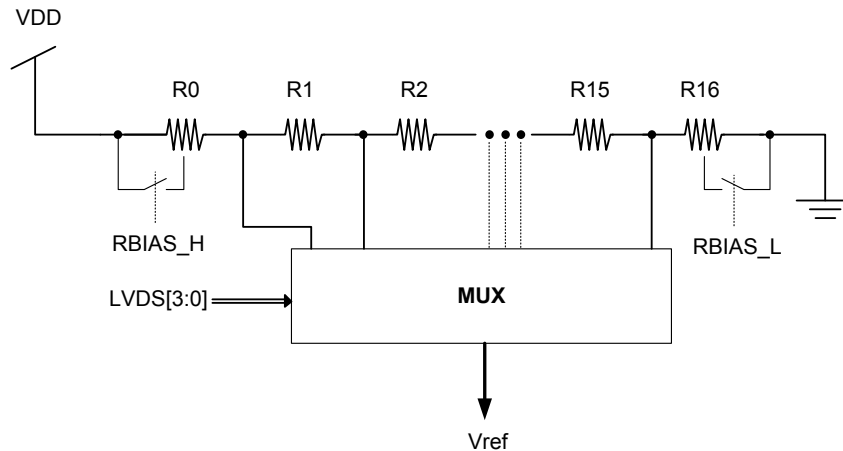


图 30 比较器的结构框图

3.17.1 比较器参考电压 (Vref)

内部参考电压是由各种电阻建立的，来提供不同的参考电压值。RBIAS_H和RBIAS_L是用来选择最大的和最小的Vref值，LVDS[3:0]用来选择 16 种电压值中的其中一种。



Vref由RBIAS_H, RBIAS_L 和 LVDS[3:0]决定。LVDS[3:0]用来选择一种参考电压，表格如下：

LVDS[3:0]	RBIAS_H=1 RBIAS_L=0	RBIAS_H=0 RBIAS_L=1
0000	67/128 V _{DD}	34/128 V _{DD}
0001	64/128 V _{DD}	32/128 V _{DD}
0010	59/128 V _{DD}	28/128 V _{DD}
0011	54/128 V _{DD}	25/128 V _{DD}
0100	50/128 V _{DD}	22/128 V _{DD}
0101	47/128 V _{DD}	20/128 V _{DD}

0110	45/128 V _{DD}	19/128 V _{DD}
0111	44/128 V _{DD}	17/128 V _{DD}
1000	42/128 V _{DD}	16/128 V _{DD}
1001	40/128 V _{DD}	15/128 V _{DD}
1010	38/128 V _{DD}	14/128 V _{DD}
1011	37/128 V _{DD}	13/128 V _{DD}
1100	35/128 V _{DD}	12/128 V _{DD}
1101	34/128 V _{DD}	11/128 V _{DD}
1110	33/128 V _{DD}	10/128 V _{DD}
1111	32/128 V _{DD}	10/128 V _{DD}

表 20 参考电压Vref选项表

注意 1: Vref的误差为±0.1V。

注意 2: 在睡眠模式 (Halt mode) 和待机模式 (Standby mode) 下, RBIAS_H 和 RBIAS_L 必须设置为 0 来避免电源损耗。

比较器的正输入源由PS[1:0] (寄存器 CMPPCR[3:2])决定。

表格如下:

PS[1:0]	正输入源
00	PA0
01	PA2
10	Vref
11	---

表 21 正输入源选择

比较器的负输入源由NS[1:0] (寄存器 CMPPCR[1:0])决定, 表格如下:

NS[1:0]	负输入源
00	PA1
01	PA3
10	Bandgap (0.6V)
11	Vref

表 22 负输入源选择

有两种方式可以取得比较器的输出结果: 一是通过寄存器轮询方式, 另一个是通过探查输出脚。

比较器输出可以由 LVDOUT (寄存器 PCON1[6])轮询。

在输出口探查比较器输出，设置 CMPOE (寄存器 OSCCR[6]) 为 1，然后 PB1 的实时状态就是比较器的输出结果。需要注意的是，当 CMPOE=1 时，PWM3 功能将会关闭。

3.18 ADC模数转换器

NY8BE62D提供 11+2 通道 12 位ADC模数转换器，可将模拟信号转换为 12 位数字值。ADC参考电压可选外部引脚PA0 输入或由内部VDD, 4V, 3V, 2V提供。模拟输入可从外部模拟输入通道PA0~PA4 引脚与PB0~PB5 引脚选择，也可选择内部 1/4VDD或GND为模拟输入通道。ADC时钟 (ADCLK) 能够选择 $F_{INST}/1$, $F_{INST}/2$, $F_{INST}/8$ 或 $F_{INST}/16$ 四种。ADC采样时间可选择为 1 个ADCLK, 2 个ADCLK, 4 个ADCLK或 8 个ADCLK四种。在ADC工作前，先设置ADEN=1，再将START设置为 1，ADC开始模数转换。读取寄存器EOC=0 表示ADC还在转换中，EOC=1 表示ADC已完成一次模数转换。如果寄存器ADIE=1 且GIE设置为 1，在EOC自动从 0→1 后，ADC中断标志ADIF位将被硬件设为 1 并处理此中断请求。结构框图如下：

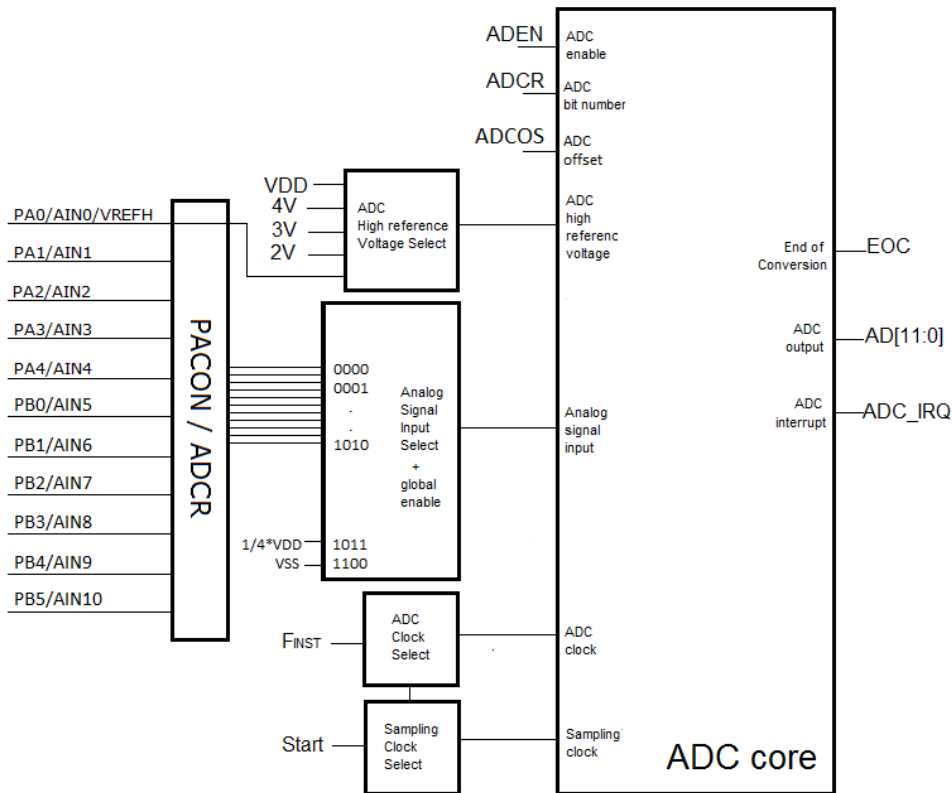


图 31 ADC结构框图

3.18.1 ADC 参考电压

ADC内建 5 种高参考电压源，可由寄存器ADVREFH来设置。这些高参考电压源分别是一个外部电压 (PA0) 及四个内部电压 (VDD, 4V, 3V, 2V)。当EVHENB=1，ADC参考电压是外部电压源，由引脚 PA0 提供。此模式下，PA0 引脚输入的参考电压必须在VDD~2V之间。当EVHENB=0，ADC参考电压由VHS[1:0]选择的内部电压源。如果VHS[1:0]=11，ADC参考电压为VDD。如果VHS[1:0]=10，ADC参考电压为内部 4V。如果VHS[1:0]=01，ADC参考电压为内部 3V。如果VHS[1:0]=00，ADC参考电压为内部 2V。引脚VDD电压值不得低于选择的ADC内部参考电压 (4V / 3V / 2V)，或电压值等于VDD。ADC采样电压范围受限与高/低参考电压。ADC低参

考电压是VSS且不能改变，高参考电压包括内部VDD/4V/3V/2V和PA0 提供的外部参考电压源。ADC参考电压范围限制（ADC高参考电压-低参考电压） $\geq 2V$ ，低参考电压是VSS=0V，所以ADC高参考电压范围是 $2V \sim VDD$ 。

ADC的模拟电压必须在ADC的最低电压和最高电压之间，如果超出这个范围，ADC的转换结果将有误（超出或为0）。

EVHENB	VHS[1:0]	参考电压
1	x x	PA0
0	1 1	VDD
0	1 0	4V
0	0 1	3V
0	0 0	2V

表 23 ADC参考电压选择

3.18.2 ADC 模拟输入通道

ADC依据CHS[3:0]与GCHS来选择模拟输入通道。GCHS为所有模拟输入通道的总开关，任何模拟输入通道在转换前必须将GCHS设置为1。

GCHS	CHS[3:0]	ADC 模拟输入通道
0	xxxx	x
1	0000	PA0
1	0001	PA1
1	0010	PA2
1	0011	PA3
1	0100	PA4
1	0101	PB0
1	0110	PB1
1	0111	PB2
1	1000	PB3
1	1001	PB4
1	1010	PB5
1	1011	$1 / 4 * VDD$
1	1100	VSS
1	1101	N.C.
1	111x	N.C.

表 24 ADC模拟输入通道选择

ADC模拟输入通道与数字I/O引脚共享。将模拟信号连接到这些IO口上可能会导致IO口漏大电流。在低电压情况下，漏电会是一个很大的问题。将寄存器PACONx / PBCONx位来配置PAx / PBx寄存器位可以解决以上问题。写“0”到寄存器PACONx / PBCONx来配置相关的PAx / PBx来作为纯粹的模拟输入引脚可以避免漏电问题，它不能作为一般IO口。

除了设置PACONx / PBCONx寄存器位之外，选择的模拟输入脚必须设置为输入模式，且内部上拉/下拉必须关闭，否则模拟输入值将会受到影响。

3.18.3 ADC 时钟（ADCLK），采样时钟（SHCLK）与位数选择

转换速度和转换准确度受ADC时钟(ADCLK)，采样时钟(SHCLK)和转换位数的影响。ADCLK是ADC的基础。

ADC转换时间是根据ADC时钟（ADCLK），采样时间（1/SHCLK）与ADC位数所计算出来的。较长的采样时间（1/SHCLK）能采样到更精确的模拟输入信号，但会增加ADC转换时间。寄存器ADCR[1:0]为选择ADC位数。选择较少的ADC位数能减少ADC转换时间，选择较多的ADC位数能增加模数转换的精准度。

寄存器ADCK[1:0]选择ADC时钟频率。

ADCK[1:0]	ADC 时钟频率
0 0	$F_{INST}/16$
0 1	$F_{INST}/8$
1 0	$F_{INST}/1$
1 1	$F_{INST}/2$

表 25 ADC 时钟选择

寄存器SHCK[1:0]选择ADC采样时间

SHCK[1:0]	ADC采样时间
0 0	1 ADCLK
0 1	2 ADCLK
1 0	4 ADCLK
1 1	8 ADCLK

表 26 ADC采样时间选择

寄存器ADCR[1:0]选择ADC转换位数。

ADCR[1:0]	ADC转换位数
0 0	8 位
0 1	10 位
1 x	12 位

表 27 ADC转换位数选择

ADC转换时间从START（开始ADC转换）写 1 开始一直到EOC从 0 → 1（结束ADC转换）为止。持续时长取决于ADC分辨率，ADC时钟和采样时间。

ADC转换时间 \approx ADC采样时间 + (ADC位数 + 2) * ADC时钟周期。

下表为ADC在不同条件下的转换时间与转换周期。

ADC位数	ADC时钟频率 (ADCLK)	ADC采样时间 SHCLK	ADC转换时间 (ADC时钟数)	F _{INST} =2MHz		F _{INST} =250K	
				转换时间	转换率	转换时间	转换率
12	F _{INST} /16	8 ADCLK	22	176us	5.68kHz	1408us	710Hz
12	F _{INST} /1	1 ADCLK	15	7.5us	133.3kHz	60us	16.7kHz
10	F _{INST} /1	1 ADCLK	13	6.5us	153.8kHz	52us	19.2kHz
8	F _{INST} /1	1 ADCLK	11	5.5us	181.8kHz	44us	22.7kHz

表 28 ADC转换时间与转换率

3.18.4 ADC 偏移误差校准

ADC偏移误差是指第一次理论代码转换与第一次实际转码转换之间的误差，第一次理论转码转换在 0.5LSB时产生。ADC偏移误差随温度，进程和电压变化，NY8BE62D的此误差可以通过寄存器ADJMD进行实时调整。关于ADC偏移误差校准的过程，九齐提供NYIDE的范例代码“ADC_Interrupt_AutoK”。

3.18.5 ADC 操作过程

依序设定ADC时钟（ADCLK），ADC采样时间(SHCLK)，ADC位数(ADCR)，ADC高参考电压（ADVREFH），选择模拟输入通道和寄存器PACON相应位，再将ADEN位设置为 1。

在ADEN设置为 1 后必须等待至少 256us（ADC电路启动时间），再将START位写 1 来启动ADC模数转换。ADC转换尚未完成时，读取EOC位会得到 0。当ADC模数转换完成后会自动将EOC位设置为 1。

3.19 看门狗定时器 (WDT)

NY8BE62D中有独立振荡器被WDT所使用。由于该振荡器与其它振荡电路无关，故在待机模式和睡眠模式中WDT仍能继续工作。

WDT能被配置字节开启或关闭。当WDT被配置字节开启时，仍然可以通过WDTEN位（寄存器PCON[7]）来开启/关闭。此外，WDT上溢后可由另一个配置字节决定复位NY8BE62D或发出的中断请求。同时，在WDT上溢后，寄存器/TO（STATUS[4]）位将被清除为 0。

WDT上溢的时基可以是 3.5 毫秒、15 毫秒、60 毫秒或 250 毫秒，由两个配置字节决定。如果将预分频器 0 分配给WDT，则可以延长上溢周期。通过将 1 写入寄存器PS0WDT位，预分频器 0 将分配给WDT。预分频器 0 对WDT的分频比由寄存器PS0SEL[2:0]位决定，而且取决于WDT的上溢机制。如果WDT上溢将复位NY8BE62D，分频速率从 1:1 到 1:128。如果选为WDT中断时，则分频速率从 1:2 到 1:256。

当预分频器 0 分配给WDT时，执行CLRWDT指令将清除WDT、预分频器 0。并设置/TO标志位为 1。

如果用户选择WDT中断机制，在WDT上溢后，寄存器WDTIF（INTF[6]）位将设置为 1。如果寄存器WDTIE（INTE[6]）位和GIE位都设置为 1，则可能产生中断请求。直到程序将 0 写入WDTIF，WDTIF才会被清除为 0。

3.20 中断

NY8BE62D提供二种中断：一种是软件中断，另一种是硬件中断。软件中断由执行指令INT来产生。硬件中断则有以下 12 种：

- Timer0 上溢中断。
- Timer1 下溢中断。
- Timer4 下溢中断。
- Timer5/CCP下溢中断。
- WDT中断。
- PVPB 输入引脚状态改变中断。
- 外部中断 0 输入引脚。
- 外部中断 1 输入引脚。
- 外部中断 2 输入引脚。
- 低电压侦测/比较器输出翻转中断。
- ADC模数转换完成中断。
- EEPROM写入完成中断。

GIE是总中断屏蔽位，必须为 1 才能使能硬件中断功能。GIE可以通过ENI指令设置 1，通过DISI指令清除为 0。

执行完指令INT后，无论GIE是置 1 还是清除为零，下一条指令都将从地址 0x001 读取。同时，GIE将由NY8BE62D自动清除为零，这将防止嵌套中断的发生。软件中断的中断服务程序最后一条指令必须是RETIE。执行此指令将设置GIE为 1 并返回中断前程序执行序列。

当发生硬件中断时，相应的中断标志位将被设置为 1。该位在程序将 0 写入该位之前不会清除为零。因此，用户可以通过轮呼相应的中断标志位得知哪个硬件引发中断。需注意只有当相应的中断使能位设置为 1 时，才能正确地读取相应的中断标志。如果相应的中断使能位设置为 1，GIE也为 1，将发生硬件中断，下一条指令将从 0x008 执行。同时，NY8BE62D将自动清除寄存器GIE位为零。如果用户想要实现嵌套中断，可以使用ENI指令作为中断服务程序的第一条指令，将GIE设置为 1，并允许其他中断事件再次中断NY8BE62D。指令RETIE必须是中断服务程序的最后一条指令，它将GIE设置为 1 并返回中断前程序执行序列。

用户应注意ENI指令不能放在RETIE指令之前，因为中断服务程序中的ENI指令将开启嵌套中断，但RETIE指令会在跳出ISR时清除中断进程，所以可能会导致中断标志误清除。

3.20.1 Timer0 上溢中断

Timer0 上溢（从 0x00 到 0xFF），如果T0IE和GIE设置为 1，寄存器T0IF位将被硬件设为 1 并处理此中断请求。

3.20.2 Timer1 下溢中断

Timer1 下溢（从 0x3FF到 0x00），如果T1IE和GIE设置为 1，寄存器T1IF位将被硬件设为 1 并处理此中断请求。

3.20.3 Timer4 下溢中断

Timer4 下溢（从 0x3FF到 0x00），如果T4IE和GIE设置为 1，寄存器T4IF位将被硬件设为 1 并处理此中断请求。

3.20.4 Timer5 下溢中断/CCP 中断

Timer5 下溢 (从 0x3FF 到 0x00), 如果 T5IE 和 GIE 设置为 1, 寄存器 T5IF 位将被硬件设为 1 并处理此中断请求。当 CCP 的捕捉和比较模式开启时, 计时器 5 中断将会被 CCP 中断代替。

3.20.5 看门狗超时中断

当 WDT 上溢且配置字节选择 WDT 超时中断时, 寄存器 WDTIF 位将被硬件设为 1, 如果 WDTIE 和 GIE 设置为 1, 此中断请求将被处理。

3.20.6 PA/PB 输入引脚状态改变中断

当 PA_x ($0 \leq x \leq 7$), PB_y ($0 \leq y \leq 5$) 设置为输入口且相应的寄存器 WUPA_x、WUPB_x 位设置为 1, 当这些选定输入口上的状态变化时, 寄存器 PABIF 位将被硬件设为 1, 如果 PABIE 和 GIE 设置为 1, 此中断请求将会被处理。需注意当 PB0/PB1/PA3/PA4 /PA5 同时设置为状态变化中断和外部中断时, 设置 EIS0=1 或 EIS1=1 或 EIS2=1 将禁止 PB0/PB1/PA3/PA4 /PA5 状态变化。

3.20.7 外部中断 0

根据 EIS0=1 和寄存器 INTEDG 的配置, 如果 INT0IE 和 GIE 设置为 1, 被选择的 PB0 或 PA4 的有效边沿将会设置寄存器 INT0IF 的位为 1 并处理此中断请求。

3.20.8 外部中断 1

根据 EIS1=1 和寄存器 INTEDG 的配置, 如果 INT1IE 和 GIE 设置为 1, 被选择的 PB1 或 PA3 的有效边沿将会设置寄存器 INT1IF 的位为 1 并处理此中断请求。

3.20.9 External 2 Interrupt 外部中断 2

根据 EIS2=1 和寄存器 INTEDG 的配置, 如果 INT2IE 和 GIE 设置为 1, 被选择的 PA5 的有效边沿将会设置寄存器 INT2IF 的位为 1 并处理此中断请求。

3.20.10 低电压侦测中断/比较器输出翻转中断

当 VDD 电压值低于 LVD 电压或比较器输出状态改变时, LVDIF/CMPPIF 将被设置为 1, 如果 LVDIE/CMPIE 和 GIE 设置为 1, 此中断请求将会被处理。

3.20.11 ADC 模数转换完成中断

当 ADC 模数转换完成时, ADC 中断会被触发, 如果 ADIE 和 GIE 设置为 1, 此中断请求将会被处理。

3.20.12 EEPROM 写入完成中断

当 EEPROM 写入完成时, EEPROM 写入完成标志将会发生, 如果 EEIE 和 GIE 设置为 1, 此中断请求将会被处理。

3.21 振荡器配置

因为NY8BE62D是双时钟IC，有高振荡时钟（ F_{HOSC} ）和低振荡时钟（ F_{LOSC} ）可选择作为系统振荡时钟（ F_{OSC} ）。可用作 F_{HOSC} 的振荡器有内部高速RC振荡器（I_HRC）、外部高速晶体振荡器（E_HXT）与外部晶体振荡器（E_XT）。可用作 F_{LOSC} 的振荡器是内部低速RC振荡器（I_LRC）与外部低速晶体振荡器（E_LXT）。

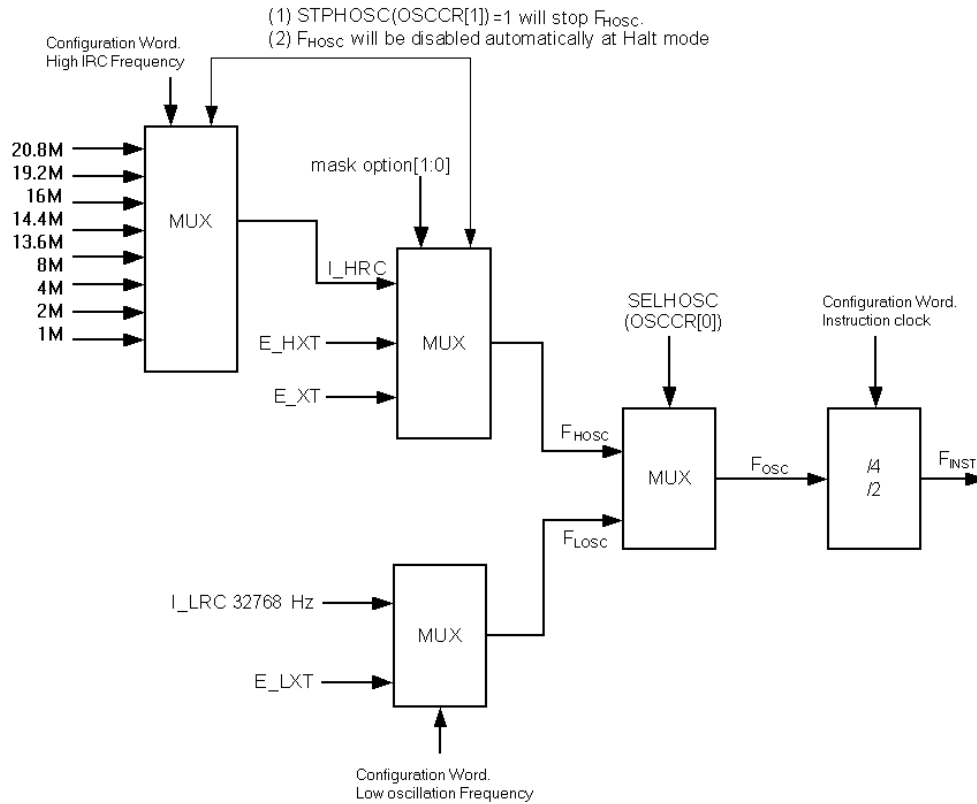


图 32 NY8BE62D振荡配置结构图

有两种配置字节来决定哪一种振荡将被用作 F_{HOSC} （系统时钟频率）。当I_HRC被选作系统时钟频率时，I_HRC的输出频率由三个配置字节决定，可以选择的频率为 1M、2M、4M、8M、16M、13.6M、14.4M、19.2M或 20.8MHz。此外，外部晶体振荡器引脚PA6 和PA7 可作为I/O脚，另一方面，PA7 引脚也可以根据配置字节的设置来输出指令时钟。如果外部晶体振荡器频率范围是 8MHz到 20MHz，配置字节请选择E_HXT。如果外部晶体振荡器是 455KHz到 6MHz，配置字节请选择E_XT。当E_HXT或E_XT被配置后，PA6/PA7 就不能被用作I/O脚，而被用作晶振输出或输入引脚，PA7 为晶振输出脚（Xout），PA6 为晶振输入脚（Xin）。

有一种配置字节来决定振荡用作 F_{LOSC} 。当配置字节选择I_LRC时，其频率约为 32768Hz。如果 F_{LOSC} 需要使用外部晶振，则配置字节应选择E_LXT且只能是 32768Hz晶体振荡器。当E_LXT被配置后，PA6/PA7 就不能被用作I/O脚，而被用作晶振输出或输入引脚，PA7 为晶振输出脚（Xout），PA6 为晶振输入脚（Xin）。

F_{HOSC} and F_{LOSC} 双时钟组合如下：

No.	F_{HOSC}	F_{LOSC}
1	I_HRC	I_LRC

2	E_HXT 或 E_XT	I_LRC
3	I_HRC	E_LXT

表 29 双时钟组合

当使用外部晶体振荡器(E_HXT, E_XT 或 E_LXT)时, 建议在Xin与Xout引脚各自接一颗电容器C1 和C2 到VSS, 除此之外, 推荐连接的一个电阻和两个电容需要根据下列的计算来提供可靠的振荡器, 参考晶体管和谐振器规格调整C1 和C2 的值。C1 和C2 数值请参考下表:

振荡模式	晶振频率(Hz)	C1, C2 (pF)
E_HXT	16M	5 ~ 10
	10M	5 ~ 30
	8M	5 ~ 20
E_XT	4M	5 ~ 30
	1M	5 ~ 30
	455K	10 ~ 100
E_LXT	32768	10 ~ 30

表 30 不同外部晶体振荡器频率所推荐的C1 和C2 电容器数值

对于 20MHz的 2 个时钟CPU循环模式下, C2 须接 18pF的电容。

为得到精准且稳定的 32768Hz频率, 选择正确的C1 和C2 电容器数值是相当重要的。你需要对你选择的特定晶体振荡器匹配C1 / C2 电容量。每家晶振厂商数据手册中都有记载低速晶体振荡器的负载电容值 (CL), 外接C1 和C2 电容器数值的计算如下公式:

$$C1=C2=2 * C_L - C_{bt}$$

C_{bt} 是 NY8BE62D 的晶振脚的内置电容, 一般为 5pF, 假设低速晶体振荡器的负载电容值 C_L=12.5pF, 依公式算出的 C1=C2=20pF。

详细请参考“[高速RC振荡频率与电源电压曲线图](#)”和“[高速RC振荡频率与温度曲线图](#)”章节, 应用上要求±1%, 也请检查烧录器和烧录座是否正确。

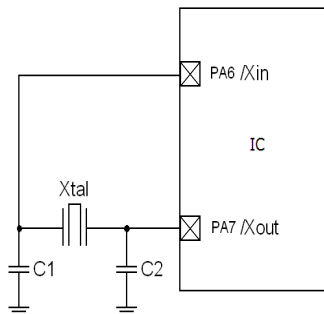
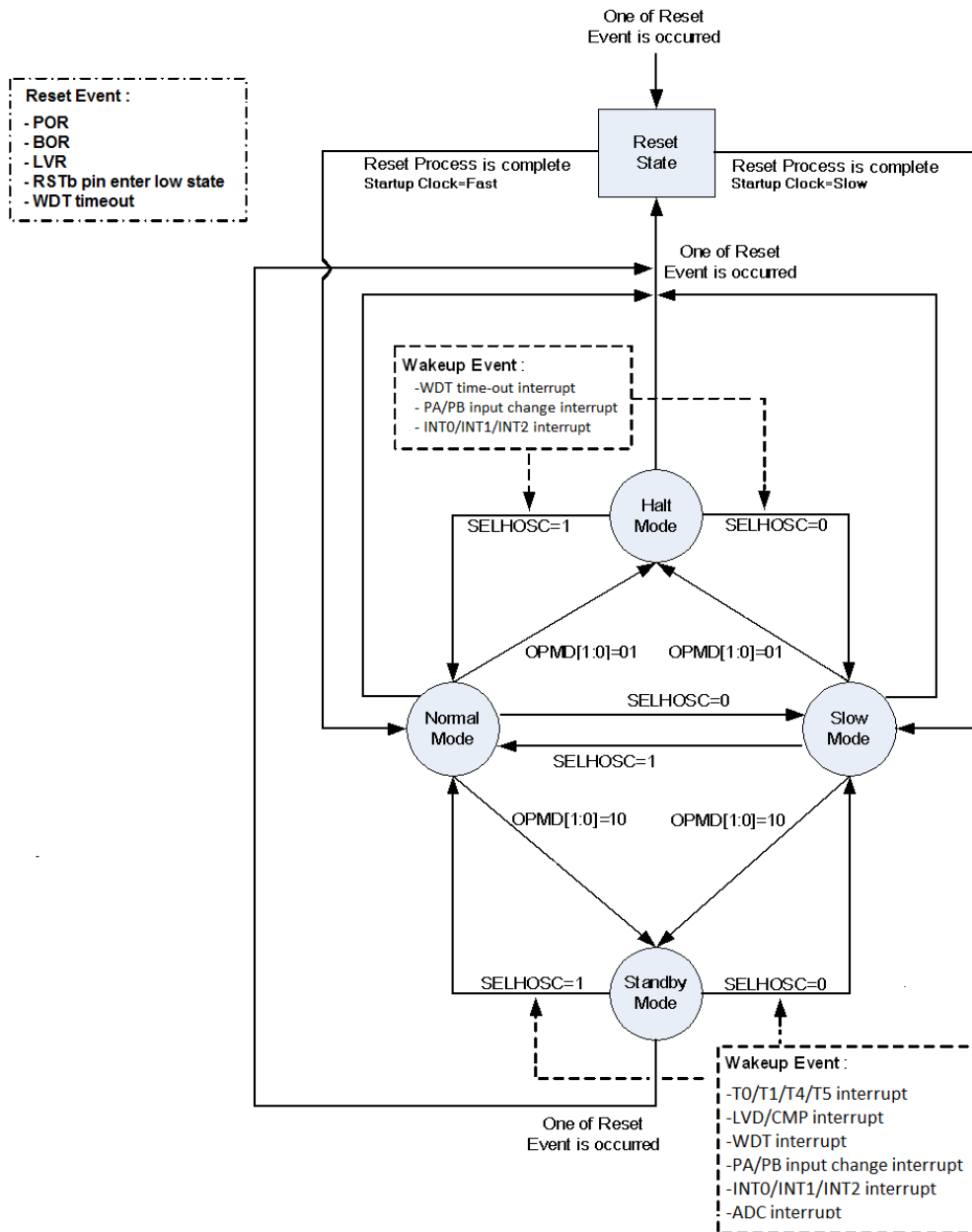


图 33 外部晶体振荡器的硬件连接图

根据寄存器SELHOSC (OSCCR [0]) 位的值, 可以选择F_{HOSC}或F_{LOSC}作为系统振荡时钟F_{OSC}。当SELHOSC为 1 时, 选择F_{HOSC}作为F_{OSC}。当SELHOSC为 0 时, 选择F_{LOSC}作为F_{OSC}。一旦确定F_{OSC}, 根据配置字节设置, 指令时钟可以选择为F_{OSC}/2 或F_{OSC}/4。

3.22 工作模式

NY8BE62D提供了四种工作模式来定制各种应用和节省电力消耗，四种模式分别是正常模式、慢速模式、待机模式和睡眠模式。正常模式被指定为高速运行模式，慢速模式被指定为低速模式，以节省功耗。在待机模式下，NY8BE62D将停止几乎所有的运作，除了定时器 0/定时器 1/定时器 4/定时器 5/ WDT，用来定期唤醒。在睡眠模式下，NY8BE62D将睡眠直到外部事件或看门狗定时器来唤醒。



四种工作模式如下图所示。

图 34 四种工作模式

3.22.1 正常模式

发生任何复位事件并且复位过程完成后，NY8BE62D将在正常模式或慢速模式下开始执行程序。重置过程后选择的模式由启动时钟配置字节决定。如果启动时钟为Fast，NY8BE62D将进入正常模式，如果启动时钟为Slow，NY8BE62D将进入慢速模式。在正常模式下，选择F_{Hosc}作为系统振荡时钟来提供最高性能，其功耗在四种操作模式中是最大的。在上电或任何重置触发器被释放后，待复位程序完成后NY8BE62D将进入正常模式。

- 指令的执行是基于F_{Hosc}且所有硬件功能可以根据相应的硬件使能位来开启/关闭。
- F_{Losc}仍运行。
- IC可由写 0 至寄存器SELHOSC（OSCCR[0]）位切换为慢速模式。
- IC可通过寄存器OPMD[1:0]（OSCCR[3:2]）位切换为待机或睡眠模式。
- 关于实时时钟的应用，NY8BE62D在运行正常模式时可同时将低频振荡时钟设为Timer0 的时钟源，这是通过设置LCKTM0 为 1 和配置字节中Timer0 时钟源来实现。

3.22.2 慢速模式

通过写 0 至寄存器SELHOSC位，NY8BE62D将进入慢速模式。在慢速模式下，为节省功耗，F_{Losc}被选为系统振荡时钟，但是IC仍然在运行。但是，F_{Hosc}不会自动被NY8BE62D关闭。因此在慢速模式下，用户可写 1 至寄存器STPHOSC（OSCCR[1]）位来停止F_{Hosc}进一步降低功耗。但需注意的是，禁止进入慢速模式同时停止F_{Hosc}，首先必须进入慢速模式，然后再关闭F_{Hosc}，否则程序可能被中止。

- 指令执行是基于F_{Losc}且所有硬件功能可以根据相应的硬件使能位来开启/关闭。
- 通过写 1 至寄存器STPHOSC位，F_{Hosc}可以被停止。
- IC可通过寄存器OPMD[1:0]位切换为待机模式或睡眠模式。
- IC可通过写 1 至寄存器SELHOSC切换到正常模式。

3.22.3 待机模式

通过写入 10b 至寄存器OPMD[1:0]，NY8BE62D将进入待机模式。然而，在待机模式下，F_{Hosc}不会自动被NY8BE62D关闭，用户必须进入先低速模式后写入 1 至寄存器STPHOSC位，以停止F_{Hosc}。大部分NY8BE62D的硬件功能会被关闭，但是如果T0EN / T1EN / T2EN / T3EN位被设置为 1 则定时器仍可运作。因此Timer0 / Timer1 / Timer4 / Timer5 溢出后NY8BE62D会被唤醒。Timer0 / Timer1 / Timer4 / Timer5 的终结时间由寄存器TMR0 / TMR1[9:0] / TMR4[9:0] / TMR5[9:0]，F_{INST}和其它配置字节决定。

- 停止执行指令且一些硬件功能可以根据相应的硬件使能位来开启/关闭。
- 由写入 1 至寄存器STPHOSC位F_{HOSC}可以被关闭。
- F_{Losc}仍保持运作。
- 如遇以下任一状况NY8BE62D便能从待机模式唤醒：
 - (a)Timer0 上溢中断 / Timer1 下溢中断 / Timer4 下溢中断 / Timer4 下溢中断
 - (b)看门狗超时中断
 - (c)PA/PB输入状态改变中断 (d)发生外部中断 0/1/2 (e)LVD/比较器输出翻转中断 (f)ADC模数转换中断。
- 在从待机模式唤醒后，如SELHOSC=1，IC将回到正常模式，如SELHOSC=0 则IC将回到慢速模式。

- 不建议在同一时间进入待机模式并改变振荡模式（正常到慢速/慢速到正常）。

3.22.4 睡眠模式

NY8BE62D通过执行SLEEP指令或写入 01b 至寄存器OPMD[1:0]位，进入睡眠模式。在进入睡眠模式后，寄存器/PD（STATUS[3]）位将清除为 0，寄存器/TO（STATUS[4]）位将设置为 1 且清除WDT并保持运作。在睡眠模式下，所有硬件功能是被关闭的，停止指令执行且NY8BE62D只能通过一些特殊事件唤醒。因此，睡眠模式是NY8BE62D最省电的模式。

- 指令执行停止，所有硬件功能关闭。
- F_{HOSC}和F_{LOSC}两者都自动关闭。
- 如遇以下任一状况NY8BE62D便能从睡眠模式中唤醒：
(a)看门狗超时中断 (b)PAPB输入状态改变中断 (c)INT0/1/2 或外部中断发生。
- 从睡眠模式唤醒后，如SELHOSC=1，IC会回到正常模式，如SELHOSC=0 则IC将会回到慢速模式。

注意：您可以在同一指令中更改STPHOSC并进入睡眠模式。

- 不建议改变振荡模式（正常到慢速/慢速到正常），并在同一时间进入待机模式。

3.22.5 唤醒稳定时间

睡眠模式的唤醒稳定时间由配置字节决定：高速振荡频率或低速振荡频率。若选择E_HXT，E_XT 或 E_LXT 其中一种作为系统振荡时钟来源，其睡眠模式的唤醒等待时间为 512*F_{osc}，若没有选择外部振荡器作为为系统振荡时钟来源，其睡眠模式的唤醒等待时间为 16*F_{osc}，由于待机模式下F_{HOSC}或F_{LOSC}仍在运行，因此无需为待机模式唤醒稳定时间。

在NY8BE62D进入待机模式或睡眠模式之前，用户可以执行指令ENI。在唤醒后，NY8BE62D将跳转到地址 0x008，以便执行中断服务程序。如果在进入待机模式或睡眠模式之前执行DISI指令，则在唤醒后执行下一条指令。

3.22.6 工作模式概述

四种工作模式概述如下：

模式	正常模式	慢速模式	待机模式	睡眠模式
F _{HOSC}	使能	STPHOSC	STPHOSC	关闭
F _{LOSC}	使能	使能	使能	关闭
指令执行	执行	执行	Stop	Stop
计时器 0/1/4/5	TxEN	TxEN	TxEN	关闭
WDT	选项和 WDTEN	选项和 WDTEN	选项和 WDTEN	选项和 WDTEN
其它硬件	硬件使能位	硬件使能位	硬件使能位	全部关闭

模式	正常模式	慢速模式	待机模式	睡眠模式
唤醒源	-	-	- 计时器 0/1/4/5 - WDT 超时 - PA/PB 输入状态改变 - 外部中断 0/1/2 - LVD/比较器中断 - ADC 模数转换结束	- WDT超时 - PA/PB输入状态改变 - 外部中断 0/1/2

表 31 工作模式概述

3.23 复位

当以下任一复位事件发生时，NY8BE62D将会进入复位状态并开始复位动作：

- 当VDD检测到上升沿时，发生上电复位（POR）。
- 当VDD电压低于预设的LVR电压时，发生LVR复位。
- RSTb引脚为低电平状态。
- WDT超时复位。

此外，所有寄存器如果初始值未知时，寄存器初始值将会被初始化或保持不变。状态位/TO和/PD可以根据复位事件来初始化。/TO和/PD的值及其相关的事件概述如下。

事件	/TO	/PD
POR, LVR	1	1
非睡眠模式时发生RSTb复位	不变	不变
睡眠模式时发生RSTb复位	1	1
非睡眠模式时发生WDT复位	0	1
睡眠模式时发生WDT复位	0	0
执行SLEEP指令	1	0
执行CLRWDT指令	1	1

表 32 /TO和/PD值和相关事件概述

复位事件发生后，NY8BE62D将会开始复位进程。无论采用什么样的振荡器，它将等待一定的周期使振荡稳定。这个周期被称为上电复位时间，它由三位配置字节决定，这个时间可能是 140us, 4.5ms, 18ms, 72ms或 288 ms。振荡器稳定后，上电复位后，在开始执行程序前，NY8BE62D将等待更多的振荡启动时间(OST)。若前一个上电复位时间为 140us时，则Fosc的时钟周期OST=1，若前一个上电复位时间为 4.5ms, 18ms, 72ms 或 288ms.时，则Fosc的时钟周期OST=16。

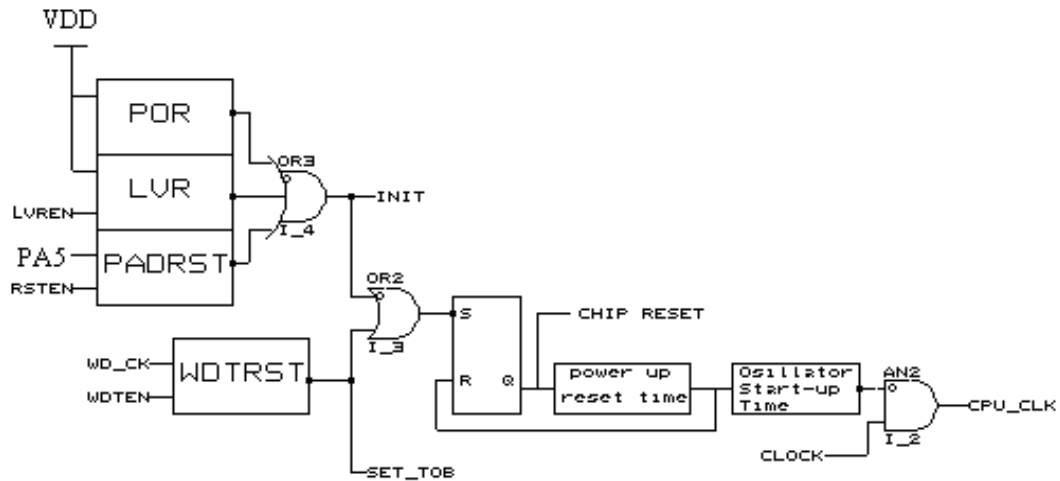


图 35 芯片复位电路框图

如果VDD缓慢上升，建议使用RSTb复位功能，如下图。

- 建议R阻值不大于 40KΩ。
- R1 值= 100Ω ~ 1KΩ时，将阻止过大电流，ESD或电气过载信号进入复位引脚。
- 二极管D使电容C能在VDD下电时快速放电。

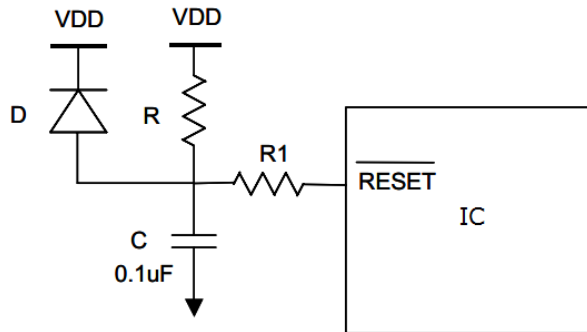


图 36 外部上电复位硬件连接图

3.24 片上仿真 (OCD)

3.24.1 功能描述

NY8BE62D内嵌片上调试仿真功能(OCD)，为开发人员提供了一种低成本的调试用户代码的方法，OCD提供了程序流程控制的调试能力，包括 3 个硬件地址断点、1 个条件寄存器中断、单步、自由运行和内存访问命令。OCD系统不占用内存映射中的任何位置，也不共享任何片上外围设备。

OCD系统使用两线串行接口SCL和SDA，在目标设备和控制调试器主机之间建立通信。SDA是用于调试数据传输的输入/输出管脚，SCL是用于与SDA同步的输入管脚。NY8BE62D还使用SCL和SDA作为控制引脚来写入和读取MTP。

3.24.2 OCD 限制

NY8BE62D是一个功能齐全的微控制器，在其有限的I/O引脚上多路复用多个功能。必须牺牲一些设备功能来为OCD系统提供资源。OCD有以下限制：

- 1、SCL/SDA引脚物理上位于同一引脚PB5/PB4 或 PA4/PA2 上。因此，它的I/O功能和共享的多功能都无法仿真。
- 2、系统时钟不能关闭，因为OCD使用该时钟来监视其内部状态：当系统处于暂停模式时，由于设备的某些部分可能没有时钟，所以执行ram/寄存器访问是无效的。读访问可能返回垃圾，或者写访问可能不会成功。但以下访问不受系统停止的影响：读取当前程序地址、当前PCL、当前中断条件和当前停止状态。

4. 指令设置

NY8BE62D为各种应用程序提供了 53 个强大的指令。

指令	助记符		说明	周期数	影响标志
	1	2			
算术指令					
ANDAR	R	d	dest = ACC & R	1	Z
IORAR	R	d	dest = ACC R	1	Z
XORAR	R	d	dest = ACC ⊕ R	1	Z
ANDIA	i		ACC = ACC & i	1	Z
IORIA	i		ACC = ACC i	1	Z
XORIA	i		ACC = ACC ⊕ i	1	Z
RRR	R	d	Rotate right R	1	C
RLR	R	d	Rotate left R	1	C
BSR	R	bit	Set bit in R	1	-
BCR	R	bit	Clear bit in R	1	-
INCR	R	d	Increase R	1	Z
DECR	R	d	Decrease R	1	Z
COMR	R	d	dest = ~R	1	Z
条件指令					
BTRSC	R	bit	Test bit in R, skip if clear	1 or 2	-
BTRSS	R	bit	Test bit in R, skip if set	1 or 2	-
INCRSZ	R	d	Increase R, skip if 0	1 or 2	-
DECRSZ	R	d	Decrease R, skip if 0	1 or 2	-
数据传送指令					
MOVAR	R		Move ACC to R	1	-
MOVR	R	d	Move R	1	Z
MOVIA	i		Move immediate to ACC	1	-
SWAPR	R	d	Swap halves R	1	-
IOST	F		Load ACC to F-page SFR	1	-
IOSTR	F		Move F-page SFR to ACC	1	-
SFUN	S		Load ACC to S-page SFR	1	-
SFUNR	S		Move S-page SFR to ACC	1	-
T0MD			Load ACC to T0MD	1	-
T0MDR			Move T0MD to ACC	1	-
TABLEA			Read ROM	2	-

指令	助记符		说明	周期数	影响标志
	1	2			
算术指令					
ADDAR	R	d	dest = R + ACC	1	Z, DC, C
SUBAR	R	d	dest = R + (~ACC)	1	Z, DC, C
ADCAR	R	d	dest = R + ACC + C	1	Z, DC, C
SBCAR	R	d	dest = R + (~ACC) + C	1	Z, DC, C
ADDIA	i		ACC = i + ACC	1	Z, DC, C
SUBIA	i		ACC = i + (~ACC)	1	Z, DC, C
ADCIA	i		ACC = i + ACC + C	1	Z, DC, C
SBCIA	i		ACC = i + (~ACC) + C	1	Z, DC, C
DAA			Decimal adjust for ACC	1	C
CMPAR	R		Compare R with ACC	1	Z, C
CLRA			Clear ACC	1	Z
CLRR			Clear R	1	Z
其它指令					
NOP			No operation	1	-
SLEEP			Go into Halt mode	1	/TO, /PD
CLRWDT			Clear Watch-Dog Timer	1	/TO, /PD
ENI			Enable interrupt	1	-
DISI			Disable interrupt	1	-
INT			Software Interrupt	3	-
RET			Return from subroutine	2	-
RETIE			Return from interrupt and enable interrupt	2	-
RETIA	i		Return, place immediate in ACC	2	-
CALLA			Call subroutine by ACC	2	-
GOTOA			unconditional branch by ACC	2	-
LCALL	adr		Call subroutine	2	-
LGOTO	adr		unconditional branch	2	-
TFUN	T		Load ACC to T-page SFR	1	
TFUNR	T		Move T-page SFR to ACC	1	

表 33 指令设置

ACC: 累加器。

adr: 地址。

bit: R-page中 8 位寄存器的位地址。

C: 进位/借位。

C=1, 加法指令有进位, 减法指令无借位。

C=0, 加法指令无进位, 减法指令有借位。

d: 目标。

若d="0", 结果存入ACC。

若d="1", 结果存入R寄存器。

DC: 半字节进位/借位标记。

dest: 目标。

F: F 页面特殊功能寄存器, F 值为 0x5~0xF。

i: 8 位立即数。

PC: 程序计数器。

PCHBUF: 程序计数器的高字节。

/PD: 睡眠标志位。

/PD=1, 上电或CLRWDT指令执行后。

/PD=0, SLEEP指令执行后。

Prescaler: 预分频器。

R: R页面特殊功能寄存器, R值为 0x00~0x7F。

S: S页面特殊功能寄存器, S值为 0x0 ~ 0x1F。

T0MD: T0MD寄存器。

TBHP: 表格指针高字节寄存器。

TBHD: 表格数据高字节寄存器。

/TO: 看门狗超时标志位。

/TO=1, 上电或执行 CLRWDT 或 SLEEP 指令后。

/TO=0, 看门狗超时。

WDT: 看门狗计时器。

Z: 清零标志。

.

ADCAR	Add ACC and R with Carry	ADDAR	Add ACC and R
语法	ADCAR R, d	语法	ADDAR R, d
操作数	$0 \leq R \leq 127$ d = 0, 1.	操作数	$0 \leq R \leq 127$ d = 0, 1.
操作	R + ACC + C dest	操作	ACC + R → dest
状态影响	Z, DC, C	状态影响	Z, DC, C
说明	ACC和R带进位加法：若d="0"，结果存入ACC；若d="1"，结果存入“R”。	说明	ACC和R加法：若d="0"，结果存入ACC；若d="1"，结果存入“R”。
周期	1	周期	1
举例	ADCAR R, d 执行指令前： ACC=0x12, R=0x34, C=1, d=1, 执行指令后： R=0x47, ACC=0x12, C=0.	举例	ADDAR R, d 执行指令前： ACC=0x12, R=0x34, C=1, d=1, 执行指令后： R=0x46, ACC=0x12, C=0.

ADCIA	Add ACC and Immediate with Carry	ADDIA	Add ACC and Immediate
语法	ADCIA i	语法	ADDIA i
操作数	$0 \leq i \leq 255$	操作数	$0 \leq i \leq 255$
操作	ACC + i + C ACC	操作	ACC + i ACC
状态影响	Z, DC, C	状态影响	Z, DC, C
说明	ACC和8位立即数带进位加法，结果存入ACC。	说明	ACC和8位立即数加法，结果存入ACC。
周期	1	周期	1
举例	ADCIA i 执行指令前： ACC=0x12, i=0x34, C=1, 执行指令后： ACC=0x47, C=0.	举例	ADDIA i 执行指令前： ACC=0x12, i=0x34, C=1, 执行指令后： ACC=0x46, C=0.

ANDAR	AND ACC and R
语法	ANDAR R, d
操作数	$0 \leq R \leq 127$ $d = 0, 1.$
操作	ACC & R \rightarrow dest
状态影响	Z
说明	ACC和R做“AND”运算；若d=“0”，结果存入ACC；若d=“1”，结果存入“R”。
周期	1
举例	ANDAR R, d 执行指令前： ACC=0x5A, R=0xAF, d=1. 执行指令后： R=0x0A, ACC=0x5A, Z=0.

BCR	Clear Bit in R
语法	BCR R, bit
操作数	$0 \leq R \leq 127$ $0 \leq \text{bit} \leq 7$
操作	$0 \rightarrow R[\text{bit}]$
状态影响	--
说明	将R寄存器的bit位(0~7) 清 0。
周期	1
举例	BCR R, B2 执行指令前： R=0x5A, B2=0x3, 执行指令后： R=0x52.

ANDIA	AND Immediate with ACC
语法	ANDIA i
操作数	$0 \leq i \leq 255$
操作	ACC & i \rightarrow ACC
状态影响	Z
说明	ACC和 8 位立即数做“AND”运算。
周期	1
举例	ANDIA i 执行指令前： ACC=0x5A, i=0xAF, 执行指令后： ACC=0x0A, Z=0.

BSR	Set Bit in R
语法	BSR R, bit
操作数	$0 \leq R \leq 127$ $0 \leq \text{bit} \leq 7$
操作	$1 \rightarrow R[\text{bit}]$
状态影响	--
说明	设置R寄存器的bit位为 1。
周期	1
举例	BSR R, B2 执行指令前： R=0x5A, B2=0x2, 执行指令后： R=0x5E.

BTRSC	Test Bit in R and skip if Clear
语法	BTRSC R, bit
操作数	$0 \leq R \leq 127$ $0 \leq \text{bit} \leq 7$
操作	skip next instruction, if R[bit] = 0.
状态影响	--
说明	位判断指令，为“0”则跳过下一条指令。
周期	1 or 2(跳过)
举例	BTRSC R, B2 指令 1 指令 2 执行指令前： R=0x5A, B2=0x2, 执行指令后： 由于R[B2]=0，则指令 1 不执行， 程序直接从指令 2 开始执行。

CALLA	Call Subroutine
语法	CALLA
操作数	--
操作	PC + 1 → Top of Stack {TBHP, ACC} → PC
状态影响	--
说明	子程序调用。首先将返回地址PC+1压入栈顶，然后将TBHP[2:0]赋值给PC[10:8]，将ACC赋值给PC[7:0]。
周期	2
举例	CALLA 执行指令前： TBHP=0x02, ACC=0x34. PC=A0. Stack pointer=1. 执行指令后： PC=0x234, Stack[1]=A0+1, Stack pointer=2

BTRSS	Test Bit in R and skip if Set
语法	BTRSS R, bit
操作数	$0 \leq R \leq 127$ $0 \leq \text{bit} \leq 7$
操作	skip next instruction, if R[bit] = 1.
状态影响	--
说明	位判断指令，为“1”则跳过下一条指令。
周期	1 or 2(跳过)
举例	BTRSS R, B2 指令 2 指令 3 执行指令前： R=0x5A, B2=0x3, 执行指令后： 由于R[B2]=1，则指令 2 不执行， 直接从指令 3 开始执行。

CLRA	Clear ACC
语法	CLRA
操作数	--
操作	00h → ACC 1 → Z
状态影响	Z
说明	ACC清零，Z标志位置“1”。
周期	1
举例	CLRA 执行指令前： ACC=0x55, Z=0. 执行指令后： ACC=0x00, Z=1.

CLRR	Clear R
语法	CLRR R
操作数	$0 \leq R \leq 127$
操作	00h \rightarrow R 1 \rightarrow Z
状态影响	Z
说明	寄存器R清零，Z标志位置“1”。
周期	1
举例	CLRR R 执行指令前： R=0x55, Z=0. 执行指令后： R=0x00, Z=1.

COMR	Complement R
语法	COMR R, d
操作数	$0 \leq R \leq 127$ d = 0, 1.
操作	$\sim R \rightarrow \text{dest}$
状态影响	Z
说明	R寄存器取反，结果存入d； d=“0”，结果存入ACC； d=“1”，结果存入R。
周期	1
举例	COMR, d 执行指令前： R=0xA6, d=1, Z=0. 执行指令后： R=0x59, Z=0.

CLRWDT	Clear Watch-Dog Timer
语法	CLRWDT
操作数	--
操作	00h \rightarrow WDT, 00h \rightarrow WDT prescaler 1 \rightarrow /TO 1 \rightarrow /PD
状态影响	/TO, /PD
说明	清WDT计数器和预分频器；/TO和/PD标志位置“1”。
周期	1
举例	CLRWDT 执行指令前： /TO=0 执行指令后： /TO=1

CMPAR	Compare ACC and R
语法	CMPAR R
操作数	$0 \leq R \leq 127$
操作	R - ACC \rightarrow (No restore)
状态影响	Z, C
说明	ACC和R比较：执行R-ACC，不改变ACC和R的值，只改变Z和C标志位。
周期	1
举例	CMPAR R 执行指令前： R=0x34, ACC=12, Z=0, C=0. 执行指令后： R=0x34, ACC=12, Z=0, C=1.

DAA Convert ACC Data Format from Hexadecimal to Decimal

语法	DAA
操作数	--
操作	ACC(hex) \rightarrow ACC(dec)
状态影响	C
说明	将累加器中的 16 进制数调整为十进制数，该指令必须紧跟在加法指令后。
周期	1
举例	DISI ADDAR R,d DAA ENI 执行指令前： ACC=0x28, R=0x25, d=0. 执行指令后： ACC=0x53, C=0.

DECRSZ Decrease R, skip if 0

语法	DECRSZ R, d
操作数	$0 \leq R \leq 127$ $d = 0, 1.$
操作	$R - 1 \rightarrow \text{dest},$ skip if result = 0
状态影响	--
说明	R 先- 1, 若d="0", 结果存入ACC; 若d="1", 结果存入R, 若结果为"0" 则跳过下一条指令, 改为执行NOP 指令, 因此结果为"0"时要执行两个 周期。
周期	1 or 2(跳过)
举例	DECRSZ R, d 指令 2 指令 3 执行指令前： R=0x1, d=1, Z=0. 执行指令后： R=0x0, Z=1, 操作结果为 0, 指 令 2 被跳过。

DECR Decrease R

语法	DECR R, d
操作数	$0 \leq R \leq 127$ $d = 0, 1.$
操作	$R - 1 \rightarrow \text{dest}$
状态影响	Z
说明	R - 1, 若d="0", 结果存入ACC; 若d="1", 结果存入R。
周期	1
举例	DECR R, d 执行指令前： R=0x01, d=1, Z=0. 执行指令后： R=0x00, Z=1.

DISI Disable Interrupt Globally

语法	DISI
操作数	--
操作	Disable Interrupt, $0 \rightarrow \text{GIE}$
状态影响	--
说明	GIE设置为 0, 关闭总中断。
周期	1
举例	DISI 执行指令前： GIE=1, 执行指令后： GIE=0.

ENI	Enable Interrupt Globally
语法	ENI
操作数	--
操作	Enable Interrupt, $1 \rightarrow GIE$
状态影响	--
说明	GIE设置为 1，开启总中断。
周期	1
举例	ENI 执行指令前： GIE=0, 执行指令后： GIE=1.

INCR	Increase R
语法	INCR R, d
操作数	$0 \leq R \leq 127$ $d = 0, 1.$
操作	$R + 1 \rightarrow \text{dest}.$
状态影响	Z
说明	R+ 1, 若d="0", 结果存入ACC; 若d="1", 结果存入R。
周期	1
举例	INCR R, d 执行指令前： R=0xFF, d=1, Z=0. 执行指令后： R=0x00, Z=1.

GOTOA	Unconditional Branch
语法	GOTOA
操作数	--
操作	{TBHP, ACC} \rightarrow PC
状态影响	--
说明	无条件跳转指令，ACC 值写入 PC[7:0]；TBHP[2:0] 值写入 PC[10:8]。
周期	2
举例	GOTOA 执行指令前： PC=A0. TBHP=0x02, ACC=0x34. 执行指令后： PC=0x234

INCRSZ	Increase R, skip if 0
语法	INCRSZ R, d
操作数	$0 \leq R \leq 127$ $d = 0, 1.$
操作	$R + 1 \rightarrow \text{dest},$ skip if result = 0
状态影响	--
说明	R先+ 1, 若d="0", 结果存入ACC; 若d="1", 结果存入R。若结果为"0" 则跳过下一条指令（执行NOP指令）。
周期	1 or 2(跳过)
举例	INCRSZ R, d 指令 2, 指令 3. 执行指令前： R=0xFF, d=1, Z=0. 执行指令后： R=0x00, Z=1, 因结果为 0, 程序 跳过指令 2。

INT	Software Interrupt
语法	INT
操作数	--
操作	PC + 1 → Top of Stack, 001h → PC
状态影响	--
说明	软中断指令。首先将返回地址 (PC+1) 压入栈顶, 然后将 001H 的地址装入PC[10:0]。
周期	3
举例	INT 执行指令前: PC=address of INT code 执行指令后: PC=0x01

IORIA	OR Immediate with ACC
语法	IORIA i
操作数	$0 \leq i \leq 255$
操作	ACC i → ACC
状态影响	Z
说明	ACC和 8 位立即数做“OR”运算, 结果存入ACC。
周期	1
举例	IORIA i 执行指令前: i=0x50, ACC=0xAA, Z=0. 执行指令后: ACC=0xFA, Z=0.

IORAR	OR ACC with R
语法	IORAR R, d
操作数	$0 \leq R \leq 127$ d = 0, 1.
操作	ACC R → dest
状态影响	Z
说明	ACC和R做“OR”运算, 若d=“0”, 结果存入ACC; 若d=“1”, 结果存入R。
周期	1
举例	IORAR R, d 执行指令前: R=0x50, ACC=0xAA, d=1, Z=0. 执行指令后: R=0xFA, ACC=0xAA, Z=0.

IOST	Load F-page SFR from ACC
语法	IOST F
操作数	$5 \leq F \leq 15$
操作	ACC → F-page SFR
状态影响	--
说明	将ACC的值赋给F-page特殊寄存器
周期	1
举例	IOST F 执行指令前: F=0x55, ACC=0xAA. 执行指令后: F=0xAA, ACC=0xAA.

IOSTR	Move F-page SFR to ACC
语法	IOSTR F
操作数	$5 \leq F \leq 15$
操作	F-page SFR \rightarrow ACC
状态影响	--
说明	将F-page特殊寄存器数值给ACC。
周期	1
举例	IOSTR F 执行指令前: F=0x55, ACC=0xAA. 执行指令后: F=0x55, ACC=0x55.

LGOTO	Unconditional Branch
语法	LGOTO adr
操作数	$0 \leq \text{adr} \leq 2047$
操作	adr \rightarrow PC[10:0].
状态影响	--
说明	无条件长跳转，11 位立即数写入 PC[10:0]。
周期	2
举例	LGOTO Level 执行指令前: PC=A0. 执行指令后: PC=address of Level.

LCALL	Call Subroutine
语法	LCALL adr
操作数	$0 \leq \text{adr} \leq 2047$
操作	PC + 1 \rightarrow Top of Stack, adr \rightarrow PC[10:0]
状态影响	--
说明	长调用子程序。首先将PC+1 压入栈顶，然后将 11 位立即数载入 PC[10:0]。
周期	2
举例	LCALL SUB 执行指令前: PC=A0, Stack level=1 执行指令后: PC=address of SUB, Stack[1]= A0+1, Stack pointer =2.

MOVAR	Move ACC to R
语法	MOVAR R
操作数	$0 \leq R \leq 127$
操作	ACC \rightarrow R
状态影响	--
说明	ACC赋值给R-page寄存器。
周期	1
举例	MOVAR R 执行指令前: R=0x55, ACC=0xAA. 执行指令后: R=0xAA, ACC=0xAA.

MOVIA Move Immediate to ACC

语法 MOVIA i

操作数 $0 \leq i \leq 255$

操作 $i \rightarrow \text{ACC}$

状态影响 --

说明 8 位立即数赋值给ACC。

周期 1

举例 MOVIA i
 执行指令前:
 i=0x55, ACC=0xAA.
 执行指令后:
 ACC=0x55.

NOP No Operation

语法 NOP

操作数 --

操作 No operation.

状态影响 --

说明 空操作。

周期 1

举例 NOP
 执行指令前:
 PC=A0
 执行指令后:
 PC=A0+1

MOVR Move R to ACC or R

语法 MOVR R, d

操作数 $0 \leq R \leq 127$
 $d = 0, 1.$

操作 $R \rightarrow \text{dest}$

状态影响 Z

说明 R-page寄存器赋值给d, 若d="0",
 结果存入ACC; 若d="1", 结果存入
 寄存器R。指令执行后, 通过状态标
 位Z检查R是否为 0。

周期 1

举例 MOVR R, d
 执行指令前:
 R=0x0, ACC=0xAA, Z=0, d=0.
 执行指令后:
 R=0x0, ACC=0x00, Z=1.

**RETIE Return from Interrupt and
 Enable Interrupt Globally**

语法 RETIE

操作数 --

操作 Top of Stack \rightarrow PC
 $1 \rightarrow \text{GIE}$

状态影响 --

说明 中断返回, 栈顶地址载入PC同时使
 能中断。

周期 2

举例 RETIE
 执行指令前:
 GIE=0, Stack level=2.
 执行指令后:
 GIE=1, PC=Stack[2], Stack
 pointer=1.

RETIA Return with Data in ACC

语法 RETIA i
 操作数 $0 \leq i \leq 255$
 操作 $i \rightarrow \text{ACC}$,
 Top of Stack $\rightarrow \text{PC}$
 状态影响 --
 说明 带参数返回：8 位立即数赋值给 ACC，栈顶地址载入 PC。

周期 2
 举例 RETIA i
 执行指令前：
 Stack pointer =2. i=0x55,
 ACC=0xAA.
 执行指令后：
 PC=Stack[2], Stack pointer =1.
 ACC=0x55.

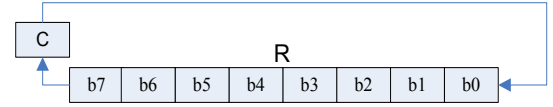
RET Return from Subroutine

语法 RET
 操作数 --
 操作 Top of Stack $\rightarrow \text{PC}$
 状态影响 --
 说明 子程序返回，栈顶载入 PC。

周期 2
 举例 RET
 执行指令前：
 Stack level=2.
 执行指令后：
 PC=Stack[2], Stack level=1.

RLR Rotate Left R Through Carry

语法 RLR R, d
 操作数 $0 \leq R \leq 127$
 $d = 0, 1$.
 操作 $C \rightarrow \text{dest}[0]$, $R[7] \rightarrow C$,
 $R[6:0] \rightarrow \text{dest}[7:1]$

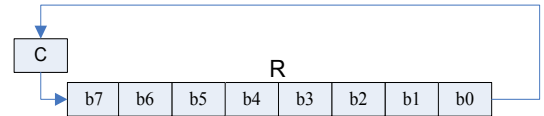


状态影响 C
 说明 带进位 R 循环左移：若 d="0"，结果存入 ACC；若 d="1"，结果存入 R。

周期 1
 举例 RLR R, d
 执行指令前：
 R=0xA5, d=1, C=0.
 执行指令后：
 R=0x4A, C=1.

RRR Rotate Right R Through Carry

语法 RRR R, d
 操作数 $0 \leq R \leq 127$
 $d = 0, 1$.
 操作 $C \rightarrow \text{dest}[7]$, $R[7:1] \rightarrow \text{dest}[6:0]$,
 $R[0] \rightarrow C$



状态影响 C
 说明 带进位 R 循环右移：若 d="0"，结果存入 ACC；若 d="1"，结果存入 R。

周期 1
 举例 RRR R, d
 执行指令前：
 R=0xA5, d=1, C=0.
 执行指令后：
 R=0x52, C=1.

SBCAR	Subtract ACC and Carry from R	SBCIA	Subtract ACC and Carry from Immediate
语法	SBCAR R, d	语法	SBCIA i
操作数	$0 \leq R \leq 127$ $d = 0, 1.$	操作数	$0 \leq i \leq 255$
操作	$R + (\sim\text{ACC}) + C \rightarrow \text{dest}$	操作	$i + (\sim\text{ACC}) + C \rightarrow \text{dest}$
状态影响	Z, DC, C	状态影响	Z, DC, C
说明	R和ACC带借位减法，若d="0"，结果存入ACC；若d="1"，结果存入R。	说明	常数和ACC带借位减法，结果存入ACC。
周期	1	周期	1
举例	SBCAR R, d (a) 执行指令前： R=0x05, ACC=0x06, d=1, C=0, 执行指令后： R=0xFE, C=0. (-2) (b) 执行指令前： R=0x05, ACC=0x06, d=1, C=1, 执行指令后： R=0xFF, C=0. (-1) (c) 执行指令前： R=0x06, ACC=0x05, d=1, C=0, 执行指令后： R=0x00, C=1. (-0), Z=1. (d) 执行指令前： R=0x06, ACC=0x05, d=1, C=1, 执行指令后： R=0x1, C=1. (+1)	举例 SBCIA i (a) 执行指令前： i=0x05, ACC=0x06, C=0, 执行指令后： ACC=0xFE, C=0. (-2) (b) 执行指令前： i=0x05, ACC=0x06, C=1, 执行指令后： ACC=0xFF, C=0. (-1) (c) 执行指令前： i=0x06, ACC=0x05, C=0, 执行指令后： ACC=0x00, C=1. (-0), Z=1. (d) 执行指令前： i=0x06, ACC=0x05, C=1, 执行指令后： ACC=0x1, C=1. (+1)	
		SFUN	Load S-page SFR from ACC
		语法	SFUN S
		操作数	$0 \leq S \leq 31$
		操作	$\text{ACC} \rightarrow \text{S-page SFR}$
		状态影响	--
		说明	ACC写到S-page特殊寄存器。
		周期	1
		举例	SFUN S 执行指令前： S=0x55, ACC=0xAA. 执行指令后： S=0xAA, ACC=0xAA.

SFUNR Move S-page SFR from ACC

语法	SFUNR S
操作数	$0 \leq S \leq 31$
操作	S-page SFR \rightarrow ACC
状态影响	--
说明	读S-page特殊寄存器到ACC
周期	1
举例	SFUNR S 执行指令前: S=0x55, ACC=0xAA. 执行指令后: S=0x55, ACC=0x55.

SUBAR Subtract ACC from R

语法	SUBAR R, d
操作数	$0 \leq R \leq 127$ $d = 0, 1.$
操作	$R - ACC \rightarrow \text{dest}$
状态影响	Z, DC, C
说明	R 减去ACC, 若d="0", 结果存入ACC。若d="1", 结果存入R。
周期	1
举例	SBCAR R, d (a) 执行指令前: R=0x05, ACC=0x06, d=1, 执行指令后: R=0xFF, C=0. (-1) (b) 执行指令前: R=0x06, ACC=0x05, d=1, 执行指令后: R=0x01, C=1. (+1)

SLEEP Enter Halt Mode

语法	SLEEP
操作数	--
操作	00h \rightarrow WDT, 00h \rightarrow WDT prescaler 1 \rightarrow /TO 0 \rightarrow /PD
状态影响	/TO, /PD
说明	WDT和分频器 0 清零。/TO标志为0, /PD清零, IC进入睡眠。
周期	1
举例	SLEEP 执行指令前: /PD=1, /TO=0. 执行指令后: /PD=0, /TO=1.

SUBIA Subtract ACC from Immediate

语法	SUBIA i
操作数	$0 \leq i \leq 255$
操作	$i - ACC \rightarrow ACC$
状态影响	Z, DC, C
说明	8 位立即数减ACC, 结果存入ACC。
周期	1
举例	SUBIA i (a) 执行指令前: i=0x05, ACC=0x06. 执行指令后: ACC=0xFF, C=0. (-1) (b) 执行指令前: i=0x06, ACC=0x05, d=1, 执行指令后: ACC=0x01, C=1. (+1)

SWAPR	Swap High/Low Nibble in R
语法	SWAPR R, d
操作数	$0 \leq R \leq 127$ d = 0, 1.
操作	R[3:0] → dest[7:4]. R[7:4] → dest[3:0]
状态影响	--
说明	寄存器半字节交换, 若d="0", 结果存入ACC; 若d="1", 结果存入R。
周期	1
举例	SWAPR R, d 执行指令前: R=0xA5, d=1. 执行指令后: R=0x5A.

T0MD	Load ACC to T0MD
语法	T0MD
操作数	--
操作	ACC → T0MD
状态影响	--
说明	ACC写入T0MD寄存器。
周期	1
举例	T0MD 执行指令前: T0MD=0x55, ACC=0xAA. 执行指令后: T0MD=0xAA.

TABLEA	Read ROM data
语法	TABLEA
操作数	--
操作	ROM data{ TBHP, ACC } [7:0] → ACC ROM data{TBHP, ACC} [13:8] → TBHD.
状态影响	--
说明	ROM查表指令, 高字节存入TBHD, 低字节存入ACC。
周期	2
举例	TABLEA 执行指令前: TBHP=0x02, CC=0x34. TBHD=0x01. ROM data[0x234]= 0x35AA 执行指令后: TBHD=0x35, ACC=0xAA.

T0MDR	Move T0MD to ACC
语法	T0MDR
操作数	--
操作	T0MD → ACC
状态影响	--
说明	读T0MD寄存器到ACC。
周期	1
举例	T0MDR 执行指令前: T0MD=0x55, ACC=0xAA. 执行指令后: ACC=0x55.

XORAR	Exclusive-OR ACC with R
语法	XORAR R, d
操作数	$0 \leq R \leq 127$ $d = 0, 1.$
操作	$ACC \oplus R \rightarrow dest$
状态影响	Z
说明	ACC和R做“XOR”运算，若d=“0”，结果存入ACC；若d=“1”，结果存入R。
周期	1
举例	XORAR R, d 执行指令前： R=0xA5, ACC=0xF0, d=1. 执行指令后： R=0x55.

TFUN	Load T-page SFR from ACC
语法	TFUN T
操作数	$0 \leq T \leq 31$
操作	ACC \rightarrow T-page SFR
状态影响	--
说明	ACC写到T-page特殊寄存器。
周期	1
举例	TFUN T 执行指令前： T=0x55, ACC=0xAA. 执行指令后： T=0xAA, ACC=0xAA.

XORIA	Exclusive-OR Immediate with ACC
语法	XORIA i
操作数	$0 \leq i \leq 255$
操作	$ACC \oplus i \rightarrow ACC$
状态影响	Z
说明	ACC和 8 位立即数做“XOR”运算。
周期	1
举例	XORIA i 执行指令前： i=0xA5, ACC=0xF0. 执行指令后： ACC=0x55.

TFUNR	Move T-page SFR from ACC
语法	TFUNR T
操作数	$0 \leq T \leq 31$
操作	T-page SFR \rightarrow ACC
状态影响	--
说明	读T-page特殊寄存器到ACC
周期	1
举例	TFUNR T 执行指令前： T=0x55, ACC=0xAA. 执行指令后： T=0x55, ACC=0x55.

5. 配置字节表

项目	名称	选项
1	高频振荡模式	1. I_HRC 2. E_HXT 3. E_XT
2	低频振荡模式	1. I_LRC 2. E_LXT
3	内部高速 RC 频率	1. 13.6MHz 2. 14.4MHz 3. 16MHz 4. 19.2MHz 5. 20.8MHz
4	除频器	1. div 1 2. div 2 3. div 4 4. div 8 5. div 16
5	外部高频振荡器	1. 8MHz > F _{HOSC} > 6MHz 2. 10MHz > F _{HOSC} ≧ 8MHz 3. 12MHz > F _{HOSC} ≧ 10MHz 4. 16MHz > F _{HOSC} ≧ 12MHz 5. 20MHz > F _{HOSC} ≧ 16MHz 6. 20MHz
6	指令时钟	1. 2 个振荡周期 2. 4 个振荡周期
7	看门狗定时器 (WDT)	1. 看门狗开启 (软件控制) 2. 看门狗关闭 (永远关闭)
8	看门狗定时器事件	1. 看门狗复位 2. 看门狗中断
9	定时器 0 时钟源	1. EX_CK10 2. 低频振荡器 (I_LRC/E_LXT)
10	PA.5	1. PA.5 is I/O 2. PA.5 is reset
11	PA.7	1. PA.7 is I/O 2. PA.7 is 指令 clock output
12	IR 引脚	1. PB1 2. PA3
13	上电复位时间	1. 140us 2. 4.5ms 3. 18ms 4. 72ms 5. 288ms
14	看门狗定时器时基	1. 3.5ms 2. 15ms 3. 60ms 4. 250ms
15	噪声过滤功能 (High_EFT)	1. 开启 2. 关闭
16	LVR 开关设定	1. 寄存器控制 2. 寄存器控制+睡眠模式自动关闭 3. LVR 永远开启 4. 一般模式开启+睡眠模式自动关闭
17	LVR 电压	1. 1.6V 2. 1.8V 3. 2.0V 4. 2.2V 5. 4V 6. 2.7V 7. 3.0V 8. 3.3V 9. 3.6V 10. 2V
18	VDD 电压	1. 3.0V 2. 4.5V 3. 5.0V
19	EEWR 超时	1. 开启 2. 关闭
20	EEPL 模式	1. 一次 2. 连续
21	灌电流	1. 小灌电流 2. 一般灌电流 3. 大灌电流
22	推电流	1. 小推电流 2. 一般推电流
23	比较器输入引脚选择	1. 开启 2. 关闭
24	读取输出口数据	1. I/O 口 2. 寄存器

项目	名称	选项
25	E_LXT 强化起振开关设定	1. 自动关闭 2. 寄存器关闭
26	EX_CKIO to Inst. Clock	1. 同步 2. 不同步
27	上电时钟源	1. 高速 (I_HRC/E_HXT/E_XT) 2. 低速 (I_LRC/E_LXT)
28	输入高电压 (VIH)	1. 0.7VDD 2. 0.5VDD
33	输入低电压 (VIL)	1. 0.3VDD 2. 0.2VDD
36	输入电压(VIH/VIL)	1. 施密特触发器 2. 无施密特触发器
37	PWM1 输出引脚	1. PB1 2. PB4
38	PWM2 输出引脚	1. PA4 2. PB5
39	PWM3 输出引脚	1. PA2 2. PA7
40	INT0 输入	1. PB0 2. PB5
41	INT1 输入	1. PB1 2. PA3
42	INT2 输入	1. 无 2. PA5
43	外部时钟输入 1	1. PA1 2. PA2

表 34 配置字节

6. 电气特性

6.1 最大绝对值

符号	参数	额定值	单位
$V_{DD} - V_{SS}$	工作电压	-0.5 ~ +6.0	V
V_{IN}	输入电压	$V_{SS}-0.3V \sim V_{DD}+0.3$	V
T_{OP}	工作温度	-40 ~ +85	°C
T_{ST}	储存温度	-40 ~ +125	°C

6.2 直流电气特性

($F_{INST}=F_{HOSC}/4$, $F_{HOSC}=16MHz@I_HRC$, WDT开启, 环境温度 $T_A=25^{\circ}C$)

符号	参数	V_{DD}	最小值	典型值	最大值	单位	条件
V_{DD}	工作电压	--	3.3	--	5.5	V	$F_{INST}=20.8MHz @ I_HRC/2$
			3.0				$F_{INST}=19.2MHz @ I_HRC/2$
			2.7				$F_{INST}=16MHz @ I_HRC/2 \& E_HXT/2$
			2.7				$F_{INST}=14.4MHz @ I_HRC/2$
			2.7				$F_{INST}=13.6MHz @ I_HRC/2$
			2.2				$F_{INST}=20.8MHz @ I_HRC/4$
			2.2				$F_{INST}=19.2MHz @ I_HRC/4$
			2.0				$F_{INST}=14.4MHz @ I_HRC/4$
			2.0				$F_{INST}=13.6MHz @ I_HRC/4$
			2.0				$F_{INST}=16MHz @ I_HRC/4 \& E_HXT/4$
			2.0				$F_{INST}=8MHz @ I_HRC/2 \& E_HXT/2$
			1.8				$F_{INST}=8MHz @ I_HRC/4 \& E_HXT/4$
			1.6				$F_{INST}=4MHz @ I_HRC/2 \& E_XT /2$
							$F_{INST}=4MHz @ I_HRC/4$
V_{IH}	输入高电平	5V	4.0	--	--	V	RSTb (0.8 V_{DD})
		3V	2.4	--	--	V	所有I/O引脚, EX_CKIO/1, INT0/1 CMOS (0.7 V_{DD})
		5V	3.5	--	--		
		3V	2.1	--	--	V	所有I/O引脚, EX_CKIO/1 TTL (0.5 V_{DD})
		5V	2.5	--	--	V	所有I/O引脚, EX_CKI, INT 无施密特触发器 (0.4 V_{DD})
		3V	1.5	--	--		
		3V	1.2	--	--		
V_{IL}	输入低电平	5V	--	--	1.0	V	RSTb (0.2 V_{DD})
		3V	--	--	0.6	V	所有I/O引脚, EX_CKIO/1, INT0/1 CMOS option (0.3 V_{DD})
		5V	--	--	1.5		
		3V	--	--	0.9	V	所有I/O引脚, EX_CKIO/1 TTL电平(0.2 V_{DD})
		5V	--	--	1.0		
		3V	--	--	0.6		

符号	参数	V _{DD}	最小值	典型值	最大值	单位	条件
		5V	--	--	2.0	V	所有I/O引脚, EX_CKI, INT 无施密特触发器(0.4 V _{DD})
		3V	--	--	1.2		
I _{OH}	输出高推电流 (小电流)	5V	--	2.4	--	mA	V _{OH} =4.0V
		3V	--	1.3	--		V _{OH} =2.0V
	输出高推电流 (一般电流)	5V	--	26	--	mA	V _{OH} =4.0V
		3V	--	14	--		V _{OH} =2.0V
I _{OL}	输出低灌电流 (小电流)	5V	--	12	--	mA	V _{OL} =1.0V
		3V	--	6.7	--		V _{OL} =1.0V
	输出低灌电流 (一般电流)	5V	--	34	--	mA	V _{OL} =1.0V
		3V	--	20	--		V _{OL} =1.0V
I _{OL}	输出低灌电流 (大电流)	5V	--	95	--	mA	V _{OL} =1.0V
		3V	--	60	--		V _{OL} =1.0V
	输出低灌电流 (定电流)	5V	--	26	--	mA	V _{OL} =1.0V
		3V	--	24	--		V _{OL} =1.0V
I _{LIR}	红外输出大灌电流	5V	--	420	--	mA	V _{OL} =1.0V, Large IR
		3V	--	340	--		
I _{OP}	工作电流	正常模式					
		5V	--	4.1	--	mA	F _{HOSC} =20.8MHz @ I _{HRC} /2
		3V	--	2.6	--		
		5V	--	3.1	--	mA	F _{HOSC} =20.8MHz @ I _{HRC} /4
		3V	--	2.0	--		
		5V	--	3.8	--	mA	F _{HOSC} =19.2MHz @ I _{HRC} /2
		3V	--	2.5	--		
		5V	--	2.9	--	mA	F _{HOSC} =19.2MHz @ I _{HRC} /4
		3V	--	1.9	--		
		5V	--	3.3	--	mA	F _{HOSC} =14.4MHz @ I _{HRC} /2
		3V	--	2.2	--		
		5V	--	2.6	--	mA	F _{HOSC} =14.4MHz @ I _{HRC} /4
		3V	--	1.8	--		
		5V	--	3.1	--	mA	F _{HOSC} =13.6MHz @ I _{HRC} /2
		3V	--	2.0	--		
		5V	--	2.5	--	mA	F _{HOSC} =13.6MHz @ I _{HRC} /4
		3V	--	1.7	--		
		5V	--	3.3	--	mA	F _{HOSC} =16MHz @ I _{HRC} /2 & E _{HXT} /2
		3V	--	2.1	--		
		5V	--	2.6	--	mA	F _{HOSC} =16MHz @ I _{HRC} /4 & E _{HXT} /4
3V	--	1.8	--				
5V	--	2.5	--	mA	F _{HOSC} =8MHz @ I _{HRC} /2 & E _{HXT} /2		
3V	--	1.7	--				
5V	--	1.6	--	mA	F _{HOSC} =8MHz @ I _{HRC} /4 & E _{HXT} /4		
3V	--	1.1	--				

符号	参数	V _{DD}	最小值	典型值	最大值	单位	条件		
		5V	--	1.6	--	mA	F _{HOSC} =4MHz @ I _{HRC} /2 & E _{XT} /2		
		3V	--	1.1	--				
		5V	--	1.2	--	mA	F _{HOSC} =4MHz @ I _{HRC} /4 & E _{XT} /4		
		3V	--	0.8	--				
		5V	--	0.9	--	mA	F _{HOSC} =1MHz @ I _{HRC} /2 & E _{XT} /2		
		3V	--	0.5	--				
		5V	--	0.7	--	mA	F _{HOSC} =1MHz @ I _{HRC} /4 & E _{XT} /4		
		3V	--	0.4	--				
		慢速模式							
		5V	--	14	--	uA	F _{HOSC} 关闭, F _{LOSC} =32KHz @ I _{LRC} /2		
		3V	--	10	--				
		5V	--	19	--	uA	F _{HOSC} 关闭, F _{LOSC} =32KHz @ E _{LXT} /2		
		3V	--	12	--				
		5V	--	9	--	uA	F _{HOSC} 关闭, F _{LOSC} =32KHz @ I _{LRC} /4		
3V	--	6	--						
5V	--	13	--	uA	F _{HOSC} 关闭, F _{LOSC} =32KHz @ E _{LXT} /4				
3V	--	7	--						
I _{STB}	待机电流	5V	--	3.8	--	uA	待机模式, F _{HOSC} 关闭, F _{LOSC} =32KHz @ I _{LRC} /4		
		3V	--	1.6	--				
I _{HALT}	睡眠电流	5V	--	--	0.5	uA	睡眠模式, WDT关闭		
		3V	--	--	0.2				
		5V	--	--	5.0	uA	睡眠模式, WDT开启		
		3V	--	--	2.0				
R _{PH}	上拉电阻	5V	--	60	--	KΩ	上拉电阻(不包括PA5)		
		3V	--	120	--				
		5V	--	90	--	KΩ	上拉电阻(PA5)		
		3V	--	90	--				
R _{PL}	下拉电阻	5V	--	50	--	KΩ	下拉电阻		
		3V	--	100	--				

6.3 OSC 特性

(量测条件: 电压 V_{DD} 、环境温度 T_A 与烧录时条件相同)

参数	最小值	典型值	最大值	单位	条件
I_HRC 偏移量 (通过转接座烧录)			± 1	%	用转接座直接从烧录器烧录。
I_HRC 偏移量 (通过夹具烧录)			± 3	%	正确设置自动机台夹具时烧录。
I_LRC 偏移量 (通过夹具烧录)			± 5	%	-

6.4 比较器/LVD电气特性

($V_{DD}=5V$, $V_{SS}=0V$, $T_A=25^\circ C$ 除其他指定说明外)

符号	参数	最小值	典型值	最大值	单位	条件
V_{IVR}	比较器输入电压范围	0	--	5	V	$F_{HOSC}=1MHz$
T_{ENO}	比较器启动等待时间	--	20	--	μs	$F_{HOSC}=1MHz$
I_{CO}	比较器电流消耗	--	135	--	μA	$F_{HOSC}=1MHz$, P2V mode
I_{LVD}	LVD电流消耗	--	150	--	μA	$F_{HOSC}=1MHz$, LVD=4.3V
E_{LVD}	LVD电压误差	--	--	3	%	$F_{HOSC}=1MHz$, LVD=4.3V

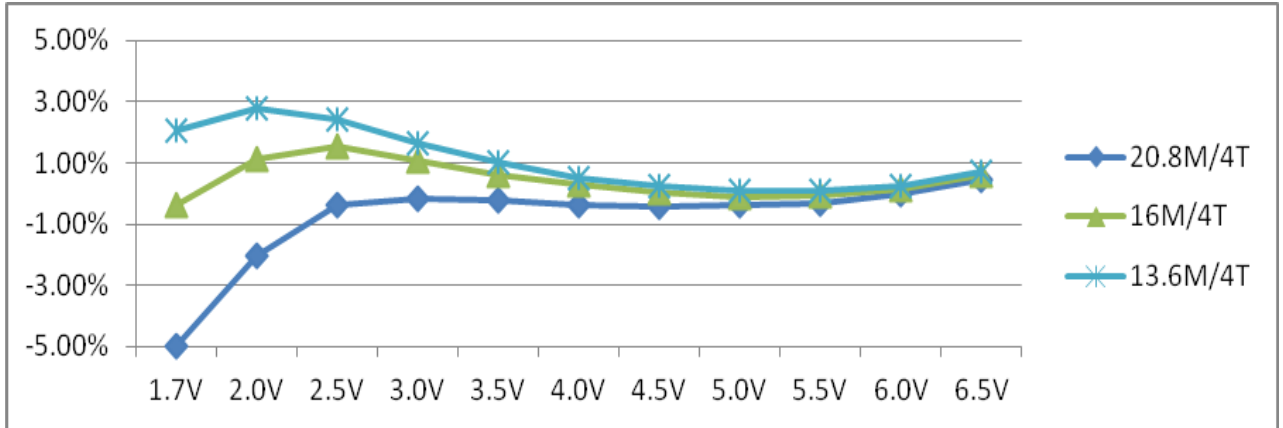
6.5 ADC电气特性

($V_{DD}=5V$, $V_{SS}=0V$, $T_A=25^\circ C$ 除其他指定说明外)

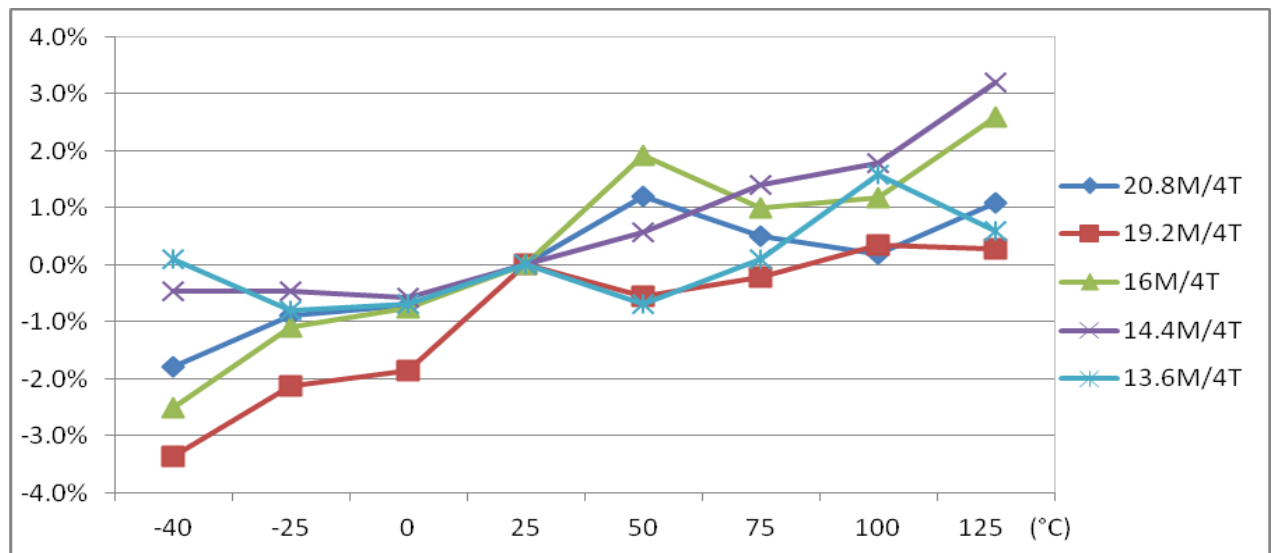
符号	参数	最小值	典型值	最大值	单位	条件
V_{REFH}	模拟参考电压范围	2V	--	V_{DD}	V	由引脚PA0 输入
V_{REF4}	内部参考电压 4V, $V_{DD}=5V$	3.96	4	4.04	V	
V_{REF3}	内部参考电压 3V, $V_{DD}=5V$	2.97	3	3.03	V	
V_{REF2}	内部参考电压 2V, $V_{DD}=5V$	1.98	2	2.02	V	
V_{REF}	内部参考电压 V_{DD} , $V_{DD}=5V$	--	V_{DD}	--	V	
--	内部参考电压	$V_{REF}+0.5$	--	--	V	最小电压
--	ADC 模拟输入电压	0	--	V_{REFH}	V	
$I_{OP(ADC)}$	ADC电流消耗	--	0.5	--	mA	
ADCLK	ADC 时钟频率	32K	--	1M	Hz	
ADCYCLE	ADC转换时间周期	16	--		1/ADCLK	SHCLK=2 ADC clock
ADC _{sample}	ADC 转换率	--	--	125	K/sec	$V_{DD}=5V$
DNL	非线性微分误差	± 1	--	--	LSB	$V_{DD}=5.0V$, $AV_{REFH}=5V$, $FADSMP=62.5K$
INL	非线性积分误差	± 2	--	--	LSB	
NMC	无缺码分辨率	10	11	12	Bits	

6.6 特性图

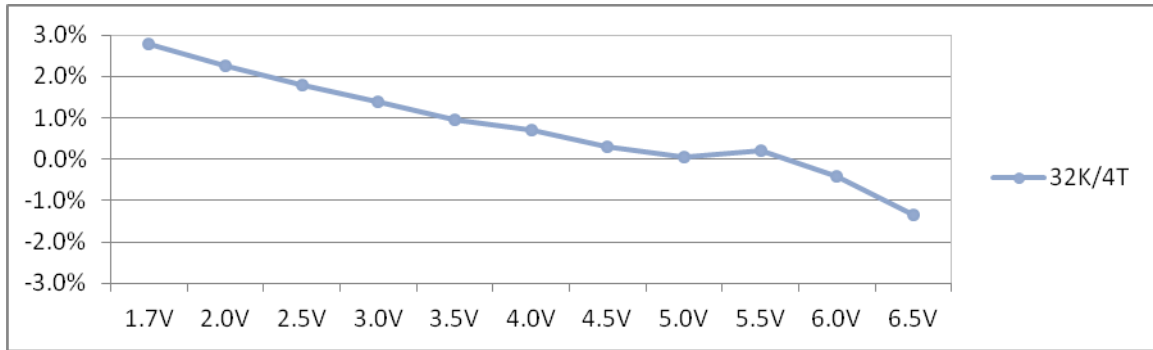
6.6.1 高速 RC 振荡频率 (I_HRC)与电源电压(VDD)曲线图



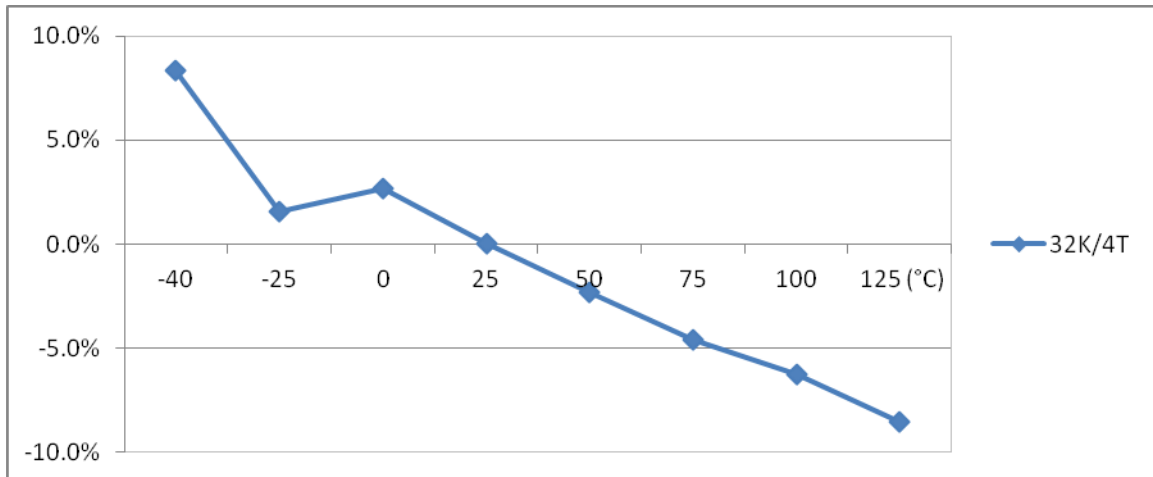
6.6.2 高速 RC 振荡频率 (I_HRC)与温度曲线图



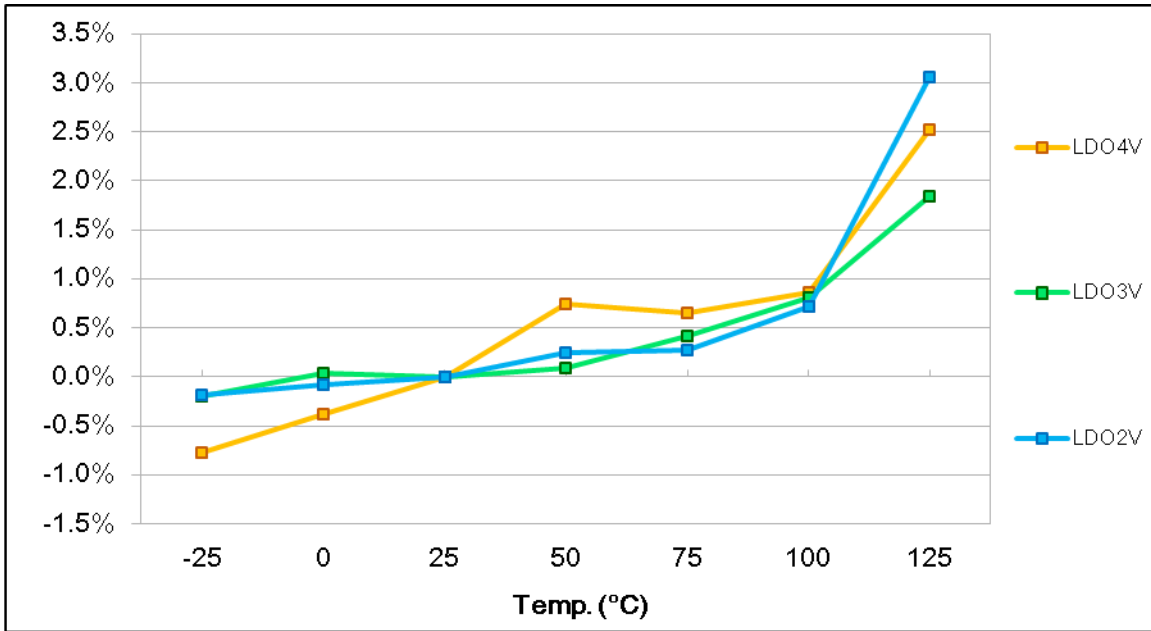
6.6.3 低速 RC 振荡频率(I_LRC)与电源电压(VDD)曲线图



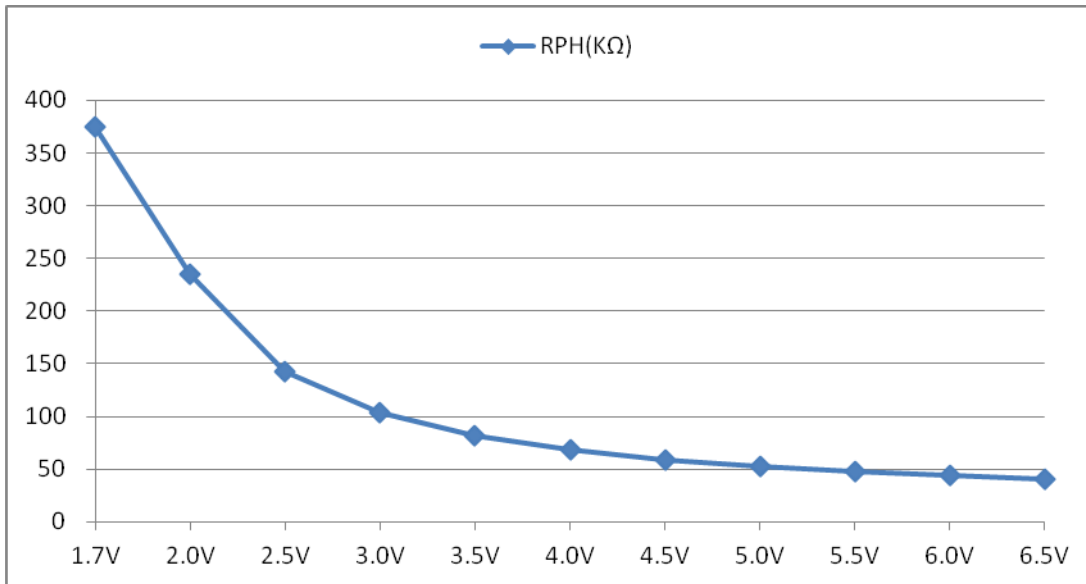
6.6.4 低速 RC 振荡频率(I_LRC)与温度曲线图



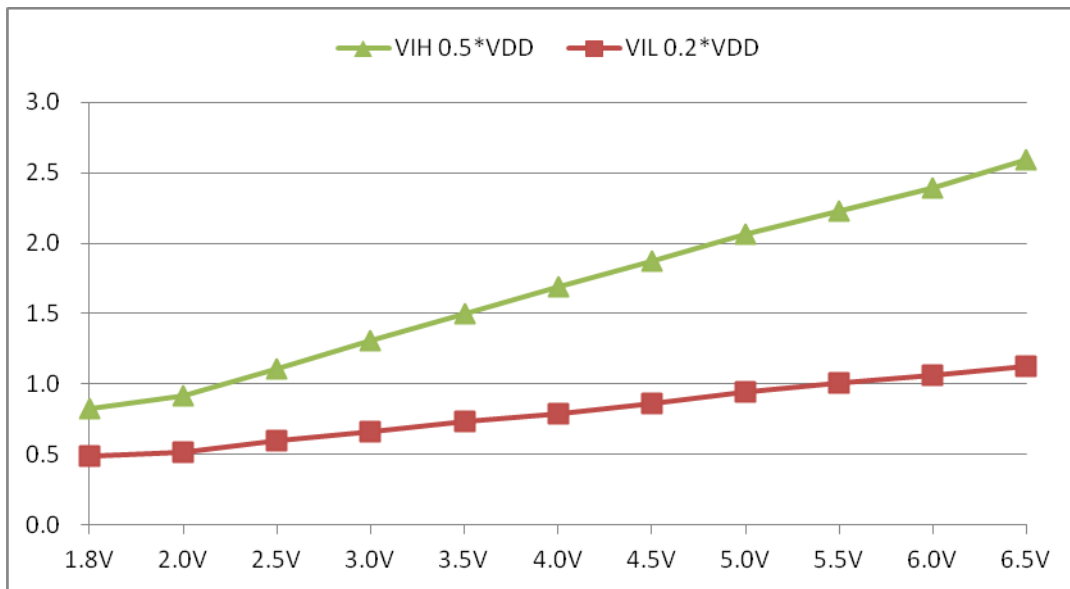
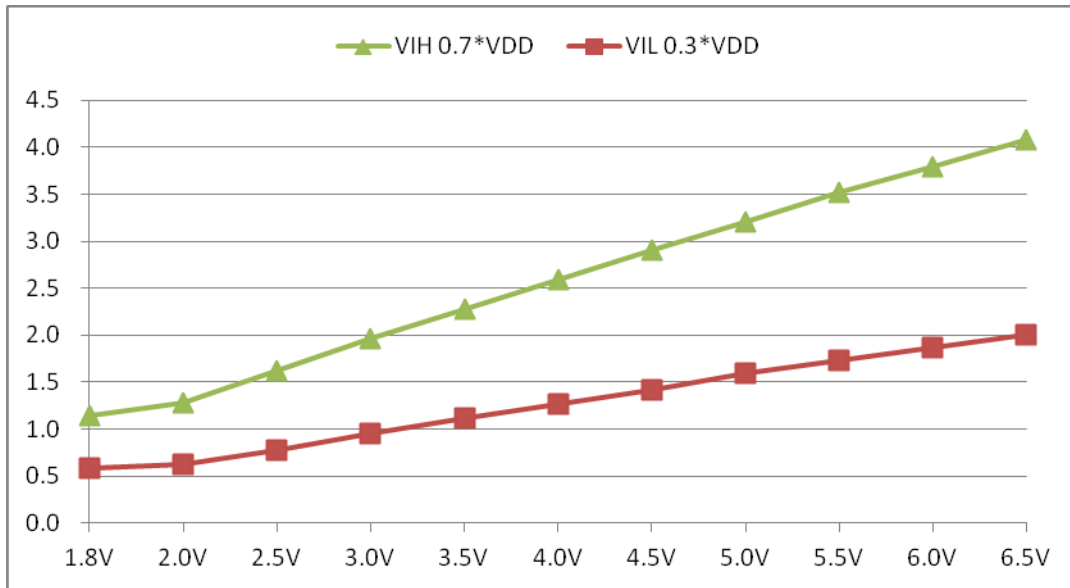
6.6.5 内部参考电压 LDO 与温度曲线图

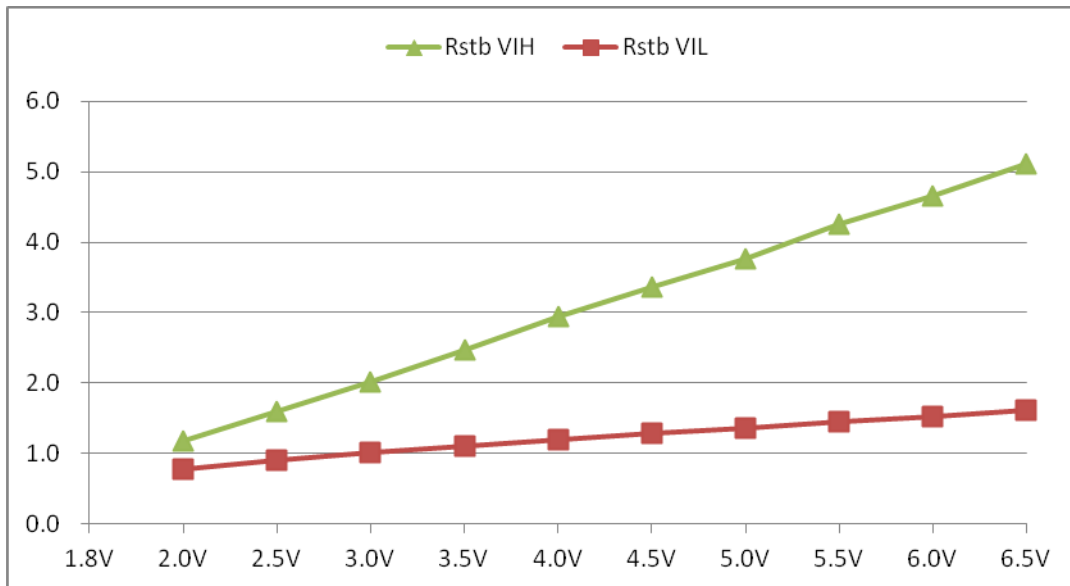
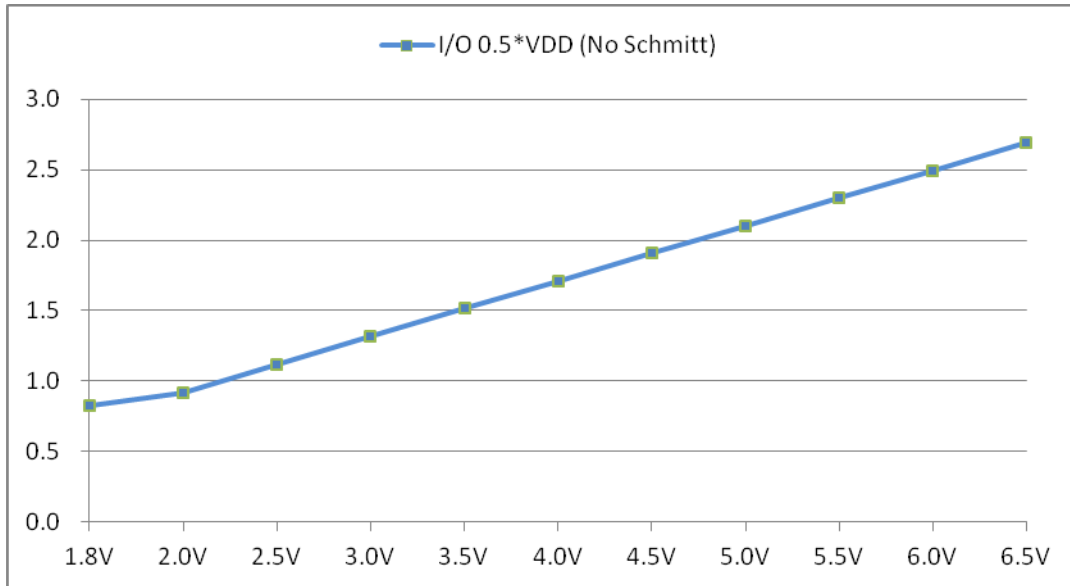


6.6.6 内部上拉电阻与电源电压(VDD)曲线图



6.6.7 VIH/VIL 与电源电压(VDD)曲线图





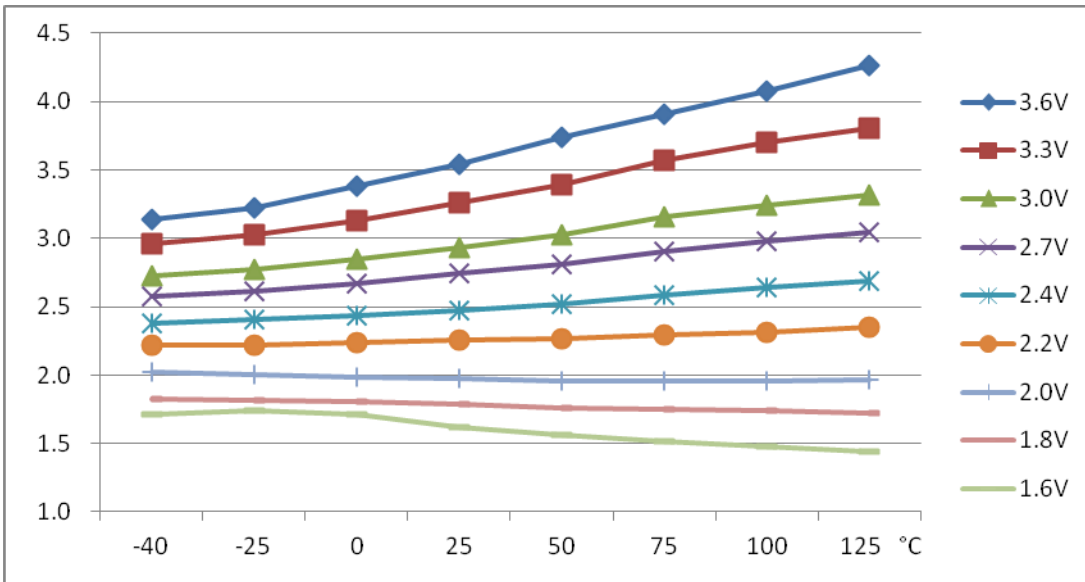
6.7 建议工作电压

建议工作电压（温度范围：-40 °C ~ +85 °C）

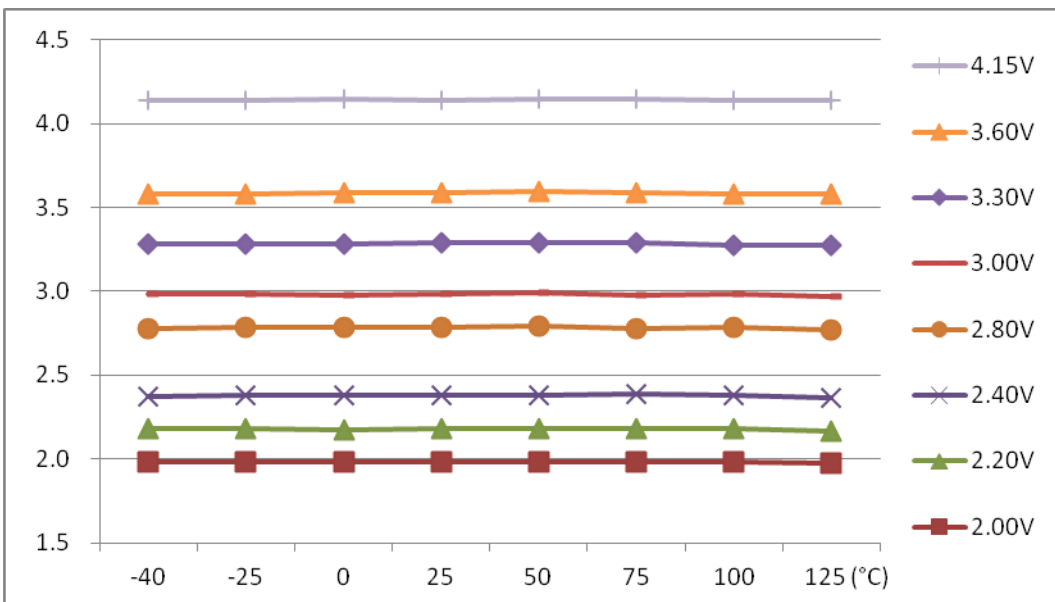
频率	最小电压	最大电压	LVR: 默认值 (25°C)	LVR: 建议值 (-40°C ~ +85°C)
20.8M/2T	3.3V	5.5V	3.6V	3.6V
19.2M/2T	3.0V	5.5V	3.6V	3.6V
16M/2T	2.7V	5.5V	3.0V	3.3V
14.4M/2T	2.7V	5.5V	3.3V	3.3V
13.6M/2T	2.7V	5.5V	3.3V	3.3V
20.8M/4T	2.2V	5.5V	2.4V	2.7V

19.2M/4T	2.2V	5.5V	2.4V	2.7V
14.4M/4T	2.0V	5.5V	2.2V	2.4V
13.6M/4T	2.0V	5.5V	2.2V	2.4V
16M/4T	2.0V	5.5V	2.2V	2.4V
8M/4T	1.8V	5.5V	2.0V	2.2V
E_XT ≦ 6M/2T	1.8V	5.5V	1.8V	2.0V
4M/4T	1.6V	5.5V	1.8V	2.0V

6.8 LVR电压与温度曲线图

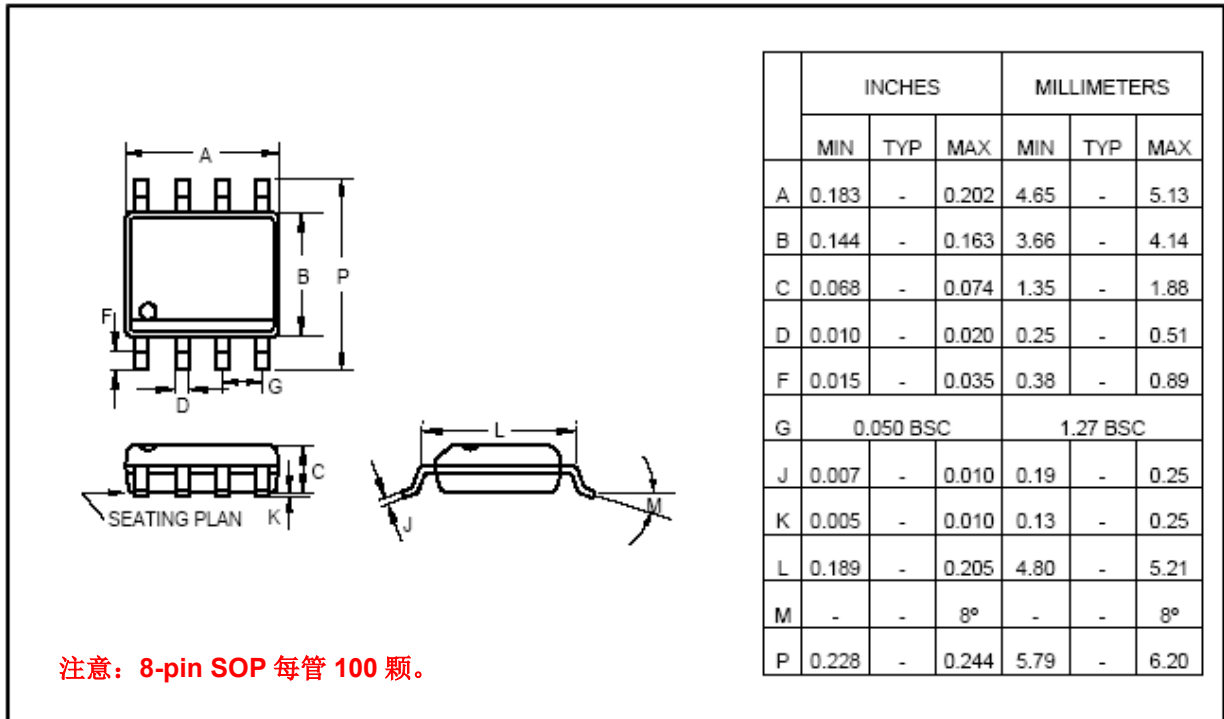


6.9 LVD电压与温度曲线图

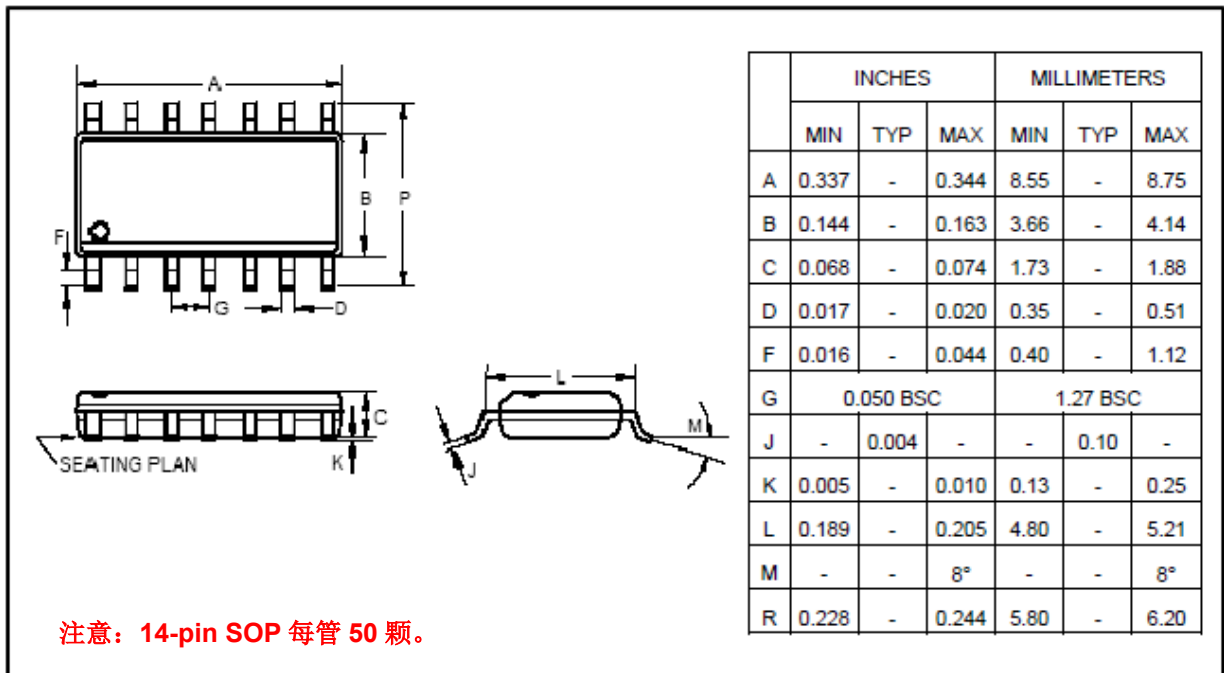


7. 封装尺寸

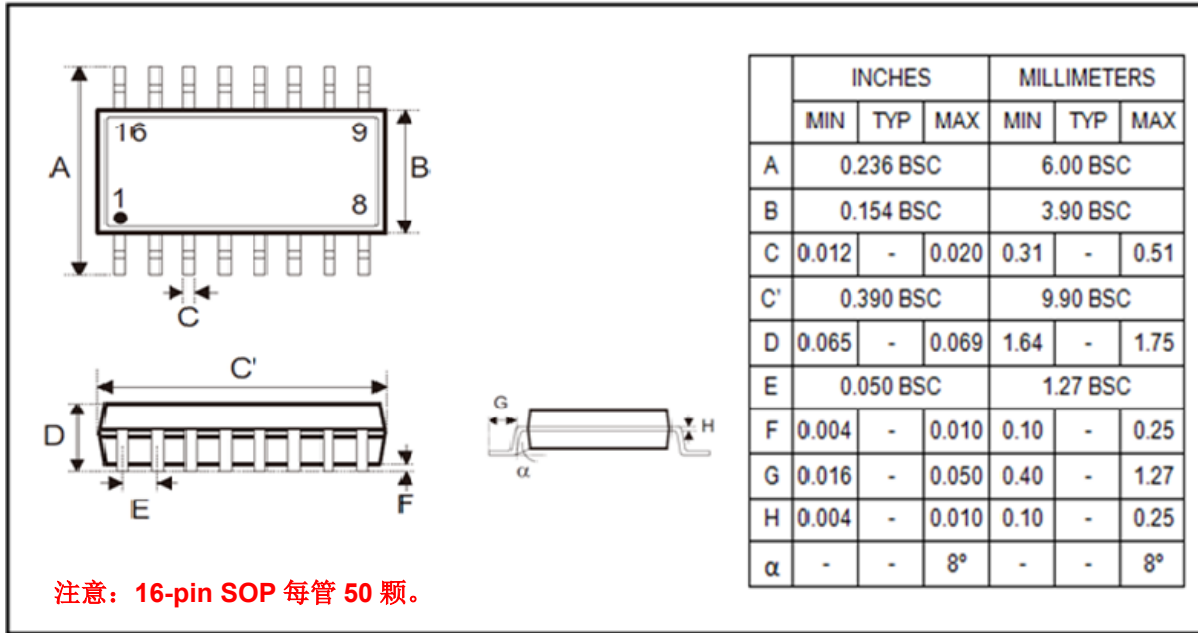
7.1 8 引脚SOP (150 毫寸)



7.2 14 引脚SOP (150 毫寸)



7.3 16 引脚SOP (150 英寸)



8. 订购信息

产品名称	封装类型	引脚数	封装尺寸	配送方式
NY8BE62DS8	SOP	8	150 mil	卷装：每卷 2.5K 颗。 管装：每管 100 颗。
NY8BE62DS14	SOP	14	150 mil	卷装：每卷 2.5K 颗。 管装：每管 50 颗。
NY8BE62DS16	SOP	16	150 mil	卷装：每卷 2.5K 颗。 管装：每管 50 颗。