



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

数
据
手
册

NY8B060D

6 I/O + 12-通道 ADC 8 位 EPROM-Based 单片机

Version 1.1

Aug. 31, 2022

本文内容是由英文规格书翻译, 目的是为了您的阅读更加方便。它无法跟随原稿的更新, 敬请参考英文规格书以获得更准确的信息。

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
1.0	2022/06/30	新发布。	-
1.1	2022/08/31	1. 修正寄存器名称: TM3RH → PWM3RH 2. 修正错误: 3.1.8, 3.1.9, 3.1.20, 3.3.6. 3.4.2	13, 25 17, 18, 22, 27, 29

目 录

1. 概述	7
1.1 功能	7
1.2 NY8B060D 与 NY8B062E、NY8B062D 的主要差异	9
1.3 系统框图	9
1.4 引脚图	10
1.5 引脚说明	10
2. 内存结构	11
2.1 程序存储器	11
2.2 数据存储器	12
3. 功能概述	15
3.1 R-page特殊功能寄存器	15
3.1.1 <i>INDF</i> (间接寻址寄存器)	15
3.1.2 <i>TMR0</i> (定时器0寄存器)	15
3.1.3 <i>PCL</i> (程序计数器低字节)	15
3.1.4 <i>STATUS</i> (状态寄存器)	16
3.1.5 <i>FSR</i> (数据指针寄存器)	16
3.1.6 <i>PortA</i> (<i>PortA</i> 数据寄存器)	17
3.1.7 <i>PortB</i> (<i>PortB</i> 数据寄存器)	17
3.1.8 <i>PCON</i> (<i>Power</i> 寄存器)	17
3.1.9 <i>BWUCON</i> (<i>PortB</i> 唤醒控制寄存器)	18
3.1.10 <i>PCHBUF</i> (程序计数器高字节)	18
3.1.11 <i>ABPLCON</i> (<i>PortA/PortB</i> 下拉电阻控制寄存器)	18
3.1.12 <i>BPHCON</i> (<i>PortB</i> 上拉电阻控制寄存器)	19
3.1.13 <i>INTE</i> (中断使能寄存器)	19
3.1.14 <i>INTF</i> (中断标志寄存器)	20
3.1.15 <i>ADMD</i> (<i>ADC</i> 模式寄存器)	20
3.1.16 <i>ADR</i> (<i>ADC</i> 时钟, <i>ADC</i> 中断标志位与 <i>ADC</i> 转换结果低四位数据寄存器)	21
3.1.17 <i>ADD</i> (<i>ADC</i> 转换结果高八位数据寄存器)	21
3.1.18 <i>ADVREFH</i> (<i>ADC</i> 参考电压寄存器)	22
3.1.19 <i>ADCR</i> (<i>ADC</i> 采样时间与 <i>ADC</i> 位数寄存器)	22
3.1.20 <i>AWUCON</i> (<i>PortA</i> 唤醒控制寄存器)	22
3.1.21 <i>PACON</i> (<i>ADC</i> 引脚数模控制寄存器)	23

3.1.22	ADJMD (ADC 调校模式)	23
3.1.23	INTEDG (外部中断控制寄存器)	23
3.1.24	TMRH (定时器 1 高字节寄存器)	24
3.1.25	ANAEN (比较器使能寄存器)	24
3.1.26	RFC (电阻频率转换控制寄存器)	24
3.1.27	PWM3RH (PWM3 DUTY高字节寄存器)	25
3.2	T0MD (定时器 0 控制寄存器)	25
3.3	F-page特殊功能寄存器	26
3.3.1	IOSTA (PortA I/O控制寄存器)	26
3.3.2	IOSTB (PortB I/O控制寄存器)	26
3.3.3	APHCON (PortA上拉电阻控制寄存器)	27
3.3.4	PS0CV (预分频器 0 寄存器)	27
3.3.5	BODCON (PortB开漏控制寄存器)	27
3.3.6	CMPCR (比较器控制寄存器)	27
3.3.7	PCON1 (Power控制寄存器 1)	28
3.4	S-page特殊功能寄存器	29
3.4.1	TMR1 (定时器 1 寄存器)	29
3.4.2	T1CR1 (定时器 1 控制寄存器 1)	29
3.4.3	T1CR2 (定时器 1 控制寄存器 2)	30
3.4.4	PWM1DUTY (PWM1 占空比寄存器)	31
3.4.5	PS1CV (预分频器 1 寄存器)	31
3.4.6	BZ1CR (蜂鸣器 1 控制寄存器)	31
3.4.7	IRCR (IR控制寄存器)	32
3.4.8	TBHP (表格指针高字节寄存器)	33
3.4.9	TBHD (表格数据高字节寄存器)	33
3.4.10	P2CR1 (PWM2 控制寄存器 1)	33
3.4.11	PWM2DUTY (PWM2 占空比寄存器)	34
3.4.12	OSCCR (振荡器控制寄存器)	34
3.4.13	P3CR1 (PWM3 控制寄存器 1)	35
3.4.14	PWM3DUTY (PWM3 占空比寄存器)	35
3.5	I/O Port	35
3.5.1	IO引脚结构框图	37
3.6	定时器 0	43
3.7	定时器 1/PWM1/Buzzer1	44
3.8	PWM2	46

3.9	PWM3	47
3.10	电阻/频率转换器模式 (RFC)	47
3.11	IR Carrier	48
3.12	低电压侦测 (LVD)	48
3.13	电压比较器	50
3.14	ADC模数转换器	52
3.14.1	ADC参考电压	53
3.14.2	ADC模拟输入通道	54
3.14.3	ADC时钟 (ADCLK), 采样时钟 (SHCLK) 与位数选择	54
3.14.4	ADC操作顺序	55
3.15	看门狗定时器 (WDT)	55
3.16	中断	56
3.16.1	Timer0 上溢中断	56
3.16.2	Timer1 下溢中断	56
3.16.3	看门狗超时中断	57
3.16.4	PA/PB输入状态改变中断	57
3.16.5	外部中断1输入	57
3.16.6	低电压侦测中断	57
3.16.7	ADC模数转换完成中断	57
3.17	振荡器配置	57
3.18	工作模式	58
3.18.1	正常模式	59
3.18.2	慢速模式	60
3.18.3	待机模式	60
3.18.4	睡眠模式	60
3.18.5	唤醒等待时间	61
3.18.6	工作模式概述	61
3.19	复位	61
4.	指令设置	64
5.	配置字节表	80
6.	电气特性	81
6.1	最大绝对值	81
6.2	直流电气特性	81

6.3	OSC 特性	83
6.4	比较器/LVD电气特性	83
6.5	ADC电气特性	83
6.6	特性图	84
6.6.1	高速RC振荡频率与电源电压曲线图	84
6.6.2	高速RC振荡频率与温度曲线图	84
6.6.3	内部线性稳压器与电源电压曲线图	85
6.6.4	内部线性稳压器与温度曲线图	85
6.6.5	内部上拉电阻与电源电压曲线图	86
6.6.6	内部上拉电阻与温度曲线图	86
6.6.7	VIH/VIL与电源电压曲线图	87
6.6.8	VIH/VIL与温度曲线图	88
6.7	建议工作电压	89
6.8	LVR与温度曲线图	89
6.9	LVD与温度曲线图	90
6.10	LDO与温度曲线图	90
7.	芯片脚位坐标图	91
8.	封装尺寸	92
8.1	8 引脚SOP (150 毫寸)	92
9.	订购信息	92

1. 概述

NY8B060D是以EPROM作为存储器的8位单片机，专为家电或量测等等的I/O应用设计。采用CMOS制程并同时提供客户低成本、高性能、及高性价比等显著优势。NY8B060D核心建立在RISC精简指令集架构可以很容易地做编程和控制，共有55条指令。除了少数指令需要两个指令时钟，大多数指令都是一个指令时钟能完成，可以让用户轻松地以过程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。

NY8B060D内建高精度五加二通道12位ADC模数转换器，与高精度电压比较器，足以应付各种模拟接口的侦测与量测。

在I/O的资源方面，NY8B060D有6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脚位都能通过控制相关的寄存器达成如上拉或下拉电阻或开漏（Open-Drain）输出。此外针对红外线遥控的产品方面，NY8B060D内置了可选择频率的红外载波发射口。

NY8B060D有二组定时器，可用系统时钟当作一般的计时应用或者从外部讯号触发来计数。另外NY8B060D提供3组10位的PWM输出，1组蜂鸣器输出，用来驱动马达、LED、或蜂鸣器等等。

NY8B060D采用双时钟机制，高速振荡时钟或者低速振荡时钟都由内部RC振荡或外部晶振输入。在双时钟机制下，NY8B060D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。

在省电的模式下，如待机模式（Standby mode）与睡眠模式（Halt mode）中，有多个中断源可以触发来唤醒NY8B060D进入正常模式（Normal mode）或慢速模式（Slow mode）来处理突发事件。

1.1 功能

- 宽广的工作电压：
 - 2.0V ~ 5.5V @系统时钟 \leq 8MHz。
 - 2.2V ~ 5.5V @系统时钟 $>$ 8MHz。
- 宽广的工作温度：-40°C ~ 85°C。
- 1Kx14位的程序存储器空间（EPROM）。
- 64字节的通用数据存储器空间（SRAM）。
- 6根可分别单独控制输入输出方向的I/O脚（GPIO）、PA[5,4,2]、PB[3,2,1]。
- PA[5,4,2]及PB[3,2,1]可选择输入时使用内部下拉电阻。
- PA[5,4,2]及PB[3,2,1]可选择输入时使用上拉电阻。
- PB[3,2,1]可选择开漏输出（Open-Drain）。
- PA[5]可选择当作输入或开漏输出（Open-Drain）。
- 所有I/O脚输出可选择一般灌电流（Normal Sink Current）或大灌电流（Large Sink Current）。
- 8级深度硬件堆栈（Stack）。
- 存取数据有直接或间接寻址模式。
- 一组8位上数定时器（Timer0）包含可编程的预分频器。

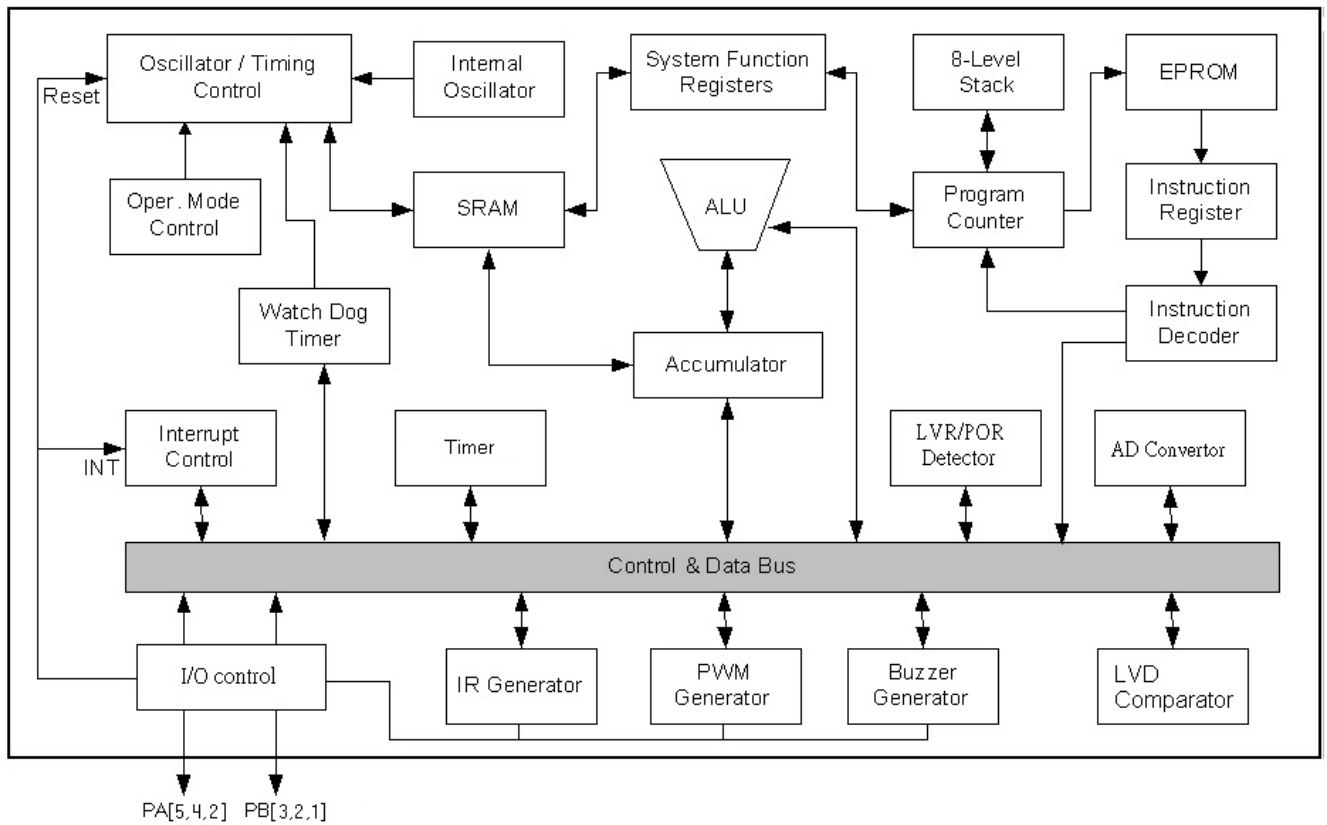
- 一组 10 位下数定时器 (Timer1) 可选自动重载或连续下数计时。
- 三个 10 位的脉冲宽度调变输出 (PWM1、2、3)。
- 一个蜂鸣器输出 (BZ1)。
- 38/57KHz红外线载波 (IR) 频率可供选择, 同时载波之极性也可以通过寄存器选择。
- 内置准确的低电压侦测电路 (LVD)。
- 内置五加二通道 12 位ADC模数转换器 (Analog to Digital Converter)。
- 内置精准的电压比较器 (Voltage Comparator)。
- 内置上电复位电路 (POR)。
- 内置低压复位功能 (LVR)。
- 内置看门狗计时 (WDT), 可由配置字节 (Configuration Word) 控制开/关。
- 内置电阻/频率转换器 (RFC) 功能。
- 双时钟机制, 系统时钟可以随时切换高速振荡或者低速振荡。
 - 高速振荡时钟: I_HRC (1~20MHz内部高速RC振荡)
 - 低速振荡时钟: 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 mode)、慢速模式 (Slow mode)、待机模式 (Standby mode) 与睡眠模式 (Halt mode)。
- 七种硬件中断:
 - Timer0 上溢中断。
 - Timer1 下溢中断。
 - WDT中断。
 - PA/PB输入状态改变中断。
 - 一组外部中断。
 - 低电压侦测中断。
 - ADC模数转换完成中断。
- NY8B060D在待机模式 (Standby mode) 下的七种唤醒中断:
 - Timer0 上溢中断。
 - Timer1 下溢中断。
 - WDT中断。
 - PA/PB输入状态改变中断。
 - 一组外部中断。
 - 低电压侦测中断。
 - ADC模数转换完成中断。
- NY8B060D在睡眠模式 (Halt mode) 下的三种唤醒中断:
 - WDT中断。
 - PA/PB输入状态改变中断。
 - 一组外部中断。

1.2 NY8B060D 与 NY8B062E、NY8B062D 的主要差异

项目	功能	NY8B060D	NY8B062E	NY8B062D
1	ADC 偏移校准*	是	是	---
2	ADC 电力损耗	500uA @5V	500uA @5V	3mA @5V
4	I/O 输入施密特触发	开启/关闭	开/关闭	---
5	比较器	与 LVD 共享	轨对轨	轨对轨

*注: 需要增加 ADC 零点校准初始化程式 (可参考 NYIDE 范例程式)

1.3 系统框图



1.4 引脚图

NY8B060D提供一种封装类型：SOP8。

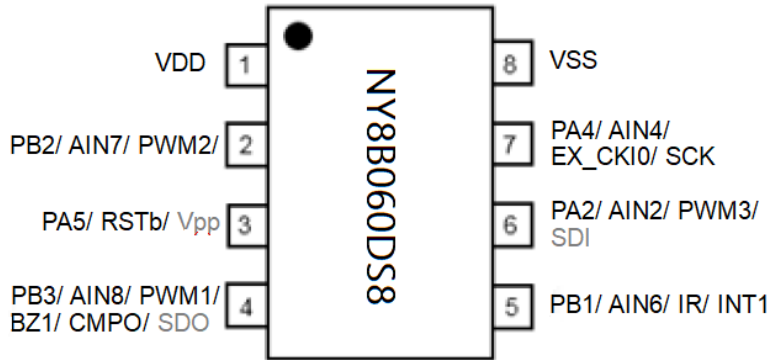


图 1 封装引脚图

1.5 引脚说明

引脚名	I/O	描述
PA2 AIN2 PWM3 SDI	I/O	PA2 是一个双向I/O引脚，也可当作比较器输入引脚。 AIN2 为ADC的模拟输入引脚。 PA2 可输出PWM3。 PA2 也是编程数据输入SDI。
PA4 AIN4 EX_CKIO SCK	I/O	PA4 是一个双向I/O引脚。 AIN4 为ADC的模拟输入引脚。 PA4 可当作定时器 0 / 1 外部时钟来源EX_CKIO。 PA4 也是编程时钟输入SCK。
PA5 RSTb Vpp	I/O	PA5 可选择当作输入脚或开漏输出脚。 PA5 可当作复位引脚RSTb。 如果该引脚的电压超过 7.75V，IC会进入EPROM编程模式。
PB1 AIN6 IR INT1	I/O	PB1 是一个双向I/O引脚。 AIN6 为ADC的模拟输入引脚。 如果启用红外模式，该引脚为红外载波输出。 PB1 也可以当成外部中断 1 的输入引脚INT1。 另外，该引脚也能作为ADC外部高参考电压源。
PB2 AIN7 PWM2 F _{INST} OUT	I/O	PB2 是一个双向I/O引脚。 AIN7 为ADC的模拟输入引脚。 PB2 可输出PWM2。 PB2 也可作为指令时钟源。
PB3 AIN8 PWM1/ BZ1/ CMPO SDO	I/O	PB3 是一个双向I/O引脚。 AIN8 为ADC的模拟输入引脚。 PB3 可输出BZ1，PWM1 或当作比较器输出引脚CMPO。 PB3 也是编程数据输出SDO。
VDD	-	电源正端。
VSS	-	电源负端。

2. 内存结构

NY8B060D存储器分为两类：分别是程序存储器和数据存储器。

2.1 程序存储器

NY8B060D程序存储器空间是 1Kx14 位。因此，10 位宽的程序计数器（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。

NY8B060D提供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。

NY8B060D程序存储器地址 0x3FE~0x3FF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。

NY8A060D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。

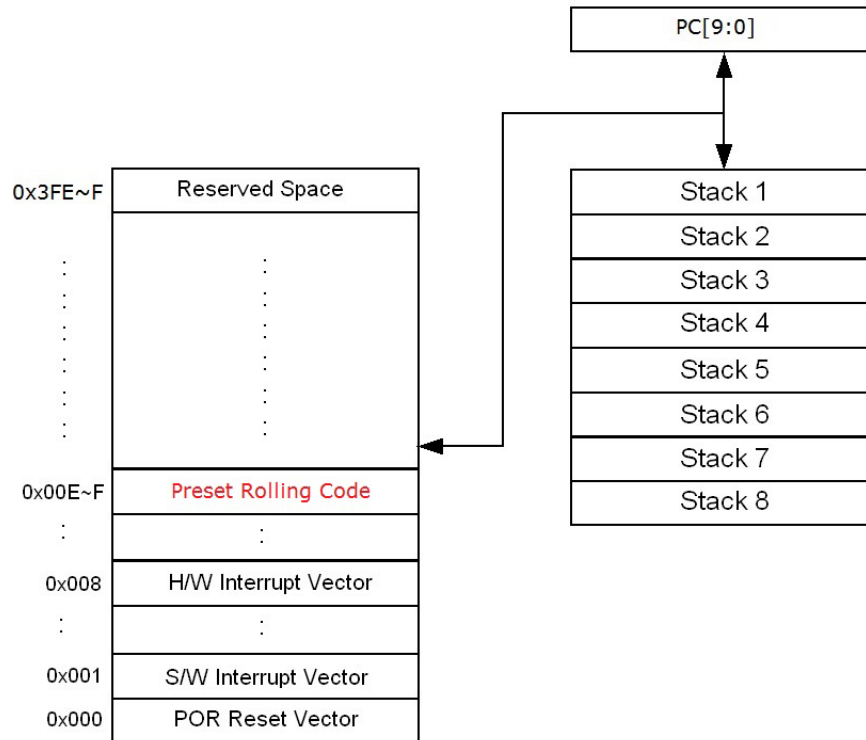


图 2 程序存储器对应地址

2.2 数据存储

根据用于存取数据存储器的指令，数据存储器可分为三类：**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。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，可通过数据指针寄存器（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。

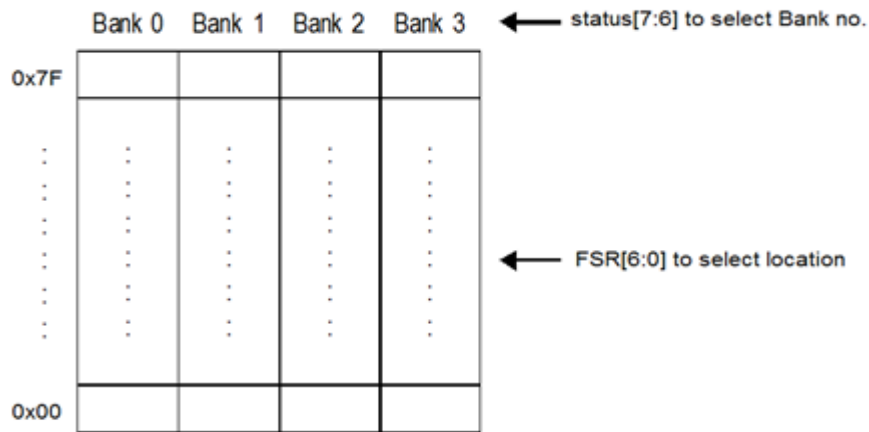


图3 间接寻址方式存取数据存储器

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。

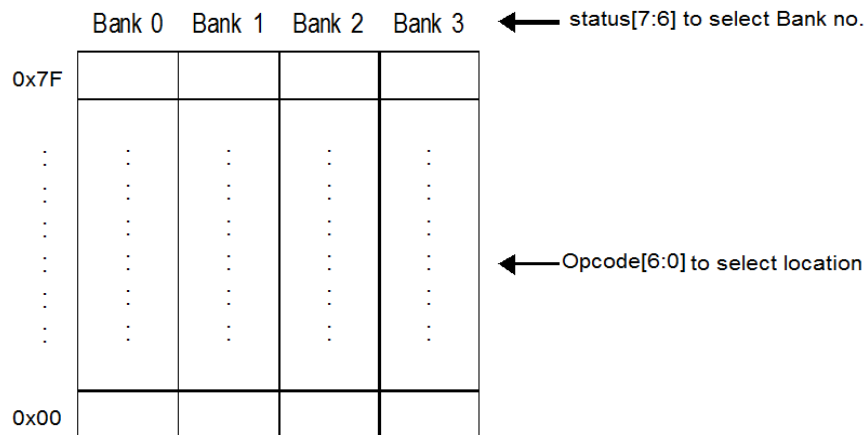


图4 直接寻址方式存取数据存储器

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0x5F，如表1所示。

NY8B060D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。

地址 \ Status [7:6]	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				映射至Bank 0		
0x1C	PWM3RH						
0x1D ~ 0x1E	-				-	-	-
0x1F	-				-	-	-
0x20 ~ 0x5F	通用寄存器				映射至bank 0	映射至bank 0	映射至bank 1

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。当F-page和S-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的地址说明如下表。

地址	特殊功能寄存器种类	F-page SFR	S-page SFR
0x0		-	TMR1
0x1		-	T1CR1
0x2		-	T1CR2
0x3		-	PWM1DUTY
0x4		-	PS1CV
0x5		IOSTA	BZ1CR
0x6		IOSTB	IRCR
0x7		-	TBHP
0x8		-	TBHD
0x9		APHCON	-
0xA		PS0CV	P2CR1
0xB		-	-
0xC		BODCON	PWM2DUTY
0xD		-	-
0xE		CMPCR	-
0xF		PCON1	OSCCR
0x10		-	-
0x11		-	P3CR1
0x12		-	-
0x13		-	PWM3DUTY
0x14		-	-
0x15		-	-

表 2 F-page特殊功能寄存器和S-page特殊功能寄存器地址表

3. 功能概述

本章节将详细描述NY8B060D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（PCL）是一个 10 位寄存器，分高 2 位和低 8 位。当程序执行了一个指令，同时PC数值增 1，除了某些指令会直接更改PC数值。PC高字节（PC[9:8]）并不能直接存取，必须藉由PCHBUF寄存器完成存取。LGOTO指令的PC[9:0]是从指令码取得。

LCALL指令的PC[9:0]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]	-	/TO	/PD	Z	DC	C
读写属性			读/写	读/写	-	读/写(*2)	读/写(*1)	读/写	读/写	读/写
初始值			0	0	X	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。

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	-	FSR[6:0]						
读写属性			-	读/写						
初始值			X	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。

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	-	-	PA5	PA4	-	PA2	-	-
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(PA[5,4,2])							

读取PortA时，若特定脚位被配置为输入脚，将得到该脚位输入状态。然而，若该脚位被配置为输出脚，依据配置字节选项RD_OPT，得到该脚位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	-	-	-	-	PB3	PB2	PB1	-
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(PB3~PB1)							

读取PortB时，若特定脚位被配置为输入脚，将得到该脚位输入状态。然而，若该脚位被配置为输出脚，依据配置字节选项RD_OPT，得到该脚位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	-	-	-
读写属性			读/写							
初始值			1	1	0	1	1	X	X	X

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，开启WDT。

WDTEN=0，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	-	-	-	-	WUPB3	WUPB2	WUPB1	-
读写属性			-	-	-	-	读/写	读/写	读/写	-
初始值			X	X	X	X	0	0	0	X

WUPBx: 开启/关闭PBx唤醒功能, $1 \leq x \leq 3$ 。

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	-	-	-	-	-	-	PCHBUF[1:0]	
读写属性			-	-	-	-	-	-	W	
初始值			X	X	X	X	X	X	00	

PCHBUF[1:0]: 程序计数器PC的第9位到第8位。

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	-	-	/PLPA2	-	-
读写属性			读/写							
初始值			1	1	1	X	X	1	X	X

/PLPAx: 关闭/开启PA2 下拉电阻。

/PLPAx=1, 关闭PA2 下拉电阻。

/PLPAx=0, 开启PA2 下拉电阻。

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

/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	-	-	-	-	/PHPB3	/PHPB2	/PHPB1	-
读写属性			-	-	-	-	读/写	读/写	读/写	-
初始值			X	X	X	X	1	1	1	X

/PHPBx: 关闭/开启PBx上拉电阻, $1 \leq x \leq 3$ 。

/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	-	LVDIE	T1IE	-	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	X	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。

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。

LVDIE: 低电压侦测中断使能位。

LVDIE=1 时, 开启低电压侦测中断。

LVDIE=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	-	PABIF	T0IF
读写属性			读/写	读/写	-	读/写	读/写	-	读/写	读/写
初始值（Note）			0	0	X	0	0	X	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。

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。

0010: 选择PA2 引脚为ADC模拟输入通道。

0100: 选择PA4 引脚为ADC模拟输入通道。

0110: 选择PB1 引脚为ADC模拟输入通道。

0111: 选择PB2 引脚为ADC模拟输入通道。

1000: 选择PB3 引脚为ADC模拟输入通道。

1011: 选择内部 1/4 VDD为ADC模拟输入通道。

1100: 选择GND为ADC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= $F_{INST}/16$, 01: ADC时钟= $F_{INST}/8$, 10: ADC时钟= $F_{INST}/1$, 11: ADC时钟= $F_{INST}/2$ 。

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			0	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。

注意: EVHENB必须为0。

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	-	-	-	PBCON3	SHCK1	SHCK0	ADCR1	ADCR0
读写属性			-	-	-	读/写	读/写	读/写	读/写	读/写
初始值			X	X	X	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。

PBCON3: PB3 模拟引脚选择位。

0=PB3 作为ADC模拟输入引脚或数字IO引脚。

1=PB3 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	-	-	WUPA5	WUPA4	-	WUPA2	-	-
读写属性			-	-	读/写	读/写	-	读/写	-	-
初始值			X	X	0	0	X	0	X	X

WUPAx: 开启/关闭PAx唤醒功能, x =2, 4, 5。

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	-	PACON4	-	PACON2	-	-
读写属性			读/写	读/写	-	读/写	-	读/写	-	-
初始值			0	0	X	0	X	0	X	X

PACONx: PA引脚选择位, $x = 2, 4$ 。

0=PAx 作为ADC模拟输入引脚或数字IO引脚。

1=PAx 仅作为ADC模拟输入引脚。

PBCONx: PB模拟引脚选择位, $1 \leq x \leq 2$ 。

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 调校模式)

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]: 调校选择位, $0 \leq x \leq 4$ 。

00000 = 偏移 0 mV

11111 = 偏移 12.5 mV

ADJ_SIGN: 调校标志位。

0 = ADC数据递减。

1 = ADC数据递增。

注意: 应用时, 请参考NYIDE范例程序“ADC Interrupt_Auto Calibration”。

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	-	-	EIS1	-	INT1G1	INT1G0	-	-
读写属性			-	-	读/写	-	读/写	读/写	-	-
初始值			X	X	0	X	0	1	X	X

EIS1: 外部中断 1 引脚选择位。

EIS1=1 时, PB1 选择为外部中断 1 引脚。

EIS1=0 时, PB1 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。

3.1.24 TMRH (定时器 1 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
0010	PA2
0100	PA4

PSEL[3:0]	RFC PAD
0101	PA5
1001	PB1
1010	PB2
1011	PB3

表 3 选择RFC输入引脚

3.1.27 PWM3RH (PWM3 DUTY 高字节寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM3RH	R	0x1C	-	-	-	-	-	-	PWM3 DUTY9	PWM3 DUTY8
读写属性			-	-	-	-	-	-	读/写	读/写
初始值			-	-	-	-	-	-	X	X

PWM3DUTY9~8: PWM3 占空比高 2 位。

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 (Note)			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，上升沿时定时器 0 加一。

T0CE=0 时，下降沿时定时器 0 加一。

注意：T0CE应用在外部 EX_CKIO 脚作为定时器 0 时钟源。

T0CS: 定时器 0 时钟源选择。

T0CS=0 时，选择指令时钟F_{INST}。

注意：T0CS必须一直为 0。

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。

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特殊功能寄存器

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	-	-	IOPA5	IOPA4	-	IOPA2	-	-
读写属性			-	-	读/写	读/写	-	读/写	-	-
初始值			X	X	1	1	X	1	X	X

IOPAx: PAx I/O模式选择，x = 2, 4, 5。

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	-	-	-	-	IOPB3	IOPB2	IOPB1	-
读写属性			-	-	-	-	读/写	读/写	读/写	-
初始值			X	X	X	X	1	1	1	X

IOPBx: PBx I/O模式选择，1 ≤ x ≤ 3。

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	-	-	/PLPA5	/PHPA4	-	/PHPA2	-	-
读写属性			读/写							
初始值			X	X	1	1	X	1	X	X

/PHPAx: 关闭/开启 PAx上拉电阻, x =2, 4。

/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。

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			读							
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	-	-	-	-	ODPB3	ODPB2	ODPB1	-
读写属性			-	-	-	-	读/写	读/写	读/写	-
初始值			X	X	X	X	0	0	0	X

ODPBx: 开启/关闭PBx的开漏, $1 \leq x \leq 3$ 。

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	-	RBIASH	RBIASL	CMPF_INV	PS1	PS0	NS1	NS0
读写属性			读/写							
初始值			X	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	-
01	-
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]	比较器正输入源
00	-
01	PA2
10	Vref
11	-

表 6 比较器正输入源选择

CMPF_INV: 比较器负输出控制位。

CMPF_INV = 1: 比较器负输出。

CMPF_INV = 0: 比较器正输出。

RBIAS_L, RBIAS_H: 设置相应的参考电压。

(请参考 3.13.1 章节)

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	LVDOOUT	LVDS3	LVDS2	LVDS1	LVDS0	-	T0EN
读写属性			读/写(1*)	读	读/写	读/写	读/写	读/写	-	读/写
初始值			0	X	1	1	1	1	X	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]	电压
0000	1.9V
0001	2.0V
0010	2.2V
0011	2.4V
0100	2.6V
0101	2.8V

LVDS[3:0]	电压
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 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。

(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	-	-	TM1_HRC	T1OS	T1RL	T1EN
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 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 计数选项
0	0	定时器 1 从 0x3FF 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

T1_HRC: 定时器 1 时钟源选择。

T1_HRC =1, PWM1, 2, 3 和定时器 1 的时钟源是高速振荡时钟。

T1_HRC =0, PWM1, 2, 3 和定时器 1 的时钟源根据 T1CS 寄存器位决定。

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, PB3 输出 PWM1。

PWM1OEN=0, PB3 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

PS1SEL[2:0]	预分频比选项
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。

T1CE：定时器 1 外部时钟触发沿选项。

T1CE=1 时，EX_CKIO 脚下降沿时定时器 1 减一。

T1CE=0 时，EX_CKIO 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 脚作为外部时钟输入。

T1CS=0 时，选择指令时钟 F_{INST} 。

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							

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			写	-	-	-	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	定时器 1 bit 0
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	-	-	-	-	-	IRCSEL	IRF57K	IREN
读写属性			-	-	-	-	-	写	写	写
初始值			X	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 脚数据是 1 时，IR 载波会被产生。

IRCSEL=1 且 I/O 脚数据是 0 时，IR 载波会被产生。

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. Type	57KHz	38KHz	条件
High IRC (4MHz)	64	96	HIRC 模式（不论系统时钟频率是多少，IR 模块的输入时钟都设定为 4MHz）

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	-	-	-	-	-	-	TBHP1	TBHP0
读写属性			-	-	-	-	-	-	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，程序计数器寄存器会通过 TBHP[2:0] 与 ACC 指向目标地址。ACC 是 PC[9:0] 的低字节，TBHP[1:0] 是 PC[9:0] 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, PB2 输出PWM2。

PWM2OEN=0, PB2 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							

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。

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 输出PWM3。

PWM3OEN=0, PA2 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							

PWM3DUTY[7:0]寄存器用于定义PWM3 的占空比。

3.5 I/O Port

NY8B060D提供 6 个I/O口 (PA[5,4,2]和PB[3:1]), 用户可以由寄存器PORTA和PORTB读写这些脚位。每个I/O脚位都有一个对应的寄存器控制位以定义该脚位是输入或输出口, 寄存器IOSTA[5,4,2]定义PA[5,4,2]为输入或输出口, 寄存器IOSTB[3:1]定义PB[3:1]为输入或输出口。

当一个I/O脚位被配置为输入口, 它可以由寄存器开启或关闭内部上拉/下拉电阻。寄存器APHCON[4,2] 和PCON[4]用于开启或关闭PA[5,4,2]的内部上拉电阻。寄存器APHCON[5], PCON[6] 和 ABPLCON[2]用于开启或关闭PA[5,4,2]的内部下拉电阻。寄存器BPHCON[3:1]则是用于开启或关闭PB[3:1]的内部上拉电阻。ABPLCON[7:5]则是用于开启或关闭PB[3:1]的内部下拉电阻。

当一个I/O脚位被配置为输出口, 可由寄存器开启或关闭开漏。寄存器BODCON[3:1]决定PB[3:1]是否为开漏输出脚。

I/O口功能摘要如下表:

功能		PA[4,2]	PA[5]	PB[3:1]
输入	上拉电阻	V	V	V
	下拉电阻	V	V	X

功能		PA[4,2]	PA[5]	PB[3:1]
输出	开漏	X	V	V

表 13 I/O端口功能摘要

在PA和PB的每个I/O脚都有输入状态改变产生中断功能。寄存器AWUCON[5,4,2]和 BWUCON[3:1]会使能或禁能任一PA和PB脚位的唤醒功能。只要AWUCON和BWUCON对应到的任一PA和PB脚位被置为 1 时，且在此输入脚位有状态改变时，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。

NY8B060D提供 1 个外部中断，当寄存器EIS1 (INTEDG[5]) 设定为 1，PB1 则被当作外部中断 1 的输入脚。

注意：当 PB1 同时设定成输入状态改变触发脚和外部中断脚，外部中断有较高的优先权，而 PB1 输入状态改变触发脚则会被关闭，但 PB3~PB2 输入状态改变触发脚不会被影响。

NY8B060D提供红外线IR载波生成器。当IREN=1 时，PB1 输出 IR 红外线载波，当IREN=0 时，不产生 IR 红外线载波。

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NY8B060D发生复位。

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 (T0MD) 为 1 和LCK_TM0 为 0，PA4 可当作定时器 0 外部时钟源EX_CK10。如果寄存器T1CS 为 1，PA4 可当作定时器 1 外部时钟源EX_CK10。

如果寄存器CMPOE=1，PB3 也可以当作比较器输出引脚。如果寄存器PWM1OEN (T1CR1[7]) 为 1，PB3 也可以当作脉冲宽度调制PWM1 输出。若寄存器BZ1EN (BZ1CR[7]) 为 1，PB3 也可以当作蜂鸣器 1 输出。PB3 复用引脚输出优先级为比较器输出 > PWM1 输出 > 蜂鸣器 1 输出。

若寄存器PWM2OEN (T2CR1[7]) 为 1，PB2 可以当作PWM2 输出。

当IO配置为输出口，每个引脚可在配置字节设置为一般灌电流 (19mA@VDD=3V)，大灌电流 (28mA@VDD=3V)。如下表 21 所示。

配置字节	一般灌电流	大灌电流
PXcurrent	0	1
PXcsc	0	0

表 14 灌电流模式选择 (X=A,B)

3.5.1 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。

VNEN: 开启为比较器负输入引脚。

CMPVP, CMPVN: 比较器正输入与负输入引脚。

RD_TYPE: 选择读取脚位或数据锁存器。

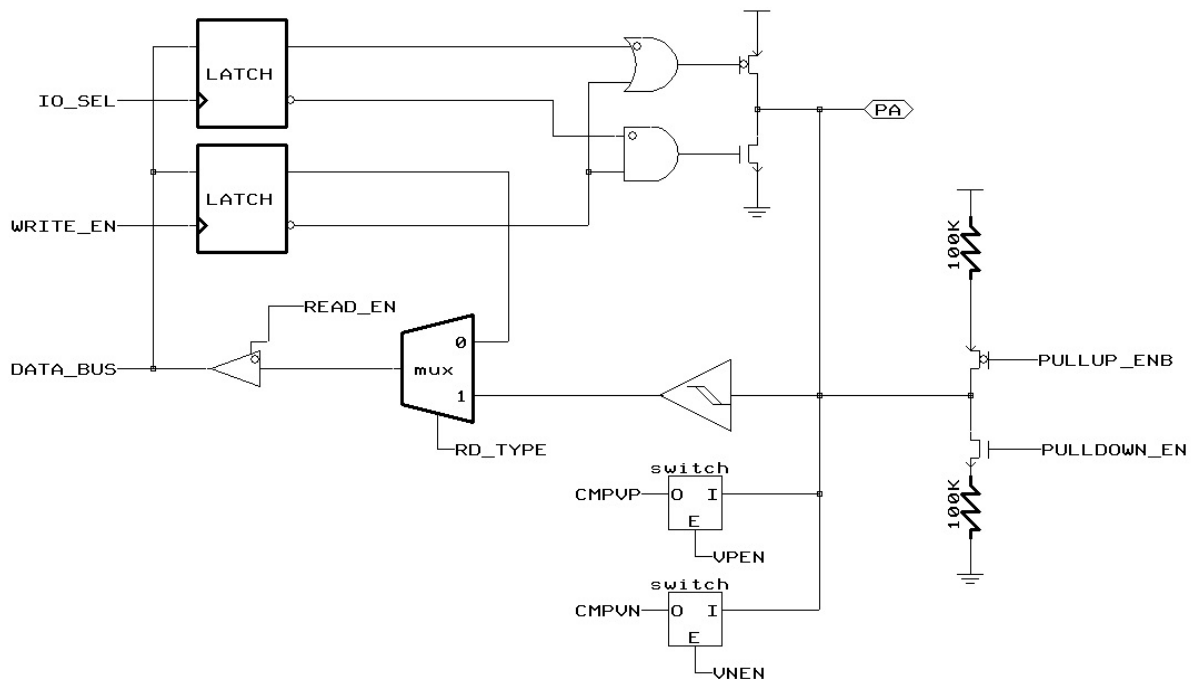


图 5 PA2 引脚结构框图

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。

RD_TYPE: 选择读取脚位或数据锁存器。

EX_CKIO: Timer0,1 外部时钟输入。

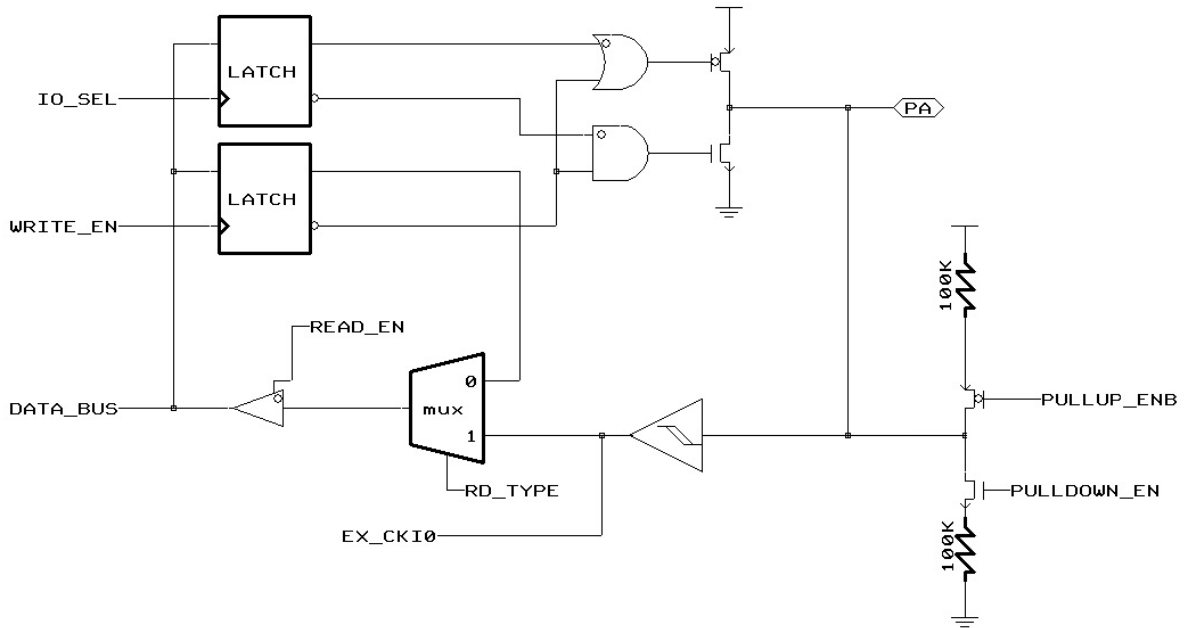


图 6 PA4 引脚结构框图

RSTPAD_EN: 开启外部复位引脚。

RSTB_IN: 复位引脚输入。

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。

RD_TYPE: 选择读取脚位或数据锁存器。

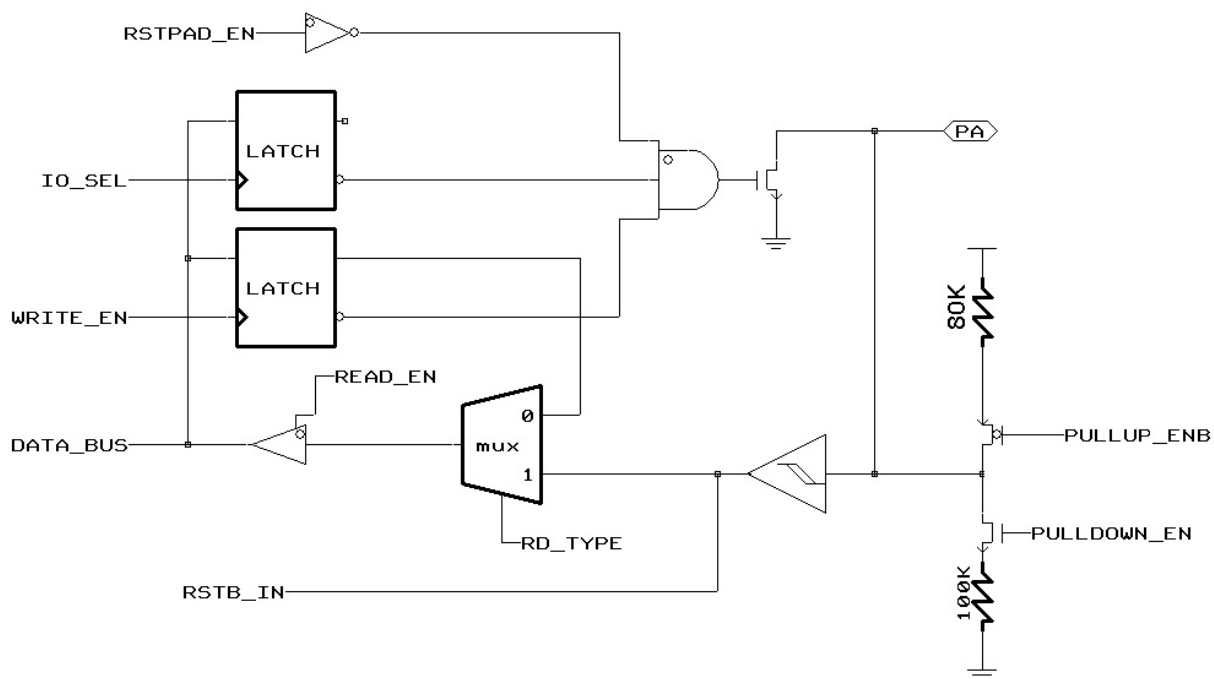


图 7 PA5 引脚结构框图

- 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。
- PULLUP_ENB: 开启内部上拉电阻。
- PULLDOWN_EN: 开启内部下拉电阻。
- RD_TYPE: 选择读取脚位或数据锁存器。
- EIS1: 使能外部中断 1 功能。
- INTEDG[3:2]: 选择外部中断 1 触发沿。
- EX_INT1: 外部中断 1 信号。
- WUB: 使能PB唤醒功能。
- SET_PBIF: PB唤醒标志。

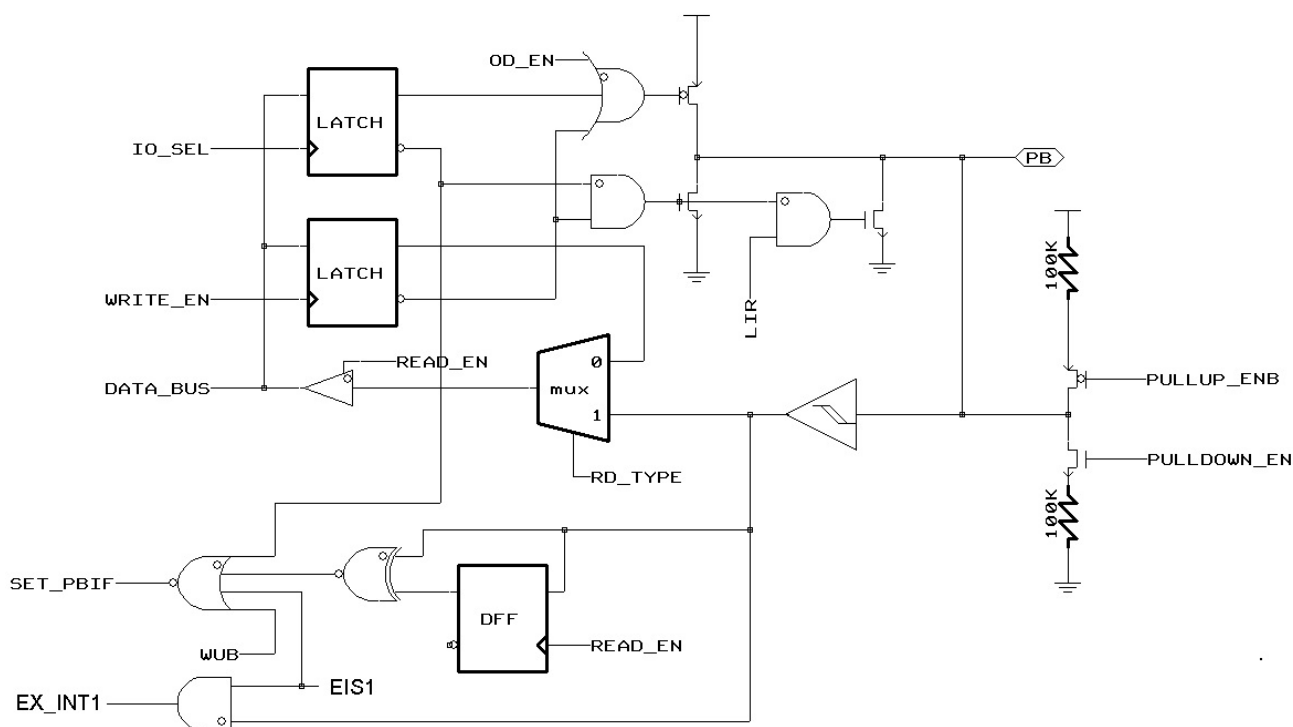


图 8 PB1 引脚结构框图

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。

PULLUP_ENB: 开启内部上拉电阻。

PULLDOWN_EN: 开启内部下拉电阻。

RD_TYPE: 选择读取脚位或数据锁存器。

WUB: 使能PB唤醒功能。

SET_PBIF: PB唤醒标志。

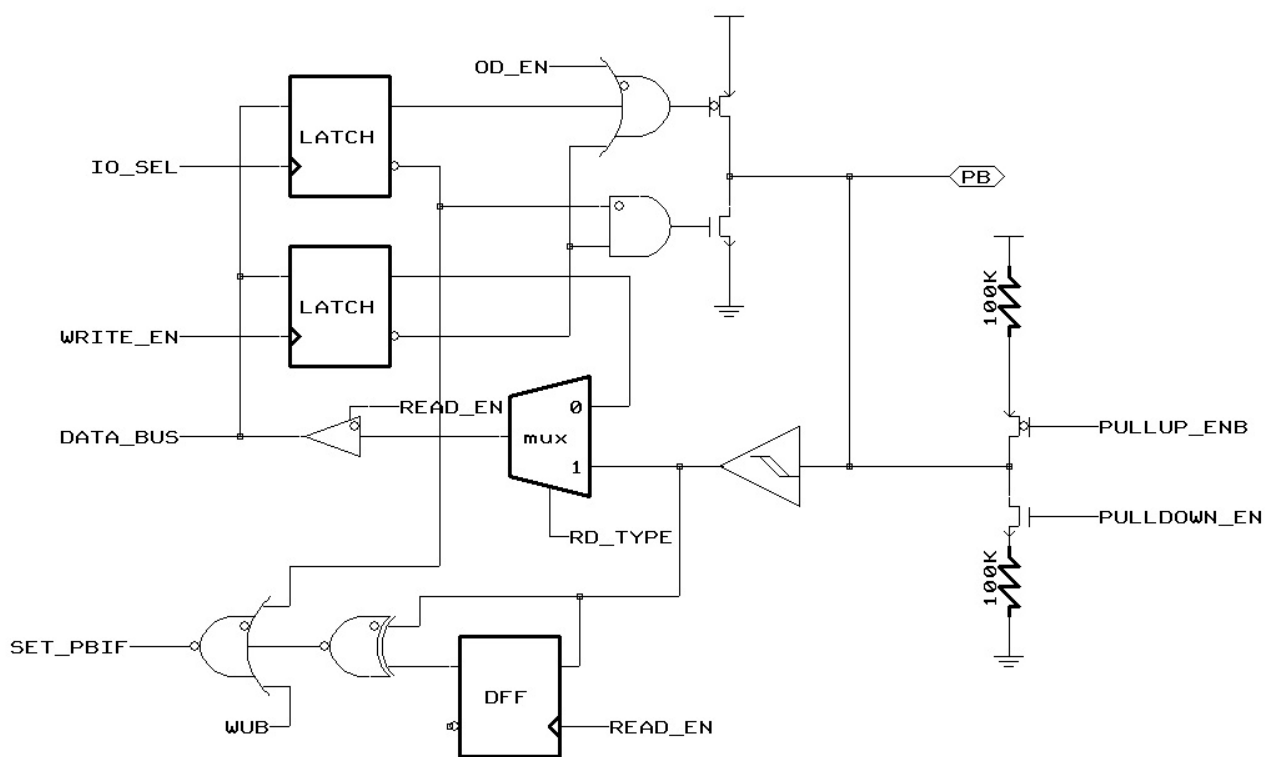


图 9 PB2 引脚结构框图

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。

PULLUP_ENB: 开启内部上拉电阻。

PULLDOWN_EN: 开启内部下拉电阻。

RD_TYPE: 选择读取脚位或数据锁存器。

WUB: 使能PB唤醒功能。

SET_PBIF: PB唤醒标志。

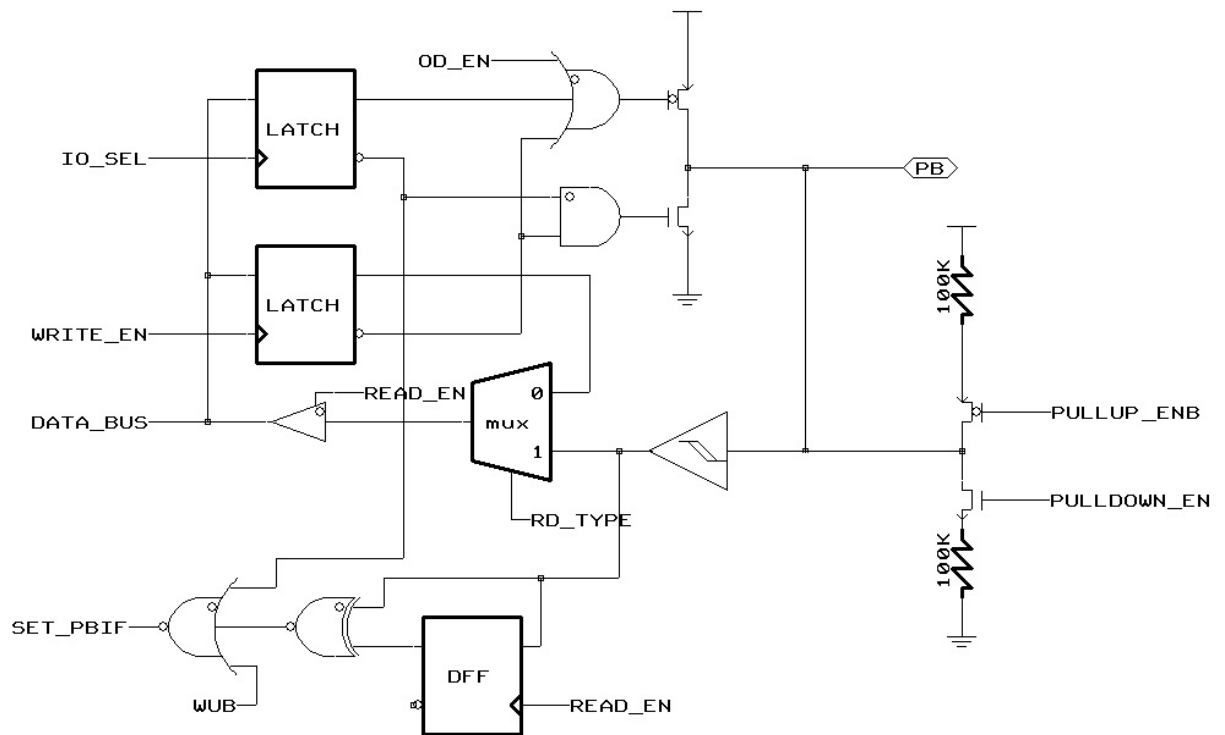


图 10 PB3 引脚结构框图

3.6 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}、外部时钟输入脚 EX_CKIO 或低频振荡 I_LRC 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 是 1 且 LCK_TM0 为 1，会选择低频振荡 I_LRC 当作定时器 0 时钟源。汇总成表格如下。(也请参考表 21)

定时器 0 时钟源	T0CS	LCKTM0	定时器 0 来源	低频振荡
Instruction clock	0	X	X	X
EX_CKIO	1	0	X	X
		X	0	
I_LRC	1	1	1	0

表 15 定时器 0 时钟源摘要

当使用低频振荡器频率作为 Timer0 时钟源时，建议使用预分频器 0 (见下面的描述)，并且比率设置为 4 以上，否则可能会出现漏计数。

在定时器 0 时钟源提供给定时器 0 之前，如果寄存器位 PS0WDT (T0MD[3]) 清除到 0，定时器 0 可能被预分频器 0 分割。当通过指令将 0 写入 PS0WDT 时，预分频器 0 被分配给定时器 0，并且在执行该指令后，预分频器 0 将被清除。预分频器 0 的分割率由寄存器位 PS0SEL[2:0] 决定，该寄存器位从 1:2 到 1:256。

当定时器 0 上溢，寄存器 TOIF (INTF[0]) 将会设定为 1，以表明定时器 0 发生上溢中断。如果寄存器 TOIE (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 TOIF，TOIF 才会被清除。

定时器 0 与 WDT 的结构框图如下图：

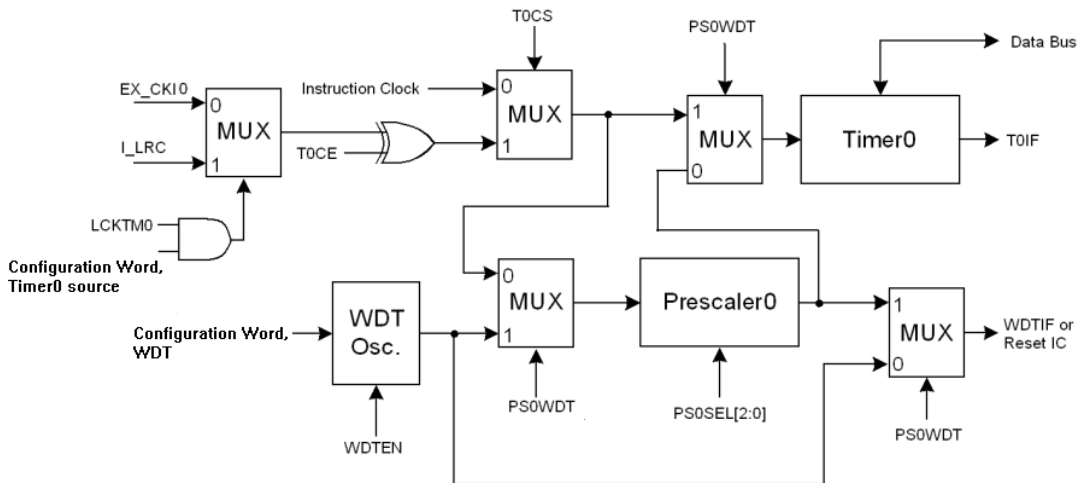


图 11 定时器 0 与 WDT 结构框图

3.7 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。Timer1 内建了自动重载功能，定时器 1 重载寄存器用双缓冲区存储重载数据。当用户写定时器 1 重载寄存器时，先写入定时器 1 高 2 位 (TMRH[5:4]) 再写入 TMR1。当 T1EN=1 时，定时器 1 重载寄存器会在定时器 1 发生上溢时将数据更新至定时器 1 计数器中。当 T1EN=0 时，定时器 1 重载寄存器会立即写入定时器 1。读取定时器 1 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：

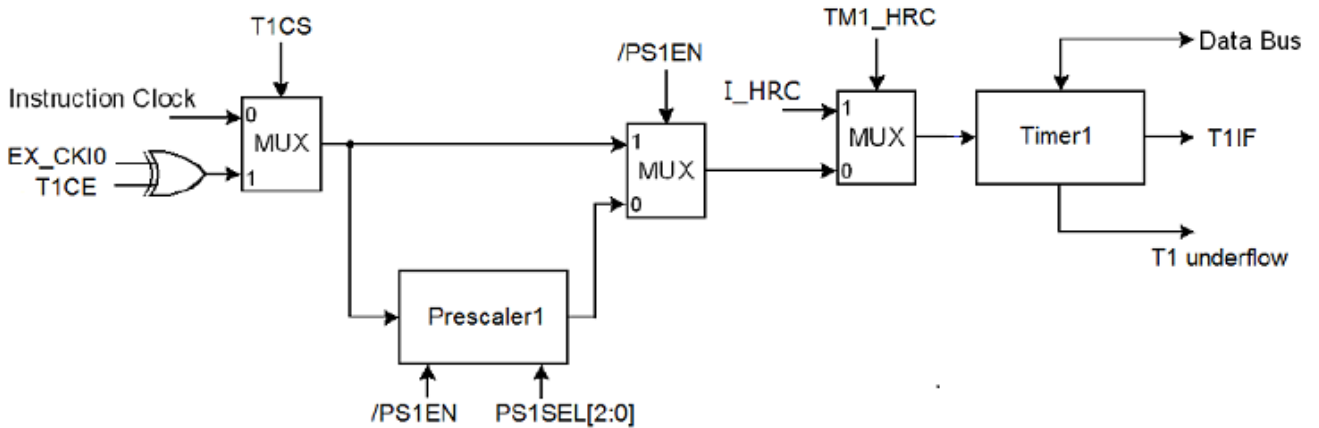


图 12 定时器 1 结构框图

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]) 可决定时钟源是指令时钟 F_{INST} 或外部时钟 EX_CKIO。当 T1CS 为 0，指令时钟会被选择当做时钟源。当 T1CS 为 1，则是 EX_CKIO 当做时钟源。当 EX_CKIO 被选取，寄存器控制位 T1CE (T1CR2[4]) 可决定 EX_CKIO 的时钟触发沿。当 T1CE 是 1，EX_CKIO 的下降沿将让定时器 1 计数减一。当 T1CE 是 0，EX_CKIO 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]) 会决定计数的初始值。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：

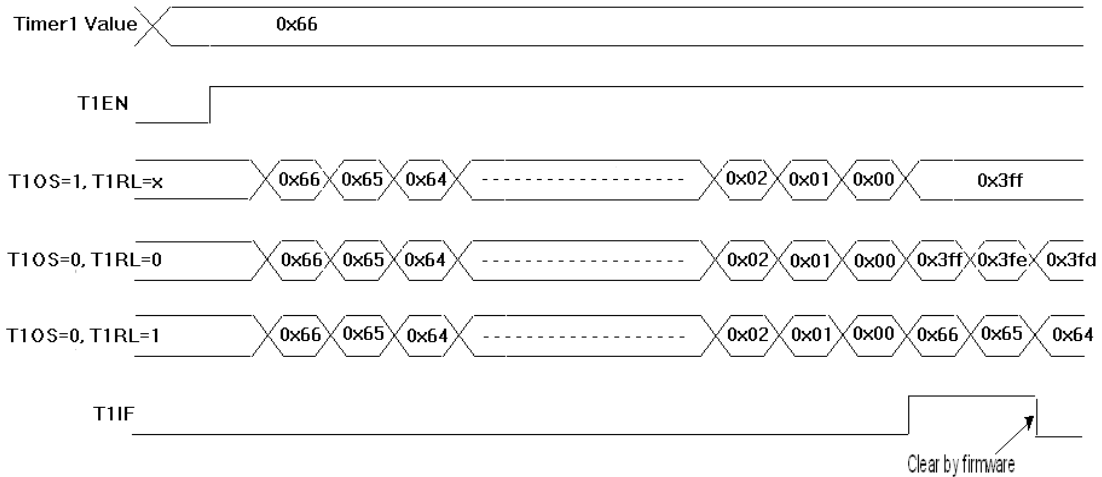


图 13 定时器 1 时序图

当寄存器PWM1OEN (T1CR1[7]) 设定为 1, PB3 为PWM1 输出。当PWM1OEN为 1, PB3 会自动成为输出脚。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。

PWM1 的占空比与帧率皆可编程的。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。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]。当用户写 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 发生上溢后, PWM1DUTY寄存器会重新更新。

PWM1 的结构框图如下:

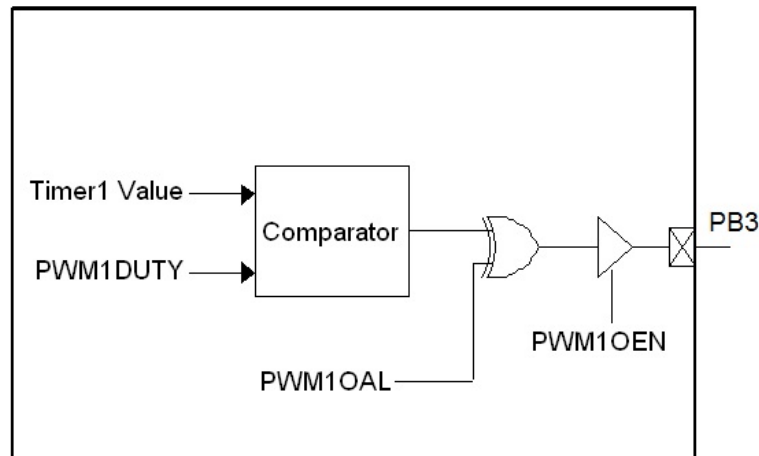


图 14 PWM1 结构框图

当寄存器BZ1EN (BZ1CR1[7]) 设定成 1 且使能配置字节, PB3 为蜂鸣器 1 输出。此外, PB3 会自动成为输出脚。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:

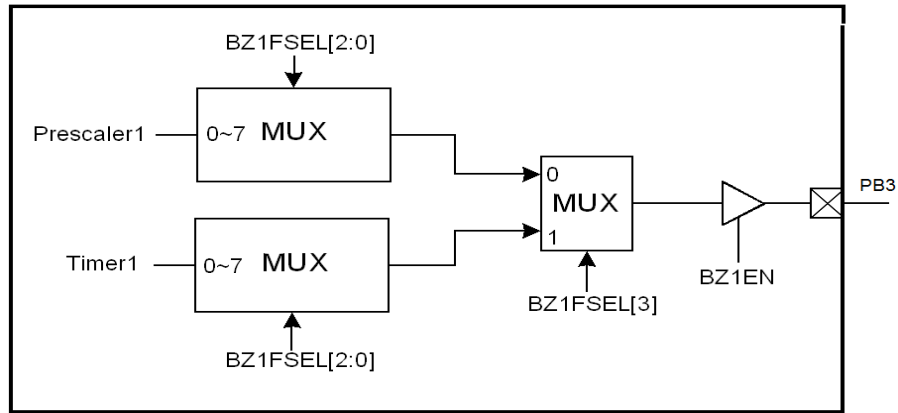


图 15 蜂鸣器 1 结构框图

注意：PB3 复用引脚输出优先级为 PWM1 输出 > 蜂鸣器 1 输出。

3.8 PWM2

当寄存器PWM2OEN (T2CR1[7]) 设定为 1, PB2 为PWM2 输出。当PWM2OEN为 1, PB2 会自动成为输出脚。PWM2 输出的有效状态是由寄存器PWM2OAL (T2CR1[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。

PWM2 的占空比与帧率皆可编程的。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。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TMRH[5:4] + TMR1[7:0]。当用户写 PWM2DUTY时, 先写入PWM2DUTY[9:8]高 2 位 (TMRH[3:2]) 再写入PWM2DUTY[7:0], 等到定时器 1 发生上溢后, PWM2DUTY寄存器会重新更新。

PWM2 的结构框图如下:

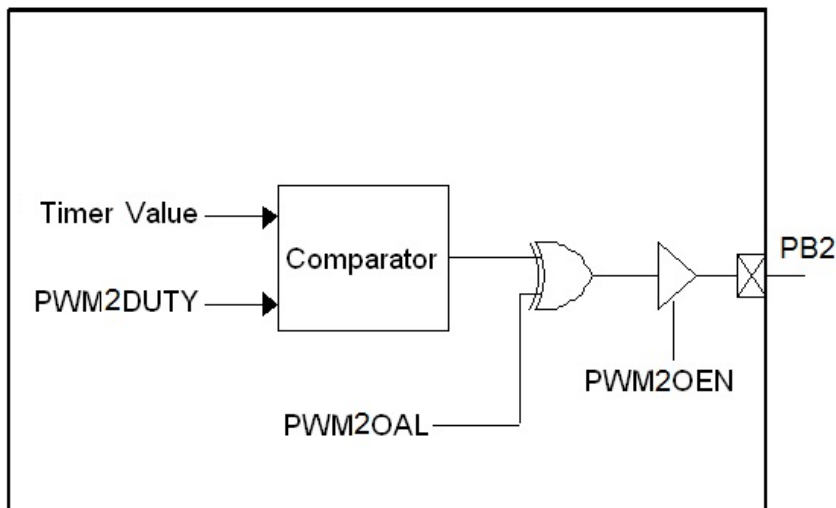


图 16 PWM2 结构框图

3.9 PWM3

当寄存器PWM3OEN (T3CR1[7]) 设定为 1, PA2 为PWM3 输出。当PWM3OEN为 1, PA2 会自动成为输出脚。PWM3 输出的有效状态是由寄存器PWM3OAL (T3CR1[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。

PWM3 的占空比与帧率皆可编程的。占空比是由寄存器PWM3RH[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。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TMRH[5:4] + TMR1[7:0]。当用户写PWM3DUTY时, 先写入PWM3DUTY[9:8]高 2 位 (PWM3RH[1:0]) 再写入PWM3DUTY[7:0], 等到定时器 1 发生上溢后, PWM3DUTY寄存器会重新更新。

PWM3 的结构框图如下:

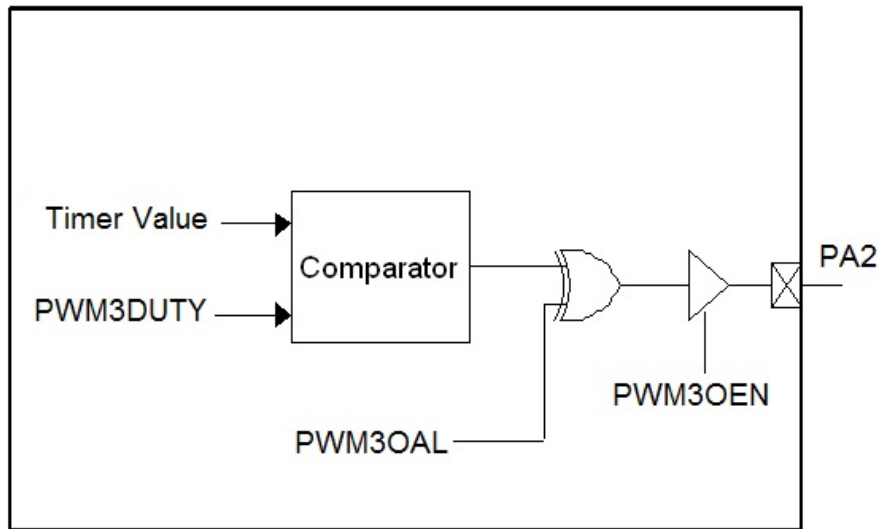


图 17 PWM3 结构框图

3.10 电阻/频率转换器模式 (RFC)

NY8B060D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。当被选择的输入引脚状态被识别为 0 (RFC输入引脚电压低于 V_{IL}), 定时器 1 将会持续计数, 当被选择的输入引脚状态被识别为 1 (RFC输入引脚电压高于 V_{IH}), 定时器 1 停止计数。RFC模式的工作原理如下图所示: 使用PSEL3~0 从 6 个NY8B060D输入引脚中选择一个RFC输入引脚。RFCEN用于将Timer1 使能信号在正常的使能信号T1EN和RFC选择的输入状态之间进行切换。

RFC模式的一个应用是测量电容-电阻充电时间。如下图, 当PSEL3~0=0x02, PA2 为RFC输入引脚。首先设置PA2 输出 0 (低于 V_{IL}), 接着清除定时器 1 的内容, 设置PA2 为输入引脚并开启RFC模式, 定时器 1 会开始计数。这时RC电路开始对PA2 引脚充电。当PA2 引脚电压高于 V_{IH} 时, 定时器 1 会停止计数。定时器 1 相当于记录RC电路充电时间。

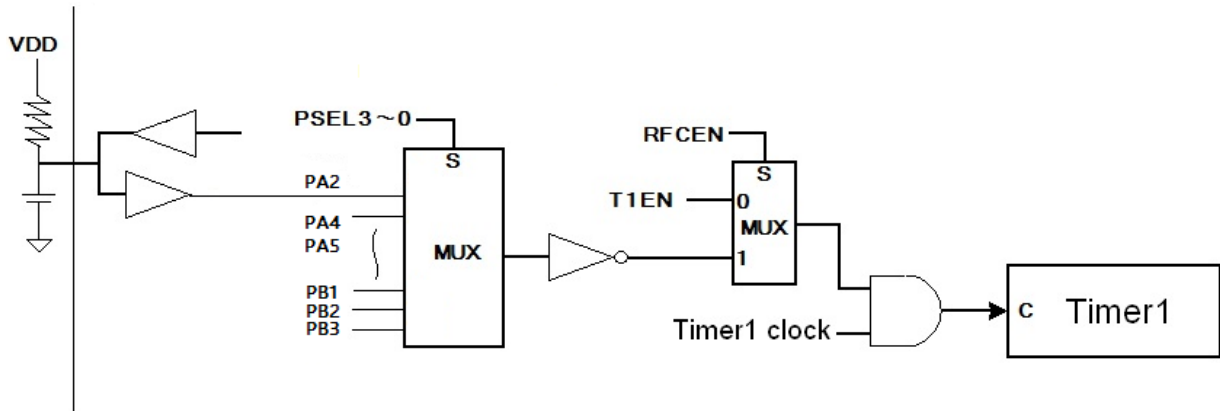


图 18 RFC结构框图

3.11 IR Carrier

寄存器IREN (IRCR[0]) 被设定为 1 后, PB1 为红外线载波输出, 而PB1 会自动成为输出脚。当IREN清零, PB1 将会成为一般I/O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。由于红外线载波频率是由高速振荡频率FHOSC导出的, 所以当使用外部晶体时, 必须明确系统振荡的频率。当采用内部高频振荡时, 该寄存器会被忽略, 并且为IR模块提供 4MHz的时钟。

红外线载波的极性会根据PB1 输出数据来决定。当寄存器IRCSEL (IRCR[2]) 为 1 且PB1 输出数据为 0 时, 红外线载波将由PB1 输出。当寄存器IRCSEL (IRCR[2]) 为 0 且PB1 输出数据为 1 时, 红外线载波将由PB1 输出。

红外线载波的极性如下图所示:

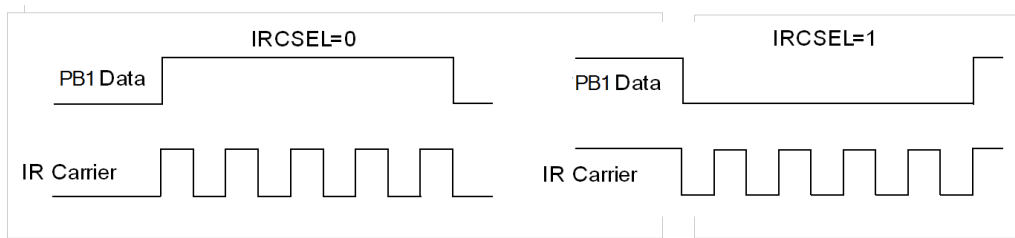


图 19 红外线载波的极性 vs. PB1 数据

3.12 低电压侦测 (LVD)

NY8B060D内置准确的低电压侦测电路来侦测VDD电压电平。将LVDEN (寄存器PCON[5]) 设为 1 后, 当VDD 电压低于下表LVDS[3:0]选择的电压值时, 读取LVDOUT会得到 0。如果开启LVD中断使能位, LVD中断标志位将会被设置为 1, 且如果GIE=1 时, 程序将跳入中断子程序。此外, LVD实际状态输出可以通过寄存器PCON1[6] 进行轮询。以下是LVD框图:

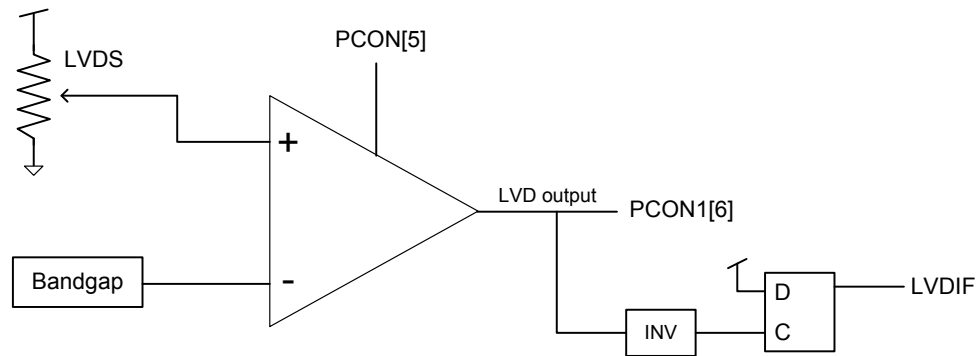


图 20 LVD结构框图

LVD电压选择如下表所示:

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 16 选择LVD电压

注意:

LVD 的迟滞电压 (由低到高) 约为 0.1V。

在电池充电应用中 (检测电压由低到高),

LVD 电压选择表如下:

LVDS[3:0]	Voltage
0000	--
0001	--
0010	(2.2+0.1) V
0011	(2.4+0.1) V
0100	(2.6+0.1) V
0101	(2.8+0.1) V

LVDS[3:0]	Voltage
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 控制流程如下:

步骤 1: 通过 LVDS[3:0]选择 LVD 电压。

步骤 2: 设置 CMPCR = 0x0A 。

步骤 3: 设置 PCON[5]=1 (启用 LVD)

步骤 4: 通过 PCON1[6]检查 LVD 状态。

注意: 如果改变 LVD 电压 LVDS[3:0], 用户必须等待至少 50us(@F_{HOSC}=1MHz)才能通过 PCON1[6]获得正确的 LVD 状态。

3.13 电压比较器

NY8B060D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可以与GPIO共享。

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。

比较器结构框图如下图所示:

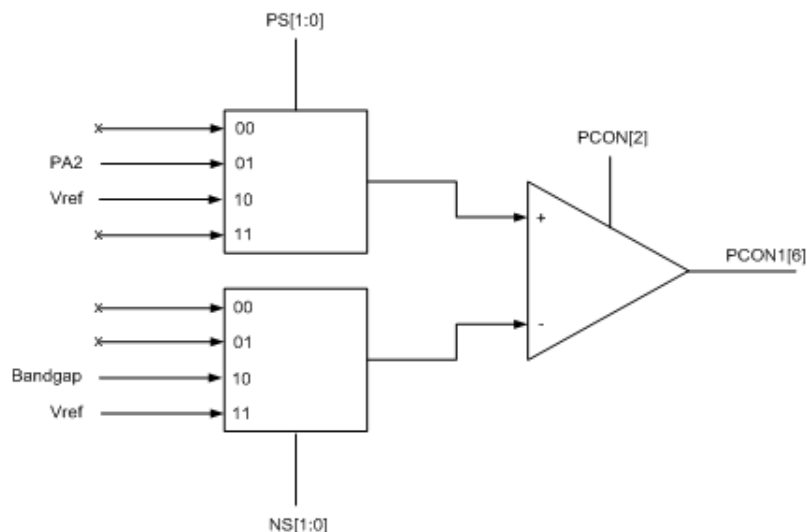


图 21 比较器结构框图

3.13.1 比较器参考电压 (Vref)

内部参考电压Vref由串联电阻构成，提供不同等级的参考电压。RBIAS_H 和RBIAS_L 用于选择 Vref 的最大值和最小值，LVDS[3:0]用于从 16 个电压等级中选择一个。

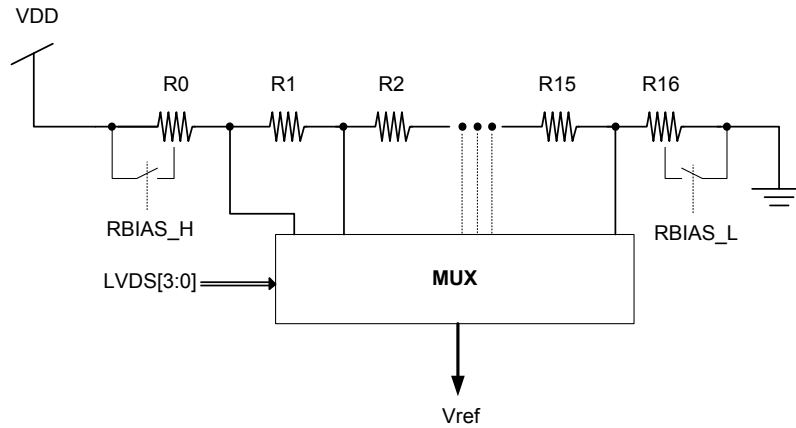


图 22 Vref 硬件连接

Vref由RBIAS_H, RBIAS_L 和 LVDS[3:0]决定。LVDS[3:0]用于从 16 个参考电压中选择一个，如下表所示。

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 17 内部参考电压Vref选择表

注意：Vref的误差为±0.1V。

比较器的正输入源由PS[1:0]（寄存器CMPPCR[3:2]）确定。如下表所示：

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

表 18 正输入源选择

比较器的负输入源由NS[1:0]（寄存器CMPPCR[1:0]）确定。如下表所示：

NS[3:0]	负输入源选择
00	--
01	--
10	Bandgap (0.65V)
11	Vref

表 19 负输入源选择

比较器输出结果的获取方式有两种：一种是通过寄存器轮询，另一种是通过探测输出引脚。

比较器输出可以通过 LVDOUT（寄存器 PCON1[6]）进行轮询。

要在输出端探测比较器输出，可以将 CMPOE(寄存器 OSCCR[6])设置为 1，然后 PB1 将是比较器输出的实时状态。需要注意的是，当 CMPOE=1 时，启用 PWM3 功能将会被禁用。

3.14 ADC模数转换器

NY8B060D提供 6+1 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参考电压可选由内部产生的VDD，4V，3V或 2V提供。模拟输入可以从输入引脚PA2，PA4，PB1~PB3 或内部 1/4VDD中选择。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开始工作前，先设置ADEN=1，然后设置START=1，ADC开始将模拟信号转换为数字信号。EOC=0 表示ADC正在处理中。EOC=1 表示ADC转换结束。如果ADIE=1 且全局中断被启用，则在EOC由低转高后会发出ADC中断。结构框图如下所示：

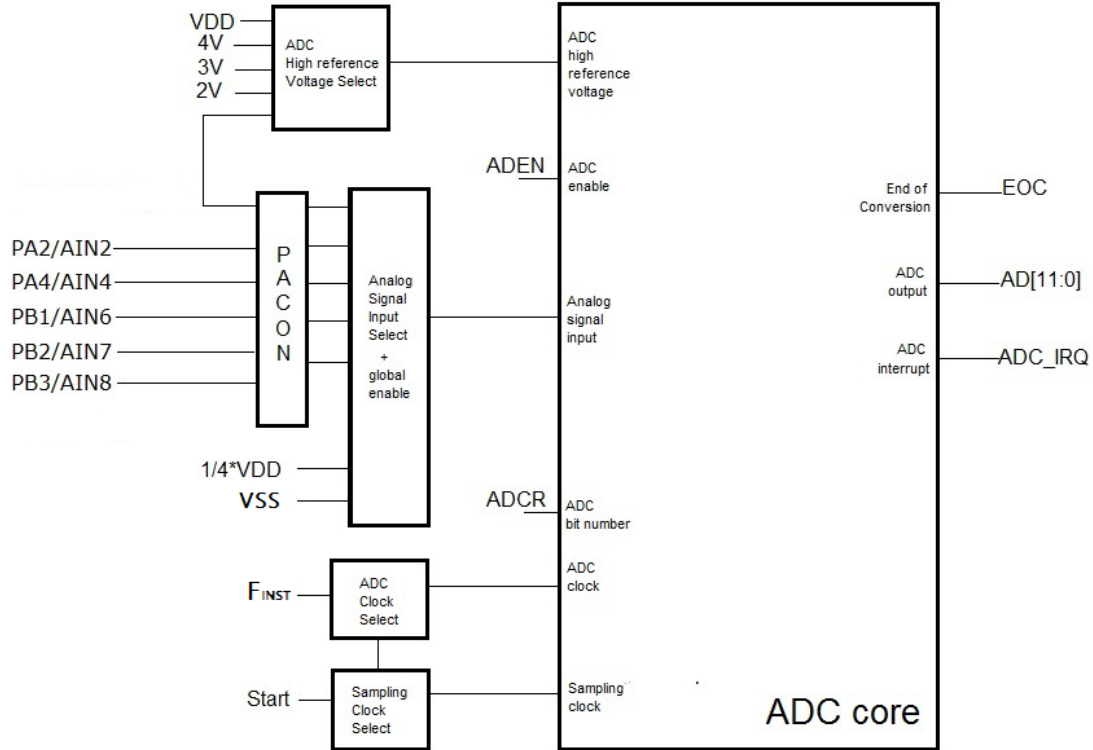


图 23 ADC结构框图

3.14.1 ADC 参考电压

ADC可选 4 种参考电压，可由寄存器ADVREFH来设置。ADC内置四个高参考电压源，由ADVREFH寄存器控制。这些高参考电压源是内部电压源(VDD, 4V, 3V, 2V)。

当 EVHENB 位为 0 时，ADC 参考电压来自 VHS[1:0]选择的内部电压源。如果 VHS[1:0]为“11”，则 ADC 参考电压为 VDD。如果 VHS[1:0]为“10”，则 ADC 参考电压为 4V。当 VHS[1:0]为“01”时，ADC 参考电压为 3V。如果 VHS[1:0]为“00”，则 ADC 参考电压为 2V。内部参考电压应用的限制是 VDD 不能低于每一个内部电压的电平，或电平等于 VDD。ADC 的采样电压范围受参考电压高低的限制。ADC 低参考电压为 VSS，不可改变。ADC 高参考电压包括内部 VDD/4V/3V/2V。ADC 参考电压范围限制为(ADC 高参考电压-低参考电压)≥ 2V。ADC 低参考电压为 VSS=0V。因此 ADC 高参考电压范围为 2V ~ VDD。

ADC模拟输入信号电压必须从ADC低参考电压到ADC高参考电压。如果ADC模拟输入信号电压超过这个范围，ADC转换结果是不可预料的(满量程或零)。

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

表 20 选择ADC参考电压

3.14.2 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	0010	PA2
1	0100	PA4
1	0110	PB1
1	0111	PB2
1	1000	PB3
1	1011	1 / 4 * VDD
1	1100	GND

表 21 ADC模拟输入通道选择

ADC 输入引脚与数字 I/O 引脚共用。连接模拟信号到这些引脚可能会导致额外的电流泄漏在 I/O 引脚。在断电模式下，上述漏电流将是一个大问题。为解决上述问题，PACON[2,4]为 PA[2,4]配置寄存器，PACON[6:7]为 PB[1:2]配置寄存器，ADCR[4]为 PB[3]配置寄存器。向 PACON 和 ADCR 写“1” [4:6]会将相关 PA/PB 引脚配置为纯模拟输入引脚，避免电流泄漏，不能作为正常的 I/O 使用。

除了设置 PACON 和 ADCR 寄存器外，所选的模拟输入引脚必须设置为输入模式，内部的拉高/拉下必须禁用，否则模拟输入电平可能会受到影响。

3.14.3 ADC 时钟（ADCLK），采样时钟（SHCLK）与位数选择

转换速度和转换精度受ADC时钟(ADCLK)、采样时钟(SHCLK)和转换位数的选择影响。ADCLK是ADC的基准时钟。SAR ADC在运行过程中，位运算与ADCLK同步。SHCLK是模拟信号采样时间的持续时间，较大的SHCLK会更紧密地恢复原始模拟信号电平，但会降低ADC的转换速度。反之亦然。ADC可以根据ADCR[1:0]寄存器位来选择不同的转换位数。有 3 种选择，12 位、10 位和 8 位。转换位数越少，ADC转换速度越快，但有效ADC位数越少。转换位数越多，转换速度越慢，但精度越高。

ADC时钟由F_{INST}导出，可从ADCK[1:0]中选择。

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 22 ADC时钟选择

采样时钟宽度由ADCLK导出，并可从SHCK[1:0]中选择。

SHCK[1:0]	ADC采样时间
0 0	1 个ADC时钟周期

0 1	2 个ADC时钟周期
1 0	4 个ADC时钟周期
1 1	8 个ADC时钟周期

表 23 ADC采样时间选择

ADC位数选择来自ADCR[1:0]。

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

表 24 ADC位数选择

ADC转换时间从START（开始ADC转换）一直到EOC=1（结束ADC转换）。持续时间取决于ADC分辨率和ADC时钟速率和采样时间。

ADC 转换时间 ≈ 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)	Fcpu=2MHz		Fcpu=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 25 ADC转换时间与转换率

3.14.4 ADC 操作顺序

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

在ADEN设置为 1 后必须等待 256us (ADC电路启动时间), 再将START位写 1 来启动ADC模数转换。ADC转换过程中, 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。

3.15 看门狗定时器 (WDT)

NY8B060D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上溢后可由配置字节决定的复位NY8B060D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上溢的时基由配置字节决定,可以是 3.5 毫秒、15 毫秒、60 毫秒或 250 毫秒。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。如果WDT上溢将复位NY8B060D,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CLRWDWT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。

3.16 中断

NY8B060D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 7 种:

- Timer0 上溢中断。
- Timer1 下溢中断。
- WDT中断。
- PA/PB 输入状态改变中断。
- 外部中断 1 输入。
- 低电压侦测中断。
- ADC模数转换完成中断。

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NY8B060D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, NY8B060D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NY8B060D。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指令可能会误清除中断标志。

3.16.1 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。

3.16.2 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。

3.16.3 看门狗超时中断

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

3.16.4 PA/PB 输入状态改变中断

当PA_x (x = 2, 4, 5), PB_y (y = 1, 2, 3) 设置为输入口且相应的寄存器WUPA_x、WUPB_x位设置为 1，且如果PABIE和GIE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 并处理此中断请求。需注意当PB1 设置为状态变化中断和外部中断时，设置EIS1=1 将禁止PB1 状态变化中断。

3.16.5 外部中断 1 输入

根据EIS1=1 和寄存器INTEDG的配置，如果INT1IE和GIE设置为 1，PB1 引脚上的有效边沿触发会让寄存器INT1IF位将被硬件设为 1 并处理此中断请求。

3.16.6 低电压侦测中断

当VDD电压水平低于LVD电压，读取LVDOUT（寄存器PCON1[6]）会得到 0。如果LVDIE和GIE设置为 1，寄存器LVDIF位将被硬件设为 1 并处理此中断请求。

3.16.7 ADC 模数转换完成中断

当ADC模数转换完成时，如果ADIE和GIE设置为 1，寄存器ADIF位将被硬件设为 1 并处理此中断请求。

3.17 振荡器配置

因为NY8B060D是双时钟IC，有高振荡时钟（F_{HOSC}）和低振动时钟（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}）。可用作F_{LOSC}的振荡器是内部低速RC振荡器（I_{LRC}）。

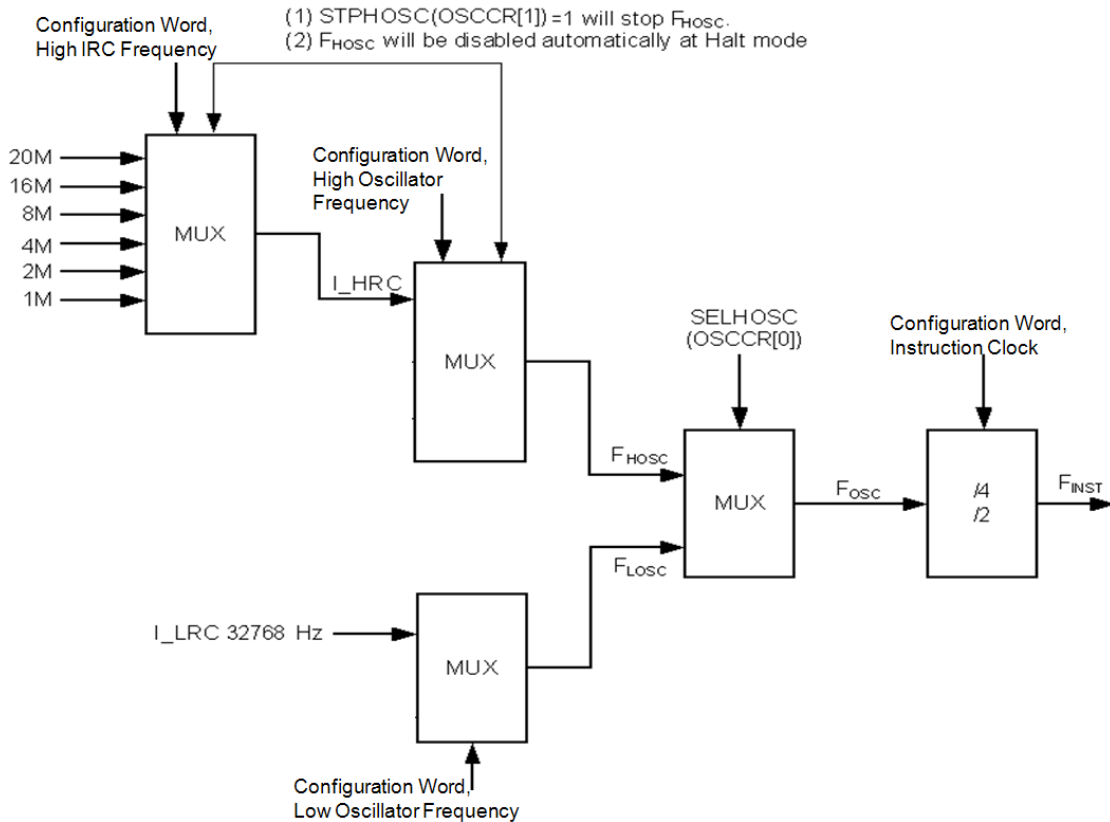


图 24 NY8B060D 振荡配置结构图

有两个配置字来决定哪一个振荡器将被用作 F_{HOSC} 。当选择 I_HRC 作为 F_{HOSC} 时， I_HRC 输出频率由三个配置字决定，可以是 1M、2M、4M、8M、16M 或 20MHz。

有一个配置字来确定哪个振荡器将被用作 F_{LOSC} 。选择 I_LRC 时，其频率约为 32768Hz。

以下列出了 F_{HOSC} 和 F_{LOSC} 的双时钟组合。

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

表 26 双时钟组合

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 $F_{OSC}/4$ 。

3.18 工作模式

NY8B060D提供了四种操作方式来定制各种应用和节省电力消耗，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，NY8B060D将停止几乎所有的运作，可由定时器 0 / 定时器 1 / 看门狗来唤醒。在睡眠模式下，NY8B060D将睡眠直到外部事件或看门狗触发来唤醒。

四种工作模式如下图所示。

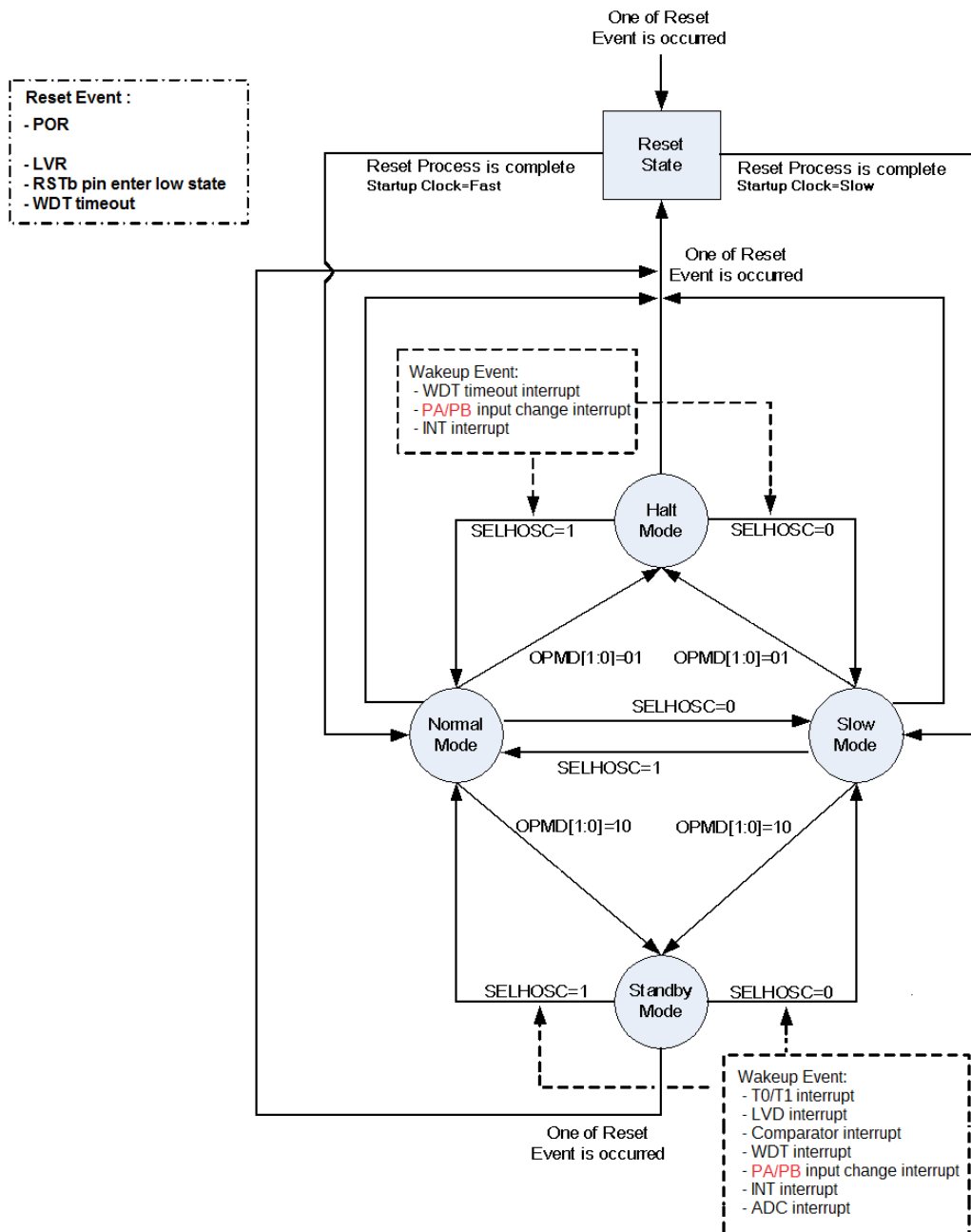


图 25 四种工作模式

3.18.1 正常模式

发生任何复位事件并且复位过程完成后，NY8B060D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 F_{HOSC} ，NY8B060D将进入正常模式，如果启动时钟为 F_{LOSC} ，NY8B060D将进入慢速模式。在正常模式下，为提供最高性能而以 F_{HOSC} 作为系统振荡时钟，其功耗在四种操作模式中将是最大的。在上电或任何重置触发器被释放后，待复位程序完成NY8B060D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, NY8B060D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。

3.18.2 慢速模式

通过写 0 至寄存器SELHOSC位, NY8B060D将进入慢速模式。在低速模式下, 为节省功耗, F_{Losc}被选为系统振荡时钟。然而, F_{Hosc}将不会自动被NY8B060D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}, 必须先进入慢速模式, 然后关闭F_{Hosc}。

- 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。

3.18.3 待机模式

通过写入 10b至寄存器OPMD[1:0], NY8B060D将进入待机模式。然而, 在待机模式下, F_{Hosc}不会自动被NY8B060D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}。部分NY8B060D的硬件功能会被关闭, 如T0EN / T1EN位被设置为 1 则定时器仍可运作。因此Timer0 / Timer1 溢出后NY8B060D会被唤醒。

- 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IC便能从待机模式唤醒:
(a)Timer0 上溢中断 / Timer1 下溢中断 (b)看门狗超时中断 (c)PA/PB输入状态改变中断 (d)外部中断。
- 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)。

3.18.4 睡眠模式

通过执行SLEEP指令或写入 01b至寄存器OPMD[1:0]位, NY8B060D将进入睡眠模式。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NY8B060D只能通过一些特殊事件唤醒。因此, 睡眠模式是NY8B060D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IC便能从睡眠模式中唤醒：
(a)看门狗超时中断 (b)PA/PB输入状态改变中断 (c)INT0/1 外部中断。
- 从睡眠模式唤醒后，如SELHOSC=1，IC将回到正常模式，如SELHOSC=0 则IC将回到慢速模式。
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。

3.18.5 唤醒等待时间

睡眠模式的唤醒等待时间由配置字决定：高频振荡或低频振荡。唤醒周期设置为 $16 \cdot F_{osc}$ 。另一方面，待机模式不需要唤醒等待时间，因为 F_{Hosc} 或 F_{Losc} 仍然在待机模式下运行。

在NY8B060D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，NY8B060D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。

3.18.6 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	停止	停止
定时器 0/1	TxEN	TxEN	TxEN	关闭
WDT	配置和WDTEN	配置和WDTEN	配置和WDTEN	配置和WDTEN
其它硬件	硬件使能位	硬件使能位	硬件使能位	全部关闭
唤醒源	-	-	- Timer0/1 上溢 - WDT超时 - PA/PB输入状态改变 -外部中断 1 - LVD中断 - ADC转换完成	- WDT超时 - PA/PB输入状态改变 -外部中断 1

表 27 工作模式概述

3.19 复位

当以下任一复位事件发生时，NY8B060D将会进入复位状态并开始复位动作：

- 当VDD检测到上升沿时为上电复位。
- 当VDD电压低于预设的LVR电压时，为LVR复位。
- RSTb引脚为低电平。
- WDT超时复位。

此外，所有寄存器如果初始值未知时，寄存器将会被初始化为初始值或保持不变。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
----	-----	-----

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 28 /TO和/PD值和相关事件概述

复位事件发生后，NY8B060D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。上电复位时间后，NY8B060D将等待振荡器进一步启动时间(OST)，才开始执行程序。如果上电时间为 140us，则OST= F_{osc} 的 1 个时钟周期，如果上电时间为 4.5ms、18ms、72ms或 288ms，则OST= F_{osc} 的 16 个时钟周期。

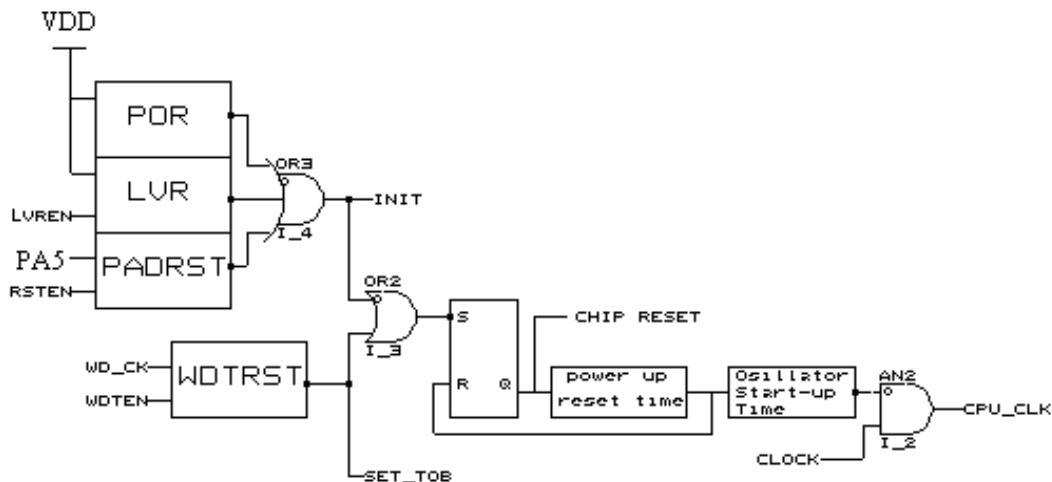


图 26 芯片复位电路框图

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断电时快速放电。

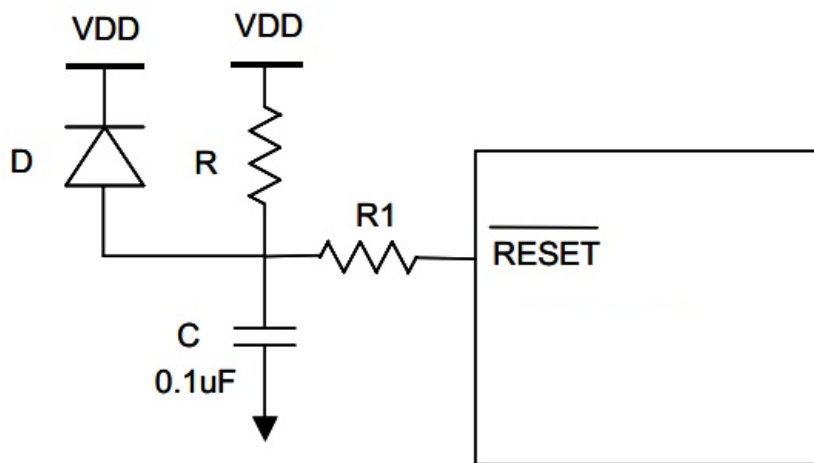


图 27 外部上电复位硬件连接图

4. 指令设置

NY8B060D为各种应用程序提供了 55 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	2	-
LCALL	adr		Call subroutine	2	-
LGOTO	adr		unconditional branch	2	-

表 29 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 ~ 0x15。

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
语法	ADCAR R, d
操作数	$0 \leq R \leq 127$ $d = 0, 1$
操作	$R + ACC + C \rightarrow dest$
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"。
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。

ADDAR	Add ACC and R
语法	ADDAR R, d
操作数	$0 \leq R \leq 127$ $d = 0, 1$
操作	$ACC + R \rightarrow dest$
状态影响	Z, DC, C
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
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
语法	ADCIA i
操作数	$0 \leq i < 255$
操作	$ACC + i + C \rightarrow ACC$
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。
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。

ADDIA	Add ACC and Immediate
语法	ADDIA i
操作数	$0 \leq i < 255$
操作	$ACC + i \rightarrow ACC$
状态影响	Z, DC, C
说明	ACC和 8 位立即数加法，结果存入ACC。
周期	1
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	BCR	Clear Bit in R
语法	ANDAR R, d	语法	BCR R, bit
操作数	$0 \leq R \leq 127$ $d = 0, 1$	操作数	$0 \leq R \leq 127$ $0 \leq \text{bit} \leq 7$
操作	ACC & R \rightarrow dest	操作	$0 \rightarrow R[\text{bit}]$
状态影响	Z	状态影响	--
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”。	说明	将R寄存器的bit位(0~7)清0。
周期	1	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。	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	BSR	Set Bit in R
语法	ANDIA i	语法	BSR R, bit
操作数	$0 \leq i < 255$	操作数	$0 \leq R \leq 127$ $0 \leq \text{bit} \leq 7$
操作	ACC & i \rightarrow ACC	操作	$1 \rightarrow R[\text{bit}]$
状态影响	Z	状态影响	--
说明	ACC和8位立即数做“AND”运算。	说明	设置R寄存器的bit位为1。
周期	1	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。	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预分频器 (若开启) 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=0x12, Z=0, C=0。 执行指令后: R=0x34, ACC=0x12, Z=0, C=1。

DAA	Convert ACC Data Format from Hexadecimal to Decimal	DECRSZ	Decrease R, Skip if 0
语法	DAA	语法	DECRSZ R, d
操作数	--	操作数	$0 \leq R \leq 127$ $d = 0, 1$
操作	ACC(hex) → ACC(dec)	操作	$R - 1 \rightarrow \text{dest}$, Skip if result = 0
状态影响	C	状态影响	--
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。	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	周期	1 or 2（跳过）
举例	ADDAR R, d DAA 执行指令前： ACC=0x28, R=0x25, d=0。 执行指令后： ACC=0x53, C=0。	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	DISI	Disable Interrupt Globally
语法	DECR R, d	语法	DISI
操作数	$0 \leq R \leq 127$ $d = 0, 1$	操作数	--
操作	$R - 1 \rightarrow \text{dest}$	操作	Disable Interrupt, $0 \rightarrow \text{GIE}$
状态影响	Z	状态影响	--
说明	R - 1，若d="0"，结果存入ACC；若d="1"，结果存入R。	说明	GIE设置为 0，关闭总中断。
周期	1	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。	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 → 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} → 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(skip)
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	$0 \leq i < 255$
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	$0 \leq i < 255$
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

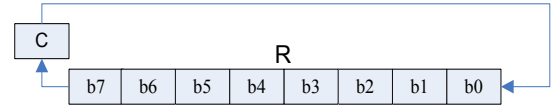
MOVR	Move to ACC or R
语法	MOVR R, d
操作数	$0 \leq R \leq 127$ d = 0, 1
操作	$R \rightarrow \text{dest}$
状态影响	Z
说明	R-page寄存器赋值给d, 若d="0", 结果存入ACC; 若d="1", 结果存入寄存器R。指令执行后, 通过状态标志Z检查R是否为0。
周期	1
举例	MOVR R, d 执行指令前: R=0x0, ACC=0xAA, Z=0, d=0。 执行指令后: R=0x0, ACC=0x00, Z=1。

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 level=1。

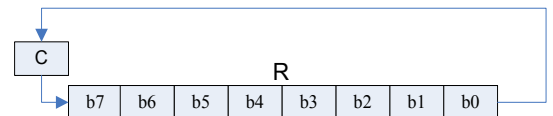
RETIA	Return with Data in ACC
语法	RETIA i
操作数	$0 \leq i < 255$
操作	$i \rightarrow \text{ACC}$ Top of Stack \rightarrow PC
状态影响	--
说明	带参数返回：8 位立即数赋值给 ACC，栈顶地址载入 PC。
周期	2
举例	RETIA i 执行指令前： Stack pointer =2, i=0x55, ACC=0xAA。 执行指令后： PC=Stack[2], Stack pointer =1, ACC=0x55。

RET	Return from Subroutine
语法	RET
操作数	--
操作	Top of Stack \rightarrow 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
语法	SBCAR R, d
操作数	$0 \leq R \leq 127$ $d = 0, 1$
操作	$R + (\sim\text{ACC}) + C \rightarrow \text{dest}$
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。
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)

SBCIA	Subtract ACC and Carry from Immediate
语法	SBCIA i
操作数	$0 \leq i < 255$
操作	$i + (\sim\text{ACC}) + C \rightarrow \text{dest}$
状态影响	Z, DC, C
说明	常数和ACC带借位减法，结果存入ACC。
周期	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	$0 \leq S \leq 21$
操作	ACC \rightarrow 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	SUBAR	Subtract ACC from R
语法	SFUNR S	语法	SUBAR R, d
操作数	$0 \leq S \leq 21$	操作数	$0 \leq R \leq 127$ $d = 0, 1$
操作	S-page SFR \rightarrow ACC	操作	$R - ACC \rightarrow \text{dest}$
状态影响	--	状态影响	Z, DC, C
说明	读S-page特殊寄存器到ACC	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	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。	举例	SUBAR 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	SUBIA	Subtract ACC from Immediate
语法	SLEEP	语法	SUBIA i
操作数	--	操作数	$0 \leq i < 255$
操作	00h \rightarrow WDT 00h \rightarrow WDT prescaler 1 \rightarrow /TO 0 \rightarrow /PD	操作	$i - ACC \rightarrow ACC$
状态影响	/TO, /PD	状态影响	Z, DC, C
说明	WDT和分频器0 清零。/TO标志为0, /PD清零, IC进入睡眠。	说明	8 位立即数减ACC, 结果存入ACC。
周期	1	周期	1
举例	SLEEP 执行指令前: /PD=1, /TO=0。 执行指令后: /PD=0, /TO=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	$R[3:0] \rightarrow \text{dest}[7:4]$, $R[7:4] \rightarrow \text{dest}[3:0]$
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 \rightarrow 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] \rightarrow ACC。 ROM data{TBHP, ACC} [13:8] \rightarrow 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, ACC=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 \rightarrow 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。

XORIA	Exclusive-OR Immediate with ACC
语法	XORIA i
操作数	$0 \leq i < 255$
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。

5. 配置字节表

项目	名称	选项
1	高频振荡模式	1. I_HRC
2	低频振荡模式	1. I_LRC
3	内部高速 RC 频率	1. 1MHz 2. 2MHz 3. 4MHz 4. 8MHz 5. 16MHz 6. 20MHz
4	指令时钟	1. 2 个振荡周期 2. 4 个振荡周期
5	看门狗定时器	1. 看门狗开启（寄存器控制） 2. 看门狗关闭（永远关闭）
6	看门狗定时器事件	1. 看门狗复位 2. 看门狗中断
7	定时器 0 时钟源	1. EX_CKIO 2. 低频振荡器（I_LRC/E_LXT）
8	PA.5	1. PA.5 为 I/O 口 2. PA.5 为复位脚
9	PA.7	1. PA.7 为 I/O 口 2. PA.7 输出指令时钟
10	上电复位时间	1. 140us 2. 4.5ms 3. 18ms 4. 72ms 5. 288ms
11	看门狗定时器时基	1. 3.5ms 2. 15ms 3. 60ms 4. 250ms
12	LVR 开关设定	1. 寄存器控制 2. LVR 永远开启
13	LVR 电压	1. 1.6V 2. 1.8V 3. 2.0V 4. 2.2V 5. 2.4V 6. 2.7V 7. 3.0V 8. 3.3V 9. 3.6V 10. 4.2V
14	VDD 电压	1. 3.0V 2. 4.5V 3. 5.0V
15	I/O 口灌电流类型(PA5 除外)	1. 一般灌电流 2. 大灌电流
16	比较器输入引脚选择	1. 开启 2. 关闭
17	读取输出口数据	1. I/O 口 2. 寄存器
18	E_LXT 强化起振开关设定	1. 自动关闭 2. 寄存器关闭
19	EX_CKIO to Inst. Clock	1. 同步 2. 不同步
20	上电时钟源	1. 高速（I_HRC） 2. 低速（I_LRC）
21	输入高电压（VIH）	1. CMOS（0.7VDD） 2. TTL（0.5VDD）
22	输入低电压（VIL）	1. CMOS（0.3VDD） 2. TTL（0.2VDD）

表 30 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	$F_{INST}=20MHz @ I_{HRC}/2$
			2.2				$F_{INST}=20MHz @ I_{HRC}/4$
			3.0				$F_{INST}=16MHz @ I_{HRC}/2$
			2.0				$F_{INST}=16MHz @ I_{HRC}/4$
			2.0				$F_{INST}=8MHz @ I_{HRC}/2$
			1.8				$F_{INST}=4MHz @ I_{HRC}/2$
			1.6				$F_{INST}=32KHz @ I_{LRC}/2$
			V_{IH}				输入高电平
3V	2.4	--		--	V	所有I/O引脚, EX_CK11, INT1 CMOS (0.7 V_{DD})	
5V	3.5	--		--			
3V	2.1	--		--	V	所有I/O引脚, EX_CK11 TTL (0.5 V_{DD})	
5V	2.5	--		--			
3V	1.5	--		--			
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_CK11, INT1 CMOS option (0.3 V_{DD})
		5V	--	--	1.5		
		3V	--	--	0.9	V	所有I/O引脚, EX_CK11 TTL option (0.2 V_{DD})
		5V	--	--	1.0		
		3V	--	--	0.6		
I_{OH}	输出驱动电流	5V	--	18	--	mA	$V_{OH}=4.0V$
		3V	--	10	--		$V_{OH}=2.0V$
I_{OL}	输出大灌电流	5V	--	43	--	mA	$V_{OL}=1.0V$
		3V	--	28	--		
I_{OL}	输出一般灌电流	5V	--	26	--	mA	$V_{OL}=1.0V$
		3V	--	16	--		
I_{IR}	红外输出灌电流	5V	--	43	--	mA	$V_{OL}=1.0V$
		3V	--	28	--		
I_{OP}	工作电流	正常模式					
		5V	--	1.7	--	mA	$F_{HOSC}=20MHz @ I_{HRC}/2$
		3V	--	0.7	--		

符号	参数	V _{DD}	最小值	典型值	最大值	单位	条件
		5V	--	1.4	--	mA	F _{HOSC} =20MHz @ I _{HRC} /4
		3V	--	0.5	--		
		5V	--	1.6	--	mA	F _{HOSC} =16MHz @ I _{HRC} /2
		3V	--	0.6	--		
		5V	--	1.3	--	mA	F _{HOSC} =16MHz @ I _{HRC} /4
		3V	--	0.5	--		
		5V	--	1.3	--	mA	F _{HOSC} =8MHz @ I _{HRC} /2
		3V	--	0.5	--		
		5V	--	1.1	--	mA	F _{HOSC} =8MHz @ I _{HRC} /4
		3V	--	0.4	--		
		5V	--	1.1	--	mA	F _{HOSC} =4MHz @ I _{HRC} /2
		3V	--	0.4	--		
		5V	--	1.0	--	mA	F _{HOSC} =4MHz @ I _{HRC} /4
		3V	--	0.4	--		
		5V	--	1.0	--	mA	F _{HOSC} =1MHz @ I _{HRC} /2
		3V	--	0.3	--		
		5V	--	1.0	--	mA	F _{HOSC} =1MHz @ I _{HRC} /4
		3V	--	0.3	--		
慢速模式							
		5V	--	11	--	uA	F _{HOSC} 关闭, F _{LOSC} =32KHz @ I _{LRC} /2
		3V	--	6.1	--		
		5V	--	7.3	--	uA	F _{HOSC} 关闭, F _{LOSC} =32KHz @ I _{LRC} /4
		3V	--	4.3	--		
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	--	2.6	--		
I _{HALT}	睡眠电流	5V	--	--	0.5	uA	睡眠模式, 关闭WDT
		3V	--	--	0.2		
		5V	--	--	5.0	uA	睡眠模式, 开启WDT
		3V	--	--	3.0		
R _{PH}	上拉电阻	5V	--	50	--	KΩ	上拉电阻(不包含PA5)
		3V	--	100	--		
		5V	--	85	--	KΩ	上拉电阻(PA5)
		3V	--	85	--		
R _{PL}	下拉电阻	5V	--	50	--	KΩ	下拉电阻
		3V	--	100	--		

6.3 OSC 特性

(测量条件VDD, TA温度等于程序条件)

参数	最小值	典型值	最大值	单位	条件
烧录座的I_HRC偏差			±1	%	烧录座直接安装在烧录器上。
烧录机台的I_HRC偏差			±3	%	正确设置烧录机台的条件。
烧录机台的I_LRC偏差			±5	%	

6.4 比较器/LVD电气特性

(VDD=5V, VSS=0V, 环境温度 TA=25°C)

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	--	us	F _{HOSC} =1MHz
I _{CO}	比较器电流消耗	--	135	--	uA	F _{HOSC} =1MHz, P2V模式
I _{LVD}	LVD电流消耗	--	150	--	uA	F _{HOSC} =1MHz, LVD=4.3V
E _{LVD}	LVD电压误差	--	--	3	%	F _{HOSC} =1MHz, LVD=4.3V

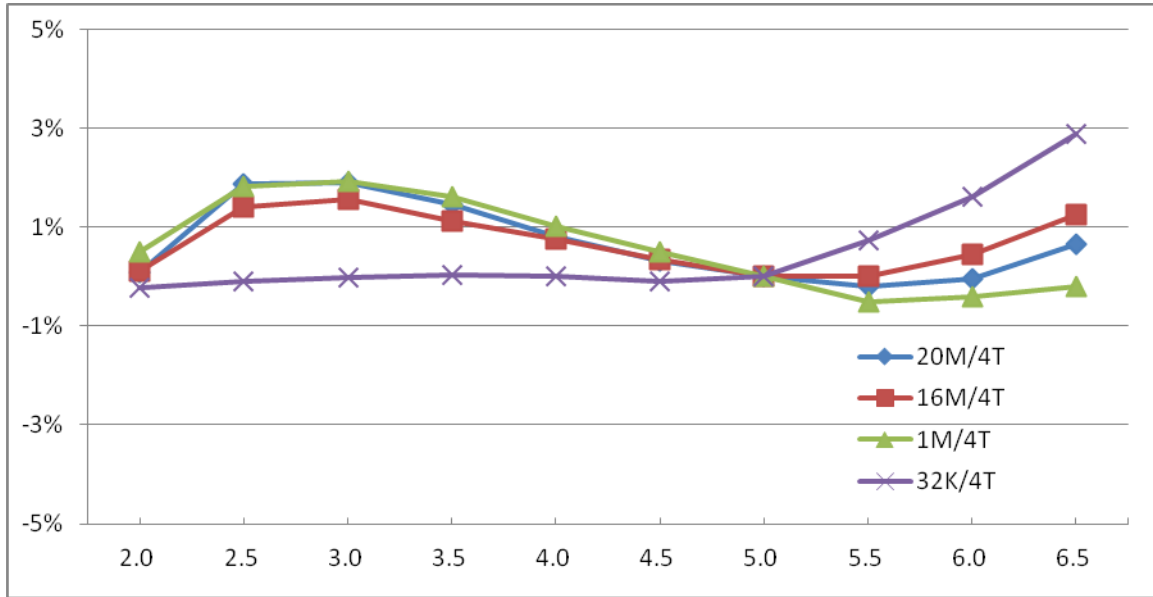
6.5 ADC电气特性

(VDD=5V, VSS=0V, 环境温度TA=25°C)

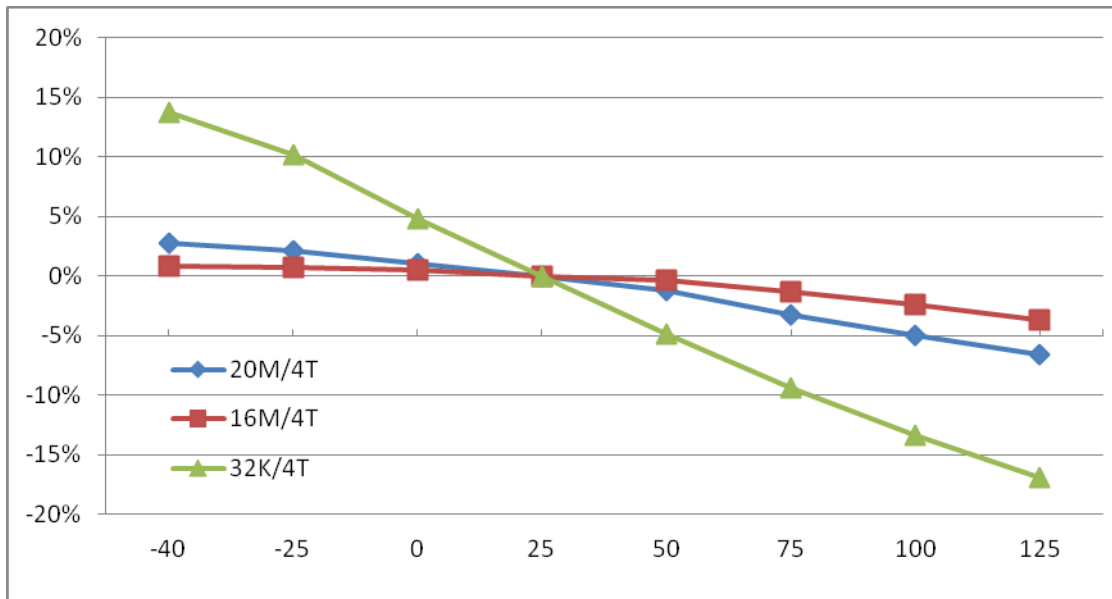
符号	参数	最小值	典型值	最大值	单位	条件
V _{REFH}	模拟参考电压范围	2V	--	V _{DD}	V	外部参考电压
V _{REF4}	内部 4V ADC V _{DD} =5V	3.96	4	4.04	V	--
V _{REF3}	内部 3V, V _{DD} =5V	2.97	3	3.03	V	--
V _{REF2}	内部 2V, V _{DD} =5V	1.98	2	2.02	V	--
V _{REF}	内部 V _{DD} , V _{DD} =5V	--	V _{DD}	--	V	--
--	ADC选择内部参考电压时, V _{DD} 最低电压水平	V _{REF} +0.5	--	--	V	最低电源电压
--	模拟通道输入电压	0	--	V _{REFH}	V	--
I _{OP(ADC)}	ADC工作电流	--	0.5	--	mA	--
ADCLK	ADC时钟	32K	--	1M	Hz	--
ADC _{CYCLE}	ADC转换时间周期	--	--		1/ADCLK	SHCLK=2 个ADC时钟
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, AVREFH=5V, ADC转换率=62.5K
INL	非线性积分误差	±2	--	--	LSB	
NMC	无缺码分辨率	10	11	12	位	

6.6 特性图

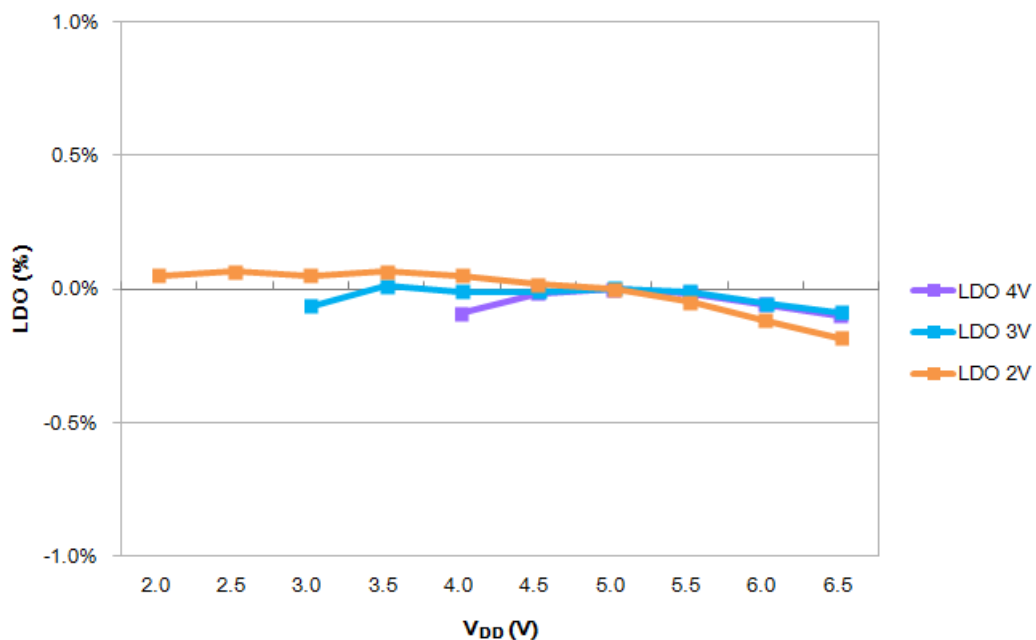
6.6.1 高速 RC 振荡频率与电源电压曲线图



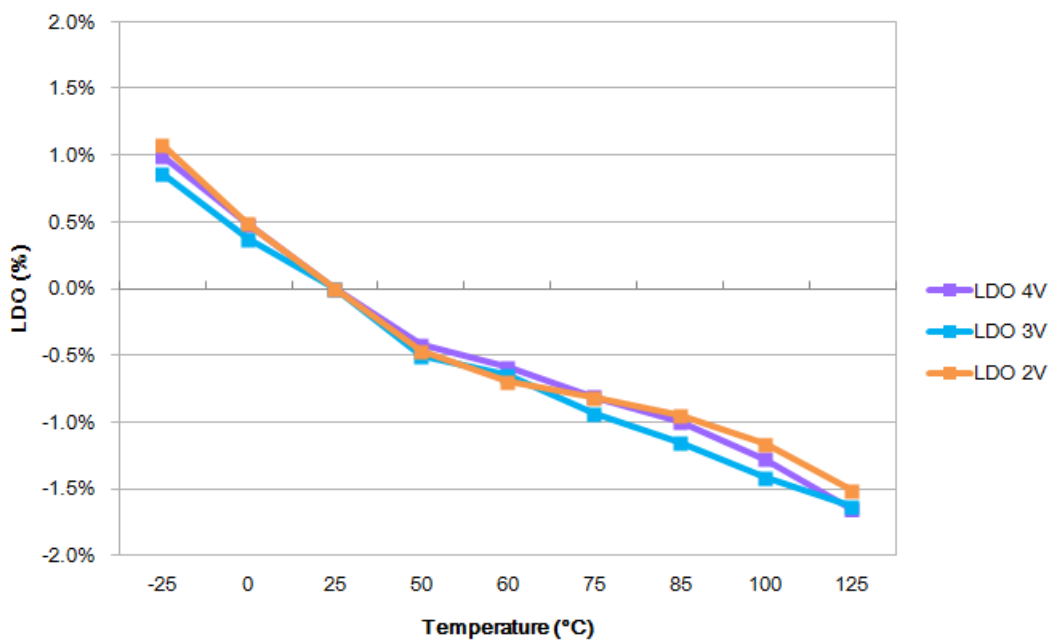
6.6.2 高速 RC 振荡频率与温度曲线图



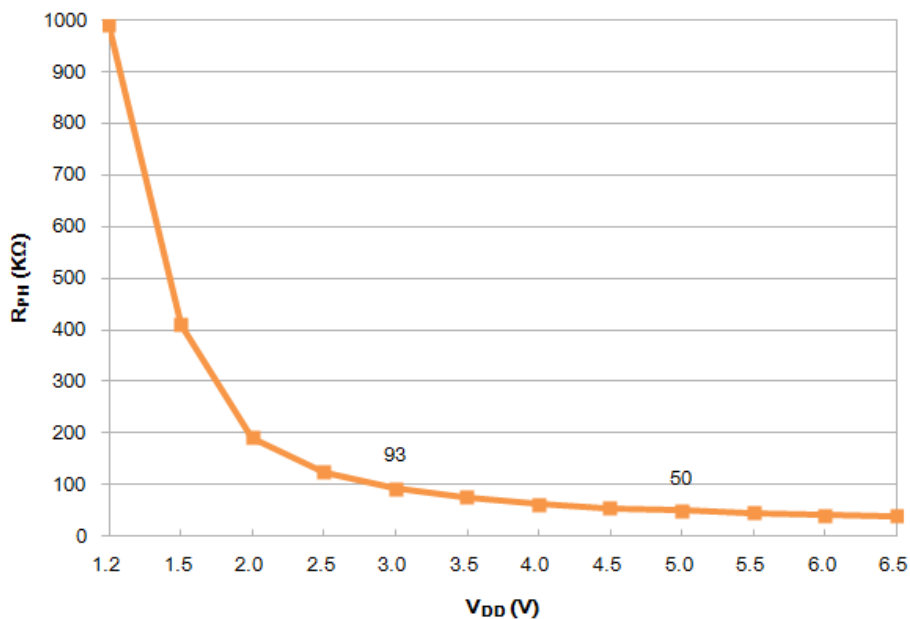
6.6.3 内部线性稳压器与电源电压曲线图



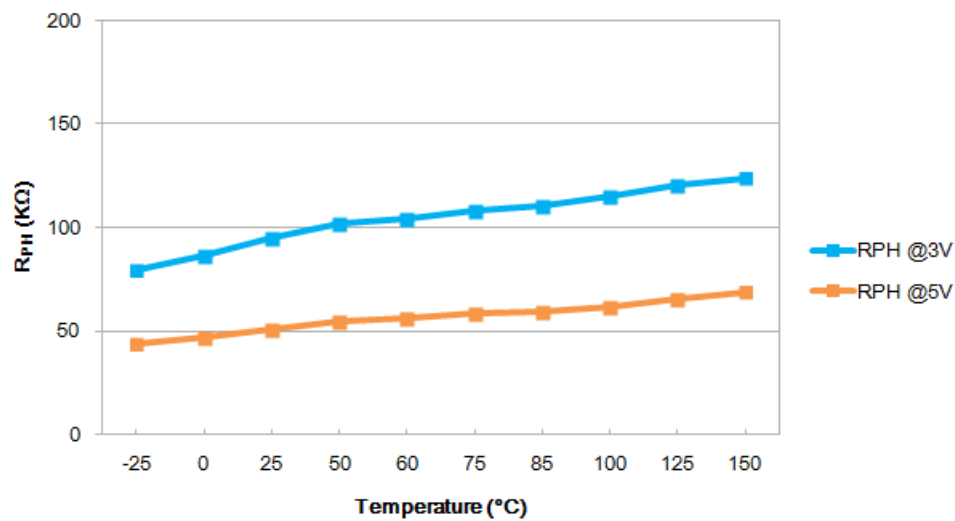
6.6.4 内部线性稳压器与温度曲线图



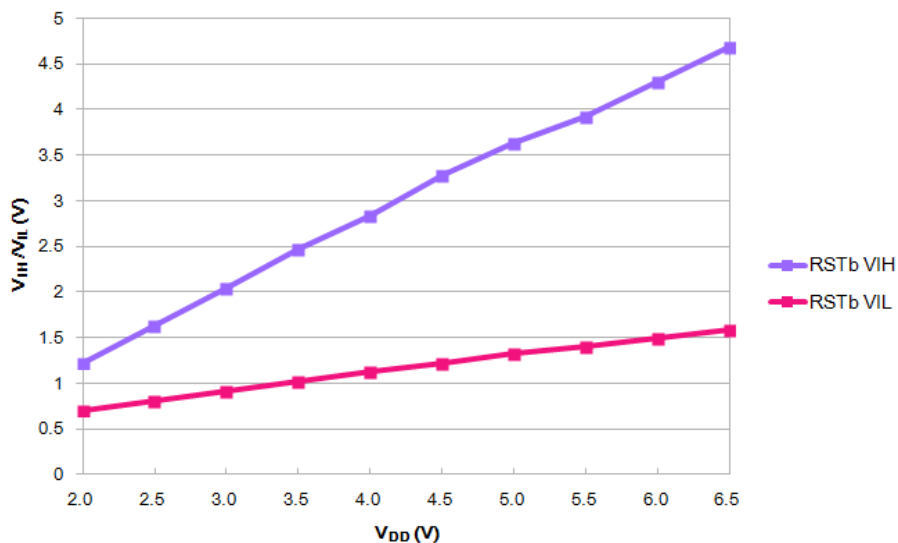
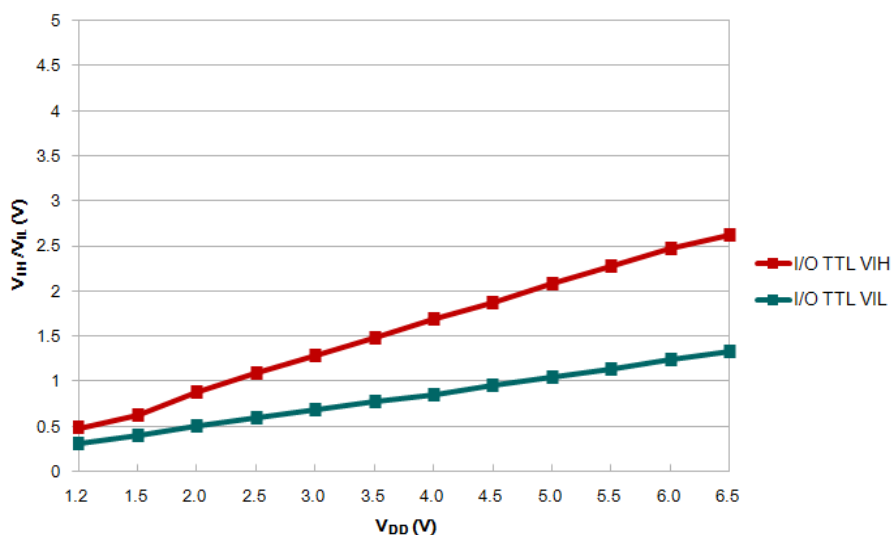
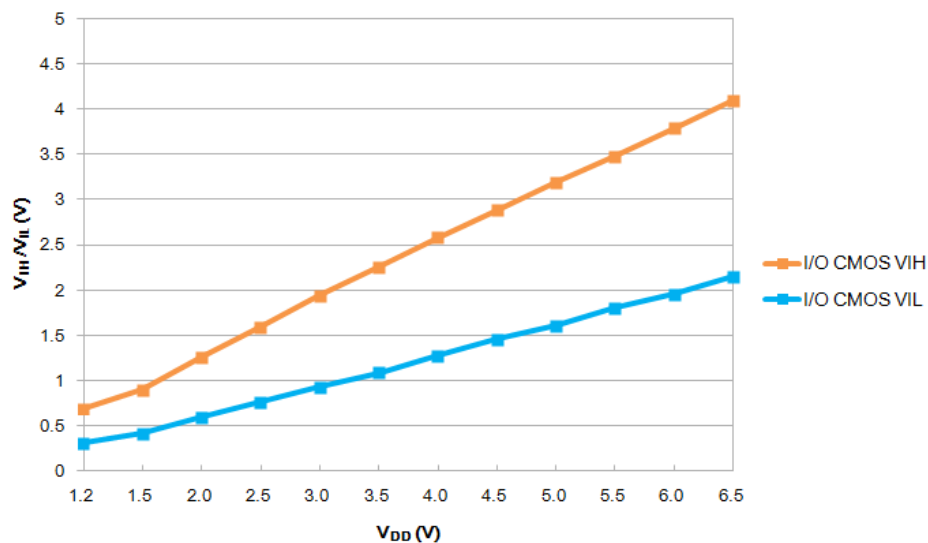
6.6.5 内部上拉电阻与电源电压曲线图



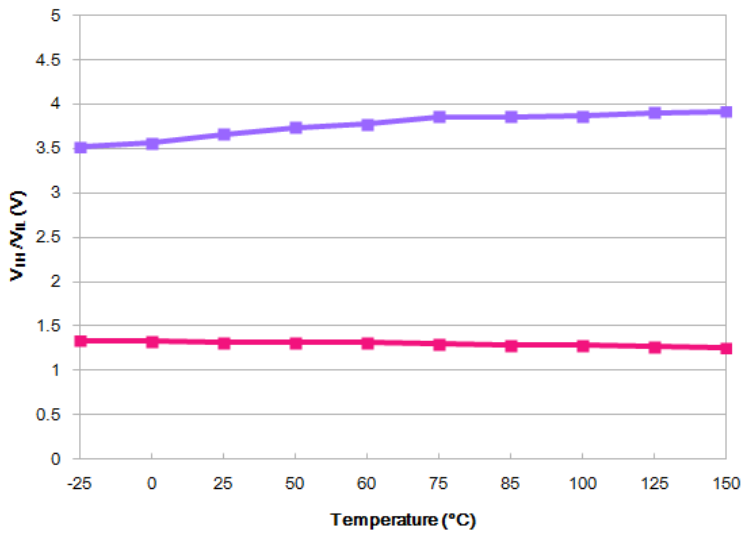
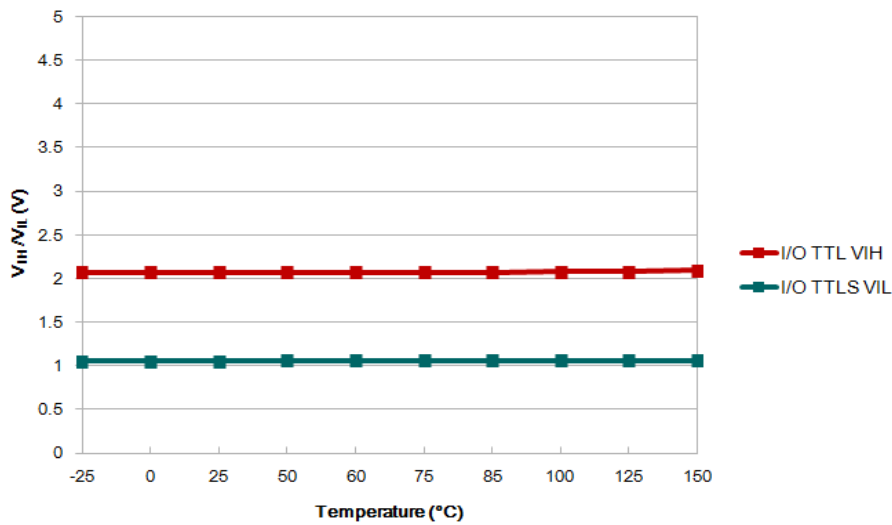
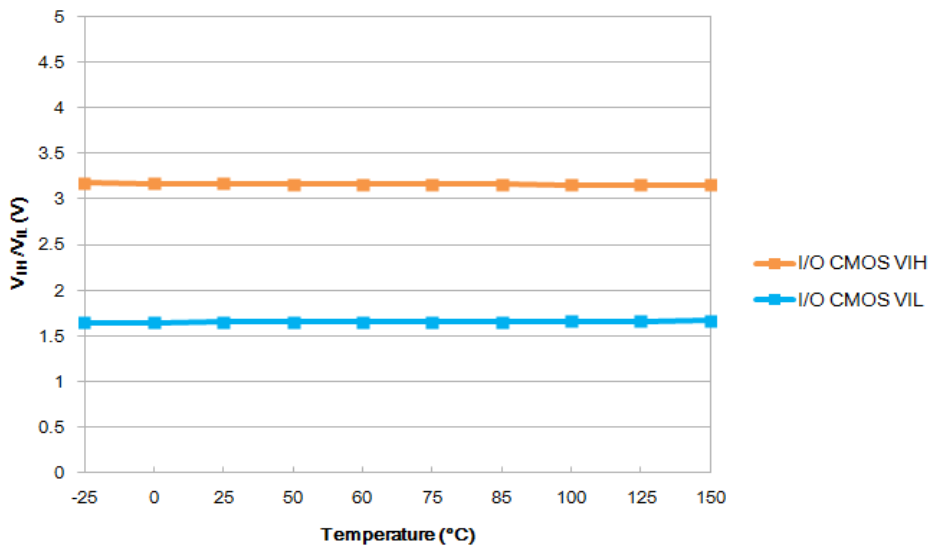
6.6.6 内部上拉电阻与温度曲线图



6.6.7 VIH/VIL 与电源电压曲线图



6.6.8 VIH/VIL 与温度曲线图

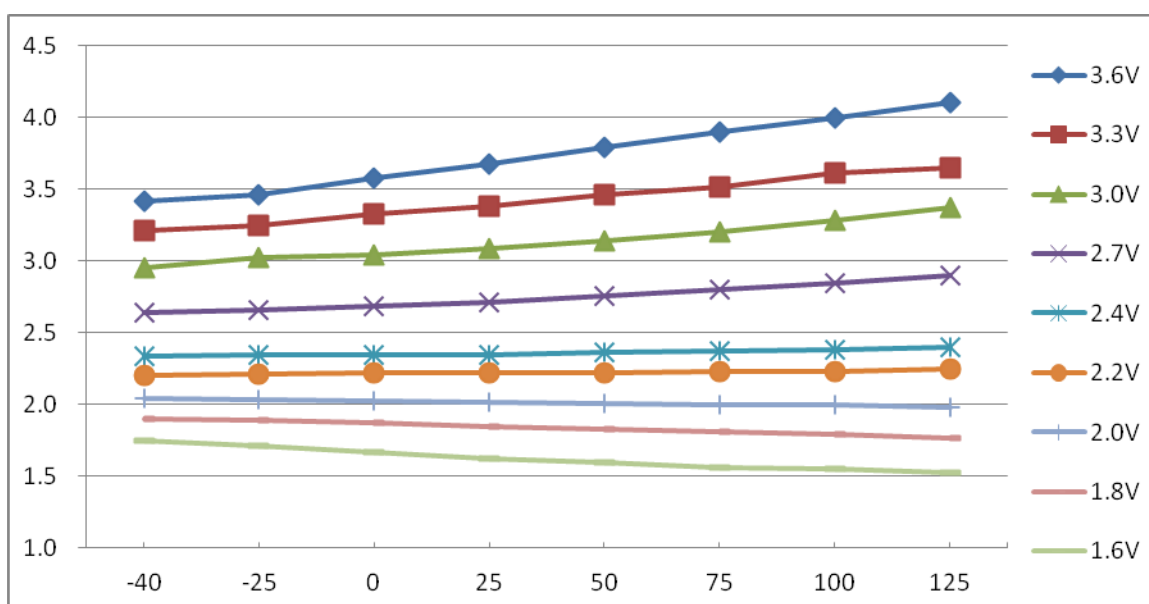


6.7 建议工作电压

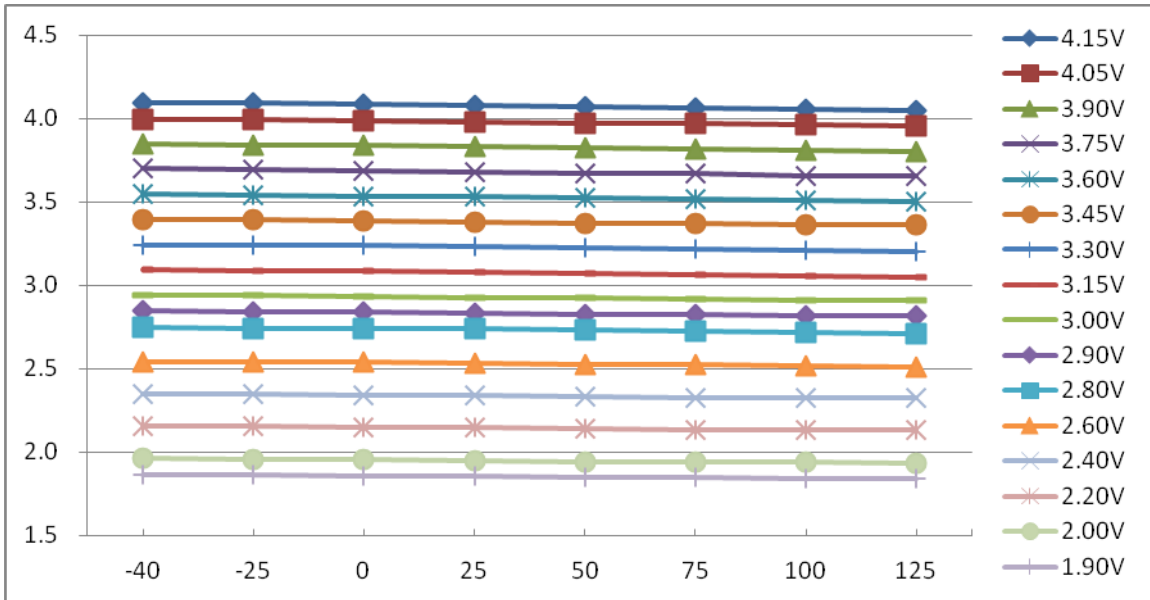
建议工作电压（温度范围：-40 °C ~ +85 °C）

频率	最小电压	最大电压	LVR: 默认值 (25 °C)	LVR: 建议值 (-40 °C ~ +85 °C)
20M/2T	3.3V	5.5V	3.6V	3.6V
16M/2T	3.0V	5.5V	3.0V	3.6V
20M/4T	2.2V	5.5V	2.4V	2.7V
16M/4T	2.0V	5.5V	2.2V	2.4V
8M(2T or 4T)	2.0V	5.5V	2.2V	2.4V
≤6M(2T or 4T)	1.8V	5.5V	2.0V	2.2V

6.8 LVR与温度曲线图

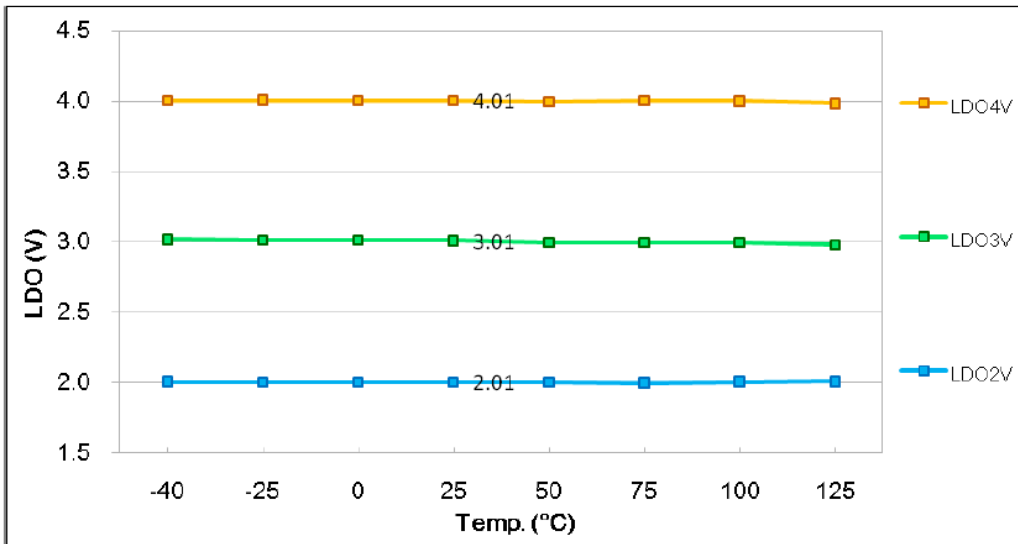


6.9 LVD与温度曲线图

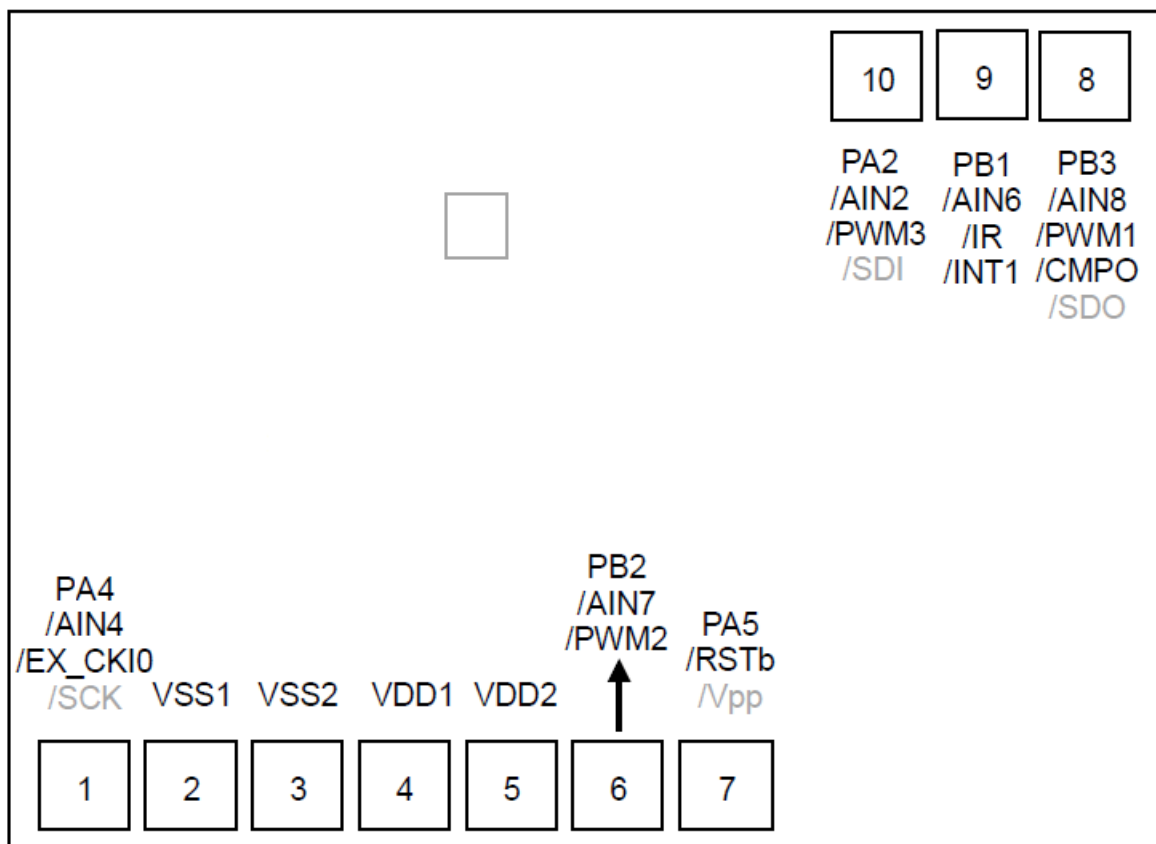


6.10 LDO与温度曲线图

LDO vs. Temperature



7. 芯片脚位坐标图



8. 封装尺寸

8.1 8 引脚SOP (150 毫寸)

	INCHES			MILLIMETERS		
	MIN	TYP	MAX	MIN	TYP	MAX
A	0.183	-	0.202	4.65	-	5.13
B	0.144	-	0.163	3.66	-	4.14
C	0.068	-	0.074	1.35	-	1.88
D	0.010	-	0.020	0.25	-	0.51
F	0.015	-	0.035	0.38	-	0.89
G	0.050 BSC			1.27 BSC		
J	0.007	-	0.010	0.19	-	0.25
K	0.005	-	0.010	0.13	-	0.25
L	0.189	-	0.205	4.80	-	5.21
M	-	-	8°	-	-	8°
P	0.228	-	0.244	5.79	-	6.20

注意：8-pin SOP 每管 100 颗。

9. 订购信息

产品名称	封装类型	引脚数	封装尺寸	配送方式
NY8B060DS8	SOP	8	150 mil	卷装：每卷 2.5K颗 管装：每管 100 颗