



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

数
据
手
册

NY8TM52D

6 I/O +4 个触控键的 8 位 MTP-Based 单片机

Version 1.0

Apr. 15, 2022

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	2022/04/15	初版。	-

目 录

1. 概述	7
1.1 功能.....	7
1.2 系统框图.....	9
1.3 引脚图.....	9
1.4 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>PortB</i> (PortB 数据寄存器).....	17
3.1.7 <i>PCON</i> (Power 寄存器).....	17
3.1.8 <i>BWUCON</i> (Port B 唤醒控制寄存器).....	17
3.1.9 <i>PCHBUF</i> (程序计数器高字节).....	18
3.1.10 <i>BPLCON</i> (Port B 下拉电阻控制寄存器).....	18
3.1.11 <i>BPHCON</i> (PortB 上拉电阻控制寄存器).....	18
3.1.12 <i>INTE</i> (中断使能寄存器).....	18
3.1.13 <i>INTF</i> (中断标志寄存器).....	19
3.1.14 <i>INTEDG</i> (外部中断控制寄存器).....	20
3.1.15 <i>TMRH</i> (定时器高字节寄存器).....	20
3.1.16 <i>ANAEN</i> (模拟电路寄存器).....	20
3.1.17 <i>PWM3RH</i> (PWM3 占空比寄存器).....	20
3.2 <i>T0MD</i> 定时器0控制寄存器.....	21
3.3 F-page 特殊功能寄存器.....	22
3.3.1 <i>IOSTB</i> (Port B I/O 控制寄存器).....	22
3.3.2 <i>PS0CV</i> (预分频器0寄存器).....	22
3.3.3 <i>BODCON</i> (Port B 开漏控制寄存器).....	22

3.3.4	CMPCR (比较器控制寄存器)	23
3.3.5	PCON1 (Power 控制寄存器 1)	23
3.4	S-page 特殊功能寄存器	24
3.4.1	TMR1 (定时器 1 寄存器)	24
3.4.2	T1CR1 (定时器 1 控制寄存器 1)	24
3.4.3	T1CR2 (定时器 1 控制寄存器 2)	25
3.4.4	PWM1DUTY (PWM1 占空比寄存器)	26
3.4.5	PS1CV (预分频器 1 寄存器)	26
3.4.6	BZ1CR (蜂鸣器 1 控制寄存器)	27
3.4.7	IRCR (IR 控制寄存器)	27
3.4.8	TBHP (表格指针高字节寄存器)	28
3.4.9	TBHD (表格数据高字节寄存器)	28
3.4.10	P2CR1 (PWM2 控制寄存器 1)	29
3.4.11	PWM2DUTY (PWM2 占空比寄存器)	29
3.4.12	OSCCR (振荡器控制寄存器)	29
3.4.13	P3CR1 (PWM3 控制寄存器 1)	30
3.4.14	PWM3DUTY (PWM3 占空比寄存器)	30
3.5	T-page 特殊功能寄存器	31
3.5.1	INTE3 (中断使能寄存器 3)	31
3.5.2	INTF3 (中断标志寄存器 3)	31
3.5.3	TPCKS (触控脚时钟寄存器)	31
3.5.4	CASR (触控外部电容选择寄存器)	32
3.5.5	TPCHS (触控脚通道选择寄存器)	32
3.5.6	TPCR (触控脚控制寄存器)	32
3.5.7	触控脚计数器低位寄存器	33
3.5.8	触控脚计数器高位寄存器	33
3.5.8	TPPADEN (触控脚使能寄存器)	33
3.6	I/O Port	34
3.6.1	IO 引脚结构框图	35
3.7	定时器 0	40
3.8	定时器 1/PWM1/Buzzer1	41
3.9	PWM2	44
3.10	PWM3	44
3.11	IR Carrier	45
3.12	低电压侦测 (LVD)	45
3.13	电压比较器	47
3.13.1	比较器参考电压	48
3.14	触控脚位	49

3.15	看门狗定时器 (WDT)	50
3.16	中断	51
3.16.1	Timer0 上溢中断	51
3.16.2	Timer1 下溢中断	51
3.16.3	看门狗超时中断	51
3.16.4	PB 输入状态改变中断	52
3.16.5	外部中断	52
3.16.6	LVD 低电压侦测中断	52
3.16.7	触控脚位比较器中断	52
3.16.8	触控脚位计数器上溢中断	52
3.17	振荡器配置	52
3.18	工作模式	53
3.18.1	Normal Mode	54
3.18.2	慢速模式	54
3.18.3	待机模式	54
3.18.4	睡眠模式	55
3.18.5	唤醒稳定时间	55
3.18.6	工作模式概述	56
3.19	复位	56
4.	指令设置	58
5.	配置字节表	74
6.	电气特性	75
6.1	最大绝对值	75
6.2	直流电气特性	75
6.3	特性图	77
6.3.1	高速 RC 振荡频率与电源电压曲线图	77
6.3.2	高速 RC 振荡频率与温度曲线图	77
6.3.3	低速 RC 振荡频率与电源电压曲线图	78
6.3.4	低速 RC 振荡频率与温度曲线图	78
6.3.5	上拉电阻与电源电压曲线图	79
6.3.6	上拉电阻与温度曲线图	79
6.3.7	输入高电平/输入低电平与电源电压曲线图	80
6.3.8	输入高电平/输入低电平与温度曲线图	81
6.4	建议工作电压	82
6.5	LVR 电压与温度曲线图	83
6.6	LVD 电压与温度曲线图	83

7.	芯片脚位坐标图	84
8.	封装尺寸	84
8.1	6 引脚 SOT23-6 (63 mil).....	84
8.2	8 引脚 SOP (150 mil).....	85
8.3	8 引脚 DIP (300 mil).....	85
9.	订购信息	86

1. 概述

NY8TM52D是以MTP作为程序记忆体的 8 位微控制器，适合IO产品的应用，例如触摸遥控器、风扇/灯光控制或是玩具周边等等。而MTP作为程序记忆体能更方便且有效率的开发产品。NY8TM52D核心建立在RISC精简指令集架构可以很容易地编辑和控制，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，可以让用户轻松地以程序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。

在I/O的资源方面，NY8TM52D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脚位都有附加的程序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，NY8TM52D内建了可选择频率的红外载波发射口。期中有 4 根I/O具备触控开关功能，可经由寄存器控制为触控开关输入脚。NY8TM52D内建电压比较器，也可由寄存器选择I/O当电压输入与比较结果输出。

NY8TM52D有两组计时器，可用系统频率当作一般的计时的应用或者从外部信号触发来计数。另外NY8TM52D提供 3 组 10 位解析度的PWM输出及一组蜂鸣器输出，可用来驱动马达、LED或蜂鸣器等等。

NY8TM52D采用双时钟机制，高速振荡或者低速振荡都由内部RC振荡输入。在双时钟机制下，NY8TM52D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。

在省电模式下如待机模式(Standby mode)与睡眠模式(Halt mode)中，有多种中断实践触发NY8TM52D进入正常模式(Normal) 或 低速模式(Slow mode) 来处理中断事件。

在省电的模式下如待机模式(Standby mode)与睡眠模式(Halt mode)中，有多种事件可触发中断唤醒NY8TM52D进入正常操作模式(Normal) 或慢速模式(Slow mode) 来处理突发事件。

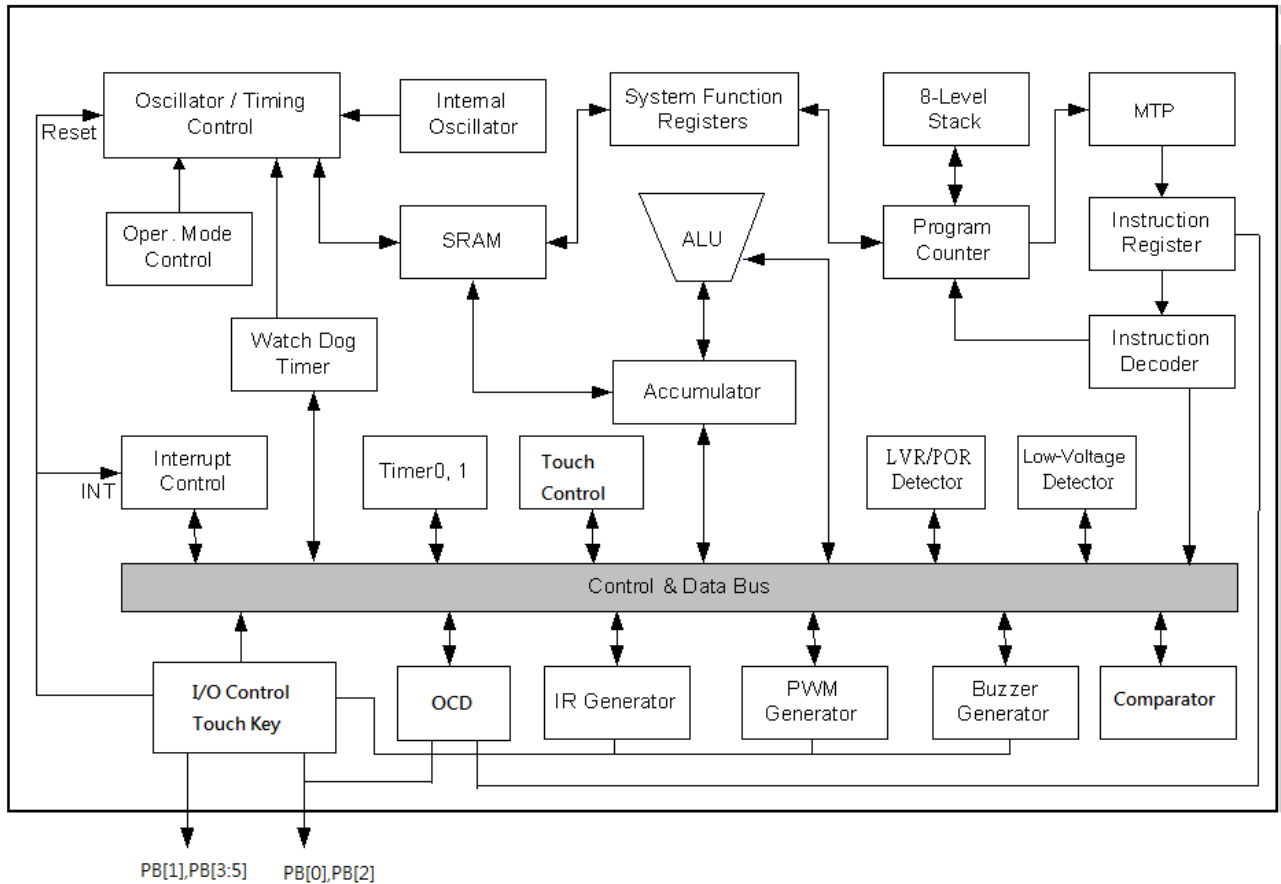
1.1 功能

- 宽广的工作电压：
 - 2.4V ~ 5.5V @系统频率 ≤ 8MHz。
 - 3.0V ~ 5.5V @系统频率 > 8MHz。
- 宽广的工作温度：-40°C ~ 85°C。
- 2Kx14 bits MTP
- 128 bytes SRAM。
- 6 根可分别单独控制输入输出方向的I/O脚(GPIO)、PB[5:0]。
- PB[3:0]可选择输入时使用内建下拉电阻。
- PB[5:0]可选择上拉电阻或开漏极输出可 (Open-Drain)。
- 8 层程序堆栈(Stack)。
- 存取数据有直接或间接定址模式。
- 一组 8 位上数计时器(Timer0)包含可程序化的频率预除线路。
- 一组 10 位下数计时器(Timer1) 可选重复载入或连续下数计时。

- 三個 10 位元的脈衝寬度調變輸出(PWM1/2/3)。 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红外线载波频率可供选择，同时载波极性也可根据数据作选择。
- 内建上电复位电路(POR)。
- 内建低压复位功能(LVR)。
- 内建 16 段低电压侦测功能(LVD)。
- 内建准确的电压比较器(Voltage Comparator)。
- 内建 4 键触控输入脚。
- 内建看门狗计时(WDT)，可由程序软件控制开关。
- 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： I_HRC (1~20MHz内部高速RC振荡)
 - 低速振荡： 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)。
- 八种硬件中断：
 - Timer0 溢位中断。
 - Timer1 借位中断。
 - WDT 中断。
 - PB 输入状态改变中断。
 - 外部中断输入。
 - 低电压侦测中断。
 - 触控比较器中断。
 - 触控计数器溢位中断。
- NY8TM52D在待机模式(Standby mode)下的八种唤醒中断：
 - Timer0 溢位中断。
 - Timer1 借位中断。
 - WDT中断。
 - PB 输入状态改变中断。
 - 外部中断输入。
 - 低电压侦测中断。
 - 触控比较器中断。
 - 触控计数器溢位中断。
- NY8TM52D在睡眠模式(Halt mode)下的五种唤醒中断：
 - WDT 中断。
 - PB 输入状态改变中断。

- 外部中断输入。
 - 触控比较器中断。
 - 触控计数器溢位中断。
- NY8TM52D提供两种电流档位 (Normal/Large)。

1.2 系统框图



1.3 引脚图

NY8TM52D提供三种封装类型：SOP8，DIP8 及 SOT23-6。

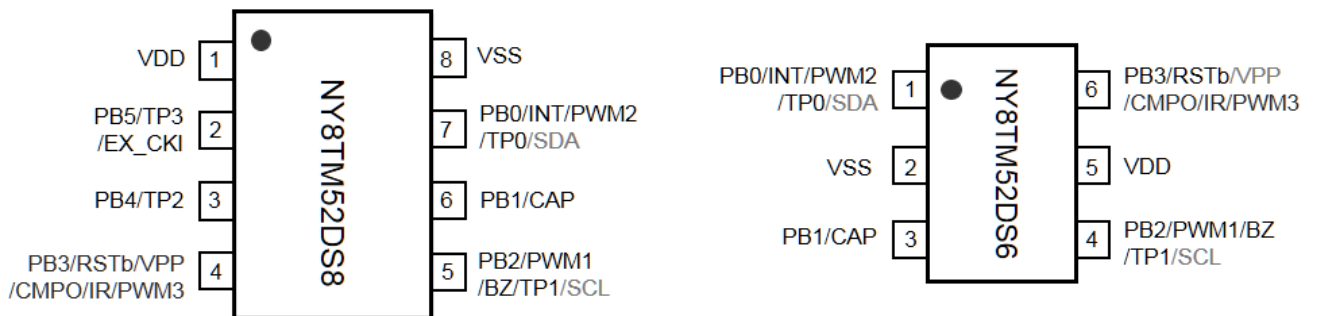


图 1 封装引脚图

1.4 引脚说明

引脚名称	I/O	描述
PB0 / INT / PWM2 / TP0 / SDA	I/O	PB0 是一个双向I/O引脚，也可当作比较器输入引脚，触控脚。 当EIS=1 和 INTIE=1 时，PB0 是外部中断输入引脚。 PB0 可输出PWM2。 TP0 是触控脚位 0。 PB0 也是编程数据输入SDA。
PB1 /CAP	I/O	PB1 是一个双向I/O引脚，也可作外部电容触控脚。 CAP是外部电容触控脚。
PB2 / PWM1 / BZ1 / TP1 / SCL	I/O	PB2 是一个双向I/O引脚，也可当作比较器输入引脚，触控脚。 PB2 可输出PWM1。 PB2 可输出BUZZER1。 TP1 是触控脚位 1。 PB2 也是编程数据输入SCL。
PB3 / RSTb /CMPO /VPP /IR /PWM3	I/O	PB3 是一个双向I/O引脚。 PB3 可作复位引脚RSTb，如果RSTb为低电平时，NY8TM52D会复位。 PB3 可作比较器输出脚CMPO。 当PB3 引脚的电压高于 9.8V时，NY8TM52D会进入MTP编程模式。 如果IR模式开启，PB3 可作IR载波输出。 PB3 可输出PWM3。
PB4 /TP2	I/O	PB4 是一个双向I/O引脚，也可当作比较器输入引脚，触控脚。 PB4 可作指令时钟输出脚。 TP2 是触控脚位 2。
PB5 /TP3 / EX_CK1	I/O	PB5 是一个双向I/O引脚，也可当作比较器输入引脚，触控脚。 TP3 是触控脚位 3。 PB5 可作定时器时钟源EX_CK1。
VDD	-	电源正端。
VSS	-	电源负端。

2. 内存结构

NY8TM52D存储器分为两类：分别是程序存储器和数据存储器。数据存储器又细分为SRAM和Register 存储器。

2.1 程序存储器

NY8TM52D程序存储器空间是 2Kx14 位。因此，11 位宽的计数器（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。

NY8TM52D提供GOTOA和CALLA等指令去访问程序空间的 256 个地址。还提供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。

NY8TM52D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。

NY8TM52D程序存储器地址 0x00E~0x00F是预置的滚码，可用来释放和用作普通程序空间。

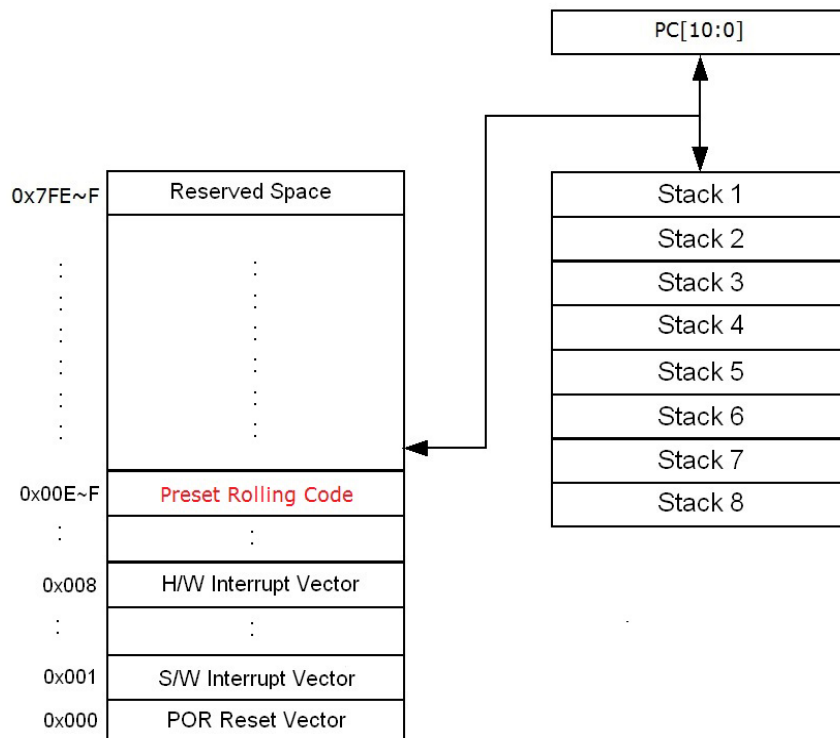


图 2 程序存储器对映地址

2.2 数据存储

根据用于存取数据存储器的指令，数据存储器可分为四类：一个是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、T-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。

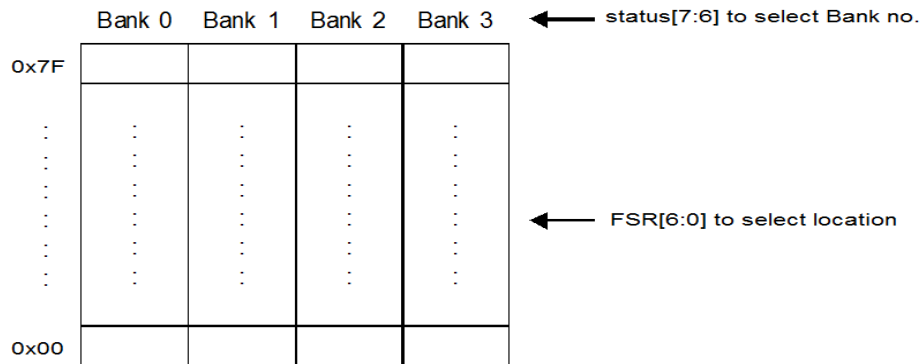


图 3 间接寻址方式存取数据存储器

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。

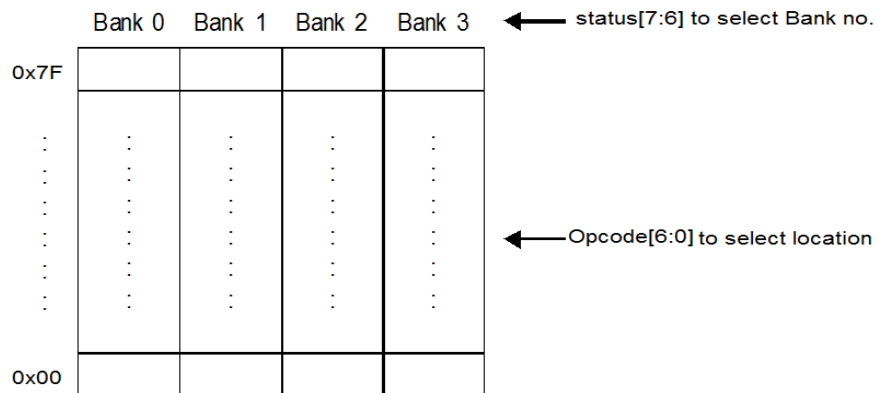


图 4 直接寻址方式存取数据存储器

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Bank0数据存储器的0x20到0x7F地址与Bank1的0x20到0x3F地址，其他bank的0x20到0x7F址映像如表1所示。

NY8TM52D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。

FSR[7:6] Address	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	-			
0x6	PORTB			
0x7	-			
0x8	PCON			
0x9	BWUCON			
0xA	PCHBUF			
0xB	BPLCON			
0xC	BPHCON			
0xD	-			
0xE	INTE			
0xF	INTF			
0x10 ~ 0x17	-			
0x18	INTEDG			
0x19	TMRH			
0x1A	ANAEN			
0x1B				
0x1C	PWM3RH			
0x1D ~ 0x1F	-			
0x20 ~ 0x3F	通用寄存器	通用寄存器	映射至Bank0	映射至Bank1
0x40 ~ 0x7F	通用寄存器	映射至Bank0	映射至Bank0	映射至Bank0

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和T-page的地址说明如下表。

特殊功能寄存器种类	F-page SFR	S-page SFR	T-page SFR
0x0	-	TMR1	
0x1	-	T1CR1	
0x2	-	T1CR2	
0x3	-	PWM1DUTY	
0x4	-	PS1CV	
0x5	-	BZ1CR	
0x6	IOSTB	IRCR	
0x7	-	TBHP	INTE3
0x8	-	TBHD	INTF3
0x9	-	-	TPCKS
0xA	PS0CV	P2CR1	CASR
0xB	-	-	TPCHS
0xC	BODCON	PWM2DUTY	TPCR
0xD	-	-	TPCNTL
0xE	CMPCR		TPCNTH
0xF	PCON1	OSCCR	TPPADEN
0x10	-	-	-
0x11	-	P3CR1	-
0x12	-	-	-
0x13	-	PWM3DUTY	-
0x14 ~ 0x1F	-	-	-

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

3. 功能概述

本章节将详细描述NY8TM52D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_CK1或低频震荡器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。

以LGOTO指令来说，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。

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。

GP5: 通用寄存器数据位。

(*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	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 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	GP7	GP6	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。

GP7~6: 通用寄存器数据位。

3.1.7 PCON (Power 寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	R	0x8	WDTEN	GP6	LVDEN	GP4	LVREN	GP2	GP1	GP0
读写属性			读/写							R
初始值			1	0	0	1	1	0	0	0

GP6,4,2,1,0: 通用寄存器数据位。

LVREN: 开启/关闭 LVR。

LVREN=1 时, 开启LVR。

LVREN=0 时, 关闭LVR。

LVDEN: 开启/关闭 LVD。

LVDEN=1 时, 开启LVD。

LVDEN=0 时, 关闭LVD。

WDTEN: 开启/关闭 WDT。

WDTEN=1 时, 开启WDT。

WDTEN=0 时, 关闭WDT。

3.1.8 BWUCON (Port B 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	1	1	1	1	1	1

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。

3.1.9 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[2:0]		
读写属性			-	-	-	-	-	读/写		
初始值			X	X	X	X	X	000		

PCHBUF[2:0]: 程序计数器PC的第十个位到第八个位。

3.1.10 BPLCON (Port B 下拉电阻控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BPLCON	R	0xB	/PLPB3	/PLPB2	/PLPB1	/PLPB0	-	-	-	-
读写属性			读/写	读/写	读/写	读/写	-	-	-	-
初始值			1	1	1	1	X	X	X	X

/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。

3.1.11 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。

3.1.12 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	-	WDTIE	-	LVDIE	T1IE	INTIE	PBIE	TOIE
读写属性			-	读/写	-	读/写	读/写	读/写	读/写	读/写
初始值			X	0	X	0	0	0	0	0

TOIE: 定时器 0 上溢 (overflow) 中断使能位。

TOIE=1 时, 开启定时器 0 上溢中断。

TOIE=0 时, 关闭定时器 0 上溢中断。

PBIE: PortB输入状态变化中断使能位。

PBIE=1 时, 开启PortB输入状态变化中断。

PBIE=0 时, 关闭PortB输入状态变化中断。

INTIE: 外部中断使能位。

INTIE=1 时，开启外部中断。

INTIE=0 时，关闭外部中断。

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。

3.1.13 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	-	WDTIF	-	LVDIF	T1IF	INTIF	PBIF	T0IF
读写属性			-	读/写	-	读/写	读/写	读/写	读/写	读/写
初始值(note*)			X	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。

PBIF: Port B输入状态变化中断标志位。

PBIF=1 时，发生Port B输入状态变化中断。

PBIF必须由程序清零。

INTIF: 外部中断标志位。

INTIF=1 时，发生外部中断。

INTIF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。

注意: INTE中断Bit未使能，相应的中断标志位为0。

3.1.14 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	-	-	-	EIS	-	-	INTG1	INTG0
读写属性			-	-	-	读/写	-	-	读/写	读/写
初始值			X	X	X	0	X	X	0	1

INTG1~0: INT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。

EIS: 外部中断引脚选择位。

EIS=1 时, PB0 选择为外部中断引脚。

EIS=0 时, PB0 选择为GPIO。

3.1.15 TMRH（定时器高字节寄存器）

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。

PWM1DUTY9~8: PWM1 占空比高 2 位。

PWM2DUTY9~8: PWM2 占空比高 2 位。

3.1.16 ANAEN（模拟电路寄存器）

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	0	X	X	X	X	X	X

COMPEN: 开启/关闭电压比较器。

COMPEN=1, 开启电压比较器。

COMPEN=0, 关闭电压比较器。

3.1.17 PWM3RH（PWM3 占空比寄存器）

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

PWM3RH	R	0x1C	-	-	-	-	-	-	PWM3DUTY9	PWM3DUTY8
读写属性			-	-	-	-	-	-	读/写	读/写
初始值			X	X	X	X	X	X	X	X

PWM3DUTY9~8: PWM3 占空比数据的最高有效位是 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 3 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: 在开启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_CK1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_CK10 脚或低频震荡I_LRC/E_LXT。

T0CS=0 时，选择指令时钟 F_{INST} 。

LCKTM0: 当T0CS=1 时，定时器 0 时钟源可随意选择低频振荡器，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_CKI脚被选择当作定时器 0 时钟源。

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

GP6: 通用读写器寄存器。

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 IOSTB (Port B 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。

3.3.2 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 器的目前计数值。

3.3.3 BODCON (Port B 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	GP7	GP6	ODPB5	ODPB4	ODPB3	ODPB2	ODPB1	ODPB0
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

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。

GP7~6: 通用寄存器数据位。

3.3.4 CMPCR (比较器控制寄存器)

名称	SFR 类型	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPCR	F	0xE	-	RBIAS_H	RBIAS_L	CMP_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	PB2
01	PB5
10	频带间隙 (0.6V)
11	Vref

PS[1:0]: 比较器正输入源选择。

PS[1:0]	正输入源
00	PB0
01	PB4
10	Vref
11	---

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: 设置相应参考电压。

3.3.5 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	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	0	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。

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。

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。

LVDOUT: LVD输出位，只读。

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
1101	3.90V
1110	4.05V
1111	4.15V

表 4 LVD 电压选择

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[9:0]一起写到定时器 1 重载缓存器中。

3.4.2 T1CR1 (定时器 1 控制寄存器 1)

名称	SFR 类型	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T1CR1	S	0x1	PWM1OEN	PWM1OAL	-	-	-	T1OS	T1RL	T1EN

读写属性	W	W	-	-	-	读/写	读/写	读/写
初始值	0	0	X	X	X	0	0	0

This register is used to configure Timer1 functionality.

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 5 定时器 1 功能

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，PB2 输出PWM1。

PWM1OEN=0，PB2 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]	Dividing Rate
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 6 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/开启预除器 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_CKI脚下降沿时定时器 1 减一。

T1CE=0 时，EX_CKI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_CKI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			W							
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			R							
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	-	-	-	W			
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 7 蜂鸣器BZ1 输出 (PB2) 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			-	-	-	-	-	W	W	W
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 3.14) 可以当作IR时钟源。

2. 不同振荡类型的分频比。IRF57K=1, IR载波频率是 $4MHz/64 = 62.5KHz$ 。

OSC. 类型	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 8 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			-	-	R	R	R	R	R	R
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	-	-	TM1HRC	-	-	-
读写属性			W	W	-	-	W	-	-	-
初始值			0	0	X	X	0	X	X	X

此寄存器用于配置PWM2 的功能。

TM1_HRC: 定时器 1 时钟源选项。

TM1HRC=1, PWM1, 2, 3 & Timer 1 时钟源是高频振荡时钟。

TM1HRC=0, PWM1, 2, 3 & Timer 1 时钟源选项取决于T1CS寄存器。

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 时, PB0 输出PWM2。

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			W							
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	

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。

SELHOSC: 系统振荡器选择 (F_{osc})。

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

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

STPHOSC: 关闭/开启高频率震荡器 (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 9 选择OPMD[1:0]的操作模式

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			W	W	-	-	-	-	-	-
初始值			0	0	X	X	X	X	X	X

此寄存器用于配置PWM3 的功能。

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 时, PB3 输出PWM2。

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			W							
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, TM3RH[1:0]与PWM3DUTY[7:0]寄存器用于定义PWM3 的占空比。

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	GP7	-	-	-	-	-	TPOVIE	TPCPIE
读写属性			读/写	-	-	-	-	-	读/写	读/写
初始值			0	0	0	0	0	0	0	0

GP7: 通用寄存器数据位。

TPOVIE: 触控计数器上溢中断使能位。

1 = 开启触控计数器上溢中断。

0 = 关闭触控计数器上溢中断。

TPCPIE: 触控比较器中断使能位。

1 = 开启触控比较器中断。

0 = 关闭触控比较器中断。

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	-	-	-	-	-	-	TPOVIF	TPCPIF
读写属性			-	-	-	-	-	-	读/写	读/写
初始值			0	0	0	0	0	0	0	0

TPOVIF: 触控脚计数器上溢中断标志位。

1 = 触控脚计数器上溢中断发生。

0 = 触控脚计数器上溢中断未发生。

TPCPIF: 触控脚比较器中断标志位。

1 = 触控脚比较器中断发生。

0 = 触控脚比较器中断未发生。

写入 TPCR 或 TPCHS, 将会清除 TPOVIF 和 TPCPIF

3.5.3 TPCKS (触控脚时钟寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TPCKS	T	0x9	-	-	-	WKUPT	-	TPCK2	TPCK1	TPCK0
读写属性			-	-	-	读/写	-	读/写		
初始值			0	0	0	0	0	0	1	0

WKUPT: 触控慢速模式唤醒时间选项。

1= 16Hz (64ms)

0= 32Hz (32ms)

TPCK2~TPCK0: 触控调变时钟选项。

100: 1.12MHz

011: 1.31MHz
 010: 1MHz
 001: 0.88MHz
 000: 0.75MHz

3.5.4 CASR (触控外部电容选择寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TPCAXS	T	0xA	-	CA[6:0]						
读写属性			读/写							
初始值			0	0	0	0	0	0	0	0

CA6~CA0: 外部电容选项。

1111 1111 : 127*C array unit
 1111 1110 : 126*C array unit
 0000 0001: 1*C array unit
 0000 0000: 0*C array unit

注意: C array unit = 0.05pF

3.5.5 TPCHS (触控脚通道选择寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TPCHS	T	0xB	-	-	-	-	-	CHS2	CHS1	CHS0
读写属性			-	-				读/写		
初始值			0	0	0	0	0	0	0	0

CHS2~CHS0: touch pad channel select 触控脚通道选项。

111 = 选择所有触控脚。
 110 = 选择指定按键。
 100~101: 预留。
 011 = 选择 TP3
 010 = 选择 TP2
 001 = 选择 TP1
 000 = 选择 TP0

3.5.6 TPCR (触控脚控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TPCR	T	0xC	-	-	-	-	-	TPMD2	TPMD1	TPMD0
读写属性			-					读/写		
初始值			X	X	X	X	X	0	0	0

TPMD2~TPMD0 触控脚运算模式

111~110:预留。

- 101: 慢速扫描模式
- 100 = 预留。
- 011 = Cs 电容放电
- 010 = TPCHOFF 所有通道都不选择
- 001 = TPRUN 扫描 TPCHS 选择的通道
- 000 = TPSTP 停止所有触控脚

当更换触控键或清除 TPINT 标志（TPOVIF，TPCPIF）后，TPMD 将会回到 TPCHOFF 模式。

3.5.7 触控脚计数器低位寄存器

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TPCNTL	T	0xD	TPCNT7	TPCNT6	TPCNT5	TPCNT4	TPCNT3	TPCNT2	TPCNT1	TPCNT0
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

R:触控计数器低位数据。

W:重载触控计数器低位数据。

3.5.8 触控脚计数器高位寄存器

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TPCNTH	T	0xE	-	-	-	-	TPCNT11	TPCNT10	TPCNT9	TPCNT8
读写属性			-				读/写	读/写	读/写	读/写
初始值			X	X	X	X	1	1	1	1

R:触控计数器高位数据。

W:重载触控计数器高位数据。

TPCNT<11:0>:当地址被读到时，该数据就是触控脚计数器的值。

3.5.8 TPPADEN（触控脚使能寄存器）

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TPPADEN	T	0xF	-	-	-	-	TP3EN	TP2EN	TP1EN	TP0EN
读写属性			-				读/写	读/写	读/写	读/写
初始值			X	X	X	X	0	0	0	0

TPPADEN<3:0>: I/O 口是触控脚或不是。

TP0EN=1 PB0 当作 TP0, TPEN0=0 PB0 为 I/O。

TP1EN=1 PB2 当作 TP1, TPEN1=0 PB2 为 I/O。

TP2EN=1 PB4 当作 TP2, TPEN2=0 PB4 为 I/O。

TP3EN=1 PB5 当作 TP3, TPEN3=0 PB5 为 I/O。

3.6 I/O Port

NY8TM52D提供 6 个I/O口 (PB[5:0])，用户可以由寄存器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IOSTB[5:0]定义PB[5:0]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BPHCON[5:0]用于开启或关闭PB[5:0]的内部上拉电阻。寄存器BPLCON[7:4]则是用于开启或关闭PB[3:0]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[5:0]决定PB[5:0]是否为开漏输出脚。

I/O口功能摘要如下表：

功能		PB[3:0]	PB[5:4]
输入	上拉电阻	V	V
	下拉电阻	V	X
输出	开漏	V	V

表 10 I/O端口功能摘要

在PB的每个I/O脚都有输入状态改变产生中断功能。寄存器BWUCON[5:0]会使能或禁用任一PB脚位的唤醒功能。只要 BWUCON对应到的任一PB脚位被置为 1 时，且在此输入脚位有状态改变时，寄存器PBIF (INTF[1]) 就会被设为 1。如果寄存器PBIE (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。

NY8TM52D提供 1 个外部中断，当寄存器EIS (INTEDG[4]) 设定为 1，PB0 则被当作外部中断的输入脚。

注意：当PB0 同时设定成输入状态改变触发脚与外部中断脚，外部中断有较高的优先权，而PB0 输入状态改变触发脚则会被关闭，但其它输入状态改变触发脚不会被影响。

NY8TM52D提供红外线IR载波生成器。IR载波生成器是由寄存器IREN (IRCR[0]) 开启，PB3 会输出红外线载波。

由配置字节决定PB3 可否当作外部复位输入RSTb。当PB3 为低电平时将导致NY8TM52D发生复位。

当NY8TM52D处于正常模式或待机模式时，如果配置字节开启则PB4 检测到指令时钟。

除此之外，如果T0MD T0CS=1 和LCK_TM0=0 时，PB5 可作定时器 0 的外部时钟源EX_CKI。如果T1CS=1，PB5 可作定时器 1 的外部时钟源。

如果寄存器PWM1OEN (T1CR1[7]) 为 1，PB2 可以当作脉冲宽度调制PWM输出。若寄存器BZ1EN (BZ1CR[7]) 为 1，PB2 也可以当作蜂鸣器输出。

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。

OD_EN: 开启开漏功能。

PULLUP_ENB: 开启内部上拉电阻。

PULLDOWN_EN: 开启内部下拉电阻。

RD_类型: 选择读取脚位或数据锁存器。

SFEN: 开启特殊功能 (PWM2)。

SFDT: 特殊功能数据。

EIS: 开启外部中断功能。

INTEDGE: 外部中断边沿选择。

EX_INT: 外部中断信号。

WUB: 开启PortB 唤醒功能。

SET_PBIF: 设置PortB 唤醒标志。

CMPEN: 开启为比较器输入引脚。

TPEN: 开启为触控键输入引脚。

CMPVP: 比较器正输入引脚。

TPIN: 触控键输入。

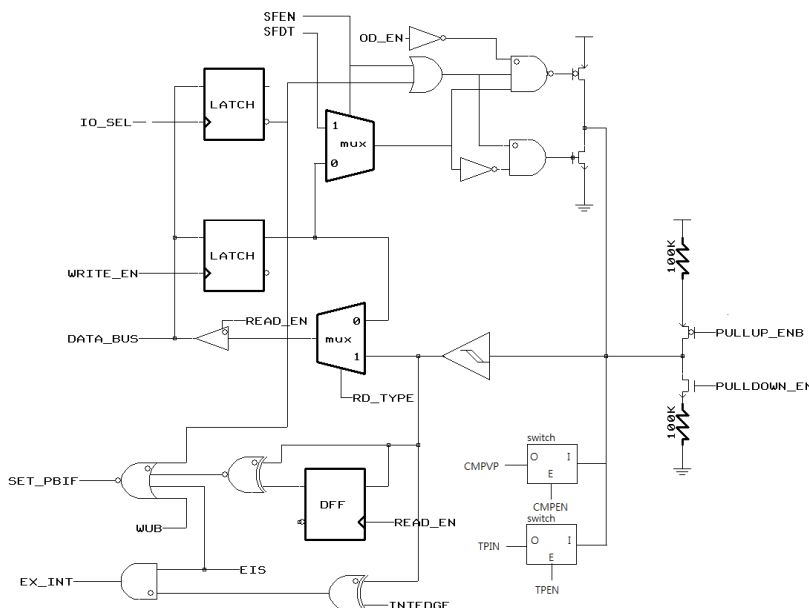


图 5 PB0 引脚结构框图

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: 开启开漏功能。

PULLUP_ENB: 开启内部上拉电阻。

PULLDOWN_EN: 开启内部下拉电阻。

RD_类型: 选择读取脚位或数据锁存器。

SFEN: 开启特殊功能 (IR)。

SFDT: 特殊功能数据。

WUB: 开启PortB 唤醒功能。

SET_PBIF: 设置PortB 唤醒标志。

CAP_EN: 启为外部电容引脚。

INITIAL: 电容放电。

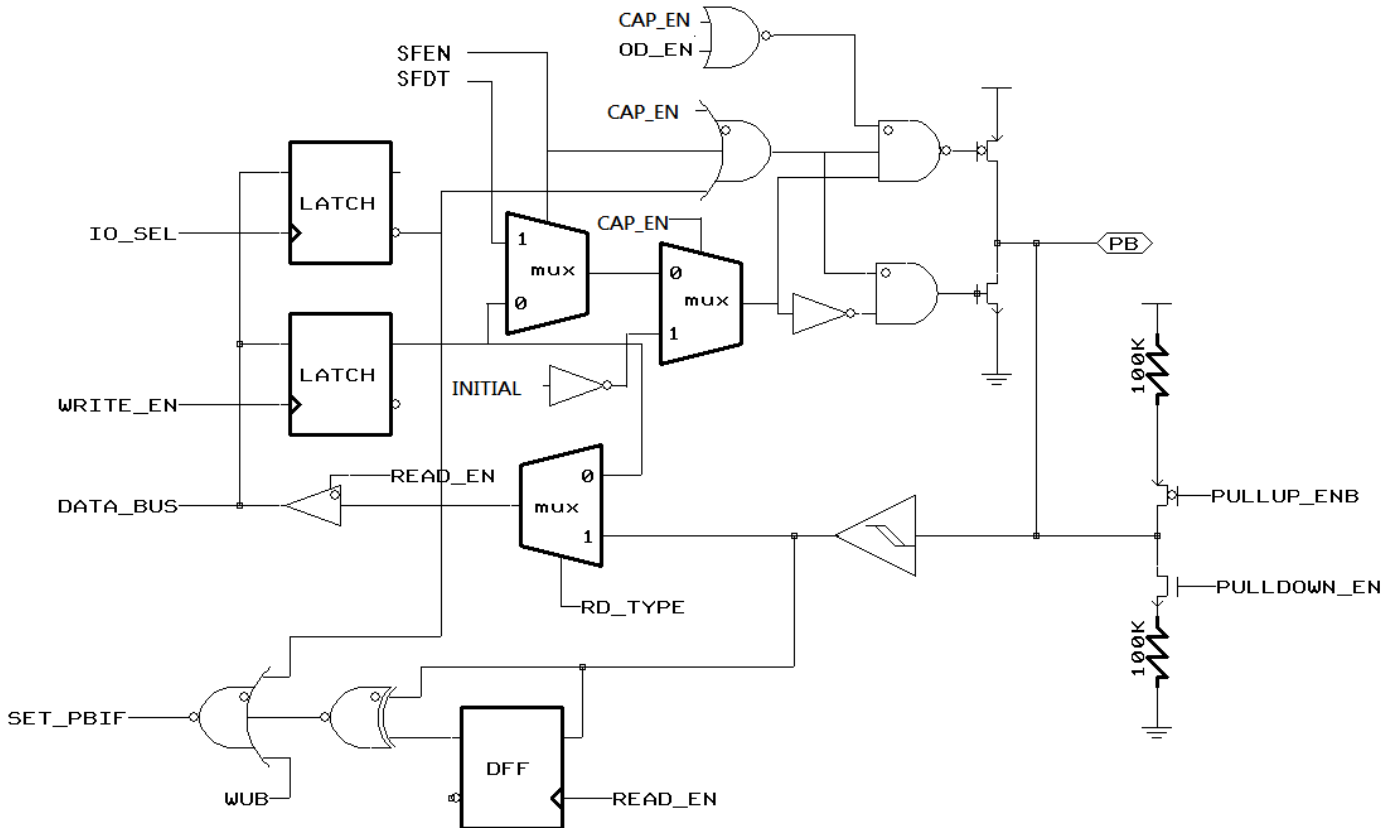


图 6 PB1 引脚结构框图

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: 开启开漏功能。

PULLUP_ENB: 开启内部上拉电阻。

PULLDOWN_EN: 开启内部下拉电阻。

RD_类型: 选择读取脚位或数据锁存器。

SFEN: 开启特殊功能 (PWM1/BUZZER1)。

SFDT: 特殊功能数据。

WUB: 开启PortB 唤醒功能。

SET_PBIF: 设置PortB 唤醒标志。

EX CKI: 外部时钟输入。

CMPEN: 开启为比较器输入引脚。

TPEN: 开启为触控键输入引脚。

CMPVP: 比较器正输入引脚。

TPIN: 触控键输入。

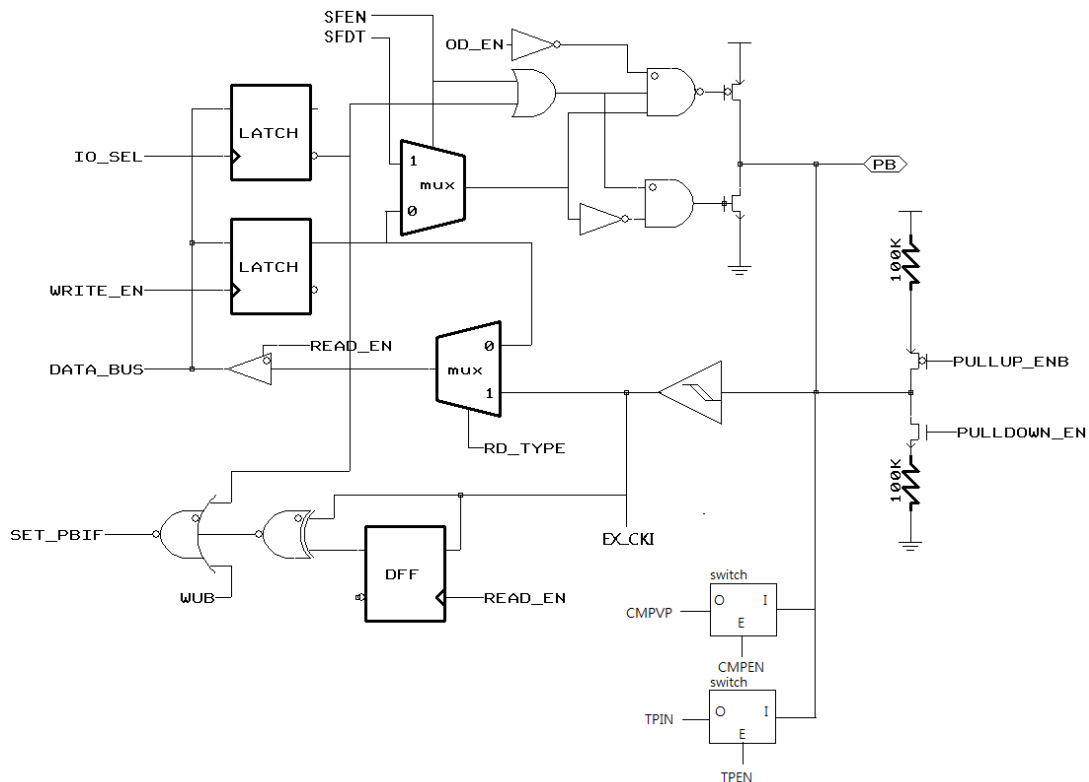


图 7 PB2 引脚结构框图

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。

RSTPAD_EN: 开启复位引脚。

RSTB_IN: 复位输入脚。

PULLUP_ENB: 开启内部上拉电阻。

PULLDOWN_EN: 开启内部下拉电阻。

RD_类型: 选择读取脚位或数据锁存器。

WUB: 开启PortB 唤醒功能。

SET_PBIF: 设置PortB 唤醒标志。

CMPOE: 开启为比较器输出引脚。

PCON1[6]: 比较器输出。

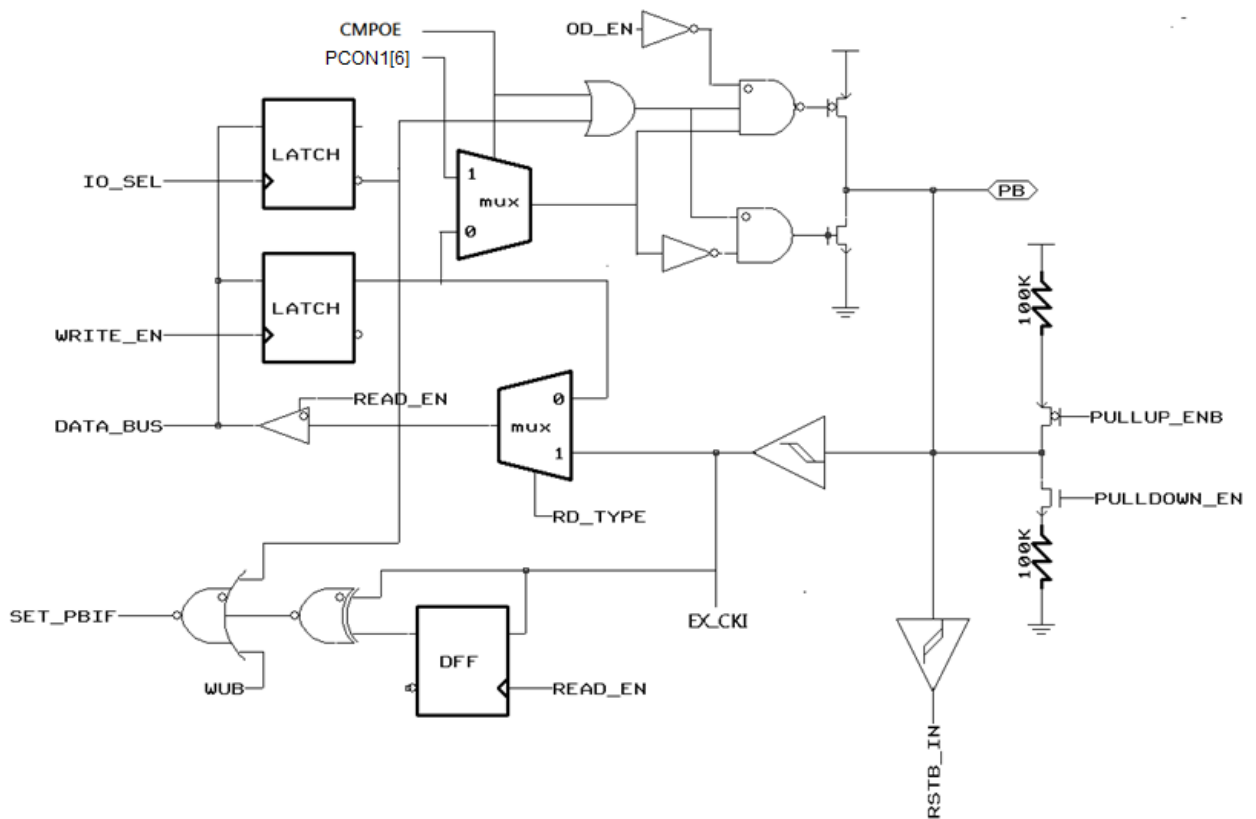


图 8 PB3 引脚结构框图

- 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: 开启开漏功能。
- PULLUP_ENB: 开启内部上拉电阻。
- RD_类型: 选择读取脚位或数据锁存器。
- WUB: 开启PortB 唤醒功能。
- SET_PBIF: 设置PortB 唤醒标志。
- CMPEN: 开启为比较器输入引脚。
- TPEN: 开启为触控键输入引脚。
- CMPVN: 比较器负输入引脚。
- TPIN: 触控键输入。

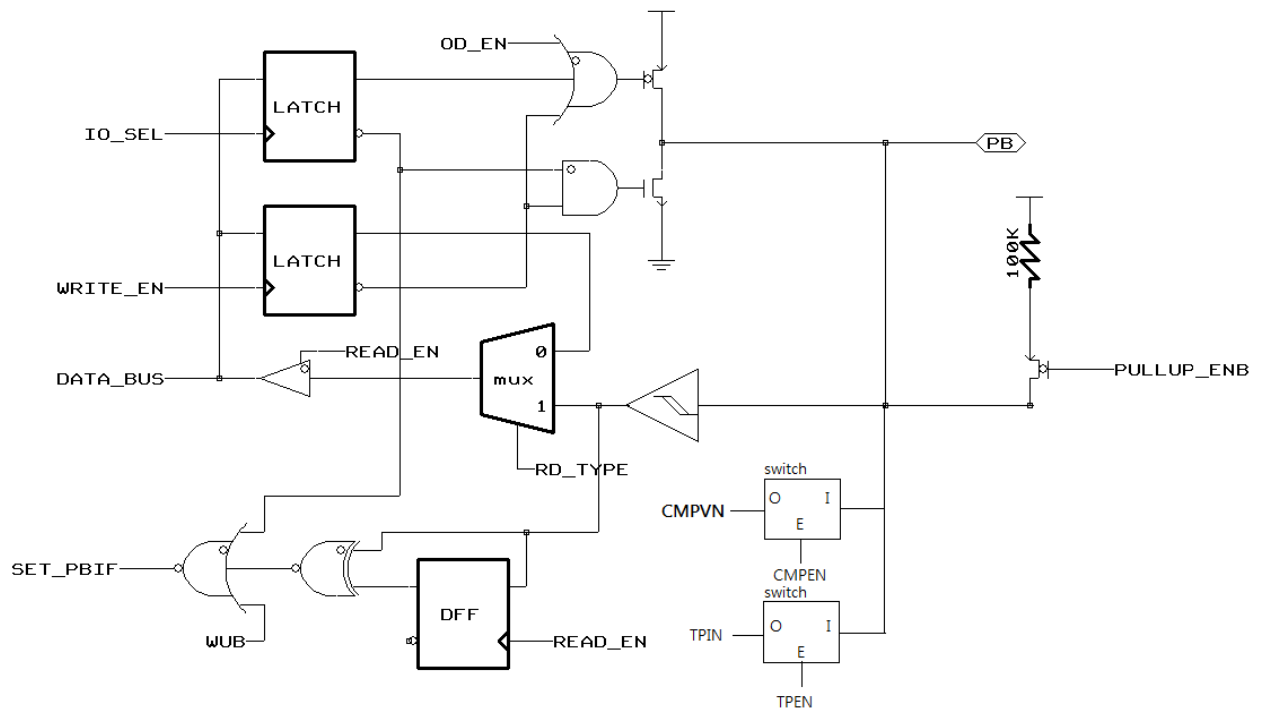


图 9 PB4/PB5 引脚结构框图

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}、外部时钟输入脚 EX_CKI 或低频震荡 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 为 1 且 LCK_TM0 为 0，EX_CKI 会被当作定时器 0 时钟源。当 T0CS 是 1 且 LCK_TM0 为 1，会选择低频震荡 I_LRC 当作定时器 0 时钟源。汇总成表格如下。(也请参考表 11)

定时器 0 时钟源	T0CS	LCKTM0	定时器 0 来源
Instruction clock	0	X	X
EX_CKI	1	0	X
		X	0
I_LRC	1	1	1

表 11 定时器 0 时钟源摘要

寄存器 T0CE (T0MD[4]) 可决定 EX_CKI 或 I_LRC 的时钟触发沿选择。当 T0CE 是 1，EX_CKI 或 I_LRC 的上升沿将让定时器 0 计数加一。当 T0CE 是 0，EX_CKI 或 I_LRC 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 与寄存器 PS0CV。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。

定时器 0 时钟源默认为指令时钟。如果外部时钟脚 EX_CKI 或低频震荡 I_LRC 被用来当作定时器 0 时钟源，用户必须注意分频后的频率不能超过指令时钟，否则会导致错误计数。当 I_LRC 同时被当作定时器 0 时钟源与指令时钟，NY8TM52D 必须指定预分频器 0 到定时器 0，且须注意预分频器 0 的预分频比不得小于 4。当配置字节设定为异步 (Async.)，定时器 0 时外部钟源 EX_CKI 频率就可高于指令时钟。EX_CKI 能输入的最大频率取决于变化过程。

当定时器 0 上溢，寄存器 TOIF (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 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：

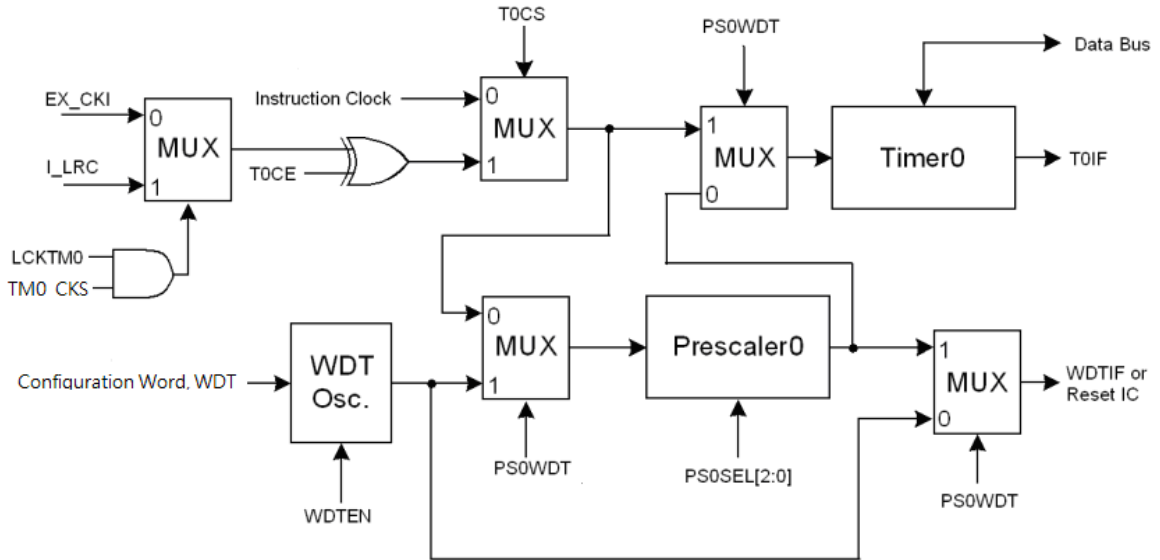


图 10 定时器 0 与WDT结构框图

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。定时器 1 创建自动重载功能和定时器 1 用双重缓存器重载储存数据。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：

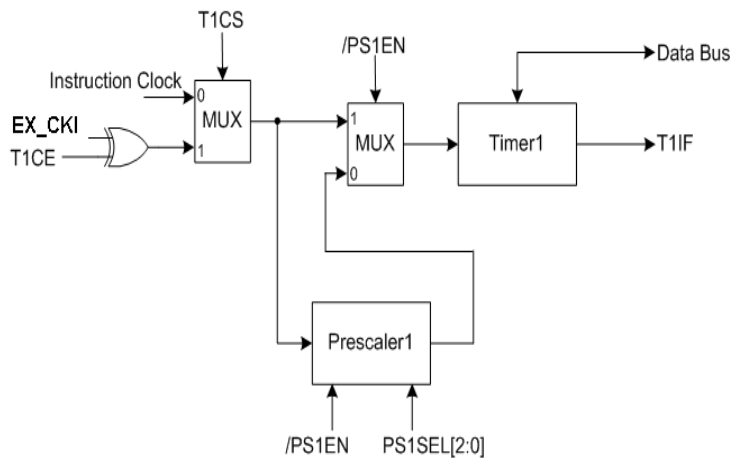


图 11 定时器 1 结构框图

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_CKI。当T1CS为 0, 指令时钟会被选择当做时钟源。当T1CS为 1, 则是EX_CKI当做时钟源。当EX_CKI被选取, 寄存器控制位T1CE (T1CR2[4]) 可决定EX_CKI的时钟触发沿。当T1CE是 1, EX_CKI的上升沿将让定时器 1 计数减一。当T1CE是 0, EX_CKI的下降沿将让定时器 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 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 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, 标明定时器 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:

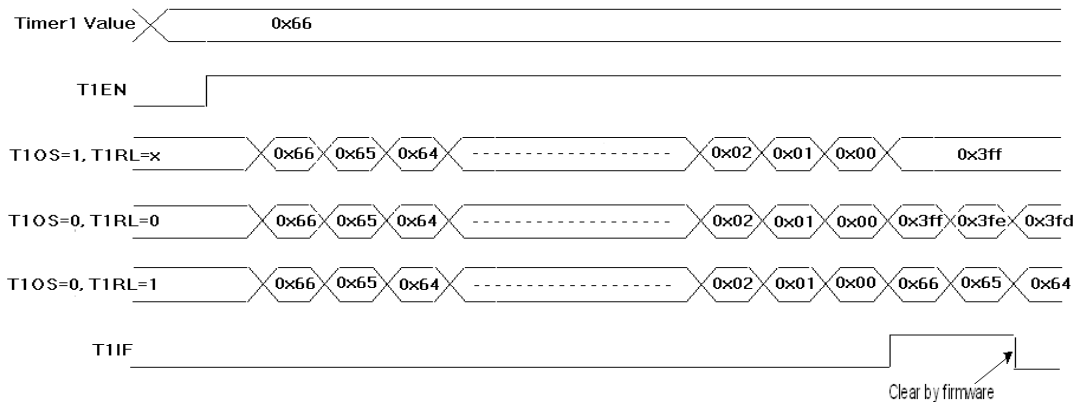


图 12 定时器 1 时序图

当寄存器PWM1OEN (T1CR1[7]) 设定为 1, PB2 为PWM1 输出。当PWM1OEN为 1, PB2 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 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]。当用户写入PWM1DUTY时, 先写入PWM1DUTY[9:8]高 2 位 (TMRH[1:0]), 再写入PWM1DUTY[7:0]时, 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:

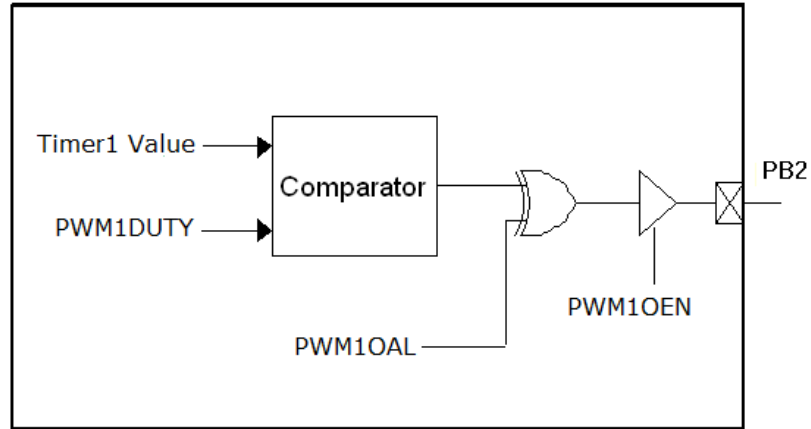


图 13 PWM1 结构框图

当寄存器BZ1EN (BZ1CR1[7]) 设定成 1 且使能配置字节, PB2 为蜂鸣器 1 输出。当BZ1EN设定为 1, PB2 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:

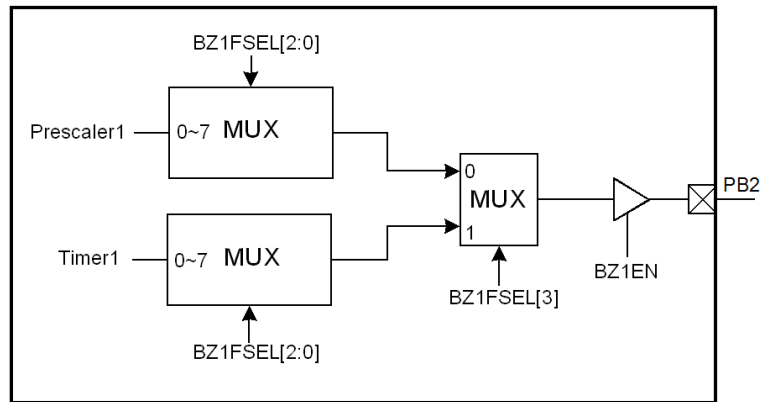


图 14 蜂鸣器 1 结构框图

3.9 PWM2

当寄存器PWM2OEN (P2CR1[7]) 设定为 1, PB2 或PA0 为PWM2 输出。当PWM2OEN为 1, PB2 或PA0 会自动成为输出脚。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。

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[9:0]为 0, PWM2 无法输出占空比。当PWM2DUTY[9:0]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[9:0]数值必须小于或等于TMR1[9:0]。先写入PWM2 高 2 位 (TMRH[3:2]) 再写入PWM2DUTY[7:0]时, 就可更新PWM2DUTY[9:0]重载缓存寄存器。等到定时器 1 下溢后, PWM2DUTY[9:0]重载缓存寄存器才会写入PWM2DUTY[9:0]。

PWM2 的结构框图如下:

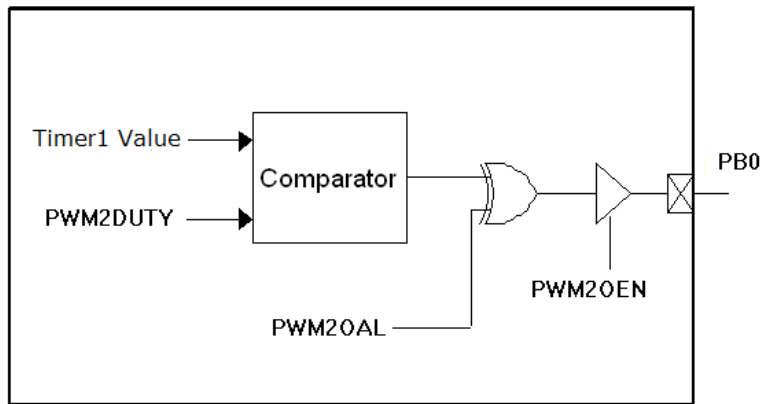


图 15 PWM2 结构框图

3.10 PWM3

当寄存器PWM3OEN (T3CR[7]) 设定为 1, PB3 为PWM3 输出。当PWM3OEN为 1, PB3 会自动成为输出脚。PWM3 输出的有效状态是由寄存器PWM3OAL (T3CR[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 的占空比与帧率是可编程的。占空比是由寄存器PWM3DUTY决定。当PWM3DUTY为 0, PWM3 无法输出占空比。当PWM3DUTY为 0x3FF, PWM3 将输出 1023 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]。先写入PWM3 高 2 位 (PWM3RH[1:0]) 再写入PWM3DUTY[7:0]时, 就可更新PWM3DUTY[9:0]重载缓存寄存器。等到定时器 1 下溢后, PWM3DUTY[9:0]重载缓存寄存器才会写入PWM3DUTY[9:0]。

PWM3 的结构框图如下:

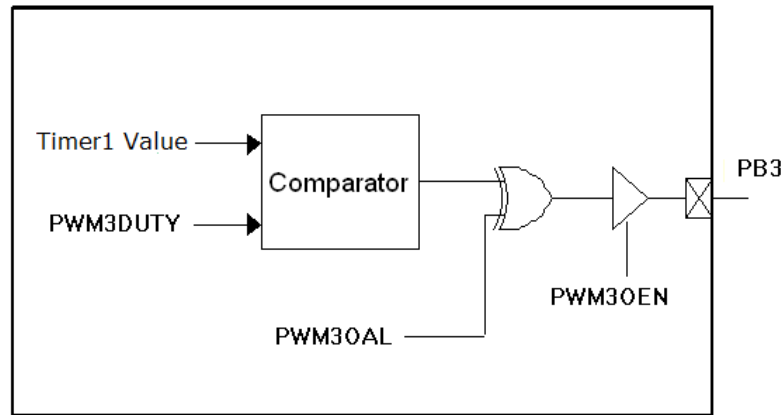


图 16 PWM3 结构框图

3.11 IR Carrier

寄存器IREN (IRCR[0]) 被设定为 1 后, PB3 为红外线载波输出, 而PB3 会自动成为输出脚。当IREN清零, PB3 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 62.5KHz。当IRF57K为 0, 频率是 41.6KHz。

红外线载波的极性会根据PB3 输出数据所决定。当寄存器IRCSEL (IRCR[2]) 为 1 且PB3 输出数据为 0, 红外线载波将由PB3 输出。当寄存器IRCSEL (IRCR[2]) 为 0 且PB3 输出数据为 1, 红外线载波将由PB3 输出。红外线载波的极性如下图所示:

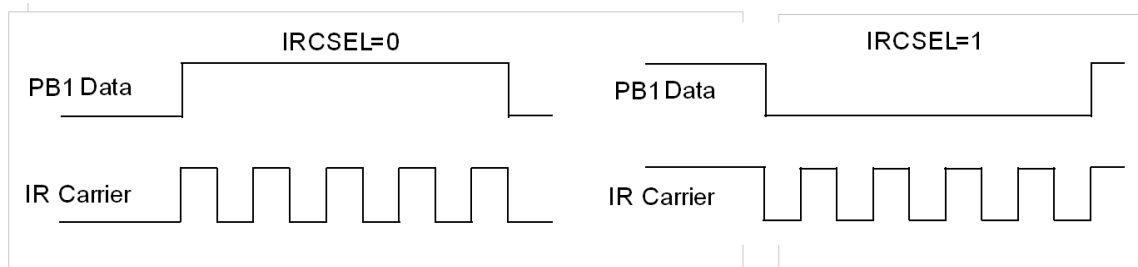


图 17 红外线载波的极性 vs. 输出数据

3.12 低电压侦测 (LVD)

NY8TM52D内置准确的低电压侦测电路来侦测VDD电压水平。将LV DEN (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LV DOUT (寄存器PCON1[6]) 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结构框图如下:

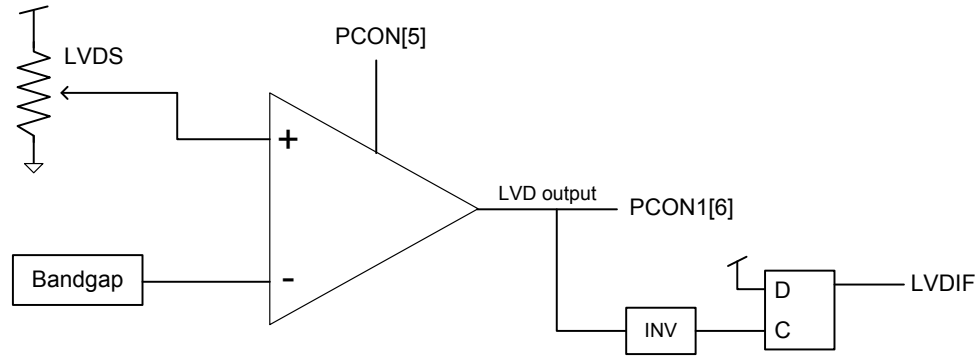


图 18 LVD 结构框图

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 12 选择LVD电压

注意: LVD 的滞后电压是 0.1V, 用于电池充电应用中 (检测电压是由低到高), LVD 电压选择如下表所示:

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
0110	(2.9+0.1) V
0111	(3.0+0.1) V

LVDS[3:0]	Voltage
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：依 LVDS[3:0]选择 LVD 电压。

步骤 2：设置 CMPCR = 0x0A

步骤 3：设置 PCON[5]=1

步骤 4：通过 PCON1[6]检查 LVD 状态。

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

3.13 电压比较器

NY8TM52D内置一组电压比较器。比较器的正输入源与负输入源和GPIO口复用，内部参考电压只能在P2V模式下提供给比较器的负输入源使用。

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。

比较器的框图如下：

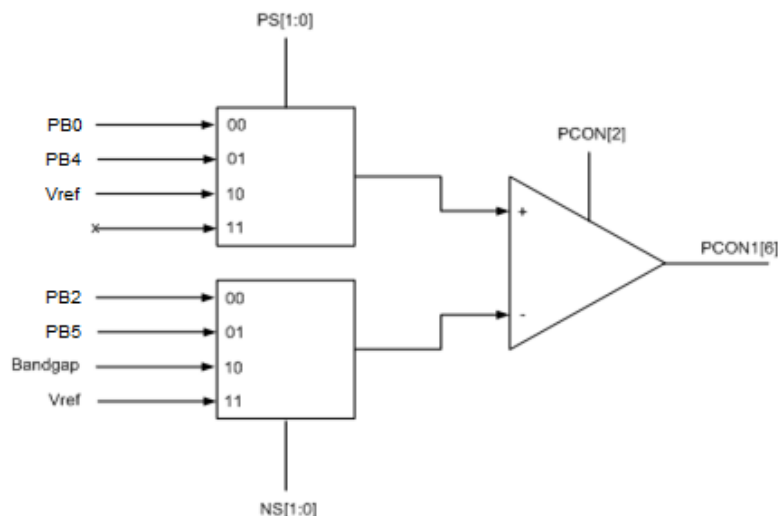


图 19 比较器的框图

3.13.1 比较器参考电压

内部参考电压Vref是由不同的电阻来提供不同的参考电压值。RBIAS_H和RBIAS_L用来选择最大和最小的Vref值。LVDS[3:0]用于选择 16 种电压中的一种。

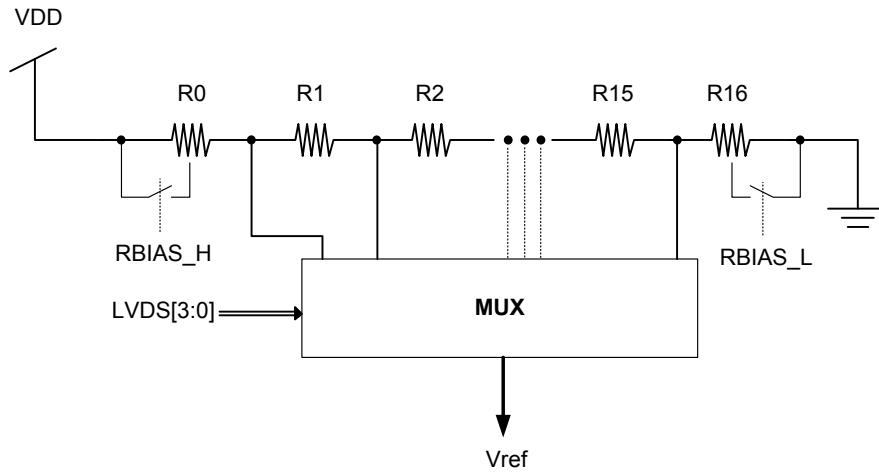


图 20 Vref 的硬件连接

Vref由RBIAS_H, RBIAS_L 和 LVDS[3:0]决定, LVDS[3:0]用来选择 16 种参考电压中的一种, 表格如下:

LVDS[3:0]	RBIAS_H=1 RBIAS_L=0	RBIAS_H=0 RBIAS_L=1
0000	65/128 V _{DD}	31/128 V _{DD}
0001	62/128 V _{DD}	29/128 V _{DD}
0010	56/128 V _{DD}	26/128 V _{DD}
0011	52/128 V _{DD}	23/128 V _{DD}
0100	48/128 V _{DD}	20/128 V _{DD}
0101	44/128 V _{DD}	18/128 V _{DD}
0110	43/128 V _{DD}	17/128 V _{DD}
0111	41/128 V _{DD}	16/128 V _{DD}
1000	39/128 V _{DD}	14/128 V _{DD}
1001	37/128 V _{DD}	13/128 V _{DD}
1010	35/128 V _{DD}	12/128 V _{DD}
1011	34/128 V _{DD}	11/128 V _{DD}
1100	32/128 V _{DD}	10/128 V _{DD}
1101	31/128 V _{DD}	9/128 V _{DD}
1110	30/128 V _{DD}	8/128 V _{DD}
1111	29/128 V _{DD}	7/128 V _{DD}

表 13 参考电压 Vref 选择表

注意: Vref的误差为±0.1V。

比较器的正向输入由PS[1:0]决定（寄存器CMPCR[3:2]）。

表格如下：

PS[1:0]	正向输入
00	PB0
01	PB4
10	Vref
11	---

表 14 正向输入选择表

比较器的负向输入由NS[1:0]决定（寄存器CMPCR[1:0]）。

表格如下：

NS[1:0]	负向输入
00	PB2
01	PB5
10	Bandgap (0.6V)
11	Vref

表 15 负向输入选择

有两种方式可以取得比较器的比较结果，一种是寄存器轮询，另一种是探查输出脚位。

比较器输出可以由 LVDOUT 轮询（寄存器 PCON1[6]）取得。

在输出脚位探查比较器输出，设置 CMPOE(寄存器 OSCCR[6])为 1，PB3 引脚将实时输出比较器结果。

注意，当 CMPOE=1 时，PWM3 功能若已开启了，此时将会关闭。

3.14 触控脚位

NY8TM52D可以用触控功能取代机械开关或按钮。内建的LDO稳压器为触控传感器在触控应用中提供了稳定的电容传感设计。

通过TPPADEN SFR来定义 1~4 个触控按键。通过TPCHS SFR来扫描触控按键。触按键的调制时钟或选择 1.31M, 1.12M, 1M, 0.88M 或 0.75M。触控脚位在运行前，触控键计数器可以由TPCNTH 和 TPCNTL来设置，然后设置TPRUN，触控键将会开始转换为触控传感计数器。外部触控电容的电压超出参考电压或触控计数器上溢都将终止调制，并且TPCPIF=1 或 TPOVIF=1。

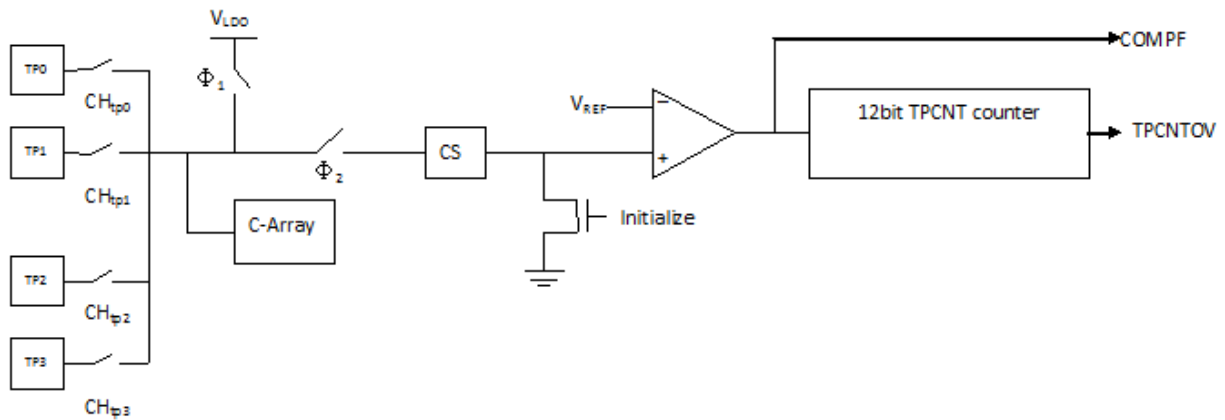
NY8TM52D支持触控键慢速模式来减少CPU待机模式中的电源损耗。慢速模式设置扫描频率为 16Hz 或 32Hz，然后在触控慢速模式中设置TPRUN，调制将每 62ms或 31ms自动开始。

写入TPCHS SFR将会重载TPCNTH和TPCNTL到触控计数器并且设置TPMD为TPCHSOFF。

写入TPMD SFR也会重载TPCNTH和TPCNTL到触控计数器。

清除触控中断标志将会重载TPCNTH和TPCNTL到触控计数器。

在TPRUN模式，清除触控中断标志将会设置TPMD 为 TPCHSOFF，但不在触控慢速模式。



3.15 看门狗定时器 (WDT)

NY8TM52D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，仍然可以通过寄存器WDTEN (PCON[7]) 位来开启/关闭。此外，WDT上溢后可由配置字节决定的复位NY8TM52D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 3.5ms、15ms、60ms或 250ms。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NY8TM52D，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。

3.16 中断

NY8TM52D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 8 种：

- Timer0 上溢中断。
- Timer1 下溢中断。
- WDT中断。
- PB 输入状态改变中断。
- 外部中断输入。
- 低电压侦测中断。
- 触控引脚比较器中断。
- 触控引脚计数器上溢中断。

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NY8TM52D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，用户可以通过轮询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，NY8TM52D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NY8TM52D。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 PB 输入状态改变中断

当PBx (0 ≤ x ≤ 5) 设置为输入口且相应的寄存器WUPBx位设置为 1，且如果PBIE和GIE设置为 1，当这些选定输入口上的状态变化时，寄存器PBIF位将被硬件设为 1 并处理此中断请求。需注意当PB0 同时设置为状态变化中断和外部中断时，外部中断标志EIS将禁止PB0 状态变化中断。

3.16.5 外部中断

根据EIS=1 和寄存器INTEDG的配置，如果INTIE和GIE设置为 1，PB0 引脚上的有效边沿触发会让寄存器INTIF 位将被硬件设为 1 并处理此中断请求。

3.16.6 LVD 低电压侦测中断

当VDD电压水平低于LVD电压，读取LVDOOUT（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 触控脚位比较器中断

当触控外部电容电压超出触控比较器参考电压，TPCMP标志将会由低变到高，且设置寄存器TPCPIF。如果TPCPIE和GIE都设置为 1，此中断将会被处理。

3.16.8 触控脚位计数器上溢中断

当触控计数器上溢后设置TPOVIF，如果TPOVIF和GIE都设置为 1，此中断将会被处理。

3.17 振荡器配置

因为NY8TM52D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})。可用作F_{Losc}的振荡器是内部低速RC振荡器 (I_{LRC})。

- (1) STPHOSC(OSCCR[1])=1 will stop F_{Hosc}
- (2) F_{Hosc} will be disabled automatically at Halt mode

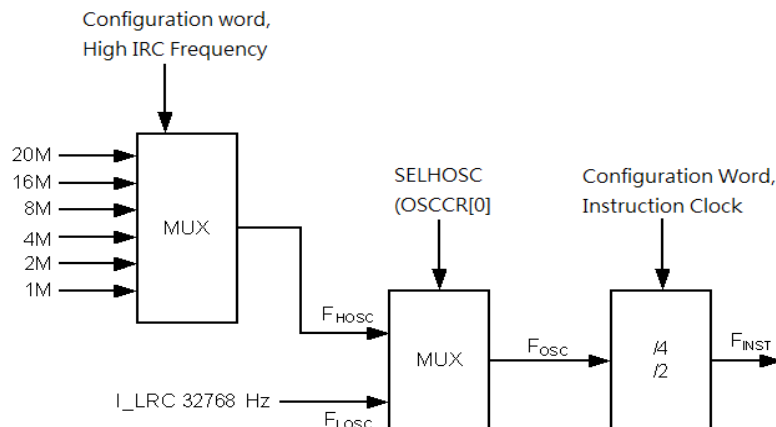


图 21 NY8TM52D 振荡配置结构图

系统时钟频率可由三个配置字节设置为I_HRC，可以选择的频率为 1M、2M、4M、8M、16M或 20MHz。

当配置字节选择I_LRC时，其频率约为 32768Hz。

根据寄存器SELHOSC(OSCCR[0])的值来选择F_{HOSC} 或 F_{LOSC}作为系统晶振。当SELHOSC 为 1 时，F_{HOSC}选择作系统晶振，当SELHOSC 为 0 时，F_{LOSC}选择作系统晶振，一旦Fosc被决定了，指令时钟会根据配置字节变成 Fosc/2 或 Fosc/4。

3.18 工作模式

NY8TM52D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，NY8TM52D将停止几乎所有的运作，可由定时器、比较器、LVD、看门狗与外部事件等来唤醒。在睡眠模式下，NY8TM52D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。

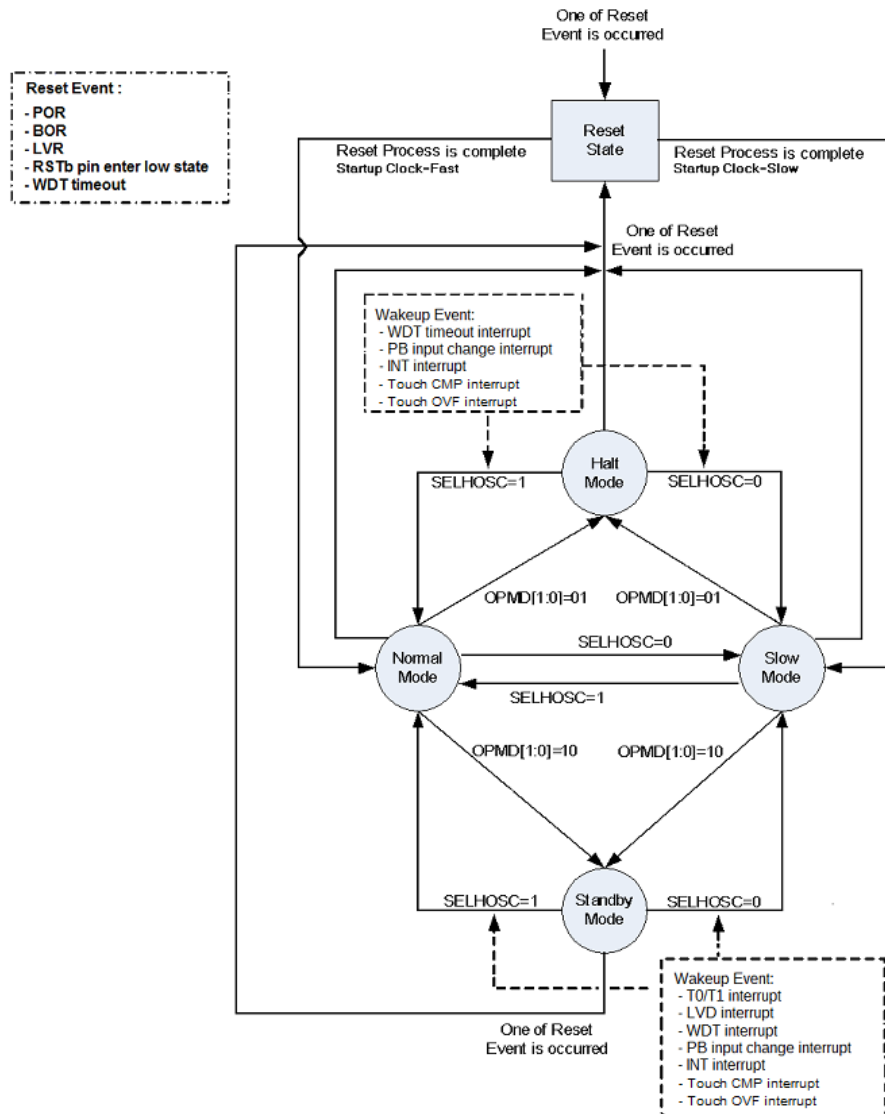


图 22 四个工作模式

3.18.1 Normal Mode

发生任何复位事件并且复位过程完成后，NY8TM52D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} ，NY8TM52D将进入正常模式，如果启动时钟为 F_{LOSC} ，NY8TM52D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NY8TM52D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，NY8TM52D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，NY8TM52D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NY8TM52D关闭。因此在慢速模式下，用户可写 0 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]，NY8TM52D将进入待机模式。然而，在待机模式下， F_{HOSC} 不会自动被NY8TM52D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NY8TM52D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NY8TM52D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) PB输入状态改变中断，(d) 外部中断，(e) LVD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並改变振荡模式（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，NY8TM52D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NY8TM52D只能通过一些特殊事件唤醒。因此，睡眠模式是NY8TM52D最省电的模式。

- 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。
- 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）PB输入状态改变中断，（c）INT或外部中断发生。
- 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 唤醒稳定时间

睡眠模式的唤醒等待时间为 $16 \cdot F_{osc}$ ，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NY8TM52D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，NY8TM52D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	Disabled
F _{LOSC}	使能	使能	使能	Disabled
指令执行	执行	执行	停止	停止
计时器 0/1	T0EN / T1EN	T0EN / T1EN	T0EN / T1EN	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 上溢 - Timer1 下溢 - WDT 超时 - PB输入状态改变 - 外部中断 - LVD中断 - Touch CMP 中断 - Touch OVF 中断	- WDT 超时 - PB 输入状态改变 - 外部中断 - Touch CMP中断 - Touch OVF中断

表 12 工作模式概述

3.19 复位

当以下任一复位事件发生时，NY8TM52D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
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 13 /TO和/PD值和相关事件概述

复位事件发生后，NY8TM52D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，NY8TM52D将等待 Fosc 的 16 个时钟周期（OST，振荡器启动时间）后完成复位。

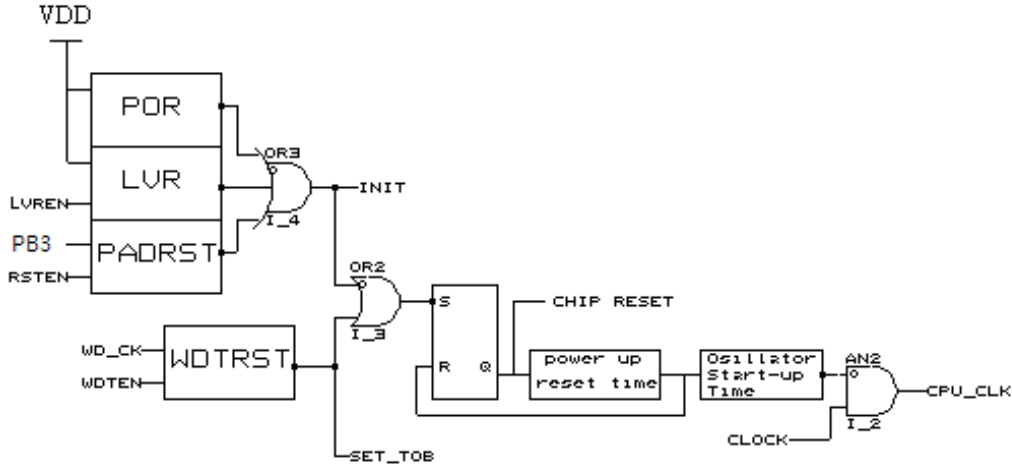


图 23 芯片复位电路框图

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。

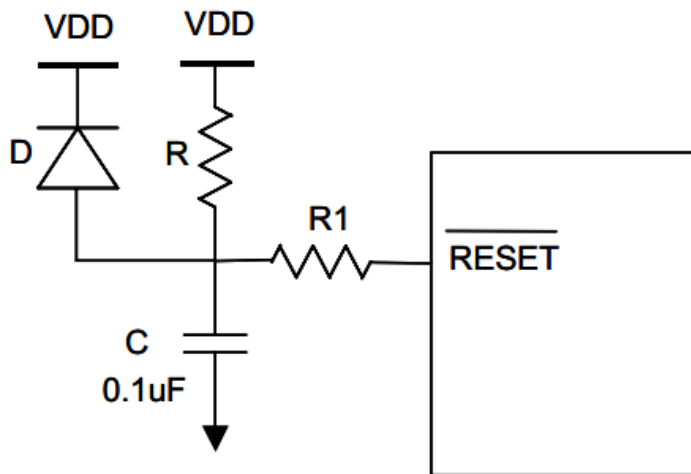


图 24 外部上电复位硬件连接图

4. 指令设置

NY8TM52D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
数据传送指令					
TFUN	T		Load ACC to T-page SFR	1	
TFUNR	T		Move T-page SFR to ACC	1	
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	-

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 0x00~0x3F。

S: S页面特殊功能寄存器, S值为 0x0 ~ 0x13。

T: T页面特殊功能寄存器, T值为 0x0 ~ 0x7F。

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	$0 \leq R \leq 63$ $d = 0, 1.$
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	$0 \leq R \leq 63$ $d = 0, 1.$
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	Add the contents of ACC with the 8-bit immediate data i. The result is placed in 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	$0 \leq R \leq 63$. $d = 0, 1$.	操作数	$0 \leq R \leq 63$ $0 \leq \text{bit} \leq 7$
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	$0 \leq R \leq 63$ $0 \leq \text{bit} \leq 7$
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	$0 \leq R \leq 63$ $0 \leq \text{bit} \leq 7$
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 8 位立即地址载入 PC[7:0], 将 TBHP[2:0] 载入 PC[10:8]。
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	$0 \leq R \leq 63$ $0 \leq \text{bit} \leq 7$
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	$0 \leq R \leq 63$
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	$0 \leq R \leq 63$ d = 0, 1.
操作	$\sim R \rightarrow \text{dest}$
状态影响	Z
说明	R寄存器取补, 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	$0 \leq R \leq 63$
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	DECRSZ	Decrease R, Skip if 0
语法	DAA	语法	DECRSZ R, d
操作数	--	操作数	$0 \leq R \leq 63$ $d = 0, 1.$
操作	ACC(hex) \rightarrow ACC(dec)	操作	R - 1 \rightarrow 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	$0 \leq R \leq 63$ $d = 0, 1.$	操作数	--
操作	R - 1 \rightarrow dest	操作	Disable Interrupt, 0 \rightarrow GIE
状态影响	Z	状态影响	--
说明	Decrease R. If d is 0, the result is stored in ACC. If d is 1, the result is stored back to 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	INCR	Increase R
语法	ENI	语法	INCR R, d
操作数	--	操作数	$0 \leq R \leq 63$ $d = 0, 1.$
操作	Enable Interrupt, $1 \rightarrow$ GIE	操作	$R + 1 \rightarrow$ dest.
状态影响	--	状态影响	Z
说明	GIE设置为 1，开启总中断。	说明	$R + 1$ ，若 $d="0"$ ，结果存入ACC； 若 $d="1"$ ，结果存入R。
周期	1	周期	1
举例	ENI 执行指令前： GIE=0. 执行指令后： GIE=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	INCRSZ	Increase R, Skip if 0
语法	GOTOA	语法	INCRSZ R, d
操作数	--	操作数	$0 \leq R \leq 63$ $d = 0, 1.$
操作	{TBHP, ACC} \rightarrow PC	操作	$R + 1 \rightarrow$ dest, Skip if result = 0
状态影响	--	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]。	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	2	周期	1 or 2（跳过）
举例	GOTOA 执行指令前： PC=A0. TBHP=0x02, ACC=0x34. 执行指令后： PC=0x234.	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。

INCRSZ	Increase R, Skip if 0
语法	INCRSZ R, d

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.

IORAR	OR ACC with R
语法	IORAR R, d
操作数	$0 \leq R \leq 63$ 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.

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.

IOST	Load F-page SFR from ACC
语法	IOST F
操作数	$0 \leq F \leq 15$
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	$0 \leq F \leq 15$
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	$0 \leq R \leq 63$
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	$0 \leq R \leq 63$ $d = 0, 1.$
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

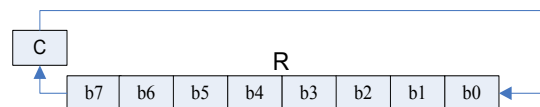
语法	RETIA i
操作数	$0 \leq i < 255$
操作	$i \rightarrow \text{ACC}$, Top of Stack \rightarrow PC
状态影响	--
说明	带参数返回：8 位立即数赋值给 ACC，栈顶地址载入 PC，GIE 标志为 0。
周期	2
举例	RETIA i 执行指令前： GIE=0, Stack pointer =2, i=0x55, ACC=0xAA. 执行指令后： GIE=1, PC=Stack[2], Stack pointer =1, ACC=0x55.

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

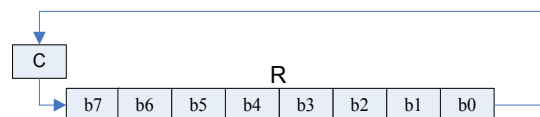
语法	RLR R, d
操作数	$0 \leq R \leq 63$ $d = 0, 1.$
操作	$C \rightarrow \text{dest}[0], R[7] \rightarrow C,$ $R[6:0] \rightarrow \text{dest}[7:1]$



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	$0 \leq R \leq 63$ $d = 0, 1.$
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	$0 \leq R \leq 63$ $d = 0, 1.$
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)

SBCAR	Subtract ACC and Carry from R
语法	SBCAR R, d
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	$0 \leq S \leq 15$
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

语法	SFUNR S
操作数	$0 \leq S \leq 15$
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.

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.

SUBAR Subtract ACC from R

语法	SUBAR R, d
操作数	$0 \leq R \leq 63$ d = 0, 1.
操作	R - ACC \rightarrow 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)

SUBIA	Subtract ACC from Immediate
语法	SUBIA i
操作数	$0 \leq i < 255$
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： $ACC=0x01, C=1. (+1)$

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： $TBHP=0x02, CC=0x34$. $TBHD=0x01$. $ROM\ data[0x234]= 0x35AA$ 执行指令后： $TBHD=0x35, ACC=0xAA$.

SWAPR	Swap High/Low Nibble in R
语法	SWAPR R, d
操作数	$0 \leq R \leq 63$ $d = 0, 1$.
操作	$R[3:0] \rightarrow dest[7:4]$. $R[7:4] \rightarrow dest[3:0]$
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： $T0MD=0xAA$.

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.

TFUN Load T-page SFR from ACC

语法	TFUN T
操作数	$0 \leq T \leq 14$
操作	ACC → 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.

TFUNR Move T-page SFR to ACC

语法	TFUNR T
操作数	$0 \leq T \leq 14$
操作	T-page SFR → 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.

XORAR Exclusive-OR ACC with R

语法	XORAR R, d
操作数	$0 \leq R \leq 63$ 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，若d=“0”，结果存入ACC；若d=“1”，结果存入R。
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	内部高速 RC 频率	1. 1MHz 2. 2MHz 3. 4MHz 4. 8MHz 5. 16MHz 6. 20MHz
2	指令时钟	1. 2 个振荡周期 2. 4 个振荡周期
3	看门狗定时器	1. 看门狗开启（程序控制） 2. 看门狗关闭（永远关闭）
4	看门狗定时器事件	1. 看门狗复位 2. 看门狗中断
5	定时器 0 时钟源	1. EX_CKI 2. I_LIRC 低频振荡器
6	PB.2	1. PB.2 为 I/O 口 2. PB.2 为 PWM 输出 3. PB.2 为蜂鸣器输出
7	PB.3	1. PB.3 为 I/O 口 2. PB.3 为复位脚
8	PB.4	1. PB.4 为 I/O 口 2. PB.4 输出指令时钟
9	上电复位时间	1. 140us 2. 4.5ms 3. 18ms 4. 72ms 5. 288ms
10	看门狗定时器时基	1. 3.5ms 2. 15ms 3. 60ms 4. 250ms
11	LVR 开关设定	1. 寄存器控制 2. 寄存器控制+ 睡眠模式关闭 3. LVR 永远开启 4. 工作模式开启+ 睡眠模式关闭
12	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. 3.4V 9. 3.6V 10. 4.2V
13	VDD 电压	1. 3.0V 2. 4.5V 3. 5.0V
14	读取输出口数据	1. I/O 口 2. 寄存器
15	EX_CKI to Inst. Clock	1. 同步 2. 不同步
16	上电时钟源	1. I_HRC 高速 2. I_LIRC 低速
17	输入高电压 (VIH)	1. CMOS (0.7VDD) 2. TTL (0.5VDD)
18	输入高电压 (VIL)	1. CMOS (0.3VDD) 2. TTL (0.2VDD)
19	史密特触发输入	1. 开启 2. 关闭
20	灌电流	1. 正常电流 2. 大电流
21	比较器输入引脚选择	1. 开启 2. 关闭

表 15 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

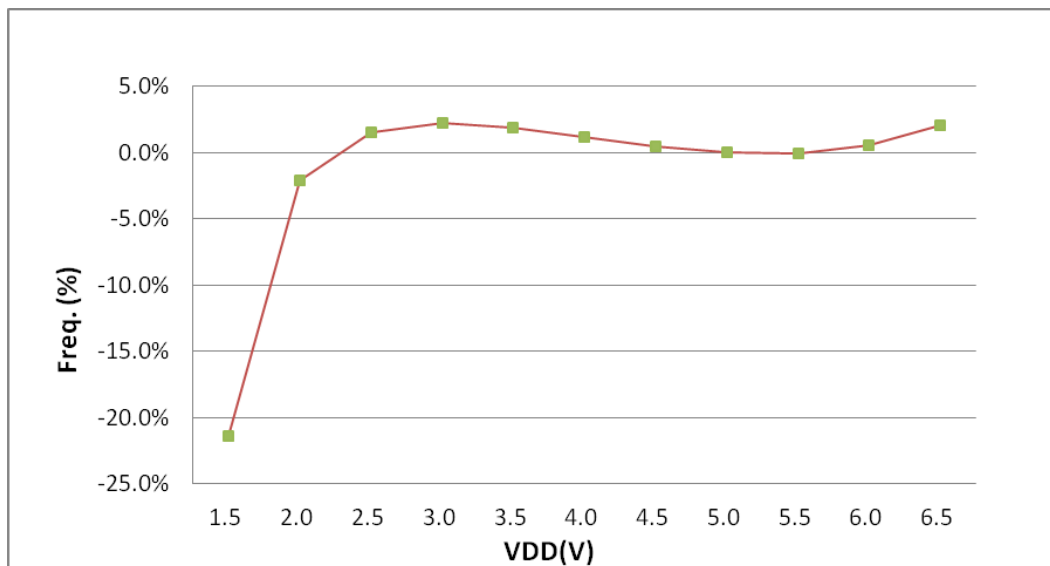
($F_{INST}=F_{HOSC}/4$, $F_{HOSC}=16MHz@I_HRC$, WDT开启, 环境温度 $T_A=25^{\circ}C$)

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				$F_{INST}=8MHz @ I_HRC/4$
			1.6				$F_{INST}=4MHz @ I_HRC/4$
			1.6				$F_{INST}=2MHz @ I_HRC/4$
			1.6				$F_{INST}=1MHz @ I_HRC/4$
			1.6				$F_{INST}=32KHz @ I_LRC/4 \& I_LRC/2$
V_{IH}	输入高电平	5V	4.0	--	--	V	RSTb (0.8 V_{DD})
		3V	2.4	--	--		
		5V	3.5	--	--	V	所有I/O引脚, EX_CKI, INT CMOS (0.7 V_{DD})
		3V	2.1	--	--		
		5V	2.5	--	--	V	All other I/O pins, EX_CKI, INT TTL (0.5 V_{DD})
		3V	1.5	--	--		
		5V	2.1	--	--	V	All other I/O pins, EX_CKI, INT 无施密特触发输入(0.5 V_{DD})
3V	1.3	--	--				
V_{IL}	输入低电平	5V	--	--	1.0	V	RSTb (0.2 V_{DD})
		3V	--	--	0.6		
		5V	--	--	1.5	V	所有I/O引脚, EX_CKI, INT CMOS (0.3 V_{DD})
		3V	--	--	0.9		
		5V	--	--	1.0	V	所有I/O引脚, EX_CKI, INT TTL (0.2 V_{DD})
		3V	--	--	0.6		
		5V	--	--	2.1	V	所有I/O引脚, EX_CKI, INT 无施密特触发输入 (0.5 V_{DD})
3V	--	--	1.3				
I_{OH}	输出驱动(Drive)电流	5V	--	-16	--	mA	$V_{OH}=4.0V$
		3V	--	-10	--		$V_{OH}=2.0V$
I_{OL}	输出灌(Normal	5V	--	25	--	mA	$V_{OL}=1.0V$

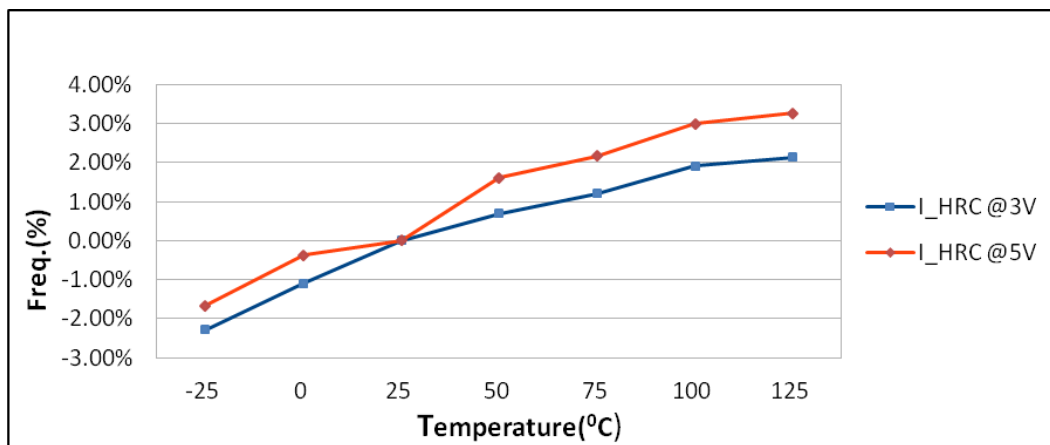
符号	参数	V _{DD}	最小值	典型值	最大值	单位	条件
	Sink)电流	3V	--	15	--	mA	V _{OL} =1.0V
	输出灌(Large Sink)电流	5V	--	40	--		
I _{IR}	红外输出灌电流(Sink)	5V	--	40	--	mA	V _{OL} =1.0V, Large Sink On.
		3V	--	22	--		
I _{OP}	工作电流	正常模式					
		5V	--	3.7	--	mA	F _{HOSC} =20MHz @ I _{HRC} /2
		3V	--	2.6	--		
		5V	--	2.9	--	mA	F _{HOSC} =20MHz @ I _{HRC} /4
		3V	--	2.2	--		
		5V	--	3.4	--	mA	F _{HOSC} =16MHz @ I _{HRC} /2
		3V	--	2.5	--		
		5V	--	2.8	--	mA	F _{HOSC} =16MHz @ I _{HRC} /4
		3V	--	2.1	--		
		5V	--	2.8	--	mA	F _{HOSC} =8MHz @ I _{HRC} /2
		3V	--	2.1	--		
		5V	--	2.3	--	mA	F _{HOSC} =8MHz @ I _{HRC} /4
		3V	--	1.9	--		
		5V	--	2.3	--	mA	F _{HOSC} =4MHz @ I _{HRC} /2
		3V	--	1.9	--		
		5V	--	0.8	--	mA	F _{HOSC} =4MHz @ I _{HRC} /4
		3V	--	0.7	--		
		5V	--	0.5	--	mA	F _{HOSC} =1MHz @ I _{HRC} /2
		3V	--	0.4	--		
		5V	--	0.3	--	mA	F _{HOSC} =1MHz @ I _{HRC} /4
		3V	--	0.3	--		
		慢速模式					
		5V	--	16	--	uA	F _{HOSC} 关闭, F _{LOSC} =32KHz @ I _{LRC} /2
		3V	--	13	--		
		5V	--	10	--	uA	F _{HOSC} 关闭, F _{LOSC} =32KHz @ I _{LRC} /4
		3V	--	7	--		
I _{STB}	待机电流	5V	--	3.0	--	uA	待机模式, F _{HOSC} 关闭, F _{LOSC} =32KHz @ I _{LRC} /4
		3V	--	1.3	--		
I _{HALT}	睡眠电流	5V	--	--	0.5	uA	睡眠模式, 关闭WDT
		3V	--	--	0.2		
		5V	--	--	5	uA	睡眠模式, 开启WDT
		3V	--	--	2		
R _{PH}	上拉电阻	5V	--	53	--	kΩ	上拉电阻
		3V	--	106	--		
R _{PL}	下拉电阻	5V	--	57	--	kΩ	下拉电阻
		3V	--	114	--		

6.3 特性图

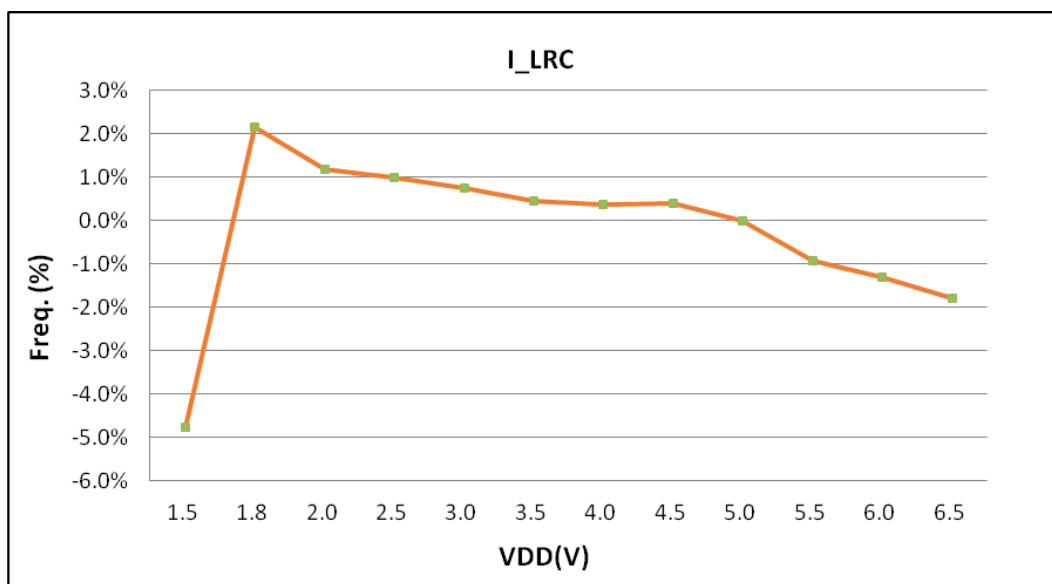
6.3.1 高速 RC 振荡频率与电源电压曲线图



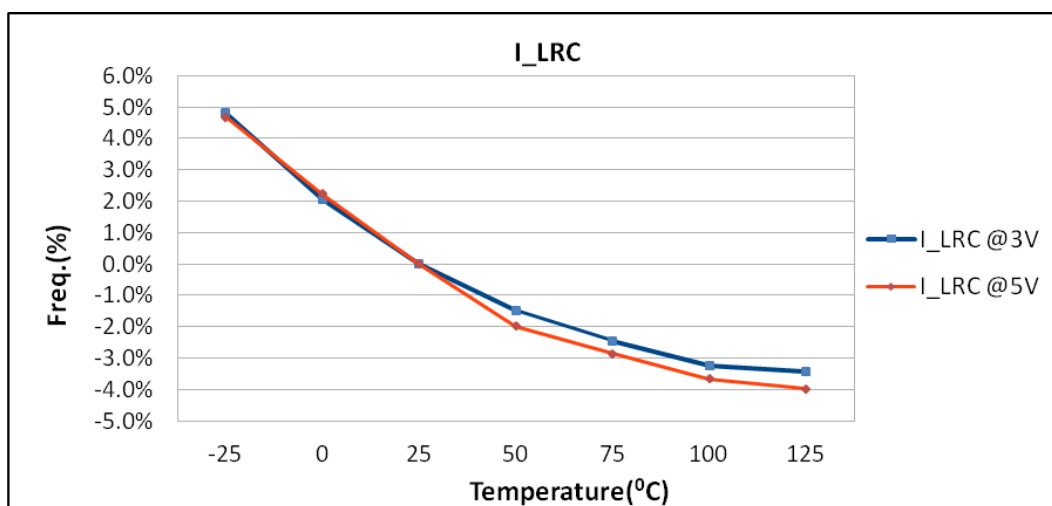
6.3.2 高速 RC 振荡频率与温度曲线图



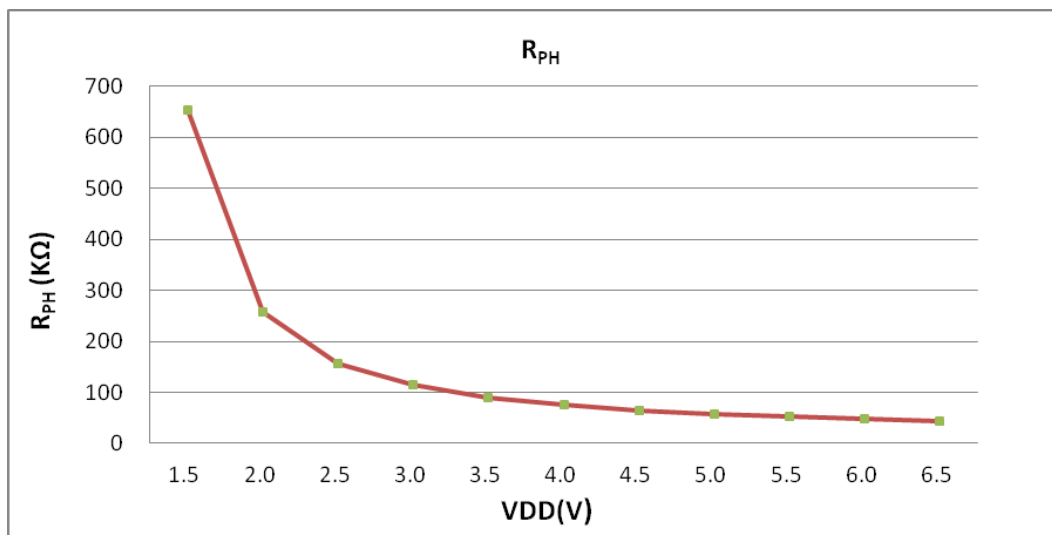
6.3.3 低速 RC 振荡频率与电源电压曲线图



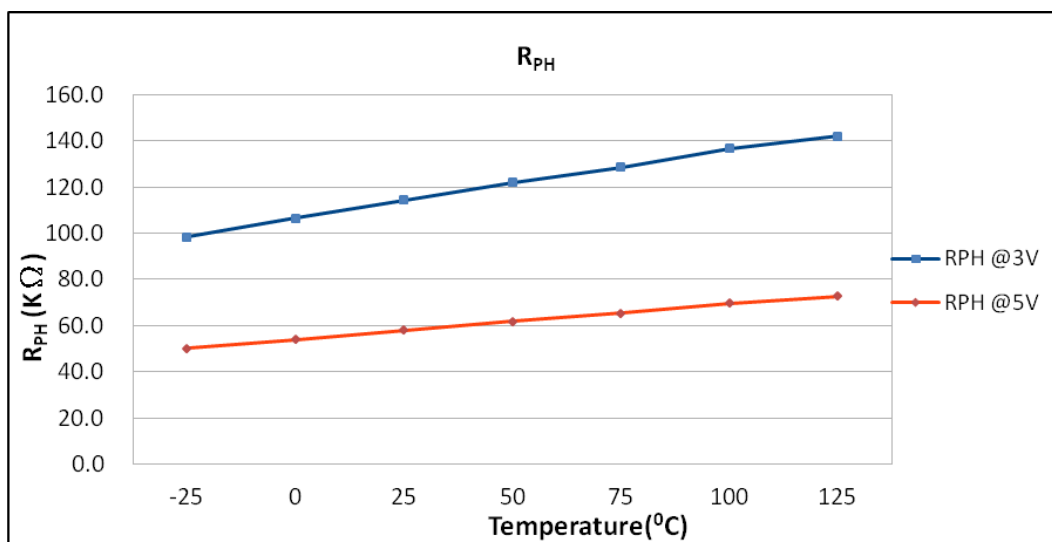
6.3.4 低速 RC 振荡频率与温度曲线图



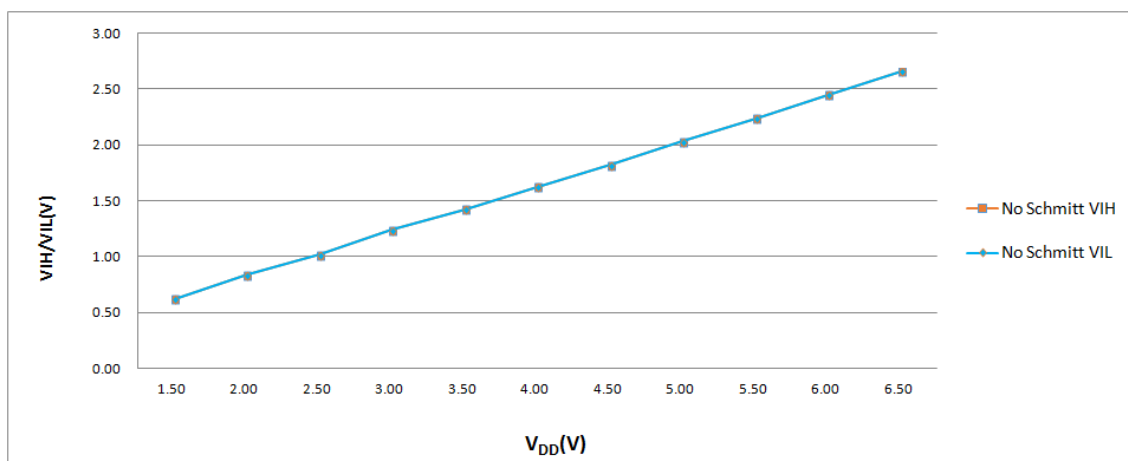
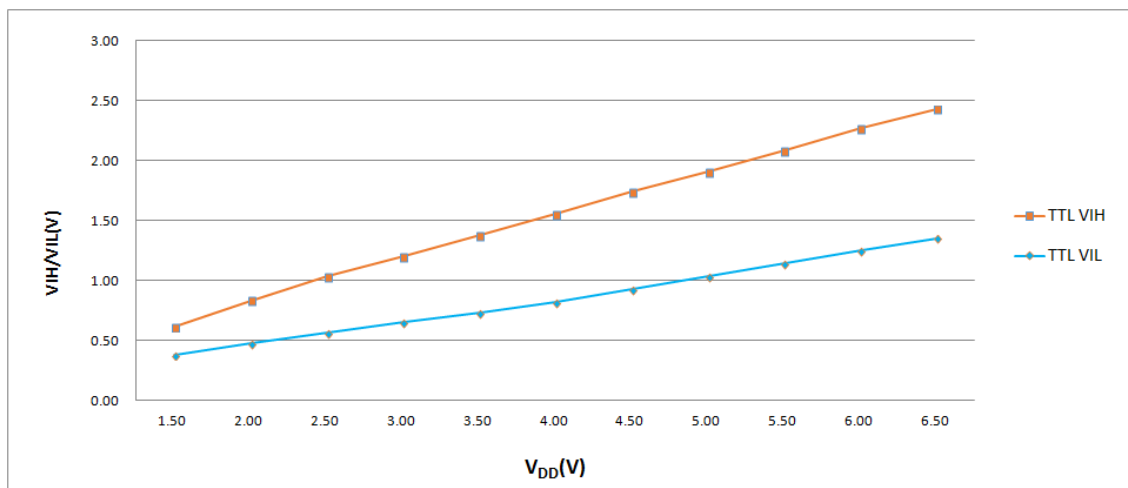
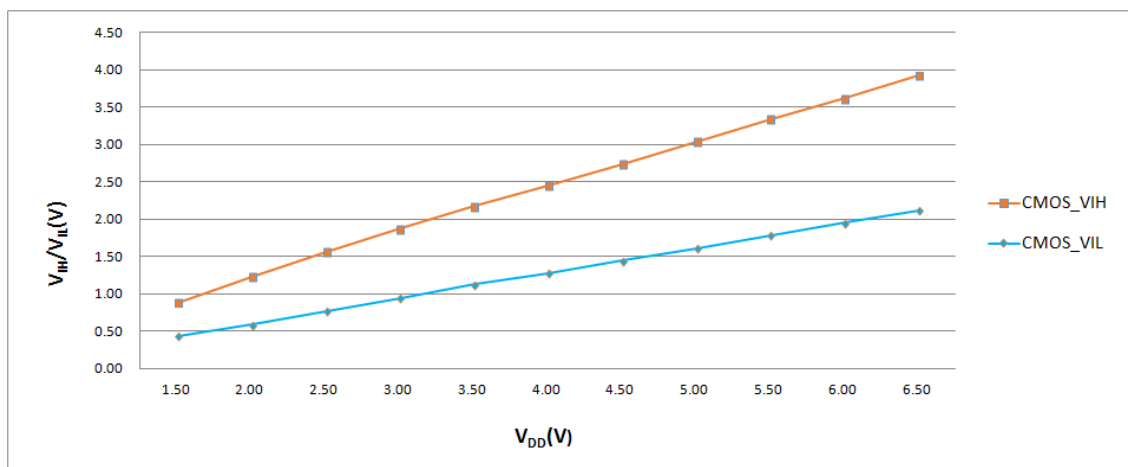
6.3.5 上拉电阻与电源电压曲线图

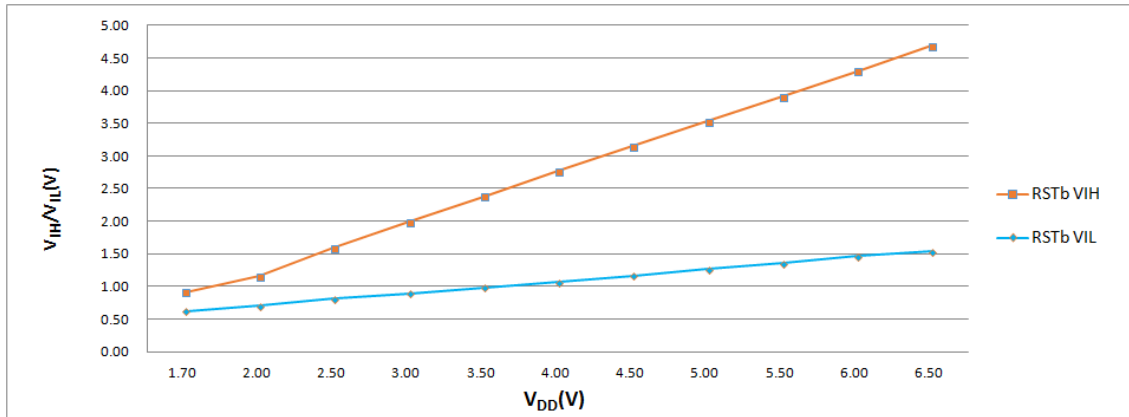


6.3.6 上拉电阻与温度曲线图

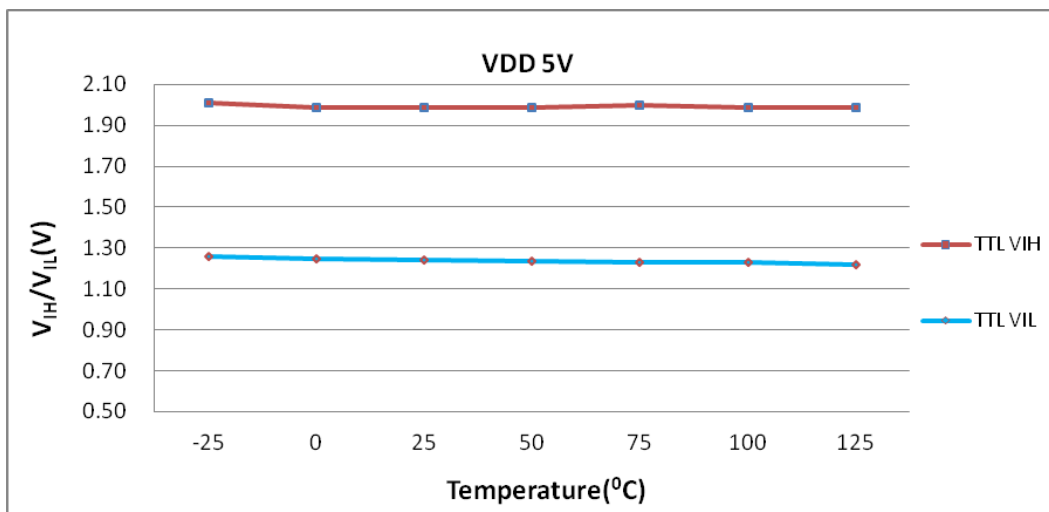
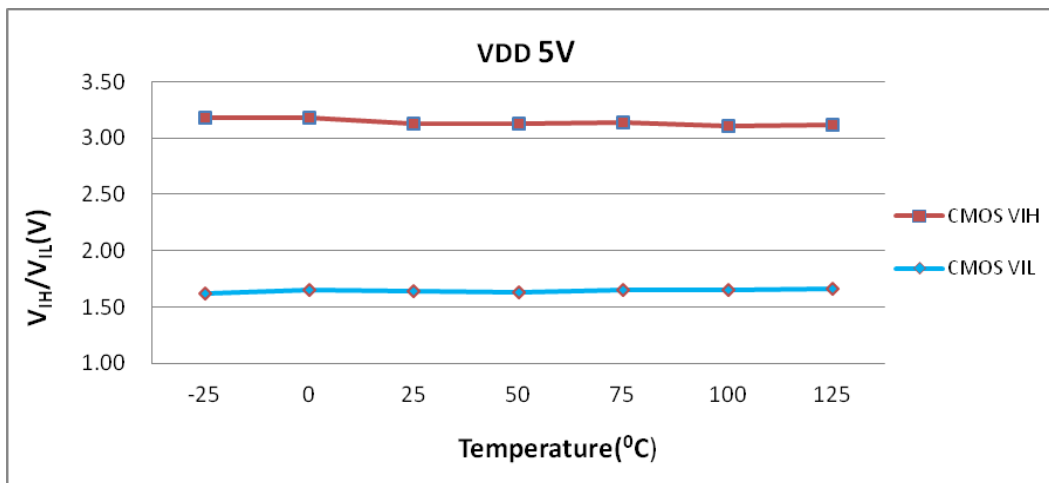


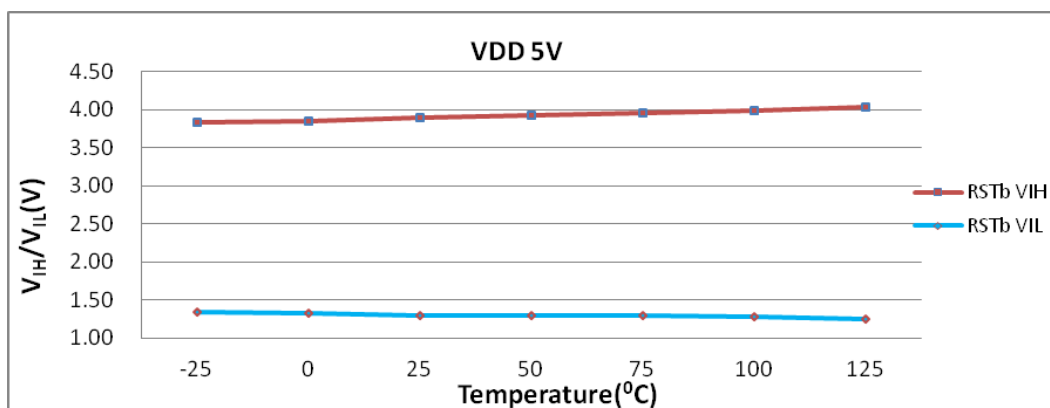
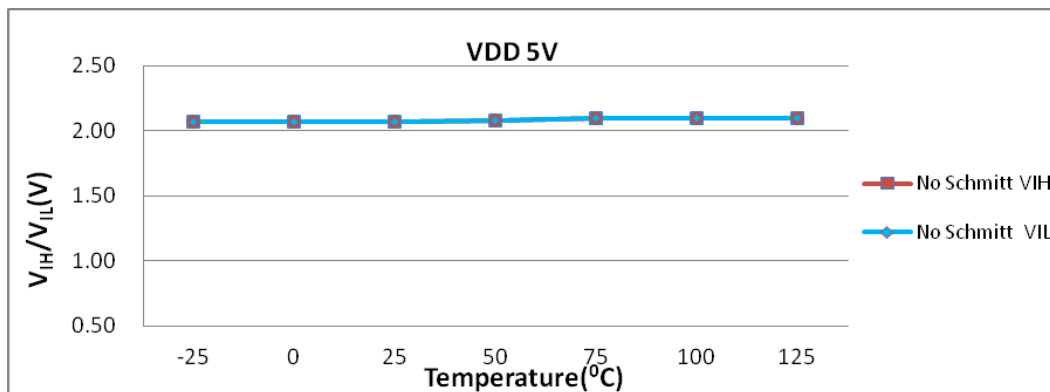
6.3.7 输入高电平/输入低电平与电源电压曲线图





6.3.8 输入高电平/输入低电平与温度曲线图



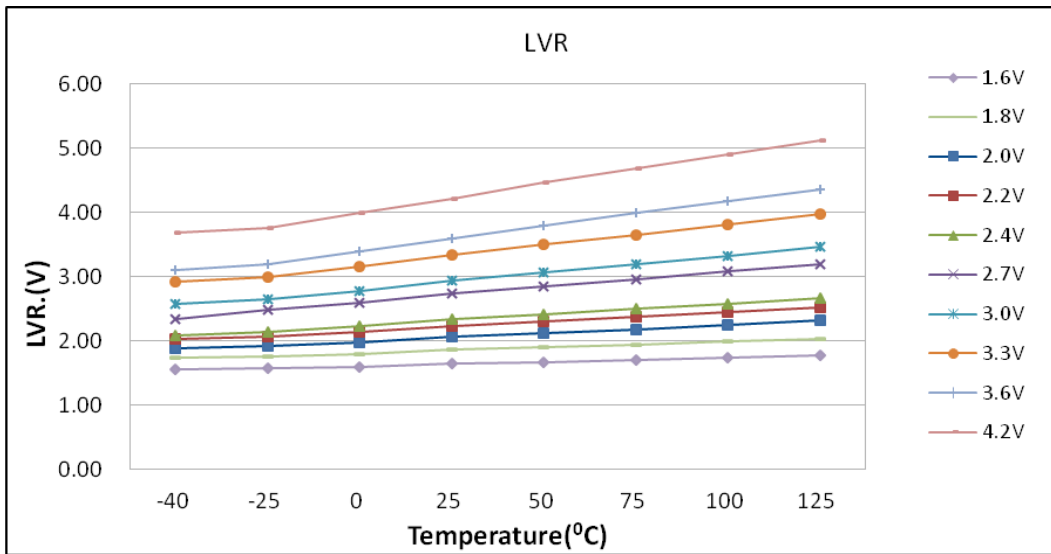


6.4 建议工作电压

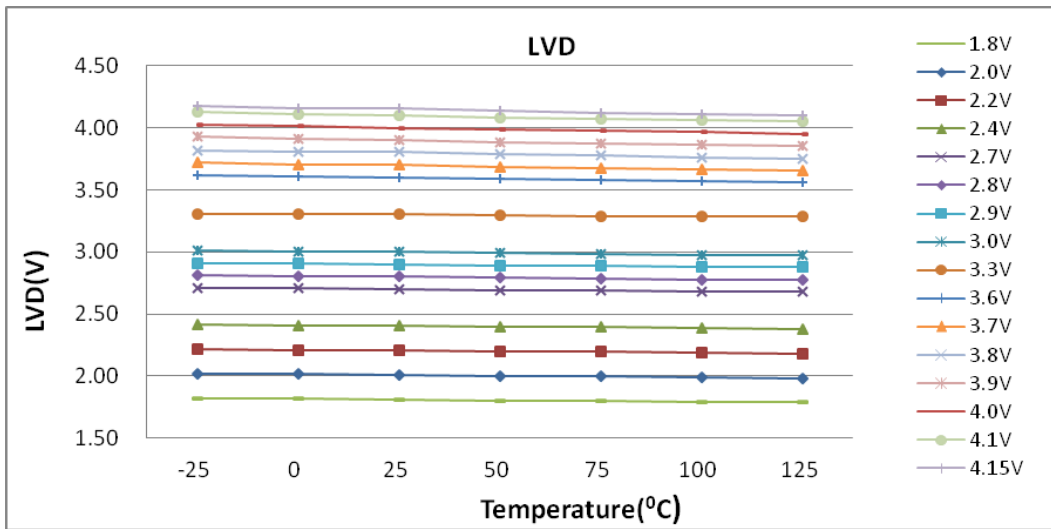
建议工作电压（温度范围：-40 °C ~ +85 °C）

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	3.0V	5.5V	3.0V	3.4V
16M/2T	2.6V	5.5V	3.0V	3.0V
20M/4T	2.1V	5.5V	2.4V	2.4V
16M/4T or 8M/2T	1.9V	5.5V	2.2V	2.4V
8M/4T or 4M/2T	1.5V	5.5V	2.0V	2.2V
4M/4T or 2M/2T	1.5V	5.5V	2.0V	2.2V
2M/4T or 1M/2T	1.5V	5.5V	2.0V	2.2V
1M/4T	1.5V	5.5V	2.0V	2.2V

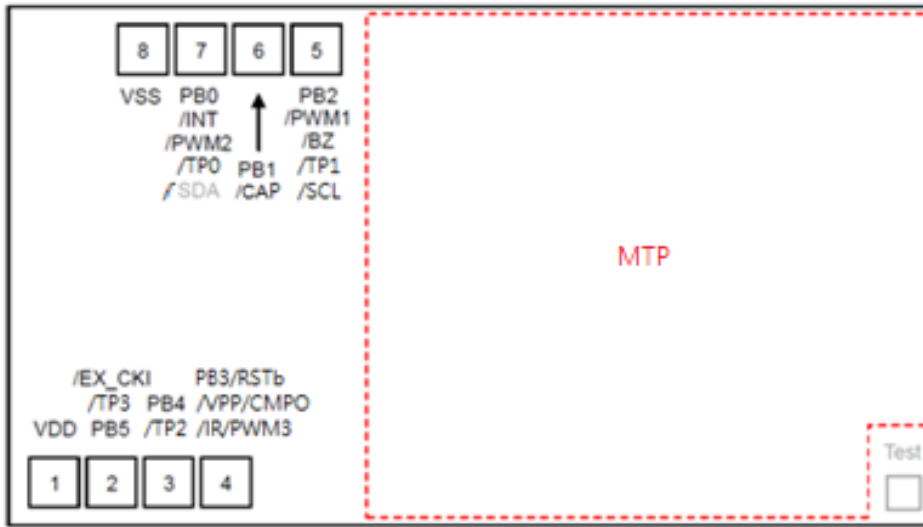
6.5 LVR电压与温度曲线图



6.6 LVD电压与温度曲线图

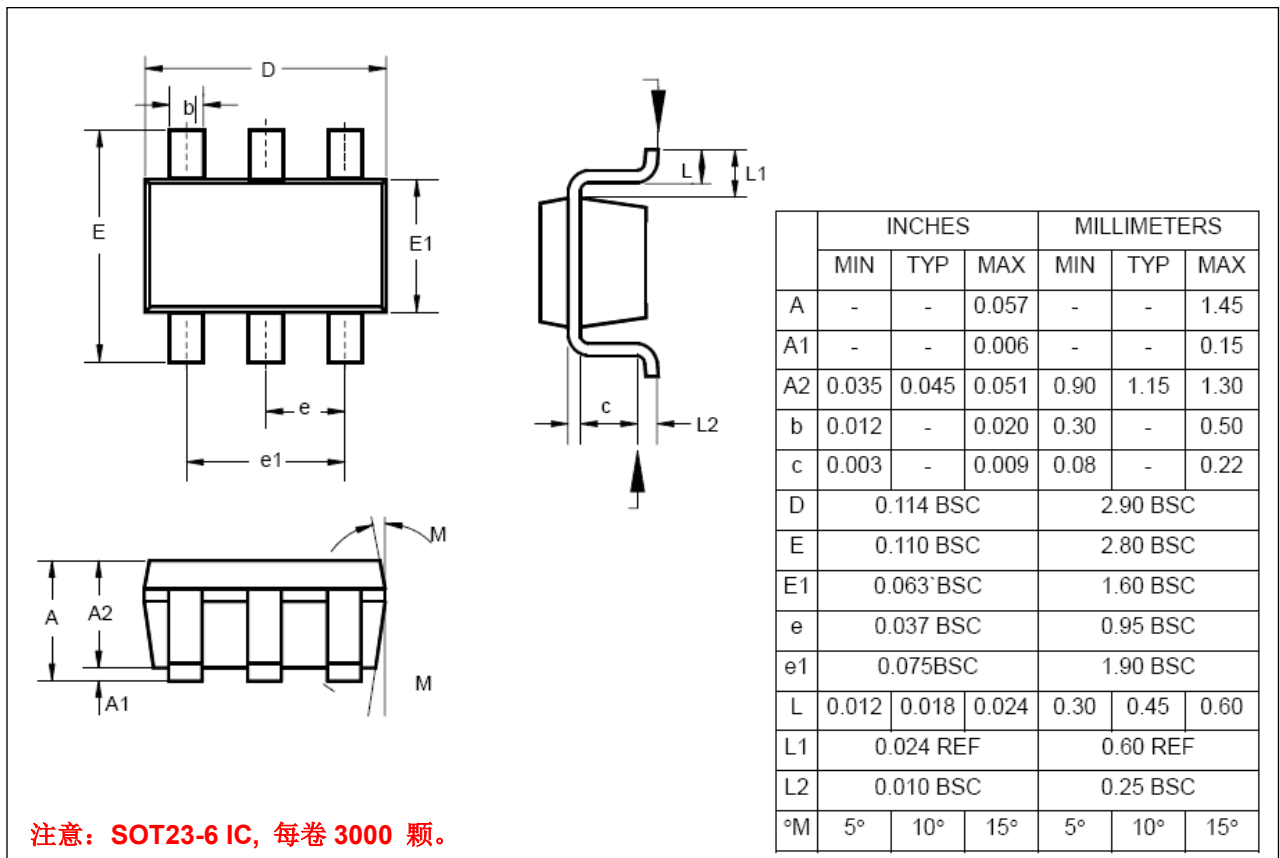


7. 芯片脚位坐标图

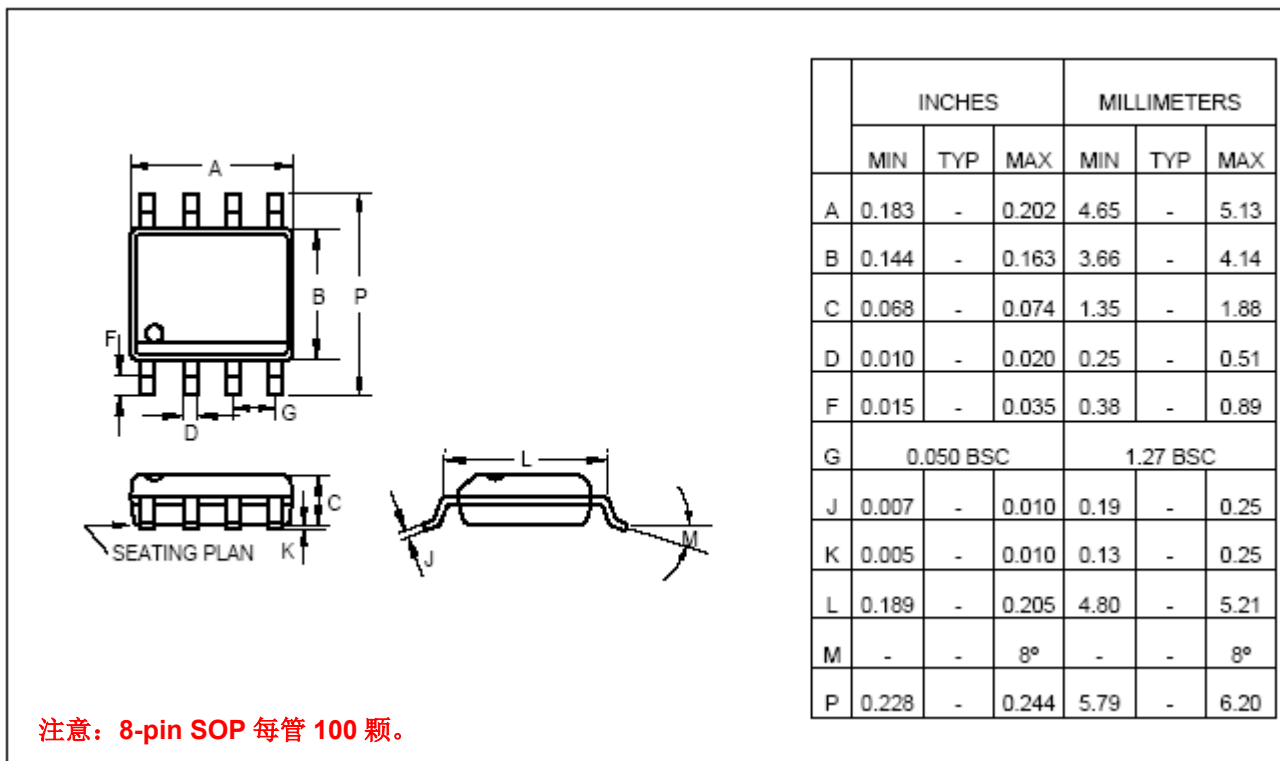


8. 封装尺寸

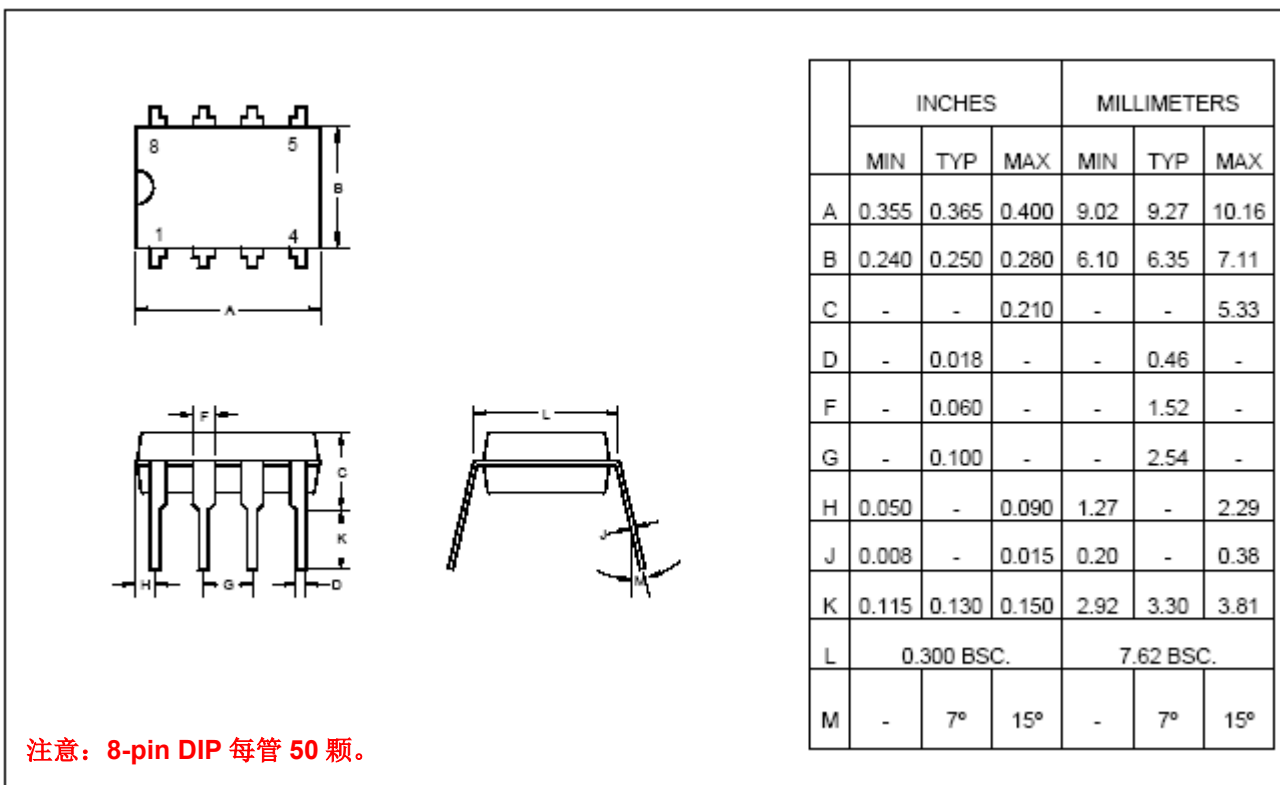
8.1 6 引脚SOT23-6 (63 mil)



8.2 8 引脚SOP (150 mil)



8.3 8 引脚 DIP (300 mil)



9. 订购信息

产品名称	封装类型	引脚数	封装尺寸	配送方式
NY8TM52D	Die	--	--	--
NY8TM52DS6	SOT23-6	6	63 mil	<u>卷装</u> : 每卷 3K颗
NY8TM52DS8	SOP	8	150 mil	<u>卷装</u> : 每卷 2.5K颗 <u>管装</u> : 每管 100 颗
NY8TM52DP8	PDIP	8	300 mil	<u>管装</u> : 每管 50 颗