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

数
据
手
册

NY8A052E

14 I/O 8 位 EPROM-Based 单片机

Version 1.2

Nov. 28, 2024

改 版 记 录

版本	日期	内 容 描 述	修正页
1.0	2023/08/22	初版。	-
1.1	2024/07/05	修改T1CR1 初始值	26
1.2	2024/11/28	更新 “表 13　内部参考电压选择表”数值。	49

目 录

修改T1CR1 初始值 2

1. 概述 7

1.1 功能 7

1.2 NY8A052E 与 NY8A053E 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 11

2. 内存结构 12

2.1 程序存储器 12

2.2 数据存储器 13

3. 功能概述 16

3.1 R-page特殊功能寄存器 16

3.1.1 INDF（间接寻址寄存器） 16

3.1.2 TMR0（定时器 0 寄存器） 16

3.1.3 PCL（程序计数器低字节） 16

3.1.4 STATUS（状态寄存器） 17

3.1.5 FSR（数据指针寄存器） 17

3.1.6 PortA（PortA数据寄存器） 17

3.1.7 PortB（PortB数据寄存器） 18

3.1.8 PCON（Power寄存器） 18

3.1.9 BWUCON（PortB唤醒控制寄存器） 19

3.1.10 PCHBUF（程序计数器高字节） 19

3.1.11 ABPLCON（PortA/PortB下拉电阻控制寄存器） 19

3.1.12 BPHCON（PortB上拉电阻控制寄存器） 20

3.1.13 INTE（中断使能寄存器） 20

3.1.14 INTF（中断标志寄存器） 21

3.2 T0MD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 IOSTA（PortA I/O控制寄存器） 22

3.3.2 IOSTB（PortB I/O控制寄存器） 23

3.3.3 APHCON（PortA上拉电阻控制寄存器） 23

3.3.4 PS0CV（预分频器 0 寄存器）...... 23

3.3.5 BODCON（PortB开漏控制寄存器）...... 24

3.3.6 CMPCR（比较器控制寄存器）...... 24

3.3.7 PCON1（Power控制寄存器 1）...... 25

3.4 S-page特殊功能寄存器......26

3.4.1 TMR1（定时器 1 寄存器）...... 26

3.4.2 T1CR1（定时器 1 控制寄存器 1）...... 26

3.4.3 T1CR2（定时器 1 控制寄存器 2）...... 27

3.4.4 PWM1DUTY（PWM1 占空比寄存器）...... 28

3.4.5 PS1CV（预分频器 1 寄存器）...... 28

3.4.6 BZ1CR（蜂鸣器 1 控制寄存器）...... 28

3.4.7 IRCR（IR控制寄存器）...... 29

3.4.8 TBHP（表格指针高字节寄存器）...... 30

3.4.9 TBHD（表格数据高字节寄存器）...... 30

3.4.10 P2CR1（PWM2 控制寄存器 1）...... 30

3.4.11 PWM2DUTY（PWM2 占空比寄存器）...... 31

3.4.12 TMRH（定时器 1 高字节寄存器）...... 31

3.4.13 TM34RH（PWM3/4 高字节寄存器）...... 31

3.4.14 OSCCR（振荡器控制寄存器）...... 31

3.4.15 P3CR1（PWM3 控制寄存器 1）...... 32

3.4.16 PWM3DUTY（PWM3 占空比寄存器）...... 32

3.4.17 P4CR1（PWM4 控制寄存器 1）...... 33

3.4.18 PWM4DUTY（PWM4 占空比寄存器）...... 33

3.5 I/O Port 33

3.5.1 IO引脚结构框图...... 35

3.6 定时器 0 42

3.7 定时器 1/PWM1/Buzzer1 43

3.8 PWM2/3/4 45

3.9 红外载波 45

3.10 低电压侦测（LVD）......46

3.11 电压比较器......47

3.11.1 比较器内部参考电压 (Vref)...... 48

3.12 看门狗定时器（WDT）......50

3.13 中断......51

3.13.1 Timer0 上溢中断 51
3.13.2 Timer1 下溢中断 52
3.13.3 看门狗超时中断 52
3.13.4 PB输入状态改变中断 52
3.13.5 外部中断输入 52
3.13.6 LVD 中断 52
3.14 振荡器配置 52
3.15 工作模式 54
3.15.1 正常模式 56
3.15.2 慢速模式 56
3.15.3 待机模式 56
3.15.4 睡眠模式 57
3.15.5 唤醒稳定时间 57
3.15.6 工作模式概述 57
3.16 复位 58
4. 指令集 60
5. 配置字节表 76
6. 电气特性 78
6.1 最大绝对值 78
6.2 直流电气特性 78
6.3 OSC电气特性 80
6.4 比较器/LVD电气特性 80
6.5 特性曲线图 80
6.5.1 高速RC振荡频率 (I_HRC) 与电源电压(VDD) 曲线图 80
6.5.2 频率与温度曲线图 81
6.5.3 低速RC振荡频率(I_LRC) 与电源电压(VDD) 曲线图 81
6.6 建议工作电压 81
6.7 LVR电压与温度曲线图 82
7. 晶圆脚位图 82
8. 封装尺寸 83
8.1 8 引脚SOP（150 毫寸） 83
8.2 14 引脚SOP（150 毫寸） 83

8.3 16 引脚SOP（150 毫寸） 84

9. 订购信息 84

1. 概述

NY8A052E是以EPROM作为内存的 8 位单片机，专为多IO产品的应用而设计，例如遥控器、风扇/灯光控制或是游乐器周边等等。采用CMOS制程并同时提供客户低成本、高性能等显著优势。NY8A052E核心建立在RISC精简指令集架构可以很容易地做编辑和控制，共有 55 条指令。除了少数指令需要 2 个时序，大多数指令都是 1 个时序即能完成，可以让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，NY8A052E有 14 根弹性的双向I/O脚，每个I/O脚都有单独的寄存器控制为输入或输出脚。而且每一个I/O脚位都有附加的程控功能如上拉或下拉电阻或开漏极(Open-Drain) 输出。此外针对红外线摇控的产品方面，NY8A052E内建了可选择频率的红外载波发射口。

NY8A052E有两组定时器，可用系统频率当作一般的计时的应用或者从外部讯号触发来计数。另外NY8A052E提供 4 组 10 位分辨率的PWM输出或者蜂鸣器输出，可用来驱动马达、LED、或蜂鸣器等等。

NY8A052E采用双时钟机制，高速振荡或者低速振荡都可以分别选择内部RC振荡或外部Crystal输入。在双时钟机制下，NY8A052E可选择多种工作模式如正常模式(Normal)、慢速模式(Slow mode)、待机模式(Standby mode) 与睡眠模式(Halt mode)，可节省电力消耗，延长电池寿命。并且单片机在使用内部RC高速振荡时，低速振荡可以同时使用外部精准的晶振计时。可以维持高速处理同时又能精准计算真实时间。

在省电的模式下如待机模式(Standby mode)与睡眠模式(Halt mode)中，有多个中断源可以触发中断唤醒NY8A052E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

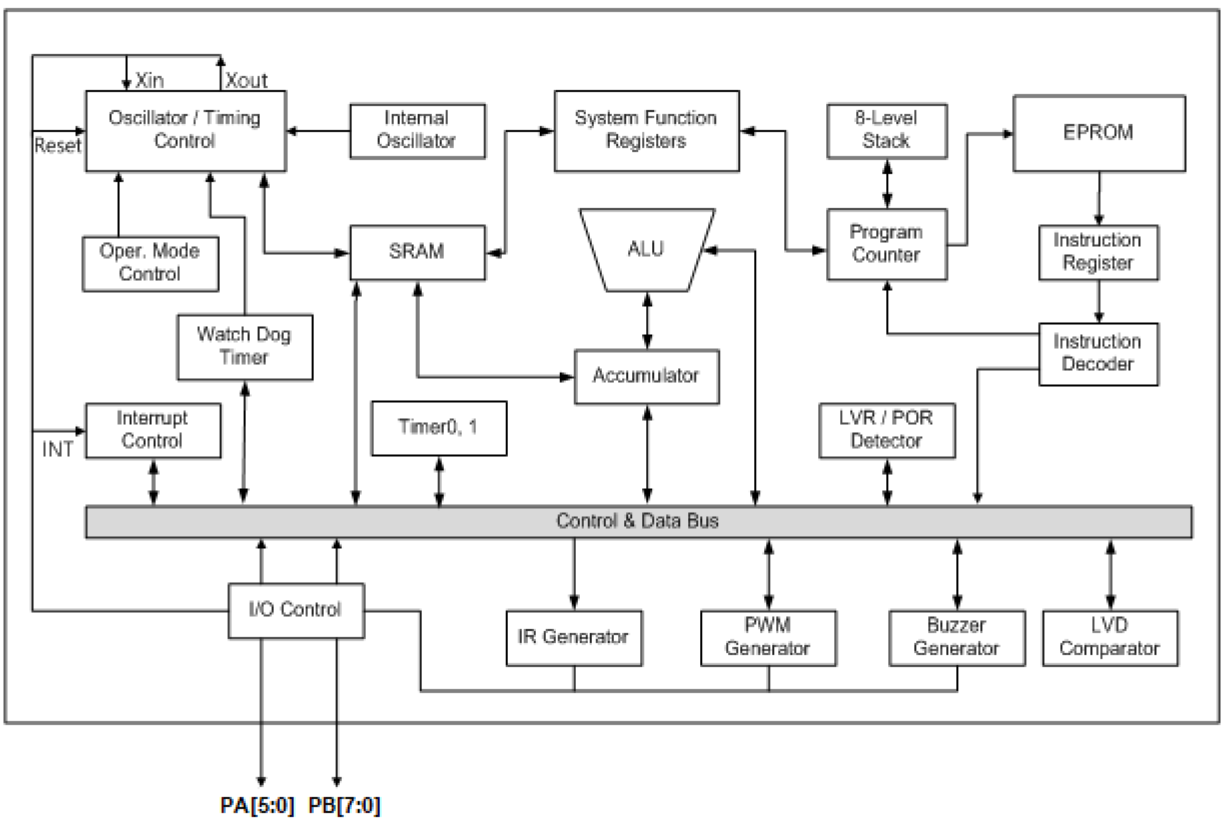
- 宽广的工作电压：
  - 2.0V ~ 5.5V @系统频率≦8MHz。
  - 2.2V ~ 5.5V @系统频率＞8MHz。
- 宽广的工作温度：-40°C ~ 85°C。
- 1.5Kx14 位的程序存储器空间。
- 96 字节的通用数据存储器空间。
- 内建准确的低电压侦测电路(LVD)。14 级电压选择 (2.2V ~ 4.15V)
- 内建准确的电压比较器(Voltage Comparator)。
- 14 根可分别单独控制输入输出方向的I/O脚(GPIO)、PA[5:0]、PB[7:0]。
- PA[5:0]及PB[3:0]可选择输入时使用内建下拉电阻。
- PB[7:0]可选择上拉电阻或开漏极输出(Open-Drain)。
- 8 层程序堆栈(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)可选重复载入或连续下数计时。

- 4 路 10 位的脉冲宽度调变输出(PWM1~PWM4)，PWM1/2/3/4 共享Timer1。
- 一个蜂鸣器输出(BZ1)。
- 38/57KHz红外线载波频率可供选择，同时载波之极性也可以根据数据作选择。
- 红外线载波发射口。
- 内建上电复位电路(POR)。
- 内建低压复位功能(LVR)。
- 内建看门狗计时(WDT)，可由程序韧体控制开关。
- 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: E_HXT (超过 6MHz外部高速石英振荡)
E_XT (455K~6MHz外部石英振荡)
I_HRC (1~20MHz内部高速RC振荡)
  - 低速振荡: E_LXT (32KHz外部低速石英振荡)
I_LRC (内部 32KHz低速RC振荡)
- 四种工作模式可随系统需求调整电流消耗：正常模式(Normal)、慢速模式(Slow mode)、待机模式(Standby mode) 与 睡眠模式(Halt mode)。
- 六种硬件中断：
  - 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。
- NY8A052E在待机模式(Standby mode)下的六种唤醒中断：
  - 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。
- NY8A052E在睡眠模式(Halt mode)下的三种唤醒中断：
  - WDT 中断。
  - PB 输入状态改变中断。
  - 外部中断输入。

1.2 NY8A052E 与 NY8A053E 的主要差异

序号	功 能	NY8A052E	NY8A053E
1.	ROM	1.5Kx14	1Kx14
2	SRAM	96 bytes	64 bytes
3	IO 数	14 (增加 PA4，PA5)	12
4	PortA 上拉电阻	有	没有
5	PB3 输出口	可输出高电平	只能做开漏极输出
6	IO 口输出小电流选项	PA4，PA5，PB0~PB3	没有

1.3 系统框图



1.4 引脚图

NY8A052E提供三种封装类型：SOP16，SOP14 及SOP8。

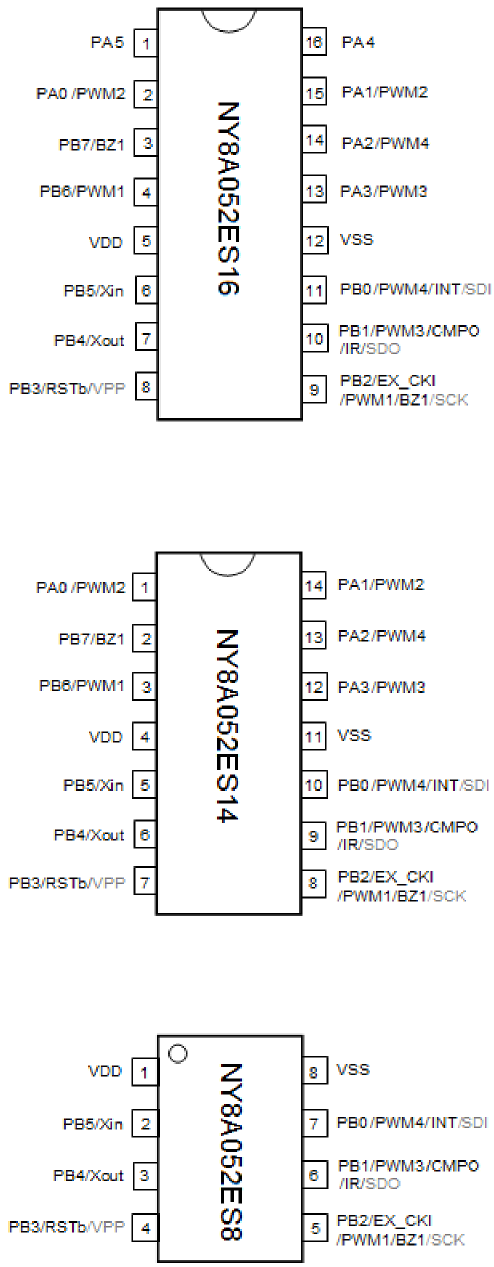


图 1 封装引脚图

1.5 引脚说明

引脚名称	I/O	描述
PA0 PWM2	I/O	PA0 是一个双向I/O引脚。 PA0 依配置字节设定可输出PWM2。
PA1 PWM2	I/O	PA1 是一个双向I/O引脚。 PA1 依配置字节设定可输出PWM2。
PA2 PWM4	I/O	PA2 是一个双向I/O引脚。 PA2 依配置字节设定可输出PWM4。
PA3 PWM3	I/O	PA3 是一个双向I/O引脚。 PA3 依配置字节设定可输出PWM3。
PA4	I/O	PA4 是一个双向I/O引脚。
PA5	I/O	PA5 是一个双向I/O引脚。
PB0 PWM4 INT SDI	I/O	PB0 是一个双向I/O引脚。 PB0 依配置字节设定可输出PWM4。 PB0 是外部中断输入引脚当EIS=1&INTIE=1。 PB0 也是编程数据输入SDI。
PB1 PWM3/ IR/ CMPO SDO	I/O	PB1 是一个双向I/O引脚。 PB1 可输出PWM3，IR或当作比较器输出引脚CMPO。 PB1 也是编程数据输出SDO。
PB2 EX_CKI SCK PWM1 BZ1	I/O	PB2 是一个双向I/O引脚。 PB2 可当作定时器 0 / 1 外部时钟来源EX_CKI。 PB2 也是编程时钟输入SCK。 PB2 依配置字节设定可输出PWM1。 PB2 依配置字节设定可输出BZ1。 PWM比BZ1 的优先级高。
PB3 RSTb VPP	I/O	PB3 可选择当作输入脚或开漏输出脚。 PB3 可当作复位引脚RSTb。当RSTb引脚为低电平时，NY8A052E会复位。 当VPP超过 7.75V，NY8A052E会进入EPROM烧写模式。
PB4 Xout	I/O	PB4 是一个双向I/O引脚，当用于I_HRC与I_LRC。 PB4 也可以当成指令时钟输出。 PB4 可当作晶振输出引脚Xout，当用于E_HXT，E_XT与E_LXT。
PB5 Xin	I/O	PB5 是一个双向I/O，引脚当用于I_HRC与I_LRC。 PB5 可当作晶振输入引脚Xin，当用于E_HXT，E_XT与E_LXT。
PB6 PWM1	I/O	PB6 是一个双向I/O引脚。 PB6 依配置字节设定可输出PWM1。
PB7 BZ1	I/O	PB7 是一个双向I/O引脚。 PB7 依配置字节设定可输出BZ1。
VDD	-	电源正极。
VSS	-	电源负极。

2. 内存结构

NY8A052E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

NY8A052E程序存储器空间是 1.5Kx14 位。因此，11 位宽的程序计数器（PC）可以访问程序存储器的任何地址。

程序存储器的一些位置保留给中断入口。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。

NY8A052E提供CALL、GOTOA和CALLA等指令去访问程序空间的256个地址。还提供GOTO指令去访问程序空间512个地址，LCALL和LGOTO指令访问程序空间的任何地址。

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

NY8A052E程序存储器地址 0x5FE~0x5FF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。

NY8A052E程序存储器地址 0x00E~0x00F是默认滚码地址。如果用户在不设置滚码时可当作程序区使用。

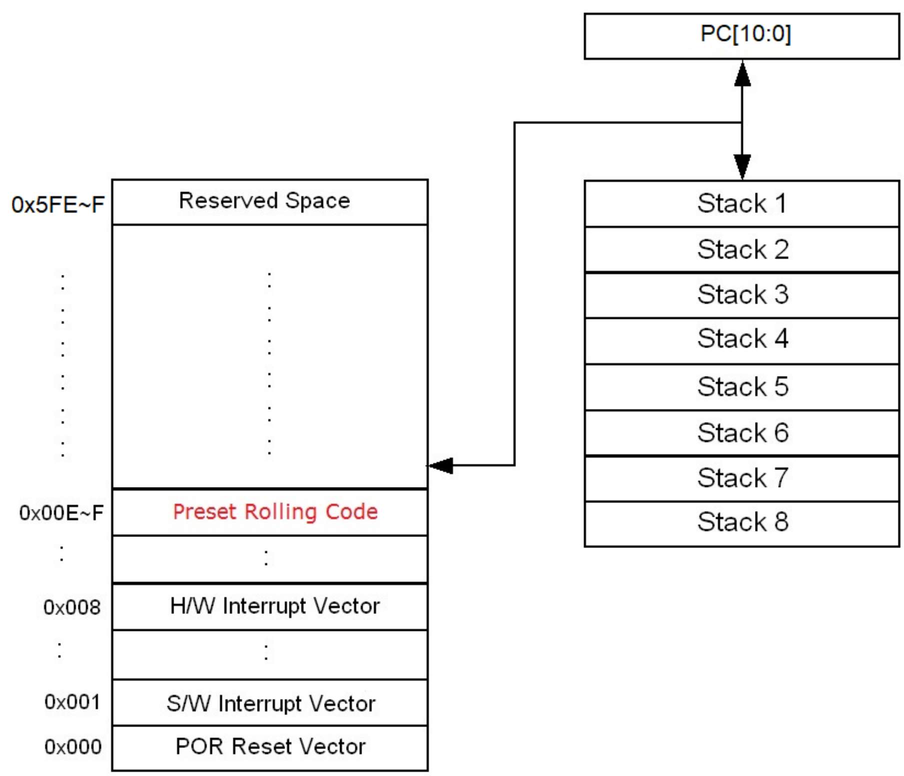


图 2　程序存储器对映地址

2.2 数据存储器

根据用于访问数据存储器的指令，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特殊功能寄存器可用直接寻址方式和间接寻址方式来进行访问。

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。

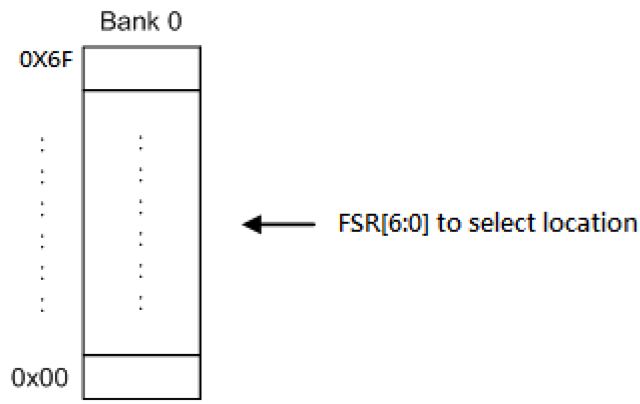


图 3　间接寻址方式访问数据存储器

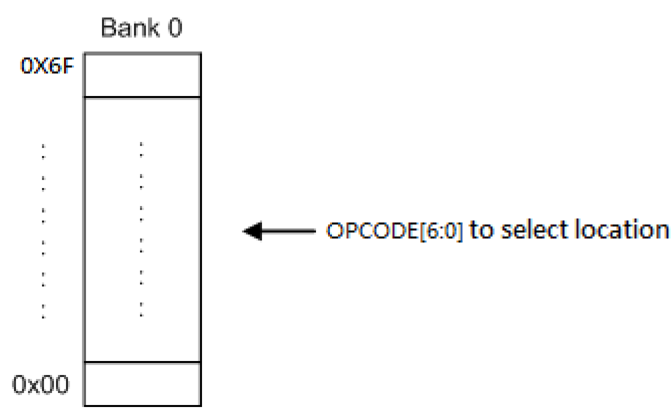


图 4　直接寻址方式访问数据存储器

R-page特殊功能寄存器可以通过一般的指令访问，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 0xF。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 0x10 到 0x6F地址。

NY8A052E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	(Bank 0)
0x0	INDF
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 ~ 0x6F	通用寄存器

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访问，S-page特殊功能寄存器只能被指令SFUN和SFUNR访问。

寄存器名称和F-page、S-page的地址说明如下表。

特殊功能寄存器种类 地址	F-page SFR	S-page SFR
0x0	-	TMR1
0x1	-	T1CR1
0x2	-	T1CR2
0x3	-	PWM1DUTY
0x4	-	PS1CV
0x5	IOSTA	BZ1CR
0x6	IOSTB	IRCR
0x7	-	TBHP

特殊功能寄存器种类 地址	F-page SFR	S-page SFR
0x8	-	TBHD
0x9	APHCON	-
0xA	PS0CV	P2CR1
0xB	-	-
0xC	BODCON	PWM2DUTY
0xD	-	TMRH
0xE	CMPCR	TM34RH
0xF	PCON1	OSCCR
0x10		
0x11		P3CR1
0x12		
0x13		PWM3DUTY
0x14		
0x15		
0x16		P4CR1
0x17		
0x18		PWM4DUTY

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NY8A052E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。

通过设置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_CKI或低频振荡器I_LRC/E_LXT中择一。

3.1.3 PCL（程序计数器低字节）

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

程序计数器（PCL）是一个 11 位寄存器。PCL值会随着指令一个一个被执行时逐渐增加，除了某些指令会直接更改PC数值。PC高字节（PC[10:8]），不能被直接访问，必须通过PCHBUF寄存器完成访问。

以GOTO指令来说，PC[8:0]是从指令码取得，而PC[10:9]是从PCHBUF[2:1]加载。CALL指令的PC[7:0]是从指令码取得，而PC[10:8]是从PCHBUF[2:0]加载。下一个PC地址（PC+1），将会存到堆栈的顶部。LGOTO指令的PC[10:0]是从指令码取得。

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

3.1.4 STATUS（状态寄存器）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
STATUS	R	0x3	GP7	GP6	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CLRWDT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CLRWDT或SLEEP指令后。

/TO=0 时，发生WDT上溢。

GP7，GP6，GP5：通用寄存器数据位。

(*1) 可以被SLEEP指令清除。

(*2) 可以由CLRWDT指令设定。

3.1.5 FSR（数据指针寄存器）

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

FSR[6:0]：从指定Bank数据存储器的 96 个寄存器中选择一个。

GP7：通用寄存器数据位。

3.1.6 PortA（PortA 数据寄存器）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PortA	R	0x5	GP7	GP6	PA5	PA4	PA3	PA2	PA1	PA0

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（PA5~PA0）

读取PortA时，若特定脚位被配置为输入脚或输出脚，将得到该脚位输入状态。当写入PortA时，数据是被写入PortA的输出数据锁存器中。

GP7~6：通用寄存器数据位。

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	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
读/写属性			读/写							
初始值			数据锁存器值是xxxxxxxx，读取值则是xxxxxxxx端口值（PB7~PB0）							

读取PortB时，若特定脚位被配置为输入脚或输出脚，将得到该脚位输入状态。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	EIS	LVDEN	GP4	LVREN	CMPEN	GP1	GP0
读/写属性			读/写							
初始值			1	0	0	0	1	0	0	0

GP4，GP1，GP0：通用寄存器数据位。

CMPEN：开启/关闭比较器。

CMPEN=1 时，开启比较器。

CMPEN=0 时，关闭比较器。

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。

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。

WDTEN：开启/关闭 WDT。

WDTEN=1 时，开启WDT。

WDTEN=0 时，关闭WDT。

3.1.9 BWUCON（PortB 唤醒控制寄存器）

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BWUCON	R	0x9	WUPB7	WUPB6	WUPB5	WUPB4	WUPB3	WUPB2	WUPB1	WUPB0
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

WUPBx：开启/关闭PBx唤醒功能，0 ≤ x ≤ 7。

WUPBx=1 时，开启PBx唤醒功能。

WUPBx=0 时，关闭PBx唤醒功能。

3.1.10 PCHBUF（程序计数器高字节）

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

PCHBUF[2:0]：程序计数器PC的第 10 个位，第 9 个位和第 8 个位。

XSPD_STP：写 1 停止外部晶振 32.768KHz起振强化功能。

3.1.11 ABPLCON（PortA/PortB 下拉电阻控制寄存器）

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

/PLPAx：关闭/开启PAx下拉电阻，0 ≤ x ≤ 3。

/PLPAx=1，关闭PAx下拉电阻。

/PLPAx=0，开启PAx下拉电阻。

/PLPBx：关闭/开启PBx下拉电阻，0 ≤ x ≤ 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	/PHPB7	/PHPB6	/PHPB5*	/PHPB4*	/PHPB3	/PHPB2	/PHPB1	/PHPB0
读/写属性			读/写							
初始值			1	1	1	1	1	1	1	1

/PHPBx：关闭/开启PBx上拉电阻，0 ≤ x ≤ 7。

/PHPBx=1 时，关闭PBx上拉电阻。

/PHPBx=0 时，开启PBx上拉电阻。

注意：PB4 与PB5 作为晶振引脚时应该关闭内置上拉电阻。

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	-	WDTIE	-	LVDIE	T1IE	INTIE	PBIE	T0IE
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

T0IE：定时器 0 上溢中断使能位。

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。

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。

3.1.14 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	-	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：PortB输入状态变化中断标志位。

PBIF=1 时，发生PortB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：当对应的INTE寄存器控制位未使能，读取中断标志是 0。

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访问。

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T0MD	-	-	LCKTM0	INTEDG	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，下降沿时定时器 0 加一。

T0CE=0 时，上升沿时定时器 0 加一。

注意：T0CE应用在外部 EX_CKI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_CKI脚或低频振荡I_LRC/E_LXT。

T0CS=0 时，选择指令时钟F_{INST}。

INTEDG：INT沿触发选择位。

INTEDG=1 时，引脚PB0 输入上升沿将触发INTIF。

INTEDG=0 时，引脚PB0 输入下降沿将触发INTIF。

LCKTM0：当T0CS=1 时，定时器 0 时钟源可被选择作低频振荡器。

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，低频振荡I_LRC/E_LXT为定时器 0 时钟源。

注意：有关定时器 0 时钟源选择的详细说明，请参考定时器 0 章节。

3.3 F-page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	IOPA3	IOPA2	IOPA1	IOPA0
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

IOPAx：PAx I/O模式选择，0 ≤ x ≤ 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	IOPB7	IOPB6	IOPB5	IOPB4	IOPB3	IOPB2	IOPB1	IOPB0
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

IOPBx：PBx I/O模式选择，0 ≤ x ≤ 7。

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	/PLPA4	/PHPA5	/PHPA4	/PHPA3	/PHPA2	/PHPA1	/PHPA0
读/写属性			读/写							
初始值			1	1	1	1	1	1	1	1

/PHPAx：关闭/开启PAx上拉电阻，x=0~5。

/PHPAx=1 时，关闭 PAx上拉电阻。

/PHPAx=0 时，开启 PAx上拉电阻。

/PLPAx：关闭/开启下拉电阻PAx，x=4~5。

/PLPAx=1 时，关闭 PAx下拉电阻。

/PLPAx=0 时，开启 PAx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			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。

3.3.5 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	ODPB7	ODPB6	ODPB5	ODPB4	ODPB3	ODPB2	ODPB1	ODPB0
读/写属性			读/写							
初始值			0	0	0	0	0	0	0	0

ODPBx：开启/关闭PBx的开漏，x=0~7。

ODPBx=1 时，开启PBx的开漏。

ODPBx=0 时，关闭PBx的开漏。

3.3.6 CMPCR（比较器控制寄存器）

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

NS[1:0]：比较器负输入源选择。

NS[1:0]	负输入源
00	PA1
01	PA3
10	Bandgap (0.65V)
11	Vref

PS[1:0]：比较器正输入源选择。

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

CMPF_INV：比较器输出反相控制位。

CMPF_INV = 1，反相比较器输出。

CMPF_INV = 0，同相比较器输出。

RBIAS_L，RBIAS_H：设置比较器参考电压电平。

(有关于比较器的细节，请看比较器功能说明章节 3.11.1)

注意：在进入睡眠模式前，RBIAS_L和 RBIAS_H必须设置为“00”，避免电流损耗。

3.3.7 PCON1（Power 控制寄存器 1）

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

T0EN：开启/关闭定时器 0。

T0EN=1 时，开启定时器 0。

T0EN=0 时，关闭定时器 0。

LVDS3~0：选择LVD电压。

LVDS[3:0]	电压
0000	---
0001	---
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

LVDOUT：LVD输出位。

GIE：开启/关闭总中断使能位。

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。

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 8 位定时器 1 中的低字节目前计数值。当写入寄存器TMR1 时，它将向定时器 1 写入数据，重新加载寄存器并更新定时器 1 当前内容。

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	-	-	-	T1OS	T1RL	T1EN
读/写属性			读/写	读/写	-	-	-	读/写	读/写	读/写
初始值			0	0	X	X	X	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 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 从重载值下数到 0x00。 当下溢发生，0x3FF被重载至定时器 1 并继续下数。
0	1	定时器 1 从重载的数值下数到 0x00。 当下溢发生，重载值被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，PB6 或PB2 输出PWM1。

PWM1OEN=0，PB6 或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]	预分频比选项
000	1:2
001	1:4
010	1:8
011	1:16
100	1:32
101	1:64
110	1:128
111	1:256

表 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]，否则中断可能会误触发。

/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_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							

此寄存器仅能写入。当定时器 1 被使能并开始倒数计时后，PWM1 输出会保持在非有效位状态。当定时器 1 数值等于PWM1DUTY，PWM1 输出有效位状态直到发生下溢。

定时器 1 重新加载的数值储存在TMR1 寄存器，以用来定义PWM1 帧率，PWM1DUTY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

BZ1FSEL[3:0]	BZ1 频率选项	
	时钟源	预分频比
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 输出频率选项

BZ1EN：开启/关闭蜂鸣器 1 输出。

BZ1EN=1 时，开启蜂鸣器 1。

BZ1EN=0 时，关闭蜂鸣器 1。

3.4.7 IRCR（IR 控制寄存器）

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

IREN：开启/关闭IR载波输出。

IREN=1 时，开启IR载波输出。

IREN=0 时，关闭IR载波输出。

IRF57K：IR载波频率选择。

IRF57K=1 时，IR载波频率是 57KHz。

IRF57K=0 时，IR载波频率是 38KHz。

IRCSEL：IR载波极性选择。

IRCSEL=0 且I/O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。

IROSC358M：选择使用的外部晶振频率的类型。

若选择 I_HRC此位将被忽略。

IROSC358M=1，外部晶振频率用 3.58MHz。

IROSC358M=0，外部晶振频率用 455KHz。

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。

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）
Xtal 3.58MHz	64	96	Xtal 模式 & IROSC358M=1
Xtal 455KHz	8	12	Xtal 模式 & IROSC358M=0

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目标地址，此目标地址是由TBHP[2:0]与ACC组成。ACC是PC[10:0]的低字节，TBHP[2:0]是PC[10:0]的高字节。

3.4.9 TBHD（表格数据高字节寄存器）

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

当指令TABLEA被执行后，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 时，PA1 或PA0 输出PWM2。

PWM2OEN=0 时，PA1 或PA0 是GPIO。

3.4.11 PWM2DUTY（PWM2 占空比寄存器）

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

定时器 1 重新加载的数值储存在TMRH[5:4]与TMR1[7:0]寄存器，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 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	S	0xD	-	-	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。

3.4.13 TM34RH（PWM3/4 高字节寄存器）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TM34RH	S	0xE	-	-	-	-	PWM4 DUTY9	PWM4 DUTY8	PWM3 DUTY9	PWM3 DUTY8
读/写属性			-	-	-	-	读/写	读/写	读/写	读/写
初始值			X	X	X	X	X	X	X	X

PWM3DUTY9~8：PWM3 占空比高 2 位。

PWM4DUTY9~8：PWM4 占空比高 2 位。

3.4.14 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	-	CMPOEN	-	-	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 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。

CMPOEN：开启/关闭比较器输出到PB1 引脚。

CMPOEN=1，开启比较器输出到PB1 引脚。

CMPOEN=0，关闭比较器输出到PB1 引脚。

注意：PB1 作为比较器输出引脚的优先级高于PWM3 或 IR功能。

3.4.15 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 时，PA3 或PB1 输出PWM2。

PWM3OEN=0 时，PA3 或PB1 是GPIO。

3.4.16 PWM3DUTY（PWM3 占空比寄存器）

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

定时器 1 重新加载的数值储存在TMRH[5:4]与TMR1[7:0]寄存器，用来定义PWM3 帧率，TM34RH[1:0]与PWM3DUTY[7:0]寄存器用于定义PWM3 的占空比。

3.4.17 P4CR1（PWM4 控制寄存器 1）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P4CR1	S	0x16	PWM4OEN	PWM4OAL	-		-	-	-	-
读/写属性			读/写	读/写	-	-	-	-	-	-
初始值			0	0	X	X	X	X	X	X

PWM4OAL：定义PWM4 输出有效状态。

PWM4OAL=1 时，PWM4 为低电平有效位输出。

PWM4OAL=0 时，PWM4 为高电平有效位输出。

PWM4OEN：开启/关闭PWM4 输出。

PWM4OEN=1 时，PA2 或PB0 输出PWM2。

PWM4OEN=0 时，PA2 或PB0 是GPIO。

3.4.18 PWM4DUTY（PWM4 占空比寄存器）

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

定时器 1 重新加载的数值储存在TMRH[5:4]与TMR1[7:0]寄存器，用来定义PWM4 帧率，TM34RH[3:2]与PWM4DUTY[7:0]寄存器用于定义PWM4 的占空比。

3.5 I/O Port

NY8A052E提供 14 个I/O口（PA[5:0]和PB[7:0]），用户可以通过寄存器PORTA和PORTB依次读写这些脚位。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[5:0]定义PA[5:0]为输入口或输出口，寄存器IOSTB[7:0]定义PB[7:0]为输入口或输出口。

当一个I/O脚位被配置为输入口时，它可以通过寄存器开启或关闭内部上拉/下拉电阻。寄存器APHCON[5:0]用于开启或关闭PA[5:0]的内部上拉电阻，寄存器BPHCON[7:0]用于开启或关闭PB[7:0]的内部上拉电阻，寄存器ABPLCON[7:4] 则是用于开启或关闭PB[3:0]的内部下拉电阻。

当一个I/O脚位被配置为输出口时，会有一个相应的且单独的寄存器来选择作为开漏极输出脚。寄存器BODCON[7:0]决定PB[7:0]是否为开漏输出脚。

I/O口功能摘要如下表：

功能		PA[5:0]	PB[3:0]	PB[7:4]
输入	上拉电阻	V	V	V
	下拉电阻	V	V	X
输出	开漏输出	X	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[7:0]会使能或禁止任PB脚位的唤醒功能。只要PB的任意一个引脚被BWUCON的对应位选中，则所选引脚发生电平变化时，寄存器位PBIF (INTF[1])置 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。

NY8A052E提供 1 个外部中断，当寄存器EIS（PCON[6]）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。

NY8A052E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，PB1 会输出红外线载波。配置字IR电流决定IR载波的sink电流值。

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NY8A052E发生复位。

当配置字节设置外置晶振（E_HXT，E_XT or E_LXT）用于高速振荡时钟或低速振荡时钟时，PB5 晶振输入引脚（Xin），PB4 晶振输出引脚（Xout）。

当系统振荡选择I_HRC或I_LRC模式，不选择E_HXT、E_XT或E_LXT模式时，如果使能配置字，则在PB4 上可以观察到指令时钟。

如果寄存器T0MD T0CS为 1 和LCK_TM0 为 0，PB2 可以当作定时器 0 外部时钟源。如果寄存器T1CS为 1，PB2 可以当作定时器 1 外部时钟源。

此外，PB6 和PB2 可以当作PWM1 输出，PB7 和PB2 可当作蜂鸣器 1 输出。如果寄存器PWM1OEN为 1，PB6/PB2 可当作PWM1 输出。若寄存器BZ1EN（BZ1CR[7]）为 1，PB7/PB2 可当作蜂鸣器 1 输出。

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I/O数据到总线。
PULLDOWN_EN:	开启内部下拉电阻。
PULLUP_ENB:	开启内部上拉电阻。
RD_TYPE :	选择读取脚位或数据锁存器。

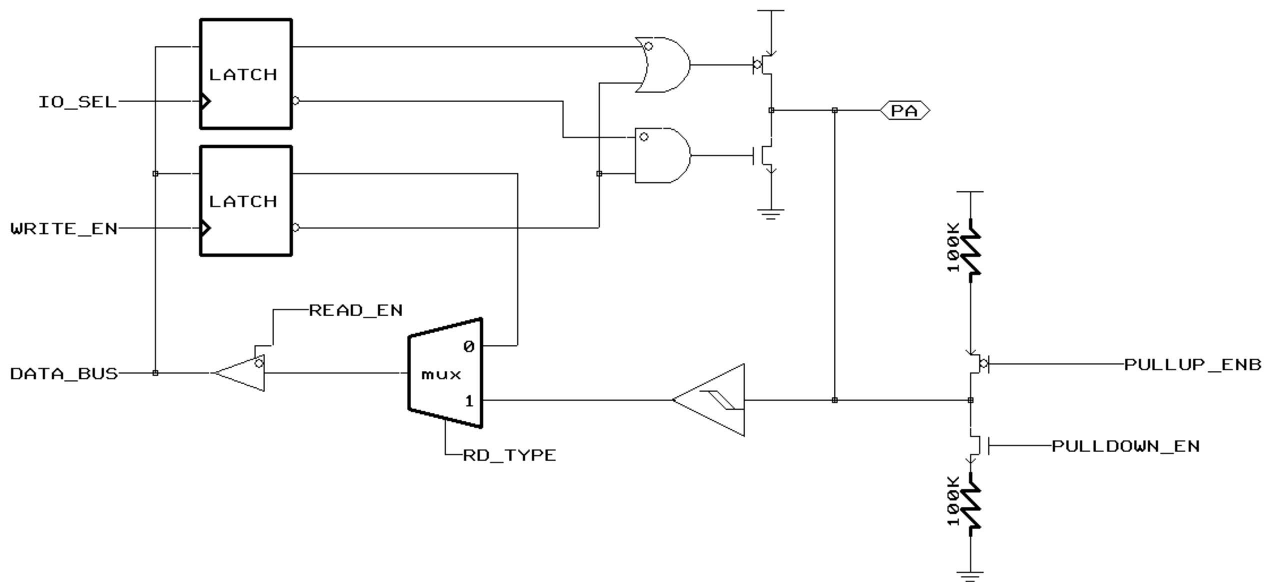


图 5 PA[5:0] 引脚结构框图

OD_EN: 开启开漏极。

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I/O数据到总线。

PULLDOWN_EN: 开启内部下拉电阻。

PULLUP_ENB: 开启内部上拉电阻。

WUBx: 使能PB唤醒功能。

INTEDGE: 外部中断上升沿和下降沿选择。

SET_PBIF: 设置PB唤醒标志。

RD_TYPE: 选择读取脚位或数据锁存器。

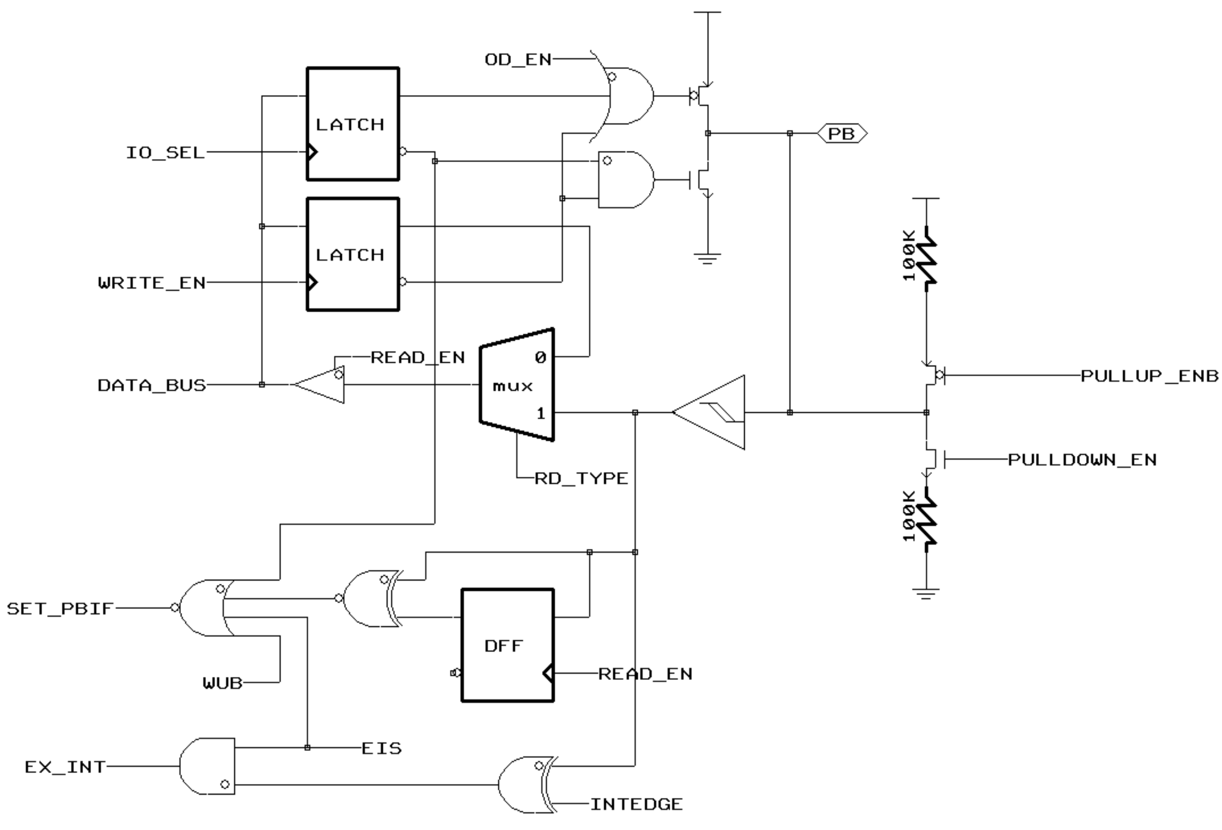


图 6 PB0 引脚结构框图

OD_EN:	开启开漏极。
IREN:	红外载波输出使能。
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I/O数据到总线。
PULLDOWN_EN:	开启内部下拉电阻。
PULLUP_ENB:	开启内部上拉电阻。
WUBx:	使能PB唤醒功能。
RD_TYPE:	选择读取脚位或数据锁存器。
SET_PBIF:	设置PB唤醒标志。

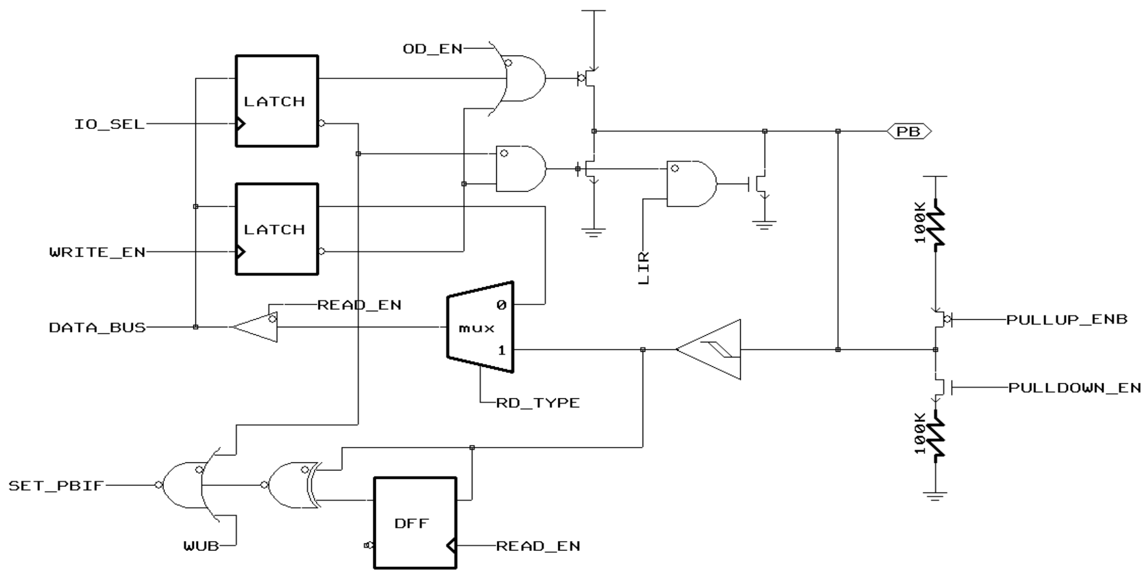


图 7　PB1 引脚结构框图

OD_EN: 开启开漏极。

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I/O数据到总线。

PULLDOWN_EN: 开启内部下拉电阻。

PULLUP_ENB: 开启内部上拉电阻。

WUBx: 使能PB唤醒功能。

SET_PBIF: 设置PB唤醒标志。

RD_TYPE: 选择读取脚位或数据锁存器。

EX_CKI: 定时器 0 外部输入引脚。

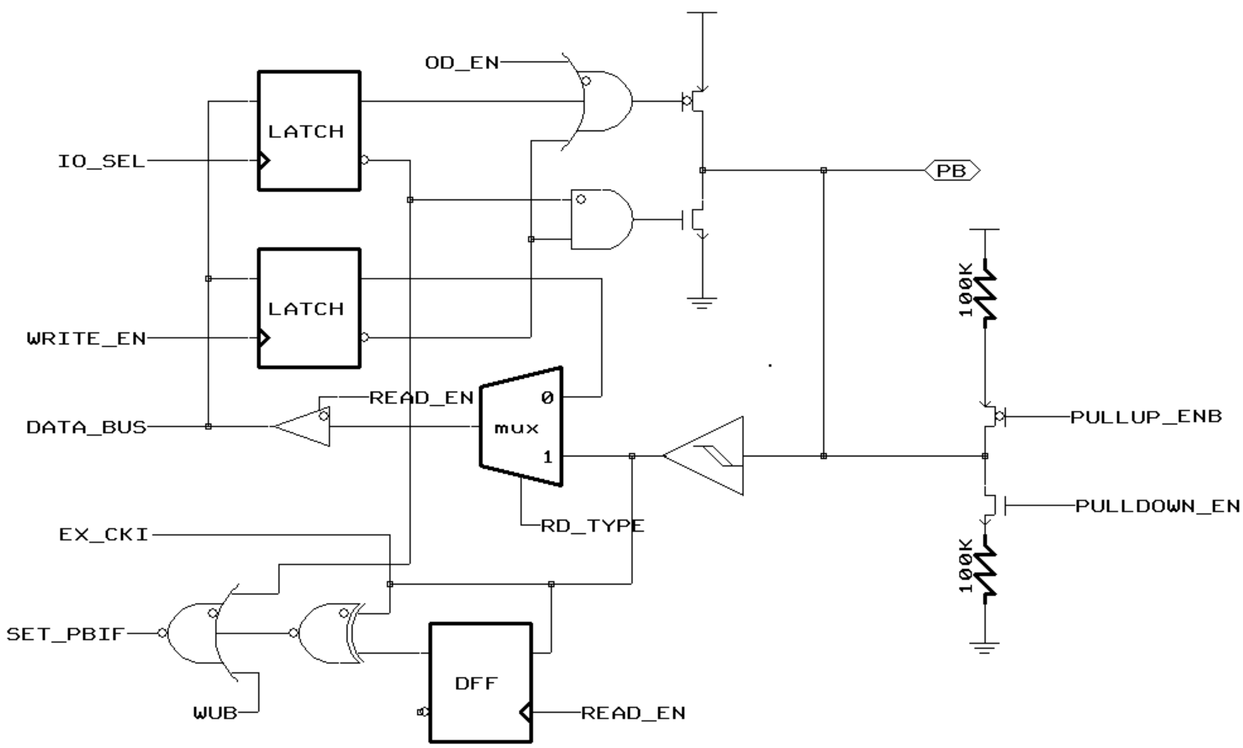


图 8 PB2 引脚结构框图

PB3_RSTPAD:	通过配置位将PB3 设置为复位引脚。
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I/O数据到总线。
PULLDOWN_EN:	开启内部下拉电阻。
PULLUP_ENB:	开启内部上拉电阻。
WUBx:	使能PB唤醒功能。
SET_PBIF:	设置PB唤醒标志。
RD_TYPE:	选择读取脚位或数据锁存器。
RSTB_IN:	复位引脚输入。

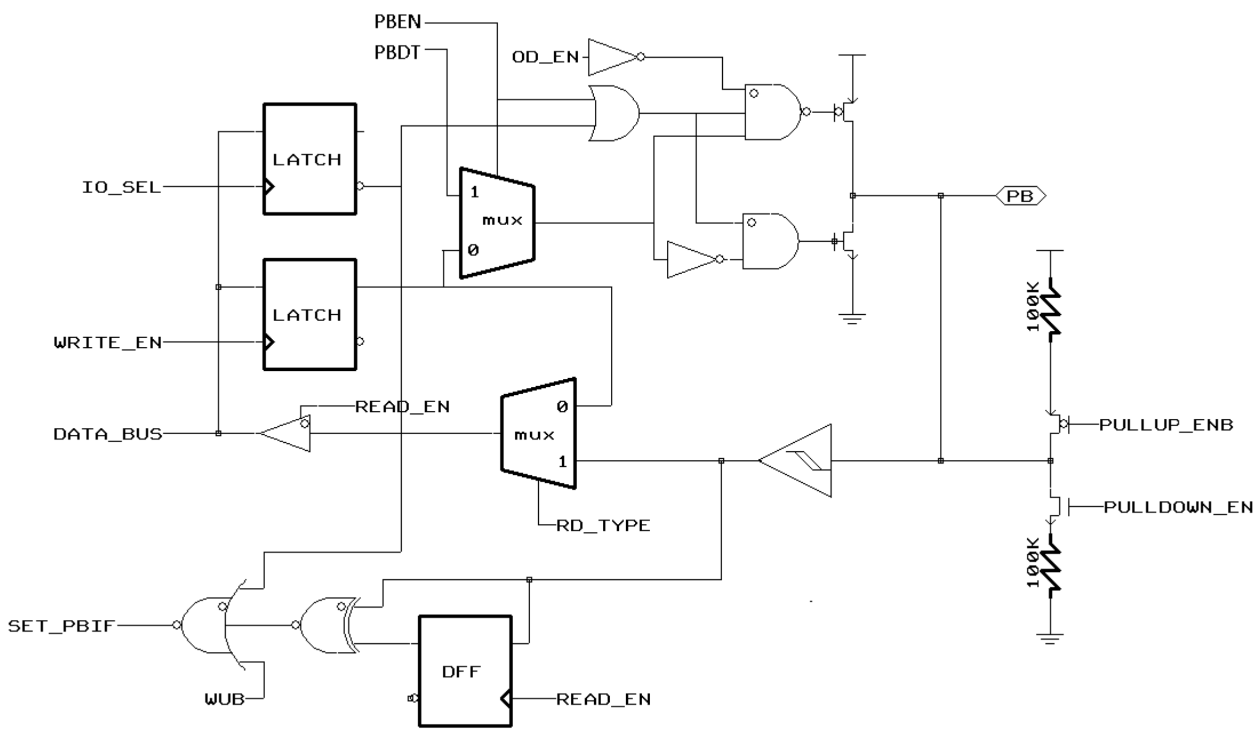


图 9 PB3 引脚结构框图

OD_EN: 开启开漏极。

XTL_EN: 通过配置位将PB4，PB5 设置为晶振引脚。

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I/O数据到总线。

PULLUP_ENB: 开启内部上拉电阻。

WUBx: 使能PB唤醒功能。

RD_TYPE: 选择读取脚位或数据锁存器。

SET_PBIF: 设置PB唤醒标志。

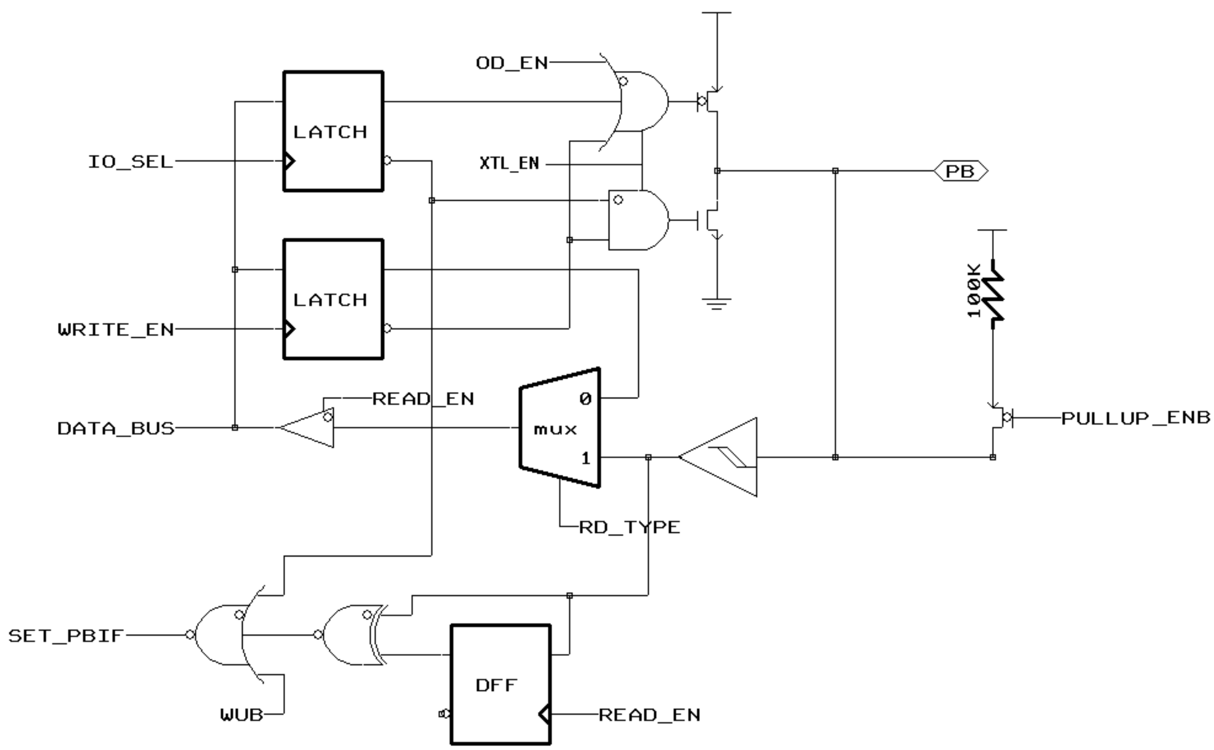


图 10 PB4，PB5 引脚结构框图

OD_EN:	开启开漏极。
DATA_BUS:	PWM或Buzzer数据
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I/O数据到总线。
PULLUP_ENB:	开启内部上拉电阻。
WUB:	使能PB唤醒功能。
RD_TYPE:	选择读取脚位或数据锁存器。
SET_PBIF:	设置PB唤醒标志。

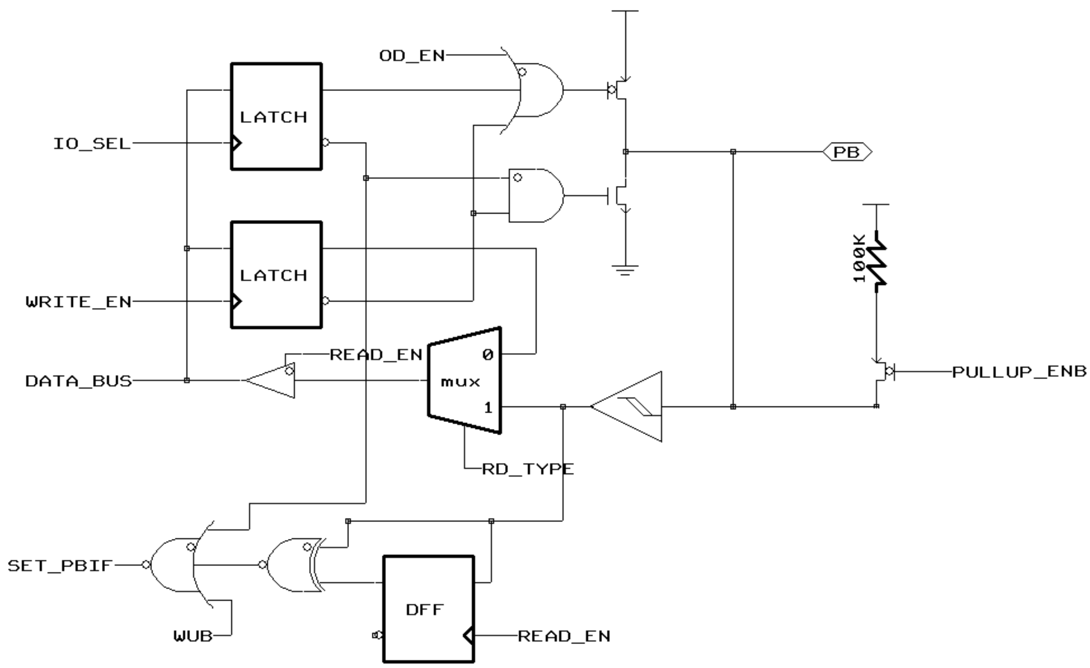


图 11 PB7，PB6 引脚结构框图

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_CKI或低频振荡I_LRC/E_LXT中择一。当T0CS为 0，指令时钟会被选择当作定时器 0 时钟源。当T0CS为 1 且LCK_TM0 为 0，EX_CKI会被当作定时器 0 时钟源。当T0CS是 1 且LCK_TM0 为 1，会选择低频振荡I_LRC/E_LXT当作定时器 0 时钟源。汇总成表格如下。（也请参考表 12）

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_CKI	1	0	X	X
		X	0	
E_LXT	1	1	1	1
I_LRC	1	1	1	0

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/E_LXT的时钟触发沿选择。当T0CE是 1，EX_CKI或I_LRC/E_LXT的下降沿将让定时器 0 计数加一。当T0CE是 0，EX_CKI或I_LRC/E_LXT的上升沿将让定时器 0 计数加一。当使用I_LRC作为定时器 0 时钟源时，建议使用预分频器 0(见下文)，并将比率设置为 4 以上，否则可能会出现漏计数。

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。

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

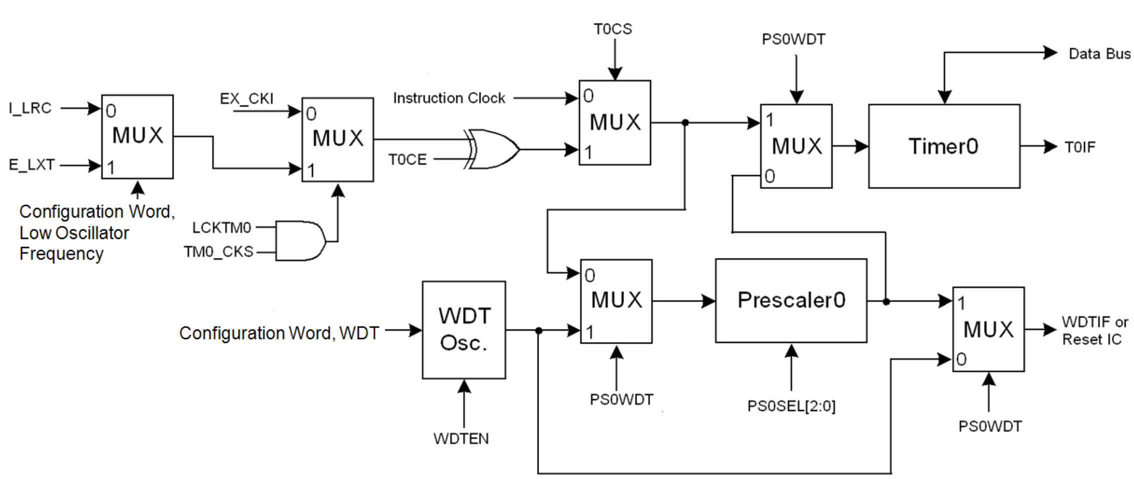


图 12　定时器 0 与WDT结构框图

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。对定时器 1 的写入将同时写入定时器 1 重新加载寄存器(T1rld)和定时器 1 计数器。对定时器 1 的读取将显示定时器 1 当前计数值的内容。

定时器 1 的结构框图如下图所示：

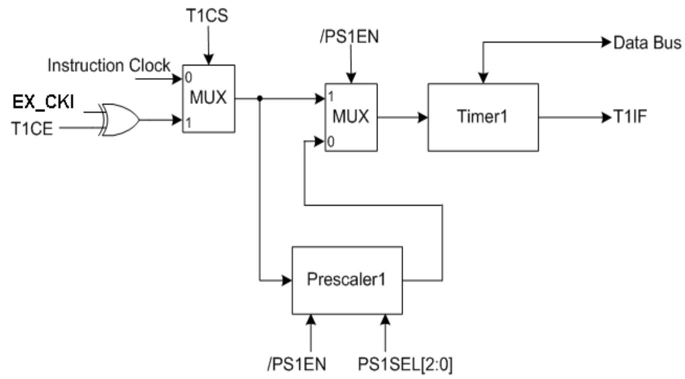


图 13　定时器 1 结构框图

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 1，EX_CKI会被选择当做时钟源。当T1CS为 0，指令时钟会被选择当做时钟源。当EX_CKI被选择，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的下降沿将让定时器 1 计数减一。当T1CE是 0，EX_CKI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 的初始值下数到 0x00，例如，当下溢发生时。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 从 0xFF开始继续下数。

当定时器 1 下溢时，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 的时序图如下图所示。

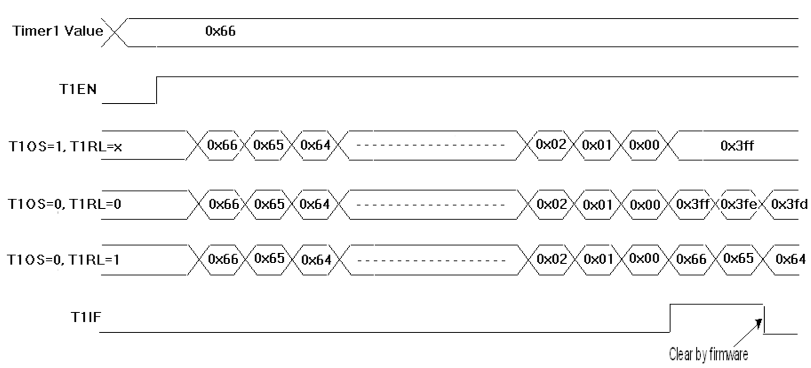


图 14　定时器 1 时序图

当寄存器PWM1OEN（T1CR1[7]）设定为 1，PB2/PB6 为PWM1 输出。当PWM1OEN为 1，PB2/PB6 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]。当用户在写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 下溢后，PWM1DUTY就会更新。

PWM1 的结构框图如下：

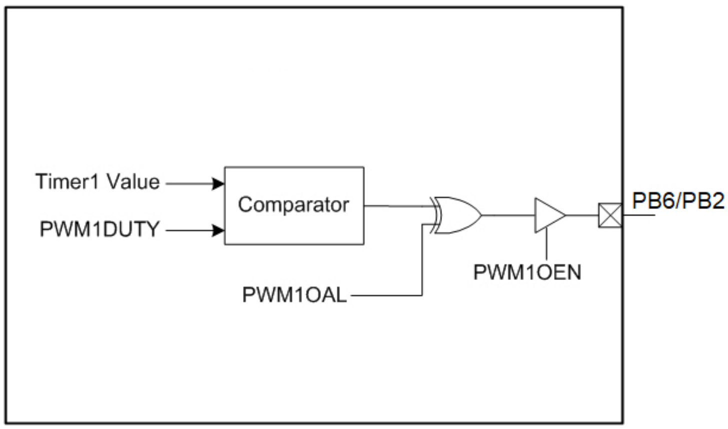


图 15 PWM1 结构框图

当寄存器BZ1EN（BZ1CR1[7]）设定成 1 且使能配置字节时，PB2/PB7 为蜂鸣器 1 输出。当BZ1EN (BZ1CR1[7]) 设定为 1，PB2/PB7 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：

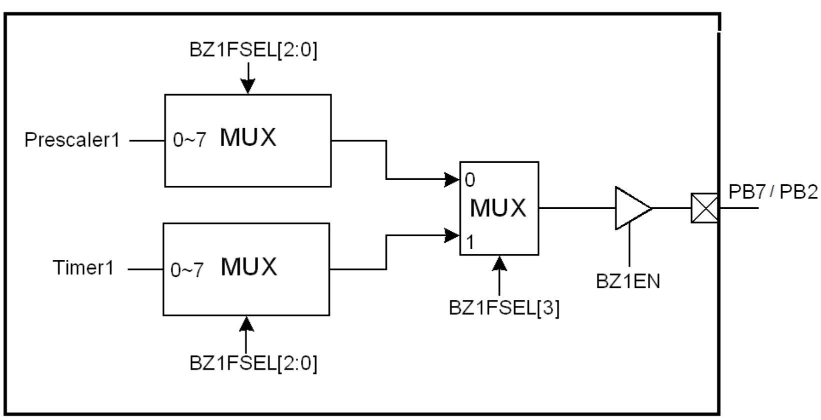


图 16 蜂鸣器 1 结构框图

3.8 PWM2/3/4

当寄存器PWMxOEN（x=2,3,4）设定为 1，PortA或PortB为PWM2/3/4 输出。而且，PortA或PortB会自动成为输出脚。PWM2/3/4 输出的有效状态是由寄存器PWMxOAL决定。当PWMxOAL为 1，PWM2/3/4 为低电平有效输出；PWMxOAL为 0，PWM2/3/4 为高电平有效输出。

PWM2/3/4 的占空比与帧率皆可编程的。占空比是由寄存器PWMxDUTY决定。当PWMxDUTY为 0，PWM2/3/4 无法输出占空比。当PWMxDUTY为 0x3FF，PWM2/3/4 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，PWMxDUTY数值必须小于或等于TMR1[9:0]。另外，由于PWM2/3/4 都与Timer1 共享，用户需要设置P2CR1/P3CR1/P4CR1。用户在写PWMxDUTY时，先写入PWMxDUTY[9:8]高 2 位，再写入PWMxDUTY[7:0]，等到定时器 1 下溢后，PWMxDUTY就会更新。PWM2/3/4 的框图如下图所示。

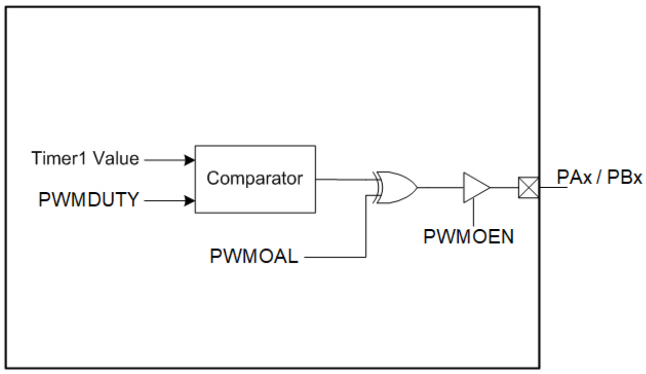


图 17 PWM2/3/4 结构框图

3.9 红外载波

寄存器IREN（IRCR[0]）被设定为 1 后，PB1 为红外线载波输出，而且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。由于红外载波频率来源于高频系统振荡F_{HOSC}，所以在使用外部晶振时，有必要指定系统振荡的频率。寄存器位IROSC358M (IRCR[7])用于向NY8A052E提供此信息。当IROSC358M为 1 时，外部晶振频率为 3.58MHz，当IROSC358M为 0 时，外部晶振频率为 455KHz。当采用内部高频振荡时，该寄存器将被忽略，并为红外模块提供 4MHz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，PB1 将会输出红外线载波。当寄存器IRCSEL（IRCR[2]）为 0 且PB1 输出数据为 1，PB1 将会输出红外线载波。

红外线载波的极性如下图所示：

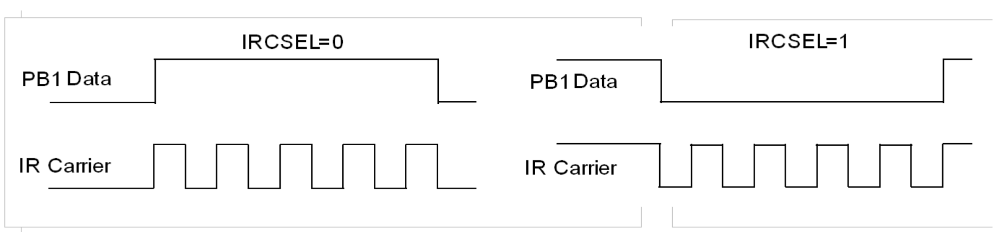


图 18 红外线载波的极性 vs. PB1 数据

3.10 低电压侦测（LVD）

NY8A052E内置准确的低电压侦测电路来侦测VDD电压。将LVDEN设为 1（寄存器PCON[5]）后, 当VDD电压低于下表 LVDS[3:0] 选择的电压值时，LVD输出会变成低电平。如果开启LVD中断使能位，LVD中断标志位将会变成高电平，且GIE=1 时，程序将跳入中断子程序。LVD结构框图如下：

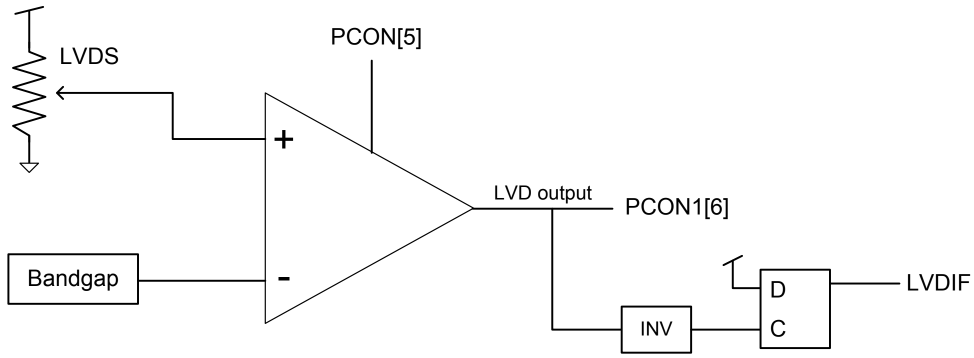


图 19　LVD结构框图

LVD电压选择表如下：

LVDS[3:0]	Voltage
0000	--
0001	--
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　LVD电压选择

注意：LVD侦测电压若由低至高时具有迟滞效应，约 0.1V；

以电池充电检测应用为例，因为侦测电压是由低到高变化，所以LVD电压参照表应为：

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
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

低电压侦测操作流程：

第一步　由 LVDS[3:0] 选择 LVD 电压

第二步　设置 CMPCR = 0x0A

第三步　设置 PCOM[5]=1　(开启 LVD)

第四步　从 PCON1[6] 检查 LVD 状态

注意：如果 LVD 电压 LVDS[3:0]被改变，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。

3.11 电压比较器

NY8A052E提供电压比较器和内部参考电压，具有各种模拟比较方式。比较器的正输入源与负输入源和GPIO 口共享。

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：

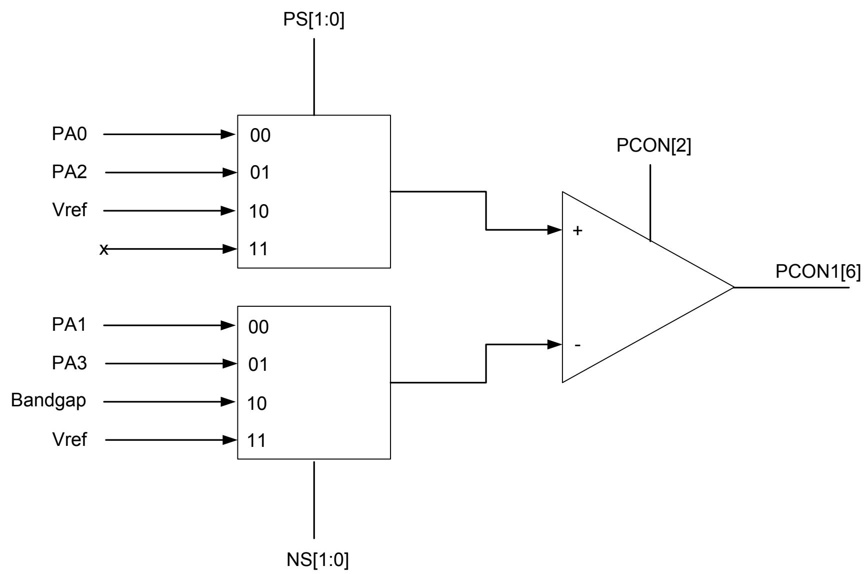


图 20 比较器结构框图

3.11.1 比较器内部参考电压 (Vref)

内部参考电压Vref由串联电阻构成，提供不同电平的参考电压。RBIAS_H和RBIAS_L用于选择Vref的最大值和最小值，LVDS[3:0]用于选择 16 个电压中的一个。

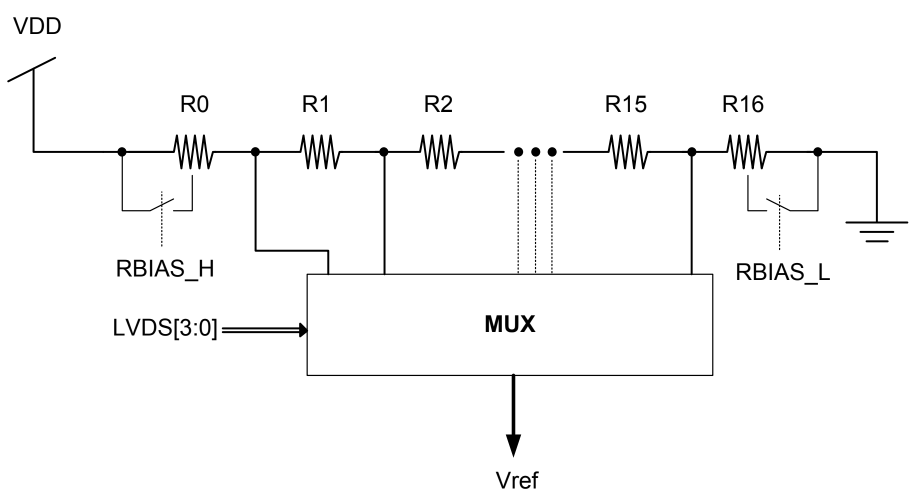


图 21 Vref 结构框图

通过设定RBIAS_H，RBIAS_L 和 LVDS[3:0] 可以选择不同的内部参考电压，LVDS[3:0]用于选择 16 个电压中的一个。设定表如下：

Condition		
LVD		trim
RBIAS[H:L]	LVDS[3:0]	VDD=5V V_{IN-} = $VDD_{\_REF}$ (V)
10 0000b		0.530*VDD =
10 0001b		0.504*VDD =
10 0010b		0.461*VDD =
10 0011b		0.426*VDD =
10 0100b		0.394*VDD =
10 0101b		0.365*VDD =
10 0110b		0.355*VDD =
10 0111b		0.345*VDD =
10 1000b		0.329*VDD =
10 1001b		0.311*VDD =
10 1010b		0.301*VDD =
10 1011b		0.287*VDD =
10 1100b		0.275*VDD =
10 1101b		0.266*VDD =
10 1110b		0.259*VDD =
10 1111b		0.251*VDD =
1 0000b		0.273*VDD =
1 0001b		0.251*VDD =
1 0010b		0.222*VDD =
1 0011b		0.197*VDD =
1 0100b		0.174*VDD =
1 0101b		0.156*VDD =
1 0110b		0.147*VDD =
1 0111b		0.138*VDD =
1 1000b		0.127*VDD =
1 1001b		0.117*VDD =
1 1010b		0.107*VDD =
1 1011b		0.098*VDD =
1 1100b		0.093*VDD =
1 1101b		0.085*VDD =
1 1110b		0.079*VDD =
1 1111b		0.075*VDD =

表 13 内部参考电压选择表

注意：

1. 参考电压偏差 ±0.1V。

2. RBIAS_L和 RBIAS_H必须在睡眠模式前设置为 “00”，以防止漏电。

比较器的正输入源是由 PS[1:0] (寄存器 CMPCR[3:2]). 来决定。表格如下：

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

表 14 正输入源选择

比较器的负输入源是由 NS[1:0] (寄存器 CMPCR[1:0]) 来决定。表格如下：

NS[1:0]	负输入源
00	PA1
01	PA3
10	Bandgap (0.65V)
11	Vref

表 15 负输入源选择

获得比较器输出结果的方法有两种：一种是通过寄存器轮询，另一种是通过探测输出脚。

比较器输出可以通过 LVDOUT(寄存器 PCON1[6])轮询。

要在输出脚探测比较器输出，将 CMPOE(寄存器 OSCCR[6])设置为 1，则 PB1 会输出比较器的实时状态。需要注意的是，当 CMPOE=1 时，如果 PWM3 功能被启用，则该功能将被禁用。

3.12 看门狗定时器（WDT）

NY8A052E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，其工作仍然可以通过寄存器WDTEN（PCON[7]）位来控制。此外，WDT上溢后的机制可以复位NY8A052E或发出由另一个配置字确定的中断请求。同时，WDT上溢后，寄存器位/TO (STATUS[4])清零。

WDT上溢的时基由两个配置字节决定，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]决定并取决于WDT的上溢机制。如果WDT上溢将复位NY8A052E，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。如果WDT上溢会使NY8A052E复位，分频速率从 1:1 到 1:128，如果WDT上溢会使NY8A052E中断，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。

3.13 中断

NY8A052E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。硬件中断则有以下六种：

- 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。

GIE是总中断使能位，必须为 1 才能开启硬件中断功能。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NY8A052E自动清除为零，这样做是为了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，相应的中断标志位寄存器INTF将被设置为 1。该位在程序将 0 写入之后才会被清除为零。因此，用户可以通过轮询寄存器INTF来获取导致硬件中断的事件信息。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寄存器INTF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，NY8A052E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NY8A052E。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。

需要注意的是，ENI指令不能放在RETIE指令之前，因为中断服务程序中的ENI指令会触发嵌套中断，而RETIE会在跳出ISR后清除内部中断处理，因此有可能出现中断标志被错误清除的情况。

3.13.1 Timer0 上溢中断

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

3.13.2 Timer1 下溢中断

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

3.13.3 看门狗超时中断

当WDT超时且配置字节选择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。

3.13.4 PB 输入状态改变中断

当PBx（0 ≤ x ≤ 7）设置置为输入引脚，并将相应的寄存器位WUPBx设置为 1 时，所选I/O引脚的状态变化将设置寄存器PBIF位。如果PBIE和GIE被设置为 1，这个中断请求将被处理。注意当PB0 同时设为状态变化中断和外部中断时，外部中断使能EIS=1 将关闭PB0 状态变化操作。

3.13.5 外部中断输入

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。

3.13.6 LVD 中断

当VDD电平低于LVD电压时，LVD标志由高变低，并设置寄存器位LVDIF=1。如果LVDIE和GIE设置为 1，这个中断请求将被处理。

3.14 振荡器配置

因为NY8A052E是双时钟IC，有高速振荡时钟（F_{HOSC}）和低速振动时钟（F_{LOSC}）来选择作为系统振荡时钟（F_{OSC}）。可用作F_{HOSC}的振荡器有内部高速RC振荡器（I_HRC）、外部高速晶体振荡器（E_HXT）和外部晶体振荡器（E_XT）。可用作F_{LOSC}的振荡器有内部低速RC振荡器（I_LRC）和外部低速晶体振荡器（E_LXT）。

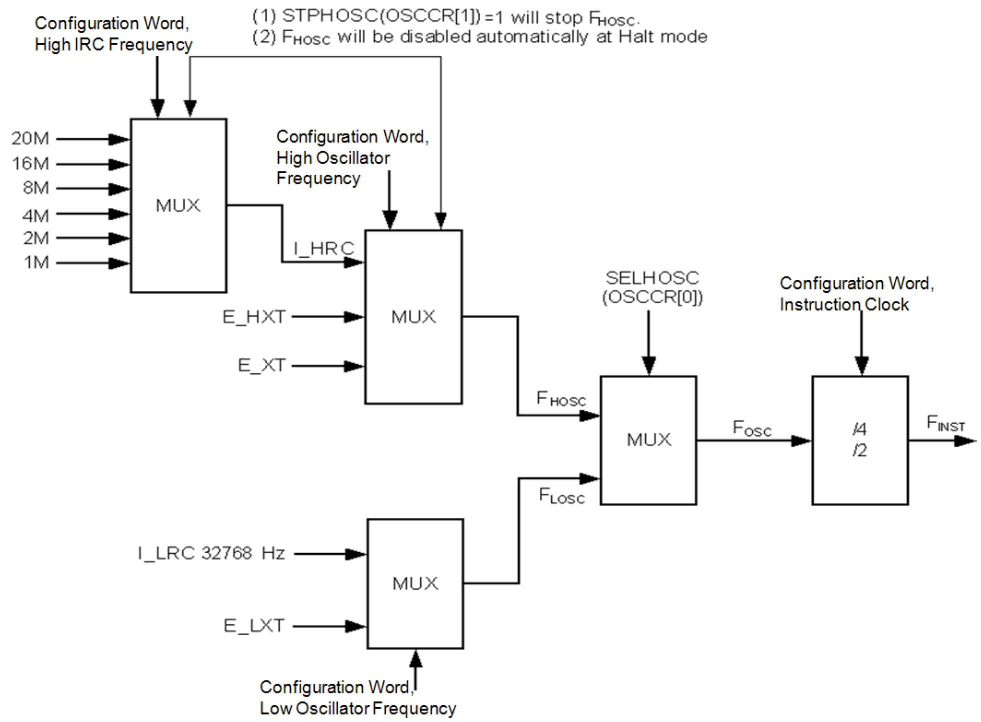


图 22 NY8A052E振荡配置结构图

有两个配置字节来决定将被用作F_{HOSC} 的振荡器。当I_HRC被选作F_{HOSC}时，I_HRC的输出频率由三个配置字节决定，且可以选择的频率为 1M、2M、4M、8M、16M或 20MHz。此外，外部晶振脚PB4 和PB5 可用作I/O脚。另一方面，PB4 也可以通过一个配置字节设置来输出指令时钟。如果F_{HOSC}需要使用频率为 8MHz ~ 20MHz的外部晶振，建议使用E_HXT。如果F_{HOSC}需要使用频率为 455KHz ~ 6MHz的外部晶振，建议使用E_XT。当使用E_HXT或E_XT时，PB4/PB5 不能作为一般I/O脚。它们必须用作晶振输出脚和输入脚。PB4 为晶振输出脚(Xout)，PB5为晶振输入脚(Xin)。

有一个配置字节来决定将被用作F_{LOSC} 的振荡器。当选择I_LRC时，其频率约为 32768Hz。如果F_{LOSC}需要外部晶振，选择E_LXT，而且只能是 32768Hz晶体振荡器。当选择E_LXT时，PB4/PB5 不能当作一般I/O脚。它们必须用作晶振输出脚和输入脚。PB4 为晶振输出脚(Xout)，PB5 为晶振输入脚(Xin)。

F_{HOSC} 和 F_{LOSC}双时钟组合如下：

No.	FHOSC	FLOSC
1	I_HRC	I_LRC
2	E_HXT or E_XT	I_LRC
3	I_HRC	E_LXT

表 16 双时钟组合

当E_HXT、E_XT或E_LXT作为振荡之一时，将晶体或谐振器连接到Xin和Xout上提供振荡。此外，为了提供可靠的振荡，建议如下图连接一个电阻和两个电容，参考晶体或谐振器的规格，采用合适的C1 或C2 值。C1 和C2 的推荐值如下表所示。

振荡模式	振荡频率 (Hz)	C1, C2 (pF)

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

表 17 不同外部晶体振荡器频率所推荐的C1 和C2 电容器数值

对于 2 个时钟 CPU 周期模式下的 20MHZ 谐振器，必须使用 18pF C2 电容。

对于精度 32.768k 的晶体，建议设置 C1=C2=20pF 电容。

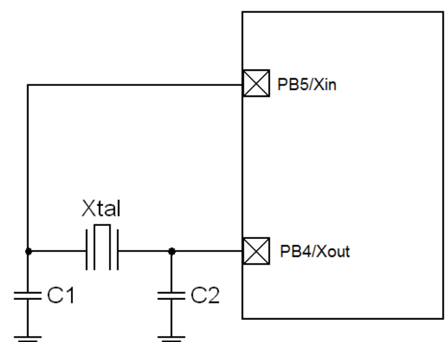


图 23 外部晶体振荡器的硬件连接图

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

3.15 工作模式

NY8A052E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，NY8A052E 将停止几乎所有的运作，除了定时器 0、定时器 1、看门狗用来定期唤醒。在睡眠模式下，NY8A052E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。

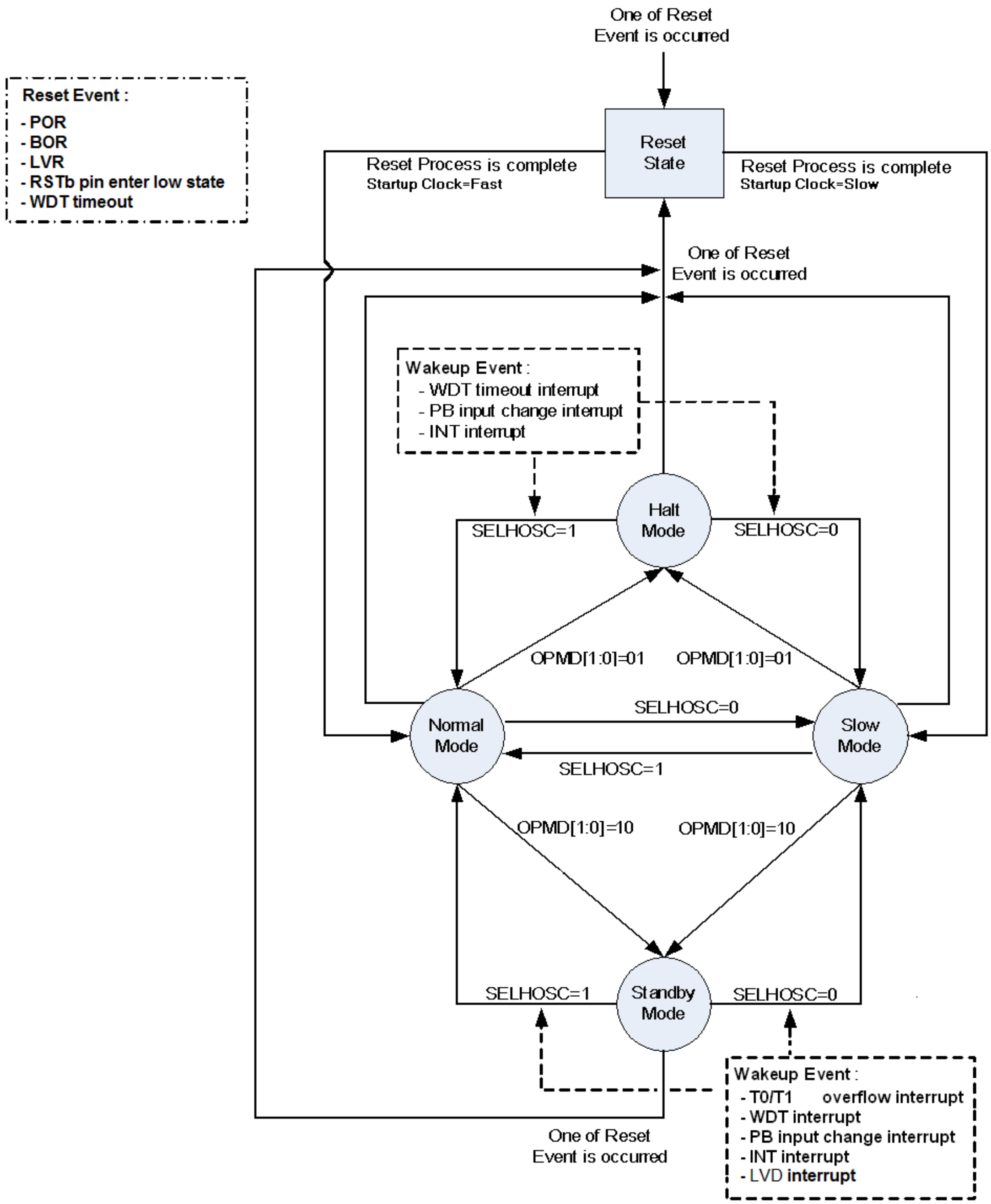

Reset Event :
- POR
- BOR
- LVR
- RSTb pin enter low state
- WDT timeout
One of Reset Event is occurred
Reset State
Reset Process is complete
Startup Clock=Fast
Reset Process is complete
Startup Clock=Slow
One of Reset Event is occurred
Wakeup Event :
- WDT timeout interrupt
- PB input change interrupt
- INT interrupt
Halt Mode
SELHOSC=1
SELHOSC=0
OPMD[1:0]=01
OPMD[1:0]=01
Normal Mode
Slow Mode
SELHOSC=0
SELHOSC=1
OPMD[1:0]=10
OPMD[1:0]=10
Standby Mode
SELHOSC=1
SELHOSC=0
One of Reset Event is occurred
Wakeup Event :
- T0/T1 overflow interrupt
- WDT interrupt
- PB input change interrupt
- INT interrupt
- LVD interrupt

图 24 四种工作模式

3.15.1 正常模式

发生任何复位事件并且复位过程完成后，NY8A052E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。如果启动时钟为快速，NY8A052E将进入正常模式，如果启动时钟为慢速，NY8A052E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。在上电或任何复位触发器被释放后，待复位程序完成NY8A052E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，NY8A052E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。

3.15.2 慢速模式

通过写 0 至寄存器SELHOSC位，NY8A052E将进入慢速模式。在慢速模式下，为节省功耗，F_{LOSC}被选为系统振荡时钟。然而，F_{HOSC}将不会自动被NY8A052E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。

3.15.3 待机模式

通过写入 10b至寄存器OPMD[1:0]，NY8A052E将进入待机模式。然而，在待机模式下，F_{HOSC}不会自动被NY8A052E关闭，用户必须进入先慢速模式后写入 1 至寄存器STPHOSC位，以停止F_{HOSC}。部分NY8A052E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NY8A052E会被唤醒。结束时间由寄存器TMR0/TMR1、F_{INST}和Timer0/Timer1 的其他配置决定。

- 停止执行指令且一些硬件功能可以根据相应的硬件使能位来开启/关闭。
- 由写入 1 至寄存器STPHOSC位F_{HOSC}可以被关闭。
- F_{LOSC}仍保持运作。
- 如遇以下任一状况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）发生INT外部中断
  （e）LVD中断

- 在从待机模式唤醒后，如果SELHOSC=1，IC将回到正常模式；如果SELHOSC=0，则IC将回到慢速模式。
- 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。

3.15.4 睡眠模式

通过执行SLEEP指令或写入 01b至寄存器OPMD[1:0]位，NY8A052E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NY8A052E只能通过一些特殊事件唤醒。因此，睡眠模式是NY8A052E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）PB输入状态改变中断

  （c）INT或外部中断。
- 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：用户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），并在同一时间进入睡眠模式。

3.15.5 唤醒稳定时间

睡眠模式的唤醒稳定时间由配置字决定：高速振荡频率或低速振荡频率。如果选择E_HXT、E_XT和E_LXT中的一个，则唤醒周期为 512*F_{OSC}。如果不选择XT模式，唤醒时间设置为 16*Fosc。另一方面，待机模式不需要唤醒稳定时间，因为F_{HOSC}或F_{LOSC}仍在待机模式下运行。

在NY8A052E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，NY8A052E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指令或将寄存器GIE位写 0，则在唤醒后执行下一条指令。

3.15.6 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
Timer0/1	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
其它模块	模块使能位	模块使能位	模块使能位	全部关闭

模式	正常模式	慢速模式	待机模式	睡眠模式
唤醒源	-	-	- Timer0 上溢 - Timer1 下溢 - WDT超时 - PB输入状态改变 -外部中断 - LVD中断	- WDT超时 - PB输入状态改变 - 外部中断

表 18　工作模式概述

3.16 复位

当以下任一复位事件发生时，NY8A052E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。
- 看门狗超时复位.

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 19　/TO和/PD值和相关事件概述

复位事件发生后，NY8A052E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，它由两位配置字节决定，这个时间可能是 140us，4.5ms，18ms，72ms或 288 ms。振荡器稳定后，NY8A052E将等待 Fosc 的 16 个时钟周期（OST，振荡器启动时间）后完成复位。

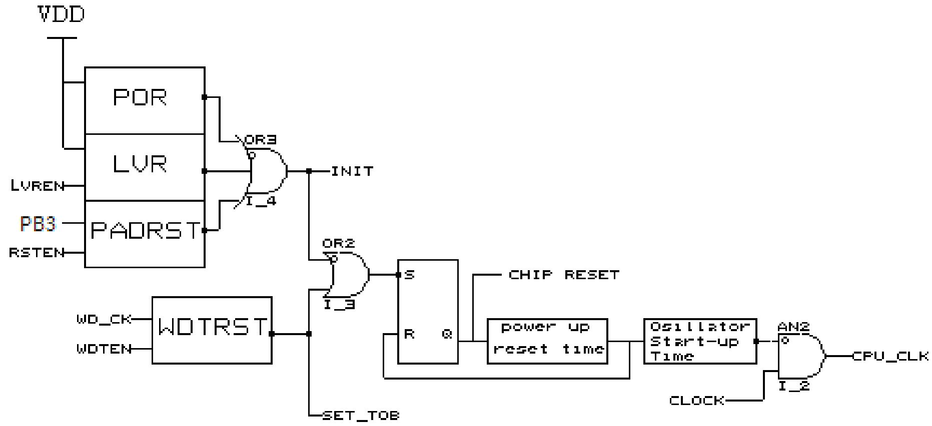


图 25 芯片复位电路框图

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。

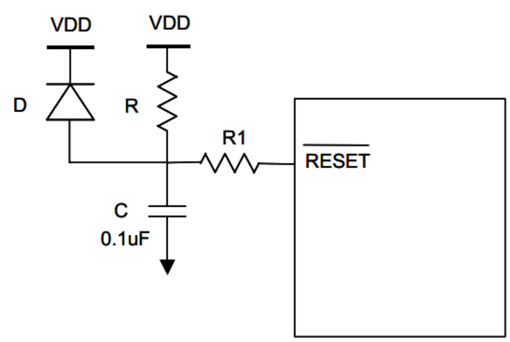


图 26 外部上电复位硬件连接图

4. 指令集

NY8A052E为各种应用程序提供了 55 个强大的指令。

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

表格 20 指令集

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 ~ 0x6F。

S：S页面特殊功能寄存器，S值为 0x0 ~ 0x18。

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 ≤ R ≤ 63 d = 0, 1.
操作	R + ACC + C → 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 指令执行前： ACC=0x12, R=0x34, C=1, d=1, 指令执行后： R=0x47, ACC=0x12, C=0.

ADDAR	Add ACC and R
语法	ADDAR R, d
操作数	0 ≤ R ≤63 d = 0, 1.
操作	ACC + R → 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 指令执行前： ACC=0x12, R=0x34,C=1, d=1, 指令执行后： R=0x46, ACC=0x12, C=0.

ADCIA	Add ACC and Immediate with Carry
语法	ADCIA i
操作数	0 ≤ i < 255
操作	ACC + i + C → 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 指令执行前： ACC=0x12, i=0x34, C=1, 指令执行后： ACC=0x47, C=0.

ADDIA	Add ACC and Immediate
语法	ADDIA i
操作数	0 ≤ i < 255
操作	ACC + i → 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 指令执行前： ACC=0x12, i=0x34, C=1, 指令执行后： ACC=0x46, C=0.

ANDAR	AND ACC and R
语法	ANDAR R, d
操作数	0 ≤ R ≤ 63. d = 0, 1.
操作	ACC & R → dest
状态影响	Z
说明	ACC和R做“AND”运算，若d=“0”，结果存入ACC；若d=“1”，结果存入“R”。
周期	1
举例	ANDAR R, d 指令执行前： ACC=0x5A, R=0xAF, d=1. 指令执行后： R=0x0A, ACC=0x5A, Z=0.

BCR	Clear Bit in R
语法	BCR R, bit
操作数	0 ≤ R ≤ 63 0 ≤ bit ≤ 7
操作	0 → R[bit]
状态影响	--
说明	将R寄存器的bit位（0~7）清 0。
周期	1
举例	BCR R,B2 指令执行前： R=0x5A, B2=0x3, 指令执行后： R=0x52.

ANDIA	AND Immediate with ACC
语法	ANDIA i
操作数	0 ≤ i < 255
操作	ACC & i → ACC
状态影响	Z
说明	ACC和 8 位立即数做“AND”运算。结果存入ACC。
周期	1
举例	ANDIA i 指令执行前： ACC=0x5A, i=0xAF, 指令执行后： ACC=0x0A, Z=0.

BSR	Set Bit in R
语法	BSR R, bit
操作数	0 ≤ R ≤ 63 0 ≤ bit ≤ 7
操作	1 → R[bit]
状态影响	--
说明	设置R寄存器的bit位为 1。
周期	1
举例	BSR R,B2 指令执行前： R=0x5A, B2=0x2, 指令执行后： R=0x5E.

BTRSC	Test Bit in R and Skip if Clear
语法	BTRSC R, bit
操作数	$0 \leq R \leq 63$ $0 \leq bit \leq 7$
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 指令执行前： R=0x5A, B2=0x2, 指令执行后： 由于R[B2]=0，则指令 1 不执行，程序直接从指令 2 开始执行。

CALL	Call Subroutine
语法	CALL adr
操作数	$0 \leq adr < 255$
操作	PC + 1 → Top of Stack {PCHBUF, adr} → 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 PCHBUF[1:0] 载入 PC[9:8]。
周期	2
举例	CALL SUB 指令执行前： PC=A0. Stack pointer=1 指令执行后： PC=address of SUB, Stack[1] = A0+1, Stack pointer=2.

BTRSS	Test Bit in R and Skip if Set
语法	BTRSS R, bit
操作数	$0 \leq R \leq 63$ $0 \leq bit \leq 7$
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 指令执行前： R=0x5A, B2=0x3, 指令执行后： 由于R[B2]=1，则指令 2 不执行，直接从指令 3 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[1:0]赋值给PC[9:8]，将ACC赋值给PC[7:0]。
周期	2
举例	CALLA 指令执行前： TBHP=0x02, ACC=0x34. PC=A0. Stack pointer=1. 指令执行后： PC=0x234, Stack[1]=A0+1, Stack pointer=2

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 指令执行前： ACC=0x55,　Z=0. 指令执行后： ACC=0x00, Z=1.

CLRWDT	Clear Watch-Dog Timer
语法	CLRWDT
操作数	--
操作	00h→WDT, 00h→WDT prescaler 1→ /TO 1→ /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 指令执行前： /TO=0 指令执行后： /TO=1

CLRR	Clear R
语法	CLRR　R
操作数	0 ≤ R ≤ 63
操作	00h→R 1→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 指令执行前： R=0x55, Z=0. 指令执行后： R=0x00, Z=1.

COMR	Complement R
语法	COMR　R, d
操作数	0 ≤ R ≤ 63 d = 0, 1.
操作	~R→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 指令执行前： R=0xA6, d=1, Z=0. 指令执行后： R=0x59, Z=0.

CMPAR	Compare ACC and R
语法	CMPAR R
操作数	0 ≤ R ≤ 63
操作	R - ACC → (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 指令执行前： R=0x34, ACC=12, Z=0, C=0. 指令执行后： R=0x34, ACC=12, Z=0, C=1.

DECR	Decrease R
语法	DECR R, d
操作数	0 ≤ R ≤ 63 d = 0, 1.
操作	R - 1 → dest
状态影响	Z
说明	R - 1，若d=“0”，结果存入ACC； 若d=“1”，结果存入R。
周期	1
举例	DECR R, d 指令执行前： R=0x01, d=1, Z=0. 指令执行后： R=0x00, Z=1.

DAA	Convert ACC Data Format from Hexadecimal to Decimal
语法	DAA
操作数	--
操作	ACC(hex) → ACC(dec)
状态影响	C
说明	将累加器中的 16 进制数调整为十进制数，该指令必须紧跟在加法指令后。
周期	1
举例	DISI ADDAR R,d DAA ENI 指令执行前： ACC=0x28, R=0x25, d=0. 指令执行后： ACC=0x53, C=0.

DECRSZ	Decrease R, Skip if 0
语法	DECRSZ R, d
操作数	0 ≤ R ≤ 63 d = 0, 1.
操作	R - 1 → dest, Skip if result = 0
状态影响	--
说明	R 先- 1，若d=“0”，结果存入ACC；若d=“1”，结果存入R，若结果为“0”则跳过下一条指令，改为执行NOP指令，因此结果为“0”时要执行两个周期。
周期	1 or 2(skip)
举例	DECRSZ R, d 指令 2 instruction3 指令执行前： R=0x1, d=1, Z=0. 指令执行后： R=0x0，Z=1，操作结果为 0，指令 2 被跳过。

DISI	Disable Interrupt Globally
语法	DISI
操作数	--
操作	Disable Interrupt, 0 → GIE
状态影响	--
说明	GIE设置为 0, 关闭总中断。
周期	1
举例	DISI 指令执行前： GIE=1. 指令执行后： GIE=0.

GOTO	Unconditional Branch
语法	GOTO adr
操作数	0 ≤ adr < 511
操作	{PCHBUF, adr} → PC
状态影响	--
说明	无条件短跳转指令，9 位地址写入PC[8:0]，PCHBUF[1]写入PC[9]。
周期	2
举例	GOTO Level 指令执行前： PC=A0. 指令执行后： PC=address of Level.

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 指令执行前： GIE=0. 指令执行后： GIE=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[1:0]值写入PC[9:8]。
周期	2
举例	GOTOA 指令执行前： PC=A0. TBHP=0x02, ACC=0x34. 指令执行后： PC=0x234.

INCR	Increase R
语法	INCR R, d
操作数	0 ≤ R ≤ 63 d = 0, 1.
操作	R + 1 → 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 指令执行前： R=0xFF, d=1, Z=0. 指令执行后： R=0x00, Z=1.

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0x001的地址装入PC[9:0]。
周期	3
举例	INT 指令执行前： PC=address of INT code 指令执行后： PC=0x01

INCRSZ	Increase R, Skip if 0
语法	INCRSZ R, d
操作数	0 ≤ R ≤ 63 d = 0, 1.
操作	R + 1 → 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, instruction3. 指令执行前： R=0xFF, d=1, Z=0. 指令执行后： R=0x00，Z=1，因结果为 0，程序跳过指令 2。

IORAR	OR ACC with R
语法	IORAR R, d
操作数	0 ≤ R ≤ 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 指令执行前： R=0x50, ACC=0xAA, d=1, Z=0. 指令执行后： R=0xFA, ACC=0xAA, Z=0.

IORIA	OR Immediate with ACC
语法	IORIA i
操作数	0 ≦ 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 指令执行前： i=0x50, ACC=0xAA, Z=0. 指令执行后： ACC=0xFA, Z=0.

IOSTR	Move F-page SFR to ACC
语法	IOSTR F
操作数	5 ≦ F ≦ 15
操作	F-page SFR → 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 指令执行前： F=0x55, ACC=0xAA. 指令执行后： F=0x55, ACC=0x55.

IOST	Load F-page SFR from ACC
语法	IOST F
操作数	0 ≦ F ≦ 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 指令执行前： F=0x55, ACC=0xAA. 指令执行后： F=0xAA, ACC=0xAA.

LCALL	Call Subroutine
语法	LCALL adr
操作数	0 ≦ adr ≦ 1023
操作	PC + 1 → Top of Stack, adr → PC[9:0]
状态影响	--
说明	长调用子程序。首先将PC+1 压入栈顶， 然后将 10 位立即数载入PC[9:0]。
周期	2
举例	LCALL SUB 指令执行前： PC=A0. Stack level=1 指令执行后： PC=address of SUB, Stack[1]= A0+1, Stack pointer =2.

LGOTO	Unconditional Branch
语法	LGOTO adr
操作数	0 ≤ adr ≤ 1023
操作	adr→PC[9:0].
状态影响	--
说明	无条件长跳转，10 位立即数写入PC[9:0]
周期	2
举例	LGOTO Level 指令执行前： PC=A0. 指令执行后： PC=address of Level.

MOVIA	Move Immediate to ACC
语法	MOVIA i
操作数	0 ≤ i < 255
操作	i→ACC
状态影响	--
说明	8 位立即数赋值给ACC。
周期	1
举例	MOVIA i 指令执行前： i=0x55, ACC=0xAA. 指令执行后： ACC=0x55.

MOVAR	Move ACC to R
语法	MOVAR R
操作数	0 ≤ R ≤ 63
操作	ACC→R
状态影响	--
说明	ACC赋值给R-page寄存器。
周期	1
举例	MOVAR R 指令执行前： R=0x55, ACC=0xAA. 指令执行后： R=0xAA, ACC=0xAA.

MOVR	Move R to ACC or R
语法	MOVR R, d
操作数	0 ≤ R ≤ 63 d = 0, 1.
操作	R→dest
状态影响	Z
说明	R-page寄存器赋值给d，若d=“0”，结果存入ACC；若d=“1”，结果存入寄存器R。指令执行后，通过状态标位Z检查R是否为 0。
周期	1
举例	MOVR R, d 指令执行前： R=0x0, ACC=0xAA, Z=0, d=0. 指令执行后： R=0x0, ACC=0x00, Z=1.

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 指令执行前： PC=A0 指令执行后： PC=A0+1

RETIA	Return with Data in ACC
语法	RETIA i
操作数	0 ≤ i < 255
操作	i→ACC, Top of Stack →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 1。
周期	2
举例	RETIA i 指令执行前： GIE=0, Stack pointer =2. i=0x55, ACC=0xAA. 指令执行后： GIE=1, PC=Stack[2], Stack pointer =1. ACC=0x55.

RETIE	Return from Interrupt and Enable Interrupt Globally
语法	RETIE
操作数	--
操作	Top of Stack→PC 1→GIE
状态影响	--
说明	中断返回，栈顶地址载入PC同时使能中断，GIE标志为 1。
周期	2
举例	RETIE 指令执行前： GIE=0, Stack level=2. 指令执行后： GIE=1, PC=Stack[2], Stack pointer=1.

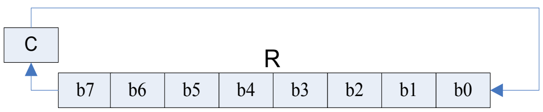
RET	Return from Subroutine
语法	RET
操作数	--
操作	Top of Stack→PC
状态影响	--
说明	子程序返回，栈顶载入PC。
周期	2
举例	RET 指令执行前： Stack level=2. 指令执行后： PC=Stack[2], Stack level=1.

RLR **Rotate Left R Through Carry**

语法 RLR R, d

操作数 $0 \le R \le 63$
d = 0, 1.

操作 C → dest[0], R[7] → C,
R[6:0] → 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

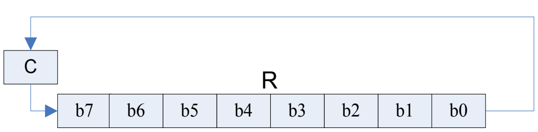
举例 RLR R, d
指令执行前：
R=0xA5, d=1, C=0.
指令执行后：
R=0x4A, C=1.

RRR **Rotate Right R Through Carry**

语法 RRR R, d

操作数 $0 \le R \le 63$
d = 0, 1.

操作 C → dest[7], R[7:1] → dest[6:0],
R[0] → 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
指令执行前：
R=0xA5, d=1, C=0.
指令执行后：
R=0x52, C=1.

SBCAR **Subtract ACC and Carry from R**

语法 SBCAR R, d

操作数 $0 \le R \le 63$
d = 0, 1.

操作 R + (~ACC) + C → 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) 指令执行前：
R=0x05, ACC=0x06, d=1, C=0.
指令执行后：
R=0xFE, C=0. (-2)

(b) 指令执行前：
R=0x05, ACC=0x06, d=1, C=1.
指令执行后：
R=0xFF, C=0. (-1)

(c) 指令执行前：
R=0x06, ACC=0x05, d=1, C=0.
指令执行后：
R=0x00, C=1. (-0), Z=1.

(d) 指令执行前：
R=0x06, ACC=0x05, d=1, C=1.
指令执行后：
R=0x1, C=1. (+1)

SBCIA	Subtract ACC and Carry from Immediate
语法	SBCIA i
操作数	0 ≤ i < 255
操作	i + (~ACC) + C → 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) 指令执行前： i=0x05, ACC=0x06, C=0. 指令执行后： ACC=0xFE, C=0. (-2) (b) 指令执行前： i=0x05, ACC=0x06, C=1. 指令执行后： ACC=0xFF, C=0. (-1) (c) 指令执行前： i=0x06, ACC=0x05, C=0. 指令执行后： ACC=0x00, C=1. (-0), Z=1. (d) 指令执行前： i=0x06, ACC=0x05, C=1. 指令执行后： ACC=0x1, C=1. (+1)

SFUN	Load S-page SFR to ACC
语法	SFUN S
操作数	0 ≤ S ≤15
操作	ACC → 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 指令执行前： S=0x55, ACC=0xAA. 指令执行后： S=0xAA, ACC=0xAA.

SFUNR	Move S-page SFR from ACC
语法	SFUNR S
操作数	0 ≤ S ≤ 15
操作	S-page SFR → ACC
状态影响	--
说明	读S-page特殊寄存器到ACC
周期	1
举例	SFUNR S 指令执行前： S=0x55, ACC=0xAA. 指令执行后： S=0x55, ACC=0x55.

SLEEP	Enter Halt Mode
语法	SLEEP
操作数	--
操作	00h → WDT, 00h → WDT prescaler 1 → /TO 0 → /PD
状态影响	/TO, /PD
说明	WDT和分频器 0 清零。/TO标志为1, /PD清零, IC进入睡眠。
周期	1
举例	SLEEP 指令执行前： /PD=1, /TO=0. 指令执行后： /PD=0, /TO=1.

SUBAR	Subtract ACC from R
语法	SUBAR R, d
操作数	0 ≦ R ≦ 63 d = 0, 1.
操作	R – ACC → 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	SUBAR R, d (a) 指令执行前： R=0x05, ACC=0x06, d=1. 指令执行后： R=0xFF, C=0. (-1) (b) 指令执行前： R=0x06, ACC=0x05, d=1. 指令执行后： R=0x01, C=1. (+1)

SUBIA	Subtract ACC from Immediate
语法	SUBIA i
操作数	0 ≦ i < 255
操作	i – ACC → 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) 指令执行前： i=0x05, ACC=0x06. 指令执行后： ACC=0xFF, C=0. (-1) (b) 指令执行前： i=0x06, ACC=0x05, d=1, . 指令执行后： ACC=0x01, C=1. (+1)

SWAPR	Swap High/Low Nibble in R
语法	SWAPR R, d
操作数	0 ≦ R ≦ 63 d = 0, 1.
操作	R[3:0] → dest[7:4]. R[7:4] → dest[3:0]
状态影响	--
说明	寄存器半字节交换，若d=“0”，结果存入ACC；若d=“1”，结果存入R。
周期	1
举例	SWAPR R, d 指令执行前： R=0xA5, d=1. 指令执行后： R=0x5A.

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

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

XORAR	Exclusive-OR ACC with R
语法	XORAR R, d
操作数	0 ≤ R ≤ 63 d = 0, 1.
操作	ACC ⊕ R → dest
状态影响	Z
说明	ACC和R做“XOR”运算
周期	1
举例	XORAR R, d 指令执行前： R=0xA5, ACC=0xF0, d=1. 指令执行后： R=0x55.

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 指令执行前： T0MD=0x55, ACC=0xAA. 指令执行后： ACC=0x55.

XORIA	Exclusive-OR Immediate with ACC
语法	XORIA i
操作数	0 ≤ i < 255
操作	ACC ⊕ i → 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 指令执行前： i=0xA5, ACC=0xF0. 指令执行后： ACC=0x55.

5. 配置字节表

Item	名称	Options
1	High Oscillator Frequency 高频振荡模式	1. I_HRC 2. E_HXT 3. E_XT
2	Low Oscillator Frequency 低频振荡模式	1. I_LRC 2. E_LXT
3	High IRC Frequency 内部高速 RC 频率	1. 1MHz 2. 2MHz 3. 4MHz 4. 8MHz 5. 16MHz 6. 20MHz
4	High Crystal Oscillator 外部高频振荡器	1. 6MHz＜F_{HOSC}≦8MHz 2. 8MHz＜F_{HOSC}≦10MHz 3. 10MHz＜F_{HOSC}≦12MHz 4. 12MHz＜F_{HOSC}≦16MHz 5. 16MHz＜F_{HOSC}≦20MHz
5	Instruction Clock 指令时钟	1. 2 oscillator period 2 个振荡周期 2. 4 oscillator period 4 个振荡周期
6	WDT 看门狗定时器	1. Watchdog Enable (Software control) 看门狗开启（软件控制） 2. Watchdog Disable (Always disable) 看门狗关闭（永远关闭）
7	WDT Event 看门狗定时器事件	1. Watchdog Reset 看门狗复位 2. Watchdog Interrupt 看门狗中断
8	Timer0 source 定时器 0 时钟源	1. EX_CKI 2. Low Oscillator（低频振荡器）
9	PB.3	1. PB.3 is I/O PB.3 是 I/O 脚 2. PB.3 is reset PB.3 是复位脚
10	PB.4	1. PB.4 is I/O PB.4 是 I/O 脚 2. PB.4 is instruction clock output PB.4 是指令时钟输出
11	Startup Time 上电复位时间	1. 140us 2. 4.5ms 3. 18ms 4. 72ms 5. 288ms
12	WDT Time Base 看门狗定时器时基	1. 3.5ms 2. 15ms 3. 60ms 4. 250ms
13	LVR Setting LVR 开关设定	1. Register Control 寄存器控制 2. LVR Always On LVR 永远开启
14	LVR Voltage 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. 3.6 V
15	VDD Voltage VDD 电压	1. 3.0V 2. 4.5V 3. 5.0V
16	Read Output Data 读取输出口数据	1. I/O Port I/O 口 2. Register 寄存器
17	E_LXT Backup Control E_LXT 强化起振开关设定	1. Auto Off 自动关闭 2. Register Off 寄存器关闭
18	EX_CKI to Inst. Clock EX_CKI 到指令时钟	1. Sync 同步 2. Async 不同步
19	Startup Clock 上电时钟源	1. Fast (I_HRC/E_HXT/E_XT) 高速 2. Slow (I_LRC/E_LXT) 低速
20	PWM1 Output Pin PWM1 输出	1. PB.6 2. PB.2
21	PWM2 Output Pin PWM2 输出	1. PA.1 2. PA.0
22	PWM3 Output Pin PWM3 输出	1. PB.1 2. PA.3

Item	名称	Options	
23	PWM4 Output Pin PWM4 输出	1. PB.0	2. PA.2
24	PWM Resolution PWM 分辨率	1. 10-bit	2. 8-bit
25	Buzzer Output Pin 蜂鸣器输出	1. PB.7	2. PB.2
26	Input Schmitt Trigger 输入施密特触发	1 Enable 开启	2. Disable (1/2 VDD) 关闭
27	Input High Voltage (V_{IH}) 输入高电压 (VIH)	1. 0.7VDD	2. 0.5VDD
28	Input Low Voltage (V_{IL}) 输入低电压 (VIL)	1. 0.3VDD	2. 0.2VDD

表 21　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 ~ V_{DD}+0.3	V
T_{OP}	工作温度	-40 ~ +85	°C
T_{ST}	储存温度	-40 ~ +125	°C

***注意：在极端条件下长时间工作，温度可能会导致可靠性问题。**

6.2 直流电气特性

(所有参考F_{INST}=F_{HOSC}/4，F_{HOSC}=16MHz@I_HRC，WDT 开启，环境温度 T_A=25°C，除其他指定说明外。)

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
			2.7				F_{INST}=16MHz @ E_HXT/2
			2.0				F_{INST}=16MHz @ E_HXT/4
			1.6*				F_{INST}=8MHz @ I_HRC/4 & I_HRC/2
							F_{INST}=8MHz @ E_HXT/4 & E_HXT/2
			1.6*				F_{INST}=4MHz @ I_HRC/4 & I_HRC/2
							F_{INST}=4MHz @ E_XT/4 & E_XT/2
			1.6*				F_{INST}=32KHz @ I_LRC/4 & I_LRC/2
							F_{INST}=32KHz @ E_LXT/4 & E_LXT/2
V_{IH}	输入高电平	5V	4.0	--	--	V	RSTb (0.8V_{DD})
		3V	2.4	--	--		
		5V	3.5	--	--	V	所有I/O引脚，EX_CKI, INT CMOS 选项 (0.7V_{DD})
		3V	2.1	--	--		
		5V	2.5	--	--	V	所有I/O引脚，EX_CKI TTL 选项 (0.5V_{DD})
		3V	1.5	--	--		
V_{IL}	输入低电平	5V	--	--	1.0	V	RSTb (0.2V_{DD})
		3V	--	--	0.6		
		5V	--	--	1.5	V	所有I/O引脚，EX_CKI, INT CMOS 选项 (0.3V_{DD})
		3V	--	--	0.9		
		5V	--	--	1.0	V	所有I/O引脚，EX_CKI TTL 选项 (0.2V_{DD})
		3V	--	--	0.6		
I_{OH}	输出高推电流 （小电流）	5V	--	-20	--	mA	V_{OH}=4.0V
		3V	--	-10	--		V_{OH}=2.0V
I_{OL}	输出低灌电流 （大电流）	5V	--	40	--	mA	V_{OL}=1.0V
		3V	--	26	--		
I_{IR}	IR 灌电流	5V	--	40	--	mA	IR
		3V	--	26	--		

符号	参数	V_{DD}	最小值	典型值	最大值	单位	条件
		正常模式					
I_{OP}	工作电流	5V	--	2.7	--	mA	F_{HOSC}=20MHz @ I_HRC/2 & E_HXT/2
		3V	--	1.3	--		
		5V	--	2.1	--	mA	F_{HOSC}=20MHz @ I_HRC/4 & E_HXT/4
		3V	--	1.0	--		
		5V	--	2.4	--	mA	F_{HOSC}=16MHz @ I_HRC/2 & E_HXT/2
		3V	--	1.1	--		
		5V	--	1.9	--	mA	F_{HOSC}=16MHz @ I_HRC/4 & E_HXT/4
		3V	--	0.8	--		
		5V	--	1.5	--	mA	F_{HOSC}=8MHz @ I_HRC/2 & E_HXT/2
		3V	--	0.7	--		
		5V	--	1.3	--	mA	F_{HOSC}=8MHz @ I_HRC/4 & E_HXT/4
		3V	--	0.5	--		
		5V	--	1.1	--	mA	F_{HOSC}=4MHz @ I_HRC/2 & E_HXT/2
		3V	--	0.5	--		
		5V	--	0.9	--	mA	F_{HOSC}=4MHz @ I_HRC/4 & E_HXT/4
		3V	--	0.4	--		
		5V	--	0.9	--	mA	F_{HOSC}=1MHz @ I_HRC/2 & E_HXT/2
		3V	--	0.4	--		
		5V	--	0.8	--	mA	F_{HOSC}=1MHz @ I_HRC/4 & E_HXT/4
		3V	--	0.4	--		
		慢速模式					
		5V	--	6.0	--	uA	F_{HOSC} 关闭， F_{LOSC}=32KHz @ I_LRC/2
		3V	--	2.7	--		
		5V	--	7.0	--	uA	F_{HOSC} 关闭， F_{LOSC}=32KHz @ E_LXT/2.
		3V	--	2.9	--		
		5V	--	4.3	--	uA	F_{HOSC} 关闭， F_{LOSC}=32KHz @ I_LRC/4
		3V	--	1.8	--		
		5V	--	5.3	--	uA	F_{HOSC} 关闭， F_{LOSC}=32KHz @ E_LXT/4.
		3V	--	2.0	--		
I_{STB}	待机电流	5V	--	2.7	--	uA	待机模式，F_{HOSC} 关闭， F_{LOSC}=32KHz @ I_LRC/4
		3V	--	1.1	--		
I_{HALT}	睡眠电流	5V	--	--	0.5	uA	睡眠模式 WDT关闭，.
		3V	--	--	0.2		
		5V	--	--	5.0	uA	睡眠模式 WDT开启.
		3V	--	--	3.0		
R_{PH}	上拉电阻	5V	--	50	--	kΩ	上拉电阻
		3V	--	100	--		
R_{PL}	下拉电阻	5V	--	50	--	kΩ	下拉电阻
		3V	--	100	--		

***注意：这些参数仅供设计参考，并非针对每个芯片进行测试。**

6.3 OSC电气特性

（测量条件VDD电压，T_A 温度等于编程条件）

参数	最小值	典型值	最大值	单位	条件
烧录座的 I_HRC 偏差			±1	%	直接将烧录座装在Writer上。
烧录机台的 I_HRC 偏差			±3	%	正确设置烧录机台。
烧录机台的 I_LRC 偏差			±5	%	

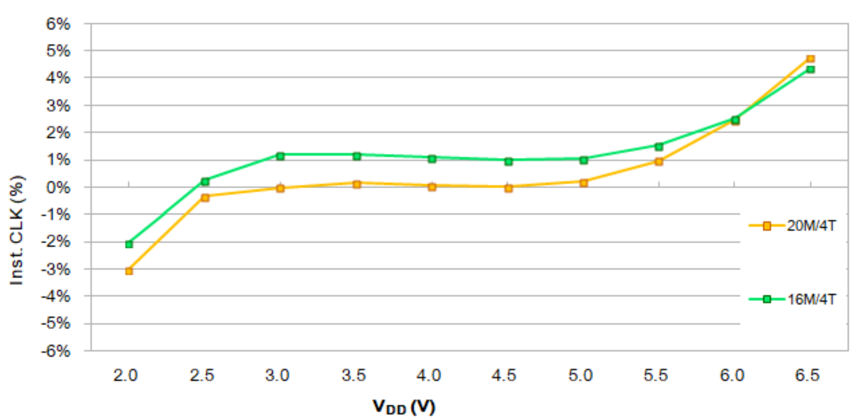
6.4 比较器/LVD电气特性

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

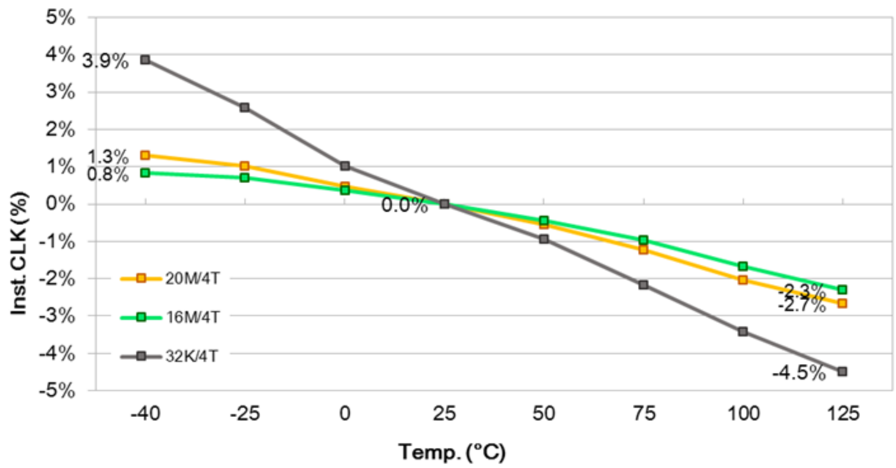
符号	参数	最小值	典型值	最大值	单位	条件
V_{IVR}	比较器输入电压范围	0	--	VDD-1.5	V	F_{HOSC}=1MHz
T_{ENO}	比较器启动等待时间	--	20	--	us	F_{HOSC}=1MHz
I_{CO}	比较器电流消耗	--	70	--	uA	F_{HOSC}=1MHz, P2V mode
I_{LVD}	LVD电流消耗	--	85	--	uA	F_{HOSC}=1MHz, LVD=4.3V
E_{LVD}	LVD电压误差	--	--	3	%	F_{HOSC}=1MHz, LVD=4.3V

6.5 特性曲线图

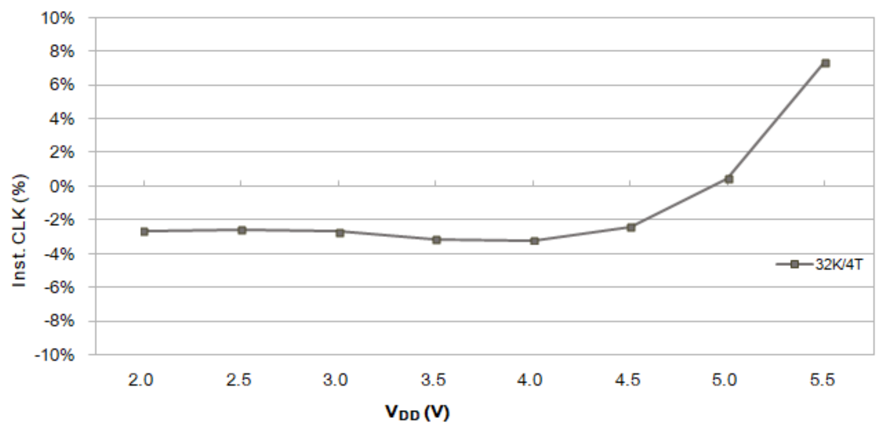
6.5.1 高速 RC 振荡频率(I_HRC)与电源电压(VDD)曲线图



6.5.2 频率与温度曲线图



6.5.3 低速 RC 振荡频率(I_LRC)与电源电压(VDD)曲线图



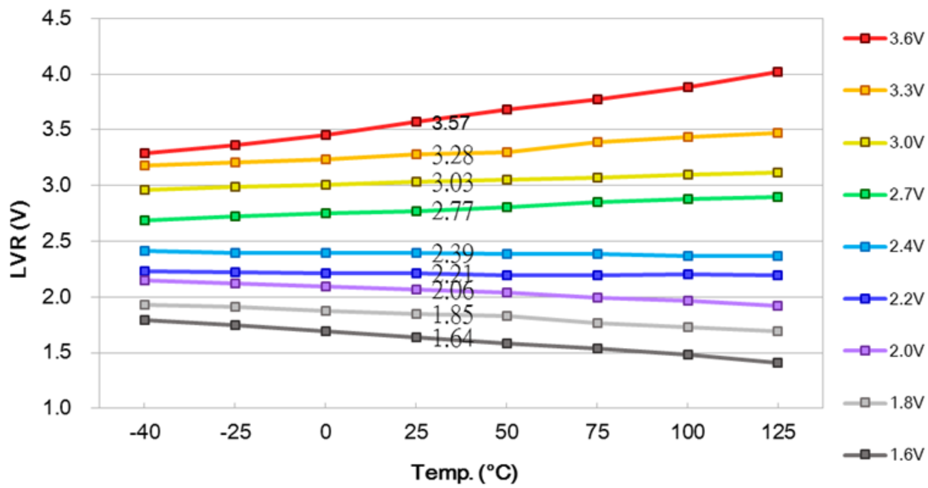
6.6 建议工作电压

建议工作电压（温度范围：-40 °C ~ +85 °C）

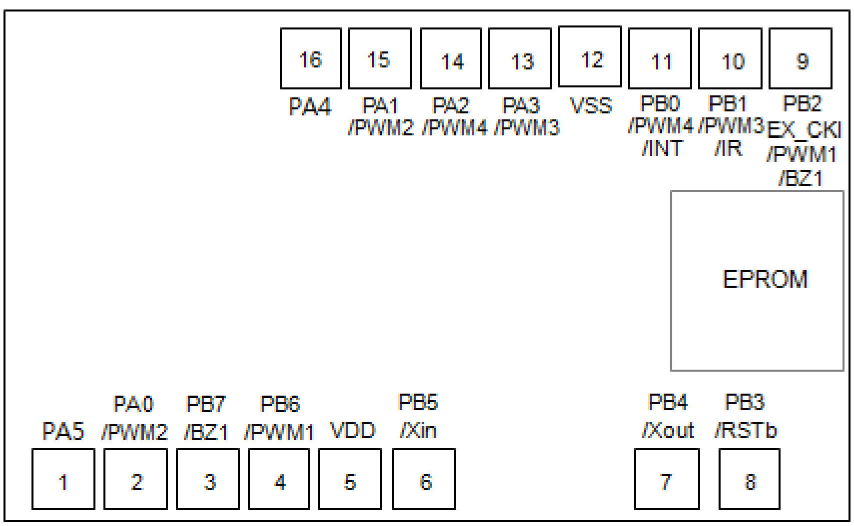
频率	最小电压*	最大电压	LVR：默认 (25 °C)	LVR：建议 (-40 °C ~ +85 °C)
20M/2T	3.3V	5.5V	3.6V	3.6V
16M/2T	2.7V	5.5V	3.0V	3.3V
20M/4T	2.2V	5.5V	2.4V	2.7V
16M/4T	2.0V	5.5V	2.2V	2.4V
8M/4T	1.6V	5.5V	1.8V	2.0V
≦6M/4T	1.6V	5.5V	1.8V	2.0V

***注意：这些参数仅供设计参考，并非针对每个芯片进行测试。**

6.7 LVR电压与温度曲线图

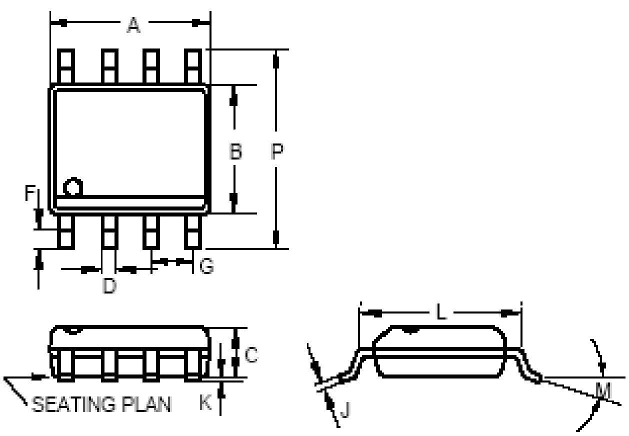


7. 晶圆脚位图



8. 封装尺寸

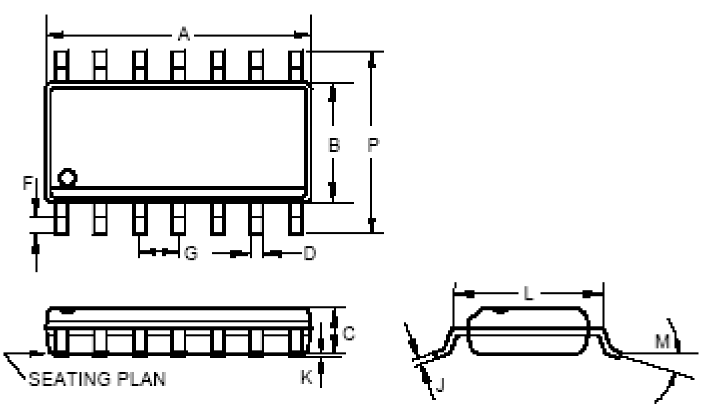
8.1 8 引脚SOP（150 毫寸）



	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	-	-	8º	-	-	8º
P	0.228	-	0.244	5.79	-	6.20

注意：8 引脚 SOP，每管 100 颗。

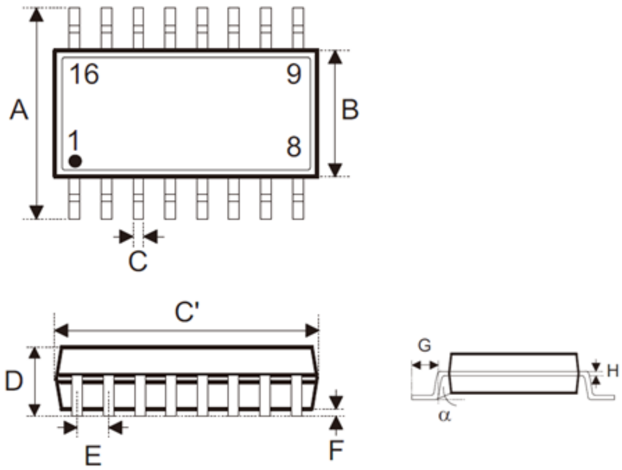
8.2 14 引脚SOP（150 毫寸）



	INCHES			MILLIMETERS		
	MIN	TYP	MAX	MIN	TYP	MAX
A	0.337	-	0.344	8.55	-	8.75
B	0.144	-	0.163	3.66	-	4.14
C	0.068	-	0.074	1.73	-	1.88
D	0.017	-	0.020	0.35	-	0.51
F	0.016	-	0.044	0.40	-	1.12
G	0.050 BSC			1.27 BSC		
J	-	0.004		-	0.10	-
K	0.005	-	0.010	0.13	-	0.25
L	0.189	-	0.205	4.80	-	5.21
M	-	-	8º	-	-	8º
P	0.228	-	0.244	5.80	-	6.20

注意：14 引脚 SOP，每管 50 颗。

8.3 16 引脚SOP（150 毫寸）



注意：16 引脚 SOP，每管 50 颗。

	INCHES			MILLIMETERS		
	MIN	TYP	MAX	MIN	TYP	MAX
A	0.236 BSC			6.00 BSC		
B	0.154 BSC			3.90 BSC		
C	0.012	-	0.020	0.31	-	0.51
C'	0.390 BSC			9.90 BSC		
D	0.065	-	0.069	1.64	-	1.75
E	0.050 BSC			1.27 BSC		
F	0.004	-	0.010	0.10	-	0.25
G	0.016	-	0.050	0.40	-	1.27
H	0.004	-	0.010	0.10	-	0.25
α	-	-	8°	-	-	8°

9. 订购信息

产品名称	封装类型	引脚数	封装尺寸	配送方式
NY8A052ES8	SOP	8	150 mil	卷带：每卷 2.5K。 管装：每管 100 颗。
NY8A052ES14	SOP	14	150 mil	卷带：每卷 2.5K。 管装：每管 50 颗。
NY8A052ES16	SOP	16	150 mil	卷带：每卷 2.5K。 管装：每管 50 颗。