2022.04

MC802

8位 EPROM-Based 单片机
（2 Touch Key + 4 I/O）

Version V1.0

改 版 记 录

版本	*发布日期*	*内 容 描 述*	*修正位置*
1.0	2022-04-10	新版本发布	
1.1		修正部分语句	

目录

1．概述

MC802 带 2 个自校正容性触摸按键功能和 4 个 I/O 口的单片机，是以 EPROM 作为记忆体的 8 位微控制器，专为多 IO 产品的应用而设计，例如遥控器、风扇/灯光控制或是游戏机周边等等。采用 CMOS 制程并同时提供客户低成本、高性能等显著优势。MC802 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式控制完成不同的应用。

在触摸功能方面，MC802 内部搭建 2 路带自校正功能电容式触摸按键的功能，它可以通过任何非导电介质（如玻璃和塑料）来感应电容变化。这种电容感应的开关可以应用在很多电子产品上，提高产品的附加值。该触摸按键为硬件式电容触摸方案，触摸相关逻辑通过内部硬件系统处理，灵敏度的调整由外部电容去调整灵敏度，无需内核软件去干预，所以，在程序处理方面，触摸逻辑算法不会占据内核处理器任何资源，在使用上，用户可以更加便捷使用内部触摸功能。

在 I/O 的资源方面，MC802 内部有 6 根弹性的双向 I/O 脚（实际封装出来有 4 个双向 I/O 脚，内建 2 个 IO 与触摸输出接口相连接），每个 I/O 脚都有单独的寄存器控制为输入或输出脚。而且每一个 I/O 脚位都有附加的程式控制功能如上拉或下拉电阻或开漏极(Open-Drain) 输出。此外针对红外线摇控的产品方面，MC802 内建了可选择频率的红外载波发射口。

MC802 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讯号触发来计数。另外 MC802 提供 1 组 8 位解析度的 PWM 输出或者蜂鸣器输出，可用来驱动马达、LED、或蜂鸣器等等。

MC802 触摸按键功搭载的是自主研发 I2_TOUCH 的 IP，基于电容式充放电原理，实现 2 路带自校准功能电容式触摸按键。I2_TOUCH 是一个独立于 MC802 内核的硬件触摸模块，通过并行接口，提供给内核有效触摸信息，I2_TOUCH 内部可以根据触摸有无被按键下信息，实现自动切换正常模式(Normal)与睡眠模式(Halt mode)之间功能切换，有效节电力消耗延长电池寿命。

MC802 内核搭载是 NY8A051 的单片机，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，NY8A051 内核可选择多种工作模式如正常模式(Normal)、慢速模式(Slow mode)、待机模式(Standby mode) 与睡眠模式(Halt mode)可节省电力消耗延长电池寿命。

在省电的模式下如待机模式(Standby mode)与睡眠模式(Halt mode)中，有多种事件可以触发中断唤醒 MC802 内核进入正常操作模式(Normal) 或慢速模式(Slow mode) 来处理突发事件。

1.1 功能

- **工作电压：**
  - 2.0V ~ 5.5V @系统频率≦8MHz。
  - 2.2V ~ 5.5V @系统频率 > 8MHz

- I2_TOUCH 的 IP 为 2.5V ~ 5.5V(运用 TOUCH 功能，电压选择>2.5V)

- 工作温度：-40°C ~ 85°C。
- 高达 ±5KV 的 ESD。
- 内建 16 阶准确的低电压侦测电路功能。
  - 1Kx14 bits EPROM。
  - 48 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[5:4]与 i2_TOUCH 触摸输出口 OUT[1:0]连接。
- PB[3:0]可选择输入时使用内建下拉电阻。
- 输入有三种组态可选(TTL/CMOS/Without Schmitt)。
- PB[3]内建上拉电阻及输出高推功能。
- PB[5:0]可选择上拉电阻或开漏极输出(Open-Drain)。PB[5:4]一般选择输入状态。
- 所有 I/O 脚输出可选择小灌电流(Small Sink Current)或一般灌电流(Normal Sink Current)。
- 所有 I/O 脚输出可选择小推电流(Small Drive Current)或一般推电流(Normal Drive Current)，除 PB3 外。
- 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)包含可程序化的频率预除线路。
- 一组 8 位下数定时器(Timer1)可选重复载入或连续下数计时。
- 一个 8 位的脉冲宽度调变输出(PWM1)。
- 一个蜂鸣器输出(BZ1)。
- 38/57KHz 红外线载波频率可供选择，同时载波的极性也可以根据数据作选择。
- 内建上电复位电路(POR)。
- 内建低压复位功能(LVR)。
- 内建看门狗计时(WDT)，可由程序固件控制开关。
- 内核采用双时钟机制，系统可以随时切换高速振荡或者低速振荡。
  - 高速振荡：I_HRC (内部 1~20MHz 高速 RC 振荡)
  - 低速振荡：I_LRC (内部 32KHz 低速 RC 振荡)
- 内核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 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。

- MC802 内核在待机模式(Standby mode)下的六种唤醒中断：
  - 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。

- MC802 内核在睡眠模式(Halt mode)下的三种唤醒中断:
  - PB 输入状态改变中断。
  - 外部中断输入。
  - 低电压侦测中断。

1.2 方框图

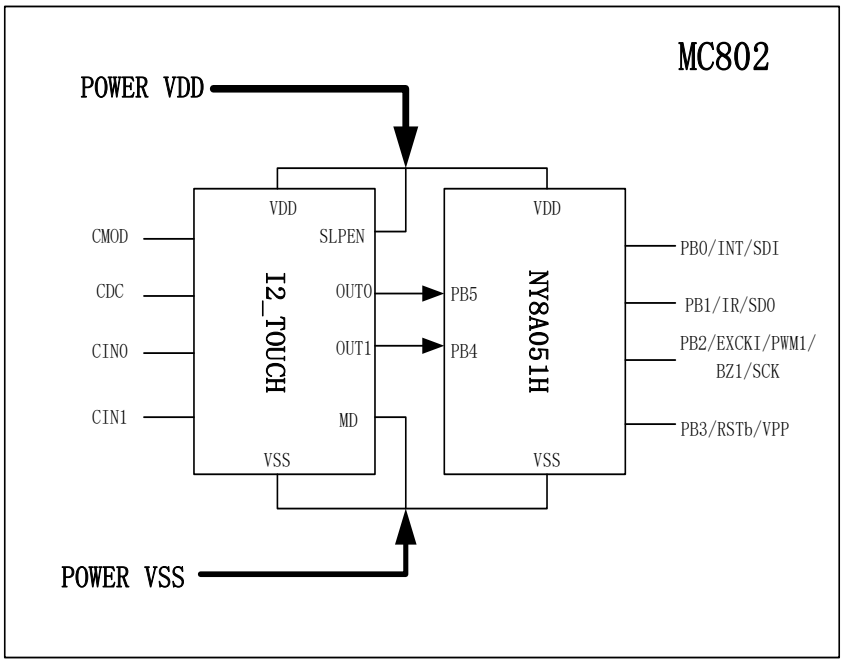


图1-1： MC802系统总框图

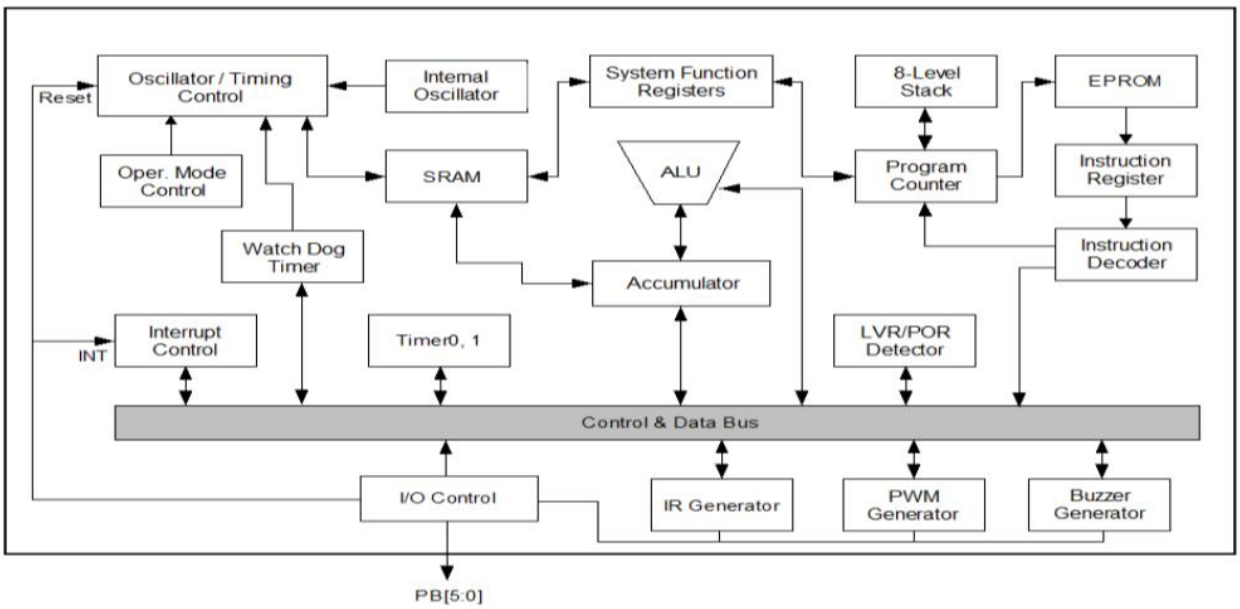


图 1-2： NY8A051H 系统框图

MC802 是由 I2_TOUCH 和 NY8A051H 两部分组成。

I2_TOUCH 主要的功能是执行电容触摸检测功能及触摸按键相关逻辑算法，并将有无触摸信息值传递至输出口 OUT[0:1]。

NY8A051H 是以 EPROM 作为记忆体的 8 位微控制器，专为多 IO 产品的应用而设计。

I2_TOUCH 与 NY8A051H 两部分设计是通过 OUT0 与 PB5，OUT1 与 PB4 相互连接，NY8A051H 通过 PB5，PB4 内部 PAD 获取 I2_TOUCH 的有无触摸按键信息。

1.3 引脚图

MC802采用封装类型：MSOP10

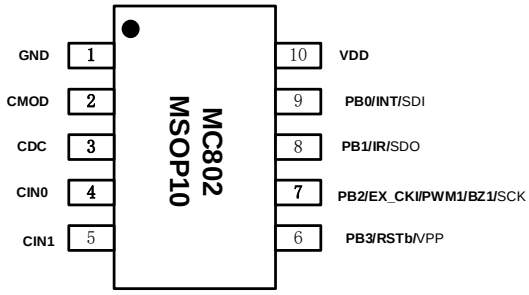


图 1-3： 封装引脚简图

1.4 引脚

引脚汇总

顺序	名称	类型	功能	备注
1	GND	pwr	电源地	-
2	CMOD	I/O	接电荷收集电容	-
3	CDC	I/O	接灵敏度电容	-
4	CIN0	I/O	感应按键0检测输入	未使用悬空
5	CIN1	I/O	感应按键1检测输入	未使用悬空
6	PB3 RSTb Vpp	I/O	PB3 是一个双向I/O引脚。 PB3 可作复位引脚RSTb，若RSTb是低的， PB3 将重置NY8A051H。 B3 也可当成编程脚 Vpp。	
7	PB2 EX_CKI PWM1 BZ1 SCK Comparator	I/O	PB2 是一个双向I/O引脚。 PB2 可当作定时器频率来源EX_CKI。 PB2 可提供PWM输出。 PB2 可当作蜂鸣器输出。 PB2 也可当成编程脚SCK。 PB2 也可当作比较器输出。	
8	PB1 IR SDO	I/O	PB1 是一个双向I/O引脚。 如果启用红外模式，该引脚为红外载波输出。 PB1 也可当成编程脚SDO。	

9	PB0 INT SDI	I/O	PB0 是一个双向I/O引脚。 当EIS=1 & INTIE=1 时，PB0 是外部中断的输入引脚。 PB0 可当成编程脚SDI	
10	VDD	Pwr	电源	-
内部连接	OUT0	I/O	感应按键0输出，与NY8A051H 的PB5连接	I2_TOUCH输出口
内部连接	OUT1	I/O	感应按键1输出，与NY8A051H 的PB4连接	I2_TOUCH输出口
内部连接	PB5	I/O	软件端口设置为输入，获取按键0输出信息，与I2_TOUCH的OUT0连接	运用设置为输入口
内部连接	PB4	I/O	软件端口设置为输入，获取按键1输出信息，与I2_TOUCH的OUT1连接	运用设置为输入口

管脚类型

I 输入 I/O 输入/输出 Pwr 电源 / 地

2．I2_TOUCH 功能

MC802 内部 I2_TOUCH 是一个独立于 MC802 内核的硬件触摸模块。支持检测 2 个触摸感应通道是否被触摸，无需单片机内核干预，并通过 OUT[0:1]传输给 MC802 内部单片机 NY8A051H 的 PB[5:4]端口。

2.1 引脚功能

CMOD：电荷收集电容输入端，接固定值的电容 4.7nF，和灵敏度无关。

CDC：接灵敏度电容，电容范围是最小 5pf ， 最大 100pf。根据使用环境选择合适的电容值，数值越小，灵敏度越高。

CIN0~CIN1：接感应盘，是感应电容的输入检测端口，串接 3KΩ电阻。

OUT0~OUT1：并行一对一输出端口，分别对应 CIN0~CIN1。端口内部结构为带上拉电阻的 NMOS 开漏输出，输出弱高或强低电平，有效电平是强低电平。 OUT0~OUT1 是直接输出模式：检测到手指触摸，输出由高电平变低电平，手指离开后，输出由低电平变高电平。

时段	时段1	时段2	时段3	时段4	时段5	时段6
动作	芯片复位	无手指	手指触摸	无手指	手指触摸	无手指
触摸输出	弱高	弱高	低电平	弱高	低电平	弱高

2.2 I2_TOUCH 初始化

上电复位后，芯片需要 200ms 进行初始化，计算感应管脚的环境电容，然后才能正常工作。

2.3 自校正

根据外部环境温度和湿度等的漂移，按键电容基准参考值也会发生漂移，芯片会自动调整校正每个按键的电容基准参考值，以适应当前环境的变化。

当检测到按键后，芯片会立即停止校正一段时间，这段时间大约 100 秒。停止校正时间一到，芯片会继续自校正，如果当前按键还是持续有效，按键信息会被当做环境的漂移立即被更新，也就是说检测按键有效的时间不会超过 100 秒。

2.4 触摸反应时间

每个通道大约每隔 4.5ms 采样一次。经过按键消抖处理以后，检测到按键按下的反应时间大概是 28 毫秒，检测按键离开的反应时间大概是 18 毫秒。所以检测按键的最快频率大概是每秒 20 次。

2.5 睡眠模式

如果没有触摸的时间超过 70 秒，芯片自动进入睡眠模式。在睡眠模式中，按键的采样间隔会变长，电流消耗（Idd）会减小。如果检测到按键，芯片马上离开睡眠模式，进入正常模式。如下图所示：

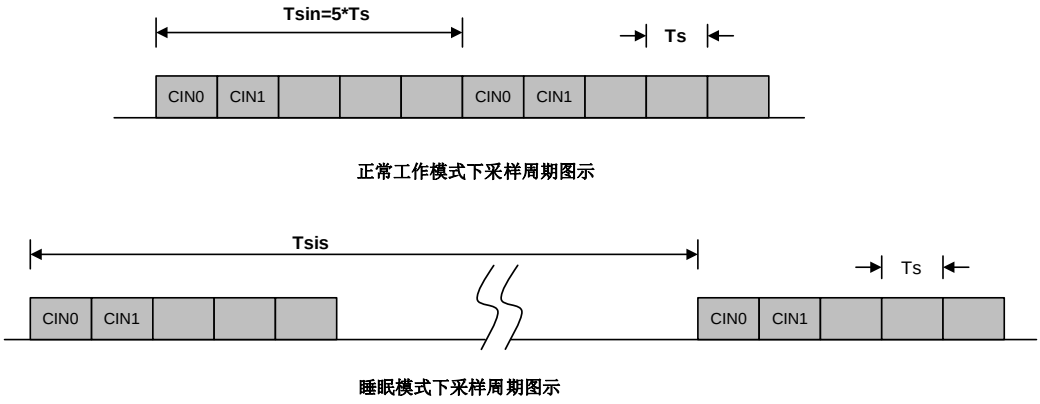


图 2-1：正常和睡眠模式下采样周期图示

注 Ts ：单个按键采样周期，长度大约860微秒

Tsin ：正常工作模式下采样间隔，长度大约4.5毫秒

Tsis ：睡眠模式下采样间隔，长度大约230毫秒

3．内存结构

MC802 内部单片机部分 NY8A051H 存储器分为两类：分别是程序存储器和数据存储器。

3.1 程序内存

NY8A051H 程序存储器空间是 1K。因此，程序计数器（PC）是 10 位宽，以解决程序存储器的任何位置。程序存储器的某些位置保留为中断入口。复位向量地址位于 0x000，软件中断向量地址位于 0x001，内部和外部硬件中断向量地址位于 0x008。

NY8A051H 提供 CALL、GOTOA 和 CALLA 等指令去处理程序空间的 256 个位置。还提供 GOTO 指令去解决程序空间 512 个位置、LCALL 和 LGOTO 指令解决程序空间的任何位置。

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

NY8A051H 程序只读存储器地址 0x3FE~0x3FF 是预留空间。如果用户试图在这些地址写代码会得到意想不到的假函数。

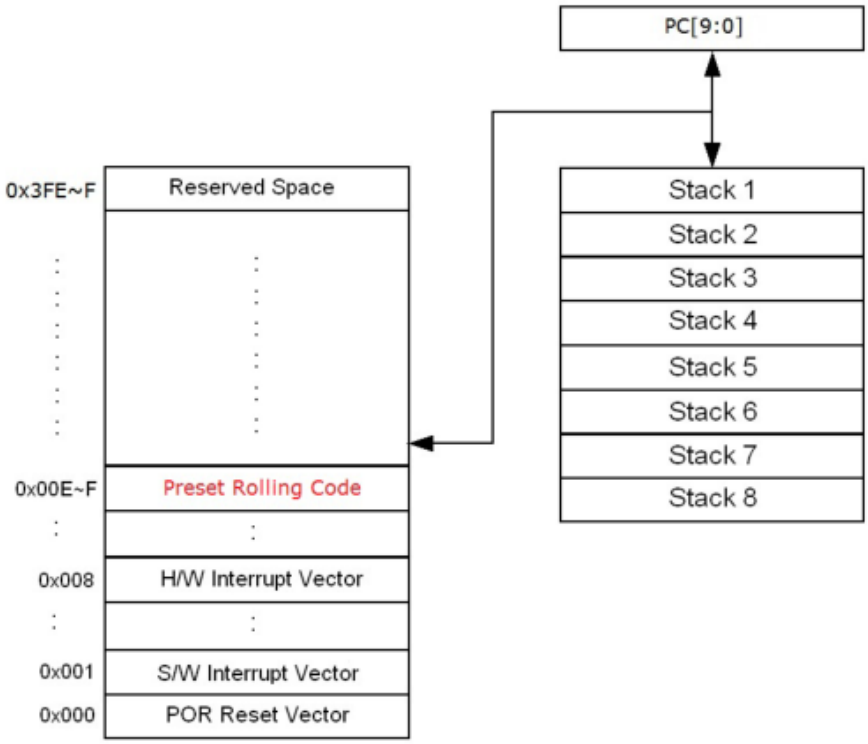


图 3-1： 程序内存映射地址

3.2 数据存储器

根据用于存取数据存储器的指令，数据存储器可分为三类：R-page 特殊功能寄存器(SFR)和通用寄存器(GPR)、F-page 特殊功能寄存器、S-page 特殊功能寄存器。GPR 是由静态存储器组成，用户可以使用它们来存储变量或中间结果。

R-page 数据存储器分为 4 组存储页面，可直接或间接通过 SFR 寄存器（亦为文件选择寄存器 FSR）使用。FSR[7:6]作为存储页面寄存器 BK[1:0]从 4 个存储页面中选择一个。

R-page 寄存器可分为直接寻址方式和间接寻址方式。

数据存储使用的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。存储页面选择是同 FSR[7:6]决定，区位选择则是由 FSR[5:0]而定。

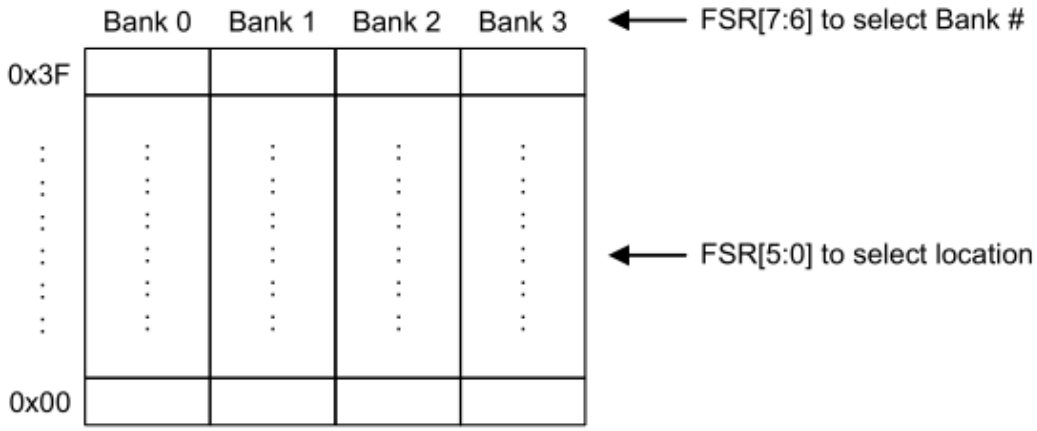


图 3-2： 数据存储器存起的简介寻址方式

下面描述了数据存储使用的直接寻址方式。存储页面选择是由 FSR[7:6]决定，而区位选择则是由 op-code[5:0]指令直接决定。

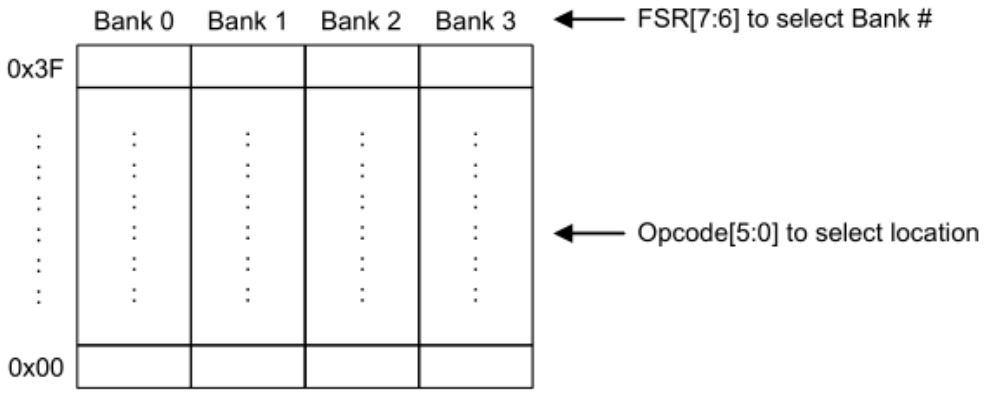


图 3-3： 数据存储器存起的简介寻址方式

特殊功能寄存器 R-page 可以通过一般的指令使用，如算术指令和数据移动指令。R-page 占用了从 0x0 到 Bank 0 的 0xF。然而，Bank 1、Bank 2 和 Bank 3 的相同地址会映射回 Bank 0。换句话说，R-page 完全存在于 Bank 0。GPR 完全占用了存储页面的 0x10 到 0x3F 和其他存储页面 0x10 到 0x3F 映射后的下表所示。

NY8A051H 寄存器名称和特殊功能寄存器 R-page 的映射地址说明如下表：

FSR[7:6] / Address	00 (Bank 0)	01 (Bank 1)	10 (Bank 2)	11 (Bank 3)
0x0	INDF	与Bank 0 相同的映射		
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 ~ 0x1F	通用寄存器	映射至bank0		
0x20 ~ 0x3F	通用寄存器	映射至bank0		

表 3-1 特殊功能寄存器 R-page 地址映射表

特殊功能寄存器 F-page 只能被指令 IOST 和 IOSTR 使用，特殊功能寄存器 S-page 只能被指令 SFUN 和 SFUNR 使用。当 F-page 和 S-page 寄存器被使用时，FSR[7:6]存储页面选择位会被忽略。寄存器名称和 F-page、S-page 的映射地址说明如下表。

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	-	BZ1CR
0x6	IOSTB	IRCR
0x7	-	TBHP
0x8	-	TBHD
0x9	-	
0xA	PS0CV	
0xB	-	
0xC	BODCON	
0xD	-	
0xE	-	
0xF	PCON1	OSCCR

表 3-2 特殊功能寄存器 F-page 和 S-page 地址映射表

4. 功能描述

本章节将详细描述 MC802 内部的 NY8A051H 的操作方式。

4.1 R 页面特殊功能寄存器

4.1.1 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。任何指令存取接寻址寄存器时，实际上是读取文件选择寄存器所选择的寄存器。

4.1.2 TMR0 （定时器 T0 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							

当读取定时器 T0 寄存器时，实际上读取 Timer0 目前运行数值。

写入定时器 T0 寄存器会变化 Timer0 目前的数值。

藉由 T0MD 与配置字节设定，Timer0 时钟源可以从指令时钟 F_{INST}、外部脚位 EX_CKI 或低振荡频率中择一。

4.1.3 PCL（PC[9:0]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			xxxxxxxx							

程序计数寄存器是最显着的 10 位 PC 字节（LSB）。在指令一一被执行后，PCL 会增加，除了某些指令会直接变化 PC。PC 的高字节如 PC[9:8]并不能直接存取。更新 PC[9:8]必须藉由 PCHBUF 寄存器完成。

以 GOTO 指令来说，PC[8:0]是从指令值取得而[9]是从 PCHBUF[1]加载。CALL 指令的 PC[7:0]是从指令值取得而 PC[9:8]是从 PCHBUF[1:0]加载。下一个 PC 地址如 PC+1 时，将会存到堆栈的顶部。LGOTO 指令的 PC[9:0]是从指令值取得。

LCALL 指令的 PC[9:0]是从指令值取得；当下一个 PC 地址如 PC+1 时，将被存到堆栈的顶部。

4.1.4 STATUS （状态寄存器）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
STATUS	R	0x3	STATUS [7:0]							
读/写属性			读/写							

初始值	xxxxxxxx

状态寄存器包含算术指令的结果和导致复位的原因。

C：进位/借位标志位

C=1 时，加法指令或减法指令不借位都会导致进位。

C=0 时，加法指令或减法指令不借位都不会导致进位。

DC：半进位/半借用标志位

DC=1 时，加法指令或第四个 LSB 的减法指令不借位都会导致第四个 LSB 的进位。

DC=0 时，加法指令或第四个 LSB 的减法指令不借位都不会导致第四个 LSB 的进位。

Z：零位

Z=1 时，逻辑运算的结果是零。

Z=0 时，逻辑运算的结果不为零。

/PD：掉电标志位

/PD=1 时，上电或执行 CLRWDT 指令后。

/PD=0 时，执行 SLEEP 指令后。

/TO：时间上溢标志位

/TO=1 时，上电或执行 CLRWDT 或 SLEEP 指令后。

/TO=0 时，发生 WDT 超时。

GP7 、GP6 、GP5 ： 通用读写寄存器。

(*1)：可以被 SLEEP 指令清除。

(*2)：可以由 CLRWDT 指令设定。

4.1.5 FSR（文件选择寄存器）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FSR	R	0x4	BK[1:0]		FSR [5:0]					
读/写属性			读/写							
初始值			0	0		X	X	X	X	X

FSR[5:0]：从特定存储区的 64 个寄存器中选择一个。

BK[1:0] ： 以 NY8A051H 为例，存储区寄存器并不会被使用，因为 NY8A051H 仅有一个存储区。

4.1.6 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	GP7	PB5	PB4	PB3	PB2	PB1	PB0
读/写属性			读/写							
初始值			数据锁存值是xxxxxx，读取值则是xxxxxx端口值（PB5~PB0）							

取 PortB 时，若特定脚位被配置为输入脚，将得到该状态。然而，若该脚位被配置为输出脚，不论是得到该脚位的状态或相对应的输出数据锁存值，都是依据配置字 RD_OPT。当写入 PortB 时，数据是被写入 PortB 的输出输去闩锁中。

GP7 ，GP6 ： 通用读/写寄存器。

4.1.7 PCON（电源控制寄存器）

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

GP5~0 ： 一般通用读/写寄存器。

LVREN ： 打开/关闭 LVR。

LVREN=1 时，打开 LVR。

LVREN=0 时，关闭 LVR。

EIS ： 外部中断选择位。

EIS=1 时，PB0 是外部中断。

EIS=0 时，PB0 是通用读/写寄存器。

LVDEN ： 打开/关闭 LVD。

LVDEN=1 时，打开 LVD。

LVDEN=0 时，关闭 LVD。

WDTEN ： 打开/关闭 WDT。

WDTEN=1 时，打开 WDT。

WDTEN=0 时，关闭 WDT。

CMPEN ： 打开/关闭 CMP。

CMPEN=1 时，打开 CMP。

CMPEN=0 时，关闭 CMP。

4.1.8 BWUCON（PortB 唤醒控制寄存器 ）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BWUCON	R	0x9	-	-	WUPB5	WUPB4	WUPB3	WUPB2	WUPB1	WUPB0
读/写属性					读/写	读/写	读/写	读/写	读/写	读/写
初始值			X	X	1	1	1	1	1	1

WUPBx ： 打开/关闭 PBx 唤醒功能，$0 \le x \le 5$。

WUPBx=1 时，开启 PBx 唤醒功能。

WUPBx=0 时，关闭 PBx 唤醒功能。

4.1.9 PCHBUF（程序计数器的高字节）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCHBUF	R	0xA	-	-	-	-	-	GP5	PCHBUF[1:0]	
读/写属性			-	-	-	-	-	读/写	写	
初始值			X	X	X	X	X	0	0 0	

PCHBUF[1:0] ： 第 9 个位的缓冲器、PC 的第八个位。

GP5 ： 通用读写寄存器。

4.1.10 BPLCON（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
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		1	1	1

/PLPBx ： 关闭/使能 PBx 下拉电阻，0 ≤ x ≤ 3。

/PLPBx=1 时，关闭 PBx 下拉电阻。

/PLPBx=0 时，开启 PBx 下拉电阻

4.1.11 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/使能 PBx 上拉电阻，0 ≤ x ≤ 5。

/PHPBx=1 时，关闭 PBx 上拉电阻。

/PHPBx=0 时，开启 PBx 上拉电阻。

4.1.12 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	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 T0 溢出（overflow）中断使能。

T0IE=1 时，打开定时器 T0 溢出中断。

T0IE=0 时，关闭定时器 T0 溢出中断。

PBIE ： PortB 输入变化中断使能。

PBIE=1 时，打开 PortB 输入变化中断。

PBIE=0 时，关闭 PortB 输入变化中断。

INTIE ： 外部中断使能。

INTIE=1 时，打开外部中断。

INTIE=0 时，关闭外部中断。

T1IE ： 定时器 T1 溢出中断使能。

T1IE=1 时，打开定时器 T1 溢出中断。

T1IE=0 时，关闭定时器 T1 溢出中断。

LVDIE ： LVD 中断使能。

LVDIE=1 时，打开 LVD 中断。

LVDIE=0 时，关闭 LVD 中断。

WDTIE ： WDT 超时中断使能。

WDTIE=1 时，打开 WDT 超时中断。

WDTIE=0 时，关闭 WDT 超时中断。

4.1.13 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(注意*)			-	读/写	-	读/写	读/写	读/写	读/写	读/写
初始值			0	0	0	0	0	0	0	0

T0IF ： 定时器 T0 溢出中断标志位。

T0IF=1 时，发生定时器 T0 溢出中断。

T0IF 必须被硬体清零。

PBIF ： PortB 输入变化中断标志位。

PBIF=1 时，发生 PortB 输入变化中断。

PBIF 必须被硬体清零。

INTIF ： 外部中断标志位。

INTIF=1 时，发生外部中断。

INTIF 必须被硬体清零。

T1IF ： 定时器 T1 下溢中断标志位。

T1IF=1 时，发生定时器 T1 下溢中断。

T1IF 必须被硬体清零。

LVDIF ： LVD 中断标志位。

LVDIF=1 时，发生 LVD 中断。

LVDIF 必须被硬体清零。

WDTIF ： WDT 超时中断标志位。

WDTIF=1 时，发生 WDT 超时中断。

WDTIF 必须被硬体清零。

注意：当对应的 INTE bit 未使能，读取中断标志是 0 。

4.2 T0MD 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
TMOD	-	-	LCKTM0	INTEDG	T0CS	T0CE	PS0WDT		PS0SEL[2:0]	
读/写属性			读/写							
初始值（注意*）			0	0	1	1	1		111	

PS0SEL[2:0] ： 选择预分频器 P0 分配比。分配比是根据预分频器 P0 分配给定时器 T0 或 WDT 决定。当预分频器 P0 被分配给 WDT，分配比取决于选择哪种超时机制。

PS0SEL[2:0]	分配比		
	PS0WDT=0 （ 定时器T0 ）	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

PS0WDT：预分频器 P0 分配。

PS0WDT=1 时，预分频器 P0 被分配到 WDT。

PS0WDT=0 时，预分频器 P0 被分配到定时器 T0。

注意：在使能看门狗或时间中断前，要设定 PS0WDT 和 PS0SEL[2:0] ，否则复位或中断可能导致错误触发。

T0CE：定时器 T0 外部时钟沿选择。

T0CE=1 时，定时器 T0 将加一当高到低转换发生在 EX_CKI 脚。

T0CE=0 时，定时器 T0 将加一当低到高转换发生在 EX_CKI 脚。

注意： T0CE 也应用在低频振荡频率作为定时器 T0 时钟源条件。

T0CS ：定时器 T0 时钟源选择。

T0CS=1 时，选择 EX_CKI 脚的外部时钟或低振荡频率。

T0CS=0 时，选择指令时钟 F_{INST} 。

INTEDG ：外部中断的边沿选择。

INTEDG=1，当上升沿发生在 PB0 脚时，INTIF 将被设定。

INTEDG=0，当下降沿发生在 PB0 脚时，INTIF 将被设定。

LCKTM0 ：T0CS=1 时，定时器 T0 时钟源可以选择低频振荡器。

T0CS=0 时，指令时钟 F_{INST} 被选作定时器 T0 时钟源。

T0CS=1 时，LCKTM0=0 时，EX_CKI 脚的外部时钟被选择当作定时器 T0 时钟源。

T0CS=1 时，LCKTM0=1 时，低振荡频率输出代替 EX_CKI 脚作为定时器 T0 时钟源。

注意：有关定时器 T0 时钟源选择的详细说明，请参考定时器 T0 章节。

4.3 F 页面特殊功能寄存器

4.3.1 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	GP7	GP6	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			0	0	1	1	1	1	1	1

IOPBx ： PBx I/O 模式选择，0 ≤ x ≤ 5。

IOPBx=1 时，PBx 是输入模式。

IOPBx=0 时，PBx 是输出模式。

GP7 ，GP6 ： 通用寄存器。

4.3.2 PS0CV（预分频器 P0 计数器值寄存器）

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

读 PS0CV 时，会得到预分频器 P0 计数器的目前数值。

4.3.3 BODCON（(PortB 开漏极寄存器）

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

ODPBx ： 打开/关闭 PBx 的开漏 极 ， 0 ≤ x ≤ 5。

ODPBx=1 时，打开 PBx 的开漏极。

ODPBx=0 时，关闭 PBx 的开漏极。

4.3.4 CMPCR （比较器电压选择控制寄存器）

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

NS[1:0] ： 比较器反向输入选择。

NS[1:0]	反向输入
00	PB1
1	--
10	Bandgap (0.6V)
11	Vref

PS[1:0] ： 比较器正向输入选择。

NS[1:0]	正向输入
00	PB0
1	--
10	Vref
11	---

CMPF_INV ： 比较器输出反相控制位。

CMPF_INV = 1，反向比较器输出。

CMPF_INV = 0，正向比较器输出。

RBIAS_L ，RBIAS_H ： 设置相应的电压参考电平。*（请参 考章节 4.10.1 ）*

4.3.4 PCON1（电源控制寄存器 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*)	-	读/写	读/写	读/写	读/写	读/写	读/写
初始值			0	0						

T0EN ： 打开/关闭定时器 T0。

T0EN=1 时，打开定时器 T0。

T0EN=0 时，关闭定时器 T0。

GIE ： 全局中断使能位。

GIE=1 时，打开所有未标记中断。

GIE=0 时，关闭所有的中断。

LVDOUT：低电压检测输出，只读。

LVDS3~0 ： 从 11 种 LVD 电压中选择其中一个

LVDS[3:0]	LVDL Voltage	LVDH Voltage	说明
0000	2.00V	2.03V	考虑I2_TOUCH工作电压
0001	2.20V	2.23V	
0010	2.38V	2.41V	
0011	2.79V	2.82V	
0100	2.88V	2.91V	
0101	2.97V	3.00V	
0110	3.27V	3.30V	
0111	3.58V	3.61V	
1000	3.90V	3.93V	
1001	1.93V	1.96V	
1010	4.04V	4.07V	
1011	2.58V	2.61V	
1100	4.13V	4.16V	
1101	3.13V	3.16V	
1110	3.45V	3.48V	

表 3-1 LVD 电压选择

LVDL 表示 VDD 电压从高过渡到低，LVDH 表示 VDD 电压由低过渡到高。

GP1：通用读写寄存器。

(1*) ：由指令 ENI 所设定、指令 DISI 清除、指令 IOSTR 读取。

4.4 S 页面特殊功能寄存器

4.4.1 TMR1 （定时器 T1 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，会得到目前 8-bit 递减计数定时器 T1 的目前数值。当写入 TMR1 寄存器时，数据将写入定时器 T1 寄存器并更新定时器 T1 目前内容。

4.4.2 T1CR1（定时器 T1 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			写入	写入	-	-	-	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 T1 功能。

T1EN：使能/关闭定时器 T1。

T1EN=1 时，开启定时器 T1。

T1EN=0 时，关闭定时器 T1。

T1RL：当不停止模式被选择（T1OS=0），配置定时器 T1 递减计算器制。

T1RL=1 时，初始值从 TMR1 寄存器被重新加载。

T1RL=0 时，发生溢出也继续从 0xFF 递减计数。

T1OS：当达到溢出时，配置定时器 T1 操作模式。

T1OS=1 时，单次模式。定时器 T1 会从初始值到 0x00 计数一次。

T1OS=0 时，不停止模式。溢出后，定时器 T1 会持续递减计数。

T1OS	T1RL	
0	0	定时器T1 从重载的数值倒数到 0x00。 当达到溢出，0xFF被重载并继续递减计数。
0	1	定时器T1 从重载的数值倒数到 0x00。 当达到溢出，再次重载数值并寄续递减计数
1	X	定时器T1 从初始值倒数到 0x00。 当达到溢出，定时器T1 停止计数。

表 3-1 定时器 T1 功能

PWM1OAL：定义 PWM1 输出活动状态。

PWM1OAL=1 时，PWM1 输出为低电平。

PWM1OAL=0 时，PWM1 输出为高电平。

PWM1OEN：打开/关闭 PWM1 输出。

PWM1OEN=1 时，PB2 会输出 PWM1。

PWM1OEN=0 时，PB2 是通用 IO。

4.4.3 T1CR2（定时器 T1 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 T1 功能。

PS1SEL[2:0] ：预分频器 PS1 分配比选项。

PS1SEL[2:0]	分配比
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 3-2　预分频器 PS1 分配比

注意：须设定 PS1SEL[2:0] 在 PS1EN=1 ， 否则中断可能会被误触发。

/PS1EN：关闭/打开预分频器 PS1。

/PS1EN=1 时，关闭预分频器 PS1。

/PS1EN=0 时，开启预分频器 PS1。

T1CE：定时器 T1 外部时钟边沿选项。

T1CE=1 时，当高到低转换发生在 EX_CKI 脚时，定时器 T1 会减一。

T1CE=0 时，当低到高转换发生在 EX_CKI 脚时，定时器 T1 会减一。

T1CS：定时器 T1 时钟源选项。

T1CS=1 时，选择在 EX_CKI 脚的外部时钟。

T1CS=0 时，选择指令时钟。

4.4.4 PWM1DUTY（PWM1 占空寄存器）

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

此寄存器仅能写入。当定时器 T1 被开启并开始计时后，PWM1 输出会保持在非活动状态。当定时器 T1 数值等同 PWM1DUTY，PWM1 输出会进入活动状态直到发生溢出。

定时器 T1 重加载的数值储存在 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。

4.4.5 PS1CV（ 预分频器 PS1 计数值寄存器）

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

读取 PS1CV 时，将会得到预分频器 PS1 计数器的目前数值。

4.4.6 BZ1CR（蜂鸣器 BZ1 计数寄存器）

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

BZ1FSEL[3:0] ： BZ1 输出的频率选项。

BZ1FSEL[3:0]	BZ1 频率选项	
	时钟源	时钟源
0000	预分频器PS1 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T1 输出	定时器T1 bit 0
1001		定时器T1 bit 1
1010		定时器T1 bit 2
1011		定时器T1 bit 3
1100		定时器T1 bit 4
1101		定时器T1 bit 5
1110		定时器T1 bit 6
1111		定时器T1 bit 7

表 3-3 蜂鸣器 1 输出（PB2）频率选项

BZ1EN：打开/关闭 BZ1 输出。

BZ1EN=1 时，开启蜂鸣器 1。

BZ1EN=0 时，关闭蜂鸣器 1。

4.4.7 IRCR（IR 控制寄存器）

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

IREN：打开/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：选择 IR 载波频率。

IRF57K=1 时，IR 载波频率是 57KHz。

IRF57K=0 时，IR 载波频率是 38KHz。

IRCSEL：选择 IR 载波极性。

IRCSEL=0，且 I/O 脚数据是 1 时，产生 IR 载波。

IRCSEL=1，且 I/O 脚数据是 0 时，产生 IR 载波。

注意:

1. 仅有高振荡（ F_{HOSC} ）（ 详见章节 3.12 ） 可以当作 IR 时钟源。
2. 不同振荡类型的分配比。

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

表 4-4 不同振荡类型的分配比

4.4.8 TBHP（ 表格存取高字节地址指针寄存器）

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，目标地址是由 TBHP[2:0]与 ACC 组成。ACC 是 PC[9:0]的低字节，TBHP[1:0]是 PC[9:0]的高字节。TBHP[2]是 NY8A051H 的通用寄存器。

4.4.9 TBHD（ 表格存取高字节数据寄存器）

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，寻址 ROM 的高字节内容被加载到 TBHD[5:0]寄存器。寻址 ROM 的低字节内容则被加载到 ACC。

4.4.10 OSCCR（ 振荡控制寄存器）

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：系统振荡选项（Fosc）

SELHOSC=1 时，Fosc 是高频率振荡（F_{HOSC}）

SELHOSC=0 时，Fosc 是低频率振荡（F_{LOSC} ）

STPHOSC：关闭/打开高频率振荡（F_{HOSC}）。

STPHOSC=1 时，FHOSC 会停止振荡并被关闭。

STPHOSC=0 时，FHOSC 保持振荡。

CMPOEN ：打开/关闭比较器输出。

CMPOEN=1 时，比较器输出打开。

CMPOEN =0 时，比较器输出关闭。

OPMD[1:0] ： 选择操作模式。

OPMD[1:0]	操作模式
00	一般模式
01	停止模式
10	待机模式
11	保留

表 4-5 选择 OPMD[1:0]的操作模式

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

4.5 I/O 端口

NY8A051H 提供 6 个 I/O 脚位（PB[5:0]）（其中，PB[5:4] 与 I2_TOUCH 的输出口 OUT[1:0]内部相连接，用户可以使用只有 4 个 I/O 引脚）。用户可以藉由寄存器 PortB 读写这些脚位。每个 I/O 脚位都有一个对应的寄存器位以定义该脚位是输入或输出脚。寄存器 IOSTB[5:0]定义 PB[5:0]输入/输出方向。

当一个 I/O 脚位被配置为输入脚，它可以藉由寄存器开启或关闭上拉或下拉电阻。寄存器 BPHCON[5:0]用于开启或关闭 PB[5:0]的上拉电阻。寄存器 BPLCON[7:4]则是用于开启或关闭 PB[3:0]的下拉电阻。

当一个 I/O 脚位被配置为输出脚，相对应的个别寄存器会选择开漏极输出脚。寄存器 BODCON[5:0]决定 PB[5:0]是否为开漏极。

I/O 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	开漏极	V	V

表 4-6 I/O 功能摘要

电平变化在PB的每个I/O脚可能会产生中断。寄存器BWUCON[5:0]会在选择PB任一I/O脚位时产生中断。只要选择到对应到BWUCON的任一PB脚位，且电平变化发生在任一以选择的脚位时，寄存器位 PBIF (INTF[1])就会设定为 1。如果寄存器位 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。NY8A051H仅提供一个外部中断，当寄存器位 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 电平 变化触发与外部中断，外部中断有较高的优先，而 PB0 电平 变化触发则会被关闭，但 PB5~PB1 电平变化触发不会被影响。

NY8A051H提供IR载波生成器。IR载波生成器是由寄存器位 IREN (IRCR[0])使能，脚位PB1 会输出载波。

PB3 可当作外部复位输入，由配置字决定。当低电平信号应用于PB3 将会导致NY8A051H进入复位程序。.

当NY8A051H处于一般模式或待机模式并打开配置字节，用户可以在PB4 观察指令时钟。

如果T0MD T0CS=1 和LCK_TM0=0，PB2 可以当作定时器T0 外部时钟源EX_CKI。如果T1CS=1，PB2 可以当作定时器T1 外部时钟源。

如果T1CR1[7] PWM1OEN=1 并且对应到PB.2 组态，PB2 也可以当作脉冲宽度调制输出。若BZ1CR[7] BZ1EN=1并对应到PB.2 配置字，PB2 也可以当作蜂鸣器输出。

4.5.1 IO 引脚结构框图

IO_SEL：设定引脚的属性为输入或输出。

WRITE_EN：将资料写入引脚。

READ_EN：读取引脚。

OD_EN：打开开漏极。

PULLUP_ENB：打开Pull-High。

PULLDOWN_EN：打开Pull-Low。

RD_TYPE：选择读取脚位或闩锁。

EIS：打开外部中断功能。

INTEDGE：选择外部中断边沿。

EX_INT：外部中断信号。

WUB：打开port B唤醒。

SET_PBIF：port B 唤醒标志。

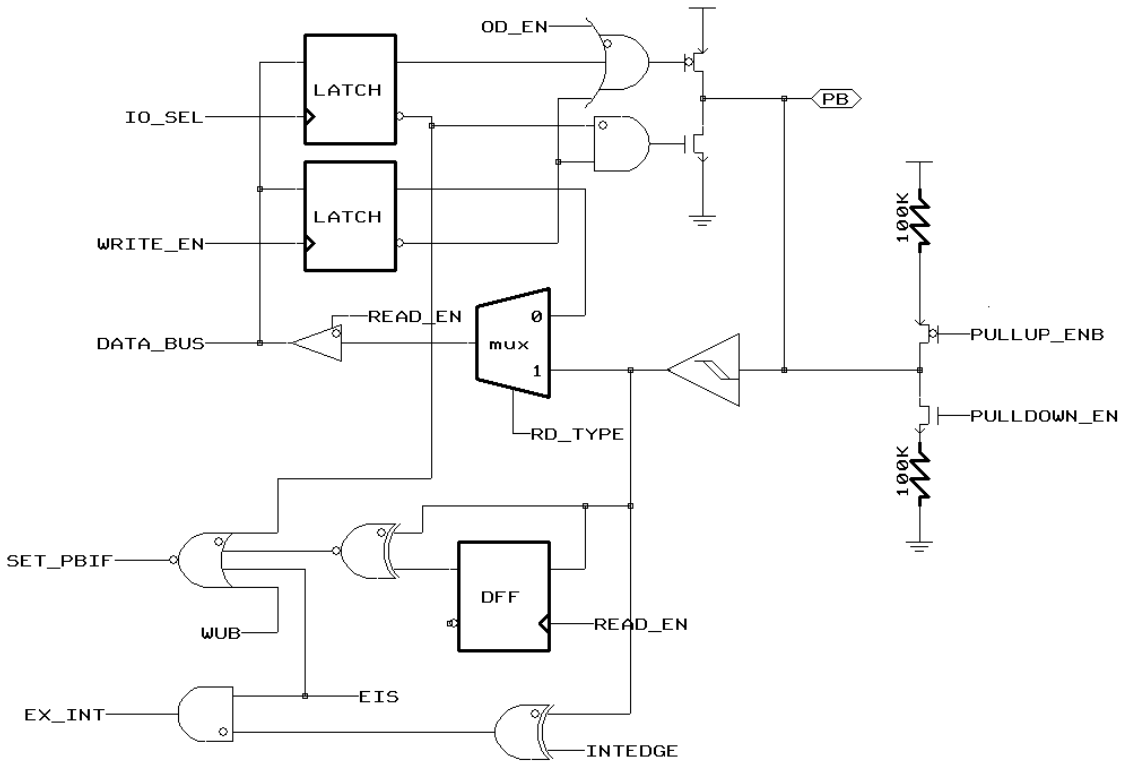


图 4-1　PB0 结构框图

IO_SEL：设定引脚属性为输入或输出。

WRITE_EN：将资料写入引脚。

READ_EN：读取引脚。

OD_EN：打开开漏极。

PULLUP_ENB：打开Pull-High。

PULLDOWN_EN：打开Pull-Low。

RD_TYPE：选择读取脚位或闩锁。

IREN：打开IR功能。

IRDT：IR资料。

WUB：打开port B唤醒。

SET_PBIF：port B唤醒标志。

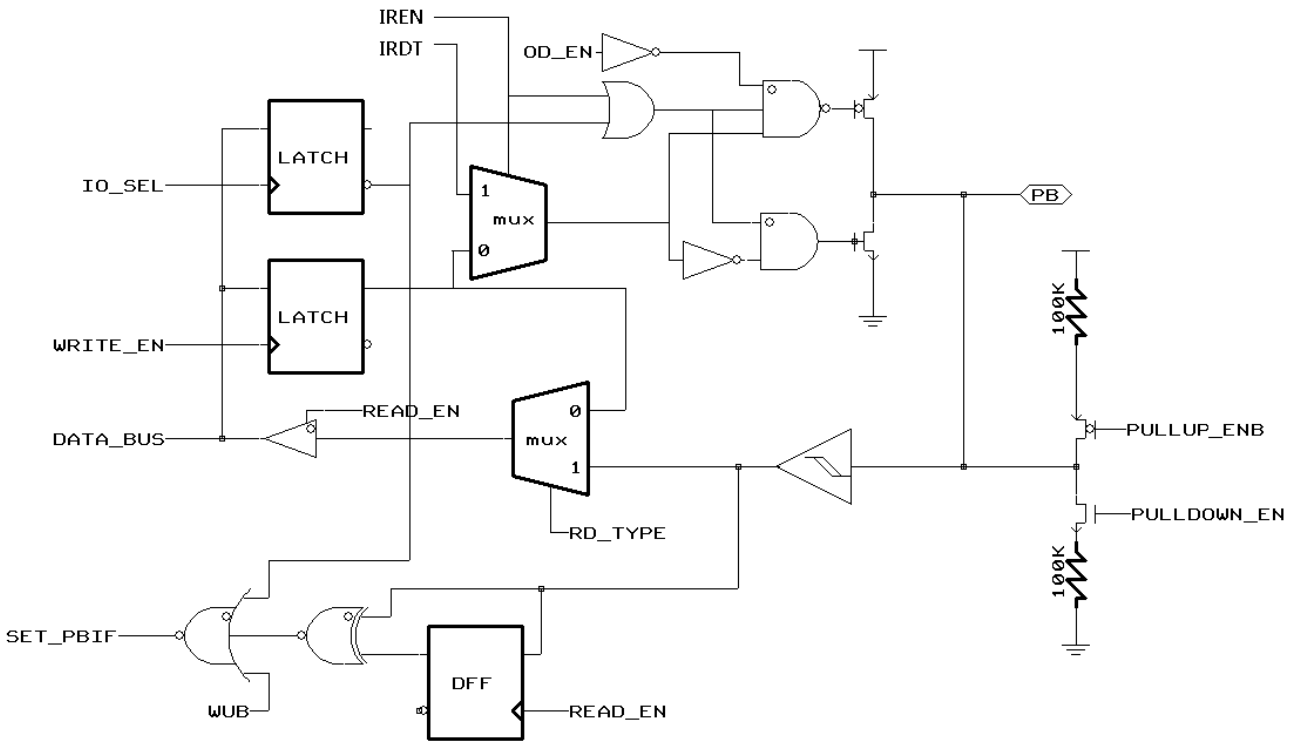


图 4-2　PB1 结构框图

IO_SEL：设定引脚属性为输入或输出。

WRITE_EN：将资料写入引脚。

READ_EN：读取引脚。

OD_EN：打开开漏极。

PULLUP_ENB：打开Pull-High。

PULLDOWN_EN：打开Pull-Low。

RD_TYPE：选择读取脚位或闩锁。

PBEN：打开PMW/BUZZER。

PBDT：PMW/BUZZER资料。

WUB：打开port B唤醒。

SET_PBIF：port B 唤醒标志。

EX CKI：外部时钟输入。

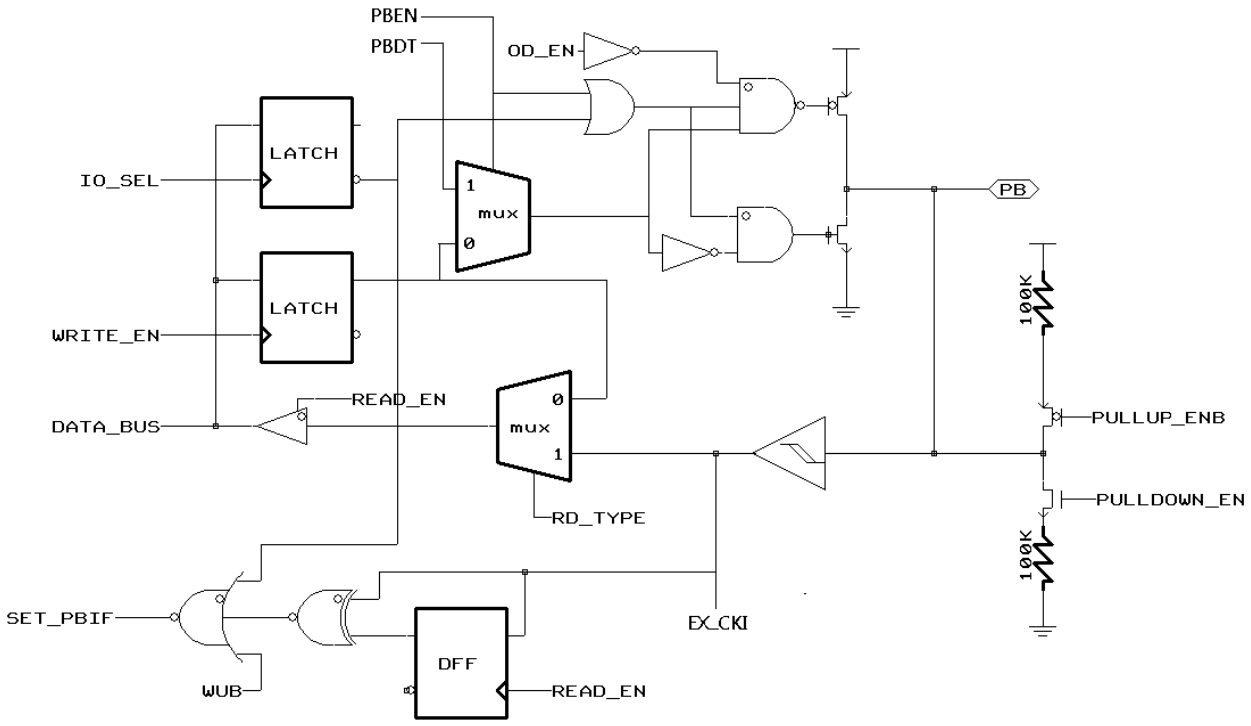


图 4-3　PB2 结构框图

IO_SEL：设定引脚属性为输入或输出。

WRITE_EN：将资料写入引脚。

READ_EN：读取引脚。

OD_EN：打开开漏极。

PULLUP_ENB：打开Pull-High。

PULLDOWN_EN：打开Pull-Low。

RD_TYPE：选择读取脚位或闩锁。

PBEN：打开PMW/BUZZER。

PBDT：PMW/BUZZER资料。

WUB：打开port B唤醒。

SET_PBIF：port B唤醒标志。

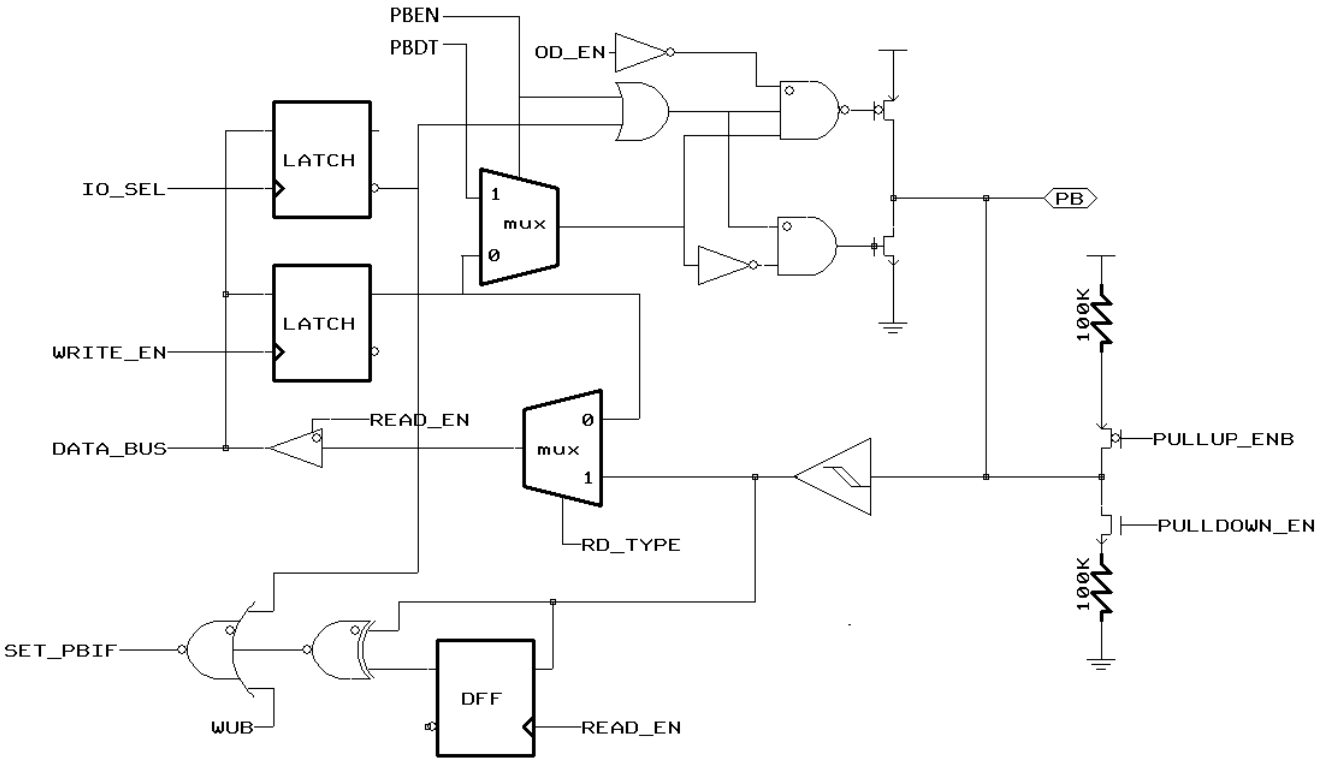


图 4-3　PB3 结构框图

IO_SEL：设定引脚属性为输入或输出。

WRITE_EN：将资料写入引脚。

READ_EN：读取引脚。

OD_EN：打开开漏极。

PULLUP_ENB：打开Pull-High。

RD_TYPE：选择读取脚位或闩锁。

WUB：打开port B唤醒。

SET_PBIF：port B唤醒标志。

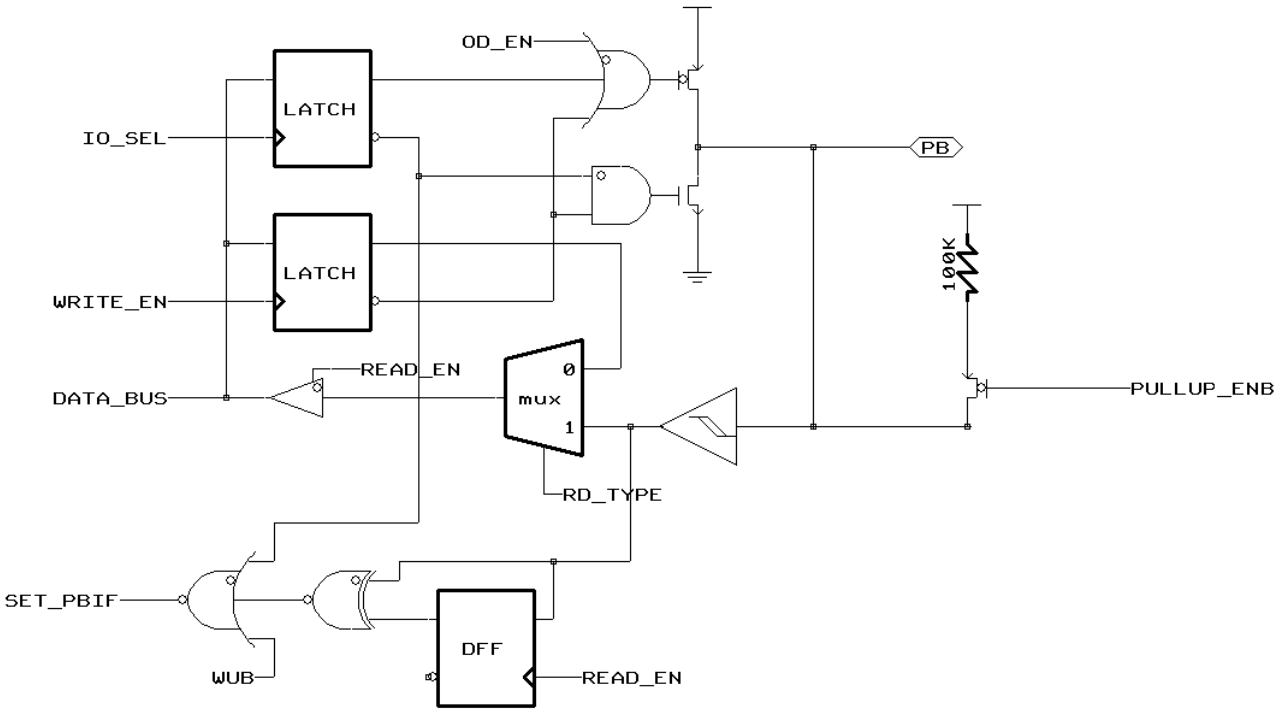


图 4-4　PB4/PB5 结构框图

4.6 定时器 T0

定时器T0 是 8 位递增定时器，藉由寄存器T0EN (PCON1[0]) 位打开操作。写入定时器T0 将会设定其初始值；读取定时器T0 时则会显示目前的计数值。

定时器T0 的时钟源根据寄存器T0CS位、LCK_TM0 (T0MD[5]与T0MD[7]) 位可以从指令时钟、外部脚位EX_CKI或低速时钟低振荡器频率中选择。当T0CS为 0，指令时钟会被选择当作定时器T0 时钟源。当T0CS是 1且LCK_TM0为 0，EX_CKI 会被当作定时器T0 时钟源。当T0CS是 1 且LCK_TM0 为 1 (定时器T0 来源必须设定为1)，会选择低振荡器频率 (I_LRC)输出。汇总成表格如下。(也请参下图所示)

定时器 T0 时钟源	T0CS	LCKTM0	定时器 T0 来源
Instruction clock	0	X	X
EX_CKI	1	0	X
		X	0
I_LRC	1	1	1

表 4-7 Timer0 时钟源控制概述

激活EX_CKI的边沿或低振荡器频率，寄存器T0CE (T0MD[4]) 位可以增加定时器T0。当T0CE是 1，EX_CKI或低振荡器频率上的高到低转换将增加定时器T0。在定时器T0 时钟源被使用前，如果寄存器PS0WDT (T0MD[3])

位被清零，定时器T0 时钟源可以由预分频器P0 分割。当藉由指令在PS0WDT写入 0，预分频器 P0 会被指定到定时器T0，且会在执行指令后被清除。预分频器P0 的分割率是由寄存器PS0SEL[2:0] 位 1:2 到 1:256 所决定。

进入定时器T0 之前，定时器T0 时钟源默认与指令时钟同步。如果EX_CKI或低振荡器频率被用来当作定时器T0时钟源，用户必须注意他们的频率不能超过指令时钟，否则会导致错误计数。当低振荡器频率同时被当作定时器T0 时钟源与指令时钟，用户必须指定NY8A051H预分频器P0 到定时器T0，预分频器P0 的分割率不得少于 4。配置字 (EX_CKI到指令时钟) 可以减少这样的限制。选项设定为异步，定时器T0 时钟源就不会与指令时钟同步，所以EX_CKI的输入频率可以高于指令时钟。输入EX_CKI的频率最大值是依据过程变量。

当定时器T0 溢出，寄存器T0IF (INTF[0])将会设定为 1 以标明定时器T0 发生溢出。如果寄存器T0IE (INTE[0])位与GIE都设定为 1，会发声中断的请求，并执行中断服务程序。直到硬体写入 0 到T0IF ， T0IF才会被清除。

定时器T0 的结构框图与WDT如下图:

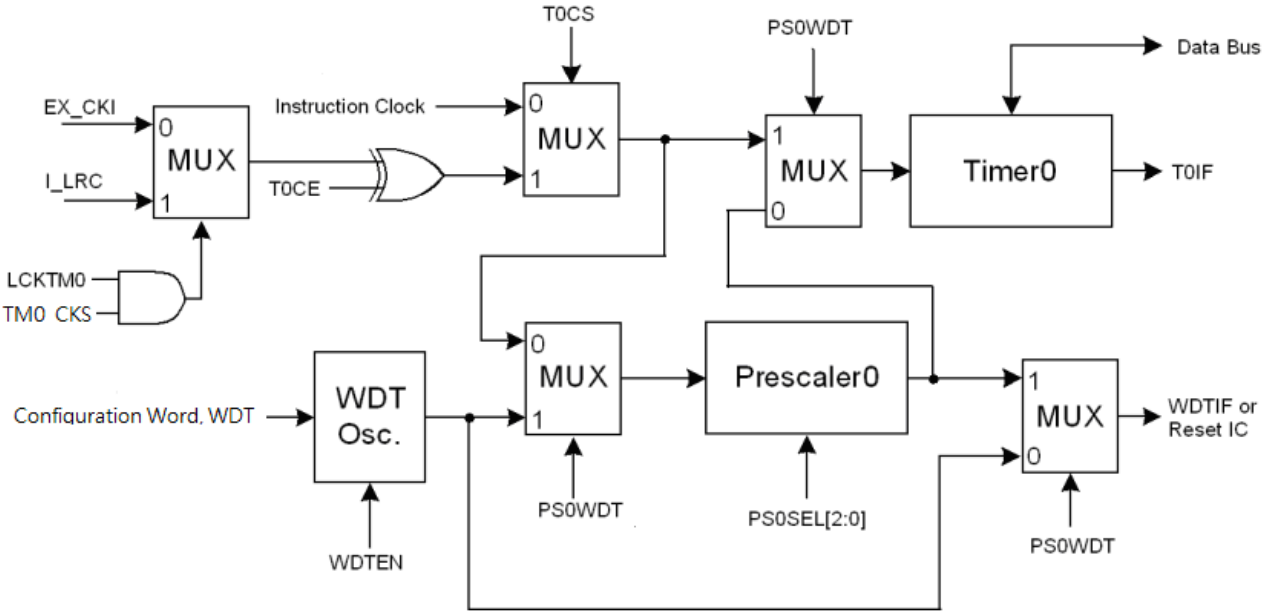


图 4-5 定时器 T0 与 WDT 结构框图

4.7 Timer1/PWM1/Buzzer1

定时器T1 是具有预分频器P1 的 8 位递减定时器，其分频率是可编程的。定时器T1 的输出可以被用于产生PWM1输出与Buzzer1 输出。写入定时器T1 时也会写入定时器T1 重载寄存器(T1rld)与定时器T1 计数器。读取定时器T1汇显示定时器T1 目前技术值的内容。

定时器T1 的结构框图如下图所示:

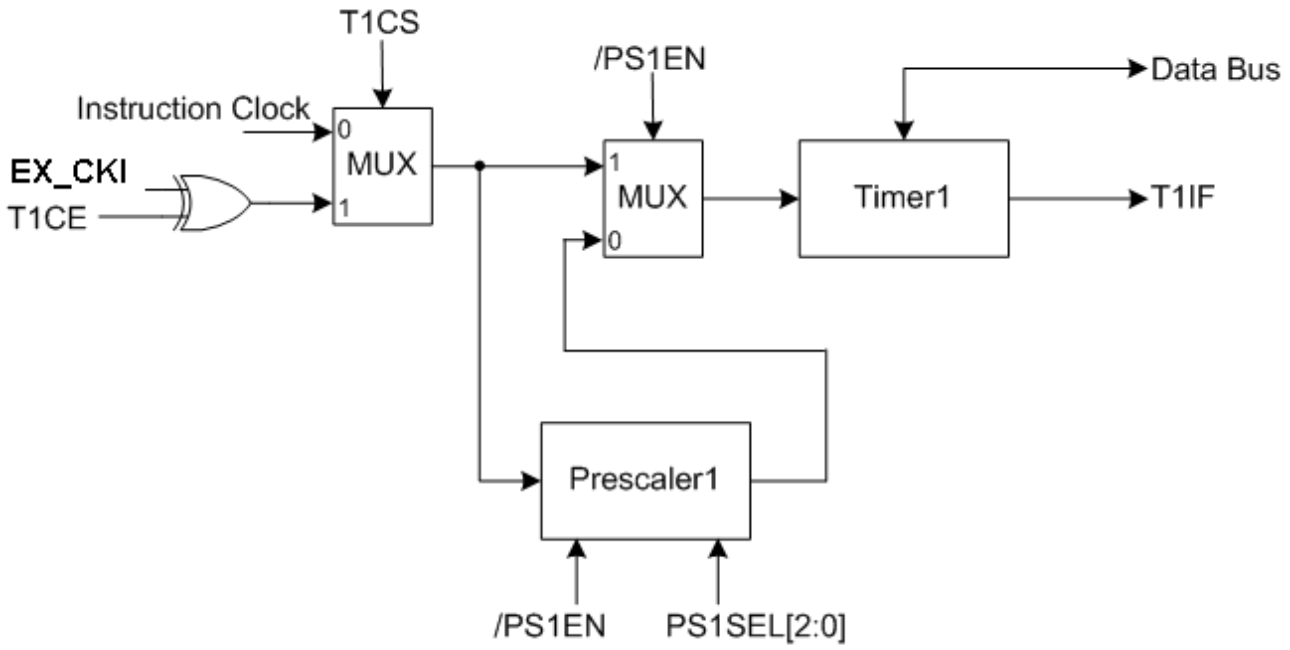


图 4-6　定时器 T1 结构框图

定时器T1 的操作方式可以藉由寄存器T1EN (T1CR1[0])位开启或关闭。定时器T1 被打开后，它的时钟源是由寄存器T1CS (T1CR2[5])位决定是指令时钟或EX_CKI脚位。当T1CS为 1，EX_CKI会被选择当做时钟源。当T1CS为 0，则是指令时钟当做时钟源。当EX_CKI被选取，减少定时器T1 有效边沿是由寄存器T1CE (T1CR2[4])位决定。当T1CE为 1，EX_CKI的高到低转换将会减少定时器T1。当T1CE为 0，EX_CKI的低到高转换将会减少定时器T1。时钟源应用到定时器T1 之前，预分频器P1 可以进一步划分所选的时钟源。预分频器P1 可藉由写入 0 到寄存器/PS1EN (T1CR2[3]位打开，寄存器PS1SEL[2:0] (T1CR2[2:0])位可以决定其分割率 1:2 到 1:256。Current value of 预分频器P1 目前数值可以藉由读取寄存器PS1CV取得。

定时器T1 提供两种操作模式：单触发模式与不停止模式。当寄存器T1OS (T1CR1[2]) 位为 1，即选择单触发。下溢发生即是定时器T1 会从储存在寄存器TMR1 的初始值递减计数到 0x00。当寄存器T1OS (T1CR1[2])为 0，表示选择不停止模式。当下溢发生，寄存器T1RL (T1CR1[1])位会决定下一个递减计数的开始。当T1RL为 1，储存在寄存器TMR1 的初始值会被重新储存并从其初始值开始下一个递减计数。当T1RL为 0，定时器T1 从 0xFF开始下一个递减计数。

当定时器T1下溢，寄存器T1IF (INTF[3])位会被设定为1并标明定时器T1发生下溢事件。如果寄存器T1IE (INTE[3])位与GIE同时设定为 1，会发生中断请求且执行中断服务程序。直到硬体写入 0 至T1IF，T1IF才会被清除。

定时器T1 时序图如下图所示：

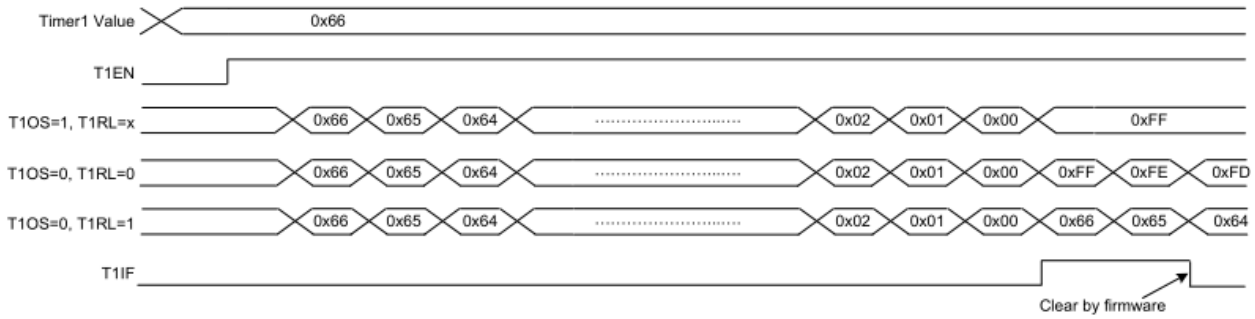


图 4-7 定时器 T1 时序图

当寄存器PWM1OEN (T1CR1[7])位被设定为 1 且配置字PB.2 是PWM，PWM1 输出可使用于I/O脚PB2。当PWM1OEN=1，PB2 会自动成为输入脚。PWM1 输出的活动状态是由寄存器PWM1OAL (T1CR1[6])位决定。 当PWM1OAL为 1，PWM1 输出是低电平；PWM1OAL为 0，PWM1 则是高电平。PWM1 的占空比与帧率都是可编程的。占空比是由寄存器PWM1DUTY决定。当PWM1DUTY为 0, PWM1 输出会无法激活。当PWM1DUTY为 0xFF，PWM1 输出将会激活 255 定时器T1 输出时钟。帧率是由TMR1 初始值所决定。因此，PWM1DUTY数值必须小于或等于TMR1。PWM1 的结构框图如下:

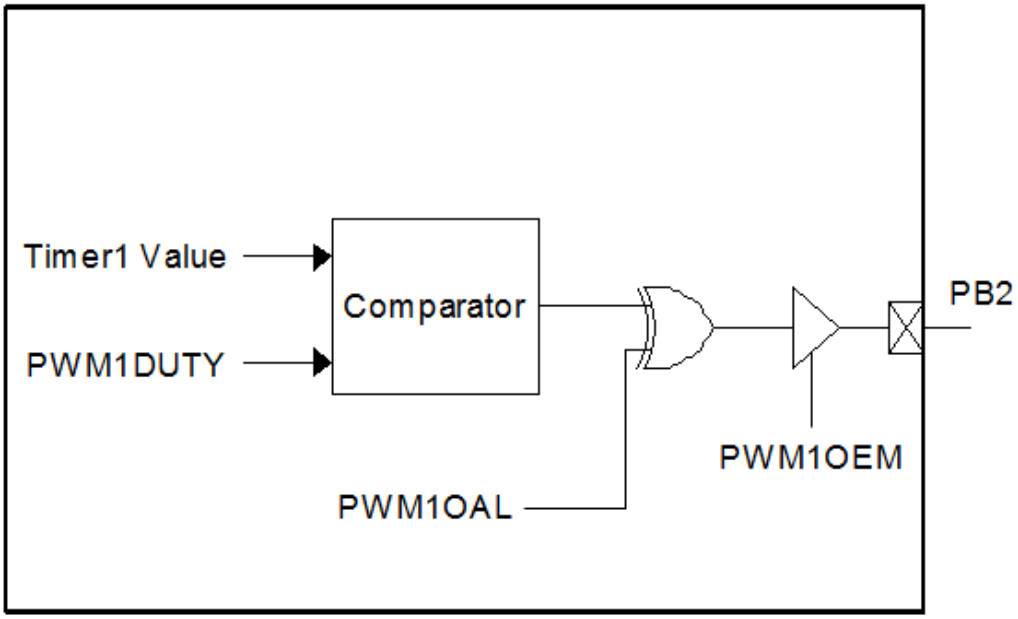


图 4-8 PWM1 结构框图

当寄存器BZ1EN (BZ1CR1[7]) 设定成 1，相对应的配置字PB.2 是蜂鸣器时，Buzzer1 输出可以在I/O 脚PB2使用。当BZ1EN设定为 1，PB2 将会自动成为输出脚。BZ1 的频率是由寄存器BZ1FSEL[3:0] (BZ1CR[3:0])位决定可以衍生自定时器T1 输出或预分频器P1 输出。当BZ1FSEL[3]为 0，预分频器P1 输出被选择为产生BZ1 输出。当BZ1FSEL[3]为 1，定时器T1 输出被选为产生BZ1 输出。为了产生所有种类的频率，分割率的范围是 1:2 到1:256。Buzzer1 结构框图如下所示:

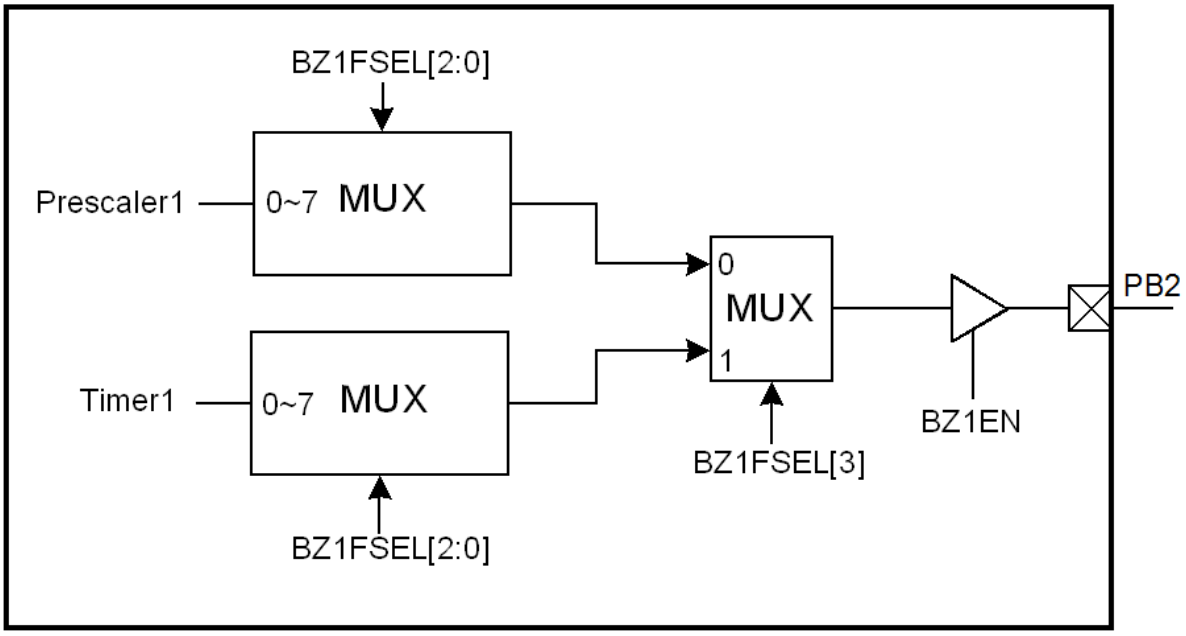


图 4-9 Buzzer1 结构框图

4.8 红外线载波

寄存器IREN (IRCR[0])被设定为 1 后，会产生红外线载波，而PB1 会自动成为输出脚。当IREN清零，PB1 将会如它被配置的一样，成为一般I/O脚。

红外线载波频率是由寄存器IRF57K (IRCR[1])位所选择。当IRF57K为 1，红外线载波频率是 57KHz；当IRF57K为0，频率是 38KHz。

红外线载波的活动状态(极性)是根据PB1 输出资料所选择。当寄存器IRCSEL (IRCR[2])位为 1 且PB1 输出数据为0，红外线载波将存在PB1。当寄存器IRCSEL (IRCR[2])位为 0 且PB1 输出数据为 1，红外线载波将存在PB1。红外线载波的极性如下图所示:

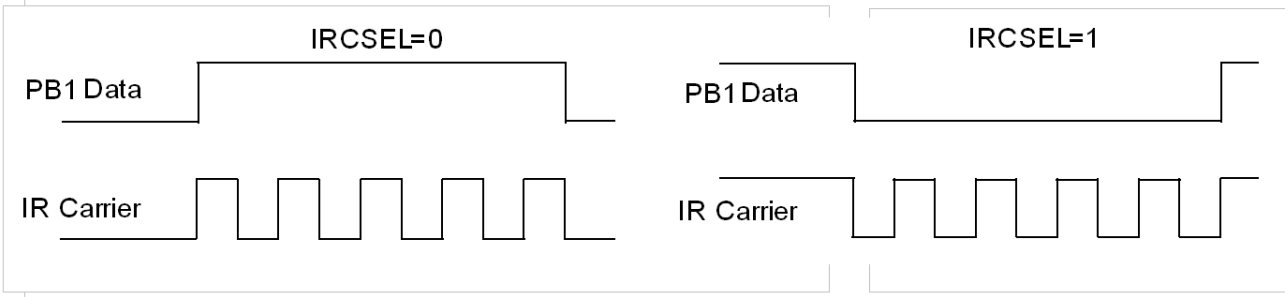


图 4-10 红外线载波的极性 vs.输出数据

4.9 低电压检测（LVD）

NY8A051H低电压检测器(LVD)内置精确的带隙基准，准确检测VDD电平。当LVDEN(寄存器PCON[5])=1 时，VDD电压值低于LVDS[3:0]选择的LVD电压，LVD输出会变低。如果LVD中断被启用，LVD中断标志将是高的，如果GIE=1，它将强制程序执行中断服务程序。此外，LVD实时状态输出可以通过寄存器PCON1[6]进行轮询。以下是LVD结构框图：

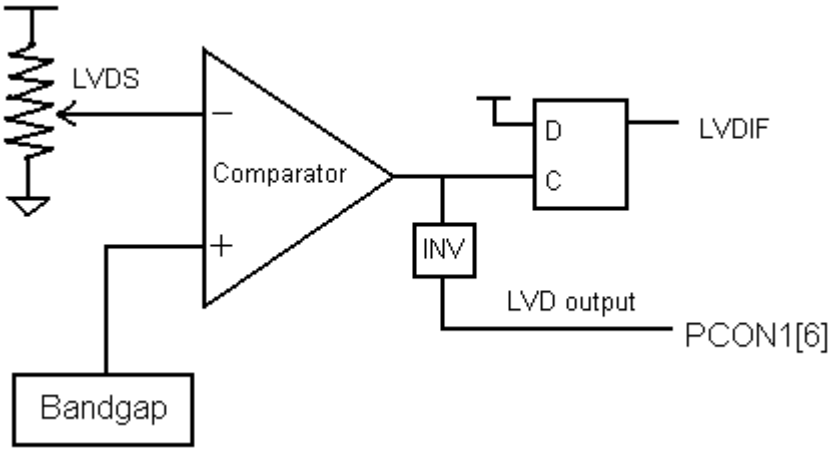


图 4-11　LVD 结构框图

LVD电压选择表如下：

LVDS[3:0]	LVDL Voltage	LVDH Voltage	说明
0000	2.00V	2.03V	
0001	2.20V	2.23V	
0010	2.38V	2.41V	
0011	2.79V	2.82V	
0100	2.88V	2.91V	
0101	2.97V	3.00V	
0110	3.27V	3.30V	
0111	3.58V	3.61V	
1000	3.90V	3.93V	
1001	1.93V	1.96V	
1010	4.04V	4.07V	
1011	2.58V	2.61V	
1100	4.13V	4.16V	
1101	3.13V	3.16V	
1110	3.45V	3.48V	

表 3-8　LVD 电压选择

4.10 电压比较器

NY8A051H提供各种模拟比较方式的电压比较器和内部参考电压。比较器的非反相和反相输入可以与GPIO共享。CMPEN(寄存器PCON[2])用于启用和禁用比较器。当CMPEN=0(默认)时，比较器被禁用。当CMPEN=1 时，比较器启用。在暂停模式下，比较器是自动禁用的。

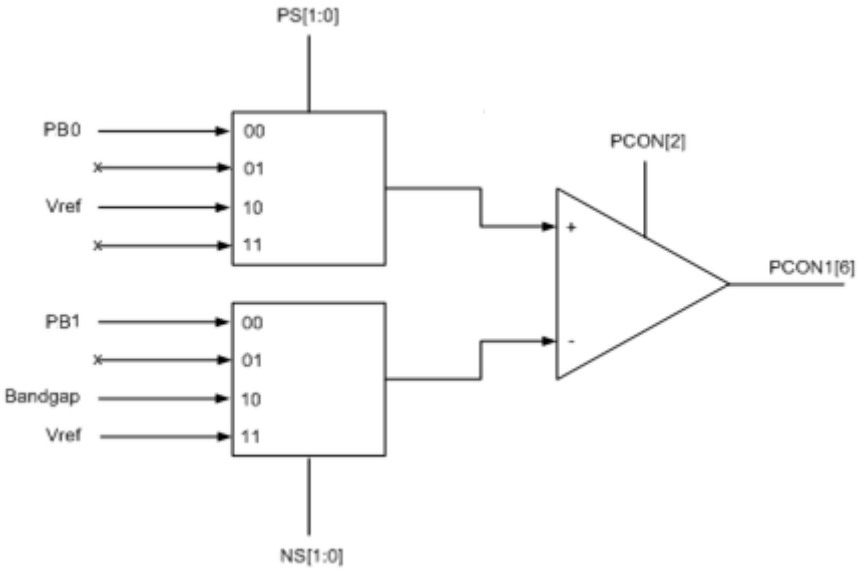


图 4-12 比较器结构图

4.10.1 比较器参考电压（Vref）

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

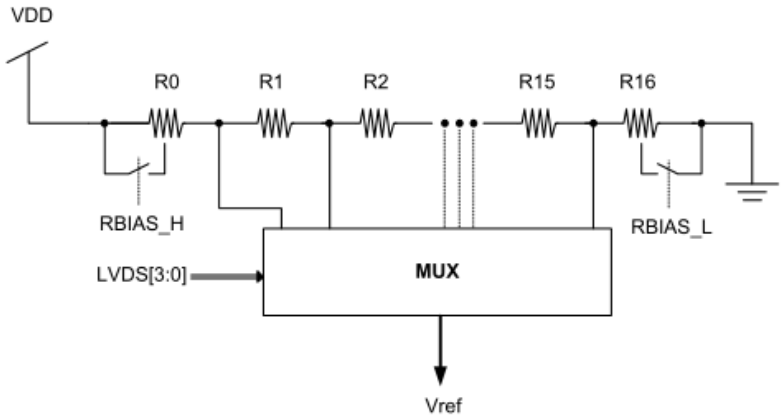


图 4-12 Vref 硬件连接

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

LVDS[3:0]	RBIAS_H=1 RBIAS_L=0	RBIAS_H=0 RBIAS_L=1
0000	65/128 VDD	33/128 VDD
0001	59/128 VDD	29/128 VDD
0010	56/128 VDD	26/128 VDD
0011	47/128 VDD	21/128 VDD
0100	45/128 VDD	20/128 VDD
0101	44/128 VDD	19/128 VDD
0110	40/128 VDD	16/128 VDD
0111	37/128 VDD	14/128 VDD
1000	34/128 VDD	12/128 VDD
1001	67/128 VDD	34/128 VDD
1010	33/128 VDD	11/128 VDD
1011	50/128 VDD	23/128 VDD
1100	32/128 VDD	10/128 VDD
1101	41/128 VDD	17/128 VDD
1110	38/128 VDD	15/128 VDD
1111	35/128 VDD	13/128 VDD

表 4-9 参考电压 Vref 选择表

注意： Vref 的误差为 ±0.1V 。

比较器的正相输入由PS[1:0]确定(寄存器CMPCR[3:2])。

表格如下所示:

PS[1:0]	正相输入
00	PB0
01	--
10	Vref
11	---

表 4-10 正相输入选择

比较器的反相输入由NS[1:0]确定(寄存器CMPCR[1:0])。

表格如下所示:

NS[1:0]	反相输入
00	PB0
01	--
10	Bandgap (0.65V)
11	Vref

表 4-11 反相输入选择

有两种方法可以获得比较器的输出结果：一种是通过寄存器轮询，另一种是通过探测输出板。

比较器输出可以通过LVDOUT(寄存器PCON1[6])进行轮询。

为了探测输出板上的比较器输出，设置CMPOE(寄存器OSCCR[6])为 1，然后PB2 将是比较器输出的实时状态。

4.11 看门狗定时器 （WDT）

NY8A051H中有一个单片的独立振荡器被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能被配置字节使能或禁止。当WDT被配置字节使能时，在程序执行过程中，其操作仍然可以通过寄存器WDTEN (PCON[7])位来控制。此外，WDT超时后的机制可以复位NY8A051H或发出由另一个配置字节决定的中断请求。同时，在WDT超时后，寄存器/TO (STATUS[4])位将被清除为 0。

WDT超时周期的基线由两个配置字决定，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，则可以延长超时周期。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，取决于WDT超时机制。如果WDT超时将复位NY8A051H，分频速率从 1：1 到 1：128，且如果WDT超时将中断NY8A051H，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 并且设置/ TO标志位为 1。

如果用户选择中断WDT超时机制，在WDT超时后，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。直到固件将 0 写入WDTIF，WDTIF才会被清除为0。

4.12 中断

NY8A051H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引起。硬件中断则有以下六种:

- 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。
- LVD中断。

GIE是全局中断使能标志，必须为 1 才能使能硬件中断功能。GIE可以通过ENI指令设置，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NY8A051H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GIVE为 1 并返回原始执行顺序。

当发生任何硬件中断时，中断标志寄存器INTF的相应位将被设置为 1。该位在固件将 0 写入该位之前不会清除为零。因此，用户可以通过轮呼寄存器INTF获得哪个事件引起硬件中断的信息。需注意只有当中断使能寄存器INTE

的相应位设置为 1 时，才会读取相应的中断标志。如果中断使能寄存器INTE的相应位设置为 1，GIE也为 1，将发生硬件中断，下一条指令将从 0x008 中读取。同时，NY8A051H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NY8A051H。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 并返回原始执行序列。

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指令将触发嵌套中断，但RETIE将在ISR跳出后清除内部中断处理，因此中断标志可能被错误清除。

4.12.1 Timer0 溢出中断

Timer0 溢出（从 0x00 到 0xFF）将设置寄存器T0IF位。如果T0IE和GIE设置为 1，则将处理此中断请求。

4.12.2 Timer1 借位中断

Timer1 借位（从 0xFF到 0x00）将设置寄存器T1IF位。如果T1IE和GIE设置为 1，则将处理此中断请求。

4.12.3 看门狗超时中断

当WDT超时且配置字选择WDT超时将产生中断请求时，它将设置寄存器WDTIF位。如果WDTIE和GIE设置为1，则将处理此中断请求。

4.12.4 PB 输入状态改变中断

当PBx（0 ≤ x ≤ 5）配置为输入引脚且相应的寄存器WUPBx位设置为 1 时，这些选定I/O引脚上的电平变化将设置寄存器PBIF位。如果PBIE和GIE设置为 1，则将处理此中断请求。需注意当PB0 同时设置为电平变化中断和外部中断时，外部中断标志EIS将禁止PB0 电平变化操作。

4.12.5 外部中断输入

根据EIS=1 和INTEDG的配置，I/O引脚PB0 上选定的有效边沿将设置寄存器位INTIF，如果INTIE和GIE设置为1，则将处理此中断请求。

4.12.6 LVD 中断

当VDD电平低于LVD电压时，LVD标志位由高变低，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。

4.13 振动配置

因为NY8A051H是双时钟IC，有高振荡（F_{HOSC}）和低振动（F_{LOSC}）可选择作为系统振荡（F_{OSC}）。可用作F_{HOSC}的振荡器是内部高RC振荡器（I_HRC）。可用作F_{LOSC} 的振荡器是内部低RC振荡器（I_LRC）。

(1) STPHOSC(OSCCR[1])=1 will stop FHOSC
(2) FHOSC will be disabled automatically at Halt mode

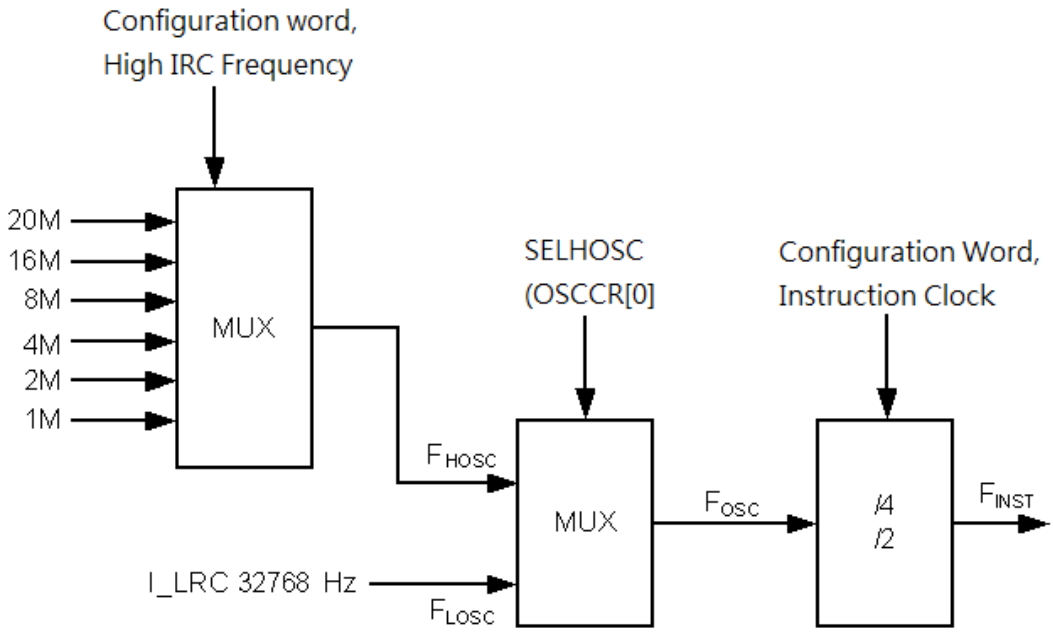


图 4-12 NY8A051H 振荡配置结构图

I_HRC输出频率由三个配置字决定，可以是 1M、2M、4M、8M、16M或 20MHz。

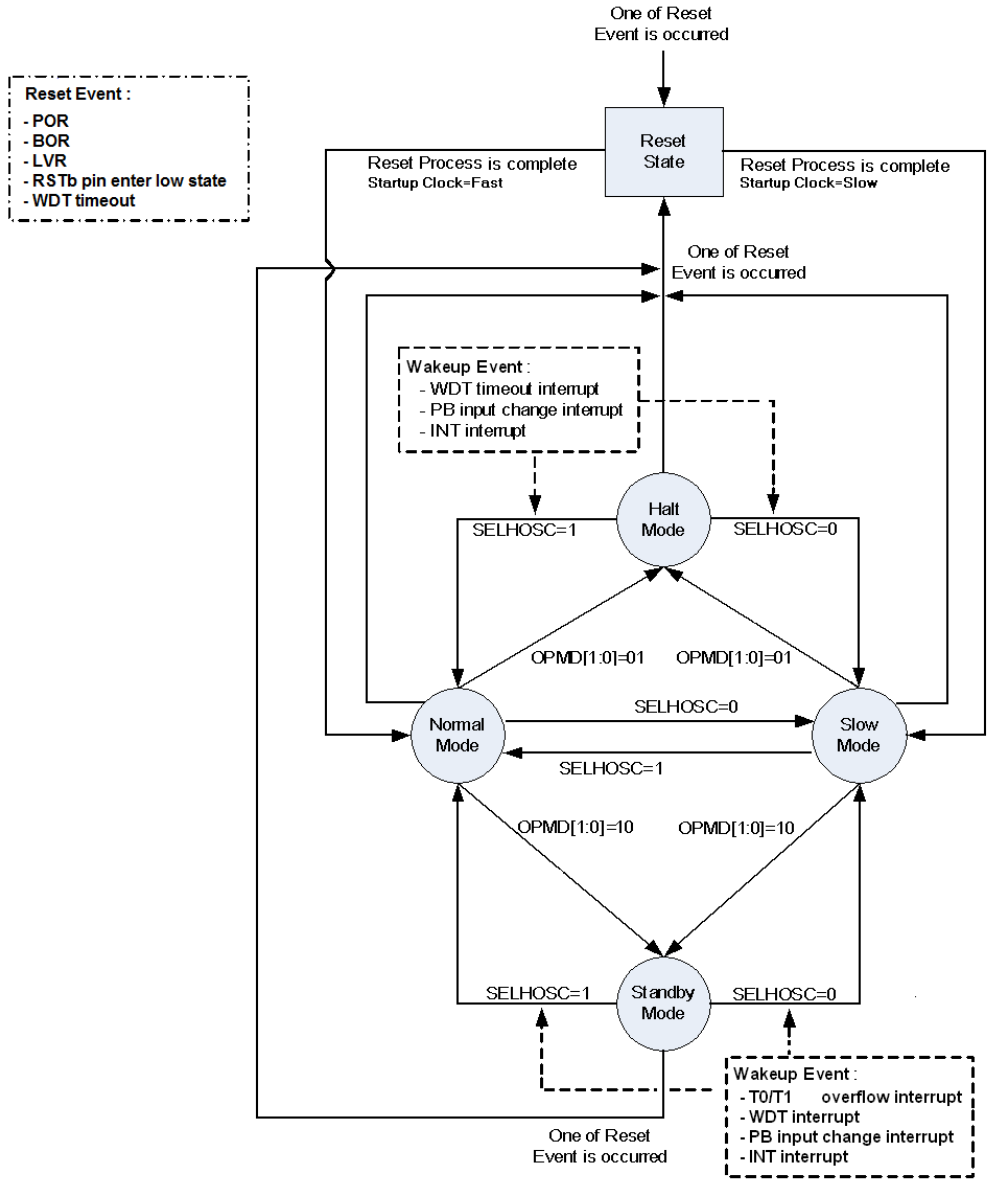
当选择I_LRC时，其频率以 32768Hz为中心。

根据寄存器SELHOSC (OSCCR [0])位的值，可以选择FHOSC或FLOSC 作为系统振荡FOSC 。当SELHOSC为 1 时，选择FHOSC作为FOSC 。当SELHOSC为 0 时，选择FLOSC 作为FOSC 。一旦确定FOSC ，根据配置字的值，指令时钟FINST可以是FOSC /2 或FOSC /4。

4.14 振动配置

NY8A051H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，NY8A051H将停止几乎所有的运作，除了定时器T0/T1/看门狗定时器以定期唤醒。在睡眠模式下，NY8A051H将睡眠直到外部事件或定时器触发电路来唤醒。

四种操作模式如下图所示。



4.14.1 正常模式

发生任何复位事件并且复位过程完成后，NY8A051H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决定。如果启动时钟为I_HRC，NY8A051H将进入正常模式，如果启动时钟为I_LRC，NY8A051H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，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NY8A051H将进入正常模式。

- 指令的执行是基于F_{HOSC}和所有外围模块可以主动根据相应的模块使能位。
- F_{LOSC} 仍处于活动状态并运行。
- IC可藉由写 0 至寄存器SELHOSC (OSCCR[0])位切换为慢速模式。
- IC可通过编程寄存器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，NY8A051H在运行正常模式可同时为低频时钟。低振荡频率连接到Timer0 时钟，这是通过设置LCKTM0 为 1 和相应的配置字Timer0 源设置为 1 来实现。

4.14.2 慢速模式

通过写 0 至寄存器SELHOSC位，NY8A051H将进入慢速模式。在低速模式下，为节省功耗，F_{LOSC}被选为系统振荡器但仍能保持IC运行。然而，F_{HOSC}将不会自动被NY8A051H关闭。因此在慢速模式下，用户可写 0 至寄存器STPHOSC (OSCCR[1])位进一步降低功耗。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}或该程序可能会继续。

- 指令执行是基于F_{LOSC} 和所有外围模块可以主动根据相应的模块使能位。
- 通过写 1 至寄存器STPHOSC位，F_{HOSC}可以被关闭。
- IC可通过编程寄存器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切换为正常模式。

4.14.3 待机模式

通过写入 10b至寄存器OPMD[1:0]位，NY8A051H将进入待机模式。 然而，在待机模式下，F_{HOSC}不会自动被NY8A051H关闭，用户必须进入低速模式并写入 1 至寄存器STPHOSC位，以停止F_{HOSC}振荡寄存器。大部分NY8A051H的外围模块会被关闭，如T0EN/T1EN位被设置为 1 则定时器仍可运作。因此Timer0/Timer1 到期后NY8A051H能唤醒，保质期是由寄存器TMR0/TMR1、F_{INST} 和用于Timer0/Timer1 的其他配置而定。

- 指令执行的停止和一些外设模块可以根据相应的模块使能位被激活。
- 藉由写入 1 至寄存器STPHOSC位F_{HOSC}可以被关闭。
- F_{LOSC}仍保持活跃并持续运作。
- 如遇以下任一状况IC便能从待机模式唤醒：

  (a)Timer0 溢出中断/Timer1 借位中断 (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SELHOSC=1，IC将回复到正常模式，如SELHOSC=0 则IC将回复到慢速模式。
- 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。

4.14.4 睡眠模式

通过执行SLEEP指令或写入 01b至寄存器OPMD[1:0]位，NY8A051H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WDT将清除但保持运作。

在睡眠模式下，所有周围模块是被关闭的，指令执行是停止的且NY8A051H只能通过一些特殊事件唤醒。因此，睡眠模式是NY8A051H所提供最省电的模式。

- 指令执行停止，所有外围模块关闭。
- FHOSC和FLOSC 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)发生INT外部中断。
- 从睡眠模式唤醒后，如SELHOSC=1，IC将回复到正常模式，如SELHOSC=0 则IC将回复到慢速模式。

  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。

4.14.5 唤醒稳定时间

睡眠模式的唤醒稳定时间为 16* FOSC，由于待机模式下FHOSC或FLOSC 仍在运行，因此无需为待机模式唤醒稳定时间。

在NY8A051H进入待机模式或睡眠模式之前，用户可以执行指令ENI。在这种情况下，NY8A051H将跳转到地址 0x008，以便在唤醒后执行中断服务程序。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。

4.14.6 操作模式概述

四种操作模式概述如下：

模式	正常模式	慢速模 式	待机模式	睡眠模式
FHOSC	使能	STPHOSC	STPHOSC	关闭
FLOSC	使能	使能	使能	关闭
指令执行	执行	执行	停止	停止
计时器 0/1	T0使能/ T1使能	T0使能/ T1使能	T0使能/T1使能	关闭
WDT	选项和WDT使能	选项和WDT使能	选项和WDT使能	选项和WDT使能
其它模块	模块使能位	模块使能位	模块使能位	关闭所有
唤醒源	-		- Timer0 上溢 - Timer1 下溢 - WDT超时 - PB输入口改变 -中断 - LVD	- WDT超时 - PB输入口改变 -中断

表 4-12 操作模式概述

4.15 复位

当以下任一复位事件发生时，NY8A051H将会进入复位状态并开始复位动作：

- 当VDD检测到上升沿时，出现上电复位。
- 当工作电压低于预设的电压时，出现低压复位。

- RSTb引脚输入低电平。
- 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。

Event	/TO	/PD
POR, BOR, LVR	1	1
RSTb复位（工作状态）	不变	不变
RSTb复位（睡眠状态）	1	0
WDT溢出复位（工作状态）	0	1
WDT溢出复位（睡眠状态）	0	0
执行SLEEP指令	1	0
执行CLRWDT指令	1	1

表 4-13　/TO 和/PD 值和相关事件概述

复位事件发生后，NY8A051H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。这个周期被称为升高复位时间，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，NY8A051H将进一步等待 Fosc 的 16 个时钟周期 (OST，振荡器启动时间)，复位过程完成。

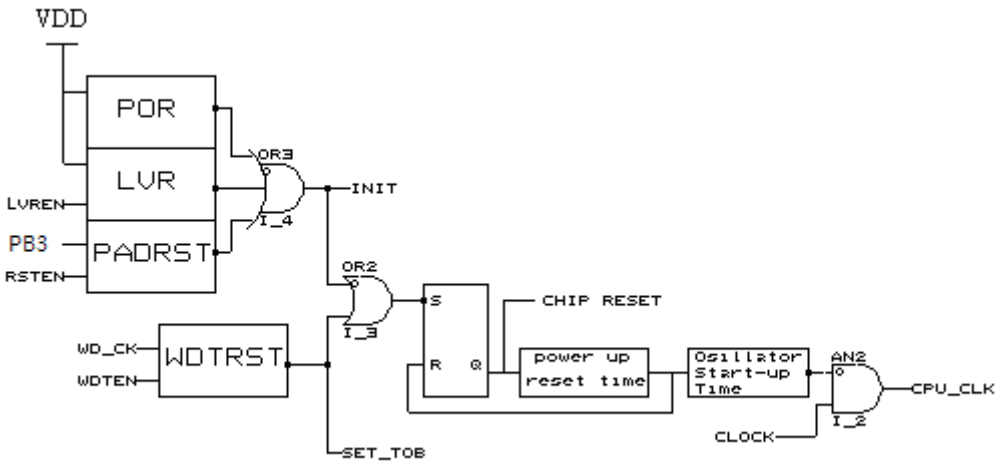


图 4-13　芯片复位电路框图

为使VDD缓慢升高，建议使用RSTb复位，如下图。

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

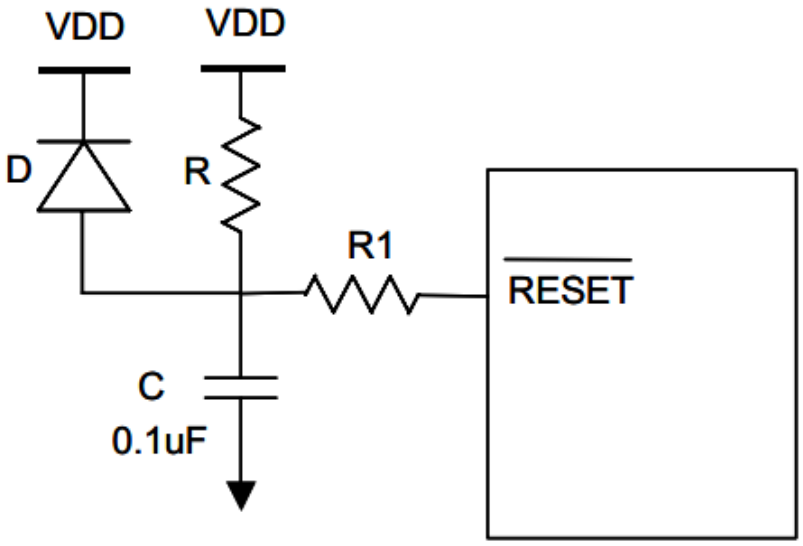


图 4-14 外部上电复位应用

5. 指令设置

NY8A051H为各种应用程序提供了 55 个强大的指令。

指令	操作符		操作	周期	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	bit	Rotate left R	1	C
BSR	R	bit	Set bit in R	1	-
BSR	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	操作符		操作	周期	标志
	1	2			
算术指令					
ADDAR	R	d	dest = ACC & R	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	d	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 5-1 指令设置

ACC：累加器。

adr：即时寻址。

bit：一个 8 位寄存器R的位寻址。

C：进位/借位。

C=1，加法指令发生进位，减法指令不发生借位。

C=0，加法指令不发生进位，减法指令发生借位。

d ： 目标

若d=“0”，结果存入ACC。

若d=“1”，结果存入R寄存器。

DC：数字进位标记。

dest：目标。

F：F 页面特殊功能寄存器，F 值为 0x5~0xF。

i：8 位即时数据。

PC：程序计数器。

PCHBUF：程序计数器的高位缓冲器。

/PD ：电源中断标志位。

/PD=1，电源升高或CLRWDT指令执行后。

/PD=0，SLEEP指令执行后。

Prescaler：预分频器 0 分频率。

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 ~ 0xF。

T0MD：T0MD寄存器。

TBHP：ROM高位目标地址。

TBHD：ROM高位目标地址的存储数据。

/TO：计时器溢出标志位。

/TO=1，电压上升或执行 CLRWDT 或 SLEEP 指令后。

/TO=0，WDT 溢出。

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 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 ≤ 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 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 ≤ 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 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 ≤ 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 执行指令前： ACC=0x12, i=0x34, C=1. 执行指令后： ACC=0x46, C=0.

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

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“与”运算
周期	1
举例	ANDIA i 执行指令前： ACC=0x5A, i=0xAF. 执行指令后： ACC=0x0A, Z=0.

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 执行指令前： R=0x5A, B2=0x3. 执行指令后： R=0x52.

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
周期	1
举例	BSR R,B2 执行指令前： R=0x5A, B2=0x2. 执行指令后： R=0x5E.

BTRSC	Test Bit in R and Skip if Clear
语法	BTRSC R, bit
操作数	0 ≤ R ≤ 63 0 ≤ bit ≤ 7
操作	Skip next instruction, if R[bit] = 0.
状态影响	--
说明	位测试指令，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。

BTRSS	Test Bit in R and Skip if Set
语法	BTRSS R, bit
操作数	0 ≤ R ≤ 63 0 ≤ bit ≤ 7
操作	Skip next instruction, if R[bit] = 1.
状态影响	--
说明	位测试指令，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。

CALL	Call Subroutine
语法	CALL adr
操作数	0 ≤ 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 执行指令前： PC=A0. Stack pointer=1 执行指令后： PC=address of SUB, Stack[1] = A0+1, Stack pointer=2.

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 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.

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.

CLRWDT	Clear Watch-Dog Timer
语法	CLRWDT
操作数	--
操作	00h → WDT, 00h → WDT预分频器（若开启） 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 执行指令前： /TO=0 执行指令后： /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 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 ≤ 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 执行指令前： R=0xA6, d=1, Z=0. 执行指令后： 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ACC-R，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.

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递减，若d="0"，结果存入ACC；若d="1"，结果存入R。
周期	1
举例	DECR R, d 执行指令前： R=0x01, d=1, Z=0. 执行指令后： R=0x00, Z=1.

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	ADDAR R,d DAA 执行指令前： ACC=0x28, R=0x25, d=0. 执行指令后： ACC=0x53, C=0.

DECRSZ	Decrease R, Skip if 0
语法	DECRSZ R, d
操作数	0 ≤ R ≤ 63 d = 0, 1.
操作	R - 1 → dest, Skip if result = 0
状态影响	--
说明	R 先递减，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，因此这条指令要执行两个周期。
周期	1 or 2（跳过）
举例	DECRSZ R, d 指令 2 指令 3 执行指令前： R=0x1, d=1, Z=0. 执行指令后： R=0x0, Z=1，操作结果为 0，指令 2 被跳过。

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，关闭全局中断
周期	1
举例	DISI 执行指令前： GIE=1. 执行指令后： GIE=0.

ENI	Enable Interrupt Globally
语法	ENI
操作数	--
操作	Enable Interrupt, 1 → GIE
状态影响	--
说明	GIE设置为 1，开启全局中断
周期	1
举例	ENI 执行指令前： GIE=0. 执行指令后： GIE=1.

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位立即地址l装入PC<8:0>，PCHBUF[1]装入PC[9]
周期	2
举例	GOTO Level 执行指令前： PC=A0. 执行指令后： PC=address of Level.

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值装入PC PC<7:0>；TBHP[1:0] 值装入PC<9:8>
周期	2
举例	GOTOA 执行指令前： PC=A0. TBHP=0x02, ACC=0x34. 执行指令后： PC=0x234.

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递增，若d="0"，结果存入ACC； 若d="1"，结果存入R
周期	1
举例	INCR R, d 执行指令前： R=0xFF, d=1, Z=0. 执行指令后： R=0x00, Z=1.

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[9:0]。
周期	3
举例	INT 执行指令前： PC=address of INT code. 执行指令后： 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先递增，若d="0"，结果存入ACC； 若d="1"，结果存入R。若结果为"0"则跳过下一条指令，执行NOP指令
周期	1 or 2(skip)
举例	INCRSZ R, d 指令 2 指令 3 执行指令前： R=0xFF, d=1, Z=0. 执行指令后： R=0x00, Z=1，因结果为 0，程序跳过指令 2

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做或运算，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 ≤ i < 255
操作	ACC | i → ACC
状态影响	Z
说明	ACC和 8 位立即数做或运算，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.

IOSTR	Move F-page SFR to ACC
语法	IOSTR F
操作数	0 ≤ F ≤ 15
操作	F-page SFR → ACC
状态影响	--
说明	将F页面特殊寄存器的F赋值给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.

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页面特殊寄存器的F
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.

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 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.

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位立即数装入PC[9:0]
周期	2
举例	LGOTO Level 执行指令前： PC=A0. 执行指令后： 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 执行指令前： i=0x55, ACC=0xAA. 执行指令后： 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
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.

MOVR	Move 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赋值给d，若d="0"，结果存入ACC；若d="1"，结果存入R。指令执行后，通过状态标杆位Z检查R是否为 0。
周期	1
举例	MOVR R, d 执行指令前： R=0x0, ACC=0xAA, Z=0, d=0. 执行指令后： R=0x0, ACC=0x00, Z=1.

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

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 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.

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
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.

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 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 ≤ R ≤ 63 d = 0, 1.
操作	R[7] → C, R[6:0] → dest[7:1], C → dest[0]

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 ≤ R ≤ 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 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 ≤ R ≤ 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) 执行指令前： R=0x05, ACC=0x06, d=1, C=0. 执行指令后： R=0xFE, C=0. (-2) (b) 执行指令前： R=0x05, ACC=0x06, d=1, C=1. 执行指令后： R=0xFF, C=0. (-1) (c) 执行指令前： R=0x06, ACC=0x05, d=1, C=0. 执行指令后： R=0x00, C=1. (-0), Z=1. (d) 执行指令前： R=0x06, ACC=0x05, d=1, C=1. 执行指令后： R=0x1, C=1. (+1)

SBCIA	Subtract ACC and Carry from Immediate
语法	SBCIA i
操作数	0 ≤ 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) 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)

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页面特殊函数寄存器
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.

SFUN	Load S-page SFR from ACC
语法	SFUN S
操作数	0 ≤ S ≤15
操作	ACC → S-page SFR
状态影响	--
说明	写S页面特殊函数寄存器
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.

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0分频器清零。/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 ≤ 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	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 ≤ i < 255
操作	i – ACC → ACC
状态影响	Z, DC, C
说明	常数减ACC，结果存入ACC
周期	1
举例	SUBIA i (a) 执行指令前： i=0x05, ACC=0x06. 执行指令后： ACC=0xFF, C=0. (-1) (b) 执行指令前： i=0x06, ACC=0x05, d=1. 执行指令后： ACC=0x01, C=1. (+1)

SWAPR	Swap High/Low Nibble in R
语法	SWAPR R, d
操作数	0 ≤ 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 执行指令前： R=0xA5, d=1. 执行指令后： 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TBH，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.

T0MD	Load ACC to T0MD
语法	T0MD
操作数	--
操作	ACC → T0MD
状态影响	--
说明	写T0 模式寄存器
周期	1
举例	T0MD 执行指令前： T0MD=0x55, ACC=0xAA. 执行指令后： T0MD=0xAA.

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“异或”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.

T0MDR	Move T0MD to ACC
语法	T0MDR
操作数	--
操作	T0MD → ACC
状态影响	--
说明	读T0 模式寄存器
周期	1
举例	T0MDR 执行指令前： T0MD=0x55, ACC=0xAA. 执行指令后： 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和常数做“异或”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.

6. 配置表

项目	名称	选项
1	High IRC Frequency (IRC 高频率)	1. 1MHz 2. 2MHz 3. 4MHz 4. 8MHz 5. 16MHz 6. 20MHz
2	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)
3	WDT (看门狗定时器)	1. Watchdog Enable (Software control) (看门狗使能（软件控制）) 2. Watchdog Disable (Always disable) (看门狗关闭（永远关闭）)
4	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)
5	Timer0 source (定时源)	1. EX_CKI 2. I_LRC
6	PB.2	1. PB.2 is I/O 2. PB.2 is PWM 3. PB.2 is Buzzer
7	PB.3	1. PB.3 is I/O 2. PB.3 is reset
8	PB.4	1. PB.4 is I/O 2. PB.4 is instruction clock output
9	Startup Time (启动时间)	1. 140us 2. 4.5ms 3. 18ms 4. 72ms 5. 288ms
10	WDT Time Base (看门狗定时器时基)	1. 3.5ms 2. 15ms 3. 60ms 4. 250ms
11	LVR Setting (LVR 设定)	1．Register Control(寄存器控制) 2. Register Control + Halt mode Off(寄存器控制+睡眠模式关闭) 3. Always On (一直) 4. Operation mode On + Halt mode Off(工作模式开启+睡眠模式关闭)
12	LVR Voltage (复位电压)	1. 1.6V 2. 1.8V 3. 2.0V 4. 2.2V 5. 2.4V 6. 2.7V 7. 3.0V 8. 3.3V 9. 3.6V 10. 4.2V
13	VDD Voltage (VDD 电压)	1. 3.0V 2. 4.5V 3. 5.0V
14	Read Output Data (读取输出数据)	1. I/O port(I/O 端口) 2. Register(寄存器)
15	EX_CKI to Inst. Clock	1. Sync(同步) 2. Async (不同步)
16	Startup Clock (时钟源)	1. I_HRC(高速振荡) 2. I_LRC (低速振荡)
17	Input High Voltage (VIH) (输入高电压（VIH）)	1. 0.7V_{DD} 2. 0.5 V_{DD}
18	Input Low Voltage (VIL) (输入低电压（VIL）)	1. 0.3 V_{DD} 2. 0.2 V_{DD}
19	Input Voltage Schmitt Trigger (输入电压施密特触发)	1. Enable (depend on 17, 18) 2. Disable (17, 18 no use)
20	Sink / Drive Current (灌/拉电流)	1. Small 2. Normal

7. 电器特性

7.1 MC802 最大绝对值

符号	参数	额定值	单位
V_{DD} - VSS	电源电压	-0.5 ~ +6.0	V
VIN	输入电压	VSS -0.3V ~ V_{DD} +0.3	V
TOP	运行温度	-40 ~ +85	°C
TST	储存温度	-40 ~ +125	°C

7.2 直流电气特性

7.2.1 NY8A051H 电气特性参数

(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	VDD	最小值	典型值	最大值	单位	条件
VDD	工作电压		3.0		5.5	V	F_{INST} =20MHz @ I_HRC/2
			2.2				F_{INST} =20MHz @ I_HRC/4
			2.7				F_{INST} =16MHz @ I_HRC/2
			2.0				F_{INST} =16MHz @ I_HRC/4
			2.0				F_{INST} =8MHz @ I_LRC/4 & I_LRC/2
			1.6				F_{INST} =4MHz @ I_LRC/4 & I_LRC/2
			1.6				F_{INST} =32KHz @ I_LRC/4 & I_LRC/2
V_{IH}	输入高电压	5V	4.0	--	--	V	RSTb (0.8VDD)
		3V	2.4	--	--		
		5V	3.5	--	--	V	All other I/O pins, EX_CKI, INT (0.7VDD)
		3V	2.1	--	--		
		5V	2.5	--	--	V	All other I/O pins, EX_CKI, INT TTL (0.5VDD)
		3V	1.5	--	--		
		5V	--	2.5	--	V	All other I/O pins, EX_CKI, INT No Schmitt Trigger (0.5VDD)
		3V	--	1.5	--		
V_{IL}	输入低电压	5V	--	--	1.0	V	RSTb (0.2VDD)
		3V	--	--	0.6		
		5V	--	--	1.5	V	All other I/O pins, EX_CKI, INT (0.3VDD)
		3V	--	--	0.9		
		5V	--	--	1.0	V	All other I/O pins, EX_CKI, INT TTL (0.2VDD)
		3V	--	--	0.6		
		5V	--	2.5	--	V	All other I/O pins, EX_CKI, INT No Schmitt Trigger (0.5VDD)
		3V	--	1.5	--		
I_{OH}	高输出电流 (小电流)	5V	--	2.2	--	mA	V_{OH} =4.0V
		3V	--	1.2	--		V_{OH} =2.0V
	高输出电流 (正常电流)	5V	--	2.0	--	mA	V_{OH} =4.0V
		3V	--	12	--		V_{OH} =2.0V

I_{OL}	低输出电流(小电流)	5V	--	7.8	--	mA	V_{OL} =1.0V
		3V	--	4.5	--		
	低输出电流(正常电流)	5V	--	42	--	mA	V_{OL} =1.0V
		3V	--	26	--		
I_{IR}	红外吸收电流	5V	--	45	--	mA	V_{OL} =1.0V，Normal IR
		3V	--	26	--		
I_{OP}	工作电流	正常模式					
		5V	--	1.7	--	mA	F_{HOSC} =20MHz @ I_HRC/2
		3V	--	0.7	--		
		5V	--	1.4	--	mA	F_{HOSC} =20MHz @ I_HRC/4
		3V	--	0.5	--		
		5V	--	1.5	--	mA	F_{HOSC} =16MHz @ I_HRC/2
		3V	--	0.6	--		
		5V	--	1.3	--	mA	F_{HOSC} =16MHz @ I_HRC/4
		3V	--	0.5	--		
		5V	--	1.2	--	mA	F_{HOSC} =8MHz @ I_HRC/2
		3V	--	0.4	--		
		5V	--	0.9	--	mA	F_{HOSC} =8MHz @ I_HRC/4
		3V	--	0.3	--		
		5V	--	0.8	--	mA	F_{HOSC} =4MHz @ I_HRC/2
		3V	--	0.3	--		
		5V	--	0.7	--	mA	F_{HOSC} =4MHz @ I_HRC/4
		3V	--	0.2	--		
		5V	--	0.5	--	mA	F_{HOSC} =1MHz @ I_HRC/2
		3V	--	0.2	--		
		5V	--	0.5	--	mA	F_{HOSC} =1MHz @ I_HRC/4
		3V	--	0.2	--		
		慢速模式					
		5V	--	9.9	--	uA	F_{HOSC} disabled, F_{LOSC} =32KHz @ I_LRC/2
		3V	--	4.9	--		
		5V	--	6.6	--	uA	F_{HOSC} disabled, F_{LOSC} =32KHz @ I_LRC/4
		3V	--	3.6	--		
I_{STB}	待机电流	5V	--	3.5	--	uA	待机模式，F_{HOSC} disabled, F_{LOSC} =32KHz @ I_LRC/4
		3V	--	2.5	--		
I_{HALT}	睡眠电流	5V	--	--	0.5	uA	睡眠模式，禁止WDT
		3V	--	--	0.2		
		5V	--	--	5.0	uA	睡眠模式，启用WDT
		3V	--	--	3.0		
R_{PH}	上拉电阻	5V	--	65	--	kΩ	上拉电阻（不包括PB3）
		3V	--	120	--		
		5V	--	85	--	kΩ	上拉电阻（PB3
		3V	--	85	--		
R_{PL}	下拉电阻	5V	--	55	--	kΩ	下拉电阻
		3V	--	105	--		

7.2.2 I2_TOUCH 电气特性参数

电气参数 T_A = 25℃

特性	符号	条件		最小值	典型值	最大值	单位
工作电压	VDD			2.6		5.5	V
电流损耗	Idd	VDD=5.0V	正常模式		0.73		mA
		VDD=3V			0.51		mA
		VDD=5.0V	睡眠模式		13		uA
		VDD=3V			10		uA
上电初始化时间	Tini				200		ms
感应管脚电容范围	Cin					2.5*Cdc[1]	
OUT[0:1]输出电阻	Zo	delta Cin > 0.2pF			50		Ohm
		delta Cin < 0.2pF			10K		
OUT[0:1]输出灌电流	Isk	VDD=5V				10.0	mA
最小检测电容	delta_Cin	CDC=15pf			0.2		pF

注：[1] 如果感应管脚寄生电容超过2.5倍的Cdc电容，芯片不能正常工作（绝大多数情况无需考虑这个限制）

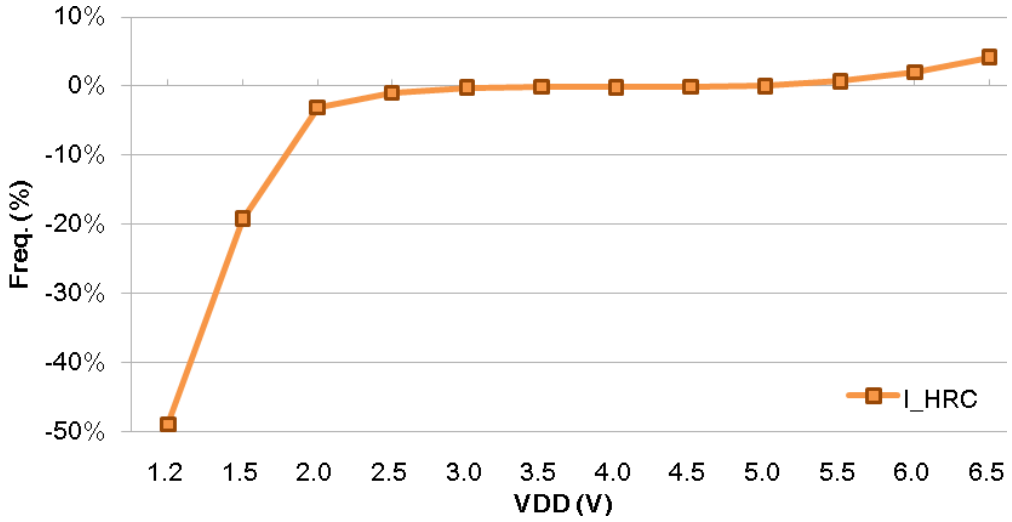
7.3 OSC 特性

(测量条件VDD，T_A 温度等于程序条件)

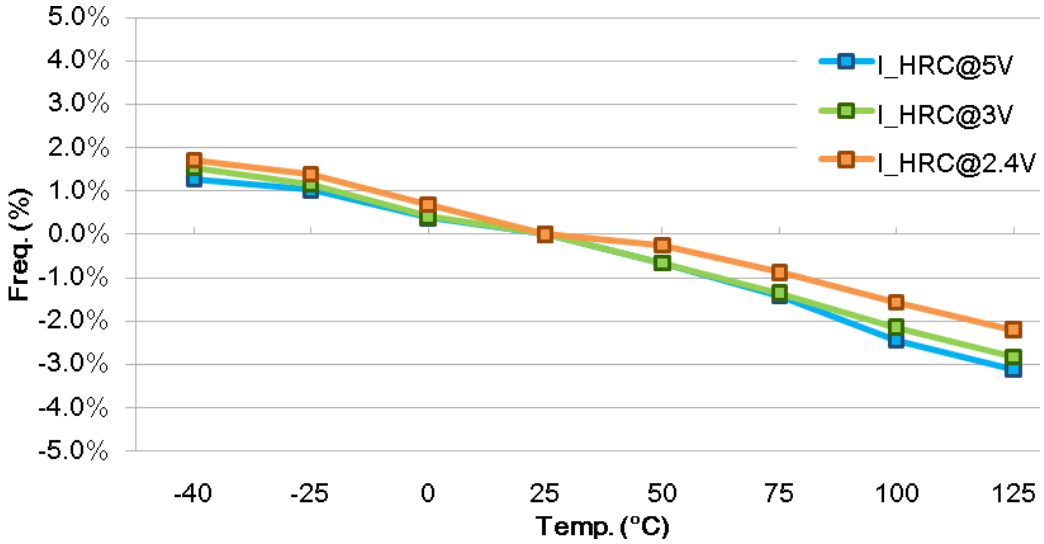
参数	最小值	典型是	最大值	单位	条件
烧录座的I_HRC偏差			±1	%	烧录座直接安装在烧录器上。
烧录机台的I_HRC偏差			±3	%	正确设置烧录机台的条件。
烧录机台的I_LRC偏差			±8	%	

7.4 特性图

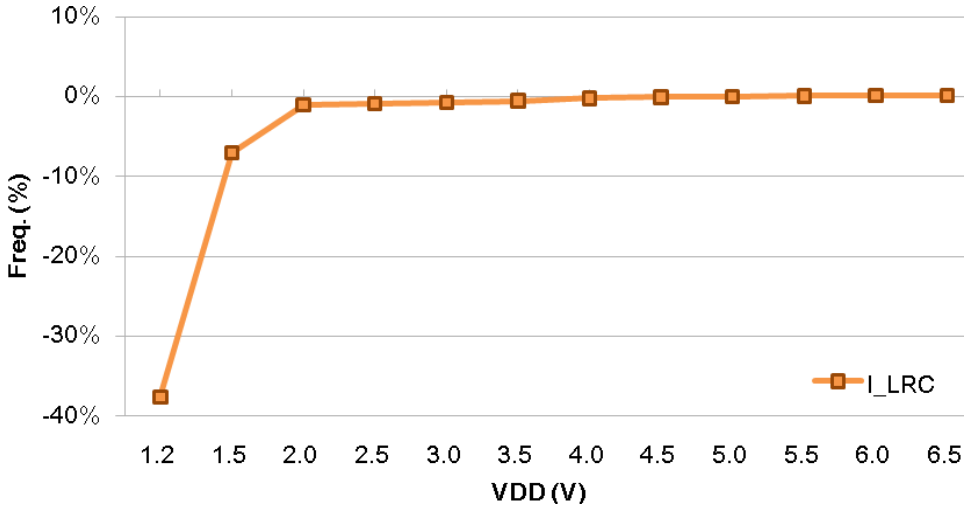
7.4.1 频率与高速振荡电压



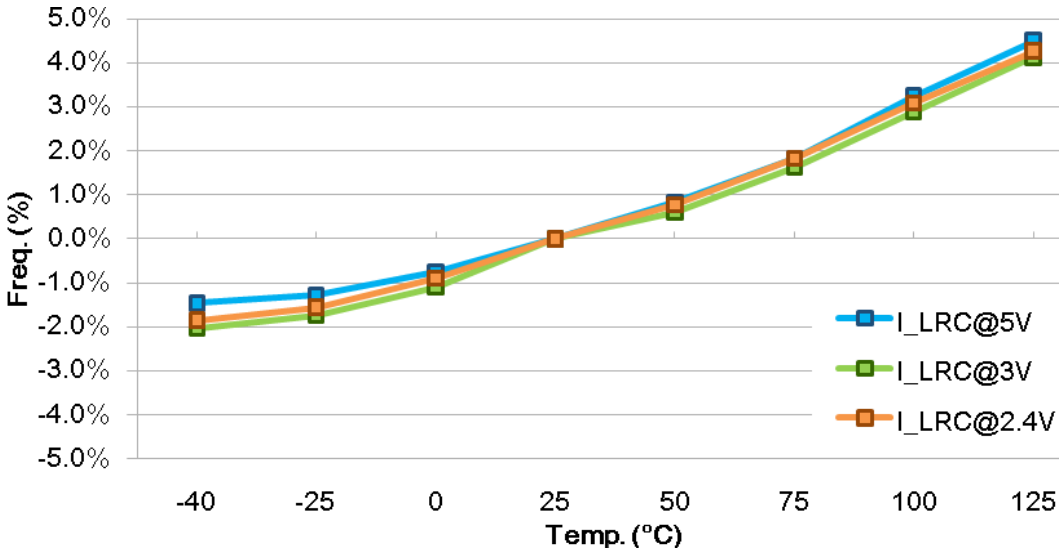
7.4.2 频率与高速振荡温度



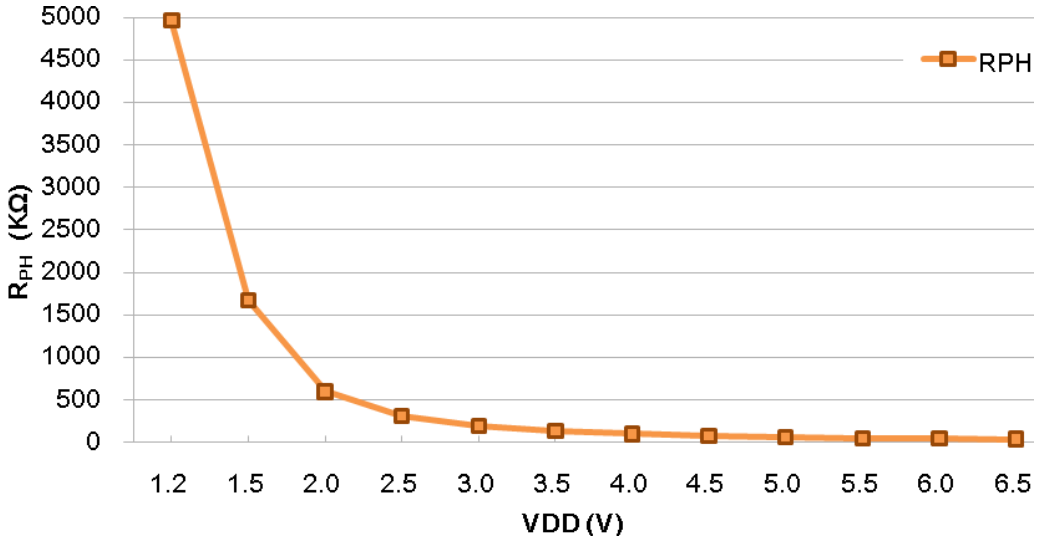
7.4.3 频率与低速振荡电压



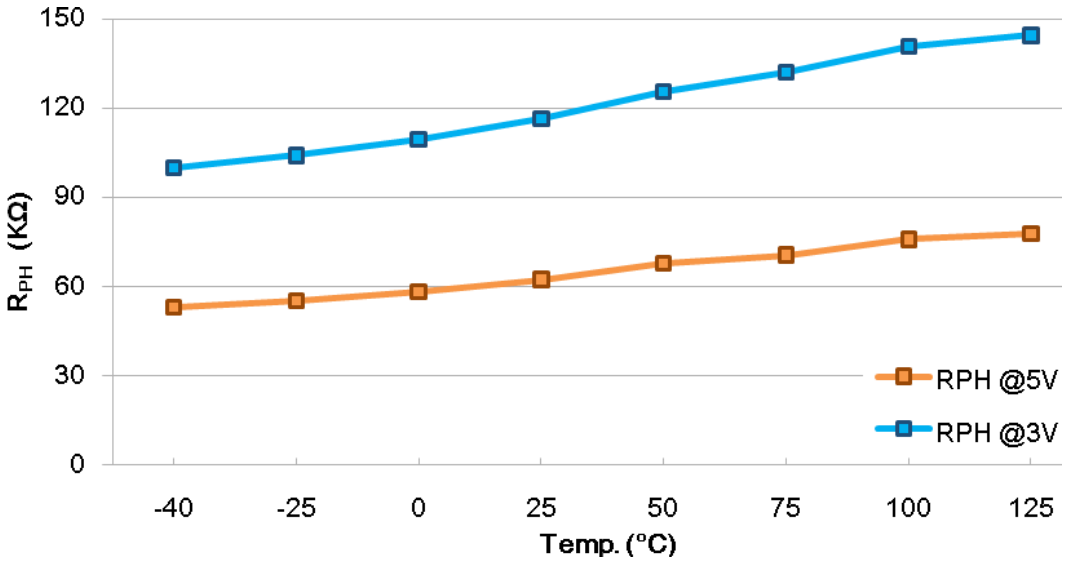
7.4.4 频率与低速振荡温度



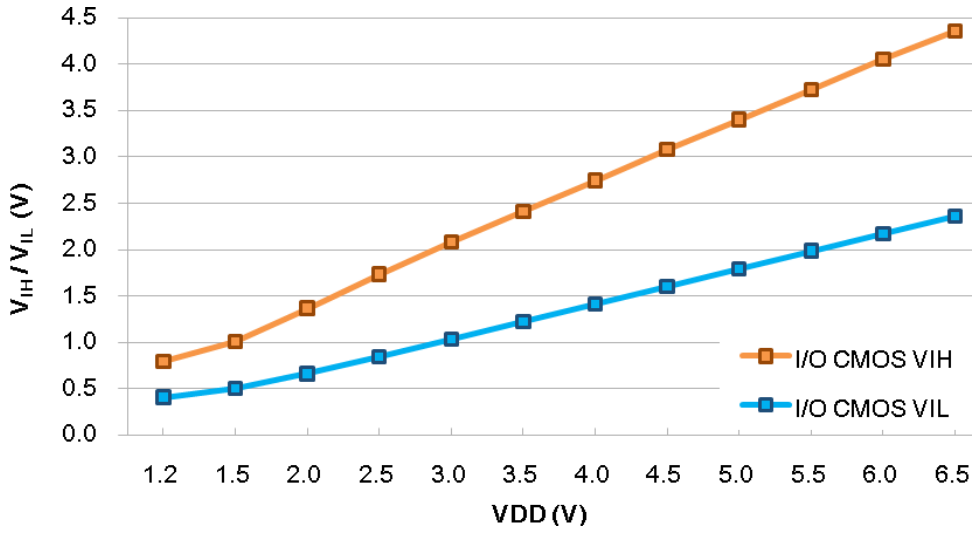
7.4.5 上拉电阻与 与 VDD

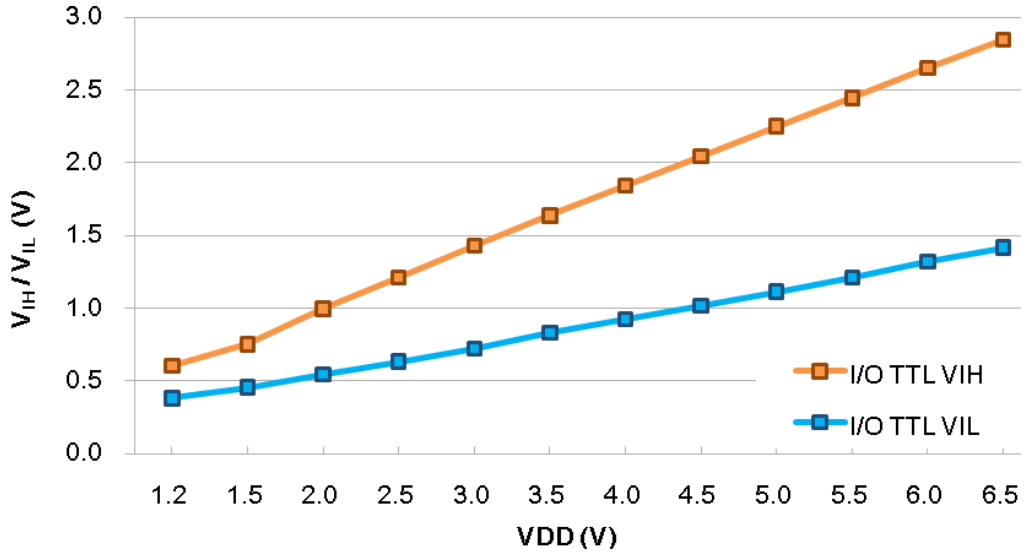


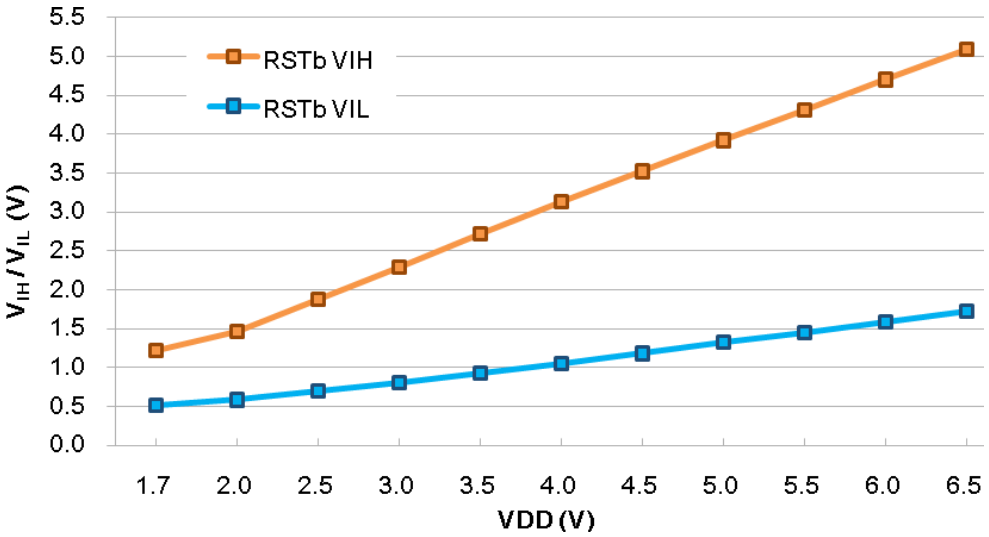
7.4.6 上拉电阻与温度



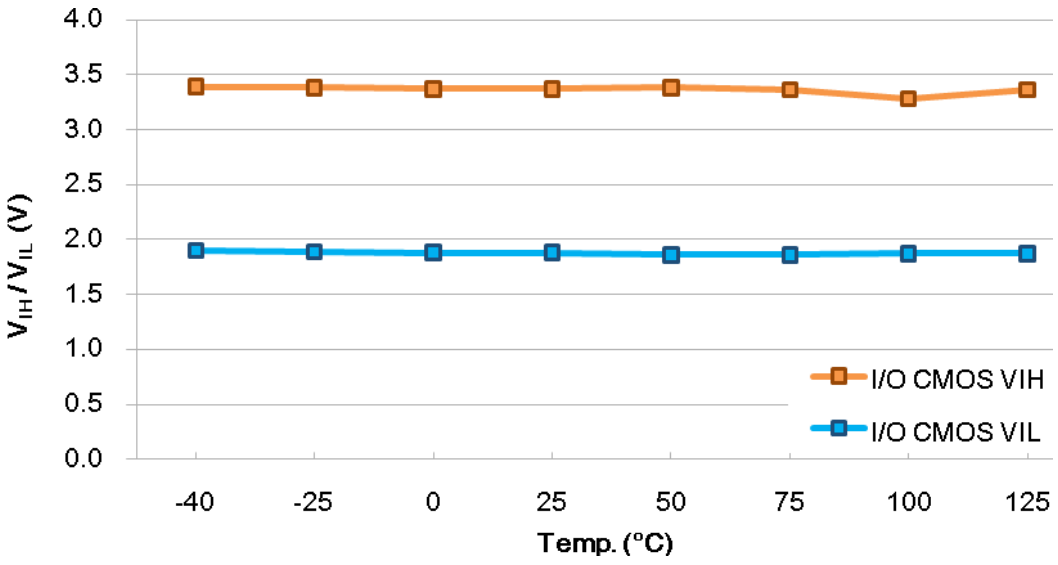
7.4.7 VIH/VIL 与 VDD

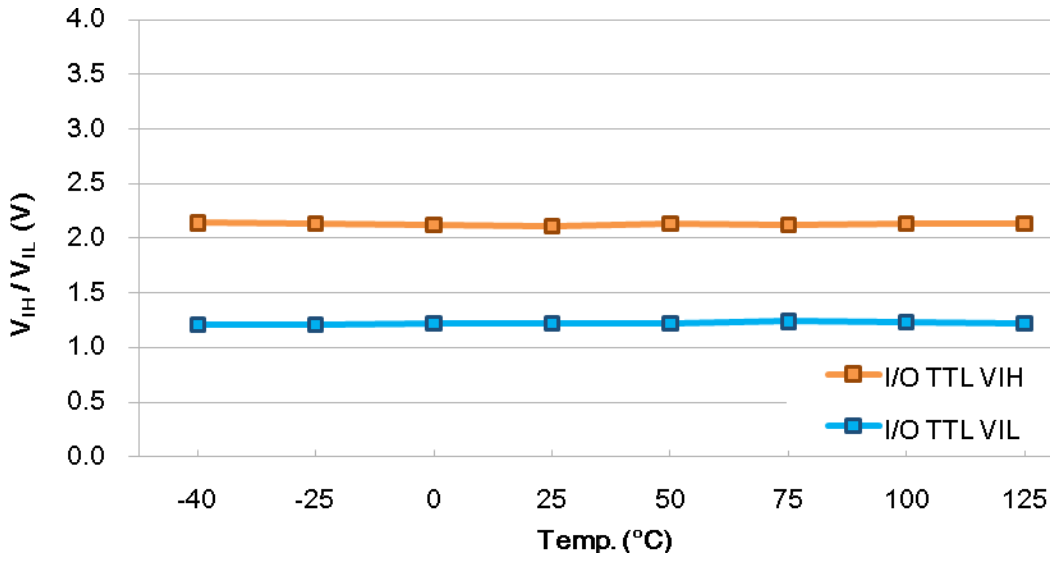






7.4.8 VIH/VIL 与温度



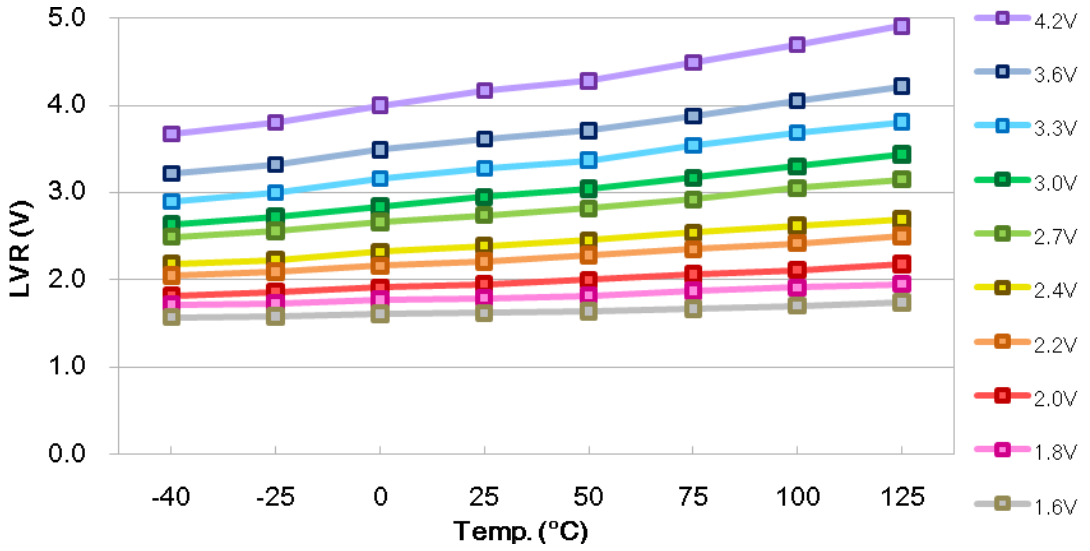


7.5 建议工作电压

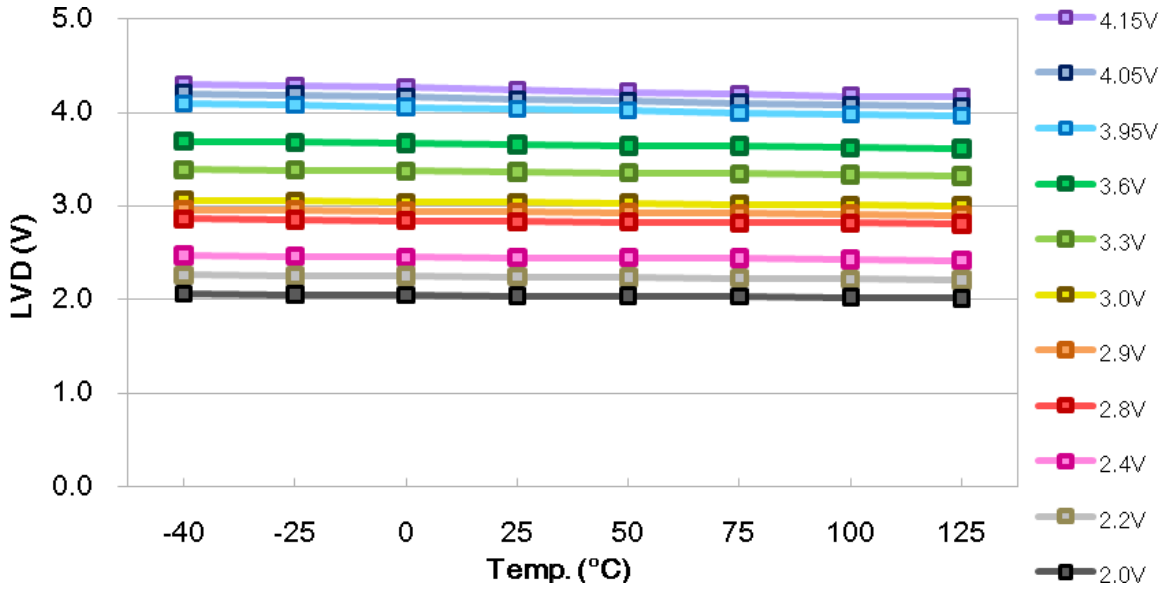
建议工作电压（温度范围：-40 °C ~ +85 °C

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	3.3V	3.6V
16M/2T	2.7V	5.5V	2.7V	3.0V
20M/4T	2.2V	5.5V	2.2V	2.4V
16M/4T	2.0V	5.5V	2.0V	2.2V
8M/2T	2.0V	5.5V	2.0V	2.2V
≦4M(2T or 4T)	1.6V	5.5V	1.6V	1.8V

7.6 LVR 与温度



7.7 LVD 与温度



8. 应用参考

8.1 应用电路

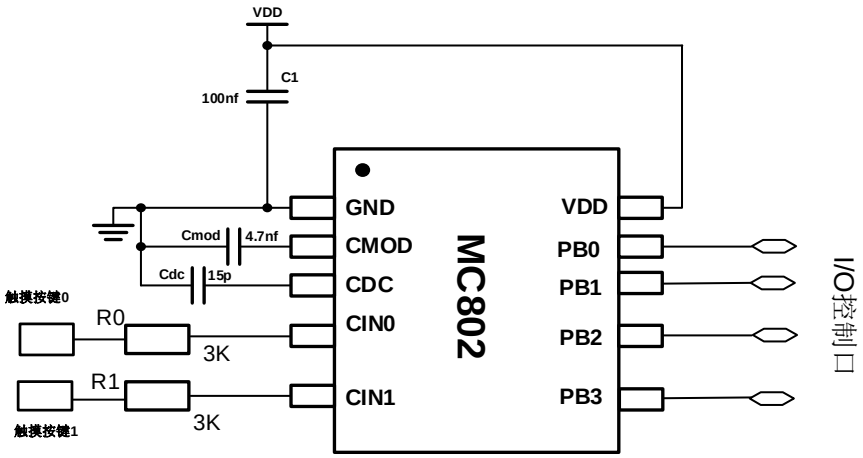


图7-1 ： 应用电路

注:

Cmod是电荷收集电容，通常取值范围在1nf~10nf，典型值是4.7nf。

Cdc是灵敏度电容，取值范围是最小5pf，最大100pf，电容取值越小，灵敏度越高。

9. 封装尺寸（MSOP10）

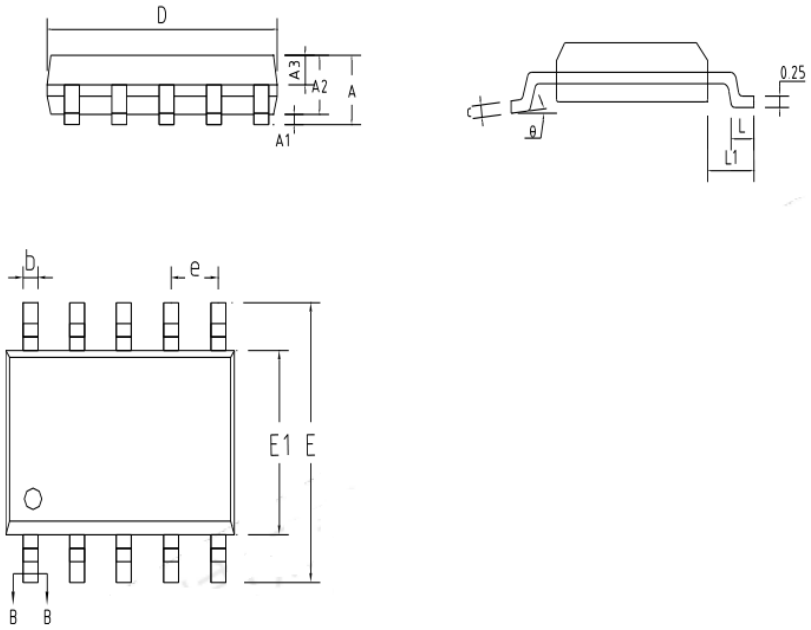


图 9-1： MSOP10封装示例

符号	毫米单位(mm)			英寸单位		
	最小	典型	最大	最小	典型	最大
A	--	--	1.10	--	--	0.043
A1	0.05	--	0.15	0.002	--	0.006
A2	0.75	0.85	0.95	0.03	0.034	0.037
b	0.18	--	0.26	0.007	--	0.01
A3	0.30	0.35	0.40	0.012	0.014	0.015
D	2.90	3.0	3.10	0.114	0.118	0.122
E	4.70	4.90	5.10	0.185	0.193	0.200
E1	2.90	3.0	3.10	0.114	0.118	0.122
e	0.50BSC			0.0197 BSC		
c	0.15	--	0.19	0.006	--	0.008
L	0.40	--	0.70	0.016	--	0.028
L1	0.95REF			0.037 REF		
θ	0°	--	8°	0°	--	8°