九齐科技股份有限公司

Nyquest Technology Co., Ltd.

数据手册

NY8A054D

14 I/O + 5-通道 PWM 8 位 EPROM-Based 单片机

Version 1.2

Jul. 24, 2018

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

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

改 版 记 录

版本	日期	内 容 描 述	修正页
1.0	2017/08/30	新发布。	-
1.1	2017/11/15	1. 修改 PA/PB 输入状态改变中断描述。 2. 修改引脚图，封装尺寸，订购信息。 3. 修改“表 24 /TO 和/PD 值和相关事件概述”。	8, 63, 64, 67, 68, 69, 70 9, 94, 95 70
1.2	2018/07/24	1. 修改图 2 程序存储器对映地址。 2. 修改PCON1 初始值。 3. 修改电压比较器说明。 4. 更新RLR指令说明。 5. 更新直流电器特性。	11 28 60 84 89

目　录

1. 概述 7

1.1 功能 7

1.2 系统框图 9

1.3 引脚图 9

1.4 引脚说明 10

2. 内存结构 11

2.1 程序存储器 11

2.2 数据存储器 12

3. 功能概述 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） 18

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

3.1.15 AWUCON（PortA唤醒控制寄存器） 22

3.1.16 INTEDG（外部中断控制寄存器） 22

3.1.17 TMRH（定时器 1 高字节寄存器） 22

3.1.18 RFC（电阻频率转换控制寄存器） 23

3.1.19 TM34RH（定时器 3 高字节寄存器） 23

3.1.20 INTE2（第 2 中断使能寄存器） 24

3.2 T0MD定时器 0 控制寄存器 24

3.3 F-page特殊功能寄存器.....25
3.3.1 IOSTA（PortA I/O控制寄存器）..... 25
3.3.2 IOSTB（PortB I/O控制寄存器）..... 25
3.3.3 APHCON（PortA上拉电阻控制寄存器）..... 26
3.3.4 PS0CV（预分频器 0 寄存器）..... 26
3.3.5 BODCON（PortB开漏控制寄存器）..... 26
3.3.6 CMPCR（比较器控制寄存器）..... 26
3.3.7 PCON1（Power控制寄存器 1）..... 28
3.4 S-page特殊功能寄存器 28
3.4.1 TMR1（定时器 1 寄存器）..... 28
3.4.2 T1CR1（定时器 1 控制寄存器 1）..... 29
3.4.3 T1CR2（定时器 1 控制寄存器 2）..... 29
3.4.4 PWM1DUTY（PWM1 占空比寄存器）..... 30
3.4.5 PS1CV（预分频器 1 寄存器）..... 30
3.4.6 BZ1CR（蜂鸣器 1 控制寄存器）..... 31
3.4.7 IRCR（IR控制寄存器）..... 31
3.4.8 TBHP（表格指针高字节寄存器）..... 32
3.4.9 TBHD（表格数据高字节寄存器）..... 32
3.4.10 P2CR1（PWM2 控制寄存器 1）..... 33
3.4.11 PWM2DUTY（PWM2 占空比寄存器）..... 33
3.4.12 OSCCR（振荡器控制寄存器）..... 34
3.4.13 TMR3（定时器 3 寄存器）..... 35
3.4.14 T3CR1（定时器 3 控制寄存器 1）..... 35
3.4.15 T3CR2（定时器 3 控制寄存器 2）..... 36
3.4.16 PWM3DUTY（PWM3 占空比寄存器）..... 37
3.4.17 PS3CV（预分频器 3 寄存器）..... 37
3.4.18 P4CR1（PWM4 控制寄存器 1）..... 37
3.4.19 PWM4DUTY（PWM4 占空比寄存器）..... 38
3.4.20 P5CR1（PWM5 控制寄存器 1）..... 38
3.4.21 PWM5DUTY（PWM5 占空比寄存器）..... 39
3.4.22 PWM5RH（PWM5 占空比高字节寄存器）..... 39
3.5 I/O Port 39
3.5.1 IO引脚结构框图..... 41
3.6 定时器 0.....51
3.7 定时器 1/PWM1/Buzzer1.....52

3.8 PWM2 54
3.9 定时器 3/PWM3 55
3.10 PWM4 57
3.11 PWM5 58
3.12 电阻/频率转换器模式（RFC） 58
3.13 IR Carrier 59
3.14 低电压侦测（LVD） 59
3.15 电压比较器 60
3.16 看门狗定时器（WDT） 63
3.17 中断 63
3.17.1 Timer0 上溢中断 64
3.17.2 Timer1 下溢中断 64
3.17.3 Timer3 下溢中断 64
3.17.4 看门狗超时中断 64
3.17.5 PA/PB输入状态改变中断 64
3.17.6 外部中断 0 输入 64
3.17.7 外部中断 1 输入 64
3.17.8 低电压侦测中断 65
3.17.9 比较器输出翻转中断 65
3.18 振荡器配置 65
3.19 工作模式 67
3.19.1 正常模式 68
3.19.2 慢速模式 68
3.19.3 待机模式 68
3.19.4 睡眠模式 69
3.19.5 唤醒稳定时间 69
3.19.6 工作模式概述 69
3.20 复位 70
4. 指令设置 72
5. 配置字节表 88
6. 电气特性 89
6.1 最大绝对值 89
6.2 直流电气特性 89

6.3 比较器/LVD电气特性 91
6.4 特性图....91
6.4.1 高速RC振荡频率与电源电压曲线图.... 91
6.4.2 高速RC振荡频率与温度曲线图.... 91
6.4.3 低速RC振荡频率与电源电压曲线图.... 92
6.4.4 低速RC振荡频率与温度曲线图.... 92
6.5 建议工作电压 92
6.6 LVR电压与温度曲线图....93
7. 芯片脚位坐标图 93
8. 封装尺寸 94
8.1 8 引脚SOP（150 毫寸） 94
8.2 16 引脚SOP（150 毫寸） 94
9. 订购信息 95

1. 概述

NY8A054D是以EPROM作为存储器的 8 位单片机，专为多组PWM的应用而设计。例如灯控，遥控车应用。采用CMOS制程并同时提供客户低成本、高性能、及高性价比等。NY8A054D核心建立在RISC精简指令集架构可以很容易地做编程和控制，共有 55 条指令。除了少数指令需要两个指令时钟，大多数指令都是一个指令时钟能完成，可以让用户轻松地以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，NY8A054D 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 脚位都能通过控制相关的寄存器达成如上拉或下拉电阻或开漏（Open-Drain）输出。此外针对红外线摇控的产品方面，NY8A054D 内置了可选择频率的红外载波发射口。

NY8A054D 有三组定时器，可用系统时钟当作一般的计时应用或者从外部讯号触发来计数。另外 NY8A054D 提供 5 组 10 位的 PWM 输出，1 组蜂鸣器输出，可用来驱动马达、LED、或蜂鸣器等等。

NY8A054D 采用双时钟机制，高速振荡时钟或者低速振荡时钟都由内部 RC 振荡或外部晶振输入。在双时钟机制下，NY8A054D 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。

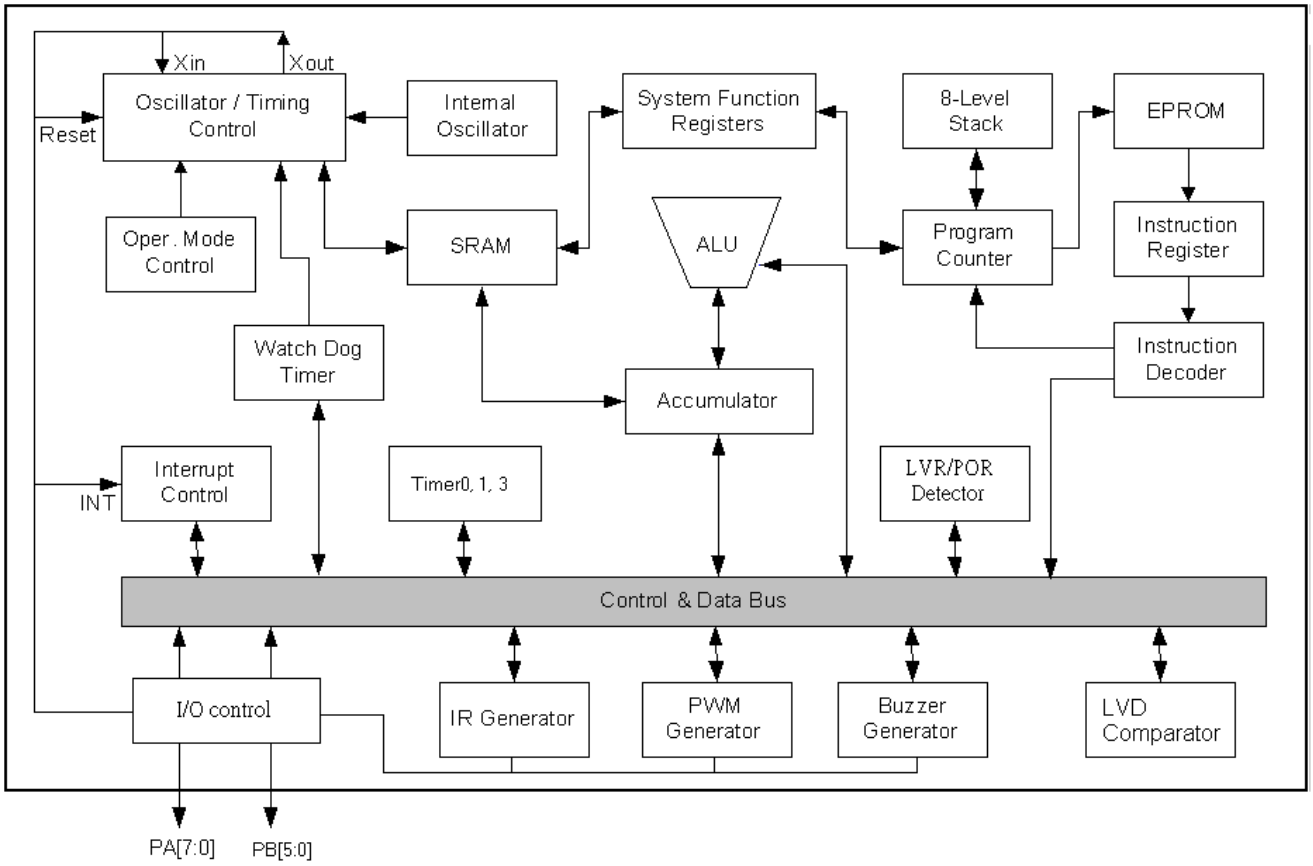
在省电的模式下，如待机模式（Standby mode）与睡眠模式（Halt mode）中，有多个中断源可以触发来唤醒 NY8A054D 进入正常操作模式（Normal mode）或慢速模式（Slow mode）来处理突发事件。

1.1 功能

- 宽广的工作电压：（指令时钟為 4 個CPU时钟，亦即 4T模式）
  - 2.0V ~ 5.5V @系統时钟 ≦8MHz。
  - 2.2V ~ 5.5V @系統时钟 >8MHz。
- 宽广的工作温度：-40°C ~ 85°C。
- 2Kx14 位的程序存储器空间。
- 128 字节的通用数据存储器空间。
- 14 根可分别单独控制输入输出方向的I/O脚（GPIO）、PA[7:0]、PB[5:0]。
- PA[5,3:0]及PB[3:0]可选择输入时使用内部下拉电阻。
- PA[7:6,4:0]及PB[5:0]可选择输入时使用内部上拉电阻。
- PB[5:0]可选择内部上拉电阻或开漏输出（Open-Drain）。
- PA[5]可选择当作输入或开漏输出（Open-Drain）。
- 8 级深度硬件堆栈（Stack）。
- 访问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。
- 二组 10 位下数定时器（Timer1,3）可选自动重载或连续下数计时。
- 五个 10 位的脉冲宽度调变输出（PWM1、2、3、4、5），PWM1/2 共享Timer1；PWM3/4/5 共享Timer3。

- 一个蜂鸣器输出（BZ1）。
- 38/57KHz红外线载波（IR）频率可供选择，同时载波之极性也可以通过寄存器选择。
- 内置准确的低电压侦测电路（LVD）。
- 内置准确的电压比较器（Voltage Comparator）。
- 内置上电复位电路（POR）。
- 内置低压复位功能（LVR）。
- 内置看门狗计时（WDT），可由配置字节（Configuration Word）控制开关。
- 内置电阻/频率转换器（RFC）功能。
- 双时钟机制，系统时钟可以随时切换高速振荡或者低速振荡。
  - 高速振荡时钟: E_HXT（超过 6MHz外部高速晶振）
    E_XT（455K~6MHz外部晶振）
    I_HRC（1~20MHz内部高速RC振荡）
  - 低速振荡时钟： E_LXT（32KHz外部低速晶振）
    I_LRC（内部 32KHz低速RC振荡）
- 四种工作模式可随系统需求调整电流消耗：正常模式（Normal mode）、慢速模式（Slow mode）、待机模式（Standby mode）与睡眠模式（Halt mode）。
- 九种硬件中断：
  - Timer0 上溢中断。
  - Timer1 下溢中断。
  - Timer3 下溢中断。
  - WDT中断。
  - PA/PB输入状态改变中断。
  - 两组外部中断。
  - 低电压侦测中断。
  - 比较器输出翻转中断。
- NY8A054D在待机模式（Standby mode）下的九种唤醒中断：
  - Timer0 上溢中断。
  - Timer1 下溢中断。
  - Timer3 下溢中断。
  - WDT中断。
  - PA/PB输入状态改变中断。
  - 两组外部中断。
  - 低电压侦测中断。
  - 比较器输出翻转中断。
- NY8A054D在睡眠模式（Halt mode）下的四种唤醒中断：
  - WDT中断。
  - PA/PB输入状态改变中断。
  - 两组外部中断。

1.2 系统框图



1.3 引脚图

NY8A054D提供二种封装类型：SOP8 及SOP16。

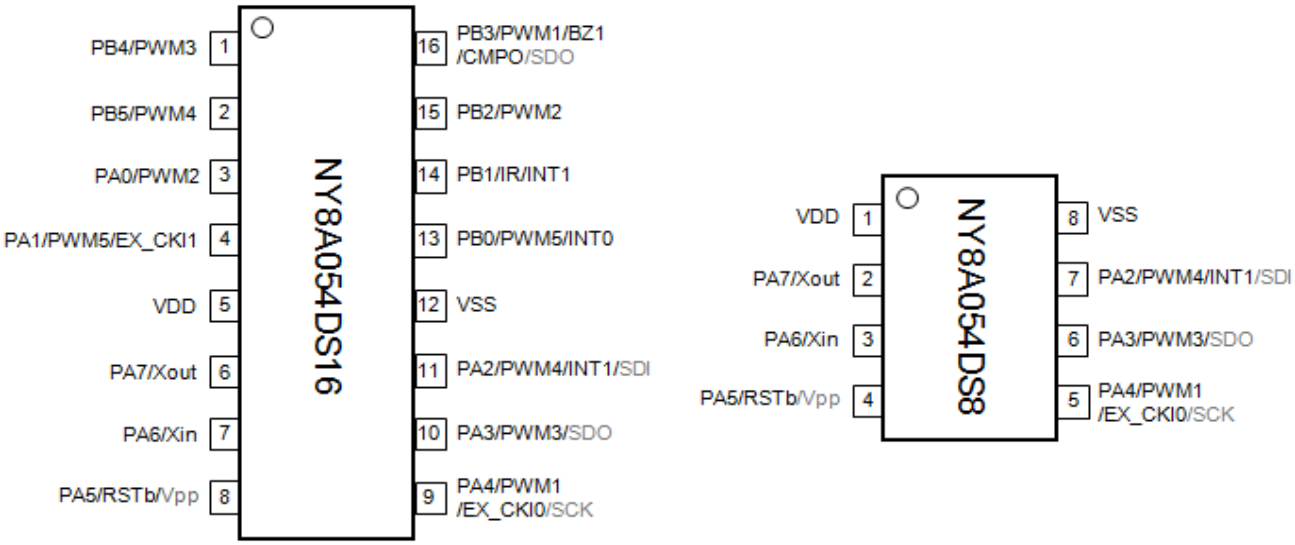


图 1　封装引脚图

1.4 引脚说明

引脚名	I/O	描述
PA0/ PWM2	I/O	PA0 是一个双向I/O引脚，也可当作比较器输入引脚。 PA0 可输出PWM2。
PA1/ PWM5/ EX_CKI1	I/O	PA1 是一个双向I/O引脚，也可当作比较器输入引脚。 PA1 可输出PWM5。 PA1 可当作定时器 3 外部时钟来源EX_CKI1。
PA2/ PWM4/ INT1/ SDI	I/O	PA2 是一个双向I/O引脚，也可当作比较器输入引脚。 PA2 可输出PWM4。 PA2 是外部中断 1 的输入引脚INT1。 PA2 也是编程数据输入SDI。
PA3/ PWM3/ SDO	I/O	PA3 是一个双向I/O引脚，也可当作比较器输入引脚。 PA3 可输出PWM3。 PA3 也是编程数据输出SDO。
PA4/ PWM1/ EX_CKI0/ SCK	I/O	PA4 是一个双向I/O引脚。 PA4 可输出PWM1。 PA4 可当作定时器 0 / 1 外部时钟来源EX_CKI0。 PA4 也是编程时钟输入SCK。
PA5/ RSTb/ Vpp	I/O	PA5 可选择当作输入脚或开漏输出脚。 PA5 可当作复位引脚RSTb。 PA5 也是编程高压输入Vpp。
PA6/ Xin	I/O	PA6 是一个双向I/O引脚。 PA6 可当作晶振输入引脚Xin。
PA7/ Xout	I/O	PA7 是一个双向I/O引脚。 PA7 可当作晶振输出引脚Xout。 PA7 也可以当成指令时钟输出。
PB0/ PWM5/ INT0	I/O	PB0 是一个双向I/O引脚。 PB0 可输出PWM5。 PB0 是外部中断 0 的输入引脚INT0。
PB1/ IR/ INT1	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 是外部中断 1 的输入引脚INT1。
PB2/ PWM2	I/O	PB2 是一个双向I/O引脚。 PB2 可输出PWM2。
PB3/ PWM1/ BZ1/ CMPO/ SDO	I/O	PB3 是一个双向I/O引脚。 PB3 可输出BZ1，PWM1 或当作比较器输出引脚CMPO。 PB3 也是编程数据输出SDO。
PB4/ PWM3	I/O	PB4 是一个双向I/O引脚。 PB4 可输出PWM3。
PB5/ PWM4	I/O	PB5 是一个双向I/O引脚。 PB5 可输出PWM4。
VDD	P	电源正端。
VSS	P	电源负端。

2. 内存结构

NY8A054D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

NY8A054D程序存储器空间是 2Kx14 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。

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。NY8A054D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。

NY8A054D程序存储器地址 0x7FE~0x7FF是保留地址。如果用户在这些地址写入程序可能会发生无法预期的程序执行错误。

NY8A054D程序存储器地址 0x00E~0x00F是Preset Rolling Code地址。如果用户在不设置滚码时可当作程序区使用。

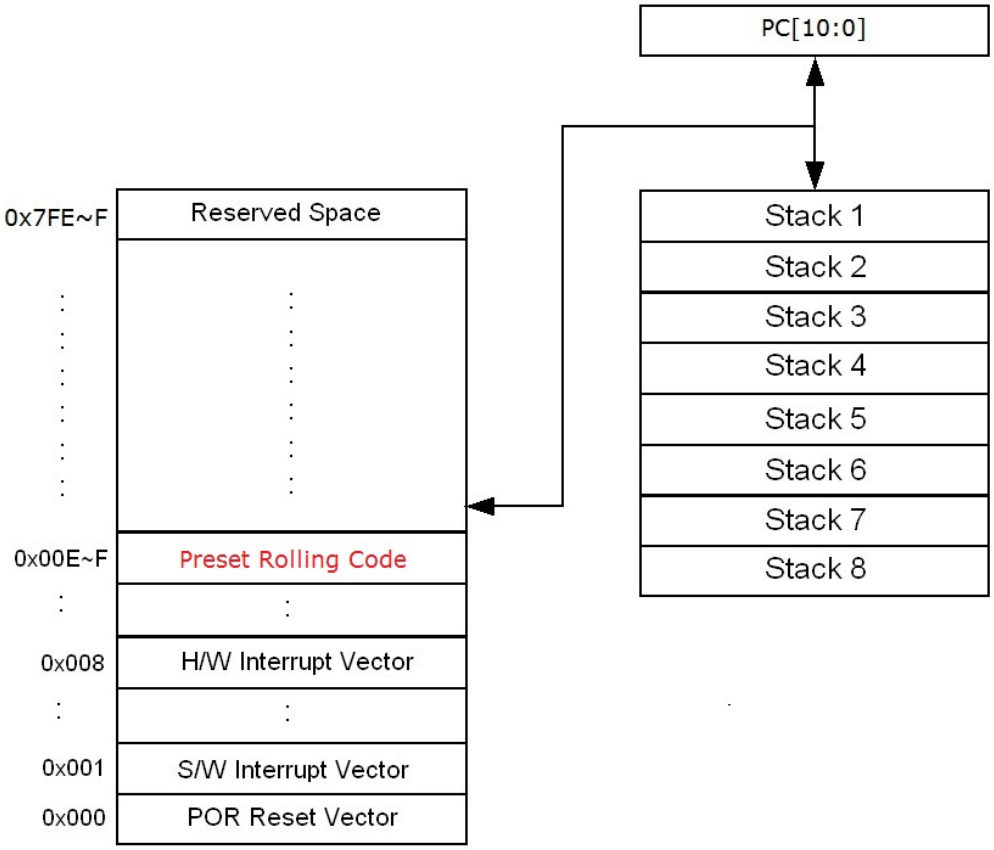


图 2　程序存储器对映地址

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

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

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。Bank选择是由STATUS[7:6]决定，地址选择则是由FSR[6:0]而定。

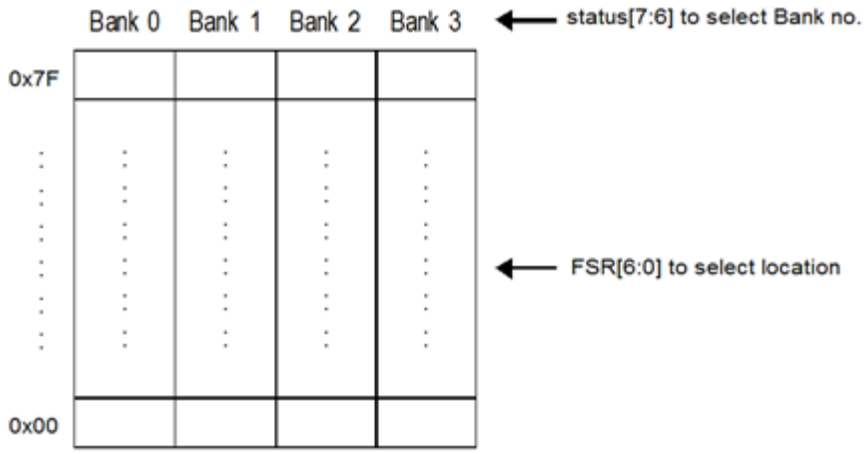


图 3　间接寻址方式访问数据存储器

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

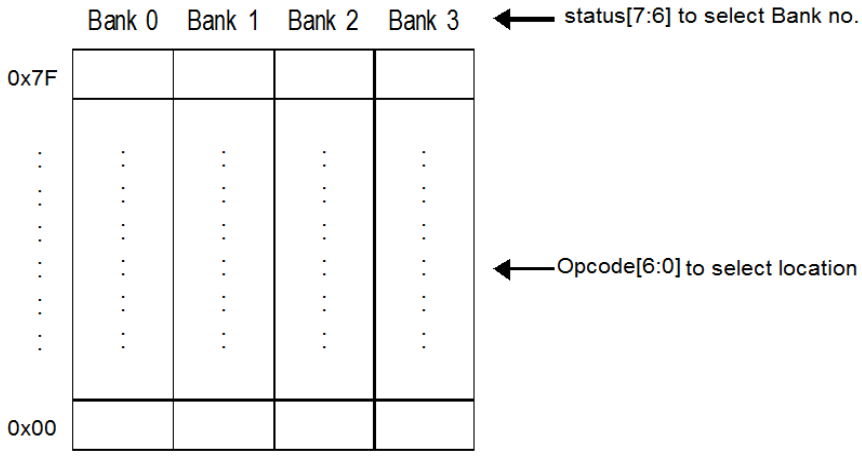


图 4　直接寻址方式访问数据存储器

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 0x1F。然而，Bank 1、Bank 2 和Bank 3 的相同地址会映像到Bank 0。换句话说，R-page特殊

功能寄存器只存在于Bank 0。GPR占用了Bank 0 数据存储器的 0x20 到 0x7F地址与Bank1 地址 0x20 到 0x3F，如表 1 所示。

NY8A054D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。

<table>
<tr><th>Status [7:6]
地址</th><th>00
(Bank 0)</th><th>01
(Bank 1)</th><th>10
(Bank 2)</th><th>11
(Bank 3)</th></tr>
<tr><td>0x0</td><td>INDF</td><td colspan="3" rowspan="28">映射至Bank 0</td></tr>
<tr><td>0x1</td><td>TMR0</td></tr>
<tr><td>0x2</td><td>PCL</td></tr>
<tr><td>0x3</td><td>STATUS</td></tr>
<tr><td>0x4</td><td>FSR</td></tr>
<tr><td>0x5</td><td>PORTA</td></tr>
<tr><td>0x6</td><td>PORTB</td></tr>
<tr><td>0x7</td><td>-</td></tr>
<tr><td>0x8</td><td>PCON</td></tr>
<tr><td>0x9</td><td>BWUCON</td></tr>
<tr><td>0xA</td><td>PCHBUF</td></tr>
<tr><td>0xB</td><td>ABPLCON</td></tr>
<tr><td>0xC</td><td>BPHCON</td></tr>
<tr><td>0xD</td><td>-</td></tr>
<tr><td>0xE</td><td>INTE</td></tr>
<tr><td>0xF</td><td>INTF</td></tr>
<tr><td>0x10</td><td>-</td></tr>
<tr><td>0x11</td><td>-</td></tr>
<tr><td>0x12</td><td>-</td></tr>
<tr><td>0x13</td><td>-</td></tr>
<tr><td>0x14</td><td>-</td></tr>
<tr><td>0x15</td><td>AWUCON</td></tr>
<tr><td>0x16</td><td>-</td></tr>
<tr><td>0x17</td><td>-</td></tr>
<tr><td>0x18</td><td>INTEDG</td></tr>
<tr><td>0x19</td><td>TMRH</td></tr>
<tr><td>0x1A</td><td>-</td></tr>
<tr><td>0x1B</td><td>RFC</td><td colspan="3" rowspan="2">映射至Bank 0</td></tr>
<tr><td>0x1C</td><td>TM34RH</td></tr>
<tr><td>0x1D ~ 0x1E</td><td>-</td><td colspan="3">-</td></tr>
</table>

Status [7:6] / 地址	00 (Bank 0)	01 (Bank 1)	10 (Bank 2)	11 (Bank 3)
0x1F	INTE2	映射至*Bank 0*		
0x20 ~ 0x3F	通用寄存器	通用寄存器	映射至*bank 0*	映射至*Bank 1*
0x40 ~ 0x7F	通用寄存器	映射至*bank 0*	映射至*bank 0*	映射至*bank 0*

表 1　R-page特殊功能寄存器地址映像表

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

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
0x8	-	TBHD
0x9	APHCON	-
0xA	PS0CV	P2CR1
0xB	-	-
0xC	BODCON	PWM2DUTY
0xD	-	-
0xE	CMPCR	-
0xF	PCON1	OSCCR
0x10	-	TMR3
0x11	-	T3CR1
0x12	-	T3CR2
0x13	-	PWM3DUTY
0x14	-	PS3CV
0x15	-	-
0x16	-	P4CR1
0x17	-	-
0x18	-	PWM4DUTY
0x19	-	-

特殊功能寄存器种类 / 地址	F-page SFR	S-page SFR
0x1A	-	-
0x1B	-	P5CR1
0x1C	-	-
0x1D	-	PWM5DUTY
0x1E	-	-
0x1F	-	PWM5RH

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NY8A054D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_CKI0或低频震荡器I_LRC/E_LXT中择一。

3.1.3 PCL（程序计数器低字节）

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

程序计数器（PC）是一个 11 位寄存器，分高 3 位和低 8 位。当程序执行了一个指令，同时PC数值会增加，除了某些指令会直接更改PC数值。PCL寄存器可访问PC低字节（PC[7:0]），PC高字节（PC[10:8]）并不能直接访问，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（OP-Code）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	BK[1]	BK[0]	GP5	/TO	/PD	Z	DC	C
读写属性			读/写	读/写	读/写	读/写(*2)	读/写(*1)	读/写	读/写	读/写
初始值			0	0	0	1	1	X	X	X

状态寄存器包含算术/逻辑指令的结果和是否发生看门狗超时复位。

C：进位/借位标志位。

C=1 时，加法运算有进位或减法运算无借位。

C=0 时，加法运算无进位或减法运算有借位。

DC：半进位/半借位标志位。

DC=1 时，加法运算低四位有进位或减法运算时没有向高四位借位。

DC=0 时，加法运算低四位无进位或减法运算时有向高四位借位。

Z：零位。

Z=1 时，算术或逻辑运算的结果是零。

Z=0 时，算术或逻辑运算的结果不为零。

/PD：睡眠模式标志位。

/PD=1 时，上电或执行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。

GP5：通用寄存器数据位。

BK[1:0]：Bank选择位，BK[1:0]=00b 选择Bank0，BK[1:0]=01b 选择Bank1，BK[1:0]=10b选择Bank2。BK[1:0]=11b，选择Bank3。

(*1) 可以被SLEEP指令清除。

(*2) 可以由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 128 个寄存器中选择一个。

GP7：通用寄存器数据位。

3.1.6 PortA（PortA 数据寄存器）

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

读取PortA时，若特定脚位被配置为输入脚，将得到该脚位输入状态。然而，若该脚位被配置为输出脚，依据配置字节，得到该脚位的状态或相对应的输出数据锁存值。当写入PortA时，数据是被写入PortA的输出数据锁存器中。

3.1.7 PortB（PortB 数据寄存器）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PortB	R	0x6	GP7	GP6	PB5	PB4	PB3	PB2	PB1	PB0
读写属性			读/写							
初始值			数据锁存器值是xxxxxx，读取值则是xxxxxx端口值（PB5~PB0）							

读取PortB时，若特定脚位被配置为输入脚，将得到该脚位输入状态。然而，若该脚位被配置为输出脚，依据配置字节，得到该脚位的状态或相对应的输出数据锁存值。当写入PortB时，数据是被写入PortB的输出数据锁存器中。

GP7~6：通用寄存器数据位。

3.1.8 PCON（Power 寄存器）

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

GP6，GP4，GP1，GP0：通用寄存器数据位。

LVREN：开启/关闭 LVR。
LVREN=1 时，开启LVR。
LVREN=0 时，关闭LVR。

LVDEN：开启/关闭 LVD。
LVDEN=1 时，开启LVD。
LVDEN=0 时，关闭LVD。

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

CMPEN：开启/关闭比较器。
CMPEN=1 时，开启比较器。
CMPEN=0 时，关闭比较器。

3.1.9 BWUCON（PortB 唤醒控制寄存器）

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

WUPBx：开启/关闭PBx唤醒功能，0 ≤ x ≤ 5。

WUPBx=1 时，开启PBx唤醒功能。

WUPBx=0 时，关闭PBx唤醒功能。

3.1.10 PCHBUF（程序计数器高字节）

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

PCHBUF[2:0]：程序计数器PC的第十个位到第八个位。

XSPD_STP：写 1 停止外部晶振 32.768KHz起振强化功能。

3.1.11 ABPLCON（PortA/PortB 下拉电阻控制寄存器）

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

/PLPAx：关闭/开启PAx下拉电阻，0 ≤ 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	-	-	/PHPB5	/PHPB4	/PHPB3	/PHPB2	/PHPB1	/PHPB0
读写属性			-	-	读/写	读/写	读/写	读/写	读/写	读/写
初始值			X	X	1	1	1	1	1	1

/PHPBx：关闭/开启PBx上拉电阻，0 ≤ x ≤ 5。

/PHPBx=1 时，关闭PBx上拉电阻。

/PHPBx=0 时，开启PBx上拉电阻。

3.1.13 INTE（中断使能寄存器）

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

T0IE：定时器 0 上溢（overflow）中断使能位。

T0IE=1 时，开启定时器 0 上溢中断。

T0IE=0 时，关闭定时器 0 上溢中断。

PABIE：PortA / PortB输入状态变化中断使能位。

PABIE=1 时，开启PortA/ PortB输入状态变化中断。

PABIE=0 时，关闭PortA/ PortB输入状态变化中断。

INT0IE：外部中断 0 使能位。

INT0IE=1 时，开启外部中断 0。

INT0IE=0 时，关闭外部中断 0。

T1IE：定时器 1 下溢（underflow）中断使能位。

T1IE=1 时，开启定时器 1 下溢中断。

T1IE=0 时，关闭定时器 1 下溢中断。

LVDIE：低电压侦测中断使能位。

LVDIE=1 时，开启低电压侦测中断。

LVDIE=0 时，关闭低电压侦测中断。

WDTIE：WDT上溢中断使能位。

WDTIE=1 时，开启WDT上溢中断。

WDTIE=0 时，关闭WDT上溢中断。

INT1IE：外部中断 1 使能位。

INT1IE=1 时，开启外部中断 1。

INT1IE=0 时，关闭外部中断 1。

3.1.14 INTF（中断标志寄存器）

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

T0IF：定时器 0 上溢中断标志位。

T0IF=1 时，发生定时器 0 上溢中断。

T0IF必须由程序清零。

PABIF：PortA / PortB输入状态变化中断标志位。

PABIF=1 时，发生PortA / PortB输入状态变化中断。

PABIF必须由程序清零。

INT0IF：外部中断 0 标志位。

INT0IF=1 时，发生外部 0 中断。

INT0IF必须由程序清零。

T1IF：定时器 1 下溢中断标志位。

T1IF=1 时，发生定时器 1 下溢中断。

T1IF必须由程序清零。

LVDIF：低电压侦测中断标志位。

LVDIF=1，发生低电压侦测中断。

LVDIF必须由程序清零。

WDTIF：WDT超时上溢标志位。

WDTIF=1 时，发生WDT上溢中断。

WDTIF必须由程序清零。

INT1IF：外部中断 1 标志位。

INT1IF=1 时，发生外部 1 中断。

INT1IF必须由程序清零。

注意：当对应的INTE寄存器控制位未使能，读取中断标志是 0。

3.1.15 AWUCON（PortA 唤醒控制寄存器）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
AWUCON	R	0x15	WUPA7	WUPA6	WUPA5	WUPA4	WUPA3	WUPA2	WUPA1	WUPA0
读写属性			读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值			1	1	1	1	1	1	1	1

WUPAx：开启/关闭PAx唤醒功能，0 ≤ x ≤ 7。

WUPAx=1 时，开启PAx唤醒功能。

WUPAx=0 时，关闭PAx唤醒功能。

3.1.16 INTEDG（外部中断控制寄存器）

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

EIS1：外部中断 1 引脚选择位。

EIS1=1 时，PB1/PA2 选择为外部中断 1 引脚。

EIS1=0 时，PB1/PA2 选择为GPIO。

EIS0：外部中断 0 引脚选择位。

EIS0=1 时，PB0 选择为外部中断 0 引脚。

EIS0=0 时，PB0 选择为GPIO。

INT1G1~0：INT1 沿触发选择位。

00：保留，01：上升沿触发，10：下降沿触发，11：上升/下降沿触发。

INT0G1~0：INT0 沿触发选择位。

00：保留，01：上升沿触发，10：下降沿触发，11：上升/下降沿触发。

3.1.17 TMRH（定时器 1 高字节寄存器）

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

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

读取这 2 位将得到定时器 1 第 9 位与第 8 位目前计数值。

PWM2DUTY9~8：PWM2 占空比高 2 位。

PWM1DUTY9~8：PWM1 占空比高 2 位。

3.1.18 RFC（电阻频率转换控制寄存器）

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

RFCEN：关闭/开启RFC。

RFCEN=1，开启RFC。

RFCEN=0，关闭RFC。

PSEL[3:0]：选择RFC输入引脚。

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

表 3　选择RFC输入引脚

3.1.19 TM34RH（定时器 3 高字节寄存器）

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TM34RH	R	0x1C	-	-	TMR39	TMR38	PWM4 DUTY9	PWM4 DUTY8	PWM3 DUTY9	PWM3 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

TMR39~8：定时器 3 高 2 位。写这 2 位将覆写定时器 3 第 9 位与第 8 位重载值。

读取这 2 位将得到定时器 3 第 9 位与第 8 位目前计数值。

PWM3DUTY9~8：PWM3 占空比高 2 位。

PWM4DUTY9~8：PWM4 占空比高 2 位。

3.1.20 INTE2（第 2 中断使能寄存器）

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTE2	R	0x1F	-	-	-	T3IF	-	-	-	T3IE
读写属性			-	-	-	读/写	-	-	-	读/写
初始值			-	-	-	0	-	-	-	0

T3IF：定时器 3 下溢中断标志位。

T3IF=1 时，发生定时器 3 下溢中断。

T3IF必须由程序清零。

T3IE：定时器 3 下溢（underflow）中断使能位。

T3IE=1 时，开启定时器 3 下溢中断。

T3IE=0 时，关闭定时器 3 下溢中断。

3.2 T0MD定时器 0 控制寄存器

T0MD是可读写寄存器，但只能由指令T0MD / T0MDR访问。

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

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

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

表 4　预分频器 0 的预分频比选项

PS0WDT：预分频器 0 分配选择。

PS0WDT=1 时，预分频器 0 被分配到WDT。

PS0WDT=0 时，预分频器 0 被分配到定时器 0。

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

T0CE：定时器 0 外部时钟源触发沿选择。

T0CE=1 时，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_CKI0 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_CKI0 脚或低频震荡I_LRC/E_LXT。

T0CS=0 时，选择指令时钟F_{INST}。

GP6：通用读写器寄存器。

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

T0CS=1 时，LCKTM0=0 时，外部EX_CKI0 脚被选择当作定时器 0 时钟源。

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

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	IOPA7	IOPA6	IOPA5	IOPA4	IOPA3	IOPA2	IOPA1	IOPA0
读写属性			读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值			1	1	1	1	1	1	1	1

IOPAx：PAx I/O模式选择，0 ≤ x ≤ 7。

IOPAx=1 时，PAx设为输入口。

IOPAx=0 时，PAx设为输出口。

3.3.2 IOSTB（PortB I/O 控制寄存器）

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

IOPBx：PBx I/O模式选择，0 ≤ x ≤ 5。

IOPBx=1 时，PBx设为输入口。

IOPBx=0 时，PBx设为输出口。

3.3.3 APHCON（PortA 上拉电阻控制寄存器）

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

/PHPAx：关闭/开启PAx上拉电阻，0 ≤ x ≤ 7。

/PHPAx=1 时，关闭 PAx上拉电阻。

/PHPAx=0 时，开启 PAx上拉电阻。

/PLPA5：关闭/开启下拉电阻PA5。

/PLPA5=1 时，关闭 PA5 下拉电阻。

/PLPA5=0 时，开启 PA5 下拉电阻。

注意：PA6 与 PA7 作为晶振引脚时应关闭内置上拉电阻。

3.3.4 PS0CV（预分频器 0 寄存器）

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

读取PS0CV时，会得到预分频器 0 器的目前计数值。

3.3.5 BODCON（PortB 开漏控制寄存器）

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

ODPBx：开启/关闭PBx的开漏，0 ≤ x ≤ 5。

ODPBx=1 时，开启PBx的开漏。

ODPBx=0 时，关闭PBx的开漏。

3.3.6 CMPCR（比较器控制寄存器）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPCR	F	0xE	PS3	PS2	PS1	PS0	VS3	VS2	VS1	VS0
读写属性			读/写							
初始值			0	0	0	0	0	0	0	0

VS[3:0]，PS[3:0]：当VS[3:0]=0 时，为P2P模式。VS[3:0]不为 0 时为P2V模式。

选择为P2V模式时，VS[3:0]选择其中内部参考电压输入比较器负输入源。PS[3:0]选择PA0~PA4 其中一引脚输入比较器正输入源。当比较器在P2P模式，寄存器VS[3:0]必须设为 0，寄存器PS[3:0]选择下表 4 种组合中的其中 2 根引脚到比较器的正输入源与负输入源。有关于P2P模式的细节，请看比较器功能说明章节 3.15。

VS[3:0]	比较器参考电压	PS[3:0]	比较器正输入源
0000	P2P mode	0000	PA0
0001	1 / 16 VDD	0001	PA1
0010	2 / 16 VDD	0010	PA2
0011	3 / 16 VDD	0011	PA3
0100	4 / 16 VDD	0100	-
0101	5 / 16 VDD	0101	-
0110	6 / 16 VDD	0110	-
0111	7 / 16 VDD	0111	-
1000	8 / 16 VDD	1000	-
1001	9 / 16 VDD	1001	-
1010	10 / 16 VDD	1010	-
1011	11 / 16 VDD	1011	-
1100	12 / 16 VDD	1100	-
1101	13 / 16 VDD	1101	-
1110	14 / 16 VDD	1110	-
1111	15 / 16 VDD	1111	-

表 5　P2V模式

PS[3:0]	比较器正输入源	比较器负输入源
0000	PA0	PA1
0001	PA1	PA0
0010	PA2	PA3
0011	PA3	PA2
0100	-	-
0101	-	-
0110	-	-
0111	-	-
1000	-	-
1001	-	-
1010	-	-
1011	-	-
1100	-	-
1101	-	-
1110	-	-
1111	-	-

表 6　P2P模式（VS[3:0] = 4’b0000）

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	GP5	LVDS2	LVDS1	LVDS0	GP1	T0EN
读写属性			读/写(1*)	读	读/写	读/写	读/写	读/写	读/写	读/写
初始值			0	X	0	1	1	1	0	1

T0EN：开启/关闭定时器 0。

T0EN=1 时，开启定时器 0。

T0EN=0 时，关闭定时器 0。

LVDS2~0：选择LVD电压。

LVDS[2:0]	电压
000	2.0V
001	2.2V
010	2.4V
011	2.7V
100	3.0V
101	3.3V
110	3.6V
111	4.3V

表 7　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。

GP5，GP1：通用寄存器数据位。

(1*)：由指令 ENI 设置 1、指令 DISI 清除、指令 IOSTR 所读取。

3.4 S-page特殊功能寄存器

3.4.1 TMR1（定时器 1 寄存器）

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

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

3.4.2 T1CR1（定时器 1 控制寄存器 1）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T1CR1	S	0x1	PWM1OEN	PWM1OAL	-	-	-	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[9:0]寄存器被重新加载。

T1RL=0 时，当下溢发生，定时器 1 继续从 0x3FF下数。

T1OS：当下溢发生，设置定时器 1 操作模式。

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

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

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

表 8　定时器 1 功能

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 为电平高有效位输出。

PWM1OEN：开启/关闭PWM1 输出。

PWM1OEN=1，PB3 或PA4 输出PWM1。

PWM1OEN=0，PB3 或PA4 为GPIO。

3.4.3 T1CR2（定时器 1 控制寄存器 2）

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

该寄存器用于配置定时器 1 功能。

PS1SEL[2:0]：预分频器 1 预分频比选项。

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

表 9　预分频器 1 预分频比选项

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

/PS1EN：关闭/开启预除器 1。

/PS1EN=1 时，关闭预分频器 1。

/PS1EN=0 时，开启预分频器 1。

T1CE：定时器 1 外部时钟触发沿选项。

T1CE=1 时，EX_CKI0 脚下降沿时定时器 1 减一。

T1CE=0 时，EX_CKI0 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_CKI0 脚作为外部时钟输入。

T1CS=0 时，选择指令时钟F_{INST}。

3.4.4 PWM1DUTY（PWM1 占空比寄存器）

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

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

3.4.5 PS1CV（预分频器 1 寄存器）

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

读取PS1CV时，将会得到预分频器 1 的目前数值。

3.4.6 BZ1CR（蜂鸣器 1 控制寄存器）

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

BZ1FSEL[3:0]：BZ1 输出频率选项。

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

表 10　蜂鸣器BZ1 输出频率选项

BZ1EN：开启/关闭蜂鸣器 1 输出。

BZ1EN=1 时，开启蜂鸣器 1。

BZ1EN=0 时，关闭蜂鸣器 1。

3.4.7 IRCR（IR 控制寄存器）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IRCR	S	0x6	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类型。

若选择 I_HRC此位将被忽略。

IROSC358M=1，外部晶振频率请用 3.58MHz。

IROSC358M=0，外部晶振频率请用 455KHz。

注意：

1. 仅有高速震荡时钟 F_{HOSC}（详见章节 3.18）可以当作IR时钟源。

2. 不同振荡类型的分频比。

OSC. Type	57KHz	38KHz	条件
High IRC	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	E_XT 模式& IROSC358M=1
Xtal 455KHz	8	12	E_XT 模式& IROSC358M=0

表 11　不同振荡类型的分频比

3.4.8 TBHP（表格指针高字节寄存器）

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

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

3.4.9 TBHD（表格数据高字节寄存器）

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

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

3.4.10 P2CR1（PWM2 控制寄存器 1）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2CR1	S	0xA	PWM2OEN	PWM2OAL	-	-	-	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

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 为电平高有效位输出。

PWM2OEN：开启/关闭PWM2 输出。

PWM2OEN=1 时，PB2 或PA0 输出PWM2。

PWM2OEN=0 时，PB2 或PA0 是I/O口。

T1EN：开启/关闭定时器 1。**(*)**

T1EN=1 时，开启定时器 1。

T1EN=0 时，关闭定时器 1。

T1RL：当连续模式被选择（T1OS=0），选择定时器 1 下数方式。**(*)**

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

T1RL=0 时，当下溢发生，定时器 1 继续从 0x3FF下数。

T1OS：当下溢发生，设置定时器 1 操作模式。**(*)**

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

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

注意：P2CR1 中的 T1EN、T1RL、T1OS 位用于 PWM2 输出，必须与 T1CR1 中的 T1EN、T1RL、T1OS 位设定一致。

3.4.11 PWM2DUTY（PWM2 占空比寄存器）

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

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

3.4.12 OSCCR（振荡器控制寄存器）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSCCR	S	0xF	CMPOUT	CMPOE	CMPIF	CMPIE	OPMD[1:0]		STPHOSC	SELHOSC
读写属性			读	读/写	读/写	读/写	读/写		读/写	读/写
初始值			X	0	0	0	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/开启高频率震荡器（F_{HOSC}）。

STPHOSC=1 时，F_{HOSC}会停止振荡并被关闭。

STPHOSC=0 时，F_{HOSC}保持振荡。

OPMD[1:0]：选择操作模式。

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

表 12　选择OPMD[1:0]的操作模式

CMPIE：比较器中断使能位。

CMPIE=1 时，开启比较器中断。

CMPIE=0 时，关闭比较器中断。

CMPIF：比较器输出翻转中断标志位。

CMPIF=1，发生比较器输出翻转中断。

CMPIF必须由程序清零。

CMPOE：开启/关闭比较器输出到PB3 引脚。

CMPOE=1 时，开启比较器输出到PB3 引脚。

CMPOE=0 时，关闭比较器输出到PB3 引脚。

注意：比较器输出到 PB3 引脚优先于 PWM1/BUZZER1。

CMPOUT：比较器输出状态位。

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

3.4.13 TMR3（定时器 3 寄存器）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMR3	S	0x10	TMR3[7:0]							
读写属性			读/写							
初始值			XXXXXXXX							

当读取TMR3 寄存器时，会得到 10 位定时器 3 中的低字节目前计数值。写TMR3 时，会将TM34RH[5:4]和TMR3[7:0]一起写到定时器 3 重载缓存器中。

3.4.14 T3CR1（定时器 3 控制寄存器 1）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T3CR1	S	0x11	PWM3OEN	PWM3OAL	-	-	-	T3OS	T3RL	T3EN
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 3 功能。

T3EN：开启/关闭定时器 3。

T3EN=1 时，开启定时器 3。

T3EN=0 时，关闭定时器 3。

T3RL：当连续模式被选择（T3OS=0），选择定时器 3 下数方式。

T3RL=1 时，当下溢发生，定时器 3 初始值从TMR3[9:0]寄存器被重新加载。

T3RL=0 时，当下溢发生，定时器 3 继续从 0x3FF下数。

T3OS：当下溢发生，设置定时器 3 操作模式。

T3OS=1 时，单次计数模式（One-Shot mode）。定时器 3 会从初始值到 0x00 计数一次。

T3OS=0 时，连续计数模式（Non-Stop mode）。下溢后，定时器 3 会持续下数。

T3OS	T3RL	定时器 3 计数选项
0	0	定时器 3 从 0x3FF下数到 0x00。 当下溢发生，0x3FF被重载至定时器 3 并继续下数。
0	1	定时器 3 从重载的数值下数到 0x00。 当下溢发生，定时器 3 从TMR3[9:0]重新载入数值并继续下数。
1	x	定时器 3 从初始值下数到 0x00。 当下溢发生，定时器 3 停止下数。

表 13　定时器 3 功能

PWM3OAL：定义PWM3 输出有效状态。

PWM3OAL=1 时，PWM3 为低电平有效位输出。

PWM3OAL=0 时，PWM3 为电平高有效位输出。

PWM3OEN：开启/关闭PWM3 输出。

PWM3OEN=1，PA3 或PB4 输出PWM3。

PWM3OEN=0，PA3 或PB4 为GPIO。

3.4.15 T3CR2（定时器 3 控制寄存器 2）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T3CR2	S	0x12	-	-	T3CS	T3CE	/PS3EN	PS3SEL[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 3 功能。

PS3SEL[2:0]：预分频器 3 预分频比选项。

PS3SEL[2:0]	预分频比选项
000	1:2
001	1:4
010	1:8
011	1:16
100	1:32
101	1:64
110	1:128
111	1:256

表 14　预分频器 3 预分频比选项

注意：在PS3EN=1 前须先设定PS3SEL[2:0]，否则可能会误发生中断。

/PS3EN：关闭/开启预除器 3。

/PS3EN=1 时，关闭预分频器 3。

/PS3EN=0 时，开启预分频器 3。

T3CE：定时器 3 外部时钟触发沿选项。

T3CE=1 时，EX_CKI1 脚下降沿时定时器 3 减一。

T3CE=0 时，EX_CKI1 脚上升沿时定时器 3 减一。

T3CS：定时器 3 时钟源选项。

T3CS=1 时，选择EX_CKI1 脚作为外部时钟输入。

T3CS=0 时，选择指令时钟F_{INST}。

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 3 重新加载的数值储存在TM34RH[5:4]与TMR3[7:0]寄存器，以用来定义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 PS3CV（预分频器 3 寄存器）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS3CV	S	0x14	PS3CV[7:0]							
读写属性			读							
初始值			1	1	1	1	1	1	1	1

读取PS3CV时，将会得到预分频器 3 的目前数值。

3.4.18 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	-	-	-	T3OS	T3RL	T3EN
读写属性			读/写	读/写	-	-	-	读/写	读/写	读/写
初始值			0	0	X	X	X	X	X	0

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 为电平高有效位输出。

PWM4OEN：开启/关闭PWM4 输出。

PWM4OEN=1，PA2 或PB5 输出PWM4。

PWM4OEN=0，PA2 或PB5 为GPIO。

T3EN：开启/关闭定时器 3。**(*)**

T3EN=1 时，开启定时器 3。

T3EN=0 时，关闭定时器 3。

T3RL：当连续模式被选择（T3OS=0），选择定时器 3 下数方式。**(*)**

T3RL=1 时，当下溢发生，定时器 3 初始值从TMR3[9:0]寄存器被重新加载。

T3RL=0 时，当下溢发生，定时器 3 继续从 0x3FF下数。

T3OS：当下溢发生，设置定时器 3 操作模式。**(*)**

T3OS=1 时，单次计数模式（One-Shot mode）。定时器 3 会从初始值到 0x00 计数一次。

T3OS=0 时，连续计数模式（Non-Stop mode）。下溢后，定时器 3 会持续下数。

注意：P4CR1 中的T3EN、T3RL、T3OS位用于PWM4 输出，必须与T3CR1 中的T3EN、T3RL、T3OS位设定一致。

3.4.19 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 3 重新加载的数值储存在TM34RH[5:4]与TMR3[7:0]寄存器，以用来定义PWM4 帧率，TM34RH[3:2]与PWM4DUTY[7:0]寄存器用于定义PWM4 的占空比。

3.4.20 P5CR1（PWM5 控制寄存器 1）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P5CR1	S	0x1B	PWM5OEN	PWM5OAL	-	-	-	T3OS	T3RL	T3EN
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

PWM5OAL：定义PWM5 输出有效状态。

PWM5OAL=1 时，PWM5 为低电平有效位输出。

PWM5OAL=0 时，PWM5 为电平高有效位输出。

PWM5OEN：开启/关闭PWM5 输出。

PWM5OEN=1，PA1 或PB0 输出PWM5。

PWM5OEN=0，PA1 或PB0 为GPIO。

T3EN：开启/关闭定时器 3。**(*)**

T3EN=1 时，开启定时器 3。

T3EN=0 时，关闭定时器 3。

T3RL：当连续模式被选择（T3OS=0），选择定时器 3 下数方式。**(*)**

T3RL=1 时，当下溢发生，定时器 3 初始值从TMR3[9:0]寄存器被重新加载。

T3RL=0 时，当下溢发生，定时器 3 继续从 0x3FF下数。

T3OS：当下溢发生，设置定时器 3 操作模式。**(*)**

T3OS=1 时，单次计数模式（One-Shot mode）。定时器 3 会从初始值到 0x00 计数一次。

T3OS=0 时，连续计数模式（Non-Stop mode）。下溢后，定时器 3 会持续下数。

注意：P5CR1 中的T3EN、T3RL、T3OS位用于PWM5 输出，必须与T3CR1 中的T3EN、T3RL、T3OS位设定一致。

3.4.21 PWM5DUTY（PWM5 占空比寄存器）

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

定时器 3 重新加载的数值储存在TM34RH[5:4]与TMR3[7:0]寄存器，以用来定义PWM5 帧率，PWM5RH[1:0]与PWM5DUTY[7:0]寄存器用于定义PWM5 的占空比。

3.4.22 PWM5RH（PWM5 占空比高字节寄存器）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM5RH	S	0x1F	-	-	-	-	-	-	PWM5DUTY9	PWM5DUTY8
读写属性			-	-	-	-	-	-	读/写	读/写
初始值			X	X	X	X	X	X	X	X

PWM5DUTY9~8：PWM5 占空比高 2 位。

3.5 I/O Port

NY8A054D提供 14 个I/O口（PA[7:0]和PB[5:0]），用户可以由寄存器PORTA和PORTB读写这些脚位。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[7:0]定义PA[7:0]为输入口或输出口，寄存器IOSTB[5:0]定义PB[5:0]为输入口或输出口。

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

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

I/O口功能摘要如下表：

功能		PA[3:0]	PA[7:6] PA[4]	PA[5]	PB[3:0]	PB[5:4]
输入	上拉电阻	V	V	X	V	V
	下拉电阻	V	X	V	V	X
输出	开漏	X	X	总是	V	V

表 15　I/O端口功能摘要

在PA和PB的每个I/O脚都有输入状态改变产生中断功能。寄存器AWUCON[7:0]和BWUCON[5:0]会使能或禁止任一PA和PB脚位的唤醒功能。只要AWUCON和BWUCON对应到的任一PA和PB脚位被置为 1 时，且在此输入脚位

有状态改变时，寄存器PABIF（INTF[1]）就会被设为 1。如果寄存器PABIE（INTE[1]）与GIE（PCON1[7]）同时设定为 1，将发生中断要求并执行中断服务程序。

NY8A054D提供 2 个外部中断，当寄存器EIS0（INTEDG[4]）设定为 1，PB0 则被当作外部中断 0 的输入脚。当寄存器EIS1（INTEDG[5]）设定为 1，PB1 或PA2 则被当作外部中断 1 的输入脚。

注意：当PB0 或PB1/PA2 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 或PB1/PA2 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。

NY8A054D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。

由配置字节决定PA5 可否当作外部复位输入RSTb。当PA5 为低电平时将导致NY8A054D发生复位。

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

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

如果寄存器T0CS（T0MD[5]）为 1 和LCK_TM0（T0MD[7]）为 0，EX_CKI0 可以当作定时器 0 外部时钟源。如果寄存器T1CS（T1CR2[5]）为 1，EX_CKI0 可以当作定时器 1 外部时钟源。如果寄存器T3CS（T3CR2[5]）为 1，EX_CKI1 可以当作定时器 3 外部时钟源。

如果寄存器CMPOE=1，PB3 也可以当作比较器输出引脚。如果寄存器PWM1OEN（T1CR1[7]）为 1，PB3 或PA4 也可以当作脉冲宽度调制PWM1 输出。若寄存器BZ1EN （BZ1CR[7]）为 1，PB3 也可以当作蜂鸣器 1 输出。PB3 复用引脚输出优先级为比较器输出 > PWM1 输出 > 蜂鸣器 1 输出。

当寄存器P2CR1[7] PWM2OEN=1，PB2 或PA0 就会输出PWN2。

3.5.1 IO 引脚结构框图

IO_SEL：设定引脚为输入或输出口。

WRITE_EN：将数据写入引脚。

READ_EN：读取引脚状态。

PULLUP100K_ENB：开启内部上拉电阻。

PULLDOWN_EN：开启内部下拉电阻。

VPEN：开启为比较器正输入引脚。

VNEN：开启为比较器负输入引脚。

CMPVP，CMPVN：比较器正输入与负输入引脚。

RD_TYPE：选择读取脚位或数据锁存器。

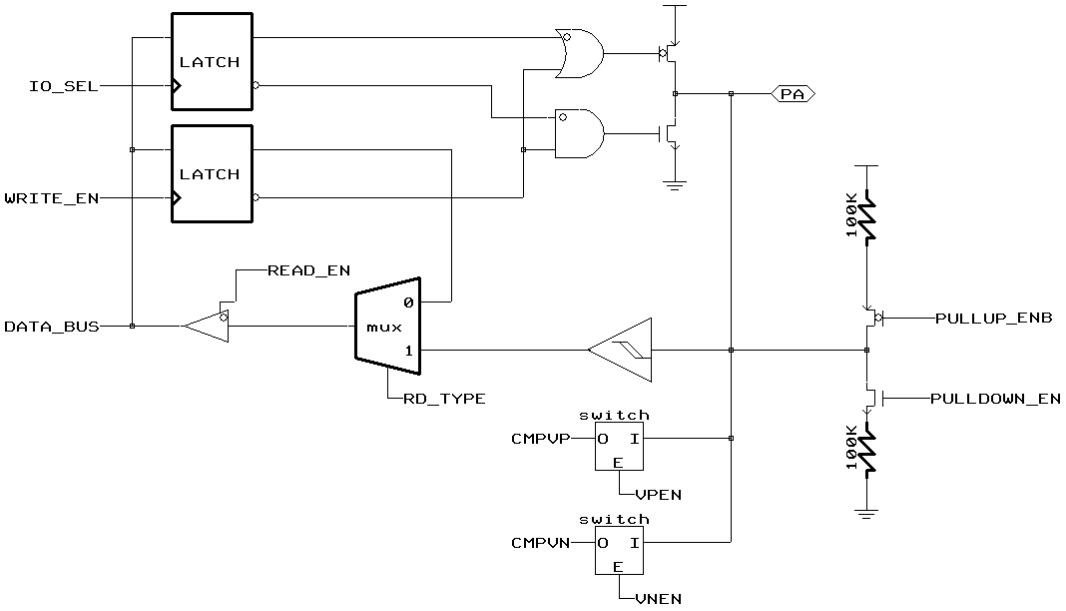


图 5　PA[3:2]，PA0 引脚结构框图

IO_SEL：设定引脚为输入或输出口。

WRITE_EN：将数据写入引脚。

READ_EN：读取引脚状态。

PULLUP100K_ENB：开启内部上拉电阻。

PULLDOWN_EN：开启内部下拉电阻。

VPEN：开启为比较器正输入引脚。

VNEN：开启为比较器负输入引脚。

CMPVP，CMPVN：比较器正输入与负输入引脚。

RD_TYPE：选择读取脚位或数据锁存器。

EX_CKI1：Timer 3 外部时钟输入。

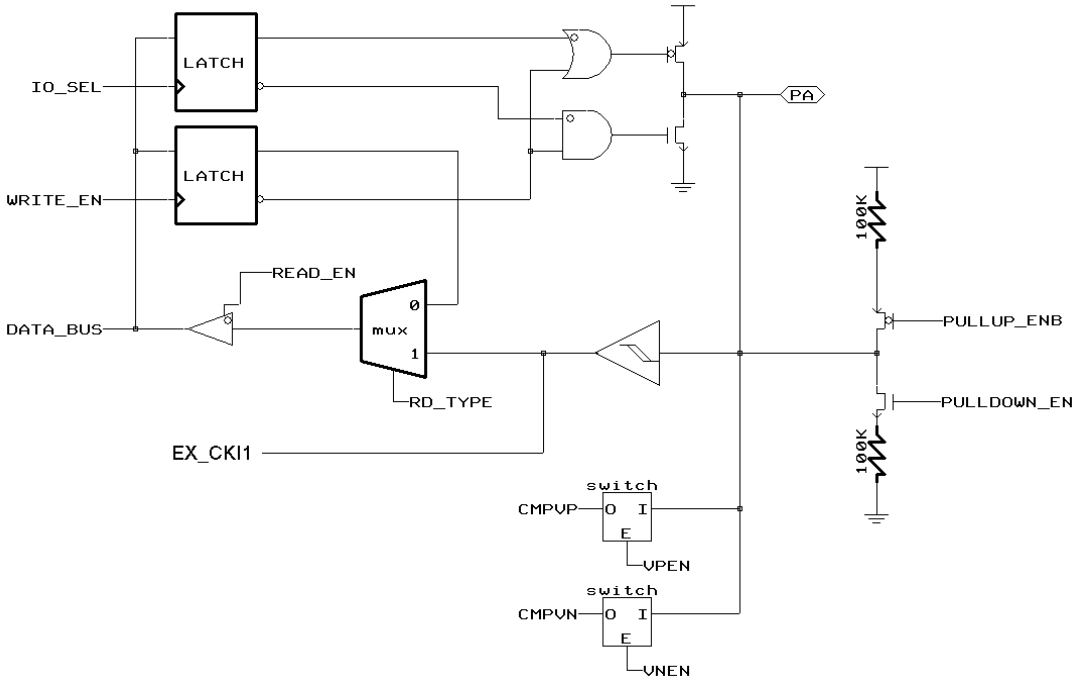


图 6　PA1 引脚结构框图

IO_SEL：设定引脚为输入或输出口。

WRITE_EN：将数据写入引脚。

READ_EN：读取引脚状态。

PULLUP100K_ENB：开启内部上拉电阻。

RD_TYPE：选择读取脚位或数据锁存器。

EX_CKI0：Timer0,1 外部时钟输入。

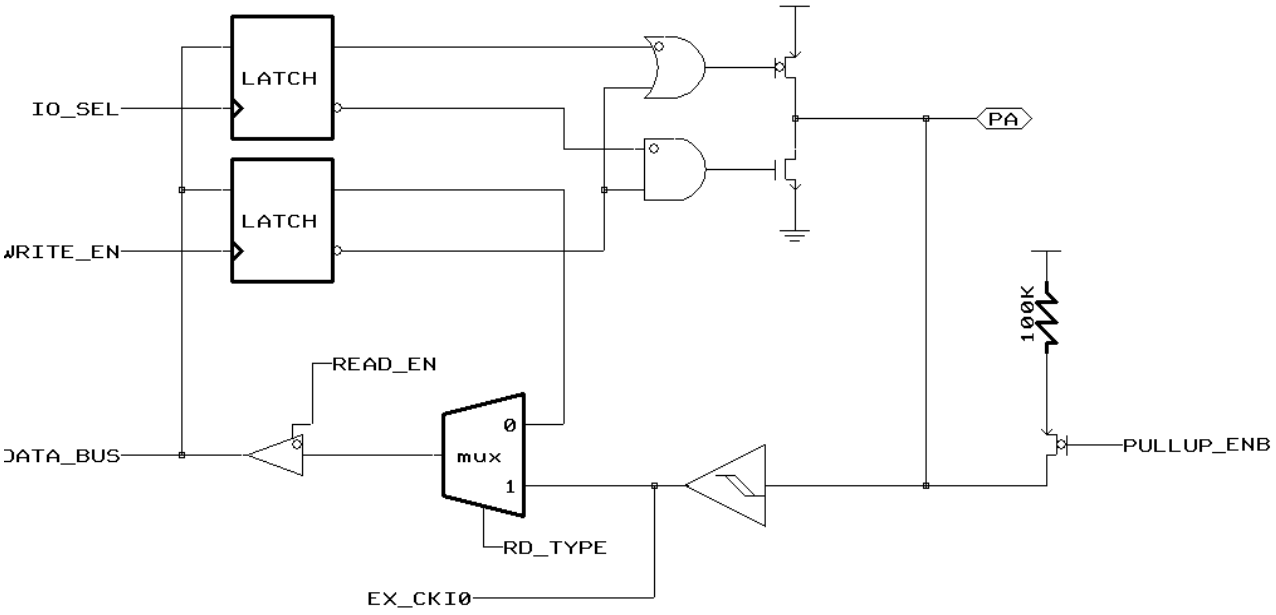


图 7　PA4 引脚结构框图

RSTPAD_EN：开启外部复位引脚。

RSTB_IN：复位引脚输入。

IO_SEL：设定引脚为输入或输出口。

WRITE_EN：将数据写入引脚。

READ_EN：读取引脚状态。

PULLDOWN_EN：开启内部下拉电阻。

RD_TYPE：选择读取脚位或数据锁存器。

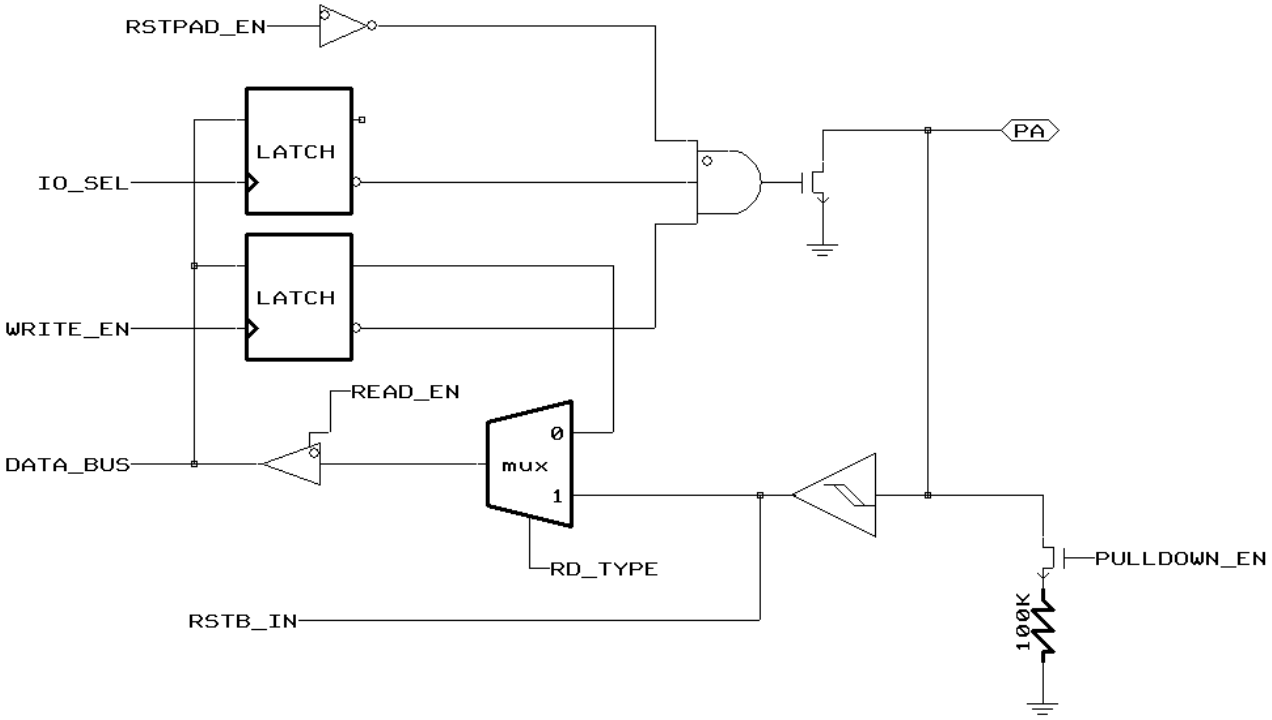


图 8　PA5 引脚结构框图

XTL_EN：开启外置晶振。

IO_SEL：设定引脚为输入或输出口。

WRITE_EN：将数据写入引脚。

READ_EN：读取引脚状态。

PULLUP100K_ENB：开启内部上拉电阻。

RD_TYPE：选择读取脚位或数据锁存器。

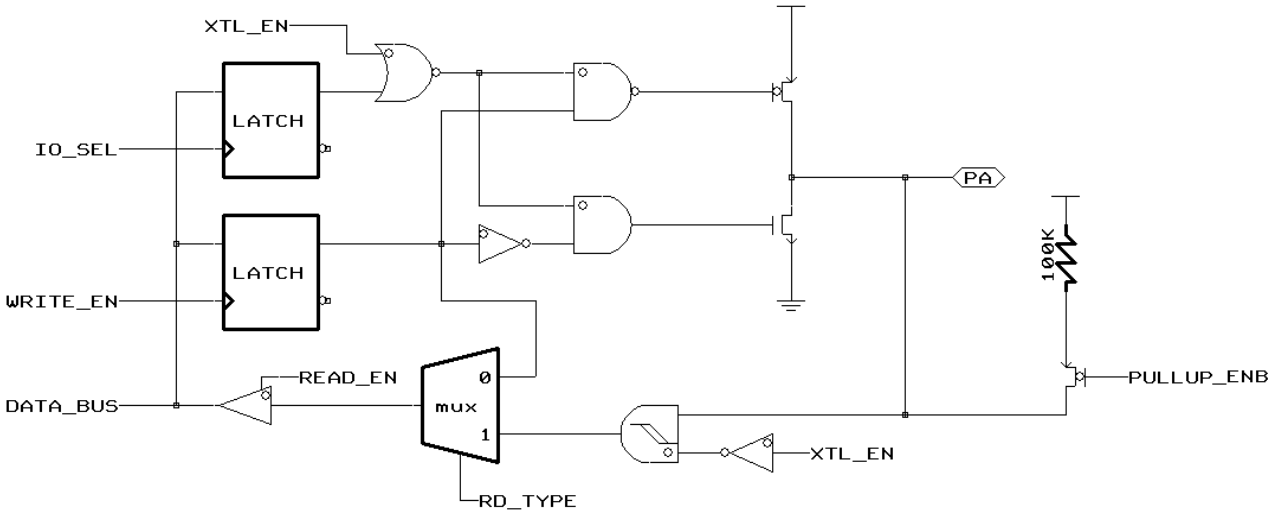


图 9　PA6，PA7 引脚结构框图

IO_SEL：设定引脚为输入或输出口。

WRITE_EN：将数据写入引脚。

READ_EN：读取引脚状态。

OD_EN：开启开漏。

PULLUP100K_ENB：开启内部上拉电阻。

PULLDOWN_EN：开启内部下拉电阻。

RD_TYPE：选择读取脚位或数据锁存器。

EIS0：使能外部中断 0 功能。

INTEDG[1:0]：选择外部中断 0 触发沿。

EX_INT0：外部中断 0 信号。

WUB：使能PB唤醒功能。

SET_PBIF：PB唤醒标志。

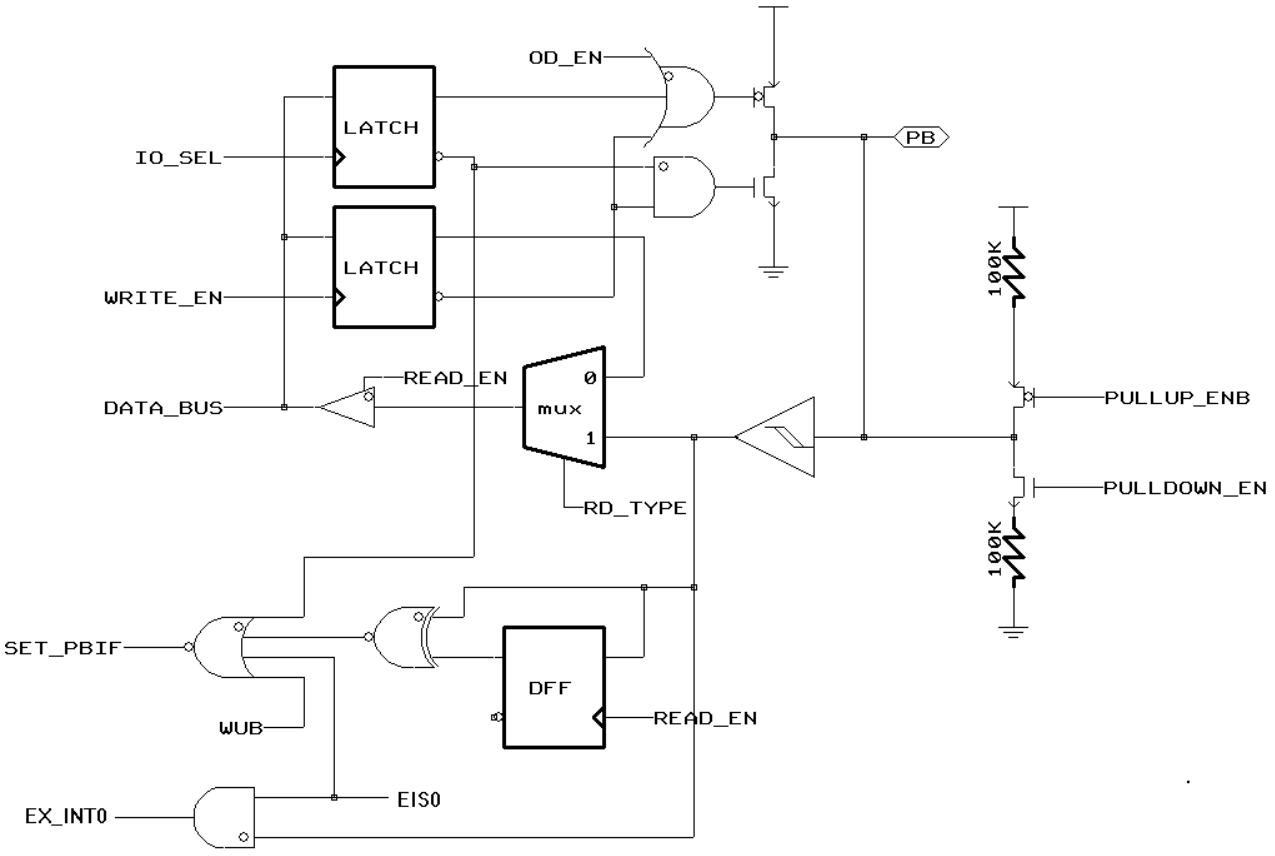


图 10　PB0 引脚结构框图

IO_SEL：设定引脚为输入或输出口。

WRITE_EN：将数据写入引脚。

READ_EN：读取引脚状态。

OD_EN：开启开漏。

PULLUP100K_ENB：开启内部上拉电阻。

PULLDOWN_EN：开启内部下拉电阻。

RD_TYPE：选择读取脚位或数据锁存器。

EIS1：使能外部中断 1 功能。

INTEDG[3:2]：选择外部中断 1 触发沿。

EX_INT1：外部中断 1 信号。

WUB：使能PB唤醒功能。

SET_PBIF：PB唤醒标志。

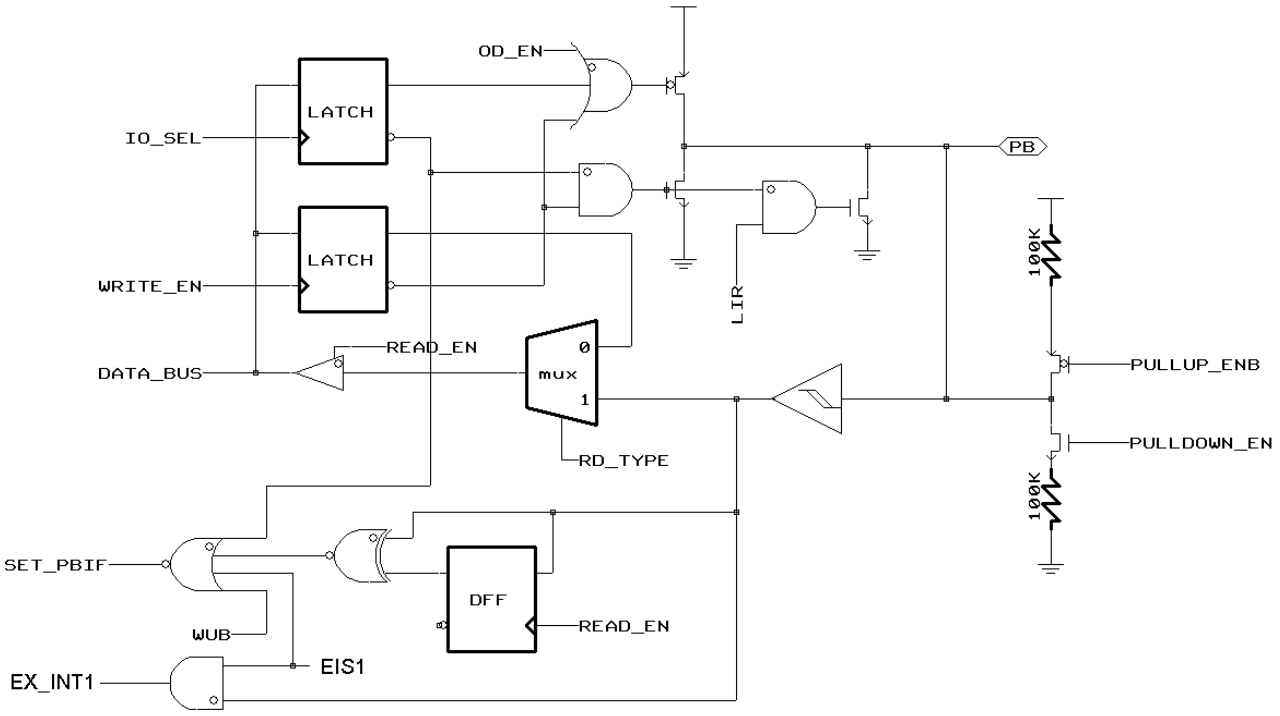


图 11　PB1 引脚结构框图

IO_SEL：设定引脚为输入或输出口。

WRITE_EN：将数据写入引脚。

READ_EN：读取引脚状态。

OD_EN：开启开漏。

PULLUP100K_ENB：开启内部上拉电阻。

PULLDOWN_EN：开启内部下拉电阻。

RD_TYPE：选择读取脚位或数据锁存器。

WUB：使能PB唤醒功能。

SET_PBIF：PB唤醒标志。

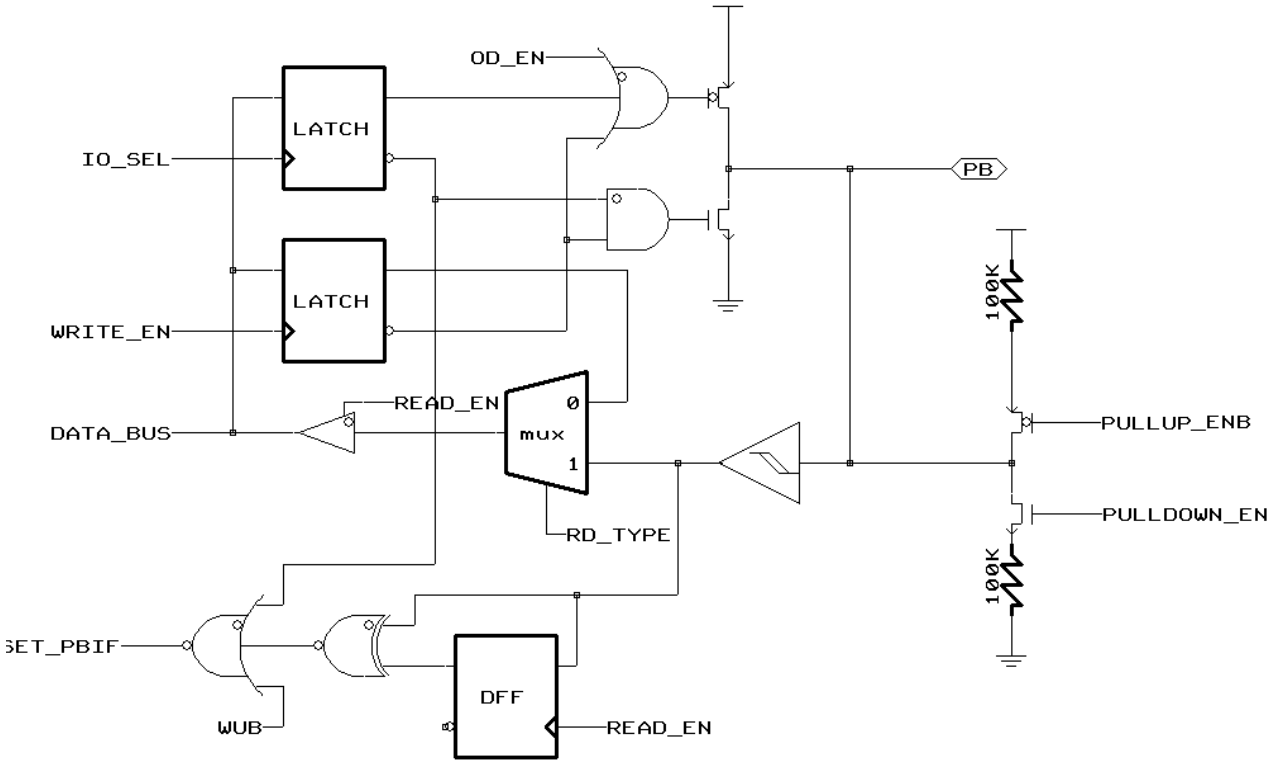


图 12　PB2 引脚结构框图

IO_SEL：设定引脚为输入或输出口。

WRITE_EN：将数据写入引脚。

READ_EN：读取引脚状态。

OD_EN：开启开漏。

PULLUP100K_ENB：开启内部上拉电阻。

PULLDOWN_EN：开启内部下拉电阻。

RD_TYPE：选择读取脚位或数据锁存器。

WUB：使能PB唤醒功能。

SET_PBIF：PB唤醒标志。

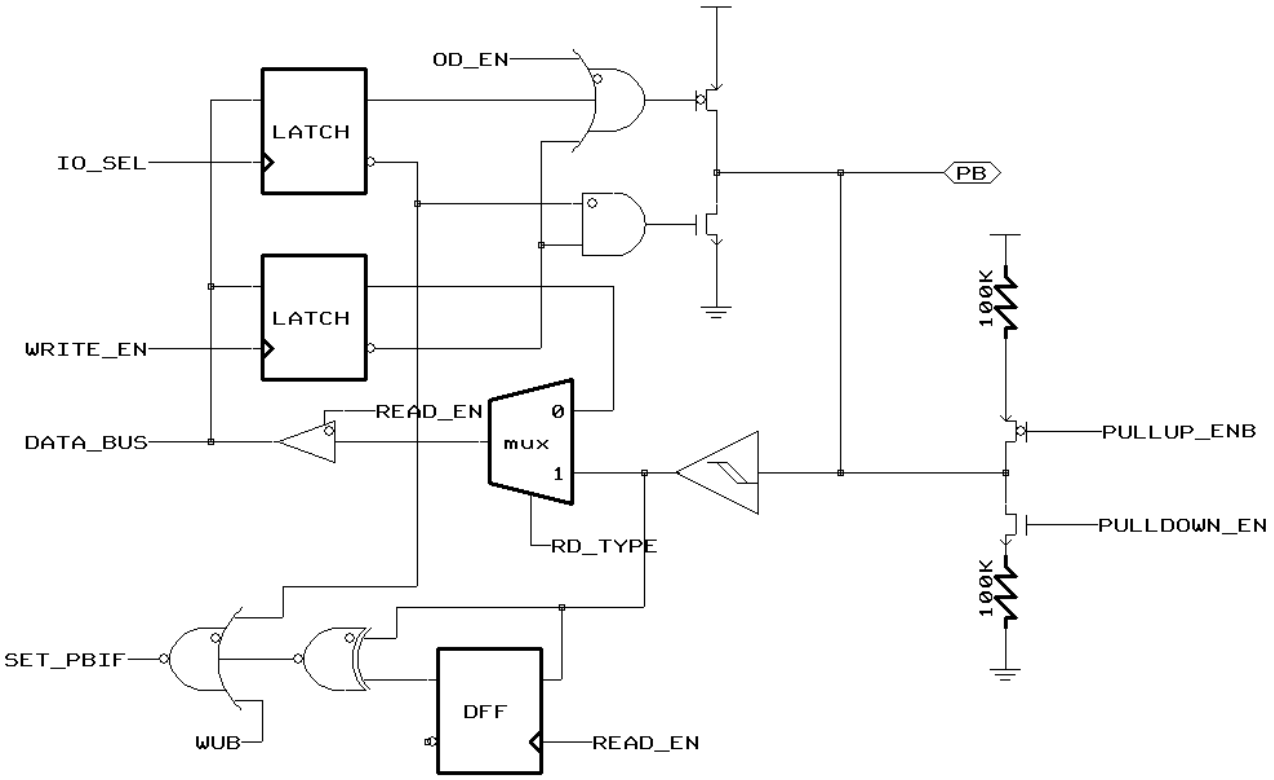


图 13　PB3 引脚结构框图

IO_SEL：设定引脚为输入或输出口。

WRITE_EN：将数据写入引脚。

READ_EN：读取引脚状态。

OD_EN：开启开漏。

PULLUP100K_ENB：开启内部上拉电阻。

RD_TYPE：选择读取脚位或数据锁存器。

WUB：使能PB唤醒功能。

SET_PBIF：PB唤醒标志。

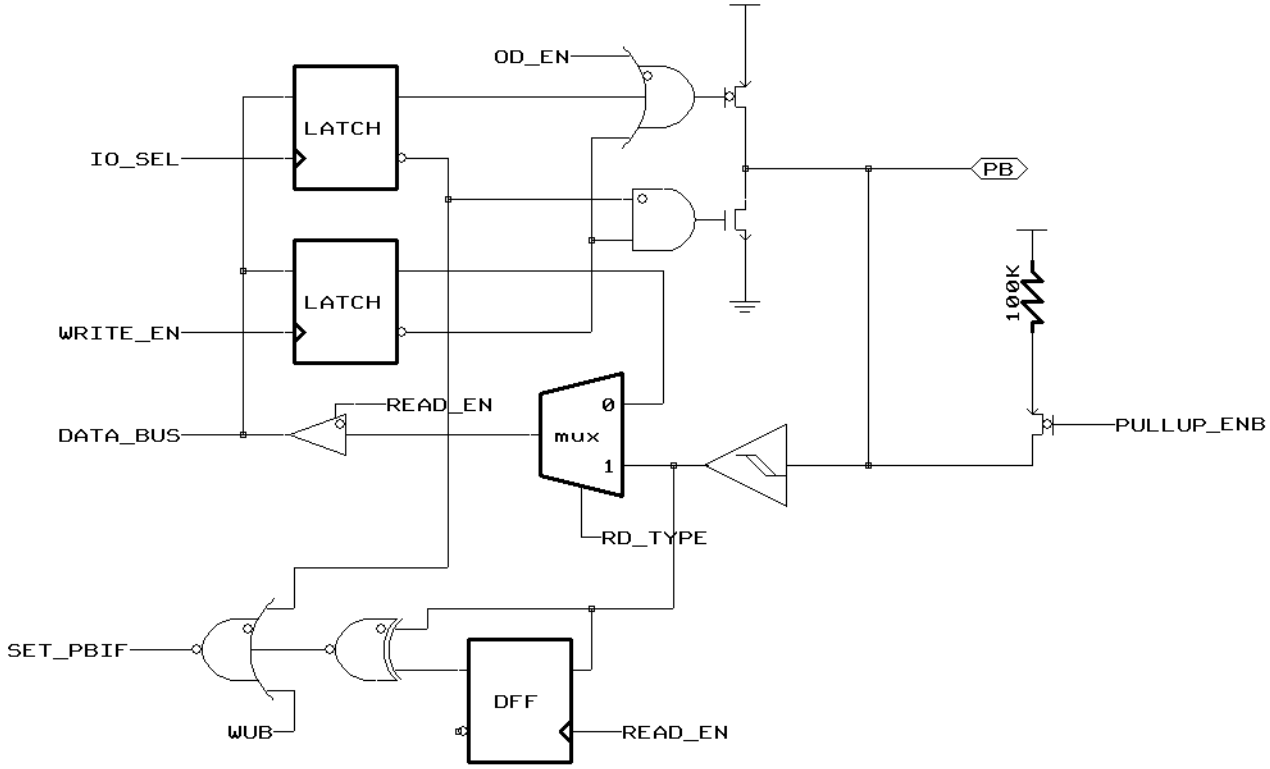


图 14　PB4，PB5 引脚结构框图

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_CKI0 或低频震荡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_CKI0 会被当作定时器 0 时钟源。当T0CS是 1 且LCK_TM0 为 1，会选择低频震荡I_LRC/E_LXT当作定时器 0 时钟源。汇总成表格如下。（也请参考表 16）

定时器 0 时钟源	T0CS	LCKTM0	定时器 0 来源	低频震荡
Instruction clock	0	X	X	X
EX_CKI0	1	0	X	X
		X	0	
E_LXT	1	1	1	1
I_LRC	1	1	1	0

表 16　定时器 0 时钟源摘要

寄存器T0CE（T0MD[4]）可决定EX_CKI0 或I_LRC/E_LXT的时钟触发沿选择。当T0CE是 1，EX_CKI0 或I_LRC/E_LXT的下降沿将让定时器 0 计数加一。当T0CE是 0，EX_CKI0 或I_LRC/E_LXT的上升沿将让定时器 0 计数加一。

如果寄存器PS0WDT（T0MD[3]）为 0，定时器 0 时钟源可以由预分频器 0 所分频，预分频器 0 会被指定到定时器 0，且会在PS0WDT设为 0 时清除Timer0 與寄存器PS0CV。寄存器PS0SEL[2:0] （T0MD[2:0]）决定预分频器 0 的预分频比，其数值从 1:2 到 1:256。

定时器 0 时钟源默认为指令时钟。如果外部时钟脚EX_CKI0 或低频震荡I_LRC/E_LXT被用来当作定时器 0 时钟源，用户必须注意分频后的频率不能超过指令时钟，否则会导致错误计数。当I_LRC/E_LXT同时被当作定时器 0 时钟源与指令时钟，用户必须指定预分频器 0 到定时器 0，且须注意预分频器 0 的预分频比不得小于 4。当配置字节设定为异步（Async.），定时器 0 时外部钟源EX_CKI0 频率就可高于指令时钟。

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：

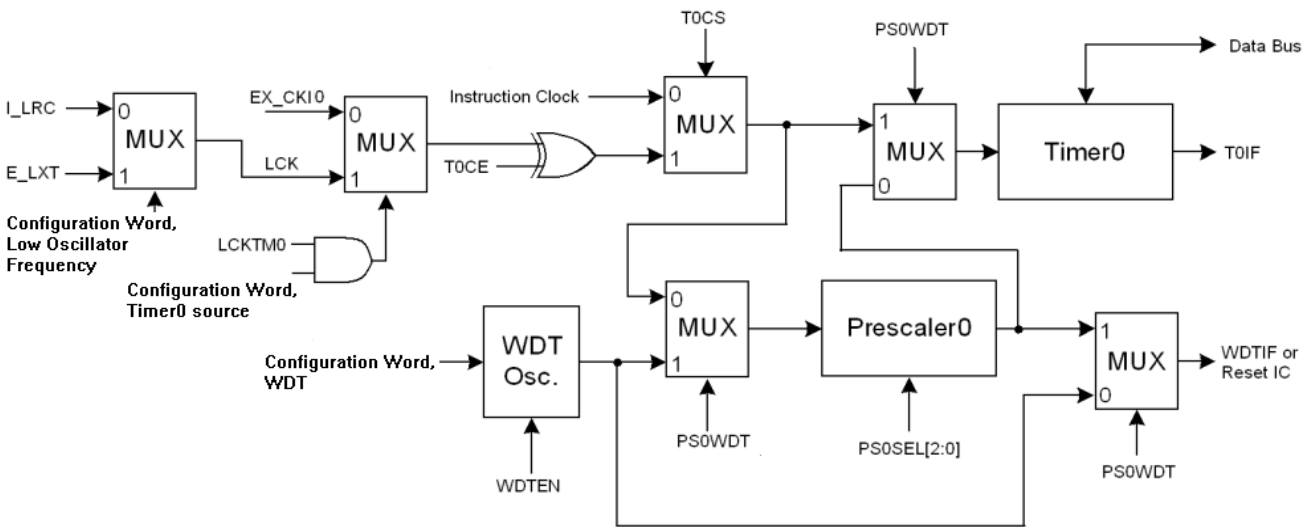


图 15　定时器 0 与 WDT 结构框图

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。写入定时器 1 高 2 位（TMRH[5:4]）再写入TMR1[7:0]时，就可更新定时器 1 重载缓存寄存器。当T1EN=0 时，定时器 1 重载缓存寄存器会立即写入定时器 1。当T1EN=1 时，会等到定时器 1 下溢后，定时器 1 重载缓存寄存器才会写入定时器 1。读取寄存器TMR1[9:0]会显示定时器 1 目前计数数值的内容。

定时器 1 的结构框图如下图所示：

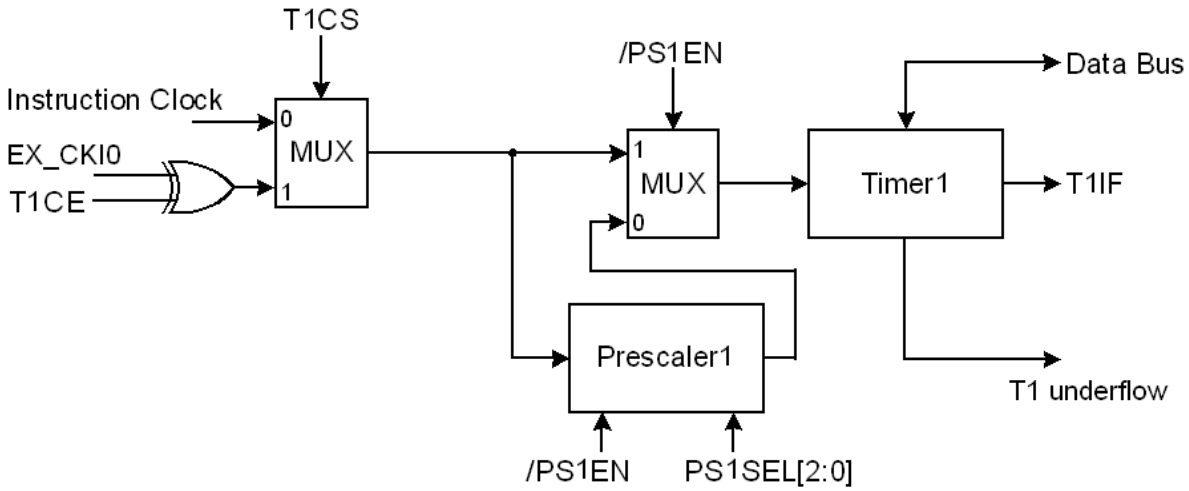


图 16　定时器 1 结构框图

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_CKI0。当T1CS为 0，指令时钟会被选择当做时钟源。当T1CS为 1，则是EX_CKI0 当做时钟源。当EX_CKI0 被选取，寄存器控制位T1CE （T1CR2[4]）可决定EX_CKI0 的时钟触发

沿。当T1CE是 1，EX_CKI0 的下降沿将让定时器 1 计数减一。当T1CE是 0，EX_CKI0 的上升沿将让定时器 1 计数减一。定时器 1 时钟源可以由预分频器 1 所分频。寄存器/PS1EN（T1CR2[3]）为 0，可开启预分频器 1。寄存器PS1SEL[2:0]（T1CR2[2:0]）可以决定其预分频比从 1:2 到 1:256。预分频器 1 的目前数值可以由读取寄存器PS1CV取得。

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

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

定时器 1 时序图如下图所示：

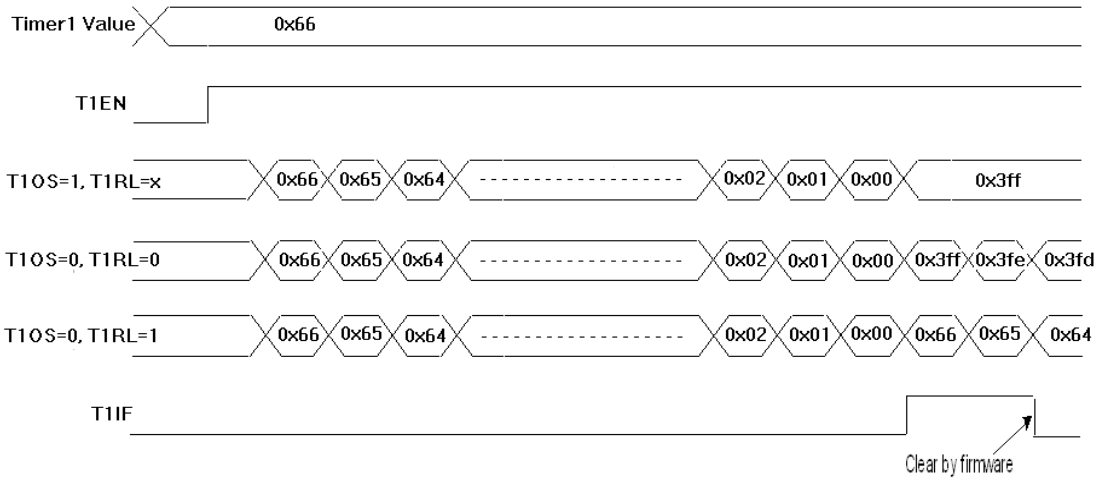


图 17　定时器 1 时序图

当寄存器PWM1OEN（T1CR1[7]）设定为 1，PB3 或PA4 为PWM1 输出。当PWM1OEN为 1，PB3 或PA4 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 為低电平有效输出；PWM1OAL为 0，PWM1 為高电平有效输出。

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[9:0]为 0，PWM1 无法输出占空比。当PWM1DUTY[9:0]为 0x3FF，PWM1 将输出 1023/1024 的占空比（当PWM1OAL为 0）。帧率是由TMRH[5:4] +TMR1[7:0]初始值所决定。因此，PWM1DUTY[9:0]数值必须小于或等于TMR1[9:0]。先写入PWM1 高 2 位（TMRH[1:0]） 再写入PWM1DUTY[7:0]时，就可更新PWM1DUTY[9:0]重载缓存寄存器。等到定时器 1 下溢后，PWM1DUTY[9:0]重载缓存寄存器才会写入PWM1DUTY[9:0]。

PWM1 的结构框图如下：

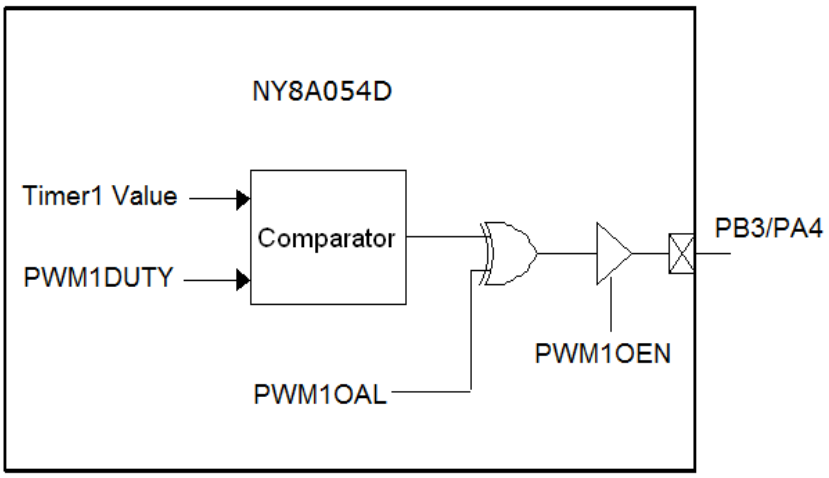


图 18　PWM1 结构框图

当寄存器BZ1EN（BZ1CR1[7]）设定成 1 且使能配置字节，PB3 为蜂鸣器 1 输出。当BZ1EN设定为 1，PB3 会自动成为输出脚。BZ1 的频率是由寄存器BZ1FSEL[3:0]（BZ1CR[3:0]）决定，可以选择从定时器 1 输出或预分频器 1 输出。当BZ1FSEL[3]为 0，预分频器 1 输出被选择来产生BZ1 输出。当BZ1FSEL[3]为 1，定时器 1 输出被选来产生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：

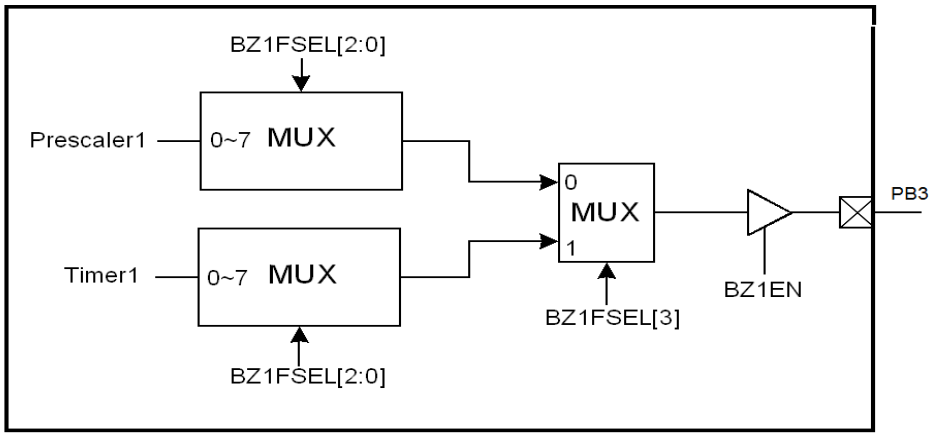


图 19　蜂鸣器 1 结构框图

注意：PB3 复用引脚输出优先级为 PWM1 输出 > 蜂鸣器 1 输出。

3.8　PWM2

当寄存器PWM2OEN（P2CR1[7]）设定为 1，PB2 或PA0 为PWM2 输出。当PWM2OEN为 1，PB2 或PA0 会自动成为输出脚。PWM2 输出的有效状态是由寄存器PWM2OAL（P2CR1[6]）决定。当PWM2OAL为 1，PWM2 为低电平有效输出；PWM2OAL为 0，PWM2 为高电平有效输出。

PWM2 的占空比与帧率皆可编程的。占空比是由寄存器TMRH[3:2]和PWM2DUTY[7:0]决定。当PWM2DUTY[9:0]为 0，PWM2 无法输出占空比。当PWM2DUTY[9:0]为 0x3FF，PWM2 将输出 1023/1024 的占空比（當PWM2OAL为 0）。帧率是由TMRH[5:4] +TMR1[7:0]初始值所决定。因此，PWM2DUTY[9:0]数值必须小于或等于TMR1[9:0]。

先写入PWM2 高 2 位（TMRH[3:2]）再写入PWM2DUTY[7:0]时，就可更新PWM2DUTY[9:0]重载缓存寄存器。等到定时器 1 下溢后，PWM2DUTY[9:0]重载缓存寄存器才会写入PWM2DUTY[9:0]。因為PWM2 與PWM1 共享定時器 1，所以使用者設定P2CR1 與T1CR1 中的T1EN、T1RL、T1OS值必須一致才能成功輸出PWM2。

PWM2 的结构框图如下：

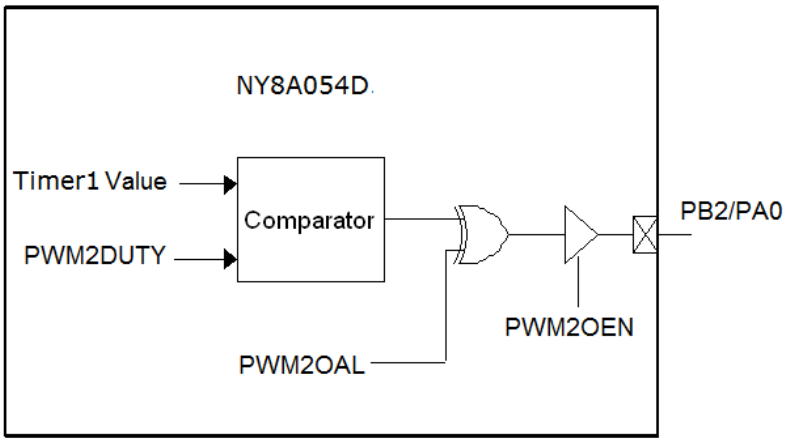


图 20　PWM2 结构框图

3.9　定时器 3/PWM3

定时器 3 是具有预分频器 3 的 10 位下数定时器，其预分频比是可编程的。定时器 3 的输出可以被用于产生PWM3输出。写入定时器 3 高 2 位（TM34RH[5:4]）再写入TMR3[7:0]时，就可更新定时器 3 重载缓存寄存器。当T3EN=0时，定时器 3 重载缓存寄存器会立即写入定时器 3。当T3EN=1 时，会等到定时器 3 下溢后，定时器 3 重载缓存寄存器才会写入定时器 3。读取寄存器TMR3[9:0]会显示定时器 3 目前计数数值的内容。

定时器 3 的结构框图如下图所示：

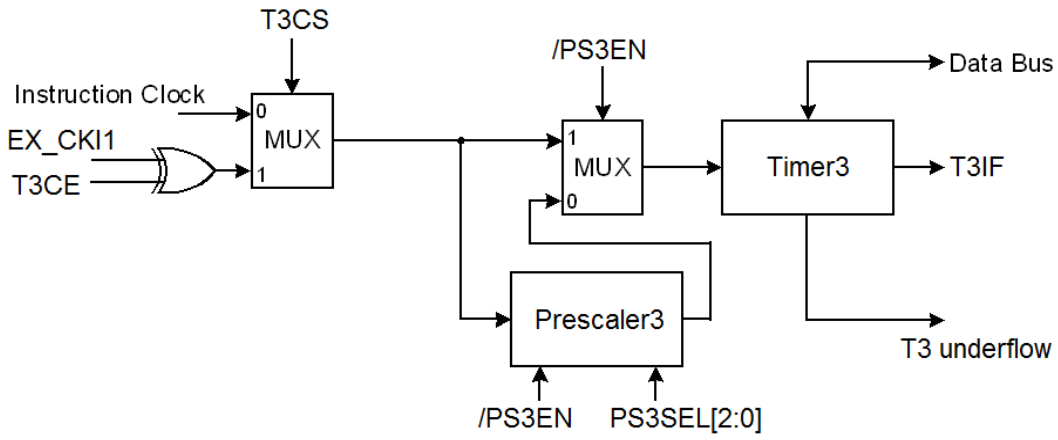


图 21　定时器 3 结构框图

定时器 3 的操作可以由寄存器T3EN（T3CR1[0]）开启或关闭。开启定时器 3 后，寄存器T3CS（T3CR2[5]）可决定时钟源是指令时钟F_{INST}或外部时钟EX_CKI1。当T3CS为 0，指令时钟会被选择当做时钟源。当T3CS为 1，

则是EX_CKI1 当做时钟源。当EX_CKI1 被选取，寄存器控制位T3CE（T3CR2[4]）可决定EX_CKI1 的时钟触发沿。当T3CE是 1，EX_CKI1 的下降沿将让定时器 3 计数减一。当T3CE是 0，EX_CKI1 的上升沿将让定时器 3 计数减一。定时器 3 时钟源可以由预分频器 3 所分频。寄存器 /PS3EN（T3CR2[3]）为 0，可开启预分频器 3。寄存器 PS3SEL[2:0]（T3CR2[2:0]）可以决定其预分频比从 1:2 到 1:256。预分频器 3 的目前数值可以由读取寄存器PS3CV取得。

定时器 3 提供两种计数模式:单次计数与连续计数。当寄存器T3OS（T3CR1[2]）为 1，即为单次计数模式。定时器 3 从储存在寄存器TMR3[9:0]的初始值下数到 0x00，当下溢发生时，定时器 3 停止计数。当寄存器T3OS（T3CR1[2]）为 0，即为连续计数模式。当下溢发生，寄存器T3RL（T3CR1[1]）会决定计数的初始值。当T3RL为 1，定时器 3 从寄存器TMR3[9:0]重新载入数值作为初始值并继续下数。当T3RL为 0，定时器 3 以 0x3FF作为初始值并继续下数。

当定时器 3 下溢，寄存器T3IF（INTE2[4]）会被设定为 1，标明定时器 3 发生下溢中断。如果寄存器T3IE（INTE2[0]）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T3IF，T3IF才会被清除。

定时器 3 时序图如下图所示：

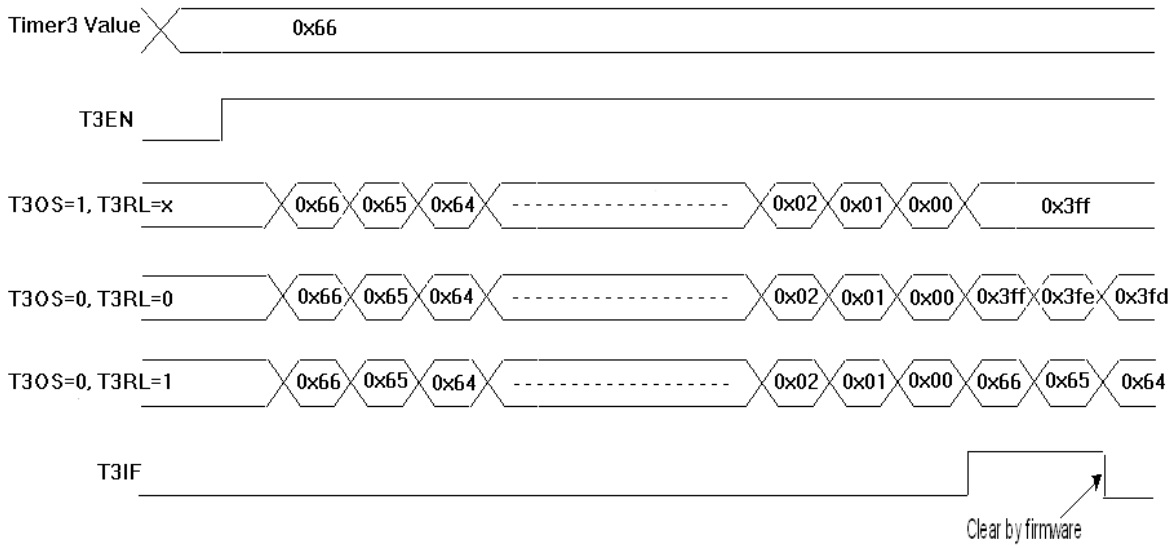


图 22　定时器 3 结构框图

当寄存器PWM3OEN（T3CR1[7]）设定为 1，PA3 或PB4 为PWM3 输出。当PWM3OEN为 1，PA3 或PB4 会自动成为输出脚。PWM3 输出的有效状态是由寄存器PWM3OAL（T3CR1[6]）决定。当PWM3OAL为 1，PWM3 為低电平有效输出；PWM3OAL为 0，PWM3 為高电平有效输出。

PWM3 的占空比与帧率皆可编程的。占空比是由寄存器TM34RH[1:0]和PWM3DUTY[7:0]决定。先寫当PWM3DUTY[9:0]为 0，PWM3 无法输出占空比。当PWM3DUTY[9:0]为 0x3FF，PWM3 将输出 1023/1024 的占空比（當PWM3OAL为 0）。帧率是由TM34RH[5:4] +TMR3[7:0]初始值所决定。因此，PWM3DUTY[9:0]数值必须小于或等于TMR3[9:0]。先写入PWM3 高 2 位（TM34RH[1:0]）再写入PWM3DUTY[7:0]时，就可更新PWM3DUTY[9:0]重载缓存寄存器。等到定时器 3 下溢后，PWM3DUTY[9:0]重载缓存寄存器才会写入PWM3DUTY[9:0]。

PWM3 的结构框图如下：

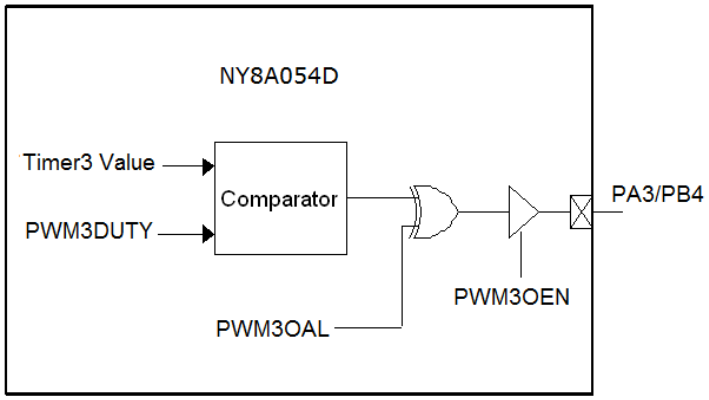


图 23　PWM3 结构框图

3.10 PWM4

当寄存器PWM4OEN（P4CR1[7]）设定为 1，PB5 或PA2 为PWM4 输出。当PWM4OEN为 1，PB5 或PA2 会自动成为输出脚。PWM4 输出的有效状态是由寄存器PWM4OAL（P4CR1[6]）决定。当PWM4OAL为 1，PWM4 為低电平有效输出；PWM4OAL为 0，PWM4 為高电平有效输出。

PWM4 的占空比与帧率皆可编程的。占空比是由寄存器TM34RH[3:2]和PWM4DUTY[7:0]决定。当PWM4DUTY[9:0]为 0，PWM4 无法输出占空比。当PWM4DUTY[9:0]为 0x3FF，PWM4 将输出 1023/1024 的占空比（當PWM4OAL为 0）。帧率是由TM34RH[5:4] +TMR3[7:0]初始值所决定。因此，PWM4DUTY[9:0]数值必须小于或等于TMR3[9:0]。先写入PWM4 高 2 位（TM34RH[3:2]）再写入PWM4DUTY[7:0]时，就可更新PWM4DUTY[9:0]重载缓存寄存器。等到定时器 3 下溢后，PWM4DUTY[9:0]重载缓存寄存器才会写入PWM4DUTY[9:0]。因为PWM4 与PWM3 共享定时器 3，所以使用者设定P4CR1 与T3CR1 中的T3EN、T3RL、T3OS值必须一致才能成功输出PWM4。

PWM4 的结构框图如下：

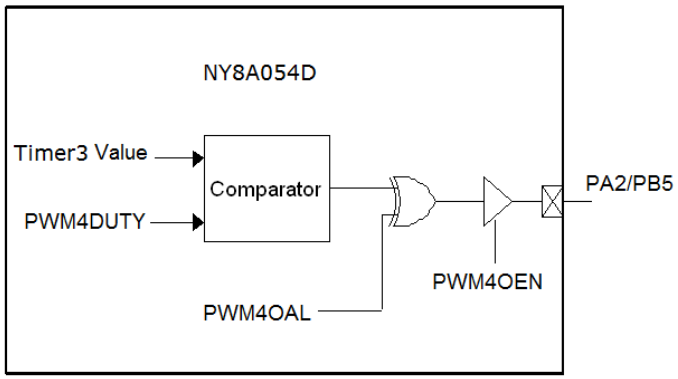


图 24　PWM4 结构框图

3.11 PWM5

当寄存器PWM5OEN（P5CR1[7]）设定为 1，PB0 或PA1 为PWM5 输出。当PWM5OEN为 1，PB0 或PA1 会自动成为输出脚。PWM5 输出的有效状态是由寄存器PWM5OAL（P5CR1[6]）决定。当PWM5OAL为 1，PWM5 為低电平有效输出；PWM5OAL为 0，PWM5 為高电平有效输出。

PWM5 的占空比与帧率皆可编程的。占空比是由寄存器PWM5RH[1:0]和PWM5DUTY[7:0]决定。当PWM5DUTY[9:0]为 0，PWM5 无法输出占空比。当PWM5DUTY[9:0]为 0x3FF，PWM5 将输出 1023/1024 的占空比（當PWM5OAL为 0）。帧率是由TM34RH[5:4] +TMR3[7:0]初始值所决定。因此，PWM5DUTY[9:0]数值必须小于或等于TMR3[9:0]。先写入PWM5 高 2 位（PWM5RH[1:0]）再写入PWM5DUTY[7:0]时，就可更新PWM5DUTY[9:0]重载缓存寄存器。等到定时器 3 下溢后，PWM5DUTY[9:0]重载缓存寄存器才会写入PWM5DUTY[9:0]。因為PWM5 與PWM3 共享定時器 3，所以使用者設定P5CR1 與T3CR1 中的T3EN、T3RL、T3OS值必須一致才能成功輸出PWM5。

PWM5 的结构框图如下：

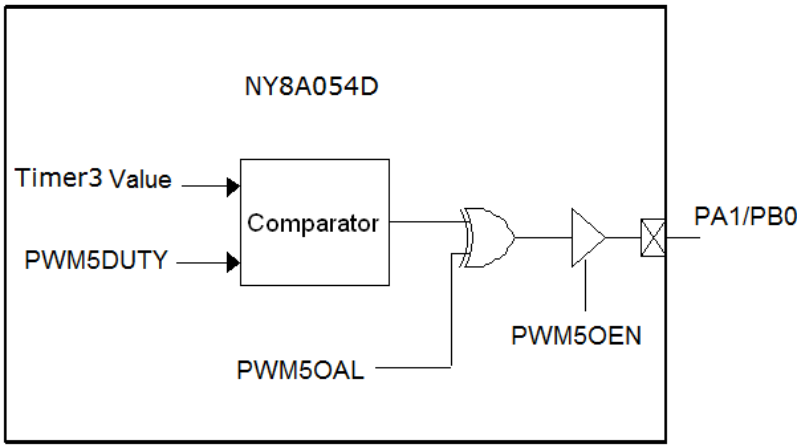


图 25　PWM5 结构框图

3.12 电阻/频率转换器模式（RFC）

NY8A054D内置RFC功能，当开启RFC功能（RFCEN=1），选择的RFC输入引脚的电压位准将会控制定时器 1 的计数行为。只要RFC输入引脚电压低于V_{IL}，定时器 1 将会持续计数，等到RFC输入引脚电压高于V_{IH}，定时器 1 才会停止计数。PSEL3~0 用來選擇RFC输入引脚。

如下图，当PSEL3~0=0x01，PA1 为RFC输入引脚。首先设置PA1 输出 0（低于V_{IL}），接着设置TMR1[9:0]初始值后将PA1 设置为输入引脚，定时器 1 会开始下数。这时RC电路开始对PA1 引脚充电。当PA1 引脚电压高于V_{IH}时，定时器 1 会停止计数。定时器 1 相当于纪录RC电路充电时间。

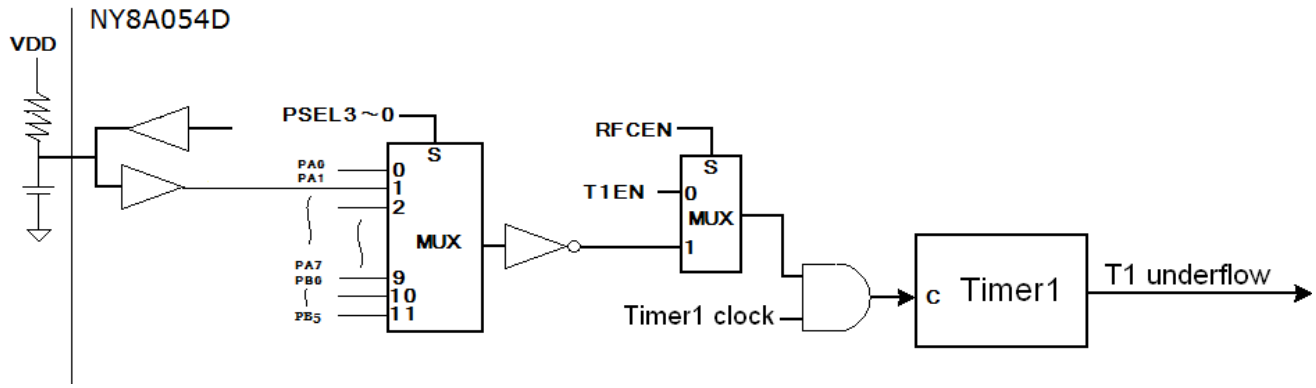


图 26　RFC 结构框图

3.13 IR Carrier

寄存器IREN（IRCR[0]）被设定为 1 后，PB1 为红外线载波输出，而PB1 会自动成为输出脚。当IREN清零，PB1 将会成为一般I/O脚。

红外线载波频率是由寄存器IRF57K（IRCR[1]）所选择。当IRF57K为 1，红外线载波频率是 57KHz。当IRF57K为 0，频率是 38KHz。红外线载波输出是以高速振荡时钟除频得来。如果使用外置晶振 3.58MHz当作高速振荡时钟来源，寄存器IROSC358M（IRCR[7]）请设为 1。如果使用外置晶振 455KHz当作高速振荡时钟来源，寄存器IROSC358M（IRCR[7]）请设为 0。若选择 I_HRC当作高速振荡时钟来源，此時IR模块时钟永远是 4MHz，IROSC358M（IRCR[7]）将被忽略。

红外线载波的极性会根据PB1 输出数据所决定。当寄存器IRCSEL（IRCR[2]）为 1 且PB1 输出数据为 0，红外线载波将由PB1 輸出。当寄存器IRCSEL（IRCR[2]）为 0 且PB1 输出数据为 1，红外线载波将由PB1 輸出。红外线载波的极性如下图所示：

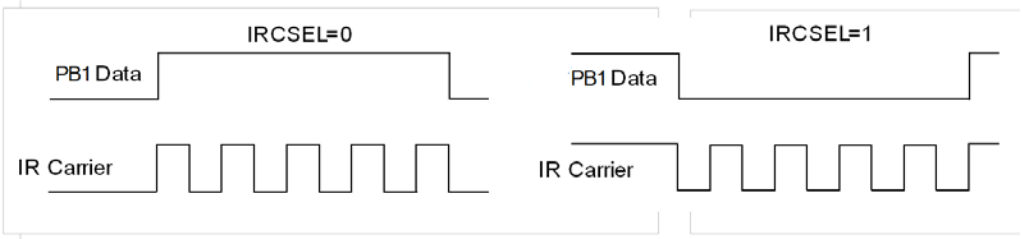


图 27　红外线载波的极性 vs. PB1 数据

3.14 低电压侦测（LVD）

NY8A054D内置准确的低电压侦测电路来侦测VDD电压水平。将LVDEN设为 1（寄存器PCON[5]）后, 当VDD电压低于下表LVDS[2:0]选择的电压值时，读取LVDOUT（寄存器PCON1[6]）会得到 0。如果开启LVD中断使能位且GIE=1 时，LVD中断标志位将会被设置为 1，程序将跳入中断子程序。

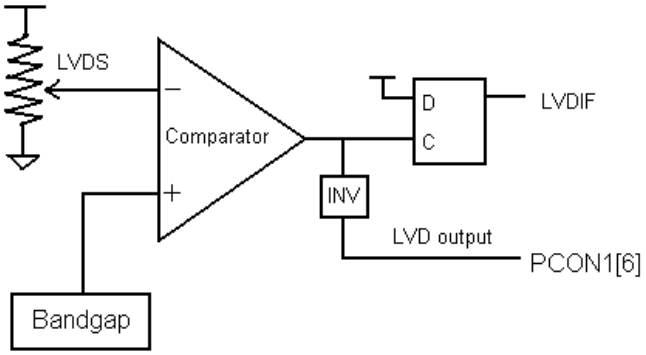


图 28　LVD结构框图

LVDS[2:0]	Voltage
000	2.0V
001	2.2V
010	2.4V
011	2.7V
100	3.0V
101	3.3V
110	3.6V
111	4.3V

表 17　选择LVD电压

3.15 电压比较器

NY8A054D内置一组电压比较器。比较器的正输入源与负输入源和GPIO口复用，内部参考电压只能在P2V模式下提供给比较器的负输入源使用。

CMPEN（寄存器PCON[2]）用来开启或关闭比较器，在睡眠模式（Halt mode）中比较器将自动关闭。

NY8A054D的电压比较器有P2V或P2P两种模式，由VS[3:0]（寄存器CMPCR[3:0]）来选择。当VS[3:0]=0，比较器为P2P模式。当VS[3:0]=1~15，比较器为P2V模式。比较器正/负输入源如果为GPIO复用引脚，必须在配置字节（Configuration Word）中勾选为“Comparator Input”。

P2V模式是将内部参考电压（负输入源）与选择的外部模拟输入引脚（正输入源）做电压水平比较。

P2V模式硬件图框入下图所示：

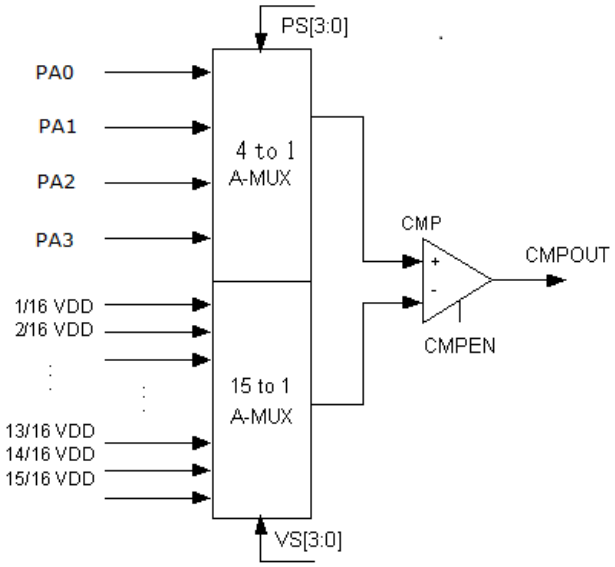


图 29　比较器P2V模式结构框图

在P2V模式中，比较器负输入源由VS[3:0]用来选择内部参考电压 1/16 VDD ~ 15/16 VDD。

VS[3：0]	内部参考电压
0000	P2P mode
0001	1/16 V_{DD}
0010	2/16 V_{DD}
0011	3/16 V_{DD}
0100	4/16 V_{DD}
0101	5/16 V_{DD}
0110	6/16 V_{DD}
0111	7/16 V_{DD}
1000	8/16 V_{DD}
1001	9/16 V_{DD}
1010	10/16 V_{DD}
1011	11/16 V_{DD}
1100	12/16 V_{DD}
1101	13/16 V_{DD}
1110	14/16 V_{DD}
1111	15/16 V_{DD}

表 18　选择P2V模式中比较器负输入源的内部参考电压

在P2V模式中，比较器正输入源由PS[3:0]（寄存器CMPCR[7:4]）用来选择PA3~0 其中一根引脚。

PS[3：0]	引脚选择
0000	PA0
0001	PA1
0010	PA2
0011	PA3
0100 ~ 1111	-

表 19　P2V模式中比较器正输入源引脚选择

P2P模式中，比较器正/负输入源都为外部引脚。须设置VS[3:0]=0，由PS[3:0]来选择 4 种外部输入引脚组合。

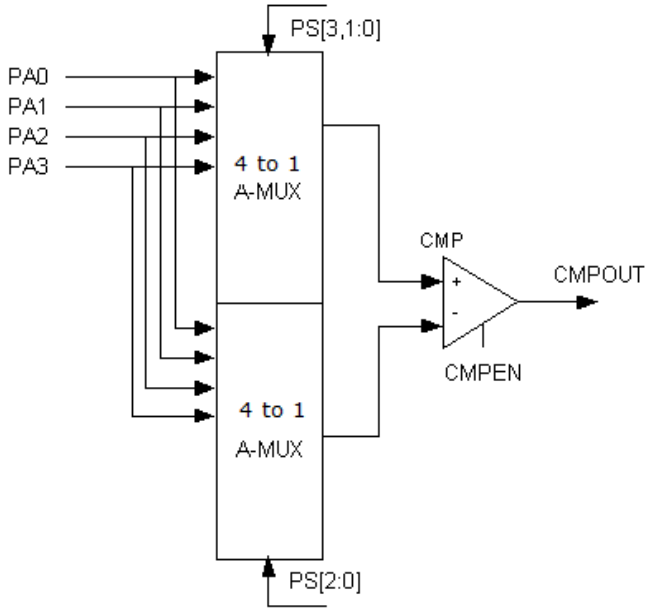


图 30　比较器P2P模式结构框图

PS[3：0]	负输入源	正输入源
0000	PA0	PA1
0001	PA1	PA0
0010	PA2	PA3
0011	PA3	PA2
0100~1111	-	-

表 20　比较器P2P模式输入源引脚选择

有三种方式可以取得比较器的比较结果。一是通过硬件中断设置，二是轮询 CMPOUT（寄存器 OSCCR[7]），是由 CMPO 引脚（PB3）输出。

使用比较器硬件中断功能时，须设置 CMPEN=1 与 CMPIE=1。先读取 CMPOUT（寄存器 OSCCR[7]）来清除上一次比较器比较结果，再清除中断标志位 CMPIF=0。等到下一次比较器输出翻转时，CMPIF 中断标志位将会被

设置为 1。

设置 CMPOE（寄存器 OSCCR[6]）为 1，PB3 引脚将实时输出比较器结果，此时 PWM1 会自动关闭。

3.16 看门狗定时器（WDT）

NY8A054D中有独立振荡器被WDT所使用。由于该振荡器与其它振荡电路无关，故在待机模式和睡眠模式中WDT仍能继续工作。

WDT能被配置字节开启或关闭。当WDT被配置字节开启时，仍然可以通过寄存器WDTEN（PCON[7]）位来开启/关闭。此外，WDT上溢后可由配置字节决定的复位NY8A054D或发出的中断请求。同时，在WDT上溢后，寄存器/TO（STATUS[4]）位将被清除为 0。

WDT上溢的时基由配置字节决定，可以是 3.5 毫秒、15 毫秒、60 毫秒或 250 毫秒。如果将预分频器 0 分配给WDT，则可以延长上溢周期。通过将 1 写入寄存器PS0WDT位，预分频器 0 将分配给WDT。预分频器 0 对WDT的分频比由寄存器PS0SEL [2:0]位决定。如果WDT上溢将复位NY8A054D，分频速率从 1:1 到 1:128。如果选为WDT中断时，则分频速率从 1:2 到 1:256。

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

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

3.17 中断

NY8A054D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。硬件中断则有以下九种：

- Timer0 上溢中断。
- Timer1 下溢中断。
- Timer3 下溢中断。
- WDT中断。
- PA/PB输入状态改变中断。
- 外部中断 0 输入。
- 外部中断 1 输入。
- 低电压侦测中断。
- 比较器输出翻转中断。

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

执行完指令INT后，无论GIE是置 1 还是清除为零，下一条指令都将从地址 0x001 读取。同时，GIE将由NY8A054D自动清除为零，这将防止嵌套中断的发生。软件中断的中断服务程序最后一条指令必须是RETIE。执行此指令将设置GIE为 1 并返回中断前程序执行序列。

当发生硬件中断时，相应的中断标志位将被设置为 1。该位在程序将 0 写入该位之前不会清除为零。因此，用户可以通过轮呼相应的中断标志位得知哪个硬件引发中断。需注意只有当相应的中断使能位设置为 1 时，才能正确地读取相应的中断标志。如果相应的中断使能位设置为 1，GIE也为 1，将发生硬件中断，下一条指令将从 0x008 执行。同时，NY8A054D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NY8A054D。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 并返回中斷前程序执行序列。

用户应注意ENI指令不能放在RETIE指令之前，因为中断服务程序中的ENI指令将开启嵌套中断，但RETIE指令可能会误清除中断标志。

3.17.1 Timer0 上溢中断

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

3.17.2 Timer1 下溢中断

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

3.17.3 Timer3 下溢中断

Timer3 下溢（从 0x3FF到 0x00），如果T3IE和GIE设置为 1，寄存器T3IF位将被硬件设为 1 并处理此中断请求。

3.17.4 看门狗超时中断

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

3.17.5 PA/PB 输入状态改变中断

当PAx（$0 \le x \le 7$），PBy（$0 \le y \le 5$）设置为输入口且相应的寄存器WUPAx、WUPBx位设置为 1，且如果PABIE和GIE设置为 1，当这些选定输入口上的状态变化时，寄存器PABIF位将被硬件设为 1 并处理此中断请求。需注意当PB0、PB1/PA2 同时设置为状态变化中断和外部中断时，设置EIS0=1 或EIS1=1 将禁止PB0、PB1/PA2 状态变化中断。

3.17.6 外部中断 0 输入

根据EIS0=1 和寄存器INTEDG的配置，如果INT0IE和GIE设置为 1，PB0 引脚上的有效边沿触发会让寄存器INT0IF位将被硬件设为 1 并处理此中断请求。

3.17.7 外部中断 1 输入

根据EIS1=1 和寄存器INTEDG的配置，如果INT1IE和GIE设置为 1，PB1/PA2 引脚上的有效边沿沿触发会让寄存器INT1IF位将被硬件设为 1 并处理此中断请求。

3.17.8 低电压侦测中断

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

3.17.9 比较器输出翻转中断

当比较器输出状态翻转时，如果CMPIE和GIE设置为 1，寄存器CMPIF位将被硬件设为 1 并处理此中断请求。请先读取CMPOUT（寄存器OSCCR[7]）来清除上一次比较器比较结果。

3.18 振荡器配置

因为NY8A054D是双时钟IC，有高振荡时钟（F_{HOSC}）和低振动时钟（F_{LOSC}）可选择作为系统振荡时钟（F_{OSC}）。可用作F_{HOSC}的振荡器有内部高速RC振荡器（I_HRC）、外部高速晶体振荡器（E_HXT）与外部晶体振荡器（E_XT）。可用作F_{LOSC}的振荡器是内部低速RC振荡器（I_LRC）与外部低速晶体振荡器（E_LXT）。

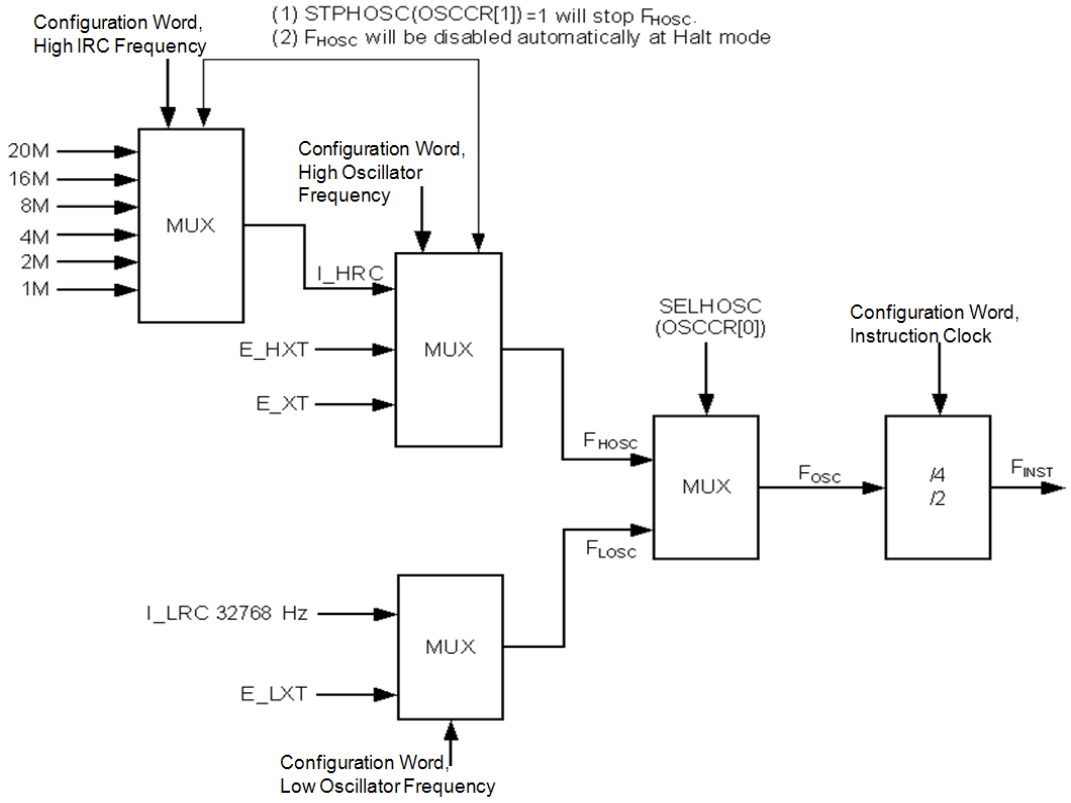


图 31　NY8A054D 振荡配置结构图

系统时钟频率可由配置字节设置为I_HRC，可以选择的频率为 1M、2M、4M、8M、16M或 20MHz。此时PA7 引脚也可以输出指令时钟。外部晶体振荡器连接引脚为PA6（Xin）与PA7（Xout）。如果外部晶体振荡器是 8MHz 到 20MHz，配置字节请选择E_HXT。如果外部晶体振荡器是 455KHz到 6MHz，配置字节请选择E_XT。当选择E_HXT或E_XT时，PA6/PA7 不能当作一般I/O口。它们必须当作晶振的输出（PA7）引脚与输入（PA6）引脚。

当配置字节选择I_LRC时，其频率约为 32768Hz。如果需要使用低速晶体振荡器，只能是 32768Hz晶体振荡器，配置字节请选择E_LXT。当选择E_LXT，PA6/PA7 不能当作一般I/O口。它们必须当作晶振的输出（PA7）引脚与输入（PA6）引脚。

No.	F_{HOSC}	F_{LOSC}
1	I_HRC	I_LRC
2	E_HXT 或 E_XT	I_LRC
3	I_HRC	E_LXT

表 21　双时钟组合

当使用外部晶体振荡器时，建议在Xin与Xout引脚各自接一颗电容器C1 和C2 到VSS，C1 和C2 数值请参考下表

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
E_LXT	32768	5 ~ 30

表 22　不同外部晶体振荡器频率所推荐的C1 和C2 电容器数值

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

C1=C2=2*CL-Cbt

其中Cbt是NY8A054D内部电容值，大约为 5pF。假设低速晶体振荡器的负载电容值CL=12.5pF，依公式算出的C1=C2=20pF。

I_HRC在 25°C条件下的准确度为 ±1%。

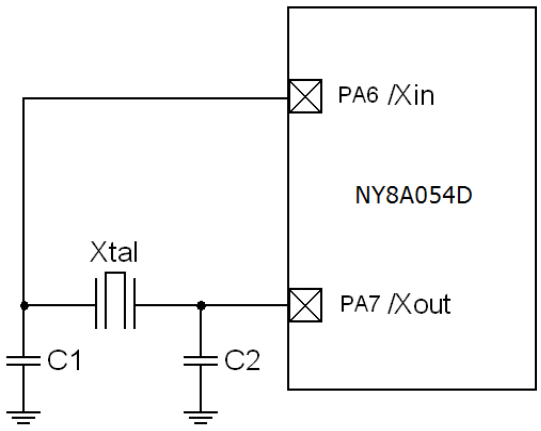


图 32　外部晶体振荡器的硬件连接图

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}，根据配置字节設置，指令时钟可以选择为F_{OSC}/2 或F_{OSC}/4。

3.19 工作模式

NY8A054D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，NY8A054D将停止几乎所有的运作，可由定时器、比较器、LVD、看门狗与外部事件等来唤醒。在睡眠模式下，NY8A054D将睡眠直到外部事件或看门狗定时器来唤醒。

四种工作模式如下图所示。

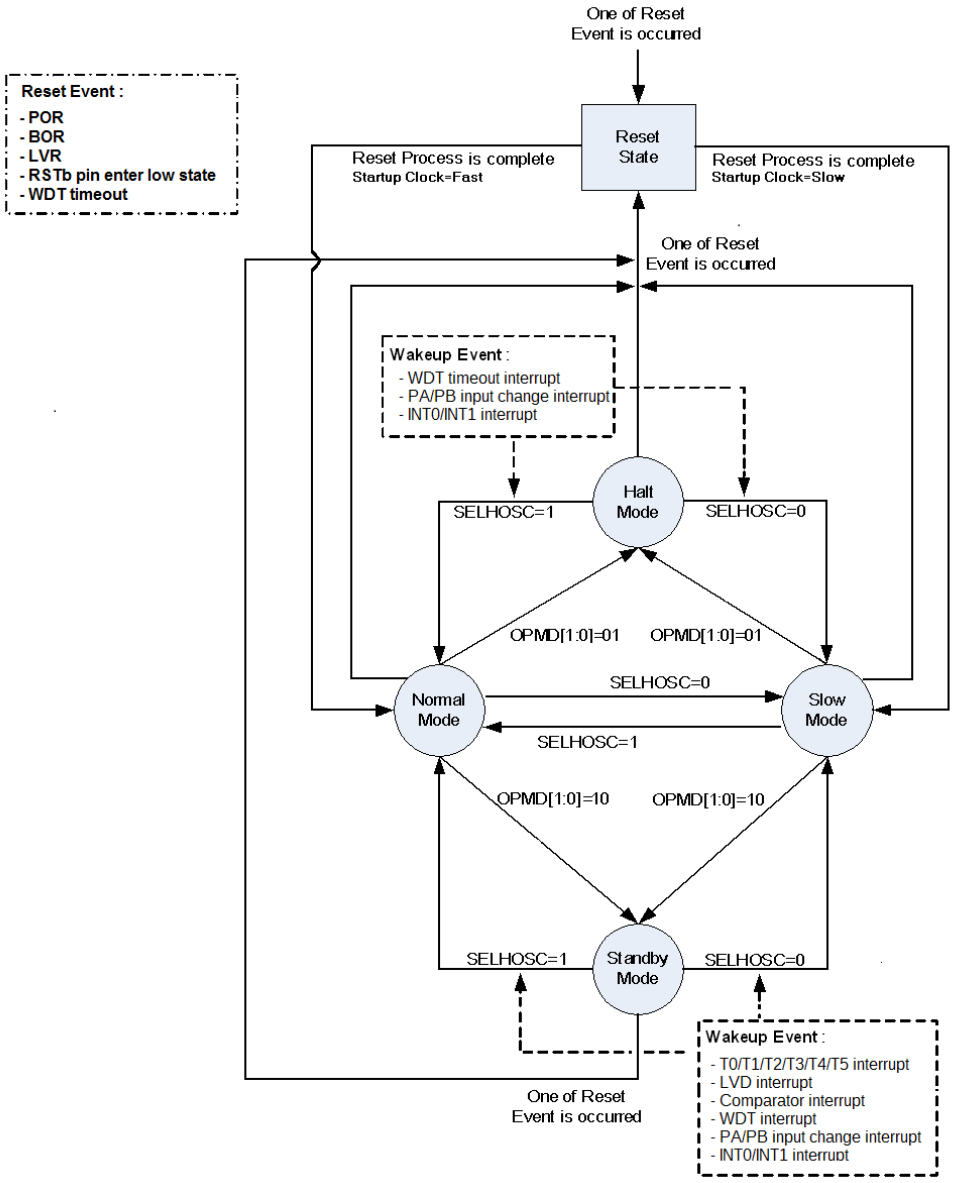


图 33　四种工作模式

3.19.1 正常模式

发生任何复位事件并且复位过程完成后，NY8A054D将在正常模式或慢速模式下开始执行程序。重置过程后选择的模式由启动时钟配置字节决定。如果启动时钟为F_{HOSC}，NY8A054D将进入正常模式，如果启动时钟为F_{LOSC}，NY8A054D将进入慢速模式。在正常模式下，为提供最高性能而以F_{HOSC}作为系统振荡时钟，其功耗在四种操作模式中将是最大的。在上电或任何重置触发器被释放后，待复位程序完成NY8A054D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，NY8A054D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。

3.19.2 慢速模式

通过写 0 至寄存器SELHOSC位，NY8A054D将进入慢速模式。在低速模式下，为节省功耗，F_{LOSC}被选为系统振荡时钟。然而，F_{HOSC}将不会自动被NY8A054D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。

3.19.3 待机模式

通过写入 10b至寄存器OPMD[1:0]，NY8A054D将进入待机模式。然而，在待机模式下，F_{HOSC}不会自动被NY8A054D关闭，用户必须进入先低速模式後写入 1 至寄存器STPHOSC位，以停止F_{HOSC}。部分NY8A054D的硬件功能会被关闭，如T0EN/T1EN/T3EN位被设置为 1 则定时器仍可运作。因此Timer0 / Timer1 / Timer3 溢出后NY8A054D会被唤醒。

- 停止执行指令且一些硬件功能可以根据相应的硬件使能位来开启/关闭。
- 由写入 1 至寄存器STPHOSC位F_{HOSC}可以被关闭。
- F_{LOSC}仍保持运作。
- 如遇以下任一状况IC便能从待机模式唤醒：
  （a）Timer0 上溢中断 / Timer1 下溢中断 / Timer3 下溢中断 （b）看门狗超时中断 （c）PA/PB输入状态改变中断 （d）外部中断 0/1 （e）LVD中断 （f）比较器输出翻转中断。
- 在从待机模式唤醒后，如SELHOSC=1，IC将回到正常模式，如SELHOSC=0 则IC将回到慢速模式。
  不建议在同一时间进入待机模式並改变振荡模式（正常到慢速/慢速到正常）。

3.19.4 睡眠模式

通过执行SLEEP指令或写入 01b至寄存器OPMD[1:0]位，NY8A054D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NY8A054D只能通过一些特殊事件唤醒。因此，睡眠模式是NY8A054D最省电的模式。

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

  注意：您可以在同一指令中更改STPHOSC并进入睡眠模式。
- 不建议改变振荡模式（正常到慢速/慢速到正常），并在同一时间进入待机模式。

3.19.5 唤醒稳定时间

若外部晶体振荡器为系统振荡时钟来源，其睡眠模式的唤醒等待时间为 512*Fosc，若内部RC振荡器为系统振荡时钟来源，其睡眠模式的唤醒等待时间为 16*Fosc，由于待机模式下F_{HOSC}或F_{LOSC}仍在运行，因此无需为待机模式唤醒稳定时间。

在NY8A054D进入待机模式或睡眠模式之前，用户可以执行指令ENI。在唤醒後，NY8A054D将跳转到地址0x008，以便执行中断服务程序。如果在进入待机模式或睡眠模式之前执行DISI指令，则在唤醒后执行下一条指令。

3.19.6 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
计时器 0/1/3	TxEN	TxEN	TxEN	关闭
WDT	配置和WDTEN	配置和WDTEN	配置和WDTEN	配置和WDTEN
其它硬件	硬件使能位	硬件使能位	硬件使能位	全部关闭

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 - Timer3 下溢 - WDT超时 - PA/PB输入狀態改变 -外部中断 0/1 - LVD中断 -比较器中断	- WDT超时 - PA/PB输入狀態改变 - 外部中断 0/1

表 23　工作模式概述

3.20 复位

当以下任一复位事件发生时，NY8A054D将会进入复位状态并开始复位动作：

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

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

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

表 24　/TO和/PD值和相关事件概述

复位事件发生后,NY8A054D将会开始复位进程。无论采用什么样的振荡器，它将等待一定的周期使振荡稳定。这个周期被称为上电复位时间，它由三位配置字节决定，这个时间可能是 140us，4.5ms，18ms，72ms或 288 ms。振荡器稳定后，NY8A054D将等待 Fosc 的 16 个时钟周期（OST，振荡器启动时间）后完成复位。若上电复位时间设为 140us时，将等待 Fosc 的 1 个时钟周期后完成复位。

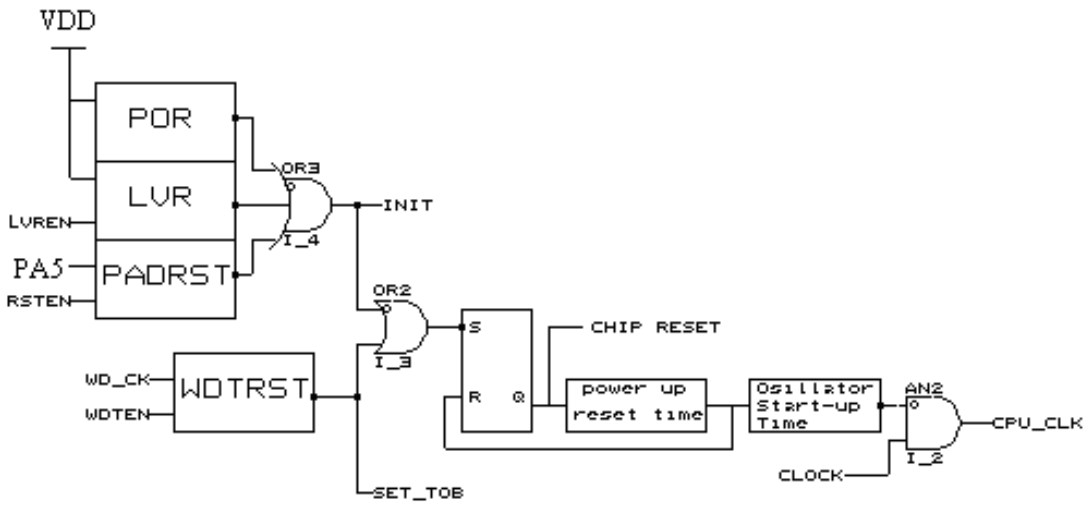


图 34　芯片复位电路框图

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，将阻止過大电流，ESD或电气过载信号进入复位引脚。
- 二极管D使电容C能在VDD下电时快速放电。

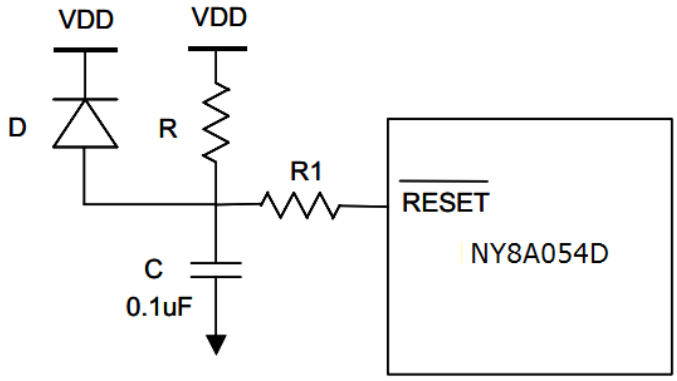


图 35　外部上电复位硬件连接图

4. 指令设置

NY8A054D为各种应用程序提供了 55 个强大的指令。

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

指令	助记符		说明	周期数	影響标志
	1	2			
算术指令					
ADDAR	R	d	dest = R + ACC	1	Z， DC，
SUBAR	R	d	dest = R + (~ACC)	1	Z， DC，
ADCAR	R	d	dest = R + ACC + C	1	Z， DC，
SBCAR	R	d	dest = R + (~ACC) + C	1	Z， DC，
ADDIA	i		ACC = I + ACC	1	Z， DC，
SUBIA	i		ACC = i + (~ACC)	1	Z， DC，
ADCIA	i		ACC = i + ACC + C	1	Z， DC，
SBCIA	i		ACC = i + (~ACC) + C	1	Z， DC，
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	2	-
CALLA			Call subroutine by ACC	2	-
GOTOA			unconditional branch by	2	-
CALL	adr		Call subroutine	2	-
GOTO	adr		unconditional branch	2	-
LCALL	adr		Call subroutine	2	-
LGOTO	adr		unconditional branch	2	-

表 25　指令设置

ACC：累加器。

adr：地址。

bit：R-page中 8 位寄存器的位地址。

C：进位/借位。

C=1，加法指令有进位，减法指令无借位。

C=0，加法指令无进位，减法指令有借位。

d：目标。

若d=“0”，结果存入ACC。

若d=“1”，结果存入R寄存器。

DC：半字节进位/借位标记。

dest：目标。

F：F 页面特殊功能寄存器，F 值为 0x5~0xF。

i：8 位立即数。

PC：程序计数器。

PCHBUF：程序计数器的高字节。

/PD：睡眠标志位。

/PD=1，上电或CLRWDT指令执行后。

/PD=0，SLEEP指令执行后。

Prescaler：预分频器。

R：R页面特殊功能寄存器，R值为 0x00~0x7F。

S：S页面特殊功能寄存器，S值为 0x0 ~ 0x1F。

T0MD：T0MD寄存器。

TBHP：表格指针高字节寄存器。

TBHD：表格数据高字节寄存器。

/TO：看门狗超时标志位。

/TO=1，上電或执行 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 ≤ 127 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 ≤127 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 ≤ 127 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 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“AND”运算。
周期	1
举例	ANDIA i 执行指令前： ACC=0x5A, i=0xAF。 执行指令后： ACC=0x0A, Z=0。

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

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

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[2:0] 载入 PC[10:8]。
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[2:0]赋值给PC[10:8]，将ACC赋值给PC[7:0]。
周期	2
举例	CALLA 执行指令前 TBHP =0x02, ACC =0x34, PC =A0。 Stack pointer=1。 执行指令后： PC=0x234, Stack[1]=A0+1, Stack pointer=2。

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 ≤ 127
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 ≤ 127 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 ≤ 127
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 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 ≤ 127 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 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 ≤ 127 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（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, 关闭总中断。
周期	1
举例	DISI 执行指令前： GIE=1。 执行指令后： 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[2:1]写入PC[10:9]。
周期	2
举例	GOTO Level 执行指令前： PC=A0。 执行指令后： 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，开启總中断。
周期	1
举例	ENI 执行指令前： GIE=0。 执行指令后： 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值寫入PC[7:0]；TBHP[2:0]值寫入PC[10:8]。
周期	2
举例	GOTOA 执行指令前： PC=A0, TBHP=0x02, ACC=0x34 执行指令后： PC=0x234。

INCR	Increase R
语法	INCR R, d
操作数	0 ≤ R ≤ 127 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 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[10:0]。
周期	3
举例	INT 执行指令前： PC=address of INT code。 执行指令后： PC=0x01。

INCRSZ	Increase R, Skip if 0
语法	INCRSZ R, d
操作数	0 ≤ R ≤ 127 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 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 ≤ 127 d = 0, 1
操作	ACC | R → dest
状态影响	Z
说明	ACC和R做"OR"运算，若d="0"，结果存入ACC；若d="1"，结果存入R。
周期	1
举例	IORAR R, d 执行指令前： R=0x50, ACC=0xAA, d=1, Z=0。 执行指令后： R=0xFA, ACC=0xAA, Z=0。

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 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-page特殊寄存器数值给ACC。
周期	1
举例	IOSTR F 执行指令前： F=0x55, ACC=0xAA。 执行指令后： F=0x55, ACC=0x55。

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 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 ≤ 2047
操作	PC + 1 → Top of Stack, adr → PC[10:0]
状态影响	--
说明	长调用子程序。首先将PC+1 压入栈顶，然后将 11 位立即数载入PC[10:0]。
周期	2
举例	LCALL SUB 执行指令前： PC=A0, Stack level=1。 执行指令后： PC=address of SUB, Stack[1]=A0+1, Stack pointer =2。

LGOTO	Unconditional Branch
语法	LGOTO adr
操作数	0 ≤ adr ≤ 2047
操作	adr → PC[10:0]
状态影响	--
说明	无条件长跳转，11位立即数寫入PC[10:0]。
周期	2
举例	LGOTO Level 执行指令前： PC=A0。 执行指令后： PC=address of Level。

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 ≤ 127
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 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 ≤ 127 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 执行指令前： R=0x0, ACC=0xAA, Z=0, d=0。 执行指令后： R=0x0, ACC=0x00, Z=1。

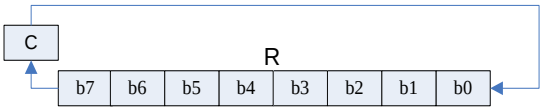
NOP	No Operation
语法	NOP
操作数	--
操作	No operation.
状态影响	--
说明	空操作。
周期	1
举例	NOP 执行指令前： PC=A0 执行指令后： PC=A0+1

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0。
周期	2
举例	RETIA i 执行指令前： GIE=0, Stack pointer =2, i=0x55, ACC=0xAA。 执行指令后： GIE=0, 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。

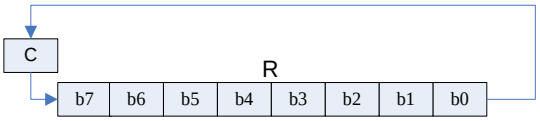
RET	Return from Subroutine
语法	RET
操作数	--
操作	Top of Stack → PC
状态影响	--
说明	子程序返回，栈顶载入PC。
周期	2
举例	RET 执行指令前： Stack level=2。 执行指令后： PC=Stack[2], Stack level=1。

RLR	Rotate Left R Through Carry
语法	RLR R, d
操作数	0 ≤ R ≤ 127 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 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 ≤ 127 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 ≤ 127 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 ≤ 31
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 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 ≤ 31
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 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和分频器 0 清零。/TO标志为0, /PD清零, IC进入睡眠。
周期	1
举例	SLEEP 执行指令前： /PD=1, /TO=0。 执行指令后： /PD=0, /TO=1。

SUBAR	Subtract ACC from R
语法	SUBAR R, d
操作数	0 ≤ R ≤ 127 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)

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

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) 执行指令前： i=0x05, ACC=0x06。 执行指令后： ACC=0xFF, C=0。(-1) (b) 执行指令前： i=0x06, ACC=0x05, d=1. 执行指令后： ACC=0x01, C=1。(+1)

TABLEA	Read ROM data
语法	TABLEA
操作数	--
操作	ROM data{ TBHP, ACC } [7:0] → ACC。 ROM data{TBHP, ACC} [13:8] → TBHD。
状态影响	--
说明	ROM查表指令，高字节存入TBHD，低字节存入ACC。
周期	2
举例	TABLEA 执行指令前： TBHP=0x02, ACC=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	ACC写入T0MD寄存器。
周期	1
举例	T0MD 执行指令前： T0MD=0x55, ACC=0xAA。 执行指令后： T0MD=0xAA。

XORAR	Exclusive-OR ACC with R
语法	XORAR R, d
操作数	0 ≤ R ≤ 127 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 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T0MD寄存器到ACC。
周期	1
举例	T0MDR 执行指令前： T0MD=0x55, ACC=0xAA。 执行指令后： ACC=0x55。

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 执行指令前： i=0xA5, ACC=0xF0。 执行指令后： ACC=0x55。

5. 配置字节表

项目	名称	选项
1	高频振荡模式	1. I_HRC 2. E_HXT 3. E_XT
2	低频振荡模式	1. I_LRC 2. E_LXT
3	内部高速 RC 频率	1. 1MHz 2. 2MHz 3. 4MHz 4. 8MHz 5. 16MHz 6. 20MHz
4	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	指令时钟	1. 2 个振荡周期 2. 4 个振荡周期
6	看门狗定时器	1. 看门狗开启（程序控制） 2. 看门狗关闭（永远关闭）
7	看门狗定时器事件	1. 看门狗复位 2. 看门狗中断
8	定时器 0 时钟源	1. EX_CKI0 2. 低频振荡器（I_LRC/E_LXT）
9	PA.5	1. PA.5 为 I/O 口 2. PA.5 为复位脚
10	PA.7	1. PA.7 为 I/O 口 2. PA.7 输出指令时钟
11	上电复位时间	1. 140us 2. 4.5ms 3. 18ms 4. 72ms 5. 288ms
12	看门狗定时器时基	1. 3.5ms 2. 15ms 3. 60ms 4. 250ms
13	LVR 开关设定	1. 寄存器控制 2. LVR 永远开启
14	LVR 电压	1. 1.6V 2. 1.8V 3. 2.0V 4. 2.2V 5. 2.4V 6. 2.7V 7. 3.0V 8. 3.3V 9. 3.6V 10. 4.2V
15	VDD 电压	1. 3.0V 2. 4.5V 3. 5.0V
16	比较器输入引脚选择	1. 开启 2. 关闭
17	读取输出口数据	1. I/O 口 2. 寄存器
18	E_LXT 强化起振开关设定	1. 自动关闭 2. 寄存器关闭
19	EX_CKI0 to Inst. Clock	1. 同步 2. 不同步
20	上电时钟源	1. 高速（I_HRC/E_HXT/E_XT） 2. 低速（I_LRC/E_LXT）
21	PWM1 输出引脚	1. PB3 2. PA4
22	PWM2 输出引脚	1. PB2 2. PA0
23	PWM3 输出引脚	1. PB4 2. PA3
24	PWM4 输出引脚	1. PB5 2. PA2
25	PWM5 输出引脚	1. PB0 2. PA1
26	输入高电压（VIH）	1. CMOS（0.7VDD） 2. TTL（0.5VDD）
27	输入低电压（VIL）	1. CMOS（0.3VDD） 2. TTL（0.2VDD）
28	INT1 输入引脚选择	1. PB1 2. PA2

表 26 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

6.2 直流电气特性

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

符号	参数	V_{DD}	最小值	典型值	最大值	单位	条件
V_{DD}	工作电压	--	3.3	--	5.5	V	F_{INST}=20MHz @ I_HRC/2
			2.2				F_{INST}=20MHz @ I_HRC/4
			2.7				F_{INST}=16MHz @ E_HXT/2
			2.0				F_{INST}=16MHz @ E_HXT/4
			2.0				F_{INST}=8MHz @ I_HRC/4 & I_HRC/2
			2.0				F_{INST}=8MHz @ E_HXT/4 & E_HXT/2
			1.8				F_{INST}=4MHz @ I_HRC/4 & I_HRC/2
			1.8				F_{INST}=4MHz @ E_XT/4 & E_XT/2
			1.8				F_{INST}=32KHz @ I_LRC/4 & I_LRC/2
			1.8				F_{INST}=32KHz @ E_LXT/4 & E_LXT/2
V_{IH}	输入高电平	5V	4.0	--	--	V	RSTb（0.8 V_{DD}）
		3V	2.4	--	--		
		5V	3.5	--	--	V	所有I/O引脚，EX_CKI0/1，INT0/1 CMOS（0.7 V_{DD}）
		3V	2.1	--	--		
		5V	2.5	--	--	V	所有I/O引脚，EX_CKI0/1 TTL（0.5 V_{DD}）
		3V	1.5	--	--		
V_{IL}	输入低电平	5V	--	--	1.0	V	RSTb（0.2 V_{DD}）
		3V	--	--	0.6		
		5V	--	--	1.5	V	所有I/O引脚，EX_CKI0/1，INT0/1 CMOS option（0.3 V_{DD}）
		3V	--	--	0.9		
		5V	--	--	1.0	V	所有I/O引脚，EX_CKI0/1 TTL option（0.2 V_{DD}）
		3V	--	--	0.6		
I_{OH}	输出驱动电流	5V	--	18	--	mA	V_{OH}=4.0V
		3V	--	10	--		V_{OH}=2.0V
I_{OL}	输出灌电流	5V	--	41	--	mA	V_{OL}=1.0V
		3V	--	26	--		
I_{IR}	红外输出灌电流	5V	--	43	--	mA	V_{OL}=1.0V
		3V	--	27	--		

符号	参数	V_{DD}	最小值	典型值	最大值	单位	条件
I_{OP}	工作电流	正常模式					
		5V	--	3.1	--	mA	F_{HOSC}=20MHz @ I_HRC/2 & E_HXT/2
		3V	--	1.4	--		
		5V	--	2.1	--	mA	F_{HOSC}=20MHz @ I_HRC/4 & E_HXT/4
		3V	--	0.9	--		
		5V	--	2.6	--	mA	F_{HOSC}=16MHz @ I_HRC/2 & E_HXT/2
		3V	--	1.1	--		
		5V	--	1.8	--	mA	F_{HOSC}=16MHz @ I_HRC/4 & E_HXT/4
		3V	--	0.8	--		
		5V	--	1.6	--	mA	F_{HOSC}=8MHz @ I_HRC/2 & E_HXT/2
		3V	--	0.7	--		
		5V	--	1.2	--	mA	F_{HOSC}=8MHz @ I_HRC/4 & E_HXT/4
		3V	--	0.5	--		
		5V	--	1.2	--	mA	F_{HOSC}=4MHz @ I_HRC/2 & E_XT/2
		3V	--	0.5	--		
		5V	--	1.0	--	mA	F_{HOSC}=4MHz @ I_HRC/4 & E_XT/4
		3V	--	0.4	--		
		5V	--	0.9	--	mA	F_{HOSC}=1MHz @ I_HRC/2 & E_XT/2
		3V	--	0.4	--		
		5V	--	0.8	--	mA	F_{HOSC}=1MHz @ I_HRC/4 & E_XT/4
		3V	--	0.3	--		
		慢速模式					
		5V	--	6.9	--	uA	F_{HOSC} 关闭， F_{LOSC}=32KHz @ I_LRC/2
		3V	--	3.0	--		
		5V	--	7.9	--	uA	F_{HOSC} 关闭， F_{LOSC}=32KHz @ E_LXT/2
		3V	--	3.2	--		
		5V	--	4.8	--	uA	F_{HOSC} 关闭， F_{LOSC}=32KHz @ I_LRC/4
		3V	--	2.0	--		
		5V	--	5.8	--	uA	F_{HOSC} 关闭， F_{LOSC}=32KHz @ E_LXT/4
		3V	--	2.2	--		
I_{STB}	待机电流	5V	--	3.2	--	uA	待机模式，F_{HOSC}关闭， F_{LOSC}=32KHz @ I_LRC/4
		3V	--	1.2	--		
I_{HALT}	睡眠电流	5V	--	--	0.5	uA	睡眠模式，关闭WDT
		3V	--	--	0.2		
		5V	--	--	5.0	uA	睡眠模式，开启WDT
		3V	--	--	2.0		
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	--		

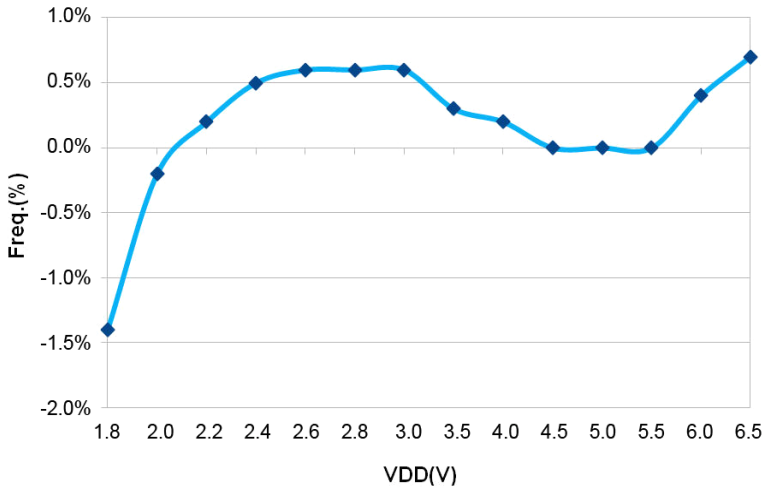
6.3 比较器/LVD电气特性

（V_{DD}=5V，V_{SS}=0V，环境温度 T_A=25°C）

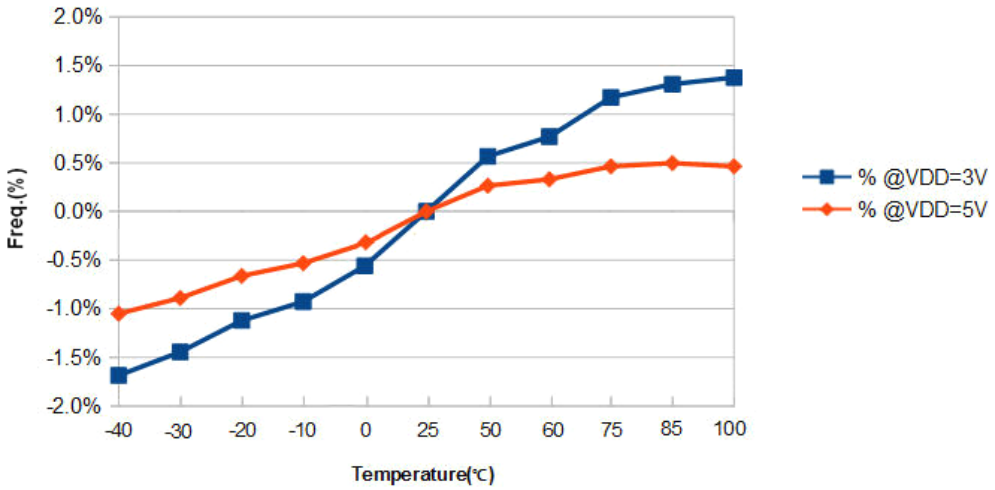
符号	参数	最小值	典型值	最大值	单位	条件
V_{IVR}	比较器输入电压范围	0	--	5	V	F_{HOSC}=1MHz
T_{ENO}	比较器启动等待时间	--	20	--	ms	F_{HOSC}=1MHz
I_{CO}	比较器电流消耗	--	250	--	uA	F_{HOSC}=1MHz，P2V模式
I_{LVD}	LVD电流消耗	--	300	--	uA	F_{HOSC}=1MHz，LVD=4.3V
E_{LVD}	LVD电压误差	--	--	3	%	F_{HOSC}=1MHz，LVD=4.3V

6.4 特性图

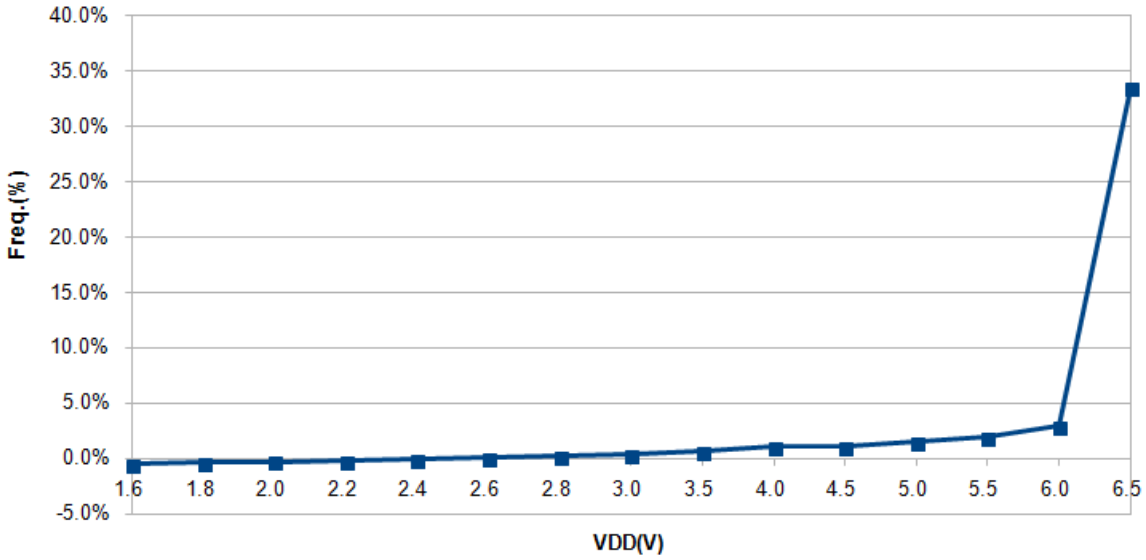
6.4.1 高速 RC 振荡频率与电源电压曲线图



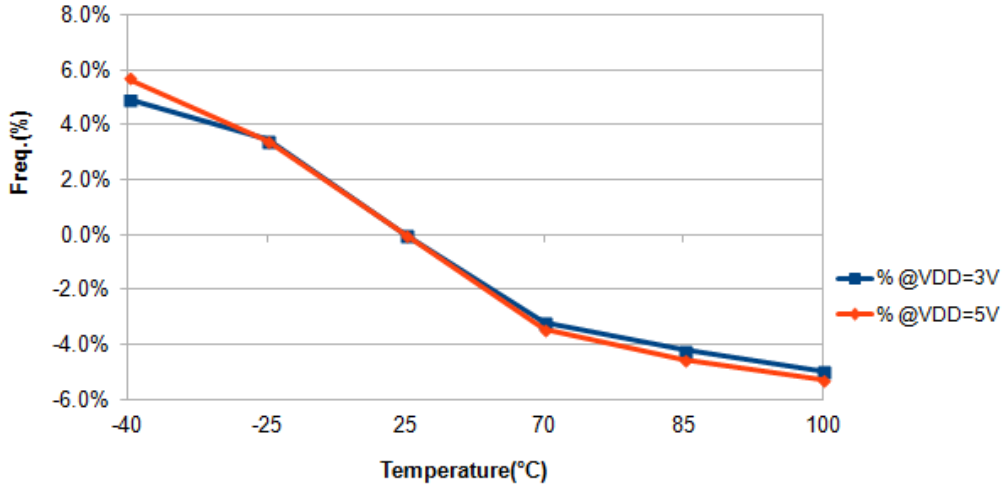
6.4.2 高速 RC 振荡频率与温度曲线图



6.4.3 低速 RC 振荡频率与电源电压曲线图



6.4.4 低速 RC 振荡频率与温度曲线图

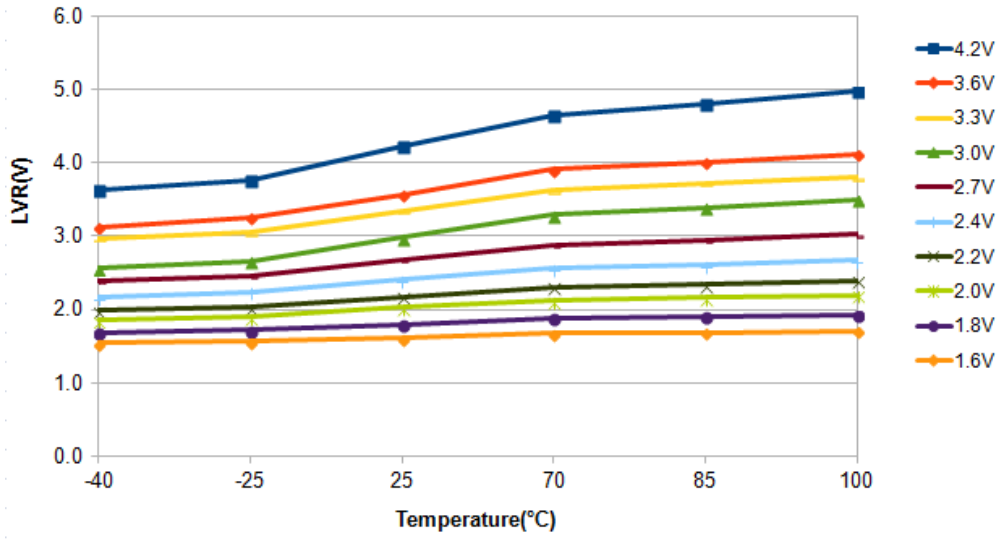


6.5 建议工作电压

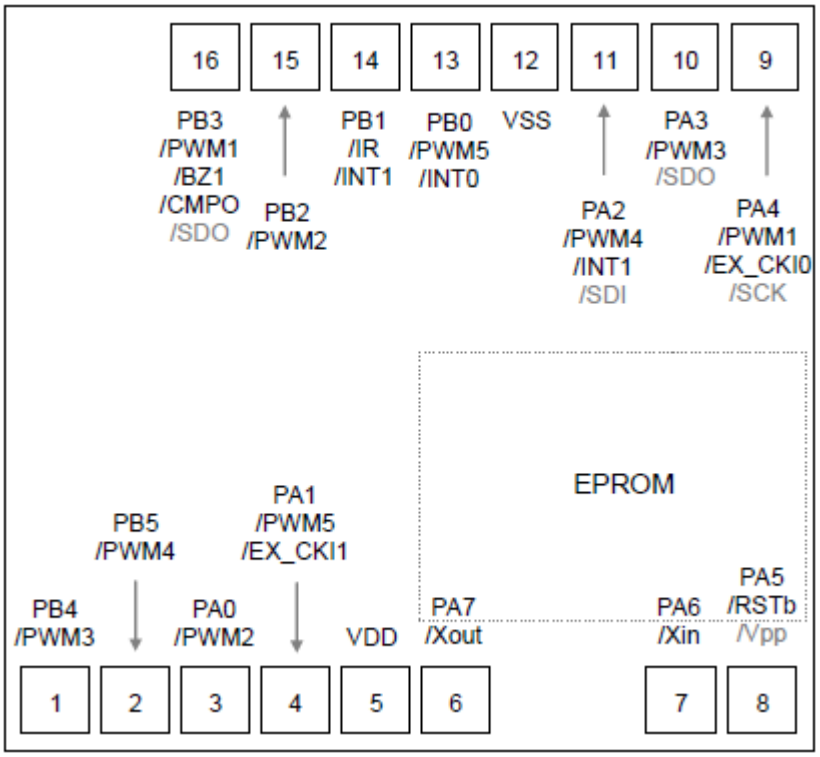
建议工作电压（温度范围：-40 °C ~ +85 °C）

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	3.3V	5.5V	3.6V	4.2V
16M/2T	2.7V	5.5V	3.0V	3.6V
20M/4T	2.2V	5.5V	2.4V	3.0V
16M/4T	2.0V	5.5V	2.2V	2.4V
8M（2T or 4T）	2.0V	5.5V	2.2V	2.4V
≦6M（2T or 4T）	1.8V	5.5V	2.0V	2.2V

6.6 LVR电压与温度曲线图

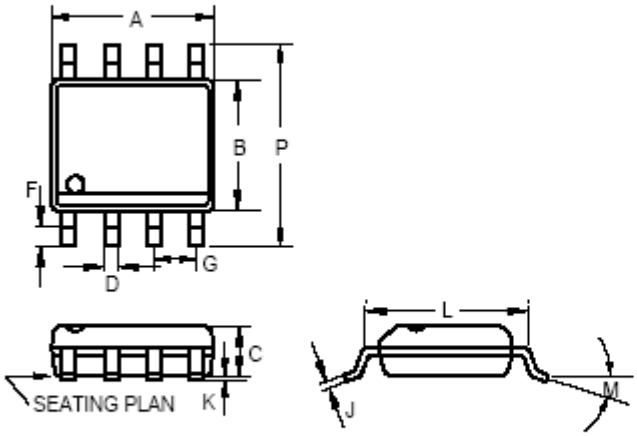


7. 芯片脚位坐标图



8. 封装尺寸

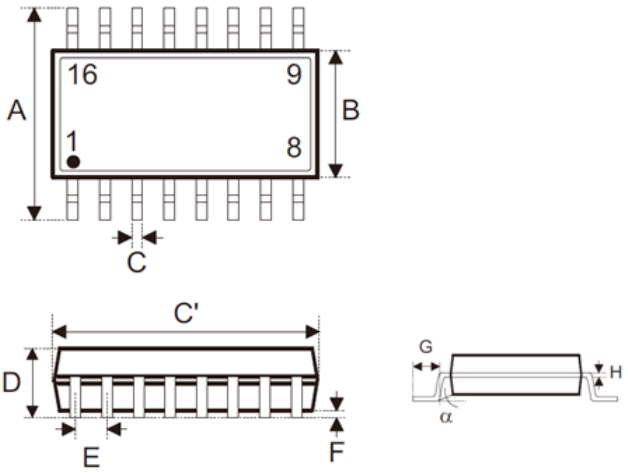
8.1 8 引脚SOP（150 毫寸）



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

注意：8-pin SOP 每管 100 颗。

8.2 16 引脚SOP（150 毫寸）



	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°

注意：16-pin SOP 每管 50 颗。

9. 订购信息

产品名称	*封装类型*	*引脚数*	*封装尺寸*	*配送方式*
NY8A054D	Die	--	--	--
NY8A054DS8	SOP	8	150 mil	卷装：每卷 2.5K颗 管装：每管 100 颗
NY8A054DS16	SOP	16	150 mil	卷装：每卷 2.5K颗 管装：每管 50 颗