九齊科技股份有限公司
Nyquest Technology Co., Ltd.

DATA SHEET

NY8B060D

6 I/O + 12-ch ADC 8-bit EPROM-Based MCU

Version 1.0

Jun. 30, 2022

Revision History

Version	*Date*	*Description*	*Modified Page*
1.0	2022/06/30	Formal release.	

Table of Contents

1. 概述

NY8B060D 是以EPROM作為記憶體的 8 位元微控制器，專為家電或量測等等的I/O應用設計。採用CMOS製程並同時提供客戶低成本、高性能、及高性價比等顯著優勢。NY8B060D 核心建立在RISC精簡指令集架構可以很容易地做編輯和控制，共有 55 條指令。除了少數指令需要 2 個時序，大多數指令都是 1 個時序即能完成，可以讓使用者輕鬆地以程式控制完成不同的應用。因此非常適合各種中低記憶容量但又複雜的應用。

NY8B060D內建高精度五加二通道十二位元類比數位轉換器，與高精度電壓比較器，足以應付各種類比介面的偵測與量測。

在I/O的資源方面，NY8B060D 有 6 根彈性的雙向I/O腳，每個I/O腳都有單獨的暫存器控制為輸入或輸出腳。而且每一個I/O腳位都有附加的程式控制功能如上拉或下拉電阻或開漏極(Open-Drain) 輸出。此外針對紅外線搖控的產品方面，NY8B060D內建了可選擇頻率的紅外載波發射口。

NY8B060D 有二組計時器，可用系統頻率當作一般的計時的應用或者從外部訊號觸發來計數。另外NY8B060D 提供 3 組 10 位元解析度的PWM輸出，1 組蜂鳴器輸出可用來驅動馬達、LED、或蜂鳴器等等。

NY8B060D 採用雙時鐘機制，高速振盪或者低速振盪都由內部RC振盪輸入。在雙時鐘機制下，NY8B060D 可選擇多種工作模式如正常模式(Normal)、慢速模式(Slow mode)、待機模式(Standby mode) 與睡眠模式(Halt mode)可節省電力消耗延長電池壽命。

在省電的模式下如待機模式(Standby mode) 與睡眠模式(Halt mode)中，有多種事件可以觸發中斷喚醒NY8B060D進入正常操作模式(Normal) 或 慢速模式(Slow mode) 來處理突發事件。

1.1 功能

- 寬廣的工作電壓：
  - 2.0V ~ 5.5V @系統頻率 ≦8MHz。
  - 2.2V ~ 5.5V @系統頻率 > 8MHz。
- 寬廣的工作溫度：-40°C ~ 85°C。
- 1Kx14 bits EPROM。
- 64 bytes SRAM。
- 6 根可分別單獨控制輸入輸出方向的I/O腳(GPIO)、PA[5,4,2]、PB[3,2,1]
- PA[5,4,2] 及 PB[3,2,1] 可選擇輸入時使用內建下拉電阻。
- PA[5,4,2] 及 PB[3,2,1] 可選擇輸入時使用上拉電阻。
- PB[3,2,1] 可選擇開漏極輸出(Open-Drain)。
- PA[5] 可選擇當作輸入或開漏極輸出(Open-Drain)。
- 所有I/O腳輸出可選擇一般灌電流(Normal Sink Current)或大灌電流(Large Sink Current)。
- 8 層程式堆棧(Stack)。
- 存取資料有直接或間接定址模式。

- 一組 8 位元上數計時器(Timer0)包含可程式化的頻率預除線路。
- 一組 10 位元下數計時器(Timer1)可選重複載入或連續下數計時。
- 三個 10 位元脈衝寬度調變(PWM1, 2, 3)。
- 一個蜂鳴器輸出(BZ1)。
- 38/57KHz紅外線載波頻率可供選擇，同時載波之極性也可以根據數據作選擇。
- 內建準確的低電壓偵測電路(LVD)。
- 內建五加二通道 12 位元類比數位轉換器(Analog to Digital Converter)。
- 內建準確的電壓比較器(Voltage Comparator)。
- 內建上電復位電路(POR)。
- 內建低壓復位功能(LVR)。
- 內建看門狗計時(WDT)，可由程式韌體控制開關。
- 內建電阻頻率轉換器(RFC)功能.
- 雙時鐘機制，系統可以隨時切換高速振盪或者低速振盪。
  - 高速振盪：I_HRC (1~20MHz內部高速RC振盪)
  - 低速振盪：I_LRC (內部 32KHz低速RC振盪)
- 四種工作模式可隨系統需求調整電流消耗：正常模式(Normal)、慢速模式(Slow mode)、待機模式(Standby mode) 與 睡眠模式(Halt mode)。
- 七種硬體中斷：
  - Timer0 溢位中斷。
  - Timer1 借位中斷。
  - WDT 中斷。
  - PA/PB 輸入狀態改變中斷。
  - 一組外部中斷輸入。
  - 低電壓偵測中斷。
  - 類比數位轉換完成中斷。
- NY8B060D在待機模式(Standby mode)下的七種喚醒中斷：
  - Timer0 溢位中斷。
  - Timer1 借位中斷。
  - WDT 中斷。
  - PA/PB 輸入狀態改變中斷。
  - 一組外部中斷輸入。
  - 低電壓偵測中斷。
  - 類比數位轉換完成中斷。
- NY8B060D在睡眠模式(Halt mode)下的三種喚醒中斷：

- WDT 中斷。
- PA/PB 輸入狀態改變中斷。
- 一組外部中斷輸入。

1.2 NY8B060D 與 NY8B062E、NY8B062D 的主要差異

Item	Function	NY8B060D	NY8B062E	NY8B062D
1	ADC offset Calibration*	**Yes**	Yes	---
2	ADC power consumption	**500uA @5V**	500uA @5V	3mA @5V
4	I/O Input Schmitt Trigger	**Enable/Disable**	Enable/Disable	---
5	Comparator	**Share with LVD**	Rail-to-Rail	Rail-to-Rail

***註: 需要增加 ADC 零點校準初始化程式 (可參考 NYIDE 範例程式)**

1. General Description

NY8B060D is an EPROM based 8-bit MCU tailored for ADC based applications like home appliances or meter equipment. NY8B060D adopts advanced CMOS technology to provide customers remarkable solution with low cost, high performance. RISC architecture is applied to NY8B060D and it provides 55 instructions. All instructions are executed in single instruction cycle except program branch and skip instructions which will take two instruction cycles. Therefore, NY8B060D is very suitable for those applications that are sophisticated but compact program size is required.

NY8B060D provides 5+2 channel high-precision 12-bit analog-to-digital converter (ADC), and high-precision Low Dropout Regulator and analog voltage comparator. They are suitable for any analog interface detection and measurement applications.

As NY8B060D address I/O type applications, it can provide 6 I/O pins for applications which require abundant input and output functionality. Moreover, each I/O pin may have additional features, like Pull-High/Pull-Low resistor and open-drain output type through programming. Moreover, NY8B060D has built-in infrared (IR) carrier generator with selectable IR carrier frequency and polarity for applications which demand remote control feature.

NY8B060D also provides 2 sets of timers which can be used as regular timer based on system oscillation or event counter with external trigger clock. Moreover, NY8B060D provides 3 sets of 10-bit resolution Pulse Width Modulation (PWM) output and 1 sets of buzzer output in order to drive motor/LED and buzzer.

NY8B060D employs dual-clock oscillation mechanism, either high oscillation or low oscillation can be derived from internal resistor/capacitor oscillator. Moreover, based on dual-clock mechanism, NY8B060D provides kinds of operation mode like Normal mode, Slow mode, Standby mode and Halt mode in order to save power consumption and lengthen battery operation life. While NY8B060D operates in Standby mode and Halt mode, kinds of event will issue interrupt requests and can wake-up NY8B060D to enter Normal mode and Slow mode in order to process urgent events.

1.1 Features

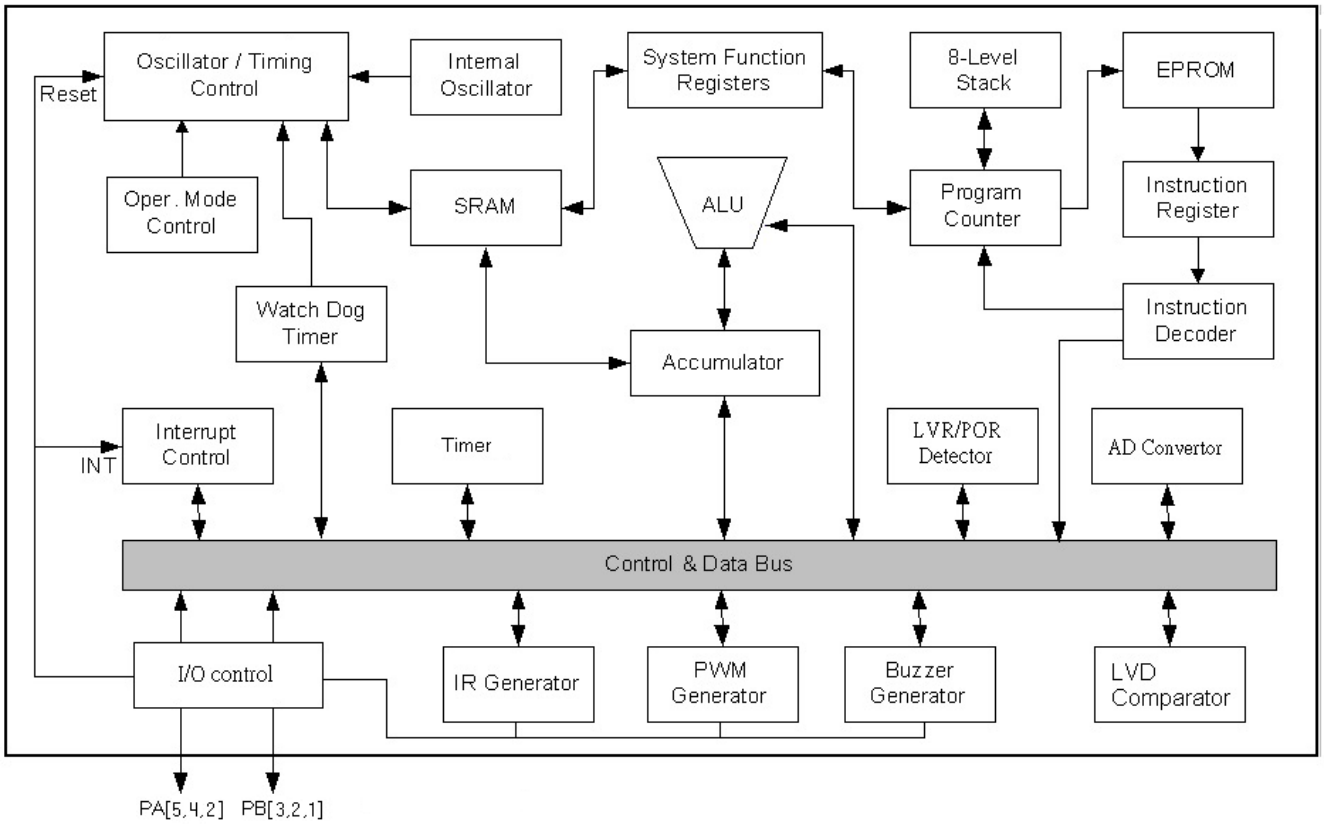
- Wide operating voltage range:
  - 2.0V ~ 5.5V @system clock ≦8MHz.
  - 2.2V ~ 5.5V @system clock > 8MHz.
- Wide operating temperature: -40°C ~ 85°C.
- 1K x 14 bits EPROM.
- 64 bytes SRAM.
- 6 general purpose I/O pins (GPIO), PA[5,4,2], PB[3,2,1], with independent direction control.
- PA[5,4,2] and PB[3,2,1] have features of Pull-Low resistor for input pin.

- PA[5,4,2] and PB[3,2,1] have features of Pull-High resistor.
- PB[3,2,1] have features of Open-Drain output.
- PA[5] have feature of input or open-drain output.
- I/O ports output current mode can be normal sink or large sink.
- 8-level hardware Stack.
- Direct and indirect addressing modes for data access.
- One 8-bit up-count timer (Timer0) with programmable prescaler.
- One 10-bit reload or continuous down-count timers (Timer1).
- Three 10-bit resolution PWM (PWM1, 2, 3) output.
- One buzzer (BZ1) output.
- Selectable 38/57KHz IR carrier frequency and high/low polarity according to data value.
- Built-in high-precision Low-Voltage Detector (LVD).
- Built-in 6+1 channel high-precision 12-bit ADC.
- Built-in high-precision Voltage Comparator.
- Built-in Power-On Reset (POR).
- Built-in Low-Voltage Reset (LVR).
- Built-in Watch-Dog Timer (WDT) enabled/disabled by firmware control.
- Built-in Resistance to Frequency Converter (RFC) function.
- Dual-clock oscillation: System clock can switch between high oscillation and low oscillation.
  - High oscillation: I_HRC (Internal High Resistor/Capacitor Oscillator ranging from 1M~20MHz)
  - Low oscillation: I_LRC (Internal 32KHz oscillator)
- Four kinds of operation mode to reduce system power consumption:
  - Normal mode, Slow mode, Standby mode and Halt mode.
- Seven hardware interrupt events:
  - Timer0 overflow interrupt.
  - Timer1 underflow interrupt.
  - WDT timeout interrupt.
  - PA/PB input change interrupt.
  - 1 set External interrupt.
  - LVD interrupt.
  - ADC end-of-convert interrupt.
- Seven interrupt events to wake-up NY8B060D from Standby mode:
  - Timer0 overflow interrupt.

- Timer1 underflow interrupt.
- WDT timeout interrupt.
- PA/PB input change interrupt.
- 1 set External interrupt.
- LVD interrupt.
- ADC end-of-convert interrupt.

- Three interrupt events to wake-up NY8B060D from Halt mode:
  - WDT timeout interrupt.
  - PA/PB input change interrupt.
  - 1 set External interrupt.

1.2 Block Diagram



1.3 Pin Assignment

NY8B060D provides one kinds of package type which is SOP8.

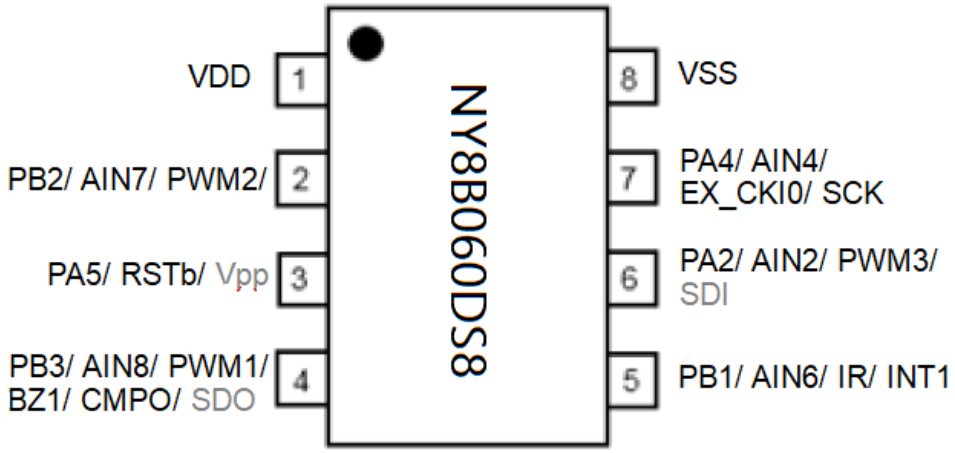


Figure 1 Package pin assignment

1.4 Pin Description

Pin Name	I/O	Description
PA2 AIN2 PWM3 SDI	I/O	PA2 is a bidirectional I/O pin, and can be comparator analog input pin. AIN2 is ADC analog input pin. PA2 can be the output of PWM3 PA2 can be programming pad SDI.
PA4 AIN4 EX_CKI0 SCK	I/O	PA4 is a bidirectional I/O pin. AIN4 is ADC analog input pin. PA4 can be the Timer0/1 clock source EX_CKI0. PA4 can be programming pad SCK.
PA5 RSTb Vpp	I/O	PA5 is an input pin or open-drain output pin. PA5 can be the reset pin RSTb. If this pin is more than 7.75V, IC will enter EPROM programming mode.
PB1 AIN6 IR INT1	I/O	PB1 is a bidirectional I/O pin. AIN6 is ADC analog input pin. If IR mode is enabled, this pin is IR carrier output. PB1 can be the input pin of external interrupt INT1. Moreover it can be ADC external high reference voltage source.
PB2 AIN7 PWM2 F_{INST} OUT	I/O	PB2 is a bidirectional I/O pin. AIN7 is ADC analog input pin. PB2 can be the output of PWM2. PB2 also can be output of instruction clock.
PB3 AIN8 PWM1/BZ1/CMPO SDO	I/O	PB3 is a bidirectional I/O pin. AIN8 is ADC analog input pin. PB3 can be the output of PWM1, Buzzer1 or comparator. PB3 can be programming pad SDO.
VDD	-	Positive power supply.
VSS	-	Ground.

2. Memory Organization

NY8B060D memory is divided into two categories: one is program memory and the other is data memory.

2.1 Program Memory

The program memory space of NY8B060D is 1K words. Therefore, the Program Counter (PC) is 10-bit wide in order to address any location of program memory.

Some locations of program memory are reserved as interrupt entrance. Power-On Reset vector is located at 0x000. Software interrupt vector is located at 0x001. Internal and external hardware interrupt vector is located at 0x008.

NY8B060D provides instructions LCALL and LGOTO to address any location of program space.

When a call or interrupt is happening, next ROM address is written to top of the stack, when RET, RETIA or RETIE instruction is executed, the top of stack data is read and load to PC.

NY8B060D program ROM address 0x3FE~0x3FF are reserved space, if user tries to write code in these addresses will get unexpected false functions.

NY8A060D program ROM address 0x00E~0x00F are preset rolling code can be released and used as normal program space.

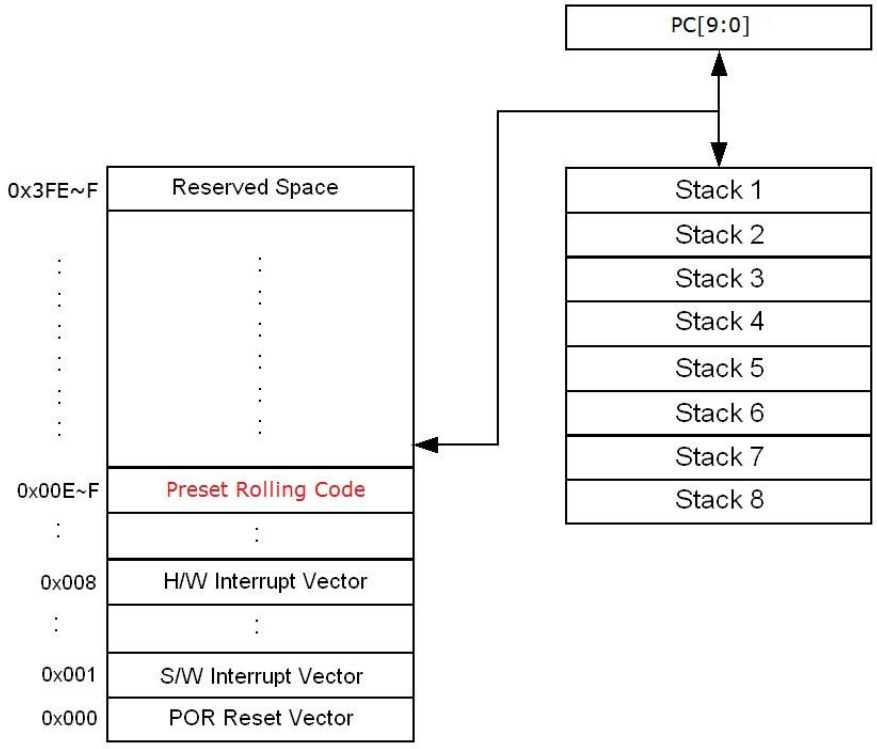


Figure 2 Program Memory Address Mapping

2.2 Data Memory

According to instructions used to access data memory, the data memory can be divided into three kinds of categories: one is R-page Special-function register (SFR) + General Purpose Register (GPR), another is F-page SFR and the other is S-page SFR. GPR are made of SRAM and user can use them to store variables or intermediate results.

R-page data memory is divided into 4 banks and can be accessed directly or indirectly through a SFR register which is File Select Register (FSR). STATUS [7:6] are used as Bank register BK[1:0] to select one bank out of the 4 banks.

R-page register can be divided into addressing mode: direct addressing mode and indirect addressing mode.

The indirect addressing mode of data memory access is described in the following graph. This indirect addressing mode is implied by accessing register INDF. The bank selection is determined by STATUS[7:6] and the location selection is from FSR[6:0].

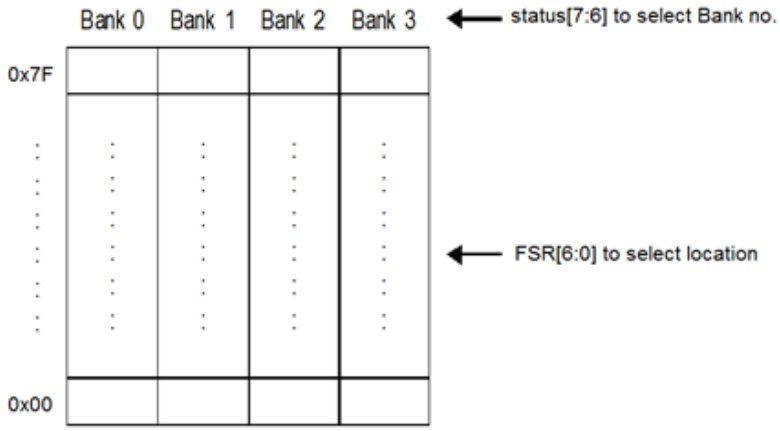


Figure 3 Indirect Addressing Mode of Data Memory Access

The direct addressing mode of data memory access is described below. The bank selection is determined by STATUS [7:6] and the location selection is from instruction op-code[6:0] immediately.

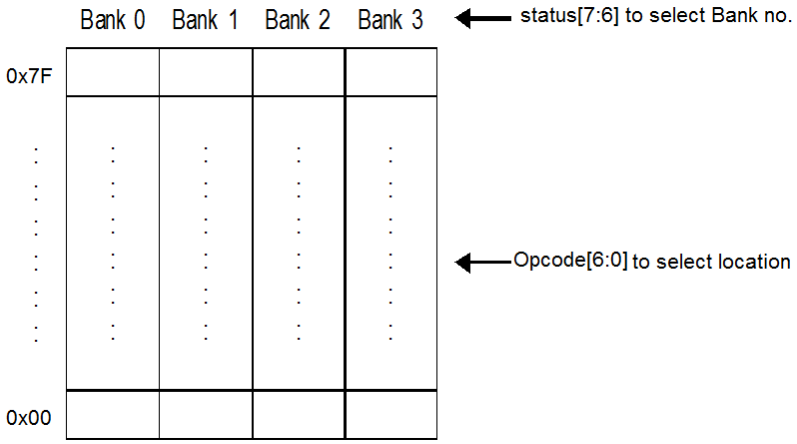


Figure 4 Direct Addressing Mode of Data Memory Access

R-page SFR can be accessed by general instructions like arithmetic instructions and data movement instructions. The R-page SFR occupies address from 0x0 to 0x1F of Bank 0. However, the same address range of Bank 1, Bank 2 and Bank 3 are mapped back to Bank 0. In other words, R-page SFR physically existed at Bank 0. The GPR physically occupy address from 0x20 to 0x7F of Bank 0 and 0x20 to 0x3F of Bank 1. Other bank in address from 0x20 to 0x7F are mapped back as the Table 1 shows.

The NY8B060D register name and address mapping of R-page SFR are described in the following table.

<table>
<tr><th>Status [7:6]
Address</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">The same mapping as 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>ADMD</td></tr>
<tr><td>0x11</td><td>ADR</td></tr>
<tr><td>0x12</td><td>ADD</td></tr>
<tr><td>0x13</td><td>ADVREFH</td></tr>
<tr><td>0x14</td><td>ADCR</td></tr>
<tr><td>0x15</td><td>AWUCON</td></tr>
<tr><td>0x16</td><td>PACON</td></tr>
<tr><td>0x17</td><td>ADJMD</td></tr>
<tr><td>0x18</td><td>INTEDG</td></tr>
<tr><td>0x19</td><td>TMRH</td></tr>
<tr><td>0x1A</td><td>ANAEN</td></tr>
<tr><td>0x1B</td><td>RFC</td><td colspan="3" rowspan="2">The same mapping as Bank 0</td></tr>
<tr><td>0x1C</td><td>TM3RH</td></tr>
<tr><td>0x1D ~0x1E</td><td>-</td><td colspan="3">-</td></tr>
<tr><td>0x1F</td><td>-</td><td colspan="3">-</td></tr>
<tr><td>0x20 ~ 0x5F</td><td>General Purpose Register</td><td>Mapped to bank0</td><td>Mapped to bank0</td><td>Mapped to Bank1</td></tr>
</table>

Table 1 R-page SFR Address Mapping

F-page SFR can be accessed only by instructions IOST and IOSTR. S-page SFR can be accessed only by instructions SFUN and SFUNR. STATUS[7:6] bank select bits are ignored while F-page and S-page register is accessed. The register name and address mapping of F-page and S-page are depicted in the following table.

SFR Category / Address	F-page SFR	S-page SFR
0x0	-	TMR1
0x1	-	T1CR1
0x2	-	T1CR2
0x3	-	PWM1DUTY
0x4	-	PS1CV
0x5	IOSTA	BZ1CR
0x6	IOSTB	IRCR
0x7	-	TBHP
0x8	-	TBHD
0x9	APHCON	-
0xA	PS0CV	P2CR1
0xB	-	-
0xC	BODCON	PWM2DUTY
0xD	-	-
0xE	CMPCR	-
0xF	PCON1	OSCCR
0X10	-	-
0X11	-	P3CR1
0X12	-	-
0X13	-	PWM3DUTY
0X14	-	-
0X15	-	-

Table 2 F-page and S-page SFR Address Mapping

3. Function Description

This chapter will describe the detailed operations of NY8B060D.

3.1 R-page Special Function Register

3.1.1 INDF (Indirect Addressing Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INDF	R	0x0	INDF[7:0]							
R/W Property			R/W							
Initial Value			xxxxxxxx							

The register INDF is not physically existed and it is used as indirect addressing mode. Any instruction accessing INDF actually accesses the register pointed by register FSR

3.1.2 TMR0 (Timer0 Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMR0	R	0x1	TMR0[7:0]							
R/W Property			R/W							
Initial Value			xxxxxxxx							

When read the register TMR0, it actually read the current running value of Timer0.

Write the register TMR0 will change the current value of Timer0.

Timer0 clock source can be from instruction clock F_{INST}, or from external pin EX_CKI0, or from Low Oscillator Frequency according to T0MD and configuration word setting.

3.1.3 PCL (Low Byte of PC[9:0])

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCL	R	0x2	PCL[7:0]							
R/W Property			R/W							
Initial Value			0x00							

The register PCL is the least significant byte (LSB) of 10-bit PC. PCL will be increased by one after one instruction is executed except some instructions which will change PC directly. The high byte of PC, i.e. PC[9:8], is not directly accessible. Update of PC[9:8] must be done through register PCHBUF.

For LGOTO instruction, PC[9:0] is from instruction word.

For LCALL instruction, PC[9:0] is from instruction word. Moreover the next PC address, i.e. PC+1, will push onto top of Stack.

3.1.4 STATUS (Status Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
STATUS	R	0x3	BK[1]	BK[0]	-	/TO	/PD	Z	DC	C
R/W Property			R/W	R/W	-	R/W(*2)	R/W(*1)	R/W	R/W	R/W
Initial Value			0	0	X	1	1	X	X	X

The register STATUS contains result of arithmetic instructions and reasons to cause reset.

C: Carry/Borrow bit

C=1, carry is occurred for addition instruction or borrow is not occurred for subtraction instruction.

C=0, carry is not occurred for addition instruction or borrow is occurred for subtraction instruction.

DC: Half Carry/half Borrow bit

DC=1, carry from the 4th LSB is occurred for addition instruction or borrow from the 4th LSB is not occurred for subtraction instruction.

DC=0, carry from the 4th LSB is not occurred for addition instruction or borrow from the 4th LSB is occurred for subtraction instruction.

Z: Zero bit

Z=1, result of logical operation is zero.

Z=0, result of logical operation is not zero.

/PD: Power down flag bit

/PD=1, after power-up or after instruction CLRWDT is executed.

/PD=0, after instruction SLEEP is executed.

/TO: Time overflow flag bit

/TO=1, after power-up or after instruction CLRWDT or SLEEP is executed.

/TO=0, WDT timeout is occurred.

BK[1:0]: Bank register is used to select one specific bank of data memory. BK[1:0]=00b, Bank 0 is selected. BK[1:0]=01b, Bank 1 is selected. BK[1:0]=10b, Bank 2 is selected. BK[1:0]=11b, Bank 3 is selected.

(*1) can be cleared by sleep instruction.

(*2) can be set by clrwdt instruction.

3.1.5 FSR (Register File Selection Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FSR	R	0x4	-	FSR[6:0]						
R/W Property			-	R/W						
Initial Value			X	X	X	X	X	X	X	X

FSR[6:0]: Select one register out of 128 registers of specific Bank.

3.1.6 PortA (PortA Data Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PortA	R	0x5	-	-	PA5	PA4	-	PA2	-	-
R/W Property			R/W							
Initial Value			Data latch value is xxxxxxxx, read value is xxxxxxxx port value(PA[5,4,2])							

While reading PortA, it will get the status of the specific pin if that pin is configured as input pin. However, if that pin is configured as output pin, whether it will get the status of the pin or the value of the corresponding output data latch is depend on the configuration option RD_OPT. While writing to PortA, data is written to PA's output data latch.

3.1.7 PortB (PortB Data Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PortB	R	0x6	-	-	-	-	PB3	PB2	PB1	-
R/W Property			R/W							
Initial Value			Data latch value is xxxxxxxx, read value is xxxxxxxx port value(PB3~PB1)							

While reading PortB, it will get the status of the specific pin if that pin is configured as input pin. However, if that pin is configured as output pin, whether it will get the status of the pin or the value of the corresponding output data latch is depend on the configuration option RD_OPT. While writing to PortB, data is written to PB's output data latch.

3.1.8 PCON (Power Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	R	0x8	WDTEN	/PLPA4	LVDEN	/PHPA5	LVREN	-	-	-
R/W Property			R/W					-	-	-
Initial Value			1	0	0	1	1	X	X	X

LVREN: Enable/disable LVR.

LVREN=1, enable LVR.

LVREN=0, disable LVR.

/PHPA5: Disable/enable PA5 Pull-High resistor.

/PHPA5=1, disable PA5 Pull-High resistor.

/PHPA5=0, enable PA5 Pull-High resistor.

LVDEN: Enable/disable LVD.

LVDEN=1, enable LVD.

LVDEN=0, disable LVD.

/PLPA4: Disable/enable PA4 Pull-Low resistor.

/PHPA4=1, disable PA4 Pull-Low resistor.

/PHPA4=0, enable PA4 Pull-Low resistor.

WDTEN: Enable/disable WDT.

WDTEN=1, enable WDT.

WDTEN=0, disable WDT.

3.1.9 BWUCON (PortB Wake-up Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BWUCON	R	0x9	-	-	-	-	WUPB3	WUPB2	WUPB1	-
R/W Property			-	-	-	-	R/W	R/W	R/W	-
Initial Value			X	X	X	X	1	1	1	X

WUPBx: Enable/disable PBx wake-up function, 1 ≤ x ≤ 3.

WUPBx=1, enable PBx wake-up function.

WUPBx=0, disable PBx wake-up function.

3.1.10 PCHBUF (High Byte of PC)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCHBUF	R	0xA	-	-	-	-	-	-	PCHBUF[1:0]	
R/W Property			-	-	-	-	-	-	W	
Initial Value			X	X	X	X	X	X	00	

PCHBUF[1:0]: Buffer of the 9^{th} ~ 8^{th} bit of PC.

3.1.11 ABPLCON (PortA/PortB Pull-Low Resistor Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ABPLCON	R	0xB	/PLPB3	/PLPB2	/PLPB1	-	-	/PLPA2	-	-
R/W Property			R/W							
Initial Value			1	1	1	X	X	1	X	X

/PLPAx: Disable/enable PA2 Pull-Low resistor.

/PLPAx=1, disable PA2 Pull-Low resistor.

/PLPAx=0, enable PA2 Pull-Low resistor.

/PLPBx: Disable/enable PBx Pull-Low resistor, 1 ≤ x ≤ 3.

/PLPBx=1, disable PBx Pull-Low resistor.

/PLPBx=0, enable PBx Pull-Low resistor.

3.1.12 BPHCON (PortB Pull-High Resistor Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BPHCON	R	0xC	-	-	-	-	/PHPB3	/PHPB2	/PHPB1	-
R/W Property			-	-	-	-	R/W	R/W	R/W	-
Initial Value			X	X	X	X	1	1	1	X

/PHPBx: Disable/enable PBx Pull-High resistor, 1 ≤ x ≤ 3.

/PHPBx=1, disable PBx Pull-High resistor.

/PHPBx=0, enable PBx Pull-High resistor.

3.1.13 INTE (Interrupt Enable Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTE	R	0xE	INT1IE	WDTIE	-	LVDIE	T1IE	-	PABIE	T0IE
R/W Property			R/W	R/W	-	R/W	R/W	-	R/W	R/W
Initial Value			0	0	X	0	0	X	0	0

T0IE: Timer0 overflow interrupt enable bit.

T0IE=1, enable Timer0 overflow interrupt.

T0IE=0, disable Timer0 overflow interrupt.

PABIE: PortA**/**PortB input change interrupt enable bit.

PABIE=1, enable PortA**/**PortB input change interrupt.

PABIE=0, disable PortA**/**PortB input change interrupt.

T1IE: Timer1 underflow interrupt enable bit.

T1IE=1, enable Timer1 underflow interrupt.

T1IE=0, disable Timer1 underflow interrupt.

LVDIE: Low-voltage detector interrupt enable bit.

LVDIE=1, enable low-voltage detector interrupt.

LVDIE=0, disable low-voltage detector interrupt.

WDTIE: WDT timeout interrupt enable bit.

WDTIE=1, enable WDT timeout interrupt.

WDTIE=0, disable WDT timeout interrupt.

INT1IE: External interrupt 1 enable bit.

INT1IE=1, enable external interrupt 1.

INT1IE=0, disable external interrupt 1.

3.1.14 INTF (Interrupt Flag Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF	R	0xF	INT1IF	WDTIF	-	LVDIF	T1IF	-	PABIF	T0IF
R/W Property			R/W	R/W	-	R/W	R/W	-	R/W	R/W
Initial Value(note*)			0	0	X	0	0	X	0	0

T0IF: Timer0 overflow interrupt flag bit.

T0IF=1, Timer0 overflow interrupt is occurred.

T0IF must be clear by firmware.

PABIF: PortA/PortB input change interrupt flag bit.

PABIF=1, PortA/PortB input change interrupt is occurred.

PABIF must be clear by firmware.

T1IF: Timer1 underflow interrupt flag bit.

T1IF=1, Timer1 underflow interrupt is occurred.

T1IF must be clear by firmware.

LVDIF: Low-voltage detector interrupt flag bit.

LVDIF=1, Low-voltage detector interrupt is occurred.

LVDIF must be clear by firmware.

WDTIF: WDT timeout interrupt flag bit.

WDTIF=1, WDT timeout interrupt is occurred.

WDTIF must be clear by firmware.

INT1IF: External interrupt 1 flag bit.

INT1IF=1, external interrupt 1 is occurred.

INT1IF must be clear by firmware.

Note: When corresponding INTE bit is not enabled, the read interrupt flag is 0.

3.1.15 ADMD (ADC mode Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADMD	R	0x10	ADEN	START	EOC	GCHS	CHS3	CHS2	CHS1	CHS0
R/W Property			R/W	W	R	R/W	R/W	R/W	R/W	R/W
Initial Value			0	0	1	0	0	0	0	0

ADEN: ADC enable bit.

ADEN=1, ADC is enabled.

START: Start an ADC conversion session.

When write 1 to this bit, start to execute ADC converting. This bit is write-only. Read this bit will get 0.

EOC: ADC status bit, read-only.

EOC=1 : ADC is end-of-convert, the ADC data present in ADR and ADD is available.

EOC=0 : ADC is in procession.

GCHS: ADC global channel select bit.

GCHS=0 : disable all ADC input channel.

GCHS=1 : enable ADC input channel.

CHS3~0: ADC input channel select bits.

0010 = select PA2 pad as ADC input,

0100 = select PA4 pad as ADC input,

0110 = select PB1 pad as ADC input,

0111 = select PB2pad as ADC input,

1000 = select PB3 pad as ADC input,

1011 = select 1/4 VDD as ADC input.

1100 = select GND as ADC input.

3.1.16 ADR (ADC clock, ADC interrupt flag and ADC LSB output Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADR	R	0x11	ADIF	ADIE	ADCK1	ADCK0	AD3	AD2	AD1	AD0
R/W Property			R/W	R/W	R/W	R/W	R	R	R	R
Initial Value			0	0	0	0	X	X	X	X

ADIF: ADC interrupt flag bit.

ADIF=1, ADC end-of-convert interrupt is occurred.

ADIF must be clear by firmware.

ADIE: ADC end-of-convert interrupt enable bit.

ADIE=1 : enable ADC interrupt.

ADIE=0 : disable ADC interrupt.

ADCK1~0: ADC clock select.

00: ADC clock=F_{INST}/16, 01: ADC clock=F_{INST}/8, 10: ADC clock=F_{INST}/1, 11: ADC clock=F_{INST}/2.

AD3~0: 12-bit low-nibble ADC data buffer.

3.1.17 ADD (ADC output data Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADD	R	0x12	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4
R/W Property			R	R	R	R	R	R	R	R
Initial Value			0	0	0	0	0	0	0	0

AD11~4: High-byte ADC data buffer.

3.1.18 ADVREFH (ADC high reference voltage Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADVREFH	R	0x13	EVHENB	-	-	-	-	-	VHS1	VHS0
R/W Property			R/W	-	-	-	-	-	R/W	R/W
Initial Value			0	X	X	X	X	X	1	1

EVHENB: ADC reference high voltage (VREFH) select control bit.

EVHENB=0: ADC reference high voltage is internal generated, the voltage selected depends on VHS1~0.

Note: EVHENB must be 0..

VHS1~0: ADC internal reference high voltage select bits.

11: VREFH=VDD 10: VREFH=4V 01: VREFH=3V 00: VREFH=2V.

3.1.19 ADCR (Sampling pulse and ADC bit Register)

Name	SFR Type	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCR	R	0x14	-	-	-	PBCON3	SHCK1	SHCK0	ADCR1	ADCR0
R/W Property			-	-	-	R/W	R/W	R/W	R/W	R/W
Initial Value			X	X	X	0	1	0	1	0

SHCK1~0: Sampling pulse width select.

00: 1 ADC clock 01: 2 ADC clock 10: 4 ADC clock 11: 8 ADC clock.

ADCR1~0: ADC conversion bit no. select.

00: 8-bit ADC 01: 10-bit ADC 1x: 12-bit ADC.

PBCON3: PB3 analog pin select.

0=PB3 can be analog ADC input or digital I/O pin.

1=PB3 is pure analog ADC input pin for power-saving.

3.1.20 AWUCON (PortA Wake-up Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
AWUCON	R	0x15	-	-	WUPA5	WUPA4	-	WUPA2	-	-
R/W Property			-	-	R/W	R/W	-	R/W	-	-
Initial Value			X	X	1	1	X	1	X	X

WUPAx: Enable/disable PAx wake-up function, x= 2, 4, 5.

WUPAx=1, enable PAx wake-up function.

WUPAx=0, disable PAx wake-up function.

3.1.21 PACON (ADC analog pin Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PACON	R	0x16	PBCON2	PBCON1	-	PACON4	-	PACON2	-	-
R/W Property			R/W	R/W	-	R/W	-	R/W	-	-
Initial Value			0	0	X	0	X	0	X	X

PACONx: PA analog pin select, x= 2, 4.

0=PAx can be analog ADC input or digital I/O pin.

1=PAx is pure analog ADC input pin for power-saving.

PBCONx: PB analog pin select, 1 ≤ x ≤ 2.

0=PBx can be analog ADC input or digital I/O pin.

1=PBx is pure analog ADC input pin for power-saving.

3.1.22 ADJMD (ADC adjustment mode)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADJMD	R	0x17	-	-	ADJ_SIGN	ADJ[4]	ADJ[3]	ADJ[2]	ADJ[1]	ADJ[0]
R/W Property			-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			X	X	0	0	0	0	0	0

ADJ[x]: adjustment bit select, 0 ≤ x ≤ 4.

00000 = offset 0 mV

11111 = offset 12.5 mV.

ADJ_SIGN: adjustment sign bit

0 = adc data decrease

1 = adc data increase

Note: For application, please refer to NYIDE example code "ADC Interrupt_Auto Calibration".

3.1.23 INTEDG (Interrupt Edge Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTEDG	R	0x18	-	-	EIS1	-	INT1G1	INT1G0	-	-
R/W Property			-	-	R/W	-	R/W	R/W	-	-
Initial Value			X	X	0	X	0	1	X	X

EIS1: External interrupt 1 select bit

EIS1=1, PB1 is external interrupt 1.

EIS1=0, PB1 is GPIO.

INT1G1~0: INT1 edge trigger select bit.

00: reserved 01: rising edge 10: falling edge 11: rising/falling edge.

3.1.24 TMRH (Timer High Byte Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMRH	R	0x19	-	-	TMR19	TMR18	PWM2 DUTY9	PWM2 DUTY8	PWM1 DUTY9	PWM1 DUTY8
R/W Property			-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			X	X	X	X	X	X	X	X

TMR19~8: Timer1 MSB 2 bits. Write these 2 bits will overwrite the 10-bit Timer1 load value of bit 9 and 8. Read these 2 bits will get the Timer1 bit9-8 current value.

PWM2DUTY9~8: PWM2 duty data MSB 2 bits.

PWM1DUTY9~8: PWM1 duty data MSB 2 bits.

3.1.25 ANAEN (Analog Circuit Enable Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ANAEN	R	0x1A	CMPEN	-	-	-	-	-	-	-
R/W Property			R/W	-	-	-	-	-	-	-
Initial Value			0	X	X	X	X	X	X	X

CMPEN: Enable/disable voltage comparator.

CMPEN=1, enable voltage comparator.

CMPEN=0, disable voltage comparator.

3.1.26 RFC (RFC Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RFC	R	0x1B	RFCEN	-	-	-	PSEL[3:0]			
R/W Property			R/W	-	-	-	R/W			
Initial Value			0	X	X	X	0000			

RFCEN: Enable/disable RFC function.

RFCEN=1, enable RFC function.

RFCEN=0, disable RFC function.

PSEL[3:0]: Select RFC pad.

PSEL[3:0]	RFC PAD
0010	PA2
0100	PA4
0101	PA5
1001	PB1
1010	PB2
1011	PB3

Table 3 RFC pad select

3.1.27 TM3RH (Timer3 High Byte Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TM3RH	R	0x1C	-	-			-	-	PWM3D9	PWM3D8
R/W Property			-	-			-	-	R/W	R/W
Initial Value			-	-			-	-	X	X

PWM3DUTY9~8: PWM3 duty data MSB 2 bits.

3.2 T0MD Register

T0MD is a readable/writeable register which is only accessed by instruction T0MD / T0MDR.

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T0MD	-	-	LCKTM0	GP6	T0CS	T0CE	PS0WDT	PS0SEL[2:0]		
R/W Property			R/W							
Initial Value(note*)			0	0	1	1	1	111		

PS0SEL[2:0]: Prescaler0 dividing rate selection. The rate depends on Prescaler0 is assigned to Timer0 or WDT. When Prescaler0 is assigned to WDT, the dividing rate is dependent on which timeout mechanism is selected.

PS0SEL[2:0]	Dividing Rate		
	PS0WDT=0 (Timer0)	PS0WDT=1 (WDT Reset)	PS0WDT=1 (WDT Interrupt)
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

PS0SEL[2:0]	Dividing Rate		
	PS0WDT=0 (Timer0)	PS0WDT=1 (WDT Reset)	PS0WDT=1 (WDT Interrupt)
111	1:256	1:128	1:256

Table 4 Prescaler0 Dividing Rate

PS0WDT: Prescaler0 assignment.

PS0WDT=1, Prescaler0 is assigned to WDT.

PS0WDT=0, Prescaler0 is assigned to Timer0.

Note: Always set PS0WDT and PS0SEL[2:0] before enabling watchdog or timer interrupt, or reset or interrupt may be falsely triggered.

T0CE: Timer0 external clock edge selection.

T0CE=1, Timer0 will increase one while high-to-low transition occurs on pin EX_CKI0.

T0CE=0, Timer0 will increase one while low-to-high transition occurs on pin EX_CKI0.

Note: T0CE is also applied to Low Oscillator Frequency as Timer0 clock source condition.

T0CS: Timer0 clock source selection.

T0CS=0, Instruction clock F_{INST} is selected.

Note: T0CS must always be 0.

GP6: General register.

LCKTM0: T0CS=0, Instruction clock F_{INST} is selected as Timer0 clock source.

Note: For more detail descriptions of Timer0 clock source select, please see Timer0 section.

3.3 F-page Special Function Register

3.3.1 IOSTA (PortA I/O Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOSTA	F	0x5	-	-	IOPA5	IOPA4	-	IOPA2	-	-
R/W Property			-	-	R/W	R/W	-	R/W	-	-
Initial Value			X	X	1	1	X	1	X	X

IOPAx: PAx I/O mode selection, x = 2, 4, 5.

IOPAx=1, PAx is input mode.

IOPAx=0, PAx is output mode.

3.3.2 IOSTB (PortB I/O Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOSTB	F	0x6	-	-	-	-	IOPB3	IOPB2	IOPB1	-

R/W Property	-	-	-	-	R/W	R/W	R/W	-
Initial Value	X	X	X	X	1	1	1	X

IOPBx: PBx I/O mode selection, 1 ≤ x ≤ 3.

IOPBx=1, PBx is input mode.

IOPBx=0, PBx is output mode.

3.3.3 APHCON (PortA Pull-High Resistor Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
APHCON	F	0x9	-	-	/PLPA5	/PHPA4	-	/PHPA2	-	-
R/W Property			R/W							
Initial Value			X	X	1	1	X	1	X	X

/PHPAx: Enable/disable Pull-High resistor of PAx, x= 2, 4.

/PHPAx=1, disable Pull-High resistor of PAx.

/PHPAx=0, enable Pull-High resistor of PAx.

/PLPA5: Enable/disable Pull-Low resistor of PA5.

/PLPA5=1, disable Pull-Low resistor of PA5.

/PLPA5=0, enable Pull-Low resistor of PA5.

3.3.4 PS0CV (Prescaler0 Counter Value Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS0CV	F	0xA	PS0CV[7:0]							
R/W Property			R							
Initial Value			1	1	1	1	1	1	1	1

While reading PS0CV, it will get current value of Prescaler0 counter.

3.3.5 BODCON (PortB Open-Drain Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BODCON	F	0xC	-	-	-	-	ODPB3	ODPB2	ODPB1	-
R/W Property			-	-	-	-	R/W	R/W	R/W	-
Initial Value			X	X	X	X	0	0	0	X

ODPBx: Enable/disable open-drain of PBx, 1 ≤ x ≤ 3.

ODPBx=1, enable open-drain of PBx.

ODPBx=0, disable open-drain of PBx.

3.3.6 CMPCR (Comparator voltage select Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPCR	F	0xE	-	RBIASH	RBIASL	CMP_INV	PS1	PS0	NS1	NS0
R/W Property			R/W							
Initial Value			X	0	0	0	0	0	0	0

NS[1:0]: Comparator inverting input select.

NS[1:0]	Inverting input
00	-
01	-
10	Bandgap (0.6V)
11	Vref

PS[1:0]: Comparator non-inverting input select

PS[1:0]	Non-inverting input
00	-
01	PA2
10	Vref
11	---

CMPF_INV: Comparator output inverse control bit.

CMPF_INV = 1, Inverse comparator output.

CMPF_INV = 0, Non-inverse comparator output.

RBIAS_L, RBIAS_H: Set corresponding voltage reference levels

(please refer to chapter 3.13.1)

3.3.7 PCON1 (Power Control Register1)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON1	F	0xF	GIE	LVDOUT	LVDS3	LVDS2	LVDS1	LVDS0	-	T0EN
R/W Property			R/W(1*)	R	R/W	R/W	R/W	R/W	-	R/W
Initial Value			0	X	1	1	1	1	X	1

T0EN: Enable/disable Timer0.

T0EN=1, enable Timer0.

T0EN=0, disable Timer0.

LVDS3~0: Select one of the 16 LVD voltage.

LVDS[3:0]	Voltage
0000	1.9V
0001	2.0V
0010	2.2V
0011	2.4V
0100	2.6V
0101	2.8V
0110	2.9V
0111	3.0V
1000	3.15V
1001	3.30V
1010	3.45V
1011	3.60V
1100	3.75V
1101	3.90V
1110	4.05V
1111	4.15V

Table 7 LVD voltage select

LVDOUT: Low voltage detector output, read-only.

GIE: Global interrupt enable bit.

GIE=1, enable all unmasked interrupts.

GIE=0, disable all interrupts.

(1*) : set by instruction ENI, clear by instruction DISI, read by instruction IOSTR.

3.4 S-page Special Function Register

3.4.1 TMR1 (Timer1 Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMR1	S	0x0	TMR1[7:0]							
R/W Property			R/W							
Initial Value			XXXXXXXX							

When reading register TMR1, it will obtain current value of 10-bit down-count Timer1 at TMR1[9:0]. When writing register TMR1, it will write data from TMRH[5:4] and Timer1 reload register to TMR1[9:0] current content.

3.4.2 T1CR1 (Timer1 Control Register1)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T1CR1	S	0x1	PWM1OEN	PWM1OAL	-	-	T1_HRC	T1OS	T1RL	T1EN
R/W Property			R/W	R/W	-	-	-	R/W	R/W	R/W
Initial Value			0	0	X	X	X	0	0	0

This register is used to configure Timer1 functionality.

T1EN: Enable/disable Timer1.

T1EN=1, enable Timer1.

T1EN=0, disable Timer1.

T1RL: Configure Timer1 down-count mechanism while Non-Stop mode is selected (T1OS=0).

T1RL=1, initial value is reloaded from reload register TMR1[9:0].

T1RL=0, continuous down-count from 0x3FF when underflow is occurred.

T1OS: Configure Timer1 operating mode while underflow is reached.

T1OS=1, One-Shot mode. Timer1 will count once from the initial value to 0x00.

T1OS=0, Non-Stop mode. Timer1 will keep down-count after underflow.

T1OS	T1RL	Timer1 Down-Count Functionality
0	0	Timer1 will count from reload value down to 0x00. When underflow is reached, 0x3FF is reloaded and continues down-count.
0	1	Timer1 will count from reload value down to 0x00. When underflow is reached, reload value is reloaded and continues to down-count.
1	x	Timer1 will count from initial value down to 0x00. When underflow is reached, Timer1 will stop down-count.

Table 8 Timer1 Functionality

T1_HRC: Timer1 clock source selection.

T1_HRC =1, PWM1,2,3 & Timer 1 clock source is High Oscillator Clock.

T1_HRC =0, PWM1,2,3 & Timer 1 clock source selection depends on T1CS register bit.

PWM1OAL: Define PWM1 output active state.

PWM1OAL=1, PWM1 output is active low.

PWM1OAL=0, PWM1 output is active high.

PWM1OEN: Enable/disable PWM1 output.

PWM1OEN=1, PWM1 output will be present on PB3.

PWM1OEN=0, PB3 is GPIO.

3.4.3 T1CR2 (Timer1 Control Register2)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

T1CR2	S	0x2	-	-	T1CS	T1CE	/PS1EN	PS1SEL[2:0]		
R/W Property			-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			X	X	1	1	1	1	1	1

This register is used to configure Timer1 functionality.

PS1SEL[2:0]: Prescaler1 dividing rate selection.

PS1SEL[2:0]	Dividing Rate
000	1:2
001	1:4
010	1:8
011	1:16
100	1:32
101	1:64
110	1:128
111	1:256

Table 9 Prescaler1 Dividing Rate

Note: Always set PS1SEL[2:0] at /PS1EN=1, or interrupt may be falsely triggered.

/PS1EN: Disable/enable Prescaler1.

/PS1EN=1, disable Prescaler1.

/PS1EN=0, enable Prescaler1.

T1CE: Timer1 external clock edge selection.

T1CE=1, Timer1 will decrease one while high-to-low transition occurs on pin EX_CKI0.

T1CE=0, Timer1 will decrease one while low-to-high transition occurs on pin EX_CKI0.

T1CS: Timer1 clock source selection.

T1CS=1, External clock on pin EX_CKI0 is selected.

T1CS=0, Instruction clock is selected.

3.4.4 PWM1DUTY (PWM1 Duty Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1DUTY	S	0x3	PWM1DUTY[7:0]							
R/W Property			W							
Initial Value			XXXXXXXX							

The reload value of 10-bit Timer1 stored on registers TMRH[5:4] and TMR1[7:0] is used to define the PWM1 frame rate, and registers TMRH[1:0] and PWM1DUTY[7:0] is used to define the duty cycle of PWM1.

3.4.5 PS1CV (Prescaler1 Counter Value Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

PS1CV	S	0x4	PS1CV[7:0]							
R/W Property			R							
Initial Value			1	1	1	1	1	1	1	1

While reading PS1CV, it will get current value of Prescaler1 counter.

3.4.6 BZ1CR (Buzzer1 Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BZ1CR	S	0x5	BZ1EN	-	-	-	BZ1FSEL[3:0]			
R/W Property			W	-	-	-	W			
Initial Value			0	X	X	X	1	1	1	1

BZ1FSEL[3:0]: Frequency selection of BZ1 output.

BZ1FSEL[3:0]	BZ1 Frequency Selection	
	Clock Source	Dividing Rate
0000	Prescaler1 output	1:2
0001		1:4
0010		1:8
0011		1:16
0100		1:32
0101		1:64
0110		1:128
0111		1:256
1000	Timer1 output	Timer1 bit 0
1001		Timer1 bit 1
1010		Timer1 bit 2
1011		Timer1 bit 3
1100		Timer1 bit 4
1101		Timer1 bit 5
1110		Timer1 bit 6
1111		Timer1 bit 7

Table 10 Buzzer1 Output Frequency Selection

BZ1EN: Enable/Disable BZ1 output.

BZ1EN=1, enable Buzzer1.

BZ1EN=0, disable Buzzer1.

3.4.7 IRCR (IR Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IRCR	S	0x6	-	-	-	-	-	IRCSEL	IRF57K	IREN
R/W Property			-	-	-	-	-	W	W	W
Initial Value			X	X	X	X	X	0	0	0

IREN: Enable/Disable IR carrier output.

IREN=1, enable IR carrier output.

IREN=0, disable IR carrier output.

IRF57K: Selection of IR carrier frequency.

IRF57K=1, IR carrier frequency is 57KHz.

IRF57K=0, IR carrier frequency is 38KHz.

IRCSEL: Polarity selection of IR carrier.

IRCSEL=0, IR carrier will be generated when I/O pin data is 1.

IRCSEL=1, IR carrier will be generated when I/O pin data is 0.

Note:

1. Only high oscillation (F_{HOSC}) (See section 3.17) can be used as IR clock source.

2. Division ratio for different oscillation type.

OSC. Type	57KHz	38KHz	Conditions
High IRC(4MHz)	64	96	HIRC mode (the input to IR module is set to 4MHz no matter what system clock is)

Table 11 Division ratio for different oscillation type

3.4.8 TBHP (Table Access High Byte Address Pointer Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TBHP	S	0x7	-	-	-	-	-	-	TBHP1	TBHP0
R/W Property			-	-	-	-	-	-	R/W	R/W
Initial Value			X	X	X	X	X	X	X	X

When instruction CALLA, GOTOA or TABLEA is executed, the target address is constituted by TBHP[1:0] and ACC. ACC is the Low Byte of PC[9:0] and TBHP[1:0] is the high byte of PC[9:0].

3.4.9 TBHD (Table Access High Byte Data Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TBHD	S	0x8	-	-	TBHD5	TBHD4	TBHD3	TBHD2	TBHD1	TBHD0
R/W Property			-	-	R	R	R	R	R	R
Initial Value			X	X	X	X	X	X	X	X

When instruction TABLEA is executed, high byte of content of addressed ROM is loaded into TBHD[5:0] register. The Low Byte of content of addressed ROM is loaded to ACC.

3.4.11 T2CR1 (Timer2 Control Register1)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2CR1	S	0xA	PWM2OEN	PWM2OAL	-	-	-	-	-	-
R/W Property			R/W	R/W	-	-	-	-	-	-
Initial Value			0	0	X	X	X	X	X	X

PWM2OAL: Define PWM2 output active state.

PWM2OAL=1, PWM2 output is active low.

PWM2OAL=0, PWM2 output is active high.

PWM2OEN: Enable/disable PWM2 output.

PWM2OEN=1, PWM2 output will be present on PB2.

PWM2OEN=0, PB2 is GPIO.

3.4.13 PWM2DUTY (PWM2 Duty Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2DUTY	S	0xC	PWM2DUTY[7:0]							
R/W Property			W							
Initial Value			XXXXXXXX							

The reload value of 10-bit Timer1 stored on registers TMRH[5:4] and TMR1[7:0] is used to define the PWM2 frame rate, and registers TMRH[3:2] and PWM2DUTY[7:0] is used to define the duty cycle of PWM2.

3.4.16 OSCCR (Oscillation Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSCCR	S	0xF	-	CMPOE	-	-	OPMD[1:0]		STPHOSC	SELHOSC
R/W Property			-	R/W	-	-	R/W		R/W	R/W
Initial Value			X	0	X	X	00		0	1

SELHOSC: Selection of system oscillation (F_{OSC}).

SELHOSC=1, F_{OSC} is high-frequency oscillation (F_{HOSC}).

SELHOSC=0, F_{OSC} is low-frequency oscillation (F_{LOSC}).

STPHOSC: Disable/enable high-frequency oscillation (F_{HOSC}).

STPHOSC=1, F_{HOSC} will stop oscillation and be disabled.

STPHOSC=0, F_{HOSC} keep oscillation.

OPMD[1:0]: Selection of operating mode.

OPMD[1:0]	Operating Mode
00	Normal mode
01	Halt mode
10	Standby mode
11	reserved

Table 15 Selection of Operating Mode by OPMD[1:0]

CMPOE: Disable/enable comparator output to pad PB3.

CMPOE=1, enable comparator output to pad PB3.

CMPOE=0, disable comparator output to pad PB3.

Note: Comparator output to pad PB3 has higher priority than pwm1/buzzer1 output.

3.4.18 P3CR1 (PWM3 Control Register1)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3CR1	S	0x11	PWM3OEN	PWM3OAL	-	-	-	-	-	-
R/W Property			R/W	R/W	-	-	-	-	-	-
Initial Value			0	0	X	X	X	X	X	X

PWM3OAL: Define PWM3 output active state.

PWM3OAL=1, PWM3 output is active low.

PWM3OAL=0, PWM3 output is active high.

PWM3OEN: Enable/disable PWM3 output.

PWM3OEN=1, PWM3 output will be present on PA2.

PWM3OEN=0, PA2 is GPIO.

3.4.20 PWM3DUTY (PWM3 Duty Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM3DUTY	S	0x13	PWM3DUTY[7:0]							
R/W Property			W							
Initial Value			XXXXXXXX							

The reload value of 10-bit Timer1 stored on registers TMRH[5:4] and TMR1[7:0] is used to define the PWM3 frame rate, and registers TM3RH[1:0] and PWM3DUTY[7:0] is used to define the duty cycle of PWM3.

3.5 I/O Port

NY8B060D provides 6 I/O pins which are PA[5,4,2] and PB[3:1]. User can read/write these I/O pins through registers PORTA and PORTB respectively. Each I/O pin has a corresponding register bit to define it is input pin or output pin. Register IOSTA[5,4,2] define the input/output direction of PA[PA[5,4,2]. Register IOSTB[3:1] define the input/output direction of PB[3:1].

When an I/O pin is configured as input pin, it may have Pull-High resistor or Pull-Low resistor which is enabled or disabled through registers. Register APHCON[4,2] and PCON[4] are used to enable or disable Pull-High resistor of PA[5,4,2]. Register APHCON[5], PCON[6] and ABPLCON[2] are used to enable or disable Pull-Low resistor of PA[5,4,2]. Register BPHCON[3:1] are used to enable or disable Pull-High resistor of PB[3:1]. Register ABPLCON[7:5] are used to enable or disable Pull-Low resistor of PB[3:1].

When an PortB I/O pin is configured as output pin, there is a corresponding and individual register to select as Open-Drain output pin. Register BODCON[3:1] determine PB[3:1] is Open-Drain or not.

The summary of Pad I/O feature is listed in the table below.

Feature		PA[4,2]	PA[5]	PB[3:1]
Input	Pull-High Resistor	V	V	V
	Pull-Low Resistor	V	V	V
Output	Open-Drain	X	V	V

Table 19 Summary of Pad I/O Feature

The level change on each I/O pin of PA and PB may generate interrupt request. Register AWUCON[5,4,2] and BWUCON[3:1] will select which I/O pin of PA and PB may generate this interrupt. As long as any pin of PA and PB is selected by corresponding bit of AWUCON and BWUCON, the register bit PABIF (INTF[1]) will set to 1 if there is a level change occurred on any selected pin. An interrupt request will occur and interrupt service routine will be executed if register bit PABIE (INTE[1]) and GIE (PCON1[7]) are both set to 1.

There is two external interrupt provided by NY8B060D. When register bit EIS0 (INTEDG[4]) is set to 1, PB0 is used as input pin for external interrupt 0. When register bit EIS1 (INTEDG[5]) is set to 1, PB1 is used as input pin for external interrupt 1.

Note: When PB1 is set as level change operation and external interrupt, the external interrupt will have higher priority, and the PB1 level change operation will be disabled. But PB3~PB2 level change function are not affected.

NY8B060D provides IR carrier generation output. When IREN=1, the IR carrier output will be present on PB1 pad. When IREN=0, the IR carrier will not be generated.

PA5 can be used as external reset input determined by a configuration word. When an active-low signal is applied to PA5, it will cause NY8B060D to enter reset process.

Moreover, PA4 can be timer 0 external clock source EX_CKI0 if T0MD T0CS=1 and LCK_TM0=0. PA4 can be timer 1 external clock source EX_CKI0 if T1CS=1.

Moreover, PB3 can be comparator output if CMPOE=1. PB3 can be PWM1 output If T1CR1[7] PWM1OEN=1. PB3 can be Buzzer1 output if BZ1CR[7] BZ1EN=1. The output priority of PB3 is comparator output > PWM1 output > Buzzer1 output.

PB2 can be PWM2 output If T2CR1[7] PWM2OEN=1.

When configured as output, the sink current of each pin can be normal (19mA for V_{DD} =3V), large (28mA for V_{DD} =3V) according to configuration words. Check the following table for sink current mode setting:

Configuration Word	Normal Sink	Large Sink
PXcurrent	0	1
PXcsc	0	0

Table 20 Sink current mode setting (X=A, B)

3.5.1 Block Diagram of IO Pins

IO_SEL: set pad attribute as input or output

WRITE_EN: write data to pad.

READ_EN: read pad.

PULLUP_ENB: enable Pull-High.

PULLDOWN_EN: enable Pull-Low.

VPEN: enable pad to comparator non-inverting input.

VNEN: enable pad to comparator inverting input.

CMPVP, CMPVN: comparator non-inverting and inverting input.

RD_TYPE: select read pin or read latch.

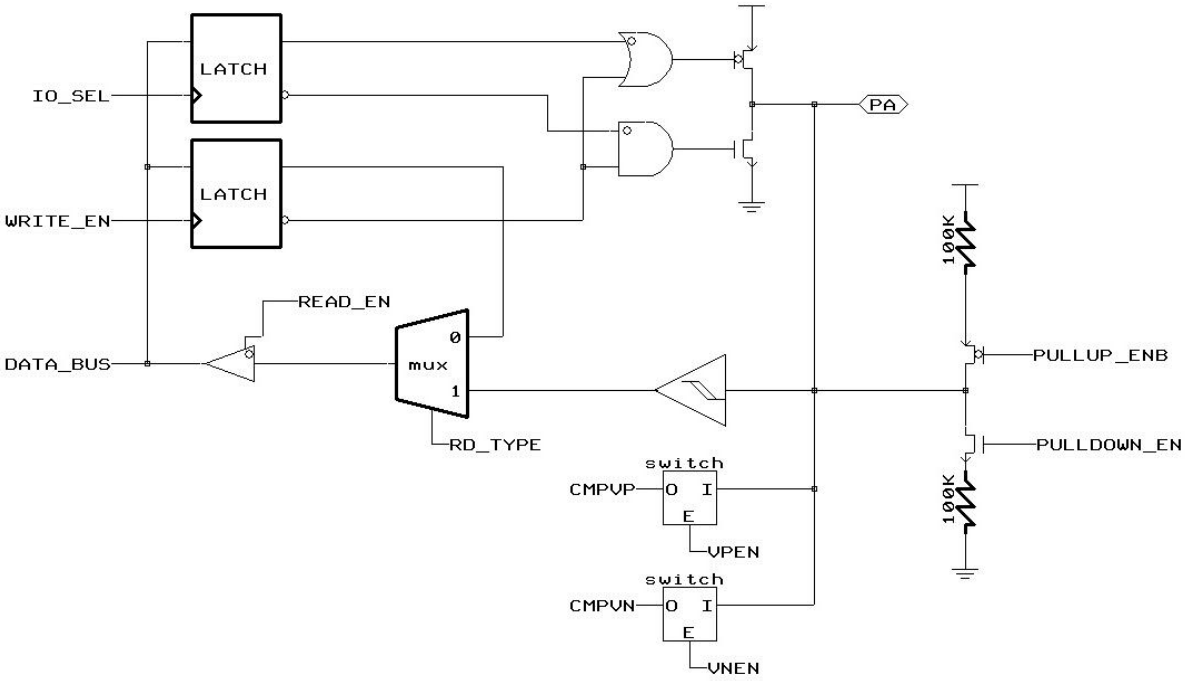


Figure 5 Block Diagram of PA2

IO_SEL: set pad attribute as input or output.

WRITE_EN: write data to pad.

READ_EN: read pad.

PULLUP_ENB: enable Pull-High resister.

PULLDOWN_EN: enable Pull-Low resister.

RD_TYPE: select read pin or read latch.

EX_CKI0: external clock for Timer0, 1.

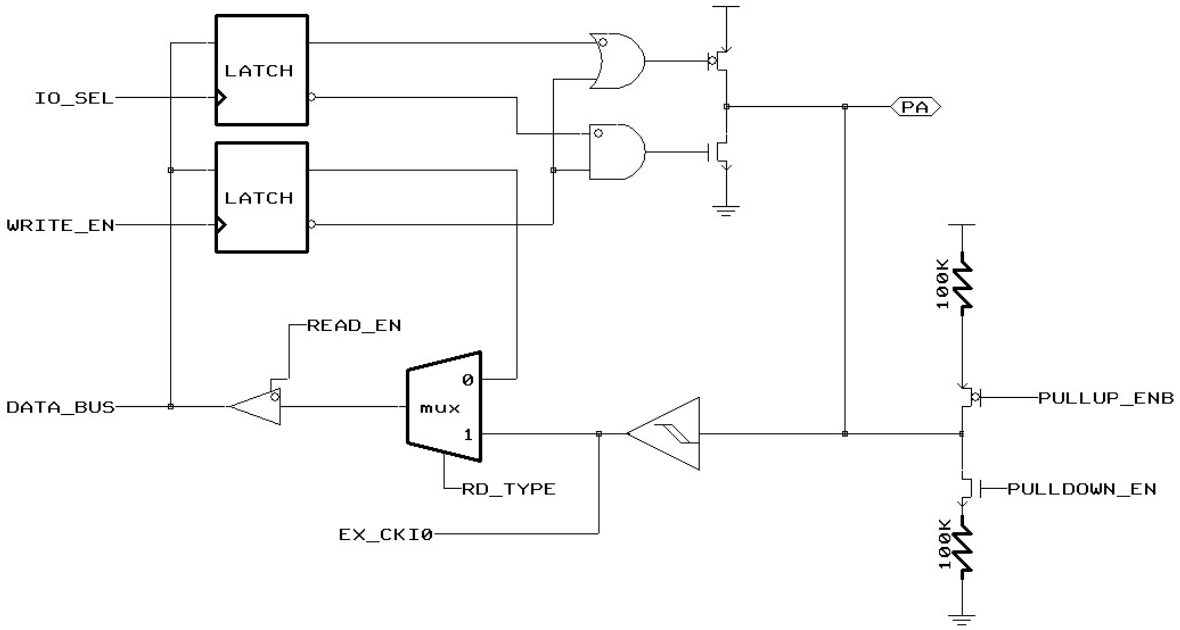


Figure 7 Block Diagram of PA4

RSTPAD_EN: enable PA5 as reset pin.

RSTB_IN: reset signal input.

IO_SEL: set pad attribute as input or output.

WRITE_EN: write data to pad.

READ_EN: read pad.

PULLUP_ENB: enable Pull-High.

PULLDOWN_EN: enable Pull-Low.

RD_TYPE: select read pin or read latch.

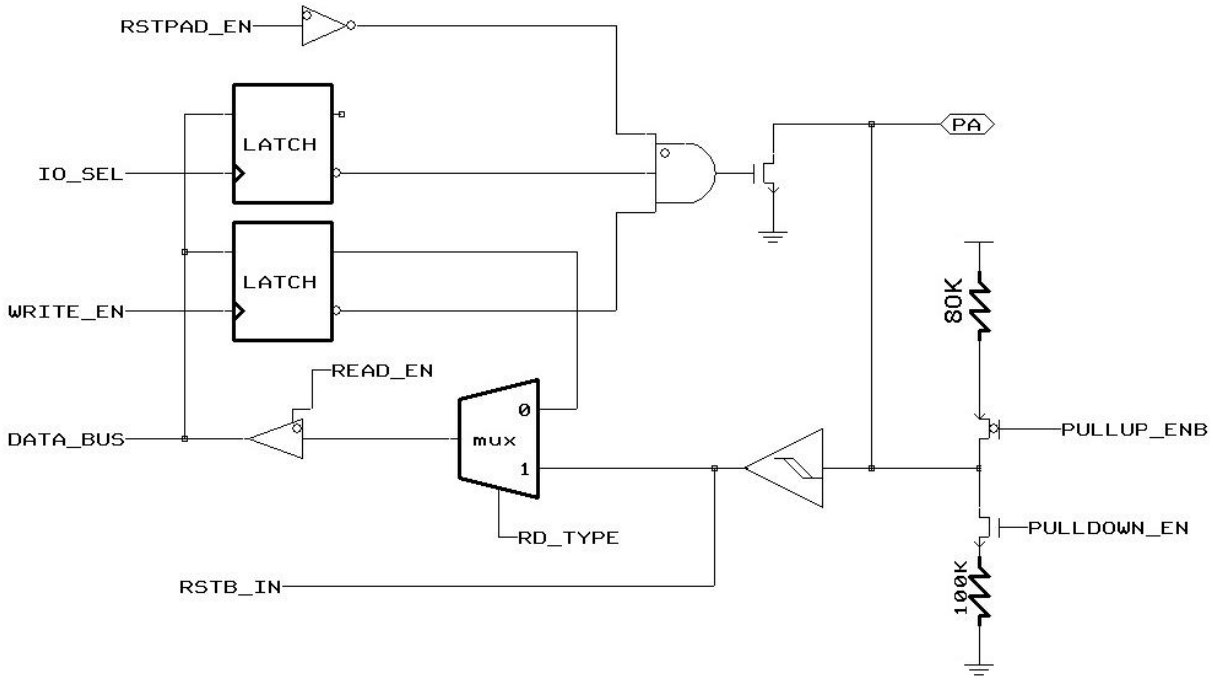


Figure 8 Block Diagram of PA5

IO_SEL: set pad attribute as input or output.

WRITE_EN: write data to pad.

READ_EN: read pad.

OD_EN: enable open-Drain.

PULLUP_ENB: enable Pull-High.

PULLDOWN_EN: enable Pull-Low.

RD_TYPE: select read pin or read latch.

EIS1: external interrupt function enable.

INTEDG[3:2]: external interrupt edge select.

EX_INT1: external interrupt signal.

WUB: port B wake-up enable.

SET_PBIF: port B wake-up flag.

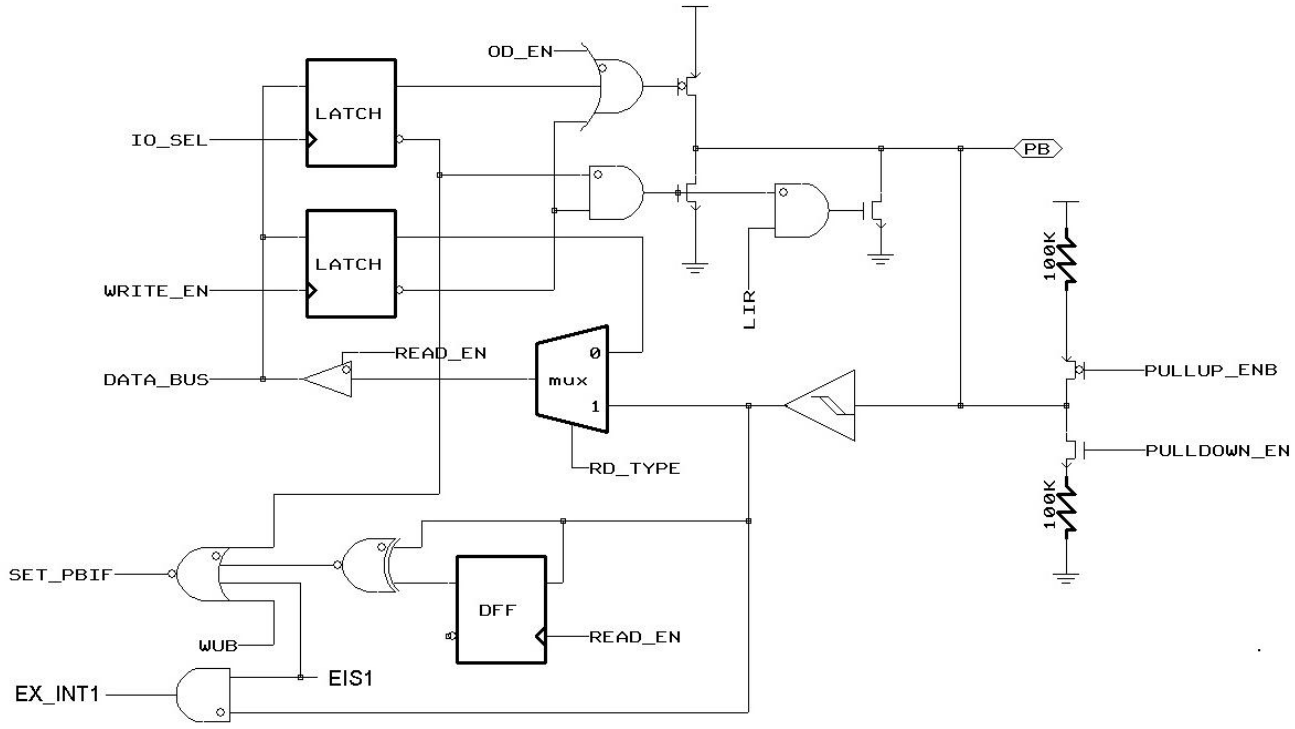


Figure 11 Block Diagram of PB1

IO_SEL: set pad attribute as input or output.

WRITE_EN: write data to pad.

READ_EN: read pad.

OD_EN: enable open-Drain.

PULLUP_ENB: enable Pull-High.

PULLDOWN_EN: enable Pull-Low.

RD_TYPE: select read pin or read latch.

WUB: port B wake-up enable.

SET_PBIF: port B wake-up flag.

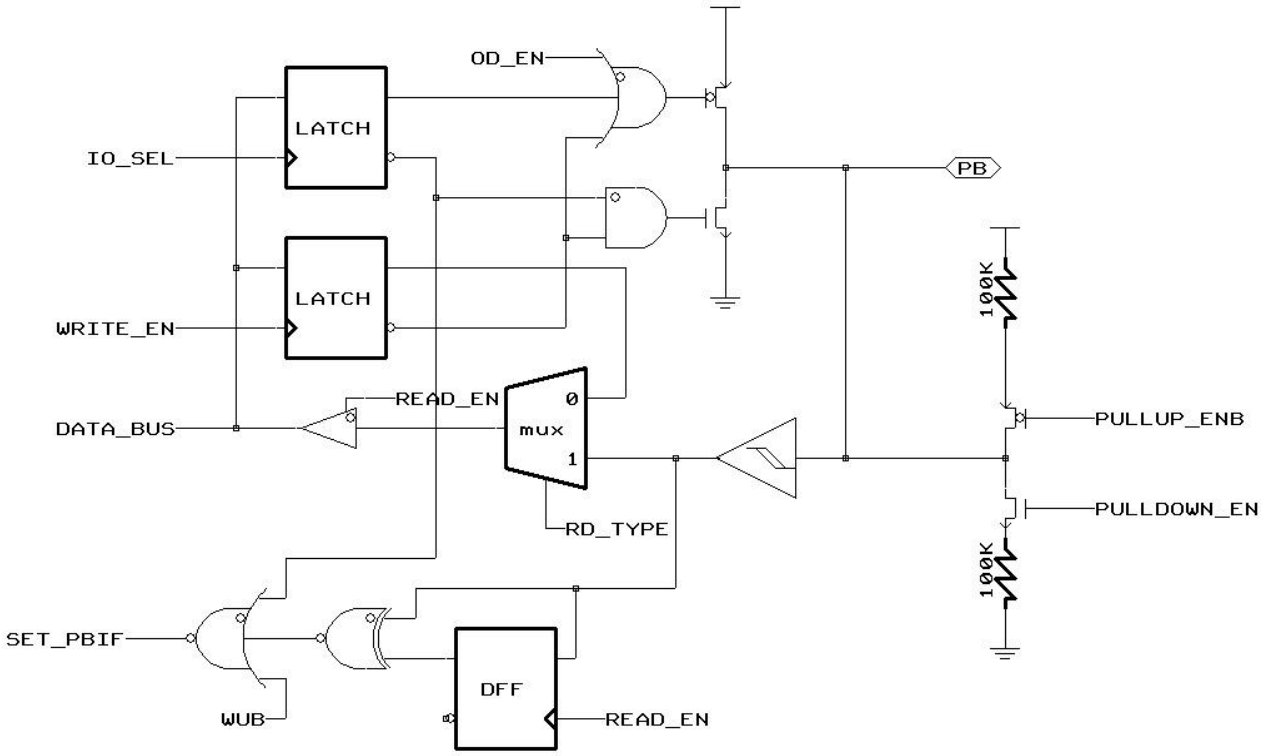


Figure 12 Block Diagram of PB2

IO_SEL: set pad attribute as input or output.

WRITE_EN: write data to pad.

READ_EN: read pad.

OD_EN: enable open-Drain.

PULLUP_ENB: enable Pull-High.

PULLDOWN_EN: enable Pull-Low.

RD_TYPE: select read pin or read latch.

WUB: port B wake-up enable.

SET_PBIF: port B wake-up flag.

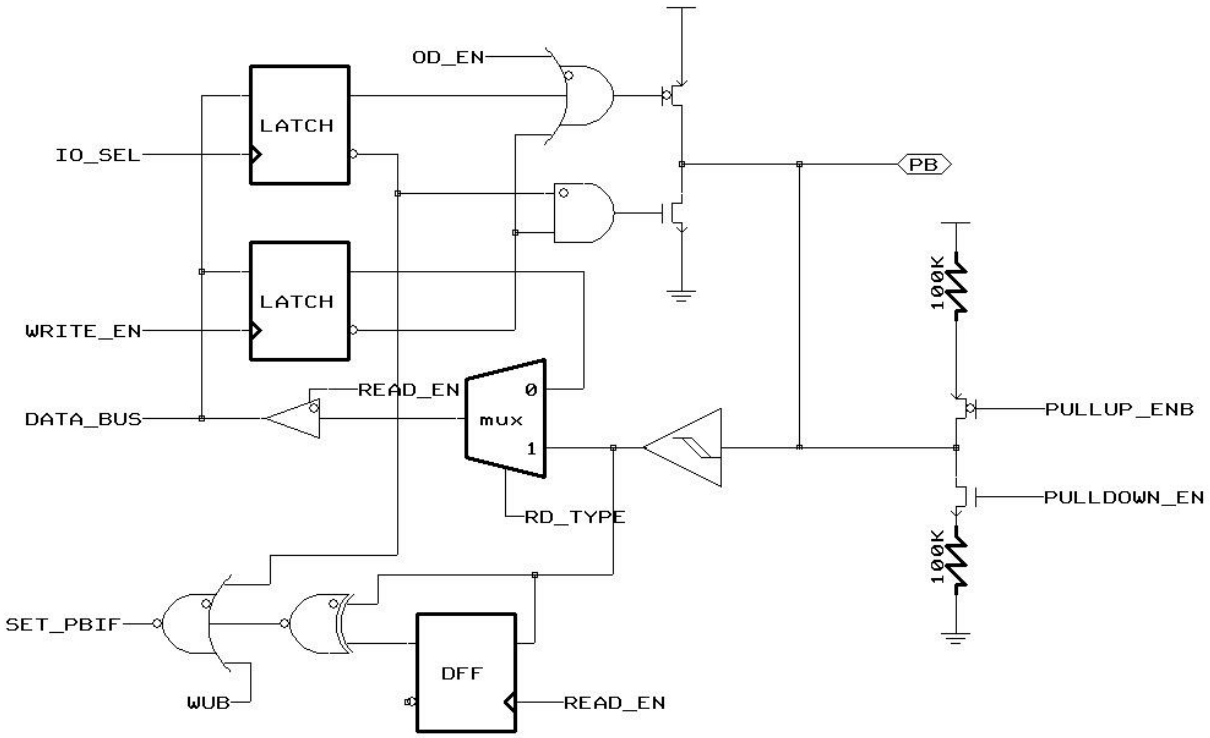


Figure 13 Block Diagram of PB3

3.6 Timer0

Timer0 is an 8-bit up-count timer and its operation is enabled by register bit T0EN (PCON1[0]). Writing to Timer0 will set its initial value. Reading from Timer0 will show its current count value.

The clock source to Timer0 can be from instruction clock or low speed clock Low Oscillator Frequency according to register bit T0CS and LCK_TM0 (T0MD[5] and T0MD[7]). When T0CS is 0, instruction clock is selected as Timer0 clock source. When T0CS is 1 and LCK_TM0 is 1 (and Timer0 source must set to 1), Low Oscillator Frequency output is selected. Summarized table is shown below. (Also check Figure 15)

Timer0 clock source	T0CS	LCKTM0	Timer0 source	Low Oscillator Frequency
Instruction clock	0	X	X	X
EX_CKI0	1	0	X	X
		X	0	
I_LRC	1	1	1	0

Table 21 Summary of Timer0 clock source control

When using Low Oscillator Frequency as Timer0 clock source, it is suggested to use prescaler0 (see below descriptions) and the ratio set to more than 4, or missing count may happen.

Before Timer0 clock source is supplied to Timer0, it can be divided by Prescaler0 if register bit PS0WDT (T0MD[3]) is clear to 0. When writing 0 to PS0WDT by instruction, Prescaler0 is assigned to Timer0 and Prescaler0 will be clear after this instruction is executed. The dividing rate of Prescaler0 is determined by register bits PS0SEL[2:0] which is from 1:2 to 1:256.

When Timer0 is overflow, the register bit T0IF (INTF[0]) will be set to 1 to indicate Timer0 overflow event is occurred. If register bit T0IE (INTE[0]) and GIE are both set to 1, interrupt request will occur and interrupt service routine will be executed. T0IF will not be clear until firmware writes 0 to T0IF.

The block diagram of Timer0 and WDT is shown in the figure below.

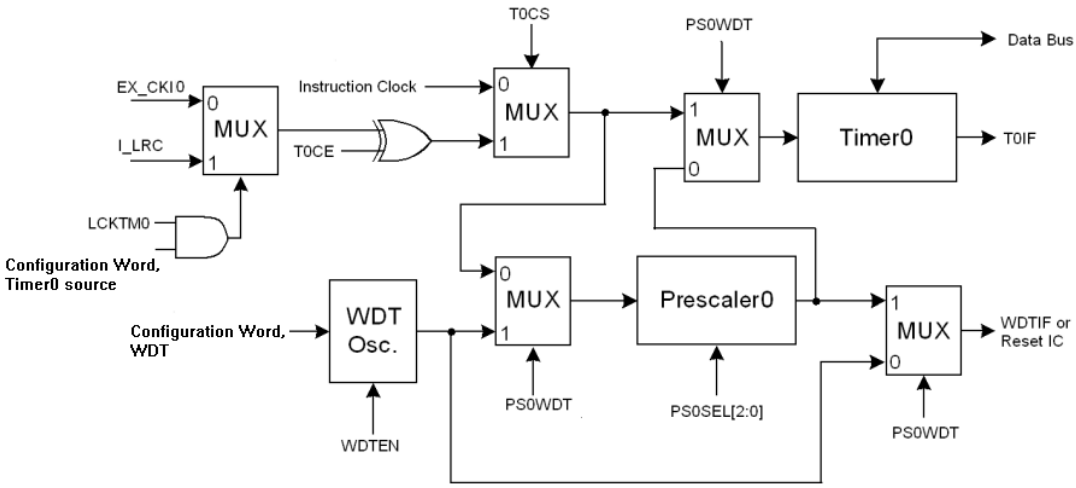


Figure 15 Block Diagram of Timer0 and WDT

3.7 Timer1 / PWM1 / Buzzer1

Timer1 is an 10-bit down-count timer with Prescaler1 whose dividing rate is programmable. The output of Timer1 can be used to generate PWM1 output and Buzzer1 output. Timer1 builds in auto-reload function and Timer1 reload register stores reload data with double buffers. When user write Timer1 reload register, write Timer1 MSB 2 bits(TMRH[5:4]) first and write TMR1 second, Timer1 reload register will be updated to Timer1 counter after Timer1 overflow occurs when T1EN=1. If T1EN=0, Timer1 reload register will be updated to Timer1 counter after write TMR1 immediately. A read to the Timer1 will show the content of the Timer1 current count value.

The block diagram of Timer1 is shown in the figure below.

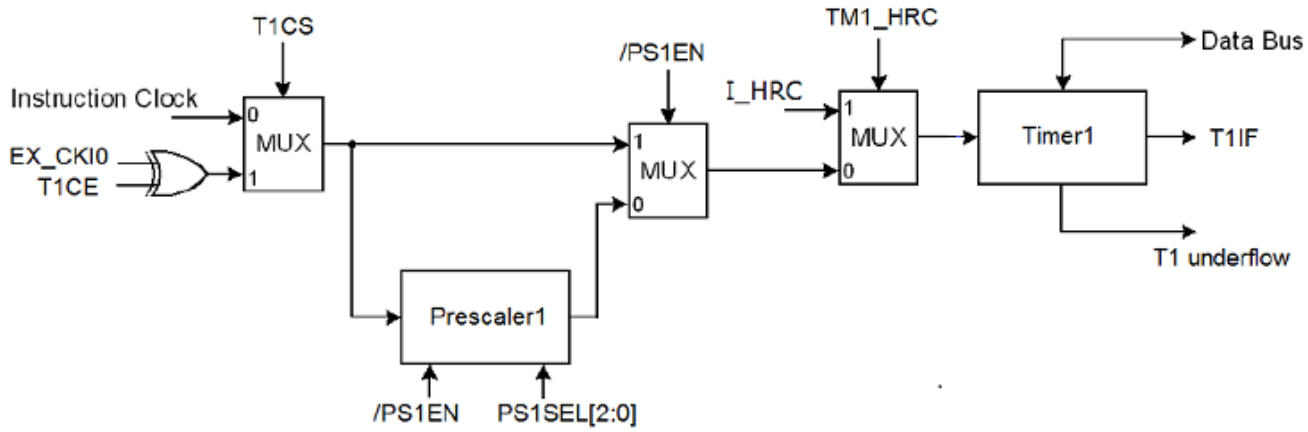


Figure 16 Block Diagram of Timer1

The operation of Timer1 can be enabled or disabled by register bit T1EN (T1CR1[0]). After Timer1 is enabled, its clock source can be instruction clock or pin EX_CKI0 which is determined by register bit T1CS (T1CR2[5]). When T1CS is 1, EX_CKI0 is selected as clock source. When T1CS is 0, instruction clock is selected as clock source. When EX_CKI0 is selected, the active edge to decrease Timer1 is determined by register bit T1CE (T1CR2[4]). When T1CE is 1, high-to-low transition on EX_CKI0 will decrease Timer1. When T1CE is 0, low-to-high transition on EX_CKI0 will decrease Timer1. The selected clock source can be divided further by Prescaler1 before it is applied to Timer1. Prescaler1 is enabled by writing 0 to register bit /PS1EN (T1CR2[3]) and the dividing rate is from 1:2 to 1:256 determined by register bits PS1SEL[2:0] (T1CR2[2:0]). Current value of Prescaler1 can be obtained by reading register PS1CV.

Timer1 provides two kinds of operating mode: one is One-Shot mode and the other is Non-Stop mode. When register bit T1OS (T1CR1[2]) is 1, One-Shot mode is selected. Timer1 will count down once from initial value stored on register TMR1[9:0] to 0x00, i.e. underflow is occurred. When register bit T1OS (T1CR1[2]) is 0, Non-Stop mode is selected. When underflow is occurred, there are two selections to start next down-count which is determined by register bit T1RL (T1CR1[1]). When T1RL is 1, the initial value stored on register TMR1[9:0] will be restored and start next down-count from this initial value. When T1RL is 0, Timer1 will start next down-count from 0x3FF.

When Timer1 is underflow, the register bit T1IF (INTF[3]) will be set to 1 to indicate Timer1 underflow event is occurred. If register bit T1IE (INTE[3]) and GIE are both set to 1, interrupt request will occur and interrupt service routine will be executed. T1IF will not be clear until firmware writes 0 to T1IF.

The timing chart of Timer1 is shown in the following figure.

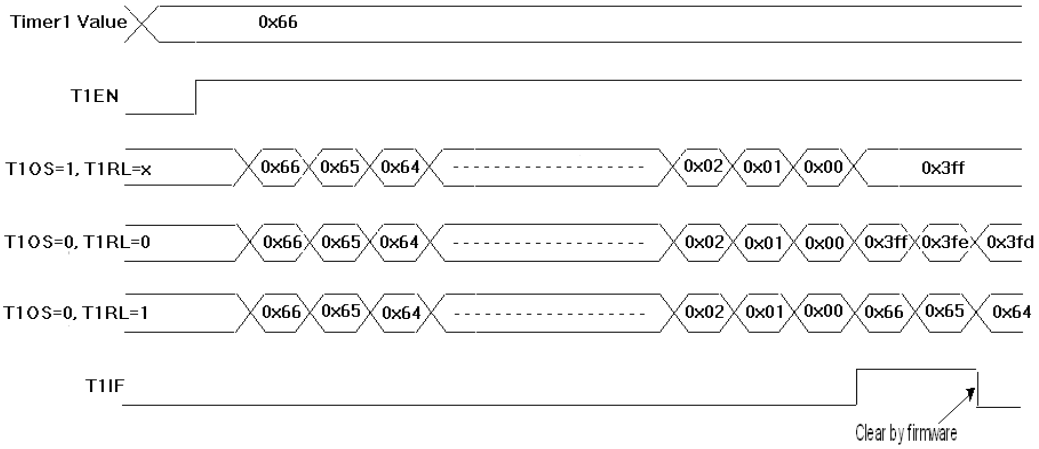


Figure 17 Timer1 Timing Chart

The PWM1 output can be available on I/O pin PB3 when register bit PWM1OEN (T1CR1[7]) is set to 1. Moreover, PB3 will become output pin automatically. The active state of PWM1 output is determined by register bit PWM1OAL (T1CR1[6]). When PWM1OAL is 1, PWM1 output is active low. When PWM1OAL is 0, PWM1 output is active high. Moreover, the duty cycle and frame rate of PWM1 are both programmable. The duty cycle is determined by registers TMRH[1:0] and PWM1DUTY[7:0]. When PWM1DUTY is 0, PWM1 output will be never active. When PWM1DUTY is 0x3FF, PWM1 output will be active for 1023 Timer1 input clocks. The frame rate is determined by TMRH[5:4] + TMR1[7:0] initial value. Therefore, PWM1DUTY value must be less than or equal to TMRH[5:4] + TMR1[7:0]. When user write PWM1DUTY, write PWM1DUTY[9:8] MSB 2 bits(TMRH[1:0]) first and write PWM1DUTY[7:0] second, PWM1 duty register will be updated after Timer1 overflow occurs. The block diagram of PWM1 is illustrated in the following figure.

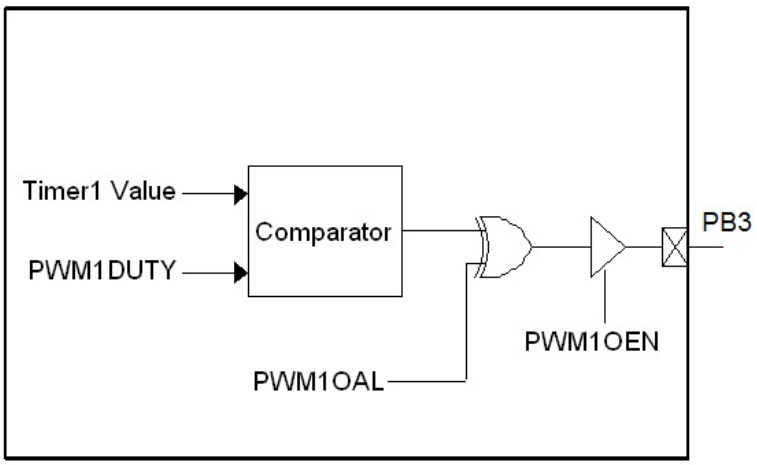


Figure 18 PWM1 Block Diagram

The Buzzer1 output (BZ1) can be available on I/O pin PB3 when register bit BZ1EN (BZ1CR1[7]) is set to 1. Moreover, PB3 will become output pin automatically. The frequency of BZ1 can be derived from Timer1 output or Prescaler1 output and dividing rate is determined by register bits BZ1FSEL[3:0] (BZ1CR[3:0]). When BZ1FSEL[3] is 0, Prescaler1 output is selected to generate BZ1 output. When BZ1FSEL[3] is 1, Timer1 output is selected to generate BZ1 output. The dividing rate can be from 1:2 to 1:256 in order to generate all kinds of frequency. The block diagram of Buzzer1 is illustrated in the following figure.

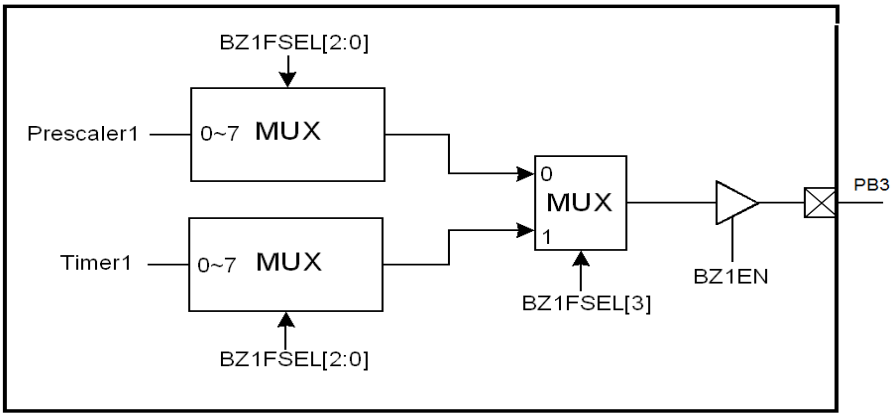


Figure 19 Buzzer1 Block Diagram

Note: When PWM1 and Buzzer1 are both enabled, PWM1 will have the higher priority for PB3 output.

3.8 PWM2

The PWM2 output can be available on I/O pin PB2 when register bit PWM2OEN (P2CR1[7]) is set to 1. Moreover, PB2 will become output pin automatically. The active state of PWM2 output is determined by register bit PWM2OAL (T2CR1[6]). When PWM2OAL is 1, PWM2 output is active low. When PWM2OAL is 0, PWM2 output is active high. Moreover, the duty cycle and frame rate of PWM2 are both programmable. The duty cycle is determined by register TMRH[3:2],PWM2DUTY[7:0]. When PWM2DUTY is 0, PWM2 output will be never active. When PWM2DUTY is 0x3FF, PWM2 output will be active for 1023 Timer1 input clocks. The frame rate is determined by TMRH[5:4] +TMR1[7:0] initial value. Therefore, PWM2DUTY value must be less than or equal to TMRH[5:4] + TMR1[7:0]. When user write PWM2DUTY, write PWM2DUTY[9:8] MSB 2 bits(TMRH[3:2]) first and write PWM2DUTY[7:0] second, PWM2 duty register will be updated after Timer1 overflow occurs. The block diagram of PWM2 is illustrated in the following figure.

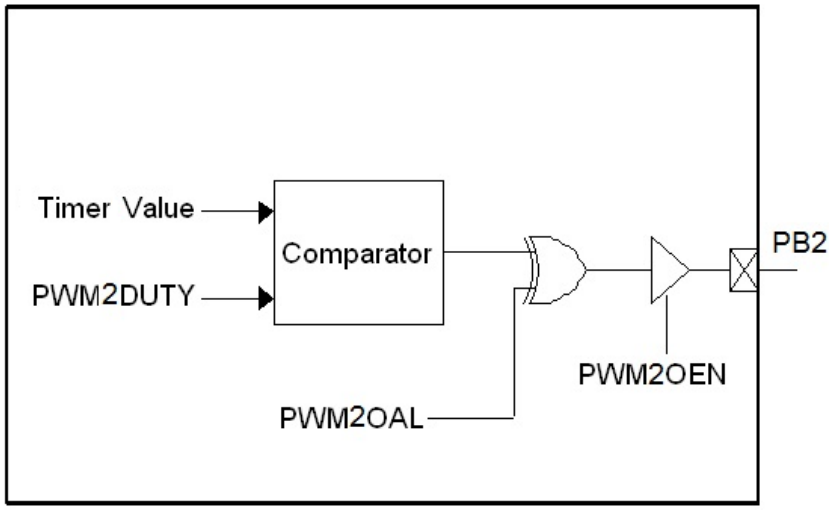


Figure 22 PWM2 Block Diagram

3.9 PWM3

The PWM3 output can be available on I/O pin PA2 when register bit PWM3OEN (P3CR1[7]) is set to 1. Moreover, PA2 will become output pin automatically. The active state of PWM3 output is determined by register bit PWM3OAL (T3CR1[6]). When PWM3OAL is 1, PWM3 output is active low. When PWM3OAL is 0, PWM3 output is active high. Moreover, the duty cycle and frame rate of PWM3 are both programmable. The duty cycle is determined by register TM3RH[1:0],PWM3DUTY[7:0]. When PWM3DUTY is 0, PWM3 output will be never active. When PWM3DUTY is 0x3FF, PWM3 output will be active for 1023 Timer1 input clocks. The frame rate is determined by TMRH[5:4] + TMR1[7:0] initial value. Therefore, PWM3DUTY value must be less than or equal to TMRH[5:4] + TMR1[7:0]. When user write PWM3DUTY, write PWM3DUTY[9:8] MSB 2 bits(TM3RH[1:0]) first and write PWM3DUTY[7:0] second, PWM3 duty register will be updated after Timer1 overflow occurs. The block diagram of PWM3 is illustrated in the following figure.

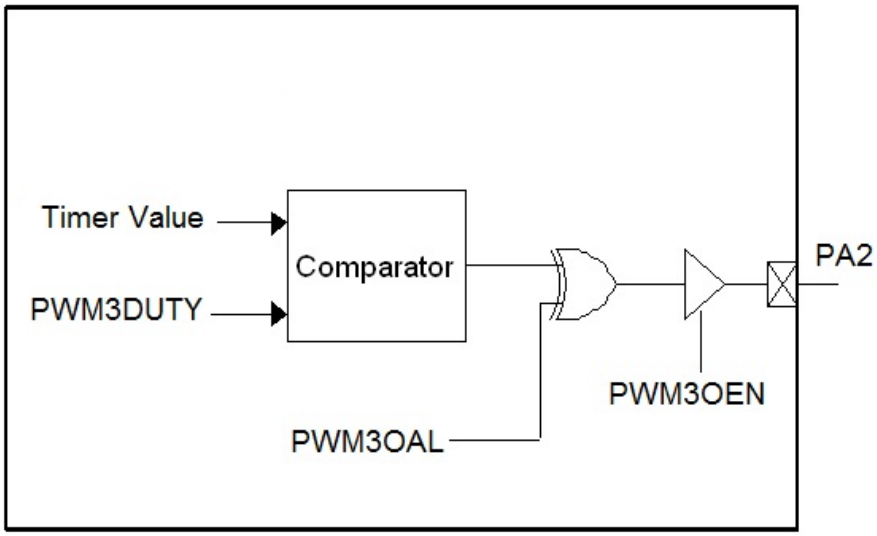


Figure 26 PWM3 Block Diagram

3.10 RFC Mode

NY8B060D has built-in RFC mode. Once RFC mode is enabled, the selected input pad state will take control of the Timer1 counting. When the selected input pad is recognized as 0 state (The input pad voltage is smaller than V_{IL}), Timer1 keeps counting. When this selected pad is recognized as 1 (The input pad voltage is larger than V_{IH}) , Timer1 stops counting. The following figure shows how RFC mode operates: PSEL3~0 is used to select one RFC input pad out of 6 NY8B060D pads. RFCEN is used to switch the Timer1 enable signal between the normal enable signal T1EN and RFC selected input state.

One application of RFC mode is to measure the capacitor-resistor charging time, As the figure shows, when PSEL3~0=0x02, PA2 is selected as RFC input pad. At first the PA2 is set as output low (the voltage of PA2 is discharged to 0). Next step, clear Timer1 content, set PA2 as input and enable RFC mode. Then Timer1 will start counting, and the RC circuit will start charging PA2. As PA2 is charged to the V_{IH} voltage, the Timer1 counting is stopped because PA2 input is high. The Timer1 content will show the RC circuit charging time. (Note: Timer1 is down-count.)

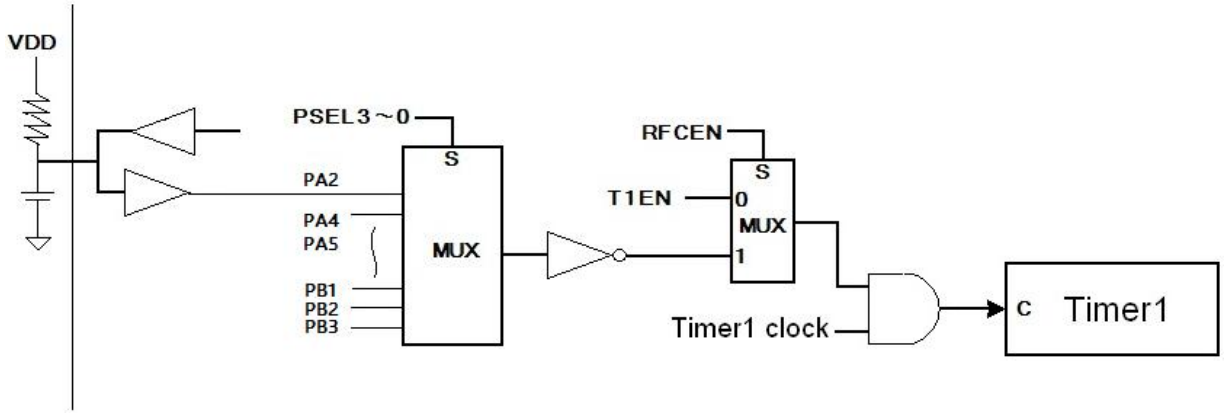


Figure 28 RFC Block Diagram

3.11 IR Carrier

The IR carrier will be generated after register bit IREN (IRCR[0]) is set to 1. Moreover, PB1 will become output pin automatically. When IREN is clear to 0, PB1 will become general I/O pin as it was configured.

The IR carrier frequency is selectable by register bit IRF57K (IRCR[1]). When IRF57K is 1, IR carrier frequency is 57KHz. When IRF57K is 0, IR carrier frequency is 38KHz. Because IR carrier frequency is derived from high frequency system oscillation F_{HOSC}, it is necessary to specify what frequency is used as system oscillation when external crystal is used. When internal high frequency oscillation is adopted, this register will be ignored, and it will provide 4MHz clock to IR module.

The active state (polarity) of IR carrier is selectable according to PB1 output data. When register bit IRCSEL (IRCR[2]) is 1, IR carrier will be present on pin PB1 when its output data is 0. When register bit IRCSEL

(IRCR[2]) is 0, IR carrier will be present on pin PB1 when its output data is 1. The polarity of IR carrier is shown in the following figure.

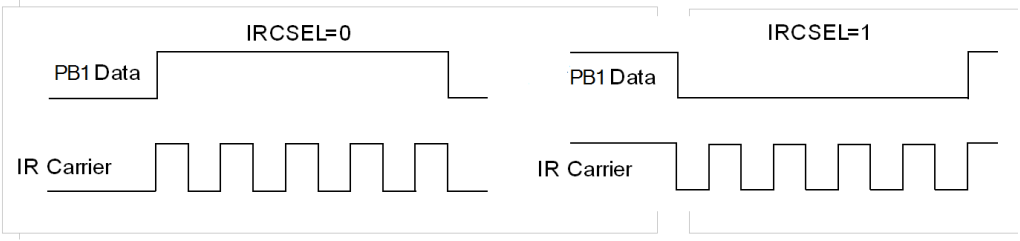


Figure 29 Polarity of IR Carrier vs. Output Data

3.12 Low Voltage Detector (LVD)

NY8B060D low voltage detector (LVD) built-in precise band-gap reference for accurately detecting V_{DD} level. If LVDEN (register PCON[5]) =1 and V_{DD} voltage value falls below LVD voltage which is selected by LVDS[3:0] as table shown below, the LVD output will become low. If the LVD interrupt is enabled, the LVD interrupt flag will be high and if GIE=1 it will force the program to execute interrupt service routine. Moreover, LVD real-state output can be polled by register PCON1[6]. The following is LVD block diagram:

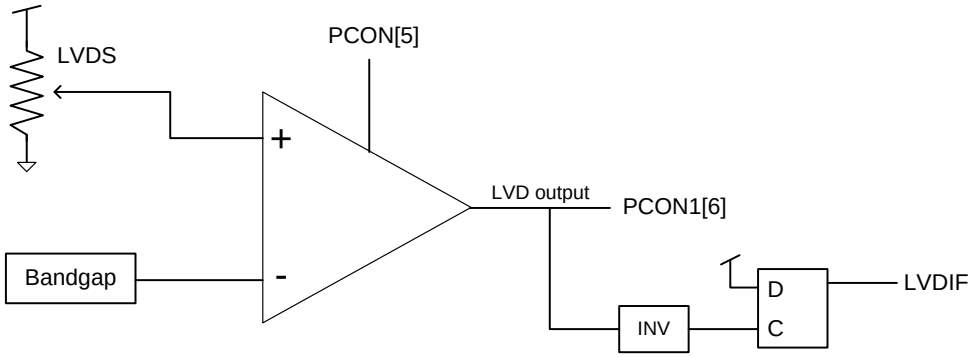


Figure 19 LVD block diagram

The following table is LVD voltage select table.

LVDS[3:0]	Voltage
0000	1.9V
0001	2.0V
0010	2.2V
0011	2.4V
0100	2.6V
0101	2.8V
0110	2.9V
0111	3.0V
1000	3.15V

LVDS[3:0]	Voltage
1001	3.30V
1010	3.45V
1011	3.60V
1100	3.75V
1101	3.90V
1110	4.05V
1111	4.15V

Table 12 LVD voltage select

Note:

The hysteresis voltage (from low to high) of LVD is about 0.1V.

In battery charging applications (detected voltage is from low to high),

the LVD voltage select table should be as followed:

LVDS[3:0]	Voltage
0000	--
0001	--
0010	(2.2+0.1) V
0011	(2.4+0.1) V
0100	(2.6+0.1) V
0101	(2.8+0.1) V
0110	(2.9+0.1) V
0111	(3.0+0.1) V
1000	(3.15+0.1) V
1001	(3.30+0.1) V
1010	(3.45+0.1) V
1011	(3.60+0.1) V
1100	(3.75+0.1) V
1101	(3.90+0.1) V
1110	(4.05+0.1) V
1111	(4.15+0.1) V

The LVD control flow is as the following:

Step1: Select LVD voltage by LVDS[3:0]

Step2: Set CMPCR = 0x0A

Step3: Set PCON[5]=1 (enable LVD)

Step4: Check LVD status by PCON1[6]

Note: If LVD voltage LVDS[3:0] is changed, user must wait at least 50us(@F_{HOSC}=1MHz) to get correct LVD status by PCON1[6]

3.13 Voltage Comparator

NY8B060D provides voltage comparator and internal reference voltage with various analog comparing mode. The comparator non-inverting and inverting input can share with GPIO.

CMPEN (register PCON[2]) is used to enable and disable comparator. When CMPEN=0(default), comparator is disabled. When CMPEN=1, the comparator is enabled. In halt mode the comparator is disabled automatically.

The structure of comparator is shown in the following figure:

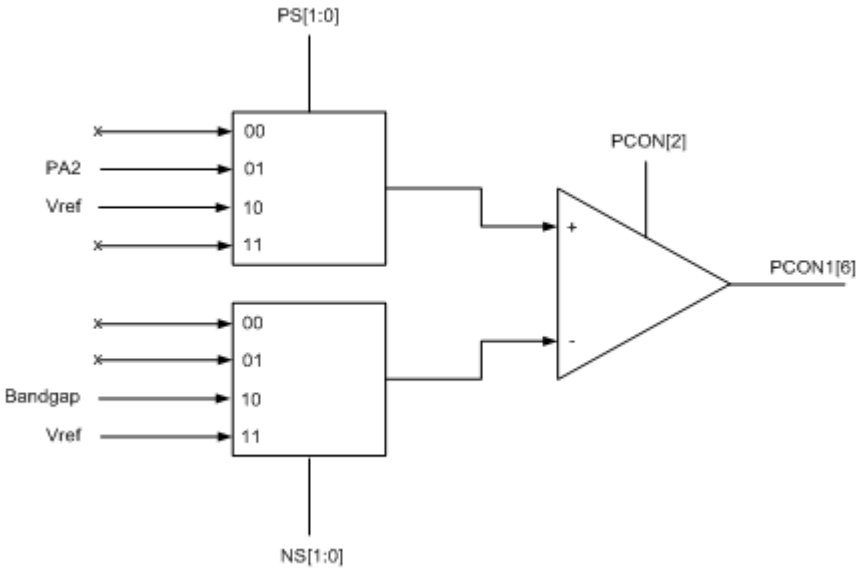


Figure 20 Comparator block diagram

3.13.1 Comparator Reference Voltage (Vref)

The internal reference voltage Vref is built by series resistance to provide different level of reference voltage. RBIAS_H and RBIAS_L are used to select the maximum and minimum values of Vref, and LVDS[3:0] are used to select one of 16 voltage levels.

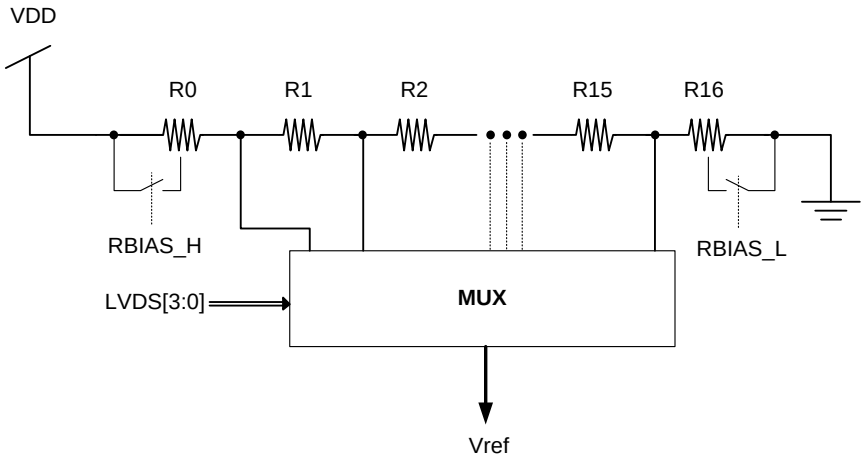


Figure 21 Vref hardware connection

The **Vref** is determined by RBIAS_H, RBIAS_L and LVDS[3:0]. The LVDS[3:0] is used to select one out of 16 reference voltages, the table shown below.

LVDS[3:0]	RBIAS_H=1 RBIAS_L=0	RBIAS_H=0 RBIAS_L=1
0000	67/128 V_{DD}	34/128 V_{DD}
0001	64/128 V_{DD}	32/128 V_{DD}
0010	59/128 V_{DD}	28/128 V_{DD}
0011	54/128 V_{DD}	25/128 V_{DD}
0100	50/128 V_{DD}	22/128 V_{DD}
0101	47/128 V_{DD}	20/128 V_{DD}
0110	45/128 V_{DD}	19/128 V_{DD}
0111	44/128 V_{DD}	17/128 V_{DD}
1000	42/128 V_{DD}	16/128 V_{DD}
1001	40/128 V_{DD}	15/128 V_{DD}
1010	38/128 V_{DD}	14/128 V_{DD}
1011	37/128 V_{DD}	13/128 V_{DD}
1100	35/128 V_{DD}	12/128 V_{DD}
1101	34/128 V_{DD}	11/128 V_{DD}
1110	33/128 V_{DD}	10/128 V_{DD}
1111	32/128 V_{DD}	10/128 V_{DD}

Table 13 The reference voltage Vref selection table

Note: The deviation of Vref is ±0.1V.

The non-inverting input of the comparator is determined by PS[1:0] (register CMPCR[3:2]).

The table is shown below

PS[1:0]	Non-inverting input
00	--
01	PA2
10	Vref
11	---

Table 14 Non-inverting input select

The inverting input of the comparator is determined by NS[1:0] (register CMPCR[1:0]).

The table is shown below

NS[1:0]	Inverting input
00	--
01	--
10	Bandgap (0.65V)
11	Vref

Table 15 Inverting input select

There are two ways to get the comparator output result: one is through register polling, the other is through probing output pad.

Comparator output can be polled by LVDOUT (register PCON1[6]).

To probe comparator output at output pad, set CMPOE (register OSCCR[6]) to 1, then PB1 will be the real-time state of the comparator output. It is noted that when CMPOE=1, the PWM3 function will be disabled if it is enabled.

3.14 Analog-to-Digital Convertor (ADC)

NY8B060D provide 6+1 channel 12-bit SAR ADC to transfer analog signal into 12-bits digital data. The ADC high reference voltage is selectable. They can be internal generated voltage VDD, 4V, 3V or 2V. The Analog input is selected from analog signal input pin PA2, PA4, PB1~PB3 or from internal generated 1 / 4 *VDD. The ADC clock ADCLK can be selected to be F_{INST}/1, F_{INST}/2, F_{INST}/8 or F_{INST}/16. The Sampling pulse width can be selected to be ADCLK*1, ADCLK*2, ADCLK*4 or ADCLK*8. Set ADEN=1 before ADC take into operation. Then set START=1, the ADC will start to convert analog signal to digital. EOC=0 means ADC is in processing. EOC=1 indicate ADC is at end of conversion. If ADIE=1 and global interrupt is enabled, the ADC interrupt will issue after EOC low go high. The block diagram is as following figure.

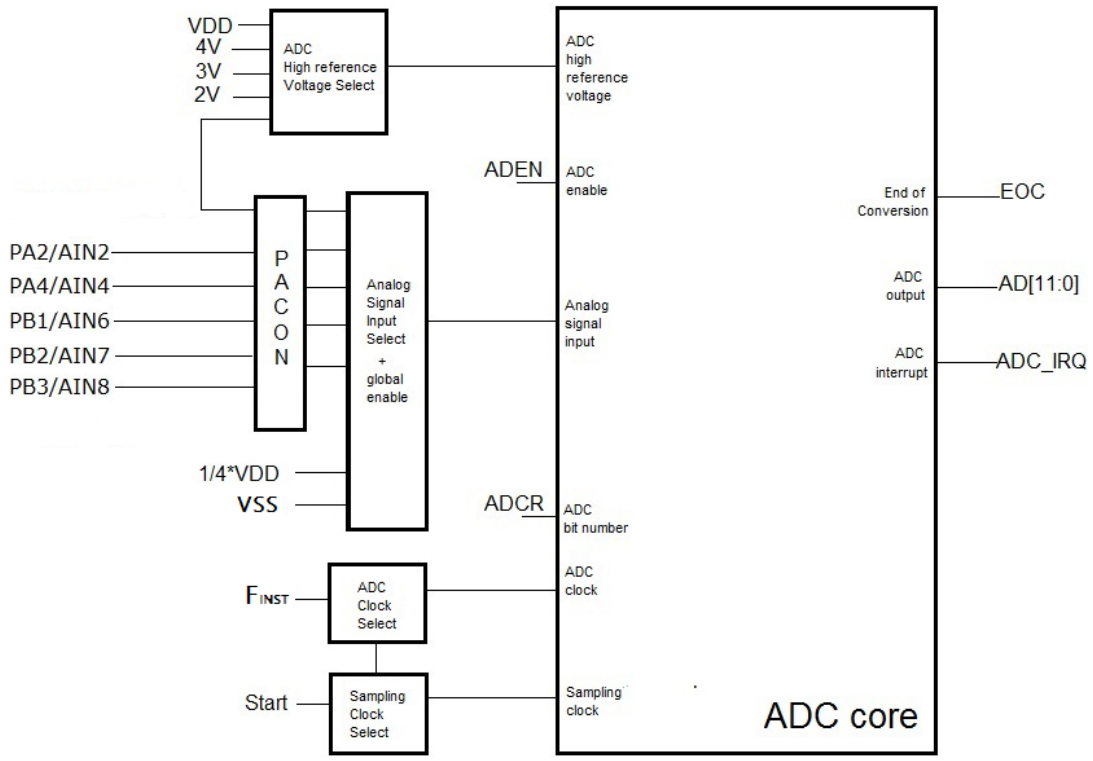


Figure 33 ADC block diagram

3.14.1 ADC reference voltage

ADC is built-in four high reference voltage source controlled by ADVREFH register. These high reference voltage sources are internal voltage source (VDD, 4V, 3V, 2V). If EVHENB bit is 0, ADC reference voltage is from internal voltage source selected by VHS[1:0]. If VHS[1:0] is “11”, ADC reference voltage is VDD. If VHS[1:0] is “10”, ADC reference voltage is 4V. If VHS[1:0] is “01”, ADC reference voltage is 3V. If VHS[1:0] is “00”, ADC reference voltage is 2V. The limitation of internal reference voltage application is VDD can't below each of internal voltage level, or the level is equal to VDD. ADC sampling voltage range is limited by high/low reference voltage. The ADC low reference voltage is VSS and not changeable. The ADC high reference voltage includes internal VDD/4V/3V/2V. The ADC reference voltage range limitation is (ADC high reference voltage – low reference voltage) ≥ 2V. ADC low reference voltage is VSS=0V. So ADC high reference voltage range is 2V ~ VDD.

ADC analog input signal voltage must be from ADC low reference voltage to ADC high reference voltage. If the ADC analog input signal voltage is over this range, The ADC converting result is unexpected (full scale or zero).

EVHENB	VHS[1:0]	Reference voltage
1	x x	--
0	1 1	VDD

0	1 0	4V
0	0 1	3V
0	0 0	2V

Table 26 ADC reference voltage select

3.14.2 ADC analog input channel

ADC use CHS[3:0] and GCHS to select analog input source. GCHS is global channel select. Namely, GCHS must be 1 before any analog input source can be selected and converted.

GCHS	CHS[3:0]	ADC analog input source
0	xxxx	x
1	0010	PA2
1	0100	PA4
1	0110	PB1
1	0111	PB2
1	1000	PB3
1	1011	1 / 4 * VDD
1	1100	GND

Table 27 ADC analog input source select

ADC input pins are shared with digital I/O pins. Connect an analog signal to these pin may cause extra current leakage in I/O pins. In the power down mode, the above leakage current will be a big problem. PACON[2,4] are PA[2,4] configuration register, PACON[6:7] are PB[1:2] configuration register and ADCR[4] are PB[3] configuration register to solve above problem. Write “1” to PACON and ADCR[4:6] will configure related PA /PB pin as pure analog input pin to avoid current leakage, and it can't be use as normal I/O.

Except setting the PACON and ADCR register, the selected analog input pin must be set as input mode and the internal pull-high / pull-down must be disabled, otherwise the analog input level may be affected.

3.14.3 ADC clock (ADCLK), sampling clock (SHCLK) and bit number

Conversion speed and conversion accuracy are affected by the selection of the ADC clock (ADCLK), sampling pulse width (SHCLK) and conversion bit number. ADCLK is the base clock of ADC. During the operation of SAR ADC, bit operation is synchronized with ADCLK. SHCLK is the duration of analog signal sampling time, larger SHCLK will restore original analog signal level more closely but it will slow down the ADC conversion speed. Vise versa. The ADC can select different conversion bit number which is depended on ADCR[1:0] register bits. There are 3 types to select, which is 12-bit, 10-bit and 8-bit. Less conversion bit number will speed up the ADC conversion rate but the effective ADC bit is less. More conversion bit number will slow down the conversion rate but the accuracy is more.

The ADC clock is derived from F_{INST} and is selectable from ADCK[1:0].

ADCK[1:0]	ADC clock
0 0	F_{INST}/16
0 1	F_{INST}/8
1 0	F_{INST}/1
1 1	F_{INST}/2

Table 28 ADC clock select

The Sampling clock width is derived from ADCLK and is selectable from SHCK[1:0].

SHCK[1:0]	Sampling clock
0 0	1 ADCLK
0 1	2 ADCLK
1 0	4 ADCLK
1 1	8 ADCLK

Table 29 ADC sampling clock select

ADC bit number select is from ADCR[1:0].

ADCR[1:0]	Conversion bit number
0 0	8-bit
0 1	10-bit
1 x	12-bit

Table 30 conversion bit number select

The ADC converting time is from START(Start to ADC convert) to EOC=1 (End of ADC convert). The duration is depending on ADC resolution and ADC clock rate and sampling clock width.

ADC conversion time ≈ sampling clock width + (ADC bit number + 2) * ADCLK width.

The following table is some example conversion time and conversion rate of ADC.

Bit No.	ADC clock	SHCLK	Conversion Time (ADCLK No.)	F_{INST}=2MHz		F_{INST}=250K	
				Time	Rate	Time	Rate
12	F_{INST}/16	8 ADCLK	22	176us	5.68kHz	1408us	710Hz
12	F_{INST}/1	1 ADCLK	15	7.5us	133.3kHz	60us	16.7kHz
10	F_{INST}/1	1 ADCLK	13	6.5us	153.8kHz	52us	19.2kHz
8	F_{INST}/1	1 ADCLK	11	5.5us	181.8kHz	44us	22.7kHz

Table 31 ADC Conversion time

3.14.4 ADC operation

Set ADC clock (ADCLK), sampling clock width (SHCLK), conversion bit number (ADCR), ADC high reference voltage (ADVREFH), select input channel and PACON related bit. Then set ADEN=1.

After setting ADEN=1, it must wait at least 256us (ADC internal bias stable time) before ADC can operate. Write START to 1 to start an ADC conversion session. During ADC is in processing EOC=0. Polling EOC=1 or wait for ADC interrupt at the end of ADC conversion.

3.15 Watch-Dog Timer (WDT)

There is an on-chip free-running oscillator in NY8B060D which is used by WDT. As this oscillator is independent of other oscillation circuits, WDT may still keep working during Standby mode and Halt mode.

WDT can be enabled or disabled by a configuration word. When WDT is enabled by configuration word, its operation still can be controlled by register bit WDTEN (PCON[7]) during program execution. Moreover, the mechanism after WDT time-out can reset NY8B060D or issue an interrupt request which is determined by another configuration word. At the same time, register bit /TO (STATUS[4]) will be clear to 0 after WDT time-out.

The baseline of WDT time-out period can be 3.5 ms, 15 ms, 60 ms or 250 ms which is determined by two configuration words. The time-out period can be lengthened if Prescaler0 is assigned to WDT. Prescaler0 will be assigned to WDT by writing 1 to register bit PS0WDT. The dividing rate of Prescaler0 for WDT is determined by register bits PS0SEL[2:0] and depends on WDT time-out mechanism. The dividing rate is from 1:1 to 1:128 if WDT time-out will reset NY8B060D and dividing rate is from 1:2 to 1:256 if WDT time-out will interrupt NY8B060D.

When Prescaler0 is assigned to WDT, the execution of instruction CLRWDT will clear WDT, Prescaler0 and set /TO flag to 1.

If user selects interrupt for WDT time-out mechanism, register bit WDTIF (INTF[6]) will set to 1 after WDT is expired. It may generate an interrupt request if register bit WDTIE (INTE[6]) and GIE both set to 1. WDTIF will not be clear until firmware writes 0 to WDTIF.

3.16 Interrupt

NY8B060D provides two kinds of interrupt: one is software interrupt and the other is hardware interrupt. Software interrupt is caused by execution of instruction INT. There are 11 hardware interrupts:

- Timer0 overflow interrupt.
- Timer1 underflow interrupt.
- WDT timeout interrupt.
- PA/PB input change interrupt.
- External 1 interrupt.
- LVD interrupt.
- ADC end-of-convert interrupt.

GIE is global interrupt enable flag. It has to be 1 to enable hardware interrupt functions. GIE can be set by ENI instruction and clear to 0 by DISI instruction.

After instruction INT is executed, no matter GIE is set or clear, the next instruction will be fetched from address 0x001. At the same time, GIE will be clear to 0 by NY8B060D automatically. This will prevent nested interrupt from happening. The last instruction of interrupt service routine of software interrupt has to be RETIE. Execution of this instruction will set GIE to 1 and return to original execution sequence.

While any of hardware interrupts is occurred, the corresponding bit of interrupt flag will be set to 1. This bit will not be clear until firmware writes 0 to this bit. Therefore user can obtain information of which event causes hardware interrupt by polling the corresponding bit of interrupt flag. Note that only when the corresponding interrupt enable bit is set to 1, will the corresponding interrupt flag be read. And if the corresponding interrupt enable bit is set to 1 and GIE is also 1, hardware interrupt will occur and next instruction will be fetched from 0x008. At the same time, the register bit GIE will be clear by NY8B060D automatically. If user wants to implement nested interrupt, instruction ENI can be used as the first instruction of interrupt service routine which will set GIE to 1 again and allow other interrupt events to interrupt NY8B060D again. Instruction RETIE has to be the last instruction of interrupt service routine which will set GIE to 1 and return to original execution sequence.

It should be noted that ENI instruction cannot be placed right before RETIE instruction because ENI instruction in interrupt service routine will trigger nested interrupt, but RETIE will clear internal interrupt processing after jump out of ISR, so it is possible for interrupt flag to be falsely cleared.

3.16.1 Timer0 Overflow Interrupt

Timer0 overflow (from 0x00 to 0xFF) will set register bit T0IF. This interrupt request will be serviced if T0IE and GIE are set to 1.

3.16.5 WDT Timeout Interrupt

When WDT is timeout and the configuration word selects WDT timeout will generate interrupt request, it will set register bit WDTIF. This interrupt request will be serviced if WDTIE and GIE are set to 1.

3.16.6 PA/PB Input Change Interrupt

When PAx, $0 \le x \le 7$, PBy, $0 \le y \le 5$ is configured as input pin and corresponding register bit WUPAx, WUPBx is set to 1, a level change on these selected I/O pin(s) will set register bit PABIF. This interrupt request will be serviced if PABIE and GIE are set to 1. Note when PB0 or PB1 is both set as level change interrupt and external interrupt, the external interrupt enable EIS0 or EIS1=1 will disable PB0 or PB1 level change operation.

3.16.7 External 1 Interrupt

According to the configuration of EIS1=1 and INTEDG, the selected active edge on I/O pin PB1 will set register bit INT1IF and this interrupt request will be served if INT1IE and GIE are set to 1.

3.16.9 LVD Interrupt

When V_{DD} level falls below LVD voltage, LVD flag will go from high to low, and set the register bit LVDIF=1. This interrupt request will be serviced if LVDIE and GIE are set to 1.

3.16.11 ADC end of conversion Interrupt

The ADC interrupt is triggered whenever an ADC end-of-convert signal is issued. This interrupt request will be serviced if ADIE and GIE are set to 1.

3.17 Oscillation Configuration

Because NY8B060D is a dual-clock IC, there are high oscillation (F_{HOSC}) and low oscillation (F_{LOSC}) that can be selected as system oscillation (F_{OSC}). The oscillator which could be used as F_{HOSC} is internal high RC oscillator (I_HRC). The oscillator which could be used as F_{LOSC} is internal low RC oscillator (I_LRC).

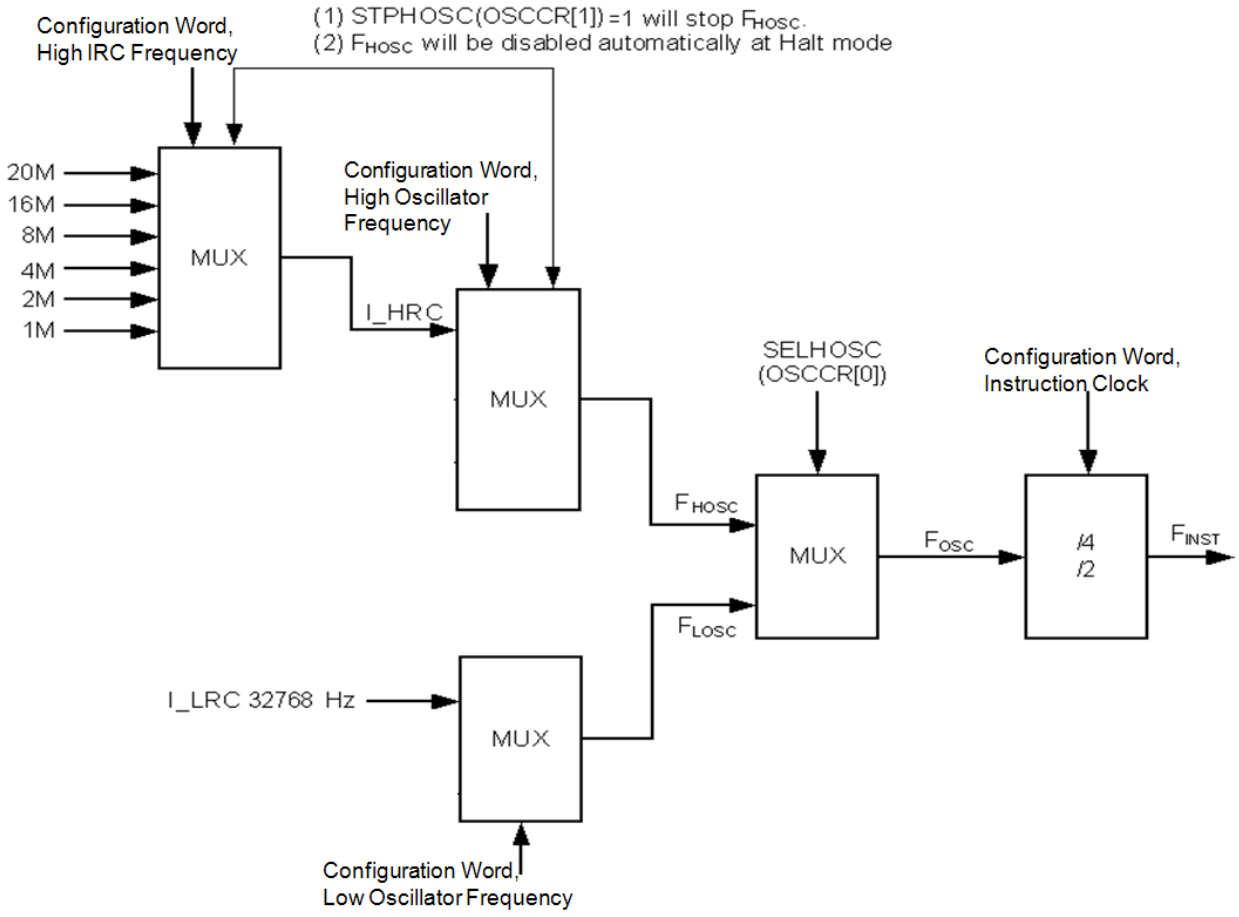


Figure 34 Oscillation Configuration of NY8B060D

There are two configuration words to determine which oscillator will be used as F_{HOSC}. When I_HRC is selected as F_{HOSC}, I_HRC output frequency is determined by three configuration words and it can be 1M, 2M, 4M, 8M, 16M or 20MHz.

There is one configuration word to determine which oscillator will be used as F_{LOSC}. When I_LRC is selected, its frequency is centered on 32768Hz.

The dual-clock combinations of F_{HOSC} and F_{LOSC} are listed below.

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

Table 32 Dual-clock combinations

Either F_{HOSC} or F_{LOSC} can be selected as system oscillation F_{OSC} according to the value of register bit SELHOSC (OSCCR[0]). When SELHOSC is 1, F_{HOSC} is selected as F_{OSC}. When SELHOSC is 0, F_{LOSC} is selected as F_{OSC}. Once F_{OSC} is determined, the instruction clock F_{INST} can be $F_{OSC}/2$ or $F_{OSC}/4$ according to value of a configuration word.

3.18 Operating Mode

NY8B060D provides four kinds of operating mode to tailor all kinds of application and save power consumptions. These operating modes are Normal mode, Slow mode, Standby mode and Halt mode. Normal mode is designated for high-speed operating mode. Slow mode is designated for low-speed mode in order to save power consumption. At Standby mode, NY8B060D will stop almost all operations except Timer0/Timer1 /WDT in order to wake-up periodically. At Halt mode, NY8B060D will sleep until external event or WDT trigger IC to wake-up. The block diagram of four operating modes is described in the following figure.

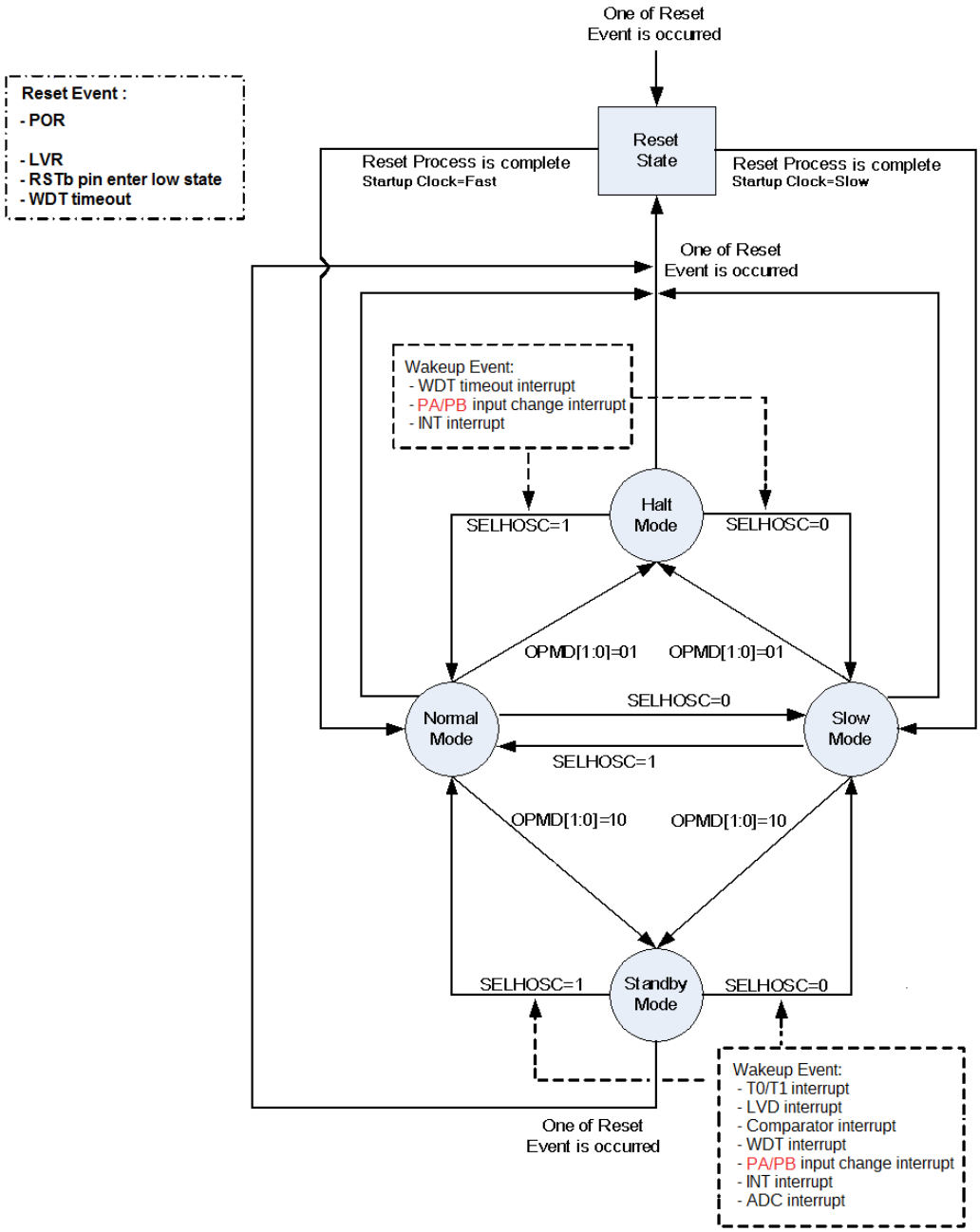


Figure 36 Four Operating Modes

3.18.1 Normal Mode

After any Reset Event is occurred and Reset Process is completed, NY8B060D will begin to execute program under Normal mode or Slow mode. Which mode is selected after Reset Process is determined by the Startup Clock configuration word. If Startup Clock=fast, NY8B060D will enter Normal mode, if Startup Clock=Slow, NY8B060D will enter Slow mode. At Normal mode, F_{HOSC} is selected as system oscillation in order to provide highest performance and its power consumption will be the largest among four operating modes. After power on or any reset trigger is released, NY8B060D will enter Normal mode after reset process is completed.

- Instruction execution is based on F_{HOSC} and all peripheral modules may be active according to corresponding module enable bit.
- The F_{LOSC} is still active and running.
- IC can switch to Slow mode by writing 0 to register bit SELHOSC (OSCCR[0]).
- IC can switch to Standby mode or Halt mode by programming register bits OPMD[1:0] (OSCCR[3:2]).
- For real time clock applications, the NY8B060D can run in normal mode, at the same time the low-frequency clock Low Oscillator Frequency connects to Timer0 clock. This is made possible by setting LCKTM0 to 1 and corresponding configuration word Timer0 source setting to 1.

3.18.2 Slow Mode

NY8B060D will enter Slow mode by writing 0 to register bit SELHOSC. At Slow mode, F_{LOSC} is selected as system oscillation in order to save power consumption but still keep IC running. However, F_{HOSC} will not be disabled automatically by NY8B060D. Therefore user can write 1 to register bit STPHOSC (OSCCR[1]) in slow mode to reduce power consumption further. But it is noted that it is forbidden to enter slow mode and stop F_{HOSC} at the same time, one must enter slow mode first, then disable $F_{HOSC,}$ or the program may hang on.

- Instruction execution is based on F_{LOSC} and all peripheral modules may be active according to corresponding module enable bit.
- F_{HOSC} can be disabled by writing 1 to register bit STPHOSC.
- IC can switch to Standby mode or Halt mode by programming register bits OPMD[1:0].
- IC can switch to Normal mode by writing 1 to SELHOSC.

3.18.3 Standby Mode

NY8B060D will enter Standby mode by writing 10b to register bits OPMD[1:0]. At Standby mode, however, F_{HOSC} will not be disabled automatically by NY8B060D and user has to enter slow mode and write 1 to register bit STPHOSC in order to stop F_{HOSC} oscillation. Most of NY8B060D peripheral modules are disabled but Timer can be still active if register bit T0EN/T1EN is set to 1. Therefore NY8B060D can wake-up after Timer0/Timer1 is expired. The expiration period is determined by the register TMR0/TMR1[9:0], F_{INST} and other configurations for Timer0/Timer1.

- Instruction execution is stop and some peripheral modules may be active according to corresponding module enable bit.
- F_{HOSC} can be disabled by writing 1 to register bit STPHOSC.
- The F_{LOSC} is still active and running.
- IC can wake-up from Standby mode if any of (a) Timer0/Timer1 (overflow/underflow) interrupt, (b) WDT timeout interrupt, (c) PA/PB input change interrupt or (d) INT external interrupt is happened.
- After wake-up from Standby mode, IC will return to Normal mode if SELHOSC=1, IC will return to Slow mode if SELHOSC=0.
- It is not recommended to change oscillator mode (normal to slow / slow to normal) and enter standby mode at the same time.

3.18.4 Halt Mode

NY8B060D will enter Halt mode by executing instruction SLEEP or writing 01b to register bits OPMD[1:0]. After entering Halt mode, register bit /PD (STATUS[3]) will be clear to 0, register bit /TO (STATUS[4]) will be set to 1 and WDT will be clear but keep running.

At Halt mode, all of peripheral modules are disabled, instruction execution is stop and NY8B060D can only wake-up by some specific events. Therefore, Halt mode is the most power saving mode provided by NY8B060D.

- Instruction execution is stop and all peripheral modules are disabled.
- F_{HOSC} and F_{LOSC} are both disabled automatically.
- IC can wake-up from Halt mode if any of (a) WDT timeout interrupt, (b) PA/PB input change interrupt or (c) INT or external interrupt is happened.
- After wake-up from Halt mode, IC will return to Normal mode if SELHOSC=1, IC will return to Slow mode if SELHOSC=0.

  Note: Users can change STPHOSC and enter Halt mode in the same instruction.
- It is not recommended to change oscillator mode (normal to slow or slow to normal) and enter halt mode at the same time.

3.18.5 Wake-up Stable Time

The wake-up stable time of Halt mode is determined by Configuration word: High Oscillator Frequency or Low Oscillator Frequency. 16*Fosc would be set as wake up period. On the other hand, there is no need of wake-up stable time for Standby mode because either F_{HOSC} or F_{LOSC} is still running at Standby mode.

Before NY8B060D enter Standby mode or Halt mode, user may execute instruction ENI. At this condition, NY8B060D will branch to address 0x008 in order to execute interrupt service routine after wake-up. If

instruction DISI is executed before entering Standby mode or Halt mode, the next instruction will be executed after wake-up.

3.18.6 Summary of Operating Mode

The summary of four operating modes is described in the following table.

Mode	Normal	Slow	Standby	Halt
F_{HOSC}	Enabled	STPHOSC	STPHOSC	Disabled
F_{LOSC}	Enabled	Enabled	Enabled	Disabled
Instruction Execution	Executing	Executing	Stop	Stop
Timer0/1	TxEN	TxEN	TxEN	Disabled
WDT	Option and WDTEN	Option and WDTEN	Option and WDTEN	Option and WDTEN
Other Modules	Module enable bit	Module enable bit	Module enable bit	All disabled
Wake-up Source	-	-	- Timer0/1 overflow - WDT timeout - PA/PB input change - INT1 - LVD interrupt - Comparator interrupt - ADC end-of-convert	- WDT timeout - PA/PB input change - INT1

Table 34 Summary of Operating Modes

3.19 Reset Process

NY8B060D will enter Reset State and start Reset Process when one of following Reset Event is occurred:

- Power-On Reset (POR) is occurred when V_{DD} rising is detected.
- Low-Voltage Reset (LVR) is occurred when operating V_{DD} is below pre-defined voltage.
- Pin RSTb is low state.
- WDT timeout reset.

Moreover, value of all registers will be initialized to their initial value or unchanged if its initial value is unknown. The status bits /TO and /PD could be initialized according to which event causes reset. The /TO and /PD value and its associated event is summarized in the table below.

Event	/TO	/PD
POR, LVR	1	1
RSTb reset from non-Halt mode	unchanged	unchanged
RSTb reset from Halt mode	1	1
WDT reset from non-Halt mode	0	1
WDT reset from Halt mode	0	0
SLEEP executed	1	0
CLRWDT executed	1	1

Table 35 Summary of /TO & /PD Value and its Associated Event

After Reset Event is released, NY8B060D will start Reset Process. It will wait certain amount of period for oscillation stable no matter what kind of oscillator is adopted. This period is called power-up reset time and is determined by three-bit configuration words which can be 140us, 4.5ms, 18ms, 72ms or 288ms. After power-up reset time, NY8B060D will wait for further oscillator start-up time (OST) before it starts to execute program. OST=1 clock cycle of F_{OSC} if the previous power-up time is 140us, OST=16 clock cycles of F_{OSC} if the previous power-up time is 4.5ms, 18ms, 72ms or 288ms.

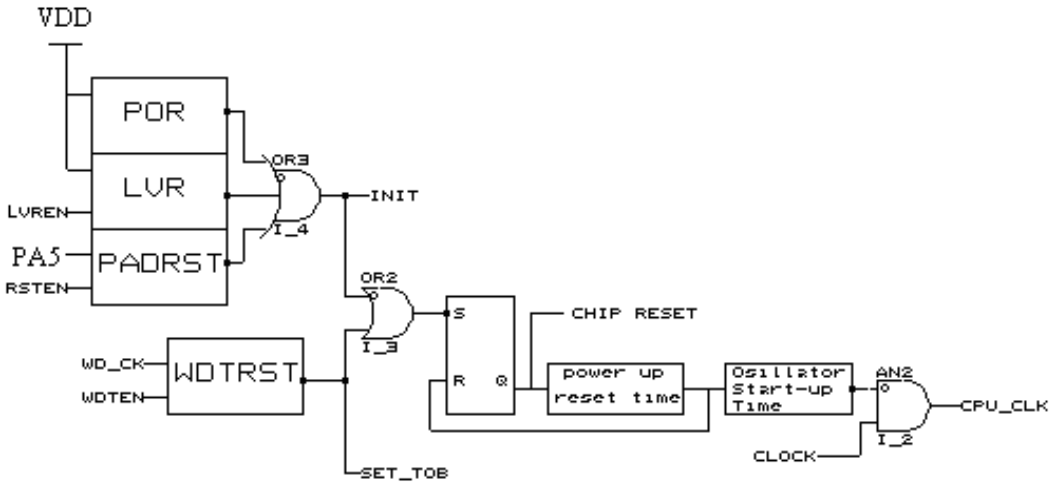


Figure 37 Block diagram of on-chip reset circuit

For slow V_{DD} power-up, it is recommended to use RSTb reset, as the following figure.

- It is recommended the R value should be not greater than 40KΩ.
- The R1 value=100Ω to 1KΩ will prevent high current, ESD or Electrical overstress flowing into reset pin.
- The diode helps discharge quickly when power down.

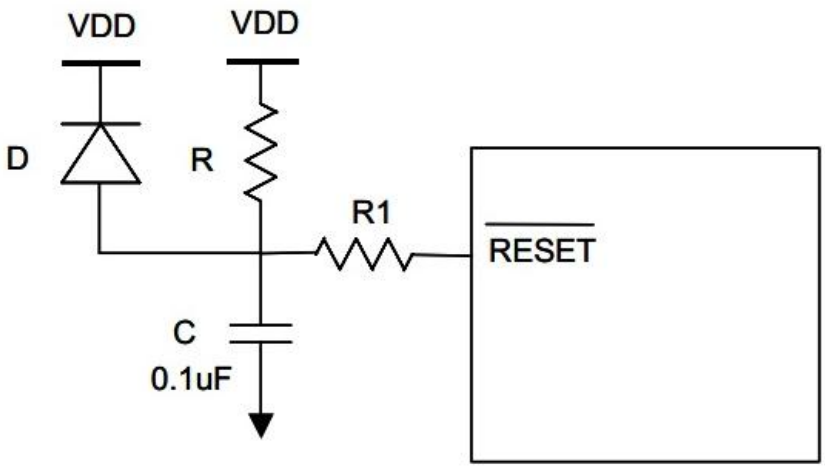


Figure 38 Block Diagram of Reset Application

4. Instruction Set

NY8B060D provides 55 powerful instructions for all kinds of applications.

Inst.	OP 1	OP 2	Operation	Cyc.	Flag
Arithmetic Instructions					
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
Conditional Instructions					
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	-
Data Transfer Instructions					
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	-

Inst.	OP 1	OP 2	Operation	Cyc.	Flag
Arithmetic Instructions					
ADDAR	R	d	dest = R + ACC	1	Z, DC, C
SUBAR	R	d	dest = R + (~ACC)	1	Z, DC, C
ADCAR	R	d	dest = R + ACC + C	1	Z, DC, C
SBCAR	R	d	dest = R + (~ACC) + C	1	Z, DC, C
ADDIA	i		ACC = i + ACC	1	Z, DC, C
SUBIA	i		ACC = i + (~ACC)	1	Z, DC, C
ADCIA	i		ACC = i + ACC + C	1	Z, DC, C
SBCIA	i		ACC = i + (~ACC) + C	1	Z, DC, C
DAA			Decimal adjust for ACC	1	C
CMPAR	R		Compare R with ACC	1	Z, C
CLRA			Clear ACC	1	Z
CLRR			Clear R	1	Z
Other Instructions					
NOP			No operation	1	-
SLEEP			Go into Halt mode	1	/TO, /PD
CLRWDT			Clear Watch-Dog Timer	1	/TO, /PD
ENI			Enable interrupt	1	-
DISI			Disable interrupt	1	-
INT			Software Interrupt	3	-
RET			Return from subroutine	2	-
RETIE			Return from interrupt and enable interrupt	2	-
RETIA	i		Return, place immediate in ACC	2	-
CALLA			Call subroutine by ACC	2	-
GOTOA			unconditional branch by ACC	2	-
LCALL	adr		Call subroutine	2	-
LGOTO	adr		unconditional branch	2	-

Table 36 Instruction Set

ACC: Accumulator.

adr: immediate address.

bit: bit address within an 8-bit register R.

C: Carry/Borrow bit

C=1, carry is occurred for addition instruction or borrow is **NOT** occurred for subtraction instruction.

C=0, carry is not occurred for addition instruction or borrow **IS** occurred for subtraction instruction.

d: Destination

If d is “0”, the result is stored in the ACC.

If d is “1”, the result is stored back in register R.

DC: Digital carry flag.

dest: Destination.

F: F-page SFR, F is 0x5 ~ 0xF.

i: 8-bit immediate data.

PC: Program Counter.

PCHBUF: High Byte Buffer of Program Counter.

/PD: Power down flag bit

/PD=1, after power-up or after instruction CLRWDT is executed.

/PD=0, after instruction SLEEP is executed.

Prescaler: Prescaler0 dividing rate.

R: R-page SFR, R is 0x0 ~0x7F.

S: S-page SFR, S is 0x0 ~ 0x15.

T0MD: T0MD register.

TBHP: The high-Byte at target address in ROM.

TBHD: Store the high-Byte data at target address in ROM.

/TO: Time overflow flag bit

/TO=1, after power-up or after instruction CLRWDT or SLEEP is executed.

/TO=0, WDT timeout is occurred.

WDT: Watchdog Timer Counter.

Z: Zero flag

.

ADCAR	Add ACC and R with Carry
Syntax:	ADCAR R, d
Operand:	0 ≤ R ≤ 127 d = 0, 1.
Operation:	R + ACC + C → dest
Status affected:	Z, DC, C
Description:	Add the contents of ACC and register R with Carry. If d is 0, the result is stored in ACC. If d is 1, the result is stored back to R.
Cycle	1
Example:	ADCAR R, d before executing instruction: ACC=0x12, R=0x34, C=1, d=1, after executing instruction: R=0x47, ACC=0x12, C=0.

ADDAR	Add ACC and R
Syntax:	ADDAR R, d
Operand:	0 ≤ R ≤127 d = 0, 1.
Operation:	ACC + R → dest
Status affected:	Z, DC, C
Description:	Add the contents of ACC and R. If d is 0, the result is stored in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	ADDAR R, d before executing instruction: ACC=0x12, R=0x34,C=1, d=1, after executing instruction: R=0x46, ACC=0x12, C=0.

ADCIA	Add ACC and Immediate with Carry
Syntax:	ADCIA i
Operand:	0 ≤ i < 255
Operation:	ACC + i + C → ACC
Status affected:	Z, DC, C
Description:	Add the contents of ACC and the 8-bit immediate data i with Carry. The result is placed in ACC.
Cycle:	1
Example:	ADCIA i before executing instruction: ACC=0x12, i=0x34, C=1, after executing instruction: ACC=0x47, C=0.

ADDIA	Add ACC and Immediate
Syntax:	ADDIA i
Operand:	0 ≤ i < 255
Operation:	ACC + i → ACC
Status affected:	Z, DC, C
Description:	Add the contents of ACC with the 8-bit immediate data i. The result is placed in ACC.
Cycle:	1
Example:	ADDIA i before executing instruction: ACC=0x12, i=0x34, C=1, after executing instruction: ACC=0x46, C=0.

ANDAR	AND ACC and R
Syntax:	ANDAR R, d
Operand:	0 ≤ R ≤ 127 d = 0, 1.
Operation:	ACC & R →dest
Status affected:	Z
Description:	The content of ACC is AND'ed with R. If d is 0, the result is stored in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	ANDAR R, d before executing instruction: ACC=0x5A, R=0xAF, d=1. after executing instruction: R=0x0A, ACC=0x5A, Z=0.

ANDIA	AND Immediate with ACC
Syntax:	ANDIA i
Operand:	0 ≤ i < 255
Operation:	ACC & i →ACC
Status affected:	Z
Description:	The content of ACC register is AND'ed with the 8-bit immediate data i. The result is placed in ACC.
Cycle:	1
Example:	ANDIA i before executing instruction: ACC=0x5A, i=0xAF, after executing instruction: ACC=0x0A, Z=0.

BCR	Clear Bit in R
Syntax:	BCR R, bit
Operand:	0 ≤ R ≤ 127 0 ≤ bit ≤ 7
Operation:	0 →R[bit]
Status affected:	--
Description:	Clear the bitth position in R.
Cycle:	1
Example:	BCR R, B2 before executing instruction: R=0x5A, B2=0x3, after executing instruction: R=0x52.

BSR	Set Bit in R
Syntax:	BSR R, bit
Operand:	0 ≤ R ≤ 127 0 ≤ bit ≤ 7
Operation:	1 →R[bit]
Status affected:	--
Description:	Set the bitth position in R.
Cycle:	1
Example:	BSR R, B2 before executing instruction: R=0x5A, B2=0x2, after executing instruction: R=0x5E.

BTRSC	Test Bit in R and Skip if Clear
Syntax:	BTRSC R, bit
Operand:	0 ≤ R ≤ 127 0 ≤ bit ≤ 7
Operation:	Skip next instruction, if R[bit] = 0.
Status affected:	--
Description:	If R[bit] = 0, the next instruction which is already fetched is discarded and a NOP is executed instead. Therefore it makes this instruction a two-cycle instruction.
Cycle:	1 or 2(skip)
Example:	BTRSC R, B2 Instruction1 Instruction2 before executing instruction: R=0x5A, B2=0x2, after executing instruction: because R[B2]=0, instruction1 will not be executed, the program will start execute instruction from instruction2.

CALLA	Call Subroutine
Syntax:	CALLA
Operand:	--
Operation:	PC + 1 →Top of Stack {TBHP, ACC} →PC
Status affected:	--
Description:	The return address (PC + 1) is pushed onto top of Stack. The contents of TBHP[2:0] is loaded into PC[10:8] and ACC is loaded into PC[7:0].
Cycle:	2
Example:	CALLA before executing instruction: TBHP=0x02, ACC=0x34. PC=A0. Stack pointer=1. after executing instruction: PC=0x234, Stack[1]=A0+1, Stack pointer=2

BTRSS	Test Bit in R and Skip if Set
Syntax:	BTRSS R, bit
Operand:	0 ≤ R ≤ 127 0 ≤ bit ≤ 7
Operation:	Skip next instruction, if R[bit] = 1.
Status affected:	--
Description:	If R[bit] = 1, the next instruction which is already fetched is discarded and a NOP is executed instead. Therefore it makes this instruction a two-cycle instruction.
Cycle:	1 or 2(skip)
Example:	BTRSS R, B2 Instruction2 Instruction3 before executing instruction: R=0x5A, B2=0x3, after executing instruction: because R[B2]=1, instruction2 will not be executed, the program will start execute instruction from instruction3.

CLRA	Clear ACC
Syntax:	CLRA
Operand:	--
Operation:	00h →ACC 1 →Z
Status affected:	Z
Description:	ACC is clear and Z is set to 1.
Cycle:	1
Example:	CLRA before executing instruction: ACC=0x55, Z=0. after executing instruction: ACC=0x00, Z=1.

CLRR	Clear R
Syntax:	CLRR R
Operand:	0 ≤ R ≤ 127
Operation:	00h → R 1 → Z
Status affected:	Z
Description:	The content of R is clear and Z is set to 1.
Cycle:	1
Example:	CLRR R before executing instruction: R=0x55, Z=0. after executing instruction: R=0x00, Z=1.

COMR	Complement R
Syntax:	COMR R, d
Operand:	0 ≤ R ≤ 127 d = 0, 1.
Operation:	~R → dest
Status affected:	Z
Description:	The content of R is complemented. If d is 0, the result is stored in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	COMR, d before executing instruction: R=0xA6, d=1, Z=0. after executing instruction: R=0x59, Z=0.

CLRWDT	Clear Watch-Dog Timer
Syntax:	CLRWDT
Operand:	--
Operation:	00h → WDT, 00h → WDT prescaler 1 → /TO 1 → /PD
Status affected:	/TO, /PD
Description:	Executing CLRWDT will reset WDT, Prescaler0 if it is assigned to WDT. Moreover, status bits /TO and /PD will be set to 1.
Cycle:	1
Example:	CLRWDT before executing instruction: /TO=0 after executing instruction: /TO=1

CMPAR	Compare ACC and R
Syntax:	CMPAR R
Operand:	0 ≤ R ≤ 127
Operation:	R - ACC → (No restore)
Status affected:	Z, C
Description:	Compare ACC and R by subtracting ACC from R with 2's complement representation. The content of ACC and R is not changed.
Cycle:	1
Example:	CMPAR R before executing instruction: R=0x34, ACC=12, Z=0, C=0. after executing instruction: R=0x34, ACC=12, Z=0, C=1.

DAA	Convert ACC Data Format from Hexadecimal to Decimal
Syntax:	DAA
Operand:	--
Operation:	ACC(hex) → ACC(dec)
Status affected:	C
Description:	Convert ACC data format from hexadecimal to decimal after addition operation and restore result to ACC. DAA instruction must be placed immediately after addition operation if decimal format is required. Please note that interrupt should be disabled before addition instruction and enabled after DAA instruction to avoid unexpected result.
Cycle:	1
Example:	DISI ADDAR R,d DAA ENI before executing instruction: ACC=0x28, R=0x25, d=0. after executing instruction: ACC=0x53, C=0.

DECRSZ	Decrease R, Skip if 0
Syntax:	DECRSZ R, d
Operand:	0 ≤ R ≤ 127 d = 0, 1.
Operation:	R - 1 → dest, Skip if result = 0
Status affected:	--
Description:	Decrease R first. If d is 0, the result is stored in ACC. If d is 1, the result is stored back to R. If result is 0, the next instruction which is already fetched is discarded and a NOP is executed instead. Therefore it makes this instruction a two-cycle instruction.
Cycle:	1 or 2(skip)
Example:	DECRSZ R, d instruction2 instruction3 before executing instruction: R=0x1, d=1, Z=0. after executing instruction: R=0x0, Z=1, and instruction will skip instruction2 execution because the operation result is zero.

DECR	Decrease R
Syntax:	DECR R, d
Operand:	0 ≤ R ≤ 127 d = 0, 1.
Operation:	R - 1 → dest
Status affected:	Z
Description:	Decrease R. If d is 0, the result is stored in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	DECR R, d before executing instruction: R=0x01, d=1, Z=0. after executing instruction: R=0x00, Z=1.

DISI	Disable Interrupt Globally
Syntax:	DISI
Operand:	--
Operation:	Disable Interrupt, 0 → GIE
Status affected:	--
Description:	GIE is clear to 0 in order to disable all interrupt requests.
Cycle:	1
Example:	DISI before executing instruction: GIE=1, After executing instruction: GIE=0.

ENI	Enable Interrupt Globally
Syntax:	ENI
Operand:	--
Operation:	Enable Interrupt, 1 → GIE
Status affected:	--
Description:	GIE is set to 1 in order to enable all interrupt requests.
Cycle:	1
Example:	ENI before executing instruction: GIE=0, After executing instruction: GIE=1.

GOTOA	Unconditional Branch
Syntax:	GOTOA
Operand:	--
Operation:	{TBHP, ACC} → PC
Status affected:	--
Description:	GOTOA is an unconditional branch instruction. The content of TBHP[2:0] is loaded into PC[10:8] and ACC is loaded into PC[7:0].
Cycle:	2
Example:	GOTOA before executing instruction: PC=A0. TBHP=0x02, ACC=0x34. after executing instruction: PC=0x234

INCR	Increase R
Syntax:	INCR R, d
Operand:	0 ≤ R ≤ 127 d = 0, 1.
Operation:	R + 1 → dest.
Status affected:	Z
Description:	Increase R. If d is 0, the result is stored in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	INCR R, d before executing instruction: R=0xFF, d=1, Z=0. after executing instruction: R=0x00, Z=1.

INCRSZ	Increase R, Skip if 0
Syntax:	INCRSZ R, d
Operand:	0 ≤ R ≤ 127 d = 0, 1.
Operation:	R + 1 → dest, Skip if result = 0
Status affected:	--
Description:	Increase R first. If d is 0, the result is stored in ACC. If d is 1, the result is stored back to R. If result is 0, the next instruction which is already fetched is discarded and a NOP is executed instead. Therefore it makes this instruction a two-cycle instruction.
Cycle:	1 or 2(skip)
Example:	INCRSZ R, d instruction2, instruction3. before executing instruction: R=0xFF, d=1, Z=0. after executing instruction: R=0x00, Z=1. And the program will skip instruction2 execution because the operation result is zero.

INT	Software Interrupt
Syntax:	INT
Operand:	--
Operation:	PC + 1 → Top of Stack, 001h → PC
Status affected:	--
Description:	Software interrupt. First, return address (PC + 1) is pushed onto the Stack. The address 0x001 is loaded into PC[10:0].
Cycle:	3
Example:	INT before executing instruction: PC=address of INT code after executing instruction: PC=0x01

IORIA	OR Immediate with ACC
Syntax:	IORIA i
Operand:	0 ≤ i < 255
Operation:	ACC | i → ACC
Status affected:	Z
Description:	OR ACC with 8-bit immediate data i. The result is stored in ACC.
Cycle:	1
Example:	IORIA i before executing instruction: i=0x50, ACC=0xAA, Z=0. after executing instruction: ACC=0xFA, Z=0.

IORAR	OR ACC with R
Syntax:	IORAR R, d
Operand:	0 ≤ R ≤ 127 d = 0, 1.
Operation:	ACC | R → dest
Status affected:	Z
Description:	OR ACC with R. If d is 0, the result is stored in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	IORAR R, d before executing instruction: R=0x50, ACC=0xAA, d=1, Z=0. after executing instruction: R=0xFA, ACC=0xAA, Z=0.

IOST	Load F-page SFR from ACC
Syntax:	IOST F
Operand:	5 ≤ F ≤ 15
Operation:	ACC → F-page SFR
Status affected:	--
Description:	F-page SFR F is loaded by content of ACC.
Cycle:	1
Example:	IOST F before executing instruction: F=0x55, ACC=0xAA. after executing instruction: F=0xAA, ACC=0xAA.

IOSTR	Move F-page SFR to ACC
Syntax:	IOSTR F
Operand:	5 ≤ F ≤ 15
Operation:	F-page SFR → ACC
Status affected:	--
Description:	Move F-page SFR F to ACC.
Cycle:	1
Example:	IOSTR F before executing instruction: F=0x55, ACC=0xAA. after executing instruction: F=0x55, ACC=0x55.

LGOTO	Unconditional Branch
Syntax:	LGOTO adr
Operand:	0 ≤ adr ≤ 2047
Operation:	adr → PC[10:0].
Status affected:	--
Description:	LGOTO is an unconditional branch instruction. The 11-bit immediate address adr is loaded into PC[10:0].
Cycle:	2
Example:	LGOTO Level before executing instruction: PC=A0. after executing instruction: PC=address of Level.

LCALL	Call Subroutine
Syntax:	LCALL adr
Operand:	0 ≤ adr ≤ 2047
Operation:	PC + 1 → Top of Stack, adr → PC[10:0]
Status affected:	--
Description:	The return address (PC + 1) is pushed onto top of Stack. The 11-bit immediate address adr is loaded into PC[10:0].
Cycle:	2
Example:	LCALL SUB before executing instruction: PC=A0. Stack level=1 after executing instruction: PC=address of SUB, Stack[1]= A0+1, Stack pointer =2.

MOVAR	Move ACC to R
Syntax:	MOVAR R
Operand:	0 ≤ R ≤ 127
Operation:	ACC → R
Status affected:	--
Description:	Move content of ACC to R.
Cycle:	1
Example:	MOVAR R before executing instruction: R=0x55, ACC=0xAA. after executing instruction: R=0xAA, ACC=0xAA.

MOVIA	Move Immediate to ACC
Syntax:	MOVIA i
Operand:	$0 \leq i < 255$
Operation:	$i \rightarrow ACC$
Status affected:	--
Description:	The content of ACC is loaded with 8-bit immediate data i.
Cycle:	1
Example:	MOVIA i before executing instruction: i=0x55, ACC=0xAA. after executing instruction: ACC=0x55.

NOP	No Operation
Syntax:	NOP
Operand:	--
Operation:	No operation.
Status affected:	--
Description:	No operation.
Cycle:	1
Example:	NOP before executing instruction: PC=A0 after executing instruction: PC=A0+1

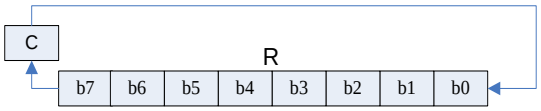
MOVR	Move R to ACC or R
Syntax:	MOVR R, d
Operand:	$0 \leq R \leq 127$ d = 0, 1.
Operation:	$R \rightarrow$ dest
Status affected:	Z
Description:	The content of R is move to destination. If d is 0, destination is ACC. If d is 1, destination is R and it can be used to check whether R is zero according to status flag Z after execution.
Cycle:	1
Example:	MOVR R, d before executing instruction: R=0x0, ACC=0xAA, Z=0, d=0. after executing instruction: R=0x0, ACC=0x00, Z=1.

RETIE	Return from Interrupt and Enable Interrupt Globally
Syntax:	RETIE
Operand:	--
Operation:	Top of Stack $\rightarrow$ PC $1 \rightarrow$ GIE
Status affected:	--
Description:	The PC is loaded from top of Stack as return address and GIE is set to 1.
Cycle:	2
Example:	RETIE before executing instruction: GIE=0, Stack level=2. after executing instruction: GIE=1, PC=Stack[2], Stack pointer=1.

RETIA	Return with Data in ACC
Syntax:	RETIA i
Operand:	0 ≤ i < 255
Operation:	i → ACC, Top of Stack → PC
Status affected:	--
Description:	ACC is loaded with 8-bit immediate data i and PC is loaded from top of Stack as return address.
Cycle:	2
Example:	RETIA i before executing instruction: Stack pointer =2. i=0x55, ACC=0xAA. after executing instruction: PC=Stack[2], Stack pointer =1. ACC=0x55.

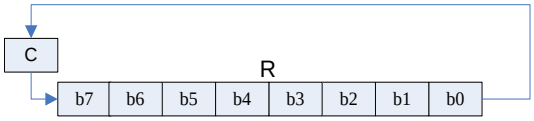
RET	Return from Subroutine
Syntax:	RET
Operand:	--
Operation:	Top of Stack → PC
Status affected:	--
Description:	PC is loaded from top of Stack as return address.
Cycle:	2
Example:	RET before executing instruction: Stack level=2. after executing instruction: PC=Stack[2], Stack level=1.

RLR	Rotate Left R Through Carry
Syntax:	RLR R, d
Operand:	0 ≤ R ≤ 127 d = 0, 1.
Operation:	C → dest[0], R[7] → C, R[6:0] → dest[7:1]



Status affected:	C
Description:	The content of R is rotated one bit to the left through flag Carry. If d is 0, the result is placed in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	RLR R, d before executing instruction: R=0xA5, d=1, C=0. after executing instruction: R=0x4A, C=1.

RRR	Rotate Right R Through Carry
Syntax:	RRR R, d
Operand:	0 ≤ R ≤ 127 d = 0, 1.
Operation:	C → dest[7], R[7:1] → dest[6:0], R[0] → C



Status affected:	C
Description:	The content of R is rotated one bit to the right through flag Carry. If d is 0, the result is placed in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	RRR R, d before executing instruction: R=0xA5, d=1, C=0. after executing instruction: R=0x52, C=1.

SBCAR	Subtract ACC and Carry from R
Syntax:	SBCAR R, d
Operand:	0 ≤ R ≤ 127 d = 0, 1.
Operation:	R + (~ACC) + C → dest
Status affected:	Z, DC, C
Description:	Subtract ACC and Carry from R with 2's complement representation. If d is 0, the result is placed in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	SBCAR R, d (a) before executing instruction: R=0x05, ACC=0x06, d=1, C=0, after executing instruction: R=0xFE, C=0. (-2) (b) before executing instruction: R=0x05, ACC=0x06, d=1, C=1, after executing instruction: R=0xFF, C=0. (-1) (c) before executing instruction: R=0x06, ACC=0x05, d=1, C=0, after executing instruction: R=0x00, C=1. (-0), Z=1. (d) before executing instruction: R=0x06, ACC=0x05, d=1, C=1, after executing instruction: R=0x1, C=1. (+1)

SBCIA	Subtract ACC and Carry from Immediate
Syntax:	SBCIA i
Operand:	0 ≤ i < 255
Operation:	i + (~ACC) + C → dest
Status affected:	Z, DC, C
Description:	Subtract ACC and Carry from 8-bit immediate data i with 2's complement representation. The result is placed in ACC.
Cycle:	1
Example:	SBCIA i (a) before executing instruction: i=0x05, ACC=0x06, C=0, after executing instruction: ACC=0xFE, C=0. (-2) (b) before executing instruction: i=0x05, ACC=0x06, C=1, after executing instruction: ACC=0xFF, C=0. (-1) (c) before executing instruction: i=0x06, ACC=0x05, C=0, after executing instruction: ACC=0x00, C=1. (-0), Z=1. (d) before executing instruction: i=0x06, ACC=0x05, C=1, after executing instruction: ACC=0x1, C=1. (+1)

SFUN	Load S-page SFR from ACC
Syntax:	SFUN S
Operand:	0 ≤ S ≤ 21
Operation:	ACC → S-page SFR
Status affected:	--
Description:	S-page SFR S is loaded by content of ACC.
Cycle:	1
Example:	SFUN S before executing instruction: S=0x55, ACC=0xAA. after executing instruction: S=0xAA, ACC=0xAA.

SFUNR	Move S-page SFR to ACC
Syntax:	SFUNR S
Operand:	0 ≤ S ≤ 21
Operation:	S-page SFR → ACC
Status affected:	--
Description:	Move S-page SFR S to ACC.
Cycle:	1
Example:	SFUNR S before executing instruction: S=0x55, ACC=0xAA. after executing instruction: S=0x55, ACC=0x55.

SLEEP	Enter Halt Mode
Syntax:	SLEEP
Operand:	--
Operation:	00h → WDT, 00h → WDT prescaler 1 → /TO 0 → /PD
Status affected:	/TO, /PD
Description:	WDT and Prescaler0 are clear to 0. /TO is set to 1 and /PD is clear to 0. IC enter Halt mode.
Cycle:	1
Example:	SLEEP before executing instruction: /PD=1, /TO=0. after executing instruction: /PD=0, /TO=1.

SUBAR	Subtract ACC from R
Syntax:	SUBAR R, d
Operand:	0 ≤ R ≤ 127 d = 0, 1.
Operation:	R – ACC → dest
Status affected:	Z, DC, C
Description:	Subtract ACC from R with 2's complement representation. If d is 0, the result is placed in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	SUBAR R, d (a) before executing instruction: R=0x05, ACC=0x06, d=1, after executing instruction: R=0xFF, C=0. (-1) (b) before executing instruction: R=0x06, ACC=0x05, d=1, after executing instruction: R=0x01, C=1. (+1)

SUBIA	Subtract ACC from Immediate
Syntax:	SUBIA i
Operand:	0 ≤ i < 255
Operation:	i – ACC → ACC
Status affected:	Z, DC, C
Description:	Subtract ACC from 8-bit immediate data i with 2's complement representation. The result is placed in ACC.
Cycle:	1
Example:	SUBIA i (a) before executing instruction: i=0x05, ACC=0x06. after executing instruction: ACC=0xFF, C=0. (-1) (b) before executing instruction: i=0x06, ACC=0x05, d=1, after executing instruction: ACC=0x01, C=1. (+1)

SWAPR	Swap High/Low Nibble in R
Syntax:	SWAPR R, d
Operand:	0 ≤ R ≤ 127 d = 0, 1.
Operation:	R[3:0] → dest[7:4]. R[7:4] → dest[3:0]
Status affected:	--
Description:	The high nibble and low nibble of R is exchanged. If d is 0, the result is placed in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	SWAPR R, d before executing instruction: R=0xA5, d=1. after executing instruction: R=0x5A.

T0MD	Load ACC to T0MD
Syntax:	T0MD
Operand:	--
Operation:	ACC → T0MD
Status affected:	--
Description:	The content of T0MD is loaded by ACC.
Cycle:	1
Example:	T0MD before executing instruction: T0MD=0x55, ACC=0xAA. after executing instruction: T0MD=0xAA.

TABLEA	Read ROM data
Syntax:	TABLEA
Operand:	--
Operation:	ROM data{ TBHP, ACC } [7:0] → ACC ROM data{TBHP, ACC} [13:8] → TBHD.
Status affected:	--
Description:	The 8 least significant bits of ROM pointed by {TBHP[2:0], ACC} is placed to ACC. The 6 most significant bits of ROM pointed by {TBHP[2:0], ACC} is placed to TBHD[5:0].
Cycle:	2
Example:	TABLEA before executing instruction: TBHP=0x02, CC=0x34. TBHD=0x01. ROM data[0x234]= 0x35AA after executing instruction: TBHD=0x35, ACC=0xAA.

T0MDR	Move T0MD to ACC
Syntax:	T0MDR
Operand:	--
Operation:	T0MD → ACC
Status affected:	--
Description:	Move the content of T0MD to ACC.
Cycle:	1
Example:	T0MDR before executing instruction T0MD=0x55, ACC=0xAA. after executing instruction ACC=0x55.

XORAR	Exclusive-OR ACC with R
Syntax:	XORAR R, d
Operand:	$0 \leq R \leq 127$ d = 0, 1.
Operation:	$ACC \oplus R \rightarrow dest$
Status affected:	Z
Description:	Exclusive-OR ACC with R. If d is 0, the result is placed in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	XORAR R, d before executing instruction: R=0xA5, ACC=0xF0, d=1. after executing instruction: R=0x55.

XORIA	Exclusive-OR Immediate with ACC
Syntax:	XORIA i
Operand:	$0 \leq i < 255$
Operation:	$ACC \oplus i \rightarrow ACC$
Status affected:	Z
Description:	Exclusive-OR ACC with 8-bit immediate data i. The result is stored in ACC.
Cycle:	1
Example:	XORIA i before executing instruction: i=0xA5, ACC=0xF0. after executing instruction: ACC=0x55.

5. Configuration Words

Item	Name	Options
1	High Oscillator Frequency	1. I_HRC
2	Low Oscillator Frequency	1. I_LRC
3	High IRC Frequency	1. 1MHz 2. 2MHz 3. 4MHz 4. 8MHz 5. 16MHz 6. 20MHz
4	Instruction Clock	1. 2 oscillator period 2. 4 oscillator period
5	WDT	1. Watchdog Enable (Software control) 2. Watchdog Disable (Always disable)
6	WDT Event	1. Watchdog Reset 2. Watchdog Interrupt
7	Timer0 Source	1. EX_CKI0 2. Low Oscillator (I_LRC/E_LXT)
8	PA.5	1. PA.5 is I/O 2. PA.5 is reset
9	PB.2	1. PB.2 is I/O 2. PB.2 is instruction clock output
10	Startup Time	1. 140us 2. 4.5ms 3. 18ms 4. 72ms 5. 288ms
11	WDT Time Base	1. 3.5ms 2. 15ms 3. 60ms 4. 250ms
12	LVR Setting	1. Register Control 2. LVR Always On
13	LVR Voltage	1. 1.6V 2. 1.8V 3. 2.0V 4. 2.2V 5. 2.4V 6. 2.7V 7. 3.0V 8. 3.3V 9. 3.6V 10. 4.2V
14	VDD Voltage	1. 3.0V 2. 4.5V 3. 5.0V
15	Sink current (exclude PA5)	1. Large 2. Normal
16	Comparator Input pin select	1. Enable 2. Disable
17	Read Output Data	1. I/O Port 2. Register
18	EX_CKI0 to Inst. Clock	1. Sync 2. Async
19	Startup Clock	1. Fast (I_HRC) 2. Slow (I_LRC)
20	Input Schmitt Trigger	1 Enable 2. Disable (0.5VDD)
21	Input High Voltage (VIH)	1. 0.7VDD 2. 0.5VDD
22	Input Low Voltage (VIL)	1. 0.3VDD 2. 0.2VDD

Table 37 Configuration Words

6. Electrical Characteristics

6.1 Absolute Maximum Rating

Symbol	Parameter	Rated Value	Unit
V_{DD} - V_{SS}	Supply voltage	-0.5 ~ +6.0	V
V_{IN}	Input voltage	V_{SS}-0.3V ~ V_{DD}+0.3	V
T_{OP}	Operating Temperature	-40 ~ +85	°C
T_{ST}	Storage Temperature	-40 ~ +125	°C

6.2 DC Characteristics

(All refer F_{INST}=F_{HOSC}/4, F_{HOSC}=16MHz@I_HRC, WDT enabled, ambient temperature T_A=25°C unless otherwise specified.)

Symbol	Parameter	V_{DD}	Min.	Typ.	Max.	Unit	Condition
V_{DD}	Operating voltage	--	3.3	--	5.5	V	F_{INST}=20MHz @ I_HRC/2
			2.2				F_{INST}=20MHz @ I_HRC/4
			3.0				F_{INST}=16MHz @ I_HRC/2
			2.0				F_{INST}=16MHz @ I_HRC/4
			2.0				F_{INST}=8MHz @ I_HRC/2
			1.8				F_{INST}=4MHz @ I_HRC/2
			1.6				F_{INST}=32KHz @ I_LRC/2
V_{IH}	Input high voltage	5V	4.0	--	--	V	RSTb (0.8 V_{DD})
		3V	2.4	--	--		
		5V	3.5	--	--	V	All other I/O pins, EX_CKI1, INT1 CMOS option (0.7 V_{DD})
		3V	2.1	--	--		
		5V	2.5	--	--	V	All other I/O pins, EX_CKI1 TTL option (0.5 V_{DD})
		3V	1.5	--	--		
V_{IL}	Input low voltage	5V	--	--	1.0	V	RSTb (0.2 V_{DD})
		3V	--	--	0.6		
		5V	--	--	1.5	V	All other I/O pins, EX_CKI1, INT1 0.3 V_{DD} option
		3V	--	--	0.9		
		5V	--	--	1.0	V	All other I/O pins, EX_CKI1 0.2 V_{DD} option
		3V	--	--	0.6		
I_{OH}	Output high current	5V	--	18	--	mA	V_{OH}=4.0V
		3V	--	10	--		V_{OH}=2.0V
I_{OL}	Output low current (Large current)	5V	--	43	--	mA	V_{OL}=1.0V
		3V	--	28	--		
I_{OL}	Output low current (Normal current)	5V	--	26	--	mA	V_{OL}=1.0V
		3V	--	16	--		
I_{IR}	IR sink current	5V	--	43	--	mA	V_{OL}=1.0V
		3V	--	28	--		
I_{OP}	Operating current	**Normal Mode**					
		5V	--	1.7	--	mA	F_{HOSC}=20MHz @ I_HRC/2
		3V	--	0.7	--		

Symbol	Parameter	V_{DD}	Min.	Typ.	Max.	Unit	Condition
		5V	--	1.4	--	mA	F_{HOSC}=20MHz @ I_HRC/4
		3V	--	0.5	--		
		5V	--	1.6	--	mA	F_{HOSC}=16MHz @ I_HRC/2
		3V	--	0.6	--		
		5V	--	1.3	--	mA	F_{HOSC}=16MHz @ I_HRC/4
		3V	--	0.5	--		
		5V	--	1.3	--	mA	F_{HOSC}=8MHz @ I_HRC/2
		3V	--	0.5	--		
		5V	--	1.1	--	mA	F_{HOSC}=8MHz @ I_HRC/4
		3V	--	0.4	--		
		5V	--	1.1	--	mA	F_{HOSC}=4MHz @ I_HRC/2
		3V	--	0.4	--		
		5V	--	1.0	--	mA	F_{HOSC}=4MHz @ I_HRC/4
		3V	--	0.4	--		
		5V	--	1.0	--	mA	F_{HOSC}=1MHz @ I_HRC/2
		3V	--	0.3	--		
		5V	--	1.0	--	mA	F_{HOSC}=1MHz @ I_HRC/4
		3V	--	0.3	--		
		Slow Mode					
		5V	--	11	--	uA	F_{HOSC} disabled, F_{LOSC}=32KHz @ I_LRC/2
		3V	--	6.1	--		
		5V	--	7.3	--	uA	F_{HOSC} disabled, F_{LOSC}=32KHz @ I_LRC/4
		3V	--	4.3	--		
I_{STB}	Standby current	5V	--	3.8	--	uA	Standby mode, F_{HOSC} disabled, F_{LOSC}=32KHz @ I_LRC/4
		3V	--	2.6	--		
I_{HALT}	Halt current	5V	--	--	0.5	uA	Halt mode, WDT disabled.
		3V	--	--	0.2		
		5V	--	--	5.0	uA	Halt mode, WDT enabled.
		3V	--	--	3.0		
R_{PH}	Pull-High resistor	5V	--	50	--	KΩ	Pull-High resistor (not include PA5)
		3V	--	100	--		
		5V	--	85	--	KΩ	Pull-High resistor (PA5)
		3V	--	85	--		
R_{PL}	Pull-Low resistor	5V	--	50	--	KΩ	Pull-Low resistor
		3V	--	100	--		

6.3 OSC Characteristics

(Measurement conditions V_{DD} Voltage, T_A Temperature are equal to programming conditions.)

Parameter	Min.	Typ.	Max.	Unit	Condition
I_HRC deviation by socket			±1	%	Socket installed directly on writer.
I_HRC deviation by handler			±3	%	Handler condition with correct setup.
I_LRC deviation by handler			±5	%	

6.4 Comparator / LVD Characteristics

(V_{DD}=5V, V_{SS}=0V, T_A=25°C unless otherwise specified.)

Symbol	Parameter	Min.	Typ.	Max.	Unit	Condition
V_{IVR}	Comparator input voltage range	0	--	5	V	F_{HOSC}=1MHz
T_{ENO}	Comparator enable to output valid	--	20	--	us	F_{HOSC}=1MHz
I_{CO}	Operating current of comparator	--	135	--	uA	F_{HOSC}=1MHz, P2V mode
I_{LVD}	Operating current of LVD	--	150	--	uA	F_{HOSC}=1MHz, LVD=4.3V
E_{LVD}	LVD voltage error	--	--	3	%	F_{HOSC}=1MHz, LVD=4.3V

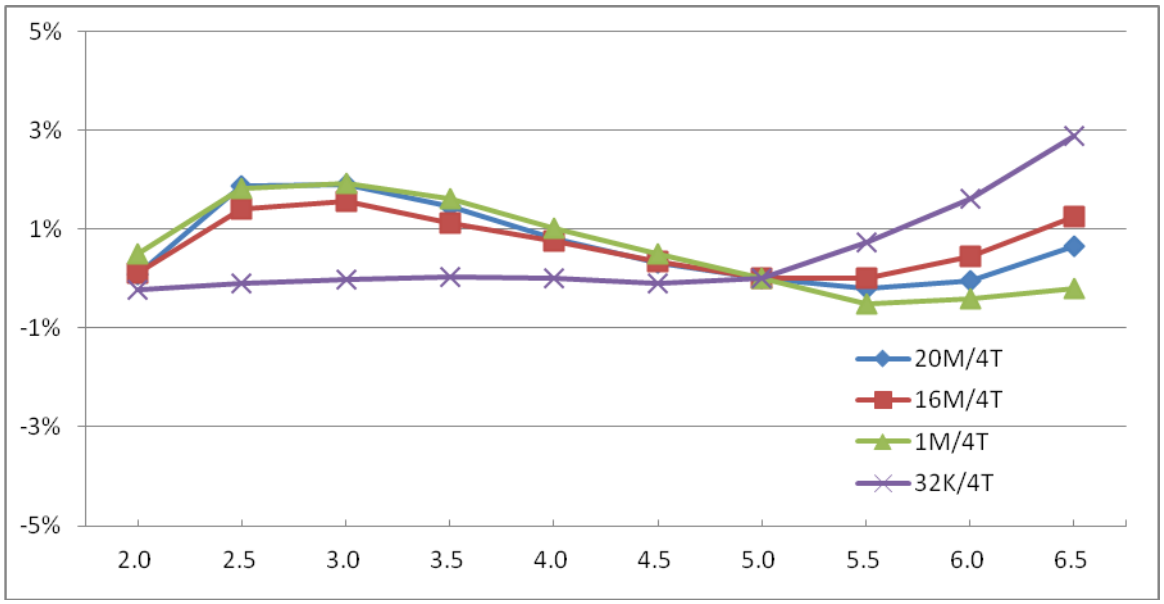
6.5 ADC Characteristics

(V_{DD}=5V, V_{SS}=0V, T_A=25°C unless otherwise specified.)

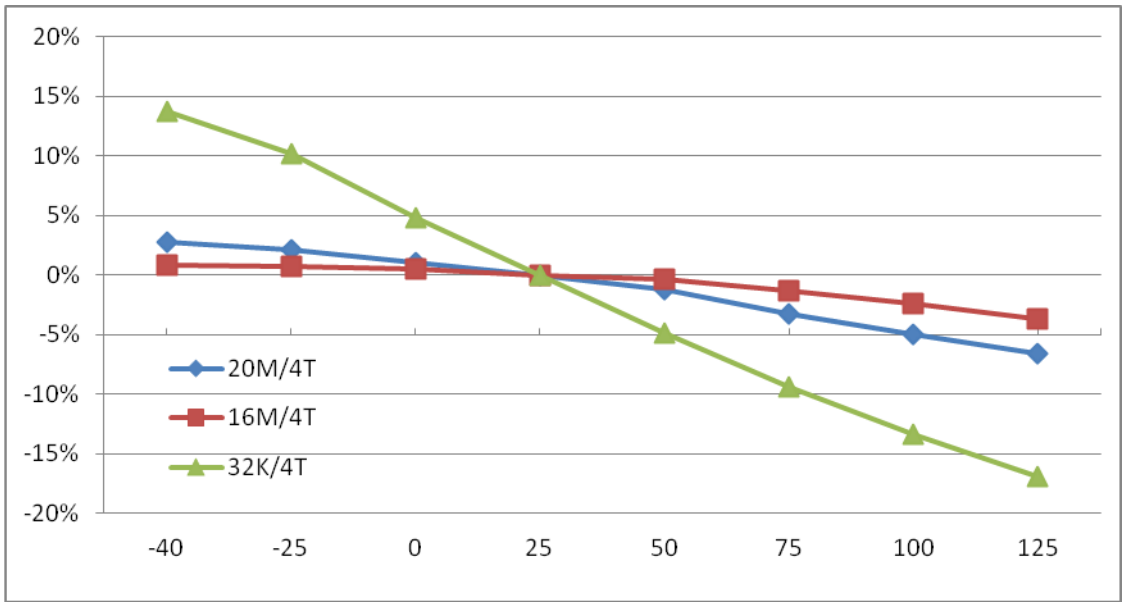
Symbol	Parameter	Min.	Typ.	Max.	Unit	Condition
V_{REFH}	VREFH input voltage	2V	--	V_{DD}	V	Ext. reference voltage
V_{REF4}	Int. 4V reference voltage, V_{DD}=5V	3.96	4	4.04	V	
V_{REF3}	Int. 3V reference voltage, V_{DD}=5V	2.97	3	3.03	V	
V_{REF2}	Int. 2V reference voltage, V_{DD}=5V	1.98	2	2.02	V	
V_{REF}	Int. V_{DD} reference voltage, V_{DD}=5V	--	V_{DD}	--	V	
	Internal reference supply voltage	V_{REF}+0.5	--	--	V	Minimum supply voltage
	ADC analog input voltage	0	--	V_{REFH}	V	
$I_{OP(ADC)}$	ADC current consumption	--	0.5	--	mA	
ADCLK	ADC Clock Frequency	32K	--	1M	Hz	
ADCYCLE	ADC Conversion Cycle Time	16	--		1/ADCLK	SHCLK=2 ADC clock
ADC_{sample}	ADC Sampling Rate	--	--	125	K/sec	V_{DD}=5V
DNL	Differential Nonlinearity	±1	--	--	LSB	V_{DD}=5.0V, AVREFH=5V, FADSMP=62.5K
INL	Integral Nonlinearity	±2	--	--	LSB	
NMC	No Missing Code	10	11	12	Bits	

6.6 Characteristic Graph

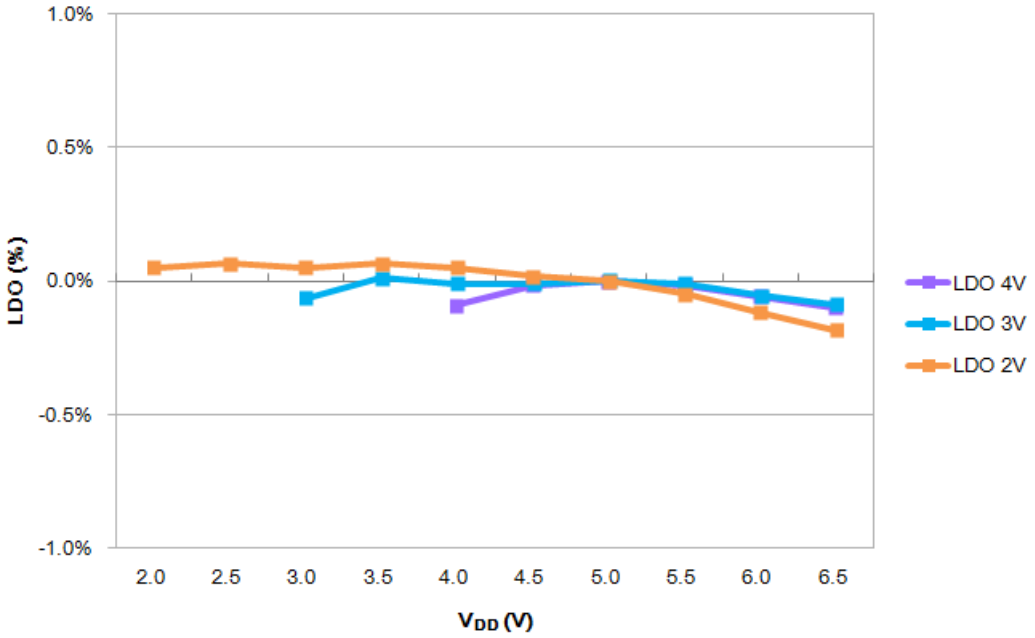
6.6.1 Frequency vs. V_{DD} of I_HRC and I_LRC



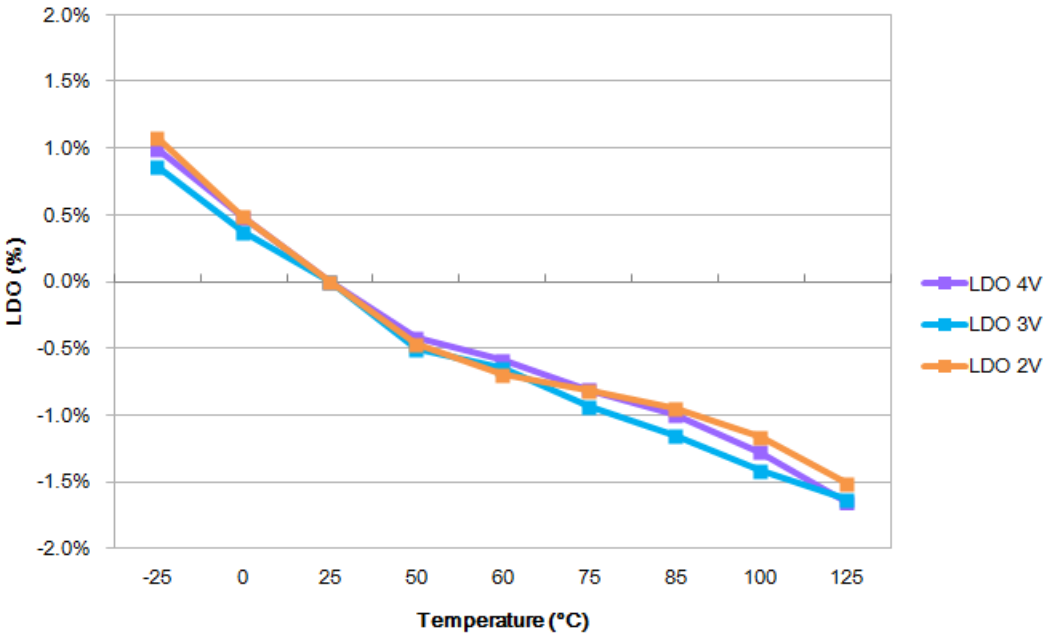
6.6.2 Frequency vs. Temperature of I_HRC and I_LRC



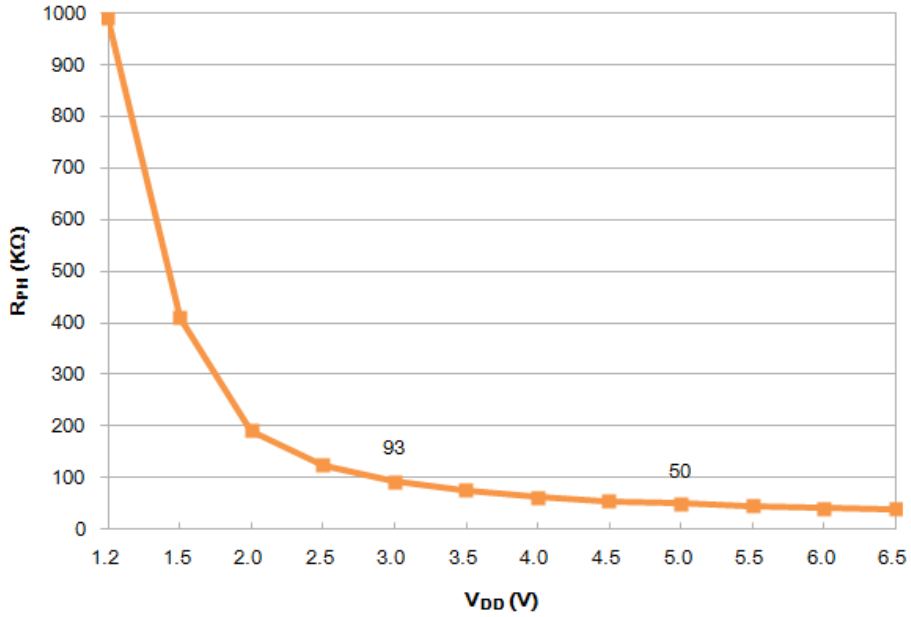
6.6.5 Low Dropout Regulator vs. V_{DD}



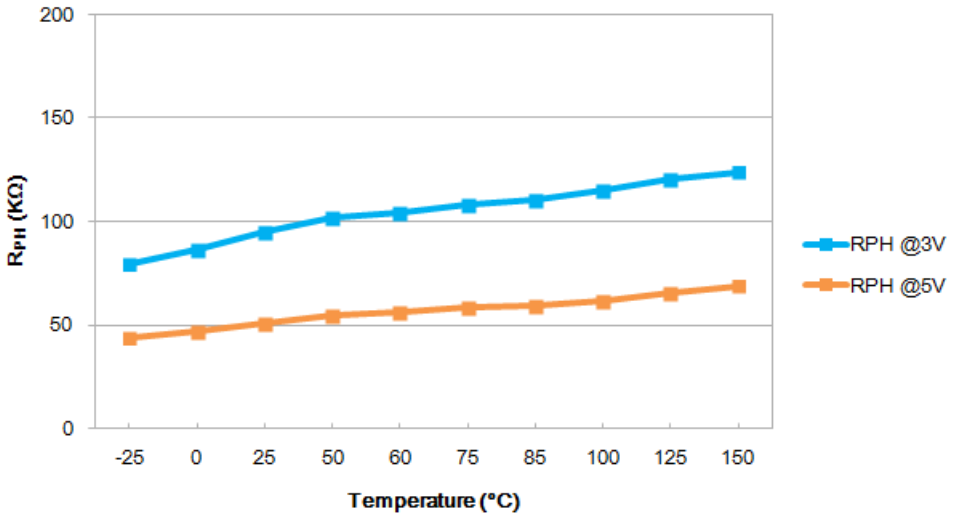
6.6.6 Low Dropout Regulator vs. Temperature



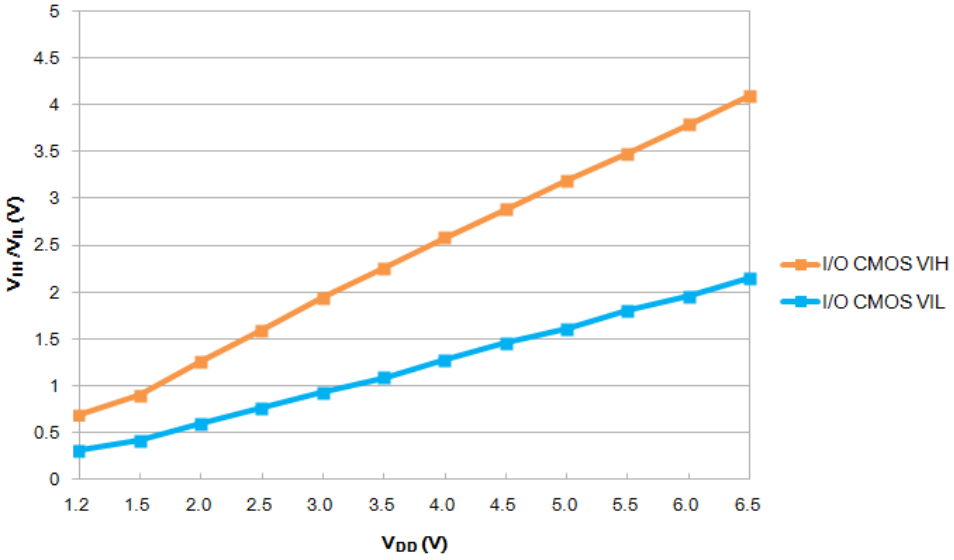
6.6.7 Pull High Resistor vs. VDD

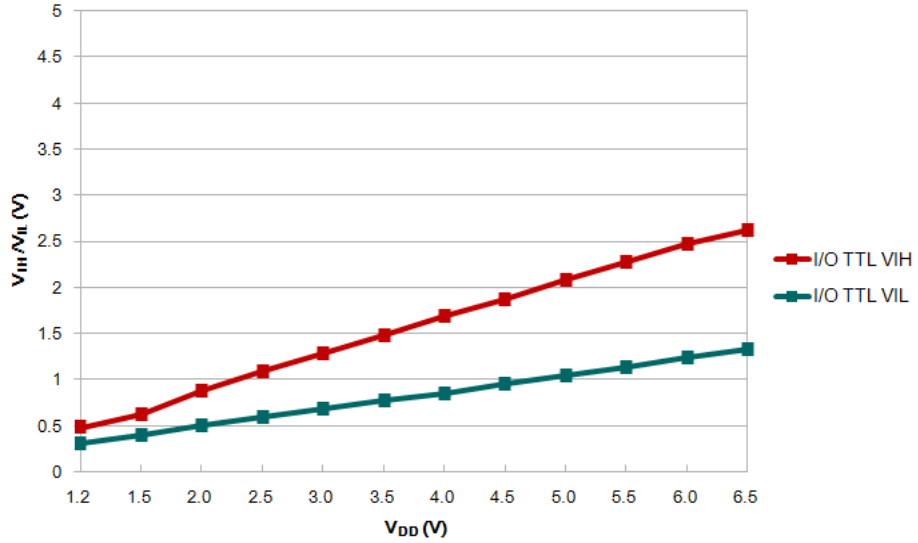


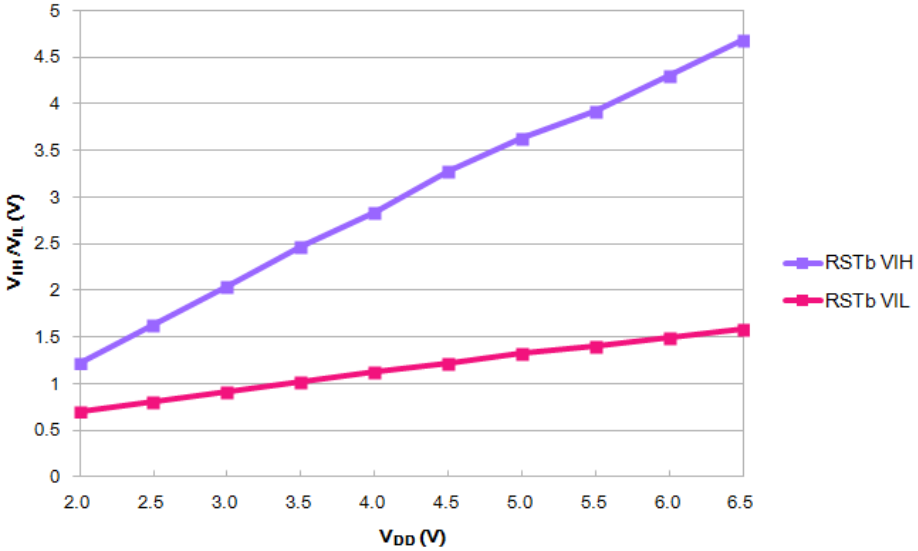
6.6.8 Pull High Resistor vs. Temperature



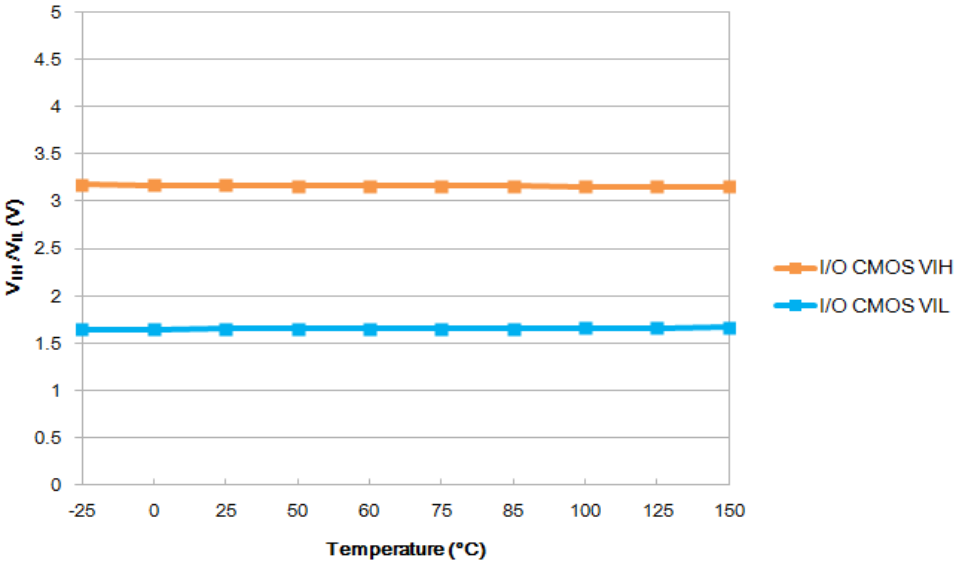
6.6.9 VIH/VIL vs. VDD

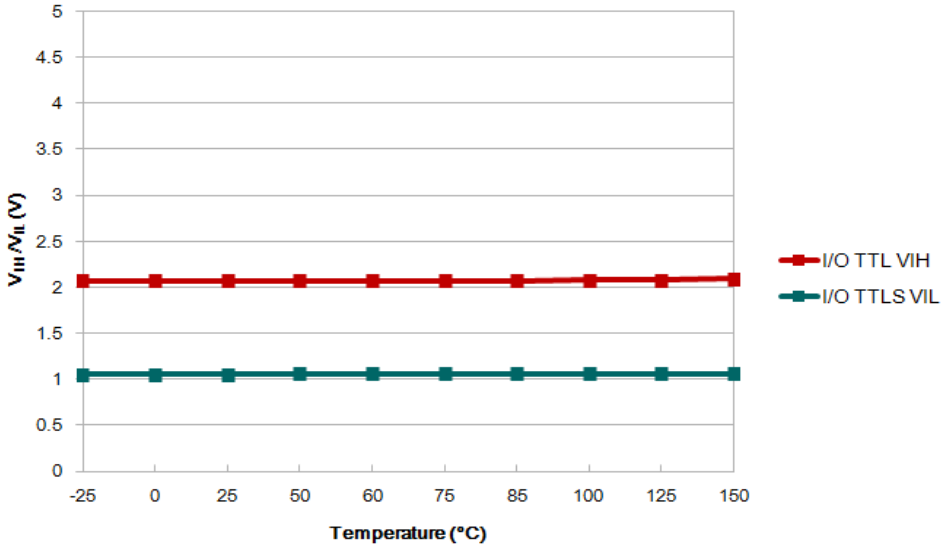


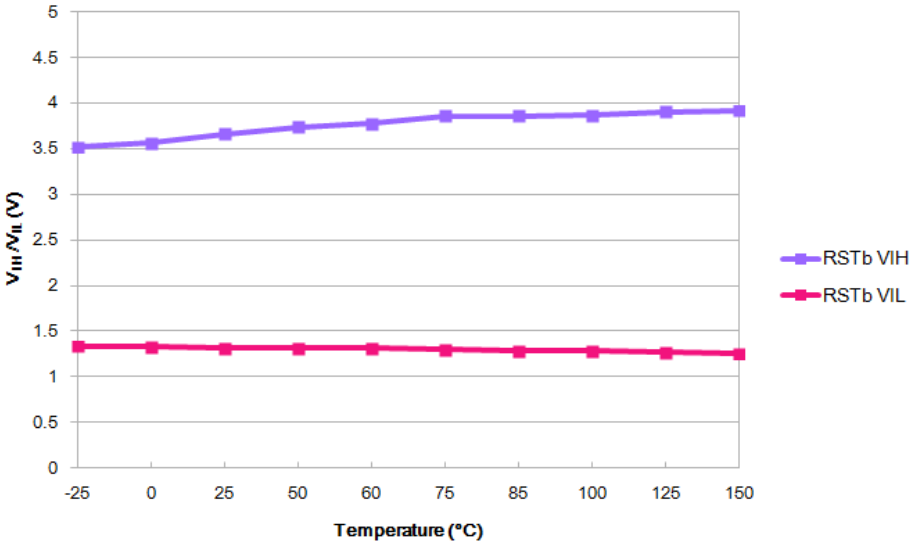




6.6.10 VIH/VIL vs. Temperature





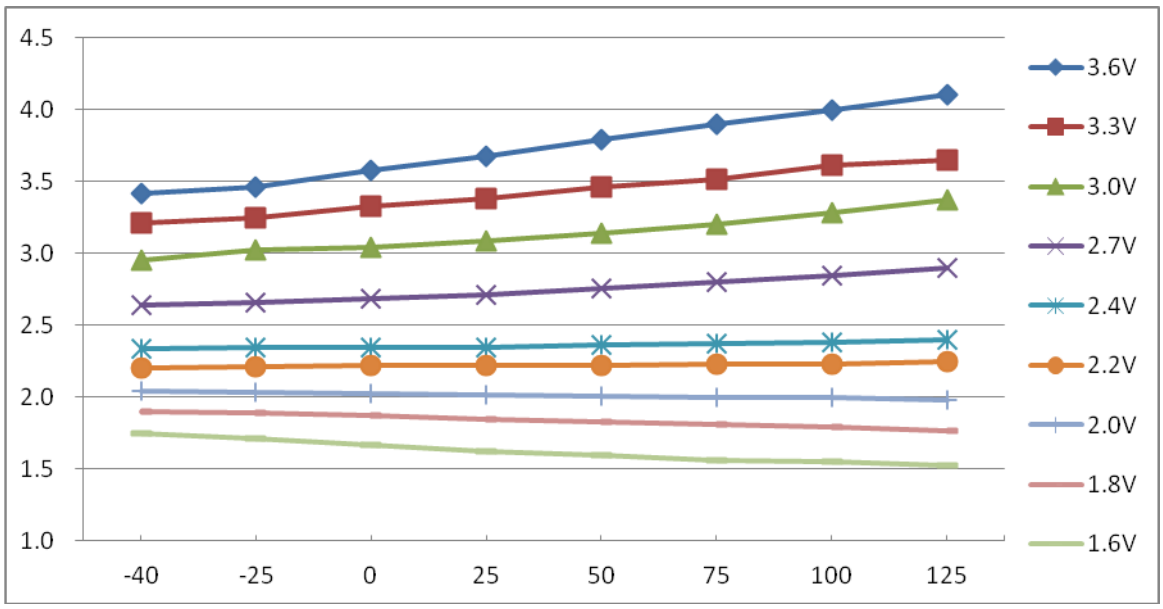


6.7 Recommended Operating Voltage

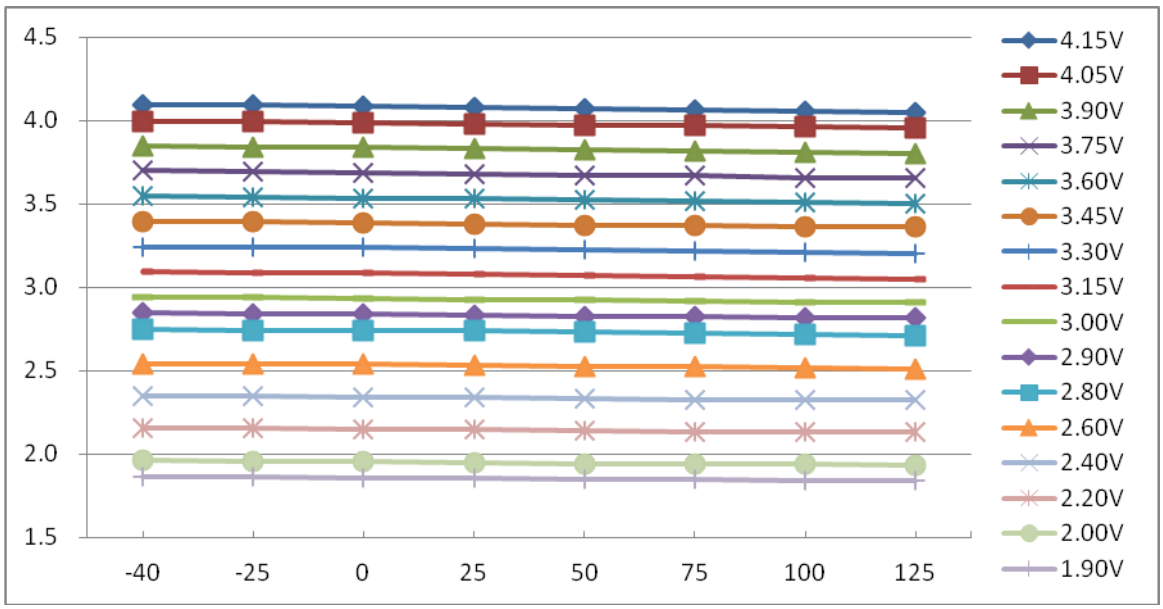
Recommended Operating Voltage (Temperature range: -40°C ~ +85°C)

Frequency	Min. Voltage	Max. Voltage	LVR: default (25°C)	LVR: Recommended (-40°C ~ +85°C)
20M/2T	3.3V	5.5V	3.6V	3.6V
16M/2T	3.0V	5.5V	3.3V	3.6V
20M/4T	2.2V	5.5V	2.4V	2.7V
16M/4T	2.0V	5.5V	2.2V	2.4V
8M(2T or 4T)	2.0V	5.5V	2.2V	2.4V
≦6M(2T or 4T)	1.8V	5.5V	2.0V	2.2V

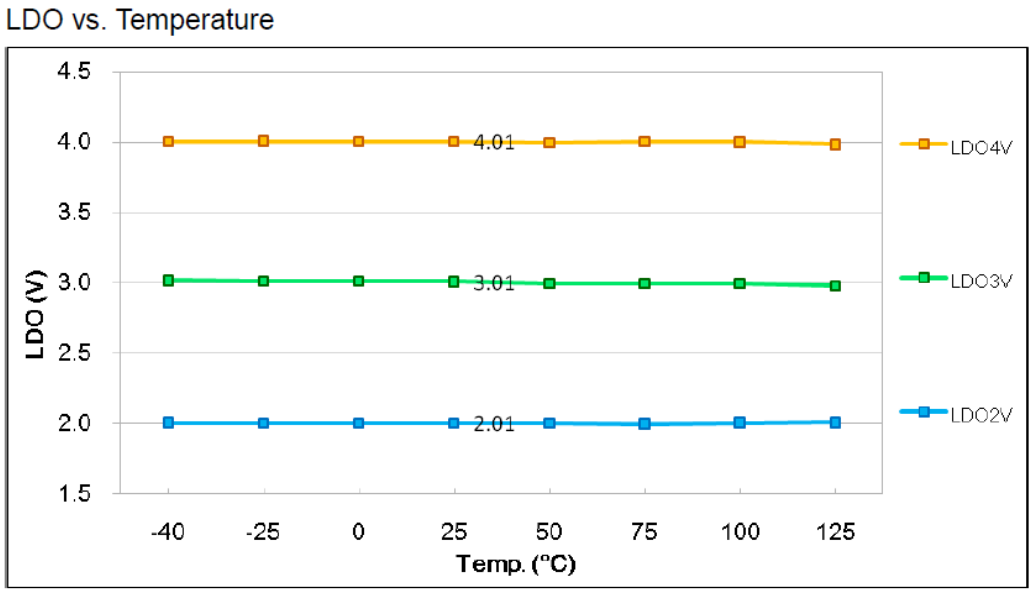
6.8 LVR vs. Temperature



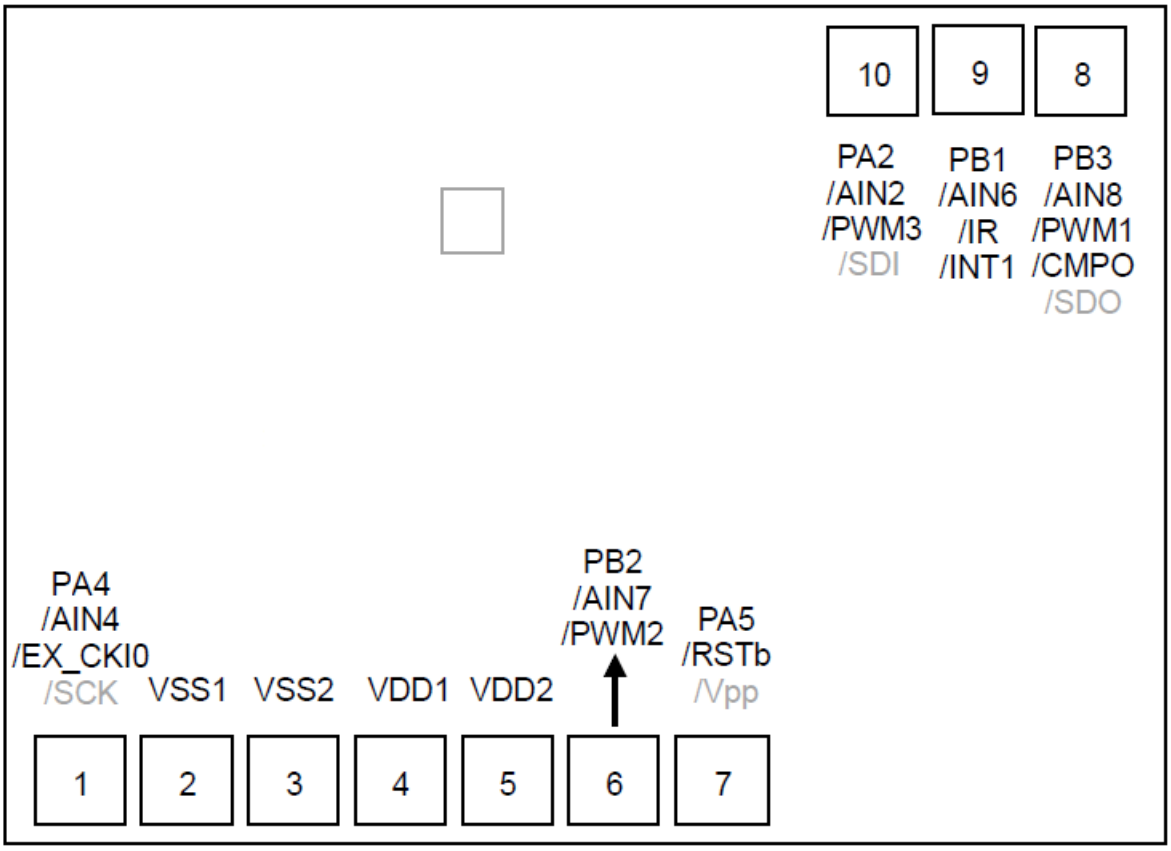
6.9 LVD vs. Temperature



6.10 LDO vs. Temperature

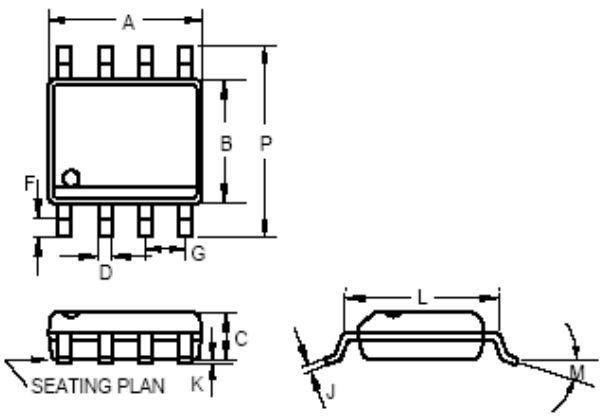


7. Die Pad Diagram



8. Package Dimension

8.1 8-Pin Plastic SOP (150 mil)



	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

Note: For 8-pin SOP, 100 units per tube.

9. Ordering Information

P/N	*Package Type*	*Pin Count*	*Package Width*	*Shipping*
NY8B060DS8	SOP	8	150 mil	Tape & Reel: 2.5K pcs per Reel Tube: 100 pcs per Tube