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

DATA SHEET

NY8A056A

16 I/O 8-bit EPROM-Based MCU

Version 1.6

May 31, 2020

Revision History

Version	Date	Description	Modified Page
1.0	2015/11/26	The first version.	-
1.1	2016/06/16	Add SOP16 package type of NY8A056AS16.	13, 95, 99
1.2	2017/11/14	1. Modify Table 25 Summary of /TO & /PD Value and its Associated Event. 2. Modify “E_LXT Backup Control” option in Configuration Words.	69 87
1.3	2018/07/17	1. Modify Figure 2 Program Memory Address Mapping. 2. Modify Initial Value of PCON1. 3. Modify LVD description. 4. Update RLR instruction description.	15 28 57 83
1.4	2018/10/11	Modify LVD voltage error value to ±10%.	90
1.5	2020/03/23	Add OSC Characteristics.	90
1.6	2020/05/31	Remove and inhibit LVD function	21, 24

Table of Contents

1. 概述

NY8A056A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、及高抗電磁干擾等顯著優勢。NY8A056A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。

在I/O的資源方面，NY8A056A有 16 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，NY8A056A內建了可選擇頻率的大電流輸出紅外載波發射口，電流可在 3V供電時達到 340mA。

NY8A056A有三組計時器，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NY8A056A提供 2 組 8 位元解析度的PWM輸出或者蜂鳴器輸出可用來驅動馬達、LED、或蜂鳴器等等。

NY8A056A採用雙時鐘機制，高速振盪或者低速振盪都可以分別選擇內部RC振盪或外部Crystal輸入。在雙時鐘機制下，NY8A056A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。並且微控制器在使用內部RC高速振盪時，低速振盪可以同時使用外部精準的Crystal計時。可以維持高速處理同時又能精準計算真實時間。

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NY8A056A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：(指令週期為 4 個CPU clock，亦即 4T模式)
  - 2.0V ~ 5.5V @系統頻率≦8MHz。
  - 2.2V ~ 5.5V @系統頻率 > 8MHz。
- 寬廣的工作温度：-40°C ~ 85°C。
- 超過±8KV的ESD。
- 雜訊過濾功能(Noise Filter)打開時可容忍超過±4KV的EFT。(操作電壓@5V)
- 1Kx14 bits EPROM。
- 96 bytes SRAM。
- 16 根可分別單獨控制輸入輸出方向的I/O腳(GPIO)、PA[7:0]、PB[7:0]。
- PA[5, 3:0]及PB[3:0]可選擇輸入時使用內建下拉電阻。
- PA[7:6, 4:0]及PB[7:0]可選擇輸入時使用上拉電阻，上拉電阻值可選擇 100kΩ 或 1MΩ。
- PB[7:0]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腳輸出可選擇定灌電流(Constant Sink Current)或一般灌電流(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。
- 二組 8 位元下數計時器(Timer1, 2)可選重複載入或連續下數計時，二組 8 位元下數計時器可合併成一組 16 位元計時器。
- 二個脈衝寬度調變(PWM1, 2)。
- 二個蜂鳴器輸出(BZ1, 2)。
- 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。
- 大電流輸出紅外線載波發射口，可選一般或 340mA灌電流。
- 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。
- 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: E_HXT (超過 6MHz外部高速石英振盪)
    E_XT (455K~6MHz外部石英振盪)
    I_HRC (1~20MHz內部高速RC振盪)
  - 低速振盪: E_LXT (32KHz外部低速石英振盪)
    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)。
- 八種硬體中斷：
  - Timer0 溢位中斷。
  - Timer1 借位中斷。
  - Timer2 借位中斷。
  - WDT 中斷。
  - PB 輸入狀態改變中斷。
  - 外部中斷輸入。
  - 低電壓偵測中斷。
  - 比較器輸出轉態中斷。
- NY8A056A在待機模式(Standby mode)下的八種喚醒中斷：
  - Timer0 溢位中斷。
  - Timer1 借位中斷。
  - Timer2 借位中斷。
  - WDT 中斷。

- PB 輸入狀態改變中斷。
- 外部中斷輸入。
- 低電壓偵測中斷。
- 比較器輸出轉態中斷。

- NY8A056A在睡眠模式(Halt mode)下的三種喚醒中斷：
  - WDT 中斷。
  - PB 輸入狀態改變中斷。
  - 外部中斷輸入。

1. General Description

NY8A056A is an EPROM based 8-bit MCU tailored for I/O based applications like home appliances or meter equipment. NY8A056A adopts advanced CMOS technology to provide customers remarkable solution with low cost, high performance and high noise immunity benefits. RISC architecture is applied to NY8A056A 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, NY8A056A is very suitable for those applications that are sophisticated but compact program size is required.

As NY8A056A address I/O type applications, it can provide 16 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, NY8A056A has built-in large infrared (IR) carrier generator (340mA@3V) with selectable IR carrier frequency and polarity for applications which demand remote control feature.

NY8A056A also provides 3 sets of timers which can be used as regular timer based on system oscillation or event counter with external trigger clock. Moreover, NY8A056A provides 2 sets of 8-bit resolution Pulse Width Modulation (PWM) output and buzzer output in order to drive motor/LED and buzzer.

NY8A056A employs dual-clock oscillation mechanism, either high oscillation or low oscillation can be derived from internal resistor/capacitor oscillator or external crystal oscillator. Moreover, based on dual-clock mechanism, NY8A056A 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. Moreover, it is possible to use internal high-frequency oscillator as CPU operating clock source and external 32KHz crystal oscillator as timer clock input, so as to accurate count real time and maintain CPU working power.

While NY8A056A operates in Standby mode and Halt mode, kinds of event will issue interrupt requests and can wake-up NY8A056A to enter Normal mode and Slow mode in order to process urgent events.

1.1 Features

- Wide operating voltage range: (@ 4 CPU clock per instruction, i.e. 4T mode)
  - 2.0V ~ 5.5V @system clock ≦8MHz.
  - 2.2V ~ 5.5V @system clock > 8MHz.
- Wide operating temperature: -40°C ~ 85°C.
- High ESD over ±8KV.
- High EFT over ±4KV with Noise Filter Enable. (Operating voltage @5V)
- 1K x 14 bits EPROM.
- 96 bytes SRAM.
- 16 general purpose I/O pins (GPIO), PA[7:0], PB[7:0], with independent direction control.

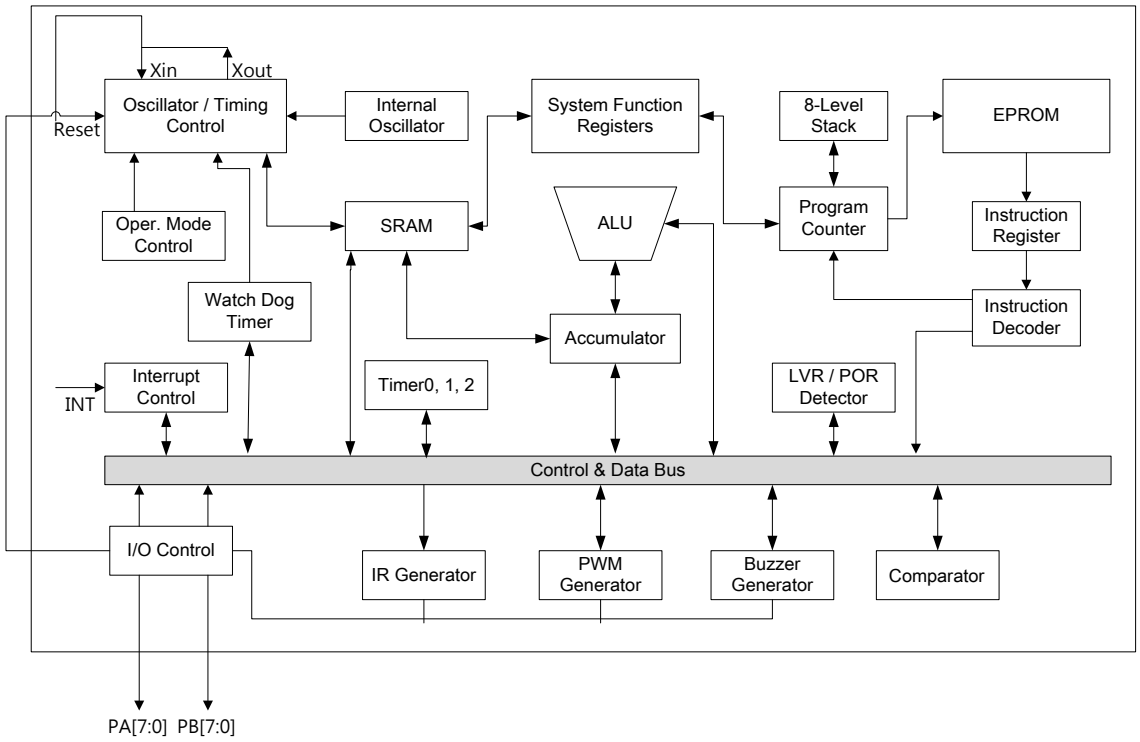
- PA[5, 3:0] and PB[3:0] have features of Pull-Low resistor for input pin.
- PA[7:6, 4:0] and PB[7:0] have features of Pull-High resistor, the value of Pull-High resistor can be 100kΩ or 1MΩ.
- PB[7:0] have features of Open-Drain output.
- PA[5] have feature of input or open-drain output.
- I/O ports output current mode can be constant sink, 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.
- Two 8-bit reload or continuous down-count timers (Timer1, 2), these 2 timers can be consisted to be a 16-bit timer.
- Two 8-bit resolution PWM (PWM1, 2) output.
- Two buzzer (BZ1, 2) output.
- Selectable 38/57KHz IR carrier frequency and high/low polarity according to data value.
- IR carrier sink current can be normal sink current or 340mA large sink current.
- 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.
- Dual-clock oscillation: System clock can switch between high oscillation and low oscillation.
  - High oscillation: E_HXT (External High Crystal Oscillator, above 6MHz)
    E_XT (External Crystal Oscillator, 455K~6MHz)
    I_HRC (Internal High Resistor/Capacitor Oscillator ranging from 1M~20MHz)
  - Low oscillation: E_LXT (External Low Crystal Oscillator, about 32KHz)
    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.
  - Timer2 underflow interrupt.
  - WDT timeout interrupt.
  - PB input change interrupt.
  - External interrupt.

- Comparator output status change interrupt.

- Seven interrupt events to wake-up NY8A056A from Standby mode:
  - Timer0 overflow interrupt.
  - Timer1 underflow interrupt.
  - Timer2 underflow interrupt.
  - WDT timeout interrupt.
  - PB input change interrupt.
  - External interrupt.
  - Comparator output status change interrupt.

- Three interrupt events to wake-up NY8A056A from Halt mode:
  - WDT timeout interrupt.
  - PB input change interrupt.
  - External interrupt.

1.2 Block Diagram



1.3 Pin Assignment

NY8A056A provides four kinds of package type which are SOP16, SOP18, DIP18 and SSOP20.

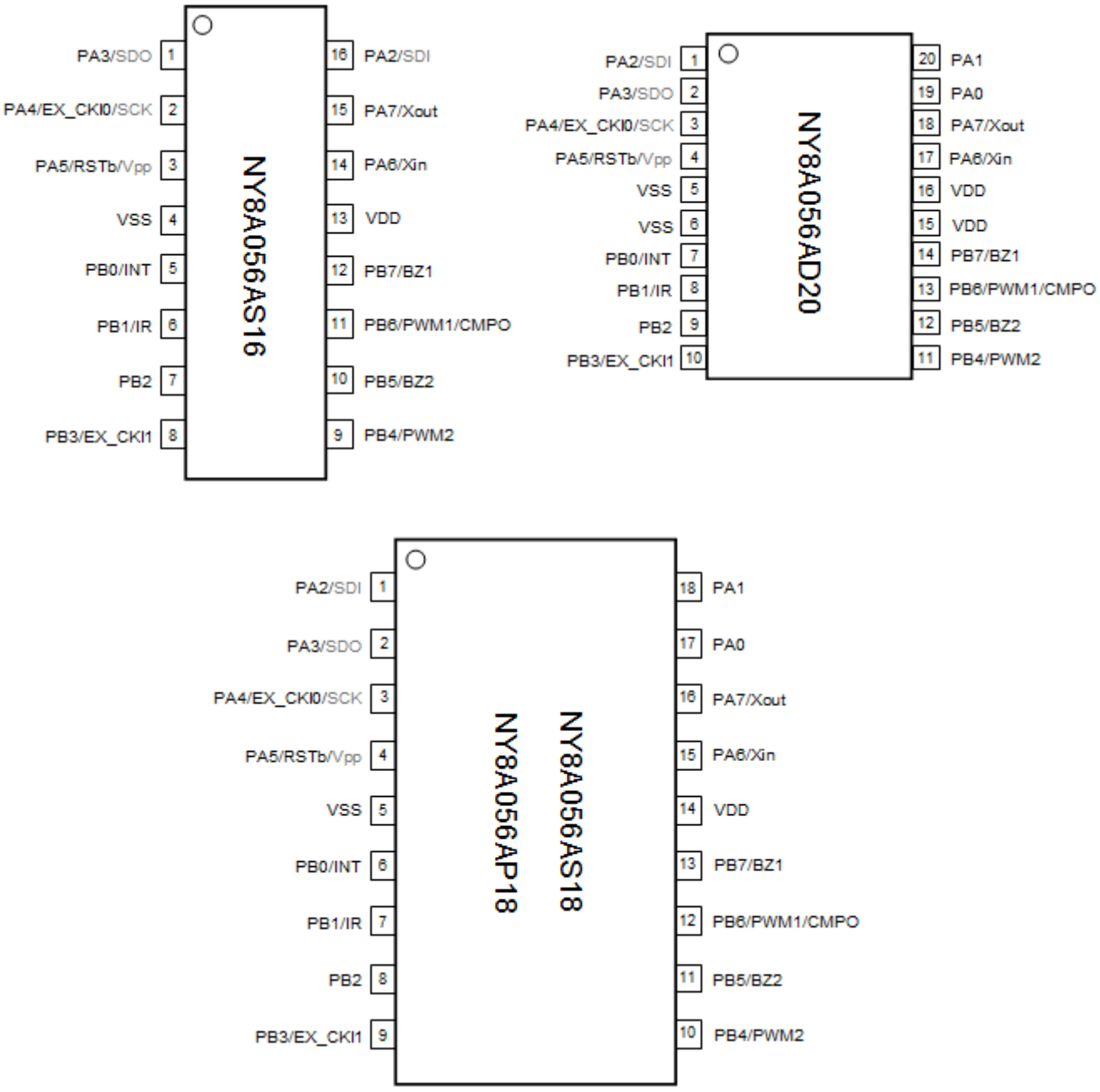


Figure 1 Package pin assignment

1.4 Pin Description

Pin Name	I/O	Description
PA0 ~ PA1	I/O	PA0~PA1 is bidirectional I/O pins, and can be analog input pins.
PA2/ SDI	I/O	PA2 is a bidirectional I/O pin, and can be analog input pin. PA2 can be programming pad SDI.
PA3/ SDO	I/O	PA3 is a bidirectional I/O pin, and can be analog input pin. PA3 can be programming pad SDO.
PA4/ EX_CKI0/ SCK	I/O	PA4 is a bidirectional I/O pin, and can be 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 8.5V, it also can make NY8A056A enter EPROM programming mode.
PA6/ Xin	I/O	PA6 is a bidirectional I/O pin, and can be analog input pin. PA6 can be the input pin of crystal oscillator Xin.
PA7/ Xout	I/O	PA7 is a bidirectional I/O pin, and can be analog input pin. PA7 can be the output pin of crystal oscillator Xout. PA7 also can be output of instruction clock.
PB0/ INT	I/O	PB0 is a bidirectional I/O pin, and can be analog input pin. PB0 can be the input pin of external interrupt when EIS=1 & INTIE=1.
PB1/ IR	I/O	PB1 is a bidirectional I/O pin, and can be analog input pin. If IR mode is enabled, this pin is IR carrier output with normal or large sink current.
PB2	I/O	PB2 is a bidirectional I/O pin, and can be analog input pin.
PB3/ EX_CKI1	I/O	PB3 is a bidirectional I/O pin, and can be analog input pin. PB3 can be the timer2 clock source EX_CKI1.
PB4/ PWM2	I/O	PB4 is a bidirectional I/O pin, and can be analog input pin. PB4 can be the output pin of PMW2 signal.
PB5/ BZ2	I/O	PB5 is a bidirectional I/O pin, and can be analog input pin. PB5 can be the output of Buzzer2 signal.
PB6/ PWM1/ CMPO	I/O	PB6 is a bidirectional I/O pin, and can be analog input pin. PB6 can be the output of PWM1 signal. PB6 can be the comparator output.
PB7/ BZ1	I/O	PB7 is a bidirectional I/O pin, and can be analog input pin. PB7 can be the output of Buzzer1 signal.
VDD	-	Positive power supply.
VSS	-	Ground.

2. Memory Organization

NY8A056A 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 NY8A056A 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.

NY8A056A provides instruction CALL, GOTOA, CALLA to address 256 location of program space. NY8A056A provides instruction GOTO to address 512 location of program space. NY8A056A also 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.

NY8A056A program ROM address 0x3FE~0x3FF are reserved space, if user tries to write code in these addresses will get unexpected false functions.

NY8A056A program ROM address 0x00E~0x00F are preset rolling code can be released and used as normal program space.

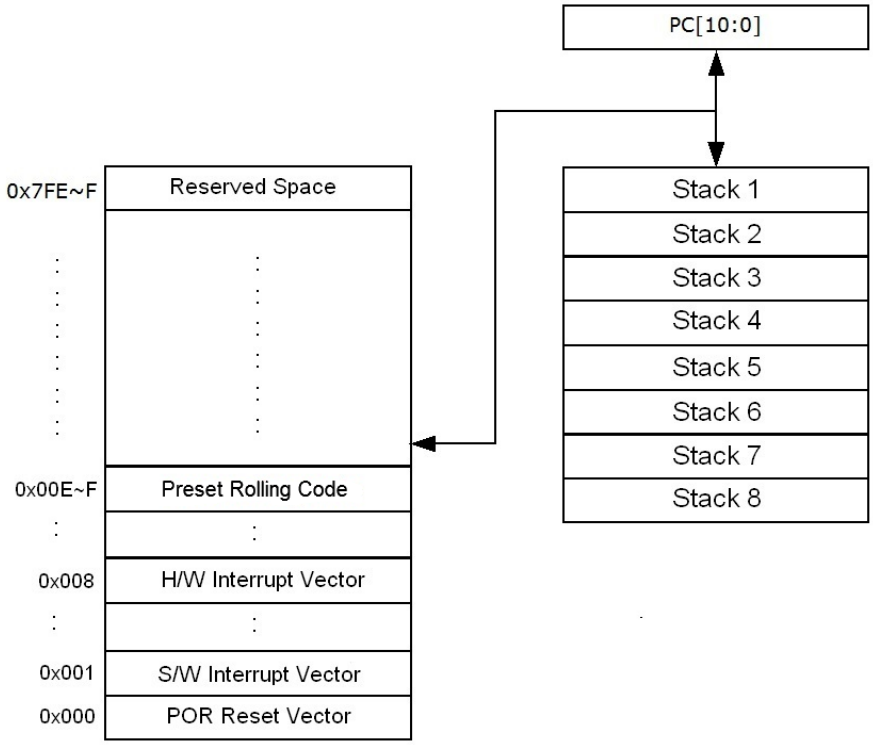


Figure 2 Program Memory Address Mapping

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). FSR[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 FSR[7:6] and the location selection is from FSR[5:0].

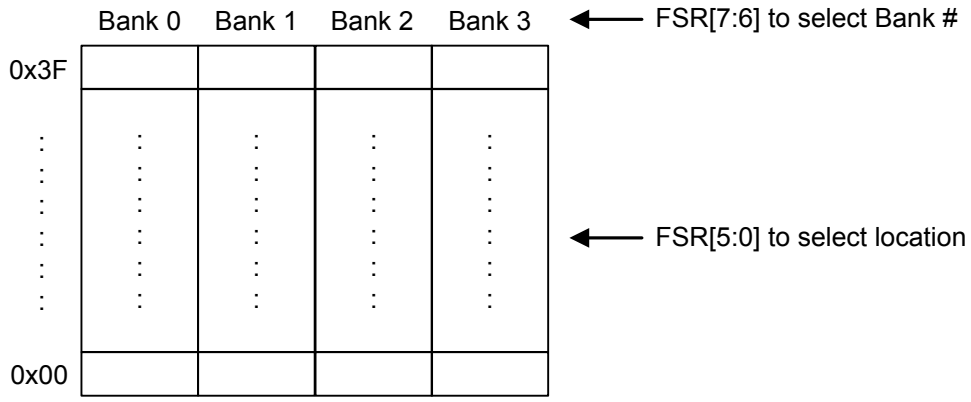


Figure 3 Indirect Addressing Mode of Data Memory Access

The direct addressing mode of data memory access is described below. The bank selection is determined by FSR[7:6] and the location selection is from instruction op-code[5:0] immediately.

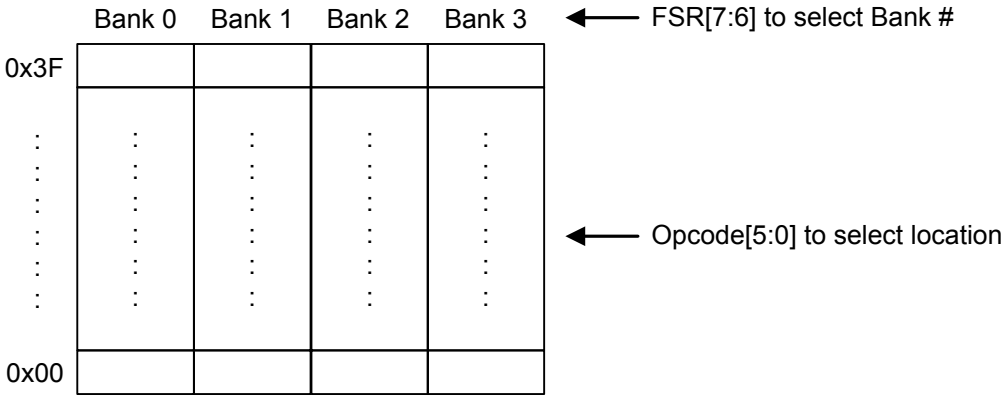


Figure 4 Direct Addressing Mode of Data Memory Access

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 0xF 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 0x10 to 0x3F of Bank 0 and 0x10 to 0x3F of Bank 1. Other bank in address from 0x10 to 0x3F are mapped back as the Table 1 shows.

The NY8A056A register name and address mapping of R-page SFR are described in the following table.

<table>
<tr><th>FSR[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="15">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 ~ 0x1F</td><td>General Purpose Register</td><td>General Purpose Register</td><td>Mapped to bank0</td><td>Mapped to bank1</td></tr>
<tr><td>0x20 ~ 0x3F</td><td>General Purpose Register</td><td>General Purpose Register</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. FSR[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	TMR2
0xA	PS0CV	T2CR1
0xB	-	T2CR2
0xC	BODCON	PWM2DUTY
0xD	-	PS2CV
0xE	CMPCR	BZ2CR
0xF	PCON1	OSCCR

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

3. Function Description

This chapter will describe the detailed operations of NY8A056A.

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 GOTO instruction, PC[8:0] is from instruction word and PC[9] is loaded from PCHBUF[1]. For CALL instruction, PC[7:0] is from instruction word and PC[9:8] is loaded from PCHBUF[1:0]. Moreover the next PC address, i.e. PC+1, will push onto top of Stack. 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	GP7	GP6	GP5	/TO	/PD	Z	DC	C
R/W Property			R/W	R/W	R/W	R/W(*2)	R/W(*1)	R/W	R/W	R/W
Initial Value			0	0	0	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.

GP7, GP6, GP5: General purpose read/write register bit.

(*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	BK[1:0]		FSR[5:0]					
R/W Property			R/W							
Initial Value			0	0	X	X	X	X	X	X

FSR[5:0]: Select one register out of 64 registers of specific Bank.

BK[1:0]: Bank register 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.

3.1.6 PortA (PortA Data Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PortA	R	0x5	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
R/W Property			R/W							
Initial Value			Data latch value is xxxxxxxx, read value is xxxxxxxx port value(PA7~PA0)							

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	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
R/W Property			R/W							
Initial Value			Data latch value is xxxxxxxx, read value is xxxxxxxx port value(PB7~PB0)							

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	EIS	LVDEN	GP4	LVREN	CMPEN	GP1	GP0
R/W Property			R/W							
Initial Value			1	0	0	0	1	0	0	0

GP4, GP1, GP0: General read/write register bits.

CMPEN: Enable/disable voltage comparator.
CMPEN=1, enable voltage comparator.
CMPEN=0, disable comparator.

LVREN: Enable/disable LVR.
LVREN=1, enable LVR.
LVREN=0, disable LVR.

LVDEN*: Enable/disable LVD.
LVDEN=1, enable LVD.
LVDEN=0, disable LVD.

Note: Must set LVDEN=0 to disable LVD function.

EIS: External interrupt select bit

EIS=1, PB0 is external interrupt.

EIS=0, PB0 is GPIO.

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	WUPB7	WUPB6	WUPB5	WUPB4	WUPB3	WUPB2	WUPB1	WUPB0
R/W Property			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			1	1	1	1	1	1	1	1

WUPBx: Enable/disable PBx wake-up function, 0 ≤ x ≤ 7.

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	DCMEN	XSPD_STP	-	-	-	GP2	PCHBUF[1:0]	
R/W Property			R/W	W	-	-	-	R/W		
Initial Value			0	0	X	X	X	000		

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

GP2: General read/write register bit.

XSPD_STP: Write 1 to stop crystal 32.768K speed-up function, write-only.

DCMEN: Enable/Disable constant sink function.

DCMEN=1, enable constant sink function.

DCMEN=0, disable constant sink function.

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	/PLPB0	/PLPA3	/PLPA2	/PLPA1	/PLPA0
R/W Property			R/W							
Initial Value			1	1	1	1	1	1	1	1

/PLPAx: Disable/enable PAx Pull-Low resistor, 0 ≤ x ≤ 3.

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

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

/PLPBx: Disable/enable PBx Pull-Low resistor, 0 ≤ 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	/PHPB7	/PHPB6	/PHPB5	/PHPB4	/PHPB3	/PHPB2	/PHPB1	/PHPB0
R/W Property			R/W							
Initial Value			1	1	1	1	1	1	1	1

/PHPBx: Disable/enable PBx Pull-High resistor, 0 ≤ x ≤ 7.

/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	-	WDTIE	T2IE	LVDIE	T1IE	INTIE	PBIE	T0IE
R/W Property			-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			x	0	0	0	0	0	0	0

T0IE: Timer0 overflow interrupt enable bit.

T0IE=1, enable Timer0 overflow interrupt.

T0IE=0, disable Timer0 overflow interrupt.

PBIE: PortB input change interrupt enable bit.

PBIE=1, enable PortB input change interrupt.

PBIE=0, disable PortB input change interrupt.

INTIE: External interrupt enable bit.

INTIE=1, enable external interrupt.

INTIE=0, disable external 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.

Note: Must set LVDIE=0.

T2IE: Timer2 underflow interrupt enable bit.

T2IE=1, enable Timer2 underflow interrupt.

T2IE=0, disable Timer2 underflow interrupt.

WDTIE: WDT timeout interrupt enable bit.

WDTIE=1, enable WDT timeout interrupt.

WDTIE=0, disable WDT timeout interrupt.

3.1.14 INTF (Interrupt Flag Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF	R	0xF	-	WDTIF	T2IF	LVDIF	T1IF	INTIF	PBIF	T0IF
R/W Property			-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value(note*)			0	0	0	0	0	0	0	0

T0IF: Timer0 overflow interrupt flag bit.

T0IF=1, Timer0 overflow interrupt is occurred.

T0IF must be clear by firmware.

PBIF: PortB input change interrupt flag bit.

PBIF=1, PortB input change interrupt is occurred.

PBIF must be clear by firmware.

INTIF: External interrupt flag bit.

INTIF=1, external interrupt is occurred.

INTIF 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.

T2IF: Timer2 underflow interrupt flag bit.

T2IF=1, Timer2 underflow interrupt is occurred.

T2IF must be clear by firmware.

WDTIF: WDT timeout interrupt flag bit.

WDTIF=1, WDT timeout interrupt is occurred.

WDTIF must be clear by firmware.

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

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	INTEDG	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
111	1:256	1:128	1:256

Table 3 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=1, External clock on pin EX_CKI0 or Low Oscillator Frequency (I_LRC or E_LXT) is selected.

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

INTEDG: Edge selection of external interrupt.

INTEDG=1, INTIF will be set while rising edge occurs on pin PB0.

INTEDG=0, INTIF will be set while falling edge occurs on pin PB0.

LCKTM0: When T0CS=1, timer 0 clock source can be optionally selected to be low-frequency oscillator.

T0CS=0, Instruction clock F_{INST} is selected as timer0 clock source.

T0CS=1, LCKTM0=0, external clock on pin EX_CKI0 is selected as timer0 clock source.

T0CS=1, LCKTM0=1, Low Oscillator Frequency (I_LRC or E_LXT, depends on configuration word Low Oscillator Frequency) output replaces pin EX_CKI0 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	IOPA7	IOPA6	IOPA5	IOPA4	IOPA3	IOPA2	IOPA1	IOPA0
R/W Property			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			1	1	1	1	1	1	1	1

IOPAx: PAx I/O mode selection, 0 ≤ x ≤ 7.

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	IOPB7	IOPB6	IOPB5	IOPB4	IOPB3	IOPB2	IOPB1	IOPB0
R/W Property			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			1	1	1	1	1	1	1	1

IOPBx: PBx I/O mode selection, 0 ≤ x ≤ 7.

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	/PHPA7	/PHPA6	/PLPA5	/PHPA4	/PHPA3	/PHPA2	/PHPA1	/PHPA0
R/W Property			R/W							
Initial Value			1	1	1	1	1	1	1	1

/PHPAx: Enable/disable Pull-High resistor of PAx, x=0~4, 6~7.

/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.

Note: When PA6 and PA7 are used as crystal oscillator pads, the Pull-High resistor should not enable. Or the oscillation may fail.

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	ODPB7	ODPB6	ODPB5	ODPB4	ODPB3	ODPB2	ODPB1	ODPB0
R/W Property			R/W							
Initial Value			0	0	0	0	0	0	0	0

ODPBx: Enable/disable open-drain of PBx, 0 ≤ x ≤ 7.

ODPBx=1, enable open-drain of PBx.

ODPBx=0, disable open-drain of PBx.

GP3: General purpose register bit.

3.3.6 CMPCR (Comparator voltage select Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPCR	F	0xE	PS3	PS2	PS1	PS0	VS3	VS2	VS1	VS0
R/W Property			R/W							
Initial Value			0	0	0	0	0	0	0	0

VS[3:0], PS[3:0]: When VS[3:0]=0, the comparator is in P2P mode, else it is in P2V mode.

When the comparator is in P2V mode, VS[3:0] select one of 15 reference voltages as the inverting input of the comparator. And PS[3:0] determine one of 15 pads as the non-inverting input of the comparator.

When the comparator is in P2P mode, VS[3:0] is fixed 0, and PS[3:0] select 2 pads out of 16 combinations to be the inverting and non-inverting input of the comparator. For detail P2P mode please see function description comparator section.

VS[3:0]	V- of Comparator	PS[3:0]	Selected pad
0000	P2P mode	0000	PA0
0001	1 / 16 V_{DD}	0001	PA1
0010	2 / 16 V_{DD}	0010	PA2
0011	3 / 16 V_{DD}	0011	PA3
0100	4 / 16 V_{DD}	0100	PA4
0101	5 / 16 V_{DD}	0101	NC
0110	6 / 16 V_{DD}	0110	PA6

VS[3:0]	V- of Comparator	PS[3:0]	Selected pad
0111	7 / 16 V_{DD}	0111	PA7
1000	8 / 16 V_{DD}	1000	PB0
1001	9 / 16 V_{DD}	1001	PB1
1010	10 / 16 V_{DD}	1010	PB2
1011	11 / 16 V_{DD}	1011	PB3
1100	12 / 16 V_{DD}	1100	PB4
1101	13 / 16 V_{DD}	1101	PB5
1110	14 / 16 V_{DD}	1110	PB6
1111	15 / 16 V_{DD}	1111	PB7

Table 4 P2V Mode

PS[3:0]	Non-inverting input	Inverting input
0000	PA0	PA1
0001	PA1	PA0
0010	PA2	PA3
0011	PA3	PA2
0100	PA0	PB3
0101	PA1	PB2
0110	PA2	PB5
0111	PA3	PB4
1000	PB2	PA1
1001	PB3	PA0
1010	PB4	PA3
1011	PB5	PA2
1100	PB2	PB3
1101	PB3	PB2
1110	PB4	PB5
1111	PB5	PB4

Table 5 P2P Mode (VS[3:0] = 4'b0000)

3.3.7 PCON1 (Power Control Register1)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON1	F	0xF	GIE	LVDOUT	GP5	LVDS2	LVDS1	LVDS0	GP1	T0EN
R/W Property			R/W(1*)	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			0	x	0	1	1	1	0	1

T0EN: Enable/disable Timer0.

T0EN=1, enable Timer0.

T0EN=0, disable Timer0.

LVDS2~0: Select one of the 8 LVD voltage.

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

Table 6 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.

GP1, GP5: General purpose read/write register.

(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 8-bit down-count Timer1. When writing register TMR1, it will both write data to timer1 reload register and update Timer1 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	-	-	-	T1OS	T1RL	T1EN
R/W Property			W	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.

T1RL=0, continuous down-count from 0xFF 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, 0xFF 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 7 Timer1 Functionality

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 PB6.

PWM1OEN=0, PB6 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 8 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							

This register is write-only. After Timer1 is enabled and start down-count, PWM1 output will keep at inactive state. While Timer1 value is equal to PWM1DUTY, PWM1 output will become active state until underflow is occurred.

Moreover, the reload value of Timer1 stored on register TMR1 is used to define the PWM1 frame rate and register PWM1DUTY 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 9 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	IROSC358M	-	-	-	-	IRCSEL	IRF57K	IREN
R/W Property			W	-	-	-	-	W	W	W
Initial Value			0	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.

IROSC358M: When external crystal is used, this bit is determined according to what kind of crystal is used.

This bit is ignored if internal high frequency oscillation is used.

IROSC358M=1, crystal frequency is 3.58MHz.

IROSC358M=0, crystal frequency is 455KHz.

Note:

1. Only high oscillation (F_{HOSC}) (See section 3.15) 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)
Xtal 3.58MHz	64	96	Xtal mode & IROSC358M=1
Xtal 455KHz	8	12	Xtal mode & IROSC358M=0

Table 10 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	-	-	-	-	-	TBHP2	TBHP1	TBHP0
R/W Property			-	-	-	-	-	R/W	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[2:0] and ACC. ACC is the Low Byte of PC[9:0] and TBHP[1:0] is the high byte of PC[9:0]. TBHP[2] is general register for NY8A056A.

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.

3.4.10 TMR2 (Timer2 Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMR2	S	0x9	TMR2[7:0]							
R/W Property			R/W							
Initial Value			XXXXXXXX							

When reading register TMR2, it will obtain current value of 8-bit down-count Timer2. When writing register TMR2, it will both write data to timer2 reload register and update Timer2 current content.

3.4.11 T2CR1 (Timer2 Control Register1)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2CR1	S	0xA	PWM2OEN	PWM2OAL	-	-	-	T2OS	T2RL	T2EN
R/W Property			W	W	-	-	-	R/W	R/W	R/W
Initial Value			0	0	X	X	X	0	0	0

This register is used to configure Timer2 functionality.

T2EN: Enable/disable Timer2.

T2EN=1, enable Timer2.

T2EN=0, disable Timer2.

T2RL: Configure Timer2 down-count mechanism while Non-Stop mode is selected (T2OS=0).

T2RL=1, initial value is reloaded from reload register TMR2.

T2RL=0, continuous down-count from 0xFF when underflow is occurred.

T2OS: Configure Timer2 operating mode while underflow is reached.

T2OS=1, One-Shot mode. Timer2 will count once from the initial value to 0x00.

T2OS=0, Non-Stop mode. Timer2 will keep down-count after underflow.

T2OS	T2RL	Timer2 Down-Count Functionality
0	0	Timer2 will count from reload value down to 0x00. When underflow is reached, 0xFF is reloaded and continues down-count.
0	1	Timer2 will count from reload value down to 0x00. When underflow is reached, reload value is reloaded and continues to down-count.
1	x	Timer2 will count from initial value down to 0x00. When underflow is reached, Timer2 will stop down-count.

Table 11 Timer2 Functionality

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=2, PWM2 output will be present on PB4.

PWM2OEN=0, PB4 is GPIO.

3.4.12 T2CR2 (Timer2 Control Register2)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2CR2	S	0xB	-	-	T2CS	T2CE	/PS2EN	PS2SEL[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 Timer2 functionality.

PS2SEL[2:0]: Prescaler2 dividing rate selection.

PS2SEL[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 12 Prescaler2 Dividing Rate

Note: Always set PS2SEL[2:0] at /PS2EN=1, or interrupt may be falsely triggered.

/PS2EN: Disable/enable Prescaler2.

/PS2EN=1, disable Prescaler2.

/PS2EN=0, enable Prescaler2.

T2CE: Timer2 external clock edge selection.

T2CE=1, Timer2 will decrease one while high-to-low transition occurs on pin EX_CKI1.

T2CE=0, Timer2 will decrease one while low-to-high transition occurs on pin EX_CKI1.

T2CS: Timer2 clock source selection.

T2CS=1, External clock on pin EX_CKI1 is selected.

T2CS=0, Instruction clock is selected.

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							

This register is write-only. After Timer2 is enabled and start down-count, PWM2 output will keep at inactive state. While Timer2 value is equal to PWM2DUTY, PWM2 output will become active state until underflow is occurred.

Moreover, the reload value of Timer2 stored on register TMR2 is used to define the PWM2 frame rate and register PWM2DUTY is used to define the duty cycle of PWM2.

3.4.14 PS2CV (Prescaler2 Counter Value Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS2CV	S	0xD	PS2CV[7:0]							
R/W Property			R							
Initial Value			1	1	1	1	1	1	1	1

While reading PS2CV, it will get current value of Prescaler2 counter.

3.4.15 BZ2CR (Buzzer2 Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BZ2CR	S	0xE	BZ2EN	-	-	-	BZ2FSEL[3:0]			
R/W Property			W	-	-	-	W			
Initial Value			0	X	X	X	1	1	1	1

BZ2FSEL[3:0]: Frequency selection of BZ2 output.

BZ2FSEL[3:0]	BZ2 Frequency Selection	
	Clock Source	Dividing Rate
0000	Prescaler2 output	1:2
0001		1:4
0010		1:8
0011		1:16
0100		1:32
0101		1:64
0110		1:128
0111		1:256

BZ2FSEL[3:0]	BZ2 Frequency Selection	
	Clock Source	Dividing Rate
1000	Timer2 output	Timer2 bit 0
1001		Timer2 bit 1
1010		Timer2 bit 2
1011		Timer2 bit 3
1100		Timer2 bit 4
1101		Timer2 bit 5
1110		Timer2 bit 6
1111		Timer2 bit 7

Table 13 Buzzer2 Output Frequency Selection

BZ2EN: Enable/Disable BZ2 output.

BZ2EN=1, enable Buzzer2.

BZ2EN=0, disable Buzzer2.

3.4.16 OSCCR (Oscillation Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSCCR	S	0xF	CMPOUT	CMPOE	CMPIF	CMPIE	OPMD[1:0]		STPHOSC	SELHOSC
R/W Property			R	R/W	R/W	R/W	R/W		R/W	R/W
Initial Value			X	0	0	0	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 14 Selection of Operating Mode by OPMD[1:0]

CMPIE: Enable/Disable of comparator interrupt.

CMPIE=1, Enable of comparator interrupt.

CMPIE=0, Disable of comparator interrupt.

CMPIF: Comparator output change state interrupt is occurred.

CMPIF=1, comparator interrupt is occurred.

CMPIF must be clear by firmware.

CMPOE: Disable/enable comparator output to pad PB6.

CMPOE=1, enable comparator output to pad PB6.

CMPOE=0, disable comparator output to pad PB6.

Note: Comparator output to pad PB6 has higher priority than PWM1 output to pad PB6.

CMPOUT: Comparator output status, read-only.

Note: STPHOSC cannot be changed with SELHOSC or OPMD at the same time. STPHOSC cannot be changed with OPMD at the same time during SELHOSC=1.

3.5 I/O Port

NY8A056A provides 16 I/O pins which are PA[7:0] and PB[7:0]. 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[7:0] define the input/output direction of PA[7:0]. Register IOSTB[7:0] define the input/output direction of PB[7:0].

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[7:6, 4:0] are used to enable or disable Pull-High resistor of PA[7:6, 4:0]. Register APHCON[5] and ABPLCON[3:0] are used to enable or disable Pull-Low resistor of PA[5, 3:0]. Register BPHCON[7:0] are used to enable or disable Pull-High resistor of PB[7:0]. Register ABPLCON[7:4] are used to enable or disable Pull-Low resistor of PB[3:0].

Each I/O group(PA,PB) can set its Pull-High strength(100kΩ or 1MΩ) by configuration word phstrengthA and phstrengthB.

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[7:0] determine PB[7:0] is Open-Drain or not.

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

Feature		PA[3:0]	PA[7:6]PA[4]	PA[5]	PB[7:4]	PB[3:0]
Input	Pull-High Resistor	V	V	X	V	V
	Pull-Low Resistor	V	X	V	X	V
Output	Open-Drain	X	X	always	V	V

Table 15 Summary of Pad I/O Feature

The level change on each I/O pin of PB may generate interrupt request. Register BWUCON[7:0] will select which I/O pin of PB may generate this interrupt. As long as any pin of PB is selected by corresponding bit of BWUCON, the register bit PBIF (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 PBIE (INTE[1]) and GIE (PCON1[7]) are both set to 1.

There is one external interrupt provided by NY8A056A. When register bit EIS (PCON[6]) is set to 1, PB0 is used as input pin for external interrupt.

Note: When PB0 is both set as level change operation and external interrupt, the external interrupt will have higher priority, and the PB0 level change operation will be disabled. But PB7~PB1 level change function are not affected.

NY8A056A can provide IR carrier generation. IR carrier generation is enabled by register bit IREN (IRCR[0]) and carrier will be present on a PB1 pin. Configuration word IR Current determines sink current value of IR carrier. When IR Current=Large, the sink current=340mA, When IR Current=Normal, the sink current=60mA.

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 NY8A056A to enter reset process.

When external crystal (E_HXT, E_XT or E_LXT) is adopted for high oscillation or low oscillation according to setting of configuration words, PA6 will be used as crystal input pin (Xin) and PA7 will be used as crystal output pin (Xout).

When I_HRC or I_LRC mode is selected as system oscillation and E_HXT, E_XT or E_LXT is not adopted, instruction clock is observable on PA7 if a configuration word is enabled.

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. PB3 can be timer2 external clock source EX_CKI1 if T2CS=1.

Moreover, PB6 can be PWM1 output If T1CR1[7] PWM1OEN=1. PB7 can be Buzzer1 output if BZ1CR[7] BZ1EN=1. PB4 can be PWM2 output If T2CR1[7] PWM2OEN=1. PB5 can be Buzzer2 output if BZ2CR[7] BZ2EN=1.

When configured as output, the sink current of each pin can be normal (20mA for V_{DD} =3V), large(60mA for V_{DD} =3V) or constant(20mA for V_{DD} =2~5.5V) according to configuration words. Check the following table for sink current mode setting:

Configuration word	Normal sink	Large sink	Constant sink
PXcurrent	0	1	1
PXcsc	0	0	1

Table 16 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.

PULLUP1M_ENB: enable 1MΩ Pull-High.

PULLUP100K_ENB: enable 100kΩ 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.

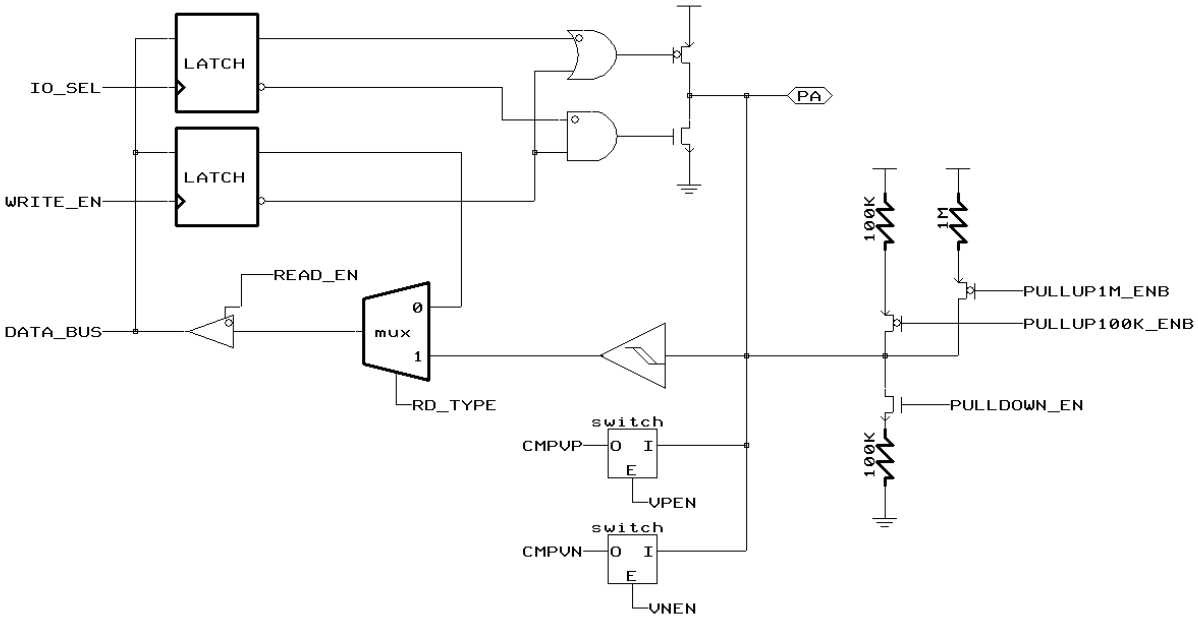


Figure 5 Block Diagram of PA[3:0]

IO_SEL: set pad attribute as input or output

WRITE_EN: write data to pad.

READ_EN: read pad.

PULLUP1M_ENB: enable 1MΩ Pull-High.

PULLUP100K_ENB: enable 100kΩ Pull-High.

VPEN: enable pad to comparator non-inverting input.

CMPVP: comparator non-inverting input.

RD_TYPE: select read pin or read latch.

EX_CKI0: external clock for timer0, 1.

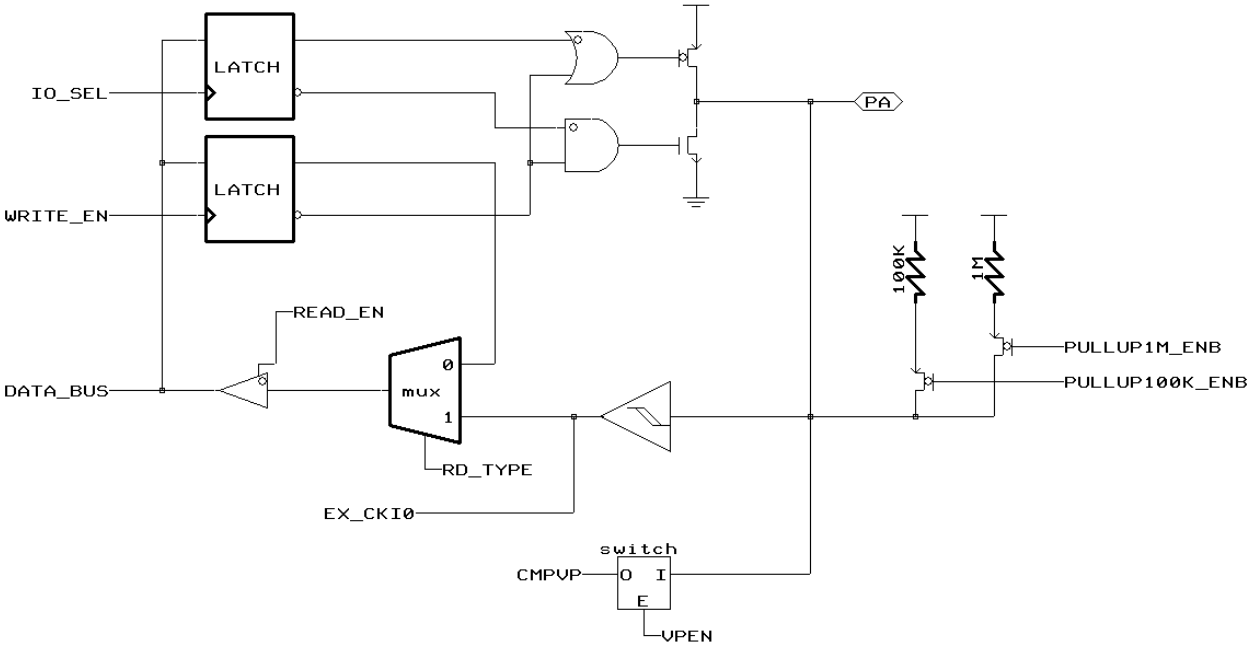


Figure 6 Block Diagram of PA4

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.

PULLDOWN_EN: enable Pull-Low.

VPEN: enable pad to comparator non-inverting input.

CMPVP: comparator non-inverting input.

RD_TYPE: select read pin or read latch.

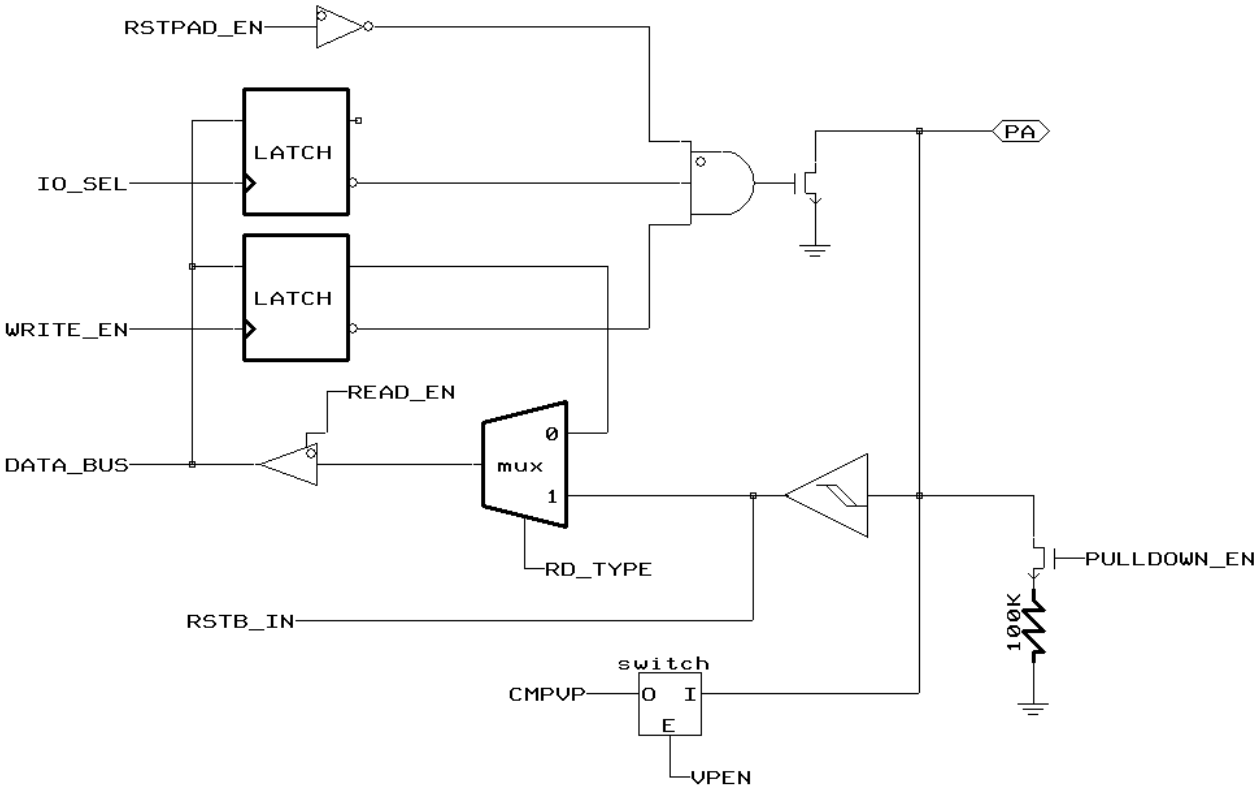


Figure 7 Block Diagram of PA5

XTL_EN: enable crystal oscillation mode.

IO_SEL: set pad attribute as input or output

WRITE_EN: write data to pad.

READ_EN: read pad.

PULLUP1M_ENB: enable 1MΩ Pull-High.

PULLUP100K_ENB: enable 100kΩ Pull-High.

VPEN: enable pad to comparator non-inverting input.

CMPVP: comparator non-inverting input.

RD_TYPE: select read pin or read latch.

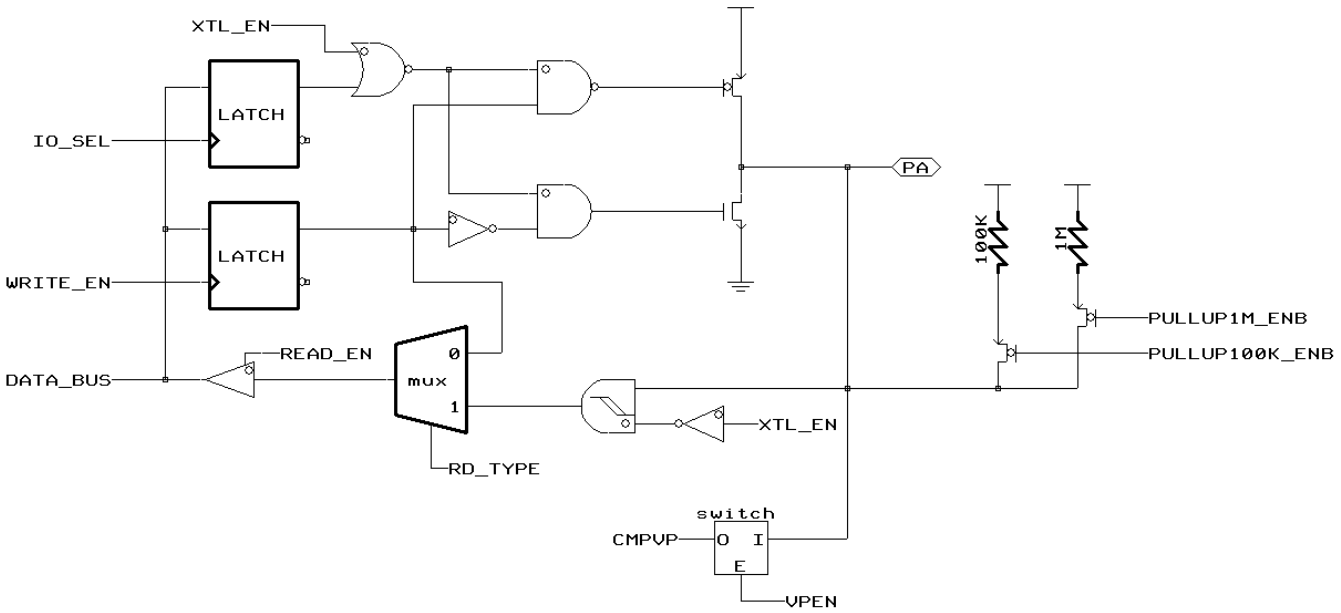


Figure 8 Block Diagram of PA6, PA7

IO_SEL: set pad attribute as input or output

WRITE_EN: write data to pad.

READ_EN: read pad.

OD_EN: enable open-Drain.

PULLUP1M_ENB: enable 1MΩ Pull-High.

PULLUP100K_ENB: enable 100kΩ Pull-High.

PULLDOWN_EN: enable Pull-Low.

VPEN: enable pad to comparator non-inverting input.

CMPVP: comparator non-inverting input.

RD_TYPE: select read pin or read latch.

EIS: external interrupt function enable.

INTEDGE: external interrupt edge select.

EX_INT: external interrupt signal.

WUB: port B wake-up enable.

SET_PBIF: port B wake-up flag.

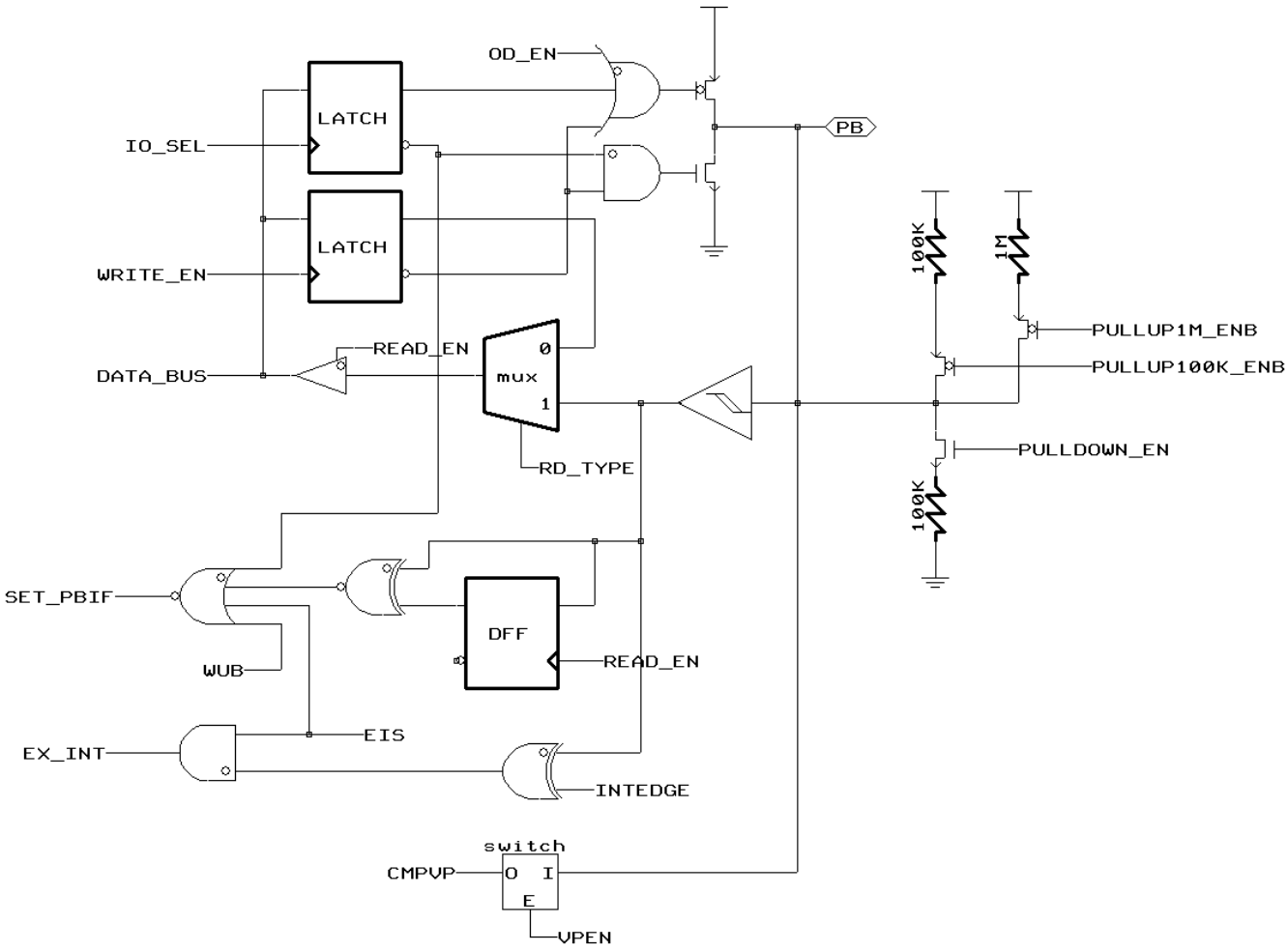


Figure 9 Block Diagram of PB0

IO_SEL: set pad attribute as input or output

WRITE_EN: write data to pad.

READ_EN: read pad.

OD_EN: enable open-Drain.

PULLUP1M_ENB: enable 1MΩ Pull-High.

PULLUP100K_ENB: enable 100kΩ Pull-High.

PULLDOWN_EN: enable Pull-Low.

VPEN: enable pad to comparator non-inverting input.

CMPVP: comparator non-inverting input.

RD_TYPE: select read pin or read latch.

WUB: port B wake-up enable.

SET_PBIF: port B wake-up flag.

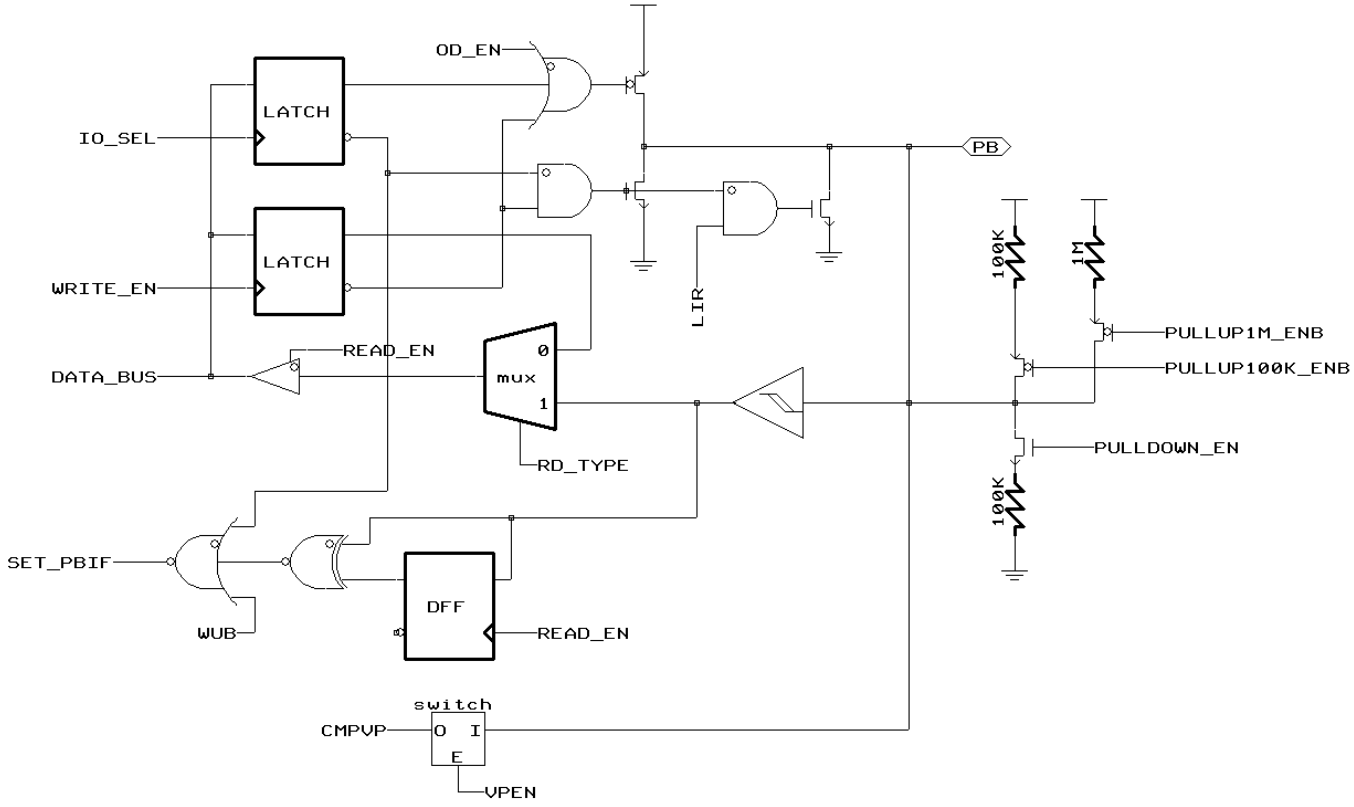


Figure 10 Block Diagram of PB1

IO_SEL: set pad attribute as input or output

WRITE_EN: write data to pad.

READ_EN: read pad.

OD_EN: enable open-Drain.

PULLUP1M_ENB: enable 1MΩ Pull-High.

PULLUP100K_ENB: enable 100kΩ 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.

WUB: port B wake-up enable.

SET_PBIF: port B wake-up flag.

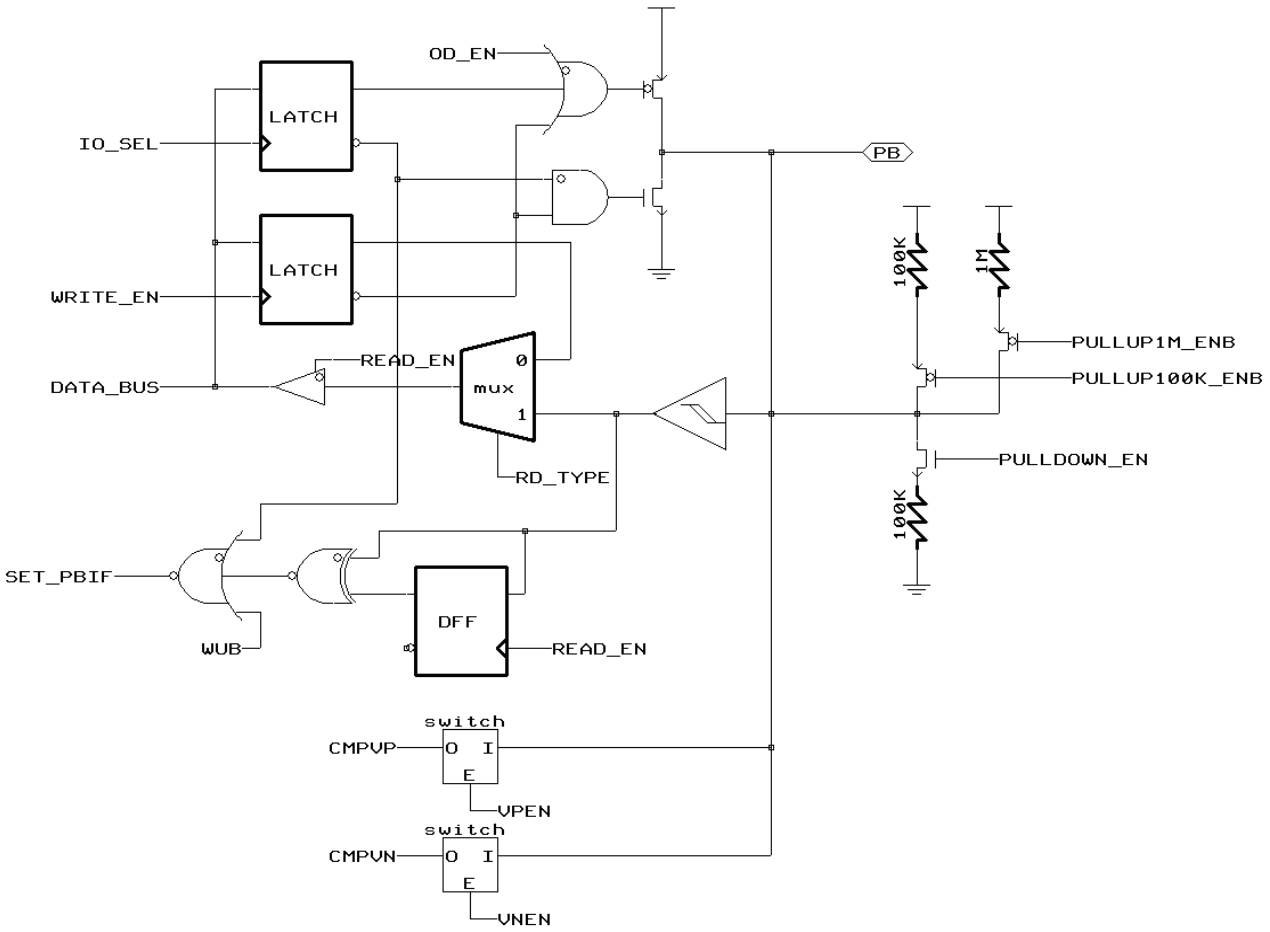


Figure 11 Block Diagram of PB2

IO_SEL: set pad attribute as input or output

WRITE_EN: write data to pad.

READ_EN: read pad.

EX_CKI1: external clock for timer2.

OD_EN: enable open-Drain.

PULLUP1M_ENB: enable 1MΩ Pull-High.

PULLUP100K_ENB: enable 100kΩ 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.

WUB: port B wake-up enable.

SET_PBIF: port B wake-up flag.

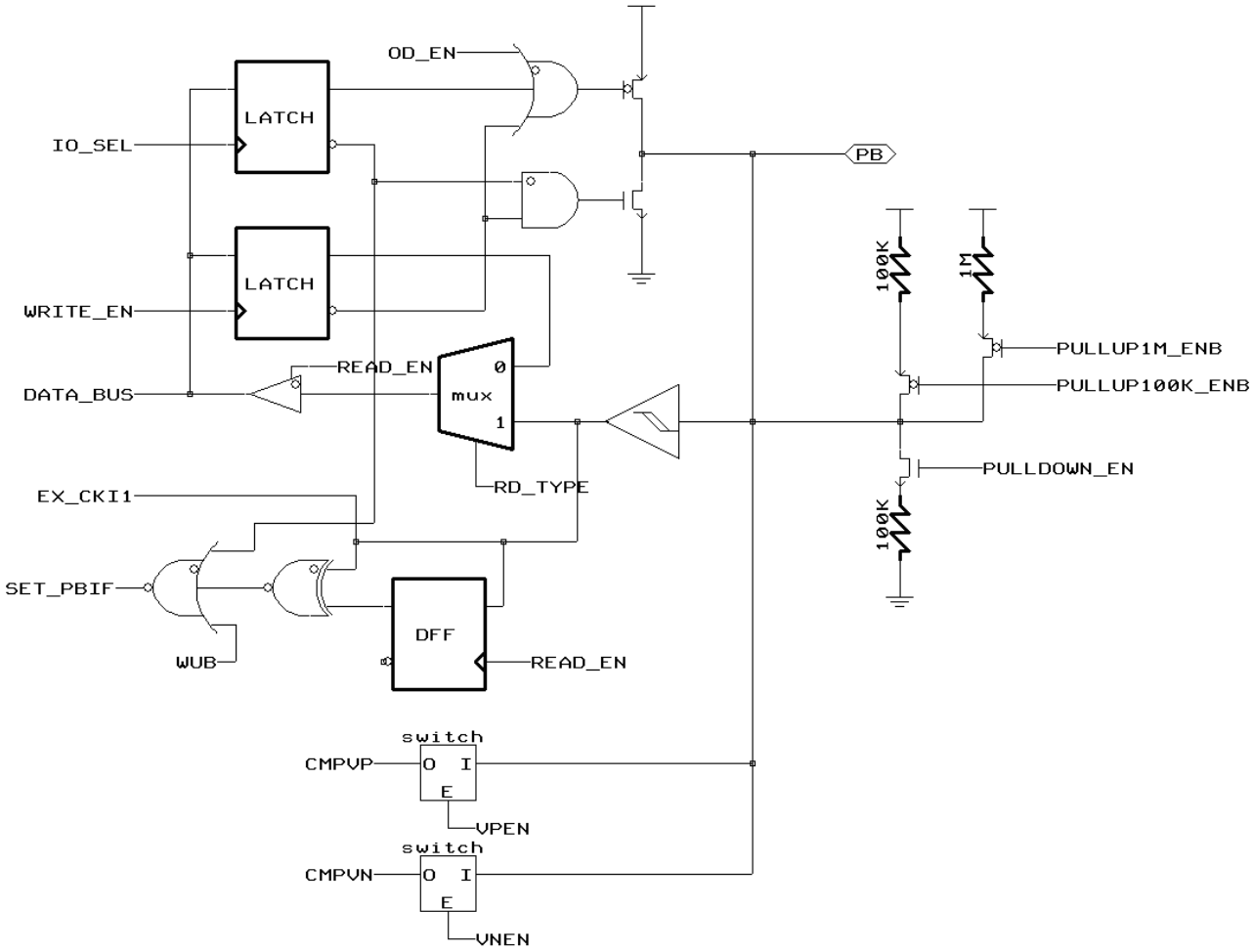


Figure 12 Block Diagram of PB3

IO_SEL: set pad attribute as input or output

WRITE_EN: write data to pad.

READ_EN: read pad.

OD_EN: enable open-Drain.

PULLUP1M_ENB: enable 1MΩ Pull-High.

PULLUP100K_ENB: enable 100kΩ Pull-High.

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.

WUB: port B wake-up enable.

SET_PBIF: port B wake-up flag.

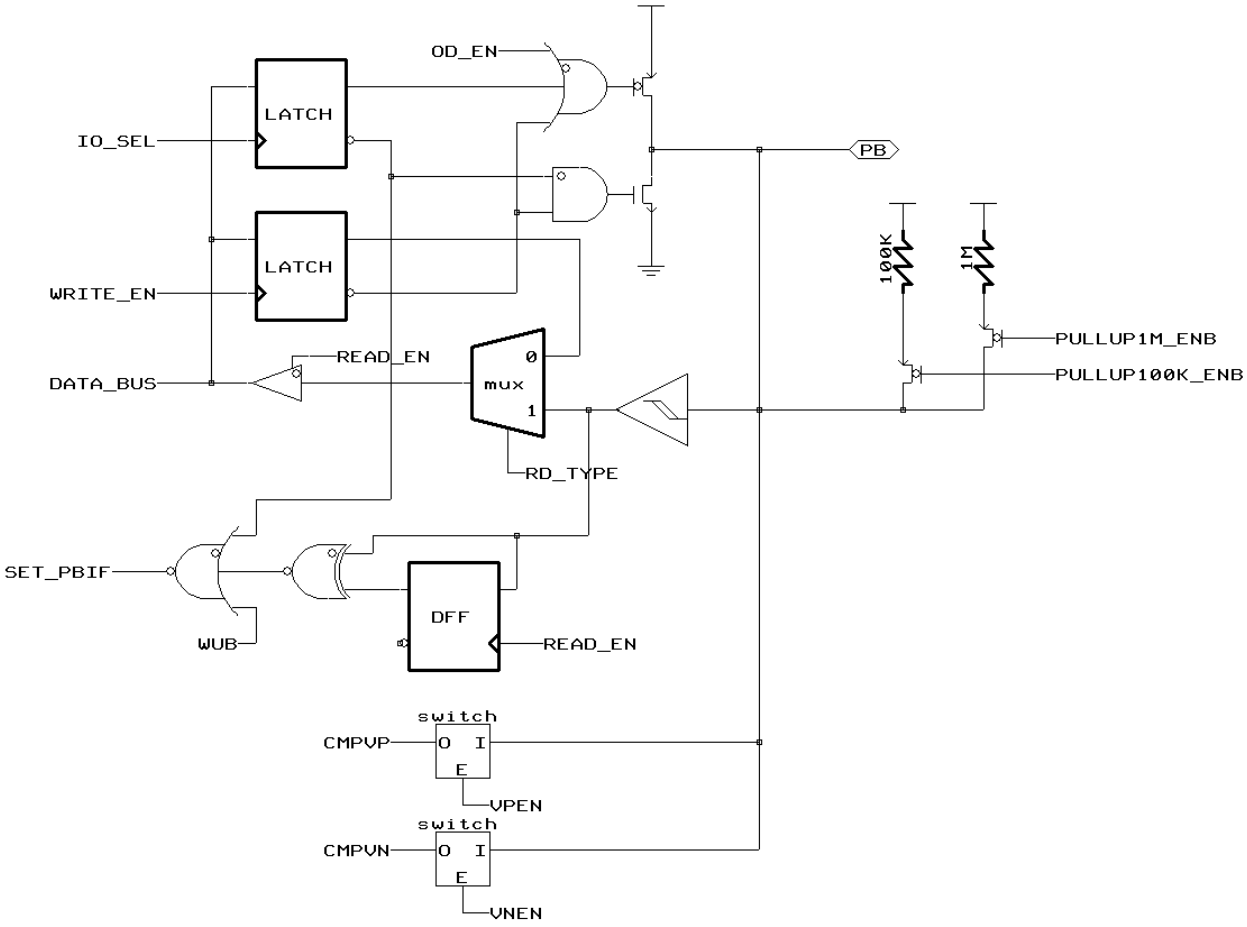


Figure 13 Block Diagram of PB4, PB5

IO_SEL: set pad attribute as input or output

WRITE_EN: write data to pad.

READ_EN: read pad.

OD_EN: enable open-Drain.

PULLUP1M_ENB: enable 1MΩ Pull-High.

PULLUP100K_ENB: enable 100kΩ Pull-High.

VPEN: enable pad to comparator non-inverting input.

CMPVP: comparator non-inverting input.

RD_TYPE: select read pin or read latch.

WUB: port B wake-up enable.

SET_PBIF: port B wake-up flag.

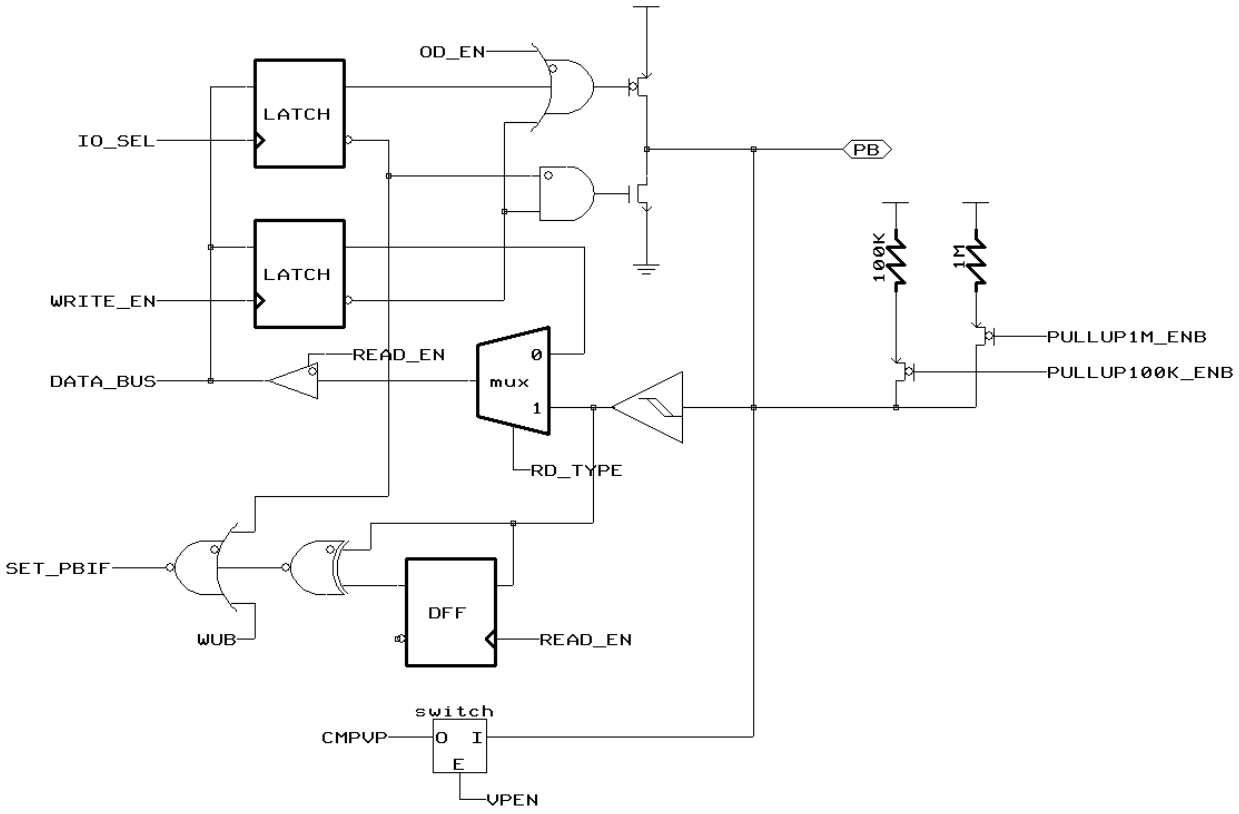


Figure 14 Block Diagram of PB6, PB7

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, external pin EX_CKI0 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 0, EX_CKI0 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 (I_LRC or E_LXT, depends on configuration word) 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	
E_LXT	1	1	1	1
I_LRC	1	1	1	0

Table 17 Summary of Timer0 clock source control

Moreover the active edge of EX_CKI0 or Low Oscillator Frequency to increase Timer0 can be selected by register bit T0CE (T0MD[4]). When T0CE is 1, high-to-low transition on EX_CKI0 or Low Oscillator Frequency will increase Timer0. When T0CE is 0, low-to-high transition on EX_CKI0 or Low Oscillator Frequency will increase Timer0. 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 Tmer0 and WDT is shown in the figure below.

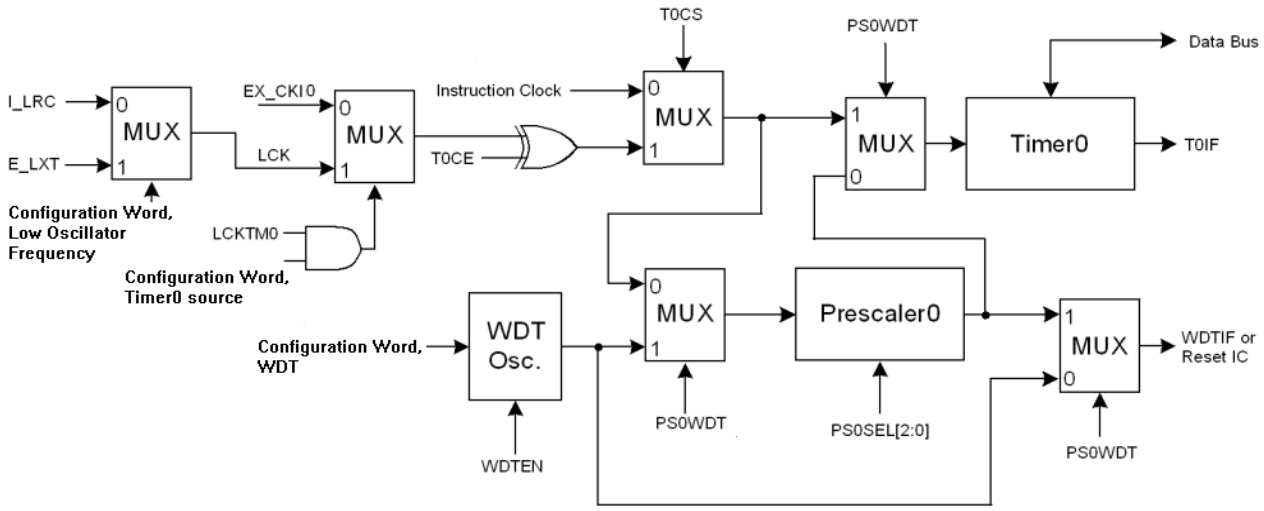


Figure 15 Block Diagram of Timer0 and WDT

3.7 Timer1 / PWM1 / Buzzer1

Timer1 is an 8-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. A write to the Timer1 will both write to a timer1 reload register (T1rld) and timer1 counter. 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.

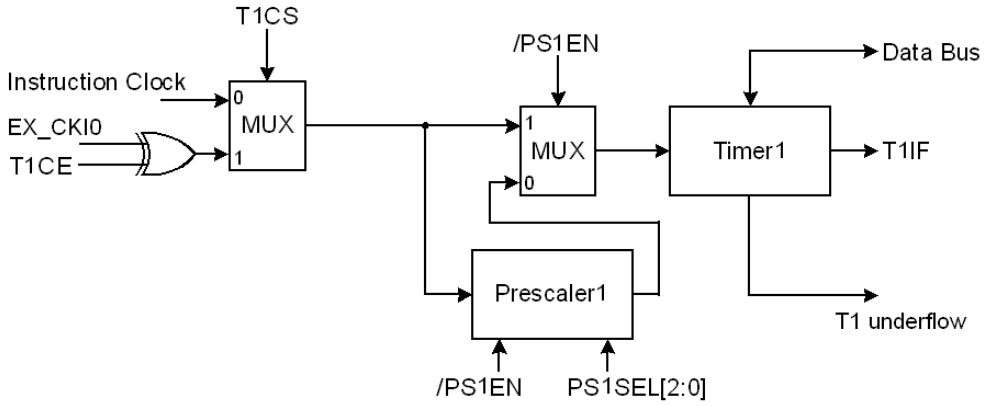


Figure 16 Block Diagram of Timer1

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 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 will be restored and start next down-count from this initial value. When T1RL is 0, Timer1 will start next down-count from 0xFF.

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.

Note that the timer1 underflow output is connected to timer2 for 16-bit timer option.

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

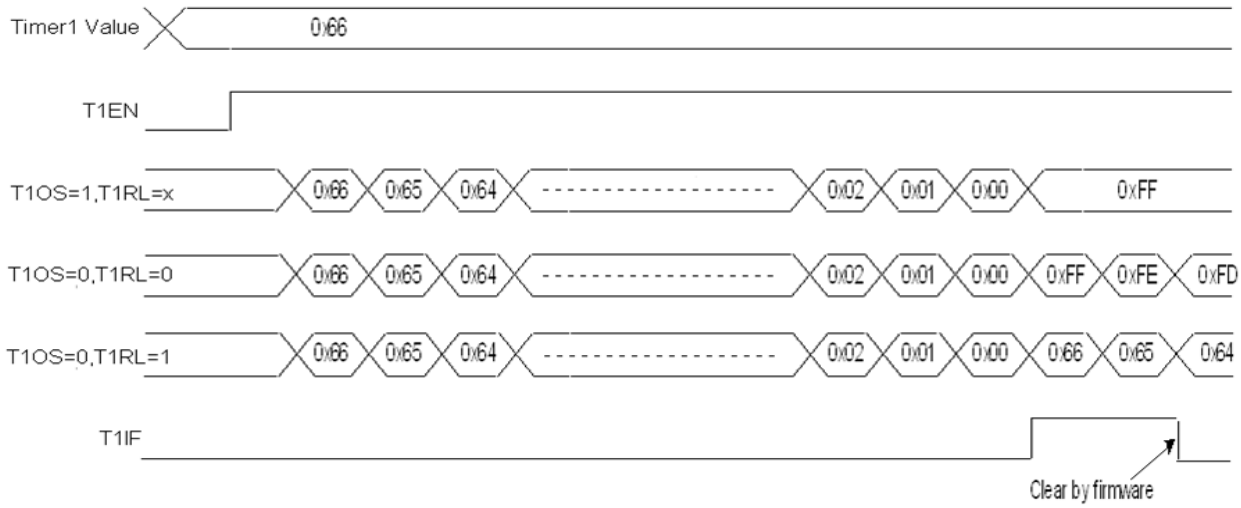


Figure 17 Timer1 Timing Chart

The PWM1 output can be available on I/O pin PB6 when register bit PWM1OEN (T1CR1[7]) is set to 1. Moreover, PB6 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 register PWM1DUTY. When PWM1DUTY is 0, PWM1 output will be never active. When PWM1DUTY is 0xFF, PWM1 output will be active for 255 Timer1 input clocks. The frame rate is determined by TMR1 initial value. Therefore, PWM1DUTY value must be less than or equal to TMR1. The block diagram of PWM1 is illustrated in the following figure.

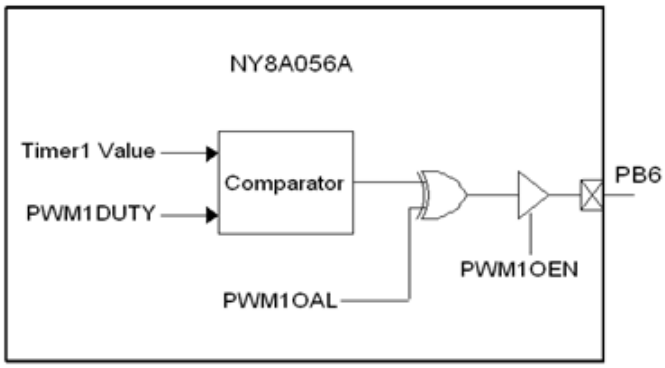


Figure 18 PWM1 Block Diagram

The Buzzer1 output (BZ1) can be available on I/O pin PB7 when register bit BZ1EN (BZ1CR1[7]) is set to 1. Moreover, PB7 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.

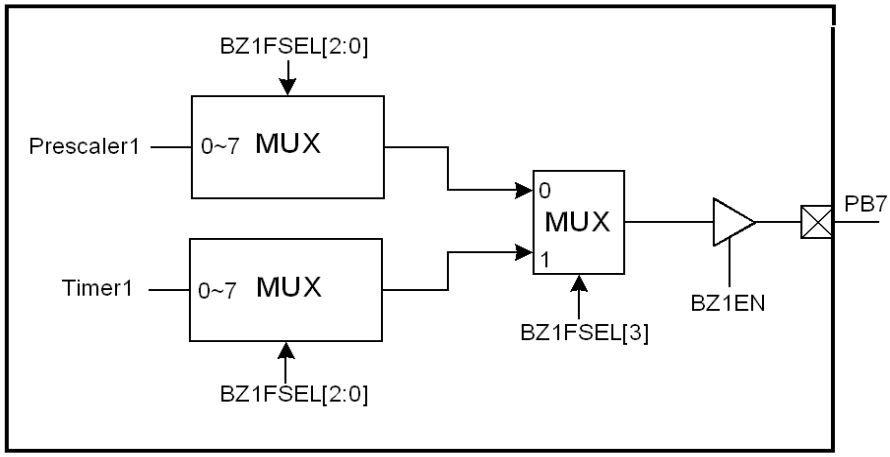


Figure 19 Buzzer1 Block Diagram

3.8 Timer2 / PWM2 / Buzzer2

Timer2 is an 8-bit down-count timer with Prescaler2 whose dividing rate is programmable. The output of Timer2 can be used to generate PWM2 output and Buzzer2 output. A write to the Timer2 will both write to a timer2 reload register (T2rld) and timer2 counter. A read to the timer2 will show the content of the Timer2 current count value.

The block diagram of Timer2 is shown in the figure below.

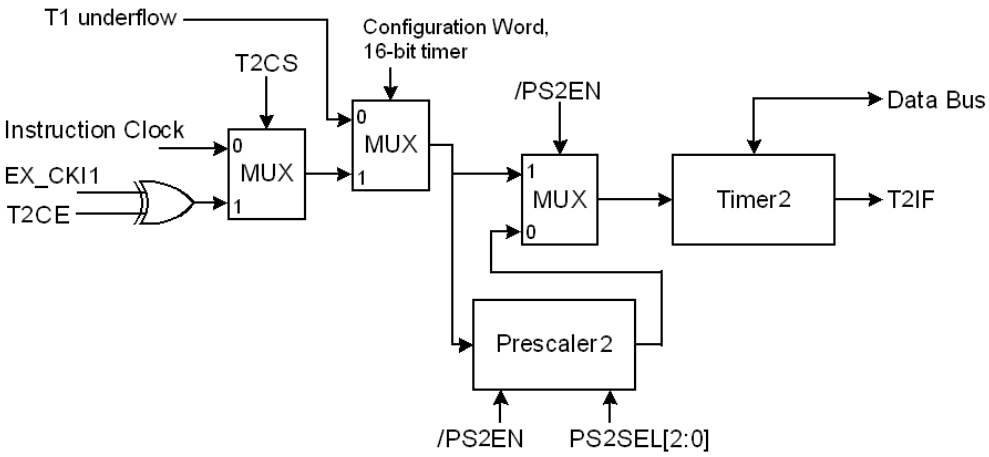


Figure 20 Block Diagram of Timer2

The operation of Timer2 can be enabled or disabled by register bit T2EN (T2CR1[0]). After Timer2 is enabled, its clock source can be instruction clock or pin EX_CKI1 which is determined by register bit T2CS (T2CR2[5]). When T2CS is 1, EX_CKI1 is selected as clock source. When T2CS is 0, instruction clock is selected as clock source. When EX_CKI1 is selected, the active edge to decrease Timer2 is determined by register bit T2CE (T2CR2[4]). When T2CE is 1, high-to-low transition on EX_CKI1 will decrease Timer2. When T2CE is 0, low-to-high transition on EX_CKI1 will decrease Timer2.

There is 16-bit timer option which connects timer1 and timer2 in series. In this mode the clock source of timer2 is the underflow output of timer1. More details please see the following sections.

The selected clock source can be divided further by Prescaler2 before it is applied to Timer2. Prescaler2 is enabled by writing 0 to register bit /PS2EN (T2CR2[3]) and the dividing rate is from 1:2 to 1:256 determined by register bits PS2SEL[2:0] (T2CR2[2:0]). Current value of Prescaler2 can be obtained by reading register PS2CV.

Timer2 provides two kinds of operating mode: one is One-Shot mode and the other is Non-Stop mode. When register bit T2OS (T2CR1[2]) is 1, One-Shot mode is selected. Timer2 will count down once from initial value stored on register TMR2 to 0x00, i.e. underflow is occurred. When register bit T2OS (T2CR1[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 T2RL (T2CR1[1]). When T2RL is 1, the initial value stored on register TMR2 will be restored and start next down-count from this initial value. When T2RL is 0, Timer2 will start next down-count from 0xFF.

When Timer2 is underflow, the register bit T2IF (INTF[5]) will be set to 1 to indicate Timer2 underflow event is occurred. If register bit T2IE (INTE[5]) and GIE are both set to 1, interrupt request will occur and interrupt service routine will be executed. T2IF will not be clear until firmware writes 0 to T2IF.

The timing chart of Timer2 is shown in the following figure.

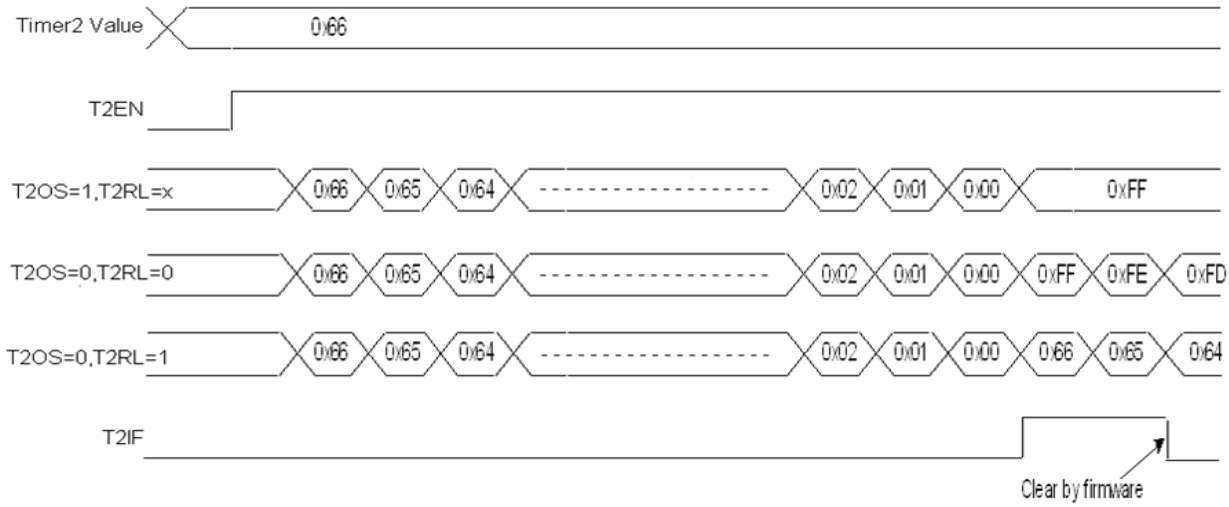


Figure 21 Timer2 Timing Chart

The PWM2 output can be available on I/O pin PB4 when register bit PWM2OEN (T2CR1[7]) is set to 1. Moreover, PB4 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 PWM2DUTY. When PWM2DUTY is 0, PWM2 output will be never active. When PWM2DUTY is 0xFF, PWM2 output will be active for 255 Timer2 input clocks. The frame rate is determined by TMR2 initial value. Therefore, PWM2DUTY value must be less than or equal to TMR2. The block diagram of PWM2 is illustrated in the following figure.

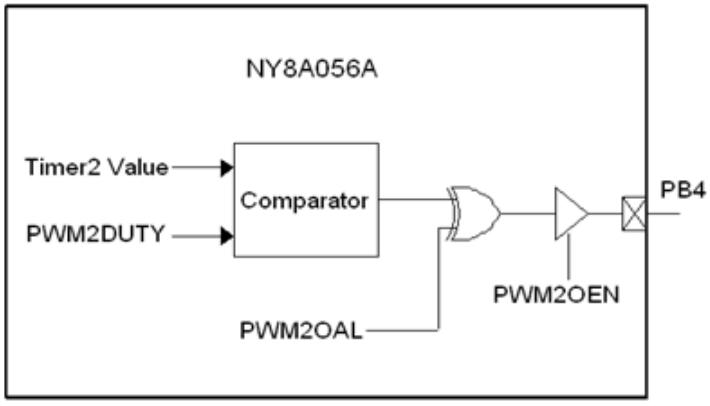


Figure 22 PWM2 Block Diagram

The Buzzer2 output (BZ2) can be available on I/O pin PB5 when register bit BZ2EN (BZ2CR1[7]) is set to 1. Moreover, PB5 will become output pin automatically. The frequency of BZ2 can be derived from Timer2 output or Prescaler2 output and dividing rate is determined by register bits BZ2FSEL[3:0] (BZ2CR[3:0]). When BZ2FSEL[3] is 0, Prescaler2 output is selected to generate BZ2 output. When BZ2FSEL[3] is 1, Timer2 output is selected to generate BZ2 output. The dividing rate can be from 1:2 to 1:256 in order to generate all kinds of frequency. The block diagram of Buzzer2 is illustrated in the following figure.

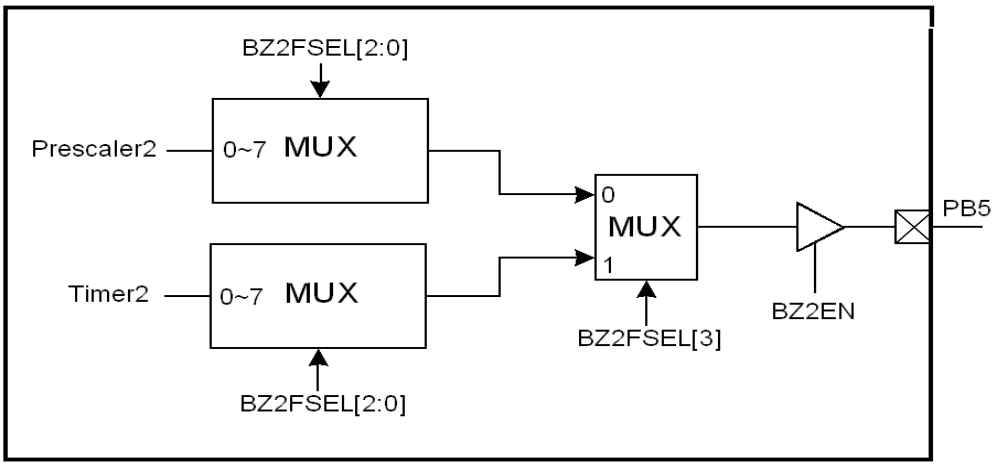


Figure 23 Buzzer2 Block Diagram

3.9 16 bit Timer Mode

Timer1 and Timer2 can be combined in series and become a 16-bit timer when configuration word 16-bit timer is set to 0 (low active). When in 16-bit timer mode, the timer1 clock source can be instruction clock or external clock EX_CKI0 as usual, but the clock source of timer2 is replaced by timer1 underflow output, as the block diagram shown below.

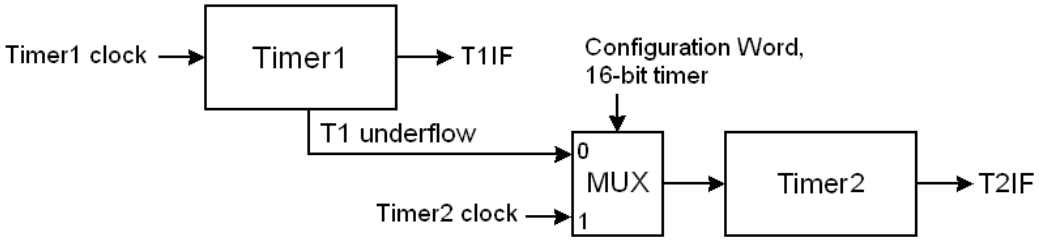


Figure 24 16-bit timer Block Diagram

In 16-bit timer mode, the PWM, buzzer, prescaler and interrupts function is still working for timer1 and timer2 respectively, the only difference is the clock source of timer2 is from timer1 underflow.

3.10 IR Carrier

NY8A056A provides two kinds of IR carrier according to its sink capability. One is Normal IR carrier whose sink current depends on how I/O pin (PB1) is configured. The other is Large IR carrier whose sink current is 340mA. This feature is determined by a configuration word. When Normal IR is selected, IR carrier will be preset on I/O pin PB1 with sink current 60mA. When Large IR is selected, IR carrier will be preset on I/O pin PB1 with sink current 340mA.

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. Register bit IROSC358M (IRCR[7]) is used to provide NY8A056A this information. When IROSC358M is 1, frequency of external crystal is 3.58MHz and when IROSC358M is 0, frequency of external crystal is 455KHz. 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.

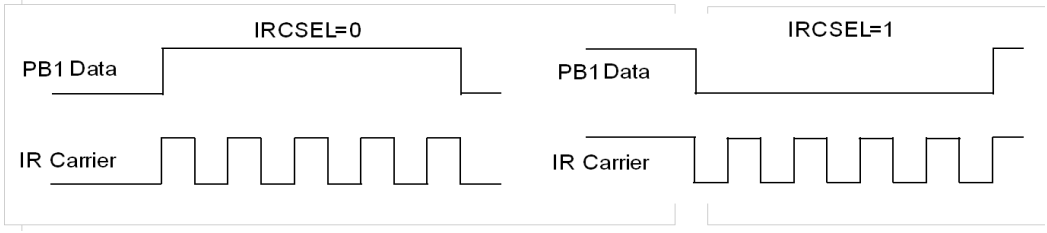


Figure 25 Polarity of IR Carrier vs. Output Data

3.11 Voltage Comparator

NY8A056A provides 1 set of voltage comparator and internal reference voltage with various analog comparing mode. The comparator non-inverting and inverting input can share with GPIO. The internal reference voltage can only routed to inverting input of comparator.

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.

NY8A056A comparator has two operating mode, which is P2V mode and P2P mode. These two modes are determined by VS[3:0] (register CMPCR[3:0]). When VS[3:0]=0, the comparator is in P2P mode. When VS[3:0]=1~15, it is in P2V mode. The pads used in the comparator are set as analog pads in the configuration words Analog Pin Select.

P2V mode has the function of comparing voltage between a designated analog pad and a designated reference. The structure of P2V mode is shown in the following figure:

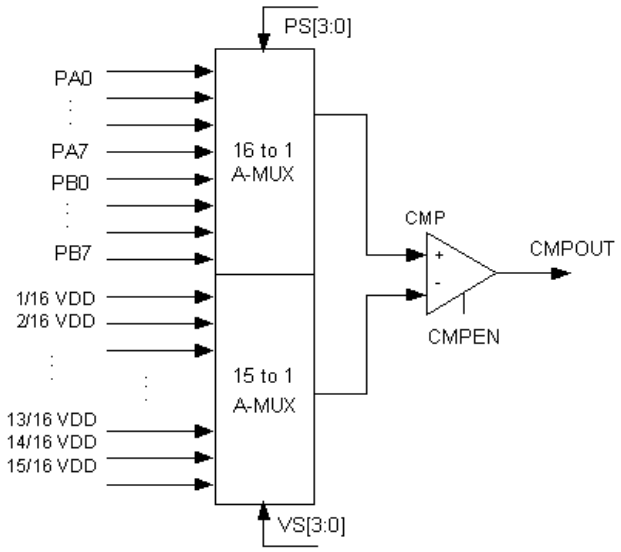


Figure 27 Comparator P2V mode block diagram

In P2V mode, the inverting input of the comparator is determined by VS[3:0]. VS[3:0] is used to select one out of 15 reference voltages, which is 1/16 V_{DD} to 15/16 V_{DD} as the table shown below.

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

Table 19 P2V mode reference voltage select

In P2V mode, the non-inverting input of the comparator is determined by PS[3:0] (register CMPCR[7:4]). PS[3:0] select one out of 15 pads PA7~6, PA4~0 and PB7~0 as the non-inverting input of the comparator. The table is shown below.

PS[3:0]	Selected pad
0000	PA0
0001	PA1
0010	PA2
0011	PA3
0100	PA4
0101	NC
0110	PA6
0111	PA7
1000	PB0
1001	PB1
1010	PB2
1011	PB3
1100	PB4
1101	PB5
1110	PB6
1111	PB7

Table 20 P2V mode pad select

The P2P mode has the function of comparing voltage between two analog pads. In this mode VS[3:0]=0, PS[3:0] select 2 out of 8 analog pads to be the non-inverting and inverting input of the comparator. The selection table is as the below.

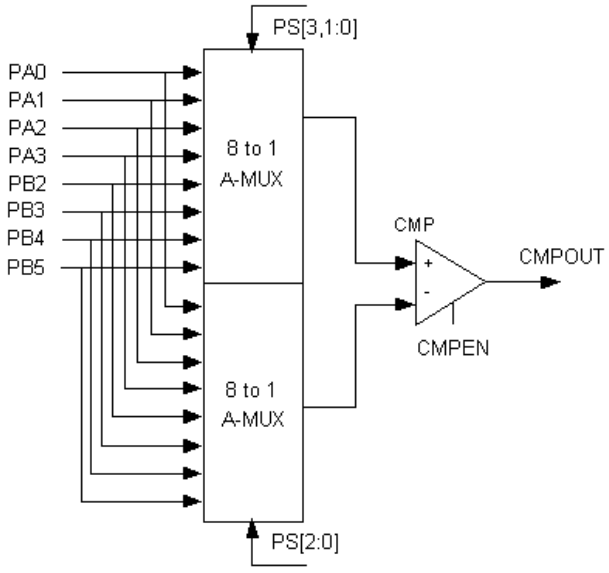


Figure 28 Comparator P2P mode block diagram

PS[3:0]	Non-inverting input	Inverting input

PS[3:0]	Non-inverting input	Inverting input
0000	PA0	PA1
0001	PA1	PA0
0010	PA2	PA3
0011	PA3	PA2
0100	PA0	PB3
0101	PA1	PB2
0110	PA2	PB5
0111	PA3	PB4
1000	PB2	PA1
1001	PB3	PA0
1010	PB4	PA3
1011	PB5	PA2
1100	PB2	PB3
1101	PB3	PB2
1110	PB4	PB5
1111	PB5	PB4

Table 21 P2P mode pads select

There are 3 ways to get the comparator output result: One is through interrupt mechanism, one is through register polling, another is through probing output pad.

To use comparator interrupt function, set CMPEN=1 and CMPIE=1, then read register OSCCR which will end the mismatch condition of comparator output and registered comparator output, then clear interrupt flag CMPIF. When comparator output change state, the CMPIF will be set to 1, thus entering interrupt service routine.

Comparator output can be polled by CMPOUT (register OSCCR[7]).

To probe comparator output at output pad, set CMPOE (register OSCCR[6]) to 1, then PB6 will be the real-time state of the comparator output. It is noted that when CMPOE=1, the PWM1 function will be disabled if it is enabled.

3.12 Watch-Dog Timer (WDT)

There is an on-chip free-running oscillator in NY8A056A 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 NY8A056A 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 NY8A056A and dividing rate is from 1:2 to 1:256 if WDT time-out will interrupt NY8A056A.

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.

3.13 Interrupt

NY8A056A 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 7 hardware interrupts:

- Timer0 overflow interrupt.
- Timer1 underflow interrupt.
- Timer2 underflow interrupt.
- WDT timeout interrupt.
- PB input change interrupt.
- External interrupt.
- Comparator output status change 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 NY8A056A 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 Register INTF 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 register INTF. Note that only when the corresponding bit of Interrupt Enable register INTE is set to 1, will the corresponding interrupt flag be read. And if the corresponding bit of Interrupt Enable Register INTE 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 NY8A056A 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 NY8A056A 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.

3.13.1 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.

3.13.2 Timer1 Underflow Interrupt

Timer1 underflow (from 0xFF to 0x00) will set register bit T1IF. This interrupt request will be serviced if T1IE and GIE are set to 1.

3.13.3 Timer2 Underflow Interrupt

Timer2 underflow (from 0xFF to 0x00) will set register bit T2IF. This interrupt request will be serviced if T2IE and GIE are set to 1.

3.13.4 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.

3.13.5 PB Input Change Interrupt

When PBx, $0 \le x \le 7$, is configured as input pin and corresponding register bit WUPBx is set to 1, a level change on these selected I/O pin(s) will set register bit PBIF. This interrupt request will be serviced if PBIE and GIE are set to 1. Note when PB0 is both set as level change interrupt and external interrupt, the external interrupt enable EIS=1 will disable PB0 level change operation.

3.13.6 External Interrupt

According to the configuration of EIS=1 and INTEDG, the selected active edge on I/O pin PB0 will set register bit INTIF and this interrupt request will be served if INTIE and GIE are set to 1.

3.13.7 Comparator Output Status Change Interrupt

The comparator interrupt is triggered whenever a change occurs on the comparator output status. This interrupt request will be serviced if CMPIE and GIE are set to 1. Note that before the comparator interrupt could happen, reading register OSCCR is needed to clear the previous comparator output status difference.

3.14 Oscillation Configuration

Because NY8A056A 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 oscillators which could be used as F_{HOSC} are internal high RC oscillator (I_HRC), external high crystal oscillator (E_HXT) and external crystal oscillator (E_XT). The oscillators which could be used as F_{LOSC} are internal low RC oscillator (I_LRC) and external low crystal oscillator (E_LXT).

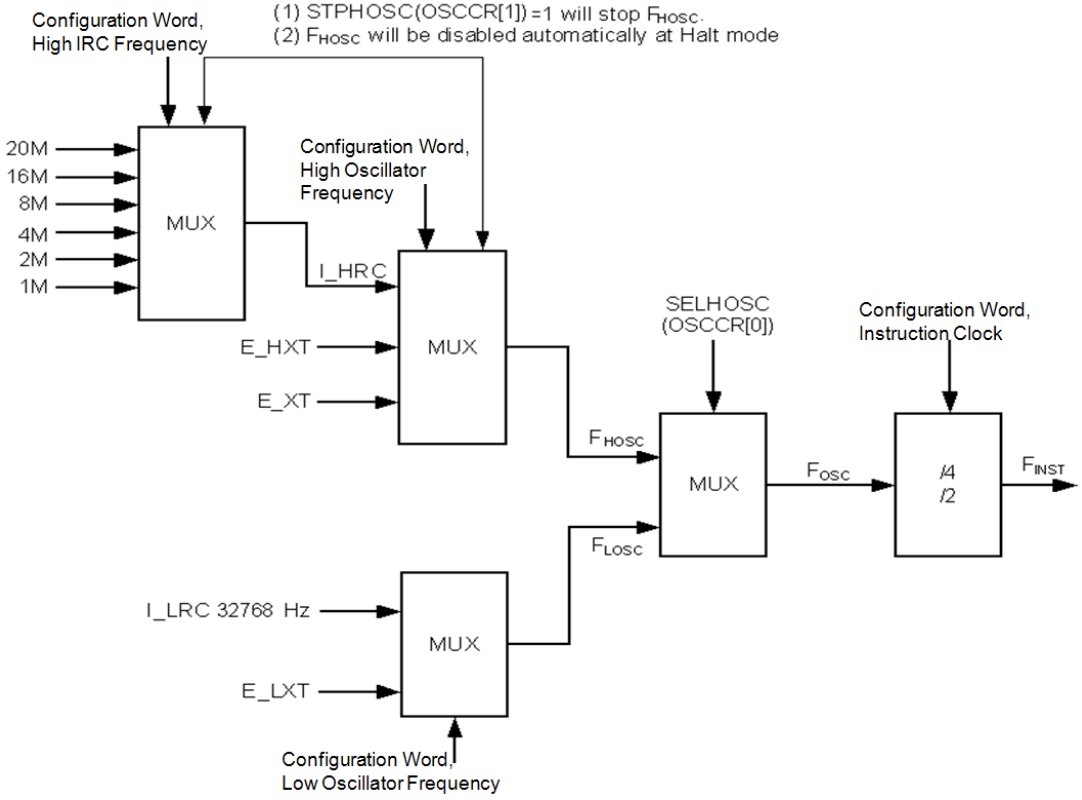


Figure 29 Oscillation Configuration of NY8A056A

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. Moreover, external crystal oscillator pads PA6 and PA7 can be used as I/O pins. On the other hand, PA7 can be the output pin of instruction clock according to a configuration word's setting. If F_{HOSC} required external crystal whose frequency ranges from 8MHz to 20MHz, E_HXT is recommended. If F_{HOSC} required external crystal whose frequency ranges from 455KHz to 6MHz, E_XT is recommended. When E_HXT or E_XT is adopted, PA6/PA7 can not be used as I/O pins. They must be used as crystal output pin and input pin. PA7 is crystal output pin (Xout) and PA6 is crystal input pin (Xin).

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. If F_{LOSC} required external crystal, E_LXT is selected and only 32768Hz crystal is allowed. When E_LXT is adopted, PA6/PA7 cannot be used as I/O pins. They must be used as crystal

output pin and input pin. PA7 is crystal output pin (Xout) and PA6 is crystal input pin (Xin). The dual-clock combinations of F_{HOSC} and F_{LOSC} are listed below.

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

Table 22 Dual-clock combinations

When E_HXT, E_XT or E_LXT is used as one of oscillations, the crystal or resonator is connected to Xin and Xout to provide oscillation. Moreover, a resistor and two capacitors are recommended to connect as following figure in order to provide reliable oscillation, refer to the specification of crystal or resonator to adopt appropriate C1 or C2 value. The recommended value of C1 and C2 are listed in the table below.

Oscillation Mode	Crystal Frequency (Hz)	C1, C2 (pF)
E_HXT	16M	5 ~ 10
	10M	5 ~ 30
	8M	5 ~ 20
E_XT	4M	5 ~ 30
	1M	5 ~ 30
	455K	10 ~ 100
E_LXT	32768	5 ~ 30

Table 23 Recommended C1 and C2 Value for Different Kinds of Crystal Oscillation

For 20MHZ resonator in 2 clock CPU cycle mode, an 18pF C2 capacitor is a must.

Moreover, for precision 32.768k crystal, it is recommended to set C1=C2=15pF capacitor.

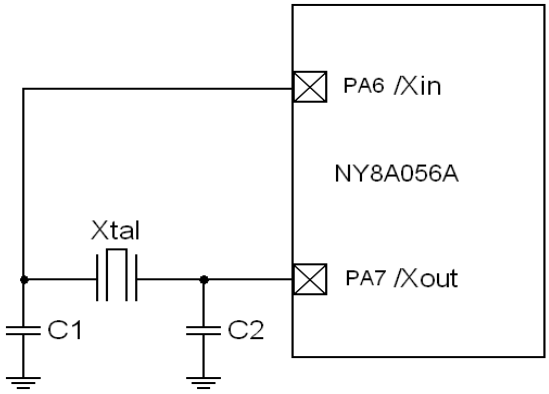


Figure 30 Connection for External Crystal Oscillation

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.

3.15 Operating Mode

NY8A056A 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, NY8A056A will stop almost all operations except Timer0/Timer1/Timer2 /WDT in order to wake-up periodically. At Halt mode, NY8A056A 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.

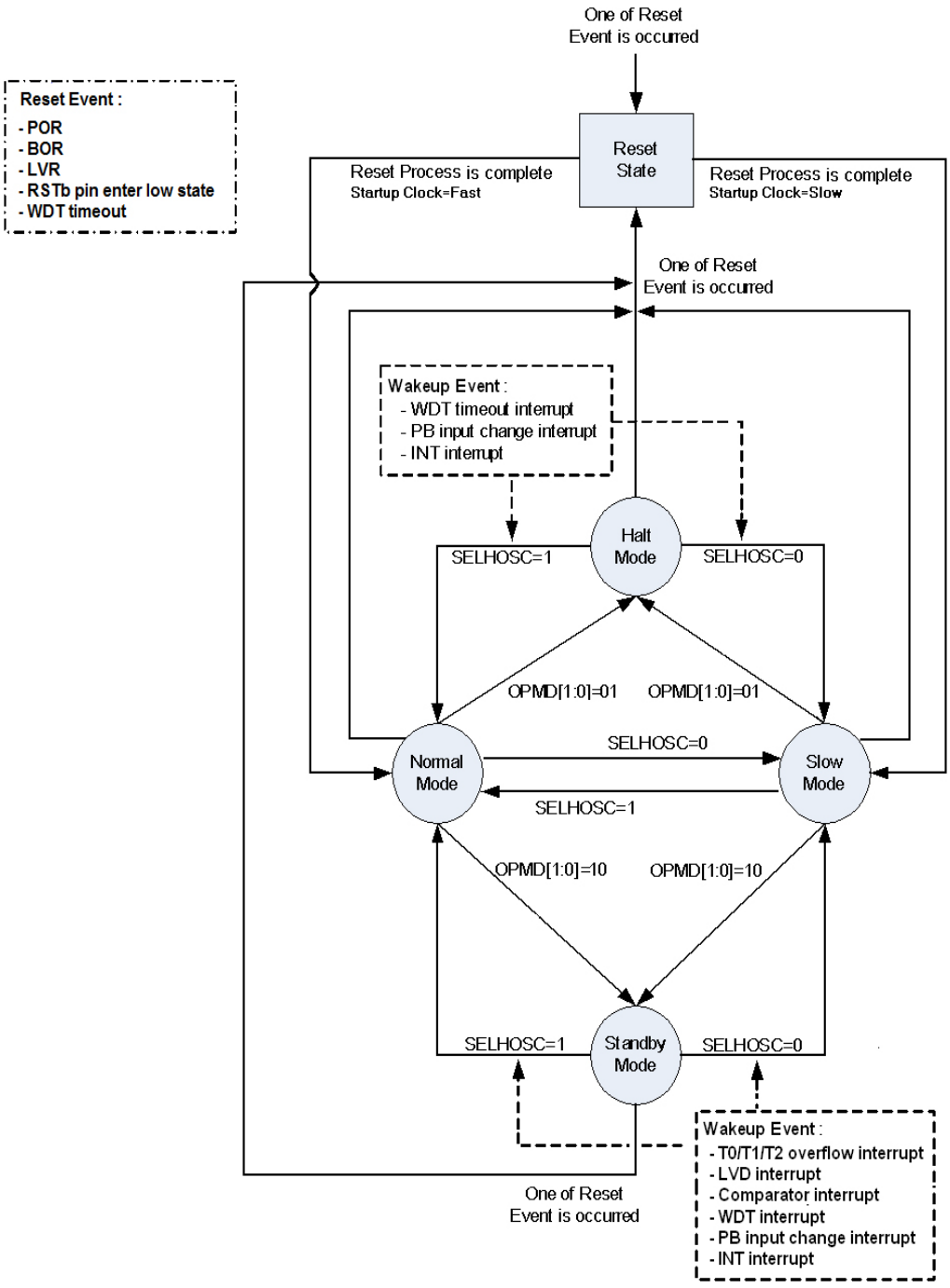


Figure 31 Four Operating Modes

3.15.1 Normal Mode

After any Reset Event is occurred and Reset Process is completed, NY8A056A 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, NY8A056A will enter Normal mode, if Startup Clock=Slow, NY8A056A 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, NY8A056A 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 NY8A056A 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.

3.15.2 Slow Mode

NY8A056A 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 NY8A056A. Therefore user can write 0 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 FLOSC and all peripheral modules may be active according to corresponding module enable bit.
- FHOSC 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.

3.15.3 Standby Mode

NY8A056A 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 NY8A056A and user has to enter slow mode and write 1 to register bit STPHOSC in order to stop F_{HOSC} oscillation. Most of NY8A056A peripheral modules are disabled but Timer can be still active if register bit T0EN/T1EN/T2EN is set to 1. Therefore NY8A056A can wake-up after Timer0/Timer1/Timer2 is expired. The expiration period is determined by the register TMR0/TMR1/TMR2, F_{INST} and other configurations for Timer0/Timer1/Timer2.

- Instruction execution is stop and some peripheral modules may be active according to corresponding module enable bit.
- FHOSC can be disabled by writing 1 to register bit STPHOSC.
- The FLOSC is still active and running.
- IC can wake-up from Standby mode if any of (a) Timer0/Timer1/Timer2 (overflow/underflow) interrupt, (b) WDT timeout interrupt, (c) 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.

3.15.4 Halt Mode

NY8A056A 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 NY8A056A can only wake-up by some specific events. Therefore, Halt mode is the most power saving mode provided by NY8A056A.

- Instruction execution is stop and all peripheral modules are disabled.
- FHOSC and FLOSC are both disabled automatically.
- IC can wake-up from Halt mode if any of (a) WDT timeout interrupt, (b) 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.

3.15.5 Wake-up Stable Time

The wake-up stable time of Halt mode is determined by Configuration word: High Oscillator Frequency or Low Oscillator Frequency. If one of E_HXT, E_XT and E_LXT is selected, the wake-up period would be 512*F_{OSC}. And if no XT mode are selected, 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 NY8A056A enter Standby mode or Halt mode, user may execute instruction ENI. At this condition, NY8A056A 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.

3.15.6 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
Timer0/1/2	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	-	-	- Timer0/1/2 overflow - WDT timeout - PB input change - INT - LVD interrupt - Comparator interrupt	- WDT timeout - PB input change - INT

Table 24 Summary of Operating Modes

3.16 Reset Process

NY8A056A 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

Event	/TO	/PD
WDT reset from non-Halt mode	0	1
WDT reset from Halt mode	0	0
SLEEP executed	1	0
CLRWDT executed	1	1

Table 25 Summary of /TO & /PD Value and its Associated Event

After Reset Event is released, NY8A056A 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 oscillator is stable, NY8A056A will wait further 16 clock cycles of F_{HOSC} (oscillator start-up time, OST) and Reset Process is complete.

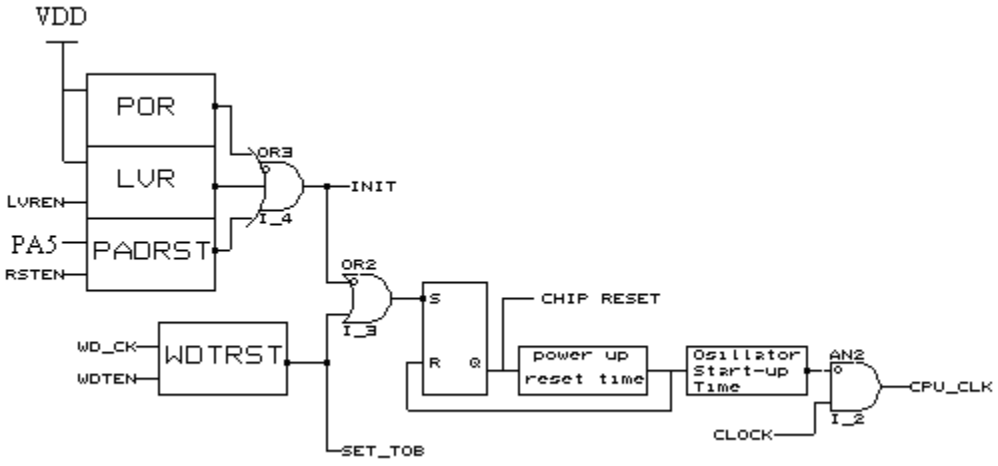


Figure 32 Block diagram of on-chip reset circuit

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.

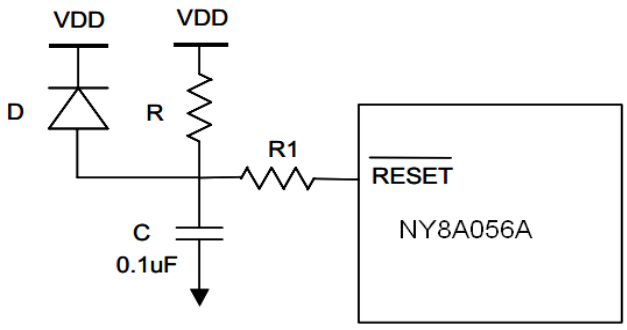


Figure 33 Block Diagram of Reset Application

4. Instruction Set

NY8A056A 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	-
CALL	adr		Call subroutine	2	-
GOTO	adr		unconditional branch	2	-
LCALL	adr		Call subroutine	2	-
LGOTO	adr		unconditional branch	2	-

Table 26 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 0x00 ~0x3F.

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

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 ≤ 63 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 ≤63 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 ≤ 63. 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 ≤ 63 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 ≤ 63 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 ≤ 63 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.

BTRSS	Test Bit in R and Skip if Set
Syntax:	BTRSS R, bit
Operand:	0 ≤ R ≤ 63 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.

CALL	Call Subroutine
Syntax:	CALL adr
Operand:	0 ≤ adr < 255
Operation:	PC + 1 → Top of Stack {PCHBUF, adr} → PC
Status affected:	--
Description:	The return address (PC + 1) is pushed onto top of Stack. The 8-bit immediate address adr is loaded into PC[7:0] and PCHBUF[1:0] is loaded into PC[9:8].
Cycle:	2
Example:	CALL SUB before executing instruction: PC=A0. Stack pointer=1 after executing instruction: PC=address of SUB, Stack[1] = A0+1, Stack pointer=2.

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[1:0] is loaded into PC[9:8] 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

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.

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

CLRR	Clear R
Syntax:	CLRR R
Operand:	0 ≤ R ≤ 63
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 ≤ 63 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.

CMPAR	Compare ACC and R
Syntax:	CMPAR R
Operand:	0 ≤ R ≤ 63
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.

DECR	Decrease R
Syntax:	DECR R, d
Operand:	0 ≤ R ≤ 63 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.

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 ≤ 63 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.

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.

GOTO	Unconditional Branch
Syntax:	GOTO adr
Operand:	0 ≤ adr < 511
Operation:	{PCHBUF, adr} → PC
Status affected:	--
Description:	GOTO is an unconditional branch instruction. The 9-bit immediate address adr is loaded into PC[8:0] and PCHBUF[1] is loaded into PC[9].
Cycle:	2
Example:	GOTO Level before executing instruction: PC=A0. after executing instruction: PC=address of Level.

GOTOA	Unconditional Branch
Syntax:	GOTOA
Operand:	--
Operation:	{TBHP, ACC} → PC
Status affected:	--
Description:	GOTOA is an unconditional branch instruction. The content of TBHP[1:0] is loaded into PC[9:8] 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 ≤ 63 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.

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[9:0].
Cycle:	3
Example:	INT before executing instruction: PC=address of INT code after executing instruction: PC=0x01

INCRSZ	Increase R, Skip if 0
Syntax:	INCRSZ R, d
Operand:	0 ≤ R ≤ 63 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.

IORAR	OR ACC with R
Syntax:	IORAR R, d
Operand:	0 ≤ R ≤ 63 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.

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.

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.

IOST	Load F-page SFR from ACC
Syntax:	IOST F
Operand:	0 ≤ 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.

LCALL	Call Subroutine
Syntax:	LCALL adr
Operand:	0 ≤ adr ≤ 1023
Operation:	PC + 1 → Top of Stack, adr → PC[9:0]
Status affected:	--
Description:	The return address (PC + 1) is pushed onto top of Stack. The 10-bit immediate address adr is loaded into PC[9:0].
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.

LGOTO	Unconditional Branch
Syntax:	LGOTO adr
Operand:	0 ≤ adr ≤ 1023
Operation:	adr → PC[9:0].
Status affected:	--
Description:	LGOTO is an unconditional branch instruction. The 10-bit immediate address adr is loaded into PC[9:0].
Cycle:	2
Example:	LGOTO Level before executing instruction: PC=A0. after executing instruction: PC=address of Level.

MOVIA	Move Immediate to ACC
Syntax:	MOVIA i
Operand:	0 ≤ i < 255
Operation:	i → 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.

MOVAR	Move ACC to R
Syntax:	MOVAR R
Operand:	0 ≤ R ≤ 63
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.

MOVR	Move R to ACC or R
Syntax:	MOVR R, d
Operand:	0 ≤ R ≤ 63 d = 0, 1.
Operation:	R → 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.

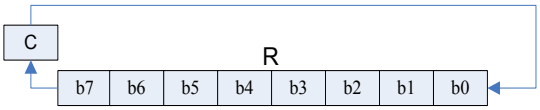
NOP	No Operation
Syntax:	NOP
Operand:	--
Operation:	No operation.
Status affected:	--
Description:	No operation.
Cycle:	1
Example:	NOP before executing instruction: PC=A0 after executing instruction: PC=A0+1

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

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.

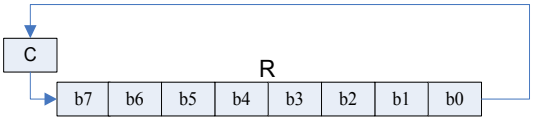
RET	Return from Subroutine
Syntax:	RET
Operand:	--
Operation:	Top of Stack $\rightarrow$ PC
Status affected:	--
Description:	PC is loaded from top of Stack as return address.
Cycle:	2
Example:	RET before executing instruction: Stack level=2. after executing instruction: PC=Stack[2], Stack level=1.

RLR	Rotate Left R Through Carry
Syntax:	RLR R, d
Operand:	0 ≤ R ≤ 63 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 ≤ 63 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 ≤ 63 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 ≤15
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 ≤ 15
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 ≤ 63 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:	SBCAR 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)

SWAPR	Swap High/Low Nibble in R
Syntax:	SWAPR R, d
Operand:	0 ≤ R ≤ 63 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.

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)

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.

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.

XORAR	Exclusive-OR ACC with R
Syntax:	XORAR R, d
Operand:	$0 \leq R \leq 63$ d = 0, 1.
Operation:	ACC ⊕ R → 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.

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.

XORIA	Exclusive-OR Immediate with ACC
Syntax:	XORIA i
Operand:	$0 \leq i < 255$
Operation:	ACC ⊕ i → 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. E_HXT 3. E_XT
2	Low Oscillator Frequency	1. I_LRC 2. E_LXT
3	High IRC Frequency	1. 1MHz 2. 2MHz 3. 4MHz 4. 8MHz 5. 16MHz 6. 20MHz
4	High Crystal Oscillator	1. 6MHz < F_{HOSC} ≤ 8MHz 2. 8MHz < F_{HOSC} ≤ 10MHz 3. 10MHz < F_{HOSC} ≤ 12MHz 4. 12MHz < F_{HOSC} ≤ 16MHz 5. 16MHz < F_{HOSC} ≤ 20MHz
5	Instruction Clock	1. 2 oscillator period 2. 4 oscillator period
6	WDT	1. Watchdog Enable (Software control) 2. Watchdog Disable (Always disable)
7	WDT Event	1. Watchdog Reset 2. Watchdog Interrupt
8	Timer0 Source	1. EX_CKI0 2. Low Oscillator
9	16-bit Timer	1. Disable 2. Enable
10	PA.5	1. PA.5 is I/O 2. PA.5 is reset
11	PA.7	1. PA.7 is I/O 2. PA.7 is instruction clock output
12	IR Current	1. Normal=60mA 2. Large=340mA
13	Startup Time	1. 140us 2. 4.5ms 3. 18ms 4. 72ms 5. 288ms
14	WDT Time Base	1. 3.5ms 2. 15ms 3. 60ms 4. 250ms
15	Noise Filter (High_EFT)	1. Enable 2. Disable
16	LVR Setting	1. Register Control 2. LVR Always On
17	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
18	V_{DD} Voltage	1. 3.0V 2. 4.5V 3. 5.0V
19	PA Pull-High Resistor	1. Weak 2. Strong
20	PB Pull-High Resistor	1. Weak 2. Strong
21	Sink current type	1. Normal 2. Large 3. Constant
22	Analog Input pin select	1. Enable 2. Disable
23	Read Output Data	1. I/O Port 2. Register
24	E_LXT Backup Control	1. Auto off 2. Register off
25	EX_CKI0 to Inst. Clock	1. Sync 2. Async
26	Startup Clock	1. Fast (I_HRC/E_HXT/E_XT) 2. Slow (I_LRC/E_LXT)

Table 27 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 @ E_HXT/2
			2.0				F_{INST}=16MHz @ E_HXT/4
			2.0				F_{INST}=8MHz @ I_LRC/4 & I_LRC/2
							F_{INST}=8MHz @ E_LXT/4 & E_LXT/2
			1.8				F_{INST}=4MHz @ I_LRC/4 & I_LRC/2
							F_{INST}=4MHz @ E_LXT/4 & E_LXT/2
			1.8				F_{INST}=32KHz @ I_LRC/4 & I_LRC/2
							F_{INST}=32KHz @ E_LXT/4 & E_LXT/2
V_{IH}	Input high voltage	5V	4.0	--	--	V	RSTb, EX_CKI, INT ($0.8\ V_{DD}$)
		3V	2.4	--	--		
		5V	3.5	--	--	V	All other I/O pins ($0.7\ V_{DD}$)
		3V	2.1	--	--		
V_{IL}	Input low voltage	5V	--	--	1.0	V	RSTb, EX_CKI, INT ($0.2\ V_{DD}$)
		3V	--	--	0.6		
		5V	--	--	1.5	V	All other I/O pins ($0.3\ V_{DD}$)
		3V	--	--	0.9		
I_{OH}	Output high current	5V		-36		mA	V_{OH}=4.0V
		3V		-20			V_{OH}=2.0V
I_{OL}	Output low current (Large current)	5V		90		mA	V_{OL}=1.0V
		3V		60			
I_{OL}	Output low current (Constant current)	5V		20		mA	V_{OL}=1.0V
		3V		19			
I_{LIR}	IR sink current	5V		420		mA	V_{OL}=1.0V, LIR
		3V		340			
I_{OP}	Operating current	**Normal Mode**					
		5V		2.82		mA	F_{HOSC}=20MHz @ I_HRC/2 & E_HXT/2
		3V		1.41			
		5V		2.14		mA	F_{HOSC}=20MHz @ I_HRC/4 & E_HXT/4
		3V		0.98			

Symbol	Parameter	V_{DD}	Min.	Typ.	Max.	Unit	Condition
		5V		2.76		mA	F_{HOSC}=16MHz @ I_HRC/2 & E_HXT/2
		3V		1.27			
		5V		2.10		mA	F_{HOSC}=16MHz @ I_HRC/4 & E_HXT/4
		3V		0.95			
		5V		1.87		mA	F_{HOSC}=8MHz @ I_HRC/2 & E_HXT/2
		3V		0.86			
		5V		1.54		mA	F_{HOSC}=8MHz @ I_HRC/4 & E_HXT/4
		3V		0.68			
		5V		1.21		mA	F_{HOSC}=4MHz @ I_HRC/2 & E_HXT/2
		3V		0.55			
		5V		1.04		mA	F_{HOSC}=4MHz @ I_HRC/4 & E_HXT/4
		3V		0.45			
		5V		0.93		mA	F_{HOSC}=1MHz @ I_HRC/2 & E_HXT/2
		3V		0.40			
		5V		0.88		mA	F_{HOSC}=1MHz @ I_HRC/4 & E_HXT/4
		3V		0.37			
		Slow Mode					
		5V		8.4		uA	F_{HOSC} disabled, F_{LOSC}=32KHz @ I_LRC/2
		3V		3.5			
		5V		8.9		uA	F_{HOSC} disabled, F_{LOSC}=32KHz @ E_LXT/2.
		3V		3.5			
		5V		6.4		uA	F_{HOSC} disabled, F_{LOSC}=32KHz @ I_LRC/4
		3V		2.5			
		5V		7.0		uA	F_{HOSC} disabled, F_{LOSC}=32KHz @ E_LXT/4.
		3V		2.5			
I_{STB}	Standby current	5V		5		uA	Standby mode, F_{HOSC} disabled, F_{LOSC}=32KHz @ I_LRC/4
		3V		2			
I_{HALT}	Halt current	5V			0.5	uA	Halt mode, WDT disabled.
		3V			0.2		
		5V			5.0	uA	Halt mode, WDT enabled.
		3V			2.0		
R_{PH}	Pull-High resistor	5V		55		kΩ	Pull-High resistor
		3V		105			
R_{PL}	Pull-Low resistor	5V		55		kΩ	Pull-Low resistor
		3V		105			

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	%	

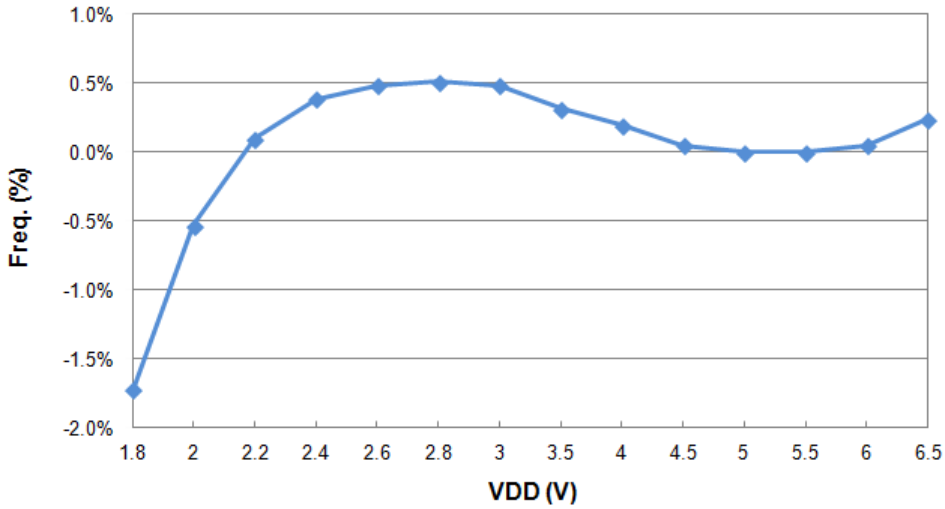
6.4 Comparator

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

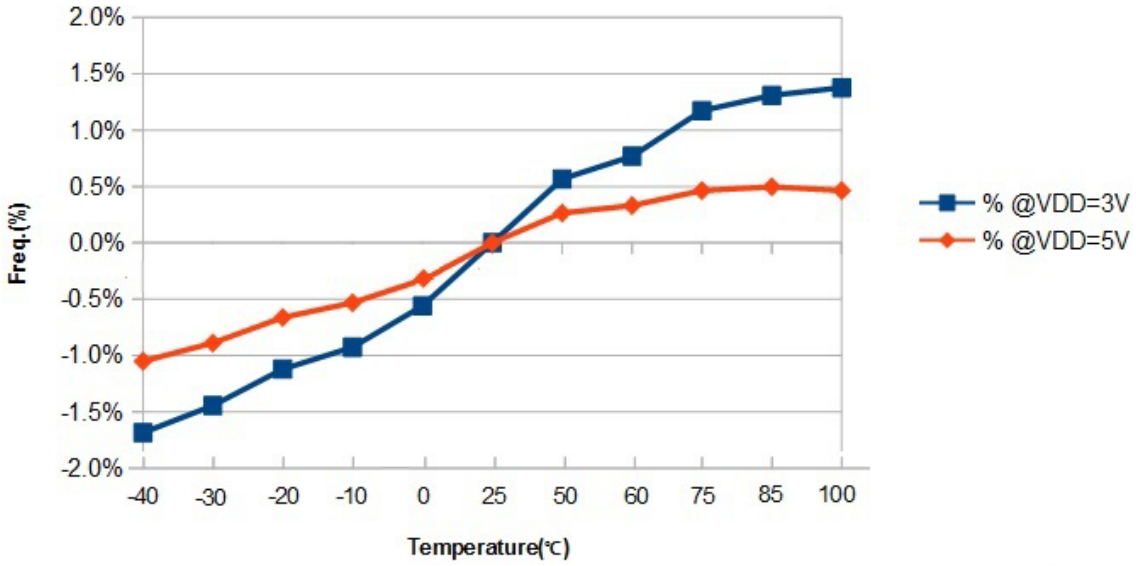
Symbol	Parameter	Min.	Typ.	Max.	Unit	Condition
V_{IVR}	Comparator input voltage range	0	-	5	V	F_{HOSC}=1MHz
T_{ENO}	Comparator enable to output valid	-	20	-	ms	F_{HOSC}=1MHz
I_{CO}	Operating current of comparator	-	250	-	uA	F_{HOSC}=1MHz, P2V mode

6.5 Characteristic Graph

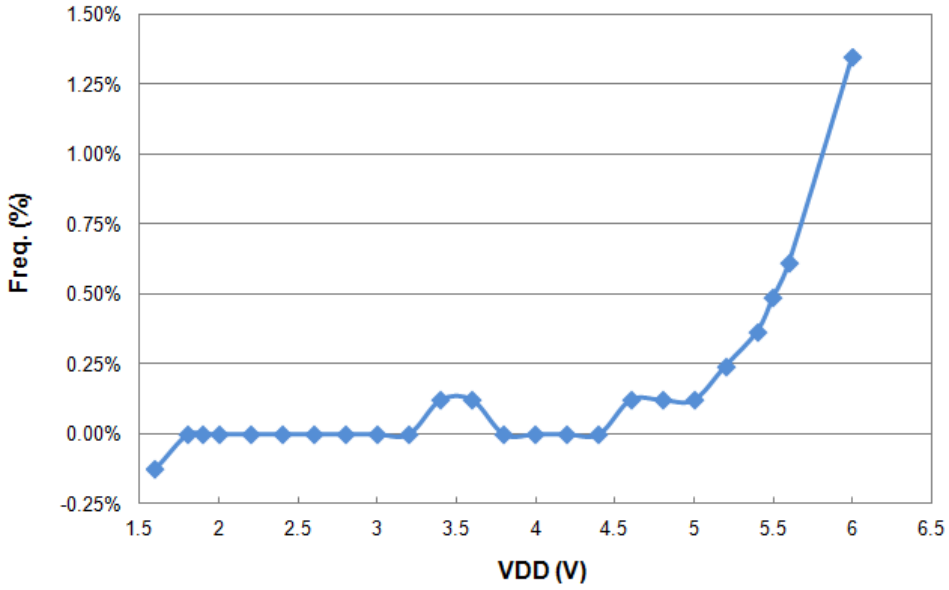
6.5.1 Frequency vs. V_{DD} of I_HRC



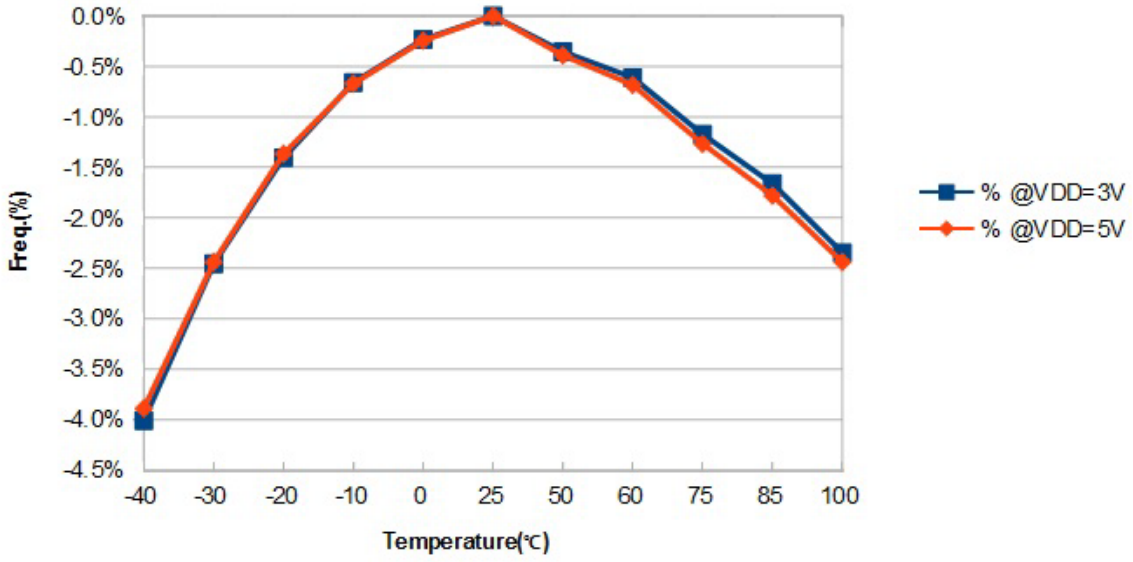
6.5.2 Frequency vs. Temperature of I_HRC



6.5.3 Frequency vs. V_{DD} of I_LRC



6.5.4 Frequency vs. Temperature of I_LRC

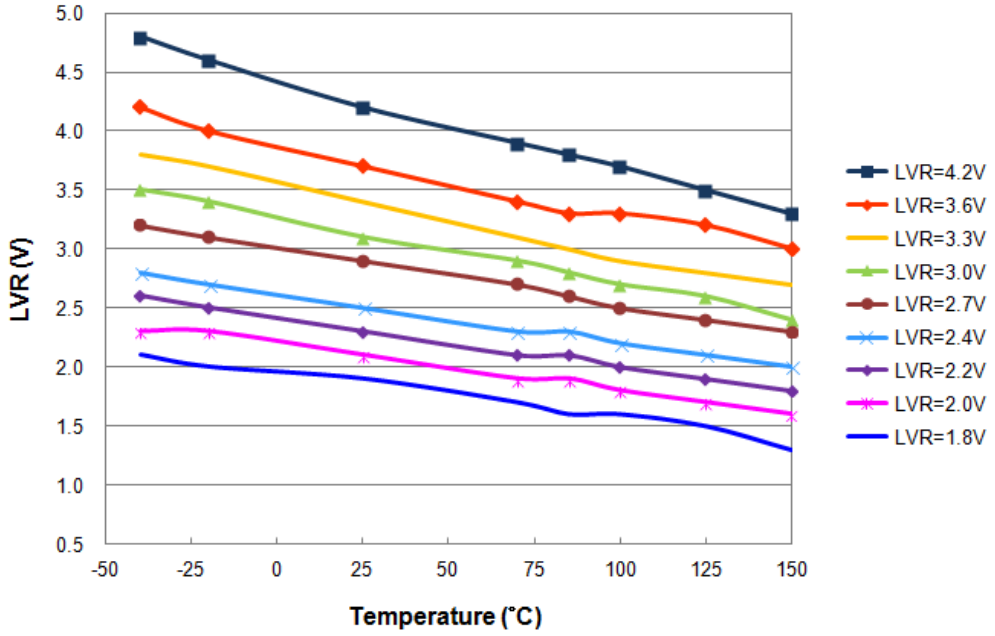


6.6 Recommended Operating Voltage

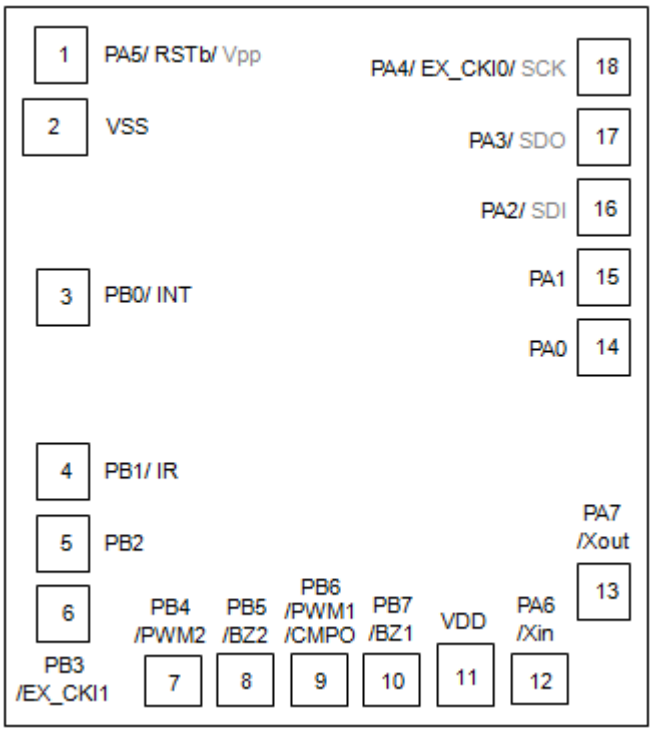
Recommended Operating Voltage (Temperature range: -40°C ~ +85°C)

Frequency	Min. Voltage	Max. Voltage	LVR: default (25°C)	LVR: Recommended (-40°C ~ +85°C)
20M/2T	3.3V	5.5V	3.6V	4.2V
16M/2T	3.0V	5.5V	3.0V	3.6V
20M/4T	2.2V	5.5V	2.4V	3.0V
16M/4T	2.0V	5.5V	2.2V	2.4V
8M(2T or 4T)	2.0V	5.5V	2.2V	2.4V
≦6M(2T or 4T)	1.8V	5.5V	1.8V	2.0V

6.7 LVR vs. Temperature

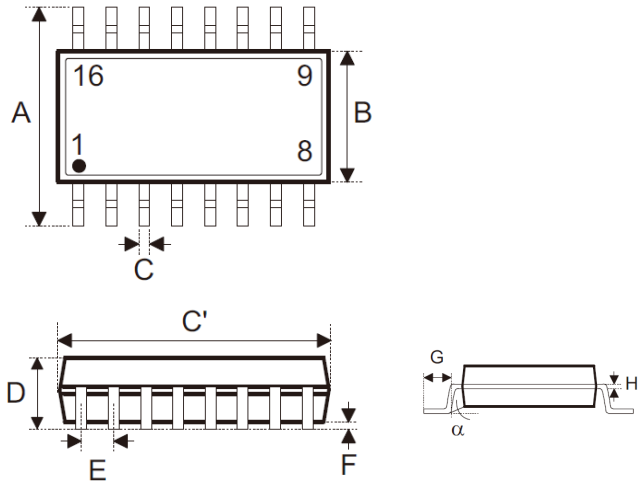


7. Die Pad Diagram



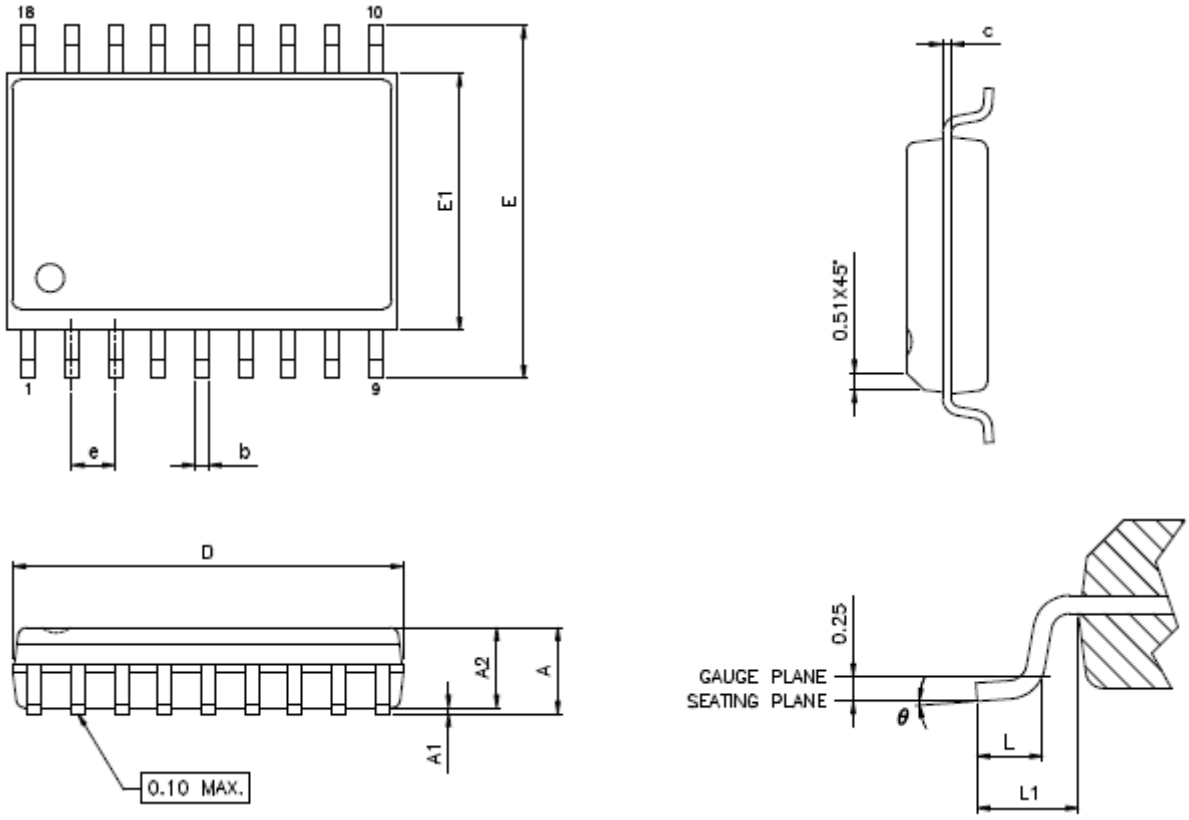
8. Package Dimension

8.1 16-Pin Plastic SOP (150 mil)



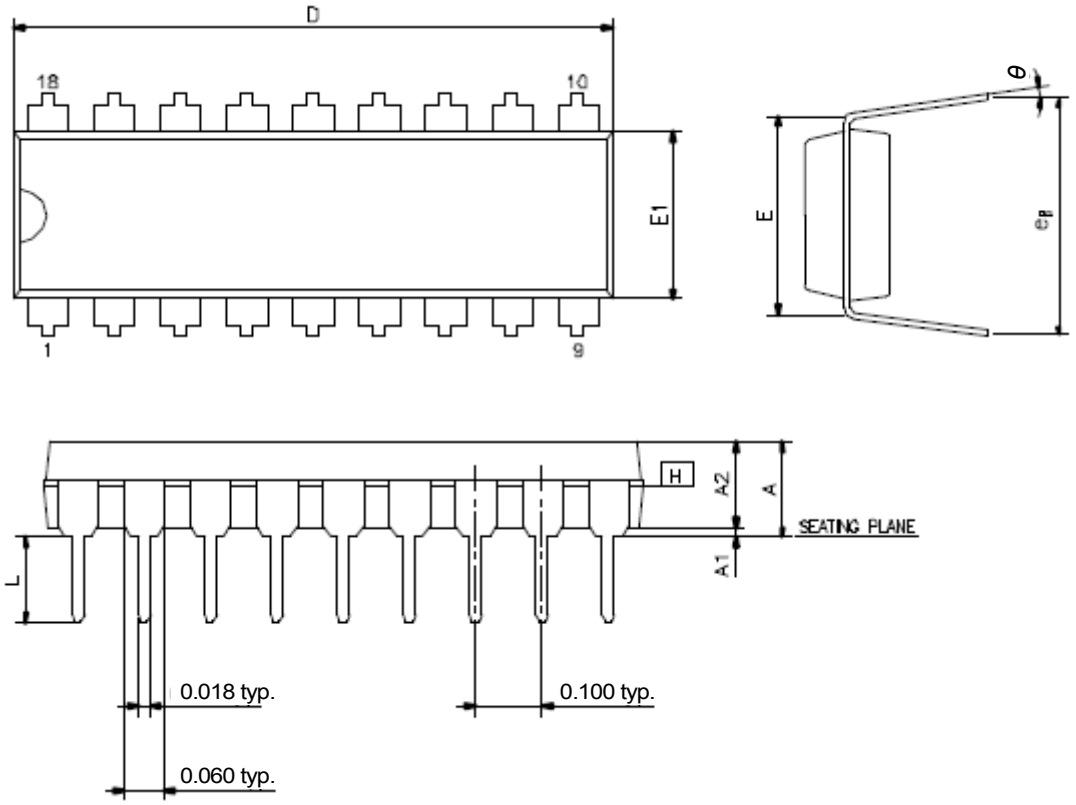
	INCHES			MILLIMETERS		
	MIN	TYP	MAX	MIN	TYP	MAX
A		0.236 BSC			6.00 BSC	
B		0.154 BSC			3.90 BSC	
C	0.012	-	0.020	0.31	-	0.51
C'		0.390 BSC			9.90 BSC	
D	0.065	-	0.069	1.64	-	1.75
E		0.050 BSC			1.27 BSC	
F	0.004	-	0.010	0.10	-	0.25
G	0.016	-	0.050	0.40	-	1.27
H	0.004	-	0.010	0.10	-	0.25
α	-	-	8°	-	-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

8.2 18-Pin Plastic SOP (300 mil)



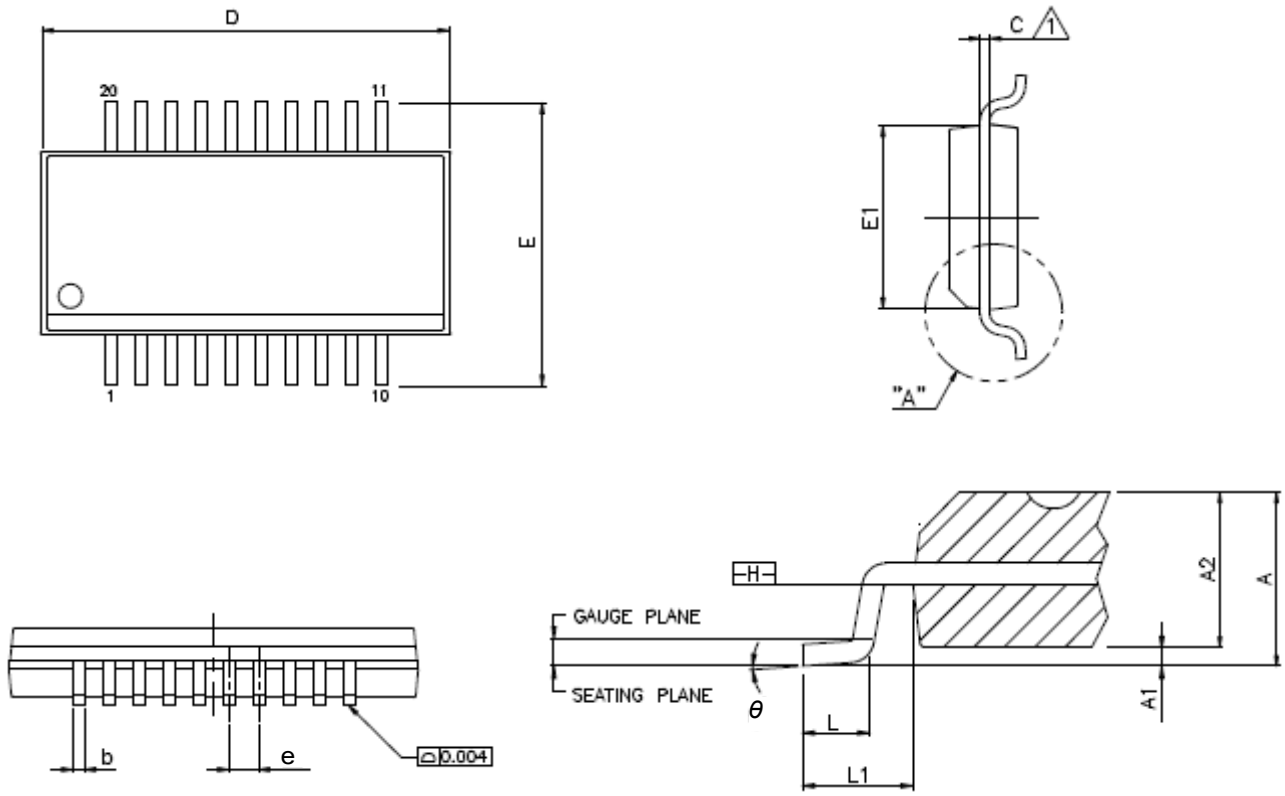
SYMBOLS	INCHES			MILLIMETERS		
	MIN	TYP	MAX	MIN	TYP	MAX
A	-	-	0.104	-	-	2.65
A1	0.004	-	0.012	0.10	-	0.30
A2	0.087	0.091	0.094	2.2	2.3	2.4
b	0.014	-	0.017	0.35	-	0.44
c	0.010	-	0.012	0.25	-	0.31
D	0.443	0.451	0.459	11.25	11.45	11.65
E	0.398	0.406	0.413	10.10	10.30	10.50
E1	0.287	0.295	0.303	7.30	7.50	7.70
e	0.050 BSC			1.27 BSC		
L	0.028	-	0.039	0.70	-	1.00
L1	0.055 REF			1.40 REF		
θ	0°	-	8°	0°	-	8°

8.3 18-Pin Plastic DIP (300 mil)



SYMBOLS	INCHES			MILLIMETERS		
	MIN	TYP	MAX	MIN	TYP	MAX
A	0.152	-	0.167	3.85	-	4.25
A1	0.026	-	0.033	0.65	-	0.85
A2	0.126	-	0.134	3.20	-	3.40
D	0.896	-	0.904	22.76	-	22.96
E	0.300	-	0.310	7.62	-	7.87
E1	0.246	-	0.254	6.25	-	6.45
L	0.126	-	0.134	3.20	-	3.40
e_B	0.319	-	0.346	8.10	-	8.80
θ	-	5°	-	-	5°	-

8.4 20-Pin Plastic 0.65SSOP (209 mil, Lead pitch 0.65mm)



SYMBOLS	INCHES			MILLIMETERS		
	MIN	TYP	MAX	MIN	TYP	MAX
A	0.065	-	0.073	1.65	-	1.85
A1	0.008	-	0.012	0.20	-	0.30
A2	0.057	-	0.061	1.45	-	1.55
b	-	0.012	-	-	0.30	-
c	-	0.006	-	-	0.15	-
D	0.281	0.283	0.285	7.15	7.20	7.25
E	0.301	0.307	0.313	7.65	7.80	7.95
E1	0.207	0.209	0.211	5.25	5.30	5.35
e	-	0.026	-	-	0.65	-
L	0.024	-	0.031	0.60	-	0.80
L1	0.047	-	0.051	1.20	-	1.30
θ	0°	4°	8°	0°	4°	8°

9. Ordering Information

P/N	*Package Type*	*Pin Count*	*Package Width*	*Shipping*
NY8A056A	Die	--	--	--
NY8A056AS16	SOP	16	150 mil	Tape & Reel: 2.5K pcs per Reel Tube: 50 pcs per Tube
NY8A056AS18	SOP	18	300 mil	Tape & Reel: 1.5K pcs per Reel Tube: 40 pcs per Tube
NY8A056AP18	PDIP	18	300 mil	Tube: 20 pcs per Tube
NY8A056AD20	SSOP	20	209 mil	Tape & Reel: 2K pcs per Reel Tube: 69 pcs per Tube