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

DATA SHEET

NY8TE64A

18 I/O, 13-ch ADC & 12-ch Touch Pad

8-bit EPROM-Based MCU

Version 1.0

Jun. 30, 2022

Revision History

Version	*Date*	*Description*	*Modified Page*
0.1	2021/04/30	Preliminary Edition	-
0.2	2021/08/04	Internal Modified	
0.3	2021/10/18	Timer5、PWM5 and CCP are NOT allowed to use.	
0.4	2021/10/21	1. Update pin assignment 2. Update features description	15, 16 9, 13
0.5	2021/11/30	1. Updated operating voltage 2. Fixed SSBEN share pin is PB7 3. Add Notice “If set TMxHRC=1 (x=1,4,5), user must disable Prescaler. “ 4. Updated Timer1/4/5 block diagrams 5. Updated figures of SPI masters connect with slaves 6. Updated Recommended Operating Voltage Table	9, 12 60 48, 54, 57 91, 95, 97 125, 126 161
0.6	2022/03/22	Update EEPROM Notice	26
0.7	2022/04/15	1. Revised V_HRC description 2. Amend UART baud rate formula 3. Revised operation mode diagram and table	11 71 116, 119
1.0	2022/06/30	1. Removed “version A” description 2. Removed Timer5 and CCP function 3. Update EEPROM Notice	- - 26, 27

Table of Contents

1. 概述

NY8TE64A 是以MTP作為程式記憶體，並以EEPROM作為非揮發性資料記憶體的 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。NY8TE64A 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。

NY8TE64A 內建高精度十二加二通道十二位元類比數位轉換器，與高精度電壓比較器，足以應付各種類比介面的偵測與量測。

在I/O的資源方面，NY8TE64A 有 18 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，NY8TE64A內建了可選擇頻率的大電流輸出紅外載波發射口(PA3)。

NY8TE64A 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NY8TE64A 提供 4 組 10 位元解析度的PWM輸出，1 組蜂鳴器輸出可用來驅動馬達、LED、或蜂鳴器等等。

NY8TE64A 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，NY8TE64A 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NY8TE64A 進入正常操作模式(Normal) 或 慢速模式(Slow mode) 來處理突發事件。

NY8TE64A內建除錯仿真電路，利用兩個腳位與很少的外接硬體，就能實現在線仿真器的大多數功能，例如設定程式執行條件與中斷條件，片上單步執行，以及查看及設定各種暫存器的內容。仿真的執行效果將比一般的仿真器更接近實際IC運作。

NY8TE64A內建的一組變頻振盪器，能彈性選擇輸出頻率區段，以適用於多樣的霧化器元件，更可以程式細調輸出頻率， 細度足以應付霧化器的元件偏差。

1.1 功能

- 寬廣的工作電壓：
  - 3.3V ~ 5.5V @ F_{INST} = 8MHz。
  - 2.2 V ~ 5.5V @ F_{INST}＜4MHz。
- 寬廣的工作溫度：-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)
- 4Kx14 bits MTP。

- 288 bytes SRAM。
- 256 bytes EEPROM。(一萬次讀寫)
- 18 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] 、PC[1:0]。
- PA[5:0]、PB[3:0] 及 PC[1:0] 可選擇輸入時使用內建下拉電阻。
- PA[7:0]、PB[7:0] 及 PC[1:0]可選擇輸入時使用上拉電阻。
- PB[7:0]、PC[1:0] 可選擇開漏極輸出(Open-Drain)。
- 所有I/O腳輸出可選擇小灌電流(Small Sink Current) 或一般灌電流(Normal Sink Current)或大灌電流(Large Sink Current) ，唯PB[7:6]、PC[0]不能選擇Small Sink Current。PA[4]可另外選擇Ultra Sink Current。
- 所有I/O腳輸出可選擇小推電流(Small Drive Current)或一般推電流(Normal Drive Current) ，唯PB[7:6]、PC[0]只能選擇Normal Drive Current。PA[4]可另外選擇Ultra Drive 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。
- 兩組 10 位元下數計時器(Timer1, 4)可選重複載入或連續下數計時。
- 四個 10 位元脈衝寬度調變(PWM1, 2, 3, 4)。
- 一個蜂鳴器輸出(BZ1)。
- 38/57KHz紅外線載波頻率可供選擇，同時載波之極性也可以根據數據作選擇。
- 大電流輸出紅外線載波發射口。
- 內建準確的低電壓偵測電路(LVD)。
- 內建十二加二通道 12 位元類比數位轉換器(Analog to Digital Converter)。
- 內建準確的電壓比較器(Voltage Comparator)。
- 內建上電復位電路(POR)。
- 內建低壓復位功能(LVR)。
- 內建看門狗計時(WDT)，可由程式韌體控制開關。
- 內建電阻頻率轉換器(RFC)功能。
- 雙時鐘機制，系統可以隨時切換高速振盪或者低速振盪。
  - 高速振盪：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)。

- 十四種硬體中斷：
  - Timer0 溢位中斷。
  - Timer1 借位中斷。
  - Timer4 借位中斷。
  - WDT 中斷。
  - PA/PB 輸入狀態改變中斷。
  - 三組外部中斷輸入。
  - 低電壓偵測中斷。
  - 比較器輸出轉態中斷。
  - 類比數位轉換完成中斷。
  - 串接介面模組(SIM)中斷
  - 觸摸計數器溢位中斷。
  - 觸摸比較中斷。
  - UART 介面模組讀或寫中斷。
  - EEPROM寫入完成中斷。
- NY8TE64A在待機模式(Standby mode)下的十二種喚醒中斷：
  - Timer0 溢位中斷。
  - Timer1 借位中斷。
  - Timer4 借位中斷。
  - WDT 中斷。
  - PA/PB 輸入狀態改變中斷。
  - 三組外部中斷輸入。
  - 低電壓偵測中斷。
  - 比較器輸出轉態中斷。
  - 類比數位轉換完成中斷。
  - 串接介面模組(SIM)中斷
  - 觸摸計數器溢位中斷。
  - 觸摸比較中斷。
- NY8TE64A在睡眠模式(Halt mode)下的三種喚醒中斷：
  - WDT 中斷。
  - PA/PB 輸入狀態改變中斷。
  - 三組外部中斷輸入。
- NY8TE64A有 12 通道的觸摸偵測腳位。
- 內建變頻振盪器(V_HRC) 提供 32MHz/20.8MHz/16MHz/13.6MHz 四種選擇。可微調精度至 ±0.1%。
- 內建二線控制的除錯仿真電路(On Chip Debug)。

1. General Description

NY8TE64A is a MTP based 8-bit MCU with EEPROM data memory built-in, tailored for I/O based applications like home appliances or meter equipment. NY8TE64A adopts advanced CMOS technology to provide customers remarkable solution with low cost, high performance and high noise immunity benefits. RISC architecture is applied to NY8TE64A 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, NY8TE64A is very suitable for those applications that are sophisticated but compact program size is required.

NY8TE64A provides 12 + 2 channel high-precision 12-bit analog-to-digital converter (ADC), and high-precision analog voltage comparator. They are suitable for any analog interface detection and measurement applications.

As NY8TE64A address I/O type applications, it can provide 18 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, NY8TE64A has built-in large infrared (IR) carrier generator with selectable IR carrier frequency and polarity for applications which demand remote control feature.

NY8TE64A 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, NY8TE64A provides 5 sets of 10-bit resolution Pulse Width Modulation (PWM) output and 1 sets of buzzer output in order to drive motor/LED and buzzer.

NY8TE64A 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, NY8TE64A 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 NY8TE64A operates in Standby mode and Halt mode, kinds of event will issue interrupt requests and can wake-up NY8TE64A to enter Normal mode and Slow mode in order to process urgent events.

NY8TE64A provide on-chip debug (OCD) facilities as a low cost alternative to ICE. With OCD and a minimum of extra hardware, NY8BE62D can do ICE tasks such as free running, single stepping, break point setting and internal ram/register accessing.

NY8TE4A built-in a variable high speed oscillator can be optioned for different center frequency. Users can adjust this frequency in the program in a very small frequency step. With this feature ensures that NY8TE64A will find excellent use in different ultrasonic nebulizer applications.

1.1 Features

- Wide operating voltage range:
  - 3.3V ~ 5.5V @ F_{INST} = 8MHz。
  - 2.2V ~ 5.5V @ F_{INST}＜4MHz。

- Wide operating temperature: -40°C ~ 85°C.
- High ESD over ±8KV.
- High EFT over ±4KV with Noise Filter Enable. (operating voltage @5V)
- 4K x 14 bits MTP.
- 288 bytes SRAM.
- 256 bytes EEPROM. (Endurance: 10,000 times)
- 18 general purpose I/O pins (GPIO), PA[7:0], PB[7:0], PC[1:0] with independent direction control.
- PA[5, 3:0] and PB[3:0] and PC[1:0] have features of Pull-Low resistor for input pin.
- PA[7:0] and PB[7:0] and PC[1:0] have features of Pull-High resistor, the value of Pull-High resistor can be 100KΩ or 1MΩ. (PA5 is about 80KΩ)
- PB[7:0] and PC[1:0] have features of Open-Drain output.
- I/O ports output current mode can be small sink, normal sink or large sink. PB[7:6] and PC[0] can not be small sink current mode. PA[4] cab be ultra sink current mode.
- I/O ports output current mode can be small drive, normal drive. PB[7:6] and PC[0] can not be small drive current mode. PA[4] cab be ultra drive current mode.
- 8-level hardware Stack.
- Direct and indirect addressing modes for data access.
- One 8-bit up-count timer (Timer0) with programmable prescaler.
- Two10-bit reload or continuous down-count timers (Timer1, 4).
- Four 10-bit resolution PWM (PWM1, 2, 3, 4) output.
- One buzzer (BZ1) output.
- Selectable 38/57KHz IR carrier frequency and high/low polarity according to data value.
- Built-in high-precision Low-Voltage Detector (LVD).
- Built-in 12+2 channel high-precision 12-bit ADC.
- Built-in high-precision Voltage Comparator.
- Built-in Power-On Reset (POR).
- Built-in Low-Voltage Reset (LVR).
- Built-in Watch-Dog Timer (WDT) enabled/disabled by firmware control.
- Built-in Resistance to Frequency Converter (RFC) function.
- Dual-clock oscillation: System clock can switch between high oscillation and low oscillation.
  - High oscillation: E_HXT (External High Crystal Oscillator, above 6MHz)
    E_XT (External Crystal Oscillator, 455K~6MHz)

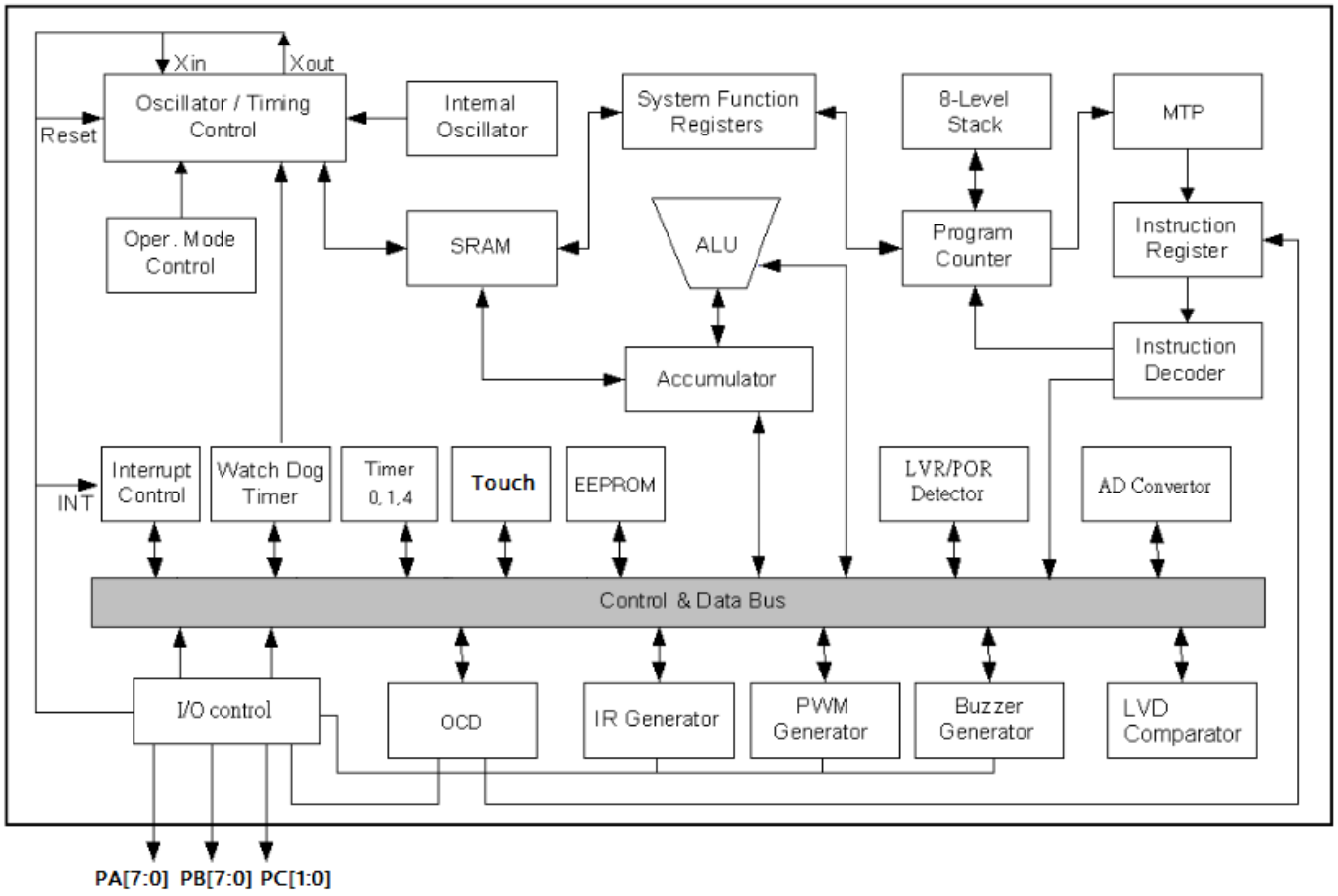
I_HRC (Internal High Resistor/Capacitor Oscillator ranging from 1M~20MHz)

- 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.
- Fourteen hardware interrupt events:
  - Timer0 overflow interrupt.
  - Timer1 underflow interrupt.
  - Timer4 underflow interrupt.
  - WDT timeout interrupt.
  - PA/PB input change interrupt.
  - 3 set External interrupt.
  - LVD interrupt.
  - Comparator output status change interrupt.
  - ADC end-of-convert interrupt.
  - Serial Interface Module (SIM)interrupt.
  - Touch counter overflow interrupt.
  - Touch comparator interrupt.
  - UART interface module read interrupt or write interrupt.
  - End of EEPROM write interrupt.
- Twelve interrupt events to wake-up NY8TE64A from Standby mode:
  - Timer0 overflow interrupt.
  - Timer1 underflow interrupt.
  - Timer4 underflow interrupt.
  - WDT timeout interrupt.
  - PA/PB input change interrupt.
  - External interrupts. (INT0, INT1, INT2).
  - LVD interrupt.
  - Comparator output status change interrupt.
  - ADC end-of-convert interrupt.
  - Serial Interface Module (SIM) interrupt.
  - Touch counter overflow interrupt.
  - Touch comparator interrupt.
- Three kinds of interrupt events to wake-up NY8TE64A from Halt mode:
  - WDT timeout interrupt.
  - PA/PB input change interrupt.
  - External interrupts. (INT0, INT1, INT2)

- Built-in 12 channel touch pad.
- Build-in 4 variable oscillator options: 32MHz/20.8MHz/16MHz/13.6MHz. Fine-tuned accuracy to ±0.1%.
- Build-in 2 wires on chip debug circuit (OCD).

1.2 Block Diagram



1.3 Pin Assignment

NY8TE64A provides two kinds of package type which are SOP20 and SOP-16.

20 pin

					VSS	1	20	**VDD**							
	XIN				**PA6**	2	19	**PA4**		AIN4	PWM4	EX_CK0	INT0		
	XOUT	(PWM3)			**PA7**	3	18	**PA3 (SCL1)**	TP3	AIN3		(IR)	INT1		
INT2	RSTB			TP4	**PA5**	4	17	**PA2 (SDA1)**	TP2	AIN2	PWM3	EXCKI			
T3OUT	BZ1	PWM2	AIN7	TP5	**PB3**	5	16	**PA1**	TP1	AIN1		(EX_CKI)			
			AIN6	TP6	**PB2**	6	15	**PA0**	TP0	AIN0		VREFH			
(INT1)	IR	PWM1	AIN5	TP7	**PB1**	7	14	**PB5**	TP11	AIN9	(PWM2)	IIC_SDA	(INT0)	SPI_MISO	
				CAP	**PB0**	8	13	**PB4**	TP10	AIN8	PWM1	IIC_SCL	T1OUT	SPI_SCK	
					(SDA0) PC0	9	12	**PB6**	TP9	AIN10		UART_TX		SPI_MOSI	
					(SCL0) PC1	10	11	**PB7**	TP8	AIN11		UART_RX		SPI_SSB	

16 Pin

					VDD	1	16	**VSS**							
	XIN				**PA6**	2	15	**PA4**		AIN4	PWM4	EX_CK0	INT0		
	XOUT	(PWM3)			**PA7**	3	14	**PA3 (SCL1)**	TP3	AIN3		(IR)	INT1		
INT2	RSTB			TP4	**PA5**	4	13	**PA2 (SDA1)**	TP2	AIN2	PWM3	EXCKI			
T3OUT	BZ1	PWM2	AIN7	TP5	**PB3**	5	12	**PB5**	TP11	AIN9	(PWM2)	IIC_SDA	(INT0)	SPI_MISO	
			AIN6	TP6	**PB2**	6	11	**PB4**	TP10	AIN8	PWM1	IIC_SCL	T1OUT	SPI_SCK	
(INT1)	IR	PWM1	AIN5	TP7	**PB1**	7	10	**PB6**	TP9	AIN10		UART_TX		SPI_MOSI	
				CAP	**PB0**	8	9	**PB7**	TP8	AIN11		UART_RX		SPI_SSB	

Figure 2 Package pin assignment of NY8TE64A

1.4 Pin Description

Pin Name	I/O	Description
PA0 AIN0 VREFH TP0	I/O	PA0 is bidirectional I/O pin, and can be comparator input pin. PA0 can be ADC analog input pin. PA0 can be ADC external high reference voltage source. PA0 can be Touch Pad0
PA1 AIN1 EX_CKI1 TP1	I/O	PA1 is bidirectional I/O pin, and can be comparator input pin. PA1 can be ADC analog input pin. PA1 can be Timer4/5 clock source EX_CKI1. PA1 can be Touch Pad1
PA2 AIN2 PWM3 EX_CKI1 TP2 (OCD_SDA)	I/O	PA2 is bidirectional I/O pin, and can be comparator input pin. PA2 can be ADC analog input pin. PA2 can be output of PWM3. PA2 can be Timer4/5 clock source EX_CKI1. PA2 can be Touch Pad2 PA2 can be programming pad SDA.
PA3 AIN3 IR INT1 TP3 (OCD_SDA)	I/O	PA3 is bidirectional I/O pin, and can be comparator input pin. PA3 can be ADC analog input pin. PA3 can be IR carrier output pin. PA3 can be input pin of external interrupt INT1 PA3 can be Touch Pad3 PA3 can be programming pad SDA.
PA4 AIN4 EX_CKI0 INT0 PWM4	I/O	PA4 is a bidirectional I/O pin. PA4 can be ADC analog input pin. PA4 can be the Timer0/1 clock source EX_CKI0. PA4 can be the input pin of external interrupt INT0. PA4 can be output of PWM4.
PA5 RSTb INT2 TP4	I/O	PA5 is a bidirectional I/O pin. PA5 can be the reset pin RSTb. PA5 can be the input pin of external interrupt INT2. PA5 can be Touch Pad4.
PA6 Xin	I/O	PA6 is a bidirectional I/O pin, and can be comparator analog input pin. PA6 can be the input pin of crystal oscillator Xin.
PA7 Xout PWM3	I/O	PA7 is a bidirectional I/O pin, and can be comparator analog input pin. PA7 can be the output pin of crystal oscillator Xout. PA7 also can be output of instruction clock. PA7 can be output of PWM3 by option select
PB0 CAP	I/O	PB0 is a bidirectional I/O pin. PB0 can be external reference capacitor. (define by register)

Pin Name	I/O	Description
PB1 AIN5 VREF IR TP7 PWM1 INT1	I/O	PB1 is a bidirectional I/O pin. PB1 can be ADC analog input pin. PB1 can be ADC external high reference voltage source. PB1 can be IR carrier output pin. PB1 can be Touch Pad7 PB1 can be output of PWM1 PB1 can be input pin of external interrupt INT1
PB2 AIN6 TP6	I/O	PB2 is a bidirectional I/O pin. PB2 can be ADC analog input pin. PB2 can be Touch Pad6.
PB3 AIN7 BZ1 CMPO PWM2 TP5	I/O	PB3 is a bidirectional I/O pin. PB3 can be ADC analog input pin.. PB3 can be the output Buzzer1. PB3 can be comparator output. PB3 can be output of PWM2. PB3 can be Touch Pad5.
PB4 AIN8 PWM1 T1OUT TP10 SPI_SCK IIC_SCL	I/O	PB4 is a bidirectional I/O pin PB4 can be analog input pin. PB4 can be output of PWM1.. PB4 can be Timer1 output pin. (T1OUT toggles when Timer1 underflow occurs) PB4 can be Touch Pad10 PB4 can be clock pin in SPI mode. PB4 can be clock pin in IIC mode.
PB5 AIN9 TP11 SPI_MISO IIC_SDA	I/O	PB5 is a bidirectional I/O pin. PB5 can be analog input. PB5 can be Touch Pad11. PB5 can be MISO(Master In Slave Out) pin in SPI mode. PB5 can be Data pin in IIC mode.
PB6 AIN10 SPI_MOSI UART_TX TP9	I/O	PB6 is a bidirectional I/O pin. PB6 can be MOSI(Master Out Slave In) pin in SPI mode. PB6 can be Tx pin in UART mode. PB6 can be Touch Pad9.
PB7 AIN11 TP8 SPI_SSB UART_RX	I/O	PB7 is a bidirectional I/O pin. PB7 can be analog input pin. PB7 can be Touch Pad8. PB7 can SSB(Slave Select, active-low) pin in SPI mode. PB7 can be Rx pin in UART mode.
PC0 OCD_SDA	I/O	PC0 is a bidirectional I/O pin PC0 also can be programming pad SDA.
PC1 OCD_SCL	I/O	PC1 is a bidirectional I/O pin PC1 can be programming pad SCL.

Pin Name	I/O	Description
VDD	-	Positive power supply.
VSS	-	Ground.

2. Memory Organization

NY8TE64A 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 NY8TE64A is 4K words. Therefore, the Program Counter (PC) is 12-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.

NY8TE64A provides instruction GOTOA, CALLA to address 256 location of program space. NY8TE64A also provides instructions FCALL and FGOTO 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.

NY8TE64A program ROM address 0xFFE~0xFFF are reserved space, if user tries to write code in these addresses will get unexpected false functions.

NY8TE64A program ROM address 0x00E~0x00F are preset rolling code can be released and used as normal program space.

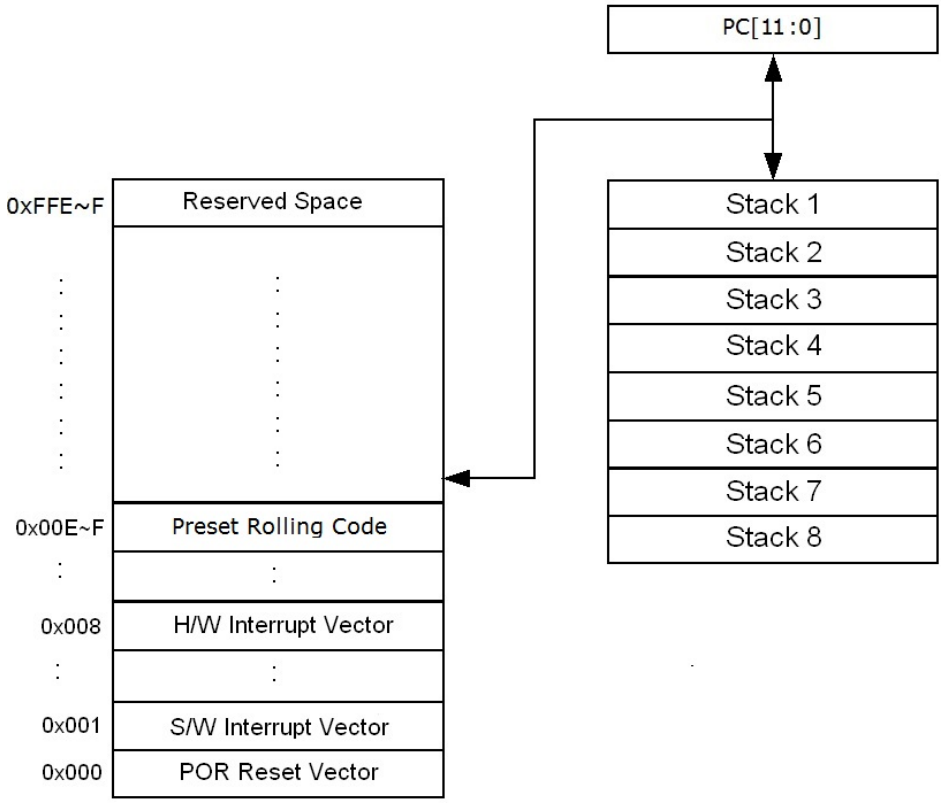


Figure 3 Program Memory Address Mapping

2.2 Data Memory

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

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

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

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

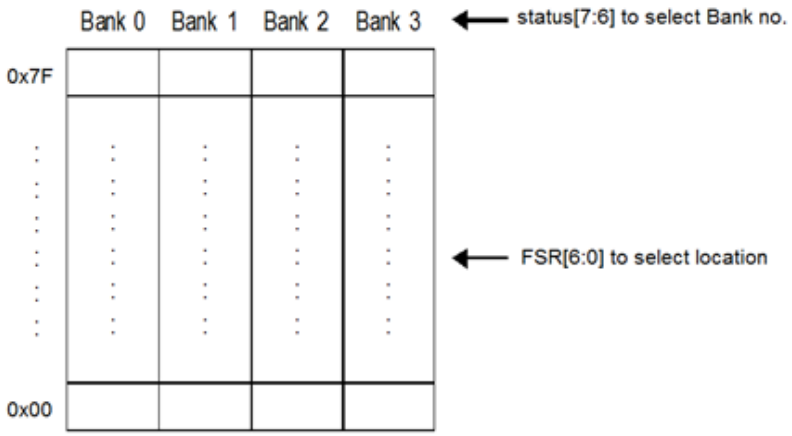


Figure 4 Indirect Addressing Mode of Data Memory Access

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

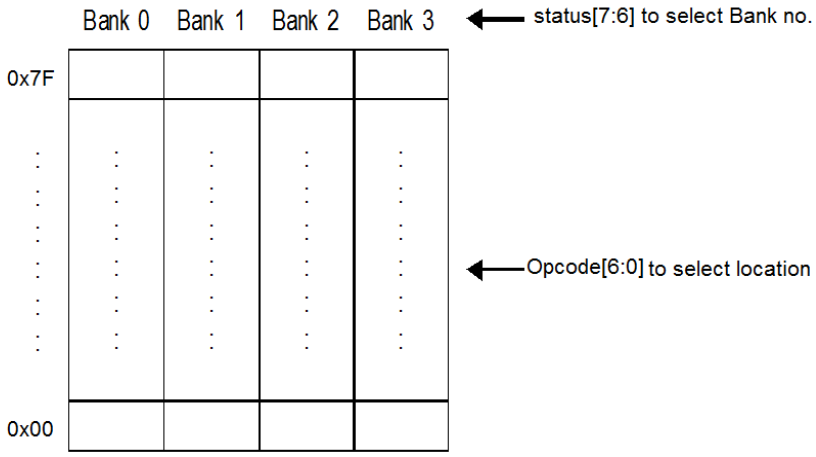


Figure 5 Direct Addressing Mode of Data Memory Access

R-page SFR can be accessed by general instructions like arithmetic instructions and data movement instructions. The R-page SFR occupies address from 0x0 to 0x1F of Bank 0. However, the same address range of Bank 1, Bank 2 and Bank 3 are mapped back to Bank 0. In other words, R-page SFR physically existed at Bank 0. The GPR physically occupy address from 0x20 to 0x7F of Bank 0 and 0x40 to 0x7F of Bank 1, 2 and 3. Other bank in address from 0x20 to 0x3F are mapped back as the Table 1 shows.

The NY8TE64A register name and address mapping of R-page SFR are described in the following table.

Status [7:6] **Address**	**00** **(Bank 0)**	**01** **(Bank 1)**	**10** **(Bank 2)**	**11** **(Bank 3)**
0x0	INDF	*The same mapping as Bank 0*		
0x1	TMR0			
0x2	PCL			
0x3	STATUS			
0x4	FSR			
0x5	PORTA			
0x6	PORTB			
0x7	PORTC			
0x8	PCON			
0x9	BWUCON			
0xA	PCHBUF			
0xB	ABPLCON			
0xC	BPHCON			
0xD	CPHCON			
0xE	INTE			
0xF	INTF			
0x10	ADMD			
0x11	ADR			
0x12	ADD			
0x13	ADVREFH			
0x14	ADCR			
0x15	AWUCON			
0x16	PACON			
0x17	ADJMD			
0x18	INTEDG			
0x19	TMRH			
0x1A	ANAEN			
0x1B	RFC	*The same mapping as Bank 0*		
0x1C	TM4RH			
0x1D	OSCCALH	-		
0x1E	OSCCALL			
0x1F	INTE2	*The same mapping as Bank 0*		

Status [7:6] Address	00 (Bank 0)	01 (Bank 1)	10 (Bank 2)	11 (Bank 3)
0x20 ~ 0x3F	General Purpose Register	*Mapped to bank0*	*Mapped to bank0*	*Mapped to bank0*
0x40 ~ 0x7F	General Purpose Register	*General Purpose Register*	*General Purpose Register*	*General Purpose Register*

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

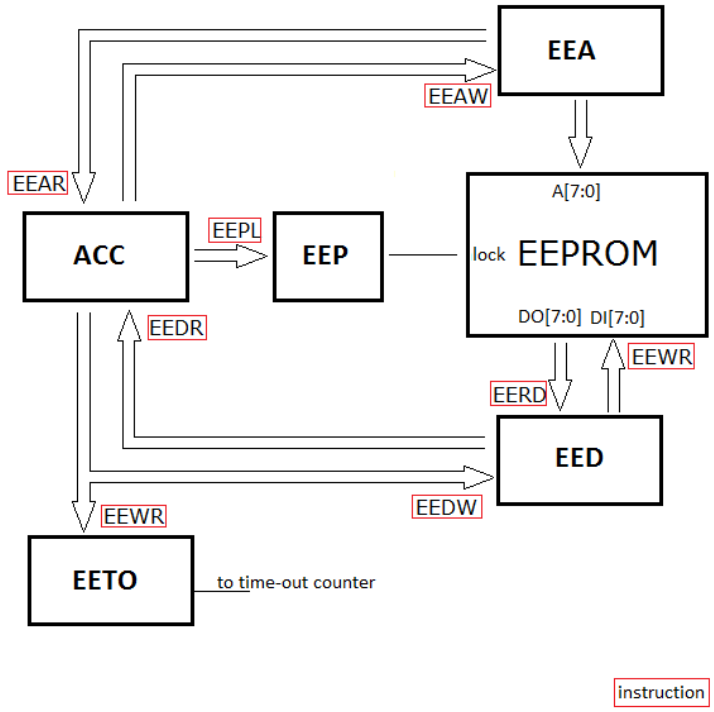
SFR Category Address	F-page SFR	S-page SFR	T-page SFR
0x0	-	TMR1	SIMCR
0x1	-	T1CR1	MADR
0x2	-	T1CR2	MFDR
0x3	-	PWM1DUTY	MCR
0x4	-	PS1CV	MSR
0x5	IOSTA	BZ1CR	SIMDR
0x6	IOSTB	IRCR	SPCR
0x7	IOSTC	TBHP	INTE3
0x8	-	TBHD	INTF3
0x9	APHCON	-	TPCKS
0xA	PS0CV	P2CR1	CASR
0xB	CPLCON	-	TPCHS
0xC	BODCON	PWM2DUTY	TPCR
0xD	CODCON	-	TPCNTL
0xE	CMPCR	-	TPCNTH
0xF	PCON1	OSCCR	TPPADEN
0X10	-	-	TPPADEN1
0X11	-	P3CR1	CASR1
0X12	-	-	CASR2
0X13	-	PWM3DUTY	-
0X14	-	-	TPCNTL1
0X15	-	TMR4	TPCNTL2
0X16	-	T4CR1	-
0X17	-	T4CR2	TPCNTH2
0X18	-	PWM4DUTY	THR/RBR
0X19	-	PS4CV	LCR
0X1A	-	-	LSR

SFR Category Address	F-page SFR	S-page SFR	T-page SFR
0X1B	-	-	DLL
0X1C	-	-	DLH
0X1D	-	-	-
0X1E	-	-	-
0X1F	-	-	PWMDB

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

2.3 EEPROM Memory

Read and write access to EEPROM memory take place indirectly through 3 special function registers, namely, EEA, EED and EEP. EEA register holds the EEPROM address to be access. EED register holds the data to be written, or the data read at the address in EEA. EEP holds the unlock key to write EEPROM data. The following figure shows the block diagram how EEPROM operates.



NY8TE64A provides 7 instructions to control the data flow between these 3 special register and 128-byte EEPROM. EEAR/EEAW read/write the EEA register. EEDR/EEDW read/write the EED register. EERD/EEWR /EEPL instructions, on the other hand, control the data path between EEPROM and EED register, according to the EEPROM address provided by EEA. The following table describes the EEPROM instructions.

Mnemonic Operands	Description	Status affected
EEAR	Read EEA to ACC	-
EEAW	Write EEA from ACC	-
EEDR	Read EED to ACC	-
EEDW	Write EED from ACC	-
EERD	Read EEPROM(EEA),and wite to EED register	ACC is unknown
EEWR	Write EEPROM(EEA) with data from EED register	-
EEPL	Write serial codes to unlock EE write protect	-

Before writing EEPROM data with the EEWR instruction, 3 consecutive code must be written to the EEP register with the EEPL instruction. These 3 code are C9H, 3AH and D3H. After these 3 code are written to EEP register, the EEPROM write protection is unlocked and data will be written to EEPROM by applying the EEWR instruction. The following are example code of EEPROM write unlock process.

```
; Write serial codes to lock/unlock EEP
    DISI
    MOVIA   0xC9
    EEPL
    MOVIA   0x3A
    EEPL
    MOVIA   0xD3
    EEPL
    ENI

; Write EEPROM data
    DISI
    MOVIA   0x45
    EEAW
    MOVIA   0x23
    EEDW
    MOVIA   EE_TO_8ms
    EEWR
    ENI
; Read EEPROM data
    DISI
    MOVIA
    EEAW
```

EERD

EEDR

ENI

When EEWR is successfully executed and completed, an EEWIF interrupt will be launched if the EEWIE is set to 1 and the global interrupt GIE is enabled.

Note:

1. ***To prevent unwanted halt in a running program if EEPROM writes fail, enabling the watchdog reset or enabling EEPROM write time-out functions is suggested.***
2. ***When writing EEPROM, be sure NOT to enable other interrupts.***
3. ***When user writes data to EEPROM, it is suggested to turn on LVD =2.4V function to monitor VDD.***
4. ***When VDD is under 2.4v, data cannot be written into EEPROM. User can use LVD & LVR function to prevent writing data to EEPROM fail.***

2.3.1 EEA (EEPROM Address Register)

Name	SFR Type	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EEA	EE	EEA[7:0]							
R/W Property		R/W							
Initial Value		XXXXXXX							

EEA[7:0]: Point to EEPROM address.

2.3.2 EED (EEPROM Data Register)

Name	SFR Type	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EED	EE	EED[7:0]							
R/W Property		R/W							
Initial Value		xxxxxxxx							

EED[7:0]: EEPROM data register . Read/Write this register by instruction EEDR/EEDW.

2.3.3 EEPL (EEPROM write protect Register)

Name	SFR Type	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EEPL	EE	PL[7:0]							
R/W Property		W							
Initial Value		xxxxxxxx							

PL[7:0]: EEPROM lock/unlock code. Written by instruction EEPL.

2.3.4 EETO (EEPROM Time-Out Register)

Name	SFR Type	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EETO	EE	-	-	-	-	-	TO2	TO1	TO0
R/W Property		-	-	-	-	-	W	W	W
Initial Value		X	X	X	X	X	X	X	X

TO[2:0]: EEPROM write time out period register. Written by EEWR instruction.

The time-out period is shown in the following table:

TO[2:0]	Time-out period
000	1ms
001	2ms
010	4ms
011	8ms
100	16ms
101	32ms
110	16ms
111	32ms

Ex:

```
MOVIA   0x01    ; setting write time-out period as 4ms
EEWR            ; write EEPROM
```

Note:

1. ***Before writing EEPROM, be sure to set initial data into EETO[2:0] at first.***
2. ***When writing EEPROM, be sure NOT to enable other interrupts.***
3. ***EEPROM time-out reference:***

   Time-out period set 4ms, if V_{OPL} = 2.2V ~2.3V

   Time-out period set 2ms, if V_{OPL} = 2.4V ~2.6V

   Time-out period set 1ms, if V_{OPL} ≧ 2.7V

3. Function Description

This chapter will describe the detailed operations of NY8TE64A.

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[11:0])

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 12-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[11:8], is not directly accessible. Update of PC[11:8] must be done through register PCHBUF.

For LGOTO instruction, PC[11:0] is from instruction word.

For LCALL instruction, PC[11:0] is from instruction word. Moreover the next PC address, i.e. PC+1, will push onto top of Stack.

3.1.4 STATUS (Status Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
STATUS	R	0x3	BK[1]	BK[0]	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*1

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

/PD=0, after instruction SLEEP is executed.

/TO: Time overflow flag bit*2

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

/TO=0, WDT timeout is occurred.

GP5: General purpose read/write register bit.

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

(*1) can be cleared by SLEEP instruction.

(*2) can be set by CLRWDT instruction.

3.1.5 FSR (Register File Selection Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FSR	R	0x4	GP7	FSR[6:0]						
R/W Property			R/W							
Initial Value			0	X	X	X	X	X	X	X

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

GP7: general register.

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 PortC (PortC Data Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PortC	R	0x7	-	-	-	-	-	-	PC1	PC0
R/W Property			-						R/W	
Initial Value			Data latch value is xx, read value is xx port value(PC1~PC0)							

While reading PortC, 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 PortC, data is written to PC's output data latch.

3.1.9 PCON (Power Control Register)

Name	SFR Type	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	R	0x8	WDTEN	/PLPA4	LVDEN	/PHPA5	LVREN	GP2	EEW_ERR	EELOCK
R/W Property			R/W						R	
Initial Value			1	1	0	1	1	0	0	1

GP2: General read/write register bits.

EELOCK: EEPROM write LOCK flag.

EELOCK=1, the EEPROM write is in lock state.

EELOCK=0, the EEPROM write is in unlock state.

EEW_ERR: EEPROM write Time-out flag. (Note: EEWR_TO configuration must be enabled)

EEW_ERR=1, the EEPROM write is time-out.

EEW_ERR=0, the EEPROM write is not time-out.

LVREN: Enable/disable LVR.

LVREN=1, enable LVR.

LVREN=0, disable LVR.

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

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

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

LVDEN: Enable/disable LVD.

LVDEN=1, enable LVD.

LVDEN=0, disable LVD.

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

/PLPA4=1, disable PA4 Pull-Low resistor.

/PLPA4=0, enable PA4 Pull-Low resistor.

WDTEN: Enable/disable WDT.

WDTEN=1, enable WDT.

WDTEN=0, disable WDT.

3.1.10 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			0	0	0	0	0	0	0	0

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

WUPBx=1, enable PBx wake-up function.

WUPBx=0, disable PBx wake-up function.

3.1.11 PCHBUF (High Byte of PC)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCHBUF	R	0xA	-	XSPD_STP	-		PCHBUF[3:0]			
R/W Property			-	W	-		R/W			
Initial Value			X	0	X		0			

PCHBUF[3:0]: Buffer of the 11^{th} ~ 8^{th} bit of PC.

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

3.1.12 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.

3.1.13 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	R/W	R/W	R/W	R/W	R/W	R/W	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.

3.1.14 CPHCON (PortC Pull-High Resistor Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CPHCON	R	0xD	-	-	-	-	-	-	/PHPC1	/PHPC0
R/W Property			-	-	-	-	-	-	R/W	R/W
Initial Value			X	X	X	X	X	X	1	1

/PHPCx: Disable/enable PCx Pull-High resistor, 0 ≤ x ≤ 1.

/PHPCx=1, disable PCx Pull-High resistor.

/PHPCx=0, enable PCx Pull-High resistor.

3.1.15 INTE (Interrupt Enable Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTE	R	0xE	INT1IE	WDTIE	-	LVDIE	T1IE	INT0IE	PABIE	T0IE
R/W Property			R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
Initial Value			0	0	X	0	0	0	0	0

T0IE: Timer0 overflow interrupt enable bit.

T0IE=1, enable Timer0 overflow interrupt.

T0IE=0, disable Timer0 overflow interrupt.

PABIE: PortA/PortB input change interrupt enable bit.

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

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

INT0IE: External interrupt 0 enable bit.

INT0IE=1, enable external interrupt 0.

INT0IE=0, disable external interrupt 0.

T1IE: Timer1 underflow interrupt enable bit.

T1IE=1, enable Timer1 underflow interrupt.

T1IE=0, disable Timer1 underflow interrupt.

LVDIE: Low-voltage detector interrupt enable bit.

LVDIE=1, enable low-voltage detector interrupt.

LVDIE=0, disable low-voltage detector interrupt.

WDTIE: WDT timeout interrupt enable bit.

WDTIE=1, enable WDT timeout interrupt.

WDTIE=0, disable WDT timeout interrupt.

INT1IE: External interrupt 1 enable bit.

INT1IE=1, enable external interrupt 1.

INT1IE=0, disable external interrupt 1.

3.1.16 INTF (Interrupt Flag Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF	R	0xF	INT1IF	WDTIF	-	LVDIF	T1IF	INT0IF	PABIF	T0IF
R/W Property			R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
Initial Value(note*)			0	0	X	0	0	0	0	0

T0IF: Timer0 overflow interrupt flag bit.

T0IF=1, Timer0 overflow interrupt is occurred.

T0IF must be clear by firmware.

PABIF: PortA/PortB input change interrupt flag bit.

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

PABIF must be clear by firmware.

INT0IF: External interrupt 0 flag bit.

INT0IF=1, external interrupt 0 is occurred.

INT0IF must be clear by firmware.

T1IF: Timer1 underflow interrupt flag bit.

T1IF=1, Timer1 underflow interrupt is occurred.

T1IF must be clear by firmware.

LVDIF: Low-voltage detector interrupt flag bit.

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

LVDIF must be clear by firmware.

WDTIF: WDT timeout interrupt flag bit.

WDTIF=1, WDT timeout interrupt is occurred.

WDTIF must be clear by firmware.

INT1IF: External interrupt 1 flag bit.

INT1IF=1, external interrupt 1 is occurred.

INT1IF must be clear by firmware.

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

3.1.17 ADMD (ADC mode Register)

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

ADEN: ADC enable bit.

ADEN=1, ADC is enabled.

START: Start an ADC conversion session.

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

EOC: ADC status bit, read-only.

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

EOC=0 : ADC is in procession.

GCHS: ADC global channel select bit.

GCHS=0 : disable all ADC input channel.

GCHS=1 : enable ADC input channel.

CHS3~0: ADC input channel select bits.

0000=select PA0 pad as ADC input,

0001=select PA1 pad as ADC input,

0010=select PA2 pad as ADC input,

0011=select PA3 pad as ADC input,

0100=select PA4 pad as ADC input,

0101=select PB1 pad as ADC input,

0110=select PB2 pad as ADC input,

0111=select PB3 pad as ADC input,

1000=select PB4 pad as ADC input,

1001=select PB5 pad as ADC input,

1010=select PB6 pad as ADC input,

1011=select PB7 pad as ADC input,

1100=select 1/4 VDD as ADC input.

1101=select GND as ADC input.

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

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

ADIF: ADC interrupt flag bit.

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

ADIF must be clear by firmware.

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

ADIE=1 : enable ADC interrupt.

ADIE=0 : disable ADC interrupt.

ADCK1~0: ADC clock select.

00: ADC clock=Fcpu/16, 01: ADC clock=Fcpu/8, 10: ADC clock=Fcpu/1, 11: ADC clock=Fcpu/2.

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

3.1.19 ADD (ADC output data Register)

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

AD11~4: High-byte ADC data buffer.

3.1.20 ADVREFH (ADC high reference voltage Register)

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

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

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

EVHENB=1: ADC reference high voltage is supplied by external pin PA0.

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

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

3.1.21 ADCR (Sampling pulse and ADC bit Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCR	R	0x14	PB CON7	PB CON6	PB CON5	PB CON4	SHCK1	SHCK0	ADCR1	ADCR0
R/W Property			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			0	0	0	0	1	0	1	0

SHCK1~0: Sampling pulse width select.

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

ADCR1~0: ADC conversion bit no. select.

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

PBCONx: PBx analog pin select, 4 ≤ x ≤ 7.

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

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

3.1.22 AWUCON (PortA Wake-up Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
AWUCON	R	0x15	WUPA7	WUPA6	WUPA5	WUPA4	WUPA3	WUPA2	WUPA1	WUPA0
R/W Property			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			0	0	0	0	0	0	0	0

WUPAx: Enable/disable PAx wake-up function, 0 ≤ x ≤ 7.

WUPAx=1, enable PAx wake-up function.

WUPAx=0, disable PAx wake-up function.

3.1.23 PACON (ADC analog pin Register)

Name	SFR Type	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PACON	R	0x16	PBCON3	PBCON2	PBCON1	PACON4	PACON3	PACON2	PACON1	PACON0
R/W Property			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			0	0	0	0	0	0	0	0

PACONx: PAx analog pin select, 0 ≤ x ≤ 4.

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

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

PBCONx: PBx analog pin select, 1 ≤ x ≤ 3.

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

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

3.1.23 ADJMD (ADC analog pin Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADJMD	R	0x17	-	-	ADJ_SIGN	ADJ4	ADJ3	ADJ2	ADJ1	ADJ0
R/W Property			-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			X	X	0	0	0	0	0	0

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

00000 = offset 0mV

11111 = offset 11mV.

ADJ_SIGN: adjustment sign bit

0 = adc data decrease

1 = adc data increase

Note: For application, please refer to NYIDE example code "ADC_Interrupt _AutoK".

3.1.24 INTEDG (Interrupt Edge Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTEDG	R	0x18	INT2DEG	EIS2	EIS1	EIS0	INT1G1	INT1G0	INT0G1	INT0G0
R/W Property			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			0	0	0	0	0	1	0	1

EIS2: External interrupt 2 select bit

EIS2=1, PA5 is external interrupt 2.

EIS2=0, PA5 is GPIO.

EIS1: External interrupt 1 select bit

EIS1=1, PB1/PA3 is external interrupt 1.

EIS1=0, PB1/PA3 is GPIO.

EIS0: External interrupt 0 select bit

EIS0=1, PA4 is external interrupt 0.

EIS0=0, PA4 is GPIO.

INT1G1~0: INT1 edge trigger select bit.

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

INT0G1~0: INT0 edge trigger select bit.

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

INT2DEG: INT2 edge trigger select bit.

0: falling edge, 1: rising edge.

3.1.25 TMRH (Timer High Byte Register)

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

TMR19, TMR18: Timer1 MSB 2 bits. Write these 2 bits will overwrite the 10-bit Timer1 load value of bit 9 and 8. Read these 2 bits will get the Timer1 bit9-8 current value.

PWM2DUTY9~8: PWM2 duty data MSB 2 bits.

PWM1DUTY9~8: PWM1 duty data MSB 2 bits.

3.1.26 ANAEN (Analog Circuit Enable Register)

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

CMPEN: Enable/disable voltage comparator.

CMPEN=1, enable voltage comparator.

CMPEN=0, disable voltage comparator.

3.1.27 RFC (RFC Register)

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

RFCEN: Enable/disable RFC function.

RFCEN=1, enable RFC function.

RFCEN=0, disable RFC function.

PSEL[3:0]: Select RFC pad.

PSEL[3:0]	RFC PAD
0000	PA0
0001	PA1
0010	PA2
0011	PA3
0100	PA4
0101	PA5
0110	PA6
0111	PA7
1000	PB0
1001	PB1
1010	PB2
1011	PB3
1100	PB4
1101	PB5
1110	PB6
1111	PB7

Table 3 RFC pad select

3.1.28 TM4RH (Timer4 High Byte Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TM4RH	R	0x1C	TMR49	TMR48	-	-	PWM4D9	PWM4D8	PWM3D9	PWM3D8
R/W Property			R/W	R/W	-	-	R/W	R/W	R/W	R/W
Initial Value			X	X	X	X	X	X	X	X

TMR49, TMR48: Timer4 MSB 2 bits. Write these 2 bits will overwrite the 10-bit Timer4 load value of bit 9 and 8. Read these 2 bits will get the Timer4 bit9-8 current value.

PWM4DUTY9~8: PWM4 duty data MSB 2 bits.

PWM3DUTY9~8: PWM3 duty data MSB 2 bits.

3.1.29 OSCCAL (Internal Oscillator Calibration)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSCCALH	R	0x1D	-	-	-	-	-	OSC CAL10	OSC CAL9	OSC CAL8
R/W Property			-	-	-	-	-	R/W	R/W	R/W
Initial Value			X	X	X	X	X	V_HRC 8 bits option trim data bit[7:5]		

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSCCALL	R	0x1E	OSC CAL7	OSC CAL6	OSC CAL5	OSC CAL4	OSC CAL3	OSC CAL2	OSC CAL1	OSC CAL0
R/W Property			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			V_HRC 8 bits option trim data bit[4:0]					1	0	0

OSCCAL10~0: V_HRC 8 bits calibration data load to OSCCAL10~OSCCAL3 and user can adjust OSCCAL10~0 at slow mode.

3.1.30 INTE2 (Interrupt Enable and Flag 2nd. Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTE2	R	0x1F	INT2IF	T4IF	-	-	INT2IE	T4IE	-	-
R/W Property			R/W	R/W	-	-	R/W	R/W	-	-
Initial Value			0	0	X	X	0	0	X	X

INT2IF: External interrupt 2 flag bit.
INT2IF=1, external interrupt 2 is occurred.
INT2IF must be clear by firmware.

T4IF: Timer4 underflow interrupt flag bit.
T4IF=1, Timer4 underflow interrupt is occurred.
T4IF must be clear by firmware.

INT2IE: External interrupt 2 enable bit.
INT2IE=1, enable external interrupt 2.
INT2IE=0, disable external interrupt 2.

T4IE: Timer4 underflow interrupt enable bit.
T4IE=1, enable Timer4 underflow interrupt.
T4IE=0, disable Timer4 underflow interrupt.

3.2 T0MD Register

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

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

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

PS0SEL[2:0]	Dividing Rate		
	PS0WDT=0 (Timer0)	PS0WDT=1 (WDT Reset)	PS0WDT=1 (WDT Interrupt)
000	1:2	1:1	1:2
001	1:4	1:2	1:4
010	1:8	1:4	1:8
011	1:16	1:8	1:16
100	1:32	1:16	1:32
101	1:64	1:32	1:64
110	1:128	1:64	1:128
111	1:256	1:128	1:256

Table 4 Prescaler0 Dividing Rate

PS0WDT: Prescaler0 assignment.

PS0WDT=1, Prescaler0 is assigned to WDT.

PS0WDT=0, Prescaler0 is assigned to Timer0.

Note: Always set PS0WDT and PS0SEL[2:0] before enabling watchdog or timer0 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.

GP6: General register.

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 IOSTC (PortC I/O Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOSTC	F	0x7	-	-	-	-	-	-	IOPC1	IOPC0
R/W Property			-	-	-	-	-	-	R/W	R/W
Initial Value			X	X	X	X	X	X	1	1

IOPCx: PCx I/O mode selection, 0 ≤ x ≤ 1.

IOPCx=1, PCx is input mode.

IOPCx=0, PCx is output mode.

3.3.4 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.

3.3.5 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.

3.3.6 CPLCON (PortC Pull-Low Resistor Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CPLCON	F	0xB	-	-	-	-	-	-	/PLPC1	/PLPC0
R/W Property			-						R/W	
Initial Value			X	X	X	X	X	X	1	1

/PLPCx: Disable/enable PCx Pull-Low resistor, 0 ≤ x ≤ 1.

/PLPCx=1, disable PCx Pull-Low resistor.

/PLPCx=0, enable PCx Pull-Low resistor.

3.3.7 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	R/W	R/W	R/W	R/W	R/W	R/W	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.

3.3.8 CODCON (PortC Open-Drain Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CODCON	F	0xD	-	-	-	-	-	-	ODPC1	ODPC0
R/W Property			-	-	-	-	-	-	R/W	R/W
Initial Value			X	X	X	X	X	X	0	0

ODPCx: Enable/disable open-drain of PCx, 0 ≤ x ≤ 1.

ODPCx=1, enable open-drain of PCx.

ODPCx=0, disable open-drain of PCx.

3.3.9 CMPCR (Comparator voltage select Control Register)

Name	SFR Type	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPCR	F	0xE	-	RBIAS_H	RBIAS_L	CMP_INV	PS1	PS0	NS1	NS0
R/W Property			-	R/W						
Initial Value			X	0	0	0	1	1	0	0

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

NS[1:0]	Inverting input
00	PA1
01	PA3
10	Bandgap (0.6V)
11	Vref

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

PS[1:0]	Non-inverting input
00	PA0
01	PA2
10	Vref
11	---

CMPF_INV: Comparator output inverse control bit.

CMPF_INV = 1, Inverse comparator output.

CMPF_INV = 0, Non-inverse comparator output.

RBIAS_L, RBIAS_H: Set corresponding voltage reference levels

(please refer to chapter 3.16.1)

Note: RBIAS_H and RBIAS_L must be set as "0" to avoid power consumption in Halt mode or Standby mode.

3.3.10 PCON1 (Power Control Register1)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON1	F	0xF	GIE	LVDOUT	LVDS3	LVDS2	LVDS1	LVDS0	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.

GIE: Global interrupt enable bit.

GIE=1, enable all unmasked interrupts.

GIE=0, disable all interrupts.

Set by instruction ENI, clear by instruction DISI, read by instruction IOSTR.

GP1: General purpose register bit.

LVDOUT: Low voltage detector output, read-only.

LVDS3~0 : Select LVD voltage.

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

Table 4 LVD voltage select

3.4 S-page Special Function Register

3.4.1 TMR1 (Timer1 Register)

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

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

3.4.2 T1CR1 (Timer1 Control Register1)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T1CR1	S	0x1	PWM1OEN	PWM1OAL	TM1OE	VFSEL1	TM1_HRC	T1OS	T1RL	T1EN
R/W Property			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			0	0	0	0	0	0	0	0

This register is used to configure Timer1 functionality.

T1EN: Enable/disable Timer1.

T1EN=1, enable Timer1.

T1EN=0, disable Timer1.

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

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

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

T1OS: Configure Timer1 operating mode while underflow is reached.

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

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

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

Table 8 Timer1 Functionality

TM1_HRC: Timer1 clock source selection.

TM1HRC=1, PWM1, 2, 3 & Timer 1 clock source is High Oscillator clock.

TM1HRC=0, PWM1, 2, 3 & Timer 1 clock source selection depends on T1CS register bit

Note: If set Timer1 clock source is high oscillator clock (TM1HRC=1), user must disable precaler1.

VFSEL1: Timer1 special clock source selection.

TM1HRC=1, PWM1, 2, 3 & Timer 1 clock source is special High Oscillator clock.

TM1HRC=0, PWM1, 2, 3 & Timer 1 clock source selection depends on T1CS register bit

Note: VFSEL1 has higher priority than TM1_HRC.

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 PB4.

PWM1OEN=0, PB4 is GPIO.

TM1OE: Enable/disable Timer1 match output, T1OUT toggle output when Timer1 underflow occurs.

TM1OE=1, enable T1OUT output to pad PB4.

TM1OE=0, PB4 is GPIO.

Note: T1OUT output to pad PB4 has higher priority than PWM1 output.

3.4.3 T1CR2 (Timer1 Control Register2)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T1CR2	S	0x2	-	-	T1CS	T1CE	/PS1EN	PS1SEL[2:0]		
R/W Property			-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			X	X	1	1	1	1	1	1

This register is used to configure Timer1 functionality.

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

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

Table 9 Prescaler1 Dividing Rate

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

/PS1EN: Disable/enable Prescaler1.

/PS1EN=1, disable Prescaler1.

/PS1EN=0, enable Prescaler1.

T1CE: Timer1 external clock edge selection.

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

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

T1CS: Timer1 clock source selection.

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

T1CS=0, Instruction clock is selected.

3.4.4 PWM1DUTY (PWM1 Duty Register)

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

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

3.4.5 PS1CV (Prescaler1 Counter Value Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS1CV	S	0x4	PS1CV[7:0]							
R/W Property			R							
Initial Value			1	1	1	1	1	1	1	1

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

3.4.6 BZ1CR (Buzzer1 Control Register)

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

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

BZ1FSEL[3:0]	BZ1 Frequency Selection	
	Clock Source	Dividing Rate
0000	Prescaler1 output	1:2
0001		1:4
0010		1:8
0011		1:16

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

Table 10 Buzzer1 Output Frequency Selection

BZ1EN: Enable/Disable BZ1 output.

BZ1EN=1, enable Buzzer1.

BZ1EN=0, disable Buzzer1.

3.4.7 IRCR (IR Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IRCR	S	0x6	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 3.17) can be used as IR clock source.

2. Division ratio for different oscillation type.

OSC. Type	57KHz	38KHz	Conditions
High IRC(4MHz)	64	96	HIRC mode (the input to IR module is set to 4MHz no matter what system clock is)
Xtal 3.58MHz	64	96	Xtal mode & IROSC358M=1
Xtal 455KHz	8	12	Xtal mode & IROSC358M=0

Table 11 Division ratio for different oscillation type

3.4.8 TBHP (Table Access High Byte Address Pointer Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TBHP	S	0x7	-	-	-	-	TBHP3	TBHP2	TBHP1	TBHP0
R/W Property			-	-	-	-	R/W	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[3:0] and ACC. ACC is the Low Byte of PC[11:0] and TBHP[3:0] is the high byte of PC[11:0].

3.4.9 TBHD (Table Access High Byte Data Register)

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

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

3.4.10 P2CR1 (PWM2 Control Register1)

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

This register is used to configure PWM2 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=1, PWM2 output will be present on PB5.

PWM2OEN=0, PB6 is GPIO.

3.4.11 PWM2DUTY (PWM2 Duty Register)

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

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

3.4.12 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 15 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 PB3.

CMPOE=1, enable comparator output to pad PB3.

CMPOE=0, disable comparator output to pad PB3.

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

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.

3.4.13 P3CR1 (PWM3 Control Register1)

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

This register is used to configure PWM3 functionality.

PWM3OAL: Define PWM3 output active state.

PWM3OAL=1, PWM3 output is active low.

PWM3OAL=0, PWM3 output is active high.

PWM3OEN: Enable/disable PWM3 output.

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

PWM3OEN=0, PA2 is GPIO.

3.4.14 PWM3DUTY (PWM3 Duty Register)

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

The reload value of 10-bit Timer1 stored on registers TMR[5:4] and TMR1[7:0] is used to define the PWM3 frame rate, and registers TM4RH[1:0] and PWM3DUTY[7:0] is used to define the duty cycle of PWM3.

3.4.15 TMR4 (Timer4 Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMR4	S	0x15	TMR4[7:0]							
R/W Property			R/W							

Initial Value	XXXXXXXX

When reading register TMR4, it will obtain current value of 10-bit down-count Timer4 at TMR4[9:0]. When writing register TMR4, it will write data from TM4RH[7:6] and Timer4 reload register to TMR4[9:0] current content.

3.4.16 T4CR1 (Timer4 Control Register1)

Name	SFR Type	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T4CR1	S	0x16	PWM4OEN	PWM4OAL	-	VFSEL4	TM4_HRC	T4OS	T4RL	T4EN
R/W Property			R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
Initial Value			0	0	X	0	0	0	0	0

This register is used to configure Timer4 functionality.

T4EN: Enable/disable Timer4.

T4EN=1, enable Timer4.

T4EN=0, disable Timer4.

T4RL: Configure Timer4 down-count mechanism while Non-Stop mode is selected (T4OS=0).

T4RL=1, initial value is reloaded from reload register TMR4[9:0].

T4RL=0, continuous down-count from 0x3FF when underflow is occurred.

T4OS: Configure Timer4 operating mode while underflow is reached.

T4OS=1, One-Shot mode. Timer4 will count once from the initial value to 0x00.

T4OS=0, Non-Stop mode. Timer4 will keep down-count after underflow.

T4OS	T4RL	Timer4 Down-Count Functionality
0	0	Timer4 will count from reload value down to 0x00. When underflow is reached, 0x3FF is reloaded and continues down-count.
0	1	Timer4 will count from reload value down to 0x00. When underflow is reached, reload value is reloaded and continues to down-count.
1	x	Timer4 will count from initial value down to 0x00. When underflow is reached, Timer4 will stop down-count.

Table 8 Timer4 Functionality

TM4_HRC: Timer4 clock source selection.

TM4HRC=1, PWM4 & Timer 4 clock source is High Oscillator clock.

TM4HRC=0, PWM4 & Timer 4 clock source selection depends on T4CS register bit

Note: If set Timer4 clock source is high oscillator clock (TM4HRC=1), user must disable precaler4.

VFSEL4: Timer4 special clock source selection.

TM4HRC=1, PWM4 & Timer 4 clock source is special High Oscillator clock.

TM4HRC=0, PWM4 & Timer 4 clock source selection depends on T4CS register bit

Note: VFSEL4 has higher priority than TM4_HRC.

PWM4OAL: Define PWM4 output active state.

PWM4OAL=1, PWM4 output is active low.

PWM4OAL=0, PWM4 output is active high.

PWM4OEN: Enable/disable PWM4 output.

PWM4OEN=1, PWM4 output will be present on PA4.

PWM4OEN=0, PA4 is GPIO.

3.4.17 T4CR2 (Timer4 Control Register2)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T4CR2	S	0x17	-	-	T4CS	T4CE	/PS4EN	PS4SEL[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 Timer4 functionality.

PS4SEL[2:0]: Prescaler4 dividing rate selection.

PS4SEL[2:0]	Dividing Rate
000	1:2
001	1:4
010	1:8
011	1:16
100	1:32
101	1:64
110	1:128
111	1:256

Table 9 Prescaler4 Dividing Rate

Note: Always set PS4SEL[2:0] at /PS1EN=1, or interrupt may be falsely triggered.

/PS4EN: Disable/enable Prescaler4

/PS4EN=1, disable Prescaler4.

/PS4EN=0, enable Prescaler4.

T4CE: Timer4 external clock edge selection.

T4CE=1, Timer4 will decrease one while high-to-low transition occurs on pin EX_CKI1.

T4CE=0, Timer4 will decrease one while low-to-high transition occurs on pin EX_CKI1.

T4CS: Timer4 clock source selection.

T1CS=1, External clock on pin EX_CKI1 is selected.(option=1:PA2, option=0:PA1)

T1CS=0, Instruction clock is selected.

3.4.18 PWM4DUTY (PWM4 Duty Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM4DUTY	S	0x18	PWM4DUTY[7:0]							
R/W Property			W							
Initial Value			XXXXXXXX							

The reload value of 10-bit Timer4 stored on registers TM4RH[7:6] and TMR4[7:0] is used to define the PWM4 frame rate, and registers TM4RH[3:2] and PWM4DUTY[7:0] is used to define the duty cycle of PWM4.

3.4.19 PS4CV (Prescaler4 Counter Value Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS1CV	S	0x19	PS4CV[7:0]							
R/W Property			R							
Initial Value			1	1	1	1	1	1	1	1

While reading PS4CV, it will get current value of Prescaler4 counter.

3.5 T-page Special Function Register

3.5.1 SIMCR (Serial interface Mode Control Register)

Name	SFR Type	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SIMCR	T	0x0	SIMC1 (SPE)	SIMC0 (MEN)	MSTA	SSB_PAD	RX_PAD EN	TX_PAD EN	RCLK_P ADEN	BAUDOZ _PADEN
R/W Property			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			0	0	0	0	0	0	0	0

SIMC1 : SPI mode enable

SIMC1 =1, SPI mode enable .

SIMC1 =0, SPI mode disable .

SIMC0 : IIC mode enable

SIMC0 =1, IIC mode enable .

SIMC0 =0, IIC mode disable .

MSTA : Selection of master mode /slave mode . (include IIC mode and SPI mode)

MSTA =1, select master mode .

MSTA =0, select slave mode .

SSB_PAD : Decide PB7 is pin share as SSBEN or not , this functionality is used at SPI mode .

SSB_PAD =1, set PB7 as SSB(Slave Select Bar).

SSB_PAD =0, set PB7 as GPIO.

RX_PADEN : UART interface RXPAD .

RX_PADEN =1, set PB7 as UART RX signal input .

RX_PADEN =0, set PB7 as GPIO.

TX_PADEN : UART interface TXPAD.

TX_PADEN =1, set PB6 as UART TX signal output.

TX_PADEN =0, set PB6 as GPIO .

RCLK_PADEN : UART interface RCLK_PADEN .

RCLK_PADEN =1, set PB5 as UART RX CKT clock input ,frequency =baud rate*/16 .

RCLK_PADEN =0, set PB5 as GPIO.

BAUDOZ_PADEN : UART interface Baudrate Freq*16 ouput pad .

RCLK_PADEN =1, set PB4 as UART TX CKT clock output ,frequency =baud rate *16 .

RCLK_PADEN =0, set PB4 as GPIO.

Note: Do NOT enable SPI mode with IIC mode or UART mode at the same time.

3.5.2 MADR (IIC mode Address register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MADR	T	0x1	MAD7	MAD6	MAD5	MAD4	MAD3	MAD2	MAD1	-
R/W Property			R/W	R/W	R/W	R/W	R/W	R/W	R/W	-
Initial Value			0	0	0	0	0	0	0	X

MAD1-MAD7: MAD1-MAD7 are the slave address bits of IIC mode

3.5.3 MFDR (IIC mode frequency register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MFDR	T	0x2	-	-	-	FD4	FD3	FD2	FD1	FD0
R/W Property			-	-	-	R/W	R/W	R/W	R/W	R/W
Initial Value			X	X	X	0	0	0	0	0

FD0-FD4: FD0-FD4 are used for clock rate selection, the serilal bit clock frequency is equal to the CPU clock divide by the divider shown in table as below.

Example : CPU CLOCK=1MHZ , FD[4 :0]=2 , IIC MODE CLK FREQ=1MHZ/28=35.7KHZ

FD4	FD3	FD2	FD1	FD0	DIVIDER
0	0	0	0	0	22
0	0	0	0	1	24
0	0	0	1	0	28

FD4	FD3	FD2	FD1	FD0	DIVIDER
1	0	0	0	0	352
1	0	0	0	1	384
1	0	0	1	0	448

FD4	FD3	FD2	FD1	FD0	DIVIDER
0	0	0	1	1	34
0	0	1	0	0	44
0	0	1	0	1	48
0	0	1	1	0	56
0	0	1	1	1	68
0	1	0	0	0	88
0	1	0	0	1	96
0	1	0	1	0	112
0	1	0	1	1	136
0	1	1	0	0	176
0	1	1	0	1	192
0	1	1	1	0	224
0	1	1	1	1	272

FD4	FD3	FD2	FD1	FD0	DIVIDER
1	0	0	1	1	544
1	0	1	0	0	704
1	0	1	0	1	768
1	0	1	1	0	896
1	0	1	1	1	1088
1	1	0	0	0	1408
1	1	0	0	1	1536
1	1	0	1	0	1792
1	1	0	1	1	2176
1	1	1	0	0	2816
1	1	1	0	1	3072
1	1	1	1	0	3584
1	1	1	1	1	4352

3.5.4 MCR (IIC mode control register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MCR	T	0x3	-	-	-	MTX	TXAK	-	-	-
R/W Property			-	-	-	R/W	R/W	-	-	-
Initial Value			X	X	X	0	0	X	X	X

MTX : Select transmit or receive mode in IIC mode.

1 = Transmit mode.

0 = Receive mode.

TXAK: Set acknowledge bit in IIC mode.

1 = Do not send acknowledge signal

0 = Send acknowledge at 9th lock bit

3.5.5 MSR (IIC mode status register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MSR	T	0x4	MCF	MAAS	MBB	MAL	-	SRW	MIF	RXAK
R/W Property			R	R	R	R/W	-	R	R/W	R
Initial Value			1	0	0	0	X	0	0	1

The MIF and MAL bits are software clearable, while the other bits are read only.

MCF : Data Transfer Complete

1 = A byte has been completed

0 = A byte is being transfer

MAAS: Addressed as Slave

1 = Currently addressed as slave

0 = not currently addressed

When MAAS is set, the MIF (IIC MODE interrupt) bit is also set.

IF MAAS=1,Then CPU needs to check the SRW bit and set its MTX bit accordingly.

MBB : Bus Busy

1 = Bus busy

0 = Bus idle

When a START signal is detected, MBB is set. When a STOP signal is detected, it is cleared.

MAL : Arbitration Lost

1 = lost arbitration in master mode

0 = No arbitration lost

This arbitration lost flag is set when the IIC MODE master loses arbitration during a Master transmission mode. When MAL is set, the MIF bit is also set. This bit must be cleared by software

SRW : Slave R/W select

1 = Read from slave, from calling master

0 = Write to slave from calling master

When MAAS is set, the R/W command bit of the calling address sent from the master is latched into this SRW bit. By checking this bit, device can then select slave transmit/receive mode by configuring MTX bit of the IIC MODE Control register

MIF : IIC interrupt

1 = IIC interrupt has occurred.

0 = IIC interrupt has not occurred.

When this bit is set, an interrupt is generated to the CPU if SIMIE is set. This bit is set when one of the following events occurs:

1) Completion of one byte of data transfer. It is set at the falling edge of the 9th clock - MCF set.

2) A match of the calling address with its own specific address in slave mode - MAAS set.

3) A loss of bus arbitration - MAL set.

This bit must be cleared by software in the interrupt routine.

RXAK : Receive Acknowledge

1 = No acknowledgment signal detected.

0 = Acknowledge signal detected after 8 bits data transmitted.

3.5.6 SIMDR (Serial interface mode data register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SIMDR	T	0x5	SIMD7	SIMD6	SIMD5	SIMD4	SIMD3	SIMD2	SIMD1	SIMD0
R/W Property			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			0	0	0	0	0	0	0	0

SIMD7-SIMD0 : if iic mode enable , they are iic mode data register .

If spi mode enable , they are spi mode data register .

3.5.7 SPCR (SPI control & status register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPCR	T	0x6	SPIF	WCOL	-	MODF	CPOL	CKEG	SPR[1:0]	
R/W Property			R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
Initial Value			0	0	X	0	0	0	0	0

SPIF: SPI Flag

1 = Transmission complete

0 = Transmission not complete

Read SPCR ,and then read or write SIMDR will clear SPIF and WCOL .

WCOL: Write Collision Bit

1 = Invalid write to SIMDR

0 = No invalid write to SIMDR

Read SPCR ,and then read or write SIMDR will clear SPIF and WCOL .

MODF: Mode Fault Bit

1 = SSBEN pulled low while MODF bit set

0 = SSBEN not pulled low while MODF bit set

Read SPCR , and then write SPCR will clear MODF

CPOL: Clock Polarity Bit

1 = SCK pin at logic 1 between transmissions, idle high

0 = SCK pin at logic 0 between transmissions, idle low

CKEG: SPI SCK clock active edge type selection

CPOL=1

0: SCK is high base level and data capture at SCK rising edge

1: SCK is high base level and data capture at SCK falling edge

CPOL=0

0: SCK is low base level and data capture at SCK falling edge

1: SCK is low base level and data capture at SCK rising edge

SPR[1:0] — SPI Clock Rate Bits

SPR[1:0]	SPI clock rate
00	system Clock □2
01	system Clock □4
10	system Clock □16
11	system Clock

3.5.8 INTE3 (Interrupt Enable 3th Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTE3	T	0x7	SIMIE	EEIE	-	LSRIE	TXIE	RXIE	TPOVIE	TPCPIE
R/W Property			R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
Initial Value			0	0	X	0	0	0	0	0

SIMIE:Serial interface mode interrupt enable bit

1 = Enable Serial interface mode interrupt

0 = Disable Serial interface mode interrupt

EEIE: End of EEPROM Writing interrupt enable bit.

EEIE=1, enable End of EEPROM Writing interrupt.

EEIE=0, disable End of EEPROM Writing interrupt.

LSRIE: LSR interrupt enable bit

1 = enable receiver line status interrupt

0 = disable receiver line status interrupt

TXIE : Transmitt holding register (THR) empty interrupt enable bit

1 = enable Transmitt holding register (THR) empty interrupt

0 = disable Transmitt holding register (THR) empty interrupt

RXIE : Receive one bye completely interrupt enable bit

1 = enable Receive one bye completely interrupt

0 = disable Receive one bye completely interrupt

TPOVIE:Touch pad counter overflow interrupt enable bit

1 = enable Touch pad counter overflow interrupt

0 = disable Touch pad counter overflow interrupt

TPCPIE:Touch pad compare completely interrupt enable bit

1 = enable Touch pad compare completely interrupt

0 = disable Touch pad compare completely interrupt

3.5.9 INTF3 (Interrupt Flag 3th Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF3	T	0x8	SIMIF	EEIF	-	LSRIF	THRE	READY	TPOVIF	TPCPIF
R/W Property			R	R/W	-	R/W	R/W	R/W	R/W	R/W
Initial Value			0	0	X	0	1	0	0	0

SIMIF: Serial interface mode interrupt Flag bit

1 = Serial interface mode interrupt is occurred

0 = Serial interface mode interrupt is not occurred

SIMIF show status of MIF if IIC mode enable, and SIMIF show status of SPIF if SPI mode enable .

EEIF: Enf of EEPROM writing interrupt flag bit.

EEIF=1, Enf of EEPROM writing interrupt is occurred.

EEIF must be clear by firmware.

LSRIF: LSR interrupt flag.

1: line status interrupt flag.

0: line status

THRE: Transmitter holding register(THR) empty flag

This bit indicates the controller is ready to accept a new character for transmission. It is set to logical 1 when a character is transferred from the transmitter holding register(TBR) into the transmitter shift register.

Write data to TBR will clear this flag.

READY: Data ready flag

It is set logical 1 whenever a complete incoming character has been received and transferred into the receiver buffer register.

Read the RBR data will clear this flag.

TPOVIF: Touch pad counter overflow interrupt Flag bit

1 = Touch pad counter overflow interrupt is occurred

0 = Touch pad counter overflow interrupt is not occurred

TPCPIF: Touch pad compare completely interrupt Flag bit

1 = Touch pad compare completely interrupt is occurred

0 = Touch pad compare completely interrupt is not occurred

Write TPCR or TPCHS, will clear TPOVIF and TPCPIF.

※When using the Touch function, do not write or clear INTF3; if TPOVIF and TPCPIF are directly written to 1, the Touch detection will be stopped, and if it is cleared to 0, the Touch detection will be executed on the internal circuit again.

3.5.10 TPCKS (Touch pad scan frequency register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TPCKS	T	0x9	-	-	-	WKUPT	-	TPCK2	TPCK1	TPCK0
R/W Property			-	-	-	R/W	-	R/W		
Initial Value			X	X	X	0	X	0	1	0

WKUPT: Touch slow mode wakeup period select

1= 16Hz (64ms)

0= 32Hz (32ms)

TPCK2~TPCK0: touch pad scan frequency select

100: 1.125MHz

011: 1.3125MHz

010: 1MHz

001: 0.875MHz

000: 0.75MHz

3.5.11 CASR (Touch pad extra capacitance select register0)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TPCAX0	T	0xA	-	CA0[6:0]						
R/W Property			-	R/W						
Initial Value			x	0	0	0	0	0	0	0

CA06~CA00: extra capacitance select

1111 1111 : 127*C array unit

1111 1110 : 126*C array unit

0000 0001: 1*C array unit

0000 0000: 0*C array unit

Note : C array unit = 0.6125p

3.5.12 TPCHS (Touch pad channel select register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TPCHS	T	0xB	-	-	-	CHS4	CHS3	CHS2	CHS1	CHS0
R/W Property			-	R	-	R/W				
Initial Value			x	x	x	0	0	0	0	0

CHS0~CHS4: Touch pad channel select, see as following table

TPCR	TPCHS[4:0]	Select channel	Pre-load data	Carray select
X01[1]	0xxxx[2]	TPx[3]	TPCNTH & TPCNTL	CASR
X01	10000	TP0~TP3(G0)	TPCNTH & TPCNTL	CASR
X01	10001	TP4~TP7(G1)	TPCNTH & TPCNTL1	CASR1
X01	10010	TP8~TP11(G2)	TPCNTH2 & TPCNTL2	CASR2
101	1010X	G0-G1-G0-G1….		
X01	10110	INKEY[4]	TPCNTH & TPCNTL	CASR
X01	10111	All Touch-key[5]	TPCNTH & TPCNTL	CASR
101	110XX	G0-G1-G2-G0….		

[1]

X01:mean could be 5(Touch slow mode) or 1(Touch scan mode)

[2] 0XXXX: mean could be 00000~01011

[3] TPx :mean could be TP0-TP11

[4]l NKEY: Internal Touch-key

[5] All Touch-key: All Touch-key which TPPADEN enable

G0~G2: mean Group0~Group2

3.5.13 TPCR (Touch pad control register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TPCR	T	0xC	-	-	-	-	-	TPMD2	TPMD1	TPMD0
R/W Property			-					R/W		
Initial Value			X	X	X	X	X	0	0	0

TPMD2~TPMD0 (Touch pad operation mode)

101 = TPSLOW (Touch slow mode, scan the channel which TPCHS select)

100= reserved

011= Initialize (Discharge Cs)

010= reserved

001= TPRUN (Scan the channel which TPCHS select).

000= TPSTP (Touch pad all stop)

3.5.14 TPCNTL0 (Touch pad low counter register)

Name	SFR Type	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TPCNTL0	T	0xD	TPCNT7	TPCNT6	TPCNT5	TPCNT4	TPCNT3	TPCNT2	TPCNT1	TPCNT0
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

R: Touch counter X low byte data

W: Pre-load Touch counter X low byte data

3.5.15 TPCNTH (Touch pad high counter register)

Name	SFR Type	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TPCNTH0/1	T	0xE	TPCNT1_11	TPCNT1_10	TPCNT1_9	TPCNT1_8	TPCNT11	TPCNT10	TPCNT9	TPCNT8
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

R: Touch counter X high byte data

W: Pre-load Touch counter X high byte data

TPCNT0<11:0>: when address is read, the data is touch pad counter value

when address is write, the data will be 1's complement write to pre-load data value

TPCNT1<11:0>: when address is read, the data is touch pad counter value

when address is write, the data will be 1's complement write to pre-load data value

3.5.16 TPPADEN (Touch pad enable register0)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TPPADEN0	T	0xF	TP7EN	TP6EN	TP5EN	TP4EN	TP3EN	TP2EN	TP1EN	TP0EN
R/W Property			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			0	0	0	0	0	0	0	0

TPPADEN0<3:0>: I/O is Touch pad or not

TP0EN=1 PA2 act as TP0, TP0EN =0 PA2 is I/O.

TP1EN=1 PA3 act as TP1, TP1EN =0 PA3 is I/O

TP2EN=1 PA5 act as TP2, TP2EN =0 PA5 is I/O

TP3EN=1 PA6 act as TP3, TP3EN =0 PA6 is I/O

TP4EN=1 PA7 act as TP4, TP4EN =0 PA7 is I/O.

TP5EN=1 PB1 act as TP5, TP5EN =0 PB1 is I/O

TP6EN=1 PB2 act as TP6, TP6EN =0 PB2 is I/O

TP7EN=1 PB3 act as TP7, TP7EN =0 PB3 is I/O

TP0EN-TP11EN or ENINKEY have any one is enable, PB0 will used as CAP_PIN .

3.5.17 TPPADEN1 (Touch pad enable register1)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TPPADEN1	T	0x10	-	-	-	-	TP11EN	TP10EN	TP9EN	TP8EN
R/W Property			-	-	-	-	R/W	R/W	R/W	R/W
Initial Value			X	X	X	X	0	0	0	0

TPPADEN1<3:0>: I/O is Touch pad or not

TP8EN =1 PB4 act as TP8, TP8EN =8 ,PB4 is I/O.

TP9EN =1 PB5 act as TP9, TP9EN =9 ,PB5 is I/O

TP10EN =1 PB6 act as TP10, TP10EN =10, PB6 is I/O

TP11EN =1 PB7 act as TP11, TP11EN =11, PB7 is I/O

TP0EN - TP11EN or ENINKEY have any one is enable , PB0 will used as CAPN .

3.5.18 CASR1(Touch pad extra capacitance select register1)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TPCAX1	T	0x11	-	CA1[6:0]						
R/W Property			-	R/W						
Initial Value			X	0	0	0	0	0	0	0

CA16~CA10: extra capacitance select, used by Touch G1

1111 1111 : 127*C array unit

1111 1110 : 126*C array unit

0000 0001: 1*C array unit

0000 0000: 0*C array unit

Note : C array unit = 0.6125p

3.5.19 CASR2(Touch pad extra capacitance select register2)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TPCAX2	T	0x12	-	CA2[6:0]						
R/W Property			-	R/W						
Initial Value			X	0	0	0	0	0	0	0

CA26~CA20: extra capacitance select, used by Touch G2

1111 1111 : 127*C array unit

1111 1110 : 126*C array unit

0000 0001: 1*C array unit

0000 0000: 0*C array unit

Note : C array unit = 0.6125p

3.5.20 TPCNTL1 (Touch pad low counter register1)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TPCNTL1	T	0x14	TPCNT1_7	TPCNT1_6	TPCNT1_5	TPCNT1_4	TPCNT1_3	TPCNT1_2	TPCNT1_1	TPCNT1_0
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

R: Touch counter G1 X low byte data

W: Pre-load Touch counter G1 X low byte data

3.5.21 TPCNTL2 (Touch pad low counter register2)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TPCNTL2	T	0x15	TPCNT2_7	TPCNT2_6	TPCNT2_5	TPCNT2_4	TPCNT2_3	TPCNT2_2	TPCNT2_1	TPCNT2_0
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

R: Touch counter G2 X low byte data

W: Pre-load Touch counter G2 X low byte data

3.5.22 TPCNTH2 (Touch pad high counter register2)

Name	SFR Type	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TPCNTH2	T	0x17	-	-	-	-	TPCNT2_11	TPCNT2_10	TPCNT2_9	TPCNT2_8
R/W Property			-	-	-	-	R/W	R/W	R/W	R/W
Initial Value			X	X	X	X	1	1	1	1

R: Touch counter X high byte data

W: Pre-load Touch counter X high byte data

TPCNT2<11:0>: when address is read, the data is touch pad counter value

When address is write, the data will be 1's complement write to pre-load data value

3.5.23 THR/RBR (Transmit holding register /Receive Buffer Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
THR/RBR	T	0x18	URD7	URD6	URD5	URD4	URD3	URD2	URD1	URD0
R/W Property			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			X	X	X	X	X	X	X	X

URD7~URD0: If write data to the register, they are UART transmit data register.

If read data from the register, they are UART receive data register.

3.5.24 LCR (Line Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCR	T	0x19	LOOP	SBRK	PSTUCK	PEVEN	PREN	STPS	WL1	WL0
R/W Property			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			0	0	0	0	0	0	0	0

WL1~WL0: Word length select bits:

WL[1:0]	Word length bits
00	5 bits per character
01	6 bits per character
10	7 bits per character
11	8 bits per character

STPS: number of STOP bits.

STPS	Word Length	Number of Stop bits
0	X	1
1	5	1.5
1	6,7,8	2

PREN: Parity Enable

1: A parity bit is generated(transmit data) or check(receive data) between the last data word and stop bit of the serial data.

0: No parity is generated.

PEVEN: Even parity select

1: an even number of bits is sent or checked.

0: an odd number of bits is sent or checked.

PSTUCK: Stuck Parity

1: With PEVEN=1, the parity bit is sent and then detected by the receiver as a logic 0.

With PEVEN=0, the parity bit is logic 1.

SBRK: Set Break.

1: the serial output is forced to the spacing (logic 0) state and remains there regardless of other transmitter activity.

LOOP: loop back test enable

1: The output of the transmitter shift is loop back to the receiver shift register input.

0: disable the loop back test.

3.5.25 LSR (Line Status Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LSR	T	0x1A	-	TSRE	-	BKINT	FERR	PERR	OERR	-
R/W Property			-	-	-	-	-	-	-	-
Initial Value			X	1	X	0	0	0	0	X

OVERR: Over run error flag.

This bit indicates that data in the receiver's buffer register was not read by MCU before the next character was transferred into the register thereby destroying the previous character.

1: Over run happen.

0: Not happen over run.

PERR: Parity error flag

1: Detect a parity error.

0: No parity error.

FERR: Frame error flag

This bit indicates the received character did not have a valid stop bit. It is set to logical 1 whenever the stop bit following the last data bit or parity bit is detected as a zero bit.

BKINT: Break interrupt flag

This bit is set to logical 1 whenever the receiver data input is held in the spacing state (logical 0) for longer than a full word transmission time.

TSRE: Transmitter shift register(TSR) empty:

This bit is set to logical 1 whenever the transmitter holding register and the transmitter shift register are both empty. It is reset to logical 0 whenever THR or TSR contains a data character,

3.5.26 DLL (Baud Rate Divisor Latch LSB Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DLL	T	0x1B	DLL7	DLL6	DLL5	DLL4	DLL3	DLL2	DLL1	DLL0
R/W Property			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			0	0	0	0	0	0	0	0

DLL[7:0]: Baud rate divider LSB bit.

3.5.27 DLH (Baud Rate Divisor Latch MSB Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DLH	T	0x1C	DLH7	DLH6	DLH5	DLH4	DLH3	DLH2	DLH1	DLH0
R/W Property			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			0	0	0	0	0	0	0	0

DLH[7:0]: Baud rate divider MSB bit.

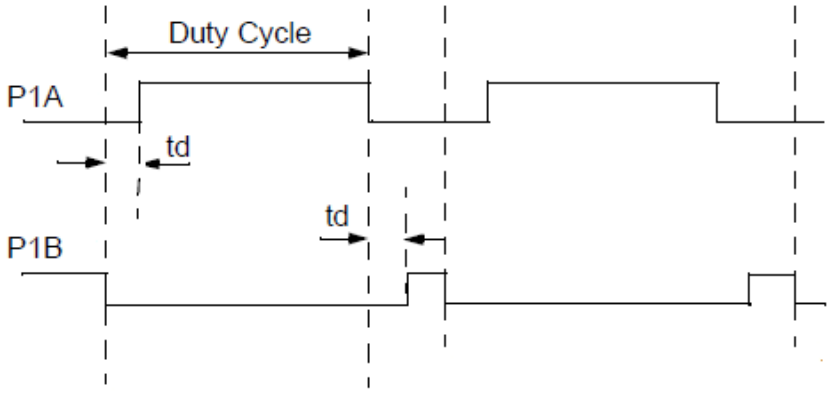
Baud rate = HIRC_freg÷(16xN), N= [DLH[7:0], DLL[7:0]]

3.5.28 PWMDB (PWM Dead Band Control Register)

Name	SFR Type	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDB	T	0x1F	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
R/W Property			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			X	X	X	X	X	X	X	X

PWMDB[7:0]: PWM dead band (delay) count for Half-Bridge output mode.

Numbers of CPU cycles (F_{INST}) between the P1A transition and P1B transition.



3.6 I/O Port

NY8TE64A provides 18 I/O pins which are PA[7:0], PB[7:0] and PC[1:0]. User can read/write these I/O pins through registers PORTA, PORTB and PORTC 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]. Register IOSTC[1:0] define the input/output direction of PC[1:0].

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], PCON[4] and APHCON[4:0] are used to enable or disable Pull-High resistor of PA[7: 0]. Register APHCON[5], PCON[6] and ABPLCON[3:0] are used to enable or disable Pull-Low resistor of PA[50]. 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]. Register CPHCON[1:0] are used to enable or disable Pull-High resistor of PC[1:0]. Register CPLCON[1:0] are used to enable or disable Pull-Low resistor of PC[1:0]. 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. When an PortC I/O pin is configured as output pin, there is a corresponding and individual register to select as Open-Drain output pin. Register CODCON[1:0] determine PC[1:0] is Open-Drain or not.

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

Feature		PA[5:0]	PA[7:6]	PB[3:0]	PB[7:4]	PC[1:0]
Input	Pull-High Resistor	V	V	V	V	V
	Pull-Low Resistor	V	X	V	X	V
Output	Open-Drain	X	X	V	V	V

Table 19 Summary of Pad I/O Feature

The level change on each I/O pin of PA and PB may generate interrupt request. Register AWUCON[7:0] and BWUCON[7:0] will select which I/O pin of PA and PB may generate this interrupt. As long as any pin of PA and PB is selected by corresponding bit of AWUCON and BWUCON, the register bit PABIF (INTF[1]) will set to 1 if there is a level change occurred on any selected pin. An interrupt request will occur and interrupt service routine will be executed if register bit PABIE (INTE[1]) and GIE (PCON1[7]) are both set to 1.

There is three external interrupt provided by NY8TE64A. When register bit EIS0 (INTEDG[4]) is set to 1, PB5/PA4 is used as input pin for external interrupt 0. When register bit EIS1 (INTEDG[5]) is set to 1, PB1/PA3 is used as input pin for external interrupt 1. When register bit EIS2 (INTEDG[6]) is set to 1, PA5 is used as input pin for external interrupt 2.

Note: When PA3, PA4 or PA5 is both set as level change operation and external interrupt, the external interrupt will have higher priority, and the PA3, PA4 or PA5 level change operation will be disabled.

NY8TE64A provides IR carrier generation output. It is depended both on register bit IREN (IRCR[0]) and configuration word IR Current. As the table 20 shows : When IREN=1 and IR current option=Normal, the IR sink

current=60mA and carrier output will be presented on PB1 pad. When IREN=1 IR sink current output will be presented on PA3 pad. When IREN=0, the IR carrier will not be generated.

IREN Register	IR Current Option	Current Value (mA)	IR Pad
0	X	-	-
1	Normal/Large	20/50	PB1/ PA3

Table 20 IR carrier selection

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 NY8TE64A 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. PA2 or PA1 can be Timer4 external clock source EX_CKI1 if T4CS=1.

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

PA2/PA7 can be PWM3 output If P3CR1[7] PWM3OEN=1.

PB3/PB5 can be PWM2 output If P2CR1[7] PWM2OEN=1.

PB1/PB4 can be PWM1 output If T1CR1[7] PWM1OEN=1.

PB4 can be T1OUT output if T1CR1[5] TM1OE=1.The output priority of PB4 is T1OUT > PWM1.

When configured as output, the sink current of each pin can be normal (15mA for V_{DD} =3V), large (35mA for V_{DD} =3V) or small (3mA for V_{DD} =2~5.5V) according to configuration words.

IIC mode, PB4 is used as SCK, PB5 is used as SDA .

SPI mode, PB4 is used as SCK, PB5is used as MISO , PB6 is used as MOSI ,PB7 can be SSBEN if SIMCR[4]=1 .

UART mode, PB4 TX clock output, PB5 RX clock input PB6 is used as TX and PB7 is used as RX.

3.6.1 Block Diagram of IO Pins

IO_SEL: set pad attribute as input or output

WRITE_EN: write data to pad.

READ_EN: read pad.

PULLUP_ENB: enable Pull-High.

PULLDOWN_EN: enable Pull-Low.

VPEN: enable pad to comparator non-inverting input.

CMPV: comparator non-inverting i

RD_TYPE: select read pin or read latch.

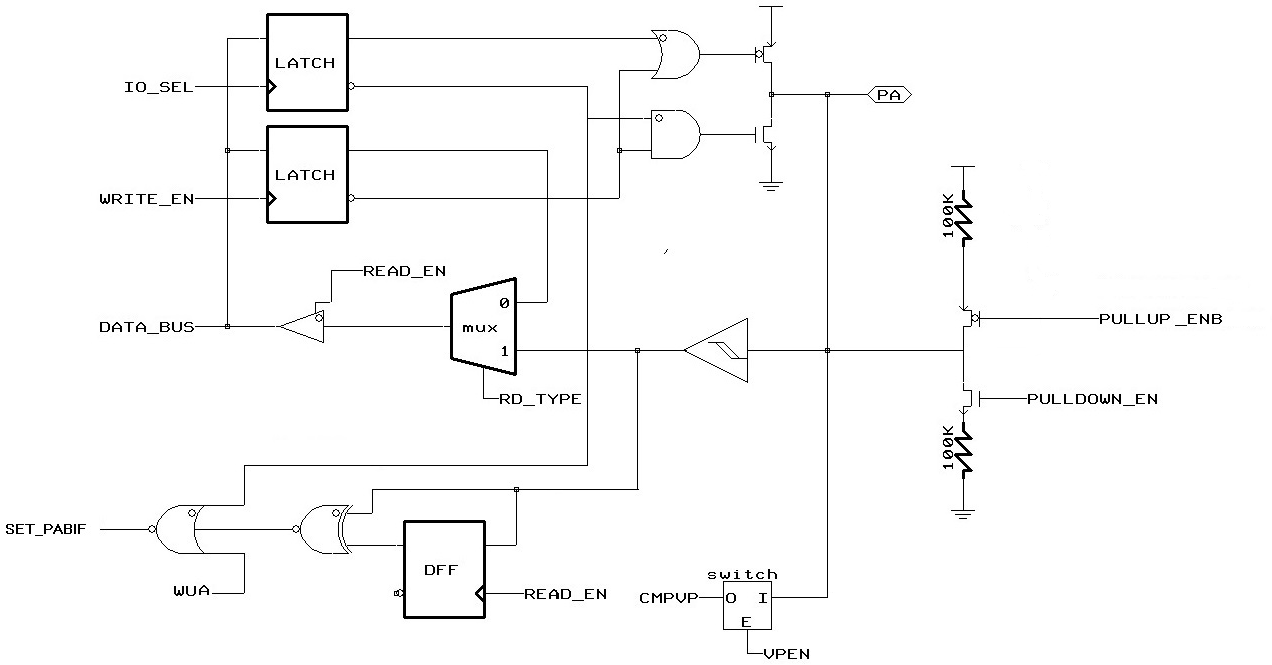


Figure 6 Block Diagram of PA0

IO_SEL: set pad attribute as input or output

WRITE_EN: write data to pad.

READ_EN: read pad.

PULLUP_ENB: enable Pull-High.

PULLDOWN_EN: enable Pull-Low.

VNEN: enable pad to comparator inverting input.

CMPVN: comparator inverting i

RD_TYPE: select read pin or read latch.

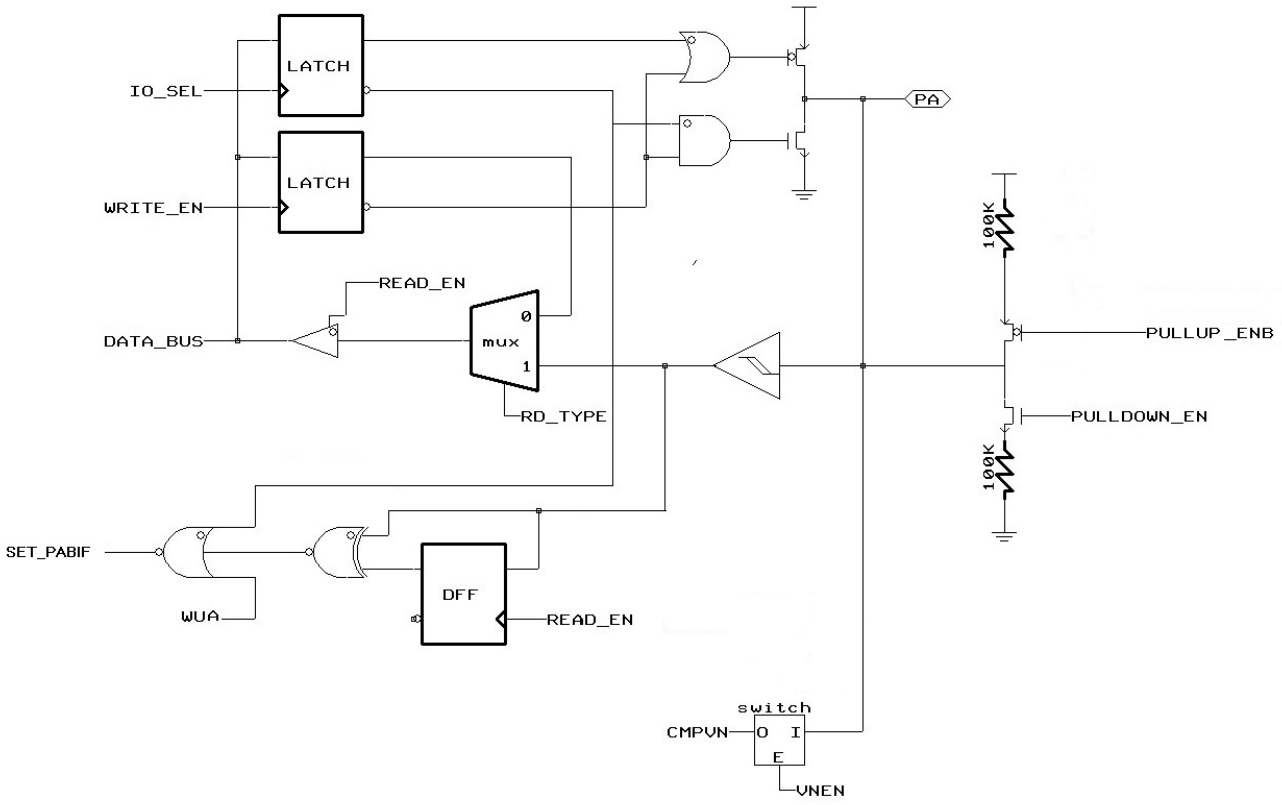


Figure 7 Block Diagram of PA1

IO_SEL: set pad attribute as input or output.

WRITE_EN: write data to pad.

READ_EN: read pad.

PULLUP_ENB: enable Pull-High.

PULLDOWN_EN: enable Pull-Low.

VPEN: enable pad to comparator non-inverting input.

VNEN: enable pad to comparator inverting input.

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

RD_TYPE: select read pin or read latch.

EX_CKI1: external clock for Timer4, 5.

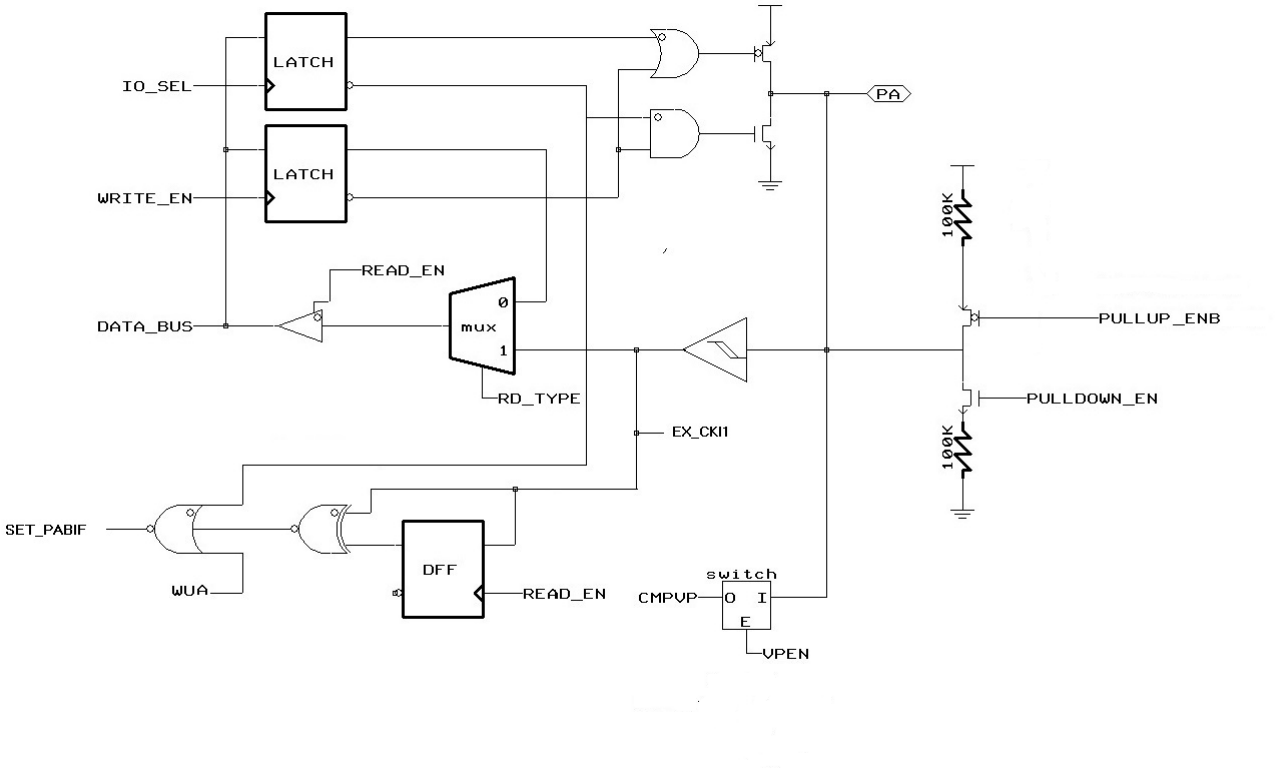


Figure 8 Block Diagram of PA2

IO_SEL: set pad attribute as input or output.

WRITE_EN: write data to pad.

READ_EN: read pad.

PULLUP_ENB: enable Pull-High.

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.

EIS1: external interrupt function enable.

EX_INT1: external interrupt signal.

RD_TYPE: select read pin or read latch.

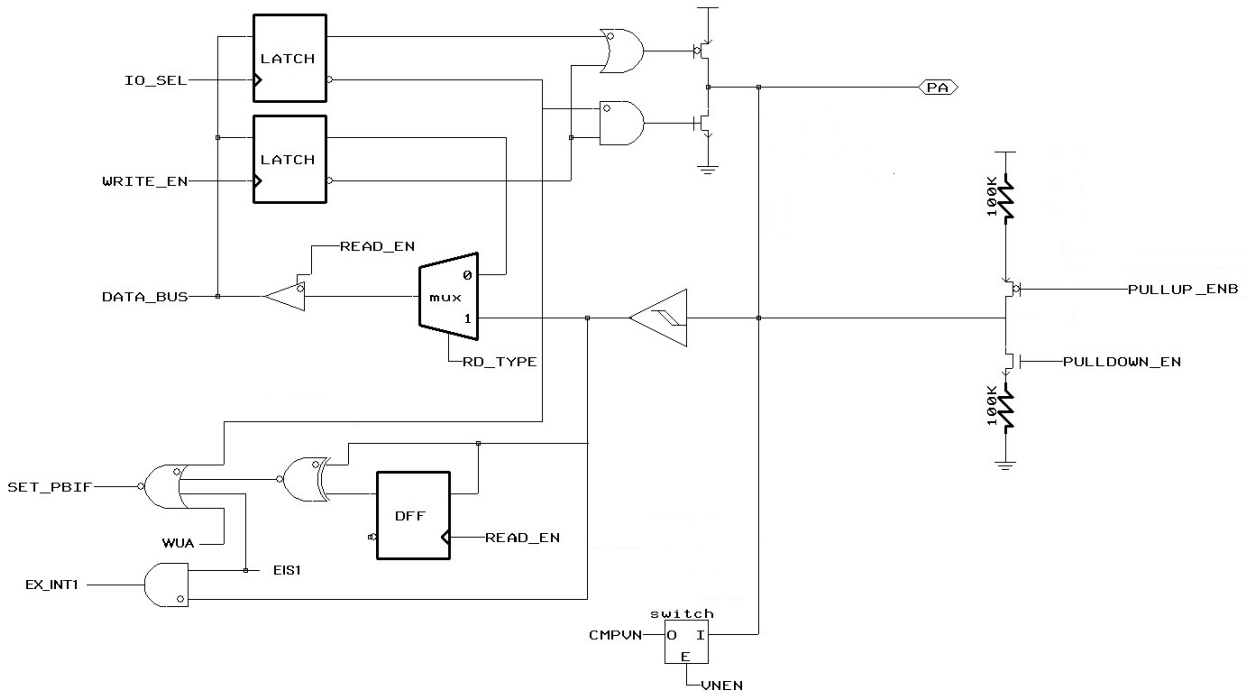


Figure 9 Block Diagram of PA3

IO_SEL: set pad attribute as input or output.

WRITE_EN: write data to pad.

READ_EN: read pad.

PULLUP_ENB: enable Pull-High.

PULLDOWN_EN: enable Pull-Low.

VPEN: enable pad to comparator non-inverting input.

CMPVP: comparator non-inverting input.

EIS0: external interrupt function enable.

EX_INT0 external interrupt signal.

RD_TYPE: select read pin or read latch.

EX_CKI0: external clock for Timer0, 1.

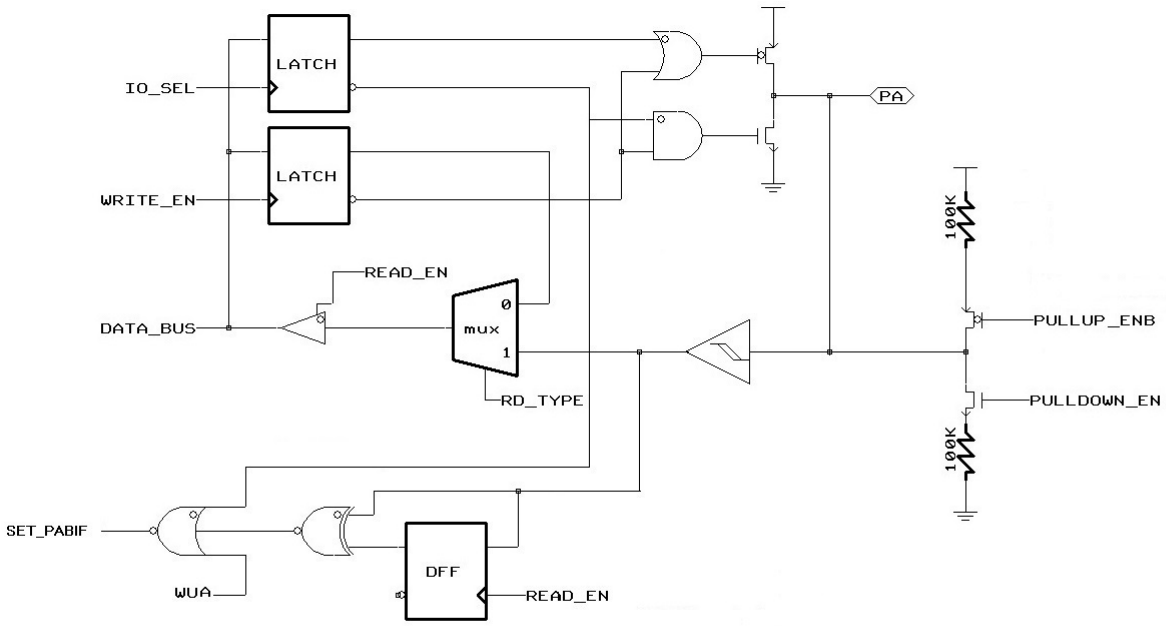


Figure 10 Block Diagram of PA4

RSTPAD_EN: enable PA5 as reset pin.

RSTB_IN: reset signal input.

IO_SEL: set pad attribute as input or output.

WRITE_EN: write data to pad.

READ_EN: read pad.

PULLUP_ENB: enable Pull-High.

PULLDOWN_EN: enable Pull-Low.

EIS2: external interrupt function enable.

EX_INT2: external interrupt signal.

RD_TYPE: select read pin or read latch.

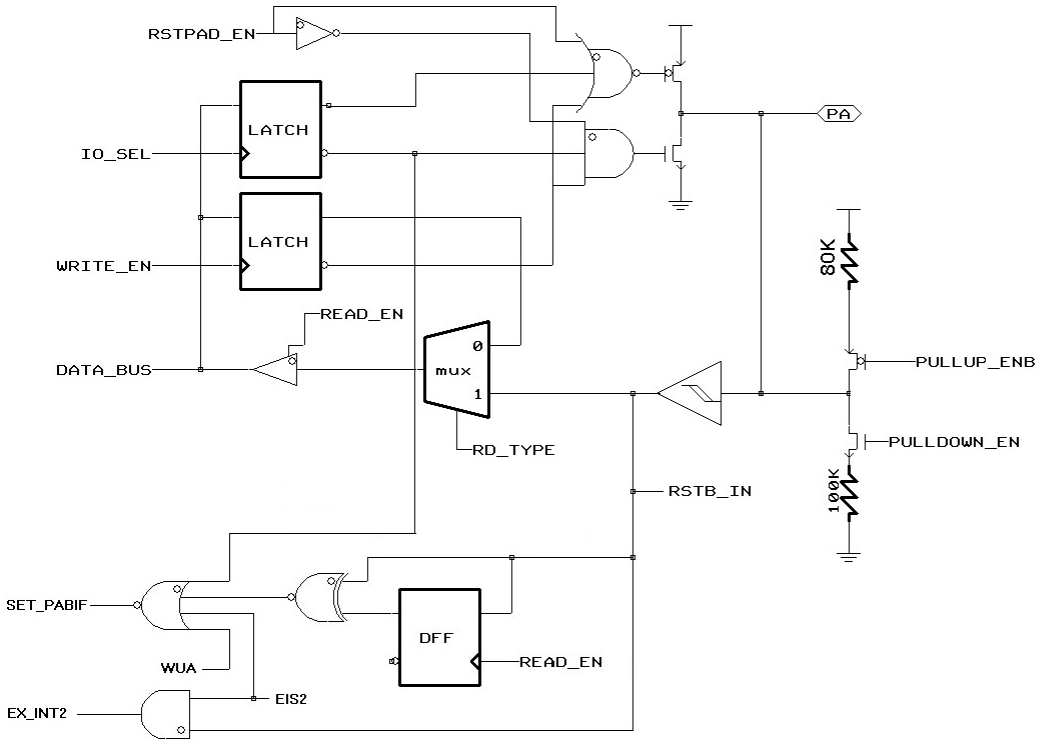


Figure 11 Block Diagram of PA5

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.

PULLUP_ENB: enable Pull-High.

RD_TYPE: select read pin or read latch.

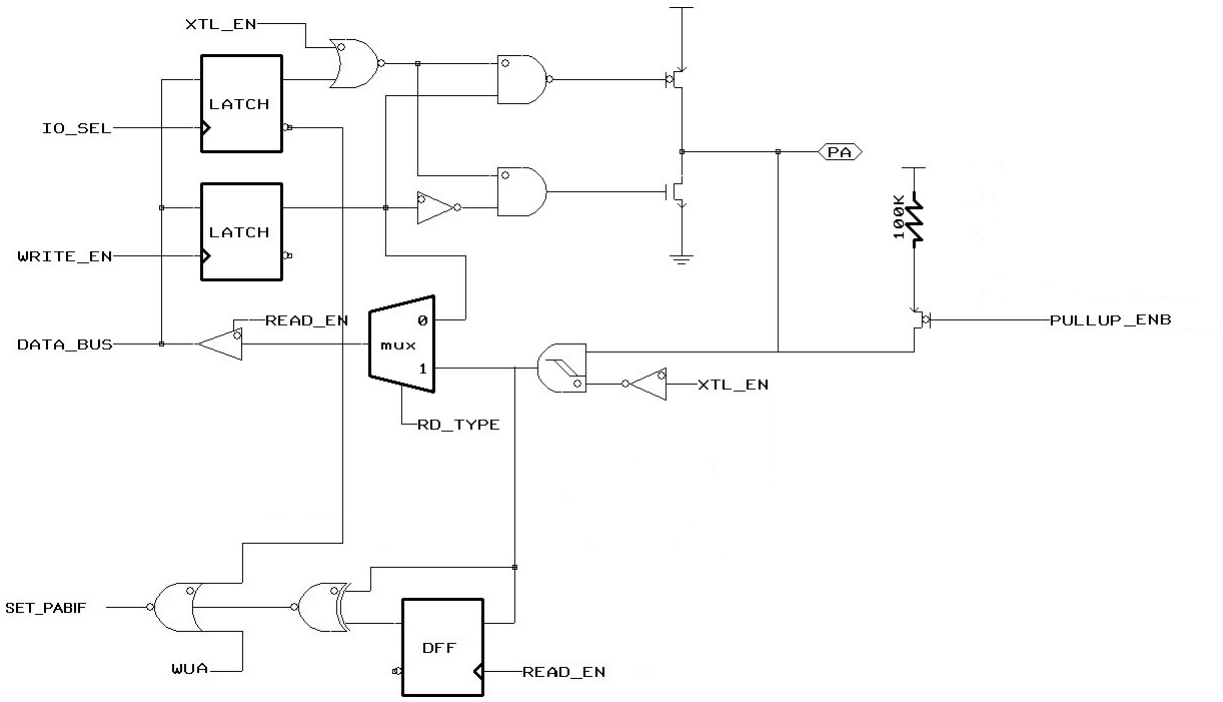


Figure 12 Block Diagram of PA6, PA7

IO_SEL: set pad attribute as input or output.

WRITE_EN: write data to pad.

READ_EN: read pad.

OD_EN: enable open-Drain.

PULLUP_ENB: enable Pull-High.

PULLDOWN_EN: enable Pull-Low.

RD_TYPE: select read pin or read latch.

WUB: port B wake-up enable.

SET_PBIF: port B wake-up flag.

CAP_EN: enable pad to external capacitor.

INITIAL: Discharge capacitor

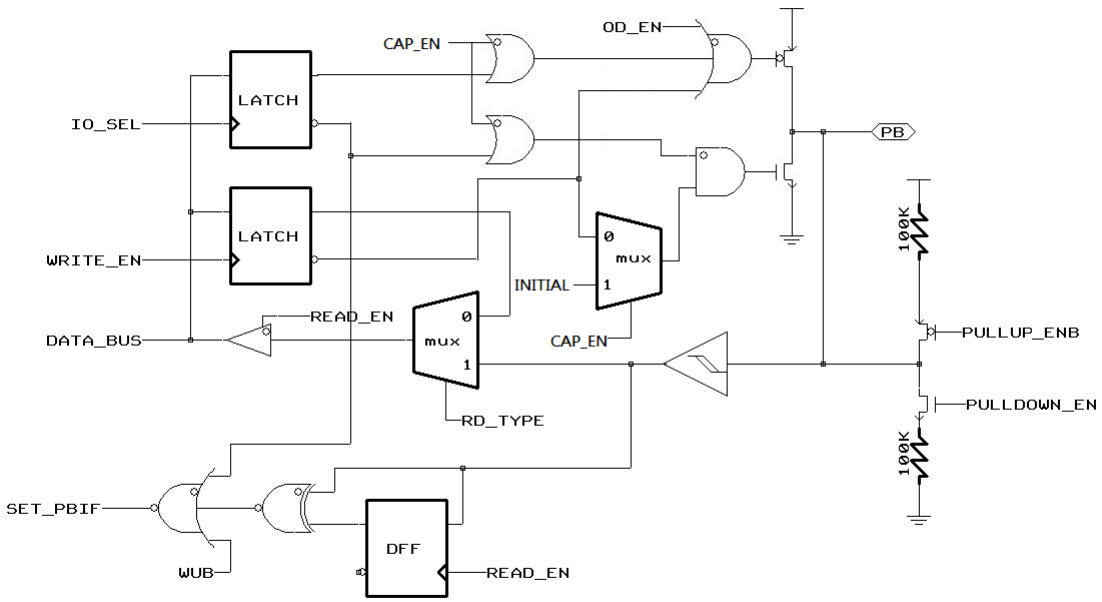


Figure 13 Block Diagram of PB0

IO_SEL: set pad attribute as input or output.

WRITE_EN: write data to pad.

READ_EN: read pad.

OD_EN: enable open-Drain.

PULLUP_ENB: enable Pull-High.

PULLDOWN_EN: enable Pull-Low.

RD_TYPE: select read pin or read latch.

WUB: port B wake-up enable.

SET_PBIF: port B wake-up flag.

EIS: external interrupt function enable.

EX_INT: external interrupt signal.

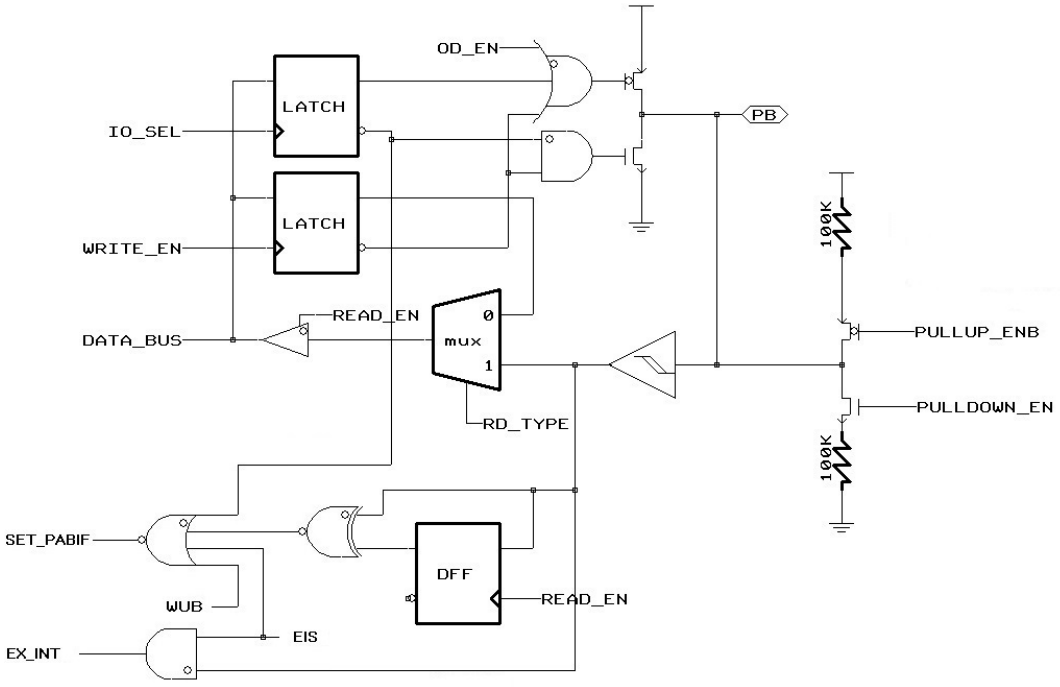


Figure 14 Block Diagram of PB1

IO_SEL: set pad attribute as input or output.

WRITE_EN: write data to pad.

READ_EN: read pad.

OD_EN: enable open-Drain.

PULLUP_ENB: enable Pull-High.

PULLDOWN_EN: enable Pull-Low.

RD_TYPE: select read pin or read latch.

WUB: port B wake-up enable.

SET_PBIF: port B wake-up flag.

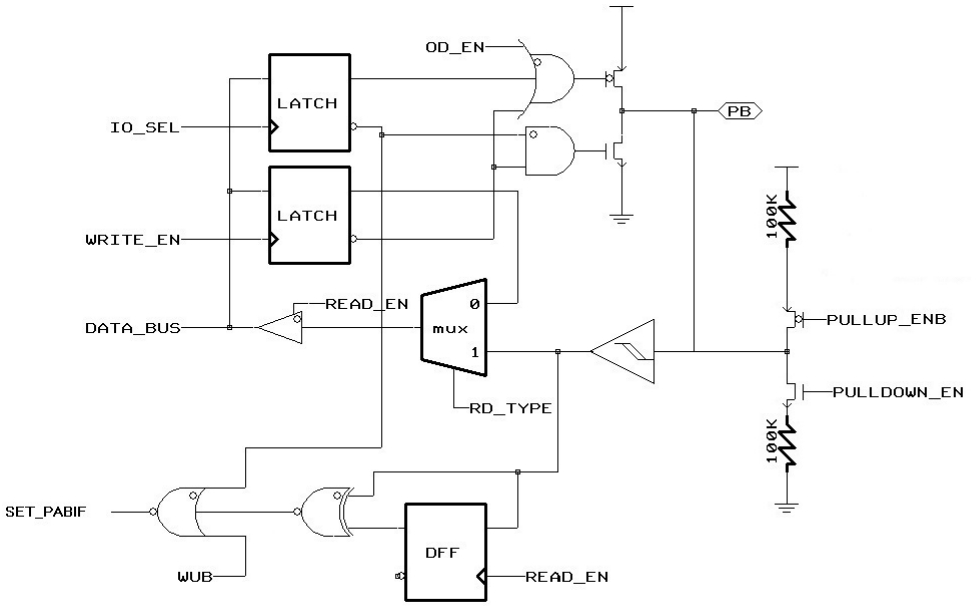


Figure 15 Block Diagram of PB2 & PB3

IO_SEL: set pad attribute as input or output.

WRITE_EN: write data to pad.

READ_EN: read pad.

OD_EN: enable open-Drain.

PULLUP_ENB: enable Pull-High.

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.

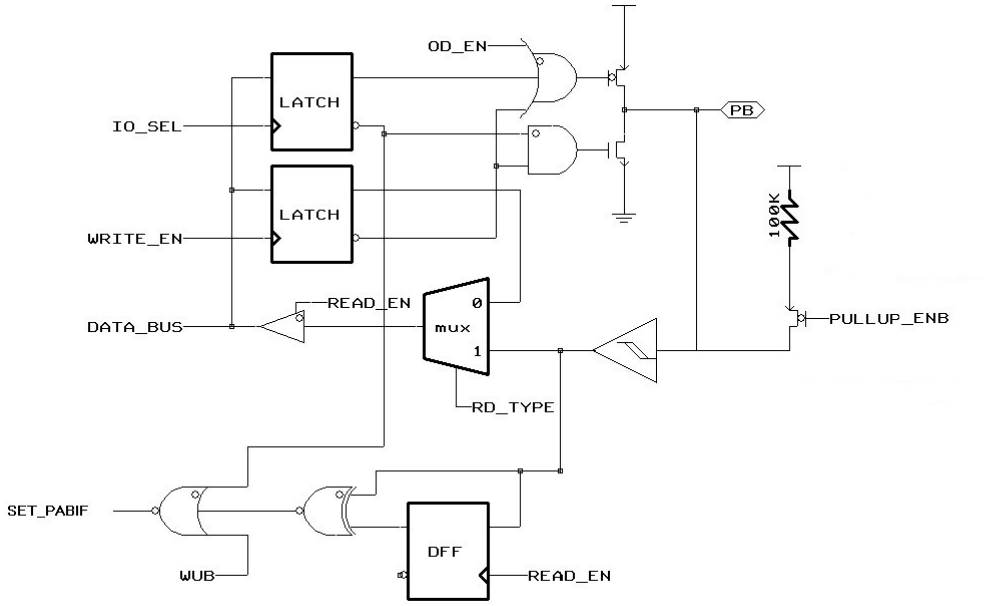


Figure 16 Block Diagram of PB4 & PB6 & PB7

IO_SEL: set pad attribute as input or output.

WRITE_EN: write data to pad.

READ_EN: read pad.

OD_EN: enable open-Drain.

PULLUP_ENB: enable Pull-High.

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.

EIS: external interrupt function enable.

EX_INT: external interrupt signal.

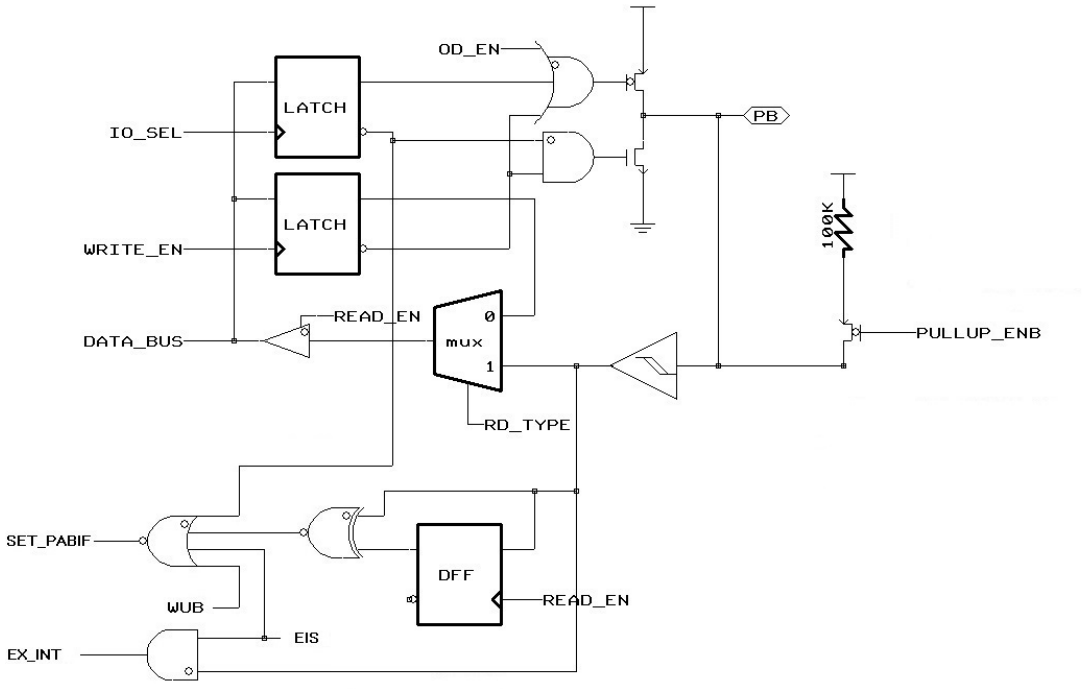


Figure 17 Block Diagram of PB5

IO_SEL: set pad attribute as input or output.

WRITE_EN: write data to pad.

READ_EN: read pad.

OD_EN: enable open-Drain.

PULLUP_ENB: enable Pull-High.

PULLDOWN_EN: enable Pull-Low.

RD_TYPE: select read pin or read latch.

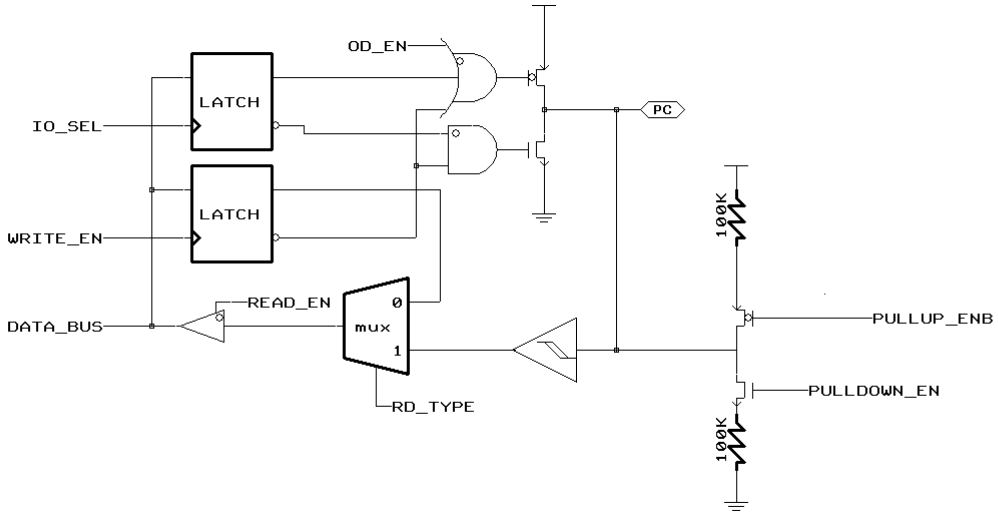


Figure 18 Block Diagram of PC0, PC1

3.7 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 Table 22)

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 22 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 Timer0 and WDT is shown in the figure below.

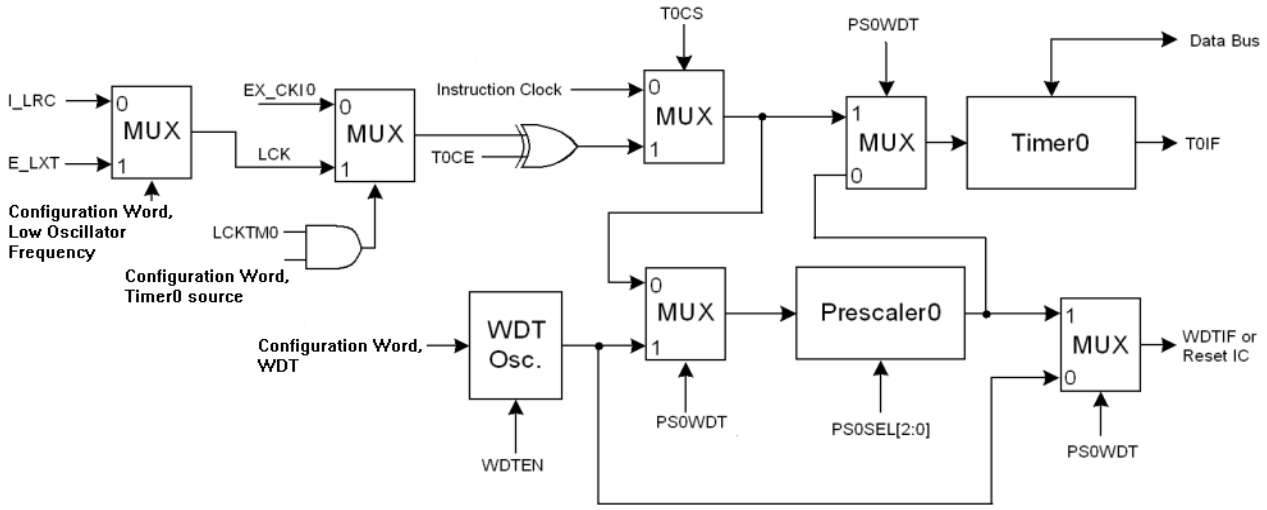


Figure 19 Block Diagram of Timer0 and WDT

3.8 Timer1 / PWM1 / Buzzer1/ PWM2 /PWM3

Timer1 is an 10-bit down-count timer with Prescaler1 whose dividing rate is programmable. The output of Timer1 can be used to generate PWM1 output/PWM2 output/PWM3 output and Buzzer1 output. Timer1 builds in auto-reload function and Timer1 reload register stores reload data with double buffers. When user write Timer1 reload register, write Timer1 MSB 2 bits(TMRH[5:4]) first and write TMR1 second, Timer1 reload register will be updated to Timer1 counter after Timer1 overflow occurs when T1EN=1. If T1EN=0, Timer1 reload register will be updated to Timer1 counter after write TMR1 immediately. A read to the Timer1 will show the content of the Timer1 current count value.

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

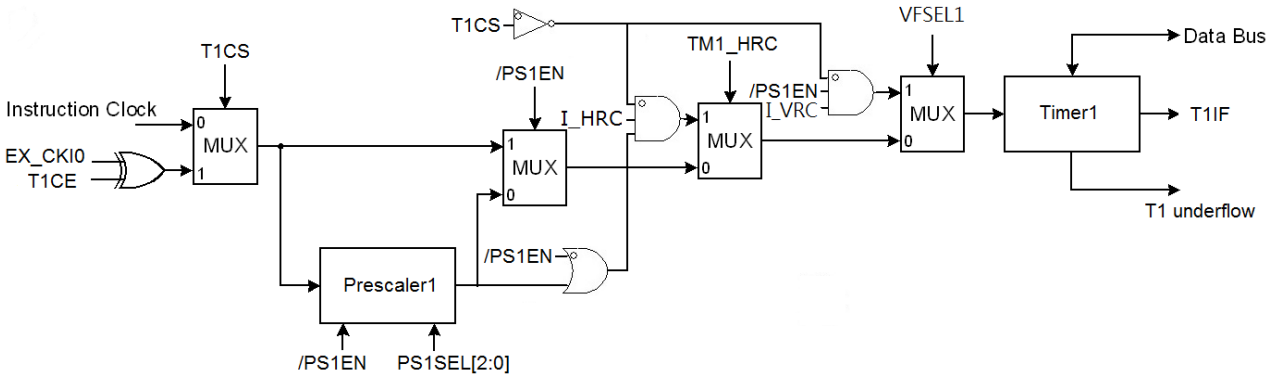


Figure 20 Block Diagram of Timer1

The operation of Timer1 can be enabled or disabled by register bit T1EN (T1CR1[0]). After Timer1 is enabled, its clock source can be instruction clock or pin EX_CKI0 which is determined by register bit T1CS (T1CR2[5]). When T1CS is 1, EX_CKI0 is selected as clock source. When T1CS is 0, instruction clock is selected as clock source. When EX_CKI0 is selected, the active edge to decrease Timer1 is determined by register bit T1CE

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

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

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

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

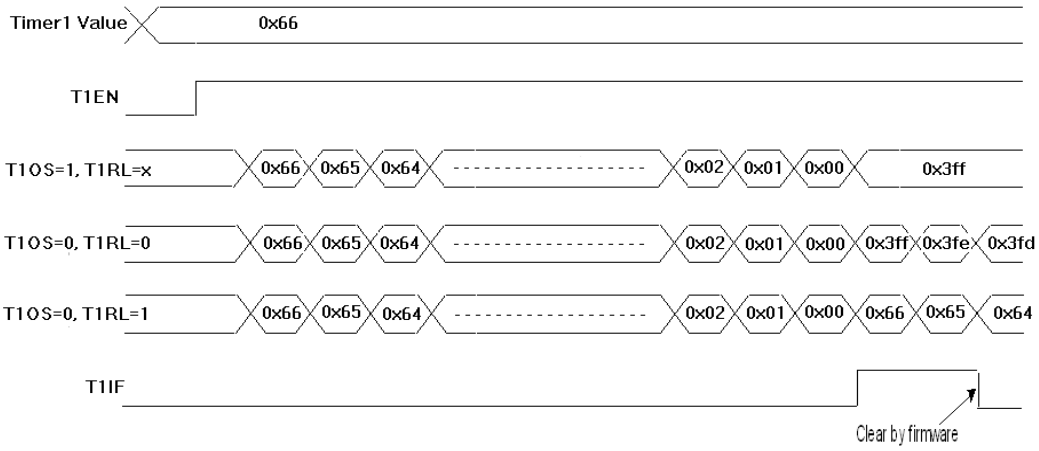


Figure 21 Timer1 Timing Chart

The PWM1 output can be available on I/O pin PB4 or PB1 which is decided by option ,when register bit PWM1OEN (T1CR1[7]) is set to 1. Moreover, PB4 or PB1 will become output pin automatically. The active state of PWM1 output is determined by register bit PWM1OAL (T1CR1[6]). When PWM1OAL is 1, PWM1 output is active low. When PWM1OAL is 0, PWM1 output is active high. Moreover, the duty cycle and frame rate of PWM1 are both programmable. The duty cycle is determined by registers TMRH[1:0] and PWM1DUTY[7:0]. When PWM1DUTY is 0, PWM1 output will be never active. When PWM1DUTY is 0x3FF, PWM1 output will be active for 1023 Timer1 input clocks. The frame rate is determined by TMRH[5:4] + TMR1[7:0] initial value. Therefore, PWM1DUTY value must be less than or equal to TMRH[5:4] + TMR1[7:0]. When user write

PWM1DUTY, write PWM1DUTY[9:8] MSB 2 bits(TMRH[1:0]) first and write PWM1DUTY[7:0] second, PWM1 duty register will be updated after Timer1 overflow occurs. The block diagram of PWM1 is illustrated in the following figure.

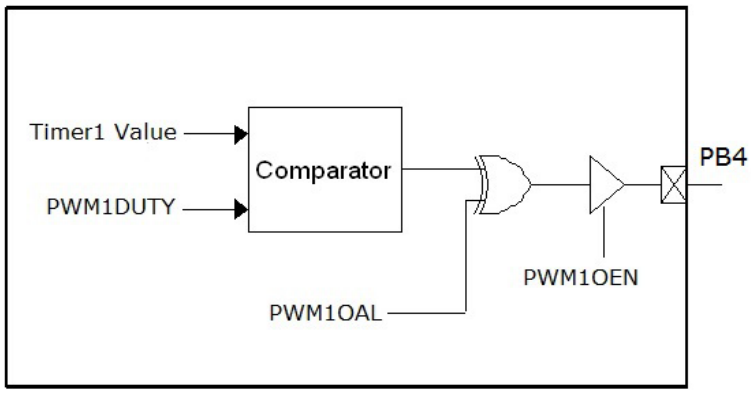


Figure 22 PWM1 Block Diagram

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

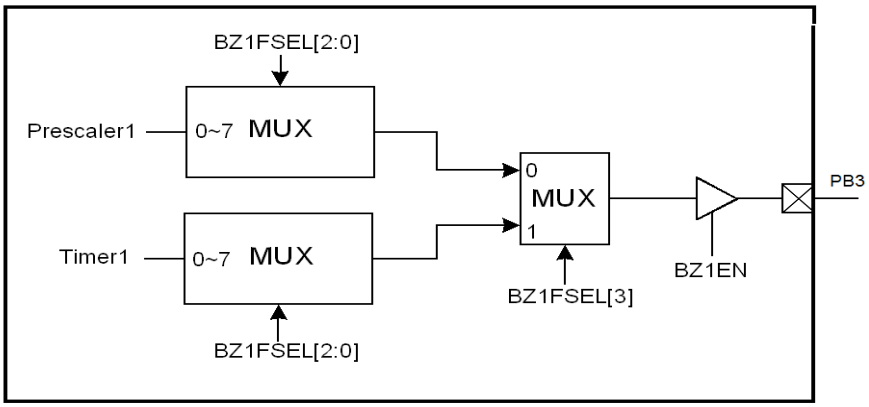


Figure 23 Buzzer1 Block Diagram

3.9 PWM2

The PWM2 output can be available on I/O pin PB3 or PB5 when register bit PWM2OEN (P2CR1[7]) is set to 1. Moreover, PB3 or PB5 will become output pin automatically. The active state of PWM2 output is determined by register bit PWM2OAL (P2CR1[6]). When PWM2OAL is 1, PWM2 output is active low. When PWM2OAL is 0, PWM2 output is active high. Moreover, the duty cycle and frame rate of PWM2 are both programmable. The duty cycle is determined by register TMRH[3:2],PWM2DUTY[7:0]. When PWM2DUTY is 0, PWM2 output will be

never active. When PWM2DUTY is 0x3FF, PWM2 output will be active for 1023 Timer1 input clocks. The frame rate is determined by TMRH[5:4],TMR1[7:0] initial value. Therefore, PWM2DUTY value must be less than or equal to TMR1[9:0]. When user write PWM2DUTY, write PWM2DUTY[9:8] MSB 2 bits(TMRH[3:2]) first and write PWM2DUTY[7:0] second, PWM2 duty register will be updated after Timer1 overflow occurs. The block diagram of PWM2 is illustrated in the following figure.

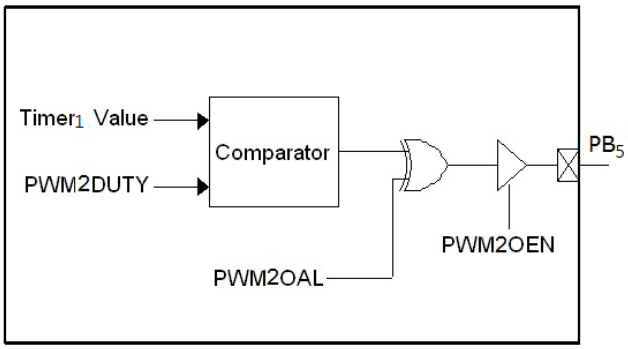


Figure 24 PWM2 Block Diagram

3.10 PWM3

The PWM3 output can be available on I/O pin PA2 or PA7 which is decided by option, when register bit PWM3OEN (P3CR1[7]) is set to 1. Moreover, PA2 or PA7 will become output pin automatically. The active state of PWM3 output is determined by register bit PWM3OAL (P3CR1[6]). When PWM3OAL is 1, PWM3 output is active low. When PWM3OAL is 0, PWM3 output is active high. Moreover, the duty cycle and frame rate of PWM3 are both programmable. The duty cycle is determined by register TM4RH[1:0],PWM3DUTY[7:0]. When PWM3DUTY is 0, PWM3 output will be never active. When PWM3DUTY is 0x3FF, PWM3 output will be active for 1023 Timer1 input clocks. The frame rate is determined by TMRH[5:4],TMR1[7:0] initial value. Therefore, PWM3DUTY value must be less than or equal to TMR1[9:0]. When user write PWM3DUTY, write PWM3DUTY[9:8] MSB 2 bits(TM4RH[1:0]) first and write PWM3DUTY[7:0] second, PWM3 duty register will be updated after Timer1 overflow occurs. The block diagram of PWM3 is illustrated in the following figure.

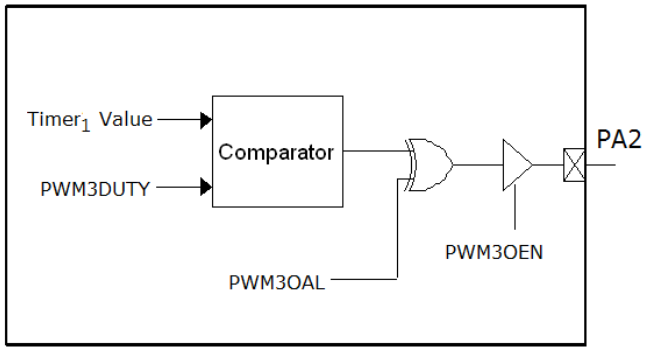


Figure 25 PWM3 Block Diagram

3.11 Timer4 / PWM4

Timer4 is an 10-bit down-count timer with Prescaler4 whose dividing rate is programmable. The output of Timer4 can be used to generate PWM4 output . Timer4 builds in auto-reload function and Timer4 reload register stores reload data with double buffers. When user write Timer4 reload register, write Timer4 MSB 2 bits(TM4RH[7:6]) first and write TMR4 second, Timer4 reload register will be updated to Timer4 counter after Timer4 overflow occurs when T4EN=1. If T4EN=0, Timer4 reload register will be updated to Timer4 counter after write TMR4 immediately. A read to the Timer4 will show the content of the Timer4 current count value. The block diagram of Timer4 is shown in the figure below.

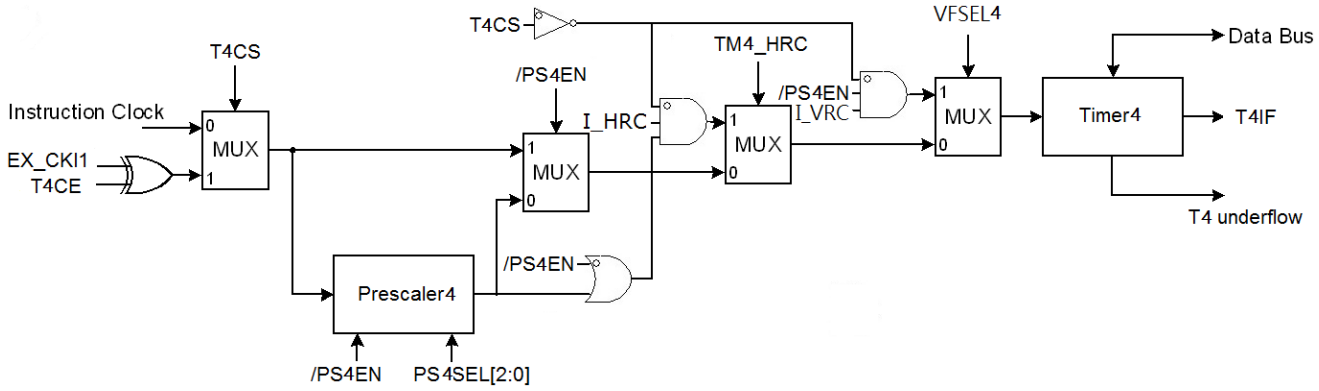


Figure 26 Block Diagram of Timer4

The operation of Timer4 can be enabled or disabled by register bit T4EN (T4CR1[0]). After Timer4 is enabled, its clock source can be instruction clock, pin EX_CKI1 or I_HRC output, which is determined by register bit T4CS (T4CR2[5]) and TM4_HRC(T4CR1[3]). When T4CS is 1 and TM4_HRC is 0, EX_CKI1 is selected as clock source. When T4CS is 0 and TM4_HRC is 0, instruction clock is selected as clock source. When TM4_HRC is 1, I_HRC output is selected as clock source. When EX_CKI1 is selected, the active edge to decrease Timer4 is determined by register bit T4CE (T4CR2[4]). When T4CE is 1, high-to-low transition on EX_CKI1 will decrease Timer1. When T4CE is 0, low-to-high transition on EX_CKI1 will decrease Timer4. The selected clock source can be divided further by Prescaler4 before it is applied to Timer4. Prescaler4 is enabled by writing 0 to register bit /PS4EN (T4CR2[3]) and the dividing rate is from 1:2 to 1:256 determined by register bits PS4SEL[2:0] (T4CR2[2:0]). Current value of Prescaler4 can be obtained by reading register PS4CV.

Timer4 provides two kinds of operating mode: one is One-Shot mode and the other is Non-Stop mode. When register bit T4OS (T4CR1[2]) is 1, One-Shot mode is selected. Timer4 will count down once from initial value stored on register TMR4[9:0] to 0x00, i.e. underflow is occurred. When register bit T4OS (T4CR1[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 T4RL (T4CR1[1]). When T4RL is 1, the initial value stored on register TMR4[9:0] will be restored and start next down-count from this initial value. When T4RL is 0, Timer1 will start next down-count from 0x3FF.

When Timer4 is underflow, the register bit T4IF (INTE2[6]) will be set to 1 to indicate Timer4 underflow event is occurred. If register bit T4IE (INTE2[2]) and GIE are both set to 1, interrupt request will occur and interrupt service routine will be executed. T4IF will not be clear until firmware writes 0 to T4IF.

The timing chart of Timer4 is shown in the following figure.

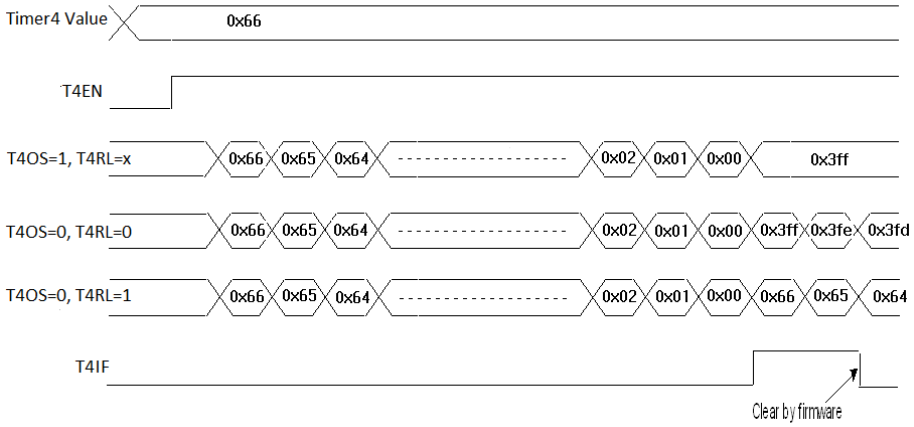


Figure 27 Timer4 Timing Chart

The PWM4 output is available on I/O pin PA4. When register bit PWM4OEN (T4CR1[7]) is set to 1, the corresponded PWM4 pin will became output pin automatically. The active state of PWM4 output is determined by register bit PWM4OAL (T4CR1[6]). When PWM4OAL is 1, PWM4 output is active low. When PWM4OAL is 0, PWM4 output is active high. Moreover, the duty cycle and frame rate of PWM4 are both programmable. The duty cycle is determined by registers TM4RH[3:2] and PWM4DUTY[7:0]. When PWM4DUTY is 0, PWM4 output will be never active. When PWM4DUTY is 0x3FF, PWM4 output will be active for 1023 Timer4 input clocks. The frame rate is determined by TM4RH[7:6] + TMR4[7:0] initial value. Therefore, PWM4DUTY value must be less than or equal to TM4RH[7:6] + TMR4[7:0]. When user write PWM4DUTY, write PWM4DUTY[9:8] MSB 2 bits(TM4RH[3:2]) first and write PWM4DUTY[7:0] second, PWM4 duty register will be updated after Timer4 overflow occurs. The block diagram of PWM4 is illustrated in the following figure.

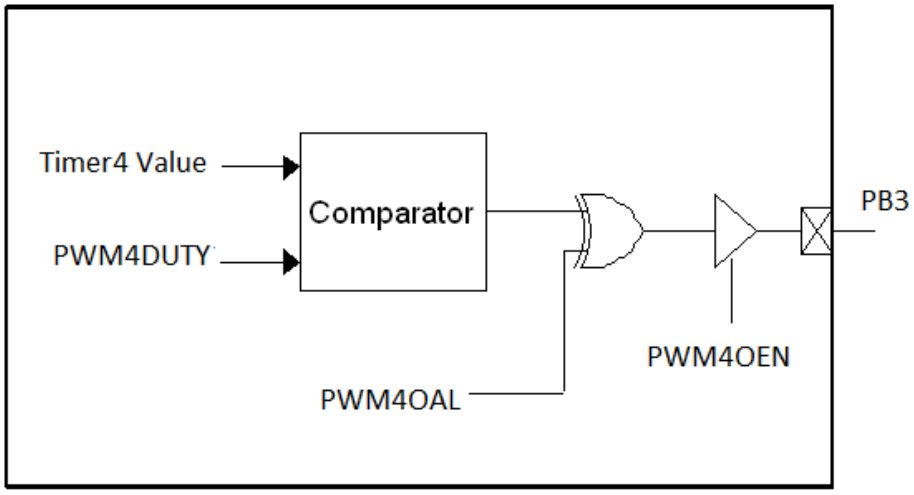


Figure 28 PWM4 Block Diagram

3.12 RFC Mode

NY8TE64A has built-in RFC mode. Once RFC mode is enabled, the selected input pad state will take control of the Timer1 counting. When the selected input pad is recognized as 0 state (The input pad voltage is smaller than V_{IL}), Timer1 keeps counting. When this selected pad is recognized as 1 (The input pad voltage is larger than V_{IH}), Timer1 stops counting. The following figure shows how RFC mode operates: PSEL3~0 is used to select one RFC input pad out of 16 NY8TE64A pads. RFCEN is used to switch the Timer1 enable signal between the normal enable signal T1EN and RFC selected input state.

One application of RFC mode is to measure the capacitor-resistor charging time, As the figure shows, when PSEL3~0=0x01, PA1 is selected as RFC input pad. At first the PA1 is set as output low (the voltage of PA1 is discharged to 0). Next step, clear Timer1 content, set PA1 as input and enable RFC mode. Then Timer1 will start counting, and the RC circuit will start charging PA1. As PA1 is charged to the V_{IH} voltage, the Timer1 counting is stopped because PA1 input is high. The Timer1 content will show the RC circuit charging time. (Note: Timer1 is down-count.)

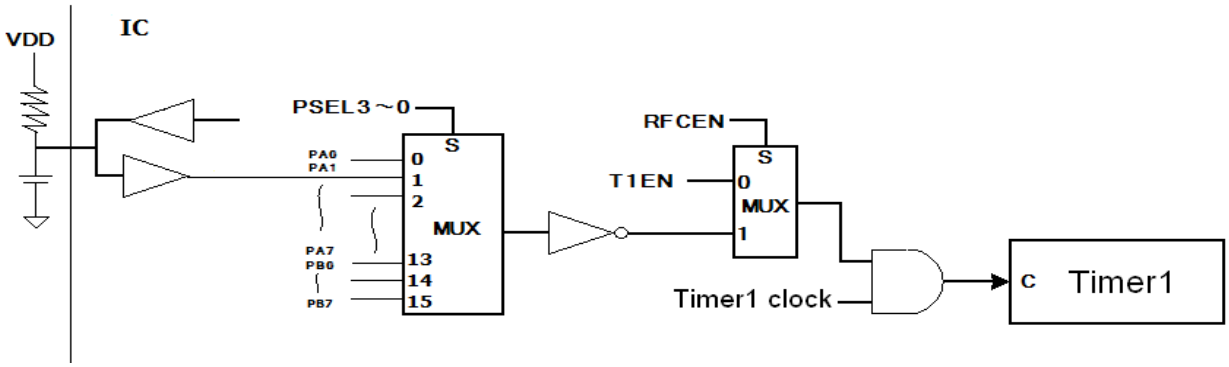


Figure 32 RFC Block Diagram

3.13 IR Carrier

According to the IR pad configuration, the IR output pad of NY8TE64A can be PB1 or PA3. The IR carrier will be generated after register bit IREN (IRCR[0]) is set to 1. Moreover, PB1 or PA3 will become output pin automatically. When IREN is clear to 0, PB1 or PA3 will become general I/O pin as it was configured.

The IR carrier will be generated after register bit IREN (IRCR[0]) is set to 1. Moreover, PB1 or PA3 will become output pin automatically. When IREN is clear to 0, PB1 or PA3 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 NY8TE64A 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 or PA3 output data. When register bit IRCSEL (IRCR[2]) is 1, IR carrier will be present on pin PB1 or PA3 when its output data is 0. When register bit IRCSEL (IRCR[2]) is 0, IR carrier will be present on pin PB1 or PA3 when its output data is 1. The polarity of IR carrier is shown in the following figure.

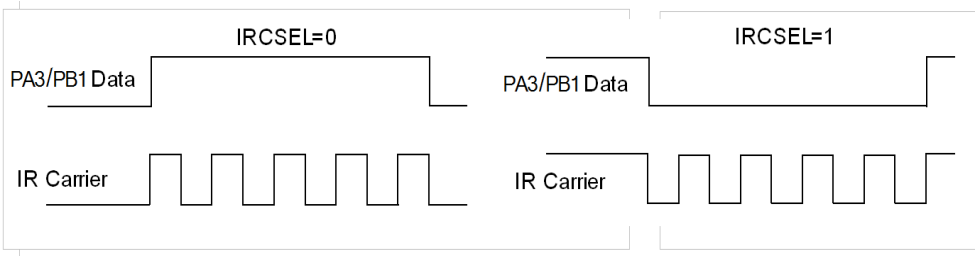


Figure 33 Polarity of IR Carrier vs. Output Data

3.14 Low Voltage Detector (LVD)

NY8TE64A low voltage detector (LVD) built-in precise band-gap reference for accurately detecting V_{DD} level. If LVDEN(register PCON[5])=1 and V_{DD} voltage value falls below LVD voltage which is selected by LVDS[3:0] as table shown below, the LVD output will become low. If the LVD interrupt is enabled, the LVD interrupt flag will be high and if GIE=1 it will force the program to execute interrupt service routine. Moreover, LVD real-state output can be polled by register PCON1[6]. The following is LVD block diagram:

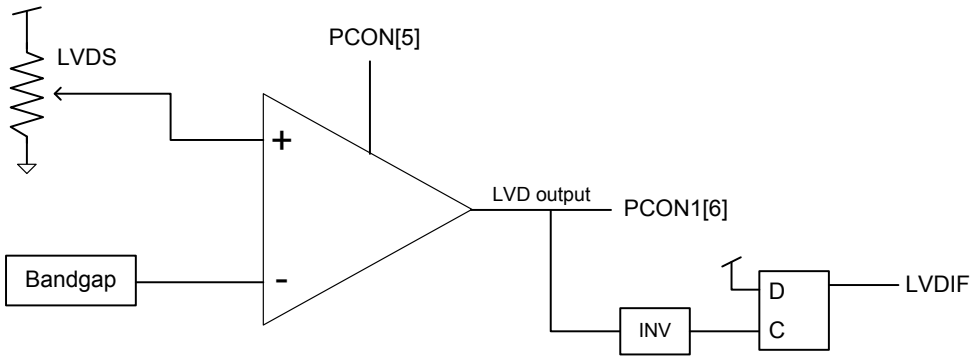


Figure 34 LVD block diagram

The following table is LVD voltage select table.

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

Table 23 LVD voltage select

Note:

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

2. In battery charging applications (detected voltage is from low to high), the LVD voltage select table should be as followed:

LVDS[3:0]	Voltage
0000	--
0001	--

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

The LVD control flow is as the following:

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

Step2: Set CMPCR = 0x0A

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

Step4: Check LVD status by PCON1[6]

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

3.15 Voltage Comparator

NY8TE64A 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 ANAEN[7]) is used to enable and disable comparator. When CMPEN=0(default), comparator is disabled. When CMPEN=1, the comparator is enabled. In halt mode the comparator is disabled automatically.

The structure of comparator is shown in the following figure:

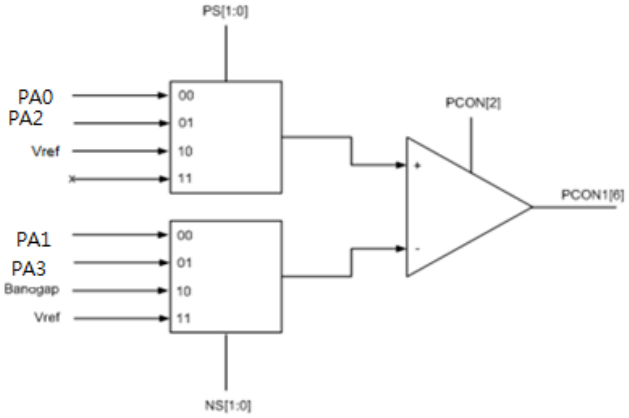


Figure 35 Vref hardware connection

3.15.1 Comparator Reference Voltage (Vref)

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

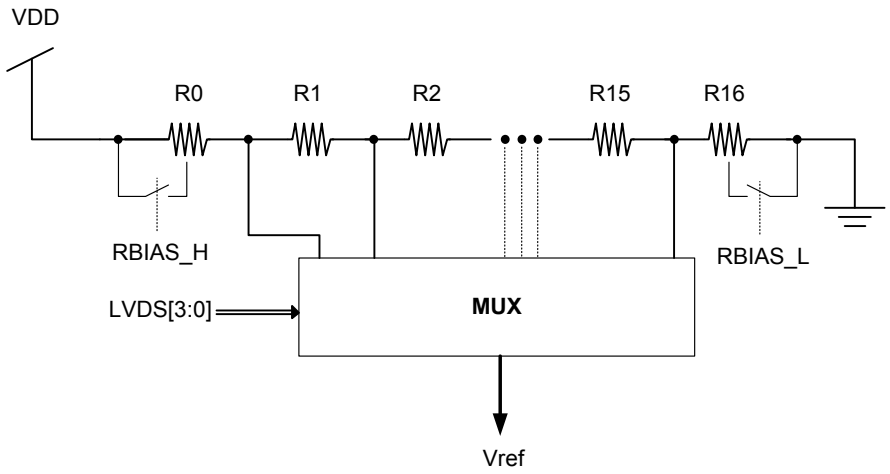


Figure 36 Vref hardware connection

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

LVDS[3:0]	RBIAS_H=1 RBIAS_L=0	RBIAS_H=0 RBIAS_L=1
0000	65/128 V_{DD}	31/128 V_{DD}
0001	62/128 V_{DD}	29/128 V_{DD}
0010	56/128 V_{DD}	26/128 V_{DD}

LVDS[3:0]	RBIAS_H=1 RBIAS_L=0	RBIAS_H=0 RBIAS_L=1
0011	52/128 V_{DD}	23/128 V_{DD}
0100	48/128 V_{DD}	20/128 V_{DD}
0101	44/128 V_{DD}	18/128 V_{DD}
0110	43/128 V_{DD}	17/128 V_{DD}
0111	41/128 V_{DD}	16/128 V_{DD}
1000	39/128 V_{DD}	14/128 V_{DD}
1001	37/128 V_{DD}	13/128 V_{DD}
1010	35/128 V_{DD}	12/128 V_{DD}
1011	34/128 V_{DD}	11/128 V_{DD}
1100	32/128 V_{DD}	10/128 V_{DD}
1101	31/128 V_{DD}	9/128 V_{DD}
1110	30/128 V_{DD}	8/128 V_{DD}
1111	29/128 V_{DD}	7/128 V_{DD}

Table 13 The reference voltage Vref selection table

Note: The deviation of Vref is ±0.1V.

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

The table is shown below

PS[1:0]	Non-inverting input
00	PA0
01	PA2
10	Vref
11	---

Table 14 Non-inverting input select

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

The table is shown below

NS[1:0]	Inverting input
00	PA1
01	PA3
10	Bandgap (0.6V)

11	Vref

Table 15 Inverting input select

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

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

To probe comparator output at output pad, set CMPOE (register OSCCR[6]) to 1, then PB3 will be the real-time state of the comparator output. It is noted that when CMPOE=1, the PWM3 function will be disabled if it is enabled.

3.16 Analog-to-Digital Convertor (ADC)

NY8TE64A provide 12+2 channel 12-bit SAR ADC to transfer analog signal into 12-bits digital data. The ADC high reference voltage is selectable. They can be external voltage from PA0, or internal generated voltage VDD, 4V, 3V or 2V. The Analog input is selected from analog signal input pin PA0~PA4, PB1~PB7 or from internal generated 1 / 4 *VDD. The ADC clock ADCLK can be selected to be Fcpu/1, Fcpu/2, Fcpu/8 or Fcpu/16. The Sampling pulse width can be selected to be ADCLK*1, ADCLK*2, ADCLK*4 or ADCLK*8. Set ADEN=1 before ADC take into operation. Then set START=1, the ADC will start to convert analog signal to digital. EOC=0 means ADC is in processing. EOC=1 indicate ADC is at end of conversion. If ADIE=1 and global interrupt is enabled, the ADC interrupt will issue after EOC low go high. The block diagram is as following figure.

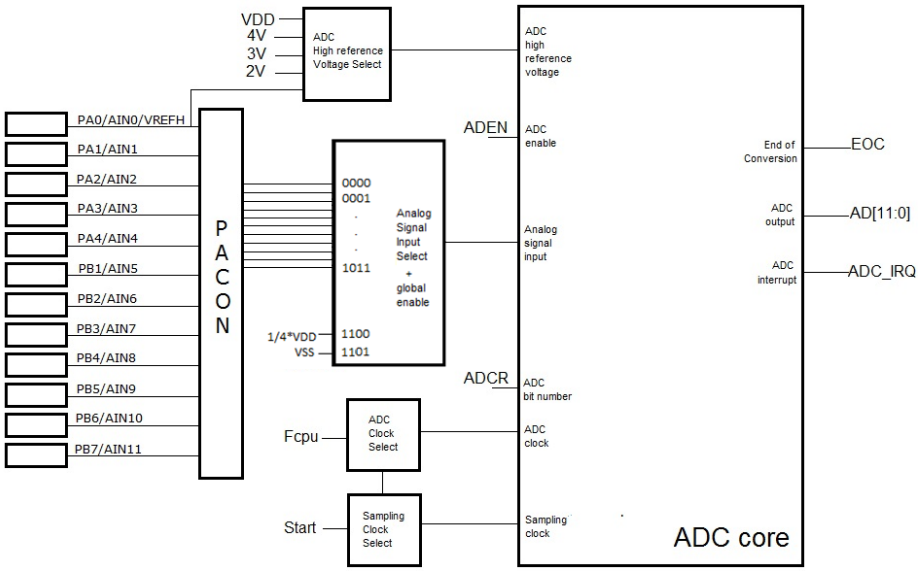


Figure 37 ADC block diagram

3.16.1 ADC reference voltage

ADC is built-in five high reference voltage source controlled by ADVREFH register. These high reference voltage source are one external voltage source (PA0) and four internal voltage source (VDD, 4V, 3V, 2V).

When EVHENB bit is “1”, ADC reference voltage is external voltage source from PA0. In this mode PA0 must input a voltage between VDD and 2V. If EVHENB bit is 0, ADC reference voltage is from internal voltage source selected by VHS[1:0]. If VHS[1:0] is “11”, ADC reference voltage is VDD. If VHS[1:0] is “10”, ADC reference voltage is 4V. If VHS[1:0] is “01”, ADC reference voltage is 3V. If VHS[1:0] is “00”, ADC reference voltage is 2V. The limitation of internal reference voltage application is VDD can't below each of internal voltage level, or the level is equal to VDD. ADC sampling voltage range is limited by high/low reference voltage. The ADC low reference voltage is VSS and not changeable. The ADC high reference voltage includes internal VDD/4V/3V/2V and external reference voltage source from PA0 pin. The ADC reference voltage range limitation is (ADC high reference voltage – low reference voltage) ≥ 2V. ADC low reference voltage is VSS=0V. So ADC high reference voltage range is 2V ~ VDD.

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

EVHENB	VHS[1:0]	Reference voltage
1	x x	PA0
0	1 1	VDD
0	1 0	4V
0	0 1	3V
0	0 0	2V

Table 27 ADC reference voltage select

3.16.2 ADC analog input channel

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

GCHS	CHS[3:0]	ADC analog input source
0	x x x x	x
1	0 0 0 0	PA0
1	0 0 0 1	PA1
1	0 0 1 0	PA2
1	0 0 1 1	PA3
1	0 1 0 0	PA4
1	0 1 0 1	PB1
1	0 1 1 0	PB2
1	0 1 1 1	PB3
1	1 0 0 0	PB4
1	1 0 0 1	PB5
1	1 0 1 0	PB6

GCHS	CHS[3:0]	ADC analog input source
1	1 0 1 1	PB7
1	1 1 0 0	1 / 4 * VDD
1	1 1 0 1	VSS
1	1 1 1 X	N.C.

Table 28 ADC analog input source select

ADC input pins are shared with digital I/O pins. Connect an analog signal to these pin may cause extra current leakage in I/O pins. In the power down mode, the above leakage current will be a big problem. Register bit PACONx/PBCONx is PAx/PBx configuration register bit to solve above problem. Write “1” to PACONx/ PBCONx register bit will configure related PAx/PBx pin as pure analog input pin to avoid current leakage, and it can't be use as normal I/O.

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

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

Conversion speed and conversion accuracy are affected by the selection of the ADC clock (ADCLK), sampling pulse width (SHCLK) and conversion bit number. ADCLK is the base clock of ADC. During the operation of SAR ADC, bit operation is synchronized with ADCLK. SHCLK is the duration of analog signal sampling time, larger SHCLK will restore original analog signal level more closely but it will slow down the ADC conversion speed. Vise versa. The ADC can select different conversion bit number which is depended on ADCR[1:0] register bits. There are 2 bit number to select, which is 12-bit, 10-bit and 8-bit. Less conversion bit number will speed up the ADC conversion rate but the effective ADC bit is less. More conversion bit number will slow down the conversion rate but the accuracy is more.

The ADC clock is derived from Fcpu and is selectable from ADCK[1:0].

ADCK[1:0]	ADC clock
0 0	Fcpu/16
0 1	Fcpu/8
1 0	Fcpu/1
1 1	Fcpu/2

Table 29 ADC clock select

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

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

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

Table 30 ADC sampling clock select

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

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

Table 31 conversion bit number select

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

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

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

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

Table 32 ADC Conversion time

3.16.4 ADC offset calibration

ADC offset error is defined as the deviation between the first ideal code transition and the first actual code transition. The first ideal code transition take place at 0.5LSB. ADC offset error varies with temperature, process and voltage. ADC offset error can be real-time adjusted in NY8BE62D through ADJMD register. Nyquest provides the NYIDE example code “ADC_Interrupt_AutoK” for ADC offset calibration process.

3.16.5 ADC operation

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

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

3.17 Watch-Dog Timer (WDT)

There is an on-chip free-running oscillator in NY8TE64A 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 NY8TE64A 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 NY8TE64A and dividing rate is from 1:2 to 1:256 if WDT time-out will interrupt NY8TE64A.

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.

3.18 Interrupt

NY8TE64A 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 12 hardware interrupts:

- Timer0 overflow interrupt.
- Timer1 underflow interrupt.
- Timer4 underflow interrupt.
- WDT timeout interrupt.
- PA/PB input change interrupt.
- External 0 interrupt.
- External 1 interrupt
- External 2 interrupt
- LVD interrupt.
- Comparator output status change interrupt.
- ADC end-of-convert interrupt.
- SIM interrupt.(serial interface mode 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 NY8TE64A automatically. This will prevent nested interrupt from happening. The last instruction of interrupt service routine of software interrupt has to be RETIE. Execution of this instruction will set GIE to 1 and return to original execution sequence.

While any of hardware interrupts is occurred, the corresponding bit of interrupt flag will be set to 1. This bit will not be clear until firmware writes 0 to this bit. Therefore user can obtain information of which event causes hardware interrupt by polling the corresponding bit of interrupt flag. Note that only when the corresponding interrupt enable bit is set to 1, will the corresponding interrupt flag be read. And if the corresponding interrupt enable bit is set to 1 and GIE is also 1, hardware interrupt will occur and next instruction will be fetched from 0x008. At the same time, the register bit GIE will be clear by NY8TE64A 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 NY8TE64A 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.

3.18.1 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.

3.18.2 Timer1 Underflow Interrupt

Timer1 underflow (from 0x3FF to 0x00) will set register bit T1IF. This interrupt request will be serviced if T1IE and GIE are set to 1.

3.18.3 Timer4 Underflow Interrupt

Timer4 underflow (from 0x3FF to 0x00) will set register bit T4IF. This interrupt request will be serviced if T4IE and GIE are set to 1.

3.18.4 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.

3.18.5 PA/PB Input Change Interrupt

When PAx, 0 ≤ x ≤ 7, PBy, 0 ≤ y ≤ 7 is configured as input pin and corresponding register bit WUPAx, WUPBx is set to 1, a level change on these selected I/O pin(s) will set register bit PABIF. This interrupt request will be serviced if PABIE and GIE are set to 1. Note when PA3, PA4 or PA5 is both set as level change interrupt and external interrupt, the external interrupt enable EIS0, EIS1 or EIS2=1 will disable PA3, PA4 or PA5 level change operation.

3.18.6 External 0 Interrupt

According to the configuration of EIS0=1 and INTEDG, the selected active edge on I/O pin PA4 will set register bit INT0IF and this interrupt request will be served if INT0IE and GIE are set to 1.

3.18.7 External 1 Interrupt

According to the configuration of EIS1=1 and INTEDG, the selected active edge on I/O pin PA3 will set register bit INT1IF and this interrupt request will be served if INT1IE and GIE are set to 1.

3.18.8 External 2 Interrupt

According to the configuration of EIS2=1 and INTEDG, the selected active edge on I/O pin PA5 will set register bit INT2IF and this interrupt request will be served if INT2IE and GIE are set to 1.

3.18.9 LVD Interrupt

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

3.18.10 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.

3.18.11 ADC end of conversion Interrupt

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

3.18.12 EEPROM write complete Interrupt

When EEPROM write is completed, the EEPROM write complete flag is issued. This interrupt request will be serviced if EEIE and GIE are set to 1.

3.18.13 SERAIL interface mode interrupt

The SIM interrupt is triggered whenever an SPIF of spi mode or MIFof iic mode is issued. This interrupt request will be serviced if SIMIE and GIE are set to 1.

3.19 Oscillation Configuration

Because NY8TE64A 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).

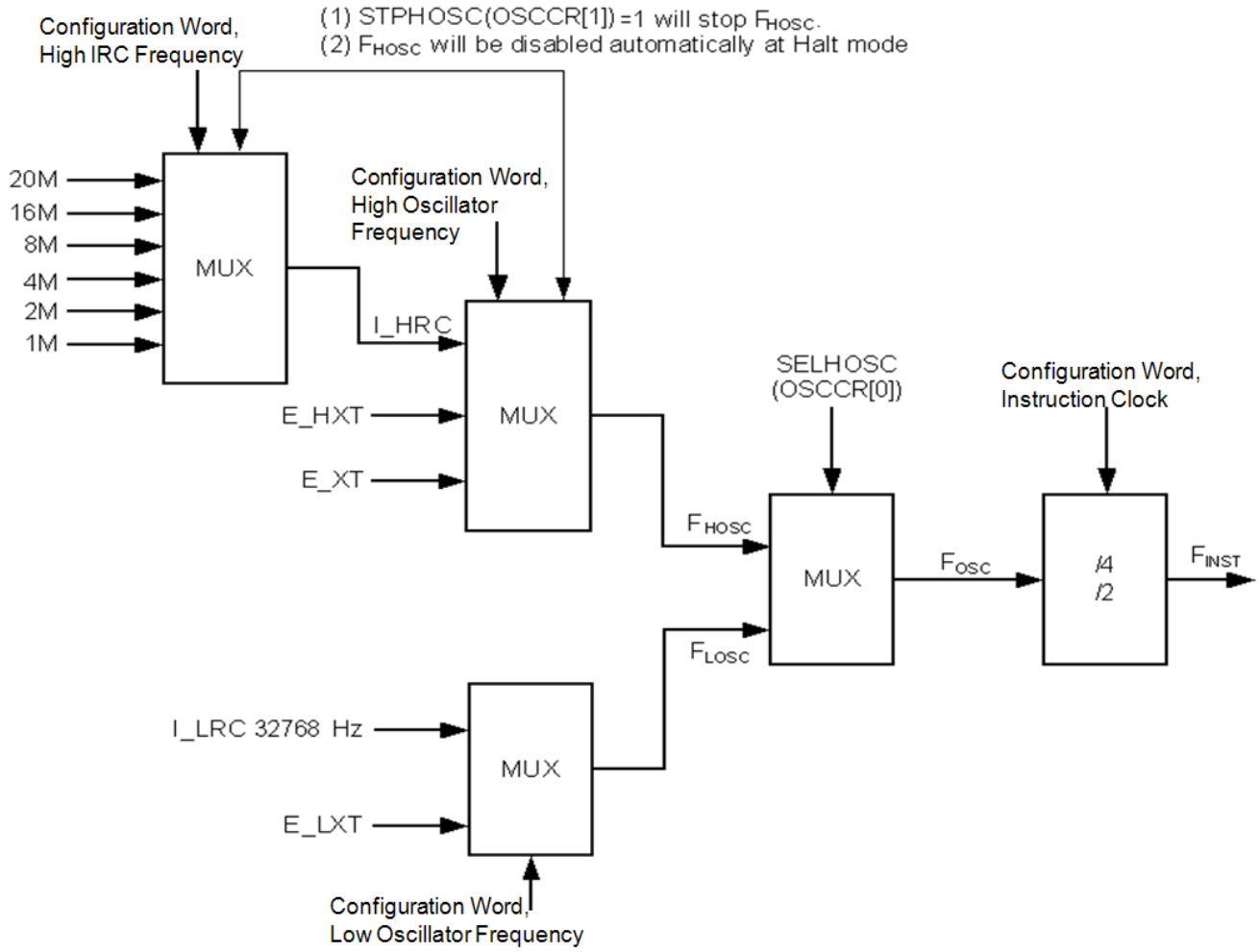


Figure 38 Oscillation Configuration of NY8TE64A

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 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).

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 33 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 34 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.

To get precise and stable 32.768k frequency, choosing the right C1 and C2 value is important. You need to match the C1 / C2 capacitance to the specific crystal you chose. Every crystal datasheet lists something called the Load Capacitance (CL), C1 and C2 value is chosen with the following formula:

C1=C2=2*CL-Cbt

Where Cbt is the NY8TE64A crystal pad built-in capacitance, which is about 5pF. For example, for crystal CL=12.5P, C1=C2=20pF is recommended.

The accuracy of I_HRC is ±1% at 25°C commercial conditions.

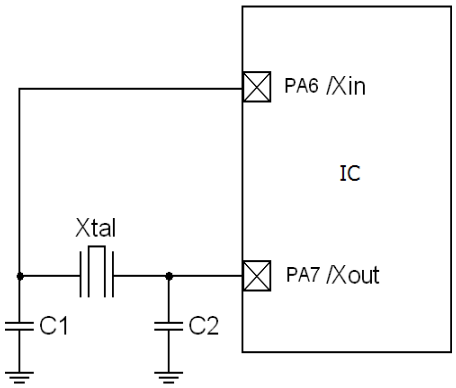


Figure 39 Connection for External Crystal Oscillation

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.

3.20 Operating Mode

NY8TE64A 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, NY8TE64A will stop almost all operations except Timer0/Timer1/Timer4/ Timer5/WDT in order to wake-up periodically. At Halt mode, NY8TE64A 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.

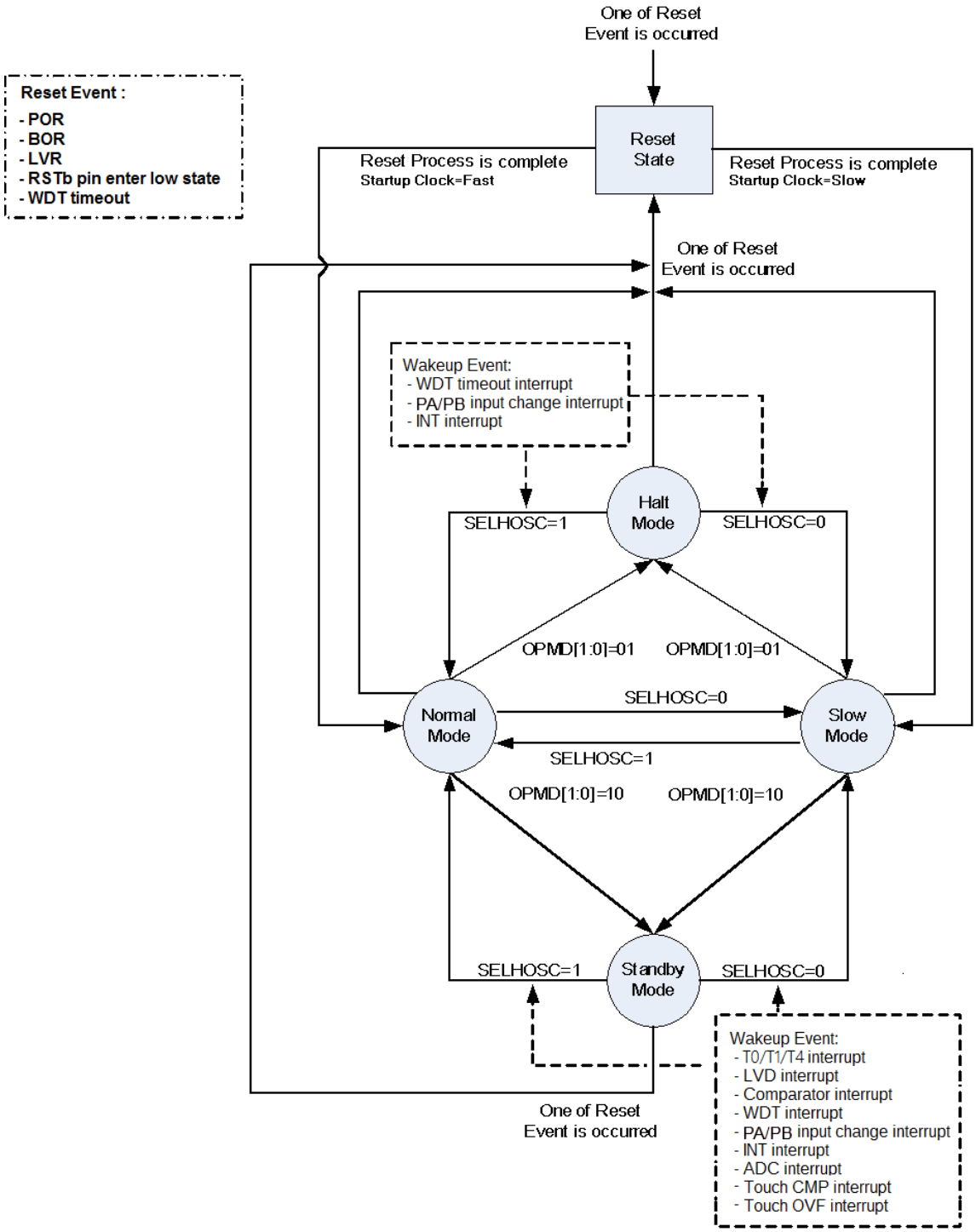


Figure 40 Four Operating Modes

3.20.1 Normal Mode

After any Reset Event is occurred and Reset Process is completed, NY8TE64A 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, NY8TE64A will enter Normal mode, if Startup Clock=Slow, NY8TE64A 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, NY8TE64A 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 NY8TE64A 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.

3.20.2 Slow Mode

NY8TE64A 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 NY8TE64A. 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.

3.20.3 Standby Mode

NY8TE64A 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 NY8TE64A and user has to enter slow mode and write 1 to register bit STPHOSC in order to stop F_{HOSC} oscillation. Most of NY8TE64A peripheral modules are disabled but Timer can be still active if register bit T0EN/T1EN/T4EN/T5EN is set to 1. Therefore NY8TE64A can wake-up after Timer0/Timer1/Timer4/Timer5 is expired. The expiration period is determined by the register TMR0/TMR1[9:0]/TMR4[9:0]/TMR5[9:0], F_{INST} and other configurations for Timer0/Timer1/Timer4/Timer5.

- 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/Timer4/Timer5 (overflow/underflow) interrupt, (b) WDT timeout interrupt, (c) PA/PB input change interrupt, (d) INT external interrupt is happened, (e) LVD interrupt, (f) Comparator output status change interrupt or (g) ADC end-of-convert interrupt.
- 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.

3.20.4 Halt Mode

NY8TE64A 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 NY8TE64A can only wake-up by some specific events. Therefore, Halt mode is the most power saving mode provided by NY8TE64A.

- 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) PA/PB input change interrupt or (c) INT or external interrupt is happened.
- After wake-up from Halt mode, IC will return to Normal mode if SELHOSC=1, IC will return to Slow mode if SELHOSC=0.

Note: Users can change STPHOSC and enter Halt mode in the same instruction.

- It is not recommended to change oscillator mode (normal to slow or slow to normal) and enter halt mode at the same time.

3.20.5 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 NY8TE64A enter Standby mode or Halt mode, user may execute instruction ENI. At this condition, NY8TE64A 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.

3.20.6 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
Timer0/1/2/3	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	-	-	- Timer0/1/4/5 overflow - WDT timeout - PA/PB input change - INT0/1/2 - LVD interrupt - Comparator interrupt - ADC end-of-convert - Touch CMP INT - Touch OVF INT	- WDT timeout - PA/PB input change - INT0/1/2 - Touch CMP INT - Touch OVF INT

Table 35 Summary of Operating Modes

3.21 Reset Process

NY8TE64A 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 36 Summary of /TO & /PD Value and its Associated Event

After Reset Event is released, NY8TE64A will start Reset Process. It will wait certain amount of period for oscillation stable no matter what kind of oscillator is adopted. This period is called power-up reset time and is determined by three-bit configuration words which can be 140us, 4.5ms, 18ms, 72ms or 288ms. After power-up reset time, NY8TE64A will wait for further oscillator start-up time (OST) before it starts to execute program. OST=1 clock cycle of F_{OSC} if the previous power-up time is 140us, OST=16 clock cycles of F_{OSC} if the previous power-up time is 4.5ms, 18ms, 72ms or 288ms.

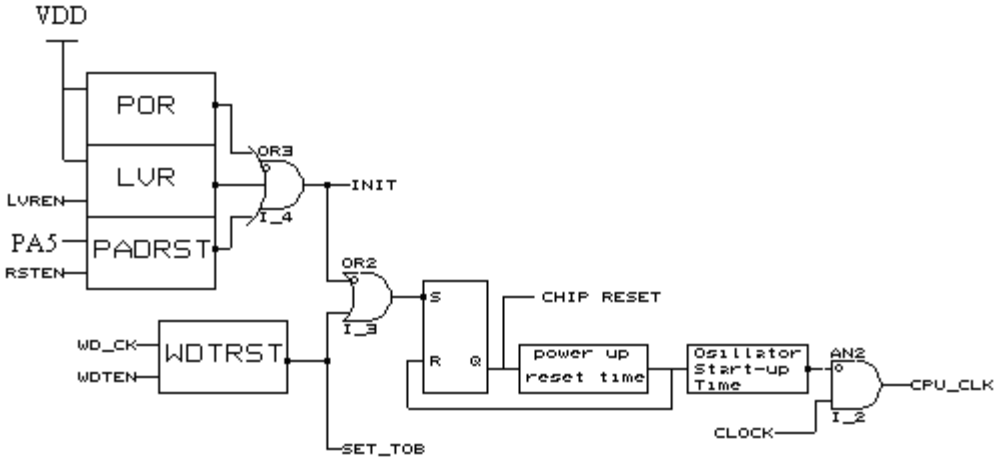


Figure 41 Block diagram of on-chip reset circuit

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.

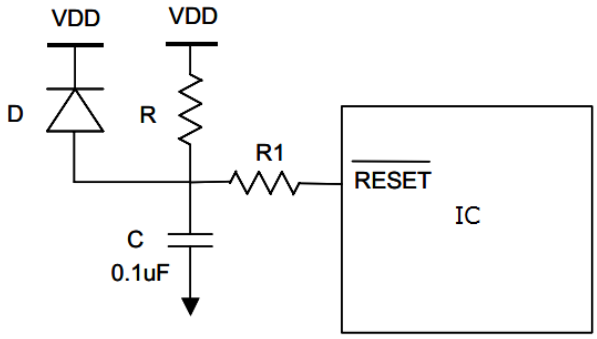


Figure 42 Block Diagram of Reset Application

3.22 SPI MODE

Features of the SPI include:

- Full-duplex operation
- Four programmable master mode frequencies
- Serial clock with programmable polarity and phase
- End of transmission interrupt flag
- Write collision error flag
- Bus contention error flag

NY8TE64A will enter SPI MODE by writing SIMCR[7]=1. It includes four line SPI interface if SIMCR[4]=1,and It includes three line SPI interface if SIMCR[4]=0 .

The communication is full duplex and operates as a slave/master type, where the device can be either master or slave. The device is master If SIMCR[5]=1 , it is slave if SIMCR[5]=0

The SPI interface is a full duplex synchronous serial data link. It is a four line interface with pin names SCK, MISO,MOSI and SSBEN. When choose three line interface , SSBEN is omitted , no matter master or slave are always enable.

Only a master SPI can initiate transmissions. Software begins the transmission from a master SPI by writing to the SPI data register (SIMDR). The SIMDR does not buffer data being transmitted from the SPI. Data written to the SIMDR goes directly into the shift register and begins the transmission immediately under the control of the serial clock. The transmission ends after eight cycles of the serial clock when the SPI flag (SPIF) becomes set. At the same time that SPIF becomes set, the data shifted into the master SPI from the receiving device transfers to the SIMDR. The SIMDR buffers data being received by the SPI. Before the master SPI sends the next byte, software must clear the SPIF bit by reading the SPCR and then read or write the SIMDR.

In a slave SPI, data enters the shift register under the control of the serial clock from the master SPI. After a byte enters the shift register of a slave SPI, it transfers to the SIMDR. To prevent an overrun condition, slave software must then read the byte in the SIMDR before another byte enters the shift register and is ready to transfer to the SIMDR.

Read SPCR ,and then read or write SIMDR will clear SPIF and WCOL .

Read SPCR , and then write SPCR will clear MODF

Figure 40~42 show examples how to work when master connect with slave .

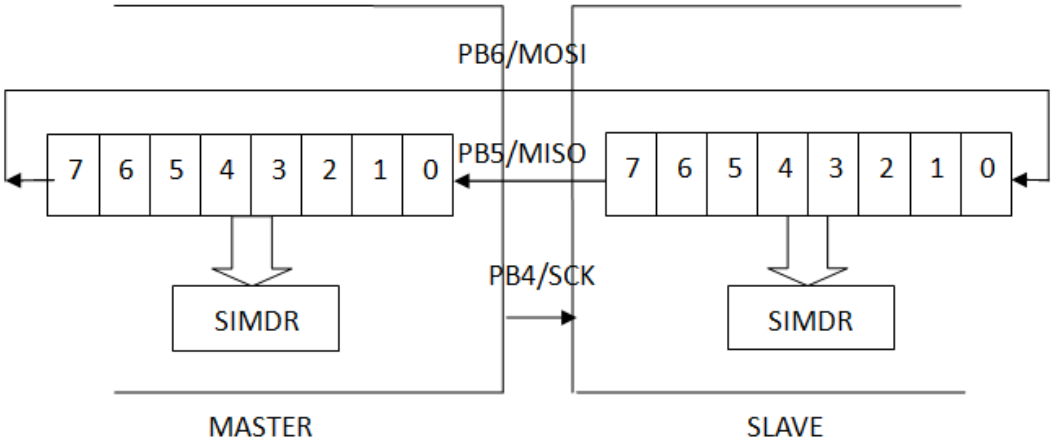


Figure 43 Single Master/Slave

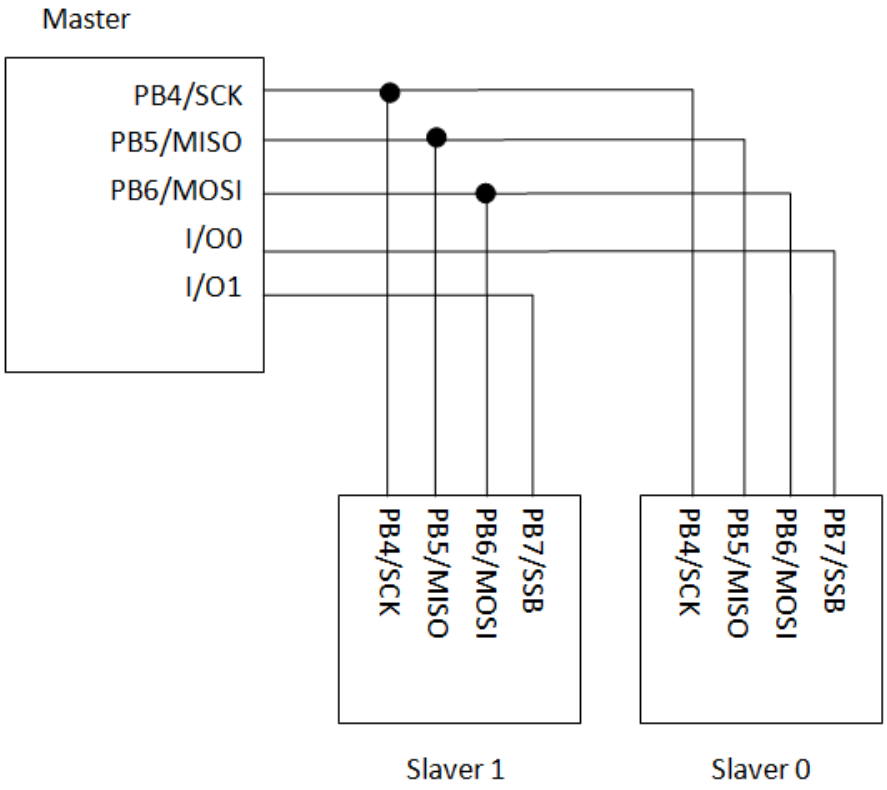


Figure 44 Single Master and Multi Slaves

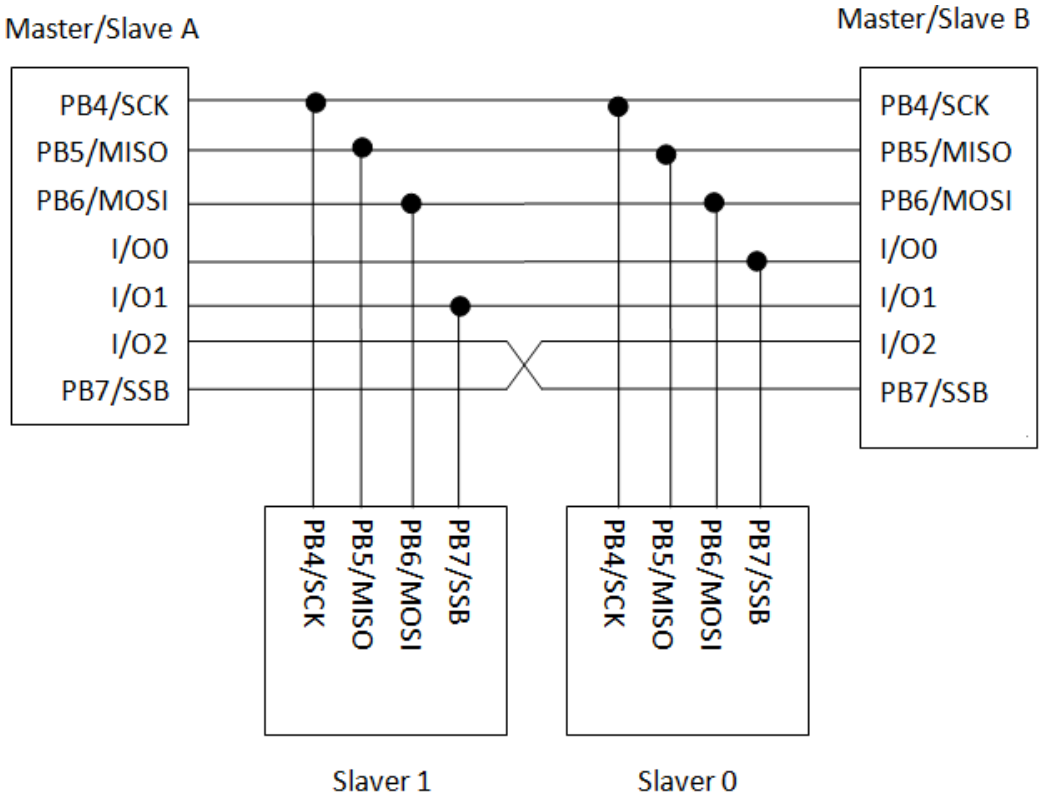


Figure 45 Multi Masters and Multi Slaves

3.22.1 Serial Clock Polarity and Phase

There are four kind of type to accommodate the different serial communication requirements of peripheral devices, depending upon the configurations of the CPOL (SPCR[3])bit and CKEG(SPCR[2]) bit.

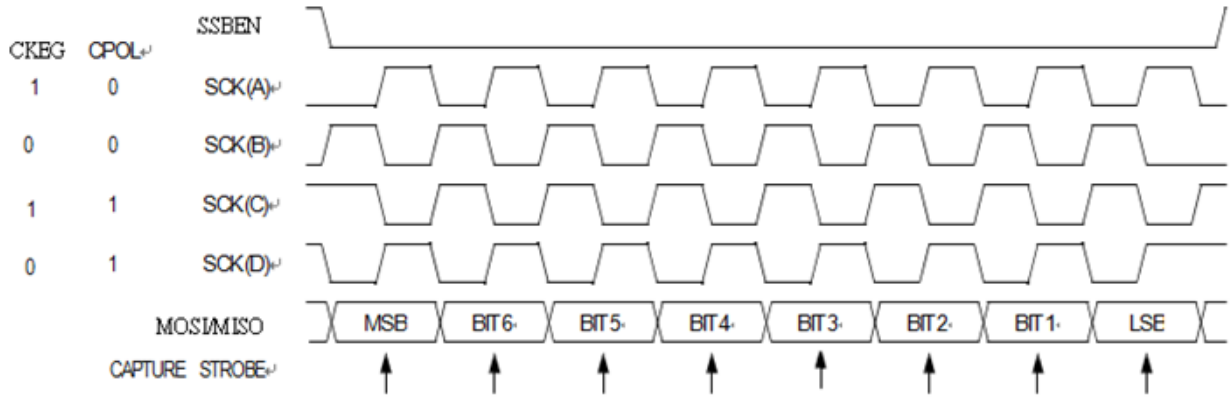


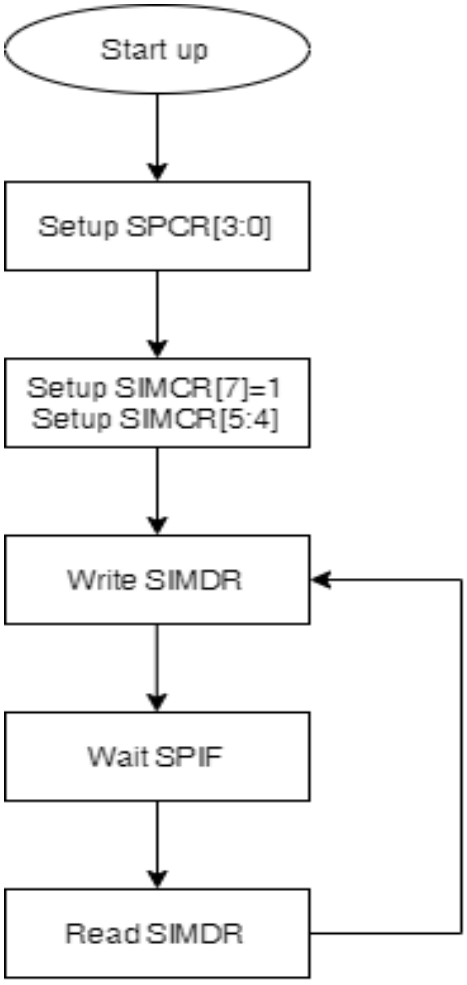
Figure 46 Shows how the CPOL and CKEG bits affect the clock/data timing.

There are four kind of different SPI master clock rate , configuration by SPCR[1:0] (SPR[1:0]) , SPR[1:0]=00/01/10/11 will choose system clock/2 , system clock /4 , system clock /16 , system clock/32.

User must set up SPCR[3:0] firstly , and then enable SPI module by set SIMCR[7]=1 , otherwise

hardware errors will happen .

The following picture is SPI flow chart



3.22.2 SPI error Conditions

These conditions produce SPI system errors:

(1) MODF (Mode fault error) is happened if PB7/SSBEN is at logic 0 when in master mode .

(2) Writing to the SPDR during a transmission causes a write-collision error and sets the WCOL bit in the SPCR The error does not affect the transmission of the previously written byte, but the byte that caused the error is lost .

(3) Failing to read the SPDR before the next incoming byte sets the SPIF bit

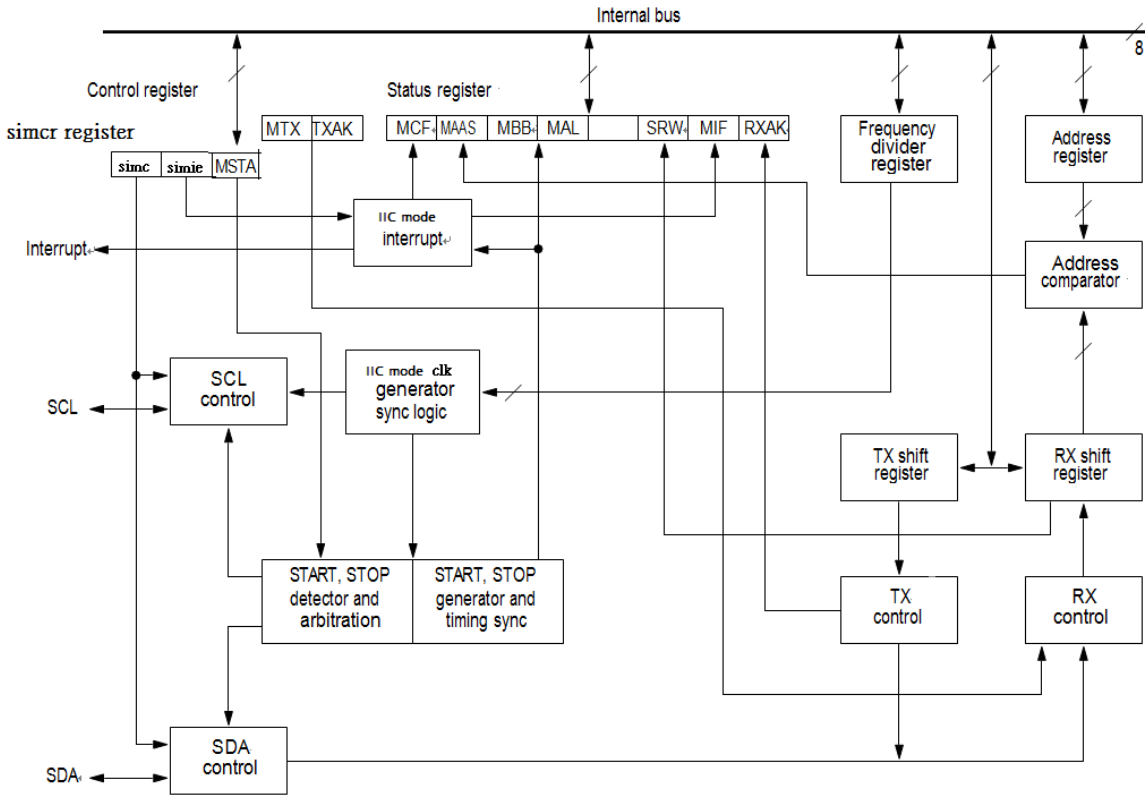
3.23 IIC MODE

NY8TE64A will enter IIC mode by writing SIMCR=1. the IIC mode standard is a two wire bus . This two-wire bus minimizes the interconnection between devices and eliminates the need for address decoders, having the advantage of less PCB routing and economic hardware structure. IIC mode is suitable for applications requiring communications in a short distance among a number of devices.

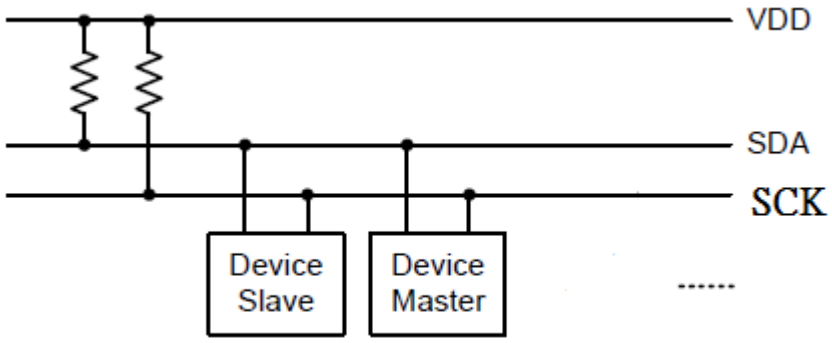
IIC mode has the following feature

- Multi-master operation
- 32 software programmable serial clock frequencies
- Software selectable acknowledge bit
- Interrupt driven byte-by-byte data transfer
- Arbitration lost driven interrupt
- Calling address identification interrupt
- Generate/detect the start, stop and acknowledge signals
- Repeated START signal generation
- Bus busy detection

The following figure is IIC mode function block.



The following figure is a example for IIC operation. One master device and one slave device. SDA and SCK need pull high resistor.

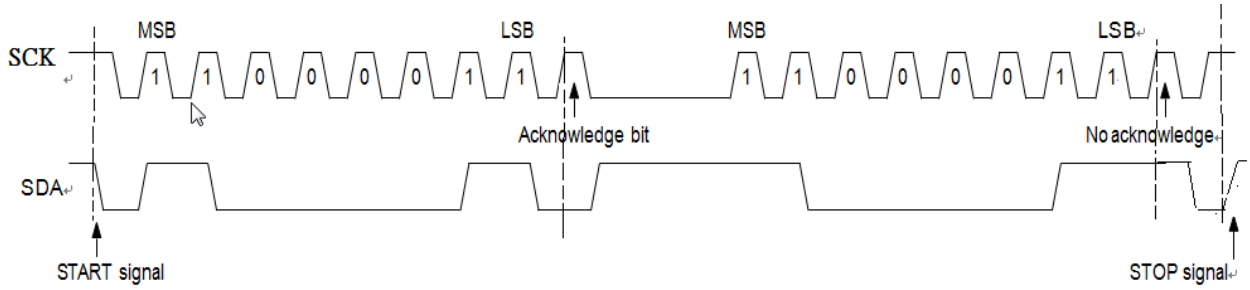


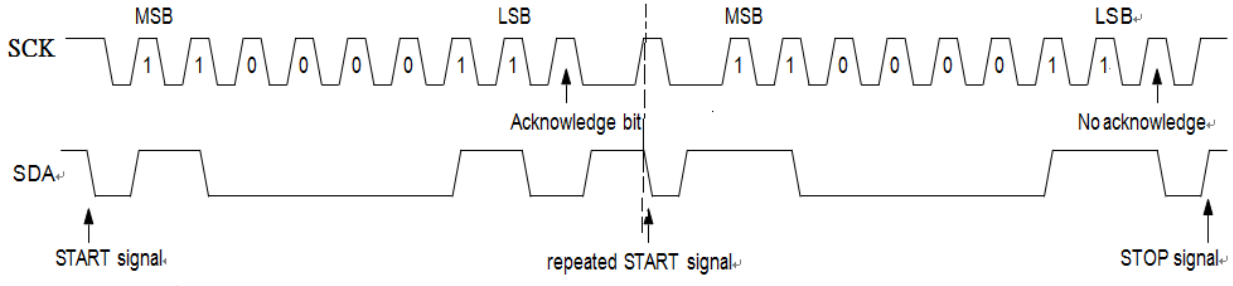
IIC Master/Slave Bus Connection

3.23.1 IIC MODE protocol

A standard communication is composed of four parts, they are (1) START (2)slave address transfer (3) data transfer (4) STOP .

The following diagram show SDA/SCK





3.23.2 IIC MODE operating

NY8TE64A will enter IIC mode by writing SIMCR=1.The device will enter master mode if SIMCR[5](MSTA)=1, and The device will enter slave mode if SIMCR[5](MSTA)=0.

Upon reset, SIMCR[5] is cleared. When this bit is changed from 0 to 1, a START signal is generated on the bus, and the master mode is selected. When this bit is changed from 1 to 0, a STOP signal is generated and the operation mode changes from master to slave.

MCR[4]=1 , IIC MODE is set for transmit , otherwise MCR[4]=0, IIC MODE is set for receive mode .

Each time , we want to TX/RX data , firstly we go to clear MSR[1](MIF) . And then Clear MSTA to 0 if MAL=1.

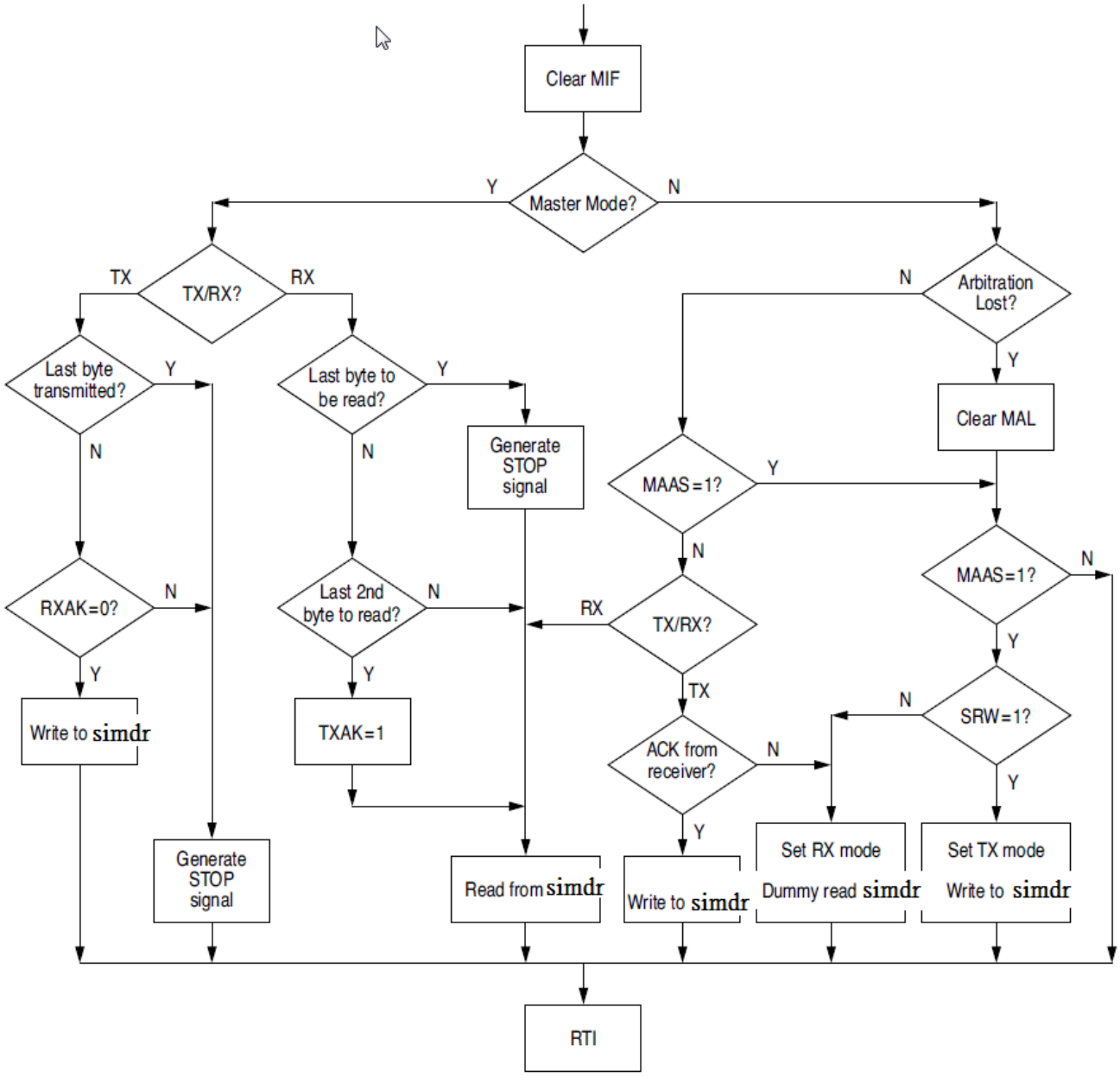
In Master/TX mode, after transmit 8 bit of data, will read MSR[0] to check receiving 9th clock sending acknowledge or not . if receiving acknowledge bit, then continues to write SIMDR , otherwise generating STOP signal to terminal communication .

In master/RX mode, set TXAK=0 and receive data, it will send acknowledge signal at 9th clock bit. If last 2 byte of data have been read from SIMDR, and then set TXAK=1. If last bye have been read from SIMDR, then generate STOP signal.

In Slave mode, check MSR[6] (MASS) if it is first byte . If MSR[6](MASS)=1 ,then continue to check if SRW is high or low . IF SRW is high , then set MCR[4](MTX)=1 to enter TX mode and write data to SIMDR ,else SRW is low, then set MCR[4](MTX)=0 to RX mode and dummy read SIMDR .

In Slave/TX mode, after sending 8 bit of data, then read MSR[0](RXAK) to check if receiving 9th clock sending acknowledge or not . If receiving acknowledge bit, then continuing to write SIMDR, otherwise set RX mode and dummy read SIMDR.

The following diagram show the suggested flow.



3.23.3 Arbitration Lost

This interface circuit is a true multi-master system which allows more than one master to be connected. If two or more masters try to control the bus at the same time, a clock synchronization procedure determines the bus clock. The clock low period is equal to the longest clock low period among the masters; and the clock high period is the shortest among the masters. A data arbitration procedure determines the priority. A master will lose arbitration if it transmits a logic “1” while the others transmit logic “0”, the losing master will immediately switch over to slave receive mode and stops its data and clock outputs. The transition from master to slave mode will not generate a STOP condition. Meanwhile, a software bit will be set by hardware to indicate loss of arbitration.

3.24 Touch Pad

NY8TE64A can replace the mechanical switch or button with the touch pad function. A built-in LDO regulator for touch sensor can provide a stable capacitive sensing design for touch application.

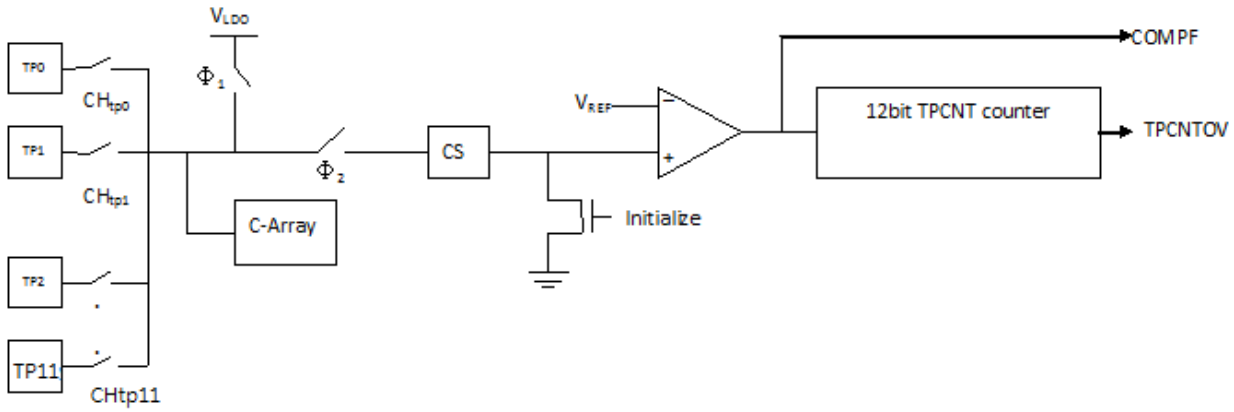
Multiple Touch-Keys is from 1 touch key up to 12 touch keys decided by TPPADEN0 SFR and TPPADEN1 SFR. Scan which touch key chosen by TPCHS SFR & TPCR SFR. The modulation clock for touch key can be selected to be 1.31M, 1.12M, 1M, 0.88M or 0.75M. IN norm scan and single key scan or G0 scan ,the touch key counter can be set by TPCNTH0 and TPCNTL0 before Touch-Pad take into operation. IN norm scan and G1 scan ,the touch key counter can be set by TPCNTH1 and TPCNTL1 before Touch-Pad take into operation. IN norm scan and G2 scan ,the touch key counter can be set by TPCNTH2 and TPCNTL2 before Touch-Pad take into operation.Then set TPRUN, the touch key will start to convert the touch sensing counter. Either touch external capacitor voltage beyond the comparator reference voltage or the touch counter overflow will terminate the modulation, also set TPCPIF=1 or TPOVIF=1 relative.

NY8TE64A support touch key slow mode to reduce the power consumption in CPU standby mode. The slow mode can be set 16Hz or 32Hz scan period. Then set TPRUN in TP slow mode, the modulation will start every 62ms or 31ms automatic. The scan mode can be single key ,G0 , G1,G2 , G0-G1-G0-G1,G0-G1-G2-G0-G1-G2 decided by TPCRD and TPCHS .

Write the TPCHS SFR will reload the TPCNTH and TPCNTL to touch counter and set TPMD to TPCHSOFF.

Write the TPMD SFR will also reload the TPCNTH and TPCNTL to touch counter.

In TPRUN mode, clear the touch interrupt flag will set TPMD to TPCHSOFF, but not in touch slow mode.



3.25 UART

The programmable asynchronous communications interface (UART) megafunction provides data formatting and control to a serial communication channel.

The megafunction has select, read/write, interrupt and bus interface logic features that allow data transfers over an 8-bit bi-directional parallel data bus system. With proper formatting and error checking, the megafunction can

transmit and receive serial data, supporting asynchronous operation.

- Full double buffering
- Asynchronous operation
- Independently controlled Transmit, Line Status and Receive Interrupts
- Programmable data word length (5 - 8 bit), parity and stop bits
- Parity, overrun and framing error checking
- Programmable Baud Rate Generator allows division of any reference clock by 1 to (2^{16}-1) and generates an internal 16 X Clock
- False start bit detection
- Automatic break generation and detection
- Internal diagnostic capabilities

The transmitter section is composed of a Transmit Holding Register (THR) and a Transmit Shift Register (TSR). Writing to THR will transfer the contents of the data bus (DIN 7-0) to the Transmit Holding Register every time that the THR or TSR is empty. This write operation should be done when Transmit Holding Register Empty (THRE) is set

This register contains the assembled received data. On the falling edge of the start bit, the receiver section starts its operations. The start bit is valid if the RXDATA is still low at the middle sample of Start bit, thus preventing the receiver from assembling a false data character

The Line Control Register is used to specify the data communication format. The break feature, parity, stop bits and word length can be changed by writing to the appropriate bits in LSR.

3.26 On-Chip Debug (OCD)

3.26.1 Function Description

NY8TE64A is embedded in an on-chip debugger(OCD) providing developers with a low cost method for debugging user code. The OCD gives debug capability of complete program flow control with 3 hardware address breakpoints, 1 conditional register break, single step, free running, and non-intrusive commands for memory access. The OCD system does not occupy any locations in the memory map and does not share any on-chip peripherals.

The OCD system uses a two-wire serial interface, SCL and SDA, to establish communication between the target device and the controlling debugger host. SDA is an input/output pin for debug data transfer and SCL is an input pin for synchronization with SDA. The NY8TE64A also use SCL and SDA as control pins to write and read MTP.

3.26.2 Limitation of OCD

NY8TE64A is a fully-featured microcontroller that multiplexes several functions on its limited I/O pins. Some device functionality must be sacrificed to provide resources for OCD system. The OCD has the following limitations:

1. The SCL/SDA pins are physically located on the same pin PC1/PC0 or PA3/PA2. Therefore, neither its I/O function nor shared multi-functions can be emulated.

2. System clock cannot be turned off because OCD uses this clock to monitor its internal status: When the system is in halt mode, it is invalid to perform ram/registers accesses because parts of the device may not be clocked. A read access could return garbage or a write access might not succeed. But the following accesses are not effected by the system halt : Read current program address , current PCL, current break condition and current halt status.

4. Instruction Set

NY8TE64A 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	-
TFUN	T		Load ACC to T-page SFR	1	-
TFUNR	T		Move T-page SFR to ACC	1	-
T0MD			Load ACC to T0MD	1	-
T0MDR			Move T0MD to ACC	1	-
TABLEA			Read ROM	2	-

Inst.	OP 1	OP 2	Operation	Cyc.	Flag
Arithmetic Instructions					
ADDAR	R	d	dest = R + ACC	1	Z, DC, C
SUBAR	R	d	dest = R + (~ACC)	1	Z, DC, C
ADCAR	R	d	dest = R + ACC + C	1	Z, DC, C
SBCAR	R	d	dest = R + (~ACC) + C	1	Z, DC, C
ADDIA	i		ACC = i + ACC	1	Z, DC, C
SUBIA	i		ACC = i + (~ACC)	1	Z, DC, C
ADCIA	i		ACC = i + ACC + C	1	Z, DC, C
SBCIA	i		ACC = i + (~ACC) + C	1	Z, DC, C
DAA			Decimal adjust for ACC	1	C
CMPAR	R		Compare R with ACC	1	Z, C
CLRA			Clear ACC	1	Z
CLRR			Clear R	1	Z
Other Instructions					
NOP			No operation	1	-
SLEEP			Go into Halt mode	1	/TO, /PD
CLRWDT			Clear Watch-Dog Timer	1	/TO, /PD
ENI			Enable interrupt	1	-
DISI			Disable interrupt	1	-
INT			Software Interrupt	3	-
RET			Return from subroutine	2	-
RETIE			Return from interrupt and enable interrupt	2	-
RETIA	i		Return, place immediate in ACC	2	-
CALLA			Call subroutine by ACC	2	-
GOTOA			unconditional branch by ACC	2	-
LCALL	adr		Call subroutine(2K)	2	-
LGOTO	adr		unconditional branch(2K)	2	-
FCALL	adr		Call subroutine (4K)	2	-
FGOTO	adr		unconditional branch(4K)	2	-

Table 37 Instruction Set

ACC: Accumulator.

adr: immediate address.

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

C: Carry/Borrow bit

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

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

d: Destination

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

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

DC: Digital carry flag.

dest: Destination.

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

i: 8-bit immediate data.

PC: Program Counter.

PCHBUF: High Byte Buffer of Program Counter.

/PD: Power down flag bit

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

/PD=0, after instruction SLEEP is executed.

Prescaler: Prescaler0 dividing rate.

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

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

T0MD: T0MD register.

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

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

/TO: Time overflow flag bit

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

/TO=0, WDT timeout is occurred.

WDT: Watchdog Timer Counter.

Z: Zero flag

.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

FGOTO	Unconditional Branch
Syntax:	FGOTO adr
Operand:	0 ≤ adr ≤ 4095
Operation:	adr → PC[11:0].
Status affected:	--
Description:	LGOTO is an unconditional branch instruction. The 12-bit immediate address adr is loaded into PC[11:0].
Cycle:	2
Example:	FGOTO Level before executing instruction: PC=A0. after executing instruction: PC=address of Level.

FCALL	Call Subroutine
Syntax:	FCALL adr
Operand:	0 ≤ adr ≤ 4095
Operation:	PC + 1 → Top of Stack, adr → PC[11:0]
Status affected:	--
Description:	The return address (PC + 1) is pushed onto top of Stack. The 12-bit immediate address adr is loaded into PC[11:0].
Cycle:	2
Example:	FCALL SUB before executing instruction: PC=A0. Stack level=1 after executing instruction: PC=address of SUB, Stack[1]= A0+1, Stack pointer =2.

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

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

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

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

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

IORIA	OR Immediate with ACC
Syntax:	IORIA i
Operand:	$0 \leq i < 255$
Operation:	ACC | i $\rightarrow$ 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.

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

IOSTR	Move F-page SFR to ACC
Syntax:	IOSTR F
Operand:	$5 \leq F \leq 15$
Operation:	F-page SFR $\rightarrow$ 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.

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

LGOTO	Unconditional Branch
Syntax:	LGOTO adr

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

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

MOVR	Move R to ACC or R
Syntax:	MOVR R, d
Operand:	0 ≤ R ≤ 127 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.

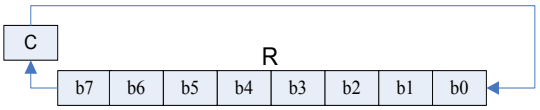
NOP	No Operation
Syntax:	NOP
Operand:	--
Operation:	No operation.
Status affected:	--
Description:	No operation.
Cycle:	1
Example:	NOP before executing instruction: PC=A0 after executing instruction: PC=A0+1

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.
Cycle:	2
Example:	RETIA i before executing instruction: Stack pointer =2. i=0x55, ACC=0xAA. after executing instruction: PC=Stack[2], Stack pointer =1. ACC=0x55.

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.

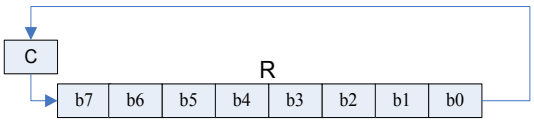
RET	Return from Subroutine
Syntax:	RET
Operand:	--
Operation:	Top of Stack $\rightarrow$ PC
Status affected:	--
Description:	PC is loaded from top of Stack as return address.
Cycle:	2
Example:	RET before executing instruction: Stack level=2. after executing instruction: PC=Stack[2], Stack level=1.

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



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

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



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

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

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

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

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

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

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

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

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

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.

TFUN	Load T-page SFR from ACC
Syntax:	TFUN T
Operand:	0 ≤ T ≤ 14
Operation:	ACC → T-page SFR
Status affected:	--
Description:	T-page SFR T is loaded by content of ACC.
Cycle:	1
Example:	TFUN T before executing instruction: T=0x55, ACC=0xAA. after executing instruction: T=0xAA, 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.

TFUNR	Move T-page SFR to ACC
Syntax:	TFUNR T
Operand:	0 ≤ T ≤ 14
Operation:	T-page SFR → ACC
Status affected:	--
Description:	Move T-page SFR T to ACC.
Cycle:	1
Example:	TFUNR T before executing instruction: T=0x55, ACC=0xAA. after executing instruction: T=0x55, ACC=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.

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

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

5. Configuration Words

Item	Name	Options
1	High Oscillator Frequency	1. I_HRC 2. 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 (I_LRC/E_LXT)
9	PA.5	1. PA.5 is I/O 2. PA.5 is reset
10	PA.7	1. PA.7 is I/O 2. PA.7 is instruction clock output
11	IR Output Pin	1. PB1 2. PA3
12	Startup Time	1. 140us 2. 4.5ms 3. 18ms 4. 72ms 5. 288ms
13	WDT Time Base	1. 3.5ms 2. 15ms 3. 60ms 4. 250ms
14	Noise Filter (High_EFT)	1. Enable 2. Disable
15	LVR Setting	1. Register Control 2. Register Control + Halt mode Off 3. Always On 4. Operation mode On + Halt mode Off
16	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
17	VDD Voltage	1. 3.0V 2. 4.5V 3. 5.0V
18	PWM1 Output Pin	1. PB1 2. PB4
19	PWM2 Output Pin	1. PB3 2. PB5
20	PWM3 Output Pin	1. PA2 2. PA7
21	Sink current type	1. Normal 2. Large 3. Constant
22	Comparator 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)
27	Input High Voltage (V_{IH})	1. 0.7VDD 2. 0.5VDD
28	Input Low Voltage (V_{IL})	1. 0.3VDD 2. 0.2VDD
29	VREFH Pin	1. PA0 2. PB1

Item	Name	Options
30	INT0 Input Pin	1. PB5 2. PA4
31	INT1 Input Pin	1. PB1 2. PA3
32	INT2 Input Pin	1. Disable 2. PA5

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}=16MHz @ I_HRC/2
			2.4				F_{INST}=20MHz @ I_HRC/4
			2.2				F_{INST}= 8MHz @I_HRC/2
			2.0				F_{INST}= 4MHz @ I_HRC/2
			2.0				F_{INST}= 32KHz @ I_LRC/2
V_{IH}	Input high voltage	5V	4.0	--	--	V	RSTb (0.8 V_{DD})
		3V	2.4	--	--		
		5V	3.5	--	--	V	All other I/O pins, EX_CKI0/1, INT0/1/2 CMOS option (0.7 V_{DD})
		3V	2.1	--	--		
		5V	2.5	--	--	V	All other I/O pins, EX_CKI0/1 TTL option (0.5 V_{DD})
		3V	1.5	--	--		
V_{IL}	Input low voltage	5V	--	--	1.0	V	RSTb (0.2 V_{DD})
		3V	--	--	0.6		
		5V	--	--	1.5	V	All other I/O pins, EX_CKI0/1, INT0/1/2 CMOS option (0.3 V_{DD})
		3V	--	--	0.9		
		5V	--	--	1.0	V	All other I/O pins, EX_CKI0/1 TTL option (0.2 V_{DD})
		3V	--	--	0.6		
I_{OH}	Output high current (Small Current)	5V	--	1.7	--	mA	V_{OH}=4.0V
		3V	--	0.9	--		V_{OH}=2.0V
I_{OH}	Output high current (Normal Current)	5V		16.7			V_{OH}=4.0V
		3V		9.7			V_{OH}=2.0V
I_{OL}	Output low current (Small current)	5V	--	7.2	--	mA	V_{OL}=1.0V
		3V	--	4.0	--		
I_{OL}	Output low current (Normal current)	5V	--	24.7	--	mA	V_{OL}=1.0V
		3V	--	14.6	--		
I_{OL}	Output low current (Large current)	5V	--	50	--	mA	V_{OL}=1.0V
		3V	--	34.5	--		
			--		--	mA	
			--		--		
I_{OP}	Operating current	**Normal Mode**					
		5V	--		--	mA	

Symbol	Parameter	V_{DD}	Min.	Typ.	Max.	Unit	Condition
		3V	--		--		
		5V	--	2.32	--	mA	F_{HOSC}=20MHz @ I_HRC/4 & E_HXT/4
		3V	--	1.31	--		
		5V	--	2.95	--	mA	F_{HOSC}=16MHz @ I_HRC/2 & E_HXT/2
		3V	--	1.65	--		
		5V	--	1.95	--	mA	F_{HOSC}=16MHz @ I_HRC/4 & E_HXT/4
		3V	--	1.11	--		
		5V	--	1.73	--	mA	F_{HOSC}=8MHz @ I_HRC/2 & E_HXT/2
		3V	--	0.99	--		
		5V	--	1.22	--	mA	F_{HOSC}=8MHz @ I_HRC/4 & E_HXT/4
		3V	--	0.71	--		
		5V	--	1.11	--	mA	F_{HOSC}=4MHz @ I_HRC/2 & E_XT/2
		3V	--	0.66	--		
		5V	--	0.83	--	mA	F_{HOSC}=4MHz @ I_HRC/4 & E_XT/4
		3V	--	0.52	--		
		5V	--	0.61	--	mA	F_{HOSC}=1MHz @ I_HRC/2 & E_XT/2
		3V	--	0.41	--		
		5V	--	0.53	--	mA	F_{HOSC}=1MHz @ I_HRC/4 & E_XT/4
		3V	--	0.37	--		
		Slow Mode					
		5V	--	8.0	--	uA	F_{HOSC} disabled, F_{LOSC}=32KHz @ I_LRC/2 & E_LXT/2
		3V	--	6.0	--		
		5V	--	5.3	--	uA	F_{HOSC} disabled, F_{LOSC}=32KHz @ I_LRC/4 & E_LXT/4
		3V	--	4.3	--		
I_{STB}	Standby current	5V	--	2.7	--	uA	Standby mode, F_{HOSC} disabled, F_{LOSC}=32KHz @ I_LRC/4
		3V	--	1.5	--		
I_{HALT}	Halt current	5V	--	--	0.5	uA	Halt mode, WDT disabled.
		3V	--	--	0.2		
		5V	--	--	5.0	uA	Halt mode, WDT enabled.
		3V	--	--	3.0		
R_{PH}	Pull-High resistor	5V	--	50	--	KΩ	Pull-High resistor (Without PA5)
		3V	--	90	--		
		5V	--	80	--	KΩ	Pull-High resistor (PA5)
		3V	--	80	--		
R_{PL}	Pull-Low resistor	5V	--	50	--	KΩ	Pull-Low resistor
		3V	--	90	--		

6.3 Comparator / LVD Characteristics

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

Symbol	Parameter	Min.	Typ.	Max.	Unit	Condition
V_{IVR}	Comparator input voltage range	0	--	5	V	F_{HOSC}=1MHz
T_{ENO}	Comparator enable to output valid	--	20	--	ms	F_{HOSC}=1MHz
I_{CO}	Operating current of comparator	--	250	--	uA	F_{HOSC}=1MHz, P2V mode
I_{LVD}	Operating current of LVD	--	300	--	uA	F_{HOSC}=1MHz, LVD=4.3V
E_{LVD}	LVD voltage error	--	--	3	%	F_{HOSC}=1MHz, LVD=4.3V

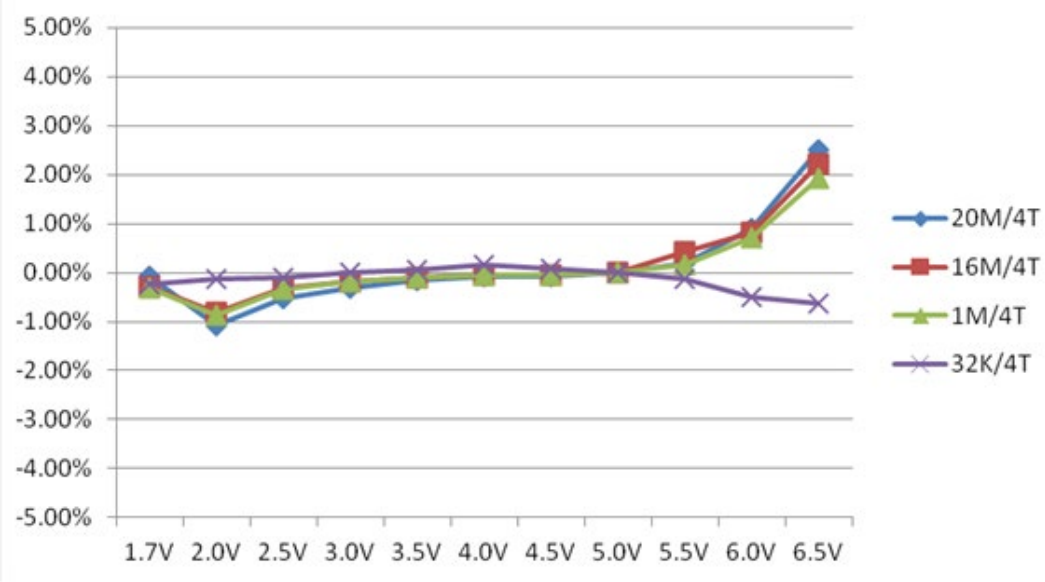
6.4 ADC Characteristics

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

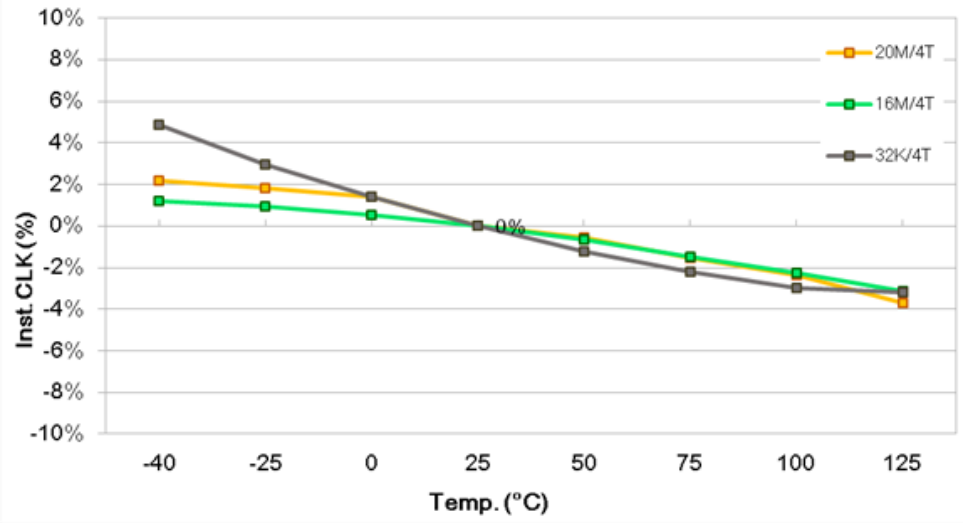
Symbol	Parameter	Min.	Typ.	Max.	Unit	Condition
V_{REFH}	VREFH input voltage	2V	--	V_{DD}	V	Ext. reference voltage
V_{REF4}	Int. 4V reference voltage, V_{DD}=5V	3.95	4	4.05	V	
V_{REF3}	Int. 3V reference voltage, V_{DD}=5V	2.95	3	3.05	V	
V_{REF2}	Int. 2V reference voltage, V_{DD}=5V	1.95	2	2.05	V	
V_{REF}	Int. V_{DD} reference voltage, V_{DD}=5V	--	V_{DD}	--	V	
	Internal reference supply voltage	V_{REF}+0.5	--	--	V	Minimum supply voltage
	ADC analog input voltage	0	--	V_{REFH}	V	
	ADC enable time	256	--	--	us	Ready to start convert after set ADENB="1".
$I_{OP(ADC)}$	ADC current consumption	--	0.3	--	mA	
ADCLK	ADC Clock Frequency	--	--	1M	Hz	
ADCYCLE	ADC Conversion Cycle Time	16	--		1/ADCLK	SHCLK=2 ADC clock
ADC_{sample}	ADC Sampling Rate	--	--	125	K/sec	V_{DD}=5V
DNL	Differential Nonlinearity	±1	--	--	LSB	V_{DD}=5.0V, AVREFH=5V, FADSMP=62.5K
INL	Integral Nonlinearity	±2	--	--	LSB	
NMC	No Missing Code	10	11	12	Bits	

6.5 Characteristic Graph

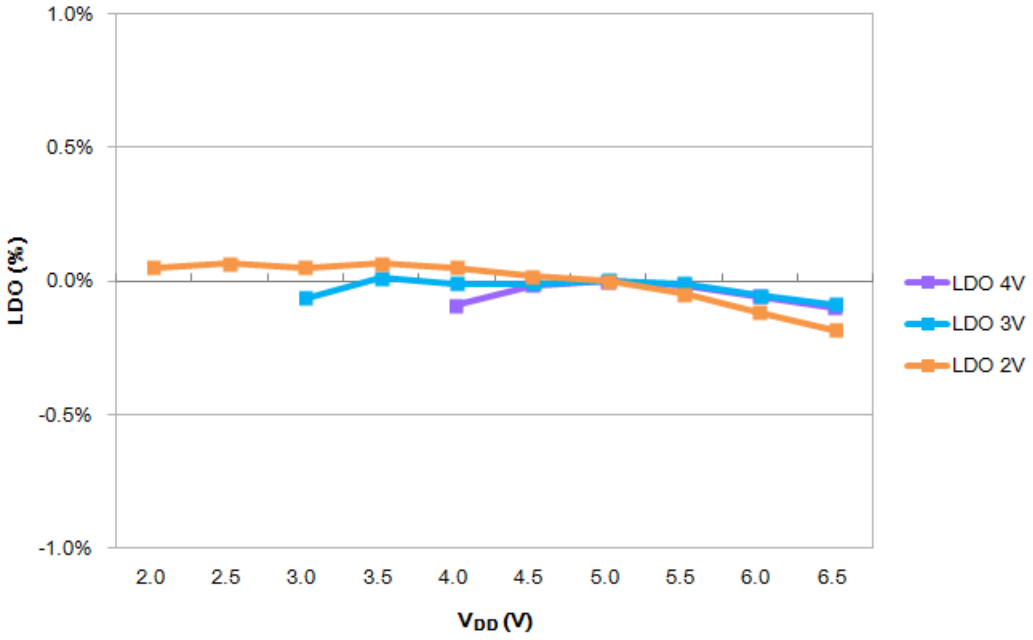
6.5.1 Frequency vs. V_{DD} of I_HRC and I_LRC



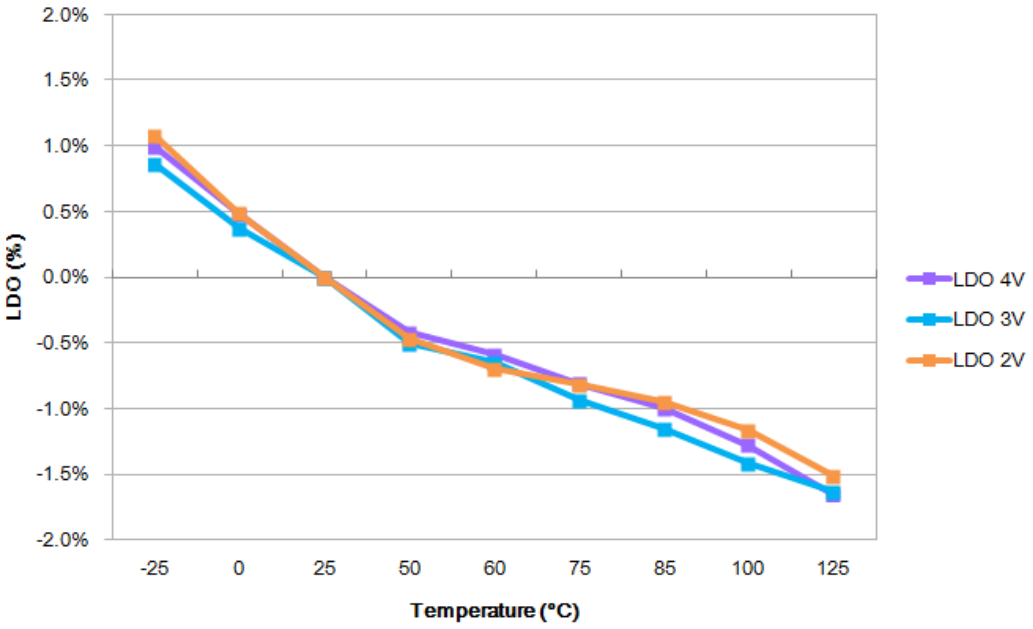
6.5.2 Frequency vs. Temperature of I_HRC and I_LRC



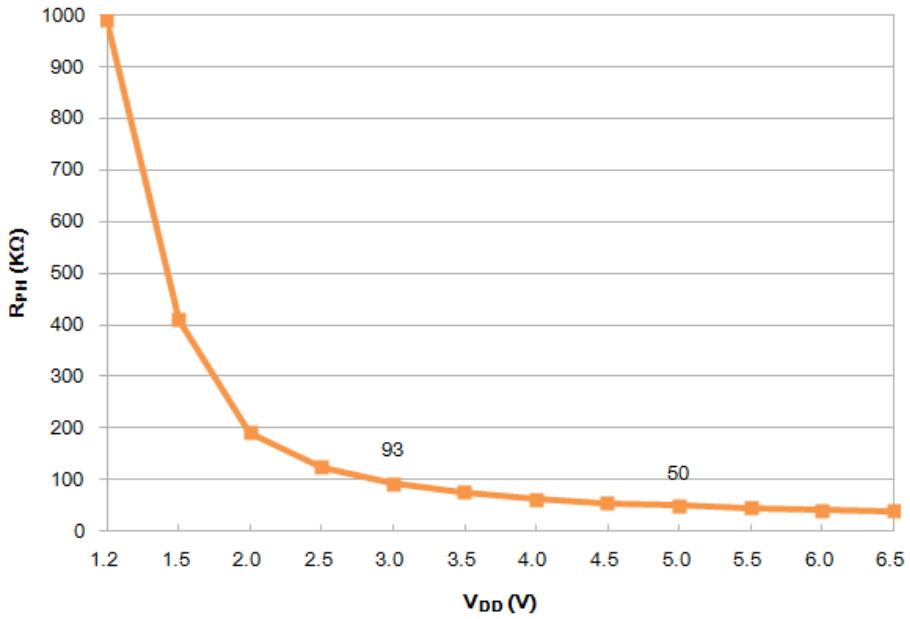
6.5.5 Low Dropout Regulator vs. V_{DD}



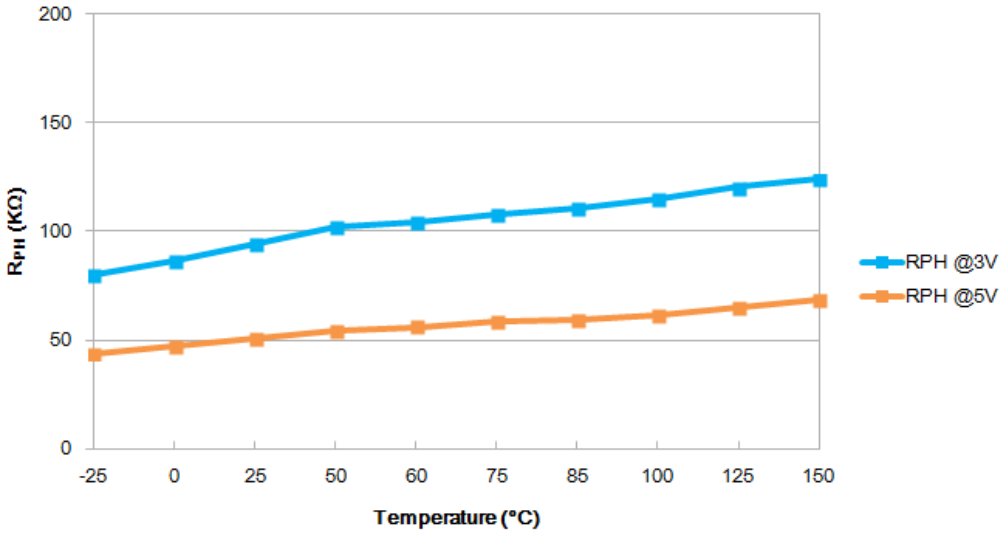
6.5.6 Low Dropout Regulator vs. Temperature



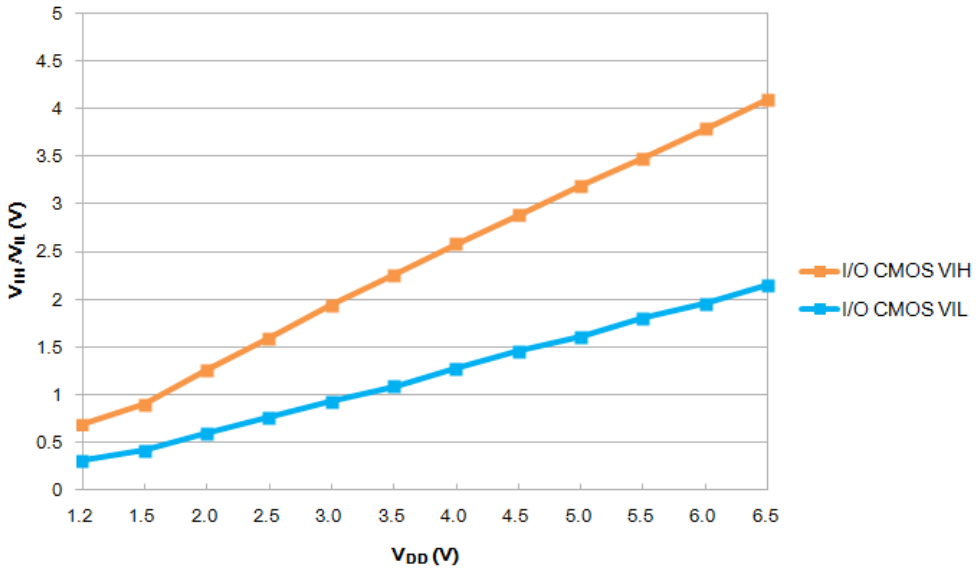
6.5.7 Pull High Resistor vs. V_{DD}

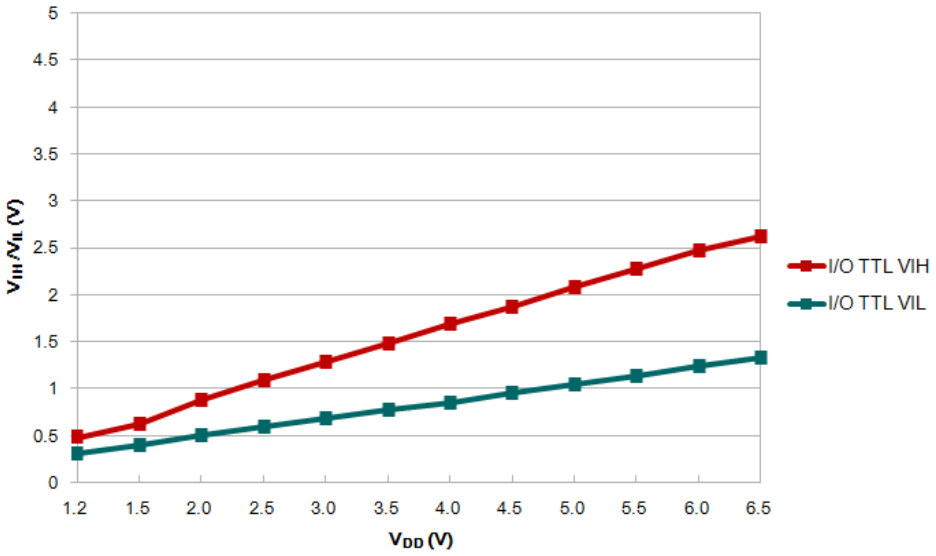


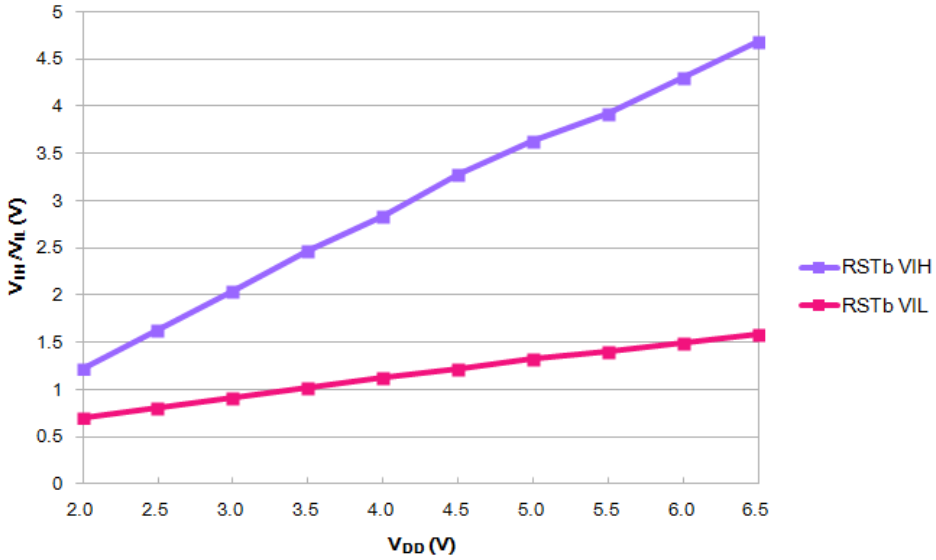
6.5.8 Pull High Resistor vs. Temperature



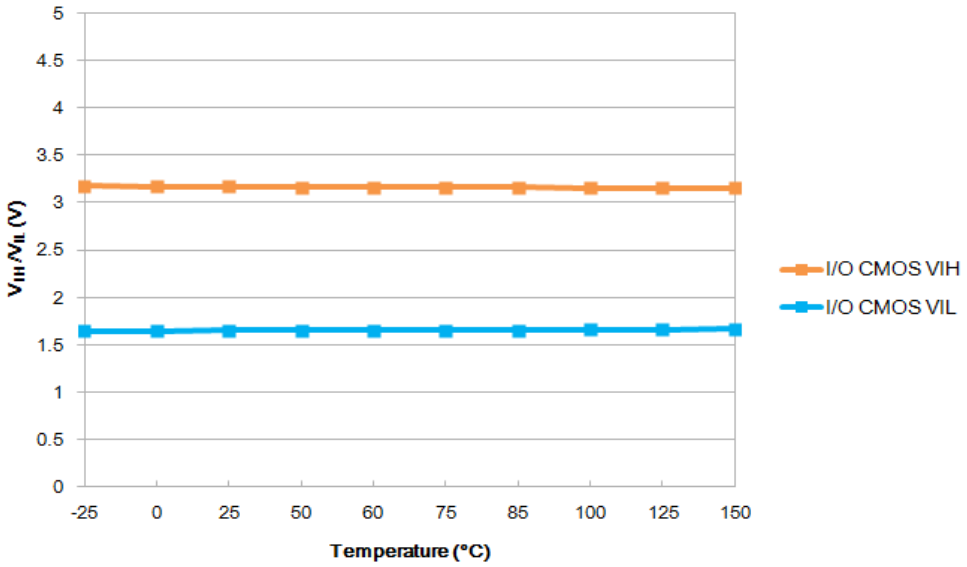
6.5.9 V_{IH}/V_{IL} vs. V_{DD}

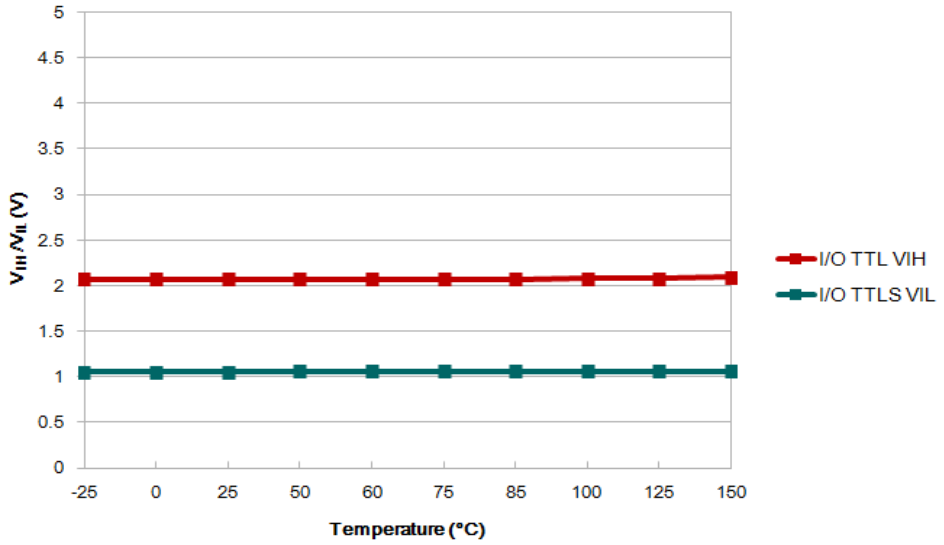


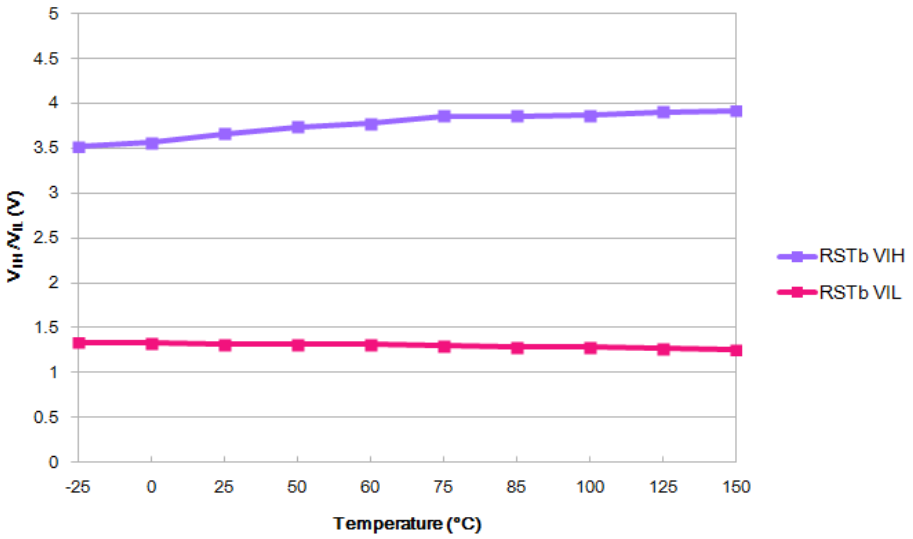




6.5.10 V_{IH}/V_{IL} vs. Temperature





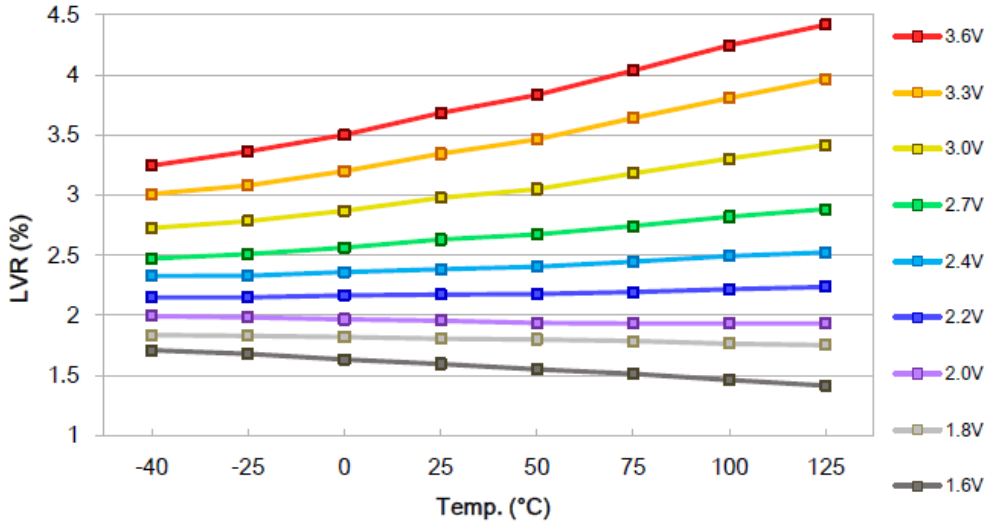


6.6 Recommended Operating Voltage

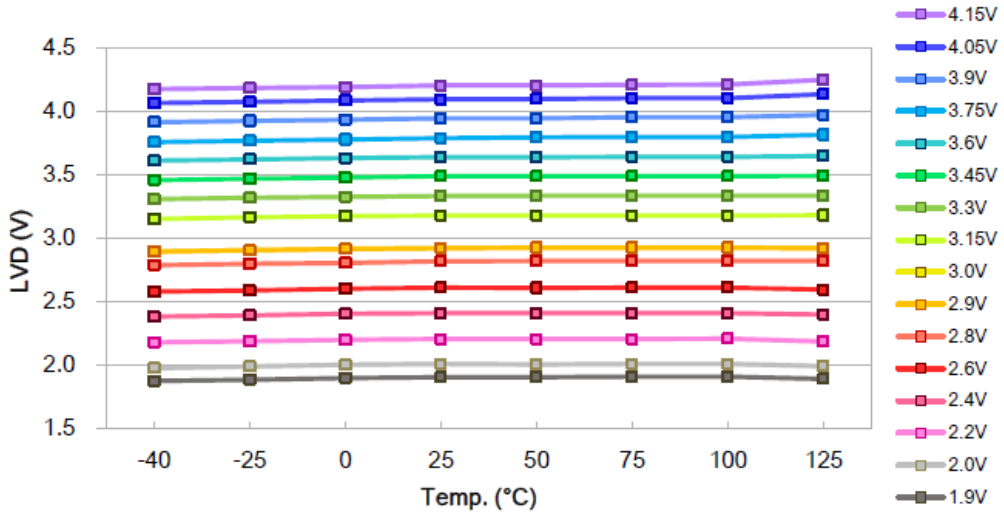
Recommended Operating Voltage (Temperature range: -40°C ~ +85°C)

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

6.7 LVR vs. Temperature

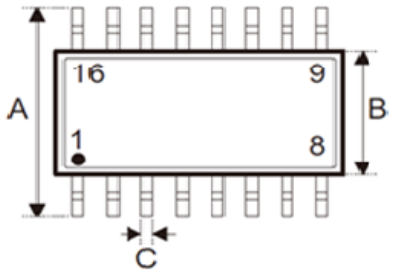


6.8 LVD vs. Temperature



7. Package Dimension

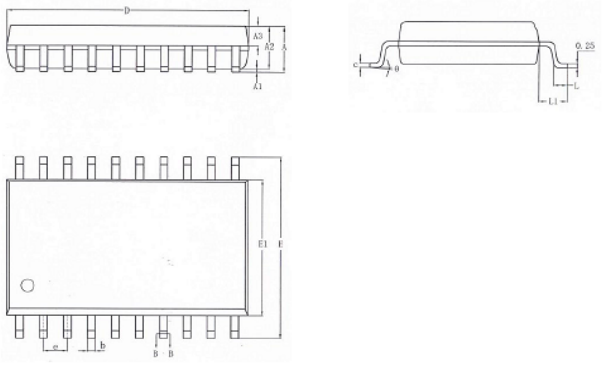
7.1 16-Pin Plastic SOP (150 mil)



	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°

Note: For 16-Pin SOP, 50 units per tube.

7.2 20-Pin Plastic SOP (300 mil)



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	2.65
A1	0.10	—	0.30
A2	2.25	2.30	2.35
A3	0.97	1.02	1.07
b	0.35	—	0.43
b1	0.34	0.37	0.40
c	0.25	—	0.29
c1	0.24	0.25	0.26
D	12.70	12.80	12.90
E	10.10	10.30	10.50
E1	7.40	7.50	7.60
e	1.27BSC		
L	0.70	—	1.00
L1	1.40REF		
θ	0	—	8°

Note: For 20-Pin SOP, 35 units per tube.

8. Ordering Information

P/N	*Package Type*	*Pin Count*	*Package Width*	*Shipping*
NY8TE64AS16	SOP	16	150 mil	Tube: 50 pcs per Tube
NY8TE64AS20	SOP	20	300 mil	Tube: 35 pcs per Tube.