AT8B62F1

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

Version 1.2

Nov. 28, 2025

Revision History

Version	Date	Description	Modified Page
1.0	2025/05/22	Formal release.	-
1.1	2025/09/04	1. Add 2 sets of external interrupts (INT0/INT1). 2. Add PA4 High Level Hold 1M spec. 3. Update 6.2 DC Characteristics.	13 8, 12, 117 117
1.2	2025/11/28	1. Update chapter 3.4.28 & 29 PWM5 setting. 2. Update Comparator / LVD Characteristics. 3. Update Recommended LVR Voltage.	53 119 124

Table of Contents

1. 概述

AT8B62F1 是以EPROM作為記憶體的 8 位元微控制器，專為家電或量測等等的I/O應用設計。採用CMOS製程並同時提供客戶低成本、高性能、及高性價比等顯著優勢。AT8B62F1 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。

AT8B62F1 內建高精度十一加一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，AT8B62F1 有 14 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，AT8B62F1 內建了可選擇頻率的紅外載波發射口。

AT8B62F1 有四組計時器，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AT8B62F1 提供 5 組 10 位元解析度的PWM 可用來驅動馬達、LED、或蜂鳴器等等。AT8B62F1 也設置有CCP模組可實現捕捉，PWM可實現具備死區互補的全橋/半橋PWM輸出。

AT8B62F1 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，AT8B62F1 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AT8B62F1 進入正常操作模式(Normal) 或 慢速模式(Slow mode) 來處理突發事件。

AT8B62F1 的PWM1/PWM2/PWM3/PWM4/PWM5 其計數器來源可不經除級。

1.1 功能

- 寬廣的工作電壓：
  - 3.0V ~ 5.5V @20MHz/2T.
  - 1.6V ~ 5.5V @4MHz/4T.
- 寬廣的工作溫度：-40°C ~ 85°C。
- 2Kx14 bits EPROM。
- 128 bytes SRAM。
- 14 根可分別單獨控制輸入輸出方向的I/O腳(GPIO)、PA[7:0]、PB[5:0]。
- PA[5, 3:0] 及 PB[3:0] 可選擇輸入時使用內建下拉電阻。
- PA4 可選擇High Level Hold 1M 高阻態功能。
- PA[7:0] 及 PB[5:0] 可選擇輸入時使用上拉電阻。
- PB[5:0] 可選擇開漏極輸出(Open-Drain)。

- PA[5] 可選擇當作輸入或開漏極輸出(Open-Drain)。
- 所有I/O腳輸出可選擇一般灌電流(Normal Sink Current)或大灌電流(Large Sink Current)，除PA5 外。
- 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。
- 三組 10 位元下數計時器(Timer1, 2, 3)可選重複載入或連續下數計時。Timer可設定頻率預除或頻率不經除級。
- 五個 10 位元脈衝寬度調變(PWM1, 2, 3, 4, 5)。PWM1 使用Timer1, PWM2 使用Timer2, PWM3/4/5 共用 Timer3。都可不經除級。
- 一組全功能CCP(Compare/Capture/PWM(HB,FB)。FB/HB PWM皆有死區互補功能。
- 三個蜂鳴器輸出(BZ1, 2, 3)。
- 38/57KHz紅外線載波頻率可供選擇，同時載波之極性也可以根據數據作選擇。
- 內建準確的低電壓偵測電路(LVD)。
- 內建十一加一通道 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)。
- 十二種硬體中斷：
  - Timer0 溢位中斷。
  - Timer1 借位中斷。
  - Timer2 借位中斷或CCP 中斷。
  - Timer3 借位中斷。
  - WDT 中斷。
  - PA/PB 輸入狀態改變中斷。
  - 兩組外部中斷輸入(INT0/INT1)。

- 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°

- AT8B62F1 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。
  - Timer2 借位中斷或CCP 中斷。
  - Timer3 借位中斷。
  - WDT 中斷。
  - PA/PB 輸入狀態改變中斷。
  - 兩組外部中斷輸入(INT0/INT1)。
  - 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。
- AT8B62F1 在睡眠模式(Halt mode)下的四種喚醒中斷：
  - WDT 中斷。
  - PA/PB 輸入狀態改變中斷。
  - 兩組外部中斷輸入(INT0/INT1)。

1.2 AT8B62F 與 AT8B62F1 的主要差異

Item	Function	AT8B62F	AT8B62F1
1	ADC offset Calibration	Yes	Yes
2	ADC Resolution DNL:+/-1;INL: +/-2 Lsb	V	V
3	CCP	X	V
4	PWM Channel	1, 2, 3, 4	1, 2, 3, 4, 5
5	Dead band	X	V
6	PA4 High Level Hold 1M	X	V
7	PWM 不經除級 I_HRC	X	V
8	PA5 Output High	X	V

(有關 ADC 零點校準流程，可參考 *NYIDE* 範例程式)

1. General Description

AT8B62F1 is an EPROM based 8-bit MCU tailored for ADC based applications like home appliances or meter equipment. AT8B62F1 adopts advanced CMOS technology to provide customers remarkable solution with low cost, high performance. RISC architecture is applied to AT8B62F1 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, AT8B62F1 is very suitable for those applications that are sophisticated but compact program size is required.

AT8B62F1 provides 11+1 channel high-precision 12-bit analog-to-digital converter (ADC), and high-precision Low Dropout Regulator and analog voltage comparator. They are suitable for any analog interface detection and measurement applications.

As AT8B62F1 address I/O type applications, it can provide 14 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, AT8B62F1 has built-in infrared (IR) carrier generator with selectable IR carrier frequency and polarity for applications which demand remote control feature.

AT8B62F1 also provides 4 sets of timers which can be used as regular timer based on system oscillation or event counter with external trigger clock. Moreover, AT8B62F1 provides 5 sets of 10-bit resolution Pulse Width Modulation (PWM) to drive the LED and buzzer. AT8B62F1 have CCP modules (Capture/Compare/Enhanced PWM), Enhanced PWM provide Full-bridged PWM and Half-bridged PWM with dead band functions.

AT8B62F1 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, AT8B62F1 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 AT8B62F1 operates in Standby mode and Halt mode, kinds of event will issue interrupt requests and can wake-up AT8B62F1 to enter Normal mode and Slow mode in order to process urgent events.

AT8B62F1 PWM1/PWM2/PWM3/PWM4/PWM5 can use internal high frequency (16M, 20MHz) as the clock source.

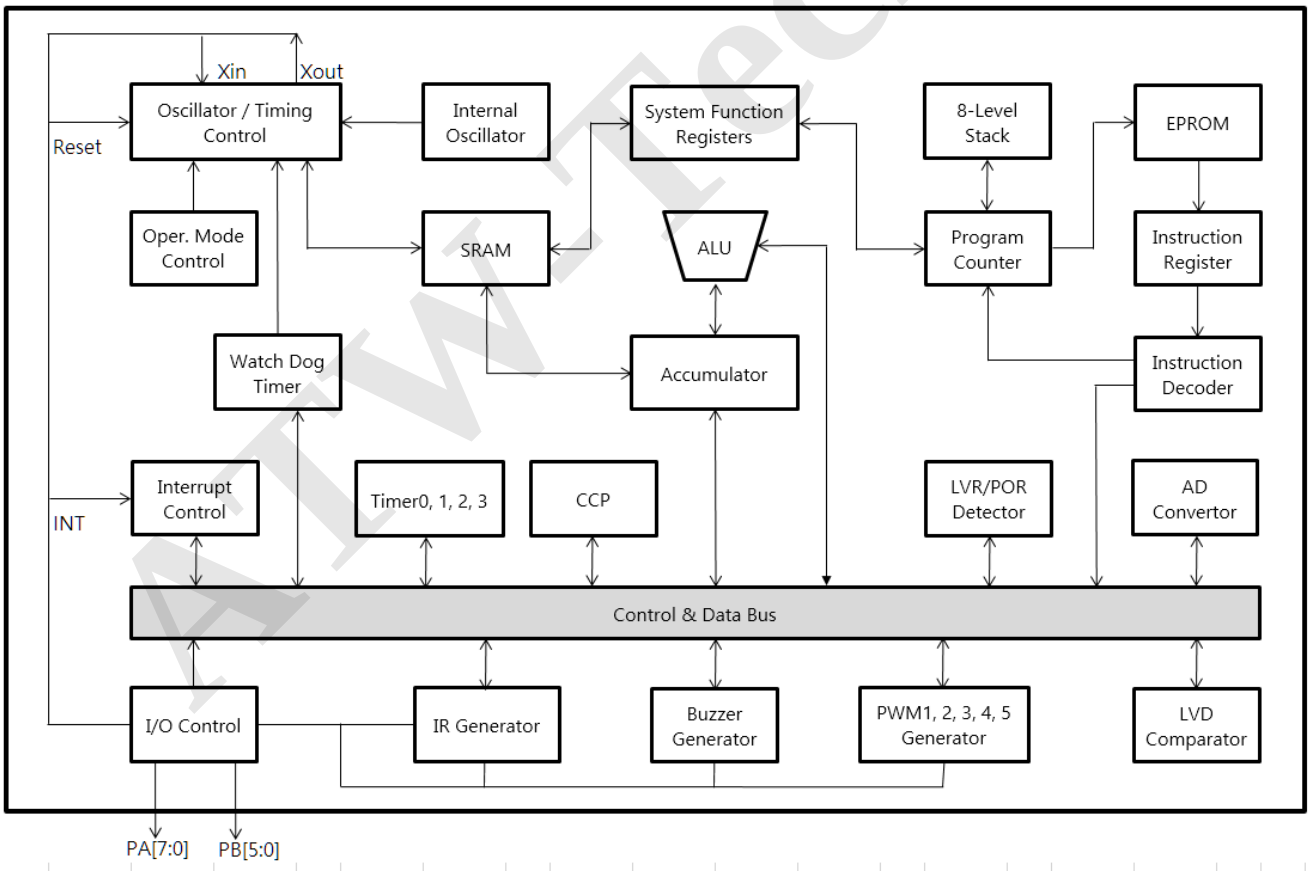
1.1 Features

- Wide operating voltage range:
  - 2.7V ~ 5.5V @20MHz/2T.
  - 1.6V ~ 5.5V @4MHz/4T.
- Wide operating temperature: -40°C ~ 85°C.
- 2K x 14 bits EPROM.
- 128 bytes SRAM.

- 14 general purpose I/O pins (GPIO), PA [7:0], PB[5:0], with independent direction control.
- PA[5, 3:0] and PB[3:0] have features of Pull-Low resistor for input pin.
- PA4 have high-level hold with 1MΩ pull-up resistor.
- PA[7:0] and PB[5:0] have features of Pull-High resistor.
- PB[5:0] have features of Open-Drain output.
- PA[5] have feature of input or open-drain output.
- I/O ports output current mode can be normal sink or large sink (exclude PA5).
- 8-level hardware Stack.
- Direct and indirect addressing modes for data access.
- One 8-bit up-count timer (Timer0) with programmable prescaler.
- Three 10-bit reload or continuous down-count timers (Timer1, 2, 3). Timer1/2/3 can be configured with a frequency pre-scaler or operate without frequency pre-scaler.
- Five 10-bit resolution PWM (1, 2, 3, 4, 5) output. PWM1 use Timer1; PWM2 use Timer2; PWM3/4/5 use Timer3. All PWM outputs can use internal high frequency (16M, 20MHz) as the clock source for Timer.
- One full function CCP(Compare/Capture/PWM). PWM provide Full-bridged PWM and Half-bridged PWM with dead band functions
- Three buzzer (BZ1, 2, 3) 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 11+1 channel high-precision 12-bit ADC.
- Built-in high-precision Voltage Comparator.
- Built-in Power-On Reset (POR).
- Built-in Low-Voltage Reset (LVR).
- Built-in Watch-Dog Timer (WDT) enabled/disabled by firmware control.
- Built-in Resistance to Frequency Converter (RFC) function.
- Dual-clock oscillation: System clock can switch between high oscillation and low oscillation.
  - High oscillation: 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.

- Twelve hardware interrupt events:
  - Timer0 overflow interrupt.
  - Timer1 underflow interrupt.
  - Timer2 underflow interrupt or CCP interrupt
  - Timer3 underflow interrupt.
  - WDT timeout interrupt.
  - PA/PB input change interrupt.
  - 2 set External interrupt(INT0/INT1).
  - LVD interrupt.
  - Comparator output status change interrupt.
  - ADC end-of-convert interrupt.
- Twelve interrupt events to wake-up AT8B62F1 from Standby mode:
  - Timer0 overflow interrupt.
  - Timer1 underflow interrupt.
  - Timer2 underflow interrupt or CCP interrupt.
  - Timer3 underflow interrupt.
  - WDT timeout interrupt.
  - PA/PB input change interrupt.
  - 2 set External interrupt(INT0/INT1).
  - LVD interrupt.
  - Comparator output status change interrupt.
  - ADC end-of-convert interrupt.
- Four interrupt events to wake-up AT8B62F1 from Halt mode:
  - WDT timeout interrupt.
  - PA/PB input change interrupt.
  - 2 set External interrupt(INT0/INT1).

1.2 Block Diagram



1.3 Pin Assignment

AT8B62F1 provides three kinds of package type which are SOP16, SOP14 and SOP8.

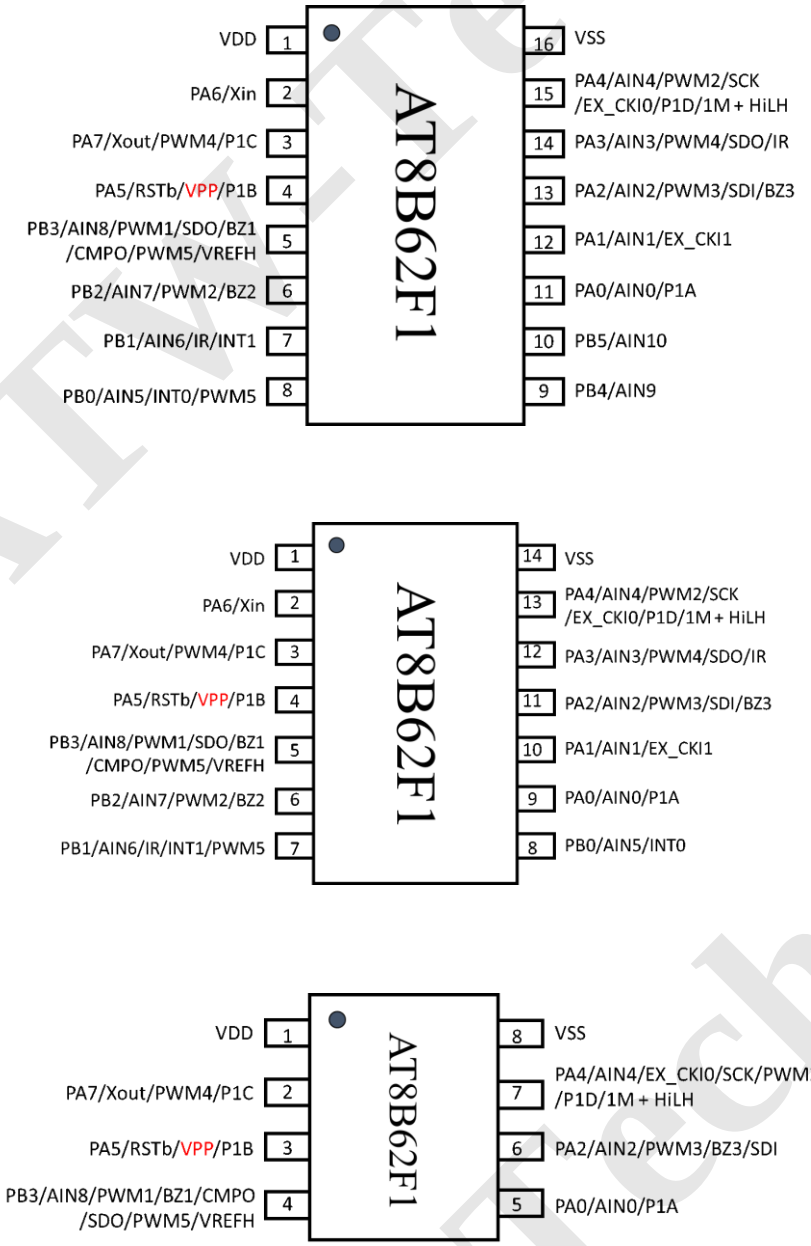


Figure 1 Package pin assignment

1.4 Pin Description

Pin Name	I/O	Description
PA0 AIN0 P1A	I/O	PA0 is bidirectional I/O pin, and can be comparator analog input pins. PA0 can be ADC analog input pin, AN0. PA0 can be P1A output in CCP mode
PA1 AIN1 EX_CKI1	I/O	PA1 is bidirectional I/O pin, and can be comparator analog input pins. PA1 can be ADC analog input pin, AN1 PA1 can be Timer2/3 clock source EX_CKI1.
PA2 AIN2 PWM3/BZ3 SDI	I/O	PA2 is a bidirectional I/O pin, and can be comparator analog input pin. PA2 can be ADC analog input pin, AN2. PA2 can be the output of PWM3 or Buzzer3. PA2 can be programming pad SDI.
PA3 AIN3 PWM4 IR SDO	I/O	PA3 is a bidirectional I/O pin, and can be comparator analog input pin. PA3 can be ADC analog input pin, AN3. PA3 can be the output of PWM4 If IR mode is enabled, this pin is IR carrier output. PA3 also can be programming pad SDO.
PA4 AIN4 PWM2 EX_CKI0 SCK P1D	I/O	PA4 is a bidirectional I/O pin, and have 1MΩ pull-up resistor. PA4 can be ADC analog input pin, AN4. PA4 can be the output of PWM2 PA4 can be the Timer0/1 clock source EX_CKI0. PA4 can be programming pad SCK. PA4 can be P1D output in CCP mode
PA5 RSTb Vpp P1B	I/O	PA5 is an input pin or open-drain output pin. PA5 can be the reset pin RSTb. If Vpp is more than 7.75V, it also makes IC enter EPROM programming mode. PA5 can be P1B output in CCP mode
PA6 Xin	I/O	PA6 is a bidirectional I/O pin. PA6 can be the input pin of crystal oscillator Xin.
PA7 PWM4 Xout P1C	I/O	PA7 is a bidirectional I/O pin. PA7 can be the output of PWM4 PA7 can be the output pin of crystal oscillator Xout. PA7 can be P1C output in CCP mode
PB0 AIN5 INT0 PWM5	I/O	PB0 is a bidirectional I/O pin. PB0 can be ADC analog input pin, AN5. PB0 can be the input pin of external interrupt INT0. PB0 can be the output of PWM5
PB1 AIN6 IR INT1	I/O	PB1 is a bidirectional I/O pin. PB1 can be ADC analog input pin, AN6. If IR mode is enabled, this pin is IR carrier output. PB1 can be the input pin of external interrupt INT1.
PB2 AIN7 PWM2/BZ2	I/O	PB2 is a bidirectional I/O pin. PB2 can be ADC analog input pin, AN7. PB2 can be the output of PWM2 or Buzzer2.

Pin Name	I/O	Description
PB3 AIN8 PWM1/BZ1 CMPO SDO PWM5 VREFH	I/O	PB3 is a bidirectional I/O pin. PB3 can be ADC analog input pin, AN8. PB3 can be the output of PWM1 or Buzzer1 PB3 can be the output of comparator. PB3 can be programming pad SDO. PB3 can be the output of PWM5 PB3 can be ADC external high reference voltage source.
PB4 AIN9	I/O	PB4 is a bidirectional I/O pin. PB4 can be ADC analog input pin, AN9.
PB5 AIN10	I/O	PB5 is a bidirectional I/O pin. PB5 can be ADC analog input pin, AN10.
VDD	-	Positive power supply.
VSS	-	Ground.

2. Memory Organization

AT8B62F1 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 AT8B62F1 is 2K words. Therefore, the Program Counter (PC) is 11-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.

AT8B62F1 provides instruction GOTOA, CALLA to address 256 location of program space. AT8B62F1 also provides instructions LCALL and LGOTO to address any location of program space.

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

AT8B62F1 program ROM address 0x7FE~0x7FF are reserved space, if user tries to write code in these addresses will get unexpected false functions.

AT8B62F1 program ROM address 0x00E~0x00F are preset rolling code can be released and used as normal program space.

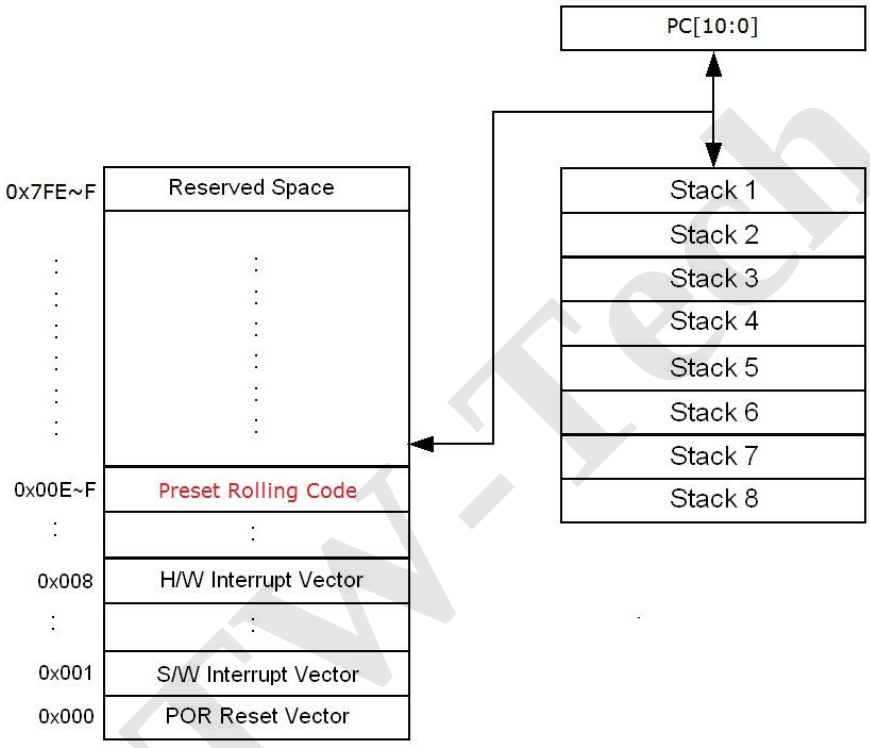


Figure 2 Program Memory Address Mapping

2.2 Data Memory

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

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

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

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

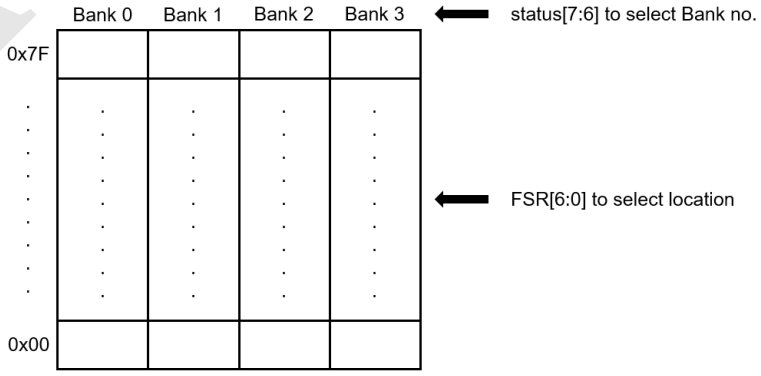


Figure 3 Indirect Addressing Mode of Data Memory Access

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

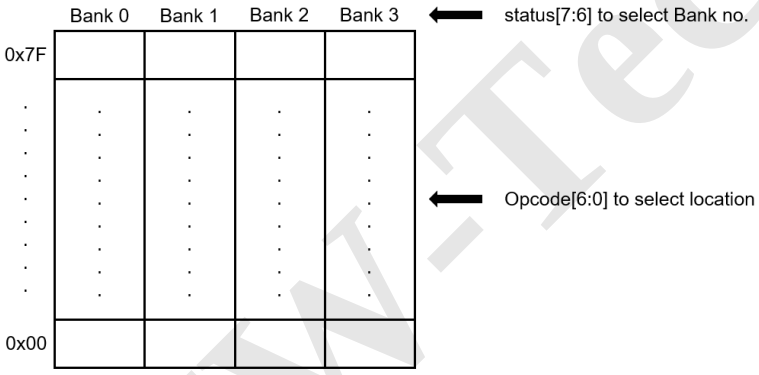


Figure 4 Direct Addressing Mode of Data Memory Access

R-page SFR can be accessed by general instructions like arithmetic instructions and data movement instructions. The R-page SFR occupies address from 0x0 to 0x1F of Bank 0. However, the same address range of Bank 1, Bank 2 and Bank 3 are mapped back to Bank 0. In other words, R-page SFR physically existed at

Bank 0. The GPR physically occupy address from 0x20 to 0x7F of Bank 0 and 0x20 to 0x3F of Bank 1. Other bank in address from 0x20 to 0x7F are mapped back as the Table 1 shows.

The AT8B62F1 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	-			
0x8	PCON			
0x9	BWUCON			
0xA	PCHBUF			
0xB	ABPLCON			
0xC	BPHCON			
0xD	-			
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	TM3RH			
0x1D ~0x1E	-	-		
0x1F	INTE2	*The same mapping as Bank 0*		
0x20 ~ 0x3F	General Purpose Register	General Purpose Register	*Mapped to bank0*	*Mapped to Bank1*
0x40 ~ 0x7F	General Purpose Register	*Mapped to bank0*	*Mapped to bank0*	*Mapped to bank0*

Table 1 R-page SFR Address Mapping

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

SFR Category / Address	F-page SFR	S-page SFR
0x0	-	TMR1
0x1	-	T1CR1
0x2	-	T1CR2
0x3	-	PWM1DUTY
0x4	-	PS1CV
0x5	IOSTA	BZ1CR
0x6	IOSTB	IRCR
0x7	-	TBHP
0x8	-	TBHD
0x9	APHCON	TMR2
0xA	PS0CV	T2CR1
0xB	-	T2CR2
0xC	BODCON	PWM2DUTY
0xD	-	PS2CV
0xE	CMPCR	BZ2CR
0xF	PCON1	OSCCR
0X10	-	TMR3
0X11	-	T3CR1
0X12	-	T3CR2
0X13	-	PWM3DUTY
0X14	-	PS3CV
0X15	-	BZ3CR
0X16	-	P4CR1
0X17	-	-
0X18	-	PWM4DUTY
0x19		CCPCON
0X1A		PWMDB
0X1B		P5CR1
0X1D		PWM5DUTY
0X1F		PWM5RH

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

3. Function Description

This chapter will describe the detailed operations of AT8B62F1.

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[10:0])

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							

Bit[7:0] The register PCL is the least significant byte (LSB) of 11-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[10:8], is not directly accessible. Update of PC[10:8] must be done through register PCHBUF.

For LGOTO instruction, PC[10:0] is from instruction word.

For LCALL instruction, PC[10:0] 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.

Bit0 **C:** Carry/Borrow flag

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.

Bit1 **DC:** Half Carry/half Borrow flag

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.

Bit2 **Z:** Zero flag

Z=1, result of logical operation is zero.

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

Bit3 **/PD:** Power down flag bit

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

/PD=0, after instruction SLEEP is executed.

Bit4 **/TO:** Time overflow flag

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

/TO=0, WDT timeout is occurred.

Bit5 **GP5:** General purpose read/write register bit.

Bit[7:6] **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.

Bit7 **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	GP7	GP6	PB5	PB4	PB3	PB2	PB1	PB0
R/W Property			R/W							
Initial Value			Data latch value is xxxxxxxx, read value is xxxxxxxx port value(PB5~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.

Bit[7:6] **GP7~6 :** general purpose register bits.

3.1.8 PCON (Power Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	R	0x8	WDTEN	GP6	LVDEN	/PHPA5	LVREN	GP2	GP1	GP0
R/W Property			R/W							
Initial Value			1	0	0	1	1	0	0	0

Bit[2:0] **GP2, GP1, GP0:** General read/write register bits.

Bit3 **LVREN:** Enable/disable LVR.

LVREN=1, enable LVR.

LVREN=0, disable LVR.

Bit4 **/PHPA5:** Disable/enable PA5 Pull-High resistor.

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

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

Bit5 **LVDEN:** Enable/disable LVD.

LVDEN=1, enable LVD.

LVDEN=0, disable LVD.

Bit6 **GP6:** General purpose register bits

Bit7 **WDTEN:** Enable/disable WDT.

WDTEN=1, enable WDT.

WDTEN=0, disable WDT.

3.1.9 BWUCON (PortB Wake-up Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BWUCON	R	0x9	-	-	WUPB5	WUPB4	WUPB3	WUPB2	WUPB1	WUPB0
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

Bit[5:0] **WUPBx:** Enable/disable PBx wake-up function, 0 ≤ x ≤ 5.

WUPBx=1, enable PBx wake-up function.

WUPBx=0, disable PBx wake-up function.

3.1.10 PCHBUF (High Byte of PC)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCHBUF	R	0xA	-	XSPD_STP	-			PCHBUF[2:0]		
R/W Property			-	W	-			R/W		
Initial Value			X	0	X			000		

Bit[2:0] **PCHBUF[2:0]**: Buffer of the 10^{th} ~ 8^{th} bit of PC.

Bit6 **XSPD_STP:** Write 1 to stop crystal 32.768K speed-up function, write-only.

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

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ABPLCON	R	0xB	/PLPB3	/PLPB2	/PLPB1	/PLPB0	/PLPA3	/PLPA2	/PLPA1	/PLPA0
R/W Property			R/W							
Initial Value			1	1	1	1	1	1	1	1

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

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

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

Bit[7:4] **/PLPBx:** Disable/enable PBx Pull-Low resistor, 0 ≤ x ≤ 3.

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

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

3.1.12 BPHCON (PortB Pull-High Resistor Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BPHCON	R	0xC	-	-	/PHPB5	/PHPB4	/PHPB3	/PHPB2	/PHPB1	/PHPB0
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

Bit[5:0] **/PHPBx:** Disable/enable PBx Pull-High resistor, 0 ≤ x ≤ 5.

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

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

3.1.13 INTE (Interrupt Enable Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTE	R	0xE	INT1IE	WDTIE	T2IE/ CCPIE	LVDIE	T1IE	INT0IE	PABIE	T0IE
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

Bit0 **T0IE:** Timer0 overflow interrupt enable bit.

T0IE=1, enable Timer0 overflow interrupt.

T0IE=0, disable Timer0 overflow interrupt.

Bit1 **PABIE:** PortA/PortB input change interrupt enable bit.

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

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

Bit2 **INT0IE:** External interrupt 0 enable bit.

INT0IE=1, enable external interrupt 0.

INT0IE=0, disable external interrupt 0.

Bit3 **T1IE:** Timer1 underflow interrupt enable bit.

T1IE=1, enable Timer1 underflow interrupt.

T1IE=0, disable Timer1 underflow interrupt.

Bit4 **LVDIE:** Low-voltage detector interrupt enable bit.

LVDIE=1, enable low-voltage detector interrupt.

LVDIE=0, disable low-voltage detector interrupt.

Bit5 **T2IE/CCPIE:** When CCP capture or compare mode is enabled, this interrupt is used as CCP interrupt enable bits, otherwise it is Timer2 underflow interrupt enable bit.

T2IE=1, enable interrupt.

T2IE=0, disable interrupt.

Bit6 **WDTIE:** WDT timeout interrupt enable bit.

WDTIE=1, enable WDT timeout interrupt.

WDTIE=0, disable WDT timeout interrupt.

Bit7 **INT1IE:** External interrupt 1 enable bit.

INT1IE=1, enable external interrupt 1.

INT1IE=0, disable external interrupt 1.

3.1.14 INTF (Interrupt Flag Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF	R	0xF	INT1IF	WDTIF	T2IF/ CCPIF	LVDIF	T1IF	INT0IF	PABIF	T0IF
R/W Property			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value(note*)			0	0	0	0	0	0	0	0

Bit0 **T0IF:** Timer0 overflow interrupt flag bit.

T0IF=1, Timer0 overflow interrupt is occurred.

T0IF =0 must be clear by firmware.

Bit1 **PABIF:** PortA/PortB input change interrupt flag.

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

PABIF=0 must be clear by firmware.

Bit2 **INT0IF:** External interrupt 0 flag.

INT0IF=1, external interrupt 0 is occurred.

INT0IF=0 must be clear by firmware.

Bit3 **T1IF:** Timer1 underflow interrupt flag.

T1IF=1 Timer1 underflow interrupt is occurred.

T1IF=0 must be clear by firmware.

Bit4 **LVDIF:** Low-voltage detector interrupt flag.

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

LVDIF=0 must be clear by firmware.

Bit5 **T2IF/CCPIF:** When CCP capture or compare mode is enabled, this interrupt is used as CCP interrupt flag bits, otherwise it is T2 underflow interrupt flag

T2IF=1, interrupt is occurred.

T2IF=0 must be clear by firmware.

Bit6 **WDTIF**: WDT timeout interrupt flag.

WDTIF=1, WDT timeout interrupt is occurred.

WDTIF=0 must be clear by firmware.

Bit7 **INT1IF**: External interrupt 1 flag.

INT1IF=1, external interrupt 1 is occurred.

INT1IF =0 must be clear by firmware.

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

3.1.15 ADMD (ADC Mode Register)

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

Bit[3:0] **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 PB0 pad as ADC input,

0110=select PB1 pad as ADC input,

0111=select PB2pad as ADC input,

1000=select PB3 pad as ADC input,

1001=select PB4 pad as ADC input,

1010=select PB5 pad as ADC input,

1011=select 1/4 VDD as ADC input.

1100=select GND as ADC input.

Bit4 **GCHS:** ADC global channel select bit.

GCHS=0 : disable all ADC input channel.

GCHS=1 : enable ADC input channel.

Bit5 **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.

Bit6 **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.

Bit7 **ADEN:** ADC enable bit.

ADEN=1, ADC is enabled.

3.1.16 ADR (ADC Clock, ADC Interrupt Flag and ADC LSB Output Register)

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

Bit[3:0] **AD3~0:** 12-bit low-nibble ADC data buffer.

Bit[5:4] **ADCK1~0:** ADC clock select.

00: ADC clock=F_{INST}/16,

01: ADC clock=F_{INST}/8,

10: ADC clock=F_{INST}/1,

11: ADC clock=F_{INST}/2.

Bit6 **ADIE:** ADC end-of-convert interrupt enable bit.

ADIE=1 : enable ADC interrupt.

ADIE=0 : disable ADC interrupt.

Bit7 **ADIF**: ADC interrupt flag bit.

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

ADIF=0 must be clear by firmware.

3.1.17 ADD (ADC Output Data Register)

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

Bit[7:0] **AD11~4:** High-byte ADC data buffer.

3.1.18 ADVREFH (ADC High Reference Voltage Register)

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

Bit7 **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.

Bit [1:0] **VHS1~0:** ADC internal reference high voltage select bits.

11: VREFH=VDD,

10: VREFH=4V,

01 VREFH=3V

00: VREFH=2V.

3.1.19 ADCR (Sampling Pulse and ADC Bit Register)

Name	SFR Type	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCR	R	0x14	-	PBCON5	PBCON4	PBCON3	SHCK1	SHCK0	ADCR1	ADCR0
R/W Property			-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			X	0	0	0	1	0	1	0

Bit[1:0] **ADCR1~0:** ADC conversion bit no. select.

00: 8-bit ADC

01: 10-bit ADC

1x: 12-bit ADC.

Bit[3:2] **SHCK1~0:** Sampling pulse width select.

00: 1 ADC clock

01: 2 ADC clock

10: 4 ADC clock

11: 8 ADC clock.

Bit[6:4] **PBCONx:** PB analog pin select, 3 ≤ x ≤ 5.

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

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

3.1.20 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			1	1	1	1	1	1	1	1

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

WUPAx=1, enable PAx wake-up function.

WUPAx=0, disable PAx wake-up function.

3.1.21 PACON (ADC Analog Pin Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PACON	R	0x16	PBCON2	PBCON1	PBCON0	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

Bit[4:0] **PACONx:** PA 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.

Bit[7:5] **PBCONx:** PB analog pin select, 0 ≤ x ≤ 2.

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

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

3.1.22 ADJMD (ADC adjustment mode)

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

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

00000=offset 0mV

11111=offset 12.5mV.

Bit5 **ADJ_SIGN:** adjustment sign bit

0= adc data decrease

1=adc data increase

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

3.1.23 INTEDG (Interrupt Edge Register)

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

Bit[1:0] **INT0G1~0:** INT0 edge trigger select bit.

00: reserved

01: rising edge

10: falling edge

11: rising/falling edge.

Bit[3:2] **INT1G1~0:** INT1 edge trigger select bit.

00: reserved

01: rising edge

10: falling edge

11: rising/falling edge.

Bit4 **EIS0:** External interrupt 0 select bit

EIS0=1, PB0 is external interrupt 0.

EIS0=0, PB0 is GPIO.

Bit5 **EIS1:** External interrupt 1 select bit

EIS1=1, PB1 is external interrupt 1.

EIS1=0, PB1 is GPIO.

3.1.24 TMRH (Timer High Byte Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMRH	R	0x19	TMR29	TMR28	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	R/W	R/W
Initial Value			X	X	X	X	X	X	X	X

Bit[1:0] **PWM1DUTY9~8:** PWM1 duty data MSB 2 bits.

Bit[3:2] **PWM2DUTY9~8:** PWM2 duty data MSB 2 bits.

Bit[5:4] **TMR19~8:** Timer1 MSB 2 bits. Write these 2 bits will overwrite the 10-bit Timer1 load value of bit 9 and 8. Read these 2 bits will get the Timer1 bit9-8 current value.

Bit[7:6] **TMR29~8:** Timer2 MSB 2 bits. Write these 2 bits will overwrite the 10-bit Timer2 load value of bit 9 and 8. Read these 2 bits will get the Timer2 bit9-8 current value.

3.1.25 ANAEN (Analog Circuit Enable Register)

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

Bit7 **CMPEN:** Enable/disable voltage comparator.

CMPEN=1, enable voltage comparator.

CMPEN=0, disable voltage comparator.

3.1.26 RFC (RFC Register)

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

Bit[3:0] **RFCEN:** Enable/disable RFC function.

RFCEN=1, enable RFC function.

RFCEN=0, disable RFC function.

Bit7 **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

Table 3 RFC pad select

3.1.27 TM3RH (Timer3 High Byte Register)

Name	SFR Type	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TM3RH	R	0x1C	-	-	TMR39	TMR38	PWM4 DUTY9	PWM4 DUTY8	PWM3 DUTY9	PWM3 DUTY8
R/W Property			-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			-	-	X	X	X	X	X	X

Bit[1:0] **PWM3DUTY9~8:** PWM3 duty data MSB 2 bits.

Bit[3:2] **PWM4DUTY9~8:** PWM4 duty data MSB 2 bits.

Bit[5:4] **TMR39~38:** Timer3 MSB 2 bits. Write these 2 bits will overwrite the 10-bit Timer3 load value of bit 9 and 8. Read these 2 bits will get the Timer3 bit9-8 current value.

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

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTE2	R	0x1F	-	-	-	T3IF	-	-	-	T3IE
R/W Property			-	-	-	R/W	-	-	-	R/W
Initial Value			-	-	-	0	-	-	-	0

Bit0 **T3IE:** Timer3 underflow interrupt enable bit.

T3IE=1, enable Timer3 underflow interrupt.

T3IE=0, disable Timer3 underflow interrupt.

Bit4 **T3IF:** Timer3 underflow interrupt flag bit.

T3IF=1, Timer3 underflow interrupt is occurred.

T3IF=0 must be clear by firmware.

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		

Bit[2:0] **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

Bit3 **PS0WDT:** Prescaler0 assignment.

PS0WDT=1, Prescaler0 is assigned to WDT.

PS0WDT=0, Prescaler0 is assigned to Timer0.

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

Bit4 **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.

Bit5 **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.

Bit6 **GP6:** General register.

Bit7 **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

Bit[7:0] **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	-	-	IOPB5	IOPB4	IOPB3	IOPB2	IOPB1	IOPB0
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

Bit[5:0] **IOPBx:** PBx I/O mode selection, 0 ≤ x ≤ 5.

IOPBx=1, PBx is input mode.

IOPBx=0, PBx is output mode.

3.3.3 APHCON (PortA Pull-High Resistor Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
APHCON	F	0x9	/PHPA7	/PHPA6	/PLPA5	/PHPA4	/PHPA3	/PHPA2	/PHPA1	/PHPA0
R/W Property			R/W							
Initial Value			1	1	1	1	1	1	1	1

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

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

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

Bit5 **/PLPA5:** Enable/disable Pull-Low resistor of PA5.

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

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

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

3.3.4 PS0CV (Prescaler0 Counter Value Register)

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

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

3.3.5 BODCON (PortB Open-Drain Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BODCON	F	0xC	-	-	ODPB5	ODPB4	ODPB3	ODPB2	ODPB1	ODPB0
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

Bit[5:0] **ODPBx:** Enable/disable open-drain of PBx, 0 ≤ x ≤ 5.

ODPBx=1, enable open-drain of PBx.

ODPBx=0, disable open-drain of PBx.

3.3.6 CMPCR (Comparator voltage select Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPCR	F	0xE	PS3	PS2	PS1	PS0	VS3	VS2	VS1	VS0
R/W Property			R/W							
Initial Value			0	0	0	0	1	1	0	0

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

When the comparator is in P2V mode, VS[3:0] select one of 15 reference voltages as the inverting input of the comparator. And PS[3:0] determine one of 11 pads as the non-inverting input of the comparator.

When the comparator is in P2P mode, VS[3:0] is fixed 0, and PS[3:0] select 2 pads out of 4 combinations to be the inverting and non-inverting input of the comparator. For detail P2P mode please see function description comparator section.

VS[3:0]	V- of Comparator	PS[3:0]	Selected pad
0000	P2P mode	0000	PA0
0001	1 / 16 V_{DD}	0001	PA1
0010	2 / 16 V_{DD}	0010	PA2
0011	3 / 16 V_{DD}	0011	PA3
0100	4 / 16 V_{DD}	0100	-
0101	5 / 16 V_{DD}	0101	-
0110	6 / 16 V_{DD}	0110	-
0111	7 / 16 V_{DD}	0111	-
1000	8 / 16 V_{DD}	1000	-
1001	9 / 16 V_{DD}	1001	-
1010	10 / 16 V_{DD}	1010	-
1011	11 / 16 V_{DD}	1011	-
1100	12 / 16 V_{DD}	1100	-
1101	13 / 16 V_{DD}	1101	-
1110	14 / 16 V_{DD}	1110	-
1111	15 / 16 V_{DD}	1111	-

Table 5 P2V Mode

PS[3:0]	Non-inverting input	Inverting input
0000	PA0	PA1
0001	PA1	PA0
0010	PA2	PA3
0011	PA3	PA2

Table 6 P2P Mode (VS[3:0] = 4'b0000)

3.3.7 PCON1 (Power Control Register1)

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

Bit0 **T0EN:** Enable/disable Timer0.

T0EN=1, enable Timer0.

T0EN=0, disable Timer0

Bit1 **GP1:** General purpose read/write register.

Bit[4:2] **LVDS2~0:** Select one of the 8 LVD voltage.

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

Table 7 LVD voltage select

Bit5 **GP5:** General purpose read/write register.

Bit6 **LVDOUT:** Low voltage detector output, read-only.

Bit7 **GIE:** Global interrupt enable bit.

GIE=1, enable all unmasked interrupts.

GIE=0, disable all interrupts.

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

3.4 S-page Special Function Register

3.4.1 TMR1 (Timer1 Register)

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

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

3.4.2 T1CR1 (Timer1 Control Register1)

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

This register is used to configure Timer1 functionality.

Bit0 **T1EN:** Enable/disable Timer1.

T1EN=1, enable Timer1.

T1EN=0, disable Timer1.

Bit1 **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.

Bit2 **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

Bit3 **TM1_HRC:** Timer1 clock source from Internal High frequency(I_HRCx1)

TM1_HRC=1: Enable. Timer1 clock source from Internal High frequency (I_HRCx1)

TM1_HRC=0: Disable. Timer1 clock source from CPU operating clock.

Note: When Timer By Passed prescaler (TMx_HRC=1), it's minimum operating LVR voltage will be higher.

Bit6 **PWM1OAL:** Define PWM1 output active state.

PWM1OAL=1, PWM1 output is active low.

PWM1OAL=0, PWM1 output is active high.

Bit7 **PWM1OEN:** Enable/disable PWM1 output.

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

PWM1OEN=0, PB3 is GPIO.

3.4.3 T1CR2 (Timer1 Control Register2)

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

This register is used to configure Timer1 functionality.

Bit[2:0] **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.

Bit3 **/PS1EN:** Disable/enable Prescaler1.

/PS1EN=1, disable Prescaler1.

/PS1EN=0, enable Prescaler1.

Bit4 **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.

Bit5 **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

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

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

Table 10 Buzzer1 Output Frequency Selection

Bit7 **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

Bit0 **IREN:** Enable/Disable IR carrier output.

IREN=1, enable IR carrier output.

IREN=0, disable IR carrier output.

Bit1 **IRF57K:** Selection of IR carrier frequency.

IRF57K=1, IR carrier frequency is 57KHz.

IRF57K=0, IR carrier frequency is 38KHz.

Bit2 **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.

Bit7 **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	-	-	-	-	-	TBHP2	TBHP1	TBHP0
R/W Property			-	-	-	-	-	R/W	R/W	R/W
Initial Value			X	X	X	X	X	X	X	X

When instruction CALLA, GOTOA or TABLEA is executed, the target address is constituted by TBHP[2:0] and ACC. ACC is the Low Byte of PC[10:0] and TBHP[2:0] is the high byte of PC[10:0].

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 TMR2 (Timer2 Register)

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

When reading register TMR2, it will obtain current value of 10-bit down-count Timer2 at TMR2[7:0]. When writing register TMR2, it will write data from TMRH[7:6] and Timer2 reload register to Timer2[9:0] current content.

3.4.11 T2CR1 (Timer2 Control Register1)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2CR1	S	0xA	PWM2OEN	PWM2OAL	-	-	TM2_HRC	T2OS	T2RL	T2EN
R/W Property			R/W	R/W	-	-	R/W	R/W	R/W	R/W
Initial Value			0	0	X	X	0	0	0	0

This register is used to configure Timer2 functionality.

Bit0 **T2EN:** Enable/disable Timer2.

T2EN=1, enable Timer2.

T2EN=0, disable Timer2.

Bit1 **T2RL:** Configure Timer2 down-count mechanism while Non-Stop mode is selected (T2OS=0).

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

T2RL=0, continuous down-count from 0x3FF when underflow is occurred.

Bit2 **T2OS:** Configure Timer2 operating mode while underflow is reached.

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

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

Bit3 **TM2_HRC:** Timer2 clock source from Internal High frequency(I_HRCx1)

TM2_HRC=1 Enable ,Timer2 clock source from Internal High frequency(I_HRCx1).

TM2_HRC=0 Disable ,Timer2 clock source from CPU operating clock.

Note: When Timer By Passed prescaler(TMx_HRC=1), it's minimum operating LVR voltage will be higher.

T2OS	T2RL	Timer2 Down-Count Functionality
0	0	Timer2 will count from reload value down to 0x00. When underflow is reached, 0x3FF is reloaded and continues down-count.
0	1	Timer2 will count from reload value down to 0x00. When underflow is reached, reload value is reloaded and continues to down-count.
1	x	Timer2 will count from initial value down to 0x00. When underflow is reached, Timer2 will stop down-count.

Table 12 Timer2 Functionality

Bit6 **PWM2OAL:** Define PWM2 output active state.

PWM2OAL=1, PWM2 output is active low.

PWM2OAL=0, PWM2 output is active high.

Bit7 **PWM2OEN**: Enable/disable PWM2 output.

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

PWM2OEN=0, PB2 is GPIO.

3.4.12 T2CR2 (Timer2 Control Register2)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2CR2	S	0xB	-	-	T2CS	T2CE	/PS2EN	PS2SEL[2:0]		
R/W Property			-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value			X	X	1	1	1	1	1	1

This register is used to configure Timer2 functionality.

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

PS2SEL[2:0]	Dividing Rate
000	1:2
001	1:4
010	1:8
011	1:16
100	1:32
101	1:64
110	1:128
111	1:256

Table 13 Prescaler2 Dividing Rate

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

Bit3 **/PS2EN:** Disable/enable Prescaler2.

/PS2EN=1, disable Prescaler2.

/PS2EN=0, enable Prescaler2.

Bit4 **T2CE:** Timer2 external clock edge selection.

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

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

Bit5 **T2CS:** Timer2 clock source selection.

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

T2CS=0, Instruction clock is selected.

3.4.13 PWM2DUTY (PWM2 Duty Register)

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

The reload value of 10-bit Timer2 stored on registers TMRH[7:6] and TMR2[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.

3.4.14 PS2CV (Prescaler2 Counter Value Register)

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

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

3.4.15 BZ2CR (Buzzer2 Control Register)

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

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

BZ2FSEL[3:0]	BZ2 Frequency Selection	
	Clock Source	Dividing Rate
0000	Prescaler2 output	1:2
0001		1:4
0010		1:8
0011		1:16
0100		1:32
0101		1:64
0110		1:128
0111		1:256
1000	Timer2 output	Timer2 bit 0
1001		Timer2 bit 1
1010		Timer2 bit 2
1011		Timer2 bit 3
1100		Timer2 bit 4
1101		Timer2 bit 5
1110		Timer2 bit 6
1111		Timer2 bit 7

Table 14 Buzzer2 Output Frequency Selection

Bit7 **BZ2EN:** Enable/Disable BZ2 output.

BZ2EN=1, enable Buzzer2.

BZ2EN=0, disable Buzzer2.

3.4.16 OSCCR (Oscillation Control Register)

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

Bit0 **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}).

Bit1 **STPHOSC:** Disable/enable high-frequency oscillation (F_{HOSC}).

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

STPHOSC=0, F_{HOSC} keep oscillation.

Bit[3:2] **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]

Bit4 **CMPIE:** Enable/Disable of comparator interrupt.

CMPIE=1, Enable of comparator interrupt.

CMPIE=0, Disable of comparator interrupt.

Bit5 **CMPIF:** Comparator output change state interrupt is occurred.

CMPIF=1, comparator interrupt is occurred.

CMPIF must be clear by firmware.

Bit6 **CMPOE:** Disable/enable comparator output to pad PB3.

CMPOE=1, enable comparator output to pad PB3.

CMPOE=0, disable comparator output to pad PB3.

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

Bit7 **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 SELHOSC1.

3.4.17 TMR3 (Timer3 Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMR3	S	0x10	TMR3[7:0]							
R/W Property			R/W							
Initial Value			XXXXXXXX							

When reading register TMR3, it will obtain current value of 10-bit down-count Timer3 at TMR3[7:0]. When writing register TMR3, it will write data from TM3RH[5:4] and Timer3 reload register to Timer3[9:0] current content.

3.4.18 T3CR1 (Timer3 Control Register1)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T3CR1	S	0x11	PWM3OEN	PWM3OAL	-	-	TM3_HRC	T3OS	T3RL	T3EN
R/W Property			R/W	R/W	-	-	R/W	R/W	R/W	R/W
Initial Value			0	0	X	X	0	0	0	0

This register is used to configure Timer3 functionality.

Bit0 **T3EN:** Enable/disable Timer3.

T3EN=1, enable Timer3.

T3EN=0, disable Timer3.

Bit1 **T3RL:** Configure Timer3 down-count mechanism while Non-Stop mode is selected (T3OS=0).

T3RL=1, initial value is reloaded from reload register TMR3.

T3RL=0, continuous down-count from 0x3FF when underflow is occurred.

Bit2 **T3OS:** Configure Timer3 operating mode while underflow is reached.

T3OS=1, One-Shot mode. Timer3 will count once from the initial value to 0x00.

T3OS=0, Non-Stop mode. Timer3 will keep down-count after underflow.

T3OS	T3RL	Timer3 Down-Count Functionality
0	0	Timer3 will count from reload value down to 0x00. When underflow is reached, 0x3FF is reloaded and continues down-count.
0	1	Timer3 will count from reload value down to 0x00. When underflow is reached, reload value is reloaded and continues to down-count.
1	x	Timer3 will count from initial value down to 0x00. When underflow is reached, Timer3 will stop down-count.

Table 16 Timer3 Functionality

Bit3 **TM3_HRC:** Timer3 clock source from Internal High frequency (I_HRCx1).

TM3_HRC=1 Enable Timer3 clock source from Internal High frequency (I_HRCx1)

TM3_HRC=0 Disable, Timer3 clock source from CPU operating clock

Note: When Timer By Passed prescaler(TMx_HRC=1), it's minimum operating LVR voltage will be higher.

Bit6 **PWM3OAL:** Define PWM3 output active state.

PWM3OAL=1, PWM3 output is active low.

PWM3OAL=0, PWM3 output is active high.

Bit7 **PWM3OEN:** Enable/disable PWM3 output.

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

PWM3OEN=0, PA2 is GPIO.

3.4.19 T3CR2 (Timer3 Control Register2)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T3CR2	S	0x12	-	-	T3CS	T3CE	/PS3EN	PS3SEL[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 Timer3 functionality.

Bit[2:0] **PS3SEL[2:0]:** Prescaler3 dividing rate selection.

PS3SEL[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 17 Prescaler3 Dividing Rate

Note: Always set PS3SEL[2:0] at /PS3EN=1, or interrupt may be falsely triggered.

Bit3 **/PS3EN:** Disable/enable Prescaler3.

/PS3EN=1, disable Prescaler3.

/PS3EN=0, enable Prescaler3.

Bit4 **T3CE:** Timer3 external clock edge selection.

T3CE=1, Timer3 will decrease one while high-to-low transition occurs on pin EX_CKI1.

T3CE=0, Timer3 will decrease one while low-to-high transition occurs on pin EX_CKI1.

Bit5 **T3CS:** Timer3 clock source selection.

T3CS=1, External clock on pin EX_CKI1 is selected.

T3CS=0, Instruction clock is selected.

3.4.20 PWM3DUTY (PWM3 Duty Register)

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

The reload value of 10-bit Timer3 stored on registers TM3RH[5:4] and TMR3[7:0] is used to define the PWM3 frame rate, and registers TM3RH[1:0] and PWM3DUTY[7:0] is used to define the duty cycle of PWM3.

3.4.21 PS3CV (Prescaler3 Counter Value Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS3CV	S	0x14	PS3CV[7:0]							
R/W Property			R							
Initial Value			1	1	1	1	1	1	1	1

While reading PS3CV, it will get current value of Prescaler3 counter.

3.4.22 BZ3CR (Buzzer3 Control Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BZ3CR	S	0x15	BZ3EN	-	-	-	BZ3FSEL[3:0]			
R/W Property			W	-	-	-	W			
Initial Value			0	X	X	X	1	1	1	1

Bit[3:0] **BZ3FSEL[3:0]:** Frequency selection of BZ3 output.

BZ3FSEL[3:0]	BZ3 Frequency Selection	
	Clock Source	Dividing Rate
0000	Prescaler3 output	1:2
0001		1:4
0010		1:8
0011		1:16
0100		1:32
0101		1:64
0110		1:128
0111		1:256
1000	Timer3 output	Timer3 bit 0
1001		Timer3 bit 1
1010		Timer3 bit 2
1011		Timer3 bit 3
1100		Timer3 bit 4
1101		Timer3 bit 5
1110		Timer3 bit 6
1111		Timer3 bit 7

Table 18 Buzzer3 Output Frequency Selection

Bit7 **BZ3EN:** Enable/Disable BZ3 output.

BZ3EN=1, enable Buzzer3.

BZ3EN=0, disable Buzzer3.

3.4.23 P4CR1 (PWM4 Control Register1)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P4CR1	S	0x16	PWM4OEN	PWM4OAL	-	-	-	-	-	-
R/W Property			R/W	R/W	-	-	-	-	-	-
Initial Value			0	0	X	X	X	X	X	X

Bit6 **PWM4OAL:** Define PWM4 output active state.

PWM4OAL=1, PWM4 output is active low.

PWM4OAL=0, PWM4 output is active high.

Bit7 **PWM4OEN:** Enable/disable PWM4 output.

PWM4OEN=1, PWM4 output will be present on PA3 or PA7.

PWM2OEN=0, PA3 or PA7 is GPIO.

Note: PWM4 output pin is defined by NYIDE Config Block Setting.

3.4.24 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 Timer3 stored on registers TM3RH[5:4] and TMR3[7:0] is used to define the PWM4 frame rate, and registers TM3RH[3:2] and PWM4DUTY[7:0] is used to define the duty cycle of PWM4.

3.4.25 CCPCON (CCP Control Register)

Name	SFR Type	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCPCON	S	0x19	PWM2M1	PWM2M0	FBCH1	FBCH0	CCPM3	CCPM2	CCPM1	CCPM0
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

CCPM[3:2]=00/01/10: Capture or compare mode. PB2 is capture input or compare output.

CCPM[3:2]=11:

PWM2M1~0 = 00 → PWM single output.

PWM2M1~0 = 01 → PWM Full-Bridge output forward.

PWM2M1~0 = 10 → PWM Half-bridge output.

PWM2M1~0 = 11 → PWM Full-Bridge output reverse.

T2IF/CCPIF=1, interrupt is occurred.

Note: T2IF/CCPIF must be clear by firmware.

CCPM[3:0]:

0000 = CCP off.

0010 = Compare mode, toggle output on match.

0100 = Capture mode, capture at every falling edge.

0101 = Capture mode, capture at every rising edge.

0110 = Capture mode, capture at every 4th rising edge.

0111 = Capture mode, capture at every 16th rising edge.

1000 = Compare mode, set output and interrupt on match.

1001 = Compare mode, clear output and interrupt on match.

1010 = Compare mode, only interrupt on match.

1011 = Compare mode, trigger ADC and interrupt on match.

1100 = PWM mode, P1A/P1C active high, P1D/P1B active high.

1101 = PWM mode, P1A/P1C active high, P1D/P1B active low.

1110 = PWM mode, P1A/P1C active low, P1D/P1B active high.

1111 = PWM mode, P1A/P1C active low, P1D/P1B active low.

Note: P1A is PA0, P1B is PA5, P1C is PA7, P1D is PA4.

FBCH1~0: Full band change direction gap

00 = 1 CPU cycle.

01 = 4 CPU cycle.

1x = 16 CPU cycle.

Note: Compare/Capture mode step:

a. Set timer2 operating mode: set T1OS/T1RL, set T2EN = PWM2OEN = 0.

b. Set timer1 clock source, timer1 prescaler.

c. Set timer1 / timer2 reload/initial value.

d. Set pwm1duty / pwm2duty.

e. Enable compare / capture mode.

f. Disable compare / capture mode before update pwm / timer data.

3.4.26 PWMDB (Dead band control Register)

Name	SFR Type	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDB	S	0x1A	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			0	0	0	0	0	0	0	0

DB[7:0] Define the dead band width of the CCP PWM mode. The dead band unit is in CPU cycle.

td= F_{INST} *(DB[7:0])

3.4.27 P5CR1 (PWM5 Control Register1)

Name	SFR Type	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P5CR1	S	0x1B	PWM5OEN	PWM5OAL	-	-	-	-	-	-
R/W Property			R/W	R/W	-	-	-	-	-	-
Initial Value			0	0	X	X	X	X	X	X

Bit6 **PWM5OAL:** Define PWM5 output active state.

PWM5OAL=1, PWM5 output is active low.

PWM5OAL=0, PWM5 output is active high.

Bit7 **PWM5OEN:** Enable/disable PWM5 output.

PWM5OEN=1, PWM5 output will be present on PB3 or PB0.

PWM5OEN=0, PB3 or PB0 is GPIO.

Note: PWM5 output pin is defined by NYIDE Config Block Setting.

3.4.28 PWM5DUTY (PWM5 Duty Register)

Name	SFR Type	Addr.	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM5DUTY	S	0x1D	PWM5DUTY[7:0]							
R/W Property			W							
Initial Value			XXXXXXXX							

The reload value of 10-bit Timer3 stored on registers TM3RH[5:4] and TMR3[7:0] is used to define the PWM5 frame rate, and registers PWM5RH[1:0] and PWM5DUTY[7:0] is used to define the duty cycle of PWM5.

3.4.29 PWM5RH (PWM5 Duty Register) of MSB[9:8]

Name	SFR Type	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM5RH	S	0x1F	-	-	-	-	-	-	PWM5 DUTY9	PWM5 DUTY8
R/W Property			-	-	-	-	-	-	R/W	R/W
Initial Value			X	X	X	X	X	X	X	X

Bit[1:0] PWM5DUTY[9:8] is used to define the duty cycle MSB of PWM5.

3.5 I/O Port

AT8B62F1 provides 14 I/O pins which are PA[7:0] and PB[5:0]. User can read/write these I/O pins through registers PORTA and PORTB respectively. Each I/O pin has a corresponding register bit to define it is input pin or output pin. Register IOSTA[7:0] define the input/output direction of PA[7:0]. Register IOSTB[5:0] define the input/output direction of PB[5:0].

When an I/O pin is configured as input pin, it may have Pull-High resistor or Pull-Low resistor which is enabled or disabled through registers. Register APHCON[7:6, 4:0] are used to enable or disable Pull-High resistor of PA[7:6, 4:0]. Register APHCON[5] and ABPLCON[3:0] are used to enable or disable Pull-Low resistor of PA[5, 3:0]. Register BPHCON[5:0] are used to enable or disable Pull-High resistor of PB[5:0]. Register ABPLCON[7:4] are used to enable or disable Pull-Low resistor of PB[3:0]. PCON[4] is used to enable or disable Pull-High resistor of PA[5].

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[5:0] determine PB[5:0] is Open-Drain or not.

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

Feature		PA[3:0]	PA[7:6]&PA[4]	PA[5]	PB[3:0]	PB[5:4]
Input	Pull-High Resistor	V	V	V	V	V
	Pull-Low Resistor	V	X	V	V	X
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[5:0] 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 two external interrupt provided by AT8B62F1. When register bit EIS0 (INTEDG[4]) is set to 1, PB0 is used as input pin for external interrupt 0. When register bit EIS1 (INTEDG[5]) is set to 1, PB1 is used as input pin for external interrupt 1.

Note: When PB0 or PB1 is both set as level change operation and external interrupt, the external interrupt will have higher priority, and the PB0 or PB1 level change operation will be disabled. But PB5~PB2 level change function are not affected.

AT8B62F1 provides IR carrier generation output. When IREN=1, the IR carrier output will be present on PB1 pad. When IREN=0, the IR carrier will not be generated.

PA5 can be used as external reset input determined by a configuration word. When an active-low signal is applied to PA5, it will cause AT8B62F1 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. PA1 can be Timer2/Timer3 external clock source EX_CKI1 if T2CS/T3CS=1.

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

PB2 can be PWM2 output If T2CR1[7] PWM2OEN=1. PB2 can be Buzzer2 output if BZ2CR[7] BZ2EN=1. The output priority of PB2 is PWM2>Buzzer2.

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

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

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

3.5.1 Block Diagram of IO Pins

IO_SEL: set pad attribute as input or output

WRITE_EN: write data to pad.

READ_EN: read pad.

WUA: port A wake-up enable.

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.

SET_PAIF: port A wake-up flag.

PAEN:

PA0: CCP P1A function enable

PA2: PWM3/BZ3 function enable

PA3: PWM4/SDO function enable

PADT:

PA0DT: CCP P1A data

PA2DT: PWM3/BZ3 data

PA3DT: PWM4/SDO data

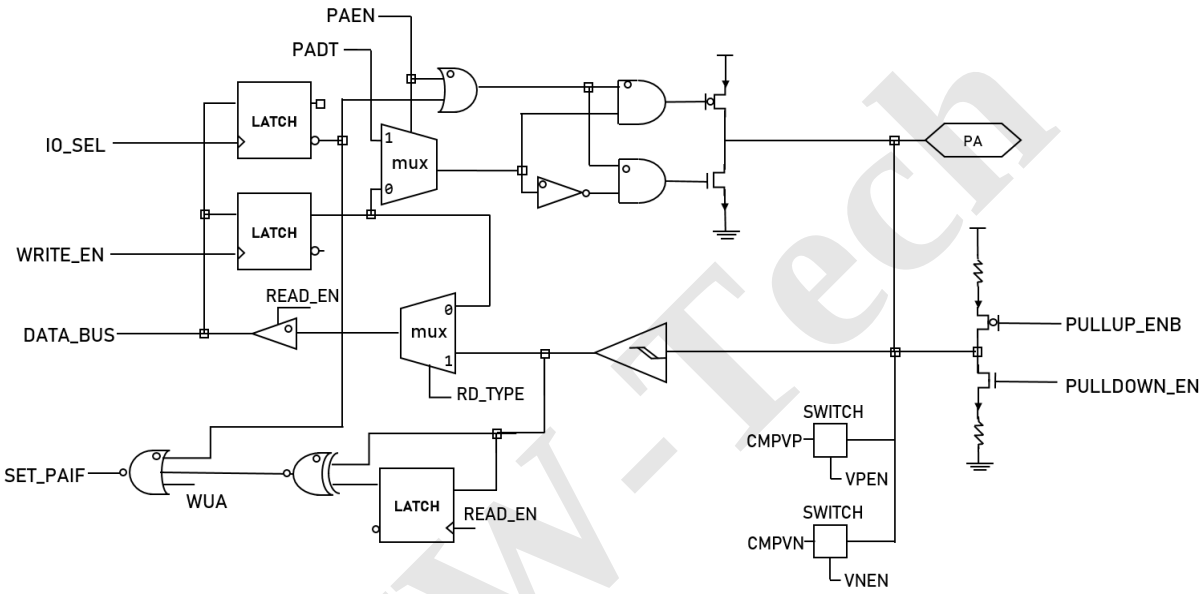


Figure 5 Block Diagram of PA[3:2]&PA0

IO_SEL: set pad attribute as input or output.

WRITE_EN: write data to pad.

READ_EN: read pad.

WUA: port A wake-up enable.

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.

SET_PAIF: port A wake-up flag.

EX_CKI1: external clock for Timer2, 3.

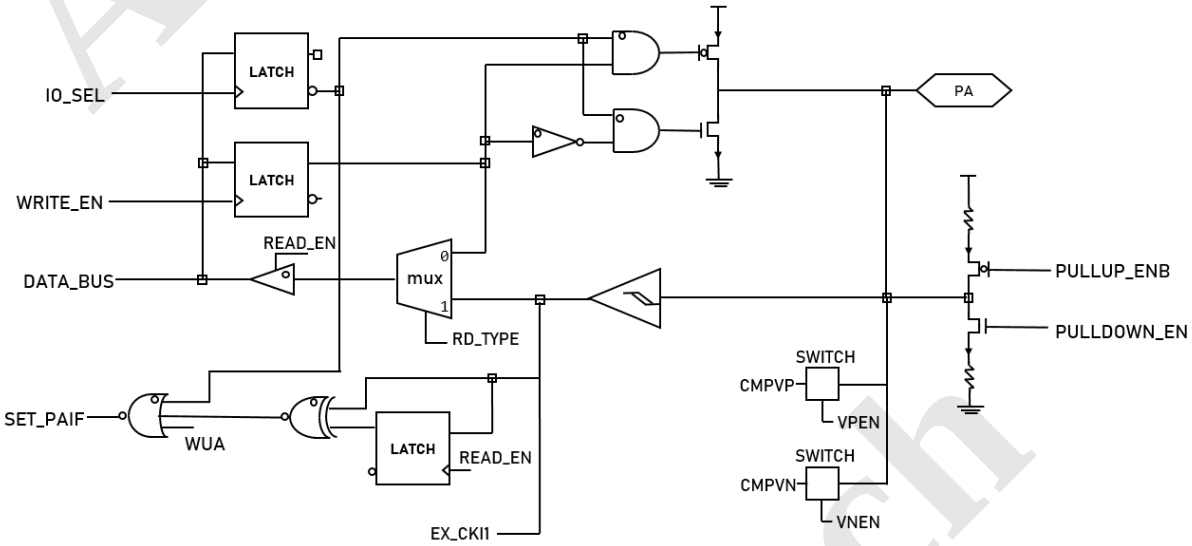


Figure 6 Block Diagram of PA1

IO_SEL: set pad attribute as input or output.

WRITE_EN: write data to pad.

READ_EN: read pad.

WUA: port A wake-up enable.

PULLUP_ENB: enable Pull-High.

RD_TYPE: select read pin or read latch.

SET_PAIF: port A wake-up flag.

PAEN: PWM2/CCP P1D function enable.

PADT: PWM2/CCP P1D data

EX_CKI0: external clock for Timer0, 1.

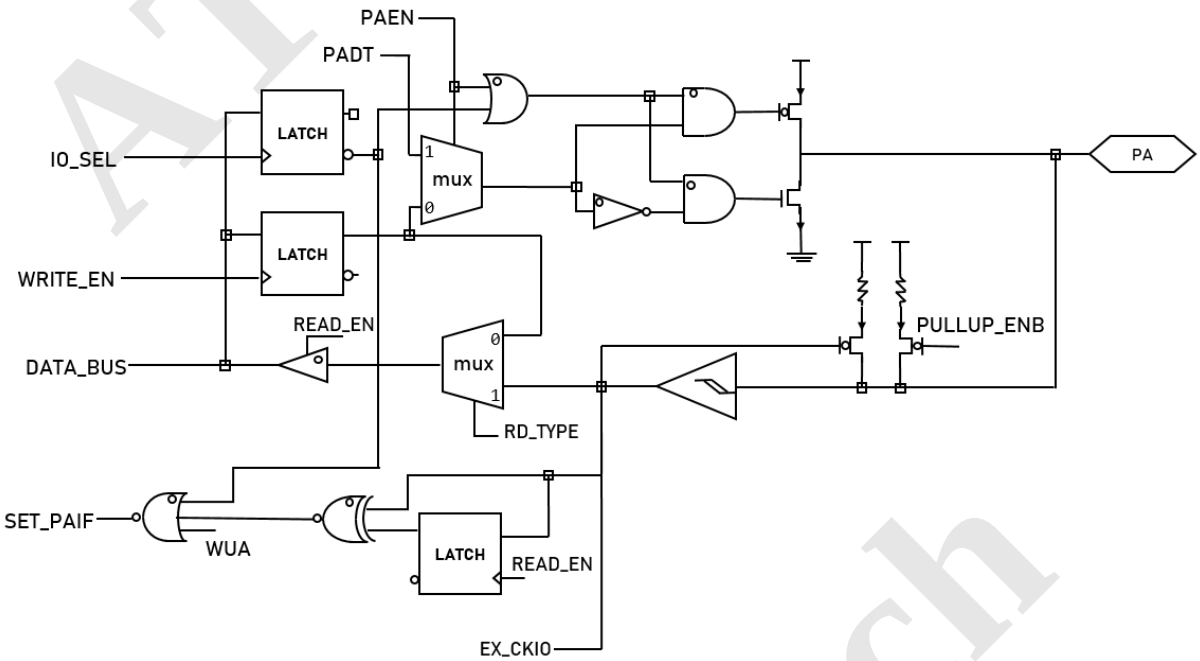


Figure 7 Block Diagram of PA4

RSTPAD_EN: enable PA5 as reset pin.

RSTB_IN: reset signal input.

IO_SEL: set pad attribute as input or output.

WRITE_EN: write data to pad.

READ_EN: read pad.

WUA: port A wake-up enable.

OD_EN: enable open-Drain.

PULLUP_ENB: enable Pull-High.

PULLDOWN_EN: enable Pull-Low.

RD_TYPE: select read pin or read latch.

SET_PAIF: port A wake-up flag.

PAEN: CCP P1B function enable.

PADT: CCP P1B data.

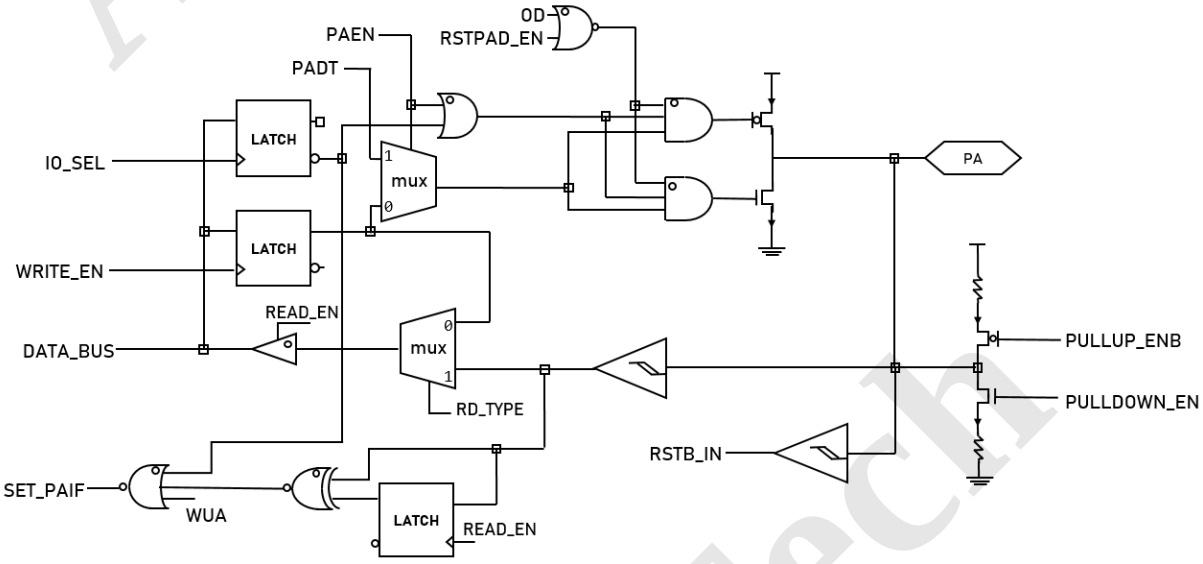


Figure 8 Block Diagram of PA5

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.

WUA: port A wake-up enable.

PULLUP_ENB: enable Pull-High.

RD_TYPE: select read pin or read latch.

SET_PAIF: port A wake-up flag

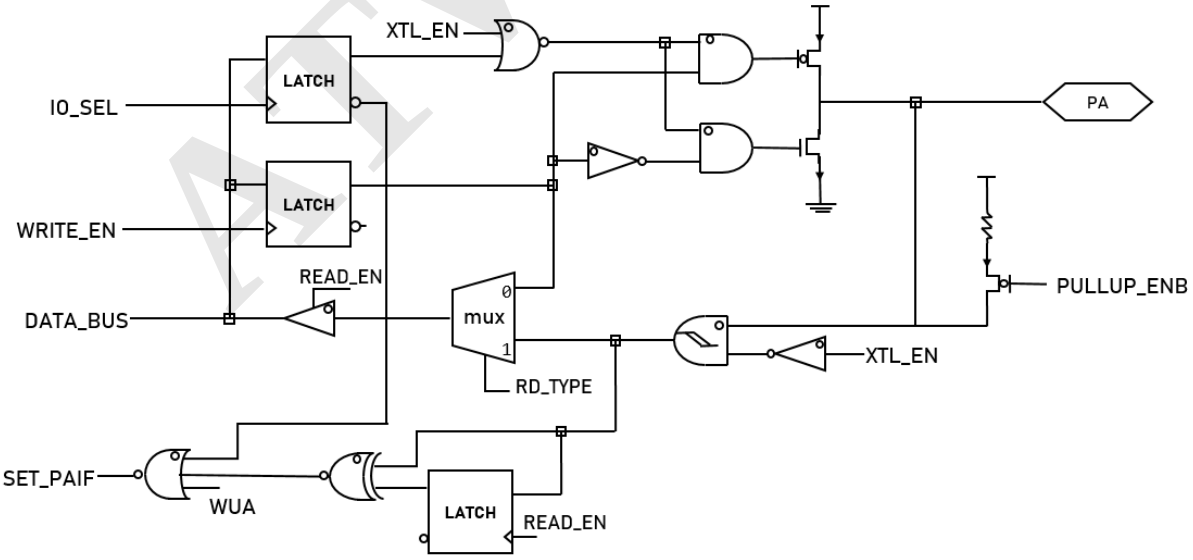


Figure 9 Block Diagram of PA6

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.

WUA: port A wake-up enable.

SET_PAIF: port A wake-up flag

PAEN: PWM4/CCP P1C function enable.

PADT: PWM4/CCP P1C data.

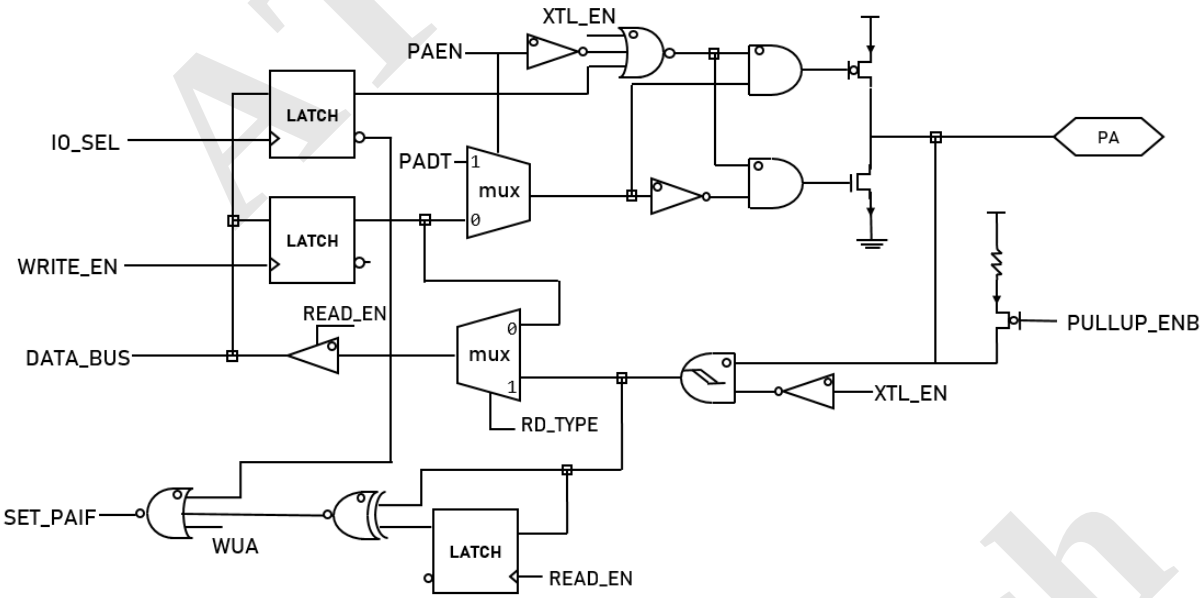


Figure 10 Block Diagram of PA7

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.

EIS0: external interrupt function enable.

EX_INT0: external interrupt signal.

WUB: port B wake-up enable.

SET_PBIF: port B wake-up flag.

PBEN: PWM5 function enable.

PBDT: PWM5 data

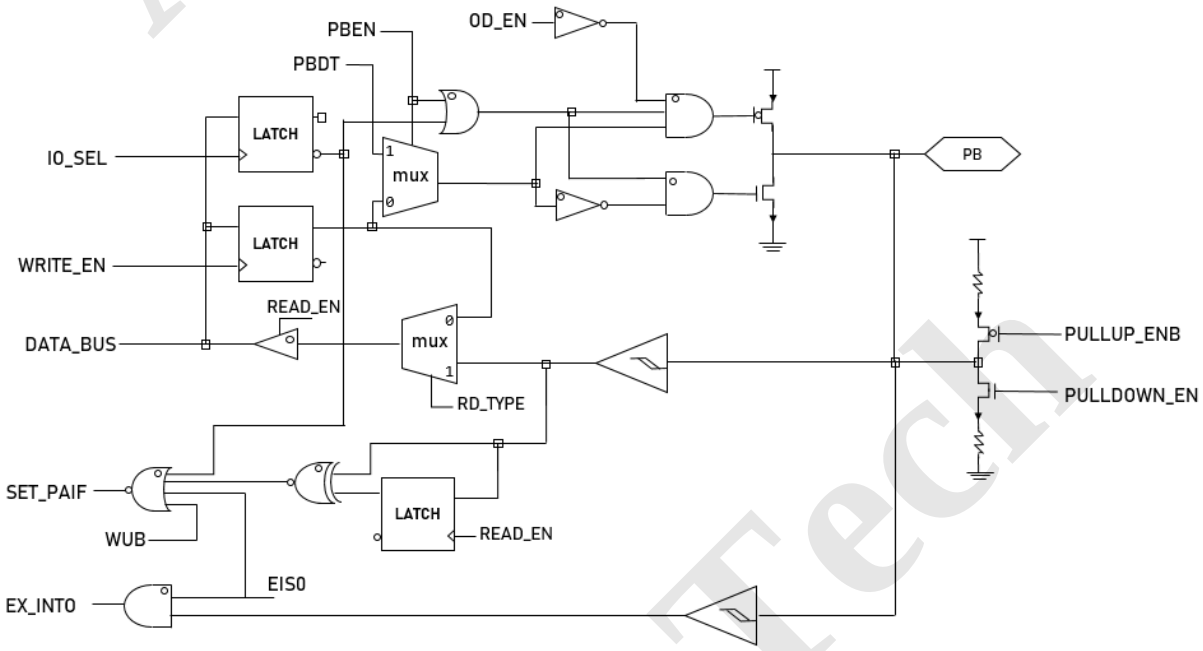


Figure 10 Block Diagram of PB0

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.

EIS1: external interrupt function enable.

EX_INT1: external interrupt signal.

WUB: port B wake-up enable.

SET_PBIF: port B wake-up flag.

PBEN: IR function enable.

PBDT: IR data.

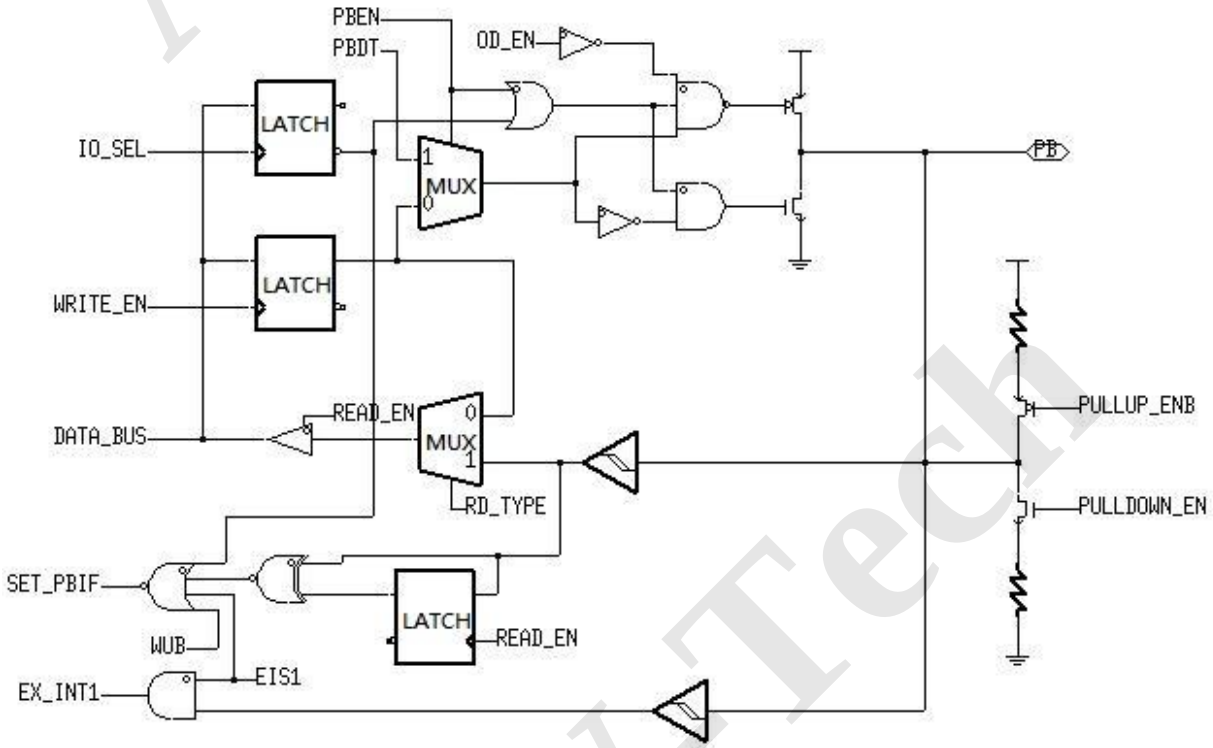


Figure 11 Block Diagram of PB1

IO_SEL: set pad attribute as input or output.

WRITE_EN: write data to pad.

READ_EN: read pad.

OD_EN: enable open-Drain.

PULLUP_ENB: enable Pull-High.

PULLDOWN_EN: enable Pull-Low.

RD_TYPE: select read pin or read latch.

WUB: port B wake-up enable.

SET_PBIF: port B wake-up flag.

PBEN:

PB2: PWM2/BZ2 function enable

PB3: PWM1/PWM5/SDO/BZ1/CMPO function enable.

PBDT:

PB2: PWM2/BZ2 data.

PB3: PWM1/PWM5/SDO/BZ1/CMPO data.

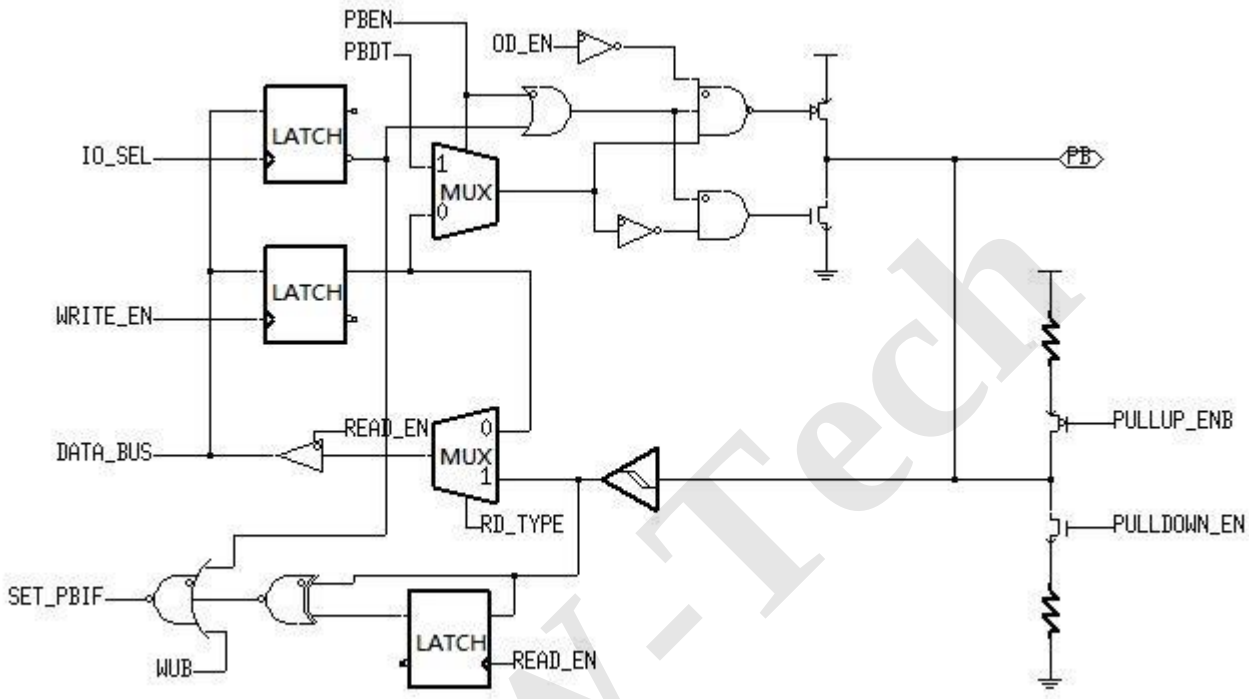


Figure 12 Block Diagram of PB2, PB3

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.

PULLDOWN_EN: enable Pull-Low.

RD_TYPE: select read pin or read latch.

WUB: port B wake-up enable.

SET_PBIF: port B wake-up flag.

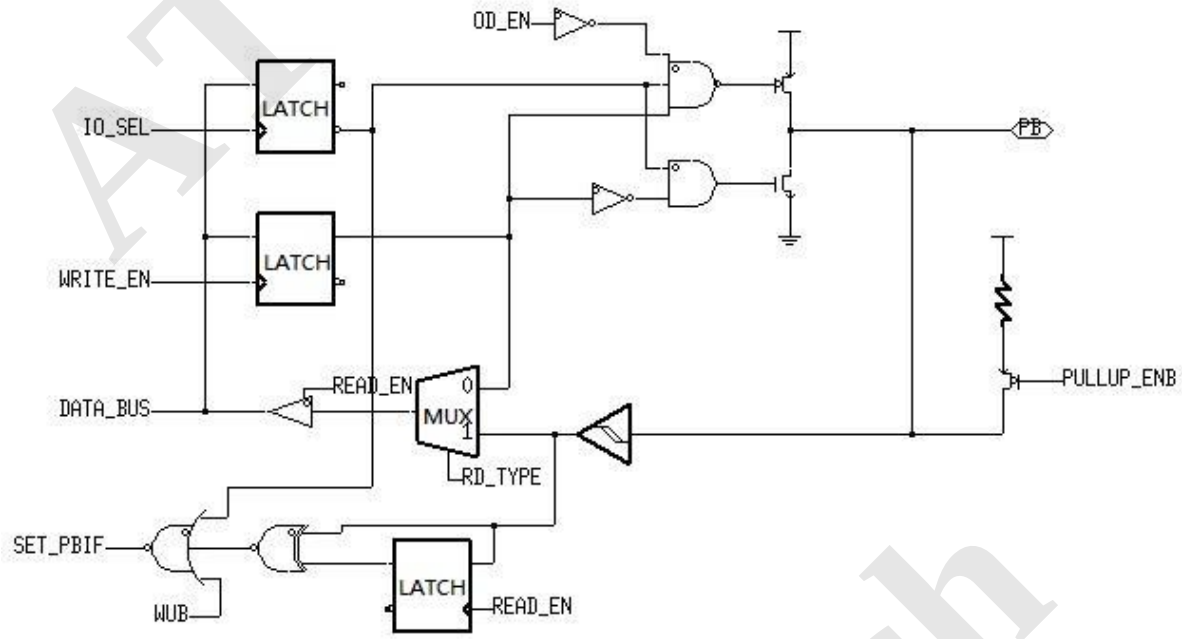


Figure 13 Block Diagram of PB3

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.

RD_TYPE: select read pin or read latch.

WUB: port B wake-up enable.

SET_PBIF: port B wake-up flag.

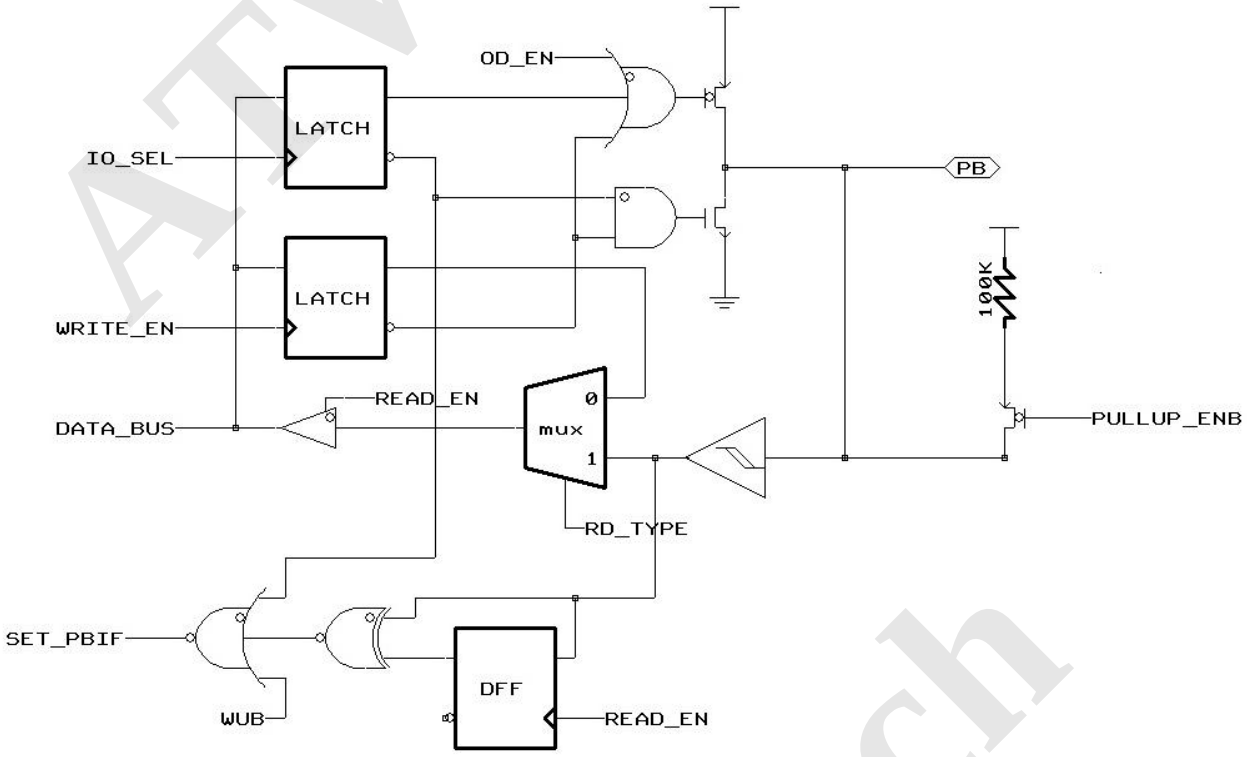


Figure 14 Block Diagram of PB4,5

3.6 Timer0

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

The clock source to Timer0 can be from instruction clock, external pin EX_CKI0 or low speed clock Low Oscillator Frequency according to register bit T0CS and LCK_TM0 (T0MD[5] and T0MD[7]). When T0CS is 0, instruction clock is selected as Timer0 clock source. When T0CS is 1 and LCK_TM0 is 0, EX_CKI0 is selected as Timer0 clock source. When T0CS is 1 and LCK_TM0 is 1 (and Timer0 source must set to 1), Low Oscillator Frequency (I_LRC or E_LXT, depends on configuration word) output is selected. Summarized table is shown below. (Also check Figure 15)

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

Table 21 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.

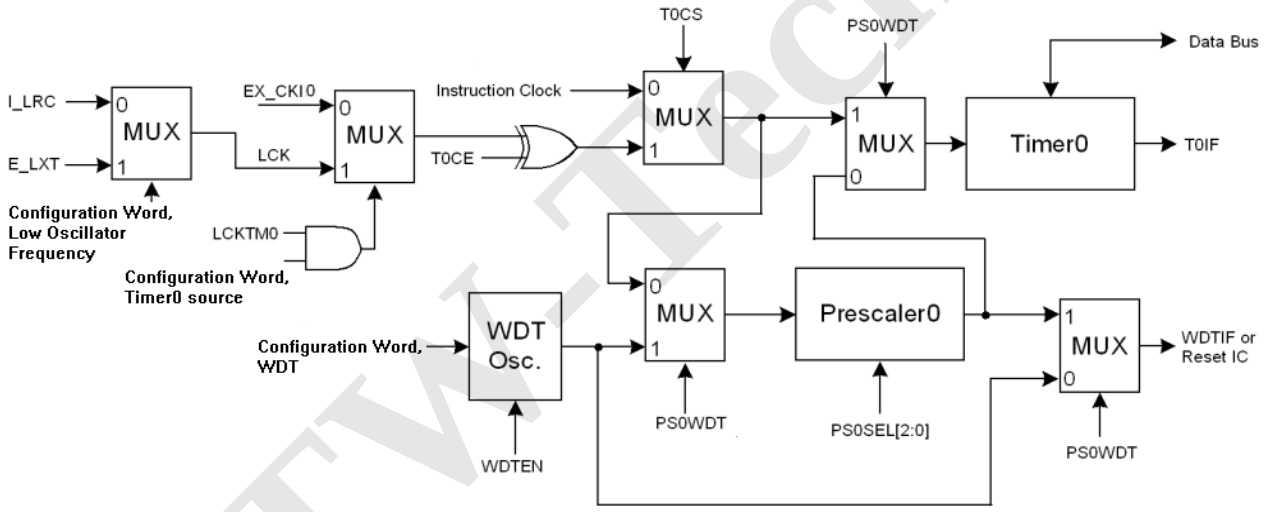


Figure 15 Block Diagram of Timer0 and WDT

3.7 Timer1 / PWM1 / Buzzer1

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

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

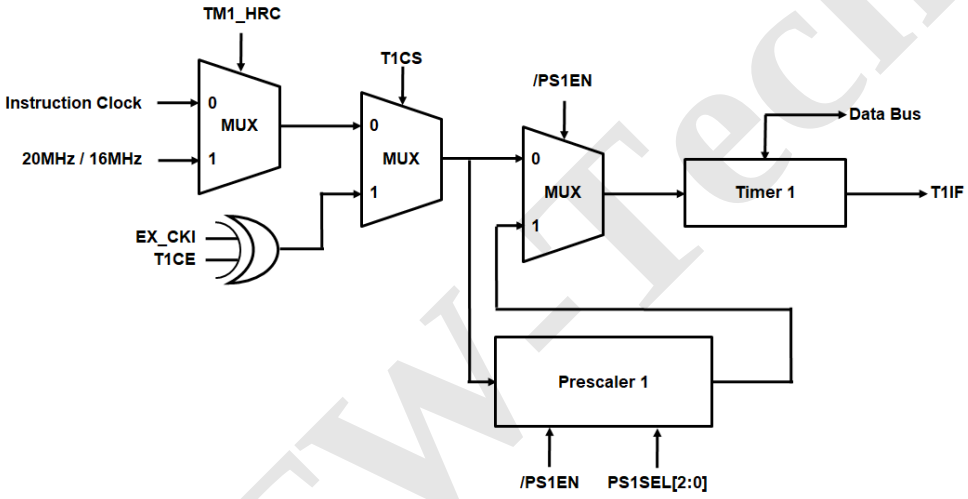


Figure16 Block Diagram of Timer1

The operation of Timer1 can be enabled or disabled by register bit T1EN (T1CR1[0]). After Timer1 is enabled, its clock source can be instruction clock or pin EX_CKI0 which is determined by register bit T1CS (T1CR2[5]).

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

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

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

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

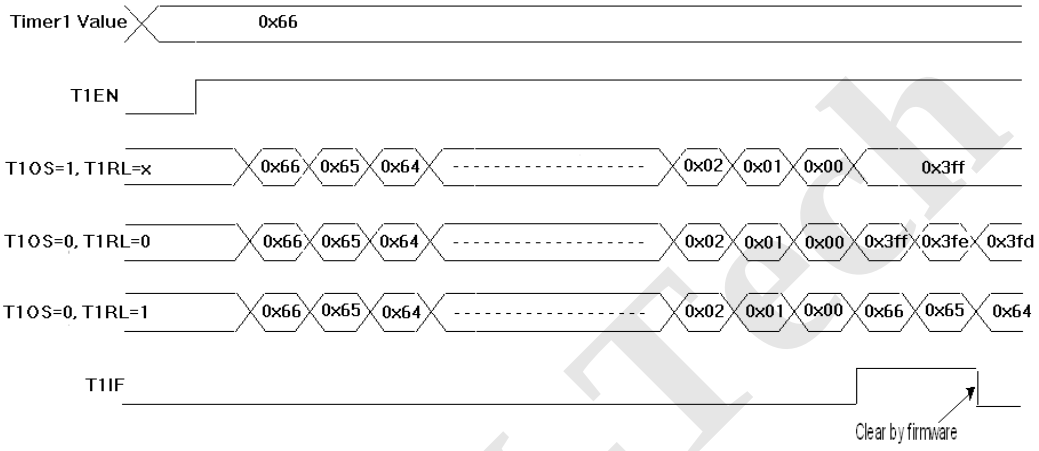


Figure 17 Timer1 Timing Chart

The PWM1 output can be available on I/O pin PB3 when register bit PWM1OEN (T1CR1[7]) is set to 1. Moreover, PB3 will become output pin automatically. The active state of PWM1 output is determined by register bit PWM1OAL (T1CR1[6]). When PWM1OAL is 1, PWM1 output is active low. When PWM1OAL is 0, PWM1 output is active high. Moreover, the duty cycle and frame rate of PWM1 are both programmable. The duty cycle is determined by registers TMRH[1:0] and PWM1DUTY[7:0]. When PWM1DUTY is 0, PWM1 output will be never active. When PWM1DUTY is 0x3FF, PWM1 output will be active for 1023 Timer1 input clocks. The frame rate is

determined by TMRH[5:4] + TMR1[7:0] initial value. Therefore, PWM1DUTY value must be less than or equal to TMRH[5:4] + TMR1[7:0]. When user write PWM1DUTY, write PWM1DUTY[9:8] MSB 2 bits(TMRH[1:0]) first and write PWM1DUTY[7:0] second, PWM1 duty register will be updated after Timer1 overflow occurs. The block diagram of PWM1 is illustrated in the following figure.

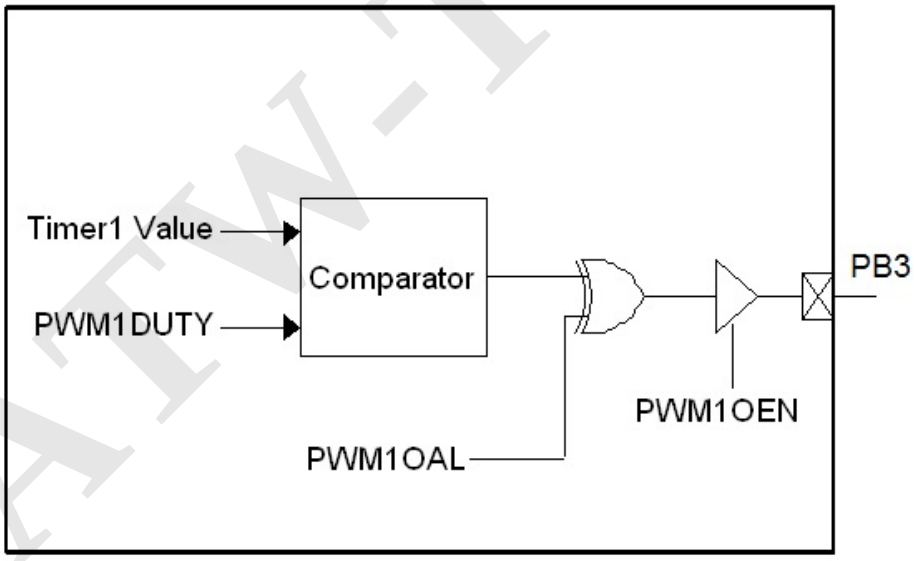


Figure 18 PWM1 Block Diagram

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

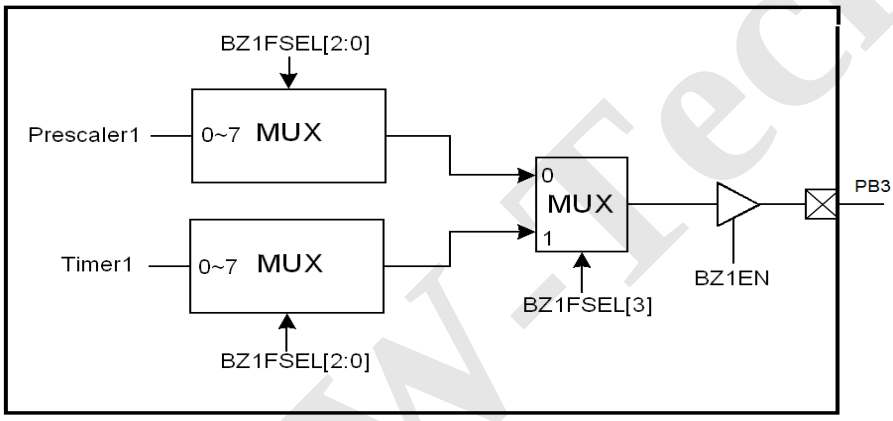


Figure 19 Buzzer1 Block Diagram

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

3.8 Timer2 / PWM2 / Buzzer2

Timer2 is a 10-bit down-count timer with Prescaler2 whose dividing rate is programmable. The output of Timer2 can be used to generate PWM2 output and Buzzer2 output. Timer2 builds in auto-reload function and Timer2 reload register stores reload data with double buffers. When user write Timer2 reload register, write Timer2 MSB 2 bits(TMRH[7:6]) first and write TMR2 second, Timer2 reload register will be updated to Timer2 counter after Timer2 overflow occurs when T2EN=1. If T2EN=0, Timer2 reload register will be updated to Timer2 counter after write TMR2 immediately. A read to the Timer2 will show the content of the Timer2 current count value.

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

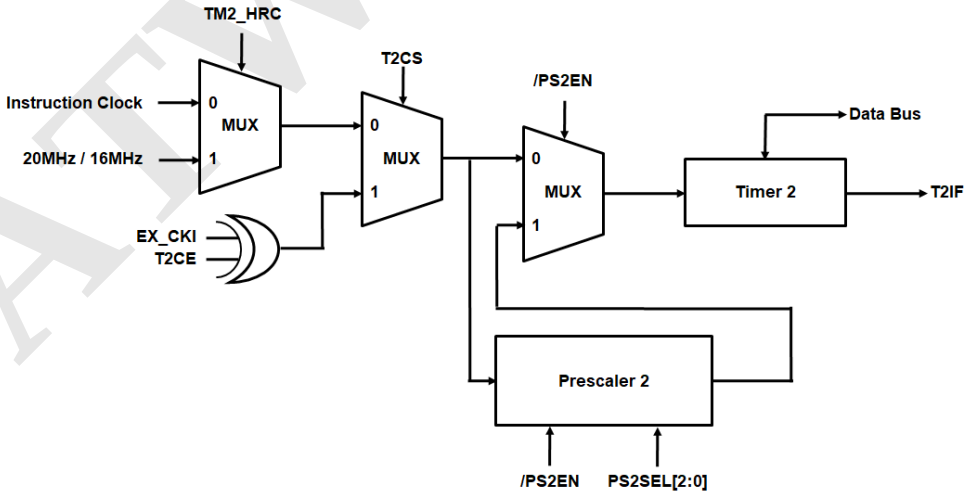


Figure 20 Block Diagram of Timer2

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

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

Timer2 provides two kinds of operating mode: one is One-Shot mode and the other is Non-Stop mode. When register bit T2OS (T2CR1[2]) is 1, One-Shot mode is selected. Timer2 will count down once from initial value stored on register TMR2[9:0] to 0x00, i.e. underflow is occurred. When register bit T2OS (T2CR1[2]) is 0, Non-Stop mode is selected. When underflow is occurred, there are two selections to start next down-count which is determined by register bit T2RL (T2CR1[1]). When T2RL is 1, the initial value stored on register TMR2[9:0] will be restored and start next down-count from this initial value. When T2RL is 0, Timer2 will start next down-count from 0x3FF.

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

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

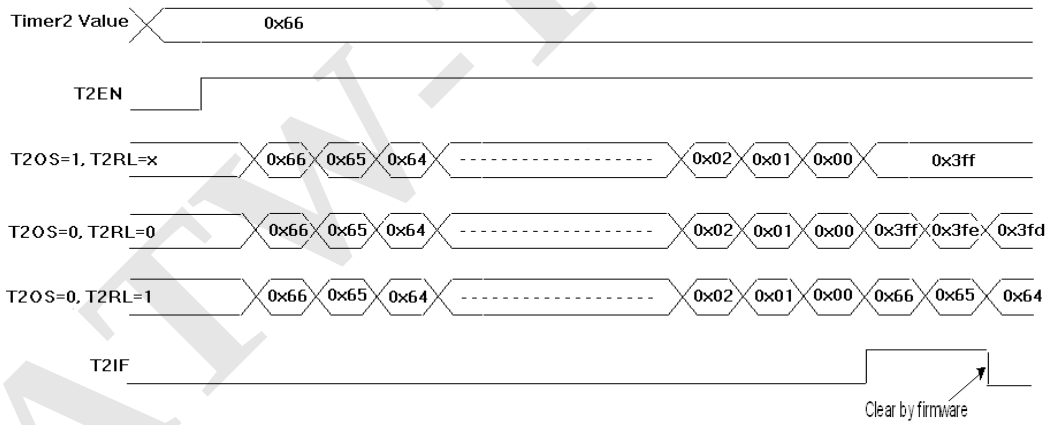


Figure 21 Timer2 Timing Chart

The PWM2 output can be available on I/O pin PA4 or PB2 when register bit PWM2OEN (T2CR1[7]) is set to 1. Moreover, PA4 or PB2 will become output pin automatically. The active state of PWM2 output is determined by register bit PWM2OAL (T2CR1[6]). When PWM2OAL is 1, PWM2 output is active low. When PWM2OAL is 0, PWM2 output is active high. Moreover, the duty cycle and frame rate of PWM2 are both programmable. The duty cycle is determined by register TMRH[3:2], PWM2DUTY[7:0]. When PWM2DUTY is 0, PWM2 output will be never active. When PWM2DUTY is 0x3FF, PWM2 output will be active for 1023 Timer2 input clocks. The frame rate is determined by TMRH[7:6], TMR2[7:0] initial value. Therefore, PWM2DUTY value must be less than or equal to TMR2[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 Timer2 overflow occurs. The block diagram of PWM2 is illustrated in the following figure.

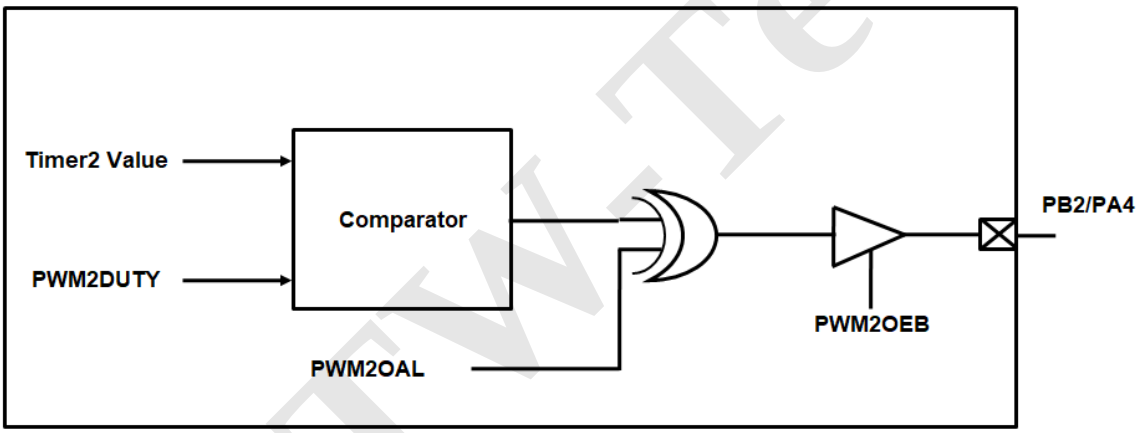


Figure 22 PWM2 Block Diagram

The Buzzer2 output (BZ2) can be available on I/O pin PB2 when register bit BZ2EN (BZ2CR1[7]) is set to 1. Moreover, PB2 will become output pin automatically. The frequency of BZ2 can be derived from Timer2 output or Prescaler2 output and dividing rate is determined by register bits BZ2FSEL[3:0] (BZ2CR[3:0]). When BZ2FSEL[3] is 0, Prescaler2 output is selected to generate BZ2 output. When BZ2FSEL[3] is 1, Timer2 output is selected to generate BZ2 output. The dividing rate can be from 1:2 to 1:256 in order to generate all kinds of frequency. The block diagram of Buzzer2 is illustrated in the following figure.

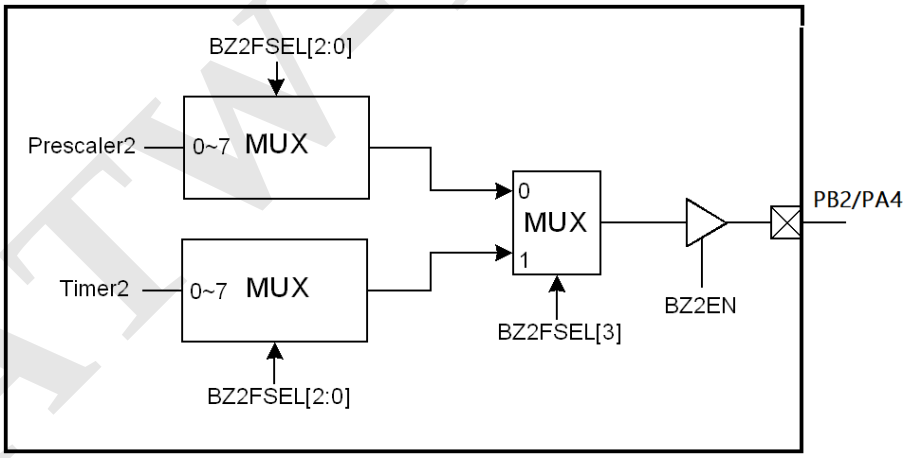


Figure 23 Buzzer2 Block Diagram

3.9 Timer3 / PWM3 / PWM4 / PWM5 / Buzzer3

Timer3 is a 10-bit down-count timer with Prescaler3 whose dividing rate is programmable. The output of Timer3 can be used to generate PWM3 output. Timer3 builds in auto-reload function and Timer3 reload register stores reload data with double buffers. When users write Timer3 reload register, write Timer3 MSB 2 bits(TM3RH[5:4]) first and write TMR3 second, Timer3 reload register will be updated to Timer3 counter after Timer3 overflow occurs when T3EN=1. If T3EN=0, Timer3 reload register will be updated to Timer3 counter after write TMR3 immediately. A read to the Timer3 will show the content of the Timer3 current count value.

The block diagram of Timer3 is shown in the figure below.

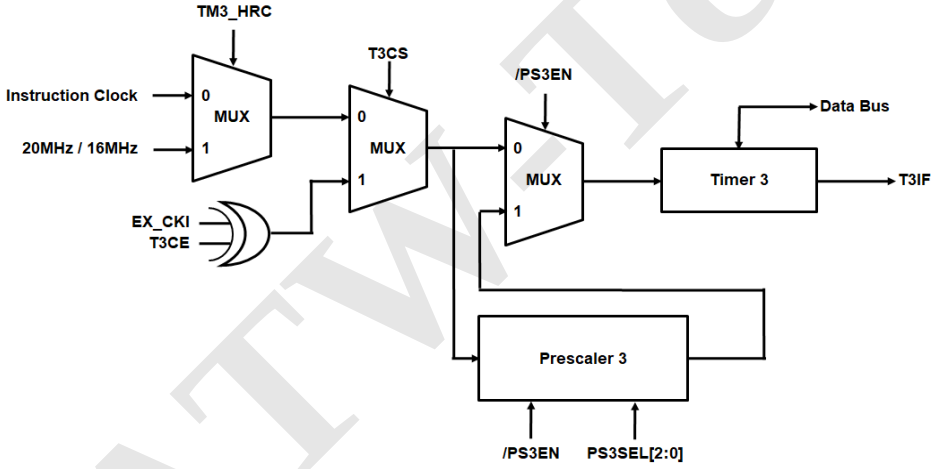


Figure 24 Block Diagram of Timer3

The operation of Timer3 can be enabled or disabled by register bit T3EN (T3CR1[0]). After Timer3 is enabled, its clock source can be instruction clock or pin EX_CKI1 which is determined by register bit T3CS (T3CR2[5]). When T3CS is 1, EX_CKI1 is selected as clock source. When T3CS is 0, instruction clock is selected as clock source. When EX_CKI1 is selected, the active edge to decrease Timer3 is determined by register bit T3CE (T3CR2[4]). When T3CE is 1, high-to-low transition on EX_CKI1 will decrease Timer3. When T3CE is 0, low-to-high transition on EX_CKI1 will decrease Timer3.

The selected clock source can be divided further by Prescaler3 before it is applied to Timer3. Prescaler3 is enabled by writing 0 to register bit /PS3EN (T3CR2[3]) and the dividing rate is from 1:2 to 1:256 determined by register bits PS3SEL[2:0] (T3CR2[2:0]). Current value of Prescaler3 can be obtained by reading register PS3CV.

Timer3 provides two kinds of operating mode: one is One-Shot mode and the other is Non-Stop mode. When register bit T3OS (T3CR1[2]) is 1, One-Shot mode is selected. Timer3 will count down once from initial value stored on register TMR3[9:0] to 0x00, i.e. underflow is occurred. When register bit T3OS (T3CR1[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 T3RL (T3CR1[1]). When T3RL is 1, the initial value stored on register TMR3[9:0] will be restored and start next down-count from this initial value. When T3RL is 0, Timer3 will start next down-count from 0x3FF.

When Timer3 is underflow, the register bit T3IF (INTE2[4]) will be set to 1 to indicate Timer3 underflow event is occurred. If register bit T3IE (INTE2[0]) and GIE are both set to 1, interrupt request will occur and interrupt service routine will be executed. T3IF will not be clear until firmware writes 0 to T3IF.

The timing chart of Timer3 is shown in the following figure.

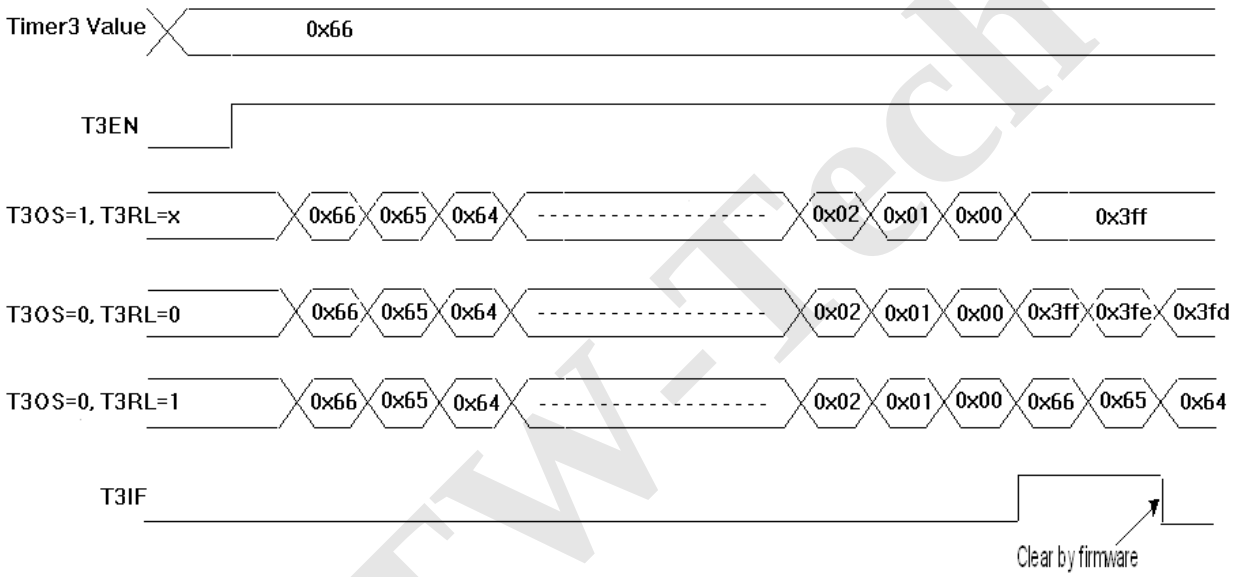


Figure 25 Timer3 Timing Chart

The PWM3 output can be available on I/O pin PA2 when register bit PWM3OEN (T3CR1[7]) is set to 1. Moreover, PA2 will become output pin automatically. The active state of PWM3 output is determined by register bit PWM3OAL (T3CR1[6]). When PWM3OAL is 1, PWM3 output is active low. When PWM3OAL is 0, PWM3 output is active high. Moreover, the duty cycle and frame rate of PWM3 are both programmable. The duty cycle is determined by register TM3RH[1:0], PWM3DUTY[7:0]. When PWM3DUTY is 0, PWM3 output will be never active. When PWM3DUTY is 0x3FF, PWM3 output will be active for 1023 Timer3 input clocks. The frame rate is determined by TM3RH[5:4], TMR3[7:0] initial value. Therefore, PWM3DUTY value must be less than or equal to TMR3[9:0]. When user write PWM3DUTY, write PWM3DUTY[9:8] MSB 2 bits(TM3RH[1:0]) first and write PWM3DUTY[7:0] second, PWM3 duty register will be updated after Timer3 overflow occurs. The block diagram of PWM3 is illustrated in the following figure.

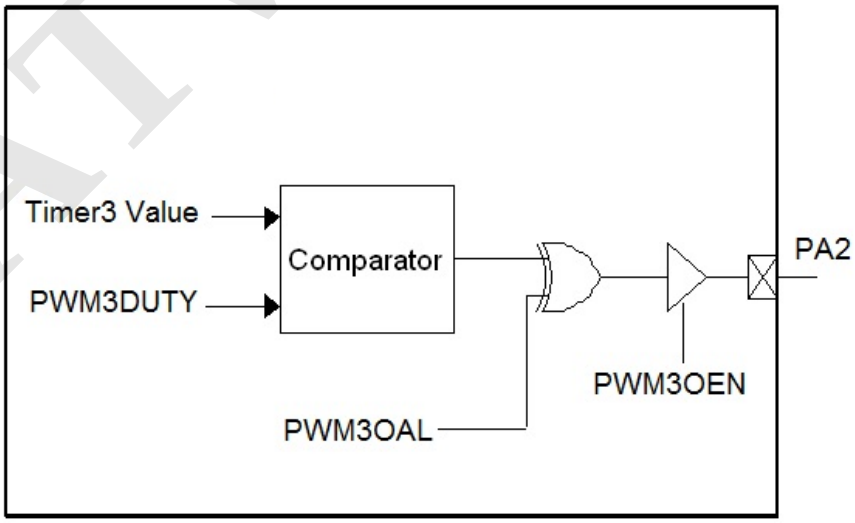


Figure 26 PWM3 Block Diagram

The Buzzer3 output (BZ3) can be available on I/O pin PA2 when register bit BZ3EN (BZ3CR1[7]) is set to 1. Moreover, PA2 will become output pin automatically. The frequency of BZ3 can be derived from Timer3 output or Prescaler3 output and dividing rate is determined by register bits BZ3FSEL[3:0] (BZ3CR[3:0]). When BZ3FSEL[3] is 0, Prescaler3 output is selected to generate BZ3 output. When BZ3FSEL[3] is 1, Timer3 output is selected to generate BZ3 output. The dividing rate can be from 1:2 to 1:256 in order to generate all kinds of frequency. The block diagram of Buzzer3 is illustrated in the following figure.

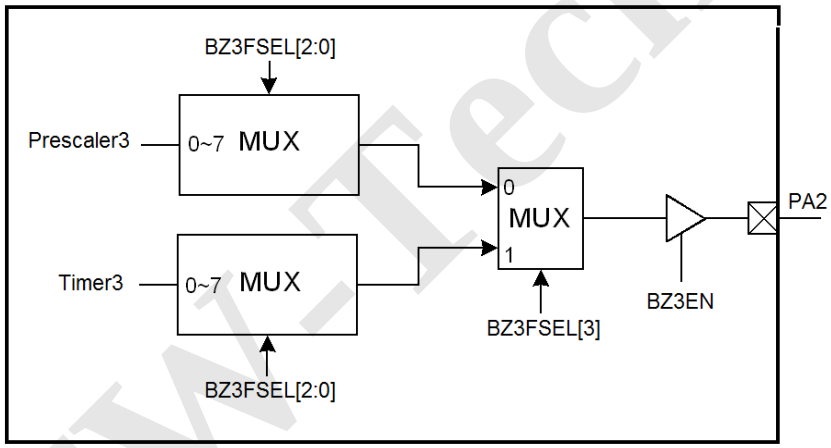


Figure 27 Buzzer3 Block Diagram

The PWM4 output can be available on I/O pin PA3 or PA7 when register bit PWM4OEN (P4CR1[7]) is set to 1. Moreover, PA3 or PA7 will become output pin automatically. The active state of PWM4 output is determined by register bit PWM4OAL (P4CR1[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 register TM3RH[3:2], PWM4DUTY[7:0]. When PWM4DUTY is 0, PWM4 output will be never active. When PWM4DUTY is 0x3FF, PWM4 output will be active for 1023 Timer3 input clocks. The frame rate is determined by TM3RH[5:4], TMR3[7:0] initial value. Therefore, PWM4DUTY value must be less than or equal to TMR3[9:0]. When user write PWM4DUTY, write PWM4DUTY[9:8] MSB 2 bits(TM3RH[3:2]) first and write PWM4DUTY[7:0] second, PWM4 duty register will be updated after Timer3 overflow occurs. The block diagram of PWM4 is illustrated in the following figure.

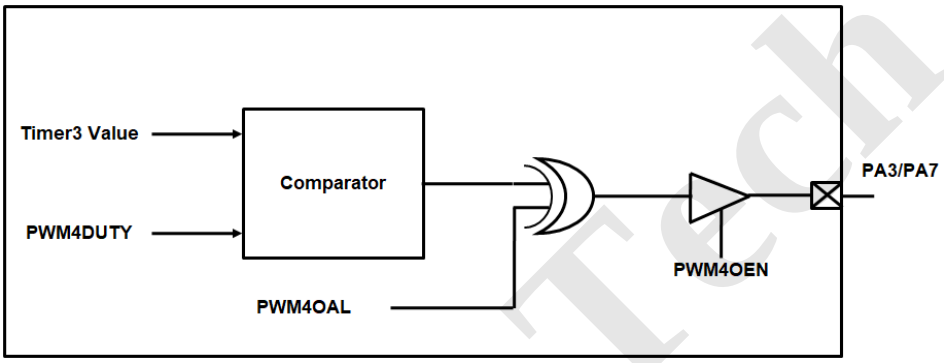


Figure 28 PWM4 Block Diagram

The PWM5 output can be available on I/O pin PB0 or PB3 when register bit PWM5OEN (P5CR1[7]) is set to 1. Moreover, PB0 or PB3 will become output pin automatically. The active state of PWM5 output is determined by register bit PWM5OAL (P5CR1[6]). When PWM5OAL is 1, PWM5 output is active low. When PWM5OAL is 0, PWM5 output is active high. Moreover, the duty cycle and frame rate of PWM5 are both programmable. The duty cycle is determined by register TM3RH[3:2], PWM5DUTY[7:0]. When PWM5DUTY is 0, PWM5 output will be never active. When PWM5DUTY is 0x3FF, PWM5 output will be active for 1023 Timer3 input clocks. The frame rate is determined by TM3RH[5:4], TMR3[7:0] initial value. Therefore, PWM5DUTY value must be less than or

equal to TMR3[9:0]. When user write PWM5DUTY, write PWM5DUTY[9:8] MSB 2 bits(PWM5RH[1:0]) first and write PWM5DUTY[7:0] second, PWM5 duty register will be updated after Timer3 overflow occurs. The block diagram of PWM5 is illustrated in the following figure.

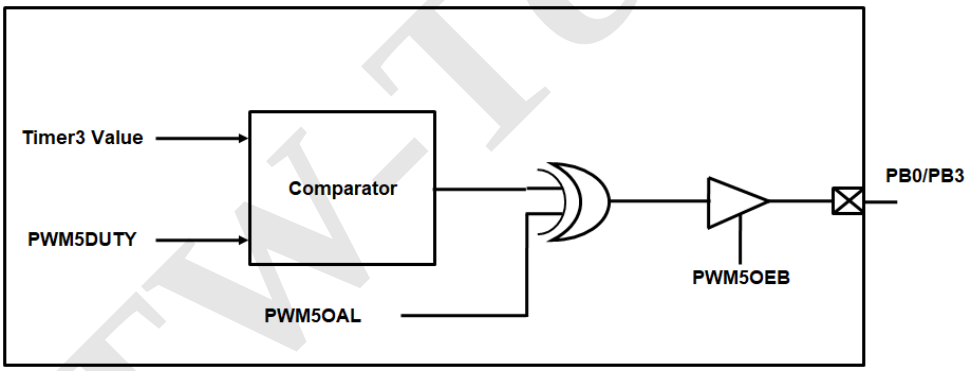


Figure 29 PWM5 Block Diagram

3.10 CCP Mode

The CCP (Capture/Compare/PWM) register (CCPR) is comprised of two 8-bit registers: CCPRL (low byte) and CCPRH (high byte). The CCPCON and PWMDB registers control the operation of CCP. The capture, compare mode use16-bit timer and PWM mode use10-bit timer respectively. The CCP timers and registers utilize the existing timers and registers. The following table shows the registers and timers resources shared with CCP modules. Note that when AT8B62F1 is in CCP mode, related timer/PWM function are disabled.

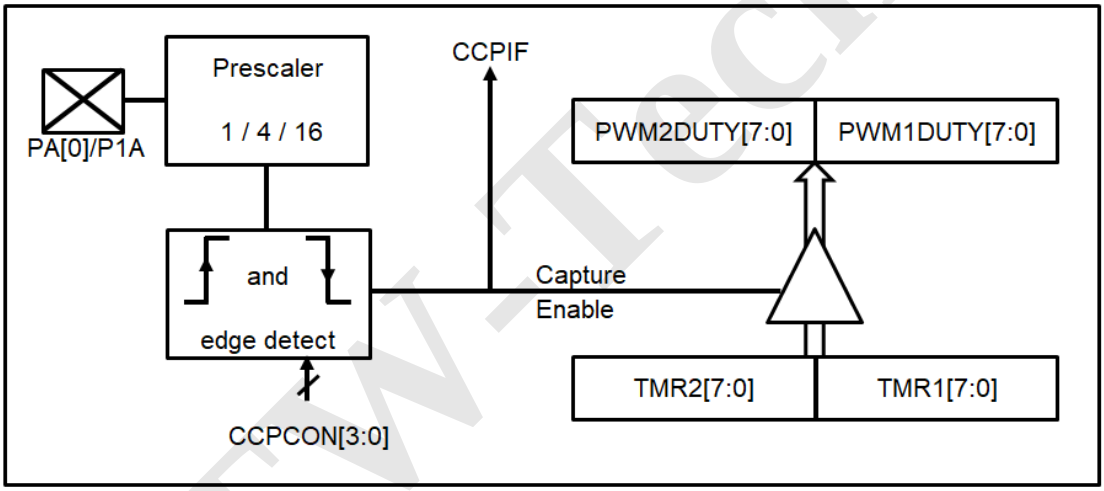
CCP mode	CCP register/timer	Shared Timers	Shared Registers
Capture/Compare	CCPL	-	PWM1DUTY[7:0]
Capture/Compare	CCPH	-	PWM2DUTY[7:0]
PWM	PWMDUTY	-	PWM2DUTY[9:0]
PWM	PWMDB	-	-
Capture/Compare	Capture/Compare timer	Timer2[7:0] (MSB) + Timer1[7:0] (LSB)	-
PWM	PWM timer	Timer2[9:0]	-

3.10.1 Capture Mode

In capture mode, CCPRH:CCPRL (That is, PWM2DUTY[7:0]:PWM1DUTY[7:0]) captures the 16-bit value of capture timer register when an event occurs on pin P1A (PA0). An event is defined as:

- every falling edge
- every rising edge
- every 4th rising edge
- every 16th rising edge

The block diagram of capture mode is as the following:



An event is selected by control bits CCPM[3:0]. When a capture is made, the interrupt request flag bit CCPIF is set.

In capture mode, the P1A(PA0) should be configured ad an input. If it is an output pin, a write to the port can cause a capture condition.

In capture mode, capture timer must be running in CPU clock synchronous mode, or the capture operation may not work.

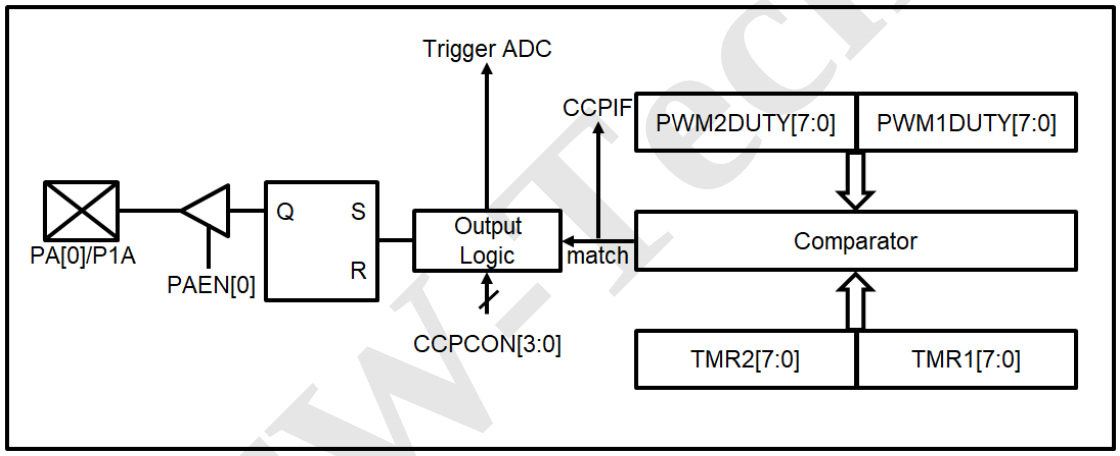
The capture event counter (or prescaler) is cleared when not in capture mode. Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared.

3.10.2 Compare Mode

In compare mode, the 16-bit CCPR (That is, PWM2DUTY[7:0]:PWM1DUTY[7:0]) register value is constantly compared against the compare timer value. When a match occurs, the CCP1 (PA0) pin is :

- driven high
- driven low
- toggle output
- remain unchanged (only interrupt)
- Trigger ADC if ADC is enabled

The compare mode block diagram is as the below shown:



The action on the pin is based on the value of control bit CCPM[3:0]. When compare match occurs, interrupt flag bit CCPIF is set.

In compare mode, the user must configure the CCP (PA0) pin as an output.

In compare mode, compare timer must be running in CPU clock synchronous mode, or the compare operation may not work.

The CCP module will not automatically configures the pin as an output when the module is enabled.

To be used for the Compare features, the selected timers must be running in Timer mode or Synchronized Counter mode. In Asynchronous Counter mode, the Compare operation may not work. Note the CCP timer is running in a down-count mode.

3.10.3 CCP PWM Mode

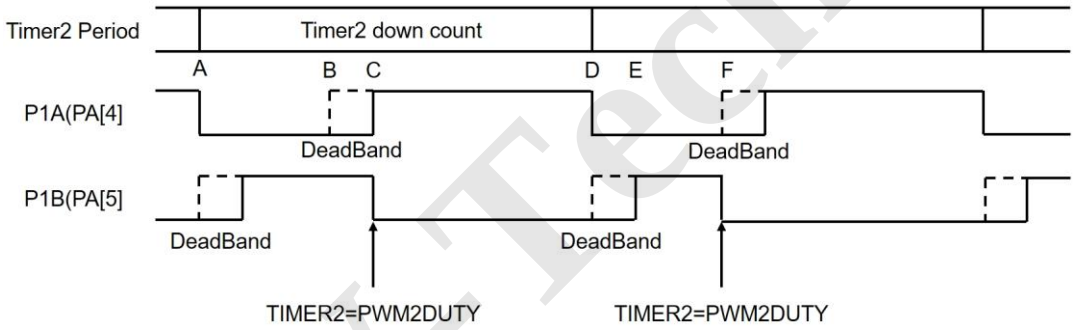
In CCP PWM mode, the CCP module produces up to a 10-bit resolution PWM output. The PWM pads are P1A(PA0), P1B(PA5), P1C(PA7) and P1D(PA4). The PWM period and duty are specified by Timer2[9:0] and PWM2DUTY[9:0] registers.

Register PWM2M[1:0] in the CCPCON register allows one of the following configurations:

- Single output: P1A output. P1B, P1C, P1D assigned as normal I/O.
- Half-Bridge output: P1A, P1B modulated with deadband control. P1C, P1D assign as normal I/O.
- Full-Bridge output, Forward mode: P1D modulated, P1A active. P1B, P1C inactive.
- Full-Bridge output, Reverse mode: P1B modulated, P1C active. P1A, P1D inactive.

In single output mode, P1A (PA0) pin is used as PWM output. PA0 must be set as output.

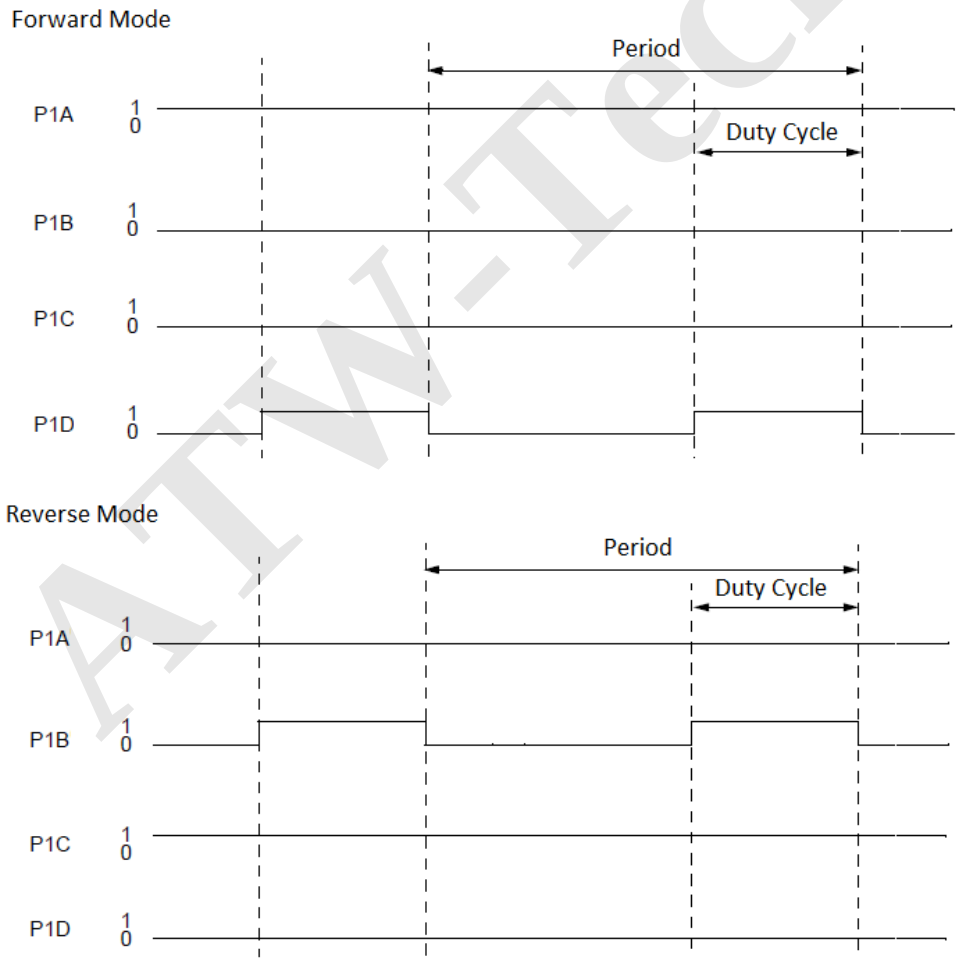
In Half-Bridge mode, P1A (PA0) pin has the PWM output signal, P1B (PA5) pin has the complementary PWM output signal. In this mode, PA0 and PA5 must be set as outputs.



In the Half-Bridge output mode, a digitally programmable deadband delay is available to avoid shoot-through current from destroying the bridge power switch. PWMDB[7:0] is deadband delay count for Half-Bridge mode, the delay unit is in CPU cycle.

As the above timing shown, timer2 is a 10-bit down counter, P1A is PWM2 output, P1B is PWM2 complementary output. AT point B, Timer2 value is equal to PWM2DUTY. At point D, Timer2 count to zero and reload value. If there no deadband control, P1A output should be active from point B to Point D. P1B output should be active from point D to point F. A non-zero deadband will delay P1A rising point from B to C and delay P1B rising point from D to E. The deadband zone are the time from B to C and from D to E.

In Full-Bridge mode, four pins are used as outputs; however, only two outputs are active at a time. In the Forward mode, P1A(PA0) is continuously activated and P1D(PA4) is modulated. In the Reverse mode, P1C(PA7) is continuously activated and P1B(PA5) is modulated. In this mode, PA0, PA5, PA7 and PA4 must be set as outputs. The following timing diagram shows Full-Bridge Forward and Reverse condition.



3.11 RFC Mode

AT8B62F1 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 14 AT8B62F1 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.)

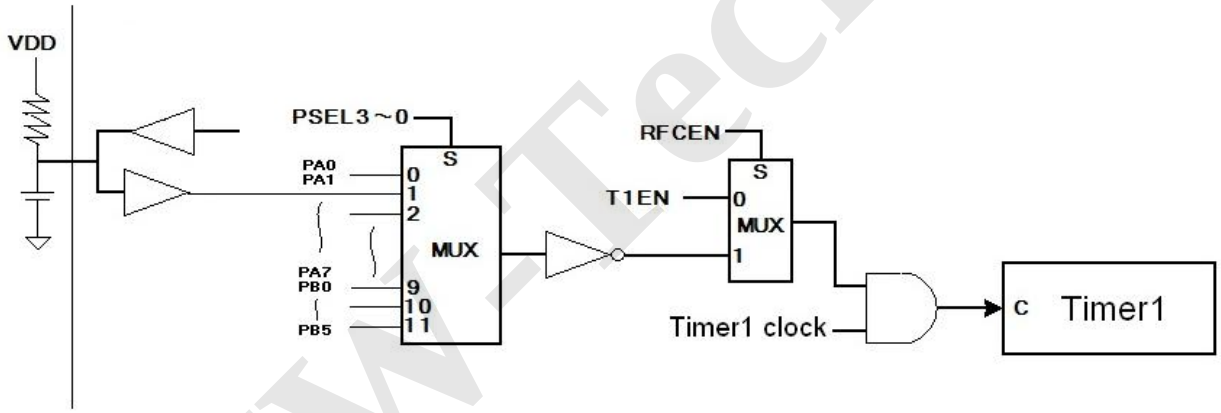


Figure 28 RFC Block Diagram

3.12 IR Carrier

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

The IR carrier frequency is selectable by register bit IRF57K (IRCR[1]). When IRF57K is 1, IR carrier frequency is 57KHz. When IRF57K is 0, IR carrier frequency is 38KHz. Because IR carrier frequency is derived from high frequency system oscillation F_{HOSC}, it is necessary to specify what frequency is used as system oscillation when external crystal is used. Register bit IROSC358M (IRCR[7]) is used to provide AT8B62F1 this information. When IROSC358M is 1, frequency of external crystal is 3.58MHz and when IROSC358M is 0, frequency of external crystal is 455KHz. When internal high frequency oscillation is adopted, this register will be ignored, and it will provide 4MHz clock to IR module.

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

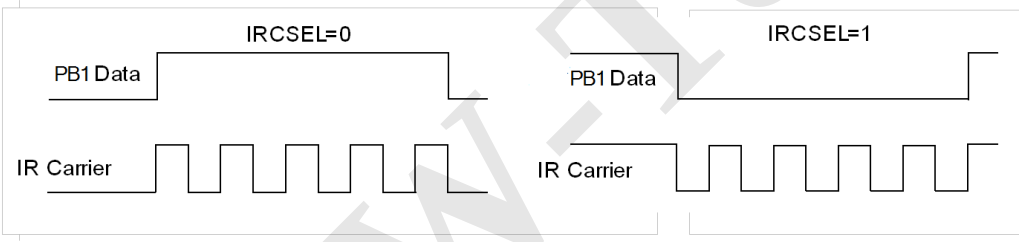


Figure 29 Polarity of IR Carrier vs. Output Data

3.13 Low Voltage Detector (LVD)

AT8B62F1 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[2:0] 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:

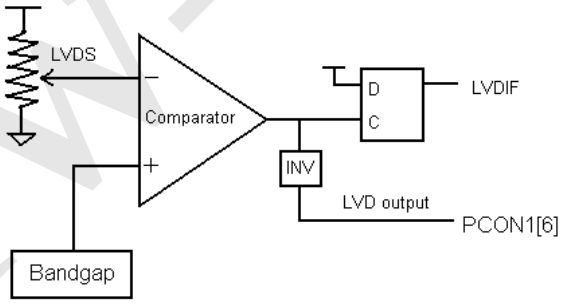


Figure 30 LVD block diagram

The following table is LVD voltage select table.

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

Table 22 LVD voltage select

3.14 Voltage Comparator

AT8B62F1 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.

AT8B62F1 comparator has two operating modes, which is P2V mode and P2P mode. These two modes are determined by VS[3:0] (register CMPCR[3:0]). When VS[3:0]=0, the comparator is in P2P mode. When VS[3:0]=1~15, it is in P2V mode. The pads used in the comparator are set as analog pads in the configuration words “Comparator Input” Pin Select.

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

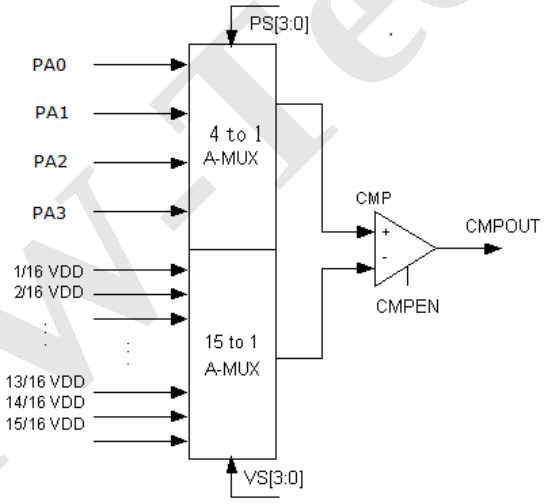


Figure 31 Comparator P2V mode block diagram

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

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

Table23 P2V mode reference voltage select

In P2V mode, the non-inverting input of the comparator is determined by PS[3:0] (register CMPCR[7:4]). PS[3:0] select one out of 4 pads PA0~3 as the non-inverting input of the comparator. The table is shown below.

PS[3:0]	Selected Pad
0000	PA0
0001	PA1
0010	PA2
0011	PA3
0100 ~1111	-

Table24 P2V mode pad select

The P2P mode has the function of comparing voltage between two analog pads. In this mode VS[3:0]=0, PS[3:0] select 2 out of 4 analog pads to be the non-inverting and inverting input of the comparator. The selection table is as the below.

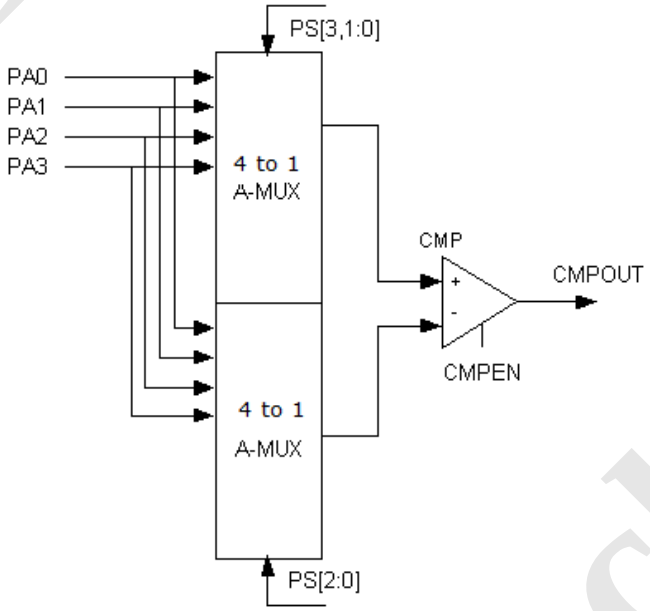


Figure32 Comparator P2P mode block diagram

PS[3:0]	Non-inverting input	Inverting input
0000	PA0	PA1
0001	PA1	PA0
0010	PA2	PA3
0011	PA3	PA2
0100~1111	-	-

Table 25 P2P mode pads select

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

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

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

To probe comparator output at output pad, set CMPOE (register OSCCR[6]) to 1, then PB3 will be the real-time state of the comparator output. It is noted that when CMPOE=1, the PWM1 function will be disabled if it is enabled.

3.15 Analog-to-Digital Convertor (ADC)

AT8B62F1 provide 11+1 channel 12-bit SAR ADC to transfer analog signal into 12-bits digital data. The ADC high reference voltage is selectable. They can be 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, PB0~PB5 or from internal generated 1 / 4 *VDD. The ADC clock ADCLK can be selected to be F_{INST}/1, F_{INST}/2, F_{INST}/8 or F_{INST}/16. The Sampling pulse width can be selected to be ADCLK*1, ADCLK*2, ADCLK*4 or ADCLK*8. Set ADEN=1 before ADC take into operation. Then set START=1, the ADC will start to convert analog signal to digital. EOC=0 means ADC is in processing. EOC=1 indicate ADC is at end of conversion. If ADIE=1 and global interrupt is enabled, the ADC interrupt will issue after EOC low go high. The block diagram is as following figure.

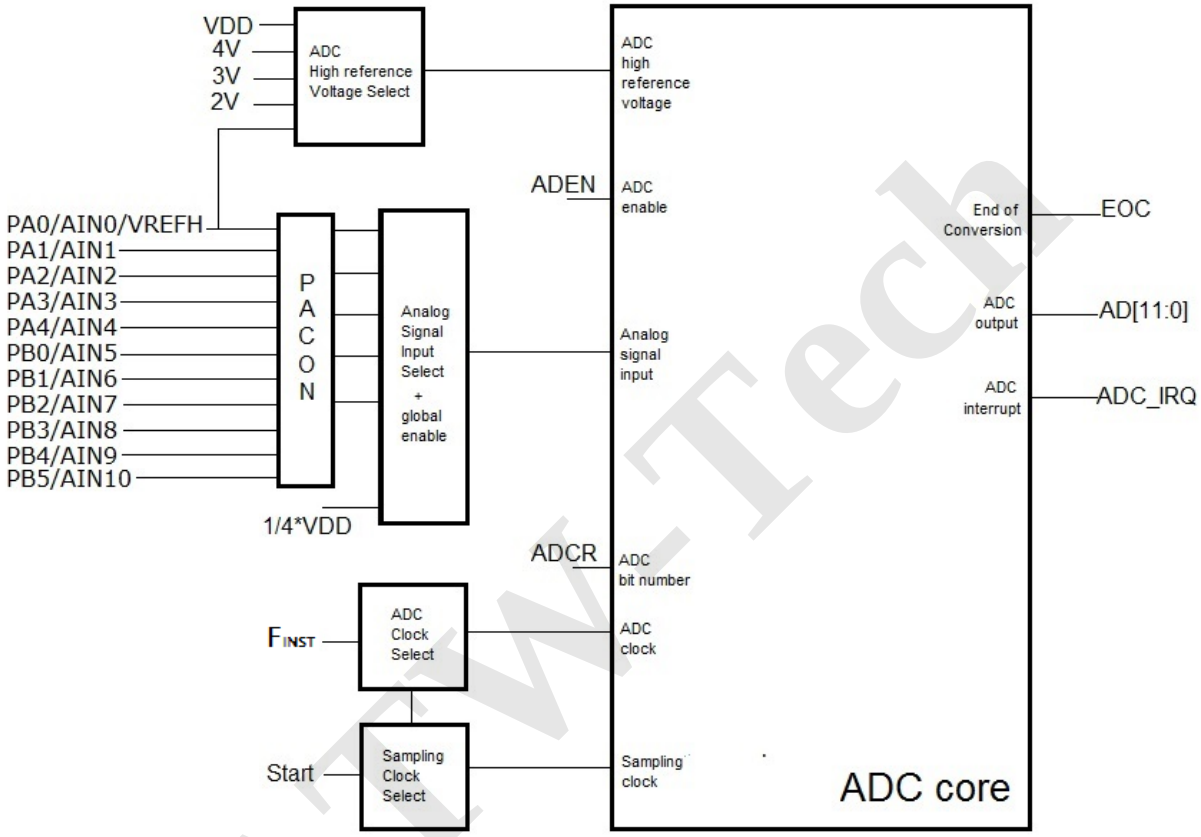


Figure33 ADC block diagram

3.15.1 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 26 ADC reference voltage select

3.15.2 ADC Analog Input Channel

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

GCHS	CHS[3:0]	ADC analog input source
0	xxxx	x
1	0000	PA0
1	0001	PA1
1	0010	PA2
1	0011	PA3
1	0100	PA4
1	0101	PB0
1	0110	PB1
1	0111	PB2

GCHS	CHS[3:0]	ADC analog input source
1	1000	PB3
1	1001	PB4
1	1010	PB5
1	1011	1 / 4 * VDD
1	11xx	N.C.

Table 27 ADC analog input source select

ADC input pins are shared with digital I/O pins. Connect an analog signal to these pin may cause extra current leakage in I/O pins. In the power down mode, the above leakage current will be a big problem. PACON[0:4] are PA[0:4] configuration register, PACON[5:7] are PB[0:2] configuration register and ADCR[4:6] are PB[3:5] configuration register to solve above problem. Write “1” to PACON and ADCR[4:6] will configure related PA /PB pin as pure analog input pin to avoid current leakage, and it can't be use as normal I/O.

Except setting the PACON and ADCR[4:6] register, the selected analog input pin must be set as input mode and the internal pull-high / pull-low must be disabled, otherwise the analog input level may be affected.

3.15.3 ADCClock (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 F_{INST} and is selectable from ADCK[1:0].

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

Table 28 ADC clock select

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

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

Table 29 ADC sampling clock select

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

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

Table 30 conversion bit number select

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

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

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

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

Table 31 ADC Conversion time

3.15.4 ADC Operation

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

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

3.16 Watch-Dog Timer (WDT)

There is an on-chip free-running oscillator in AT8B62F1 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 AT8B62F1 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 3.5ms, 15ms, 60ms or 250ms 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 AT8B62F1 and dividing rate is from 1:2 to 1:256 if WDT time-out will interrupt AT8B62F1.

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.

3.17 Interrupt

AT8B62F1 provides two kinds of interrupt: one is software interrupt and the other is hardware interrupt. Software interrupt is caused by execution of instruction INT. There are 11 hardware interrupts:

- Timer0 overflow interrupt.
- Timer1 underflow interrupt.
- Timer2 underflow interrupt or CCP interrupt.
- Timer3 underflow interrupt.
- WDT timeout interrupt.
- PA/PB input change interrupt.
- External 0 interrupt.
- External 1 interrupt
- LVD interrupt.
- Comparator output status change interrupt.
- ADC end-of-convert interrupt.

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

After instruction INT is executed, no matter GIE is set or clear, the next instruction will be fetched from address 0x001. At the same time, GIE will be clear to 0 by AT8B62F1 automatically. 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 AT8B62F1 automatically. 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.

3.17.1 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.

3.17.2 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.

3.17.3 Timer2 Underflow Interrupt or CCP interrupt

Timer2 underflow (from 0x3FF to 0x00) or CCP will set register bit T2IF/CCPIF. This interrupt request will be serviced if T2IE/CCPIE and GIE are set to 1.

3.17.4 Timer3 Underflow Interrupt

Timer3 underflow (from 0x3FF to 0x00) will set register bit T3IF. This interrupt request will be serviced if T3IE and GIE are set to 1.

3.17.5 WDT Timeout Interrupt

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

3.17.6 PA/PB Input Change Interrupt

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

3.17.7 External 0 Interrupt

According to the configuration of EIS0=1 and INTEDG, the selected active edge on I/O pin PB0 will set register bit INT0IF and this interrupt request will be served if INT0IE and GIE are set to 1.

3.17.8 External 1 Interrupt

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

3.17.9 LVD Interrupt

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

3.17.10 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.

3.17.11 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 Oscillation Configuration

Because AT8B62F1 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).

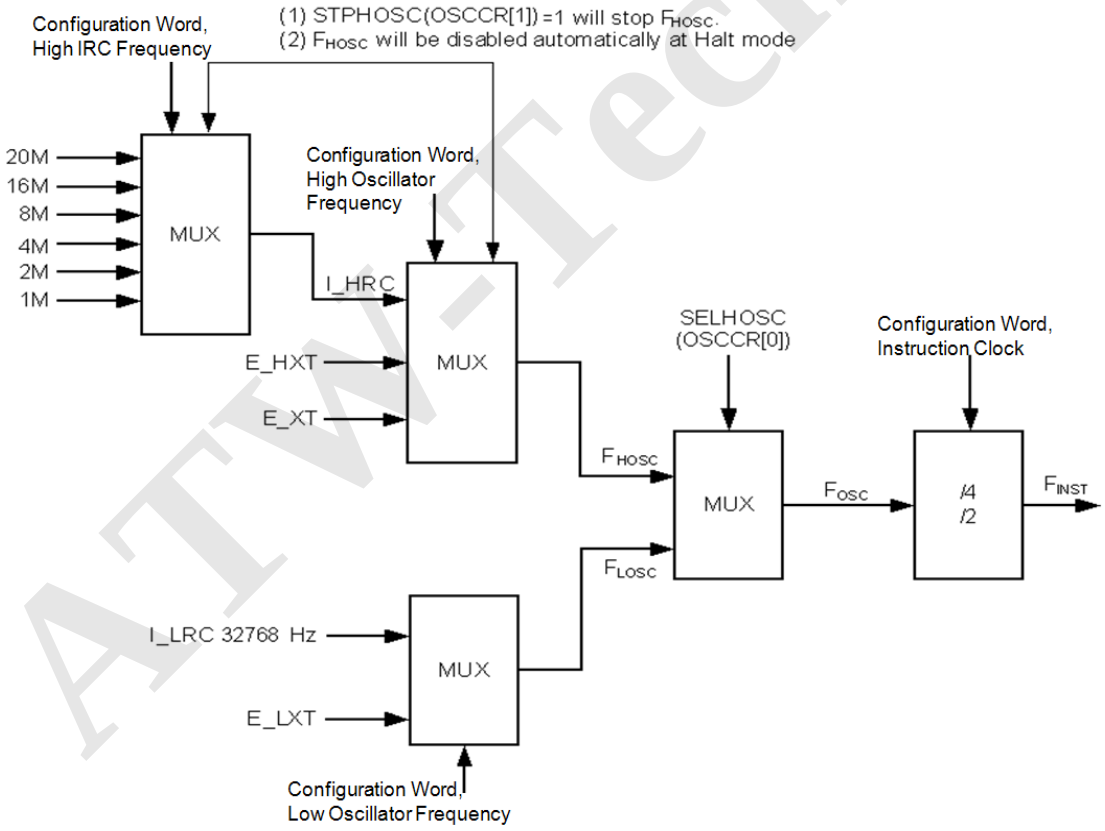


Figure34 Oscillation Configuration of AT8B62F1

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.	F_{HOSC}	F_{LOSC}
1	I_HRC	I_LRC
2	E_HXT or E_XT	I_LRC
3	I_HRC	E_LXT

Table 32 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 33 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 AT8B62F1 crystal pad built-in capacitance, which is about 5pF. For example, for crystal CL=12.5pF, C1=C2=20pF is recommended.

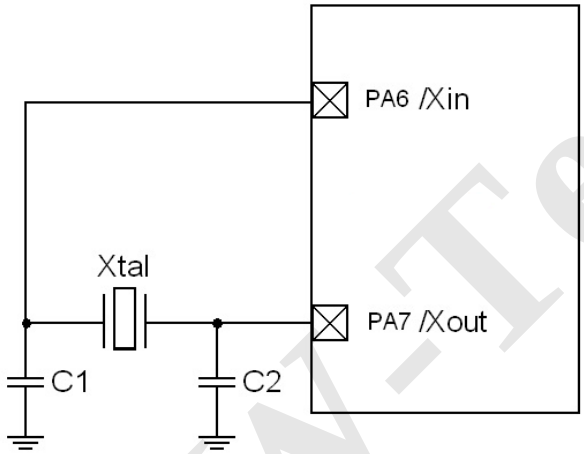


Figure 35 Connection for External Crystal Oscillation

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.

3.19 Operating Mode

AT8B62F1 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, AT8B62F1 will stop almost all operations except Timer0/Timer1/Timer2/ Timer3/WDT in order to wake-up periodically. At Halt mode, AT8B62F1 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.

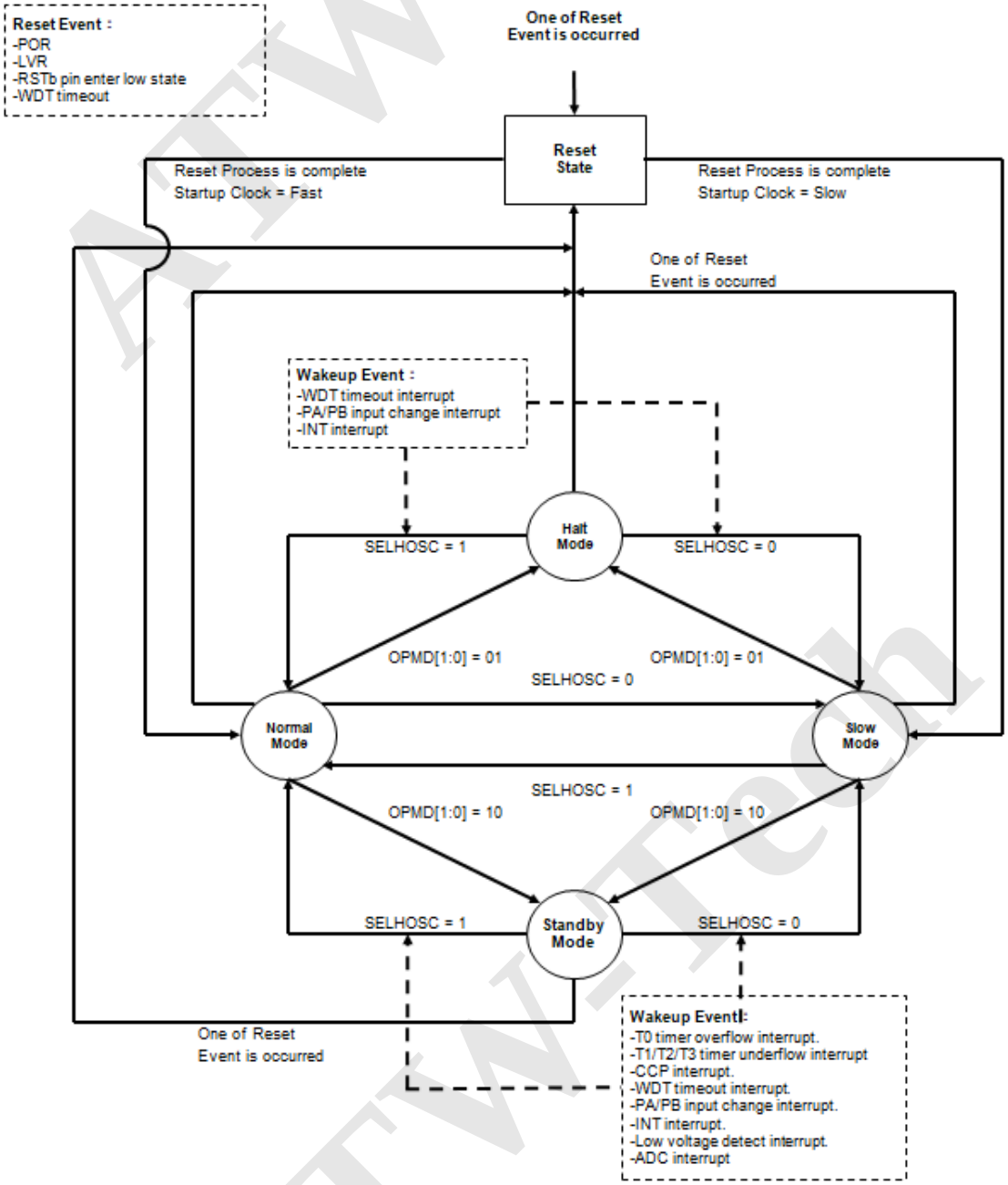


Figure36 Four Operating Modes

3.19.1 Normal Mode

After any Reset Event is occurred and Reset Process is completed, AT8B62F1 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, AT8B62F1 will enter Normal mode, if Startup Clock=Slow, AT8B62F1 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, AT8B62F1 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 AT8B62F1 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.

3.19.2 Slow Mode

AT8B62F1 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 AT8B62F1. Therefore user can write 1 to register bit STPHOSC (OSCCR[1]) in slow mode to reduce power consumption further. But it is noted that it is forbidden to enter slow mode and stop F_{HOSC} at the same time, one must enter slow mode first, then disable $F_{HOSC,}$ or the program may hang on.

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

3.19.3 Standby Mode

AT8B62F1 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 AT8B62F1 and user has to enter slow mode and write 1 to register bit STPHOSC in order to stop F_{HOSC} oscillation. Most of AT8B62F1 peripheral modules are disabled but Timer can be still active if register bit T0EN/T1EN/T2EN/T3EN is set to 1. Therefore AT8B62F1 can wake-up after Timer0/Timer1/Timer2/Timer3 is expired. The expiration period is determined by the register TMR0/TMR1[9:0]/TMR2[9:0]/TMR3[9:0], F_{INST} and other configurations for Timer0/Timer1/Timer2/Timer3.

- Instruction execution is stop and some peripheral modules may be active according to corresponding module enable bit.
- F_{HOSC} can be disabled by writing 1 to register bit STPHOSC.
- The F_{LOSC} is still active and running.
- IC can wake-up from Standby mode if any of (a) Timer0/Timer1/Timer2/Timer3 (overflow/underflow) interrupt, (b) WDT timeout interrupt, (c) PA/PB input change interrupt or (d) INT external interrupt is happened.
- After wake-up from Standby mode, IC will return to Normal mode if SELHOSC=1, IC will return to Slow mode if SELHOSC=0.
- It is not recommended to change oscillator mode (normal to slow / slow to normal) and enter standby mode at the same time.

3.19.4 Halt Mode

AT8B62F1 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 AT8B62F1 can only wake-up by some specific events. Therefore, Halt mode is the most power saving mode provided by AT8B62F1.

- Instruction execution is stop and all peripheral modules are disabled.
- F_{HOSC} and F_{LOSC} are both disabled automatically.
- IC can wake-up from Halt mode if any of (a) WDT timeout interrupt, (b) PA/PB input change interrupt or (c) INT or external interrupt is happened.
- After wake-up from Halt mode, IC will return to Normal mode if SELHOSC=1, IC will return to Slow mode if SELHOSC=0.
  Note: Users can change STPHOSC and enter Halt mode in the same instruction.
- It is not recommended to change oscillator mode (normal to slow or slow to normal) and enter halt mode at the same time.

3.19.5 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 AT8B62F1 enter Standby mode or Halt mode, user may execute instruction ENI. At this condition, AT8B62F1 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. The user should have NOP*2 after the SLEEP instruction of Wake-up from Sleep.

```
;choice one way to enter Halt mode
    ;sleep                                  ; 1. Execute instruction to enters Halt mode (from Normal mode)
    movia   C_Halt_Mode | C_FHOSC_Sel       ; 2. Set OSCCR register to enters Halt mode (from Normal mode)
    sfun    OSCCR                           ; Pr_SYS_Ctrl
;-------------------------------------------------------------------
    nop                                     ; for Interrupt latency time
    nop
    bcr     PORTB,3                         ; while wakeup from Halt mode then set PB3 outputs low
    movia   ~C_INT_PABKey
    movar   INTF                            ; Clear PABIF(PortB input change interrupt flag bit)
```

3.19.6 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 overflow - Timer1/2/3 underflow -CCP interrupt - WDT timeout - PA/PB input change - INT0/1 - LVD interrupt - Comparator interrupt - ADC end-of-convert	- WDT timeout - PA/PB input change - INT0/1

Table 34 Summary of Operating Modes

3.20 Reset Process

AT8B62F1 will enter Reset State and start Reset Process when one of following Reset Event is occurred:

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

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

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

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

After Reset Event is released, AT8B62F1 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, AT8B62F1 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.

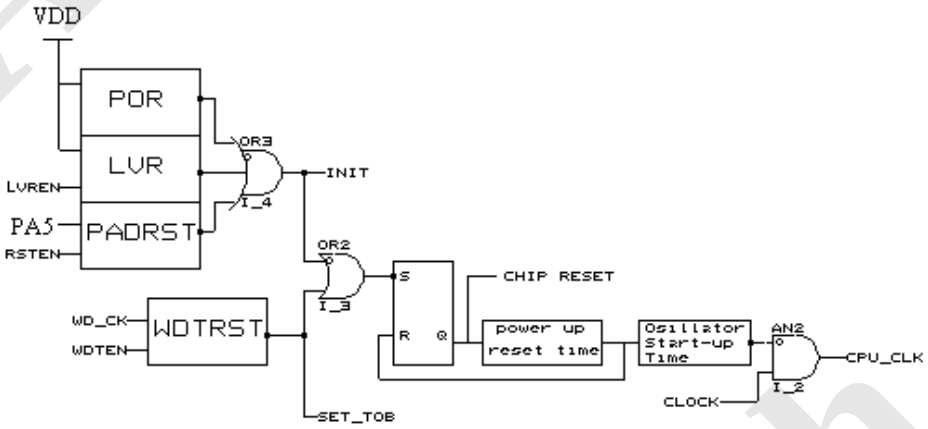


Figure37 Block diagram of on-chip reset circuit

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.

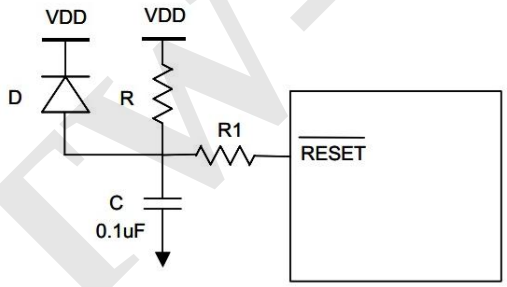


Figure38 Block Diagram of Reset Application

4. Instruction Set

AT8B62F1 provides 55 powerful instructions for all kinds of applications.

Inst.	OP 1	OP 2	Operation	Cyc.	Flag
Arithmetic Instructions					
ANDAR	R	d	dest = ACC & R	1	Z
IORAR	R	d	dest = ACC | R	1	Z
XORAR	R	d	dest = ACC ⊕ R	1	Z
ANDIA	i		ACC = ACC & i	1	Z
IORIA	i		ACC = ACC | i	1	Z
XORIA	i		ACC = ACC ⊕ i	1	Z
RRR	R	d	Rotate right R	1	C
RLR	R	d	Rotate left R	1	C
BSR	R	bit	Set bit in R	1	-
BCR	R	bit	Clear bit in R	1	-
INCR	R	d	Increase R	1	Z
DECR	R	d	Decrease R	1	Z
COMR	R	d	dest = ~R	1	Z
Conditional Instructions					
BTRSC	R	bit	Test bit in R, skip if clear	1 or 2	-
BTRSS	R	bit	Test bit in R, skip if set	1 or 2	-
INCRSZ	R	d	Increase R, skip if 0	1 or 2	-
DECRSZ	R	d	Decrease R, skip if 0	1 or 2	-
Data Transfer Instructions					
MOVAR	R		Move ACC to R	1	-
MOVR	R	d	Move R	1	Z
MOVIA	i		Move immediate to ACC	1	-
SWAPR	R	d	Swap halves R	1	-
IOST	F		Load ACC to F-page SFR	1	-
IOSTR	F		Move F-page SFR to ACC	1	-
SFUN	S		Load ACC to S-page SFR	1	-
SFUNR	S		Move S-page SFR to ACC	1	-
T0MD			Load ACC to T0MD	1	-
T0MDR			Move T0MD to ACC	1	-
TABLEA			Read ROM	2	-

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

Table36 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 ≤ R ≤ 127 d = 0, 1.
Operation:	R - 1 → dest, Skip if result = 0
Status affected:	--
Description:	Decrease R first. If d is 0, the result is stored in ACC. If d is 1, the result is stored back to R. If result is 0, the next instruction which is already fetched is discarded and a NOP is executed instead. Therefore it makes this instruction a two-cycle instruction.
Cycle:	1 or 2(skip)
Example:	DECRSZ R, d instruction2 instruction3 before executing instruction: R=0x1, d=1, Z=0. after executing instruction: R=0x0, Z=1, and instruction will skip instruction2 execution because the operation result is zero.

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

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

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

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.

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

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.

INCRSZ	Increase R, Skip if 0

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

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

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

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

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

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

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

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

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

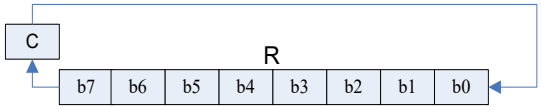
NOP	No Operation
Syntax:	NOP
Operand:	--
Operation:	No operation.
Status affected:	--
Description:	No operation.
Cycle:	1
Example:	NOP before executing instruction: PC=A0 after executing instruction: PC=A0+1

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

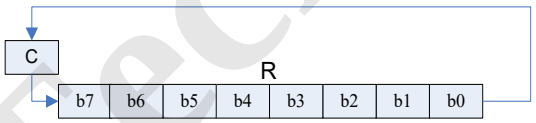
RETIA	Return with Data in ACC
Syntax:	RETIA i
Operand:	0 ≤ i < 255
Operation:	i → ACC, Top of Stack → PC
Status affected:	--
Description:	ACC is loaded with 8-bit immediate data i and PC is loaded from top of Stack as return address.
Cycle:	2
Example:	RETIA i before executing instruction: Stack pointer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

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

RLR	Rotate Left R Through Carry
Syntax:	RLR R, d
Operand:	0 ≤ R ≤ 127 d = 0, 1.
Operation:	C → dest[0], R[7] → C, R[6:0] → dest[7:1]
Status affected:	C
Description:	The content of R is rotated one bit to the left through flag Carry. If d is 0, the result is placed in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	RLR R, d before executing instruction: R=0xA5, d=1, C=0. after executing instruction: R=0x4A, C=1.



RRR	Rotate Right R Through Carry
Syntax:	RRR R, d
Operand:	0 ≤ R ≤ 127 d = 0, 1.
Operation:	C → dest[7], R[7:1] → dest[6:0], R[0] → C
Status affected:	C
Description:	The content of R is rotated one bit to the right through flag Carry. If d is 0, the result is placed in ACC. If d is 1, the result is stored back to R.
Cycle:	1
Example:	RRR R, d before executing instruction: R=0xA5, d=1, C=0. after executing instruction: R=0x52, C=1.



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

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

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

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

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

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

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

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

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

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

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

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

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

5. Configuration Words

Item	Name	Options
1	High Oscillator Frequency	1. I_HRC 2. 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. 8MHz > F_{HOSC} > 6MHz 2. 10MHz > F_{HOSC} ≧ 8MHz 3. 12MHz > F_{HOSC} ≧ 10MHz 4. 16MHz > F_{HOSC} ≧ 12MHz 5. 20MHz > F_{HOSC} ≧ 16MHz 6. 20MHz
5	Instruction Clock	0. 4 oscillator period 1. 2 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. PA5 is I/O 2. PA5 is reset
10	PA.7	1. PA7 is I/O 2. PA7 is instruction clock output
11	Startup Time	1. 140us 2. 4.5ms 3. 18ms 4. 72ms 5. 288ms
12	WDT Time Base	1. 3.5ms 2. 15ms 3. 60ms 4. 250ms
13	LVR Setting	1. Register Control 2. LVR Always On
14	LVR Voltage	1. 1.8V 2. 2.0V 3. 2.2V 4. 2.4V 5. 2.7V 6. 3.0V 7. 3.3V 8. 3.6V
15	VDD Voltage	1. 3.0V 2. 4.5V 3. 5.0V
16	Sink current (exclude PA5)	1. Large 2. Normal
17	Comparator Input pin select	1. Enable 2. Disable
18	Read Output Data	1. I/O Port 2. Register
19	E_LXT Backup Control	1. Auto Off 2. Register Off
20	EX_CKI0 to Inst. Clock	1. Sync 2. Async
21	Startup Clock	1. Fast (I_HRC/E_HXT/E_XT) 2. Slow (I_LRC/E_LXT)
22	Input Schmitt Trigger	1 Enable 2. Disable (0.5VDD)
23	Input High Voltage (VIH)	1. 0.7VDD 2. 0.5VDD
24	Input Low Voltage (VIL)	1. 0.3VDD 2. 0.2VDD
25	PWM2 output	1. PA4 2. PB2
26	PWM4 output	1. PA3 2. PA7 3. Disable
27	PWM5 output	1. PB3 2. PB0 3. Disable

Table 37 Configuration Words

6. Electrical Characteristics

6.1 Absolute Maximum Rating

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

6.2 DC Characteristics

(All refer F_{INST}=F_{HOSC}/4, F_{HOSC}=16MHz or 20MHz@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	--	3.0	--	5.5	V	F_{INST}=10MHz @I_HRC 20MHz/2T
			2.2				F_{INST}=5MHz @I_HRC 20MHz/4T
			2.7				F_{INST}=8MHz @I_HRC 16MHz/2T
			2.0				F_{INST}=4MHz @I_HRC 16MHz/4T
							F_{INST}=4MHz @E_HXT 8MHz/2T
			1.8				F_{INST}=2MHz @I_HRC 8MHz/4T
							F_{INST}=2MHz @I_HRC 4MHz/2T
			1.6				F_{INST}=1MHz @I_HRC 4MHz/4T
			1.6				F_{INST}=8KHz @I_LRC: 32KHz/4T
							F_{INST}=8KHz @E_LXT: 32KHz/4T
V_{IH}	Input high voltage	5V	4.1	--	--	V	RSTb (0.8 V_{DD})
		3V	2.4	--	--		
		5V	3.5	--	--	V	All other I/O pins, EX_CKI0/1, INT0/1 (0.7 V_{DD})
		3V	2.1	--	--		
		5V	2.5	--	--	V	All other I/O pins, EX_CKI0/1 (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 (0.3 V_{DD})
		3V	--	--	0.9		
		5V	--	--	1.1	V	All other I/O pins, EX_CKI0/1 (0.2 V_{DD})
		3V	--	--	0.6		
I_{OH}	Output high current	5V	--	18	--	mA	V_{OH}=4.0V
		3V	--	10	--		V_{OH}=2.0V
I_{OL}	Output low current (Large current)	5V	--	40	--	mA	V_{OL}=1.0V
		3V	--	25	--		
I_{OL}	Output low current (Normal current)	5V	--	28	--	mA	V_{OL}=1.0V
		3V	--	17	--		
I_{IR}	IR sink current	5V	--	43	--	mA	V_{OL}=1.0V
		3V	--	28	--		

Symbol	Parameter	V_{DD}	Min.	Typ.	Max.	Unit	Condition
I_{OP}	Operating current	**Normal Mode**					
		5V	--	1.7	--	mA	F_{INST} =10MHz @I_HRC&E_HXT 20MHz/2T
		3V	--	0.7	--		
		5V	--	1.4	--	mA	F_{INST} =5MHz/4T @I_HRC&E_HXT 20MHz/4T
		3V	--	0.5	--		
		5V	--	1.6	--	mA	F_{INST} =8MHz @I_HRC & E_HXT 16MHz/2T
		3V	--	0.6	--		
		5V	--	1.3	--	mA	F_{INST} =4MHz @I_HRC & E_HXT 16MHz/4T
		3V	--	0.5	--		
		5V	--	1.1	--	mA	F_{INST} = 2MHz @I_HRC& E_HXT 8MHz/4T
		3V	--	0.4	--		
		5V	--	1.0	--	mA	F_{INST} =1MHz @I_HRC & E_XT 4MHz/4T
		3V	--	0.4	--		
		Slow Mode					
		5V	--	11	--	uA	F_{HOSC} disabled, F_{LOSC}=32KHz/2T (I_LRC)
		3V	--	6.1	--		
		5V	--	11	--	uA	F_{HOSC} disabled, F_{LOSC}=32KHz/2T (E_LXT)
		3V	--	4.9	--		
		5V	--	7.3	--	uA	F_{HOSC} disabled, F_{LOSC}=32KHz/4T (I_LRC)
		3V	--	4.3	--		
		5V	--	8.5	--	uA	F_{HOSC} disabled, F_{LOSC}=32KHz/4T (E_LXT)
		3V	--	3.6	--		
I_{STB}	Standby current	5V	--	3.8	--	uA	Standby mode, F_{HOSC} disabled, F_{LOSC}=32KHz/4T (I_LRC)
		3V	--	2.6	--		
I_{HALT}	Halt current	5V	--	--	0.5	uA	Halt mode, WDT disabled.
		3V	--	--	0.2		
		5V	--	--	5.0	uA	Halt mode, WDT enabled.
		3V	--	--	3.0		
R_{PH}	Pull-High resistor	5V	--	50	--	KΩ	Pull-High resistor (include PA4)
		3V	--	100	--		
		5V	--	85	--	KΩ	Pull-High resistor (PA5)
		3V	--	85	--		
		5V		0.6		MΩ	Pull-High resistor(PA4 1M+HiLH)
		3V		1		MΩ	
R_{PL}	Pull-Low resistor	5V	--	50	--	KΩ	Pull-Low resistor
		3V	--	100	--		

6.3 OSC Characteristics

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

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

6.4 Comparator / LVD Characteristics

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

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

***Note: These parameters are for design reference, not tested for each chip.**

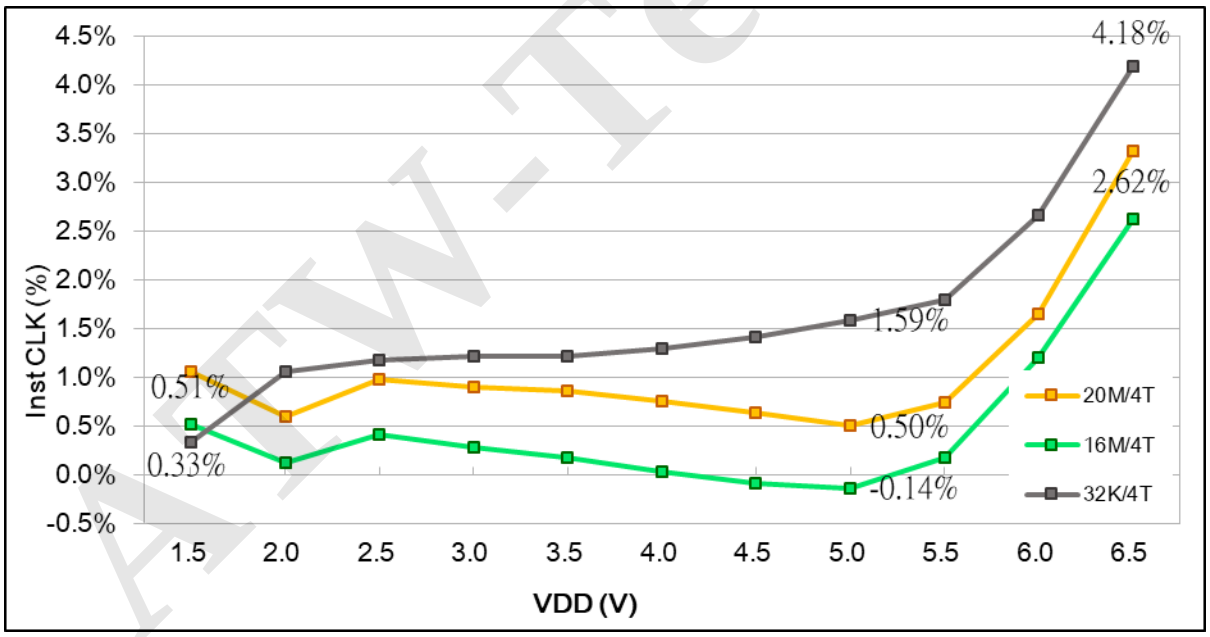
6.5 ADC Characteristics

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

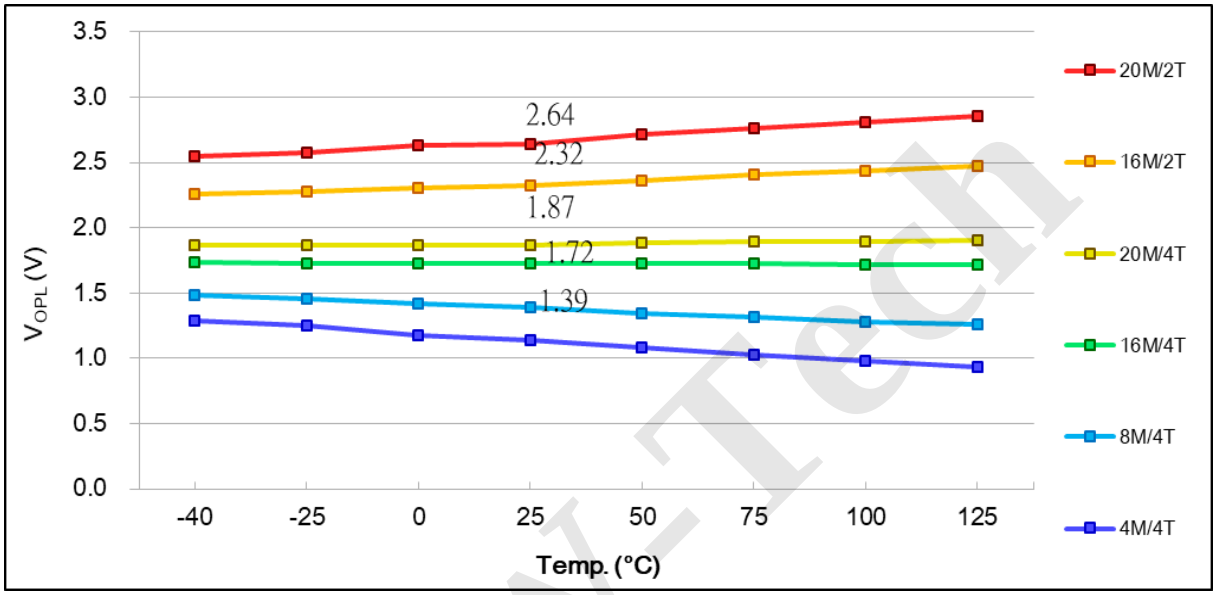
Symbol	Parameter	Min.	Typ.	Max.	Unit	Condition
V_{REFH}	VREFH input voltage	2V	--	V_{DD}	V	Ext. reference voltage
V_{REF4}	Int. 4V reference voltage, V_{DD}=5V	3.96	4	4.04	V	
V_{REF3}	Int. 3V reference voltage, V_{DD}=5V	2.97	3	3.03	V	
V_{REF2}	Int. 2V reference voltage, V_{DD}=5V	1.98	2	2.02	V	
V_{REF}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

6.6 Characteristic Graph

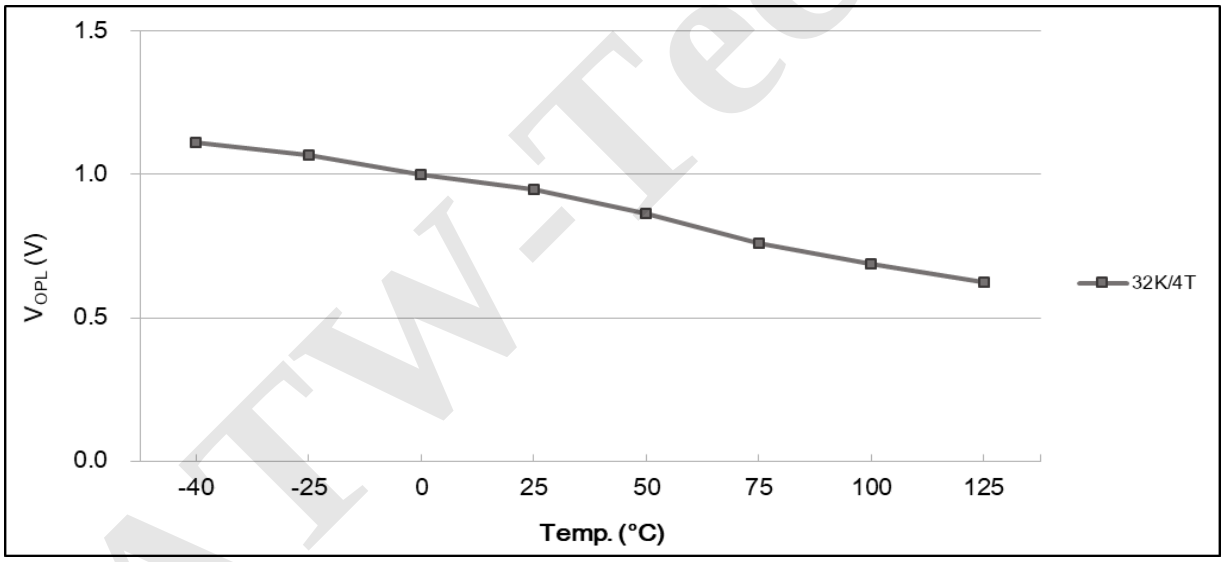
6.6.1 Frequency vs. V_{DD} of I_HRC, I_LRC



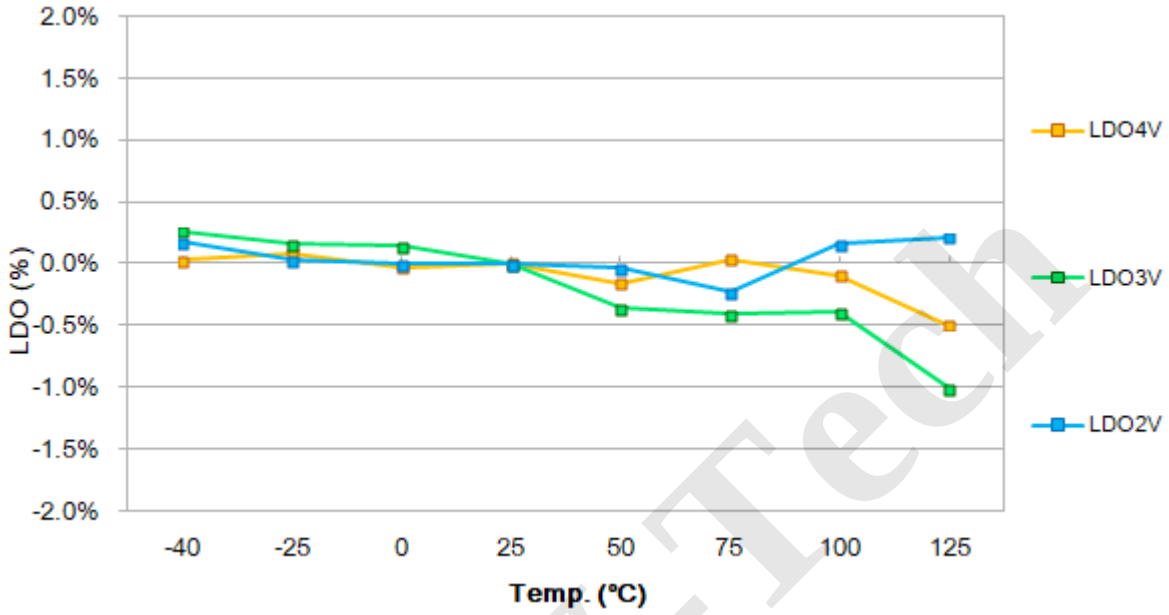
6.6.2 Frequency vs. Temperature of I_HRC



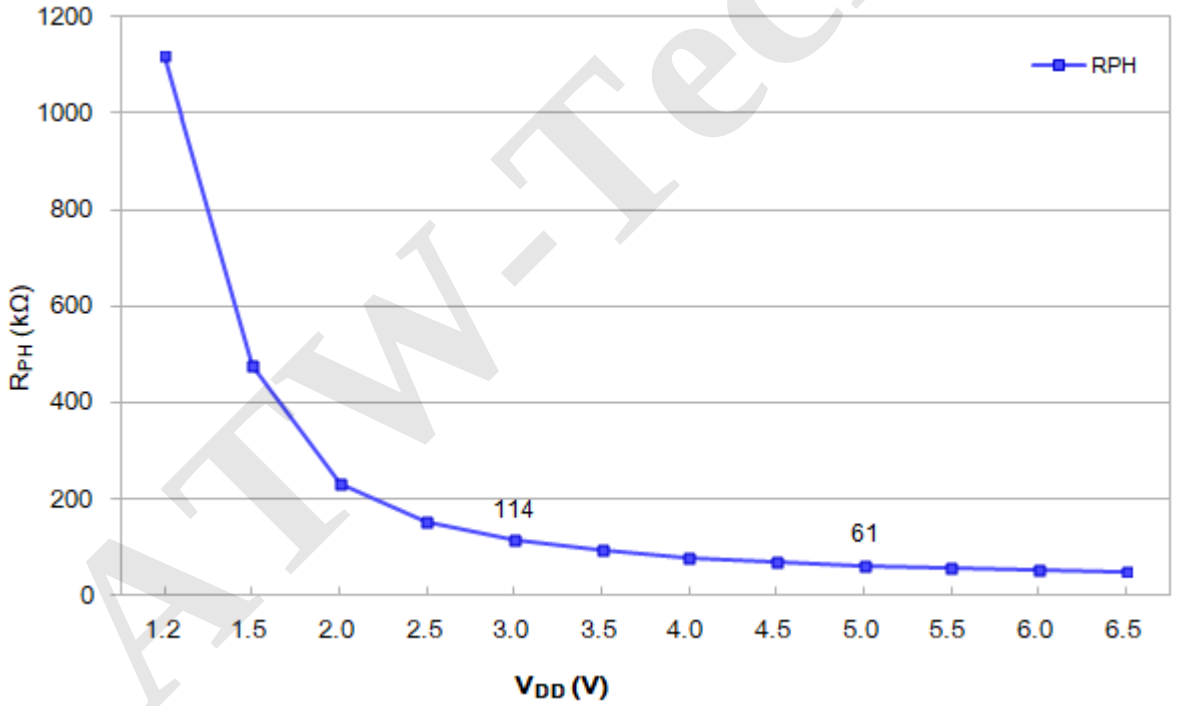
6.6.3 Frequency vs. Temperature of I_LRC



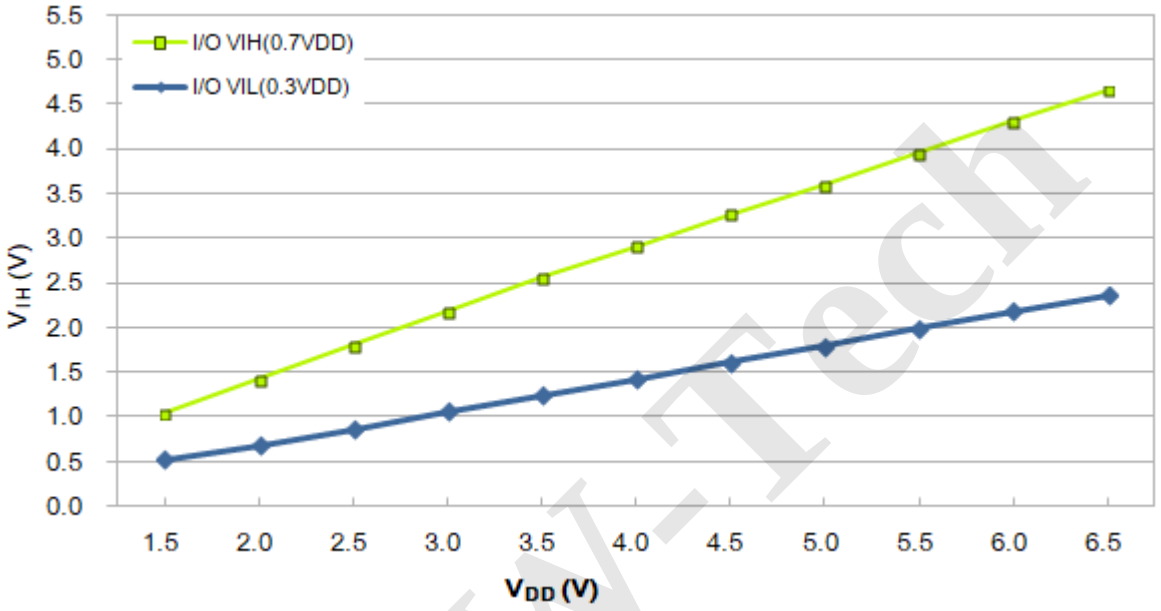
6.6.4 Low Dropout Regulator vs. Temperature

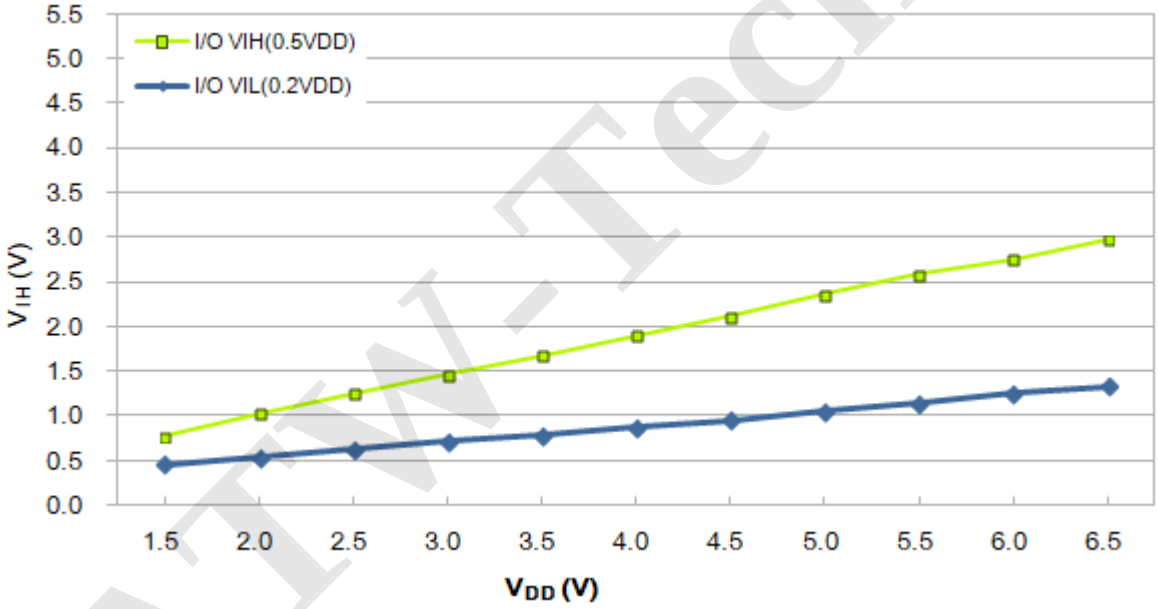


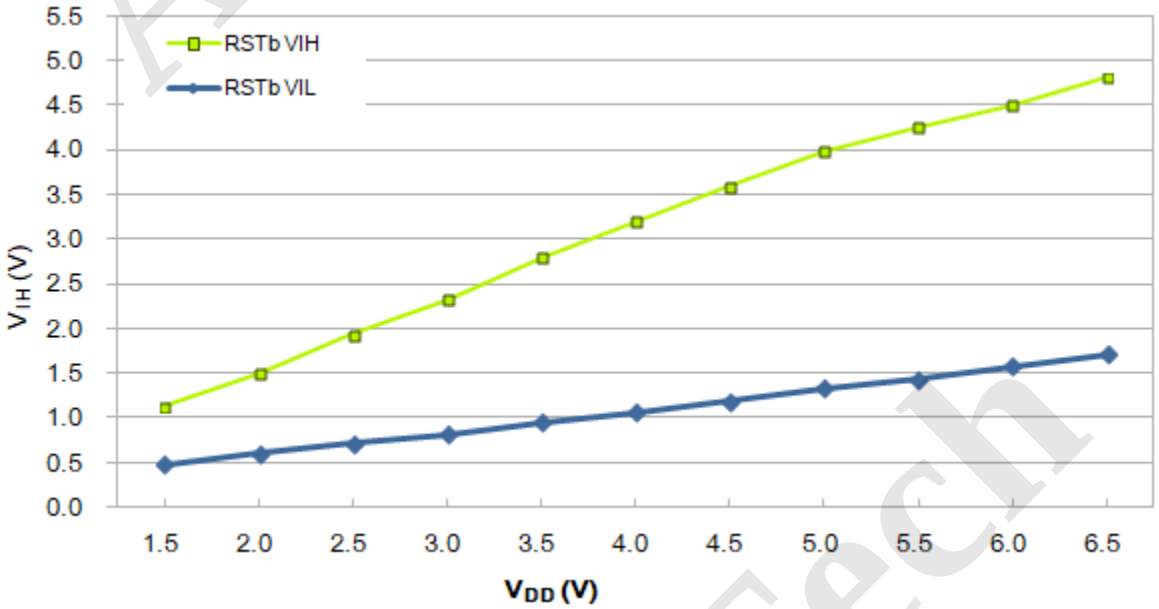
6.6.5 Pull High Resistor vs. VDD



6.6.6 VIH/VIL vs. VDD







6.7 Recommended Operating Voltage

Frequency	LVR default Setting	LVR Setting (Min. @ 25℃)
20M/2T	3.6V	3.3V
16M/2T	3.3V	3.0V
20M/4T	2.4V	2.2V
16M/4T	2.2V	2.0V
8M/4T	1.8V	1.8V
4M/4T	1.8V	1.8V

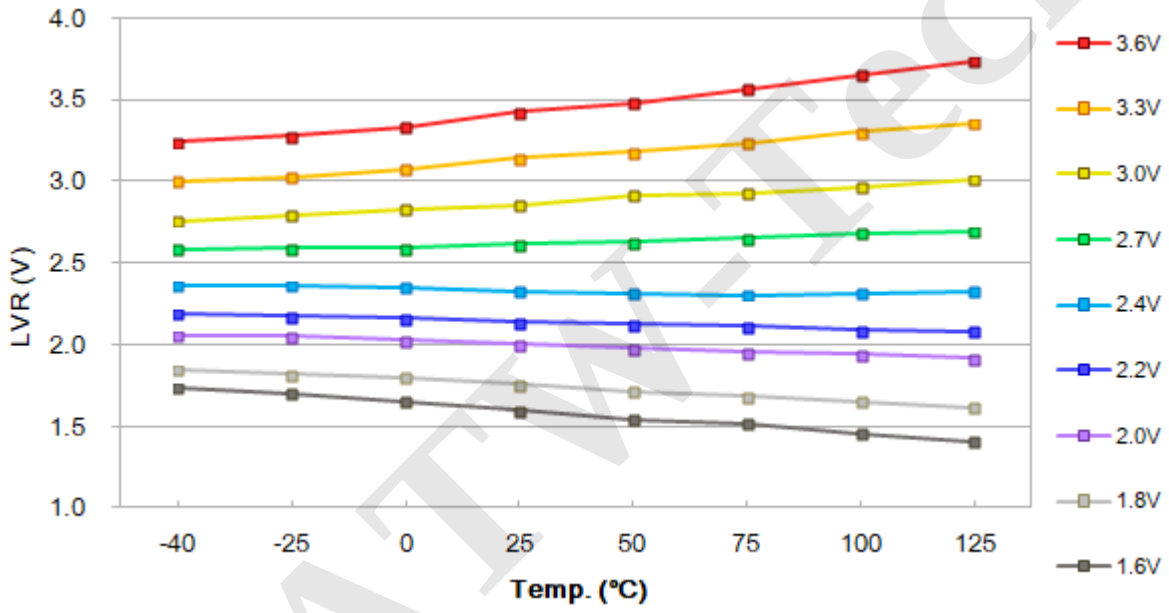
*Note: These parameters are for design reference, not tested for each chip.

Bypass Prescaler

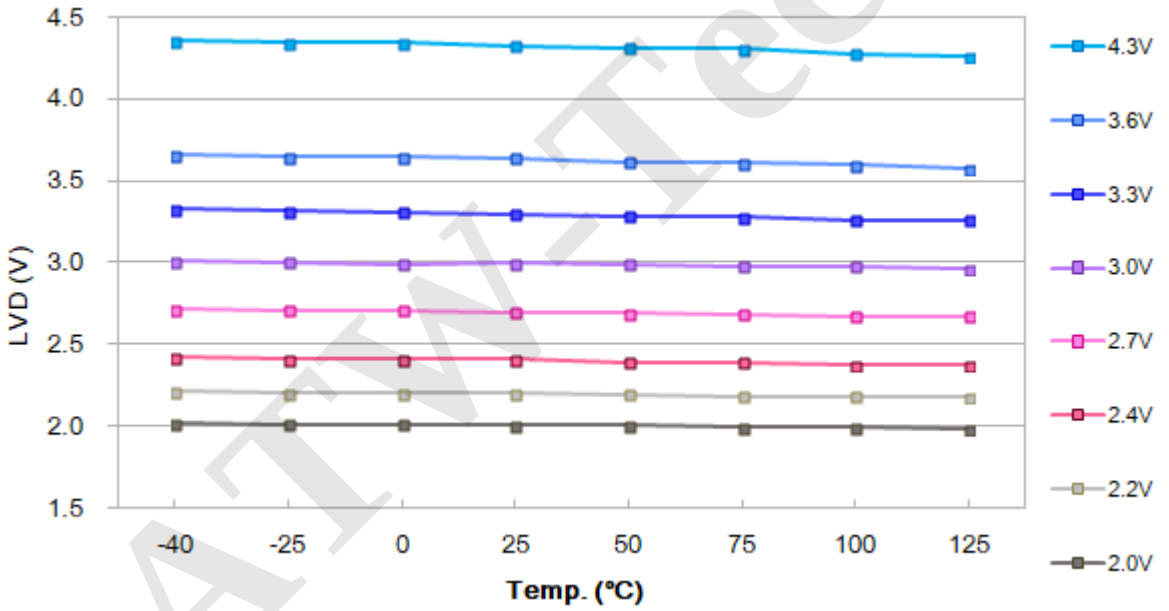
Frequency	LVR : Recommended Bypass Prescaler
20M/2T	2.7V @ I_HRC=20MHz
16M/2T	2.7V @ I_HRC=16MHz
20M/4T	2.7V @ I_HRC=20MHz
16M/4T	2.4V @ I_HRC=16MHz
8M/4T	2.4V @ I_HRC=8MHz
4M/4T	2.4V @ I_HRC=4MHz

*Note: These parameters are for design reference, not tested for each chip.

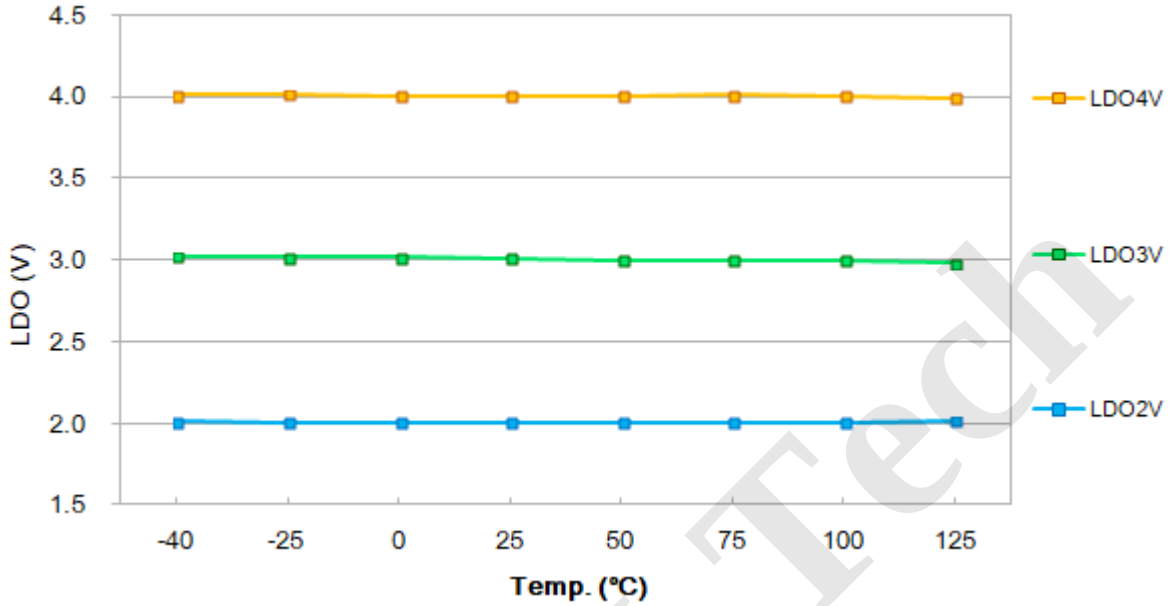
6.8 LVR vs. Temperature



6.9 LVD vs. Temperature

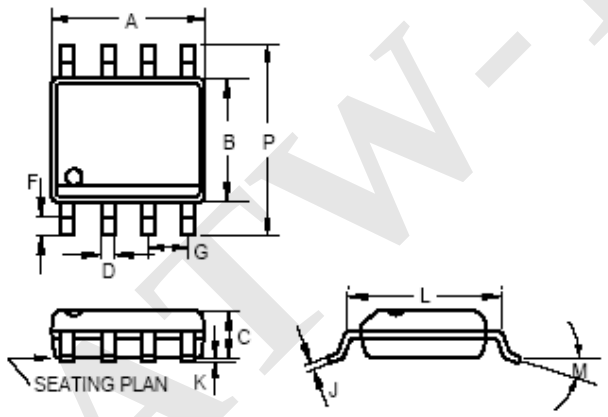


6.10 LDO vs. Temperature



7. Package Dimension

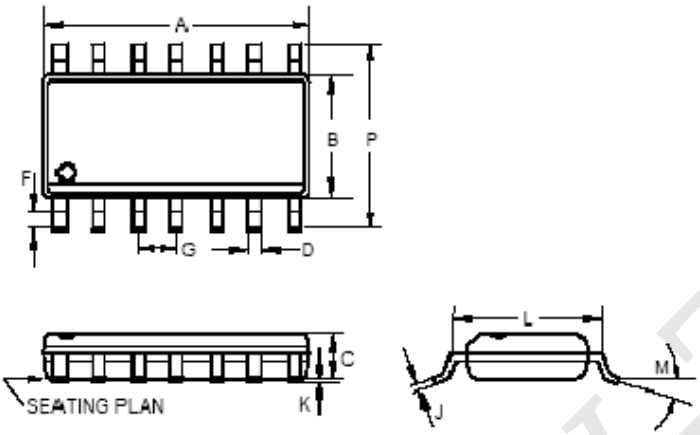
7.1 8-Pin Plastic SOP (150 mil)



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

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

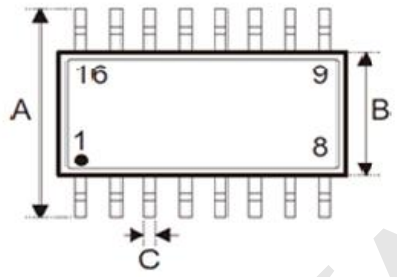
7.2 14-Pin Plastic SOP (150 mil)

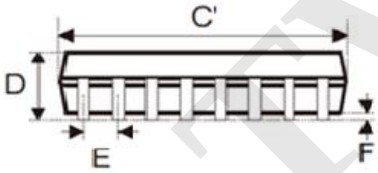


	INCHES			MILLIMETERS		
	MIN	TYP	MAX	MIN	TYP	MAX
A	0.337	-	0.344	8.55	-	8.75
B	0.144	-	0.163	3.66	-	4.14
C	0.068	-	0.074	1.73	-	1.88
D	0.017	-	0.020	0.35	-	0.51
F	0.016	-	0.044	0.40	-	1.12
G	0.050 BSC			1.27 BSC		
J	-	0.004	-	-	0.10	-
K	0.005	-	0.010	0.13	-	0.25
L	0.189	-	0.205	4.80	-	5.21
M	-	-	8°	-	-	8°
R	0.228	-	0.244	5.80	-	6.20

Note: For 14-pin SOP, 50 units per tube.

7.3 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.

8. Ordering Information

P/N	*Package Type*	*Pin Count*	*Package Width*	*Shipping*
AT8B62F1S8	SOP	8	150 mil	Tape & Reel: 2.5K pcs per Reel Tube: 100 pcs per Tube
AT8B62F1S14	SOP	14	150 mil	Tape & Reel: 2.5K pcs per Reel Tube: 50 pcs per Tube
AT8B62F1S16	SOP	16	150 mil	Tape & Reel: 2.5K pcs per Reel Tube: 50 pcs per Tube