SONiX Technology Co., Ltd.

SN8F5754 Series
Datasheet

8051-based Microcontroller

SN8F5754
SN8F5753
SN8F5752

1 Device Overview

1.1 Features

- **Enhanced 8051 microcontroller** with reduced instruction cycle time (up to 12 times 80C51)
- Up to 8 MHz flexible CPU frequency
- Internal 32 MHz Clock Generator (IHRC), 1 MHz to 16 MHz crystal, and external synchronous clock source selections
- 1-set external low clock 32.768 kHz crystal
- **Memory configuration**
  16 KB on-chip Flash program memory (IROM)
  256 bytes on-chip internal RAM (IRAM)
  1 KB on-chip external RAM (XRAM)
- **14 interrupt sources with priority levels control and unique interrupt vectors**
- 11 internal interrupts
- 3 external interrupts: INT0/1/2
- 2 set of DPTR
- 2 set 8/16-bit timers with 4 operation modes
- 1 set 16-bit timers with adjustable cycle
- **2 set 16-bit PWM generators**:
  Each PWM generator has 4 output channels with individual duty cycle control
- **16-bit CRC Generator**
- **12-bit SAR ADC** with 16 external and 3 internal channels, and 4 internal reference voltages
- **23 channel Capacitive touch**
- **2 set UARTs with individual interrupt vector**
- **1 set I2C** interface with SMBus Support
- **One channel 2KHz/4KHz buzzer output**
- **On-Chip Debug Support**:
  Single-wire debug interface
  3 hardware breakpoints
  Unlimited software breakpoints
  ROM data security/protection
- **Debug interface ON/OFF control**
- Watchdog and programmable external reset
- 1.8 low voltage detectors
- Wide supply voltage (1.8 V – 5.5 V) and temperature (-40 °C to 85 °C) range

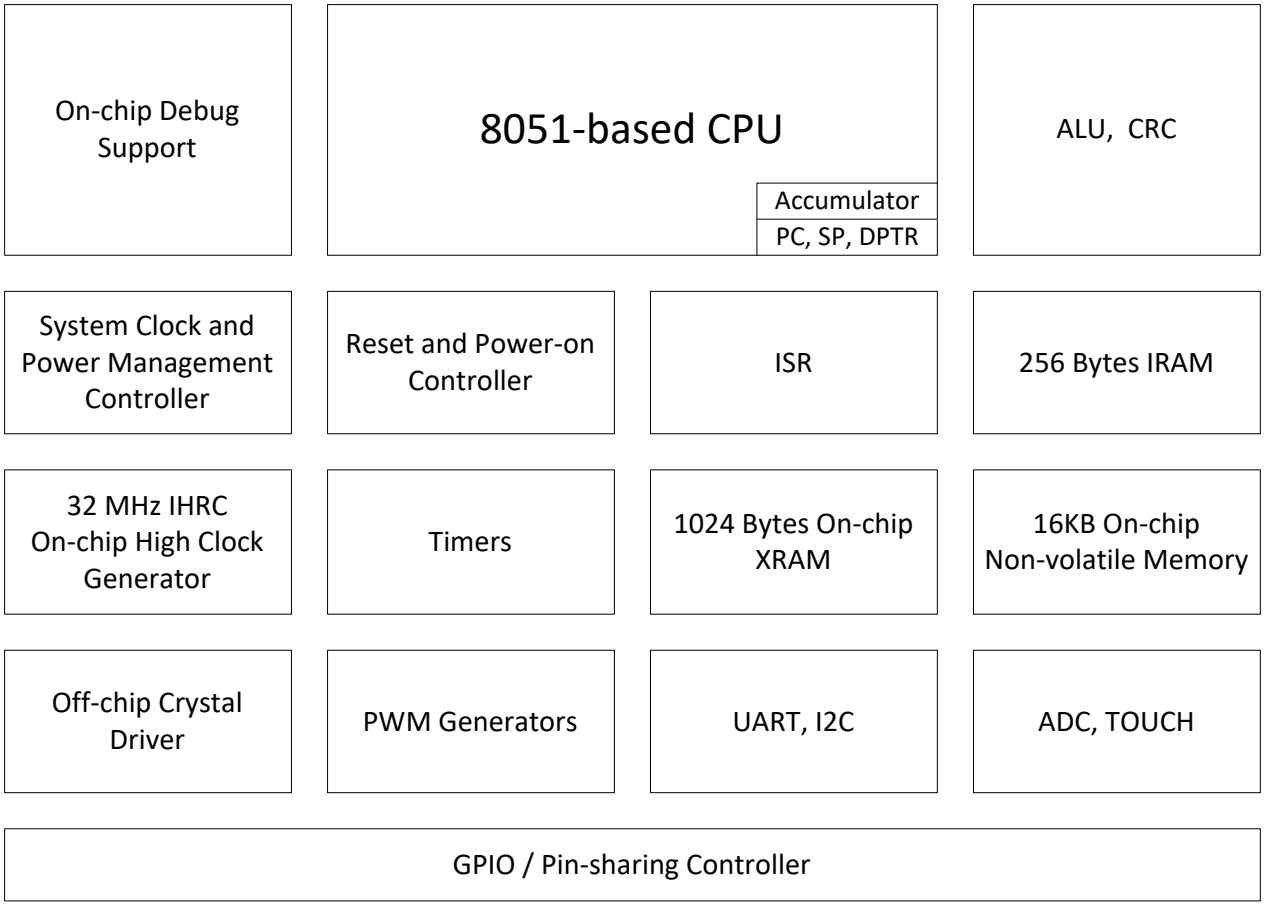
1.2 Applications

- Home automation
- Touch Key

1.3 Features Selection Table

	I/O	PWM Channels	I2C	UART	ADC ext. Channels	Touch	Ext. INT	CRC	Package Types
SN8F5754	25	8	1	2	16	23	3	V	SKDIP28 SOP28 TSSOP28
SN8F5753	21	6	1	2	16	19	3	V	SOP24 TSSOP24
SN8F5752	17	6	1	2	13	15	3	V	SOP20

1.4 Block Diagram



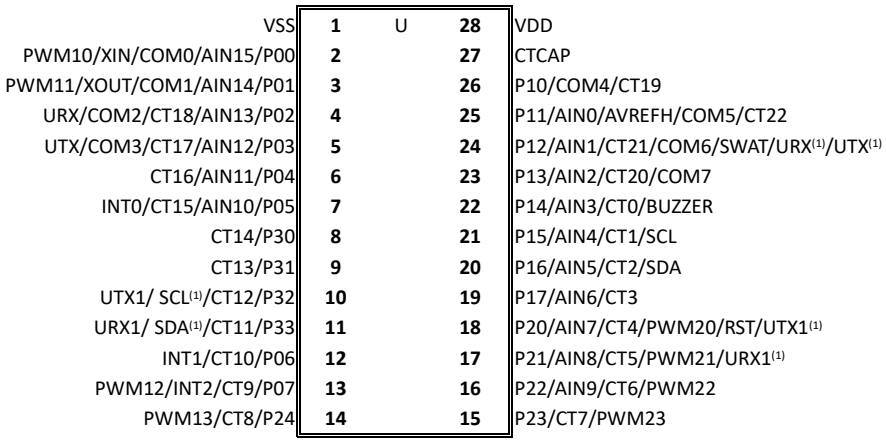
2 Table of Contents

3 Revision History

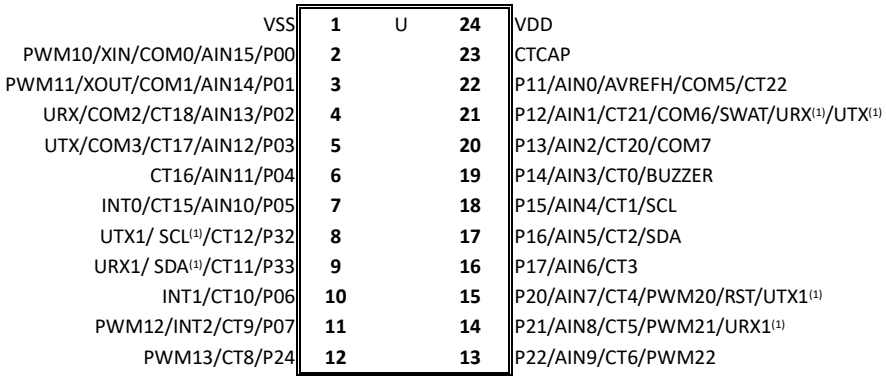
Revision	Date	Description
1.0	July. 2021	First issue.
1.1	Nov. 2021	Modify typing errors.
1.2	Apr. 2024	1. Add note for pin assignments and GPIO section about not pin out I/O. 2. Add note for ISP operation. 3. Modify electrical characteristic section. 4. Modify memory description in feature table 5. Modify typing error
1.3	Jun.2024	1. Modify ordering Information section. 2. Remove Sample codes.
1.4	Jan.2025	1. Modify PWM duty descriptions. 2. Update GPIO description

4 Pin Assignments

4.1 SN8F5754 (SKDIP28/SOP28/TSSOP28)

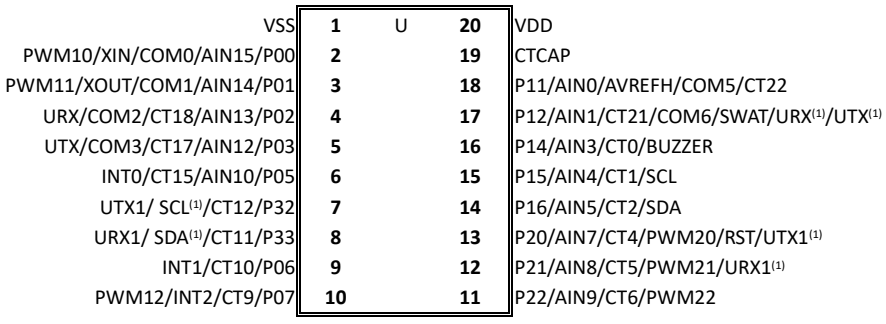


4.2 SN8F5753 (SOP24/TSSOP24)



✷ ***Note: The pins which are not pin-out shall be set correctly to decrease power consumption in low-power modes. Strongly recommended to set these pins as input pull-up.***

4.3 SN8F5752 (SOP20)



✷ ***Note: The pins which are not pin-out shall be set correctly to decrease power consumption in low-power modes. Strongly recommended to set these pins as input pull-up.***

4.4 Pin Descriptions

Power Pins

Pin Name	Type	Description
VDD	Power	Power supply
VSS	Power	Ground (0 V)

Port 0

Pin Name	Type	Description
P0.0	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
PWM10	Digital Output	PWM1: programmable PWM output
COM0	Analog Output	1/2*VDD voltage generator output pin.
AIN15	Analog Input	ADC input channel 15.
XIN	Analog Input	System clock: external clock input
P0.1	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
PWM11	Digital Output	PWM1 : programmable PWM output
COM1	Analog Output	1/2*VDD voltage generator output pin.
AIN14	Analog Input	ADC input channel 14.
XOUT	Analog Output	System clock: drive external crystal/resonator
P0.2	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
URX	Digital Input	UART0 : reception pin.
COM2	Analog Output	1/2*VDD voltage generator output pin.
AIN13	Analog Input	ADC input channel 13
CT18	Analog Input	Cap-sensing touch input channel 18.
P0.3	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
UTX	Digital Output	UART0 : transmission pin.
COM3	Analog Output	1/2*VDD voltage generator output pin.
AIN12	Analog Input	ADC input channel 12.

CT17	Analog Input	Cap-sensing touch input channel 17.
P0.4	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
AIN11	Analog Input	ADC input channel 11.
CT16	Analog Input	Cap-sensing touch input channel 16.
P0.5	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
INT0	Digital Input	INT0 : external interrupt 0.
AIN10	Analog Input	ADC input channel 10.
CT15	Analog Intput	Cap-sensing touch input channel 15.
P0.6	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
INT1	Digital Input	INT1 : external interrupt 1.
CT10	Analog Input	Cap-sensing touch input channel 10.
P0.7	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
PWM12	Digital Output	PWM1: programmable PWM output.
INT2	Digital Input	INT2: external interrupt 2.
CT9	Analog Input	Cap-sensing touch input channel 9.

Port 1

Pin Name	Type	Description
P1.0	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
COM4	Analog Output	1/2*VDD voltage generator output pin.
CT19	Analog Input	Cap-sensing touch input channel 19.
P1.1	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
COM5	Analog Output	1/2*VDD voltage generator output pin.
AIN0	Analog Input	ADC: input channel 0.
AVREFH	Analog Input	ADC: external reference voltage.
CT22	Analog Input	Cap-sensing touch input channel 22.
P1.2	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SWAT	Digital I/O	Debug interface.
URX(1)	Digital Input	UART0: reception pin.
UTX(1)	Digital Output	UART0: transmission pin.
COM6	Analog Output	1/2*VDD voltage generator output pin.
AIN1	Analog Input	ADC: input channel 1.
CT21	Analog Input	Cap-sensing touch input channel 21.
P1.3	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
COM7	Analog Output	1/2*VDD voltage generator output pin.
AIN2	Analog Input	ADC: input channel 2.
CT20	Analog Input	Cap-sensing touch input channel 20.
P1.4	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
BUZZER	Digital Output	BUZZER: 2KHz/4KHz buzzer output.
AIN3	Analog Input	ADC: input channel 3.
CT0	Analog Input	Cap-sensing touch input channel 0.
P1.5	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SCL	Digital I/O	I2C: clock output (master) or clock input (slave).
AIN4	Analog Input	ADC: input channel 4.
CT1	Analog Input	Cap-sensing touch input channel 1.
P1.6	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SDA	Digital I/O	I2C: data pin.

AIN5	Analog Input	ADC: input channel 5.
CT2	Analog Input	Cap-sensing touch input channel 2.
P1.7	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
AIN6	Analog Input	ADC: input channel 6.
CT3	Analog Input	Cap-sensing touch input channel 3.

Port 2

Pin Name	Type	Description
P2.0	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
UTX1(1)	Digital Output	UART1: transmission pin.
PWM20	Digital Output	PWM2: programmable PWM output.
RST	Digital Input	System reset (active low).
AIN7	Analog Input	ADC input channel 7.
CT4	Analog Input	Cap-sensing touch input channel 4.
P2.1	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
URX1(1)	Digital Output	UART1: reception pin.
PWM21	Digital Output	PWM2: programmable PWM output.
AIN8	Analog Input	ADC input channel 8.
CT5	Analog Input	Cap-sensing touch input channel 5.
P2.2	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
PWM22	Digital Output	PWM2: programmable PWM output.
AIN9	Analog Input	ADC input channel 9.
CT6	Analog Input	Cap-sensing touch input channel 6.
P2.3	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
PWM23	Digital Output	PWM2: programmable PWM output.
CT7	Analog Input	Cap-sensing touch input channel 7.
P2.4	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
PWM13	Digital Output	PWM1: programmable PWM output.
CT8	Analog Input	Cap-sensing touch input channel 8.

Port 3

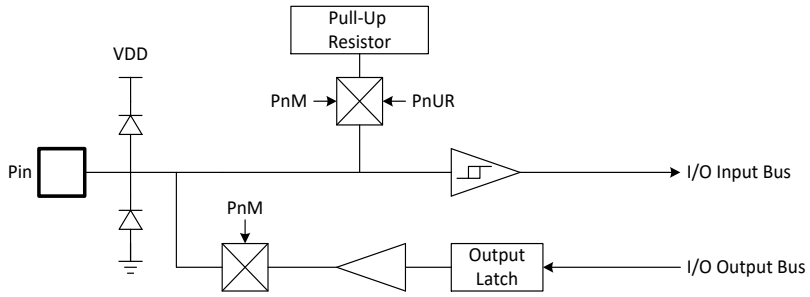
Pin Name	Type	Description
P3.0	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
CT14	Analog Input	Cap-sensing touch input channel 14.
P3.1	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
CT13	Analog Input	Cap-sensing touch input channel 13.
P3.2	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
UTX1	Digital Output	UART1: transmission pin.
SCL (1)	Digital I/O	I2C: clock output (master) or clock input (slave).
CT12	Analog Input	Cap-sensing touch input channel 12.
P3.3	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
URX1	Digital Input	UART1: reception pin.
SDA (1)	Digital I/O	I2C: data pin.
CT11	Analog Input	Cap-sensing touch input channel 11.

4.5 Pin Characteristic

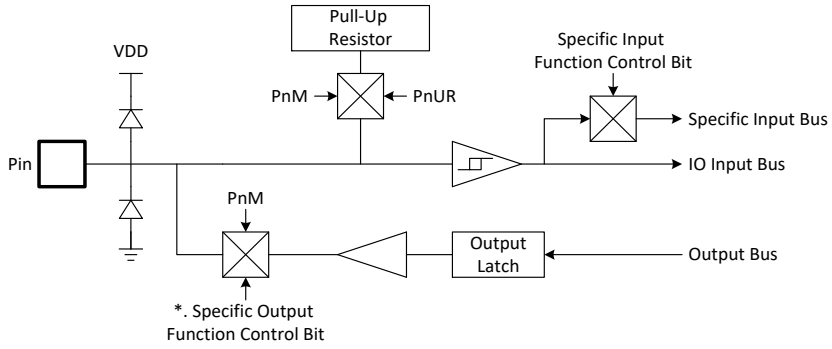
Port	Open-Drain	1/2*VDD Voltage	Sink Current 100mA VSS+1.5V	External Interrupt	Wakeup (Level change)	Shared Pin
P0.0	-	V	V	-	V	AIN15/XIN/COM0/PWM10
P0.1	-	V	V	-	V	AIN14/XOUT/COM1/PWM11
P0.2	V	V	V	-	V	AIN13/COM2/CT18/URX
P0.3	V	V	V	-	V	AIN12/COM3/CT17/UTX
P0.4	-	-	-	-	V	AIN11/CT16
P0.5	-	-	-	V	V	AIN10/CSOUT/INT0/CT15
P0.6	-	-	-	V	V	CLKOUT/INT1/CT10
P0.7	-	-	-	V	V	INT2/CT9/PWM12
P1.0	-	V	V	-	V	COM4/CT19
P1.1	-	V	V	-	V	AIN0/AVREFH/COM5/CT22
P1.2	V	V	V	-	V	SWAT/AIN1/COM6/CT21/URX(1)/UTX(1)
P1.3	-	V	V	-	V	AIN2/COM7/CT20
P1.4	-	-	-	-	V	AIN3/BUZZER/CT0
P1.5	-	-	-	-	V	AIN4/CT1/SCL
P1.6	-	-	-	-	V	AIN5/CT2/SDA
P1.7	-	-	-	-	V	AIN6/CT3
P2.0	V	-	-	-	-	RST/AIN7/CT4/UTX1(1)/PWM20
P2.1	V	-	-	-	-	AIN8/CT5/URX1(1)/PWM21
P2.2	-	-	-	-	-	AIN9/CT6/PWM22
P2.3	-	-	-	-	-	CT7/PWM23
P2.4	-	-	-	-	-	CT8/PWM13
P3.0	-	-	-	-	-	CT14
P3.1	-	-	-	-	-	CT13
P3.2	V	-	-	-	-	CT12/SCL(1)/UTX1
P3.3	V	-	-	-	-	CT11/SDA(1)/URX1
CTCAP	-	-	-	-	-	

4.6 Pin Circuit Diagrams

Normal Bi-direction I/O Pin.

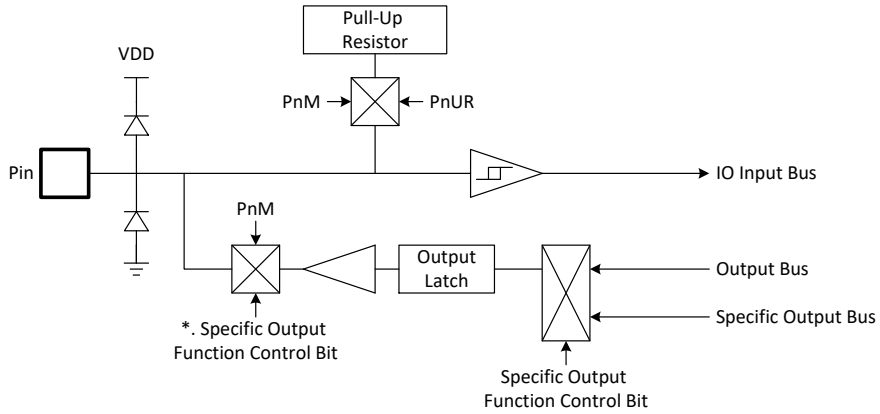


Bi-direction I/O Pin Shared with Specific Digital Input Function, e.g. INT2.



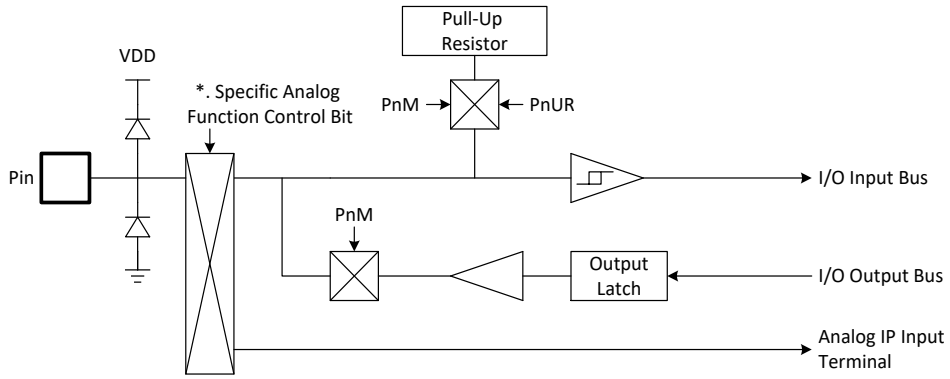
***. Some specific functions switch I/O direction directly, not through PnM register.**

Bi-direction I/O Pin Shared with Specific Digital Output Function, e.g. PWM, UART.



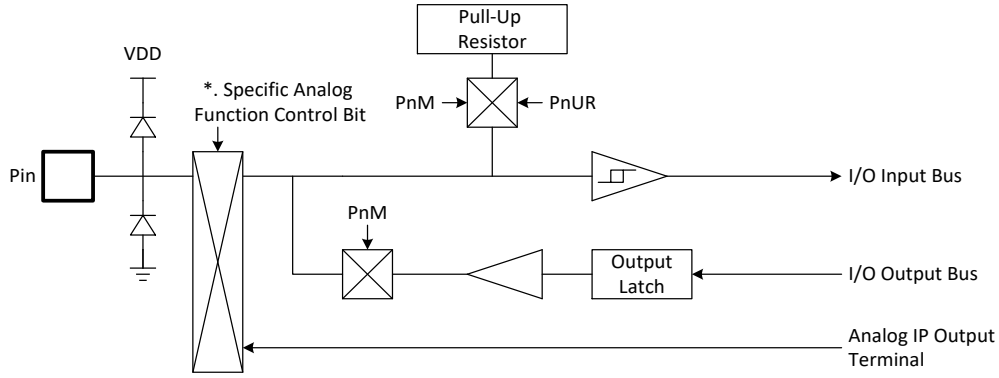
***. Some specific functions switch I/O direction directly, not through PnM register.**

Bi-direction I/O Pin Shared with Specific Analog Input Function, e.g. XIN, ADC.



***. Some specific functions switch I/O direction directly, not through PnM register.**

Bi-direction I/O Pin Shared with Specific Analog Output Function, e.g. XOUT…



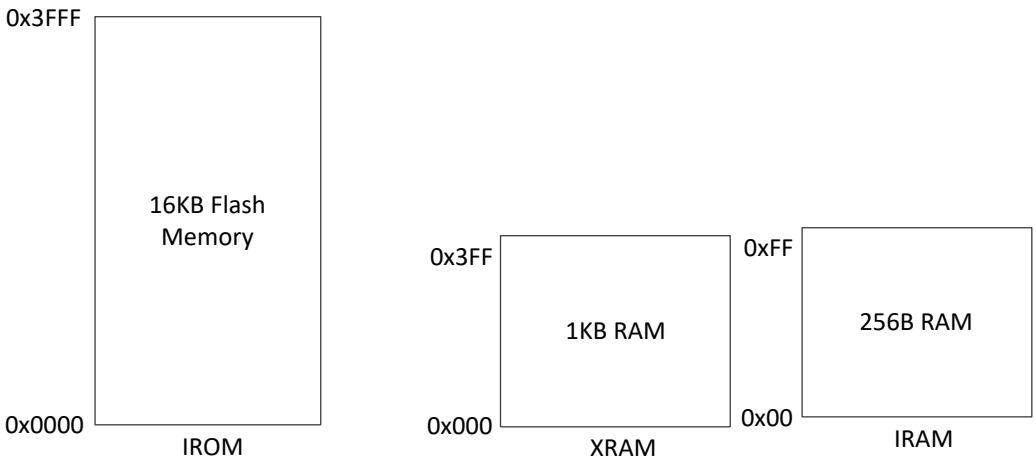
***. Some specific functions switch I/O direction directly, not through PnM register.**

5 CPU

SN8F5000 family is an enhanced 8051 microcontroller (MCU). It is fully compatible with MCS-51 instructions, hence the ability to cooperate with modern development environment (e.g. Keil C51). SN8F5000 CPU has 9.4 to 12.1 times faster than the original 8051 at the same frequency.

5.1 Memory Organization

SN8F5754 builds in three on-chip memories: internal RAM (IRAM), external RAM (XRAM), program memory (IROM) and information block memory (IBROM). The internal RAM is a 256-byte RAM which has higher access performance (direct and indirect addressing). By contrast, the external RAM has 1024-byte of size, but it requires a longer access period. The program memory is a 16 KB non-volatile memory and has a maximum 8 MHz speed limitation.

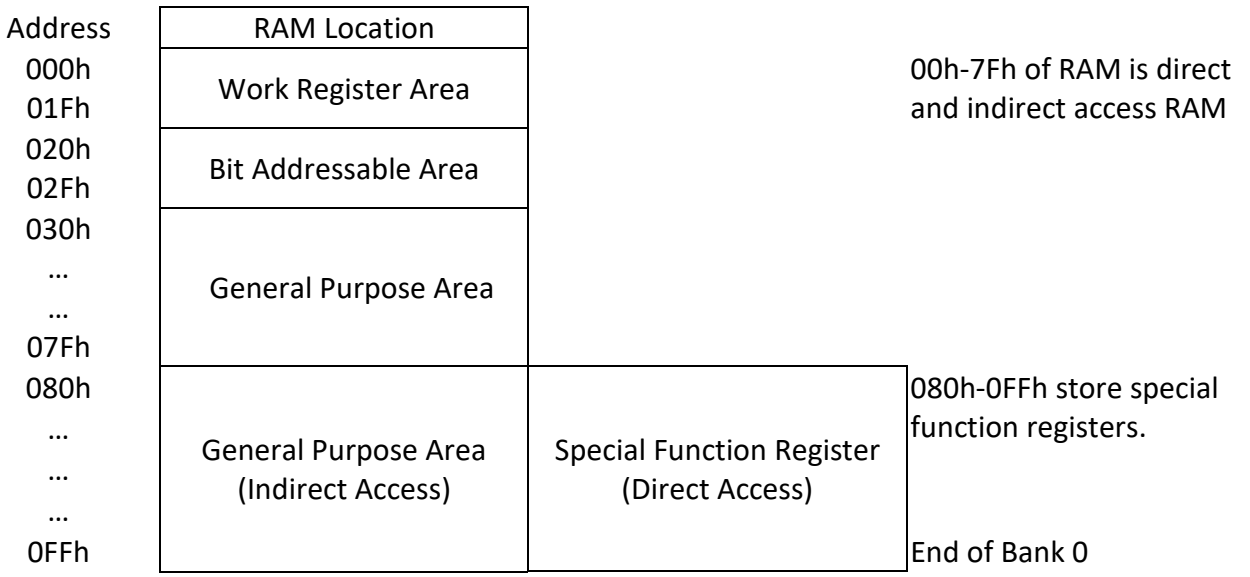


✷ ***Note:***

1. IBROM is read-only and can't be programmed by user

5.2 Internal RAM (IRAM)

256 X 8-bit RAM (Internal Data Memory)



The 256-byte data RAM in internal data memory is a standard 8051 RAM access configuration. The upper 128-byte RAM is general purpose RAM and can configure by direct addressing access and indirect addressing access. The lower 128-byte can be indirect access RAM in general purpose or direct access RAM in special function register (SFR).

- 0x0000-0x007F: General purpose RAM contains work register area and bit addressable area. In this area, direct or indirect addressing can be used.
- 0x0000-0x001F: Work register area includes 4-bank. Each bank has 8 work registers (R0 - R7) which is selected by RS0/RS1 in PSW register.
- 0x0020-0x002F: Bit addressable area.

In the bit addressable area, user can read or write any single bit in this range by using the unique address for that bit. Supports 16bytes bit addressable RAM area giving 128 addressable bits. Each bit has individual address in the range from 00H to 7FH. Thus, the bit can be addressed directly. Bit0 of the byte 20H has bit address 00H and Bit 7 of the byte 20H has bit address 07H. Bit0 of the byte 2FH has bit address 78H and Bit 7 of the byte 2FH has bit address 7FH. When set “SETB 42H”, it means the bit2 of the byte 28H is set.

	Byte Address	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
Bit Addressable Area	**0x20**	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
	0x21	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
	0x22	0x10	0x11	0x12	0x13	0x14	0x15	0x16	0x17

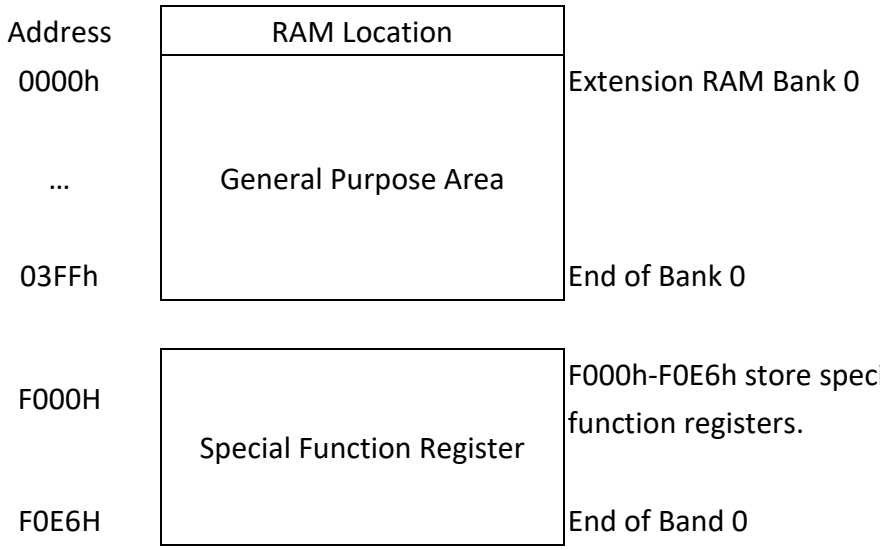
	0x23	0x18	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F
	0x24	0x20	0x21	0x22	0x23	0x24	0x25	0x26	0x27
	0x25	0x28	0x29	0x2A	0x2B	0x2C	0x2D	0x2E	0x2F
	0x26	0x30	0x31	0x32	0x33	0x34	0x35	0x36	0x37
	0x27	0x38	0x39	0x3A	0x3B	0x3C	0x3D	0x3E	0x3F
	0x28	0x40	0x41	0x42	0x43	0x44	0x45	0x46	0x47
	0x29	0x48	0x49	0x4A	0x4B	0x4C	0x4D	0x4E	0x4F
	0x2A	0x50	0x51	0x52	0x53	0x54	0x55	0x56	0x57
	0x2B	0x58	0x59	0x5A	0x5B	0x5C	0x5D	0x5E	0x5F
	0x2C	0x60	0x61	0x62	0x63	0x64	0x65	0x66	0x67
	0x2D	0x68	0x69	0x6A	0x6B	0x6C	0x6D	0x6E	0x6F
	0x2E	0x70	0x71	0x72	0x73	0x74	0x75	0x76	0x77
	0x2F	0x78	0x79	0x7A	0x7B	0x7C	0x7D	0x7E	0x7F

- 0x0080~0x00FF: General purpose area in indirect addressing access or special function register in direct addressing access.

5.3 External RAM (XRAM)

1024 X 8-bit XRAM (Extension Data Memory)

The external RAM enlarges the capacity of variables; it is the lowest access performance in the contrast of internal RAM. Since frequently used variables and local variables are expected to store in internal RAM, the vast majority of external RAM usages are specific. It can be allocated as a variable storage area for lower priority tasks, or look-up table preloaded from ROM to speed up the access period.

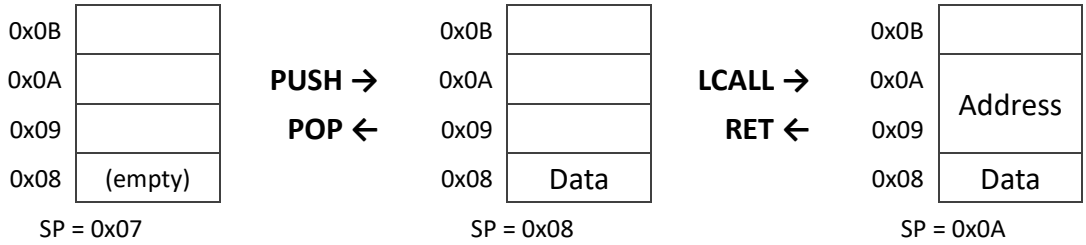


The upper 1024-byte XRAM is general purpose RAM and can configure by MOVX instruction access.

The lower 231 bytes of XRAM are special function registers (SFRs) that can also be accessed by MOVX instruction.

5.4 Stack

Stack can be assigned to any area of internal RAM (IRAM). However, it requires manual assignment to ensure its area does not overlap other RAM's variables. An overflow or underflow stack could also mistakenly overwrite other RAM's variables; thus, these factors should be considered while arrange the size of stack.



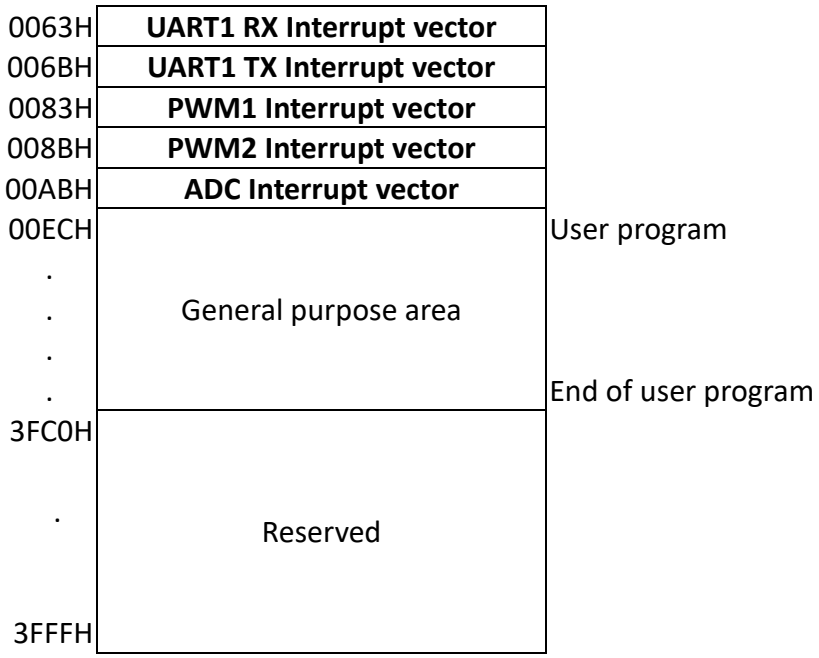
By default, stack pointer (SP register) points to 0x07 which means the stack area begin at IRAM address 0x08. In other word, if a planned stack area is assigned from IRAM address 0xC0, the appropriate SP register is anticipated to set at 0xBF after system reset.

An assembly PUSH instruction costs one byte of stack. LCALL, ACALL instructions and interrupt respectively costs two bytes stack. POP-instruction decreases one count, and a RET/RETI subtract two counts of stack pointer.

5.5 Program Memory (IROM)

The program memory is a non-volatile storage area where stores code, look-up ROM table, and other data with occasional modification. It can be updated by debug tools like SN-Link3, and a program can also self-update via in-system program process (refer to In-system Program).

Address	**ROM**	Comment
0000H	Reset vector	Reset vector
0001H 0002H	General purpose area	User program
0003H	**INT0 Interrupt vector**	**Interrupt vector**
000BH	**TIMER0 Interrupt vector**	
0013H	**INT1 Interrupt vector**	
001BH	**TIMER1 Interrupt vector**	
0023H	**INT2 Interrupt vector**	
0033H	**TIMER3 Interrupt vector**	
0043H	**I2C Interrupt vector**	
0053H	**UART0 RX Interrupt vector**	
005BH	**UART0 TX Interrupt vector**	



The ROM includes reset vector, Interrupt vector, general purpose area and reserved area. The reset vector is program beginning address. The interrupt vector is the head of interrupt service routine when any interrupt occurring. The general purpose area is main program area including main loop, sub-routines and data table.

- 0x0000 Reset vector: Program counter points to 0x0000 after any reset events (power on reset, reset pin reset, watchdog reset, LVD reset…).
- 0x0001~0x0002: General purpose area to process system reset operation.
- 0x0003~0x00EB: Multi interrupt vector area. Each of interrupt events has a unique interrupt vector.
- 0x00EC~0x3FBF: General purpose area for user program and ISP (EEPROM function).
- 0x3FC0~0x3FF6: General purpose area for user program. Do not execute ISP.
- 0x3FF6~0x3FFF: Reserved area. Do not execute ISP.

5.6 Program Memory Security

The SN8F5754 provides security options at the disposal of the designer to prevent unauthorized access to information stored in FLASH memory. When enable security option, the ROM code is secured and not dumped complete ROM contents. ROM security rule is all address ROM data protected and outputs 0x00.

5.7 Data Pointer

A data pointer helps to specify the XRAM and IROM address while performing MOVX and MOVC instructions. The microcontroller has two set of data pointer (DPH/DPL and DPH1/DPL1) which is selectable by DPS register. The DPC register controls two functions: next DPTR selection and

automatically increase/decrease DPTR function.

The next DPTR selection can specify which DPTR is anticipated to use after perform MOVX @DPTR instruction. In other word, the DPS can automatically swap between the two data pointers. To enable this function: write 0 to DPSEL and fill 1 to NDPS firstly, then write 1 to DPSEL and fill 0 to NDPS register.

The automatically increase/decrease DPTR function can make an increment or decrement after perform MOVX @DPTR instruction. As a result, it enables a continuous external RAM access without re-specified DPTR value. Those functions are controlled by the DPC Register, where there are separate DPC register bits for each DPTR, to provide high flexibility in data transfers. The DPC Register address 0x93 points to the window where the actual DPC is selected using the DPS Register, same as for the DPTR.

5.8 Stack and Data Pointer Register

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SP	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
DPL	DPL7	DPL6	DPL5	DPL4	DPL3	DPL2	DPL1	DPL0
DPH	DPH7	DPH6	DPH5	DPH4	DPH3	DPH2	DPH1	DPH0
DPL1	DPL17	DPL16	DPL15	DPL14	DPL13	DPL12	DPL11	DPL10
DPH1	DPH17	DPH16	DPH15	DPH14	DPH13	DPH12	DPH11	DPH10
DPS	-	-	-	-	-	-	-	DPSEL
DPC	-	-	-	-	NDPS	ATMS	ATMD	ATME

SP Register (0x81)

Bit	Field	Type	Initial	Description
7..0	SP	R/W	0x07	Stack pointer

DPL Register (0x82)

Bit	Field	Type	Initial	Description
7..0	DPL[7:0]	R/W	0x00	Low byte of DPTR0

DPH Register (0x83)

Bit	Field	Type	Initial	Description
7..0	DPH[7:0]	R/W	0x00	High byte of DPTR0

DPL1 Register (0x84)

Bit	Field	Type	Initial	Description

7..0	DPL1[7:0]	R/W	0x00	Low byte of DPTR1

DPH1 Register (0x85)

Bit	Field	Type	Initial	Description
7..0	DPH1[7:0]	R/W	0x00	High byte of DPTR1

DPS Register (0x92)

Bit	Field	Type	Initial	Description
7..1	Reserved	R	0x00	
0	DPSEL	R/W	0	DPTR selection 0: DPH/DPL (DPTR0) is selected 1: DPH1/DPL1 (DPTR1) is selected

DPC Register (0x93)

Bit	Field	Type	Initial	Description
7..4	Reserved	R	0x0	
3	NDPS	R/W	0	Next DPTR selection The DPSEL loads this bit automatically after perform any MOVX @DPTR instruction.
2..1	ATMS/ATMD	R/W	00	Automatically increase/decrease DPTR (if ATME applied) 00: +1 after any MOVX @DPTR instruction 01: -1 after any MOVX @DPTR instruction 10: +2 after any MOVX @DPTR instruction 11: -2 after any MOVX @DPTR instruction
0	ATME	R/W	0	Automatically increase/decrease DPTR function 0: Disable 1: Enable

6 Special Function Registers

The SN8F5754 offers many functions that can be controlled via special function registers. Special function registers are placed at both the IRAM address and the XRAM address. The IRAM registers can be accessed by direct addressing, while the XRAM registers are accessed via the MOVX instruction. Register description will be classified according to the addressing range. The 8-bit address refers to the IRAM register and the 16-bit address refers to the XRAM register.

6.1 IRAM Special Function Register Memory Map

BIN / HEX	000	001	010	011	100	101	110	111
F8		P0M	P1M	P2M	P3M	-	-	PFLAG
F0	B	CRCM	CRCDATL	CRCDATH	SYSMOD	-	-	SRST
E8	-	-	-	-	-	-	-	-
E0	ACC	-	-	-	-	-	-	-
D8	-	BZM	I2CDAT	I2CADR	I2CCON	I2CSTA	SMBSEL	SMBDST
D0	PSW	ADT	ADM	ADB	ADR	VREFH	BIASCTL	COMCTL
C8	IEN2	PW1M	PW1CH	PW2M	PW2CH	-	-	-
C0	-	-	-	CSCON	CSCON4	CSCAVGL	CSCAVGH	-
B8	IEN1	IP1	PECMD	PEROML	PEROMH	PERAM	-	DEGCMD
B0	P3	T3M	T3CL	T3CH	T3RL	T3RH	-	-
A8	IEN0	IP0	FRQL	FRQH	FRQCMD	LFRQL	LFRQH	CLKCAL
A0	P2	-	-	-	-	IRCON	IRCON2	IRCON3
98	IEN3	S0BUF	S0RELL	S0RELH	S1CON	S1BUF	S1RELL	S1RELH
90	P1	-	DPS	DPC	SMODMX	SMODCAL	S0CON2	S0CON
88	TCON	TMOD	TL0	TL1	TH0	TH1	TCON0	PEDGE
80	P0	SP	DPL	DPH	DPL1	DPH1-	WDTR	PCON

✸ ***Note: All SFRs in the left-most column are bit-addressable. (Every 0x0/0x8-ending SFR addresses are bit-addressable).***

6.2 IRAM Special Function Register Description

0x80 - 0x9F Registers Description

Register	Address	Description
P0	0x80	Port 0 data buffer.
SP	0x81	Stack pointer register.
DPL	0x82	Data pointer 0 low byte register.
DPH	0x83	Data pointer 0 high byte register.
DPL1	0x84	Data pointer 1 low byte register.
DPH1	0x85	Data pointer 1 high byte register.
WDTR	0x86	Watchdog timer clear register.
PCON	0x87	System mode register.
TCON	0x88	Timer 0 / 1 controls register.
TMOD	0x89	Timer 0 / 1 mode register.
TL0	0x8A	Timer 0 counting low byte register.
TL1	0x8B	Timer 1 counting low byte register.
TH0	0x8C	Timer 0 counting high byte register.
TH1	0x8D	Timer 1 counting high byte register.
TCON0	0x8E	Timer 0 / 1 clock controls register.
PEDGE	0x8F	External interrupt edge controls register.
P1	0x90	Port 1 data buffer.
-	0x91	-
DPS	0x92	Data pointer selects register.
DPC	0x93	Data pointer controls register.
SMODMX	0x94	UART pins selection matrix.
SMODCAL	0x95	UARTs baud rate controls register.
S0CON2	0x96	UARTs baud rate control s register.
S0CON	0x97	UART0 control register.
IEN3	0x98	Interrupts enable register
S0BUF	0x99	UART0 data buffer.
S0RELL	0x9A	UART0 reload low byte register.
S0RELH	0x9B	UART0 reload high byte register.
S1CON	0x9C	UART1 control register.
S1BUF	0x9D	UART1 data buffer.
S1RELL	0x9E	UART1 reload low byte register.
S1RELH	0x9F	UART1 reload high byte register.

0xA0 - 0xBF Registers Description

Register	Address	Description
P2	0xA0	Port2 data buffer.
-	0xA1	-
-	0xA2	-
-	0xA3	-
-	0xA4	-
IRCON	0xA5	Interrupts request register.
IRCON2	0xA6	Interrupts request register.
IRCON3	0xA7	Interrupts request register.
IEN0	0xA8	Interrupts enable register
IP0	0xA9	Interrupts priority register.
FRQL	0xAA	Clock fine tuning controls buffer low byte.
FRQH	0xAB	Clock fine tuning controls buffer high byte.
FRQCMD	0xAC	Clock fine tuning command register.
LPRQL	0xAD	Low clock fine tuning controls buffer low byte.
LPRQH	0xAE	Low clock fine tuning controls buffer high byte.
CLKCAL	0xAF	ILRC auto-calibration register.
P3	0xB0	Port 3 data buffer.
T3M	0xB1	Timer 3 controls register.
T3CL	0xB2	Timer 3 counting low byte register.
T3CH	0xB3	Timer 3 counting high byte register.
T3RL	0xB4	Timer 3 cycle controls buffer low byte.
T3RH	0xB5	Timer 3 cycle controls buffer high byte
-	0xB6	-
-	0xB7	-
IEN1	0xB8	Interrupts enable register.
IP1	0xB9	Interrupts priority register.
PECMD	0xBA	In-System Program command register.
PEROML	0xBB	In-System Program ROM address low byte.
PEROMH	0xBC	In-System Program ROM address high byte.
PERAM	0xBD	In-System Program RAM mapping address.
-	0xBE	-
DEGCMD	0xBF	Debug mode switch control register.

0xC0 - 0xDF Registers Description

Register	Address	Description
-	0xC0	-
-	0xC1	-
-	0xC2	-
-	0xC3	-
-	0xC4	-
-	0xC5	-
-	0xC6	-
-	0xC7	-
IEN2	0xC8	Interrupts enable register.
PW1M	0xC9	PWM1 controls register.
PW1CH	0xCA	PWM1 channels control register.
PW2M	0xCB	PWM2 controls register.
PW2CH	0xCC	PWM2 channels control register.
-	0xCD	-
-	0xCE	-
-	0xCF	-
PSW	0xD0	System flag register.
ADT	0xD1	ADC offset calibration control register.
ADM	0xD2	ADC controls register.
ADB	0xD3	ADC data buffer.
ADR	0xD4	ADC resolution selects register.
VREFH	0xD5	ADC reference voltage controls register.
BIASCTL	0xD6	Half-VDD control register.
COMCTL	0xD7	Half-VDD control register.
-	0xD8	-
BZM	0xD9	Buzzer controls register.
I2CDAT	0xDA	I2C data buffer.
I2CADR	0xDB	Own I2C slave address.
I2CCON	0xDC	I2C interface operation control register.
I2CSTA	0xDD	I2C Status Code.
SMBSEL	0xDE	SMBus mode controls register.
SMBDST	0xDF	SMBus internal timeout register.

0xE0 - 0xFF Registers Description

Register	Address	Description
ACC	0xE0	Accumulator register.
-	0xE1	-
-	0xE2	-
-	0xE3	-
-	0xE4	-
-	0xE5	-
-	0xE6	-
-	0xE7	-
-	0xE8	-
-	0xE9	-
-	0xEA	-
-	0xEB	-
-	0xEC	-
-	0xED	-
-	0xEE	-
-	0xEF	-
B	0xF0	Multiplication/ division instruction data buffer.
CRCM	0xF1	CRC controls register.
CRCDATL	0xF2	CRC data low byte register.
CRCDATH	0xF3	CRC data high byte register.
SYSMOD	0xF4	System controls register.
-	0xF5	-
-	0xF6	-
SRST	0xF7	Software reset controls register.
-	0xF8	-
P0M	0xF9	Port 0 input/output mode register.
P1M	0xFA	Port 1 input/output mode register.
P2M	0xFB	Port 2 input/output mode register.
P3M	0xFC	Port 3 input/output mode register.
-	0xFD	-
-	0xFE	-
PFLAG	0xFF	Reset flag register.

6.3 System Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ACC	ACC7	ACC6	ACC5	ACC4	ACC3	ACC2	ACC1	ACC0
B	B7	B6	B5	B4	B3	B2	B1	B0
PSW	CY	AC	F0	RS1	RS0	OV	F1	P

ACC Register (0xE0)

Bit	Field	Type	Initial	Description
7..0	ACC[7:0]	R/W	0x00	The ACC is an 8-bit data register responsible for transferring or manipulating data between ALU and data memory. If the result of operating is overflow (OV) or there is carry (C or AC) and parity (P) occurrence, then these flags will be set to PSW register.

B Register (0xF0)

Bit	Field	Type	Initial	Description
7..0	B[7:0]	R/W	0x00	The B register is used during multiplying and division instructions. It can also be used as a scratch-pad register to hold temporary data.

PSW Register (0xD0)

Bit	Field	Type	Initial	Description
7	CY	R/W	0	Carry flag. 0: Addition without carry, subtraction with borrowing signal, rotation with shifting out logic “0”, comparison result < 0. 1: Addition with carry, subtraction without borrowing, rotation with shifting out logic “1”, comparison result ≥ 0.
6	AC	R/W	0	Auxiliary carry flag. 0: If there is no a carry-out from 3rd bit of Accumulator in BCD operations. 1: If there is a carry-out from 3rd bit of Accumulator in BCD operations.
5	F0	R/W	0	General purpose flag 0. General purpose flag available for user.

4..3	RS[1:0]	R/W	00	Register bank select control bit, used to select working register bank. 00: 00H – 07H (Bnak0) 01: 08H – 0FH (Bnak1) 10: 10H – 17H (Bnak2) 11: 18H – 1FH (Bnak3)
2	OV	R/W	0	Overflow flag. 0: Non-overflow in Accumulator during arithmetic Operations. 1: overflow in Accumulator during arithmetic Operations.
1	F1	R/W	0	General purpose flag 1. General purpose flag available for user.
0	P	R	0	Parity flag. Reflects the number of ‘1’s in the Accumulator. 0: if Accumulator contains an even number of ‘1’s. 1: Accumulator contains an odd number of ‘1’s.

6.4 XRAM Special Function Register Memory Map

BIN / HEX	000	001	010	011	100	101	110	111
F088	PW20DH	PW20DL	PW21DH	PW21DL	PW22DH	PW22DL	PW23DH	PW23DL
F080								
F078	PW10DH	PW10DL	PW11DH	PW11DL	PW12DH	PW12DL	PW13DH	PW13DL
F070	PW1YH	PW1YL	PW2YH	PW2YL	-	-	-	-
F068	-	-	-	-	-	-	-	-
F060	-	-	-	-	-	-	-	-
F058	-	-	-	-	-	-	-	-
F050	-	-	-	-	-	-	-	-
F048	-	-	-	-	-	-	-	-
F040	-	-	-	-	-	-	-	-
F038	-	-	-	-	-	-	-	-

F030	-	-	-	-	-	-	-	-
F028	-	-	-	-	-	-	-	-
F020	-	P1OC	-	P3OC	-	-	-	-
F018	P0CON	P1CON	P2CON	P3CON	-	-	-	-
F010	-	P1W	-	-	-	-	-	-
F008	-	-	-	-	-	-	-	-
F000	P0UR	P1UR	P2UR	P3UR	-	-	-	-

✷ ***Note: the XRAM registers are accessed via the MOVX instruction.***

6.5 XRAM Special Function Register Description

0xF000 - 0xF01F Registers Description

Register	Address	Description
P0UR	0xF000	Port0 pull-up controls register.
P1UR	0xF001	Port1 pull-up controls register.
P2UR	0xF002	Port2 pull-up controls register.
P3UR	0xF003	Port3 pull-up controls register.
-	0xF004	-
-	0xF005	-
-	0xF006	-
-	0xF007	-
-	0xF008	-
-	0xF009	-
-	0xF00A	-
-	0xF00B	-
-	0xF00C	-
-	0xF00D	-
-	0xF00E	-
-	0xF00F	-
-	0xF010	-
P1W	0xF011	Port 1 wake-up controls register.
-	0xF012	-
-	0xF013	-
-	0xF014	-
-	0xF015	-
-	0xF016	-
-	0xF017	-
P0CON	0xF018	Port 0 configuration controls register.
P1CON	0xF019	Port 1 configuration controls register.
P2CON	0xF01A	Port 2 configuration controls register.
P3CON	0xF01B	Port 3 configuration controls register.
-	0xF01C	-
-	0xF01D	-
-	0xF01E	-
-	0xF01F	-

0xF020 - 0xF03F Registers Description

Register	Address	Description
P0OC	0xF020	Open drain controls register.
P1OC	0xF021	Open drain controls register.
P2OC	0xF022	Open drain controls register.
P3OC	0xF023	Open drain controls register.
-	0xF024	-
-	0xF025	-
-	0xF026	-
-	0xF027	-
-	0xF028	-
-	0xF029	-
-	0xF02A	-
-	0xF02B	-
-	0xF02C	-
-	0xF02D	-
-	0xF02E	-
-	0xF02F	-
-	0xF030	-
-	0xF031	-
-	0xF032	-
-	0xF033	-
-	0xF034	-
-	0xF035	-
-	0xF036	-
-	0xF037	-
-	0xF038	-
-	0xF039	-
-	0xF03A	-
-	0xF03B	-
-	0xF03C	-
-	0xF03D	-
-	0xF03E	-
-	0xF03F	-

0xF060 - 0xF07F Registers Description

Register	Address	Description
-	0xF060	-
-	0xF061	-
-	0xF062	-
-	0xF063	-
-	0xF064	-
-	0xF065	-
-	0xF066	-
-	0xF067	-
-	0xF068	-
-	0xF069	-
-	0xF06A	-
-	0xF06B	-
-	0xF06C	-
-	0xF06D	-
-	0xF06E	-
-	0xF06F	-
PW1YL	0xF070	PW1 cycle controls buffer low byte.
PW1YH	0xF071	PW1 cycle controls buffer high byte.
PW2YL	0xF072	PW2 cycle controls buffer low byte.
PW2YH	0xF073	PW2 cycle controls buffer high byte.
-	0xF074	-
-	0xF075	-
-	0xF076	-
-	0xF077	-
PW10DL	0xF078	PW10 duty controls buffer low byte.
PW10DH	0xF079	PW10 duty controls buffer high byte.
PW11DL	0xF07A	PW11 duty controls buffer low byte.
PW11DH	0xF07B	PW11 duty controls buffer high byte.
PW12DL	0xF07C	PW12 duty controls buffer low byte.
PW12DH	0xF07D	PW12 duty controls buffer high byte.
PW13DL	0xF07E	PW13 duty controls buffer low byte.
PW13DH	0xF07F	PW13 duty controls buffer high byte.

0xF080 - 0xF09F Registers Description

Register	Address	Description
-	0xF080	-
-	0xF081	-
-	0xF082	-
-	0xF083	-
-	0xF084	-
-	0xF085	-
-	0xF086	-
-	0xF087	-
PW20DL	0xF088	PW20 duty controls buffer low byte.
PW20DH	0xF089	PW20 duty controls buffer high byte.
PW21DL	0xF08A	PW21 duty controls buffer low byte.
PW21DH	0xF08B	PW21 duty controls buffer high byte.
PW22DL	0xF08C	PW22 duty controls buffer low byte.
PW22DH	0xF08D	PW22 duty controls buffer high byte.
PW23DL	0xF08E	PW23 duty controls buffer low byte.
PW23DH	0xF08F	PW23 duty controls buffer high byte.
-	0xF090	-
-	0xF091	-
-	0xF092	-
-	0xF093	-
-	0xF094	-
-	0xF095	-
-	0xF096	-
-	0xF097	-
-	0xF098	-
-	0xF099	-
-	0xF09A	-
-	0xF09B	-
-	0xF09C	-
-	0xF09D	-
-	0xF09E	-
-	0xF09F	-

6.6 Register Declaration

SN8F5754 has many registers to control various functions, but SFR name is not predefined in the C51 / A51 compiler. To make programming easier and therefore need to add header files to declare SFR name.

When using the assembly code programs, please add the following sentence.

```
$NOMOD51 ; Do not recognize the 8051-specific predefined special register.
#define __XRAM_SFR_H__ ; For XRAM Register Declaration.
#include <SN8F5754.H>
```

When using the C code programs, please add the following sentence.

```
#define __XRAM_SFR_H__ // For XRAM Register Declaration.
#include <SN8F5754.H>
```

After adding the header file, user can use name of registers to program. During compilation, the compiler will register name translate into register position through the header file.

Different devices need to use a different header file to declare, but the option file is to use the same.

Device	Header file	Options file
SN8F5754	SN8F5754.h	OPTIONS_SN8F5754.A51
SN8F5753	SN8F5754.h	OPTIONS_SN8F5754.A51
SN8F5752	SN8F5754.h	OPTIONS_SN8F5754.A51

✷ ***Note:***

1. The XRAM registers are declared in the main program file and are available for all program files.

2. The XRAM registers are declared only in the main program to avoid duplicate declarations. Therefore, "# define __XRAM_SFR_H__" word string do not add to other program files, to avoid compilation errors.

3. Before including the header file, you must define the "__XRAM_SFR_H__" word string to avoid compilation errors.

6.7 XRAM Register Description

Although the XRAM register is accessed through the MOVX instruction, it is also possible to program code with the SFR name without referencing the address.

When using the assembly code program, you can refer to the following method.

```
MOV     DPTR, #P0UR
MOV     A, #08H
MOVX    @DPTR, A

MOV     DPTR, #P0UR
MOVX    A, @DPTR
MOV     TMP, A ; TMP is direct addressing variable
```

When using the C code program, you can refer to the following method.

```
P0UR = 0x08;
TMP = P0UR; // TMP is direct addressing variable
```

When using assembly code, there is another way to write code faster. The header file has two macros to provide quick and easy programming.

```
WRXSFR  XRAM register, direct addressing variable (or register, #data)
RDXSFR  direct addressing variable (or register), XRAM register
```

Example:

```
WRXSFR  P0UR, #08H
WRXSFR  P0UR, B
WRXSFR  PW2YH, TMP ; TMP is direct addressing variable

RDXSFR  B, POUR
RDXSFR  TMP, P0UR ; TMP is direct addressing variable
```

✸ ***Note:***

1. Using macro does not increase the speed of instructions, but simplifies the program code.

2. The macro is already included in the header file and can be used directly.

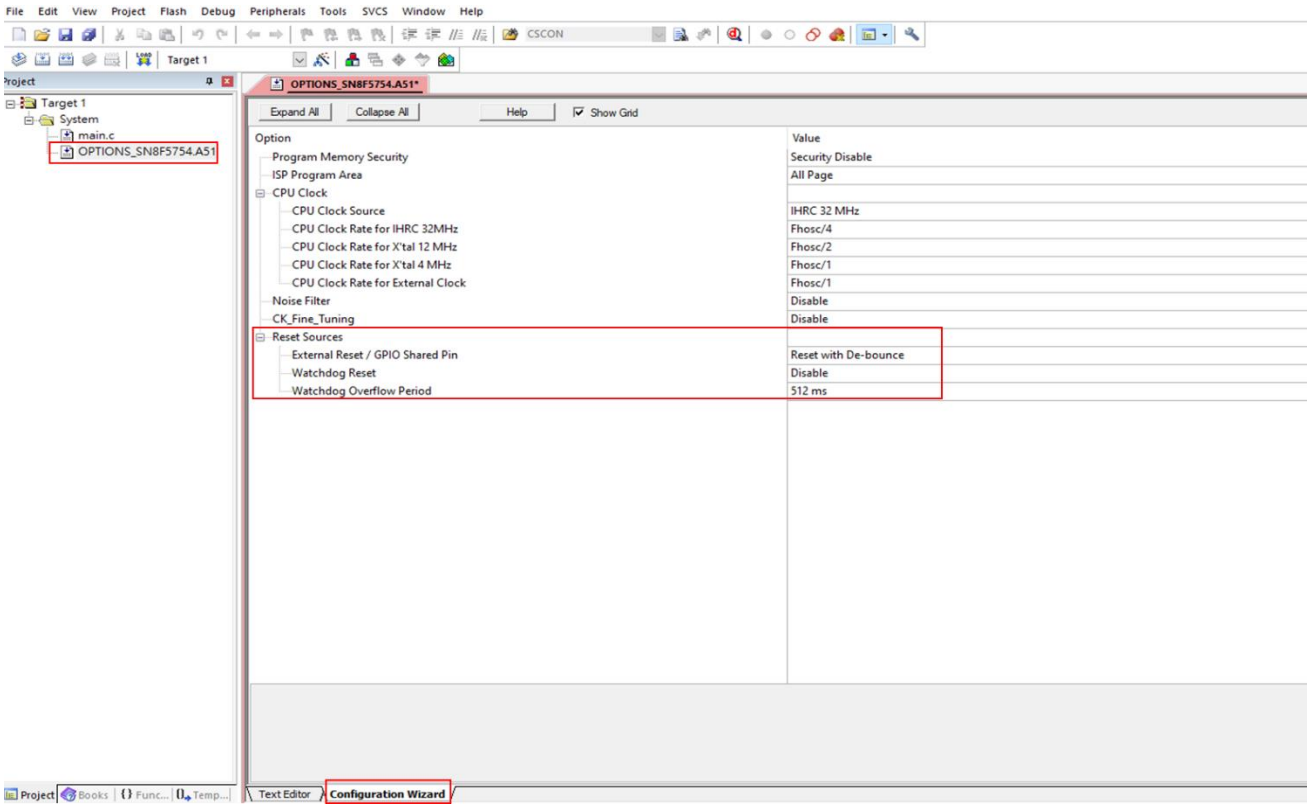
7 Reset and Power-on Controller

The reset and power-on controller has four reset sources: low voltage detectors (LVDs), watchdog, programmable external reset pin, and software reset. The first three sources would trigger an additional power-on sequence. Subsequently, the microcontroller initializes all registers and starts program execution with its reset vector (ROM address 0x0000).

7.1 Configuration of Reset and Power-on Controller

SONiX publishes a *SN8F5754_OPTIONS.A51 file* in *SN-Link Driver for Keil C51.exe* (downloadable on cooperative website: www.sonix.com.tw). This *options file* contains appropriate parameters of reset sources and CPU clock source selection, and is strongly recommended to add to Keil project. *SN8F5000 Debug Tool Manual* provides the further detail of this configuration. The option items are as following:

- Program Memory Security
- ISP Program Area
- CPU Clock
- Noise Filter
- CK_Fine_Tuning
- Reset Source : External Reset / GPIO Shared Pin
- Reset Source : Watchdog Reset & Overflow Period



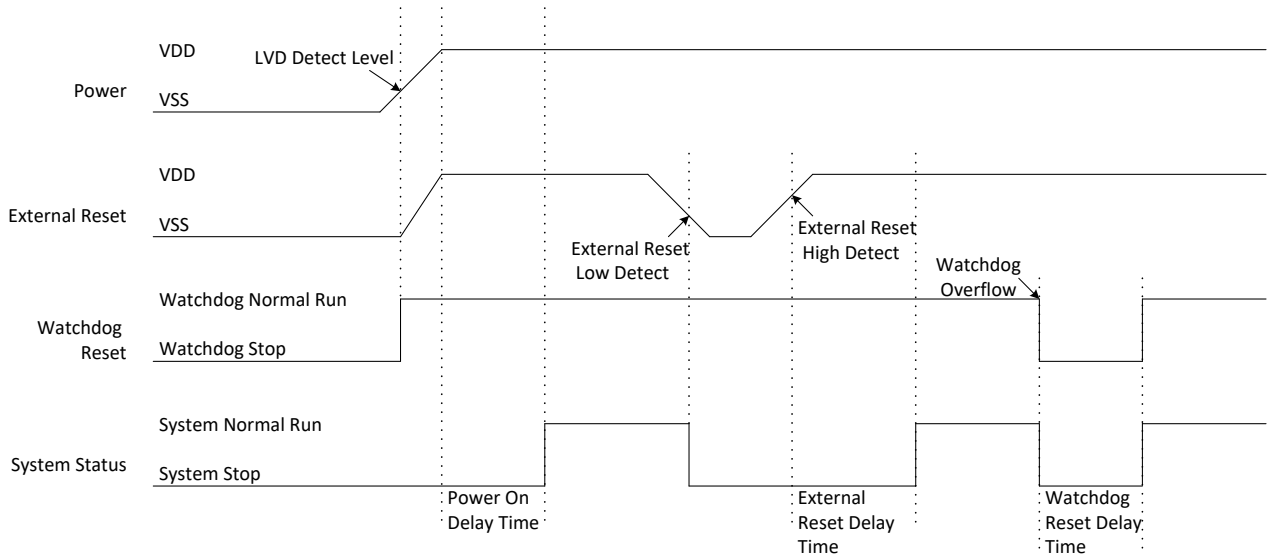
The code option is the system hardware configurations including oscillator type, noise filter option, watchdog timer operation, reset pin option and flash ROM security control. The code option items are as following table:

Code Option	Content	Function Description
Program Memory Security	Security Disable	Disable ROM code Security function
	Security Enable	Enable ROM code Security function
	Security Configuration	All address ROM data are protected expect address 0x3F00 ~ 0x3FDF, only address 0x3F00 ~ 0x3FDF ROM data can be accessed
ISP Program Area	All Page	All address can perform ISP function
	Page 504~ Page 510	Only address 0x3F00 ~ 0x3FDF can perform ISP function
CPU Clock Source	IHRC 32MHz	High speed internal 32MHz RC. XIN/XOUT pins are bi-direction GPIO mode
	X'tal 12MHz	High speed crystal /resonator (e.g. 12MHz) for external high clock oscillator
	X'tal 4MHz	Standard crystal /resonator (e.g. 4M) for external high clock oscillator
	External Clock	XIN pin connect external clock (1M ~32M), XOUT pin is bi-direction GPIO mode
	IHRC 32MHz + LX'tal	High speed internal 32MHz RC. XIN/XOUT pins for Low speed crystal /resonator (e.g. 32768Hz)
CPU Clock Rate	000 = fosc / 128 001 = fosc / 64 010 = fosc / 32 011 = fosc / 16 100 = fosc / 8 101 = fosc / 4 110 = fosc / 2 111 = fosc / 1	Fcpu clock rate
Noise Filter	Disable	Disable Noise Filter
	Enable	Enable Noise Filter
CK_Fine_Tuning	Disable	Disable CK_Fine_Tuning
	Enable	Enable CK_Fine_Tuning
External Reset	Reset with De-bounce	Enable External reset pin with De-bounce
	Reset without De-bounce	Enable External reset pin without De-bounce

	GPIO with P20	Enable P20
Watchdog Reset	Always	Watchdog timer is always on enable even in STOP mode and IDLE mode
	Enable	Enable watchdog timer. Watchdog timer stops in STOP mode and IDLE mode
	Disable	Disable Watchdog function
Watchdog Overflow Period	64ms	Watchdog timer clock source F_{ILRC} /1
	128ms	Watchdog timer clock source F_{ILRC} /2
	256ms	Watchdog timer clock source F_{ILRC} /4
	512ms	Watchdog timer clock source F_{ILRC} /8

7.2 Power-on Sequence

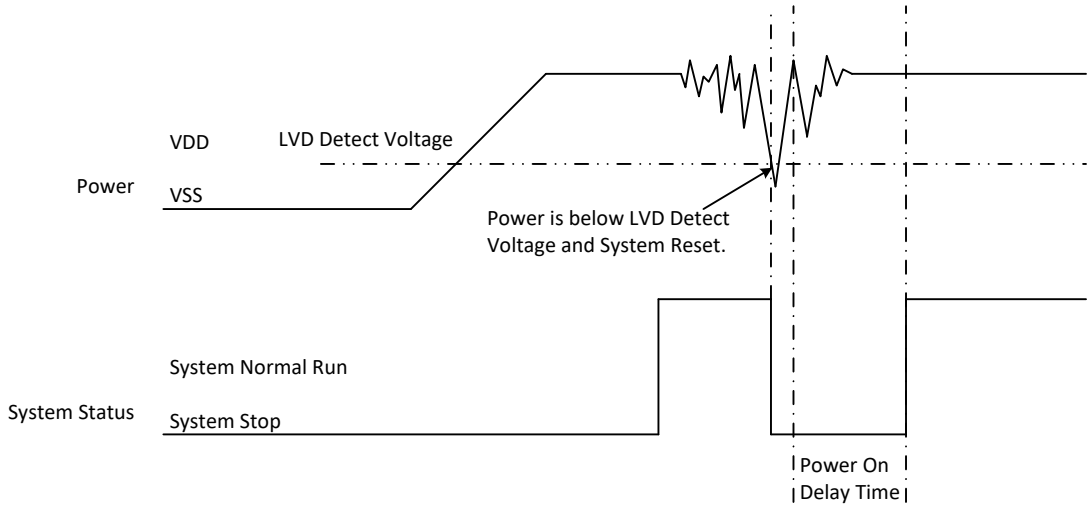
A power-on sequence would be triggered by LVD, watchdog, and external reset pin. It takes place between the end of reset signal and program execution. Overall, it includes two stages: power stabilization period, and clock stabilization period.



The power stabilization period spends 4.6 ms in typical condition. Afterward the microcontroller fetches CPU Clock Source selection automatically. The selected clock source would be driven, and the system counts 2048 times of the clock period and 5 times of the internal low-speed oscillator clocks to ensure its reliability.

7.3 LVD Reset

The low voltage detectors monitor VDD pin's voltage at only one level: 1.8 V. Depend on low voltage detection configuration, the comparison result can be seen as a system reset signal. The table below lists low voltage detection configuration, LVD_L, and the results of VDD pin's condition.



Condition	LVD_L
VDD ≤ 1.8 V	Reset

7.4 Watchdog Reset

Watchdog is a periodic reset signal generator for the purpose of monitoring the execution flow. Its internal timer is expected to be cleared in a check point of program flow; therefore, the actual reset signal would be generated only after a software problem occurs. Writing 0x5A to WDTR is the proper method to place a check point in program.

```
1    WDTR = 0x5A;
```

Watchdog timer interval time = 1024 * 1/ (Internal Low-Speed oscillator frequency/WDT Pre-scalar)

= 1024 / (F_{ILRC} /WDT Pre-scaler) …sec

Internal low-speed oscillator	WDT pre-scaler	Watchdog interval time
F_{ILRC} =16 kHz	F_{ILRC} /1	1024/(16000/1)=64ms
	F_{ILRC} /2	1024/(16000/2)=128ms
	F_{ILRC} /4	1024/(16000/4)=256ms
	F_{ILRC} /8	1024/(16000/8)=512ms

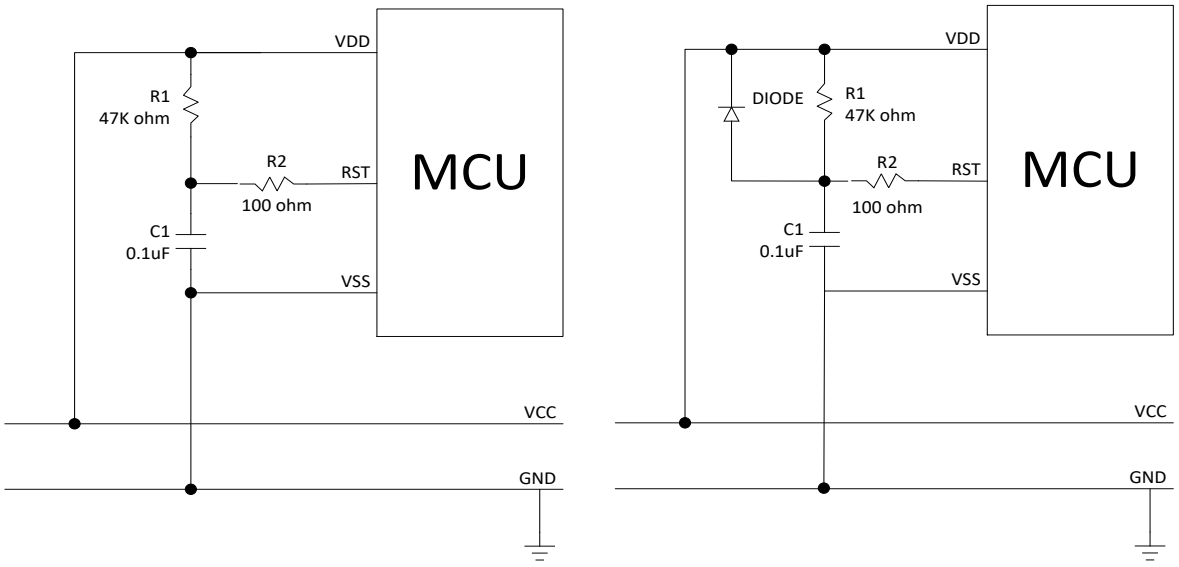
The operation mode of watchdog is configurable in options file:
Always mode counts its internal timer in all CPU operation modes (normal, IDLE, SLEEP);
Enable mode counts its internal timer during CPU stays in normal mode, and it would not trigger watchdog reset in IDLE and STOP modes;
Disable mode suspends its internal timer at all CPU modes, and the watchdog would not trigger in this condition.

When watchdog is operating in always mode, the system will consume additional power.

7.5 External Reset Pin

Programmable external reset pin is configurable in *options file*. Once it is enabled, it monitors its shared pin's logic level. A logical low (lower than 30% of VDD) would immediately trigger system reset until the input is recovered to high (lager than 70% of VDD).

An optional de-bounce period can improve reset signal's stability. Instead of immediate reset, the system reset requires an 8-ms-long logic low to avoid bouncing from a button key. Any signal lower than de-bounce period would not affect the CPU's execution.



✸ ***Note:***

1. The reset circuit is no any protection against unusual power or brown out reset on the left side of the figure.

2. The R2 100 ohm resistor of "Simply reset circuit" and "Diode & RC reset circuit" is necessary to limit any current flowing into reset pin from external capacitor C in the event of reset pin breakdown due to Electrostatic Discharge (ESD) or Electrical Over-stress (EOS) on the right side of the figure.

7.6 Software Reset

A software reset would be generated after consecutively set SRSTREQ register. As a result, this procedure enables firmware's ability to reset microcontroller (e.g. reset after firmware update). The following sample C code repeatedly set the least bit of SRST register to perform software reset.

```
1    SRST = 0x01;
2    SRST = 0x01;
```

7.7 Reset and Power-on Controller Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PFLAG	POR	WDT	RST	WARM	-	-	-	-
SRST	-	-	-	-	-	-	-	SRSTREQ
WDTR	WDTR7	WDTR6	WDTR5	WDTR4	WDTR3	WDTR2	WDTR1	WDTR0

PFLAG Register(0xFF)

Bit	Field	Type	Initial	Description
7	POR	R	-	This bit is automatically set if the microcontroller has been reset by LVD.
6	WDT	R	-	This bit is automatically set if the microcontroller has been reset by watchdog.
5	RST	R	-	This bit is automatically set if the microcontroller has been reset by external reset pin.
4	WARM	R	0	This bit is automatically set if the Ext. 32.768kHz crystal warm up is ready.
3..0	Reserved	R	0	

SRST Register(0xF7)

Bit	Field	Type	Initial	Description
7..1	Reserved	R	0	
0	SRSTREQ	R/W	-	Read: This bit is automatically set if the microcontroller has been reset by software reset. Write: Consecutively set this bit for two times to trigger software reset.

WDTR Register (0x86)

Bit	Field	Type	Initial	Description
7..0	WDTR[7:0]	W	-	Watchdog clear is controlled by WDTR register. Moving 0x5A data into WDTR is to reset watchdog timer.

8 System Clock and Power Management

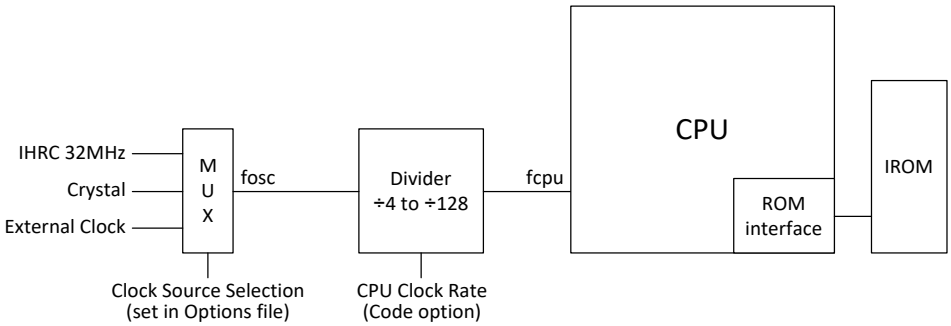
For power saving purpose, the microcontroller built in three different operation modes: normal, IDLE, and STOP mode.

The normal mode means that CPU and peripheral functions are under normally execution. The system clock is based on the combination of source selection, clock divider, and program memory wait state. IDLE mode is the situation that temporarily suspends CPU clock and its execution, yet it remains peripherals' functionality (e.g. timers, PWM, UART, and I2C). STOP mode disables all functions and clock generator until a wakeup signal to return normal mode.

8.1 System Clock

The microcontroller includes an on-chip clock generator (IHRC 32MHz), crystal/resonator driver, and an external clock input. The reset and power-on controller automatically loads clock source selection during power-on sequence. Therefore, the selected clock source is seen as 'fosc' domain which is a fixed frequency at any time.

Subsequently, the selected clock source (fosc) is divided by 4 to 128 times which is controlled by code option.



ROM interface is built in between CPU and IROM (program memory). It optionally extends the data fetching cycle in order to support lower speed program memory.

IROM fetching cycle (Instruction cycle) ≦ 8MHz

System clock rate and program memory extended cycle limitation as follows.

Code Option CPU Clock Source	Code Option CPU Clock Rate
IHRC 32M	000 = fosc / 128

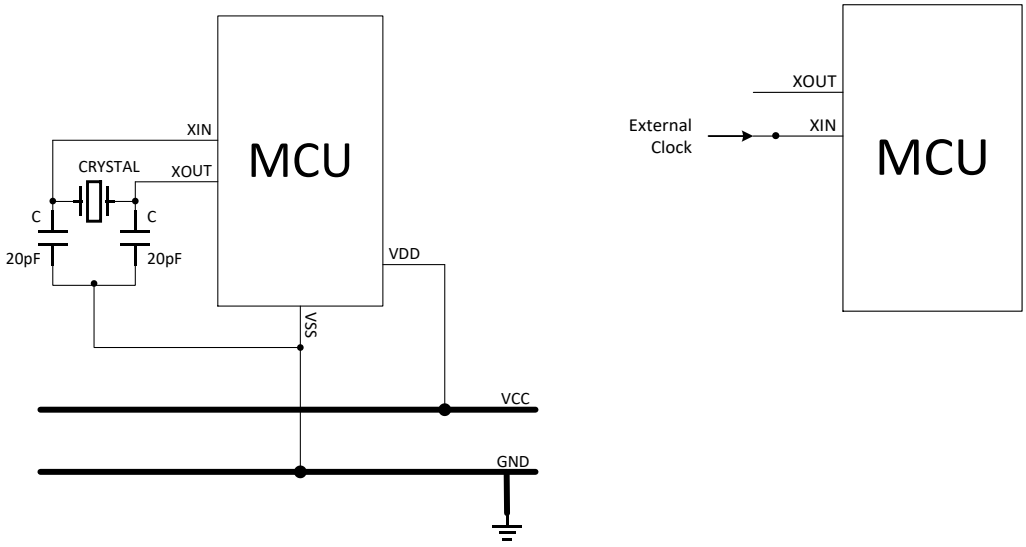
	001 = fosc / 64 010 = fosc / 32 011 = fosc / 16 100 = fosc / 8 101 = fosc / 4
X’tal 12M (Crystal 8-16MHz)	000 = fosc / 128 001 = fosc / 64 010 = fosc / 32 011 = fosc / 16 100 = fosc / 8 101 = fosc / 4 110 = fosc / 2
X’tal 4M (Crystal 1-4MHz)	000 = fosc / 128 001 = fosc / 64 010 = fosc / 32 011 = fosc / 16 100 = fosc / 8 101 = fosc / 4 110 = fosc / 2 111 = fosc / 1
External Clock (1-32MHz)	000 = fosc / 128 001 = fosc / 64 010 = fosc / 32 011 = fosc / 16 100 = fosc / 8 101 = fosc / 4 110 = fosc / 2 111 = fosc / 1

8.2 High Speed Clock

High-speed clock has internal and external two-type. The external high-speed clock includes 4MHz, 12MHz crystal/ceramic and external clock input mode. The internal high-speed oscillator is 32MHz RC type. These high-speed oscillators are selected by *OPTIONS _ SN8F5754.A51*.

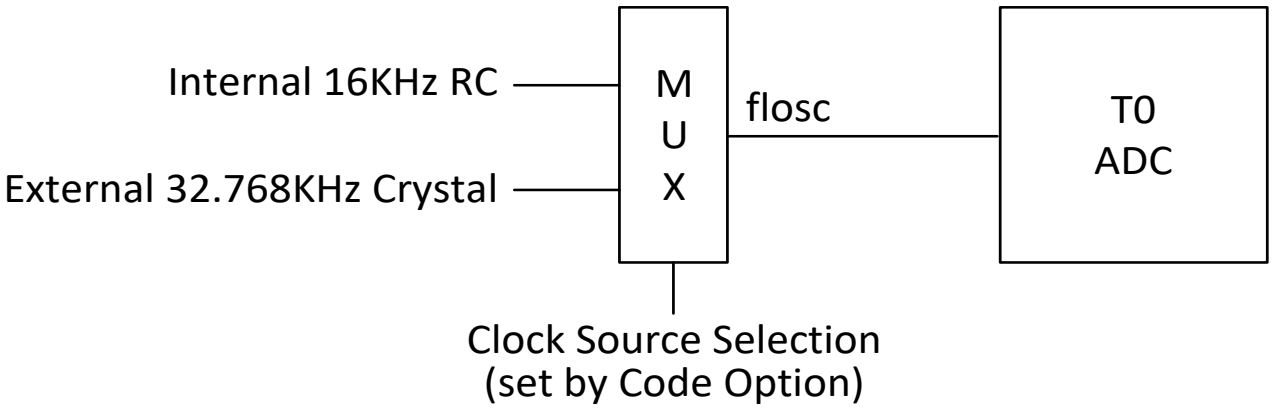
- IHRC 32M: The system high-speed clock source is internal high-speed 32MHz RC type oscillator. In the mode, XIN and XOUT pins are bi-direction GPIO mode, and not to connect any external oscillator device.

- X'tal 12M: The system high-speed clock source is external high-speed crystal/ceramic. The oscillator bandwidth is 8MHz~16MHz and connected to XIN/XOUT pins with 20pF capacitors to ground.
- X'tal 4M: The system high-speed clock source is external high-speed crystal/resonator. The oscillator bandwidth is 1MHz~4MHz and connected to XIN/XOUT pins with 20pF capacitors to ground.
- External Clock: The system high-speed clock source is external clock input mode. The input signal only connects to XIN pin, and the XOUT pin is bi-direction GPIO mode.



8.3 Low Speed Clock

SN8F5754 supplies low speed clock (flosc) for specific functions, such as T0 and ADC. The flosc has two clock sources selection: internal 16 kHz RC (ILRC) and external 32.768 kHz crystal, which is controlled by Code Option.



In external 32.768 kHz crystal mode, XIN and XOUT pins switch to crystal mode to drive an off-chip 32.768 kHz crystal after system loaded code option. Microcontroller supplies WARM bit (the 4th bit

of PFLAG) to indicate crystal warm-up status. The WARM bit is set when crystal warm-up procedure ready and cleared when crystal stop.

8.4 Noise Filter

The Noise Filter controlled by Noise Filter option is a low pass filter and supports crystal mode. The purpose is to filter high rate noise coupling on high clock signal from external oscillator. In high noisy environment, enable Noise Filter option is the strongly recommendation to reduce noise effect.

8.5 Power Management

After the end of reset signal and power-on sequence, the CPU starts program execution at the speed of fcpu. Overall, the CPU and all peripherals are functional in this situation (categorized as normal mode).

The least two bits of PCON register (IDLE at bit 0 and STOP at bit 1) control the microcontroller's power management unit.

If IDLE bit is set by program, only CPU clock source would be gated. Consequently, peripheral functions (such as timers, PWM, and I2C) and clock generator (IHRC 32 MHz/crystal driver) remain execution in this status. Any change from P0/P1 input and interrupt events can make the microcontroller turns back to normal mode, and the IDLE bit would be cleared automatically.

If STOP bit is set, by contrast, CPU, peripheral functions, and clock generator are suspended. Data storage in registers and RAM would be kept in this mode. Any change from P0/P1 can wake up the microcontroller and resume system's execution. STOP bit would be cleared automatically.

For user who is develop program in C language, IDLE and STOP macros is strongly recommended to control the microcontroller's system mode, instead of set IDLE and STOP bits directly.

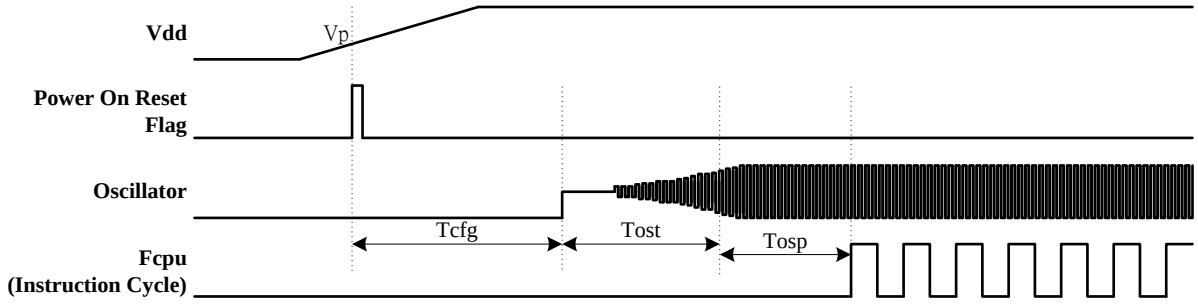
```
1     IDLE();
2     STOP();
```

SN8F5754 build in STWK bit (the 0th bit in SYSMOD register) to enable or disable flosc clock in STOP mode. If STWK=0, both fosc and flosc are suspended in STOP mode. If STWK=1, flosc clock keeps running in STOP mode.

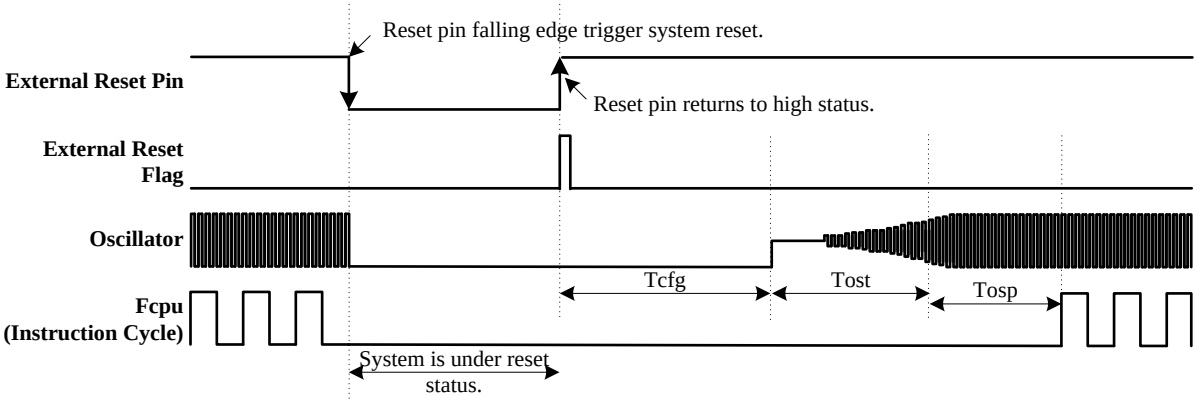
8.6 System Clock Timing

Parameter	Symbol	Description	Typical
Hardware configuration time	Tcfg	8* F_{ILRC} +2^{17}*F_{IHRC}	4.6ms @ F_{ILRC} = 16kHz & F_{IHRC} = 32MHz
Oscillator start up time	Tost	The start-up time is depended on oscillator's material, factory and architecture. Normally, the low-speed oscillator's start-up time is lower than high-speed oscillator. The RC type oscillator's start-up time is faster than crystal type oscillator.	-
Oscillator warm-up time	Tosp	Oscillator warm-up time of reset condition. 2048*F_{hosc} + 4*F_{ILRC} (Power on reset, LVD reset, watchdog reset, external reset pin active.)	762us @ F_{hosc} = 4MHz 378us @ F_{hosc} = 16MHz 314us @ F_{hosc} = 32MHz
		Oscillator warm-up time of power down mode wake-up condition. 2048*F_{hosc} + 4*F_{ILRC} ……Crystal/resonator type oscillator, e.g. 32768Hz crystal, 4MHz crystal, 16MHz crystal… 64*F_{hosc} + 4*F_{ILRC}……RC type oscillator, e.g. internal high-speed RC type oscillator.	X'tal: 762us @ F_{hosc} = 4MHz 378us @ F_{hosc} = 16MHz RC: 252us @ F_{hosc} = 32MHz

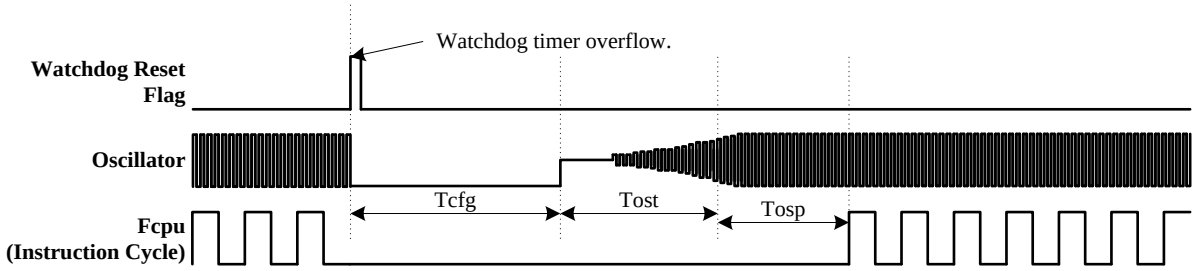
- Power On Reset Timing



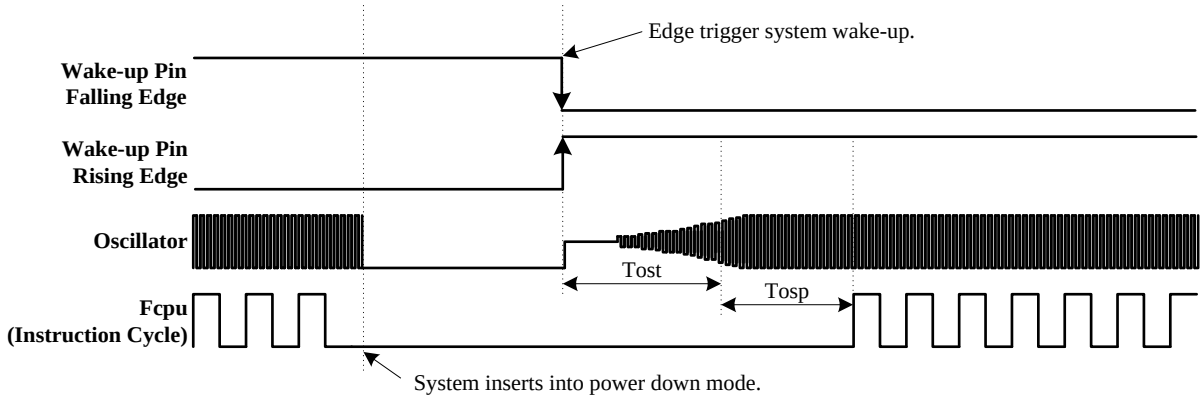
- External Reset Pin Reset Timing



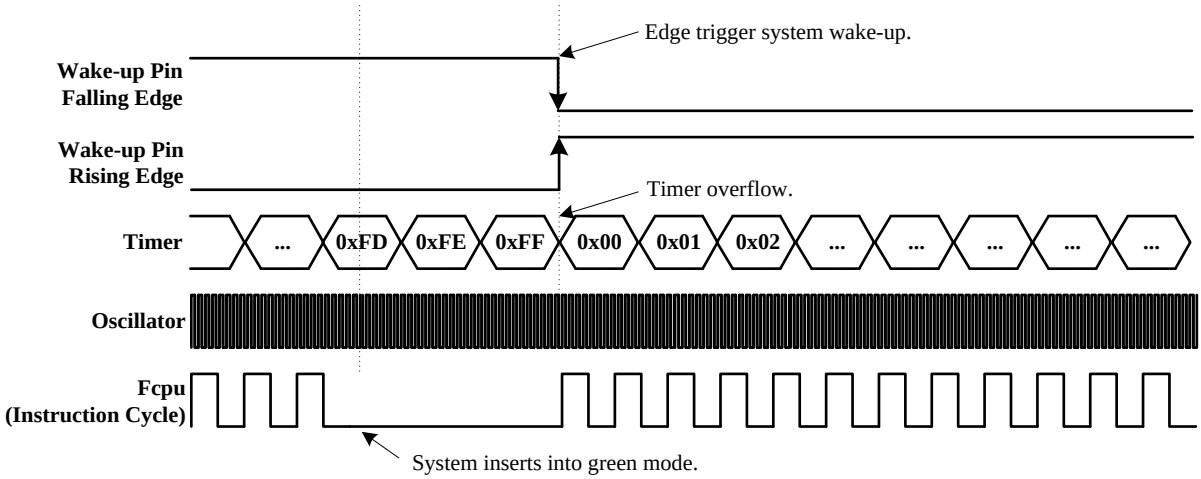
- Watchdog Reset Timing



- STOP Mode Wake-up Timing

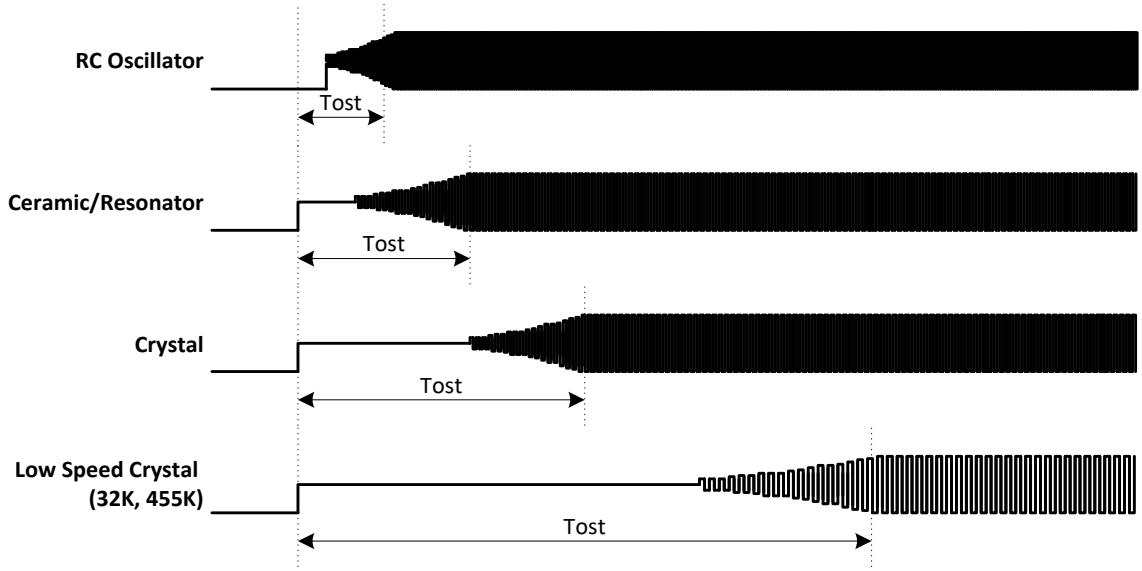


- IDLE Mode Wake-up Timing



- Oscillator Start-up Time

The start-up time is depended on oscillator's material, factory and architecture. Normally, the low-speed oscillator's start-up time is lower than high-speed oscillator.



8.7 System Clock and Power Management Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCON	-	-	-	-	-	GF0	STOP	IDLE
P1W	P17W	P16W	P15W	P14W	P13W	P12W	P11W	P10W
SYSMOD	-	-	-	-	-	-	-	STWK

P1W Register (0xF011)

Bit	Field	Type	Initial	Description
7..0	P1nW	R/W	0	0: Disable P1.n wakeup functionality 1: Enable P1.n wakeup functionality

PCON Register (0x87)

Bit	Field	Type	Initial	Description
7..3	Reserve	R	0	
2	GF0	R/W	0	General Purpose Flag
1	STOP	R/W	0	1: Microcontroller switch to STOP mode
0	IDLE	R/W	0	1: Microcontroller switch to IDLE mode

SYSMOD Register (0xC6)

Bit	Field	Type	Initial	Description
7..1	Reserved	R	0x00	
0	STWK	R/W	0	0: Both fosc and flosc are suspended in STOP mode. 1: flosc keep running in STOP mode*.

* Before entering STOP mode, the STWK bit setting must be earlier than the STOP bit.

9 System Operating Mode

The chip builds in three operating mode for difference clock rate and power saving reason. These modes control oscillators, op-code operation and analog peripheral devices' operation.

- Normal mode: System high-speed operating mode
- IDLE mode: System idle mode (Green mode)
- STOP mode: System power saving mode (Sleep mode)

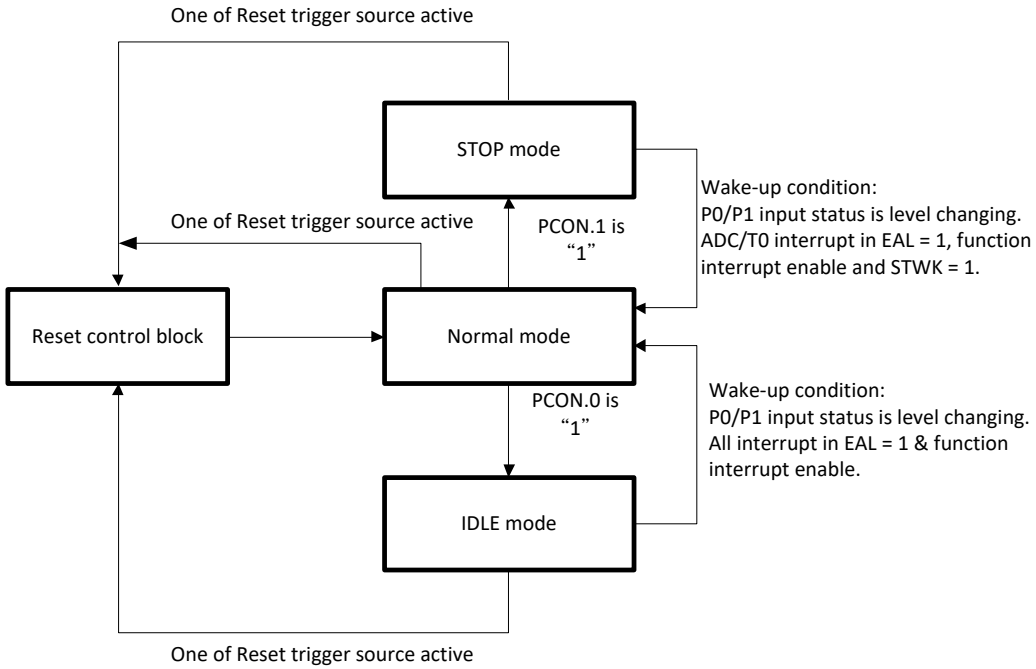


Table 9-1 The operating mode clock control

Operating Mode	Normal Mode	IDLE Mode	STOP Mode
IHRC	IHRC, IHRC RTC: Running Ext. OSC: Disable	IHRC, IHRC RTC: Running Ext. OSC: Disable	Stop
ILRC	Running	Running	STWK=1: Running Watchdog always: Running
Ext. OSC	IHRC: Disable IHRC OSC, EXT. OSC : Running	IHRC: Disable IHRC RTC, Ext. OSC : Running	Stop
CPU instruction	Executing	Stop	Stop
Timer 0 (Timer, Event counter)	Active by TR0	Active by TR0	Active as Timer 0 clock source is Flosc (ILRC or Ext.

			32kHz) & STWK=1
Timer 1 (Timer, Event counter)	Active by TR1	Active by TR1	Inactive
Timer 3	Active as enable	Active as enable	Inactive
PWM1/2	Active as enable	Active as enable	Inactive
UART0/1	Active as enable	Active as enable	Inactive
I2C	Active as enable	Active as enable	Inactive
ADC	Active as enable	Active as enable	Active as ADC clock source is Flosc (ILRC or Ext. 32kHz) & STWK=1
Watchdog timer	By Watchdog Code option	By Watchdog Code option	By Watchdog Code option
Internal interrupt	All active	All active	ADC/T0 interrupt is active when STWK = 1 and clock source is Flosc. Other inactive.
External interrupt	All active	All active	All inactive
Wakeup source	-	(1)P0, P1, Reset. (2)All interrupt in EAL = 1 & function interrupt enable.	(1)P0, P1, Reset. (2)ADC/T0 enable & clock source is Flosc (ILRC or Ext. 32kHz) & STWK=1 & function interrupt enable.

- Ext. OSC: External high-speed oscillator (XIN/XOUT).
- IHRC: Internal high-speed oscillator RC type.
- ILRC: Internal low-speed oscillator RC type.

9.1 Normal Mode

The Normal Mode is system high clock operating mode. The system clock source is from high speed oscillator. The program is executed. After power on and any reset trigger released, the system inserts into normal mode to execute program. When the system is wake-up from STOP/IDLE mode, the system also inserts into normal mode. In normal mode, the high speed oscillator is active, and the power consumption is largest of all operating modes.

- The program is executed, and full functions are controllable.
- The system rate is high speed.
- The high speed oscillator and internal low speed RC type oscillator are active.
- Normal mode can be switched to other operating modes through PCON register.
- STOP/IDLE mode is wake-up to normal mode.

9.2 STOP Mode

The STOP mode is the system ideal status. No program execution and oscillator operation. Only internal regulator is active to keep all control gates status, register status and SRAM contents. The STOP mode is waked up by P0/P1 hardware level change trigger. P0 wake-up function is always enables. The STOP mode is wake-up to normal mode. Inserting STOP mode is controlled by stop bit of PCON register. When stop = 1, the system inserts into STOP Mode. After system wake-up from STOP mode, the stop bit is disabled (zero status) automatically.

- The program stops executing, and full functions are disabled.
- All oscillators including external high speed oscillator, internal high speed oscillator and internal low speed oscillator stop.
- Only internal regulator is active to keep all control gates status, register status and SRAM contents.
- The system inserts into normal mode after wake-up from STOP mode.
- The STOP mode wake-up source is P0/P1 level change trigger, ADC/T0 enable & clock source is Flosc (ILRC or Ext. 32kHz) & STWK=1 & function interrupt enable.

9.3 IDLE Mode

The IDLE mode is another system ideal status not like STOP mode. In STOP mode, all functions and hardware devices are disabled. But in IDLE mode, the system clock source keeps running, so the power consumption of IDLE mode is larger than STOP mode. In IDLE mode, the program isn't executed, but the timer with wake-up function is active as enabled, and the timer clock source is the non-stop system clock. The IDLE mode has 2 wake-up sources. One is the P0/P1 level change trigger wake-up. The other one is any interrupt in EAL = 1 & function interrupt enable. That's mean users can setup any function with interrupt enable, and the system is waked up until the interrupt issue. Inserting IDLE mode is controlled by idle bit of PCON register. When idle = 1, the system inserts into IDLE mode. After system wake-up from IDLE mode, the idle bit is disabled (zero status) automatically.

- The program stops executing, and full functions are disabled.
- Only the timer with wake-up function is active.
- The oscillator to be the system clock source keeps running, and the other oscillators operation is depend on system operation mode configuration.
- If inserting IDLE mode from normal mode, the system insets to normal mode after wake-up.
- The IDLE mode wake-up sources are P0/P1 level change trigger.
- If the function clock source is system clock, the functions are workable as enabled and under IDLE mode, e.g. Timer, PWM, event counter…
- All interrupt in EAL = 1 & function interrupt enable can wake-up in IDLE mode.

9.4 Wake up

Under STOP mode (sleep mode) or idle mode, program doesn't execute. The wakeup trigger can wake the system up to normal mode. The wakeup trigger sources are external trigger (P0/P1 level change) and internal trigger (any interrupt in EAL = 1 & function interrupt enable). The wakeup function builds in interrupt operation issued request flag and trigger system executing interrupt service routine as system wakeup occurrence.

When the system is in STOP mode the high clock oscillator stops. When waked up from STOP mode, MCU waits for 2048 external high-speed oscillator clocks + 4 internal low-speed oscillator clocks and 64 internal high-speed oscillator clocks + 4 internal low-speed oscillator clocks as the wakeup time to stable the oscillator circuit. After the wakeup time, the system goes into the normal mode.

The value of the external high clock oscillator wakeup time is as the following.

The Wakeup time = 1/Fosc * 2048 + 1/Flosc * 4 + high clock start-up time

Example: In STOP mode (sleep mode), the system is waked up. After the wakeup time, the system goes into normal mode. The wakeup time is as the following.

The wakeup time = 1/Fosc * 2048 + 1/Flosc * 4 = 0.762 ms (Fosc = 4MHz)

The total wakeup time = 0.762 ms + oscillator start-up time

The value of the internal high clock oscillator RC type wakeup time is as the following.

The Wakeup time = 1/Fosc * 64 + 1/Flosc * 4 + high clock start-up time

Example: In STOP mode (sleep mode), the system is waked up. After the wakeup time, the system goes into normal mode. The wakeup time is as the following.

The wakeup time = 1/Fosc * 64 + 1/Flosc * 4= 252 us (Fhosc = 32MHz)

✷ ***Note: The high clock start-up time is depended on the VDD and oscillator type of high clock.***

Under STOP mode and green mode, the I/O ports with wakeup function are able to wake the system up to normal mode. The wake-up trigger edge is level changing in rising edge or falling edge. The Port 0 and Port 1 have wakeup function. Port 0 wakeup functions always enables, but the Port 1 is controlled by the P1W register.

P1W Register (0xF011)

Bit	Field	Type	Initial	Description
7..0	P1nW	R/W	0	0: Disable P1.n wakeup functionality 1: Enable P1.n wakeup functionality

10 Interrupt

The MCU provides 14 interrupt sources (3 external and 11 interrupt) with 4 priority levels. Each interrupt source includes one or more interrupt request flag(s). When interrupt event occurs, the associated interrupt flag is set to logic 1. If both interrupt enable bit and global interrupt (EAL=1) are enabled, the interrupt request is generated and interrupt service routine (ISR) will be started. Some interrupt request flags must be cleared by software. However, most interrupt request flags can be cleared by hardware automatically. In the end, ISR is finished after complete the RETI instruction. The summary of interrupt source, interrupt vector, priority order and control bit are shown as the table below.

Table 10-1 The interrupt list

Interrupt	Enable Interrupt	Request (IRQ)	IRQ Clearance	Priority / Vector
System Reset	-	-	-	0 / 0x0000
INT0	EX0	IE0	Automatically	1 / 0x0003
Timer 3	ET3	TF3	Automatically	2 / 0x0033
PWM1	EPW1	PW1F	Automatically	3 / 0x0083
I2C	EI2C	SI	By firmware	4 / 0x0043
Timer 0	ET0	TF0	Automatically	5 / 0x000B
PWM2	EPW2	PW2F	Automatically	6 / 0x008B
INT1	EX1	IE1	Automatically	7 / 0x0013
UART0 RX	EU0RX	RI0	By firmware	8 / 0x0053
Timer 1	ET1	TF1	Automatically	9 / 0x001B
UART0 TX	EU0TX	TI0	By firmware	10 / 0x005B
INT2	EX2	IE2	Automatically	11 / 0x0023
UART1 RX	EU1RX	RI1	By firmware	12 / 0x0063
ADC	EADC	ADCF	Automatically	13 / 0x00AB
UART1 TX	EU1TX	TI1	By firmware	14 / 0x006B

10.1 Interrupt Operation

Interrupt operation is controlled by interrupt request flag and interrupt enable bits. Interrupt request flag is interrupt source event indicator, no matter what interrupt function status (enable or disable). Both interrupt enable bit and global interrupt (EAL=1) are enabled, the system executes interrupt operation when each of interrupt request flags is active. The program counter points to interrupt vector (0x03 – 0xEB) and execute ISR.

10.2 Interrupt Priority

Each interrupt source has its specific default priority order. If two interrupts occurs simultaneously, the higher priority ISR will be service first. The lower priority ISR will be serviced after the higher priority ISR completes. The next ISR will be service after the previous ISR complete, no matter the priority order.

For special priority needs, 4-level priority levels (Level 0 – Level 3) are used. All interrupt sources are classified into 6 priority groups (Group0 – Group5). Each group can be set one specific priority level. Priority level is selected by IP0/IP1 registers. Level 3 is the highest priority and Level 0 is the lowest. The interrupt sources inside the same group will share the same priority level. With the same priority level, the priority rule follows default priority.

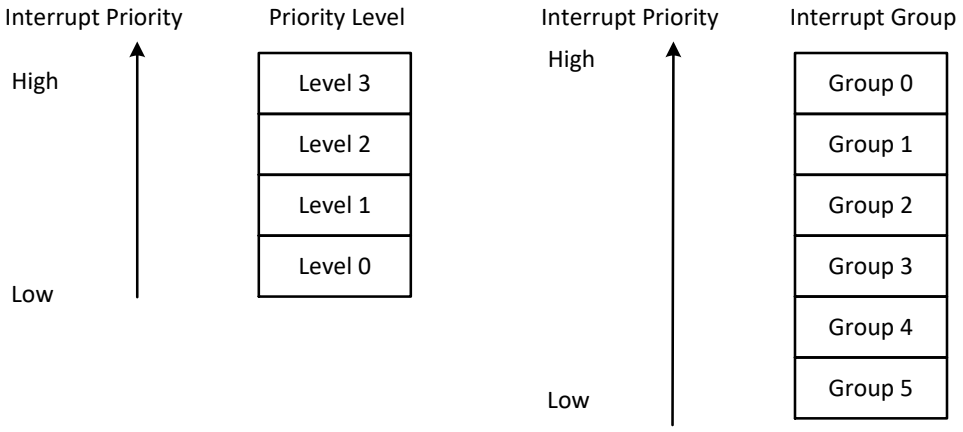
Priority Level	IP1.x	IP0.x
Level 0	0	0
Level 1	0	1
Level 2	1	0
Level 3	1	1

The ISR with the higher priority level can be serviced first; even can break the on-going ISR with the lower priority level. The ISR with the lower priority level will be pending until the ISR with the higher priority level completes.

Group	Interrupt Source				
Group 0	INT0	Timer 3	PWM1	-	I2C
Group 1	Timer 0	-	PWM2	-	-
Group 2	INT1	-	-	-	UART0 RX
Group 3	Timer 1	-	-	-	UART0 TX
Group 4	INT2	-	-	-	UART1 RX
Group 5	-	-	ADC	-	UART1 TX

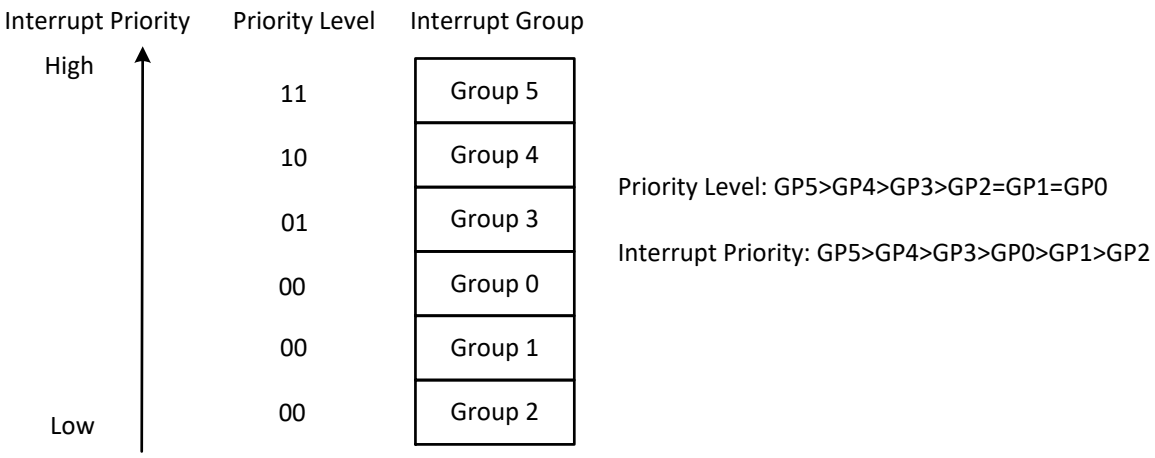
When more than one interrupt request occur, the highest priority request must be executed first. Choose the highest priority request according natural priority and priority level. The steps are as the following:

1. Choose the groups which have the highest priority level between all groups.
2. Choose the group which is the highest nature priority between the groups with the highest priority level.
3. Choose the ISR which has the highest nature priority inside the group with the highest priority.



As the example, group5 has the highest priority level and group0~group2 have the lowest priority level. It means the interrupt vector in group5 has the highest interrupt priority, the 2nd interrupt priority in group4 and the 3rd interrupt priority in group3. Group0~ group2 have the same priority level thus the nature priority rule will be followed. Therefore, interrupt priority will be group5> group4> group3> group0> group1> group2.

```
MOV     IP0, #00101000B         ; Set group0 - group5 in different priority level.
MOV     IP1, #00110000B
```



IP0, IP1 Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IP0	-	-	IP05	IP04	IP03	IP02	IP01	IP00
IP1	-	-	IP15	IP14	IP13	IP12	IP11	IP10

IP0 Register (0xA9)

Bit	Field	Type	Initial	Description
5..0	IP0[5:0]	R/W	0	Interrupt priority. Each bit together with corresponding bit from IP1 register specifies the priority level of the respective interrupt priority group.
Else	Reserved	R	0	

IP1 Register (0xB9)

Bit	Field	Type	Initial	Description
5..0	IP1[5:0]	R/W	0	Interrupt priority. Each bit together with corresponding bit from IP0 register specifies the priority level of the respective interrupt priority group.
Else	Reserved	R	0	

10.3 Interrupt Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IEN0	EAL	-	-	EX2	ET1	EX1	ET0	EX0
IEN1	-	-	EU1RX	EU1TX	EU0RX	EU0TX	-	EI2C
IEN2	-	-	ET3	-	EPW2	EPW1	-	EADC
IRCON	-	-	-	-	-	-	-	IE2
IRCON2	-	-	TF3	-	PW2F	PW1F	-	ADCF
TCON	TF1	TR1	TF0	TR0	IE1	-	IE0	-
S0CON	S0M0	S0M1	S0M20	REN0	TB80	RB80	TI0	RI0
S1CON	S1M0	S1M1	S1M20	REN1	TB81	RB81	TI1	RI1
I2CCON	CR2	ENS1	STA	STO	SI	AA	CR1	CR0

IEN0 Register (0xA8)

Bit	Field	Type	Initial	Description
7	EAL	R/W	0	Enable all interrupt control bit. 0: Disable all interrupt function. 1: Enable all interrupt function.
6..5	Reserved	R	0	
4	EX2	R/W	0	External P0.7 interrupt (INT2) control bit. 0: Disable INT2 interrupt function. 1: Enable INT2 interrupt function.
3	ET1	R/W	0	T1 timer interrupt control bit. 0: Disable T1 interrupt function. 1: Enable T1 interrupt function.
2	EX1	R/W	0	External P0.6 interrupt (INT1) control bit. 0: Disable INT1 interrupt function. 1: Enable INT1 interrupt function.
1	ET0	R/W	0	T0 timer interrupt control bit. 0: Disable T0 interrupt function. 1: Enable T0 interrupt function
0	EX0	R/W	0	External P0.5 interrupt (INT0) control bit. 0: Disable INT0 interrupt function. 1: Enable INT0 interrupt function.

IEN1 Register (0xB8)

Bit	Field	Type	Initial	Description
7..6	Reserved	R	0	
5	EU1RX	R/W	0	UART1 RX interrupt control bit. 0: Disable UART RX interrupt function. 1: Enable UART RX interrupt function.
4	EU1TX	R/W	0	UART1 TX interrupt control bit. 0: Disable UART TX interrupt function. 1: Enable UART TX interrupt function.
3	EU0RX	R/W	0	UART0 RX interrupt control bit. 0: Disable UART RX interrupt function. 1: Enable UART RX interrupt function.
2	EU0TX	R/W	0	UART0 TX interrupt control bit. 0: Disable UART TX interrupt function. 1: Enable UART TX interrupt function.
1	Reserved	R	0	

0	EI2C	R/W	0	I2C interrupt control bit. 0: Disable I2C interrupt function. 1: Enable I2C interrupt function.

IEN2 Register (0xC8)

Bit	Field	Type	Initial	Description
7..6	Reserved	R	0	
5	ET3	R/W	0	Timer 3 interrupt control bit. 0: Disable T3 interrupt function. 1: Enable T3 interrupt function.
4	Reserved	R	0	
3	EPW2	R/W	0	PWM2 interrupt control bit. 0 = Disable PWM2 interrupt function. 1 = Enable PWM2 interrupt function.
2	EPW1	R/W	0	PWM1 interrupt control bit. 0 = Disable PWM1 interrupt function. 1 = Enable PWM1 interrupt function.
1	Reserved	R	0	
0	EADC	R/W	0	ADC interrupt control bit. 0: Disable ADC interrupt function. 1: Enable ADC interrupt function.

IRCON Register (0xA5)

Bit	Field	Type	Initial	Description
7..1	Reserved	R	0	
0	IE2	R/W	0	External P0.7 interrupt (INT2) request flag 0: None INT2 interrupt request. 1: INT2 interrupt request.

IRCON2 Register (0xA6)

Bit	Field	Type	Initial	Description
7..6	Reserved	R	0	
5	TF3	R/W	0	Timer 3 interrupt request flag. 0: None T3 interrupt request 1: T3 interrupt request.
4	Reserved	R	0	
3	PW2F	R/W	0	PWM2 interrupt request flag. 0: None PWM2 interrupt request

				1: PWM2 interrupt request.
2	PW1F	R/W	0	PWM1 interrupt request flag. 0: None PWM1 interrupt request 1: PWM1 interrupt request.
1	Reserved	R	0	
0	ADCF	R/W	0	ADC interrupt request flag. 0: None ADC interrupt request. 1: ADC interrupt request.

TCON Register (0x88)

Bit	Field	Type	Initial	Description
7	TF1	R/W	0	Timer 1 interrupt request flag. 0: None T1 interrupt request 1: T1 interrupt request.
5	TF0	R/W	0	Timer 0 interrupt request flag. 0: None T0 interrupt request 1: T0 interrupt request.
3	IE1	R/W	0	External P0.6 interrupt (INT1) request flag 0: None INT1 interrupt request. 1: INT1 interrupt request.
1	IE0	R/W	0	External P0.5 interrupt (INT0) request flag 0: None INT1 interrupt request. 1: INT1 interrupt request.
Else				Refer to other chapter(s)

S0CON Register (0x97)

Bit	Field	Type	Initial	Description
1	TI0	R/W	0	UART0 transmit interrupt request flag. It indicates completion of a serial transmission at UART0. It is set by hardware at the end of bit 8 in mode 0 or at the beginning of a stop bit in other modes. It must be cleared by software. 0: None UART0 transmit interrupt request. 1: UART0 transmit interrupt request.
0	RI0	R/W	0	UART0 receive interrupt request flag. It is set by hardware after completion of a serial reception at UART0. It is set by hardware at the end of bit 8 in mode 0 or in the middle of a stop bit in other modes. It must

				be cleared by software. 0: None UART0 receive interrupt request. 1: UART0 receive interrupt request.
Else				Refer to other chapter(s)

S1CON Register (0x9C)

Bit	Field	Type	Initial	Description
1	TI1	R/W	0	UART1 transmit interrupt request flag. It indicates completion of a serial transmission at UART1. It is set by hardware at the end of bit 8 in mode 0 or at the beginning of a stop bit in other modes. It must be cleared by software. 0: None UART1 transmit interrupt request. 1: UART1 transmit interrupt request.
0	RI1	R/W	0	UART1 receive interrupt request flag. It is set by hardware after completion of a serial reception at UART1. It is set by hardware at the end of bit 8 in mode 0 or in the middle of a stop bit in other modes. It must be cleared by software. 0: None UART1 receive interrupt request. 1: UART1 receive interrupt request.
Else				Refer to other chapter(s)

I2CCON Register (0xDC)

Bit	Field	Type	Initial	Description
3	SI	R/W	0	Serial interrupt flag The SI is set by hardware when one of 25 out of 26 possible I2C states is entered. The only state that does not set the SI is state F8h, which indicates that no relevant state information is available. The SI flag must be cleared by software. In order to clear the SI bit, ‘0’ must be written to this bit. Writing a ‘1’ to SI bit does not change value of the SI.
Else				Refer to other chapter(s)

11 GPIO

The microcontroller has up to 60 bidirectional general purpose I/O pin (GPIO). Unlike the original 8051 only has open-drain output, SN8F5754 builds in push-pull output structure to improve its driving performance.

11.1 Input and Output Control

The input and output direction control is configurable through P0M to P3M registers. These bits specify each pin that is either input mode or output mode.

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0M	P07M	P06M	P05M	P04M	P03M	P02M	P01M	P00M
P1M	P17M	P16M	P15M	P14M	P13M	P12M	P11M	P10M
P2M	-	-	-	P24M	P23M	P22M	P21M	P20M
P3M	-	-	-	-	P33M	P32M	P31M	P30M
P0OC	-	-	-	-	P03OC	P02OC	-	-
P1OC						P12OC		
P2OC							P21OC	P20OC
P3OC					P33OC	P32OC		

P0M: 0xF9

Bit	Field	Type	Initial	Description
7	P07M	R/W	0	Mode selection of P0.7 0: Input mode 1: Output mode
6	P06M	R/W	0	Mode selection of P0.6 0: Input mode 1: Output mode
5	P05M	R/W	0	Mode selection of P0.5 0: Input mode 1: Output mode
4	P04M	R/W	0	Mode selection of P0.4 0: Input mode 1: Output mode
3	P03M	R/W	0	Mode selection of P0.3 0: Input mode 1: Output mode
2	P02M	R/W	0	Mode selection of P0.2

				0: Input mode 1: Output mode
1	P01M	R/W	0	Mode selection of P0.1 0: Input mode 1: Output mode
0	P00M	R/W	0	Mode selection of P0.0 0: Input mode 1: Output mode

P1M: 0xFA

Bit	Field	Type	Initial	Description
7	P17M	R/W	0	Mode selection of P1.7 0: Input mode 1: Output mode
6	P16M	R/W	0	Mode selection of P1.6 0: Input mode 1: Output mode
5	P15M	R/W	0	Mode selection of P1.5 0: Input mode 1: Output mode
4	P14M	R/W	0	Mode selection of P1.4 0: Input mode 1: Output mode
3	P13M	R/W	0	Mode selection of P1.3 0: Input mode 1: Output mode
2	P12M	R/W	0	Mode selection of P1.2 0: Input mode 1: Output mode
1	P11M	R/W	0	Mode selection of P1.1 0: Input mode 1: Output mode
0	P10M	R/W	0	Mode selection of P1.0 0: Input mode 1: Output mode

P2M: 0xFB

Bit	Field	Type	Initial	Description
4	P24M	R/W	0	Mode selection of P2.4 0: Input mode 1: Output mode
3	P23M	R/W	0	Mode selection of P2.3 0: Input mode 1: Output mode
2	P22M	R/W	0	Mode selection of P2.2 0: Input mode 1: Output mode
1	P21M	R/W	0	Mode selection of P2.1 0: Input mode 1: Output mode
0	P20M	R/W	0	Mode selection of P2.0 0: Input mode 1: Output mode

P3M: 0xFC

Bit	Field	Type	Initial	Description
3	P33M	R/W	0	Mode selection of P3.3 0: Input mode 1: Output mode
2	P32M	R/W	0	Mode selection of P3.2 0: Input mode 1: Output mode
1	P31M	R/W	0	Mode selection of P3.1 0: Input mode 1: Output mode
0	P30M	R/W	0	Mode selection of P3.0 0: Input mode 1: Output mode

✸ ***Note: The pins which are not pin-out shall be set correctly to decrease power consumption in low-power modes. Strongly recommended to set these pins as input pull-up.***

P0OC Register (0xF020)

Bit	Field	Type	Initial	Description
7..4	Reserved	R	0	
3	P03OC	R/W	0	P0.3 open-drain control bit 0: Disable 1: Enable
2	P02OC	R/W	0	P0.2 open-drain control bit 0: Disable 1: Enable
1..0	Reserved	R	0	

P1OC Register (0xF021)

Bit	Field	Type	Initial	Description
7..3	Reserved	R	0	
2	P12OC	R/W	0	P1.2 open-drain control bit 0: Disable 1: Enable
1..0	Reserved	R	0	

P2OC Register (0xF022)

Bit	Field	Type	Initial	Description
7..2	Reserved	R	0	
1	P21OC	R/W	0	P2.1 open-drain control bit 0: Disable 1: Enable
0	P20OC	R/W	0	P2.0 open-drain control bit 0: Disable 1: Enable

P3OC Register (0xF023)

Bit	Field	Type	Initial	Description
7..4	Reserved	R	0	
3	P33OC	R/W	0	P3.3 open-drain control bit 0: Disable 1: Enable
2	P32OC	R/W	0	P3.2 open-drain control bit 0: Disable

				1: Enable
1..0	Reserved	R	0	

11.2 Input Data and Output Data

By a read operation from any registers of P0 to P3, the current pin's logic level would be fetch to represent its external status. This operation remains functional even the pin is shared with other function like UART and I2C which can monitor the bus condition in some case.

A write P0 to P3 register value would be latched immediately, yet the value would be outputted until the mapped P0M – P3M is set to output mode. If the pin is currently in output mode, any value set to P0 to P3 register would be presented on the pin immediately.

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0	P07	P06	P05	P04	P03	P02	P01	P00
P1	P17	P16	P15	P14	P13	P12	P11	P10
P2	-	-	-	P24	P23	P22	P21	P20
P3	-	-	-	-	P33	P32	P31	P30

P0: 0x80

Bit	Field	Type	Initial	Description
7	P07	R/W	1	Read: P0.7 pin's logic level Write 1/0: Output logic high or low (applied if P07M = 1)
6	P06	R/W	1	Read: P0.6 pin's logic level Write 1/0: Output logic high or low (applied if P06M = 1)
5	P05	R/W	1	Read: P0.5 pin's logic level Write 1/0: Output logic high or low (applied if P05M = 1)
4	P04	R/W	1	Read: P0.4 pin's logic level Write 1/0: Output logic high or low (applied if P04M = 1)
3	P03	R/W	1	Read: P0.3 pin's logic level Write 1/0: Output logic high or low (applied if P03M = 1)
2	P02	R/W	1	Read: P0.2 pin's logic level Write 1/0: Output logic high or low (applied if P02M = 1)
1	P01	R/W	1	Read: P0.1 pin's logic level Write 1/0: Output logic high or low (applied if P01M = 1)
0	P00	R/W	1	Read: P0.0 pin's logic level Write 1/0: Output logic high or low (applied if P00M = 1)

P1: 0x90

Bit	Field	Type	Initial	Description
7	P17	R/W	1	Read: P1.7 pin's logic level Write 1/0: Output logic high or low (applied if P17M = 1)
6	P16	R/W	1	Read: P1.6 pin's logic level Write 1/0: Output logic high or low (applied if P16M = 1)
5	P15	R/W	1	Read: P1.5 pin's logic level Write 1/0: Output logic high or low (applied if P15M = 1)
4	P14	R/W	1	Read: P1.4 pin's logic level Write 1/0: Output logic high or low (applied if P14M = 1)
3	P13	R/W	1	Read: P1.3 pin's logic level Write 1/0: Output logic high or low (applied if P13M = 1)
2	P12	R/W	1	Read: P1.2 pin's logic level Write 1/0: Output logic high or low (applied if P12M = 1)
1	P11	R/W	1	Read: P1.1 pin's logic level Write 1/0: Output logic high or low (applied if P11M = 1)
0	P10	R/W	1	Read: P1.0 pin's logic level Write 1/0: Output logic high or low (applied if P10M = 1)

P2: 0xA0

Bit	Field	Type	Initial	Description
4	P24	R/W	1	Read: P2.4 pin's logic level Write 1/0: Output logic high or low (applied if P24M = 1)
3	P23	R/W	1	Read: P2.3 pin's logic level Write 1/0: Output logic high or low (applied if P23M = 1)
2	P22	R/W	1	Read: P2.2 pin's logic level Write 1/0: Output logic high or low (applied if P22M = 1)
1	P21	R/W	1	Read: P2.1 pin's logic level Write 1/0: Output logic high or low (applied if P21M = 1)
0	P20	R/W	1	Read: P2.0 pin's logic level Write 1/0: Output logic high or low (applied if P20M = 1)

P3: 0xB0

Bit	Field	Type	Initial	Description
3	P33	R/W	1	Read: P3.3 pin's logic level Write 1/0: Output logic high or low (applied if P33M = 1)
2	P32	R/W	1	Read: P3.2 pin's logic level Write 1/0: Output logic high or low (applied if P32M = 1)

1	P31	R/W	1	Read: P3.1 pin's logic level Write 1/0: Output logic high or low (applied if P31M = 1)
0	P30	R/W	1	Read: P3.0 pin's logic level Write 1/0: Output logic high or low (applied if P30M = 1)

11.3 On-chip Pull-up Resisters

The P0UR to P3UR registers are mapped to each pins' internal 100k ohm (in typical value) pull-up resister.

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0UR	P07UR	P06UR	P05UR	P04UR	P03UR	P02UR	P01UR	P00UR
P1UR	P17UR	P16UR	P15UR	P14UR	P13UR	P12UR	P11UR	P10UR
P2UR	-	-	-	P24UR	P23UR	P22UR	P21UR	P20UR
P3UR	-	-	-	-	P33UR	P32UR	P31UR	P30UR

P0UR: 0xF000

Bit	Field	Type	Initial	Description
7	P07UR	R/W	0	On-chip pull-up resister control of P0.7 0: Disable* 1: Enable
6	P06UR	R/W	0	On-chip pull-up resister control of P0.6 0: Disable* 1: Enable
5	P05UR	R/W	0	On-chip pull-up resister control of P0.5 0: Disable* 1: Enable
4	P04UR	R/W	0	On-chip pull-up resister control of P0.4 0: Disable* 1: Enable
3	P03UR	R/W	0	On-chip pull-up resister control of P0.3 0: Disable* 1: Enable
2	P02UR	R/W	0	On-chip pull-up resister control of P0.2 0: Disable* 1: Enable
1	P01UR	R/W	0	On-chip pull-up resister control of P0.1 0: Disable* 1: Enable

0	P00UR	R/W	0	On-chip pull-up resister control of P0.0 0: Disable* 1: Enable

* Recommended disable pull-up resister if the pin is output mode or analog function

P1UR: 0xF001

Bit	Field	Type	Initial	Description
7	P17UR	R/W	0	On-chip pull-up resister control of P1.7 0: Disable* 1: Enable
6	P16UR	R/W	0	On-chip pull-up resister control of P1.6 0: Disable* 1: Enable
5	P15UR	R/W	0	On-chip pull-up resister control of P1.5 0: Disable* 1: Enable
4	P14UR	R/W	0	On-chip pull-up resister control of P1.4 0: Disable* 1: Enable
3	P13UR	R/W	0	On-chip pull-up resister control of P1.3 0: Disable* 1: Enable
2	P12UR	R/W	0	On-chip pull-up resister control of P1.2 0: Disable* 1: Enable
1	P11UR	R/W	0	On-chip pull-up resister control of P1.1 0: Disable* 1: Enable
0	P10UR	R/W	0	On-chip pull-up resister control of P1.0 0: Disable* 1: Enable

* Recommended disable pull-up resister if the pin is output mode or analog function

P2UR: 0xF002

Bit	Field	Type	Initial	Description
4	P24UR	R/W	0	On-chip pull-up resister control of P2.4 0: Disable* 1: Enable

3	P23UR	R/W	0	On-chip pull-up resister control of P2.3 0: Disable* 1: Enable
2	P22UR	R/W	0	On-chip pull-up resister control of P2.2 0: Disable* 1: Enable
1	P21UR	R/W	0	On-chip pull-up resister control of P2.1 0: Disable* 1: Enable
0	P20UR	R/W	0	On-chip pull-up resister control of P2.0 0: Disable* 1: Enable

* Recommended disable pull-up resister if the pin is output mode or analog function

P3UR: 0xF003

Bit	Field	Type	Initial	Description
3	P33UR	R/W	0	On-chip pull-up resister control of P3.3 0: Disable* 1: Enable
2	P32UR	R/W	0	On-chip pull-up resister control of P3.2 0: Disable* 1: Enable
1	P31UR	R/W	0	On-chip pull-up resister control of P3.1 0: Disable* 1: Enable
0	P30UR	R/W	0	On-chip pull-up resister control of P3.0 0: Disable* 1: Enable

* Recommended disable pull-up resister if the pin is output mode or analog function

11.4 Pin Shared with Analog Function

The microcontroller builds in analog functions, such as ADC. The Schmitt trigger of input channel is strongly recommended to switch off if the pin's shared analog function is enabled.

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0CON	P0CON7	P0CON6	P0CON5	P0CON4	P0CON3	P0CON2	P0CON1	P0CON0
P1CON	P1CON7	P1CON6	P1CON5	P1CON4	P1CON3	P1CON2	P1CON1	P1CON0
P2CON	-	-	-	P2CON4	P2CON3	P2CON2	P2CON1	P2CON0

P3CON	-	-	-	-	P3CON3	P3CON2	P3CON1	P3CON0

P0CON: 0XF018

Bit	Field	Type	Initial	Description
7	P0CON7	R/W	0	P0.7 configuration control bit*. 0: P0.7 can be analog input pin or digital GPIO pin. 1: P0.7 is pure analog input pin and can't be a digital GPIO pin.
6	P0CON6	R/W	0	P0.6 configuration control bit*. 0: P0.6 can be analog input pin or digital GPIO pin. 1: P0.6 is pure analog input pin and can't be a digital GPIO pin.
5	P0CON5	R/W	0	P0.5 configuration control bit*. 0: P0.5 can be analog input pin or digital GPIO pin. 1: P0.5 is pure analog input pin and can't be a digital GPIO pin.
4	P0CON4	R/W	0	P0.4 configuration control bit*. 0: P0.4 can be analog input pin or digital GPIO pin. 1: P0.4 is pure analog input pin and can't be a digital GPIO pin.
3	P0CON3	R/W	0	P0.3 configuration control bit*. 0: P0.3 can be analog input pin or digital GPIO pin. 1: P0.3 is pure analog input pin and can't be a digital GPIO pin.
2	P0CON2	R/W	0	P0.2 configuration control bit*. 0: P0.2 can be analog input pin or digital GPIO pin. 1: P0.2 is pure analog input pin and can't be a digital GPIO pin.
1	P0CON1	R/W	0	P0.1 configuration control bit*. 0: P0.1 can be analog input pin or digital GPIO pin. 1: P0.1 is pure analog input pin and can't be a digital GPIO pin.
0	P0CON0	R/W	0	P0.0 configuration control bit*. 0: P0.0 can be analog input pin or digital GPIO pin. 1: P0.0 is pure analog input pin and can't be a digital GPIO pin.

* P0CON [7:0], P1CON [7:0], P2CON [7:0], P3CON [7:0] will configure related Port0 , Port1, Port2, Port3 pin as pure analog input pin to avoid current leakage.

P1CON: 0XF019

Bit	Field	Type	Initial	Description
7	P1CON7	R/W	0	P1.7 configuration control bit*. 0: P1.7 can be analog input pin or digital GPIO pin. 1: P1.7 is pure analog input pin and can't be a digital GPIO pin.
6	P1CON6	R/W	0	P1.6 configuration control bit*. 0: P1.6 can be analog input pin or digital GPIO pin. 1: P1.6 is pure analog input pin and can't be a digital GPIO pin.
5	P1CON5	R/W	0	P1.5 configuration control bit*. 0: P1.5 can be analog input pin or digital GPIO pin. 1: P1.5 is pure analog input pin and can't be a digital GPIO pin.
4	P1CON4	R/W	0	P1.4 configuration control bit*. 0: P1.4 can be analog input pin or digital GPIO pin. 1: P1.4 is pure analog input pin and can't be a digital GPIO pin.
3	P1CON3	R/W	0	P1.3 configuration control bit*. 0: P1.3 can be analog input pin or digital GPIO pin. 1: P1.3 is pure analog input pin and can't be a digital GPIO pin.
2	P1CON2	R/W	0	P1.2 configuration control bit*. 0: P1.2 can be analog input pin or digital GPIO pin. 1: P1.2 is pure analog input pin and can't be a digital GPIO pin.
1	P1CON1	R/W	0	P1.1 configuration control bit*. 0: P1.1 can be analog input pin or digital GPIO pin. 1: P1.1 is pure analog input pin and can't be a digital GPIO pin.
0	P1CON0	R/W	0	P1.0 configuration control bit*. 0: P1.0 can be analog input pin or digital GPIO pin. 1: P1.0 is pure analog input pin and can't be a digital GPIO pin.

* P0CON [7:0], P1CON [7:0], P2CON [7:0], P3CON [7:0] will configure related Port0 , Port1, Port2, Port3 pin as pure analog input pin to avoid current leakage.

P2CON: 0xF01A

Bit	Field	Type	Initial	Description
4	P2CON4	R/W	0	P2.4 configuration control bit*. 0: P2.4 can be analog input pin or digital GPIO pin. 1: P2.4 is pure analog input pin and can't be a digital GPIO pin.
3	P2CON3	R/W	0	P2.3 configuration control bit*. 0: P2.3 can be analog input pin or digital GPIO pin. 1: P2.3 is pure analog input pin and can't be a digital GPIO pin.
2	P2CON2	R/W	0	P2.2 configuration control bit*. 0: P2.2 can be analog input pin or digital GPIO pin. 1: P2.2 is pure analog input pin and can't be a digital GPIO pin.
1	P2CON1	R/W	0	P2.1 configuration control bit*. 0: P2.1 can be analog input pin or digital GPIO pin. 1: P2.1 is pure analog input pin and can't be a digital GPIO pin.
0	P2CON0	R/W	0	P2.0 configuration control bit*. 0: P2.0 can be analog input pin or digital GPIO pin. 1: P2.0 is pure analog input pin and can't be a digital GPIO pin.

* P0CON [7:0], P1CON [7:0], P2CON [7:0], P3CON [7:0] will configure related Port0 , Port1, Port2, Port3 pin as pure analog input pin to avoid current leakage.

P3CON: 0xF01B

Bit	Field	Type	Initial	Description
3	P3CON3	R/W	0	P3.3 configuration control bit*. 0: P3.3 can be analog input pin or digital GPIO pin. 1: P3.3 is pure analog input pin and can't be a digital GPIO pin.
2	P3CON2	R/W	0	P3.2 configuration control bit*. 0: P3.2 can be analog input pin or digital GPIO pin. 1: P3.2 is pure analog input pin and can't be a digital GPIO pin.
1	P3CON1	R/W	0	P3.1 configuration control bit*. 0: P3.1 can be analog input pin or digital GPIO pin. 1: P3.1 is pure analog input pin and can't be a digital

				GPIO pin.
0	P3CON0	R/W	0	P3.0 configuration control bit*. 0: P3.0 can be analog input pin or digital GPIO pin. 1: P3.0 is pure analog input pin and can't be a digital GPIO pin.

* P0CON [7:0], P1CON [7:0], P2CON [7:0], P3CON [7:0] will configure related Port0 , Port1, Port2, Port3 pin as pure analog input pin to avoid current leakage.

12 External Interrupt

INT0, INT1 and INT2 are external interrupt trigger sources. Build in edge trigger configuration function and edge direction is selected by PEDGE and PEDGE1 register. When both external interrupt (EX0/EX1/EX2) and global interrupt (EAL) are enabled, the external interrupt request flag (IE0/IE1/IE2) will be set to "1" as edge trigger event occurs. The program counter will jump to the interrupt vector (ORG 0x0003/0x0013/0x0023) and execute interrupt service routine. Interrupt request flag will be cleared by hardware before

12.1 External Interrupt Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PEDGE	-	-	EX2G1	EX2G0	EX1G1	EX1G0	EX0G1	EX0G0
IEN0	-	-	-	EX2	ET1	EX1	ET0	EX0
TCON	TF1	TR1	TF0	TR0	IE1	-	IE0	-
IRCON	-	-	-	-	-	-	-	IE2

PEDGE Register (0x8F)

Bit	Field	Type	Initial	Description
7..6	Reserved	R	0	
5..4	EX2G[1:0]	R/W	10	External interrupt 2 trigger edge control register. 00: Reserved. 01: Rising edge trigger. 10: Falling edge trigger (default) 11: Both rising and falling edge trigger
3..2	EX1G[1:0]	R/W	10	External interrupt 1 trigger edge control register. 00: Reserved. 01: Rising edge trigger. 10: Falling edge trigger (default) 11: Both rising and falling edge trigger

1..0	EX0G[1:0]	R/W	10	External interrupt 0 trigger edge control register. 00: Reserved. 01: Rising edge trigger. 10: Falling edge trigger (default) 11: Both rising and falling edge trigger

IEN0 Register (0xA8)

Bit	Field	Type	Initial	Description
7	EAL	R/W	0	Enable all interrupt control bit. 0: Disable all interrupt function. 1: Enable all interrupt function.
6..5	Reserved	R	0	
4	EX2	R/W	0	External P0.7 interrupt (INT2) control bit. 0: Disable INT2 interrupt function. 1: Enable INT2 interrupt function.
2	EX1	R/W	0	External P0.6 interrupt (INT1) control bit. 0: Disable INT1 interrupt function. 1: Enable INT1 interrupt function.
0	EX0	R/W	0	External P0.5 interrupt (INT0) control bit. 0: Disable INT0 interrupt function. 1: Enable INT0 interrupt function.

TCON Register (0x88)

Bit	Field	Type	Initial	Description
3	IE1	R/W	0	External P0.6 interrupt (INT1) request flag 0: None INT1 interrupt request. 1: INT1 interrupt request.
1	IE0	R/W	0	External P0.5 interrupt (INT0) request flag 0: None INT0 interrupt request. 1: INT0 interrupt request.
Else				Refer to other chapter(s)

IRCON Register (0xA5)

Bit	Field	Type	Initial	Description

7..1	Reserved	R	0	
0	IE2	R/W	0	External P0.7 interrupt (INT2) request flag 0: None INT2 interrupt request. 1: INT2 interrupt request.

13 Half VDD Generator

Half VDD generator has supplied voltage level of 0.5*VDD. The bias generator is enabled by BIASEN bit. When half VDD generator enables (BIASEN=1), the reference voltage could be measured from specific IO controlled by COMx.

13.1 Half VDD Generator Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BIASCTL	BIASEN	-	-	-	-	-	-	-
COMCTL	COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0

BIASCTL Register (0xD6)

Bit	Field	Type	Initial	Description
7..6	BIASEN	R/W	0	Half VDD generator control bit 0: Disable half VDD generator control bit. 1: Enable half VDD generator control bit.
6..0	Reserved	R	0	

COMCTL Register (0xD7)

Bit	Field	Type	Initial	Description
7	COM7	R/W	0	Half VDD output control bit. 0: Disable COM7 outputs half VDD reference voltage. 1: Enable COM7 outputs half VDD reference voltage.
6	COM6	R/W	0	Half VDD output control bit. 0: Disable COM6 outputs half VDD reference voltage. 1: Enable COM6 outputs half VDD reference voltage.
5..0				et cetera

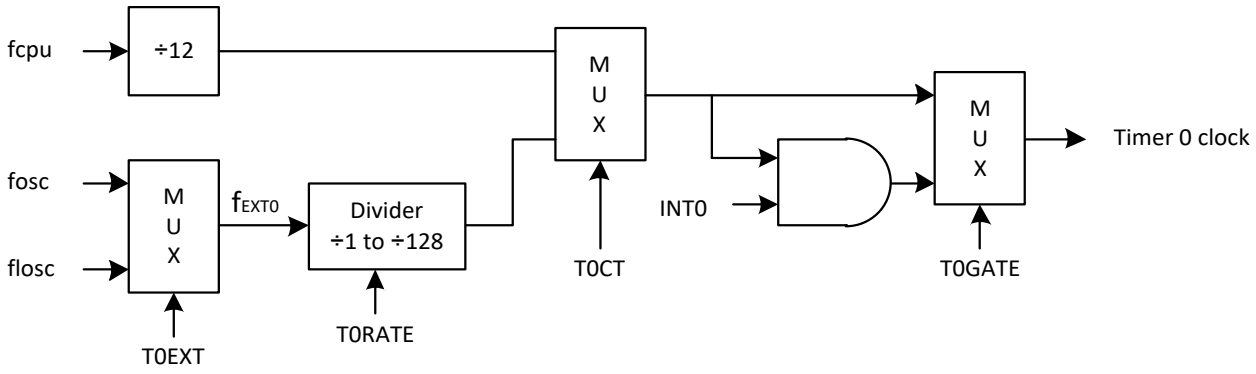
14 Timer 0 and Timer 1

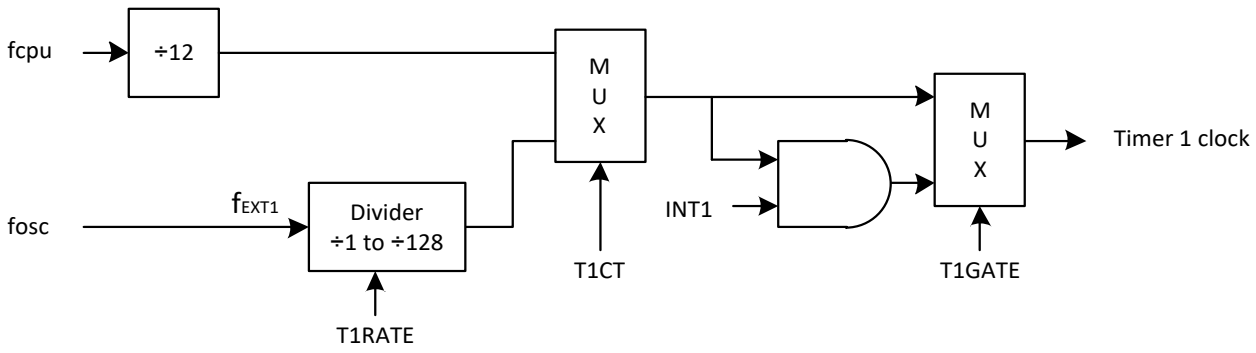
Timer 0 and Timer 1 are two independent binary up timers. Timer 0 has four different operation modes: (1) 13-bit up counting timer, (2) 16-bit up counting timer, (3) 8-bit up counting timer with specified reload value support, and (4) separated two 8-bit up counting timer. By contrast, Timer 1 has only mode 0 to mode 2 which are same as Timer 0. Timer 0 and Timer 1 respectively support ET0 and ET1 interrupt function.

When Timer 0 clock source is flosc and STWK=1, Timer 0 can work in stop mode and waked up from stop mode by Timer 0 interrupt.

14.1 Timer 0 and Timer 1 Clock Selection

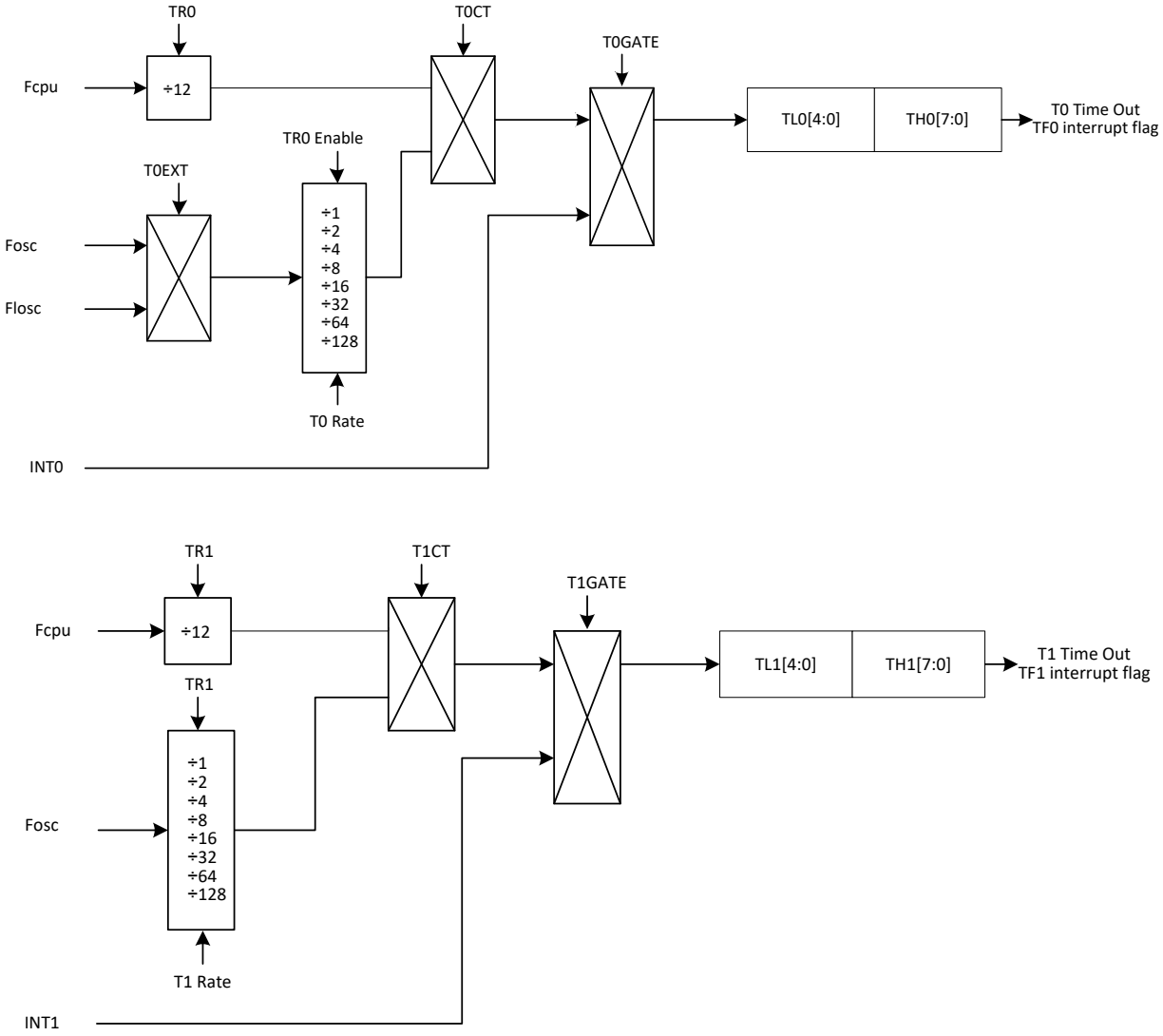
The figures below illustrate the clock selection circuit of Timer 0 and Timer 1. Timer 0 has three clock sources selection: fcpu, fosc, and flosc. All clock sources can be gated (pause) by INT0 pin if T0GATE is applied. Timer 1 clock sources selection: fcpu and fosc. All clock sources can be gated (pause) by INT1 pin if T1GATE is applied. Overall, the major difference between the two timers is that Timer 0 additionally supports flosc clock source (low speed clock).





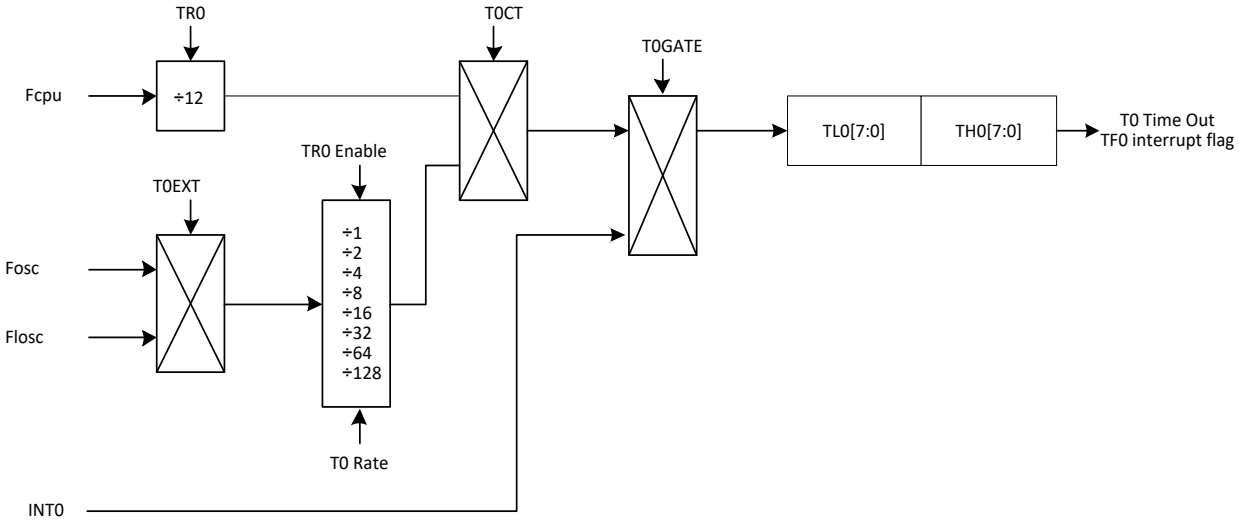
14.2 Mode 0: 13-bit Up Counting Timer

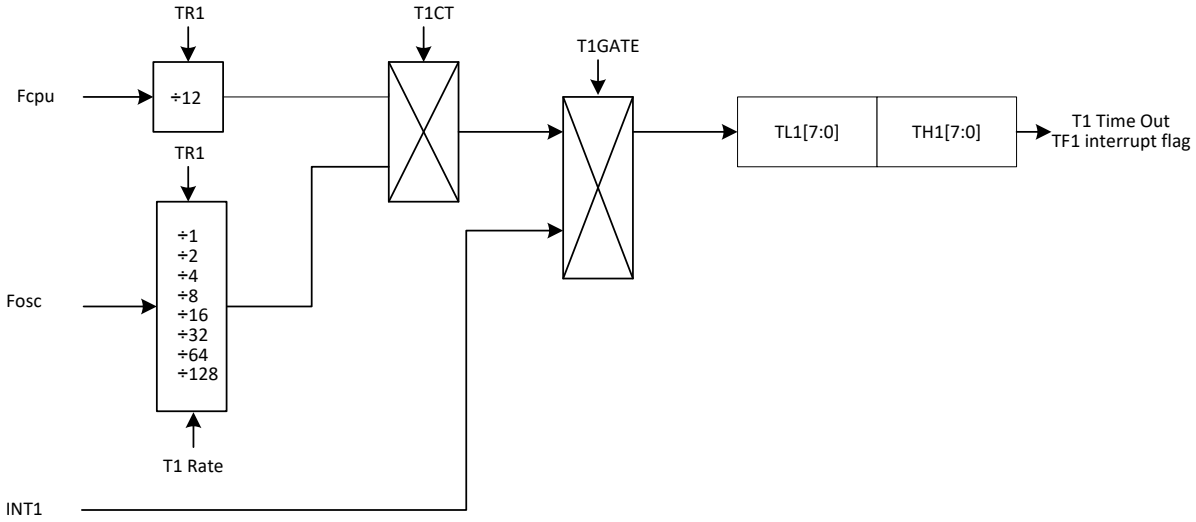
Timer 0 and Timer 1 in mode 0 is a 13-bit up counting timer (the upper 3 bits of TL0 is suspended). Once the timer's counter is overflow (counts from 0xFF1F to 0x0000), TF0/TF1 flag would be issued immediately. This flag is readable by firmware if ET0/ET1 does not apply, or can be handled by interrupt controller if ET0/ET1 is applied.



14.3 Mode 1: 16-bit Up Counting Timer

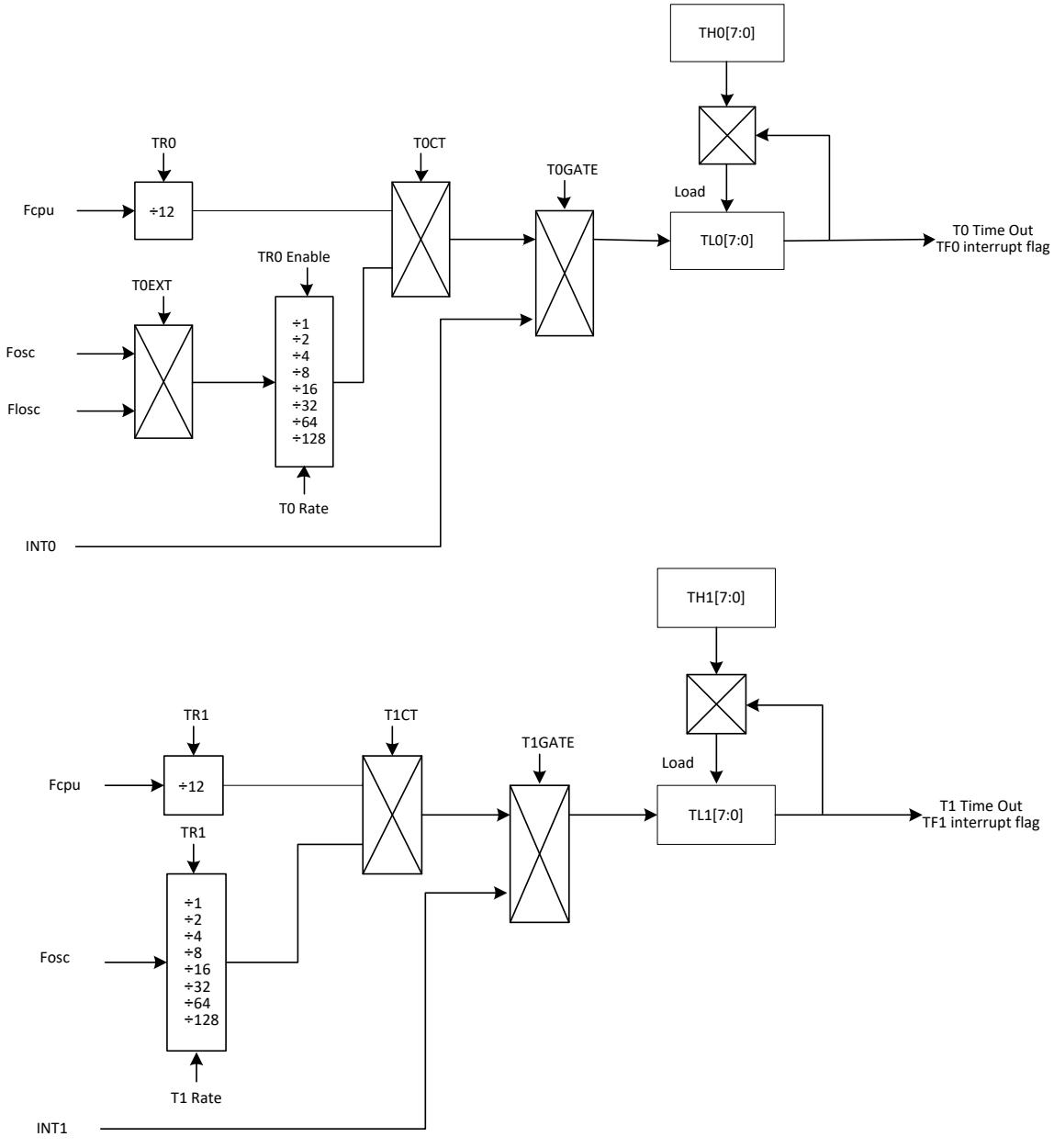
Timer 0 and Timer 1 in mode 1 is a 16-bit up counting timer. Once the timer's counter overflow is occurred (from 0xFFFF to 0x0000), TF0/TF1 would be issued which is readable by firmware or can be handled by interrupt controller (if ET0/ET1 applied).





14.4 Mode 2: 8-bit Up Counting Timer with Specified Reload Value Support

Timer 0 and Timer 1 in mode 2 is an 8-bit up counting timer (TL0/TL1) with a specifiable reload value. An overflow event (TL0/TL1 counts from 0xFF to 0x00) issues its TF0/TF1 flag for firmware or interrupt controller; meanwhile, the timer duplicates TH0/TH1 value to TL0/TL1 register in the same time. As a result, the timer is actually counts from 0xFF to the value of TH0/TH1.

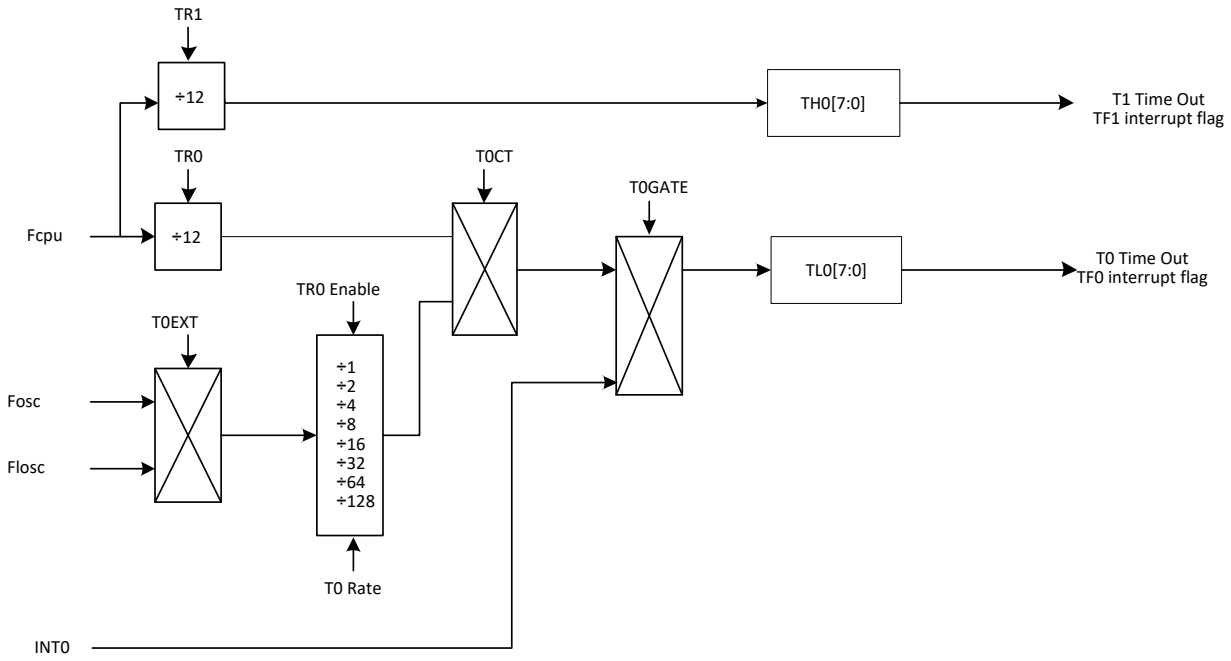


14.5 Mode 3 (Timer 0 only): Separated Two 8-bit Up Counting Timer

Mode 3 treats TH0 and TL0 as two separated 8-bit timers. TL0 is an 8-bit up counting timer with RTC support or two clock sources selection (fcpu and fosc), whereas TH0 clock source is fixed at fcpu/12. Only TL0 clock source can be gated (pause) by INT0 pin if T0GATE is applied.

In this mode TL0 counter is enabled by TR0, and its overflow signal is reflected in TF0 flag. TH0 counter is controlled by TR1, and TF1 flag is also occupied by TH0 overflow signal.

Timer 1 cannot issue any overflow event in this situation, and it can be seen as a self-counting timer without flag support.



14.6 Timer 0 and Timer 1 Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IEN0	EAL	-	-	EX2	ET1	EX1	ET0	EX0
TCON	TF1	TR1	TF0	TR0	IE1	-	IE0	-
TCON0	T0EXT	T0RATE2	T0RATE1	T0RATE0	-	T1RATE2	T1RATE1	T1RATE0
TMOD	T1GATE	T1CT	T1M1	T1M0	T0GATE	T0CT	T0M1	T0M0
TH0	TH07	TH06	TH05	TH04	TH03	TH02	TH01	TH00
TL0	TL07	TL06	TL05	TL04	TL03	TL02	TL01	TL00
TH1	TH17	TH16	TH15	TH14	TH13	TH12	TH11	TH10
TL1	TL17	TL16	TL15	TL14	TL13	TL12	TL11	TL10

IEN0 Register (0xA8)

Bit	Field	Type	Initial	Description
7	EAL	R/W	0	Interrupts enable. Refer to Chapter Interrupt
3	ET1	R/W	0	Timer 1 interrupt 0: Disable 1: Enable
1	ET0	R/W	0	Timer 0 interrupt 0: Disable 1: Enable
Else				Refer to other chapter(s)

TH0 / TH1 Registers (TH0: 0x8C, TH1: 0x8D)

Bit	Field	Type	Initial	Description
7..0	TH0/TH1	R/W	0x00	High byte of Timer 0 and Timer 1 counter

TL0 / TL1 Register (TL0: 0x8A, TL1: 0x8B)

Bit	Field	Type	Initial	Description
7..0	TL0/TL1	R/W	0x00	Low byte of Timer 0 and Timer 1 counter

TCON Register (0x88)

Bit	Field	Type	Initial	Description
7	TF1	R/W	0	Timer 1 overflow event 0: Timer 1 does not have any overflow event 1: Timer 1 has overflowed This bit can be cleared automatically by interrupt handler, or manually by firmware
6	TR1	R/W	0	Timer 1 function 0: Disable 1: Enable
5	TF0	R/W	0	Timer 0 overflow event 0: Timer 0 does not have any overflow event 1: Timer 0 has overflowed This bit can be cleared automatically by interrupt handler, or manually by firmware
4	TR0	R/W	0	Timer 0 function 0: Disable 1: Enable
Else				Refer to other chapter(s)

TCON0 Register (0x8E)

Bit	Field	Type	Initial	Description
7	T0EXT	R/W	0	Timer 0 f_{EXT0} clock source selection. 0: fosc 1: flosc
6..4	T0RATE[2:0]	R/W	000	Clock divider of Timer 0 external clock source*(1) 000: f_{EXT0} / 128 001: f_{EXT0} / 64 010: f_{EXT0} / 32 011: f_{EXT0} / 16 100: f_{EXT0} / 8 101: f_{EXT0} / 4 110: f_{EXT0} / 2 111: f_{EXT0} / 1
3	Reserved	R	0	
2..0	T1RATE[2:0]	R/W	000	Clock divider of Timer 1 external clock source*(2) 000: f_{EXT1} / 128 001: f_{EXT1} / 64 010: f_{EXT1} / 32 011: f_{EXT1} / 16 100: f_{EXT1} / 8 101: f_{EXT1} / 4 110: f_{EXT1} / 2 111: f_{EXT1} / 1

*(1) f_{EXT0} = fosc or flosc.

*(2) f_{EXT1} = fosc.

TMOD Register (0x89)

Bit	Field	Type	Initial	Description
7	T1GATE	R/W	0	Timer 1 gate control mode 0: Disable 1: Enable, Timer 1 clock source is gated by INT1
6	T1CT	R/W	0	Timer 1 clock source selection 0: f_{Timer1} = fcpu / 12 1: $f_{Timer\ 1}$ = f_{EXT1} / T1RATE (refer to T1RATE) *(1)
5..4	T1M[1:0]	R/W	00	Timer 1 operation mode 00: 13-bit up counting timer 01: 16-bit up counting timer 10: 8-bit up counting timer with reload support 11: Reserved
3	T0GATE	R/W	0	Timer 0 gate control mode 0: Disable 1: Enable, Timer 0 clock source is gated by reserved
2	T0CT	R/W	0	Timer 0 clock source selection 0: f_{Timer0} = fcpu / 12 1: f_{Timer0} = f_{EXT0} / T0RATE (refer to T0RATE) *(2)
1..0	T0M[1:0]	R/W	00	Timer 0 operation mode 00: 13-bit up counting timer 01: 16-bit up counting timer 10: 8-bit up counting timer with reload support 11: Separated two 8-bit up counting timer

*(1) f_{EXT1} = fosc.

*(2) f_{EXT0} = fosc or flosc.

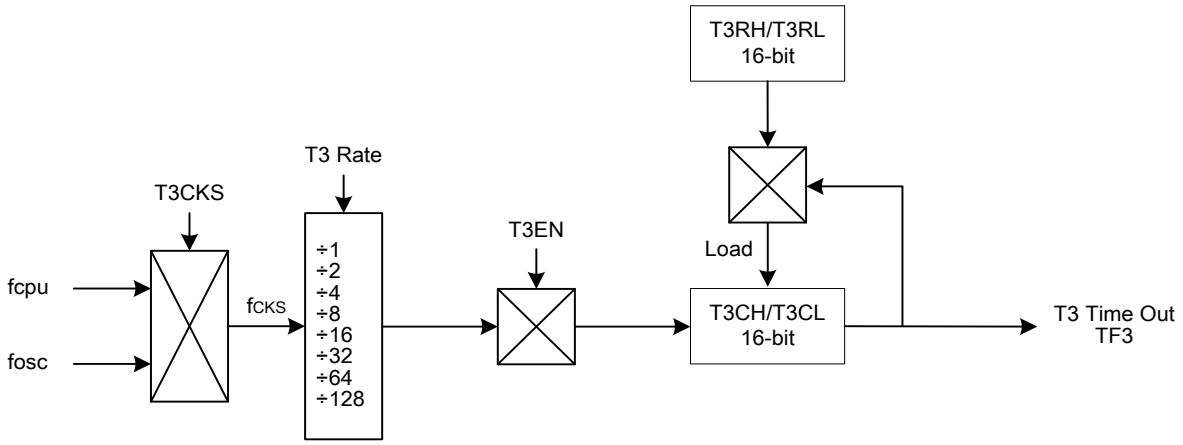
15 Timer 3

Timer 3 has 16-bit up counting timer with specified reload value support. An overflow event (T3CH/T3CL counts from 0xFFFF to 0x0000) issues its TF3 flag for firmware or interrupt controller; meanwhile, the timer duplicates T3RH/T3RL value to T3CH/T3CL register in the same time. As a result, the timer is actually counts from 0xFFFF to the value of T3RH/T3RL.

The Timer 3 build in IDLE Mode wake-up function if interrupt enable. When timer overflow occurs (counts from 0xFFFF to 0x0000), TF3 would be issued immediately which can read/write by firmware. Timer 3 interrupt function is controlled by ET3.

15.1 Timer 3 Clock Selection

Timer 3 has two clock source options: fcpu, fosc. Timer 3 selects the clock source via the T3CKS bit and selects the frequency division by the T3RATE [2:0] bits. The Clock division range is /1 ~ /128.



15.2 Timer 3 Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T3M	T3EN	T3RATE2	T3RATE1	T3RATE0	T3CKS	-	-	-
T3CH	T3C15	T3C14	T3C13	T3C12	T3C11	T3C10	T3C9	T3C8
T3CL	T3C7	T3C6	T3C5	T3C4	T3C3	T3C2	T3C1	T3C0
T3RH	T3R15	T3R14	T3R13	T3R12	T3R11	T3R10	T3R9	T3R8
T3RL	T3R7	T3R6	T3R5	T3R4	T3R3	T3R2	T3R1	T3R0
IEN0	EAL	-	-	EX2	ET1	EX1	ET0	EX0
IEN2	-	-	ET3	-	EPW2	EPW1	-	EADC
IRCON2	-	-	TF3	-	PW2F	PW1F	-	ADCF

T3M Register (0xB1)

Bit	Field	Type	Initial	Description
7	T3EN	R/W	0	Timer 3 function 0: Disable 1: Enable
6..4	T3RATE[2:0]	R/W	000	Clock divider of Timer 3 f_{CKS} clock source 000: f_{CKS} / 128 001: f_{CKS} / 64 010: f_{CKS} / 32 011: f_{CKS} / 16 100: f_{CKS} / 8 101: f_{CKS} / 4 110: f_{CKS} / 2 111: f_{CKS} / 1
3	T3CKS	R/W	0	Timer 3 f_{CKS} clock source selection. 0: fcpu 1: fosc
2..0	Reserved	R/W	000	

T3CH/T3CL Registers (T3CH: 0xB3, T3CL: 0xB2)

Bit	Field	Type	Initial	Description
7..0	T3CH/L	R/W	0x00	16-bit Timer 3 counter buffer.

T3RH/T3RL Registers (T3RH: 0xB5, T3RL: 0xB4)

Bit	Field	Type	Initial	Description
7..0	T3RH/L	R/W	0x00	16-bit Timer 3 counter reload buffer.

IEN0 Register (0xA8)

Bit	Field	Type	Initial	Description
7	EAL	R/W	0	Interrupts enable. Refer to Chapter Interrupt
Else				Refer to other chapter(s)

IEN2 Register (0xC8)

Bit	Field	Type	Initial	Description
5	ET3	R/W	0	Timer 3 interrupt control bit. 0: Disable 1: Enable
Else				Refer to other chapter(s)

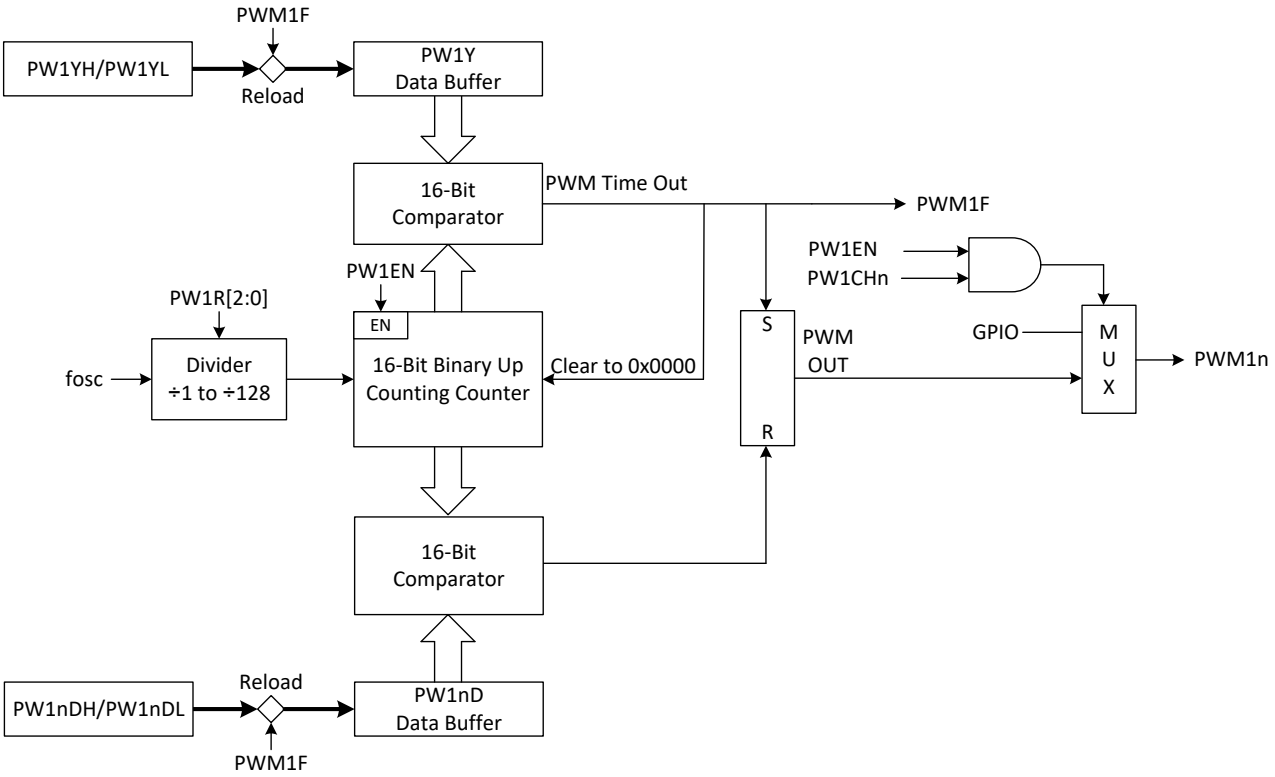
IRCON2 Register (0xA6)

Bit	Field	Type	Initial	Description
5	TF3	R/W	0	Timer 3 overflow event 0: Timer 3 does not have any overflow event 1: Timer 3 has overflowed This bit can be cleared automatically by interrupt handler, or manually by firmware
Else				Refer to other chapter(s)

16 PWM1

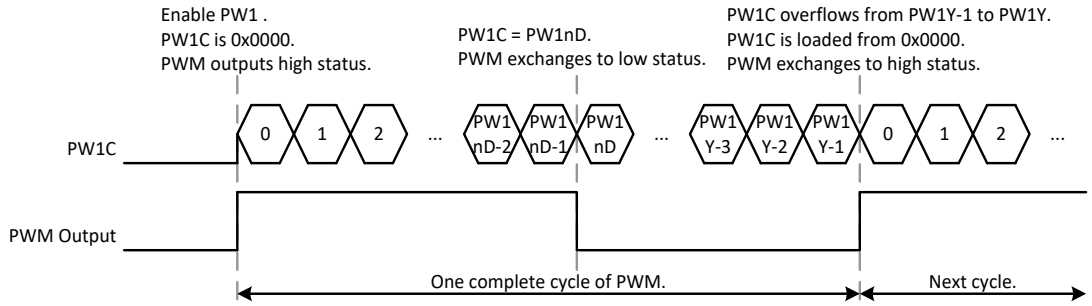
The PW1 timer is a 16-bit up counting timer and supports 4-channel general PWM function. By the counter reaches the up-boundary value (PW1Y), it clears its counter and triggers an interrupt signal. PWM's duty cycle is controlled by PW10D~PW13D register. Each PWM channel has its own duty control.

The PW1 function has 4 programmable channels shared with GPIO pins and controlled by PW1CH[3:0] bits. The output operation must be through enabled each bit/channel of PW1CH[3:0] bits. The enabled PWM channel exchanges from GPIO to PWM output. When the PW1CH[3:0] bits disables, the PWM channel returns to last status of GPIO mode. The PW1 timer builds in IDLE Mode wake-up function if interrupt enable. When timer overflow occurs (counts from PW1Y-1 to PW1Y), PW1F would be issued immediately which can read/write by firmware. PW1 interrupt function is controlled by EPW1.



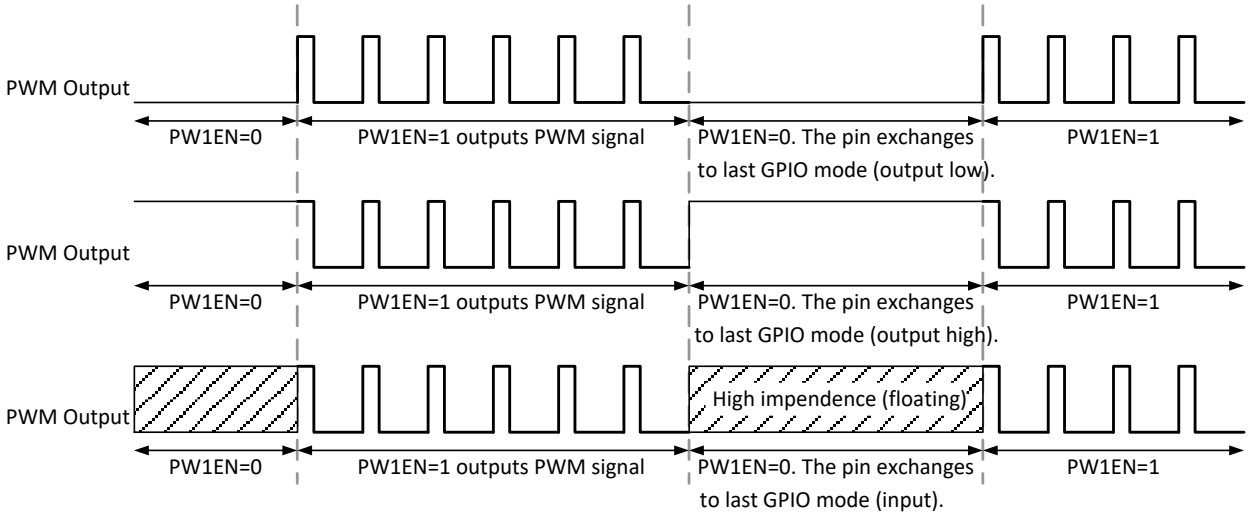
16.1 General PWM

PW1 timer builds in PWM function controlled by PW1EN register and PW1CH bits. PWM10 - PWM13 are output pins. Those output pins are shared with GPIO pin controlled by PW1CH[3:0] bits. When output PWM function, we must be set PW1EN =1. When PWM output signal synchronize finishes, the PWM channel exchanges from GPIO to PWM output. When PW1EN = 0, the PWM channel returns to GPIO mode and last status. PWM signal is generated from the result of PW1Y and PW1nD comparison combination. When PW1C starts to count or returns to 0x0000, the PWM outputs high status which is the PWM initial status. PW1C is loaded new data from PW1Y register to decide PWM cycle and resolution. PW1C keeps counting, and the system compares PW1C and PW1nD. When PW1C=PW1nD, the PWM output status exchanges to low and PW1C keeps counting. When PW1 timer overflow occurs (PW1Y-1 to 0x0000), and one cycle of PWM signal finishes. PW1C is reloaded from 0x0000 automatically, and PWM output status exchanges to high for next cycle. PW1nD decides the high duty duration, and PW1Y decides the resolution and cycle of PWM. PW1nD can't be larger than PW1Y, or the PWM signal is error. PWM clock source is fosc, PW1RATE[2:0] bits: 000 = fosc/128, 001 = fosc/64, 010 = fosc/32, 011 = fosc/16, 100 = fosc/8, 101 = fosc/4, 110 = fosc/2, 111 = fosc/1.



PWM Period = PW1Y

PWM duty = PW1nD / PW1Y



16.2 PWM1 Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW1M	PW1EN	PW1R2	PW1R1	PW1R0	-	-	-	-
PW1CH	-	-	-	-	PW1CH3	PW1CH2	PW1CH1	PW1CH0
PW1YH	PW1Y15	PW1Y14	PW1Y13	PW1Y12	PW1Y11	PW1Y10	PW1Y9	PW1Y8
PW1YL	PW1Y7	PW1Y6	PW1Y5	PW1Y4	PW1Y3	PW1Y2	PW1Y1	PW1Y0
PW10DH	PW10D15	PW10D14	PW10D13	PW10D12	PW10D11	PW10D10	PW10D9	PW10D8
PW10DL	PW10D7	PW10D6	PW10D5	PW10D4	PW10D3	PW10D2	PW10D1	PW10D0
PW11DH	PW11D15	PW11D14	PW11D13	PW11D12	PW11D11	PW11D10	PW11D9	PW11D8
PW11DL	PW11D7	PW11D6	PW11D5	PW11D4	PW11D3	PW11D2	PW11D1	PW11D0
PW12DH	PW12D15	PW12D14	PW12D13	PW12D12	PW12D11	PW12D10	PW12D9	PW12D8
PW12DL	PW12D7	PW12D6	PW12D5	PW12D4	PW12D3	PW12D2	PW12D1	PW12D0
PW13DH	PW13D15	PW13D14	PW13D13	PW13D12	PW13D11	PW13D10	PW13D9	PW13D8
PW13DL	PW13D7	PW13D6	PW13D5	PW13D4	PW13D3	PW13D2	PW13D1	PW13D0
IEN0	EAL	-	-	EX2	ET1	EX1	ET0	EX0
IEN2	-	-	ET3	-	EPW2	EPW1	-	EADC
IRCON2	-	-	TF3	-	PW2F	PW1F	-	ADCF

PW1M Registers (PW1M: 0xC9)

Bit	Field	Type	Initial	Description
7	PW1EN	R/W	0	PW1 function 0: Disable 1: Enable*
6..4	PW1R[2:0]	R/W	000	PWM timer clock source 000: fosc / 128 001: fosc / 64 010: fosc / 32 011: fosc / 16 100: fosc / 8 101: fosc / 4 110: fosc / 2 111: fosc / 1
3..0	Reserved	R	0	

* When the period is setting 0x0000, after PWM is set enable bit, the PWM will stop and the period can't update.

PW1CH Registers (PW1CH: 0xCA)

Bit	Field	Type	Initial	Description
7..4	Reserved	R	0	
3	PW1CH3	R/W	0	PWM13 shared-pin control 0: GPIO 1: PWM output (shared with P2.4)
2	PW1CH2	R/W	0	PWM12 shared-pin control 0: GPIO 1: PWM output (shared with P0.7)
1	PW1CH1	R/W	0	PWM11 shared-pin control 0: GPIO 1: PWM output (shared with P0.1)
0	PW1CH0	R/W	0	PWM10 shared-pin control 0: GPIO 1: PWM output (shared with P0.0)

PW1YH/PW1YL Registers (PW1YH: 0xF071, PW1YL: 0xF070)

Bit	Field	Type	Initial	Description
7..0	PW1YH/L	R/W	0x00	16-bit PWM1 period control*.

* The period configuration must be setup completely before starting PWM function.

PW10DH/PW10DL Registers (PW10DH: 0xF079, PW10DL: 0xF078)

Bit	Field	Type	Initial	Description
7..0	PW10DH/L	R/W	0x00	16-bit PWM1 duty control.

PW11DH/PW11DL Registers (PW11DH: 0xF07B, PW11DL: 0xF07A)

Bit	Field	Type	Initial	Description
7..0	PW11DH/L	R/W	0x00	16-bit PWM1 duty control.

PW12DH/PW12DL Registers (PW12DH: 0xF07D, PW12DL: 0xF07C)

Bit	Field	Type	Initial	Description
7..0	PW12DH/L	R/W	0x00	16-bit PWM1 duty control.

PW13DH/PW13DL Registers (PW13DH: 0xF07F, PW13DL: 0xF07E)

Bit	Field	Type	Initial	Description
7..0	PW13DH/L	R/W	0x00	16-bit PWM1 duty control.

IEN0 Register (0xA8)

Bit	Field	Type	Initial	Description
7	EAL	R/W	0	Interrupts enable. Refer to Chapter Interrupt
Else				Refer to other chapter(s)

IEN2 Register (0xC8)

Bit	Field	Type	Initial	Description
2	EPW1	R/W	0	PWM1 interrupt control bit. 0 = Disable PWM1 interrupt function. 1 = Enable PWM1 interrupt function.
Else				Refer to other chapter(s)

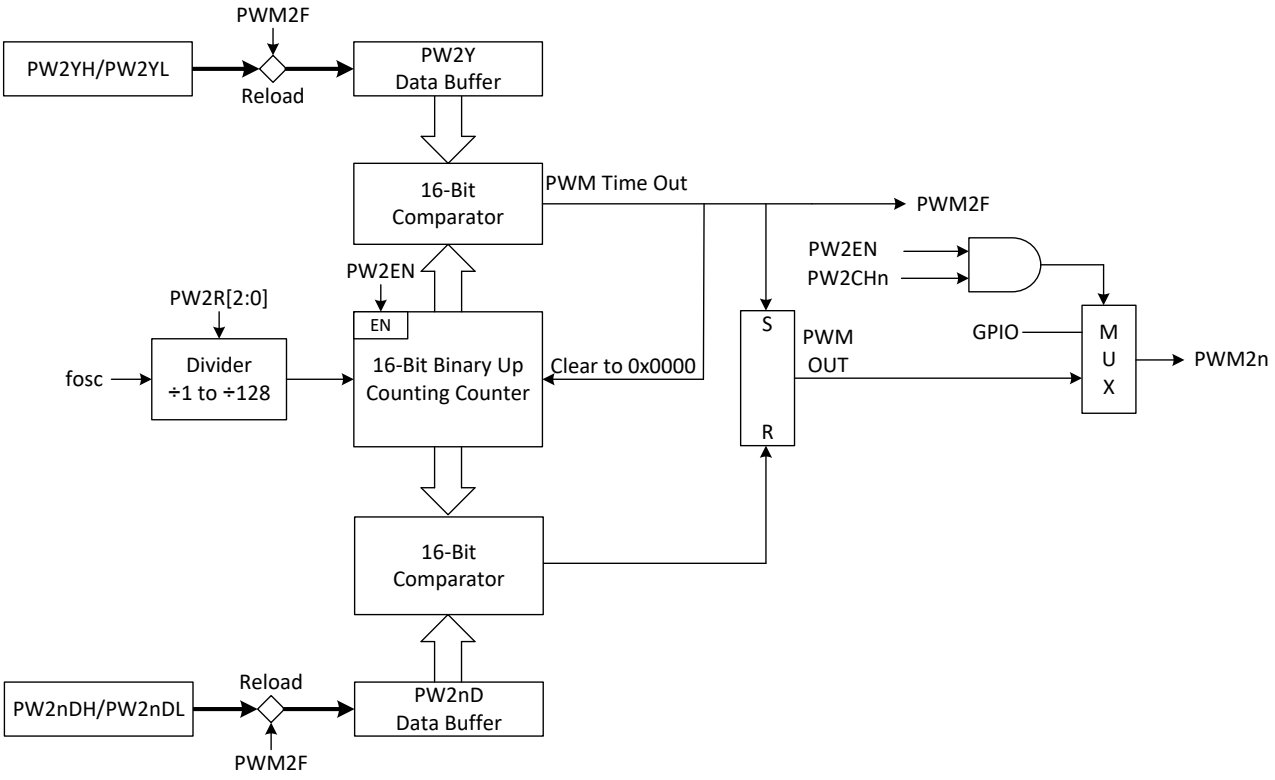
IRCON2 Register (0xA6)

Bit	Field	Type	Initial	Description
2	PW1F	R/W	0	PWM1 interrupt request flag. 0: None PWM1 interrupt request 1: PWM1 interrupt request.
Else				Refer to other chapter(s)

17 PWM2

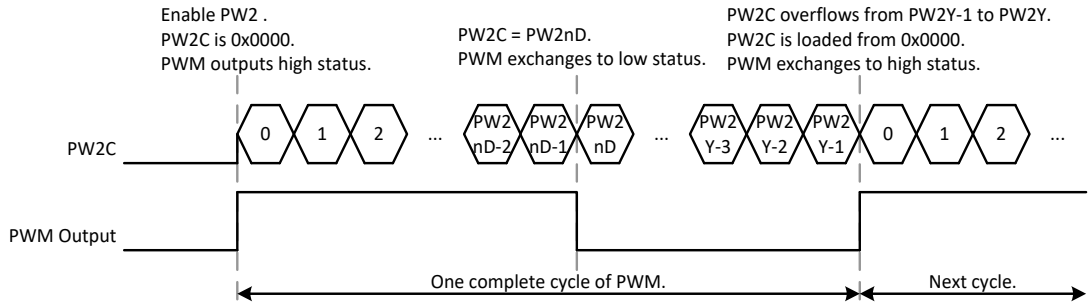
The PW2 timer is a 16-bit up counting timer and supports 4-channel general PWM function. By the counter reaches the up-boundary value (PW2Y), it clears its counter and triggers an interrupt signal. PWM's duty cycle is controlled by PW20D~PW23D register. Each PWM channel has its own duty control.

The PW2 function has 4 programmable channels shared with GPIO pins and controlled by PW2CH[3:0] bits. The output operation must be through enabled each bit/channel of PW2CH[3:0] bits. The enabled PWM channel exchanges from GPIO to PWM output. When the PW2CH[3:0] bits disables, the PWM channel returns to last status of GPIO mode. The PW2 timer builds in IDLE Mode wake-up function if interrupt enable. When timer overflow occurs (counts from PW2Y-1 to PW2Y), PW2F would be issued immediately which can read/write by firmware. PW2 interrupt function is controlled by EPW2.



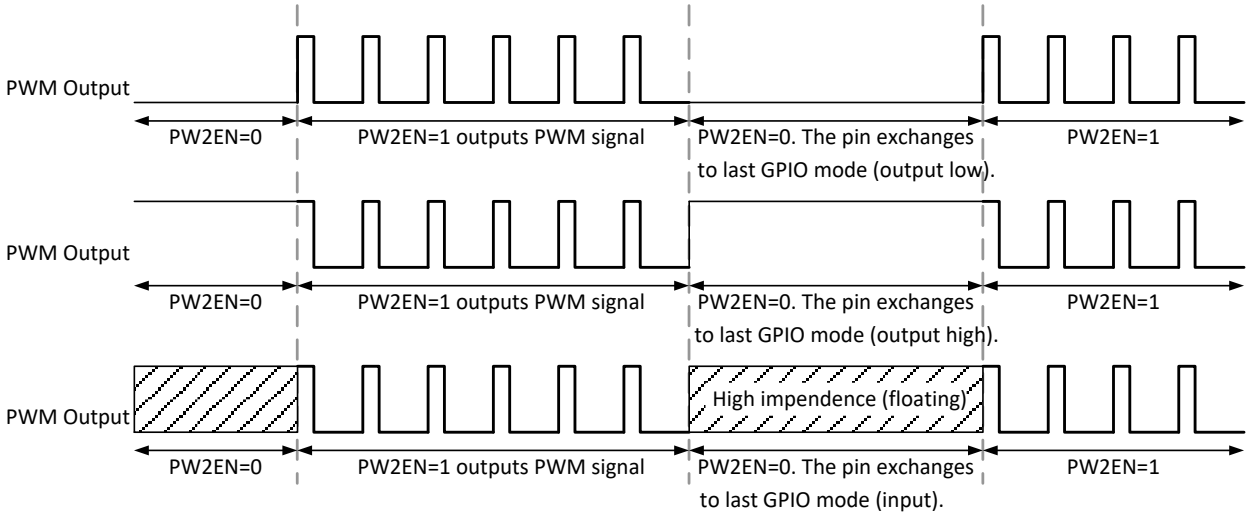
17.1 General PWM

PW2 timer builds in PWM function controlled by PW2EN register and PW2CH bits. PWM20 – PWM23 are output pins. Those output pins are shared with GPIO pin controlled by PW2CH[3:0] bits. When output PWM function, we must be set PW2EN =1. When PWM output signal synchronize finishes, the PWM channel exchanges from GPIO to PWM output. When PW2EN = 0, the PWM channel returns to GPIO mode and last status. PWM signal is generated from the result of PW2Y and PW2nD comparison combination. When PW2C starts to count or returns to 0x0000, the PWM outputs high status which is the PWM initial status. PW2C is loaded new data from PW2Y register to decide PWM cycle and resolution. PW2C keeps counting, and the system compares PW2C and PW2nD. When PW2C=PW2nD, the PWM output status exchanges to low and PW2C keeps counting. When PW2 timer overflow occurs (PW2Y-1 to 0x0000), and one cycle of PWM signal finishes. PW2C is reloaded from 0x0000 automatically, and PWM output status exchanges to high for next cycle. PW2nD decides the high duty duration, and PW2Y decides the resolution and cycle of PWM. PW2nD can't be larger than PW2Y, or the PWM signal is error. PWM clock source is fosc, PW2RATE[2:0] bits: 000 = fosc/128, 001 = fosc/64, 010 = fosc/32, 011 = fosc/16, 100 = fosc/8, 101 = fosc/4, 110 = fosc/2, 111 = fosc/1.



PWM Period = PW2Y

PWM duty = PW2nD / PW2Y



17.2 PWM2 Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW2M	PW2EN	PW2R2	PW2R1	PW2R0	-	-	-	-
PW2CH	-	-	-	-	PW2CH3	PW2CH2	PW2CH1	PW2CH0
PW2YH	PW2Y15	PW2Y14	PW2Y13	PW2Y12	PW2Y11	PW2Y10	PW2Y9	PW2Y8
PW2YL	PW2Y7	PW2Y6	PW2Y5	PW2Y4	PW2Y3	PW2Y2	PW2Y1	PW2Y0
PW20DH	PW20D15	PW20D14	PW20D13	PW20D12	PW20D11	PW20D10	PW20D9	PW20D8
PW20DL	PW20D7	PW20D6	PW20D5	PW20D4	PW20D3	PW20D2	PW20D1	PW20D0
PW21DH	PW21D15	PW21D14	PW21D13	PW21D12	PW21D11	PW21D10	PW21D9	PW21D8
PW21DL	PW21D7	PW21D6	PW21D5	PW21D4	PW21D3	PW21D2	PW21D1	PW21D0
PW22DH	PW22D15	PW22D14	PW22D13	PW22D12	PW22D11	PW22D10	PW22D9	PW22D8
PW22DL	PW22D7	PW22D6	PW22D5	PW22D4	PW22D3	PW22D2	PW22D1	PW22D0
PW23DH	PW23D15	PW23D14	PW23D13	PW23D12	PW23D11	PW23D10	PW23D9	PW23D8
PW23DL	PW23D7	PW23D6	PW23D5	PW23D4	PW23D3	PW23D2	PW23D1	PW23D0
IEN0	EAL	-	-	EX2	ET1	EX1	ET0	EX0
IEN2	-	-	ET3	-	EPW2	EPW1	-	EADC
IRCON2	-	-	TF3	-	PW2F	PW1F	-	ADCF

PW2M Registers (PW2M: 0xCB)

Bit	Field	Type	Initial	Description
7	PW2EN	R/W	0	PW2 function 0: Disable 1: Enable*
6..4	PW2R[2:0]	R/W	000	PWM timer clock source 000: fosc / 128 001: fosc / 64 010: fosc / 32 011: fosc / 16 100: fosc / 8 101: fosc / 4 110: fosc / 2 111: fosc / 1
3..0	Reserved	R	0	

* When the period is setting 0x0000, after PWM is set enable bit, the PWM will stop and the period can’t update.

PW2CH Registers (PW2CH: 0xCC)

Bit	Field	Type	Initial	Description
7..4	Reserved	R	0	
3	PW2CH3	R/W	0	PWM23 shared-pin control 0: GPIO 1: PWM output (shared with P2.3)
2	PW2CH2	R/W	0	PWM22 shared-pin control 0: GPIO 1: PWM output (shared with P2.2)
1	PW2CH1	R/W	0	PWM21 shared-pin control 0: GPIO 1: PWM output (shared with P2.1)
0	PW2CH0	R/W	0	PWM20 shared-pin control 0: GPIO 1: PWM output (shared with P2.0)

PW2YH/PW2YL Registers (PW2YH: 0xF073, PW2YL: 0xF072)

Bit	Field	Type	Initial	Description
7..0	PW2YH/L	R/W	0x00	16-bit PWM2 period control*.

* The period configuration must be setup completely before starting PWM function (before PW2EN=1).

PW20DH/PW20DL Registers (PW20DH: 0xF089, PW20DL: 0xF088)

Bit	Field	Type	Initial	Description
7..0	PW20DH/L	R/W	0x00	16-bit PWM2 duty control.

PW21DH/PW21DL Registers (PW21DH: 0xF08B, PW21DL: 0xF08A)

Bit	Field	Type	Initial	Description
7..0	PW21DH/L	R/W	0x00	16-bit PWM2 duty control.

PW22DH/PW22DL Registers (PW22DH: 0xF08D, PW22DL: 0xF08C)

Bit	Field	Type	Initial	Description
7..0	PW22DH/L	R/W	0x00	16-bit PWM2 duty control.

PW23DH/PW23DL Registers (PW23DH: 0xF08F, PW23DL: 0xF08E)

Bit	Field	Type	Initial	Description
7..0	PW23DH/L	R/W	0x00	16-bit PWM2 duty control.

IEN0 Register (0xA8)

Bit	Field	Type	Initial	Description
7	EAL	R/W	0	Interrupts enable. Refer to Chapter Interrupt
Else				Refer to other chapter(s)

IEN2 Register (0xC8)

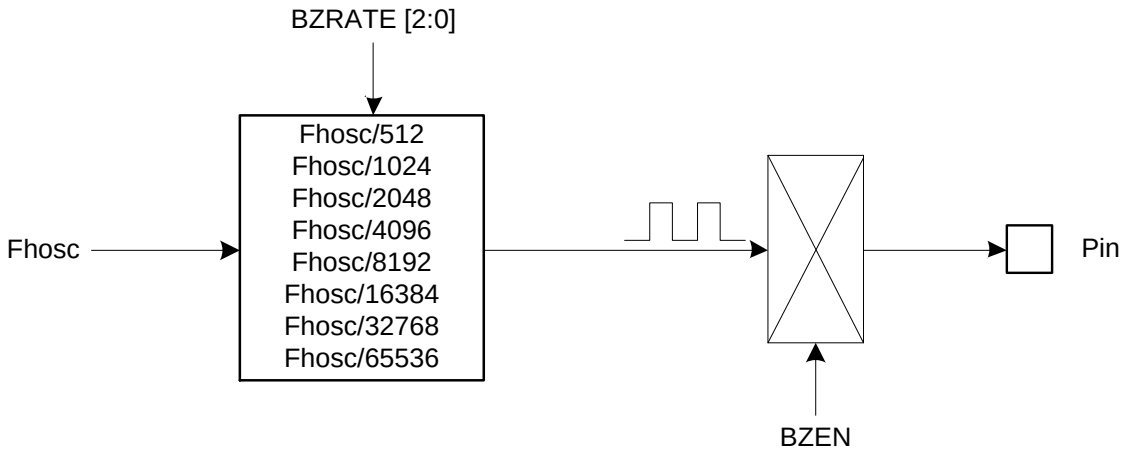
Bit	Field	Type	Initial	Description
3	EPW2	R/W	0	PWM2 interrupt control bit. 0 = Disable PWM2 interrupt function. 1 = Enable PWM2 interrupt function.
Else				Refer to other chapter(s)

IRCON2 Register (0xA6)

Bit	Field	Type	Initial	Description
3	PW2F	R/W	0	PWM2 interrupt request flag. 0: None PWM2 interrupt request 1: PWM2 interrupt request.
Else				Refer to other chapter(s)

18 2K/4K BUZZER GENERATOR

The MCU builds in Buzzer generator to drive external buzzer device. The buzzer generator purpose is to drive 2KHz or 4KHz buzzer. Adjusting buzzer output frequency is through BZM register. The buzzer output pin is shared with GPIO. When BZEN = 1, the pin outputs buzzer carry signal. When BZEN = 0, the pin returns to GPIO last condition (input mode, output high or output low status).



- The buzzer frequency is divided from Fhosc controlled by BZrate bits, and Fcpu decides the buzzer frequency. The selection table is as following.

BZrate [2:0]	Buzzer Rate Division	Buzzer Rate			
		IHRC_32MHz	X'tal = 16MHz	X'tal = 12MHz	X'tal = 4MHz
000	Fhosc/512	64KHz	32KHz	24KHz	8KHz
001	Fhosc/1024	32KHz	16KHz	12KHz	4KHz
010	Fhosc/2048	16KHz	8KHz	6KHz	2KHz
011	Fhosc/4096	8KHz	4KHz	3KHz	1KHz
100	Fhosc/8192	4KHz	2KHz	1.5KHz	500Hz
101	Fhosc/16384	2KHz	1KHz	750Hz	250Hz
110	Fhosc/32768	1KHz	500Hz	375Hz	125Hz
111	Fhosc/65536	500Hz	250Hz	188Hz	63Hz

- The buzzer target frequency is 2KHz and 4KHz. It is important to choice a good Fhosc rate to obtain the correct buzzer frequency. The above table shows 2KHz/4KHz buzzer frequency configurations.

18.1 Buzzer Register

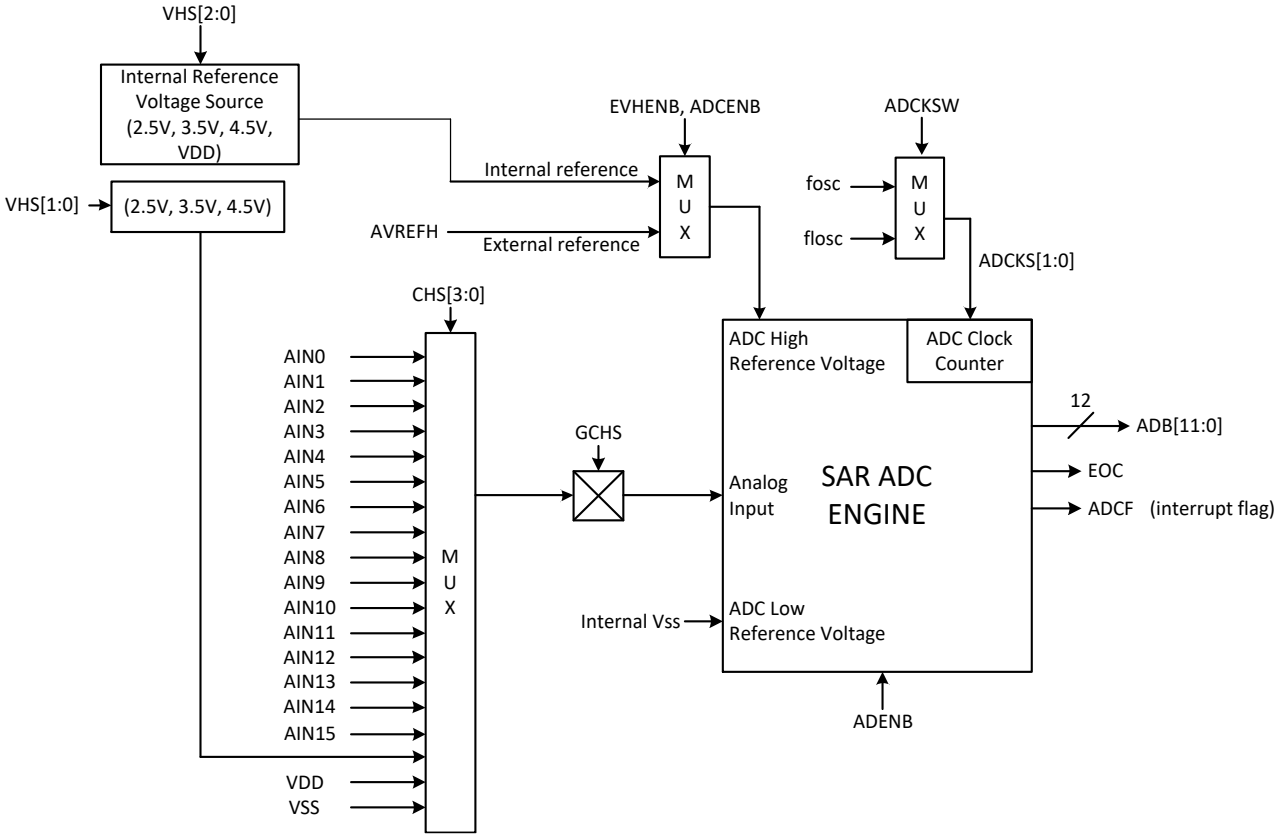
Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BZM	BZEN	BZrate2	BZrate1	BZrate0	-	-	-	-

BZM Register (0xD9)

Bit	Field	Type	Initial	Description
7	BZEN	R/W	0	Buzzer output control bit. 0: Disable BZ output and BZ output pin(P1.4) transfers to I/O last status. 1: Enable BZ output and disable GPIO(P1.4) function.
6..4	BZRATE[2:0]	R/W	000	Buzzer rate control bits. 000: Fhosc/512 001: Fhosc/1024 010: Fhosc/2048 011: Fhosc/4096 100: Fhosc/8192 101: Fhosc/16384 110: Fhosc/32768 111: Fhosc/65536
Else	Reserved	R	0	

19 ADC

The analog to digital converter (ADC) is SAR structure with 16-input sources and up to 4096-step resolution to transfer analog signal into 12-bits digital buffers. The ADC builds in 16-channel input source to measure 16 different analog signal sources. The ADC resolution is 12-bit. The ADC has four clock rates to decide ADC converting rate. The ADC reference high voltage includes 5 sources. Four internal power source including VDD, 4.5V, 3.5V and 2.5V. The other one is external reference voltage input pin from AVREFH pin. The ADC builds in P0CON/P1CON/P2CON registers to set pure analog input pin. After setup ADENB and ADS bits, the ADC starts to convert analog signal to digital data. ADC can work in idle mode. After ADC operating, the system would be waked up from idle mode to normal mode if interrupt enable. When ADC clock source is flosc and STWK=1, ADC can work in stop mode and waked up from stop mode by ADC interrupt.



19.1 Configurations of Operation

These configurations must be setup completely before starting ADC converting. ADC is configured using the following steps:

1. Choose and enable the start of conversion ADC input channel. (By CHS[4:0] bits and GCHS bit)
2. The GPIO mode of ADC input channel must be set as input mode. (By PnM register)
3. The internal pull-up resistor of ADC input channel must be disabled. (By PnUR register)
4. The configuration control bit of ADC input channel must be set. (By PnCON register)
5. Choose ADC high reference voltage. (By VREFH register)
6. Choose ADC Clock Source and Clock Rate. (By ADCKSW and ADCKS[1:0] bits)
7. After setup ADENB bits, the ADC ready to convert analog signal to digital data.

When ADC IP is enabled by ADENB bit, it is necessary to make an ADC start-up by program. Writing a 1 to the ADS bit of register ADM. After setup ADENB and ADS bits, the ADC starts to convert analog signal to digital data. The ADS bit is reset to logic 0 when the conversion is complete. When the conversion is complete, the ADC circuit will set EOC and ADCF bits to “1” and the digital data outputs in ADB and ADR registers. If ADC interrupt function is enabled (EADC = 1), the ADC interrupt request occurs and executes interrupt service routine when ADCF is “1” after ADC converting. Clear ADCF by hardware automatically in interrupt procedure.

19.2 ADC input channel

The ADC builds in 16-channel input source (AIN0 – AIN15) to measure 16 different analog signal sources controlled by CHS[4:0] and GCHS bits. The AIN16 is internal 2.5V or 3.5V or 4.5V input channel. There is no any input pin from outside. In this time ADC reference voltage must be internal VDD and External voltage, not internal 2.5V or 3.5V or 4.5V. AIN16 can be a good battery detector for battery system. To select appropriate internal AVREFH level and compare value, a high performance and cheaper low battery detector is built in the system. AIN17 is VDD channel. AIN18 is VSS channel. Other channel is reserved.

CHS[4:0]	Channel	Pin name	Remark
00000	AIN0	P1.1	-
00001	AIN1	P1.2	-
00010	AIN2	P1.3	-
00011	AIN3	P1.4	-
00100	AIN4	P1.5	-
00101	AIN5	P1.6	-
00110	AIN6	P1.7	-
00111	AIN7	P2.0	-
01000	AIN8	P2.1	-
01001	AIN9	P2.2	-
01010	AIN10	P0.5	-
01011	AIN11	P0.4	-
01100	AIN12	P0.3	-
01101	AIN13	P0.2	-
01110	AIN14	P0.1	-
01111	AIN15	P0.0	-
10000	AIN16	Internal 2.5V or 3.5V or 4.5V	Battery detector channel
10001	AIN17	VDD	-
10010	AIN18	VSS	-
Other	-	-	Reserved

19.2.1 Pin Configuration

ADC input channels are shared with Port0, Port1 and Port2. ADC channel selection is through CHS[4:0] bit. Only one pin of Port0, Port1 and Port2 can be configured as ADC input in the same time. The pins of Port0, Port1 and Port2 configured as ADC input channel must be set input mode, disable internal pull-up and enable P0CON/P1CON/P2CON first by program. After selecting ADC input channel through CHS[4:0], set GCHS bit as “1” to enable ADC channel function.

ADC input pins are shared with digital I/O pins. Connect an analog signal to CMOS digital input pin, especially, the analog signal level is about 1/2 VDD will cause extra current leakage. In the power down mode, the above leakage current will be a big problem. Unfortunately, if users connect more than one analog input signal to Port0, Port1, Port2 will encounter above current leakage situation. Write “1” into PnCON register will configure related pin as pure analog input pin to avoid current leakage.

Note that When ADC pin is general I/O mode, the bit of P0CON/P1CON and P2CON must be set to “0”, or the digital I/O signal would be isolated.

19.3 Reference Voltage

The ADC builds in five high reference voltage source controlled through VREFH register. There are one external voltage source and four internal voltage source (VDD, 4.5V, 3.5V, 2.5V). When EVHENB bit is “1”, ADC reference voltage is external voltage source from AVREFH/P1.1. In the condition, P1.1 GPIO mode must be set as input mode and disable internal pull-up resistor.

If EVHENB bit is “0”, ADC reference high voltage is from internal voltage source selected by VHS[1:0] bits. If VHS[1:0] is “11”, ADC reference high voltage is VDD. If VHS[1:0] is “10”, ADC reference high voltage is 4.5V. If VHS[1:0] is “01”, ADC reference high voltage is 3.5V. If VHS[1:0] is “00”, ADC reference high voltage is 2.5V. The limitation of internal high reference voltage application is VDD can’t below each of internal high voltage level, or the level is equal to VDD. If AIN16 channel is selected as internal 2.5V or 3.5V or 4.5V input channel. There is no any input pin from outside. In this time ADC high reference voltage must be internal VDD or External voltage, not internal 2.5V/3.5V/4.5V.

19.3.1 Signal Format

ADC sampling voltage range is limited by high/low reference voltage. The ADC low reference voltage is VSS. The ADC high reference voltage includes internal VDD/4.5V/3.5V/2.5V and external reference voltage source from P1.1/AVREFH pin controlled by EVHENB bit. ADC reference voltage range limitation is “(ADC high reference voltage - low reference voltage) ≥ 2.0V”. ADC low reference voltage is VSS = 0V. So ADC high reference voltage range is 2.0V to VDD. The range is ADC external high reference voltage range.

- ADC Internal Low Reference Voltage = 0V.
- ADC Internal High Reference Voltage = VDD/4.5V/3.5V/2.5V. (EVHENB=0)
- ADC External High Reference Voltage = 2.0V to VDD. (EVHENB=1)

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

- ADC Low Reference Voltage ≤ ADC Sampled Input Voltage ≤ ADC High Reference Voltage

19.4 Converting Time

The ADC converting time is from ADS=1 (Start to ADC convert) to EOC=1 (End of ADC convert). The converting time duration is depend on ADC clock rate. 12-bit ADC’s converting time is
1/ (ADC clock /4)*16 sec. ADC has two clock sources: fosc or flosc, which is controlled by ADCKSW bit. ADCKS[1:0] bits: 00 = fosc/32 or flosc/32, 01 = fosc/16 or flosc/16, 10 = fosc/2 or flosc/2, 11 = fosc/8 or flosc/8.

The ADC converting time affects ADC performance. If input high rate analog signal, it is necessary to select a high ADC converting rate. If the ADC converting time is slower than analog signal variation rate, the ADC result would be error. So to select a correct ADC clock rate to decide a right ADC converting rate is very important.

$$\text{12 bits ADC conversion time} = \frac{16}{\text{ADC clock rate}/4}$$

When ADCKSW=0:

ADCKS[1:0]	ADC clock rate	fosc = 16MHz		fosc = 32MHz	
		Converting time	Converting rate	Converting time	Converting rate
00	fosc/32	1/(16MHz/32/4)*16 = 128us	7.8125kHz	1/(32MHz/32/4)*16 = 64us	15.625kHz
01	fosc/16	1/(16MHz/16/4)*16 = 64us	15.625kHz	1/(32MHz/16/4)*16 = 32us	31.25kHz
10	fosc/2	1/(16MHz/2/4)*16 = 8us	125kHz	1/(32MHz/2/4)*16 = 4us	250kHz
11	fosc/8	1/(16MHz/8/4)*16 = 32us	31.25kHz	1/(32MHz/8/4)*16 = 16us	62.5kHz

When ACKSW=1:

ADCKS[1:0]	ADC clock rate	flosc = 16KHz		flosc = 32.768KHz	
		Converting time	Converting rate	Converting time	Converting rate
00	flosc/32	1/(16KHz/32/4)*16 = 128ms	7.8125Hz	1/(32.768KHz/32/4)*16 = 62.5ms	16Hz
01	flosc/16	1/(16KHz/16/4)*16 = 64ms	15.625Hz	1/(32.768KHz/16/4)*16 = 31.25ms	32Hz
10	flosc/2	1/(16KHz/2/4)*16 = 8ms	125Hz	1/(32.768KHz/2/4)*16 = 3.90625ms	256Hz
11	flosc/8	1/(16KHz/8/4)*16 = 32ms	31.25Hz	1/(32.768KHz/8/4)*16 = 15.625ms	64Hz

19.5 Data Buffer

ADC data buffer is 12-bit length to store ADC converter result. The high byte is ADB register, and the low-nibble is ADR[3:0] bits. The ADB register is only 8-bit register including bit 4 – bit 11 ADC data. To combine ADB register and the low-nibble of ADR will get full 12-bit ADC data buffer. The ADC data buffer is a read-only register and the initial status is unknown after system reset.

Table 18-1 The AIN input voltage vs. ADB output data

AIN n	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4	ADB3	ADB2	ADB1	ADB0
0/4096*VREFH	0	0	0	0	0	0	0	0	0	0	0	0
1/4096*VREFH	0	0	0	0	0	0	0	0	0	0	0	1
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
4094/4096*VREFH	1	1	1	1	1	1	1	1	1	1	1	0
4095/4096*VREFH	1	1	1	1	1	1	1	1	1	1	1	1

19.6 ADC Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADM	ADENB	ADS	EOC	CHS4	CHS3	CHS2	CHS1	CHS0
ADB	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4
ADR	ADCKSW	GCHS	ADCKS1	ADCKS0	ADB3	ADB2	ADB1	ADB0
VREFH	EVHENB	-	-	-	-	VHS2	VHS1	VHS0
P0CON	P0CON7	P0CON6	P0CON5	P0CON4	P0CON3	P0CON2	P0CON1	P0CON0
P1CON	P1CON7	P1CON6	P1CON5	P1CON4	P1CON3	P1CON2	P1CON1	P1CON0
P2CON	-	-	-	P2CON4	P2CON3	P2CON2	P2CON1	P2CON0
IEN0	EAL	-	-	EX2	ET1	EX1	ET0	EX0
IEN2	-	-	ET3	-	EPW2	EPW1	-	EADC
IRCON2	-	-	TF3	-	PW2F	PW1F	-	ADCF

ADM Register (0xD2)

Bit	Field	Type	Initial	Description
7	ADENB	R/W	0	ADC control bit. In stop mode, disable ADC to reduce power consumption. 0: Disable 1: Enable
6	ADS	R/W	0	ADC conversion control Write 1: Start ADC conversion (automatically cleared by the end of conversion)
5	EOC	R/W	0	ADC status bit. 0: ADC progressing 1: End of conversion (automatically set by hardware; manually cleared by firmware)
4..0	CHS[4:0]	R/W	0x00	ADC input channel select bit. 00000: AIN0, 00001: AIN1, 00010: AIN2, 00011: AIN3, 00100: AIN4, 00101: AIN5, 00110: AIN6, 00111: AIN7, 01000: AIN8, 01001: AIN9, 01010: AIN10, 01011:AIN11, 01100: AIN12, 01101: AIN13, 01110: AIN14, 01111: AIN15, 10000: AIN16*(1), 10001: VDD, 10010: VSS, Others: Reserved.

*(1) The AIN16 is internal 2.5V or 3.5V or 4.5V input channel. There is no any input pin from outside. In this time ADC reference voltage must be internal VDD and External voltage, not internal 2.5V or 3.5V or 4.5V.

ADB Register (0xD3)

Bit	Field	Type	Initial	Description
7..0	ADB[11:4]	R	-	ADC Result Bit [11:4]* in 12-bit ADC resolution mode.

* ADC data buffer is 12-bit length to store ADC converter result. The high byte is ADB register, and the low-nibble is ADR[3:0] bits.

ADR Register (0xD4)

Bit	Field	Type	Initial	Description
7	ADCKSW	R/W	0	ADC clock source select bit 0: fosc 1: flosc
6	GCHS	R/W	0	ADC global channel select bit. 0: Disable AIN channel. 1: Enable AIN channel.
5..4	ADCKS[1:0]	R/W	00	ADC's clock rate select bit. 00 = fosc/32, 01 = fosc/16, 10 = fosc/2, 11 = fosc/8 or 00 = flosc/32, 01 = flosc/16, 10 = flosc/2, 11 = flosc/8
3..0	ADB[3:0]	R	-	ADC Result Bit [3:0]* in 12-bit ADC resolution mode.

* ADC data buffer is 12-bit length to store ADC converter result. The high byte is ADB register, and the low-nibble is ADR[3:0] bits.

VREFH Register (0xD5)

Bit	Field	Type	Initial	Description
7	EVHENB	R/W	0	ADC internal reference high voltage control bit. 0: Enable ADC internal VREFH function. AVREFH/P1.1 pin is GPIO. 1: Disable ADC internal VREFH function. AVREFH/P1.1 pin is external AVREFH*(1) input pin.
2..0	VHS[2:0]	R/W	00	ADC internal reference high voltage selects bits. *(2) 000: VREFH = 2.5V. 001: VREFH = 3.5V. 010: VREFH = 4.5V. 011: VREFH = VDD. 100: VREFH = VDD and AIN16 = 2.5V. 101: VREFH = VDD and AIN16 = 3.5V. 110: VREFH = VDD and AIN16 = 4.5V. Others: Reserved.
Else	Reserved	R/W	0	

*(1) The AVREFH level must be between the VDD and 2.0V.

*(2) If AIN16 channel is selected as internal 2.5V or 3V.5 or 4.5V input channel. There is no any input pin from outside. In this time ADC reference high voltage must be internal VDD or External voltage, not internal 2.5V/3.5V/4.5V.

P0CON: 0xF018, P1CON: 0xF019, P2CON: 0xF01A

Bit	Field	Type	Initial	Description
7	P0CON7	R/W	0	P0.7 configuration control bit*. 0: P0.7 can be analog input pin or digital GPIO pin. 1: P0.7 is pure analog input pin and can't be a digital GPIO pin.
6	P0CON6	R/W	0	P0.6 configuration control bit*. 0: P0.6 can be analog input pin or digital GPIO pin. 1: P0.6 is pure analog input pin and can't be a digital GPIO pin.
5..0				et cetera

* P0CON [7:0], P1CON [7:0], P2CON [4:0] will configure related Port0, Port1, Port2 pin as pure analog input pin to avoid current leakage.

IEN0 Register (0xA8)

Bit	Field	Type	Initial	Description
7	EAL	R/W	0	Interrupts enable. Refer to Chapter Interrupt
Else				Refer to other chapter(s)

IEN2 Register (0xC8)

Bit	Field	Type	Initial	Description
0	EADC	R/W	0	ADC interrupt control bit. 0: Disable ADC interrupt function. 1: Enable ADC interrupt function.
Else				Refer to other chapter(s)

IRCON2 Register (0xA6)

Bit	Field	Type	Initial	Description
0	ADCF	R/W	0	ADC interrupt request flag. 0 = None ADC interrupt request. 1 = ADC interrupt request.
Else				Refer to other chapter(s)

20 UART0

The UART0 provides a flexible full-duplex synchronous/asynchronous receiver/transmitter. The serial interface provides an up to 500 kHz flexible full-duplex transmission. It can operate in four modes (one synchronous and three asynchronous). Mode0 is a shift register mode and operates as synchronous transmitter/receiver. In Mode1-Mode3 the UART operates as asynchronous transmitter/receiver with 8-bit or 9-bit data. The transfer format has start bit, 8-bit/ 9-bit data and stop bit. Transmission is started by writing to the S0BUF register. After reception, input data are available after completion of the reception in the S0BUF register. TB80/RB80 bit can be used as the 9th bit for transmission and reception in 9-bit UART mode. Programmable baud rate supports different speed peripheral devices.

The UART features include the following:

- Full-duplex, 2-wire synchronous/asynchronous data transfer.
- Programmable baud rate.
- 8-bit shift register: operates as synchronous transmitter/receiver
- 8-bit / 9-bit UART: operates as asynchronous transmitter/receiver with 8 or 9-bit data bits and programmable baud rate.

20.1 UART Operation

The UART0 UTX0 and URX0 pins are shared with GPIO. In synchronous mode, the UTX0/URX0 shared pins must set output high by software. In asynchronous mode (8-bit/9-bit UART), the UTX0 shared pins must set output high and URX0 set input high by software. Thus, URX0/UTX0 pins will transfers to UART purpose. When UART disables, the UART pins returns to GPIO last status.

The UTX0/URX0 pins also support open-drain structure. The open-drain option is controlled by PnOC bit. When PnOC=0, disable UTX0/URX0 open-drain structure. When PnOC=1, enable UTX0/URX0 open-drain structure. If enable open-drain structure, UTX0/URX0 pin must set high level (IO mode control will be ignored) and need external pull-up resistor.

The UART0 supports interrupt function. EU0TX and EU0RX are UART0 transfer interrupt function control bits. UART0 transmitter and receiver interrupt function is controlled by EU0TX and EU0RX respectively. EU0TX=0/EU0RX=0, disable transmitter/receiver interrupt function. EU0TX=1/ EU0RX=1, enable UART transmitter/ receiver interrupt function. When UART0 interrupt function enable, the program counter points to interrupt vector to do UART0 interrupt service routine after UART operating. TI0/RI0 is UART0 interrupt request flag, and also to be the UART operating status indicator when interrupt is disabled. TI0 and RI0 must clear by software.

UART0 provides four operating mode (one synchronous and three asynchronous) controlled by S0CON register. These modes can be support in different baud rate and communication protocols.

S0M0	S0M1	Mode	Synchronization	Clock Rate	Start Bit	Data Bits	Stop Bit	UART pins' mode and data
0	0	0	Synchronous	Fcpu/12	X	8	X	UTX0 pin: P03M=1 and P03=1 URX0 pin: Transmitter: P02M=1 and P02=1 Receiver: P02M=0 and P02=1
0	1	1	Asynchronous	Baud rate generator or T1 overflow rate	1	8	1	
1	0	2	Asynchronous	Fcpu/64 or Fcpu/32	1	9	1	
1	1	3	Asynchronous	Baud rate generator or T1 overflow rate	1	9	1	

20.2 Mode 0: Synchronous 8-bit Receiver/Transmitter

Mode0 is a shift register mode. It operates as synchronous transmitter/receiver. The UTX0 pin output shift clock for both transmit and receive condition. The URX0 pin is used to transmit and receive data. 8-bit data will be transmit and receive with LSB first. The baud rate is fcpu/12. Data transmission is started by writing data to S0BUF register. In the end of the 8th bit transmission, the TI0 flag is set. Data reception is controlled by REN0 bit and clearing RI0 bits. When REN0=1 and RI0 is from 1 to 0, data transmission starts and the RI0 flag is set at the end of the 8th bit reception.

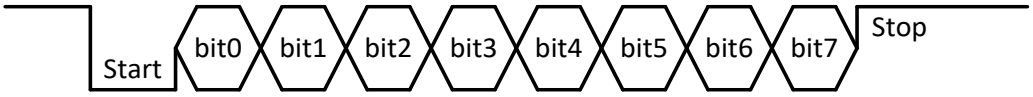
20.3 Mode 1: 8-bit Receiver/Transmitter with Variable Baud Rate

Mode1 supports an asynchronous 8-bit UART with variable baud rate. The transfer format includes 1 start bit, 8 data bits (LSB first) and 1 stop bit. Data is transmitted by UTX0 pin and received by URX0 pin. The baud rate clock source can be baud rate generator or T1 overflow controlled by BD0 bit. When BD0=0, the baud rate clock source is from T1 overflow. When BD0=1, the baud rate clock source is from baud rate generator controlled by S0RELH and S0RELL. Additionally, the baud rate can be doubled by SMOD0 bit.

Data transmission is controlled by TEN0 bit. After transmission configuration, load transmitted data

into S0BUF, and then UART0 starts to transmit the pocket. The TI0 flag is set at the beginning of the stop bit.

Data reception is controlled by REN0 bit. When REN0=1, data reception function is enabled. Data reception starts by receiving the start bit for master terminal, URX0 detects the falling edge of start bit, and then the RI0 flag is set in the middle of a stop bit. Until reception completion, input data is stored in S0BUF register and the stop bit is stored in RB80.



20.4 Mode 2: 9-bit Receiver/Transmitter with Fixed Baud Rate

Mode2 supports an asynchronous 9-bit UART with fixed baud rate. The transfer format includes 1 start bit, 9 data bits (LSB first) and 1 stop bit. Data is transmitted by UTX0 pin and received by URX0 pin. The baud rate clock source is fixed to fcpu/64 or fcpu/32 and is controlled by SMOD0 bit. When SMOD0=0, baud rate is fcpu/64. When SMOD0=1, baud rate is fcpu/32.

Data transmission is controlled by TEN0 bit. After transmission configuration, load transmitted data into S0BUF, and then UART0 starts to transmit the pocket. The 9th data bit is taken from TB80. The TI0 flag is set at the beginning of the stop bit.

Data reception is controlled by REN0 bit. When REN0=1, data reception function is enabled. Data reception starts by receiving the start bit for master terminal, URX0 detects the falling edge of start bit, and then the RI0 flag is set in the middle of a stop bit. Until reception completion, lower 8-bit input data is stored in S0BUF register and the 9th bit is stored in RB80.

20.5 Mode 3: 9-bit Receiver/Transmitter with Variable Baud Rate

Mode3 supports an asynchronous 9-bit UART with variable baud rate. The transfer format includes 1 start bit, 9 data bits (LSB first) and 1 stop bit. Data is transmitted by UTX0 pin and received by URX0 pin. The different between Mode2 and Mode3 is baud rate selection. In the Mode3, the baud rate clock source can be baud rate generator or T1 overflow controlled by BD0 bit. When BD0=0, the baud rate clock source is from T1 overflow. When BD0=1, the baud rate clock source is from baud rate generator controlled by S0RELH and S0RELL. Additionally, the baud rate can be doubled by SMOD0 bit.

Data transmission is controlled by TEN0 bit. After transmission configuration, load transmitted data into S0BUF, and then UART starts to transmit the pocket. The 9th data bit is taken from TB80. The TI0 flag is set at the beginning of the stop bit.

Data reception is controlled by REN0 bit. When REN0=1, data reception function is enabled. Data reception starts by receiving the start bit for master terminal, URX0 detects the falling edge of start bit, and then the RI0 flag is set in the middle of a stop bit. Until reception completion, lower 8-bit input data is stored in S0BUF register and the 9th bit is stored in RB80.

20.6 Multiprocessor Communication

UART supports multiprocessor communication between a master device and one or more slaver device in Mode2 and Mode3 (9-bit UART). The master identifies correct slavers by using the 9th data bit. When the communication starts, the master transmits a specific address byte with the 9th bit is set “1” to selected slavers, and then transmits a data byte with the 9th bit is set “0” in the following transmission.

Multiprocessor communication is controlled by S0M20 bit. When S0M20=0, disable multiprocessor communication. When S0M20=1, enable multiprocessor communication. If S0M20 is set, the UART0 reception interrupt is only generated when the 9th received bit is “1” (RB80). The slavers will compare received data with its own address data by software. If address byte is match, the slavers clear S0M20 bit to enable interrupt function in the following data transmission. The slavers with unmatched address, their S0M20 keep in “1” and will not generate interrupt in the following data transmission.

20.7 Baud Rate Control

The UART0 mode 0 has a fixed baud rate at fcpu/12, and the mode 2 has two baud rate selection which is chosen by SMOD0 bit: fcpu/64 (SMOD0 = 0) and fcpu/32 (SMOD0 = 1).

The baud rate of UART0 mode 1 and mode 3 is generated by either S0RELH/S0RELL registers (BD0 = 1) or Timer 1 overflow period (BD0 = 0). The SMOD0 bit doubles the frequency from the generator.

If the S0RELH/S0RELL is selected (BD0 = 1) in mode 1 and 3, the baud rate is generated as following equation.

$$\text{Baud Rate} = 2^{\text{SMOD0}} \times \frac{\text{fcpu}}{64 \times (1024 - \text{S0REL})} bps$$

Table 20-1 Recommended Setting for Common UART0 Baud Rates (fcpu = 8 MHz)

Baud Rate	SMOD0	S0RELH	S0RELL	Accuracy
4800	0	0x03	0xE6	0.16 %
9600	0	0x03	0XF3	0.16 %
19200	1	0x03	0xF3	0.16 %
38400	1	0x03	0xF9	-6.99 %
56000	1	0x03	0xFB	-10.71 %
57600	1	0x03	0xFC	8.51 %
115200	1	0x03	0xFE	8.51 %
128000	1	0x03	0xFE	-2.34 %
250000	1	0x03	0xFF	0 %

If the Timer 1 overflow period is selected (BD0 = 0) in mode 1 and 3, the baud rate is generated as following equation. The Timer 1 must be in 8-bit auto-reload mode which can generate periodically overflow signals.

$$\text{Baud Rate} = 2^{\text{SMOD0}} \times \frac{\text{T1 clock rate}}{32 \times (256 - \text{TH1})} bps$$

Table 20-2 Recommended Setting T1 overflow period (T1 clock=32M) for Common UART0 Baud Rates (fcpu = 8 MHz)

Baud Rate	SMOD0	Timer Period	TH1/TL1	Accuracy
4800	0	6.510 us	0x30	0.16 %
9600	1	6.510 us	0x30	0.16 %
19200	1	3.255 us	0x98	0.16 %
38400	1	1.628 us	0xCC	0.16 %
56000	1	1.116 us	0xDC	-0.80 %
57600	1	1.085 us	0xDD	-0.80 %
115200	1	0.543 us	0xEF	2.08 %
128000	1	0.488 us	0xF0	-2.40 %

✷ ***Note:***

1. When baud rate generator source is T1 overflow rate, the max counter value is 0xFB. (Only supports 0x00~0xFB).

2. When baud rate generator source is T1 overflow rate, T1 overflow rate must be greater four times to the system clock fcpu.

20.8 Power Saving

The UART0 module has clock gating function for saving power. When REN0 bit is 0, the UART0 module internal clocks are halted to reduce power consumption. UART0 relevant register (S0CON, S0CON2, S0BUF, S0RELL, S0RELH and SMOD0 bit) are unable to access.

Conversely, when REN0 bit is 1, UART0 internal clocks are run, and registers can access. The REN0 bit must be set to 1, before the initial setting UART.

20.9 UART Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SMODMX	-	-	-	-	UR1MX1	UR1MX0	UR0MX1	UR0MX0
SMODCAL	-	SMOD1	SMOD0	-	-	-	-	-
S0CON2	-	BD1	BD0	-	-	-	TEN1	TEN0
S0CON	S0M0	S0M1	S0M20	REN0	TB80	RB80	TI0	RI0
S0BUF	S0BUF7	S0BUF6	S0BUF5	S0BUF4	S0BUF3	S0BUF2	S0BUF1	S0BUF0
S0RELH	-	-	-	-	-	-	S0REL9	S0REL8
S0RELL	S0REL7	S0REL6	S0REL5	S0REL4	S0REL3	S0REL2	S0REL1	S0REL0
IEN0	EAL	-	-	EX2	ET1	EX1	ET0	EX0
IEN1	-	-	EU1RX	EU1TX	EU0RX	EU0TX	-	EI2C
P0OC	-	-	-	-	P03OC	P02OC	-	-
P1OC	-	-	-	-	-	P12OC	-	-

SMODMX Register (0x94)

Bit	Field	Type	Initial	Description
1..0	UR0MX	R/W	00	UART0 IO select 00 = UTX is P03 and URX is P02 01 = Reserved 10 = UTX(1) is P12 11 = URX(1) is P12
Else				Refer to other chapter(s)

SMODCAL Register (0x95)

Bit	Field	Type	Initial	Description
5	SMOD0	R/W	0	UART0 baud rate control In UART mode 0: Unused. In UART mode 1, 3: The baud rate generated as the equation in section 19.7 (Baud Rate Control). 0 = fcpu/64 1 = fcpu/32
Else				Refer to other chapter(s)

S0CON2 Register (0x96)

Bit	Field	Type	Initial	Description
5	BD0	R/W	0	Baud rate generators selection (mode 1, 3) 0: Timer 1 overflow period 1: Controlled by S0RELH, S0RELL registers
0	TEN0	R/W	0	UART0 TX control bit 0: UART0 TX disable 1: UART0 TX enable
Else				Refer to other chapter(s)

S0CON Register (0x97)

Bit	Field	Type	Initial	Description
7..6	S0M[0:1]	R/W	00	UART0 IO select 00: Mode 0 01: Mode 1 10: Mode 2 11: Mode 3
5	S0M20	R/W	0	Multiprocessor communication (mode 2, 3) 0: Disable 1: Enable
4	REN0	R/W	0	UART0 module (and reception function) 0: Disable for power saving* 1: Enable for UART operating
3	TB80	R/W	0	The 9th bit transmission data (mode 2, 3)
2	RB80	R/W	0	The 9th bit data from reception
1	TI0	R/W	0	UART0 interrupt flag of transmission
0	RI0	R/W	0	UART0 interrupt flag of reception

* When REN0 bit is 0, UART0 relevant register are unable to access, and the module internal clocks are halted.

✷ ***Note: TI0 and RI0 are clear by software when interrupt is enabled.***

S0BUF Register (0x99)

Bit	Field	Type	Initial	Description
7..0	S0BUF	R/W	0x00	Action of writing data triggers UART0 communication (LSB first). Reception data is available to read by the end of packages.

S0RELH/S0RELL Registers (S0RELH: 0x9B, S0RELL: 0x9A)

Bit	Field	Type	Initial	Description
15..10	Reserved	R	0x00	
9..0	S0REL[9:0]	R/W	0x00	S0RELH[1:0] & S0RELL[7:0]. UART0 Reload Register is used for UART0 baud rate generation.

IEN0 Register (0xA8)

Bit	Field	Type	Initial	Description
7	EAL	R/W	0	Interrupts enable. Refer to Chapter Interrupt
Else				Refer to other chapter(s)

IEN1 Register (0xB8)

Bit	Field	Type	Initial	Description
3	EU0RX	R/W	0	UART0 receiver interrupt enable
2	EU0TX	R/W	0	UART0 transmitter interrupt enable
Else				Refer to other chapter(s)

P0OC Register (0xF020)

Bit	Field	Type	Initial	Description
2	P03OC	R/W	0	0: Switch P0.3 (URX0) to push-pull mode 1: Switch P0.3 (URX0) to open-drain mode
1	P02OC	R/W	0	0: Switch P0.2(UTX0) to push-pull mode 1: Switch P0.2 (UTX0) to open-drain mode
Else				Refer to other chapter(s)

P1OC Register (0xF021)

Bit	Field	Type	Initial	Description
2	P12OC	R/W	0	0: Switch P1.2(URX0/UTX0) to push-pull mode 1: Switch P1.2 (URX0/UTX0) to open-drain mode
Else				Refer to other chapter(s)

21 UART1

The UART1 provides a flexible full-duplex synchronous/asynchronous receiver/transmitter. The serial interface provides an up to 500 kHz flexible full-duplex transmission. It can operate in four modes (one synchronous and three asynchronous). Mode0 is a shift register mode and operates as synchronous transmitter/receiver. In Mode1-Mode3 the UART operates as asynchronous transmitter/receiver with 8-bit or 9-bit data. The transfer format has start bit, 8-bit/ 9-bit data and stop bit. Transmission is started by writing to the S1BUF register. After reception, input data are available after completion of the reception in the S1BUF register. TB81/RB81 bit can be used as the 9th bit for transmission and reception in 9-bit UART mode. Programmable baud rate supports different speed peripheral devices.

The UART features include the following:

- Full-duplex, 2-wire synchronous/asynchronous data transfer.
- Programmable baud rate.
- 8-bit shift register: operates as synchronous transmitter/receiver
- 8-bit / 9-bit UART: operates as asynchronous transmitter/receiver with 8 or 9-bit data bits and programmable baud rate.

21.1 UART Operation

The UART1 UTX1 and URX1 pins are shared with GPIO. In synchronous mode, the UTX1/URX1 shared pins must set output high by software. In asynchronous mode (8-bit/9-bit UART), the UTX1 shared pins must set output high and URX1 set input high by software. Thus, URX1/UTX1 pins will transfers to UART purpose. When UART disables, the UART pins returns to GPIO last status.

The UTX1/URX1 pins also support open-drain structure. The open-drain option is controlled by PnOC bit. When PnOC=0, disable UTX1/URX1 open-drain structure. When PnOC=1, enable UTX1/URX1 open-drain structure. If enable open-drain structure, UTX1/URX1 pin must set high level (IO mode control will be ignored) and need external pull-up resistor.

The UART1 supports interrupt function. EU1TX and EU1RX is UART1 transfer interrupt function control bit. UART1 transmitter and receiver interrupt function is controlled by EU1TX and EU1RX respectively. EU1TX=0/EU1RX=0, disable transmitter/receiver interrupt function. EU1TX=1/ EU1RX=1, enable UART transmitter/ receiver interrupt function. When UART1 interrupt function enable, the program counter points to interrupt vector to do UART1 interrupt service routine after UART operating. TI1/RI1 is UART1 interrupt request flag, and also to be the UART operating status indicator when interrupt is disabled. TI1 and RI1 must clear by software.

UART1 provides four operating mode (one synchronous and three asynchronous) controlled by S1CON register. These modes can be support in different baud rate and communication protocols.

S1M0	S1M1	Mode	Synchronization	Clock Rate	Start Bit	Data Bits	Stop Bit	UART pins' mode and data
0	0	0	Synchronous	Fcpu/12	X	8	X	UTX1 pin: P32M=1 and P32=1 URX1 pin: Transmitter: P33M=1 and P33=1 Receiver: P33M=0 and P33=1
0	1	1	Asynchronous	Baud rate generator or T1 overflow rate	1	8	1	
1	0	2	Asynchronous	Fcpu/64 or Fcpu/32	1	9	1	
1	1	3	Asynchronous	Baud rate generator or T1 overflow rate	1	9	1	

21.2 Mode 0: Synchronous 8-bit Receiver/Transmitter

Mode0 is a shift register mode. It operates as synchronous transmitter/receiver. The UTX1 pin output shift clock for both transmit and receive condition. The URX1 pin is used to transmit and receive data. 8-bit data will be transmit and receive with LSB first. The baud rate is fcpu/12. Data transmission is started by writing data to S1BUF register. In the end of the 8th bit transmission, the TI1 flag is set. Data reception is controlled by REN1 bit and clearing RI1 bits. When REN1=1 and RI1 is from 1 to 0, data transmission starts and the RI1 flag is set at the end of the 8th bit reception.

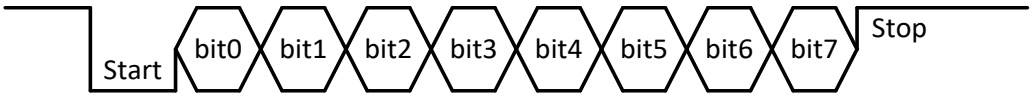
21.3 Mode 1: 8-bit Receiver/Transmitter with Variable Baud Rate

Mode1 supports an asynchronous 8-bit UART with variable baud rate. The transfer format includes 1 start bit, 8 data bits (LSB first) and 1 stop bit. Data is transmitted by UTX1 pin and received by URX1 pin. The baud rate clock source can be baud rate generator or T1 overflow controlled by BD1 bit. When BD1=0, the baud rate clock source is from T1 overflow. When BD1=1, the baud rate clock source is from baud rate generator controlled by S1RELH and S1RELL. Additionally, the baud rate

can be doubled by SMOD1 bit.

Data transmission is controlled by TEN1 bit. After transmission configuration, load transmitted data into S1BUF, and then UART1 starts to transmit the pocket. The TI1 flag is set at the beginning of the stop bit.

Data reception is controlled by REN1 bit. When REN1=1, data reception function is enabled. Data reception starts by receiving the start bit for master terminal, URX1 detects the falling edge of start bit, and then the RI1 flag is set in the middle of a stop bit. Until reception completion, input data is stored in S1BUF register and the stop bit is stored in RB81.



21.4 Mode 2: 9-bit Receiver/Transmitter with Fixed Baud Rate

Mode2 supports an asynchronous 9-bit UART with fixed baud rate. The transfer format includes 1 start bit, 9 data bits (LSB first) and 1 stop bit. Data is transmitted by UTX1 pin and received by URX1 pin. The baud rate clock source is fixed to fcpu/64 or fcpu/32 and is controlled by SMOD1 bit. When SMOD1=0, baud rate is fcpu/64. When SMOD1=1, baud rate is fcpu/32.

Data transmission is controlled by TEN1 bit. After transmission configuration, load transmitted data into S1BUF, and then UART1 starts to transmit the pocket. The 9th data bit is taken from TB81. The TI1 flag is set at the beginning of the stop bit.

Data reception is controlled by REN1 bit. When REN1=1, data reception function is enabled. Data reception starts by receiving the start bit for master terminal, URX1 detects the falling edge of start bit, and then the RI1 flag is set in the middle of a stop bit. Until reception completion, lower 8-bit input data is stored in S1BUF register and the 9th bit is stored in RB81.

21.5 Mode 3: 9-bit Receiver/Transmitter with Variable Baud Rate

Mode3 supports an asynchronous 9-bit UART with variable baud rate. The transfer format includes 1 start bit, 9 data bits (LSB first) and 1 stop bit. Data is transmitted by UTX1 pin and received by URX1 pin. The different between Mode2 and Mode3 is baud rate selection. In the Mode3, the baud rate clock source can be baud rate generator or T1 overflow controlled by BD1 bit. When BD1=0, the baud rate clock source is from T1 overflow. When BD1=1, the baud rate clock source is

from baud rate generator controlled by S1RELH and S1RELL. Additionally, the baud rate can be doubled by SMOD1 bit.

Data transmission is controlled by TEN1 bit. After transmission configuration, load transmitted data into S1BUF, and then UART starts to transmit the pocket. The 9th data bit is taken from TB81. The TI1 flag is set at the beginning of the stop bit.

Data reception is controlled by REN1 bit. When REN1=1, data reception function is enabled. Data reception starts by receiving the start bit for master terminal, URX1 detects the falling edge of start bit, and then the RI1 flag is set in the middle of a stop bit. Until reception completion, lower 8-bit input data is stored in S1BUF register and the 9th bit is stored in RB81.

21.6 Multiprocessor Communication

UART supports multiprocessor communication between a master device and one or more slaver device in Mode2 and Mode3 (9-bit UART). The master identifies correct slavers by using the 9th data bit. When the communication starts, the master transmits a specific address byte with the 9th bit is set “1” to selected slavers, and then transmits a data byte with the 9th bit is set “0” in the following transmission.

Multiprocessor communication is controlled by S1M20 bit. When S1M20=0, disable multiprocessor communication. When S1M20=1, enable multiprocessor communication. If S1M20 is set, the UART1 reception interrupt is only generated when the 9th received bit is “1” (RB81). The slavers will compare received data with its own address data by software. If address byte is match, the slavers clear S1M20 bit to enable interrupt function in the following data transmission. The slavers with unmatched address, their S1M20 keep in “1” and will not generate interrupt in the following data transmission.

21.7 Baud Rate Control

The UART1 mode 0 has a fixed baud rate at fcpu/12, and the mode 2 has two baud rate selection which is chosen by SMOD1 bit: fcpu/64 (SMOD1 = 0) and fcpu/32 (SMOD1 = 1).

The baud rate of UART1 mode 1 and mode 3 is generated by either S1RELH/S1RELL registers (BD1 = 1) or Timer 1 overflow period (BD1 = 0). The SMOD1 bit doubles the frequency from the generator.

If the S1RELH/S1RELL is selected (BD1 = 1) in mode 1 and 3, the baud rate is generated as following

equation.

$$\text{Baud Rate} = 2^{\text{SMOD1}} \times \frac{\text{fcpu}}{64 \times (1024 - \text{S1REL})} bps$$

Table 21-1 Recommended Setting for Common UART1 Baud Rates (fcpu = 8 MHz)

Baud Rate	SMOD1	S2RELH	S2RELL	Accuracy
4800	0	0x03	0xE6	0.16 %
9600	0	0x03	0XF3	0.16 %
19200	1	0x03	0xF3	0.16 %
38400	1	0x03	0xF9	-6.99 %
56000	1	0x03	0xFB	-10.71 %
57600	1	0x03	0xFC	8.51 %
115200	1	0x03	0xFE	8.51 %
128000	1	0x03	0xFE	-2.34 %
250000	1	0x03	0xFF	0 %

If the Timer 1 overflow period is selected (BD1 = 0) in mode 1 and 3, the baud rate is generated as following equation. The Timer 1 must be in 8-bit auto-reload mode which can generate periodically overflow signals.

$$\text{Baud Rate} = 2^{\text{SMOD1}} \times \frac{\text{T1 clock rate}}{32 \times (256 - \text{TH1})} bps$$

Table 21-2 Recommended Setting T1 overflow period (T1 clock=32M) for Common UART1 Baud Rates (fcpu = 8 MHz)

Baud Rate	SMOD1	Timer Period	TH1/TL1	Accuracy
4800	0	6.510 us	0x30	0.16 %
9600	1	6.510 us	0x30	0.16 %
19200	1	3.255 us	0x98	0.16 %
38400	1	1.628 us	0xCC	0.16 %
56000	1	1.116 us	0xDC	-0.80 %
57600	1	1.085 us	0xDD	-0.80 %
115200	1	0.543 us	0xEF	2.08 %
128000	1	0.488 us	0xF0	-2.40 %

✸ ***Note:***

1. When baud rate generator source is T1 overflow rate, the max counter value is

0xFB. (Only supports 0x00~0xFB).

2. When baud rate generator source is T1 overflow rate, the system clock fcpu must be greater four times to T1 overflow rate.

21.8 Power Saving

The UART1 module has clock gating function for saving power. When REN1 bit is 0, the UART1 module internal clocks are halted to reduce power consumption. UART1 relevant register (S1CON, S0CON2, S1BUF, S1RELL, S1RELH and SMOD1 bit) are unable to access.

Conversely, when REN1 bit is 1, UART1 internal clocks are run, and registers can access. The REN1 bit must be set to 1, before the initial setting UART.

21.9 UART Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SMODMX	-	-	-	-	UR1MX1	UR1MX0	UR0MX1	UR0MX0
SMODCAL	-	SMOD1	SMOD0	-	-	-	-	-
S0CON2	-	BD1	BD0	-	-	-	TEN1	TEN0
S1CON	S1M0	S1M1	S1M20	REN1	TB81	RB81	TI1	RI1
S1BUF	S1BUF7	S1BUF6	S1BUF5	S1BUF4	S1BUF3	S1BUF2	S1BUF1	S1BUF0
S1RELH	-	-	-	-	-	-	S1REL9	S1REL8
S1RELL	S1REL7	S1REL6	S1REL5	S1REL4	S1REL3	S1REL2	S1REL1	S1REL0
IEN0	EAL	-	-	EX2	ET1	EX1	ET0	EX0
IEN1	-	-	EU1RX	EU1TX	EU0RX	EU0TX	-	EI2C
P2OC	-	-	-	-	-	-	P21OC	P20OC
P3OC	-	-	-	-	P33OC	P32OC	-	-

SMODMX Register (0x94)

Bit	Field	Type	Initial	Description
3..2	URX1MX	R/W	00	UART1 IO select 00 = UTX1 is P32 and URX1 is P33 01 = UTX1(1) is P20 and URX1(1) is P21 10 = Reserved 11 = Reserved
Else				Refer to other chapter(s)

SMODCAL Register (0x95)

Bit	Field	Type	Initial	Description
6	SMOD1	R/W	0	UART1 baud rate control. In UART mode 0: Unused. In UART mode 1, 3: The baud rate generated as the equation in section 19.7 (Baud Rate Control). In UART mode 2: 0: fcpu/64 1: fcpu/32
Else				Refer to other chapter(s)

S0CON2 Register (0x96)

Bit	Field	Type	Initial	Description
6	BD1	R/W	0	Baud rate generators selection (mode 1, 3) 0: Timer 1 overflow period 1: Controlled by S1RELH, S1RELL registers
1	TEN1	R/W	0	UART1 TX control bit 0: UART1 TX disable 1: UART1 TX enable
Else				Refer to other chapter(s)

S1CON Register (0x9C)

Bit	Field	Type	Initial	Description
7..6	S1M[0:1]	R/W	00	UART1 mode selection 00: Mode 0 01: Mode 1 10: Mode 2 11: Mode 3
5	S1M20	R/W	0	Multiprocessor communication (mode 2, 3) 0: Disable 1: Enable
4	REN1	R/W	0	UART1 module (and reception function)

				0: Disable for power saving* 1: Enable for UART operating
3	TB81	R/W	0	The 9th bit transmission data (mode 2, 3)
2	RB81	R/W	0	The 9th bit data from reception
1	TI1	R/W	0	UART1 interrupt flag of transmission
0	RI1	R/W	0	UART1 interrupt flag of reception

* When REN1 bit is 0, UART1 relevant register are unable to access, and the module internal clocks are halted.

✷ ***Note: TI1 and RI1 are clear by software when interrupt is enabled.***

S1BUF Register (0x9D)

Bit	Field	Type	Initial	Description
7..0	S1BUF	R/W	0x00	Action of writing data triggers UART1 communication (LSB first). Reception data is available to read by the end of packages.

S1RELH/S1RELL Registers (S1RELH: 0x9F, S1RELL: 0x9E)

Bit	Field	Type	Initial	Description
15..10	Reserved	R	0x00	
9..0	S1REL[9:0]	R/W	0x00	S1RELH[1:0] & S1RELL[7:0]. UART1 Reload Register is used for UART1 baud rate generation.

IEN0 Register (0xA8)

Bit	Field	Type	Initial	Description
7	EAL	R/W	0	Interrupts enable. Refer to Chapter Interrupt
Else				Refer to other chapter(s)

IEN1 Register (0xB8)

Bit	Field	Type	Initial	Description
5	EU1RX	R/W	0	UART1 receiver interrupt enable
4	EU1TX	R/W	0	UART1 transmitter interrupt enable
Else				Refer to other chapter(s)

P2OC Register (0xF022)

Bit	Field	Type	Initial	Description
1	P21OC	R/W	0	0: Switch P2.1 (URX1) to push-pull mode 1: Switch P2.1 (URX1) to open-drain mode

0	P20OC	R/W	0	0: Switch P2.0(UTX1) to push-pull mode 1: Switch P2.0 (UTX1) to open-drain mode
Else				Refer to other chapter(s)

P3OC Register (0xF023)

Bit	Field	Type	Initial	Description
3	P33OC	R/W	0	0: Switch P3.3 (URX1) to push-pull mode 1: Switch P3.3 (URX1) to open-drain mode
2	P32OC	R/W	0	0: Switch P3.2 (UTX1) to push-pull mode 1: Switch P3.2 (UTX1) to open-drain mode
Else				Refer to other chapter(s)

22 I2C

The I2C is a serial communication interface for data exchanging from one MCU to one MCU or other hardware peripherals. The device can transmit data as a master or a slave with two bi-directional IO, SDA (Serial data output) and SCL (Serial clock input).

When a master transmit data to a slave, it's called "WRITE" operation; when a slave transmit data to a master, it's called "READ" operation. It also supports multi-master communication and keeps data transmission correctly by an arbitration method to decide one master has the control on bus and transmit its data.

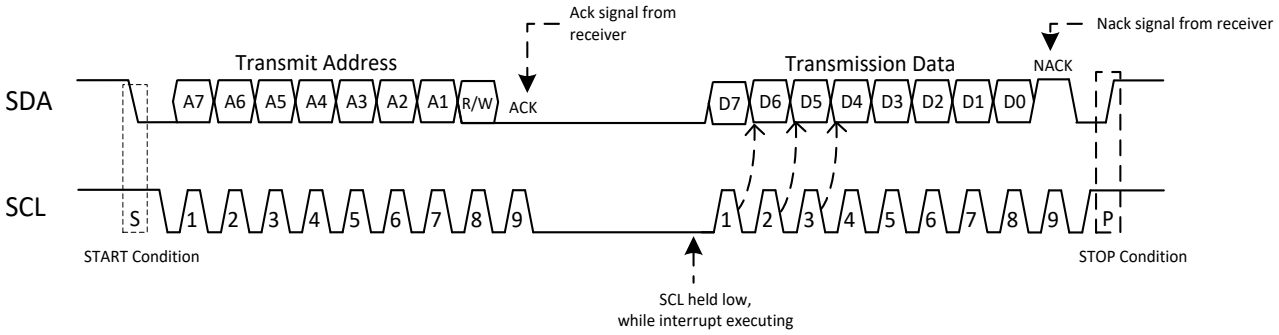
- Master Tx, Rx Mode
- Slave Tx, Rx mode (with general address call) for multiplex slave in single master situation.
- 2-wire synchronous data transfer/receiver.
- Support 100K/400K clock rate.

22.1 I2C Protocol

I2C transmission structure includes a START(S) condition, 8-bit address byte, one or more data byte and a STOP (P) condition. START condition is generated by master to initial any transmission.

Data is transmitted with the Most Significant Bit (MSB) first. In address byte, the higher 7-bit is address bit and the lowest bit is data direction (R/W) bit. When R/W=0, it assigns a "WRITE" operation. When R/W=1, it assigns a "READ" operation.

After each byte is received, the receiver (a master or a slave) must send an acknowledge (ACK). If transmitter can't receive an ACK, it will recognize a not acknowledge (NACK). In WRITE operation, the master will transmit data to the slave and then waits for ACK from slave. In READ operation, the slave will transmit data to the master and then waits for ACK from master. In the end, the master will generate a STOP condition to finish transmission.

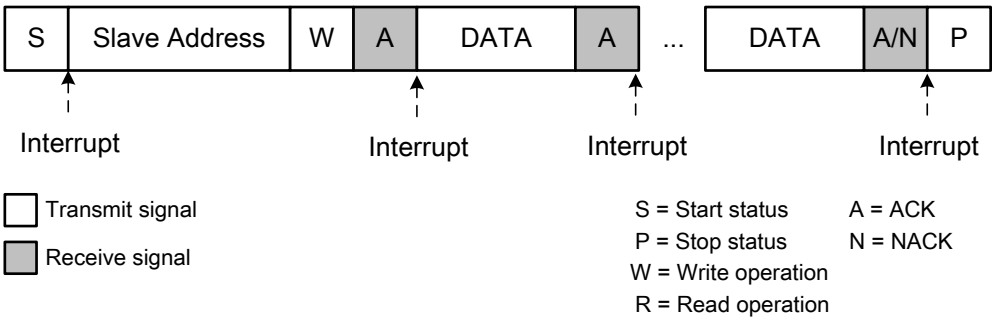


22.2 I2C Transfer Modes

The I2C can operate as a master/slave to execute the 8-bit serial data transmission/reception operation. Thus, the module can operate in one of four modes: Master Transmitter, Master Receiver, Slave Transmitter and Slave Receiver.

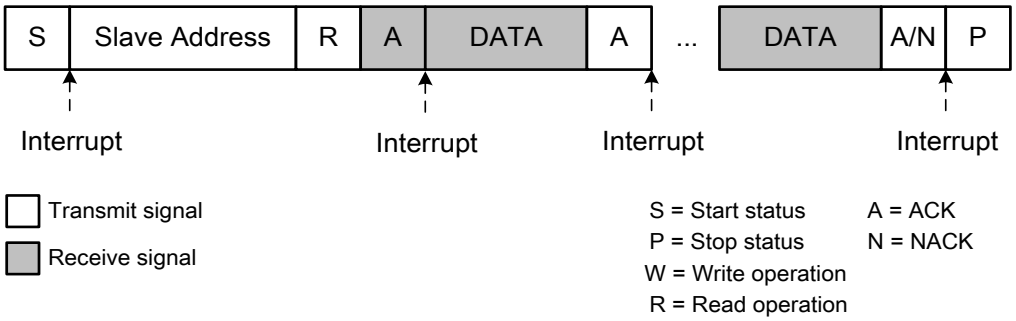
22.2.1 Master Transmitter Mode

The master transmits information to the slave. The serial data is output via SDA while the serial clock is output on SCL. Data transmission starts via generate a START(S) signal. After the START signal, the specific address byte of slave device is sent. The address byte includes 7-bit address bit and an 8th data direction (R/W) bit. The R/W is set “0” to enable the master transmission. In the following, the master transmits one or more data byte to the slaver. After each data is transmitted, the master waits for the acknowledge (ACK) from the slave. In the end, the master generates a STOP (P) signal to terminate the data transmission.



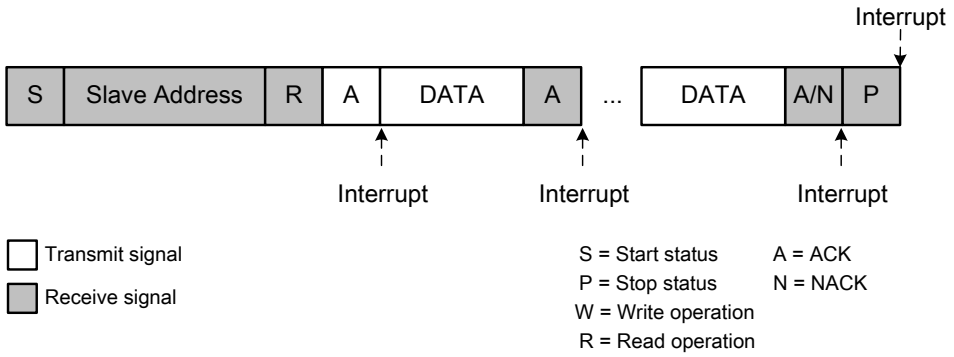
22.2.2 Master Receiver Mode

The master receives the information from the slave. The serial data input via SDA while the serial clock output on SCL. Data reception starts via generate a START(S) signal. After the START signal, the specific address byte of slave device is sent. The address byte includes 7-bit address bit and an 8th data direction (R/W) bit. The R/W is set “1” to enable the master reception. In the following, the master receives one or more data byte from the slaver. After each data is received, the master generates the acknowledge (ACK) or not acknowledge (NACK) to the slave via the status of AA bit. In the end, the master generates a STOP (P) signal to terminate the data transmission.



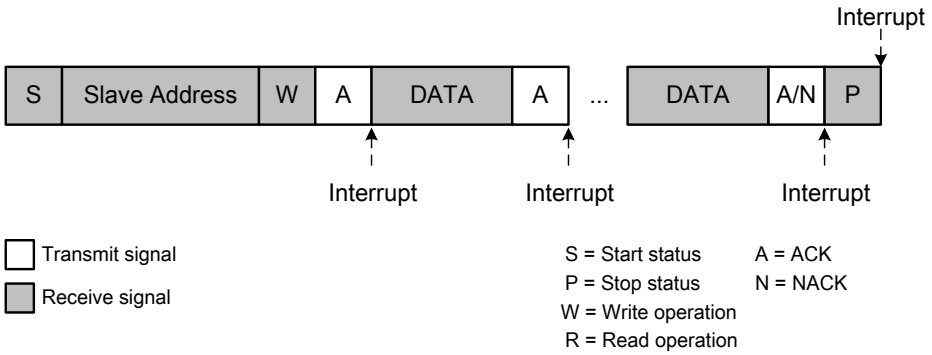
22.2.3 Slave Transmitter Mode

The slave transmits information to the master. The serial data output via SDA while the serial clock input on SCL. Data transmission starts via receive a START(S) signal from the master. After the START signal, the specific address byte of slave device is received. The address byte includes 7-bit address bit and an 8th data direction (R/W) bit. The R/W is set “1” to enable the slave transmission. If the received address byte match the address in I2CADR register, the slave generate an acknowledge (ACK). Otherwise, if general call address condition is set (GC=1), the slave also generate an acknowledge (ACK) after general call address (0x00) is received. In the following, the slave transmits one or more data byte to the master. After each data is transmitted, the slave waits for the acknowledge (ACK) from the master. In the end, the slave receives a STOP (P) signal from the master to terminate the data transmission.



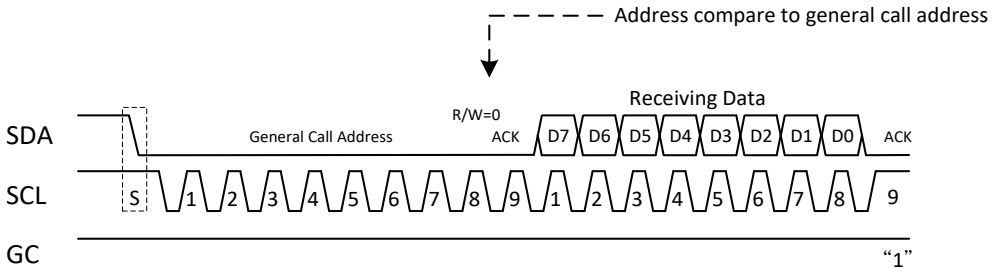
22.2.4 Slave Receiver Mode

The slave receives information from the master. Both the serial data and the serial clock are input on SDA and SCL. Data reception starts via receive a START(S) signal from the master. After the START signal, the specific address byte of slave device is received. The address byte includes 7-bit address bit and an 8th data direction (R/W) bit. The R/W is set “0” to enable the slave reception. If the received address byte match the address in I2CADR register, the slave generate an acknowledge (ACK). Otherwise, if general call address condition is set (GC=1), the slave also generate an acknowledge (ACK) after general call address (0x00) is received. In the following, the slave receives one or more data byte from the master. After each data is receives, the slave generates the acknowledge (ACK) or not acknowledge (NACK) to the master via the status of AA bit. In the end, the slave receives a STOP (P) signal from the master to terminate the data transmission.



22.3 General Call Address

In I2C bus, the first 7-bit is the slave address. Only the address matches slave address, the slave will response an ACK. The exception is the general call address which can address all slave devices. When this address occur, all devices should response an acknowledge (ACK). The general call address is a special address which is reserved as all “0” of 7-bit address. The general call address function is control by GC bit. Set this bit will enable general call address and clear it will disable. When GC=1, the general call address will be recognized. When GC=0, the general call address will be ignored.



22.4 Serial Clock Generator

In master mode, the SCL clock rate generator’s is controlled by CR[2:0] bit of I2CCON register.

When CR[1:0]=00~11, SCL clock rate is from internal clock generator.

$$\text{SCL Clock Rate} = \frac{\text{Fhosc}}{\text{Prescaler}} (\text{Prescaler} = 320\sim40)$$

When CR2=1, SCL clock rate is from Timer 1 overflow rate .

$$\text{SCL Clock Rate} = \frac{\text{Timer 1 Overflow}}{8}$$

The table below shows the clock rate under different setting.

CR2	CR1	CR0	Fhosc=4M	Fhosc=8M	Fhosc=16M	Fhosc=32M	Clock Divider

0	0	0	100	200	400	-	40
0	0	1	50	100	200	400	80
0	1	0	25	50	100	200	160
0	1	1	12.5	25	50	100	320
1	X	X	(Timer 1 overflow rate)/8				

✸ *Note:*

1. The first step of I2C operation is to setup the I2C pins' mode. Must be set "input mode" in SDA/SCL pins.

2. When clock generator source is T1 overflow rate, the max counter value is 0xF9. (Only supports 0x00~0xF9). And in this time if T1 clock rate is IHRC_32MHz, SCL maximum clock rate is 800kHz.

3. If user wants to generate SCL clock rate is 100kHz/400kHz, you can set T1 counter value is 0xD8/0xF6 easily.

22.5 Synchronization and Arbitration

In multi-master condition, more than one master may transmit on bus in the same time. It must be decided which master has the control of bus and complete its transmission. Clock synchronization and arbitration are used to configure multi-master transmission. Clock synchronization is executed by synchronizing the SCL signal with another devices.

When two masters want to transmit data in the same, the clock synchronization will start by the High to Low transition on the SCL. If master 1 clock set LOW first, it holds the SCL in LOW status until the clock transit to HIGH status. However, if another master clock still keep LOW status, the Low to High transition of master 1 may not change SCL status (SCL keep LOW). In the other word, SCL keep LOW by the master with the longest clock time in LOW status. The SCL will transit from LOW to HIGH when the all devices clock transit to HIGH status. In the duration, the master1 will keep in HIGH status and wait for SCL transition (from LOW to HIGH), then continue its transmission. After clock synchronization, all devices clock and SCL clock are the same. Arbitration is used to decide which master can complete its transmission by SDA signal. Two masters may send out a START condition and transmit data on bus in the same time. They may influence by each other. Arbitration will force one master to lose the control on bus. Data transmission will keep until

master output different data signal. If one master transmits HIGH status and another master transmits LOW status, the SDA will be pull low. The master output High will detect the different with SDA and lose the control on bus. The master with LOW status wins the bus control and continues its transmission. There is no data miss during arbitration.

22.6 System Management Bus Extension

The optional System Management Bus (SMBus) protocol hardware supports 3 types timeout detection: (1) Tmext Timeout Detection: The cumulative stretch clock cycles within one byte. (2)Tsext Timeout Detection: The cumulative stretch clock cycles between start and stop condition. (3)Timeout Detection: The clock low measurement.

Timeout detection is controlled by SMBSEL and SMBDST registers. The SMBEXE bit of SMBSEL is SMBus extension function enable bit. When SMBEXE=1, SMBus extension function is enabled. Otherwise, Disable SMBus extension function. Timeout type and period setting is controlled by SMBTOP[2:0] and SMBDST. The period of SMBus timeout is controlled by three 16-bit buffers of Tmext, Tsext and Tout. The equation is as following.

$$\text{Tmext/Tsext/Tout} = \frac{\text{Timeout Period(sec)xFhosc(Hz)}}{1024}$$

Tmext is support by two 8-bit register of Tmext_L and Tmext_H . Tmext_L hold the low byte and Tmext_H hold high byte. Tsext is support by two 8-bit register of Tsext_L and Tsext_H . Tsext_L hold the low byte and Tsext_H hold high byte. Tout is support by two 8-bit register of Tout_L and Tout_H . Tout_L hold the low byte and Tout_H hold high byte.

Type	Time out period	Fhosc=32MHz	
		DEC	HEX
Tmext	5ms	156	9C
Tsext	25ms	781	30D
Tout	35ms	1094	446

By the setting of SMBTOP[2:0] to choose register type (as the table below), and write to register by write data to SMBDST register.

SMBTOP[2:0]	SMBDST	Description
000	Tmext_L	Select the low byte of Tmext register.
001	Tmext_H	Select the high byte of Tmext register.
010	Tsext_L	Select the low byte of Tsext register.
011	Tsext_H	Select the high byte of Tsext register.
100	Tout_L	Select the low byte of Tout register.

101	Tout_H	Select the high byte of Tout register.

When the SMBus extension function is enabled the lower 3-bit of I2CSTA hold the information about time out as the table below.

I2CSTA	Description
XXXX X000	No timeout errors.
XXXX XXX1	Tout timeout error.
XXXX XX1X	Tsext timeout error.
XXXX X1XX	Tmext timeout error.

22.7 Power Saving

The I2C module has clock gating function for saving power. When ENS1 bit is 0, the I2C module internal clocks are halted to reduce power consumption. I2C relevant register (I2CDAT, I2CADR, I2CCON, I2CSTA, SMBSEL and SMBDST) are unable to access. Conversely, when ENS1 bit is 1, I2C internal clocks are run, and registers can access. The ENS1 bit must be set to 1, before the initial setting I2C.

22.8 I2C Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CDAT	I2CDAT7	I2CDAT6	I2CDAT5	I2CDAT4	I2CDAT3	I2CDAT2	I2CDAT1	I2CDAT0
I2CADR	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0	GC
I2CCON	CR2	ENS1	STA	STO	SI	AA	CR1	CR0
I2CSTA	I2CSTA7	I2CSTA6	I2CSTA5	I2CSTA4	I2CSTA3	I2CSTA2	I2CSTA1	I2CSTA0
SMBSEL	SMBEXE	-	-	-	I2CMX	SMBTOP2	SMBTOP1	SMBTOP0
SMBDST	SMBD7	SMBD6	SMBD5	SMBD4	SMBD3	SMBD2	SMBD1	SMBD0
IEN0	EAL	EX4	EX3	EX2	ET1	EX1	ET0	EX0
IEN1	-	-	EU1RX	EU1TX	EU0RX	EU0TX	-	EI2C
P1M	P17M	P16M	P15M	P14M	P13M	P12M	P11M	P10M
P3M	P37M	P36M	P35M	P34M	P33M	P32M	P31M	P30M

I2CDAT Register (0xDA)

Bit	Field	Type	Initial	Description
7:0	I2CDAT[7:0]	R/W	0x00	The I2CDAT register contains a byte to be transmitted through I2C bus or a byte which has just been received through I2C bus. The CPU can read from and write to this 8-bit, directly addressable SFR while it is not in the process of byte shifting. The I2CDAT register is not shadowed or double buffered so the user should only read I2CDAT when an I2C interrupt occurs.

I2CADR Register (0xDB)

Bit	Field	Type	Initial	Description
7:1	I2CADR[6:0]	R/W	0x00	I2C slave address
0	GC	R/W	0	General call address (0X00) acknowledgment 0: ignored 1: recognized

I2CCON Register (0xDC)

Bit	Field	Type	Initial	Description
7,1,0	CR[2:0]	R/W	0	I2C clock rate 000: fhosc/40 001: fhosc /80 010: fhosc /160 011: fhosc /320 1XX: Timer 1 overflow-period/8
6	ENS1	R/W	0	I2C functionality 0: Disable for power saving* 1: Enable for I2C operating
5	STA	R/W	0	START flag 0: No START condition is transmitted. 1: A START condition is transmitted if the bus is free.
4	STO	R/W	0	STOP flag 0: No STOP condition is transmitted. 1: A STOP condition is transmitted to the I2C bus in master mode.
3	SI	R/W	0	Serial interrupt flag The SI is set by hardware when one of 25 out of 26 possible I2C states is entered. The only state that does

				not set the SI is state F8h, which indicates that no relevant state information is available. The SI flag must be cleared by software. In order to clear the SI bit, '0' must be written to this bit. Writing a '1' to SI bit does not change value of the SI.
2	AA	R/W	0	Assert acknowledge flag 0: A NACK will be returned when a byte has received 1: An ACK will be returned when a byte has received

* When ENS1 bit is 0, I2C relevant register are unable to access, and the module internal clocks are halted.

I2CSTA Register (0xDD)

Bit	Field	Type	Initial	Description
7:3	I2CSTA[7:3]	R	11111	I2C Status Code
2..0	I2CSTA[2:0]	R	000	SMBus Status Code

I2C status code and status

Mode	Status Code	Status of the I2C	Application software response					Next action taken by I2C hardware
			To/from I2CDAT	TO I2CCON				
				STA	STO	SI	AA	
Master Transmitter/ Receiver	08H	A START condition has been transmitted	Load SLA+R/W	X	0	0	X	SLA+R/W will be transmitted; ACK will be received
	10H	A repeated START condition has been transmitted.	Load SLA+R	X	0	0	X	SLA+R/W will be transmitted; ACK will be received
			Load SLA+W					SLA+W will be transmitted; I2C will be switched to MST/TRX mode.
Master Transmitter	18H	SLA+W has been transmitted; ACK has been received	Load data byte	0	0	0	X	Data byte will be transmitted; ACK will be received.
			No action	1	0	0	X	Repeated START will be transmitted.
			No action	0	1	0	X	STOP condition will be transmitted; STO flag will be reset.
			No action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset.
	20H	SLA+W has been transmitted; not ACK has been received	Load data byte*	0	0	0	X	Data byte will be transmitted; ACK will be received.
			No action	1	0	0	X	Repeated START will be transmitted.
			No action	0	1	0	X	STOP condition will be transmitted; STO flag will be reset.
			No action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset.
	28H	Data byte in I2CDAT has been transmitted; ACK has been received	Load data byte	0	0	0	X	Data byte will be transmitted; ACK bit will be received.
			No action	1	0	0	X	Repeated START will be transmitted.
			No action	0	1	0	X	STOP condition will be transmitted; STO flag will be reset.
			No action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset.
	30H	Data byte in I2CDAT has been transmitted; not ACK has been received	Load data byte*	0	0	0	X	Data byte will be transmitted; ACK will be received.
			No action	1	0	0	X	Repeated START will be transmitted.
			No action	0	1	0	X	STOP condition will be transmitted; STO flag will be reset.
			No action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset.
Master Receiver	40H	SLA+R has been transmitted; ACK has been received	No action	0	0	0	0	Data byte will be received; not ACK will be returned
			No action	0	0	0	1	Data byte will be received; ACK will be returned
	48H	SLA+R has been transmitted; not ACK has been received	No action	1	0	0	X	Repeated START condition will be transmitted
			No action	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
			No action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
	50H	Data byte has been received; ACK has been returned	Read data byte	0	0	0	0	Data byte will be received; not ACK will be returned
			Read data byte	0	0	0	1	Data byte will be received; ACK will be returned
	58H	Data byte has been received; not ACK has been returned	Read data byte	1	0	0	X	Repeated START condition will be transmitted
			Read data byte	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
			Read data byte	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset

Mode	Status Code	Status of the I2C	Application software response					Next action taken by I2C hardware
			To/from I2CDAT	TO I2CCON				
				STA	STO	SI	AA	
Slave Receiver	60H	Own SLA+W has been received; ACK has been returned	No action	X	0	0	0/1	Data byte will be received and not ACK/ACK will be returned
	68H	Arbitration lost in SLA+R/W as master; own SLA+W has been received, ACK returned	No action	X	0	0	0/1	Data byte will be received and not ACK/ACK will be returned
	70H	General call address (00H) has been received; ACK has been returned	No action	X	0	0	0/1	Data byte will be received and not ACK/ACK will be returned
	78H	Arbitration lost in SLA+R/W as master; general call address has been received, ACK returned	No action	X	0	0	0/1	Data byte will be received and not ACK/ACK will be returned
	80H	Previously addressed with own SLV address; DATA has been received; ACK returned	Read data byte	X	0	0	0/1	Data byte will be received and not ACK/ACK will be returned
	88H	Previously addressed with own SLA; DATA byte has been received; not ACK returned	Read data byte	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address
			Read data byte	0	0	0	1	Switched to not addressed SLV mode; own SLA or general call address will be recognized
			Read data byte	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address; START condition will be transmitted when the bus becomes free
			Read data byte	1	0	0	1	Switched to not addressed SLV mode; own SLA or general call address will be recognized; START condition will be transmitted when the bus becomes free
	90H	Previously addressed with general call address; DATA has been received; ACK returned	Read data byte	X	0	0	0/1	Data byte will be received and not ACK/ACK will be returned
	98H	Previously addressed with general call address; DATA has been received; not ACK returned	Read data byte	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address
			Read data byte	0	0	0	1	Switched to not addressed SLV mode; own SLA or general call address will be recognized
			Read data byte	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address; START condition will be transmitted when the bus becomes free
			Read data byte	1	0	0	1	Switched to not addressed SLV mode; own SLA or general call address will be recognized; START condition will be transmitted when the bus becomes free
	A0H	A STOP condition or repeated START condition has been received while still addressed as SLV/REC or SLV/TRX	No action	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address
			No action	0	0	0	1	Switched to not addressed SLV mode; own SLA or general call address will be recognized
			No action	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address; START condition will be transmitted when the bus becomes free
			No action	1	0	0	1	Switched to not addressed SLV mode; own SLA or general call address will be recognized; START condition will be transmitted when the bus becomes free
Slave Transmitter	A8H	Own SLA+R has been received; ACK has been returned	Load data byte	X	0	0	0	Last data byte will be transmitted and ACK will be received
			Load data byte	X	0	0	1	Data byte will be transmitted; ACK will be received.
	B0H	Arbitration lost in SLA+R/W as master; own SLA+R has been received, ACK has been returned.	Load data byte	X	0	0	0	Last data byte will be transmitted and ACK will be received
			Load data byte	X	0	0	1	Data byte will be transmitted; ACK will be received.
	B8H	Data byte has been transmitted; ACK will be received.	Load data byte	X	0	0	0	Last data byte will be transmitted and ACK will be received
			Load data byte	X	0	0	1	Data byte will be transmitted; ACK will be received.
	C0H	Data byte has been transmitted; not ACK has been received.	No action	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address.
			No action	0	0	0	1	Switched to not addressed SLV mode; own SLA or general call address will be recognized.
			No action	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address; START condition will be transmitted when the bus becomes free.
			No action	1	0	0	1	Switched to not addressed SLV mode; own SLA or general call address will be recognized; START condition will be transmitted when the bus becomes free.
	C8H	Last data byte has been	No action	0	0	0	0	Switched to not addressed SLV mode; no recognition of own

		transmitted; ACK has been received.						SLA or general call address.
			No action	0	0	0	1	Switched to not addressed SLV mode; own SLA or general call address will be recognized.
			No action	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address; START condition will be transmitted when the bus becomes free.
			No action	1	0	0	1	Switched to not addressed SLV mode; own SLA or general call address will be recognized; START condition will be transmitted when the bus becomes free.
Miscellaneous	F8H	No relevant state information available; SI=0	No action	No action				Wait or proceed current transfer
	38H	Arbitration lost	No action	0	0	0	X	I2C will be released; A start condition will be transmitted.
			No action	1	0	0	X	When the bus becomes free. (enter to a master mode)
	00H	Bus error during MST or selected slave modes	No action	0	1	0	X	Only the internal hardware is affected in the MST or addressed SLV modes. In all cases, the bus is released and I2C is switched to the not addressed SLV mode. STO flag is reset.

"SLA" means slave address, "R" means R/W=1, "W" means R/W=0

*For applications where NACK doesn't mean the end of communication.

SMBSEL Register (0xDE)

Bit	Field	Type	Initial	Description
7	SMBEXE	R/W	0	SMBus extension functionality 0: Disable 1: Enable
Else	Reserved	R/W	0	
3	I2CMX	R/W	0	I2C IO select bit. 0:I2C SCL is P1.5/ SDA is P1.6. 1: I2C SCL(1) is P3.2/ SDA(1) is P3.3.
2..0	SMBTOP[2:0]	R/W	000	SMBus timeout register

SMBDST Register (0xDF)

Bit	Field	Type	Initial	Description
7..0	SMBD[7:0]	R/W	0x00	This register is used to provide a read/write access port to the SMBus timeout registers. Data read or written to that register is actually read or written to the Timeout Register which is pointed by the SMBSEL register.

IEN0 Register (0xA8)

Bit	Field	Type	Initial	Description
7	EAL	R/W	0	Interrupts enable. Refer to Chapter Interrupt
Else				Refer to other chapter(s)

IEN1 Register (0xB8)

Bit	Field	Type	Initial	Description

0	EI2C	R/W	0	Interrupts enable. Refer to Chapter Interrupt
Else				Refer to other chapter(s)

P1M Register (0xFA)

Bit	Field	Type	Initial	Description
5	P16M	R/W	0	0: Set P1.6 (SDA) as input mode (required) ~~1: Set P1.6 (SDA) as output mode~~*
4	P15M	R/W	0	0: Set P1.5 (SCL) as input mode (required) ~~1: Set P1.5 (SCL) as output mode~~*
Else				Refer to other chapter(s)

* The P16M and P15M require be set input mode.

P3M Register (0xFC)

Bit	Field	Type	Initial	Description
3	P33M	R/W	0	0: Set P3.3 (SDA) as input mode (required) ~~1: Set P3.3 (SDA) as output mode~~*
2	P32M	R/W	0	0: Set P3.2 (SCL) as input mode (required) ~~1: Set P3.2 (SCL) as output mode~~
Else				Refer to other chapter(s)

* The P33M and P32M require be set input mode.

23 CRC

The CRC (cyclic redundancy check) calculation unit is used to get a CRC code from 8-bit data word and a generator polynomial. Among other applications, CRC-based techniques are used to verify data transmission or storage integrity. The CRC calculation circuit helps compute a signature of the software during runtime, to be compared with a reference signature generated at link time and stored at a given memory location.

The CRC generator keeps running in IDLE mode but doesn't have wake-up function. In stop mode, the CRC generator is stop.

- Support
  CRC-16 polynomial: X16+X15+X2+1
  CRC-16-CCITT polynomial: X16+X12+X5+1
- Handles 8-bit data size
- Single input/output 8-bit data register
- Input buffer to avoid bus stall during calculation
- 32KB ROM calculating time is 32ms.

23.1 CRC Calculation Procedure

The CRC generator supports 2 calculation modes to obtain CRC code including EEPROM auto-calculation mode and data calculation mode.

EEPROM auto-calculation mode procedure:

- Support Select CRC execute range by CRCROM bit. If CRCROM = 0, the CRC executes range is 0x0000~0x1FFF. Otherwise, the CRC executes range is 0x0000~0x3FFF.
- Select CRC polynomial by CRCPOL bit. When CRCPOL=0, CRC-16-CCITT (X16+X12+X5+1) is selected. Otherwise, the CRC-16(X16+X15+X2+1) is selected.
- Set CRCRST bit to reset initial seed value/ CRC data buffer and BUSY bit to 0.
- Set URCRCEN bit to start CRC calculation and check BUSY bit.
- CRC calculation is finished until BUSY bit from 1 to 0. Read CRC code from CRCDATH/L.

Data calculation mode procedure:

- Select CRC polynomial by CRCPOL bit. When CRCPOL=0, CRC-16-CCITT (X16+X12+X5+1) is selected. Otherwise, the CRC-16(X16+X15+X2+1) is selected.
- Set CRCRST bit to reset initial seed value/ CRC result and BUSY bit to 0.
- Key in data to CRCDATL.
- Set URCRCEN bit to start CRC calculation and check BUSY bit.
- CRC calculation is finished until BUSY bit from 1 to 0. Read CRC code from CRCDATH/L

23.2 CRC Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CRCM	URCRCEN	BUSY	CRCRST	CRCPOL	CRCROM1	CRCROM0	-	CRCEN
CRCDATL	CRCDAT7	CRCDAT6	CRCDAT5	CRCDAT4	CRCDAT3	CRCDAT2	CRCDAT1	CRCDAT0
CRCDATH	CRCDAT15	CRCDAT14	CRCDAT13	CRCDAT12	CRCDAT11	CRCDAT10	CRCDAT9	CRCDAT8

CRCM Register (0xF1)

Bit	Field	Type	Initial	Description
7	URCRCEN	R/W	0	Enable bit of the CRC calculation for the 16KB/32KB User ROM 0: Disable 1: Enable
6	BUSY	R/W	0	Disable. Stop/Finish operation 0: CRC calculation finished. 1: CRC calculation is in process.
5	CRCRST	R/W	0	Reset bit. 0: No effect. 1: Reset the CRC calculation circuit.
4	CRCPOL	R/W	0	CRC Polynomial 0: CRC-16-CCITT 1: CRC-16
3..2	CRCROM [1:0]	R/W	0x00	CRC EEPROM calculation range. 00: 0x0000~0x0FFF (1/4 ROM). 01: 0x0000~0x1FFF (1/2 ROM). 10: 0x0000~0x3EFF(Reserve last 8 page) 11:0x0000~0x3FDD(Reserve last 1 page + 2bytes)
0	CRCEN	R/W	0	Enable bit of the CRC IP 0: Disable. 1: Enable. **(Only support "IHRC Mode")**

CRCDATL Register (0xF2)

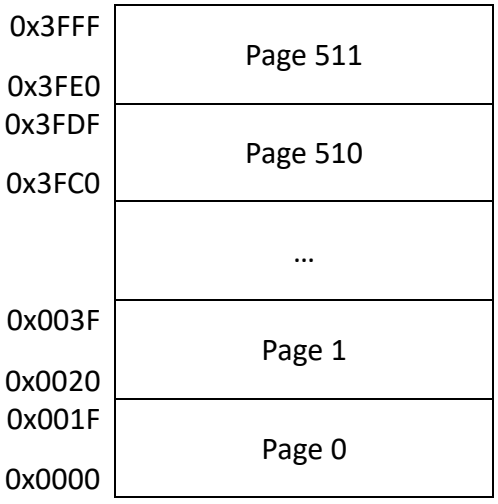
Bit	Field	Type	Initial	Description
7..0	CRCDAT[7:0]	R/W	0x00	Low byte of 16-bit CRC data.

CRCDATH Register (0xF3)

Bit	Field	Type	Initial	Description
7..0	CRCDAT[15:8]	R/W	0x00	High byte of 16-bit CRC data.

24 In-System Program

SN8F5754 builds in an on-chip 16 KB program memory, aka IROM, which is equally divided to 512 pages (32 bytes per page). The in-system program is a procedure that enables a firmware to freely modify every page's data; in other word, it is the channel to store value(s) into the non-volatile memory and/or live update firmware.



Program memory (IROM)

24.1 Page Program

Because each page of the program memory has 32 bytes in length, a page program procedure requires 32 bytes IRAM as its data buffer.

ISP ROM MAP		ROM address bit0~bit5 (hex) =0
ROM address bit6~bit15 (hex)	0000	These pages include reset vector and interrupt sector. We strongly recommend to reserve the area not to do ISP erase.
	0020	
	…	
	00C0	
	00E0	
	0100	One ISP Program Page
	0140	One ISP Program Page
	…	One ISP Program Page
	2000	One ISP Program Page
	2020	One ISP Program Page
	2040	One ISP Program Page
	…	One ISP Program Page
	3FA0	One ISP Program Page
	3FC0	One ISP Program Page
	3FE0	This page includes ROM reserved area. We strongly recommend to reserve the area not to do ISP erase.

These configurations must be setup completely before starting Page Program. ISP is configured using the following steps:

1. Save program data into IRAM. The data continues for 32 bytes.
2. Set the start address of the content location to PERAM.
3. Set the start address of the anticipated update area to PEROM [15:6]. (By PEROMH/PRROML registers)
4. Write '0x5A' into PECMD [7:0] to trigger ISP function.
5. Write 'NOP' instruction twice.

As an example, assume the 510th page of program memory (IROM, 0x3FC0 – 0x3FDF) is the anticipated update area; the content is already stored in IRAM address 0x40 – 0x5F. To perform the in-system program, simply write starting IROM address 0x3FC0 to EPROMH/EPROML registers, and then specify buffer starting address 0x40 to EPRAM register. Subsequently, write '0x5A' into PECMD [7:0] registers to duplicate the buffer's data to 510th page of IROM.

In general, every page has the capability to be modified by in-system program procedure. However, since the first and least pages (page 0 and 511) respectively stores reset vector and information for power-on controller, incorrectly perform page program (such as turn off power while programming) may cause faulty power-on sequence / reset.

✷ ***Note:***

1. Watch dog timer should be clear before the Flash write (program) operation, or watchdog timer would overflow and reset system during ISP operating.

2. Don't execute ISP flash ROM program operation for the first page and the last page, or affect program operation.

3. ISP operation (page program) actually perform Flash erase and program procedures in the background. Don't execute ISP flash ROM program operation in low-voltage condition (ex. VDD < 2.5V), or ISP operation maybe not complete before power-off.

24.2 In-system Program Register

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PERAM	PERAM7	PERAM6	PERAM5	PERAM4	PERAM3	PERAM2	PERAM1	PERAM0
PEROMH	PEROM15	PEROM14	PEROM13	PEROM12	PEROM11	PEROM10	PEROM9	PEROM8
PEROML	PEROM7	PEROM6	PEROM5	PEROM4	PEROM3	PEROM2	PEROM1	PEROM0

PECMD	PECMD7	PECMD6	PECMD5	PECMD4	PECMD3	PECMD2	PECMD1	PECMD0

PERAM Register (0xBD)

Bit	Field	Type	Initial	Description
7..0	PERAM[7:0]	R/W	0x00	The first address of data buffer (IRAM)

PEROMH Register (0xBC)

Bit	Field	Type	Initial	Description
7..0	PEROM[15:8]	R/W	0x00	The first address (15^{th}– 8^{th} bit) of program page (IROM)

PEROML Register (0xBB)

Bit	Field	Type	Initial	Description
7..0	PEROM[7:0]	R/W	000	The first address (7^{th}– 0^{th} bit) of program page (IROM)

PECMD Register (0xBA)

Bit	Field	Type	Initial	Description
7..0	PECMD[7:0]	W	-	0x5A: Start page program procedure*(1)

*(1) Not permitted to write any other to PECMD register.

25 Clock Fine-Tuning

SN8F5754 builds in clock fine-tuning function that is a procedure to fine-tune system clock frequency by firmware. The function is enabled by code option (CK_Fine_Tuning). When CK_Fine_Tuning = 0, the clock fine-tuning function is disabled. When CK_Fine_Tuning = 1, the clock fine-tuning function is enabled. After system power-on, the 10-bit initial clock trim value will be loaded to FRQ[9:0] buffer by hardware. The trim value corresponds to IHRC 32MHz. Change the trim value of FRQ[9:0] to modify internal clock frequency.

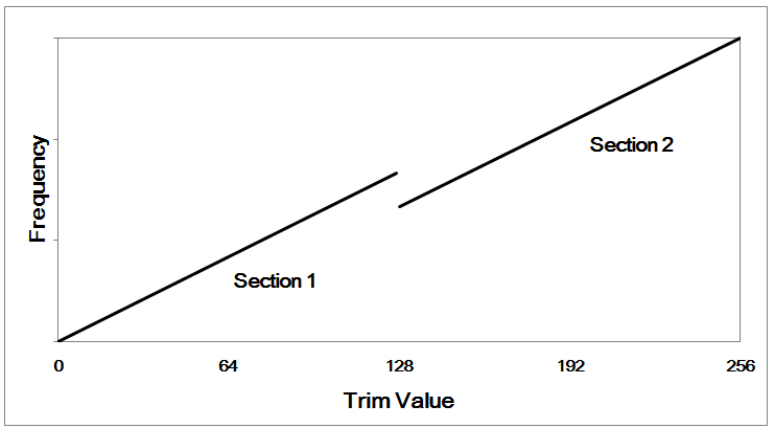
25.1 Clock Trim Section

Clock fine-tuning consists of 8 trim sections as Table 25-1. Each section includes 128 trim steps and each step is about (32MHz *0.1%) in the same section. The larger the Trim value, the faster the frequency.

Table 25-1 Clock Trim Section

Section	Trim Value	Frequency
1	000H ~ 07FH	Low
2	080H ~ 0FFH	
3	100H ~ 17FH	
4	180H ~ 1FFH	
5	200H ~ 27FH	
6	280H ~ 2FFH	
7	300H ~ 37FH	
8	380H ~ 3FFH	High

Each adjacent section has a frequency gap as below. Thus the frequency in trim value =127 will faster than trim value =128.



25.2 Clock Fine-Tuning Procedure

Theses configurations must be setup completely before starting clock fine-tuning. The steps as the following:

1. Select code option CK_Fine_Tuning = 1 to enable clock fine-tuning function.
2. As the Max. IROM fetching cycle is 16MHz, it is recommended to set PWSC[2:0]=7 at first to avoid system error.
3. Read 10-bit 32MHz trim value from FRQ[9:0]
4. Check the fine-tuning range from the Table 25-1.
5. Write the new clock trim value to FRQ[9:0].
6. Write '0x3C' into FRQCMD [7:0] to trigger clock fine-tuning function.

✸ ***Note: Please check IROM fetching cycle ≦ 16MHz to avoid system error.***

25.3 Clock Fine-Tuning Register

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FRQH	-	-	-	-	-	-	FRQ9	FRQ8
FRQL	FRQ7	FRQ6	FRQ5	FRQ4	FRQ3	FRQ2	FRQ1	FRQ0
FRQCMD	FRQCMD7	FRQCMD6	FRQCMD5	FRQCMD4	FRQCMD3	FRQCMD2	FRQCMD1	FRQCMD0

FRQ Register (FRQH:0xAB, FRQL:0xAA)

Bit	Field	Type	Initial	Description
9..0	FRQ[9:0]	R/W	0x00	The system clock calibration value

FRQCMD Register (0xAC)

Bit	Field	Type	Initial	Description
7..0	FRQCMD[7:0]	W	0x00	0x3C: Start clock fine-tuning procedure Else values: Reserved*(1)

*(1) Not permitted to write any other to FRQCMD register.

26 ILRC Auto-Calibration

26.1 Auto-Calibration Operation

ILRC auto-calibration function is hardware calibration to support 16KHz ILRC. The calibration function is enabled by CALEN bit. After setup CALEN bit, the system starts to calibrate ILRC frequency to 16KHz. Calibration time is about 20ms. The CALEN bit is reset to logic 0 when the calibration is complete. After the calibration is complete, the circuit will set CALDONE bit if ILRC calibration success.

26.2 Clock Fine-Tuning Register

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LFRQH	-	-	-	-	-	-	FRQ9	FRQ8
LFRQL	FRQ7	FRQ6	FRQ5	FRQ4	FRQ3	FRQ2	FRQ1	FRQ0
FRQCMD	FRQCMD7	FRQCMD6	FRQCMD5	FRQCMD4	FRQCMD3	FRQCMD2	FRQCMD1	FRQCMD0
CLKCAL	CALEN	CALDONE	-	-	-	-	-	-

LFRQ Register (FRQH:0xAD, FRQL:0xAE)

Bit	Field	Type	Initial	Description
8..0	LFRQ[8:0]	R/W	0x00	The system ILRC calibration value

FRQCMD Register (0xAC)

Bit	Field	Type	Initial	Description
7..0	FRQCMD[7:0]	W	0x00	0x4B: Start ILRC calibration procedure Else values: Reserved

CLKCAL Register (0xAF)

Bit	Field	Type	Initial	Description
7	CALEN	R/W	0	ILRC calibration enable bit 0: Disable 1: Enable (automatically cleared by the end of calibration)
6	CALDONE	R/W	0	ILRC calibration result 0: Calibration fail. 1: Calibration pass, ILRC frequency is 16KHz

27 Single Wire Asynchronous Interface (SWAT)

SWAT function is a single wire transmission interface for system debug mode.

- Support single-wire debug interface.
- Direct debug access to all memories, registers and peripherals.
- Up two breakpoints.

27.1 Debug Mode

System debug mode is default enabled. When system power on, SWAT function is enabled and P1.2 acts as SWAT pin for debug interface. User can disable SWAT function by DEGCMD register in order to use P1.2 as GPIO. When DEGCMD= 0x00, SWAT function is disabled and P1.2 acts as GPIO. User can set DEGCMD= 0xA5 to enable SWAT function again. SWAT function can't active under idle/ stop mode. User code must disable SWAT function before system enters idle/ stop mode.

27.2 Internal Pull-Up Resistors on SWAT pin

To avoid any uncontrolled IO levels, the device embeds internal pull-up resistor on SWAT input pin. Once a SWAT function is disabled by DEGCMD register, the GPIO controller takes control again.

27.3 SWAT Register

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DEGCMD	DEGCMD7	DEGCMD6	DEGCMD5	DEGCMD4	DEGCMD3	DEGCMD2	DEGCMD1	DEGCMD0

DEGCMD Register (0xBF)

Bit	Field	Type	Initial	Description
7..0	DEGCMD[7:0]	R/W	0xA5	SWAT pin control 0xA5: Enable SWAT pin. (P1.2 acts as SWAT pin and internal pull-up is enabled) 0x00: Disable SWAT pin. (P1.2 acts as GPIO pin and internal pull-up is disabled) Else values: Reserved*(1)

*(1) Not permitted to write any other to DEGCMD register.

28 Electrical Characteristics

28.1 Absolute Maximum Ratings

Voltage applied at VDD to VSS ..... - 0.3V to 6.0V
Voltage applied at any pin to VSS..... - 0.3V to VDD+0.3V
Operating ambient temperature.....-40°C to 85°C
Storage ambient temperature ..... -40°C to 125°C

28.2 System Operation Characteristics

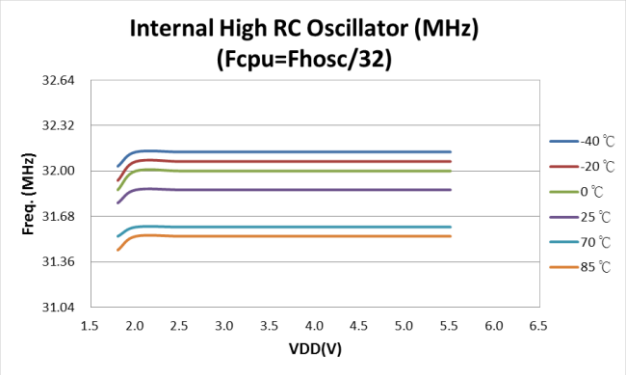
	Parameter	Test Condition	Min	TYP	MAX	UNIT
VDD	Operating voltage	fcpu = 1MHz	1.8		5.5	V
V_{DR}	RAM data retention Voltage		0.55			V
V_{POR}	VDD rising rate*		0.05			V/ms
I_{DD1}	Normal mode supply current (32MHz IHRC)	VDD = 3V, fcpu = 1MHz		2.31		mA
		VDD = 5V, fcpu = 1MHz		2.32		mA
		VDD = 3V, fcpu = 4MHz		2.82		mA
		VDD = 5V, fcpu = 4MHz		2.84		mA
		VDD = 3V, fcpu = 8MHz		3.51		mA
		VDD = 5V, fcpu = 8MHz		3.52		mA
	Normal mode supply current (16MHz Crystal)	VDD = 3V, fcpu = 1MHz		2.18		mA
		VDD = 5V, fcpu = 1MHz		2.78		mA
		VDD = 3V, fcpu = 4MHz		2.71		mA
		VDD = 5V, fcpu = 4MHz		3.30		mA
		VDD = 3V, fcpu = 8MHz		3.39		mA
		VDD = 5V, fcpu = 8MHz		3.98		mA
I_{DD2}	STOP mode supply current	VDD = 3V		1.7	5.5	µA
		VDD = 5V		1.9	6.0	µA
I_{DD3}	STOP mode supply current (ILRC enable STWK=1)	VDD = 5V		2.5		µA
I_{DD5}	IDLE mode supply current (fcpu = 1MHz)	VDD = 3V, 32MHz IHRC		0.81		mA
		VDD = 5V, 32MHz IHRC		0.82		mA
		VDD = 3V, 16MHz Crystal		0.71		mA
		VDD = 5V, 16MHz Crystal		1.31		mA
F_{IHRC}	Internal high clock generator	VDD = 1.8V to 5.5V, 25°C	31.84	32	32.16	MHz
		VDD = 1.8V to 5.5V, -40°C to 85°C	31.36	32	32.64	MHz

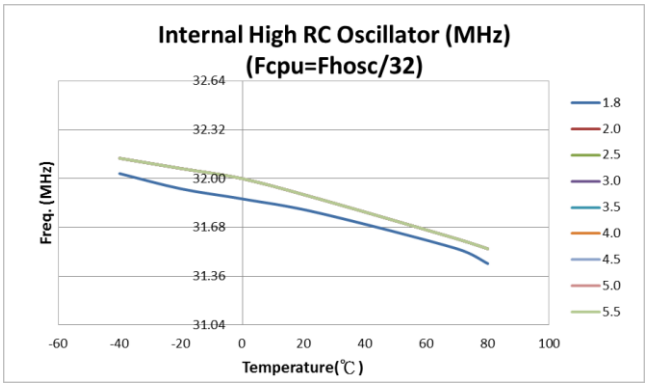
V_{LVD18}	LVD18 detect voltage	25°C	1.7	1.8	1.9	V
		-40°C to 85°C	1.6	1.8	2.0	V
V_{ESD_HBM}	ESD human body mode		4000			V
V_{ESD_MM}	ESD machine mode		400			V

* Parameter(s) with star mark are non-verified design reference. Ambient temperature is 25°C.

- IHRC Frequency - Temperature Graph

  The IHRC Graphs are for design guidance, not tested or guaranteed. In some graphs, the data presented are outside specified operating range. This is for information only and devices are guaranteed to operate properly only within the specified range.





28.3 GPIO Characteristics

	Parameter	Test Condition	MIN	TYP	MAX	UNIT
V_{IL}	Low-level input voltage		VSS		0.3VDD	V
V_{IH}	High-level input voltage		0.7VDD		VDD	V
I_{LEKG}	I/O port input leakage current	V_{IN} = VDD			2	µA
R_{UP}	Pull-up resister	VDD = 3V	100	200	300	kΩ
		VDD = 5V	50	100	150	kΩ
I_{OH1}	I/O output source current (P0, P1, P2, P3)	VDD = 5V, V_O = VDD-0.5V	12	15		mA
I_{OL1}	I/O sink current (P04-P07, P14-P17, P2, P3)	VDD = 5V, V_O = VSS+0.5V	15	20		mA
I_{OL2}	I/O sink current (P00-P03, P10-P13)	VDD = 5V, V_O = VSS+0.5V	80	100		mA
V_{bias}	1/2*VDD Bias Voltage (P00-P03, P10-P13)	VDD = 5V	2.4	2.5	2.6	V

* Ambient temperature is 25°C.

28.4 ADC Characteristics

	Parameter	Test Condition	MIN	TYP	MAX	UNIT
V_{ADC}	Operating voltage		2.0		5.5	V
V_{AIN}	AIN channels input voltage	VDD = 5V	0		V_{REFH}	V
V_{REFH}	AVREFH pin input voltage	VDD = 5V	2.0		VDD	V
V_{IREF}	Internal VDD reference voltage	VDD = 5V		VDD		V
	Internal 4.5V reference voltage	VDD = 5V	4.455	4.5	4.545	V
	Internal 3.5V reference voltage	VDD = 5V	3.465	3.5	3.535	V
	Internal 2.5V reference voltage	VDD = 5V	2.487	2.5	2.513	V
I_{AD}	ADC current consumption	VDD = 3V		0.65		mA
		VDD = 5V		0.75		mA
f_{ADCLK}	ADC clock	VDD = 5V			16	MHz
f_{ADSMP}	ADC sampling rate	VDD = 5V			250	kHz
t_{ADEN}	ADC function enable period	VDD = 5V	100			µs
DNL	Differential nonlinearity*	f_{ADSMP} = 62.5kHz		±1		LSB
		f_{ADSMP} = 250kHz		±1		LSB
INL	Integral Nonlinearity*	f_{ADSMP} = 62.5kHz		±1.5		LSB
		f_{ADSMP} = 250kHz		±1.5		LSB
NMC	No missing code*	f_{ADSMP} = 62.5kHz	10	11	12	Bit
		f_{ADSMP} = 250kHz		11		Bit
V_{OFFSET}	Input offset voltage	Non-trimmed	-10	0	10	mV

* Parameters with star mark: VDD = 5V, V_{REFH} = 2.4V, 25°C.

28.5 Flash Memory Characteristics

	Parameter	Test Condition	MIN	TYP	MAX	UNIT
V_{dd}	Supply voltage		1.8		5.5	V
T_{en}	Endurance time	25°C		*100K		cycle
I_{wrt}	Write current	25°C		3	4	mA
T_{wrt}	Write time	Write 1 page=32 bytes, 25°C		6	8	ms

* Parameters with star mark are non-verified design reference.

29 Instruction Set

This chapter categorizes the SONIX microcontroller's comprehensive assembly instructions. It includes five categories—arithmetic operation, logic operation, data transfer operation, Boolean manipulation, and program branch—which are fully compatible with standard 8051.

Symbol description

Symbol	Description
Rn	Working register R0 - R7
direct	One of 128 internal RAM locations or any Special Function Register
@Ri	Indirect internal or external RAM location addressed by register R0 or R1
#data	8-bit constant (immediate operand)
#data16	16-bit constant (immediate operand)
bit	One of 128 software flags located in internal RAM, or any flag of bit-addressable Special Function Registers
addr16	Destination address for LCALL or LJMP, can be anywhere within the 64-Kbyte page of program memory address space
addr11	Destination address for ACALL or AJMP, within the same 2-Kbyte page of program memory as the first byte of the following instruction
rel	SJMP and all conditional jumps include an 8-bit offset byte. Its range is +127/-128 bytes relative to the first byte of the following instruction
A	Accumulator

Arithmetic operations

Mnemonic	Description
ADD A, Rn	Add register to accumulator
ADD A, direct	Add directly addressed data to accumulator
ADD A, @Ri	Add indirectly addressed data to accumulator
ADD A, #data	Add immediate data to accumulator
ADDC A, Rn	Add register to accumulator with carry
ADDC A, direct	Add directly addressed data to accumulator with carry
ADDC A, @Ri	Add indirectly addressed data to accumulator with carry
ADDC A, #data	Add immediate data to accumulator with carry
SUBB A, Rn	Subtract register from accumulator with borrow
SUBB A, direct	Subtract directly addressed data from accumulator with borrow
SUBB A, @Ri	Subtract indirectly addressed data from accumulator with borrow
SUBB A, #data	Subtract immediate data from accumulator with borrow
INC A	Increment accumulator
INC Rn	Increment register
INC direct	Increment directly addressed location
INC @Ri	Increment indirectly addressed location
INC DPTR	Increment data pointer
DEC A	Decrement accumulator
DEC Rn	Decrement register
DEC direct	Decrement directly addressed location
DEC @Ri	Decrement indirectly addressed location
MUL AB	Multiply A and B
DIV	Divide A by B
DA A	Decimally adjust accumulator

Logic operations

Mnemonic	Description
ANL A, Rn	AND register to accumulator
ANL A, direct	AND directly addressed data to accumulator
ANL A, @Ri	AND indirectly addressed data to accumulator
ANL A, #data	AND immediate data to accumulator
ANL direct, A	AND accumulator to directly addressed location
ANL direct, #data	AND immediate data to directly addressed location
ORL A, Rn	OR register to accumulator

ORL A, direct	OR directly addressed data to accumulator
ORL A, @Ri	OR indirectly addressed data to accumulator
ORL A, #data	OR immediate data to accumulator
ORL direct, A	OR accumulator to directly addressed location
ORL direct, #data	OR immediate data to directly addressed location
XRL A, Rn	Exclusive OR (XOR) register to accumulator
XRL A, direct	XOR directly addressed data to accumulator
XRL A, @Ri	XOR indirectly addressed data to accumulator
XRL A, #data	XOR immediate data to accumulator
XRL direct, A	XOR accumulator to directly addressed location
XRL direct, #data	XOR immediate data to directly addressed location
CLR A	Clear accumulator
CPL A	Complement accumulator
RL A	Rotate accumulator left
RLC A	Rotate accumulator left through carry
RR A	Rotate accumulator right
RRC A	Rotate accumulator right through carry
SWAP A	Swap nibbles within the accumulator

Data transfer operations

Mnemonic	Description
MOV A, Rn	Move register to accumulator
MOV A, direct	Move directly addressed data to accumulator
MOV A, @Ri	Move indirectly addressed data to accumulator
MOV A, #data	Move immediate data to accumulator
MOV Rn, A	Move accumulator to register
MOV Rn, direct	Move directly addressed data to register
MOV Rn, #data	Move immediate data to register
MOV direct, A	Move accumulator to direct
MOV direct, Rn	Move register to direct
MOV direct1, direct2	Move directly addressed data to directly addressed location
MOV direct, @Ri	Move indirectly addressed data to directly addressed location
MOV direct, #data	Move immediate data to directly addressed location
MOV @Ri, A	Move accumulator to indirectly addressed location
MOV @Ri, direct	Move directly addressed data to indirectly addressed location
MOV @Ri, #data	Move immediate data to in directly addressed location

MOV DPTR, #data16	Load data pointer with a 16-bit immediate
MOVC A, @A+DPTR	Load accumulator with a code byte relative to DPTR
MOVC A, @A+PC	Load accumulator with a code byte relative to PC
MOVX A, @Ri	Move external RAM (8-bit address) to accumulator
MOVX A, @DPTR	Move external RAM (16-bit address) to accumulator
MOVX @Ri, A	Move accumulator to external RAM (8-bit address)
MOVX @DPTR, A	Move accumulator to external RAM (16-bit address)
PUSH direct	Push directly addressed data onto stack
POP direct	Pop directly addressed location from stack
XCH A, Rn	Exchange register with accumulator
XCH A, direct	Exchange directly addressed location with accumulator
XCH A, @Ri	Exchange indirect RAM with accumulator
XCHD A, @Ri	Exchange low-order nibbles of indirect and accumulator

Boolean manipulation

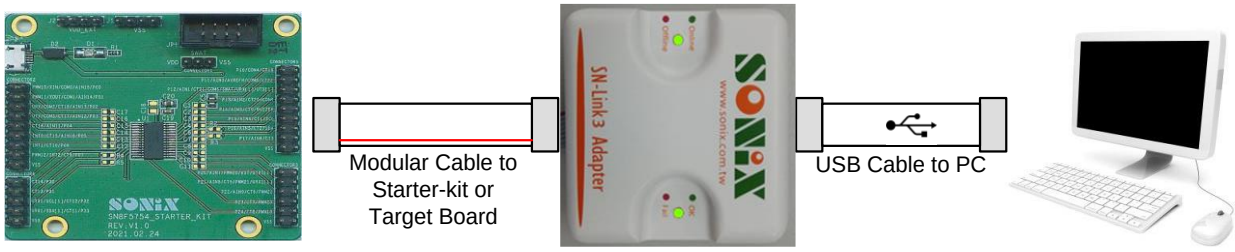
Mnemonic	Description
CLR C	Clear carry flag
CLR bit	Clear directly addressed bit
SETB C	Set carry flag
SETB bit	Set directly addressed bit
CPL C	Complement carry flag
CPL bit	Complement directly addressed bit
ANL C, bit	AND directly addressed bit to carry flag
ANL C, /bit	AND complement of directly addressed bit to carry
ORL C, bit	OR directly addressed bit to carry flag
ORL C, /bit	OR complement of directly addressed bit to carry
MOV C, bit	Move directly addressed bit to carry flag
MOV bit, C	Move carry flag to directly addressed bit

Program branches

Mnemonic	Description
ACALL addr11	Absolute subroutine call
LCALL addr16	Long subroutine call
RET	Return from subroutine
RETI	Return from interrupt
AJMP addr11	Absolute jump
LJMP addr16	Long jump
SJMP rel	Short jump (relative address)
JMP @A+DPTR	Jump indirect relative to the DPTR
JZ rel	Jump if accumulator is zero
JNZ rel	Jump if accumulator is not zero
JC rel	Jump if carry flag is set
JNC rel	Jump if carry flag is not set
JB bit, rel	Jump if directly addressed bit is set
JNB bit, rel	Jump if directly addressed bit is not set
JBC bit, rel	Jump if directly addressed bit is set and clear bit
CJNE A, direct, rel	Compare directly addressed data to accumulator and jump if not equal
CJNE A, #data, rel	Compare immediate data to accumulator and jump if not equal
CJNE Rn, #data, rel	Compare immediate data to register and jump if not equal
CJNE @Ri, #data, rel	Compare immediate to indirect and jump if not equal
DJNZ Rn, rel	Decrement register and jump if not zero
DJNZ direct, rel	Decrement directly addressed location and jump if not zero
NOP	No operation for one cycle

30 Development Environment

SONIX provides an Embedded ICE emulator system to offer SN8F5754 firmware development. The platform is an in-circuit debugger and controlled by Keil C51 IDE software on Microsoft Windows platform. The platform includes SN-Link3, SN8F5754 Starter-kit and Keil C51 IDE software to build a high-speed, low cost, powerful and multi-task development environment including emulator, debugger and programmer. To execute emulation is like run real chip because the emulator circuit integrated in SN8F5754 to offer a real development environment.



30.1 Minimum Requirement

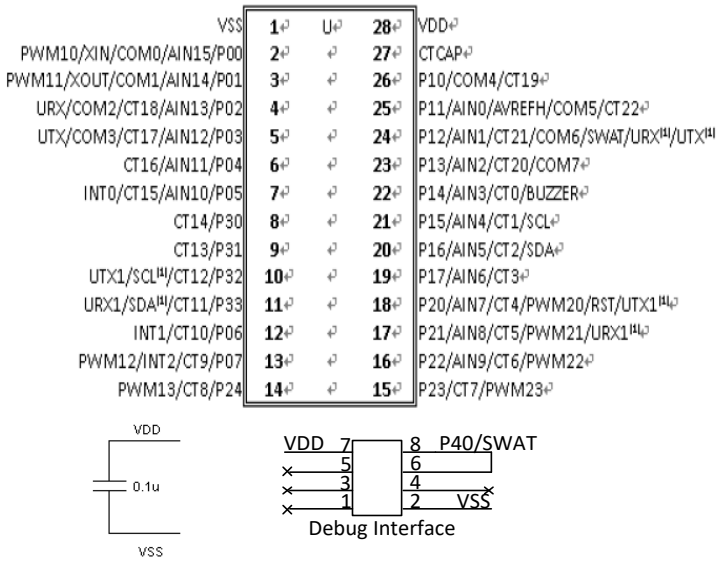
The following items are essential to build up an appropriate development environment. The compatibility is verified on listed versions, and is expected to execute perfectly on later version. SN-Link related information is available to download on SONiX website (www.sonix.com.tw);
Keil C51 is downloadable on www.keil.com/c51.

- **SN-Link3 Adapter** with updated firmware version 1.02
- **SN-Link Driver for Keil C51** version 1.00.317
- **Keil C51** version 9.50a and 9.54a or greater.

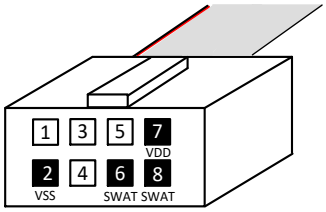
30.2 Debug Interface Hardware

The circuit below demonstrates the appropriate method to connect microcontroller's SWAT pin and SN-Link3 Adapter.

Before starting debug, microcontroller's power (VDD) must be switched off. Connect the SWAT to both 6th and 8th pins of SN-Link, and respectively link VDD and VSS to 7th pin and 2nd pin. A handshake procedure would be automatically started by turn on the microcontroller, and SN-Link's green LED (Run) indicates the success of connection (refer *SN8F5000 Debug Tool Manual* for further detail).



example circuit

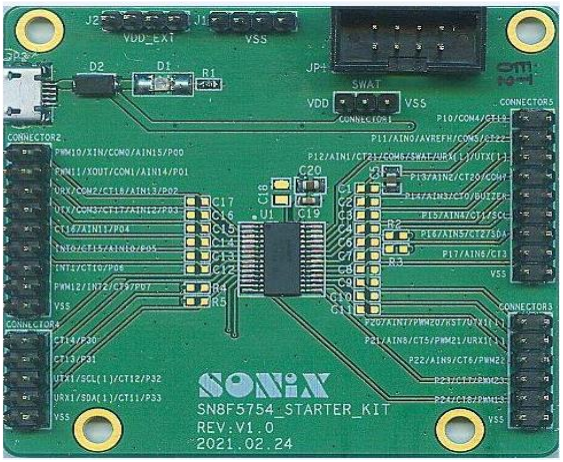


SN-Link header

30.3 Development Tool

SN-Link3 Adapter

Starter-Kit support SN8F5754, 5753, 5752



MP5 Writer

31 SN8F5754 Starter-Kit

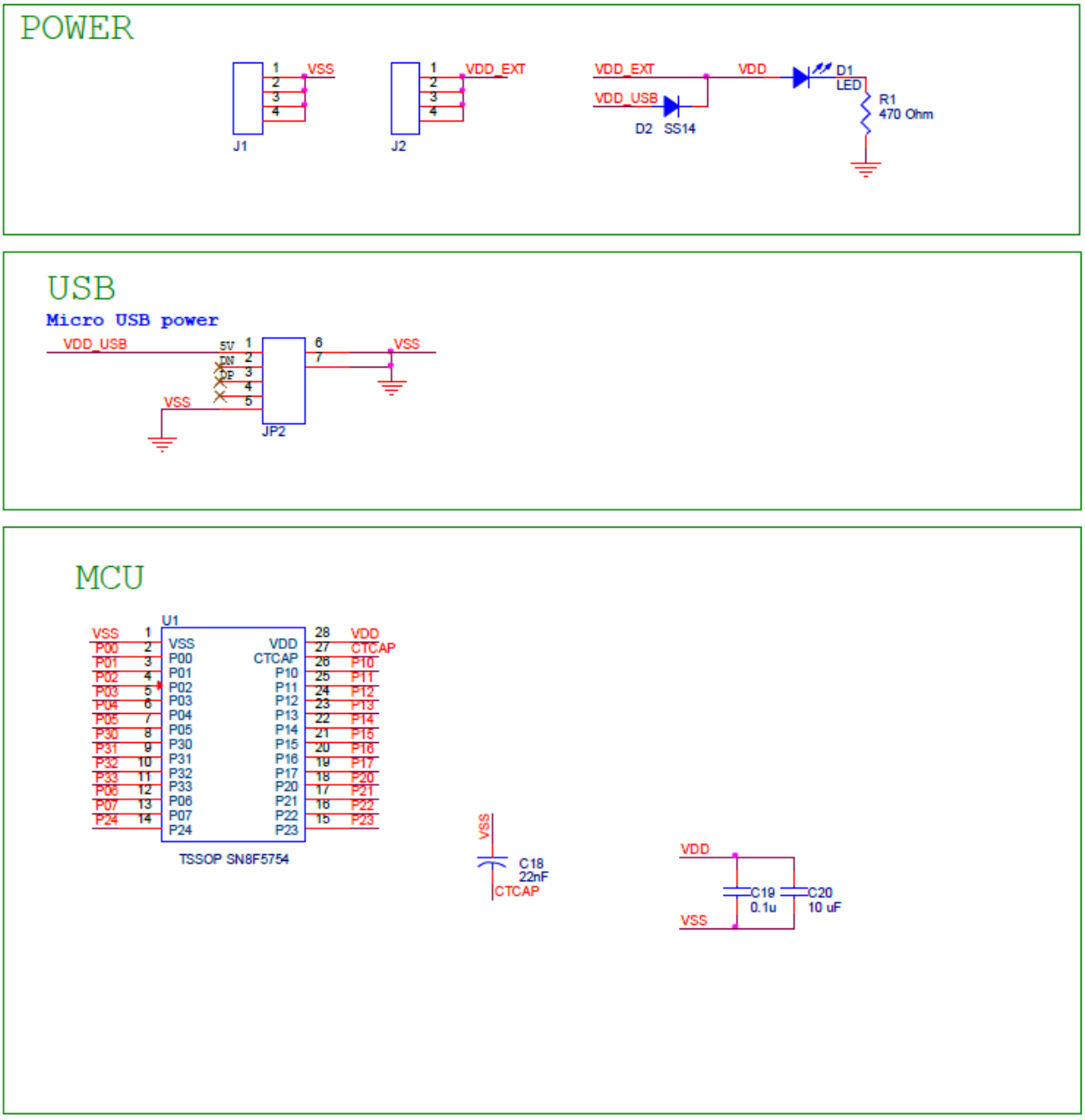
SN8F5000 Starter-Kit provides easy-development platform. It includes SN8F5000 family real chip and I/O connectors to input signal or drive device of user's application. It is a simple platform to develop application as target board not ready. The Starter-Kit can be replaced by target board, because SN8F5000 family integrates embedded ICE in-circuit debugger circuitry.

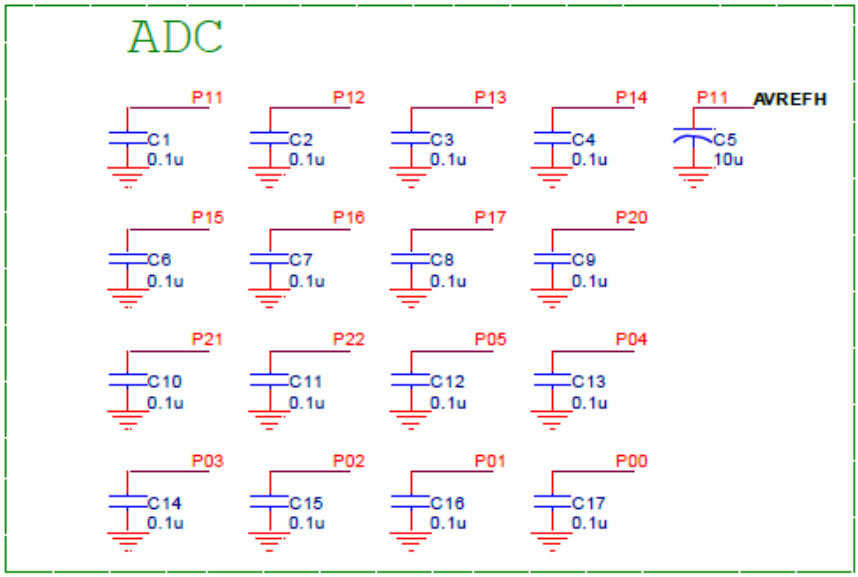
31.1 Configurations of Circuit

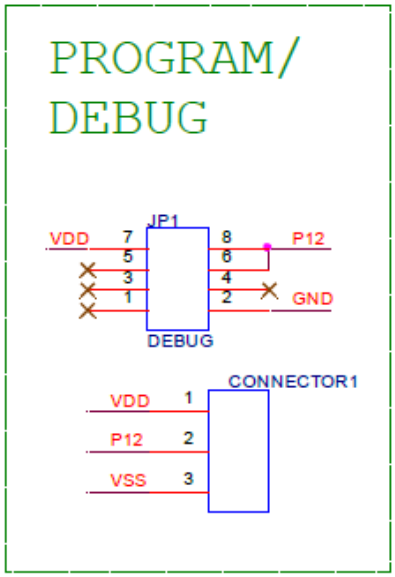
These configurations must be setup completely before starting Starter-Kit developing.

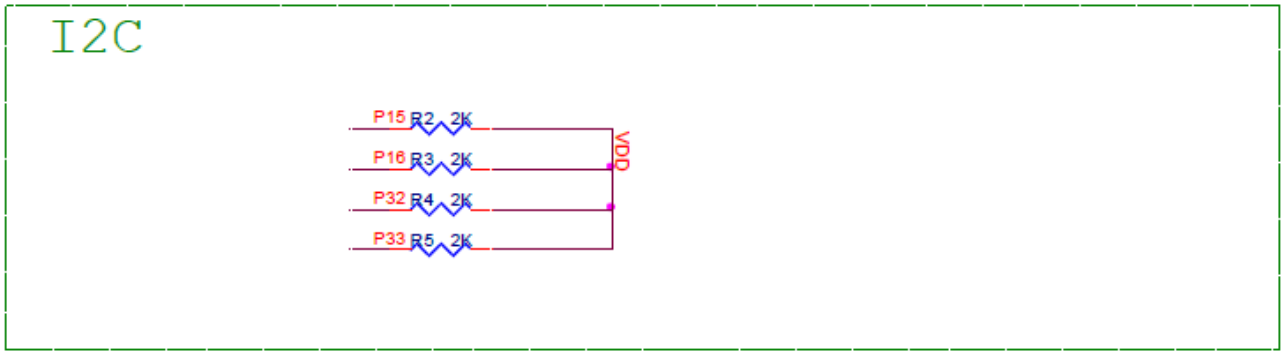
1. Confirm to the circuit board whether elements are complete.
2. The power source of Starter-Kit circuit is chosen from 5.0V, 3.3V, external power or Micro USB via jumper.
3. The "XIN" pin and the "XOUT" pin need to connect crystal/resonator oscillator components when system clock is setting crystal or RTC mode.
4. The "XIN" pin needs to connect external clock source when system clock is setting external clock input mode.
5. The Debug Port can connect SN-LINK Adapter for emulation or download code.
6. The MCU LED will light up and SN8F5000 family chip will be connected to power when power (VDD) is switched on.

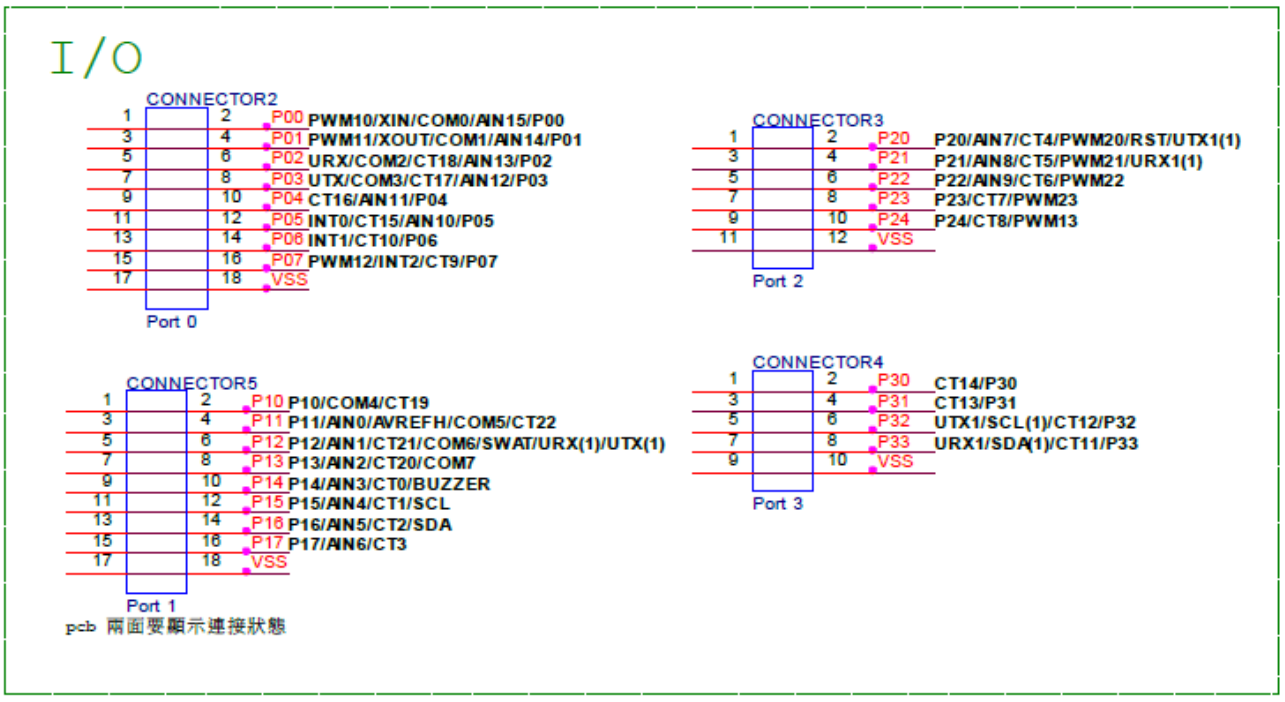
31.2 Schematic



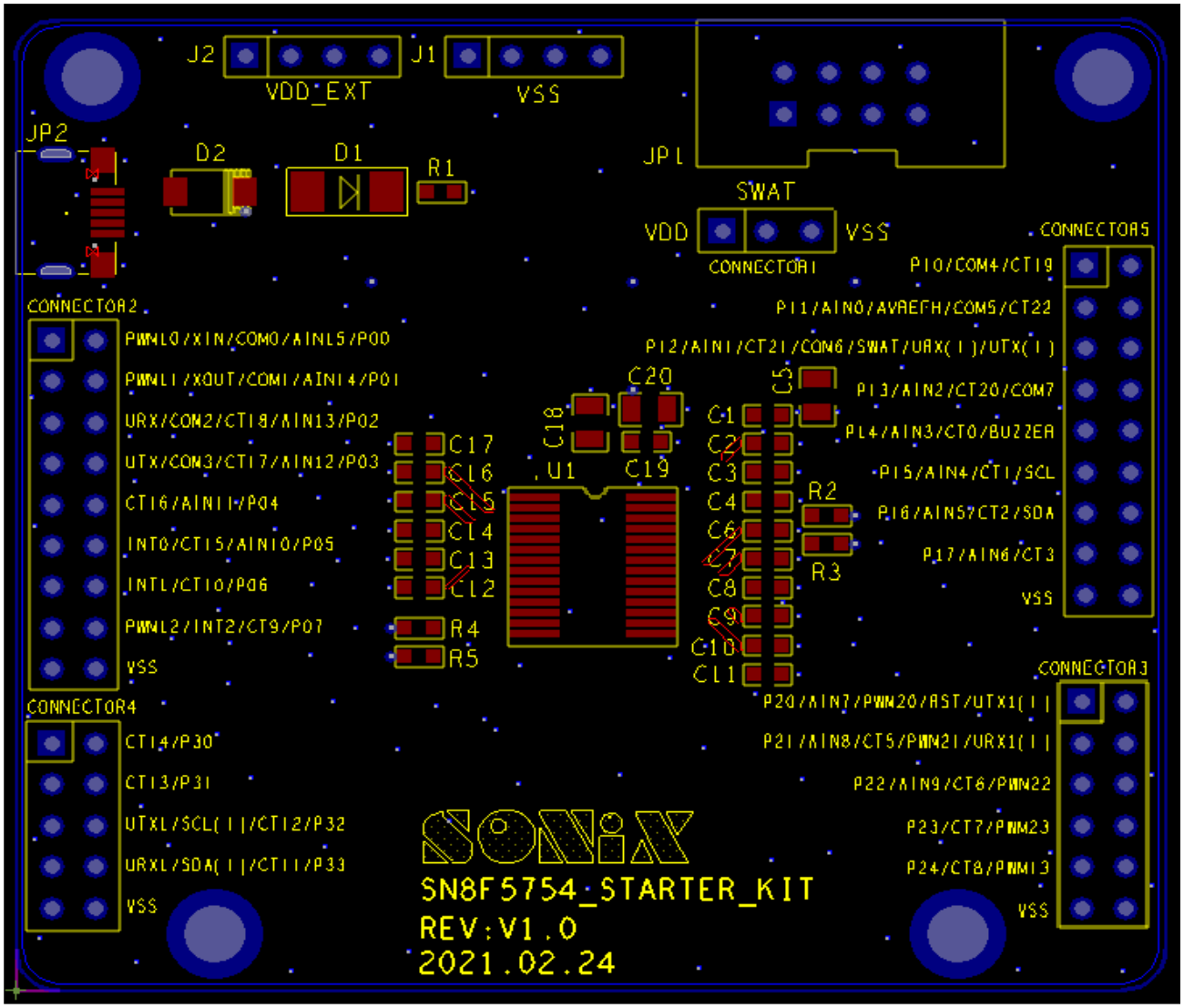

ADC
P11
C1
0.1u
P12
C2
0.1u
P13
C3
0.1u
P14
C4
0.1u
P11
AVREFH
C5
10u
P15
C6
0.1u
P16
C7
0.1u
P17
C8
0.1u
P20
C9
0.1u
P21
C10
0.1u
P22
C11
0.1u
P05
C12
0.1u
P04
C13
0.1u
P03
C14
0.1u
P02
C15
0.1u
P01
C16
0.1u
P00
C17
0.1u


PROGRAM/
DEBUG
JP1
VDD
7
5
3
1
8
6
4
2
P12
GND
DEBUG
CONNECTOR1
VDD
1
P12
2
VSS
3


I2C
P15
R2
2K
P16
R3
2K
P32
R4
2K
P33
R5
2K
VDD


I/O
CONNECTOR2
P00 PWM10/XIN/COM0/AIN15/P00
P01 PWM11/XOUT/COM1/AIN14/P01
P02 URX/COM2/CT18/AIN13/P02
P03 UTX/COM3/CT17/AIN12/P03
P04 CT16/AIN11/P04
P05 INT0/CT15/AIN10/P05
P06 INT1/CT10/P06
P07 PWM12/INT2/CT9/P07
VSS
Port 0
CONNECTOR3
P20 P20/AIN7/CT4/PWM20/RST/UTX1(1)
P21 P21/AIN8/CT5/PWM21/URX1(1)
P22 P22/AIN9/CT6/PWM22
P23 P23/CT7/PWM23
P24 P24/CT8/PWM13
VSS
Port 2
CONNECTOR5
P10 P10/COM4/CT19
P11 P11/AIN0/AVREFH/COM5/CT22
P12 P12/AIN1/CT21/COM6/SWAT/URX(1)/UTX(1)
P13 P13/AIN2/CT20/COM7
P14 P14/AIN3/CT0/BUZZER
P15 P15/AIN4/CT1/SCL
P16 P16/AIN5/CT2/SDA
P17 P17/AIN6/CT3
VSS
Port 1
CONNECTOR4
P30 CT14/P30
P31 CT13/P31
P32 UTX1/SCL(1)/CT12/P32
P33 URX1/SDA(1)/CT11/P33
VSS
Port 3
pcb 兩面要顯示連接狀態

31.3 Floor Plan of PCB layout



31.4 Component Description

Number	Description
C1 – C4, C6-C17	16-ch ADC capacitors.
C5	AVREFH capacitor.
C19-C20	MCU VDD capacitors
D1	MCU direct LED
J1,J2	External power source.
JP2	Micro USB port, VDD power source from USB 5V
JP1	SN-LINK Port
CONNECTOR1	SN-LINK Port (3-pin port)
CONNECTOR2-5	I/O connectors.

R2 – R5	I2C pull-high resisters.
U1	SN8F5754F real chip (Sonix standard option).

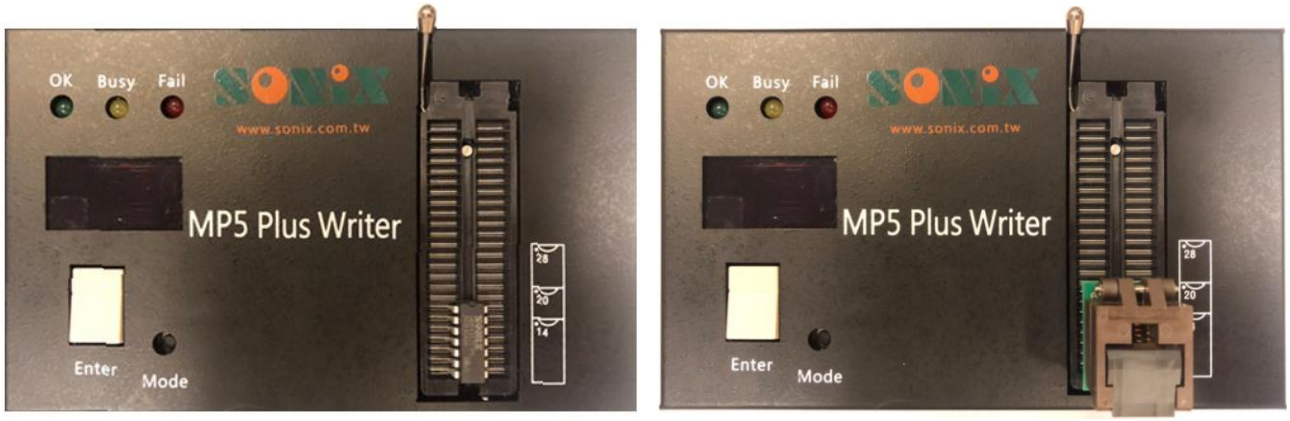
32 ROM Programming Pin

SN8F5754 Series Flash ROM erase/program/verify support SN-Link and MP5 Writer

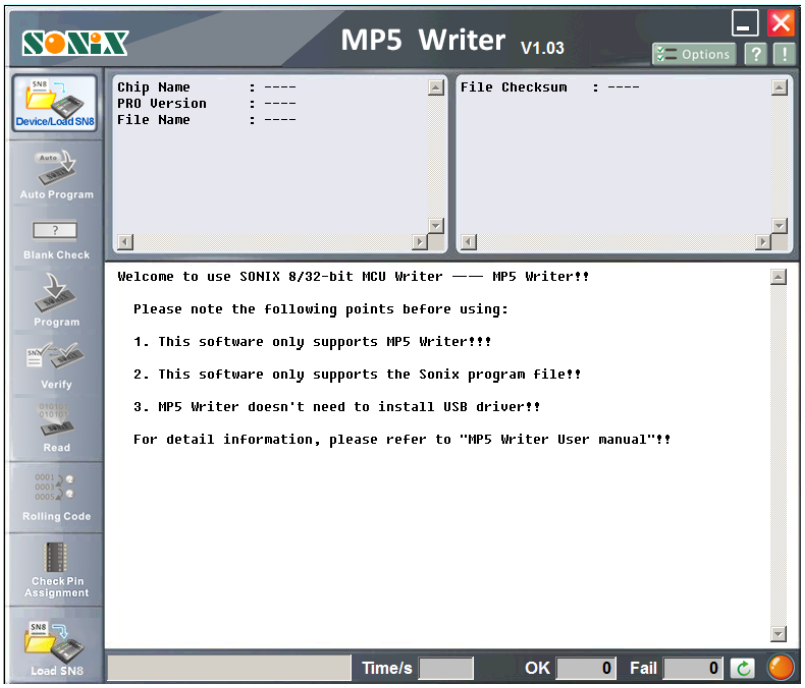
- SN-Link: Debug interface and on board programming.
- MP5 Writer: For SN8F5754 series version mass programming.

32.1 MP5 Hardware Connecting

Different package type with MCU programming connecting is as following, TSSOP and SOP illustration.

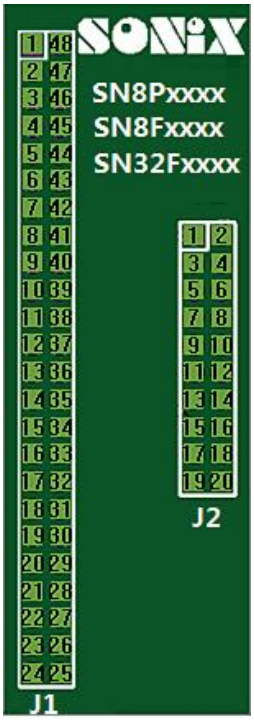


MP5 Software operation interface is as following.



32.2 MP5 Writer Transition Board Socket Pin Assignment

MP5 Writer Transition Board:



32.3 MP5 Writer Programming Pin Mapping

MP5 Writer programming pins as following.

Writer Connector		MCU Pin Number	SN8F5754S		SN8F5753S		SN8F5752S	
J2 Pin Number	J2 Pin Name		MCU Pin Number	J1 Pin Number	MCU Pin Number	J1 Pin Number	MCU Pin Number	J1 Pin Number
1	VDD	VDD	28	38	24	36	20	34
2	GND	VSS	1	11	1	13	1	15
7, 9	SWAT	P1.2	24	34	21	33	17	31
20	PDB	P0.0	2	12	2	14	2	16

32.4 SN-Link ISP Programming

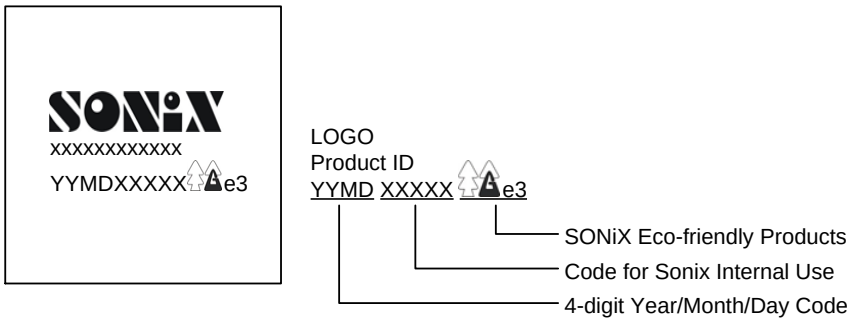
SN-Link ISP programming hardware and software are as following.

32.5 SN-Link ISP Programming Pin Mapping

SN-Link Connector		MCU Pin Number	SN8F5754S	SN8F5753S	SN8F5752S
Pin Number	Pin Name		Pin Number	Pin Number	Pin Number
7	VDD	VDD	28	24	20
2	GND	VSS	1	1	1
6,8	SWAT	P1.2	24	21	17

33 Ordering Information

The figure below is an example of the marking. Contents such as the product ID or symbol may vary according to different packages.

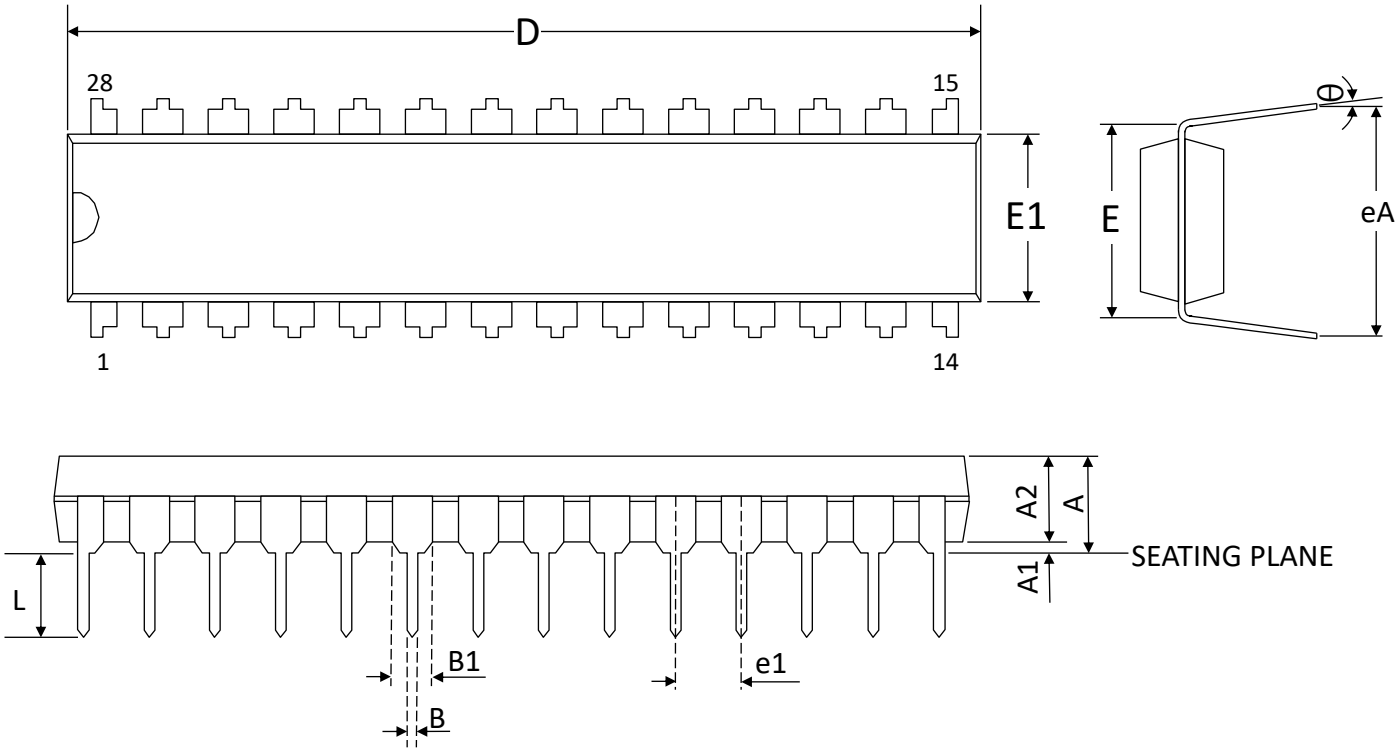


33.1 Device Nomenclature

Full Name	Packing Type
SN8F5755W	Wafer
SN8F5754H	Dice
SN8F5754KG	SKDIP, 28 pins, Green package
SN8F5754SG	SOP, 28 pins, Green package
SN8F5754TG	TSSOP, 28 pins, Green package
SN8F5753SG	SOP, 24 pins, Green package
SN8F5753TG	TSSOP, 24 pins, Green package
SN8F5752SG	SOP, 20 pins, Green package

34 Package Information.

34.1 SKDIP28

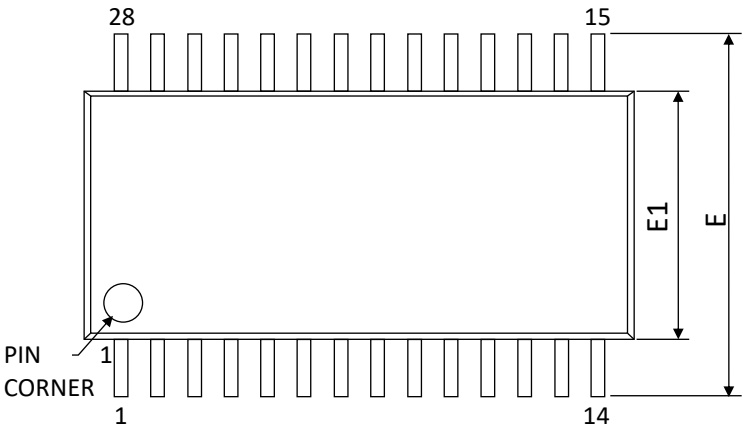


SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	4.45	--	--	0.175
A1	0.38	--	--	0.015	--	--
A2	3.18	3.30	3.43	0.125	0.130	0.135
B	0.46 typ.			0.018 typ.		
B1	1.52 typ.			0.060 typ.		
D	35.18	35.31	35.56	1.385	1.390	1.400
E	7.87 BSC			0.310 BSC		
E1	7.19	7.32	7.44	0.283	0.288	0.293
e1	2.54 typ.			0.100 typ.		
L	3.05	3.30	3.56	0.120	0.130	0.140
eA	8.38	8.89	9.40	0.330	0.350	0.370
θ	0°	7°	15°	0°	7°	15°

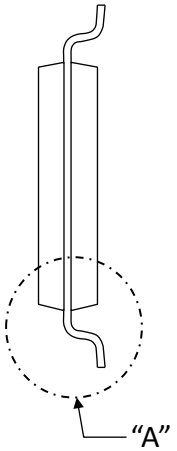
Notes：

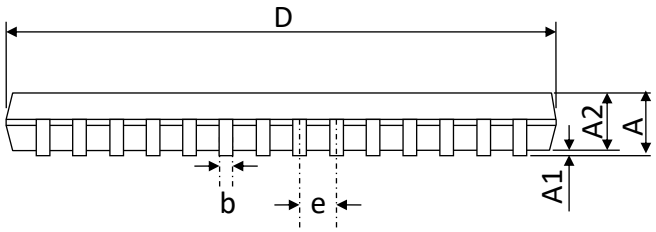
1. JEDEC OUTLINE : MS-015 AH
2. CONTROLLING DIMENSION：inch

34.2 TSSOP28

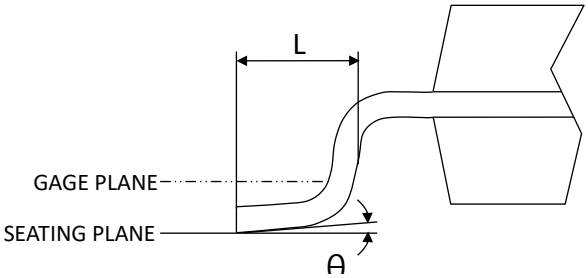


TOP VIEW





SIDE VIEW



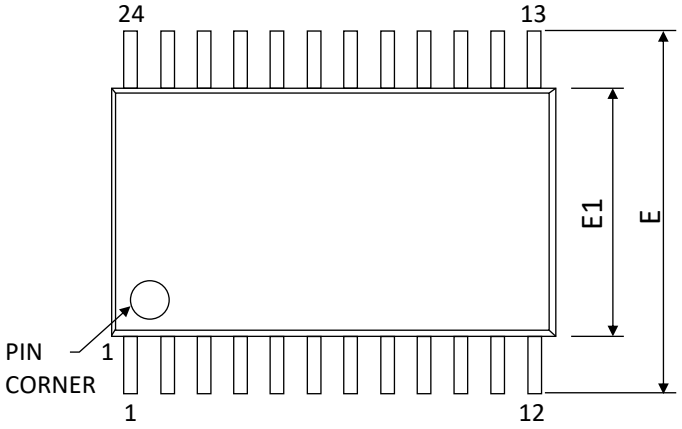
DETAIL "A"

SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	1.20	--	--	0.047
A1	0.00	--	0.15	0.000	--	0.006
A2	0.80	1.00	1.05	0.031	0.039	0.041
b	0.19	--	0.30	0.007	--	0.012
D	9.60	9.70	9.80	0.378	0.382	0.386
E	6.40 BSC.			0.252 BSC.		
E1	4.30	4.40	4.50	0.169	0.173	0.177
e	0.65 BSC.			0.026 BSC.		
L	0.45	0.60	0.75	0.018	0.024	0.030
θ	0°	--	8°	0°	--	8°

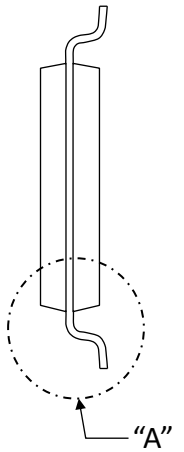
Notes：

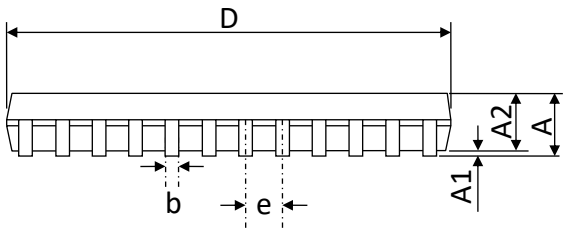
3. CONTROLLING DIMENSION：mm
4. JEDEC OUTLINE : MO-153
5. DIMENSION 'D' DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BERRES.
6. DIMENSION 'E1' DOES NOT INCLUDE INTERLEAD FLASH OR PROTRUSION.
7. DIMENSION 'b' DOES NOT INCLUDE DAMBAR PROTRUSION.

34.3 TSSOP24

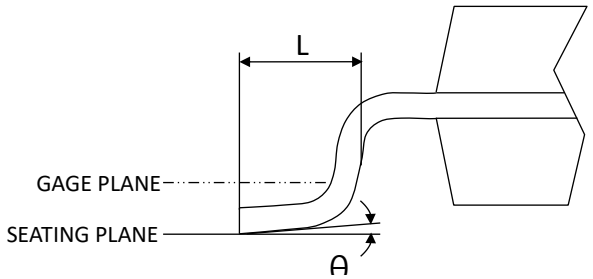


TOP VIEW





SIDE VIEW



DETAIL "A"

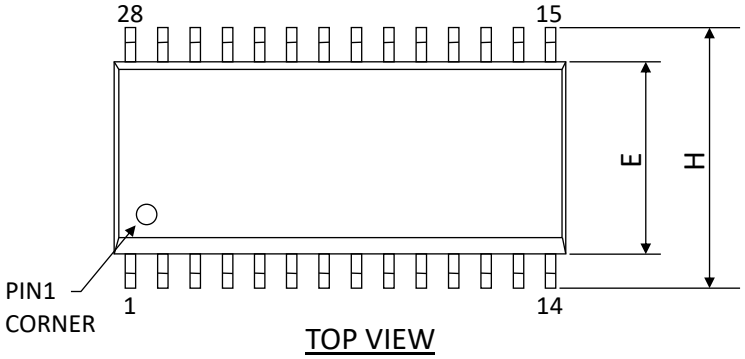
SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	1.20	--	--	0.047
A1	0.00	--	0.15	0.000	--	0.006
A2	0.80	1.00	1.05	0.031	0.039	0.041
b	0.19	--	0.30	0.007	--	0.012
D	7.70	7.80	7.90	0.303	0.307	03.11
E	6.40 BSC.			0.252 BSC.		
E1	4.30	4.40	4.50	0.169	0.173	0.177
e	0.65 BSC.			0.026 BSC.		
L	0.45	0.60	0.75	0.018	0.024	0.030
θ	0°	--	8°	0°	--	8°

Notes：

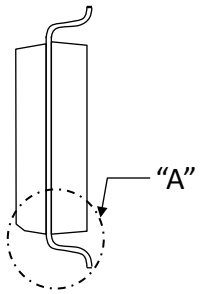
1. CONTROLLING DIMENSION：mm
2. JEDEC OUTLINE : MO-153
3. DIMENSION 'D' DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BERRES.
4. DIMENSION 'E1' DOES NOT INCLUDE INTERLEAD FLASH OR PROTRUSION.

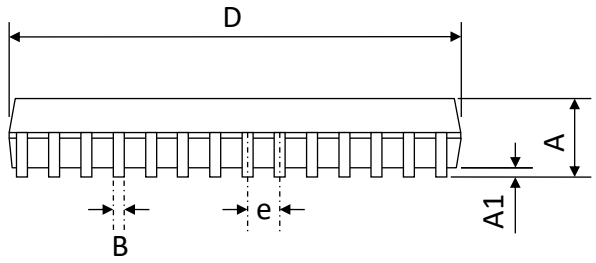
5. DIMENSION ‘b’ DOES NOT INCLUDE DAMBAR PROTRUSION.

34.4 SOP28

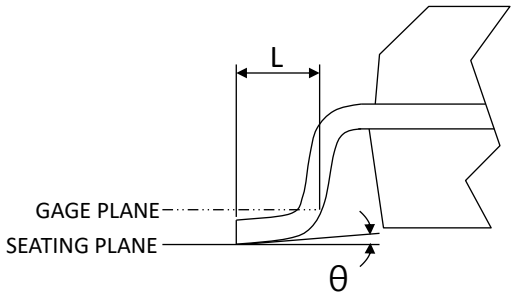


TOP VIEW





SIDE VIEW



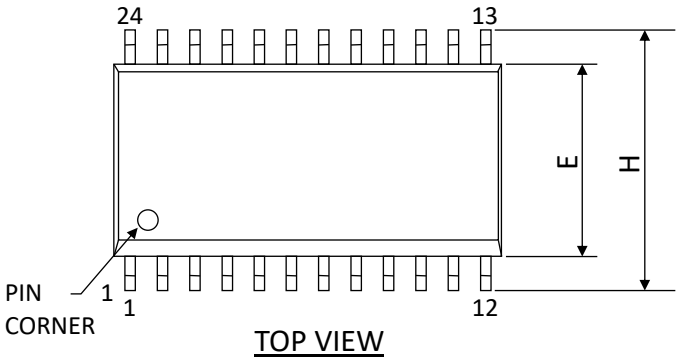
DETAIL “A”

SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	2.65	--	--	0.104
A1	0.10	--	0.30	0.004	--	0.011
B	0.31	0.41	0.51	0.012	0.016	0.020
D	17.70	18.20	18.70	0.697	0.716	0.736
E	7.50 BSC			0.295 BSC		
e	1.27 BSC			0.050 BSC		
H	10.30 BSC			405 BSC		
L	0.40	--	1.27	0.016	--	0.050
θ	0°	4 °	8°	0°	4 °	8°

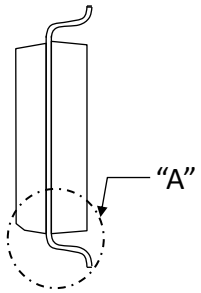
Notes：

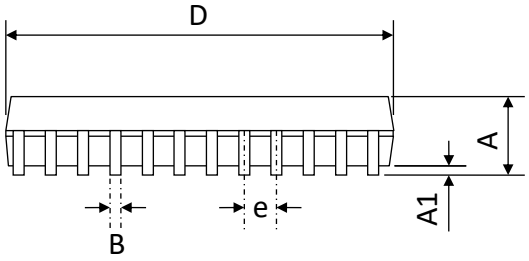
1. CONTROLLING DIMENSION：mm
2. JEDEC OUTLINE : MO-119 AB

34.5 SOP24

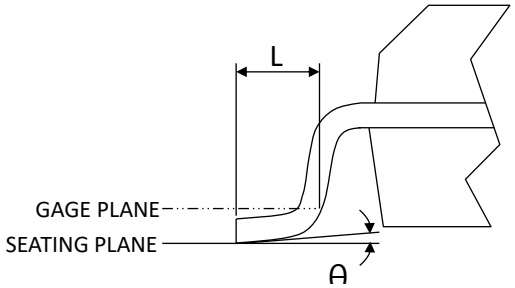


TOP VIEW





SIDE VIEW



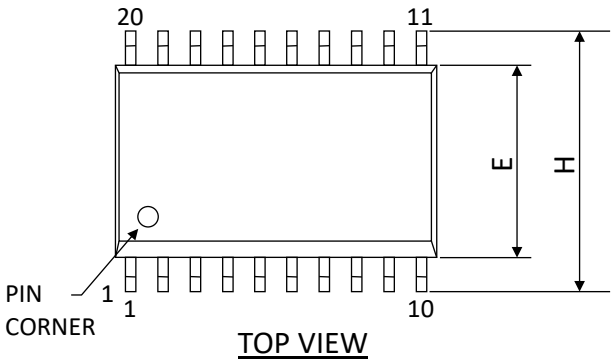
DETAIL "A"

SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	2.65	--	--	0.104
A1	0.10	--	0.30	0.004	--	0.011
B	0.31	0.41	0.51	0.012	0.016	0.020
D	15.30	15.50	15.70	0.602	0.618	0.618
E	7.50 BSC			0.295 BSC		
e	1.27 BSC			0.050 BSC		
H	10.30 BSC			0.405 BSC		
L	0.4	--	1.27	0.015	--	0.05
θ	0°	4 °	8°	0°	4 °	8°

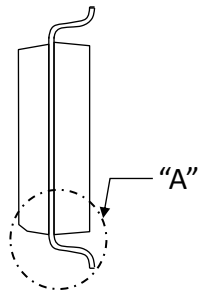
Notes：

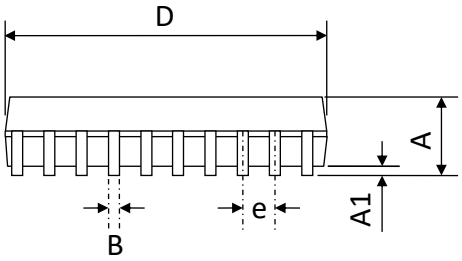
8. CONTROLLING DIMENSION：mm
9. JEDEC OUTLINE : MO-119 AA

34.6 SOP20

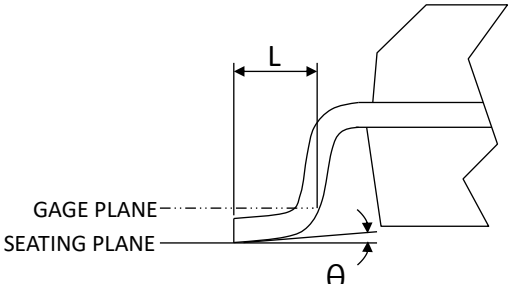


TOP VIEW





SIDE VIEW



DETAIL "A"

SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	2.65	--	--	0.104
A1	0.10	--	0.30	0.004	--	0.012
B	0.31	0.41	0.51	0.012	0.016	0.020
D	12.80 BSC			0.503		
E	7.50 BSC			0.295		
e	1.27 BSC			0.050 BSC		
H	10.30 BSC			0.405		
L	0.40	--	1.27	0.016	--	0.050
θ	0°	4°	8°	0°	4°	8°

Notes：

1. CONTROLLING DIMENSION：mm
2. JEDEC OUTLINE : MO-013 AC

35 Appendix: Reference Document

Sonix provides reference document for users to help them quickly familiar SN8F5000 family (downloadable on cooperative website: www.sonix.com.tw).

Document Name	Description
SN8F5000 Starter-Kit User Manual	This documentation introduces SN8F5000 family all Starter-Kit, providing the user selects an appropriate starter-kit for development.
SN8F5000 Family Instruction Set	The document details the 8051 instruction set, and a simple example illustrates operation.
SN8F5000 Family Instruction Mapping Table	This document supplies the information about mapping assembly instructions from 8-Bit Flash/ OTP Type to 8051 Flash Type.
SN8F5000 Packaging Information	This documentation introduces SN8F5000 family microcontrollers' mechanical data, such as height, width and pitch information.
SN8F5000 Debug Tool Manual	This document teaches the user to install software Keil C51, and helped create a new project to be developed.

SN8F5754 Series Datasheet

8051-based Microcontroller

Corporate Headquarters
10F-1, No. 36, Taiyuan St.
Chupei City, Hsinchu, Taiwan
TEL: +886-3-5600888
FAX: +886-3-5600889

Taipei Sales Office
15F-2, No. 171, Songde Rd.
Taipei City, Taiwan
TEL: +886-2-27591980
FAX: +886-2-27598180
mkt@sonix.com.tw
sales@sonix.com.tw

Hong Kong Sales Office
Unit 2603, No. 11, Wo Shing St.
Fo Tan, Hong Kong
TEL: +852-2723-8086
FAX: +852-2723-9179
hk@sonix.com.tw

Shenzhen Contact Office
High Tech Industrial Park,
Shenzhen, China
TEL: +86-755-2671-9666
FAX: +86-755-2671-9786
mkt@sonix.com.tw
sales@sonix.com.tw

USA Office
TEL: +1-714-3309877
TEL: +1-949-4686539
tlightbody@earthlink.net

Japan Office
2F, 4 Chome-8-27 Kudanminami
Chiyoda-ku, Tokyo, Japan
TEL: +81-3-6272-6070
FAX: +81-3-6272-6165
jpsales@sonix.com.tw

FAE Support via email
8-bit Microcontroller Products:
sa1fae@sonix.com.tw
All Products: fae@sonix.com.tw