SONiX Technology Co., Ltd.

SN8F5721 Series
Datasheet

8051-based Microcontroller

SN8F5721
SN8F57211
SN8F57212
SN8F57213
SN8F57214
SN8F57215
SN8F57216
SN8F57217

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)

Memory configuration

- 4 KB on-chip Flash memory (IROM) with in-system program support
- 32 bytes information block memory (IBROM)
- 256 bytes internal RAM (IRAM)
- **10 interrupt sources with priority levels control and unique interrupt vectors**
- 9 internal interrupts
- 1 external interrupts: INT0
- 1 set of DPTR
- 2 set 8/16-bit timers with 4 operation modes
- **2 set 16-bit timers with PWM generator**

  PWM generator has 6 output channels with individual duty, inverters and frequency control
- **12-bit SAR ADC** with 12 external and 1 internal channels, and 4 internal reference voltages
- **1 set UART** interface
- **1 set I2C** interface with SMBus Support
- **1 set SPI** interface
- **On-Chip Debug Support**:

  Single-wire debug interface

  2 hardware breakpoints

  Unlimited software breakpoints

  ROM data security/protection

  Debug interface ON/OFF control
- Watchdog and programmable external reset
- 1.8-V low voltage detector
- Wide supply voltage (1.8 V – 5.5 V) and temperature (-40 °C to 85 °C) range

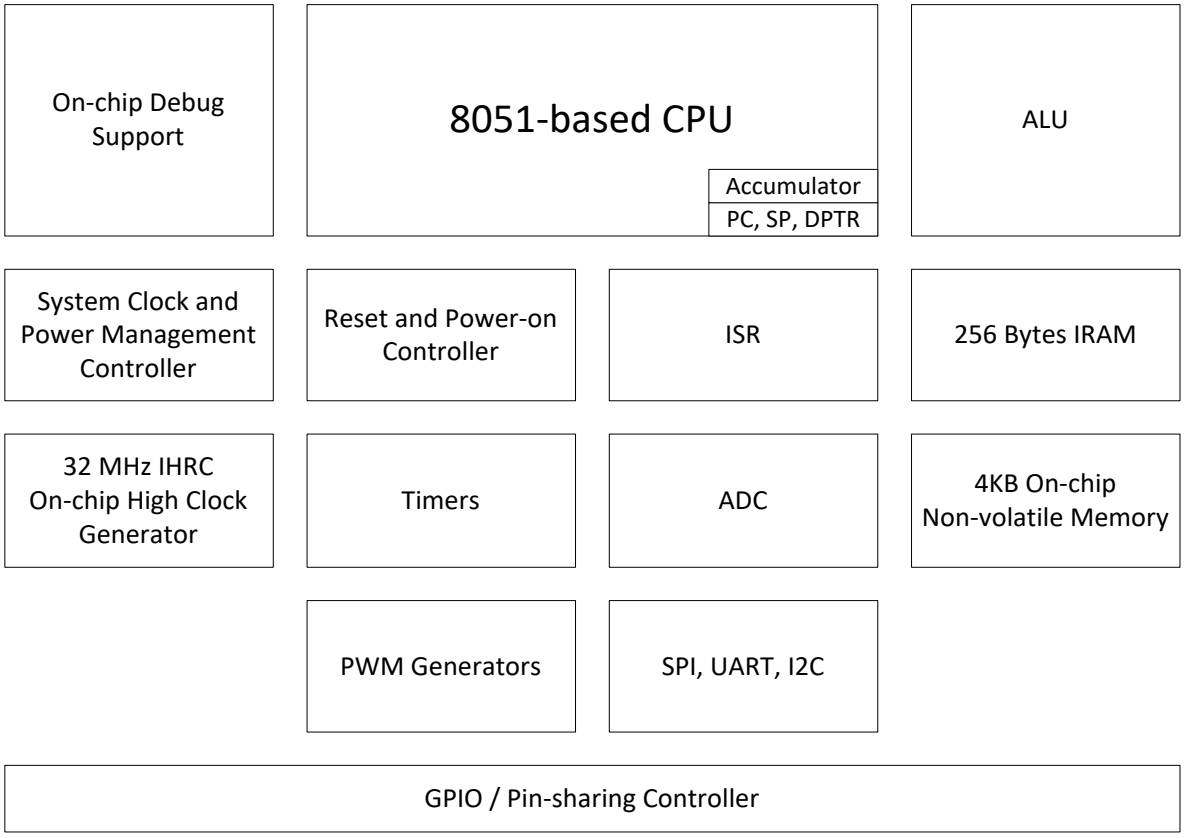
1.2 Applications

- Brushless DC motor
- Home automation
- Household
- Other

1.3 Features Selection Table

Device	I/O	PWM Channels	I2C	SPI	UART	ADC ext. Channels	Ext. INT	Package Types
SN8F5721	12	12	1	1	1	12	1	DIP14, SOP14, QFN16 3x3
SN8F57211	8	8	1	1	1	8	0	MSOP10
SN8F57212	8	8	1	1	1	8	1	MSOP10
SN8F57213	6	6	1	1	1	6	1	SOP8
SN8F57214	12	12	1	1	1	12	1	SOP14
SN8F57215	6	6	1	1	1	6	1	SOP8
SN8F57216	12	12	1	1	1	12	1	SOP14
SN8F57217	12	12	1	1	1	12	1	TSSOP14

1.4 Block Diagram



2 Table of Contents

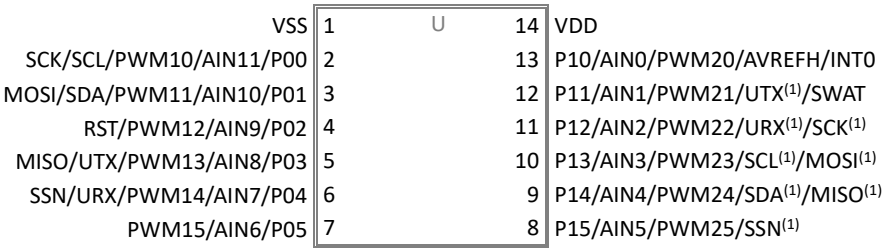
3 Revision History

Revision	Date	Description
1.0	Apr. 2020	First issue
1.1	Aug. 2020	1. Repair an error, omission, etc. 2. Add I2C electrical characteristic. 3. Modify electrical characteristic. 4. Add 5721JG package type and add 57212, 57213, 57214, 57215, 57216, 57217 part number.
1.2	Oct. 2020	1. Modify debug interface hardware section. 2. Modify electrical characteristic.
1.3	June. 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 ordering Information section. 6. Modify typing errors. 7. Remove Sample codes.
1.4	Nov. 2024	1. Chapter 30.2 SOP14 (Package Information)

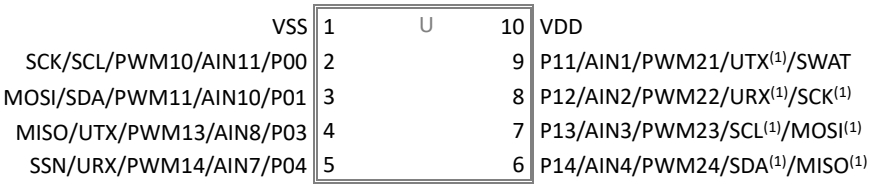
4 Pin Assignments

4.1 Pin Diagrams

- SN8F5721P/S (DIP14/SOP14)

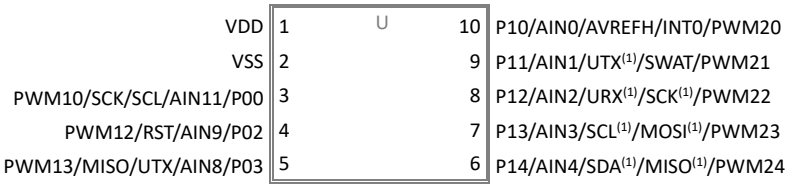


- SN8F57211A (MSOP10)



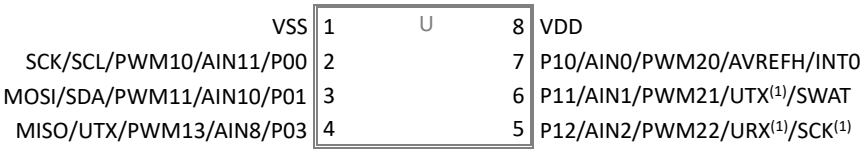
✷ ***Note: The pins which are not pin-out shall be set correctly to decrease power consumption in low-power modes. Strongly recommended to set these pins as input pull-up.***

- SN8F57212A (MSOP10)



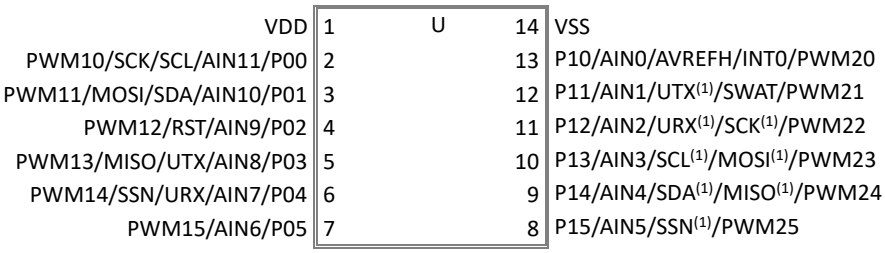
✷ ***Note: The pins which are not pin-out shall be set correctly to decrease power consumption in low-power modes. Strongly recommended to set these pins as input pull-up.***

- SN8F57213S (SOP8)



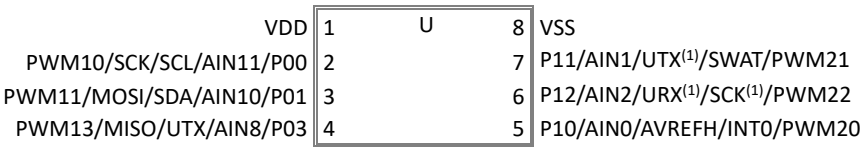
> ✸ ***Note: The pins which are not pin-out shall be set correctly to decrease power consumption in low-power modes. Strongly recommended to set these pins as input pull-up.***

- SN8F57214S (SOP14)



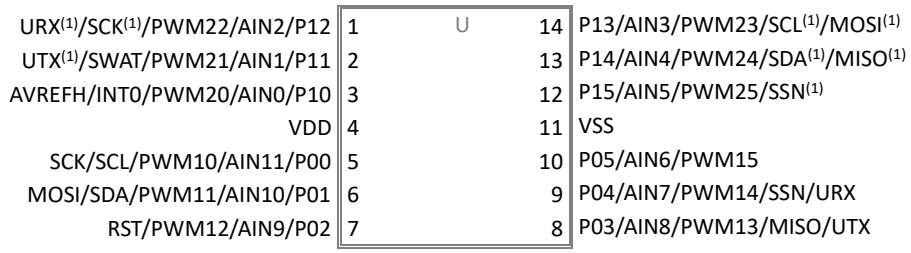
> ✸ ***Note: The pins which are not pin-out shall be set correctly to decrease power consumption in low-power modes. Strongly recommended to set these pins as input pull-up.***

- SN8F57215S (SOP8)



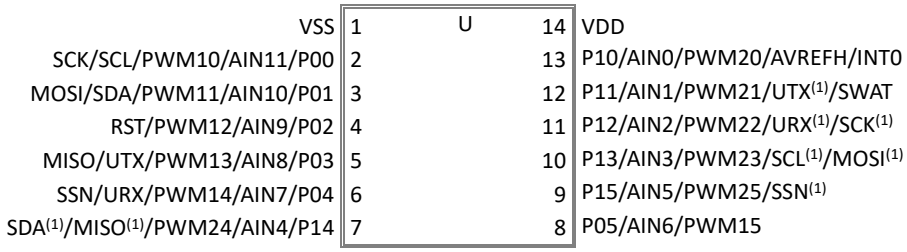
> ✸ ***Note: The pins which are not pin-out shall be set correctly to decrease power consumption in low-power modes. Strongly recommended to set these pins as input pull-up.***

- SN8F57216S (SOP14)



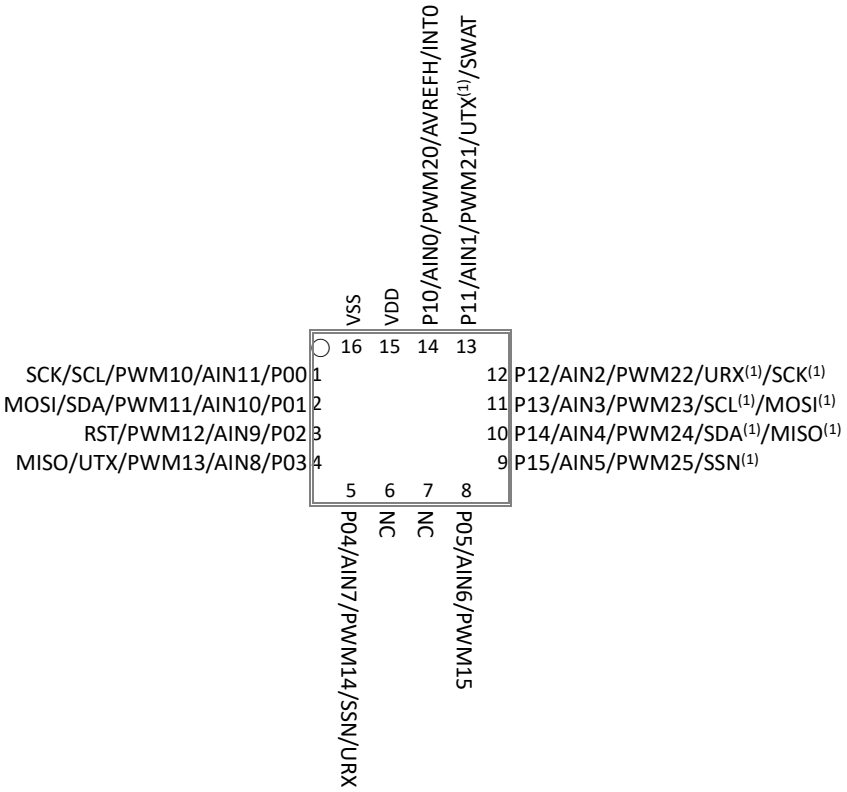
> ✷ ***Note: The pins which are not pin-out shall be set correctly to decrease power consumption in low-power modes. Strongly recommended to set these pins as input pull-up.***

- SN8F57217T (TSSOP14)



> ✷ ***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.***

- SN8F5721J (QFN16 3x3)



Note :
1: I2C output is alternately SCL/SDA or /SCL(1) /SDA(1).
2: UART output is alternately UTX/URX or /UTX(1) /URX(1)
3: SPI output is alternately SCK/MOSI/MISO/SSN or /SCK(1)/MOSI(1)/MISO(1)/SSN(1)

> ✸ ***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.***

4.2 Pin Allocation Table

- SN8F5721P/S (DIP14/SOP14), 57214S/57216S(SOP14), 57217T(TSSOP14), 5721J(QFN16 3x3)

Port	Open-Drain	Wakeup	External Interrupt	ADC	ADC External Reference	UART	I2C	SPI	PWM	External Reset	Debug interface
P0.0	V	V	-	AIN11	-	-	SCL	SCK	PWM10	-	-
P0.1	V	V	-	AIN10	-	-	SDA	MOSI	PWM11	-	-
P0.2	-	V	-	AIN9	-	-	-	-	PWM12	RST	-
P0.3	V	V	-	AIN8	-	UTX	-	MISO	PWM13	-	-
P0.4	V	V	-	AIN7	-	URX	-	SSN	PWM14	-	-
P0.5	-	V	-	AIN6	-	-	-	-	PWM15	-	-
P1.0	-	V	INT0	AIN0	AVREFH	-	-	-	PWM20	-	-
P1.1	V	V	-	AIN1	-	UTX(1)	-	-	PWM21	-	SWAT
P1.2	V	V	-	AIN2	-	URX(1)	-	SCK(1)	PWM22	-	-
P1.3	V	V	-	AIN3	-	-	SCL(1)	MOSI(1)	PWM23	-	-
P1.4	V	V	-	AIN4	-	-	SDA(1)	MISO(1)	PWM24	-	-
P1.5	-	V	-	AIN5	-	-	-	SSN(1)	PWM25	-	-

- SN8F57211A/57212A (MSOP10)

Port	Open-Drain	Wakeup	External Interrupt	ADC	ADC External Reference	UART	I2C	SPI	PWM	External Reset	Debug interface
P0.0	V	V	-	AIN11	-	-	SCL	SCK	PWM10	-	-
P0.1	V	V	-	AIN10	-	-	SDA	MOSI	PWM11	-	-
P0.3	V	V	-	AIN8	-	UTX	-	MISO	PWM13	-	-
P0.4	V	V	-	AIN7	-	URX	-	SSN	PWM14	-	-
P1.1	V	V	-	AIN1	-	UTX(1)	-	-	PWM21	-	SWAT
P1.2	V	V	-	AIN2	-	URX(1)	-	SCK(1)	PWM22	-	-
P1.3	V	V	-	AIN3	-	-	SCL(1)	MOSI(1)	PWM23	-	-
P1.4	V	V	-	AIN4	-	-	SDA(1)	MISO(1)	PWM24	-	-

- SN8F57213S/57215S (SOP8)

Port	Open-Drain	Wakeup	External Interrupt	ADC	ADC External Reference	UART	I2C	SPI	PWM	External Reset	Debug interface
P0.0	V	V	-	AIN11	-	-	SCL	SCK	PWM10	-	-
P0.1	V	V	-	AIN10	-	-	SDA	MOSI	PWM11	-	-
P0.3	V	V	-	AIN8	-	UTX	-	MISO	PWM13	-	-
P1.0	-	V	INT0	AIN0	AVREFH	-	-	-	PWM20	-	-
P1.1	V	V	-	AIN1	-	UTX(1)	-	-	PWM21	-	SWAT
P1.2	V	V	-	AIN2	-	URX(1)	-	SCK(1)	PWM22	-	-

4.3 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.
AIN11	Analog Input	ADC: input channel 11.
SCL	Digital I/O	I2C: clock output (master) or clock input (slave).
SCK	Digital I/O	SPI: clock output (master) or clock input (slave).
PWM10	Digital Output	PWM1: programmable PWM output.
P0.1	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
AIN10	Analog Input	ADC: input channel 10.
SDA	Digital I/O	I2C: data pin.
MOSI	Digital I/O	SPI: transmission pin (master) or reception pin (slave).
PWM11	Digital Output	PWM1: programmable PWM output.
P0.2	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
AIN9	Analog Input	ADC: input channel 9.
Reset	Digital Input	System reset (active low).
PWM12	Digital Output	PWM1: programmable PWM output.
P0.3	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
AIN8	Analog Input	ADC: input channel 8.
UTX	Digital Output	UART: transmission pin.
MISO	Digital I/O	SPI: reception pin (master) or transmission pin (slave).
PWM13	Digital Output	PWM1: programmable PWM output.
P0.4	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
AIN7	Analog Input	ADC: input channel 7.
URX	Digital Input	UART: reception pin.
SSN	Digital Input	SPI: salve selection pin (slave mode).
PWM14	Digital Output	PWM1: programmable PWM output.

P0.5	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.
PWM15	Digital Output	PWM1: programmable PWM output.

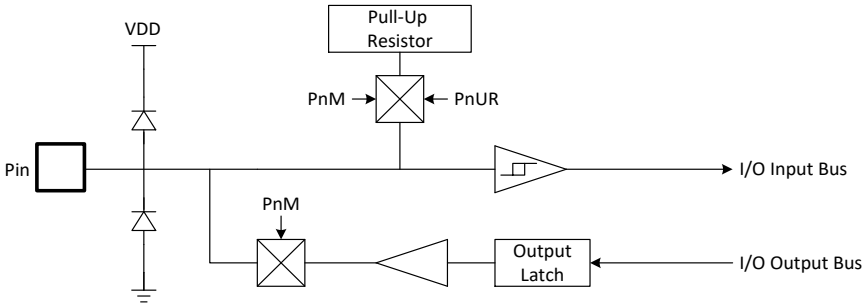
Port 1

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.
AIN0	Analog Input	ADC: input channel 0.
AVREFH	Analog Input	ADC: external reference voltage.
INT0	Digital Input	INT0: external interrupt 0.
PWM20	Digital Output	PWM2: programmable PWM output.
P1.1	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
AIN1	Analog Input	ADC: input channel 1.
UTX(1)	Digital Output	UART: transmission pin.
SWAT	Digital I/O	Debug interface.
PWM21	Digital Output	PWM2: programmable PWM output.
P1.2	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
AIN2	Analog Input	ADC: input channel 2.
URX(1)	Digital Input	UART: reception pin.
SCK(1)	Digital I/O	SPI: clock output (master) or clock input (slave).
PWM22	Digital Output	PWM2: programmable PWM output.
P1.3	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
AIN3	Analog Input	ADC: input channel 3.
SCL(1)	Digital I/O	I2C: clock output (master) or clock input (slave).
MOSI(1)	Digital I/O	SPI: transmission pin (master) or reception pin (slave).
PWM23	Digital Output	PWM2: programmable PWM output.
P1.4	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
AIN4	Analog Input	ADC: input channel 4.
SDA(1)	Digital I/O	I2C: data pin.
MISO(1)	Digital I/O	SPI: reception pin (master) or transmission pin (slave).
PWM24	Digital Output	PWM2: programmable PWM output.
P1.5	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode.

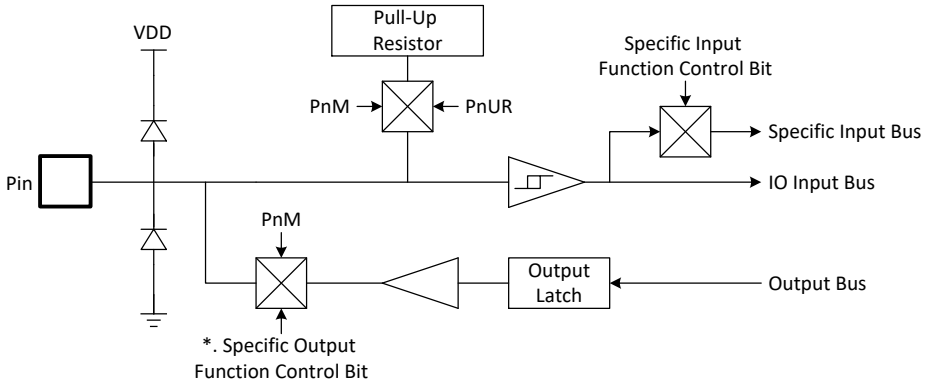
		Built-in pull-up resisters. Level change wake-up.
AIN5	Analog Input	ADC: input channel 5.
SSN(1)	Digital Input	SPI: salve selection pin (slave mode).
PWM25	Digital Output	PWM2: programmable PWM output.

4.4 Pin Circuit Diagrams

Normal Bi-direction I/O Pin.

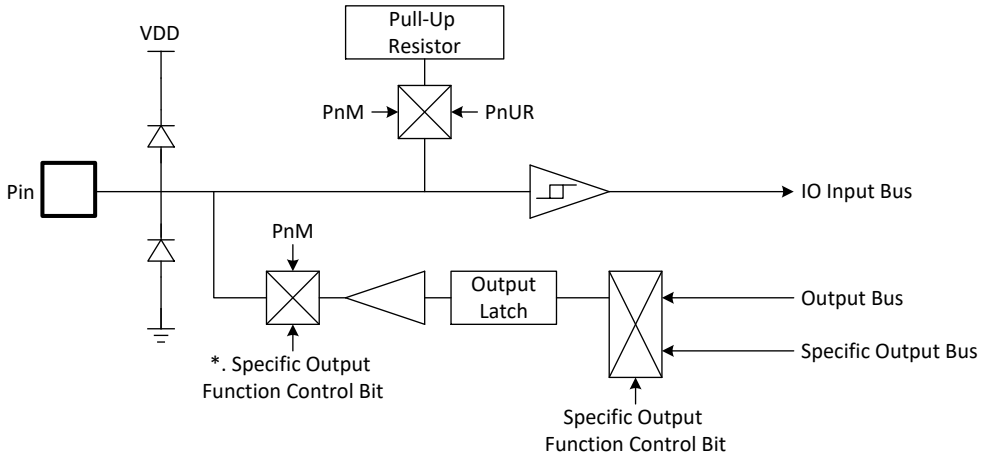


Bi-direction I/O Pin Shared with Specific Digital Input Function, e.g. INT0, UART.



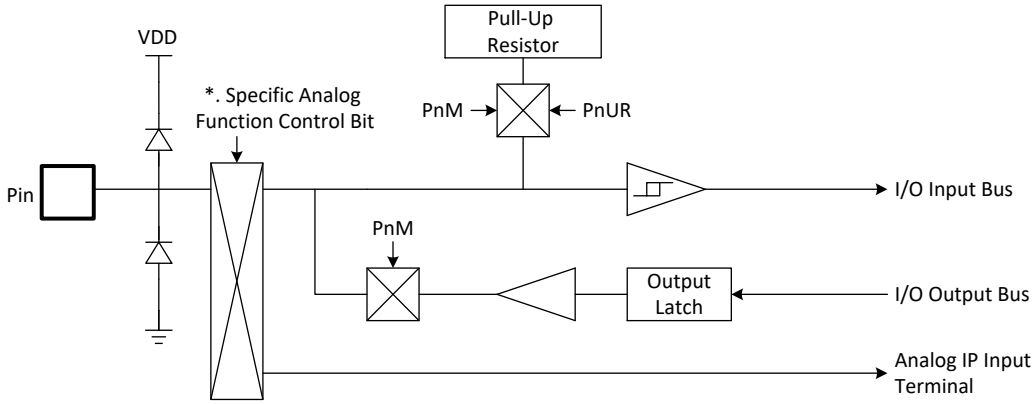
***. Some specific functions switch I/O direction directly, not through PnM register.**

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. ADC.



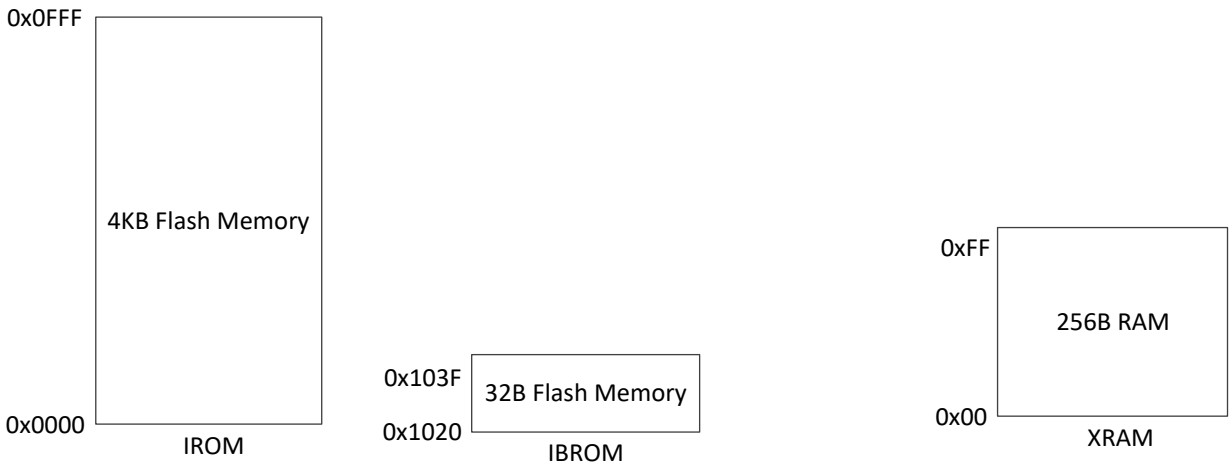
***. Some specific functions switch I/O direction directly, not through PnM register.**

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). Generally speaking, SN8F5000 CPU has 9.4 to 12.1 times faster than the original 8051 at the same frequency.

5.1 Memory Organization

SN8F5721 builds in three on-chip memories: internal RAM (IRAM), 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). The program memory is a 4 KB non-volatile memory and has a maximum 8 MHz speed limitation. The information block memory is a 32-byte non-volatile memory and supports user to read device tracking data or series number.



Information block access function supports user can access 64 bits Unique ID. Content description is specific information including lot ID, wafer ID and XY-coordinate ID. But this area is read-only and can’t be erased or programmed by user.

Information block memory		
Address	Description	Note
0x1021	Lot_ID	UID[7:0]
0x1022	Lot_ID	UID[15:8]
0x1023	Lot_ID	UID[23:16]
0x1024	Lot_ID	UID[31:24]
0x1025	Lot_ID	UID[39:32]
0x1026	Wafer_ID	UID[47:40]
0x1027	X-coordinate_ID	UID[55:48]
0x1028	Y-coordinate_ID	UID[63:56]

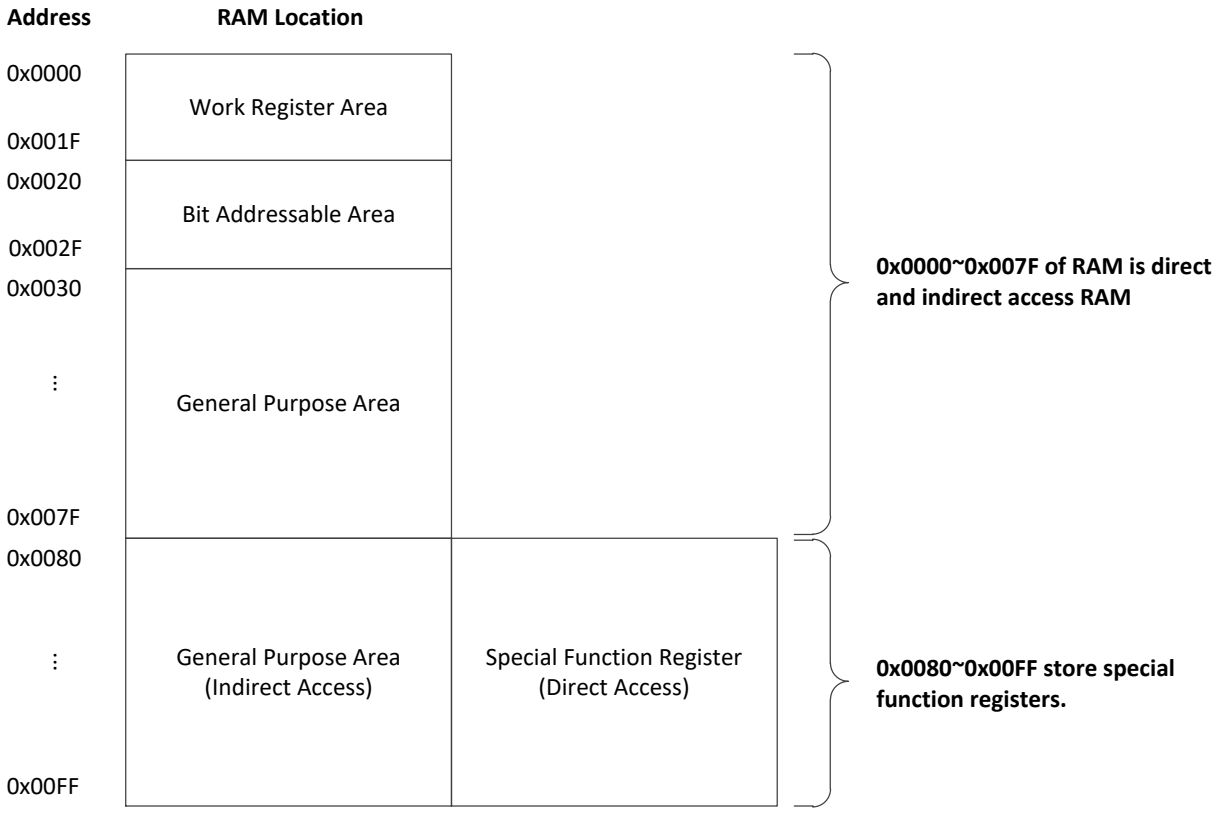
✷ ***Note:***

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

2. IBROM address 0x1020 and 0x1029~0x103F are reserved area.

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”,

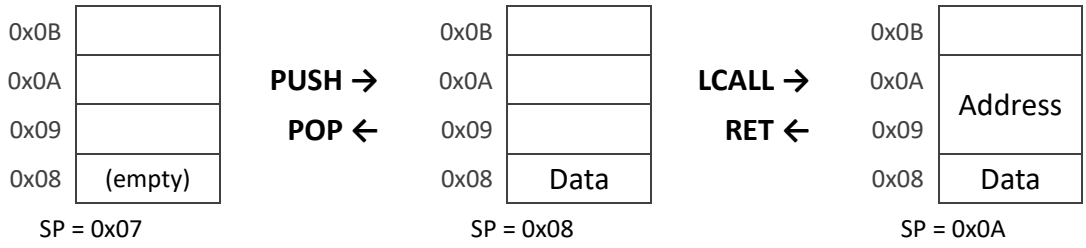
it means the bit2 of the byte 28H is set.

Bit Addressable Area	Byte Address	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
	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 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.

✷ ***Note: Stack and IRAM share the same area, Keil C51 compiler will not display "error" or "warning" when overlap condition is occurred so user must pay attention.***

5.4 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**	
001BH	**TIMER1 Interrupt vector**	
0043H	**I2C Interrupt vector**	
004BH	**SPI Interrupt vector**	
0053H	**UART RX Interrupt vector**	
005BH	**UART TX Interrupt vector**	
0083H	**PWM1 Interrupt vector**	
008BH	**PWM2 Interrupt vector**	
00ABH	**ADC Interrupt vector**	
00ACH	General purpose area	User program End of user program
0FF6H 0FF7H . 0FFDH 0FFEH 0FFFH	Reserved	

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.
- 0x0003~0x00AB: Multi interrupt vector area. Each of interrupt events has a unique interrupt vector.
- 0x00AC~0x0FDF: General purpose area for user program and ISP (EEPROM function).
- 0x0FE0~0x0FF6: General purpose area for user program. Do not execute ISP.

- 0x0FF6~0x0FFF: Reserved area. Do not execute ISP.

5.5 Program Memory Security

The SN8F5721 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.

5.6 Data Pointer

A data pointer helps to specify the IROM address while performing MOVC instructions. The microcontroller has one set of data pointer (DPH/DPL).

5.7 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

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

6 Special Function Registers

6.1 Special Function Register Memory Map

BIN / HEX	000	001	010	011	100	101	110	111
F8	CLKCAL	P0M	P1M	-	PW25DL	PW25DH	-	PFLAG
F0	B	P0UR	P1UR	LFRQL	LFRQH	PW24DL	PW24DH	SRST
E8	-	-	-	-	-	-	-	-
E0	ACC	SPSTA	SPCON	SPDAT	PW23DL	PW23DH	-	TCON0
D8	-	-	I2CDAT	I2CADR	I2CCON	I2CSTA	SMBSEL	SMBDST
D0	PSW	-	ADM	ADB	ADR	VREFH	P0CON	-
C8	PW2M	PW2CH	PW2NV	PW2O	-	FRQL	FRQH	DEGCMD
C0	-	PW21DL	PW21DH	PW22DL	PW22DH	FRQCMD	SYSMOD	-
B8	IEN1	IP1	PW20DL	PW20DH	-	P0OC	-	IRCON2
B0	PW1M	PW1CH	PW1NV	PW1O	PW15DL	PW15DH	PW2YL	PW2YH
A8	IEN0	IP0	PW12DL	PW12DH	PW13DL	PW13DH	PW14DL	PW14DH
A0	-	PW1YL	PW1YH	PW10DL	PW10DH	PW11DL	PW11DH	-
98	S0CON	S0BUF	IEN2	S0CON2	S0RELL	S0RELH	-	P1CON
90	P1	P1W	-	-	PECMD	PEROML	PEROMH	PERAM
88	TCON	TMOD	TL0	TL1	TH0	TH1	-	PEDGE
80	P0	SP	DPL	DPH	-	-	WDTR	PCON

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

6.2 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.
-	0x84	-
-	0x85	-
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.
-	0x8E	-
PEDGE	0x8F	External interrupt edge controls register.
P1	0x90	Port 1 data buffer.
P1W	0x91	Port 1 wake-up controls register.
-	0x92	-
-	0x93	-
PECMD	0x94	In-System Program command register.
PEROML	0x95	In-System Program ROM address low byte
PEROMH	0x96	In-System Program ROM address high byte
PERAM	0x97	In-System Program RAM mapping address
S0CON	0x98	UART control register.
S0BUF	0x99	UART data buffer.
IEN2	0x9A	Interrupts enable register.
S0CON2	0x9B	UART baud rate controls register.
S0RELL	0x9C	UART reload low byte register.
S0RELH	0x9D	UART reload high byte register.
-	0x9E	-
P1CON	0x9F	Port 1 configuration controls register.

0xA0 - 0xBF Registers Description

Register	Address	Description
-	0xA0	-
PW1YL	0xA1	PWM1 cycle controls buffer low byte.
PW1YH	0xA2	PWM1 cycle controls buffer high byte.
PW10DL	0xA3	PWM10 duty controls buffer low byte.
PW10DH	0xA4	PWM10 duty controls buffer high byte.
PW11DL	0xA5	PWM11 duty controls buffer low byte.
PW11DH	0xA6	PWM11 duty controls buffer high byte.
-	0xA7	-
IEN0	0xA8	Interrupts enable register
IP0	0xA9	Interrupts priority register.
PW12DL	0xAA	PWM12 duty controls buffer low byte.
PW12DH	0xAB	PWM12 duty controls buffer high byte.
PW13DL	0xAC	PWM13 duty controls buffer low byte.
PW13DH	0xAD	PWM13 duty controls buffer high byte.
PW14DL	0xAE	PWM14 duty controls buffer low byte.
PW14DH	0xAF	PWM14 duty controls buffer high byte.
PW1M	0xB0	PWM1 controls register.
PW1CH	0xB1	PWM1 channel control register.
PW1NV	0xB2	PWM1 inverse control register.
PW1O	0xB3	PWM1 frequency control register.
PW15DL	0xB4	PWM15 duty controls buffer low byte.
PW15DH	0xB5	PWM15 duty controls buffer high byte.
PW2YL	0xB6	PWM2 cycle controls buffer low byte.
PW2YH	0xB7	PWM2 cycle controls buffer high byte.
IEN1	0xB8	Interrupts enable register.
IP1	0xB9	Interrupts priority register.
PW20DL	0xBA	PWM20 duty controls buffer low byte.
PW20DH	0xBB	PWM20 duty controls buffer high byte.
-	0xBC	-
P0OC	0xBD	Open drain controls register.
-	0xBE	-
IRCON2	0xBF	Interrupts request register.

0xC0 - 0xDF Registers Description

Register	Address	Description
-	0xC0	-
PW21DL	0xC1	PWM21 duty controls buffer low byte.
PW21DH	0xC2	PWM21 duty controls buffer high byte.
PW22DL	0xC3	PWM22 duty controls buffer low byte.
PW22DH	0xC4	PWM22 duty controls buffer high byte.
FRQCMD	0xC5	Clock fine tuning command register.
SYSMOD	0xC6	System controls register.
-	0xC7	-
PW2M	0xC8	PWM2 controls register.
PW2CH	0xC9	PWM2 channel control register.
PW2NV	0xCA	PWM2 inverse control register.
PW2O	0xCB	PWM2 frequency control register.
-	0xCC	-
FRQL	0xCD	Clock fine tuning controls buffer low byte
FRQH	0xCE	Clock fine tuning controls buffer high byte
DEGCMD	0xCF	Debug mode switch control register.
PSW	0xD0	System flag register.
-	0xD1	-
ADM	0xD2	ADC controls register.
ADB	0xD3	ADC data buffer.
ADR	0xD4	ADC resolution selects register.
VREFH	0xD5	ADC reference voltage controls register.
P0CON	0xD6	Port 0 configuration controls register.
-	0xD7	-
-	0xD8	-
-	0xD9	-
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.
SPSTA	0xE1	SPI statuses register.
SPCON	0xE2	SPI control register.
SPDAT	0xE3	SPI data buffer.
PW23DL	0xE4	PWM23 duty controls buffer low byte.
PW23DH	0xE5	PWM23 duty controls buffer high byte.
-	0xE6	-
TCON0	0xE7	Timer 0 / 1 clock controls register.
-	0xE8	-
-	0xE9	-
-	0xEA	-
-	0xEB	-
-	0xEC	-
-	0xED	-
-	0xEE	-
-	0xEF	-
B	0xF0	Multiplication/ division instruction data buffer.
P0UR	0xF1	Port 0 pull-up resister controls register.
P1UR	0xF2	Port 1 pull-up resister controls register.
LFRQL	0xF3	Low clock fine tuning controls buffer low byte.
LFRQH	0xF4	Low clock fine tuning controls buffer high byte.
PW24DL	0xF5	PWM24 duty controls buffer low byte.
PW24DH	0xF6	PWM24 duty controls buffer high byte.
SRST	0xF7	Software reset controls register.
CLKCAL	0xF8	ILRC auto-calibration register.
P0M	0xF9	Port 0 input/output mode register.
P1M	0xFA	Port 1 input/output mode register.
-	0xFB	-
PW25DL	0xFC	PWM25 duty controls buffer low byte.
PW25DH	0xFD	PWM25 duty controls buffer high byte.
-	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 without borrowing signal, rotation with shifting out logic “0” 1: Addition with carry, subtraction with borrowing, rotation with shifting out logic “1”
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 Register Declaration

SN8F5721 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.
#include <SN8F5721.H>
```

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

```
#include <SN8F5721.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
SN8F5721	SN8F5721.h	OPTIONS_SN8F5721.A51
SN8F57211	SN8F57211.h	
SN8F57213	SN8F57213.h	

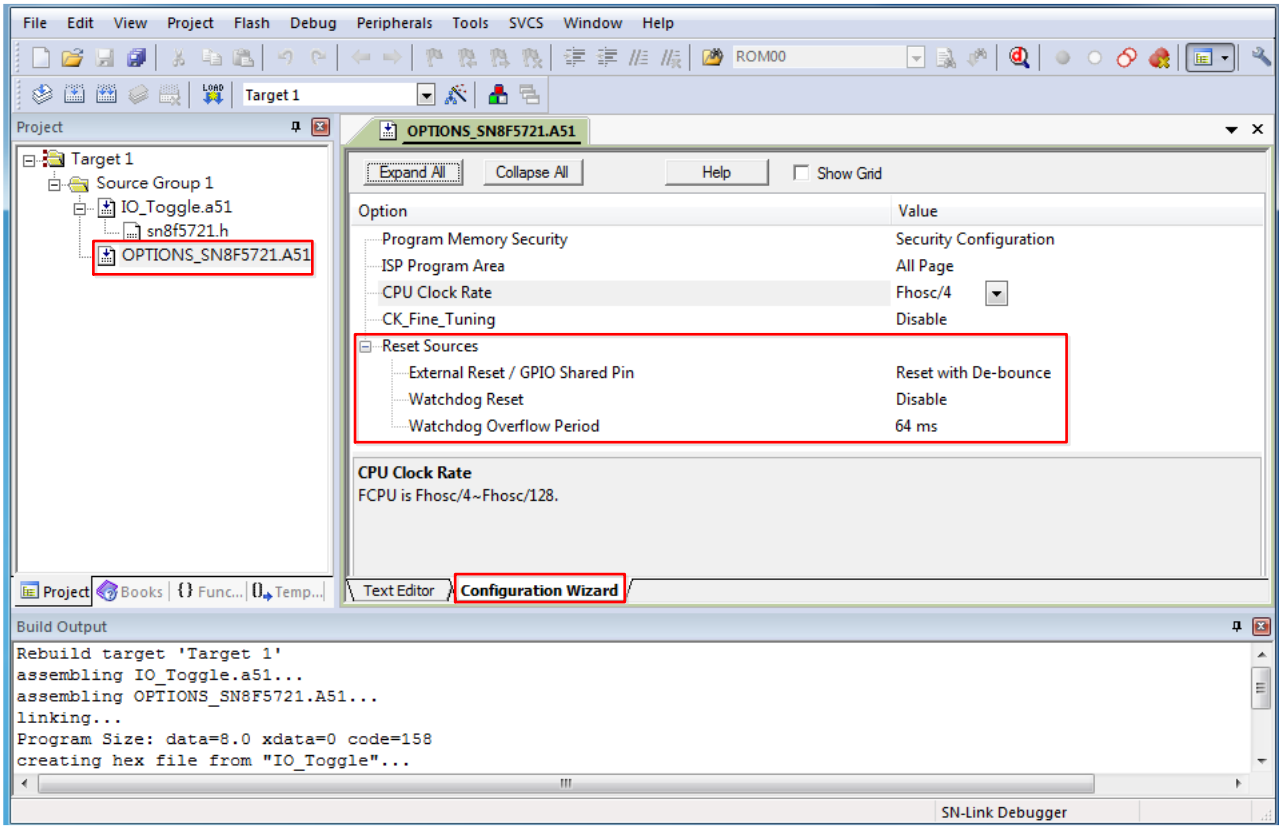
7 Reset and Power-on Controller

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 an *OPTIONS_SN8F5721.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 Rate
- CK_Fine_Tuning
- Reset Source : External Reset / GPIO Shared Pin
- Reset Source : Watchdog Reset & Overflow Period

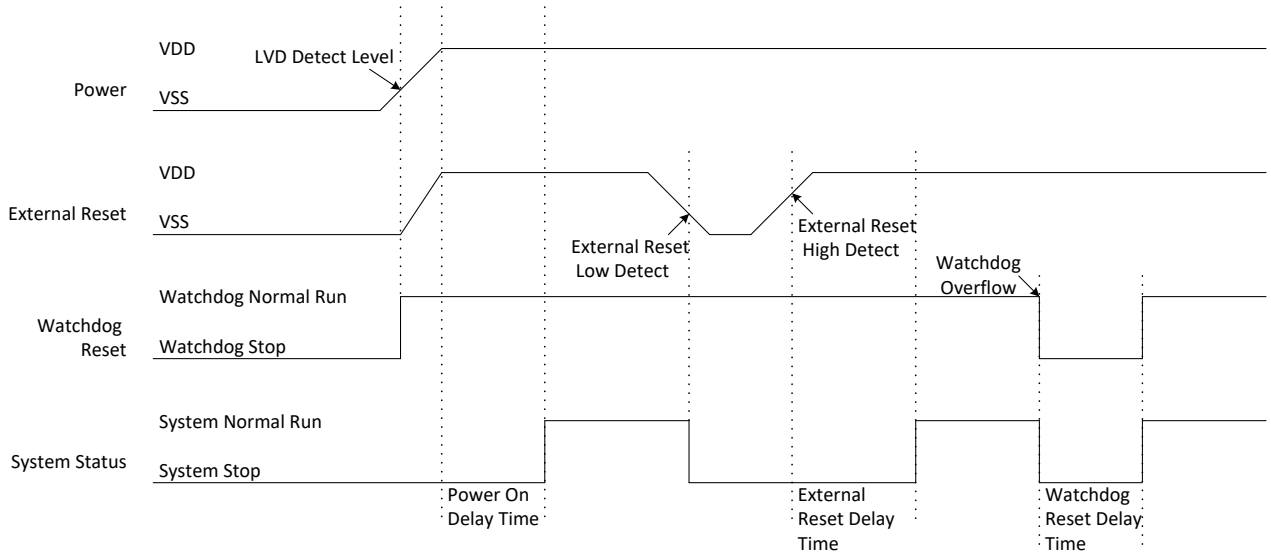


The code option is the system hardware configurations including noise filter option, watchdog timer operation, reset pin option and flash ROM security control. The code option items are as following table:

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 0x0F00 ~ 0x0FDF, only address 0x0F00 ~ 0x0FDF ROM data can be accessed
ISP Program Area	All Page	All address can perform ISP function
	Page 120~ Page 126	Only address 0x0F00 ~ 0x0FDF can perform ISP function
CPU Clock Rate	000 = fosc / 128 001 = fosc / 64 010 = fosc / 32 011 = fosc / 16 100 = fosc / 8 101 = fosc / 4	Fcpu clock rate
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 P02	Enable P02
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.

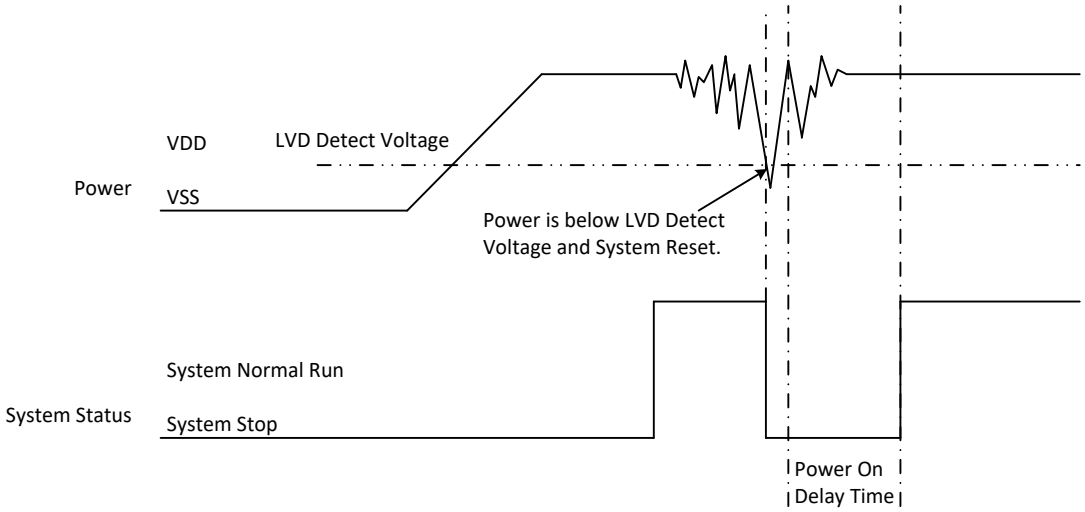


The power stabilization period spends 4.6 ms in typical condition. Afterward the microcontroller fetches CPU Clock Source selection automatically. The selected clock source would be driven, and the system counts 2048 times of the clock period and 4 times of the internal low-speed oscillator clocks to ensure its reliability.

✷ ***Note: In high power noise environment, user can put 10ohm resistor in the front of 0.1uF capacitor & VDD PAD to suppress power noise and avoid IC damage.***

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

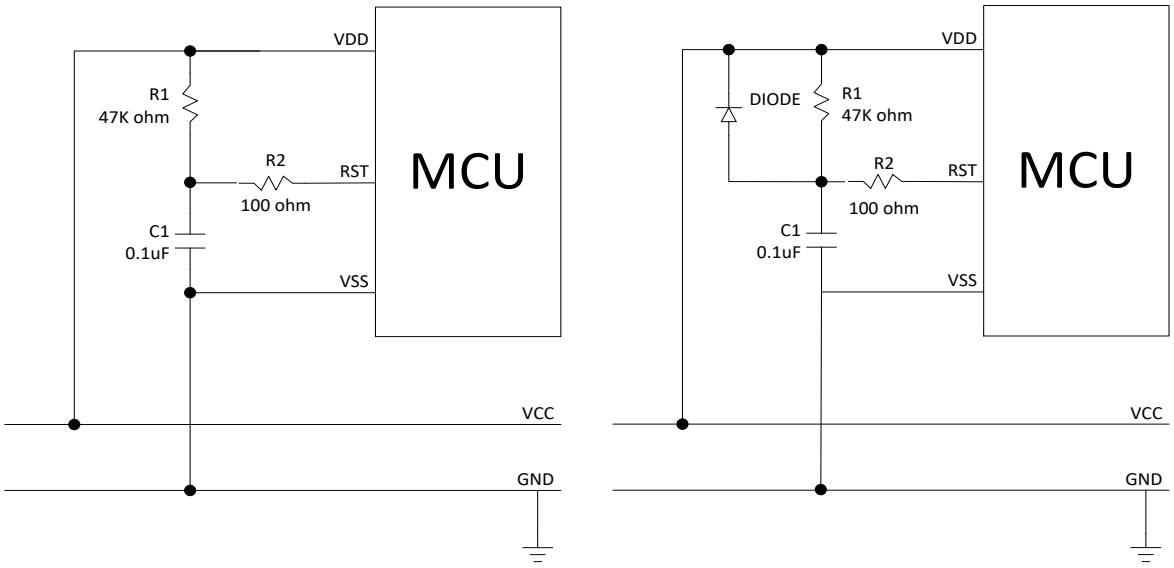
The operation mode of watchdog is configurable in options file:
Always mode counts its internal timer in all CPU operation modes (normal, IDLE, SLEEP);
Enable mode counts its internal timer during CPU stays in normal mode, and it would not trigger watchdog reset in IDLE and STOP modes;
Disable mode suspends 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	-	-	-	-	-
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..0	Reserved	R	0	

SRST Register (0xF7)

Bit	Field	Type	Initial	Description
7..1	Reserved	R	0	
0	SRSTREQ	R/W	0	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

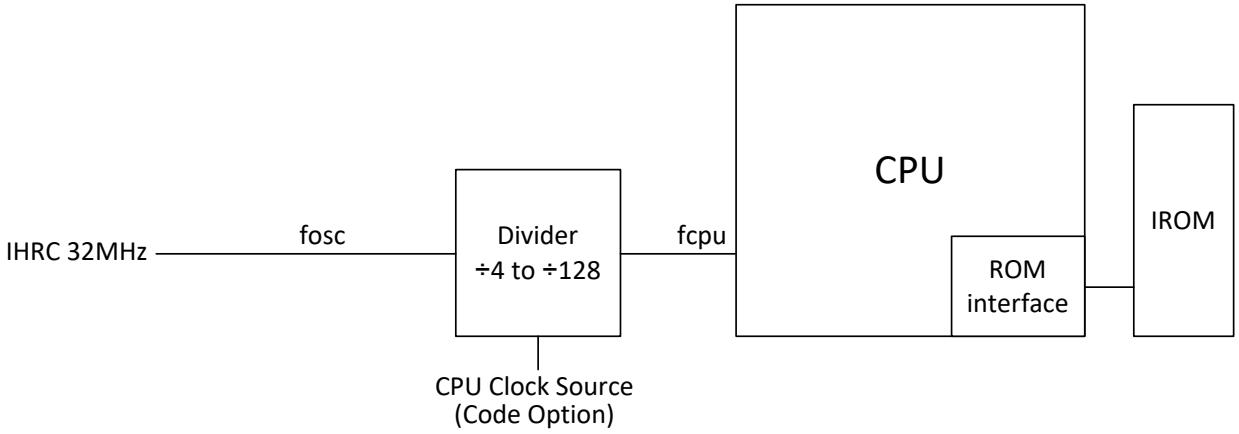
For power saving purpose, the microcontroller built in three different operation modes: normal, IDLE, and STOP mode.

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). 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

8.2 High Speed Clock

High-speed clock has only internal type. The internal high-speed oscillator is 32MHz RC type.

- IHRC 32M: The system high-speed clock source is internal high-speed 32MHz RC type oscillator.

8.3 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) 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.

- Any function can work in IDLE mode. Only CPU is suspended
- The IDLE mode wake-up sources are P0/P1 level change trigger and any interrupt event.

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.

- CPU, peripheral functions, and clock generator are suspended.

- The STOP mode wake-up source is P0/P1 level change trigger.

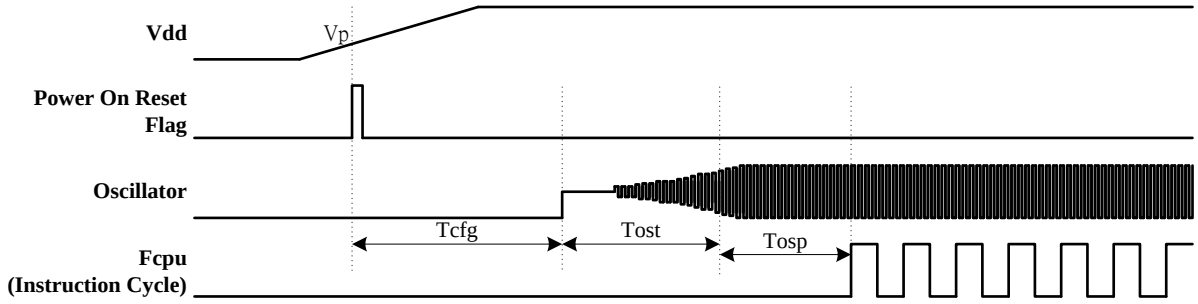
For user who is develop program in C language, IDLE and STOP macros is strongly recommended to control the microcontroller's system mode, instead of set IDLE and STOP bits directly.

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

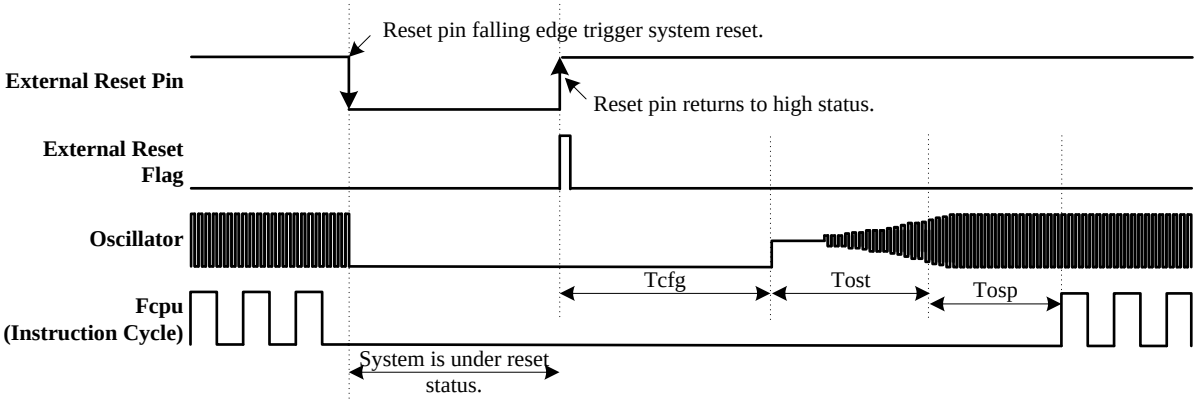
8.4 System Clock Timing

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.)	314us @ F_{hosc} = 32MHz
		Oscillator warm-up time of power down mode wake-up condition. $64*F_{hosc} + 4*F_{ILRC}$……RC type oscillator, e.g. internal high-speed RC type oscillator.	RC: 252us @ F_{hosc} = 32MHz

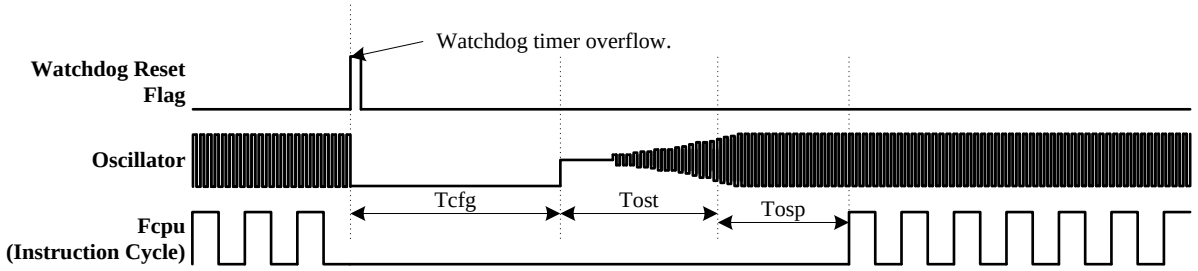
- Power On Reset Timing



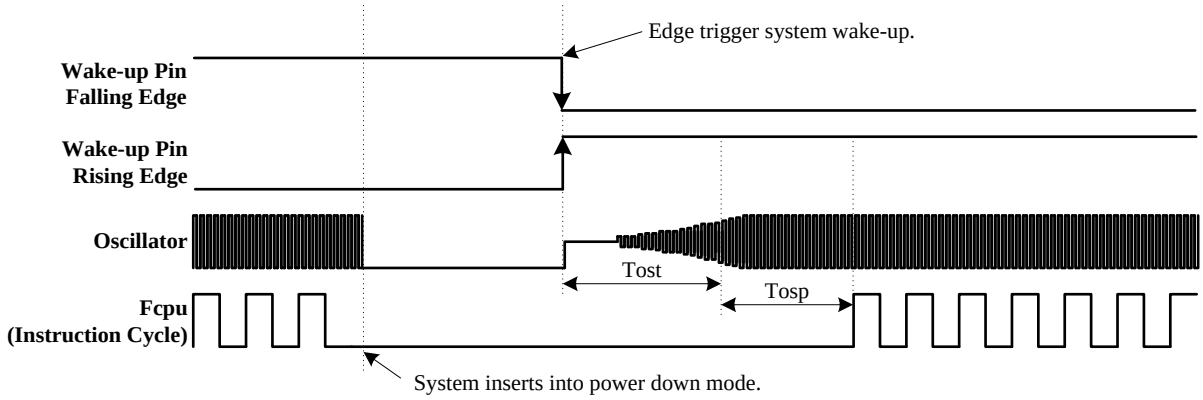
- External Reset Pin Reset Timing



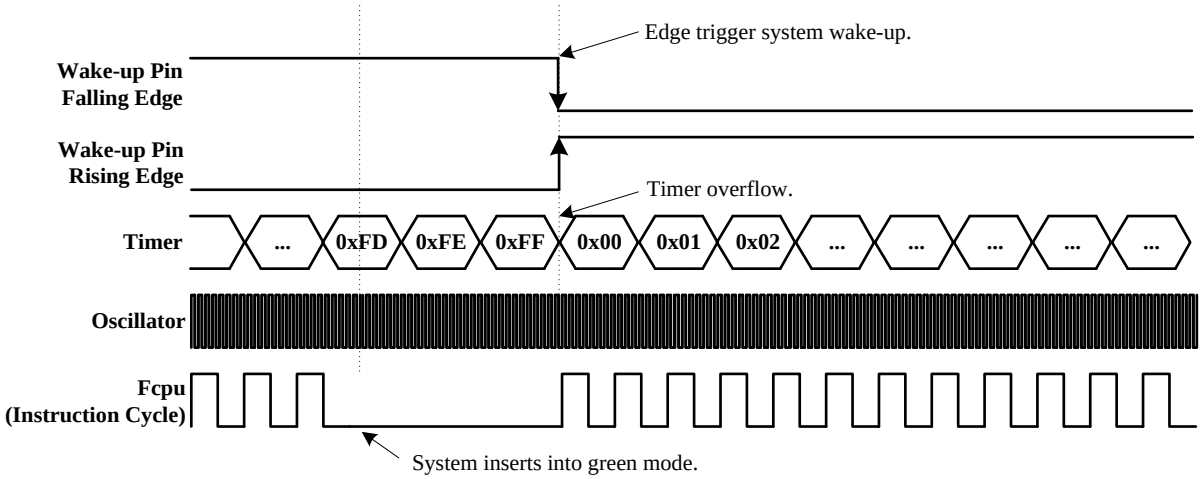
- Watchdog Reset Timing



- Power Down Mode Wake-up Timing

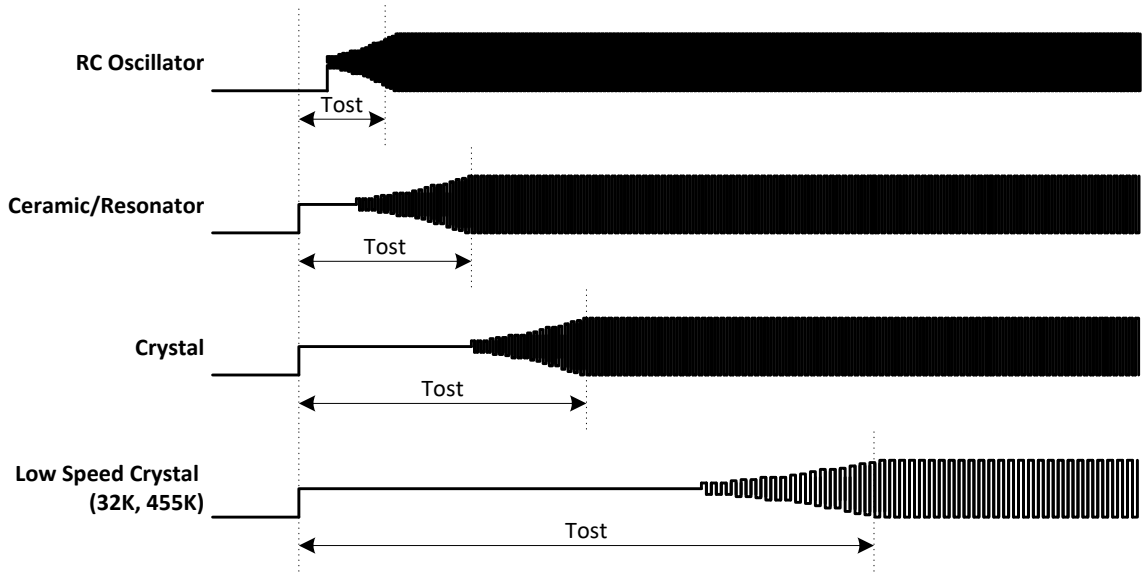


- Idle Mode Wake-up Timing



- 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.



8.5 System Clock and Power Management Registers

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

PCON Register (0x87)

Bit	Field	Type	Initial	Description
7				Refer to other chapter(s)
6..3	Reserved	R	0x00	
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

P1W Register (0x91)

Bit	Field	Type	Initial	Description
7..6	Reserved	R	0x00	
5..0	P1nW	R/W	0	0: Disable P1.n wakeup functionality 1: Enable P1.n wakeup functionality

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)

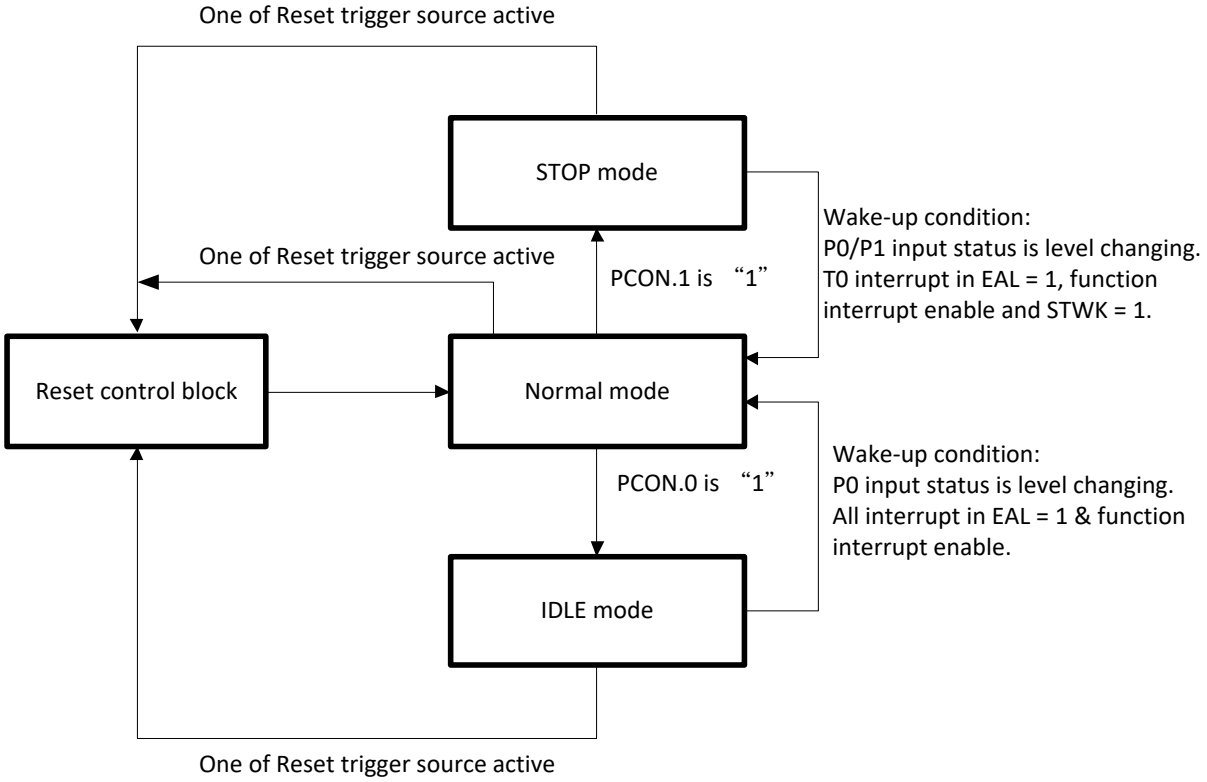


Table 9-1 The operating mode clock control

Operating Mode	Normal Mode	IDLE Mode	STOP Mode
IHRC	IHRC: Running	IHRC: Running	Stop
ILRC	Running	Running	STWK=1: Running Watchdog always: Running Other : 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 & STWK=1
Timer 1 (Timer, Event counter)	Active by TR1	Active by TR1	Inactive
PWM1/PWM2	Active as enable	Active as enable	Inactive
UART	Active as enable	Active as enable	Inactive
I2C	Active as enable	Active as enable	Inactive
SPI	Active as enable	Active as enable	Inactive
ADC	Active as enable	Active as enable	Inactive
Watchdog timer	By Watchdog Code option	By Watchdog Code option	By Watchdog Code option
Internal interrupt	All active	All active	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) T0 enable & clock source is Flosc (ILRC) & STWK=1 & function interrupt enable.

- 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 actives, 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 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 actives 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 and P1 wake-up function is controlled by P1W register. The STOP mode is also waked up by T0 interrupt when T0 clock source is Flosc and STWK bit is set. 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 internal high speed oscillator and internal low speed oscillator stop.
- Only internal regulator actives 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 sources include P0/P1 level change trigger and T0 interrupt when T0 clock source is Flosc (ILRC) and STWK bit is set.

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 actives 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 actives.
- 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 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 internal high clock oscillator RC type wakeup time is as the following.

The Wakeup time = 1/Fosc * 64 (sec) + 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, Flosc = 16KHz)

✷ ***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. Port 0 and Port 1 have wakeup function. Port 0 wakeup functions always enable. Port 1 wake-up function is controlled by P1W register.

P1W Register (0x91)

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 10 interrupt sources (1 external and 9 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
PWM1	EPW1	PW1F	Automatically	2 / 0x0083
I2C	EI2C	SI	By firmware	3 / 0x0043
Timer 0	ET0	TF0	Automatically	4 / 0x000B
PWM2	EPW2	PW2F	Automatically	5 / 0x008B
SPI	ESPI	SPIF/MODF	By firmware	6 / 0x004B
UART RX	EURX	RI0	By firmware	7 / 0x0053
Timer 1	ET1	TF1	Automatically	8 / 0x001B
UART TX	EUTX	TI0	By firmware	9 / 0x005B
ADC	EADC	ADCF	Automatically	10 / 0x00AB

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 actives. The program counter points to interrupt vector (0x03 – 0xAB) 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.

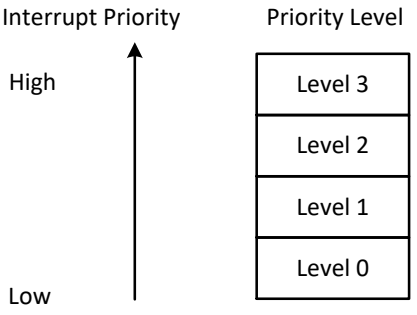
Priority Level	IP1.x	IP0.x
Level 0	0	0
Level 1	0	1
Level 2	1	0
Level 3	1	1

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.

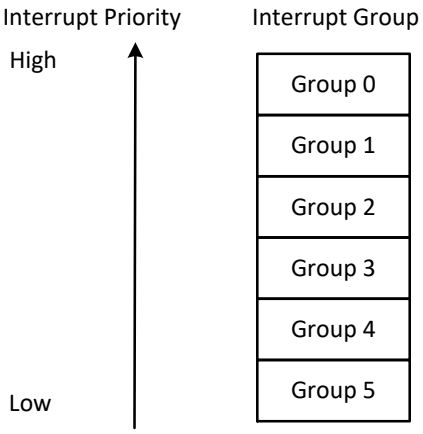
Group	Interrupt Source				
Group 0	INT0	-	PW1	-	I2C
Group 1	T0	-	PW2	-	SPI
Group 2	-	-	-	-	UART RX
Group 3	T1	-	-	-	UART TX
Group 4	-	-	-	-	-
Group 5	-	-	ADC	-	-

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.



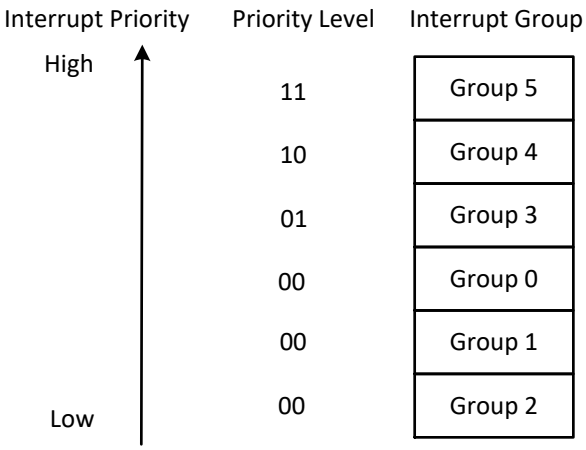
Higher priority level has higher priority



All groups within the same priority level
IP0.0 = IP0.1 = IP0.2 = IP0.3 = IP0.4 = IP0.5
IP1.0 = IP1.1 = IP1.2 = IP1.3 = IP1.4 = IP1.5

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
```



Priority Level: GP5>GP4>GP3>GP2=GP1=GP0

Interrupt Priority: GP5>GP4>GP3>GP0>GP1>GP2

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	-	-	-	ET1	-	ET0	EX0
IEN1	-	-	-	-	EURX	EUTX	ESPI	EI2C
IEN2	-	-	-	-	EPW2	EPW1	-	EADC
IRCON2	-	-	-	-	PW2F	PW1F	-	ADCF
TCON	TF1	TR1	TF0	TR0	-	-	IE0	-
S0CON	SM0	SM1	SM20	REN0	TB80	RB80	TI0	RI0
I2CCON	CR2	ENS1	STA	STO	SI	AA	CR1	CR0
SPSTA	SPIF	WCOL	SSERR	MODF	-	-	-	-

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.
3	ET1	R/W	0	T1 timer interrupt control bit. 0: Disable T1 interrupt function. 1: Enable T1 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 P1.0 interrupt (INT0) control bit. 0: Disable INT0 interrupt function. 1: Enable INT0 interrupt function.
Else	Reserved	R	0	

IEN1 Register (0xB8)

Bit	Field	Type	Initial	Description
3	EURX	R/W	0	UART RX interrupt control bit. 0: Disable UART RX interrupt function. 1: Enable UART RX interrupt function.
2	EUTX	R/W	0	UART TX interrupt control bit. 0: Disable UART TX interrupt function. 1: Enable UART TX interrupt function.

1	ESPI	R/W	0	SPI interrupt control bit. 0: Disable SPI interrupt function. 1: Enable SPI interrupt function.
0	EI2C	R/W	0	I2C interrupt control bit. 0: Disable I2C interrupt function. 1: Enable I2C interrupt function.
Else	Reserved	R	0	

IEN2 Register (0x9A)

Bit	Field	Type	Initial	Description
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.
0	EADC	R/W	0	ADC interrupt control bit. 0: Disable ADC interrupt function. 1: Enable ADC interrupt function.
Else	Reserved	R	0	

IRCON2 Register (0xBF)

Bit	Field	Type	Initial	Description
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.
0	ADCF	R/W	0	ADC interrupt request flag. 0: None ADC interrupt request. 1: ADC interrupt request.
Else	Reserved	R	0	

TCON Register (0X88)

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

S0CON Register (0x98)

Bit	Field	Type	Initial	Description
1	TI0	R/W	0	UART transmit interrupt request flag. It indicates completion of a serial transmission at UART. 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 UART transmit interrupt request. 1: UART transmit interrupt request.
0	RI0	R/W	0	UART receive interrupt request flag. It is set by hardware after completion of a serial reception at UART. 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 UART receive interrupt request. 1: UART 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)

SPSTA Register (0xE1)

Bit	Field	Type	Initial	Description
7	SPIF	R/W	0	Serial Peripheral Data Transfer Flag Set by hardware upon data transfer completion. Cleared by reading the SPSTA register with the SPIF bit set, and then reading the SPDAT register.
4	MODF	R/W	0	Mode Fault Flag Set by hardware when the "ssn" pin level is in conflict with actual mode of the SPI_MS controller (configured as master while externally selected as slave). Cleared by hardware when the "ssn" pin is at appropriate level and the SPCON register be writed any value.
Else				Refer to other chapter(s)

Example: Defining Interrupt Vector. The interrupt service routine is following user program.

```
                ORG     0               ; 0000H
                JMP     START           ; Jump to user program address.
                …
                ORG     0X0003          ; Jump to interrupt service routine address.
                JMP     ISR_INT0
                ORG     0X000B
                JMP     ISR_T0
                …
                ORG     0X0083
                JMP     ISR_PWM1
                …
                ORG     0X00AC
START:                                  ; 00ACH, The head of user program.
                …                       ; User program.
                …
                …
                JMP     START           ; End of user program.
                …
ISR_ INT0:                              ; The head of interrupt service routine.
                PUSH    ACC             ; Save ACC to stack buffer.
                PUSH    PSW             ; Save PSW to stack buffer.
                …
                POP     PSW             ; Load PSW from stack buffer.
                POP     ACC             ; Load ACC from stack buffer.
                RETI                    ; End of interrupt service routine.
ISR_T0:                                 ;
                PUSH    ACC             ; Save ACC to stack buffer.
                PUSH    PSW             ; Save PSW to stack buffer.
                …
                POP     PSW             ; Load PSW from stack buffer.
                POP     ACC             ; Load ACC from stack buffer.
                RETI                    ; End of interrupt service routine.
                …
ISR_ PWM1                               ;
                PUSH    ACC             ; Save ACC to stack buffer.
                PUSH    PSW             ; Save PSW to stack buffer.
                …
                POP     PSW             ; Load PSW from stack buffer.
                POP     ACC             ; Load ACC from stack buffer.
                RETI                    ; End of interrupt service routine.


                END                     ; End of program.
```

11 GPIO

The microcontroller has up to 12 bidirectional general purpose I/O pin (GPIO). Unlike the original 8051 only has open-drain output, SN8F5721 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 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	-	-	P05M	P04M	P03M	P02M	P01M	P00M
P1M	-	-	P15M	P14M	P13M	P12M	P11M	P10M
P0OC	P14OC	P13OC	P12OC	P11OC	P04OC	P03OC	P01OC	P00OC

P0M: 0xF9

Bit	Field	Type	Initial	Description
7..6	Reserved	R	0	
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..6	Reserved	R	0	
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

P0OC Register (0xBD)

Bit	Field	Type	Initial	Description
7	P14OC	R/W	0	P1.4 open-drain control bit. 0: Disable open-drain mode 1: Enable open-drain mode
6	P13OC	R/W	0	P1.3 open-drain control bit. 0: Disable open-drain mode 1: Enable open-drain mode
5	P12OC	R/W	0	P1.2 open-drain control bit. 0: Disable open-drain mode 1: Enable open-drain mode
4	P11OC	R/W	0	P1.1 open-drain control bit. 0: Disable open-drain mode 1: Enable open-drain mode
3	P04OC	R/W	0	P0.4 open-drain control bit. 0: Disable open-drain mode

				1: Enable open-drain mode
2	P03OC	R/W	0	P0.3 open-drain control bit. 0: Disable open-drain mode 1: Enable open-drain mode
1	P01OC	R/W	0	P0.1 open-drain control bit. 0: Disable open-drain mode 1: Enable open-drain mode
0	P00OC	R/W	0	P0.0 open-drain control bit. 0: Disable open-drain mode 1: Enable open-drain mode

* Recommended disable open-drain output if P11 executes debug interface function.

11.2 Input Data and Output Data

By a read operation from any register of P0 to P1, 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 P1 register value would be latched immediately, yet the value would be outputted until the mapped P0M – P1M is set to output mode. If the pin is currently in output mode, any value set to P0 to P1 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	-	-	P05	P04	P03	P02	P01	P00
P1	-	-	P15	P14	P13	P12	P11	P10

P0: 0x80

Bit	Field	Type	Initial	Description
7..6	Reserved	R	0	
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..6	Reserved	R	0	
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)

11.3 On-chip Pull-up Resisters

The P0UR to P1UR register are mapped to each pins' internal 100 kΩ (in typical value) pull-up resister.

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0UR	-	-	P05UR	P04UR	P03UR	P02UR	P01UR	P00UR
P1UR	-	-	P15UR	P14UR	P13UR	P12UR	P11UR	P10UR

P0UR: 0xF1

Bit	Field	Type	Initial	Description
7..6	Reserved	R	0	
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: 0xF2

Bit	Field	Type	Initial	Description
7..6	Reserved	R	0	
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

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	-	-	P0CON5	P0CON4	P0CON3	P0CON2	P0CON1	P0CON0
P1CON	-	-	P1CON5	P1CON4	P1CON3	P1CON2	P1CON1	P1CON0

P0CON: 0xD6

Bit	Field	Type	Initial	Description
5..0	P0CON[5:0]	R/W	0	P0 configuration control bit*. 0: P0 can be analog input pin or digital GPIO pin. 1: P0 is pure analog input pin and can't be a digital GPIO pin.

P1CON: 0x9F

Bit	Field	Type	Initial	Description
5..0	P1CON[5:0]	R/W	0	P1 configuration control bit*. 0: P1 can be analog input pin or digital GPIO pin. 1: P1 is pure analog input pin and can't be a digital GPIO pin.

* P0CON [5:0], P1CON [5:0] will configure related Port0, Port1 pin as pure analog input pin to avoid current leakage.

12 External Interrupt

INT0 is external interrupt trigger sources. Build in edge trigger configuration function and edge direction is selected by PEDGE register. When both external interrupt (EX0) and global interrupt (EAL) are enabled, the external interrupt request flag (IE0) will be set to “1” as edge trigger event occurs. The program counter will jump to the interrupt vector (ORG 0x0003) and execute interrupt service routine. Interrupt request flag will be cleared by hardware before ISR is executed.

12.1 External Interrupt Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PEDGE	-	-	-	-	-	-	EX0G1	EX0G0
IEN0	EAL	-	-	-	ET1	-	ET0	EX0
TCON	TF1	TR1	TF0	TR0	-	-	IE0	-

PEDGE Register (0x8F)

Bit	Field	Type	Initial	Description
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
Else	Reserved	R	0	

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.
0	EX0	R/W	0	External P1.0 interrupt (INT0) control bit. 0: Disable INT0 interrupt function. 1: Enable INT0 interrupt function.
Else				Refer to other chapter(s)

TCON Register (0x88)

Bit	Field	Type	Initial	Description
1	IE0	R/W	0	External P1.0 interrupt (INT0) request flag 0: None INT0 interrupt request. 1: INT0 interrupt request.
Else				Refer to other chapter(s)

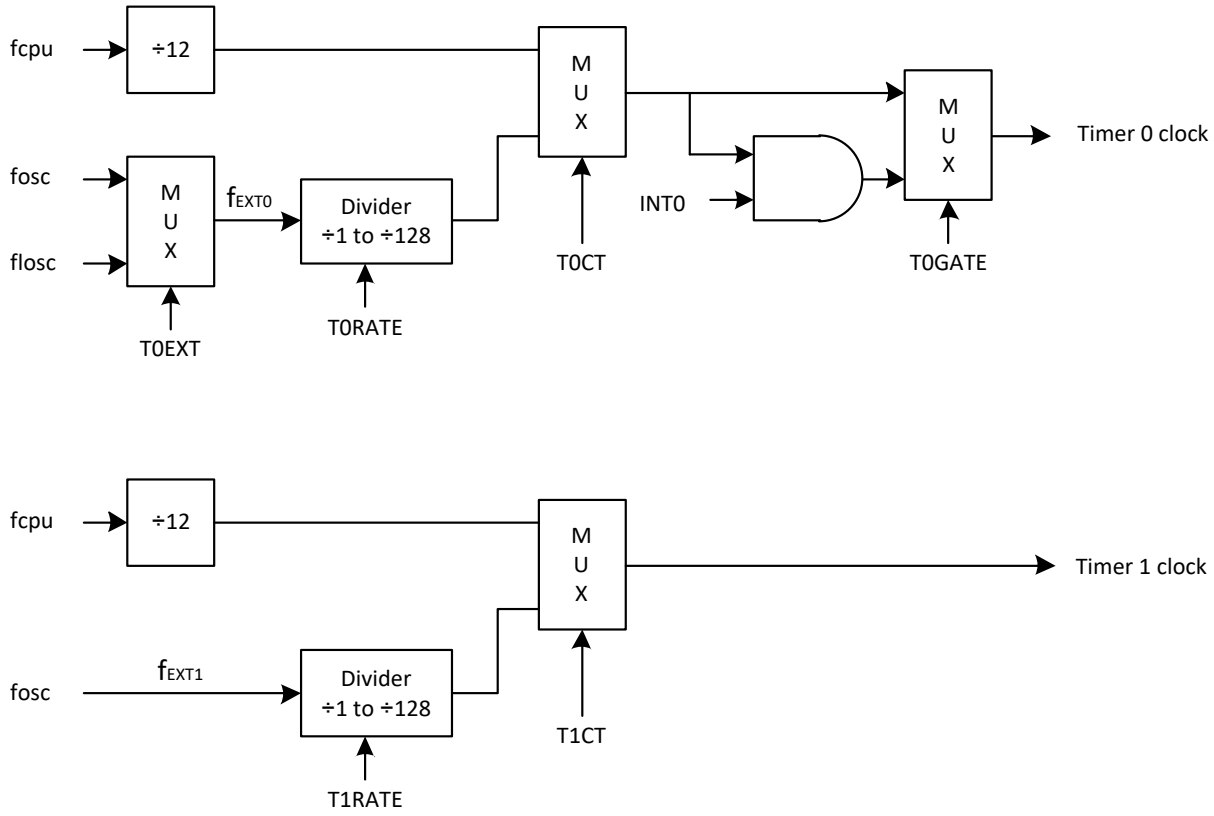
13 Timer 0 and Timer 1

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.

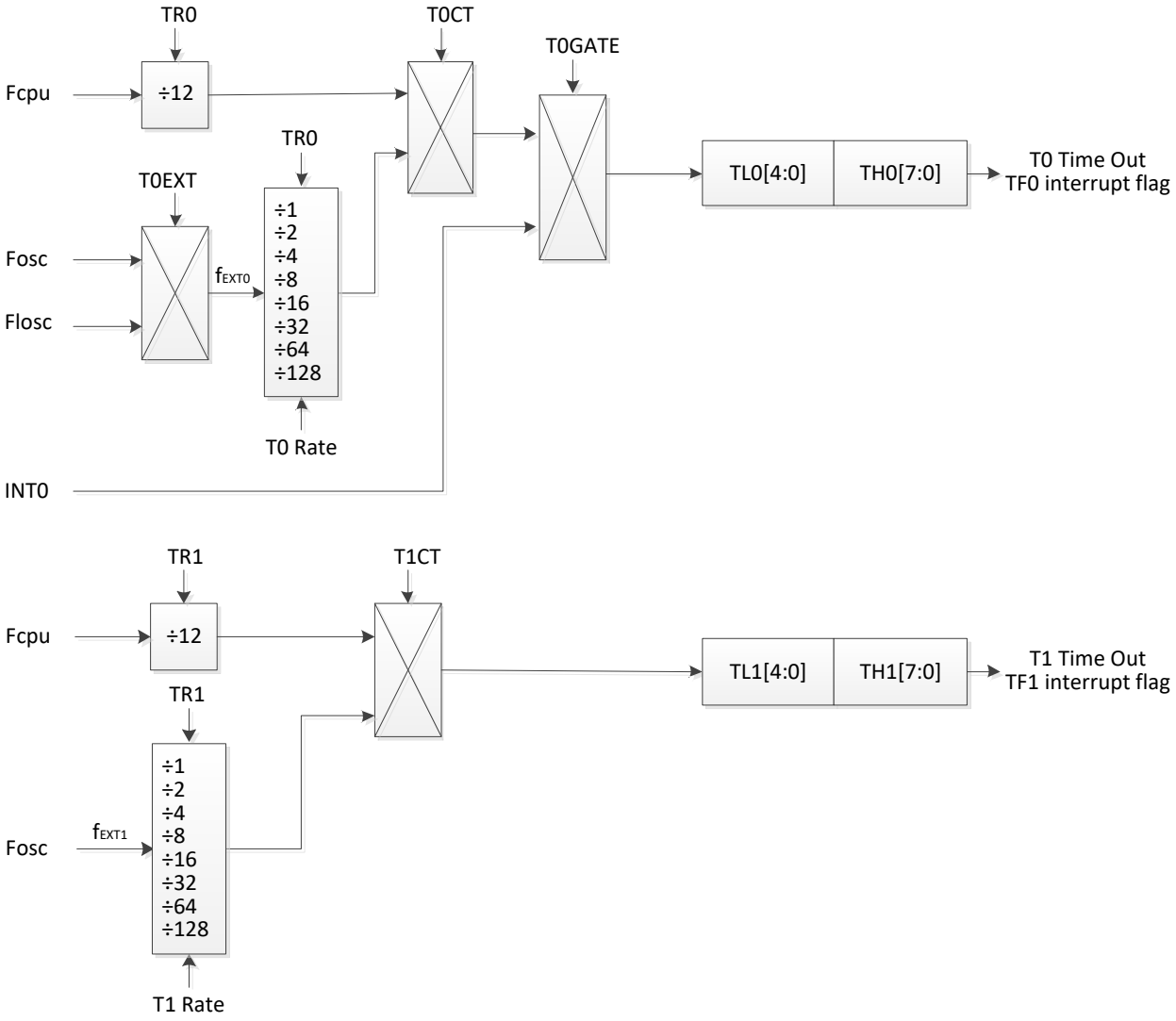
13.1 Timer 0 and Timer 1 Clock Selection

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. Overall, the major difference between the two timers is that Timer 0 additionally supports flosc clock source (low speed clock).



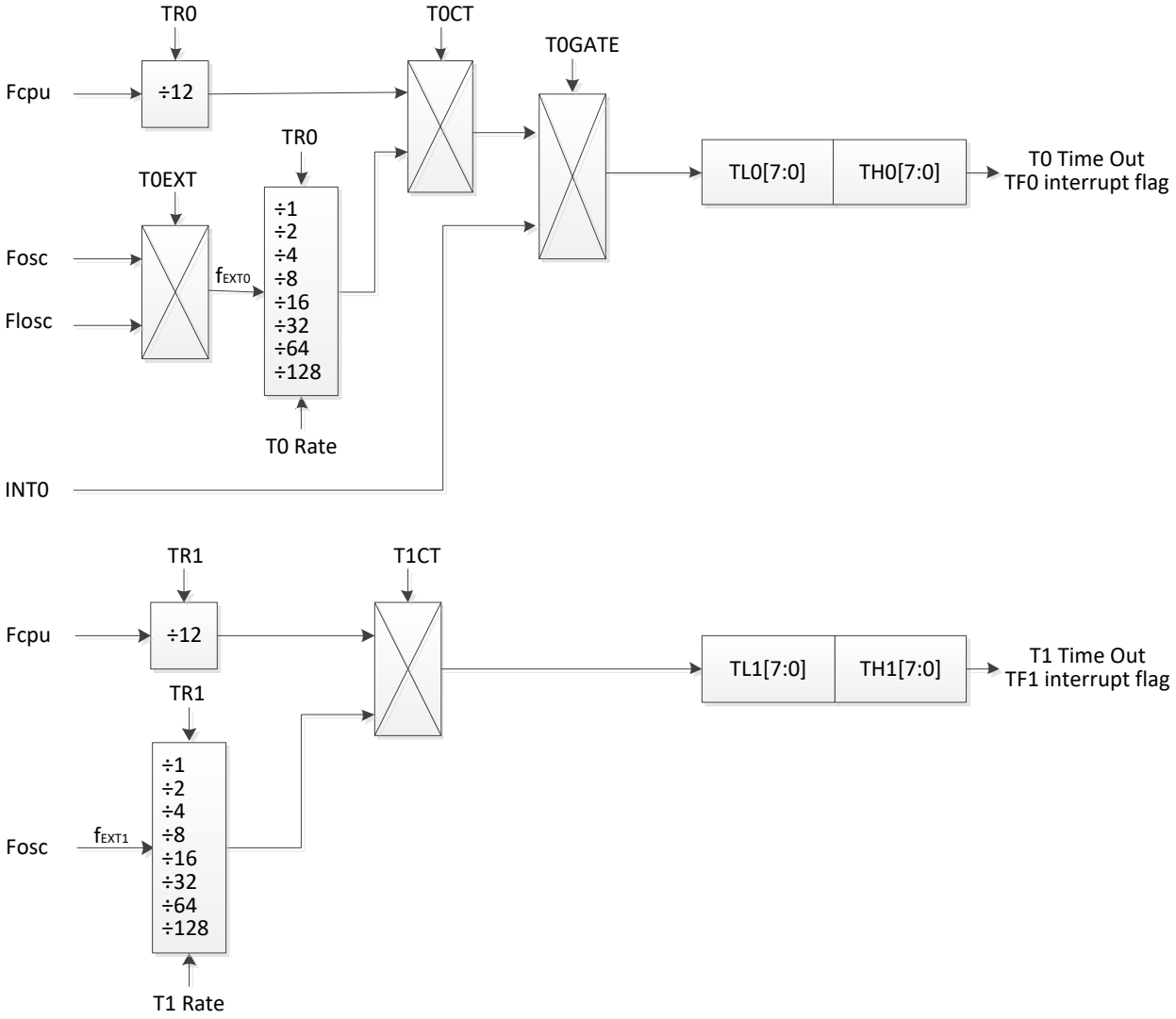
13.2 Mode 0: 13-bit Up Counting Timer

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 and writable by firmware if ET0/ET1 does not apply, or can be handled by interrupt controller if ET0/ET1 is applied.



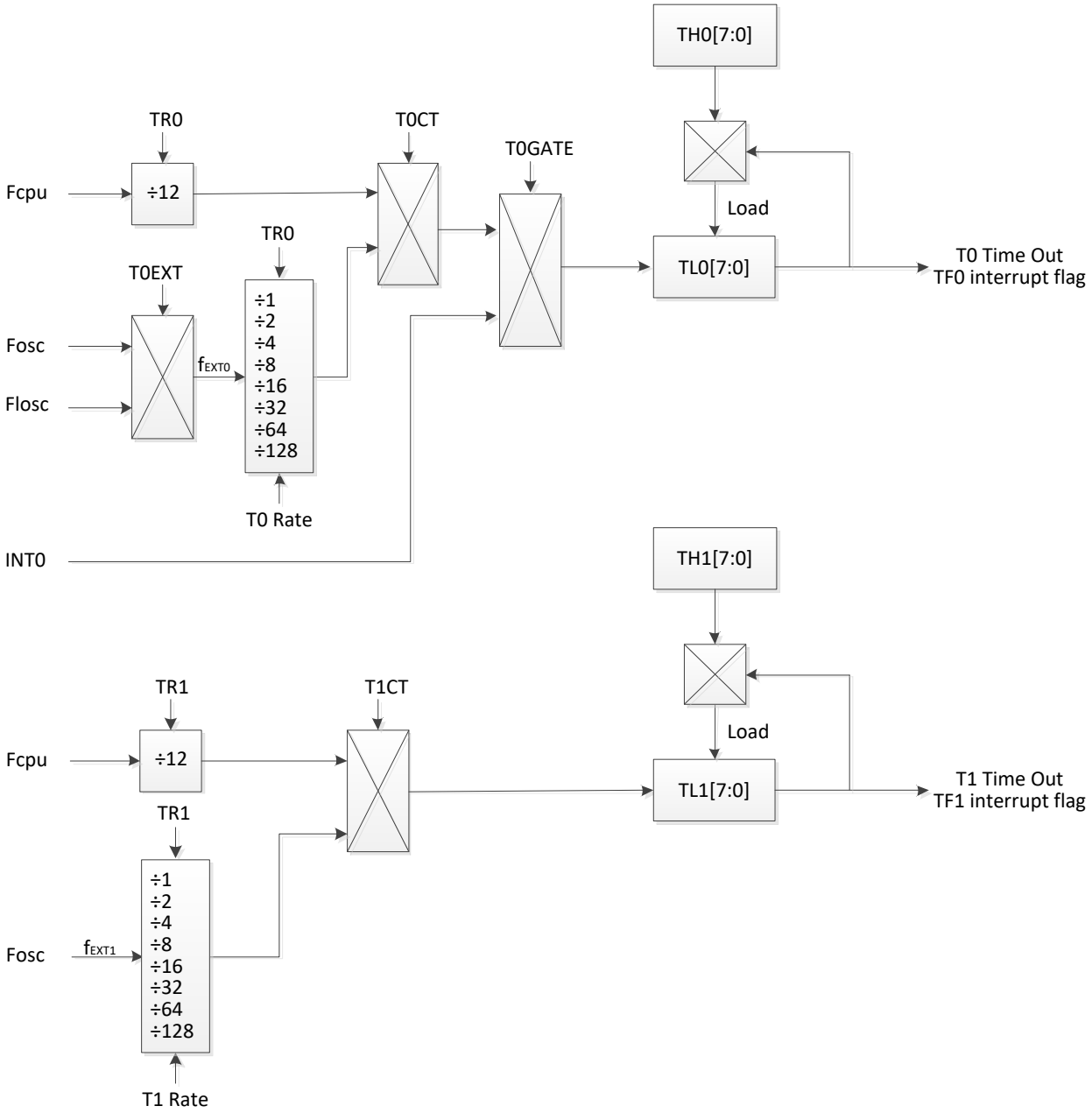
13.3 Mode 1: 16-bit Up Counting Timer

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 and writable by firmware or can be handled by interrupt controller (if ET0/ET1 applied).



13.4 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.

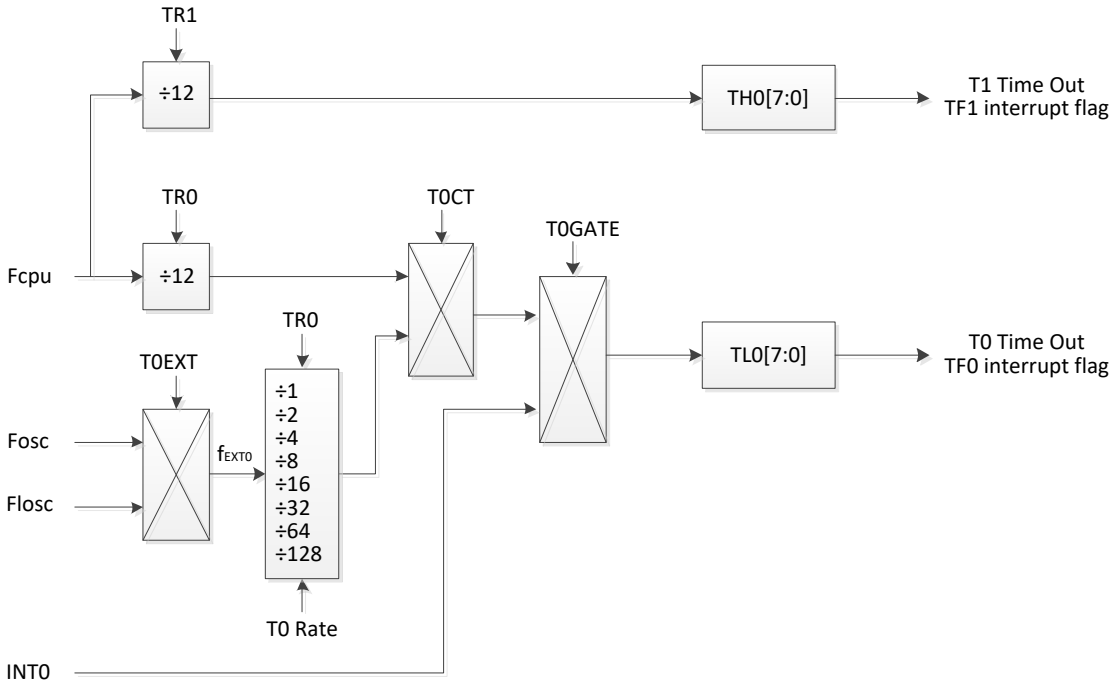


13.5 Mode 3 (Timer 0 only): Separated Two 8-bit Up Counting Timer

Mode 3 treats TH0 and TL0 as two separated 8-bit timers. TL0 is an 8-bit up counting timer with 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.



13.6 Timer 0 and Timer 1 Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TCON	TF1	TR1	TF0	TR0	-	-	IE0	-
TCON0	T0EXT	T0RATE2	T0RATE1	T0RATE0	-	T1RATE2	T1RATE1	T1RATE0
TMOD	-	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	EAL	-	-	-	ET1	-	ET0	EX0

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
3	Reserved	R	0	
2	Reserved	R	0	
1	IE0	R/W	0	Refer to INT0
0	Reserved	R	0	

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)

TCON0 Register (0xE7)

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 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 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

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

TMOD Register (0x89)

Bit	Field	Type	Initial	Description
7	Reserved	R	0	
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 INT0
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

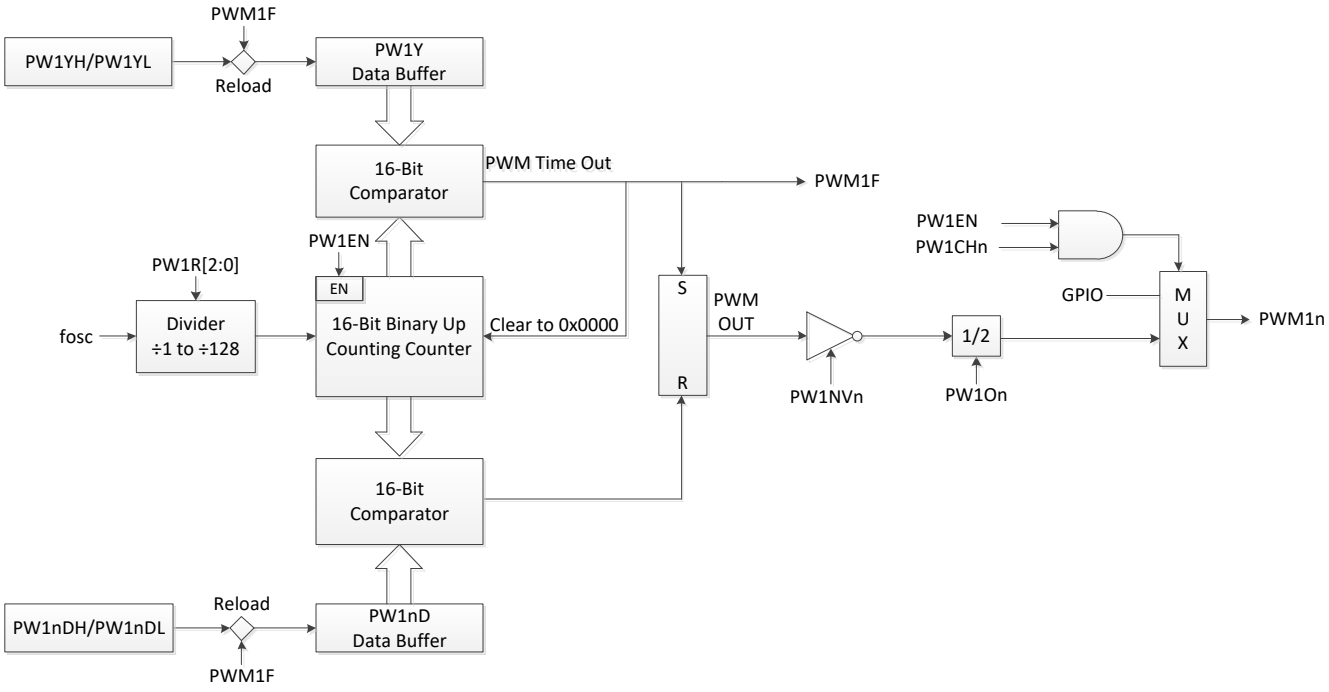
*(1) f_{EXT1} = fosc.

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

14 PWM1

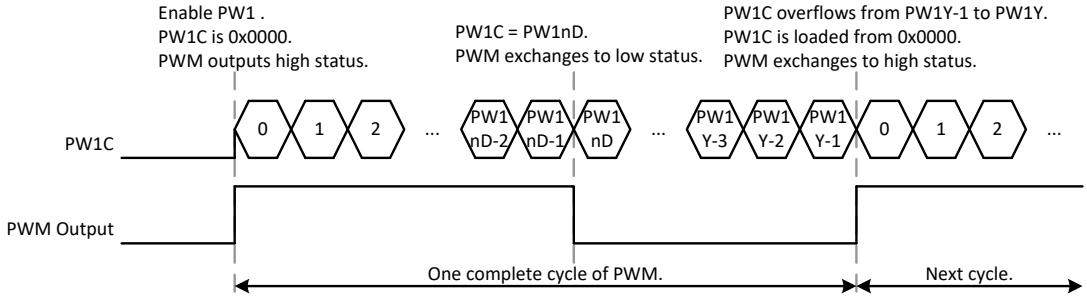
The PW1 timer is a 16-bit up counting timer and supports 6-channel general PWM function. By the counter reaches the up-boundary value (PW1Y), it clears its counter and triggers an interrupt signal. PWM's duty cycle is controlled by PW10D~PW15D register. Each PWM channel has its own duty control.

The PWM function has 6 programmable channels shared with GPIO pins and controlled by PW1CH[5:0] bits. The output operation must be through enabled each bit/channel of PW1CH[5:0] bits. The enabled PWM channel exchanges from GPIO to PWM output. When the PW1CH[5:0] 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.



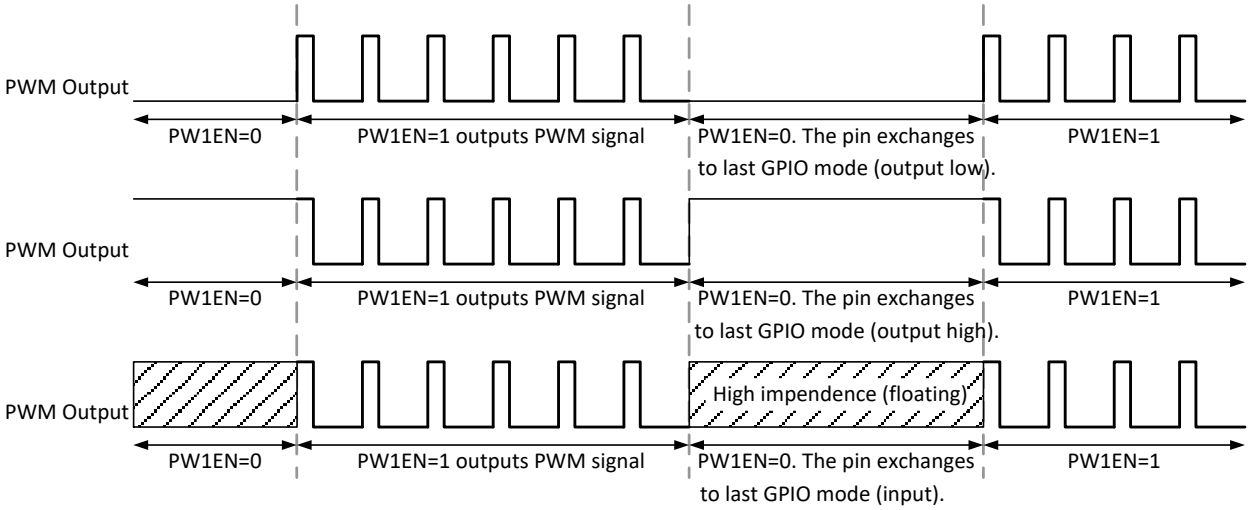
14.1 General PWM

PW1 timer builds in PWM function controlled by PW1EN register and PW1CH bits. PWM10 - PWM15 are output pins. Those output pins are shared with GPIO pin controlled by PW1CH[5:0] 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.



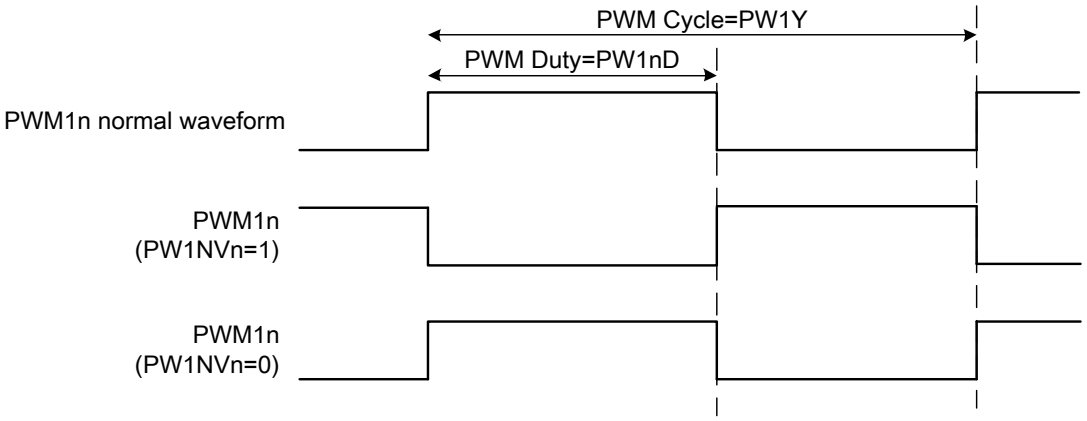
PWM Period = PW1Y

PWM duty = (PW1nD): (PW1Y-PW1nD)

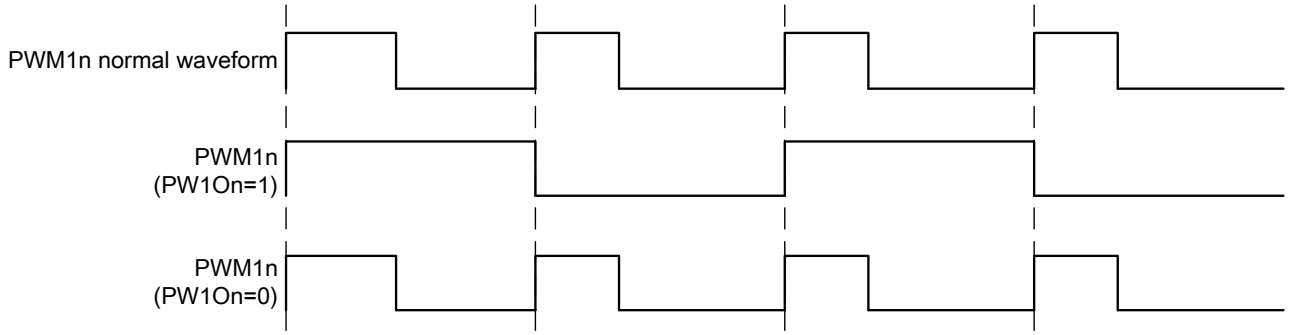


14.2 Inverse and Frequency Mode

The PWM builds in inverse output function. The inverse mode is controlled by PW1NV[5:0]. When PW1NVn= 1, the PWM1n outputs the inverse PWM signal of PWM. When PW1NVn = 0, the PWM1n outputs the non-inverse PWM signal of PWM. The inverse PWM output waveform is below diagram.



The PWM has frequency mode to change PWM output frequency. The frequency mode is controlled by PW1O[5:0]. When PW1On=1, PWM1n pin outputs 1/2*frequency PWM signal. When PW1On=0, PWMn pin outputs 1*frequency PWM signal.



14.3 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	-	-	PW1CH5	PW1CH4	PW1CH3	PW1CH2	PW1CH1	PW1CH0
PW1NV	-	-	PW1NV5	PW1NV4	PW1NV3	PW1NV2	PW1NV1	PW1NV0
PW1O	-	-	PW1O5	PW1O4	PW1O3	PW1O2	PW1O1	PW1O0
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
PW14DH	PW14D15	PW14D14	PW14D13	PW14D12	PW14D11	PW14D10	PW14D9	PW14D8
PW14DL	PW14D7	PW14D6	PW14D5	PW14D4	PW14D3	PW14D2	PW14D1	PW14D0
PW15DH	PW15D15	PW15D14	PW15D13	PW15D12	PW15D11	PW15D10	PW15D9	PW15D8
PW15DL	PW15D7	PW15D6	PW15D5	PW15D4	PW15D3	PW15D2	PW15D1	PW15D0
IEN0	EAL	-	-	-	ET1	-	ET0	EX0
IEN2	-	-	-	-	EPW2	EPW1	-	EADC
IRCON2	-	-	-	-	PW2F	PW1F	-	ADCF

PW1M Registers (0xB0)

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
Else	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 (0xB1)

Bit	Field	Type	Initial	Description
7..6	Reserved	R	0	
5	PW1CH5	R/W	0	PWM15 shared-pin control 0: GPIO 1: PWM output (shared with P0.5)

4	PW1CH4	R/W	0	PWM14 shared-pin control 0: GPIO 1: PWM output (shared with P0.4)
3	PW1CH3	R/W	0	PWM13 shared-pin control 0: GPIO 1: PWM output (shared with P0.3)
2	PW1CH2	R/W	0	PWM12 shared-pin control 0: GPIO 1: PWM output (shared with P0.2)
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)

PW1NV Registers (0xB2)

Bit	Field	Type	Initial	Description
7..6	Reserved	R	0	
5	PW1NV5	R/W	0	PWM15 shared-pin inverse control 0: Non-inverse. 1: Inverse.
4	PW1NV4	R/W	0	PWM14 shared-pin inverse control 0: Non-inverse. 1: Inverse.
3	PW1NV3	R/W	0	PWM13 shared-pin inverse control 0: Non-inverse. 1: Inverse.
2	PW1NV2	R/W	0	PWM12 shared-pin inverse control 0: Non-inverse. 1: Inverse.
1	PW1NV1	R/W	0	PWM11 shared-pin inverse control 0: Non-inverse. 1: Inverse.
0	PW1NV0	R/W	0	PWM10 shared-pin inverse control 0: Non-inverse. 1: Inverse.

PW1O Registers (0xB3)

Bit	Field	Type	Initial	Description
7..6	Reserved	R	0	
5	PW1O5	R/W	0	PWM15 shared-pin frequency control 0: 1*Frequency PWM signal. 1: 1/2 *Frequency PWM signal.
4	PW1O4	R/W	0	PWM14 shared-pin frequency control 0: 1*Frequency PWM signal. 1: 1/2 *Frequency PWM signal.
3	PW1O3	R/W	0	PWM13 shared-pin frequency control 0: 1*Frequency PWM signal. 1: 1/2 *Frequency PWM signal.
2	PW1O2	R/W	0	PWM12 shared-pin frequency control 0: 1*Frequency PWM signal. 1: 1/2 *Frequency PWM signal.
1	PW1O1	R/W	0	PWM11 shared-pin frequency control 0: 1*Frequency PWM signal. 1: 1/2 *Frequency PWM signal.
0	PW1O0	R/W	0	PWM10 shared-pin frequency control 0: 1*Frequency PWM signal. 1: 1/2 *Frequency PWM signal.

PW1YH/PW1YL Registers (PW1YH: 0xA2, PW1YL: 0xA1)

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: 0xA4, PW10DL: 0xA3)

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

PW11DH/PW11DL Registers (PW11DH: 0xA6, PW11DL: 0xA5)

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

PW12DH/PW12DL Registers (PW12DH: 0xAB, PW12DL: 0xAA)

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

PW13DH/PW13DL Registers (PW13DH: 0xAD, PW13DL: 0xAC)

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

PW14DH/PW14DL Registers (PW14DH: 0xAF, PW14DL: 0xAE)

Bit	Field	Type	Initial	Description
7..0	PW14DH/L	R/W	0x00	16-bit PWM1 duty control.

PW15DH/PW15DL Registers (PW15DH: 0xB5, PW15DL: 0xB4)

Bit	Field	Type	Initial	Description
7..0	PW15DH/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 (0x9A)

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)

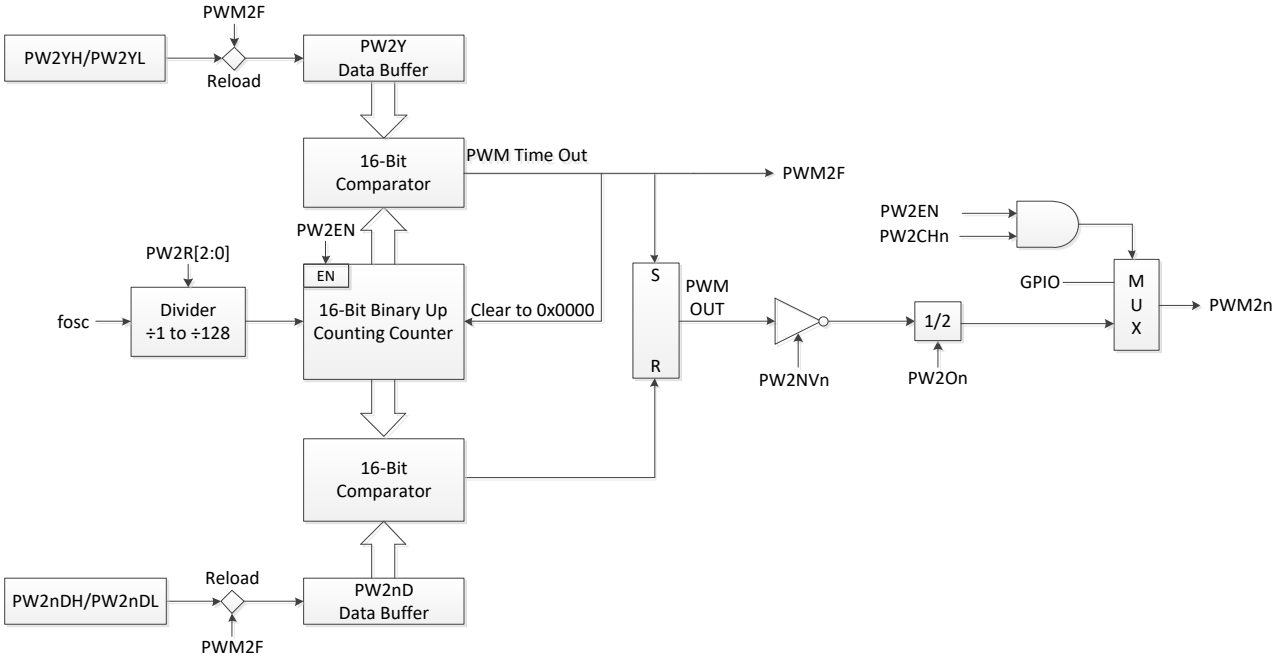
IRCON2 Register (0xBF)

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)

15 PWM2

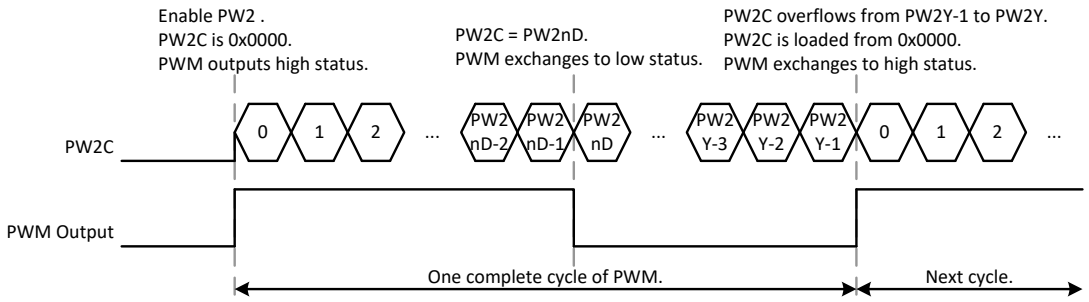
The PW2 timer is a 16-bit up counting timer and supports 6-channel general PWM function. By the counter reaches the up-boundary value (PW2Y), it clears its counter and triggers an interrupt signal. PWM's duty cycle is controlled by PW20D~PW25D register. Each PWM channel has its own duty control.

The PWM function has 6 programmable channels shared with GPIO pins and controlled by PW2CH[5:0] bits. The output operation must be through enabled each bit/channel of PW2CH[5:0] bits. The enabled PWM channel exchanges from GPIO to PWM output. When the PW2CH[5:0] 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.



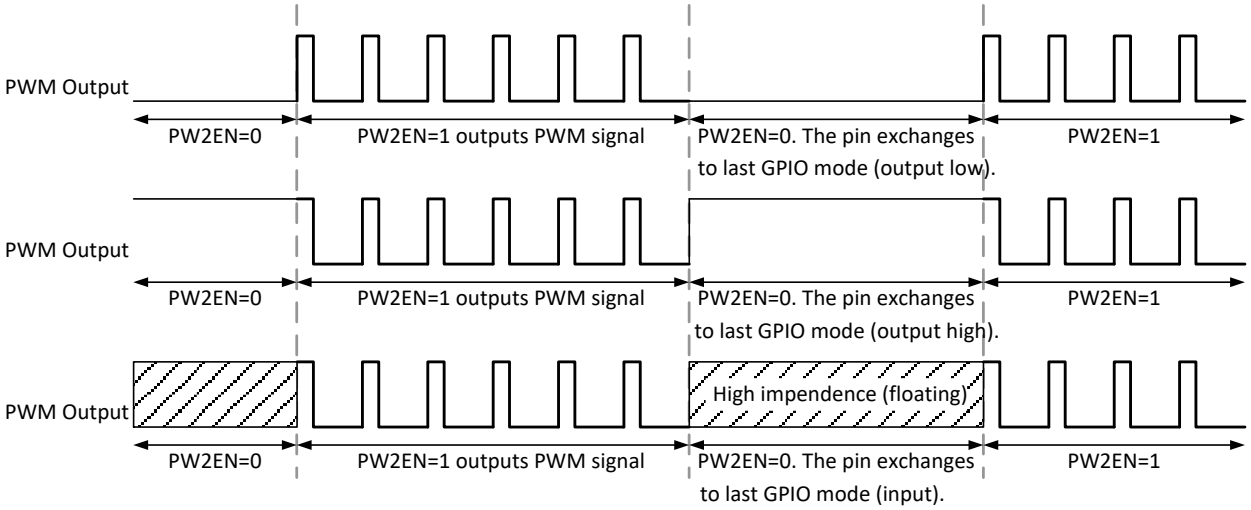
15.1 General PWM

PW2 timer builds in PWM function controlled by PW2EN register and PW2CH bits. PWM20 – PWM25 are output pins. Those output pins are shared with GPIO pin controlled by PW2CH[5:0] 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.



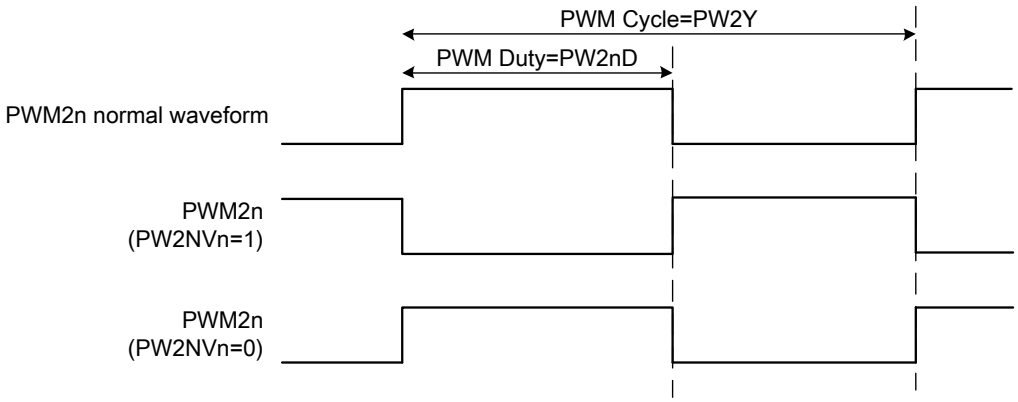
PWM Period = PW2Y

PWM duty = (PW2nD): (PW2Y-PW2nD)

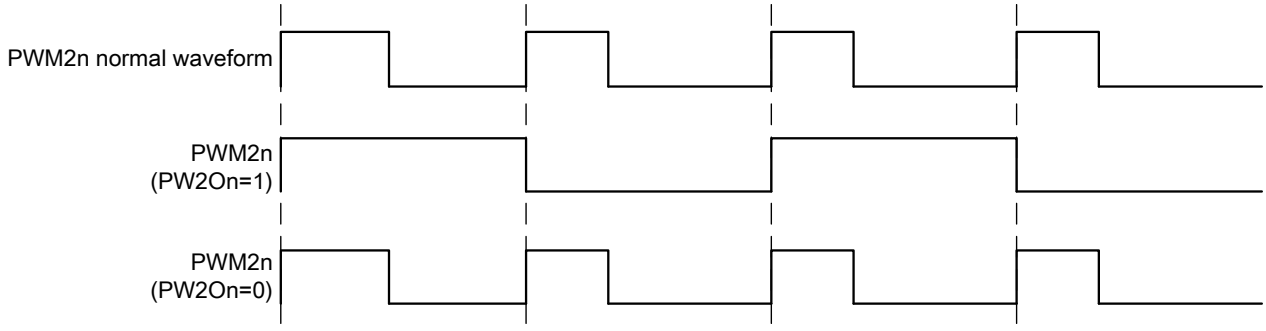


15.2 Inverse and Frequency Mode

The PWM builds in inverse output function. The inverse mode is controlled by PW2NV[5:0]. When PW2NVn= 1, the PWM2n outputs the inverse PWM signal of PWM. When PW2NVn = 0, the PWM2n outputs the non-inverse PWM signal of PWM. The inverse PWM output waveform is below diagram.



The PWM has frequency mode to change PWM output frequency. The frequency mode is controlled by PW2O[5:0]. When PW2On=1, PWM2n pin outputs 1/2*frequency PWM signal. When PW2On=0, PWM2n pin outputs 1*frequency PWM signal.



15.3 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	-	-	PW2CH5	PW2CH4	PW2CH3	PW2CH2	PW2CH1	PW2CH0
PW2NV	-	-	PW2NV5	PW2NV4	PW2NV3	PW2NV2	PW2NV1	PW2NV0
PW2O	-	-	PW2O5	PW2O4	PW2O3	PW2O2	PW2O1	PW2O0
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
PW24DH	PW24D15	PW24D14	PW24D13	PW24D12	PW24D11	PW24D10	PW24D9	PW24D8
PW24DL	PW24D7	PW24D6	PW24D5	PW24D4	PW24D3	PW24D2	PW24D1	PW24D0
PW25DH	PW25D15	PW25D14	PW25D13	PW25D12	PW25D11	PW25D10	PW25D9	PW25D8
PW25DL	PW25D7	PW25D6	PW25D5	PW25D4	PW25D3	PW25D2	PW25D1	PW25D0
IEN0	EAL	-	-	-	ET1	-	ET0	EX0
IEN2	-	-	-	-	EPW2	EPW1	-	EADC
IRCON2	-	-	-	-	PW2F	PW1F	-	ADCF

PW2M Registers (0xC8)

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
Else	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 (0xC9)

Bit	Field	Type	Initial	Description
7..6	Reserved	R	0	
5	PW2CH5	R/W	0	PWM25 shared-pin control 0: GPIO 1: PWM output (shared with P1.5)
4	PW2CH4	R/W	0	PWM24 shared-pin control

				0: GPIO 1: PWM output (shared with P1.4)
3	PW2CH3	R/W	0	PWM23 shared-pin control 0: GPIO 1: PWM output (shared with P1.3)
2	PW2CH2	R/W	0	PWM22 shared-pin control 0: GPIO 1: PWM output (shared with P1.2)
1	PW2CH1	R/W	0	PWM21 shared-pin control 0: GPIO 1: PWM output (shared with P1.1)
0	PW2CH0	R/W	0	PWM20 shared-pin control 0: GPIO 1: PWM output (shared with P1.0)

PW2NV Registers (0xCA)

Bit	Field	Type	Initial	Description
7..6	Reserved	R	0	
5	PW2NV5	R/W	0	PWM25 shared-pin inverse control 0: Non-inverse. 1: Inverse.
4	PW2NV4	R/W	0	PWM24 shared-pin inverse control 0: Non-inverse. 1: Inverse.
3	PW2NV3	R/W	0	PWM23 shared-pin inverse control 0: Non-inverse. 1: Inverse.
2	PW2NV2	R/W	0	PWM22 shared-pin inverse control 0: Non-inverse. 1: Inverse.
1	PW2NV1	R/W	0	PWM21 shared-pin inverse control 0: Non-inverse. 1: Inverse.
0	PW2NV0	R/W	0	PWM20 shared-pin inverse control 0: Non-inverse. 1: Inverse.

PW2O Registers (0xCB)

Bit	Field	Type	Initial	Description
7..6	Reserved	R	0	
5	PW2O5	R/W	0	PWM25 shared-pin frequency control 0: 1*Frequency PWM signal. 1: 1/2 *Frequency PWM signal.
4	PW2O4	R/W	0	PWM24 shared-pin frequency control 0: 1*Frequency PWM signal. 1: 1/2 *Frequency PWM signal.
3	PW2O3	R/W	0	PWM23 shared-pin frequency control 0: 1*Frequency PWM signal. 1: 1/2 *Frequency PWM signal.
2	PW2O2	R/W	0	PWM22 shared-pin frequency control 0: 1*Frequency PWM signal. 1: 1/2 *Frequency PWM signal.
1	PW2O1	R/W	0	PWM21 shared-pin frequency control 0: 1*Frequency PWM signal. 1: 1/2 *Frequency PWM signal.
0	PW2O0	R/W	0	PWM20 shared-pin frequency control 0: 1*Frequency PWM signal. 1: 1/2 *Frequency PWM signal.

PW2YH/PW2YL Registers (PW2YH: 0xB7, PW2YL: 0xB6)

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.

PW20DH/PW20DL Registers (PW20DH: 0xBB, PW20DL: 0xBA)

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

PW21DH/PW21DL Registers (PW21DH: 0xC2, PW21DL: 0xC1)

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

PW22DH/PW22DL Registers (PW22DH: 0xC4, PW22DL: 0xC3)

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

PW23DH/PW23DL Registers (PW23DH: 0xE5, PW23DL: 0xE4)

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

PW24DH/PW24DL Registers (PW24DH: 0xF6, PW24DL: 0xF5)

Bit	Field	Type	Initial	Description
7..0	PW24DH/L	R/W	0x00	16-bit PWM2 duty control.

PW25DH/PW25DL Registers (PW25DH: 0xFD, PW25DL: 0xFC)

Bit	Field	Type	Initial	Description
7..0	PW25DH/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 (0x9A)

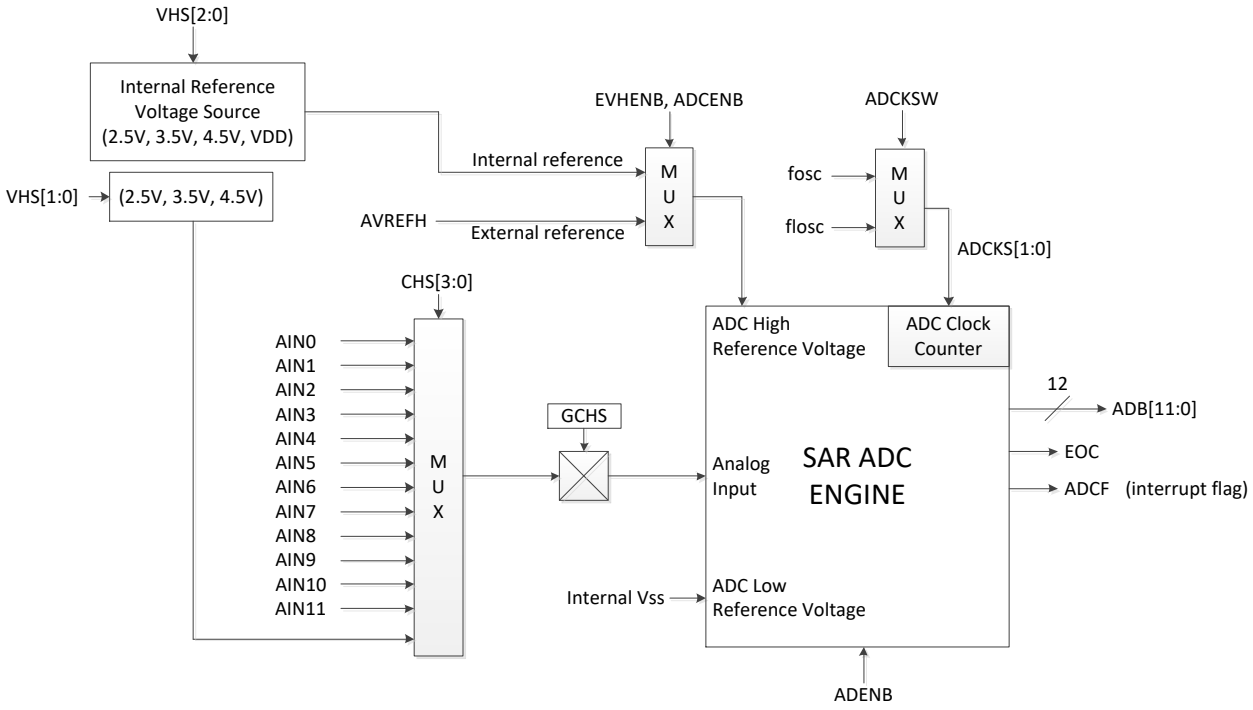
Bit	Field	Type	Initial	Description
3	EPW2	R/W	0	PWM2 interrupt control bit. 0 = Disable PWM2 interrupt function. 1 = Enable PWM2 interrupt function.
Else				Refer to other chapter(s)

IRCON2 Register (0xBF)

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)

16 ADC

The analog to digital converter (ADC) is SAR structure with 12-input sources and up to 4096-step resolution to transfer analog signal into 12-bits digital buffers. The ADC builds in 12-channel input source to measure 12 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 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 but inactive in stop mode. After ADC operating, the system would be waked up from idle mode to normal mode if interrupt enable.



16.1 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[3:0] 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.

16.2 ADC input channel

The ADC builds in 12-channel input source (AIN0 – AIN11) to measure 12 different analog signal sources controlled by CHS[3:0] and GCHS bits. The AIN12 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, not internal 2.5V or 3.5V or 4.5V. AIN12 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.

CHS[3:0]	Channel	Pin name	Remark
0000	AIN0	P1.0	-
0001	AIN1	P1.1	-
0010	AIN2	P1.2	-
0011	AIN3	P1.3	-
0100	AIN4	P1.4	-
0101	AIN5	P1.5	-
0110	AIN6	P0.5	-
0111	AIN7	P0.4	-
1000	AIN8	P0.3	-
1001	AIN9	P0.2	-
1010	AIN10	P0.1	-
1011	AIN11	P0.0	-
1100	AIN12	Internal 2.5V or 3.5V or 4.5V	Battery detector channel
Others	-	-	Reserved

16.2.1 Pin Configuration

ADC input channels are shared with Port0 and Port1. ADC channel selection is through CHS[3:0] bit. Only one pin of Port0 and Port1 can be configured as ADC input in the same time. The pins of Port0 and Port1 configured as ADC input channel must be set input mode, disable internal pull-up and enable P0CON/P1CON first by program. After selecting ADC input channel through CHS[3:0], 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 COMS 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 and Port1 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 must be set to “0”, or the digital I/O signal would be isolated.

16.3 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/P1.0. In the condition, P1.0 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[2:0] bits. If VHS2 is “1”, 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 AIN12 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, not internal 2.5V/3.5V/4.5V.

16.3.1 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 P1.0/AVREFH pin controlled by EVHENB bit. ADC reference voltage range limitation is “(ADC high reference voltage - low reference voltage) ≥ 2V”. ADC low reference voltage is Vss = 0V. So ADC high reference voltage range is 2V 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 = 2V 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

16.4 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 of 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.

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

When ADCKSW=0:

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

When ACKSW=1:

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

16.5 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 16-1 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

16.6 ADC Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADM	ADENB	ADS	EOC	-	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	-	-	P0CON5	P0CON4	P0CON3	P0CON2	P0CON1	P0CON0
P1CON	-	-	P1CON5	P1CON4	P1CON3	P1CON2	P1CON1	P1CON0
IEN0	EAL	-	-	-	ET1	-	ET0	EX0
IEN2	-	-	-	-	EPW2	EPW1	-	EADC
IRCON2	-	-	-	-	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)
4	Reserved	R	0	

3..0	CHS[3:0]	R/W	0x00	ADC input channel select bit. 0000: AIN0, 0001: AIN1, 0010: AIN2, 0011: AIN3, 0100: AIN4, 0101: AIN5, 0110: AIN6, 0111: AIN7, 1000: AIN8, 1001: AIN9, 1010: AIN10, 1011:AIN11, 1100: AIN12*(1), others: Reserved.

*(1) The AIN12 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 source 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/P1.0 pin is GPIO. 1: Disable ADC internal VREFH function. AVREFH/P1.0

				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: Reserved. 100: VREFH = VDD. (If CHS[3:0]=1100, AIN12 = 2.5V). 101: VREFH = VDD. (If CHS[3:0]=1100, AIN12 = 3.5V). 110: VREFH = VDD. (If CHS[3:0]=1100, AIN12 = 4.5V). 111: Reserved.

* (1) The AVREFH level must be between the VDD and 2V.

* (2) If AIN12 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 reference high voltage must be internal VDD, not internal 2.5V/3.5V/4.5V.

P0CON Register (0xD6)

Bit	Field	Type	Initial	Description
5..0	P0CON[5:0]	R/W	0x00	P0 configuration control bit*. 0: P0 can be analog input pin (ADC input pin) or digital GPIO pin. 1: P0 is pure analog input pin and can't be a digital GPIO pin.
Else	Reserved	R	0	

* P0CON [5:0] will configure related Port0 pin as pure analog input pin to avoid current leakage.

P1CON Register (0x9F)

Bit	Field	Type	Initial	Description
5..0	P1CON[5:0]	R/W	0x00	P1 configuration control bit*. 0: P1 can be analog input pin (ADC input pin) or digital GPIO pin. 1: P1 is pure analog input pin and can't be a digital GPIO pin.
Else	Reserved	R	0	

* P1CON [5:0] will configure related Port1 pin as pure analog input pin to avoid current leakage.

IEN2 Register (0x9A)

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 (0xBF)

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)

17 UART

The UART provides a flexible full-duplex synchronous/asynchronous receiver/transmitter. The serial interface provides an up to 0.25MHz 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.

17.1 UART Operation

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

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

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

UART 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.

Table 17-1 UART Operating Mode

SM0	SM1	Mode	Synchronization	Clock Rate	Start Bit	Data Bits	Stop Bit	UART pins' mode and data
0	0	0	Synchronous	Fcpu/12	X	8	X	UTX pin: P03M=1 and P03=1 P11M=1 and P11 =1 URX pin: Transmitter: P04M=1 and P04=1/ P12M=1 and P12=1 Receiver: P04M=0 and P04=1/ P12M=0 and P12=1
0	1	1	Asynchronous	Baud rate generator or T1 overflow rate	1	8	1	UTX/URX pin mode and data switch by hardware.
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	

17.2 Mode 0: Synchronous 8-bit Receiver/Transmitter

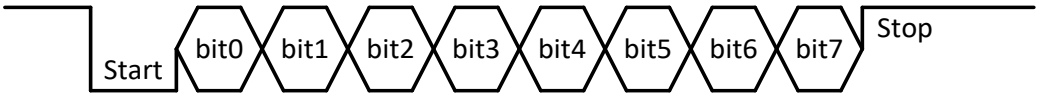
Mode0 is a shift register mode. It operates as synchronous transmitter/receiver. The UTX pin output shift clock for both transmit and receive condition. The URX pin is used to transmit and receive data. 8-bit data will be transmit and receive with LSB first. The baud rate is fcpu/12. Data transmission 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.

17.3 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 UTX pin and received by URX pin. The baud rate clock source can be baud rate generator or T1 overflow controlled by BD bit. When BD=0, the baud rate clock source is from T1 overflow. When BD=1, the baud rate clock source is from baud rate generator controlled by S0RELH and S0RELL. Additionally, the baud rate can be doubled by SMOD 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 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, URX 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.



17.4 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 UTX pin and received by URX pin. The baud rate clock source is fixed to fcpu/64 or fcpu/32 and is controlled by SMOD bit. When SMOD=0, baud rate is fcpu/64. When SMOD=1, baud rate is fcpu/32.

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, URX 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.

17.5 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 UTX pin and received by URX 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 BD bit. When BD=0, the baud rate clock source is from T1 overflow. When BD=1, the baud rate clock source is from baud rate generator controlled by S0RELH and S0RELL. Additionally, the baud rate can be doubled by SMOD 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, URX 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.

17.6 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 SM20 bit. When SM20=0, disable multiprocessor communication. When SM20=1, enable multiprocessor communication. If SM20 is set, the UART 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 SM20 bit to enable interrupt function in the following data transmission. The slavers with unmatched address, their SM20 keep in “1” and will not generate interrupt in the following data transmission.

17.7 Baud Rate Control

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

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

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

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

Table 17-2 Recommended Setting for Common UART Baud Rates (fcpu = 8 MHz)

Baud Rate	SMOD	S0RELH	S0RELL	Accuracy
4800	0	0x03	0xE6	0.16 %
9600	0	0x03	0xF3	0.16 %
19200	0	0x03	0xF3	0.16 %
38400	0	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 (BD = 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.

$$\text{Baud Rate} = 2^{\text{SMOD}} \times \frac{1}{32 \times \text{Timer 1 period}} \ bps$$

Table 17-3 Recommended Setting T1 overflow period (T1 clock=32M) for Common UART Baud Rates (fcpu = 8 MHz)

Baud Rate	SMOD	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 T1 overflow rate must be greater four times to system clock fcpu.

17.8 Power Saving

The UART module has clock gating function for saving power. When REN0/TEN0 bit is 0, the UART module internal clocks are halted to reduce power consumption. UART relevant register (S0CON, S0CON2, S0BUF, S0RELL, S0RELH and SMOD bit) are unable to access.

Conversely, when REN0/TEN0 bit is 1, UART internal clocks are run, and registers can access. The REN0/TEN0 bit must be set to 1, before the initial setting UART.

17.9 UART Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
S0CON	SM0	SM1	SM20	REN0	TB80	RB80	TI0	RI0
S0CON2	BD	-	-	TEN0	-	-	-	URMX
S0BUF	S0BUF7	S0BUF6	S0BUF5	S0BUF4	S0BUF3	S0BUF2	S0BUF1	S0BUF0
PCON	SMOD	-	-	-	-	GF0	STOP	IDLE
S0RELH	-	-	-	-	-	-	S0REL9	S0REL8
S0RELL	S0REL7	S0REL6	S0REL5	S0REL4	S0REL3	S0REL2	S0REL1	R0REL0
IEN0	EAL	-	-	-	ET1	-	ET0	EX0
IEN1	-	-	-	-	EURX	EUTX	ESPI	EI2C
P0OC	P14OC	P13OC	P12OC	P11OC	P04OC	P03OC	P01OC	P00OC
P0M	-	-	P05M	P04M	P03M	P02M	P01M	P00M
P1M	-	-	P15M	P14M	P13M	P12M	P11M	P10M
P0	-	-	P05	P04	P03	P02	P01	P00
P1	-	-	P15	P14	P13	P12	P11	P10

S0CON Register (0x98)

Bit	Field	Type	Initial	Description
7..6	SM[0:1]	R/W	00	UART mode selection 00: Mode 0 01: Mode 1 10: Mode 2 11: Mode 3
5	SM20	R/W	0	Multiprocessor communication (mode 2, 3) 0: Disable 1: Enable
4	REN0	R/W	0	UART reception function enable bit. 0: Disable for power saving*

				1: Enable for UART reception. URX pin enable.
3	TB0	R/W	0	The 9th bit transmission data (mode 2, 3)
2	RB0	R/W	0	The 9th bit data from reception
1	TI0	R/W	0	UART interrupt flag of transmission
0	RI0	R/W	0	UART interrupt flag of reception

* When REN0 and TEN0 bit both are 0, UART 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.***

S0CON2 Register (0x9B)

Bit	Field	Type	Initial	Description
7	BD	R/W	0	Baud rate generators selection (mode 1, 3) 0: Timer 1 overflow period 1: Controlled by S0RELH, S0RELL registers
4	TEN0	R/W	0	UART transmission function enable bit. 0: Disable for power saving* 1: Enable for UART transmission. UTX pin enable.
0	URMX	R/W	0	UART UTX/URX pin select bit 0: P03 (UTX) and P04 (URX) are selected. 1: P11 (UTX(1)) and P12 (URX(1)) are selected.
Else	Reserved	R	0x00	

* When REN0 and TEN0 bit both are 0, UART relevant register are unable to access, and the module internal clocks are halted.

S0BUF Register (0x99)

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

PCON Register (0x87)

Bit	Field	Type	Initial	Description
7	SMOD	R/W	0	UART baud rate control In UART mode 0: Unused. In UART mode 1, 3: The baud rate is generated as the equation in section 17.7 (Baud Rate Control).

				In UART mode 2: 0: fcpu/64 1: fcpu/32
6..0				Refer to other chapter(s)

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.
Else	Reserved	R	0	

IEN1 Register (0xB8)

Bit	Field	Type	Initial	Description
3	EURX	R/W	0	UART RX interrupt control bit. 0: Disable UART RX interrupt function. 1: Enable UART RX interrupt function.
2	EUTX	R/W	0	UART TX interrupt control bit. 0: Disable UART TX interrupt function. 1: Enable UART TX interrupt function.
Else	Reserved	R	0	

P0OC Register (0xE4)

Bit	Field	Type	Initial	Description
5	P12OC	R/W	0	P1.2 open-drain control bit. 0: Disable open-drain mode 1: Enable open-drain mode
4	P11OC	R/W	0	P1.1 open-drain control bit. 0: Disable open-drain mode 1: Enable open-drain mode
3	P04OC	R/W	0	P0.4 open-drain control bit. 0: Disable open-drain mode 1: Enable open-drain mode
2	P03OC	R/W	0	P0.3 open-drain control bit. 0: Disable open-drain mode 1: Enable open-drain mode
Else				Refer to other chapter(s)

P0M Register (0xF9)

Bit	Field	Type	Initial	Description
4	P04M	R/W	0	URX pin mode control bit. Refer to Table 17-1.* 0: Set P0.4 (URX) as input mode. 1: Set P0.4 (URX) as output mode.
3	P03M	R/W	0	UTX pin mode control bit. Refer to Table 17-1.* 0: Set P0.3 (UTX) as input mode. 1: Set P0.3 (UTX) as output mode.
Else				Refer to other chapter(s)

* The URX and UTX respectively require input and output mode selection to receive/transmit data appropriately in mode 0.

P1M Register (0xFA)

Bit	Field	Type	Initial	Description
2	P12M	R/W	0	URX pin mode control bit. Refer to Table 17-1.* 0: Set P1.2 (URX(1)) as input mode. 1: Set P1.2 (URX(1)) as output mode.
1	P11M	R/W	0	UTX pin mode control bit. Refer to Table 17-1.* 0: Set P1.1 (UTX(1)) as input mode. 1: Set P1.1 (UTX(1)) as output mode.
Else				Refer to other chapter(s)

* The URX and UTX respectively require input and output mode selection to receive/transmit data appropriately in mode 0.

P0 Register (0x80)

Bit	Field	Type	Initial	Description
4	P04	R/W	0	URX pin data control bit. Refer to Table 17-1.* 0: Set P0.4 (URX) is low. 1: Set P0.4 (URX) is High.
3	P03	R/W	0	UTX pin data control bit. Refer to Table 17-1.* 0: Set P0.3 (UTX) is low. 1: Set P0.3 (UTX) is High.
Else				Refer to other chapter(s)

* The URX and UTX respectively require input and output mode selection to receive/transmit data appropriately in mode 0.

P0 Register (0x90)

Bit	Field	Type	Initial	Description
2	P12	R/W	0	URX pin data control bit. Refer to Table 17-1.* 0: Set P1.2 (URX(1)) is low. 1: Set P1.2 (URX(1)) is High.

1	P11	R/W	0	UTX pin data control bit. Refer to Table 17-1.* 0: Set P1.1 (UTX(1)) is low. 1: Set P1.1 (UTX(1)) is High.
Else				Refer to other chapter(s)

* The URX and UTX respectively require input and output mode selection to receive/transmit data appropriately in mode 0.

18 SPI

The SPI a serial communicate interface for data exchanging from one MCU to one MCU or other hardware peripherals. It is a simple 8-bit interface without a major definition of protocol, packet or control bits. The SPI transceiver includes three pins, clock (SCK), data input and data output (MISO/ MOSI) to send data between master and slaver terminals. An optional slave select pin (SSN) can be enabled by register in slave mode. The SPI interface builds in 4-mode which are the clock idle status and the clock phases.

- Full-duplex, 3-wire synchronous data transfer.
- Master (SCK is clock output) or Slave (SCK is clock input) operation.
- Seven SPI Master baud rates.
- Slave Clock rate up to fcpu/8.
- 8-bit data transmitted MSB first, LSB last.
- Serial clock with programmable polarity and phase.
- Master Mode fault error flag with MCU interrupt capability.
- Write collision flag protection.

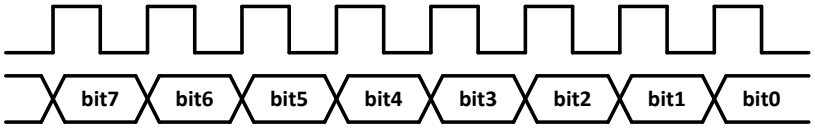
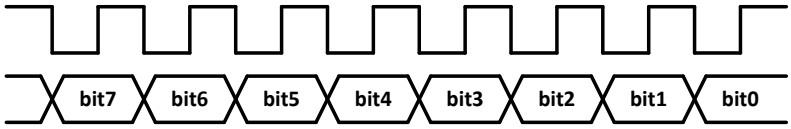
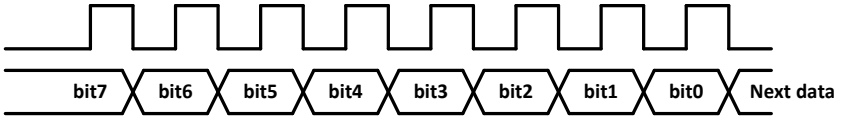
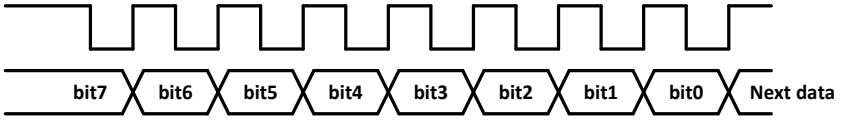
18.1 SPI Operation

The SPCON register can control SPI operating function, such as: transmit/receive, clock rate, data transfer direction, SPI clock idle status and clock control phase and enable this circuit. This SPI circuit will transmit or receive 8-bit data automatically by setting SPEN in SPCON register and write or read SPDAT register.

CPOL bit is designed to control SPI clock idle status. CPHA bit is designed to control the clock edge direction of data receive. CPOL and CPHA bits decide the SPI format. The SPI data transfer direction is MSB bit to LSB bit.

The SPI supports 4-mode format controlled by CPOL and CPHA bits. The edge direction is “Data Transfer Edge”. When setting rising edge that means to receive and transmit one bit data at SCK rising edge, and data transition is at SCK falling edge. When setting falling edge, that means to receive and transmit one bit data at SCK falling edge, and data transition is at SCK rising edge.

“CPHA” is the clock phase bit controls the phase of the clock on which data is sampled. When CPHA=1, the SCK first edge is for data transition, and receive and transmit data is at SCK 2nd edge. When CPHA=0, the 1st bit is fixed already, and the SCK first edge is to receive and transmit data. The SPI data transfer timing as following figure:

CPOL	CPHA	Diagrams	Description
0	1		SCK idle status = Low. The transfer first bit = MSB. SCK data transfer edge = Falling edge.
1	1		SCK idle status = High. The transfer first bit = MSB. SCK data transfer edge = Rising edge.
0	0		SCK idle status = Low. The transfer first bit = MSB. SCK data transfer edge = Rising edge.
1	0		SCK idle status = High. The transfer first bit = MSB. SCK data transfer edge = Falling edge.

The SPI supports interrupt function. ESPI is SPI interrupt function control bit. ESPI=0, disable SPI interrupt function. ESPI=1, enable SPI interrupt function. When SPI interrupt function enable, the program counter points to interrupt vector to do SPI interrupt service routine after SPI operating. SPIF is SPI interrupt request flag, and also to be the SPI operating status indicator when ESPI= 0, but cleared by reading the SPSTA, SPDAT registers.

SPI builds in chip selection function to implement SPI multi-device mode. One master communicating with several slave devices in SPI bus, and the chip selection decides the pointed device. The chip selection pin is SSN pin.

SPI builds in SPIMX bit to select SPI pins. SPIMX=0, SCK(P00)/ MOSI(P01)/ MISO(P03)/ SSN(P04) are selected. SPIMX=1, SCK(1)(P12)/ MOSI(1) (P13)/ MISO(1) (P14)/ SSN(1) (P15) are selected. When SPI function is enabled, SCK/ MOSI/ SSN share pin switch to communication mode by hardware.

The SPI pins also support open-drain structure. The open-drain option is controlled by PnOC bits. When PnOC=0, disable SPI open-drain structure. When PnOC=1, enable SPI open-drain structure. If enable open-drain structure, SPI pins must be set input mode and need external pull-up resistor.

18.2 SPI Master

The SPI master mode has seven types of clock generator from fcpu/2 to fcpu/128. Generated clock is outputted through SCK pin (shared with P0.0/P1.2) and its idle status is controlled by CPOL.

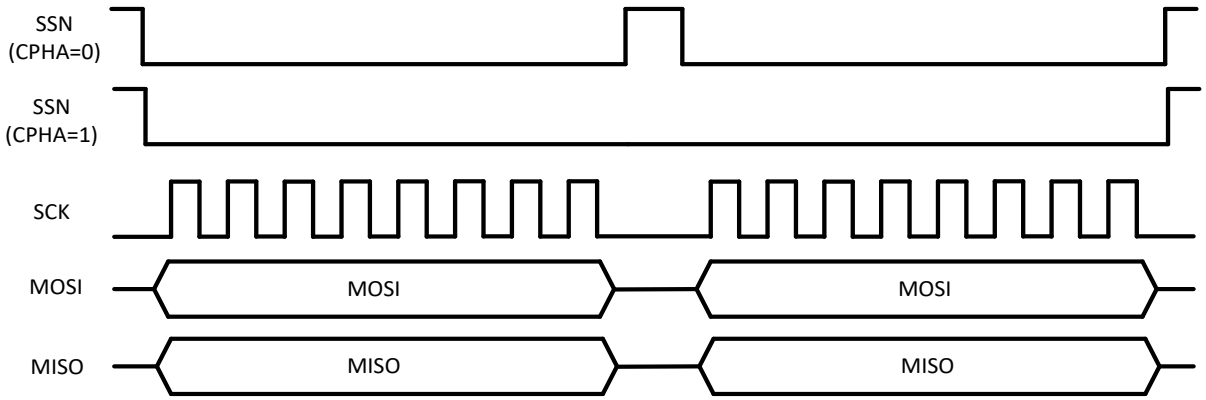
The phase of data input and output is automatically specified by CPHA register. In master mode MOSI pin (shared with P0.1/P1.3) plays the role of data output, and MISO pin (shared with P0.3/P1.4) fetches data from slave device. A SPI communication is started by writing SPDAT register; the received data from MISO is available to read after the end of data transmission.

The master mode has two status flags with interrupt function:
SPIF register indicates the end of one byte data communication. An interrupt would be issued at the same time if ESPI bit is enabled.
MODF is issued by SSN (shared with P0.4/P1.5) low status while transmission. This interrupt source can be masked by setting SSDIS bit.

18.3 SPI Slave

The SPI slave mode monitors SCK pin to control its MISO and MOSI communication. However, the maximum clock rate is limited at fcpu/8. Slave device(s) are expected to specify its CPOL and CPHA setting as the same configuration of the connected SPI bus.

The slave mode treats MOSI pin as its data input, and MISO pin as its data transmission. By default, the SSDIS register is low which means the slave select pin (SSN) is functional. A SPI communication would be processed if the SSN is low status. Thus, a slave device is suspended if its SSN is high status. But in CPHA = 0, Strictly SSN must follow each 8-bit data needs to be included with falling edge and rising edge, CPHA=1 is not limitation.



The slave mode has two status flags with interrupt function:
SPIF indicates the end of one byte data communication. The original SPDAT's value has been transmitted, and the received data from MOSI is ready to be read on SPDAT.
MODF indicates that the slave select pin (SSN) has turned high before a completion of one byte communication. In other word, the last time of SPI communication is broken.

18.4 Power Saving

The SPI module has clock gating function for saving power. When SPEN bit is 0, the SPI module internal clocks are halted to reduce power consumption. SPI relevant register (SPCON, SPSTA and SPDAT) are unable to access. Conversely, when SPEN bit is 1, SPI internal clocks are run, and registers can access. The SPEN bit must be set to 1, before the initial setting SPI.

18.5 SPI Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPCON	SPR2	SPEN	SSDIS	MATR	CPOL	CPHA	SPR1	SPR0
SPSTA	SPIF	WCOL	SSERR	MODF	-	-	-	SPIMX
SPDAT	SPDAT7	SPDAT6	SPDAT5	SPDAT4	SPDAT3	SPDAT2	SPDAT1	SPDAT0
IEN0	EAL	-	-	-	ET1	-	ET0	EX0
IEN1	-	-	-	-	EURX	EUTX	ESPI	EI2C
P0OC	P14OC	P13OC	P12OC	P11OC	P04OC	P03OC	P01OC	P00OC
P0M	-	-	P05M	P04M	P03M	P02M	P01M	P00M
P1M	-	-	P15M	P14M	P13M	P12M	P11M	P10M

SPCON Register (0xE2)

Bit	Field	Type	Initial	Description
7,1,0	SM[2:0]	R/W	000	SPI baud rate generator (master mode only) 000: fcpu/2 001: fcpu/4 010: fcpu/8 011: fcpu/16 100: fcpu/32 101: fcpu/64 110: fcpu/128 111: reserved
6	SPEN	R/W	0	SPI communication function 0: Disable for power saving* 1: Enable for SPI operating

5	SSDIS	R/W	0	Slave select pin function (MSTR = 0, CPHA = 0 only) 0: Enable slave selection pin (SSN) function 1: Disable slave select pin (SSN) function
4	MSTR	R/W	1	SPI mode 0: Slave mode 1: Master mode
3	CPOL	R/W	0	SCK pin idle status 0: SCK idle low 1: SCK idle high
2	CPHA	R/W	1	Clock phase of data latch control 0: Data latched by the first of clock edge 1: Data latched by the second of clock edge

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

SPSTA Register (0xE1)

Bit	Field	Type	Initial	Description
7	SPIF	R	0	SPI complete communication flag Set automatically at the end of communication Cleared automatically by reading SPSTA, SPDAT registers
6	WCOL	R	0	Write collision flag Set automatically if write SPDAT during communication Cleared automatically by reading SPSTA, SPDAT registers
5	SSERR	R	0	Synchronous slave select pin error Set automatically if SSN error controlling Cleared automatically by clear SPEN
4	MODF	R	0	Mode fault flag
3..1	Reserved	R	0x00	
0	SPIMX	R/W	0	SPI pin select bit 0: SPI pin are P00(SCK), P01(MOSI), P03(MISO) and P04(SSN) 1: SPI pin are P12(SCK(1)), P13(MOSI(1)), P14(MISO(1)) and P15(SSN(1))

SPDAT Register (0xE3)

Bit	Field	Type	Initial	Description
7..0	SPDAT	R/W	0x00	Master mode: action of writing data triggers SPI communication; reception data is readable after the end of one byte communication (SPIF automatically set).

				Slave mode: written data would be transmitted by SCK input; reception data is available to read after the end of one byte communication (SPIF automatically set).

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.
Else	Reserved	R	0	

IEN1 Register (0xB8)

Bit	Field	Type	Initial	Description
1	ESPI	R/W	0	SPI interrupt control bit. 0: Disable SPI interrupt function. 1: Enable SPI interrupt function.
Else	Reserved	R	0	

P0OC Register (0xE4)

Bit	Field	Type	Initial	Description
7	P14OC	R/W	0	P1.4 open-drain control bit. 0: Disable open-drain mode 1: Enable open-drain mode
6	P13OC	R/W	0	P1.3 open-drain control bit. 0: Disable open-drain mode 1: Enable open-drain mode
5	P12OC	R/W	0	P1.2 open-drain control bit. 0: Disable open-drain mode 1: Enable open-drain mode
2	P03OC	R/W	0	P0.3 open-drain control bit. 0: Disable open-drain mode 1: Enable open-drain mode
1	P01OC	R/W	0	P0.1 open-drain control bit. 0: Disable open-drain mode 1: Enable open-drain mode
0	P00OC	R/W	0	P0.0 open-drain control bit. 0: Disable open-drain mode 1: Enable open-drain mode

Else	Refer to other chapter(s)

P0M Register (0xF9)

Bit	Field	Type	Initial	Description
4	P04M	R/W	0	0: Set P0.4 (SSN) as input mode slave mode 1: Set P0.4 (SSN) as output mode master mode
Else				Refer to other chapter(s)

*If slave mode with SSN function: essentially to set SSN as input mode.

P1M Register (0xFA)

Bit	Field	Type	Initial	Description
5	P15M	R/W	0	0: Set P1.5 (SSN) as input mode slave mode 1: Set P1.5 (SSN) as output mode master mode
Else				Refer to other chapter(s)

*If slave mode with SSN function: essentially to set SSN as input mode.

19 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.

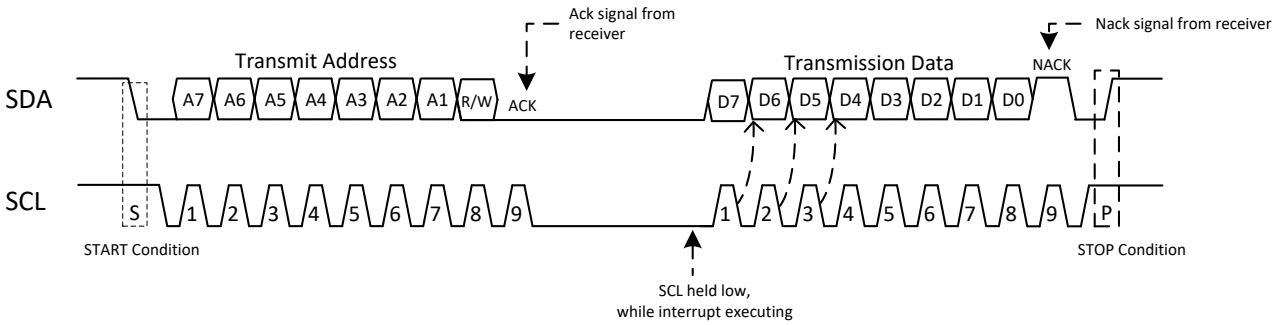
- Master Tx, Rx Mode
- Slave Tx, Rx mode (with general address call) for multiplex slave in single master situation.
- 2-wire synchronous data transfer/receiver.
- Support 100K/400K clock rate.

19.1 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 "WRITR" 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.

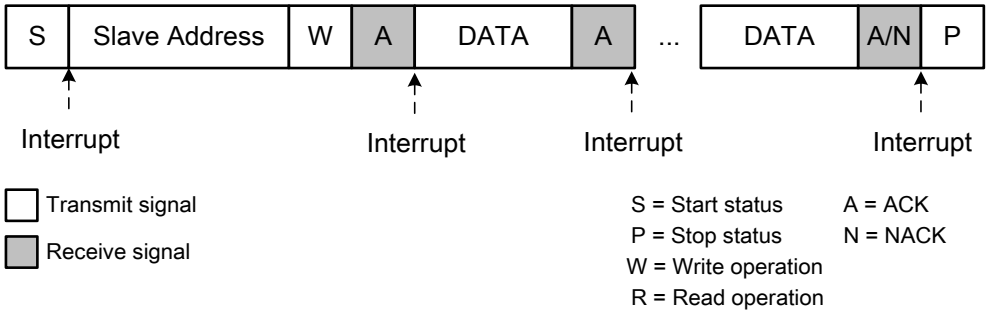


19.2 I2C Transfer Modes

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.

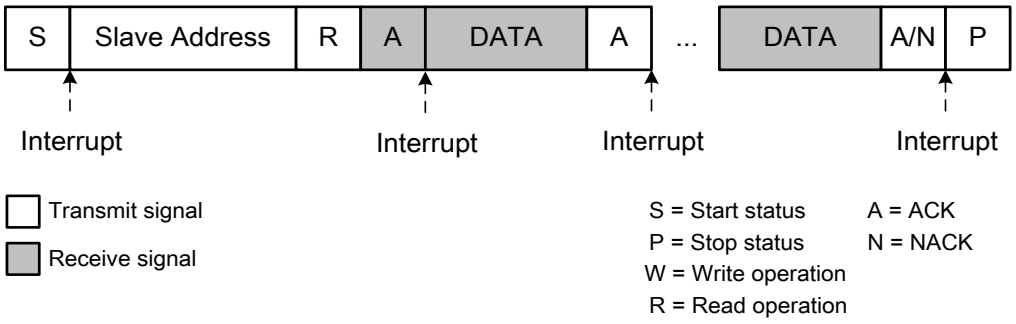
19.2.1 Master Transmitter Mode

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.



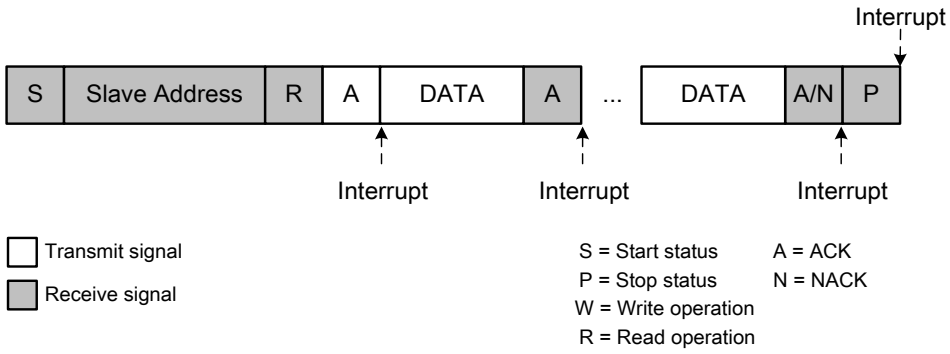
19.2.2 Master Receiver Mode

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.



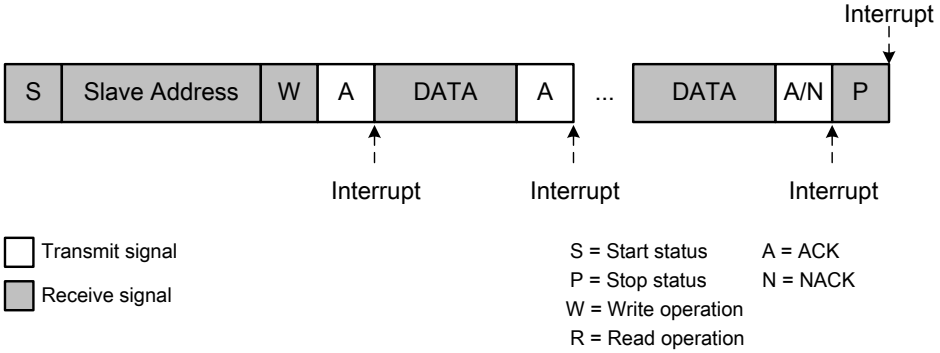
19.2.3 Slave Transmitter Mode

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.



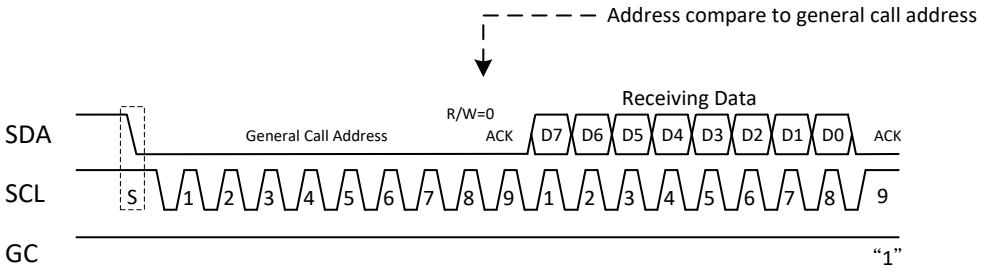
19.2.4 Slave Receiver Mode

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.



19.3 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.



19.4 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.

$$\text{SCL Clock Rate} = \frac{\text{Fhosc}}{\text{Prescaler}} (\text{Prescaler} = 320 \sim 40)$$

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	I2C Prescaler	Bit Frequency (kHz) Fhosc = 32MHz
0	0	0	40	-
0	0	1	80	400
0	1	0	160	200
0	1	1	320	100
1	X	X	(Timer 1 overflow rate)/8	

✷ ***Note:***

1. When I2C function is enabled, SDA/SCL pins switch to I2C pins' mode by hardware.

2. When clock generator source is T1 overflow rate, the max counter value is 0xF9. (Only supports 0x00~0xF9).

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

19.5 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 anther 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 anther 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 anther 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.

19.6 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.

19.7 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.

19.8 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	-	-	-	ET1	-	ET0	EX0
IEN1	-	-	-	-	EURX	EUTX	ESPI	EI2C
P0M	-	-	P05M	P04M	P03M	P02M	P01M	P00M
P1M	-	-	P15M	P14M	P13M	P12M	P11M	P10M

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/W	X	0	0	X	SLA+R/W will be transmitted; ACK will be received
			Load SLA+R/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 transmitted; 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.
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: P0.0 (SCL) and P0.1 (SDA) are selected. 1: P1.3 (SCL(1)) and P1.4 (SDA(1)) are selected.
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)

P0M Register (0xF9)

Bit	Field	Type	Initial	Description
1	P01M	R/W	0	0: Set P0.1 (SDA) as input mode (required) 1: Set P0.1 (SDA) as output mode*
0	P00M	R/W	0	0: Set P0.0 (SCL) as input mode (required) 1: Set P0.0 (SCL) as output mode*
Else				Refer to other chapter(s)

* The P00M and P01M require be set input mode.

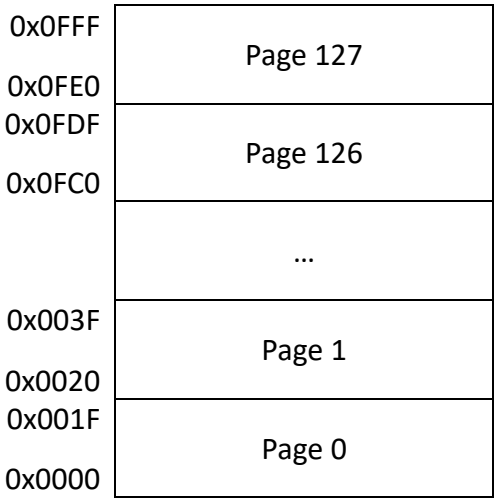
P1M Register (0xFA)

Bit	Field	Type	Initial	Description
4	P14M	R/W	0	0: Set P1.4 (SDA) as input mode (required) 1: Set P1.4 (SDA) as output mode*
3	P13M	R/W	0	0: Set P1.3 (SCL) as input mode (required) 1: Set P1.3 (SCL) as output mode*
Else				Refer to other chapter(s)

* The P13M and P14M require be set input mode.

20 In-System Program

SN8F5721 builds in an on-chip 4 KB program memory, aka IROM, which is equally divided to 128 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)

20.1 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~bit4 (hex) =0
ROM address bit5~bit15 (hex)	0000	These pages include reset vector and interrupt sector. We strongly recommend to reserve the area not to do ISP erase.
	0020	
	0040	
	…	
	00C0	
	00E0	
	0100	One ISP Program Page
	0120	One ISP Program Page
	…	One ISP Program Page
	0300	One ISP Program Page
	0320	One ISP Program Page
	…	One ISP Program Page
	0700	One ISP Program Page
	0720	One ISP Program Page
	…	One ISP Program Page
	0FE0	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 [15:5]. (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 126th page of program memory (IROM, 0x0FC0 – 0x0FDF) is the anticipated update area; the content is already stored in IRAM address 0x60 – 0x7F. To perform the in-system program, simply write starting IROM address 0x0FC0 to EPROMH/EPROML registers, and then specify buffer starting address 0x60 to EPRAM register. Subsequently, write ‘0x5A’ into PECMD [7:0] registers to duplicate the buffer’s data to 126th 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 127) 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.

20.2 Byte Program

Byte program supports one byte memory program, one byte program procedure requires 1 byte IRAM as its data buffer.

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

1. Save program data into IRAM. The data only for 1 byte.
2. Set the start address of the content location to PERAM.
3. Set the start address of the anticipated update area to PEROM [15:0]. (By PEROMH/PRROML registers)
4. Write ‘0x1E’ into PECMD [7:0] to trigger ISP function.
5. Write ‘NOP’ instruction twice.

As an example, assume the address 0x0FC5 of IRPM is the anticipated update area; the content is already stored in IRAM address 0x60. To perform the in-system byte program, simply write starting IROM address 0x0FC5 to EPROMH/EPROML registers, and then specify buffer starting address 0x60 to EPRAM register. Subsequently, write ‘0x1E’ into PECMD [7:0] registers to duplicate the buffer’s data to the address 0x0FC5 of IROM.

✷ ***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. We strongly recommend not performing ISP function during power on and off, this is risky and recommended to execute the ISP function when the power supply is stable.

4. Software protection is recommended to prevent data was written incorrectly.

5. 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.

20.3 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 (0x97)

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

PEROMH Register (0x96)

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 (0x95)

Bit	Field	Type	Initial	Description

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

PECMD Register (0x94)

Bit	Field	Type	Initial	Description
7..0	PECMD[7:0]	W	0x0	0x5A: Start page program procedure 0x1E: Start byte program procedure Else values: Reserved*(1)

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

21 Clock Fine-Tuning

SN8F5721 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.

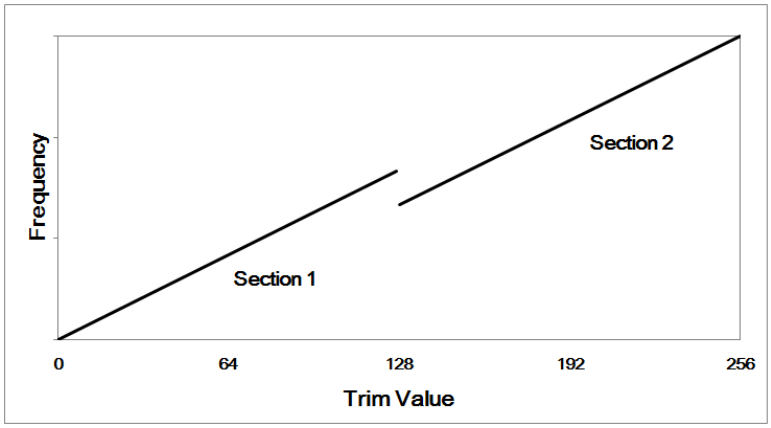
21.1 Clock Trim Section

Clock fine-tuning consists of 8 trim sections as Table 21-1. 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 21-1 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.



21.2 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 8MHz, 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 Table21-1.
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 ≦ 8MHz to avoid system error.***

21.3 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: 0xCE, FRQL: 0xCD)

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

FRQCMD Register (0xC5)

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

22 ILRC Auto-Calibration

22.1 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.

22.2 Auto-Calibration Register

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LFRQH	-	-	-	-	-	-	-	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: 0xF4, FRQL: 0xF3)

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

FRQCMD Register (0xC5)

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

CLKCAL Register (0xF8)

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

23 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.

23.1 Debug Mode

System debug mode is default enabled. When system power on, SWAT function is enabled and P1.1 acts as SWAT pin for debug interface. User can disable SWAT function by DEGCMD register in order to use P1.1 as GPIO. When DEGCMD= 0x00, SWAT function is disabled and P1.1 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.

23.2 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.

23.3 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 (0xCF)

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

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

24 Electrical Characteristics

24.1 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

24.2 System Operation Characteristics

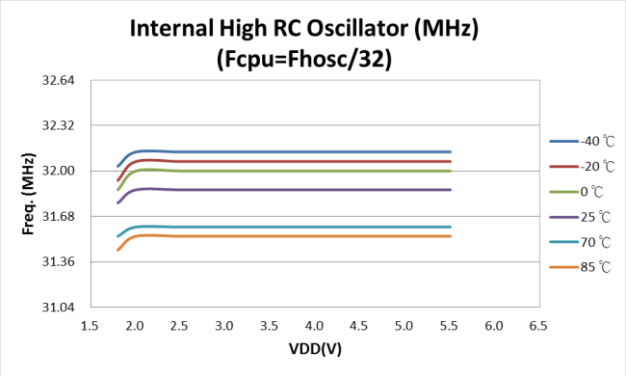
	Parameter	Test Condition	Min	TYP	MAX	UNIT
VDD	Operating voltage	fcpu = 1MHz	1.8		5.5	V
V_{DR}	RAM data retention Voltage		0.55			V
V_{POR}	VDD rising rate*		0.05			V/ms
I_{DD1}	Normal mode supply current	VDD = 3V, fcpu = 1MHz		2.35		mA
		VDD = 5V, fcpu = 1MHz		2.40		mA
		VDD = 3V, fcpu = 4MHz		2.70		mA
		VDD = 5V, fcpu = 4MHz		2.80		mA
		VDD = 3V, fcpu = 8MHz		3.25		mA
		VDD = 5V, fcpu = 8MHz		3.30		mA
I_{DD2}	STOP mode supply current	VDD = 3V		1.4	3.8	µA
		VDD = 5V		1.6	4.0	µA
I_{DD3}	STOP mode supply current (ILRC enable STWK=1) *	VDD = 5V		2.2	4.6	µA
I_{DD4}	IDLE mode supply current (fcpu = 1MHz)	VDD = 3V, 32MHz IHRC		0.71		mA
		VDD = 5V, 32MHz IHRC		0.73		mA
F_{IHRC}	Internal high clock generator	VDD = 1.8V to 5.5V, 25°C	-0.5%	32	+0.5%	MHz
		VDD = 1.8V to 5.5V, 25°C Life Test in OLT Stress	-1.0%	32	+1.0%	MHz
		VDD = 1.8V to 5.5V, -40°C to 85°C	*-2.0%	32	+2.0%	MHz
F_{ILRC}	Internal low clock generator*	VDD = 1.8V to 5.5V, 25°C Non-trimmed	-25%	16	+75%	KHz
		VDD = 1.8V to 5.5V, 25°C Trimmed	-3.0%	16	+3.0%	KHz
V_{LVD18}	LVD18 detect voltage	25°C	1.6	1.7	1.8	V
		-40°C to 85°C	1.5	1.7	1.9	V

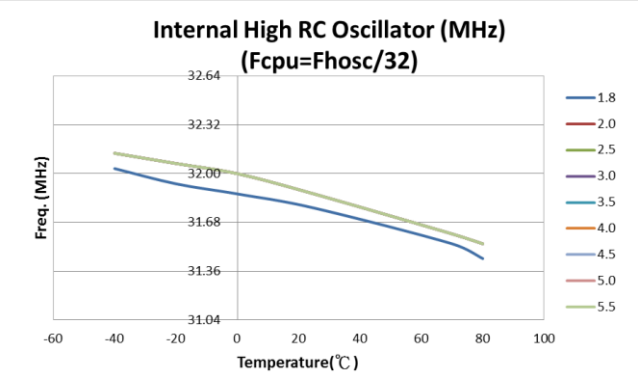
V_{ESD_HBM}	ESD human body mode	4000	V
V_{ESD_MM}	ESD machine mode	350	V

* 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.





24.3 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_{OH}	I/O output source current	VDD = 5V, V_O = VDD-0.5V	12	15		mA
I_{OL}	I/O sink current	VDD = 5V, V_O = VSS+0.5V	15	19		mA

* Ambient temperature is 25°C.

24.4 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		VDD	V
V_{IREF}	Internal VDD reference voltage	VDD = 5V		VDD		V
	Internal 4.5V reference voltage	VDD = 5V	-1%	4.5	+1%	V
	Internal 3.5V reference voltage	VDD = 5V	-1%	3.5	+1%	V
	Internal 2.5V reference voltage	VDD = 5V	-0.5%	2.5	+0.5%	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	-5	0	5	mV

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

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

24.5 Flash Memory Characteristics

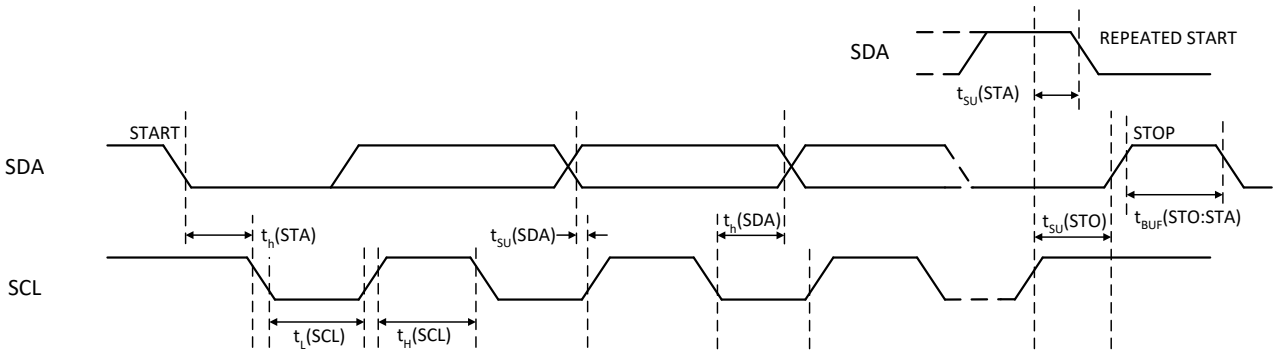
	Parameter	Test Condition	MIN	TYP	MAX	UNIT
V_{dd}	Supply voltage		1.8		5.5	V
T_{pen}	Page Endurance time	25°C		*100K		cycle
T_{ben}	Byte endurance time	25°C		*10K		cycle
I_{wrt}	Write current	25°C		3	4	mA
T_{wrt}	Write time	Write 1 page=32 bytes, 25°C		6	8	ms

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

24.6 I2C interface characteristics

Symbol	Parameter	Typical Formula	UNIT
t_H(SCL)	SCL clock high time	2* fsck	µs
t_L(SCL)	SCL clock low time	2* fsck	µs
t_{SU}(SDA)	SDA setup time	2* fsck -7*fhosc	µs
t_h(SDA)	SDA data hold time	7*fhosc	µs
t_h(STA)	START condition hold time	4* fsck	µs
t_{SU}(STA)	Repeated START condition setup time	2* fsck	µs
t_{SU}(STO)	STOP condition setup time	4* fsck +8*fhosc	µs
T_{BUF}(STO:STA)	STOP to START condition time	4/3* fsck +10*fhosc	µs

*Standard mode I2C fsck = 100KHz, Fast mode I2C fsck = 400KHz, fhosc=32MHz



25 Instruction Set

This chapter categorizes the SN8F5721 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

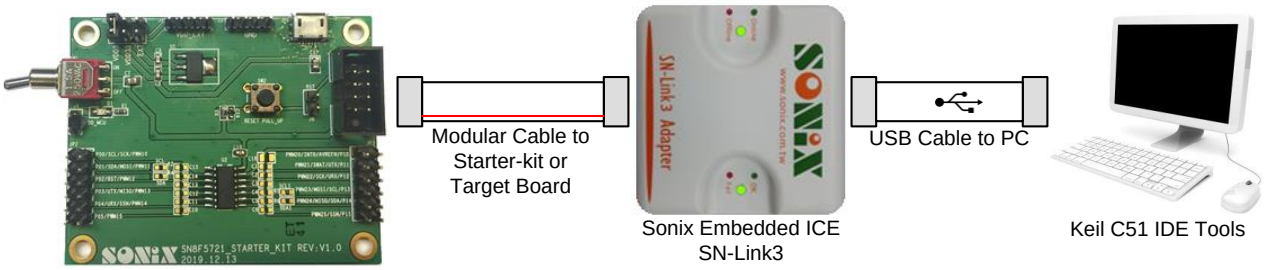
Mnemonic	Description
CLR C	Clear carry flag
CLR bit	Clear directly addressed bit
SETB C	Set carry flag
SETB bit	Set directly addressed bit
CPL C	Complement carry flag
CPL bit	Complement directly addressed bit
ANL C, bit	AND directly addressed bit to carry flag
ANL C, /bit	AND complement of directly addressed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

26 Development Environment

SONIX provides an Embedded ICE emulator system to offer SN8F5721 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, SN8F5721 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 SN8F5721 to offer a real development environment.



26.1 Minimum Requirement

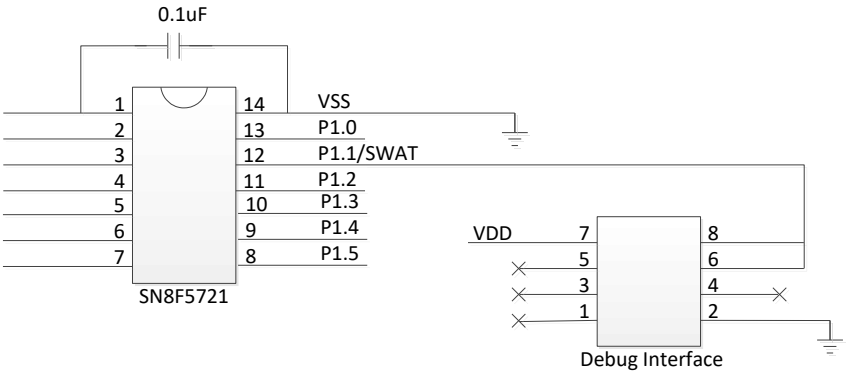
The following items are essential to build up an appropriate development environment. The compatibility is verified on listed versions, and is expected to execute perfectly on later version. SN-Link related information is available to download on SONiX website (www.sonix.com.tw);
Keil C51 is downloadable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

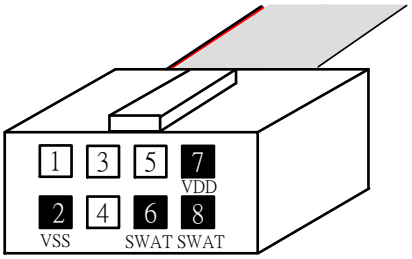
26.2 Debug Interface Hardware

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

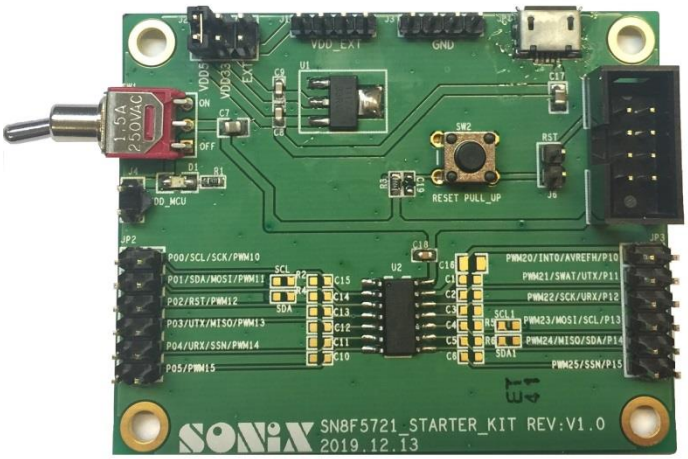


SN-Link header

26.3 Development Tool

SN-Link3 Adapter

Starter-Kit support SN8F5721-SN8F57213



MP5 Writer

27 SN8F5721 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.

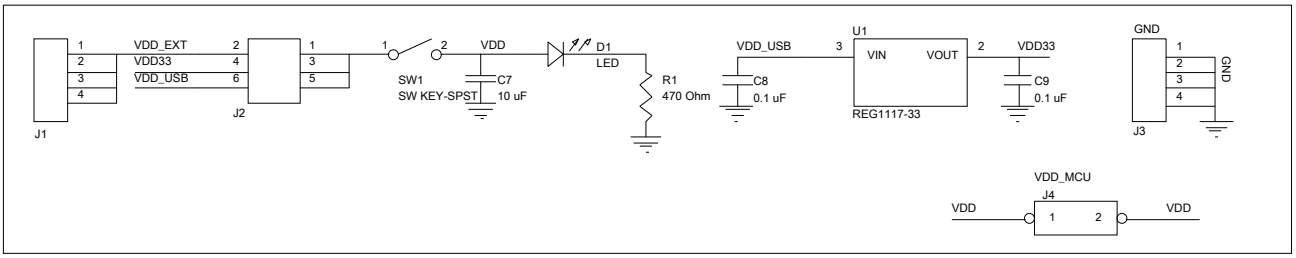
27.1 Configurations of Circuit

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

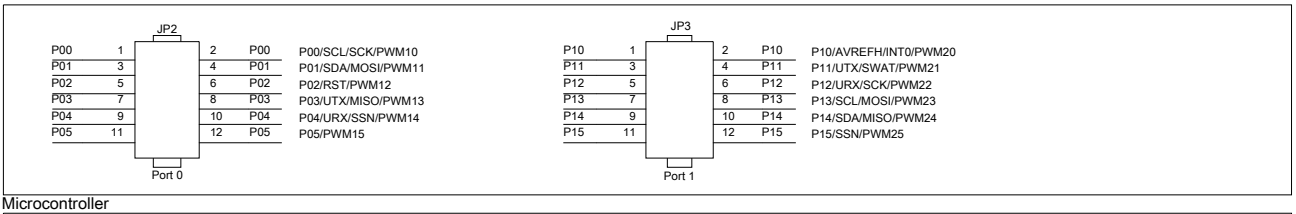
1. Confirm to the circuit board whether elements are complete.
2. The power source of Starter-Kit circuit is chosen from external power or Micro USB via jumper.
3. The power source comes from 5.0V or 3.3V which must be connect to Micro USB.
4. If the power source is chosen from external power, then external power source connects to EXT pin.
5. The "RST" pin needs to connect pull high resister to VDD when external reset is chosen to use.
6. The Debug Port can connect SN-LINK Adapter for emulation or download code.
7. The MCU LED will light up and SN8F5000 family chip will be connected to power when power (VDD) is switched on.

27.2 Schematic

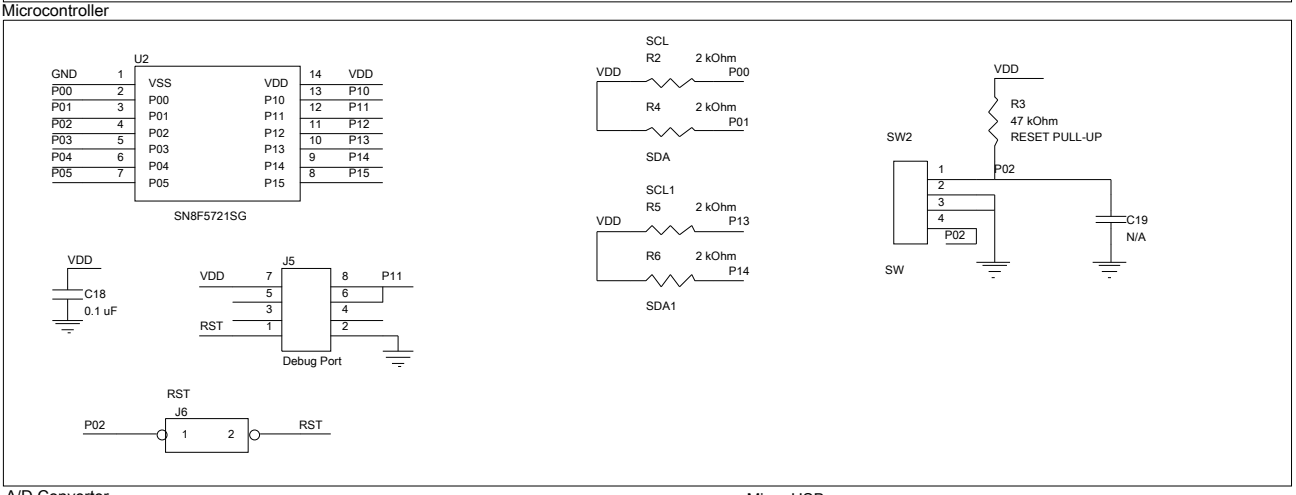
Power Source



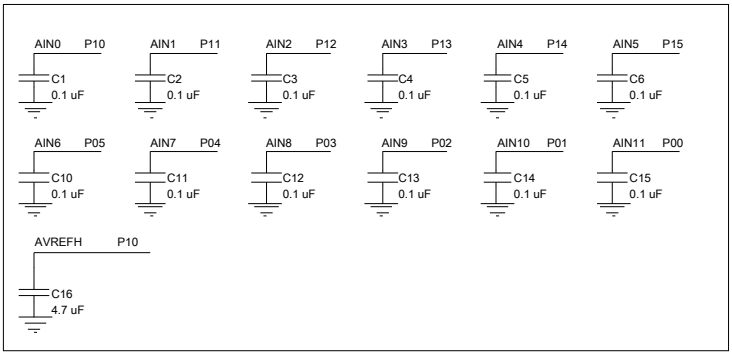
GPIO Interface



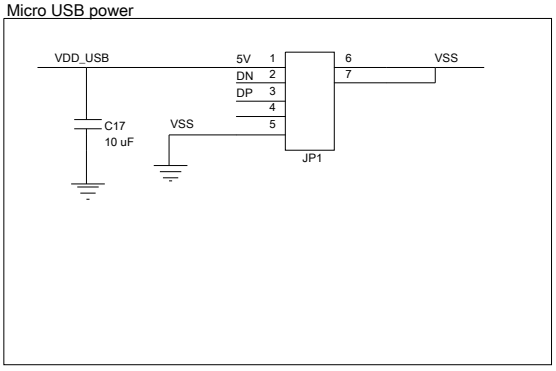
Microcontroller



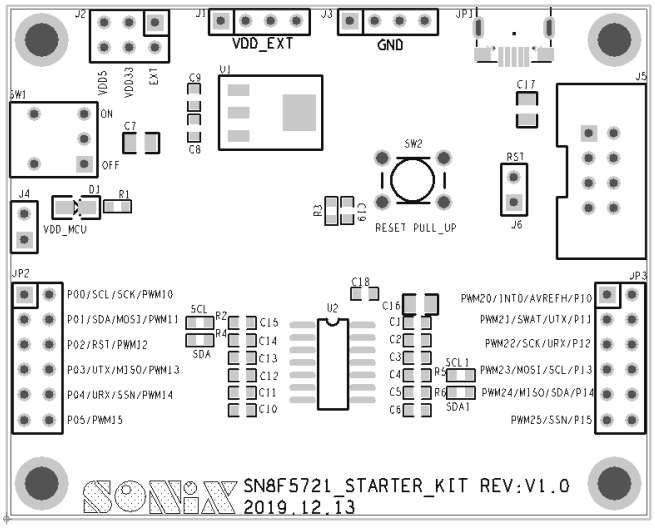
A/D Converter



Micro USB power



27.3 Floor Plan of PCB layout



27.4 Component Description

Number	Description
C1 – C6, C10-C15	12-ch ADC capacitors.
C16	AVREFH capacitor.
D1	MCU LED
J1/J3	External power source.
SW2	External reset trigger source
J2	VDD power source is 5.0V, 3.3V or external power.
J5	Debug Port
JP2/JP3	I/O connector.
R3, C19	External reset pull-high resister and capacitor.
R2, R4, R5, R6	I2C pull-high resisters.
SW1	Target power (VDD) switch
U2	SN8F5721S real chip (Sonix standard option).
JP1	Micro USB port

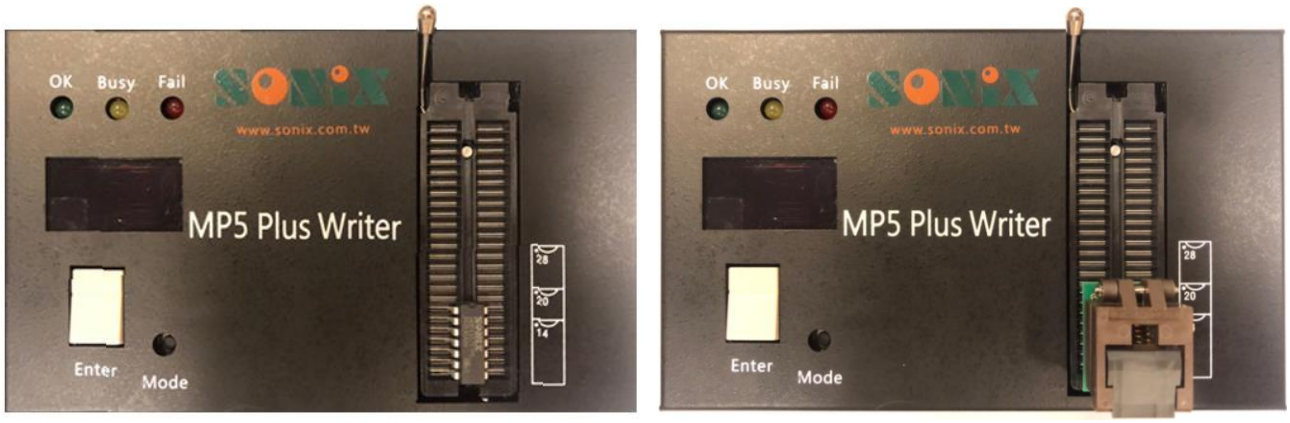
28 ROM Programming Pin

SN8F5721 Series Flash ROM erase/program/verify support SN-Link and MP5 Writer

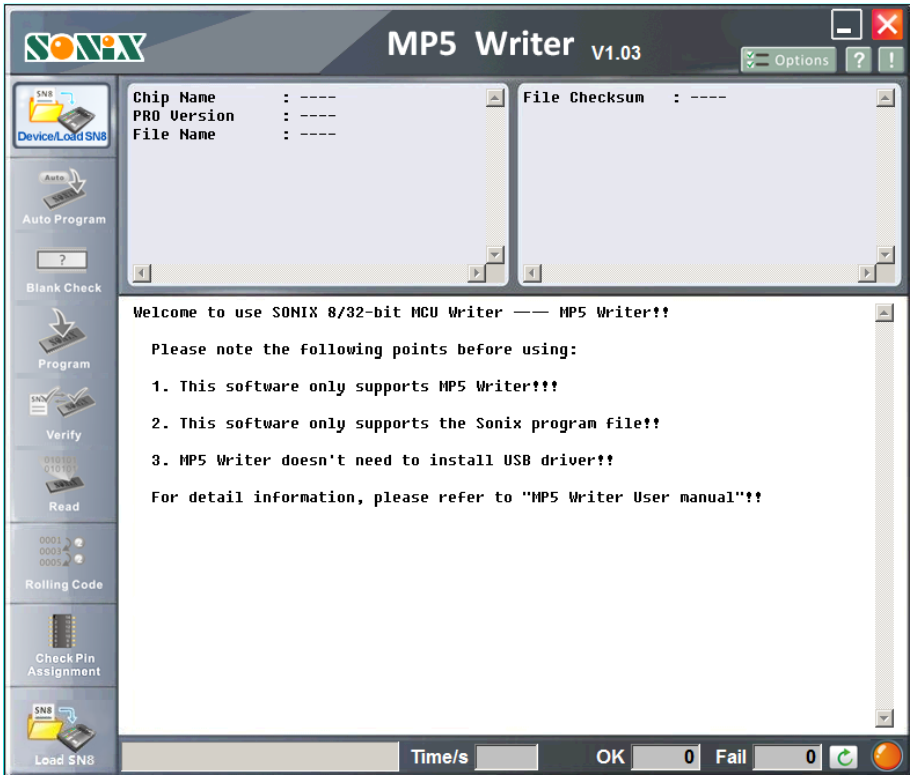
- SN-Link: Debug interface and on board programming.
- MP5 Writer: For SN8F5721 series version mass programming.

28.1 MP5 Hardware Connecting

Different package type with MCU programming connecting is as following, DIP and SOP Illustration.

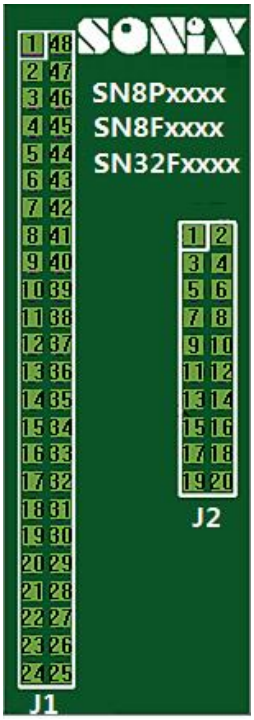


MP5 Software operation interface is as following.



28.2 MP5 Writer Transition Board Socket Pin Assignment

MP5 Writer Transition Board:



28.3 MP5 Writer Programming Pin Mapping

Writer Connector		MCU Pin Number	SN8F5721P/S,57217T		SN8F57211A		SN8F57212A		SN8F57213S	
J2 Pin Number	J2 Pin Name		MCU Pin Number	J1 Pin Number	MCU Pin Number	J1 Pin Number	MCU Pin Number	J1 Pin Number	MCU Pin Number	J1 Pin Number
1	VDD	VDD	14	31	10	29	1	20	8	28
2	GND	VSS	1	18	1	20	2	21	1	21
7, 9	SWAT	P1.1	12	29	9	28	9	28	6	26
20	PDB	P1.2	11	28	8	27	8	27	5	25

Writer Connector		MCU Pin Number	SN8F5721J		SN8F57214S		SN8F57215S		SN8F57216S	
J2 Pin Number	J2 Pin Name		MCU Pin Number	J1 Pin Number	MCU Pin Number	J1 Pin Number	MCU Pin Number	J1 Pin Number	MCU Pin Number	J1 Pin Number
1	VDD	VDD	15	31	1	18	1	21	4	21
2	GND	VSS	16	32	14	31	8	28	11	28
7, 9	SWAT	P1.1	13	29	12	29	7	27	2	19
20	PDB	P1.2	12	28	11	28	6	26	1	18

28.4 SN-Link ISP Programming

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

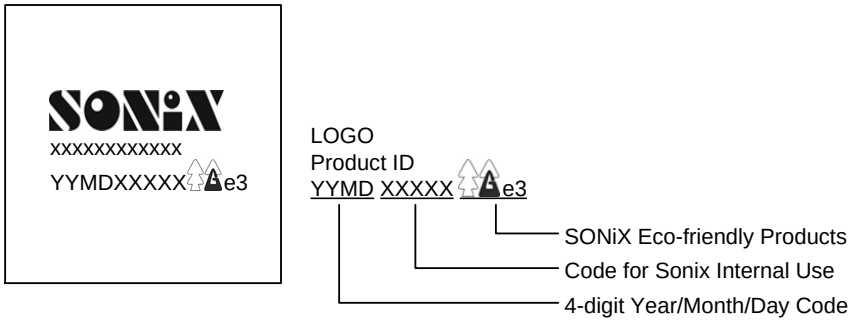
28.5 SN-Link ISP Programming Pin Mapping

SN-Link Connector		MCU Pin Number	SN8F5721P/S	SN8F57211A	SN8F57212A	SN8F57213S	SN8F57214S
Pin Number	Pin Name		Pin Number	Pin Number	Pin Number	Pin Number	Pin Number
7	VDD	VDD	14	10	1	8	1
2	GND	VSS	1	1	2	1	14
6, 8	SWAT	P1.1	12	9	9	6	12

SN-Link Connector		MCU Pin Number	SN8F5721J	SN8F57215S	SN8F57216S	SN8F57217T
Pin Number	Pin Name		Pin Number	Pin Number	Pin Number	Pin Number
7	VDD	VDD	15	1	4	14
2	GND	VSS	16	8	11	1
6, 8	SWAT	P1.1	13	7	2	12

29 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.

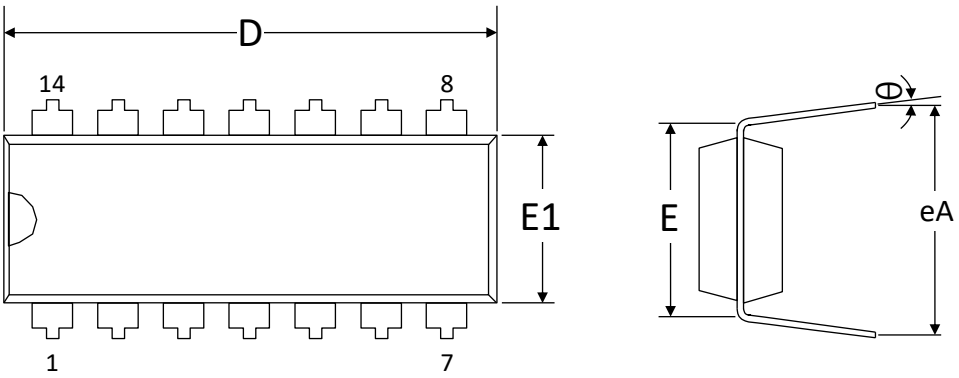


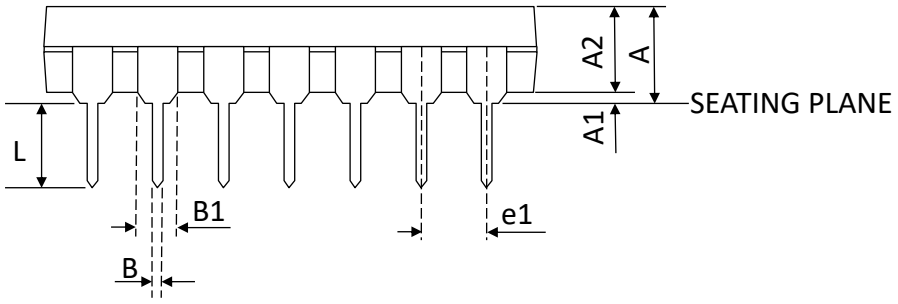
29.1 Device Nomenclature

Full Name	Packing Type
SN8F5721W	Wafer
SN8F5721H	Dice
SN8F5721PG	PDIP, 14 pins, Green package
SN8F5721SG	SOP, 14 pins, Green package
SN8F5721JG	QFN16 3x3, 16 pins, Green package
SN8F57211AG	MSOP, 10 pins, Green package
SN8F57212AG	MSOP, 10 pins, Green package
SN8F57213SG	SOP, 8 pins, Green package
SN8F57214SG	SOP, 14 pins, Green package
SN8F57215SG	SOP, 8 pins, Green package
SN8F57216SG	SOP, 14 pins, Green package
SN8F57217TG	SOP, 14 pins, Green package

30 Package Information

30.1 P-DIP14



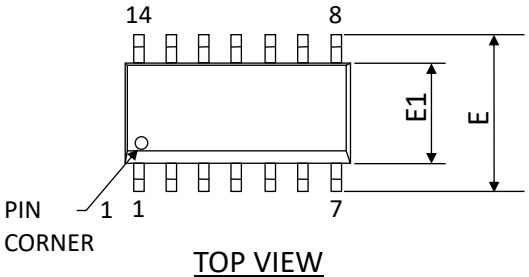


SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	5.33	--	--	0.210
A1	0.38	--	--	0.015	--	--
A2	3.18	3.30	3.43	0.125	0.130	0.135
B	0.46 typ.			0.018 typ.		
B1	1.52 typ.			0.060 typ.		
D	18.67	19.05	19.69	0.735	0.750	0.775
E	7.62 BSC			0.300BSC		
E1	6.22	6.35	6.48	0.245	0.250	0.255
e1	2.54 typ.			0.100 typ.		
L	2.92	3.30	3.81	0.115	0.130	0.150
eA	7.62	9.02	9.53	0.300	0.355	0.375
θ	0°	7°	15°	0°	7°	15°

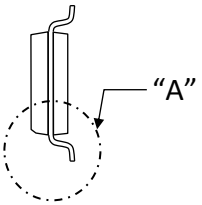
Notes：

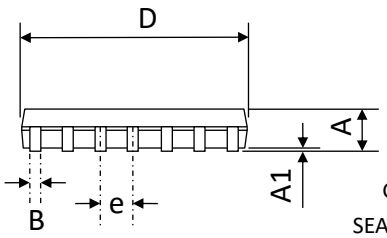
1. JEDEC OUTLINE：MS-001 AA
2. CONTROLLING DIMENSION：inch

30.2 SOP14

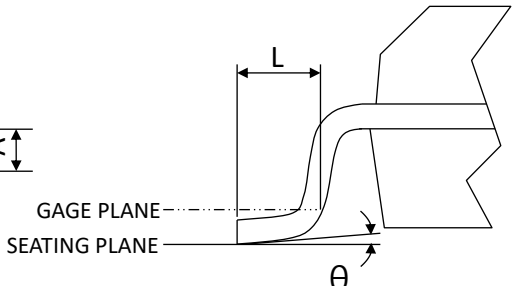


TOP VIEW





SIDE VIEW



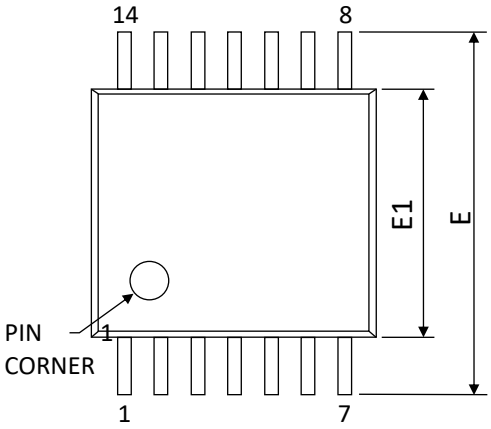
DETAIL "A"

SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	1.75	--	--	0.069
A1	0.05	--	0.25	0.002	--	0.010
B	0.31	--	0.51	0.012	--	0.02
D	8.65 BSC			0.340 BSC		
E1	3.90 BSC			0.154 BSC		
e	1.27 BSC			0.050 BSC		
E	6.00 BSC			0.236 BSC		
L	0.4	--	1.27	0.015	--	0.050
θ	0°	--	8°	0°	--	8°

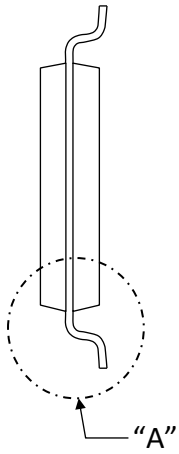
Notes：

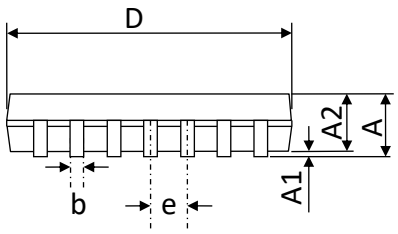
1. CONTROLLING DIMENSION：mm
2. JEDEC OUTLINE : MS-012 AB

30.3 TSSOP14

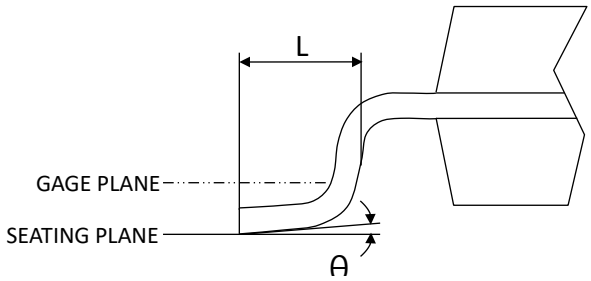


TOP VIEW





SIDE VIEW



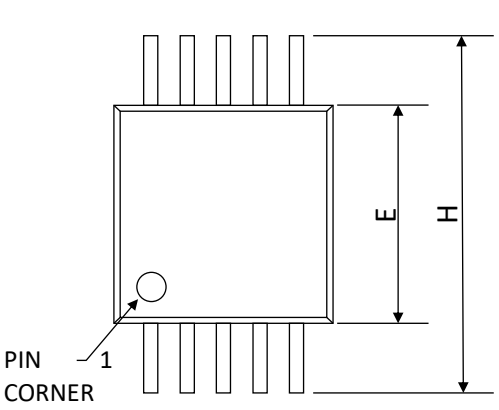
DETAIL "A"

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	4.80	--	5.20	0.189	--	0.205
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°

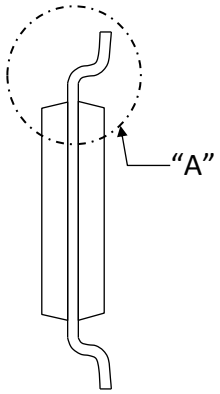
Notes：

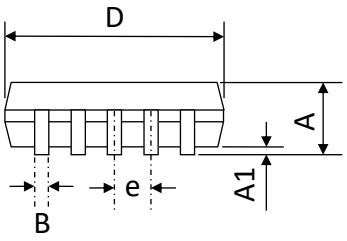
1. CONTROLLING DIMENSION：mm
2. JEDEC OUTLINE : MO-153
3. DIMENSION 'D' DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BERRES.
4. DIMENSION 'E1' DOES NOT INCLUDE INTERLEAD FLASH OR PROTRUSION.
5. DIMENSION 'b' DOES NOT INCLUDE DAMBAR PROTRUSION.

30.4 MSOP10

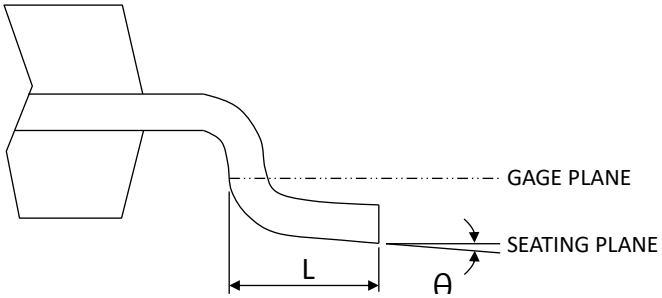


TOP VIEW





SIDE VIEW



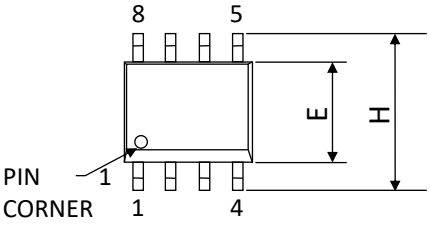
DETAIL "A"

SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	1.10	--	--	0.043
A1	0.00	--	0.15	0.000	--	0.006
B	0.17	--	0.27	0.007	--	0.011
D	3.00 BSC			0.118 BSC		
E	3.00 BSC			0.118 BSC		
e	0.50 BSC			0.020 BSC		
H	4.9 BSC			0.193 BSC		
L	0.40	0.60	0.80	0.016	0.024	0.031
θ	0°	4°	8°	0°	4°	8°

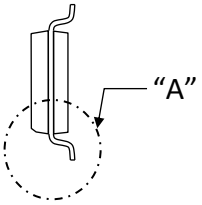
Notes：

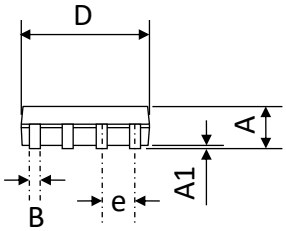
1. CONTROLLING DIMENSION：mm
2. JEDEC OUTLINE : MO-187 BA

30.5 SOP8

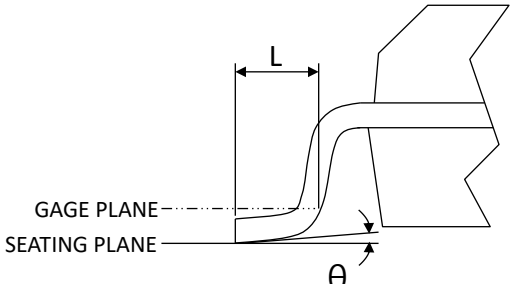


TOP VIEW





SIDE VIEW



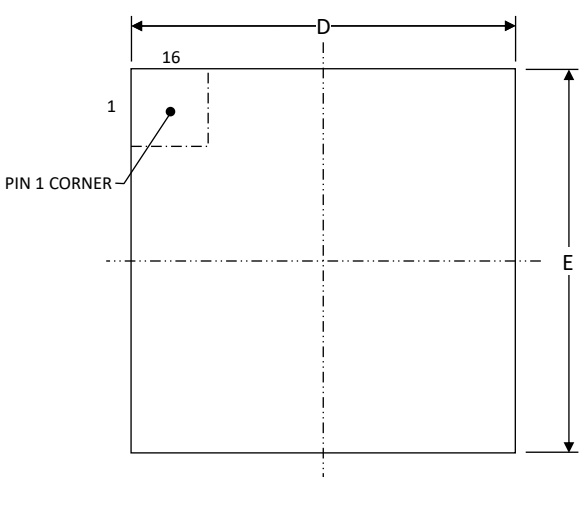
DETAIL "A"

SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	1.75	--	--	0.069
A1	0.10	--	0.25	0.004	--	0.010
B	0.31	--	0.51	0.012	--	0.020
D	4.90 BSC			0.193 BSC		
E	3.90 BSC			0.153 BSC		
e	1.27 BSC			0.050 BSC		
H	6.00 BSC			0.236 BSC		
L	0.40	--	1.27	0.016	--	0.050
θ	0°	--	8°	0°	--	8°

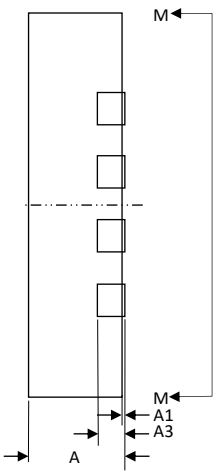
Notes：

1. CONTROLLING DIMENSION：mm
2. JEDEC OUTLINE : MS-012 AA

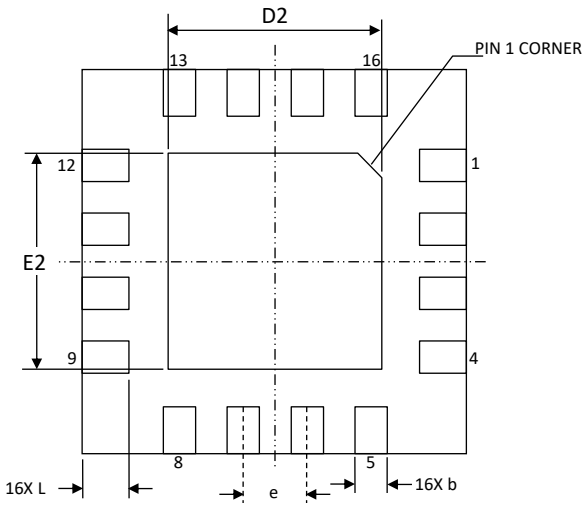
30.6 QFN16 3x3



TOP VIEW



SIDE VIEW



BOTTOM VIEW

VIEW M - M

SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	0.70	0.80	0.90	0.028	0.031	0.035
A1	0.00	0.02	0.05	0.000	0.001	0.002
A3		0.20 REF			0.008 REF	
b	0.18	0.25	0.30	0.007	0.010	0.012
D		3.00 BSC			0.118 BSC	
E		3.00 BSC			0.118 BSC	
e		0.50 BSC			0.020 BSC	
D2	1.40	1.60	1.80	0.055	0.063	0.070
E2	1.40	1.60	1.80	0.055	0.063	0.070
L	0.25	0.35	0.45	0.010	0.014	0.018

Notes：

1. CONTROLLING DIMENSION：MILLIMETER (mm)

31 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 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.

SN8F5721 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
Zhongliang Ziyun Building, District 22, Lingzhiyuan Community,
Xin'an Street, Bao'an District,
Shenzhen City, 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