SONiX Technology Co., Ltd.

SN8F5713 Series Datasheet

8051-based Microcontroller

SN8F5713
SN8F5712
SN8F5711
SN8F57131
SN8F57112
SN8F57113

1 Device Overview

1.1 Features

- **Enhanced 8051 microcontroller** with reduced instruction cycle time (up to 12 times 80C51)
- Up to 8 MHz flexible CPU frequency
- Internal 32 MHz Clock Generator (IHRC), 1 MHz to 16 MHz crystal, and external synchronous clock source selections
- Real-time clock with 32.768 kHz crystal
- **Memory configuration**
  8 KB on-chip Flash program memory (IROM)
  256 bytes on-chip internal RAM (IRAM)
  512 bytes on-chip external RAM (XRAM)
- **10 interrupt sources with priority levels control and unique interrupt vectors**
  7 internal interrupts
  3 external interrupts: INT0, INT1, INT2
- 1 set of DPTR
- 2 set 8/16-bit timers with 4 operation modes
- **1 set 16-bit PWM generators**:
  PWM generator has 8 output channels with individual duty cycle control
- **12-bit SAR ADC** with 13 external and 1 internal channels, and 4 internal reference voltages
- **15 channel Capacitive touch**
- **UART, I2C** interface with SMBus Support
- **On-Chip Debug Support**:
  Single-wire debug interface
  3 hardware breakpoints
  Unlimited software breakpoints
  ROM data security/protection
- Watchdog and programmable external reset
- 1.8/2.4/3.3-V low voltage detectors
- Wide supply voltage (1.8 V – 5.5 V) and temperature (-40 °C to 85 °C) range
- **Hardware Multiplication/Division Unit**

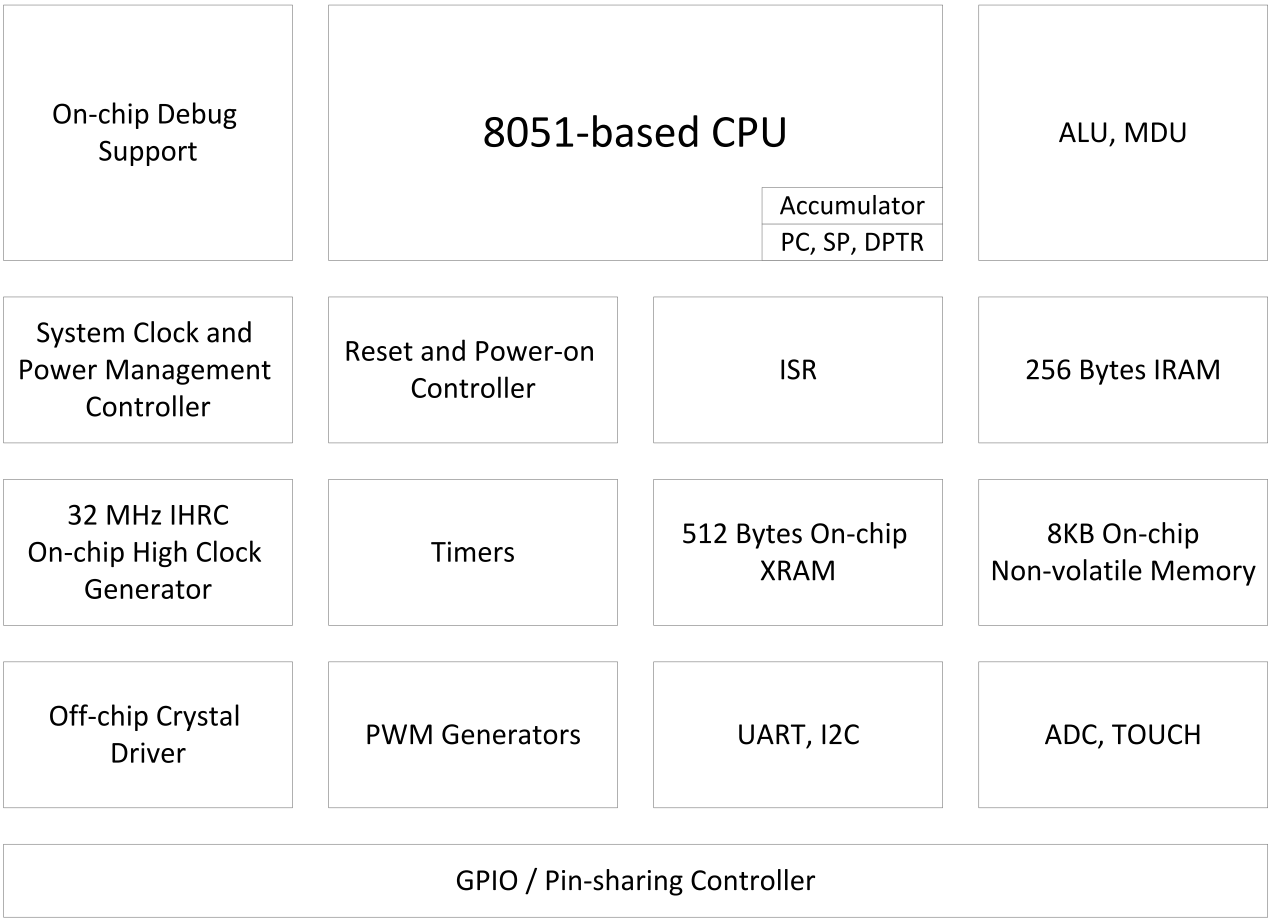
1.2 Applications

- Home automation
- Touch Key

1.3 Features Selection Table

	I/O	PWM Channels	I2C	SPI	UART	ADC ext. Channels	Touch	Ext. INT	Package Types
SN8F5713	21	8	V	-	V	13	15	3	SOP24 TSSOP24 QFN24
SN8F5712	17	8	V	-	V	11	12	2	DIP20 SOP20
SN8F5711	13	8	V	-	V	9	10	2	DIP16 SOP16
SN8F57131	11	7	V	-	V	7	8	2	DIP14 SOP14
SN8F57112	5	3	V	-	V	3	3	-	SOP8
SN8F57113	6	4	V	-	V	4	-	-	SOP8

1.4 Block Diagram


On-chip Debug Support
8051-based CPU
Accumulator
PC, SP, DPTR
ALU, MDU
System Clock and Power Management Controller
Reset and Power-on Controller
ISR
256 Bytes IRAM
32 MHz IHRC On-chip High Clock Generator
Timers
512 Bytes On-chip XRAM
8KB On-chip Non-volatile Memory
Off-chip Crystal Driver
PWM Generators
UART, I2C
ADC, TOUCH
GPIO / Pin-sharing Controller

2 Table of Contents

3 Revision History

Revision	Date	Description
1.0	Mar. 2018	First issue.
1.1	Apr. 2018	Modify UART & PWM register.
1.2	May. 2018	1. Add STOP mode current in ILRC enable. 2. Modify CPU frequency from 32MHz to 8MHz. 3. Modify UART baud rates table.
1.3	Nov. 2018	1. Repair an error, omission, etc. 2. MP5 Writer Programming Pin Mapping adds normal mode and high speed mode sections. 3. Remove SKDIP package type. 4. Modify Pin Circuit Diagrams section. 5. Modify Starter-Kit section. 6. Add P25 Pin Information.
1.4	Apr. 2019	1. Repair an error, omission, etc. 2. Modify ADC input offset range. 3. Modify power on sequence and system clock timing. 4. Modify Package Information section. 5. Modify Timer0/ Timer1 section. 6. Remove ADT register and offset calibration description. 7. Add SN8F57112SG & SN8F57113SG Package Type. 8. Remove Noise Detect Option.
1.5	Aug. 2019	1. Remove ADC VDD/ VSS channel. 2. Modify features selection table and device nomenclature.
1.6	Apr. 2020	1. Modify IO/ PSW register description. 2. Modify Pin Assignments description. 3. Modify Watchdog Reset Time description. 4. Modify wake-up time. 5. Remove I2C 6MHz frequency. 6. Remove MP5 High Speed Mode. 7. Modify Electrical Characteristic section. 8. Modify ROM Programming Pin section.
1.7	Feb. 2021	1. Remove IEN1 register for typing error.
1.8	Feb. 2023	1. Modify package information section.
1.9	April. 2024	1. Add note for pin assignments and GPIO section about not pin out I/O. 2. Add note for ISP operation. 3. Modify electrical characteristic section.

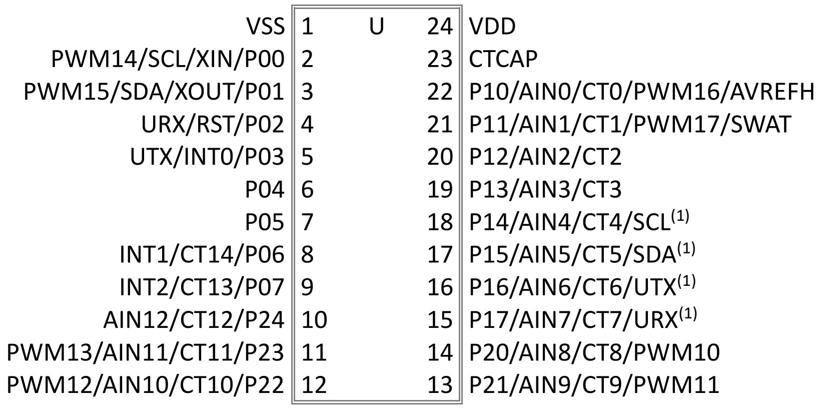
		4. Modify memory description in feature table. 5. Modify typing error.
2.0	Jun.2024	1. Modify ordering Information section. 2. Remove Sample codes.
2.1	Feb.2025	1. Modify PWM duty descriptions. 2. Modify GPIO register descriptions.

SONIX reserves the right to make change without further notice to any products herein to improve reliability, function or design. SONIX does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. SONIX products are not designed, intended, or authorized for us as components in systems intended, for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the SONIX product could create a situation where personal injury or death may occur. Should Buyer purchase or use SONIX products for any such unintended or unauthorized application. Buyer shall indemnify and hold SONIX and its officers, employees, subsidiaries, affiliates and distributors harmless against all claims, cost, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use even if such claim alleges that SONIX was negligent regarding the design or manufacture of the part.

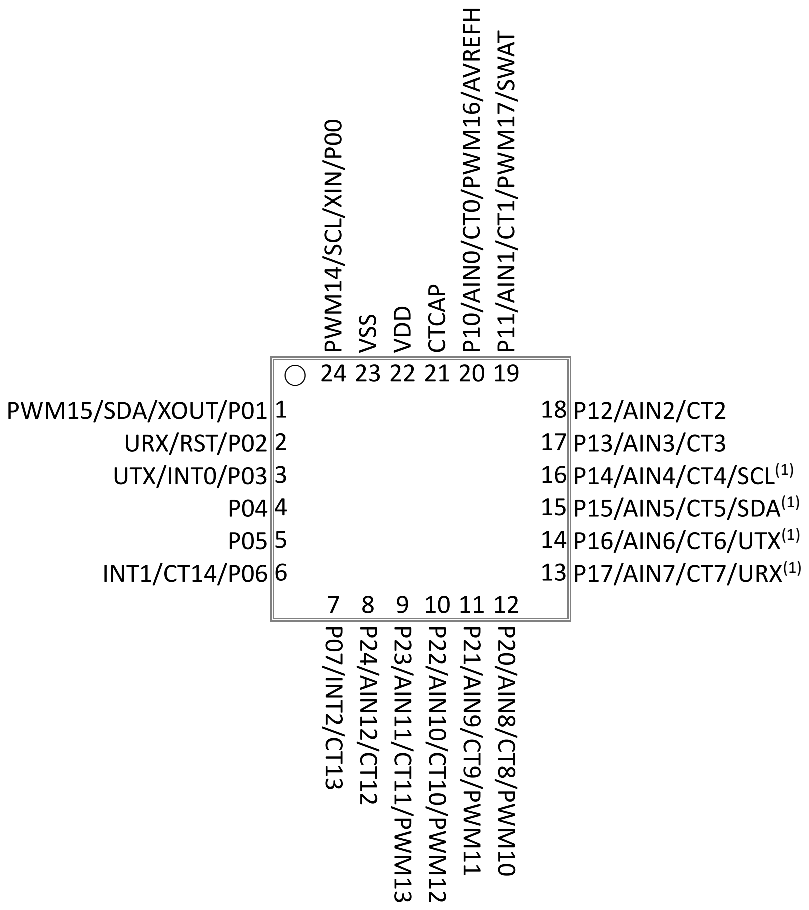
4 Pin Assignments

4.1 Pin Diagrams

- SN8F5713S/T (SOP24/TSSOP24)



- SN8F5713J (QFN24)



- SN8F5712P/S (DIP20/SOP20)

VSS	1	U	20	VDD
PWM14/SCL/XIN/P00	2		19	CTCAP
PWM15/SDA/XOUT/P01	3		18	P10/AIN0/CT0/PWM16/AVREFH
URX/RST/P02	4		17	P11/AIN1/CT1/PWM17/SWAT
UTX/INT0/P03	5		16	P14/AIN4/CT4/SCL(1)
P05	6		15	P15/AIN5/CT5/SDA(1)
INT1/CT14/P06	7		14	P16/AIN6/CT6/UTX(1)
AIN12/CT12/P24	8		13	P17/AIN7/CT7/URX(1)
PWM13/AIN11/CT11/P23	9		12	P20/AIN8/CT8/PWM10
PWM12/AIN10/CT10/P22	10		11	P21/AIN9/CT9/PWM11

✷ ***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.***

- SN8F5711P/S (DIP16/SOP16)

VSS	1	U	16	VDD
PWM14/SCL/XIN/P00	2		15	CTCAP
PWM15/SDA/XOUT/P01	3		14	P10/AIN0/CT0/PWM16/AVREFH
UTX/INT0/P03	4		13	P11/AIN1/CT1/PWM17/SWAT
INT1/CT14/P06	5		12	P16/AIN6/CT6/UTX(1)
AIN12/CT12/P24	6		11	P17/AIN7/CT7/URX(1)
PWM13/AIN11/CT11/P23	7		10	P20/AIN8/CT8/PWM10
PWM12/AIN10/CT10/P22	8		9	P21/AIN9/CT9/PWM11

✷ ***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.***

- SN8F57131P/S (DIP14/SOP14)

VSS	1	U	14	VDD
PWM14/SCL/XIN/P00	2		13	CTCAP
PWM15/SDA/XOUT/P01	3		12	P11/AIN1/CT1/PWM17/SWAT
UTX/INT0/P03	4		11	P16/AIN6/CT6/UTX(1)
INT1/CT14/P06	5		10	P17/AIN7/CT7/URX(1)
PWM13/AIN11/CT11/P23	6		9	P20/AIN8/CT8/PWM10
PWM12/AIN10/CT10/P22	7		8	P21/AIN9/CT9/PWM11

- ✸ ***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.***

- SN8F57112S (SOP8)

VSS	1	U	8	VDD
PWM15/SDA/XOUT/P01	2		7	CTCAP
PWM14/SCL/XIN/P00	3		6	P11/AIN1/CT1/PWM17/SWAT
UTX(1) / CT6/AIN6/P16	4		5	P17/AIN7/CT7/URX(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.***

- SN8F57113S (SOP8)

VSS	1	U	8	VDD
PWM15/SDA/XOUT/P01	2		7	P10/AIN0/CT0/PWM16/AVREFH
PWM14/SCL/XIN/P00	3		6	P11/AIN1/CT1/PWM17/SWAT
UTX(1) / CT6/AIN6/P16	4		5	P17/AIN7/CT7/URX(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.***

Note :
1: I2C output is alternately SCL/SDA or /SCL(1) /SDA(1). UART output is alternately UTX/URX or /UTX(1) /URX(1)

4.2 Pin Allocation Table

- SN8F5713S/T/J (SOP24/TSSOP24/QFN24)

Port	Open-Drain	1/2*VDD Bias Voltage	Sink Current 100mA VSS+1.5V	Wakeup	External Interrupt	ADC	Cap-sensing touch	UART	I2C	PWM	External Clock	External Reset	Debug interface
P0.0	-	V	V	V	-	-	-	-	SCL	PWM14	XIN	-	-
P0.1	-	V	V	V	-	-	-	-	SDA	PWM15	XOUT	-	-
P0.2	V	V	V	V	-	-	-	URX	-	-	-	RST	-
P0.3	V	V	V	V	INT0	-	-	UTX	-	-	-	-	-
P0.4	-	-	-	V	-	-	-	-	-	-	-	-	-
P0.5	-	-	-	V	-	-	-	-	-	-	-	-	-
P0.6	-	-	-	V	INT1	-	CT14	-	-	-	-		-
P0.7	-	-	-	V	INT2	-	CT13	-	-	-	-		-
P1.0	-	V	V	V	-	AVREFH/ AIN0	CT0	-	-	PWM16	-	-	-
P1.1	-	V	V	V	-	AIN1	CT1	-	-	PWM17	-	-	SWAT
P1.2	-	V	V	V	-	AIN2	CT2	-	-	-	-	-	-
P1.3	-	V	V	V	-	AIN3	CT3	-	-	-	-	-	-
P1.4	-	-	-	V	-	AIN4	CT4	-	SCL[(1)]	-	-	-	-
P1.5	-	-	-	V	-	AIN5	CT5	-	SDA[(1)]	-	-	-	-
P1.6	V	-	-	V	-	AIN6	CT6	UTX[(1)]	-	-	-	-	-
P1.7	V	-	-	V	-	AIN7	CT7	URX[(1)]	-	-	-	-	-
P2.0	-	-	-	-	-	AIN8	CT8	-	-	PWM10	-		-
P2.1	-	-	-	-	-	AIN9	CT9	-	-	PWM11	-	-	-
P2.2	-	-	-	-	-	AIN10	CT10	-	-	PWM12	-	-	-
P2.3	-	-	-	-	-	AIN11	CT11	-	-	PWM13	-	-	-
P2.4	-	-	-	-	-	AIN12	CT12	-	-	-	-	-	-
CTCAP	-	-	-	-	-	-	CTCAP	-	-	-	-	-	-

- SN8F5712P/S (DIP20/SOP20)

Port	Open-Drain	1/2*VDD Bias Voltage	Sink Current 100mA VSS+1.5V	Wakeup	External Interrupt	ADC	Cap-sensing touch	UART	I2C	PWM	External Clock	External Reset	Debug interface
P0.0	-	V	V	V	-	-	-	-	SCL	PWM14	XIN	-	-
P0.1	-	V	V	V	-	-	-	-	SDA	PWM15	XOUT	-	-
P0.2	V	V	V	V	-	-	-	URX	-	-	-	RST	-
P0.3	V	V	V	V	INT0	-	-	UTX	-	-	-	-	-
P0.5	-	-	-	V	-	-	-	-	-	-	-	-	-
P0.6	-	-	-	V	INT1	-	CT14	-	-	-	-		-
P1.0	-	V	V	V	-	AVREFH/ AIN0	CT0	-	-	PWM16	-	-	-
P1.1	-	V	V	V	-	AIN1	CT1	-	-	PWM17	-	-	SWAT
P1.4	-	-	-	V	-	AIN4	CT4	-	SCL[1]	-	-	-	-
P1.5	-	-	-	V	-	AIN5	CT5	-	SDA[1]	-	-	-	-
P1.6	V	-	-	V	-	AIN6	CT6	UTX[1]	-	-	-	-	-
P1.7	V	-	-	V	-	AIN7	CT7	URX[1]	-	-	-	-	-
P2.0	-	-	-	-	-	AIN8	CT8	-	-	PWM10	-		-
P2.1	-	-	-	-	-	AIN9	CT9	-	-	PWM11	-	-	-
P2.2	-	-	-	-	-	AIN10	CT10	-	-	PWM12	-	-	-
P2.3	-	-	-	-	-	AIN11	CT11	-	-	PWM13	-	-	-
P2.4	-	-	-	-	-	AIN12	CT12	-	-	-	-	-	-
CTCAP	-	-	-	-	-	-	CTCAP	-	-	-	-	-	-

- SN8F5711P/S (DIP16/SOP16)

Port	Open-Drain	1/2*VDD Bias Voltage	Sink Current 100mA VSS+1.5V	Wakeup	External Interrupt	ADC	Cap-sensing touch	UART	I2C	PWM	External Clock	External Reset	Debug interface
P0.0	-	V	V	V	-	-	-	-	SCL	PWM14	XIN	-	-
P0.1	-	V	V	V	-	-	-	-	SDA	PWM15	XOUT	-	-
P0.3	V	V	V	V	INT0	-	-	UTX	-	-	-	-	-
P0.6	-	-	-	V	INT1	-	CT14	-	-	-	-		-
P1.0	-	V	V	V	-	AVREFH/ AIN0	CT0	-	-	PWM16	-	-	-
P1.1	-	V	V	V	-	AIN1	CT1	-	-	PWM17	-	-	SWAT
P1.6	V	-	-	V	-	AIN6	CT6	UTX(1)	-	-	-	-	-
P1.7	V	-	-	V	-	AIN7	CT7	URX(1)	-	-	-	-	-
P2.0	-	-	-	-	-	AIN8	CT8	-	-	PWM10	-		-
P2.1	-	-	-	-	-	AIN9	CT9	-	-	PWM11	-	-	-
P2.2	-	-	-	-	-	AIN10	CT10	-	-	PWM12	-	-	-
P2.3	-	-	-	-	-	AIN11	CT11	-	-	PWM13	-	-	-
P2.4	-	-	-	-	-	AIN12	CT12	-	-	-	-	-	-
CTCAP	-	-	-	-	-	-	CTCAP	-	-	-	-	-	-

- SN8F57131P/S (DIP14/SOP14)

Port	Open-Drain	1/2*VDD Bias Voltage	Sink Current 100mA VSS+1.5V	Wakeup	External Interrupt	ADC	Cap-sensing touch	UART	I2C	PWM	External Clock	External Reset	Debug interface
P0.0	-	V	V	V	-	-	-	-	SCL	PWM14	XIN	-	-
P0.1	-	V	V	V	-	-	-	-	SDA	PWM15	XOUT	-	-
P0.3	V	V	V	V	INT0	-	-	UTX	-	-	-	-	-
P0.6	-	-	-	V	INT1	-	CT14	-	-	-	-		-
P1.1	-	V	V	V	-	AIN1	CT1	-	-	PWM17	-	-	SWAT
P1.6	V	-	-	V	-	AIN6	CT6	UTX(1)	-	-	-	-	-
P1.7	V	-	-	V	-	AIN7	CT7	URX(1)	-	-	-	-	-
P2.0	-	-	-	-	-	AIN8	CT8	-	-	PWM10	-		-
P2.1	-	-	-	-	-	AIN9	CT9	-	-	PWM11	-	-	-
P2.2	-	-	-	-	-	AIN10	CT10	-	-	PWM12	-	-	-
P2.3	-	-	-	-	-	AIN11	CT11	-	-	PWM13	-	-	-
CTCAP	-	-	-	-	-	-	CTCAP	-	-	-	-	-	-

- SN8F57112S (SOP8)

Port	Open-Drain	1/2*VDD Bias Voltage	Sink Current 100mA VSS+1.5V	Wakeup	External Interrupt	ADC	Cap-sensing touch	UART	I2C	PWM	External Clock	External Reset	Debug interface
P0.0	-	V	V	V	-	-	-	-	SCL	PWM14	XIN	-	-
P0.1	-	V	V	V	-	-	-	-	SDA	PWM15	XOUT	-	-
P1.1	-	V	V	V	-	AIN1	CT1	-	-	PWM17	-	-	SWAT
P1.6	V	-	-	V	-	AIN6	CT6	UTX(1)	-	-	-	-	-
P1.7	V	-	-	V	-	AIN7	CT7	URX(1)	-	-	-	-	-
CTCAP	-	-	-	-	-	-	CTCAP	-	-	-	-	-	-

- SN8F57113S (SOP8)

Port	Open-Drain	1/2*VDD Bias Voltage	Sink Current 100mA VSS+1.5V	Wakeup	External Interrupt	ADC	Cap-sensing touch	UART	I2C	PWM	External Clock	External Reset	Debug interface
P0.0	-	V	V	V	-	-	-	-	SCL	PWM14	XIN	-	-
P0.1	-	V	V	V	-	-	-	-	SDA	PWM15	XOUT	-	-
P1.0	-	V	V	V	-	AVREFH/ AIN0	CT0	-	-	PWM16	-	-	-
P1.1	-	V	V	V	-	AIN1	CT1	-	-	PWM17	-	-	SWAT
P1.6	V	-	-	V	-	AIN6	CT6	UTX[1]	-	-	-	-	-
P1.7	V	-	-	V	-	AIN7	CT7	URX[1]	-	-	-	-	-

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 XIN SCL PWM14	Digital I/O Analog Input Digital I/O Digital Output	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up. System clock: external clock input. I2C: clock output (master) or clock input (slave). PWM: programmable PWM output.
P0.1 XOUT SDA PWM15	Digital I/O Analog Output Digital I/O Digital Output	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up. System clock: drive external crystal/resonator. I2C: data pin. PWM: 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.

RST	Digital Input	System reset (active low).
URX	Digital Input	UART: reception pin.
P0.3	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
UTX	Digital Output	UART: transmission pin.
INT0	Digital Input	INT0: external interrupt 0.
P0.4	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
P0.5	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
P0.6	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
INT1	Digital Input	INT1: external interrupt 1.
CT14	Analog Input	CT14: Cap-sensing touch input channel 14.
P0.7	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
INT2	Digital Input	INT2: external interrupt 2.
CT13	Analog Input	CT13: Cap-sensing touch input channel 13.

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.
PWM16	Digital Output	PWM: programmable PWM output.
CT0	Analog Input	CT0: Cap-sensing touch input channel 0.
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.
SWAT	Digital I/O	Debug interface.
PWM17	Digital Output	PWM: programmable PWM output.
CT1	Analog Input	CT1: Cap-sensing touch input channel 1.
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.
CT2	Analog Input	CT2: Cap-sensing touch input channel 2.
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.
CT3	Analog Input	CT3: Cap-sensing touch input channel 3.
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.
SCL[1]	Digital I/O	I2C: clock output (master) or clock input (slave).
CT4	Analog Input	CT4: Cap-sensing touch input channel 4.
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.
SDA[1]	Digital I/O	I2C: data pin.
CT5	Analog Input	CT5: Cap-sensing touch input channel 5.
P1.6	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.
UTX[1]	Digital Output	UART: transmission pin.
CT6	Analog Input	CT6: Cap-sensing touch input channel 6
P1.7	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[1]	Digital Input	UART: reception pin.
CT7	Analog Input	CT7: Cap-sensing touch input channel 7.

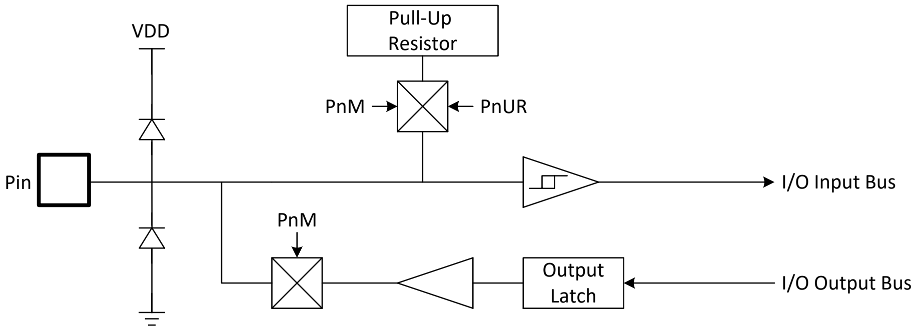
Port 2

Pin Name	Type	Description
P2.0	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
AIN8	Analog Input	ADC: input channel 8.
PWM10	Digital Output	PWM: programmable PWM output.
CT8	Analog Input	CT8: Cap-sensing touch input channel 8.
P2.1	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
AIN9	Analog Input	ADC: input channel 9.
PWM11	Digital Output	PWM: programmable PWM output.
CT9	Analog Input	CT9: Cap-sensing touch input channel 9.
P2.2	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.

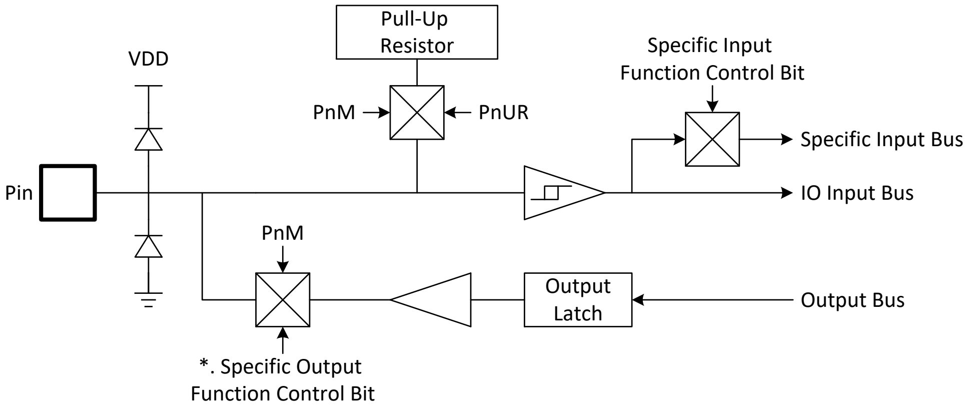
AIN10	Analog Input	ADC: input channel 10.
PWM12	Digital Output	PWM: programmable PWM output.
CT10	Analog Input	CT10: Cap-sensing touch input channel 10.
P2.3	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
AIN11	Analog Input	ADC: input channel 11.
PWM13	Digital Output	PWM: programmable PWM output.
CT11	Analog Input	CT11: Cap-sensing touch input channel 11.
P2.4	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
AIN12	Analog Input	ADC: input channel 12.
CT12	Analog Input	CT12: Cap-sensing touch input channel 12.
P2.5	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.

4.4 Pin Circuit Diagrams

Normal Bi-direction I/O Pin.

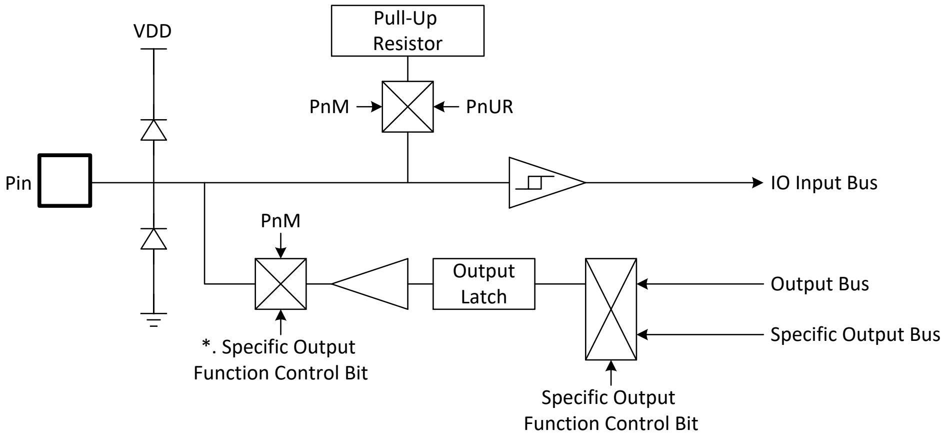


Bi-direction I/O Pin Shared with Specific Digital Input Function, e.g. INT2.



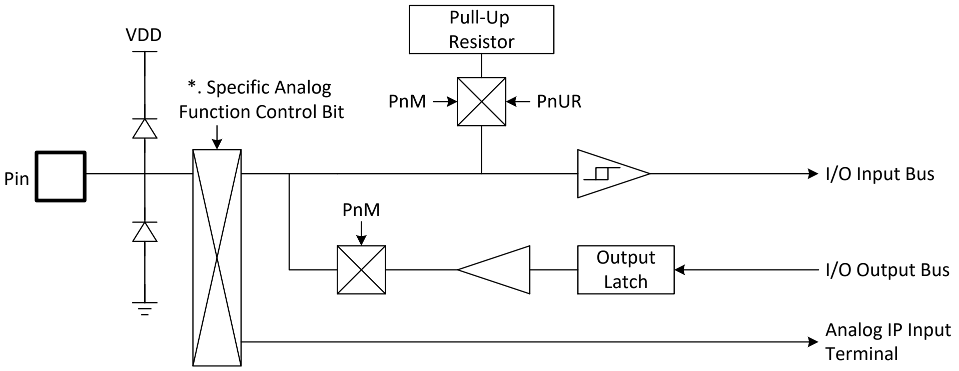
***. Some specific functions switch I/O direction directly, not through PnM register.**

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



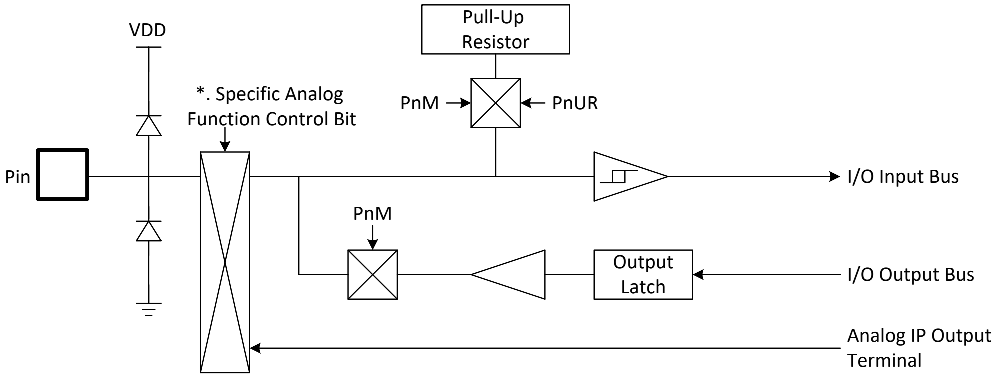
***. Some specific functions switch I/O direction directly, not through PnM register.**

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



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

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



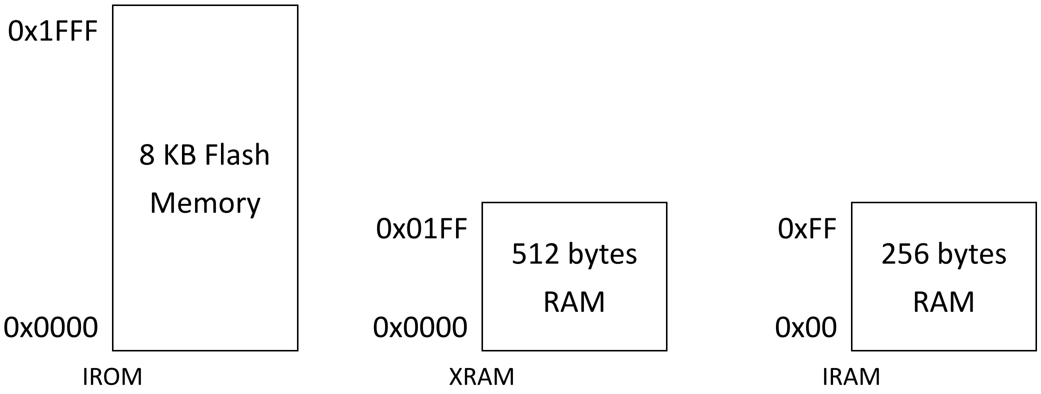
***. Some specific functions switch I/O direction directly, not through PnM register.**

5 CPU

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

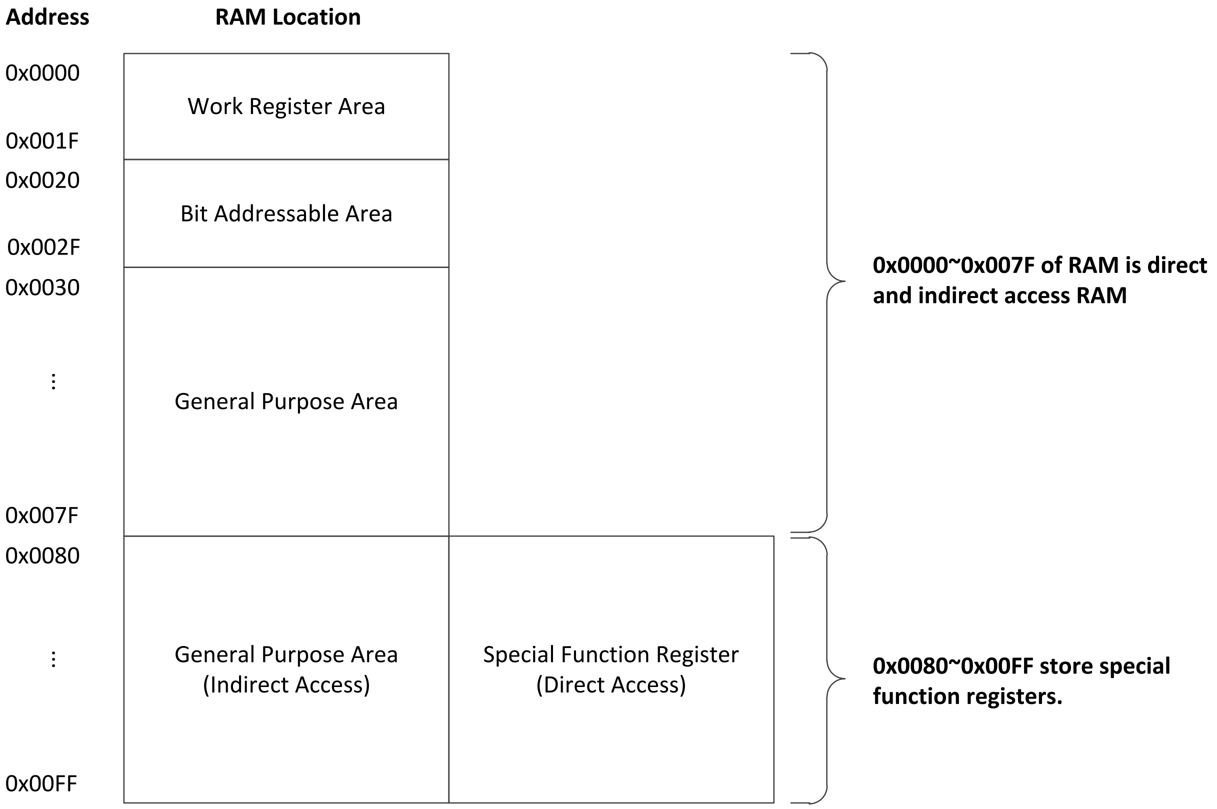
5.1 Memory Organization

SN8F5713 builds in three on-chip memories: internal RAM (IRAM), external RAM (XRAM), and program memory (IROM). The internal RAM is a 256-byte RAM which has higher access performance (direct and indirect addressing). By contrast, the external RAM has 512-byte of size, but it requires a longer access period. The program memory is a 8 KB non-volatile memory and has a maximum 8 MHz speed limitation.



5.2 Internal RAM (IRAM)

256 X 8-bit RAM (Internal Data Memory)


Address
RAM Location
0x0000
Work Register Area
0x001F
0x0020
Bit Addressable Area
0x002F
0x0030
⋮
General Purpose Area
0x007F
0x0080
⋮
General Purpose Area (Indirect Access)
Special Function Register (Direct Access)
0x00FF
0x0000~0x007F of RAM is direct and indirect access RAM
0x0080~0x00FF store special function registers.

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

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

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

	Byte Address	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
Bit Addressable Area	**0x20**	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
	0x21	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
	0x22	0x10	0x11	0x12	0x13	0x14	0x15	0x16	0x17
	0x23	0x18	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F
	0x24	0x20	0x21	0x22	0x23	0x24	0x25	0x26	0x27
	0x25	0x28	0x29	0x2A	0x2B	0x2C	0x2D	0x2E	0x2F
	0x26	0x30	0x31	0x32	0x33	0x34	0x35	0x36	0x37
	0x27	0x38	0x39	0x3A	0x3B	0x3C	0x3D	0x3E	0x3F
	0x28	0x40	0x41	0x42	0x43	0x44	0x45	0x46	0x47
	0x29	0x48	0x49	0x4A	0x4B	0x4C	0x4D	0x4E	0x4F
	0x2A	0x50	0x51	0x52	0x53	0x54	0x55	0x56	0x57
	0x2B	0x58	0x59	0x5A	0x5B	0x5C	0x5D	0x5E	0x5F
	0x2C	0x60	0x61	0x62	0x63	0x64	0x65	0x66	0x67
	0x2D	0x68	0x69	0x6A	0x6B	0x6C	0x6D	0x6E	0x6F
	0x2E	0x70	0x71	0x72	0x73	0x74	0x75	0x76	0x77
	0x2F	0x78	0x79	0x7A	0x7B	0x7C	0x7D	0x7E	0x7F

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

5.3 External RAM (XRAM)

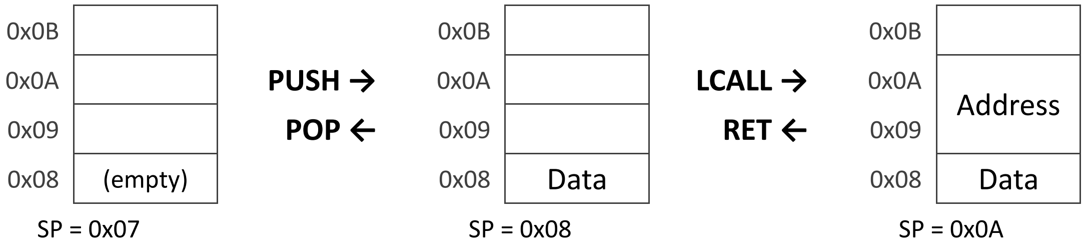
512 X 8-bit SRAM (Extension Data Memory)

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

Address	RAM Location	
0000h	General Purpose Area	Extension RAM Bank 0
…		
01FFh		End of Bank 0

5.4 Stack

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



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

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

✷ ***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.***

5.5 Program Memory (IROM)

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

Address	**ROM**	Comment
0000H	Reset vector	Reset vector
0001H 0002H	General purpose area	User program
0003H	**INT0 Interrupt vector**	**Interrupt vector**
000BH	**PWM1 Interrupt vector**	
0013H	**UART TX Interrupt vector**	
001BH	**ADC Interrupt vector**	
0043H	**INT2 Interrupt vector**	
0053H	**UART RX Interrupt vector**	
0063H	**TIMER0 Interrupt vector**	
0083H	**INT1 Interrupt vector**	
0093H	**I2C Interrupt vector**	
00A3H	**TIMER1 Interrupt vector**	
00A4H	General purpose area	User program End of user program
1FF6H 1FF7H . 1FFDH 1FFEH 1FFFH	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.
- 0x0003~0x00A3: Multi interrupt vector area. Each of interrupt events has a unique interrupt vector.
- 0x00A4~0x1FDF: General purpose area for user program and ISP (EEPROM function).

- 0x1FE0~0x1FF6: General purpose area for user program. Do not execute ISP.
- 0x1FF6~0x1FFF: Reserved area. Do not execute ISP.

5.6 Program Memory Security

The SN8F5713 provides security options at the disposal of the designer to prevent unauthorized access to information stored in FLASH memory. When enable security option, the ROM code is secured and not dumped complete ROM contents. ROM security rule is all address ROM data protected and outputs 0x00.

5.7 Data Pointer

A data pointer helps to specify the XRAM and IROM address while performing MOVX and MOVC instructions. The microcontroller has one set of data pointer (DPH/DPL). The DPC register controls automatically increase/decrease DPTR function.

The automatically increase/decrease DPTR function can make an increment or decrement after perform MOVX @DPTR instruction. As a result, it enables a continuous external RAM access without re-specified DPTR value.

5.8 Stack and Data Pointer Register

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SP	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
DPL	DPL7	DPL6	DPL5	DPL4	DPL3	DPL2	DPL1	DPL0
DPH	DPH7	DPH6	DPH5	DPH4	DPH3	DPH2	DPH1	DPH0
DPC	-	-	-	-	-	ATMS	ATMD	ATME

SP Register (0x81)

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

DPL Register (0x82)

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

DPH Register (0x83)

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

DPC Register (0x93)

Bit	Field	Type	Initial	Description
7..3	Reserved	R	00	
2..1	ATMS/ATMD	R/W	00	Automatically increase/decrease DPTR (if ATME applied) 00: +1 after any MOVX @DPTR instruction 01: -1 after any MOVX @DPTR instruction 10: +2 after any MOVX @DPTR instruction 11: -2 after any MOVX @DPTR instruction
0	ATME	R/W	0	Automatically increase/decrease DPTR function 0: Disable 1: Enable

6 Special Function Registers

6.1 Special Function Register Memory Map

BIN HEX	000	001	010	011	100	101	110	111
F8	-	P0M	P1M	P2M	FRQL	FRQH	FRQCMD	PFLAG
F0	B	P0UR	P1UR	P2UR	P0BIAS	P1BIAS	-	SRST
E8	-	MD0	MD1	MD2	MD3	MD4	MD5	ARCON
E0	ACC	-	-	-	P0OC	CLKSEL	CLKCMD	TCON0
D8	S0CON2	-	I2CDAT	I2CADR	I2CCON	I2CSTA	SMBSEL	SMBDST
D0	PSW	-	ADM	ADB	ADR	VREFH	P2CON	-
C8	-	PW16DL	PW16DH	PW17DL	PW17DH	-	SYSMOD	-
C0	CSCON	CSCON1	CSCON2	SGMOD1	SGMOD2	-	-	-
B8	-	IP1	S0RELH	PW14DL	PW14DH	PW15DL	PW15DH	IRCON2
B0	-	CSCH	CSCL	CCAL	-	-	PWCH	-
A8	IEN0	IP0	S0RELL	PW12DL	PW12DH	PW13DL	PW13DH	
A0	P2	PW1M	PW1YL	PW1YH	PW10DL	PW10DH	PW11DL	PW11DH
98	S0CON	S0BUF	IEN2	-	-	-	P0CON	P1CON
90	P1	P1W	-	DPC	PECMD	PEROML	PEROMH	PERAM
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	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 low byte register.
DPH	0x83	Data pointer 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.
CKCON	0x8E	Extended cycle controls register.
PEDGE	0x8F	External interrupt edge controls register.
P1	0x90	Port 1 data buffer.
P1W	0x91	Port 1 wake-up controls register.
-	0x92	-
DPC	0x93	Data pointer controls register.
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.
-	0x9B	-
-	0x9C	-
-	0x9D	-
P0CON	0x9E	Port 0 configuration controls register.
P1CON	0x9F	Port 1 configuration controls register.

0xA0 - 0xBF Registers Description

Register	Address	Description
P2	0xA0	Port 2 data buffer.
PW1M	0xA1	PW1 controls register.
PW1YL	0xA2	PW1 cycle controls buffer low byte.
PW1YH	0xA3	PW1 cycle controls buffer high byte.
PW10DL	0xA4	PW10 duty controls buffer low byte.
PW10DH	0xA5	PW10 duty controls buffer high byte.
PW11DL	0xA6	PW11 duty controls buffer low byte.
PW11DH	0xA7	PW11 duty controls buffer high byte.
IEN0	0xA8	Interrupts enable register.
IP0	0xA9	Interrupts priority register.
S0RELL	0xAA	UART reload low byte register.
PW12DL	0xAB	PW12 duty controls buffer low byte.
PW12DH	0xAC	PW12 duty controls buffer high byte.
PW13DL	0xAD	PW13 duty controls buffer low byte.
PW13DH	0xAE	PW13 duty controls buffer high byte.
-	0xAF	-
-	0xB0	-
-	0xB1	-
-	0xB2	-
-	0xB3	-
-	0xB4	-
-	0xB5	-
PWCH	0xB6	PWM channel control buffer.
-	0xB7	-
-	0xB8	-
IP1	0xB9	Interrupts priority register.
S0RELH	0xBA	UART reload high byte register.
PW14DL	0xBB	PW14 duty controls buffer low byte.
PW14DH	0xBC	PW14 duty controls buffer high byte.
PW15DL	0xBD	PW15 duty controls buffer low byte.
PW15DH	0xBE	PW15 duty controls buffer high byte.
IRCON2	0xBF	Interrupts request register.

0xC0 - 0xDF Registers Description

Register	Address	Description
-	0xC0	-
-	0xC1	-
-	0xC2	-
-	0xC3	-
-	0xC4	-
-	0xC5	-
-	0xC6	-
-	0xC7	-
-	0xC8	-
PW16DL	0xC9	PW16 duty controls buffer low byte.
PW16DH	0xCA	PW16 duty controls buffer high byte.
PW17DL	0xCB	PW17 duty controls buffer low byte.
PW17DH	0xCC	PW17 duty controls buffer high byte.
-	0xCD	-
SYSMOD	0xCE	System mode register.
-	0xCF	-
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.
P2CON	0xD6	Port 2 configuration controls register.
-	0xD7	-
S0CON2	0xD8	UART baud rate controls register.
-	-	-
I2CDAT	0xDA	I2C data buffer.
I2CADR	0xDB	Own I2C slave address.
I2CCON	0xDC	I2C interface operation control register.
I2CSTA	0xDD	I2C Status Code.
SMBSEL	0xDE	SMBus mode controls register.
SMBDST	0xDF	SMBus internal timeout register.

0xE0 - 0xFF Registers Description

Register	Address	Description
ACC	0xE0	Accumulator register.
-	0xE1	-
-	0xE2	-
-	0xE3	-
P0OC	0xE4	Open drain controls register.
CLKSEL	0xE5	Clock switch selects register.
CLKCMD	0xE6	Clock switch controls Register.
TCON0	0xE7	Timer 0 / 1 clock controls register.
-	0xE8	-
MD0	0xE9	MDU controls register 0.
MD1	0xEA	MDU controls register 1.
MD2	0xEB	MDU controls register 2.
MD3	0xEC	MDU controls register 3.
MD4	0xED	MDU controls register 4.
MD5	0xEE	MDU controls register 5.
ARCON	0xEF	MDU Arithmetic control register.
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.
P2UR	0xF3	Port 2 pull-up resister controls register.
P0BIAS	0xF4	Port 0 1/2*VDD bias controls register.
P1BIAS	0xF5	Port 1 1/2*VDD bias controls register.
-	0xF6	-
SRST	0xF7	Software reset controls register.
-	0xF8	-
P0M	0xF9	Port 0 input/output mode register.
P1M	0xFA	Port 1 input/output mode register.
P2M	0xFB	Port 2 input/output mode register.
FRQL	0xFC	Clock fine tuning controls buffer low byte
FRQH	0xFD	Clock fine tuning controls buffer high byte
FRQCMD	0xFE	Clock fine tuning command register.
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

SN8F5713 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 <SN8F5713.H>
```

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

```
#include <SN8F5713.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
SN8F5713	SN8F5713.h	OPTIONS_SN8F5713.A51
SN8F5712	SN8F5712.h	
SN8F5711	SN8F5711.h	
SN8F57131	SN8F57131.h	
SN8F57112	SN8F57112.h	
SN8F57113	SN8F57113.h	

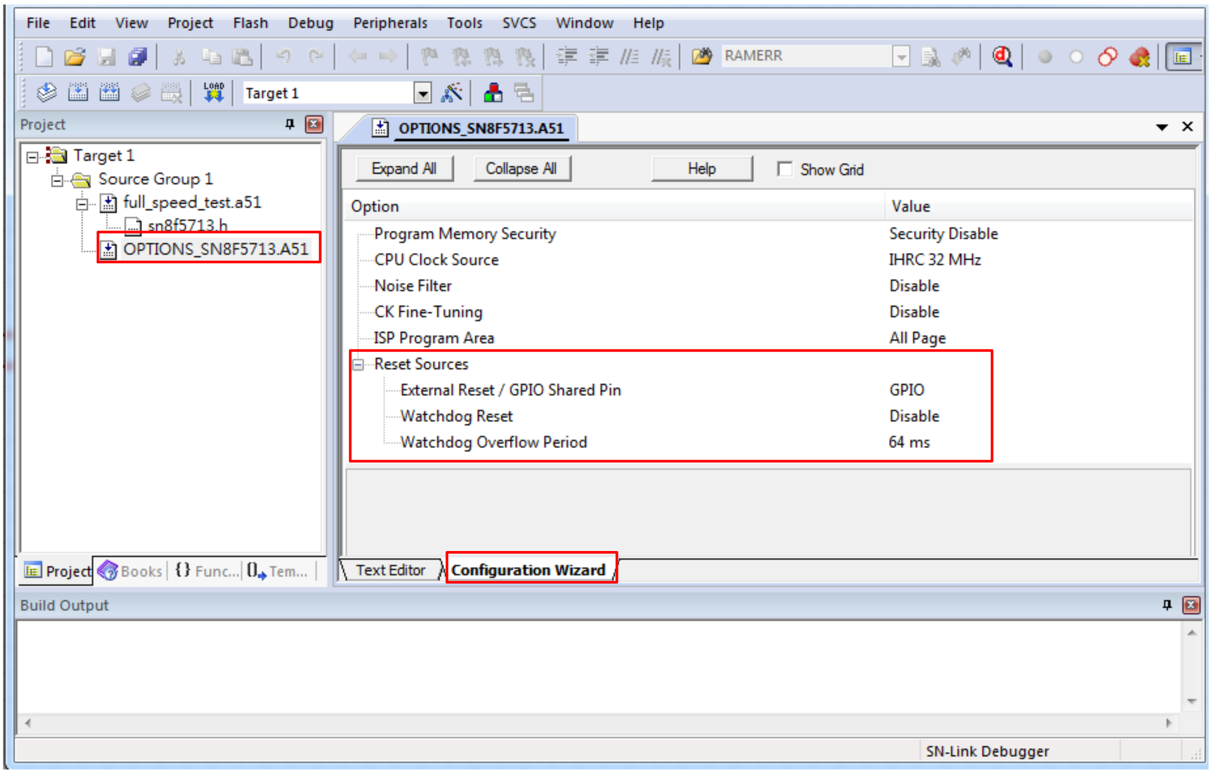
7 Reset and Power-on Controller

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

7.1 Configuration of Reset and Power-on Controller

SONiX publishes a *SN8F5713_OPTIONS.A51 file* in *SN-Link Driver for Keil C51.exe* (downloadable on cooperative website: www.sonix.com.tw). This *options file* contains appropriate parameters of reset sources and CPU clock source selection, and is strongly recommended to add to Keil project. *SN8F5000 Debug Tool Manual* provides the further detail of this configuration. The option items are as following:

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



Option	Value
Program Memory Security	Security Disable
CPU Clock Source	IHRC 32 MHz
Noise Filter	Disable
CK Fine-Tuning	Disable
ISP Program Area	All Page
Reset Sources	
External Reset / GPIO Shared Pin	GPIO
Watchdog Reset	Disable
Watchdog Overflow Period	64 ms

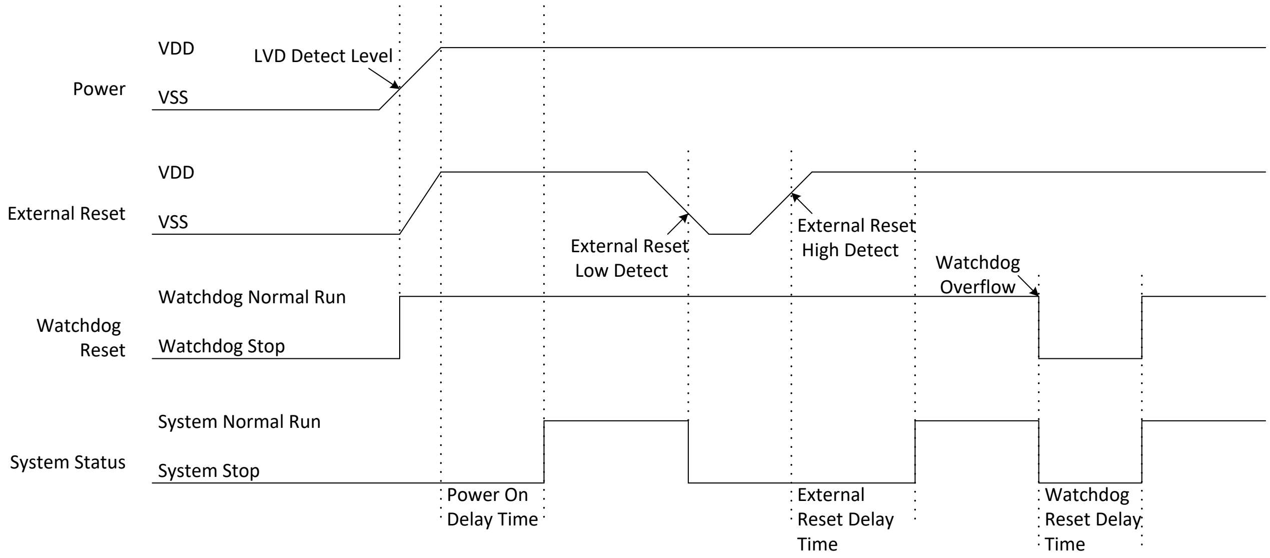
The code option is the system hardware configurations including oscillator type, noise filter option, watchdog timer operation, LVD option, reset pin option and flash ROM security control. The code option items are as following table:

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 0x1F00 ~ 0x1FDF, only address 0x1F00 ~ 0x1FDF ROM data can be accessed
CPU Clock Source	IHRC 32MHz	High speed internal 32MHz RC. XIN/XOUT pins are bi-direction GPIO mode
	IHRC 32MHz with RTC	High speed internal 32MH RC with low speed crystal/resonator (e.g. 32.768kHz). Low speed crystal/resonator for Timer 0 real time clock.
	X’tal 12MHz	High speed crystal /resonator (e.g. 12MHz) for external high clock oscillator
	X’tal 4MHz	Standard crystal /resonator (e.g. 4M) for external high clock oscillator
	External Clock	XIN pin connect external clock (1M ~32M), XOUT pin is bi-direction GPIO mode
CK_Fine_Tuning	Disable	Disable CK_Fine_Tuning
	Enable	Enable CK_Fine_Tuning
ISP Program Area	All Page	All address can perform ISP function
	Page 248~ Page 254	Only address 0x1F00 ~ 0x1FDF can perform ISP function
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.

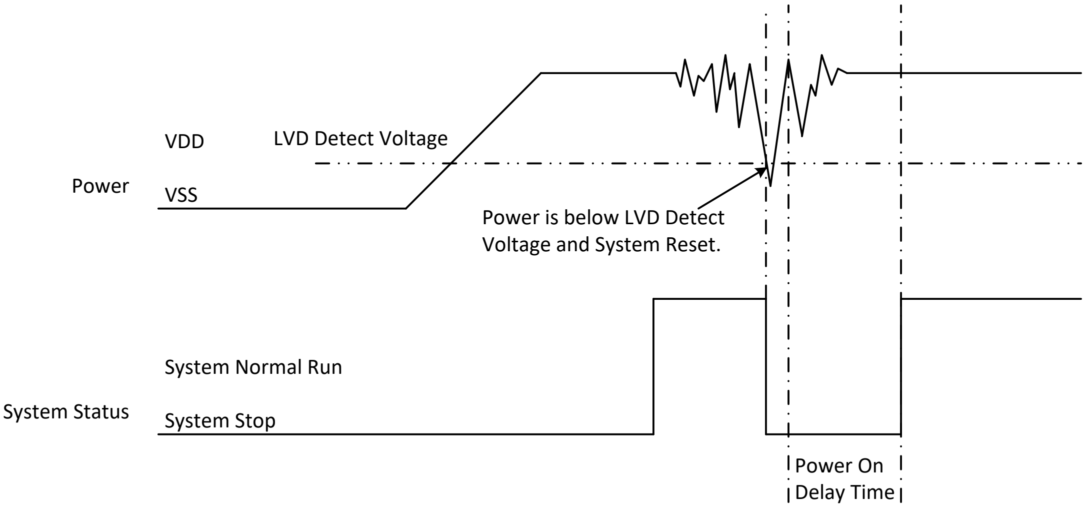


The power stabilization period spends 4.6 ms in typical condition. Afterward the microcontroller fetches CPU Clock Source selection automatically. The selected clock source would be driven, and the system counts 2048 times of the clock period and 4 times of the internal low-speed oscillator clocks to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

```
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;

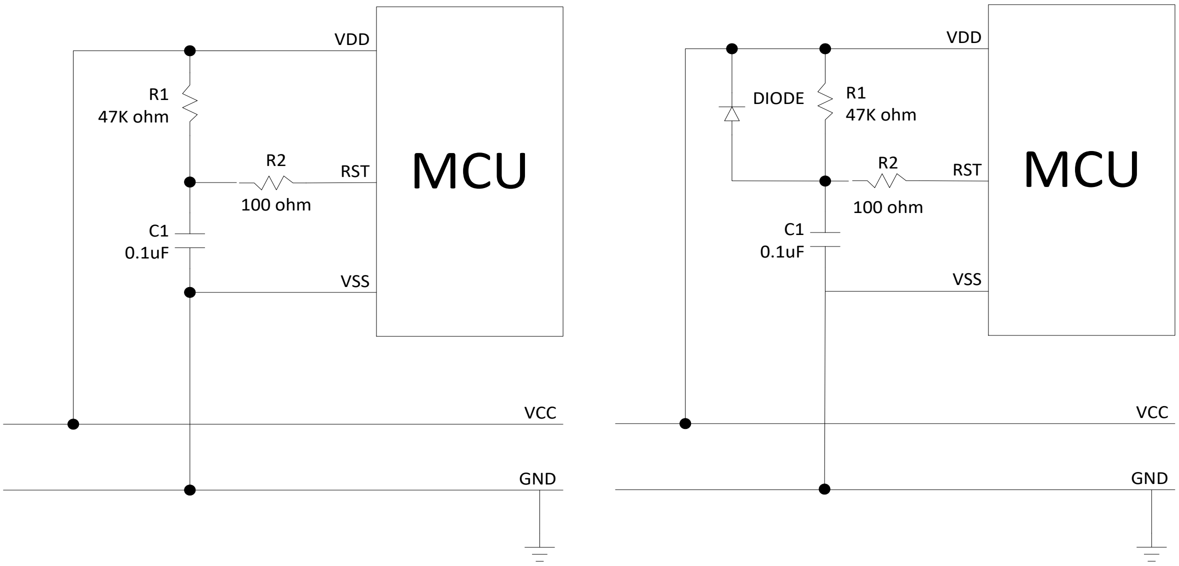
Disable mode suspends its internal timer at all CPU modes, and the watchdog would not trigger in this condition.

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.

```
SRST = 0x01;
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

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

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

WDTR Register (0x86)

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

8 System Clock and Power Management

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

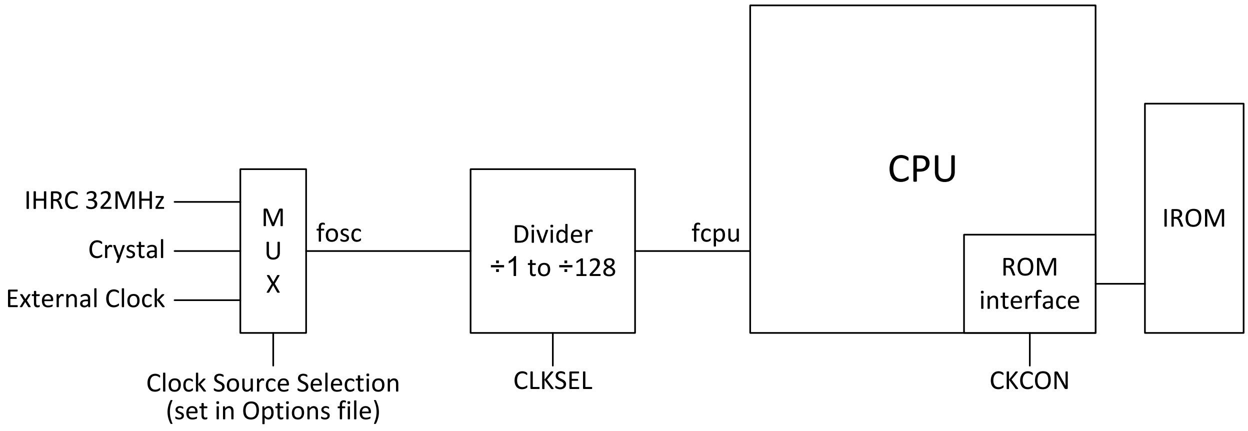
The normal mode means that CPU and peripheral functions are under normally execution. The system clock is based on the combination of source selection, clock divider, and program memory wait state. IDLE mode is the situation that temporarily suspends CPU clock and its execution, yet it remains peripherals’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

8.1 System Clock

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

Subsequently, the selected clock source (fosc) is divided by 1 to 128 times which is controlled by CLKSEL register. The CPU input the divided clock as its operation base (named fcpu). Applying CLKSEL’s setting when CLKCMD register be written 0x69.

```
CKCON = 0x70;     // For change safely the system clock
CLKSEL = 0x05;    // Set fcpu = fosc / 4
CLKCMD = 0x69;    // Apply CLKSEL’s setting
CKCON = 0x00;    // IROM fetch = fcpu / 1
```



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

✷ ***Note: For user develop program in C language or assembly, the first line of the program "must be set" CLKSEL= 0x07~0x00, CLKMD= 0x69 and then set CKCON= 0x00~0x70, this priority cannot be modified.***

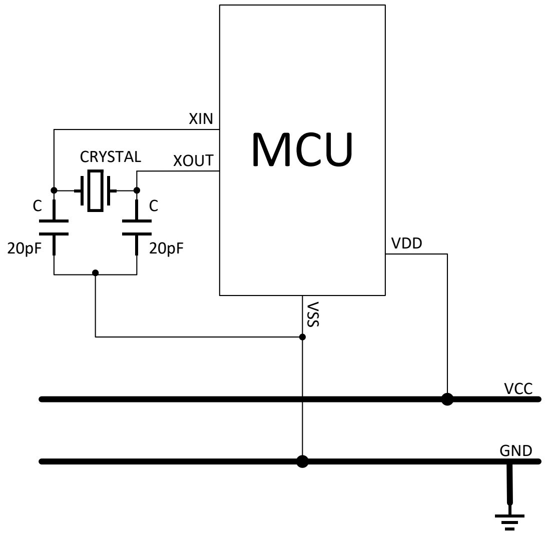
System clock rate and program memory extended cycle limitation as follows.

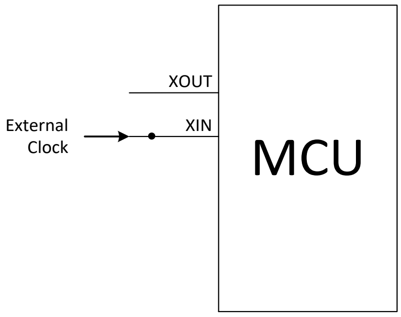
Code Option CPU Clock Source	Fcpu = CLKSEL[2:0]	IROM Fetch = CKCON[6:4]
IHRC 32M IHRC 32M with RTC External Clock (16-32MHz)	**Only Support** 000 = fosc / 128 001 = fosc / 64 010 = fosc / 32 011 = fosc / 16 100 = fosc / 8 101 = fosc / 4	**Support** **000 = fcpu / 1 => Recommend!** 001 = fcpu / 2 010 = fcpu / 3 011 = fcpu / 4 100 = fcpu / 5 101 = fcpu / 6 110 = fcpu / 7 111 = fcpu / 8
X'tal 12M (Crystal 8-16MHz) External Clock (8-16MHz)	**Only Support** 000 = fosc / 128 001 = fosc / 64 010 = fosc / 32 011 = fosc / 16 100 = fosc / 8 101 = fosc / 4 110 = fosc / 2	
X'tal 12M (Crystal 4-8MHz) X'tal 4M (Crystal 1-4MHz) External Clock (1-8MHz)	**Support** 000 = fosc / 128 001 = fosc / 64 010 = fosc / 32 011 = fosc / 16 100 = fosc / 8 101 = fosc / 4 110 = fosc / 2 111 = fosc / 1	

8.2 High Speed Clock

High-speed clock has internal and external two-type. The external high-speed clock includes 4MHz, 12MHz crystal/ceramic and external clock input mode. The internal high-speed oscillator is 32MHz RC type. These high-speed oscillators are selected by *SN8F5713_OPTIONS.A51*.

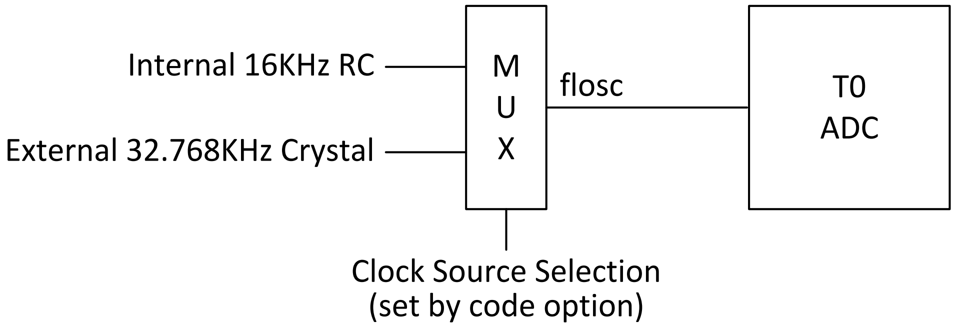
- IHRC 32M: The system high-speed clock source is internal high-speed 32MHz RC type oscillator. In the mode, XIN and XOUT pins are bi-direction GPIO mode, and not to connect any external oscillator device.
- IHRC 32M with RTC: The system high-speed clock source is internal high-speed 32MHz RC type oscillator. The RTC clock source is external low-speed 32768Hz crystal. The XIN and XOUT pins are defined to drive external 32768Hz crystal and disables GPIO function.
- X’tal 12M: The system high-speed clock source is external high-speed crystal/ceramic. The oscillator bandwidth is 10MHz~16MHz and connected to XIN/XOUT pins with 20pF capacitors to ground.
- X’tal 4M: The system high-speed clock source is external high-speed crystal/resonator. The oscillator bandwidth is 1MHz~10MHz and connected to XIN/XOUT pins with 20pF capacitors to ground.
- External Clock: The system high-speed clock source is external clock input mode. The input signal only connects to XIN pin, and the XOUT pin is bi-direction GPIO mode.





8.3 Low Speed Clock

SN8F5713 supplies low speed clock (flosc) for specific functions, such as T0 and ADC. The flosc has two clock sources selection: internal 16KHz RC (ILRC) and external 32.768KHz crystal, which is controlled by code option IHRC 32M RTC. In IHRC 32M RTC mode, P0.0 and P0.1 pin switch to crystal mode to drive an off-chip 32.768KHz crystal. The crystal is connected to XIN/XOUT pins with 20pF capacitors to ground.



8.4 Power Management

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

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

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

- 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.

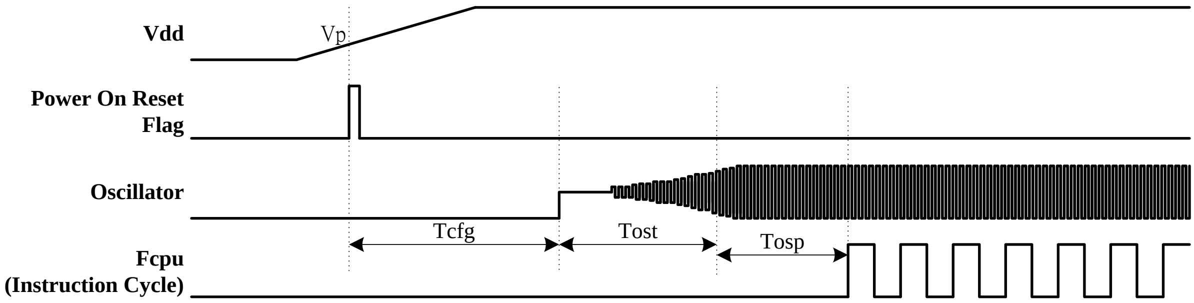
```
IDLE();
STOP();
```

SN8F5713 build in STWK bit (the 0th bit in SYSMOD register) to enable or disable flosc clock in STOP mode. If STWK=0, both fosc and flosc are suspended in STOP mode. If STWK=1, flosc clock keeps running in STOP mode.

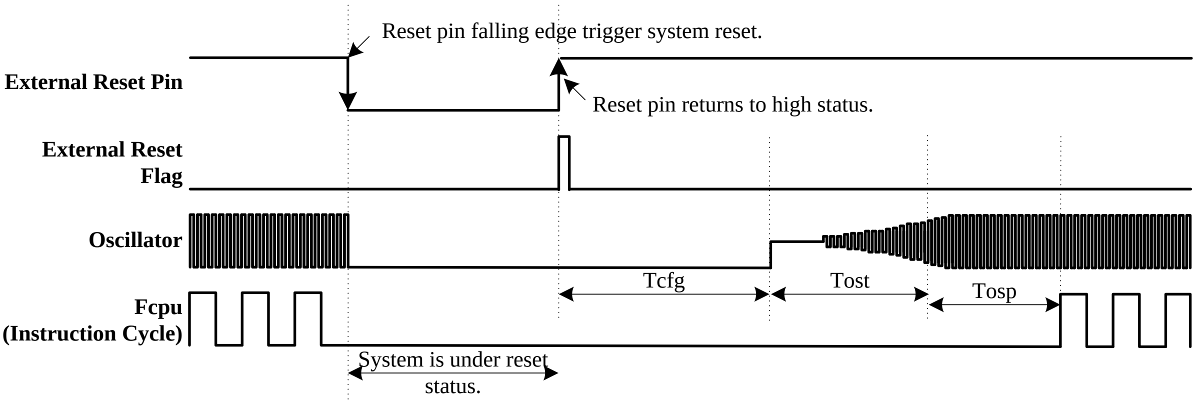
8.5 System Clock Timing

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

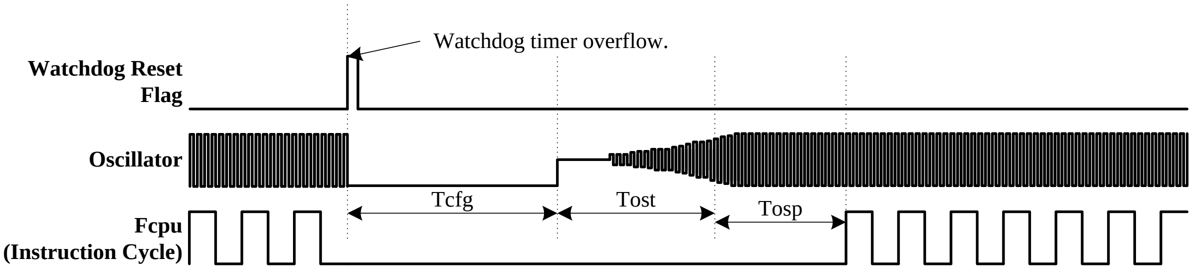
- Power On Reset Timing



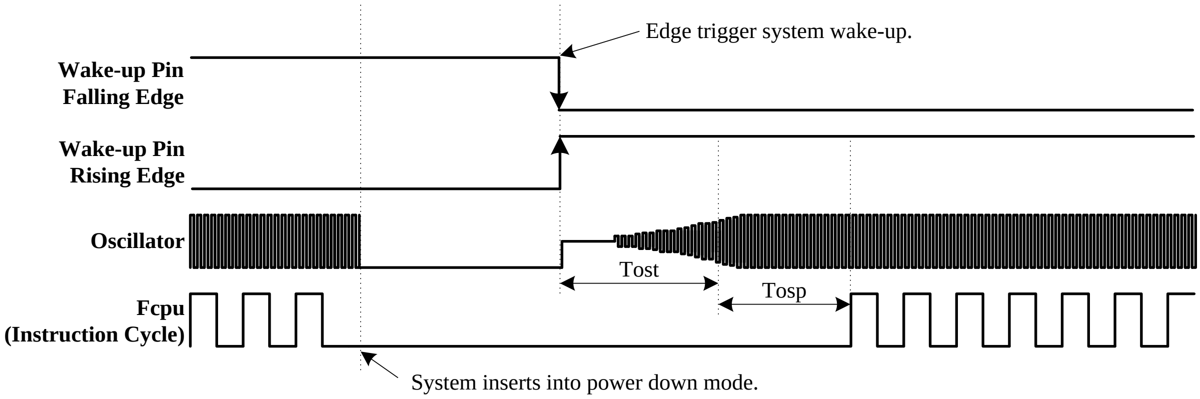
- External Reset Pin Reset Timing


Reset pin falling edge trigger system reset.
External Reset Pin
Reset pin returns to high status.
External Reset Flag
Oscillator
Tcfg
Tost
Tosp
Fcpu (Instruction Cycle)
System is under reset status.

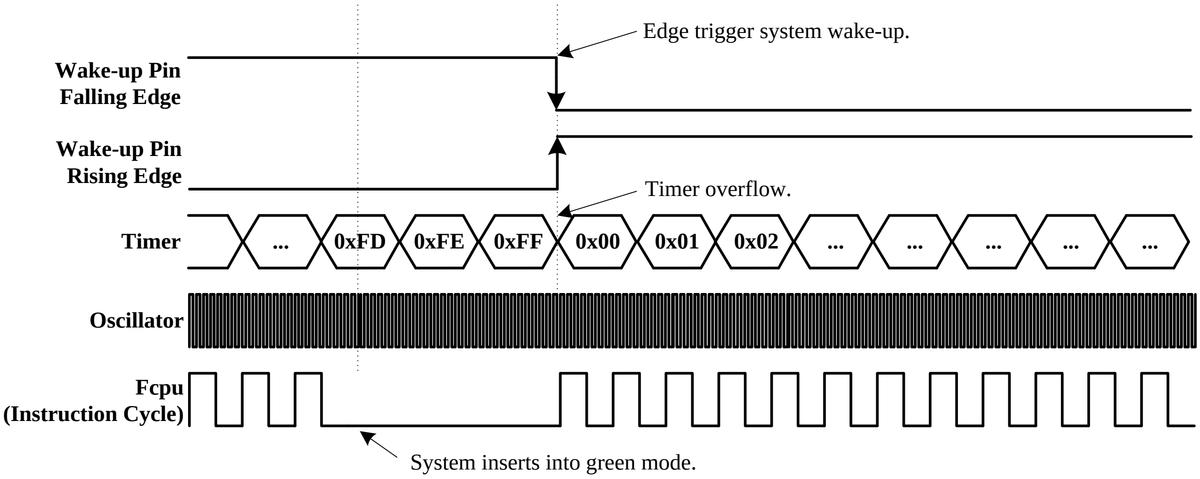
- Watchdog Reset Timing



- Power Down Mode Wake-up Timing

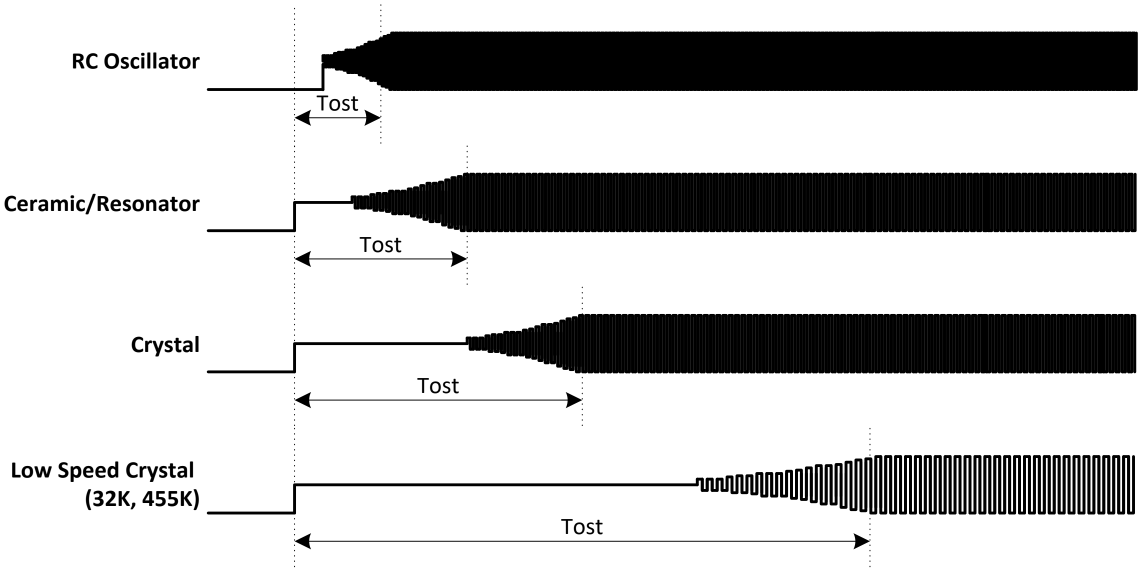


- Idle Mode Wake-up Timing



- 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.



8.6 System Clock and Power Management Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CKCON	-	PWSC2	PWSC1	PWSC0	-	-	-	-
CLKSEL	-	-	-	-	-	CLKSEL2	CLKSEL1	CLKSEL0
CLKCMD	CMD7	CMD6	CMD5	CMD4	CMD3	CMD2	CMD1	CMD0
PCON	SMOD	-	-	-	P2SEL	GF0	STOP	IDLE
P1W	P17W	P16W	P15W	P14W	P13W	P12W	P11W	P10W
SYSMOD	-	-	-	-	-	-	BIASEN	STWK

CKCON Register (0x8E)

Bit	Field	Type	Initial	Description
7	Reserved	R	0	
6..4	PWSC[2:0]	R/W	111	Extended cycle(s) applied to reading program memory 000: non 001: 1 cycle 010: 2 cycles 011: 3 cycles 100: 4 cycles 101: 5 cycles 110: 6 cycles 111: 7 cycles
3..0	Reserved	R	0	

CLKCMD Register (0xE6)

Bit	Field	Type	Initial	Description
7..0	CMD[7:0]	W	0x00	Writing 0x69 to apply CLKSEL’s setting.

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

CLKSEL Register (0xE5)

Bit	Field	Type	Initial	Description
7..3	Reserved	R	0x00	
2..0	CLKSEL[2:0]	R/W	111	CLKSEL would be applied by writing CLKCMD. 000: fcpu = fosc / 128 001: fcpu = fosc / 64 010: fcpu = fosc / 32 011: fcpu = fosc / 16 100: fcpu = fosc / 8 101: fcpu = fosc / 4 110: fcpu = fosc / 2 111: fcpu = fosc / 1

PCON Register (0x87)

Bit	Field	Type	Initial	Description
7				Refer to other chapter(s)
6..4	Reserved	R	0x00	
3	P2SEL	R/W	1	High-order address byte configuration bit. Chooses the higher byte of address (“XRAM [15:8]”) during MOVX @Ri operations 0: The “XRAM[15:8]” = “P2REG”. The “P2REG” is the contents of Port2 output register. 1: The “XRAM[15:8]” = 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

SYSMOD Register (0xCE)

Bit	Field	Type	Initial	Description
7..2	Reserved	R	0x00	
1				Refer to other chapter(s)
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)

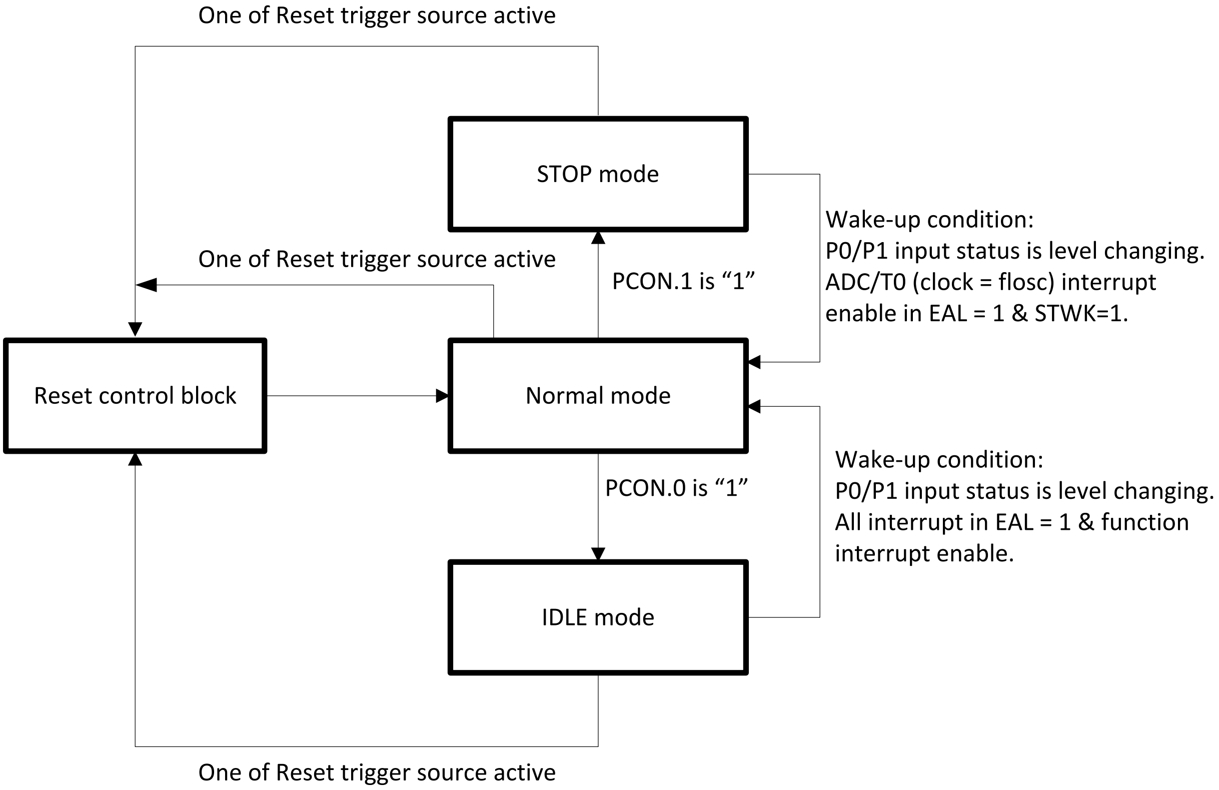

One of Reset trigger source active
STOP mode
Wake-up condition:
P0/P1 input status is level changing.
ADC/T0 (clock = flosc) interrupt
enable in EAL = 1 & STWK=1.
One of Reset trigger source active
PCON.1 is "1"
Reset control block
Normal mode
PCON.0 is "1"
Wake-up condition:
P0/P1 input status is level changing.
All interrupt in EAL = 1 & function
interrupt enable.
IDLE mode
One of Reset trigger source active

Table 9-1 The operating mode clock control

Operating Mode	Normal Mode	IDLE Mode	STOP Mode
IHRC	IHRC, IHRC RTC: Running Ext. OSC: Disable	IHRC, IHRC RTC: Running Ext. OSC: Disable	Stop
ILRC	Running	Running	STWK=1: Running Watchdog always: Running
Ext. OSC	IHRC: Disable IHRC RTC, Ext. OSC : Running	IHRC: Disable IHRC RTC, Ext. OSC : Running	Stop
CPU instruction	Executing	Stop	Stop
Timer 0 (Timer, Event counter)	Active by TR0	Active by TR0	Inactive
Timer 1 (Timer, Event counter)	Active by TR1	Active by TR1	Inactive
PWM1	Active as enable	Active as enable	Inactive
I2C/UART	Active as enable	Active as enable	Inactive
ADC	Active as enable	Active as enable	Active as ADC clock source is Flosc (ILRC or Ext. 32kHz) & STWK=1
Watchdog timer	By Watchdog Code option	By Watchdog Code option	By Watchdog Code option
Internal interrupt	All active	All active	All inactive
External interrupt	All active	All active	All inactive
Wakeup source	-	P0, P1, Reset, All interrupt in EAL = 1 & function interrupt enable	P0, P1, Reset, ADC/T0 enable & clock source is Flosc (ILRC or Ext. 32kHz) & STWK=1 & function interrupt enable.

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

9.1 Normal Mode

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

- The program is executed, and full functions are controllable.
- The system rate is high speed.
- The high speed oscillator and internal low speed RC type oscillator 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 ADC/T0 interrupt when ADC/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 external high speed oscillator, 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 ADC/T0 interrupt when ADC/T0 clock source is Flosc (ILRC or Ext. 32kHz) 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 2048 external high-speed oscillator clocks + 4 internal low-speed oscillator clocks and 64 internal high-speed oscillator clocks + 4 internal low-speed oscillator clocks as the wakeup time to stable the oscillator circuit. After the wakeup time, the system goes into the normal mode.

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

The Wakeup time = 1/Fosc * 2048 (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 * 2048 + 1/Flosc * 4 = 0.762 ms (Fosc = 4MHz, Flosc = 16KHz)

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

The value of the internal high clock oscillator RC type wakeup time is as the following.
The Wakeup time = 1/Fosc * 64 (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. The Port 0 and Port 1 have wakeup function. Port 0 wakeup functions always enables, but the Port 1 is controlled by the P1W register.

P1W Register (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 (3 external and 7 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
INT1	EX1	IE1	Automatically	2 / 0x0083
INT2	EX2	IE2	Automatically	3 / 0x0043
PWM1	EPWM1	PWM1F	Automatically	4 / 0x000B
UART TX	EU0TX	TI0	By firmware	5 / 0x0013
I2C	EI2C	SI	By firmware	6 / 0x0093
UART RX	EU0RX	RI0	By firmware	7 / 0x0053
ADC	EADC	ADCF	Automatically	8 / 0x001B
Timer 1	ET1	TF1	Automatically	9 / 0x00A3
Timer 0	ET0	TF0	Automatically	10 / 0x0063

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 – 0xA3) 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.

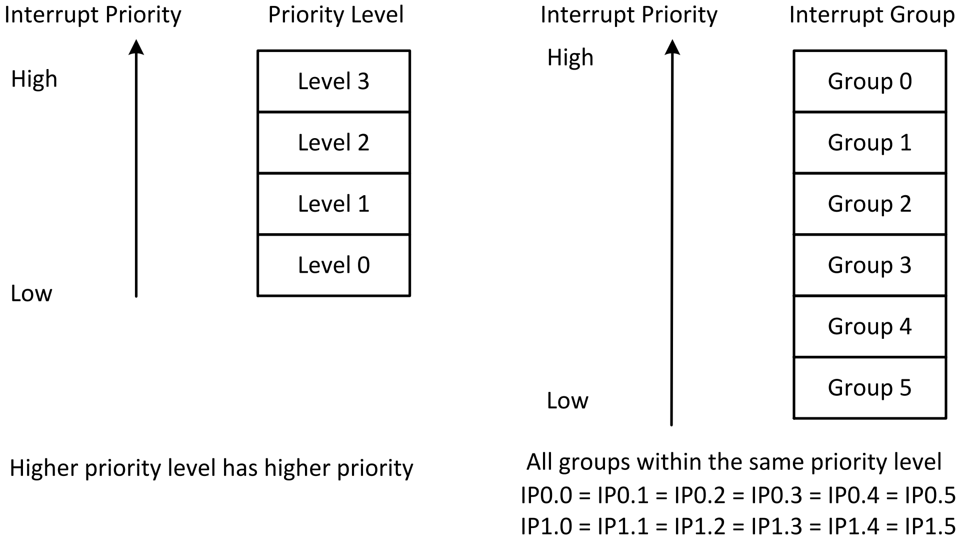
Priority Level	IP1.x	IP0.x
Level 0	0	0
Level 1	0	1
Level 2	1	0
Level 3	1	1

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	INT1	INT2	-
Group 1	PWM1	-	-	-
Group 2	UART TX	I2C	UART RX	-
Group 3	ADC	-	-	-
Group 4	T1	T0	-	-
Group 5	-	-	-	-

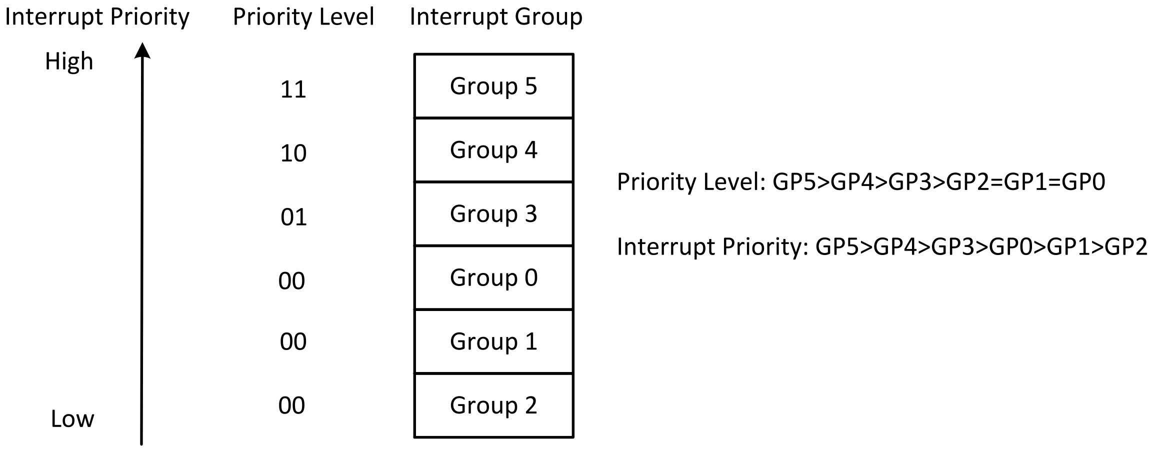
When more than one interrupt request occur, the highest priority request must be executed first. Choose the highest priority request according natural priority and priority level. The steps are as the following:

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.


Interrupt Priority
Priority Level
High
Low
Level 3
Level 2
Level 1
Level 0
Higher priority level has higher priority
Interrupt Priority
Interrupt Group
High
Low
Group 0
Group 1
Group 2
Group 3
Group 4
Group 5
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
```



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	EI2C	EU0RX	EU0TX	ET1	EX1	ET0	EX0
IEN2	-	-	-	-	EPWM1	EX2	-	EADC
IRCON2	-	-	-	-	PWM1F	IE2	-	ADCF
TCON	TF1	TR1	TF0	TR0	IE1	-	IE0	-
S0CON	SM0	SM1	SM20	REN0	TB80	RB80	TI0	RI0
I2CCON	CR2	ENS1	STA	STO	SI	AA	CR1	CR0

IEN0 Register (0XA8)

Bit	Field	Type	Initial	Description
7	EAL	R/W	0	Enable all interrupt control bit. 0: Disable all interrupt function. 1: Enable all interrupt function.
6	EI2C	R/W	0	I2C interrupt control bit. 0: Disable I2C interrupt function. 1: Enable I2C interrupt function.
5	EU0RX	R/W	0	UART RX interrupt control bit. 0: Disable UART RX interrupt function. 1: Enable UART RX interrupt function.
4	EU0TX	R/W	0	UART TX interrupt control bit. 0: Disable UART TX interrupt function. 1: Enable UART TX interrupt function.
3	ET1	R/W	0	T1 timer interrupt control bit. 0: Disable T1 interrupt function. 1: Enable T1 interrupt function.
2	EX1	R/W	0	External P0.6 interrupt (INT1) control bit. 0: Disable INT1 interrupt function. 1: Enable INT1 interrupt function.
1	ET0	R/W	0	T0 timer interrupt control bit. 0: Disable T0 interrupt function. 1: Enable T0 interrupt function
0	EX0	R/W	0	External P0.3 interrupt (INT0) control bit. 0: Disable INT0 interrupt function. 1: Enable INT0 interrupt function.

IEN2 Register (0X9A)

Bit	Field	Type	Initial	Description
3	EPWM1	R/W	0	PWM1 interrupt control bit. 0 = Disable PWM1 interrupt function. 1 = Enable PWM1 interrupt function.
2	EX2	R/W	0	External P0.7 interrupt (INT2) control bit. 0: Disable INT2 interrupt function. 1: Enable INT2 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	

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

IRCON2 Register (0XBF)

Bit	Field	Type	Initial	Description
3	PWM1F	R/W	0	PWM1 interrupt request flag. 0: None PWM1 interrupt request 1: PWM1 interrupt request.
2	IE2	R/W	0	External P0.7 interrupt (INT2) request flag 0: None INT2 interrupt request. 1: INT2 interrupt request.
0	ADCF	R/W	0	ADC interrupt request flag. 0: None ADC interrupt request. 1: ADC interrupt request.
Else	Reserved	R	0	

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)

10.4 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       0X0043
                JMP       ISR_INT2
                …
                ORG       0X0063
                JMP       ISR_T0
                …
                ORG       0X00ECH
START:                                  ; 00ECH, 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_INT2                                ;
                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 MDU

The multiplication division unit is an on-chip arithmetic co-processor which enables the microcontroller to perform additional extended arithmetic operations. This unit provides 32-bit unsigned division, 16-bit unsigned multiplication, shift and normalize operations. These operations are identified by the different sequences of writing MD0 to MD5 registers.

11.1 Multiplication (16-bit x 16-bit)

The elements of a multiplication include three parts: multiplicand, multiplier and product. To start a multiplication requires following writing sequence: MD0 (low byte of multiplicand), MD4 (low byte of multiplier), MD1 (high byte of multiplicand), and MD5 (high byte of multiplier).

By the end of writing MD5 register, the multiplication is automatically started and takes 11 CPU cycles for its operation. The product of this term operation would be available to read by a specific sequence: MD0 (LSB), MD1, MD2, and MD3 (MSB) registers.

11.2 Division (32-bit/16-bit and 16-bit/16-bit)

The MDU supports two kind of division: 32-bit by 16-bit, and 16-bit by 16-bit. The first operation takes 17 CPU cycles to compute, whereas the second one takes 9 cycles only.

A 32-bit division started by a specific sequence of writing registers: MD0, MD1, MD2, MD3, MD4, and MD5. In this case, the 32-bit dividend is expected to store in MD3 (most significant bit) to MD0 registers, and 16-bit divisor is stored in MD5 and MD4 registers (MSB in MD5 register).

A 16-bit division operation cooperates with four registers only. The 16-bit dividend is stored in MD1 and MD0 registers, and the 16-bit divisor is stored in MD5 and MD4 registers (MD1 and MD5 for most signification bit). The appropriate performing sequence is ‘MD0, MD1, MD4, and MD5.’

The MDU starts computing from MD5 register is written. It spends 9 or 17 CPU cycles, depends on the length of dividend, before the outcome is generated. The quotient is stored in MD3 to MD0 registers for 32-bit division, and MD1 to MD0 registers for 16-bit division (LSB in MD0 register). The reminder would be placed in MD5 (MSB) and MD4 registers no matter which division is performed. However, reading MD5 register must be the last operation to indicate the full division is completed.

11.3 Shifting and Normalizing

The shifting and normalizing operations rotate the 32-bit registers (MD3 to MD0, MSB in MD3) for a certain or uncertain time.

In shift operation, the 32-bit unsigned integer is shifted left or right by a specified number of bits. The direction and shifting number is specified in ARCON register. A shift operation takes 3 to 18 CPU cycles depends on the shift time.

In normalizing operation, the 32-bit unsigned integer would be shifted left repeatedly until the most significant bit (7th bit of MD3 register) is 1. A normalizing operation takes 4 to 19 CPU cycles depends on the actual shift time.

Both shifting and normalizing operations are started by proper sequence of writing registers: MD0, MD1, MD2, MD3, and finally ARCON register. The result would be place in MD0 to MD3 registers which should be read in the sequence of MD0, MD1, MD2, and MD3.

11.4 Cooperate with Keil C51

Because Keil C51 supports both of hardware and software multiplication/division operators, a command line '#pragma mdu_r515' is required in C to enable the hardware MDU functionality for higher performance. Subsequently, Keil C51 would compile mathematic operators with MDU support.

```
#include <SN8F5713.H>
#pragma mdu_r515          //Keil C51 MDU command line
```

11.5 The Error Flag (MDEF)

The "MDEF" error flag indicates an improperly performed operation (when one of the arithmetic operations has been restarted or interrupted by a new operation). The error flag mechanism is automatically enabled with the first write operation to "MD0" and disabled with the final read instruction from "MD3" (multiplication or shift/normalize) or "MD5" (division) in phase three.

The error flag is set when:
There is a write access to 'MDx' registers (any of 'MD0' to 'MD5' and ARCON) during phase two of MDU operation (restart or calculations interrupting)
There is a read access to one of MDx registers during phase two of MDU operation when the error flag mechanism is enabled. In such condition error flag is set but the calculation is not interrupted.
The error flag is reset only after read access to "ARCON" register. The error flag is read only.

11.6 The Overflow Flag (MDOV)

The MDOV overflow flag is set when one of the following conditions occurs:

Division by zero
Multiplication with a result greater than 0000 FFFFh
Start of normalizing if the most significant bit of MD3 is set (MD3.7= 1)
Any operation of the MDU that does not match the above conditions clears the overflow flag. Note that the overflow flag is exclusively controlled by hardware. It cannot be written.

11.7 MDU Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MD0	MD07	MD06	MD05	MD04	MD03	MD02	MD01	MD00
MD1	MD17	MD16	MD15	MD14	MD13	MD12	MD11	MD10
MD2	MD27	MD26	MD25	MD24	MD23	MD22	MD21	MD20
MD3	MD37	MD36	MD35	MD34	MD33	MD32	MD31	MD30
MD4	MD47	MD46	MD45	MD44	MD43	MD42	MD41	MD40
MD5	MD57	MD56	MD55	MD54	MD53	MD52	MD51	MD50
ARCON	MDEF	MDOV	SLR	SC4	SC3	SC2	SC1	SC0

MD Registers (MD0 – MD5: 0xE9 – 0xEE)

Bit	Field	Type	Initial	Description
7..0	MD[7:0]	R/W	0x00	Multiplication/Division Registers

ARCON Register (0xEF)

Bit	Field	Type	Initial	Description
7	MDEF	R/W	0	MDU error flag MDEF Indicates an improperly performed operation (when one of the arithmetic operations has been restarted or interrupted by a new operation).
6	MDOV	R/W	0	MDU overflow flag Overflow occurrence in the MDU operation.
5	SLR	R/W	0	Shift direction 0: Shift left operation 1: Shift right operation
4..0	SC[4:0]	R/W	0x00	Shift counter Write 0x00: Perform normalizing. The actual shift time would be readable after operation. Write else values: Specify the times of shift operation.

12 GPIO

The microcontroller has up to 21 bidirectional general purpose I/O pin (GPIO). Unlike the original 8051 only has open-drain output, SN8F5713 builds in push-pull output structure to improve its driving performance.

12.1 Input and Output Control

The input and output direction control is configurable through P0M to P2M registers. These bits specify each pin that is either input mode or output mode.

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0M	P07M	P06M	P05M	P04M	P03M	P02M	P01M	P00M
P1M	P17M	P16M	P15M	P14M	P13M	P12M	P11M	P10M
P2M	-	-	P25M	P24M	P23M	P22M	P21M	P20M
P0OC	-	-	-	-	P17OC	P16OC	P03OC	P02OC

P0M: 0xF9

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

✸ ***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.***

P1M: 0xFA

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

P2M: 0xFB

Bit	Field	Type	Initial	Description
7..6	Reserved	R	0	
5	P25M	R/W	0	Mode selection of P2.5 0: Input mode 1: Output mode
4	P24M	R/W	0	Mode selection of P2.4 0: Input mode 1: Output mode

3	P23M	R/W	0	Mode selection of P2.3 0: Input mode 1: Output mode
2	P24M	R/W	0	Mode selection of P2.2 0: Input mode 1: Output mode
1	P23M	R/W	0	Mode selection of P2.1 0: Input mode 1: Output mode
0	P23M	R/W	0	Mode selection of P2.0 0: Input mode 1: Output mode

P0OC Register (0xE4)

Bit	Field	Type	Initial	Description
Else	Reserved	R	0	
3	P17OC	R/W	0	P1.7 open-drain control bit 0: Disable 1: Enable
2	P16OC	R/W	0	P1.6 open-drain control bit 0: Disable 1: Enable
1	P03OC	R/W	0	P0.3 open-drain control bit 0: Disable 1: Enable
0	P02OC	R/W	0	P0.2 open-drain control bit 0: Disable 1: Enable

12.2 Input Data and Output Data

By a read operation from any registers of P0 to P2, 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 P2 register value would be latched immediately, yet the value would be outputted until the mapped P0M – P2M is set to output mode. If the pin is currently in output mode, any value set to P0 to P2 register would be presented on the pin immediately.

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0	P07	P06	P05	P04	P03	P02	P01	P00
P1	P17	P16	P15	P14	P13	P12	P11	P10
P2	-	-	P25	P24	P23	P22	P21	P20

P0: 0x80

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

P1: 0x90

Bit	Field	Type	Initial	Description
7	P17	R/W	1	Read: P1.7 pin's logic level Write 1/0: Output logic high or low (applied if P17M = 1)

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

P2: 0xA0

Bit	Field	Type	Initial	Description
7..6	Reserved	R	0	
5	P25	R/W	1	Read: P2.5 pin’s logic level Write 1/0: Output logic high or low (applied if P25M = 1)
4	P24	R/W	1	Read: P2.4 pin’s logic level Write 1/0: Output logic high or low (applied if P24M = 1)
3	P23	R/W	1	Read: P2.3 pin’s logic level Write 1/0: Output logic high or low (applied if P23M = 1)
2	P22	R/W	1	Read: P2.2 pin’s logic level Write 1/0: Output logic high or low (applied if P22M = 1)
1	P21	R/W	1	Read: P2.1 pin’s logic level Write 1/0: Output logic high or low (applied if P21M = 1)
0	P20	R/W	1	Read: P2.0 pin’s logic level Write 1/0: Output logic high or low (applied if P20M = 1)

12.3 On-chip Pull-up Resisters

The P0UR to P2UR registers are mapped to each pins' internal 100k ohm (in typical value) pull-up resister.

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0UR	P07UR	P06UR	P05UR	P04UR	P03UR	P02UR	P01UR	P00UR
P1UR	P17UR	P16UR	P15UR	P14UR	P13UR	P12UR	P11UR	P10UR
P2UR	-	-	P25UR	P24UR	P23UR	P22UR	P21UR	P20UR

P0UR: 0xF1

Bit	Field	Type	Initial	Description
7	P07UR	R/W	0	On-chip pull-up resister control of P0.7 0: Disable* 1: Enable
6	P06UR	R/W	0	On-chip pull-up resister control of P0.6 0: Disable* 1: Enable
5	P05UR	R/W	0	On-chip pull-up resister control of P0.5 0: Disable* 1: Enable
4	P04UR	R/W	0	On-chip pull-up resister control of P0.4 0: Disable* 1: Enable
3	P03UR	R/W	0	On-chip pull-up resister control of P0.3 0: Disable* 1: Enable
2	P02UR	R/W	0	On-chip pull-up resister control of P0.2 0: Disable* 1: Enable
1	P01UR	R/W	0	On-chip pull-up resister control of P0.1 0: Disable* 1: Enable
0	P00UR	R/W	0	On-chip pull-up resister control of P0.0 0: Disable* 1: Enable

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

P1UR: 0xF2

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

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

P2UR: 0xF3

Bit	Field	Type	Initial	Description
7..6	Reserved	R	0	
5	P25UR	R/W	0	On-chip pull-up resister control of P2.5 0: Disable* 1: Enable
4	P24UR	R/W	0	On-chip pull-up resister control of P2.4 0: Disable* 1: Enable
3	P23UR	R/W	0	On-chip pull-up resister control of P2.3

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

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

12.4 Pin Shared with Analog Function

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

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0CON	P0CON7	P0CON6	-	-	-	-	-	-
P1CON	P1CON7	P1CON6	P1CON5	P1CON4	P1CON3	P1CON2	P1CON1	P1CON0
P2CON	-	-	P2CON5	P2CON4	P2CON3	P2CON2	P2CON1	P2CON0
SYSMOD	-	-	-	-	-	-	BIASEN	STWK

P0CON: 0x9E, P1CON: 0x9F, P2CON: 0XD6, SYSMOD: 0XCE, P0BIAS: 0XF4, P1BIAS: 0XF5

P0CON Register (0x9E)

Bit	Field	Type	Initial	Description
7..6	P0CON[7:6]	R/W	0x00	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.

* P0CON [7:0] will configure related Port0 pin as pure analog input pin to avoid current leakage.

P1CON Register (0x9F)

Bit	Field	Type	Initial	Description
7..0	P1CON[7:0]	R/W	0x00	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.

* P1CON [7:0] will configure related Port1 pin as pure analog input pin to avoid current leakage.

P2CON Register (0XD6)

Bit	Field	Type	Initial	Description
4..0	P2CON[5:0]	R/W	0x0	P2 configuration control bit*. 0: P2 can be analog input pin or digital GPIO pin. 1: P2 is pure analog input pin and can't be a digital GPIO pin.

* P2CON [5:0] will configure related Port2 pin as pure analog input pin to avoid current leakage.

SYSMOD Register (0xCE)

Bit	Field	Type	Initial	Description
7..2	Reserved	R	0x00	
1	BIASEN	R/W	0	0: Disable 1/2*VDD bias voltage function. 1: Enable 1/2*VDD bias voltage function.
0				Refer to other chapter(s)

P0BIAS Register (0XF4)

Bit	Field	Type	Initial	Description
3..0	P0BIAS[3:0]	R/W	0x0	P0.0-P0.3 1/2*VDD bias voltage control bit. 0: Disable 1/2*VDD bias voltage output. 1: Enable 1/2*VDD bias voltage output.

P1BIAS Register (0XF5)

Bit	Field	Type	Initial	Description
3..0	P1BIAS[3:0]	R/W	0x0	P1.0-P1.3 1/2*VDD bias voltage control bit. 0: Disable 1/2*VDD bias voltage output. 1: Enable 1/2*VDD bias voltage output.

13 External Interrupt

INT0, INT1 and INT2 are external interrupt trigger sources. Build in edge trigger configuration function and edge direction is selected by PEDGE register. When both external interrupt (EX0/EX1/EX2) and global interrupt (EAL) are enabled, the external interrupt request flag (IE0/IE1/IE2) will be set to “1” as edge trigger event occurs. The program counter will jump to the interrupt vector (ORG 0x0003/0x0043/0x0083) and execute interrupt service routine. Interrupt request flag will be cleared by hardware before ISR is executed.

13.1 External Interrupt Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PEDGE	-	-	EX2G1	EX2G0	EX1G1	EX1G0	EX0G1	EX0G0
IEN0	EAL	EI2C	EU0RX	EU0TX	ET1	EX1	ET0	EX0
TCON	TF1	TR1	TF0	TR0	IE1	-	IE0	-
IEN2	-	-	-	-	EPWM1	EX2	-	EADC
IRCON2	-	-	-	-	PWM1F	IE2	-	ADCF

PEDGE Register (0X8F)

Bit	Field	Type	Initial	Description
5..4	EX2G[1:0]	R/W	10	External interrupt 2 trigger edge control register. 00: Reserved. 01: Rising edge trigger. 10: Falling edge trigger (default) 11: Both rising and falling edge trigger
3..2	EX1G[1:0]	R/W	10	External interrupt 1 trigger edge control register. 00: Reserved. 01: Rising edge trigger. 10: Falling edge trigger (default) 11: Both rising and falling edge trigger
1..0	EX0G[1:0]	R/W	10	External interrupt 0 trigger edge control register. 00: Reserved. 01: Rising edge trigger. 10: Falling edge trigger (default) 11: Both rising and falling edge trigger
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.
2	EX1	R/W	0	External P0.6 interrupt (INT1) control bit. 0: Disable INT1 interrupt function. 1: Enable INT1 interrupt function.
0	EX0	R/W	0	External P0.3 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
3	IE1	R/W	0	External P0.6 interrupt (INT1) request flag 0: None INT1 interrupt request. 1: INT1 interrupt request.
1	IE0	R/W	0	External P0.3 interrupt (INT0) request flag 0: None INT0 interrupt request. 1: INT0 interrupt request.
Else				Refer to other chapter(s)

IEN2 Register (0X9A)

Bit	Field	Type	Initial	Description
2	EX2	R/W	0	External P0.7 interrupt (INT2) control bit. 0: Disable INT2 interrupt function. 1: Enable INT2 interrupt function.
Else				Refer to other chapter(s)

IRCON2 Register (0XBF)

Bit	Field	Type	Initial	Description
2	IE2	R/W	0	External P0.7 interrupt (INT2) request flag 0: None INT2 interrupt request. 1: INT2 interrupt request.
Else				Refer to other chapter(s)

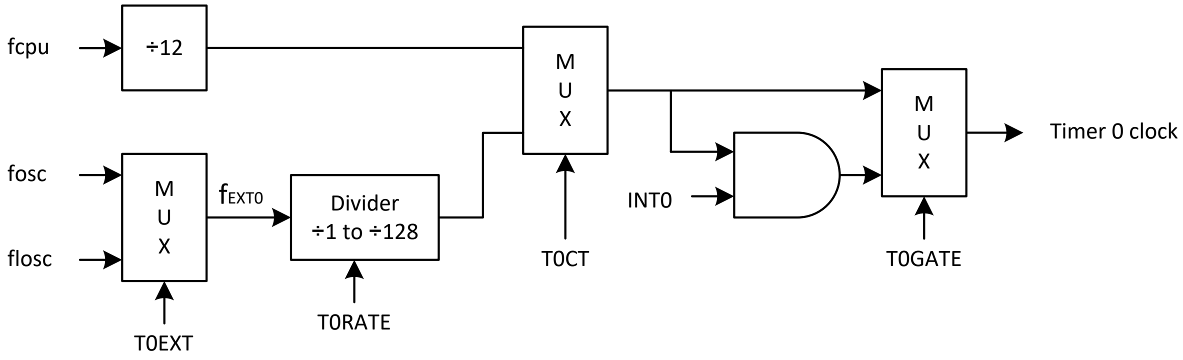
14 Timer 0 and Timer 1

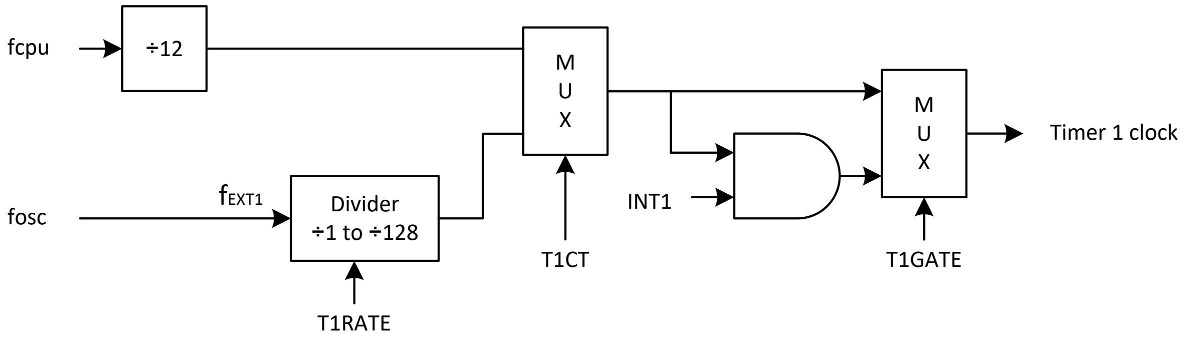
Timer 0 and Timer 1 are two independent binary up timers. Timer 0 has four different operation modes: (1) 13-bit up counting timer, (2) 16-bit up counting timer, (3) 8-bit up counting timer with specified reload value support, and (4) separated two 8-bit up counting timer. By contrast, Timer 1 has only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

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

14.1 Timer 0 and Timer 1 Clock Selection

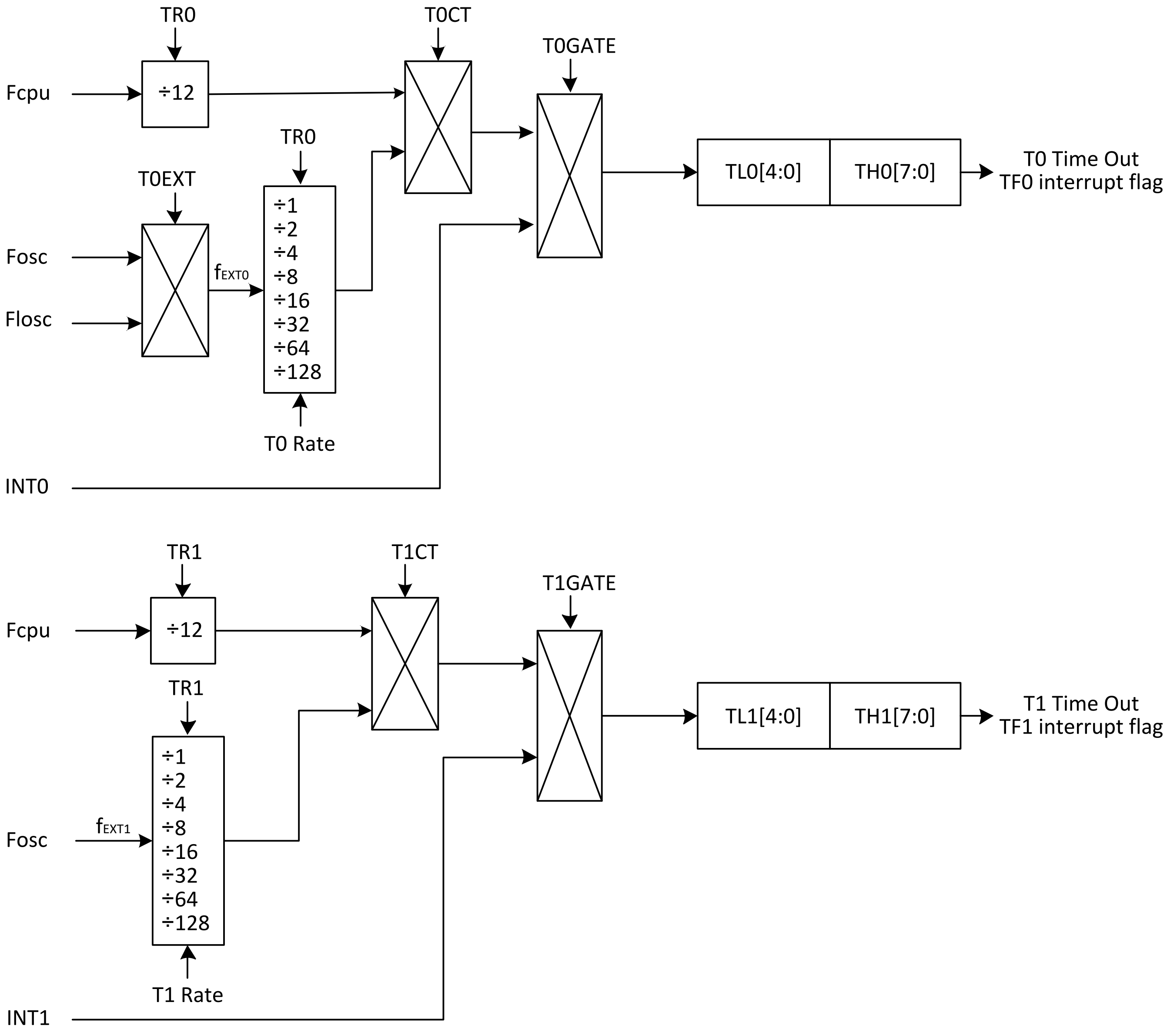
The figures below illustrate the clock selection circuit of Timer 0 and Timer 1. Timer 0 has three clock sources selection: fcpu, fosc, and flosc. All clock sources can be gated (pause) by INT0 pin if T0GATE is applied. Timer 1 clock sources selection: fcpu and fosc. All clock sources can be gated (pause)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





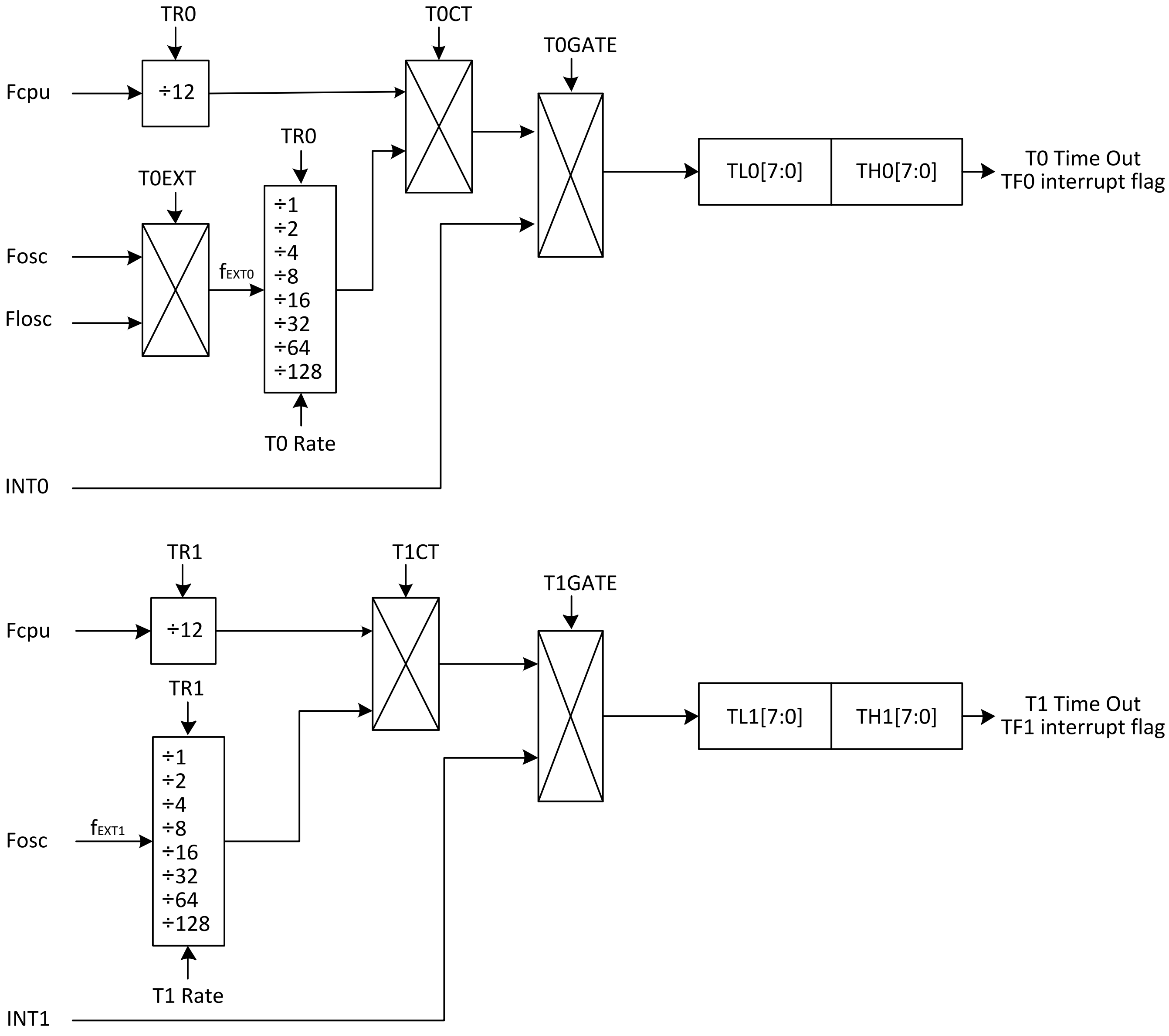
14.2 Mode 0: 13-bit Up Counting Timer

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



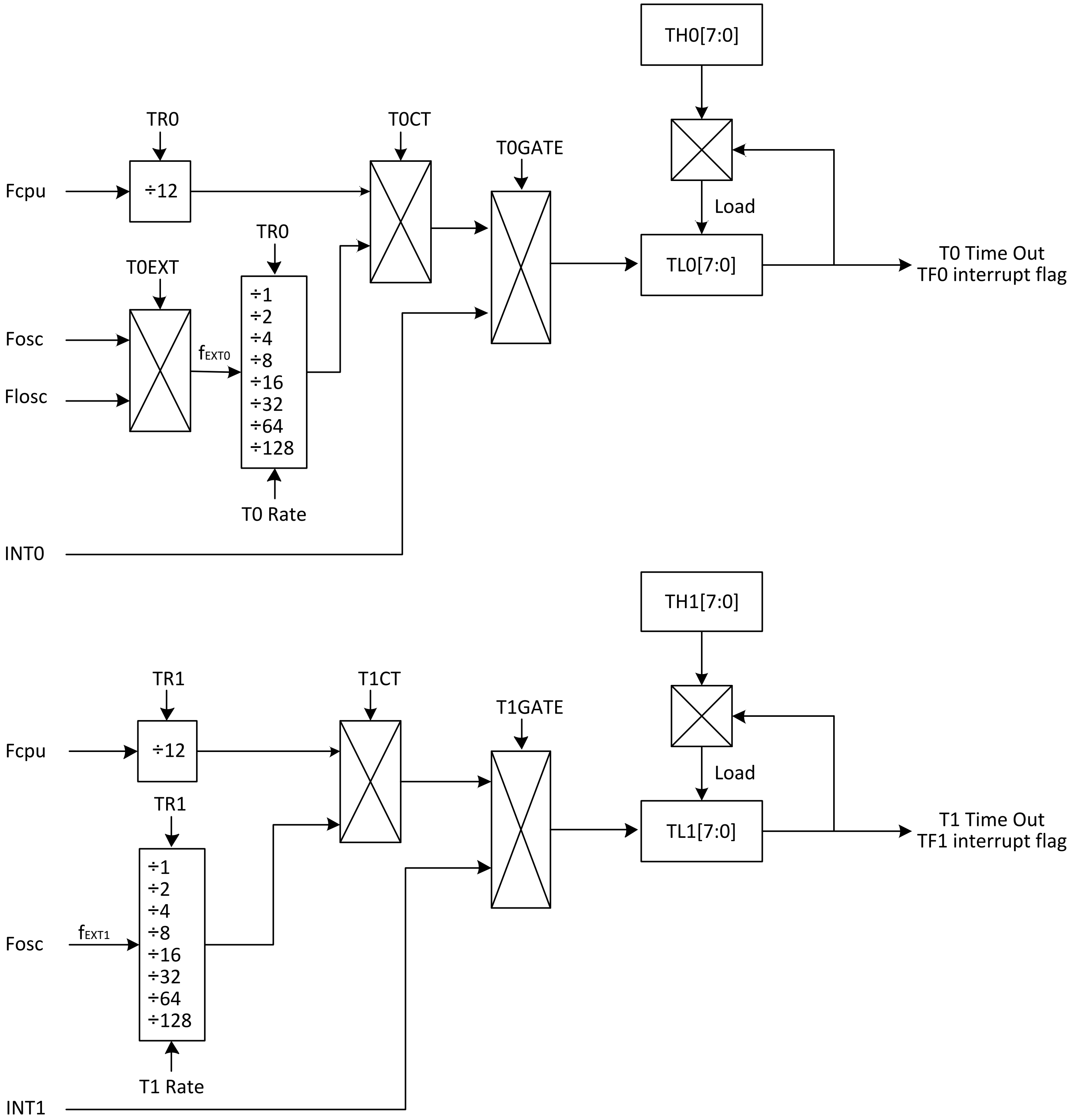
14.3 Mode 1: 16-bit Up Counting Timer

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



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

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

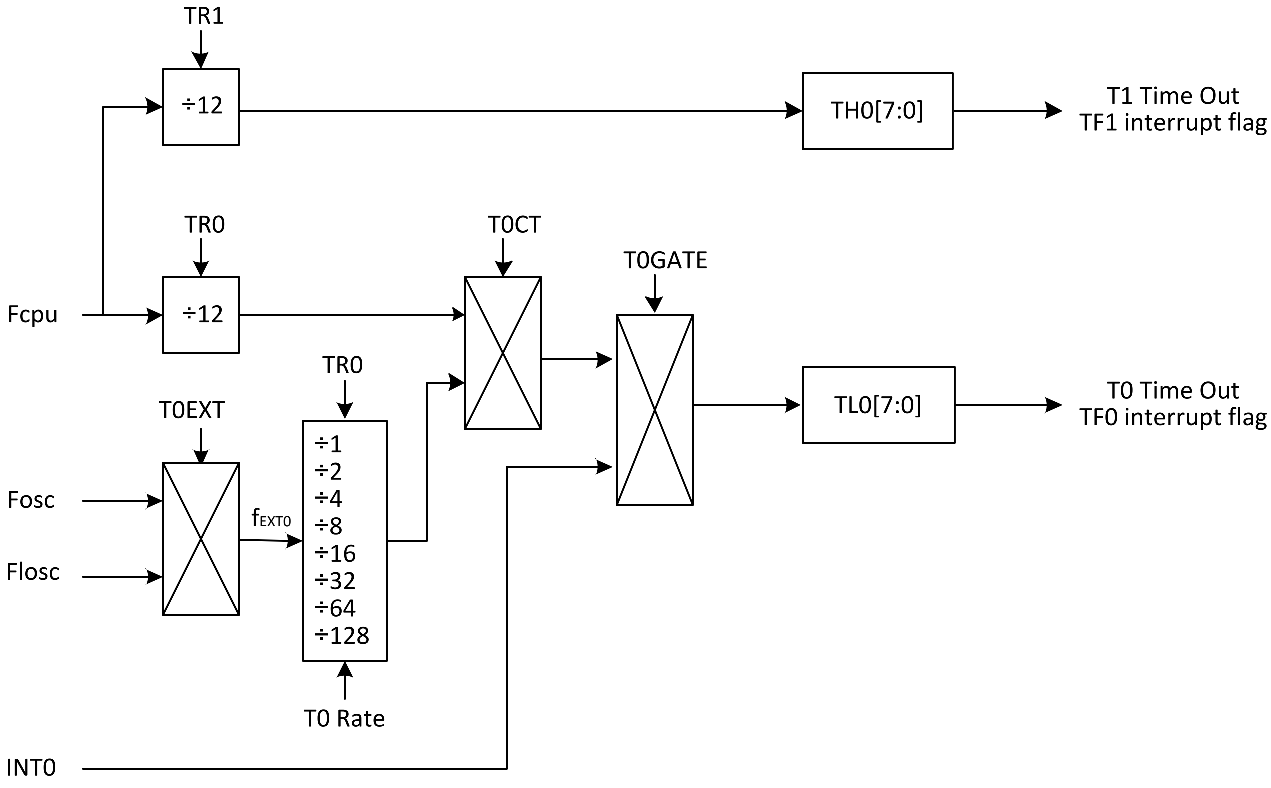


14.5 Mode 3 (Timer 0 only): Separated Two 8-bit Up Counting Timer

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

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

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



14.6 Timer 0 and Timer 1 Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IEN0	EAL	EI2C	EU0RX	EU0TX	ET1	EX1	ET0	EX0
TCON	TF1	TR1	TF0	TR0	IE1	-	IE0	-
TCON0	T0EXT	T0RATE2	T0RATE1	T0RATE0	-	T1RATE2	T1RATE1	T1RATE0
TMOD	T1GATE	T1CT	T1M1	T1M0	T0GATE	T0CT	T0M1	T0M0
TH0	TH07	TH06	TH05	TH04	TH03	TH02	TH01	TH00
TL0	TL07	TL06	TL05	TL04	TL03	TL02	TL01	TL00

TH1	TH17	TH16	TH15	TH14	TH13	TH12	TH11	TH10
TL1	TL17	TL16	TL15	TL14	TL13	TL12	TL11	TL10

IEN0 Register (0xA8)

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

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

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

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

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

TCON Register (0x88)

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

				1: Timer 0 has overflowed This bit can be cleared automatically by interrupt handler, or manually by firmware
4	TR0	R/W	0	Timer 0 function 0: Disable 1: Enable
3	IE1	R/W	0	Refer to INT1
2	Reserved	R	0	
1	IE0	R/W	0	Refer to INT0
0	Reserved	R	0	

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 0 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

TMOD Register (0x89)

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

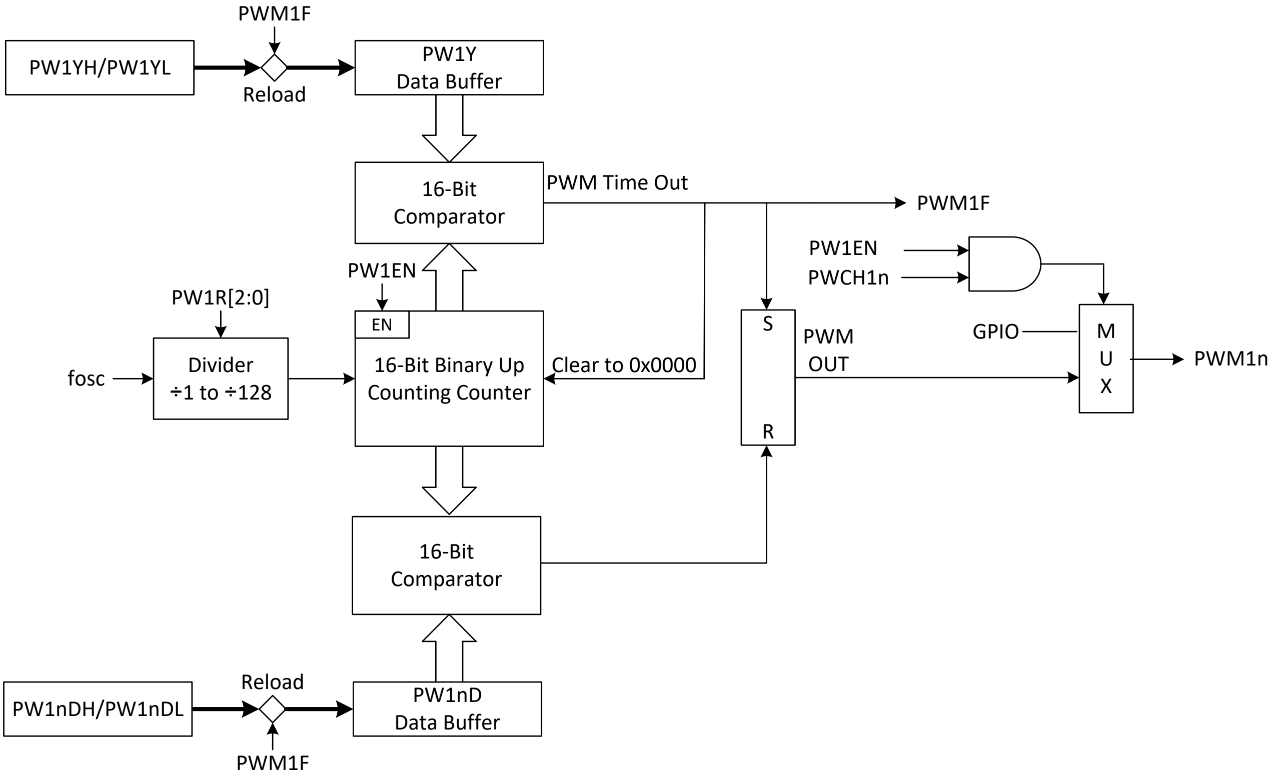
*(1) f_{EXT1} = fosc.

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

15 PWM

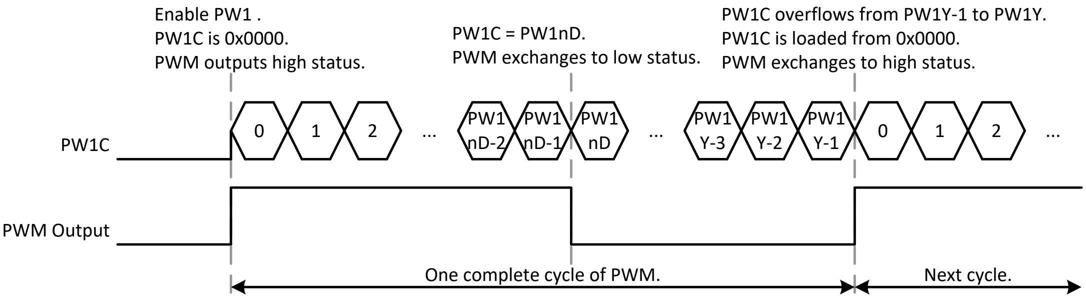
The PW1 timer is a 16-bit up counting timer and supports 8-channel general PWM function. By the counter reaches the up-boundary value (PW1Y), it clears its counter and triggers an interrupt signal. PWM's duty cycle is controlled by PW10D~PW17D register. Each PWM channel has its own duty control.

The PWM function has 8 programmable channels shared with GPIO pins and controlled by PWCH[7:0] bit. The output operation must be through enabled each bit/channel of PWCH[7:0] bits. The enabled PWM channel exchanges from GPIO to PWM output. When the PWCH[7:0] bits disables, the PWM channel returns to last status of GPIO mode. The PW1 timer build in IDLE Mode wake-up function if interrupt enable. When timer overflow occurs (counts from PW1Y-1 to PW1Y), PWM1F would be issued immediately which can read/write by firmware. PW1 interrupt function is controlled by EPWM1.



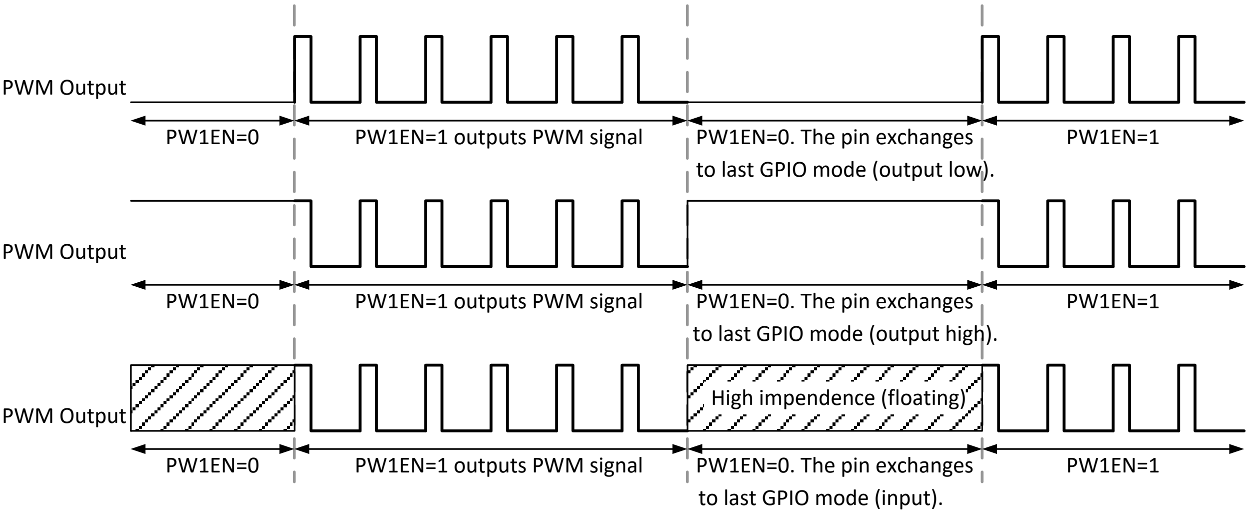
15.1 General PWM

PW1 timer builds in PWM function controlled by PW1EN and PWCH register. PWM10 - PWM17 are output pins. Those output pins are shared with GPIO pin controlled by PWCH[7:0] 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 counts from 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): (PW1nD-PW1Y)


PWM Output
PW1EN=0
PW1EN=1 outputs PWM signal
PW1EN=0. The pin exchanges to last GPIO mode (output low).
PW1EN=1
PWM Output
PW1EN=0
PW1EN=1 outputs PWM signal
PW1EN=0. The pin exchanges to last GPIO mode (output high).
PW1EN=1
PWM Output
High impendence (floating)
PW1EN=0
PW1EN=1 outputs PWM signal
PW1EN=0. The pin exchanges to last GPIO mode (input).
PW1EN=1

15.2 PWM Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW1M	PW1EN	PW1R2	PW1R1	PW1R0	-	-	-	-
PWCH	PWCH17	PWCH16	PWCH15	PWCH14	PWCH13	PWCH12	PWCH11	PWCH10
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
PW16DH	PW16D15	PW16D14	PW16D13	PW16D12	PW16D11	PW16D10	PW16D9	PW16D8
PW16DL	PW16D7	PW16D6	PW16D5	PW16D4	PW16D3	PW16D2	PW16D1	PW16D0
PW17DH	PW17D15	PW17D14	PW17D13	PW17D12	PW17D11	PW17D10	PW17D9	PW17D8
PW17DL	PW17D7	PW17D6	PW17D5	PW17D4	PW17D3	PW17D2	PW17D1	PW17D0
IEN0	EAL	EI2C	EU0RX	EU0TX	ET1	EX1	ET0	EX0
IEN2	-	-	-	-	EPWM1	EX2	-	EADC
IRCON2	-	-	-	-	PWM1F	IE2	-	ADCF

PWCH Register (0xB6)

Bit	Field	Type	Initial	Description
7	PWCH17	R/W	0	PWM1 shared-pin control 0: GPIO 1: PWM output (shared with P1.1)
6	PWCH16	R/W	0	PWM1 shared-pin control 0: GPIO 1: PWM output (shared with P1.0)
5	PWCH15	R/W	0	PWM1 shared-pin control 0: GPIO 1: PWM output (shared with P0.1)
4	PWCH14	R/W	0	PWM1 shared-pin control 0: GPIO 1: PWM output (shared with P0.0)
3	PWCH13	R/W	0	PWM1 shared-pin control 0: GPIO 1: PWM output (shared with P2.3)
2	PWCH12	R/W	0	PWM1 shared-pin control 0: GPIO 1: PWM output (shared with P2.2)
1	PWCH11	R/W	0	PWM1 shared-pin control 0: GPIO 1: PWM output (shared with P2.1)
0	PWCH10	R/W	0	PWM1 shared-pin control 0: GPIO 1: PWM output (shared with P2.0)

PW1M Registers (PW1M: 0xA1)

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

				100: fosc / 8 101: fosc / 4 110: fosc / 2 111: fosc / 1
3..0	Reserved	R	0	

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

PW1YH/PW1YL Registers (PW1YH: 0xA3, PW1YL: 0xA2)

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: 0xA5, PW10DL: 0xA4)

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

PW11DH/PW11DL Registers (PW11DH: 0xA7, PW11DL: 0xA6)

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

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

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

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

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

PW14DH/PW14DL Registers (PW14DH: 0xBC, PW14DL: 0xBB)

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

PW15DH/PW15DL Registers (PW15DH: 0xBE, PW15DL: 0xBD)

Bit	Field	Type	Initial	Description
7..0	PW15DH/L	R/W	0x00	16-bit PWM1 duty control.

PW16DH/PW16DL Registers (PW16DH: 0xCA, PW16DL: 0XC9)

Bit	Field	Type	Initial	Description
7..0	PW16DH/L	R/W	0x00	16-bit PWM1 duty control.

PW17DH/PW17DL Registers (PW17DH: 0xCC, PW17DL: 0xCB)

Bit	Field	Type	Initial	Description
7..0	PW17DH/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)

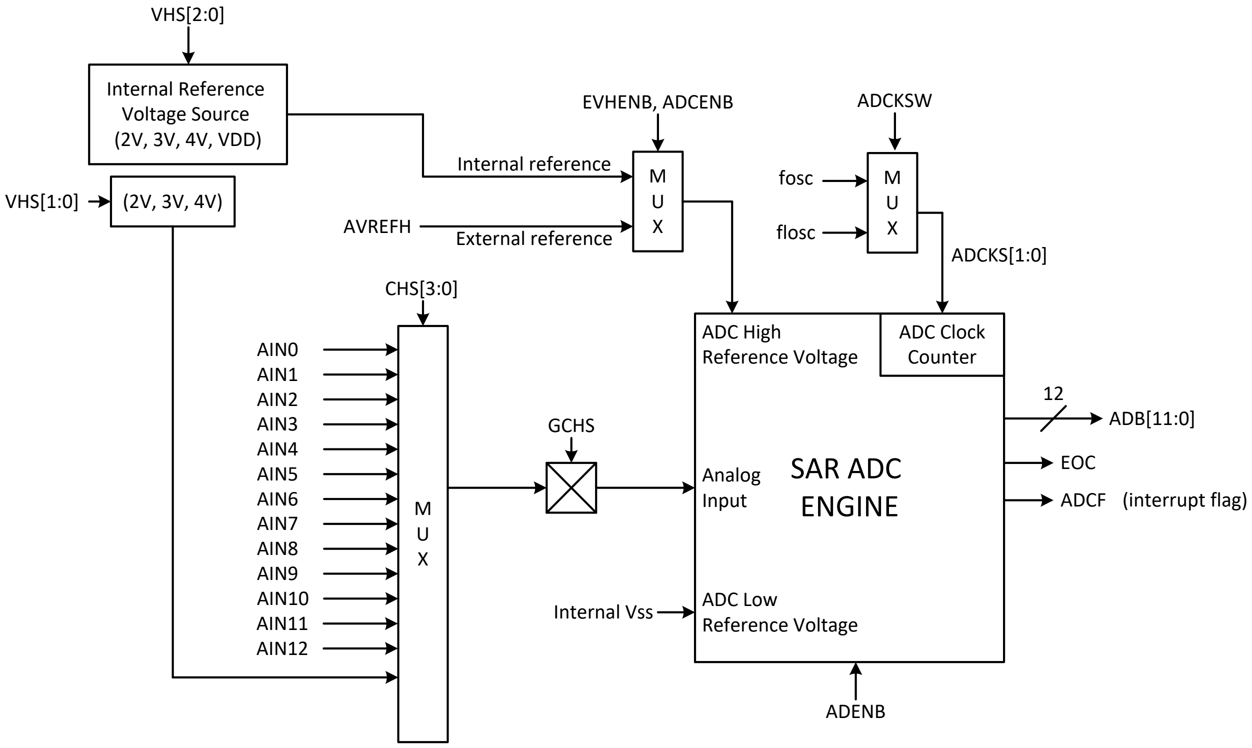
Bit	Field	Type	Initial	Description
3	EPWM1	R/W	0	PWM1 interrupt control bit. 0 = Disable PWM1 interrupt function. 1 = Enable PWM1 interrupt function.
Else				Refer to other chapter(s)

IRCON2 Register (0XBF)

Bit	Field	Type	Initial	Description
3	PWM1F	R/W	0	PWM1 interrupt request flag. 0: None PWM1 interrupt request 1: PWM1 interrupt request.
Else				Refer to other chapter(s)

16 ADC

The analog to digital converter (ADC) is SAR structure with 13-input sources and up to 4096-step resolution to transfer analog signal into 12-bits digital buffers. The ADC builds in 13-channel input source to measure 13 different analog signal sources. The ADC resolution is 12-bit. The ADC has four clock rates to decide ADC converting rate. The ADC high reference voltage includes 5 sources. Four internal power source including VDD, 4V, 3V and 2V. The other one is external reference voltage input pin from AVREFH pin. The ADC builds in P1CON/P2CON registers to set pure analog input pin. After setup ADENB and ADS bits, the ADC starts to convert analog signal to digital data. ADC can work in idle mode. After ADC operating, the system would be waked up from idle mode to normal mode if interrupt enable. When ADC clock source is flosc and STWK=1, ADC can work in stop mode and waked up from stop mode by ADC interrupt.



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

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

16.2 ADC input channel

The ADC builds in 13-channel input source (AIN0 – AIN12) to measure 13 different analog signal sources controlled by CHS[4:0] and GCHS bits. AIN13 channel is reserved. The AIN14 is internal 2V or 3V or 4V 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 2V or 3V or 4V. AIN14 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[4:0]	Channel	Pin name	Remark
00000	AIN0	P1.0	-
00001	AIN1	P1.1	-
00010	AIN2	P1.2	-
00011	AIN3	P1.3	-
00100	AIN4	P1.4	-
00101	AIN5	P1.5	-
00110	AIN6	P1.6	-
00111	AIN7	P1.7	-
01000	AIN8	P2.0	-
01001	AIN9	P2.1	-
01010	AIN10	P2.2	-
01011	AIN11	P2.3	-
01100	AIN12	P2.4	-
01101	AIN13	-	Reserved
01110	AIN14	Internal 2V or 3V or 4V	Battery detector channel
01111~1111	-	-	Reserved

16.2.1 Pin Configuration

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

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

Note that When ADC pin is general I/O mode, the bit of P1CON and P2CON 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, 4V, 3V, 2V). 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 high reference voltage is from internal voltage source selected by VHS[1:0] bits. If VHS[1:0] is "11", ADC high reference voltage is VDD. If VHS[1:0] is "10", ADC high reference voltage is 4V. If VHS[1:0] is "01", ADC high reference voltage is 3V. If VHS[1:0] is "00", ADC high reference voltage is 2V. 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 AIN14 channel is selected as internal 2V or 3V or 4V input channel. There is no any input pin from outside. In this time ADC high reference voltage must be internal VDD or External voltage, not internal 2V/3V/4V.

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/4V/3V/2V 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/4V/3V/2V. (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/16 or flosc/16, 01 = fosc/8 or flosc/8, 10 = fosc/1or flosc/1, 11 = fosc/2 or flosc/2.

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 = 16MHz		fosc = 32MHz	
		Converting time	Converting rate	Converting time	Converting rate
00	fosc/16	1/(16MHz/16/4)*16 = 64us	15.625kHz	1/(32MHz/16/4)*16 = 32us	31.25kHz
01	fosc/8	1/(16MHz/8/4)*16 = 32us	31.25kHz	1/(32MHz/8/4)*16 = 16us	62.5kHz
10	fosc	1/(16MHz/4)*16 = 4us	250kHz	1/(32MHz/4)*16 = 2us	500kHz
11	fosc/2	1/(16MHz/2/4)*16 = 8us	125kHz	1/(32MHz/2/4)*16 = 4us	250kHz

When ADCKSW=1:

ADCKS[1:0]	ADC clock rate	flosc = 16KHz		flosc = 32.768KHz	
		Converting time	Converting rate	Converting time	Converting rate
00	flosc/16	1/(16KHz/16/4)*16 = 64ms	15.625Hz	1/(32.768KHz/16/4)*16 = 31.25ms	32Hz
01	flosc/8	1/(16KHz/8/4)*16 = 32ms	31.25Hz	1/(32.768KHz/8/4)*16 = 15.625ms	64Hz
10	flosc	1/(16KHz/4)*16 = 4ms	250Hz	1/(32.768KHz/4)*16 = 1.953ms	512Hz
11	flosc/2	1/(16KHz/2/4)*16 = 8ms	125Hz	1/(32.768KHz /2/4)*16 = 3.906ms	256Hz

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	CHS4	CHS3	CHS2	CHS1	CHS0
ADB	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4
ADR	ADCKSW	GCHS	ADCKS1	ADCKS0	ADB3	ADB2	ADB1	ADB0
VREFH	EVHENB	-	-	-	-	VHS2	VHS1	VHS0
P1CON	P1CON7	P1CON6	P1CON5	P1CON4	P1CON3	P1CON2	P1CON1	P1CON0
P2CON	-	-	P2CON5	P2CON4	P2CON3	P2CON2	P2CON1	P2CON0
IEN0	EAL	EI2C	EU0RX	EU0TX	ET1	EX1	ET0	EX0
IEN2	-	-	-	-	EPWM1	EX2	-	EADC
IRCON2	-	-	-	-	PWM1F	IE2	-	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..0	CHS[4:0]	R/W	0x00	ADC input channel select bit. 00000: AIN0, 00001: AIN1, 00010: AIN2, 00011: AIN3, 00100: AIN4, 00101: AIN5, 00110: AIN6, 00111: AIN7, 01000: AIN8, 01001: AIN9, 01010: AIN10, 01011:AIN11, 01100: AIN12, 01101: Reserved, 01110: AIN14*(1), Others: Reserved.

*(1) The AIN14 is internal 2V or 3V or 4V 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 2V or 3V or 4V.

ADB Register (0xD3)

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

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

ADR Register (0xD4)

Bit	Field	Type	Initial	Description
7	ADCKSW	R/W	0	ADC clock source select bit 0: fosc 1: flosc
6	GCHS	R/W	0	ADC global channel select bit. 0: Disable AIN channel. 1: Enable AIN channel.
5..4	ADCKS[1:0]	R/W	00	ADC's clock rate select bit. 00 = fosc/16, 01 = fosc/8, 10 = fosc/1, 11 = fosc/2 or 00 = flosc/16, 01 = flosc/8, 10 = flosc/1, 11 = flosc/2
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 high reference 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 high reference voltage selects bits. *(2) 000: VREFH = 2.0V. 001: VREFH = 3.0V. 010: VREFH = 4.0V. 011: VREFH = VDD. 100: VREFH = VDD and AIN14 = 2.0V. 101: VREFH = VDD and AIN14 = 3.0V. 110: VREFH = VDD and AIN14 = 4.0V. Others: Reserved.
Else	Reserved	R/W	0	

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

*(2) If AIN14 channel is selected as internal 2V or 3V or 4V input channel. There is no any input pin from outside. In this time ADC high reference voltage must be internal VDD or External voltage, not internal 2V/3V/4V.

P1CON Register (0x9F)

Bit	Field	Type	Initial	Description
7..0	P1CON[7:0]	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.

* P1CON [7:0] will configure related Port1 pin as pure analog input pin to avoid current leakage.

P2CON Register (0XD6)

Bit	Field	Type	Initial	Description
7..0	P2CON[5:0]	R/W	0x0	P2 configuration control bit*. 0: P2 can be analog input pin (ADC input pin) or digital GPIO pin. 1: P2 is pure analog input pin and can’t be a digital GPIO pin.

* P2CON [5:0] will configure related Port2 pin as pure analog input pin to avoid current leakage.

IEN0 Register (0xA8)

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

IEN2 Register (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 shared pins must set output high and URX set input high by software. 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. EU0TX and EU0RX is UART transfer interrupt function control bit. UART transmitter and receiver interrupt function is controlled by EU0TX and EU0RX respectively. EU0TX=0/EU0RX=0, disable transmitter/receiver interrupt function. EU0TX=1/ EU0RX=1, enable UART transmitter/ receiver interrupt function. When 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.

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 P16M=1 and P16=1 URX pin: Transmitter: P02M=1 and P02=1 P17M=1 and P17=1 Receiver: P02M=0 and P02=1 P17M=0 and P17=1
0	1	1	Asynchronous	Baud rate generator or T1 overflow rate	1	8	1	UTX pin: P03M=1 and P03=1 P16M=1 and P16=1 URX pin: P02M=0 P17M=0
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

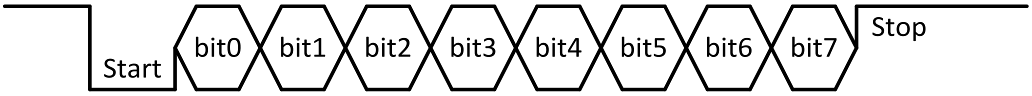
Mode0 is a shift register mode. It operates as synchronous transmitter/receiver. The UTX pin output shift clock for both transmit and receive condition. The URX pin is used to transmit and receive data. 8-bit data will be transmit and receive with LSB first. The baud rate is fcpu/12. Data transmission 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 REN0 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.

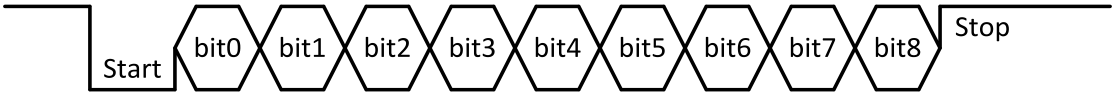


17.4 Mode 2: 9-bit Receiver/Transmitter with Fixed Baud Rate

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 REN0 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.

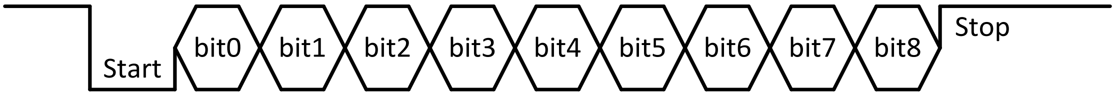


17.5 Mode 3: 9-bit Receiver/Transmitter with Variable Baud Rate

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 REN0 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/64 (SMOD = 0) and fcpu/32 (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.

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

Table 17-1 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	1	0x03	0xF3	0.16 %
38400	1	0x03	0xF9	-6.99 %
56000	1	0x03	0xFB	-10.71 %
57600	1	0x03	0xFC	8.51 %
115200	1	0x03	0xFE	8.51 %
128000	1	0x03	0xFE	-2.34 %
250000	1	0x03	0xFF	0 %

If the Timer 1 overflow period is selected (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.

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

Table 17-2 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 system clock fcpu must

be greater four times to T1 overflow rate.

17.8 Power Saving

The UART module has clock gating function for saving power. When REN0 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 bit is 1, UART internal clocks are run, and registers can access. The REN0 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	-	-	-	-	-	-	URMX
S0BUF	S0BUF7	S0BUF6	S0BUF5	S0BUF4	S0BUF3	S0BUF2	S0BUF1	S0BUF0
PCON	SMOD	-	-	STWK	P2SEL	GF0	STOP	IDLE
S0RELH	-	-	-	-	-	-	S0REL9	S0REL8
S0RELL	S0REL7	S0REL6	S0REL5	S0REL4	S0REL3	S0REL2	S0REL1	R0REL0
IEN0	EAL	EI2C	EU0RX	EU0TX	ET1	EX1	ET0	EX0
P0OC	-	-	-	-	P17OC	P16OC	P03OC	P02OC
P0M	P07M	P06M	P05M	P04M	P03M	P02M	P01M	P00M
P0	P07	P06	P05	P04	P03	P02	P01	P00
P1M	P17M	P16M	P15M	P14M	P13M	P12M	P11M	P10M
P1	P17	P16	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 module (and reception function) 0: Disable for power saving* 1: Enable for UART operating
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 bit is 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 (0xD8)

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
6..1	Reserved	R	0x00	
0	URMX	R/W	0	0 = UART TX and RX pins are P03 and P02 1 = UART TX and RX pins are P16 and P17

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)

S0RELH/S0RELL Registers (S0RELH: 0xBA, S0RELL: 0xAA)

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

IEN0 Register (0xA8)

Bit	Field	Type	Initial	Description
7	EAL	R/W	0	Interrupts enable. Refer to Chapter Interrupt
5	EU0RX	R/W	0	Enable UART RX interrupt
4	EU0TX	R/W	0	Enable UART TX interrupt
Else				Refer to other chapter(s)

P0OC Register (0xE4)

Bit	Field	Type	Initial	Description
3	P17OC	R/W	0	0: Switch P1.7 (URX) to push-pull mode 1: Switch P1.7 (URX) to open-drain mode*
2	P16OC	R/W	0	0: Switch P1.6 (UTX) to push-pull mode 1: Switch P1.6 (UTX) to open-drain mode
1	P03OC	R/W	0	0: Switch P0.3 (UTX) to push-pull mode 1: Switch P0.3 (UTX) to open-drain mode
0	P02OC	R/W	0	0: Switch P0.2(URX) to push-pull mode 1: Switch P0.2 (URX) to open-drain mode*

* Setting P02OC/P17OC as high causes URX cannot receive data.

P0M Register (0xF9)

Bit	Field	Type	Initial	Description
3	P03M	R/W	0	0: Set P0.3 (UTX) as input mode* 1: Set P0.3 (UTX) as output mode (required)
2	P02M	R/W	0	0: Set P0.2 (URX) as input mode (required) 1: Set P0.2 (URX) 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.

P0 Register (0x80)

Bit	Field	Type	Initial	Description
3	P03	R/W	1	0: Set P0.3 (UTX) always low* 1: Make P0.3 (UTX) can output UART data (required)
2	P02	R/W	1	This bit is available to read at any time for monitoring the bus statue.
Else				Refer to other chapter(s)

* Setting P03 initially high because UART block drive the shared pin low signal only.

P1M Register (0xFA)

Bit	Field	Type	Initial	Description
7	P17M	R/W	0	0: Set P1.7 (URX) as input mode (required) 1: Set P1.7 (URX) as output mode*
6	P16M	R/W	0	0: Set P1.6 (UTX) as input mode* 1: Set P1.6 (UTX) as output mode (required)
Else				Refer to other chapter(s)

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

P1 Register (0x90)

Bit	Field	Type	Initial	Description
7	P17	R/W	1	This bit is available to read at any time for monitoring the bus statue.
6	P16	R/W	1	0: Set P1.6 (UTX) always low* 1: Make P1.6 (UTX) can output UART data (required)
Else				Refer to other chapter(s)

* Setting P16 initially high because UART block drive the shared pin low signal only.

18 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.

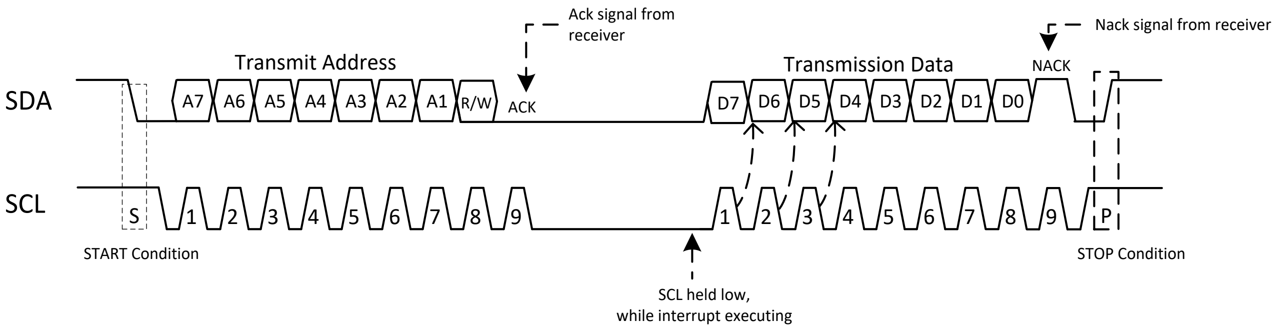
- Master Tx, Rx Mode
- Slave Tx, Rx mode (with general address call) for multiplex slave in single master situation.
- 2-wire synchronous data transfer/receiver.
- Support 100K/400K clock rate.

18.1 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.

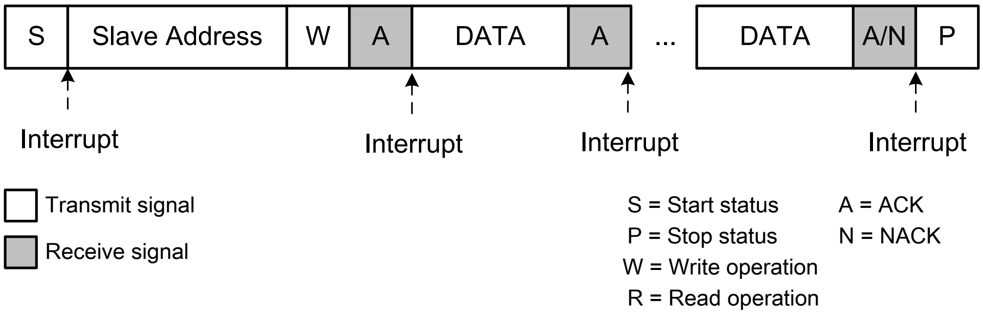


18.2 I2C Transfer Modes

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.

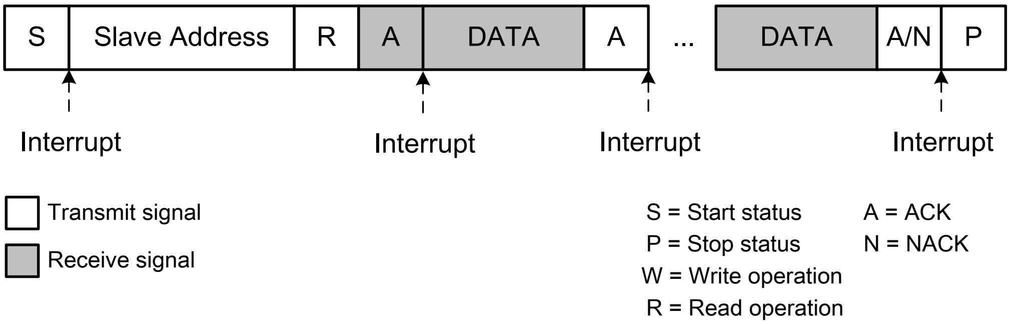
18.2.1 Master Transmitter Mode

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.



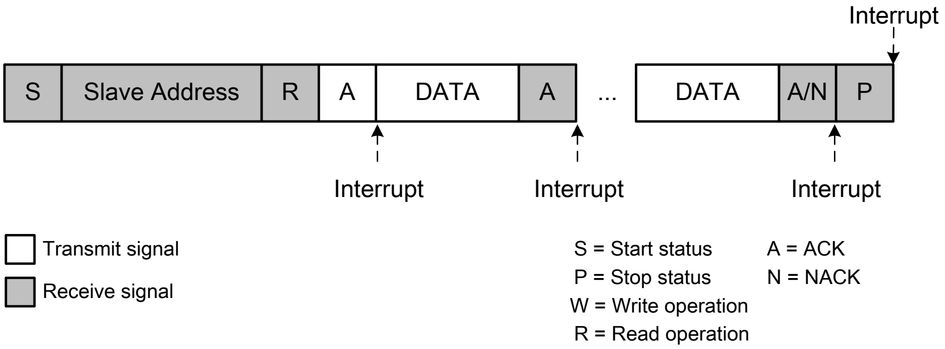
18.2.2 Master Receiver Mode

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.



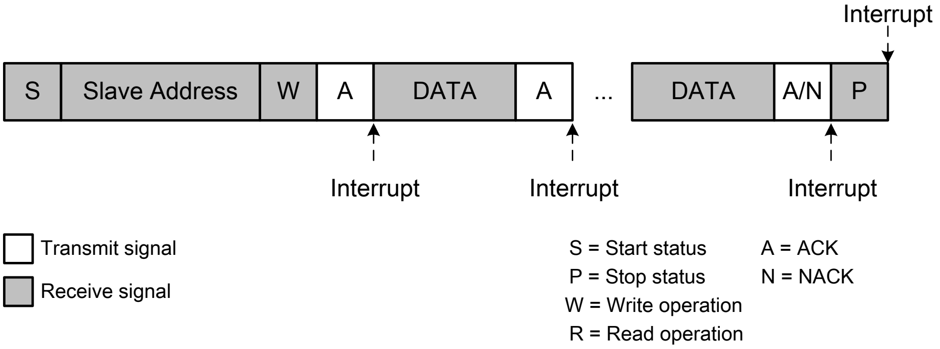
18.2.3 Slave Transmitter Mode

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.



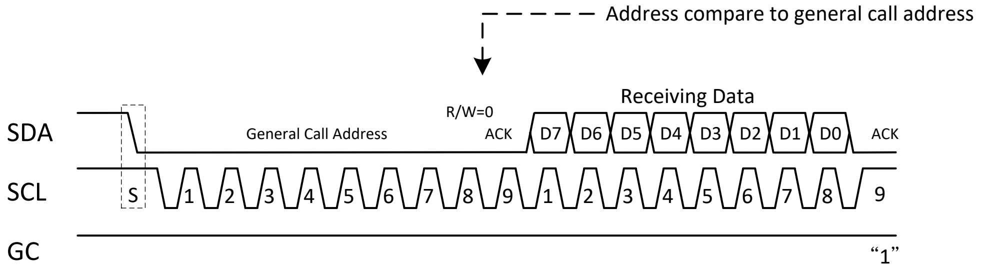
18.2.4 Slave Receiver Mode

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.



18.3 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.



18.4 Serial Clock Generator

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

When CR[2:0]=000~110, SCL clock rate is from internal clock generator.

$$\text{SCL Clock Rate} = \frac{\text{Fcpu}}{\text{Prescaler}} (\text{Prescaler} = 256\sim60)$$

When CR[2:0]=111, 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	Bit Frequency (kHz)
			Prescaler	8MHz
0	0	0	256	31
0	0	1	224	36
0	1	0	192	42
0	1	1	160	50
1	0	0	960	8
1	0	1	120	67
1	1	0	60	133
1	1	1	(Timer 1 overflow rate)/8	

✸ ***Note:***

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

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

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

18.5 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.

18.6 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.

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

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	Fcpu=8MHz	
		DEC	HEX
Tmext	5ms	39	27
Tsext	25ms	195	C3
Tout	35ms	273	111

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.

18.7 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.

18.8 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	EI2C	EU0RX	EU0TX	ET1	EX1	ET0	EX0
P0M	P07M	P06M	P05M	P04M	P03M	P02M	P01M	P00M
P1M	P17M	P16M	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: fcpu/256 001: fcpu/224 010: fcpu/192 011: fcpu/160 100: fcpu/960 101: fcpu/120 110: fcpu/60 111: Timer 1 overflow-period/8
6	ENS1	R/W	0	I2C functionality 0: Disable for power saving* 1: Enable for I2C operating
5	STA	R/W	0	START flag 0: No START condition is transmitted. 1: A START condition is transmitted if the bus is free.
4	STO	R/W	0	STOP flag 0: No STOP condition is transmitted. 1: A STOP condition is transmitted to the I2C bus in master mode.
3	SI	R/W	0	Serial interrupt flag The SI is set by hardware when one of 25 out of 26 possible I2C states is entered. The only state that does not set the SI is state F8h, which indicates that no relevant state information is available. The SI flag must be cleared by software. In order to clear the SI bit, ‘0’ must be written to this bit. Writing a ‘1’ to SI bit does not change value of the SI.
2	AA	R/W	0	Assert acknowledge flag 0: A NACK will be returned when a byte has received 1: An ACK will be returned when a byte has received

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

I2CSTA Register (0xDD)

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

I2C status code and status

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

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

			Read data byte	1	0	0	1	Switched to not addressed SLV mode; own SLA or general call address will be recognized; START condition will be transmitted when the bus becomes free
	90H	Previously addressed with general call address; DATA has been received; ACK returned	Read data byte	X	0	0	0/1	Data byte will be received and not ACK/ACK will be returned
	98H	Previously addressed with general call address; DATA has been received; not ACK returned	Read data byte	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address
			Read data byte	0	0	0	1	Switched to not addressed SLV mode; own SLA or general call address will be recognized
			Read data byte	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address; START condition will be transmitted when the bus becomes free
			Read data byte	1	0	0	1	Switched to not addressed SLV mode; own SLA or general call address will be recognized; START condition will be transmitted when the bus becomes free
	A0H	A STOP condition or repeated START condition has been received while still addressed as SLV/REC or SLV/TRX	No action	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address
			No action	0	0	0	1	Switched to not addressed SLV mode; own SLA or general call address will be recognized
			No action	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address; START condition will be transmitted when the bus becomes free
			No action	1	0	0	1	Switched to not addressed SLV mode; own SLA or general call address will be recognized; START condition will be transmitted when the bus becomes free
Slave Transmitter	A8H	Own SLA+R has been received; ACK has been returned	Load data byte	X	0	0	0	Last data byte will be transmitted and ACK will be received
			Load data byte	X	0	0	1	Data byte will be transmitted; ACK will be received.
	B0H	Arbitration lost in SLA+R/W as master; own SLA+R has been received, ACK has been returned.	Load data byte	X	0	0	0	Last data byte will be transmitted and ACK will be received
			Load data byte	X	0	0	1	Data byte will be transmitted; ACK will be received.
	B8H	Data byte has been transmitted; ACK will be received.	Load data byte	X	0	0	0	Last data byte will be transmitted and ACK will be received
			Load data byte	X	0	0	1	Data byte will be transmitted; ACK will be received.
	C0H	Data byte has been transmitted; not ACK has been received.	No action	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address.
			No action	0	0	0	1	Switched to not addressed SLV mode; own SLA or general call address will be recognized.
			No action	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address; START condition will be transmitted when the bus becomes free.
			No action	1	0	0	1	Switched to not addressed SLV mode; own SLA or general call address will be recognized; START condition will be transmitted when the bus becomes free.
	C8H	Last data byte has been 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
3	I2CMX	R/W	0	0 = I2C SCL and SDA pins are P00 and P01 1 = I2C SCL and SDA pins are P14 and P15
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)

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.

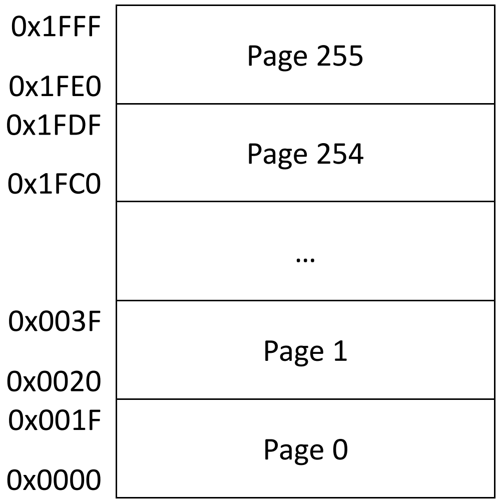
P1M Register (0xFA)

Bit	Field	Type	Initial	Description
5	P15M	R/W	0	0: Set P1.5 (SDA) as input mode (required) 1: Set P1.5 (SDA) as output mode*
4	P14M	R/W	0	0: Set P1.4 (SCL) as input mode (required) 1: Set P1.4 (SCL) as output mode*
Else				Refer to other chapter(s)

* The P14M and P15M require be set input mode.

19 In-System Program

SN8F5713 builds in an on-chip 8 KB program memory, aka IROM, which is equally divided to 256 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)

19.1 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
	1000	One ISP Program Page
	1020	One ISP Program Page
	…	One ISP Program Page
	1700	One ISP Program Page
	1720	One ISP Program Page
	…	One ISP Program Page
	1FE0	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 254th page of program memory (IROM, 0x1FC0 – 0x1FDF) 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 0x1FC0 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 254th 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 255) 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.

19.2 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 0x1FC5 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 0x1FC5 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 0x1FC5 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. ISP operation (byte/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.

19.3 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 (15th– 8th bit) of program page (IROM)

PEROML Register (0x95)

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

PECMD Register (0x94)

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

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

20 Clock Fine-Tuning

SN8F5713 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.

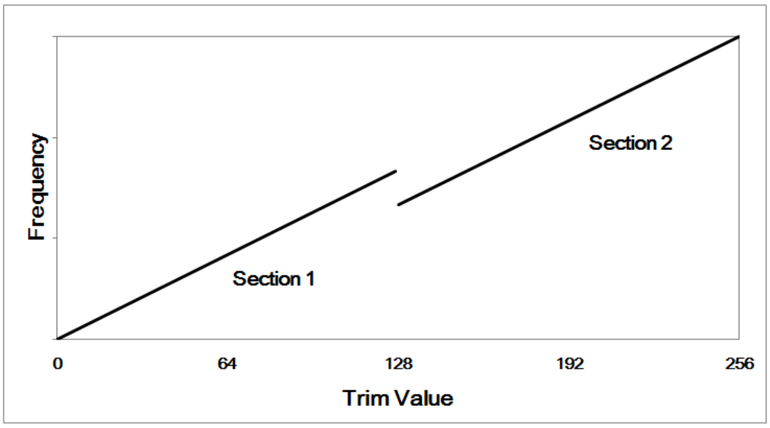
20.1 Clock Trim Section

Clock fine-tuning consists of 8 trim sections as Table 20-1. 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 20-1 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.



20.2 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 Table20-1.
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.***

20.3 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:0xFD, FRQL:0xFC)

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

FRQCMD Register (0xFE)

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

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

21 Electrical Characteristics

21.1 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

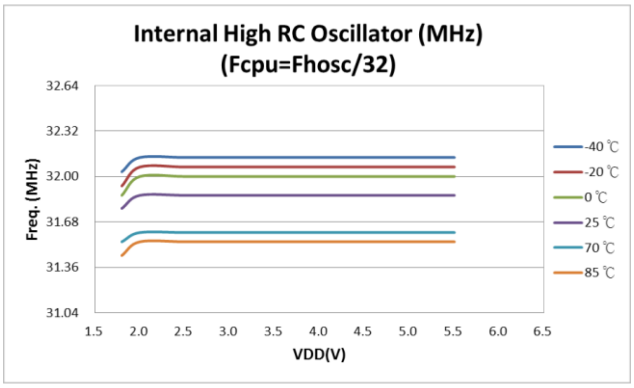
21.2 System Operation Characteristics

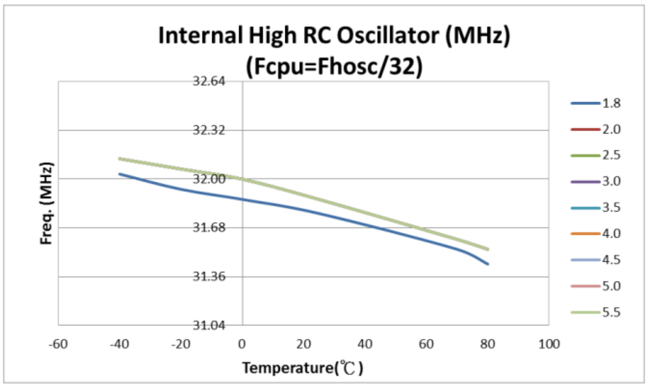
Parameter		Test Condition	Min	TYP	MAX	UNIT
VDD	Operating voltage	fcpu = 1MHz	1.8		5.5	V
V_{DR}	RAM data retention Voltage		0.55			V
V_{POR}	VDD rising rate*		0.05			V/ms
I_{DD1}	Normal mode supply current	VDD = 3V, fcpu = 1MHz		2.75		mA
		VDD = 5V, fcpu = 1MHz		2.80		mA
		VDD = 3V, fcpu = 8MHz		3.65		mA
		VDD = 5V, fcpu = 8MHz		3.70		mA
I_{DD2}	STOP mode supply current	VDD = 3V		4.5	8.5	µA
		VDD = 5V		5.0	9.0	µA
I_{DD3}	STOP mode supply current (ILRC enable STWK=1) *	VDD = 5V		5.5		uA
I_{DD4}	IDLE mode supply current (fcpu = 1MHz)	VDD = 3V, 32MHz IHRC		0.83		mA
		VDD = 5V, 32MHz IHRC		0.85		mA
		VDD = 3V, 16MHz Crystal		0.65		mA
		VDD = 5V, 16MHz Crystal		1.20		mA
		VDD = 3V, 4MHz Crystal		0.28		mA
		VDD = 5V, 4MHz Crystal		0.47		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, -40°C to 85°C	-2.0%	32	+2.0%	MHz
F_{ILRC}	Internal low clock generator	VDD = 5.0V, 25°C	-25%	16	+50%	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		3000			V
V_{ESD_MM}	ESD machine mode		300			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.





21.3 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	16		mA
I_{OL1}	I/O sink current (P04 – P07 , P14 – P17, P20 – P25)	VDD = 5V, V_O = VSS+0.5V	15	20		mA
I_{OL2}	I/O sink current (P00 – P03, P10– P13)	VDD = 5V, V_O = VSS+1.5V	80	100		mA
V_{bias}	1/2*VDD Bias Voltage (P00 – P03, P10– P13)	VDD = 5V	2.4	2.5	2.6	V

* Ambient temperature is 25°C.

21.4 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 4V reference voltage	VDD = 5V	-2%	4	+2%	V
	Internal 3V reference voltage	VDD = 5V	-2%	3	+2%	V
	Internal 2V reference voltage	VDD = 5V	-2%	2	+2%	V
I_{AD}	ADC current consumption	VDD = 3V		0.75		mA
		VDD = 5V		0.80		mA
f_{ADCLK}	ADC clock	VDD = 5V			32	MHz
f_{ADSMP}	ADC sampling rate	VDD = 5V			500	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
		f_{ADSMP} = 500kHz		±3.5		LSB
INL	Integral Nonlinearity*	f_{ADSMP} = 62.5kHz		±2		LSB
		f_{ADSMP} = 250kHz		±2		LSB
		f_{ADSMP} = 500kHz		±3		LSB
NMC	No missing code*	f_{ADSMP} = 62.5kHz	10	11	12	Bit
		f_{ADSMP} = 250kHz		10		Bit
		f_{ADSMP} = 500kHz		9		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.

21.5 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.

22 Instruction Set

This chapter categorizes the SN8F5713 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

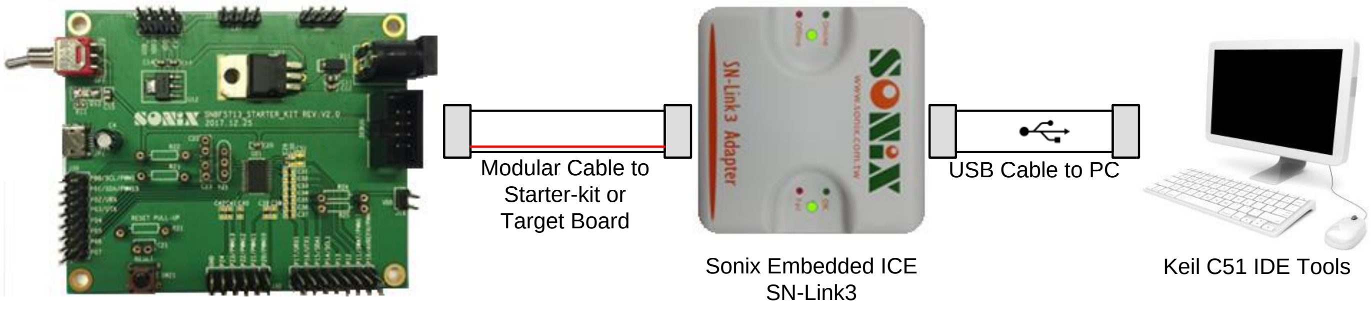
Mnemonic	Description
`CLR C`	Clear carry flag
`CLR bit`	Clear directly addressed bit
`SETB C`	Set carry flag
`SETB bit`	Set directly addressed bit
`CPL C`	Complement carry flag
`CPL bit`	Complement directly addressed bit
`ANL C, bit`	AND directly addressed bit to carry flag
`ANL C, /bit`	AND complement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

23 Development Environment

SONIX provides an Embedded ICE emulator system to offer SN8F5713 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, SN8F5713 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 SN8F5713 to offer a real development environment.



23.1 Minimum Requirement

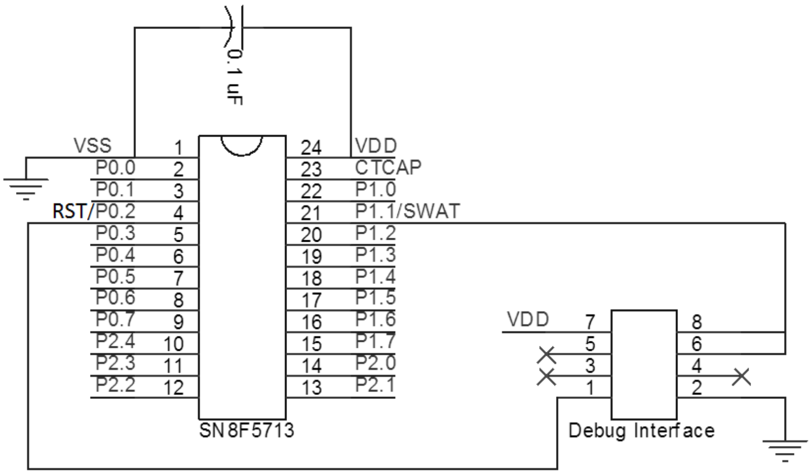
The following items are essential to build up an appropriate development environment. The compatibility is verified on listed versions, and is expected to execute perfectly on later version. SN-Link related information is available to download on SONiX website (www.sonix.com.tw);
Keil C51 is downloadable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

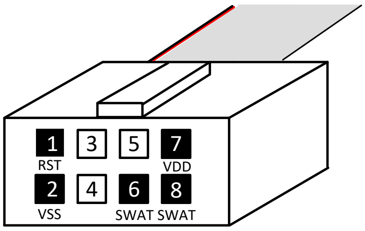
23.2 Debug Interface Hardware

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

23.3 Development Tool

SN-Link3 Adapter

Starter-Kit support SN8F5713, SN8F5712, SN8F5711/131

MP5 Writer

24 SN8F5713 Starter-Kit

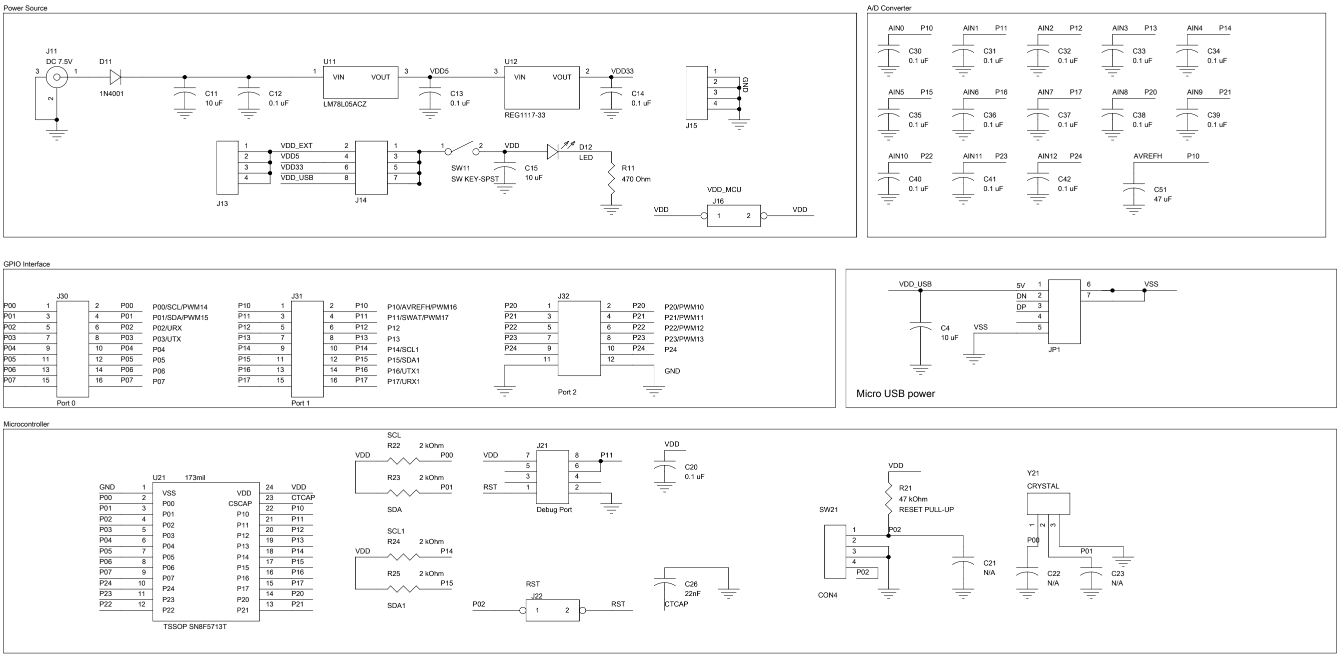
SN8F5000 Starter-Kit provides easy-development platform. It includes SN8F5000 family real chip and I/O connectors to input signal or drive device of user's application. It is a simple platform to develop application as target board not ready. The Starter-Kit can be replaced by target board, because SN8F5000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

24.1 Configurations of Circuit

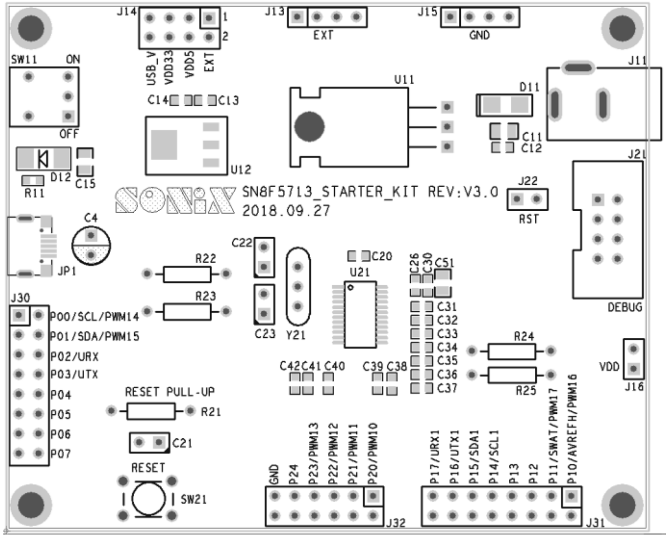
These configurations must be setup completely before starting Starter-Kit developing.

1. Confirm to the circuit board whether elements are complete.
2. The power source of Starter-Kit circuit is chosen from 5.0V, 3.3V, external power or Micro USB via jumper.
3. The power source comes from 5.0V or 3.3V which must be connect to DC 7.5V power adapter.
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 "XIN" pin and the "XOUT" pin need to connect crystal/resonator oscillator components when system clock is setting crystal or RTC mode.
7. The "XIN" pin needs to connect external clock source when system clock is setting external clock input mode.
8. The Debug Port can connect SN-LINK Adapter for emulation or download code.
9. The MCU LED will light up and SN8F5000 family chip will be connected to power when power (VDD) is switched on.

24.2 Schematic



24.3 Floor Plan of PCB layout



24.4 Component Description

Number	Description
C30 – C42	13-ch ADC capacitors.
C51	AVREFH capacitor.
D12	MCU LED
J11	DC 7.5V power adapter
J13/J15	External power source.
SW21	External reset trigger source
J14	VDD power source is 5.0V, 3.3V or external power.
J21	Debug Port
J30 – J32	I/O connector.
R21, C21	External reset pull-high resister and capacitor.
R22, R23, R24, R25	I2C pull-high resisters.
SW11	Target power (VDD) switch
U21	SN8F5713T real chip (Sonix standard option).
Y21, C22, C23,	External crystal/resonator oscillator components.
JP1	Micro USB port

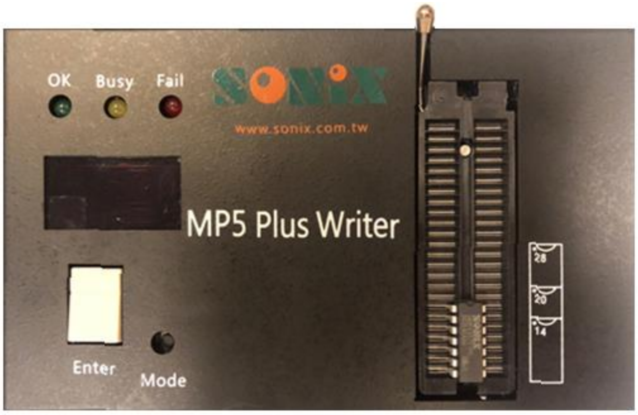
25 ROM Programming Pin

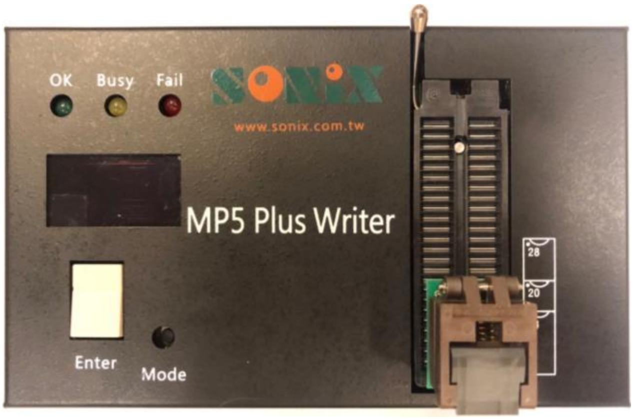
SN8F5713 Series Flash ROM erase/program/verify support SN-Link and MP5 Writer

- SN-Link: Debug interface and on board programming.
- MP5 Writer: For SN8F5713 series version mass programming.

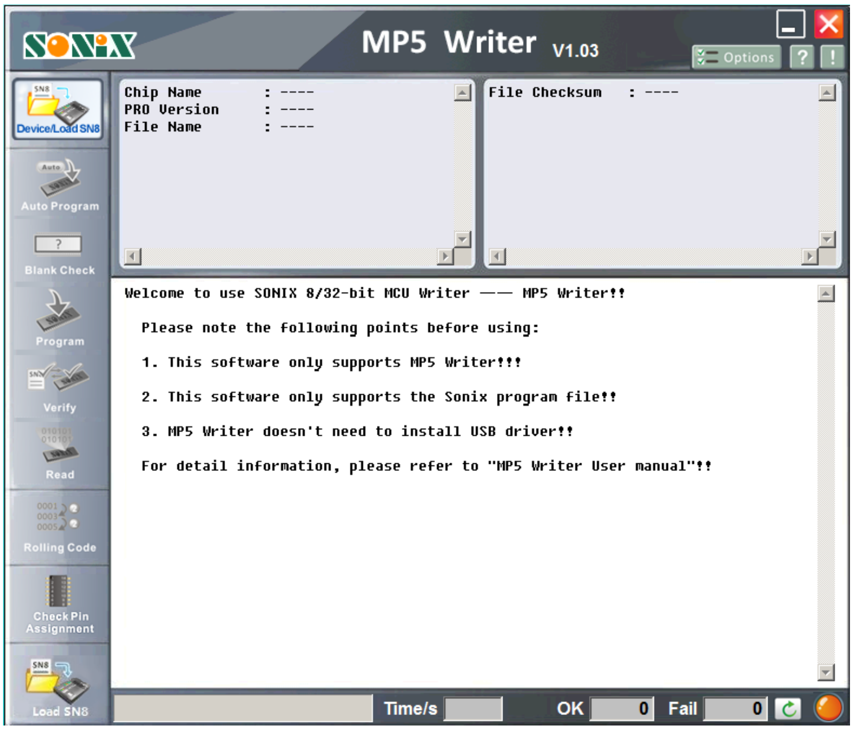
25.1 MP5 Hardware Connecting

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





MP5 Software operation interface is as following.



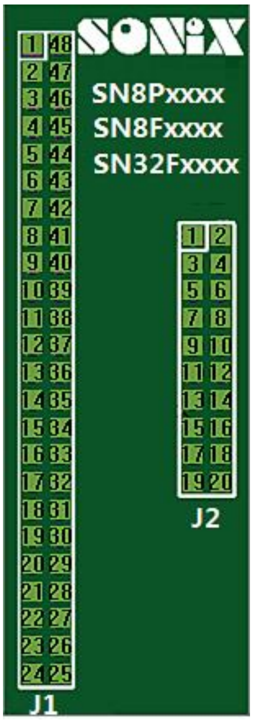
```
Chip Name      : ----
PRO Version    : ----
File Name      : ----
```

```
File Checksum   : ----
```

```
Welcome to use SONIX 8/32-bit MCU Writer —— MP5 Writer!!

  Please note the following points before using:

  1. This software only supports MP5 Writer!!!

  2. This software only supports the Sonix program file!!

  3. MP5 Writer doesn't need to install USB driver!!

  For detail information, please refer to "MP5 Writer User manual"!!
```

25.2 MP5 Writer Transition Board Socket Pin Assignment

MP5 Writer Transition Board:



25.3 MP5 Writer Programming Pin Mapping

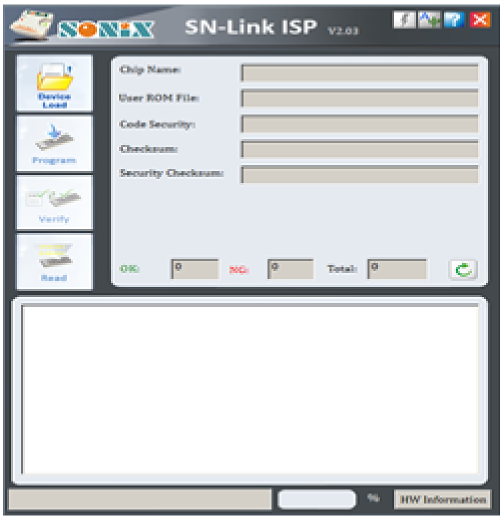
There are two modes of MP5 writer programming: normal mode and high speed mode. Normal mode requires four pins to program the code. The high speed mode requires eight pins to program the code for fast programming.

Writer Connector		MCU Pin Number	SN8F5713S/T		SN8F5712P/S		SN8F5711P/S		SN8F57131P/S	
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	24	36	20	34	16	32	14	31
2	GND	VSS	1	13	1	15	1	17	1	18
7, 9	SWAT	P1.1	21	33	17	31	13	29	12	29
20	PDB	P1.6	16	28	14	28	12	28	11	28

Writer Connector		MCU Pin Number	SN8F5713J		SN8F57112S		SN8F57113S			
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	22	34	8	28	8	28		
2	GND	VSS	23	35	1	21	1	21		
7, 9	SWAT	P1.1	19	31	6	26	6	26		
20	PDB	P1.6	14	26	4	24	4	24		

25.4 SN-Link ISP Programming

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



Chip Name:
User ROM File:
Code Security:
Checksum:
Security Checksum:

OK: 0 NG: 0 Total: 0

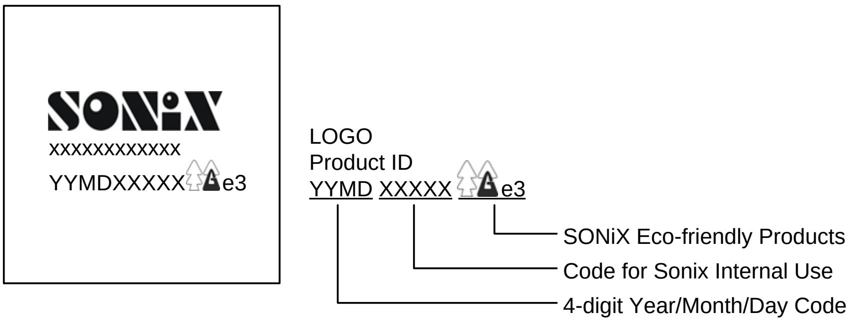
25.5 SN-Link ISP Programming Pin Mapping

SN-Link Connector		MCU Pin Number	SN8F5713S/T	SN8F5712P/S	SN8F5711P/S	SN8F57131P/S
Pin Number	Pin Name		Pin Number	Pin Number	Pin Number	Pin Number
7	VDD	VDD	24	20	16	14
2	GND	VSS	1	1	1	1
6, 8	SWAT	P1.1	21	17	13	12

SN-Link Connector		MCU Pin Number	SN8F5713J	SN8F57112S	SN8F57113S	
Pin Number	Pin Name		Pin Number	Pin Number	Pin Number	Pin Number
7	VDD	VDD	22	8	8	
2	GND	VSS	23	1	1	
6, 8	SWAT	P1.1	19	6	6	

26 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.

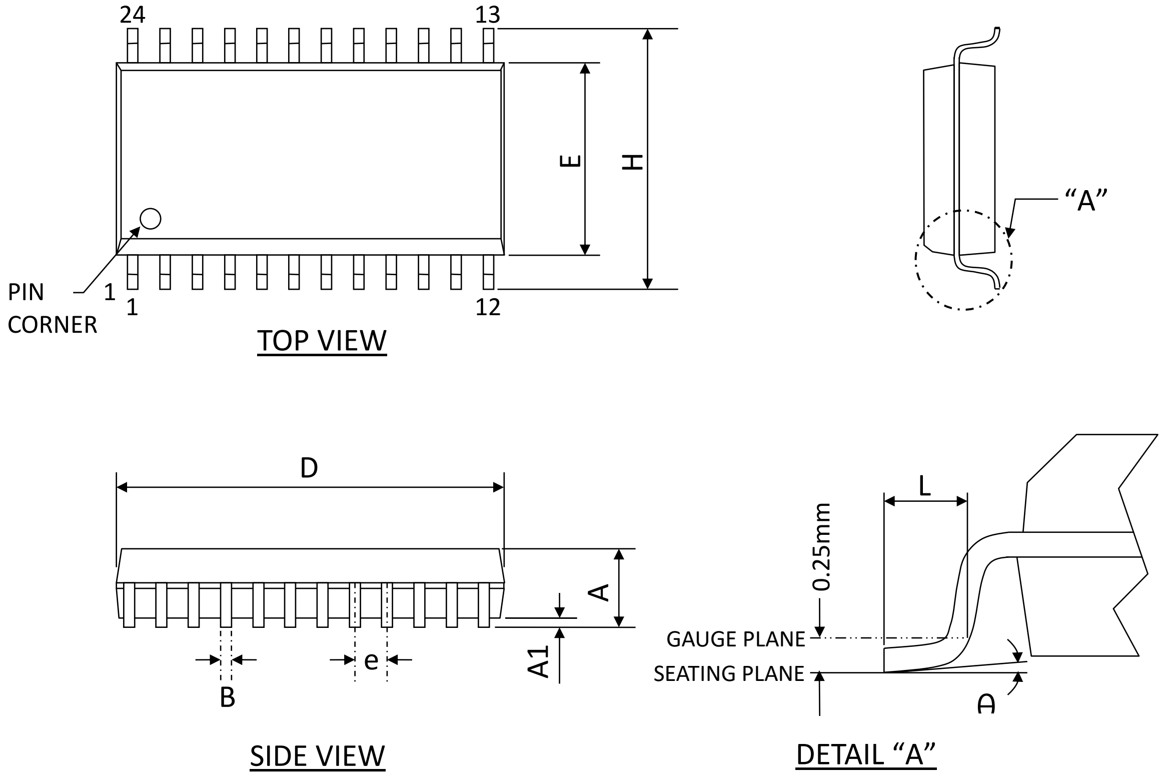

SONiX
XXXXXXXXXXXX
YYMDXXXXX e3
LOGO
Product ID
YYMD XXXXX e3
SONiX Eco-friendly Products
Code for Sonix Internal Use
4-digit Year/Month/Day Code

26.1 Device Nomenclature

Full Name	Packing Type
S8F5713W	Wafer
SN8F5713H	Dice
SN8F5713SG	SOP, 24 pins, Green package
SN8F5713TG	TSSOP, 24 pins, Green package
SN8F5713JG	QFN, 24 pins, Green package
SN8F5712PG	DIP, 20 pins, Green package
SN8F5712SG	SOP, 20 pins, Green package
SN8F5711PG	DIP, 16 pins, Green package
SN8F5711SG	SOP, 16 pins, Green package
SN8F57131PG	DIP, 14 pins, Green package
SN8F57131SG	SOP, 14 pins, Green package
SN8F57112SG	SOP, 8 pins, Green package
SN8F57113SG	SOP, 8 pins, Green package

27 Package Information

27.1 SOP24

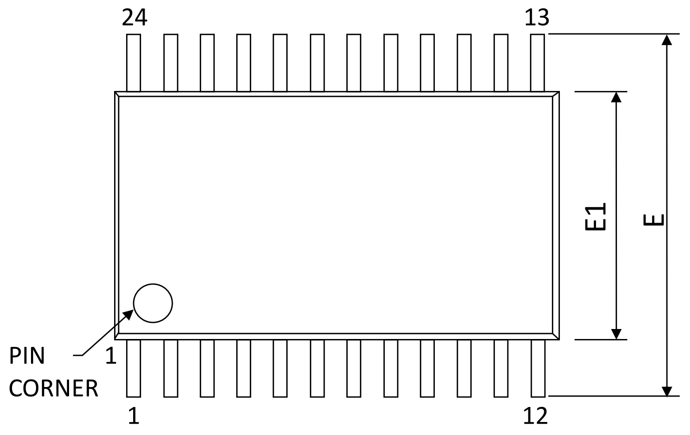


SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	2.65	--	--	0.104
A1	0.10	--	0.30	0.004	--	0.011
B	0.31	0.41	0.51	0.012	0.016	0.020
D	15.30	15.50	15.70	0.602	0.618	0.618
E	7.50 BSC			0.295 BSC		
e	1.27 BSC			0.050 BSC		
H	10.30 BSC			0.405 BSC		
L	0.4	--	1.27	0.015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

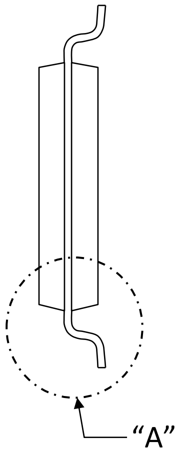
Notes：

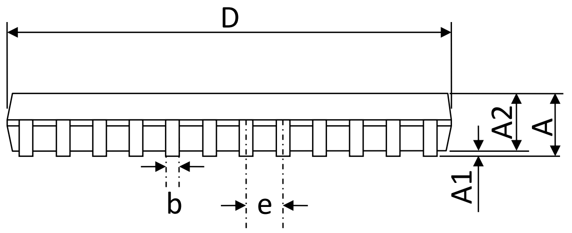
1. CONTROLLING DIMENSION：mm
2. JEDEC OUTLINE : MO-119 AA

27.2 TSSOP24

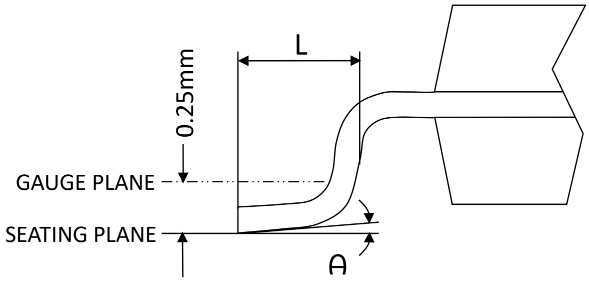


TOP VIEW





SIDE VIEW



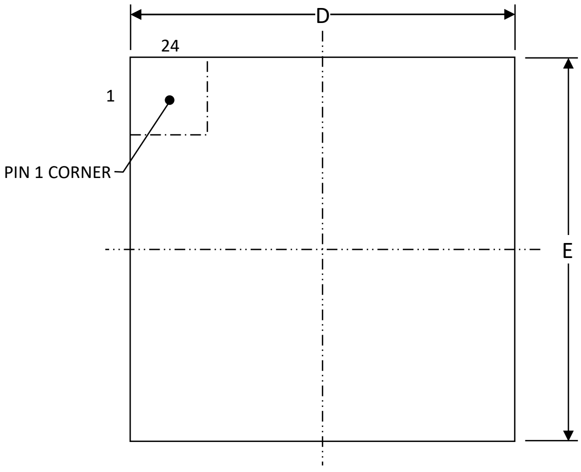
DETAIL "A"

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

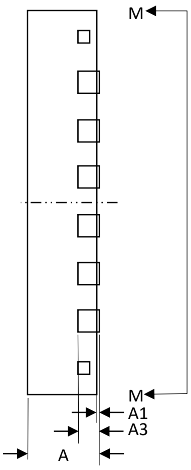
Notes：

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

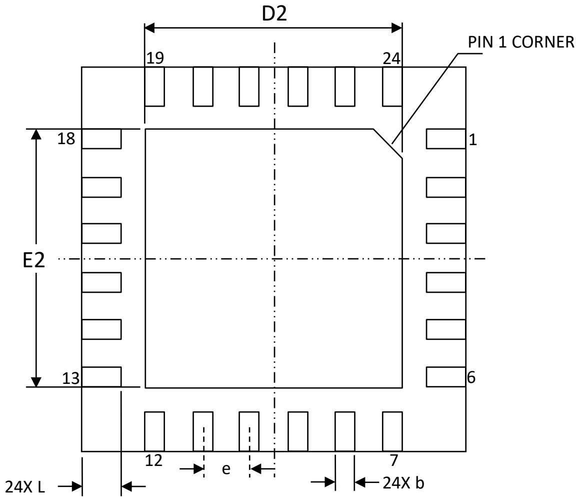
27.3 QFN24 4X4



TOP VIEW



SIDE VIEW



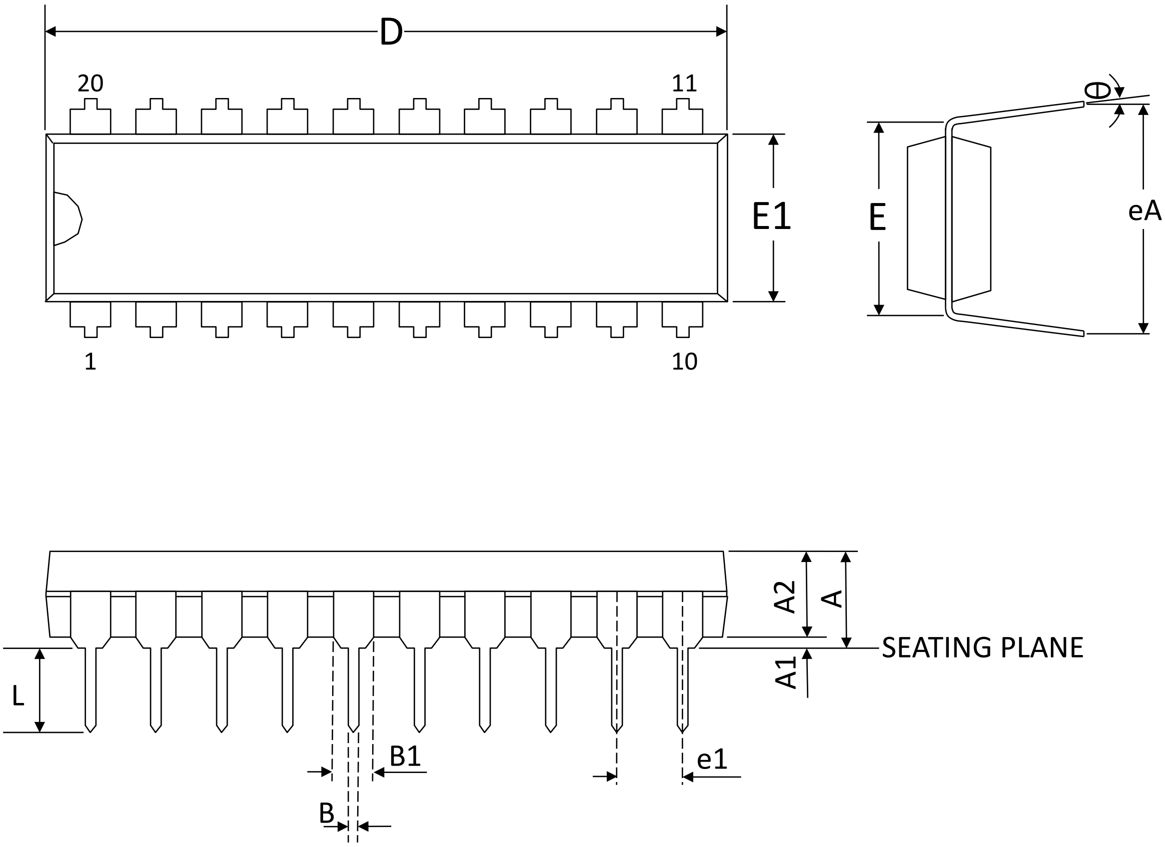
BOTTOM VIEW

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	0.203 REF			0.008 REF		
b	0.15	0.25	0.30	0.007	0.010	0.012
D	4.00 BSC			0.157 BSC		
E	4.00 BSC			0.157 BSC		
e	0.50 BSC			0.020 BSC		
D2	1.90	2.35	2.80	0.075	0.093	0.110
E2	1.90	2.35	2.80	0.075	0.093	0.110
L	0.30	0.40	0.50	0.012	0.016	0.020

Notes：

1. CONTROLLING DIMENSION：MILLIMETER (mm)

27.4 DIP20

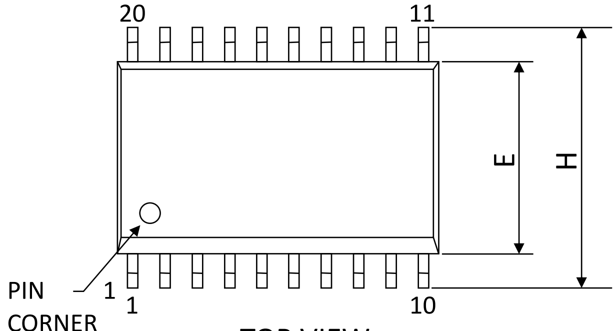


SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	4.45	--	--	0.175
A1	0.35	--	--	0.015	--	--
A2	3.18	3.30	3.43	0.125	0.130	0.135
B	0.46 typ.			0.018 typ.		
B1	1.52 typ.			0.060 typ.		
D	25.70	26.06	26.42	1.012	1.026	1.040
E	7.62 BSC.			0.300 BSC.		
E1	6.05	6.35	6.65	0.238	0.250	0.261
e1	2.54 typ.			0.100 typ.		
L	3.05	3.30	3.56	0.120	0.130	0.140
eA	7.62	9.02	9.53	0.300	0.355	0.375
θ	0°	7°	15°	0°	7°	15°

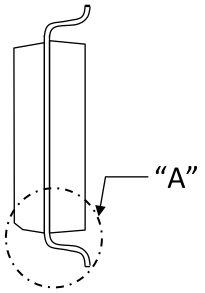
Notes：

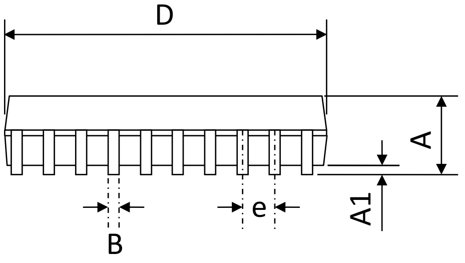
1. JEDEC OUTLINE : MS-001 AD
2. CONTROLLING DIMENSION：inch

27.5 SOP20

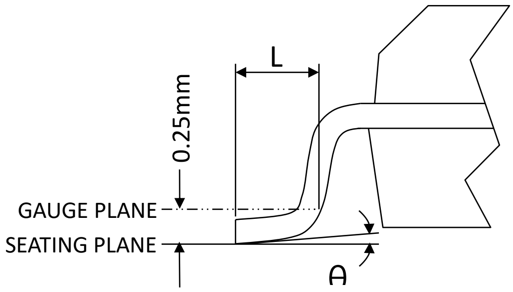


TOP VIEW





SIDE VIEW



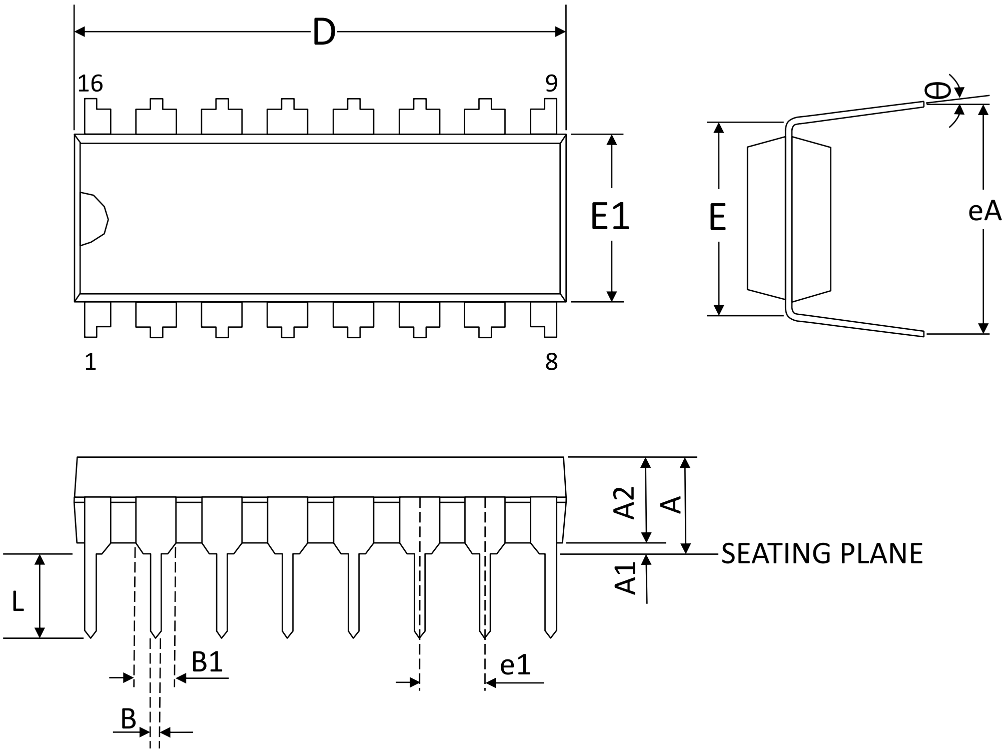
DETAIL "A"

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

Notes：

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

27.6 DIP16

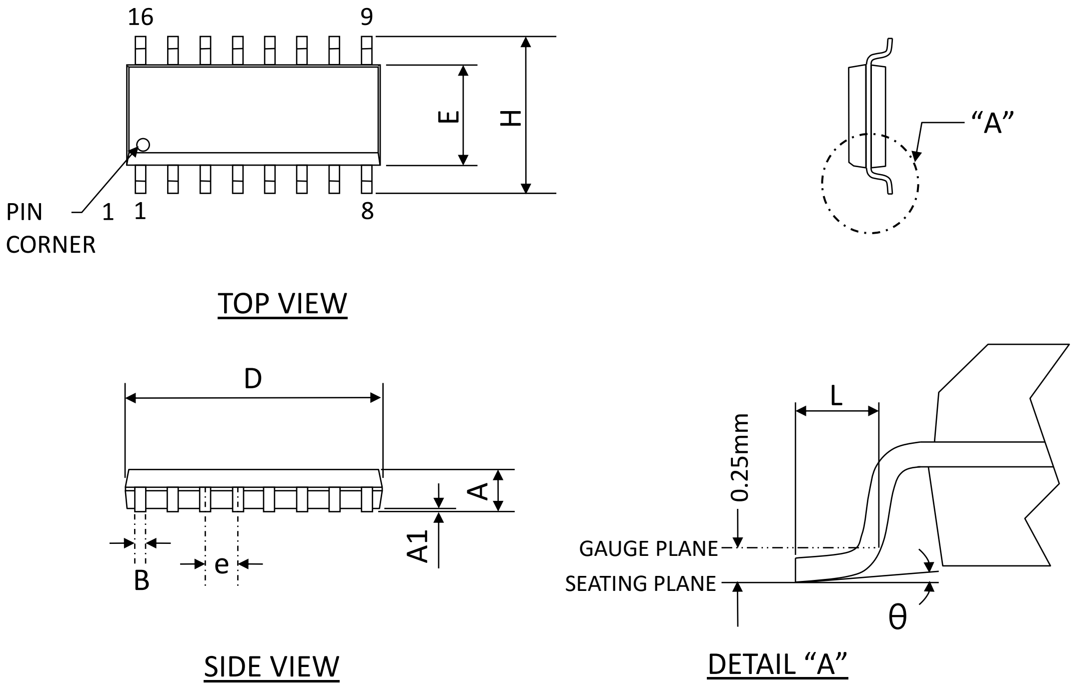


SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	5.33	--	--	0.210
A1	0.38	--	--	0.015	--	--
A2	3.18	3.30	3.43	0.125	0.130	0.135
B	0.46 typ.			0.018 typ.		
B1	1.52 typ.			0.060 typ.		
D	18.67	19.18	19.69	0.735	0.755	0.775
E	7.62 BSC.			0.300 BSC		
E1	6.22	6.35	6.48	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 BB
2. CONTROLLING DIMENSION：inch

27.7 SOP16



TOP VIEW

SIDE VIEW

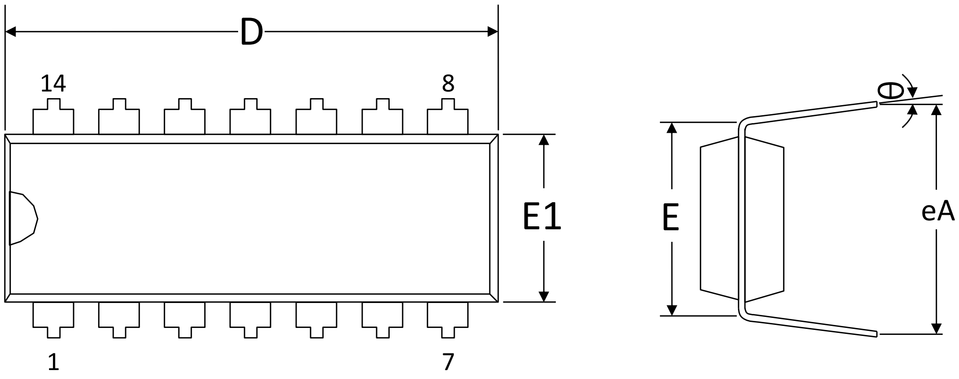
DETAIL "A"

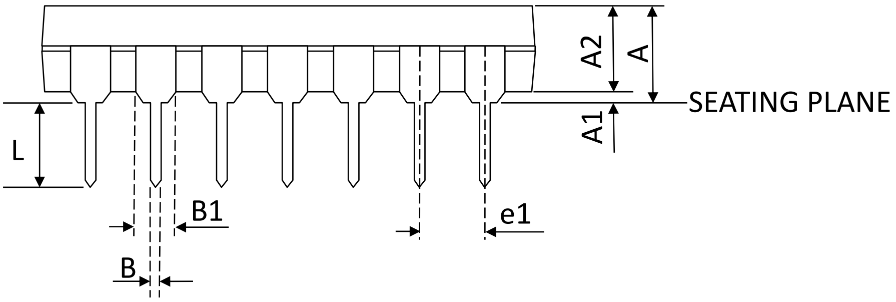
SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	1.75	--	--	0.069
A1	0.10	--	0.25	0.004	--	0.010
B	0.31	0.41	0.51	0.012	0.016	0.020
D	9.90 BSC			0.389 BSC		
E	3.90 BSC			0.153 BSC		
e	1.27 BSC			0.050 BSC		
H	6.00 BSC			0.236 BSC		
L	0.40	--	1.27	0.016	--	0.050
θ	0°	4°	8°	0°	4°	8°

Notes：

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

27.8 DIP14



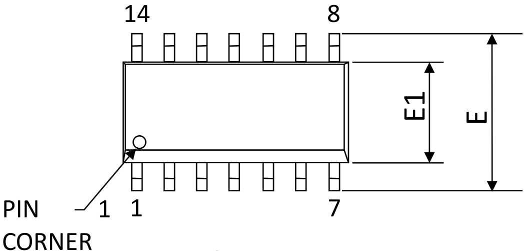


SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	5.33	--	--	0.210
A1	0.38	--	--	0.015	--	--
A2	3.18	3.30	3.43	0.125	0.130	0.135
B	0.46 typ.			0.018 typ.		
B1	1.52 typ.			0.060 typ.		
D	18.67	19.05	19.69	0.735	0.750	0.775
E	7.62 BSC			0.300BSC		
E1	6.22	6.35	6.48	0.245	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°

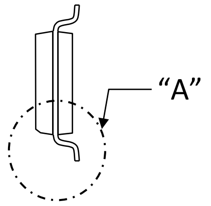
Notes：

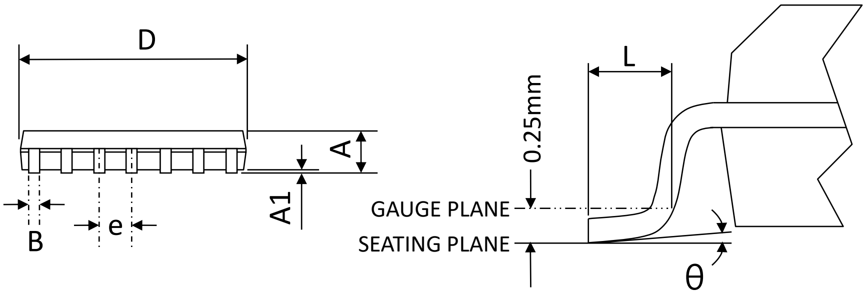
1. JEDEC OUTLINE : MS-001 AA
2. CONTROLLING DIMENSION：inch

27.9 SOP14



TOP VIEW





SIDE VIEW

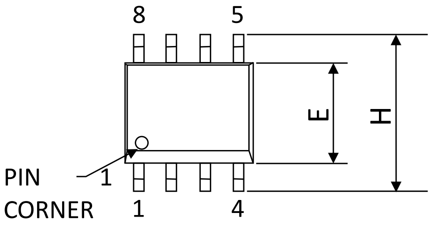
DETAIL “A”

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°

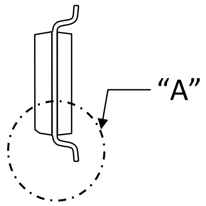
Notes：

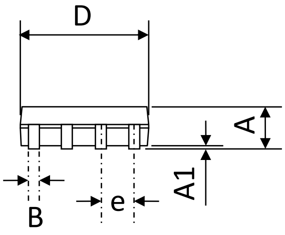
1. CONTROLLING DIMENSION：mm
2. JEDEC OUTLINE : MS-012 AB

27.10 SOP8

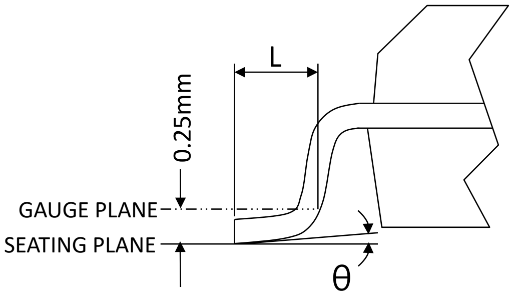


TOP VIEW





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

28 Appendix: Reference Document

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

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

SN8F5713 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