SONiX Technology Co., Ltd.

SN8F5829 Series Datasheet

8051-based Microcontroller

SN8F5829
SN8F5828
SN8F5825
SN8F58252
SN8F58253
SN8F58254
SN8F5824
SN8F58242
SN8F58243
SN8F58244
SN8F5823

1 Device Overview

1.1 Features

- **Enhanced 8051 microcontroller** with reduced instruction cycle time (up to 12 times 80C51)
- Up to 16 MHz flexible CPU frequency
- Internal 32 MHz Clock Generator (IHRC), 1 MHz to 16 MHz crystal, and external synchronous clock source selections
- 1-set external low clock 32.768 kHz crystal
- **32 KB non-volatile flash memory (IROM)** with in-system program support
- **256 bytes internal RAM (IRAM)**
- **1536 bytes external RAM (XRAM)**
- **23 interrupt sources with priority levels control and unique interrupt vectors**
- 18 internal interrupts
- 4 external interrupts: INT0/1/2/3/4
- 2 set of DPTR
- 2 set 8/16-bit timers with 4 operation modes
- 1 set 16-bit capture timer
- 2 set 16-bit timers with adjustable cycle
- **4 set 16-bit PWM generators**:
  PWM generator has 4 output channels with individual duty cycle control
- **Hardware Multiplication/Division Unit**
- **12-bit SAR ADC** with 16 external and 3 internal channels, and 4 internal reference voltages
- **24 channel Capacitive touch**
- **4*36/6*34/7*33/8*32 R-type LCD Driver with static mode**
- **3 set UARTs with individual interrupt vector**
- **1 set I2C** interface with SMBus Support
- **On-Chip Debug Support**:
  Single-wire debug interface
  5 hardware breakpoints
  Unlimited software breakpoints
  ROM data security/protection
- Watchdog and programmable external reset
- 1.8 low voltage detectors
- Wide supply voltage (1.8 V – 5.5 V) and temperature (-40 °C to 85 °C) range

1.2 Applications

- Home automation
- Touch Key

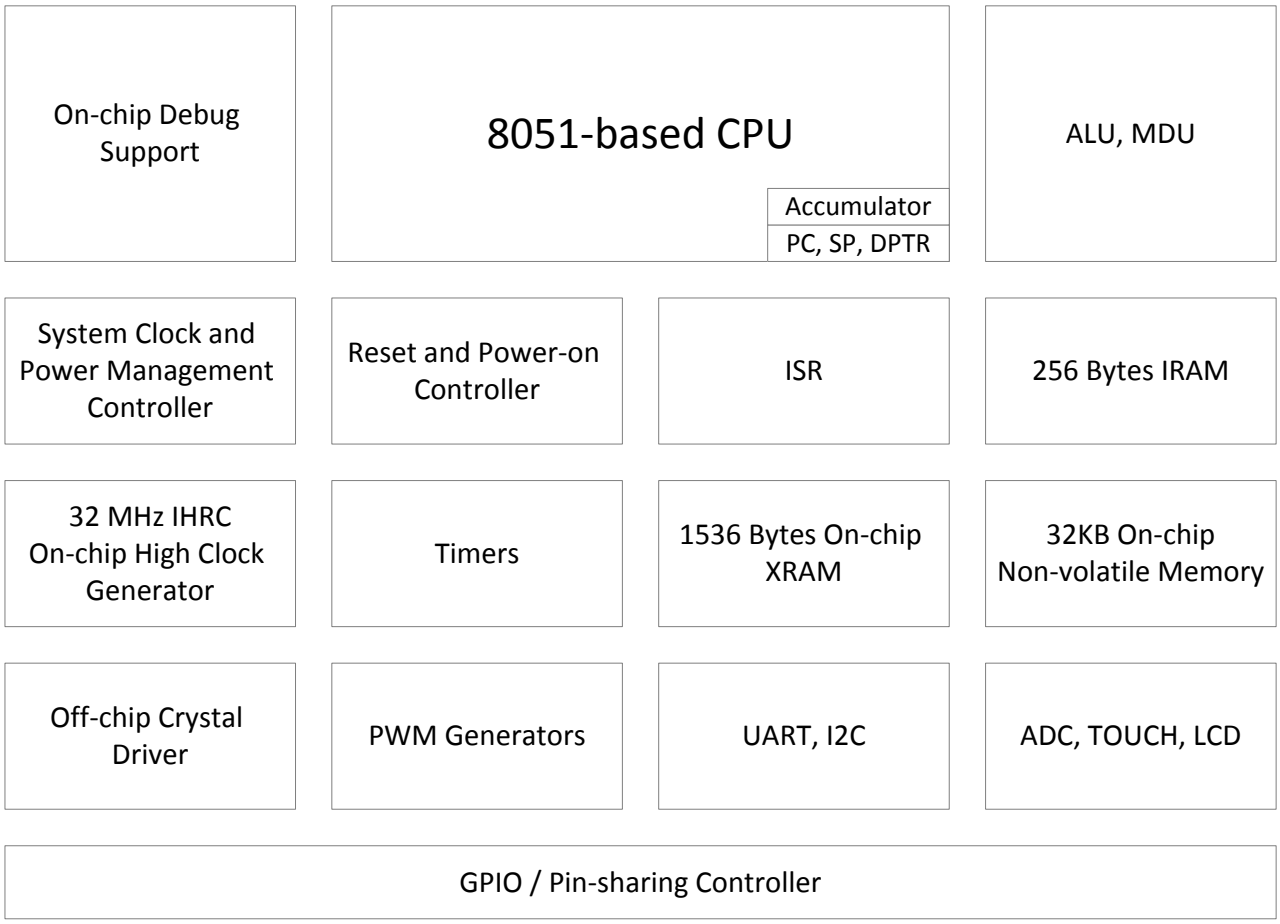
Copyright © 2019, SONiX Technology Co., Ltd.

1.3 Features Selection Table

	I/O	PWM Channels	I2C	SPI	UART	ADC ext. Channels	LCD	Touch	Ext. INT	Package Types
SN8F5829	60	16	V	-	3	16	4*36 6*34 7*33 8*32	24	4	LQFP64
SN8F5828	45	14	V	-	2	13	4*25	17	4	LQFP48
SN8F5825	29	7	V	-	1	8	4*14	13	4	LQFP32 QFN32
SN8F58252	30	8	V	-	2	9	4*15	-	4	LQFP32 QFN32
SN8F58253	30	8	V	-	2	12	-	-	4	LQFP32 QFN32
SN8F58254	30	10	V	-	2	11	-	-	4	LQFP32 QFN32
SN8F5824	25	8	V	-	1	9	-	14	4	SOP28
SN8F58242	26	6	V	-	2	7	4*11	-	4	SOP28
SN8F58243	26	6	V	-	2	11	-	-	4	SOP28
SN8F58244	26	10	V	-	2	7	-	-	4	SOP28
SN8F5823	21	5	V	-	-	8	-	11	4	SOP24

Copyright © 2019, SONiX Technology Co., Ltd.

1.4 Block Diagram



2 Table of Contents

Copyright © 2019, SONiX Technology Co., Ltd.

3 Revision History

Revision	Date	Description
1.0	Mar. 2018	First issue.
1.1	Apr. 2018	1. Add SSOP48, LQFP32, QFN32, SOP28, SOP24 pin assignment. 2. Add STOP mode current in ILRC enable. 3. Add electrical characteristic min. specification.
1.2	Jun. 2018	1. Repair an error, omission, etc. 2. Add SN8F5829 Starter-Kit chapter.
1.3	Jul. 2018	1. Repair an error, omission, etc.
1.4	Sep. 2018	1. Repair an error, omission, etc. 2. MP5 Writer Programming Pin Mapping adds normal mode and high speed mode sections. 3. Modify SOP24 outline description.
1.5	Dec. 2018	1. Modify normal mode supply current value. 2. Modify Pin Circuit Diagrams section. 3. Pin Descriptions and Pin Characteristic sections add P2.0 description. 4. P2M, P2 and P2UR registers add P2.0 function bit. 5. Add external interrupt 0 function (INT0/P2.0). Add interrupt vector 0x0003, control bit EX0 and interrupt request flag IE0. Add INT0 trigger edge control bit EX0G1 and EX0G0. 6. The capture input source of Timer 2 adds T2CC3/P2.0. 7. Add SN8F58252F/J, SN8F58253F/J, SN8F58254, SN8F58242S, SN8F58243S, SN8F58244S pin assignment. 8. Remove ADT register and offset calibration description.
1.6	Jan. 2019	1. Remove SN8F58281X and SN8F5827F pin assignment.
1.7	Apr. 2019	1. Repair an error, omission, etc. 2. Update Package Information dimensions and illustrations. 3. Modify ADC input offset range. 4. Modify power on sequence and system clock timing. 5. Modify Timer0/ Timer1 section description. 6. Code option table removes Noise Detect option.

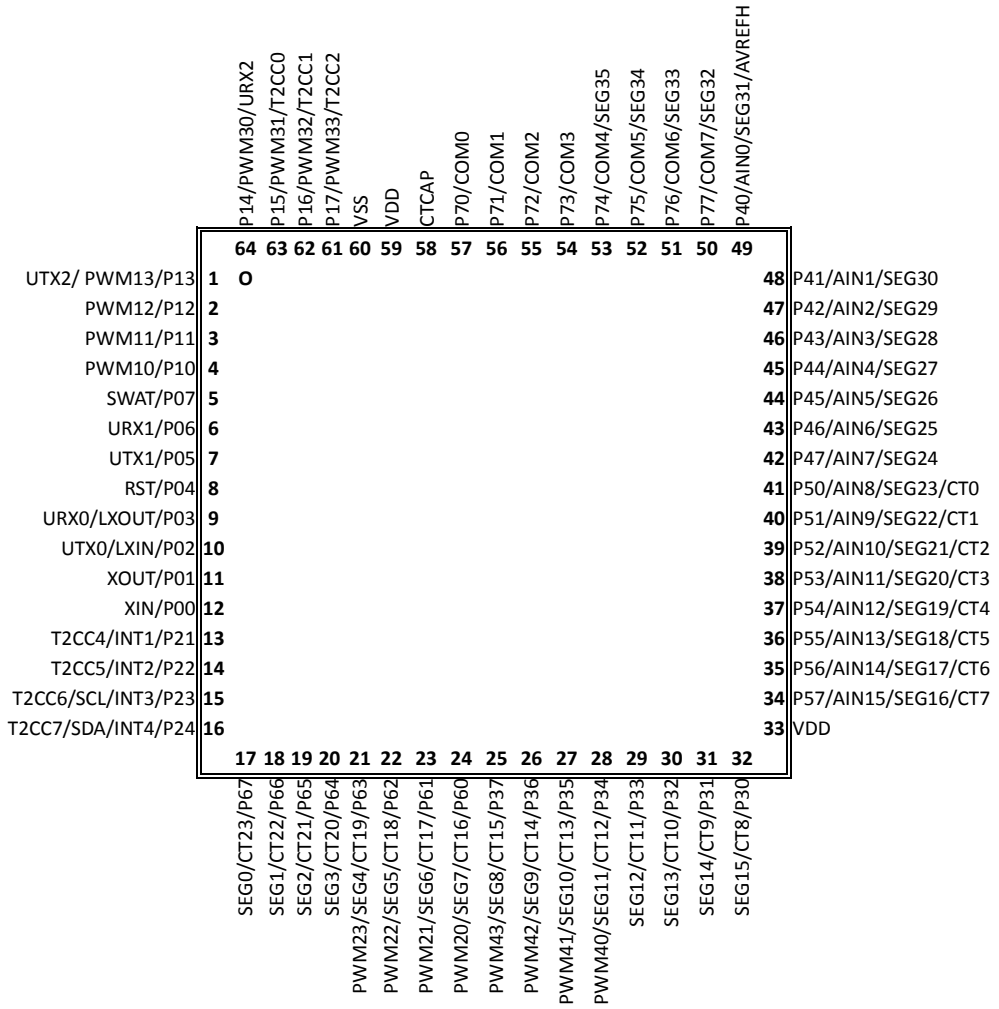
Copyright © 2019, SONiX Technology Co., Ltd.

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.

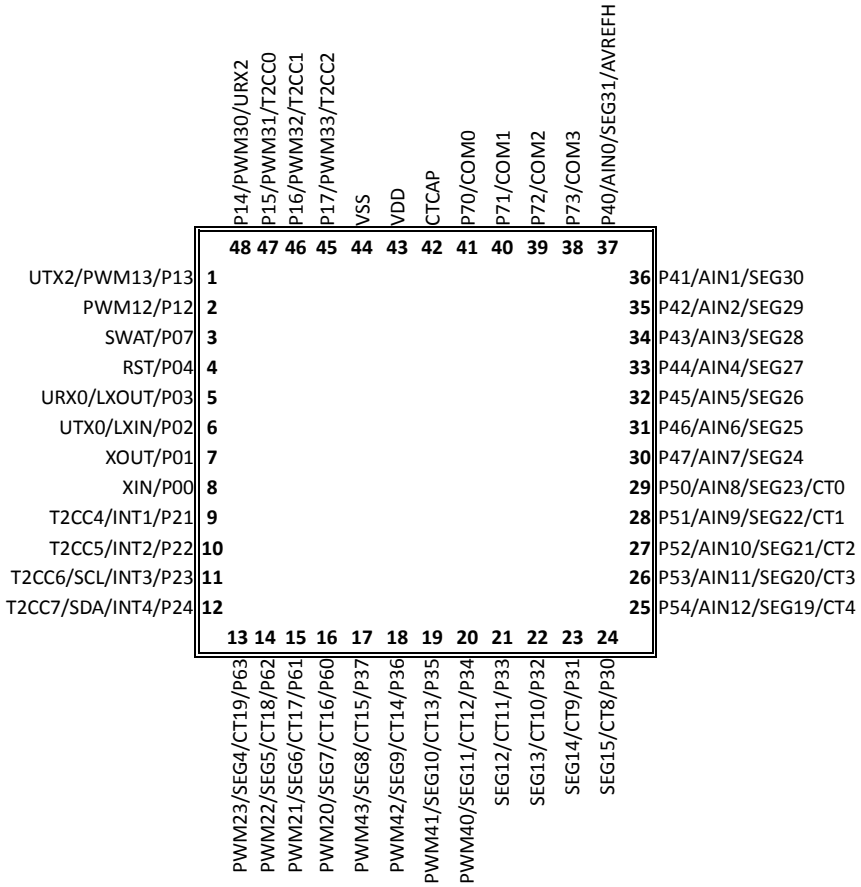
Copyright © 2019, SONiX Technology Co., Ltd.

4 Pin Assignments

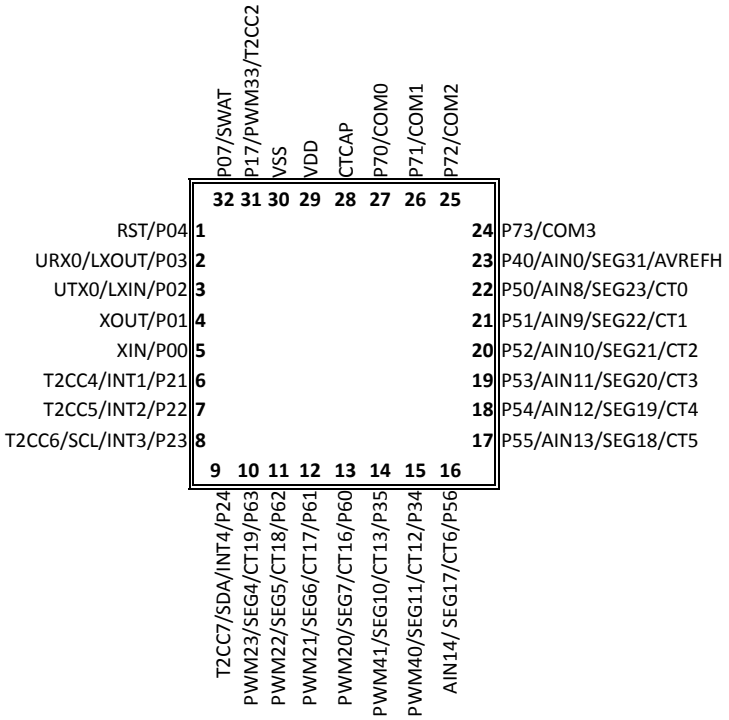
4.1 SN8F5829F (LQFP64)



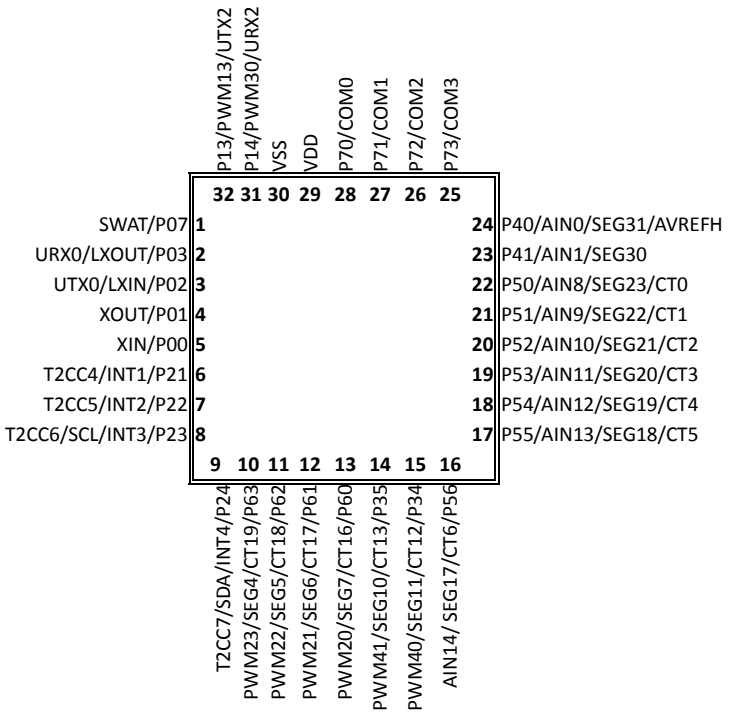
4.2 SN8F5828F (LQFP48)



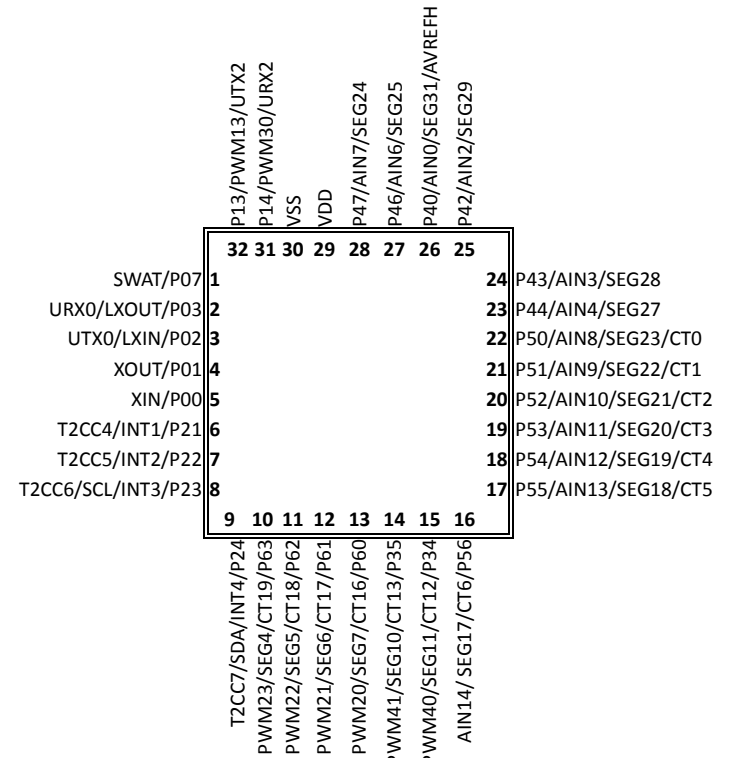
4.3 SN8F5825F/J (LQFP32/QFN32)



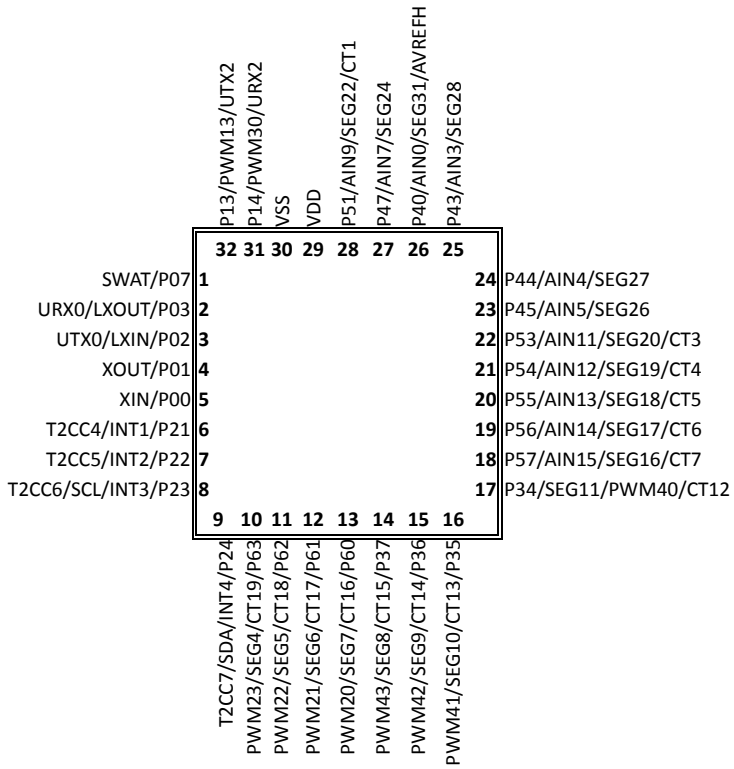
4.4 SN8F58252F/J (LQFP32/QFN32)



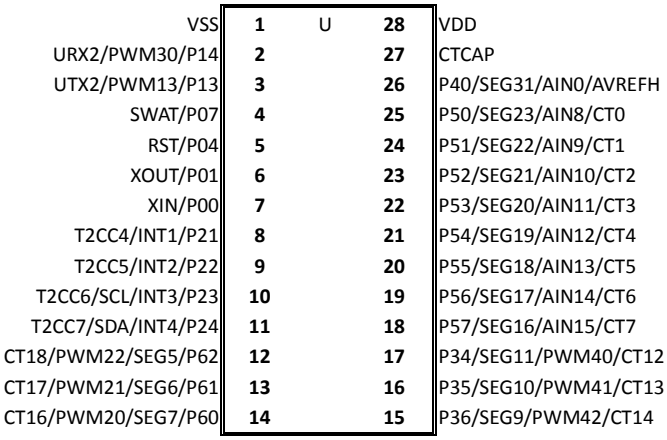
4.5 SN8F58253F/J (LQFP32/QFN32)



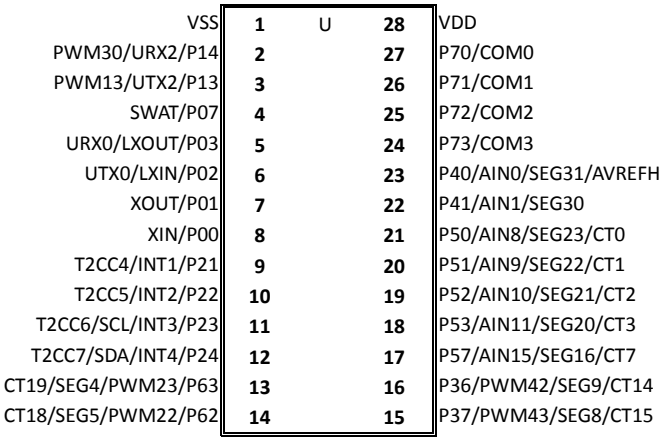
4.6 SN8F58254F/J (LQFP32/QFN32)



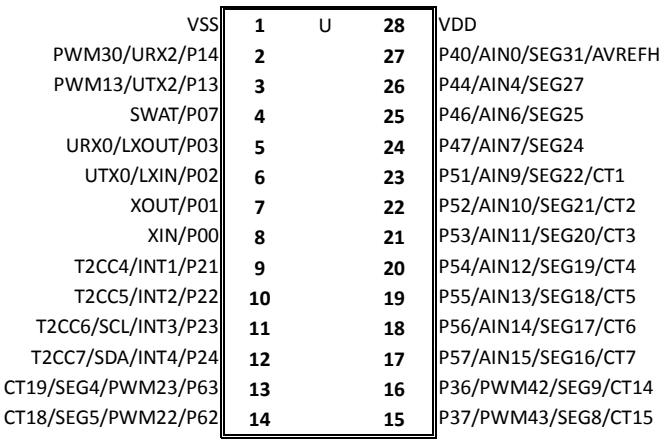
4.7 SN8F5824S (SOP28)



4.8 SN8F58242S (SOP28)

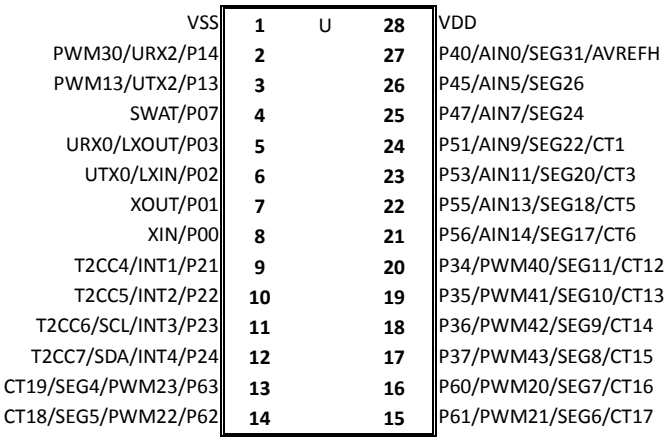


4.9 SN8F58243S (SOP28)

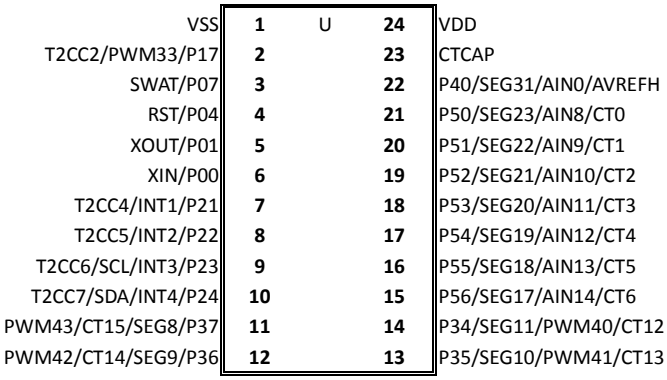


Copyright © 2019, SONiX Technology Co., Ltd.

4.10 SN8F58244S (SOP28)



4.11 SN8F5823S (SOP24)



Copyright © 2019, SONiX Technology Co., Ltd.

4.12 Pin Descriptions

Power Pins

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

Port 0

Pin Name	Type	Description
P0.0	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
XIN	Analog Input	System clock: external clock input.
P0.1	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
XOUT	Analog Output	System clock: drive external crystal/resonator.
P0.2	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
LXIN	Analog Input	Low clock: external clock input.
UTX0	Digital Output	UART0: transmission 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.
LXOUT	Analog Output	Low clock: drive external crystal.
URX0	Digital Input	UART0: reception pin.
P0.4	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
Reset	Digital Input	System reset (active low).
P0.5	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
UTX1	Digital Output	UART1: transmission pin.
P0.6	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
URX1	Digital Input	UART1: reception pin.
P0.7	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SWAT	Digital I/O	Debug interface.

Copyright © 2019, SONiX Technology Co., Ltd.

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.
PWM10	Digital Output	PWM1: programmable PWM output.
P1.1	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
PWM11	Digital Output	PWM1: programmable PWM output.
P1.2	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
PWM12	Digital Output	PWM1: programmable PWM output.
P1.3	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
PWM13	Digital Output	PWM1: programmable PWM output.
UTX2	Digital Output	UART2: transmission pin.
P1.4	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
PWM30	Digital Output	PWM3: programmable PWM output.
URX2	Digital input	UART2: reception pin.
P1.5	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
PWM31	Digital Output	PWM3: programmable PWM output.
T2CC0	Digital input	Timer 2: capture 0 input.
P1.6	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
PWM32	Digital Output	PWM3: programmable PWM output.
T2CC1	Digital input	Timer 2: capture 1 input.
P1.7	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
PWM33	Digital Output	PWM3: programmable PWM output.
T2CC2	Digital Input	Timer 2: capture 2 input.

Copyright © 2019, SONiX Technology Co., Ltd.

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.
INT0	Digital Input	INT0: external interrupt 0.
T2CC3	Digital Input	Timer 2: capture 3 input.
P2.1	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
INT1	Digital Input	INT1: external interrupt 1.
T2CC4	Digital Input	Timer 2: capture 4 input.
P2.2	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
INT2	Digital Input	INT2: external interrupt 2.
T2CC5	Digital Input	Timer 2: capture 5 input.
P2.3	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
INT3	Digital Input	INT3: external interrupt 3.
T2CC6	Digital Input	Timer 2: capture 6 input.
SCL	Digital I/O	I2C: clock output (master) or clock input (slave).
P2.4	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
INT4	Digital Input	INT4: external interrupt 4.
T2CC7	Digital Input	Timer 2: capture 7 input.
SDA	Digital I/O	I2C: data pin.

Port 3

Pin Name	Type	Description
P3.0	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SEG15	Analog Output	LCD: Segment 15 output.
CT8	Analog Input	CT8: Cap-sensing touch input channel 8.
P3.1	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SEG14	Analog Output	LCD: Segment 14 output.
CT9	Analog Input	CT8: Cap-sensing touch input channel 9.

Copyright © 2019, SONiX Technology Co., Ltd.

P3.2	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SEG13	Analog Output	LCD: Segment 13 output.
CT10	Analog Input	CT8: Cap-sensing touch input channel 10.
P3.3	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SEG12	Analog Output	LCD: Segment 12 output.
CT11	Analog Input	CT8: Cap-sensing touch input channel 11.
P3.4	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
PWM40	Digital Output	PWM4: programmable PWM output.
SEG11	Analog Output	LCD: Segment 11 output.
CT12	Analog Input	CT8: Cap-sensing touch input channel 12.
P3.5	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
PWM41	Digital Output	PWM4: programmable PWM output.
SEG10	Analog Output	LCD: Segment 10 output.
CT13	Analog Input	CT8: Cap-sensing touch input channel 13.
P3.6	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
PWM42	Digital Output	PWM4: programmable PWM output.
SEG9	Analog Output	LCD: Segment 9 output.
CT14	Analog Input	CT8: Cap-sensing touch input channel 14.
P3.7	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
PWM43	Digital Output	PWM4: programmable PWM output.
SEG8	Analog Output	LCD: Segment 8 output.
CT15	Analog Input	CT8: Cap-sensing touch input channel 15.

Port 4

Pin Name	Type	Description
P4.0	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SEG31	Analog Output	LCD: Segment 31 output.
AIN0	Analog Input	ADC: input channel 0.
AVREFH	Analog Input	ADC: external reference voltage.

Copyright © 2019, SONiX Technology Co., Ltd.

P4.1	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SEG30	Analog Output	LCD: Segment 30 output.
AIN1	Analog Input	ADC: input channel 1.
P4.2	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SEG29	Analog Output	LCD: Segment 29 output.
AIN2	Analog Input	ADC: input channel 2.
P4.3	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SEG28	Analog Output	LCD: Segment 28 output.
AIN3	Analog Input	ADC: input channel 3.
P4.4	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SEG27	Analog Output	LCD: Segment 27 output.
AIN4	Analog Input	ADC: input channel 4.
P4.5	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SEG26	Analog Output	LCD: Segment 26 output.
AIN5	Analog Input	ADC: input channel 5.
P4.6	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SEG25	Analog Output	LCD: Segment 25 output.
AIN6	Analog Input	ADC: input channel 6.
P4.7	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SEG24	Analog Output	LCD: Segment 24 output.
AIN7	Analog Input	ADC: input channel 7.

Port 5

Pin Name	Type	Description
P5.0	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SEG23	Analog Output	LCD: Segment 23 output.
AIN8	Analog Input	ADC: input channel 8.
CT0	Analog Input	CT0: Cap-sensing touch input channel 0.

Copyright © 2019, SONiX Technology Co., Ltd.

P5.1	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SEG22	Analog Output	LCD: Segment 22 output.
AIN9	Analog Input	ADC: input channel 9.
CT1	Analog Input	CT1: Cap-sensing touch input channel 1.
P5.2	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SEG21	Analog Output	LCD: Segment 21 output.
AIN10	Analog Input	ADC: input channel 10.
CT2	Analog Input	CT2: Cap-sensing touch input channel 2.
P5.3	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SEG20	Analog Output	LCD: Segment 20 output.
AIN11	Analog Input	ADC: input channel 11.
CT3	Analog Input	CT3: Cap-sensing touch input channel 3.
P5.4	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SEG19	Analog Output	LCD: Segment 19 output.
AIN12	Analog Input	ADC: input channel 12.
CT4	Analog Input	CT4: Cap-sensing touch input channel 4.
P5.5	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SEG18	Analog Output	LCD: Segment 18 output.
AIN13	Analog Input	ADC: input channel 13.
CT5	Analog Input	CT5: Cap-sensing touch input channel 5.
P5.6	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SEG17	Analog Output	LCD: Segment 17 output.
AIN14	Analog Input	ADC: input channel 14.
CT6	Analog Input	CT6: Cap-sensing touch input channel 6.
P5.7	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SEG16	Analog Output	LCD: Segment 16 output.
AIN15	Analog Input	ADC: input channel 15.
CT7	Analog Input	CT7: Cap-sensing touch input channel 7.

Copyright © 2019, SONiX Technology Co., Ltd.

Port 6

Pin Name	Type	Description
P6.0	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
PWM20	Digital Output	PWM2: programmable PWM output.
SEG7	Analog Output	LCD: Segment 7 output.
CT16	Analog Input	CT16: Cap-sensing touch input channel 16.
P6.1	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
PWM21	Digital Output	PWM2: programmable PWM output.
SEG6	Analog Output	LCD: Segment 6 output.
CT17	Analog Input	CT17: Cap-sensing touch input channel 17.
P6.2	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
PWM22	Digital Output	PWM2: programmable PWM output.
SEG5	Analog Output	LCD: Segment 5 output.
CT18	Analog Input	CT18: Cap-sensing touch input channel 18.
P6.3	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
PWM23	Digital Output	PWM2: programmable PWM output.
SEG4	Analog Output	LCD: Segment 4 output.
CT19	Analog Input	CT19: Cap-sensing touch input channel 19.
P6.4	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SEG3	Analog Output	LCD: Segment 3 output.
CT20	Analog Input	CT20: Cap-sensing touch input channel 20.
P6.5	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SEG2	Analog Output	LCD: Segment 2 output.
CT21	Analog Input	CT21: Cap-sensing touch input channel 21.
P6.6	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SEG1	Analog Output	LCD: Segment 1 output.
CT22	Analog Input	CT20: Cap-sensing touch input channel 22.
P6.7	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
SEG0	Analog Output	LCD: Segment 0 output.
CT23	Analog Input	CT20: Cap-sensing touch input channel 23.

Copyright © 2019, SONiX Technology Co., Ltd.

Port 7

Pin Name	Type	Description
P7.0	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
COM0	Analog Output	LCD: Common 0 output.
P7.1	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
COM1	Analog Output	LCD: Common 1 output.
P7.2	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
COM2	Analog Output	LCD: Common 2 output.
P7.3	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
COM3	Analog Output	LCD: Common 3 output.
P7.4	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
COM4	Analog Output	LCD: Common 4 output.
SEG35	Analog Output	LCD: Segment 35 output.
P7.5	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
COM5	Analog Output	LCD: Common 5 output.
SEG34	Analog Output	LCD: Segment 34 output.
P7.6	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
COM6	Analog Output	LCD: Common 6 output.
SEG33	Analog Output	LCD: Segment 33 output.
P7.7	Digital I/O	GPIO: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
COM7	Analog Output	LCD: Common 7 output.
SEG32	Analog Output	LCD: Segment 32 output.

Copyright © 2019, SONiX Technology Co., Ltd.

4.13 Pin Characteristic

Port	Open-Drain	Sink Current 200mA VSS+1.5V	Sink Current 400mA VSS+1.5V	External Interrupt	Wakeup (Level change)	Shared Pin
P0.0	-	-	-	-	V	XIN
P0.1	-	-	-	-	V	XOUT
P0.2	V	-	-	-	V	LXIN/UTX0
P0.3	V	-	-	-	V	LXOUT/URX0
P0.4	-	-	-	-	V	RST
P0.5	V	-	-	-	V	UTX1
P0.6	V	-	-	-	V	URX1
P0.7	-	-	-	-	V	SWAT
P1.0	-	-	-	-	V	PWM10
P1.1	-	-	-	-	V	PWM11
P1.2	-	-	-	-	V	PWM12
P1.3	V	-	-	-	V	UTX2/PWM13
P1.4	V	-	-	-	V	URX2/PWM30
P1.5	-	-	-	-	V	PWM31/T2CC0
P1.6	-	-	-	-	V	PWM32/T2CC1
P1.7	-	-	V	-	V	PWM33/T2CC2
P2.0	-	-	-	V	-	INT0/T2CC3
P2.1	-	-	-	V	-	INT1/T2CC4
P2.2	-	-	-	V	-	INT2/T2CC5
P2.3	-	-	-	V	-	INT3/T2CC6/SCL
P2.4	-	-	-	V	-	INT4/T2CC7/SDA
P3.0	-	-	-	-	-	SEG15/CT8
P3.1	-	-	-	-	-	SEG14/CT9
P3.2	-	-	-	-	-	SEG13/CT10
P3.3	-	-	-	-	-	SEG12/CT11
P3.4	-	-	-	-	-	PWM40/SEG11/CT12
P3.5	-	-	-	-	-	PWM41/SEG10/CT13
P3.6	-	-	-	-	-	PWM42/SEG9/CT14
P3.7	-	-	-	-	-	PWM43/SEG8/CT15
P4.0	-	-	-	-	-	SEG31/AIN0/AVREFH
P4.1	-	-	-	-	-	SEG30/AIN1
P4.2	-	-	-	-	-	SEG29/AIN2
P4.3	-	-	-	-	-	SEG28/AIN3
P4.4	-	-	-	-	-	SEG27/AIN4
P4.5	-	-	-	-	-	SEG26/AIN5
P4.6	-	-	-	-	-	SEG25/AIN6
P4.7	-	-	-	-	-	SEG24/AIN7
P5.0	-	-	-	-	-	SEG23/AIN8/CT0
P5.1	-	-	-	-	-	SEG22/AIN9/CT1
P5.2	-	-	-	-	-	SEG21/AIN10/CT2

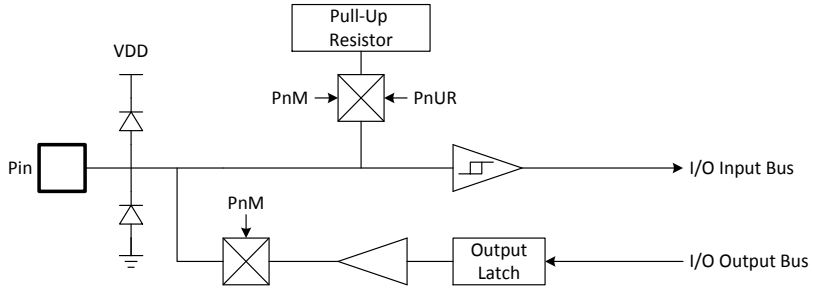
Copyright © 2019, SONiX Technology Co., Ltd.

P5.3	-	-	-	-	-	SEG20AIN11/CT3
P5.4	-	-	-	-	-	SEG19/AIN12/CT4
P5.5	-	-	-	-	-	SEG18/AIN13/CT5
P5.6	-	-	-	-	-	SEG17/AIN14/CT6
P5.7	-	-	-	-	-	SEG16/AIN15/CT7
P6.0	-	-	-	-	-	PWM20/SEG7/CT16
P6.1	-	-	-	-	-	PWM21/SEG6/CT17
P6.2	-	-	-	-	-	PWM22/SEG5/CT18
P6.3	-	-	-	-	-	PWM23/SEG4/CT19
P6.4	-	-	-	-	-	SEG3/CT20
P6.5	-	-	-	-	-	SEG2/CT21
P6.6	-	-	-	-	-	SEG1/CT22
P6.7	-	-	-	-	-	SEG0/CT23
P7.0	-	V	-	-	-	COM0
P7.1	-	V	-	-	-	COM1
P7.2	-	V	-	-	-	COM2
P7.3	-	V	-	-	-	COM3
P7.4	-	V	-	-	-	COM4/SEG35
P7.5	-	V	-	-	-	COM5/SEG34
P7.6	-	V	-	-	-	COM6/SEG33
P7.7	-	V	-	-	-	COM7/SEG32

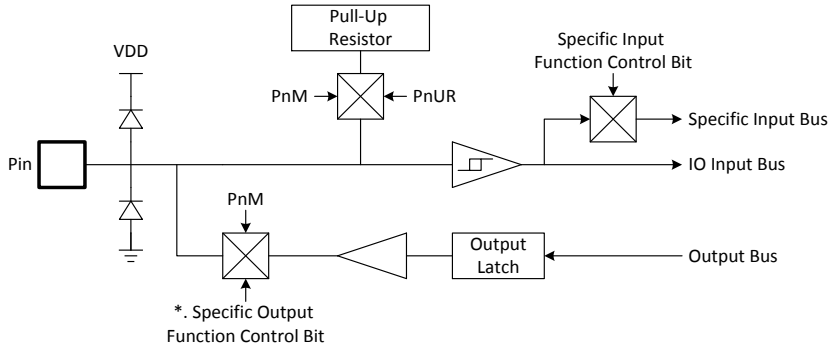
Copyright © 2019, SONiX Technology Co., Ltd.

4.14 Pin Circuit Diagrams

Normal Bi-direction I/O Pin.

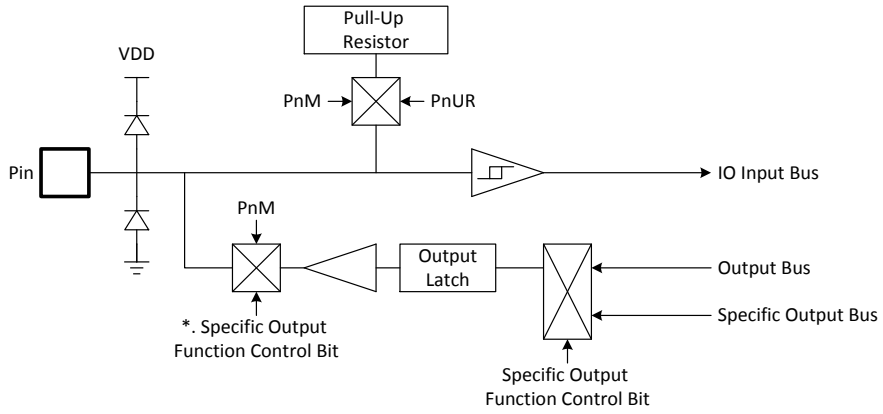


Bi-direction I/O Pin Shared with Specific Digital Input Function, e.g. INT2.



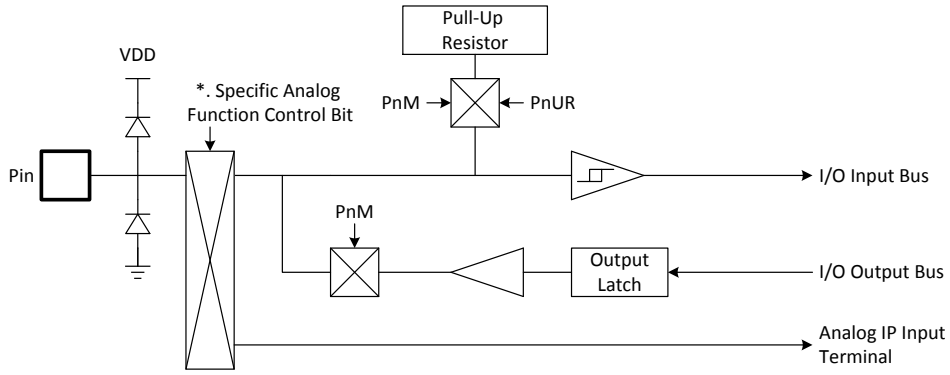
***. Some specific functions switch I/O direction directly, not through PnM register.**

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



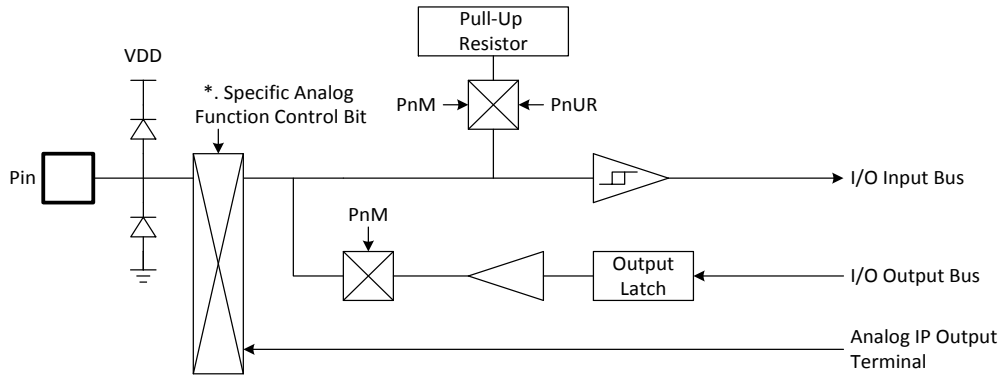
***. Some specific functions switch I/O direction directly, not through PnM register.**

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…



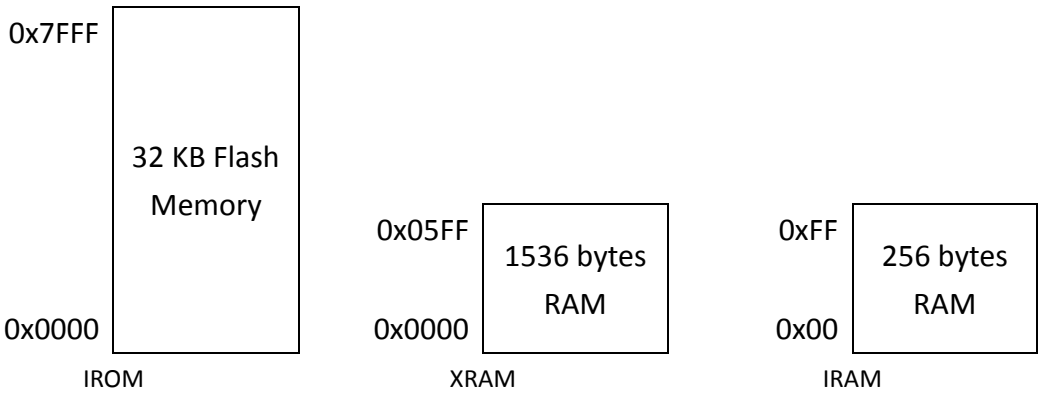
***. Some specific functions switch I/O direction directly, not through PnM register.**

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

SN8F5829 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 1536-byte of size, but it requires a longer access period. The program memory is a 32 KB non-volatile memory and has a maximum 16 MHz speed limitation.



5.2 Internal RAM (IRAM)

256 X 8-bit RAM (Internal Data Memory)

Address	RAM Location		
000h 01Fh	Work Register Area		00h-7Fh of RAM is direct and indirect access RAM
020h 02Fh	Bit Addressable Area		
030h … … 07Fh	General Purpose Area		
080h … … … 0FFh	General Purpose Area (Indirect Access)	Special Function Register (Direct Access)	080h-0FFh store special function registers. End of Bank 0

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	Bite 0	Bite 1	Bite 2	Bite 3	Bite 4	Bite 5	Bite 6	Bite 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

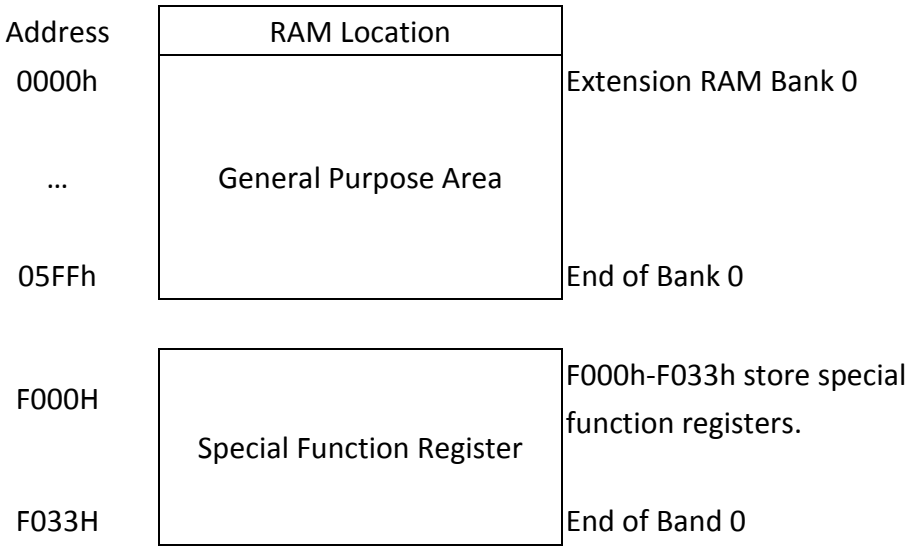
- 0x0080~0x00FF: General purpose area in indirect addressing access or special function register in direct addressing access.

Copyright © 2019, SONiX Technology Co., Ltd.

5.3 External RAM (XRAM)

1536 X 8-bit XRAM (Extension Data Memory)

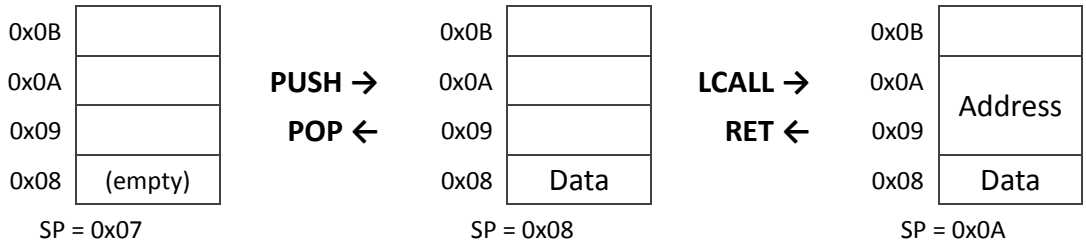
The external RAM enlarges the capacity of variables; it is the lowest access performance in the contrast of internal RAM. Since frequently used variables and local variables are expected to store in internal RAM, the vast majority of external RAM usages are specific. It can be allocated as a variable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



The upper 1536-byte XRAM is general purpose RAM and can configure by MOVX instruction access. The lower 52 bytes of XRAM are special function registers (SFRs) that can also be accessed by MOVX instruction.

5.4 Stack

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



By default, stack pointer (SP register) points to 0x07 which means the stack area begin at IRAM

address 0x08. In other word, if a planned stack area is assigned from IRAM address 0xC0, the appropriate SP register is anticipated to set at 0xBF after system reset.

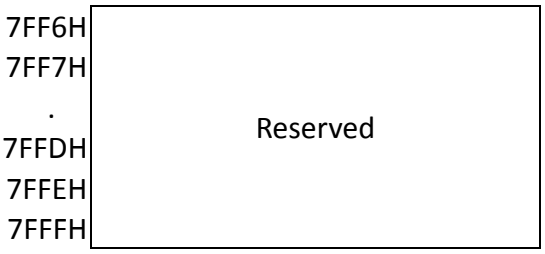
An assembly PUSH instruction costs one byte of stack. LCALL, ACALL instructions and interrupt respectively costs two bytes stack. POP-instruction decreases one count, and a RET/RETI subtract two counts of stack pointer.

5.5 Program Memory (IROM)

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

Address	ROM	Comment
0000H	Reset vector	Reset vector
0001H 0002H	General purpose area	User program
0003H	**INT0 Interrupt vector**	**Interrupt vector**
000BH	**UART1 TX Interrupt vector**	
0013H	**UART0 TX Interrupt vector**	
001BH	**ADC Interrupt vector**	
0023H	**PWM1 Interrupt vector**	
002BH	**TIMER0 Interrupt vector**	
0033H	**INT3 Interrupt vector**	
003BH	**UART2 TX Interrupt vector**	
0043H	**INT2 Interrupt vector**	
004BH	**UART1 RX Interrupt vector**	
0053H	**UART0 RX Interrupt vector**	
005BH	**LCD Interrupt vector**	
0063H	**PWM2 Interrupt vector**	
006BH	**TIMER3 Interrupt vector**	
0083H	**INT1 Interrupt vector**	
008BH	**UART2 RX Interrupt vector**	
0093H	**I2C Interrupt vector**	
00A3H	**PWM4 Interrupt vector**	
00ABH	**TIMER1 Interrupt vector**	
00B3H	**PWM3 Interrupt vector**	
00BBH	**TIMER4 Interrupt vector**	
00C3H	**INT4 Interrupt vector**	
00EBH	**TIMER2 Interrupt vector**	
00ECH	General purpose area	User program
.		
.		
.		
.		End of user program

Copyright © 2019, SONiX Technology Co., Ltd.



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

- 0x0000 Reset vector: Program counter points to 0x0000 after any reset events (power on reset, reset pin reset, watchdog reset, LVD reset…).
- 0x0001~0x0002: General purpose area to process system reset operation.
- 0x0003~0x00EB: Multi interrupt vector area. Each of interrupt events has a unique interrupt vector.
- 0x00EC~0x7FBF: General purpose area for user program and ISP (EEPROM function).
- 0x7FC0~0x7FF6: General purpose area for user program. Do not execute ISP.
- 0x7FF6~0x7FFF: Reserved area. Do not execute ISP.

5.6 Program Memory Security

The SN8F5829 provides security options at the disposal of the designer to prevent unauthorized access to information stored in FLASH memory. When enable security option, the ROM code is secured and not dumped complete ROM contents. ROM security rule is all address ROM data protected and outputs 0x00.

5.7 Data Pointer

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

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

The automatically increase/decrease DPTR function can make an increment or decrement after perform MOVX @DPTR instruction. As a result, it enables a continuous external RAM access

Copyright © 2019, SONiX Technology Co., Ltd.

without re-specified DPTR value. Those functions are controlled by the DPC Register, where there are separate DPC register bits for each DPTR, to provide high flexibility in data transfers. The DPC Register address 0x93 points to the window where the actual DPC is selected using the DPS Register, same as for the DPTR.

5.8 Stack and Data Pointer Register

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

SP Register (0x81)

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

DPL Register (0x82)

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

DPH Register (0x83)

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

Copyright © 2019, SONiX Technology Co., Ltd.

DPL1 Register (0x84)

Bit	Field	Type	Initial	Description
7..0	DPL1[7:0]	R/W	0x00	Low byte of DPTR1

DPH1 Register (0x85)

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

DPS Register (0x92)

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

DPC Register (0x93)

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

Copyright © 2019, SONiX Technology Co., Ltd.

6 Special Function Registers

The SN8F5829 offers many functions that can be controlled via special function registers. Special function registers are placed at both the IRAM address and the XRAM address. The IRAM registers can be accessed by direct addressing, while the XRAM registers are accessed via the MOVX instruction. Register description will be classified according to the addressing range. The 8-bit address refers to the IRAM register and the 16-bit address refers to the XRAM register.

6.1 IRAM Special Function Register Memory Map

BIN / HEX	000	001	010	011	100	101	110	111
F8	P7	P0M	P1M	P2M	P3M	P4M	P5M	PFLAG
F0	B	-	-	-	T4M	T4CL	T4CH	SRST
E8	P6	MD0	MD1	MD2	MD3	MD4	MD5	ARCON
E0	ACC	T4RL	T4RH	P6M	P7M	CLKSEL	CLKCMD	TCON0
D8	P5	-	I2CDAT	I2CADR	I2CCON	I2CSTA	SMBSEL	SMBDST
D0	PSW	P0OC	ADM	ADB	ADR	VREFH	P6CON	P7CON
C8	P4	S2CON	S2BUF	S2RELL	S2RELH	FRQL	FRQH	P5CON
C0	IRCON	S1CON	S1BUF	S1RELL	S1RELH	FRQCMD	SYSMOD	PEDGE1
B8	IEN1	IP1	-	-	-	-	-	IRCON2
B0	P3	T3M	T3CL	T3CH	T3RL	T3RH	CPTCL	CPTCH
A8	IEN0	IP0	LCDSEG3	LCDSEG4	T2M	T2CL	T2CH	CPTM
A0	P2	LCDCON	LCDMOD	LCDSEG	LCDBUF	LCDADR	LCDSEG1	LCDSEG2
98	S0CON	S0BUF	IEN2	S0CON2	S0RELL	S0RELH	P3CON	P4CON
90	P1	P1W	DPS	DPC	PECMD	PEROML	PEROMH	PERAM
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	PEDGE
80	P0	SP	DPL	DPH	DPL1	DPH1-	WDTR	PCON

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

Copyright © 2019, SONiX Technology Co., Ltd.

6.2 IRAM Special Function Register Description

0x80 - 0x9F Registers Description

Register	Address	Description
P0	0x80	Port 0 data buffer.
SP	0x81	Stack pointer register.
DPL	0x82	Data pointer 0 low byte register.
DPH	0x83	Data pointer 0 high byte register.
DPL1	0x84	Data pointer 1 low byte register.
DPH1	0x85	Data pointer 1 high byte register.
WDTR	0x86	Watchdog timer clear register.
PCON	0x87	System mode register.
TCON	0x88	Timer 0 / 1 controls register.
TMOD	0x89	Timer 0 / 1 mode register.
TL0	0x8A	Timer 0 counting low byte register.
TL1	0x8B	Timer 1 counting low byte register.
TH0	0x8C	Timer 0 counting high byte register.
TH1	0x8D	Timer 1 counting high byte register.
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.
DPS	0x92	Data pointer selects register.
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	UART0 control register.
S0BUF	0x99	UART0 data buffer.
IEN2	0x9A	Interrupts enable register.
S0CON2	0x9B	UARTs baud rate controls register.
S0RELL	0x9C	UART0 reload low byte register.
S0RELH	0x9D	UART0 reload high byte register.
P3CON	0x9E	Port 3 configuration controls register.
P4CON	0x9F	Port 4 configuration controls register.

Copyright © 2019, SONiX Technology Co., Ltd.

0xA0 - 0xBF Registers Description

Register	Address	Description
P2	0xA0	Port 2 data buffer.
LCDCON	0xA1	LCD controls register.
LCDMOD	0xA2	LCD mode control.
LCDSEG	0xA3	LCD segment pin control.
LCDBUF	0xA4	LCD data buffer.
LCDADR	0xA5	LCD data buffer address.
LCDSEG1	0xA6	LCD segment pin control.
LCDSEG2	0xA7	LCD segment pin control.
IEN0	0xA8	Interrupts enable register.
IP0	0xA9	Interrupts priority register.
LCDSEG3	0xAA	LCD segment pin control.
LCDSEG4	0xAB	LCD segment pin control.
T2M	0xAC	Timer 2 controls register.
T2CL	0xAD	Timer 2 counting low byte register.
T2CH	0xAE	Timer 2 counting high byte register.
CPTM	0xAF	Capture function of Timer 2 control register.
P3	0xB0	Port 3 data buffer.
T3M	0xB1	Timer 3 controls register.
T3CL	0xB2	Timer 3 counting low byte register.
T3CH	0xB3	Timer 3 counting high byte register.
T3RL	0xB4	Timer 3 cycle controls buffer low byte.
T3RH	0xB5	Timer 3 cycle controls buffer high byte
CPTCL	0xB6	Timer 2 capture buffer low byte.
CPTCH	0xB7	Timer 2 capture buffer high byte.
IEN1	0xB8	Interrupts enable register.
IP1	0xB9	Interrupts priority register.
-	0xBA	-
-	0xBB	-
-	0xBC	-
-	0xBD	-
-	0xBE	-
IRCON2	0xBF	Interrupts request register.

Copyright © 2019, SONiX Technology Co., Ltd.

0xC0 - 0xDF Registers Description

Register	Address	Description
IRCON	0xC0	Interrupts request register.
S1CON	0xC1	UART1 control register.
S1BUF	0xC2	UART1 data buffer.
S1RELL	0xC3	UART1 reload low byte register.
S1RELH	0xC4	UART1 reload high byte register.
FRQCMD	0xC5	Clock fine tuning command register.
SYSMOD	0xC6	System controls register
PEDGE1	0xC7	External interrupt edge controls register.
P4	0xC8	Port 4 data buffer.
S2CON	0xC9	UART2 control register.
S2BUF	0xCA	UART2 data buffer.
S2RELL	0xCB	UART2 reload low byte register.
S2RELH	0xCC	UART2 reload high byte register.
FRQL	0xCD	Clock fine tuning controls buffer low byte
FRQH	0xCE	Clock fine tuning controls buffer high byte
P5CON	0xCF	Port 5 configuration controls register.
PSW	0xD0	System flag register.
P0OC	0xD1	Open drain controls register.
ADM	0xD2	ADC controls register.
ADB	0xD3	ADC data buffer.
ADR	0xD4	ADC resolution selects register.
VREFH	0xD5	ADC reference voltage controls register.
P6CON	0xD6	Port 6 configuration controls register.
P7CON	0xD7	Port 7 configuration controls register.
P5	0xD8	Port 5 data buffer.
-	0xD9	-
I2CDAT	0xDA	I2C data buffer.
I2CADR	0xDB	Own I2C slave address.
I2CCON	0xDC	I2C interface operation control register.
I2CSTA	0xDD	I2C Status Code.
SMBSEL	0xDE	SMBus mode controls register.
SMBDST	0xDF	SMBus internal timeout register.

Copyright © 2019, SONiX Technology Co., Ltd.

0xE0 - 0xFF Registers Description

Register	Address	Description
ACC	0xE0	Accumulator register.
T4RL	0xE1	Timer 4 cycle controls buffer low byte.
T4RH	0xE2	Timer 4 cycle controls buffer high byte
P6M	0xE3	Port 6 input/output mode register.
P7M	0xE4	Port 7 input/output mode register.
CLKSEL	0xE5	Clock switch selects register.
CLKCMD	0xE6	Clock switch controls Register.
TCON0	0xE7	Timer 0 / 1 clock controls register.
P6	0xE8	Port 6 data buffer.
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.
-	0xF1	-
-	0xF2	-
-	0xF3	-
T4M	0xF4	Timer 4 controls register.
T4CL	0xF5	Timer 4 counting low byte register.
T4CH	0xF6	Timer 4 counting high byte register.
SRST	0xF7	Software reset controls register.
P7	0xF8	Port 7 data buffer.
P0M	0xF9	Port 0 input/output mode register.
P1M	0xFA	Port 1 input/output mode register.
P2M	0xFB	Port 2 input/output mode register.
P3M	0xFC	Port 3 input/output mode register.
P4M	0xFD	Port 4 input/output mode register.
P5M	0xFE	Port 5 input/output mode register.
PFLAG	0xFF	Reset flag register.

Copyright © 2019, SONiX Technology Co., Ltd.

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.

Copyright © 2019, SONiX Technology Co., Ltd.

PSW Register (0xD0)

Bit	Field	Type	Initial	Description
7	CY	R/W	0	Carry flag. 0: Addition without carry, subtraction with borrowing signal, rotation with shifting out logic “0”, comparison result < 0. 1: Addition with carry, subtraction without borrowing, rotation with shifting out logic “1”, comparison result ≥ 0.
6	AC	R/W	0	Auxiliary carry flag. 0: If there is no a carry-out from 3rd bit of Accumulator in BCD operations. 1: If there is a carry-out from 3rd bit of Accumulator in BCD operations.
5	F0	R/W	0	General purpose flag 0. General purpose flag available for user.
4..3	RS[1:0]	R/W	00	Register bank select control bit, used to select working register bank. 00: 00H – 07H (Bnak0) 01: 08H – 0FH (Bnak1) 10: 10H – 17H (Bnak2) 11: 18H – 1FH (Bnak3)
2	OV	R/W	0	Overflow flag. 0: Non-overflow in Accumulator during arithmetic Operations. 1: overflow in Accumulator during arithmetic Operations.
1	F1	R/W	0	General purpose flag 1. General purpose flag available for user.
0	P	R	0	Parity flag. Reflects the number of ‘1’s in the Accumulator. 0: if Accumulator contains an even number of ‘1’s. 1: Accumulator contains an odd number of ‘1’s.

Copyright © 2019, SONiX Technology Co., Ltd.

6.4 XRAM Special Function Register Memory Map

BIN / HEX	000	001	010	011	100	101	110	111
F078	-	-	-	-	-	-	-	-
F070	-	-	-	-	-	-	-	-
F068	-	-	-	-	-	-	-	-
F060	-	-	-	-	-	-	-	-
F058	-	-	-	-	-	-	-	-
F050	-	-	-	-	-	-	-	-
F048	-	-	-	-	-	-	-	-
F040	-	-	-	-	-	-	-	-
F038	-	-	-	-	-	-	-	-
F030	PW3YH	PW3YL	PW4YH	PW4YL	-	-	-	-
F028	P0UR	P1UR	P2UR	P3UR	P4UR	P5UR	P6UR	P7UR
F020	PW40DH	PW40DL	PW41DH	PW41DL	PW42DH	PW42DL	PW43DH	PW43DL
F018	PW20DH	PW20DL	PW21DH	PW21DL	PW22DH	PW22DL	PW23DH	PW23DL
F010	PW32DH	PW32DL	PW33DH	PW33DL	PW2M	PW4M	PW2YH	PW2YL
F008	PW12DH	PW12DL	PW13DH	PW13DL	PW30DH	PW30DL	PW31DH	PW31DL
F000	PW1M	PW3M	PW1YH	PW1YL	PW10DH	PW10DL	PW11DH	PW11DL

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

Copyright © 2019, SONiX Technology Co., Ltd.

6.5 XRAM Special Function Register Description

0xF000 - 0xF01F Registers Description

Register	Address	Description
PW1M	0xF000	PWM1 controls register.
PW3M	0xF001	PWM3 controls register.
PW1YH	0xF002	PWM1 cycle controls buffer high byte.
PW1YL	0xF003	PWM1 cycle controls buffer low byte.
PW10DH	0xF004	PWM10 duty controls buffer high byte.
PW10DL	0xF005	PWM10 duty controls buffer low byte.
PW11DH	0xF006	PWM11 duty controls buffer high byte.
PW11DL	0xF007	PWM11 duty controls buffer low byte.
PW12DH	0xF008	PWM12 duty controls buffer high byte.
PW12DL	0xF009	PWM12 duty controls buffer low byte.
PW13DH	0xF00A	PWM13 duty controls buffer high byte.
PW13DL	0xF00B	PWM13 duty controls buffer low byte.
PW30DH	0xF00C	PWM30 duty controls buffer high byte.
PW30DL	0xF00D	PWM30 duty controls buffer low byte.
PW31DH	0xF00E	PWM31 duty controls buffer high byte.
PW31DL	0xF00F	PWM31 duty controls buffer low byte.
PW32DH	0xF010	PWM32 duty controls buffer high byte.
PW32DL	0xF011	PWM32 duty controls buffer low byte.
PW33DH	0xF012	PWM33 duty controls buffer high byte.
PW33DL	0xF013	PWM33 duty controls buffer low byte.
PW2M	0xF014	PWM2 controls register.
PW4M	0xF015	PWM4 controls register.
PW2YH	0xF016	PWM2 cycle controls buffer high byte.
PW2YL	0xF017	PWM2 cycle controls buffer low byte.
PW20DH	0xF018	PWM20 duty controls buffer high byte.
PW20DL	0xF019	PWM20 duty controls buffer low byte.
PW21DH	0xF01A	PWM21 duty controls buffer high byte.
PW21DL	0xF01B	PWM21 duty controls buffer low byte.
PW22DH	0xF01C	PWM22 duty controls buffer high byte.
PW22DL	0xF01D	PWM22 duty controls buffer low byte.
PW23DH	0xF01E	PWM23 duty controls buffer high byte.
PW23DL	0xF01F	PWM23 duty controls buffer low byte.

Copyright © 2019, SONiX Technology Co., Ltd.

0xF020 - 0xF03F Registers Description

Register	Address	Description
PW40DH	0xF020	PWM40 duty controls buffer high byte.
PW40DL	0xF021	PWM40 duty controls buffer low byte.
PW41DH	0xF022	PWM41 duty controls buffer high byte.
PW41DL	0xF023	PWM41 duty controls buffer low byte.
PW42DH	0xF024	PWM42 duty controls buffer high byte.
PW42DL	0xF025	PWM42 duty controls buffer low byte.
PW43DH	0xF026	PWM43 duty controls buffer high byte.
PW43DL	0xF027	PWM43 duty controls buffer low byte.
P0UR	0xF028	Port 0 pull-up resister controls register.
P1UR	0xF029	Port 1 pull-up resister controls register.
P2UR	0xF02A	Port 2 pull-up resister controls register.
P3UR	0xF02B	Port 3 pull-up resister controls register.
P4UR	0xF02C	Port 4 pull-up resister controls register.
P5UR	0xF02D	Port 5 pull-up resister controls register.
P6UR	0xF02E	Port 6 pull-up resister controls register.
P7UR	0xF02F	Port 7 pull-up resister controls register.
PW3YH	0xF030	PWM3 cycle controls buffer high byte.
PW3YL	0xF031	PWM3 cycle controls buffer low byte.
PW4YH	0xF032	PWM4 cycle controls buffer high byte.
PW4YL	0xF033	PWM4 cycle controls buffer low byte.
-	0xF034	-
-	0xF035	-
-	0xF036	-
-	0xF037	-
-	0xF038	-
-	0xF039	-
-	0xF03A	-
-	0xF03B	-
-	0xF03C	-
-	0xF03D	-
-	0xF03E	-
-	0xF03F	-

Copyright © 2019, SONiX Technology Co., Ltd.

6.6 Register Declaration

SN8F5829 has many registers to control various functions, but SFR name is not predefined in the C51 / A51 compiler. To make programming easier and therefore need to add header files to declare SFR name.

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

```
$NOMOD51 ; Do not recognize the 8051-specific predefined special register.
#define __XRAM_SFR_H__ ; For XRAM Register Declaration.
#include <SN8F5829.H>
```

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

```
#define __XRAM_SFR_H__ // For XRAM Register Declaration.
#include <SN8F5829.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
SN8F5829	SN8F5829.h	OPTIONS_SN8F5829.A51
SN8F5828	SN8F5828.h	OPTIONS_SN8F5829.A51
SN8F5825	SN8F5825.h	OPTIONS_SN8F5829.A51
SN8F58252	SN8F58252.h	OPTIONS_SN8F5829.A51
SN8F58253	SN8F58253.h	OPTIONS_SN8F5829.A51
SN8F58254	SN8F58254.h	OPTIONS_SN8F5829.A51
SN8F5824	SN8F5824.h	OPTIONS_SN8F5829.A51
SN8F58242	SN8F58242.h	OPTIONS_SN8F5829.A51
SN8F58243	SN8F58243.h	OPTIONS_SN8F5829.A51
SN8F58244	SN8F58244.h	OPTIONS_SN8F5829.A51
SN8F5823	SN8F5823.h	OPTIONS_SN8F5829.A51

✷ ***Note:***

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

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

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

6.7 XRAM Register Description

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

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

```
MOV     DPTR, #PW1M
MOV     A, #08H
MOVX    @DPTR, A

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

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

```
PW1M = 0x08;
TMP = P0UR; // TMP is direct addressing variable
```

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

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

Example:

```
WRXSFR  PW1M, #08H
WRXSFR  P1UR, B
WRXSFR  PW2YH, TMP ; TMP is direct addressing variable

RDXSFR  B, PW1M
RDXSFR  TMP, P0UR ; TMP is direct addressing variable
```

✸ ***Note:***

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

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

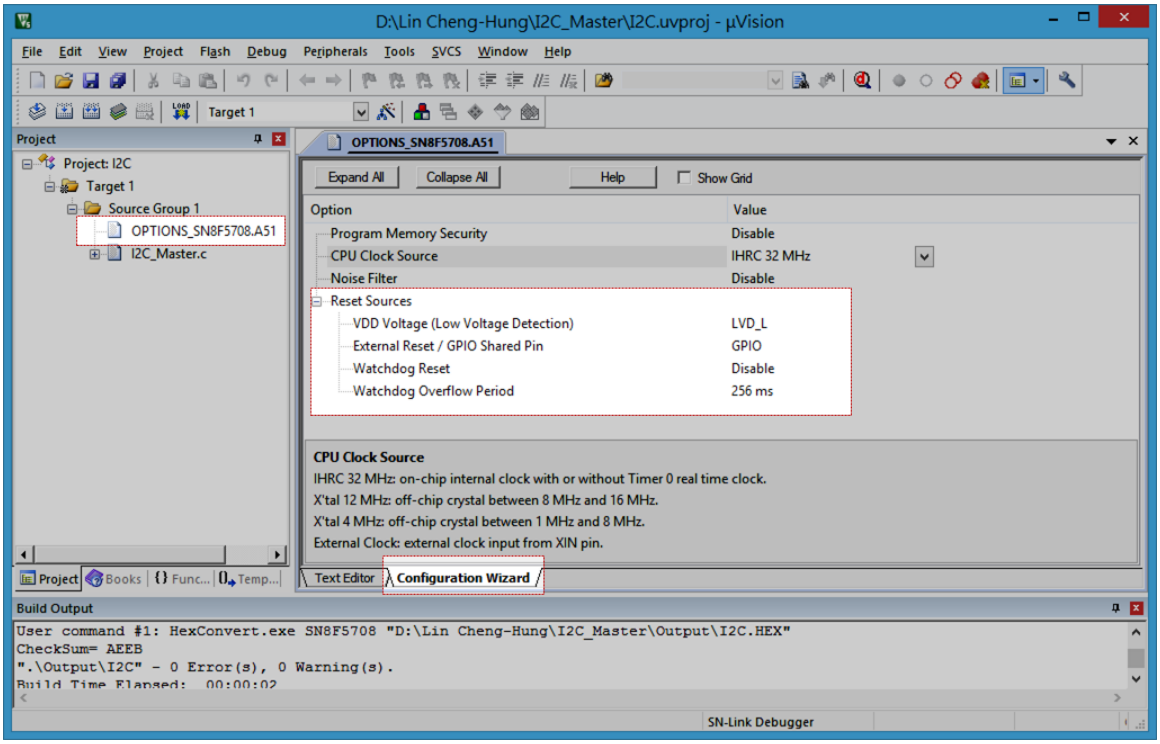
7 Reset and Power-on Controller

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

7.1 Configuration of Reset and Power-on Controller

SONiX publishes a *SN8F5829_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 : VDD Voltage (Low Voltage Detection)
- Reset Source : External Reset / GPIO Shared Pin
- Reset Source : Watchdog Reset & Overflow Period



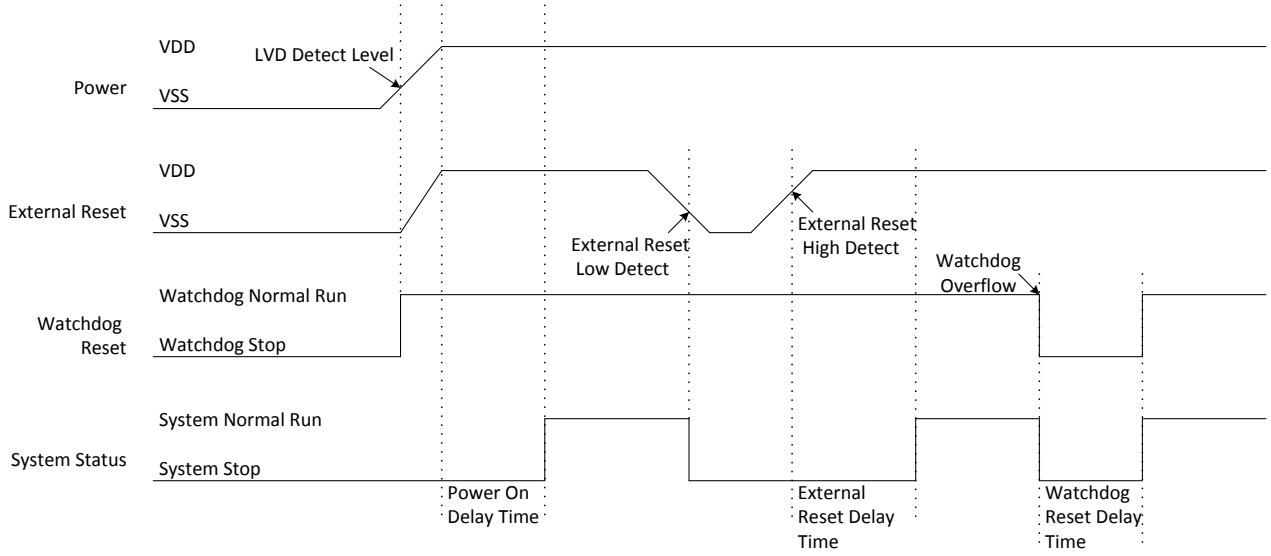
The code option is the system hardware configurations including oscillator type, noise filter option, watchdog timer operation, LVD option, reset pin option and flash ROM security control. The code option items are as following table:

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 0x7E00 ~ 0x7FBF, only address 0x7E00 ~ 0x7FBF ROM data can be accessed
CPU Clock Source	IHRC 32MHz	High speed internal 32MHz RC. XIN/XOUT pins are bi-direction GPIO mode
	X'tal 12MHz	High speed crystal /resonator (e.g. 12MHz) for external high clock oscillator
	X'tal 4MHz	Standard crystal /resonator (e.g. 4M) for external high clock oscillator
	External Clock	XIN pin connect external clock (1M ~32M), XOUT pin is bi-direction GPIO mode
Noise Filter	Disable	Disable Noise Filter
	Enable	Enable Noise Filter
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 504~ Page 510	Only address 0x7E00 ~ 0x7FBF can perform ISP function
LVD	LVD_L	LVD will reset chip if VDD is below 1.8V
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 P04	Enable P04
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} /4
	128ms	Watchdog timer clock source F_{ILRC} /8
	256ms	Watchdog timer clock source F_{ILRC} /16
	512ms	Watchdog timer clock source F_{ILRC} /32

Copyright © 2019, SONiX Technology Co., Ltd.

7.2 Power-on Sequence

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

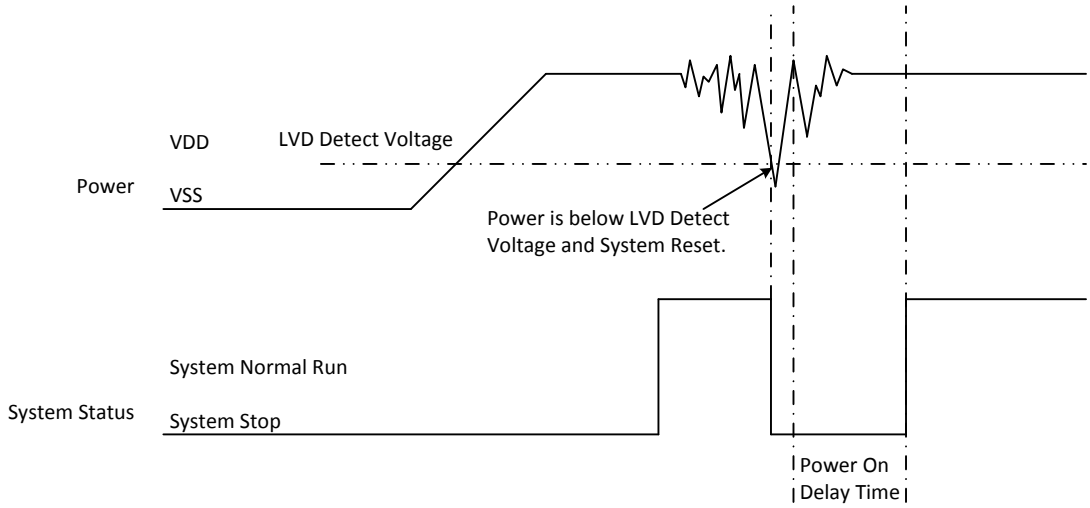


The power stabilization period spends 4.6 ms in typical condition. Afterward the microcontroller fetches CPU Clock Source selection automatically. The selected clock source would be driven, and the system counts 2048 times of the clock period and 5 times of the internal low-speed oscillator clocks to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

Copyright © 2019, SONiX Technology Co., Ltd.

7.3 LVD Reset

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



Condition	LVD_L
VDD ≤ 1.8 V	Reset

7.4 Watchdog Reset

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

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

Watchdog timer interval time = 256 * 1/ (Internal Low-Speed oscillator frequency/WDT Pre-scalar)
= 256 / (F_{ILRC} /WDT Pre-scaler) …sec

Internal low-speed oscillator	WDT pre-scaler	Watchdog interval time
F_{ILRC} =16 kHz	F_{ILRC} /4	256/(16000/4)=64ms
	F_{ILRC} /8	256/(16000/8)=128ms
	F_{ILRC} /16	256/(16000/16)=256ms
	F_{ILRC} /32	256/(16000/32)=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.

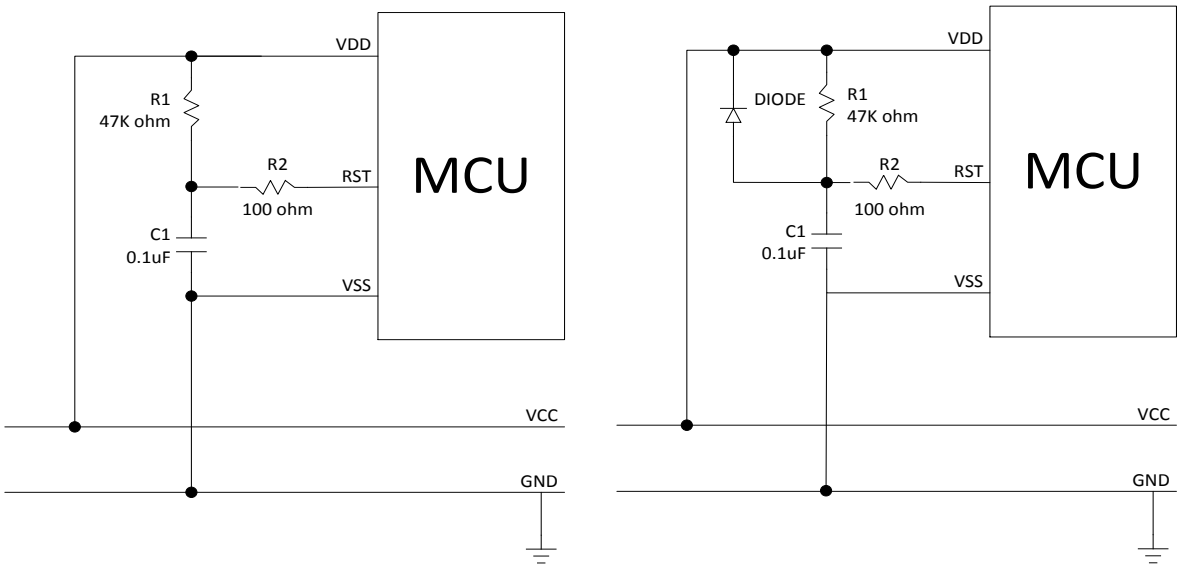
When watchdog is operating in always mode, the system will consume additional power.

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.

Copyright © 2019, SONiX Technology Co., Ltd.



✸ ***Note:***

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

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

7.6 Software Reset

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

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

7.7 Reset and Power-on Controller Registers

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

PFLAG Register

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

SRST Register

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

WDTR Register (0x86)

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

Copyright © 2019, SONiX Technology Co., Ltd.

8 System Clock and Power Management

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

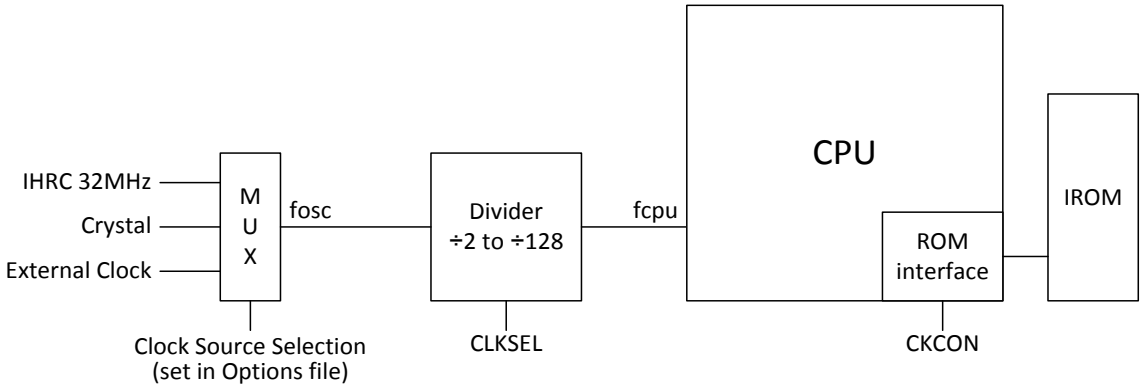
The normal mode means that CPU and peripheral functions are under normally execution. The system clock is based on the combination of source selection, clock divider, and program memory wait state. IDLE mode is the situation that temporarily suspends CPU clock and its execution, yet it remains peripherals'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

8.1 System Clock

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

Subsequently, the selected clock source (fosc) is divided by 2 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.

```
1     CKCON = 0x70;     // For change safely the system clock
2     CLKSEL = 0x86;    // Set fcpu = fosc / 2
3     CLKCMD = 0x69;    // Apply CLKSEL's setting
4     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) ≦ 16MHz

✷ ***Note: For user develop program in C language or assembly, the first line of the program "must be set" CLKSEL= 0x86~0x80, CLKMD= 0x69 and then set CKCON= 0x00~0x70, this priority cannot be modified.***

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

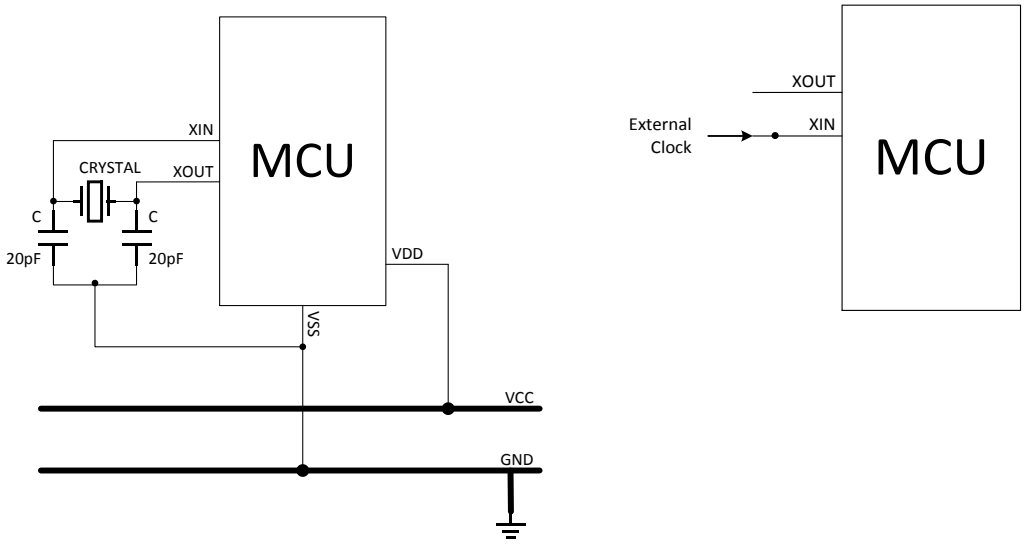
Code Option CPU Clock Source	Fcpu = CLKSEL[2:0]	IROM Fetch = CKCON[6:4]
IHRC 32M X'tal 12M (Crystal 4-16MHz) X'tal 4M (Crystal 1-4MHz) External Clock (1-32MHz)	**Only Support** 000 = fosc / 128 001 = fosc / 64 010 = fosc / 32 011 = fosc / 16 100 = fosc / 8 101 = fosc / 4 110 = fosc / 2	**Only Support**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

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 *SN8F5829_OPTIONS.A51*.

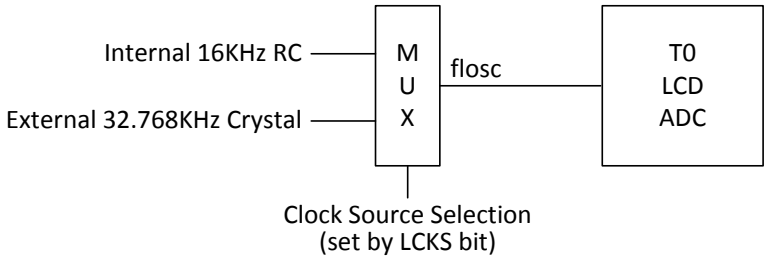
- IHRC 32M: The system high-speed clock source is internal high-speed 32MHz RC type oscillator. In the mode, XIN and XOUT pins are bi-direction GPIO mode, and not to connect any external oscillator device.
- X'tal 12M: The system high-speed clock source is external high-speed crystal/ceramic. The oscillator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

Copyright © 2019, SONiX Technology Co., Ltd.



8.3 Low Speed Clock

SN8F5829 supplies low speed clock (flosc) for specific functions, such as T0, LCD and ADC. The flosc has two clock sources selection: internal 16 kHz RC (ILRC) and external 32.768 kHz crystal, which is controlled by LCKS bit.



In external 32.768 kHz crystal mode, LXIN and LXOUT pins switch to crystal mode to drive an off-chip 32.768 kHz crystal after LCKS bit is cleared. Microcontroller supplies WARM bit (the 4th bit of PFLAG) to indicate crystal warm-up status. The WARM bit is set when crystal warm-up procedure ready and cleared when crystal stop. If 32.768 kHz crystal is abnormal stop, re-clear LCKS bit can re-start crystal warm-up procedure.

8.4 Noise Filter

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

Copyright © 2019, SONiX Technology Co., Ltd.

8.5 Power Management

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

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

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

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

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

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

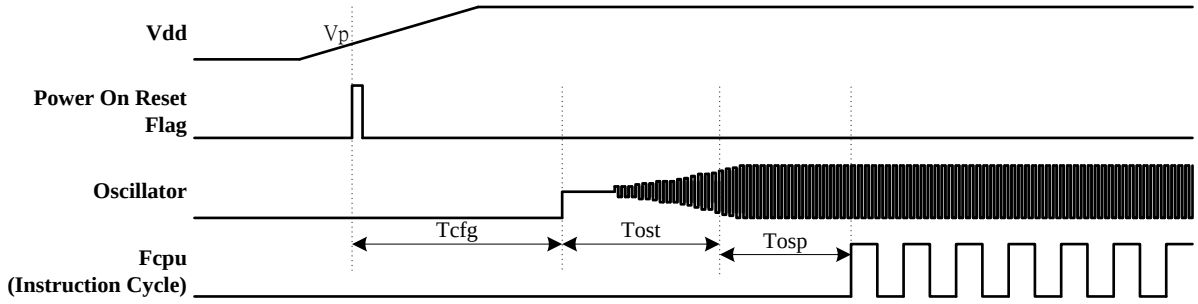
SN8F5829 build in STWK bit (the 0^{th} bit in SYSMOD register) to enable or disable flosc clock in STOP mode. If STWK=0, both fosc and flosc are suspended in STOP mode. If STWK=1, flosc clock keeps running in STOP mode.

8.6 System Clock Timing

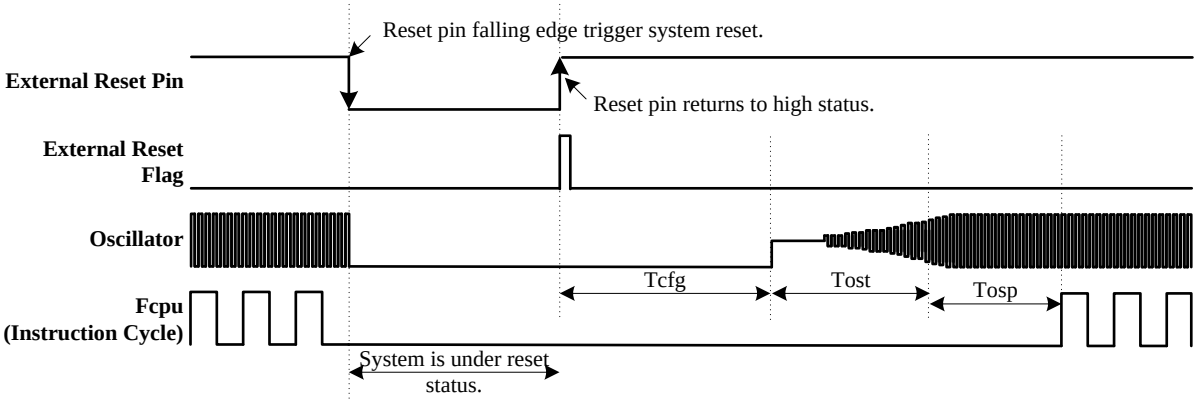
Parameter	Symbol	Description	Typical
Hardware configuration time	Tcfg	8* F_{ILRC} +2^{17}*F_{IHRC}	4.6ms @ F_{ILRC} = 16kHz & F_{IHRC} = 32MHz
Oscillator start up time	Tost	The start-up time is depended on oscillator's material, factory and architecture. Normally, the low-speed oscillator's start-up time is lower than high-speed oscillator. The RC type oscillator's start-up time is faster than crystal type oscillator.	-
Oscillator warm-up time	Tosp	Oscillator warm-up time of reset condition. 2048*F_{hosc} + 5*F_{ILRC} (Power on reset, LVD reset, watchdog reset, external reset pin active.)	825us @ F_{hosc} = 4MHz 441us @ F_{hosc} = 16MHz 377us @ F_{hosc} = 32MHz
		Oscillator warm-up time of power down mode wake-up condition. 2048*F_{hosc} + 5*F_{ILRC} ……Crystal/resonator type oscillator, e.g. 32768Hz crystal, 4MHz crystal, 16MHz crystal… 64*F_{hosc} + 5*F_{ILRC}……RC type oscillator, e.g. internal high-speed RC type oscillator.	X'tal: 825us @ F_{hosc} = 4MHz 441us @ F_{hosc} = 16MHz RC: 315us @ F_{hosc} = 32MHz

Copyright © 2019, SONiX Technology Co., Ltd.

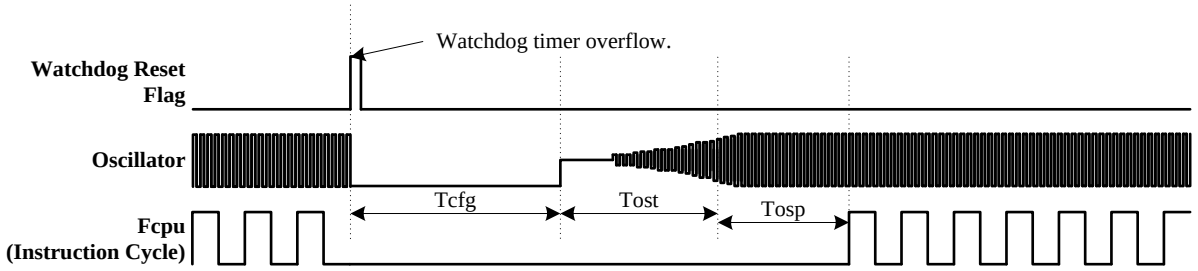
- Power On Reset Timing



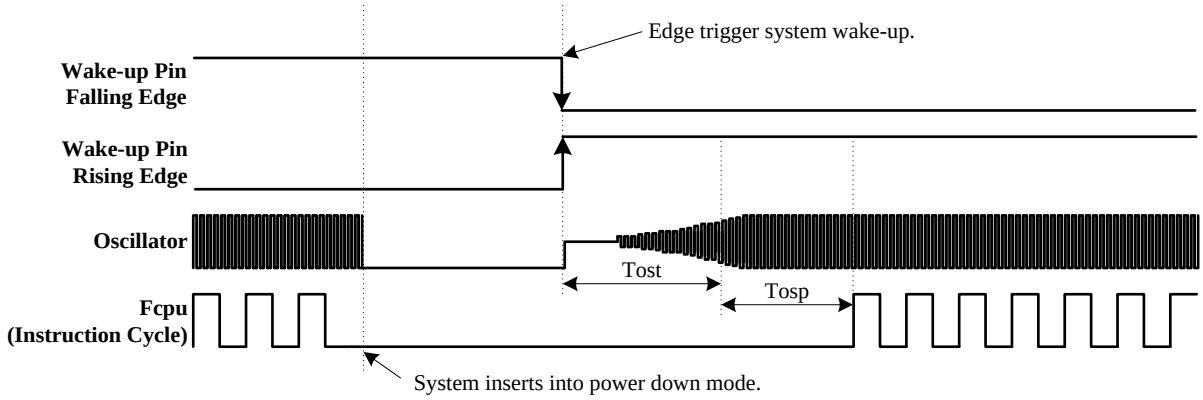
- External Reset Pin Reset Timing



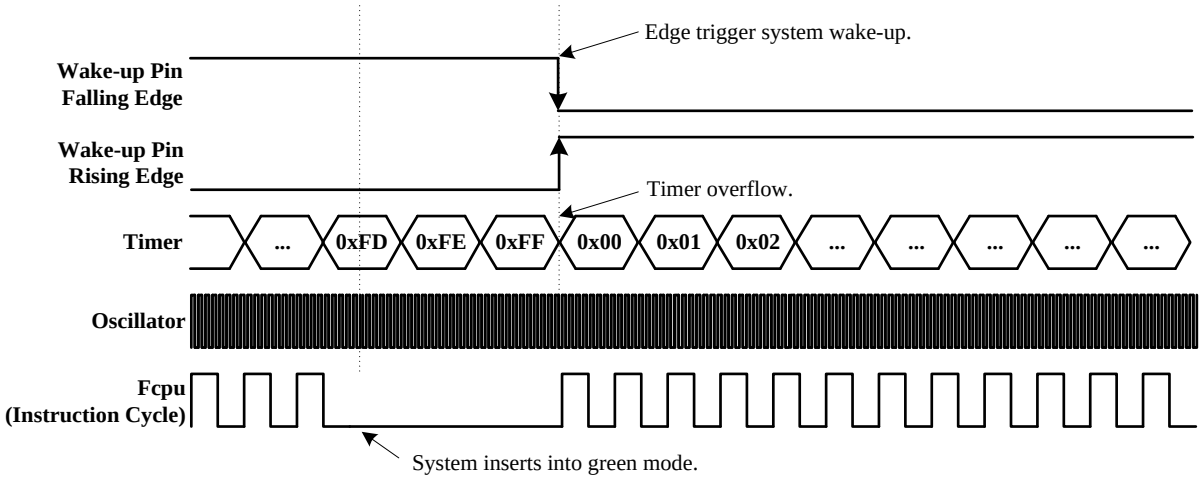
- Watchdog Reset Timing



- STOP Mode Wake-up Timing

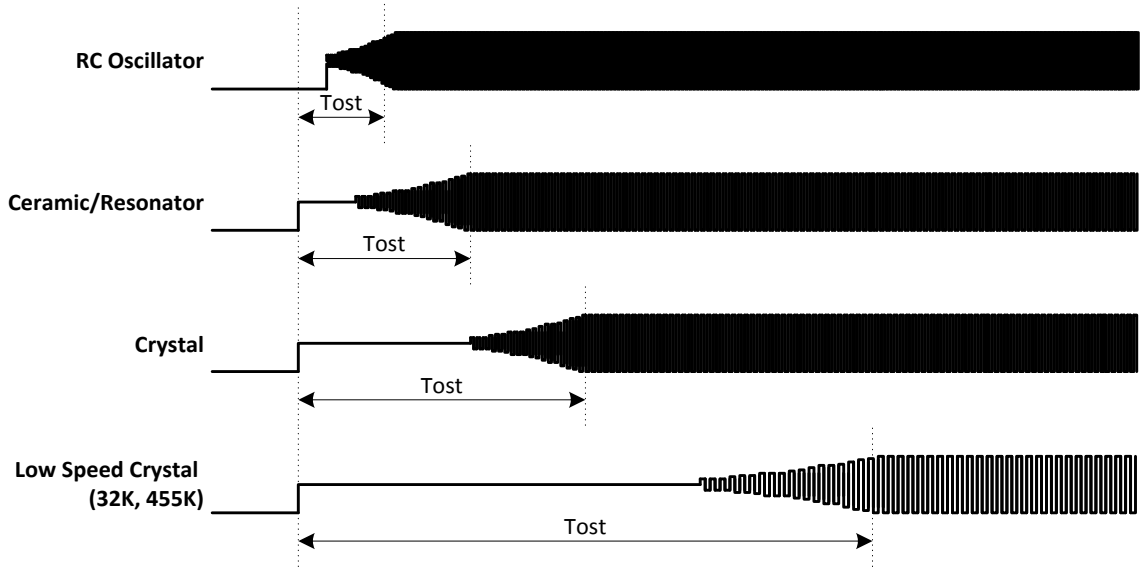


- IDLE Mode Wake-up Timing



- Oscillator Start-up Time

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



8.7 System Clock and Power Management Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CKCON	-	PWSC2	PWSC1	PWSC0	-	-	-	-
CLKSEL	LCKS	-	-	-	-	CLKSEL2	CLKSEL1	CLKSEL0
CLKCMD	CMD7	CMD6	CMD5	CMD4	CMD3	CMD2	CMD1	CMD0
PCON	SMOD2	SMOD1	SMOD0	-	P2SEL	GF0	STOP	IDLE
P1W	P17W	P16W	P15W	P14W	P13W	P12W	P11W	P10W
SYSMOD	-	-	-	-	-	-	-	STWK

Copyright © 2019, SONiX Technology Co., Ltd.

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	

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

CLKCMD Register (0xE6)

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

CLKSEL Register (0xE5)

Bit	Field	Type	Initial	Description
7	LCKS	R/W	1	Low clock source (flosc) select bit 0: flosc = External 32.768KHz crystal and LXIN/LXOUT pins drive 32.768kHz crystal. 1: flosc = Internal 16KHz RC (ILRC).
6..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

Copyright © 2019, SONiX Technology Co., Ltd.

	110: fcpu = fosc / 2 Others: Reserved.

PCON Register (0x87)

Bit	Field	Type	Initial	Description
7..5				Refer to other chapter(s)
4	Reserved	R	0	
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 (0xC6)

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

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

Copyright © 2019, SONiX Technology Co., Ltd.

9 System Operating Mode

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

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

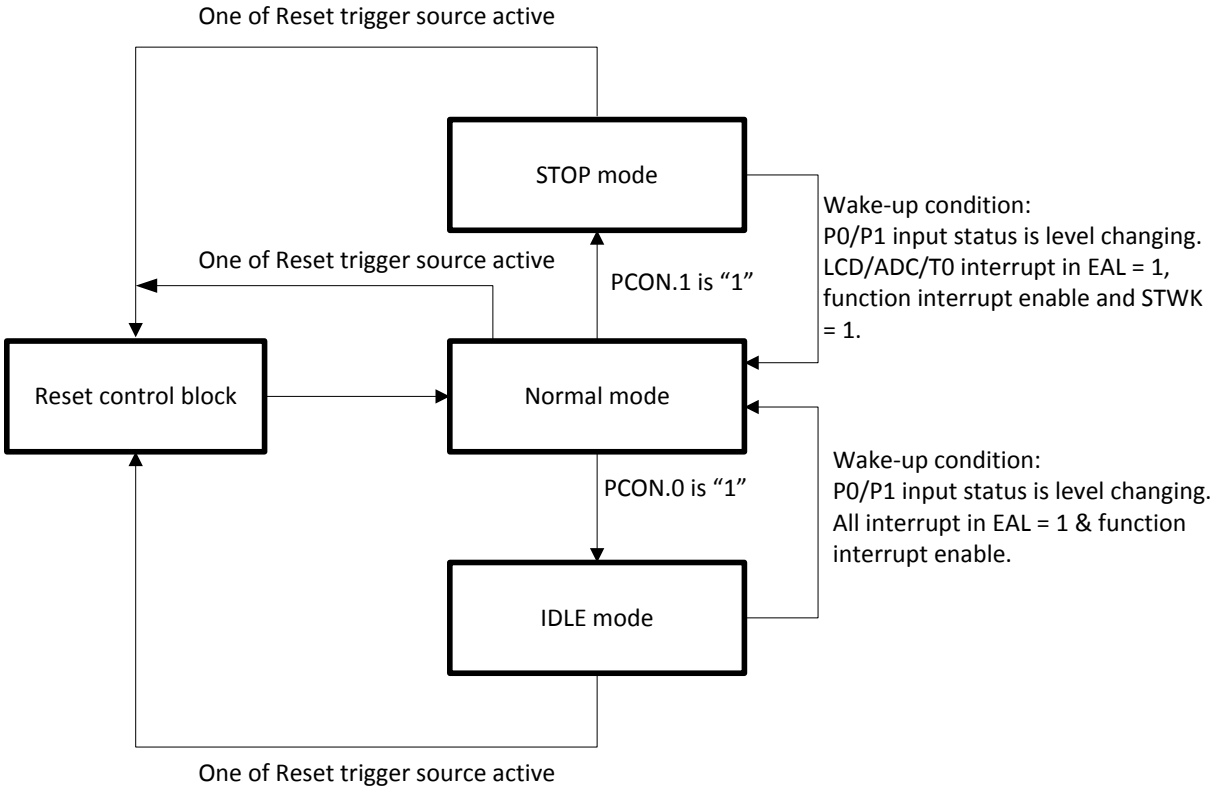


Table 9-1 The operating mode clock control

Operating Mode	Normal Mode	IDLE Mode	STOP Mode
IHRC	IHRC: Running Ext. OSC: Disable	IHRC: Running Ext. OSC: Disable	Stop
ILRC	Running	Running	STWK=1 & LCKS=1: Running Watchdog always: Running
Ext. OSC	IHRC: Disable Ext. OSC : Running	IHRC: Disable Ext. OSC : Running	Stop
Ext. low OSC	Active as LCKS = 0	Active as LCKS = 0	STWK=1 & LCKS=0: Running

Copyright © 2019, SONiX Technology Co., Ltd.

CPU instruction	Executing	Stop	Stop
Timer 0 (Timer, Event counter)	Active by TR0	Active by TR0	Active as Timer 0 clock source is Flosc (ILRC or Ext. 32kHz) & STWK=1
Timer 1 (Timer, Event counter)	Active by TR1	Active by TR1	Inactive
Timer 2 (Timer, capture)	Active as enable	Active as enable	Inactive
Timer 3/4	Active as enable	Active as enable	Inactive
PWM1/2/3/4	Active as enable	Active as enable	Inactive
UART0/1/2	Active as enable	Active as enable	Inactive
I2C	Active as enable	Active as enable	Inactive
LCD	Active as enable	Active as enable	Active as STWK = 1
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	LCD/ADC/T0 interrupt is active when STWK = 1 and clock source is Flosc. Other inactive.
External interrupt	All active	All active	All inactive
Wakeup source	-	P0, P1, Reset, All interrupt in EAL = 1 & function interrupt enable	P0, P1, Reset, LCD/ADC/T0 enable & clock source is Flosc (ILRC or Ext. 32kHz) & STWK=1 & function interrupt enable.

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

Copyright © 2019, SONiX Technology Co., Ltd.

9.1 Normal Mode

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

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

9.2 STOP Mode

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

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

Copyright © 2019, SONiX Technology Co., Ltd.

9.3 IDLE Mode

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

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

Copyright © 2019, SONiX Technology Co., Ltd.

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 + 5 internal low-speed oscillator clocks and 64 internal high-speed oscillator clocks + 5 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 * 5 + 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 * 5 = 0.825 ms (Fosc = 4MHz)

The total wakeup time = 0.825 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 * 5 + 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 * 5= 315 us (Fhosc = 32MHz)

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

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

P1W Register (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 23 interrupt sources (5 external and 18 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
INT3	EX3	IE3	Automatically	2 / 0x0033
INT1	EX1	IE1	Automatically	3 / 0x0083
INT4	EX4	IE4	Automatically	4 / 0x00C3
INT2	EX2	IE2	Automatically	5 / 0x0043
UART1 TX	EU1TX	TI1	By firmware	6 / 0x000B
UART2 TX	EU2TX	TI2	By firmware	7 / 0x003B
UART2 RX	EU2RX	RI2	By firmware	8 / 0x008B
UART1 RX	EU1RX	RI1	By firmware	9 / 0x004B
UART0 TX	EU0TX	TI0	By firmware	10 / 0x0013
I2C	EI2C	SI	By firmware	11 / 0x0093
UART0 RX	EU0RX	RI0	By firmware	12 / 0x0053
ADC	EADC	ADCF	Automatically	13 / 0x001B
LCD	ELCD	LCDF	Automatically	14 / 0x005B
PWM1	EPW1	PW1F	Automatically	15 / 0x0023
PWM3	EPW3	PW3F	Automatically	16 / 0x00B3
PWM4	EPW4	PW4F	Automatically	17 / 0x00A3
PWM2	EPW2	PW2F	Automatically	18 / 0x0063
Timer 0	ET0	TF0	Automatically	19 / 0x002B
Timer 4	ET4	TF4	Automatically	20 / 0x00BB
Timer 1	ET1	TF1	Automatically	21 / 0x00AB
Timer 2	ET2	TF2	Automatically	22 / 0x00EB
Timer 3	ET3	TF3	Automatically	23 / 0x006B

Copyright © 2019, SONiX Technology Co., Ltd.

10.1 Interrupt Operation

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

10.2 Interrupt Priority

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

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

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

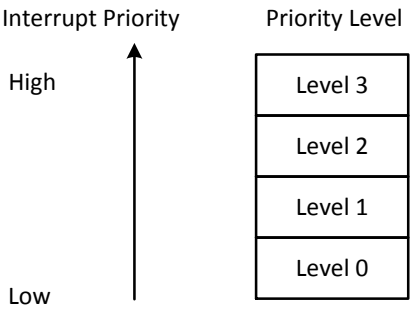
The ISR with the higher priority level can be serviced first; even can break the on-going ISR with the lower priority level. The ISR with the lower priority level will be pending until the ISR with the higher priority level completes.

Group	Interrupt Source				
Group 0	INT0	INT3	INT1	INT4	INT2
Group 1	UART1 TX	UART2 TX	UART2 RX	-	UART1 RX
Group 2	UART0 TX	-	I2C	-	UART0 RX
Group 3	ADC	-	-	-	LCD
Group 4	PWM1	PWM3	PWM4	-	PWM2
Group 5	Timer 0	Timer 4	Timer 1	Timer 2	Timer 3

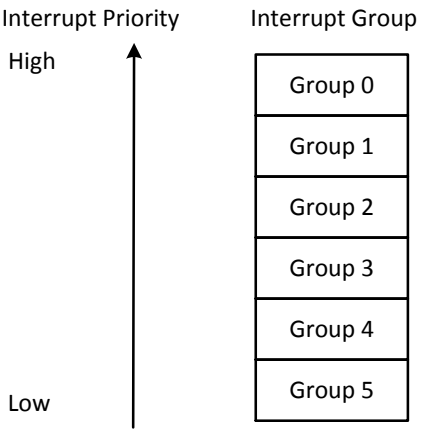
Copyright © 2019, SONiX Technology Co., Ltd.

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

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



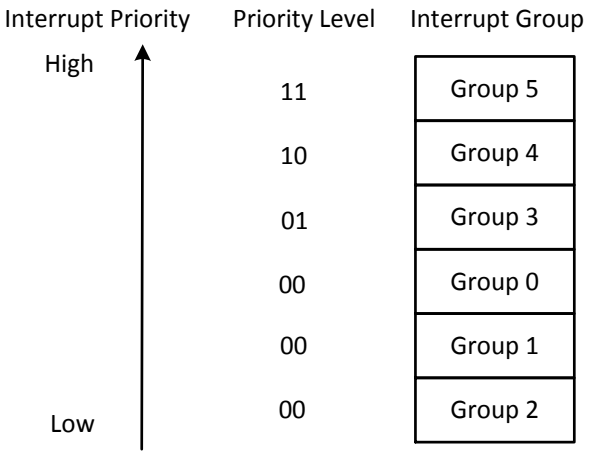
Higher priority level has higher priority



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

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

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



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

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

IP0, IP1 Registers

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

IP0 Register (0XA9)

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

IP1 Register (0XB9)

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

10.3 Interrupt Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IEN0	EAL	EX4	EX3	EX2	ET1	EX1	ET0	EX0
IEN1	EU2RX	EU2TX	EU1RX	EU1TX	EU0RX	EU0TX	-	EI2C
IEN2	ET4	ELCD	ET3	ET2	EPW2	EPW1	-	EADC
IRCON	PW4F	PW3F	EPW4	EPW3	-	IE4	IE3	IE2
IRCON2	TF4	LCDF	TF3	TF2	PW2F	PW1F	-	ADCF
TCON	TF1	TR1	TF0	TR0	IE1	-	IE0	-
S0CON	S0M0	S0M1	S0M20	REN0	TB80	RB80	TI0	RI0
S1CON	S1M0	S1M1	S1M20	REN1	TB81	RB81	TI1	RI1
S2CON	S2M0	S2M1	S2M20	REN2	TB82	RB82	TI2	RI2
I2CCON	CR2	ENS1	STA	STO	SI	AA	CR1	CR0

Copyright © 2019, SONiX Technology Co., Ltd.

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	EX4	R/W	0	External P2.4 interrupt (INT4) control bit. 0: Disable INT4 interrupt function. 1: Enable INT4 interrupt function.
5	EX3	R/W	0	External P2.3 interrupt (INT3) control bit. 0: Disable INT3 interrupt function. 1: Enable INT3 interrupt function.
4	EX2	R/W	0	External P2.2 interrupt (INT2) control bit. 0: Disable INT2 interrupt function. 1: Enable INT2 interrupt function.
3	ET1	R/W	0	T1 timer interrupt control bit. 0: Disable T1 interrupt function. 1: Enable T1 interrupt function.
2	EX1	R/W	0	External P2.1 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 P2.0 interrupt (INT0) control bit. 0: Disable INT0 interrupt function. 1: Enable INT0 interrupt function.

IEN1 Register (0XB8)

Bit	Field	Type	Initial	Description
7	EU2RX	R/W	0	UART2 RX interrupt control bit. 0: Disable UART RX interrupt function. 1: Enable UART RX interrupt function.
6	EU2TX	R/W	0	UART2 TX interrupt control bit. 0: Disable UART TX interrupt function. 1: Enable UART TX interrupt function.
5	EU1RX	R/W	0	UART1 RX interrupt control bit.

Copyright © 2019, SONiX Technology Co., Ltd.

				0: Disable UART RX interrupt function. 1: Enable UART RX interrupt function.
4	EU1TX	R/W	0	UART1 TX interrupt control bit. 0: Disable UART TX interrupt function. 1: Enable UART TX interrupt function.
3	EU0RX	R/W	0	UART0 RX interrupt control bit. 0: Disable UART RX interrupt function. 1: Enable UART RX interrupt function.
2	EU0TX	R/W	0	UART0 TX interrupt control bit. 0: Disable UART TX interrupt function. 1: Enable UART TX interrupt function.
1	Reserved	R	0	
0	EI2C	R/W	0	I2C interrupt control bit. 0: Disable I2C interrupt function. 1: Enable I2C interrupt function.

IEN2 Register (0X9A)

Bit	Field	Type	Initial	Description
7	ET4	R/W	0	Timer 4 interrupt control bit. 0: Disable T4 interrupt function. 1: Enable T4 interrupt function.
6	ELCD	R/W	0	LCD interrupt control bit. 0: Disable LCD interrupt function. 1: Enable LCD interrupt function.
5	ET3	R/W	0	Timer 3 interrupt control bit. 0: Disable T3 interrupt function. 1: Enable T3 interrupt function.
4	ET2	R/W	0	Timer 2 interrupt control bit. 0: Disable T2 interrupt function. 1: Enable T2 interrupt function.
3	EPW2	R/W	0	PWM2 interrupt control bit. 0 = Disable PWM2 interrupt function. 1 = Enable PWM2 interrupt function.
2	EPW1	R/W	0	PWM1 interrupt control bit. 0 = Disable PWM1 interrupt function. 1 = Enable PWM1 interrupt function.
1	Reserved	R/W	0	
0	EADC	R/W	0	ADC interrupt control bit.

Copyright © 2019, SONiX Technology Co., Ltd.

				0: Disable ADC interrupt function. 1: Enable ADC interrupt function.

IRCON Register (0XC0)

Bit	Field	Type	Initial	Description
7	PW4F	R/W	0	PWM4 interrupt request flag. 0: None PWM4 interrupt request 1: PWM4 interrupt request.
6	PW3F	R/W	0	PWM3 interrupt request flag. 0: None PWM3 interrupt request 1: PWM3 interrupt request.
5	EPW4	R/W	0	PWM4 interrupt control bit. 0 = Disable PWM4 interrupt function. 1 = Enable PWM4 interrupt function.
4	EPW3	R/W	0	PWM3 interrupt control bit. 0 = Disable PWM3 interrupt function. 1 = Enable PWM3 interrupt function.
3	Reserved	R	0	
2	IE4	R/W	0	External P2.4 interrupt (INT4) request flag 0: None INT4 interrupt request. 1: INT4 interrupt request.
1	IE3	R/W	0	External P2.3 interrupt (INT3) request flag 0: None INT3 interrupt request. 1: INT3 interrupt request.
0	IE2	R/W	0	External P2.2 interrupt (INT2) request flag 0: None INT2 interrupt request. 1: INT2 interrupt request.

IRCON2 Register (0XBF)

Bit	Field	Type	Initial	Description
7	TF4	R/W	0	Timer 4 interrupt request flag. 0: None T4 interrupt request 1: T4 interrupt request.
6	LCDF	R/W	0	LCD interrupt request flag. 0: None LCD interrupt request 1: LCD interrupt request.
5	TF3	R/W	0	Timer 3 interrupt request flag.

Copyright © 2019, SONiX Technology Co., Ltd.

				0: None T3 interrupt request 1: T3 interrupt request.
4	TF2	R/W	0	Timer 2 interrupt request flag. 0: None T2 interrupt request 1: T2 interrupt request.
3	PW2F	R/W	0	PWM2 interrupt request flag. 0: None PWM2 interrupt request 1: PWM2 interrupt request.
2	PW1F	R/W	0	PWM1 interrupt request flag. 0: None PWM1 interrupt request 1: PWM1 interrupt request.
1	Reserved	R/W	0	
0	ADCF	R/W	0	ADC interrupt request flag. 0: None ADC interrupt request. 1: ADC interrupt request.

TCON Register (0X88)

Bit	Field	Type	Initial	Description
7	TF1	R/W	0	Timer 1 interrupt request flag. 0: None T1 interrupt request 1: T1 interrupt request.
5	TF0	R/W	0	Timer 0 interrupt request flag. 0: None T0 interrupt request 1: T0 interrupt request.
3	IE1	R/W	0	External P2.1 interrupt (INT1) request flag 0: None INT1 interrupt request. 1: INT1 interrupt request.
1	IE0	R/W	0	External P2.0 interrupt (INT0) request flag 0: None INT1 interrupt request. 1: INT1 interrupt request.
Else				Refer to other chapter(s)

Copyright © 2019, SONiX Technology Co., Ltd.

S0CON Register (0X98)

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

S1CON Register (0XC1)

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

Copyright © 2019, SONiX Technology Co., Ltd.

S2CON Register (0XC9)

Bit	Field	Type	Initial	Description
1	TI2	R/W	0	UART2 transmit interrupt request flag. It indicates completion of a serial transmission at UART2. 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 UART2 transmit interrupt request. 1: UART2 transmit interrupt request.
0	RI2	R/W	0	UART2 receive interrupt request flag. It is set by hardware after completion of a serial reception at UART2. 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 UART2 receive interrupt request. 1: UART2 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)

Copyright © 2019, SONiX Technology Co., Ltd.

10.4 Example

Defining Interrupt Vector. The interrupt service routine is following user assembly code program.

```
            ORG     0               ; 0000H
            JMP     START           ; Jump to user program address.
            …
            ORG     0X0033          ; Jump to interrupt service routine address.
            JMP     ISR_INT3
            ORG     0X0043
            JMP     ISR_INT2
            …
            ORG     0X00AB
            JMP     ISR_T1
            …
            ORG     0X00EC
START:                              ; 00ECH, The head of user program.
            …                       ; User program.
            …
            JMP     START           ; End of user program.
            …
ISR_ INT3:                          ; 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_T1:                             ;
            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.
```

Copyright © 2019, SONiX Technology Co., Ltd.

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

Copyright © 2019, SONiX Technology Co., Ltd.

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.

```
1 #define __XRAM_SFR_H__ // For XRAM Register Declaration.
2 #include <SN8F5829.H>
3 #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

Copyright © 2019, SONiX Technology Co., Ltd.

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.

Copyright © 2019, SONiX Technology Co., Ltd.

11.8 Sample Code

The following sample code demonstrates how to perform MDU 32 bit / 16 bit.

```
#define __XRAM_SFR_H__ // For XRAM Register Declaration.
#include <SN8F5829.H>
#pragma mdu_r515  //Keil C51 MDU command line

void main(void)
{
  unsigned int Divisor;  // 16-bit divisor
  unsigned long Dividend;  // 32-bit dividend
  unsigned long Quotient;  // 32-bit Quotient
  unsigned int  Remainder;  // 16-bit Remainder

  Divisor = 0x1234;
  Dividend = 0x56789ABC;
  Quotient = Dividend / Divisor;  //0x0004C016
  Remainder = Dividend % Divisor;  //0x0A44

  while(1);
}
```

Copyright © 2019, SONiX Technology Co., Ltd.

12 GPIO

The microcontroller has up to 60 bidirectional general purpose I/O pin (GPIO). Unlike the original 8051 only has open-drain output, SN8F5829 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 P7M registers. These bits specify each pin that is either input mode or output mode.

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0M	P07M	P06M	P05M	P04M	P03M	P02M	P01M	P00M
P1M	P17M	P16M	P15M	P14M	P13M	P12M	P11M	P10M
P2M	-	-	-	P24M	P23M	P22M	P21M	P20M
P3M	P37M	P36M	P35M	P34M	P33M	P32M	P31M	P30M
P4M	P47M	P46M	P45M	P44M	P43M	P42M	P41M	P40M
P5M	P57M	P56M	P55M	P54M	P53M	P52M	P51M	P50M
P6M	P67M	P66M	P65M	P64M	P63M	P62M	P61M	P60M
P7M	P77M	P76M	P75M	P74M	P73M	P72M	P71M	P70M
P0OC	-	-	P14OC	P13OC	P06OC	P05OC	P03OC	P02OC

P0M: 0xF9, P1M: 0xFA, P2M: 0xFB, P3M: 0xFC, P4M: 0xFD, P5M: 0xFE, P6M: 0xE3, P7M: 0xE4

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..0				et cetera

Copyright © 2019, SONiX Technology Co., Ltd.

P0OC Register (0xD1)

Bit	Field	Type	Initial	Description
Else	Reserved	R	0	
5	P14OC	R/W	0	P1.4 open-drain control bit 0: Disable 1: Enable
4	P13OC	R/W	0	P1.3 open-drain control bit 0: Disable 1: Enable
3	P06OC	R/W	0	P0.6 open-drain control bit 0: Disable 1: Enable
2	P05OC	R/W	0	P0.5 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 P7, 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 P7 register value would be latched immediately, yet the value would be outputted until the mapped P0M – P7M is set to output mode. If the pin is currently in output mode, any value set to P0 to P7 register would be presented on the pin immediately.

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0	P07	P06	P05	P04	P03	P02	P01	P00
P1	P17	P16	P15	P14	P13	P12	P11	P10
P2	-	-	-	P24	P23	P22	P21	P20
P3	P37	P36	P35	P34	P33	P32	P31	P30
P4	P47	P46	P45	P44	P43	P42	P41	P40

Copyright © 2019, SONiX Technology Co., Ltd.

P5	P57	P56	P55	P54	P53	P52	P51	P50
P6	P67	P66	P65	P64	P63	P62	P61	P60
P7	P77	P76	P75	P74	P73	P72	P71	P70

P0: 0x80, P1: 0x90, P2: 0xA0, P3: 0xB0, P4: 0xC8, P5: 0xD8, P6: 0xE8, P7: 0xF8

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..0				et cetera

12.3 On-chip Pull-up Resisters

The P0UR to P7UR registers are mapped to each pins' internal 100k ohm (in typical value) pull-up resister.

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0UR	P07UR	P06UR	P05UR	P04UR	P03UR	P02UR	P01UR	P00UR
P1UR	P17UR	P16UR	P15UR	P14UR	P13UR	P12UR	P11UR	P10UR
P2UR	-	-	-	P24UR	P23UR	P22UR	P21UR	P20UR
P3UR	P37UR	P36UR	P35UR	P34UR	P33UR	P32UR	P31UR	P30UR
P4UR	P47UR	P46UR	P45UR	P44UR	P43UR	P42UR	P41UR	P40UR
P5UR	P57UR	P56UR	P55UR	P54UR	P53UR	P52UR	P51UR	P50UR
P6UR	P67UR	P66UR	P65UR	P64UR	P63UR	P62UR	P61UR	P60UR
P7UR	P77UR	P76UR	P75UR	P74UR	P73UR	P72UR	P71UR	P70UR

P0UR: 0xF028, P1UR: 0xF029, P2UR: 0xF02A, P3UR: 0xF02B,
P4UR: 0xF02C, P5UR: 0xF02D, P6UR: 0xF02E, P7UR: 0xF02F

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

Copyright © 2019, SONiX Technology Co., Ltd.

5	P05UR	R/W	0	On-chip pull-up resister control of P0.5 0: Disable* 1: Enable
4..0				et cetera

* 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
P3CON	P3CON7	P3CON6	P3CON5	P3CON4	P3CON3	P3CON2	P3CON1	P3CON0
P4CON	P4CON7	P4CON6	P4CON5	P4CON4	P4CON3	P4CON2	P4CON1	P4CON0
P5CON	P5CON7	P5CON6	P5CON5	P5CON4	P5CON3	P5CON2	P5CON1	P5CON0
P6CON	P6CON7	P6CON6	P6CON5	P6CON4	P6CON3	P6CON2	P6CON1	P6CON0
P7CON	P7CON7	P7CON6	P7CON5	P7CON4	P7CON3	P7CON2	P7CON1	P7CON0

P3CON: 0x9E, P4CON: 0x9F, P5CON: 0xCF, P6CON: 0xD6, P7CON: 0xD7

Bit	Field	Type	Initial	Description
7	P3CON7	R/W	0	P3.7 configuration control bit*. 0: P3.7 can be analog input pin or digital GPIO pin. 1: P3.7 is pure analog input pin and can't be a digital GPIO pin.
6	P3CON6	R/W	0	P3.6 configuration control bit*. 0: P3.6 can be analog input pin or digital GPIO pin. 1: P3.6 is pure analog input pin and can't be a digital GPIO pin.
5	P3CON5	R/W	0	P3.5 configuration control bit*. 0: P3.5 can be analog input pin or digital GPIO pin. 1: P3.5 is pure analog input pin and can't be a digital GPIO pin.
4..0				et cetera

* P3CON [7:0] will configure related Port3 pin as pure analog input pin to avoid current leakage.

Copyright © 2019, SONiX Technology Co., Ltd.

13 External Interrupt

INT0, INT1, INT2, INT3 and INT4 are external interrupt trigger sources. Build in edge trigger configuration function and edge direction is selected by PEDGE and PEDGE1 register. When both external interrupt (EX0/EX1/EX2/EX3/EX4) and global interrupt (EAL) are enabled, the external interrupt request flag (IE0/IE1/IE2/IE3/IE4) will be set to “1” as edge trigger event occurs. The program counter will jump to the interrupt vector (ORG 0x0003/0x0083/0x0043/0x0033/0x00C3) 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	EX3G1	EX3G0	EX2G1	EX2G0	EX1G1	EX1G0	EX0G1	EX0G0
PEDGE1	-	-	-	-	-	-	EX4G1	EX4G0
IEN0	EAL	EX4	EX3	EX2	ET1	EX1	ET0	EX0
TCON	TF1	TR1	TF0	TR0	IE1	-	IE0	-
IRCON	PW4F	PW3F	EPW4	EPW3	-	IE4	IE3	IE2

PEDGE Register (0X8F)

Bit	Field	Type	Initial	Description
7..6	EX3G[1:0]	R/W	10	External interrupt 3 trigger edge control register. 00: Reserved. 01: Rising edge trigger. 10: Falling edge trigger (default) 11: Both rising and falling edge trigger
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

Copyright © 2019, SONiX Technology Co., Ltd.

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

PEDGE1 Register (0xC7)

Bit	Field	Type	Initial	Description
Else	Reserved	R	0	
1..0	EX4G[1:0]	R/W	10	External interrupt 4 trigger edge control register. 00: Reserved. 01: Rising edge trigger. 10: Falling edge trigger (default) 11: Both rising and falling edge trigger

IEN0 Register (0XA8)

Bit	Field	Type	Initial	Description
7	EAL	R/W	0	Enable all interrupt control bit. 0: Disable all interrupt function. 1: Enable all interrupt function.
6	EX4	R/W	0	External P2.4 interrupt (INT4) control bit. 0: Disable INT4 interrupt function. 1: Enable INT4 interrupt function.
5	EX3	R/W	0	External P2.3 interrupt (INT3) control bit. 0: Disable INT3 interrupt function. 1: Enable INT3 interrupt function.
4	EX2	R/W	0	External P2.2 interrupt (INT2) control bit. 0: Disable INT2 interrupt function. 1: Enable INT2 interrupt function.
2	EX1	R/W	0	External P2.1 interrupt (INT1) control bit. 0: Disable INT1 interrupt function. 1: Enable INT1 interrupt function.
0	EX0	R/W	0	External P2.0 interrupt (INT0) control bit. 0: Disable INT0 interrupt function. 1: Enable INT0 interrupt function.

Copyright © 2019, SONiX Technology Co., Ltd.

TCON Register (0X88)

Bit	Field	Type	Initial	Description
3	IE1	R/W	0	External P2.1 interrupt (INT1) request flag 0: None INT1 interrupt request. 1: INT1 interrupt request.
1	IE0	R/W	0	External P2.0 interrupt (INT0) request flag 0: None INT0 interrupt request. 1: INT0 interrupt request.
Else				Refer to other chapter(s)

IRCON Register (0XC0)

Bit	Field	Type	Initial	Description
Else				Refer to other chapter(s)
2	IE4	R/W	0	External P2.4 interrupt (INT4) request flag 0: None INT4 interrupt request. 1: INT4 interrupt request.
1	IE3	R/W	0	External P2.3 interrupt (INT3) request flag 0: None INT3 interrupt request. 1: INT3 interrupt request.
0	IE2	R/W	0	External P2.2 interrupt (INT2) request flag 0: None INT2 interrupt request. 1: INT2 interrupt request.

13.2 Sample Code

The following sample code demonstrates how to perform INT1/INT2/INT3/INT4 with interrupt.

```
#define INT0Rsing     (1 << 0)  //INT1 trigger edge is rising edge
#define INT0Falling   (2 << 0)  //INT1 trigger edge is falling edge
#define INT0LeChge    (3 << 0)  //INT1 trigger edge is level chagne
#define EINT0         (1 << 0)  //INT1 interrupt enable

#define INT1Rsing     (1 << 2)  //INT1 trigger edge is rising edge
#define INT1Falling   (2 << 2)  //INT1 trigger edge is falling edge
#define INT1LeChge    (3 << 2)  //INT1 trigger edge is level chagne
#define EINT1         (1 << 2)  //INT1 interrupt enable

#define INT2Rsing     (1 << 4)  //INT2 trigger edge is rising edge
#define INT2Falling   (2 << 4)  //INT2 trigger edge is falling edge
#define INT2LeChge    (3 << 4)  //INT2 trigger edge is level chagne
#define EINT2         (1 << 4)  //INT2 interrupt enable

#define INT3Rsing     (1 << 6)  //INT3 trigger edge is rising edge
#define INT3Falling   (2 << 6)  //INT3 trigger edge is falling edge
```

Copyright © 2019, SONiX Technology Co., Ltd.

```
#define INT3LeChge     (3 << 6)  //INT3 trigger edge is level chagne
#define EINT3          (1 << 5)  //INT3 interrupt enable

#define INT4Rsing      (1 << 0)  //INT4 trigger edge is rising edge
#define INT4Falling    (2 << 0)  //INT4 trigger edge is falling edge
#define INT4LeChge     (3 << 0)  //INT4 trigger edge is level chagne
#define EINT4          (1 << 6)  //INT4 interrupt enable


void EnableINT(void)
{
  // INT0 falling edge, INT1 falling edge, INT2 level change, INT3 rising edge
  // INT4 rising edge
  PEDGE = INT0Falling | INT1Falling | INT2LeChge | INT3Rising;
  PEDGE1 = INT4Rising;

  // Enable INT0/INT1/INT2/INT3/INT4 interrupt
  IEN0 |= EINT0 | EINT1 | EINT2 | EINT3 | EINT4;
  // Enable total interrupt
  IEN0 |= 0x80;

  P1 = 0x00;
  P1M = 0xFF;
}

void INT0Interrupt(void) interrupt ISRInt0  //0x03
{ //IE0 clear by hardware
  P10 = ~P10;
}

void INT1Interrupt(void) interrupt ISRInt1  //0x83
{ //IE1 clear by hardware
  P11 = ~P11;
}

void INT2Interrupt(void) interrupt ISRInt2  //0x43
{ //IE2 clear by hardware
  P12 = ~P12;
}

void INT3Interrupt(void) interrupt ISRInt3  //0x33
{ //IE3 clear by hardware
  P13 = ~P13;
}

void INT4Interrupt(void) interrupt ISRInt4  //0xC3
{ //IE4 clear by hardware
  P14 = ~P14;
}
```

Copyright © 2019, SONiX Technology Co., Ltd.

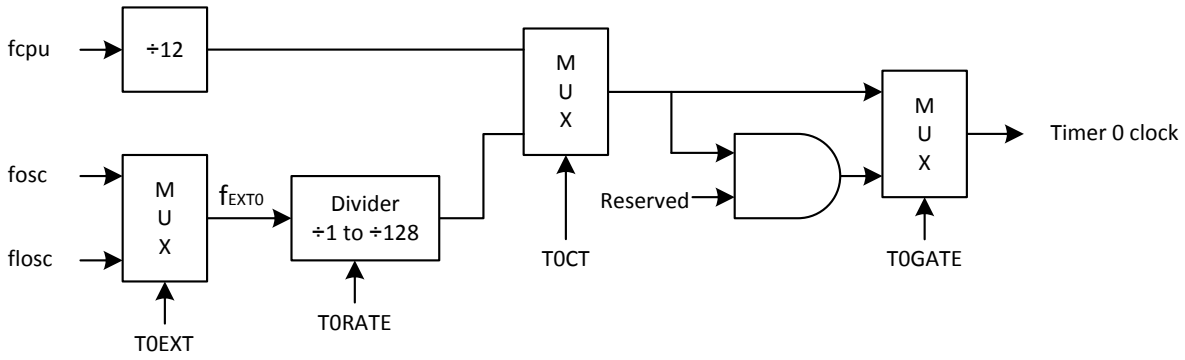
14 Timer 0 and Timer 1

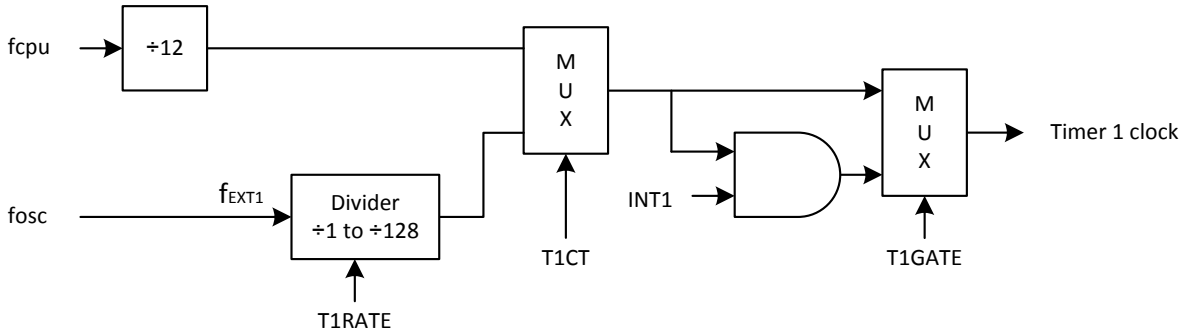
Timer 0 and Timer 1 are two independent binary up timers. Timer 0 has four different operation modes: (1) 13-bit up counting timer, (2) 16-bit up counting timer, (3) 8-bit up counting timer with specified reload value support, and (4) separated two 8-bit up counting timer. By contrast, Timer 1 has only 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

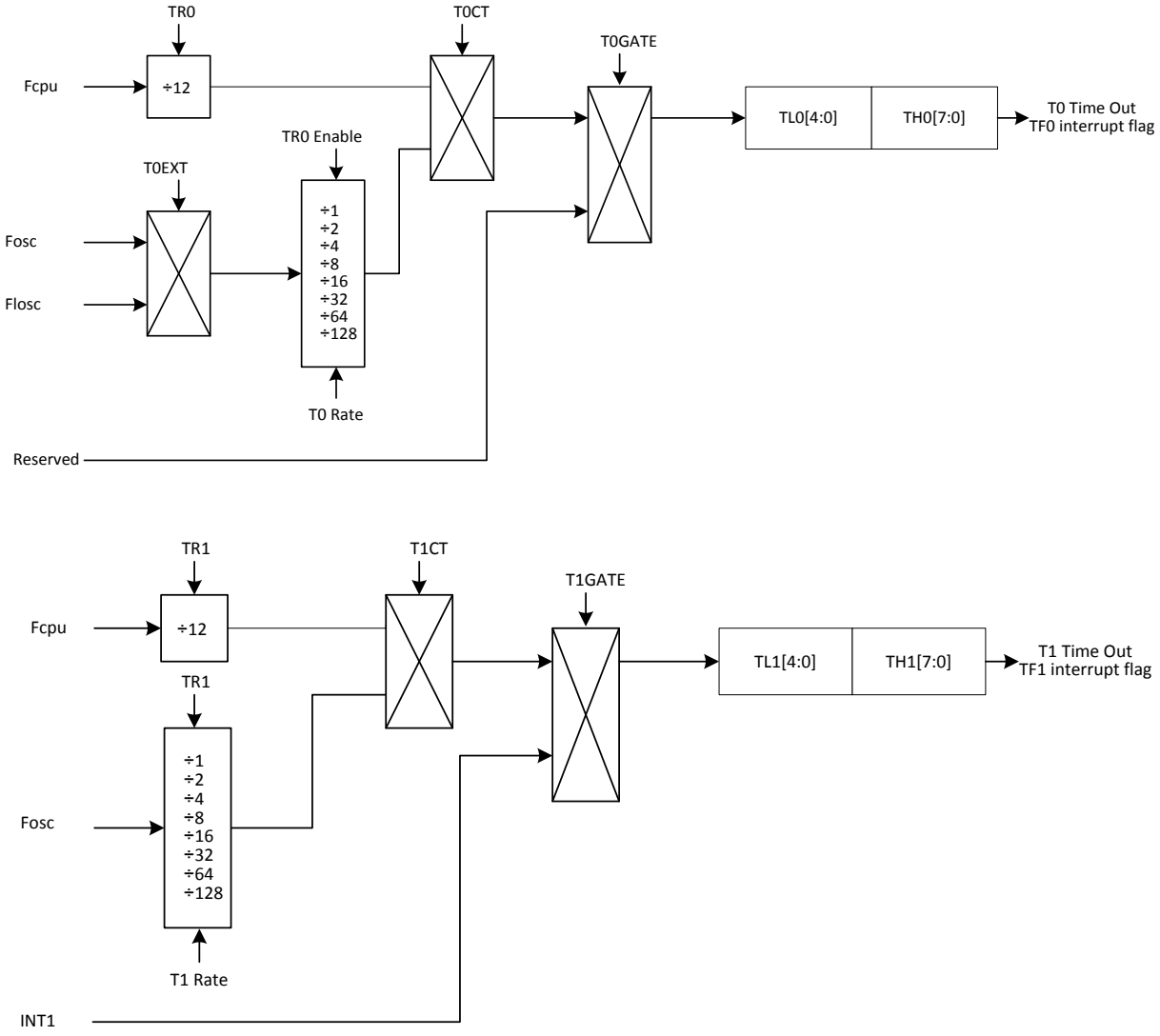
The figures below illustrate the clock selection circuit of Timer 0 and Timer 1. Timer 0 has three clock sources selection: fcpu, fosc, and flosc. All clock sources can be gated (pause) by reserved pin if T0GATE is applied. Timer 1 clock sources selection: fcpu and fosc. All clock sources can be gated (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).





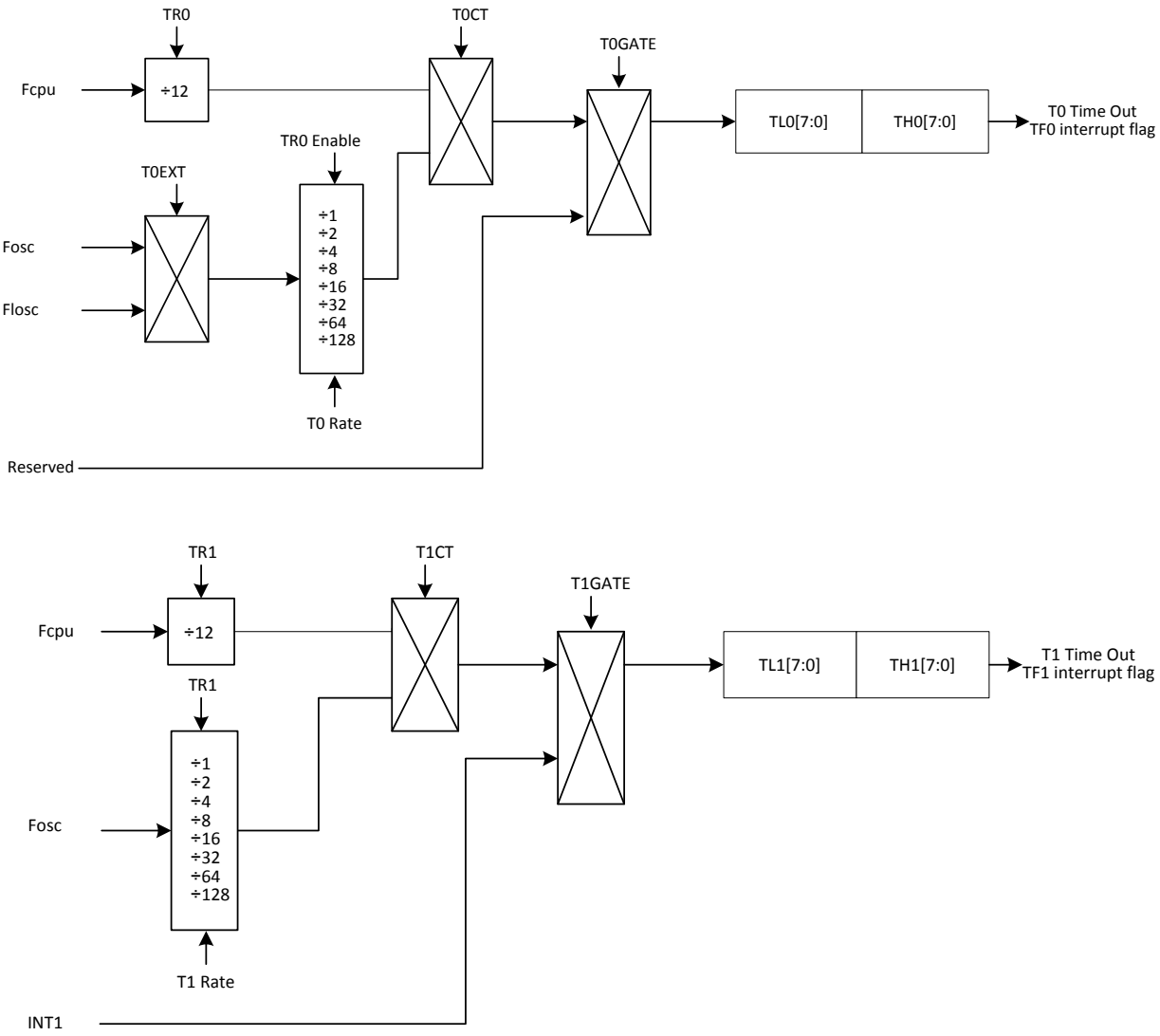
14.2 Mode 0: 13-bit Up Counting Timer

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.



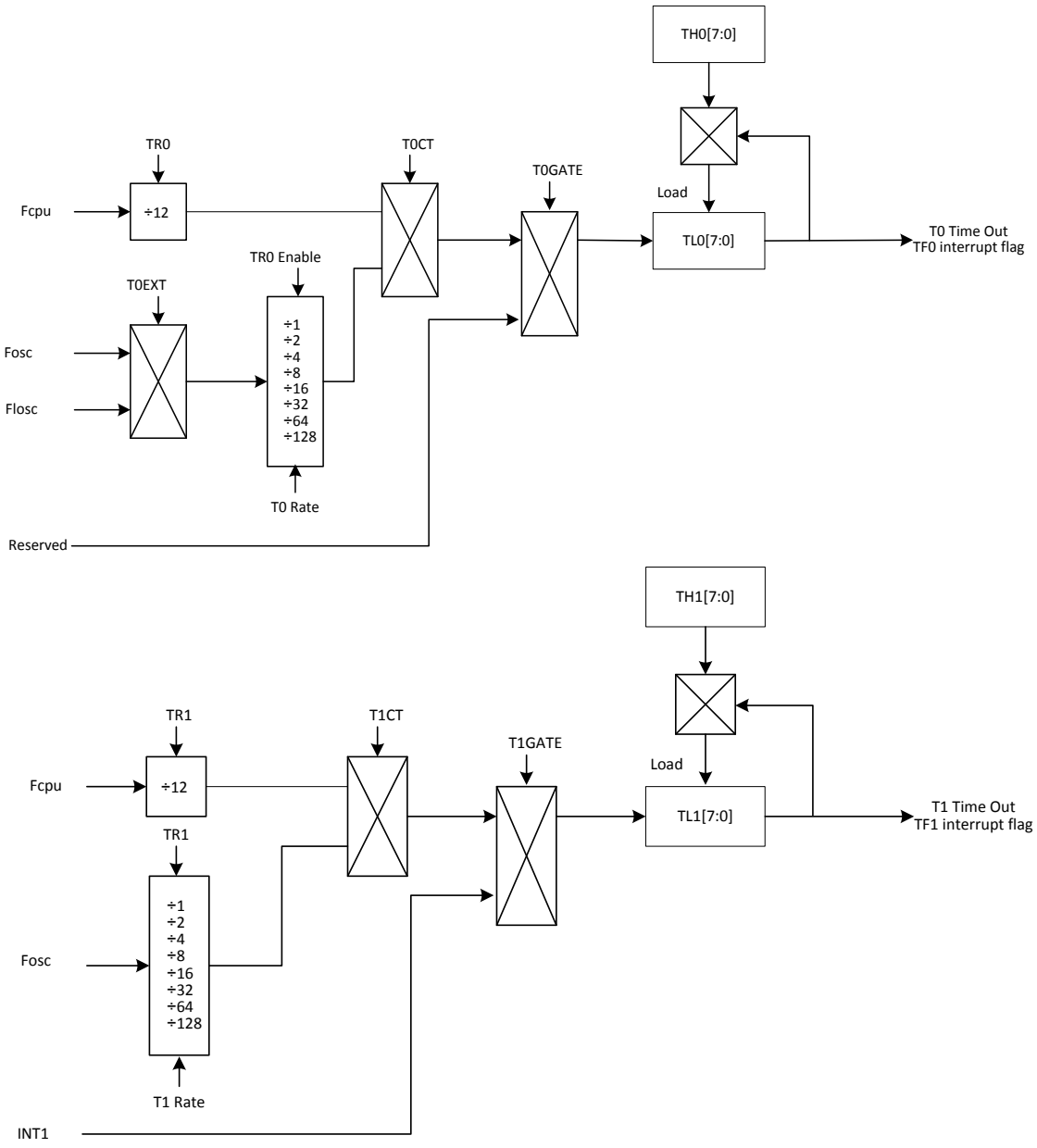
14.3 Mode 1: 16-bit Up Counting Timer

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.

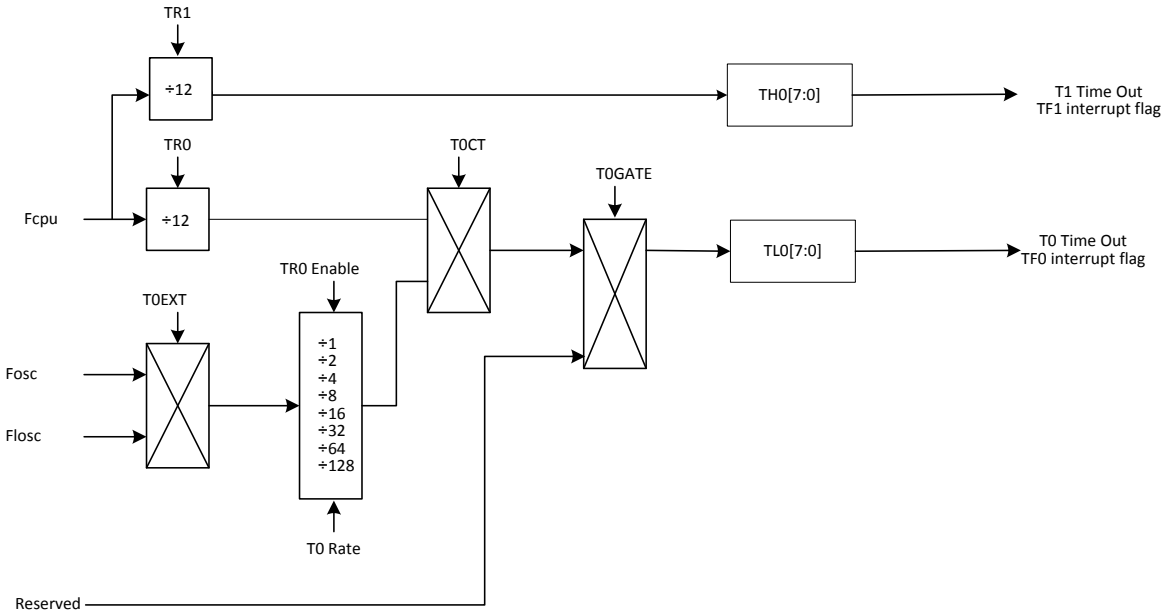


14.5 Mode 3 (Timer 0 only): Separated Two 8-bit Up Counting Timer

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 reserved 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	EX4	EX3	EX2	ET1	EX1	ET0	EX0
TCON	TF1	TR1	TF0	TR0	IE1	-	IE0	-
TCON0	T0EXT	T0RATE2	T0RATE1	T0RATE0	-	T1RATE2	T1RATE1	T1RATE0
TMOD	T1GATE	T1CT	T1M1	T1M0	T0GATE	T0CT	T0M1	T0M0
TH0	TH07	TH06	TH05	TH04	TH03	TH02	TH01	TH00
TL0	TL07	TL06	TL05	TL04	TL03	TL02	TL01	TL00
TH1	TH17	TH16	TH15	TH14	TH13	TH12	TH11	TH10
TL1	TL17	TL16	TL15	TL14	TL13	TL12	TL11	TL10

IEN0 Register (0xA8)

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

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

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

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

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

Copyright © 2019, SONiX Technology Co., Ltd.

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..0	Reserved	R/W	000	

Copyright © 2019, SONiX Technology Co., Ltd.

TCON0 Register (0xE7)

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

*(1) f_{EXT1} = fosc.

Copyright © 2019, SONiX Technology Co., Ltd.

TMOD Register (0x89)

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

*(1) f_{EXT1} = fosc.

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

Copyright © 2019, SONiX Technology Co., Ltd.

14.7 Sample Code

The following sample code demonstrates how to perform T0/T1 with interrupt.

```
#define T0Mode0     (0 << 0)  //T0 mode0, 13-bit counter
#define T0Mode1     (1 << 0)  //T0 mode1, 16-bit counter
#define T0Mode2     (2 << 0)  //T0 mode2, 8-bit auto-reload counter
#define T0Mode3     (3 << 0)  //T0 mode3, T0 two 8-bit counter/T1 no flag
#define T0ClkFcpu    (0 << 0)  //T0 clock source from Fcpu/12
#define T0ClkExt     (4 << 0)  //T0 clock source from Fosc or FRTC
#define T0ExtFosc    (0 << 4)  //T0 clock source from Fosc
#define T0ExtFlosc    (8 << 4)  //T0 clock source from Flosc

#define T1Mode0     (0 << 4)  //T1 mode0, 13-bit counter
#define T1Mode1     (1 << 4)  //T1 mode1, 16-bit counter
#define T1Mode2     (2 << 4)  //T1 mode2, 8-bit auto-reload counter
#define T1Mode3     (3 << 4)  //T1 mode3, T1 stop
#define T1GATE      (8 << 4)  //T1 gating clock by INT1
#define T1ClkFcpu    (0 << 4)  //T1 clock source from Fcpu/12
#define T1ClkExt     (4 << 4)  //T1 clock source from Fosc

void InitT0T1(void)
{
  // T0/T1_Initial
  TH0 = 0x00;
  TL0 = 0x00;
  TH1 = 0x00;
  TL1 = 0x00;
  // T0 mode0 clock source from Fosc or Flosc
  TMOD |= T0Mode0 | T0ClkExT;
  // T0 clock source = FRTC/1
  TCON0 |= T0ExtFlosc | 0x70;
  // T1 mode1, clock source from Fcpu/12
  TMOD |= T1Mode1 | T1ClkFcpu;
  // Timer 0/1 enable. Clear TF0/TF1
  TCON |= 0x50;
  // Enable T0/T1 interrupt
  IEN0 |= 0x0A;
  // Enable total interrupt
  IEN0 |= 0x80;

  P0 = 0x00;
  P0M = 0x03;
}

void T0Interrupt(void) interrupt ISRTimer0  //0x2B
{ //TF0 clear by hardware
  P00 = ~P00;
}
void T1Interrupt(void) interrupt ISRTimer1  //0xAB
{ //TF1 clear by hardware
  P01 = ~P01;
}

```

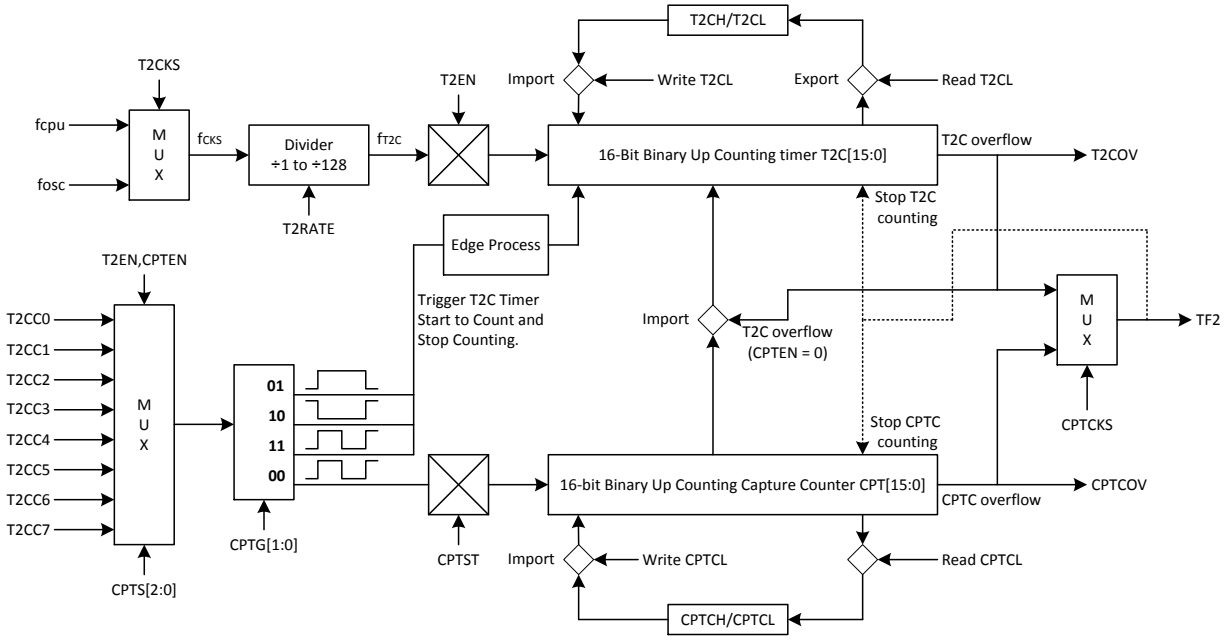
Copyright © 2019, SONiX Technology Co., Ltd.

15 Timer 2

Timer 2 includes a 16-bit binary up timer and a 16-bit binary up counter. If 16-bit timer occurs an overflow (from 0xFFFF to 0x0000) or the 16-bit counter occurs an overflow (from 0x0FFF to 0x0000), it will continue counting and issue a time-out signal to indicate Timer 2 time out event. Timer 2 can be capture timer to measure the pulse width and the cycle of input signal.

Timer 2 has six different operation modes: (1) 16-bit up counting timer with specified reload value support, (2) 16-bit low speed capture counter, (3) 16-bit high speed capture counter, (4) 16-bit high pulse width measurement counter, (5) 16-bit low pulse width measurement counter and (6) 16-bit cycle measurement counter. The measurement function have 8-input sources shared with GPIO pins and controlled by CPTS [2:0] bits.

Timer 2 builds in IDLE Mode wake-up function if interrupt enable. When timer overflow occurs or target measurement is completed, TF2 would be issued immediately which can read/write by firmware. Timer 2 interrupt function is controlled by ET2.



15.1 Timer 2 Clock Selection

The timer has two clock source options: fcpu, fosc. Timer selects the clock source via the T2CKS bit and selects the frequency division by the T2RATE [2: 0] bits. The Clock division range is /1 ~ /128.

The capture timer clock source and the measurement signal are from 8-input sources: T2CC0 ~ T2CC7. The CPTC 16-bit counter is triggered on the rising edge of the input signal for up-counting. Each complete trigger signal is a cycle.

Copyright © 2019, SONiX Technology Co., Ltd.

15.2 Operation Mode

Timer 2 selects six different operation modes through CPTEN, CPTMD and CPTG [1:0] bits. Each mode supports interrupt function. When timer overflow occurs or target measurement is completed, the TF2 is active and the system points program counter to interrupt vector to do interrupt sequence.

The CPTEN will determine Timer 2 is Timer mode or measurement mode. If CPTEN is Low, then Timer 2 is 16-bit up counting timer with specified reload value support. Conversely, Timer 2 can measure pulse width, cycle and capture duration of input signal. When CPTEN is high, the 16-bit up counting timer will stop for waiting measurement mode start.

The 16-bit CPTC counter is controlled by CPTEN bit, but the Timer 2 must be enabled. Set T2EN=1 and CPTEN=1 to enable capture or measurement function. The CPTC counter is a pure counter and no clock source to decide interval time.

Timer 2 can measure high pulse width, low pulse width, cycle and capture duration of input signal (T2CC0 ~ T2CC7) controlled by CPTG [1:0]. CPTST bit is to execute capture timer function. When CPTEN = 1 and CPTST is set as “1”, the T2C counter and the CPTC counter waits the right trigger edge to be activated. The trigger edge finds, and the 16-bit counter starts to count. When the second right edge finds or the capture is completed, the 16-counter will stop, CPTST is cleared and the TF2 is active.

The CPTG [1:0] bits select measurement modes: (1) 00: Capture Timer Function, (2) 01: Measure high pulse width of input pin, (3) 10: Measure low pulse width of input pin and (4) 11: Measure cycle of input pin. When measuring the pulse width or cycle of an input signal, Timer 2 uses only the T2C counter and ignores the CPTC counter.

Set CPTG [1:0] = 00 to enable capture timer function. The capture timer function’s purpose is to measure the period of a continuous signal. The function includes two modes for difference speed signal controlled by CPTMD bit. Each of overflow event occurs (controlled by CPTMD bit), the CPTC counter and T2C counter stop counting, CPTST bit is cleared, and TF2 is set as “1”.

Copyright © 2019, SONiX Technology Co., Ltd.

Table 15-1 The operation mode

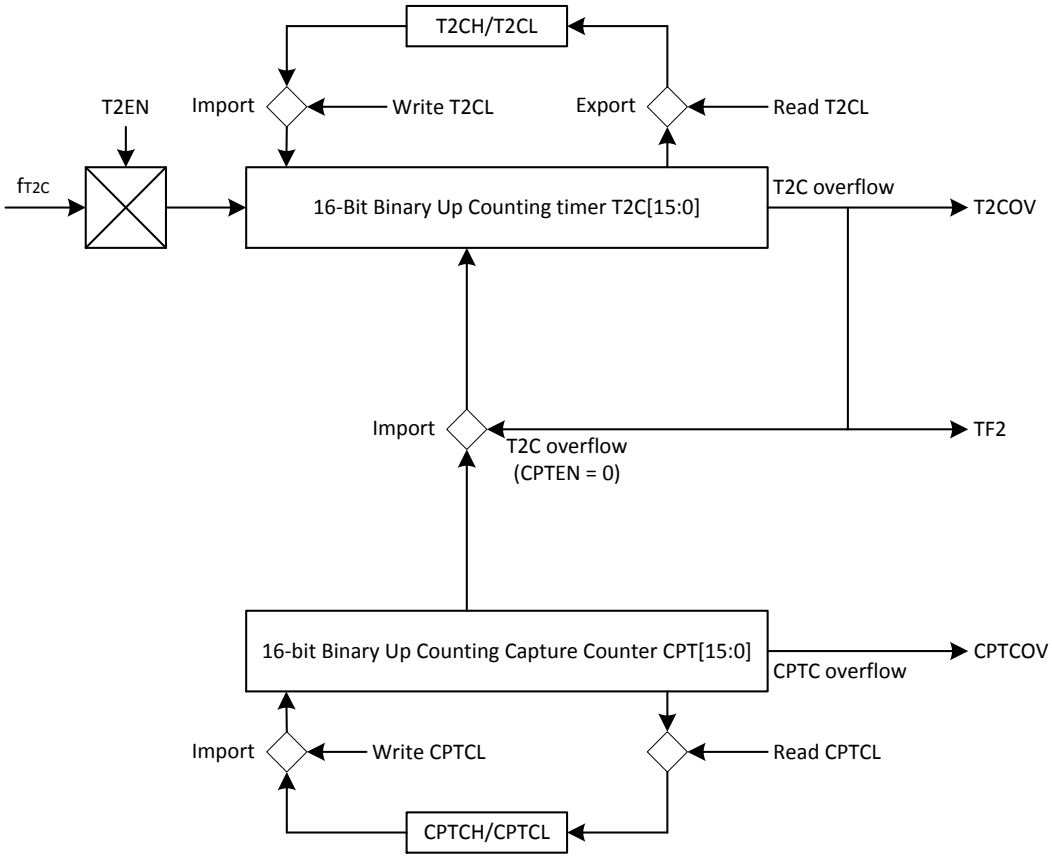
T2EN	0	1						
CPTEN	X	0	1					
CPTST	X	X	0	1				
CPTG[1:0]	X	X	X	00		01	10	11
CPTMD	X	X	X	0	1	X	X	X
T2C[15:0]	Stop	Up counting	Stop	Timer	Timer	Timer	Timer	Timer
CPTC[15:0]	Stop	Period reload buffer	Stop	Event Counter	Event Counter	inactive	inactive	inactive
Operation mode	Inactive	Timer	Wait	Low speed capture	High speed capture	High pulse width measurement	Low pulse width measurement	Cycle measurement
Auto reload	-	Enable	-	Disable	Disable	Disable	Disable	Disable
CPTST rising edge	-	-	-	Clear T2C[15:0]	**Clear CPTC[15:0]**	Clear T2C[15:0]	Clear T2C[15:0]	Clear T2C[15:0]
T2C[15:0] Start counting	-	T2EN = 1	-	First event rising	First event rising	Event rising edge	Event falling edge	First event rising edge
T2F	inactive	T2C[15:0] overflow	-	CPTC[15:0] overflow	T2C[15:0] overflow	Event falling edge	Event rising edge	Second event rising edge
CPTST is cleared	-	-	-	CPTC[15:0] overflow	T2C[15:0] overflow	Event falling edge	Event rising edge	Second event rising edge
CPTC[15:0] overflow	-	-	-	CPTCOV = 1, T2F = 1, CPTST = 0, CPTC[15:0] & T2C[15:0] stop	**CPTCOV = 1, CPTC[15:0] & T2C[15:0] run**	-	-	-
T2C[15:0] overflow	-	T2COV = 1, T2F = 1, T2C[15:0] keep run	-	**T2COV = 1, CPTC[15:0] & T2C[15:0] keep run**	T2COV = 1, T2F = 1, CPTST = 0, CPTC[15:0] & T2C[15:0] stop	T2COV = 1, T2C[15:0] keep run	T2COV = 1, T2C[15:0] keep run	T2COV = 1, T2C[15:0] keep run

T2COV flag indicates T2CH/T2CL 16-bit timer overflow and cleared by program. CPTCOV flag indicates CPTCH/CPTCL 16-bit counter overflow and cleared by program.

Copyright © 2019, SONiX Technology Co., Ltd.

15.3 Timer Mode

Timer mode is controlled by T2EN bit when CPTEN = 0. When T2EN=1, T2 timer starts to count. One count period is one clock source rate. T2C is a 16-bit binary up counter and stored in T2CH and T2CL registers. When T2C counts from 0xFFFF to 0x0000, T2 overflow condition is conformed and TF2 set as “1”. When Timer overflow occurs, the T2CH and T2CL are reload from CPTCH and CPTCL registers. T2CH and T2CL are double buffer design. If read and write T2C 16-bit counter, use read/writer T2CL to control 16-bit data latch to internal buffers. If T2 interrupt function is enabled (ET2=1), the program counter is pointed to interrupt vector to execute interrupt service routine after T2 timer overflow occurrence.



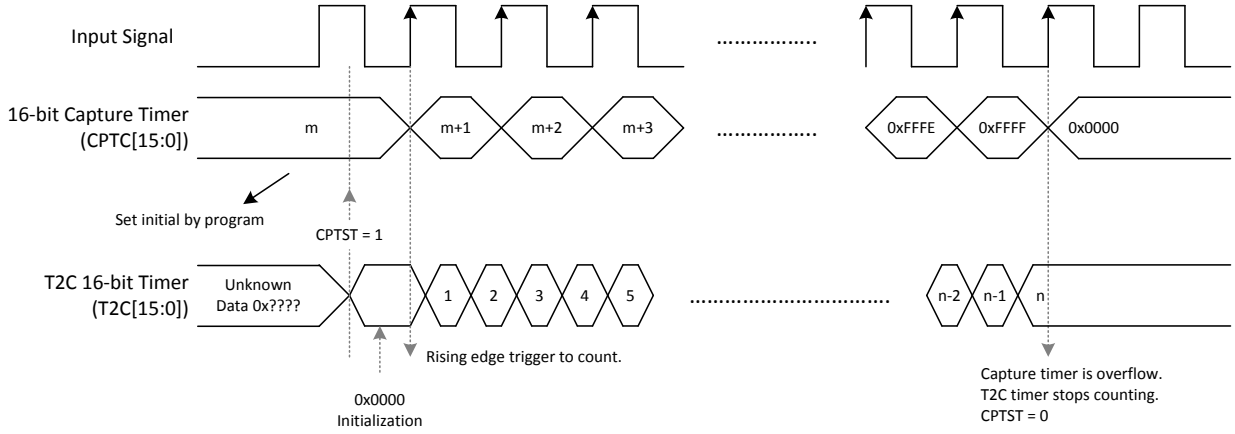
✸ ***Note:***

1. ***When CPTEN = 0, the timer reload function is enable. The timer counter reloads period value from CPTCH/CPTCL when timer counter overflow occurs.***
2. ***When CPTEN = 1, the timer reload function is disable.***

Copyright © 2019, SONiX Technology Co., Ltd.

15.4 Low Speed Capture Mode

The low speed capture mode is enabled when CPTEN = 1, CPTG [1:0] = 00 and CPTMD = 0. Input signal rate < T2 timer rate. Use T2C timer to measure input signal continuous duration. Set capture timer initial value (CPTCH, CPTCL = “m”) and clear T2C timer (T2CH, T2CL = 0x0000) by program. Set CPTST bit to start capture timer counting. Capture timer and T2C timer start counting at the first rising edge of input signal. When capture timer overflow occurs (0xFFFF to 0x0000), T2C timer stops counting, CPTST is cleared automatically, and the TF2 sets as “1”. The T2C counter value (T2CH, T2CL = “n”) is the continuous signal’s duration.

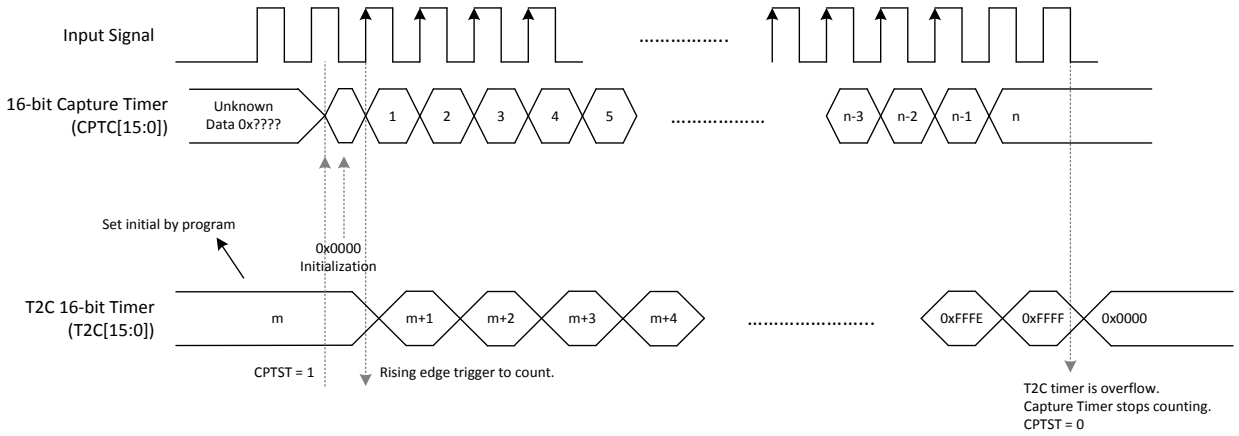


✸ ***Note: When CPTST rising edge, the T2C counter will be clear.***

15.5 High Speed Capture Mode

The high speed capture mode is enabled when CPTEN = 1, CPTG [1:0] = 00 and CPTMD = 1. Input signal rate > T2 timer rate. Set a unique timer by T2C timer to measure input signal counts. Set T2C timer initial value (T2CH, T2CL = “m”) and clear capture timer counter (CPTCH, CPTCL = 0x0000) by program. Set CPTST bit to start capture timer counting. Capture timer and T2C timer start counting at the first rising edge of input signal. When T2C timer overflow occurs (0xFFFF to 0x0000), capture timer stops counting, CPTST is cleared automatically, and the TF2 sets as “1”. The capture timer 16-bit counter value (CPTCH, CPTCL = “n”) is the continuous signal’s counts.

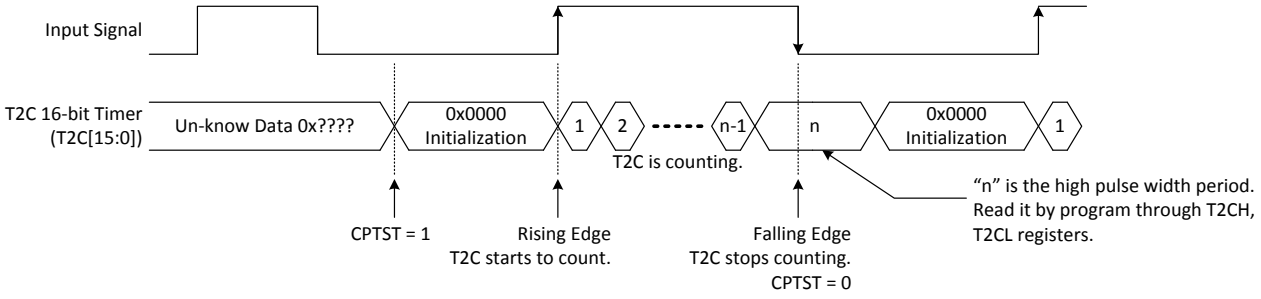
Copyright © 2019, SONiX Technology Co., Ltd.



✸ ***Note: When CPTST rising edge, the capture counter will be clear.***

15.6 High Pulse Width Measurement Mode

The high pulse width measurement mode is enabled when CPTEN = 1 and CPTG [1:0] = 01. The mode is using the rising edge of input signal to start T2C 16-bit timer and the falling edge of input signal to stop T2C 16-bit timer. If set CPTST bit in high pulse duration, the T2C timer will measure next high pulse until the rising edge occurrence. When the end of measuring high pulse width and T2C timer stops, the TF2 sets as "1", and the T2 interrupt executes as ET2 = 1.

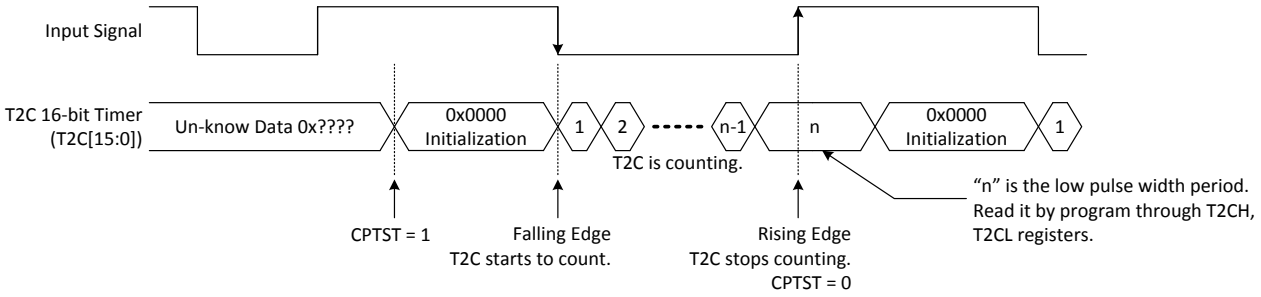


✸ ***Note: When CPTST rising edge, the T2C counter will be clear.***

15.7 Low Pulse Width Measurement Mode

The low pulse width measurement mode is enabled when CPTEN = 1 and CPTG [1:0] = 10. The mode is using the falling edge of input signal to start T2C 16-bit timer and the rising edge of input signal to stop T2C 16-bit timer. If set CPTST bit in low pulse duration, the T2C timer will measure next low pulse until the rising edge occurrence. When the end of measuring low pulse width and T2C timer stops, the TF2 sets as "1", and the T2 interrupt executes as ET2 = 1.

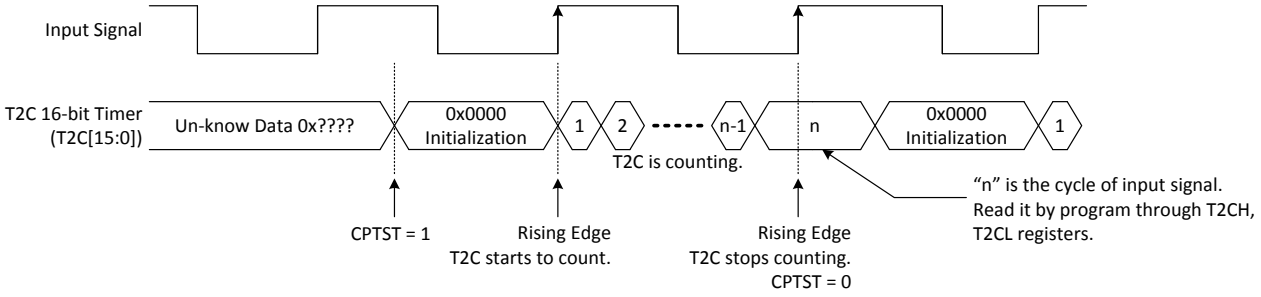
Copyright © 2019, SONiX Technology Co., Ltd.



✸ ***Note: When CPTST rising edge, the T2C counter will be clear.***

15.8 Cycle Measurement Mode

The cycle measurement mode is enabled when CPTEN = 1 and CPTG [1:0] = 11. The mode is using two the rising edge of input signal to start and stop T2C 16-bit timer. If set CPTST bit in high pulse duration, low pulse duration or falling edge, the T2C timer will measure next cycle until the rising edge occurrence. When the end of measuring cycle and T2C timer stops, the TF2 sets as "1", and the T2 interrupt executes as ET2 = 1.



✸ ***Note: When CPTST rising edge, the T2C counter will be clear.***

15.9 Timer 2 Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2M	T2EN	T2RATE2	T2RATE1	T2RATE0	T2CKS	-	T2COV	CPTCOV
T2CH	T2C15	T2C14	T2C13	T2C12	T2C11	T2C10	T2C9	T2C8
T2CL	T2C7	T2C6	T2C5	T2C4	T2C3	T2C2	T2C1	T2C0
CPTM	CPTEN	CPTS2	CPTS1	CPTS0	CPTMD	CPTST	CPTG1	CPTG0
CPTCH	CPTC15	CPTC14	CPTC13	CPTC12	CPTC11	CPTC10	CPTC9	CPTC8
CPTCL	CPTC7	CPTC6	CPTC5	CPTC4	CPTC3	CPTC2	CPTC1	CPTC0
IEN0	EAL	EX4	EX3	EX2	ET1	EX1	ET0	EX0
IEN2	ET4	ELCD	ET3	ET2	EPW2	EPW1	-	EADC
IRCON2	TF4	LCDF	TF3	TF2	PW2F	PW1F	-	ADCF

T2M Register (0xAC)

Bit	Field	Type	Initial	Description
7	T2EN	R/W	0	Timer 2 function enable bit 0: Disable Timer 2 module 1: Enable Timer 2 module
6..4	T2RATE[2:0]	R/W	000	Clock divider of Timer 2 f_{CKS} clock source 000: f_{CKS} / 128 001: f_{CKS} / 64 010: f_{CKS} / 32 011: f_{CKS} / 16 100: f_{CKS} / 8 101: f_{CKS} / 4 110: f_{CKS} / 2 111: f_{CKS} / 1
3	T2CKS	R/W	0	Timer 2 f_{CKS} clock source selection. 0: fcpu 1: fosc
2	Reserved	R	0	
1	T2COV	R/W	0	16-bit T2C timer overflow flag 0: Non overflow 1: 16-bit timer T2C[15:0] overflows
0	CPTCOV	R/W	0	16-bit capture counter overflow flag 0: Non overflow 1: 16-bit capture counter CPTC[15:0] overflows

Copyright © 2019, SONiX Technology Co., Ltd.

T2CH/T2CL Registers (T2CH: 0xAE, T2CL: 0xAD)

Bit	Field	Type	Initial	Description
7..0	T2CH/T2CL	R/W	0x00	Timer 2 16-bit timer registers. *

* Read T2 16-bit timer buffer sequence is to read T2CL first, and then read T2CH. Write T3 16-bit timer buffer sequence is to write T2CH first, and then write T2CL.

CPTM Register (0xAF)

Bit	Field	Type	Initial	Description
7	CPTEN	R/W	0	Measurement function enable bit 0: Disable 1: Enable. T2EN must be enabling.
6..4	CPTS[2:0]	R/W	0x00	Measurement function input source select bit. 000: T2CC0, 001: T2CC1, 010: T2CC2, 011: T2CC3, 100: T2CC4, 101: T2CC5, 110: T2CC6, 111: T2CC7
3	CPTMD	R/W	0	Capture timer mode control bit 0: Low speed capture mode; CPTC overflow mode. 1: high speed capture mode; T2C overflow mode.
2	CPTST	R/W	0	16-bit timer and 16-bit capture counter active control bit 0: Process end 1: Start count and processing
1..0	CPTG[1:0]	R/W	00	Measurement mode select bit 00: Capture timer mode. 01: High pulse width measurement mode. 10: Low pulse width measurement mode. 11: Cycle measurement mode.

CPTCH/CPTCL Registers (CPTCH: 0xAE, CPTCL: 0xAD)

Bit	Field	Type	Initial	Description
7..0	CPTCH/CPTCL	R/W	0x00	Timer 2 16-bit capture counter registers. *

* Read T2 16-bit timer buffer sequence is to read CPTCL first, and then read CPTCH. Write T3 16-bit timer buffer sequence is to write CPTCH first, and then write CPTCL.

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)

Copyright © 2019, SONiX Technology Co., Ltd.

IEN2 Register (0x9A)

Bit	Field	Type	Initial	Description
4	ET2	R/W	0	Timer 2 interrupt control bit. 0: Disable 1: Enable
Else				Refer to other chapter(s)

IRCON2 Register (0xBF)

Bit	Field	Type	Initial	Description
4	TF2	R/W	0	Timer 2 overflow event or measurement is completed. 0: Timer 2 does not have any overflow event, or measurement is not completed. 1: The flag is set when timer overflow occurs or target measurement is completed. This bit can be cleared automatically by interrupt handler, or manually by firmware
Else				Refer to other chapter(s)

Copyright © 2019, SONiX Technology Co., Ltd.

15.10 Sample Code

The following sample code demonstrates how to perform T2 with interrupt.

```
#define LowCaptureMode  (0 << 0)  //Enable low speed capture mode
#define HighCaptureMode  (8 << 0)  //Enable high speed capture mode
#define HighPulseMode  (1 << 0)  //Enable high pulse width measurement mode
#define LowPulseMode  (2 << 0)  //Enable low pulse width measurement mode
#define CycleMode    (3 << 7)  //Enable Cycle measurement mode
#define CPTEn        (1 << 7)  //Enable measurement function
#define EnT2CC7        (7 << 4)  //Enable T2CC7/P2.4 channel
#define T2ClkFcpu    (0 << 3)  //T2 clock source from Fcpu
#define T2ClkFosc    (1 << 3)  //T2 clock source from Fosc
#define T2En         (1 << 7)  //Enable T2 function
#define SetCPTST     (1 << 2)  //Start Measurement

void InitT2(void)
{
  // Timer 2_Initial
  T2CH = 0x00;
  T2CL = 0x00;
  // Enable Low speed capture mode, select T2CC7 channel, initial counter
  P2M &= 0xEF;
  CPTM = CPTEn | EnT2CC7 | LowCaptureMode;
  CPTCH = 0x80;
  CPTCL = 0x00;
  // T2 enable, clock = Fosc/32, clear T2COV, CPTCOV
  T2M = T2En | 0x20 | T2ClkFosc & 0xFC;

  // Enable T2 interrupt & clear TF2
  IEN2 = 0x10;
  IRC0N2 &= 0xEF;

  // Enable total interrupt
  IEN0 |= 0x80;

  P0 = 0x00;
  P0M |= 0x01;

  // Start Measurement, T2CC7 wait event clock
  CPTM |= SetCPTST;
}

void T2Interrupt(void) interrupt ISRTimer2  //0xEB
{ //TF2 clear by hardware
  P00 = ~P00;
}
```

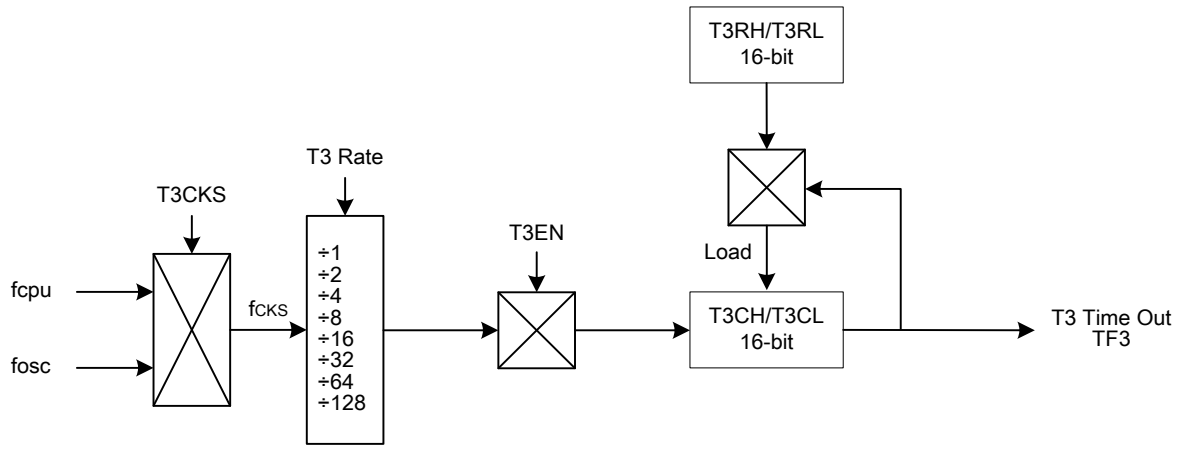
16 Timer 3

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

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

16.1 Timer 3 Clock Selection

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



16.2 Timer 3 Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T3M	T3EN	T3RATE2	T3RATE1	T3RATE0	T3CKS	-	-	-
T3CH	T3C15	T3C14	T3C13	T3C12	T3C11	T3C10	T3C9	T3C8
T3CL	T3C7	T3C6	T3C5	T3C4	T3C3	T3C2	T3C1	T3C0
T3RH	T3R15	T3R14	T3R13	T3R12	T3R11	T3R10	T3R9	T3R8
T3RL	T3R7	T3R6	T3R5	T3R4	T3R3	T3R2	T3R1	T3R0
IEN0	EAL	EX4	EX3	EX2	ET1	EX1	ET0	EX0
IEN2	ET4	ELCD	ET3	ET2	EPW2	EPW1	-	EADC
IRCON2	TF4	LCDF	TF3	TF2	PW2F	PW1F	-	ADCF

T3M Register (0xB1)

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

Copyright © 2019, SONiX Technology Co., Ltd.

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

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

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

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

IEN0 Register (0xA8)

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

IEN2 Register (0x9A)

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

IRCON2 Register (0xBF)

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

Copyright © 2019, SONiX Technology Co., Ltd.

16.3 Sample Code

The following sample code demonstrates how to perform T3 with interrupt.

```
#define T3ClkFcpu    (0 << 3)  //T3 clock source from Fcpu
#define T3ClkFosc    (1 << 3)  //T3 clock source from Fosc
#define T3En         (1 << 7)  //Enable T3 function

void InitT3(void)
{
  // Timer 3_Initial
  T3CH = 0x80;
  T3CL = 0x00;
  T3RH = 0x40;
  T3RL = 0x00;

  // T3 enable, clock = Fosc/32
  T3M = T3En | 0x20 | T3ClkFosc;

  // Enable T3 interrupt & clear TF3
  IEN2 = 0x20;
  IRCON2 &= 0xDF;

  // Enable total interrupt
  IEN0 |= 0x80;

  P0 = 0x00;
  P0M |= 0x01;
}

void T3Interrupt(void) interrupt ISRTimer3  //0x6B
{ //TF3 clear by hardware
  P00 = ~P00;
}
```

Copyright © 2019, SONiX Technology Co., Ltd.

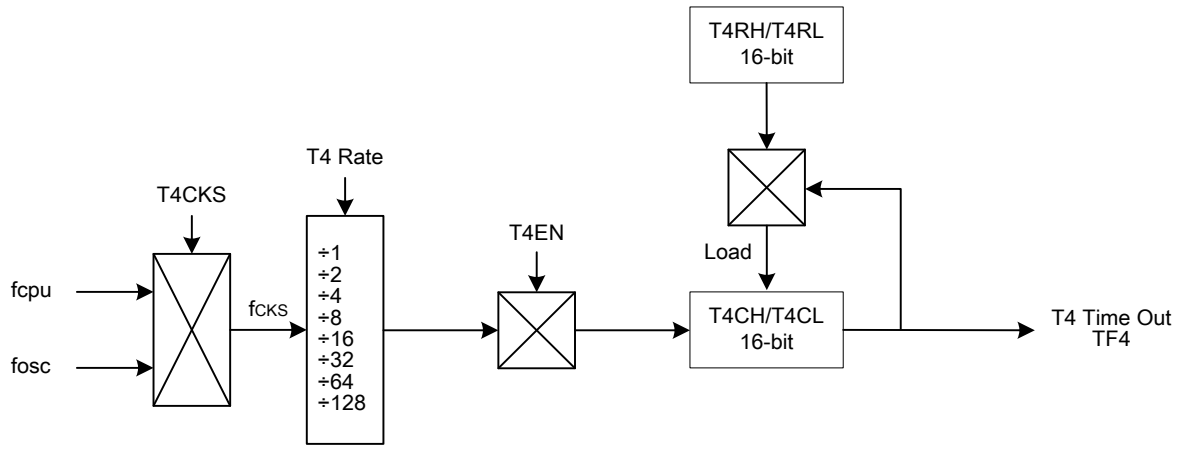
17 Timer 4

Timer 4 has 16-bit up counting timer with specified reload value support. An overflow event (T4CH/T4CL counts from 0xFFFF to 0x0000) issues its TF4 flag for firmware or interrupt controller; meanwhile, the timer duplicates T4RH/T4RL value to T4CH/T4CL register in the same time. As a result, the timer is actually counts from 0xFFFF to the value of T4RH/T4RL.

The Timer 4 build in IDLE Mode wake-up function if interrupt enable. When timer overflow occurs (counts from 0xFFFF to 0x0000), TF4 would be issued immediately which can read/write by firmware. Timer 4 interrupt function is controlled by ET4.

17.1 Timer 4 Clock Selection

Timer 4 has two clock source options: fcpu, fosc. Timer 4 selects the clock source via the T4CKS bit and selects the frequency division by the T4RATE [2: 0] bits. The Clock division range is /1 ~ /128.



17.2 Timer 4 Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T4M	T4EN	T4RATE2	T4RATE1	T4RATE0	T4CKS	-	-	-
T4CH	T4C15	T4C14	T4C13	T4C12	T4C11	T4C10	T4C9	T4C8
T4CL	T4C7	T4C6	T4C5	T4C4	T4C3	T4C2	T4C1	T4C0
T4RH	T4R15	T4R14	T4R13	T4R12	T4R11	T4R10	T4R9	T4R8
T4RL	T4R7	T4R6	T4R5	T4R4	T4R3	T4R2	T4R1	T4R0
IEN0	EAL	EX4	EX3	EX2	ET1	EX1	ET0	EX0
IEN2	ET4	ELCD	ET3	ET2	EPW2	EPW1	-	EADC
IRCON2	TF4	LCDF	TF3	TF2	PW2F	PW1F	-	ADCF

T4M Register (0xF4)

Bit	Field	Type	Initial	Description
7	T4EN	R/W	0	Timer 4 function 0: Disable 1: Enable
6..4	T4RATE[2:0]	R/W	000	Clock divider of Timer 4 f_{CKS} clock source 000: f_{CKS} / 128 001: f_{CKS} / 64 010: f_{CKS} / 32 011: f_{CKS} / 16 100: f_{CKS} / 8 101: f_{CKS} / 4 110: f_{CKS} / 2 111: f_{CKS} / 1
3	T4CKS	R/W	0	Timer 4 f_{CKS} clock source selection. 0: fcpu 1: fosc
2..0	Reserved	R/W	000	

Copyright © 2019, SONiX Technology Co., Ltd.

T4CH/T4CL Registers (T4CH: 0xF6, T4CL: 0xF5)

Bit	Field	Type	Initial	Description
7..0	T4CH/L	R/W	0x00	16-bit Timer 4 counter buffer.

T4RH/T4RL Registers (T4RH: 0xE2, T4RL: 0xE1)

Bit	Field	Type	Initial	Description
7..0	T4RH/L	R/W	0x00	16-bit Timer 4 counter reload buffer.

IEN0 Register (0xA8)

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

IEN2 Register (0x9A)

Bit	Field	Type	Initial	Description
7	ET4	R/W	0	Timer 4 interrupt control bit. 0: Disable 1: Enable
Else				Refer to other chapter(s)

IRCON2 Register (0xBF)

Bit	Field	Type	Initial	Description
7	TF4	R/W	0	Timer 4 overflow event 0: Timer 4 does not have any overflow event 1: Timer 4 has overflowed This bit can be cleared automatically by interrupt handler, or manually by firmware
Else				Refer to other chapter(s)

Copyright © 2019, SONiX Technology Co., Ltd.

17.3 Sample Code

The following sample code demonstrates how to perform T4 with interrupt.

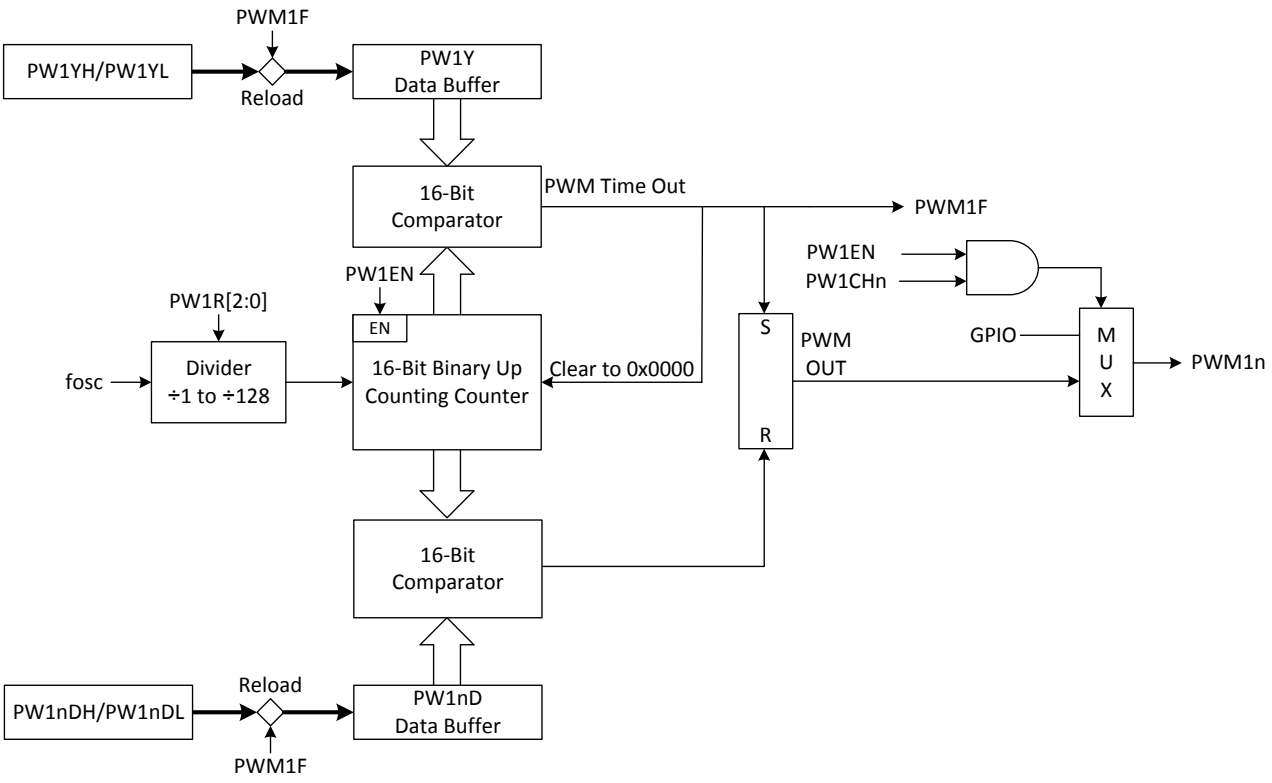
```
#define T4ClkFcpu    (0 << 3)  //T4 clock source from Fcpu
#define T4ClkFosc    (1 << 3)  //T4 clock source from Fosc
#define T4En         (1 << 7)  //Enable T4 function

void InitT4(void)
{
  // Timer 4_Initial
  T4CH = 0x80;
  T4CL = 0x00;
  T4RH = 0x40;
  T4RL = 0x00;

  // T4 enable, clock = Fosc/32
  T4M = T4En | 0x20 | T4ClkFosc;

  // Enable T4 interrupt & clear TF4
  IEN2 = 0x80;
  IRCON2 &= 0x7F;

  // Enable total interrupt
  IEN0 |= 0x80;

  P0 = 0x00;
  P0M |= 0x01;
}

void T4Interrupt(void) interrupt ISRTimer4  //0xBB
{ //TF4 clear by hardware
  P00 = ~P00;
}
```

Copyright © 2019, SONiX Technology Co., Ltd.

18 PWM1

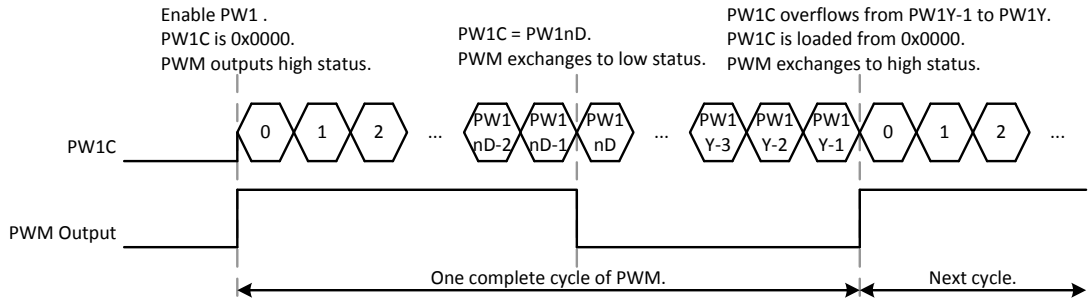
The PW1 timer is a 16-bit up counting timer and supports 4-channel general PWM function. By the counter reaches the up-boundary value (PW1Y), it clears its counter and triggers an interrupt signal. PWM's duty cycle is controlled by PW10D~PW13D register. Each PWM channel has its own duty control.

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



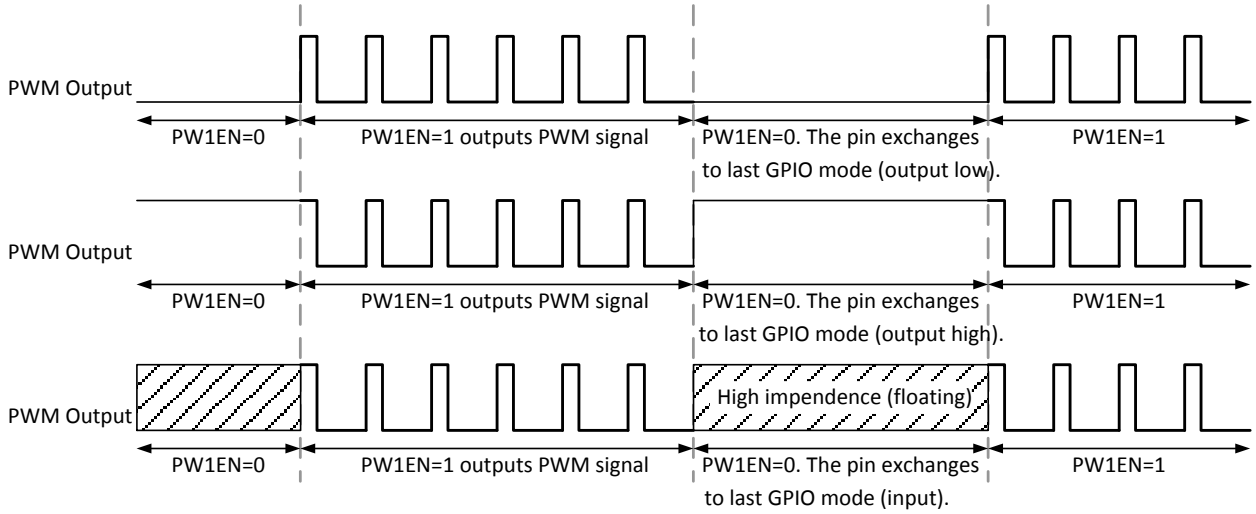
18.1 General PWM

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



PWM Period = PW1Y

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



Copyright © 2019, SONiX Technology Co., Ltd.

18.2 PWM1 Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW1M	PW1EN	PW1R2	PW1R1	PW1R0	PW1CH3	PW1CH2	PW1CH1	PW1CH0
PW1YH	PW1Y15	PW1Y14	PW1Y13	PW1Y12	PW1Y11	PW1Y10	PW1Y9	PW1Y8
PW1YL	PW1Y7	PW1Y6	PW1Y5	PW1Y4	PW1Y3	PW1Y2	PW1Y1	PW1Y0
PW10DH	PW10D15	PW10D14	PW10D13	PW10D12	PW10D11	PW10D10	PW10D9	PW10D8
PW10DL	PW10D7	PW10D6	PW10D5	PW10D4	PW10D3	PW10D2	PW10D1	PW10D0
PW11DH	PW11D15	PW11D14	PW11D13	PW11D12	PW11D11	PW11D10	PW11D9	PW11D8
PW11DL	PW11D7	PW11D6	PW11D5	PW11D4	PW11D3	PW11D2	PW11D1	PW11D0
PW12DH	PW12D15	PW12D14	PW12D13	PW12D12	PW12D11	PW12D10	PW12D9	PW12D8
PW12DL	PW12D7	PW12D6	PW12D5	PW12D4	PW12D3	PW12D2	PW12D1	PW12D0
PW13DH	PW13D15	PW13D14	PW13D13	PW13D12	PW13D11	PW13D10	PW13D9	PW13D8
PW13DL	PW13D7	PW13D6	PW13D5	PW13D4	PW13D3	PW13D2	PW13D1	PW13D0
IEN0	EAL	EX4	EX3	EX2	ET1	EX1	ET0	EX0
IEN2	ET4	ELCD	ET3	ET2	EPW2	EPW1	-	EADC
IRCON2	TF4	LCDF	TF3	TF2	PW2F	PW1F	-	ADCF

Copyright © 2019, SONiX Technology Co., Ltd.

PW1M Registers (PW1M: 0xF000)

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	PW1CH3	R/W	0	PWM13 shared-pin control 0: GPIO 1: PWM output (shared with P1.3)
2	PW1CH2	R/W	0	PWM12 shared-pin control 0: GPIO 1: PWM output (shared with P1.2)
1	PW1CH1	R/W	0	PWM11 shared-pin control 0: GPIO 1: PWM output (shared with P1.1)
0	PW1CH0	R/W	0	PWM10 shared-pin control 0: GPIO 1: PWM output (shared with P1.0)

* 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: 0xF002, PW1YL: 0xF003)

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: 0xF004, PW10DL: 0xF005)

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

Copyright © 2019, SONiX Technology Co., Ltd.

PW11DH/PW11DL Registers (PW11DH: 0xF006, PW11DL: 0xF007)

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

PW12DH/PW12DL Registers (PW12DH: 0xF008, PW12DL: 0xF009)

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

PW13DH/PW13DL Registers (PW13DH: 0xF00A, PW13DL: 0xF00B)

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

IEN0 Register (0xA8)

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

IEN2 Register (0x9A)

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

IRCON2 Register (0xBF)

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

Copyright © 2019, SONiX Technology Co., Ltd.

18.3 Sample Code

The following sample code demonstrates how to perform PW1 with interrupt.

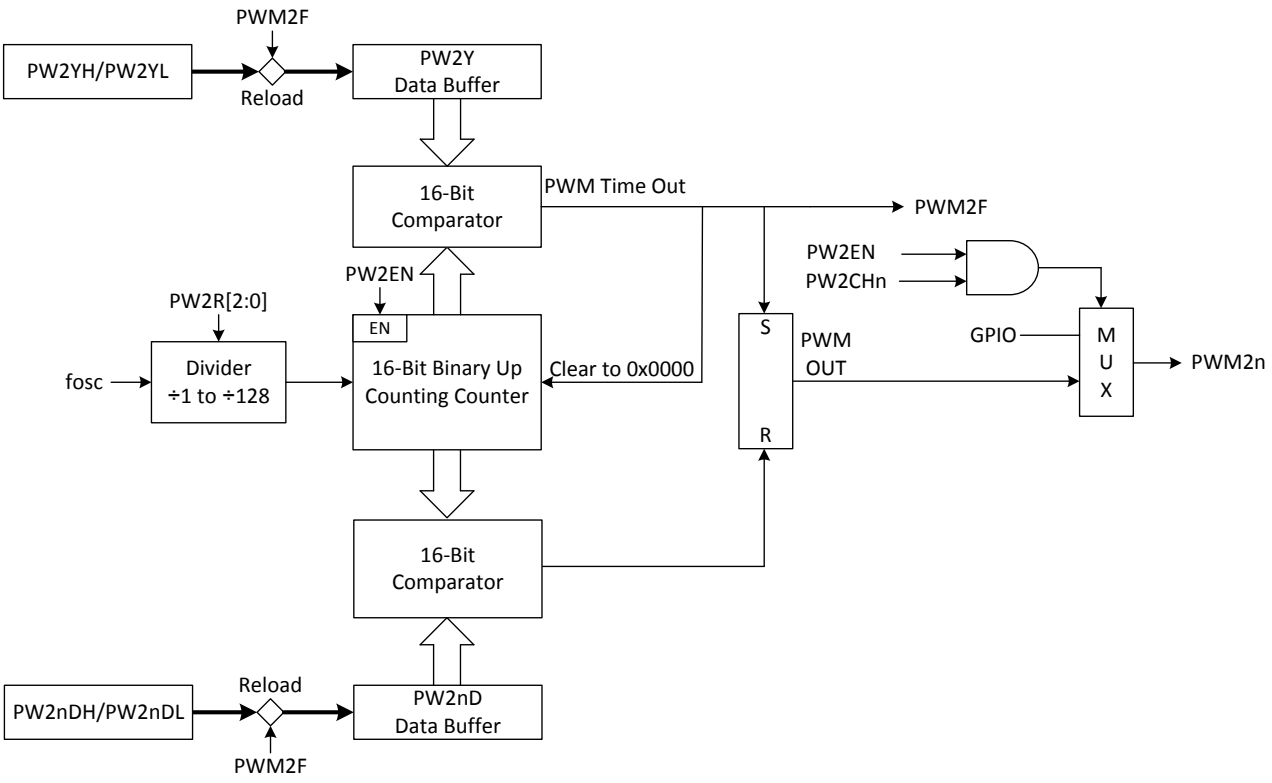
```
#define PWM10En     (1 << 0)  //Enable PWM10 output function
#define PW1En       (1 << 7)  //Enable PWM1 function

void InitPWM(void)
{
  // PWM1_Initial
  PW1YH = 0x80;
  PW1YL = 0x00;
  PW10DH = 0x40;
  PW10DL = 0x00;

  // PWM1 enable, clock = Fosc/32, PWM10 channel enable
  PW1M = PW1En | 0x20 | PWM10En;

  // Enable PWM1 interrupt & clear PWM1F
  IEN2 = 0x04;
  IRC0N2 = 0x00;

  // Enable total interrupt
  IEN0 |= 0x80;

  P0 = 0x00;
  P0M |= 0x01;
}

void PW1Interrupt(void) interrupt ISRPwm1  //0x23
{ //PWM1F clear by hardware
  P00 = ~P00;
}
```

Copyright © 2019, SONiX Technology Co., Ltd.

19 PWM2

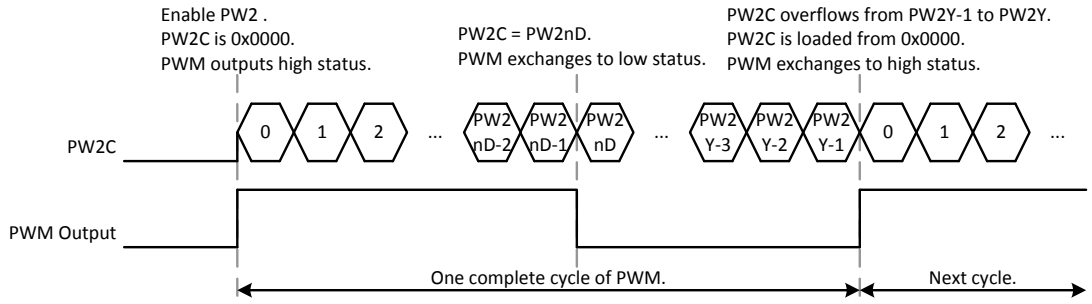
The PW2 timer is a 16-bit up counting timer and supports 4-channel general PWM function. By the counter reaches the up-boundary value (PW2Y), it clears its counter and triggers an interrupt signal. PWM's duty cycle is controlled by PW20D~PW23D register. Each PWM channel has its own duty control.

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



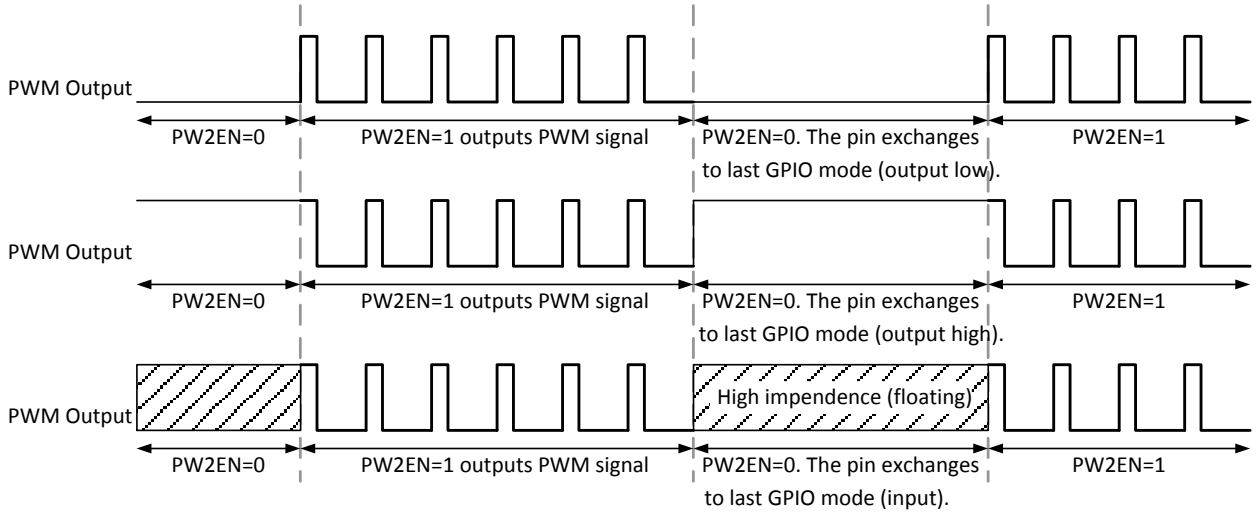
19.1 General PWM

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



PWM Period = PW2Y

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



Copyright © 2019, SONiX Technology Co., Ltd.

19.2 PWM2 Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW2M	PW2EN	PW2R2	PW2R1	PW2R0	PW2CH3	PW2CH2	PW2CH1	PW2CH0
PW2YH	PW2Y15	PW2Y14	PW2Y13	PW2Y12	PW2Y11	PW2Y10	PW2Y9	PW2Y8
PW2YL	PW2Y7	PW2Y6	PW2Y5	PW2Y4	PW2Y3	PW2Y2	PW2Y1	PW2Y0
PW20DH	PW20D15	PW20D14	PW20D13	PW20D12	PW20D11	PW20D10	PW20D9	PW20D8
PW20DL	PW20D7	PW20D6	PW20D5	PW20D4	PW20D3	PW20D2	PW20D1	PW20D0
PW21DH	PW21D15	PW21D14	PW21D13	PW21D12	PW21D11	PW21D10	PW21D9	PW21D8
PW21DL	PW21D7	PW21D6	PW21D5	PW21D4	PW21D3	PW21D2	PW21D1	PW21D0
PW22DH	PW22D15	PW22D14	PW22D13	PW22D12	PW22D11	PW22D10	PW22D9	PW22D8
PW22DL	PW22D7	PW22D6	PW22D5	PW22D4	PW22D3	PW22D2	PW22D1	PW22D0
PW23DH	PW23D15	PW23D14	PW23D13	PW23D12	PW23D11	PW23D10	PW23D9	PW23D8
PW23DL	PW23D7	PW23D6	PW23D5	PW23D4	PW23D3	PW23D2	PW23D1	PW23D0
IEN0	EAL	EX4	EX3	EX2	ET1	EX1	ET0	EX0
IEN2	ET4	ELCD	ET3	ET2	EPW2	EPW1	-	EADC
IRCON2	TF4	LCDF	TF3	TF2	PW2F	PW1F	-	ADCF

Copyright © 2019, SONiX Technology Co., Ltd.

PW2M Registers (PW2M: 0xF014)

Bit	Field	Type	Initial	Description
7	PW2EN	R/W	0	PW2 function 0: Disable 1: Enable*
6..4	PW2R[2:0]	R/W	000	PWM timer clock source 000: fosc / 128 001: fosc / 64 010: fosc / 32 011: fosc / 16 100: fosc / 8 101: fosc / 4 110: fosc / 2 111: fosc / 1
3	PW2CH3	R/W	0	PWM23 shared-pin control 0: GPIO 1: PWM output (shared with P6.3)
2	PW2CH2	R/W	0	PWM22 shared-pin control 0: GPIO 1: PWM output (shared with P6.2)
1	PW2CH1	R/W	0	PWM21 shared-pin control 0: GPIO 1: PWM output (shared with P6.1)
0	PW2CH0	R/W	0	PWM20 shared-pin control 0: GPIO 1: PWM output (shared with P6.0)

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

PW2YH/PW2YL Registers (PW2YH: 0xF016, PW2YL: 0xF017)

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

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

PW20DH/PW20DL Registers (PW20DH: 0xF018, PW20DL: 0xF019)

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

Copyright © 2019, SONiX Technology Co., Ltd.

PW21DH/PW21DL Registers (PW21DH: 0xF01A, PW21DL: 0xF01B)

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

PW22DH/PW22DL Registers (PW22DH: 0xF01C, PW22DL: 0xF01D)

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

PW23DH/PW23DL Registers (PW23DH: 0xF01E, PW23DL: 0xF01F)

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

IEN0 Register (0xA8)

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

IEN2 Register (0x9A)

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

IRCON2 Register (0xBF)

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

Copyright © 2019, SONiX Technology Co., Ltd.

19.3 Sample Code

The following sample code demonstrates how to perform PW2 with interrupt.

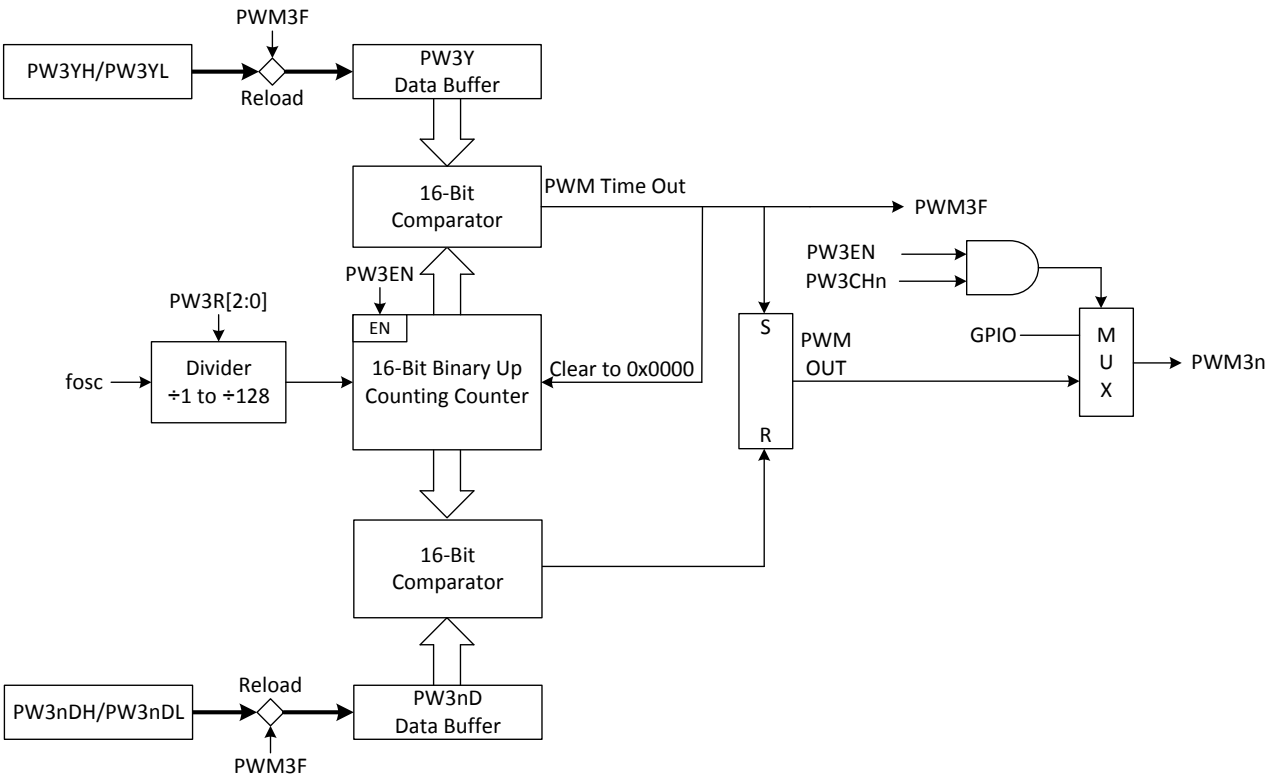
```
#define PWM20En     (1 << 0)  //Enable PWM20 output function
#define PW2En       (1 << 7)  //Enable PWM2 function

void InitPWM(void)
{
  // PWM2_Initial
  PW2YH = 0x80;
  PW2YL = 0x00;
  PW20DH = 0x40;
  PW20DL = 0x00;

  // PWM2 enable, clock = Fosc/32, PWM20 channel enable
  PW2M = PW2En | 0x20 | PWM20En;

  // Enable PWM2 interrupt & clear PWM2F
  IEN2 = 0x04;
  IRC0N2 = 0x00;

  // Enable total interrupt
  IEN0 |= 0x80;

  P0 = 0x00;
  P0M |= 0x01;
}

void PW2Interrupt(void) interrupt ISRPwm2  //0x63
{ //PWM2F clear by hardware
  P00 = ~P00;
}
```

Copyright © 2019, SONiX Technology Co., Ltd.

20 PWM3

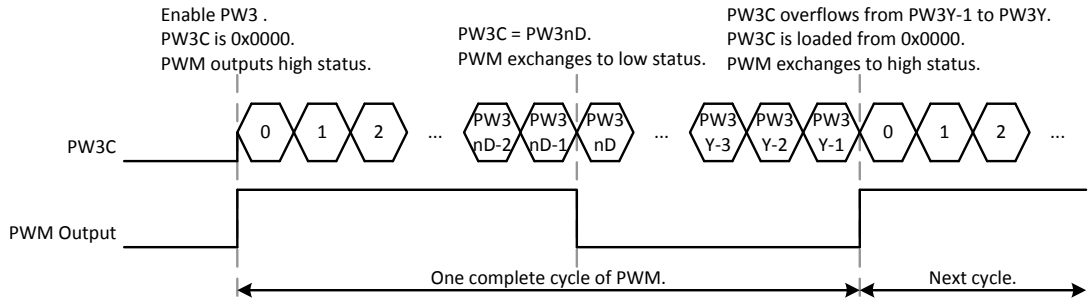
The PW3 timer is a 16-bit up counting timer and supports 4-channel general PWM function. By the counter reaches the up-boundary value (PW3Y), it clears its counter and triggers an interrupt signal. PWM's duty cycle is controlled by PW30D~PW33D register. Each PWM channel has its own duty control.

The PWM function has 4 programmable channels shared with GPIO pins and controlled by PW3CH[3:0] bits. The output operation must be through enabled each bit/channel of PW3CH[3:0] bits. The enabled PWM channel exchanges from GPIO to PWM output. When the PW3CH[3:0] bits disables, the PWM channel returns to last status of GPIO mode. The PW3 timer builds in IDLE Mode wake-up function if interrupt enable. When timer overflow occurs (counts from PW3Y-1 to PW3Y), PW3F would be issued immediately which can read/write by firmware. PW3 interrupt function is controlled by EPW3.



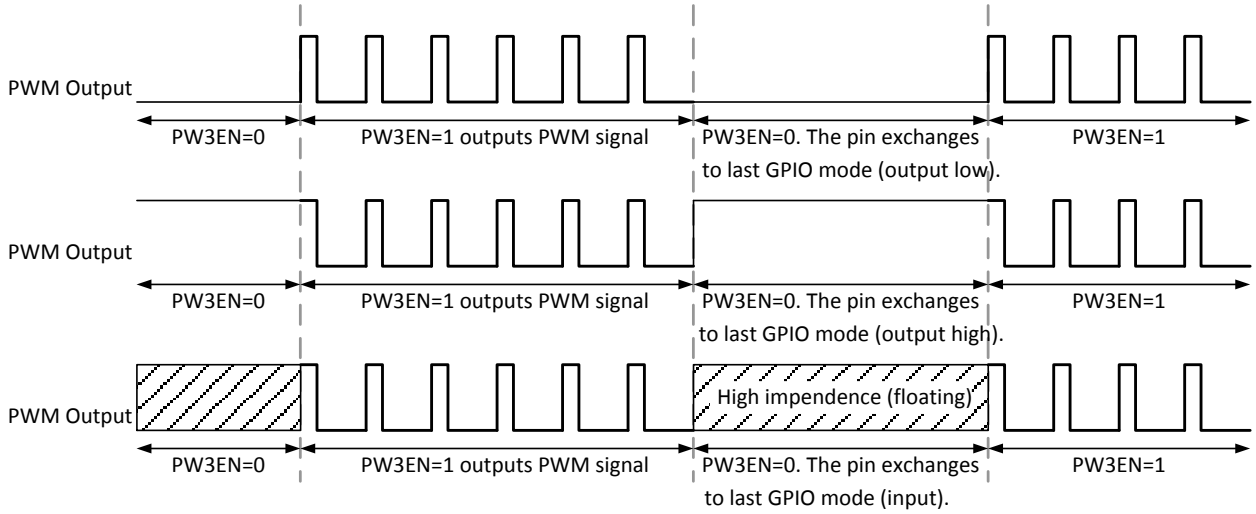
20.1 General PWM

PW3 timer builds in PWM function controlled by PW3EN register and PW3CH bits. PWM30 - PWM33 are output pins. Those output pins are shared with GPIO pin controlled by PW3CH[3:0] bits. When output PWM function, we must be set PW3EN =1. When PWM output signal synchronize finishes, the PWM channel exchanges from GPIO to PWM output. When PW3EN = 0, the PWM channel returns to GPIO mode and last status. PWM signal is generated from the result of PW3Y and PW3nD comparison combination. When PW3C starts to count or returns to 0x0000, the PWM outputs high status which is the PWM initial status. PW3C is loaded new data from PW3Y register to decide PWM cycle and resolution. PW3C keeps counting, and the system compares PW3C and PW3nD. When PW3C=PW3nD, the PWM output status exchanges to low and PW3C keeps counting. When PW3 timer overflow occurs (PW3Y-1 to 0x0000), and one cycle of PWM signal finishes. PW3C is reloaded from 0x0000 automatically, and PWM output status exchanges to high for next cycle. PW3nD decides the high duty duration, and PW3Y decides the resolution and cycle of PWM. PW3nD can't be larger than PW3Y, or the PWM signal is error. PWM clock source is fosc, PW3RATE[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 = PW3Y

PWM duty = (PW3nD): (PW3Y-PW3nD)



Copyright © 2019, SONiX Technology Co., Ltd.

20.2 PWM3 Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW3M	PW3EN	PW3R2	PW3R1	PW3R0	PW3CH3	PW3CH2	PW3CH1	PW3CH0
PW3YH	PW3Y15	PW3Y14	PW3Y13	PW3Y12	PW3Y11	PW3Y10	PW3Y9	PW3Y8
PW3YL	PW3Y7	PW3Y6	PW3Y5	PW3Y4	PW3Y3	PW3Y2	PW3Y1	PW3Y0
PW30DH	PW30D15	PW30D14	PW30D13	PW30D12	PW30D11	PW30D10	PW30D9	PW30D8
PW30DL	PW30D7	PW30D6	PW30D5	PW30D4	PW30D3	PW30D2	PW30D1	PW30D0
PW31DH	PW31D15	PW31D14	PW31D13	PW31D12	PW31D11	PW31D10	PW31D9	PW31D8
PW31DL	PW31D7	PW31D6	PW31D5	PW31D4	PW31D3	PW31D2	PW31D1	PW31D0
PW32DH	PW32D15	PW32D14	PW32D13	PW32D12	PW32D11	PW32D10	PW32D9	PW32D8
PW32DL	PW32D7	PW32D6	PW32D5	PW32D4	PW32D3	PW32D2	PW32D1	PW32D0
PW33DH	PW33D15	PW33D14	PW33D13	PW33D12	PW33D11	PW33D10	PW33D9	PW33D8
PW33DL	PW33D7	PW33D6	PW33D5	PW33D4	PW33D3	PW33D2	PW33D1	PW33D0
IEN0	EAL	EX4	EX3	EX2	ET1	EX1	ET0	EX0
IRCON	PW4F	PW3F	EPW4	EPW3	-	IE4	IE3	IE2

Copyright © 2019, SONiX Technology Co., Ltd.

PW3M Registers (PW3M: 0xF001)

Bit	Field	Type	Initial	Description
7	PW3EN	R/W	0	PW3 function 0: Disable 1: Enable*
6..4	PW3R[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	PW3CH3	R/W	0	PWM33 shared-pin control 0: GPIO 1: PWM output (shared with P1.7)
2	PW3CH2	R/W	0	PWM32 shared-pin control 0: GPIO 1: PWM output (shared with P1.6)
1	PW3CH1	R/W	0	PWM31 shared-pin control 0: GPIO 1: PWM output (shared with P1.5)
0	PW3CH0	R/W	0	PWM30 shared-pin control 0: GPIO 1: PWM output (shared with P1.4)

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

PW3YH/PW3YL Registers (PW3YH: 0xF030, PW3YL: 0xF031)

Bit	Field	Type	Initial	Description
7..0	PW3YH/L	R/W	0x00	16-bit PWM3 period control*.

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

PW30DH/PW30DL Registers (PW30DH: 0xF00C, PW30DL: 0xF00D)

Bit	Field	Type	Initial	Description
7..0	PW30DH/L	R/W	0x00	16-bit PWM3 duty control.

Copyright © 2019, SONiX Technology Co., Ltd.

PW31DH/PW31DL Registers (PW31DH: 0xF00E, PW31DL: 0xF00F)

Bit	Field	Type	Initial	Description
7..0	PW31DH/L	R/W	0x00	16-bit PWM3 duty control.

PW32DH/PW32DL Registers (PW32DH: 0xF010, PW32DL: 0xF011)

Bit	Field	Type	Initial	Description
7..0	PW32DH/L	R/W	0x00	16-bit PWM3 duty control.

PW33DH/PW33DL Registers (PW33DH: 0xF012, PW33DL: 0xF013)

Bit	Field	Type	Initial	Description
7..0	PW33DH/L	R/W	0x00	16-bit PWM3 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)

IRCON Register (0xC0)

Bit	Field	Type	Initial	Description
6	PW3F	R/W	0	PWM3 interrupt request flag. 0: None PWM3 interrupt request 1: PWM3 interrupt request.
4	EPW3	R/W	0	PWM3 interrupt control bit. 0 = Disable PWM3 interrupt function. 1 = Enable PWM3 interrupt function.
Else				Refer to other chapter(s)

Copyright © 2019, SONiX Technology Co., Ltd.

20.3 Sample Code

The following sample code demonstrates how to perform PW3 with interrupt.

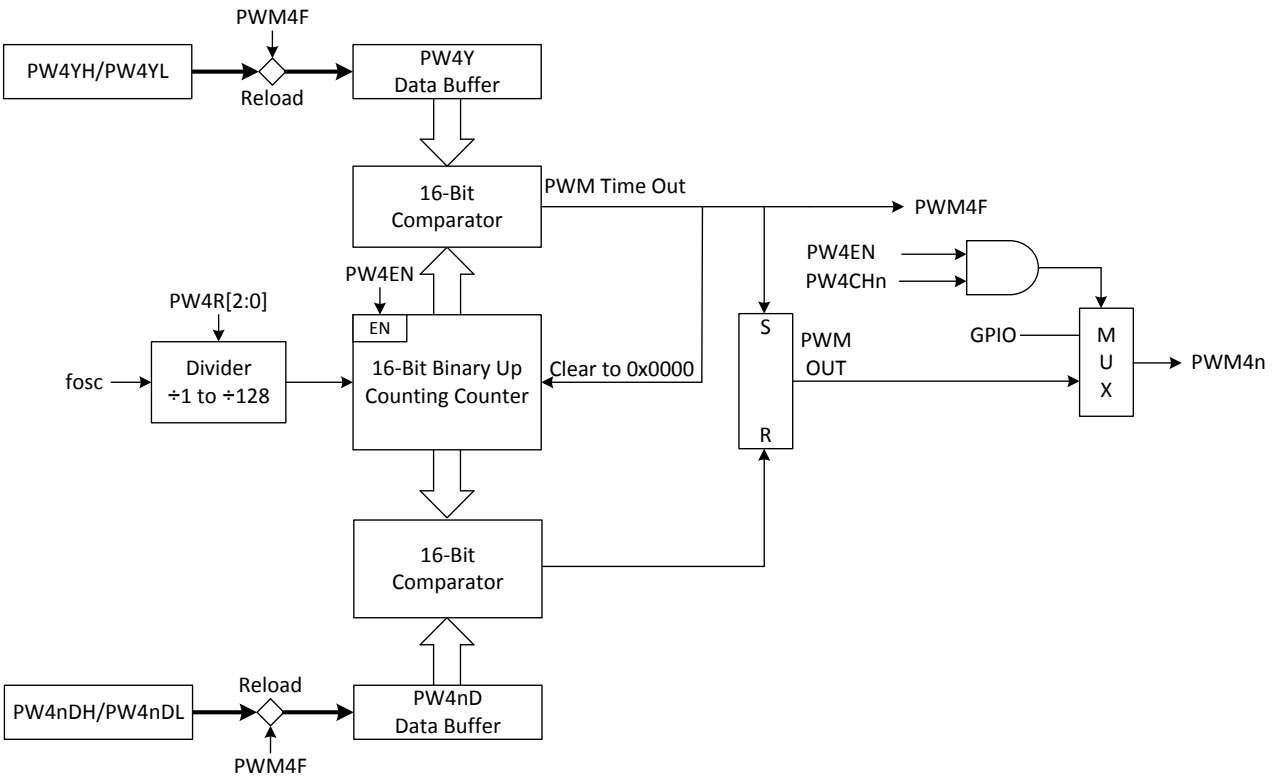
```
#define PWM30En     (1 << 0)  //Enable PWM30 output function
#define PW3En       (1 << 7)  //Enable PWM3 function

void InitPWM(void)
{
  // PWM3_Initial
  PW3YH = 0x80;
  PW3YL = 0x00;
  PW30DH = 0x40;
  PW30DL = 0x00;

  // PWM3 enable, clock = Fosc/32, PWM30 channel enable
  PW3M = PW3En | 0x20 | PWM30En;

  // Enable PWM3 interrupt & clear PWM3F
  IRCON = 0x10;

  // Enable total interrupt
  IEN0 |= 0x80;

  P0 = 0x00;
  P0M |= 0x01;
}

void PW3Interrupt(void) interrupt ISRPwm3  //0xB3
{ //PWM3F clear by hardware
  P00 = ~P00;
}
```

Copyright © 2019, SONiX Technology Co., Ltd.

21 PWM4

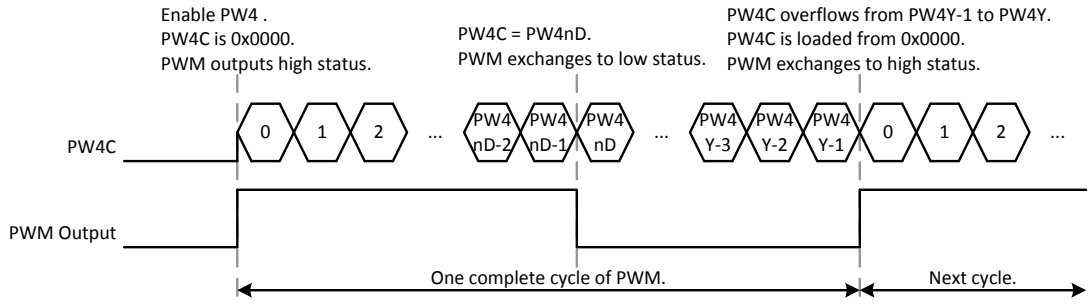
The PW4 timer is a 16-bit up counting timer and supports 4-channel general PWM function. By the counter reaches the up-boundary value (PW4Y), it clears its counter and triggers an interrupt signal. PWM's duty cycle is controlled by PW40D~PW43D register. Each PWM channel has its own duty control.

The PWM function has 4 programmable channels shared with GPIO pins and controlled by PW4CH[3:0] bits. The output operation must be through enabled each bit/channel of PW4CH[3:0] bits. The enabled PWM channel exchanges from GPIO to PWM output. When the PW4CH[3:0] bits disables, the PWM channel returns to last status of GPIO mode. The PW4 timer builds in IDLE Mode wake-up function if interrupt enable. When timer overflow occurs (counts from PW4Y-1 to PW4Y), PW4F would be issued immediately which can read/write by firmware. PW4 interrupt function is controlled by EPW4.



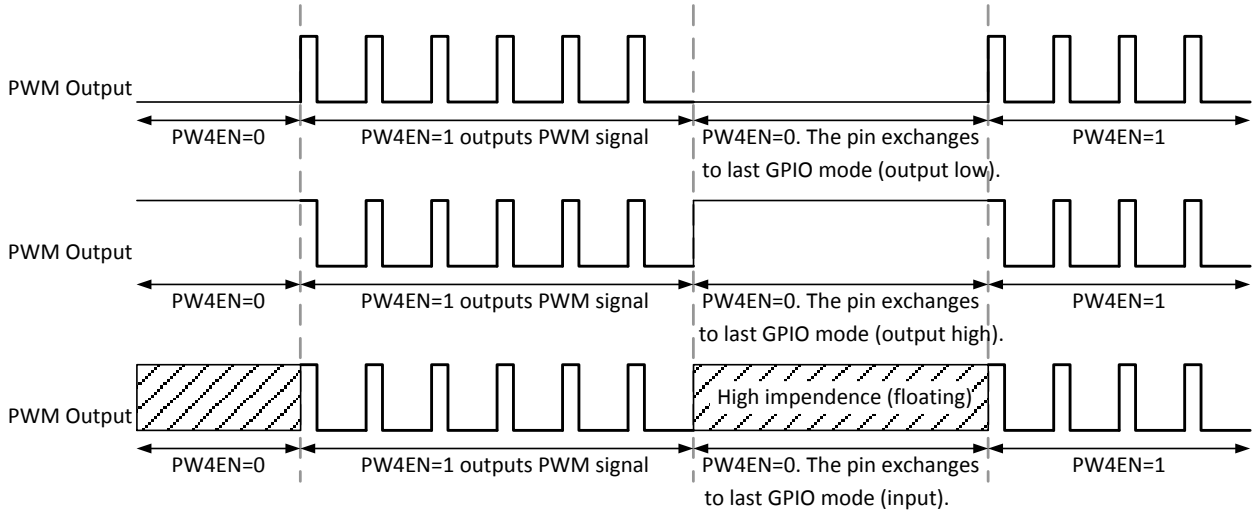
21.1 General PWM

PW4 timer builds in PWM function controlled by PW4EN register and PW4CH bits. PWM40 - PWM43 are output pins. Those output pins are shared with GPIO pin controlled by PW4CH[3:0] bits. When output PWM function, we must be set PW4EN =1. When PWM output signal synchronize finishes, the PWM channel exchanges from GPIO to PWM output. When PW4EN = 0, the PWM channel returns to GPIO mode and last status. PWM signal is generated from the result of PW4Y and PW4nD comparison combination. When PW4C starts to count or returns to 0x0000, the PWM outputs high status which is the PWM initial status. PW4C is loaded new data from PW4Y register to decide PWM cycle and resolution. PW4C keeps counting, and the system compares PW4C and PW4nD. When PW4C=PW4nD, the PWM output status exchanges to low and PW4C keeps counting. When PW4 timer overflow occurs (PW4Y-1 to 0x0000), and one cycle of PWM signal finishes. PW4C is reloaded from 0x0000 automatically, and PWM output status exchanges to high for next cycle. PW4nD decides the high duty duration, and PW4Y decides the resolution and cycle of PWM. PW4nD can't be larger than PW4Y, or the PWM signal is error. PWM clock source is fosc, PW4RATE[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 = PW4Y

PWM duty = (PW4nD): (PW4Y-PW4nD)



Copyright © 2019, SONiX Technology Co., Ltd.

21.2 PWM4 Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW4M	PW4EN	PW4R2	PW4R1	PW4R0	PW4CH3	PW4CH2	PW4CH1	PW4CH0
PW4YH	PW4Y15	PW4Y14	PW4Y13	PW4Y12	PW4Y11	PW4Y10	PW4Y9	PW4Y8
PW4YL	PW4Y7	PW4Y6	PW4Y5	PW4Y4	PW4Y3	PW4Y2	PW4Y1	PW4Y0
PW40DH	PW40D15	PW40D14	PW40D13	PW40D12	PW40D11	PW40D10	PW40D9	PW40D8
PW40DL	PW40D7	PW40D6	PW40D5	PW40D4	PW40D3	PW40D2	PW40D1	PW40D0
PW41DH	PW41D15	PW41D14	PW41D13	PW41D12	PW41D11	PW41D10	PW41D9	PW41D8
PW41DL	PW41D7	PW41D6	PW41D5	PW41D4	PW41D3	PW41D2	PW41D1	PW41D0
PW42DH	PW42D15	PW42D14	PW42D13	PW42D12	PW42D11	PW42D10	PW42D9	PW42D8
PW42DL	PW42D7	PW42D6	PW42D5	PW42D4	PW42D3	PW42D2	PW42D1	PW42D0
PW43DH	PW43D15	PW43D14	PW43D13	PW43D12	PW43D11	PW43D10	PW43D9	PW43D8
PW43DL	PW43D7	PW43D6	PW43D5	PW43D4	PW43D3	PW43D2	PW43D1	PW43D0
IEN0	EAL	EX4	EX3	EX2	ET1	EX1	ET0	EX0
IRCON	PW4F	PW3F	EPW4	EPW3	-	IE4	IE3	IE2

Copyright © 2019, SONiX Technology Co., Ltd.

PW4M Registers (PW4M: 0xF015)

Bit	Field	Type	Initial	Description
7	PW4EN	R/W	0	PW4 function 0: Disable 1: Enable*
6..4	PW4R[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	PW4CH3	R/W	0	PWM43 shared-pin control 0: GPIO 1: PWM output (shared with P3.7)
2	PW4CH2	R/W	0	PWM42 shared-pin control 0: GPIO 1: PWM output (shared with P3.6)
1	PW4CH1	R/W	0	PWM41 shared-pin control 0: GPIO 1: PWM output (shared with P3.5)
0	PW4CH0	R/W	0	PWM40 shared-pin control 0: GPIO 1: PWM output (shared with P3.4)

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

PW4YH/PW4YL Registers (PW4YH: 0xF032, PW4YL: 0xF033)

Bit	Field	Type	Initial	Description
7..0	PW4YH/L	R/W	0x00	16-bit PWM4 period control*.

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

PW40DH/PW40DL Registers (PW40DH: 0xF020, PW40DL: 0xF021)

Bit	Field	Type	Initial	Description
7..0	PW40DH/L	R/W	0x00	16-bit PWM4 duty control.

Copyright © 2019, SONiX Technology Co., Ltd.

PW41DH/PW41DL Registers (PW41DH: 0xF022, PW41DL: 0xF023)

Bit	Field	Type	Initial	Description
7..0	PW41DH/L	R/W	0x00	16-bit PWM4 duty control.

PW42DH/PW42DL Registers (PW42DH: 0xF024, PW42DL: 0xF025)

Bit	Field	Type	Initial	Description
7..0	PW42DH/L	R/W	0x00	16-bit PWM4 duty control.

PW43DH/PW43DL Registers (PW43DH: 0xF026, PW43DL: 0xF027)

Bit	Field	Type	Initial	Description
7..0	PW43DH/L	R/W	0x00	16-bit PWM4 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)

IRCON Register (0xC0)

Bit	Field	Type	Initial	Description
7	PW4F	R/W	0	PWM4 interrupt request flag. 0: None PWM4 interrupt request 1: PWM4 interrupt request.
5	EPW4	R/W	0	PWM4 interrupt control bit. 0 = Disable PWM4 interrupt function. 1 = Enable PWM4 interrupt function.
Else				Refer to other chapter(s)

Copyright © 2019, SONiX Technology Co., Ltd.

21.3 Sample Code

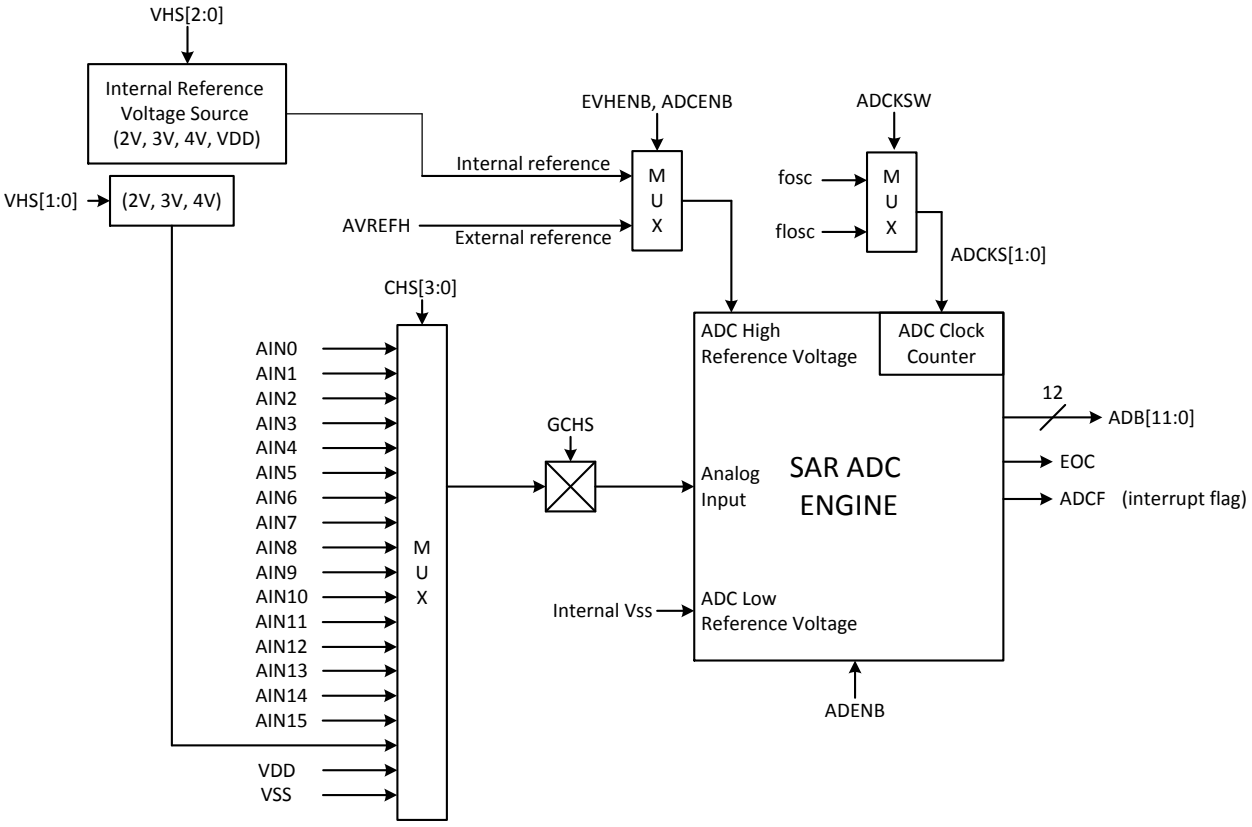
The following sample code demonstrates how to perform PW4 with interrupt.

```
#define PWM40En     (1 << 0)  //Enable PWM40 output function
#define PW4En        (1 << 7)  //Enable PWM4 function

void InitPWM(void)
{
  // PWM4_Initial
  PW4YH = 0x80;
  PW4YL = 0x00;
  PW40DH = 0x40;
  PW40DL = 0x00;

  // PWM4 enable, clock = Fosc/32, PWM40 channel enable
  PW4M = PW4En | 0x20 | PWM40En;

  // Enable PWM4 interrupt & clear PWM4F
  IRCON = 0x20;

  // Enable total interrupt
  IEN0 |= 0x80;

  P0 = 0x00;
  P0M |= 0x01;
}

void PW4Interrupt(void) interrupt ISRPwm4  //0xA3
{ //PWM4F clear by hardware
  P00 = ~P00;
}
```

Copyright © 2019, SONiX Technology Co., Ltd.

22 ADC

The analog to digital converter (ADC) is SAR structure with 16-input sources and up to 4096-step resolution to transfer analog signal into 12-bits digital buffers. The ADC builds in 16-channel input source to measure 16 different analog signal sources. The ADC resolution is 12-bit. The ADC has four clock rates to decide ADC converting rate. The ADC reference high voltage includes 5 sources. Four internal power source including VDD, 4V, 3V and 2V. The other one is external reference voltage input pin from AVREFH pin. The ADC builds in P4CON/P5CON 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.



22.1 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.

22.2 ADC input channel

The ADC builds in 16-channel input source (AIN0 – AIN15) to measure 16 different analog signal sources controlled by CHS[4:0] and GCHS bits. The AIN16 is internal 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. AIN16 can be a good battery detector for battery system. To select appropriate internal AVREFH level and compare value, a high performance and cheaper low battery detector is built in the system. AIN17 is VDD channel. AIN18 is VSS channel. Other channel is reserved.

Copyright © 2019, SONiX Technology Co., Ltd.

CHS[4:0]	Channel	Pin name	Remark
00000	AIN0	P4.0	-
00001	AIN1	P4.1	-
00010	AIN2	P4.2	-
00011	AIN3	P4.3	-
00100	AIN4	P4.4	-
00101	AIN5	P4.5	-
00110	AIN6	P4.6	-
00111	AIN7	P4.7	-
01000	AIN8	P5.0	-
01001	AIN9	P5.1	-
01010	AIN10	P5.2	-
01011	AIN11	P5.3	-
01100	AIN12	P5.4	-
01101	AIN13	P5.5	-
01110	AIN14	P5.6	-
01111	AIN15	P5.7	-
10000	AIN16	Internal 2V or 3V or 4V	Battery detector channel
10001	AIN17	VDD	-
10010	AIN18	VSS	-
Other	-	-	Reserved

22.2.1 Pin Configuration

ADC input channels are shared with Port4 and Port5. ADC channel selection is through CHS[4:0] bit. Only one pin of Port4 and Port5 can be configured as ADC input in the same time. The pins of Port4 and Port5 configured as ADC input channel must be set input mode, disable internal pull-up and enable P4CON/P5CON 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 Port4, Port5 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 P4CON and P5CON must be set to “0”, or the digital I/O signal would be isolated.

Copyright © 2019, SONiX Technology Co., Ltd.

22.3 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/P4.0. In the condition, P4.0 GPIO mode must be set as input mode and disable internal pull-up resistor.

If EVHENB bit is “0”, ADC reference high voltage is from internal voltage source selected by VHS[1:0] bits. If VHS[1:0] is “11”, ADC reference high voltage is VDD. If VHS[1:0] is “10”, ADC reference high voltage is 4V. If VHS[1:0] is “01”, ADC reference high voltage is 3V. If VHS[1:0] is “00”, ADC reference high 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 reference high voltage must be internal VDD or External voltage, not internal 2V/3V/4V.

22.3.1 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 P4.0/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

22.4 Converting Time

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

Copyright © 2019, SONiX Technology Co., Ltd.

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/32	1/(16MHz/32/4)*16 = 128us	7.8125kHz	1/(32MHz/32/4)*16 = 64us	15.625kHz
01	fosc/16	1/(16MHz/16/4)*16 = 64us	15.625kHz	1/(32MHz/16/4)*16 = 32us	31.25kHz
10	fosc/2	1/(16MHz/2/4)*16 = 8us	125kHz	1/(32MHz/2/4)*16 = 4us	250kHz
11	fosc/8	1/(16MHz/8/4)*16 = 32us	31.25kHz	1/(32MHz/8/4)*16 = 16us	62.5kHz

When ACKSW=1:

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

Copyright © 2019, SONiX Technology Co., Ltd.

22.5 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 22-1 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

22.6 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
P4CON	P4CON7	P4CON6	P4CON5	P4CON4	P4CON3	P4CON2	P4CON1	P4CON0
P5CON	P5CON7	P5CON6	P5CON5	P5CON4	P5CON3	P5CON2	P5CON1	P5CON0
IEN0	EAL	EX4	EX3	EX2	ET1	EX1	ET0	EX0
IEN2	ET4	ELCD	ET3	ET2	EPW2	EPW1	-	EADC
IRCON2	TF4	LCDF	TF3	TF2	PW2F	PW1F	-	ADCF

Copyright © 2019, SONiX Technology Co., Ltd.

ADM Register (0xD2)

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

*(1) The AIN16 is internal 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.

Copyright © 2019, SONiX Technology Co., Ltd.

ADR Register (0xD4)

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

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

VREFH Register (0xD5)

Bit	Field	Type	Initial	Description
7	EVHENB	R/W	0	ADC internal reference high voltage control bit. 0: Enable ADC internal VREFH function. AVREFH/P4.0 pin is GPIO. 1: Disable ADC internal VREFH function. AVREFH/P4.0 pin is external AVREFH*(1) input pin.
2..0	VHS[2:0]	R/W	00	ADC internal reference high voltage selects bits. *(2) 000: VREFH = 2.0V. 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 AIN16 channel is selected as internal 2V or 3V or 4V input channel. There is no any input pin from outside. In this time ADC reference high voltage must be internal VDD or External voltage, not internal 2V/3V/4V.

Copyright © 2019, SONiX Technology Co., Ltd.

P4CON Register (0x9F)

Bit	Field	Type	Initial	Description
7..0	P4CON[7:0]	R/W	0x00	P4 configuration control bit*. 0: P4 can be analog input pin (ADC input pin) or digital GPIO pin. 1: P4 is pure analog input pin and can't be a digital GPIO pin.

* P4CON [7:0] will configure related Port4 pin as pure analog input pin to avoid current leakage.

P5CON Register (0xCF)

Bit	Field	Type	Initial	Description
7..0	P5CON[7:0]	R/W	0x00	P5 configuration control bit*. 0: P5 can be analog input pin (ADC input pin) or digital GPIO pin. 1: P5 is pure analog input pin and can't be a digital GPIO pin.

* P5CON [7:0] will configure related Port5 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)

Copyright © 2019, SONiX Technology Co., Ltd.

22.7 Sample Code

The following sample code demonstrates how to perform ADC to convert AIN5 with interrupt.

```
#define ADCAIN16_VDD   (3 << 0)  //AIN16 = VDD
#define ADCAIN16_4V    (2 << 0)  //AIN16 = 4.0V
#define ADCAIN16_3V    (1 << 0)  //AIN16 = 3.0V
#define ADCAIN16_2V    (0 << 0)  //AIN16 = 2.0V
#define ADCInRefVDD    (1 << 2)  //internal reference from VDD
#define ADCExHighRef   (1 << 7)  //high reference from AVREFH/P4.0
#define ADCClkFosc     (0 << 4)  //ADC clock source = fosc
#define ADCClkFlosc    (1 << 4)  //ADC clock source = flosc
#define ADCSpeedDiv32  (0 << 4)  //ADC clock = fosc/16 or flosc/32
#define ADCSpeedDiv16  (1 << 4)  //ADC clock = fosc/8 or flosc/16
#define ADCSpeedDiv2   (2 << 4)  //ADC clock = fosc/1 or flosc/2
#define ADCSpeedDiv8   (3 << 4)  //ADC clock = fosc/2 or flosc/8
#define ADCChannelEn   (1 << 6)  //enable ADC channel
#define SelAIN5        (5 << 0)  //select ADC channel 5
#define ADCStart       (1 << 6)  //start ADC conversion
#define ADCEn          (1 << 7)  //enable ADC
#define EADC           (1 << 0)  //enable ADC interrupt
#define ClearEOC       0xDF;

unsigned int  ADCBuffer;    // data buffer

void ADCInit(void)
{
  P1  = 0x00;
  P1M = 0x80;
  // set AIN5 pin's mode at pure analog pin
  P4CON |= 0x20;     //AIN5/P15
  P4M   &= 0XDF;    //input mode
  P4UR  &= 0XDF;    //disable pull-high

  // configure ADC channel, ADC clock source and enable ADC.
  ADM = ADCEn | SelAIN5 | ADCClkFosc;
  // enable channel and select conversion speed
  ADR = ADCChannelEn | ADCSpeedDiv2;
  // configure reference voltage
  VREFH = ADCInRefVDD;

  // enable ADC interrupt
  IEN2  |= EADC;
  IEN0  |= 0x80;    //enable global interrupt

  // start ADC conversion
  ADM   |= ADCStart;
}

void ADCInterrupt(void) interrupt ISRAdc  //0x1B
{
  //ADCF clear by hardware
  P17 = ~P17;
  ADCBuffer = (ADB << 4) + (ADR & 0x0F);
  ADM &= ClearEOC;
  ADM |= ADCStart;
}
```

Copyright © 2019, SONiX Technology Co., Ltd.

23 UART0

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

The UART features include the following:

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

23.1 UART Operation

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

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

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

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

Copyright © 2019, SONiX Technology Co., Ltd.

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

23.2 Mode 0: Synchronous 8-bit Receiver/Transmitter

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

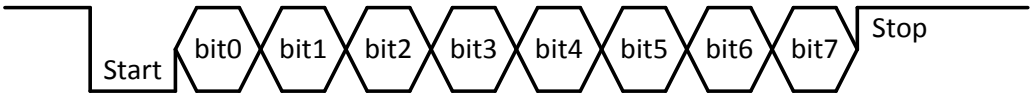
Copyright © 2019, SONiX Technology Co., Ltd.

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

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

Data transmission is controlled by REN0 bit. After transmission configuration, load transmitted data into S0BUF, and then UART0 starts to transmit the pocket. The TI0 flag is set at the beginning of the stop bit.

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



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

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

Data transmission is controlled by REN0 bit. After transmission configuration, load transmitted data into S0BUF, and then UART0 starts to transmit the pocket. The 9th data bit is taken from TB80. The TI0 flag is set at the beginning of the stop bit.

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

Copyright © 2019, SONiX Technology Co., Ltd.

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

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

Data transmission is controlled by 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, URX0 detects the falling edge of start bit, and then the RI0 flag is set in the middle of a stop bit. Until reception completion, lower 8-bit input data is stored in S0BUF register and the 9th bit is stored in RB80.

23.6 Multiprocessor Communication

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

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

Copyright © 2019, SONiX Technology Co., Ltd.

23.7 Baud Rate Control

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

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

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

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

Table 23-1 Recommended Setting for Common UART0 Baud Rates (fcpu = 16 MHz)

Baud Rate	SMOD0	S2RELH	S2RELL	Accuracy
4800	0	0x03	0xCC	0.16 %
9600	0	0x03	0xE6	0.16 %
19200	0	0x03	0xF3	0.16 %
38400	1	0x03	0xF3	0.16 %
56000	1	0x03	0xF7	-0.79 %
57600	1	0x03	0xF7	-3.55 %
115200	1	0x03	0xFC	8.51 %
128000	1	0x03	0xFC	-2.34 %
250000	1	0x03	0xFE	0 %
500000	1	0x03	0xFF	0 %

Copyright © 2019, SONiX Technology Co., Ltd.

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

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

Table 23-2 Recommended Setting T1 overflow period (T1 clock=32M) for Common UART0 Baud Rates (fcpu = 16 MHz)

Baud Rate	SMOD0	Timer Period	TH1/TL1	Accuracy
4800	0	6.510 us	0x30	0.16 %
9600	1	6.510 us	0x30	0.16 %
19200	1	3.255 us	0x98	0.16 %
38400	1	1.628 us	0xCC	0.16 %
56000	1	1.116 us	0xDC	-0.80 %
57600	1	1.085 us	0xDD	-0.80 %
115200	1	0.543 us	0xEF	2.08 %
128000	1	0.488 us	0xF0	-2.40 %
250000	1	0.250 us	0xF8	0 %

✷ ***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.

23.8 Power Saving

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

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

Copyright © 2019, SONiX Technology Co., Ltd.

23.9 UART Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
S0CON	S0M0	S0M1	S0M20	REN0	TB80	RB80	TI0	RI0
S0CON2	BD2	BD1	BD0	-	-	-	-	-
S0BUF	S0BUF7	S0BUF6	S0BUF5	S0BUF4	S0BUF3	S0BUF2	S0BUF1	S0BUF0
PCON	SMOD2	SMOD1	SMOD0	-	P2SEL	GF0	STOP	IDLE
S0RELH	-	-	-	-	-	-	S0REL9	S0REL8
S0RELL	S0REL7	S0REL6	S0REL5	S0REL4	S0REL3	S0REL2	S0REL1	S0REL0
IEN0	EAL	EX4	EX3	EX2	ET1	EX1	ET0	EX0
IEN1	EU2RX	EU2TX	EU1RX	EU1TX	EU0RX	EU0TX	-	EI2C
P0OC	-	-	P14OC	P13OC	P06OC	P05OC	P03OC	P02OC
P0M	P07M	P06M	P05M	P04M	P03M	P02M	P01M	P00M
P0	P07	P06	P05	P04	P03	P02	P01	P00

S0CON Register (0x98)

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

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

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

S0CON2 Register (0x9B)

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

S0BUF Register (0x99)

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

Copyright © 2019, SONiX Technology Co., Ltd.

PCON Register (0x87)

Bit	Field	Type	Initial	Description
5	SMOD0	R/W	0	UART0 baud rate control. In UART mode 0: Unused. In UART mode 1, 3: The baud rate generated as the equation in section 23.7 (Baud Rate Control). In UART mode 2: 0: fcpu/64 1: fcpu/32
6..0				Refer to other chapter(s)

S0RELH/S0RELL Registers (S0RELH: 0x9D, S0RELL: 0x9C)

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

IEN0 Register (0xA8)

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

IEN1 Register (0xB8)

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

P0OC Register (0xD1)

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

Copyright © 2019, SONiX Technology Co., Ltd.

P0M Register (0xF9)

Bit	Field	Type	Initial	Description
3	P03M	R/W	0	0: Set P0.3 (URX0) as input mode (required) ~~1: Set P0.3 (URX0) as output mode~~*
2	P02M	R/W	0	~~0: Set P0.2 (UTX0) as input mode~~* 1: Set P0.2 (UTX0) as output mode (required)
Else				Refer to other chapter(s)

* The URX0 and UTX0 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	This bit is available to read at any time for monitoring the bus statue.
2	P02	R/W	1	~~0: Set P0.2 (UTX) always low~~* 1: Make P0.2 (UTX) can output UART data (required)
Else				Refer to other chapter(s)

* Setting P02 initially high because UART block drive the shared pin low signal only.

Copyright © 2019, SONiX Technology Co., Ltd.

23.10 Sample Code

The following sample code demonstrates how to perform UART0 mode 1 with interrupt.

```
#define SYSUart0SM0  (0 << 6)
#define SYSUart0SM1  (1 << 6)
#define SYSUart0SM2  (2 << 6)
#define SYSUart0SM3  (3 << 6)
#define SYSUart0REN  (1 << 4)
#define SYSUart0SMOD (1 << 5)
#define SYSUart0BD   (1 << 5)
#define SYSUart0EU0RX (1 << 3)
#define SYSUart0EU0TX (1 << 2)

void SYSUart0Init(void)
{
  // set UTX0, URX0 pins' mode at here or at GPIO initialization
  P02 = 1;
  P0M = P0M | 0x04 & ~0x08;
  // configure UART0 mode between SM0 and SM3, enable URX0
  S0CON = SYSUart0SM1 | SYSUart0REN;
  // configure UART0 baud rate
  PCON = SYSUart0SMOD;
  S0CON2 = SYSUart0BD;
  S0RELH = 0x03;
  S0RELL = 0xFE;

  // enable UART0 TX/RX interrupt
  IEN1 |= SYSUartEU0TX | SYSUart0EU0RX;
  IEN0 |= 0x80;    //enable global interrupt
  // send first UTX0 data
  S0BUF = uartTxBuf;
}

void SYSUartInterrupt(void) interrupt ISRUart0tX  //0x13
{
    S0BUF = uartTxBuf;
    S0CON &= 0xFD;  //clear TI0
}

void SYSUartInterrupt(void) interrupt ISRUart0rX  //0x53
{
    uartRxBuf = S0BUF;
    S0CON &= 0xFE;  //clear RI0
}
```

Copyright © 2019, SONiX Technology Co., Ltd.

24 UART1

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

The UART features include the following:

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

24.1 UART Operation

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

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

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

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

Copyright © 2019, SONiX Technology Co., Ltd.

S1M0	S1M1	Mode	Synchronization	Clock Rate	Start Bit	Data Bits	Stop Bit	UART pins' mode and data
0	0	0	Synchronous	Fcpu/12	X	8	X	UTX1 pin: P05M=1 and P05=1 URX1 pin: Transmitter: P06M=1 and P06=1 Receiver: P06M=0 and P06=1
0	1	1	Asynchronous	Baud rate generator or T1 overflow rate	1	8	1	UTX1 pin: P05M=1 and P05=1 URX1 pin: P06M=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	

24.2 Mode 0: Synchronous 8-bit Receiver/Transmitter

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

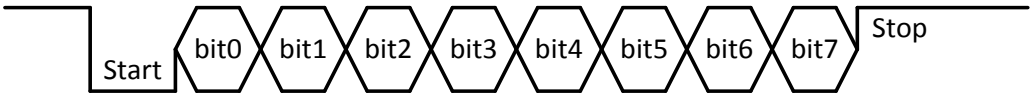
Copyright © 2019, SONiX Technology Co., Ltd.

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

Mode1 supports an asynchronous 8-bit UART with variable baud rate. The transfer format includes 1 start bit, 8 data bits (LSB first) and 1 stop bit. Data is transmitted by UTX1 pin and received by URX1 pin. The baud rate clock source can be baud rate generator or T1 overflow controlled by BD1 bit. When BD1=0, the baud rate clock source is from T1 overflow. When BD1=1, the baud rate clock source is from baud rate generator controlled by S1RELH and S1RELL. Additionally, the baud rate can be doubled by SMOD1 bit.

Data transmission is controlled by REN1 bit. After transmission configuration, load transmitted data into S1BUF, and then UART1 starts to transmit the pocket. The TI1 flag is set at the beginning of the stop bit.

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



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

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

Data transmission is controlled by REN1 bit. After transmission configuration, load transmitted data into S1BUF, and then UART1 starts to transmit the pocket. The 9th data bit is taken from TB81. The TI1 flag is set at the beginning of the stop bit.

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

Copyright © 2019, SONiX Technology Co., Ltd.

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

Mode3 supports an asynchronous 9-bit UART with variable baud rate. The transfer format includes 1 start bit, 9 data bits (LSB first) and 1 stop bit. Data is transmitted by UTX1 pin and received by URX1 pin. The different between Mode2 and Mode3 is baud rate selection. In the Mode3, the baud rate clock source can be baud rate generator or T1 overflow controlled by BD1 bit. When BD1=0, the baud rate clock source is from T1 overflow. When BD1=1, the baud rate clock source is from baud rate generator controlled by S1RELH and S1RELL. Additionally, the baud rate can be doubled by SMOD1 bit.

Data transmission is controlled by REN1 bit. After transmission configuration, load transmitted data into S1BUF, and then UART starts to transmit the pocket. The 9th data bit is taken from TB81. The TI1 flag is set at the beginning of the stop bit.

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

24.6 Multiprocessor Communication

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

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

Copyright © 2019, SONiX Technology Co., Ltd.

24.7 Baud Rate Control

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

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

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

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

Table 24-1 Recommended Setting for Common UART1 Baud Rates (fcpu = 16 MHz)

Baud Rate	SMOD1	S2RELH	S2RELL	Accuracy
4800	0	0x03	0xCC	0.16 %
9600	0	0x03	0xE6	0.16 %
19200	0	0x03	0xF3	0.16 %
38400	1	0x03	0xF3	0.16 %
56000	1	0x03	0xF7	-0.79 %
57600	1	0x03	0xF7	-3.55 %
115200	1	0x03	0xFC	8.51 %
128000	1	0x03	0xFC	-2.34 %
250000	1	0x03	0xFE	0 %
500000	1	0x03	0xFF	0 %

Copyright © 2019, SONiX Technology Co., Ltd.

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

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

Table 24-2 Recommended Setting T1 overflow period (T1 clock=32M) for Common UART1 Baud Rates (fcpu = 16 MHz)

Baud Rate	SMOD1	Timer Period	TH1/TL1	Accuracy
4800	0	6.510 us	0x30	0.16 %
9600	1	6.510 us	0x30	0.16 %
19200	1	3.255 us	0x98	0.16 %
38400	1	1.628 us	0xCC	0.16 %
56000	1	1.116 us	0xDC	-0.80 %
57600	1	1.085 us	0xDD	-0.80 %
115200	1	0.543 us	0xEF	2.08 %
128000	1	0.488 us	0xF0	-2.40 %
250000	1	0.250 us	0xF8	0 %

✷ ***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.

24.8 Power Saving

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

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

Copyright © 2019, SONiX Technology Co., Ltd.

24.9 UART Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
S1CON	S1M0	S1M1	S1M20	REN1	TB81	RB81	TI1	RI1
S0CON2	BD2	BD1	BD0	-	-	-	-	-
S1BUF	S1BUF7	S1BUF6	S1BUF5	S1BUF4	S1BUF3	S1BUF2	S1BUF1	S1BUF0
PCON	SMOD2	SMOD1	SMOD0	-	P2SEL	GF0	STOP	IDLE
S1RELH	-	-	-	-	-	-	S1REL9	S1REL8
S1RELL	S1REL7	S1REL6	S1REL5	S1REL4	S1REL3	S1REL2	S1REL1	S1REL0
IEN0	EAL	EX4	EX3	EX2	ET1	EX1	ET0	EX0
IEN1	EU2RX	EU2TX	EU1RX	EU1TX	EU0RX	EU0TX	-	EI2C
P0OC	-	-	P14OC	P13OC	P06OC	P05OC	P03OC	P02OC
P0M	P07M	P06M	P05M	P04M	P03M	P02M	P01M	P00M
P0	P07	P06	P05	P04	P03	P02	P01	P00

Copyright © 2019, SONiX Technology Co., Ltd.

S1CON Register (0xC1)

Bit	Field	Type	Initial	Description
7..6	S1M[0:1]	R/W	00	UART1 mode selection 00: Mode 0 01: Mode 1 10: Mode 2 11: Mode 3
5	S1M20	R/W	0	Multiprocessor communication (mode 2, 3) 0: Disable 1: Enable
4	REN1	R/W	0	UART1 module (and reception function) 0: Disable for power saving* 1: Enable for UART operating
3	TB81	R/W	0	The 9th bit transmission data (mode 2, 3)
2	RB81	R/W	0	The 9th bit data from reception
1	TI1	R/W	0	UART1 interrupt flag of transmission
0	RI1	R/W	0	UART1 interrupt flag of reception

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

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

S0CON2 Register (0x9B)

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

S1BUF Register (0xC2)

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

Copyright © 2019, SONiX Technology Co., Ltd.

PCON Register (0x87)

Bit	Field	Type	Initial	Description
6	SMOD1	R/W	0	UART1 baud rate control. In UART mode 0: Unused. In UART mode 1, 3: The baud rate generated as the equation in section 24.7 (Baud Rate Control). In UART mode 2: 0: fcpu/64 1: fcpu/32
6..0				Refer to other chapter(s)

S1RELH/S1RELL Registers (S1RELH: 0xC4, S1RELL: 0xC3)

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

IEN0 Register (0xA8)

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

IEN1 Register (0xB8)

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

P0OC Register (0xD1)

Bit	Field	Type	Initial	Description
3	P06OC	R/W	0	0: Switch P0.6 (URX1) to push-pull mode 1: Switch P0.6 (URX1) to open-drain mode
2	P05OC	R/W	0	0: Switch P0.5(UTX1) to push-pull mode 1: Switch P0.5 (UTX1) to open-drain mode

Copyright © 2019, SONiX Technology Co., Ltd.

P0M Register (0xF9)

Bit	Field	Type	Initial	Description
6	P06M	R/W	0	0: Set P0.6 (URX1) as input mode (required) ~~1: Set P0.6 (URX1) as output mode~~*
5	P05M	R/W	0	~~0: Set P0.5 (UTX1) as input mode~~* 1: Set P0.5 (UTX1) as output mode (required)
Else				Refer to other chapter(s)

* The URX1 and UTX1 respectively require input and output mode selection to receive/transmit data appropriately.

P0 Register (0x80)

Bit	Field	Type	Initial	Description
6	P06	R/W	1	This bit is available to read at any time for monitoring the bus statue.
5	P05	R/W	1	~~0: Set P0.5 (UTX) always low~~* 1: Make P0.5 (UTX) can output UART data (required)
Else				Refer to other chapter(s)

* Setting P05 initially high because UART block drive the shared pin low signal only.

Copyright © 2019, SONiX Technology Co., Ltd.

24.10 Sample Code

The following sample code demonstrates how to perform UART1 mode 1 with interrupt.

```
#define SYSUart1SM0  (0 << 6)
#define SYSUart1SM1  (1 << 6)
#define SYSUart1SM2  (2 << 6)
#define SYSUart1SM3  (3 << 6)
#define SYSUart1REN  (1 << 4)
#define SYSUart1SMOD (1 << 6)
#define SYSUart1BD   (1 << 6)
#define SYSUart1EU1RX (1 << 5)
#define SYSUart1EU1TX (1 << 4)

void SYSUart1Init(void)
{
  // set UTX1, URX1 pins' mode at here or at GPIO initialization
  P05 = 1;
  P0M = P0M | 0x20 & ~0x40;
  // configure UART1 mode between SM0 and SM3, enable URX1
  S1CON = SYSUart1SM1 | SYSUart1REN;
  // configure UART1 baud rate
  PCON = SYSUart1SMOD;
  S0CON2 = SYSUart1BD;
  S1RELH = 0x03;
  S1RELL = 0xFE;

  // enable UART1 TX/RX interrupt
  IEN1 |= SYSUartEU1TX | SYSUart1EU1RX;
  IEN0 |= 0x80;    //enable global interrupt
  // send first UTX1 data
  S1BUF = uartTxBuf;
}

void SYSUartInterrupt(void) interrupt ISRUart1tX  //0x0B
{
    S1BUF = uartTxBuf;
    S1CON &= 0xFD;  //clear TI1
}

void SYSUartInterrupt(void) interrupt ISRUart1rX  //0x4B
{
    uartRxBuf = S1BUF;
    S1CON &= 0xFE;  //clear RI1
}
```

Copyright © 2019, SONiX Technology Co., Ltd.

25 UART2

The UART2 provides a flexible full-duplex synchronous/asynchronous receiver/transmitter. The serial interface provides an up to 500 kHz flexible full-duplex transmission. It can operate in four modes (one synchronous and three asynchronous). Mode0 is a shift register mode and operates as synchronous transmitter/receiver. In Mode1-Mode3 the UART operates as asynchronous transmitter/receiver with 8-bit or 9-bit data. The transfer format has start bit, 8-bit/ 9-bit data and stop bit. Transmission is started by writing to the S2BUF register. After reception, input data are available after completion of the reception in the S2BUF register. TB82/RB82 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.

25.1 UART Operation

The UART2 UTX2 and URX2 pins are shared with GPIO. In synchronous mode, the UTX2/URX2 shared pins must set output high by software. In asynchronous mode (8-bit/9-bit UART), the UTX2 shared pins must set output high and URX2 set input high by software. Thus, URX2/UTX2 pins will transfers to UART purpose. When UART disables, the UART pins returns to GPIO last status.

The UTX2/URX2 pins also support open-drain structure. The open-drain option is controlled by PnOC bit. When PnOC=0, disable UTX2/URX2 open-drain structure. When PnOC=1, enable UTX2/URX2 open-drain structure. If enable open-drain structure, UTX2/URX2 pin must set high level (IO mode control will be ignored) and need external pull-up resistor.

The UART2 supports interrupt function. EU2TX and EU2RX is UART2 transfer interrupt function control bit. UART2 transmitter and receiver interrupt function is controlled by EU2TX and EU2RX respectively. EU2TX=0/EU2RX=0, disable transmitter/receiver interrupt function. EU2TX=1/ EU2RX=1, enable UART transmitter/ receiver interrupt function. When UART2 interrupt function enable, the program counter points to interrupt vector to do UART2 interrupt service routine after UART operating. TI2/RI2 is UART2 interrupt request flag, and also to be the UART operating status indicator when interrupt is disabled. TI2 and RI2 must clear by software.

UART2 provides four operating mode (one synchronous and three asynchronous) controlled by S2CON register. These modes can be support in different baud rate and communication protocols.

Copyright © 2019, SONiX Technology Co., Ltd.

S2M0	S2M1	Mode	Synchronization	Clock Rate	Start Bit	Data Bits	Stop Bit	UART pins' mode and data
0	0	0	Synchronous	Fcpu/12	X	8	X	UTX2 pin: P13M=1 and P13=1 URX2 pin: Transmitter: P14M=1 and P14=1 Receiver: P14M=0 and P14=1
0	1	1	Asynchronous	Baud rate generator or T1 overflow rate	1	8	1	UTX2 pin: P13M=1 and P13=1 URX2 pin: P14M=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	

25.2 Mode 0: Synchronous 8-bit Receiver/Transmitter

Mode0 is a shift register mode. It operates as synchronous transmitter/receiver. The UTX2 pin output shift clock for both transmit and receive condition. The URX2 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 S2BUF register. In the end of the 8th bit transmission, the TI2 flag is set. Data reception is controlled by REN2 bit and clearing RI2 bits. When REN2=1 and RI2 is from 1 to 0, data transmission starts and the RI2 flag is set at the end of the 8th bit reception.

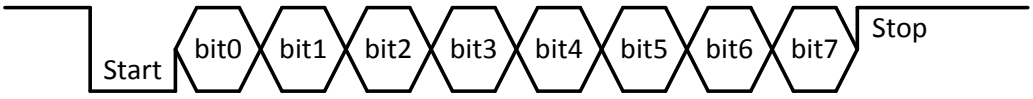
Copyright © 2019, SONiX Technology Co., Ltd.

25.3 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 UTX2 pin and received by URX2 pin. The baud rate clock source can be baud rate generator or T1 overflow controlled by BD2 bit. When BD2=0, the baud rate clock source is from T1 overflow. When BD2=1, the baud rate clock source is from baud rate generator controlled by S2RELH and S2RELL. Additionally, the baud rate can be doubled by SMOD2 bit.

Data transmission is controlled by REN2 bit. After transmission configuration, load transmitted data into S2BUF, and then UART2 starts to transmit the pocket. The TI2 flag is set at the beginning of the stop bit.

Data reception is controlled by REN2 bit. When REN2=1, data reception function is enabled. Data reception starts by receiving the start bit for master terminal, URX2 detects the falling edge of start bit, and then the RI2 flag is set in the middle of a stop bit. Until reception completion, input data is stored in S2BUF register and the stop bit is stored in RB82.



25.4 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 UTX2 pin and received by URX2 pin. The baud rate clock source is fixed to fcpu/64 or fcpu/32 and is controlled by SMOD2 bit. When SMOD2=0, baud rate is fcpu/64. When SMOD2=1, baud rate is fcpu/32.

Data transmission is controlled by REN2 bit. After transmission configuration, load transmitted data into S2BUF, and then UART2 starts to transmit the pocket. The 9th data bit is taken from TB82. The TI2 flag is set at the beginning of the stop bit.

Data reception is controlled by REN2 bit. When REN2=1, data reception function is enabled. Data reception starts by receiving the start bit for master terminal, URX2 detects the falling edge of start bit, and then the RI2 flag is set in the middle of a stop bit. Until reception completion, lower 8-bit input data is stored in S2BUF register and the 9th bit is stored in RB82.

Copyright © 2019, SONiX Technology Co., Ltd.

25.5 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 UTX2 pin and received by URX2 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 BD2 bit. When BD2=0, the baud rate clock source is from T1 overflow. When BD2=1, the baud rate clock source is from baud rate generator controlled by S2RELH and S2RELL. Additionally, the baud rate can be doubled by SMOD2 bit.

Data transmission is controlled by REN2 bit. After transmission configuration, load transmitted data into S2BUF, and then UART starts to transmit the pocket. The 9th data bit is taken from TB82. The TI2 flag is set at the beginning of the stop bit.

Data reception is controlled by REN2 bit. When REN2=1, data reception function is enabled. Data reception starts by receiving the start bit for master terminal, URX2 detects the falling edge of start bit, and then the RI2 flag is set in the middle of a stop bit. Until reception completion, lower 8-bit input data is stored in S2BUF register and the 9th bit is stored in RB82.

25.6 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 S2M20 bit. When S2M20=0, disable multiprocessor communication. When S2M20=1, enable multiprocessor communication. If S2M20 is set, the UART2 reception interrupt is only generated when the 9th received bit is “1” (RB82). The slavers will compare received data with its own address data by software. If address byte is match, the slavers clear S2M20 bit to enable interrupt function in the following data transmission. The slavers with unmatched address, their S2M20 keep in “1” and will not generate interrupt in the following data transmission.

Copyright © 2019, SONiX Technology Co., Ltd.

25.7 Baud Rate Control

The UART2 mode 0 has a fixed baud rate at fcpu/12, and the mode 2 has two baud rate selection which is chosen by SMOD2 bit: fcpu/64 (SMOD2 = 0) and fcpu/32 (SMOD2 = 1).

The baud rate of UART2 mode 1 and mode 3 is generated by either S2RELH/S2RELL registers (BD2 = 1) or Timer 1 overflow period (BD2 = 0). The SMOD2 bit doubles the frequency from the generator.

If the S2RELH/S2RELL is selected (BD2 = 1) in mode 1 and 3, the baud rate is generated as following equation.

$$\text{Baud Rate} = 2^{\text{SMOD2}} \times \frac{\text{fcpu}}{64 \times (1024 - \text{S2REL})} bps$$

Table 25-1 Recommended Setting for Common UART2 Baud Rates (fcpu = 16 MHz)

Baud Rate	SMOD2	S2RELH	S2RELL	Accuracy
4800	0	0x03	0xCC	0.16 %
9600	0	0x03	0xE6	0.16 %
19200	0	0x03	0xF3	0.16 %
38400	1	0x03	0xF3	0.16 %
56000	1	0x03	0xF7	-0.79 %
57600	1	0x03	0xF7	-3.55 %
115200	1	0x03	0xFC	8.51 %
128000	1	0x03	0xFC	-2.34 %
250000	1	0x03	0xFE	0 %
500000	1	0x03	0xFF	0 %

Copyright © 2019, SONiX Technology Co., Ltd.

If the Timer 1 overflow period is selected (BD2 = 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.

$$\text{Baud Rate} = 2^{\text{SMOD2}} \times \frac{\text{T1 clock rate}}{32 \times (256 - \text{TH1})} bps$$

Table 25-2 Recommended Setting T1 overflow period (T1 clock=32M) for Common UART2 Baud Rates (fcpu = 16 MHz)

Baud Rate	SMOD2	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 %
250000	1	0.250 us	0xF8	0 %

✷ ***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.

25.8 Power Saving

The UART2 module has clock gating function for saving power. When REN2 bit is 0, the UART2 module internal clocks are halted to reduce power consumption. UART2 relevant register (S2CON, S0CON2, S2BUF, S2RELL, S2RELH and SMOD2 bit) are unable to access.

Conversely, when REN2 bit is 1, UART2 internal clocks are run, and registers can access. The REN2 bit must be set to 1, before the initial setting UART.

Copyright © 2019, SONiX Technology Co., Ltd.

25.9 UART Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
S2CON	S2M0	S2M1	S2M20	REN2	TB82	RB82	TI2	RI2
S0CON2	BD2	BD1	BD0	-	-	-	-	-
S2BUF	S2BUF7	S2BUF6	S2BUF5	S2BUF4	S2BUF3	S2BUF2	S2BUF1	S2BUF0
PCON	SMOD2	SMOD1	SMOD0	-	P2SEL	GF0	STOP	IDLE
S2RELH	-	-	-	-	-	-	S2REL9	S2REL8
S2RELL	S2REL7	S2REL6	S2REL5	S2REL4	S2REL3	S2REL2	S2REL1	S2REL0
IEN0	EAL	EX4	EX3	EX2	ET1	EX1	ET0	EX0
IEN1	EU2RX	EU2TX	EU1RX	EU1TX	EU0RX	EU0TX	-	EI2C
P0OC	-	-	P14OC	P13OC	P06OC	P05OC	P03OC	P02OC
P1M	P17M	P16M	P15M	P14M	P13M	P12M	P11M	P10M
P1	P17	P16	P15	P14	P13	P12	P11	P10

Copyright © 2019, SONiX Technology Co., Ltd.

S2CON Register (0xC9)

Bit	Field	Type	Initial	Description
7..6	S2M[0:1]	R/W	00	UART2 mode selection 00: Mode 0 01: Mode 1 10: Mode 2 11: Mode 3
5	S2M20	R/W	0	Multiprocessor communication (mode 2, 3) 0: Disable 1: Enable
4	REN2	R/W	0	UART2 module (and reception function) 0: Disable for power saving* 1: Enable for UART operating
3	TB82	R/W	0	The 9th bit transmission data (mode 2, 3)
2	RB82	R/W	0	The 9th bit data from reception
1	TI2	R/W	0	UART2 interrupt flag of transmission
0	RI2	R/W	0	UART2 interrupt flag of reception

* When REN2 bit is 0, UART2 relevant register are unable to access, and the module internal clocks are halted.

✸ ***Note: TI2 and RI2 are clear by software when interrupt is enabled.***

S0CON2 Register (0x9B)

Bit	Field	Type	Initial	Description
7	BD2	R/W	0	Baud rate generators selection (mode 1, 3) 0: Timer 1 overflow period 1: Controlled by S2RELH, S2RELL registers
Else				Refer to other chapter(s)

S2BUF Register (0xCA)

Bit	Field	Type	Initial	Description
7..0	S2BUF	R/W	0x00	Action of writing data triggers UART2 communication (LSB first). Reception data is available to read by the end of packages.

Copyright © 2019, SONiX Technology Co., Ltd.

PCON Register (0x87)

Bit	Field	Type	Initial	Description
7	SMOD2	R/W	0	UART2 baud rate control. In UART mode 0: Unused. In UART mode 1, 3: The baud rate generated as the equation in section 25.7 (Baud Rate Control). In UART mode 2: 0: fcpu/64 1: fcpu/32
6..0				Refer to other chapter(s)

S2RELH/S2RELL Registers (S2RELH: 0xCC, S2RELL: 0xCB)

Bit	Field	Type	Initial	Description
15..10	Reserved	R	0x00	
9..0	S2REL[9:0]	R/W	0x00	S2RELH[1:0] & S2RELL[7:0]. UART2 Reload Register is used for UART2 baud rate generation.

IEN0 Register (0xA8)

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

IEN1 Register (0xB8)

Bit	Field	Type	Initial	Description
7	EU2RX	R/W	0	UART2 receiver interrupt enable
6	EU2TX	R/W	0	UART2 transmitter interrupt enable
Else				Refer to other chapter(s)

P0OC Register (0xD1)

Bit	Field	Type	Initial	Description
5	P14OC	R/W	0	0: Switch P1.4 (URX2) to push-pull mode 1: Switch P1.4 (URX2) to open-drain mode
4	P13OC	R/W	0	0: Switch P1.3(UTX2) to push-pull mode 1: Switch P1.3 (UTX2) to open-drain mode

Copyright © 2019, SONiX Technology Co., Ltd.

P1M Register (0xFA)

Bit	Field	Type	Initial	Description
4	P14M	R/W	0	0: Set P1.4 (URX2) as input mode (required) ~~1: Set P1.4 (URX2) as output mode~~*
3	P13M	R/W	0	~~0: Set P1.3 (UTX2) as input mode~~* 1: Set P1.3 (UTX2) as output mode (required)
Else				Refer to other chapter(s)

* The URX2 and UTX2 respectively require input and output mode selection to receive/transmit data appropriately.

P1 Register (0x90)

Bit	Field	Type	Initial	Description
4	P14	R/W	1	This bit is available to read at any time for monitoring the bus statue.
3	P13	R/W	1	~~0: Set P1.3 (UTX) always low~~* 1: Make P1.3 (UTX) can output UART data (required)
Else				Refer to other chapter(s)

* Setting P13 initially high because UART block drive the shared pin low signal only.

Copyright © 2019, SONiX Technology Co., Ltd.

25.10 Sample Code

The following sample code demonstrates how to perform UART2 mode 1 with interrupt.

```
#define SYSUart2SM0  (0 << 6)
#define SYSUart2SM1  (1 << 6)
#define SYSUart2SM2  (2 << 6)
#define SYSUart2SM3  (3 << 6)
#define SYSUart2REN  (1 << 4)
#define SYSUart2SMOD (1 << 7)
#define SYSUart2BD   (1 << 7)
#define SYSUart2EU2RX (1 << 7)
#define SYSUart2EU2TX (1 << 6)

void SYSUart2Init(void)
{
  // set UTX2, URX2 pins' mode at here or at GPIO initialization
  P13 = 1;
  P0M = P0M | 0x08 & ~0x10;
  // configure UART2 mode between SM0 and SM3, enable URX2
  S2CON = SYSUart2SM1 | SYSUart2REN;
  // configure UART2 baud rate
  PCON = SYSUart2SMOD;
  S0CON2 = SYSUart2BD;
  S2RELH = 0x03;
  S2RELL = 0xFE;

  // enable UART2 TX/RX interrupt
  IEN1 |= SYSUartEU2TX | SYSUart2EU2RX;
  IEN0 |= 0x80;    //enable global interrupt
  // send first UTX2 data
  S2BUF = uartTxBuf;
}

void SYSUartInterrupt(void) interrupt ISRUart2tX  //0x3B
{
    S2BUF = uartTxBuf;
    S2CON &= 0xFD;  //clear TI2
}

void SYSUartInterrupt(void) interrupt ISRUart2rX  //0x8B
{
    uartRxBuf = S2BUF;
    S2CON &= 0xFE;  //clear RI2
}
```

Copyright © 2019, SONiX Technology Co., Ltd.

26 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.

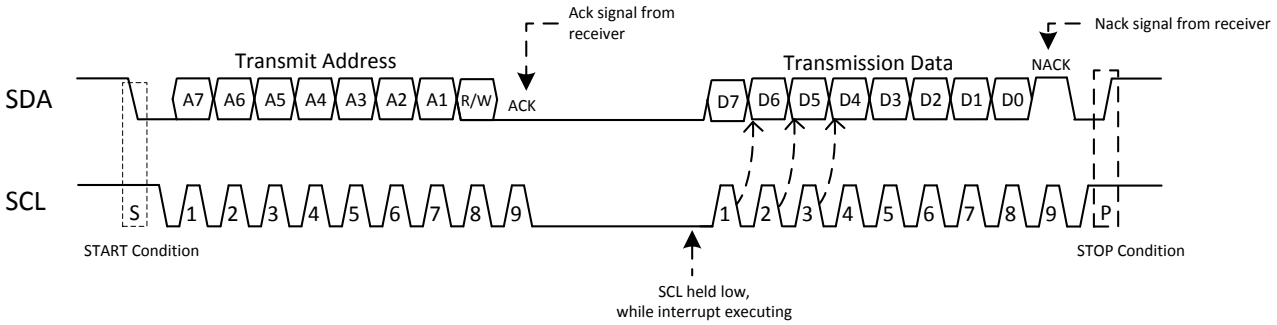
- Master Tx, Rx Mode
- Slave Tx, Rx mode (with general address call) for multiplex slave in single master situation.
- 2-wire synchronous data transfer/receiver.
- Support 100K/400K clock rate.

26.1 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.

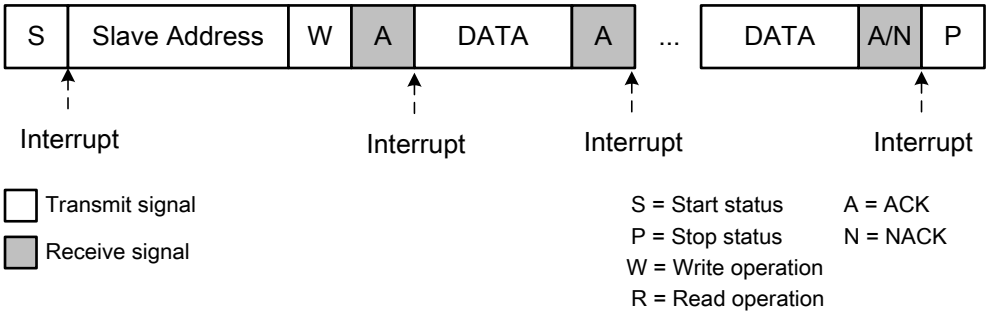


26.2 I2C Transfer Modes

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.

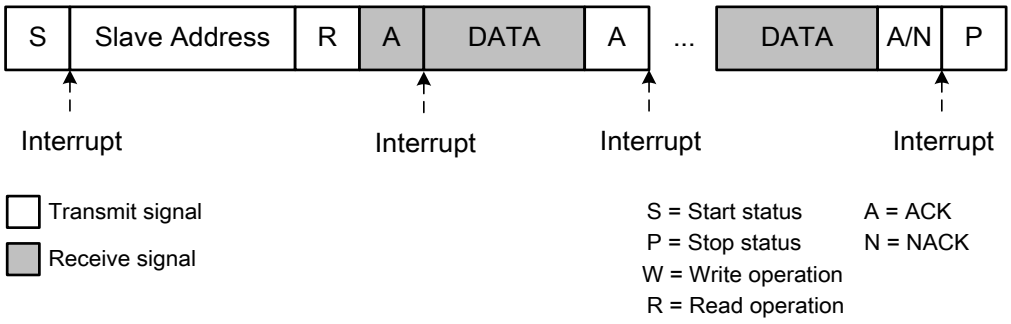
26.2.1 Master Transmitter Mode

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.



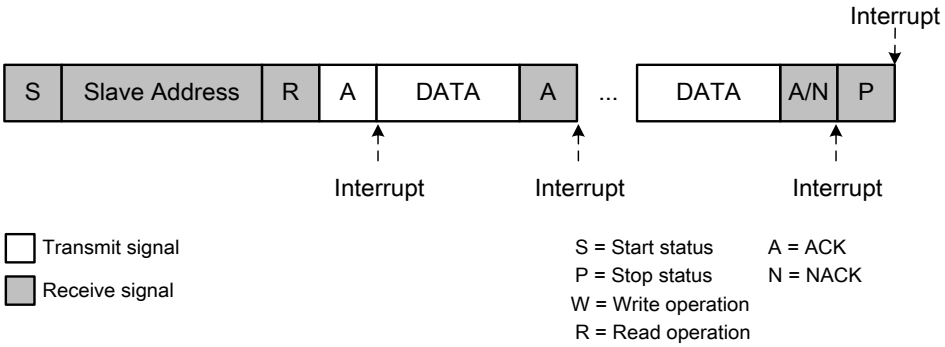
26.2.2 Master Receiver Mode

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.



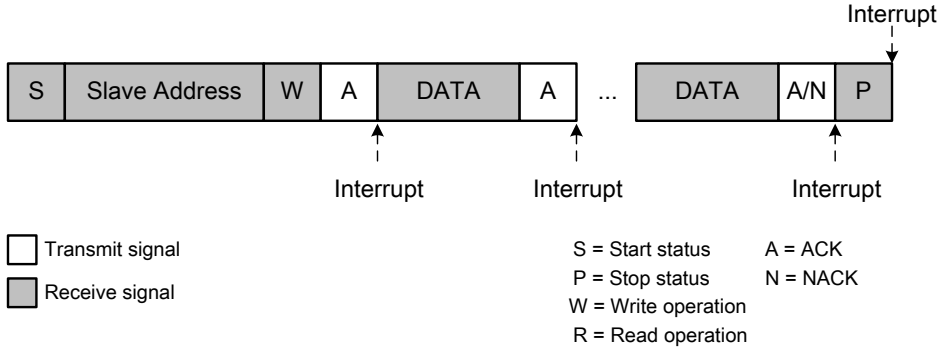
26.2.3 Slave Transmitter Mode

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.



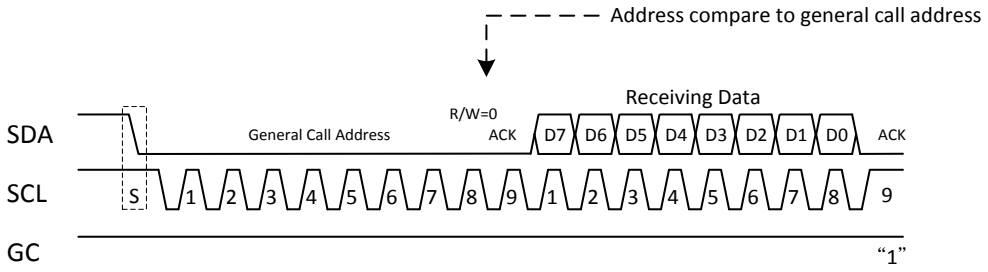
26.2.4 Slave Receiver Mode

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.



26.3 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.



26.4 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.

$$\text{SCL Clock Rate} = \frac{\text{Fcpu}}{\text{Prescaler}} (\text{Prescaler} = 256 \sim 60)$$

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	6MHz	12MHz	16MHz
0	0	0	256	23	47	63
0	0	1	224	27	54	71
0	1	0	192	31	63	83
0	1	1	160	37	75	100
1	0	0	960	6.25	12.5	17
1	0	1	120	50	100	133
1	1	0	60	100	200	266
1	1	1	(Timer 1 overflow rate)/8			

Copyright © 2019, SONiX Technology Co., Ltd.

✸ *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.

26.5 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 mater with LOW status wins the bus control and continues its transmission. There is no data miss during arbitration.

Copyright © 2019, SONiX Technology Co., Ltd.

26.6 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 Tmex, Tsext and Tout. The equation is as following.

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

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=16MHz	
		DEC	HEX
Tmext	5ms	78	4E
Tsext	25ms	391	187
Tout	35ms	547	223

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.

Copyright © 2019, SONiX Technology Co., Ltd.

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.

26.7 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.

26.8 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	-	-	-	-	SMBSTP2	SMBSTP1	SMBSTP0
SMBDST	SMBD7	SMBD6	SMBD5	SMBD4	SMBD3	SMBD2	SMBD1	SMBD0
IEN0	EAL	EX4	EX3	EX2	ET1	EX1	ET0	EX0
IEN1	EU2RX	EU2TX	EU1RX	EU1TX	EU0RX	EU0TX	-	EI2C
P2M	-	-	-	P24M	P23M	P22M	P21M	P20M

Copyright © 2019, SONiX Technology Co., Ltd.

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

Copyright © 2019, SONiX Technology Co., Ltd.

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

Copyright © 2019, SONiX Technology Co., Ltd.

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	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

Copyright © 2019, SONiX Technology Co., Ltd.

								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.

Copyright © 2019, SONiX Technology Co., Ltd.

SMBSEL Register (0xDE)

Bit	Field	Type	Initial	Description
7	SMBEXE	R/W	0	SMBus extension functionality 0: Disable 1: Enable
Else	Reserved	R/W	0	
2..0	SMBSTP[2:0]	R/W	000	SMBus timeout register

SMBDST Register (0xDF)

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

IEN0 Register (0xA8)

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

IEN1 Register (0xB8)

Bit	Field	Type	Initial	Description
0	EI2C	R/W	0	Interrupts enable. Refer to Chapter Interrupt
Else				Refer to other chapter(s)

P2M Register (0xFB)

Bit	Field	Type	Initial	Description
4	P24M	R/W	0	0: Set P2.4 (SDA) as input mode (required) ~~1: Set P2.4 (SDA) as output mode~~*
3	P23M	R/W	0	0: Set P2.3 (SCL) as input mode (required) ~~1: Set P2.3 (SCL) as output mode~~*
Else				Refer to other chapter(s)

* The P23M and P24M require be set input mode.

Copyright © 2019, SONiX Technology Co., Ltd.

26.9 Sample Code

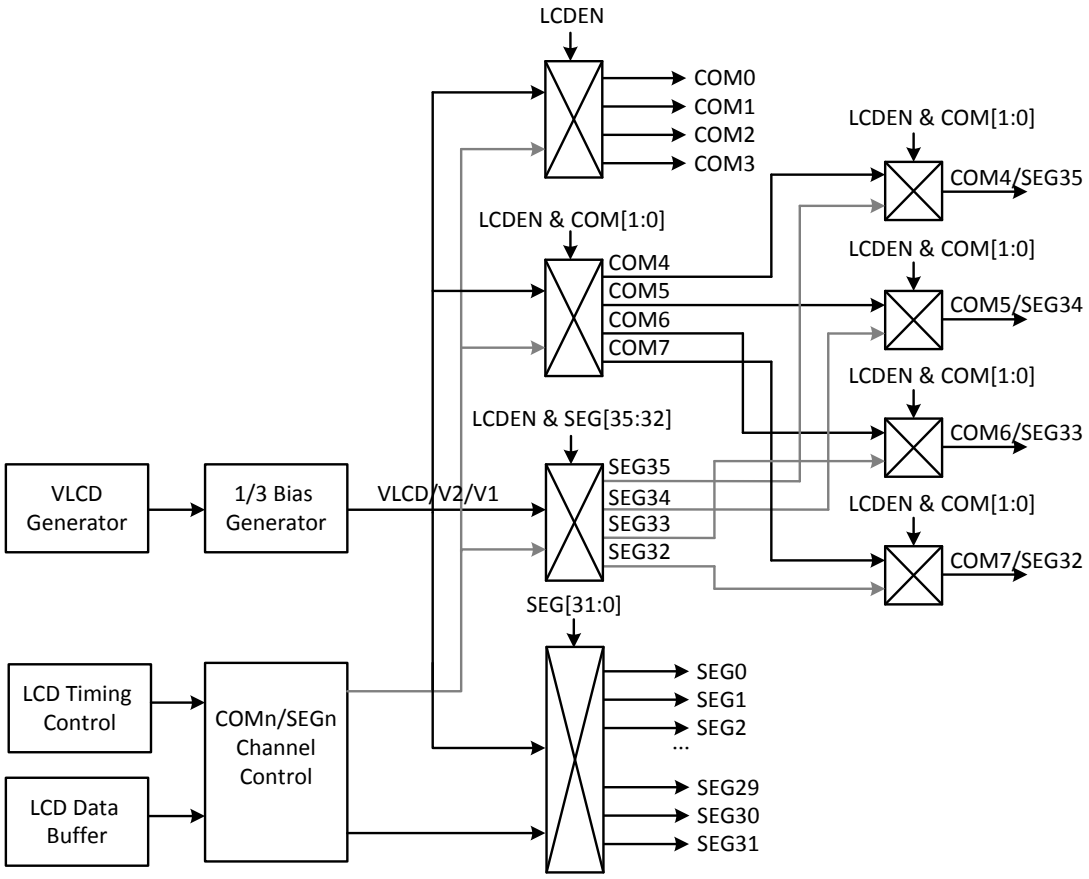
The following sample code demonstrates how to perform I2C with interrupt.

```
unsigned int  I2CAddr;
unsigned int  I2C_TXData0;
unsigned int  I2C_TXDatan;
unsigned int  I2C_RXData0;
unsigned int  I2C_RXDatan;

void I2CInit(void)
{
  P0M &= 0xFC;      // P00 and P01 as input

  // configure I2C clock(T1) and enable I2C.
  I2CCON = 0xC3;
  TMOD = 0x60;   // auto reload
  TCON0 = 0x07;  // Fosc/1
  TH1 = 0xF6;    //400kHz
  TL1 = 0xF6;    //400kHz or
  TH1 = 0xD8;    //100kHz
  TL1 = 0xD8;    //100kHz
  TR1 = 1;

  // enable I2C interrupt
  EI2C = 1;
  EAL = 1;       //enable global interrupt

  I2CCON |= 0x20;            // START (STA) = 1
}

void I2cInterrupt(void) interrupt ISRI2c  //0x93
{
    switch (I2CSTA)
    {
        // tx mode
        case 0x08:
            I2CCON &= 0xDF;           // START (STA) = 0
            I2CDAT = I2CAddr;        // Tx/Rx addr
            break;
        case 0x18:                    // write first byte
            I2CDAT = I2C_TXData0;
            break;
        case 0x28:                    // write n byte
            I2CDAT = I2C_TXDatan;
            break;
        case 0x30:                    // STOP (STO)
            I2CCON |= 0x10;
            break;
        //    rx mode
        case 0x40:                    // get slave addr
            I2CCON |= 0x04;           // AA = 1
            break;
        case 0x50:                    // read n byte
            I2C_RXData0 = I2CDAT;
            I2CCON &= 0xFB;           // AA = 0
            break;
        case 0x58:                    // read last byte & stop
```

```
54              I2C_RXDatan = I2CDAT;
55              I2CCON |= 0x10;           // STOP (STO)
56              break;
57          default:
58              I2CCON |= 0x10;           // STOP (STO)
59      }
60
61      I2CCON &= 0xF7;                   // Clear I2C flag (SI)
62  }
```

27 LCD

The LCD driver generates the control to drive a static or multiplexed LCD panel, with support for up to 36(or 34/33/32) segments multiplexed with up to 4(or 6/7/8) commons. The LCD also provides control of the LCD pixel data. The LCD driver supports a static mode controlled by LSTC control bit. If LSTC=0, the LCD driver is 1/3 bias, 1/4(or 1/6, 1/7, 1/8) duty LCD mode. If LSTC=1, the LCD driver is selected to static mode. The LCD has two clock source and four clock rates to decide LCD frame rate from 15.625Hz to 256Hz. The clock source is from internal 16kHz RC or external 32.768kHz oscillator crystal or RC type and controlled by LCDCKS bit.



In LCD mode, the LCD builds in one internal bias circuit to adjust LCD power and bias voltage. There are 40-pin GPIO shared with COM pins and SEG pins which controlled by LCDSEG, LCDSEG1, LCDSEG2, LCDSEG3, LCDSEG4 registers and LCDCOM[1:0] bits. The LCD builds in P3CON, P4CON, P5CON, P6CON and P7CON registers to set pure analog input pin. After setup LCDEN bit, the LCD starts output analog data. In static mode, the selected COM/SEG pins switch to like GPIO output mode and only output 0V and VDD voltage.

After LCD operating, the system would be waked up from idle mode to normal mode if interrupt enable. When STWK=1, LCD can work in stop mode and waked up from stop mode by LCD interrupt.

Copyright © 2019, SONiX Technology Co., Ltd.

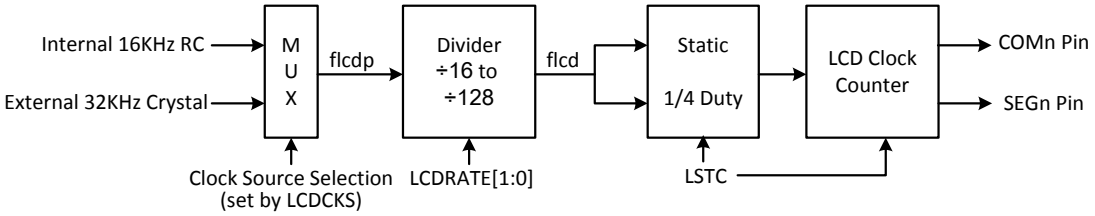
27.1 Configurations of Operation

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

1. The GPIO mode of LCD output channel must be set as low status. (By Pn register)
2. The GPIO mode of LCD output channel must be set as output mode. (By PnM register)
3. The configuration control bit of LCD output channel must be set. (By PnCON register)
4. Choose the LCD common mode. (By LCDCOM[1:0] bits)
5. Choose and enable the LCD segment channel. (By LCDSEG, LCDSEG1, LCDSEG2, LCDSEG3 and LCDSEG4 registers)
6. Choose LCD mode. (By LSTC register)
7. Choose LCD Clock Source and Clock Rate. (By LCDCKS and LCDRATE[1:0] bits)
8. Choose LCD VLCD voltage and bias resistance. (By VLCD[3:0] and LCDBIA[3:0] bits, can be ignored in static mode)
9. Program LDC data to specific data buffer. (By LCDADR and LCDBUF register)
10. After setup LCDEN bits, the LCD ready to generate waveform.

27.2 LCD Timing Control

The LCD timing control generates the clock for LCD frame rate. The LCD has two clock sources (flcdp): internal 16KHz RC and external 32.768KHz crystal, controlled by LCDCKS bit. There are 4-level pre-scaler to divide the flcdp to obtain LCD frame clock source flcd. Each LCD frame length is 2*COM mode*flcd. (COM mode = 4 ~ 8)

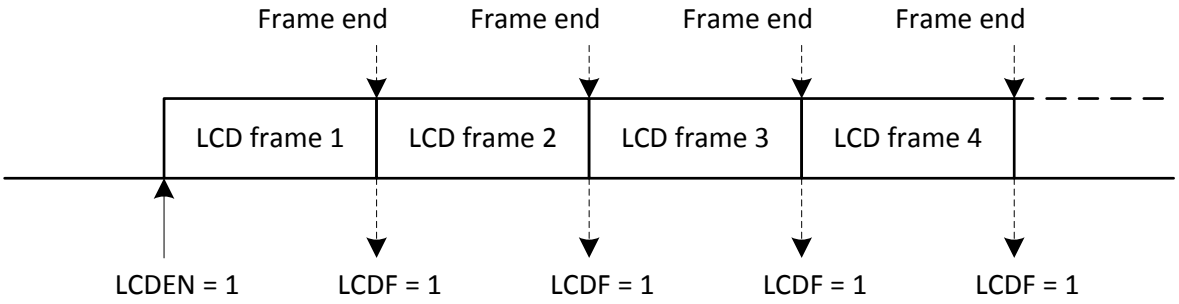


LCD Frame rate (Hz)		LCD Rate[1:0]			
		00(flcdp/16)	01(flcdp/32)	10(flcdp/64)	11(flcdp/128)
Int. 16KHz	4 COM mode	125 Hz	62.5 Hz	31.25 Hz	15.625 Hz
Ext. 32.768KHz		256 Hz	128 Hz	64 Hz	32 Hz
Int. 16KHz	6 COM mode	83.333 Hz	41.6665 Hz	20.83325 Hz	10.416625 Hz
Ext. 32.768KHz		170.666 Hz	85.333 Hz	42.6665 Hz	21.33325 Hz
Int. 16KHz	7 COM mode	71.429 Hz	35.7145 Hz	17.85725 Hz	8.928625 Hz
Ext. 32.768KHz		146.286 Hz	73.143 Hz	36.5715 Hz	18.28575 Hz
Int. 16KHz	8 COM mode	62.5 Hz	31.25 Hz	15.625 Hz	7.8125 Hz
Ext. 32.768KHz		128 Hz	64 Hz	32 Hz	16 Hz

Copyright © 2019, SONiX Technology Co., Ltd.

27.3 LCD Interrupt

The LCDF is set after one LCD frame finish. Every LCD frame will issue LCDF. LCDF will trigger LCD interrupt when ELCD = 1. The LCDF flag is cleared by hardware clear when LCD Interruption is accepted.



27.4 LCD Channel Control

The LCD include 4-COM pins, 32-SEG pins and 4-COM/SEG share pins. Each of the LCD channels is shared with GPIO. If the LCD driver is enabled (LCDEN=1), the 4-COM pins are controlled to LCD COM pin mode, but the 32-SEG pins are controlled by SEG[31:0] selection bits (By LCDSEG, LCDSEG1, LCDSEG2 and LCDSEG3 registers). The COM/SEG share pin will follow the mode selection, to switch the common or segment functions. When the COM mode is selected the pin to common function, the segment function will be ignored and the pin will switch to common function. Conversely, the segment function is enabled according to the segment control bit setting.

If the LCD driver is disabled (LCDEN=0), all COM and SEG pins are controlled to GPIO mode. Each of the LCD channels has a bias switch circuit to select a bias voltage for COM or SEG driving waveform. The bias voltage is from the LCD bias generator. The bias switch is controlled by LCD COM and SEG output control.

In LCD static mode, the bias switch of COM and SEG pins is disabled. The selected COM and SEG pins switch to like GPIO output mode and only output 0V and VDD voltage. The LCD COM and SEG output control circuits control the GPIO output latch of COM and SEG pins directly.

Copyright © 2019, SONiX Technology Co., Ltd.

Table 27-1 The LCD COM/SEG share pin control

<table>
<tr><td rowspan="2">LCDEN</td><td rowspan="2">COM mode</td><td>LCD COM</td><td colspan="5">LCD SEG</td><td colspan="4">LCD COM/SEG share</td></tr>
<tr><td>COM0~COM3</td><td>SEG0</td><td>SEG1</td><td>…</td><td>SEG30</td><td>SEG31</td><td>SEG32/ COM7</td><td>SEG33/ COM6</td><td>SEG34/ COM5</td><td>SEG35/ COM4</td></tr>
<tr><td>0</td><td>X</td><td>GPIO</td><td>GPIO</td><td>GPIO</td><td>GPIO</td><td>GPIO</td><td>GPIO</td><td>GPIO</td><td>GPIO</td><td>GPIO</td><td>GPIO</td><td>GPIO</td></tr>
<tr><td rowspan="4">1</td><td>4</td><td rowspan="4">COM</td><td colspan="5" rowspan="4">Control by SEG[31:0]
SEGn= 0, SEGn pin is controlled to GPIO mode.
SEGn= 1, SEGn pin is controlled to SEG pin mode</td><td rowspan="3">GPIO:
SEG32 = 0
SEG32:
SEG32 = 1</td><td rowspan="2">GPIO:
SEG33 = 0
SEG33:
SEG33 = 1</td><td>GPIO:
SEG34 = 0
SEG34:
SEG34 = 1</td><td>GPIO:
SEG35 = 0
SEG35:
SEG35 = 1</td></tr>
<tr><td>6</td><td rowspan="2">COM5</td><td rowspan="2">COM4</td></tr>
<tr><td>7</td><td rowspan="2">COM6</td></tr>
<tr><td>8</td><td>COM7</td></tr>
</table>

When LCDIDLE bit is set, If the LCD driver is disabled (LCDEN=0), all COM pins are controlled to V1 or VDD by LSTC bit.

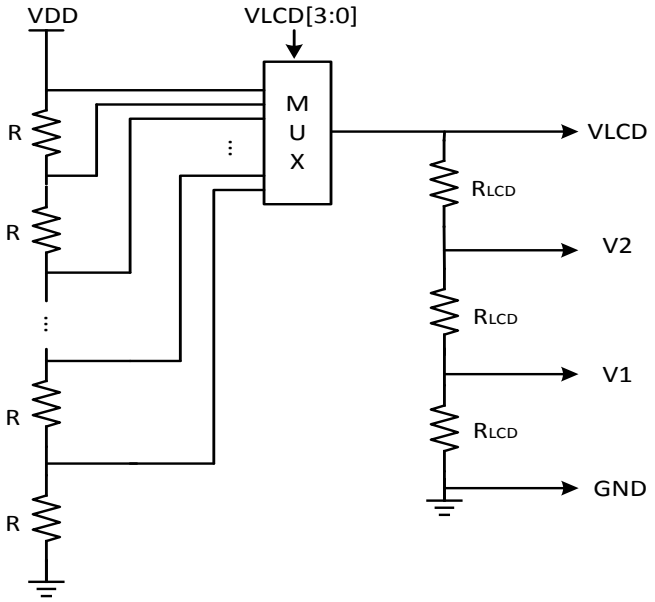
Table 27-2 The LCD COM/SEG channel status control

LCDEN	LSTC	LCDIDLE	COMn pin	SEGn pin	
				SEGn enable	COM share
0	0	0	GPIO	GPIO	GPIO
0	0	1	V1	GPIO	V1
0	1	0	GPIO	GPIO	GPIO
0	1	1	VDD	GPIO	VDD
1	0	0	LCD COM	LCD SEG	LCD COM
1	0	1	LCD COM	LCD SEG	LCD COM
1	1	0	Static COM	Static SEG	Static COM
1	1	1	Static COM	Static SEG	Static COM

Copyright © 2019, SONiX Technology Co., Ltd.

27.5 LCD Bias Generator

The LCD bias generator include LCD contrast control and 1/3 bias control. The LCD contrast control is used to set the voltage of VLCD. There are 16-stage VLCD voltage from VDD to VDD*0.5 and is controlled by VLCD[3:0]. The 1/3 bias control circuit has three resistances (R_{LCD}) for difference VLCD controlled by LCDBIA[2:0]. The R_{LCD} is range from 17.65KΩ to 300KΩ. The final 4-level LCD bias voltage source is VLCD/ V2/ V1/ GND and supply to LCD COM and SEG pins. When the static mode is active (LSTC=1), both LCD contrast control and 1/3 bias control will be turned off.



27.6 LCD Data Buffer

The LCD data buffer stores LCD map data and the density is 36x8-bit. The LCD data buffer is indirectly address memory and accessed through LCDADR and LCDBUF control registers. The LCDADR indicate the address of each LCD segment data buffer from SEG0~SEG35. Program the LCD data must set address (LCDADR) first and then write data to LCDBUF. The LCD data of each address includes eight common bits data. The LCD RAM map is as following.

LCD	LCDADR[5:0]	LCDBUF[7:0]							
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0
SEG0	000000	C7S0	C6S0	C5S0	C4S0	C3S0	C2S0	C1S0	C0S0
SEG1	000001	C7S1	C6S1	C5S1	C4S1	C3S1	C2S1	C1S1	C0S1
SEG2	000010	C7S2	C6S2	C5S2	C4S2	C3S2	C2S2	C1S2	C0S2
SEG3	000011	C7S3	C6S3	C5S3	C4S3	C3S3	C2S3	C1S3	C0S3
SEG4	000100	C7S4	C6S4	C5S4	C4S4	C3S4	C2S4	C1S4	C0S4
SEG5	000101	C7S5	C6S5	C5S5	C4S5	C3S5	C2S5	C1S5	C0S5

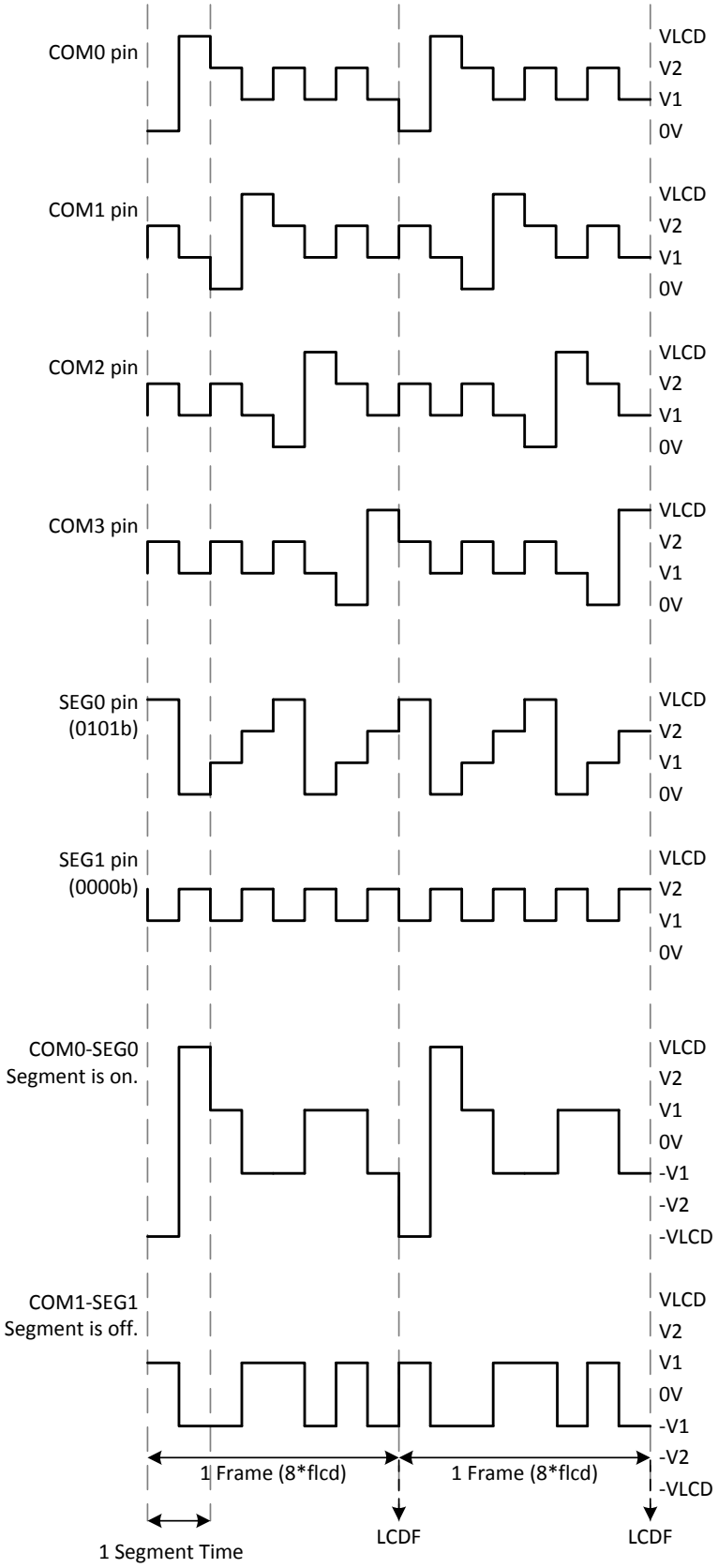
Copyright © 2019, SONiX Technology Co., Ltd.

SEG6	000110	C7S6	C6S6	C5S6	C4S6	C3S6	C2S6	C1S6	C0S6
SEG7	000111	C7S7	C6S7	C5S7	C4S7	C3S7	C2S7	C1S7	C0S7
SEG8	001000	C7S8	C6S8	C5S8	C4S8	C3S8	C2S8	C1S8	C0S8
SEG9	001001	C7S9	C6S9	C5S9	C4S9	C3S9	C2S9	C1S9	C0S9
SEG10	001010	C7S10	C6S10	C5S10	C4S10	C3S10	C2S10	C1S10	C0S10
SEG11	001011	C7S11	C6S11	C5S11	C4S11	C3S11	C2S11	C1S11	C0S11
SEG12	001100	C7S12	C6S12	C5S12	C4S12	C3S12	C2S12	C1S12	C0S12
SEG13	001101	C7S13	C6S13	C5S13	C4S13	C3S13	C2S13	C1S13	C0S13
SEG14	001110	C7S14	C6S14	C5S14	C4S14	C3S14	C2S14	C1S14	C0S14
SEG15	001111	C7S15	C6S15	C5S15	C4S15	C3S15	C2S15	C1S15	C0S15
SEG16	010000	C7S16	C6S16	C5S16	C4S16	C3S16	C2S16	C1S16	C0S16
SEG17	010001	C7S17	C6S17	C5S17	C4S17	C3S17	C2S17	C1S17	C0S17
SEG18	010010	C7S18	C6S18	C5S18	C4S18	C3S18	C2S18	C1S18	C0S18
SEG19	010011	C7S19	C6S19	C5S19	C4S19	C3S19	C2S19	C1S19	C0S19
SEG20	010100	C7S20	C6S20	C5S20	C4S20	C3S20	C2S20	C1S20	C0S20
SEG21	010101	C7S21	C6S21	C5S21	C4S21	C3S21	C2S21	C1S21	C0S21
SEG22	010110	C7S22	C6S22	C5S22	C4S22	C3S22	C2S22	C1S22	C0S22
SEG23	010111	C7S23	C6S23	C5S23	C4S23	C3S23	C2S23	C1S23	C0S23
SEG24	011000	C7S24	C6S24	C5S24	C4S24	C3S24	C2S24	C1S24	C0S24
SEG25	011001	C7S25	C6S25	C5S25	C4S25	C3S25	C2S25	C1S25	C0S25
SEG26	011010	C7S26	C6S26	C5S26	C4S26	C3S26	C2S26	C1S26	C0S26
SEG27	011011	C7S27	C6S27	C5S27	C4S27	C3S27	C2S27	C1S27	C0S27
SEG28	011100	C7S28	C6S28	C5S28	C4S28	C3S28	C2S28	C1S28	C0S28
SEG29	011101	C7S29	C6S29	C5S29	C4S29	C3S29	C2S29	C1S29	C0S29
SEG30	011110	C7S30	C6S30	C5S30	C4S30	C3S30	C2S30	C1S30	C0S30
SEG31	011111	C7S31	C6S31	C5S31	C4S31	C3S31	C2S31	C1S31	C0S31
SEG32	100000	C7S32	C6S32	C5S32	C4S32	C3S32	C2S32	C1S32	C0S32
SEG33	100001	C7S33	C6S33	C5S33	C4S33	C3S33	C2S33	C1S33	C0S33
SEG34	100010	C7S34	C6S34	C5S34	C4S34	C3S34	C2S34	C1S34	C0S34
SEG35	100011	C7S35	C6S35	C5S35	C4S35	C3S35	C2S35	C1S35	C0S35

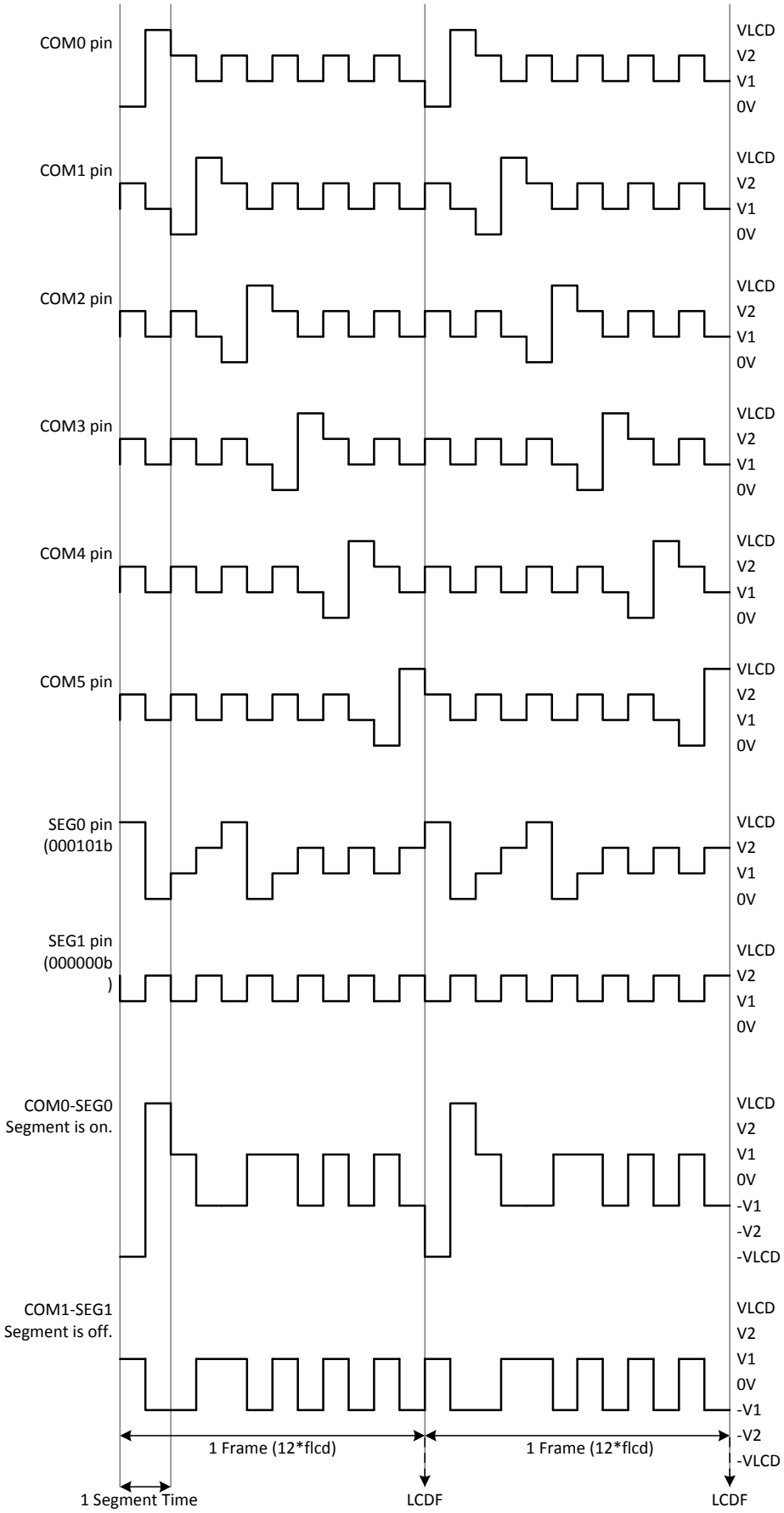
Copyright © 2019, SONiX Technology Co., Ltd.

27.7 LCD Waveform Generation

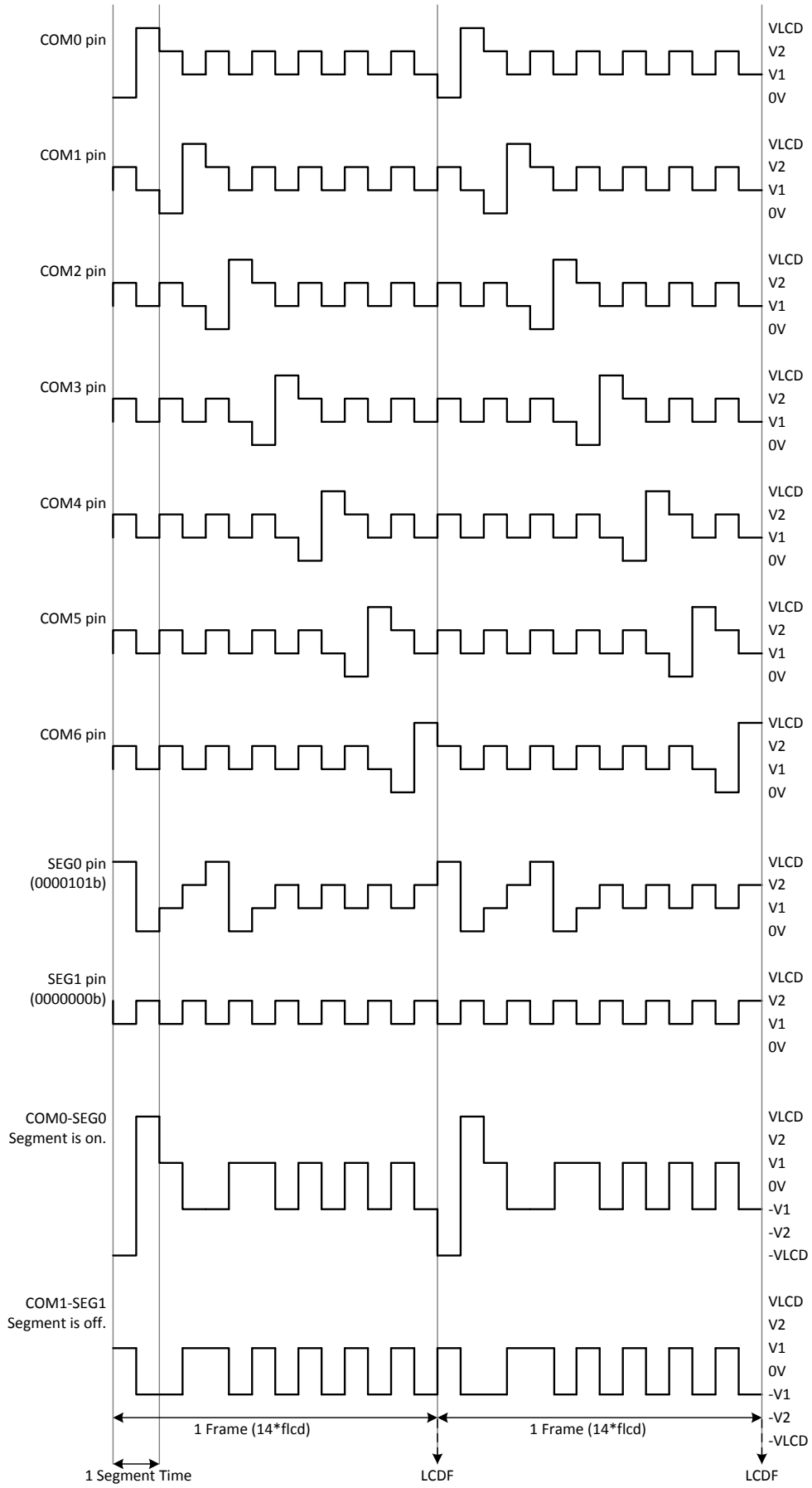
Waveform in LCD mode with 4 COM mode:



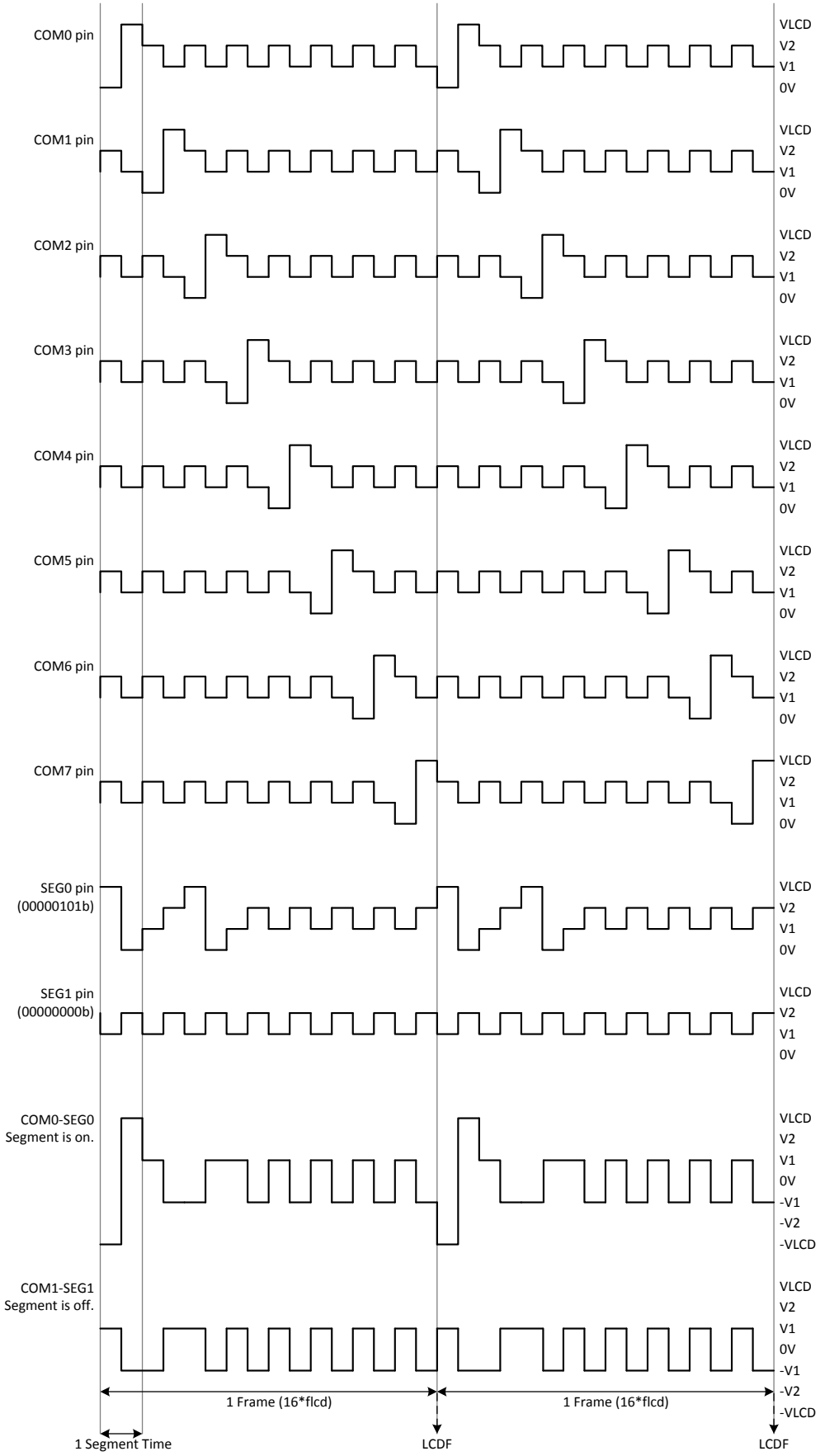
Waveform in LCD mode with 6 COM mode:



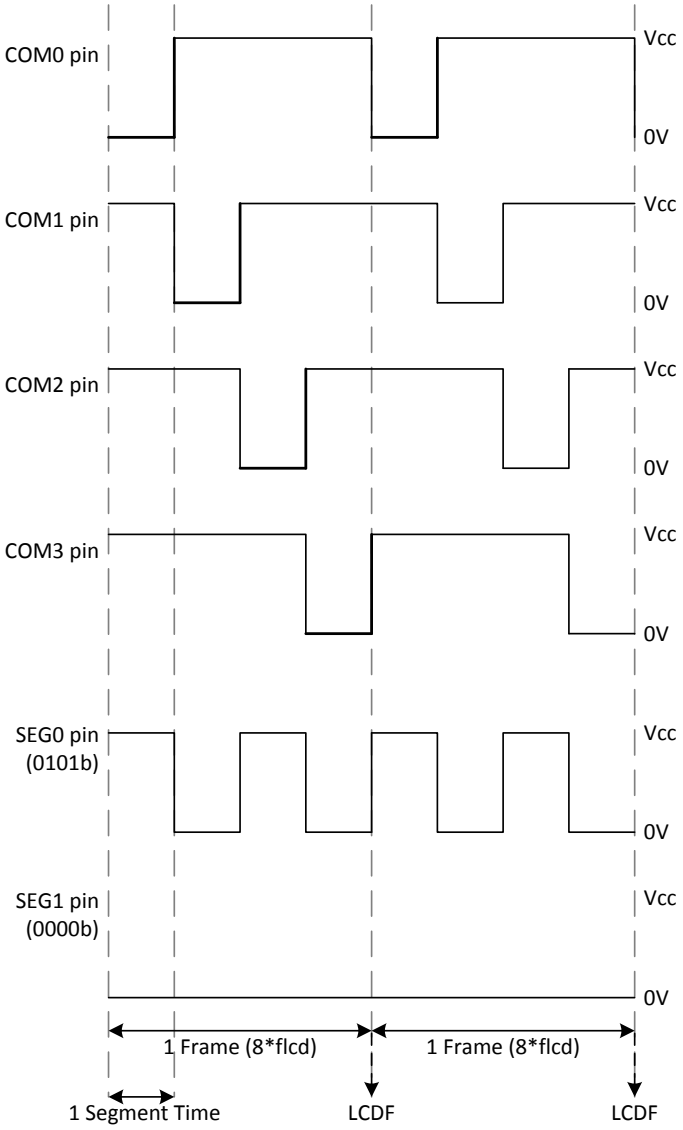
Waveform in LCD mode with 7 COM mode:



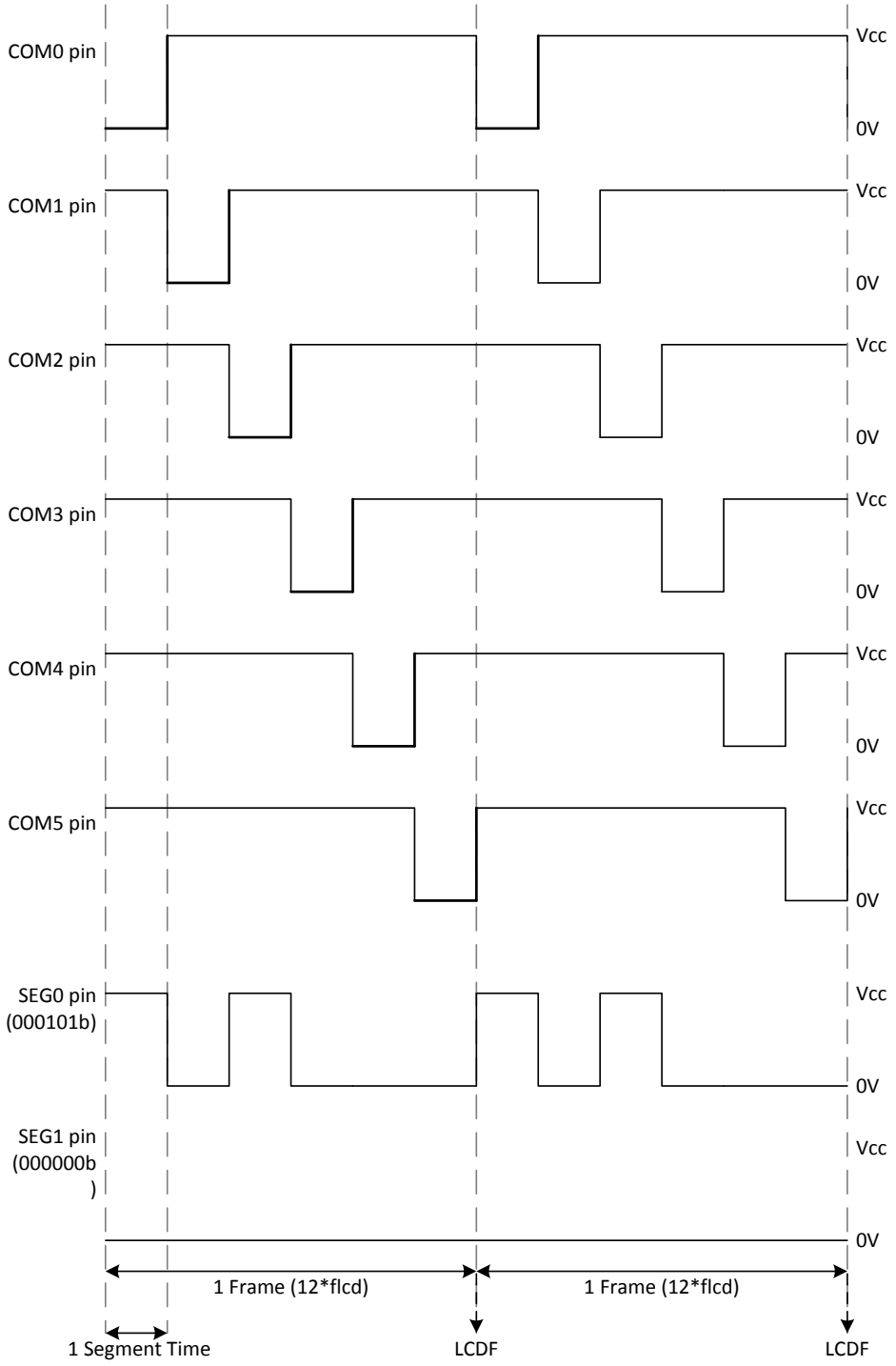
Waveform in LCD mode with 8 COM mode:



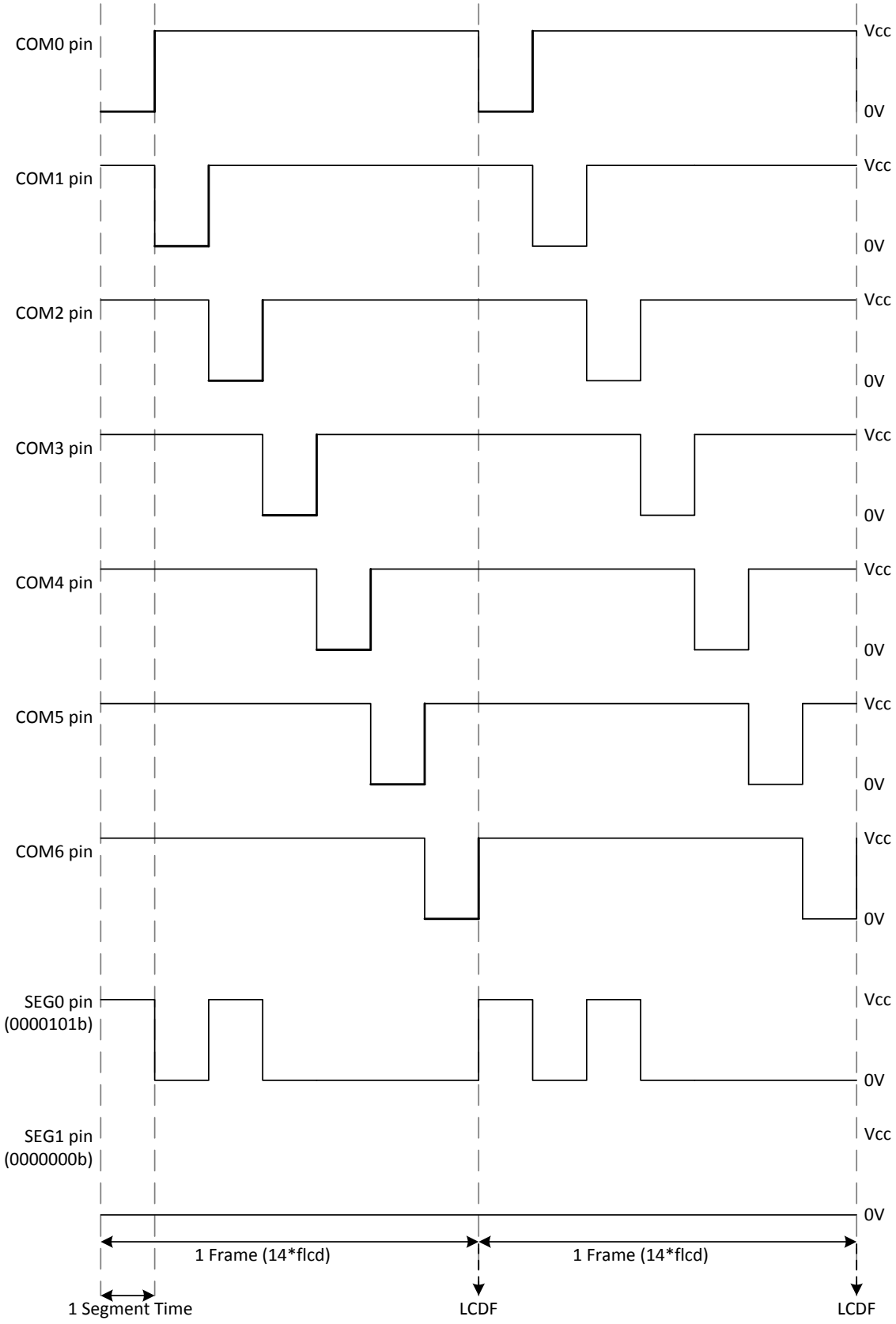
Waveform in static mode with 4 COM mode:



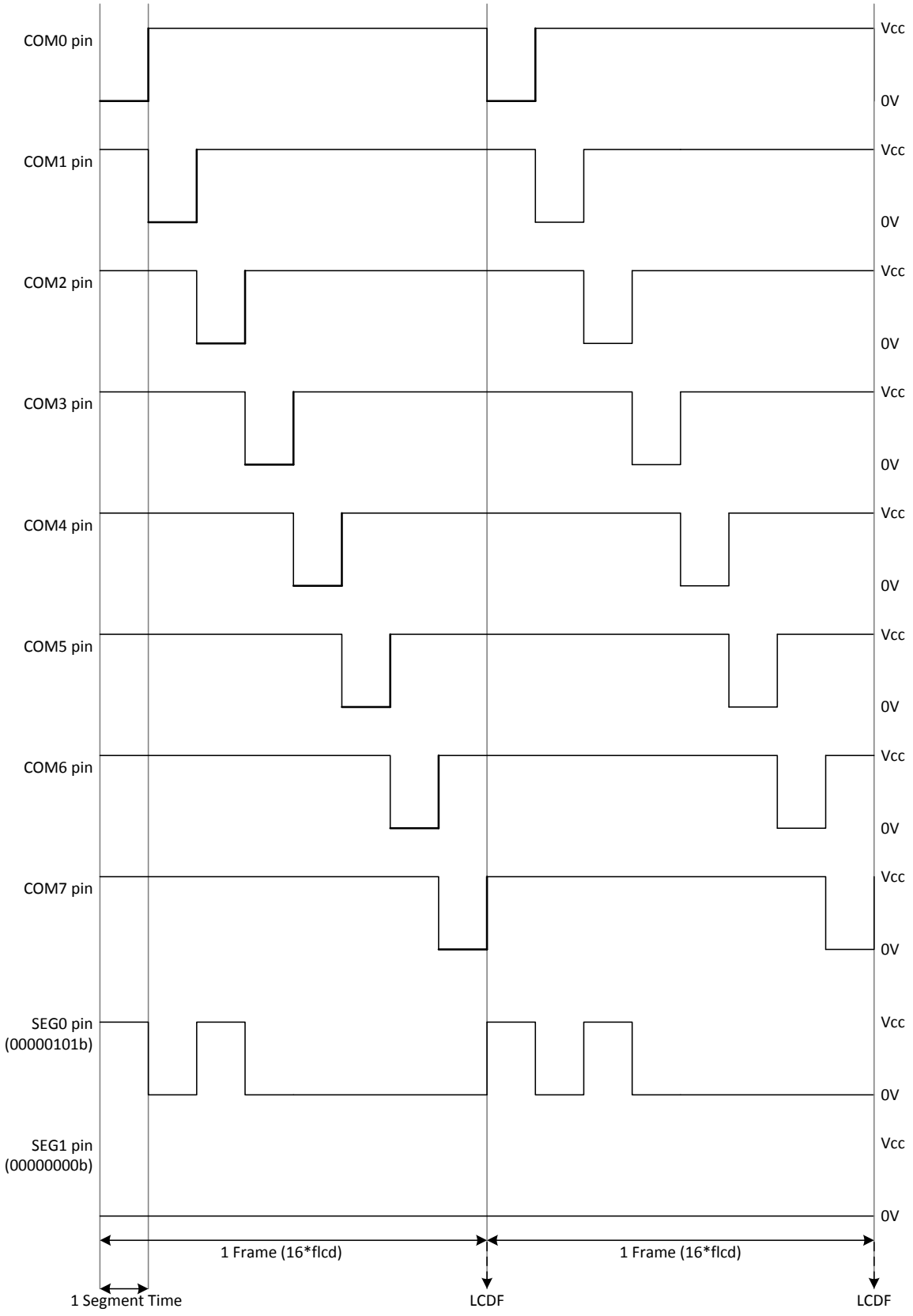
Waveform in static mode with 6 COM mode:



Waveform in static mode with 7 COM mode:

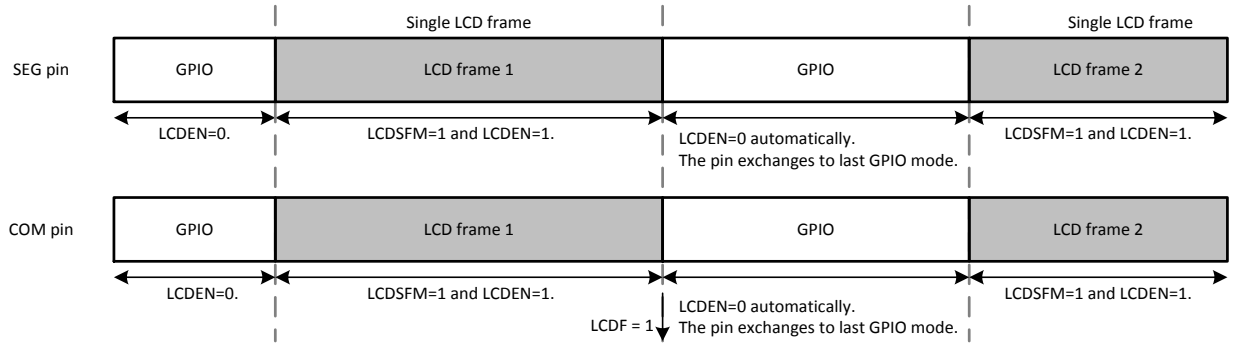


Waveform in static mode with 8 COM mode:



27.8 Single LCD Frame Function

When LCDSFM = 0, LCD frame will continuously output until LCD disable. When LCDSFM = 1 and LCDEN=1, LCD will output single LCD frame and the LCDF is issued as LCD frame finish. LCDEN bit is cleared automatically and LCD pin returns to GPIO mode (or idle status by LCDIDLE bit). To output next frame is to set LCDEN bit by program again.



✸ ***Note: If LCDIDLE bit is set, the COM pins will are controlled to idle status when LCDEN bit is cleared. The idle status is V1 or VDD by LSTC bit controls.***

27.9 LCD Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LCDCON	LCDEN	LSTC	LCDRATE1	LCDRATE0	VLCD3	VLCD2	VLCD1	VLCD0
LCDMOD	LCDIDLE	LCDSFM	LCDCOM1	LCDCOM0	LCDCKS	LCDBIA2	LCDBIA1	LCDBIA0
LCDSEG	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
LCDSEG1	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8
LCDSEG2	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16
LCDSEG3	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24
LCDSEG4	-	-	-	-	SEG35	SEG34	SEG33	SEG32
LCDBUF	LCDDAT7	LCDDAT6	LCDDAT5	LCDDAT4	LCDDAT3	LCDDAT2	LCDDAT1	LCDDAT0
LCDADR	-	-	SADR5	SADR4	SADR3	SADR2	SADR1	SADR0
P3CON	P3CON7	P3CON6	P3CON5	P3CON4	P3CON3	P3CON2	P3CON1	P3CON0
P4CON	P4CON7	P4CON6	P4CON5	P4CON4	P4CON3	P4CON2	P4CON1	P4CON0
P5CON	P5CON7	P5CON6	P5CON5	P5CON4	P5CON3	P5CON2	P5CON1	P5CON0
P6CON	P6CON7	P6CON6	P6CON5	P6CON4	P6CON3	P6CON2	P6CON1	P6CON0
P7CON	P7CON7	P7CON6	P7CON5	P7CON4	P7CON3	P7CON2	P7CON1	P7CON0
IEN0	EAL	EX4	EX3	EX2	ET1	EX1	ET0	EX0

Copyright © 2019, SONiX Technology Co., Ltd.

IEN2	ET4	ELCD	ET3	ET2	EPW2	EPW1	-	EADC
IRCON2	TF4	LCDF	TF3	TF2	PW2F	PW1F	-	ADCF

LCDCON Register (0XA1)

Bit	Field	Type	Initial	Description
7	LCDEN	R/W	0	LCD driver control bit. 0: Disable 1: Enable
6	LSTC	R/W	0	LCD static mode select bit. 0: LCD mode. 1: Static mode.
5..4	LCDRATE[1:0]	R/W	0	LCD clock rate select bit 00: flcd/16 01: flcd/32 10: flcd/64 11: flcd/128
3..0	VLCD[3:0]	R/W	0x00	LCD contrast adjustment bit. 0000: VLCD=VDD, 0001: VLCD=VDD*0.97, 0010: VLCD=VDD*0.93, 0011: VLCD=VDD*0.9, 0100: VLCD=VDD*0.87, 0101: VLCD=VDD*0.83, 0110: VLCD=VDD*0.8, 0111: VLCD=VDD*0.77, 1000: VLCD=VDD*0.73, 1001: VLCD=VDD*0.7, 1010: VLCD=VDD*0.67, 1011: VLCD=VDD*0.63, 1100: VLCD=VDD*0.6, 1101: VLCD=VDD*0.57, 1110: VLCD=VDD*0.53, 1111: VLCD=VDD*0.5

Copyright © 2019, SONiX Technology Co., Ltd.

LCDMOD Register (0XA2)

Bit	Field	Type	Initial	Description
7	LCDIDLE	R/W	0	IDLE status of the COM pin control bit when LCD disable. 0: The COM pins are GPIO. 1: The COM pins are V1 or VDD by LSTC bit.
6	LCDSFM	R/W	0	LCD single frame function control bit. 0: Disable 1: Enable
5..4	LCDCOM[1:0]	R/W	00	LCD COM mode select bit. 00: LCD 4 COM mode. 01: LCD 6 COM mode. 10: LCD 7 COM mode. 11: LCD 8 COM mode.
3	LCDCKS	R/W	0	LCD clock source (flcd) select bit. 0: Internal 16KHz RC 1: External 32KHz crystal*
2..0	LCDBIA[2:0]	R/W	000	LCD bias resistance select bit 000: Disable 001: RLCD = 25KΩ 010: RLCD = 75KΩ 011: RLCD = 18.75KΩ 100: RLCD = 300KΩ 101: RLCD = 23.08KΩ 110: RLCD = 60KΩ 111: RLCD = 17.65KΩ

*Before choose External 32KHz crystal as LCD clock source, the LCKS bit must be set to enable 32KHz crystal function.

LCDSEG Register (0XA3)

Bit	Field	Type	Initial	Description
7..0	SEG[7:0]	R/W	0x00	LCD SEGn channel control bit. 0: GPIO pin 1: SEG pin.

Copyright © 2019, SONiX Technology Co., Ltd.

LCDSEG1 Register (0XA6)

Bit	Field	Type	Initial	Description
7..0	SEG[15:8]	R/W	0x00	LCD SEGn channel control bit. 0: GPIO pin 1: SEG pin.

LCDSEG2 Register (0XA7)

Bit	Field	Type	Initial	Description
7..0	SEG[23:16]	R/W	0x00	LCD SEGn channel control bit. 0: GPIO pin 1: SEG pin.

LCDSEG3 Register (0XAA)

Bit	Field	Type	Initial	Description
7..0	SEG[31:24]	R/W	0x00	LCD SEGn channel control bit. 0: GPIO pin 1: SEG pin.

LCDSEG4 Register (0XAB)

Bit	Field	Type	Initial	Description
Else	Reserved	R	0	
3..0	SEG[35:32]	R/W	0000	LCD SEGn channel control bit. 0: GPIO pin 1: SEG pin.

LCDBUF Register (0XA4)

Bit	Field	Type	Initial	Description
7..0	LCDDAT[7:0]	R/W	0x00	LCD SEG data buffer.

LCDADR Register (0XA5)

Bit	Field	Type	Initial	Description
5..0	SADR[5:0]	R/W	000000	LCD SEG data buffer address.
Else	Reserved	R	0	

Copyright © 2019, SONiX Technology Co., Ltd.

P3CON Register (0x9E)

Bit	Field	Type	Initial	Description
7..0	P3CON[7:0]	R/W	0x00	P3 configuration control bit*. 0: P3 can be analog input pin (LCD pin) or digital GPIO pin. 1: P3 is pure analog input pin and can't be a digital GPIO pin.

* P3CON [7:0] will configure related Port0 pin as pure analog input pin to avoid current leakage.

P4CON Register (0x9F)

Bit	Field	Type	Initial	Description
7..0	P4CON[7:0]	R/W	0x00	P4 configuration control bit*. 0: P4 can be analog input pin (LCD pin) or digital GPIO pin. 1: P4 is pure analog input pin and can't be a digital GPIO pin.

* P4CON [7:0] will configure related Port1 pin as pure analog input pin to avoid current leakage.

P5CON Register (0xCF)

Bit	Field	Type	Initial	Description
7..0	P5CON[7:0]	R/W	0x0	P5 configuration control bit*. 0: P5 can be analog input pin (LCD pin) or digital GPIO pin. 1: P5 is pure analog input pin and can't be a digital GPIO pin.

* P5CON [7:0] will configure related Port2 pin as pure analog input pin to avoid current leakage.

P6CON Register (0xD6)

Bit	Field	Type	Initial	Description
7..0	P6CON[1:0]	R/W	0x0	P6 configuration control bit*. 0: P6 can be analog input pin (LCD pin) or digital GPIO pin. 1: P6 is pure analog input pin and can't be a digital GPIO pin.

* P6CON [7:0] will configure related Port3 pin as pure analog input pin to avoid current leakage.

Copyright © 2019, SONiX Technology Co., Ltd.

P7CON Register (0xD7)

Bit	Field	Type	Initial	Description
7..0	P7CON[1:0]	R/W	0x0	P7 configuration control bit*. 0: P7 can be analog input pin (LCD pin) or digital GPIO pin. 1: P7 is pure analog input pin and can't be a digital GPIO pin.

* P7CON [7:0] will configure related Port3 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
6	ELCD	R/W	0	LCD interrupt control bit. 0: Disable LCD interrupt function. 1: Enable LCD interrupt function.
Else				Refer to other chapter(s)

IRCON2 Register (0xBF)

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

Copyright © 2019, SONiX Technology Co., Ltd.

27.10 Sample Code

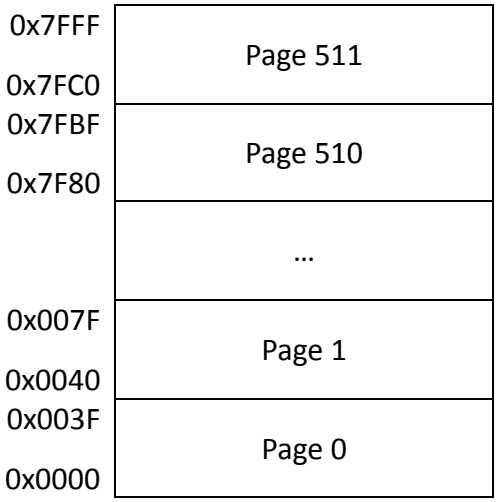
The following sample code demonstrates how to perform LCD with 6 COM mode and SEG5.

```
#define LCDBias         (2 << 0)   //RLCD = 75KΩ
#define LCDClkILRC      (0 << 3)   //LCD clock source = ILRC
#define LCDClk32K       (1 << 3)   //LCD clock source = Ext. 32KHz
#define LCDVLCD         (0 << 0)   //VLCD = VDD
#define LCDSpeedDiv16   (0 << 4)   //LCD clock = flcd/16
#define LCDSpeedDiv32   (1 << 4)   //LCD clock = flcd/32
#define LCDSpeedDiv64   (2 << 4)   //LCD clock = flcd/64
#define LCDSpeedDiv128  (3 << 4)   //LCD clock = flcd/128
#define SelSEG5         (5 << 0)   //select SEG5
#define COM4Mode        (0 << 4)   //4 COM mode
#define COM6Mode        (1 << 4)   //6 COM mode
#define COM7Mode        (2 << 4)   //7 COM mode
#define COM8Mode        (3 << 4)   //8 COM mode
#define LCDStatic       (1 << 6)   //Set static mode
#define LCDEn           (1 << 7)   //enable LCD
#define ELCD            (1 << 6)  //enable LCD interrupt

void LCDInit(void)
{
  P1  = 0x00;
  P1M = 0x80;
  P6  = 0x00;
  P7  = 0x00;
  P6M = 0x04;
  P1M = 0x3F;
  // set COM0~COM5 & SEG5 pin's mode at pure analog pin
  P6CON |= 0x04;     //SEG5/P62
  P7CON |= 0x3F;     //COM0 ~ 5 = P70 ~ P75

  // select 6 COM mode, LCD clock source and LCD Bias, configure LCD channel.
  LCDMOD = COM6Mode | LCDClkILRC | LCDBias;
  LCDSEG = SelSEG5;

  // configure LCD clock divider and VLCD.
  LCDCON = LCDSpeedDiv128 | LCDVLCD;

  // Program SEG5 LCD data
  LCDADR = 0x05;
  LCDBUF = 0x5A;

  // enable LCD interrupt
  IEN2  |= ELCD;
  IEN0  |= 0x80;    //enable global interrupt

  // enable LCD
  LCDCON |= LCDEn;
}

void LCDInterrupt(void) interrupt ISRLcd  //0x5B
{
  //LCDF clear by hardware
  P17 = ~P17;
}
```

Copyright © 2019, SONiX Technology Co., Ltd.

28 In-System Program

SN8F5829 builds in an on-chip 32 KB program memory, aka IROM, which is equally divided to 512 pages (64 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)

28.1 Page Program

Because each page of the program memory has 64 bytes in length, a page program procedure requires 64 bytes IRAM as its data buffer.

ISP ROM MAP		ROM address bit0~bit5 (hex) =0
ROM address bit6~bit15 (hex)	0000	These pages include reset vector and interrupt sector. We strongly recommend to reserve the area not to do ISP erase.
	0040	
	0080	
	00C0	
	0100	One ISP Program Page
	0140	One ISP Program Page
	…	One ISP Program Page
	3000	One ISP Program Page
	3040	One ISP Program Page
	3080	One ISP Program Page
	…	One ISP Program Page
	7F40	One ISP Program Page
	7F80	One ISP Program Page
	7FC0	This page includes ROM reserved area. We strongly recommend to reserve the area not to do ISP erase.

Copyright © 2019, SONiX Technology Co., Ltd.

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

As an example, assume the 510th page of program memory (IROM, 0x7F80 – 0x7FBF) is the anticipated update area; the content is already stored in IRAM address 0x40 – 0x7F. To perform the in-system program, simply write starting IROM address 0x7F80 to EPROMH/EPROML registers, and then specify buffer starting address 0x40 to EPRAM register. Subsequently, write '0x5A' into PECMD [7:0] registers to duplicate the buffer's data to 510th page of IROM.

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

28.2 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.

Copyright © 2019, SONiX Technology Co., Ltd.

✷ ***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.

28.3 In-system Program Register

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PERAM	PERAM7	PERAM6	PERAM5	PERAM4	PERAM3	PERAM2	PERAM1	PERAM0
PEROMH	PEROM15	PEROM14	PEROM13	PEROM12	PEROM11	PEROM10	PEROM9	PEROM8
PEROML	PEROM7	PEROM6	PEROM5	PEROM4	PEROM3	PEROM2	PEROM1	PEROM0
PECMD	PECMD7	PECMD6	PECMD5	PECMD4	PECMD3	PECMD2	PECMD1	PECMD0

PERAM Register (0x97)

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

PEROMH Register (0x96)

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

PEROML Register (0x95)

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

Copyright © 2019, SONiX Technology Co., Ltd.

28.4 Sample Code

```
unsigned cahr idata dataBuffer[64] _at_ 0xC0;  // IRAM 0xC0 to 0xFF

void SYSIspSetDataBuffer(unsigned char address, unsigned char data)
{
  dataBuffer[address & 0x3F] = data;
}

void SYSPageIspStart(unsigned int pageAddress)
{
  PISP(pageAddress, 0xC0);  //Page program
}

void SYSByteIspStart(unsigned int byteAddress)
{
  BISP(byteAddress, 0xE0);  //Byte program
}

```

Copyright © 2019, SONiX Technology Co., Ltd.

29 Clock Fine-Tuning

SN8F5829 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.

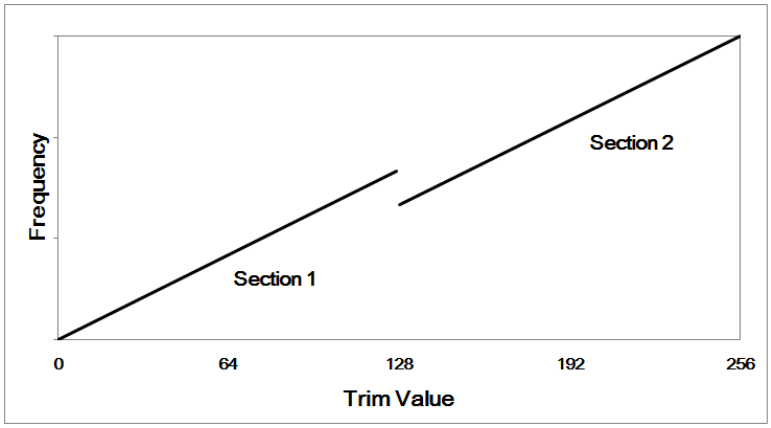
29.1 Clock Trim Section

Clock fine-tuning consists of 8 trim sections as Table 29-1. 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 29-1 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.



29.2 Clock Fine-Tuning Procedure

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

1. Select code option CK_Fine_Tuning = 1 to enable clock fine-tuning function.
2. As the Max. IROM fetching cycle is 16MHz, it is recommended to set PWSC[2:0]=7 at first to avoid system error.
3. Read 10-bit 32MHz trim value from FRQ[9:0]
4. Check the fine-tuning range from the Table 29-1.
5. Write the new clock trim value to FRQ[9:0].
6. Write '0x3C' into FRQCMD [7:0] to trigger clock fine-tuning function.

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

29.3 Clock Fine-Tuning Register

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

FRQ Register (FRQH:0xCE, FRQL:0xCD)

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

FRQCMD Register (0xC5)

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

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

Copyright © 2019, SONiX Technology Co., Ltd.

29.4 Sample Code

The following sample code demonstrates how to perform clock fine-tuning.

```
    FRQH = 0x01;
    FRQL = 0x59;       // Set FRQ[9:0]= 0x159
    FRQCMD = 0x3C;    // Apply new fine-tuning setting
```

Copyright © 2019, SONiX Technology Co., Ltd.

30 Electrical Characteristics

30.1 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
Junction Temperature..... -40°C to 125°C

30.2 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 (CKCON=0x00, 32MHz IHRC)	VDD = 3V, fcpu = 1MHz		3.45		mA
		VDD = 5V, fcpu = 1MHz		3.50		mA
		VDD = 3V, fcpu = 4MHz		4.09		mA
		VDD = 5V, fcpu = 4MHz		4.10		mA
		VDD = 3V, fcpu = 8MHz		4.90		mA
		VDD = 5V, fcpu = 8MHz		5.00		mA
		VDD = 3V, fcpu = 16MHz		6.50		mA
		VDD = 5V, fcpu = 16MHz		6.80		mA
	Normal mode supply current (CKCON=0x00, 16MHz Crystal)	VDD = 3V, fcpu = 1MHz		3.19		mA
		VDD = 5V, fcpu = 1MHz		3.73		mA
		VDD = 3V, fcpu = 4MHz		3.90		mA
		VDD = 5V, fcpu = 4MHz		4.45		mA
		VDD = 3V, fcpu = 8MHz		4.89		mA
		VDD = 5V, fcpu = 8MHz		5.45		mA
		VDD = 3V, fcpu = 16MHz		6.76		mA
		VDD = 5V, fcpu = 16MHz		7.32		mA
	Normal mode supply current (CKCON=0x00, 4MHz Crystal)	VDD = 3V, fcpu = 1MHz		2.88		mA
		VDD = 5V, fcpu = 1MHz		3.05		mA
		VDD = 3V, fcpu = 4MHz		3.58		mA
		VDD = 5V, fcpu = 4MHz		3.75		mA

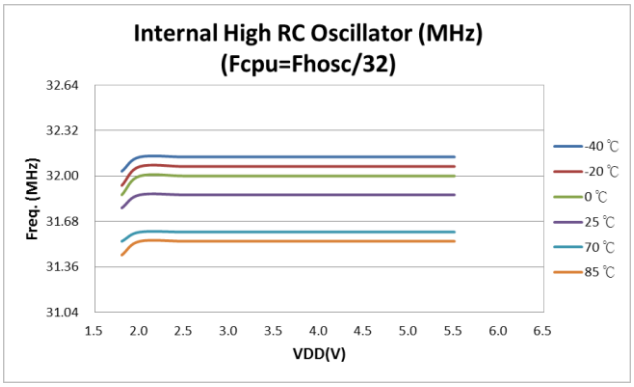
Copyright © 2019, SONiX Technology Co., Ltd.

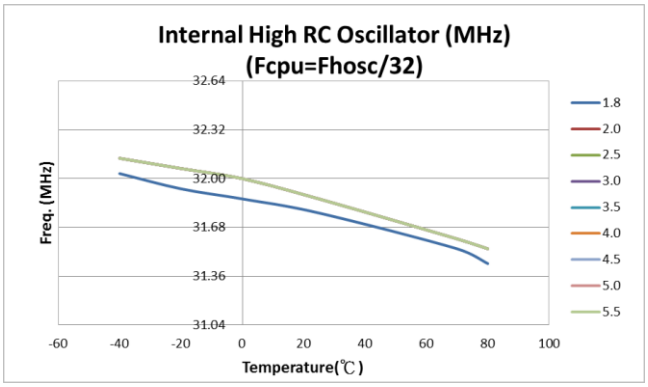
I_{DD2}	STOP mode supply current	VDD = 3V		1.6	5.5	µA
		VDD = 5V		1.9	6.0	µA
I_{DD3}	STOP mode supply current (ILRC enable STWK=1)	VDD = 5V		2.2		µA
I_{DD4}	STOP mode supply current (LCD enable, no panel, VLCD = VDD = 3V)	RLCD = 300KΩ, 32KHz XTAL		6.2		µA
		RLCD = 17.65KΩ, 32KHz XTAL		60.5		µA
		RLCD = 300KΩ, Int. 16K		5.3		µA
		RLCD = 17.65KΩ, Int. 16K		59.6		µA
I_{DD5}	IDLE mode supply current (fcpu = 1MHz)	VDD = 3V, 32MHz IHRC		0.93		mA
		VDD = 5V, 32MHz IHRC		0.94		mA
		VDD = 3V, 16MHz Crystal		0.74		mA
		VDD = 5V, 16MHz Crystal		1.30		mA
		VDD = 3V, 4MHz Crystal		0.37		mA
		VDD = 5V, 4MHz Crystal		0.54		mA
F_{IHRC}	Internal high clock generator	VDD = 1.8V to 5.5V, 25°C	31.84	32	32.16	MHz
		VDD = 1.8V to 5.5V, -40°C to 85°C	31.36	32	32.64	MHz
F_{ILRC}	Internal low clock generator (VDD = 5.0V, 25°C)	Normal mode	12	16	24	kHz
		STOP mode		10		kHz
V_{LVD18}	LVD18 detect voltage	25°C	1.7	1.8	1.9	V
		-40°C to 85°C	1.6	1.8	2.0	V

* 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.





30.3 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	20		mA
I_{OL1}	I/O sink current (P0, P10 – P16, P2, P3, P4, P5, P6)	VDD = 5V, V_O = VSS+0.5V	15	20		mA
I_{OL2}	I/O sink current (P70 – P77)	VDD = 5V, V_O = VSS+1.5V	200	250		mA
I_{OL3}	I/O sink current (P17)	VDD = 5V, V_O = VSS+1.5V	350	400		mA

* Ambient temperature is 25°C.

30.4 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	3.92	4	4.08	V
	Internal 3V reference voltage	VDD = 5V	2.94	3	3.06	V
	Internal 2V reference voltage	VDD = 5V	1.96	2	2.04	V
I_{AD}	ADC current consumption	VDD = 3V		0.41		mA
		VDD = 5V		0.73		mA
f_{ADCLK}	ADC clock	VDD = 5V			16	MHz
f_{ADSMP}	ADC sampling rate	VDD = 5V			250	kHz
t_{ADEN}	ADC function enable period	VDD = 5V	100			μs
DNL	Differential nonlinearity*	f_{ADSMP} = 62.5kHz		±1		LSB
		f_{ADSMP} = 250kHz		±1		LSB
INL	Integral Nonlinearity*	f_{ADSMP} = 62.5kHz		±2		LSB
		f_{ADSMP} = 250kHz		±2.5		LSB
NMC	No missing code*	f_{ADSMP} = 62.5kHz	10	11	12	Bit
		f_{ADSMP} = 250kHz		11		Bit

Copyright © 2019, SONiX Technology Co., Ltd.

V_{OFFSET} Input offset voltage	Non-trimmed	-10	0	10	mV

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

30.5 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=64 bytes, 25°C		6	8	ms

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

31 Instruction Set

This chapter categorizes the SN8F5829 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

Copyright © 2019, SONiX Technology Co., Ltd.

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

Copyright © 2019, SONiX Technology Co., Ltd.

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

Copyright © 2019, SONiX Technology Co., Ltd.

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

Copyright © 2019, SONiX Technology Co., Ltd.

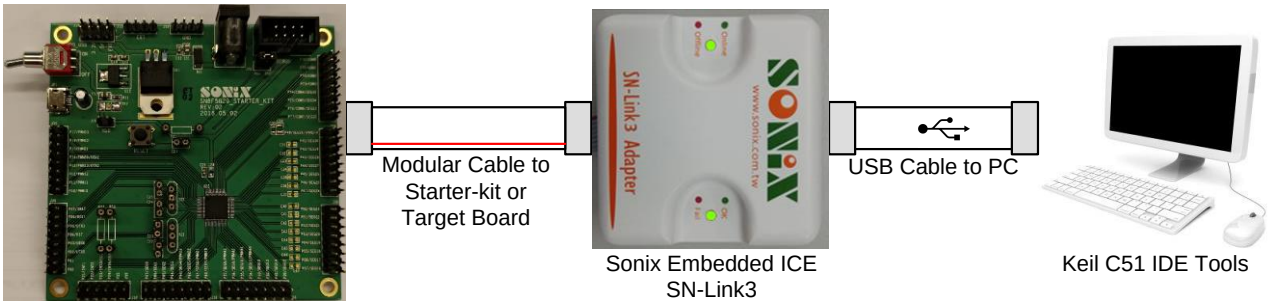
Program branches

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

Copyright © 2019, SONiX Technology Co., Ltd.

32 Development Environment

SONIX provides an Embedded ICE emulator system to offer SN8F5829 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, SN8F5829 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 SN8F5829 to offer a real development environment.



32.1 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.

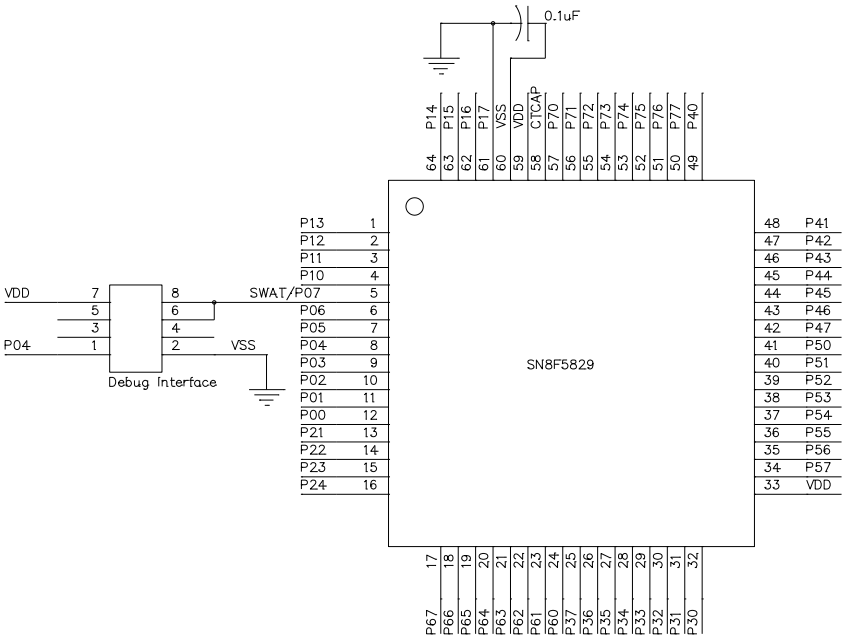
- **SN-Link3 Adapter** with updated firmware version 1.02
- **SN-Link Driver for Keil C51** version 1.00.317
- **Keil C51** version 9.50a and 9.54a or greater.

32.2 Debug Interface Hardware

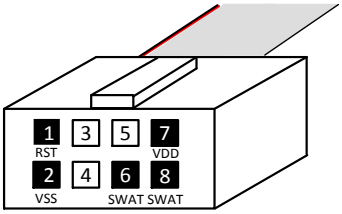
The circuit below demonstrates the appropriate method to connect microcontroller's SWAT pin and SN-Link3 Adapter.

Before starting debug, microcontroller's power (VDD) must be switched off. Connect the SWAT to both 6^{th} and 8^{th} pins of SN-Link, and respectively link VDD and VSS to 7^{th} pin and 2^{nd} 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).

Copyright © 2019, SONiX Technology Co., Ltd.



example circuit

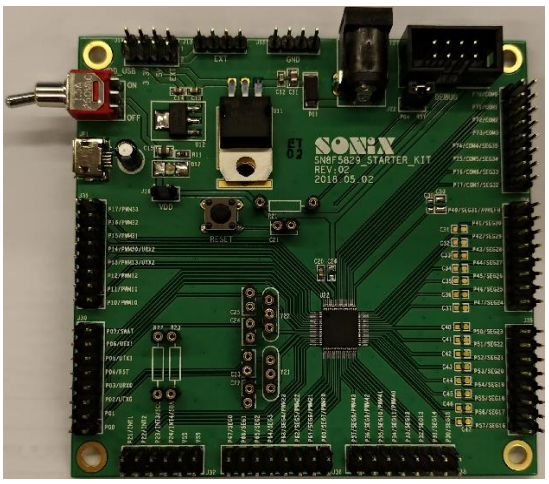


SN-Link header

32.3 Development Tool

SN-Link3 Adapter

Starter-Kit support SN8F5829-SN8F5823



MP5 Writer

33 SN8F5829 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.

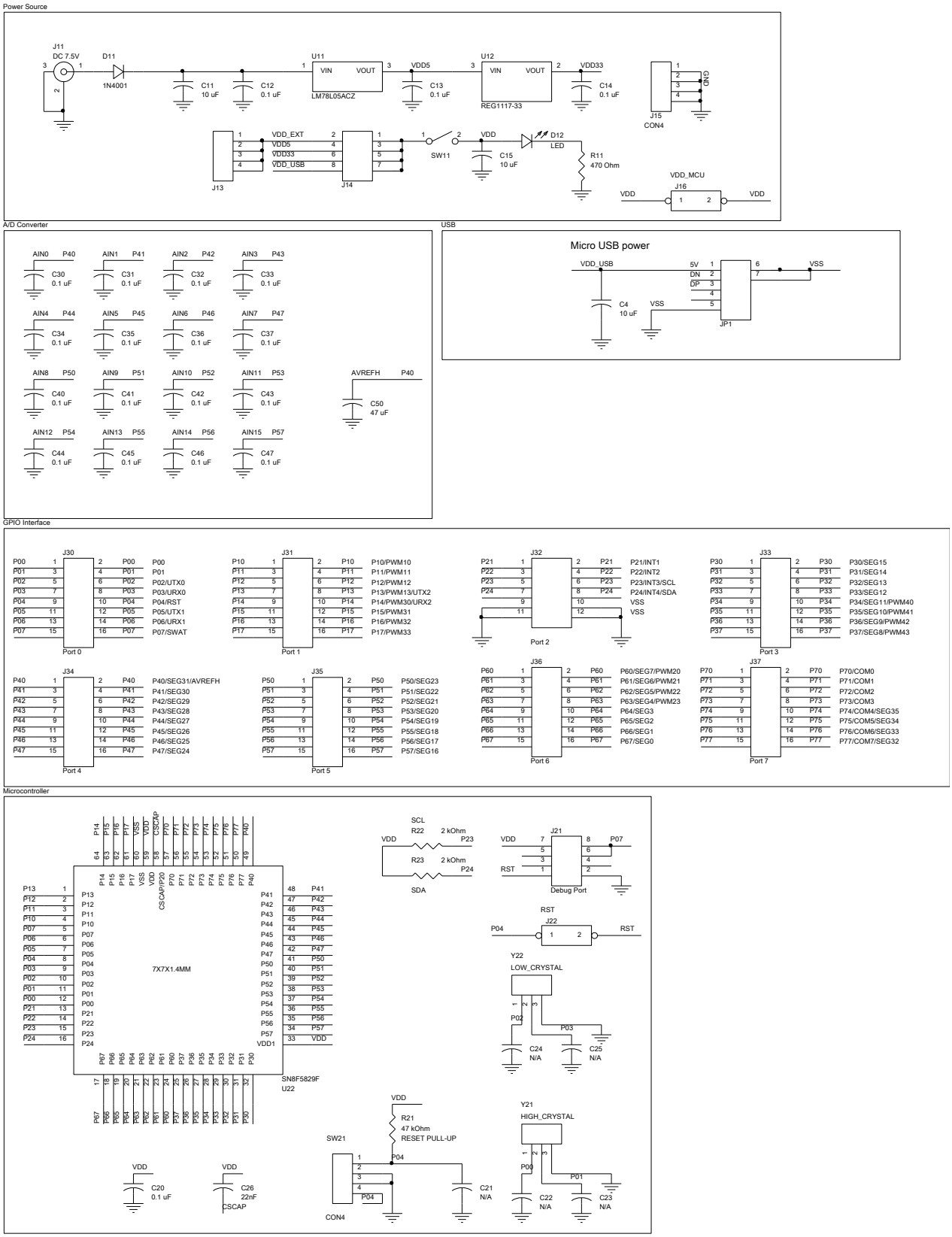
33.1 Configurations of Circuit

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

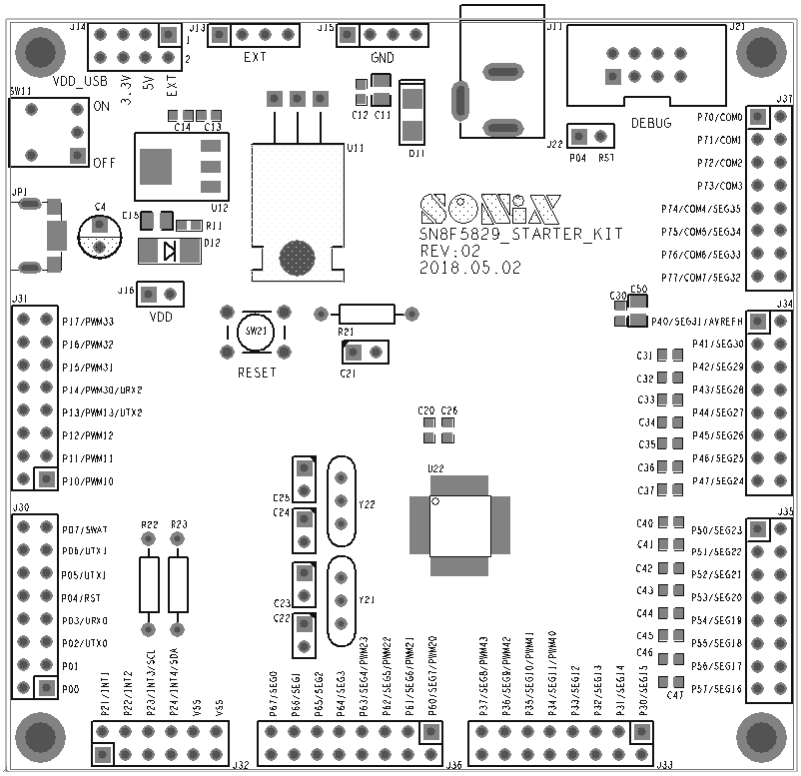
1. Confirm to the circuit board whether elements are complete.
2. The power source of Starter-Kit circuit is chosen from 5.0V, 3.3V, external power or Micro USB via jumper.
3. The power source comes from 5.0V or 3.3V which must be connect to DC 7.5V power adapter.
4. If the power source is chosen from 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.

Copyright © 2019, SONiX Technology Co., Ltd.

33.2 Schematic



33.3 Floor Plan of PCB layout



33.4 Component Description

Number	Description
C30 – C47	16-ch ADC capacitors.
C50	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 – J37	I/O connector.
R21, C21	External reset pull-high resister and capacitor.
R22, R23	I2C pull-high resisters.
SW11	Target power (VDD) switch
U22	SN8F5829F real chip (Sonix standard option).
Y21, C22, C23,	External high crystal/resonator oscillator components.
Y22, C24, C25,	External low crystal/resonator oscillator components.
JP1	Micro USB port

Copyright © 2019, SONiX Technology Co., Ltd.

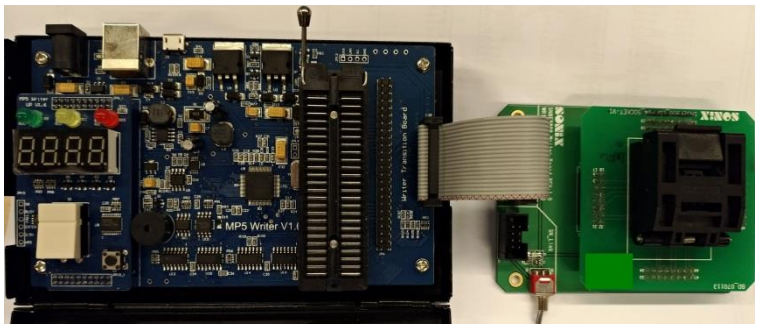
34 ROM Programming Pin

SN8F5829 Series Flash ROM erase/program/verify support SN-Link and MP5 Writer

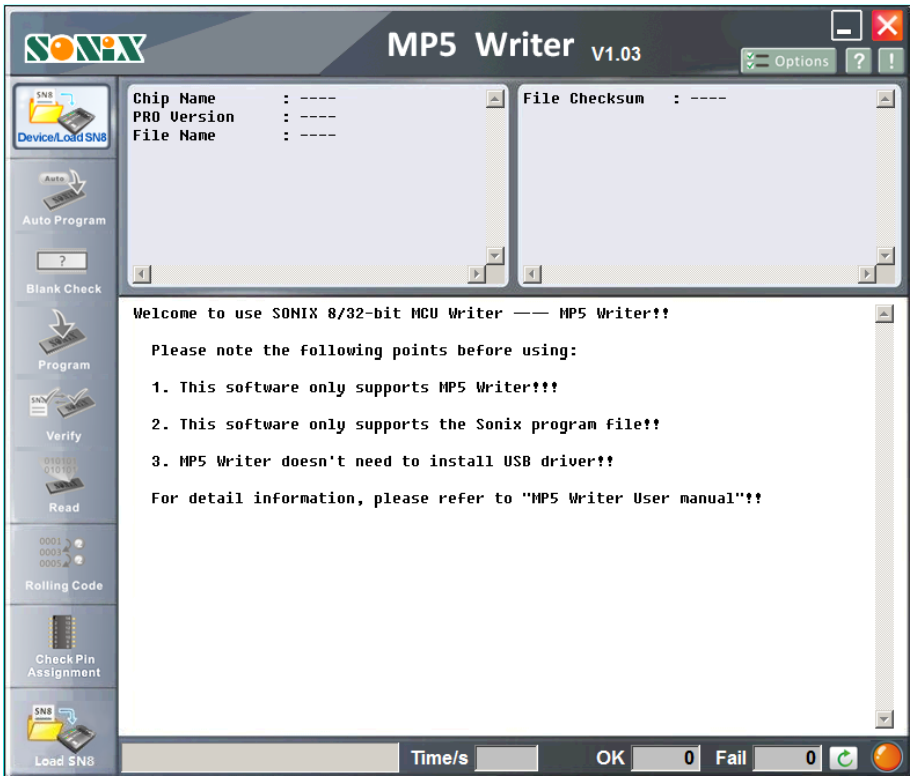
- SN-Link: Debug interface and on board programming.
- MP5 Writer: For SN8F5829 series version mass programming.

34.1 MP5 Hardware Connecting

Different package type with MCU programming connecting is as following, LQFP64 and LQFP48/44 Illustration.



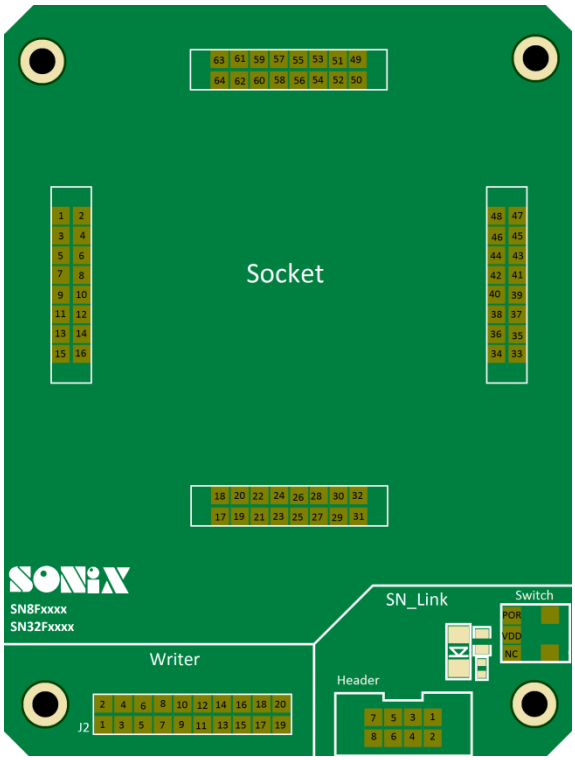
MP5 Software operation interface is as following.



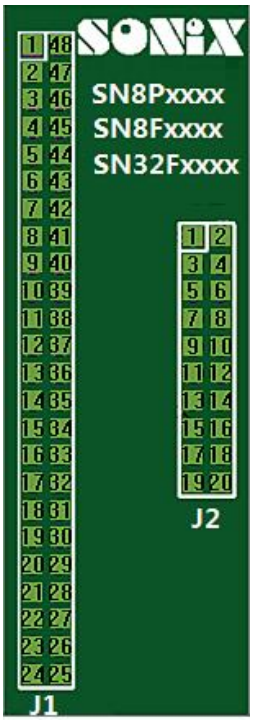
34.2 MP5 Writer Transition Board Socket Pin Assignment

MP5 Writer Transition Board:

For 64 pins



For less than or equal to 48 pins



34.3 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.

Normal mode can meet most programming needs. However, if you want to shorten the programming time, you can use the high speed mode. High speed mode requires a more stable connection environment. Please confirm whether the environment can meet the requirements before use.

Copyright © 2019, SONiX Technology Co., Ltd.

34.3.1 Normal mode

Writer Connector		MCU Pin Number	SN8F5829F		SN8F5828F	
J2 Pin Number	J2 Pin Name		MCU Pin Number	J1 Pin Number	MCU Pin Number	J1 Pin Number
1	VDD	VDD	59	-	43	43
2	GND	VSS	60	-	44	44
7	SWAT	P0.7	5	-	3	3
9	SWAT	P0.7	5	-	3	3
20	PDB	P0.1	11	-	7	7

Writer Connector		MCU Pin Number	SN8F5825F/J		SN8F58252F/J		SN8F58253F/J		SN8F58254F/J	
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	29	37	29	37	29	37	29	37
2	GND	VSS	30	38	30	38	30	38	30	38
7	SWAT	P0.7	32	40	1	9	1	9	1	9
9	SWAT	P0.7	32	40	1	9	1	9	1	9
20	PDB	P0.1	4	12	4	12	4	12	4	12

Writer Connector		MCU Pin Number	SN8F5824S		SN8F58242S		SN8F58243S		SN8F58244S	
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	28	38	28	38	28	38	28	38
2	GND	VSS	1	11	1	11	1	11	1	11
7	SWAT	P0.7	4	14	4	14	4	14	4	14
9	SWAT	P0.7	4	14	4	14	4	14	4	14
20	PDB	P0.1	6	16	7	17	7	17	7	17

Copyright © 2019, SONiX Technology Co., Ltd.

Writer Connector		MCU Pin Number	SN8F5823S	
J2 Pin Number	J2 Pin Name		MCU Pin Number	J1 Pin Number
1	VDD	VDD	24	36
2	GND	VSS	1	13
7	SWAT	P0.7	3	15
9	SWAT	P0.7	3	15
20	PDB	P0.1	5	17

34.3.2 High speed mode

Writer Connector		MCU Pin Number	SN8F5829F		SN8F5828F	
J2 Pin Number	J2 Pin Name		MCU Pin Number	J1 Pin Number	MCU Pin Number	J1 Pin Number
1	VDD	VDD	59	-	43	43
2	GND	VSS	60	-	44	44
3	DFTCLK	P2.4	16	-	12	12
5	SEL	P2.1	13	-	9	9
7	SWAT	P0.7	5	-	3	3
9	SWAT	P0.7	5	-	3	3
11	DAH	P2.2	14	-	10	10
13	DAL	P2.3	15	-	11	11
20	PDB	P0.1	11	-	7	7

Writer Connector		MCU Pin Number	SN8F5825F/J		SN8F58252F/J		SN8F58253F/J		SN8F58254F/J	
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	29	37	29	37	29	37	29	37
2	GND	VSS	30	38	30	38	30	38	30	38
3	DFTCLK	P2.4	9	17	9	17	9	17	9	17
5	SEL	P2.1	6	14	6	14	6	14	6	14
7	SWAT	P0.7	32	40	1	9	1	9	1	9
9	SWAT	P0.7	32	40	1	9	1	9	1	9
11	DAH	P2.2	7	15	7	15	7	15	7	15
13	DAL	P2.3	8	16	8	16	8	16	8	16
20	PDB	P0.1	4	12	4	12	4	12	4	12

Copyright © 2019, SONiX Technology Co., Ltd.

Writer Connector		MCU Pin Number	SN8F5824S		SN8F58242S		SN8F58243S		SN8F58244S	
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	28	38	28	38	28	38	28	38
2	GND	VSS	1	11	1	11	1	11	1	11
3	DFTCLK	P2.4	11	21	12	22	12	22	12	22
5	SEL	P2.1	8	18	9	19	9	19	9	19
7	SWAT	P0.7	4	14	4	14	4	14	4	14
9	SWAT	P0.7	4	14	4	14	4	14	4	14
11	DAH	P2.2	9	19	10	20	10	20	10	20
13	DAL	P2.3	10	20	11	21	11	21	11	21
20	PDB	P0.1	6	16	7	17	7	17	7	17

Writer Connector		MCU Pin Number	SN8F5823S	
J2 Pin Number	J2 Pin Name		MCU Pin Number	J1 Pin Number
1	VDD	VDD	24	36
2	GND	VSS	1	13
3	DFTCLK	P2.4	10	22
5	SEL	P2.1	7	19
7	SWAT	P0.7	3	15
9	SWAT	P0.7	3	15
11	DAH	P2.2	8	20
13	DAL	P2.3	9	21
20	PDB	P0.1	5	17

Copyright © 2019, SONiX Technology Co., Ltd.

34.4 SN-Link ISP Programming

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

34.5 SN-Link ISP Programming Pin Mapping

SN-Link Connector		MCU Pin Number	SN8F5829F	SN8F5828F
Pin Number	Pin Name		Pin Number	Pin Number
7	VDD	VDD	59	43
2	GND	VSS	60	44
6	SWAT	P0.7	5	3
8	SWAT	P0.7	5	3

SN-Link Connector		MCU Pin Number	SN8F5825F/J	SN8F58252F/J	SN8F58253F/J	SN8F58254F/J
Pin Number	Pin Name		Pin Number	Pin Number	Pin Number	Pin Number
7	VDD	VDD	29	29	29	29
2	GND	VSS	30	30	30	30
6	SWAT	P0.7	32	1	1	1
8	SWAT	P0.7	32	1	1	1

Copyright © 2019, SONiX Technology Co., Ltd.

SN-Link Connector		MCU Pin Number	SN8F5824S	SN8F58242S	SN8F58243S	SN8F58244S
Pin Number	Pin Name		Pin Number	Pin Number	Pin Number	Pin Number
7	VDD	VDD	28	28	28	28
2	GND	VSS	1	1	1	1
6	SWAT	P0.7	4	4	4	4
8	SWAT	P0.7	4	4	4	4

SN-Link Connector		MCU Pin Number	SN8F5823S
Pin Number	Pin Name		Pin Number
7	VDD	VDD	24
2	GND	VSS	1
6	SWAT	P0.7	3
8	SWAT	P0.7	3

Copyright © 2019, SONiX Technology Co., Ltd.

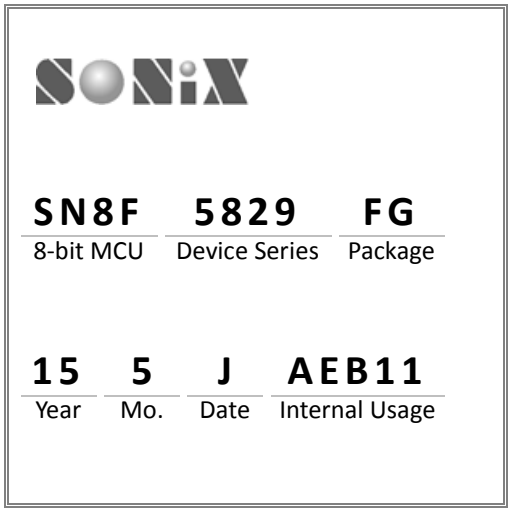
35 Ordering Information

A typical surface of SONiX microcontroller is printed with three columns: logo, device's full name, and date code.

SONiX Logo

Full Name

Date Code



35.1 Device Nomenclature

Full Name	Packing Type
S8F5829W	Wafer
SN8F5829H	Dice
SN8F5829FG	LQFP, 64 pins, Green package
SN8F5828FG	LQFP, 48 pins, Green package
SN8F5825FG	LQFP, 32 pins, Green package
SN8F5825JG	QFN, 32 pins, Green package
SN8F58252FG	LQFP, 32 pins, Green package
SN8F58252JG	QFN, 32 pins, Green package
SN8F58253FG	LQFP, 32 pins, Green package
SN8F58253JG	QFN, 32 pins, Green package
SN8F58254FG	LQFP, 32 pins, Green package
SN8F58254JG	QFN, 32 pins, Green package
SN8F5824SG	SOP, 28 pins, Green package
SN8F58242SG	SOP, 28 pins, Green package
SN8F58243SG	SOP, 28 pins, Green package
SN8F58244SG	SOP, 28 pins, Green package
SN8F5823SG	SOP, 24 pins, Green package

Copyright © 2019, SONiX Technology Co., Ltd.

35.2 Date Code

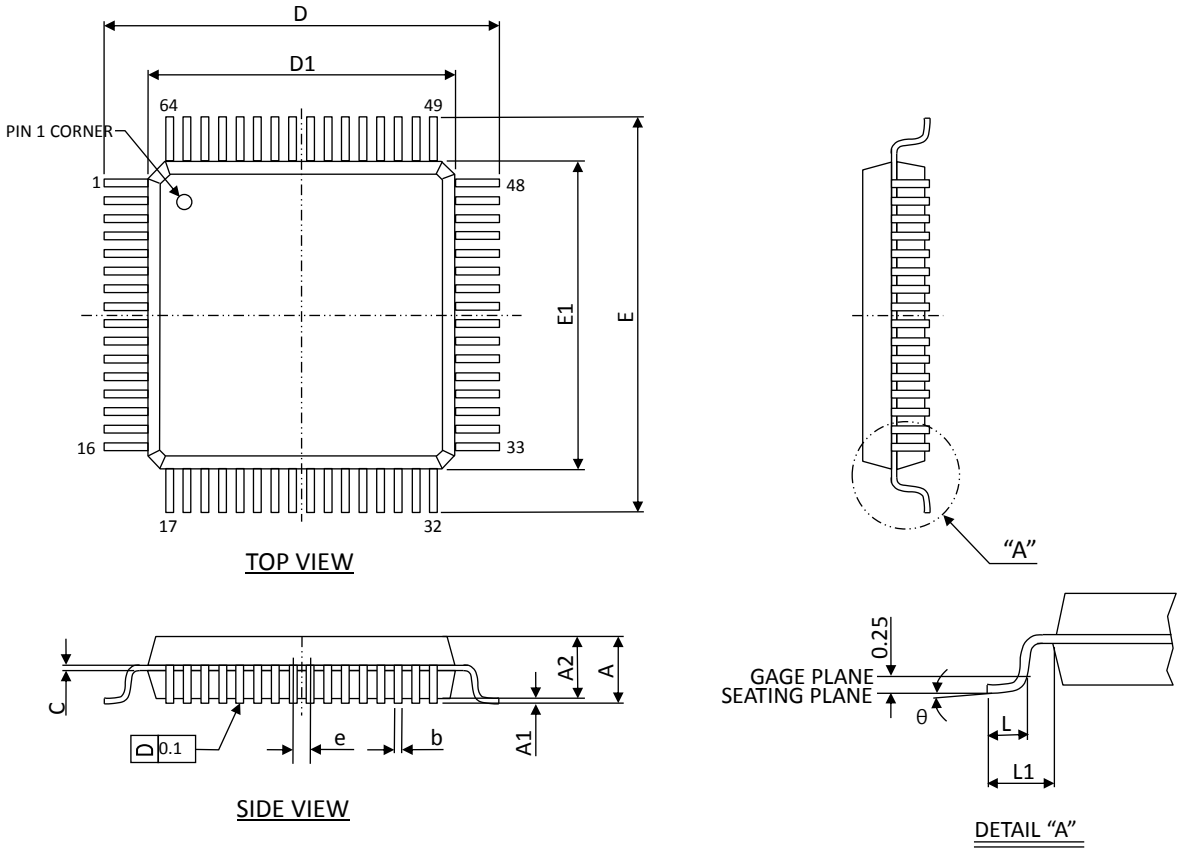
The date code includes two parts: date of manufacture and production serial code. The first part is public information which is encoded by following principles.

Year	15: 2015 16: 2016 17: 2017 et cetera
Month	1: January 2: February 3: March A: October B: November C: December et cetera
Date	1: 01 2: 02 3: 03 A: 10 B: 11 et cetera

Copyright © 2019, SONiX Technology Co., Ltd.

36 Package Information

36.1 LQFP64 7x7



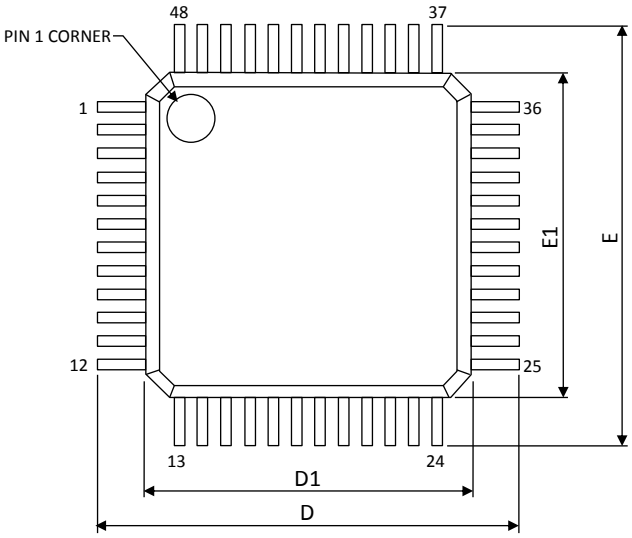
SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	1.60	--	--	0.063
A1	0.05	--	0.25	0.002	--	0.01
A2	1.35	1.40	1.45	0.053	0.055	0.057
b	0.13	0.19	0.25	0.005	0.007	0.010
c	0.09	--	0.20	0.004	--	0.008
D	9.00 BSC			0.354 BSC		
D1	7.00 BSC			0.276 BSC		
e	0.40 BSC			0.016 BSC		
E	9.00 BSC			0.354 BSC		
E1	7.00 BSC			0.276 BSC		
L	0.4	0.60	0.8	0.016	0.024	0.032
L1	1.00 REF			0.039 REF		
θ	0°	3.5°	7°	0°	3.5°	7°

Notes：

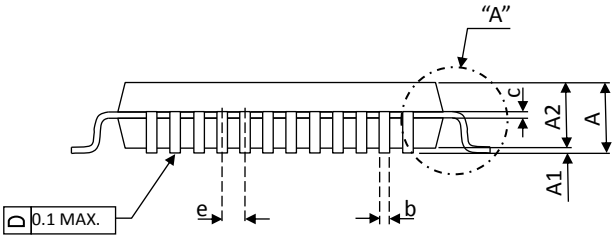
1. CONTROLLING DIMENSION：MILLIMETER (mm)
2. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION.
3. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.

Copyright © 2019, SONiX Technology Co., Ltd.

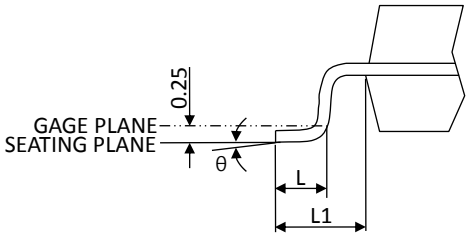
36.2 LQFP48 7x7



TOP VIEW



SIDE VIEW



DETAIL "A"

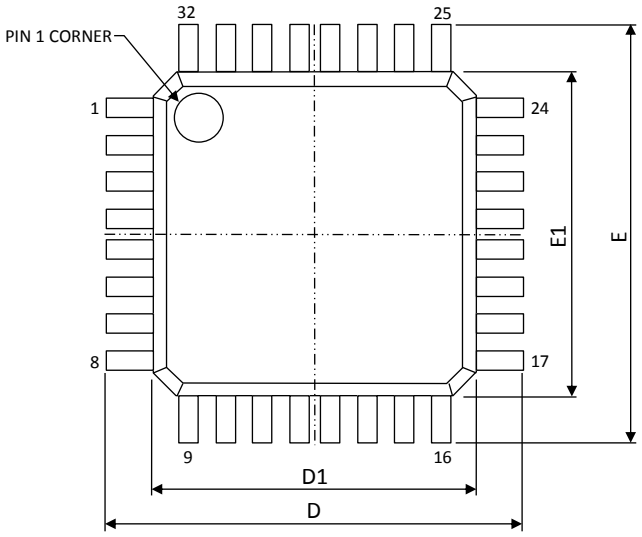
SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	1.60	--	--	0.063
A1	0.05	--	0.15	0.002	--	0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
b	0.17	0.22	0.27	0.007	0.009	0.011
c	0.09	--	0.20	0.004	--	0.008
D	9.00 BSC			0.354 BSC		
D1	7.00 BSC			0.276 BSC		
E	9.00 BSC			0.354 BSC		
E1	7.00 BSC			0.276 BSC		
e	0.50 BSC			0.020 BSC		
L	0.40	0.60	0.80	0.016	0.024	0.031
L1	1.00 REF			0.039 REF		
θ	0°	3.5°	7°	0°	3.5°	7°

Notes：

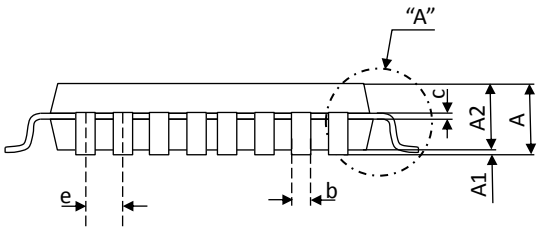
1. CONTROLLING DIMENSION：MILLIMETER (mm)
2. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION.
3. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.

Copyright © 2019, SONiX Technology Co., Ltd.

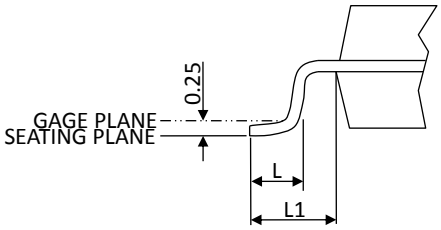
36.3 LQFP32 7x7



TOP VIEW



SIDE VIEW



DETAIL "A"

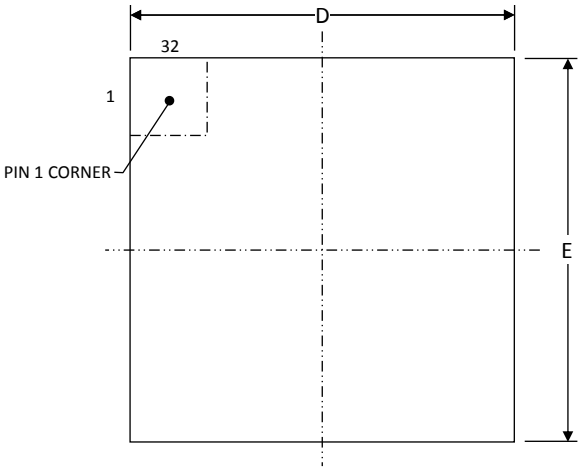
SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	1.60	--	--	0.063
A1	0.05	--	0.25	0.002	--	0.01
A2	1.35	1.40	1.45	0.053	0.055	0.057
b	0.30	--	0.45	0.012	--	0.018
c	0.09	--	0.20	0.004	--	0.008
D	9.00 BSC			0.354 BSC		
D1	7.00 BSC			0.276 BSC		
E	9.00 BSC			0.354 BSC		
E1	7.00 BSC			0.276 BSC		
e	0.80 BSC			0.031 BSC		
L	0.40	0.60	0.80	0.016	0.024	0.031
L1	1.00 REF			0.039 REF		

Notes：

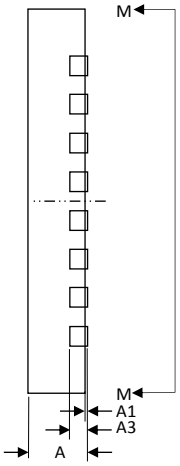
1. CONTROLLING DIMENSION：MILLIMETER (mm)
2. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION.
3. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.

Copyright © 2019, SONiX Technology Co., Ltd.

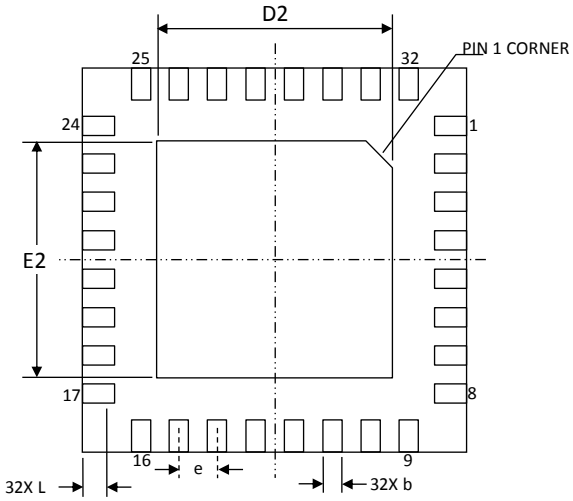
36.4 QFN32 5x5



TOP VIEW



SIDE VIEW



BOTTOM VIEW

VIEW M - M

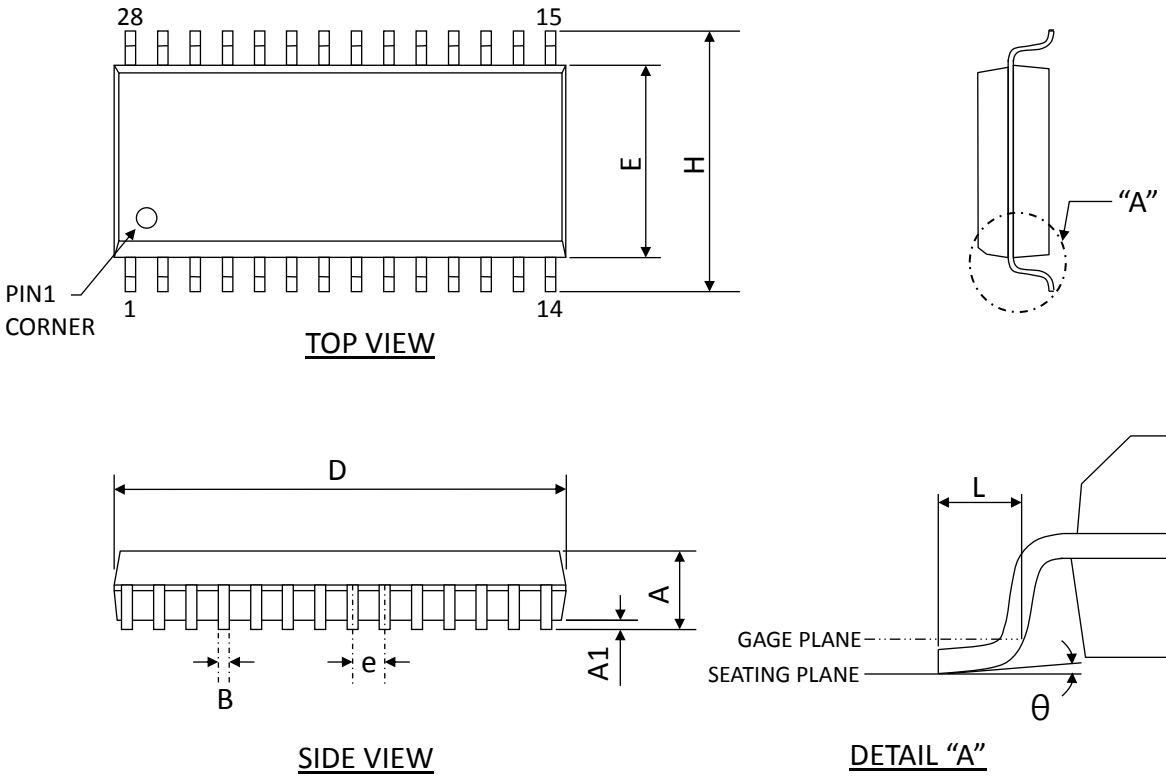
SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	0.70	0.80	0.90	0.028	0.031	0.035
A1	0.00	0.02	0.05	0.000	0.001	0.002
A3	0.203 REF			0.008 REF		
b	0.18	0.25	0.30	0.007	0.010	0.012
D	5.00 BSC			0.197 BSC		
E	5.00 BSC			0.197 BSC		
e	0.50 BSC			0.020 BSC		
D2	2.60	3.10	3.60	0.102	0.122	0.142
E2	2.60	3.10	3.60	0.102	0.122	0.142
L	0.30	0.40	0.50	0.012	0.016	0.020

Notes：

1. CONTROLLING DIMENSION：MILLIMETER (mm)

Copyright © 2019, SONiX Technology Co., Ltd.

36.5 SOP28



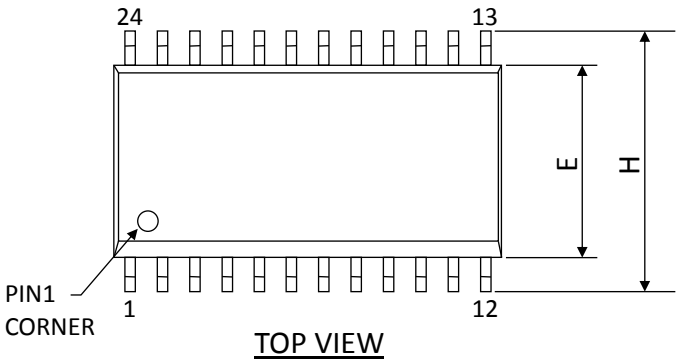
SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	2.65	--	--	0.104
A1	0.10	--	0.30	0.004	--	0.011
B	0.31	0.41	0.51	0.012	0.016	0.020
D	17.70	18.20	18.70	0.697	0.716	0.736
E	7.50 BSC			0.295 BSC		
e	1.27 BSC			0.050 BSC		
H	10.30 BSC			405 BSC		
L	0.40	--	1.27	0.016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

Notes：

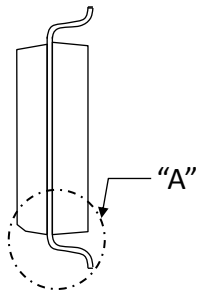
1. CONTROLLING DIMENSION：mm
2. JEDEC OUTLINE : MO-119 AB

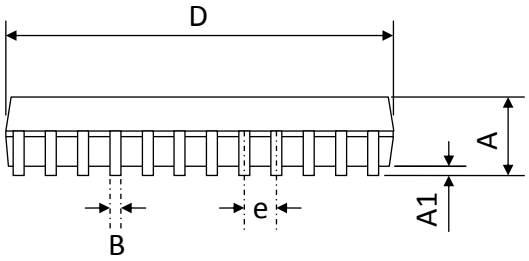
Copyright © 2019, SONiX Technology Co., Ltd.

36.6 SOP24

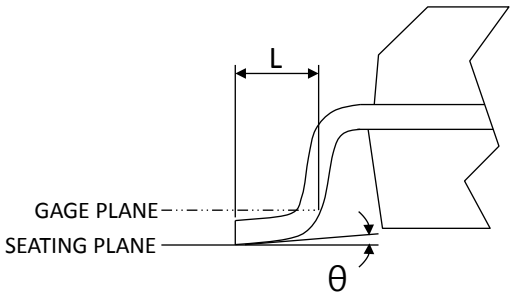


TOP VIEW





SIDE VIEW



DETAIL "A"

SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	2.65	--	--	0.104
A1	0.10	--	0.30	0.004	--	0.011
B	0.31	0.41	0.51	0.012	0.016	0.020
D	15.30	15.50	15.70	0.602	0.610	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

Copyright © 2019, SONiX Technology Co., Ltd.

37 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.

SN8F5829 Series Datasheet

8051-based Microcontroller

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

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

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

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

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

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

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

Copyright © 2019, SONiX Technology Co., Ltd.