1T 8051

8-bit Microcontroller

NuMicro® Family
ML51/ML54/ML56 Series
Technical Reference Manual

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

TABLE OF CONTENTS

List of Figures

List of Tables

1 GENERAL DESCRIPTION

The ML51/ML54/ML56 Series is a Flash embedded 1T 8051-based microcontroller. The instruction set of the ML51/ML54/ML56 Series is fully compatible with the standard 80C51 with performance enhanced.

The ML51/ML54/ML56 Series runs up to 24 MHz. ML51 16KB and 32KB Flash series voltage range from 1.8V to 5.5V. ML51 64KB Flash, ML54 and ML56 series voltage range from 1.8 ~ 3.6V. All ML51/ML54/ML56 Series contains up to 64 Kbytes Flash called APROM for programming code. Flash Supports In-Application-Programming (IAP) function, which enables on-chip firmware updates. Partial Flash can be optionally configured as Data Flash programmed by IAP and read by IAP or MOVC instruction. The ML51/ML54/ML56 Series includes an additional configurable up to 4/3/2/1 Kbytes Flash area called LDROM, in which the Boot Code normally resides for carrying out the In-System-Programming (ISP). To facilitate mass production programming and verification, the Flash is allowed to be programmed and read electronically by parallel Writer/Programmer or In-Circuit-Programming (ICP) with Nu-Link. Once programmed and verified, the programmed code can be protected by the Flash lock mechanism for not being read out by any external programming tool.

The ML51/ML54/ML56 Series provides rich peripherals including 256 bytes of SRAM, 4 Kbytes of auxiliary RAM (XRAM), up to 56 general purpose I/O, two 16-bit Timers/Counters 0/1, one 16-bit Timer2 with three-channel input capture module, one Watchdog Timer (WDT), one Self Wake-up Timer (WKT), one 16-bit auto-reload Timer3 for general purpose or baud rate generator, two UARTs with frame error detection and automatic address recognition, two ISO7816 Smartcard interface, two SPI, two I^2C, 6 enhanced PWM output channels with dead zone control, 3 sets of 2 channels PWM output channels with 3 individual configurable period, two analog comparators, eight-channel shared pin interrupt for all I/O ports, one 12-bit ADC at 500 ksps, one RTC offers programmable time tick and alarm match interrupts, equipped with LCD driver that can directly drive the LCD panel with 4 COM x 32 SEG , 6 COM x 30 SEG or 8 COM x 28 SEG and support total 14 keys + reference pad + shielding electrode touch key solution.There are a total of 31 sources with 4-level-priority interrupts capability.

The ML51/ML54/ML56 Series is equipped with four clock sources and supports on-the-fly clock switching via software control. The four clock sources include two sets of external crystal inputs (HXT, LXT), 38.4 kHz internal oscillator, and one 24 MHz internal high-precision ±5% oscillator. The ML51/ML54/ML56 Series provides additional power monitoring detection such as power-on reset and 7-level brown-out detection, which stabilizes the power-on/off sequence for a high reliability system design.

The ML51/ML54/ML56 Series microcontroller provides 3 power modes to reduce power consumption — Low power run mode, Low power Idle mode, and Power-down mode. In Low power run mode, the power consumption can be down to 15 uA at 38.4 kHz LIRC. In Low power idle mode, CPU processing is suspended by holding the Program Counter. No program code is fetched and run in low power idle mode if the power consumption does not exceed 13 uA. Power-down mode stops the whole system clock for minimum power consumption with the leakage current less than 1 uA. The system clock of the ML54 series can also be slowed down by software clock divider, which allows for flexibility between execution performance and power consumption.

Through the high performance of 1T 8051 core, low power performance of ML51/ML54/ML56 Series and rich well-designed peripherals, this series benefits for low-power, battery powered devices, general purpose, home appliances, or motor control system.

Series	V_{DD} Voltage	LCD Driver	Touch Key
ML51 32/16KB Flash Series	1.8 ~ 5.5 V	-	-
ML51 64KB Flash Series	1.8 ~3.6 V	-	-
ML54 Series	1.8 ~3.6 V	√	-
ML56 Series	1.8 ~3.6 V	√	√

2 FEATURES

Core and System	
8051	• Fully static design 8-bit high performance 1T 8051-based CMOS microcontroller. • Instruction set fully compatible with MCS-51. • 4-priority-level interrupts capability. • Dual Data Pointers (DPTRs).
Power on Reset (POR)	• POR with 1.55V threshold voltage level
Brown-out Detector (BOD)	• 7-level selection, with brown-out interrupt and reset option. (4.4V / 3.7V / 3.0V / 2.7V / 2.4V / 2.0V / 1.8V)
Low Voltage Reset (LVR)	• LVR with 1.63V threshold voltage level
Security	• 96-bit Unique ID (UID) • 128-bit Unique Customer ID (UCID) • 128-bytes security protection memory SPROM
Memories	
Flash	• Up to 64 KBytes of APROM for User Code. • 4/3/2/1 Kbytes of Flash for loader (LDROM) configure from APROM for In-System-Programmable (ISP) • Flash Memory accumulated with pages of 128 Bytes from APROM by In-Application-Programmable (IAP) means whole APROM can be use as Data Flash • An additional 128 bytes security protection memory SPROM • Code lock for security by CONFIG
SRAM	• 256 Bytes on-chip RAM. • Additional 4 KBytes on-chip auxiliary RAM (XRAM) accessed by MOVX instruction.
PDMA:	• Three modes: peripheral-to-memory, memory-to-peripheral, and memory-to-memory transfer. • Source address and destination address must be word alignment in all modes. • Memory-to-memory mode: transfer length must be word alignment.

Clocks	
External Clock Source	• 4~24 MHz High-speed external crystal oscillator (HXT) for precise timing operation • 32.768 kHz High-speed external crystal oscillator (LXT) for RTC operation
Internal Clock Source	• Default 24 MHz high-speed internal oscillator (HIRC) trimmed to ±1% (accuracy at 25 ˚C, 3.3 V), ±2% in -20~105˚C. • 38.4 kHz low-speed internal oscillator (LIRC) calibrating to ±2% by software from high-speed internal oscillator
Timers	
16-bit Timer	• Two 16-bit Timers/Counters 0 and 1 compatible with standard 8051. • One 16-bit Timer 2 with three-channel input capture module and 9 input pin can be selected. • One 16-bit auto-reload Timer 3, which can be the baud rate clock source of UARTs.
Watchdog	• 6-bit free running up counter for WDT time-out interval. • Selectable time-out interval is 1.66 ms ~ 3413.12 ms since WDT_CLK = 38.4 kHz (LIRC). • Able to wake up from Power-down or Idle mode • Interrupt or reset selectable on watchdog time-out
Wake-up Timer	• 8-bit free running up counter for time-out interva for ML51 32KB / 16KB flash series. • 16-bit free running up counter for time-out interva for ML56 / ML54 / ML51 64KB flash series. • Clock sources from LIRC or LXT. • Able self Wake-up wake up from Power-down or Idle mode, and auto reload count value. • Supports Interrupt
PWM	• Up To 12 output pins can be selected • Supports maximum clock source frequency up to 24 MHz • Supports up to Three PWM modules, each module provides 6 output channels. • Supports independent mode for PWM output • Supports complementary mode for 3 complementary paired PWM output channels • Dead-time insertion with 8-bit resolution • Supports 16-bit resolution PWM counter

	• Supports mask function and tri-state enable for each PWM pin • Supports brake function • Supports trigger ADC on the following events
RTC	• Supports real time counter and calendar counter for RTC time and calendar check. • Supports alarm time and calendar settings • Supports alarm time and calendar mask enable settings. • Selectable 12-hour or 24-hour time scale setting. • Supports Leap Year indication setting. • Supports Day of the Week counter setting. • Frequency of RTC clock source compensate by RTC_FREQADJ register. • All time and calendar message expressed in BCD format. • Supports periodic RTC Time Tick interrupt with 8 period interval options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second. • Supports RTC Time Tick and Alarm Match interrupt. • Supports chip wake-up from Idle or Power-down mode while a RTC interrupt signal is generated. • Support clock source selectable from LXT or LIRC.
Analog Interfaces	
Analog-to-Digital Converter (ADC)	• Analog input voltage range: 0 ~ AV_{DD}. • External or internal Voltage reference input selectable. • 12-bit resolution and 10-bit accuracy is guaranteed. • Up to 16 single-end analog input channels • 1 internal channels, they are band-gap voltage (VBG). • Maximum ADC peripheral clock frequency is 1 MHz. • Up to 500 KSPS sampling rate. • Software Write 1 to ADCS bit to trig ADC start. • External pin (STADC) trigger • PWM trigger.
Communication Interfaces	
UART	• Supports up to 2 UARTs: UART0, UART1 • Supports 2 Smart Card configuration as UART function as UART2 and UART3. • UART baud rate clock from HIRC or HXT. • Full-duplex asynchronous communications

	• Programmable 9th bit. • TXD and RXD pins of UART0 exchangeable via software.
I²C	• 2 sets of I²C devices • Master/Slave mode • Bidirectional data transfer between masters and slaves • Multi-master bus (no central master) • 7-bit addressing mode • Standard mode (100 kbps) and Fast mode (400 kbps). • Supports 8-bit time-out counter requesting the I²C interrupt if the I²C bus hangs up and timer-out counter overflows • Multiple address recognition (four slave addresses with mask option) • Supports hold time programmable
SPI	• 2 sets of SPI devices • Supports Master or Slave mode operation • Supports MSB first or LSB first transfer sequence • Slave mode up to 12 Mhz
ISO 7816-3	• Two sets ISO 7816-3 device • Supports ISO 7816-3 compliant T=0, T=1 • Supports full-duplex UART mode.
GPIO	• Four I/O modes: • Quasi-bidirectional mode • Push-Pull Output mode • Open-Drain Output mode • Input only with high impendence mode • Schmitt trigger input / TTL mode selectable. • Each I/O pin configured as interrupt source with edge/level trigger setting • Standard interrupt pins INT0 and INT1. • Supports high drive and high sink current I/O • I/O pin internal pull-up or pull-down resistor enabled in input mode. • Maximum I/O Speed is 24 MHz • Enabling the pin interrupt function will also enable the wake-up function • Supports 5V-tolerance function for ML51 64KB Flash/ML54/ML56 Series
LCD Driver	• Support Internal resistor bias

	• Support programmable internal VLCD charge pump mode • 1/2, 1/3, 1/4 bias selectable • 4 COM x 32 SE.G. 6 COM x 30 SE.G. 8 COM x 28 SEG • Support 1.8V to 5.5V LCD operating voltage
Touch Key	• Supports up to 14 touch keys + reference pad + shielding electrode • Supports any TK pin as reference pad and any one of CLKO pin as shielding electrode. • Programmable sensitivity levels for each channel. • Programmable scanning speed for different applications. • Supports effect when in Power-down mode. • Supports single key-scan and programmable periodic key-scan. • Programmable interrupt options for key-scan complete withardwareithout threshold control.
ESD & EFT	
ESD	• HBM 8 kV for ML51 32KB/16KB Flash Series pass • HBM 7 kV for ML51 64KB Flash/ML54/ML56 Series pass
EFT	• > ± 4.4 kV
Latch-up	• 150 mA for ML51 32KB/16KB Flash Series pass • 200 mA for ML51 64KB Flash/ML54/ML56 Series pass

3 PART INFORMATION

3.1 ML51/ML54/ML56 Series Package Type

Package	ML51			ML54	ML56
	ML51xB	ML51xC	ML51xD	ML54xD	ML56xD
MSOP10	ML51BB9AE				
TSSOP14	ML51DB9AE				
TSSOP20	ML51FB9AE				
SOP20	ML51OB9AE				
QFN20(3x3)	ML51XB9AE				
TSSOP28	ML51EB9AE	ML51EC0AE			
SOP28	ML51UB9AE	ML51UC0AE			
LQFP32	ML51PB9AE	ML51PC0AE			
QFN33(4x4)	ML51TB9AE	ML51TC0AE	ML51TD1AE		
LQFP44				ML54MD1AE	ML56MD1AE
LQFP48			ML51LD1AE	ML54LD1AE	ML56LD1AE
LQFP64			ML51SD1AE	ML54SD1AE	ML56SD1AE

3.2 ML51/ML54/ML56 Series Selection Guide

3.2.1 ML51 Series

ML51 16KB Flash Series

Part Number		ML51							
		BB9AE	DB9AE	FB9AE	OB9AE	XB9AE	EB9AE	UB9AE	PB9AE
Flash (KB)		16	16	16	16	16	16	16	16
SRAM (KB)		1	1	1	1	1	1	2	2
ISP ROM (KB)		4	4	4	4	4	4	4	4
SPROM (bytes)		128	128	128	128	128	128	128	128
System Frequency (MHz)		24	24	24	24	24	24	24	24
GPIO		7	11	16	16	17	24	24	28
16-bit Timer		4	4	4	4	4	4	4	4
PWM		5	6	6	6	6	6	6	6
Analog Comparator		-	-	-	-	-	-	2	2
Internal Voltage Reference		-	-	-	-	-	-	Y	Y
PDMA		2	2	2	2	2	2	2	2
RTC		-	-	-	-	-	-	-	-
LCD		-	-	-	-	-	-	-	-
Connectivity	ISO 7816-3	-	1	1	1	1	1	1	1
	UART	2	2	2	2	2	2	2	2
	SPI	-	1	1	1	1	1	1	1
	I^2C	1	2	2	2	2	2	2	2
12-bit SAR ADC		2	3	6	6	6	8	8	8
Package		MSOP10	TSSOP14	TSSOP20	SOP20	QFN20	TSSOP28	SOP28	LQFP32

Note:
1. ISP ROM programmable 1K/2K/3K/4KB Flash for user program loader (LDROM) share from ARPOM.
2. ISO 7816-3 configurable as standard UART function.

ML51 32KB Flash Series

Part Number		ML51					
		EC0AE	UC0AE	PC0AE	TC0AE	TC1AE	LC1AE
Flash (KB)		32	32	32	32	32	32
SRAM (KB)		2	2	2	2	2	2
ISP ROM (KB)		4	4	4	4	4	4
SPROM (bytes)		128	128	128	128	128	128
System Frequency (MHz)		24	24	24	24	24	24
GPIO		24	24	28	28	28	43
16-bit Timer		4	4	4	4	4	4
PWM		6	6	6	6	6	6
Analog Comparator		2	2	2	2	2	2
Internal Voltage Reference		Y	Y	Y	Y	Y	Y
PDMA		2	2	2	2	2	2
RTC		-	-	-	-	-	-
LCD		-	-	-	-	-	-
Connectivity	ISO 7816-3	1	1	1	1	2	2
	UART	2	2	2	2	2	2
	SPI	2[3]	2[3]	2	2	2	2
	I^2C	2	2	2	2	2	2
12-bit SAR ADC		8	8	8	8	9	10
Package		TSSOP28	SOP28	LQFP32	QFN33	QFN33	LQFP48

Note:
1. ISP ROM programmable 1K/2K/3K/4KB Flash for user program loader (LDROM) share from ARPOM.
2. ISO 7816-3 configurable as standard UART function.
3. SPI0 and SPI1 share with same SS pin in 28pin package.

ML51 64KB Flash Series

Part Number		ML51		
		TD1AE	LD1AE	SD1AE
Flash (KB)		64	64	64
SRAM (KB)		4	4	4
ISP ROM (KB)		4	4	4
SPROM (bytes)		128	128	128
System Frequency (MHz)		24	24	24
GPIO		28	43	56
16-bit Timer		4	4	4
PWM		6+2+2+2	6+2+2+2	6+2+2+2
Analog Comparator		2	2	2
Internal Voltage Reference		Y	Y	Y
PDMA		4	4	4
RTC		Y	Y	Y
LCD		-	-	-
Connectivity	ISO 7816-3	2	2	2
	UART	2	2	2
	SPI	2	2	2
	I^2C	2	2	2
12-bit SAR ADC		9	10	14
Package		QFN33	LQFP48	LQFP64

Note:

1. ISP ROM programmable 1K/2K/3K/4KB Flash for user program loader (LDROM) share from ARPOM.
2. ISO 7816-3 configurable as standard UART function.

3.2.2 ML54 Series

Part Number		ML54		
		MD1AE	LD1AE	SD1AE
Flash (KB)		64	64	64
SRAM (KB)		4	4	4
ISP ROM (KB)		4	4	4
SPROM (bytes)		128	128	128
System Frequency (MHz)		24	24	24
GPIO		38	42	55
16-bit Timer		4	4	4
PWM		6+2+2+2	6+2+2+2	6+2+2+2
Analog Comparator		2	2	2
Internal Voltage Reference		Y	Y	Y
PDMA		4	4	4
RTC		Y	Y	Y
LCD		8x17	8x18	8x28
		6x19	6x20	6x30
		4x21	4x22	4x32
Connectivity	ISO 7816-3	2	2	2
	UART	2	2	2
	SPI	2	2	2
	I^2C	2	2	2
12-bit SAR ADC		10	10	14
Package		LQFP44	LQFP48	LQFP64

Note:
1. ISP ROM programmable 1K/2K/3K/4KB Flash for user program loader (LDROM) share from ARPOM.
2. ISO 7816-3 configurable as standard UART function.

3.2.3 ML56 Series

Part Number		ML56		
		MD1AE	LD1AE	SD1AE
Flash (KB)		64	64	64
SRAM (KB)		4	4	4
ISP ROM (KB)		4	4	4
SPROM (bytes)		128	128	128
System Frequency (MHz)		24	24	24
GPIO		38	42	55
16-bit Timer		4	4	4
PWM		6+2+2+2	6+2+2+2	6+2+2+2
Analog Comparator		2	2	2
Internal Voltage Reference		Y	Y	Y
PDMA		4	4	4
RTC		Y	Y	Y
LCD		8x17	8x18	8x28
		6x19	6x20	6x30
		4x21	4x22	4x32
Touch Key		6	9	14
Connectivity	ISO 7816-3	2	2	2
	UART	2	2	2
	SPI	2	2	2
	I^2C	2	2	2
12-bit SAR ADC		10	10	14
Package		LQFP44	LQFP48	LQFP64

Note:
1. ISP ROM programmable 1K/2K/3K/4KB Flash for user program loader (LDROM) share from ARPOM.
2. ISO 7816-3 configurable as standard UART function.
3.Touch key should define 1 key as reference pin.

3.3 ML51/ML54/ML56 Series Selection Code

ML	51	F	B	9	A	E
Core	**Line**	**Package**	**Flash**	**SRAM**	**Reserve**	**Temperature**
1T 8051 Low power	51: Base 54: LCD 56: Touch	B: MSOP10 (3x3 mm) D: TSSOP14 (4.4x5.0 mm) E: TSSOP28 (4.4x9.7 mm) F: TSSOP20 (4.4x6.5 mm) L: LQFP48 (7x7 mm) M: LQFP44(10x10 mm) O: SOP20 (300 mil) P: LQFP32 (7x7 mm) S: LQFP64 (7x7 mm) T: QFN33 (4x4 mm) U: SOP28 (300 mil) X: QFN20 (3x3mm)	A: 8 KB B: 16 KB C: 32 KB D: 64 KB	0: 2 KB 1: 4 KB 2: 8/12 KB 3: 16 KB 6: 32 KB 8: 64 KB 9: 1 KB A: 96 KB		E:-40 ~ 105° C

4 PIN CONFIGURATION

4.1 Pin Configuration

Users can find pin configuaration informations in chapter 4 or by using NuTool - PinConfigure. The NuTool - PinConfigure contains all Nuvoton NuMicro® Family chip series with all part number, and helps users configure GPIO multi-function correctly and handily.

4.1.1 ML51/ML54/ML56 Series Pin Diagram

4.1.1.1 LQFP64 Package

Corresponding Part Number: ML51SD1AE/ ML54SD1AE / ML56SD1AE

ML51SD1AE

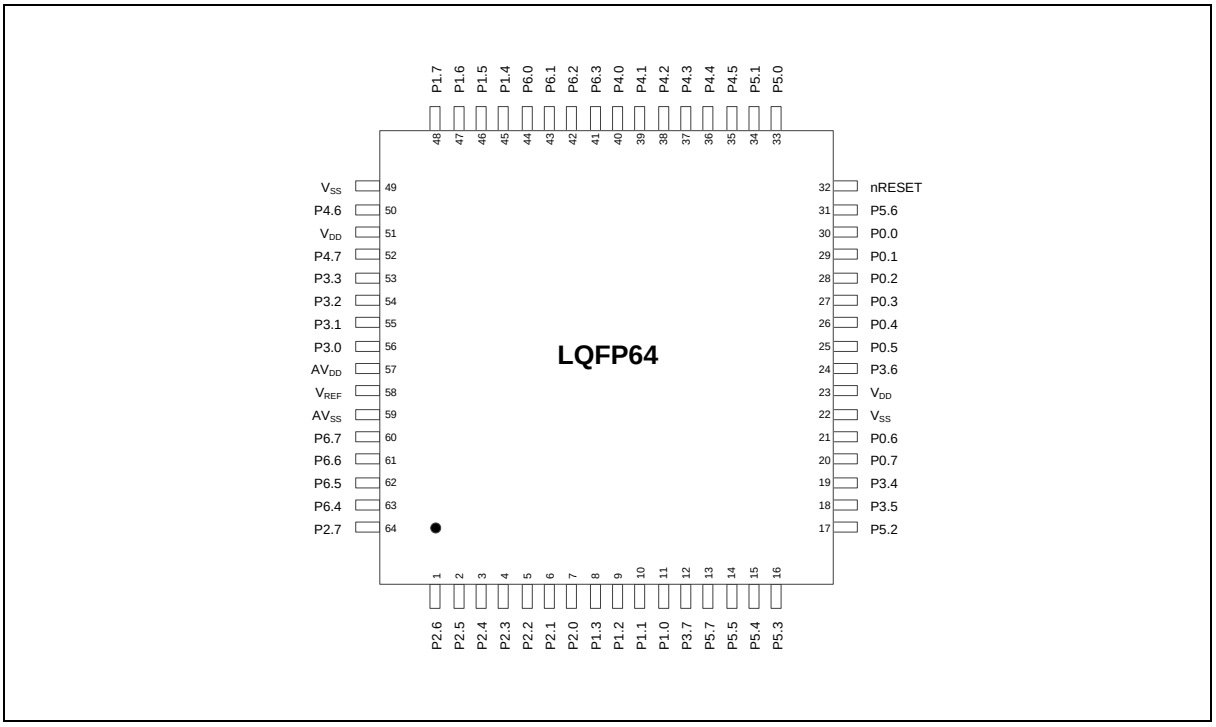


Figure 4.1-1 ML51SD1AE Pin Assignment

ML54SD1AE / ML56SD1AE

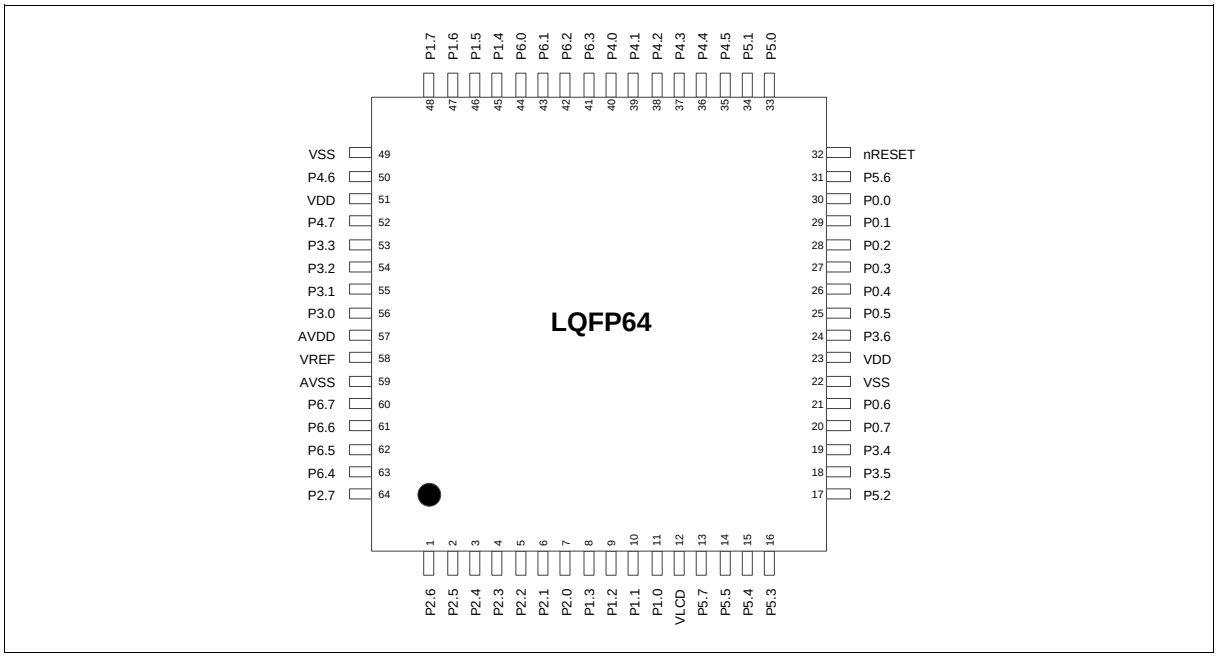


Figure 4.1-2 ML54SD1AE / ML56SD1AE Pin Assignment

4.1.1.2 LQFP48 Package

Corresponding Part Number: ML51LD1AE/ ML54LD1AE / ML56LD1AE

ML51LD1AE

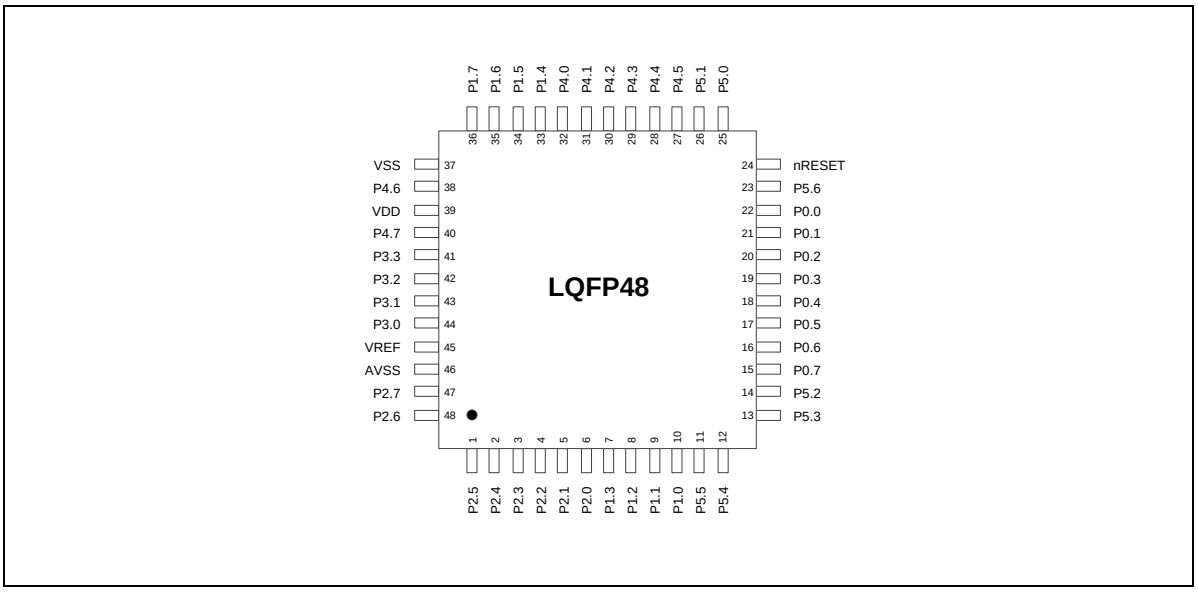


Figure 4.1-3 ML51LD1AE Pin Assignment

ML54LD1AE / ML56LD1AE

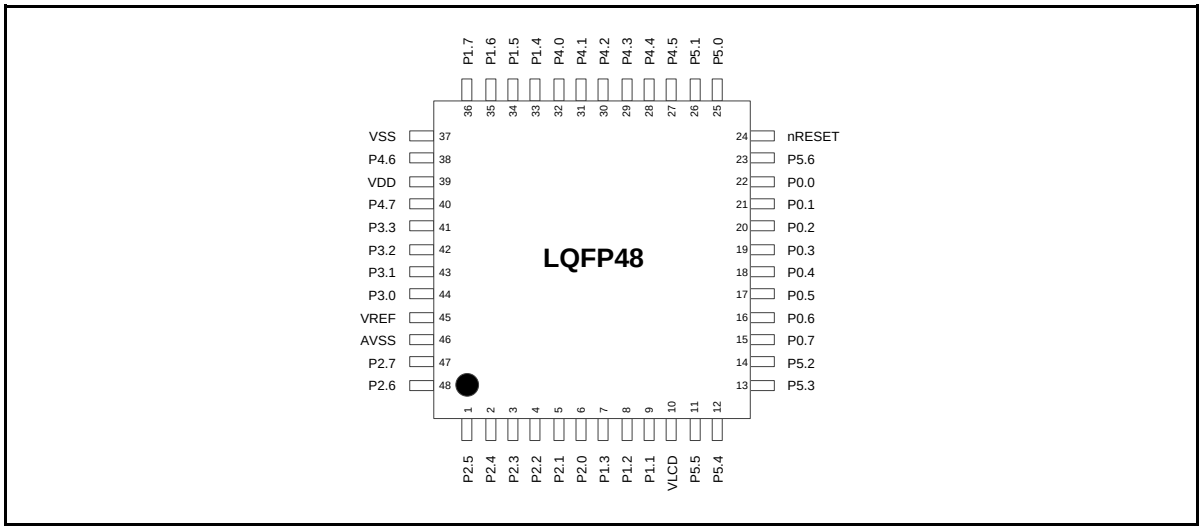


Figure 4.1-4 ML54LD1AE / ML56LD1AE Pin Assignment

4.1.1.3 LQFP44 Package

Corresponding Part Number: ML54MD1AE / ML56MD1AE

ML54MD1AE / ML56MD1AE

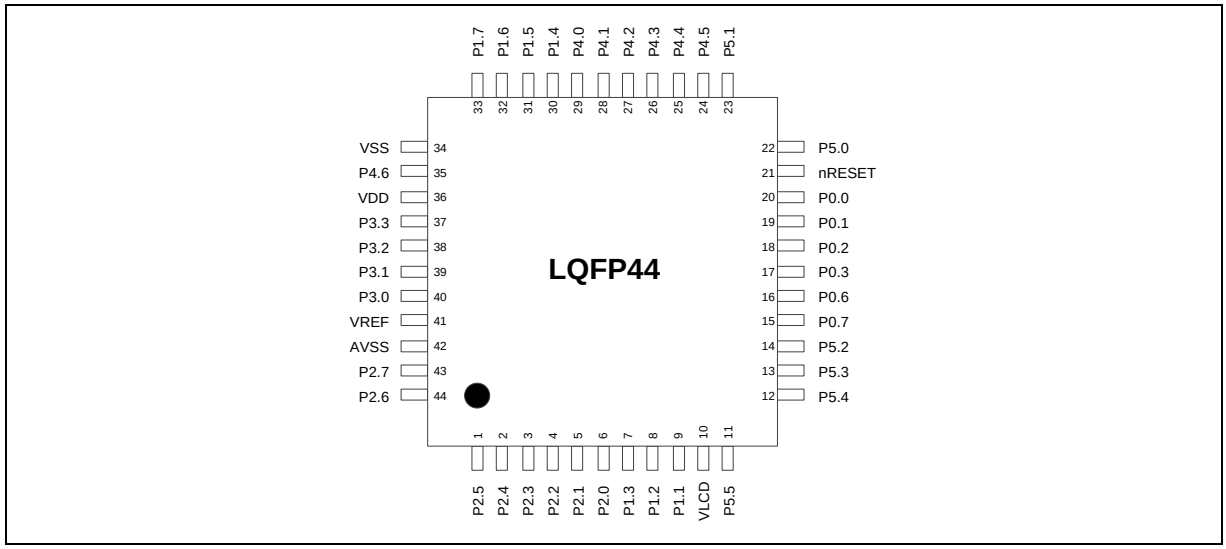


Figure 4.1-5 ML54MD1AE / ML56MD1AE Pin Assignment

4.1.1.4 QFN33 Package

Corresponding Part Number: ML51TD1AE / ML51TC0AE / ML51TB9AE

ML51TD1AE / ML51TC0AE / ML51TB9AE

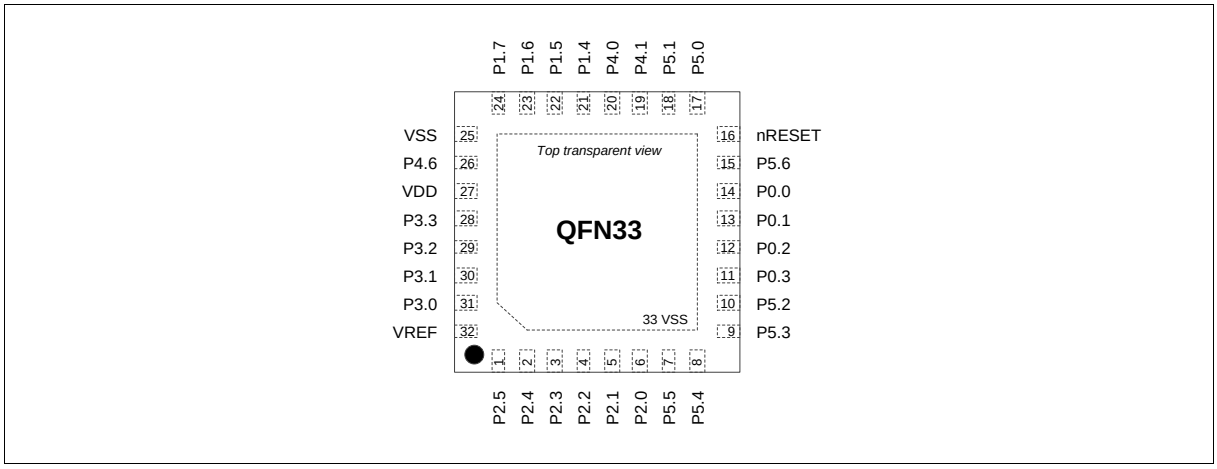


Figure 4.1-6 ML51TD1AE / ML51TC0AE / ML51TB9AE Pin Assignment

4.1.1.5 LQFP32 Package

Corresponding Part Number: ML51PC0AE / ML51PB9AE

ML51PC0AE / ML51PB9AE

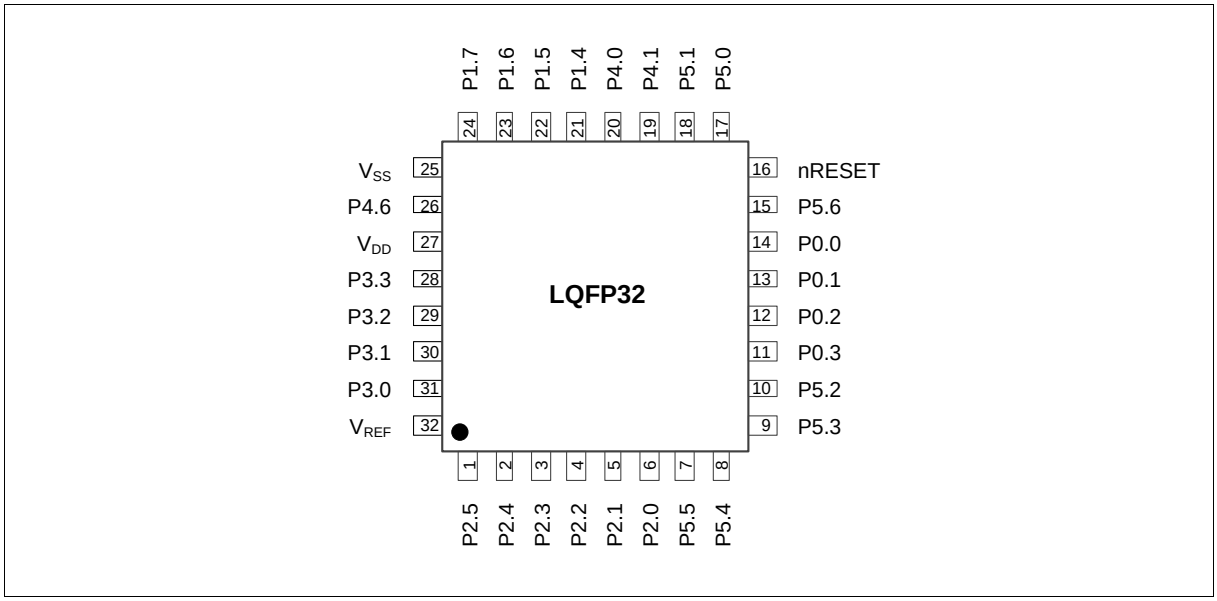


Figure 4.1-7 ML51PC0AE / ML51PB9AE Pin Assignment

4.1.1.6 TSSOP28 Package

Corresponding Part Number: ML51EC0AE / ML51EB9AE

ML51EC0AE / ML51EB9AE

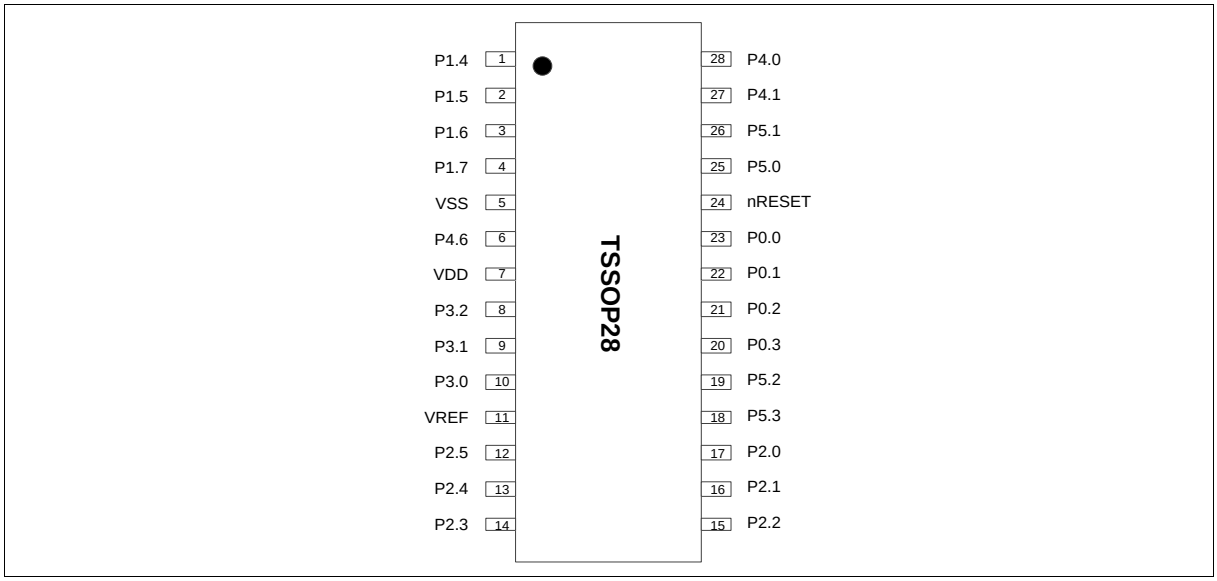


Figure 4.1-8 ML51EC0AE / ML51EB9AE Pin Assignment

4.1.1.7 SOP28 Package

Corresponding Part Number: ML51UC0AE / ML51UB9AE

ML51UC0AE / ML51UB9AE

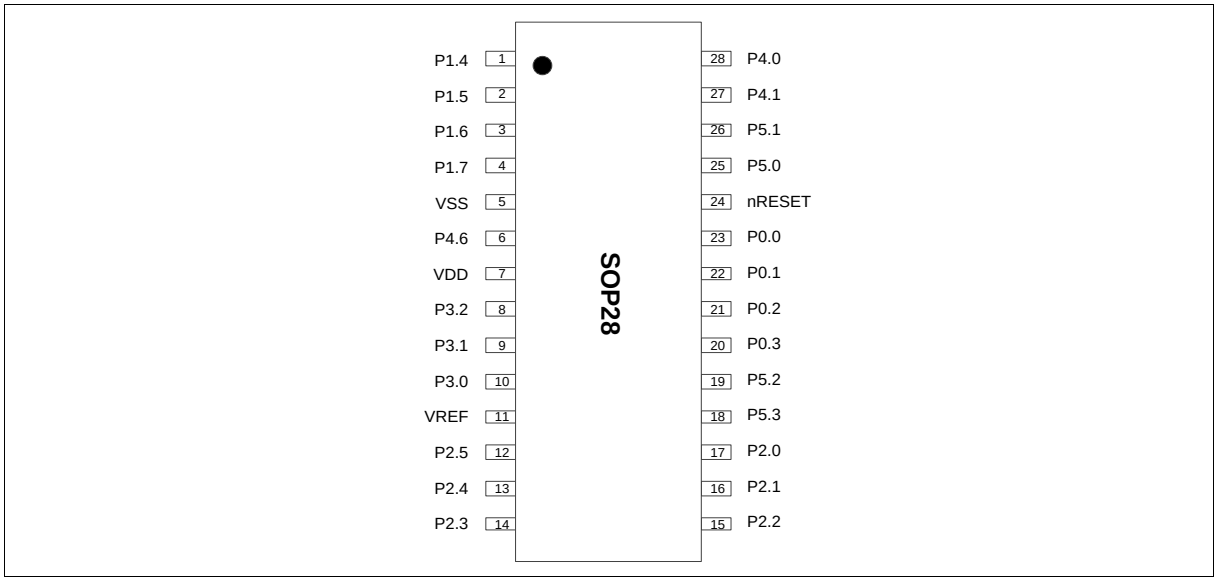


Figure 4.1-9 ML51UC0AE / ML51UB9AE Pin Assignment

4.1.1.8 TSSOP20 Package

Corresponding Part Number: ML51FB9AE

ML51FB9AE

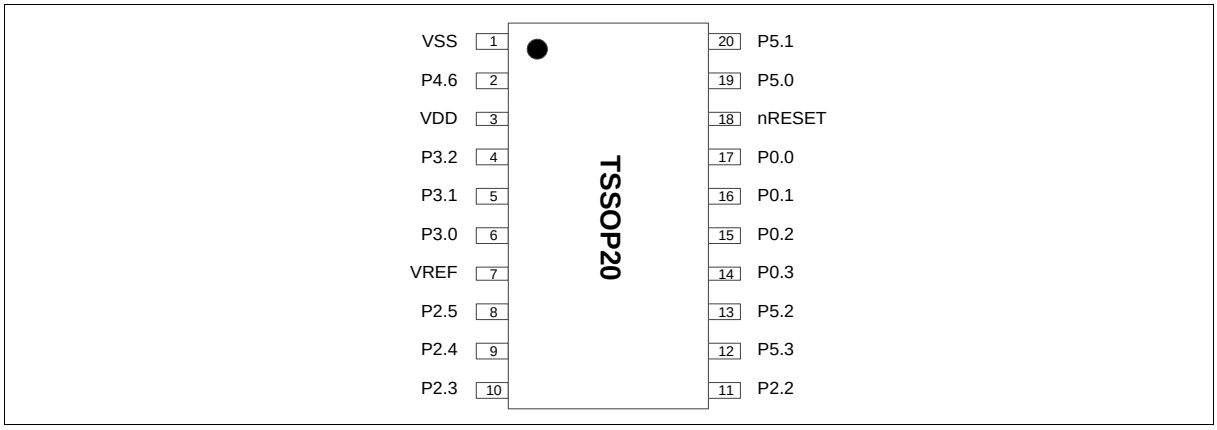


Figure 4.1-10 ML51FB9AE Pin Assignment

4.1.1.9 SOP20 Package

Corresponding Part Number: ML51OB9AE

ML51OB9AE

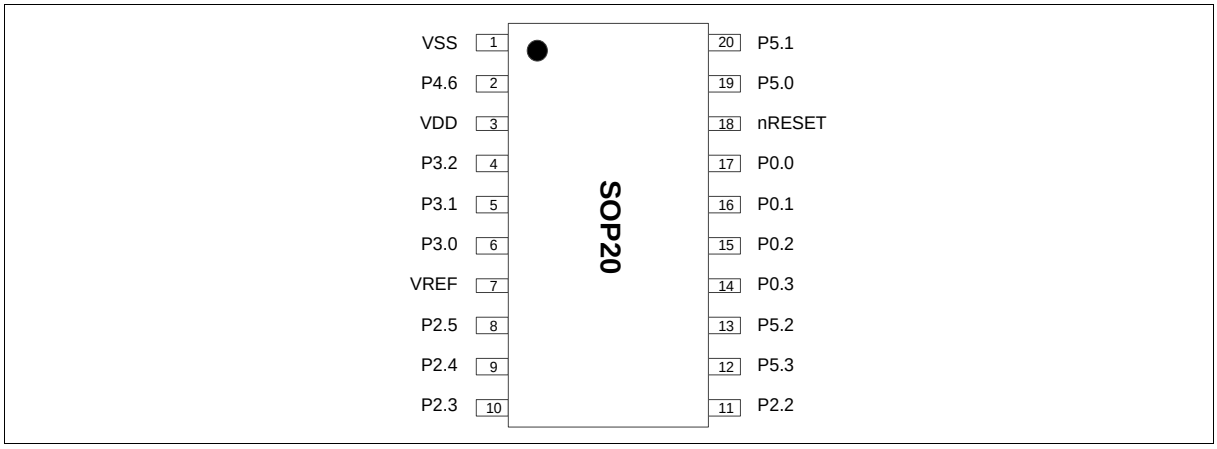


Figure 4.1-11 ML51OB9AE Pin Assignment

4.1.1.10 QFN20 Package

Corresponding Part Number: ML51XB9AE

ML51XB9AE

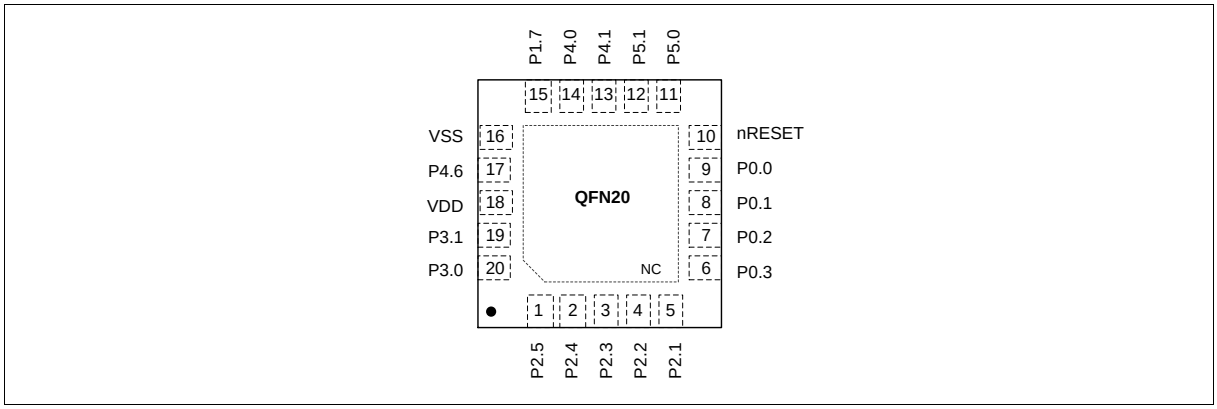


Figure 4.1-12 ML51XB9AE Pin Assignment

4.1.1.11 TSSOP14 Package

Corresponding Part Number: ML51DB9AE

ML51DB9AE

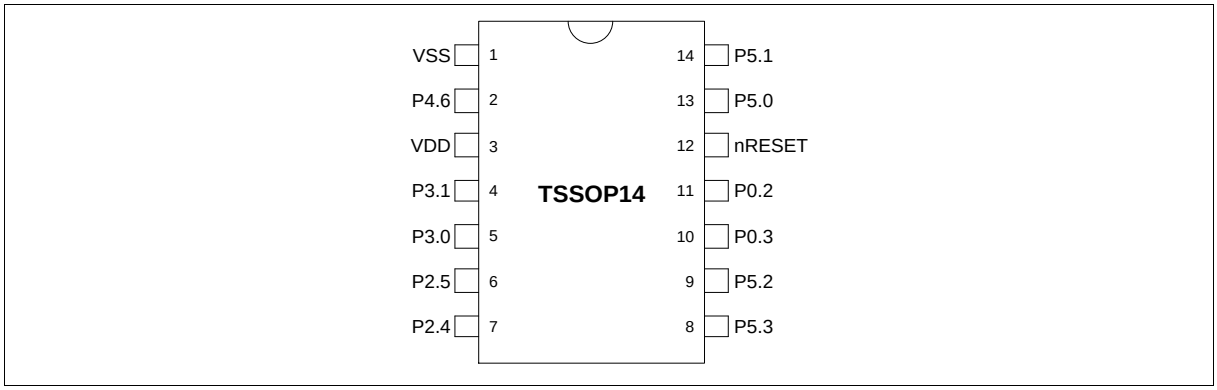


Figure 4.1-13 ML51DB9AE Pin Assignment

4.1.1.12 MSOP10 Package

Corresponding Part Number: ML51BB9AE

ML51BB9AE

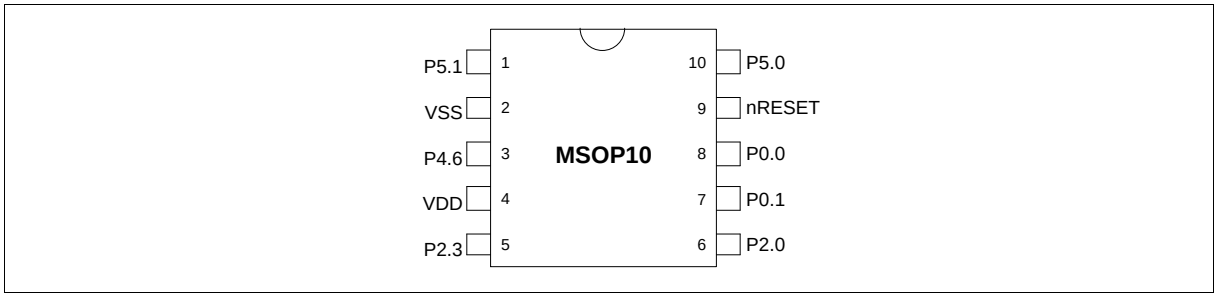


Figure 4.1-14 ML51BB9AE Pin Assignment

4.1.2 ML51/ML54/ML56 Series Multi Function Pin Diagram

4.1.2.1 LQFP64 Package

Corresponding Part Number: ML51SD1AE / ML54SD1AE / ML56SD1AE

ML51SD1AE Pin Function

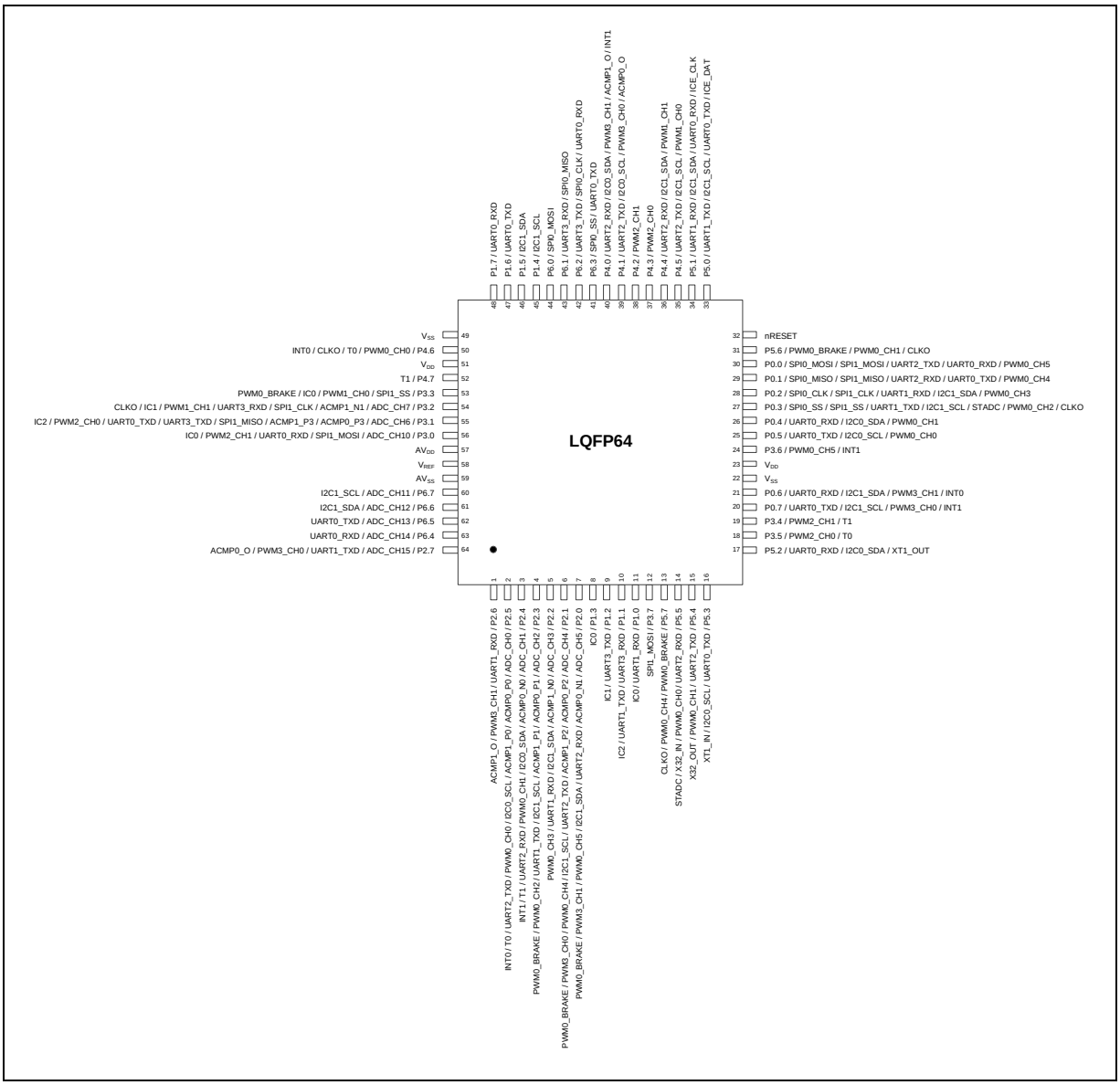


Figure 4.1-15 ML51SD1AE Multi-Function Pin assignment

Pin	ML51SD1AE Pin Function
1	P2.6 / UART1_RXD / PWM3_CH1 / ACMP1_O
2	P2.5 / ADC_CH0 / ACMP0_P0 / ACMP1_P0 / I2C0_SCL / PWM0_CH0 / UART2_TXD / T0 / INT0
3	P2.4 / ADC_CH1 / ACMP0_N0 / I2C0_SDA / PWM0_CH1 / UART2_RXD / T1 / INT1
4	P2.3 / ADC_CH2 / ACMP0_P1 / ACMP1_P1 / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE

Pin	ML51SD1AE Pin Function
5	P2.2 / ADC_CH3 / ACMP1_N0 / I2C1_SDA / UART1_RXD / PWM0_CH3
6	P2.1 / ADC_CH4 / ACMP0_P2 / ACMP1_P2 / UART2_TXD / I2C1_SCL / PWM0_CH4 / PWM3_CH0 / PWM0_BRAKE
7	P2.0 / ADC_CH5 / ACMP0_N1 / UART2_RXD / I2C1_SDA / PWM0_CH5 / PWM3_CH1 / PWM0_BRAKE
8	P1.3 / IC0
9	P1.2 / UART3_TXD / IC1
10	P1.1 / UART3_RXD / UART1_TXD / IC2
11	P1.0 / UART1_RXD / IC0
12	P3.7 / SPI1_MOSI
13	P5.7 / PWM0_BRAKE / PWM0_CH4 / CLKO
14	P5.5 / UART2_RXD / PWM0_CH0 / X32_IN / STADC
15	P5.4 / UART2_TXD / PWM0_CH1 / X32_OUT
16	P5.3 / UART0_TXD / I2C0_SCL / XT1_IN
17	P5.2 / UART0_RXD / I2C0_SDA / XT1_OUT
18	P3.5 / PWM2_CH0 / T0
19	P3.4 / PWM2_CH1 / T1
20	P0.7 / UART0_TXD / I2C1_SCL / PWM3_CH0 / INT1
21	P0.6 / UART0_RXD / I2C1_SDA / PWM3_CH1 / INT0
22	V_{SS}
23	V_{DD}
24	P3.6 / PWM0_CH5 / INT1
25	P0.5 / UART0_TXD / I2C0_SCL / PWM0_CH0
26	P0.4 / UART0_RXD / I2C0_SDA / PWM0_CH1
27	P0.3 / SPI0_SS / SPI1_SS / UART1_TXD / I2C1_SCL / STADC / PWM0_CH2 / CLKO
28	P0.2 / SPI0_CLK / SPI1_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
29	P0.1 / SPI0_MISO / SPI1_MISO / UART2_RXD / UART0_TXD / PWM0_CH4
30	P0.0 / SPI0_MOSI / SPI1_MOSI / UART2_TXD / UART0_RXD / PWM0_CH5
31	P5.6 / PWM0_BRAKE / PWM0_CH1 / CLKO
32	nRESET
33	P5.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
34	P5.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
35	P4.5 / UART2_TXD / I2C1_SCL / PWM1_CH0
36	P4.4 / UART2_RXD / I2C1_SDA / PWM1_CH1

Pin	ML51SD1AE Pin Function
37	P4.3 / PWM2_CH0
38	P4.2 / PWM2_CH1
39	P4.1 / UART2_TXD / I2C0_SCL / PWM3_CH0 / ACMP0_O
40	P4.0 / UART2_RXD / I2C0_SDA / PWM3_CH1 / ACMP1_O / INT1
41	P6.3 / SPI0_SS / UART0_TXD
42	P6.2 / UART3_TXD / SPI0_CLK / UART0_RXD
43	P6.1 / UART3_RXD / SPI0_MISO
44	P6.0 / SPI0_MOSI
45	P1.4 / I2C1_SCL
46	P1.5 / I2C1_SDA
47	P1.6 / UART0_TXD
48	P1.7 / UART0_RXD
49	V_{SS}
50	P4.6 / PWM0_CH0 / T0 / CLKO / INT0
51	V_{DD}
52	P4.7 / T1
53	P3.3 / SPI1_SS / PWM1_CH0 / IC0 / PWM0_BRAKE
54	P3.2 / ADC_CH7 / ACMP1_N1 / SPI1_CLK / UART3_RXD / PWM1_CH1 / IC1 / CLKO
55	P3.1 / ADC_CH6 / ACMP0_P3 / ACMP1_P3 / SPI1_MISO / UART3_TXD / UART0_TXD / PWM2_CH0 / IC2
56	P3.0 / ADC_CH10 / SPI1_MOSI / UART0_RXD / PWM2_CH1 / IC0
57	AV_{DD}
58	V_{REF}
59	AV_{SS}
60	P6.7 / ADC_CH11 / I2C1_SCL
61	P6.6 / ADC_CH12 / I2C1_SDA
62	P6.5 / ADC_CH13 / UART0_TXD
63	P6.4 / ADC_CH14 / UART0_RXD
64	P2.7 / ADC_CH15 / UART1_TXD / PWM3_CH0 / ACMP0_O

ML54SD1AE Pin Function

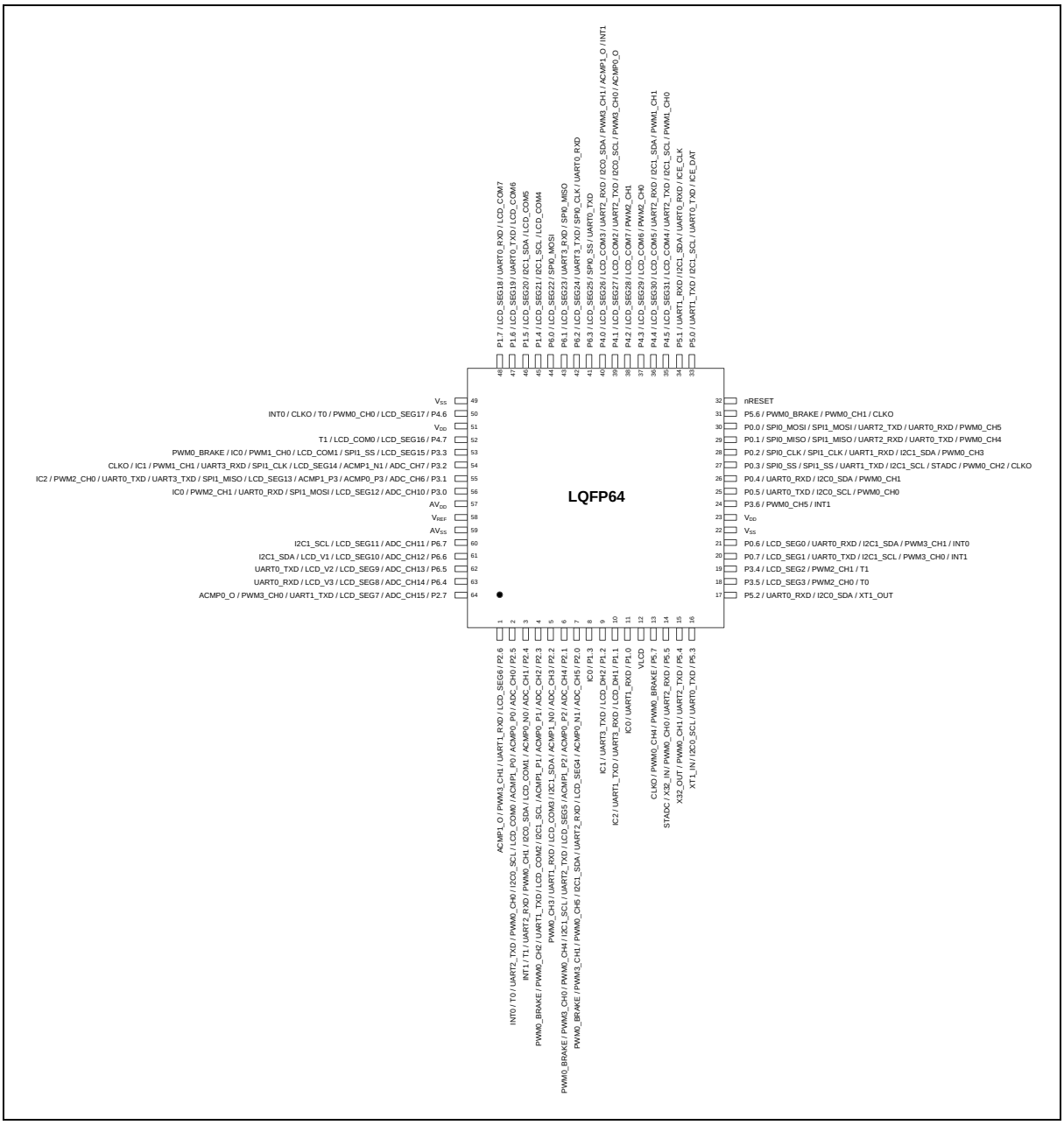


Figure 4.1-16 ML54SD1AE Multi-Function Pin assignment

Pin	ML54SD1AE Pin Function
1	P2.6 / LCD_SEG6 / UART1_RXD / PWM3_CH1 / ACMP1_O
2	P2.5 / ADC_CH0 / ACMP0_P0 / ACMP1_P0 / LCD_COM0 / I2C0_SCL / PWM0_CH0 / UART2_TXD / T0 / INT0
3	P2.4 / ADC_CH1 / ACMP0_N0 / LCD_COM1 / I2C0_SDA / PWM0_CH1 / UART2_RXD / T1 / INT1
4	P2.3 / ADC_CH2 / ACMP0_P1 / ACMP1_P1 / I2C1_SCL / LCD_COM2 / UART1_TXD / PWM0_CH2 / PWM0_BRAKE

Pin	ML54SD1AE Pin Function
5	P2.2 / ADC_CH3 / ACMP1_N0 / I2C1_SDA / LCD_COM3 / UART1_RXD / PWM0_CH3
6	P2.1 / ADC_CH4 / ACMP0_P2 / ACMP1_P2 / LCD_SEG5 / UART2_TXD / I2C1_SCL / PWM0_CH4 / PWM3_CH0 / PWM0_BRAKE
7	P2.0 / ADC_CH5 / ACMP0_N1 / LCD_SEG4 / UART2_RXD / I2C1_SDA / PWM0_CH5 / PWM3_CH1 / PWM0_BRAKE
8	P1.3 / IC0
9	P1.2 / LCD_DH2 / UART3_TXD / IC1
10	P1.1 / LCD_DH1 / UART3_RXD / UART1_TXD / IC2
11	P1.0 / UART1_RXD / IC0
12	VLCD
13	P5.7 / PWM0_BRAKE / PWM0_CH4 / CLKO
14	P5.5 / UART2_RXD / PWM0_CH0 / X32_IN / STADC
15	P5.4 / UART2_TXD / PWM0_CH1 / X32_OUT
16	P5.3 / UART0_TXD / I2C0_SCL / XT1_IN
17	P5.2 / UART0_RXD / I2C0_SDA / XT1_OUT
18	P3.5 / LCD_SEG3 / PWM2_CH0 / T0
19	P3.4 / LCD_SEG2 / PWM2_CH1 / T1
20	P0.7 / LCD_SEG1 / UART0_TXD / I2C1_SCL / PWM3_CH0 / INT1
21	P0.6 / LCD_SEG0 / UART0_RXD / I2C1_SDA / PWM3_CH1 / INT0
22	V_{SS}
23	V_{DD}
24	P3.6 / PWM0_CH5 / INT1
25	P0.5 / UART0_TXD / I2C0_SCL / PWM0_CH0
26	P0.4 / UART0_RXD / I2C0_SDA / PWM0_CH1
27	P0.3 / SPI0_SS / SPI1_SS / UART1_TXD / I2C1_SCL / STADC / PWM0_CH2 / CLKO
28	P0.2 / SPI0_CLK / SPI1_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
29	P0.1 / SPI0_MISO / SPI1_MISO / UART2_RXD / UART0_TXD / PWM0_CH4
30	P0.0 / SPI0_MOSI / SPI1_MOSI / UART2_TXD / UART0_RXD / PWM0_CH5
31	P5.6 / PWM0_BRAKE / PWM0_CH1 / CLKO
32	nRESET
33	P5.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
34	P5.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
35	P4.5 / LCD_SEG31 / LCD_COM4 / UART2_TXD / I2C1_SCL / PWM1_CH0

Pin	ML54SD1AE Pin Function
36	P4.4 / LCD_SEG30 / LCD_COM5 / UART2_RXD / I2C1_SDA / PWM1_CH1
37	P4.3 / LCD_SEG29 / LCD_COM6 / PWM2_CH0
38	P4.2 / LCD_SEG28 / LCD_COM7 / PWM2_CH1
39	P4.1 / LCD_SEG27 / LCD_COM2 / UART2_TXD / I2C0_SCL / PWM3_CH0 / ACMP0_O
40	P4.0 / LCD_SEG26 / LCD_COM3 / UART2_RXD / I2C0_SDA / PWM3_CH1 / ACMP1_O / INT1
41	P6.3 / LCD_SEG25 / SPI0_SS / UART0_TXD
42	P6.2 / LCD_SEG24 / UART3_TXD / SPI0_CLK / UART0_RXD
43	P6.1 / LCD_SEG23 / UART3_RXD / SPI0_MISO
44	P6.0 / LCD_SEG22 / SPI0_MOSI
45	P1.4 / LCD_SEG21 / I2C1_SCL / LCD_COM4
46	P1.5 / LCD_SEG20 / I2C1_SDA / LCD_COM5
47	P1.6 / LCD_SEG19 / UART0_TXD / LCD_COM6
48	P1.7 / LCD_SEG18 / UART0_RXD / LCD_COM7
49	V_{SS}
50	P4.6 / LCD_SEG17 / PWM0_CH0 / T0 / CLKO / INT0
51	V_{DD}
52	P4.7 / LCD_SEG16 / LCD_COM0 / T1
53	P3.3 / LCD_SEG15 / SPI1_SS / LCD_COM1 / PWM1_CH0 / IC0 / PWM0_BRAKE
54	P3.2 / ADC_CH7 / ACMP1_N1 / LCD_SEG14 / SPI1_CLK / UART3_RXD / PWM1_CH1 / IC1 / CLKO
55	P3.1 / ADC_CH6 / ACMP0_P3 / ACMP1_P3 / LCD_SEG13 / SPI1_MISO / UART3_TXD / UART0_TXD / PWM2_CH0 / IC2
56	P3.0 / ADC_CH10 / LCD_SEG12 / SPI1_MOSI / UART0_RXD / PWM2_CH1 / IC0
57	AV_{DD}
58	V_{REF}
59	AV_{SS}
60	P6.7 / ADC_CH11 / LCD_SEG11 / I2C1_SCL
61	P6.6 / ADC_CH12 / LCD_SEG10 / LCD_V1 / I2C1_SDA
62	P6.5 / ADC_CH13 / LCD_SEG9 / LCD_V2 / UART0_TXD
63	P6.4 / ADC_CH14 / LCD_SEG8 / LCD_V3 / UART0_RXD
64	P2.7 / ADC_CH15 / LCD_SEG7 / UART1_TXD / PWM3_CH0 / ACMP0_O

ML56SD1AE Pin Function

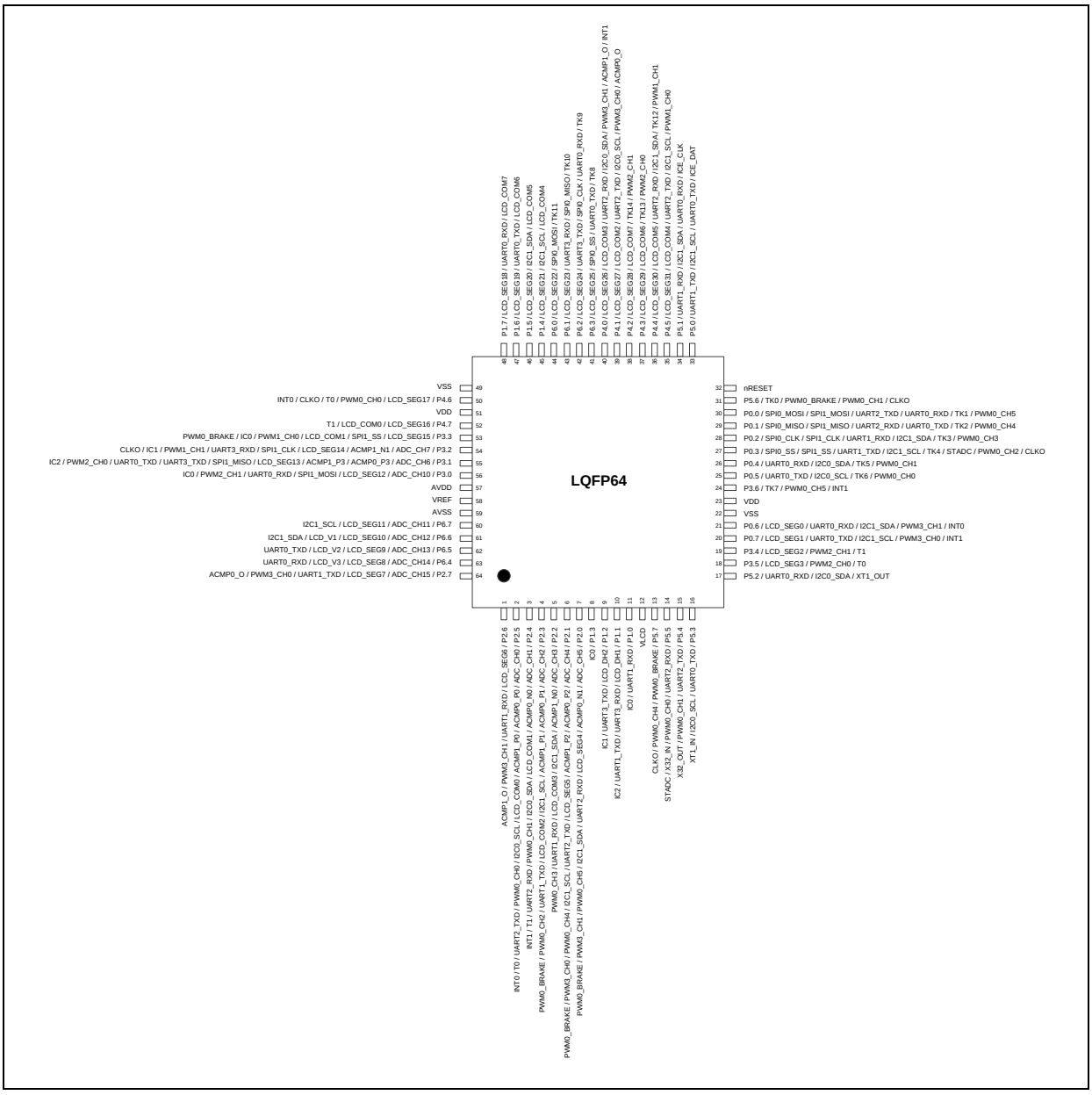


Figure 4.1-17 ML56SD1AE Multi-Function Pin assignment

Pin	ML56SD1AE Pin Function
1	P2.6/LCD_SEG6/UART1_RXD/PWM3_CH1/ACMP1_O
2	P2.5/ADC_CH0/ACMP0_P0/ACMP1_P0/LCD_COM0/I2C0_SCL/PWM0_CH0/UART2_TXD/T0/INT0
3	P2.4/ADC_CH1/ACMP0_N0/LCD_COM1/I2C0_SDA/PWM0_CH1/UART2_RXD/T1/INT1
4	P2.3/ADC_CH2/ACMP0_P1/ACMP1_P1/I2C1_SCL/LCD_COM2/UART1_TXD/PWM0_CH2/PWM0_BRAKE
5	P2.2/ADC_CH3/ACMP1_N0/I2C1_SDA/LCD_COM3/UART1_RXD/PWM0_CH3
6	P2.1/ADC_CH4/ACMP0_P2/ACMP1_P2/LCD_SEG5/UART2_TXD/I2C1_SCL/PWM0_CH4/PWM3_CH0/PWM0_BRAKE

Pin	ML56SD1AE Pin Function
7	P2.0/ADC_CH5/ACMP0_N1/LCD_SEG4/UART2_RXD/I2C1_SDA/PWM0_CH5/PWM3_CH1/PWM0_BRAKE
8	P1.3/IC0
9	P1.2/LCD_DH2/UART3_TXD/IC1
10	P1.1/LCD_DH1/UART3_RXD/UART1_TXD/IC2
11	P1.0/UART1_RXD/IC0
12	VLCD
13	P5.7/PWM0_BRAKE/PWM0_CH4/CLKO
14	P5.5/UART2_RXD/PWM0_CH0/X32_IN/STADC
15	P5.4/UART2_TXD/PWM0_CH1/X32_OUT
16	P5.3/UART0_TXD/I2C0_SCL/XT1_IN
17	P5.2/UART0_RXD/I2C0_SDA/XT1_OUT
18	P3.5/LCD_SEG3/PWM2_CH0/T0
19	P3.4/LCD_SEG2/PWM2_CH1/T1
20	P0.7/LCD_SEG1/UART0_TXD/I2C1_SCL/PWM3_CH0/INT1
21	P0.6/LCD_SEG0/UART0_RXD/I2C1_SDA/PWM3_CH1/INT0
22	VSS
23	V_{DD}
24	P3.6/TK7/PWM0_CH5/INT1
25	P0.5/UART0_TXD/I2C0_SCL/TK6/PWM0_CH0
26	P0.4/UART0_RXD/I2C0_SDA/TK5/PWM0_CH1
27	P0.3/SPI0_SS/SPI1_SS/UART1_TXD/I2C1_SCL/TK4/STADC/PWM0_CH2/CLKO
28	P0.2/SPI0_CLK/SPI1_CLK/UART1_RXD/I2C1_SDA/TK3/PWM0_CH3
29	P0.1/SPI0_MISO/SPI1_MISO/UART2_RXD/UART0_TXD/TK2/PWM0_CH4
30	P0.0/SPI0_MOSI/SPI1_MOSI/UART2_TXD/UART0_RXD/TK1/PWM0_CH5
31	P5.6/TK0/PWM0_BRAKE/PWM0_CH1/CLKO
32	nRESET
33	P5.0/UART1_TXD/I2C1_SCL/UART0_TXD/ICE_DAT
34	P5.1/UART1_RXD/I2C1_SDA/UART0_RXD/ICE_CLK
35	P4.5/LCD_SEG31/LCD_COM4/UART2_TXD/I2C1_SCL/PWM1_CH0
36	P4.4/LCD_SEG30/LCD_COM5/UART2_RXD/I2C1_SDA/TK12/PWM1_CH1
37	P4.3/LCD_SEG29/LCD_COM6/TK13/PWM2_CH0
38	P4.2/LCD_SEG28/LCD_COM7/TK14/PWM2_CH1
39	P4.1/LCD_SEG27/LCD_COM2/UART2_TXD/I2C0_SCL/PWM3_CH0/ACMP0_O
40	P4.0/LCD_SEG26/LCD_COM3/UART2_RXD/I2C0_SDA/PWM3_CH1/ACMP1_O/INT1

Pin	ML56SD1AE Pin Function
41	P6.3/LCD_SEG25/SPI0_SS/UART0_TXD/TK8
42	P6.2/LCD_SEG24/UART3_TXD/SPI0_CLK/UART0_RXD/TK9
43	P6.1/LCD_SEG23/UART3_RXD/SPI0_MISO/TK10
44	P6.0/LCD_SEG22/SPI0_MOSI/TK11
45	P1.4/LCD_SEG21/I2C1_SCL/LCD_COM4
46	P1.5/LCD_SEG20/I2C1_SDA/LCD_COM5
47	P1.6/LCD_SEG19/UART0_TXD/LCD_COM6
48	P1.7/LCD_SEG18/UART0_RXD/LCD_COM7
49	VSS
50	P4.6/LCD_SEG17/PWM0_CH0/T0/CLKO/INT0
51	V_{DD}
52	P4.7/LCD_SEG16/LCD_COM0/T1
53	P3.3/LCD_SEG15/SPI1_SS/LCD_COM1/PWM1_CH0/IC0/PWM0_BRAKE
54	P3.2/ADC_CH7/ACMP1_N1/LCD_SEG14/SPI1_CLK/UART3_RXD/PWM1_CH1/IC1/CLKO
55	P3.1/ADC_CH6/ACMP0_P3/ACMP1_P3/LCD_SEG13/SPI1_MISO/UART3_TXD/UART0_TXD/PWM2_CH0/IC2
56	P3.0/ADC_CH10/LCD_SEG12/SPI1_MOSI/UART0_RXD/PWM2_CH1/IC0
57	AV_{DD}
58	V_{REF}
59	AVSS
60	P6.7/ADC_CH11/LCD_SEG11/I2C1_SCL
61	P6.6/ADC_CH12/LCD_SEG10/LCD_V1/I2C1_SDA
62	P6.5/ADC_CH13/LCD_SEG9/LCD_V2/UART0_TXD
63	P6.4/ADC_CH14/LCD_SEG8/LCD_V3/UART0_RXD
64	P2.7/ADC_CH15/LCD_SEG7/UART1_TXD/PWM3_CH0/ACMP0_O

4.1.2.2 LQFP48 Package

Corresponding Part Number: ML51LD1AE

ML51LD1AE Pin Function

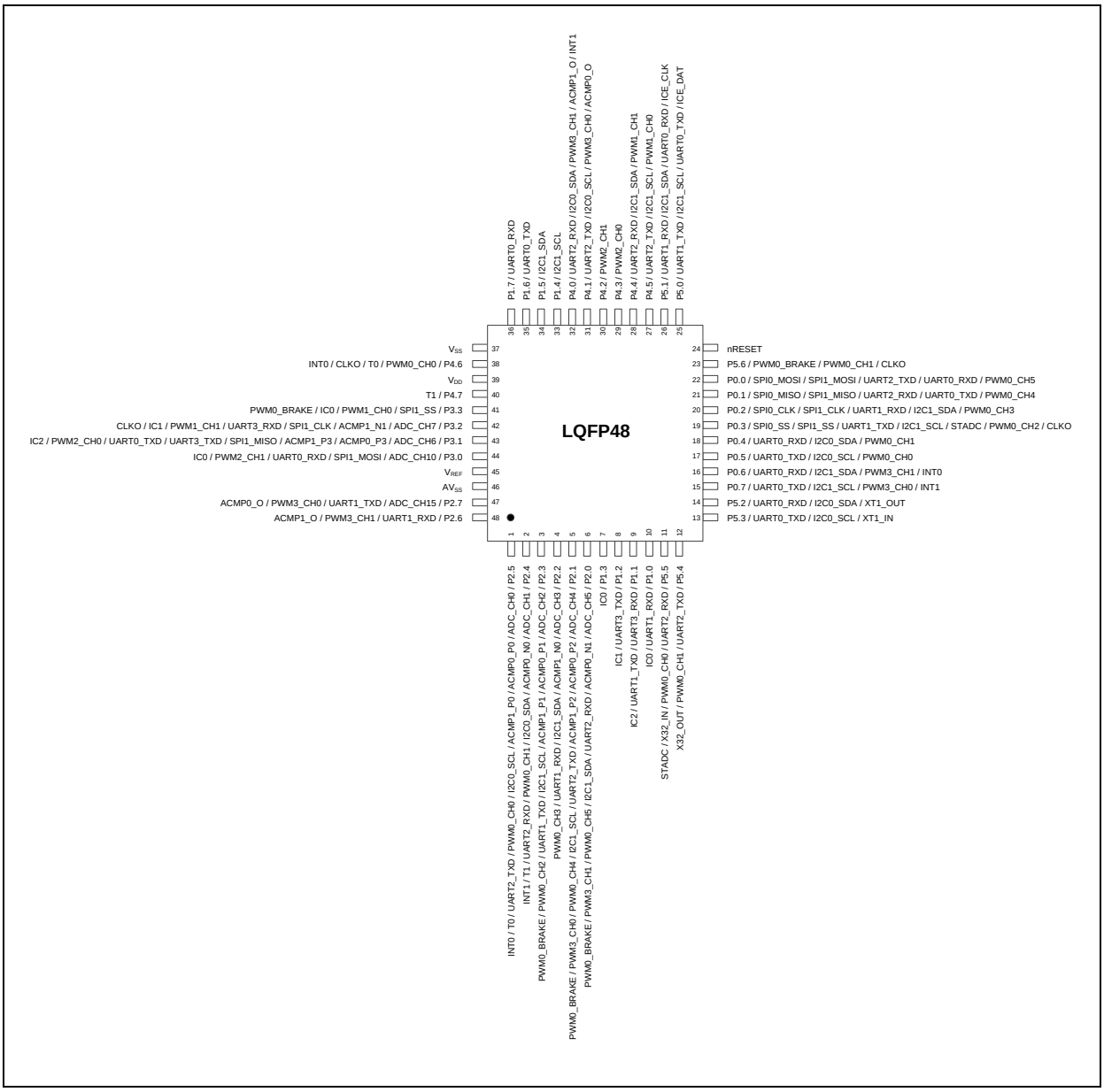


Figure 4.1-18 ML51LD1AE Multi-Function Pin assignment

Pin	ML51LD1AE Pin Function
1	P2.5 / ADC_CH0 / ACMP0_P0 / ACMP1_P0 / I2C0_SCL / PWM0_CH0 / UART2_TXD / T0 / INT0
2	P2.4 / ADC_CH1 / ACMP0_N0 / I2C0_SDA / PWM0_CH1 / UART2_RXD / T1 / INT1
3	P2.3 / ADC_CH2 / ACMP0_P1 / ACMP1_P1 / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE
4	P2.2 / ADC_CH3 / ACMP1_N0 / I2C1_SDA / UART1_RXD / PWM0_CH3

Pin	ML51LD1AE Pin Function
5	P2.1 / ADC_CH4 / ACMP0_P2 / ACMP1_P2 / UART2_TXD / I2C1_SCL / PWM0_CH4 / PWM3_CH0 / PWM0_BRAKE
6	P2.0 / ADC_CH5 / ACMP0_N1 / UART2_RXD / I2C1_SDA / PWM0_CH5 / PWM3_CH1 / PWM0_BRAKE
7	P1.3 / IC0
8	P1.2 / UART3_TXD / IC1
9	P1.1 / UART3_RXD / UART1_TXD / IC2
10	P1.0 / UART1_RXD / IC0
11	P5.5 / UART2_RXD / PWM0_CH0 / X32_IN / STADC
12	P5.4 / UART2_TXD / PWM0_CH1 / X32_OUT
13	P5.3 / UART0_TXD / I2C0_SCL / XT1_IN
14	P5.2 / UART0_RXD / I2C0_SDA / XT1_OUT
15	P0.7 / UART0_TXD / I2C1_SCL / PWM3_CH0 / INT1
16	P0.6 / UART0_RXD / I2C1_SDA / PWM3_CH1 / INT0
17	P0.5 / UART0_TXD / I2C0_SCL / PWM0_CH0
18	P0.4 / UART0_RXD / I2C0_SDA / PWM0_CH1
19	P0.3 / SPI0_SS / SPI1_SS / UART1_TXD / I2C1_SCL / STADC / PWM0_CH2 / CLKO
20	P0.2 / SPI0_CLK / SPI1_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
21	P0.1 / SPI0_MISO / SPI1_MISO / UART2_RXD / UART0_TXD / PWM0_CH4
22	P0.0 / SPI0_MOSI / SPI1_MOSI / UART2_TXD / UART0_RXD / PWM0_CH5
23	P5.6 / PWM0_BRAKE / PWM0_CH1 / CLKO
24	nRESET
25	P5.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
26	P5.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
27	P4.5 / UART2_TXD / I2C1_SCL / PWM1_CH0
28	P4.4 / UART2_RXD / I2C1_SDA / PWM1_CH1
29	P4.3 / PWM2_CH0
30	P4.2 / PWM2_CH1
31	P4.1 / UART2_TXD / I2C0_SCL / PWM3_CH0 / ACMP0_O
32	P4.0 / UART2_RXD / I2C0_SDA / PWM3_CH1 / ACMP1_O / INT1
33	P1.4 / I2C1_SCL
34	P1.5 / I2C1_SDA
35	P1.6 / UART0_TXD
36	P1.7 / UART0_RXD

Pin	ML51LD1AE Pin Function
37	VSS
38	P4.6 / PWM0_CH0 / T0 / CLKO / INT0
39	V_{DD}
40	P4.7 / T1
41	P3.3 / SPI1_SS / PWM1_CH0 / IC0 / PWM0_BRAKE
42	P3.2 / ADC_CH7 / ACMP1_N1 / SPI1_CLK / UART3_RXD / PWM1_CH1 / IC1 / CLKO
43	P3.1 / ADC_CH6 / ACMP0_P3 / ACMP1_P3 / SPI1_MISO / UART3_TXD / UART0_TXD / PWM2_CH0 / IC2
44	P3.0 / ADC_CH10 / SPI1_MOSI / UART0_RXD / PWM2_CH1 / IC0
45	V_{REF}
46	AV_{SS}
47	P2.7 / ADC_CH15 / UART1_TXD / PWM3_CH0 / ACMP0_O
48	P2.6 / UART1_RXD / PWM3_CH1 / ACMP1_O

ML54LD1AE Pin Function

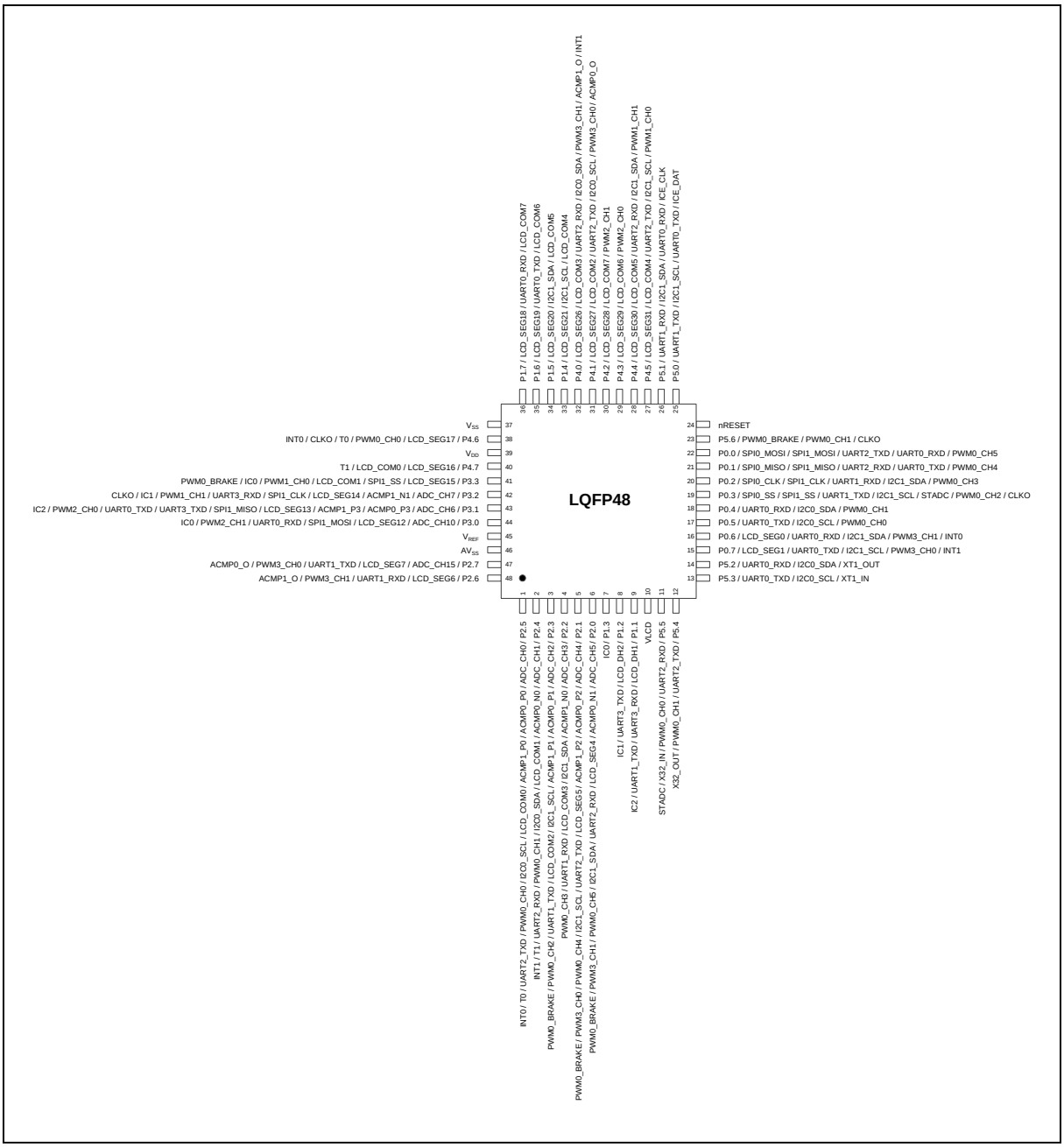


Figure 4.1-19 ML54LD1AE Multi-Function Pin assignment

Pin	ML54LD1AE Pin Function
1	P2.5 / ADC_CH0 / ACMP0_P0 / ACMP1_P0 / LCD_COM0 / I2C0_SCL / PWM0_CH0 / UART2_TXD / T0 / INT0
2	P2.4 / ADC_CH1 / ACMP0_N0 / LCD_COM1 / I2C0_SDA / PWM0_CH1 / UART2_RXD / T1 / INT1
3	P2.3 / ADC_CH2 / ACMP0_P1 / ACMP1_P1 / I2C1_SCL / LCD_COM2 / UART1_TXD / PWM0_CH2 / PWM0_BRAKE
4	P2.2 / ADC_CH3 / ACMP1_N0 / I2C1_SDA / LCD_COM3 / UART1_RXD / PWM0_CH3

Pin	ML54LD1AE Pin Function
5	P2.1 / ADC_CH4 / ACMP0_P2 / ACMP1_P2 / LCD_SEG5 / UART2_TXD / I2C1_SCL / PWM0_CH4 / PWM3_CH0 / PWM0_BRAKE
6	P2.0 / ADC_CH5 / ACMP0_N1 / LCD_SEG4 / UART2_RXD / I2C1_SDA / PWM0_CH5 / PWM3_CH1 / PWM0_BRAKE
7	P1.3 / IC0
8	P1.2 / LCD_DH2 / UART3_TXD / IC1
9	P1.1 / LCD_DH1 / UART3_RXD / UART1_TXD / IC2
10	VLCD
11	P5.5 / UART2_RXD / PWM0_CH0 / X32_IN / STADC
12	P5.4 / UART2_TXD / PWM0_CH1 / X32_OUT
13	P5.3 / UART0_TXD / I2C0_SCL / XT1_IN
14	P5.2 / UART0_RXD / I2C0_SDA / XT1_OUT
15	P0.7 / LCD_SEG1 / UART0_TXD / I2C1_SCL / PWM3_CH0 / INT1
16	P0.6 / LCD_SEG0 / UART0_RXD / I2C1_SDA / PWM3_CH1 / INT0
17	P0.5 / UART0_TXD / I2C0_SCL / PWM0_CH0
18	P0.4 / UART0_RXD / I2C0_SDA / PWM0_CH1
19	P0.3 / SPI0_SS / SPI1_SS / UART1_TXD / I2C1_SCL / STADC / PWM0_CH2 / CLKO
20	P0.2 / SPI0_CLK / SPI1_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
21	P0.1 / SPI0_MISO / SPI1_MISO / UART2_RXD / UART0_TXD / PWM0_CH4
22	P0.0 / SPI0_MOSI / SPI1_MOSI / UART2_TXD / UART0_RXD / PWM0_CH5
23	P5.6 / PWM0_BRAKE / PWM0_CH1 / CLKO
24	nRESET
25	P5.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
26	P5.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
27	P4.5 / LCD_SEG31 / LCD_COM4 / UART2_TXD / I2C1_SCL / PWM1_CH0
28	P4.4 / LCD_SEG30 / LCD_COM5 / UART2_RXD / I2C1_SDA / PWM1_CH1
29	P4.3 / LCD_SEG29 / LCD_COM6 / PWM2_CH0
30	P4.2 / LCD_SEG28 / LCD_COM7 / PWM2_CH1
31	P4.1 / LCD_SEG27 / LCD_COM2 / UART2_TXD / I2C0_SCL / PWM3_CH0 / ACMP0_O
32	P4.0 / LCD_SEG26 / LCD_COM3 / UART2_RXD / I2C0_SDA / PWM3_CH1 / ACMP1_O / INT1
33	P1.4 / LCD_SEG21 / I2C1_SCL / LCD_COM4
34	P1.5 / LCD_SEG20 / I2C1_SDA / LCD_COM5
35	P1.6 / LCD_SEG19 / UART0_TXD / LCD_COM6

Pin	ML54LD1AE Pin Function
36	P1.7 / LCD_SEG18 / UART0_RXD / LCD_COM7
37	VSS
38	P4.6 / LCD_SEG17 / PWM0_CH0 / T0 / CLKO / INT0
39	V_{DD}
40	P4.7 / LCD_SEG16 / LCD_COM0 / T1
41	P3.3 / LCD_SEG15 / SPI1_SS / LCD_COM1 / PWM1_CH0 / IC0 / PWM0_BRAKE
42	P3.2 / ADC_CH7 / ACMP1_N1 / LCD_SEG14 / SPI1_CLK / UART3_RXD / PWM1_CH1 / IC1 / CLKO
43	P3.1 / ADC_CH6 / ACMP0_P3 / ACMP1_P3 / LCD_SEG13 / SPI1_MISO / UART3_TXD / UART0_TXD / PWM2_CH0 / IC2
44	P3.0 / ADC_CH10 / LCD_SEG12 / SPI1_MOSI / UART0_RXD / PWM2_CH1 / IC0
45	V_{REF}
46	AV_{SS}
47	P2.7 / ADC_CH15 / LCD_SEG7 / UART1_TXD / PWM3_CH0 / ACMP0_O
48	P2.6 / LCD_SEG6 / UART1_RXD / PWM3_CH1 / ACMP1_O

ML56LD1AE Pin Function

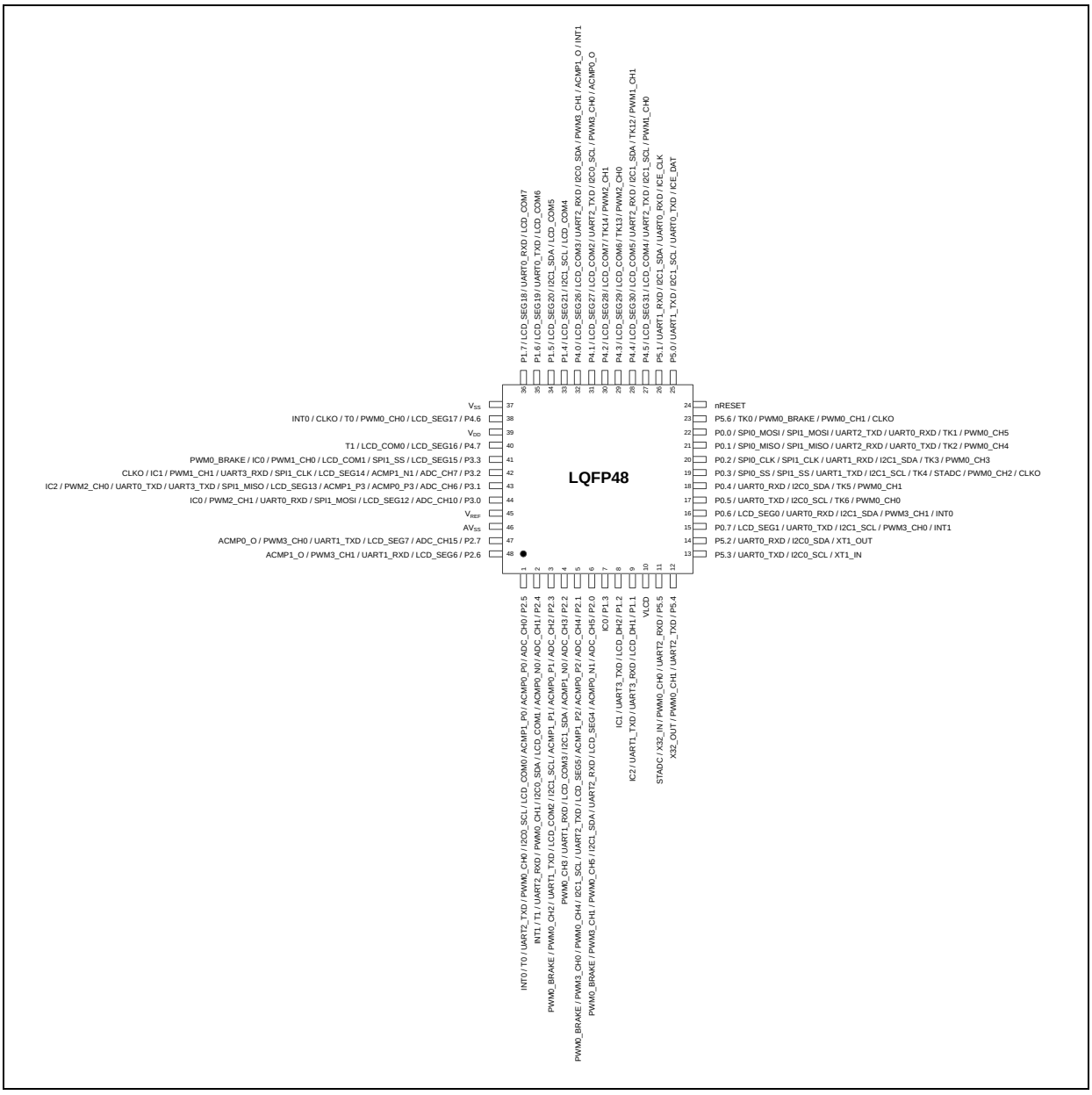


Figure 4.1-20 ML56LD1AE Multi-Function Pin assignment

Pin	ML56LD1AE/ML56LC1AE Pin Function
1	P2.5/ADC_CH0/ACMP0_P0/ACMP1_P0/LCD_COM0/I2C0_SCL/PWM0_CH0/UART2_TXD/T0/INT0
2	P2.4/ADC_CH1/ACMP0_N0/LCD_COM1/I2C0_SDA/PWM0_CH1/UART2_RXD/T1/INT1
3	P2.3/ADC_CH2/ACMP0_P1/ACMP1_P1/I2C1_SCL/LCD_COM2/UART1_TXD/PWM0_CH2/PWM0_BRAKE
4	P2.2/ADC_CH3/ACMP1_N0/I2C1_SDA/LCD_COM3/UART1_RXD/PWM0_CH3
5	P2.1/ADC_CH4/ACMP0_P2/ACMP1_P2/LCD_SEG5/UART2_TXD/I2C1_SCL/PWM0_CH4/PWM3_CH0/PWM0_BRAKE
6	P2.0/ADC_CH5/ACMP0_N1/LCD_SEG4/UART2_RXD/I2C1_SDA/PWM0_CH5/PWM3_CH1/PWM0_BRAKE

Pin	ML56LD1AE/ML56LC1AE Pin Function
7	P1.3/IC0
8	P1.2/LCD_DH2/UART3_TXD/IC1
9	P1.1/LCD_DH1/UART3_RXD/UART1_TXD/IC2
10	VLCD
11	P5.5/UART2_RXD/PWM0_CH0/X32_IN/STADC
12	P5.4/UART2_TXD/PWM0_CH1/X32_OUT
13	P5.3/UART0_TXD/I2C0_SCL/XT1_IN
14	P5.2/UART0_RXD/I2C0_SDA/XT1_OUT
15	P0.7/LCD_SEG1/UART0_TXD/I2C1_SCL/PWM3_CH0/INT1
16	P0.6/LCD_SEG0/UART0_RXD/I2C1_SDA/PWM3_CH1/INT0
17	P0.5/UART0_TXD/I2C0_SCL/TK6/PWM0_CH0
18	P0.4/UART0_RXD/I2C0_SDA/TK5/PWM0_CH1
19	P0.3/SPI0_SS/SPI1_SS/UART1_TXD/I2C1_SCL/TK4/STADC/PWM0_CH2/CLKO
20	P0.2/SPI0_CLK/SPI1_CLK/UART1_RXD/I2C1_SDA/TK3/PWM0_CH3
21	P0.1/SPI0_MISO/SPI1_MISO/UART2_RXD/UART0_TXD/TK2/PWM0_CH4
22	P0.0/SPI0_MOSI/SPI1_MOSI/UART2_TXD/UART0_RXD/TK1/PWM0_CH5
23	P5.6/TK0/PWM0_BRAKE/PWM0_CH1/CLKO
24	nRESET
25	P5.0/UART1_TXD/I2C1_SCL/UART0_TXD/ICE_DAT
26	P5.1/UART1_RXD/I2C1_SDA/UART0_RXD/ICE_CLK
27	P4.5/LCD_SEG31/LCD_COM4/UART2_TXD/I2C1_SCL/PWM1_CH0
28	P4.4/LCD_SEG30/LCD_COM5/UART2_RXD/I2C1_SDA/TK12/PWM1_CH1
29	P4.3/LCD_SEG29/LCD_COM6/TK13/PWM2_CH0
30	P4.2/LCD_SEG28/LCD_COM7/TK14/PWM2_CH1
31	P4.1/LCD_SEG27/LCD_COM2/UART2_TXD/I2C0_SCL/PWM3_CH0/ACMP0_O
32	P4.0/LCD_SEG26/LCD_COM3/UART2_RXD/I2C0_SDA/PWM3_CH1/ACMP1_O/INT1
33	P1.4/LCD_SEG21/I2C1_SCL/LCD_COM4
34	P1.5/LCD_SEG20/I2C1_SDA/LCD_COM5
35	P1.6/LCD_SEG19/UART0_TXD/LCD_COM6
36	P1.7/LCD_SEG18/UART0_RXD/LCD_COM7
37	VSS
38	P4.6/LCD_SEG17/PWM0_CH0/T0/CLKO/INT0
39	V_{DD}
40	P4.7/LCD_SEG16/LCD_COM0/T1

Pin	ML56LD1AE/ML56LC1AE Pin Function
41	P3.3/LCD_SEG15/SPI1_SS/LCD_COM1/PWM1_CH0/IC0/PWM0_BRAKE
42	P3.2/ADC_CH7/ACMP1_N1/LCD_SEG14/SPI1_CLK/UART3_RXD/PWM1_CH1/IC1/CLKO
43	P3.1/ADC_CH6/ACMP0_P3/ACMP1_P3/LCD_SEG13/SPI1_MISO/UART3_TXD/UART0_TXD/PWM2_CH0/IC2
44	P3.0/ADC_CH10/LCD_SEG12/SPI1_MOSI/UART0_RXD/PWM2_CH1/IC0
45	V_{REF}
46	AVSS
47	P2.7/ADC_CH15/LCD_SEG7/UART1_TXD/PWM3_CH0/ACMP0_O
48	P2.6/LCD_SEG6/UART1_RXD/PWM3_CH1/ACMP1_O

4.1.2.3 LQFP44 Package

ML54MD1AE Pin Function

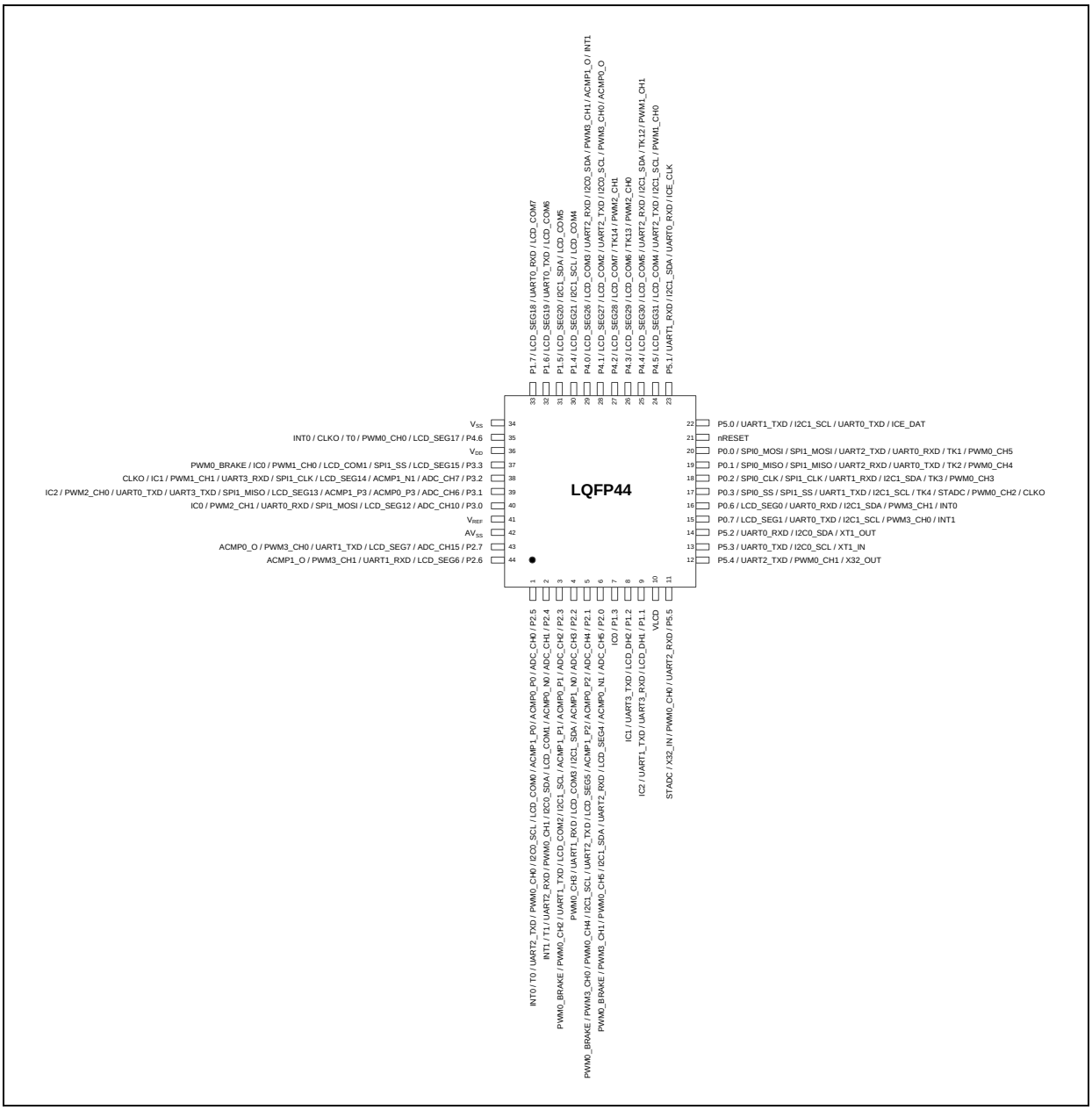


Figure 4.1-21 ML54MD1AE Multi-Function Pin assignment

Pin	ML54MD1AE Pin Function
1	P2.5 / ADC_CH0 / ACMP0_P0 / ACMP1_P0 / LCD_COM0 / I2C0_SCL / PWM0_CH0 / UART2_TXD / T0 / INT0
2	P2.4 / ADC_CH1 / ACMP0_N0 / LCD_COM1 / I2C0_SDA / PWM0_CH1 / UART2_RXD / T1 / INT1
3	P2.3 / ADC_CH2 / ACMP0_P1 / ACMP1_P1 / I2C1_SCL / LCD_COM2 / UART1_TXD / PWM0_CH2 / PWM0_BRAKE

Pin	ML54MD1AE Pin Function
4	P2.2 / ADC_CH3 / ACMP1_N0 / I2C1_SDA / LCD_COM3 / UART1_RXD / PWM0_CH3
5	P2.1 / ADC_CH4 / ACMP0_P2 / ACMP1_P2 / LCD_SEG5 / UART2_TXD / I2C1_SCL / PWM0_CH4 / PWM3_CH0 / PWM0_BRAKE
6	P2.0 / ADC_CH5 / ACMP0_N1 / LCD_SEG4 / UART2_RXD / I2C1_SDA / PWM0_CH5 / PWM3_CH1 / PWM0_BRAKE
7	P1.3 / IC0
8	P1.2 / LCD_DH2 / UART3_TXD / IC1
9	P1.1 / LCD_DH1 / UART3_RXD / UART1_TXD / IC2
10	VLCD
11	P5.5 / UART2_RXD / PWM0_CH0 / X32_IN / STADC
12	P5.4 / UART2_TXD / PWM0_CH1 / X32_OUT
13	P5.3 / UART0_TXD / I2C0_SCL / XT1_IN
14	P5.2 / UART0_RXD / I2C0_SDA / XT1_OUT
15	P0.7 / LCD_SEG1 / UART0_TXD / I2C1_SCL / PWM3_CH0 / INT1
16	P0.6 / LCD_SEG0 / UART0_RXD / I2C1_SDA / PWM3_CH1 / INT0
17	P0.3 / SPI0_SS / SPI1_SS / UART1_TXD / I2C1_SCL / STADC / PWM0_CH2 / CLKO
18	P0.2 / SPI0_CLK / SPI1_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
19	P0.1 / SPI0_MISO / SPI1_MISO / UART2_RXD / UART0_TXD / PWM0_CH4
20	P0.0 / SPI0_MOSI / SPI1_MOSI / UART2_TXD / UART0_RXD / PWM0_CH5
21	nRESET
22	P5.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
23	P5.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
24	P4.5 / LCD_SEG31 / LCD_COM4 / UART2_TXD / I2C1_SCL / PWM1_CH0
25	P4.4 / LCD_SEG30 / LCD_COM5 / UART2_RXD / I2C1_SDA / PWM1_CH1
26	P4.3 / LCD_SEG29 / LCD_COM6 / PWM2_CH0
27	P4.2 / LCD_SEG28 / LCD_COM7 / PWM2_CH1
28	P4.1 / LCD_SEG27 / LCD_COM2 / UART2_TXD / I2C0_SCL / PWM3_CH0 / ACMP0_O
29	P4.0 / LCD_SEG26 / LCD_COM3 / UART2_RXD / I2C0_SDA / PWM3_CH1 / ACMP1_O / INT1
30	P1.4 / LCD_SEG21 / I2C1_SCL / LCD_COM4
31	P1.5 / LCD_SEG20 / I2C1_SDA / LCD_COM5
32	P1.6 / LCD_SEG19 / UART0_TXD / LCD_COM6
33	P1.7 / LCD_SEG18 / UART0_RXD / LCD_COM7
34	V_{SS}

Pin	ML54MD1AE Pin Function
35	P4.6 / LCD_SEG17 / PWM0_CH0 / T0 / CLKO / INT0
36	V_{DD}
37	P3.3 / LCD_SEG15 / SPI1_SS / LCD_COM1 / PWM1_CH0 / IC0 / PWM0_BRAKE
38	P3.2 / ADC_CH7 / ACMP1_N1 / LCD_SEG14 / SPI1_CLK / UART3_RXD / PWM1_CH1 / IC1 / CLKO
39	P3.1 / ADC_CH6 / ACMP0_P3 / ACMP1_P3 / LCD_SEG13 / SPI1_MISO / UART3_TXD / UART0_TXD / PWM2_CH0 / IC2
40	P3.0 / ADC_CH10 / LCD_SEG12 / SPI1_MOSI / UART0_RXD / PWM2_CH1 / IC0
41	V_{REF}
42	AV_{SS}
43	P2.7 / ADC_CH15 / LCD_SEG7 / UART1_TXD / PWM3_CH0 / ACMP0_O
44	P2.6 / LCD_SEG6 / UART1_RXD / PWM3_CH1 / ACMP1_O

ML56MD1AE Pin Function

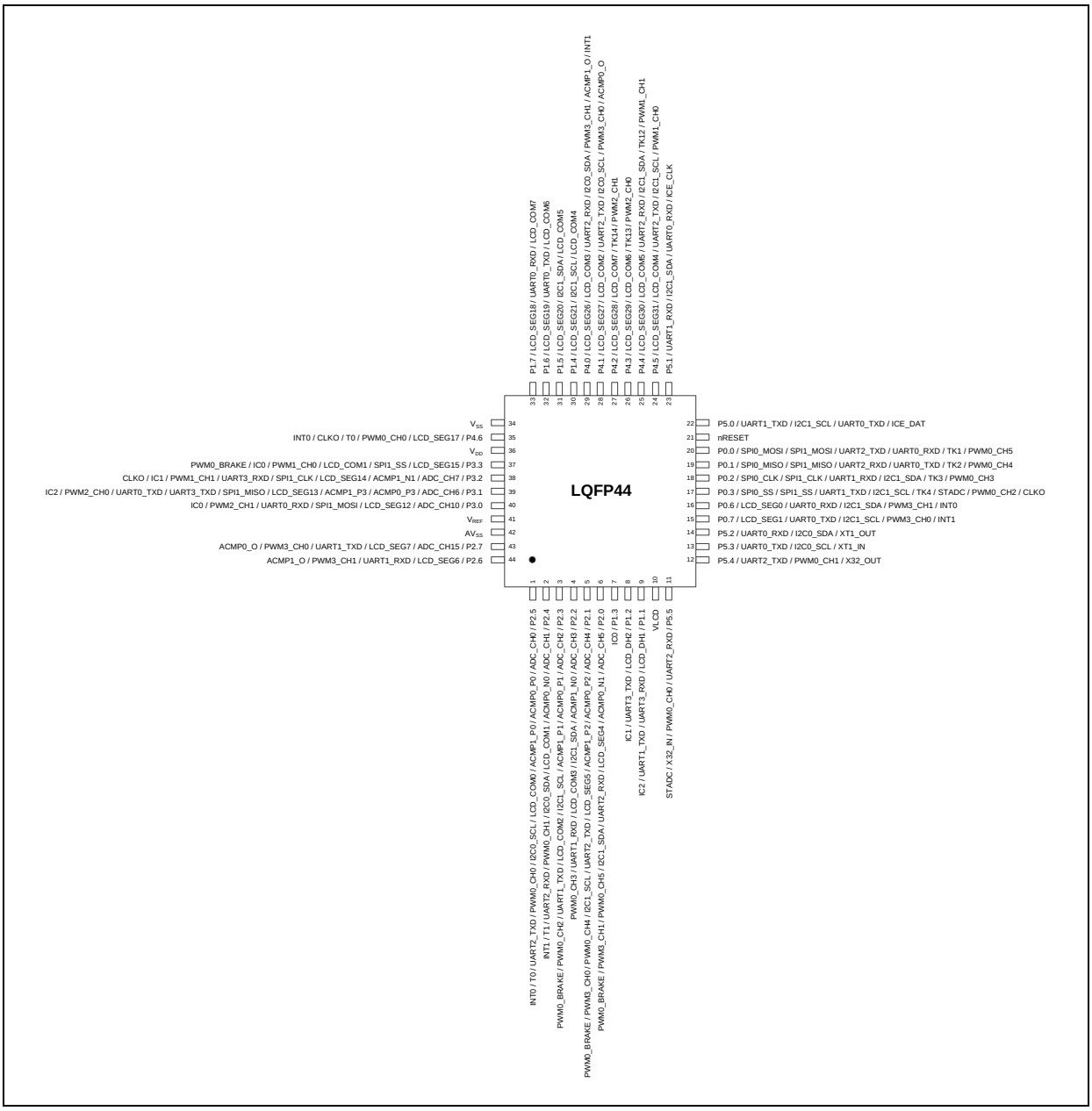


Figure 4.1-22 ML56MD1AE Multi-Function Pin assignment

Pin	ML56MD1AE Pin Function
1	P2.5/ADC_CH0/ACMP0_P0/ACMP1_P0/LCD_COM0/I2C0_SCL/PWM0_CH0/UART2_TXD/T0/INT0
2	P2.4/ADC_CH1/ACMP0_N0/LCD_COM1/I2C0_SDA/PWM0_CH1/UART2_RXD/T1/INT1
3	P2.3/ADC_CH2/ACMP0_P1/ACMP1_P1/I2C1_SCL/LCD_COM2/UART1_TXD/PWM0_CH2/PWM0_BRAKE
4	P2.2/ADC_CH3/ACMP1_N0/I2C1_SDA/LCD_COM3/UART1_RXD/PWM0_CH3
5	P2.1/ADC_CH4/ACMP0_P2/ACMP1_P2/LCD_SEG5/UART2_TXD/I2C1_SCL/PWM0_CH4/PWM3_CH0/PWM0_BRAKE

Pin	ML56MD1AE Pin Function
6	P2.0/ADC_CH5/ACMP0_N1/LCD_SEG4/UART2_RXD/I2C1_SDA/PWM0_CH5/PWM3_CH1/PWM0_BRAKE
7	P1.3/IC0
8	P1.2/LCD_DH2/UART3_TXD/IC1
9	P1.1/LCD_DH1/UART3_RXD/UART1_TXD/IC2
10	VLCD
11	P5.5/UART2_RXD/PWM0_CH0/X32_IN/STADC
12	P5.4/UART2_TXD/PWM0_CH1/X32_OUT
13	P5.3/UART0_TXD/I2C0_SCL/XT1_IN
14	P5.2/UART0_RXD/I2C0_SDA/XT1_OUT
15	P0.7/LCD_SEG1/UART0_TXD/I2C1_SCL/PWM3_CH0/INT1
16	P0.6/LCD_SEG0/UART0_RXD/I2C1_SDA/PWM3_CH1/INT0
17	P0.3/SPI0_SS/SPI1_SS/UART1_TXD/I2C1_SCL/TK4/STADC/PWM0_CH2/CLKO
18	P0.2/SPI0_CLK/SPI1_CLK/UART1_RXD/I2C1_SDA/TK3/PWM0_CH3
19	P0.1/SPI0_MISO/SPI1_MISO/UART2_RXD/UART0_TXD/TK2/PWM0_CH4
20	P0.0/SPI0_MOSI/SPI1_MOSI/UART2_TXD/UART0_RXD/TK1/PWM0_CH5
21	nRESET
22	P5.0/UART1_TXD/I2C1_SCL/UART0_TXD/ICE_DAT
23	P5.1/UART1_RXD/I2C1_SDA/UART0_RXD/ICE_CLK
24	P4.5/LCD_SEG31/LCD_COM4/UART2_TXD/I2C1_SCL/PWM1_CH0
25	P4.4/LCD_SEG30/LCD_COM5/UART2_RXD/I2C1_SDA/TK12/PWM1_CH1
26	P4.3/LCD_SEG29/LCD_COM6/TK13/PWM2_CH0
27	P4.2/LCD_SEG28/LCD_COM7/TK14/PWM2_CH1
28	P4.1/LCD_SEG27/LCD_COM2/UART2_TXD/I2C0_SCL/PWM3_CH0/ACMP0_O
29	P4.0/LCD_SEG26/LCD_COM3/UART2_RXD/I2C0_SDA/PWM3_CH1/ACMP1_O/INT1
30	P1.4/LCD_SEG21/I2C1_SCL/LCD_COM4
31	P1.5/LCD_SEG20/I2C1_SDA/LCD_COM5
32	P1.6/LCD_SEG19/UART0_TXD/LCD_COM6
33	P1.7/LCD_SEG18/UART0_RXD/LCD_COM7
34	VSS
35	P4.6/LCD_SEG17/PWM0_CH0/T0/CLKO/INT0
36	V_{DD}
37	P3.3/LCD_SEG15/SPI1_SS/LCD_COM1/PWM1_CH0/IC0/PWM0_BRAKE
38	P3.2/ADC_CH7/ACMP1_N1/LCD_SEG14/SPI1_CLK/UART3_RXD/PWM1_CH1/IC1/CLKO
39	P3.1/ADC_CH6/ACMP0_P3/ACMP1_P3/LCD_SEG13/SPI1_MISO/UART3_TXD/UART0_TXD/PWM2_CH0/IC2

Pin	ML56MD1AE Pin Function
40	P3.0/ADC_CH10/LCD_SEG12/SPI1_MOSI/UART0_RXD/PWM2_CH1/IC0
41	V_{REF}
42	AVSS
43	P2.7/ADC_CH15/LCD_SEG7/UART1_TXD/PWM3_CH0/ACMP0_O
44	P2.6/LCD_SEG6/UART1_RXD/PWM3_CH1/ACMP1_O

4.1.2.4 QFN33 Package

ML51TD1AE Pin Function

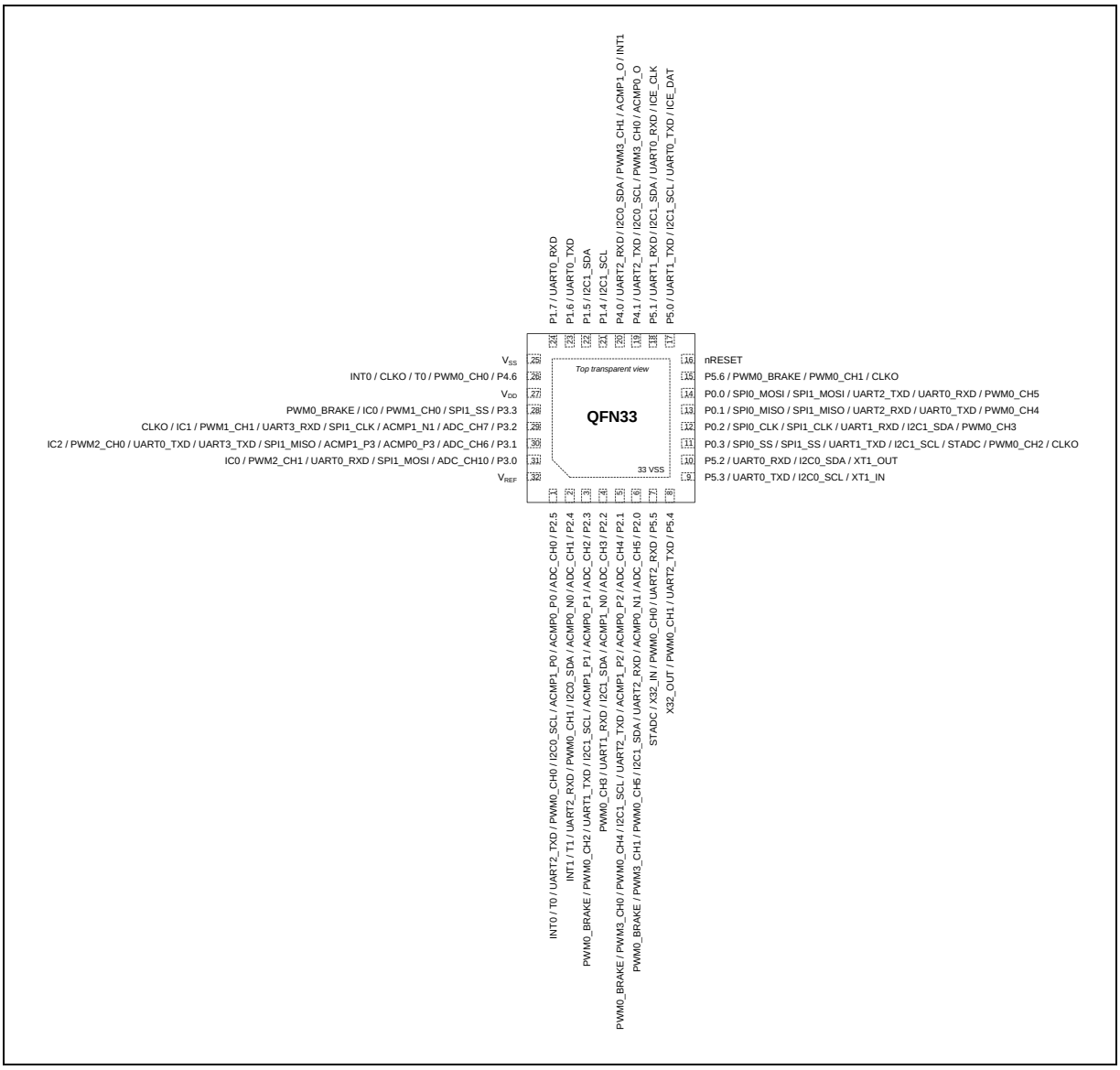


Figure 4.1-23 ML51TD1AE Multi-Function Pin assignment

Pin	ML56TD1AE Pin Function
1	P2.5/ADC_CH0/ACMP0_P0/ACMP1_P0/LCD_COM0/I2C0_SCL/PWM0_CH0/UART2_TXD/T0/INT0
2	P2.4/ADC_CH1/ACMP0_N0/LCD_COM1/I2C0_SDA/PWM0_CH1/UART2_RXD/T1/INT1
3	P2.3/ADC_CH2/ACMP0_P1/ACMP1_P1/I2C1_SCL/LCD_COM2/UART1_TXD/PWM0_CH2/PWM0_BRAKE
4	P2.2/ADC_CH3/ACMP1_N0/I2C1_SDA/LCD_COM3/UART1_RXD/PWM0_CH3
5	P2.1/ADC_CH4/ACMP0_P2/ACMP1_P2/LCD_SEG5/UART2_TXD/I2C1_SCL/PWM0_CH4/PWM3_CH0/PWM0_BRAKE
6	P2.0/ADC_CH5/ACMP0_N1/LCD_SEG4/UART2_RXD/I2C1_SDA/PWM0_CH5/PWM3_CH1/PWM0_BRAKE

Pin	ML56TD1AE Pin Function
7	P5.5/UART2_RXD/PWM0_CH0/X32_IN/STADC
8	P5.4/UART2_TXD/PWM0_CH1/X32_OUT
9	P5.3/UART0_TXD/I2C0_SCL/XT1_IN
10	P5.2/UART0_RXD/I2C0_SDA/XT1_OUT
11	P0.3/SPI0_SS/SPI1_SS/UART1_TXD/I2C1_SCL/TK4/STADC/PWM0_CH2/CLKO
12	P0.2/SPI0_CLK/SPI1_CLK/UART1_RXD/I2C1_SDA/TK3/PWM0_CH3
13	P0.1/SPI0_MISO/SPI1_MISO/UART2_RXD/UART0_TXD/TK2/PWM0_CH4
14	P0.0/SPI0_MOSI/SPI1_MOSI/UART2_TXD/UART0_RXD/TK1/PWM0_CH5
15	P5.6/TK0/PWM0_BRAKE/PWM0_CH1/CLKO
16	nRESET
17	P5.0/UART1_TXD/I2C1_SCL/UART0_TXD/ICE_DAT
18	P5.1/UART1_RXD/I2C1_SDA/UART0_RXD/ICE_CLK
19	P4.1/LCD_SEG27/LCD_COM2/UART2_TXD/I2C0_SCL/PWM3_CH0/ACMP0_O
20	P4.0/LCD_SEG26/LCD_COM3/UART2_RXD/I2C0_SDA/PWM3_CH1/ACMP1_O/INT1
21	P1.4/LCD_SEG21/I2C1_SCL/LCD_COM4
22	P1.5/LCD_SEG20/I2C1_SDA/LCD_COM5
23	P1.6/LCD_SEG19/UART0_TXD/LCD_COM6
24	P1.7/LCD_SEG18/UART0_RXD/LCD_COM7
25	VSS
26	P4.6/LCD_SEG17/PWM0_CH0/T0/CLKO/INT0
27	V_{DD}
28	P3.3/LCD_SEG15/SPI1_SS/LCD_COM1/PWM1_CH0/IC0/PWM0_BRAKE
29	P3.2/ADC_CH7/ACMP1_N1/LCD_SEG14/SPI1_CLK/UART3_RXD/PWM1_CH1/IC1/CLKO
30	P3.1/ADC_CH6/ACMP0_P3/ACMP1_P3/LCD_SEG13/SPI1_MISO/UART3_TXD/UART0_TXD/PWM2_CH0/IC2
31	P3.0/ADC_CH10/LCD_SEG12/SPI1_MOSI/UART0_RXD/PWM2_CH1/IC0
32	V_{REF}
33	VSS

ML51TC0AE / ML51TB9AE Pin Function

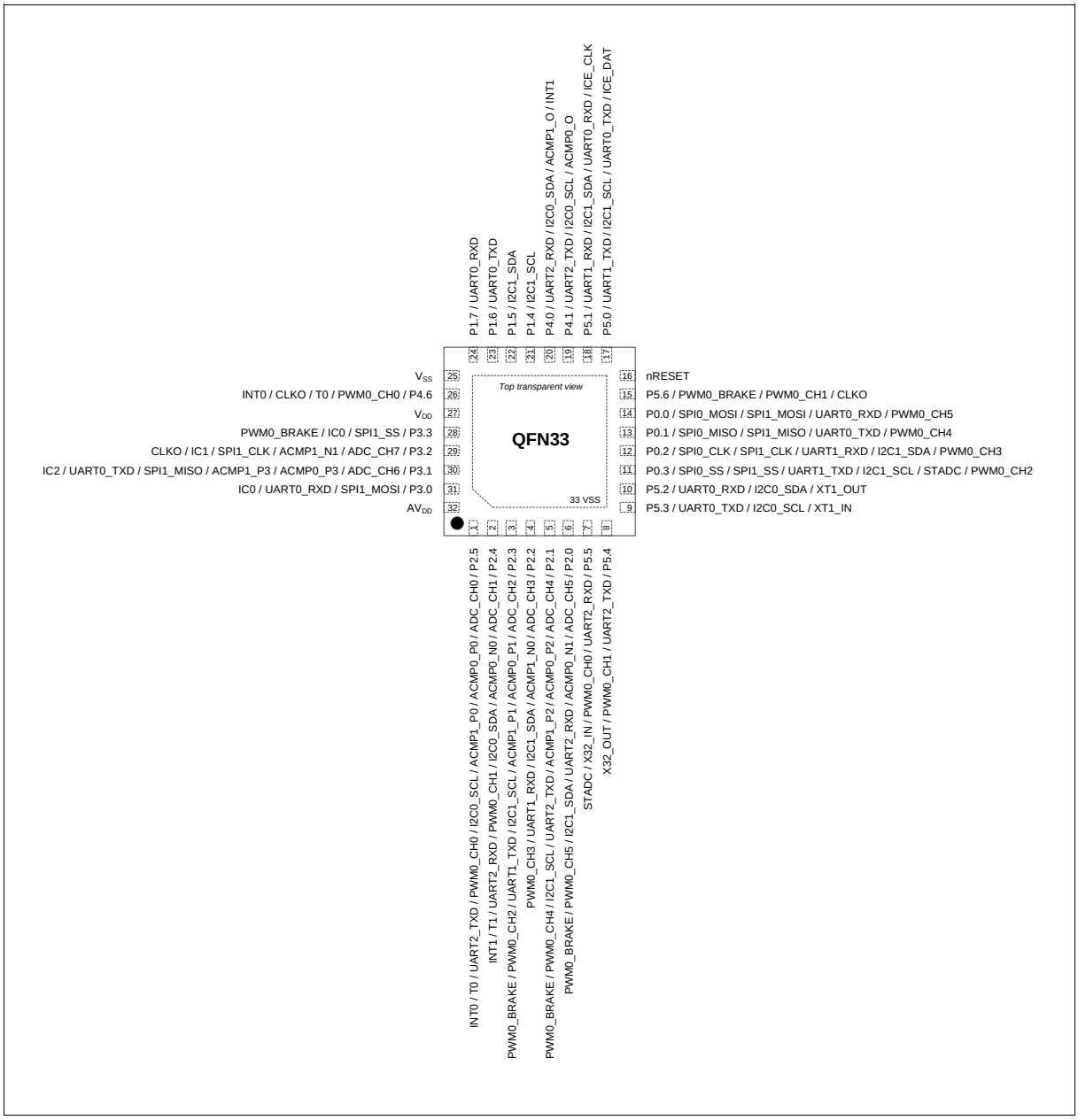


Figure 4.1-24 ML51TC0AE / ML51TB9AE Multi-Function Pin Assignment

Pin	ML51TC0AE / ML51TB9AE Pin Function
1	P2.5 / ADC_CH0 / ACMP0_P0 / ACMP1_P0 / I2C0_SCL / PWM0_CH0 / UART2_TXD / T0 / INT0
2	P2.4 / ADC_CH1 / ACMP0_N0 / I2C0_SDA / PWM0_CH1 / UART2_RXD / T1 / INT1
3	P2.3 / ADC_CH2 / ACMP0_P1 / ACMP1_P1 / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE
4	P2.2 / ADC_CH3 / ACMP1_N0 / I2C1_SDA / UART1_RXD / PWM0_CH3
5	P2.1 / ADC_CH4 / ACMP0_P2 / ACMP1_P2 / UART2_TXD / I2C1_SCL / PWM0_CH4 / PWM0_BRAKE
6	P2.0 / ADC_CH5 / ACMP0_N1 / UART2_RXD / I2C1_SDA / PWM0_CH5 / PWM0_BRAKE

Pin	ML51TC0AE / ML51TB9AE Pin Function
7	P5.5 / UART2_RXD / PWM0_CH0 / X32_IN / STADC
8	P5.4 / UART2_TXD / PWM0_CH1 / X32_OUT
9	P5.3 / UART0_TXD / I2C0_SCL / XT1_IN
10	P5.2 / UART0_RXD / I2C0_SDA / XT1_OUT
11	P0.3 / SPI0_SS / SPI1_SS / UART1_TXD / I2C1_SCL / STADC / PWM0_CH2
12	P0.2 / SPI0_CLK / SPI1_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
13	P0.1 / SPI0_MISO / SPI1_MISO / UART0_TXD / PWM0_CH4
14	P0.0 / SPI0_MOSI / SPI1_MOSI / UART0_RXD / PWM0_CH5
15	P5.6 / PWM0_BRAKE / PWM0_CH1 / CLKO
16	nRESET
17	P5.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
18	P5.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
19	P4.1 / UART2_TXD / I2C0_SCL / ACMP0_O
20	P4.0 / UART2_RXD / I2C0_SDA / ACMP1_O / INT1
21	P1.4 / I2C1_SCL
22	P1.5 / I2C1_SDA
23	P1.6 / UART0_TXD
24	P1.7 / UART0_RXD
25	V_{SS}
26	P4.6 / PWM0_CH0 / T0 / CLKO / INT0
27	V_{DD}
28	P3.3 / SPI1_SS / IC0 / PWM0_BRAKE
29	P3.2 / ADC_CH7 / ACMP1_N1 / SPI1_CLK / IC1 / CLKO
30	P3.1 / ADC_CH6 / ACMP0_P3 / ACMP1_P3 / SPI1_MISO / UART0_TXD / IC2
31	P3.0 / SPI1_MOSI / UART0_RXD / IC0
32	AV_{DD}
33	VSS

4.1.2.5 LQFP32 Package

ML51PC0AE / ML51PB9AE Pin Function

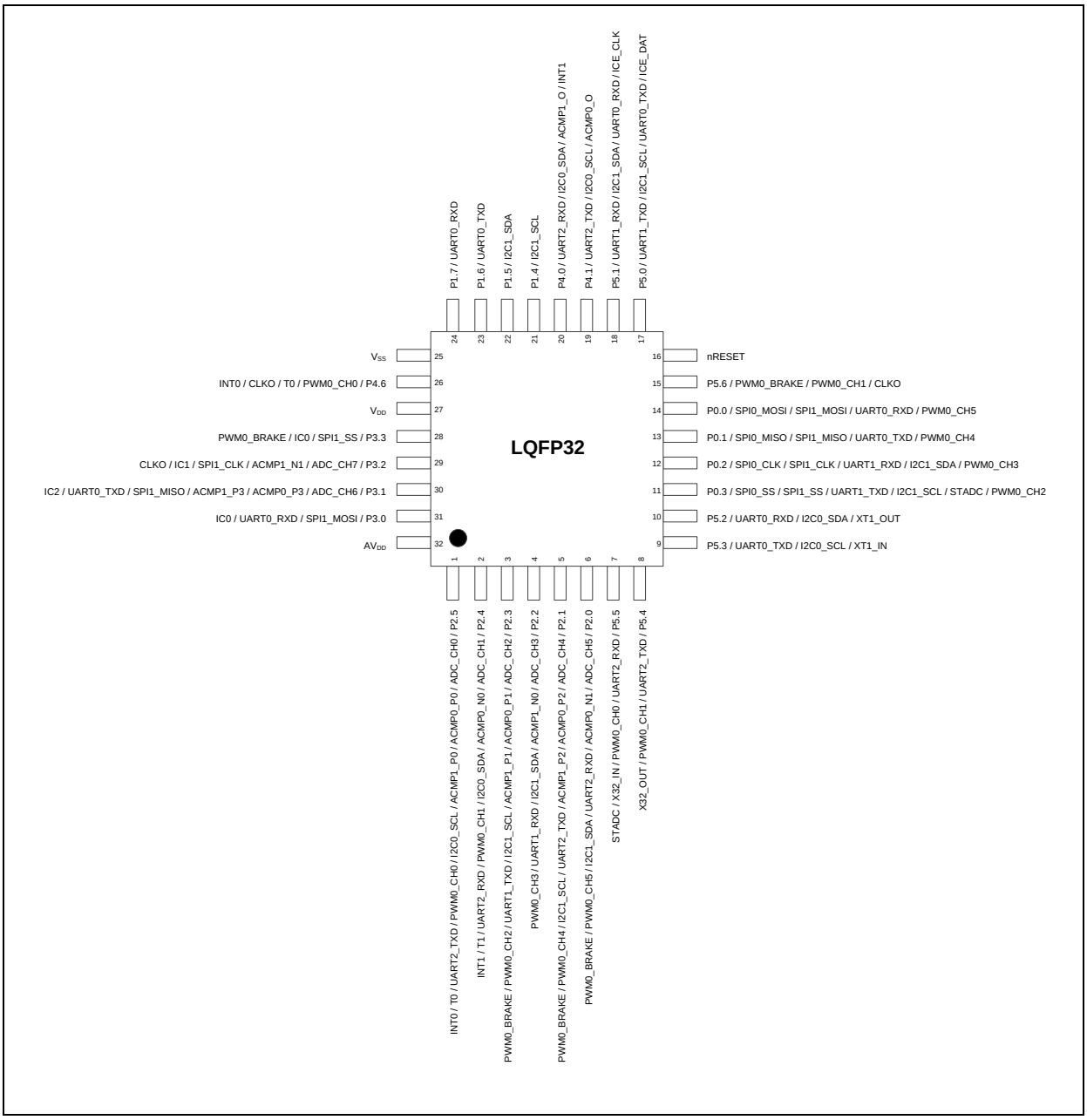


Figure 4.1-25 ML51PC0AE / ML51PB9AE Multi-Function Pin Assignment

Pin	ML51PC0AE Pin Function
1	P2.5 / ADC_CH0 / ACMP0_P0 / ACMP1_P0 / I2C0_SCL / PWM0_CH0 / UART2_TXD / T0 / INT0
2	P2.4 / ADC_CH1 / ACMP0_N0 / I2C0_SDA / PWM0_CH1 / UART2_RXD / T1 / INT1
3	P2.3 / ADC_CH2 / ACMP0_P1 / ACMP1_P1 / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE
4	P2.2 / ADC_CH3 / ACMP1_N0 / I2C1_SDA / UART1_RXD / PWM0_CH3

Pin	ML51PC0AE Pin Function
5	P2.1 / ADC_CH4 / ACMP0_P2 / ACMP1_P2 / UART2_TXD / I2C1_SCL / PWM0_CH4 / PWM0_BRAKE
6	P2.0 / ADC_CH5 / ACMP0_N1 / UART2_RXD / I2C1_SDA / PWM0_CH5 / PWM0_BRAKE
7	P5.5 / UART2_RXD / PWM0_CH0 / X32_IN / STADC
8	P5.4 / UART2_TXD / PWM0_CH1 / X32_OUT
9	P5.3 / UART0_TXD / I2C0_SCL / XT1_IN
10	P5.2 / UART0_RXD / I2C0_SDA / XT1_OUT
11	P0.3 / SPI0_SS / SPI1_SS / UART1_TXD / I2C1_SCL / STADC / PWM0_CH2
12	P0.2 / SPI0_CLK / SPI1_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
13	P0.1 / SPI0_MISO / SPI1_MISO / UART0_TXD / PWM0_CH4
14	P0.0 / SPI0_MOSI / SPI1_MOSI / UART0_RXD / PWM0_CH5
15	P5.6 / PWM0_BRAKE / PWM0_CH1 / CLKO
16	nRESET
17	P5.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
18	P5.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
19	P4.1 / UART2_TXD / I2C0_SCL / ACMP0_O
20	P4.0 / UART2_RXD / I2C0_SDA / ACMP1_O / INT1
21	P1.4 / I2C1_SCL
22	P1.5 / I2C1_SDA
23	P1.6 / UART0_TXD
24	P1.7 / UART0_RXD
25	V_{SS}
26	P4.6 / PWM0_CH0 / T0 / CLKO / INT0
27	V_{DD}
28	P3.3 / SPI1_SS / IC0 / PWM0_BRAKE
29	P3.2 / ADC_CH7 / ACMP1_N1 / SPI1_CLK / IC1 / CLKO
30	P3.1 / ADC_CH6 / ACMP0_P3 / ACMP1_P3 / SPI1_MISO / UART0_TXD / IC2
31	P3.0 / SPI1_MOSI / UART0_RXD / IC0
32	AV_{DD}

4.1.2.6 TSSOP28 Package

ML51EC0AE / ML51EB9AE Pin Function

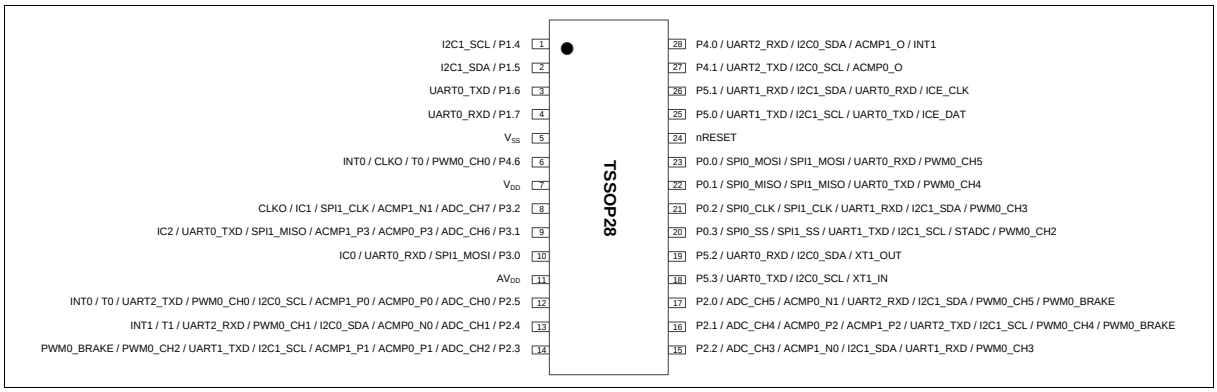


Figure 4.1-26 ML51EC0AE / ML51EB9AE Multi-Function Pin Assignment

Pin	ML51EC0AE / ML51EB9AE Pin Function
1	P1.4 / I2C1_SCL
2	P1.5 / I2C1_SDA
3	P1.6 / UART0_TXD
4	P1.7 / UART0_RXD
5	V_{SS}
6	P4.6 / PWM0_CH0 / T0 / CLKO / INT0
7	V_{DD}
8	P3.2 / ADC_CH7 / ACMP1_N1 / SPI1_CLK / IC1 / CLKO
9	P3.1 / ADC_CH6 / ACMP0_P3 / ACMP1_P3 / SPI1_MISO / UART0_TXD / IC2
10	P3.0 / SPI1_MOSI / UART0_RXD / IC0
11	AV_{DD}
12	P2.5 / ADC_CH0 / ACMP0_P0 / ACMP1_P0 / I2C0_SCL / PWM0_CH0 / UART2_TXD / T0 / INT0
13	P2.4 / ADC_CH1 / ACMP0_N0 / I2C0_SDA / PWM0_CH1 / UART2_RXD / T1 / INT1
14	P2.3 / ADC_CH2 / ACMP0_P1 / ACMP1_P1 / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE
15	P2.2 / ADC_CH3 / ACMP1_N0 / I2C1_SDA / UART1_RXD / PWM0_CH3
16	P2.1 / ADC_CH4 / ACMP0_P2 / ACMP1_P2 / UART2_TXD / I2C1_SCL / PWM0_CH4 / PWM0_BRAKE
17	P2.0 / ADC_CH5 / ACMP0_N1 / UART2_RXD / I2C1_SDA / PWM0_CH5 / PWM0_BRAKE
18	P5.3 / UART0_TXD / I2C0_SCL / XT1_IN
19	P5.2 / UART0_RXD / I2C0_SDA / XT1_OUT
20	P0.3 / SPI0_SS / SPI1_SS / UART1_TXD / I2C1_SCL / STADC / PWM0_CH2
21	P0.2 / SPI0_CLK / SPI1_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3

Pin	ML51EC0AE / ML51EB9AE Pin Function
22	P0.1 / SPI0_MISO / SPI1_MISO / UART0_TXD / PWM0_CH4
23	P0.0 / SPI0_MOSI / SPI1_MOSI / UART0_RXD / PWM0_CH5
24	nRESET
25	P5.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
26	P5.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
27	P4.1 / UART2_TXD / I2C0_SCL / ACMP0_O
28	P4.0 / UART2_RXD / I2C0_SDA / ACMP1_O / INT1

4.1.2.7 SOP28 Package

Corresponding Part Number: ML51UC0AE / ML51UB9AE

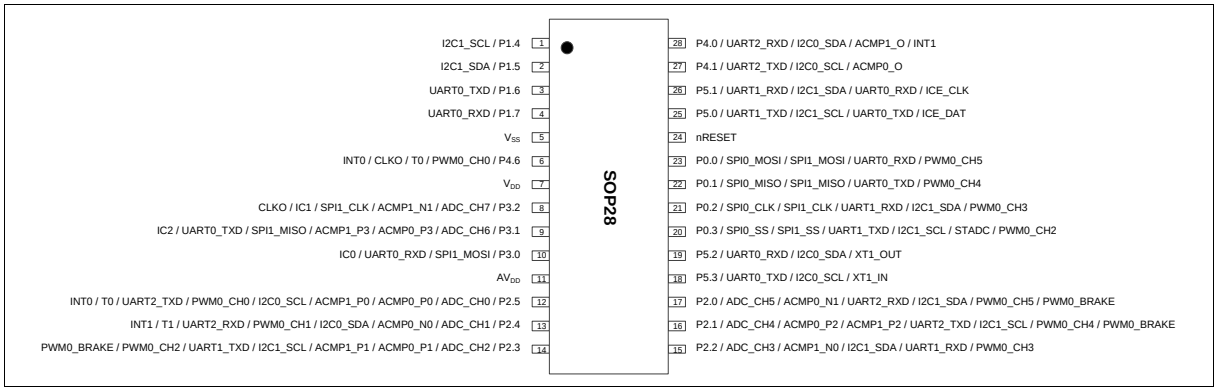


Figure 4.1-27 ML51UC0AE / ML51UB9AE Multi Function Pin Assignment

ML51UC0AE / ML51UB9AE Pin Function

Pin	ML51UC0AE / ML51UB9AE Pin Function
1	P1.4 / I2C1_SCL
2	P1.5 / I2C1_SDA
3	P1.6 / UART0_TXD
4	P1.7 / UART0_RXD
5	V_{SS}
6	P4.6 / PWM0_CH0 / T0 / CLKO / INT0
7	V_{DD}
8	P3.2 / ADC_CH7 / ACMP1_N1 / SPI1_CLK / IC1 / CLKO
9	P3.1 / ADC_CH6 / ACMP0_P3 / ACMP1_P3 / SPI1_MISO / UART0_TXD / IC2
10	P3.0 / SPI1_MOSI / UART0_RXD / IC0
11	AV_{DD}
12	P2.5 / ADC_CH0 / ACMP0_P0 / ACMP1_P0 / I2C0_SCL / PWM0_CH0 / UART2_TXD / T0 / INT0
13	P2.4 / ADC_CH1 / ACMP0_N0 / I2C0_SDA / PWM0_CH1 / UART2_RXD / T1 / INT1
14	P2.3 / ADC_CH2 / ACMP0_P1 / ACMP1_P1 / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE
15	P2.2 / ADC_CH3 / ACMP1_N0 / I2C1_SDA / UART1_RXD / PWM0_CH3
16	P2.1 / ADC_CH4 / ACMP0_P2 / ACMP1_P2 / UART2_TXD / I2C1_SCL / PWM0_CH4 / PWM0_BRAKE
17	P2.0 / ADC_CH5 / ACMP0_N1 / UART2_RXD / I2C1_SDA / PWM0_CH5 / PWM0_BRAKE
18	P5.3 / UART0_TXD / I2C0_SCL / XT1_IN
19	P5.2 / UART0_RXD / I2C0_SDA / XT1_OUT
20	P0.3 / SPI0_SS / SPI1_SS / UART1_TXD / I2C1_SCL / STADC / PWM0_CH2

Pin	ML51UC0AE / ML51UB9AE Pin Function
21	P0.2 / SPI0_CLK / SPI1_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
22	P0.1 / SPI0_MISO / SPI1_MISO / UART0_TXD / PWM0_CH4
23	P0.0 / SPI0_MOSI / SPI1_MOSI / UART0_RXD / PWM0_CH5
24	nRESET
25	P5.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
26	P5.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
27	P4.1 / UART2_TXD / I2C0_SCL / ACMP0_O
28	P4.0 / UART2_RXD / I2C0_SDA / ACMP1_O / INT1

4.1.2.8 TSSOP20 Package

ML51FB9AE Pin Function

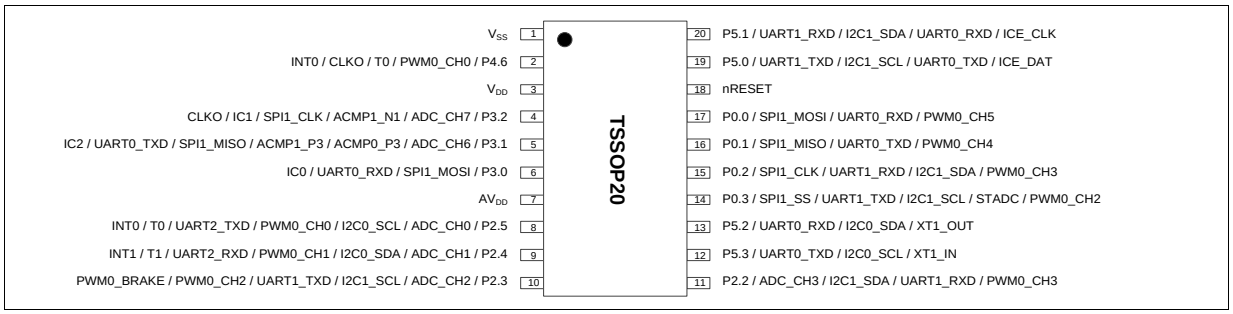


Figure 4.1-28 ML51FB9AE Multi Function Pin Assignment

Pin	ML51FB9AE Pin Function
1	V_{SS}
2	P4.6 / PWM0_CH0 / T0 / CLKO / INT0
3	V_{DD}
4	P3.2 / ADC_CH7 / ACMP1_N1 / SPI1_CLK / IC1 / CLKO
5	P3.1 / ADC_CH6 / ACMP0_P3 / ACMP1_P3 / SPI1_MISO / UART0_TXD / IC2
6	P3.0 / SPI1_MOSI / UART0_RXD / IC0
7	AV_{DD}
8	P2.5 / ADC_CH0 / I2C0_SCL / PWM0_CH0 / UART2_TXD / T0 / INT0
9	P2.4 / ADC_CH1 / I2C0_SDA / PWM0_CH1 / UART2_RXD / T1 / INT1
10	P2.3 / ADC_CH2 / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE
11	P2.2 / ADC_CH3 / I2C1_SDA / UART1_RXD / PWM0_CH3
12	P5.3 / UART0_TXD / I2C0_SCL / XT1_IN
13	P5.2 / UART0_RXD / I2C0_SDA / XT1_OUT
14	P0.3 / SPI1_SS / UART1_TXD / I2C1_SCL / STADC / PWM0_CH2
15	P0.2 / SPI1_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
16	P0.1 / SPI1_MISO / UART0_TXD / PWM0_CH4
17	P0.0 / SPI1_MOSI / UART0_RXD / PWM0_CH5
18	nRESET
19	P5.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
20	P5.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK

4.1.2.9 SOP20 Package

ML51OB9AE Pin Function

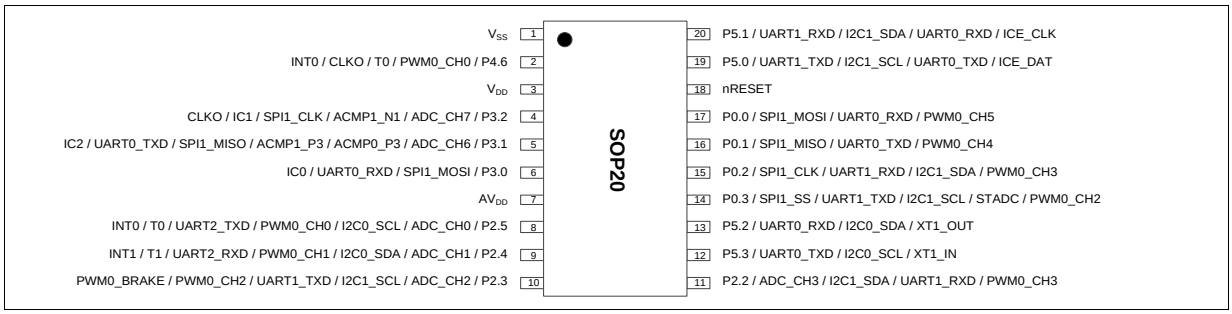


Figure 4.1-29 ML51OB9AE Multi Function Pin Assignment

Pin	ML51OB9AE Pin Function
1	V_{SS}
2	P4.6 / PWM0_CH0 / T0 / CLKO / INT0
3	V_{DD}
4	P3.2 / ADC_CH7 / ACMP1_N1 / SPI1_CLK / IC1 / CLKO
5	P3.1 / ADC_CH6 / ACMP0_P3 / ACMP1_P3 / SPI1_MISO / UART0_TXD / IC2
6	P3.0 / SPI1_MOSI / UART0_RXD / IC0
7	AV_{DD}
8	P2.5 / ADC_CH0 / I2C0_SCL / PWM0_CH0 / UART2_TXD / T0 / INT0
9	P2.4 / ADC_CH1 / I2C0_SDA / PWM0_CH1 / UART2_RXD / T1 / INT1
10	P2.3 / ADC_CH2 / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE
11	P2.2 / ADC_CH3 / I2C1_SDA / UART1_RXD / PWM0_CH3
12	P5.3 / UART0_TXD / I2C0_SCL / XT1_IN
13	P5.2 / UART0_RXD / I2C0_SDA / XT1_OUT
14	P0.3 / SPI1_SS / UART1_TXD / I2C1_SCL / STADC / PWM0_CH2
15	P0.2 / SPI1_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
16	P0.1 / SPI1_MISO / UART0_TXD / PWM0_CH4
17	P0.0 / SPI1_MOSI / UART0_RXD / PWM0_CH5
18	nRESET
19	P5.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
20	P5.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK

4.1.2.10 QFN20 Package

ML51XB9AE Pin Function

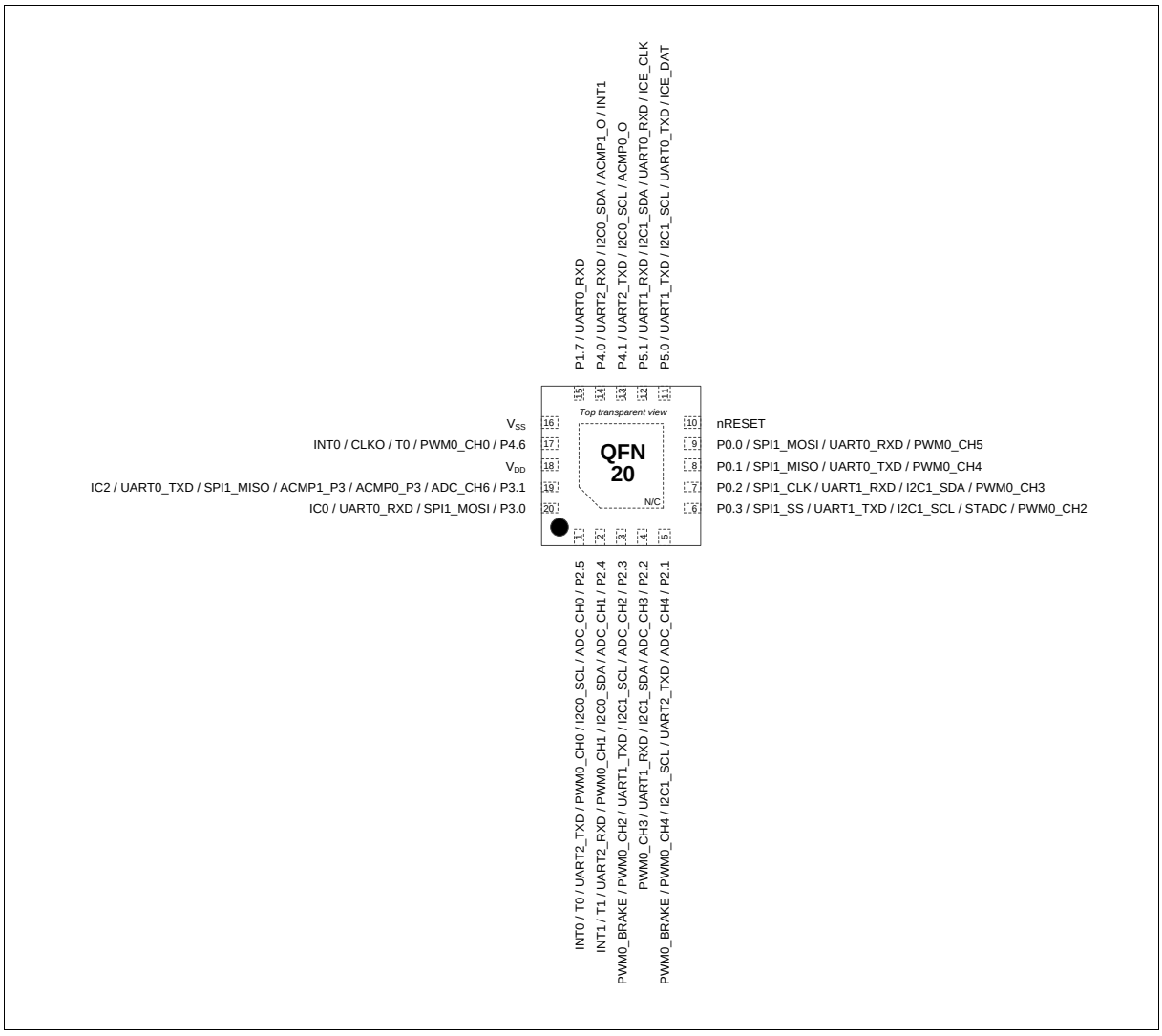


Figure 4.1-30 ML51XB9AE Multi Function Pin Assignment

Pin	ML51XB9AE Pin Function
1	P2.5 / ADC_CH0 / I2C0_SCL / PWM0_CH0 / UART2_TXD / T0 / INT0
2	P2.4 / ADC_CH1 / I2C0_SDA / PWM0_CH1 / UART2_RXD / T1 / INT1
3	P2.3 / ADC_CH2 / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE
4	P2.2 / ADC_CH3 / I2C1_SDA / UART1_RXD / PWM0_CH3
5	P2.1 / ADC_CH4 / UART2_TXD / I2C1_SCL / PWM0_CH4 / PWM0_BRAKE
6	P0.3 / SPI1_SS / UART1_TXD / I2C1_SCL / STADC / PWM0_CH2
7	P0.2 / SPI1_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
8	P0.1 / SPI1_MISO / UART0_TXD / PWM0_CH4

Pin	ML51XB9AE Pin Function
9	P0.0 / SPI1_MOSI / UART0_RXD / PWM0_CH5
10	nRESET
11	P5.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
12	P5.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
13	P4.1 / UART2_TXD / I2C0_SCL / ACMP0_O
14	P4.0 / UART2_RXD / I2C0_SDA / ACMP1_O / INT1
15	P1.7 / UART0_RXD
16	V_{SS}
17	P4.6 / PWM0_CH0 / T0 / CLKO / INT0
18	V_{DD}
19	P3.1 / ADC_CH6 / ACMP0_P3 / ACMP1_P3 / SPI1_MISO / UART0_TXD / IC2
20	P3.0 / SPI1_MOSI / UART0_RXD / IC0

4.1.2.11 TSSOP14 Package

ML51DB9AE Pin Function

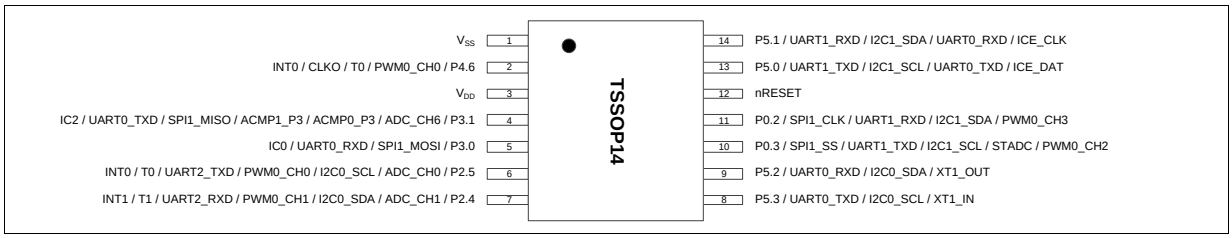


Figure 4.1-31 ML51DB9AE Multi Function Pin Assignment

Pin	ML51DB9AE Pin Function
1	V_{SS}
2	P4.6 / PWM0_CH0 / T0 / CLKO / INT0
3	V_{DD}
4	P3.1 / ADC_CH6 / ACMP0_P3 / ACMP1_P3 / SPI1_MISO / UART0_TXD / IC2
5	P3.0 / SPI1_MOSI / UART0_RXD / IC0
6	P2.5 / ADC_CH0 / I2C0_SCL / PWM0_CH0 / UART2_TXD / T0 / INT0
7	P2.4 / ADC_CH1 / I2C0_SDA / PWM0_CH1 / UART2_RXD / T1 / INT1
8	P5.3 / UART0_TXD / I2C0_SCL / XT1_IN
9	P5.2 / UART0_RXD / I2C0_SDA / XT1_OUT
10	P0.3 / SPI1_SS / UART1_TXD / I2C1_SCL / STADC / PWM0_CH2
11	P0.2 / SPI1_CLK / UART1_RXD / I2C1_SDA / PWM0_CH3
12	nRESET
13	P5.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
14	P5.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK

4.1.2.12 MSOP10 Package

ML51BB9AE Pin Function

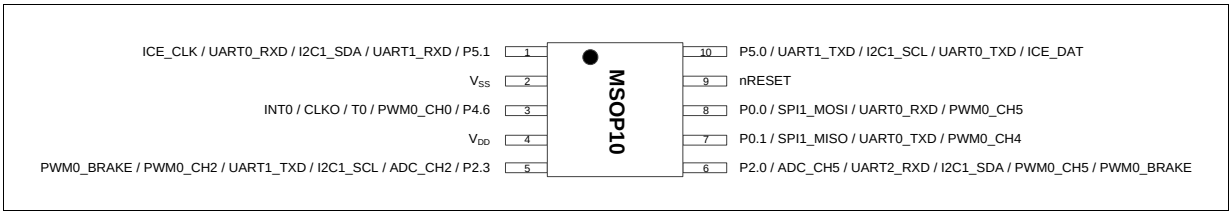


Figure 4.1-32 ML51BB9AE Pin Assignment

Pin	ML51BB9AE Pin Function
1	P5.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
2	V_{SS}
3	P4.6 / PWM0_CH0 / T0 / CLKO / INT0
4	V_{DD}
5	P2.3 / ADC_CH2 / I2C1_SCL / UART1_TXD / PWM0_CH2 / PWM0_BRAKE
6	P2.0 / ADC_CH5 / UART2_RXD / I2C1_SDA / PWM0_CH5 / PWM0_BRAKE
7	P0.1 / SPI1_MISO / UART0_TXD / PWM0_CH4
8	P0.0 / SPI1_MOSI / UART0_RXD / PWM0_CH5
9	nRESET
10	P5.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT

4.2 Pin Description

4.2.1 ML51/ML54/ML56 Series Pin Mapping

	ML54/ML56			ML51							
Pin Number	64	48	44	64	48	33/32	28	20	QFN20	14	10
P2.6	1	48	44	1	48						
P2.5	2	1	1	2	1	1	12	8	1	6	
P2.4	3	2	2	3	2	2	13	9	2	7	
P2.3	4	3	3	4	3	3	14	10	3		5
P2.2	5	4	4	5	4	4	15	11	4		
P2.1	6	5	5	6	5	5	16		5		
P2.0	7	6	6	7	6	6	17				6
P1.3	8	7	7	8	7						
P1.2	9	8	8	9	8						
P1.1	10	9	9	10	9						
P1.0	11			11	10						
VLCD	12	10	10								
P3.7				12							
P5.7	13			13							
P5.5	14	11	11	14	11	7					
P5.4	15	12	12	15	12	8					
P5.3	16	13	13	16	13	9	18	12		8	
P5.2	17	14	14	17	14	10	19	13		9	
P3.5	18			18							
P3.4	19			19							
P0.7	20	15	15	20	15						
P0.6	21	16	16	21	16						
VSS	22			22		33					
V_{DD}	23			23							
P3.6	24			24							
P0.5	25	17		25	17						
P0.4	26	18		26	18						
P0.3	27	19	17	27	19	11	20	14	6	10	
P0.2	28	20	18	28	20	12	21	15	7	11	
P0.1	29	21	19	29	21	13	22	16	8		7
P0.0	30	22	20	30	22	14	23	17	9		8
P5.6	31	23		31	23	15					
nRESET	32	24	21	32	24	16	24	18	10	12	9

	ML54/ML56			ML51							
Pin Number	64	48	44	64	48	33/32	28	20	QFN20	14	10
P5.0	33	25	22	33	25	17	25	19	11	13	10
P5.1	34	26	23	34	26	18	26	20	12	14	1
P4.5	35	27	24	35	27						
P4.4	36	28	25	36	28						
P4.3	37	29	26	37	29						
P4.2	38	30	27	38	30						
P4.1	39	31	28	39	31	19	27		13		
P4.0	40	32	29	40	32	20	28		14		
P6.3	41			41							
P6.2	42			42							
P6.1	43			43							
P6.0	44			44							
P1.4	45	33	30	45	33	21	1				
P1.5	46	34	31	46	34	22	2				
P1.6	47	35	32	47	35	23	3				
P1.7	48	36	33	48	36	24	4		15		
VSS	49	37	34	49	37	25	5	1	16	1	2
P4.6	50	38	35	50	38	26	6	2	17	2	3
V_{DD}	51	39	36	51	39	27	7	3	18	3	4
P4.7	52	40		52	40						
P3.3	53	41	37	53	41	28					
P3.2	54	42	38	54	42	29	8	4			
P3.1	55	43	39	55	43	30	9	5	19	4	
P3.0	56	44	40	56	44	31	10	6	20	5	
AV_{DD}	57	39	36	57	39	27	7				
V_{REF}	58	45	41	58	45	32	11	7			
AVSS	59	46	42	59	46				16		
P6.7	60			60							
P6.6	61			61							
P6.5	62			62							
P6.4	63			63							
P2.7	64	47	43	64	47						

4.2.2 ML51/ML54/ML56 Series Pin Functional Description

As default all GPIO type is defined as input mode. User should setting the GPIO Mode by PxMx register.

A: Analog suggest disable digial function O: output, I: input, I/O: bi-direction (Quasi)

Group	Pin Name	Type	Description
ACMP0	ACMP0_N0	A	Analog comparator 0 negative input 0 pin.
	ACMP0_N1		Analog comparator 0 negative input 1 pin.
	ACMP0_O	O	Analog comparator 0 output pin.
	ACMP0_P0	A	Analog comparator 0 positive input 0 pin.
	ACMP0_P1		Analog comparator 0 positive input 1 pin.
	ACMP0_P2		Analog comparator 0 positive input 2 pin.
	ACMP0_P3		Analog comparator 0 positive input 3 pin.
ACMP1	ACMP1_N0	A	Analog comparator 1 negative input 0 pin.
	ACMP1_N1		Analog comparator 1 negative input 1 pin.
	ACMP1_O	O	Analog comparator 1 output pin.
	ACMP1_P0	A	Analog comparator 1 positive input 0 pin.
	ACMP1_P1		Analog comparator 1 positive input 1 pin.
	ACMP1_P2		Analog comparator 1 positive input 2 pin.
	ACMP1_P3		Analog comparator 1 positive input 3 pin.
ADC	ADC_CH0	A	ADC_ channel analog input.
	ADC_CH1		ADC_ channel analog input.
	ADC_CH2		ADC_ channel analog input.
	ADC_CH3		ADC_ channel analog input.
	ADC_CH4		ADC_ channel analog input.
	ADC_CH5		ADC_ channel analog input.
	ADC_CH6		ADC_ channel analog input.
	ADC_CH7		ADC_ channel analog input.
	ADC_CH10		ADC_ channel analog input.
	ADC_CH11		ADC_ channel analog input.
	ADC_CH12		ADC_ channel analog input.
	ADC_CH13		ADC_ channel analog input.
	ADC_CH14		ADC_ channel analog input.
	ADC_CH15		ADC_ channel analog input.
CLKO	CLKO	O	Clock Out
I2C0	I2C0_SCL	I/O	I2C0 clock pin.
	I2C0_SDA	I/O	I2C0 data input/output pin.

Group	Pin Name	Type	Description
I2C1	I2C1_SCL	I/O	I2C1 clock pin.
	I2C1_SDA	I/O	I2C1 data input/output pin.
IC0	IC0	I/O	Input Capture channel 0
IC1	IC1	I/O	Input Capture channel 1
IC2	IC2	I/O	Input Capture channel 2
ICE	ICE_CLK	I	Serial wired debugger clock pin. **Note:** It is recommended to use 100 kΩ pull-up resistor on ICE_CLK pin
	ICE_DAT	O	Serial wired debugger data pin. **Note:** It is recommended to use 100 kΩ pull-up resistor on ICE_DAT pin
INT0	INT0	I	External interrupt 0 input pin.
INT1	INT1	I	External interrupt 1 input pin.
LCD	LCD_COM0	O	LCD Common 0 output.
	LCD_COM1	O	LCD Common 1 output.
	LCD_COM2	O	LCD Common 2 output.
	LCD_COM3	O	LCD Common 3 output.
	LCD_COM4	O	LCD Common 4 output.
	LCD_COM5	O	LCD Common 5 output.
	LCD_COM6	O	LCD Common 6 output.
	LCD_COM7	O	LCD Common 7 output.
	LCD_DH1	O	LCD external capacitor pin of charge pump circuit.
	LCD_DH2	O	LCD external capacitor pin of charge pump circuit.
	LCD_SEG0	O	LCD segment 0 output
	LCD_SEG1	O	LCD segment 1 output
	LCD_SEG2	O	LCD segment 2 output
	LCD_SEG3	O	LCD segment 3 output
	LCD_SEG4	O	LCD segment 4 output
	LCD_SEG5	O	LCD segment 5 output
	LCD_SEG6	O	LCD segment 6 output
	LCD_SEG7	O	LCD segment 7 output
	LCD_SEG8	O	LCD segment 8 output
	LCD_SEG9	O	LCD segment 9 output
	LCD_SEG10	O	LCD segment 10 output
	LCD_SEG11	O	LCD segment 11 output
	LCD_SEG12	O	LCD segment 12 output

Group	Pin Name	Type	Description
	LCD_SEG13	O	LCD segment 13 output
	LCD_SEG14	O	LCD segment 14 output
	LCD_SEG15	O	LCD segment 15 output
	LCD_SEG16	O	LCD segment 16 output
	LCD_SEG17	O	LCD segment 17 output
	LCD_SEG18	O	LCD segment 18 output
	LCD_SEG19	O	LCD segment 19 output
	LCD_SEG20	O	LCD segment 20 output
	LCD_SEG21	O	LCD segment 21 output
	LCD_SEG22	O	LCD segment 22 output
	LCD_SEG23	O	LCD segment 23 output
	LCD_SEG24	O	LCD segment 24 output
	LCD_SEG25	O	LCD segment 25 output
	LCD_SEG26	O	LCD segment 26 output
	LCD_SEG27	O	LCD segment 27 output
	LCD_SEG28	O	LCD segment 28 output
	LCD_SEG29	O	LCD segment 29 output
	LCD_SEG30	O	LCD segment 30 output
	LCD_SEG31	O	LCD segment 31 output
	LCD_V1	I	Input pin of the 1st most positive LCD level.
	LCD_V2	I	Input pin of the 2nd most positive LCD level.
	LCD_V3	I	Input pin of the 3rd most positive LCD level.
nRESET	nRESET	I	External reset input: active LOW, with an internal pull-up. Set this pin low reset to initial state. **Note:** It is recommended to use 10 kΩ pull-up resistor and 10 uF capacitor on nRESET pin.
PWM0	PWM0_BRAKE	I	PWM0 Brake input pin.
	PWM0_CH0	I/O	PWM0 channel 0 output/capture input.
	PWM0_CH1	I/O	PWM0 channel 1 output/capture input.
	PWM0_CH2	I/O	PWM0 channel 2 output/capture input.
	PWM0_CH3	I/O	PWM0 channel 3 output/capture input.
	PWM0_CH4	I/O	PWM0 channel 4 output/capture input.
	PWM0_CH5	I/O	PWM0 channel 5 output/capture input.
PWM1	PWM1_CH0	I/O	PWM1 channel 0 output/capture input.
	PWM1_CH1	I/O	PWM1 channel 1 output/capture input.
PWM2	PWM2_CH0	I/O	PWM2 channel 0 output/capture input.

Group	Pin Name	Type	Description
	PWM2_CH1	I/O	PWM2 channel 1 output/capture input.
PWM3	PWM3_CH0	I/O	PWM3 channel 0 output/capture input.
	PWM3_CH1	I/O	PWM3 channel 1 output/capture input.
SPI0	SPI0_CLK	I/O	SPI0 serial clock pin.
	SPI0_MISO	I/O	SPI0 MISO (Master In, Slave Out) pin.
	SPI0_MOSI	I/O	SPI0 MOSI (Master Out, Slave In) pin.
	SPI0_SS	I/O	SPI0 slave select pin.
SPI1	SPI1_CLK	I/O	SPI1 serial clock pin.
	SPI1_MISO	I/O	SPI1 MISO (Master In, Slave Out) pin.
	SPI1_MOSI	I/O	SPI1 MOSI (Master Out, Slave In) pin.
	SPI1_SS	I/O	SPI1 slave select pin.
STADC	STADC	I	ADC external trigger input.
T0	T0	I/O	External count input to Timer/Counter 0 or its toggle output.
T1	T1	I/O	External count input to Timer/Counter 1 or its toggle output.
TK	TK0	A	Touch Key 0.
	TK1	A	Touch Key 1.
	TK2	A	Touch Key 2.
	TK3	A	Touch Key 3.
	TK4	A	Touch Key 4.
	TK5	A	Touch Key 5.
	TK6	A	Touch Key 6.
	TK7	A	Touch Key 7.
	TK8	A	Touch Key 8.
	TK9	A	Touch Key 9.
	TK10	A	Touch Key 10.
	TK11	A	Touch Key 11.
	TK12	A	Touch Key 12.
	TK13	A	Touch Key 13.
	TK14	A	Touch Key 14.
UART0	UART0_RXD	I	UART0 data receiver input pin.
	UART0_TXD	O	UART0 data transmitter output pin.
UART1	UART1_RXD	I	UART1 data receiver input pin.
	UART1_TXD	O	UART1 data transmitter output pin.
UART2	UART2_RXD	I	UART2 data receiver input pin.
	UART2_TXD	O	UART2 data transmitter output pin.

Group	Pin Name	Type	Description
UART3	UART3_RXD	I	UART3 data receiver input pin.
	UART3_TXD	O	UART3 data transmitter output pin.
V_{REF}	V_{REF}	A	ADC reference voltage input. **Note:** This pin needs to be connected with a 1uF capacitor when use internal voltage reference output.
X32	X32_IN	I	External 32.768 kHz crystal input pin.
	X32_OUT	O	External 32.768 kHz crystal output pin.
XT1	XT1_IN	I	External 4~24 MHz (high speed) crystal input pin.
	XT1_OUT	O	External 4~24 MHz (high speed) crystal output pin.

5 BLOCK DIAGRAM

5.1 ML51/ML54/ML56 Series Full Function Block

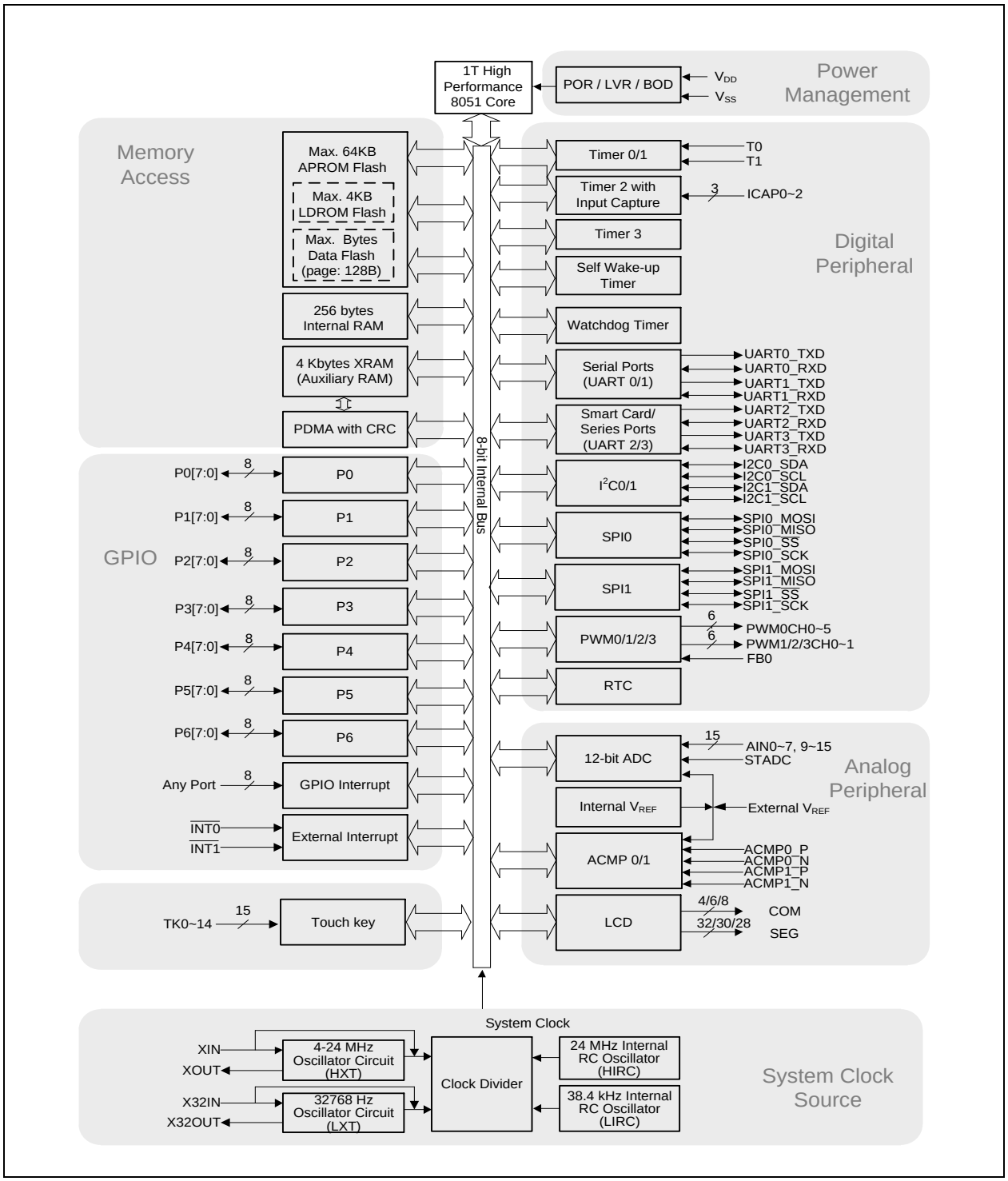


Figure 5.1-1 Functional Block Diagram

6 FUNCTIONAL DESCRIPTION

6.1 Memory Organization

A standard 80C51 based microcontroller divides the memory into two different sections, Program Memory and Data Memory. The Program Memory is used to store the instruction codes, whereas the Data Memory is used to store data or variations during the program execution.

The Data Memory occupies a separate address space from Program Memory. In ML51/ML54/ML56 Series, there are 256 bytes of internal scratch-pad RAM. For many applications those need more internal RAM, the ML51/ML54/ML56 Series provides another on-chip 4 Kbytes of RAM, which is called XRAM, accessed by MOVX instruction.

The whole embedded Flash, functioning as Program Memory, is divided into three blocks: Application ROM (APROM) normally for User Code, Loader ROM (LDROM) normally for Boot Code, and CONFIG bytes for hardware initialization. Actually, APROM and LDROM function in the same way but have different size. Each block is accumulated page by page and the page size is 128 bytes. The Flash control unit supports Erase, Program, and Read modes. The external writer tools though specific I/O pins, In-Application-Programming (IAP), or In-System-Programming (ISP) can both perform these modes.

6.1.1 Program Memory

The Program Memory stores the program codes to execute as shown in Figure 6.1-1 ML51/ML54/ML56 Series Program Memory Map. After any reset, the CPU begins execution from location 0000H.

To service the interrupts, the interrupt service locations (called interrupt vectors) should be located in the Program Memory. Each interrupt is assigned with a fixed location in the Program Memory. The interrupt causes the CPU to jump to that location with where it commences execution of the interrupt service routine (ISR). External Interrupt 0, for example, is assigned to location 0003H. If External Interrupt 0 is going to be used, its service routine should begin at location 0003H. If the interrupt is not going to be used, its service location is available as general purpose Program Memory.

The interrupt service locations are spaced at an interval of eight bytes: 0003H for External Interrupt 0, 000BH for Timer 0, 0013H for External Interrupt 1, 001BH for Timer 1, etc. If an interrupt service routine is short enough (as is often the case in control applications), it can reside entirely within the 8-Byte interval. However longer service routines should use a JMP instruction to skip over subsequent interrupt locations if other interrupts are in use.

The ML51/ML54/ML56 Series provides two internal Program Memory blocks APROM and LDROM. Although they both behave the same as the standard 8051 Program Memory, they play different rules according to their ROM size. The APROM on ML51/ML54/ML56 Series can be up to 64 Kbytes. User Code is normally put inside. CPU fetches instructions here for execution. The MOVC instruction can also read this region.

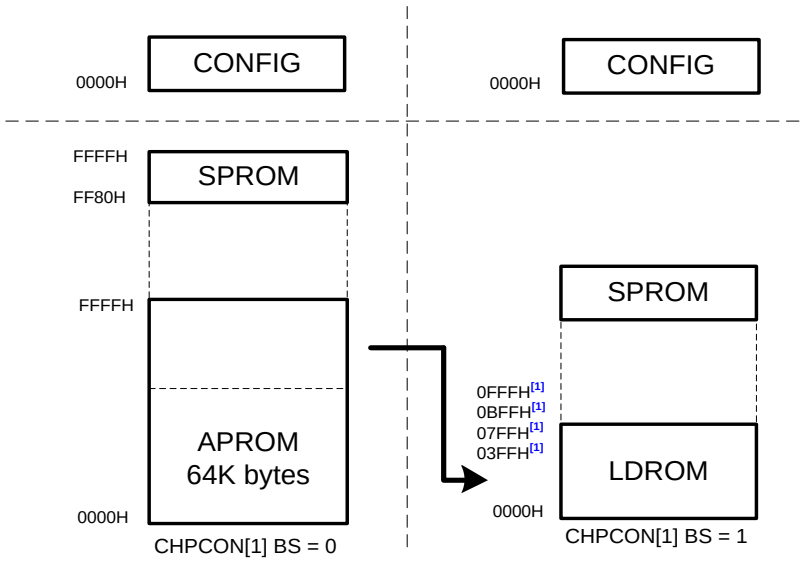
The other individual Program Memory block is called LDROM. The normal function of LDROM is to store the Boot Code for ISP. It can update APROM space and CONFIG bytes. The code in APROM can also re-program LDROM. For ISP details and configuration bit setting related with APROM and LDROM, see Section 6.4.1.5“In-System-Programming (ISP)”. Note that APROM and LDROM are hardware individual blocks, consequently if CPU re-boots from LDROM, CPU will automatically re-vector Program Counter 0000H to the LDROM start address. Therefore, CPU accounts the LDROM as an independent Program Memory and all interrupt vectors are independent from APROM.

CONFIG1

7	6	5	4	3	2	1	0
-	-	-	-	-	LDSIZE[2:0]		
-	-	-	-	-	R/W		

Factory default value: 1111 1111b

Bit	Name	Description
[2:0]	**LDSIZE[2:0]**	**LDROM Size Select** Flash size is 64KB: 111 = No LDROM. APROM is 64 Kbytes. 110 = LDROM is 1 Kbytes. APROM is 63 Kbytes. 101 = LDROM is 2 Kbytes. APROM is 62 Kbytes. 100 = LDROM is 3 Kbytes. APROM is 61 Kbytes. 0xx = LDROM is 4 Kbytes. APROM is 60 Kbytes.



[1] The logic boundary addresses of APROM and LDROM are defined by CONFIG1[2:0].

Figure 6.1-1 ML51/ML54/ML56 Series Program Memory Map

6.1.2 Security Protection Memory (SPROM)

The security protection memory (SPROM) is used to store instructions for security application. The SPROM includes 128 bytes at location address FF80H ~ FFFFH and doesn't support "whole chip erase command". Figure 6.1-2 SPROM Memory Mapping And SPROM Security Mode shows that the last byte of SPROM (address: FFFFH) is used to identify the SPROM code is non-secured or secured mode.

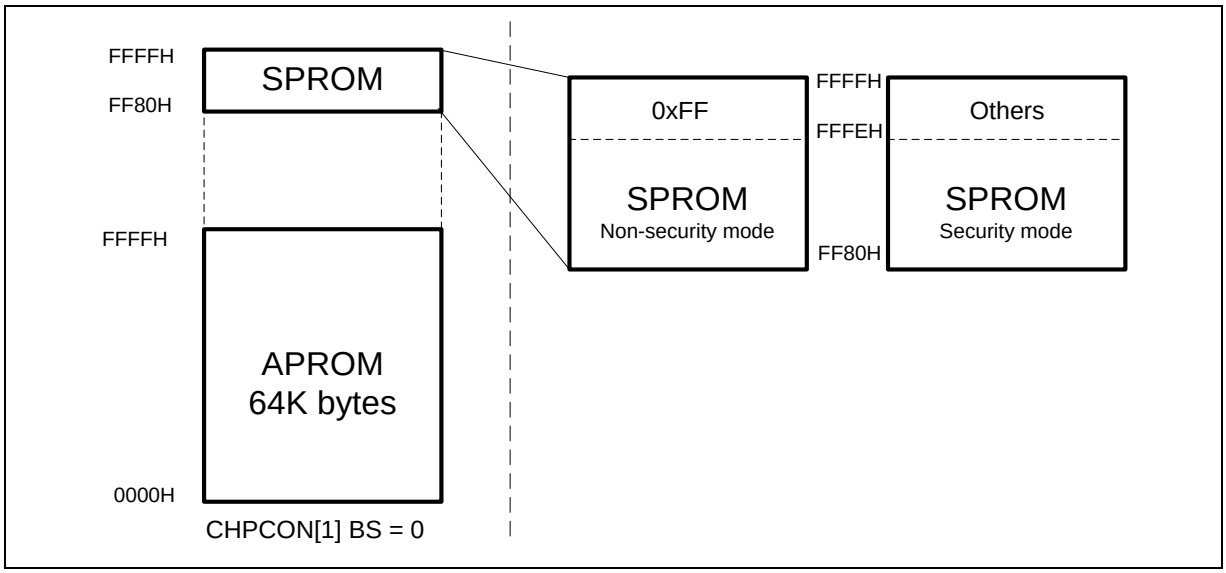


Figure 6.1-2 SPROM Memory Mapping And SPROM Security Mode

(1) SPROM non-secured mode (the last byte is 0xFF). The access behavior of SPROM is the same with APROM and LDROM. All area can be read by CPU or ISP command, and can be erased and programmed by ISP command.

(2) SPROM secured mode (the last byte is not 0xFF). In order to conceal SPROM code in secured mode, CPU only can perform instruction fetch and get data from SPROM when CPU is run at SPROM area. Otherwise, CPU will get all 00H for data access. In order to protect SPROM, the CPU instruction fetch will also get zero value when ICE (OCD) port is connected in secured code. At this mode, SPROM doesn't support ISP program, read or erase.

6.1.3 96-Bit Unique Code (UID)

Before shipping out, each ML51/ML54/ML56 Series chip was factory pre-programmed with a 96-bit width serial number, which is guaranteed to be unique for each piece of ML51/ML54/ML56 Series. The serial number is called Unique Code or UID. The user can read the Unique Code only by IAP command. More details please see Chapter 6.4.1 In-application-programming (IAP).

IAP Mode	IAPCN				IAPA[15:0] {IAPAH, IAPAL}	IAPFD[7:0]
	IAPB [1:0]	FOEN	FCEN	FCTRL [3:0]		
96-bit Unique Code read	XX	0	0	0100	0000H to 000BH	Data out

6.1.4 Data Flash

ML51/ML54/ML56 Series Data Flash is shared with APROM or LDROM. Any page of APROM or LDROM can be used as non-volatile data Flash storage and size no need special configuration. The base address of Data Flash is determined by applying IAP, For IAP details, please see Chapter 6.4.1 In-application-programming (IAP). All of embedded Flash memory is 128 bytes per page erased.

6.1.5 Data Memory

6.1.5.1 Internal data memory

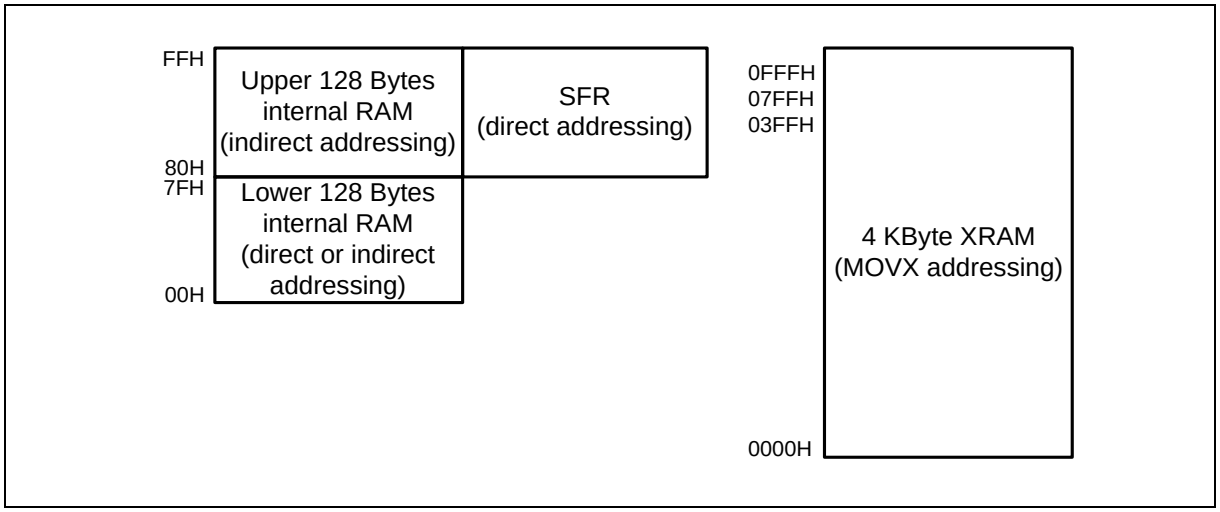


Figure 6.1-3 Data Memory Map

Figure 6.1-3 Data Memory Map shows the internal Data Memory spaces available on ML51/ML54/ML56 Series. Internal Data Memory occupies a separate address space from Program Memory. The internal Data Memory can be divided into three blocks. They are the lower 128 bytes of RAM, the upper 128 bytes of RAM, and the 128 bytes of SFR space. Internal Data Memory addresses are always 8-bit wide, which implies an address space of only 256 bytes. Direct addressing higher than 7FH will access the special function registers (SFR) space and indirect addressing higher than 7FH will access the upper 128 bytes of RAM. Although the SFR space and the upper 128 bytes of RAM share the same logic address, 80H through FFH, actually they are physically separate entities. Direct addressing to distinguish with the higher 128 bytes of RAM can only access these SFR. Sixteen addresses in SFR space are either byte-addressable or bit-addressable. The bit-addressable SFR are those whose addresses end in 0H or 8H.

The lower 128 bytes of internal RAM are present in all 80C51 devices. The lowest 32 bytes as general purpose registers are grouped into 4 banks of 8 registers. Program instructions call these registers as R0 to R7. Two bits RS0 and RS1 in the Program Status Word (PSW[3:4]) select which Register Bank is used. It benefits more efficiency of code space, since register instructions are shorter than instructions that use direct addressing. The next 16 bytes above the general purpose registers (byte-address 20H through 2FH) form a block of bit-addressable memory space (bit-address 00H through 7FH). The 80C51 instruction set includes a wide selection of single-bit instructions, and the 128 bits in this area can be directly addressed by these instructions. The bit addresses in this area are 00H through 7FH.

Either direct or indirect addressing can access the lower 128 bytes space. However, the upper 128 bytes can only be accessed by indirect addressing.

Another application implemented with the whole block of internal 256 bytes RAM is used for the stack. This area is selected by the Stack Pointer (SP), which stores the address of the top of the stack. Whenever a JMP, CALL or interrupt is invoked, the return address is placed on the stack. There is no restriction as to where the stack can begin in the RAM. By default however, the Stack Pointer contains 07H at reset. User can then change this to any value desired. The SP will point to the last used value. Therefore, the SP will be incremented and then address saved onto the stack. Conversely, while popping from the stack the contents will be read first, and then the SP is decreased.

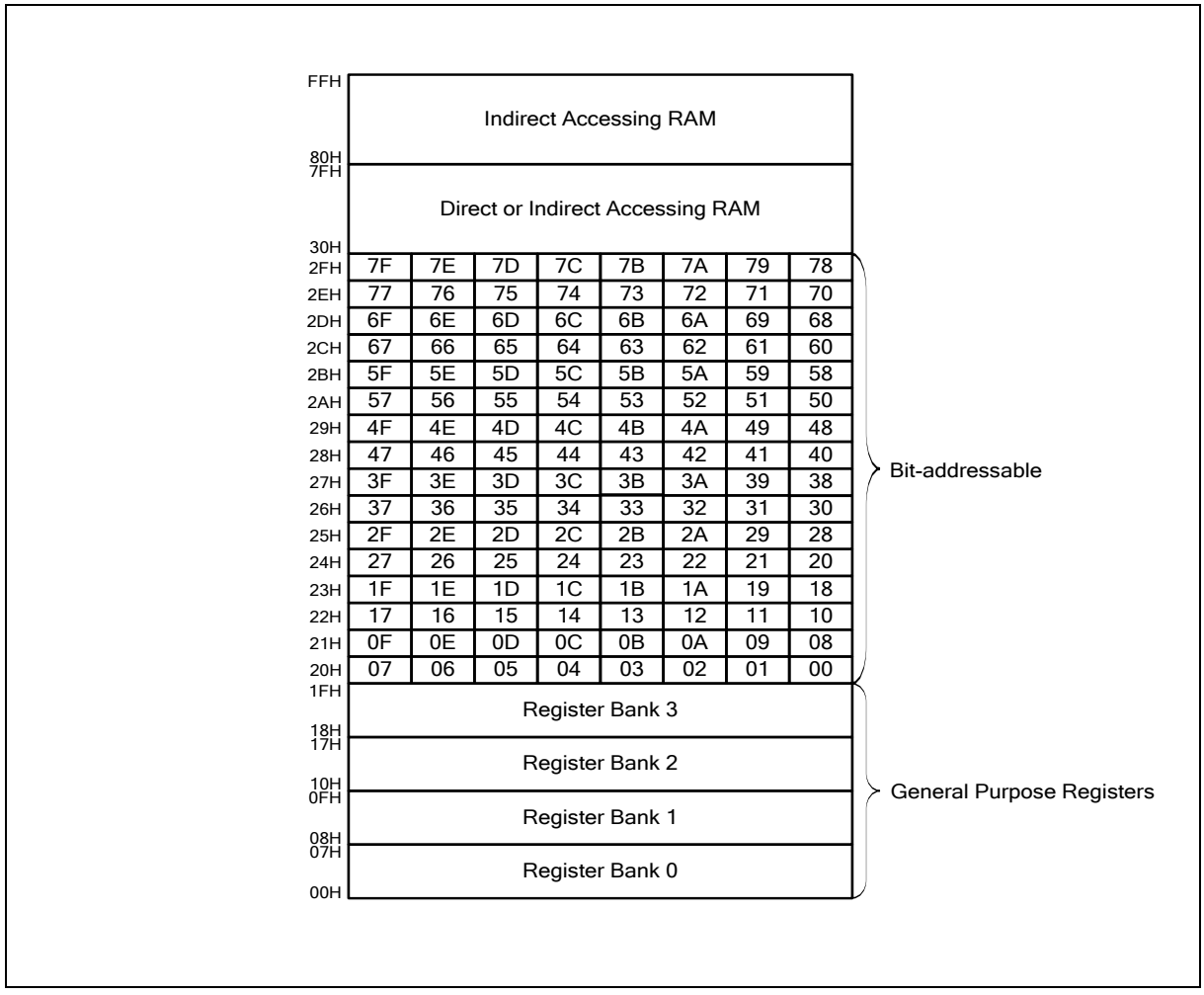


Figure 6.1-4 Internal 256 Bytes RAM Addressing

6.1.5.2 On-Chip XRAM

The ML51/ML54/ML56 Series provides additional on-chip 4 Kbytes auxiliary RAM called XRAM to enlarge the RAM space. It occupies the address space from 00H through FFH. The 4 Kbytes of XRAM are indirectly accessed by move external instruction MOVX @DPTR or MOVX @Ri. (See the demo code below.) Note that the stack pointer cannot be located in any part of XRAM.

XRAM demo code:

Assembler:

```
    MOV R0,#23H    ;write #5AH to XRAM with address @23H
    MOV A,#5AH
    MOVX   @R0,A
    MOV R1,#23H    ;read from XRAM with address @23H
    MOVX   A,@R1
    MOV DPTR,#0023H     ;write #5BH to XRAM with address @0023H
    MOV A,#5BH
    MOVX   @DPTR,A
```

```
MOV DPTR,#0023H     ;read from XRAM with address @0023H
MOVX  A,@DPTR
```

C51:

```
unsigned char temp;         //define data variable
unsigned char xdata xtemp _at_ 0x23; //define variable at xdata 0x23;
xtemp = 0x5B;             //write #5BH to XRAM with address @0023H
xtemp++;
temp = xtemp;             //read from XRAM with address @0023H
```

6.1.6 Config Bytes

The ML51/ML54/ML56 Series has several hardware configuration bytes, called CONFIG, those are used to configure the hardware options such as the security bits, system clock source, and so on. These hardware options can be re-configured through the parallel Writer, In-Circuit-Programming (ICP), or In-Application-Programming (IAP). Several functions, which are defined by certain CONFIG bits are also available to be re-configured by SFR. Therefore, there is a need to load such CONFIG bits into respective SFR bits. Such loading will occur after resets. These SFR bits can be continuously controlled via user's software.

CONFIG bits marked as "-"should always keep un-programmed.

CONFIG0

7	6	5	4	3	2	1	0
CBS	**FSYS**	**OCDPWM**	**OCDEN**	**-**	**-**	**LOCK**	**-**
R/W	R/W	R/W	R/W	-	-	R/W	-

Factory default value: 1111 1111b

Bit	Name	Description
[7]	**CBS**	**CONFIG Boot Select** This bit defines from which block that MCU re-boots after resets except software reset. 1 = MCU will re-boot from APROM after resets except software reset. 0 = MCU will re-boot from LDROM after resets except software reset.
[6]	**FSYS**	**Default Fsys Select in Power-on Sequence or ICP/HW/ICE Entry Mode.** 1 = Default Fsys select HIRC. 0 = Default Fsys select LIRC and HIRC off.
[5]	**OCDPWM**	**PWM Output State Under OCD Halt** This bit decides the output state of PWM when OCD halts CPU. 1 = Tri-state pins those are used as PWM outputs. 0 = PWM continues.
[4]	**OCDEN**	**OCD Enable** 1 = OCD Disabled. 0 = OCD Enabled. **Note:** If MCU run in OCD debug mode and OCDEN = 0, hard fault reset will disable. Only HFIF flag be asserted.
[3:2]	**-**	Reserved
[1]	**LOCK**	**Chip Lock Enable** 1 = Chip is unlocked. Flash Memory is not locked. Their contents can be read out through a parallel Writer/ICP programmer. 0 = Chip is locked. Whole Flash Memory is locked. Their contents read through a parallel Writer or ICP programmer will be all blank (FFH). Programming to Flash Memory is invalid. Note that CONFIG bytes are always unlocked and can be read. Hence, once the chip is locked, the CONFIG bytes cannot be erased or programmed individually. The only way to disable chip lock is execute “whole chip erase”. However, all data within the Flash Memory and CONFIG bits will be erased when this procedure is executed. If the chip is locked, it does not alter the IAP function.

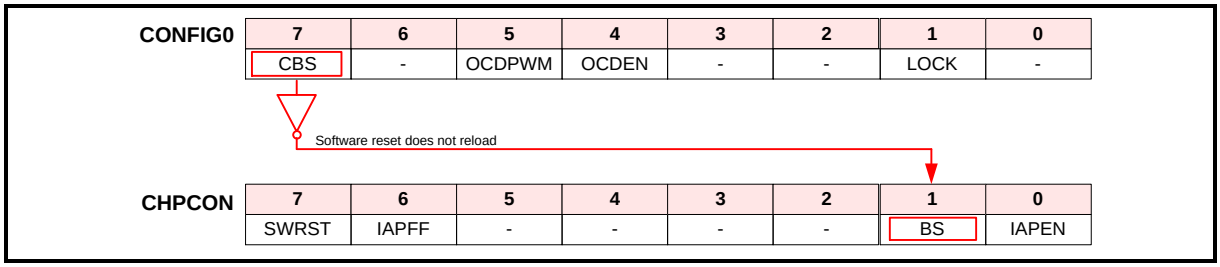


Figure 6.1-5 CONFIG0 Any Reset Reloading

CONFIG1

7	6	5	4	3	2	1	0
-	-	-	-	-	LDSIZE[2:0]		
-	-	-	-	-	R/W		

Factory default value: 1111 1111b

Bit	Name	Description
[2:0]	**LDSIZE[2:0]**	**LDROM Size Select** Flash size is 64KB: 111 = No LDROM. APROM is 64 Kbytes. 110 = LDROM is 1 Kbytes. APROM is 63 Kbytes. 101 = LDROM is 2 Kbytes. APROM is 62 Kbytes. 100 = LDROM is 3 Kbytes. APROM is 61 Kbytes. 0xx = LDROM is 4 Kbytes. APROM is 60 Kbytes.

CONFIG2

7	6	5	4	3	2	1	0
CBODEN	CBOV[2:0]			BOIAP	CBORST	-	-
R/W	R/W			R/W	R/W	-	-

Factory default value: 1111 1111b

Bit	Name	Description
[7]	**CBODEN**	**CONFIG Brown-Out Detect Enable** 1 = Brown-out detection circuit OFF. 0 = Brown-out detection circuit ON.
[6:4]	**CBOV[2:0]**	**CONFIG Brown-Out Voltage Select** 111 = V_{BOD} is 1.8V. 110 = V_{BOD} is 1.8V. 101 = V_{BOD} is 2.0V. 100 = V_{BOD} is 2.4V. 011 = V_{BOD} is 2.7V. 010 = V_{BOD} is 3.0V. 001 = V_{BOD} is 3.7V. 000 = V_{BOD} is 4.4V.
[3]	**BOIAP**	**Brown-Out Inhibiting IAP** This bit decides whether IAP erasing or programming is inhibited by brown-out status. This bit is valid only when brown-out detection is enabled. 1 = IAP erasing or programming is inhibited if V_{DD} is lower than V_{BOD}. 0 = IAP erasing or programming is allowed under any workable V_{DD}.
[2]	**CBORST**	**CONFIG Brown-Out Reset Enable** This bit decides whether a brown-out reset is caused by a power drop below V_{BOD}. 1 = Brown-out reset Enabled. 0 = Brown-out reset Disabled.

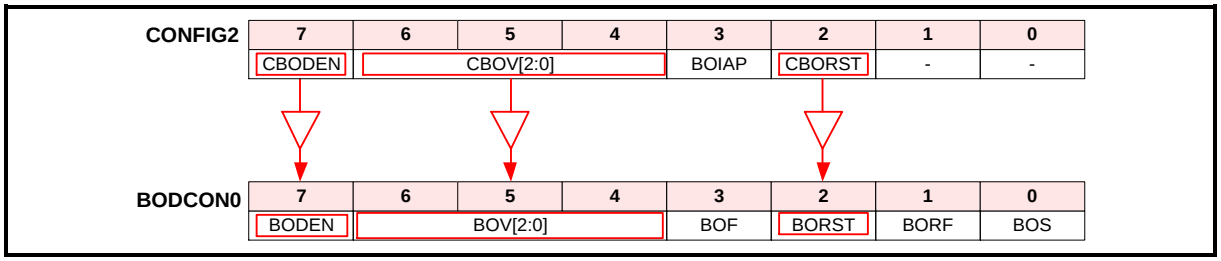


Figure 6.1-6 CONFIG2 Power-On Reset Reloading

CONFIG4

7	6	5	4	3	2	1	0
WDTEN[3:0]				-	-	-	-
R/W				-	-	-	-

Factory default value: 1111 1111b

Bit	Name	Description
[7:4]	**WDTEN[3:0]**	**WDT Enable** This field configures the WDT behavior after MCU execution. 1111 = WDT is Disabled. WDT can be used as a general purpose timer via software control. 0101 = WDT is Enabled as a time-out reset timer and it stops running during Idle or Power-down mode. Others = WDT is Enabled as a time-out reset timer and it keeps running during Idle or Power-down mode.
[3:0]	-	Reserved

6.1.7 Special Function Register (SFR)

The ML51/ML54/ML56 Series uses Special Function Registers (SFR) to control and monitor peripherals and their modes. The SFR reside in the register locations 80 to FFH and are accessed by direct addressing only. SFR those end their addresses as 0H or 8H are bit-addressable. It is very useful in cases where user would like to modify a particular bit directly without changing other bits via bit-field instructions. All other SFR are byte-addressable only. The ML51/ML54/ML56 Series contains all the SFR presenting in the standard 8051. However some additional SFR are built in. Therefore, some of unused bytes in the original 8051 have been given new functions. The SFR are listed below.

6.1.7.1 SFR Page Selection

To accommodate more than 128 SFR in the 0x80 to 0xFF address space, SFR paging has been implemented. By default, all SFR accesses target SFR Page 0. During device initialization, some SFR located on SFR Page 1 may need to be accessed. The register SFRS is used to switch SFR addressing page.

SFRS – SFR Page Selection

Register	SFR Address	Reset Value
SFRS	91H, All pages	0000_0000b

7	6	5	4	3	2	1	0
-	-	-	-	-	-	SFRPAGE[1:0]	
-	-	-	-	-	-	R/W	

Bit	Name	Description
[1:0]	**SFRPAGE[1:0]**	**SFR Page Select** 00 = Instructions access SFR Page 0. 01 = Instructions access SFR Page 1. 10 = Instructions access SFR page 2. 11 = Instructions access SFR page 3.

Switch SFR page demo code:

```
MOV SFRS,#00H          ;switch to SFR Page 0
MOV SFRS,#01H          ;switch to SFR Page 1
MOV SFRS,#02H          ;switch to SFR page 2
MOV SFRS,#03H          ;switch to SFR page 3
```

6.1.7.2 Timed Access Protection (TA)

The ML51/ML54/ML56 Series has several features such as WDT and Brown-out detection that are crucial to proper operation of the system. If leaving these Register Description unprotected, errant code may write undetermined value into them and results in incorrect operation and loss of control. To prevent this risk, the ML51/ML54/ML56 Series has a protection scheme, which limits the write access to critical SFR. This protection scheme is implemented using a timed access (TA). The following registers are related to the TA process.

TA – Timed Access

Register	SFR Address	Reset Value
TA	C7H, All pages	0000_0000 b

7	6	5	4	3	2	1	0
TA[7:0]							
W							

Bit	Name	Description
[7:0]	**TA[7:0]**	**Timed Access** The timed access register controls the access to protected SFR. To access protected bits, user should first write AAH to the TA and immediately followed by a write of 55H to TA. After these two steps, a writing permission window is opened for 4 clock cycles during this period that user may write to protected SFR.

In timed access method, the bits, which are protected, have a timed write enable window. A write is successful only if this window is active, otherwise the write will be discarded. When the software writes AAH to TA, a counter is started. This counter waits for 3 clock cycles looking for a write of 55H to TA. If the second write of 55H occurs within 3 clock cycles of the first write of AAH, then the timed access window is opened. It remains open for 4 clock cycles during which user may write to the protected bits. After 4 clock cycles, this window automatically closes. Once the window closes, the procedure should be repeated to write another protected bits. Not that the TA protected SFR are required timed access for writing but reading is not protected. User may read TA protected SFR without giving AAH and 55H to TA register. The suggestion code for opening the timed access window is shown below.

```
    (CLR EA)       ;if any interrupt is enabled, disable temporally
    MOV  TA,#0AAH
    MOV  TA,#55H
    (Instruction that writes a TA protected register)
    (SETB  EA)       ;resume interrupts enabled
```

Any enabled interrupt should be disabled during this procedure to avoid delay between these three writings. If there is no interrupt enabled, the CLR EA and SETB EA instructions can be left out.

Examples of timed assess are shown to illustrate correct or incorrect writing process.

Example 1,

```
    MOV  TA,#0AAH    ;3 clock cycles
    MOV  TA,#55H   ;3 clock cycles
    ORL  WDCON,#data    ;4 clock cycles
```

Example 2,

```
    MOV  TA,#0AAH    ;3 clock cycles
    MOV  TA,#55H   ;3 clock cycles
    NOP          ;1 clock cycle
    ANL  BODCON0,#data ;4 clock cycles
```

Example 3,

```
MOV  TA,#0AAH    ;3 clock cycles
MOV  TA,#55H   ;3 clock cycles
MOV  WDCON,#data1   ;3 clock cycles
ORL  BODCON0,#data2;4 clock cycles
```

Example 4,

```
MOV  TA,#0AAH    ;3 clock cycles
NOP         ;1 clock cycle
MOV  TA,#55H   ;3 clock cycles
ANL  BODCON0,#data ;4 clock cycles
```

In the first example, the writing to the protected bits is done before the 3-clock-cycle window closes. In example 2, however, the writing to BODCON0 does not complete during the window opening, there will be no change of the value of BODCON0. In example 3, the WDCON is successful written but the BODCON0 write is out of the 3-clock-cycle window. Therefore, the BODCON0 value will not change either. In Example 4, the second write 55H to TA completes after 3 clock cycles of the first write TA of AAH, and thus the timed access window is not opened at all, and the write to the protected byte affects nothing.

6.1.7.3 Dual DPTRs

The original 8051 contains one DPTR (data pointer) only. With single DPTR, it is difficult to move data form one address to another with wasting code size and low performance. The ML51/ML54/ML56 Series provides two data pointers. Thus, software can load both a source and a destination address when doing a block move. Once loading, the software simply switches between DPTR and DPTR1 by the active data pointer selection DPS (AUXR0.0) bit.

An example of 64 bytes block move with dual DPTRs is illustrated below. By giving source and destination addresses in data pointers and activating cyclic makes block RAM data move more simple and efficient than only one DPTR. The INC AUXR0 instruction is the shortest (2 bytes) instruction to accomplish DPTR toggling rather than ORL or ANL. For AUXR0.1 contains a hard-wired 0, it allows toggling of the DPS bit by incrementing AUXR0 without interfering with other bits in the register.

```
MOV   R0,#64        ;number of bytes to move
MOV   DPTR,#D_Addr    ;load destination address
INC   AUXR0      ;change active DPTR
MOV   DPTR,#S_Addr    ;load source address
LOOP:
MOVX  A,@DPTR    ;read source data byte
INC   AUXR0      ;change DPTR to destination
MOVX  @DPTR,A    ;write data to destination
INC   DPTR       ;next destination address
INC   AUXR0      ;change DPTR to source
INC   DPTR       ;next source address
DJNZ  R0,LOOP
INC   AUXR0      ;(optional) restore DPS
```

AUXR0 also contains a general purpose flag GF2 in its bit 3 that can be set or cleared by the user via software.

DPL – Data Pointer Low Byte

Register	SFR Address	Reset Value
DPL	82H, All pages	0000_0000b

7	6	5	4	3	2	1	0
DPTR[7:0]							
R/W							

Bit	Name	Description
[7:0]	DPTR[7:0]	**Data Pointer Low Byte** This is the low byte of 16-bit data pointer. DPL combined with DPH serve as a 16-bit data pointer DPTR to access indirect addressed RAM or Program Memory. DPS (AUXR0.0) bit decides which data pointer, DPTR or DPTR1, is activated.

DPH – Data Pointer High Byte

Register	SFR Address	Reset Value
DPH	83H, All pages	0000_0000b

7	6	5	4	3	2	1	0
DPTR[15:8]							
R/W							

Bit	Name	Description
[7:0]	**DPTR[15:8]**	**Data Pointer High Byte** This is the high byte of 16-bit data pointer. DPH combined with DPL serve as a 16-bit data pointer DPTR to access indirect addressed RAM or Program Memory. DPS (AUXR0.0) bit decides which data pointer, DPTR or DPTR1, is activated.

AUXR0 – Auxiliary Register 0

Register	SFR Address	Reset Value
AUXR0	A2H, Page 0	POR: 0000 0000b, Software reset: 1U00 0000b, nRESET pin: U100 0000b, Hard fault: UU10 0000b Others: UUU0 0000b

7	6	5	4	3	2	1	0
SWRF	**RSTPINF**	**HardF**	**HardFInt**	**GF2**	**-**	**0**	**DPS**
R/W	R/W	R/W	R/W	R/W	-	R	R/W

Bit	Name	Description
[3]	**GF2**	**General Purpose Flag 2** The general purpose flag that can be set or cleared by the user via software.
[2]	**-**	Reserved
[1]	**0**	Reserved This bit is always read as 0.
[0]	**DPS**	**Data Pointer Select** 0 = Data pointer 0 (DPTR) is active by default. 1 = Data pointer 1 (DPTR1) is active. After DPS switches the activated data pointer, the previous inactivated data pointer remains its original value unchanged.

6.1.7.4 SFR Memory Map

ML51 32KB / 16KB Flash Series

	Addr	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
0	F8	S1CON	SPI1CR0	SPI1CR1	SPI1SR	SPI1DR	DMA1BAH	EIP1	EIPH1
1			PWM0DTEN	PWM0DTCNT	PWM0MEN	PWM0MD	LVRFLTEN	-	LVRDIS
2			P0MF10	P0MF32	P0MF54	P0MF76	P1MF10	P1MF32	P1MF54
0	F0	B	DMA1TSR	MTM1DA	SPI0CR0	SPI0SR	SPI0DR	DMA0BAH	EIPH0
1			-	-	SPI0CR1	-	-	-	-
2			P1MF76	P2MF10	P2MF32	P2MF54	P2MF76	P3MF10	P3MF32
0	E8	I2C1CON	DMA0TSR	MTM0DA	DMA1CR	DMA1MA	DMA1CNT	DMA1CCNT	EIP0
1			PICON	PINEN	PIPEN	-	C2L	C2H	-
2			P3MF54	P3MF76	P4MF10	P4MF32	P4MF54	P4MF76	P5MF10
0	E0	ACC	ADCCON1	ADCCON2	ADCDLY	ADCBAH	ADCSN	ADCCN	ADCSR
1			CAPCON0	CAPCON1	CAPCON2	C0L	C0H	C1L	C1H
2			P5MF32	P5MF54	P5MF76	-	-	-	-
0	D8	P4	SC0DR	SC0EGT	SC0ETURD0	SC0ETURD1	SC0IE	SC0IS	SC0TSR
1			PWM0PL	PWM0C0L	PWM0C1L	PWM0C2L	PWM0C3L	-	PWM0CON1
2			-	-	-	-	-	-	-
0	D0	PSW	PWM0CON0	ACMPCR0	ACMPCR1	ACMPSR	ACMPVREF	SC0CR0	SC0CR1
1			PWM0PH	PWM0C0H	PWM0C1H	PWM0C2H	PWM0C3H	PWM0NP	PWM0FBD
2			-	-	-	-	-	-	-
0	C8	T2CON	T2MOD	PIF	ADCBAL	TL2	TH2	ADCMPL	ADCMPH
1			AUXR1	RCMP2L	RCMP2H	PWM0C4L	PWM0C5L	AINDIDS	
2			-	-	-	-	-	-	-
0	C0	I2C0CON	I2C0ADDR	ADCRL	ADCRH	T3CON	RL3	RH3	TA
1			CKDIV	P3M1	P3M2	PWM0C4H	PWM0C5H	PORDIS	
2			-	-	-	-	-	-	
0	B8	IP	SADEN	SADEN1	SADDR1	I2C0DAT	I2C0STAT	I2C0CLK	I2C0TOC
1			P4M1	P4M2	P4S	P4SR	P5M1	P5M2	P5S
2			-	-	-	-	-	-	-
0	B0	P3	P5	I2C1ADDR0	I2C1DAT	I2STAT1	I2C1CLK	I2C1TOC	IPH
1			P0M1	P0M2	P1M1	P1M2	P2M1	P2M2	PWM0INTC
2			-	-	-	-	-	-	
0	A8	IE	SADDR	WDCON	BODCON1	EIP2	EIPH2	IAPFD	IAPCN
1			VRFCON	VRFTRIM	ACMPCR2	P3S	P3SR	P5SR	PIPS7
2									
0	A0	P2	ADCCON0	AUXR0	BODCON0	IAPTRG	IAPUEN	IAPAL	IAPAH
1			PIPS0	PIPS1	PIPS2	PIPS3	PIPS4	PIPS5	PIPS6
2			I2C0ADDR1	I2C0ADDR2	I2C0ADDR3	I2C1ADDR1	I2C1ADDR2	I2C1ADDR3	-
0	98	SCON	SBUF	SBUF1	EIE0	EIE1	RSR	P2SR	CHPCON
1			P0S	P0SR	P1S	P1SR	P2S	-	-
2			-	-	-	-	-	-	-
0	90	P1	SFRS	DMA0CR	DMA0MA	DMA0CNT	DMA0CCNT	CKSWT	CKEN
1				P0UP	P1UP	P2UP	P3UP	P4UP	P5UP
2				-	-	-	-	-	-
0	88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	WKCON
1				P0DW	P1DW	P2DW	P3DW	P4DW	P5DW
2				-	-	-	-	-	-
0	80	P0	SP	DPL	DPH	RCTRIM0	RCTRIM1	RWK	PCON
1						LRCTRIM	-	CWK	
2						-	-	-	

Note:
[1]Unoccupied addresses in the SFR space marked in "-" are reserved for future use. Accessing these areas will have an indeterminate effect and should be avoided.

Table 6.1-1 ML51 32KB / 16KB Flash Series Special Function Register (SFR) Memory Map

ML56 / ML54 / ML51 64KB Flash Series

Page	Addr	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
0	F8	S1CON	SPI1CR0	SPI1CR1	SPI1SR	SPI1DR	DMA1BAH	EIP1	EIPH1
1			PWM0DTEN	PWM0DTCNT	PWM0MEN	PWM0MD	LVRFLTEN	-	LVRDIS
2			P0MF10	P0MF32	P0MF54	P0MF76	P1MF10	P1MF32	P1MF54
3			LCDCON	LCDCLK	LCDPTR	LCDDAT	LCDPWR	LCDBL	LCDMODE
0	F0	B	DMA1TSR	MTM1DA	SPI0CR0	SPI0SR	SPI0DR	DMA0BAH	EIPH0
1			-	-	SPI0CR1	-	-	-	-
2			P1MF76	P2MF10	P2MF32	P2MF54	P2MF76	P3MF10	P3MF32
3			LCDCPUMP	-	-	LCDCON1	LCDCPALCT0	LCDCPCT0	LCDIF
0	E8	I2C1CON	DMA0TSR	MTM0DA	DMA1CR0	DMA1MAL	DMA1CNT	DMA1CCNT	EIP0
1			PICON	PINEN	PIPEN	-	C2L	C2H	LDOTRIM
2			P3MF54	P3MF76	P4MF10	P4MF32	P4MF54	P4MF76	P5MF10
3			LCDCPALCT1	LCDCPCT1	-	-	-	-	-
0	E0	ACC	ADCCON1	ADCCON2	ADCDLY	ADCBAH	ADCSN	ADCCN	ADCSR
1			CAPCON0	CAPCON1	CAPCON2	C0L	C0H	C1L	C1H
2			P5MF32	P5MF54	P5MF76	BRCTRIM	ADCCAL	SC1CR0	SC1CR1
3			-	-	-	-	-	-	-
0	D8	P4	SC0DR	SC0EGT	SC0ETURD0	SC0ETURD1	SC0IE	SC0IS	SC0TSR
1			PWM0PL	PWM0C0L	PWM0C1L	PWM0C2L	PWM0C3L	-	PWM0CON1
2			SC1DR	SC1EGT	SC1ETURD0	SC1ETURD1	SC1IE	SC1IS	SC1TSR
3			-	-	-	-	-	-	-
0	D0	PSW	PWM0CON0	ACMPCR0	ACMPCR1	ACMPSR	ACMPVREF	SC0CR0	SC0CR1
1			PWM0PH	PWM0C0H	PWM0C1H	PWM0C2H	PWM0C3H	PWM0NP	PWM0FBD
2			PWM3PL	PWM3C0L	PWM3C1L	PWM3CON0	PWM3CON1	PWM3INTC	I2C1ADDRM
3			-	-	-	-	-	-	-
0	C8	T2CON	T2MOD	PIF	ADCBAL	TL2	TH2	ADCMPL	ADCMPH
1			AUXR1	RCMP2L	RCMP2H	PWM0C4L	PWM0C5L	AINDIDS0	-
2			PWM3PH	PWM3C0H	PWM3C1H	PWM3MD	PWM3MEN	AINDIDS1	I2C0ADDRM
3			-	-	-	-	-	PWM0FBS	AUXR3
0	C0	I2C0CON	I2C0ADDR0	ADCRL	ADCRH	T3CON	RL3	RH3	TA
1			CKDIV	P3M1	P3M2	PWM0C4H	PWM0C5H	PORDIS	
2			PWM2PL	PWM2C0L	PWM2C1L	PWM2CON0	PWM2CON1	PWM2INTC	
3			-	-	-	-	-	-	
0	B8	IP	SADEN0	SADEN1	SADDR1	I2C0DAT	I2C0STAT	I2C0CLK	I2C0TOC
1			P4M1	P4M2	P4S	P4SR	P5M1	P5M2	P5S
2			PWM2PH	PWM2C0H	PWM2C1H	PWM2MD	PWM2MEN	CWKH	RWKH [1]
3			RTCCLKFMT	-	RTCWEEKDAY	RTCLEAPYEAR	RTCTICK	RTCTAMSK	RTCCAMSK
0	B0	P3	P5	I2C1ADDR0	I2C1DAT	I2C1STAT	I2C1CLK	I2C1TOC	IPH
1			P0M1	P0M2	P1M1	P1M2	P2M1	P2M2	PWM0INTC
2			DMA2TSR	DMA2BAH	DMA2CR0	DMA2MAL	DMA2CNT	DMA2CCNT	MTM2DA
3			RTCTALMSEC	RTCTALMMIN	RTCTALMHR	RTCTALMHZ	RTCCALMDAY	RTCCALMMON	RTCCALMYEAR
0	A8	IE	SADDR0	WDCON	BODCON1	EIP2	EIPH2	IAPFD	IAPCN
1			VRFCON	VRFTRIM	ACMPCR2	P3S	P3SR	P5SR	PIPS7
2			DMA3TSR	DMA3BAH	DMA3CR0	DMA3MAL	DMA3CNT	DMA3CCNT	MTM3DA
3			RTCTIMESEC	RTCTIMEMIN	RTCTIMEHR	-	RTCCALDAY	RTCCALMON	RTCCALYEAR
0	A0	P2	ADCCON0	AUXR0	BODCON0	IAPTRG	IAPUEN	IAPAL	IAPAH
1			PIPS0	PIPS1	PIPS2	PIPS3	PIPS4	PIPS5	PIPS6
2			I2C0ADDR1	I2C0ADDR2	I2C0ADDR3	I2C1ADDR1	I2C1ADDR2	I2C1ADDR3	P6
3			RTCINIT	RTCRWEN	RTCCLKSEL	RTCFREQADJ0	RTCFREQADJ1	RTCINTEN	RTCINTSTS
0	98	SCON	SBUF	SBUF1	EIE0	EIE1	RSR	-	CHPCON
1			P0S	P0SR	P1S	P1SR	P2S	P2SR	-
2			PWM1PL	PWM1C0L	PWM1C1L	PWM1CON0	PWM1CON1	PWM1INTC	-
3			WDCONH	DMA0SEED	DMA1SEED	DMA2SEED	DMA3SEED	-	-
0	90	P1	SFRS	DMA0CR0	DMA0MAL	DMA0CNT	DMA0CCNT	CKSWT	CKEN
1				P0UP	P1UP	P2UP	P3UP	P4UP	P5UP
2				P6MF10	P6MF32	P6MF54	P6MF76	P6S	P6UP
3				DMA0CRC	DMA1CRC	DMA2CRC	DMA3CRC	-	-
0	88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	WKCON
1				P0DW	P1DW	P2DW	P3DW	P4DW	P5DW
2				PWM1C0H	PWM1C1H	PWM1MD	PWM1MEN	P6SR	P6DW
3				DMA0CR1	DMA1CR1	DMA2CR1	DMA3CR1	-	-
0	80	P0	SP	DPL	DPH	RCTRIM0	RCTRIM1	RWKL	PCON

Page	Addr	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
1						LIRCTRIM	XLTCON	CWKL	
2						P6M1	P6M2	PWM1PH	
3						-	-	-	

Note:
[1] RWKH register is only for ML56/ML54/M51 64KB flash series.
[2]Unoccupied addresses in the SFR space marked in "-" are reserved for future use. Accessing these areas will have an indeterminate effect and should be avoided.

Table 6.1-2 ML56 / ML54 / ML51 64KB Flash Series Special Function Register (SFR) Memory Map

6.2 SFRs Memory Mapping

6.2.1.1 SFR Definitions And Reset Values

Bits marked in "-" are reserved for future use. They must be kept in their own initial states. Accessing these bits may cause an unpredictable effect.

Register	Definition	Addr.	Page	MSB							LSB	Reset Value	TA
				7	6	5	4	3	2	1	0		
EIPH1	Extensive Interrupt Priority High 1	FFH	0	PSPI1H	PDMA1H	PDMA0H	PSMCH	PHFH	PWKTH	PT3H	PSH_1	00000000b	
LVRDIS	Lvr Disable	FFH	1	LVRDIS[7:0]								00000000b	Y
P1MF54	P1.5 And P1.4 Multi Function Select	FFH	2	P1MF5[3:0]				P1MF4[3:0]				00000000b	
LCDMODE	LCD Control 1	FFH	3	R_MODE	BUFEN					VLCD_MODE_1	VLCD_MODE_0	00000000b	
EIP1	Extensive Interrupt Priority 1	FEH	0	PSPI1	PDMA1	PDMA0	PSMC	PHF	PWKT	PT3	PS_1	00000000b	
-	-	FEH	1										
P1MF32	P1.3 And P1.2 Multi Function Select	FEH	2	P1MF3[3:0]				P1MF2[3:0] P3MF54				00000000b	
LCDBL	LCD Blink	FEH	3		-	-	-	BLINK	BLF_2	BLF_1	BLF_0	00000000b	
DMA1BAH	PDMA11 Base Address High Byte	FDH	0	MTMDA[7:4]				XRAMA[7:4]				00000000b	
LVRFLTEN	LVR Filter Enable	FDH	1	LVRFLTEN[7:0]								00000000b	Y
P1MF10	P1.1 And P1.0 Multi Function Select	FDH	2	P1MF1[3:0]				P1MF0[3:0]				00000000b	
LCDPWR	LCD Pwr	FDH	3							PWR_NOSAVING_1	PWR_NOSAVING_0	00000000b	
SPI1DR	Serial Pereral Data Register	FCH	0	SPI1DR[7:0]								00000000b	
PWM0MD	Pwm Mask Data	FCH	1			PMD5	PMD4	PMD3	PMD2	PMD1	PMD0	00000000b	
P0MF76	P0.7 And P0.6 Multi Function Select	FCH	2	P0MF7[3:0]				P0MF6[3:0]				00000000b	
LCDDAT	LCD Data	FCH	3	LCDDAT[7:0]								00000000b	
SPI1SR	Serial Peripheral Status Register	FBH	0	SPIF	WCOL	SPIOVF	MODF	DISMODF	DISSPIF	TXBFF		00000000b	
PWM0MEN	Pwm Mask Enable	FBH	1			PMEN5	PMEN4	PMEN3	PMEN2	PMEN1	PMEN0	00000000b	
P0MF54	P0.5 And P0.4 Multifunction Select	FBH	2	P0MF5[3:0]				P0MF4[3:0]				00000000b	
LCDPTR	LCD Data Pointer	FBH	3				LCDPTR_4	LCDPTR_3	LCDPTR_2	LCDPTR_1	LCDPTR_0	00000000b	
SPI1CR1	Serial Peripheral Control Register 1	FAH	0	-		SPR3	SPR2	TXDMAEN	RXDMAEN	SPIS1	SPIS0	00000000b	
PWM0DTCNT	Pwm Deadtime Counter	FAH	1	PWM0DTCNT[7:0]								00000000b	Y
P0MF32	P0.3 And P0.2 Multi Function Select	FAH	2	P0MF3[3:0]				P0MF2[3:0]				00000000b	
LCDCLK	LCDclockcontrol	FAH	3				LCDCKS	DISP	LCDDIV_2	LCDDIV_1	LCDDIV_0	00000000b	
SPI1CR0	Serial Peripheral Control Register 0	F9H	0	SSOE	SPIEN	LSBFE	MSTR	CPOL	CPHA	SPR1	SPR0	00000000b	
PWM0DTEN	Pwm Deadtime Enable	F9H	1				PWMnDTCNT.8		PDT45EN	PDT23EN	PDT01EN	00000000b	Y
P0MF10	P0.1 And P0.0 Multi Function Select	F9H	2	P0MF1[3:0]				P0MF0[3:0]				00000000b	
LCDCON	LCD Control 0	F9H	3	LCDEN	TYPE	BIAS_1	BIAS_0	DUTY_1	DUTY_0		LCD_IE	00000000b	
S1CON	Serial Port1 Control	F8H	A	SM0_1/FE_1	SM1_1	SM2_1	REN_1	TB8_1	RB8_1	TI_1	RI_1	00000000b	
EIPH0	Extensive Interrupt Priority High	F7H	0	PT2H	PSPIH	PFBH	PWDTH	PPWMH	PCAPH	PPIH	PI2CH	00000000b	
		F7H	1										
P3MF32	P3.3 And P3.2 Multi Function Select	F7H	2	P3MF3[3:0]				P3MF2[3:0]				00000000b	
LCDIF	LCD Interrupt Flag	F7H	3						LCDCPOVIF	LCDCPIF	LCDCPALIF	00000000b	
DMA0BAH	PDMA11 Base Address High Byte	F6H	0	MTMDA[7:4]				XRAMA[7:4]				00000000b	
-	-	F6H	1	-	-	-	-	-	-	-	-	-	-

Register	Definition	Addr.	Page	MSB							LSB	Reset Value	TA
				7	6	5	4	3	2	1	0		
P3MF10	P3.1 And P3.0 Multi Function Select	F6H	2	P3MF1[3:0]				P3MF0[3:0]				00000000b	
LCDCPCT0	LCD Charge Pump Counter Value	F6H	3	LCDCPCT[7:0]								00000000b	
SPI0DR	Spi0 Data	F5H	0	SPDR[7:0]								00000000b	
-	-	F5H	1	-								-	-
P2MF76	P2.7 And P2.6 Multi Function Select	F5H	2	P2MF7[3:0]				P2MF6[3:0]				00000000b	
LCDCPALCT0	LCD Pump Counter Alarm Setting Value Low Byte	F5H	3	LCDCPOVCT[7:0]								00000000b	Y
SPI0SR	Spi0 Status	F4H	0	SPIF	WCOL	SPIOVF	MODF	DISMODF	DISSPIF	TXBFF		00000000b	
-	-	F4H	1	-	-	-	-	-	-	-	-	-	-
P2MF54	P2.5 And P2.4 Multi Function Select	F4H	2	P2MF5[3:0]				P2MF4[3:0]				00000000b	
LCDCON1	LCD control register 1	F4H	3					LCDIS	LCDIE	RE_MODE		-	-
SPI0CR0	Spi0 Control 0	F3H	0	SSOE	SPIEN	LSBFE	MSTR	CPOL	CPHA	SPR1	SPR0	00000000b	
SPI0CR1	Spi0 Control 1	F3H	1			SPR3	SPR2	TXDMAEN	RXDMAEN	SPIS1	SPIS0	00000000b	
P2MF32	P2.3 And P2.2 Multifunction Select	F3H	2	P2MF3[3:0]				P2MF2[3:0]				00000000b	
-	-	F3H	3	-		-		-			-	-	-
MTM1DA	Memory To Memory Destination Address Low Byte	F2H	0	MDAL[7:0]								00000000b	
		F2H	1										
P2MF10	P2.1 And P2.0 Multi Function Select	F2H	2	P2MF1[3:0]				P2MF0[3:0]				00000000b	
-	-	F2H	3	-	-	-	-	-	-	-	-	-	-
DMA1TSR	PDMA11 Transfer Status Register	F1H	0						ACT	HDONE	FDONE	00000000b	
CAPCON3		F1H	1										
P1MF76	P1.7 And P1.6 Multifunction Select	F1H	2	P1MF7[3:0]				P1MF6[3:0]				00000000b	
LCDCPUMP	LCD Charge Pump Voltage Set	F1H	3			VCP_SET						00000000b	
B	B Register	F0H		B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	00000000b	
EIP0	Extensive Interrupt Priority	EFH	0	PT2	PSPI	PFB	PWDT	PPWM	PCAP	PPI	PI^{2}C	00000000b	
-		-	-										
P5MF10	P5.0 And P5.0 Multifunction Select	EFH	2	P5MF1[3:0]				P5MF0[3:0]				00000000b	
		EFH	3										
DMA1CCNT	PDMA11 Current Transfer Count	EEH	0	CCNT[7:0]								00000000b	
C2H	Input Capture2 High Byte	EEH	1	C2H[7:0]								00000000b	
P4MF76	P4.7 And P4.6 Multi Function Select	EEH	2	P4MF7[3:0]				P4MF6[3:0]				00000000b	
		EEH	3										
DMA1CNT	PDMA11 Transfer Count	EDH	0	DMA1CNT[7:0]								00000000b	
C2L	Input Capture 2 Low Byte	EDH	1	C2L[7:0]								00000000b	
P4MF54	P4.5 And P4.4 Multi Function Select	EDH	2	P4MF5[3:0]				P4MF4[3:0]				00000000b	
		EDH	3										
DMA1MAL	PDMA11 Xram Base Address Low Byte	ECH	0	DMA1MA[7:0]								00000000b	
-	-	ECH	1	-	-	-		-				-	-
P4MF32	P4.3 And P4.2 Multi Function Select	ECH	2	P4MF3[3:0]				P4MF2[3:0]				00000000b	
		ECH	3										

Register	Definition	Addr.	Page	MSB							LSB	Reset Value	TA
				7	6	5	4	3	2	1	0		
DMA1CR0	PDMA11 Control Register	EBH	0	PSSEL_3	PSSEL_2	PSSEL_1	PSSEL_0	HIE	FIE	RUN	EN	00000000b	
PIPEN	Pin Interrupt High Level/Rising Edge Enable	EBH	1	PIPEN7	PIPEN6	PIPEN5	PIPEN4	PIPEN3	PIPEN2	PIPEN1	PIPEN0	00000000b	
P4MF10	P4.1 And P4.0 Multi Function Select	EBH	2	P4MF1[3:0]				P4MF0[3:0]				00000000b	
		EBH	3										
MTM0DA	Memory To Memory Destination Address Low Byte	EAH	0	MDAL[7:0]								00000000b	
PINEN	Pin Interrupt Low Level/Falling Edge Enable	EAH	1	PINEN7	PINEN6	PINEN5	PINEN4	PINEN3	PINEN2	PINEN1	PINEN0	00000000b	
P3MF76	P3.7 And P3.6 Multi Function Select	EAH	2	P3MF7[3:0]				P3MF6[3:0]				00000000b	
LCDCPCT1	LCD Charge Pump Counter Value High Byte	EAH	3							LCDCPCT_9	LCDCPCT_8	00000000b	
DMA0TSR	PDMA10 Transfer Status Register	E9H	0						ACT	HDONE	FDONE	00000000b	
PICON	Pin Interrupt Control	E9H	1	PIT7	PIT6	PIT5	PIT4	PIT3	PIT2	PIT1	PIT0	00000000b	
P3MF54	P3.5 And P3.4 Multi Function Select	E9H	2	P3MF5[3:0]				P3MF4[3:0]				00000000b	
LCDCPALCT1	LCD Pump Counter Alarm Value High Byte	E9H	3							LCDCPOVCT_9	LCDCPOVCT_8	00000000b	Y
I2C1CON	I2c1 Control Register	E8H		I	I2CEN	STA	STO	SI	AA			00000000b	
ADCSR	Adc Status Register	E7H	0	SLOW	ADCDIV_2	ADCDIV_1	ADCDIV_0		CMPHIT	HDONE	FDONE	00000000b	
C1H	Input Capture1 High Byte	E7H	1	C1H[7:0]								00000000b	
SC1CR1	Sc1 Control Register 1	E7H	2	OPE	PBOFF	WLS1	WLS0	TXDMAEN	RXDMAEN	CLKKEEP	UARTEN	00000000b	
		E7H	3										
ADCCN	Adc Current Sampling Number	E6H	0	ADCCN[7:0]								00000000b	
C1L	Input Capture1 Low Byte	E6H	1	C1L[7:0]								00000000b	
SC1CR0	Sc1 Control Register 1	E6H	2	NSB	T	RXBGTEN	CONSEL	AUTOCEN	TXOFF	RXOFF	SCEN	00000000b	
		E6H	3										
ADCSN	Adc Sampling Number	E5H	0	ADCSN[7:0]								00000000b	
C0H	Input Capture 0 High Byte	E5H	1	C0H[7:0]								00000000b	
ADCCAL	Adc Offset Calibration Control Register	E5H	2	CALI_EN	SIGN_2	SIGN_1	SIGN_0	ADCCALI_3	ADCCALI_2	ADCCALI_1	ADCCALI_0	00000000b	
		E5H	3										
ADCBAH	Adc Ram Base Address High Byte	E4H	0					ADCBAH_3	ADCBAH_2	ADCBAH_1	ADCBAH_0	00000000b	
C0L	Input Capture 0 Low Byte	E4H	1	C0L[7:0]								00000000b	
BRCTRIM	Birc Trim Control	E4H	2	IBOOST_2	IBOOST_1	IBOOST_0	BRCTRIM_4	BRCTRIM_3	BRCTRIM_2	BRCTRIM_1	BRCTRIM_0	00000000b	
		E4H	3										
ADCDLY	Adc Trigger Delay	E3H	0	ADCDLY[7:0]								00000000b	
CAPCON2	Input Capture Control2	E3H	1		ENF2	ENF1	ENF0					00000000b	
P5MF76	P5.7 And P5.6 Multi Function Select	E3H	2	P5MF7[3:0]				P5MF6[3:0]				00000000b	
		E3H	3										
ADCCON2	Adc Control 2	E2H	0	ADFBEN	ADCMPOP	ADCMPEN	ADCMPO	ADCAQT_2	ADCAQT_1	ADCAQT_0	ADCDLY.8	00000000b	
CAPCON1	Input Capture Control 1	E2H	1			CAP2LS_1	CAP2LS_0	CAP1LS_1	CAP1LS_0	CAP0LS_1	CAP0LS_0	00000000b	
P5MF54	P5.5 And P5.4 Multi Function Select	E2H	2	P5MF5[3:0]				P5MF4[3:0]				00000000b	
		E2H	3										

Register	Definition	Addr.	Page	MSB							LSB	Reset Value	TA
				7	6	5	4	3	2	1	0		
ADCCON1	Adc Control 1	E1H	0			HIE	CONT	ETGTYP_1	ETGTYP_0	ADCEX	ADCEN	00000000b	
CAPCON0	Input Capture Control 0	E1H	1		CAPEN2	CAPEN1	CAPEN0		CAPF2	CAPF1	CAPF0	00000000b	
P5MF32	P5.3 And P5.2 Multi Function Select	E1H	2	P5MF3[3:0]				P5MF2[3:0]				00000000b	
		E1H	3										
ACC	Accumulator	E0H	A	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	00000000b	
SC0TSR	Sc0 Transfer Status Register	DFH	0	ACT	BEF	FEF	PEF	TXEMPTY	TXOV	RXEMPTY	RXOV	00001010b	
PWM0CON1	Pwm Control 1	DFH	1	PWMMOD_1	PWMMOD_0	GP	PWMTYP	FBINEN	PWMDIV_2	PWMDIV_1	PWMDIV_0	00000000b	
SC1TSR	Sc1 Transfer Status Register	DFH	2	ACT	BEF	FEF	PEF	TXEMPTY	TXOV	RXEMPTY	RXOV	00000000b	
		DFH	3										
SC0IS	Sc0 Interrupt Status Register	DEH	0	-	-	Tx_Er	ACERRIF	BGTIF	TERRIF	TBEIF	RDAIF	00000010b	
-	-	DEH	1			-	-	-	-	-	-	-	
SC1IS	Sc1 Interrupt Status Register	DEH	2	-	-	Tx_Er	ACERRIF1	BGTIF1	TERRIF1	TBEIF1	RDAIF1	00000000b	
		DEH	3										
SC0IE	Sc0 Interrupt Enable Control Register	DDH	0				ACERRIEN	BGTIEN	TERRIEN	TBEIEN	RDAIEN	00000000b	
PWM0C3L	Pwm0 Channel 3 Duty Low Byte	DDH	1	PWM0C3[7:0]								00000000b	
SC1IE	Sc1 Interrupt Enable Control Register	DDH	2				ACERRIEN1	BGTIEN1	TERRIEN1	TBEIEN1	RDAIEN1	00000000b	
		DDH	3										
SC0ETURD1	Sc0etu Rate Divider Register1	DCH	0		SCDIV_2	SCDIV_1	SCDIV_0	ETURDIV_11	ETURDIV_10	ETURDIV_9	ETURDIV_8	00110001b	
PWM0C2L	Pwm0 Channel 2 Duty Low Byte	DCH	1	PWM0C2[7:0]								00000000b	
SC1ETURD1	Sc1etu Rate Divide Rregister 1	DCH	2		SCDIV_2	SCDIV_1	SCDIV_0	ETURDIV_11	ETURDIV_10	ETURDIV_9	ETURDIV_8	00000000b	
		DDH	3										
SC0ETURD0	Sc0etu Rate Divider Register 0	DBH	0	ETURDIV[7:0]								01110001b	
PWM0C1L	Pwm0 Channe L1 Duty Low Byte	DBH	1	PWM0C1[7:0]								00000000b	
SC1ETURD0	Sc1etu Rate Divider Register 0	DBH	2	ETURDIV[7:0]								00000000b	
		DBH	3										
SC0EGT	Sc Extra Guard Time Register	DAH	0	SCnEGT[7:0]								00000000b	
PWM0C0L	Pwm0 Channel 0 Duty Low Byte	DAH	1	PWM0C0[7:0]								00000000b	
SC1EGT	Sc1 Extra Guard Time Register	DAH	2	SC1EGT[7:0]								00000000b	
		DAH	3										
SC0DR	Sc Data Register	D9H	0	SC0DR[7:0]								00000000b	
PWM0PL	Pwm Period Low Byte	D9H	1	PWM0P[7:0]								00000000b	
SC1DR	Sc1 Data Register	D9H	2	SC1DR[7:0]								00000000b	
		D9H	3										
P4	Port4	D8H	A	P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0	Outputlatch, 00000000b Input, XXXXXXXXb	
SC0CR1	Sc0 Control Register 1	D7H	0	OPE	PBOFF	WLS1	WLS0	TXDMAEN	RXDMAEN	CLKKEEP	UARTEN	00000000b	
PWM0FBD	Brake Data	D7H	1	FBF	FBINLS	FBD5	FBD4	FBD3	FBD2	FBD1	FBD0	00000000b	
I2C1ADDRM	I2c1 Address Mask	D7H	2	I2C1ADDRM								00000000b	
		D7H	3										
SC0CR0	Sc0 Control Register 0	D6H	0	NSB	T	RXBGTEN	CONSEL	AUTOCEN	TXOFF	RXOFF	SCEN	00000000b	
PWM0NP	Pwm Negative Polarity	D6H	1			PNP5	PNP4	PNP3	PNP2	PNP1	PNP0	00000000b	
PWM3INTC	Pwm3 Interrupt Control	D6H	2			INTTYP1	INTTYP0		INTSEL2	INTSEL1	INTSEL0	00000000b	
		D6H	3										
$ACMPV_{REF}$	Acmp Reference Voltage Control	D5H	0	-	CRV1CTL_2	CRV1CTL_1	CRV1CTL_0	-	CRV0CTL_2	CRV0CTL_1	CRV0CTL_0	00000000b	
PWM0C3H	Pwm0 Channel 3	D5H	1	PWM0C3[15:8]								00000000b	

Register	Definition	Addr.	Page	MSB							LSB	Reset Value	TA
				7	6	5	4	3	2	1	0		
	Duty High Byte												
PWM3CON1	Pwm3 Control 1	D5H	2	PWMMOD_1	PWMMOD_0	GP	PWMTYP	FBINEN	PWMDIV_2	PWMDIV_1	PWMDIV_0	00000000b	
		D5H	3										
ACMPSR	Analog Comparator Status Register	D4H	0				-	ACMP1O	ACMP1IF	ACMP0O	ACMP0IF	00000000b	
PWM0C2H	Pwm0 Channel 2 Duty High Byte	D4H	1	PWM0C3[15:8]								00000000b	
PWM3CON0	Pwm3 Control 0	D4H	2	PWM3RUN	LOAD	PWMF	CLRPWM					00000000b	
		D4H	3										
ACMPCR1	Analog Comparator Control Register1	D3H	0	POSSEL_1	POSSEL_0	NEGSEL_1	NEGSEL_0	WKEN	HYSEN	ACMPIE	ACMPEN	00000000b	
PWM0C1H	Pwm0 Channel 1 Duty High Byte	D3H	1	PWM0C1[15:8]								00000000b	
PWM3C1L	Pwm3 Channel 1 Duty Low Byte	D3H	2	PWM3C1[7:0]								00000000b	
		D3H	3										
ACMPCR0	Analog Comparator Control Register 0	D2H	0	POSSEL_1	POSSEL_0	NEGSEL_1	NEGSEL_0	WKEN	HYSEN	ACMPIE	ACMPEN	00000000b	
PWM0C0H	Pwm0 Channel 0 Duty High Byte	D2H	1	PWM0C1[15:8]								00000000b	
PWM3C0L	Pwm3 Channel 0 Duty Low Byte	D2H	2	PWM3C0[7:0]								00000000b	
		D2H	3										
PWM0CON0	Pwm0 Control Register 0	D1H	0	PWMRUN	LOAD	PWMF	CLRPWM					00000000b	
PWM0PH	Pwm0 Period High Byte	D1H	1	PWM0P[15:8]								00000000b	
PWM3PL	Pwm0 Period Low Byte	D1H	2	PWM3P[7:0]								00000000b	
		D1H	3										
PSW	Program Status Word	D0H	A	CY	AC	F0	RS1	RS0	OV		P	00000000b	
ADCMPH	Adc Compare High Byte	CFH	0	ADCMP[11:4]								00000000b	
-	-	CFH	1		CRVTEST	-	-	-	-	-		-	-
I2C0ADDRM	I2c0 Address Mask	CFH	2	I2C0ADDRM								00000000b	
AUXR3	Auxiliary Register 3	CFH	3					UART3DG	UART2DG	UART1DG	UART0DG	00000000b	
ADCMPL	Adc Compare Low Byte	CEH	0					ADCMPL[3]	ADCMPL[2]	ADCMPL[1]	ADCMPL[0]	00000000b	
AINDIDS0	Adc Channel Digitalinput Disconnect	CEH	1	AIN7DIDS	AIN6DIDS	AIN5DIDS	AIN4DIDS	AIN3DIDS	AIN2DIDS	AIN1DIDS	AIN0DIDS	00000000b	
AINDIDS1	Adc Channel Digitalinput Disconnect	CEH	2	AIN15DIDS	AIN14DIDS	AIN13DIDS	AIN12DIDS	AIN11DIDS	AIN10DIDS			00000000b	
PWM0FBS	Pwm Brake Source Select	CEH	3							PWM0FBS_1	PWM0FBS_0	00000000b	
TH2	Timer 2 High Byte	CDH	0	T2[15:8]								00000000b	
PWM0C5L	Pwm0 Channel5 Duty Low Byte	CDH	1	PWM0C5[7:0]								00000000b	
PWM3MEN	Pwm Mask Enable	CDH	2							PMEN1	PMEN0	00000000b	
-	**-**	CDH	3					-		-		-	-
TL2	Timer 2 Low Byte	CCH	0	T2[7:0]								00000000b	
PWM0C4L	Pwm0 Channel 4 Duty Low Byte	CCH	1	PWM0C4[7:0]								00000000b	
PWM3MD	Pwm Mask Data	CCH	2							PMD1	PMD0	00000000b	
			3										
ADCBAL	Adc Ram Base Address Low Byte	CBH	0	ADCBA[7:0]								00000000b	
RCMP2H	Timer2 Reload / Compare High Byte	CBH	1	RCMP2[15:8]								00000000b	
PWM3C1H	Pwm3 Channel1 Duty High Byte	CBH	2	PWM3C1[15:8]								00000000b	
		CBH	3										
PIF	Pin Interrupt Flags	CAH	0	PIF7	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	00000000b	
RCMP2L	Timer2 Compare Low Byte	CAH	1	RCMP2[7:0]								00000000b	
PWM3C0H	Pwm3 Channel0 Duty High Byte	CAH	2	PWM3C0[15:8]								00000000b	

Register	Definition	Addr.	Page	MSB							LSB	Reset Value	TA
				7	6	5	4	3	2	1	0		
		CAH	3										
T2MOD	Timer2 Mode	C9H	0	LDEN	T2DIV_2	T2DIV_1	T2DIV_0	CAPCR	CMPCR	LDTS_1	LDTS_0	00000000b	
AUXR1	Auxiliary Register 1	C9H	1					UART3PX	UART2PX	UART1PX	UART0PX	00000000b	
PWM3PH	Pwm Period High Byte	C9H	2	PWM3P[15:8]								00000000b	
		C9H	3										
T2CON	Timer2 Control	C8H	A	TF2					TR2		CM_RL2	00000000b	
TA	Time Access Protection	C7H	A	TA[7:0]								00000000b	
RH3	Timer3 Reload Highbyte	C6H	0	RH3[15:8]								00000000b	
PORDIS	Por Disable	C6H	1	PORDIS[7:0]								00000000b	Y
PWM2INTC	Pwm Interrupt Control	C6H	2			INTTYP1	INTTYP0		INTSEL2	INTSEL1	INTSEL0	00000000b	
		C6H	3										
RL3	Timer3 Reload Low Byte	C5H	0	RL3[7:0]								00000000b	
PWM0C5H	Pwm0 Channel 5 Duty High Byte	C5H	1	PWM0C5[15:8]								00000000b	
PWM2CON1	Pwm Control 1	C5H	2	PWMMOD_1	PWMMOD_0	GP	PWMTYP	FBINEN	PWMDIV_2	PWMDIV_1	PWMDIV_0	00000000b	
		C5H	3										
T3CON	Timer3 Control	C4H	0	SMOD_1	SMOD0_1	BRCK	TF3	TR3	T3PS_2	T3PS_1	T3PS_0	00000000b	
PWM0C4H	Pwm0 Channel 4 Duty High Byte	C4H	1	PWM0C4[15:8]								00000000b	
PWM2CON0	Pwm Control Register 0	C4H	2	PWM2RUN	LOAD	PWMF	CLRPWM					00000000b	
		C4H	3										
ADCRH	Adc Result High Byte	C3H	0	ADCR[11:4]								00000000b	
P3M2	Port3 Mode Select 2	C3H	1	P3M2.7	P3M2.6	P3M2.5	P3M2.4	P3M2.3	P3M2.2	P3M2.1	P3M2.0	00000000b	
PWM2C1L	Pwm2 Channel 1 Duty Low Byte	C3H	2	PWM2C1[7:0]								00000000b	
		C3H	3										
ADCRL	Adc Result Low Byte	C2H	0					ADCR_3	ADCR_2	ADCR_1	ADCR_0	00000000b	
P3M1	Port3 Mode Select 1	C2H	1	P3M1.7	P3M1.6	P3M1.5	P3M1.4	P3M1.3	P3M1.2	P3M1.1	P3M1.0	11111111B	
PWM2C0L	Pwm2 Channel 0 Duty Low Byte	C2H	2	PWM2C0[7:0]								00000000b	
		C2H	3										
I2C0ADDR0	I^2c Own Slave Address	C1H	0	I2C0ADDR0_7	I2C0ADDR0_6	I2C0ADDR0_5	I2C0ADDR0_4	I2C0ADDR0_3	I2C0ADDR0_2	I2C0ADDR0_1	GC	00000000b	
CKDIV	Clock Divider	C1H	1	CKDIV[7:0]								00000000b	
PWM2PL	Pwm2 Period Low Byte	C1H	2	PWM2P[7:0]								00000000b	
		C1H	3										
I2C0CON	I2c0c Control	C0H	A	I	I2CEN	STA	STO	SI	AA			00000000b	
I2C0TOC	I2c0 Time Out Counter	BFH	0						I2TOCEN	DIV	I2TOF	00000000b	
P5S	Port5 Schmitt Triggered Input	BFH	1	P5S.7	P5S.6	P5S.5	P5S.4	P5S.3	P5S.2	P5S.1	P5S.0	00000000b	
RWKH	Self Wakeup Timerre Load High Byte	BFH	2	RWK[15:8]								11111111b	
RTCCAMSK	Rtc Calendar Alarm Mask Register	BFH	3			MTENYEAR	MYEAR	MTENMON	MMON	MTENDAY	MDAY	00000000b	
I2C0CLK	I2c0 Clock	BEH	0	I2C0CLK[7:0]								00000000b	
P5M2	Port5 Mode Select 2	BEH	1	P5M2.7	P5M2.6	P5M2.5	P5M2.4	P5M2.3	P5M2.2	P5M2.1	P5M2.0	00000000b	
CWKH	Self Wakeup Timer Current Count High Byte	BEH	2	CWK[15:8]								00000000b	
RTCTAMSK	Rtc Time Alarm Mask Register	BEH	3			MTENHR	MHR	MTENMIN	MMIN	MTENSEC	MSEC	00000000b	
I2C0STAT	I2c0 Status	BDH	0	I2C0STAT_7	I2C0STAT_6	I2C0STAT_5	I2C0STAT_4	I2C0STAT_3	0	0	0	11111000b	
P5M1	Port5 Mode Select 1	BDH	1	P5M1.7	P5M1.6	P5M1.5	P5M1.4	P5M1.3	P5M1.2	P5M1.1	P5M1.0	11111111b	
PWM2MEN	Pwm Mask Enable	BDH	2							PMEN1	PMEN0	00000000b	
RTCTICK	Rtc Time Tick Register	BDH	3						TICK_2	TICK_1	TICK_0	00000000b	

Register	Definition	Addr.	Page	MSB							LSB	Reset Value	TA
				7	6	5	4	3	2	1	0		
I2C0DAT	I2c0 Data	BCH	0	I2C0DAT[7:0]								00000000b	
P4SR	Port4 Slew Rate Control	BCH	1	P4SR.7	P4SR.6	P4SR.5	P4SR.4	P4SR.3	P4SR.2	P4SR.1	P4SR.0	00000000b	
PWM2MD	Pwm Mask Data	BCH	2							PMD1	PMD0	00000000b	
RTCLEAPYEAR	Rtc Leap Year Indicator Register	BCH	3								LEAPYEAR	00000000b	
SADDR1	Slave1 Address	BBH	0	SADDR1[7:0]								00000000b	
P4S	Port4 Schmitt Triggered Input	BBH	1	P4S.7	P4S.6	P4S.5	P4S.4	P4S.3	P4S.2	P4S.1	P4S.0	00000000b	
PWM2C1H	Pwm 2 Channel1 Duty High Byte	BBH	2	PWM2C1[15:8]								00000000b	
RTCWEEKDAY	Rtc Day Of The Week Register	BBH	3						WEEKDAY_2	WEEKDAY_1	WEEKDAY_0	00000110b	
SADEN1	Slave1 Address Mask	BAH	0	SADEN1[7:0]								00000000b	
P4M2	Port4 Mode Select 2	BAH	1	P4M2.7	P4M2.6	P4M2.5	P4M2.4	P4M2.3	P4M2.2	P4M2.1	P4M2.0	00000000b	
PWM2C0H	Pwm2 Channel 0 Duty High Byte	BAH	2	PWM2C0[15:8]								00000000b	
-	-	BAH	3								-	-	-
SADEN0	Slave 0 Address Mask	B9H	0	SADEN0[7:0]								00000000b	
P4M1	Port 4 Mode Select 1	B9H	1	P4M1.7	P4M1.6	P4M1.5	P4M1.4	P4M1.3	P4M1.2	P4M1.1	P4M1.0	11111111b	
PWM2PH	Pwm Period High Byte	B9H	2	PWM2P[15:8]								00000000b	
RTCCLKFMT	Rtc Time Scale Selection Register	B9H	3							-	24HEN	00000001b	
IP	Interrupt Priority	B8H	A		PADC	PBOD	PS	PT1	PX1	PT0	PX0	00000000b	
IPH	Interrupt Priority High	B7H	0		PADCH	PBODH	PSH	PT1H	PX1H	PT0H	PX0H	00000000b	
PWM0INTC	Pwm0 Interrupt Control	B7H	1			INTTYP1	INTTYP0		INTSEL2	INTSEL1	INTSEL0	00000000b	
MTM2DA	Memory To Memory Destination Address Low Byte	B7H	2	MDAL[7:0]								00000000b	
RTCCALMYEAR	Rtc Calendar Year Loading Register	B7H	3	TENYEAR_3	TENYEAR_2	TENYEAR_1	TENYEAR_0	YEAR_3	YEAR_2	YEAR_1	YEAR_0	00000000b	
I2C1TOC	I2c1 Timeout Counter	B6H	0						I2TOCEN	DIV	I2TOF	00000000b	
P2M2	Port5 Mode Select 2	B6H	1	P2M2.7	P2M2.6	P2M2.5	P2M2.4	P2M2.3	P2M2.2	P2M2.1	P2M2.0	00000000b	
DMA2CCNT	PDMA12 Current Transfer Count	B6H	2	DMA2CCNT[7:0]								00000000b	
RTCCALMMON	Rtc Calendar Month Alarm Register	B6H	3				TENMON	MON_3	MON_2	MON_1	MON_0	00000000b	
I2C1CLK	I2c1 Clock	B5H	0	I2C1CLK[7:0]								00000000b	
P2M1	Port2 Mode Select 1	B5H	1	P2M1.7	P2M1.6	P2M1.5	P2M1.4	P2M1.3	P2M1.2	P2M1.1	P2M1.0	11111111b	
DMA2CNT	PDMA13 Transfer Count	B5H	2	DMA2CNT[7:0]								00000000b	
RTCCALMDAY	Rtc Calendar Day Alarm Register	B5H	3			TENDAY_1	TENDAY_0	DAY_3	DAY_2	DAY_1	DAY_0	00000000b	
I2C1STAT	I2c1 Status	B4H	0	I2C1STAT_7	I2C1STAT_6	I2C1STAT_5	I2C1STAT_4	I2C1STAT_3	0	0	0	11111000b	
P1M2	P1 Mode Select 2	B4H	1	P1M2.7	P1M2.6	P1M2.5	P1M2.4	P1M2.3	P1M2.2	P1M2.1	P1M2.0	00000000b	
DMA2MAL	PDMA12 Xram Base Address Low Byte	B4H	2	MAL[7:0]								00000000b	
-	-	B4H	3		-							-	
I2C1DAT	I2c1 Data	B3H	0	I2C1DAT[7:0]								00000000b	
P1M1	P1 Mode Select 1	B3H	1	P1M1.7	P1M1.6	P1M1.5	P1M1.4	P1M1.3	P1M1.2	P1M1.1	P1M1.0	11111111b	
DMA2CR0	PDMA12 Control Register 0	B3H	2	PSSEL_3	PSSEL_2	PSSEL_1	PSSEL_0	HIE	FIE	RUN	EN	00000000b	
RTCTALMHR	Rtc Time Hour Alarm Register	B3H	3			TENHR_1	TENHR_0	HR_3	HR_2	HR_1	HR_0	00000000b	
I2C1ADDR0	I2c1 Own Slave Address 0	B2H	0	I2C1ADDR0_7	I2C1ADDR0_6	I2C1ADDR0_5	I2C1ADDR0_4	I2C1ADDR0_3	I2C1ADDR0_2	I2C1ADDR0_1	GC	00000000b	
P0M2	P0 Mode Select 2	B2H	1	PnM2_7	PnM2_6	PnM2_5	PnM2_4	PnM2_3	PnM2_2	PnM2_1	PnM2_0	00000000b	
DMA2BAH	PDMA12 Xram Base Address High Byte	B2H	2	MTMDA[7:4]				XRAMA[7:4]				00000000b	
RTCTALMMIN	Rtc Time Alarm Minute Register	B2H	3		TENMIN_2	TENMIN_1	TENMIN_0	MIN_3	MIN_2	MIN_1	MIN_0	00000000b	

Register	Definition	Addr.	Page	MSB							LSB	Reset Value	TA
				7	6	5	4	3	2	1	0		
P5	Port 5	B1H	0	P5.7	P5.6	P5.5	P5.4	P5.3	P5.2	P5.1	P5.0	Outputlatch, 00000000b Input, XXXXXXXXb	
P0M1	P0 Mode Select 1	B1H	1	P0M1_7	P0M1_6	P0M1_5	P0M1_4	P0M1_3	P0M1_2	P0M1_1	P0M1_0	11111111b	
DMA2TSR	PDMA1 N Transfer Status Register	B1H	2					-	ACT	HDONE	FDONE	00000000b	
RTCTALMSEC	Rtc Time Second Alarm Register	B1H	3		TENSEC_2	TENSEC_1	TENSEC_0	SEC_3	SEC_2	SEC_1	SEC_0	00000000b	
P3	Port 3	B0H	A	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	Outputlatch, 00000000b Input, XXXXXXXXb	
IAPCN	Iap Control	AFH	0	IAPA_17	IAPA_16	FOEN	FCEN	FCTRL_3	FCTRL_2	FCTRL_1	FCTRL_0	00110000b	
PIPS7	Pin Interrupt Control Register 7	AFH	1		PSEL_2	PSEL_1	PSEL_0		BSEL_2	BSEL_1	BSEL_0	00000000b	
MTM3DA	Memory To Memory 3 Destination Address Low Byte	AFH	2	MTM3DA[7:0]								00000000b	
RTCCALYEAR	Rtc Calendar Year Loading Register	AFH	3	TENYEAR				YEAR				00010101b	
IAPFD	Iap Flash Data	AEH	0	IAPFD[7:0]								00000000b	
P5SR	P5 Slew Rate	AEH	1	P5SR.7	P5SR.6	P5SR.5	P5SR.4	P5SR.3	P5SR.2	P5SR.1	P5SR.0	00000000b	
DMA3CCNT	PDMA12 Current Transfer Count	AEH	2	DMA3CCNT[7:0]								00000000b	
RTCCALMON	Rtc Calendar Month Loading Register	AEH	3			TENMON_1	TENMON_0	MON_3	MON_2	MON_1	MON_0	00001000b	
EIPH2	Extensive Interrupt Priority High 2	ADH	0	RTCH	PDMA3H	PDMA2H	SMC1H	TKH	PPWM1H	PI2C1H	PACMPH	00000000b	
P3SR	P3 Slew Rate	ADH	1	P3SR.7	P3SR.6	P3SR.5	P3SR.4	P3SR.3	P3SR.2	P3SR.1	P3SR.0	00000000b	
DMA3CNT	PDMA13 Transfer Count	ADH	2	DMA2CNT[7:0]								00000000b	
RTCCALDAY	Rtc Calendar–Day Loading Register	ADH	3			TENDAY		DAY				00001000b	
EIP2	Extensive Interrupt Priority 2 Rcnt	ACH	0	RTC	PDMA3	PDMA2	SMC1	TK	PPWM1	PI2C1	PACMP	00000000b	
P3S	Port 3 Schmitt Triggered Input	ACH	1	P3S.7	P3S.6	P3S.5	P3S.4	P3S.3	P3S.2	P3S.1	P3S.0	00000000b	
DMA3MAL	PDMA13 Xram Base Address Low Byte	ACH	2	MAL[7:0]								00000000b	
-	-	ACH	3		-							-	
BODCON1	Brownout Detection Control 1	ABH	0						LPBOD_1	LPBOD_0	BODFLT	POR: 00000001b Others: 00000UUUb	Y
ACMPCR2	Analog Comparator Control Register 2	ABH	1	SPEED1_1	SPEED1_0	POE1	POE0	SPEED0_1	SPEED0_0	CRVSSEL	CRVEN	00000000b	
DMA3CR0	PDMA13 Control Register 0	ABH	2	PSSEL_3	PSSEL_2	PSSEL_1	PSSEL_0	HIE	FIE	RUN	EN	00000000b	
RTCTIMEHR	Rtc Time Hour Loading Register	ABH	3			TENHR_1	TENHR_0	HR_3	HR_2	HR_1	HR_0	00000000b	
WDCON	Watchdog Timer Control	AAH	0	WDTR	WDCLR	WDTF	WIDPD	WDTRF	WDPS_2	WDPS_1	WDPS_0	POR: 00000111b WDT: 00001UUUb Others: 0000UUUUb	Y
VRFTRIM	Internal Voltage Reference Trim	AAH	1									00000000b	Y
DMA3BAH	PDMA1 3 Xram Base Address High Byte	AAH	2	MTMDA[7:4]				XRAMA[7:4]				00000000b	
RTCTIMEMIN	Rtc Time Minute Loading Register	AAH	3		TENMIN_2	TENMIN_1	TENMIN_0	MIN_3	MIN_2	MIN_1	MIN_0	00000000b	
SADDR0	Slave 0 Address	A9H	0	SADDR0[7:0]								00000000b	
VRFCON	Internal V_{ref} Control	A9H	1		VRFSEL_2	VRFSEL_1	VRFSEL_0			ENLOAD	ENVRF	00000000b	Y
DMA3TSR	PDMA1 3 Transfer Status Register	A9H	2					-	ACT	HDONE	FDONE	00000000b	
RTCTIMESEC	Rtc Time Second Loading Register	A9H	3		TENSEC_2	TENSEC_1	TENSEC_0	SEC_3	SEC_2	SEC_1	SEC_0	00000000b	
IE	Interrupt Enable	A8H	A	EA	EADC	EBOD	ES	ET1	EX1	ET0	EX0	00000000b	
IAPAH	Iap Address High Byte	A7H	0	IAPA[15:8]								00000000b	
PIPS6	Pin Interrupt	A7H	1		PSEL_2	PSEL_1	PSEL_0		BSEL_2	BSEL_1	BSEL_0	00000000b	

Register	Definition	Addr.	Page	MSB							LSB	Reset Value	TA
				7	6	5	4	3	2	1	0		
	Control 6												
P6	Port6	A7H	2	P6.7	P6.6	P6.5	P6.4	P6.3	P6.2	P6.1	P6.0	Outputlatch, 00000000b Input, XXXXXXXXb	
RTCINTSTS	Rtc Interrupt Status Register	A7H	3							TICKIF	ALMIF	00000000b	
IAPAL	Iap Address Low Byte	A6H	0	IAPA[7:0]								00000000b	
PIPS5	Pin Interrupt Control 5	A6H	1		PSEL_2	PSEL_1	PSEL_0		BSEL_2	BSEL_1	BSEL_0	00000000b	
I2C1ADDR3	I2cn Own Slave Address	A6H	2	I2C1ADDR3_7	I2C1ADDR3_6	I2C1ADDR3_5	I2C1ADDR3_4	I2C1ADDR3_3	I2C1ADDR3_2	I2C1ADDR3_1	GC	00000000b	
RTCINTEN	Rtc Interrupt Enable Register	A6H	3							TICKIEN	ALMIEN	00000000b	
IAPUEN	Iap Update Enable	A5H	0				SPMEN	SPUEN	CFUEN	LDUEN	APUEN	00000000b	Y
PIPS4	Pin Interrupt Control 4	A5H	1		PSEL_2	PSEL_1	PSEL_0		BSEL_2	BSEL_1	BSEL_0	00000000b	
I2C1ADDR2	I2cn Own Slave Address	A5H	2	I2C1ADDR2_7	I2C1ADDR2_6	I2C1ADDR2_5	I2C1ADDR2_4	I2C1ADDR2_3	I2C1ADDR2_2	I2C1ADDR2_1	GC	00000000b	
RTCFREQADJ1	Rtc Frequency Compensation1 (Integer) Register	A5H	3					INTEGER				00010000b	
IAPTRG	Iap Trigger	A4H	0								IAPGO	00000000b	Y
PIPS3	Pin Interrupt Control3	A4H	1		PSEL_2	PSEL_1	PSEL_0		BSEL_2	BSEL_1	BSEL_0	00000000b	
I2C1ADDR1	I2cn Own Slave Address	A4H	2	I2C1ADDR1_7	I2C1ADDR1_6	I2C1ADDR1_5	I2C1ADDR1_4	I2C1ADDR1_3	I2C1ADDR1_2	I2C1ADDR1_1	GC	00000000b	
RTCFREQADJ0	Rtc Frequency Compensation 0 (Fraction) Register	A4H	3			FRACTION						00000000b	
BODCON0	Brownout Detection Control 0	A3H	0	BODEN	BOV2	BOV1	BOV0	BOF	BORST	BORF	BOS	POR: CCCCXC0Xb BOD: UUUUXU1Xb Others: UUUUXUUXb	Y
PIPS2	Pin Interrupt Control 2	A3H	1		PSEL_2	PSEL_1	PSEL_0		BSEL_2	BSEL_1	BSEL_0	00000000b	
I2C0ADDR3	I2cn Own Slave Address	A3H	2	I2C0ADDR3_7	I2C0ADDR3_6	I2C0ADDR3_5	I2C0ADDR3_4	I2C0ADDR3_3	I2C0ADDR3_2	I2C0ADDR3_1	GC	00000000b	
RTCCLKSEL	Rtc Clock Select Register	A3H	3								C32KS	00000000b	
AUXR0	Auxiliary Register 0	A2H	0	SWRF	RSTPINF	HardF	HardFInt	GF2		0	DPS	00000000b	
PIPS1	Pin Interrupt Control 1	A2H	1		PSEL_2	PSEL_1	PSEL_0		BSEL_2	BSEL_1	BSEL_0	00000000b	
I2C0ADDR2	I2cn Own Slave Address	A2H	2	I2C0ADDR2_7	I2C0ADDR2_6	I2C0ADDR2_5	I2C0ADDR2_4	I2C0ADDR2_3	I2C0ADDR2_2	I2C0ADDR2_1	GC	00000000b	
RTCRWEN	Rtc Access Enable Register	A2H	3					-			RWENF	00000001b	
ADCCON0	Adc Control Register 0	A1H	0	ADCF	ADCS	ETGSEL1	ETGSEL0	ADCHS3	ADCHS2	ADCHS1	ADCHS0	00000000b	
PIPS0	Pin Interrupt Control 1	A1H	1		PSEL_2	PSEL_1	PSEL_0		BSEL_2	BSEL_1	BSEL_0	00000000b	
I2C0ADDR1	I2cn Own Slave Address	A1H	2	I2C0ADDR1_7	I2C0ADDR1_6	I2C0ADDR1_5	I2C0ADDR1_4	I2C0ADDR1_3	I2C0ADDR1_2	I2C0ADDR1_1		00000000b	
RTCINIT	Rtc Initiation Register	A1H	3	INIT_7	INIT_6	INIT_5	INIT_4	INIT_3	INIT_2	INIT[1]/RTCHOLD	INIT[0]/ACTIVE	00000000b	Y
P2	Port 2	A0H	A	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	Outputlatch, 00000000b Input, XXXXXXXXb	
CHPCON	Chip Control	9FH	0	SWRST	IAPFF					BS	IAPEN	Software, 000000U0b Others, 000000C0b	Y
		9FH	1										
		9FH	2										
		9FH	3										
-	-	9EH	0		-	-	-		-	-	-	-	-
P2SR	P2 Slew Rate	9EH	1	P2SR.7	P2SR.6	P2SR.5	P2SR.4	P2SR.3	P2SR.2	P2SR.1	P2SR.0	00000000b	
PWM1INTC	Pwm Interrupt Control	9EH	2			INTTYP1	INTTYP0		INTSEL2	INTSEL1	INTSEL0	00000000b	
		9EH	3										
RSR	Reset Flag	9DH	0	LVRF	PORF	HFRF	POF	RSTPINF	BORF	WDTRF	SWRF	11010000b	

Register	Definition	Addr.	Page	MSB							LSB	Reset Value	TA
				7	6	5	4	3	2	1	0		
	Register												
P2S	Port2 Schmitt Triggered Input	9DH	1	P2S.7	P2S.6	P2S.5	P2S.4	P2S.3	P2S.2	P2S.1	P2S.0	00000000b	
PWM1CON1	Pwm1 Control 1	9DH	2	PWMMOD_1	PWMMOD_0	GP	PWMTYP	FBINEN	PWMDIV_2	PWMDIV_1	PWMDIV_0	00000000b	
DMA3SEED	PDMA1 Crc Seed	9DH	3	SEED[7:0]								00000000b	
EIE1	Extensive Interrupt Enable 1	9CH	0		EPWM123	EI2C1	ESPI1	EHFI	EWKT	ET3	ES1	00000000b	
P1SR	P2 Slewrate	9CH	1	P2SR.7	P2SR.6	P2SR.5	P2SR.4	P2SR.3	P2SR.2	P2SR.1	P2SR.0	00000000b	
PWM1CON0	Pwm Control Register 0	9CH	2	PWM1RUN	LOAD	PWMF	CLRPWM					00000000b	
DMA2SEED	PDMA1 Crc Seed	9CH	3	SEED[7:0]								00000000b	
EIE0	Extensive Interrupt Enable 1	9BH	0	ET2	ESPI0	EFB0	EWDT	EPWM0	ECAP	EPI	EI2C0	00000000b	
P1S	Port1 Schmitt Triggered Input	9BH	1	P1SR.7	P1SR.6	P1SR.5	P1SR.4	P1SR.3	P1SR.2	P1SR.1	P1SR.0	00000000b	
PWM1C1L	Pwm1 Channel 1 Duty Low Byte	9BH	2	PWM1C1[7:0]								00000000b	
DMA1SEED	PDMA1 Crc Seed	9BH	3	SEED[7:0]								00000000b	
SBUF1	Serial Port1 Data Buffer	9AH	0	SBUF1[7:0]								00000000b	
P0SR	P2 Slew Rate	9AH	1	P0SR.7	P0SR.6	P0SR.5	P0SR.4	P0SR.3	P0SR.2	P0SR.1	P0SR.0	00000000b	
PWM1C0L	Pwm1 Channel 0 Duty Low Byte	9AH	2	PWM1C0[7:0]								00000000b	
DMA0SEED	PDMA1 Crc Seed	9AH	3	SEED[7:0]								00000000b	
SBUF	Serial Port0 Data Buffer	99H	0	SBUF[7:0]								00000000b	
P0S	Port0 Schmitt Triggered Input	99H	1	P0SR.7	P0SR.6	P0SR.5	P0SR.4	P0SR.3	P0SR.2	P0SR.1	P0SR.0	00000000b	
PWM1PL	Pwm Period Low Byte	99H	2	PWM1P[7:0]								00000000b	
WDCONH	Watchdog Timer Control High Byte	99H	3								WDPS	00000000b	Y
SCON	Serial Port0 Control	98H	A	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	00000000b	
CKEN	Clock Enable	97H	0	EHXTEN	ELXTEN	HIRCEN	LIRCEN	ECLKEN	-		CKSWTF	00110100b	Y
P5UP	Port5 Pull Up Resister Control	97H	1	P5UP.7	P5UP.6	P5UP.5	P5UP.4	P5UP.3	P5UP.2	P5UP.1	P5UP.0	00000000b	
P6UP	Port6 Pull Up Resister Control	97H	2	P6UP.7	P6UP.6	P6UP.5	P6UP.4	P6UP.3	P6UP.2	P6UP.1	P6UP.0	00000000b	
		97H	3										
CKSWT	Clock Switch	96H	0	HXTST	LXTST	HIRCST	LIRCST	ECLKST	OSC_2	OSC_1	OSC_0	00110000b	Y
P4UP	Port4 Pull Up Resister Control	96H	1	P4UP.7	P4UP.6	P4UP.5	P4UP.4	P4UP.3	P4UP.2	P4UP.1	P4UP.0	00000000b	
P6S	Port6 Schmitt Triggered Input	96H	2	P6S.7	P6S.6	P6S.5	P6S.4	P6S.3	P6S.2	P6S.1	P6S.0	00000000b	
		96H	3										
DMA0CCNT	PDMA11 Current Transfer Count	95H	0	DMA0CCNT[7:0]								00000000b	
P3UP	Port3 Pull Up Resister Control	95H	1	P3UP.7	P3UP.6	P3UP.5	P3UP.4	P3UP.3	P3UP.2	P3UP.1	P3UP.0	00000000b	
P6MF76	P6.7 And P6.6 Multi Function Select	95H	2	P6MF7[3:0]				P6MF6[3:0]				00000000b	
DMA3CRC	PDMA1 Crc Checksum	95H	3	CRC[7:0]								00000000b	
DMA0CNT	PDMA10 Transfer Count	94H	0	DMA1CNT[7:0]								00000000b	
P2UP	Port2 Pull Up Resister Control	94H	1	P2UP.7	P2UP.6	P2UP.5	P2UP.4	P2UP.3	P2UP.2	P2UP.1	P2UP.0	00000000b	
P6MF54	P6.5 And P6.4 Multi Function Select	94H	2	P6MF5[3:0]				P6MF4[3:0]				00000000b	
DMA2CRC	PDMA1 Crc Checksum	94H	3	CRC[7:0]								00000000b	
DMA0MAL	PDMA10 Xram Base Address Low Byte	93H	0	MAL[7:0]								00000000b	
P1UP	Port1 Pull Up Resister Control	93H	1	P1UP.7	P1UP.6	P1UP.5	P1UP.4	P1UP.3	P1UP.2	P1UP.1	P1UP.0	00000000b	
P6MF32	P6.3 And P6.2 Multi Function Select	93H	2	P6MF3[3:0]				P6MF2[3:0]				00000000b	
DMA1CRC	PDMA1 Crc Checksum	93H	3	CRC[7:0]								00000000b	
DMA0CR0	PDMA1 0 Control	92H	0	PSSEL_3	PSSEL_2	PSSEL_1	PSSEL_0	HIE	FIE	RUN	EN	00000000b	

Register	Definition	Addr.	Page	MSB							LSB	Reset Value	TA
				7	6	5	4	3	2	1	0		
	Register												
P0UP	Port0 Pull Upresister Control	92H	1	P0UP.7	P0UP.6	P0UP.5	P0UP.4	P0UP.3	P0UP.2	P0UP.1	P0UP.0	00000000b	
P6MF10	P6.1 And P6.0 Multi Function Select	92H	2	P6MF1[3:0]				P6MF0[3:0]				00000000b	
DMA0CRC	PDMA1 Crc Checksum	92H	3	CRC[7:0]								00000000b	
SFRS	Sfr Page Selection	91H	A							SFRPAGE_1	SFRPAGE_0	00000000b	
P1	Port1	90H	A	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	Outputlatch, 00000000b Input, XXXXXXXXb	
WKCON	Self Wakeup Timer Control	8FH	0			WKTCK	WKTF	WKTR	WKPS_2	WKPS_1	WKPS_0	00000000b	
P5DW	Port 5 Pull Down Resister Control	8FH	1	P5DW.7	P5DW.6	P5DW.5	P5DW.4	P5DW.3	P5DW.2	P5DW.1	P5DW.0	00000000b	
P6DW	Port6 Pull Down Resister Control	8FH	2	P6DW.7	P6DW.6	P6DW.5	P6DW.4	P6DW.3	P6DW.2	P6DW.1	P6DW.0	00000000b	
		8FH	3										
CKCON	Clock Control	8EH	0	FASTWK	PWMCKS	T1OE	T1M	T0M	T0OE	CLOEN	-	10000000b	
P4DW	Port4 Pull Down Resister Control	8EH	1	P4DW.7	P4DW.6	P4DW.5	P4DW.4	P4DW.3	P4DW.2	P4DW.1	P4DW.0	00000000b	
P6SR	P6 Slew Rate	8EH	2	P6SR.7	P6SR.6	P6SR.5	P6SR.4	P6SR.3	P6SR.2	P6SR.1	P6SR.0	00000000b	
		8EH	3										
TH1	Timer 1 High Byte	8DH	0	TH1[7:0]								00000000b	
P3DW	Port3 Pull Down Resister Control	8DH	1	P3DW.7	P3DW.6	P3DW.5	P3DW.4	P3DW.3	P3DW.2	P3DW.1	P3DW.0	00000000b	
PWM1MEN	Pwm Mask Data	8DH	2							PMEN1	PMEN0	00000000b	
DMA3CR1	PDMA1 3 Control Register 1	8DH	3					XOROUT	REFOUT	REFIN	CRCEN	00000000b	
TH0	Timer 0 High Byte	8CH	0	TH0[7:0]								00000000b	
P2DW	Port2 Pull Down Resister Control	8CH	1	P2DW.7	P2DW.6	P2DW.5	P2DW.4	P2DW.3	P2DW.2	P2DW.1	P2DW.0	00000000b	
PWM1MD	Pwm 1 Mask Data	8CH	2							PMD1	PMD0	00000000b	
DMA2CR1	PDMA1 3 Control Register 1	8CH	3					XOROUT	REFOUT	REFIN	CRCEN	00000000b	
TL1	Timer 1 Low Byte	8BH	0	TL1[7:0]								00000000b	
P1DW	Port1 Pull Down Resister Control	8BH	1	P1DW.7	P1DW.6	P1DW.5	P1DW.4	P1DW.3	P1DW.2	P1DW.1	P1DW.0	00000000b	
PWM1C1H	Pwm 1 Channel 1 Duty High Byte	8BH	2	PWM1C1[15:8]								00000000b	
DMA1CR1	PDMA1 3 Control Register 1	8BH	3					XOROUT	REFOUT	REFIN	CRCEN	00000000b	
TL0	Timer 0 Low Byte	8AH	0	TL0[7:0]								00000000b	
P0DW	Port0 Pull Down Resister Control	8AH	1	P0DW.7	P0DW.6	P0DW.5	P0DW.4	P0DW.3	P0DW.2	P0DW.1	P0DW.0	00000000b	
PWM1C0H	Pwm 1 Channel 0 Duty High Byte	8AH	2	PWM1C0[15:8]								00000000b	
DMA0CR1	PDMA1 Control Register 1	8AH	3					XOROUT	REFOUT	REFIN	CRCEN	00000000b	
TMOD	Timer 0 And 1 Mode	89H	A	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00000000b	
TCON	Timer 0 And 1 Control	88H	A	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00000000b	
PCON	Power Control	87H	A	SMOD	SMOD0	LPR	POF	GF1	GF0	PD	IDL	POR, 00010000b Others, 000U0000b	
RWKL	Self Wakeup Timer Reload Lowbyte	86H	0	RWK[7:0]								00000000b	
CWKL	Self Wakeup Timer Current Count Low Byte	86H	1	CWK[7:0]								00000000b	
PWM1PH	Pwm Period High Byte	86H	2	PWM1P[15:8]								00000000b	
		86H	3										
RCTRIM1	Internal Rctrim Value Low Byte	85H	0								HIRCTRIM_0	0000000Xb	Y
XLTCON	Xlt Clock Control	85H	1	HSCH	HXSG_3	HXSG_2	HXSG_1			LXSG_1	LXSG_0		Y
P6M2	Port6 Mode Select 2	85H	2	P6M2.7	P6M2.6	P6M2.5	P6M2.4	P6M2.3	P6M2.2	P6M2.1	P6M2.0	00000000b	
		85H	3										
RCTRIM0	Internal Rc Trim Value High Byte	84H	0	HIRCTRIM[8:1]								XXXXXXXXb	Y

Register	Definition	Addr.	Page	MSB							LSB	Reset Value	TA
				7	6	5	4	3	2	1	0		
LIRCTRIM	Lirc Trim Value	84H	1	LIRCTRIM[7:0]								XXXXXXXXb	
P6M1	Port6 Mode Select 1	84H	2	P6M1.7	P6M1.6	P6M1.5	P6M1.4	P6M1.3	P6M1.2	P6M1.1	P6M1.0	00000000b	
		84H	3										
DPH	Data Pointer High Byte	83H	A	DPTR[15:8]								00000000b	
DPL	Data Pointer Low Byte	82H	A	DPTR[7:0]								00000000b	
SP	Stack Pointer	81H	A	SP[7:0]								00000111b	
P0	Port 0	80H	A	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	Outputlatch, 00000000b Input, XXXXXXXXb	

Table 6.2-1 SFR Definitions And Reset Values

Note:
[1] ( ) item means the page of SFRs. Without special define means all pages accessable.
[2] Reset value symbol description. 0: logic 0; 1: logic 1; U: unchanged; C: see [5]; X: see [3], [6], and [7].
[3] All I/O pins are default input-only mode (floating) after reset.
[4] These SFRs have TA protected writing.
[5] These SFRs have bits those are initialized according to CONFIG values after specified resets.
[6] BOF reset value depends on different setting of CONFIG2 and V_{DD} voltage level.
[7] BOS is a read-only flag decided by V_{DD} level while brown-out detection is enabled.

6.2.1.2 All SFR Description

Note: the reset value show as following means U-unchanged; C-initialized by CONFIG; X- base on real chip status.

Pn – Port

Register	SFR Address	Reset Value
P0	80H, All pages, Bit-addressable	1111_1111 b
P1	90H, All pages, Bit-addressable	1111_1111 b
P2	A0H, All pages, Bit-addressable	1111_1111 b
P3	B0H, All pages, Bit-addressable	1111_1111 b
P4	D8H, All pages, Bit-addressable	1111_1111 b
P5	B1H, Page 0	1111_1111 b
P6	A7H, Page 2	1111_1111 b

7	6	5	4	3	2	1	0
Pn.7	**Pn.6**	**Pn.5**	**Pn.4**	**Pn.3**	**Pn.2**	**Pn.1**	**Pn.0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:0]	**Pn[7:0]**	**Port n** Port n is an maximum 8-bit general purpose I/O port.

SP – Stack Pointer

Register	SFR Address	Reset Value
SP	81H, All pages	0000_0111b

7	6	5	4	3	2	1	0
SP[7:0]							
R/W							

Bit	Name	Description
[7:0]	**SP[7:0]**	**Stack Pointer** The Stack Pointer stores the scratch-pad RAM address where the stack begins. It is incremented before data is stored during PUSH or CALL instructions. Note that the default value of SP is 07H. This causes the stack to begin at location 08H.

DPL – Data Pointer Low Byte

Register	SFR Address	Reset Value
DPL	82H, All pages	0000_0000b

7	6	5	4	3	2	1	0
DPTR[7:0]							
R/W							

Bit	Name	Description
[7:0]	DPTR[7:0]	**Data Pointer Low Byte** This is the low byte of 16-bit data pointer. DPL combined with DPH serve as a 16-bit data pointer DPTR to access indirect addressed RAM or Program Memory. DPS (AUXR1.0) bit decides which data pointer, DPTR or DPTR1, is activated.

DPH – Data Pointer High By

Register	SFR Address	Reset Value
DPH	83H, All pages	0000_0000b

7	6	5	4	3	2	1	0
DPTR[15:8]							
R/W							

Bit	Name	Description
[7:0]	**DPTR[15:8]**	**Data Pointer High Byte** This is the high byte of 16-bit data pointer. DPH combined with DPL serve as a 16-bit data pointer DPTR to access indirect addressed RAM or Program Memory. DPS (AUXR1.0) bit decides which data pointer, DPTR or DPTR1, is activated.

RCTRIM0 – High Speed Internal Oscillator Trim 0 (TA Protected)

Register	SFR Address	Reset Value
RCTRIM0	84H, Page 0, TA protected	XXXX_XXXXb

7	6	5	4	3	2	1	0
HIRCTRIM[8:1]							
R/W							

Bit	Name	Description
[7:0]	HIRCTRIM[8:1]	High Speed Internal Oscillator Trim Value High Byte

RCTRIM1 – High Speed Internal Oscillator Trim 1 (TA Protected)

Register	SFR Address	Reset Value
RCTRIM1	85H, Page 0, TA protected	XXXX_XXXXb

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	HIRCTRIM[0]
-	-	-	-	-	-	-	R/W

Bit	Name	Description
[0]	HIRCTRIM[0]	High Speed Internal Oscillator Trim Value lowest bit

LIRCTRIM – Low Speed Internal Oscillator Trim (TA Protected)

Register	SFR Address	Reset Value
LIRCTRIM	84H, Page 1	XXXX_XXXXb

7	6	5	4	3	2	1	0
LIRCTRIM[7:0]							
R/W							

Bit	Name	Description
[7:0]	LIRCTRIM[7:0]	Low Speed Internal Oscillator Trim Value

XLTCON – XLT Clock Control (TA Protected)

Register	SFR Address	Reset Value
XLTCON	85H, Page 1, TA protected	0111_0111b

7	6	5	4	3	2	1	0
HSCH	**HXSG[2:0]**			**-**	**-**	**LXSG[1:0]**	
R/W	R/W			-	-	R/W	

Bit	Name	Description
[7]	**HSCH**	**HXT Schmitt Trigger Select** 0 = disable 1 = enable
[6:4]	**HXSG[2:0]**	**HXT Gain Value Select** 000 = L0 mode (smallest value) 001 = L1 mode 010 = L2 mode 011 = L3 mode 100 = L4 mode 101 = L5 mode 110 = L6 mode 111 = L7 mode (largest value)
[3:2]	**-**	Reserved
[1:0]	**LXSG[1:0]**	**LXT Gain Value Select** 00 = L0 mode (smallest value) 01 = L1 mode 10 = L2 mode 11 = L3 mode (largest value)

RWKL – Self Wake-up Timer Reload Low Byte

Register	SFR Address	Reset Value
RWKL	86H, Page 0	0000_0000b

7	6	5	4	3	2	1	0
RWK[7:0]							
R/W							

Bit	Name	Description
[7:0]	**RWK[7:0]**	**WKT Reload Byte** It holds the 16-bit reload value of WKT. Note that RWK should not be FFFFH if the pre-scale is 1/1 for implement limitation.

CWKL – Self Wake-up Timer Current Count Value Low Byte

Register	SFR Address	Reset Value
CWKL	86H, Page 1	0000_0000b

7	6	5	4	3	2	1	0
CWK[7:0]							
R							

Bit	Name	Description
[7:0]	**CWK[7:0]**	**WKT Current Count Value Low Byte** It is store value of WKT current count.

PCON – Power Contro

Register	SFR Address	Reset Value
PCON	87H, All pages	POR: 0001_0000b Others: 000U _0000b

7	6	5	4	3	2	1	0
SMOD	**SMOD0**	**LPR**	**POF**	**GF1**	**GF0**	**PD**	**IDL**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**SMOD**	**Serial Port 0 Double Baud Rate Enable** Setting this bit doubles the serial port baud rate when UART0 is in Mode 2 or when Timer 1 overflow is used as the baud rate source of UART0 Mode 1 or 3. See Table 6.10-1 Serial Port 0 Mode / baud rate Description for details.
[6]	**SMOD0**	**Serial Port 0 Framing Error Flag Access Enable** 0 = SCON.7 accesses to SM0 bit. 1 = SCON.7 accesses to FE bit.
[5]	**LPR**	**Low Power Run Mode** 0 = disable 1 = enable **Note:** If PD = 1 and LPR = 1 at the same time, LPR is invalid, CPU will enter Power-down mode.
[4]	**POF**	**Power-on Reset Flag** This bit will be set as 1 after a power-on reset. It indicates a cold reset, a power-on reset complete. This bit remains its value after any other resets. This flag is recommended to be cleared via software.
[3]	**GF1**	**General Purpose Flag 1** The general purpose flag that can be set or cleared by user via software.
[2]	**GF0**	**General Purpose Flag 0** The general purpose flag that can be set or cleared by user via software.
[1]	**PD**	**Power-Down Mode** Setting this bit puts CPU into Power-down mode. Under this mode, both CPU and peripheral clocks stop and Program Counter (PC) suspends. It provides the lowest power consumption. After CPU is woken up from Power-down, this bit will be automatically cleared via hardware and the program continue executing the interrupt service routine (ISR) of the very interrupt source that woke the system up before. After return from the ISR, the device continues execution at the instruction, which follows the instruction that put the system into Power-down mode. Note that If IDL bit and PD bit are set simultaneously, CPU will enter Power-down mode. Then it does not go to Idle mode after exiting Power-down.

Bit	Name	Description
[0]	**IDL**	**Idle Mode** Setting this bit puts CPU into Idle mode. Under this mode, the CPU clock stops and Program Counter (PC) suspends but all peripherals keep activated. After CPU is woken up from Idle, this bit will be automatically cleared via hardware and the program continue executing the ISR of the very interrupt source that woke the system up before. After return from the ISR, the device continues execution at the instruction which follows the instruction that put the system into Idle mode.

TCON – Timer 0 and 1 Control

Register	SFR Address	Reset Value
TCON	88H, All pages, Bit-addressable	0000_0000b

7	6	5	4	3	2	1	0
TF1	**TR1**	**TF0**	**TR0**	**IE1**	**IT1**	**IE0**	**IT0**
R/W	R/W	R/W	R/W	R (level) R/W (edge)	R/W	R (level) R/W (edge)	R/W

Bit	Name	Description
[7]	**TF1**	**Timer 1 Overflow Flag** This bit is set when Timer 1 overflows. It is automatically cleared by hardware when the program executes the Timer 1 interrupt service routine. This bit can be set or cleared by software.
[6]	**TR1**	**Timer 1 Run Control** 0 = Timer 1 Disabled. Clearing this bit will halt Timer 1 and the current count will be preserved in TH1 and TL1. 1 = Timer 1 Enabled.
[5]	**TF0**	**Timer 0 Overflow Flag** This bit is set when Timer 0 overflows. It is automatically cleared via hardware when the program executes the Timer 0 interrupt service routine. This bit can be set or cleared by software.
[4]	**TR0**	**Timer 0 Run Control** 0 = Timer 0 Disabled. Clearing this bit will halt Timer 0 and the current count will be preserved in TH0 and TL0. 1 = Timer 0 Enabled.
[3]	**IE1**	**External Interrupt 1 Edge Flag** If IT1 = 1 (falling edge trigger), this flag will be set by hardware when a falling edge is detected. It remain set until cleared via software or cleared by hardware in the beginning of its interrupt service routine. If IT1 = 0 (low level trigger), this flag follows the inverse of the INT1 input signal's logic level. Software cannot control it.
[2]	**IT1**	**External Interrupt 1 Type Select** This bit selects by which type that INT1 is triggered. 0 = INT1 is low level triggered. 1 = INT1 is falling edge triggered.
[1]	**IE0**	**External Interrupt 0 Edge Flag** If IT0 = 1 (falling edge trigger), this flag will be set by hardware when a falling edge is detected. It remain set until cleared via software or cleared by hardware in the beginning of its interrupt service routine. If IT0 = 0 (low level trigger), this flag follows the inverse of the INT0 input signal's logic level. Software cannot control it.

Bit	Name	Description
[0]	**IT0**	**External Interrupt 0 Type Select** This bit selects by which type that INT0 is triggered. 0 = INT0 is low level triggered. 1 = INT0 is falling edge triggered.

TMOD – Timer 0 and 1 Mode

Register	SFR Address	Reset Value
TMOD	89H, All pages	0000_0000b

7	6	5	4	3	2	1	0
GATE	**C/T**	**M1**	**M0**	**GATE**	**C/T**	**M1**	**M0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**GATE**	**Timer 1 Gate Control** 0 = Timer 1 will clock when TR1 is 1 regardless of INT1 logic level. 1 = Timer 1 will clock only when TR1 is 1 and INT1 is logic 1.
[6]	**C/T**	**Timer 1 Counter/Timer Select** 0 = Timer 1 is incremented by internal system clock. 1 = Timer 1 is incremented by the falling edge of the external pin T1.
[5]	**M1**	**Timer 1 Mode Select** M1 M0 Timer 1 Mode 0 0 Mode 0: 13-bit Timer/Counter 0 1 Mode 1: 16-bit Timer/Counter 1 0 Mode 2: 8-bit Timer/Counter with auto-reload from TH1 1 1 Mode 3: Timer 1 halted
[4]	**M0**	Check with bit 5 description.
[3]	**GATE**	**Timer 0 Gate Control** 0 = Timer 0 will clock when TR0 is 1 regardless of INT0 logic level. 1 = Timer 0 will clock only when TR0 is 1 and INT0 is logic 1.
[2]	**C/T**	**Timer 0 Counter/Timer Select** 0 = Timer 0 is incremented by internal system clock. 1 = Timer 0 is incremented by the falling edge of the external pin T0.
[1]	**M1**	**Timer 0 Mode Select** M1 M0 Timer 0 Mode 0 0 Mode 0: 13-bit Timer/Counter 0 1 Mode 1: 16-bit Timer/Counter 1 0 Mode 2: 8-bit Timer/Counter with auto-reload from TH0 1 1 Mode 3: TL0 as a 8-bit Timer/Counter and TH0 as a 8-bit Timer
[0]	**M0**	Check with bit 1 description

TL0 – Timer 0 Low Byt

Register	SFR Address	Reset Value
TL0	8AH, Page 0	0000_0000b

7	6	5	4	3	2	1	0
TL0[7:0]							
R/W							

Bit	Name	Description
[7:0]	**TL0[7:0]**	**Timer 0 Low Byte** The TL0 register is the low byte of the 16-bit counting register of Timer 0.

PnDW – Port n Pull-down Resister Contro

Register	SFR Address	Reset Value
P0DW	8AH, Page 1	0000_0000 b
P1DW	8BH, Page 1	0000_0000 b
P2DW	8CH, Page 1	0000_0000 b
P3DW	8DH, Page 1	0000_0000 b
P4DW	8EH, Page 1	0000_0000 b
P5DW	8FH, Page 1	0000_0000 b
P6DW	8FH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PnDW.7	**PnDW.6**	**PnDW.5**	**PnDW.4**	**PnDW.3**	**PnDW.2**	**PnDW.1**	**PnDW.0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:0]	**PnDW[7:0]**	**Pn.x Pull-Down Enable** 0 = Pn.x pull-down Disabled. 1 = Pn.x pull-down Enabled.

TL1 – Timer 1 Low Byte

Register	SFR Address	Reset Value
TL1	8BH, Page 0	0000_0000b

7	6	5	4	3	2	1	0
TL1[7:0]							
R/W							

Bit	Name	Description
[7:0]	**TL1[7:0]**	**Timer 1 Low Byte** The TL1 register is the low byte of the 16-bit counting register of Timer 1.

TH0 – Timer 0 High Byte

Register	SFR Address	Reset Value
TH0	8CH, Page 0	0000_0000b

7	6	5	4	3	2	1	0
TH0[7:0]							
R/W							

Bit	Name	Description
[7:0]	**TH0[7:0]**	**Timer 0 High Byte** The TH0 register is the high byte of the 16-bit counting register of Timer 0.

TH1 – Timer 1 High Byte

Register	SFR Address	Reset Value
TH1	8DH, Page 0	0000_0000b

7	6	5	4	3	2	1	0
TH1[7:0]							
R/W							

Bit	Name	Description
[7:0]	**TH1[7:0]**	**Timer 1 High Byte** The TH1 register is the high byte of the 16-bit counting register of Timer 1.

CKCON – Clock Control

Register	SFR Address	Reset Value
CKCON	8EH, Page 0	1000_0000b

7	6	5	4	3	2	1	0
FASTWK	PWMCKS	T1OE	T1M	T0M	T0OE	CLOEN	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-

Bit	Name	Description
[7]	FASTWK	**Fast Wakeup Enable** 0 = Faster Wakeup Disabled, when system wakeup from Power-down mode, HIRC clock stable time is about 10us. 1 = Faster Wakeup Enabled, when system wakeup from Power-down mode, HIRC clock stable time is about 3us.
[6]	PWMCKS	**PWM Clock Source Select** 0 = The clock source of PWM is the system clock FSYS. 1 = The clock source of PWM is the overflow of Timer 1.
[5]	T1OE	**Timer 1 Output Enable** 0 = Timer 1 output Disabled. 1 = Timer 1 output Enabled from T1 pin. Note that Timer 1 output should be enabled only when operating in its “Timer” mode.
[4]	T1M	**Timer 1 Clock Mode Select** 0 = The clock source of Timer 1 is the system clock divided by 12. It maintains standard 8051 compatibility. 1 = The clock source of Timer 1 is direct the system clock.
[3]	T0M	**Timer 0 Clock Mode Select** 0 = The clock source of Timer 0 is the system clock divided by 12. It maintains standard 8051 compatibility. 1 = The clock source of Timer 0 is direct the system clock.
[2]	T0OE	**Timer 0 Output Enable** 0 = Timer 0 output Disabled. 1 = Timer 0 output Enabled from T0 pin. Note that Timer 0 output should be enabled only when operating in its “Timer” mode.
[1]	CLOEN	**System Clock Output Enable** 0 = System clock output Disabled. 1 = System clock output Enabled from CLO pin. Once system clock output was enabled, only POR/BOD reset can disable it.
[0]	-	Reserved.

WKCON – Self Wake-up Timer Control

Register	SFR Address	Reset Value
WKCON	8FH, Page 0	0000_0000b

7	6	5	4	3	2	1	0
-	-	WKTCK	WKTF	WKTR	WKPS[2:0]		
-	-	R/W	R/W	R/W	R/W		

Bit	Name	Description
[7:6]	-	Reserved
[5]	WKTCK	**WKT Clock Source** This bit is set WKT clock source select bit. 0 = LIRC 1 = LXT
[4]	WKTF	**WKT Overflow Flag** This bit is set when WKT overflows. If the WKT interrupt and the global interrupt are enabled, setting this bit will make CPU execute WKT interrupt service routine. This bit is not automatically cleared via hardware and should be cleared via software.
[3]	WKTR	**WKT Run Control** 0 = WKT is halted. 1 = WKT starts running. Note that the reload register RWK can only be written when WKT is halted (WKTR bit is 0). If WKT is written while WKTR is 1, result is unpredictable.
[2:0]	WKPS[2:0]	**WKT Pre-Scalar** These bits determine the pre-scale of WKT clock. 000 = 1/1. 001 = 1/4. 010 = 1/16. 011 = 1/64. 100 = 1/256. 101 = 1/512. 110 = 1/1024. 111 = 1/2048.

SFRS – SFR Page Selection

Register	SFR Address	Reset Value
SFRS	91H, All pages	0000_0000b

7	6	5	4	3	2	1	0
-	-	-	-	-	-	SFRPAGE[1:0]	
-	-	-	-	-	-	R/W	

Bit	Name	Description
[1:0]	**SFRPAGE[1:0]**	**SFR Page Select** 00 = Instructions access SFR Page 0. 01 = Instructions access SFR Page 1. 10 = Instructions access SFR page 2. 11 = Instructions access SFR page 3.

DMAnCR – PDMAn Control Register

Register	SFR Address	Reset Value
DMA0CR0	92H, Page 0	0000_0000 b
DMA1CR0	EBH, Page 0	0000_0000 b
DMA2CR0	B3H, Page 2	0000_0000 b
DMA3CR0	ABH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PSSEL[3:0]				HIE	FIE	RUN	EN
R/W				R/W	R/W	R/W	R/W

Bit	Name	Description
[7:4]	**PSSEL[3:0]**	**Peripheral Source Select** 0000 = XRAM to XRAM 0001 = SPI0 RX 0010 = SMC0/UART2 RX. 0011 = SPI1 RX 0100 = Reserved, No peripheral source select 0101 = SPI0 TX 0110 = SMC0/UART2 TX. 0111 = SPI1 TX 1010 = SMC1/UART3 RX. 1110 = SMC1/UART3 TX. The others are reserved, no periperal source selected **Note:** 0001~0011,1010 : peripheral devices to XRAM memory 0101~0111,1110 : XRAM memory to peripheral devices
[3]	**HIE**	**PDMA HALFTransfer Done Interrupt Enable Bit** 0 = Interrupt Disabled when PDMA half transfer is done. 1 = Interrupt Enabled when PDMA half transfer is done.
[2]	**FIE**	**PDMA Full Transfer Done Interrupt Enable Bit** 0 = Interrupt Disabled when PDMA full transfer is done. 1 = Interrupt Enabled when PDMA full transfer is done.
[1]	**RUN**	**Trigger Enable Bit** 0 = No effect. 1 = PDMA data transfer Enabled. **Note 1:** When PDMA transfer completed, this bit will be cleared automatically.
[0]	**EN**	**PDMA Enable Bit** Setting this bit to 1 enables PDMA operation. If this bit is cleared, PDMA will ignore all PDMA request and Reset the internal state machine, pointers and internal buffer. The contents of all Register Description will not be cleared.

PnUP – Port n Pull-up Resister Control

Register	SFR Address	Reset Value
P0UP	92H, Page 1	0000_0000 b
P1UP	93H, Page 1	0000_0000 b
P2UP	94H, Page 1	0000_0000 b
P3UP	95H, Page 1	0000_0000 b
P4UP	96H, Page 1	0000_0000 b
P5UP	97H, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
PnUP.7	**PnUP.6**	**PnUP.5**	**PnUP.4**	**PnUP.3**	**PnUP.2**	**PnUP.1**	**PnUP.0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:0]	**PnUP[7:0]**	**Pn.x Pull-Up Enable** 0 = Pn.x pull-up Disabled. 1 = Pn.x pull-up Enabled.

DMAnMA – PDMA XRAM Base Address Low Byte

Register	SFR Address	Reset Value
DMA0MAL	93H, Page 0	0000_0000 b
DMA1MAL	ECH, Page 0	0000_0000 b
DMA2MAL	B4H, Page 2	0000_0000 b
DMA3MAL	ACH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
MAL[7:0]							
R/W							

Bit	Name	Description
[7:0]	**MAL[7:0]**	**PDMA XRAM Base Address (Low Byte)** The least significant 8 bits of XRAM address to store or read for the peripheral source data; in memory to memory transfer, this register is the source address. XRAM address = {MAH[3:0],MAL[7:0]}

DMAnCNT – PDMA Transfer Count

Register	SFR Address	Reset Value
DMA0CNT	94H, Page 0	0000_0000 b
DMA1CNT	EDH, Page 0	0000_0000 b
DMA2CNT	B5H, Page 2	0000_0000 b
DMA3CNT	ADH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
DMAnCNT[7:0]							
R/W							

Bit	Name	Description
[7:0]	**DMAnCNT[7:0]**	**PDMA Transfer Count** The total transfer count for PDMA request operation. Total transfer count = CNT[7:0] + 1

DMAnCCNT – PDMA Current Transfer Count

Register	SFR Address	Reset Value
DMA0CCNT	95H, Page 0	0000_0000 b
DMA1CCNT	EEH, Page 0	0000_0000 b
DMA2CCNT	B6H, Page 2	0000_0000 b
DMA3CCNT	AEH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
DMAnCCNT[7:0]							
R							

Bit	Name	Description
[7:0]	**DMAnCCNT[7:0]**	**PDMA Current Transfer Count** The current transfer count for PDMA request operation. Current transfer count = CCNT[7:0] **Note:** while DMAnCNT=0xFF (total transfer count = 256) and DMAnCCNT = 0x00 , If PDMA FDONE flag (DMAnTSR[0])=0, that means, 1'st byte data is not complete.If PDMA FDONE flag (DMAnTSR[0])=1, that means, all of data are transferred..

CKSWT – Clock Switch (TA Protected

Register	SFR Address	Reset Value
CKSWT	96H, PAGE 0, TA protected	0011 _0000 b

7	6	5	4	3	2	1	0
HXTST	LXTST	HIRCST	LIRCST	ECLKST	OSC[2:0]		
R	R	R	R	R	W		

Bit	Name	Description
[7]	HXTST	**High Speed External Crystal/Resonator 4 MHz to 24 MHz Status** 0 = High speed external crystal/resonator is not stable or is disabled. 1 = High speed external crystal/resonator is enabled and stable.
[6]	LXTST	**Low Speed External Crystal/Resonator 32.768 kHz Status** 0 = Low speed external crystal/resonator is not stable or is disabled. 1 = Low speed external crystal/resonator is enabled and stable.
[5]	HIRCST	**High-Speed Internal Oscillator 24 MHz Status** 0 = High-speed internal oscillator is not stable or disabled. 1 = High-speed internal oscillator is enabled and stable.
[4]	LIRCST	**Low Speed Internal Oscillator 38.4 kHz Status** 0 = Low speed internal oscillator is not stable or is disabled. 1 = Low speed internal oscillator is enabled and stable.
[3]	ECLKST	**External Clock Input Status** 0 = External clock input is not stable or disabled. 1 = External clock input is enabled and stable.
[2:0]	OSC[2:0]	**Oscillator Selection Bits** This field selects the system clock source. 00x = Internal 24 MHz oscillator. Defaul value accoding to HIRCEN(CKEN.5) enabled. 01x = External oscillator clock source accoding to ECLKEN(CKEN.3) enabled. 10x = Internal 38.4 kHz oscillator according to LIRCEN(CKEN.4) enabled. 110 = External High speed crystal/resonator clock source (4 MHz ~ 24 MHz) accoding to EHXTEN(CKEN.7) enabled. 111 = External Low speed crystal/resonator clock source (32.768 kHz) accoding to ELXTEN(CKEN.6) enabled. Note that this field is write only. The read back value of this field may not correspond to the present system clock source.

CKEN – Clock Enable

Register	SFR Address	Reset Value
CKEN	97H, PAGE 0, TA protected	0011_0100 b

7	6	5	4	3	2	1	0
EHXTEN	ELXTEN	HIRCEN	LIRCEN	ECLKEN	-		CKSWTF
R/W	R/W	R/W	R/W	R/W	-		R

Bit	Name	Description
[7]	**EHXTEN**	**External High-Speed Crystal/Resonator Enable** 1 = High-speed external crystal/resonator 4 MHz to 24 MHz Enabled. 0 = High-speed external crystal/resonator 4 MHz to 24 MHz Disabled, P5.2 and P5.3 work as general purpose I/O or other functions if ECLKEN set to 0.
[6]	**ELXTEN**	**External Low-Speed Crystal/Resonator Enable** 1 = Low-speed external crystal/resonator 32.768 kHz Enabled. 0 = Low-speed external crystal/resonator 32.768 kHz Disabled, P5.4 and P5.5 work as general purpose I/O or other functions.
[5]	**HIRCEN**	**High-Speed Internal Oscillator 24 MHz Enable** 0 = The high-speed internal oscillator Disabled. 1 = The high-speed internal oscillator Enabled. Note that once IAP is enabled by setting IAPEN (CHPCON.0), the high-speed internal 24 MHz oscillator will be enabled automatically. The hardware will also set HIRCEN and HIRCST bits. After IAPEN is cleared, HIRCEN and EHRCST resume the original values.
[4]	**LIRCEN**	**Low Speed Internal Oscillator 38.4 kHz Enable** 0 = The low speed internal oscillator Disabled. 1 = The low speed internal oscillator Enabled. Note that when (1)WDT is enabled, (2)WKT is running by the clock source of the internal 38.4 kHz oscillator ,(3) BOD is enabled, or (4)LVR filter is enabled, a write 0 to LIRCEN will be ignored. LIRCEN is always 1 and the internal 38.4 kHz oscillator is always enabled.
[3]	**ECLKEN**	**External Clock Input Enable** 1 = External clock input (XIN , P5.3) Enabled. 0 = External clock input (XIN, P5.3) Disabled, P5.2 and P5.3 work as general purpose I/O or other functions if EHXTEN set to 0.
[2:1]	**-**	Reserved
[0]	**CKSWTF**	**Clock Switch Fault Flag** 0 = The previous system clock source switch was successful. 1 = User tried to switch to an instable or disabled clock source at the previous system clock source switch. If switching to an instable clock source, this bit remains 1 until the clock source is stable and switching is successful.

SCON – Serial Port Control

Register	SFR Address	Reset Value
SCON	98H, All pages, Bit addressable	0000_0000 b

7	6	5	4	3	2	1	0
SM0/FE	**SM1**	**SM2**	**REN**	**TB8**	**RB8**	**TI**	**RI**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**SM0/FE**	**Serial Port Mode Select** SMOD0 (PCON.6) = 0: See Table 6.10-1 Serial Port 0 Mode / baud rate Description for details. SMOD0 (PCON.6) = 1: SM0/FE bit is used as frame error (FE) status flag. It is cleared by software. 0 = Frame error (FE) did not occur. 1 = Frame error (FE) occurred and detected.
[6]	**SM1**	Check with bit 7 description.
[5]	**SM2**	**Multiprocessor Communication Mode Enable** The function of this bit is dependent on the serial port 0 mode. Mode 0: This bit select the baud rate between FSYS/12 and FSYS/2. 0 = The clock runs at FSYS/12 baud rate. It maintains standard 8051compatibility. 1 = The clock runs at FSYS/2 baud rate for faster serial communication. Mode 1: This bit checks valid stop bit. 0 = Reception is always valid no matter the logic level of stop bit. 1 = Reception is valid only when the received stop bit is logic 1 and the received data matches “Given” or “Broadcast” address. Mode 2 or 3: For multiprocessor communication. 0 = Reception is always valid no matter the logic level of the 9th bit. 1 = Reception is valid only when the received 9th bit is logic 1 and the received data matches “Given” or “Broadcast” address.
[4]	**REN**	**Receiving Enable** 0 = Serial port 0 reception Disabled. 1 = Serial port 0 reception Enabled in Mode 1,2, or 3. In Mode 0, reception is initiated by the condition REN = 1 and RI = 0.
[3]	**TB8**	**9th Transmitted Bit** This bit defines the state of the 9th transmission bit in serial port 0 Mode 2 or 3. It is not used in Mode 0 or 1.

Bit	Name	Description
[2]	**RB8**	**9th Received Bit** The bit identifies the logic level of the 9th received bit in serial port 0 Mode 2 or 3. In Mode 1, RB8 is the logic level of the received stop bit. SM2 bit as logic 1 has restriction for exception. RB8 is not used in Mode 0.
[1]	**TI**	**Transmission Interrupt Flag** This flag is set by hardware when a data frame has been transmitted by the serial port 0 after the 8th bit in Mode 0 or the last data bit in other modes. When the serial port 0 interrupt is enabled, setting this bit causes the CPU to execute the serial port 0 interrupt service routine. This bit should be cleared manually via software.
[0]	**RI**	**Receiving Interrupt Flag** This flag is set via hardware when a data frame has been received by the serial port 0 after the 8th bit in Mode 0 or after sampling the stop bit in Mode 1, 2, or 3. SM2 bit as logic 1 has restriction for exception. When the serial port 0 interrupt is enabled, setting this bit causes the CPU to execute to the serial port 0 interrupt service routine. This bit should be cleared manually via software.

SBUF – Serial Port 0 Data Buffer

Register	SFR Address	Reset Value
SBUF	99H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
SBUF[7:0]							
R/W							

Bit	Name	Description
[7:0]	**SBUF[7:0]**	**Serial Port 0 Data Buffer** This byte actually consists two separate registers. One is the receiving resister, and the other is the transmitting buffer. When data is moved to SBUF, it goes to the transmitting buffer and is shifted for serial transmission. When data is moved from SBUF, it comes from the receiving register. The transmission is initiated through giving data to SBUF.

PnS – Port n Schmitt Triggered Input

Register	SFR Address	Reset Value
P0S	99H, Page 1	0000_0000 b
P1S	9BH, Page 1	0000_0000 b
P2S	9DH, Page 1	0000_0000 b
P3S	ACH, Page 1	0000_0000 b
P4S	BBH, Page 1	0000_0000 b
P5S	BFH, Page 1	0000_0000 b
P6S	96H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PnS.7	**PnS.6**	**PnS.5**	**PnS.4**	**PnS.3**	**PnS.2**	**PnS.1**	**PnS.0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:0]	**PnS[7:0]**	**Pn Schmitt Triggered Input** 0 = TTL level input of Pn.x. 1 = Schmitt triggered input of Pn.x.

WDCON – Watchdog Timer Control (TA Protected)

Register	SFR Address	Reset Value
WDCON	AAH, Page 0, TA protected	POR 0000_0001 b WDT 0000_000U b Others 0000_000U b

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	**WDPS[3]**
							R/W

Bit	Name	Description
[7:1]	-	Reserved
[0]	**WDPS[3]**	**WDT Clock Pre-Scalar Select** These bits determine the pre-scale of WDT clock from 1/1 through 1/2048. SeeTable 6.7-1 Watchdog Timer-out Interval Under Different Pre-scalars The default is the maximum pre-scale value.
Note: WDPS[3:0] are all set after power-on reset and keep unchanged after any reset other than power-on reset.		

SBUF1 – Serial Port 1 Data Buffer

Register	SFR Address	Reset Value
SBUF1	9AH, Page 0	0000 _0000 b

7	6	5	4	3	2	1	0
SBUF1[7:0]							
R/W							

Bit	Name	Description
[7:0]	**SBUF1[7:0]**	**Serial Port 1 Data Buffer** This byte actually consists two separate registers. One is the receiving resister, and the other is the transmitting buffer. When data is moved to SBUF1, it goes to the transmitting buffer and is shifted for serial transmission. When data is moved from SBUF1, it comes from the receiving register. The transmission is initiated through giving data to SBUF1.

PnSR –Port n Slew Rate Control

Register	SFR Address	Reset Value
P0SR	9AH, Page 1	0000_0000 b
P1SR	9CH, Page 1	0000_0000 b
P2SR	9EH, Page 1	0000_0000 b
P3SR	ADH, Page 1	0000_0000 b
P4SR	BCH, Page 1	0000_0000 b
P5SR	AEH, Page 1	0000_0000 b
P6SR	8EH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PnSR.7	**PnSR.6**	**PnSR.5**	**PnSR.4**	**PnSR.3**	**PnSR.2**	**PnSR.1**	**PnSR.0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:0]	**PnSR[7:0]**	**Pn.x Slew Rate** 0 = Pn.x normal output slew rate. 1 = Pn.x high-speed output slew rate.

EIE0 – Extensive Interrupt Enable

Register	SFR Address	Reset Value
EIE0	9BH, Page 0	0000 _0000 b

7	6	5	4	3	2	1	0
ET2	**ESPI0**	**EFB0**	**EWDT**	**EPWM0**	**ECAP**	**EPI**	**EI2C0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**ET2**	**Enable Timer 2 Interrupt** 0 = Timer 2 interrupt Disabled. 1 = Timer 2 interrupt Enable. When interrupt generated, TF2 (T2CON.7) set 1
[6]	**ESPI0**	**Enable SPI Interrupt** 0 = SPI interrupt Disabled. 1 = SPI interrupt Enable. When interrupt generated SPIF (SPInSR.7), SPIOVF (SPInSR.5), or MODF (SPInSR.4) set 1 .
[5]	**EFB0**	**Enable Fault Brake Interrupt** 0 = Fault Brake interrupt Disabled. 1 = Fault Brake interrupt Enable. When interrupt generated FBF (PWM0FBD.7) set 1.
[4]	**EWDT**	**Enable WDT Interrupt** 0 = WDT interrupt Disabled. 1 = WDT interrupt Enable. When interrupt generated WDTF (WDCON.5) set 1.
[3]	**EPWM0**	**Enable PWM0 Interrupt** 0 = PWM interrupt Disabled. 1 = PWM interrupt Enable. When interrupt generated PWMF (PWMnCON0.5) set 1.
[2]	**ECAP**	**Enable Input Capture Interrupt** 0 = Input capture interrupt Disabled. 1 = Input capture interrupt Enable. When interrupt generated CAPF[2:0] (CAPCON0[2:0]) set 1.
[1]	**EPI**	**Enable Pin Interrupt** 0 = Pin interrupt Disabled. 1 = Pin interrupt Enable. When interrupt generated PIF related bit set 1.
[0]	**EI2C0**	**Enable I^2C0 Interrupt** 0 = I^2C interrupt Disabled. 1 = I^2C interrupt Enable. When interrupt generated SI (I2C0CON.3) or I2TOF (I2C0TOC.0) set 1.

EIE1 – Extensive Interrupt Enable 1

Register	SFR Address	Reset Value
EIE1	9CH, Page 0	0000 _0000 b

7	6	5	4	3	2	1	0
-	EPWM123	EI2C1	ESPI1	EHFI	EWKT	ET3	ES1
-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	-	-
[6]	EPWM123	**Enable PWM123 Interrupt** 0 = PWM1/2/3 interrupt Disabled. 1 = PWM1/2/3 interrupt Enable. When interrupt generated PWMF (PWM1CON0.5) set 1.
[5]	EI2C1	**Enable I2C1 Interrupt** 0 = I^2C1 interrupt Disabled. 1 = I^2C1 interrupt Enable. When interrupt generated SI (I2C1CON.3) or I2TOF (I2C1TOC.0) set 1.
[4]	ESPI1	**Enable SPI1 Interrupt** 0 = SPI1 interrupt Disabled. 1 = SPI1 interrupt Enable. When interrupt generated SPIF (SP2SR.7), MODF (SP2SR.4) or SPIOVF (SP2SR.5) set 1
[3]	EHFI	**Enable Hard Fault Interrupt** 0 = hard fault interrupt Disabled and hard fault reset is Enabled 1 = hard fault interrupt Enable. When interrupt generated HFIF (AUXR0.4) set 1.
[2]	EWKT	**Enable WKT Interrupt** 0 = WKT interrupt Disabled. 1 = WKT interrupt Enable. When interrupt generated WKTF (WKCON.4) set 1.
[1]	ET3	**Enable Timer 3 Interrupt** 0 = Timer 3 interrupt Disabled. 1 = Timer 3interrupt Enable. When interrupt generated TF3 (T3CON.4) set 1.
[0]	ES1	**Enable Serial Port 1 Interrupt** 0 = Serial port 1 interrupt Disabled. 1 = Serial port 1 interrupt Enable. When interrupt generated TI_1 (S1CON.1) or RI_1 (S1CON.0) set 1.

RSR – Reset Flag Register

Register	SFR Address	Reset Value
RSR	9DH, Page 0	1101_0000 b

7	6	5	4	3	2	1	0
LVRF	**PORF**	**HFRF**	**POF**	**RSTPINF**	**BORF**	**WDTRF**	**SWRF**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**LVRF**	**LVR Reset Flag** 1: LVR Reset Flag is active 0: LVR Reset Flag is inactive Write 0 to clear this bit
[6]	**PORF**	**POR Reset Flag** 1: POR15 Reset Flag is active 0: POR15 Reset Flag is inactive Write 0 to clear this bit
[5]	**HFRF**	**mirrored From AUXR0.5** Clear this bit by write AUXR0.5=0 or RSR.5=0
[4]	**POF**	**mirrored From PCON.4** Clear this bit by write PCON.4=0 or RSR.4=0
[3]	**RSTPINF**	**mirrored From AUXR0.6** Clear this bit by write AUXR0.6=0 or RSR.3=0
[2]	**BORF**	**mirrored From BODCON0.1** Clear this bit by write BODCON0.1=0 or RSR.2=0
[1]	**WDTRF**	**mirrored From WDCON.3** Clear this bit by write WDCON.3=0 or RSR.1=0
[0]	**SWRF**	**Mirrored From AUXR0.7** Clear this bit by write AUXR0.7=0 or RSR.0=0

CHPCON – Chip Control (TA Protected)

Register	SFR Address	Reset Value
CHPCON	9FH, Page 0, TA protected	Software 0000_00U0 b Others 0000_00C0 b

7	6	5	4	3	2	1	0
SWRST	**IAPFF**	**-**	**-**	**-**	**-**	**BS**	**IAPEN**
W	R/W	-	-	-	-	R/W	R/W

Bit	Name	Description
[7]	**SWRST**	**Software Reset** To set this bit as logic 1 will cause a software reset. It will automatically be cleared via hardware after reset is finished.
[6]	**IAPFF**	**IAP Fault Flag** The hardware will set this bit after IAPGO (IAPTRG.0) is set if any of the following condition is met: (1) The accessing address is oversize. (2) IAPCN command is invalid. (3) IAP erases or programs updating un-enabled block. (4) IAP erasing or programming operates under VBOD while BOIAP (CONFIG2.5) remains un-programmed 1 with BODEN (BODCON0.7) as 1 and BORST (BODCON0.2) as 0. This bit should be cleared via software.
[5:2]	**-**	Reserved
[1]	**BS**	**Boot Select** This bit defines from which block that MCU re-boots after all resets. 0 = MCU will re-boot from APROM after all resets. 1 = MCU will re-boot from LDROM after all resets.
[0]	**IAPEN**	**IAP Enable** 0 = IAP function Disabled. 1 = IAP function Enabled. Once enabling IAP function, the HIRC will be turned on for timing control. To clear IAPEN should always be the last instruction after IAP operation to stop internal oscillator if reducing power consumption is concerned.

ADCCON0 – ADC Control Register 0

Register	SFR Address	Reset Value
ADCCON0	A1H, Page 0	0000_0000b

7	6	5	4	3	2	1	0
ADCF	ADCS	ETGSEL1	ETGSEL0	ADCHS3	ADCHS2	ADCHS1	ADCHS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**ADCF**	**ADC Flag** This flag is set when an A/D conversion is completed in single sampling mode, final sampling complete in continue sampling mode or comparing hit if result comparator is enabled. The ADC result can be read. While this flag is 1, ADC cannot start a new converting. This bit is cleared by software.
[6]	**ADCS**	**A/D Converting Software Start Trigger** Setting this bit 1 triggers an A/D conversion. This bit remains logic 1 during A/D converting time and is automatically cleared via hardware right after conversion complete. The meaning of writing and reading ADCS bit is different. Writing: 0 = No effect. 1 = Start an A/D converting. Reading: 0 = ADC is in idle state. 1 = ADC is busy in converting.
[5:4]	**ETGSEL[1:0]**	**External Trigger Source Select** When ADCEX (ADCCON1.1) is set, these bits select which pin output triggers ADC conversion. 00 = PWM0CH0. 01 = PWM0CH2. 10 = PWM0CH4. 11 = STADC pin.

Bit	Name	Description
[3:0]	**ADCHS[3:0]**	**A/D Converting Channel Select** This filed selects the activating analog input source of ADC. If ADCEN is 0, all inputs are disconnected. 0000 = ADC_CH0. 0001 = ADC_CH1. 0010 = ADC_CH2. 0011 = ADC_CH3. 0100 = ADC_CH4. 0101 = ADC_CH5. 0110 = ADC_CH6. 0111 = ADC_CH7. 1000 = VBG (Internal band-gap voltage 1.22V). 1001 = VTEMP. (Temperature Sensor). 1010 = ADC_CH10. 1011 = ADC_CH11. 1100 = ADC_CH12. 1101 = ADC_CH13. 1110 = ADC_CH14. 1111 = ADC_CH15.

PIPSn – Pin Interrupt Control

Register	SFR Address	Reset Value
PIPS0	A1H, Page 1	0000_0000 b
PIPS1	A2H, Page 1	0000_0000 b
PIPS2	A3H, Page 1	0000_0000 b
PIPS3	A4H, Page 1	0000_0000 b
PIPS4	A5H, Page 1	0000_0000 b
PIPS5	A6H, Page 1	0000_0000 b
PIPS6	A7H, Page 1	0000_0000 b
PIPS7	AFH, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
-	**PSEL[2:0]**			**-**	**BSEL[2:0]**		
-	R/W			-	R/W		

Bit	Name	Description
[7]	**-**	Reserved
[6:4]	**PSEL[2:0]**	**Pin Interrupt Channel Port Select** 000 = P0 Port. 001 = P1 Port. 010 = P2 Port. 011 = P3 Port. 100 = P4 Port. 101 = P5 Port. 110 = P6 Port. 111 = Reserved.
[3]	**-**	Reserved
[2:0]	**BSEL[2:0]**	**Pin Interrupt Channel Bit Select** 000 = Pn.0. 001 = Pn.1 010 =.Pn.2 011 = Pn.3. 100 = Pn.4. 101 = Pn.5. 110 = Pn.6. 111 = Pn.7. n is the PORT number, which is selected by PSEL[2:0].

AUXR0 – Auxiliary Register 0

Register	SFR Address	Reset Value
AUXR0	A2H, Page 0	POR: 0000 0000b, Software reset: 1U00 0000b, nRESET pin: U100 0000b, Hard fault: UU10 0000b Others: UUU0 0000b

7	6	5	4	3	2	1	0
SWRF	**RSTPINF**	**HFRF**	**HFIF**	**GF2**	**-**	**0**	**DPS**
R/W	R/W	R/W	R/W	R/W	-	R	R/W

Bit	Name	Description
[7]	**SWRF**	**Software Reset Flag** When the MCU is reset via software reset, this bit will be set via hardware. It is recommended that the flag be cleared via software.
[6]	**RSTPINF**	**External Reset Flag** When the MCU is reset by the external reset, this bit will be set via hardware. It is recommended that the flag be cleared via software.
[5]	**HFRF**	**Hard Fault Reset Flag** Once CPU fetches instruction address over Flash size while EHFI (EIE1.4)=0, MCU will reset and this bit will be set via hardware. It is recommended that the flag be cleared via software. **Note:** If MCU run in OCD debug mode and OCDEN = 0, Hard fault reset will disable. Only HFRF flag be asserted.
[4]	**HFIF**	**Hard Fault Interrupt Flag** Once CPU fetches instruction address over Flash size while EHFI (EIE1.4)=1. MCU will be interrupt and this bit will be set via hardware. It is recommended that the flag be cleared via software.
[3]	**GF2**	**General Purpose Flag 2** The general purpose flag that can be set or cleared by the user via software.
[2]	**-**	Reserved
[1]	**0**	Reserved This bit is always read as 0.
[0]	**DPS**	**Data Pointer Select** 0 = Data pointer 0 (DPTR) is active by default. 1 = Data pointer 1 (DPTR1) is active. After DPS switches the activated data pointer, the previous inactivated data pointer remains its original value unchanged.

BODCON0 – Brown-out Detection Control 0 (TA Protected

Register	SFR Address	Reset Value
BODCON0	A3H, Page 0, TA protected	POR,CCCC XC0X b BOD, UUUU XU1X b Others,UUUU XUUX b

7	6	5	4	3	2	1	0
BODEN	**BOV[2:0]**			**BOF**	**BORST**	**BORF**	**BOS**
R/W	R/W			R/W	R/W	R/W	R

Bit	Name	Description
[7]	**BODEN**	**Brown-Out Detection Enable** 0 = Brown-out detection circuit ON. 1 = Brown-out detection circuit OFF. Note that BOD output is not available until 2~3 LIRC clocks after enabling.
[6:4]	**BOV[2:0]**	**CONFIG Brown-Out Voltage Select** 111 = VBOD is 1.8V. 110 = VBOD is 1.8V. 101 = VBOD is 2.0V. 100 = VBOD is 2.4V. 011 = VBOD is 2.7V. 010 = VBOD is 3.0V. Following setting value only for ML51 32KB Flash series 001 = VBOD is 3.7V. 000 = VBOD is 4.4V.
[3]	**BOF**	**Brown-Out Interrupt Flag** This flag will be set as logic 1 via hardware after a V_{DD} dropping below or rising above VBOD event occurs. If both EBOD (I.E.5) and EA (I.E.7) are set, a brown-out interrupt requirement will be generated. This bit should be cleared via software.
[2]	**BORST**	**Brown-Out Reset Enable** This bit decides whether a brown-out reset is caused by a power drop below VBOD. 0 = Brown-out reset when V_{DD} drops below VBOD Disabled. 1 = Brown-out reset when V_{DD} drops below VBOD Enabled.
[1]	**BORF**	**Brown-Out Reset Flag** When the MCU is reset by brown-out event, this bit will be set via hardware. This flag is recommended to be cleared via software.

Bit	Name	Description
[0]	**BOS**	**Brown-Out Status** This bit indicates the V_{DD} voltage level comparing with VBOD while BOD circuit is enabled. It keeps 0 if BOD is not enabled. 0 = V_{DD} voltage level is higher than VBOD or BOD is disabled. 1 = V_{DD} voltage level is lower than VBOD. Note that this bit is read-only.

Note:

1. BODEN, BOV[2:0], and BORST are initialized by being directly loaded from CONFIG2 [6:4], and after all resets.
2. BOF reset value depends on different setting of CONFIG2 and V_{DD} voltage level.

IAPTRG – IAP Trigger (TA Protected)

Register	SFR Address	Reset Value
IAPTRG	A4H, Page 0, TA protected	0000 _0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	IAPGO
-	-	-	-	-	-	-	W

Bit	Name	Description
[7:1]	-	Reserved
[0]	IAPGO	**IAP Go** IAP begins by setting this bit as logic 1. After this instruction, the CPU holds the Program Counter (PC) and the IAP hardware automation takes over to control the progress. After IAP action completed, the Program Counter continues to run the following instruction. The IAPGO bit will be automatically cleared and always read as logic 0. Before triggering an IAP action, interrupts (if enabled) should be temporary disabled for hardware limitation. The program process should follows below.

IAPUEN – IAP Updating Enable (TA Protected)

Register	SFR Address	Reset Value
IAPUEN	A5H, Page 0, TA protected	0000 _0000 b

7	6	5	4	3	2	1	0
-	-	-	SPMEN	SPUEN	CFUEN	LDUEN	APUEN
-	-	-	R/WFV	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:5]	-	Reserved
[4]	SPMEN	**SPROM Memory Space Mapping Enable** 0 = CPU memory address 0xff80~0xffff is mapping to APROM memory 1 = CPU memory address 0xff80~0xffff is mapping to SPROM memory
[3]	SPUEN	**SPROM Memory Space Updated Enable(TA Protected)** 0 = Inhibit erasing or programming SPRO Mbytes by IAP 1 = Allow erasing or programming SPRO Mbytes by IAP.
[2]	CFUEN	**CONFIG Bytes Updated Enable** 0 = Inhibit erasing or programming CONFIG bytes by IAP. 1 = Allow erasing or programming CONFIG bytes by IAP.
[1]	LDUEN	**LDROM Updated Enable** 0 = Inhibit erasing or programming LDROM by IAP. 1 = Allow erasing or programming LDROM by IAP.
[0]	APUEN	**APROM Updated Enable** 0 = Inhibit erasing or programming APROM by IAP. 1 = Allow erasing or programming APROM by IAP.

IAPAL – IAP Address Low Byte

Register	SFR Address	Reset Value
IAPAL	A6H, Page 0	0000 _0000 b

7	6	5	4	3	2	1	0
IAPA[7:0]							
R/W							

Bit	Name	Description
[7:0]	**IAPA[7:0]**	**IAP Address Low Byte** IAPAL contains address IAPA[7:0] for IAP operations.

IAPAH – IAP Address High Byte

Register	SFR Address	Reset Value
IAPAH	A7H, Page 0	0000 _0000 b

7	6	5	4	3	2	1	0
IAPA[15:8]							
R/W							

Bit	Name	Description
[7:0]	**IAPA[15:8]**	**IAP Address High Byte** IAPAH contains address IAPA[15:8] for IAP operations.

IE – Interrupt Enable

Register	SFR Address	Reset Value
IE	A8H, All pages, Bit addressable	0000 _0000 b

7	6	5	4	3	2	1	0
EA	**EADC**	**EBOD**	**ES**	**ET1**	**EX1**	**ET0**	**EX0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**EA**	**Enable All Interrupt** This bit globally enables/disables all interrupts that are individually enabled. 0 = All interrupt sources Disabled. 1 = Each interrupt Enabled depending on its individual mask setting. Individual interrupts will occur if enabled.
[6]	**EADC**	**Enable ADC Interrupt** 0 = ADC interrupt Disabled. 1 = ADC interrupt Enable. When interrupt generated ADCF (ADCCON0.7) set 1.
[5]	**EBOD**	**Enable Brown-Out Interrupt** 0 = Brown-out detection interrupt Disabled. 1 = Brown-out detection interrupt Enable. When interrupt generated BOF (BODCON0.3) set 1.
[4]	**ES**	**Enable Serial Port 0 Interrupt** 0 = Serial port 0 interrupt Disabled. 1 = Serial port 0 interrupt Enable. When interrupt generated TI (SCON.1) or RI (SCON.0) set 1.
[3]	**ET1**	**Enable Timer 1 Interrupt** 0 = Timer 1 interrupt Disabled. 1 = Timer 1 interrupt Enable. When interrupt generated TF1 (TCON.7) set 1.
[2]	**EX1**	**Enable External Interrupt 1** 0 = External interrupt 1 Disabled. 1 = External interrupt 1 interrupt Enable. When interrupt generated INT1 pin set 1.
[1]	**ET0**	**Enable Timer 0 Interrupt** 0 = Timer 0 interrupt Disabled. 1 = Timer 0 interrupt Enable. When interrupt generated TF0 (TCON.5) set 1.
[0]	**EX0**	**Enable External Interrupt 0** 0 = External interrupt 0 Disabled. 1 = External interrupt 0 interrupt Enable. When interrupt generated INT0 pin set 1.

SADDRn – UART Slave Address

Register	SFR Address	Reset Value
SADDR0	A9H, Page 0	0000 _0000 b
SADDR1	BBH, Page 0	0000 _0000 b

7	6	5	4	3	2	1	0
SADDRn[7:0]							
R/W							

Bit	Name	Description
[7:0]	**SADDRn[7:0]**	**Slave n Address** This byte specifies the microcontroller's own slave address for UATR0 multi-processor communication.

VRFCON – Internal V_{REF} Control (TA Protected)

Register	SFR Address	Reset Value
VRFCON	A9H,Page 1, TA protected	0000 _0000 b

7	6	5	4	3	2	1	0
-	VRFSEL[2:0]			-	-	ENLOAD	ENVRF
-	R/W			-	-	R/W	R/W

Bit	Name	Description
[7]	-	Reserved
[6:4]	VRFSEL[2:0]	**Internal V_{REF} Output Voltage Select** This field selects V_{REF} output voltage. 000 = 1.538V , when V_{DD} > 2.0V 001 = 2.048V , when V_{DD} > 2.4V 010 = 2.560V , when V_{DD} > 2.9V 011 = 3.072V , when V_{DD} > 3.4V 100 = 4.096V , when V_{DD} > 4.5V 101 = reserved 110 = reserved 111 = reserved
[3:2]	-	Reserved
[1]	ENLOAD	**Internal V_{REF} Pre-Load Enable** 1 = Internal V_{REF} Pre-load Enabled. 0 = Internal V_{REF} Pre-load Disabled
[0]	ENVRF	**Internal V_{REF} Enable** 1 = Internal V_{REF} Enabled, 0 = Internal V_{REF} Disabled Note that a 1 uF has to add on V_{REF} pin while internal V_{REF} is enabled.

WDCON – Watchdog Timer Control (TA Protected)

Register	SFR Address	Reset Value
WDCON	AAH, Page 0, TA protected	POR 0000_0111 b WDT 0000_1UUU b Others 0000_UUUU b

7	6	5	4	3	2	1	0
WDTR	**WDCLR**	**WDTF**	**WIDPD**	**WDTRF**	**WDPS[2:0]**		
R/W	R/W	R/W	R/W	R/W	R/W		

Bit	Name	Description
[7]	**WDTR**	**WDT Run** This bit is valid only when control bits in WDTEN[3:0] (CONFIG4[7:4]) are all 1. At this time, WDT works as a general purpose timer. 0 = WDT Disabled. 1 = WDT Enabled. The WDT counter starts running.
[6]	**WDCLR**	**WDT Clear** Setting this bit will reset the WDT count to 00H. It puts the counter in a known state and prohibit the system from unpredictable reset. The meaning of writing and reading WDCLR bit is different. Writing: 0 = No effect. 1 = Clearing WDT counter. Reading: 0 = WDT counter is completely cleared. 1 = WDT counter is not yet cleared.
[5]	**WDTF**	**WDT Time-Out Flag** This bit indicates an overflow of WDT counter. This flag should be cleared by software.
[4]	**WIDPD**	**WDT Running in Idle or Power-Down Mode** This bit is valid only when control bits in WDTEN[3:0] (CONFIG4[7:4]) are all 1. It decides whether WDT runs in Idle or Power-down mode when WDT works as a general purpose timer. 0 = WDT stops running during Idle or Power-down mode. 1 = WDT keeps running during Idle or Power-down mode.
[3]	**WDTRF**	**WDT Reset Flag** When the CPU is reset by WDT time-out event, this bit will be set via hardware. This flag is recommended to be cleared via software after reset.
[2:0]	**WDPS[2:0]**	**WDT Clock Pre-Scalar Select** These bits determine the pre-scale of WDT clock from 1/1 through 1/256. Table 6.7-1 Watchdog Timer-out Interval Under Different Pre-scalars The default is the maximum pre-scale value.

Bit	Name	Description
Note: WDTRF will be cleared after power-on reset, be set after WDT reset, and remains unchanged after any other resets WDPS[3:0] are all set after power-on reset and keep unchanged after any reset other than power-on reset.		

VRFTRIM – Internal V_{REF} Trim Select (TA Protected)

Register	SFR Address	Reset Value
VRFTRIM	AAH, Page 1, TA protected	0100_0000b

7	6	5	4	3	2	1	0
-	VRFTRIM[6:0]						
-	R/W						

Address: AAH, Page 1 Reset value: 0100 0000b

Bit	Name	Description
[7]	-	Reserved
[6:0]	VRFTRIM[6:0]	**Internal V_{REF} Trim Select** default=7'b1000000 Output MAX=7'b1111110; Output MIN=7'b0000000 1111111 = Untrimmed nature voltage = Medium voltage (1000000)

BODCON1 – Brown-out Detection Control Byte 1 (TA Protected)

Register	SFR Address	Reset Value
BODCON1	ABH, Page 0, TA protected	POR 0000 0001 b Others 0000 0UUU b

7	6	5	4	3	2	1	0
-	-	-	-	-	LPBOD[1:0]		BODFLT
-	-	-	-	-	R/W		R/W

Bit	Name	Description
[7:3]	-	Reserved
[2:1]	LPBOD[1:0]	**Low Power BOD Enable** 00 = BOD normal mode. BOD circuit is always enabled. 01 = BOD low power mode 1 by turning on BOD circuit every 1.6 ms periodically. 10 = BOD low power mode 2 by turning on BOD circuit every 6.4 ms periodically. 11 = BOD low power mode 3 by turning on BOD circuit every 25.6 ms periodically.
[0]	BODFLT	**BOD Filter Control** BOD has a filter which counts 32 clocks of F_{SYS} to filter the power noise when MCU runs with HIRC, or ECLK as the system clock and BOD does not operates in its low power mode (LPBOD[1:0] = [0, 0]). In other conditions, the filter counts 2 clocks of LIRC. Note that when CPU is halted in Power-down mode. The BOD output is permanently filtered by 2 clocks of LIRC. The BOD filter avoids the power noise to trigger BOD event. This bit controls BOD filter enabled or disabled. 0 = BOD filter Disabled. 1 = BOD filter Enabled. (Power-on reset default value.)

ACMPCR2 – Analog Comparator Control Register 2

Register	SFR Address	Reset Value
ACMPCR2	ABH, Page 1	0000 _0000 b

7	6	5	4	3	2	1	0
SPEED1		POE1	POE0	SPEED0		CRVSSEL	CRVEN
R/W		R/W	R/W	R/W		R/W	R/W

Bit	Name	Description
[7:6]	**SPEED1**	**Analog Comparator 1 Speed Control** 00 = slow speed, propagation delay : 4.5us, 1.2uA (typ.) 01 = slow+ speed, propagation delay : 2.0us, 3uA (typ.) 10 = fast speed, propagation delay : 0.6us, 10uA (typ.) 11 = fast+ speed, propagation delay : 0.2us, 75uA (typ.)
[5]	**POE1**	**Analog Comparator 1 Polarity Output Enable** 0 = ACMP1 output directly. 1 = ACMP1 output inversely.
[4]	**POE0**	**Analog Comparator 0 Polarity Output Enable** 0 = ACMP0 outputs directly. 1 = ACMP0 outputs inversely.
[3:2]	**SPEED0**	**Analog Comparator 0 Speed Control** 00 = slow speed, propagation delay : 4.5us, 1.2uA (typ.) 01 = slow+ speed, propagation delay : 2.0us, 3uA (typ.) 10 = fast speed, propagation delay : 0.6us, 10uA (typ.) 11 = fast+ speed, propagation delay : 0.2us, 75uA (typ.)
[1]	**CRVSSEL**	**CRV Source Voltage Selection** 0 = V_{DD} is selected as CRV source voltage. 1 = The reference voltage (V_{REF}) is selected as CRV source voltage.
[0]	**CRVEN**	**CRV Enable Bit** 0 = CRV Disabled. 1 = CRV Enabled.

EIP2 – Extensive Interrupt Priority 2

Register	SFR Address	Reset Value
EIP2	ACH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
RTC	**PDMA3**	**PDMA2**	**SMC1**	**TK**	**PPWM1**	**PI2C1**	**PACMP**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**RTC**	RTC interrupt priority low bit
[6]	**PDMA3**	PDMA3 interrupt priority low bit
[5]	**PDMA2**	PDMA2 interrupt priority low bit
[4]	**SMC1**	SMC1 interrupt priority low bit
[3]	**TK**	Touch Key interrupt priority low bit
[2]	**PPWM1**	PPWM1 interrupt priority low bit
[1]	**PI2C1**	I^2C interrupt priority low bit
[0]	**PACMP**	ACMP interrupt priority low bit
Note: EIP2 is used in combination with the EIPH2 to determine the priority of each interrupt source. See Table 6.3-5 Interrupt Priority Level Setting for correct interrupt priority configuration.		

EIPH2 – Extensive Interrupt Priority High 2

Register	SFR Address	Reset Value
EIPH2	ADH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
RTCH	**PDMA3H**	**PDMA2H**	**SMC1H**	**TKH**	**PPWM1H**	**PI2C1H**	**PACMPH**
RW	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**RTCH**	RTCH interrupt priority high bit
[6]	**PDMA3H**	PDMA3H interrupt priority high bit
[5]	**PDMA2H**	PDMA2H interrupt priority high bit
[4]	**SMC1H**	SMC1H interrupt priority high bit
[3]	**TKH**	Touch Key interrupt priority high bit
[2]	**PPWM1H**	PPWM1H interrupt priority high bit
[1]	**PI2C1H**	I^2C interrupt priority high bit
[0]	**PACMPH**	ACMP interrupt priority high bit
Note: EIPH2 is used in combination with the EIP2 to determine the priority of each interrupt source. See Table 6.3-5 Interrupt Priority Level Setting for correct interrupt priority configuration.		

IAPFD – IAP Flash Data

Register	SFR Address	Reset Value
IAPFD	AEH, Page 0	0000 _0000 b

7	6	5	4	3	2	1	0
IAPFD[7:0]							
R/W							

Bit	Name	Description
[7:0]	**IAPFD[7:0]**	**IAP Flash Data** This byte contains Flash data, which is read from or is going to be written to the Flash Memory. User should write data into IAPFD for program mode before triggering IAP processing and read data from IAPFD for read/verify mode after IAP processing is finished.

IAPCN – IAP Control

Register	SFR Address	Reset Value
IAPCN	AFH, Page 0	0011_0000 b

7	6	5	4	3	2	1	0
IAPB[1:0]		FOEN	FCEN	FCTRL[3:0]			
R/W		R/W	R/W	R/W			

Bit	Name	Description
[7:6]	**IAPB[1:0]**	**IAP Control** This byte is used for IAP command. For details, see Figure 6.4-1 IAP Modes and Command Codes.
[5]	**FOEN**	**This Byte is Used for IAP Command.** For details, see Figure 6.4-1 IAP Modes and Command Codes.
[4]	**FCEN**	**This Byte is Used for IAP Command.** For details, see Figure 6.4-1 IAP Modes and Command Codes.
[3:0]	**FCTRL[3:0]**	**This Byte is Used for IAP Command.** For details, see Figure 6.4-1 IAP Modes and Command Codes.

PnM1 – Port n Mode Select 1

Register	SFR Address	Reset Value
P0M1	B1H, Page 1	1111_1111 b
P1M1	B3H, Page 1	1111_1111 b
P2M1	B5H, Page 1	1111_1111 b
P3M1	C2H, Page 1	1111_1111 b
P4M1	B9H, Page 1	1111_1111 b
P5M1	BDH, Page 1	1111_1111 b
P6M1	84H, Page 2	1111_1111 b

7	6	5	4	3	2	1	0
P0M1_7	**P0M1_6**	**P0M1_5**	**P0M1_4**	**P0M1_3**	**P0M1_2**	**P0M1_1**	**P0M1_0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:0]	**PnM1[7:0]**	Port n mode select 1

PnM2 – Port n Mode Select 2

Register	SFR Address	Reset Value
P0M2	B2H, Page 1	0000_0000 b
P1M2	B4H, Page 1	0000_0000 b
P2M2	B6H, Page 1	0000_0000 b
P3M2	C3H, Page 1	0000_0000 b
P4M2	BAH, Page 1	0000_0000 b
P5M2	BEH, Page 1	0000_0000 b
P6M2	85H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PnM2_7	**PnM2_6**	**PnM2_5**	**PnM2_4**	**PnM2_3**	**PnM2_2**	**PnM2_1**	**PnM2_0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:0]	**PnM2[7:0]**	Port 0 mode select 2
Note: PnM1 and PnM2 [n:0~5] are used in combination to determine the I/O mode of each pin of P0. See Table 6.5-1 Configuration for Different I/O Modes		

I2C1DAT – I2C1 Data

Register	SFR Address	Reset Value
I2C1DAT	B3H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
I2C1DAT[7:0]							
R/W							

Bit	Name	Description
[7:0]	**I2C1DAT[7:0]**	**I2C1 Data** I2CnDAT contains a byte of the I^2C data to be transmitted or a byte, which has just received. Data in I2CnDAT remains as long as SI is logic 1. The result of reading or writing I2CnDAT during I^2C transceiver progress is unpredicted. While data in I2CnDAT is shifted out, data on the bus is simultaneously being shifted in to update I2CnDAT. I2CnDAT always shows the last byte that presented on the I^2C bus. Thus the event of lost arbitration, the original value of I2CnDAT changes after the transaction.

PWMnINTC – PWM Interrupt Control

Register	SFR Address	Reset Value
PWM0INTC	B7H, Page 1	0000_0000 b
PWM1INTC	9EH, Page 2	0000_0000 b
PWM2INTC	C6H, Page 2	0000_0000 b
PWM3INTC	D6H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
-	**-**	**INTTYP1**	**INTTYP0**	**-**	**INTSEL2**	**INTSEL1**	**INTSEL0**
-	-	R/W	R/W	-	R/W	R/W	R/W

Bit	Name	Description
[7:6]	**-**	Reserved
[5:4]	**INTTYP[1:0]**	**PWM Interrupt Type Select** These bit select PWM interrupt type. 00 = Falling edge on PWMn_CH0/1/2/3/4/5 pin. 01 = Rising edge on PWMn_CH0/1/2/3/4/5 pin. 10 = Central point of a PWM period. 11 = End point of a PWM period. Note that the central point interrupt or the end point interrupt is only available while PWM operates in center-aligned type.
[3]	**-**	Reserved
[2:0]	**INTSEL[2:0]**	**PWM Interrupt Pair Select** These bits select which PWM channel asserts PWM interrupt when PWM interrupt type is selected as falling or rising edge on PWM0/1/2/3/4/5 pin.. 000 = PWMn_CH0. 001 = PWMn_CH1. 010 = PWMn_CH2. 011 = PWMn_CH3. 100 = PWMn_CH4. 101 = PWMn_CH5. Others = PWMn_CH0.

IP – Interrupt Priority

Register	SFR Address	Reset Value
IP	B8H, All pages, Bit addressable	0000_0000 b

7	6	5	4	3	2	1	0
-	PADC	PBOD	PS	PT1	PX1	PT0	PX0
-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	-	Reserved
[6]	PADC	ADC interrupt priority low bit
[5]	PBOD	Brown-out detection interrupt priority low bit
[4]	PS	Serial port 0 interrupt priority low bit
[3]	PT1	Timer 1 interrupt priority low bit
[2]	PX1	External interrupt 1 priority low bit
[1]	PT0	Timer 0 interrupt priority low bit
[0]	PX0	External interrupt 0 priority low bit
Note: used in combination with the IPH to determine the priority of each interrupt source. See Table 6.3-5 Interrupt Priority Level Setting for correct interrupt priority configuration.		

SADENn – UART Slave n Address Mask

Register	SFR Address	Reset Value
SADEN0	B9H, Page 0	0000_0000 b
SADEN1	BAH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
SADENn[7:0]							
R/W							

Bit	Name	Description
[7:0]	**SADENn[7:0]**	**Slave n Address Mask** This byte is a mask byte of UART0 that contains “don’t-care” bits (defined by zeros) to form the device’s “Given” address. The don’t-care bits provide the flexibility to address one or more slaves at a time.

I2CnDAT – I²C Data

Register	SFR Address	Reset Value
I2C0DAT	BCH, Page 0	0000_0000 b
I2C1DAT	B3H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
I2CnDAT[7:0]							
R/W							

Bit	Name	Description
[7:0]	**I2CnDAT[7:0]**	**I²Cn Data** I2CnDAT contains a byte of the I²C data to be transmitted or a byte, which has just received. Data in I2CnDAT remains as long as SI is logic 1. The result of reading or writing I2CnDAT during I²C transceiver progress is unpredicted. While data in I2CnDAT is shifted out, data on the bus is simultaneously being shifted in to update I2CnDAT. I2CnDAT always shows the last byte that presented on the I²C bus. Thus the event of lost arbitration, the original value of I2CnDAT changes after the transaction.

I2CnSTAT – I²C Status

Register	SFR Address	Reset Value
I2C0STAT	BDH, Page 0	1111_1000 b
I2C1STAT	B4H, Page 0	1111_1000 b

7	6	5	4	3	2	1	0
I2CnSTAT[7:3]					**0**	**0**	**0**
R					R	R	R

Bit	Name	Description
[7:3]	**I2CnSTAT[7:3]**	**I²Cn Status Code** The MSB five bits of I2CnSTAT contains the status code. There are 27 possible status codes. When I2CnSTAT is F8H, no relevant state information is available and SI flag keeps 0. All other 26 status codes correspond to the I²C states. When each of these status is entered, SI will be set as logic 1 and a interrupt is requested.
[2:0]	**0**	**Reserved** The least significant three bits of I2CnSTAT are always read as 0.

I2CnCLK – I²C Clock

Register	SFR Address	Reset Value
I2C0CLK	BEH, Page 0	0000_1001 b
I2C1CLK	B5H, Page 0	0000_1001 b

7	6	5	4	3	2	1	0
I2CnCLK[7:0]							
R/W							

Address: BEH, Page 0 Reset value: 0000 1001b

Bit	Name	Description
[7:0]	**I2CnCLK[7:0]**	**I²Cn Clock Setting** In master mode: This register determines the clock rate of I²C bus when the device is in a master mode. The clock rate follows the equation, $\frac{F_{SYS}}{4 \times (I2CLK + 1)}$. Note that the I2CnCLK value of 00H and 01H are not valid. This is an implement limitation. In slave mode: This byte has no effect. In slave mode, the I²C device will automatically synchronize with any given clock rate up to 400k bps.

AUXR3 – Auxiliary Register 3

Register	SFR Address	Reset Value
AUXR3	CFH, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	UART3DG	UART2DG	UART1DG	UART0DG
-	-	-	-	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:4]	-	Reserved
[3]	UART3DG	**UART3 RX Deglitch Control** 1: Deglitch is Enabled 0: Deglitch is Disabled
[2]	UART2DG	**UART2 RX Deglitch Control** 1: Deglitch is Enabled 0: Deglitch is Disabled
[1]	UART1DG	**UART1 RX Deglitch Control** 1: Deglitch is Enabled 0: Deglitch is Disabled
[0]	UART0DG	**UART0 RX Deglitch Control** 1: Deglitch is Enabled 0: Deglitch is Disabled

I2CnTOC – I2Cn Time-out Counter

Register	SFR Address	Reset Value
I2C0TOC	BFH, Page 0	0000_0000b
I2C1TOC	B6H, Page 0	0000_0000b

7	6	5	4	3	2	1	0
-	-	-	-	-	I2TOCEN	DIV	I2TOF
-	-	-	-	-	R/W	R/W	R/W

Bit	Name	Description
[7:3]	-	Reserved
[2]	I2TOCEN	**I²Cn Time-Out Counter Enable** 0 = I²C time-out counter Disabled. 1 = I²C time-out counter Enabled.
[1]	DIV	**I²Cn Time-Out Counter Clock Divider** 0 = The clock of I²C time-out counter is F_{SYS}/1. 1 = The clock of I²C time-out counter is F_{SYS}/4.
[0]	I2TOF	**I²Cn Time-Out Flag** This flag is set by hardware if 14-bit I²C time-out counter overflows. It is cleared by software.

CWKH – Self Wake-up Timer Current Count Value High Byte

Register	SFR Address	Reset Value
CWKH	BEH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
CWK[15:8]							
R/W							

Bit	Name	Description
[7:0]	**CWK[15:8]**	**WKT Current Count Value Low Byte High Byte** It is store value of WKT current count.

RWKH – Self Wake-up Timer Reload High Byte

Register	SFR Address	Reset Value
RWKH	BFH, Page 2	0000 0000b

7	6	5	4	3	2	1	0
RWK[15:8]							
R/W							

Bit	Name	Description
[7:0]	**RWK[15:8]**	**WKT Reload High Byte** It holds the 16-bit reload value of WKT. Note that RWK should not be FFFFH if the pre-scale is 1/1 for implement limitation.

I2CnCON – I²C Control

Register	SFR Address	Reset Value
I2C0CON	C0H, All pages, Bit-addressable	0000_0000 b
I2C1CON	E8H, All pages	0000_0000 b

7	6	5	4	3	2	1	0
I	**I2CEN**	**STA**	**STO**	**SI**	**AA**	**-**	**-**
R/W	R/W	R/W	R/W	R/W	R/W	-	-

Bit	Name	Description
[7]	**I**	**I²Cn Hold Time Extend Enable** 0 = I²C DATA to SCL hold time extend disabled 1 = I²C DATA to SCL hold time extend enabled, extend 8 system clock
[6]	**I2CEN**	**I²Cn Bus Enable** 0 = I²C bus Disabled. 1 = I²C bus Enabled. Before enabling the I²C, SCL and SDA port latches should be set to logic 1.
[5]	**STA**	**START Flag** When STA is set, the I²C generates a START condition if the bus is free. If the bus is busy, the I²C waits for a STOP condition and generates a START condition following. If STA is set while the I²C is already in the master mode and one or more bytes have been transmitted or received, the I²C generates a repeated START condition. Note that STA can be set anytime even in a slave mode, but STA is not hardware automatically cleared after START or repeated START condition has been detected. User should take care of it by clearing STA manually.
[4]	**STO**	**STOP Flag** When STO is set if the I²C is in the master mode, a STOP condition is transmitted to the bus. STO is automatically cleared by hardware once the STOP condition has been detected on the bus. The STO flag setting is also used to recover the I²C device from the bus error state (I2CnSTAT as 00H). In this case, no STOP condition is transmitted to the I²C bus. If the STA and STO bits are both set and the device is original in the master mode, the I²C bus will generate a STOP condition and immediately follow a START condition. If the device is in slave mode, STA and STO simultaneous setting should be avoid from issuing illegal I²C frames.

Bit	Name	Description
[3]	**SI**	**I2C0 Interrupt Flag** SI flag is set by hardware when one of 26 possible I2C status (besides F8H status) is entered. After SI is set, the software should read I2CnSTAT register to determine which step has been passed and take actions for next step. SI is cleared by software. Before the SI is cleared, the low period of SCL line is stretched. The transaction is suspended. It is useful for the slave device to deal with previous data bytes until ready for receiving the next byte. The serial transaction is suspended until SI is cleared by software. After SI is cleared, I2C bus will continue to generate START or repeated START condition, STOP condition, 8-bit data, or so on depending on the software configuration of controlling byte or bits. Therefore, user should take care of it by preparing suitable setting of registers before SI is software cleared.
[2]	**AA**	**Acknowledge Assert Flag** If the AA flag is set, an ACK (low level on SDA) will be returned during the acknowledge clock pulse of the SCL line while the I2C device is a receiver or an own-address-matching slave. If the AA flag is cleared, a NACK (high level on SDA) will be returned during the acknowledge clock pulse of the SCL line while the I2C device is a receiver or an own-address-matching slave. A device with its own AA flag cleared will ignore its own salve address and the General Call. Consequently, SI will note be asserted and no interrupt is requested. Note that if an addressed slave does not return an ACK under slave receiver mode or not receive an ACK under slave transmitter mode, the slave device will become a not addressed slave. It cannot receive any data until its AA flag is set and a master addresses it again. There is a special case of I2CnSTAT value C8H occurs under slave transmitter mode. Before the slave device transmit the last data byte to the master, AA flag can be cleared as 0. Then after the last data byte transmitted, the slave device will actively switch to not addressed slave mode of disconnecting with the master. The further reading by the master will be all FFH.
[1:0]	**-**	Reserved

I2CnADDRx – I2Cn Own Slave Address

Register	SFR Address	Reset Value
I2C0ADDR0	C1H, Page 0	0000_0000 b
I2C0ADDR1	A1H, Page 2	0000_0000 b
I2C0ADDR2	A2H, Page 2	0000_0000 b
I2C0ADDR3	A3H, Page 2	0000_0000 b
I2C1ADDR0	B2H, Page 0	0000_0000 b
I2C1ADDR1	A4H, Page 2	0000_0000 b
I2C1ADDR2	A5H, Page 2	0000_0000 b
I2C1ADDR3	A6H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
I2CnADDRx[7:1]							GC
R/W							R/W

Bit	Name	Description
[7:1]	**I2CnADDRx[7:1]**	**I2C0 Device's Own Slave Address** In master mode: These bits have no effect. In slave mode: These 7 bits define the slave address of this I2C device by user. The master should address I2C device by sending the same address in the first byte data after a START or a repeated START condition. If the AA flag is set, this I2C device will acknowledge the master after receiving its own address and become an addressed slave. Otherwise, the addressing from the master will be ignored. Note that I2CnADDRx[7:1] should not remain its default value of all 0, because address 0x00 is reserved for General Call.
[0]	**GC**	**General Call Bit** In master mode: This bit has no effect. In slave mode: 0 = The General Call is always ignored. 1 = The General Call is recognized if AA flag is 1; otherwise, it is ignored if AA is 0.

CKDIV – Clock Divider

Register	SFR Address	Reset Value
CKDIV	C1H, Page 1	0000_0000b

7	6	5	4	3	2	1	0
CKDIV[7:0]							
R/W							

Bit	Name	Description
[7:0]	**CKDIV[7:0]**	**Clock Divider** The system clock frequency F_{SYS} follows the equation below according to CKDIV value. $F_{SYS} = F_{OSC}$, while CKDIV = 00H, $F_{SYS} = \frac{F_{OSC}}{2 \times CKDIV}$, while CKDIV = 01H to FFH.

ADCRL – ADC Result Low Byte

Register	SFR Address	Reset Value
ADCRL	C2H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	ADCR[3:0]			
-	-	-	-	R			

Bit	Name	Description
[7:4]	-	Reserved
[3:0]	ADCR[3:0]	**ADC Result Low Byte** The least significant 4 bits of the ADC result stored in this register.

PWMnCxH – PWM0/1/2/3 Channel 0~5 Duty High Byte n=0,1,2,3; x=0,1,2,3,4,5

Register	SFR Address	Reset Value
PWM0C0H	D2H, Page 1	0000_0000 b
PWM0C1H	D3H, Page 1	0000_0000 b
PWM0C2H	D4H, Page 1	0000_0000 b
PWM0C3H	D5H, Page 1	0000_0000 b
PWM0C4H	C4H, Page 1	0000_0000 b
PWM0C5H	C5H, Page 1	0000_0000 b
PWM1C0H	8AH, Page 2	0000_0000 b
PWM1C1H	8BH, Page 2	0000_0000 b
PWM2C0H	BAH, Page 2	0000_0000 b
PWM2C1H	BBH, Page 2	0000_0000 b
PWM3C0H	CAH, Page 2	0000_0000 b
PWM3C1H	CBH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PWMnCx [15:8], n=0,1,2,3; x=0,1,2,3,4,5							
R/W							

Bit	Name	Description
[7:0]	**PWMnCx [15:8], n=0,1,2,3; x=0,1,2,3,4,5**	**PWMnCx Duty High Byte** This byte with PWMnCxL controls the duty of the output signal PGx from PWM generator.

ADCRH – ADC Result High Byte

Register	SFR Address	Reset Value
ADCRH	C3H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
ADCR[11:4]							
R							

Bit	Name	Description
[7:0]	**ADCR[11:4]**	**ADC Result High Byte** The most significant 8 bits of the ADC result stored in this register.

T3CON – Timer 3 Control

Register	SFR Address	Reset Value
T3CON	C4H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
SMOD_1	SMOD0_1	BRCK	TF3	TR3	T3PS[2:0]		
R/W	R/W	R/W	R/W	R/W	R/W		

Bit	Name	Description
[7]	SMOD_1	**Serial Port 1 Double Baud Rate Enable** Setting this bit doubles the serial port baud rate when UART1 is in Mode 2.see Table 6.10-2 Serial Port 1 Mode / baud rate Description for details.
[6]	SMOD0_1	**Serial Port 1 Framing Error Access Enable** 0 = S1CON.7 accesses to SM0_1 bit. 1 = S1CON.7 accesses to FE_1 bit.
[5]	BRCK	**Serial Port 0 Baud Rate Clock Source** This bit selects which Timer is used as the baud rate clock source when serial port 0 is in Mode 1 or 3. 0 = Timer 1. 1 = Timer 3.
[4]	TF3	**Timer 3 Overflow Flag** This bit is set when Timer 3 overflows. It is automatically cleared by hardware when the program executes the Timer 3 interrupt service routine. This bit can be set or cleared by software.
[3]	TR3	**Timer 3 Run Control** 0 = Timer 3 is halted. 1 = Timer 3 starts running. Note that the reload registers RH3 and RL3 can only be written when Timer 3 is halted (TR3 bit is 0). If any of RH3 or RL3 is written if TR3 is 1, result is unpredictable.
[2:0]	T3PS[2:0]	**Timer 3 Pre-Scalar** These bits determine the scale of the clock divider for Timer 3. 000 = 1/1. 001 = 1/2. 010 = 1/4. 011 = 1/8. 100 = 1/16. 101 = 1/32. 110 = 1/64. 111 = 1/128.

RL3 – Timer 3 Reload Low Byte

Register	SFR Address	Reset Value
RL3	C5H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
RL3[7:0]							
R/W							

Bit	Name	Description
[7:0]	**RL3[7:0]**	**Timer 3 Reload Low Byte** It holds the low byte of the reload value of Timer 3.

RH3 – Timer 3 Reload High Byte

Register	SFR Address	Reset Value
RH3	C6H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
RH3[15:8]							
R/W							

Bit	Name	Description
[7:0]	**RH3[15:8]**	**Timer 3 Reload High Byte** It holds the high byte of the reload value of Time 3.

PORDIS – POR Disable (TA Protected)

Register	SFR Address	Reset Value
PORDIS	C6H, Page 1, TA protected	0000_0000 b

7	6	5	4	3	2	1	0
PORDIS[7:0]							
W							

Bit	Name	Description
[7:0]	**PORDIS[7:0]**	**POR Disable** To first writing 5AH to the PORDIS and immediately followed by a writing of A5H will disable all of PORs (POR50 and POR15).

TA – Timed Access

Register	SFR Address	Reset Value
TA	C7H, All pages	0000_0000 b

7	6	5	4	3	2	1	0
TA[7:0]							
W							

Bit	Name	Description
[7:0]	**TA[7:0]**	**Timed Access** The timed access register controls the access to protected SFR. To access protected bits, user should first write AAH to the TA and immediately followed by a write of 55H to TA. After these two steps, a writing permission window is opened for 4 clock cycles during this period that user may write to protected SFR.

T2CON – Timer 2 Control

Register	SFR Address	Reset Value
T2CON	C8H, All pages, Bit addressable	0000_0000 b

7	6	5	4	3	2	1	0
TF2	**-**	**-**	**-**	**-**	**TR2**	**-**	**CM_RL2**
R/W	-	-	-	-	R/W	-	R/W

Bit	Name	Description
[7]	**TF2**	**Timer 2 Overflow Flag** This bit is set when Timer 2 overflows or a compare match occurs. If the Timer 2 interrupt and the global interrupt are enable, setting this bit will make CPU execute Timer 2 interrupt service routine. This bit is not automatically cleared via hardware and should be cleared via software.
[6:3]	**-**	Reserved
[2]	**TR2**	**Timer 2 Run Control** 0 = Timer 2 Disabled. Clearing this bit will halt Timer 2 and the current count will be preserved in TH2 and TL2. 1 = Timer 2 Enabled.
[1]	**-**	Reserved
[0]	**CM_RL2**	**Timer 2 Compare or Auto-Reload Mode Select** This bit selects Timer 2 functioning mode. 0 = Auto-reload mode. 1 = Compare mode.

T2MOD – Timer 2 Mode

Register	SFR Address	Reset Value
T2MOD	C9H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
LDEN	T2DIV[2:0]			CAPCR	CMPCR	LDTS[1:0]	
R/W	R/W			R/W	R/W	R/W	

Bit	Name	Description
[7]	LDEN	**Enable Auto-Reload** 0 = Reloading RCMP2H and RCMP2L to TH2 and TL2 Disabled. 1 = Reloading RCMP2H and RCMP2L to TH2 and TL2 Enabled.
[6:4]	T2DIV[2:0]	**Timer 2 Clock Divider** 000 = Timer 2 clock divider is 1/1. 001 = Timer 2 clock divider is 1/4. 010 = Timer 2 clock divider is 1/16. 011 = Timer 2 clock divider is 1/32. 100 = Timer 2 clock divider is 1/64. 101 = Timer 2 clock divider is 1/128. 110 = Timer 2 clock divider is 1/256. 111 = Timer 2 clock divider is 1/512.
[3]	CAPCR	**Capture Auto-Clear** This bit is valid only under Timer 2 auto-reload mode. It enables hardware auto-clearing TH2 and TL2 counter registers after they have been transferred in to RCMP2H and RCMP2L while a capture event occurs. 0 = Timer 2 continues counting when a capture event occurs. 1 = Timer 2 value is auto-cleared as 0000H when a capture event occurs.
[2]	CMPCR	**Compare Match Auto-Clear** This bit is valid only under Timer 2 compare mode. It enables hardware auto-clearing TH2 and TL2 counter registers after a compare match occurs. 0 = Timer 2 continues counting when a compare match occurs. 1 = Timer 2 value is auto-cleared as 0000H when a compare match occurs.
[1:0]	LDTS[1:0]	**Auto-Reload Trigger Select** These bits select the reload trigger event. 00 = Reload when Timer 2 overflows. 01 = Reload when input capture 0 event occurs. 10 = Reload when input capture 1 event occurs. 11 = Reload when input capture 2 event occurs.

AUXR1 – Auxiliary Register 1

Register	SFR Address	Reset Value
AUXR1	C9H, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
-	**-**	**-**	**-**	**UART3PX**	**UART2PX**	**UART1PX**	**UART0PX**
-	-	-	-	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:4]	**-**	Reserved
[3]	**UART3PX**	**Serial Port 3 RX (SMC1 DATA) /TX (SMC1 CLK) Pin Exchange** 0 = Assign UART3 RXD (SMC1 DATA) to multiple I/O pin RXD UART3 TXD (SMC CLK) to multiple I/O pin TXD 1 = Assign UART3 RXD (SMC1 DATA) to multiple I/O pin TXD UART3 TXD (SMC CLK) to multiple I/O pin RXD Note : that Pin direction is controlled by I/O type of relative pin. RXD/TXD will exchange immediately once setting or clearing this bit. User should take care of not exchanging pins during transmission or receiving. Or it may cause unpredictable situation and no warning alarms.
[2]	**UART2PX**	**Serial Port 2 RX (SMC0 DATA) /TX (SMC0 CLK) Pin Exchange** 0 = Assign UART2 RXD (SMC0 DATA) to multiple I/O pin RXD UART2 TXD (SMC CLK) to multiple I/O pin TXD 1 = Assign UART2 RXD (SMC0 DATA) to multiple I/O pin TXD UART2 TXD (SMC CLK) to multiple I/O pin RXD Note : that Pin direction is controlled by I/O type of relative pin. RXD/TXD will exchange immediately once setting or clearing this bit. User should take care of not exchanging pins during transmission or receiving. Or it may cause unpredictable situation and no warning alarms.
[1]	**UART1PX**	**Serial Port 1 RX/TX Pin Exchange** 0 = Assign UART1 RXD to multiple I/O pin RXD UART1 TXD to multiple I/O pin TXD 1 = Assign UART1 RXD to multiple I/O pin TXD UART1 TXD to multiple I/O pin RXD **Note:** that Pin direction is controlled by I/O type of relative pin. RXD/TXD will exchange immediately once setting or clearing this bit. User should take care of not exchanging pins during transmission or receiving. Or it may cause unpredictable situation and no warning alarms.

Bit	Name	Description
[0]	**UART0PX**	**Serial Port 0 RX/TX Pin Exchange** 0 = Assign UART0 RXD to multiple I/O pin RXD UART0 TXD to multiple I/O pin TXD 1 = Assign UART0 RXD to multiple I/O pin TXD UART0 TXD to multiple I/O pin RXD **Note:** that Pin direction is controlled by I/O type of relative pin. RXD/TXD will exchange immediately once setting or clearing this bit. User should take care of not exchanging pins during transmission or receiving. Or it may cause unpredictable situation and no warning alarms.

PIF – Pin Interrupt Flags

Register	SFR Address	Reset Value
PIF	CAH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
PIF7	**PIF6**	**PIF5**	**PIF4**	**PIF3**	**PIF2**	**PIF1**	**PIF0**
R (level) R/W (edge)	R (level) R/W (edge)	R (level) R/W (edge)	R (level) R/W (edge)	R (level) R/W (edge)	R (level) R/W (edge)	R (level) R/W (edge)	R (level) R/W (edge)

Bit	Name	Description
[7:0]	**PIFn**	**Pin Interrupt Channel n Flag** If the edge trigger is selected, this flag will be set by hardware if the channel n of pin interrupt detects an enabled edge trigger. This flag should be cleared by software. If the level trigger is selected, this flag follows the inverse of the input signal's logic level on the channel n of pin interrupt. Software cannot control it.

RCMP2L– Timer 2 Reload/Compare Low Byte

Register	SFR Address	Reset Value
RCMP2L	CAH, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
RCMP2[7:0]							
R/W							

Bit	Name	Description
[7:0]	**RCMP2[7:0]**	**Timer 2 Reload/Compare Low Byte** This register stores the low byte of compare value when Timer 2 is configured in compare mode. Also it holds the low byte of the reload value in auto-reload mode.

ADCBAL – ADC RAM Base Address Low Byte

Register	SFR Address	Reset Value
ADCBAL	CBH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
ADCBAL[7:0]							
R/W							

Bit	Name	Description
[7:0]	**ADCBAL[7:0]**	**ADC RAM Base Address (Low Byte)** The least significant 8 bits of RAM base address to store ADC continue sampling data. RAM base address ADCBA[11:0] = { ADCBAH[3:0], ADCBAL[7:0]}

RCMP2H – Timer 2 Reload/Compare High Byte

Register	SFR Address	Reset Value
RCMP2H	CBH, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
RCMP2[15:8]							
R/W							

Bit	Name	Description
[7:0]	**RCMP2[15:8]**	**Timer 2 Reload/Compare High Byte** This register stores the high byte of compare value when Timer 2 is configured in compare mode. Also it holds the high byte of the reload value in auto-reload mode.

TL2 – Timer 2 Low Byte

Register	SFR Address	Reset Value
TL2	CCH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
T2[7:0]							
R/W							

Bit	Name	Description
[7:0]	**T2[7:0]**	**Timer 2 Low Byte** The TL2 register is the low byte of the 16-bit counting register of Timer 2.

TH2 – Timer 2 High Byte

Register	SFR Address	Reset Value
TH2	CDH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
T2[15:8]							
R/W							

Bit	Name	Description
[7:0]	**T2[15:8]**	**Timer 2 High Byte** The TH2 register is the high byte of the 16-bit counting register of Timer 2.

ADCMPL – ADC Compare Low Byte

Register	SFR Address	Reset Value
ADCMPL	CEH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	ADCMP[3:0]			
-	-	-	-	W/R			

Bit	Name	Description
[7:4]	-	Reserved
[3:0]	ADCMP[3:0]	**ADC Compare Low Byte** The least significant 4 bits of the ADC compare value stores in this register.

AINDIDS0 – ADC Channel Digital Input Disconnect

Register	SFR Address	Reset Value
AINDIDS0	CEH, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
AIN7DIDS	**AIN6DIDS**	**AIN5DIDS**	**AIN4DIDS**	**AIN3DIDS**	**AIN2DIDS**	**AIN1DIDS**	**AIN0DIDS**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:0]	**AINnDIDS**	**ADC Channel Digital Input Disable** 0 = Enabled digital input at ADC channel n. 1 = Disabled digital input at ADC channel n . ADC channel n is read always 0.

AINDIDS1 – ADC Channel Digital Input Disconnect

Register	SFR Address	Reset Value
AINDIDS1	CEH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
AIN15DIDS	**AIN14DIDS**	**AIN13DIDS**	**AIN12DIDS**	**AIN11DIDS**	**AIN10DIDS**	**-**	**-**
R/W	R/W	R/W	R/W	R/W	R/W	-	-

Bit	Name	Description
[7:0]	**AINnDIDS**	**ADC Channel Digital Input Disable** 0 = Enabled digital input at ADC channel n. 1 = Disabled digital input at ADC channel n . ADC channel n is read always 0.

PWM0FBS – PWM Brake Source Select

Register	SFR Address	Reset Value
PWM0FBS	CEH, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	-	PWM0FBS	
						R/W	

Bit	Name	Description
[1:0]	**PWM0FBS**	**PWM Brake Source Select** 00 = GPIO (depended on Multi-function register select). 01 = Reserved 10 = ACMP0 11 = ACMP1

ADCMPH – ADC Compare High Byte

Register	SFR Address	Reset Value
ADCMPH	CFH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
ADCMP[11:4]							
W/R							

Bit	Name	Description
[7:0]	**ADCMP[11:4]**	**ADC Compare High Byte** The most significant 8 bits of the ADC compare value stores in this register.

I2CnADDRM – I2Cn Address Mask

Register	SFR Address	Reset Value
I2C0ADDRM	CFH, Page 2	0000_0000 b
I2C1ADDRM	D7H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
Mask Bit 7	**Mask Bit 6**	**Mask Bit 5**	**Mask Bit 4**	**Mask Bit 3**	**Mask Bit 2**	**Mask Bit 1**	**Mask Bit 0**
R/W							

Bit	Name	Description
[7:0]	**I2CnADDRM**	**I2Cn Address Mask** Mask with bit

PSW – Program Status Word

Register	SFR Address	Reset Value
PSW	D0H, All pages, Bit addressable	0000_0000 b

7	6	5	4	3	2	1	0
CY	**AC**	**F0**	**RS1**	**RS0**	**OV**	**F1**	**P**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Name	Description
[7]	**CY**	**Carry Flag** For a adding or subtracting operation, CY will be set when the previous operation resulted in a carry-out from or a borrow-in to the Most Significant bit, otherwise cleared. If the previous operation is MUL or DIV, CY is always 0. CY is affected by DA A instruction, which indicates that if the original BCD sum is greater than 100. For a CJNE branch, CY will be set if the first unsigned integer value is less than the second one. Otherwise, CY will be cleared.
[6]	**AC**	**Auxiliary Carry** Set when the previous operation resulted in a carry-out from or a borrow-in to the 4th bit of the low order nibble, otherwise cleared.
[5]	**F0**	**User Flag 0** The general purpose flag that can be set or cleared by user.
[4]	**RS1**	**Register Bank Selection Bits** These two bits select one of four banks in which R0 to R7 locate. RS1 / RS0 / Register Bank / RAM Address 0 / 0 / 0 / 00H to 07H 0 / 1 / 1 / 08H to 0FH 1 / 0 / 2 / 10H to 17H 1 / 1 / 3 / 18H to 1FH
[3]	**RS0**	Check with bit 4 description.
[2]	**OV**	**Overflow Flag** OV is used for a signed character operands. For a ADD or ADDC instruction, OV will be set if there is a carry out of bit 6 but not out of bit 7, or a carry out of bit 7 but not bit 6. Otherwise, OV is cleared. OV indicates a negative number produced as the sum of two positive operands or a positive sum from two negative operands. For a SUBB, OV is set if a borrow is needed into bit6 but not into bit 7, or into bit7 but not bit 6. Otherwise, OV is cleared. OV indicates a negative number produced when a negative value is subtracted from a positive value, or a positive result when a positive number is subtracted from a negative number. For a MUL, if the product is greater than 255 (00FFH), OV will be set. Otherwise, it is cleared. For a DIV, it is normally 0. However, if B had originally contained 00H, the values returned in A and B will be undefined. Meanwhile, the OV will be set.

Bit	Name	Description
[1]	**F1**	**User Flag 1** The general purpose flag that can be set or cleared by user via software.
[0]	**P**	**Parity Flag** Set to 1 to indicate an odd number of ones in the accumulator. Cleared for an even number of ones. It performs even parity check.

PWM0CON0 – PWM Control Register0

Register	SFR Address	Reset Value
PWM0CON0	D1H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
PWM0RUN	**LOAD**	**PWMF**	**CLRPWM**	**-**	**-**	**-**	**-**
R/W	R/W	R/W	R/W	-	-	-	-

Bit	Name	Description
[7]	**PWM0RUN**	**PWM0 Run Enable** 0 = PWM0 stays in idle. 1 = PWM0 starts running.
[6]	**LOAD**	**PWM New Period and Duty Load** This bit is used to load period and duty Register Description in their buffer if new period or duty value needs to be updated. The loading will act while a PWM period is completed. The new period and duty affected on the next PWM cycle. After the loading is complete, LOAD will be automatically cleared via hardware. The meaning of writing and reading LOAD bit is different. Writing: 0 = No effect. 1 = Load new period and duty in their buffers while a PWM period is completed. Reading: 0 = A loading of new period and duty is finished. 1 = A loading of new period and duty is not yet finished.
[5]	**PWMF**	**PWM Flag** This flag is set according to definitions of INTSEL[2:0] and INTTYP[1:0] in PWMnINTC. This bit is cleared by software.
[4]	**CLRPWM**	**Clear PWM Counter** Setting this bit clears the value of PWM 16-bit counter for resetting to 0000H. After the counter value is cleared, CLRPWM will be automatically cleared via hardware. The meaning of writing and reading CLRPWM bit is different. Writing: 0 = No effect. 1 = Clearing PWM 16-bit counter. Reading: 0 = PWM 16-bit counter is completely cleared. 1 = PWM 16-bit counter is not yet cleared.
[3:0]	**-**	Reserved

PWMnCON0 – PWM Control Register0

Register	SFR Address	Reset Value
PWM1CON0	9CH, Page 2	0000_0000 b
PWM2CON0	C4H, Page 2	0000_0000 b
PWM3CON0	D4H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PWMnRUN	**LOAD**	**PWMF**	**CLRPWM**	**-**	**-**	**-**	**-**
R/W	R/W	R/W	R/W	-	-	-	-

Bit	Name	Description
[7]	**PWMnRUN**	**PWMn Run Enable** 0 = PWM stays in idle. 1 = PWM starts running.
[6]	**LOAD**	**PWM New Period and Duty Load** This bit is used to load period and duty Register Description in their buffer if new period or duty value needs to be updated. The loading will act while a PWM period is completed. The new period and duty affected on the next PWM cycle. After the loading is complete, LOAD will be automatically cleared via hardware. The meaning of writing and reading LOAD bit is different. Writing: 0 = No effect. 1 = Load new period and duty in their buffers while a PWM period is completed. Reading: 0 = A loading of new period and duty is finished. 1 = A loading of new period and duty is not yet finished.
[5]	**PWMF**	**PWM Flag** This flag is set according to definitions of INTSEL[2:0] and INTTYP[1:0] in PWMnINTC. This bit is cleared by software.
[4]	**CLRPWM**	**Clear PWM Counter** Setting this bit clears the value of PWM 16-bit counter for resetting to 0000H. After the counter value is cleared, CLRPWM will be automatically cleared via hardware. The meaning of writing and reading CLRPWM bit is different. Writing: 0 = No effect. 1 = Clearing PWM 16-bit counter. Reading: 0 = PWM 16-bit counter is completely cleared. 1 = PWM 16-bit counter is not yet cleared.
[3:0]	**-**	Reserved

PWMnPH – PWM Period High Byte

Register	SFR Address	Reset Value
PWM0PH	D1H, Page 1	0000_0000 b
PWM1PH	86H, Page 2	0000_0000 b
PWM2PH	B9H, Page 2	0000_0000 b
PWM3PH	C9H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PWMnP[15:8]							
R/W							

Bit	Name	Description
[7:0]	**PWMnP[15:8]**	**PWM Period High Byte** This byte with PWMnPL controls the period of the PWM generator signal.

ACMPCR0 – Analog Comparator Control Register 0

Register	SFR Address	Reset Value
ACMPCR0	D2H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
POSSEL		NEGSEL		WKEN	HYSEN	ACMPIE	ACMPEN
R/W		R/W		R/W	R/W	R/W	R/W

Bit	Name	Description
[7:6]	**POSSEL**	**Comparator 0 Positive Input Selection** 00 = ACMP0_P0 (P2.5) pin. 01 = ACMP0_P1 (P2.3) pin. 10 = ACMP0_P2 (P2.1) pin. 11 = ACMP0_P3 (P3.0) pin.
[5:4]	**NEGSEL**	**Comparator 0 Negative Input Selection** 00 = ACMP0_N0 (P2.4) pin. 01 = Internal comparator reference voltage (CRV). 10 = VBG (Band-gap). 11 = ACMP0_N1 (P2.0)pin.
[3]	**WKEN**	**Comparator 0 Power-Down Wake-Up Enable Bit** 0 = Comparator 0 Wake-up function Disabled. 1 = Comparator 0 Wake-up function Enabled.
[2]	**HYSEN**	**Comparator 0 Hysteresis Enable Bit** 0 = Comparator 0 hysteresis Disabled. 1 = Comparator 0 hysteresis Enabled.
[1]	**ACMPIE**	**Comparator 0 Interrupt Enable Bit** 0 = Comparator 0 interrupt Disabled. 1 = Comparator 0 interrupt Enabled. If WKEN (ACMPCR1[3]) is set to 1, the wake-up interrupt function will be enabled as well.
[0]	**ACMPEN**	**Comparator 0 Enable Bit** 0 = Comparator 0 Disabled. 1 = Comparator 0 Enabled.

ACMPCR1 – Analog Comparator Control Register 1

Register	SFR Address	Reset Value
ACMPCR1	D3H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
POSSEL		NEGSEL		WKEN	HYSEN	ACMPIE	ACMPEN
R/W		R/W		R/W	R/W	R/W	R/W

Bit	Name	Description
[7:6]	**POSSEL**	**Comparator 1 Positive Input Selection** 00 = ACMP1_P0 (P2.5) pin. 01 = ACMP1_P1 (P2.3) pin. 10 = ACMP1_P2 (P2.1) pin. 11 = ACMP1_P3 (P3.0) pin.
[5:4]	**NEGSEL**	**Comparator 1 Negative Input Selection** 00 = ACMP1_N0 (P2.2) pin. 01 = Internal comparator reference voltage (CRV). 10 = VBG (Band-gap). 11 = ACMP1_N1 (P3.1)pin.
[3]	**WKEN**	**Comparator 1 Power-Down Wake-Up Enable Bit** 0 = Comparator 1 Wake-up function Disabled. 1 = Comparator 1 Wake-up function Enabled.
[2]	**HYSEN**	**Comparator 1 Hysteresis Enable Bit** 0 = Comparator 1 hysteresis Disabled. 1 = Comparator 1 hysteresis Enabled.
[1]	**ACMPIE**	**Comparator 1 Interrupt Enable Bit** 0 = Comparator 1 interrupt Disabled. 1 = Comparator 1 interrupt Enabled. If WKEN (ACMPCR2[3]) is set to 1, the wake-up interrupt function will be enabled as well.
[0]	**ACMPEN**	**Comparator 1 Enable Bit** 0 = Comparator 1 Disabled. 1 = Comparator 1 Enabled.

ACMPSR – Analog Comparator Status Register

Register	SFR Address	Reset Value
ACMPSR	D4H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
-				ACMP1O	ACMP1IF	ACMP0O	ACMP0IF
-				R	R/W	R	R/W

Bit	Name	Description
[7:4]	-	Reserved
[3]	ACMP1O	**Comparator 1 Output** Synchronized to the PCLK to allow reading by software. Cleared when the comparator 1 is disabled, i.e. ACMPEN (ACMPCR1[0]) is cleared to 0. **Note:** This bit is read only.
[2]	ACMP1IF	**Comparator 1 Interrupt Flag** This bit is set by hardware whenever the comparator 1 output changes state. This will generate an interrupt if ACMPIE (ACMPCR1[1]) is set to 1 **Note:** Write “0” to clear this bit to 0.
[1]	ACMP0O	**Comparator 0 Output** Synchronized to the PCLK to allow reading by software. Cleared when the comparator 0 is disabled, i.e. ACMPEN (ACMPCR0[0]) is cleared to 0. **Note:** This bit is read only.
[0]	ACMP0IF	**Comparator 0 Interrupt Flag** This bit is set by hardware whenever the comparator 0 output changes state. This will generate an interrupt if ACMPIE (ACMPCR0[1]) is set to 1 **Note:** Write “0” to clear this bit to 0.

ACMPV$_{REF}$ – ACMP Reference Voltage Control Register

Register	SFR Address	Reset Value
ACMPV$_{REF}$	D5H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
-	CRV1CTL[2:0]			-	CRV0CTL[2:0]		
-	R/W			-	R/W		

Bit	Name	Description
[7]	-	Reserved
[6:4]	**CRV1CTL[2:0]**	**Comparator 1 Reference Voltage Setting** CRV1 = CRV source voltage * (2/12+CRV1CTL/12).
[3]	-	Reserved
[2:0]	**CRV0CTL[2:0]**	**Comparator 0 Reference Voltage Setting** CRV0 = CRV source voltage * (2/12+CRV0CTL/12).

SCnCR0 – SC Control Register

Register	SFR Address	Reset Value
SC0CR0	D6H, Page 0	0000_0000 b
SC1CR0	E6H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
NSB	**T**	**RXBGTEN**	**CONSEL**	**AUTOCEN**	**TXOFF**	**RXOFF**	**SCEN**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**NSB**	**Stop Bit Length** This field indicates the length of stop bit. 0 = The stop bit length is 2 ETU. 1= The stop bit length is 1 ETU. **Note:** The default stop bit length is 2. SC and UART adopt NSB to program the stop bit length.
[6]	**T**	**T Mode** 0 = T0 (ISO7816-3 T = 0 mode). 1 = T1 (ISO7816-3 T = 1 mode). The T mode controls the BGT (Block Guard Time). Block guard time means the minimum bit length between the leading edges of two consecutive characters between different transfer directions. This field indicates the counter for the bit length of block guard time. According to ISO7816-3, in T = 0 mode, the software must clear T bit to 0 for real block guard time = 16.5. In T = 1 mode, the software must set T bit to 1 for real block guard time = 22.5. **Note:** In T = 0 mode, the receiver will generate the parity flag on PEF(SC0TSR[4]), if parity error is detected and also drive the parity error signal to transceiver. In T = 1 mode, the receiver will generate the parity flag on PEF(SC0TSR[4]), if parity error detected, but doesn't drive the parity error signal to transceiver. **Note:** The description please see section 6.11.5.2 Error Signal and Character Repetition
[5]	**RXBGTEN**	**Receiver Block Guard Time Function Enable Bit** 0 = Receiver block guard time function Disabled. 1 = Receiver block guard time function Enabled.
[4]	**CONSEL**	**Convention Selection** 0 = Direct convention. 1 = Inverse convention. **Note 1:** This bit is auto clear to "0", if AUTOCEN(SCnCR0[3]) is writing "1" **Note 2:** If AUTOCEN(SCnCR0[3]) is enabled, hardware will decide the convention and change the CONSEL (SCnCR0[4]) bits automatically after SCEN (SCnCR0[0]) ="1".

Bit	Name	Description
[3]	**AUTOCEN**	**Auto Convention Enable Bit** 0 = Auto-convention Disabled. 1 = Auto-convention Enabled. When hardware receives TS in answer to reset state and the TS is direct convention, CONSEL(SCnCR0[4]) will be set to 0 automatically, otherwise if the TS is inverse convention, and CONSEL (SCnCR0[4]) will be set to 1. **Note:** If software enables auto convention function, the setting step must be done before Answer to Reset state and the first data must be 0x3B or 0x3F. After hardware received first data and stored it at buffer, hardware will decided the convention and change the CONSEL (SCnCR0[4]) bits automatically.
[2]	**TXOFF**	**TX Transition Disable Bit** 0 = The transceiver Enabled. 1 = The transceiver Disabled.
[1]	**RXOFF**	**RX Transition Disable Bit** 0 = The receiver Enabled. 1 = The receiver Disabled. **Note:** If AUTOCEN (SCnCR0[3])is enabled, these fields must be ignored.
[0]	**SCEN**	**SC Engine Enable Bit** Set this bit to 1 to enable SC operation. If this bit is cleared, SC will force all transition to IDLE state **Note:** SCEN must be set to 1 before filling in other registers, or smart card will not work properly.

PWM0NP – PWM Negative Polarity

Register	SFR Address	Reset Value
PWM0NP	D6H, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
-	**-**	**PNP5**	**PNP4**	**PNP3**	**PNP2**	**PNP1**	**PNP0**
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[5:0]	**PNPn**	**PWMn Negative Polarity Output Enable** 0 = PWMn signal outputs directly on PWMn pin. 1 = PWMn signal outputs inversely on PWMn pin.

SCnCR1 – SC Control Register

Register	SFR Address	Reset Value
SC0CR1	D7H, Page 0	0000_0000 b
SC1CR1	E7H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
OPE	PBOFF	WLS[1:0]		TXDMAEN	RXDMAEN	CLKKEEP	UARTEN
R/W	R/W	R/W		R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	OPE	**Odd Parity Enable Bit** 0 = Even number of logic 1's are transmitted or check the data word and parity bits in receiving mode. 1 = Odd number of logic 1's are transmitted or check the data word and parity bits in receiving mode. **Note:** This bit has effect only when PBOFF bit is '0'.
[6]	PBOFF	**Parity Bit Disable Control** 0 = Parity bit is generated or checked between the "last data word bit" and "stop bit" of the serial data. 1 = Parity bit is not generated (transmitting data) or checked (receiving data) during transfer. **Note:** In smart card mode, this field must be '0' (default setting is with parity bit)
[5:4]	WLS[1:0]	**Word Length Selection** 00 = Word length is 8 bits. 01 = Word length is 7 bits. 10 = Word length is 6 bits. 11 = Word length is 5 bits. **Note:** In smart card mode, this WLS must be '00'
[3]	TXDMAEN	**SC/UART TX DMA Enable** This bit enables the SC/UART TX operating by through PDMA transfer, TX data needs to be ready in XRAM before SC/UART TX starting. 0 = SPI TX DMA Disabled 1 = SPI TX DMA Enabled
[2]	RXDMAEN	**SC/UART RX DMA Enable** This bit enables the SC/UART RX operating by through PDMA transfer, RX data are saved in XRAM after SC/UART RX operation. 0 = SC/UART RX DMA Disabled 1 = SC/UART RX DMA Enabled
[1]	CLKKEEP	**SC Clock Enable Bit** 0 = SC clock generation Disabled. 1 = SC clock always keeps free running.

Bit	Name	Description
[0]	**UARTEN**	**UART Mode Enable Bit** 0 = Smart Card mode. 1 = UART mode. **Note 1:**When operating in UART mode, user must set CONSEL (SCnCR0[4]) = 0 and AUTOCEN(SCnCR0[3]) = 0. **Note 2:**When operating in Smart Card mode, user must set UARTEN(SCnCR1 [0]) = 0. **Note 3:**When UART is enabled, hardware will generate a reset to reset FIFO and internal state machine.

PWM0FBD – PWM Fault Brake Data

Register	SFR Address	Reset Value
PWM0FBD	D7H, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
FBF	**FBINLS**	**FBD5**	**FBD4**	**FBD3**	**FBD2**	**FBD1**	**FBD0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**FBF**	**Fault Brake Flag** This flag is set when FBINEN is set as 1 and FB pin detects an edge, which matches FBINLS (PWM0FBD.6) selection. This bit is cleared by software. After FBF is cleared, Fault Brake data output will not be released until PWM0RUN (PWM0CON0.7) is set.
[6]	**FBINLS**	**PWM_BRAKE Pin Input Level Selection** 0 = Falling edge. 1 = Rising edge.
[5:0]	**FBDn**	**PWMn Fault Brake Data** 0 = PWMn signal is overwritten by 0 once Fault Brake asserted. 1 = PWMn signal is overwritten by 1 once Fault Brake asserted.

SCnDR – SC Data Register

Register	SFR Address	Reset Value
SC0DR	D9H, Page 0	0000_0000 b
SC1DR	D9H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
SCnDR[7:0]							
R/W							

Bit	Name	Description
[7:0]	**SCnDR[7:0]**	**SC / UART Buffer Data** This byte is used for transmitting or receiving data on SC / UART bus. A write of this byte is a write to the shift register. A read of this byte is actually a read of the read data buffer. **Note:** If SCEN(SCnCR0[0]) is not enabled, SCnDR cannot be programmed.

PWMnPL – PWM Period Low Byte

Register	SFR Address	Reset Value
PWM0PL	D9H, Page 1	0000_0000 b
PWM1PL	99H, Page 2	0000_0000 b
PWM2PL	C1H, Page 2	0000_0000 b
PWM3PL	D1H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PWMnP[7:0]							
R/W							

Bit	Name	Description
[7:0]	**PWMnP[7:0]**	**PWMn Period Low Byte** This byte with PWMnPH controls the period of the PWM generator signal.

SCnEGT – SC Extra Guard Time Register

Register	SFR Address	Reset Value
SC0EGT	DAH, Page 0	0000_0000 b
SC1EGT	DAH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
SCnEGT[7:0]							
R/W							

Bit	Name	Description
[7:0]	**SCnEGT[7:0]**	**SC Extra Guard Time** This field indicates the extra guard timer value. **Note:** The counter is ETU base .

PWMnCxL – PWM0/1/2/3 Channel 0~5 Duty Low Byte n=0,1,2,3; x=0,1,2,3,4,5

Register	SFR Address	Reset Value
PWM0C0L	DAH, Page 1	0000_0000 b
PWM0C1L	DBH, Page 1	0000_0000 b
PWM0C2L	DCH, Page 1	0000_0000 b
PWM0C3L	DDH, Page 1	0000_0000 b
PWM0C4L	CCH, Page 1	0000_0000 b
PWM0C5L	CDH, Page 1	0000_0000 b
PWM1C0L	9AH, Page 2	0000_0000 b
PWM1C1L	9BH, Page 2	0000_0000 b
PWM2C0L	C2H, Page 2	0000_0000 b
PWM2C1L	C3H, Page 2	0000_0000 b
PWM3C0L	D2H, Page 2	0000_0000 b
PWM3C1L	D3H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PWMnCx [7:0], n=0,1,2,3; x=0,1,2,3,4,5							
R/W							

Bit	Name	Description
[7:0]	**PWMnCx [7:0], n=0,1,2,3; x=0,1,2,3,4,5**	**PWMnCx Duty Low Byte** This byte with PWMnCxH controls the duty of the output signal PGx from PWM generator.

SCnETURD0 – SCn ETU Rate Divider Register

Register	SFR Address	Reset Value
SC0ETURD0	DBH, Page 0	0111_0011 b
SC1ETURD0	DBH, Page 2	0111_0011 b

7	6	5	4	3	2	1	0
ETURDIV[7:0]							
R/W							

Bit	Name	Description
[7:0]	**ETURDIV[7:0]**	**LSB Bits of ETU Rate Divider** The field indicates the LSB of clock rate divider. The real ETU is ETURDIV[11:0] + 1. **Note 1:** ETURDIV[11:0] must be greater than 0x004. **Note 2:** SCnETURD0 has to program first, then SCnETUDR2.

SCnETURD1 –SC ETU Rate Divider Register

Register	SFR Address	Reset Value
SC0ETURD1	DCH, Page 0	0011_0001 b
SC1ETURD1	DCH, Page 2	0011_0001 b

7	6	5	4	3	2	1	0
-	SCDIV[2:0]			ETURDIV[11:8]			
-	R/W			R/W			

Bit	Name	Description
[7]	-	Reserved
[6:4]	SCDIV[2:0]	**SC Clock Divider** 000 = F_{SC} is F_{SYS}/1. 001 = F_{SC} is F_{SYS}/2. 010 = F_{SC} is F_{SYS}/4. 011 = F_{SC} is F_{SYS}/8. (By default.) 100 = F_{SC} is F_{SYS}/16. 101 = F_{SC} is F_{SYS}/16. 110 = F_{SC} is F_{SYS}/16. 111 = F_{SC} is F_{SYS}/16. **Note:** that the F_{SC} clock should be 1Mhz ~ 5Mhz for ISO/IEC 7816-3 standard
[3:0]	ETURDIV[11:8]	**MSB Bits of ETU Rate Divider** The field indicates the MSB of clock rate divider. The real ETU is ETURDIV[11:0] + 1. **Note 1:** ETURDIV[11:0] must be greater than 0x004. **Note 2:** SCnETURD0 has to program first, then SCnETUDR1 .

ScnIE – SC Interrupt Enable Control Register

Register	SFR Address	Reset Value
SC0IE	DDH, Page 0	0000_0000 b
SC1IE	DDH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	ACERRIEN	BGTIEN	TERRIEN	TBEIEN	RDAIEN
-	-	-	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:5]	-	Reserved
[4]	ACERRIEN	**Auto Convention Error Interrupt Enable Bit** This field is used to enable auto-convention error interrupt. 0 = Auto-convention error interrupt Disabled. 1 = Auto-convention error interrupt Enabled.
[3]	BGTIEN	**Block Guard Time Interrupt Enable Bit** This field is used to enable block guard time interrupt. 0 = Block guard time interrupt Disabled. 1 = Block guard time interrupt Enabled.
[2]	TERRIEN	**Transfer Error Interrupt Enable Bit** This field is used to enable transfer error interrupt. The transfer error states is at SC0TSR register which includes receiver break error BEF(SC0TSR[6]), frame error FEF(SC0TSR[5]), parity error PEF(SC0TSR[4]), receiver buffer overflow error RXOV(SC0TSR[0]) and transmit buffer overflow error TXOV(SC0TSR[2]). 0 = Transfer error interrupt Disabled. 1 = Transfer error interrupt Enabled.
[1]	TBEIEN	**Transmit Buffer Empty Interrupt Enable Bit** This field is used to enable transmit buffer empty interrupt. 0 = Transmit buffer empty interrupt Disabled. 1 = Transmit buffer empty interrupt Enabled.
[0]	RDAIEN	**Receive Data Reach Interrupt Enable Bit** This field is used to enable received data interrupt. 0 = Receive data interrupt Disabled. 1 = Receive data interrupt Enabled.

ScnIS – SC Interrupt Status Register

Register	SFR Address	Reset Value
SC0IS	DEH, Page 0	0000_0010 b
SC1IS	DEH, Page 2	0000_0010 b

7	6	5	4	3	2	1	0
-		Tx_Er	ACERRIF	BGTIF	TERRIF	TBEIF	RDAIF
-		R/W	R/W	R/W	R	R	R

Bit	Name	Description
[7:6]	-	Reserved.
[5]	Tx_Er	TX transmit error flag
[4]	ACERRIF	**Auto Convention Error Interrupt Status Flag (Read Only)** This field indicates auto convention sequence error. If the received TS at ATR state is neither 0x3B nor 0x3F, this bit will be set. **Note:** This bit is read only, but it can be cleared by writing “0” to it.
[3]	BGTIF	**Block Guard Time Interrupt Status Flag (Read Only)** This field is used for block guard time interrupt status flag. **Note 1:** This bit is valid when RXBGTEN (SCnCR0[5]) is enabled. **Note 2:** This bit is read only, but it can be cleared by writing “0” to it.
[2]	TERRIF	**Transfer Error Interrupt Status Flag (Read Only)** This field is used for transfer error interrupt status flag. The transfer error states is at SC0TSR register which includes receiver break error BEF(SC0TSR[6]), frame error FEF(SC0TSR[5]), parity error PEF(SC0TSR[4]) and receiver buffer overflow error RXOV(SC0TSR[0]) and transmit buffer overflow error TXOV(SC0TSR[2]). **Note:** This field is the status flag of BEF(SC0TSR[6]), FEF(SC0TSR[5]), PEF(SC0TSR[4]), RXOV(SC0TSR[0]) and TXOV(SC0TSR[2]). So, if software wants to clear this bit, software must write “0” to each field.
[1]	TBEIF	**Transmit Buffer Empty Interrupt Status Flag (Read Only)** This field is used for transmit buffer empty interrupt status flag. **Note:** This field is the status flag of transmit buffer empty state. If software wants to clear this bit, software must write data to DAT(SCnDR[7:0]) buffer and then this bit will be cleared automatically.
[0]	RDAIF	**Receive Data Reach Interrupt Status Flag (Read Only)** This field is used for received data interrupt status flag. **Note:** This field is the status flag of received data. If software reads data from SC_DAT pin, this bit will be cleared automatically.

SCnTSR – SC Transfer Status Register

Register	SFR Address	Reset Value
SC0TSR	DFH, Page 0	0000_1010 b
SC1TSR	DFH, Page 2	0000_1010 b

7	6	5	4	3	2	1	0
ACT	**BEF**	**FEF**	**PEF**	**TXEMPTY**	**TXOV**	**RXEMPTY**	**RXOV**
R	R/W	R/W	R/W	R	R/W	R	R/W

Bit	Name	Description
[7]	**ACT**	**Transmit /Receive in Active Status Flag (Read Only)** 0 = This bit is cleare automatically when TX/RX transfer is finished 1 = This bit is set by hardware when TX/RX transfer is in active.
[6]	**BEF**	**Receiver Break Error Status Flag (Read Only)** This bit is set to logic 1 whenever the received data input (RX) held in the “spacing state” (logic 0) is longer than a full word transmission time (that is, the total time of “start bit” + data bits + parity + stop bits). . **Note:** This bit is read only, but it can be cleared by writing 0 to it.
[5]	**FEF**	**Receiver Frame Error Status Flag (Read Only)** This bit is set to logic 1 whenever the received character does not have a valid “stop bit” (that is, the stop bit following the last data bit or parity bit is detected as logic 0). **Note:** This bit is read only, but it can be cleared by writing 0 to it.
[4]	**PEF**	**Receiver Parity Error Status Flag (Read Only)** This bit is set to logic 1 whenever the received character does not have a valid “parity bit”. **Note:** This bit is read only, but it can be cleared by writing 0 to it.
[3]	**TXEMPTY**	**Transmit Buffer Empty Status Flag (Read Only)** This bit indicates TX buffer empty or not. **Note:** When TX buffer has been transferred to Transmitter Shift Register, hardware sets this bit high. It will be cleared when writing data into DAT(SCnDR[7:0]) (TX buffer not empty).
[2]	**TXOV**	**TX Overflow Error Interrupt Status Flag (Read Only)** If TX buffer is full, an additional write to DAT(SCnDR[7:0]) will cause this bit be set to “1” by hardware. **Note:** This bit is read only, but it can be cleared by writing 0 to it.
[1]	**RXEMPTY**	**Receiver Buffer Empty Status Flag(Read Only)** This bit indicates RX buffer empty or not. **Note:** When Rx buffer has been read by CPU, hardware sets this bit high. It will be cleared when SC receives any new data.

Bit	Name	Description
[0]	**RXOV**	**RX Overflow Error Status Flag (Read Only)** This bit is set when RX buffer overflow. **Note:** This bit is read only, but it can be cleared by writing 0 to it.

PWMnCON1 – PWM Control 1

Register	SFR Address	Reset Value
PWM0CON1	DFH, Page 1	0000_0000 b
PWM1CON1	9DH, Page 2	0000_0000 b
PWM2CON1	C5H, Page 2	0000_0000 b
PWM3CON1	D5H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PWMMOD[1:0]		GP	PWMTYP	FBINEN	PWMDIV[2:0]		
R/W		R/W	R/W	R/W	R/W		

Bit	Name	Description
[7:6]	**PWMMOD[1:0]**	**PWM Mode Select** 00 = Independent mode. 01 = Complementary mode. 10 = Synchronized mode. 11 = Reserved.
[5]	**GP**	**Group Mode Enable** This bit enables the group mode. If enabled, the duty of first three pairs of PWM are decided by PWM01H and PWM01L rather than their original duty Register Description. 0 = Group mode Disabled. 1 = Group mode Enabled.
[4]	**PWMTYP**	**PWM Type Select** 0 = Edge-aligned PWM. 1 = Center-aligned PWM.
[3]	**FBINEN**	**FB Pin Input Enable** 0 = PWM0 output Fault Braked by FB pin input Disabled. 1 = PWM0 output Fault Braked by FB pin input Enabled. Once an edge, which matches FBINLS (PWM0FBD.6) selection, occurs on FB pin, PWM0CH0~5 output Fault Brake data in PWMnFBD register. PWMRUN (PWM0CON0.7) will also be automatically cleared by hardware. The PWM output resumes when PWM0RUN is set again. **Note:** This bit is only vaild in PWM0

Bit	Name	Description
[2:0]	**PWMDIV[2:0]**	**PWM Clock Divider** This field decides the pre-scale of PWM clock source. 000 = 1/1. 001 = 1/2 010 = 1/4. 011 = 1/8. 100 = 1/16. 101 = 1/32. 110 = 1/64. 111 = 1/128.

A or ACC – Accumulator (Bit-addressable)

Register	SFR Address	Reset Value
ACC	E0H, All pages, Bit addressable	0000_0000 b

7	6	5	4	3	2	1	0
ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:0]	**ACC[7:0]**	**Accumulator** The A or ACC register is the standard 80C51 accumulator for arithmetic operation.

ADCCON1 – ADC Control 1

Register	SFR Address	Reset Value
ADCCON1	E1H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	HIE	CONT	ETGTYP[1:0]		ADCEX	ADCEN
-	-	R/W	R/W	R/W		R/W	R/W

Bit	Name	Description
[7:6]	-	Reserved
[5]	HIE	**ADC Half Done Interrupt Enable** 0 = ADC interrupt is not set while half of A/D conversions are complete in continue mode 1 = ADC interrupt is set while half of A/D conversions are complete in continue mode
[4]	CONT	**ADC Continue Sampling Select** 0 = ADC single sampling, ADC interrupt is set while an A/D conversion is completed 1 = ADC continue sampling. ADC interrupt is set while total A/D conversions are completed
[3:2]	ETGTYP[1:0]	**External Trigger Type Select** When ADCEX (ADCCON1.1) is set, these bits select which condition triggers ADC conversion. 00 = Falling edge on PWM0/2/4 or STADC pin. 01 = Rising edge on PWM0/2/4 or STADC pin. 10 = Central point of a PWM period. 11 = End point of a PWM period. Note that the central point interrupt or the period point interrupt is only available for PWM center-aligned type.
[1]	ADCEX	**ADC External Conversion Trigger Select** This bit to select the methods of triggering an A/D conversion. 0 = A/D conversion is started only via setting ADCS bit. 1 = A/D conversion is started via setting ADCS bit or by external trigger source depending on ETGSEL[1:0] and ETGTYP[1:0]. Note that while ADCS is 1 (busy in converting), the ADC will ignore the following external trigger until ADCS is hardware cleared.
[0]	ADCEN	**ADC Enable** 0 = ADC circuit off. 1 = ADC circuit on.

CAPCON0 – Input Capture Control 0

Register	SFR Address	Reset Value
CAPCON0	E1H, Page 1	0000_0000b

7	6	5	4	3	2	1	0
-	CAPEN2	CAPEN1	CAPEN0	-	CAPF2	CAPF1	CAPF0
-	R/W	R/W	R/W	-	R/W	R/W	R/W

Bit	Name	Description
[7]	-	Reserved
[6]	CAPEN2	**Input Capture 2 Enable** 0 = Input capture channel 2 Disabled. 1 = Input capture channel 2 Enabled.
[5]	CAPEN1	**Input Capture 1 Enable** 0 = Input capture channel 1 Disabled. 1 = Input capture channel 1 Enabled.
[4]	CAPEN0	**Input Capture 0 Enable** 0 = Input capture channel 0 Disabled. 1 = Input capture channel 0 Enabled.
[3]	-	Reserved
[2]	CAPF2	**Input Capture 2 Flag** This bit is set by hardware if the determined edge of input capture 2 occurs. This bit should cleared by software.
[1]	CAPF1	**Input Capture 1 Flag** This bit is set by hardware if the determined edge of input capture 1 occurs. This bit should cleared by software.
[0]	CAPF0	**Input Capture 0 Flag** This bit is set by hardware if the determined edge of input capture 0 occurs. This bit should cleared by software.

ADCCON2 – ADC Control 2

Register	SFR Address	Reset Value
ADCCON2	E2H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
ADFBEN	**ADCMPOP**	**ADCMPEN**	**ADCMPO**	**ADCAQT[2:0]**			**ADCDLY.8**
R/W	R/W	R/W	R	R/W			R/W

Bit	Name	Description
[7]	**ADFBEN**	**ADC Compare Result Asserting Fault Brake Enable** 0 = ADC asserting Fault Brake Disabled. 1 = ADC asserting Fault Brake Enabled. Fault Brake is asserted once its compare result ADCMPO is 1. Meanwhile, PWM channels output Fault Brake data. PWMRUN (PWMnCON0.7) will also be automatically cleared by hardware. The PWM output resumes when PWMRUN is set again.
[6]	**ADCMPOP**	**ADC Comparator Output Polarity** 0 = ADCMPO is 1 if ADCR[11:0] is greater than or equal to ADCMP[11:0]. 1 = ADCMPO is 1 if ADCR[11:0] is less than ADCMP[11:0].
[5]	**ADCMPEN**	**ADC Result Comparator Enable.** ADC result comparator to trig ADCF enable bit. Only when comparator value match the condition of ADC compare value defined ADCF will be set to 1. This condition base on ADCMPH, ADCMPL and ADCMPOP register define. The ADCF register changes to 1 only when ADC comparing result matches the condition and then enters interrupt vector if ADC interrupt is enabled. 0 = ADC result comparator trig ADCF Disabled. 1 = ADC result comparator trig ADCF Enabled. **Note:** After this bit is enabled and ADC start is triggered, the ADC keeps converting. The register ADCRH and ADCRL value will change based on the result of ADC setting and can also be read out from the register. This process only stops after ADCF is set to 1
[4]	**ADCMPO**	**ADC Comparator Output Value** This bit is the output value of ADC result comparator based on the setting of ACMPOP. This bit updates after every A/D conversion complete.
[3:1]	**ADCAQT[2:0]**	**ADC Acquisition Time** This 3-bit field decides the acquisition time for ADC sampling, following by equation below: ADC acquisition time = $\frac{4 * ADCAQT + 10}{F_{ADC}}$. The default and minimum acquisition time is 10 ADC clock cycles. Note that this field should not be changed when ADC is in converting.
[0]	**ADCDLY.8**	**ADC External Trigger Delay Counter Bit 8** See ADCDLY register.

CAPCON1 – Input Capture Control Register

Register	SFR Address	Reset Value
CAPCON1	E2H, Page 1	0000_0000b

7	6	5	4	3	2	1	0
-	-	CAP2LS[1:0]		CAP1LS[1:0]		CAP0LS[1:0]	
-	-	R/W		R/W		R/W	

Bit	Name	Description
[7:6]	-	Reserved
[5:4]	CAP2LS[1:0]	**Input Capture 2 Level Select** 00 = Falling edge. 01 = Rising edge. 10 = Either rising or falling edge. 11 = Reserved.
[3:2]	CAP1LS[1:0]	**Input Capture 1 Level Select** 00 = Falling edge. 01 = Rising edge. 10 = Either rising or falling edge. 11 = Reserved.
[1:0]	CAP0LS[1:0]	**Input Capture 0 Level Select** 00 = Falling edge. 01 = Rising edge. 10 = Either rising or falling edge. 11 = Reserved.

ADCDLY – ADC Trigger Delay Counter

Register	SFR Address	Reset Value
ADCDLY	E3H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
ADCDLY[7:0]							
R/W							

Bit	Name	Description
[7:0]	**ADCDLY[7:0]**	**ADC External Trigger Delay Counter Low Byte** This 8-bit field combined with ADCCON2.0 forms a 9-bit counter. This counter inserts a delay after detecting the external trigger. An A/D converting starts after this period of delay. External trigger delay time = $\frac{ADCDLY}{F_{ADC}}$. Note that this field is valid only when ADCEX (ADCCON1.1) is set. User should not modify ADCDLY during PWM run time if selecting PWM output as the external ADC trigger source.

CAPCON2 – Input Capture Control 2

Register	SFR Address	Reset Value
CAPCON2	E3H, Page 1	0000_0000b

7	6	5	4	3	2	1	0
-	ENF2	ENF1	ENF0	-	-	-	-
-	R/W	R/W	R/W	-	-	-	-

Bit	Name	Description
[7]	-	Reserved
[6]	ENF2	**Enable Noise Filer on Input Capture 2** 0 = Noise filter on input capture channel 2 Disabled. 1 = Noise filter on input capture channel 2 Enabled.
[5]	ENF1	**Enable Noise Filer on Input Capture 1** 0 = Noise filter on input capture channel 1 Disabled. 1 = Noise filter on input capture channel 1 Enabled.
[4]	ENF0	**Enable Noise Filer on Input Capture 0** 0 = Noise filter on input capture channel 0 Disabled. 1 = Noise filter on input capture channel 0 Enabled.
[3:0]	-	Reserved

ADCBAH – ADC RAM Base Address High Byte

Register	SFR Address	Reset Value
ADCBAH	E4H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
-				ADCBA[11:8]			
-				R/W			

Bit	Name	Description
[7:4]	-	Reserved
[3:0]	ADCBA[11:8]	**ADC RAM Base Address (High Byte)** The most significant 4 bits of RAM base address to store ADC continue sampling data. RAM base address ADCBA[11:0] = {ADCBAH[3:0], ADCBAL[7:0]}

CnL – Capture Data Low Byte, n = 0,1,2

Register	SFR Address	Reset Value
C0L	E4H, Page 1	0000_0000 b
C1L	E6H, Page 1	0000_0000 b
C2L	EDH, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
CnL[7:0]							
R/W							

Bit	Name	Description
[7:0]	**CnL[7:0]**	**Input Capture 0 Result Low Byte** The C0L register is the low byte of the 16-bit result captured by input capture 0.

ADCSN – ADC Sampling Number

Register	SFR Address	Reset Value
ADCSN	E5H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
ADCSN[7:0]							
R/W							

Bit	Name	Description
[7:0]	**ADCSN[7:0]**	**ADC Sampling Number** The total sampling numbers for ADC continue sampling select. Total sampling number= ADCSN[7:0] + 1

CnH – Capture Data High Byte, n = 1,2,3

Register	SFR Address	Reset Value
C0H	E5H, Page 1	0000_0000 b
C1H	E7H, Page 1	0000_0000 b
C2H	EEH, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
CnH[7:0]							
R/W							

Bit	Name	Description
[7:0]	**CnH[7:0]**	**Input Capture n Result High Byte** The CnH register is the high byte of the 16-bit result captured by input capture n.

ADCCN – ADC Current Sampling Number

Register	SFR Address	Reset Value
ADCCN	E6H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
ADCCN[7:0]							
R							

Bit	Name	Description
[7:0]	**ADCCN[7:0]**	**ADC Current Sampling Number** The current sampling numbers for ADC continue sampling select. The current sampling number= ADCCN[7:0] + 1

ADCSR – ADC Status Register

Register	SFR Address	Reset Value
ADCSR	E7H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
SLOW	**ADCDIV[2:0]**			**-**	**CMPHIT**	**HDONE**	**FDONE**
R/W	R/W			-	R/W	R/W	R/W

Bit	Name	Description
[7]	**SLOW**	**ADC Slow Speed Selection** This bit is used to select ADC low speed. 0 = high speed 500 R/W 1 = low speed 200 R/W
[6:4]	**ADCDIV[2:0]**	**ADC Clock Divider** 000 = FADC is FSYS/1. 001 = FADC is FSYS/2. 010 = FADC is FSYS/4. 011 = FADC is FSYS/8. 100 = FADC is FSYS/16. 101 = FADC is FSYS/32. 110 = FADC is FSYS/64. 111 = FADC is FSYS/128.
[3]	**-**	Reserved
[2]	**CMPHIT**	**ADC Comparator Hit Flag** This bit is set by hardware when ADCMPO (ADCCON2.4) flag rising **Note:** This bit can be cleared by writing 0 to it.
[1]	**HDONE**	**A/D Conversion Half Done Flag** This bit is set by hardware when half of ADCSN A/D conversions are complete in continue mode. **Note:** This bit can be cleared by writing 0 to it
[0]	**FDONE**	**A/D Conversion Full Done Flag** This bit is set by hardware when all of ADCSN A/D conversions are complete in continue mode or single conversion in single mode. **Note:** This bit can be cleared by writing 0 to it..

DMAnTSR – PDMAn Transfer Status Register

Register	SFR Address	Reset Value
DMA0TSR	E9H, Page 0	0000_0000 b
DMA1TSR	F1H, Page 0	0000_0000 b
DMA2TSR	B1H, Page 2	0000_0000 b
DMA3TSR	A9H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
-					ACT	HDONE	FDONE
-					R	R/W	R/W

Bit	Name	Description
[7:3]	-	**Reserved**
[2]	ACT	**PDMA in Active Status Flag (Read Only)** 0 = This bit is cleared automatically when PDMA transfer is done or disabled. 1 = This bit is set by hardware when PDMA transfer is in active.
[1]	HDONE	**PDMA Half Transfer Done Flag** This bit is set by hardware when PDMA half transfer is done. **Note:** This bit can be cleared by writing 0 to it.
[0]	FDONE	**PDMA Full Transfer Done Flag** This bit is set by hardware when PDMA full transfer is done. **Note:** This bit can be cleared by writing 0 to it.

PICON – Pin Interrupt Control

Register	SFR Address	Reset Value
PICON	E9H, Page 1	0000 _0000 b

7	6	5	4	3	2	1	0
PIT7	PIT6	PIT5	PIT4	PIT3	PIT2	PIT1	PIT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	PIT7	**Pin Interrupt Channel 7 Type Select** This bit selects which type that pin interrupt channel 7 is triggered. 0 = Level triggered. 1 = Edge triggered.
[6]	PIT6	**Pin Interrupt Channel 6 Type Select** This bit selects which type that pin interrupt channel 6 is triggered. 0 = Level triggered. 1 = Edge triggered.
[5]	PIT5	**Pin Interrupt Channel 5 Type Select** This bit selects which type that pin interrupt channel 5 is triggered. 0 = Level triggered. 1 = Edge triggered.
[4]	PIT4	**Pin Interrupt Channel 4 Type Select** This bit selects which type that pin interrupt channel 4 is triggered. 0 = Level triggered. 1 = Edge triggered.
[3]	PIT3	**Pin Interrupt Channel 3 Type Select** This bit selects which type that pin interrupt channel 3 is triggered. 0 = Level triggered. 1 = Edge triggered.
[2]	PIT2	**Pin Interrupt Channel 2 Type Select** This bit selects which type that pin interrupt channel 2 is triggered. 0 = Level triggered. 1 = Edge triggered.
[1]	PIT1	**Pin Interrupt Channel 1 Type Select** This bit selects which type that pin interrupt channel 1 is triggered. 0 = Level triggered. 1 = Edge triggered.

Bit	Name	Description
[0]	**PIT0**	**Pin Interrupt Channel 0 Type Select** This bit selects which type that pin interrupt channel 0 is triggered. 0 = Level triggered. 1 = Edge triggered.

MTMnDA – Memory to Memory Destination Address Low Byte

Register	SFR Address	Reset Value
MTM0DA	EAH, Page 0	0000_0000 b
MTM1DA	F2H, Page 0	0000_0000 b
MTM2DA	B7H, Page 2	0000_0000 b
MTM3DA	AFH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
MTMnDA[7:0]							
R/W							

Bit	Name	Description
[7:0]	**MTMnDA[7:0]**	**Memory to Memory Destination Address (Low Byte)** The least significant 8 bits of XRAM address are used for memory to memory destination address. XRAM destination address = {MDAH[3:0], MDAL[7:0]}

PINEN – Pin Interrupt Negative Polarity Enable.

Register	SFR Address	Reset Value
PINEN	EAH, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
PINEN7	**PINEN6**	**PINEN5**	**PINEN4**	**PINEN3**	**PINEN2**	**PINEN1**	**PINEN0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:0]	**PINENn**	**Pin Interrupt Channel n Negative Polarity Enable** This bit enables low-level/falling edge triggering pin interrupt channel n. The level or edge triggered selection depends on each control bit PITn in PICON. 0 = Low-level/falling edge detect Disabled. 1 = Low-level/falling edge detect Enabled.

PIPEN – Pin Interrupt Positive Polarity Enable.

Register	SFR Address	Reset Value
PIPEN	EBH, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
PIPEN7	**PIPEN6**	**PIPEN5**	**PIPEN4**	**PIPEN3**	**PIPEN2**	**PIPEN1**	**PIPEN0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:0]	**PIPENn**	**Pin Interrupt Channel n Positive Polarity Enable** This bit enables high-level/rising edge triggering pin interrupt channel n. The level or edge triggered selection depends on each control bit PITn in PICON. 0 = High-level/rising edge detect Disabled. 1 = High-level/rising edge detect Enabled.

EIP0 – Extensive Interrupt Priority

Register	SFR Address	Reset Value
EIP0	EFH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
PT2	**PSPI0**	**PFB**	**PWDT**	**PPWM0**	**PCAP**	**PPI**	**PI2C0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**PT2**	Timer 2 interrupt priority low bit
[6]	**PSPI0**	SPI0 interrupt priority low bit
[5]	**PFB**	Fault Brake interrupt priority low bit
[4]	**PWDT**	WDT interrupt priority low bit
[3]	**PPWM0**	PWM interrupt priority low bit
[2]	**PCAP**	Input capture interrupt priority low bit
[1]	**PPI**	Pin interrupt priority low bit
[0]	**PI2C0**	R/W interrupt priority low bit
Note: EIP0 is used in combination with the EIPH0 to determine the priority of each interrupt source. See Table 6.3-5 Interrupt Priority Level Setting for correct interrupt priority configuration.		

B – B Register (Bit-addressable)

Register	SFR Address	Reset Value
B	F0H, All pages, Bit addressable	0000_0000 b

7	6	5	4	3	2	1	0
B.7	**B.6**	**B.5**	**B.4**	**B.3**	**B.2**	**B.1**	**B.0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:0]	**B[7:0]**	**B Register** The B register is the other accumulator of the standard 80C51 .It is used mainly for MUL and DIV instructions.

LCDCPUMP – LCD Charge Pump Voltage Set

Register	SFR Address	Reset Value
LCDCPUMP	F1H, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-	-	VCP_SEL[5:0]					
-	-	R/W					

Address: F1H, Page 3 Reset value: 0000 0000b

Bit	Name	Description
[7:6]	-	Reserved
[5:0]	VCP_SEL[5:0]	**Charge Pump Voltage Set** 000000 = 5.2V 000101 = 5.0V 001010 = 4.8V 010000 = 4.6V 010101 = 4.4V 011010 = 4.2V 100000 = 4.0V 100101 = 3.8V 101010 = 3.6V 101111 = 3.4V 110100 = 3.2V 111010 = 3.0V 111111 = 2.8V

SPInCR0 – Serial Peripheral Control Register0

Register	SFR Address	Reset Value
SPI0CR0	F3H, Page 0	0000_0000 b
SPI1CR0	F9H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
SSOE	**SPIEN**	**LSBFE**	**MSTR**	**CPOL**	**CPHA**	**SPR1**	**SPR0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**SSOE**	**Slave Select Output Enable** This bit is used in combination with the DISMODF (SPInSR.3) bit to determine the feature of $\overline{SS}$ pin as shown inTable 6.12-3 Slave Select Pin Configurations. This bit takes effect only under MSTR = 1 and DISMODF = 1 condition. 0 = $\overline{SS}$ functions as a general purpose I/O pin. 1 = $\overline{SS}$ automatically goes low for each transmission when selecting external Slave device and goes high during each idle state to de-select the Slave device.
[6]	**SPIEN**	**SPI Enable** 0 = SPI function Disabled. 1 = SPI function Enabled.
[5]	**LSBFE**	**LSB First Enable** 0 = The SPI data is transferred MSB first. 1 = The SPI data is transferred LSB first.
[4]	**MSTR**	**Master Mode Enable** This bit switches the SPI operating between Master and Slave modes. 0 = The SPI is configured as Slave mode. 1 = The SPI is configured as Master mode.
[3]	**CPOL**	**SPI Clock Polarity Select** CPOL bit determines the idle state level of the SPI clock. See Figure 6.12-4 SPI Clock Formats 0 = The SPI clock is low in idle state. 1 = The SPI clock is high in idle state.
[2]	**CPHA**	**SPI Clock Phase Select** CPHA bit determines the data sampling edge of the SPI clock. See Figure 6.12-4 SPI Clock Formats 0 = The data is sampled on the first edge of the SPI clock. 1 = The data is sampled on the second edge of the SPI clock.

Bit	Name	Description
[1:0]	**SPR[1:0]**	**SPI Clock Rate Select** These four bits select four grades of SPI clock divider. The clock rates below are illustrated under F_{SYS} = 24 MHz condition. See Table 6.12-1 SPI Master Clock Rate Define Table SPR[3:0] are valid only under Master mode (MSTR = 1). If under Slave mode, the clock will automatically synchronize with the external clock on SPICLK pin from Master device up to F_{SYS}/4 communication speed.

SPInCR1 – Serial Peripheral Control Register1

Register	SFR Address	Reset Value
SPI0CR1	F3H, Page 1	0000_0000 b
SPI1CR1	FAH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
-		SPR3	SPR2	TXDMAEN	RXDMAEN	SPIS1	SPIS0
-		R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:6]	-	Reserved.
[5:4]	SPR[3:2]	**SPI Clock Rate Select** These two bits select four grades of SPI clock divider. The clock rates below are illustrated under F_{SYS} = 24 MHz condition. Table 6.12-1 SPI Master Clock Rate Define Table SPR[3:0] are valid only under Master mode (MSTR = 1). If under Slave mode, the clock will automatically synchronize with the external clock on SPICLK pin from Master device up to F_{SYS}/4 communication speed.
[3]	TXDMAEN	**SPI TX DMA Enable** This bit enables the SPI TX operating by through PDMA transfer, TX data needs to be ready in XRAM before SPI TX starting. 0 = SPI TX DMA Disabled 1 = SPI TX DMA Enabled
[2]	RXDMAEN	**SPI RX DMA Enable** This bit enables the SPI RX operating by through PDMA transfer, RX data are saved in XRAM after SPI RX operation. 0 = SPI RX DMA Disabled 1 = SPI RX DMA Enabled
[1:0]	SPIS[1:0]	**SPI Interval Time Selection Between Adjacent Bytes** SPIS[1:0] and CPHA select eight grades of SPI interval time selection between adjacent bytes. As see Table 6.12-2 SPI Clock Suspend Interval Select SPIS[1:0] are valid only under Master mode (MSTR = 1).

SPInSR – Serial Peripheral Status Register

Register	SFR Address	Reset Value
SPI0SR	F4H, Page 0	0000_0000 b
SPI1SR	FBH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
SPIF	**WCOL**	**SPIOVF**	**MODF**	**DISMODF**	**DISSPIF**	**TXBFF**	**-**
R/W	R/W	R/W	R/W	R/W	R/W	R	-

Bit	Name	Description
[7]	**SPIF**	**SPI Complete Flag** This bit is set to logic 1 via hardware while an SPI data transfer is complete or an receiving data has been moved into the SPI read buffer. If ESPI and EA are enabled, an SPI interrupt will be required. This bit should be cleared via software. Attempting to write to SPInDR is inhibited if SPIF is set.
[6]	**WCOL**	**Write Collision Error Flag** This bit indicates a write collision event. Once a write collision event occurs, this bit will be set. It should be cleared via software.
[5]	**SPIOVF**	**SPI Overrun Error Flag** This bit indicates an overrun event. Once an overrun event occurs, this bit will be set. If ESPI and EA are enabled, an SPI interrupt will be required. This bit should be cleared via software.
[4]	**MODF**	**Mode Fault Error Flag** This bit indicates a Mode Fault error event. If $\overline{SS}$ pin is configured as Mode Fault input (MSTR = 1 and DISMODF = 0) and $\overline{SS}$ is pulled low by external devices, a Mode Fault error occurs. Instantly MODF will be set as logic 1. If ESPI and EA are enabled, an SPI interrupt will be required. This bit should be cleared via software.
[3]	**DISMODF**	**Disable Mode Fault Error Detection** This bit is used in combination with the SSOE (SPInCR.7) bit to determine the feature of $\overline{SS}$ pin as shown in Table 6.12-3 Slave Select Pin Configurations. DISMODF is valid only in Master mode (MSTR = 1). 0 = Mode Fault detection Enabled. $\overline{SS}$ serves as input pin for Mode Fault detection disregard of SSOE. 1 = Mode Fault detection Disabled. The feature of $\overline{SS}$ follows SSOE bit.
[2]	**DISSPIF**	**Disable SPI Complete Interrupt** This bit is used to disable SPI complete interrupt while an SPI data transfer is complete or an receiving data has been moved into the SPI read buffer. Especially in SPI DMA operation. 0 = SPI Complete Interrupt Enabled while ESPI and EA are enabled, 1 = SPI Complete Interrupt Disabled
[1]	**TXBFF**	**SPI TX Buffer Full Flag** 0 = SPI TX buffer is empty 1 = SPI TX buffer is full

SPInDR – Serial Peripheral Data Register

Register	SFR Address	Reset Value
SPI0DR	F5H, Page 0	0000_0000 b
SPI1DR	FCH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
SPInDR[7:0]							
R/W							

Bit	Name	Description
[7:0]	**SPInDR[7:0]**	**Serial Peripheral Data** This byte is used for transmitting or receiving data on SPI bus. A write of this byte is a write to the shift register. A read of this byte is actually a read of the read data buffer. In Master mode, a write to this register initiates transmission and reception of a byte simultaneously.

DMAnBAH – PDMAn XRAM Base Address High Byte

Register	SFR Address	Reset Value
DMA0BAH	F6H, Page 0	0000_0000 b
DMA1BAH	FDH, Page 0	0000_0000 b
DMA2BAH	B2H, Page 2	0000_0000 b
DMA3BAH	AAH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
MTMDA[7:4]				XRAMA[7:4]			
R/W				R/W			

Bit	Name	Description
[7:4]	**MTMDA[7:4]**	**Memory to Memory Destination Address (High Byte)** The most significant 4 bits of XRAM address are used for memory to memory destination address. XRAM destination address = {MDAH[3:0], MDAL[7:0]}
[3:0]	**XRAMA[7:4]**	**PDMA XRAM Base Address (High Byte)** The most significant 4 bits of XRAM address to store or read for the peripheral source data; in memory to memory transfer, this register is the destination address. XRAM address = {MAH[3:0], MAL[7:0]}

EIPH0 – Extensive Interrupt Priority High

Register	SFR Address	Reset Value
EIPH0	F7H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
PT2H	**PSPI0H**	**PFBH**	**PWDTH**	**PPWM0H**	**PCAPH**	**PPIH**	**PI2C0H**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**PT2H**	Timer 2 interrupt priority high bit
[6]	**PSPI0H**	SPI0 interrupt priority high bit
[5]	**PFBH**	Fault Brake interrupt priority high bit
[4]	**PWDTH**	WDT interrupt priority high bit
[3]	**PPWM0H**	PWM0 interrupt priority high bit
[2]	**PCAPH**	Input capture interrupt priority high bit
[1]	**PPIH**	Pin interrupt priority high bit
[0]	**PI2C0H**	R/W interrupt priority high bit
Note： EIPH0 is used in combination with the EIP0 to determine the priority of each interrupt source. See Table 6.3-5 Interrupt Priority Level Setting for correct interrupt priority configuration.		

S1CON – Serial Port 1 Control

Register	SFR Address	Reset Value
S1CON	F8H, All pages, Bit addressable	0000_0000 b

7	6	5	4	3	2	1	0
SM0_1/FE_1	**SM1_1**	**SM2_1**	**REN_1**	**TB8_1**	**RB8_1**	**TI_1**	**RI_1**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**SM0_1/FE_1**	**Serial Port 1 Mode Select** SMOD0_1 (T3CON.6) = 0: See Table 6.10-2 Serial Port 1 Mode / baud rate Description for details. SMOD0_1 (T3CON.6) = 1: SM0_1/FE_1 bit is used as frame error (FE) status flag. It is cleared by software. 0 = Frame error (FE) did not occur. 1 = Frame error (FE) occurred and detected.
[6]	**SM1_1**	Check with bit 7 description.
[5]	**SM2_1**	**Multiprocessor Communication Mode Enable** The function of this bit is dependent on the serial port 1 mode. Mode 0: No effect. Mode 1: This bit checks valid stop bit. 0 = Reception is always valid no matter the logic level of stop bit. 1 = Reception is valid only when the received stop bit is logic 1 and the received data matches “Given” or “Broadcast” address. Mode 2 or 3: For multiprocessor communication. 0 = Reception is always valid no matter the logic level of the 9th bit. 1 = Reception is valid only when the received 9th bit is logic 1 and the received data matches “Given” or “Broadcast” address.
[4]	**REN_1**	**Receiving Enable** 0 = Serial port 1 reception Disabled. 1 = Serial port 1 reception Enabled in Mode 1,2, or 3. In Mode 0, reception is initiated by the condition REN_1 = 1 and RI_1 = 0.
[3]	**TB8_1**	**9th Transmitted Bit** This bit defines the state of the 9th transmission bit in serial port 1 Mode 2 or 3. It is not used in Mode 0 or 1.

Bit	Name	Description
[2]	**RB8_1**	**9th Received Bit** The bit identifies the logic level of the 9th received bit in serial port 1 Mode 2 or 3. In Mode 1, RB8_1 is the logic level of the received stop bit. SM2_1 bit as logic 1 has restriction for exception. RB8_1 is not used in Mode 0.
[1]	**TI_1**	**Transmission Interrupt Flag** This flag is set by hardware when a data frame has been transmitted by the serial port 1 after the 8th bit in Mode 0 or the last data bit in other modes. When the serial port 1 interrupt is enabled, setting this bit causes the CPU to execute the serial port 1 interrupt service routine. This bit must be cleared manually via software.
[0]	**RI_1**	**Receiving Interrupt Flag** This flag is set via hardware when a data frame has been received by the serial port 1 after the 8th bit in Mode 0 or after sampling the stop bit in Mode 1, 2, or 3. SM2_1 bit as logic 1 has restriction for exception. When the serial port 1 interrupt is enabled, setting this bit causes the CPU to execute to the serial port 1 interrupt service routine. This bit must be cleared manually via software.

PWM0DTEN – PWM Dead-time Enable (TA Protected)

Register	SFR Address	Reset Value
PWM0DTEN	F9H, Page 1, TA protected	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	PWMnDTCNT.8	-	PDT45EN	PDT23EN	PDT01EN
-	-	-	R/W	-	R/W	R/W	R/W

Bit	Name	Description
[7:5]	0	Reserved
[4]	PWMnDTCNT.8	**PWM Dead-Time Counter Bit 8** See PWMnDTCNT register.
[3]	0	Reserved
[2]	PDT45EN	**PWM4/5 Pair Dead-Time Insertion Enable** This bit is valid only when PWM4/5 is under complementary mode. 0 = No delay on GP4/GP5 pair signals. 1 = Insert dead-time delay on the rising edge of GP4/GP5 pair signals.
[1]	PDT23EN	**PWM2/3 Pair Dead-Time Insertion Enable** This bit is valid only when PWM2/3 is under complementary mode. 0 = No delay on GP2/GP3 pair signals. 1 = Insert dead-time delay on the rising edge of GP2/GP3 pair signals.
[0]	PDT01EN	**PWM0/1 Pair Dead-Time Insertion Enable** This bit is valid only when PWM0/1 is under complementary mode. 0 = No delay on GP0/GP1 pair signals. 1 = Insert dead-time delay on the rising edge of GP0/GP1 pair signals.

LCDCON – LCD Control

Register	SFR Address	Reset Value
LCDCON	F9H, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
LCDEN	**TYPE**	**BIAS[1:0]**		**DUTY[1:0]**		**-**	**LCD_IE**
R/W	R/W	R/W		R/W		-	R/W

Bit	Name	Description
[7]	**LCDEN**	**LCD Enable** 0 = LCD circuit OFF. Each COM and SEG pin functions as general purpose I/O and its multi-functions other than LCD. 1 = LCD circuit ON. COM and enabled SEG pins generate the LCD driving waveform.
[6]	**TYPE**	**Display Type** 0 = Type A 1 = Type B
[5:4]	**BIAS[1:0]**	**LCD Bias** 00 = Reserved. 01 = 1/2 bias. 10 = 1/3 bias. 11 = 1/4 bias
[3:2]	**DUTY[1:0]**	**LCD Duty** 00 = 1/4 duty. 01 = 1/6 duty. 10 = 1/8 duty. 11 = Reserved. Note that when 1/4 duty is selected, only COM0 to COM3 are used for LCD driving. COM4, COM5, COM6, COM7 are not used and functions as a general purpose I/O. When 1/6 duty is selected, only COM0 to COM5 are used for LCD driving.SEG0 and SEG1 are not available. SEG0 and SEG1 pins function are as COM4 and COM5. When 1/8 duty is selected, COM0 to COM7 are all used for LCD driving.SEG0, SEG1, SGE2, SEG3 and SEG4 are not available. SEG0, SEG1, SGE2, SEG3 and SEG4 pins function are as COM4, COM5, COM6 and COM7.
[1]	**-**	Reserved
[0]	**LCD_IE**	**LCD Interrupt Enable for CPUMP Overload Interrupt Enable** 0 = disable 1 = enable

PWM0DTCNT – PWM Dead-time Counter (TA Protected)

Register	SFR Address	Reset Value
PWM0DTCNT	FAH, Page 1, TA protected	0000_0000 b

7	6	5	4	3	2	1	0
PWM0DTCNT[7:0]							
R/W							

Bit	Name	Description
[7:0]	**PWM0DTCNT[7:0]**	**PWM Dead-Time Counter Low Byte** This 8-bit field combined with PWMnDTEN .4 forms a 9-bit PWM dead-time counter PWM0DTCNT. This counter is valid only when PWM is under complementary mode and the correspond PWMnDTEN bit for PWM pair is set. PWM dead-time = $\frac{PDTCNT + 1}{F_{SYS}}$. Note that user should not modify PWM0DTCNT during PWM run time.

LCDCLK – LCD Clock Control

Register	SFR Address	Reset Value
LCDCLK	FAH, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	LCDCKS	DISP	LCDDIV[2:0]		
-	-	-	R/W	R/W	R/W		

Bit	Name	Description
[7:5]	-	Reserved
[4]	LCDCKS	**LCD Clock Source Select** 0 = LIRC/2^4. 1 = LXT/2^4.
[3]	DISP	**DISP** The LCD display keeps display on or display off during chip power-down mode. If LXT is used as the LCD clock source, user should turn on LXT first by software. 0 = Display off. 1 = Display on.
[2:0]	LCDDIV[2:0]	**LCD Clock Divider** 000 = 1/1. 001 = 1/2. 010 = 1/4. 011 = 1/8. 100 = 1/16. 101 = 1/32. Others = Reserved.

PWMxMEN – PWMnCx Mask Enable, n=0,1,2,3;x=0,1,2,3,4,5

Register	SFR Address	Reset Value
PWM0MEN	FBH, Page 1	0000_0000 b
PWM1MEN	8DH, Page 2	0000_0000 b
PWM2MEN	BDH, Page 2	0000_0000 b
PWM3MEN	CDH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	PMEN5	PMEN4	PMEN3	PMEN2	PMEN1	PMEN0
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[5:0]	PMENn	**PWMnCx Mask Enable** 0 = PWMnCx signal outputs from its PWM generator. 1 = PWMnCx signal is masked by PMDx. **Note:** PMEN2~5 are only for PWM0.

LCDPTR – LCD Data Pointer

Register	SFR Address	Reset Value
LCDPTR	FBH, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	LCDPTR[4:0]				
-	-	-	R/W				

Bit	Name	Description
[7:5]	-	Reserved
[4:0]	LCDPTR[4:0]	**LCD Data Pointer** This field determines which LCD display data register is accessed by LCDDAT. User should fill the target pointer value in LCDPTR before accessing LCDDAT. After LCD display data is written to LCDDAT register, the LCDPTR value increases one automatically.

PWMnMD – PWM Mask Data

Register	SFR Address	Reset Value
PWM0MD	FCH, Page 1	0000_0000 b
PWM1MD	8CH, Page 2	0000_0000 b
PWM2MD	BCH, Page 2	0000_0000 b
PWM3MD	CCH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
-	**-**	**PMD5**	**PMD4**	**PMD3**	**PMD2**	**PMD1**	**PMD0**
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:6]	**0**	**Reserved** The bits are always read as 0.
[5:0]	**PMDx**	**PWMnCx Mask Data** The PWMnCx signal outputs mask data once its corresponding PMENx is set. 0 = PWMnCx signal is masked by 0. 1 = PWMnCx signal is masked by 1. **Note:** PMD2~5 are only for PWM0.

LCDDAT – LCD Data

Register	SFR Address	Reset Value
LCDDAT	FCH, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
LCDDAT[7:0]							
R/W							

Bit	Name	Description
[7:0]	**LCDDAT[7:0]**	**LCD Data** The value written into this register will be displayed to the corresponding LCD SEG and COM pins pointed by LCDPTR. 0 = LCD pixel is cleared. 1 = LCD pixel is darkened.

LVRFLTEN – LVR Filter Enable (TA Protected)

Register	SFR Address	Reset Value
LVRFLTEN	FDH, Page 1, TA protected	0000_0000 b

7	6	5	4	3	2	1	0
LVRFLTEN[7:0]							
W							

Address: FDH, Page 1 reset value: 0000 0000b

Bit	Name	Description
[7:0]	**LVRFLTEN[7:0]**	**LVR18 Filter Enable** To first writing 5AH to the LVRFLTEN and immediately followed by a writing of A5H. Others = Disabled.

LCDPWR – LCD Power Saving Mode

Register	SFR Address	Reset Value
LCDPWR	FDH, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	-	PWR_SAVING[1:0]	
-	-	-	-	-	-	R/W	

Bit	Name	Description
[7:2]	-	Reserved
[1:0]	PWR_SAVING[1:0]	**LCD_PWR_SAVING** LCD driving cycle select, turn on timing decide the driving current. 00 = always ON. No power saving 01 = Turns on 1/4 cycle 10 = Turns on 2/4 cycle 11 = Turns on 3/4 cycle

EIP1 – Extensive Interrupt Priority 1

Register	SFR Address	Reset Value
EIP1	FEH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
PSPI1	**PDMA1**	**PDMA0**	**PSMC**	**PHF**	**PWKT**	**PT3**	**PS1**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**PSPI1**	SPI1 interrupt priority low bit
[6]	**PDMA1**	PDMA1 interrupt priority low bit
[5]	**PDMA0**	PDMA0 interrupt priority low bit
[4]	**PSMC**	SMC interrupt priority low bit
[3]	**PHF**	Hard fault interrupt priority low bit
[2]	**PWKT**	WKT interrupt priority low bit
[1]	**PT3**	Timer 3 interrupt priority low bit
[0]	**PS1**	Serial port 1 interrupt priority low bit
Note: EIP1 is used in combination with the EIPH1 to determine the priority of each interrupt source. See Table 6.3-5 Interrupt Priority Level Setting for correct interrupt priority configuration.		

LCDBL – LCD Blink

Register	SFR Address	Reset Value
LCDBL	FEH, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-				BLINK	BLF[2:0]		
-				R/W	R/W		

Bit	Name	Description
[7:4]	-	Reserved
[3]	BLINK	**LCD BLINK** 0 = LCD always on 1 = LCD blink
[2:0]	BLF[2:0]	**BLINK Frequency** LCDCKS[1:0] = 00 FBLINK = FLXT/2(14+ BL_Time[2:0]) LCDCKS[1:0] = 01 FBLINK = FLIRC/2(14+ BL_Time[2:0])

EIPH1 – Extensive Interrupt Priority High 1

Register	SFR Address	Reset Value
EIPH1	FFH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
PSPI1H	**PDMA1H**	**PDMA0H**	**PSMCH**	**PHFH**	**PWKTH**	**PT3H**	**PS1H**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**PSPI1H**	SPI1 interrupt priority high bit
[6]	**PDMA1H**	PDMA1 interrupt priority high bit
[5]	**PDMA0H**	PDMA0 interrupt priority high bit
[4]	**PSMCH**	SMC interrupt priority high bit
[3]	**PHFH**	Hard fault interrupt priority high bit
[2]	**PWKTH**	WKT interrupt priority high bit
[1]	**PT3H**	Timer 3 interrupt priority high bit
[0]	**PS1H**	Serial port 1 interrupt priority high bit
Note: EIPH1 is used in combination with the EIP1 to determine the priority of each interrupt source. See Table 6.3-5 Interrupt Priority Level Setting for correct interrupt priority configuration.		

LVRDIS – LVR Disable (TA Protected)

Register	SFR Address	Reset Value
LVRDIS	FFH, Page 0, TA protected	0000_0000 b

7	6	5	4	3	2	1	0
LVRDIS[7:0]							
W							

Bit	Name	Description
[7:0]	**LVRDIS[7:0]**	**LVR Disable** To first writing 5AH to the LVRDIS and immediately followed by a writing of A5H will disable LVR.

LCDMODE – LCD Resister Mode

Register	SFR Address	Reset Value
LCDMODE	FFH, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
R_MODE	**BUF_EN**	**-**				**VLCD_MODE[1:0]**	
R/W	R/W	-				R/W	

Bit	Name	Description
[7]	**R_MODE**	**Resister Mode** 0 = LCD none resister mode 1 = LCD resister mode
[6]	**BUF_EN**	**Buffer Enable** 0 = buffer off 1 = buffer on
[5:2]	**-**	Reserved
[1:0]	**VLCD_MODE[1:0]**	**Package with VLCD Pin (ROMMAP Control Bit = 1)** This bits defined VLCD voltage source VLCD_MODE / VLCD source: 00 – Ext. VLCD 01 – AV_{DD} 10 – VCP (Charge Pump) 11 – Disable Note : VCP value base on LCDCPUMPdefine.

6.3 System Manager

6.3.1 Clock System

The ML51/ML54/ML56 Series has a wide variety of clock sources and selection features that allow it to be used in a wide range of applications while maximizing performance and minimizing power consumption. The ML51/ML54/ML56 Series provides five options of the system clock sources including internal oscillator, crystal/resonator, or external clock from XIN pin via software. The ML51/ML54/ML56 Series is embedded with two internal oscillators: one 38.4 R/W low-speed and one 24 R/W high-speed, which is factory trimmed to ±2% under all conditions. A clock divider CKDIV is also available on ML51/ML54/ML56 Series for adjustment of the flexibility between power consumption and operating performance.

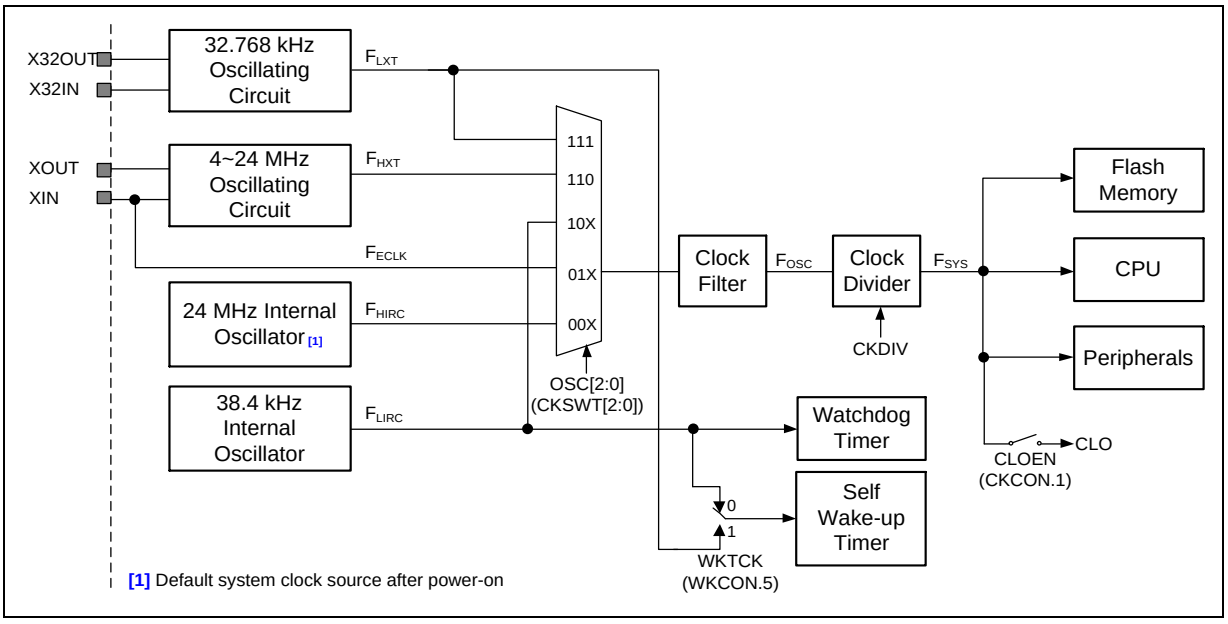


Figure 6.3-1 Clock System Block Diagram

6.3.1.1 System Clock Sources

There are a total of three system clock sources selectable in the ML51/ML54/ML56 Series including high-speed internal oscillator, low-speed internal oscillator and external clock input. Each of them can be the system clock source in the ML51/ML54/ML56 Series. Different active system clock sources also affect multi-function of P5.3/XIN, P5.2/XOUT and P5.5/X32IN, P5.4/X32OUT pins.

6.3.1.2 Internal Oscillators

There are two internal oscillators in the ML51/ML54/ML56 Series – one 24 R/W high-speed internal oscillator (HIRC) and one 38.4 R/W low-speed (LIRC). Both of them can be selected as the system clock. HIRC can be enabled by setting HIRCEN (CKEN.5). LIRC is enabled after device is powered up. User can set OSC[2:0] (CKSWT [2:0]) as [0,0,x] to select the HIRC as the system clock. By setting

OSC[2:0] as [1,0,x], LIRC will be selected as the system clock. Note that after the ML51/ML54/ML56 Series is powered, HIRC and LIRC will be both enabled and HIRC is default selected as the system clock source. While using internal oscillators, XIN, XOUT, X32I and X32O automatically switch as one general purpose I/O to expend the number of general purpose I/O.

6.3.1.3 External Crystal/Resonator or Clock Input

There are three possible clock source options of external clock sources – 4 R/W to 24 R/W high-speed crystal/resonator (HXT), 32.768 R/W low-speed crystal/resonator (LXT), and the external clock input (ECLK) through XIN pin. User can set OSC[2:0] as [0,1,x] to select ECLK as the system clock. By setting OSC[2:0] as [1,1,0], HXT will be selected as the system clock. By setting OSC[2:0] as [1,1,1], LXT will be selected as the system clock. When HXT or LXT is used as the system clock, X_{IN}/ $X32_{IN}$ and X_{OUT}/ $X32_{OUT}$ are the input and output, respectively, of an internal inverting amplifier. A crystal or resonator should be connected between XIN/ $X32_{IN}$ and XOUT/$X32O_{UT}$ pins. When enabling and selecting ECLK as the system clock source, the system clock is supplied via the XIN pin. The common application is to drive XIN with an active oscillator or clocks from another host device. Be aware that user should never feed any clock signal larger than voltage 1.8V to XIN and give a DC voltage to X_{OUT} pin which value is half of XIN, when ECLK mode is selected. Otherwise, it may break the device.

XLTCON – XLT Clock Control (TA Protected)

Register	SFR Address	Reset Value
XLTCON	85H, Page 1, TA protected	0111_0111b

7	6	5	4	3	2	1	0
HSCH	**HXSG[6:4]**			**-**	**-**	**LXSG[1:0]**	
R/W	R/W			-	-	R/W	

Bit	Name	Description
[7]	**HSCH**	**HXT Schmitt Trigger Select** 0 = disable 1 = enable
[6:4]	**HXSG[6:4]**	**HXT Gain Value Select** 000 = L0 mode (smallest value) 001 = L1 mode 010 = L2 mode 011 = L3 mode 100 = L4 mode 101 = L5 mode 110 = L6 mode 111 = L7 mode (largest value)
[3:2]	**-**	Reserved
[1:0]	**LXSG[1:0]**	**LXT Gain Value Select** 00 = L0 mode (smallest value) 01 = L1 mode 10 = L2 mode 11 = L3 mode (largest value)

6.3.1.4 System Clock Switching

The ML51/ML54/ML56 Series supports clock source switching on-the-fly by controlling CKSWT and CKEN registers via software. It provides a wide flexibility in application. Note that these SFR are writing TA protected for precaution. With this clock source control, the clock source can be switched between the external clock source and the internal oscillator, even between the high and low-speed internal oscillator. However, during clock source switching, the device requires some amount of warm-up period for an original disabled clock source. Therefore, use should follow steps below to ensure a complete clock source switching. User can enable the target clock source by writing proper value into CKEN register, wait for the clock source stable by polling its status bit in CKSWT register, and switch to the target clock source by changing OSC[2:0] (CKSWT[2:0]). After these step, the clock source switching is successful and then user can also disable the original clock source if power consumption is concerned.

Note that if not following the steps above, the hardware will take certain actions to deal with such illegal operations as follows.

1. If user tries to disable the current clock source by changing CKEN value, the device will ignore this action. The system clock will remain the original one and CKEN will remain the original value.

2. If user tries to switch the system clock source to a disabled one by changing OSC[2:0] value, OSC[2:0] value will be updated right away. But the system clock will remain the original one and CKSWTF flag will be set by hardware.

3. Once user switches the system clock source to an enabled but still instable one, the hardware will wait for stabilization of the target clock source and then switch to it in the background. During this waiting period, the device will continue executing the program with the original clock source and CKSWTF will be set as 1. After the stable flag of the target clock source (see CKSWT[7:3]) is set and the clock source switches successfully, CKSWTF will be cleared as 0 automatically by hardware.

CKSWT – Clock Switch (TA Protected)

Register	SFR Address	Reset Value
CKSWT	96H, PAGE 0, TA protected	0011 _0000 b

7	6	5	4	3	2	1	0
HXTST	**LXTST**	**HIRCST**	**LIRCST**	**ECLKST**	**OSC[2:0]**		
R	R	R	R	R	W		

Address: 96H, PAGE 0 Reset value: 0011 0000b

Bit	Name	Description
[7]	**HXTST**	**High Speed External Crystal/Resonator 4 R/W to 24 R/W Status** 0 = High speed external crystal/resonator is not stable or is disabled. 1 = High speed external crystal/resonator is enabled and stable.
[6]	**LXTST**	**Low Speed External Crystal/Resonator 32.768 R/W Status** 0 = Low speed external crystal/resonator is not stable or is disabled. 1 = Low speed external crystal/resonator is enabled and stable.
[5]	**HIRCST**	**High-Speed Internal Oscillator 24 R/W Status** 0 = High-speed internal oscillator is not stable or disabled. 1 = High-speed internal oscillator is enabled and stable.
[4]	**LIRCST**	**Low Speed Internal Oscillator 38.4 R/W Status** 0 = Low speed internal oscillator is not stable or is disabled. 1 = Low speed internal oscillator is enabled and stable.
[3]	**ECLKST**	**External Clock Input Status** 0 = External clock input is not stable or disabled. 1 = External clock input is enabled and stable.
[2:0]	**OSC[2:0]**	**Oscillator Selection Bits** This field selects the system clock source. 00x = Internal 24 R/W oscillator. Defaul value accoding to HIRCEN(CKEN.5) enabled. 01x = External oscillator clock source accoding to ECLKEN(CKEN.3) enabled. 10x = Internal 38.4 R/W oscillator according to LIRCEN(CKEN.4) enabled. 110 = External High speed crystal/resonator clock source (4 R/W ~ 24 R/W) accoding to EHXTEN(CKEN.7) enabled. 111 = External Low speed crystal/resonator clock source (32.768 R/W) accoding to ELXTEN(CKEN.6) enabled. Note that this field is write only. The read back value of this field may not correspond to the present system clock source.

CKEN – Clock Enable (TA Protected)

Register	SFR Address	Reset Value
CKEN	97H, PAGE 0, TA protected	0011_0100 b

7	6	5	4	3	2	1	0
EHXTEN	**ELXTEN**	**HIRCEN**	**LIRCEN**	**ECLKEN**	**-**		**CKSWTF**
R/W	R/W	R/W	R/W	R/W	-		R

Address: 97H, Page 0 Reset value: 0011 0100b

Bit	Name	Description
[7]	**EHXTEN**	**External High-Speed Crystal/Resonator Enable** 1 = High-speed external crystal/resonator 4 R/W to 24 R/W Enabled. 0 = High-speed external crystal/resonator 4 R/W to 24 R/W Disabled, P5.2 and P5.3 work as general purpose I/O or other functions if ECLKEN set to 0.
[6]	**ELXTEN**	**External Low-Speed Crystal/Resonator Enable** 1 = Low-speed external crystal/resonator 32.768 R/W Enabled. 0 = Low-speed external crystal/resonator 32.768 R/W Disabled, P5.4 and P5.5 work as general purpose I/O or other functions.
[5]	**HIRCEN**	**High-Speed Internal Oscillator 24 R/W Enable** 0 = The high-speed internal oscillator Disabled. 1 = The high-speed internal oscillator Enabled. Note that once IAP is enabled by setting IAPEN (CHPCON.0), the high-speed internal 24 R/W oscillator will be enabled automatically. The hardware will also set HIRCEN and HIRCST bits. After IAPEN is cleared, HIRCEN and EHRCST resume the original values.
[4]	**LIRCEN**	**Low Speed Internal Oscillator 38.4 R/W Enable** 0 = The low speed internal oscillator Disabled. 1 = The low speed internal oscillator Enabled. Note that when (1)WDT is enabled, (2)WKT is running by the clock source of the internal 38.4 R/W oscillator ,(3) BOD is enabled, or (4)LVR filter is enabled, a write 0 to LIRCEN will be ignored. LIRCEN is always 1 and the internal 38.4 R/W oscillator is always enabled.
[3]	**ECLKEN**	**External Clock Input Enable** 1 = External clock input (XIN , P5.3) Enabled. 0 = External clock input (XIN, P5.3) Disabled, P5.2 and P5.3 work as general purpose I/O or other functions if EHXTEN set to 0.
[2:1]	**-**	Reserved
[0]	**CKSWTF**	**Clock Switch Fault Flag** 0 = The previous system clock source switch was successful. 1 = User tried to switch to an instable or disabled clock source at the previous system clock source switch. If switching to an instable clock source, this bit remains 1 until the clock source is stable and switching is successful.

6.3.1.5 System Clock Divider

The oscillator frequency (F_{OSC}) can be divided down, by an integer, up to 1/510 by configuring a dividing register, CKDIV, to provide the system clock (F_{SYS}). This feature makes it possible to temporarily run the MCU at a lower rate, reducing power consumption. By dividing the clock, the MCU can retain the ability to respond to events other than those that can cause interrupts (i.e. events that allow exiting the Idle mode) by executing its normal program at a lower rate. This can often result in lower power consumption than in Idle mode. This can allow bypassing the oscillator start-up time in cases where Power-down mode would otherwise be used. The value of CKDIV may be changed by the program at any time without interrupting code execution.

CKDIV – Clock Divider

Register	SFR Address	Reset Value
CKDIV	C1H, Page 1	0000_0000b

7	6	5	4	3	2	1	0
CKDIV[7:0]							
R/W							

Bit	Name	Description
[7:0]	**CKDIV[7:0]**	**Clock Divider** The system clock frequency FSYS follows the equation below according to CKDIV value. $F_{SYS} = F_{OSC}$, while CKDIV = 00H, and $F_{SYS} = \frac{F_{OSC}}{2 \times CKDIV}$, while CKDIV = 01H to FFH.

6.3.1.6 System Clock Output

The ML51/ML54/ML56 Series provides a CLO pin that outputs the system clock. Its frequency is the same as F_{SYS}. The output enable bit is CLOEN (CKCON.1). CLO output stops when device is put in its Power-down mode because the system clock is turned off. Note that when noise problem or power consumption is important issue, user had better not enable CLO output.

CKCON – Clock Control

Register	SFR Address	Reset Value
CKCON	8EH, Page 0	1000_0000b

7	6	5	4	3	2	1	0
FASTWK	**PWMCKS**	**T1OE**	**T1M**	**T0M**	**T0OE**	**CLOEN**	**-**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-

Address: 8EH, Page 0 Reset value: 1000 0000b

Bit	Name	Description
[1]	**CLOEN**	**System Clock Output Enable** 0 = System clock output Disabled. 1 = System clock output Enabled from CLO pin. Once system clock output was enabled, only POR/BOD reset can disable it.

6.3.2 Power Management

The ML51/ML54/ML56 Series has several features that help user to control the power consumption of the device. Table 6.3-1 Table 6.3-1 Power Mode Tablelists all power mode at ML51/ML54/ML56 Series to save the power consumption. For a stable current consumption, the state and mode of each pin should be taken care of. The minimum power consumption can be attained by giving the pin state just the same as the external pulls for example output 1 if pull-high is used or output 0 if pull-low. If the I/O pin is floating, user is recommended to leave it as quasi-bidirectional mode.

Mode	Clock Source	Comment
Normal mode	Any clock source	-
Idle mode	Any clock source	Only CPU clock is stoped.
Low power run mode	Only for LIRC or LXT	-
Low power idle mode	Only for LIRC or LXT	Only CPU clock is stoped.
Power-down mode (PD)	Any clock source	1. CPU enters deep sleep mode 2. Most clocks are disabled except ACMP/LIRC/LXT, and only WDT/WKT peripheral clocks still enable if their clock sources are selected as LIRC/LXT.

Table 6.3-1 Power Mode Table

For each power mode, they have different entry setting and leaving condition. The Table 6.3-2 shows the entry setting for each power mode. When chip power-on, chip is running in normal mode. User can enter each mode by setting IDL (PCON.0), LPR (PCON.5) and PD (PCON.1).

Register/Instruction Mode	LPR (PCON.5)	PD (PCON.1)	IDL (PCON.0)
Normal mode	0	0	0
Idle mode	0	0	1
Low power run mode	1	0	0
Low power idle mode	1	0	1
Power-down mode	X	1	X

Table 6.3-2 Entry setting of Power-down mode

PCON – Power Control

Register	SFR Address	Reset Value
PCON	87H, All pages	POR: 0001_0000b Others: 000U _0000b

7	6	5	4	3	2	1	0
SMOD	**SMOD0**	**LPR**	**POF**	**GF1**	**GF0**	**PD**	**IDL**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address: 87H, All pagess POR reset value: 0001 000b, other reset value: 000U 0000b

Bit	Name	Description
[5]	**LPR**	**Low Power Run Mode** 0 = disable 1 = enable **Note:** If PD = 1 and LPR = 1 at the same time, LPR is invalid, CPU will enter Power-down mode.
[1]	**PD**	**Power-Down Mode** Setting this bit puts CPU into Power-down mode. Under this mode, both CPU and peripheral clocks stop and Program Counter (PC) suspends. It provides the lowest power consumption. After CPU is woken up from Power-down, this bit will be automatically cleared via hardware and the program continue executing the interrupt service routine (ISR) of the very interrupt source that woke the system up before. After return from the ISR, the device continues execution at the instruction, which follows the instruction that put the system into Power-down mode. Note that If IDL bit and PD bit are set simultaneously, CPU will enter Power-down mode. Then it does not go to Idle mode after exiting Power-down.
[0]	**IDL**	**Idle Mode** Setting this bit puts CPU into Idle mode. Under this mode, Program Counter (PC) suspends but the CPU clock keep running and all peripherals keep activated. After CPU is woken up from Idle, this bit will be automatically cleared via hardware and the program continue executing the ISR of the very interrupt source that woke the system up before. After return from the ISR, the device continues execution at the instruction which follows the instruction that put the system into Idle mode.

6.3.2.1 Idle Mode

Idle mode suspends CPU processing by holding the Program Counter. No program code are fetched and run in Idle mode. It forces the CPU state to be frozen. The Program Counter (PC), Stack Pointer (SP), Program Status Word (PSW), Accumulator (ACC), and the other registers hold their contents during Idle mode. The port pins hold the logical states they had at the time Idle was activated. Generally, it saves considerable power of typical half of the full operating power.

Since the clock provided for peripheral function logic circuit like timer or serial port still remain in Idle mode, the CPU can be released from the Idle mode with any of enabled interrupt sources. User can put the device into Idle mode by writing 1 to the bit IDL (PCON.0). The instruction that sets the IDL bit is the last instruction that will be executed before the device enters Idle mode.

The Idle mode can be terminated in two ways. First, as mentioned, any enabled interrupt will cause an exit. It will automatically clear the IDL bit, terminate Idle mode, and the interrupt service routine (ISR) will be executed. After using the RETI instruction to jump out of the ISR, execution of the program will be the one following the instruction, which put the CPU into Idle mode. The second way to terminate Idle mode is with any reset other than software reset. Remember that if Watchdog reset is used to exit Idle mode, the WIDPD (WDCON.4) needs to be set 1 to let WDT keep running in Idle mode.

6.3.2.2 Low Power Run Mode

The CPU and the selected peripherals are running with a low speed oscillator (LXT or LIRC). At first system clock should be switch to LXT or LIRC. And then put the device into Low power run mode by writing 1 to the bit LPR (PCON.5) .

The voltage regulator is configured in ultra-low-power mode. The microcontroller enters Low power run mode by software and can exit from this mode by software or by a reset.

6.3.2.3 Low Power Idle Mode

Only the device is in Low power run mode, user can put into Low power idle mode by writing 1 to bit IDL (PCON.0).

6.3.2.4 R/W Mode

Power-down mode is the lowest power state that the ML51/ML54/ML56 Series can enter. It remains the power consumption as A ”uA” level by stopping the system clock source. Both of CPU and peripheral functions like Timers or UART are frozen. Flash memory is put into its stop mode. All activity is completely stopped and the power consumption is reduced to the lowest possible value. The device can be put into Power-down mode by writing 1 to bit PD (PCON.1). The instruction that does this action will be the last instruction to be executed before the device enters Power-down mode. In the Power-down mode, RAM maintains its content. The port pins output the values held by their own state before Power-down respectively.

There are several ways to exit the ML51/ML54/ML56 Series from the Power-down mode. The first is with all resets except software reset. Brown-out reset will also wake up CPU from Power-down mode. Be sure that brown-out detection is enabled before the system enters Power-down. However, for least power consumption, it is recommended to enable low power BOD in Power-down mode. Of course the external pin reset and power-on reset will remove the Power-down status. After the external reset or power-on reset. The CPU is initialized and start executing program code from the beginning.

The second way to wake the ML51/ML54/ML56 Series up from the Power-down mode is by an enabled external interrupt. The trigger on the external pin will asynchronously restart the system clock. After oscillator is stable, the device executes the interrupt service routine (ISR) for the corresponding external interrupt. After the ISR is completed, the program execution returns to the instruction after the one, which puts the device into Power-down mode and continues. Interrupts that allows to wake up CPU from Power-down mode includes external interrupt INT0 and INT1, pin interrupt, WDT interrupt, WKT interrupt, and brown-out interrupt.

6.3.3 Power Monitering and Reset

The ML51/ML54/ML56 Series has several options to place device in reset condition. It also offers the software flags to indicate the source, which causes a reset. In general, most SFR go to their Reset value irrespective of the reset condition, but there are several reset source indicating flags whose state depends on the source of reset. User can read back these flags to determine the cause of reset using software. There are five ways of putting the device into reset state. They are power-on reset, brown-out reset, external reset, WDT reset, and software reset.

6.3.3.1 Power-On Rese（POR）t and Low Voltage Reset (LVR)

The ML51/ML54/ML56 Series incorporates an internal power-on reset (POR) and a low voltage reset (LVR). During a power-on process of rising power supply voltage V_{DD}, the POR or LVR will hold the MCU in reset mode when V_{DD} is lower than the voltage reference thresholds. This design makes CPU not access program Flash while the V_{DD} is not adequate performing the Flash reading. If an undetermined operating code is read from the program Flash and executed, this will put CPU and even the whole system in to an erroneous state. After a while, V_{DD} rises above the threshold where the system can work, the selected oscillator will start and then program code will execute from 0000H. At the same time, a power-on flag POF (PCON.4) will be set 1 to indicate a cold reset, a power-on process complete. Note that the contents of internal RAM will be undetermined after a power-on. It is recommended that user gives initial values for the RAM block.

The POF is recommended to be cleared to 0 via software to check if a cold reset or warm reset performed after the next reset occurs. If a cold reset caused by power off and on, POF will be set 1 again. If the reset is a warm reset caused by other reset sources, POF will remain 0. User may take a different course to check other reset flags and deal with the warm reset event. For detailed electrical characteristics.

PCON – Power Control

Register	SFR Address	Reset Value
PCON	87H, All pages	POR: 0001_0000b Others: 000U _0000b

7	6	5	4	3	2	1	0
SMOD	**SMOD0**	**LPR**	**POF**	**GF1**	**GF0**	**PD**	**IDL**
R/W	R/W	RW	R/W	R/W	R/W	R/W	R/W

Address: 87H, All pagess POR reset value: 0001 000b, other reset value: 000U 0000b

Bit	Name	Description
[4]	**POF**	**Power-on Reset Flag** This bit will be set as 1 after a power-on reset. It indicates a cold reset, a power-on reset complete. This bit remains its value after any other resets. This flag is recommended to be cleared via software.

6.3.3.2 Brown-Out Reset (BOR)

The other power monitoring function brown-out detection (BOD) circuit is used for monitoring the V_{DD} level during execution. There are eight CONFIG selectable brown-out trigger levels available for wide voltage applications. These eight nominal levels are 1.8V, 2.0V, 2.4V, 2.7V, 3.0V, 3.7V and 4.4V selected via setting CBOV[2:0] (CONFIG2[6:4]). BOD level can also be changed by setting BOV[2:0] (BODCON0[6:4]) after power-on. When V_{DD} drops to the selected brown-out trigger level (V_{BOD}), the BOD logic will either reset the MCU or request a brown-out interrupt. User may decide to being reset or generating a brown-out interrupt according to different applications. V_{BOD} also can be set by software after power-on. Note that BOD output is not available until 2~3 LIRC clocks after software enabling.

The BOD will request the interrupt while V_{DD} drops below V_{BOD} while BORST (BODCON0.2) is 0. In this case, BOF (BODCON0.3) will be set as 1. After user cleared this flag whereas V_{DD} remains below V_{BOD}, BOF will not set again. BOF just acknowledge user a power drop occurs. The BOF will also be set as 1 after V_{DD} goes higher than V_{BOD} to indicate a power resuming. The BOD circuit provides an useful status indicator BOS (BODCON0.0), which is helpful to tell a brown-out event or power resuming event occurrence. If the BORST bit is set as 1, this will enable brown-out reset function. After a brown-out reset, BORF (BODCON0.1) will be set as 1 via hardware. It will not be altered by reset other than power-on. This bit can be cleared by software. Note that all bits in BODCON0 is writing protected by timed access (TA).

CBODEN (CONFIG2.7)	CBORST (CONFIG2.2)	V_{DD} Level	BOF
1	1	> VBOD always	0
1	0	< VBOD	1
1	0	> VBOD	0
0	X	X	0

Table 6.3-3 BOF Reset Value

The ML51/ML54/ML56 Series provides low power BOD mode for saving current consumption and remaining BOD functionality with limited detection response. By setting LPBOD[1:0] (BODCON1[2:1]), the BOD circuit can be periodically enabled to sense the power voltage nominally every 1.6 ms, 6.4 ms, or 25.6 ms. It saves much power but also provides low-speed power voltage sensing. Note that the hysteresis feature will disappear in low power BOD mode.

For a noise sensitive system, the ML51/ML54/ML56 Series has a BOD filter which filters the power noise to avoid BOD event triggering unconsciously. The BOD filter is enabled by default and can be disabled by setting BODFLT (BODCON1.0) as 0 if user requires a rapid BOD response. The minimum brown-out detect pulse width is listed in LPBOD[1:0] (BODCON1)

Note: Since each level of BOD voltage is selectable no matter ML51 series V_{DD} range with 5.5V or 3.6V. The BOD enabled voltage value should be select base on the V_{DD}.

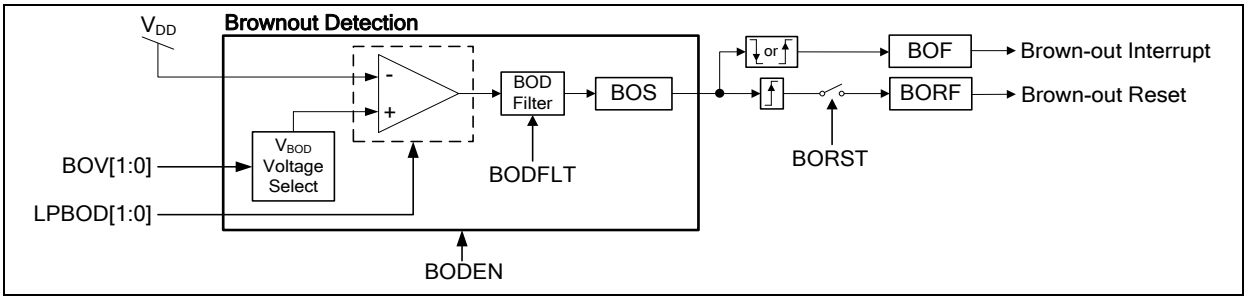


Figure 6.3-2 Brown-out Detection Block Diagram

CONFIG2

7	6	5	4	3	2	1	0
CBODEN	CBOV[2:0]			BOIAP	CBORST	-	-
R/W	R/W			R/W	R/W	-	-

Factory default value: 1111 1111b

Bit	Name	Description
[7]	CBODEN	**CONFIG Brown-Out Detect Enable** 1 = Brown-out detection circuit OFF. 0 = Brown-out detection circuit ON.
[6:4]	CBOV[2:0]	**CONFIG Brown-Out Voltage Select** 111 = V_{BOD} is 1.8V. 110 = V_{BOD} is 1.8V. 101 = V_{BOD} is 2.0V. 100 = V_{BOD} is 2.4V. 011 = V_{BOD} is 2.7V. 010 = V_{BOD} is 3.0V. 001 = V_{BOD} is 3.7V. 000 = V_{BOD} is 4.4V.
[3]	BOIAP	**Brown-Out Inhibiting IAP** This bit decides whether IAP erasing or programming is inhibited by brown-out status. This bit is valid only when brown-out detection is enabled. 1 = IAP erasing or programming is inhibited if V_{DD} is lower than V_{BOD}. 0 = IAP erasing or programming is allowed under any workable V_{DD}.
[2]	CBORST	**CONFIG Brown-Out Reset Enable** This bit decides whether a brown-out reset is caused by a power drop below V_{BOD}. 1 = Brown-out reset Enabled. 0 = Brown-out reset Disabled.

BODCON0 – Brown-out Detection Control 0 (TA Protected)

Register	SFR Address	Reset Value
BODCON0	A3H, Page 0, TA protected	POR,CCCC XC0X b BOD, UUUU XU1X b Others,UUUU XUUX b

7	6	5	4	3	2	1	0
BODEN[1]	**BOV[2:0]**[1]			**BOF**[2]	**BORST**[1]	**BORF**	**BOS**
R/W	R/W			R/W	R/W	R/W	R

Bit	Name	Description
[7]	**BODEN**	**Brown-Out Detection Enable** 0 = Brown-out detection circuit ON. 1 = Brown-out detection circuit OFF. Note that BOD output is not available until 2~3 LIRC clocks after enabling.
[6:4]	**BOV[2:0]**	**CONFIG Brown-Out Voltage Select** 111 = VBOD is 1.8V. 110 = VBOD is 1.8V. 101 = VBOD is 2.0V. 100 = VBOD is 2.4V. 011 = VBOD is 2.7V. 010 = VBOD is 3.0V. 001 = VBOD is 3.7V. 000 = VBOD is 4.4V.
[3]	**BOF**	**Brown-Out Interrupt Flag** This flag will be set as logic 1 via hardware after a V_{DD} dropping below or rising above VBOD event occurs. If both EBOD (I.E.5) and EA (I.E.7) are set, a brown-out interrupt requirement will be generated. This bit should be cleared via software.
[2]	**BORST**	**Brown-Out Reset Enable** This bit decides whether a brown-out reset is caused by a power drop below VBOD. 0 = Brown-out reset when V_{DD} drops below VBOD Disabled. 1 = Brown-out reset when V_{DD} drops below VBOD Enabled.
[1]	**BORF**	**Brown-Out Reset Flag** When the MCU is reset by brown-out event, this bit will be set via hardware. This flag is recommended to be cleared via software.
[0]	**BOS**	**Brown-Out Status** This bit indicates the V_{DD} voltage level comparing with VBOD while BOD circuit is enabled. It keeps 0 if BOD is not enabled. 0 = V_{DD} voltage level is higher than VBOD or BOD is disabled. 1 = V_{DD} voltage level is lower than VBOD. Note that this bit is read-only.

Bit	Name	Description
Note: 1.BODEN, BOV[2:0], and BORST are initialized by being directly loaded from CONFIG2 bit 7, [6:4], and 2 after all resets. 2. BOF reset value depends on different setting of CONFIG2 and V_{DD} voltage level.		

BODCON1 – Brown-out Detection Control Byte 1 (TA Protected)

Register	SFR Address	Reset Value
BODCON1	ABH, Page 0, TA protected	POR 0000 0001 b Others 0000 0UUU b

7	6	5	4	3	2	1	0
-	-	-	-	-	LPBOD[1:0]		BODFLT
-	-	-	-	-	R/W		R/W

Address: ABH, Page 0 Reset value: POR: 0000 0001b / Others:0000 0UUUb

Bit	Name	Description
[7:3]	-	Reserved
[2:1]	LPBOD[1:0]	**Low Power BOD Enable** 00 = BOD normal mode. BOD circuit is always enabled. 01 = BOD low power mode 1 by turning on BOD circuit every 1.6 ms periodically. 10 = BOD low power mode 2 by turning on BOD circuit every 6.4 ms periodically. 11 = BOD low power mode 3 by turning on BOD circuit every 25.6 ms periodically.
[0]	BODFLT	**BOD Filter Control** BOD has a filter which counts 32 clocks of F_{SYS} to filter the power noise when MCU runs with HIRC, or ECLK as the system clock and BOD does not operates in its low power mode (LPBOD[1:0] = [0, 0]). In other conditions, the filter counts 2 clocks of LIRC. Note that when CPU is halted in Power-down mode. The BOD output is permanently filtered by 2 clocks of LIRC. The BOD filter avoids the power noise to trigger BOD event. This bit controls BOD filter enabled or disabled. 0 = BOD filter Disabled. 1 = BOD filter Enabled. (Power-on reset default value.)

6.3.3.3 External Reset and Hard Fault Reset

The external reset pin $\overline{\text{RST}}$ is an input with a Schmitt trigger. An external reset is accomplished by holding the $\overline{\text{RST}}$ pin low for at least 24 system clock cycles to ensure detection of a valid hardware reset signal. The reset circuitry then synchronously applies the internal reset signal. Thus, the reset is a synchronous operation and requires the clock to be running to cause an external reset.

Once the device is in reset condition, it will remain as long as $\overline{\text{RST}}$ pin is low. After the $\overline{\text{RST}}$ high is removed, the MCU will exit the reset state and begin code executing from address 0000H. If an external reset applies while CPU is in Power-down mode, the way to trigger a hardware reset is slightly different. Since the Power-down mode stops system clock, the reset signal will asynchronously cause the system clock resuming. After the system clock is stable, MCU will enter the reset state.

There is a RSTPINF (AUXR0.6) flag, which indicates an external reset took place. After the external reset, this bit will be set as 1 via hardware. RSTPINF will not change after any reset other than a power-on reset or the external reset itself. This bit can be cleared via software.

Hard Fault reset will occur if CPU fetches instruction address over Flash size, HardF (AUXR0.5) flag will be set via hardware. HardF will not change after any reset other than a power-on reset or the external reset itself. This bit can be cleared via software. If MCU run in OCD debug mode and OCDEN = 0, hard fault reset will be disabled. Only HardF flag be asserted.

AUXR0 – Auxiliary Register 0

Register	SFR Address	Reset Value
AUXR0	A2H, Page 0	POR: 0000 0000b, Software reset: 1U00 0000b, nRESET pin: U100 0000b, Hard fault: UU10 0000b Others: UUU0 0000b

7	6	5	4	3	2	1	0
SWRF	**RSTPINF**	**HardF**	**HardFInt**	**GF2**	**-**	**0**	**DPS**
R/W	R/W	R/W	R/W	R/W	-	R	R/W

Bit	Name	Description
[6]	**RSTPINF**	**External Reset Flag** When the MCU is reset by the external reset, this bit will be set via hardware. It is recommended that the flag be cleared via software.
[5]	**HardF**	**Hard Fault Reset Flag** Once CPU fetches instruction address over Flash size while EHFI (EIE1.4)=0, MCU will reset and this bit will be set via hardware. It is recommended that the flag be cleared via software. **Note:** If MCU run in OCD debug mode and OCDEN = 0, Hard fault reset will disable. Only HardF flag be asserted.

6.3.3.4 Watchdog Timer Reset

The WDT is a free running timer with programmable time-out intervals and a dedicated internal clock source. User can clear the WDT at any time, causing it to restart the counter. When the selected time-out occurs but no software response taking place for a while, the WDT will reset the system directly and CPU will begin execution from 0000H.

Once a reset due to WDT occurs, the WDT reset flag WDTRF (WDCON.3) will be set. This bit keeps unchanged after any reset other than a power-on reset or WDT reset itself. User can clear WDTRF via software.

WDCON – Watchdog Timer Control (TA Protected)

Register	SFR Address	Reset Value
WDCON	AAH, Page 0, TA protected	POR 0000_0111 b WDT 0000_1UUU b Others 0000_UUUU b

7	6	5	4	3	2	1	0
WDTR	**WDCLR**	**WDTF**	**WIDPD**	**WDTRF**	**WDPS[2:0]**		
R/W	R/W	R/W	R/W	R/W	R/W		

Address: AAH, Page 0 Reset value: POR: 0000 0111b / WDT: 0000 1UUUb / Others: 0000 UUUUb

Bit	Name	Description
[3]	**WDTRF**	**WDT Reset Flag** When the CPU is reset by WDT time-out event, this bit will be set via hardware. This flag is recommended to be cleared via software after reset.

6.3.3.5 Software Reset

The ML51/ML54/ML56 Series provides a software reset, which allows the software to reset the whole system just similar to an external reset, initializing the MCU as it reset state. The software reset is quite useful in the end of an ISP progress. For example, if an ISP of Boot Code updating User Code finishes, a software reset can be asserted to re-boot CPU to execute new User Code immediately. Writing 1 to SWRST (CHPCON.7) will trigger a software reset. Note that this bit is writing TA protection. The instruction that sets the SWRST bit is the last instruction that will be executed before the device reset. See demo code below.

If a software reset occurs, SWRF (AUXR0.7) will be automatically set by hardware. User can check it as the reset source indicator. SWRF keeps unchanged after any reset other than a power-on reset or software reset itself. SWRF can be cleared via software.

CHPCON – Chip Control (TA Protected)

Register	SFR Address	Reset Value
CHPCON	9FH, Page 0, TA protected	Software 0000_00U0 b Others 0000_00C0 b

7	6	5	4	3	2	1	0
SWRST	**IAPFF**	**-**	**-**	**-**	**-**	**BS**	**IAPEN**
W	R/W	-	-	-	-	R/W	R/W

Bit	Name	Description
[7]	**SWRST**	**Software Reset** To set this bit as logic 1 will cause a software reset. It will automatically be cleared via hardware after reset is finished.

AUXR0 – Auxiliary Register 0

Register	SFR Address	Reset Value
AUXR0	A2H, Page 0	POR: 0000 0000b, Software reset: 1U00 0000b, nRESET pin: U100 0000b, Hard fault: UU10 0000b Others: UUU0 0000b

7	6	5	4	3	2	1	0
SWRF	**RSTPINF**	**HardF**	**HardFInt**	**GF2**	**-**	**0**	**DPS**
R/W	R/W	R/W	R/W	R/W	-	R	R/W

Bit	Name	Description
[7]	**SWRF**	**Software Reset Flag** When the MCU is reset via software reset, this bit will be set via hardware. It is recommended that the flag be cleared via software.

The software demo code is listed below.

```
    ANL  AUXR0,#01111111b ;software reset flag clear
    CLR  EA
    MOV  TA,#0Aah
    MOV  TA,#55h
    ORL  CHPCON,#10000000b  ;software reset
```

6.3.3.6 Boot Select

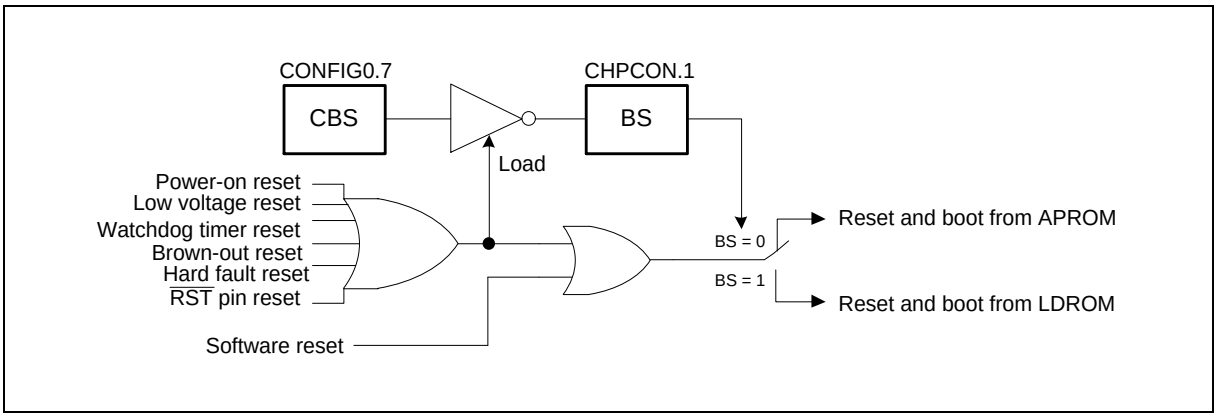


Figure 6.3-3 Boot Selecting Diagram

The ML51/ML54/ML56 Series provides user a flexible boot selection for variant application. The SFR bit BS in CHPCON.1 determines MCU booting from APROM or LDROM after any source of reset. If reset occurs and BS is 0, MCU will reboot from address 0000H of APROM. Else, the CPU will reboot from address 0000H of LDROM. Note that BS is loaded from the inverted value of CBS bit in CONFIG0.7 after all resets except software reset.

CONFIG0

7	6	5	4	3	2	1	0
CBS	**FSYS**	**OCDPWM**	**OCDEN**	**-**	**-**	**LOCK**	**-**
R/W	R/W	R/W	R/W	-	-	R/W	-

Factory default value: 1111 1111b

Bit	Name	Description
[7]	**CBS**	**CONFIG Boot Select** This bit defines from which block that MCU re-boots after resets except software reset. 1 = MCU will re-boot from APROM after resets except software reset. 0 = MCU will re-boot from LDROM after resets except software reset.

CHPCON – Chip Control (TA Protected)

Register	SFR Address	Reset Value
CHPCON	9FH, Page 0, TA protected	Software 0000_00U0 b Others 0000_00C0 b

7	6	5	4	3	2	1	0
SWRST	**IAPFF**	**-**	**-**	**-**	**-**	**BS**[1]	**IAPEN**
W	R/W	-	-	-	-	R/W	R/W

Address: 9FH, Page 0 Reset value: Software: 0000 00U0b / others: 0000 00C0b

Bit	Name	Description
[1]	**BS**	**Boot Select** This bit defines from which block that MCU re-boots after all resets. 0 = MCU will re-boot from APROM after all resets. 1 = MCU will re-boot from LDROM after all resets.

[1] BS is initialized by being loaded from the inverted value of CBS bit in CONFIG0.7 after resets except software reset. It keeps unchanged after software reset.

After the MCU is released from reset state, the hardware will always check the BS bit instead of the CBS bit to determine from which block that the device reboots.

6.3.3.7 Reset State

The reset state besides power-on reset does not affect the on-chip RAM. The data in the RAM will be preserved during the reset. After the power-on reset the RAM contents will be indeterminate.

After a reset, most of SFR go to their initial values except bits, which are affected by different reset events.. The Program Counter is forced to 0000H and held as long as the reset condition is applied. Note that the Stack Pointer is also reset to 07H and thus the stack contents may be effectively lost during the reset event even though the RAM contents are not altered.

After a reset, all peripherals and interrupts are disabled. The I/O port latches resumes FFH and I/O mode input-only.

6.3.4 Interrupt System

6.3.4.1 Interrupt Overview

The purpose of the interrupt is to make the software deal with unscheduled or asynchronous events. The ML51/ML54/ML56 Series has a four-priority-level interrupt structure with 31 interrupt sources. Each of the interrupt sources has an individual priority setting bits, interrupt vector and enable bit. In addition, the interrupts can be globally enabled or disabled. When an interrupt occurs, the CPU is expected to service the interrupt. This service is specified as an Interrupt Service Routine (ISR). The ISR resides at a predetermined address as shown in Table 6.3-4 Interrupt Vectors. When the interrupt occurs if enabled, the CPU will vector to the respective location depending on interrupt source, execute the code at this location, stay in an interrupt service state until the ISR is done. Once an ISR has begun, it can be interrupted only by a higher priority interrupt. The ISR should be terminated by a return from interrupt instruction RETI. This instruction will force the CPU return to the instruction that would have been next when the interrupt occurred.

Source	Vector Addess	Vector Number	Source	Vector Address	Vector Number
Reset	0000H	-	Serial port 1 interrupt	007BH	15
External interrupt 0	0003H	0	Timer 3 overflow	0083H	16
Timer 0 overflow	000BH	1	Self Wake-up Timer interrupt	008BH	17
External interrupt 1	0013H	2	CPU Hard Fault Interrupt	0093H	18
Timer 1 overflow	001BH	3	SMC0 Interrupt	009BH	19
Serial port 0 interrupt	0023H	4	PDMA0 Interrupt	00A3H	20
Timer 2 event	002BH	5	PDMA1 Interrupt	00ABH	21
R/W0 status/timer-out interrupt	0033H	6	SPI1 Interrupt	00B3H	22
Pin interrupt	003BH	7	ACMP Interrupt	00BBH	23
Brown-out detection interrupt	0043H	8	R/W1 status/timer-out interrupt	00C3H	24
SPI0 interrupt	004BH	9	PWM123 Interrupt	00CBH	25
WDT interrupt	0053H	10	Touch_Key interrupt	00D3H	26
ADC interrupt	005BH	11	SMC1 Interrupt	00DBH	27
Input capture interrupt	0063H	12	PDMA2 Interrupt	00E3H	28
PWM0 interrupt	006BH	13	PDMA3 Interrupt	00EBH	29
Fault Brake0 interrupt	0073H	14	RTC Interrupt	00F3H	30

Table 6.3-4 Interrupt Vectors

6.3.4.2 Enabling Interrupts

Each of individual interrupt sources can be enabled or disabled through the use of an associated

interrupt enable bit in the IE and EIE0 SFR. There is also a global enable bit EA bit (I.E.7), which can be cleared to disable all the interrupts at once. It is set to enable all individually enabled interrupts. Setting the EA bit to logic 0 disables all interrupt sources regardless of the individual interrupt-enable settings. Note that interrupts which occur when the EA bit is set to logic 0 will be held in a pending state, and will not be serviced until the EA bit is set back to logic 1. All interrupt flags that generate interrupts can also be set via software. Thereby software initiated interrupts can be generated.

Note that every interrupts, if enabled, is generated by a setting as logic 1 of its interrupt flag no matter by hardware or software. User should take care of each interrupt flag in its own interrupt service routine (ISR). Most of interrupt flags should be cleared by writing it as logic 0 via software to avoid recursive interrupt requests.

IE – Interrupt Enable

Register	SFR Address	Reset Value
IE	A8H, All pages, Bit addressable	0000 _0000 b

7	6	5	4	3	2	1	0
EA	**EADC**	**EBOD**	**ES**	**ET1**	**EX1**	**ET0**	**EX0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**EA**	**Enable All Interrupt** This bit globally enables/disables all interrupts that are individually enabled. 0 = All interrupt sources Disabled. 1 = Each interrupt Enabled depending on its individual mask setting. Individual interrupts will occur if enabled.
[6]	**EADC**	**Enable ADC Interrupt** 0 = ADC interrupt Disabled. 1 = ADC interrupt Enable. When interrupt generated ADCF (ADCCON0.7) set 1.
[5]	**EBOD**	**Enable Brown-Out Interrupt** 0 = Brown-out detection interrupt Disabled. 1 = Brown-out detection interrupt Enable. When interrupt generated BOF (BODCON0.3) set 1.
[4]	**ES**	**Enable Serial Port 0 Interrupt** 0 = Serial port 0 interrupt Disabled. 1 = Serial port 0 interrupt Enable. When interrupt generated TI (SCON.1) or RI (SCON.0) set 1.
[3]	**ET1**	**Enable Timer 1 Interrupt** 0 = Timer 1 interrupt Disabled. 1 = Timer 1 interrupt Enable. When interrupt generated TF1 (TCON.7) set 1.
[2]	**EX1**	**Enable External Interrupt 1** 0 = External interrupt 1 Disabled. 1 = External interrupt 1 interrupt Enable. When interrupt generated INT1 pin set 1.
[1]	**ET0**	**Enable Timer 0 Interrupt** 0 = Timer 0 interrupt Disabled. 1 = Timer 0 interrupt Enable. When interrupt generated TF0 (TCON.5) set 1.
[0]	**EX0**	**Enable External Interrupt 0** 0 = External interrupt 0 Disabled. 1 = External interrupt 0 interrupt Enable. When interrupt generated INT0 pin set 1.

EIE0 – Extensive Interrupt Enable

Register	SFR Address	Reset Value
EIE0	9BH, Page 0	0000 _0000 b

7	6	5	4	3	2	1	0
ET2	**ESPI0**	**EFB0**	**EWDT**	**EPWM0**	**ECAP**	**EPI**	**EI2C0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**ET2**	**Enable Timer 2 Interrupt** 0 = Timer 2 interrupt Disabled. 1 = Timer 2 interrupt Enable. When interrupt generated, TF2 (T2CON.7) set 1
[6]	**ESPI0**	**Enable SPI Interrupt** 0 = SPI interrupt Disabled. 1 = SPI interrupt Enable. When interrupt generated SPIF (SPInSR.7), SPIOVF (SPInSR.5), or MODF (SPInSR.4) set 1 .
[5]	**EFB0**	**Enable Fault Brake Interrupt** 0 = Fault Brake interrupt Disabled. 1 = Fault Brake interrupt Enable. When interrupt generated FBF (PWM0FBD.7) set 1.
[4]	**EWDT**	**Enable WDT Interrupt** 0 = WDT interrupt Disabled. 1 = WDT interrupt Enable. When interrupt generated WDTF (WDCON.5) set 1.
[3]	**EPWM0**	**Enable PWM0 Interrupt** 0 = PWM interrupt Disabled. 1 = PWM interrupt Enable. When interrupt generated PWMF (PWMnCON0.5) set 1.
[2]	**ECAP**	**Enable Input Capture Interrupt** 0 = Input capture interrupt Disabled. 1 = Input capture interrupt Enable. When interrupt generated CAPF[2:0] (CAPCON0[2:0]) set 1.
[1]	**EPI**	**Enable Pin Interrupt** 0 = Pin interrupt Disabled. 1 = Pin interrupt Enable. When interrupt generated PIF related bit set 1.
[0]	**EI2C0**	**Enable R/W0 Interrupt** 0 = R/W interrupt Disabled. 1 = R/W interrupt Enable. When interrupt generated SI (R/W0CON.3) or I2TOF (R/W0TOC.0) set 1.

EIE1 – Extensive Interrupt Enable 1

Register	SFR Address	Reset Value
EIE1	9CH, Page 0	0000 _0000 b

7	6	5	4	3	2	1	0
-	EPWM123	EI2C1	ESPI1	EHFI	EWKT	ET3	ES1
-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	-	Reserved
[6]	EPWM123	**Enable PWM123 Interrupt** 0 = PWM1/2/3 interrupt Disabled. 1 = PWM1/2/3 interrupt Enable. When interrupt generated PWMF (PWM1CON0.5) set 1.
[5]	EI2C1	**Enable R/W1 Interrupt** 0 = R/W1 interrupt Disabled. 1 = R/W1 interrupt Enable. When interrupt generated SI (R/W1CON.3) or I2TOF (R/W1TOC.0) set 1.
[4]	ESPI1	**Enable SPI1 Interrupt** 0 = SPI1 interrupt Disabled. 1 = SPI1 interrupt Enable. When interrupt generated SPIF (SP2SR.7), MODF (SP2SR.4) or SPIOVF (SP2SR.5) set 1
[3]	EHFI	**Enable Hard Fault Interrupt** 0 = hard fault interrupt Disabled and hard fault reset is Enabled 1 = hard fault interrupt Enable. When interrupt generated HFIF (AUXR0.4) set 1.
[2]	EWKT	**Enable WKT Interrupt** 0 = WKT interrupt Disabled. 1 = WKT interrupt Enable. When interrupt generated WKTF (WKCON.4) set 1.
[1]	ET3	**Enable Timer 3 Interrupt** 0 = Timer 3 interrupt Disabled. 1 = Timer 3interrupt Enable. When interrupt generated TF3 (T3CON.4) set 1.
[0]	ES1	**Enable Serial Port 1 Interrupt** 0 = Serial port 1 interrupt Disabled. 1 = Serial port 1 interrupt Enable. When interrupt generated TI_1 (S1CON.1) or RI_1 (S1CON.0) set 1.

ScnIE – SC Interrupt Enable Control Register

Register	SFR Address	Reset Value
SC0IE	DDH, Page 0	0000_0000 b
SC1IE	DDH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	ACERRIEN	BGTIEN	TERRIEN	TBEIEN	RDAIEN
-	-	-	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:5]	-	Reserved
[4]	ACERRIEN	**Auto Convention Error Interrupt Enable Bit** This field is used to enable auto-convention error interrupt. 0 = Auto-convention error interrupt Disabled. 1 = Auto-convention error interrupt Enabled.
[3]	BGTIEN	**Block Guard Time Interrupt Enable Bit** This field is used to enable block guard time interrupt. 0 = Block guard time interrupt Disabled. 1 = Block guard time interrupt Enabled.
[2]	TERRIEN	**Transfer Error Interrupt Enable Bit** This field is used to enable transfer error interrupt. The transfer error states is at SC0TSR register which includes receiver break error BEF(SC0TSR[6]), frame error FEF(SC0TSR[5]), parity error PEF(SC0TSR[4]), receiver buffer overflow error RXOV(SC0TSR[0]) and transmit buffer overflow error TXOV(SC0TSR[2]). 0 = Transfer error interrupt Disabled. 1 = Transfer error interrupt Enabled.
[1]	TBEIEN	**Transmit Buffer Empty Interrupt Enable Bit** This field is used to enable transmit buffer empty interrupt. 0 = Transmit buffer empty interrupt Disabled. 1 = Transmit buffer empty interrupt Enabled.
[0]	RDAIEN	**Receive Data Reach Interrupt Enable Bit** This field is used to enable received data interrupt. 0 = Receive data interrupt Disabled. 1 = Receive data interrupt Enabled.

DMAnCR – PDMAn Control Register

Register	SFR Address	Reset Value
DMA0CR0	92H, Page 0	0000_0000 b
DMA1CR0	EBH, Page 0	0000_0000 b
DMA2CR0	B3H, Page 2	0000_0000 b
DMA3CR0	ABH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PSSEL[3:0]				HIE	FIE	RUN	EN
R/W				R/W	R/W	R/W	R/W

Bit	Name	Description
[7:4]	**PSSEL[3:0]**	**Peripheral Source Select** 0000 = XRAM to XRAM 0001 = SPI0 RX 0010 = SMC0/UART2 RX. 0011 = SPI1 RX 0100 = Reserved, No peripheral source select 0101 = SPI0 TX 0110 = SMC0/UART2 TX. 0111 = SPI1 TX 1010 = SMC1/UART3 RX. 1110 = SMC1/UART3 TX. The others are reserved, no periperal source selected **Note:** 0001~0011,1010 : peripheral devices to XRAM memory 0101~0111,1110 : XRAM memory to peripheral devices
[3]	**HIE**	**PDMA HALFTransfer Done Interrupt Enable Bit** 0 = Interrupt Disabled when PDMA half transfer is done. 1 = Interrupt Enabled when PDMA half transfer is done.
[2]	**FIE**	**PDMA Full Transfer Done Interrupt Enable Bit** 0 = Interrupt Disabled when PDMA full transfer is done. 1 = Interrupt Enabled when PDMA full transfer is done.
[1]	**RUN**	**Trigger Enable Bit** 0 = No effect. 1 = PDMA data transfer Enabled. **Note 1:** When PDMA transfer completed, this bit will be cleared automatically.
[0]	**EN**	**PDMA Enable Bit** Setting this bit to 1 enables PDMA operation. If this bit is cleared, PDMA will ignore all PDMA request and Reset the internal state machine, pointers and internal buffer. The contents of all Register Description will not be cleared.

6.3.4.3 Interrupt Priorities

There are four priority levels for all interrupts. They are level highest, high, low, and lowest; and they are represented by level 3, level 2, level 1, and level 0. The interrupt sources can be individually set to one of four priority levels by setting their own priority bits. Table 6.3-5 Interrupt Priority Level Setting lists four priority setting. Naturally, a low level priority interrupt can itself be interrupted by a high level priority interrupt, but not by any same level interrupt or lower level. In addition, there exists a pre-defined natural priority among the interrupts themselves. The natural priority comes into play when the interrupt controller has to resolve simultaneous requests having the same priority level.

In case of multiple interrupts, the following rules apply:

1. While a low priority interrupt handler is running, if a high priority interrupt arrives, the handler will be interrupted and the high priority handler will run. When the high priority handler does “RETI”, the low priority handler will resume. When this handler does “RETI”, control is passed back to the main program.

2. If a high priority interrupt is running, it cannot be interrupted by any other source – even if it is a high priority interrupt which is higher in natural priority.

3. A low-priority interrupt handler will be invoked only if no other interrupt is already executing. Again, the low priority interrupt cannot preempt another low priority interrupt, even if the later one is higher in natural priority.

4. If two interrupts occur at the same time, the interrupt with higher priority will execute first. If both interrupts are of the same priority, the interrupt which is higher in natural priority will be executed first. This is the only context in which the natural priority matters.

Interrupt Priority Control Bits		Interrupt Priority Level
IPH/EIPH0/EIPH1/EIPH2	IP/EIP0/EIP1/EIP2	
0	0	Level 0 (lowest)
0	1	Level 1
1	0	Level 2
1	1	Level 3 (highest)

Table 6.3-5 Interrupt Priority Level Setting

This natural priority is defined as shown on Table 6.3-6 Characteristics of Each Interrupt SourceIt also summarizes the interrupt sources, flag bits, vector addresses, enable bits, priority bits, natural priority and the permission to wake up the CPU from Power-down mode. For details of waking CPU up from Power-down mode, please see Section 6.3.2.4 R/W Mode

Interrupt Source	Vector Address	Interrupt Flag	Enable Bit	Natural Priority	Priority Control Bits	PD Wake-Up
Reset	0000H	-	Always Enabled	Highest	-	Yes
CPU Hard Fault	0093H	HFIF (RSR.5)	EHFI (EIE1.3)	1	PHF, PHFH	No
External interrupt 0	0003H	IE0 (TCON.1)	EX0 (I.E.0)	2	PX0, PX0H	Yes
Brown-out	0043H	BOF (BODCON0.3)	EBOD (I.E.5)	3	PBOD, PBODH	Yes
Watchdog Timer	0053H	WDTF (WDCON.5)	EWDT (EIE0.4)	4	PWDT, PWDTH	Yes
Timer 0	000BH	TF0 (TCON.5)	ET0 (I.E.1)	5	PT0, PT0H	No
R/W0 status/time-out	0033H	SI (R/W0CON.3) I2TOF (R/W0TOC.0)	EI2C0 (EIE0.0)	6	PI2C0, PI2C0H	No
ADC	005BH	ADCF (ADCCON0.7)	EADC (I.E.6)	7	PADC, PADCH	No

Interrupt Source	Vector Address	Interrupt Flag	Enable Bit	Natural Priority	Priority Control Bits	PD Wake-Up
External interrupt 1	0013H	IE1 (TCON.3)	EX1 (I.E.2)	8	PX1, PX1H	Yes
Pin interrupt	003BH	PIF0 (PIF.0) PIF1 (PIF.1) PIF2 (PIF.2) PIF3 (PIF.3) PIF4 (PIF.4) PIF5 (PIF.5) PIF6 (PIF.6) PIF7 (PIF.7)	EPI (EIE0.1)	9	PPI, PPIH	Yes
Timer 1	001BH	TF1 (TCON.7)	ET1 (I.E.3)	10	PT1, PT1H	No
Serial port 0	0023H	RI (SCON.0) TI (SCON.1)	ES (I.E.4)	11	PS, PSH	No
PWM0 Fault Brake event	0073H	FBF (PWM0FBD.7)	EFB0 (EIE0.5)	12	PFB, PFBH	No
SPI0	004BH	SPIF (SPI0SR.7) MODF (SPI0SR.4) SPIOVF (SPI0SR.5)	ESPI0 (EIE0.6)	13	PSPI, PSPIH	No
Timer 2	002BH	TF2 (T2CON.7)	ET2 (EIE0.7)	14	PT2, PT2H	No
Input capture	0063H	CAPF0 (CAPCON0.0) CAPF1 (CAPCON0.1) CAPF2 (CAPCON0.2)	ECAP (EIE0.2)	15	PCAP, PCAPH	No
PWM0 interrupt	006BH	PWMF (PWM0CON0.5)	EPWM0 (EIE0.3)	16	PPWM0, PPWM0H	No
Serial port 1	007BH	RI_1 (S1CON.0) TI_1 (S1CON.1)	ES1 (EIE1.0)	17	PS1, PS1H	No
Timer 3	0083H	TF3 (T3CON.4)	ET3 (EIE1.1)	18	PT3, PT3H	No
Self Wake-up Timer	008BH	WKTF (WKCON.4)	EWKT (EIE1.2)	19	PWKT, PWKTH	Yes
SMC0	009BH	RDAIF (SC0IS.0) TBEIF (SC0IS.1) TERRIF (SC0IS.2) BGTIF (SC0IS.3) ACERRIF (SC0IS.4)	RDAIEN (SC0I.E.0) TBEIEN (SC0I.E.1) TERRIEN (SC0I.E.2) BGTIEN (SC0I.E.3) ACERRIEN (SC0I.E.4)	20	SMC0, SMC0H	No
PDMA0	00A3H	FDONE (DMA0TSR.0) HDONE (DMA0TSR.1)	FIE (DMA0CR0.2) HIE (DMA0CR0.3)	21	PDMA0, PDMA0H	No
PDMA1	00ABH	FDONE (DMA1TSR.0) HDONE (DMA1TSR.1)	FIE (DMA1CR0.2) HIE (DMA1CR0.3)	22	PDMA1, PDMA1H	No
SPI1	00B3H	SPIF (SPI1SR.7) MODF (SPI1SR.4) SPIOVF (SPI1SR.5)	ESPI1 (EIE1.4)	23	PSPI, PSPIH	No
ACMP	00BBH	ACMP0IF (ACMPSR.0) ACMP1IF (ACMPSR.2)	CMPIE (ACMPCR0.1) CMPIE (ACMPCR1.1)	24	PACMP, PACMPH	Yes
R/W1 status/time-out	00C3h	SI (R/W1CON.3) I2TOF (R/W1TOC.0)	EI2C1 (EIE1.5)	25	PI2C1, PI2C1H	No
PWM123 interrupt	00CBH	PWMF(PWM1CON0.5) PWMF(PWM2CON0.5) PWMF(PWM3CON0.5)	EPWM123 (EIE1.6)	26	PPWM1, PPWM1H	No

Interrupt Source	Vector Address	Interrupt Flag	Enable Bit	Natural Priority	Priority Control Bits	PD Wake-Up
Touch_Key	00D3H	TKSCIF (TKSTA0.1) TKIF (TKSTA0.2) TKIF_ALL (TKSTA0.3) TKIF0 (TKSTA1.0) TKIF1 (TKSTA1.1) TKIF2 (TKSTA1.2) TKIF3 (TKSTA1.3) TKIF4 (TKSTA1.4) TKIF5 (TKSTA1.5) TKIF6 (TKSTA1.6) TKIF7 (TKSTA1.7) TKIF8 (TKSTA2.0) TKIF9 (TKSTA2.1) TKIF10 (TKSTA2.2) TKIF11 (TKSTA2.3) TKIF12 (TKSTA2.4) TKIF13 (TKSTA2.5) TKIF14 (TKSTA2.6)	TKSCTHIE (TKINTEN.0) TKSCIE (TKINTEN.1)	27	TK, TKH	Yes
SMC1	00D3H	RDAIF (SC1IS.0) TBEIF (SC1IS.1) TERRIF (SC1IS.2) BGTIF (SC1IS.3) ACERRIF (SC1IS.4)	RDAIEN (SC1I.E.0) TBEIEN (SC1I.E.1) TERRIEN (SC1I.E.2) BGTIEN (SC1I.E.3) ACERRIEN (SC1I.E.4)	28	SMC1, SMC1H	No
PDMA2	00E3H	FDONE (DMA2TSR.0) HDONE (DMA2TSR.1)	FIE (DMA2CR0.2) HIE (DMA2CR0.3)	29	PDMA2, PDMA2H	No
PDMA3	00EBH	FDONE (DMA3TSR.0) HDONE (DMA3TSR.1)	FIE (DMA3CR0.2) HIE (DMA3CR0.3)	30	PDMA3, PDMA3H	No
RTC	00F3H	ALMIF (RTCINTSTS.0) TICKIF(RTCINTSTS.1)	ALMIEN (RTCINTEN.0) TICKIEN(RTCINTEN.1)	31	RTC, RTCH	Yes
LCD	00FBH	LCDCPOVIF (LCDIF.2) LCDCPIF (LCDIF.1) LCDCPALIF (LCDIF.0)	LCDIE (LCDCON1.2) LCDIS (LCDCON1.3)	32		No

Note:
1. While the external interrupt pin is set as edge triggered (Itx = 1), its own flag Iex will be automatically cleared if the interrupt service routine (ISR) is executed. While as level triggered (Itx = 0), Iex follows the inverse of respective pin state. It is not controlled via software.
2. TF0, TF1, or TF3 is automatically cleared if the interrupt service routine (ISR) is executed. On the contrary, be aware that TF2 is not.
3. If level triggered is selected for pin interrupt channel n, PIFn flag reflects the respective channel state. It is not controlled via software.

Table 6.3-6 Characteristics of Each Interrupt Source

IP – Interrupt Priority

Register	SFR Address	Reset Value
IP	B8H, All pages, Bit addressable	0000_0000 b

7	6	5	4	3	2	1	0
-	PADC	PBOD	PS	PT1	PX1	PT0	PX0
-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	-	Reserved
[6]	PADC	ADC interrupt priority low bit
[5]	PBOD	Brown-out detection interrupt priority low bit
[4]	PS	Serial port 0 interrupt priority low bit
[3]	PT1	Timer 1 interrupt priority low bit
[2]	PX1	External interrupt 1 priority low bit
[1]	PT0	Timer 0 interrupt priority low bit
[0]	PX0	External interrupt 0 priority low bit
Note: used in combination with the IPH to determine the priority of each interrupt source. See Table 6.3-5 Interrupt Priority Level Setting for correct interrupt priority configuration.		

IPH – Interrupt Priority High

Register	SFR Address	Reset Value
IPH	B7H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
-	PADCH	PBODH	PSH	PT1H	PX1H	PT0H	PX0H
-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	-	-
[6]	PADCH	ADC interrupt priority high bit
[5]	PBODH	Brown-out detection interrupt priority high bit
[4]	PSH	Serial port 0 interrupt priority high bit
[3]	PT1H	Timer 1 interrupt priority high bit
[2]	PX1H	External interrupt 1 priority high bit
[1]	PT0H	Timer 0 interrupt priority high bit
[0]	PX0H	External interrupt 0 priority high bit

EIP0 – Extensive Interrupt Priority

Register	SFR Address	Reset Value
EIP0	EFH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
PT2	**PSPI0**	**PFB**	**PWDT**	**PPWM0**	**PCAP**	**PPI**	**PI2C0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**PT2**	Timer 2 interrupt priority low bit
[6]	**PSPI0**	SPI0 interrupt priority low bit
[5]	**PFB**	Fault Brake interrupt priority low bit
[4]	**PWDT**	WDT interrupt priority low bit
[3]	**PPWM0**	PWM interrupt priority low bit
[2]	**PCAP**	Input capture interrupt priority low bit
[1]	**PPI**	Pin interrupt priority low bit
[0]	**PI2C0**	R/W interrupt priority low bit
Note: EIP0 is used in combination with the EIPH0 to determine the priority of each interrupt source. See Table 6.3-5 Interrupt Priority Level Setting for correct interrupt priority configuration.		

EIPH0 – Extensive Interrupt Priority High

Register	SFR Address	Reset Value
EIPH0	F7H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
PT2H	**PSPI0H**	**PFBH**	**PWDTH**	**PPWM0H**	**PCAPH**	**PPIH**	**PI2C0H**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**PT2H**	Timer 2 interrupt priority high bit
[6]	**PSPI0H**	SPI0 interrupt priority high bit
[5]	**PFBH**	Fault Brake interrupt priority high bit
[4]	**PWDTH**	WDT interrupt priority high bit
[3]	**PPWM0H**	PWM0 interrupt priority high bit
[2]	**PCAPH**	Input capture interrupt priority high bit
[1]	**PPIH**	Pin interrupt priority high bit
[0]	**PI2C0H**	R/W interrupt priority high bit
Note: EIPH0 is used in combination with the EIP0 to determine the priority of each interrupt source. See Table 6.3-5 Interrupt Priority Level Setting for correct interrupt priority configuration.		

EIP1 – Extensive Interrupt Priority 1

Register	SFR Address	Reset Value
EIP1	FEH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
PSPI1	**PDMA1**	**PDMA0**	**PSMC**	**PHF**	**PWKT**	**PT3**	**PS1**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**PSPI1**	SPI1 interrupt priority low bit
[6]	**PDMA1**	PDMA1 interrupt priority low bit
[5]	**PDMA0**	PDMA0 interrupt priority low bit
[4]	**PSMC**	SMC interrupt priority low bit
[3]	**PHF**	Hard fault interrupt priority low bit
[2]	**PWKT**	WKT interrupt priority low bit
[1]	**PT3**	Timer 3 interrupt priority low bit
[0]	**PS1**	Serial port 1 interrupt priority low bit
Note: EIP1 is used in combination with the EIPH1 to determine the priority of each interrupt source. See Table 6.3-5 Interrupt Priority Level Setting for correct interrupt priority configuration.		

EIPH1 – Extensive Interrupt Priority High Byte

Register	SFR Address	Reset Value
EIPH1	FFH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
PSPI1H	**PDMA1H**	**PDMA0H**	**PSMCH**	**PHFH**	**PWKTH**	**PT3H**	**PS1H**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**PSPI1H**	SPI1 interrupt priority high bit
[6]	**PDMA1H**	PDMA1 interrupt priority high bit
[5]	**PDMA0H**	PDMA0 interrupt priority high bit
[4]	**PSMCH**	SMC interrupt priority high bit
[3]	**PHFH**	Hard fault interrupt priority high bit
[2]	**PWKTH**	WKT interrupt priority high bit
[1]	**PT3H**	Timer 3 interrupt priority high bit
[0]	**PS1H**	Serial port 1 interrupt priority high bit
Note: EIPH1 is used in combination with the EIP1 to determine the priority of each interrupt source. See Table 6.3-5 Interrupt Priority Level Setting for correct interrupt priority configuration.		

EIP2 – Extensive Interrupt Priority 2

Register	SFR Address	Reset Value
EIP2	ACH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
RTC	**PDMA3**	**PDMA2**	**SMC1**	**TK**	**PPWM1**	**PI2C1**	**PACMP**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**RTC**	RTC interrupt priority low bit
[6]	**PDMA3**	PDMA3 interrupt priority low bit
[5]	**PDMA2**	PDMA2 interrupt priority low bit
[4]	**SMC1**	SMC1 interrupt priority low bit
[3]	**TK**	Touch Key interrupt priority low bit
[2]	**PPWM1**	PPWM1 interrupt priority low bit
[1]	**PI2C1**	R/W interrupt priority low bit
[0]	**PACMP**	ACMP interrupt priority low bit
Note: EIP2 is used in combination with the EIPH2 to determine the priority of each interrupt source. See Table 6.3-5 Interrupt Priority Level Setting for correct interrupt priority configuration.		

EIPH2 – Extensive Interrupt Priority High 2

Register	SFR Address	Reset Value
EIPH2	ADH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
RTCH	**PDMA3H**	**PDMA2H**	**SMC1H**	**TKH**	**PPWM1H**	**PI2C1H**	**PACMPH**
RW	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**RTCH**	RTCH interrupt priority high bit
[6]	**PDMA3H**	PDMA3H interrupt priority high bit
[5]	**PDMA2H**	PDMA2H interrupt priority high bit
[4]	**SMC1H**	SMC1H interrupt priority high bit
[3]	**TKH**	Touch Key interrupt priority high bit
[2]	**PPWM1H**	PPWM1H interrupt priority high bit
[1]	**PI2C1H**	R/W interrupt priority high bit
[0]	**PACMPH**	ACMP interrupt priority high bit
Note: EIPH2 is used in combination with the EIP2 to determine the priority of each interrupt source. See Table 6.3-5 Interrupt Priority Level Setting for correct interrupt priority configuration.		

AUXR0 – Auxiliary Register 0

Register	SFR Address	Reset Value
AUXR0	A2H, Page 0	POR: 0000 0000b, Software reset: 1U00 0000b, nRESET pin: U100 0000b, Hard fault: UU10 0000b Others: UUU0 0000b

7	6	5	4	3	2	1	0
SWRF	**RSTPINF**	**HFRF**	**HFIF**	**GF2**	**-**	**0**	**DPS**
R/W	R/W	R/W	R/W	R/W	-	R	R/W

Bit	Name	Description
[7]	**SWRF**	**Software Reset Flag** When the MCU is reset via software reset, this bit will be set via hardware. It is recommended that the flag be cleared via software.
[6]	**RSTPINF**	**External Reset Flag** When the MCU is reset by the external reset, this bit will be set via hardware. It is recommended that the flag be cleared via software.
[5]	**HFRF**	**Hard Fault Reset Flag** Once CPU fetches instruction address over Flash size while EHFI (EIE1.4)=0, MCU will reset and this bit will be set via hardware. It is recommended that the flag be cleared via software. **Note:** If MCU run in OCD debug mode and OCDEN = 0, Hard fault reset will disable. Only HFRF flag be asserted.
[4]	**HFIF**	**Hard Fault Interrupt Flag** Once CPU fetches instruction address over Flash size while EHFI (EIE1.4)=1. MCU will be interrupt and this bit will be set via hardware. It is recommended that the flag be cleared via software.

6.3.4.4 Interrupt Service

The interrupt flags are sampled every system clock cycle. In the same cycle, the sampled interrupts are polled and their priority is resolved. If certain conditions are met then the hardware will execute an internally generated LCALL instruction, which will vector the process to the appropriate interrupt vector address. The conditions for generating the LCALL are,

1. An interrupt of equal or higher priority is not currently being serviced.

2. The current polling cycle is the last cycle of the instruction currently being executed.

3. The current instruction does not involve a write to any enabling or priority setting bits and is not a RETI.

If any of these conditions are not met, then the LCALL will not be generated. The polling cycle is repeated every system clock cycle. If an interrupt flag is active in one cycle but not responded to for the above conditions are not met, if the flag is not still active when the blocking condition is removed, the denied interrupt will not be serviced. This means that the interrupt flag, which was once active but not serviced is not remembered. Every polling cycle is new.

The processor responds to a valid interrupt by executing an LCALL instruction to the appropriate service routine. This action may or may not clear the flag, which caused the interrupt according to different interrupt source. The hardware LCALL behaves exactly like the software LCALL instruction. This instruction saves the Program Counter contents onto the Stack RAM but does not save the Program Status Word (PSW). The PC is reloaded with the vector address of that interrupt, which caused the LCALL. Execution continues from the vectored address until an RETI instruction is executed. On execution of the RETI instruction, the processor pops the Stack and loads the PC with the contents at the top of the stack. User should take care that the status of the stack. The processor does not notice anything if the stack contents are modified and will proceed with execution from the address put back into PC. Note that a simple RET instruction would perform exactly the same process as a RETI instruction, but it would not inform the Interrupt controller that the interrupt service routine is completed. RET would leave the controller still thinking that the service routine is underway, making future interrupts impossible.

6.3.4.5 Interrupt Latency

The response time for each interrupt source depends on several factors, such as the nature of the interrupt and the instruction underway. Each interrupt flags are polled and priority decoded each system clock cycle. If a request is active and all three previous conditions are met, then the hardware generated LCALL is executed. This LCALL itself takes 4 clock cycles to be completed. Thus, there is a minimum reaction time of 5 clock cycles between the interrupt flag being set and the interrupt service routine being executed.

A longer response time should be anticipated if any of the three conditions are not met. If a higher or equal priority is being serviced, then the interrupt latency time obviously depends on the nature of the service routine currently being executed. If the polling cycle is not the last clock cycle of the instruction being executed, then an additional delay is introduced. The maximum response time (if no other interrupt is currently being serviced or the new interrupt is of greater priority) occurs if the device is performing a RETI, and then executes a longest 6-clock-cycle instruction as the next instruction. From the time an interrupt source is activated (not detected), the longest reaction time is 16 clock cycles. This period includes 5 clock cycles to complete RETI, 6 clock cycles to complete the longest instruction, 1 clock cycle to detect the interrupt, and 4 clock cycles to complete the hardware LCALL to the interrupt vector location.

Thus in a single-interrupt system the interrupt response time will always be more than 5 clock cycles and not more than 16 clock cycles.

6.4 Flash Memory Control

6.4.1 In-application-programming (IAP)

Unlike RAM's real-time operation, to update Flash data often takes long time. Furthermore, it is a quite complex timing procedure to erase, program, or read Flash data. The ML51/ML54/ML56 Series carried out the Flash operation with convenient mechanism to help user re-programming the Flash content by In-Application-Programming (IAP). IAP is an in-circuit electrical erasure and programming method through software.

After IAP enabling by setting IAPEN (CHPCON.0 with TA protected) and setting the enable bit in IAPUEN that allows the target block to be updated, user can easily fill the 16-bit target address in IAPAH and IAPAL, data in IAPFD, and command in IAPCN. Then the IAP is ready to begin by setting a triggering bit IAPGO (IAPTRG.0). Note that IAPTRG is also TA protected. At this moment, the CPU holds the Program Counter and the built-in IAP automation takes over to control the internal charge-pump for high voltage and the detail signal timing. The erase and program time is internally controlled disregard of the operating voltage and frequency. Nominally, a page-erase time is 5 ms and a byte-program time is 23.5 μs. After IAP action completed, the Program Counter continues to run the following instructions. The IAPGO bit will be automatically cleared. An IAP failure flag, IAPFF (CHPCON.6), can be check whether the previous IAP operation was successful or not. Through this progress, user can easily erase, program, and verify the Flash Memory by just taking care of pure software.

The following registers are related to IAP processing.

CONFIG2

7	6	5	4	3	2	1	0
CBODEN	**CBOV[2:0]**			**BOIAP**	**CBORST**	**-**	**-**
R/W	R/W			R/W	R/W	-	-

Factory default value: 1111 1111b

Bit	Name	Description
[3]	**BOIAP**	**Brown-Out Inhibiting IAP** This bit decide whether IAP erasing or programming is inhibited by brown-out status. This bit is valid only when brown-out detection is enabled. 1 = IAP erasing or programming is inhibited if V_{DD} is lower than V_{BOD}. 0 = IAP erasing or programming is allowed under any workable V_{DD}.

CHPCON – Chip Control (TA Protected)

Register	SFR Address	Reset Value
CHPCON	9FH, Page 0, TA protected	Software 0000_00U0 b Others 0000_00C0 b

7	6	5	4	3	2	1	0
SWRST	**IAPFF**	**-**	**-**	**-**	**-**	**BS**	**IAPEN**
W	R/W	-	-	-	-	R/W	R/W

Bit	Name	Description
[6]	**IAPFF**	**IAP Fault Flag** The hardware will set this bit after IAPGO (IAPTRG.0) is set if any of the following condition is met: (1) The accessing address is oversize. (2) IAPCN command is invalid. (3) IAP erases or programs updating un-enabled block. (4) IAP erasing or programming operates under VBOD while BOIAP (CONFIG2.5) remains un-programmed 1 with BODEN (BODCON0.7) as 1 and BORST (BODCON0.2) as 0. This bit should be cleared via software.
[0]	**IAPEN**	**IAP Enable** 0 = IAP function Disabled. 1 = IAP function Enabled. Once enabling IAP function, the HIRC will be turned on for timing control. To clear IAPEN should always be the last instruction after IAP operation to stop internal oscillator if reducing power consumption is concerned.

IAPUEN – IAP Updating Enable (TA Protected)

Register	SFR Address	Reset Value
IAPUEN	A5H, Page 0, TA protected	0000 _0000 b

7	6	5	4	3	2	1	0
-	-	-	SPMEN	SPUEN	CFUEN	LDUEN	APUEN
-	-	-	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:5]	-	Reserved
[4]	SPMEN	**SPROM Memory Space Mapping Enable** 0 = CPU memory address 0xff80~0xffff is mapping to APROM memory 1 = CPU memory address 0xff80~0xffff is mapping to SPROM memory
[3]	SPUEN	**SPROM Memory Space Updated Enable(TA Protected)** 0 = Inhibit erasing or programming SPRO R/W by IAP 1 = Allow erasing or programming SPRO R/W by IAP.
[2]	CFUEN	**CONFIG Bytes Updated Enable** 0 = Inhibit erasing or programming CONFIG bytes by IAP. 1 = Allow erasing or programming CONFIG bytes by IAP.
[1]	LDUEN	**LDROM Updated Enable** 0 = Inhibit erasing or programming LDROM by IAP. 1 = Allow erasing or programming LDROM by IAP.
[0]	APUEN	**APROM Updated Enable** 0 = Inhibit erasing or programming APROM by IAP. 1 = Allow erasing or programming APROM by IAP.

IAPCN – IAP Control

Register	SFR Address	Reset Value
IAPCN	AFH, Page 0	0011_0000 b

7	6	5	4	3	2	1	0
IAPB[1:0]		FOEN	FCEN	FCTRL[3:0]			
R/W		R/W	R/W	R/W			

Bit	Name	Description
[7:6]	**IAPB[1:0]**	**IAP Control** This byte is used for IAP command. For details, see Figure 6.4-1 IAP Modes and Command Codes.
[5]	**FOEN**	**This Byte is Used for IAP Command.** For details, see Figure 6.4-1 IAP Modes and Command Codes.
[4]	**FCEN**	**This Byte is Used for IAP Command.** For details, see Figure 6.4-1 IAP Modes and Command Codes.
[3:0]	**FCTRL[3:0]**	**This Byte is Used for IAP Command.** For details, see Figure 6.4-1 IAP Modes and Command Codes.

IAPAH – IAP Address High Byte

Register	SFR Address	Reset Value
IAPAH	A7H, Page 0	0000 _0000 b

7	6	5	4	3	2	1	0
IAPA[15:8]							
R/W							

Bit	Name	Description
[7:0]	**IAPA[15:8]**	**IAP Address High Byte** IAPAH contains address IAPA[15:8] for IAP operations.

IAPAL – IAP Address Low Byte

Register	SFR Address	Reset Value
IAPAL	A6H, Page 0	0000 _0000 b

7	6	5	4	3	2	1	0
IAPA[7:0]							
R/W							

Bit	Name	Description
[7:0]	**IAPA[7:0]**	**IAP Address Low Byte** IAPAL contains address IAPA[7:0] for IAP operations.

IAPFD – IAP Flash Data

Register	SFR Address	Reset Value
IAPFD	AEH, Page 0	0000 _0000 b

7	6	5	4	3	2	1	0
IAPFD[7:0]							
R/W							

Bit	Name	Description
[7:0]	**IAPFD[7:0]**	**IAP Flash Data** This byte contains Flash data, which is read from or is going to be written to the Flash Memory. User should write data into IAPFD for program mode before triggering IAP processing and read data from IAPFD for read/verify mode after IAP processing is finished.

IAPTRG – IAP Trigger (TA Protected)

Register	SFR Address	Reset Value
IAPTRG	A4H, Page 0, TA protected	0000 _0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	IAPGO
-	-	-	-	-	-	-	W

Bit	Name	Description
[7:1]	-	Reserved

[0]	IAPGO	**IAP Go**
		IAP begins by setting this bit as logic 1. After this instruction, the CPU holds the Program Counter (PC) and the IAP hardware automation takes over to control the progress. After IAP action completed, the Program Counter continues to run the following instruction. The IAPGO bit will be automatically cleared and always read as logic 0.
		Before triggering an IAP action, interrupts (if enabled) should be temporary disabled for hardware limitation.
		The program process should follows below.
		CLR EA
		MOV TA,#0AAH
		MOV TA,#55H
		ORL IAPTRG,#01H
		(SETB EA)

6.4.1.1 IAP Commands

The ML51/ML54/ML56 Series provides a wide range of applications to perform IAP to APROM, LDROM, or CONFIG bytes. The IAP action mode and the destination of the Flash block are defined by IAP control register IAPCN.

IAP Mode	IAPCN				IAPA[15:0] {IAPAH, IAPAL}	IAPFD[7:0]
	IAPB [1:0]	FOEN	FCEN	FCTRL [3:0]		
Company ID read	XX[1]	0	0	1011	X	DAH
Device ID read	XX	0	0	1100	Low-byte DID: 0000H High-byte DID: 0001H	Low-byte DID High-byte DID
PID Read	x, x	0	0	1100	A[15:0]=0002H for low-byte ID A[15:0]=0003H for high-byte ID	Data out D[7:0]=PID
96-bit Unique Code read	x, x	0	0	0100	A[15:0] (0x0000~0x000B)	Data out D[7:0]=UID
16-bit VBG read	x, x	0	0	0100	A[15:0] (0x000C~0x000D)	Data out D[7:0]=VBG
128-bit Die Record read	x, x	0	0	0100	A[15:0] (0x0010~0x001F)	Data out D[7:0]=DR
128-bit UCID read	x, x	0	0	0100	A[15:0] (0x0020~0x02FB)	Data out D[7:0]=UCID
APROM page-erase	00	1	0	0010	Address in[2]	FFH
APROM byte-program	00	1	0	0001	Address in	Data in
APROM byte-read	00	0	0	0000	Address in	Data out
APROM Checksum Run	00	1	0	1110	IAPAL=0x00 IAPAH (starting address)	Data in (size x256bytes)
APROM Checksum Read	00	0	0	1110	IAPAL=0x00 IAPAH (starting address)	Checksum
LDROM page-erase	01	1	0	0010	Address in[2]	FFH
LDROM byte-read	01	0	0	0000	Address in	Data out
LDROM Checksum Run	01	1	0	1110	IAPAL=0x00 IAPAH (starting address)	Data in (size x256bytes)
LDROM Checksum Read	01	0	0	1110	IAPAL=0x00 IAPAH (starting address)	Checksum

IAP Mode	IAPCN				IAPA[15:0] {IAPAH, IAPAL}	IAPFD[7:0]
	IAPB [1:0]	FOEN	FCEN	FCTRL [3:0]		
All CONFIG bytes erase	11	1	0	0010	0000H	FFH
CONFIG byte-program	11	1	0	0001	CONFIG0: 0000H CONFIG1: 0001H CONFIG2: 0002H CONFIG4: 0004H CONFIG6: 0005H	Data in
CONFIG byte-read	11	0	0	0000	CONFIG0: 0000H CONFIG1: 0001H CONFIG2: 0002H CONFIG4: 0004H CONFIG6: 0005H	Data out
SPROM page-erase	10	1	0	0010	0180H	FFH
SPROM byte-program	10	1	0	0001	0180H~01FFH	Data in
SPROM byte-read	10	0	0	0000	0180H~01FFH	Data out
SPROM Erase	1, 0	1	0	0010	A[15:0]=0x0180	FFH
SPROM Program	1, 0	1	0	0001	A[15:0] (0x0180~0x01FF)	Data in D[7:0]
SPROM Read	1, 0	0	0	0000	A[15:0] (0x0180~0x01FF)	Data in D[7:0]
SPROM Checksum Run	10	1	0	1110	IAPAL=0x80 IAPAH =0x01 (starting address)	Data in
SPROM Checksum Read	10	0	0	1110	IAPAL=0x00 IAPAH =0x01 (starting address)	Checksum

Note:
[1] “X” means “don’t care”.
[2] Each page is 128 bytes size. Therefore, the address should be the address pointed to the target page.

Figure 6.4-1 IAP Modes and Command Codes

6.4.1.2 CRC-8 for Flash check

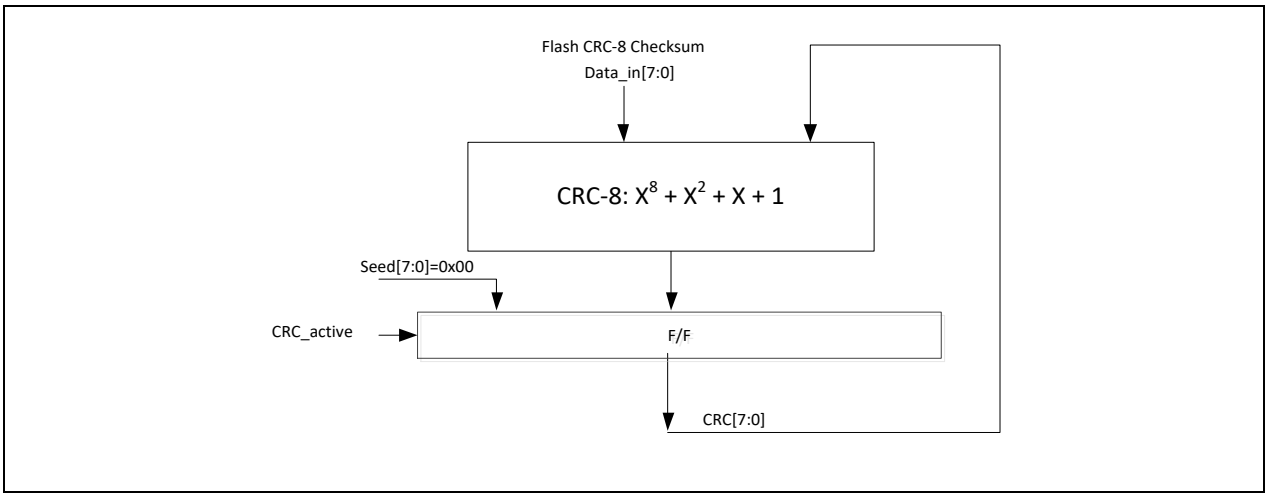


Figure 6.4-2. CRC-8 Block Diagram

6.4.1.3 IAP User Guide

IAP facilitates the updating Flash contents in a convenient way; however, user should follow some restricted laws in order that the IAP operates correctly. Without noticing warnings will possible cause undetermined results even serious damages of devices. Furthermore, this paragraph will also support useful suggestions during IAP procedures.

(1) If no more IAP operation is needed, user should clear IAPEN (CHPCON.0). It will make the system void to trigger IAP unaware. Furthermore, IAP requires the HIRC running. If the external clock source is selected, disabling IAP will stop the HIRC for saving power consumption. Note that a write to IAPEN is TA protected.

(2) When the LOCK bit (CONFIG0.1) is activated, IAP reading, writing, or erasing can still be valid.

During IAP progress, interrupts (if enabled) should be disabled temporally by clearing EA bit for implement limitation.

Do not attempt to erase or program to a page that the code is currently executing. This will cause unpredictable program behavior and may corrupt program data.

6.4.1.4 Using Flash Memory as Data Storage

In general application, there is a need of data storage, which is non-volatile so that it remains its content even after the power is off. Therefore, in general application user can read back or update the data, which rules as parameters or constants for system control. The Flash Memory array of the ML51/ML54/ML56 Series supports IAP function and any byte in the Flash Memory array may be read using the MOVC instruction and thus is suitable for use as non-volatile data storage. IAP provides erase and program function that makes it easy for one or more bytes within a page to be erased and programmed in a routine. IAP performs in the application under the control of the microcontroller's firmware. Be aware of Flash Memory writing endurance of 100,000 cycles. A demo is illustrated as follows.

Assembly demo code:

```
;*****************************************************************************
    ;   This code illustrates how to use IAP to make APROM 201h as a byte of
    ;   Data Flash when user code is executed in APROM.
;*****************************************************************************
    PAGE_ERASE_AP       EQU     00100010b
```

```
    BYTE_PROGRAM_AP     EQU     00100001b

     ORG   0000h

     MOV   TA,#0Aah              ;CHPCON is TA protected
     MOV   TA,#55h
     ORL   CHPCON,#00000001b     ;IAPEN = 1, enable IAP mode

     MOV   TA,#0Aah               ;IAPUEN is TA protected
     MOV   TA,#55h
     ORL   IAPUEN,#00000001b     ;APUEN = 1, enable APROM update

     MOV   IAPCN,#PAGE_ERASE_AP         ;Erase page 200h~27Fh
     MOV   IAPAH,#02h
     MOV   IAPAL,#00h
     MOV   IAPFD,#0FFh
     MOV   TA,#0Aah                     ;IAPTRG is TA protected
     MOV   TA,#55h
     ORL   IAPTRG,#00000001b            ;write '1' to IAPGO to trigger IAP process
     MOV   IAPCN,#BYTE_PROGRAM_AP       ;Program 201h with 55h
     MOV   IAPAH,#02h
     MOV   IAPAL,#01h
     MOV   IAPFD,#55h
     MOV   TA,#0Aah
     MOV   TA,#55h
     ORL   IAPTRG,#00000001b

     MOV   TA,#0Aah
     MOV   TA,#55h
     ANL   IAPUEN,#11111110b            ;APUEN = 0, disable APROM update

     MOV   TA,#0Aah
     MOV   TA,#55h
     ANL   CHPCON,#11111110b            ;IAPEN = 0, disable IAP mode

     MOV   DPTR,#201h
     CLR   A
     MOVC  A,@A+DPTR                    ;Read content of address 201h
     MOV   P0,A

     SJMP  $
```

C language demo code:

```
//***********************************************************************
// This code illustrates how to use IAP to make APROM 201h as a byte of
// Data Flash when user code is executed in APROM.
//***********************************************************************
    #define  PAGE_ERASE_AP   0x22
    #define  BYTE_PROGRAM_AP    0x21

  /*Data Flash, as part of APROM, is read by MOVC. Data Flash can be defined as128-
element array in "code" area from absolute address 0x0200            */
```

```
volatile unsigned char code Data_Flash[128] _at_ 0x0200;

Main (void)
{
 TA = 0Xaa;         //CHPCON is TA protected
 TA = 0x55;
 CHPCON |= 0x01;         //IAPEN = 1, enable IAP mode

 TA = 0Xaa;         //IAPUEN is TA protected
 TA = 0x55;
 IAPUEN |= 0x01;         //APUEN = 1, enable APROM update

 IAPCN = PAGE_ERASE_AP;     //Erase page 200h~27Fh
 IAPAH = 0x02;
 IAPAL = 0x00;
 IAPFD = 0Xff;
 TA = 0Xaa;         //IAPTRG is TA protected
 TA = 0x55;
 IAPTRG |= 0x01;      //write '1' to IAPGO to trigger IAP process

 IAPCN = BYTE_PROGRAM_AP;  // Program 201h with 55h
 IAPAH = 0x02;
 IAPAL = 0x01;
 IAPFD = 0x55;
 TA = 0Xaa;
 TA = 0x55;
 IAPTRG |= 0x01;      //write '1' to IAPGO to trigger IAP process

 TA = 0Xaa;         //IAPUEN is TA protected
 TA = 0x55;
 IAPUEN &= ~0x01;     //APUEN = 0, disable APROM update

 TA = 0Xaa;         //CHPCON is TA protected
 TA = 0x55;
 CHPCON &= ~0x01;     //IAPEN = 0, disable IAP mode

 P0 = Data_Flash[1];     //Read content of address 200h+1

 while(1);
}
```

6.4.1.5 In-System-Programming (ISP)

The Flash Memory supports both hardware programming and In-Application-Programming (IAP). If the product is just under development or the end product needs firmware updating in the hand of an end user, the hardware programming mode will make repeated programming difficult and inconvenient. In-System-Programming (ISP) makes it easy and possible. ISP performs Flash Memory updating without removing the microcontroller from the system. It allows a device to be re-programmed under software control. Furthermore, the capability to update the application firmware makes wide range of applications possible.

User can develop a custom Boot Code that resides in LDROM. The maximum size of LDROM is 4K Byte. User developed Boot Code can be re-programmed by parallel writer or In-Circuit-Programming (ICP) tool.

General speaking, an ISP is carried out by a communication between PC and MCU. PC transfers the new User Code to MCU through serial port. Then Boot Code receives it and re-programs into User Code through IAP commands. Nuvoton provides ISP firmware and PC application for ML51/ML54/ML56 Series. It makes user quite easy perform ISP through UART port. Please visit Nuvoton 8-bit Microcontroller website: Nuvoton 80C51 Microcontroller Technical Support. A simple ISP demo code is given below.

Assembly demo code:

```
;*****************************************************************************
;This code illustrates how to do APROM and CONFIG IAP from LDROM.
;APROM are re-programmed by the code to output P1 as 55h and P2 as aah.
;The CONFIG2 is also updated to disable BOD reset.
;User needs to configure CONFIG0 = 0x7F, CONFIG1 = 0Xfe, CONFIG2 = 0xFF.
;*****************************************************************************
     PAGE_ERASE_AP      EQU     00100010b
     BYTE_PROGRAM_AP       EQU     00100001b
     BYTE_READ_AP       EQU     00000000b
     ALL_ERASE_CONFIG      EQU     11100010b
     BYTE_PROGRAM_CONFIG   EQU     11100001b
     BYTE_READ_CONFIG      EQU     11000000b

      ORG   0000h

      CLR   EA         ;disable all interrupts
      CALL  Enable_IAP

      CALL  Enable_AP_Update
      CALL  Erase_AP        ;erase AP data
      CALL  Program_AP      ;programming AP data
      CALL  Disable_AP_Update
      CALL  Program_AP_Verify    ;verify Programmed AP data

      CALL  Read_CONFIG        ;read back CONFIG2
      CALL  Enable_CONFIG_Update
      CALL  Erase_CONFIG    ;erase CONFIG bytes
      CALL  Program_CONFIG     ;programming CONFIG2 with new data
      CALL  Disable_CONFIG_Update
      CALL  Program_CONFIG_Verify   ;verify Programmed CONFIG2

      CALL  Disable_IAP
      MOV   TA,#0Aah        ;TA protection
      MOV   TA,#55h      ;
      ANL   CHPCON,#11111101b    ;BS = 0, reset to APROM
      MOV   TA,#0Aah
      MOV   TA,#55h
      ORL   CHPCON,#80h        ;software reset and reboot from APROM

      SJMP  $

     ;*****************************************************************
     ;     IAP Subroutine
     ;*****************************************************************
     Enable_IAP:
```

```
    MOV   TA,#0Aah         ;CHPCON is TA protected
    MOV   TA,#55h
    ORL   CHPCON,#00000001b    ;IAPEN = 1, enable IAP mode
    RET

Disable_IAP:
    MOV   TA,#0Aah
    MOV   TA,#55h
    ANL   CHPCON,#11111110b    ;IAPEN = 0, disable IAP mode
    RET

Enable_AP_Update:
    MOV   TA,#0Aah         ;IAPUEN is TA protected
    MOV   TA,#55h
    ORL   IAPUEN,#00000001b    ;APUEN = 1, enable APROM update
    RET

Disable_AP_Update:
    MOV   TA,#0Aah
    MOV   TA,#55h
    ANL   IAPUEN,#11111110b    ;APUEN = 0, disable APROM update
    RET

Enable_CONFIG_Update:
    MOV   TA,#0Aah
    MOV   TA,#55h
    ORL   IAPUEN,#00000100b    ;CFUEN = 1, enable CONFIG update
    RET

Disable_CONFIG_Update:
    MOV   TA,#0Aah
    MOV   TA,#55h
    ANL   IAPUEN,#11111011b    ;CFUEN = 0, disable CONFIG update
    RET

Trigger_IAP:
    MOV   TA,#0Aah         ;IAPTRG is TA protected
    MOV   TA,#55h
    ORL   IAPTRG,#00000001b    ;write '1' to IAPGO to trigger IAP process
    RET

;*****************************************************************
;     IAP APROM Function
;*****************************************************************
Erase_AP:
    MOV   IAPCN,#PAGE_ERASE_AP
    MOV   IAPFD,#0FFh
    MOV   R0,#00h
Erase_AP_Loop:
    MOV   IAPAH,R0
    MOV   IAPAL,#00h
    CALL  Trigger_IAP
```

```
    MOV   IAPAL,#80h
    CALL  Trigger_IAP
    INC   R0
    CJNE  R0,#44h,Erase_AP_Loop
    RET

Program_AP:
    MOV   IAPCN,#BYTE_PROGRAM_AP
    MOV   IAPAH,#00h
    MOV   IAPAL,#00h
    MOV   DPTR,#AP_code
Program_AP_Loop:
    CLR   A
    MOVC  A,@A+DPTR
    MOV   IAPFD,A
    CALL  Trigger_IAP
    INC   DPTR
    INC   IAPAL
    MOV   A,IAPAL
    CJNE  A,#14,Program_AP_Loop
    RET

Program_AP_Verify:
    MOV   IAPCN,#BYTE_READ_AP
    MOV   IAPAH,#00h
    MOV   IAPAL,#00h
    MOV   DPTR,#AP_code
Program_AP_Verify_Loop:
    CALL  Trigger_IAP
    CLR   A
    MOVC  A,@A+DPTR
    MOV   B,A
    MOV   A,IAPFD
    CJNE  A,B,Program_AP_Verify_Error
    INC   DPTR
    INC   IAPAL
    MOV   A,IAPAL
    CJNE  A,#14,Program_AP_Verify_Loop
    RET

Program_AP_Verify_Error:
    CALL  Disable_IAP
    MOV   P0,#00h
    SJMP  $

;*****************************************************************
;      IAP CONFIG Function
;*****************************************************************
Erase_CONFIG:
    MOV   IAPCN,#ALL_ERASE_CONFIG
    MOV   IAPAH,#00h
    MOV   IAPAL,#00h
```

```
    MOV   IAPFD,#0FFh
    CALL  Trigger_IAP
    RET

Read_CONFIG:
    MOV   IAPCN,#BYTE_READ_CONFIG
    MOV   IAPAH,#00h
    MOV   IAPAL,#02h
    CALL  Trigger_IAP
    MOV   R7,IAPFD
    RET

Program_CONFIG:
    MOV   IAPCN,#BYTE_PROGRAM_CONFIG
    MOV   IAPAH,#00h
    MOV   IAPAL,#02h
    MOV   A,R7
    ANL   A,#11111011b
    MOV   IAPFD,A            ;disable BOD reset
    MOV   R6,A               ;temp data
    CALL  Trigger_IAP
    RET

Program_CONFIG_Verify:
    MOV   IAPCN,#BYTE_READ_CONFIG
    MOV   IAPAH,#00h
    MOV   IAPAL,#02h
    CALL  Trigger_IAP
    MOV   B,R6
    MOV   A,IAPFD
    CJNE  A,B,Program_CONFIG_Verify_Error
    RET

Program_CONFIG_Verify_Error:
    CALL  Disable_IAP
    MOV   P0,#00h
    SJMP  $

;*****************************************************************
;     APROM code
;*****************************************************************
AP_code:
    DB 75h,0B1h, 00h      ;OPCODEs of "MOV    P0M1,#0"
    DB 75h,0Ach, 00h      ;OPCODEs of "MOV    P3M1,#0"
    DB 75h,  90h, 55h     ;OPCODEs of "MOV    P1,#55h"
    DB 75h,0A0h,0Aah      ;OPCODEs of "MOV    P2,#0Aah"
    DB 80h,0Feh           ;OPCODEs of "SJMP   $"

    END
```

6.4.2 In-Circuit-Programming (ICP)

The Flash Memory can be programmed by "In-Circuit-Programming" (ICP). If the product is just under

development or the end product needs firmware updating in the hand of an end customer, the hardware programming mode will make repeated programming difficult and inconvenient. ICP method makes it easy and possible without removing the microcontroller from the system. ICP mode also allows customers to manufacture circuit boards with un-programmed devices. Programming can be done after the assembly process allowing the device to be programmed with the most recent firmware or a customized firmware.

There are three signal pins, $\overline{RST}$, ICPDA, and ICPCK, involved in ICP function. $\overline{RST}$ is used to enter or exit ICP mode. ICPDA is the data input and output pin. ICPCK is the clock input pin, which synchronizes the data shifted in to or out from MCU under programming. User should leave these three pins plus V_{DD} and GND pins on the circuit board to make ICP possible.

Nuvoton provides ICP tool for ML51/ML54/ML56 Series, which enables user to easily perform ICP through Nuvoton ICP programmer. The ICP programmer developed by Nuvoton has been optimized according to the electric characteristics of MCU. It also satisfies the stability and efficiency during production progress. For more details, please visit Nuvoton 8-bit Microcontroller website: Nuvoton 80C51 Microcontroller Technical Support.

6.4.3 On-Chip-Debugger (ICE)

6.4.3.1 Functional Description

The ML51/ML54/ML56 Series is embedded in an on-chip-debugger (OCD) providing developers with a low cost method for debugging user code, which is available on each package. The OCD gives debug capability of complete program flow control with eight hardware address breakpoints, single step, free running, and non-intrusive commands for memory access. The OCD system does not occupy any locations in the memory map and does not share any on-chip peripherals.

When the OCDEN (CONFIG0.4) is programmed as 0 and LOCK (CONFIG0.1) remains un-programmed as 1, the OCD is activated. The OCD cannot operate if chip is locked. The OCD system uses a two-wire serial interface, OCDDA and OCDCK, to establish communication between the target device and the controlling debugger host. OCDDA is an input/output pin for debug data transfer and OCDCK is an input pin for synchronization with OCDDA data. The $\overline{RST}$ pin is also necessary for OCD mode entry and exit. The ML51/ML54/ML56 Series supports OCD with Flash Memory control path by ICP writer mode, which shares the same three pins of OCD interface.

The ML51/ML54/ML56 Series uses OCDDA, OCDCK, and $\overline{RST}$ pins to interface with the OCD system. When designing a system where OCD will be used, the following restrictions must be considered for correct operation:

1. $\overline{RST}$cannot be connected directly to V_{DD} and any external capacitors connected must be removed.

2. All external reset sources must be disconnected.

3. Any external component connected on OCDDA and OCDCK must be isolated.

6.4.3.2 Limitation of OCD

The ML51/ML54/ML56 Series is a fully-featured microcontroller that multiplexes several functions on its limited I/O pins. Some device functionality must be sacrificed to provide resources for OCD system. The OCD has the following limitations:

1. The $\overline{RST}$ pin needs to be used for OCD mode selection.

2. The OCDDA pin is physically located on the same pin P5.0. Therefore, neither its I/O function nor shared multi-functions can be emulated.

3. The OCDCK pin is physically located on the same pin as P5.1. Therefore, neither its I/O function nor shared multi-functions can be emulated.

4. When the system is in Idle or Power-down mode, it is invalid to perform any accesses because parts of the device may not be clocked. A read access could return garbage or a write access might not succeed.

5. HIRC cannot be turned off because OCD uses this clock to monitor its internal status. The instruction that turns off HIRC affects nothing if executing under debug mode. When CPU enters its Power-down mode under debug mode, HIRC keeps turning on.

The ML51/ML54/ML56 Series OCD system has another limitation that non-intrusive commands cannot be executed at any time while the user's program is running. Non-intrusive commands allow a user to read or write MCU memory locations or access status and Register Description with the debug controller. A reading or writing memory or control register space is allowed only when MCU is under halt condition after a matching of the hardware address breakpoint or a single step running.

CONFIG0

7	6	5	4	3	2	1	0
CBS	**FSYS**	**OCDPWM**	**OCDEN**	**-**	**-**	**LOCK**	**-**
R/W	R/W	R/W	R/W	-	-	R/W	-

Factory default value: 1111 1111b

Bit	Name	Description
[5]	**OCDPWM**	**PWM Output State Under OCD Halt** This bit decides the output state of PWM when OCD halts CPU. 1 = Tri-state pins those are used as PWM outputs. 0 = PWM continues.
[4]	**OCDEN**	**OCD Enable** 1 = OCD Disabled. 0 = OCD Enabled. **Note:** If MCU run in OCD debug mode and OCDEN = 0, hard fault reset will disable. Only HardF flag be asserted.

Following is the OCD Relation multi-function pin define list. As default when CONFIG define OCD enabled, P5.1 and P5.0 pin auto setting as ICE_CLK and ICE_DAT function.

Group	Pin Name	GPIO	MFP	Type	Description
ICE	ICE_CLK	P5.1	MFP14	I	Serial wired debugger clock pin.
	ICE_DAT	P5.0	MFP14	O	Serial wired debugger data pin.

6.5 GPIO Port Structure and Operation

6.5.1 GPIO Mode

The ML51/ML54/ML56 Series has a maximum of 56 general purpose I/O pins which 40 bit-addressable general I/O pins grouped as 5 ports, P0 to P4, and 16 general I/O pins grouped as P5 and P6. Each port has its port control register (Px register). The writing and reading of a port control register have different meanings. A write to port control register sets the port output latch logic value, where as a read gets the port pin logic state. These four modes are quasi-bidirectional (standard 8051 port structure), push-pull, input-only, and open-drain modes. Each port spends two special function registers PxM1 and PxM2 to select the I/O mode of port Px. The list below illustrates how to select the I/O mode of Px.n. Note that the default configuration of is input-only (high-impedance) after any reset.

PnM1.X[1]	PnM2.X[1]	I/O Type
0	0	Quasi-bidirectional
0	1	Push-pull
1	0	Input-only (high-impedance)
1	1	Open-drain
Note: N = 0~5, x = 0~7		

Table 6.5-1 Configuration for Different I/O Modes

All I/O pins can be selected as TTL level inputs or Schmitt triggered inputs by selecting corresponding bit in PxS register. Schmitt triggered input has better glitch suppression capability. All I/O pins also have bit-controllable, slew rate select ability via software. The Register Description are PxSR. By default, the slew rate is slow. If user would like to increase the I/O output speed, setting the corresponding bit in PxSR, the slew rate is selected in a faster level.

For example:

```
    P0M1 |= 0x40;
    P0M2 &= 0xBF;    //Set P0.6 as input only mode
```

Quasi-Bidirectional Mode

The quasi-bidirectional mode, as the standard 8051 I/O structure, can rule as both input and output. When the port outputs a logic high, it is weakly driven, allowing an external device to pull the pin low. When the pin is pulled low, it is driven strongly and able to sink a large current. In the quasi-bidirectional I/O structure, there are two pull-high transistors. Each of them serves different purposes. One of these pull-highs, called the “very weak” pull-high, is turned on whenever the port latch contains logic 1. The “very weak” pull-high sources a very small current that will pull the pin high if it is left floating.

The second pull-high is the “strong” pull-high. This pull-high is used to speed up 0-to-1 transitions on a quasi-bidirectional port pin when the port latch changes from logic 0 to logic 1. When this occurs, the strong pull-high turns on for two-CPU-clock time to pull the port pin high quickly. Then it turns off “very weak” pull-highs continue remaining the port pin high. The quasi-bidirectional port structure is shown below.

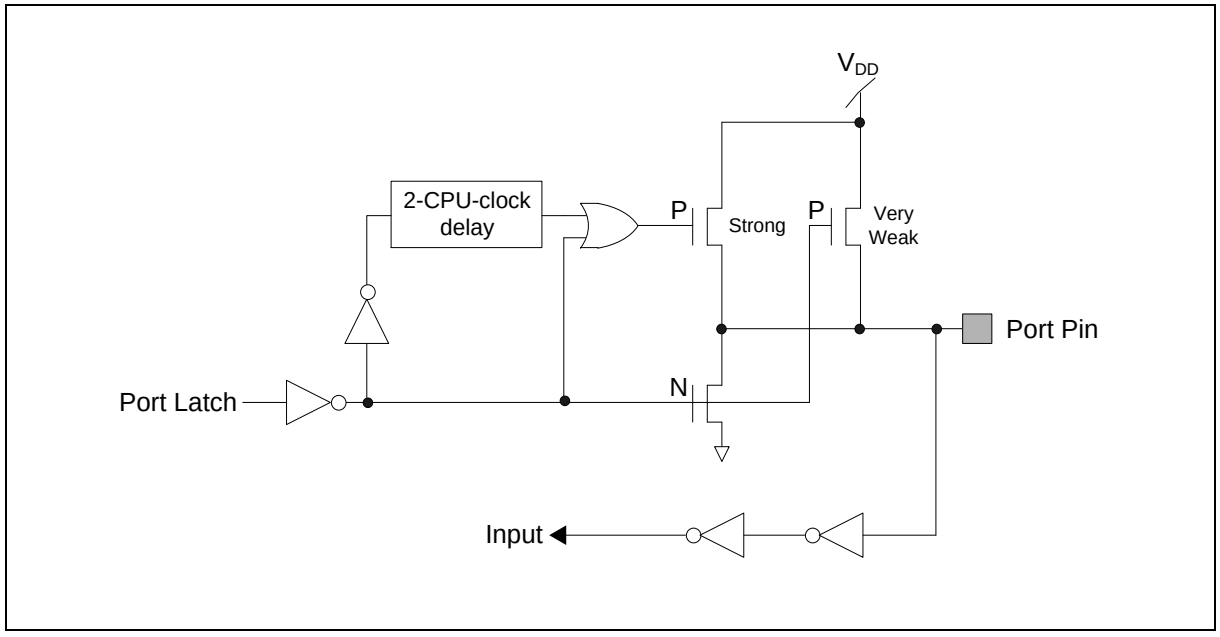


Figure 6.5-1 Quasi-Bidirectional Mode Structure

Push-Pull Mode

The push-pull mode has the same pull-low structure as the quasi-bidirectional mode, but provides a continuous strong pull-high when the port latch is written by logic 1. The push-pull mode is generally used as output pin when more source current is needed for an output driving.

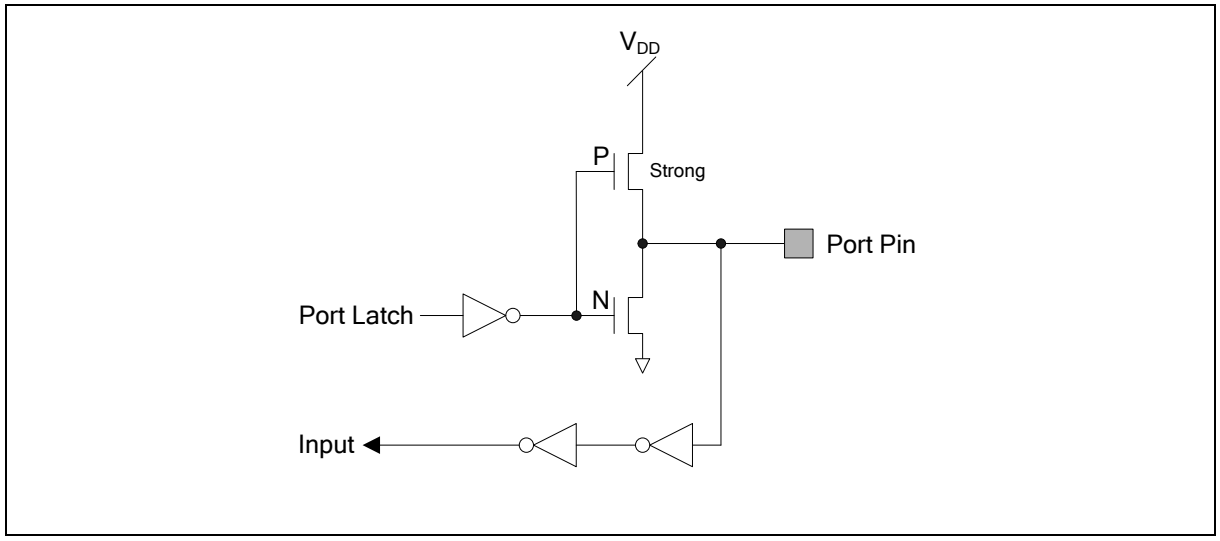


Figure 6.5-2 Push-Pull Mode Structure

Input-Only Mode

Input-only mode provides true high-impedance input path. Although a quasi-bidirectional mode I/O can also be an input pin, but it requires relative strong input source. Input-only mode also benefits to power consumption reduction for logic 0 input always consumes current from V_{DD} if in quasi-bidirectional mode. User needs to take care that an input-only mode pin should be given with a determined voltage level by external devices or resistors. A floating pin will induce leakage current especially in Power-down mode.

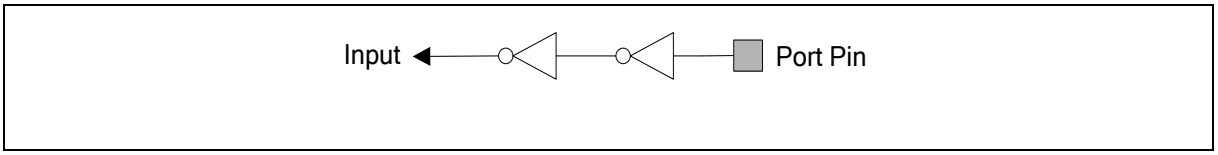


Figure 6.5-3 Input-Only Mode Structure

Open-Drain Mode

The open-drain mode turns off all pull-high transistors and only drives the pull-low of the port pin when the port latch is given by logic 0. If the port latch is logic 1, it behaves as if in input-only mode. To be used as an output pin generally as R/W lines, an open-drain pin should add an external pull-high, typically a resistor tied to V_{DD}. User needs to take care that an open-drain pin with its port latch as logic 1 should be given with a determined voltage level by external devices or resistors. A floating pin will induce leakage current especially in Power-down mode.

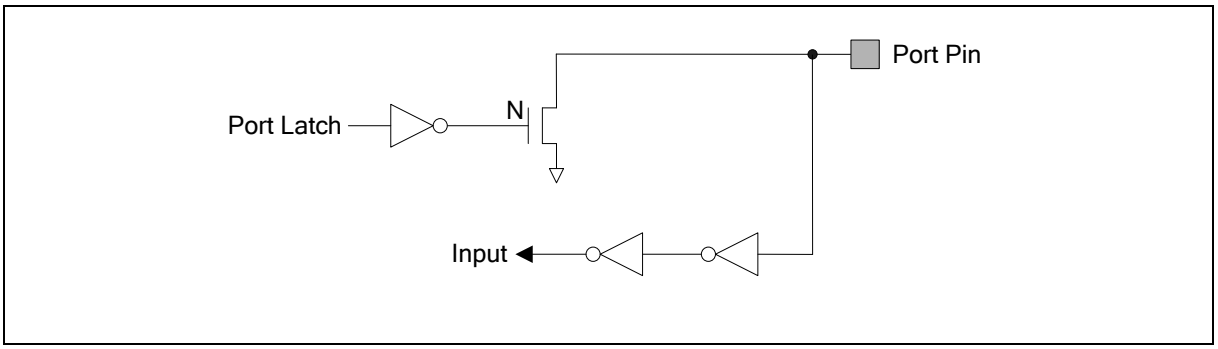


Figure 6.5-4 Open-Drain Mode Structure

6.5.1.2 Input and Output Data Control

These registers are I/O input and output data buffers. Reading gets the I/O input data. Writing forces the data output. All of these registers are bit-addressable.

Pn – Port

Register	SFR Address	Reset Value
P0	80H, All pages, Bit-addressable	1111_1111 b
P1	90H, All pages, Bit-addressable	1111_1111 b
P2	A0H, All pages, Bit-addressable	1111_1111 b
P3	B0H, All pages, Bit-addressable	1111_1111 b
P4	D8H, All pages, Bit-addressable	1111_1111 b
P5	B1H, Page 0	1111_1111 b
P6	A7H, Page 2	1111_1111 b

7	6	5	4	3	2	1	0
Pn.7	**Pn.6**	**Pn.5**	**Pn.4**	**Pn.3**	**Pn.2**	**Pn.1**	**Pn.0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:0]	**Pn[7:0]**	**Port n** Port n is an maximum 8-bit general purpose I/O port.

6.5.1.3 GPIO Mode Control

These registers control GPIO mode, which is configurable among four modes: input-only, quasi-bidirectional, push-pull, or open-drain. Each pin can be configured individually.

As default after reset all GPIO setting as input only mode.

PnM1 – Port n Mode Select 1

Register	SFR Address	Reset Value
P0M1	B1H, Page 1	1111_1111 b
P1M1	B3H, Page 1	1111_1111 b
P2M1	85H, Page 1	1111_1111 b
P3M1	C2H, Page 1	1111_1111 b
P4M1	B9H, Page 1	1111_1111 b
P5M1	BDH, Page 1	1111_1111 b
P6M1	84H, Page 2	1111_1111 b

7	6	5	4	3	2	1	0
P0M1_7	**P0M1_6**	**P0M1_5**	**P0M1_4**	**P0M1_3**	**P0M1_2**	**P0M1_1**	**P0M1_0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:0]	**PnM1[7:0]**	Port n mode select 1

PnM2 – Port n Mode Select 2

Register	SFR Address	Reset Value
P0M2	B2H, Page 1	0000_0000 b
P1M2	B4H, Page 1	0000_0000 b
P2M2	B6H, Page 1	0000_0000 b
P3M2	C3H, Page 1	0000_0000 b
P4M2	BAH, Page 1	0000_0000 b
P5M2	BEH, Page 1	0000_0000 b
P6M2	85H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PnM2_7	**PnM2_6**	**PnM2_5**	**PnM2_4**	**PnM2_3**	**PnM2_2**	**PnM2_1**	**PnM2_0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:0]	**PnM2[7:0]**	**Port n Mode Select 2** **Note:** PxM1 and PxM2 are used in combination to determine the I/O mode of each pin of Port. See Table 6.5-1 Configuration for Different I/O Modes.

6.5.1.4 GPIO Multi-Function Select

PnMF10 – Pn.1 and Pn.0 Multi-function Select

Register	SFR Address	Reset Value
P0MF10	F9H, Page 2	0000_0000 b
P1MF10	FDH, Page 2	0000_0000 b
P2MF10	F2H, Page 2	0000_0000 b
P3MF10	F6H, Page 2	0000_0000 b
P4MF10	EBH, Page 2	0000_0000 b
P5MF10	EFH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PnMF1[3:0]				**PnMF0[3:0]**			
R/W				R/W			

Bit	Name	Description
[7:4]	**PnMF1[3:0]**	Pn.1 multi-function select
[3:0]	**PnMF0[3:0]**	Pn.0 multi-function select

PnMF32 – Pn.3 and Pn.2 Multi-function Select

Register	SFR Address	Reset Value
P0MF32	FAH, Page 2	0000_0000 b
P1MF32	FEH, Page 2	0000_0000 b
P2MF32	F3H, Page 2	0000_0000 b
P3MF32	F7H, Page 2	0000_0000 b
P4MF32	ECH, Page 2	0000_0000 b
P5MF32	E1H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PnMF3[3:0]				**PnMF2[3:0]**			
R/W				R/W			

Bit	Name	Description
[7:4]	**PnMF3[3:0]**	Pn.3 multi-function select
[3:0]	**PnMF2[3:0]**	Pn.2 multi-function select

PnMF54 – Pn.5 and Pn.4 Multi-function Select

Register	SFR Address	Reset Value
P0MF54	FBH, Page 2	0000_0000 b
P1MF54	FFH, Page 2	0000_0000 b
P2MF54	F4H, Page 2	0000_0000 b
P3MF54	E9H, Page 2	0000_0000 b
P4MF54	EDH, Page 2	0000_0000 b
P5MF54	E2H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PnMF5[3:0]				**PnMF4[3:0]**			
R/W				R/W			

Bit	Name	Description
[7:4]	**PnMF5[3:0]**	Pn.5 multi-function select
[3:0]	**PnMF4[3:0]**	Pn.4 multi-function select

PnMF76 – Pn.7 and Pn.6 Multi-function Select

Register	SFR Address	Reset Value
P0MF76	FCH, Page 2	0000_0000 b
P1MF76	F1H, Page 2	0000_0000 b
P2MF76	F5H, Page 2	0000_0000 b
P3MF76	EAH, Page 2	0000_0000 b
P4MF76	EEH, Page 2	0000_0000 b
P5MF76	E3H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PnMF7[3:0]				**PnMF6[3:0]**			
R/W				R/W			

Bit	Name	Description
[7:4]	**PnMF7[3:0]**	Pn.7 multi-function select
[3:0]	**PnMF6[3:0]**	Pn.6 multi-function select

For example: As list P2.5 can define as following as anyone of the list function. If want to define as PWM0_CH0 output, please setting P2MF54 |= 0xB0.

6.5.1.5 Input Type

Each I/O pin can be configured individually as TTL input or Schmitt triggered input. Note that all of PxS registers are accessible by switching SFR page to Page 1.

PnS – Port n Schmitt Triggered Input

Register	SFR Address	Reset Value
P0S	99H, Page 1	0000_0000 b
P1S	9BH, Page 1	0000_0000 b
P2S	9DH, Page 1	0000_0000 b
P3S	ACH, Page 1	0000_0000 b
P4S	BBH, Page 1	0000_0000 b
P5S	BFH, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
PnS.7	**PnS.6**	**PnS.5**	**PnS.4**	**PnS.3**	**PnS.2**	**PnS.1**	**PnS.0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:0]	**PnS[7:0]**	**Pn Schmitt Triggered Input** 0 = TTL level input of Pn.x. 1 = Schmitt triggered input of Pn.x.

6.5.1.6 Output Slew Rate Control

Slew rate for each I/O pin is configurable individually. By default, each pin is in normal slew rate mode. User can set each control register bit to enable high-speed slew rate for the corresponding I/O pin. Note that all PxSR registers are accessible by switching SFR page to Page 1.

PnSR –Port n Slew Rate Control

Register	SFR Address	Reset Value
P0SR	9AH, Page 1	0000_0000 b
P1SR	9CH, Page 1	0000_0000 b
P2SR	9EH, Page 1	0000_0000 b
P3SR	ADH, Page 1	0000_0000 b
P4SR	BCH, Page 1	0000_0000 b
P5SR	AEH, Page 1	0000_0000 b
P6SR	8EH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PnSR.7	**PnSR.6**	**PnSR.5**	**PnSR.4**	**PnSR.3**	**PnSR.2**	**PnSR.1**	**PnSR.0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:0]	**PnSR[7:0]**	**Pn.x Slew Rate** 0 = Pn.x normal output slew rate. 1 = Pn.x high-speed output slew rate.

6.5.1.7 Pull-Up Resister Control

Pull up resister for each I/O pin is configurable individually. But even enabled the pull up resister only effect when GPIO setting as input mode. By default, after reset each pin pull high resister is disabled.

PnUP – Port n Pull-up Resister Control

Register	SFR Address	Reset Value
P0UP	92H, Page 1	0000_0000 b
P1UP	93H, Page 1	0000_0000 b
P2UP	94H, Page 1	0000_0000 b
P3UP	95H, Page 1	0000_0000 b
P4UP	96H, Page 1	0000_0000 b
P5UP	97H, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
PnUP.7	**PnUP.6**	**PnUP.5**	**PnUP.4**	**PnUP.3**	**PnUP.2**	**PnUP.1**	**PnUP.0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:0]	**PnUP[7:0]**	**P0.n Pull-Up Enable** 0 = Pn.x pull-up Disabled. 1 = Pn.x pull-up Enabled.

6.5.1.8 Pull-Down Resister Control

Pull down resister for each I/O pin is configurable individually. Even enabled the pull down resister only effect when GPIO setting as input mode. By default, after reset each pin pull high resister is disabled.

PnDW – Port n Pull-down Resister Control

Register	SFR Address	Reset Value
P0DW	8AH, Page 1	0000_0000 b
P1DW	8BH, Page 1	0000_0000 b
P2DW	8CH, Page 1	0000_0000 b
P3DW	8DH, Page 1	0000_0000 b
P4DW	8EH, Page 1	0000_0000 b
P5DW	8FH, Page 1	0000_0000 b
P6DW	8FH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PnDW.7	**PnDW.6**	**PnDW.5**	**PnDW.4**	**PnDW.3**	**PnDW.2**	**PnDW.1**	**PnDW.0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:0]	**PnDW[7:0]**	**Pn.x Pull-Down Enable** 0 = Pn.x pull-down Disabled. 1 = Pn.x pull-down Enabled.

6.5.2 External Interrupt Pins

Following is the MFP define ofr extneral interrupt pins.

Group	Pin Name	GPIO	MFP	Type	Description
INT0	INT0	P2.5	MFP15	I	External interrupt 0 input pin.
		P0.6	MFP15	I	
		P4.6	MFP15	I	
INT1	INT1	P2.4	MFP15	I	External interrupt 1 input pin.
		P0.7	MFP15	I	
		P3.6	MFP15	I	
		P4.0	MFP15	I	

Table 6.5-2 External Interrupt Pin Multi-Function Pin List

The external interrupt INT0 and INT1 can be used as interrupt sources. They are selectable to be either edge or level triggered depending on bits IT0 (TCON.0) and IT1 (TCON.2). The bits IE0 (TCON.1) and IE1 (TCON.3) are the flags those are checked to generate the interrupt. In the edge triggered mode, the INT0 or INT1 inputs are sampled every system clock cycle. If the sample is high in one cycle and low in the next, then a high to low transition is detected and the interrupts request flag IE0 or IE1 will be set. Since the external interrupts are sampled every system clock, they have to be held high or low for at least one system clock cycle. The IE0 and IE1 are automatically cleared when the interrupt service routine is called. If the level triggered mode is selected, then the requesting source has to hold the pin low till the interrupt is serviced. The IE0 and IE1 will not be cleared by the hardware on entering the service routine. In the level triggered mode, IE0 and IE1 follows the inverse value of INT0 and INT1 pins. If interrupt pins continue to be held low even after the service routine is completed, the processor will acknowledge another interrupt request from the same source. Both INT0 and INT1 can wake up the device from the Power-down mode.

TCON – Timer 0 and 1 Control (Bit-addressable)

Register	SFR Address	Reset Value
TCON	88H, All pages, Bit-addressable	0000_0000b

7	6	5	4	3	2	1	0
TF1	**TR1**	**TF0**	**TR0**	**IE1**	**IT1**	**IE0**	**IT0**
R/W	R/W	R/W	R/W	R (level) R/W (edge)	R/W	R (level) R/W (edge)	R/W

Bit	Name	Description
[3]	**IE1**	**External Interrupt 1 Edge Flag** If IT1 = 1 (falling edge trigger), this flag will be set by hardware when a falling edge is detected. It remain set until cleared via software or cleared by hardware in the beginning of its interrupt service routine. If IT1 = 0 (low level trigger), this flag follows the inverse of the INT1 input signal's logic level. Software cannot control it.
[2]	**IT1**	**External Interrupt 1 Type Select** This bit selects by which type that INT1 is triggered. 0 = INT1 is low level triggered. 1 = INT1 is falling edge triggered.
[1]	**IE0**	**External Interrupt 0 Edge Flag** If IT0 = 1 (falling edge trigger), this flag will be set by hardware when a falling edge is detected. It remain set until cleared via software or cleared by hardware in the beginning of its interrupt service routine. If IT0 = 0 (low level trigger), this flag follows the inverse of the INT0 input signal's logic level. Software cannot control it.
[0]	**IT0**	**External Interrupt 0 Type Select** This bit selects by which type that INT0 is triggered. 0 = INT0 is low level triggered. 1 = INT0 is falling edge triggered.

6.5.3 Pin Interrupt (PIT)

The ML51/ML54/ML56 Series provides pin interrupt input for each I/O pin to detect pin state if button or keypad set is used. A maximum 8-channel pin interrupt detection can be assigned by I/O port sharing. The pin interrupt is generated when any key is pressed on a keyboard or keypad, which produces an edge or level triggering event. Pin interrupt may be used to wake the CPU up from Idle or Power-down mode.

Each channel of pin interrupt can be enabled and polarity controlled independently by PIPEN and PINEN register. PICON selects which port that the pin interrupt is active. It also defines which type of pin interrupt is used – level detect or edge detect. Each channel also has its own interrupt flag. There are total eight pin interrupt flags located in PIF register. The respective flags for each pin interrupt channel allow the interrupt service routine to poll on which channel on which the interrupt event occurs. All flags in PIF register are set by hardware and should be cleared by software.

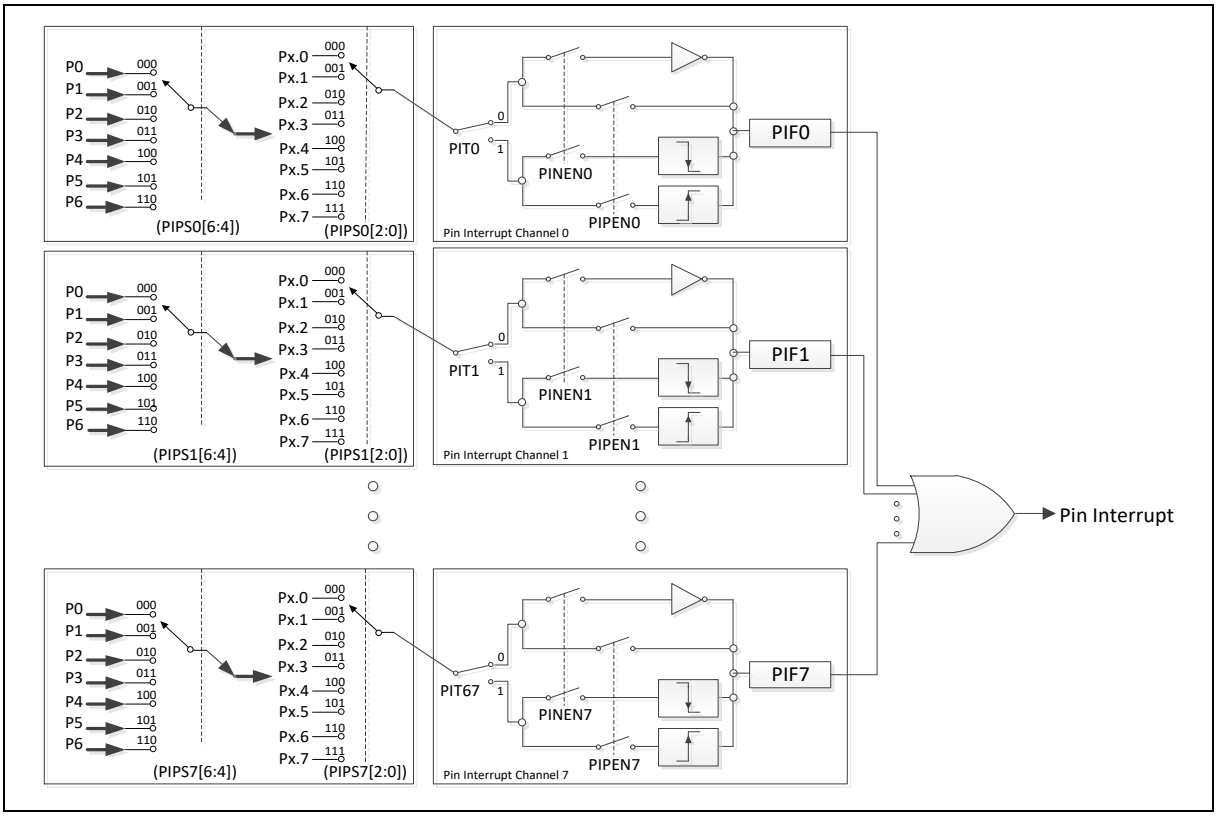


Figure 6.5-5 Pin Interface Block Diagram

Pin interrupt is generally used to detect an edge transient from peripheral devices like keyboard or keypad. During idle state, the system prefers to enter Power-down mode to minimize power consumption and waits for event trigger. Pin interrupt can wake up the device from Power-down mode.

PICON – Pin Interrupt Control

Register	SFR Address	Reset Value
PICON	E9H, Page 1	0011 _0100 b

7	6	5	4	3	2	1	0
PIT7	PIT6	PIT5	PIT4	PIT3	PIT2	PIT1	PIT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	PIT7	**Pin Interrupt Channel 7 Type Select** This bit selects which type that pin interrupt channel 7 is triggered. 0 = Level triggered. 1 = Edge triggered.
[6]	PIT6	**Pin Interrupt Channel 6 Type Select** This bit selects which type that pin interrupt channel 6 is triggered. 0 = Level triggered. 1 = Edge triggered.
[5]	PIT5	**Pin Interrupt Channel 5 Type Select** This bit selects which type that pin interrupt channel 5 is triggered. 0 = Level triggered. 1 = Edge triggered.
[4]	PIT4	**Pin Interrupt Channel 4 Type Select** This bit selects which type that pin interrupt channel 4 is triggered. 0 = Level triggered. 1 = Edge triggered.
[3]	PIT3	**Pin Interrupt Channel 3 Type Select** This bit selects which type that pin interrupt channel 3 is triggered. 0 = Level triggered. 1 = Edge triggered.
[2]	PIT2	**Pin Interrupt Channel 2 Type Select** This bit selects which type that pin interrupt channel 2 is triggered. 0 = Level triggered. 1 = Edge triggered.
[1]	PIT1	**Pin Interrupt Channel 1 Type Select** This bit selects which type that pin interrupt channel 1 is triggered. 0 = Level triggered. 1 = Edge triggered.

Bit	Name	Description
[0]	**PIT0**	**Pin Interrupt Channel 0 Type Select** This bit selects which type that pin interrupt channel 0 is triggered. 0 = Level triggered. 1 = Edge triggered.

PINEN – Pin Interrupt Negative Polarity Enable

Register	SFR Address	Reset Value
PINEN	EAH, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
PINEN7	**PINEN6**	**PINEN5**	**PINEN4**	**PINEN3**	**PINEN2**	**PINEN1**	**PINEN0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:0]	**PINENn**	**Pin Interrupt Channel n Negative Polarity Enable** This bit enables low-level/falling edge triggering pin interrupt channel n. The level or edge triggered selection depends on each control bit PITn in PICON. 0 = Low-level/falling edge detect Disabled. 1 = Low-level/falling edge detect Enabled.

PIPEN – Pin Interrupt Positive Polarity Enable

Register	SFR Address	Reset Value
PIPEN	EBH, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
PIPEN7	**PIPEN6**	**PIPEN5**	**PIPEN4**	**PIPEN3**	**PIPEN2**	**PIPEN1**	**PIPEN0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:0]	**PIPENn**	**Pin Interrupt Channel n Positive Polarity Enable** This bit enables high-level/rising edge triggering pin interrupt channel n. The level or edge triggered selection depends on each control bit PITn in PICON. 0 = High-level/rising edge detect Disabled. 1 = High-level/rising edge detect Enabled.

PIF – Pin Interrupt Flags

Register	SFR Address	Reset Value
PIF	CAH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
PIF7	**PIF6**	**PIF5**	**PIF4**	**PIF3**	**PIF2**	**PIF1**	**PIF0**
R (level) R/W (edge)	R (level) R/W (edge)	R (level) R/W (edge)	R (level) R/W (edge)	R (level) R/W (edge)	R (level) R/W (edge)	R (level) R/W (edge)	R (level) R/W (edge)

Bit	Name	Description
[7:0]	**PIFn**	**Pin Interrupt Channel n Flag** If the edge trigger is selected, this flag will be set by hardware if the channel n of pin interrupt detects an enabled edge trigger. This flag should be cleared by software. If the level trigger is selected, this flag follows the inverse of the input signal's logic level on the channel n of pin interrupt. Software cannot control it.

PIPSn – Pin Interrupt Control

Register	SFR Address	Reset Value
PIPS0	A1H, Page 1	0000_0000 b
PIPS1	A2H, Page 1	0000_0000 b
PIPS2	A3H, Page 1	0000_0000 b
PIPS3	A4H, Page 1	0000_0000 b
PIPS4	A5H, Page 1	0000_0000 b
PIPS5	A6H, Page 1	0000_0000 b
PIPS6	A7H, Page 1	0000_0000 b
PIPS7	AFH, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
-	**PSEL[2:0]**			**-**	**BSEL[2:0]**		
-	R/W			-	R/W		

Bit	Name	Description
[7]	**-**	Reserved
[6:4]	**PSEL[2:0]**	**Pin Interrupt Channel Port Select** 000 = P0 PORT. 001 = P1 PORT. 010 = P2 PORT. 011 = P3 PORT. 100 = P4 PORT. 101 = P5 PORT. 110 = P6 PORT. 111 = Reserved.
[3]	**-**	Reserved
[2:0]	**BSEL[2:0]**	**Pin Interrupt Channel Bit Select** 000 = Pn.0. 001 = Pn.1 010 =.Pn.2 011 = Pn.3. 100 = Pn.4. 101 = Pn.5. 110 = Pn.6. 111 = Pn.7. n is the PORT number, which is selected by PSEL[2:0].

6.6 Timer

6.6.1 Overview

ML51/ML54/ML56 Series provides following 16-bit Timer. Two 16-bit Timers/Counters 0 and 1 compatible with standard 8051. One 16-bit Timer 2 with three-channel input capture module and 9 input pin can be selected. One 16-bit auto-reload Timer 3, which can be the baud rate clock source of UARTs.

6.6.2 Timer/Counter 0 and 1

Timer/Counter 0 and 1 on ML51/ML54/ML56 Series are two 16-bit Timers/Counters. Each of them has two 8-bit registers those form the 16-bit counting register. For Timer/Counter 0 they are TH0, the upper 8-bit register, and TL0, the lower 8-bit register. Similarly Timer/Counter 1 has two 8-bit registers, TH1 and TL1. TCON and TMOD can configure modes of Timer/Counter 0 and 1.

The Timer or Counter function is selected by the C/$\overline{T}$ bit in TMOD. Each Timer/Counter has its own selection bit. TMOD.2 selects the function for Timer/Counter 0 and TMOD.6 selects the function for Timer/Counter 1

When configured as a “Timer”, the timer counts the system clock cycles. The timer clock is 1/12 of the system clock (F_{SYS}) for standard 8051 capability or direct the system clock for enhancement, which is selected by T0M (CKCON.3) bit for Timer 0 and T1M (CKCON.4) bit for Timer 1. In the “Counter” mode, the countering register increases on the falling edge of the external input pin T0. If the sampled value is high in one clock cycle and low in the next, a valid 1-to-0 transition is recognized on T0 or T1 pin.

The Timers 0 and 1 can be configured to automatically to toggle output whenever a timer overflow occurs. The same device pins that are used for the T0 and T1 count inputs are also used for the timer toggle outputs. This function is enabled by control bits T0OE and T1OE in the CKCON register, and apply to Timer 0 and Timer 1 respectively. The port outputs will be logic 1 prior to the first timer overflow when this mode is turned on. In order for this mode to function, the C/$\overline{T}$ bit should be cleared selecting the system clock as the clock source for the timer.

Note that the TH0 (TH1) and TL0 (TL1) are accessed separately. It is strongly recommended that in mode 0 or 1, user should stop Timer temporally by clearing TR0 (TR1) bit before reading from or writing to TH0 (TH1) and TL0 (TL1). The free-running reading or writing may cause unpredictable result.

6.6.2.1 Mode 0 (13-Bit Timer)

In Mode 0, the Timer/Counter is a 13-bit counter. The 13-bit counter consists of TH0 (TH1) and the five lower bits of TL0 (TL1). The upper three bits of TL0 (TL1) are ignored. The Timer/Counter is enabled when TR0 (TR1) is set and either GATE is 0 or INT0 (INT1) is 1. Gate setting as 1 allows the Timer to calculate the pulse width on external input pin INT0 (INT1). When the 13-bit value moves from 1FFFH to 0000H, the Timer overflow flag TF0 (TF1) is set and an interrupt occurs if enabled.

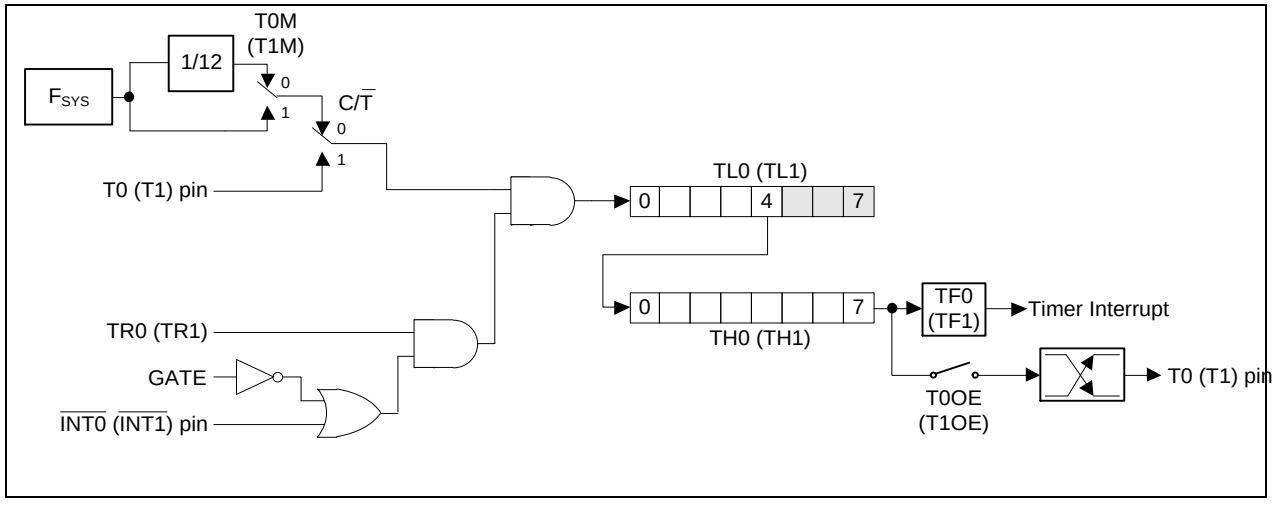


Figure 6.6-1 Timer/Counters 0 and 1 in Mode 0

6.6.2.2 Mode 1 (16-Bit Timer)

Mode 1 is similar to Mode 0 except that the counting registers are fully used as a 16-bit counter. Roll-over occurs when a count moves FFFFH to 0000H. The Timer overflow flag TF0 (TF1) of the relevant Timer/Counter is set and an interrupt will occurs if enabled.

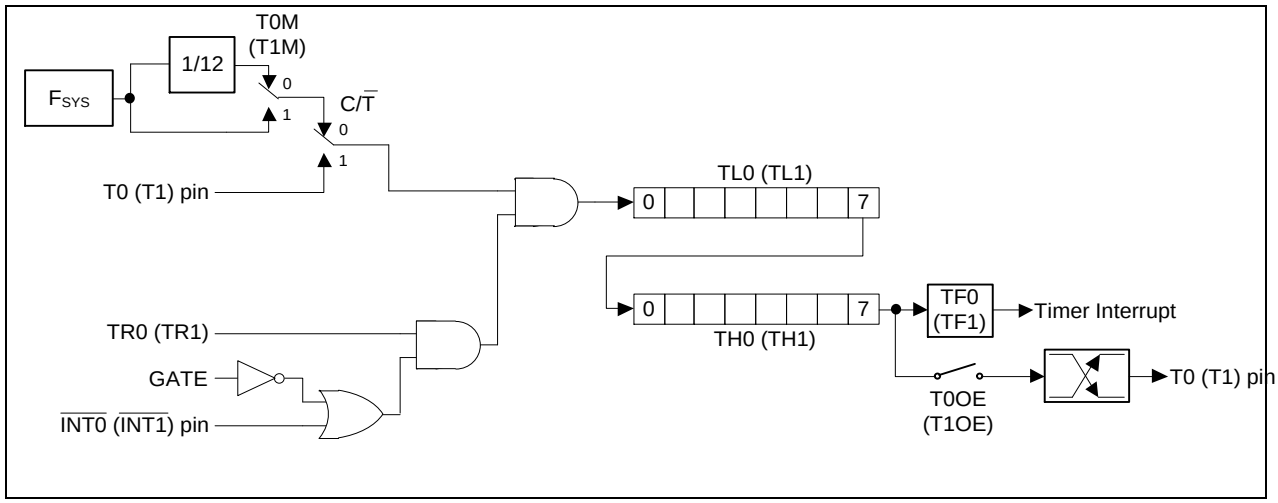


Figure 6.6-2 Timer/Counters 0 and 1 in Mode 1

6.6.2.3 Mode 2 (8-Bit Auto-Reload Timer)

In Mode 2, the Timer/Counter is in auto-reload mode. In this mode, TL0 (TL1) acts as an 8-bit count register whereas TH0 (TH1) holds the reload value. When the TL0 (TL1) register overflow, the TF0 (TF1) bit in TCON is set and TL0 (TL1) is reloaded with the contents of TH0 (TH1) and the counting process continues from here. The reload operation leaves the contents of the TH0 (TH1) register unchanged. This feature is best suitable for UART baud rate generator for it runs without continuous software intervention. Note that only Timer1 can be the baud rate source for UART. Counting is enabled by setting the TR0 (TR1) bit as 1 and proper setting of GATE and INT0 (INT1) pins. The functions of GATE and INT0 (INT1) pins are just the same as Mode 0 and 1.

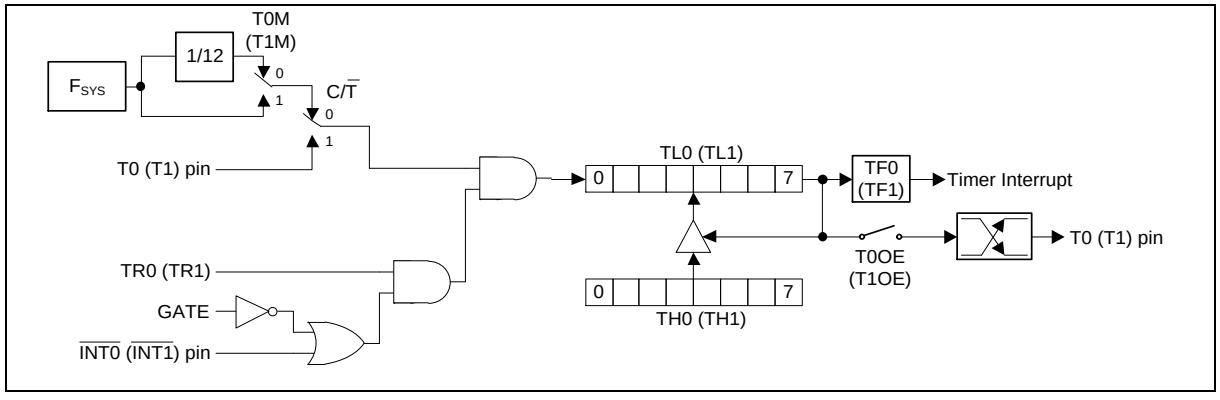


Figure 6.6-3 Timer/Counters 0 and 1 in Mode 2

6.6.2.4 Mode 3 (Two Separate 8-Bit Timers)

Mode 3 has different operating methods for Timer 0 and Timer 1. For Timer/Counter 1, Mode 3 simply freezes the counter. Timer/Counter 0, however, configures TL0 and TH0 as two separate 8 bit count registers in this mode. TL0 uses the Timer/Counter 0 control bits C/T, GATE, TR0, INT0, and TF0. The TL0 also can be used as a 1-to-0 transition counter on pin T0 as determined by C/T (TMOD.2). TH0 is forced as a clock cycle counter and takes over the usage of TR1 and TF1 from Timer/Counter 1. Mode 3

is used in case that an extra 8 bit timer is needed. If Timer/Counter 0 is configured in Mode 3, Timer/Counter 1 can be turned on or off by switching it out of or into its own Mode 3. It can still be used in Modes 0, 1 and 2 although its flexibility is restricted. It no longer has control over its overflow flag TF1 and the enable bit TR1. However Timer 1 can still be used as a Timer/Counter and retains the use of GATE, INT1 pin and T1M. It can be used as a baud rate generator for the serial port or other application not requiring an interrupt.

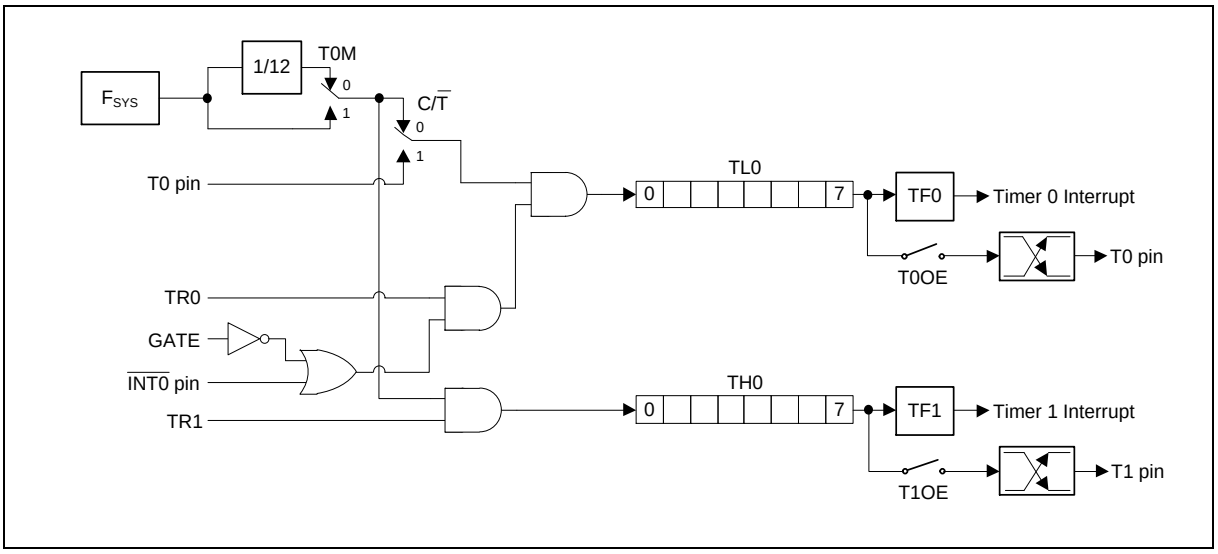


Figure 6.6-4 Timer/Counter 0 in Mode 3

6.6.2.5 Register Description

TMOD – Timer 0 and 1 Mode

Register	SFR Address	Reset Value
TMOD	89H, All pages	0000_0000b

7	6	5	4	3	2	1	0
GATE	**C/$\overline{T}$**	**M1**	**M0**	**GATE**	**C/$\overline{T}$**	**M1**	**M0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**GATE**	**Timer 1 Gate Control** 0 = Timer 1 will clock when TR1 is 1 regardless of INT1 logic level. 1 = Timer 1 will clock only when TR1 is 1 and INT1 is logic 1.
[6]	**C/$\overline{T}$**	**Timer 1 Counter/Timer Select** 0 = Timer 1 is incremented by internal system clock. 1 = Timer 1 is incremented by the falling edge of the external pin T1.
[5]	**M1**	**Timer 1 Mode Select** M1 M0 Timer 1 Mode 0 0 Mode 0: 13-bit Timer/Counter 0 1 Mode 1: 16-bit Timer/Counter 1 0 Mode 2: 8-bit Timer/Counter with auto-reload from TH1 1 1 Mode 3: Timer 1 halted
[4]	**M0**	Check with bit 5 description.
[3]	**GATE**	**Timer 0 Gate Control** 0 = Timer 0 will clock when TR0 is 1 regardless of INT0 logic level. 1 = Timer 0 will clock only when TR0 is 1 and INT0 is logic 1.
[2]	**C/$\overline{T}$**	**Timer 0 Counter/Timer Select** 0 = Timer 0 is incremented by internal system clock. 1 = Timer 0 is incremented by the falling edge of the external pin T0.
[1]	**M1**	**Timer 0 Mode Select** M1 M0 Timer 0 Mode 0 0 Mode 0: 13-bit Timer/Counter 0 1 Mode 1: 16-bit Timer/Counter 1 0 Mode 2: 8-bit Timer/Counter with auto-reload from TH0 1 1 Mode 3: TL0 as a 8-bit Timer/Counter and TH0 as a 8-bit Timer
[0]	**M0**	Check with bit 1 description

TCON – Timer 0 and 1 Control (Bit-addressable)

Register	SFR Address	Reset Value
TCON	88H, All pages, Bit-addressable	0000_0000b

7	6	5	4	3	2	1	0
TF1	**TR1**	**TF0**	**TR0**	**IE1**	**IT1**	**IE0**	**IT0**
R/W	R/W	R/W	R/W	R (level) R/W (edge)	R/W	R (level) R/W (edge)	R/W

Bit	Name	Description
[7]	**TF1**	**Timer 1 Overflow Flag** This bit is set when Timer 1 overflows. It is automatically cleared by hardware when the program executes the Timer 1 interrupt service routine. This bit can be set or cleared by software.
[6]	**TR1**	**Timer 1 Run Control** 0 = Timer 1 Disabled. Clearing this bit will halt Timer 1 and the current count will be preserved in TH1 and TL1. 1 = Timer 1 Enabled.
[5]	**TF0**	**Timer 0 Overflow Flag** This bit is set when Timer 0 overflows. It is automatically cleared via hardware when the program executes the Timer 0 interrupt service routine. This bit can be set or cleared by software.
[4]	**TR0**	**Timer 0 Run Control** 0 = Timer 0 Disabled. Clearing this bit will halt Timer 0 and the current count will be preserved in TH0 and TL0. 1 = Timer 0 Enabled.
[3]	**IE1**	**External Interrupt 1 Edge Flag** If IT1 = 1 (falling edge trigger), this flag will be set by hardware when a falling edge is detected. It remain set until cleared via software or cleared by hardware in the beginning of its interrupt service routine. If IT1 = 0 (low level trigger), this flag follows the inverse of the INT1 input signal's logic level. Software cannot control it.
[2]	**IT1**	**External Interrupt 1 Type Select** This bit selects by which type that INT1 is triggered. 0 = INT1 is low level triggered. 1 = INT1 is fallinVg edge triggered.
[1]	**IE0**	**External Interrupt 0 Edge Flag** If IT0 = 1 (falling edge trigger), this flag will be set by hardware when a falling edge is detected. It remain set until cleared via software or cleared by hardware in the beginning of its interrupt service routine. If IT0 = 0 (low level trigger), this flag follows the inverse of the INT0 input signal's logic level. Software cannot control it.

Bit	Name	Description
[0]	**IT0**	**External Interrupt 0 Type Select** This bit selects by which type that INT0 is triggered. 0 = INT0 is low level triggered. 1 = INT0 is falling edge triggered.

TL0 – Timer 0 Low Byte

Register	SFR Address	Reset Value
TL0	8AH, Page 0	0000_0000b

7	6	5	4	3	2	1	0
TL0[7:0]							
R/W							

Bit	Name	Description
[7:0]	**TL0[7:0]**	**Timer 0 Low Byte** The TL0 register is the low byte of the 16-bit counting register of Timer 0.

TH0 – Timer 0 High Byte

Register	SFR Address	Reset Value
TH0	8CH, Page 0	0000_0000b

7	6	5	4	3	2	1	0
TH0[7:0]							
R/W							

Bit	Name	Description
[7:0]	**TH0[7:0]**	**Timer 0 High Byte** The TH0 register is the high byte of the 16-bit counting register of Timer 0.

TL1 – Timer 1 Low Byte

Register	SFR Address	Reset Value
TL1	8BH, Page 0	0000_0000b

7	6	5	4	3	2	1	0
TL1[7:0]							
R/W							

Bit	Name	Description
[7:0]	**TL1[7:0]**	**Timer 1 Low Byte** The TL1 register is the low byte of the 16-bit counting register of Timer 1.

TH1 – Timer 1 High Byte

Register	SFR Address	Reset Value
TH1	8DH, Page 0	0000_0000b

7	6	5	4	3	2	1	0
TH1[7:0]							
R/W							

Bit	Name	Description
[7:0]	**TH1[7:0]**	**Timer 1 High Byte** The TH1 register is the high byte of the 16-bit counting register of Timer 1.

CKCON – Clock Control

Register	SFR Address	Reset Value
CKCON	8EH, Page 0	1000_0000b

7	6	5	4	3	2	1	0
FASTWK	**PWMCKS**	**T1OE**	**T1M**	**T0M**	**T0OE**	**CLOEN**	**-**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-

Bit	Name	Description
[5]	**T1OE**	**Timer 1 Output Enable** 0 = Timer 1 output Disabled. 1 = Timer 1 output Enabled from T1 pin. Note that Timer 1 output should be enabled only when operating in its “Timer” mode.
[4]	**T1M**	**Timer 1 Clock Mode Select** 0 = The clock source of Timer 1 is the system clock divided by 12. It maintains standard 8051 compatibility. 1 = The clock source of Timer 1 is direct the system clock.
[3]	**T0M**	**Timer 0 Clock Mode Select** 0 = The clock source of Timer 0 is the system clock divided by 12. It maintains standard 8051 compatibility. 1 = The clock source of Timer 0 is direct the system clock.
[2]	**T0OE**	**Timer 0 Output Enable** 0 = Timer 0 output Disabled. 1 = Timer 0 output Enabled from T0 pin. Note that Timer 0 output should be enabled only when operating in its “Timer” mode.

6.6.3 Timer 2 and Input Capture

Timer 2 is a 16-bit up counter cascaded with TH2, the upper 8 bits register, and TL2, the lower 8 bit register. Equipped with RCMP2H and RCMP2L, Timer 2 can operate under compare mode and auto-reload mode selected by CM/$\overline{RL2}$ (T2CON.0). An 3-channel input capture module makes Timer 2 detect and measure the width or period of input pulses. The results of 3 input captures are stores in C0H and C0L, C1H and C1L, C2H and C2L individually. The clock source of Timer 2 is from the system clock pre-scaled by a clock divider with 8 different scales for wide field application. The clock is enabled when TR2 (T2CON.2) is 1, and disabled when TR2 is 0. The following registers are related to Timer 2 function.

6.6.3.1 Block Diagram

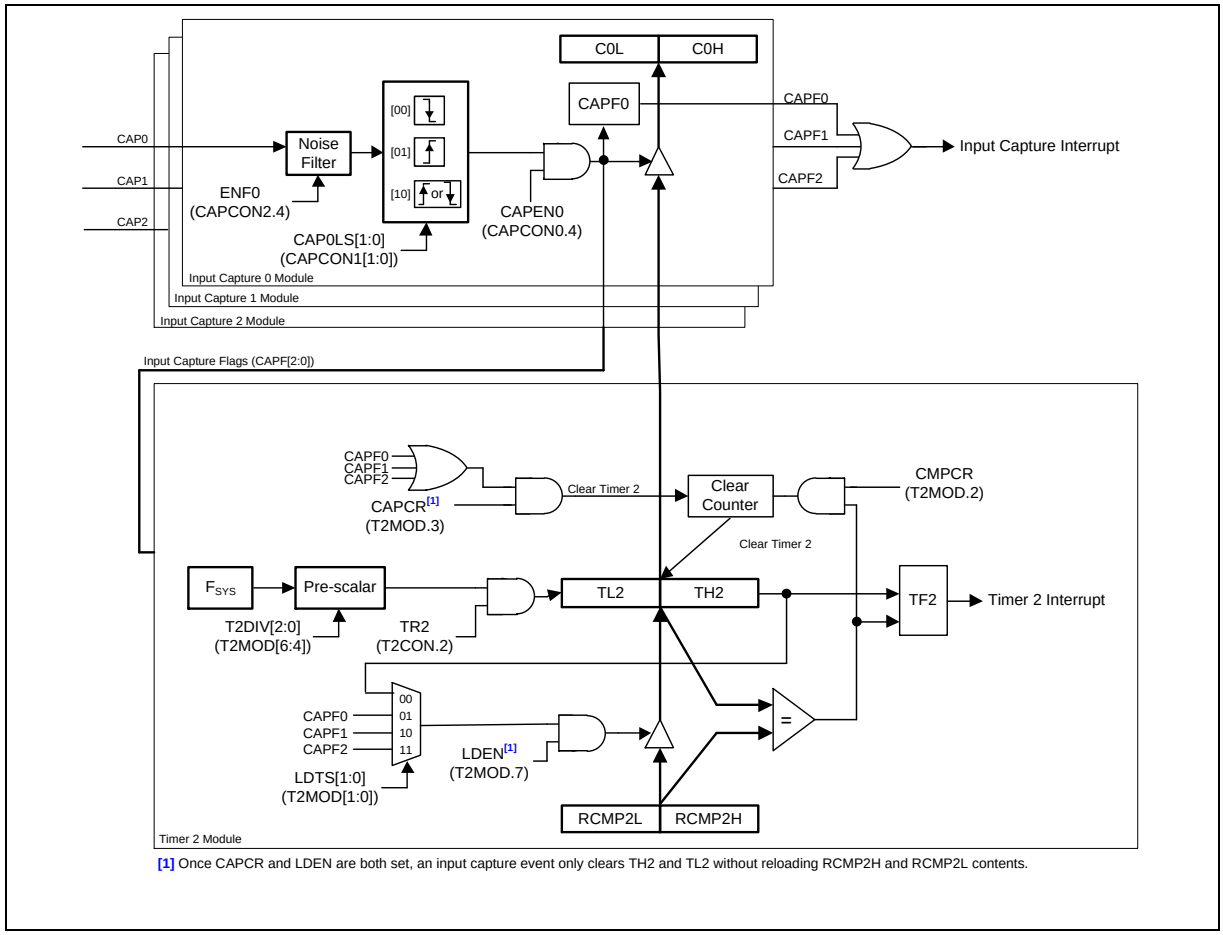


Figure 6.6-5 Timer 2 Block Diagram

Note that the TH2 and TL2 are accessed separately. It is strongly recommended that user stops Timer 2 temporally by clearing TR2 bit before reading from or writing to TH2 and TL2. The free-running reading or writing may cause unpredictable result.

6.6.3.2 Auto-Reload Mode

The Timer 2 is configured as auto-reload mode by clearing CM/$\overline{RL2}$. In this mode RCMP2H and RCMP2L registers store the reload value. The contents in RCMP2H and RCMP2L transfer into TH2 and TL2 once the auto-reload event occurs if setting LDEN bit. The event can be the Timer 2 overflow or one of the triggering event on any of enabled input capture channel depending on the LDTS[1:0] (T2MOD[1:0]) selection. Note that once CAPCR (T2MOD.3) is set, an input capture event only clears TH2 and TL2 without reloading RCMP2H and RCMP2L contents.

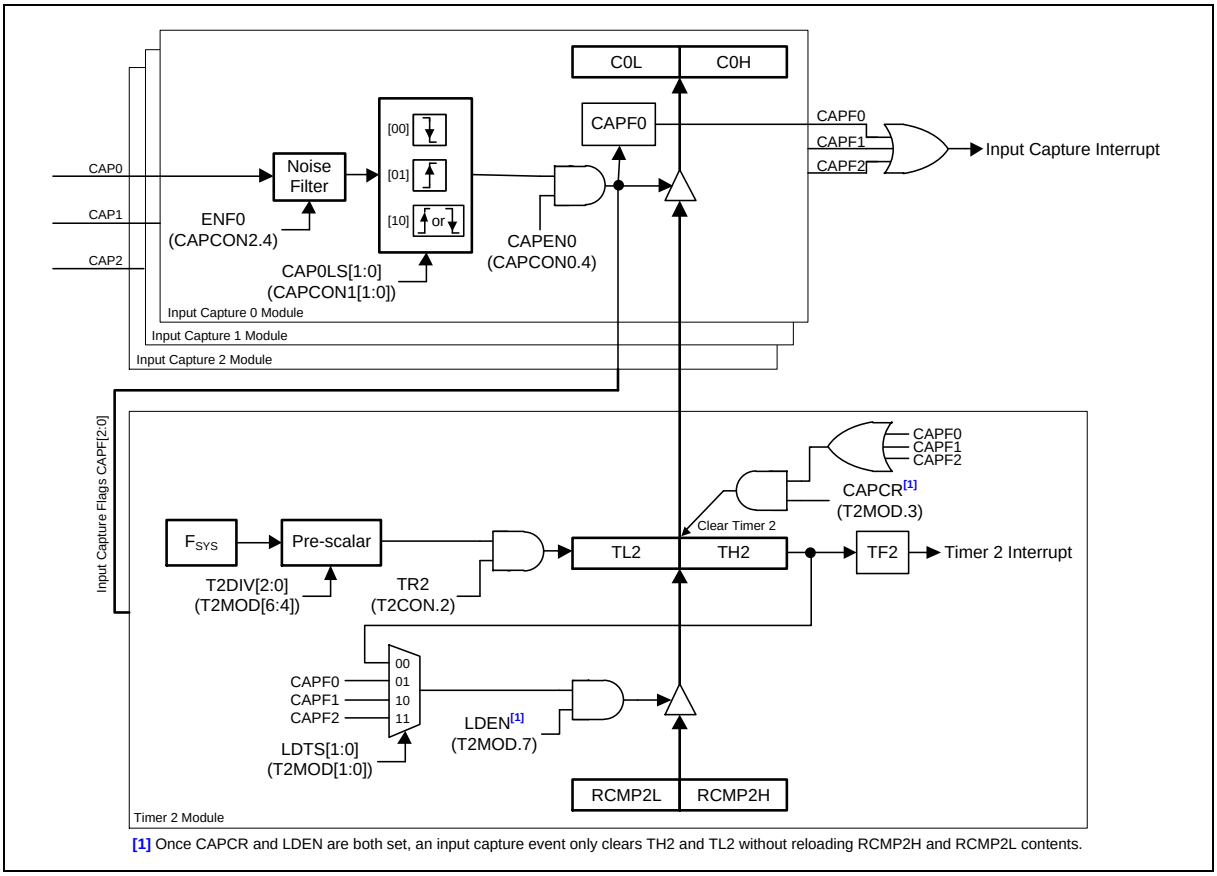


Figure 6.6-6 Timer 2 Auto-Reload Mode and Input Capture Module Functional Block Diagram

6.6.3.3 Compare Mode

Timer 2 can also be configured as the compare mode by setting CM/$\overline{RL2}$. In this mode RCMP2H and RCMP2L registers serve as the compare value registers. As Timer 2 up counting, TH2 and TL2 match RCMP2H and RCMP2L, TF2 (T2CON.7) will be set by hardware to indicate a compare match event.

Setting CMPCR (T2MOD.2) makes the hardware to clear Timer 2 counter as 0000H automatically after a compare match has occurred.

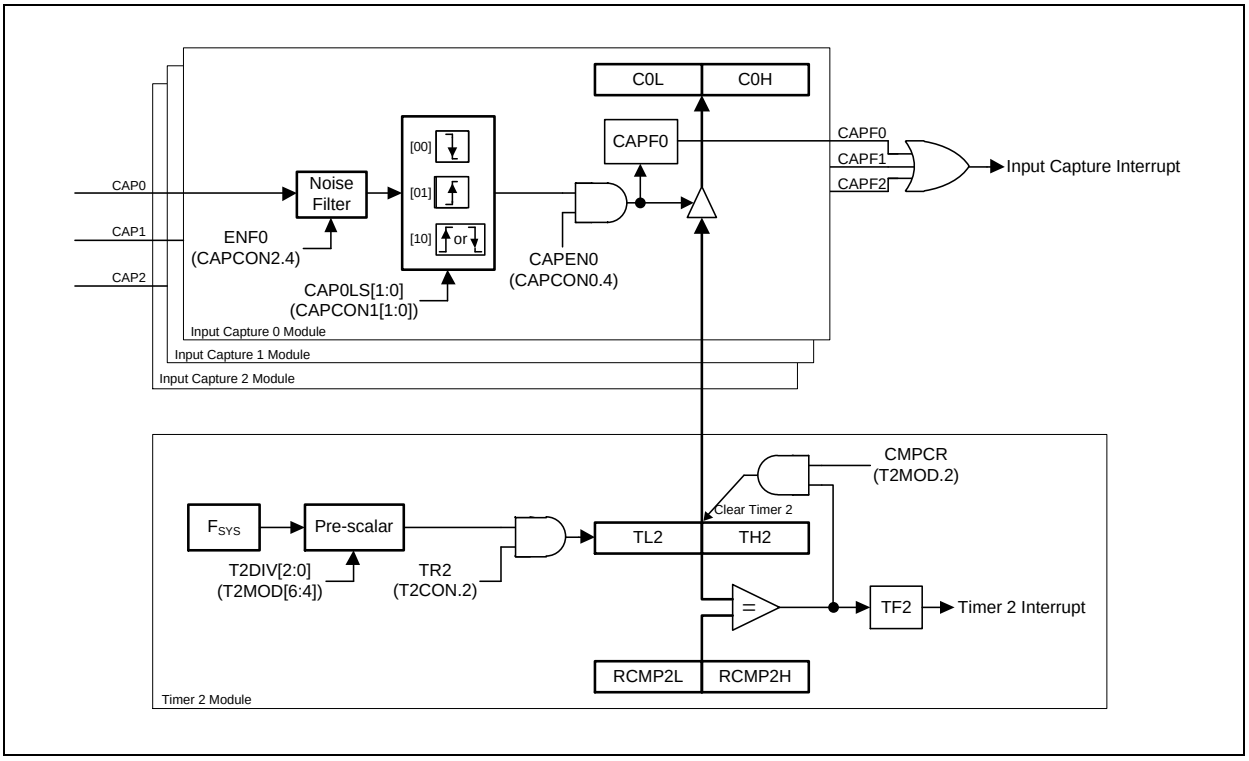


Figure 6.6-7 Timer 2 Compare Mode and Input Capture Module Functional Block Diagram

6.6.3.4 Input Capture Module

The input capture module along with Timer 2 implements the input capture function. The input capture module is configured through CAPCON0~2 registers. The input capture module supports 3-channel inputs (CAP0, CAP1, and CAP2). Each input channel consists its own noise filter, which is enabled via setting ENF0~2 (CAPCON2[6:4]). It filters input glitches smaller than four system clock cycles. Input capture channels has their own independent edge detector but share the unique Timer 2. Each trigger edge detector is selected individually by setting corresponding bits in CAPCON1. It supports positive edge capture, negative edge capture, or any edge capture. Each input capture channel has to set its own enabling bit CAPEN0~2 (CAPCON0[6:4]) before use.

While input capture channel is enabled and the selected edge trigger occurs, the content of the free running Timer 2 counter, TH2 and TL2, will be captured, transferred, and stored into the capture registers CnH and CnL. The edge triggering also causes CAPFn (CAPCON0.n) set by hardware. The interrupt will also generate if the ECAP (EIE0.2) and EA bit are both set. For three input capture flags share the same interrupt vector, user should check CAPFn to confirm which channel comes the input capture edge. These flags should be cleared by software.

The bit CAPCR (CAPCON2.3) benefits the implement of period calculation. Setting CAPCR makes the hardware clear Timer 2 as 0000H automatically after the value of TH2 and TL2 have been captured after an input capture edge event occurs. It eliminates the routine software overhead of writing 16-bit counter or an arithmetic subtraction.

6.6.3.5 Register Description

CAPCON0 – Input Capture Control 0

Register	SFR Address	Reset Value
CAPCON0	E1H, Page 1	0000_0000b

7	6	5	4	3	2	1	0
-	CAPEN2	CAPEN1	CAPEN0	-	CAPF2	CAPF1	CAPF0
-	R/W	R/W	R/W	-	R/W	R/W	R/W

Bit	Name	Description
[7]	-	Reserved
[6]	CAPEN2	**Input Capture 2 Enable** 0 = Input capture channel 2 Disabled. 1 = Input capture channel 2 Enabled.
[5]	CAPEN1	**Input Capture 1 Enable** 0 = Input capture channel 1 Disabled. 1 = Input capture channel 1 Enabled.
[4]	CAPEN0	**Input Capture 0 Enable** 0 = Input capture channel 0 Disabled. 1 = Input capture channel 0 Enabled.
[3]	-	Reserved
[2]	CAPF2	**Input Capture 2 Flag** This bit is set by hardware if the determined edge of input capture 2 occurs. This bit should cleared by software.
[1]	CAPF1	**Input Capture 1 Flag** This bit is set by hardware if the determined edge of input capture 1 occurs. This bit should cleared by software.
[0]	CAPF0	**Input Capture 0 Flag** This bit is set by hardware if the determined edge of input capture 0 occurs. This bit should cleared by software.

CAPCON1 – Input Capture Control 1

Register	SFR Address	Reset Value
CAPCON1	E2H, Page 1	0000_0000b

7	6	5	4	3	2	1	0
-	-	CAP2LS[1:0]		CAP1LS[1:0]		CAP0LS[1:0]	
-	-	R/W		R/W		R/W	

Address: E2H, Page 1 Reset value: 0000 0000b

Bit	Name	Description
[7:6]	-	Reserved
[5:4]	CAP2LS[1:0]	**Input Capture 2 Level Select** 00 = Falling edge. 01 = Rising edge. 10 = Either rising or falling edge. 11 = Reserved.
[3:2]	CAP1LS[1:0]	**Input Capture 1 Level Select** 00 = Falling edge. 01 = Rising edge. 10 = Either rising or falling edge. 11 = Reserved.
[1:0]	CAP0LS[1:0]	**Input Capture 0 Level Select** 00 = Falling edge. 01 = Rising edge. 10 = Either rising or falling edge. 11 = Reserved.

CAPCON2 – Input Capture Control 2

Register	SFR Address	Reset Value
CAPCON2	E3H, Page 1	0000_0000b

7	6	5	4	3	2	1	0
-	**ENF2**	**ENF1**	**ENF0**	**-**	**-**	**-**	**-**
-	R/W	R/W	R/W	-	-	-	-

Bit	Name	Description
[6]	**ENF2**	**Enable Noise Filer on Input Capture 2** 0 = Noise filter on input capture channel 2 Disabled. 1 = Noise filter on input capture channel 2 Enabled.
[5]	**ENF1**	**Enable Noise Filer on Input Capture 1** 0 = Noise filter on input capture channel 1 Disabled. 1 = Noise filter on input capture channel 1 Enabled.
[4]	**ENF0**	**Enable Noise Filer on Input Capture 0** 0 = Noise filter on input capture channel 0 Disabled. 1 = Noise filter on input capture channel 0 Enabled.

CnL – Capture Low Byte, n = 0,1,2

Register	SFR Address	Reset Value
C0L	E4H, Page 1	0000_0000 b
C1L	E6H, Page 1	0000_0000 b
C2L	EDH, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
CnL[7:0]							
R/W							

Bit	Name	Description
[7:0]	**CnL[7:0]**	**Input Capture n Result Low Byte** The CnL register is the low byte of the 16-bit result captured by input capture n.

CnH – Capture n High Byte, n = 1,2,3

Register	SFR Address	Reset Value
C0H	E5H, Page 1	0000_0000 b
C1H	E7H, Page 1	0000_0000 b
C2H	EEH, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
CnH[7:0]							
R/W							

Bit	Name	Description
[7:0]	**CnH[7:0]**	**Input Capture n Result High Byte** The CnH register is the high byte of the 16-bit result captured by input capture n.

6.6.4 Timer 3

Timer 3 is implemented simply as a 16-bit auto-reload, up-counting timer. The user can select the pre-scale with T3PS[2:0] (T3CON[2:0]) and fill the reload value into RH3 and RL3 registers to determine its overflow rate. User then can set TR3 (T3CON.3) to start counting. When the counter rolls over FFFFH, TF3 (T3CON.4) is set as 1 and a reload is generated and causes the contents of the RH3 and RL3 registers to be reloaded into the internal 16-bit counter. If ET3 (EIE1.1) is set as 1, Timer 3 interrupt service routine will be served. TF3 is auto-cleared by hardware after entering its interrupt service routine.

Timer 3 can also be the baud rate clock source of both UARTs. For details, please see Section 6.10.3.2“Baud Rate”.

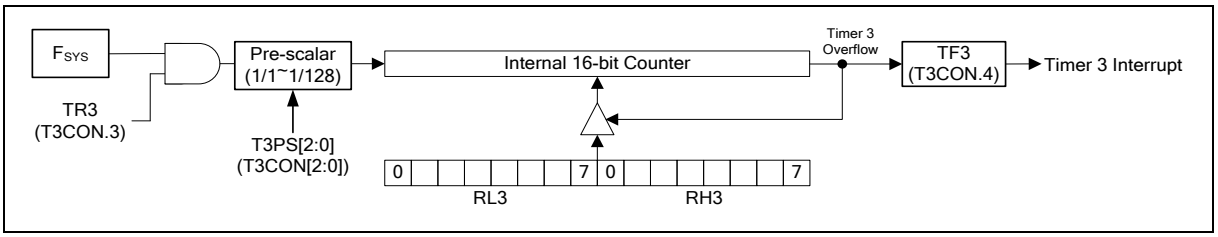


Figure 6.6-8 Timer 3 Block Diagram

Following shows the Timer 3 control register.

T3CON – Timer 3 Control

Register	SFR Address	Reset Value
T3CON	C4H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
SMOD_1	SMOD0_1	BRCK	TF3	TR3	T3PS[2:0]		
R/W	R/W	R/W	R/W	R/W	R/W		

Bit	Name	Description
[4]	**TF3**	**Timer 3 Overflow Flag** This bit is set when Timer 3 overflows. It is automatically cleared by hardware when the program executes the Timer 3 interrupt service routine. This bit can be set or cleared by software.
[3]	**TR3**	**Timer 3 Run Control** 0 = Timer 3 is halted. 1 = Timer 3 starts running. Note that the reload registers RH3 and RL3 can only be written when Timer 3 is halted (TR3 bit is 0). If any of RH3 or RL3 is written if TR3 is 1, result is unpredictable.
[2:0]	**T3PS[2:0]**	**Timer 3 Pre-Scalar** These bits determine the scale of the clock divider for Timer 3. 000 = 1/1. 001 = 1/2. 010 = 1/4. 011 = 1/8. 100 = 1/16. 101 = 1/32. 110 = 1/64. 111 = 1/128.

RL3 – Timer 3 Reload Low Byte

Register	SFR Address	Reset Value
RL3	C5H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
RL3[7:0]							
R/W							

Bit	Name	Description
[7:0]	**RL3[7:0]**	**Timer 3 Reload Low Byte** It holds the low byte of the reload value of Timer 3.

RH3 – Timer 3 Reload High Byte

Register	SFR Address	Reset Value
RH3	C6H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
RH3[7:0]							
R/W							

Bit	Name	Description
[7:0]	**RH3[7:0]**	**Timer 3 Reload High Byte** It holds the high byte of the reload value of Time 3.

6.7 Watchdog Timer (WDT)

The ML51/ML54/ML56 Series provides one Watchdog Timer (WDT). It can be configured as a time-out reset timer to reset whole device. Once the device runs in an abnormal status or hangs up by outward interference, a WDT reset recover the system. It provides a system monitor, which improves the reliability of the system. Therefore, WDT is especially useful for system that is susceptible to noise, power glitches, or electrostatic discharge. The WDT also can be configured as a general purpose timer, of which the periodic interrupt serves as an event timer or a durational system supervisor in a monitoring system, which is able to operate during Idle or Power-down mode. WDTEN[3:0] (CONFIG4[7:4]) initialize the WDT to operate as a time-out reset timer or a general purpose timer.

The Watchdog time-out interval is determined by the formula $\frac{1}{F_{LIRC} \times \text{clock divider scalar}} \times 64$, where F_{LIRC} is the frequency of internal 38.4 R/W oscillator. The following table shows an example of the Watchdog time-out interval with different pre-scales.

WDPS.3	WDPS.2	WDPS.1	WDPS.0	Clock Divider Scale	WDT Time-Out Timing[1]
0	0	0	0	1/1	1.66 ms
0	0	0	1	1/4	6.64 ms
0	0	1	0	1/8	13.31 ms
0	0	1	1	1/16	26.62 ms
0	1	0	0	1/32	53.25 ms
0	1	0	1	1/64	106.66 ms
0	1	1	0	1/128	213.12 ms
0	1	1	1	1/256	426.64 ms
1	0	0	0	1/512	853.28ms
1	0	0	1	1/1024	1706.56ms
1	0	1	0	1/2048	3413.12ms
Others				1/2048	3413.12ms
Note: This is an approximate value since the deviation of LIRC.					

Table 6.7-1 Watchdog Timer-out Interval Under Different Pre-scalars

Since the limitation of the maxima vaule of WDT timer delay. To wake up ML51/ML54/ML56 Series from idle mode or Power-down mode suggest use WKT function see Chapter 6.8 **.**

The WDT is implemented with a set of divider that divides the low-speed internal oscillator clock nominal 38.4 R/W. The divider output is selectable and determines the time-out interval. When the time-out interval is fulfilled, it will wake the system up from Idle or Power-down mode and an interrupt event will occur if WDT interrupt is enabled. If WDT is initialized as a time-out reset timer, a system reset will occur after a period of delay if without any software action.

6.7.1 Time-Out Reset Timer

When the CONFIG bits WDTEN[3:0] (CONFIG4[7:4]) is not FH, the WDT is initialized as a time-out reset timer. If WDTEN[3:0] is not 5H, the WDT is allowed to continue running after the system enters Idle or Power-down mode. Note that when WDT is initialized as a time-out reset timer, WDTR and WIDPD has no function.

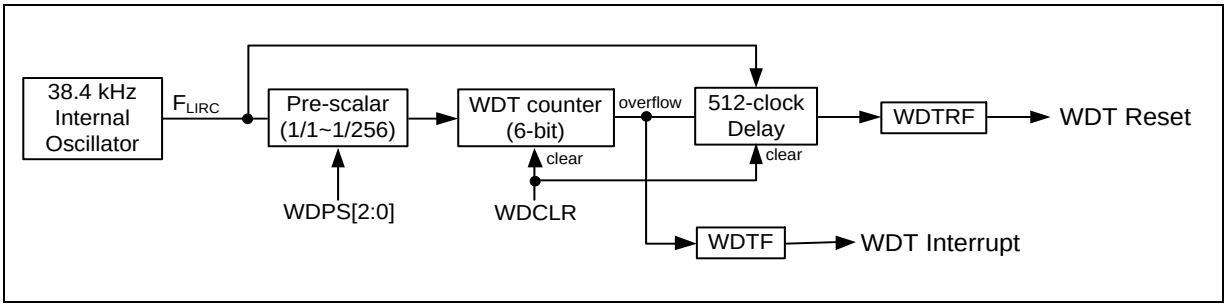


Figure 6.7-1 WDT as A Time-Out Reset Timer

After the device is powered and it starts to execute software code, the WDT starts counting simultaneously. The time-out interval is selected by the three bits WDPS[2:0] (WDCON[2:0]). When the selected time-out occurs, the WDT will set the interrupt flag WDTF (WDCON.5). If the WDT interrupt enable bit EWDT (EIE0.4) and global interrupt enable EA are both set, the WDT interrupt routine will be executed. Meanwhile, an additional 512 clocks of the low-speed internal oscillator delays to expect a counter clearing by setting WDCLR to avoid the system reset by WDT if the device operates normally. If no counter reset by writing 1 to WDCLR during this 512-clock period, a WDT reset will happen. Setting WDCLR bit is used to clear the counter of the WDT. This bit is self-cleared for user monitoring it. Once a reset due to WDT occurs, the WDT reset flag WDTRF (WDCON.3) will be set. This bit keeps unchanged after any reset other than a power-on reset. User may clear WDTRF via software. Note that all bits in WDCON require timed access writing.

The main application of the WDT with time-out reset enabling is for the system monitor. This is important in real-time control applications. In case of some power glitches or electro-magnetic interference, CPU may begin to execute erroneous codes and operate in an unpredictable state. If this is left unchecked the entire system may crash. Using the WDT during software development requires user to select proper “Feeding Dog” time by clearing the WDT counter. By inserting the instruction of setting WDCLR, it allows the code to run without any WDT reset. However If any erroneous code executes by any interference, the instructions to clear the WDT counter will not be executed at the required instants. Thus the WDT reset will occur to reset the system state from an erroneously executing condition and recover the system.

6.7.2 General Purpose Timer

There is another application of the WDT, which is used as a simple, long period timer. When the CONFIG bits WDTEN[3:0] (CONFIG4[7:4]) is FH, the WDT is initialized as a general purpose timer. In this mode, WDTR and WIDPD are fully accessed via software.

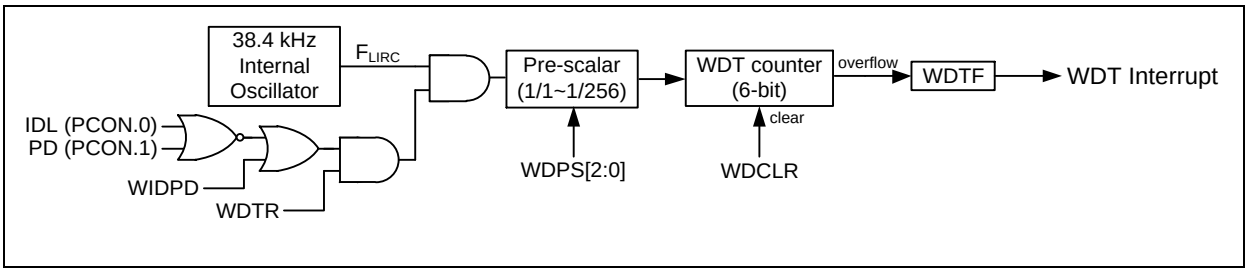


Figure 6.7-2 Watchdog Timer Block Diagram

The WDT starts running by setting WDTR as 1 and halts by clearing WDTR as 0. The WDTF flag will be set while the WDT completes the selected time interval. The software polls the WDTF flag to detect a time-out. An interrupt will occur if the individual interrupt EWDT (EIE0.4) and global interrupt enable EA is set. WDT will continue counting. User should clear WDTF and wait for the next overflow by polling WDTF flag or waiting for the interrupt occurrence.

6.7.3 Register Description

CONFIG4

7	6	5	4	3	2	1	0
WDTEN[3:0]				**-**	**-**	**-**	**-**
R/W				-	-	-	-

Factory default value: 1111 1111b

Bit	Name	Description
[7:4]	**WDTEN[3:0]**	**WDT Enable** This field configures the WDT behavior after MCU execution. 1111 = WDT is Disabled. WDT can be used as a general purpose timer via software control. 0101 = WDT is Enabled as a time-out reset timer and it stops running during Idle or Power-down mode. Others = WDT is Enabled as a time-out reset timer and it keeps running during Idle or Power-down mode.

WDCON – Watchdog Timer Control (TA Protected)

Register	SFR Address	Reset Value
WDCON	AAH, Page 0, TA protected	POR 0000_0111 b WDT 0000_1UUU b Others 0000_UUUU b

7	6	5	4	3	2	1	0
WDTR	**WDCLR**	**WDTF**	**WIDPD**	**WDTRF**	**WDPS[2:0]**		
R/W	R/W	R/W	R/W	R/W	R/W		

Bit	Name	Description
[7]	**WDTR**	**WDT Run** This bit is valid only when control bits in WDTEN[3:0] (CONFIG4[7:4]) are all 1. At this time, WDT works as a general purpose timer. 0 = WDT Disabled. 1 = WDT Enabled. The WDT counter starts running.
[6]	**WDCLR**	**WDT Clear** Setting this bit will reset the WDT count to 00H. It puts the counter in a known state and prohibit the system from unpredictable reset. The meaning of writing and reading WDCLR bit is different. Writing: 0 = No effect. 1 = Clearing WDT counter. Reading: 0 = WDT counter is completely cleared. 1 = WDT counter is not yet cleared.
[5]	**WDTF**	**WDT Time-Out Flag** This bit indicates an overflow of WDT counter. This flag should be cleared by software.
[4]	**WIDPD**	**WDT Running in Idle or Power-Down Mode** This bit is valid only when control bits in WDTEN[3:0] (CONFIG4[7:4]) are all 1. It decides whether WDT runs in Idle or Power-down mode when WDT works as a general purpose timer. 0 = WDT stops running during Idle or Power-down mode. 1 = WDT keeps running during Idle or Power-down mode.
[3]	**WDTRF**	**WDT Reset Flag** When the CPU is reset by WDT time-out event, this bit will be set via hardware. This flag is recommended to be cleared via software after reset.
[2:0]	**WDPS[2:0]**	**WDT Clock Pre-Scalar Select** These bits determine the pre-scale of WDT clock from 1/1 through 1/256. See Table 6.7-1 Watchdog Timer-out Interval Under Different Pre-scalars. The default is the maximum pre-scale value.

Bit	Name	Description
Note: 1. WDTRF will be cleared after power-on reset, be set after WDT reset, and remains unchanged after any other resets. 2. WDPS[3:0] are all set after power-on reset and keep unchanged after any reset other than power-on reset.		

6.7.4 Typical Structure of WDT Service Routine

In some application of low power consumption, the CPU usually stays in Idle mode when nothing needs to be served to save power consumption. After a while the CPU will be woken up to check if anything needs to be served at an interval of programmed period implemented by Timer 0~3. However, the current consumption of Idle mode still keeps at a “Ma” level. To further reducing the current consumption to “uA” level, the CPU should stay in Power-down mode when nothing needs to be served, and has the ability of waking up at a programmable interval. The ML51/ML54/ML56 Series is equipped with this useful function by WDT waking up. It provides a very low power internal oscillator 38.4 R/W as the clock source of the WDT. It is also able to count under Power-down mode and wake CPU up. The demo code to accomplish this feature is shown below.

```
     ORG   0000H
     LJMP  START

     ORG   0053H
     LJMP  WDT_ISR

     ORG   0100H
    ;*********************************************************************
    ;WDT interrupt service routine
    ;*********************************************************************
    WDT_ISR:
     CLR   EA
     MOV   TA,#0AAH
     MOV   TA,#55H
     ANL   WDCON,#11011111B   ;clear WDT interrupt flag
     SETB  EA
     RETI
    ;*********************************************************************
    ;Start here
    ;*********************************************************************
    START:
     MOV   TA,#0AAH
     MOV   TA,#55H
     ORL   WDCON,#00010111B   ;choose interval length and enable WDT
                              ;running during Power-down
     SETB  EWDT       ;enable WDT interrupt
     SETB  EA

     MOV   TA,#0AAH
     MOV   TA,#55H
     ORL   WDCON,#10000000B   ; WDT run
    ;*********************************************************************
    ;Enter Power-down mode
    ;*********************************************************************
```

```
LOOP:
 ORL   PCON,#02H
 LJMP  LOOP
```

6.8 Self Wake-up Timer (WKT)

6.8.1 Overview

The ML51/ML54/ML56 Series has a dedicated Self Wake-up Timer (WKT), which serves for a periodic wake-up timer in low power mode or for general purpose timer. WKT remains counting in Idle or Power-down mode. When WKT is being used as a wake-up timer, a start of WKT can occur just prior to entering a power management mode. WKT has two clock source, internal LIRC 38.4 R/W or LXT 32.768 R/W. Note that the system clock frequency must be twice over WKT clock. If WKT starts counting, the selected clock source will remain active once the device enters Idle or Power-down mode. Note that the selected clock source of WKT will not automatically enabled along with WKT configuration. User should manually enable the selected clock source and waiting for stability to ensure a proper operation.

The WKT is implemented simply as a 16-bit auto-reload, up-counting timer with pre-scale 1/1 to 1/2048 selected by WKPS[2:0] (WKCON[2:0]). User fills the reload value into RWK register to determine its overflow rate. The RWK can reloadable when counter is count to overflow. The CWK can read current count value. The WKTR (WKCON.3) can be set to start counting. When the counter rolls over FFH, WKTF (WKCON.4) is set as 1 and a reload is generated and causes the contents of the RWK register to be reloaded into the internal 8-bit counter. If EWKT (EIE1.2) is set as 1, WKT interrupt service routine will be served.

6.8.2 Block Diagram

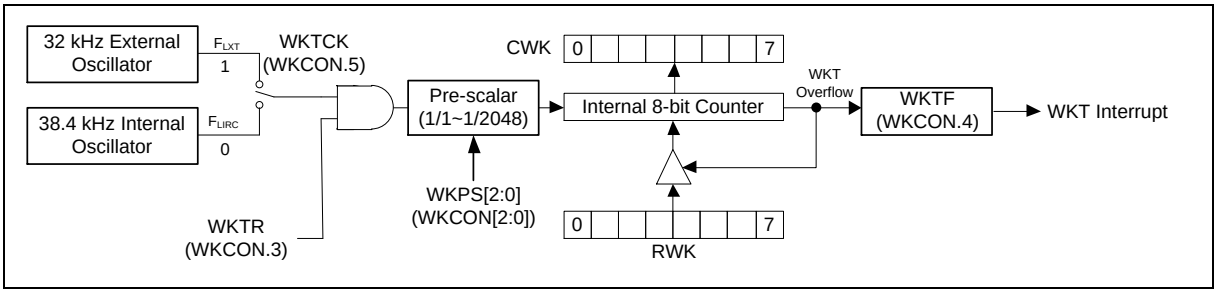


Figure 6.8-1 ML51 32/16 KB Flash Series Self Wake-Up Timer Block Diagram

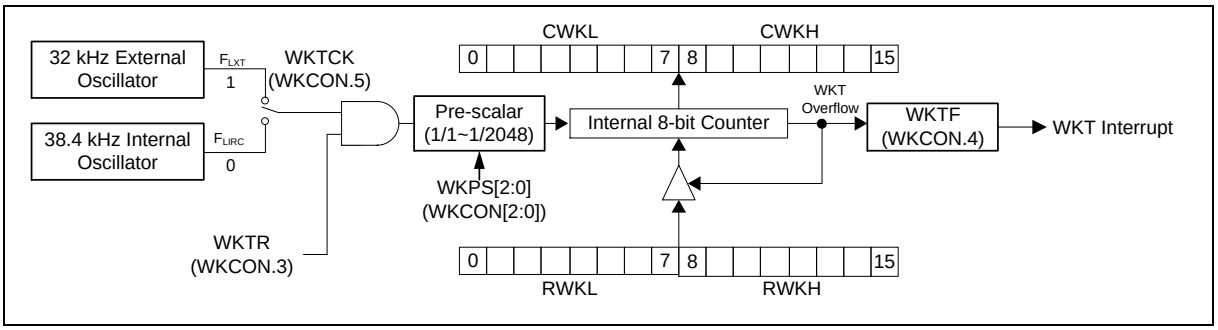


Figure 6.8-2 ML56/ML54/ML51 64 KB Flash Series Self Wake-Up Timer Block Diagram

6.8.3 Control Register

WKCON – Self Wake-up Timer Control

Register	SFR Address	Reset Value
WKCON	8FH, Page 0	0000_0000b

7	6	5	4	3	2	1	0
-	-	WKTCK	WKTF	WKTR	WKPS[2:0]		
-	-	R/W	R/W	R/W	R/W		

Bit	Name	Description
[7:6]	-	Reserved
[5]	WKTCK	**WKT Clock Source** This bit is set WKT clock source select bit. 0 = LIRC 1 = LXT
[4]	WKTF	**WKT Overflow Flag** This bit is set when WKT overflows. If the WKT interrupt and the global interrupt are enabled, setting this bit will make CPU execute WKT interrupt service routine. This bit is not automatically cleared via hardware and should be cleared via software.
[3]	WKTR	**WKT Run Control** 0 = WKT is halted. 1 = WKT starts running. Note that the reload register RWK can only be written when WKT is halted (WKTR bit is 0). If WKT is written while WKTR is 1, result is unpredictable.
[2:0]	WKPS[2:0]	**WKT Pre-Scalar** These bits determine the pre-scale of WKT clock. 000 = 1/1. 001 = 1/4. 010 = 1/16. 011 = 1/64. 100 = 1/256. 101 = 1/512. 110 = 1/1024. 111 = 1/2048.

RWKH – Self Wake-up Timer Reload High Byte

Note: This register only for ML56/ML54/ML51 64KB flash series product.

Register	SFR Address	Reset Value
RWKH	BFH, Page 2	0000 0000b

7	6	5	4	3	2	1	0
RWK[15:8]							
R/W							

Bit	Name	Description
[7:0]	RWK[15:8]	**WKT Reload High Byte** It holds the 16-bit reload value of WKT. Note that RWK should not be FFFFH if the pre-scale is 1/1 for implement limitation.

RWKL – Self Wake-up Timer Reload Low Byte

Register	SFR Address	Reset Value
RWKL	86H, Page 0	0000 0000b

7	6	5	4	3	2	1	0
RWK[7:0]							
R/W							

Bit	Name	Description
[7:0]	**RWK[7:0]**	**WKT Reload Low Byte** It holds the 16-bit reload value of WKT. Note that RWK should not be FFFFH if the pre-scale is 1/1 for implement limitation.

CWKH – Self Wake-up Timer Current Count Value High Byte

Note: This register only for ML56/ML54/ML51 64KB flash series product.

Register	SFR Address	Reset Value
CWKH	BEH, Page 2	0000 0000b

7	6	5	4	3	2	1	0
CWK[15:8]							
R/W							

Bit	Name	Description
[7:0]	**CWK[15:8]**	**WKT Current Count Value Low Byte High Byte** It is store value of WKT current count.

CWKL – Self Wake-up Timer Current Count Value Low Byte

Register	SFR Address	Reset Value
CWKL	86H, Page 1	0000 0000b

7	6	5	4	3	2	1	0
CWK[7:0]							
R							

Bit	Name	Description
[7:0]	CWK[7:0]	**WKT Current Count Value Low Byte Low Byte** It is store value of WKT current count.

6.9 Pulse Width Modulated (PWM)

6.9.1 Overview

The PWM (Pulse Width Modulation) signal is a useful control solution in wide application field. It can used on motor driving, fan control, backlight brightness tuning, LED light dimming, or simulating as a simple digital to analog converter output through a low pass filter circuit.

The ML51/ML54/ML56 Series PWM0 is especially designed for motor control by providing three pairs, maximum 16-bit resolution of PWM0 output with programmable period and duty. The architecture makes user easy to drive the one-phase or three-phase brushless DC motor (BLDC), or three-phase AC induction motor. Each of six PWM can be configured as one of independent mode, complementary mode, or synchronous mode. If the complementary mode is used, a programmable dead-time insertion is available to protect MOS turn-on simultaneously. The PWM waveform can be edge-aligned or center-aligned with variable interrupt points.

The ML51/ML54/ML56 Series PWM1/2/3 provide individual configurable period and duty. Maximum 16-bit resolution output. Each of two PWM1/2/3 can be configured as one of independent mode, complementary mode, or synchronous mode.The PWM1/2/3 waveform can be edge-aligned or center-aligned with variable interrupt points.

6.9.2 Features

- Up To 12 output pins can be selected
- Supports maximum clock source frequency up to 24 R/W
- Supports up to Three PWM modules, each module provides 6 output channels.
- Supports independent mode for PWM output
- Supports complementary mode for 3 complementary paired PWM output channels
- Dead-time insertion with 8-bit resolution
- Supports 16-bit resolution PWM counter
- Supports mask function and tri-state enable for each PWM pin
- Supports brake function
- Supports trigger ADC on the following events

6.9.3 Block Diagram

6.9.3.1 PWM0 Block Diagram

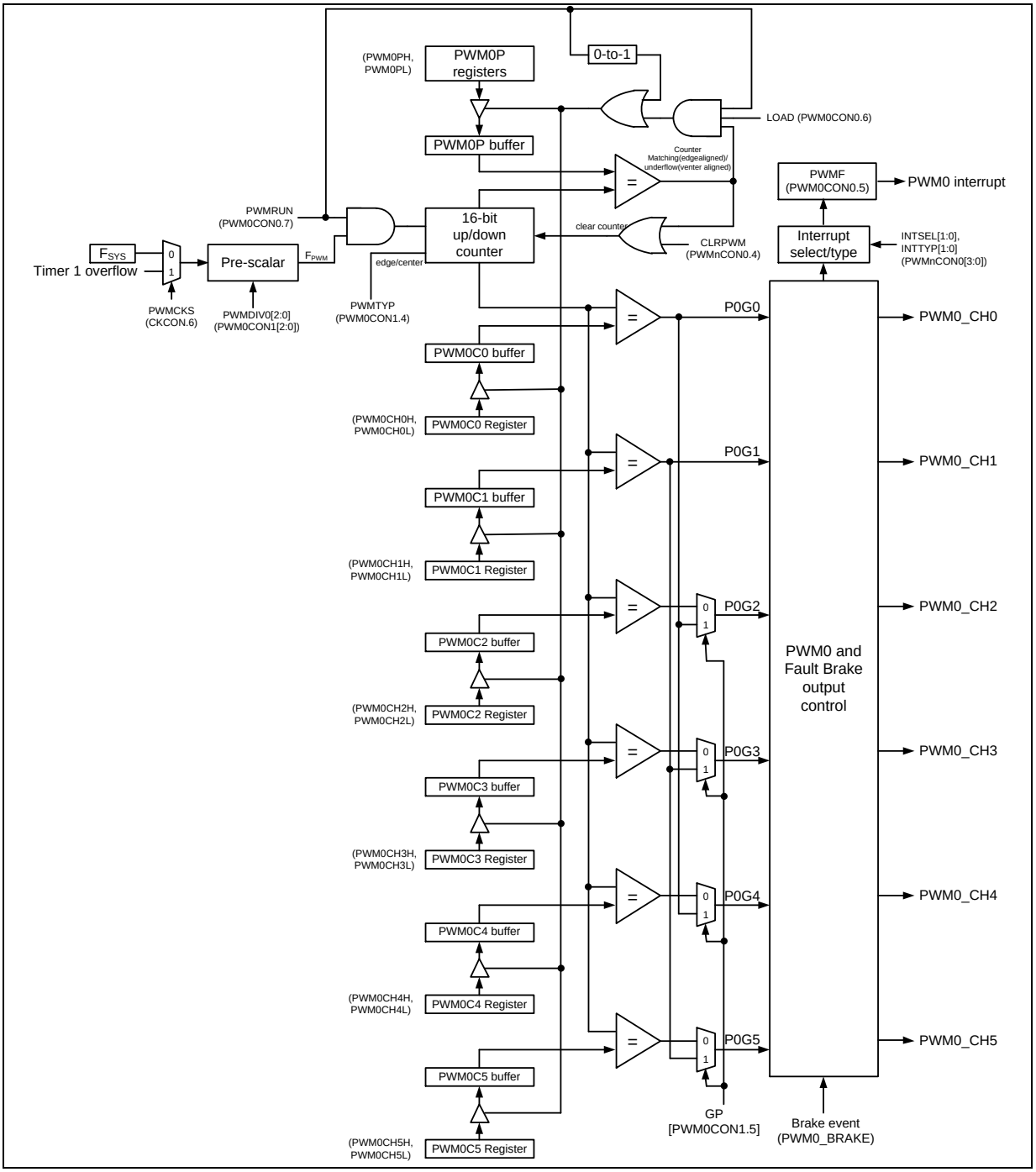


Figure 6.9-1 PWM0 Block Diagram

6.9.3.2 PWM1/ PWM2 / PWM3 Block Diagram

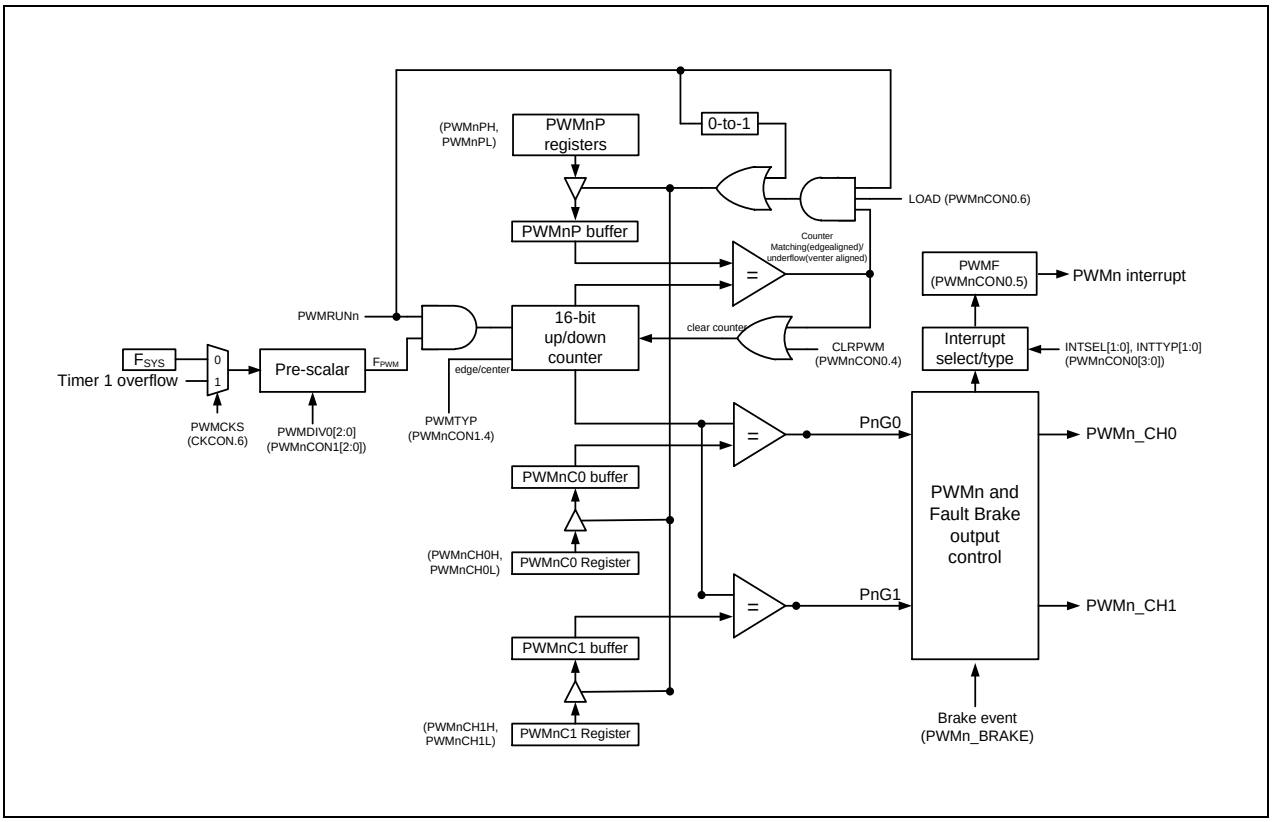


Figure 6.9-2 PWM1/ PWM2 / PWM3 Block Diagram

6.9.4 Functional Description

6.9.4.1 PWM Generator

The PWM generator is clocked by the system clock or Timer 1 overflow divided by a PWM clock pre-scalar selectable from 1/1~1/128. The PWM0/1/2/3 period is defined by effective 16-bit period registers, {PWMnPH, PWMnPL}. The period is the same for all PWM0/1/2/3 channels for they share the same 16-bit period counter. The duty of each PWM is determined independently by the value of duty registers. PWM0 has six duty registers. PWM1/2/3 has two duty registers. These PWMs output can be generated independently with different duty cycles. The interval and duty of PWM0/1/2/3 signal is generated by a 16-bit counter comparing with the period and duty registers.

Only PWM0 has group mode to facilitate the three-phase motor control, a group mode can be used by setting GP (PWM0CON1.5), which makes {PWM0C0H, PWM0C0L} and {PWM0C1H, PWM0C1L} duty register decide duties of the PWM outputs. In a three-phase motor control application, two-group PWM outputs generally are given the same duty cycle. When the group mode is enabled, {PWM0C2H, PWM0C2L}, {PWM0C3H, PWM0C3L}, {PWM0C4H, PWM0C4L} and {PWM0C5H, PWM0C5L} registers have no effect. This mean is {PWM0C2H, PWM0C2L} and {PWM0C4H, PWM0C4L} both as same as {PWM0C0H, PWM0C0L}. Also {PWM0C3H, PWM0C3L} and {PWM0C5H, PWM0C5L} are same as {PWM0C1H, PWM0C1L}.Note that enabling PWM0 does not configure the I/O pins into their output mode automatically. User should configure I/O output mode via software manually.

The PWM0 counter generates six PWM0 signals called P0G0, P0G1, P0G2, P0G3, P0G4, and P0G5. These signals will go through the PWM0 and Fault Brake output control circuit. It generates real PWM0 outputs on I/O pins. The output control circuit determines the PWM mode, dead-time insertion, mask output, Fault Brake control, and PWM polarity. The last stage is a multiplexer of PWM0 output or I/O function.

The PWM1/2/3 counter generates two PWM1/2/3 signals. These signals will go through the PWM1/2/3. It generates real PWM1/2/3 outputs on I/O pins. The output control circuit determines the PWM mode, mask output and PWM polarity. The last stage is a multiplexer of PWM1/2/3 output or I/O function.

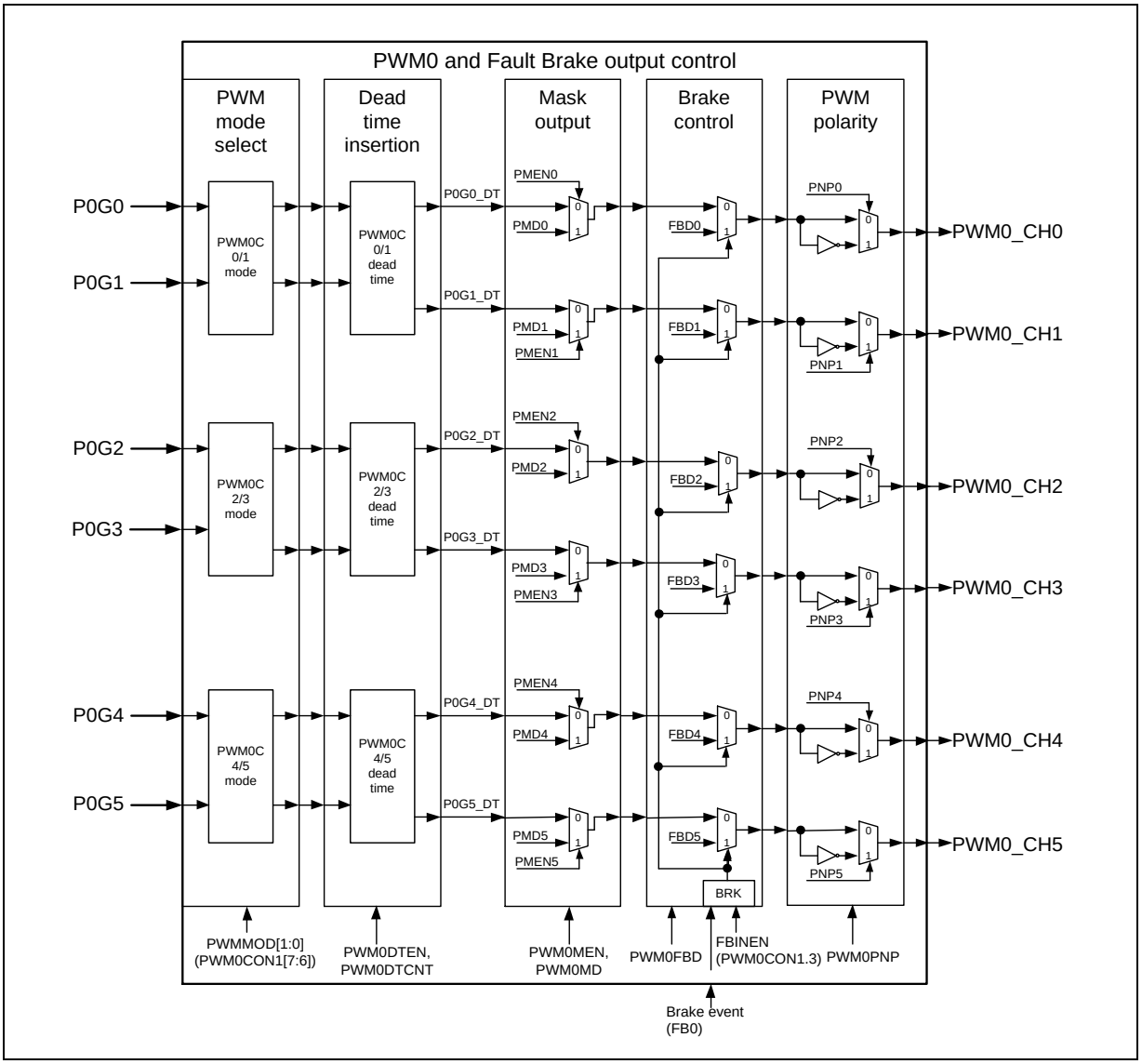


Figure 6.9-3 PWM0 and Fault Brake Output Control Block Diagram

User should follow the initialization steps below to start generating the PWM signal output. In the first step by setting CLRPWM (PWMnCON0.4), it ensures the 16-bit up counter reset for the accuracy of the first duration. After initialization and setting {PWMnPH, PWMnPL} and all {PWMnH, PWMnL} registers, PWMRUN (PWMnCON0.7) can be set as logic 1 to trigger the 16-bit counter running. PWM starts to generate waveform on its output pins. The hardware for all period and duty Register Description are double buffered designed. Therefore, {PWMnPH, PWMnPL} and all {PWMnH, PWMnL} registers can be written to at any time, but the period and duty cycle of PWM will not be updated immediately until the LOAD (PWMnCON0.6) is set and previous period is complete. This prevents glitches when updating the PWM period or duty.

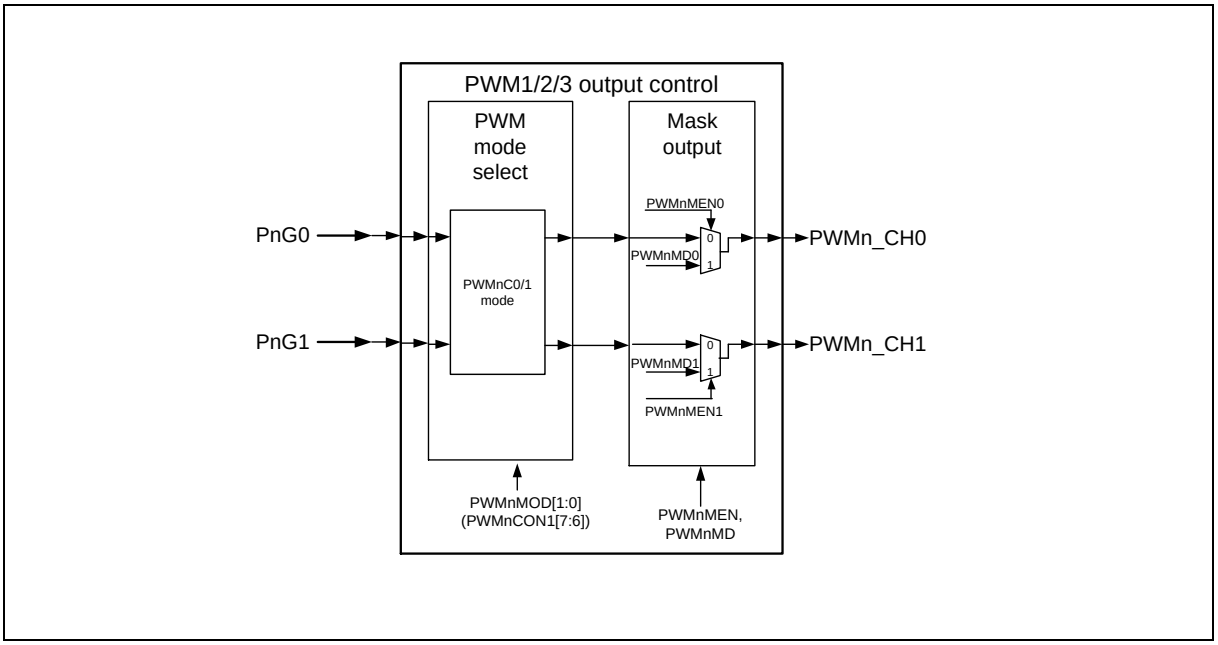


Figure 6.9-4 PWM1/2/3 Control Block Diagram

Note: *A loading of new period and duty by setting LOAD should be ensured complete by monitoring it and waiting for a hardware automatic clearing LOAD bit. Any updating of PWM Register Description during LOAD bit as logic 1 will cause unpredictable output*

PWM0CON0 – PWM Control Register0

Register	SFR Address	Reset Value
PWM0CON0	D1H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
PWM0RUN	**LOAD**	**PWMF**	**CLRPWM**	**-**	**-**	**-**	**-**
R/W	R/W	R/W	R/W	-	-	-	-

Bit	Name	Description
[7]	**PWM0RUN**	**PWM0 Run Enable** 0 = PWM0 stays in idle. 1 = PWM0 starts running.
[6]	**LOAD**	**PWM New Period and Duty Load** This bit is used to load period and duty Register Description in their buffer if new period or duty value needs to be updated. The loading will act while a PWM period is completed. The new period and duty affected on the next PWM cycle. After the loading is complete, LOAD will be automatically cleared via hardware. The meaning of writing and reading LOAD bit is different. Writing: 0 = No effect. 1 = Load new period and duty in their buffers while a PWM period is completed. Reading: 0 = A loading of new period and duty is finished. 1 = A loading of new period and duty is not yet finished.
[5]	**PWMF**	**PWM Flag** This flag is set according to definitions of INTSEL[2:0] and INTTYP[1:0] in PWMnINTC. This bit is cleared by software.
[4]	**CLRPWM**	**Clear PWM Counter** Setting this bit clears the value of PWM 16-bit counter for resetting to 0000H. After the counter value is cleared, CLRPWM will be automatically cleared via hardware. The meaning of writing and reading CLRPWM bit is different. Writing: 0 = No effect. 1 = Clearing PWM 16-bit counter. Reading: 0 = PWM 16-bit counter is completely cleared. 1 = PWM 16-bit counter is not yet cleared.
[3:0]	**-**	Reserved

PWMnCON0 – PWM Control Register0

Register	SFR Address	Reset Value
PWM1CON0	9CH, Page 2	0000_0000 b
PWM2CON0	C4H, Page 2	0000_0000 b
PWM3CON0	D4H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PWMnRUN	**LOAD**	**PWMF**	**CLRPWM**	**-**	**-**	**-**	**-**
R/W	R/W	R/W	R/W	-	-	-	-

Bit	Name	Description
[7]	**PWMnRUN**	**PWMn Run Enable** 0 = PWM stays in idle. 1 = PWM starts running.
[6]	**LOAD**	**PWM New Period and Duty Load** This bit is used to load period and duty Register Description in their buffer if new period or duty value needs to be updated. The loading will act while a PWM period is completed. The new period and duty affected on the next PWM cycle. After the loading is complete, LOAD will be automatically cleared via hardware. The meaning of writing and reading LOAD bit is different. Writing: 0 = No effect. 1 = Load new period and duty in their buffers while a PWM period is completed. Reading: 0 = A loading of new period and duty is finished. 1 = A loading of new period and duty is not yet finished.
[5]	**PWMF**	**PWM Flag** This flag is set according to definitions of INTSEL[2:0] and INTTYP[1:0] in PWMnINTC. This bit is cleared by software.
[4]	**CLRPWM**	**Clear PWM Counter** Setting this bit clears the value of PWM 16-bit counter for resetting to 0000H. After the counter value is cleared, CLRPWM will be automatically cleared via hardware. The meaning of writing and reading CLRPWM bit is different. Writing: 0 = No effect. 1 = Clearing PWM 16-bit counter. Reading: 0 = PWM 16-bit counter is completely cleared. 1 = PWM 16-bit counter is not yet cleared.
[3:0]	**-**	Reserved

PWMnCON1 – PWM Control 1

Register	SFR Address	Reset Value
PWM0CON1	DFH, Page 1	0000_0000 b
PWM1CON1	9DH, Page 2	0000_0000 b
PWM2CON1	C5H, Page 2	0000_0000 b
PWM3CON1	D5H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PWMMOD[1:0]		GP	PWMTYP	FBINEN	PWMDIV[2:0]		
R/W		R/W	R/W	R/W	R/W		

Bit	Name	Description
[2:0]	**PWMDIV[2:0]**	**PWM Clock Divider** This field decides the pre-scale of PWM clock source. 000 = 1/1. 001 = 1/2 010 = 1/4. 011 = 1/8. 100 = 1/16. 101 = 1/32. 110 = 1/64. 111 = 1/128.

CKCON – Clock Control

Register	SFR Address	Reset Value
CKCON	8EH, Page 0	0000_0000b

7	6	5	4	3	2	1	0
FASTWK	**PWMCKS**	**T1OE**	**T1M**	**T0M**	**T0OE**	**CLOEN**	**-**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-

Bit	Name	Description
[6]	**PWMCKS**	**PWM Clock Source Select** 0 = The clock source of PWM is the system clock FSYS. 1 = The clock source of PWM is the overflow of Timer 1.

PWMnPL – PWM Period Low Byte

Register	SFR Address	Reset Value
PWM0PL	D9H, Page 1	0000_0000 b
PWM1PL	99H, Page 2	0000_0000 b
PWM2PL	C1H, Page 2	0000_0000 b
PWM3PL	D1H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PWMnP[7:0]							
R/W							

Bit	Name	Description
[7:0]	**PWMnP[7:0]**	**PWMn Period Low Byte** This byte with PWMnPH controls the period of the PWM generator signal.

PWMnPH – PWM Period High Byte

Register	SFR Address	Reset Value
PWM0PH	D1H, Page 1	0000_0000 b
PWM1PH	86H, Page 2	0000_0000 b
PWM2PH	B9H, Page 2	0000_0000 b
PWM3PH	C9H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PWMnP[15:8]							
R/W							

Bit	Name	Description
[7:0]	**PWMnP[15:8]**	**PWM Period High Byte** This byte with PWMnPL controls the period of the PWM generator signal.

PWMnCxH – PWM0/1/2/3 Channel 0~5 Duty High Byte n=0,1,2,3; x=0,1,2,3,4,5

Register	SFR Address	Reset Value
PWM0C0H	D2H, Page 1	0000_0000 b
PWM0C1H	D3H, Page 1	0000_0000 b
PWM0C2H	D4H, Page 1	0000_0000 b
PWM0C3H	D5H, Page 1	0000_0000 b
PWM0C4H	C4H, Page 1	0000_0000 b
PWM0C5H	C5H, Page 1	0000_0000 b
PWM1C0H	8AH, Page 2	0000_0000 b
PWM1C1H	8BH, Page 2	0000_0000 b
PWM2C0H	BAH, Page 2	0000_0000 b
PWM2C1H	BBH, Page 2	0000_0000 b
PWM3C0H	CAH, Page 2	0000_0000 b
PWM3C1H	CBH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PWMnCx [15:8], n=0,1,2,3; x=0,1,2,3,4,5							
R/W							

Bit	Name	Description
[7:0]	**PWMnCx [15:8], n=0,1,2,3; x=0,1,2,3,4,5**	**PWMnCx Duty High Byte** This byte with PWMnCxL controls the duty of the output signal PGx from PWM generator.

PWMnCxL – PWM0/1/2/3 Channel 0~5 Duty Low Byte n=0,1,2,3; x=0,1,2,3,4,5

Register	SFR Address	Reset Value
PWM0C0L	DAH, Page 1	0000_0000 b
PWM0C1L	DBH, Page 1	0000_0000 b
PWM0C2L	DCH, Page 1	0000_0000 b
PWM0C3L	DDH, Page 1	0000_0000 b
PWM0C4L	CCH, Page 1	0000_0000 b
PWM0C5L	CDH, Page 1	0000_0000 b
PWM1C0L	9AH, Page 2	0000_0000 b
PWM1C1L	9BH, Page 2	0000_0000 b
PWM2C0L	C2H, Page 2	0000_0000 b
PWM2C1L	C3H, Page 2	0000_0000 b
PWM3C0L	D2H, Page 2	0000_0000 b
PWM3C1L	D3H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PWMnCx [7:0], n=0,1,2,3; x=0,1,2,3,4,5							
R/W							

Bit	Name	Description
[7:0]	**PWMnCx [7:0], n=0,1,2,3; x=0,1,2,3,4,5**	**PWMnCx Duty Low Byte** This byte with PWMnCxH controls the duty of the output signal PGx from PWM generator.

6.9.4.2 PWM Types

The PWM generator provides two PWM types: edge-aligned or center-aligned. PWM type is selected by PWMTYP (PWMnCON1.4).

PWMnCON1 – PWM Control 1

Register	SFR Address	Reset Value
PWM0CON1	DFH, Page 1	0000_0000 b
PWM1CON1	9DH, Page 2	0000_0000 b
PWM2CON1	C5H, Page 2	0000_0000 b
PWM3CON1	D5H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PWMMOD[1:0]		GP	PWMTYP	FBINEN	PWMDIV[2:0]		
R/W		R/W	R/W	R/W	R/W		

Bit	Name	Description
[4]	**PWMTYP**	**PWM Type Select** 0 = Edge-aligned PWM. 1 = Center-aligned PWM.

6.9.4.3 Edge-Aligned Type

In edge-aligned mode, the 16-bit counter uses single slop operation by counting up from 0000H to {PWMnPH, PWMnPL} and then starting from 0000H. The PWM generator signal (PGn before PWM and Fault Brake output control) is cleared on the compare match of 16-bit counter and the duty register {PWMnH, PWMnL} and set at the 16-bit counter is 0000H. The result PWM output waveform is left-edge aligned.

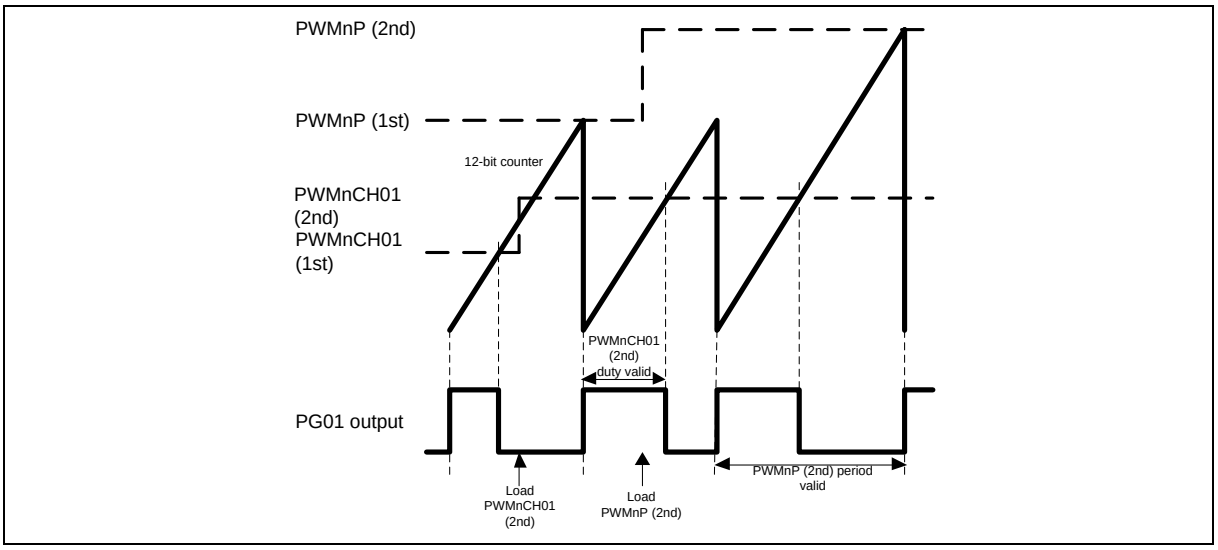


Figure 6.9-5 PWM Edge-aligned Type Waveform

The output frequency and duty cycle for edge-aligned PWM are given by following equations:

PWM frequency = $\frac{F_{PWM}}{\{PWMnPH, PWMnPL\}+1}$ (F_{PWM} is the PWM clock source frequency divided by PWMDIV).

PWM high level duty = $\frac{\{PWMnCHxH, PWMnCHxL\}}{\{PWMnPH, PWMnPL\}+1}$.

6.9.4.4 Center-Aligned Type

In center-aligned mode, the 16-bit counter use dual slop operation by counting up from 0000H to {PWMnPH, PWMnPL} and then counting down from {PWMnPH, PWMnPL} to 0000H. The PGn signal is cleared on the up-count compare match of 16-bit counter and the duty register {PWMnH, PWMnL} and set on the down-count compare match. Center-aligned PWM may be used to generate non-overlapping waveforms.

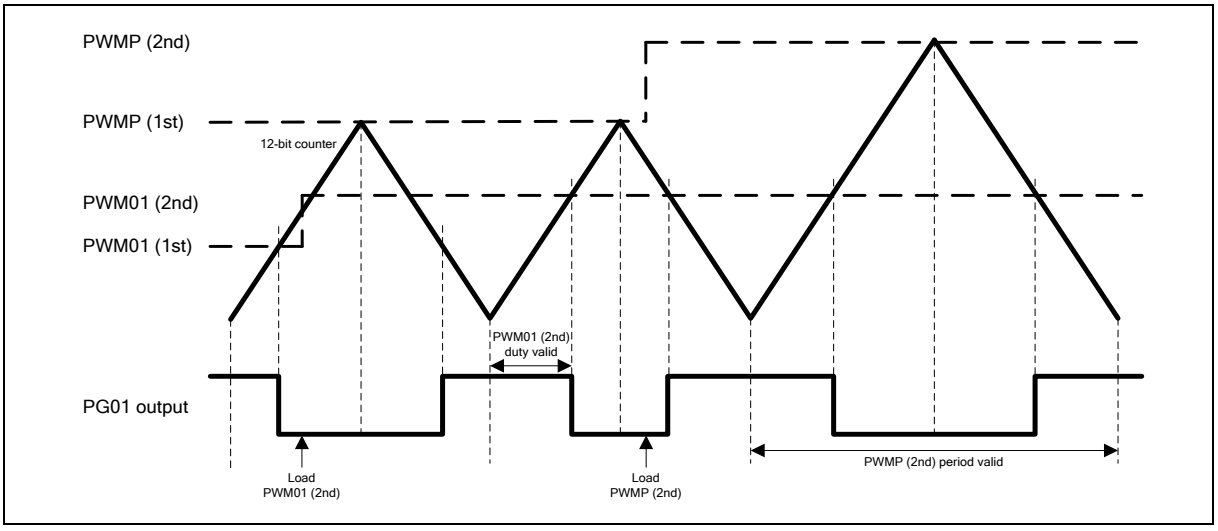


Figure 6.9-6 PWM Center-aligned Type Waveform

The output frequency and duty cycle for center-aligned PWM are given by following equations:

PWM frequency = $\frac{F_{PWM}}{2 \times \{PWMnPH, PWMnPL\}}$ (F_{PWM} is the PWM clock source frequency divided by PWMDIV).

PWM high level duty = $\frac{\{PWMnCHxH, PWMnCHxL\}}{\{PWMnPH, PWMnPL\}}$.

6.9.4.5 Operation Modes

After PGn signals pass through the first stage of the PWM and Fault Brake output control circuit. The PWM mode selection circuit generates different kind of PWM output modes with six-channel, three-pair signal PG0~PG5 . It supports independent mode, complementary mode, and synchronous mode.

PWMnCON1 – PWM Control 1

Register	SFR Address	Reset Value
PWM0CON1	DFH, Page 1	0000_0000 b
PWM1CON1	9DH, Page 2	0000_0000 b
PWM2CON1	C5H, Page 2	0000_0000 b
PWM3CON1	D5H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PWMMOD[1:0]		GP	PWMTYP	FBINEN	PWMDIV[2:0]		
R/W		R/W	R/W	R/W	R/W		

Bit	Name	Description
[7:6]	**PWMMOD[1:0]**	**PWM Mode Select** 00 = Independent mode. 01 = Complementary mode. 10 = Synchronized mode. 11 = Reserved.

Independent Mode

Independent mode is enabled when PWMMOD[1:0] (PWMnCON1[7:6]) is [0:0]. It is the default mode of PWM. PG0, PG1, PG2, PG3, PG4 and PG5 output PWM signals independently.

Complementary Mode with Dead-Time Insertion

Complementary mode is enabled when PWMMOD[1:0] = [0:1]. In this mode, PG0/2/4 output PWM signals the same as the independent mode. However, PG1/3/5 output the out-phase PWM signals of PG0/2/4 correspondingly, and ignore PG1/3/5 Duty register {PWMnH, PWMnL} (n:1/3/5). This mode makes PG0/PG1 a PWM complementary pair and so on PG2/PG3 and PG4/PG5.

In a real motor application, a complementary PWM output always has a need of “dead-time” insertion to prevent damage of the power switching device like GPIBs due to being active on simultaneously of the upper and lower switches of the half bridge, even in a “μs” duration. For a power switch device physically cannot switch on/off instantly. For the ML51/ML54/ML56 Series PWM, each PWM pair share a 9-bit dead-time down-counter PWM0DTCNT used to produce the off time between two PWM signals in the same pair. On implementation, a 0-to-1 signal edge delays after PWM0DTCNT timer underflows. The timing diagram illustrates the complementary mode with dead-time insertion of PG0/PG1 pair. Pairs of PG2/PG3 and PG4/PG5 have the same dead-time circuit. Each pair has its own dead-time enabling bit in the field of PWMnDTEN [3:0].

Note that the PWM0DTCNT and PWMnDTEN registers are all TA write protection. The dead-time control are also valid only when the PWM is configured in its complementary mode.

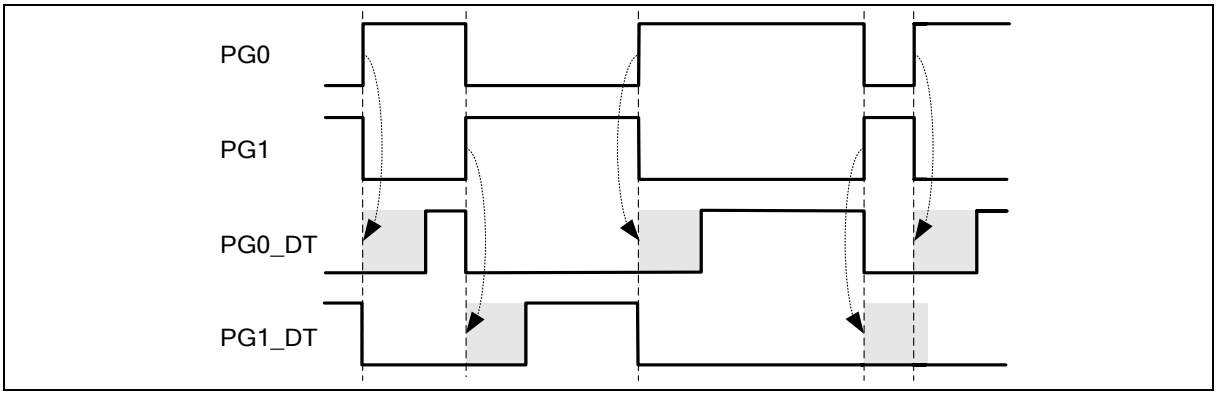


Figure 6.9-7 PWM Complementary Mode with Dead-time Insertion

PWM0DTEN – PWM Dead-time Enable (TA Protected)

Register	SFR Address	Reset Value
PWM0DTEN	F9H, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
-	**-**	**-**	**PWMnDTCNT.8**	**-**	**PDT45EN**	**PDT23EN**	**PDT01EN**
-	-	-	R/W	-	R/W	R/W	R/W

Bit	Name	Description
[7:5]	**0**	Reserved
[4]	**PWMnDTCNT.8**	**PWM Dead-Time Counter Bit 8** See PWMnDTCNT register.
[3]	**0**	Reserved
[2]	**PDT45EN**	**PWM4/5 Pair Dead-Time Insertion Enable** This bit is valid only when PWM4/5 is under complementary mode. 0 = No delay on GP4/GP5 pair signals. 1 = Insert dead-time delay on the rising edge of GP4/GP5 pair signals.
[1]	**PDT23EN**	**PWM2/3 Pair Dead-Time Insertion Enable** This bit is valid only when PWM2/3 is under complementary mode. 0 = No delay on GP2/GP3 pair signals. 1 = Insert dead-time delay on the rising edge of GP2/GP3 pair signals.
[0]	**PDT01EN**	**PWM0/1 Pair Dead-Time Insertion Enable** This bit is valid only when PWM0/1 is under complementary mode. 0 = No delay on GP0/GP1 pair signals. 1 = Insert dead-time delay on the rising edge of GP0/GP1 pair signals.

PWM0DTCNT – PWM Dead-time Counter (TA Protected)

Register	SFR Address	Reset Value
PWM0DTCNT	FAH, Page 1, TA protected	0000_0000 b

7	6	5	4	3	2	1	0
PWM0DTCNT[7:0]							
R/W							

Bit	Name	Description
[7:0]	**PWM0DTCNT[7:0]**	**PWM Dead-Time Counter Low Byte** This 8-bit field combined with PWMnDTEN .4 forms a 9-bit PWM dead-time counter PWM0DTCNT. This counter is valid only when PWM is under complementary mode and the correspond PWMnDTEN bit for PWM pair is set. PWM dead-time = $\frac{PDTCNT+1}{F_{SYS}}$. Note that user should not modify PWM0DTCNT during PWM run time.

Synchronous Mode

Synchronous mode is enabled when PWMMOD[1:0] = [1:0]. In this mode, PG0/2/4 output PWM signals the same as the independent mode. PG1/3/5 output just the same in-phase PWM signals of PG02/4 correspondingly.

6.9.4.6 Mask Output Control

Each PWM signal can be software masked by driving a specified level of PWM signal. The PWM mask output function is quite useful when controlling Electrical Commutation Motor like a BLDC. PWMnMEN contains six bits, those determine which channel of PWM signal will be masked. PWMnMD set the individual mask level of each PWM channel. The default value of PWMnMEN is 00H, which makes all outputs of PWM channels follow signals from PWM generator. Note that the masked level is reversed or not by PWM0NP setting on PWM output pins.

PWMxMEN – PWMnCx Mask Enable, n=0,1,2,3;x=0,1,2,3,4,5

Register	SFR Address	Reset Value
PWM0MEN	FBH, Page 1	0000_0000 b
PWM1MEN	8DH, Page 2	0000_0000 b
PWM2MEN	BDH, Page 2	0000_0000 b
PWM3MEN	CDH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	PMEN5	PMEN4	PMEN3	PMEN2	PMEN1	PMEN0
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[5:0]	PMENn	**PWMnCx Mask Enable** 0 = PWMnCx signal outputs from its PWM generator. 1 = PWMnCx signal is masked by PMDx. **Note:** PMEN2~5 are only for PWM0.

PWMnMD – PWM Mask Data

Register	SFR Address	Reset Value
PWM0MD	FCH, Page 1	0000_0000 b
PWM1MD	8CH, Page 2	0000_0000 b
PWM2MD	BCH, Page 2	0000_0000 b
PWM3MD	CCH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	PMD5	PMD4	PMD3	PMD2	PMD1	PMD0
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:6]	0	**Reserved** The bits are always read as 0.
[5:0]	PMDx	**PWMnCx Mask Data** The PWMnCx signal outputs mask data once its corresponding PMENx is set. 0 = PWMnCx signal is masked by 0. 1 = PWMnCx signal is masked by 1. **Note:** PMD2~5 are only for PWM0.

6.9.4.7 Fault Brake

The Fault Brake function is usually implemented in conjunction with an enhanced PWM circuit. It rules as a fault detection input to protect the motor system from damage. Fault Brake pin input (FB) is valid when FBINEN (PWMnCON1.3) is set. When Fault Brake is asserted PWM signals will be individually overwritten by PWMnFBD corresponding bits. PWMRUN (PWMnCON0.7) will also be automatically cleared by hardware to stop PWM generating. The PWM 16-bit counter will also be reset as 0000H. A indicating flag FBF will be set by hardware to assert a Fault Brake interrupt if enabled. PWMnFBD data output remains even after the FBF is cleared by software. User should resume the PWM output only by setting PWMRUN again. Meanwhile the Fault Brake state will be released and PWM waveform outputs on pins as usual. Fault Brake input has a polarity selection by FBINLS (PWMnFBD.6) bit. Note that the Fault Brake signal feed in FB pin should be longer than eight-system-clock time for FB pin input has a permanent $8/F_{SYS}$ de-bouncing, which avoids fake Fault Brake event by input noise. The other path to trigger a Fault Brake event is the ADC compare event. It asserts the Fault Brake behavior just the same as FB pin input. See Sector 6.14.4.3“ADC Conversion Result Comparator”.

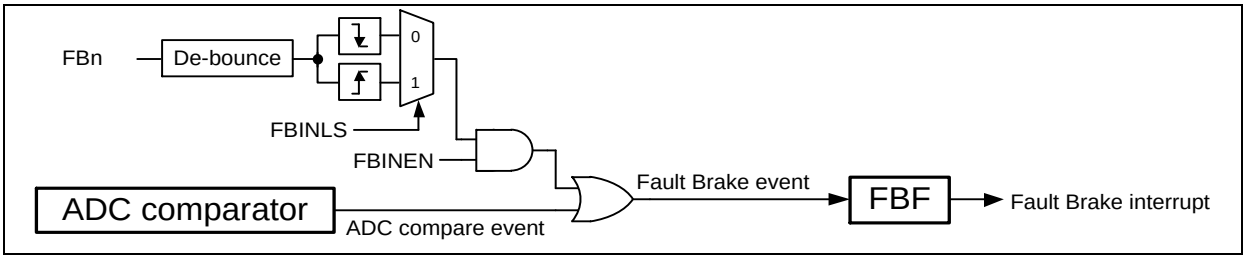


Figure 6.9-8 Fault Brake Function Block Diagram

PWMnCON1 – PWM Control 1

Register	SFR Address	Reset Value
PWM0CON1	DFH, Page 1	0000_0000 b
PWM1CON1	9DH, Page 2	0000_0000 b
PWM2CON1	C5H, Page 2	0000_0000 b
PWM3CON1	D5H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PWMMOD[1:0]		GP	PWMTYP	FBINEN	PWMDIV[2:0]		
R/W		R/W	R/W	R/W	R/W		

Bit	Name	Description
[3]	**FBINEN**	**FB Pin Input Enable** 0 = PWM0 output Fault Braked by FB pin input Disabled. 1 = PWM0 output Fault Braked by FB pin input Enabled. Once an edge, which matches FBINLS (PWM0FBD.6) selection, occurs on FB pin, PWM0CH0~5 output Fault Brake data in PWMnFBD register. PWMRUN (PWM0CON0.7) will also be automatically cleared by hardware. The PWM output resumes when PWM0RUN is set again. **Note:** This bit is only vaild in PWM0

PWMnFBD – PWM Fault Brake Data

Register	SFR Address	Reset Value
PWM0FBD	D7H, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
FBF	**FBINLS**	**FBD5**	**FBD4**	**FBD3**	**FBD2**	**FBD1**	**FBD0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**FBF**	**Fault Brake Flag** This flag is set when FBINEN is set as 1 and FB pin detects an edge, which matches FBINLS (PWM0FBD.6) selection. This bit is cleared by software. After FBF is cleared, Fault Brake data output will not be released until PWM0RUN (PWM0CON0.7) is set.
[6]	**FBINLS**	**PWM_BRAKE Pin Input Level Selection** 0 = Falling edge. 1 = Rising edge.
[5:0]	**FBDn**	**PWMn Fault Brake Data** 0 = PWMn signal is overwritten by 0 once Fault Brake asserted. 1 = PWMn signal is overwritten by 1 once Fault Brake asserted.

6.9.4.8 Polarity Control

Each PWM output channel has its independent polarity control bit, PNP0~PNP5. The default is high active level on all control fields implemented with positive logic. It means the power switch is ON when PWM outputs high level and OFF when low level. User can easily configure all setting with positive logic and then set PWMnNP bit to make PWM actually outputs according to the negative logic.

PWM0NP – PWM Negative Polarity

Register	SFR Address	Reset Value
PWM0NP	D6H, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
-	-	PNP5	PNP4	PNP3	PNP2	PNP1	PNP0
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[5:0]	PNPn	**PWMn Negative Polarity Output Enable** 0 = PWMn signal outputs directly on PWMn pin. 1 = PWMn signal outputs inversely on PWMn pin.

6.9.5 PWM Interrupt

The PWM module has a flag PWMF (PWMnCON0.5) to indicate certain point of each complete PWM period. The indicating PWM channel and point can be selected by INTSEL[2:0] and INTTYP[1:0] (PWMnINTC[2:0] and [5:4]). Note that the center point and the end point interrupts are only available when PWM operates in its center-aligned type. PWMF is cleared by software.

PWMnINTC – PWM Interrupt Control

Register	SFR Address	Reset Value
PWM0INTC	B7H, Page 1	0000_0000 b
PWM1INTC	9EH, Page 2	0000_0000 b
PWM2INTC	C6H, Page 2	0000_0000 b
PWM3INTC	D6H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
-	**-**	**INTTYP1**	**INTTYP0**	**-**	**INTSEL2**	**INTSEL1**	**INTSEL0**
-	-	R/W	R/W	-	R/W	R/W	R/W

Bit	Name	Description
[7:6]	**-**	**Reserved**
[5:4]	**INTTYP[1:0]**	**PWM Interrupt Type Select** These bit select PWM interrupt type. 00 = Falling edge on PWMn_CH0/1/2/3/4/5 pin. 01 = Rising edge on PWMn_CH0/1/2/3/4/5 pin. 10 = Central point of a PWM period. 11 = End point of a PWM period. Note that the central point interrupt or the end point interrupt is only available while PWM operates in center-aligned type.
[3]	**-**	**Reserved**
[2:0]	**INTSEL[2:0]**	**PWM Interrupt Pair Select** These bits select which PWM channel asserts PWM interrupt when PWM interrupt type is selected as falling or rising edge on PWM0/1/2/3/4/5 pin.. 000 = PWMn_CH0. 001 = PWMn_CH1. 010 = PWMn_CH2. 011 = PWMn_CH3. 100 = PWMn_CH4. 101 = PWMn_CH5. Others = PWMn_CH0.

The PWM interrupt related with PWM waveform is shown as figure below.

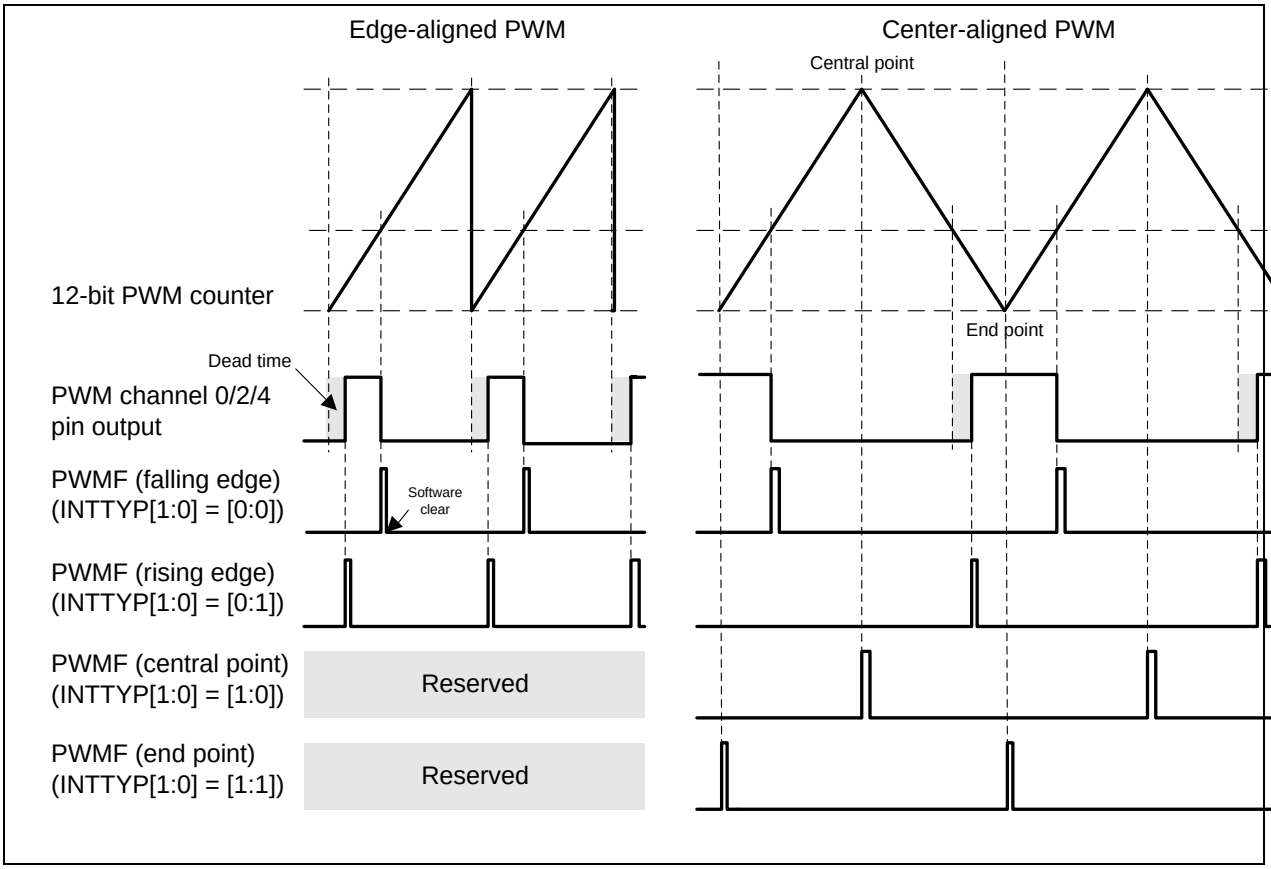


Figure 6.9-9 PWM Interrupt Type

Fault Brake event requests another interrupt, Fault Brake interrupt. It has different interrupt vector from PWM interrupt. When either Fault Brake pin input event or ADC compare event occurs, FBF (PWMnFBD.7) will be set by hardware. It generates Fault Brake interrupt if enabled. The Fault Brake interrupt enable bit is EFB0 (EIE0.5). FBF Is cleared via software.

6.9.6 Register Description

PWM0CON0 – PWM Control Register0

Register	SFR Address	Reset Value
PWM0CON0	D1H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
PWM0RUN	**LOAD**	**PWMF**	**CLRPWM**	**-**	**-**	**-**	**-**
R/W	R/W	R/W	R/W	-	-	-	-

Bit	Name	Description
[7]	**PWM0RUN**	**PWM0 Run Enable** 0 = PWM0 stays in idle. 1 = PWM0 starts running.
[6]	**LOAD**	**PWM New Period and Duty Load** This bit is used to load period and duty Register Description in their buffer if new period or duty value needs to be updated. The loading will act while a PWM period is completed. The new period and duty affected on the next PWM cycle. After the loading is complete, LOAD will be automatically cleared via hardware. The meaning of writing and reading LOAD bit is different. Writing: 0 = No effect. 1 = Load new period and duty in their buffers while a PWM period is completed. Reading: 0 = A loading of new period and duty is finished. 1 = A loading of new period and duty is not yet finished.
[5]	**PWMF**	**PWM Flag** This flag is set according to definitions of INTSEL[2:0] and INTTYP[1:0] in PWMnINTC. This bit is cleared by software.
[4]	**CLRPWM**	**Clear PWM Counter** Setting this bit clears the value of PWM 16-bit counter for resetting to 0000H. After the counter value is cleared, CLRPWM will be automatically cleared via hardware. The meaning of writing and reading CLRPWM bit is different. Writing: 0 = No effect. 1 = Clearing PWM 16-bit counter. Reading: 0 = PWM 16-bit counter is completely cleared. 1 = PWM 16-bit counter is not yet cleared.
[3:0]	**-**	Reserved

PWMnCON0 – PWM Control Register0

Register	SFR Address	Reset Value
PWM1CON0	9CH, Page 2	0000_0000 b
PWM2CON0	C4H, Page 2	0000_0000 b
PWM3CON0	D4H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PWMnRUN	**LOAD**	**PWMF**	**CLRPWM**	**-**	**-**	**-**	**-**
R/W	R/W	R/W	R/W	-	-	-	-

Bit	Name	Description
[7]	**PWMnRUN**	**PWMn Run Enable** 0 = PWM stays in idle. 1 = PWM starts running.
[6]	**LOAD**	**PWM New Period and Duty Load** This bit is used to load period and duty Register Description in their buffer if new period or duty value needs to be updated. The loading will act while a PWM period is completed. The new period and duty affected on the next PWM cycle. After the loading is complete, LOAD will be automatically cleared via hardware. The meaning of writing and reading LOAD bit is different. Writing: 0 = No effect. 1 = Load new period and duty in their buffers while a PWM period is completed. Reading: 0 = A loading of new period and duty is finished. 1 = A loading of new period and duty is not yet finished.
[5]	**PWMF**	**PWM Flag** This flag is set according to definitions of INTSEL[2:0] and INTTYP[1:0] in PWMnINTC. This bit is cleared by software.
[4]	**CLRPWM**	**Clear PWM Counter** Setting this bit clears the value of PWM 16-bit counter for resetting to 0000H. After the counter value is cleared, CLRPWM will be automatically cleared via hardware. The meaning of writing and reading CLRPWM bit is different. Writing: 0 = No effect. 1 = Clearing PWM 16-bit counter. Reading: 0 = PWM 16-bit counter is completely cleared. 1 = PWM 16-bit counter is not yet cleared.
[3:0]	**-**	Reserved

PWMnCON1 – PWM Control 1

Register	SFR Address	Reset Value
PWM0CON1	DFH, Page 1	0000_0000 b
PWM1CON1	9DH, Page 2	0000_0000 b
PWM2CON1	C5H, Page 2	0000_0000 b
PWM3CON1	D5H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PWMMOD[1:0]		GP	PWMTYP	FBINEN	PWMDIV[2:0]		
R/W		R/W	R/W	R/W	R/W		

Bit	Name	Description
[7:6]	**PWMMOD[1:0]**	**PWM Mode Select** 00 = Independent mode. 01 = Complementary mode. 10 = Synchronized mode. 11 = Reserved.
[5]	**GP**	**Group Mode Enable** This bit enables the group mode. If enabled, the duty of first three pairs of PWM are decided by PWM01H and PWM01L rather than their original duty Register Description. 0 = Group mode Disabled. 1 = Group mode Enabled.
[4]	**PWMTYP**	**PWM Type Select** 0 = Edge-aligned PWM. 1 = Center-aligned PWM.
[3]	**FBINEN**	**FB Pin Input Enable** 0 = PWM0 output Fault Braked by FB pin input Disabled. 1 = PWM0 output Fault Braked by FB pin input Enabled. Once an edge, which matches FBINLS (PWM0FBD.6) selection, occurs on FB pin, PWM0CH0~5 output Fault Brake data in PWMnFBD register. PWMRUN (PWM0CON0.7) will also be automatically cleared by hardware. The PWM output resumes when PWM0RUN is set again. **Note:** This bit is only vaild in PWM0

Bit	Name	Description
[2:0]	**PWMDIV[2:0]**	**PWM Clock Divider** This field decides the pre-scale of PWM clock source. 000 = 1/1. 001 = 1/2 010 = 1/4. 011 = 1/8. 100 = 1/16. 101 = 1/32. 110 = 1/64. 111 = 1/128.

PWMnPL – PWM Period Low Byte

Register	SFR Address	Reset Value
PWM0PL	D9H, Page 1	0000_0000 b
PWM1PL	99H, Page 2	0000_0000 b
PWM2PL	C1H, Page 2	0000_0000 b
PWM3PL	D1H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PWMnP[7:0]							
R/W							

Bit	Name	Description
[7:0]	**PWMnP[7:0]**	**PWMn Period Low Byte** This byte with PWMnPH controls the period of the PWM generator signal.

PWMnPH – PWM Period High Byte

Register	SFR Address	Reset Value
PWM0PH	D1H, Page 1	0000_0000 b
PWM1PH	86H, Page 2	0000_0000 b
PWM2PH	B9H, Page 2	0000_0000 b
PWM3PH	C9H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PWMnP[15:8]							
R/W							

Bit	Name	Description
[7:0]	**PWMnP[15:8]**	**PWM Period High Byte** This byte with PWMnPL controls the period of the PWM generator signal.

PWMnCxH – PWM0/1/2/3 Channel 0~5 Duty High Byte n=0,1,2,3; x=0,1,2,3,4,5

Register	SFR Address	Reset Value
PWM0C0H	D2H, Page 1	0000_0000 b
PWM0C1H	D3H, Page 1	0000_0000 b
PWM0C2H	D4H, Page 1	0000_0000 b
PWM0C3H	D5H, Page 1	0000_0000 b
PWM0C4H	C4H, Page 1	0000_0000 b
PWM0C5H	C5H, Page 1	0000_0000 b
PWM1C0H	8AH, Page 2	0000_0000 b
PWM1C1H	8BH, Page 2	0000_0000 b
PWM2C0H	BAH, Page 2	0000_0000 b
PWM2C1H	BBH, Page 2	0000_0000 b
PWM3C0H	CAH, Page 2	0000_0000 b
PWM3C1H	CBH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PWMnCx [15:8], n=0,1,2,3; x=0,1,2,3,4,5							
R/W							

Bit	Name	Description
[7:0]	**PWMnCx [15:8], n=0,1,2,3; x=0,1,2,3,4,5**	**PWMnCx Duty High Byte** This byte with PWMnCxL controls the duty of the output signal PGx from PWM generator.

PWMnCxL – PWM0/1/2/3 Channel 0~5 Duty Low Byte n=0,1,2,3; x=0,1,2,3,4,5

Register	SFR Address	Reset Value
PWM0C0L	DAH, Page 1	0000_0000 b
PWM0C1L	DBH, Page 1	0000_0000 b
PWM0C2L	DCH, Page 1	0000_0000 b
PWM0C3L	DDH, Page 1	0000_0000 b
PWM0C4L	CCH, Page 1	0000_0000 b
PWM0C5L	CDH, Page 1	0000_0000 b
PWM1C0L	9AH, Page 2	0000_0000 b
PWM1C1L	9BH, Page 2	0000_0000 b
PWM2C0L	C2H, Page 2	0000_0000 b
PWM2C1L	C3H, Page 2	0000_0000 b
PWM3C0L	D2H, Page 2	0000_0000 b
PWM3C1L	D3H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PWMnCx [7:0], n=0,1,2,3; x=0,1,2,3,4,5							
R/W							

Bit	Name	Description
[7:0]	**PWMnCx [7:0], n=0,1,2,3; x=0,1,2,3,4,5**	**PWMnCx Duty Low Byte** This byte with PWMnCxH controls the duty of the output signal PGx from PWM generator.

PWM0DTEN – PWM Dead-time Enable (TA Protected)

Register	SFR Address	Reset Value
PWM0DTEN	F9H, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	PWMnDTCNT.8	-	PDT45EN	PDT23EN	PDT01EN
-	-	-	R/W	-	R/W	R/W	R/W

Bit	Name	Description
[7:5]	0	Reserved
[4]	PWMnDTCNT.8	**PWM Dead-Time Counter Bit 8** See PWMnDTCNT register.
[3]	0	Reserved
[2]	PDT45EN	**PWM4/5 Pair Dead-Time Insertion Enable** This bit is valid only when PWM4/5 is under complementary mode. 0 = No delay on GP4/GP5 pair signals. 1 = Insert dead-time delay on the rising edge of GP4/GP5 pair signals.
[1]	PDT23EN	**PWM2/3 Pair Dead-Time Insertion Enable** This bit is valid only when PWM2/3 is under complementary mode. 0 = No delay on GP2/GP3 pair signals. 1 = Insert dead-time delay on the rising edge of GP2/GP3 pair signals.
[0]	PDT01EN	**PWM0/1 Pair Dead-Time Insertion Enable** This bit is valid only when PWM0/1 is under complementary mode. 0 = No delay on GP0/GP1 pair signals. 1 = Insert dead-time delay on the rising edge of GP0/GP1 pair signals.

PWM0DTCNT – PWM Dead-time Counter (TA Protected)

Register	SFR Address	Reset Value
PWM0DTCNT	FAH, Page 1, TA protected	0000_0000 b

7	6	5	4	3	2	1	0
PWM0DTCNT[7:0]							
R/W							

Bit	Name	Description
[7:0]	**PWM0DTCNT[7:0]**	**PWM Dead-Time Counter Low Byte** This 8-bit field combined with PWMnDTEN .4 forms a 9-bit PWM dead-time counter PWM0DTCNT. This counter is valid only when PWM is under complementary mode and the correspond PWMnDTEN bit for PWM pair is set. PWM dead-time = $\frac{PDTCNT + 1}{F_{SYS}}$. Note that user should not modify PWM0DTCNT during PWM run time.

PWMxMEN – PWMnCx Mask Enable, n=0,1,2,3;x=0,1,2,3,4,5

Register	SFR Address	Reset Value
PWM0MEN	FBH, Page 1	0000_0000 b
PWM1MEN	8DH, Page 2	0000_0000 b
PWM2MEN	BDH, Page 2	0000_0000 b
PWM3MEN	CDH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	PMEN5	PMEN4	PMEN3	PMEN2	PMEN1	PMEN0
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[5:0]	PMENn	**PWMnCx Mask Enable** 0 = PWMnCx signal outputs from its PWM generator. 1 = PWMnCx signal is masked by PMDx. **Note:** PMEN2~5 are only for PWM0.

PWMnMD – PWM Mask Data

Register	SFR Address	Reset Value
PWM0MD	FCH, Page 1	0000_0000 b
PWM1MD	8CH, Page 2	0000_0000 b
PWM2MD	BCH, Page 2	0000_0000 b
PWM3MD	CCH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
-	**-**	**PMD5**	**PMD4**	**PMD3**	**PMD2**	**PMD1**	**PMD0**
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:6]	**0**	**Reserved** The bits are always read as 0.
[5:0]	**PMDx**	**PWMnCx Mask Data** The PWMnCx signal outputs mask data once its corresponding PMENx is set. 0 = PWMnCx signal is masked by 0. 1 = PWMnCx signal is masked by 1. **Note:** PMD2~5 are only for PWM0.

PWMnFBD – PWM Fault Brake Data

Register	SFR Address	Reset Value
PWM0FBD	D7H, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
FBF	**FBINLS**	**FBD5**	**FBD4**	**FBD3**	**FBD2**	**FBD1**	**FBD0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**FBF**	**Fault Brake Flag** This flag is set when FBINEN is set as 1 and FB pin detects an edge, which matches FBINLS (PWM0FBD.6) selection. This bit is cleared by software. After FBF is cleared, Fault Brake data output will not be released until PWM0RUN (PWM0CON0.7) is set.
[6]	**FBINLS**	**PWM_BRAKE Pin Input Level Selection** 0 = Falling edge. 1 = Rising edge.
[5:0]	**FBDn**	**PWMn Fault Brake Data** 0 = PWMn signal is overwritten by 0 once Fault Brake asserted. 1 = PWMn signal is overwritten by 1 once Fault Brake asserted.

PWM0NP – PWM Negative Polarity

Register	SFR Address	Reset Value
PWM0NP	D6H, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
-	-	PNP5	PNP4	PNP3	PNP2	PNP1	PNP0
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[5:0]	PNPn	**PWMn Negative Polarity Output Enable** 0 = PWMn signal outputs directly on PWMn pin. 1 = PWMn signal outputs inversely on PWMn pin.

PWMnINTC – PWM Interrupt Control

Register	SFR Address	Reset Value
PWM0INTC	B7H, Page 1	0000_0000 b
PWM1INTC	9EH, Page 2	0000_0000 b
PWM2INTC	C6H, Page 2	0000_0000 b
PWM3INTC	D6H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
-	**-**	**INTTYP1**	**INTTYP0**	**-**	**INTSEL2**	**INTSEL1**	**INTSEL0**
-	-	R/W	R/W	-	R/W	R/W	R/W

Bit	Name	Description
[7:6]	**-**	Reserved.
[5:4]	**INTTYP[1:0]**	**PWM Interrupt Type Select** These bit select PWM interrupt type. 00 = Falling edge on PWMn_CH0/1/2/3/4/5 pin. 01 = Rising edge on PWMn_CH0/1/2/3/4/5 pin. 10 = Central point of a PWM period. 11 = End point of a PWM period. Note that the central point interrupt or the end point interrupt is only available while PWM operates in center-aligned type.
[3]	**-**	Reserved.
[2:0]	**INTSEL[2:0]**	**PWM Interrupt Pair Select** These bits select which PWM channel asserts PWM interrupt when PWM interrupt type is selected as falling or rising edge on PWM0/1/2/3/4/5 pin.. 000 = PWMn_CH0. 001 = PWMn_CH1. 010 = PWMn_CH2. 011 = PWMn_CH3. 100 = PWMn_CH4. 101 = PWMn_CH5. Others = PWMn_CH0.

6.10 Serial Port (UART0 & UART1)

6.10.1 Overview

The ML51/ML54/ML56 Series includes two enhanced full duplex serial ports enhanced with automatic address recognition and framing error detection. As control bits of these two serial ports are implemented the same. Generally speaking, in the following contents, there will not be any reference to serial port 1, but only to serial port 0.

Each serial port supports one synchronous communication mode, Mode 0, and three modes of full duplex UART (Universal Asynchronous Receiver and Transmitter), Mode 1, 2, and 3. This means it can transmit and receive simultaneously. The serial port is also receiving-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the register. The receiving and transmitting registers are both accessed at SBUF. Writing to SBUF loads the transmitting register, and reading SBUF accesses a physically separate receiving register. There are four operation modes in serial port. In all four modes, transmission initiates by any instruction that uses SBUF as a destination register.

6.10.2 Features

- Supports up to 2 UARTs: UART0, UART1
- Supports 2 Smart Card configuration as UART function as UART2 and UART3.
- UART baud rate clock from HIRC or HXT.
- Full-duplex asynchronous communications
- Programmable 9th bit.
- TXD and RXD pins of UART0 exchangeable via software.

6.10.3 Functional Description

6.10.3.1 Operation Mode

Mode 0

Mode 0 provides synchronous communication with external devices. Serial data centers and exits through RXD pin. TXD outputs the shift clocks. 8-bit frame of data are transmitted or received. Mode 0 therefore provides half-duplex communication because the transmitting or receiving data is via the same data line RXD. The baud rate is enhanced to be selected as $F_{SYS}/12$ if SM2 (SCON.5) is 0 or as $F_{SYS}/2$ if SM2 is 1. Note that whenever transmitting or receiving, the serial clock is always generated by the MCU. Thus any device on the serial port in Mode 0 should accept the MCU as the master. Figure 6.10-1 shows the associated timing of the serial port in Mode 0.

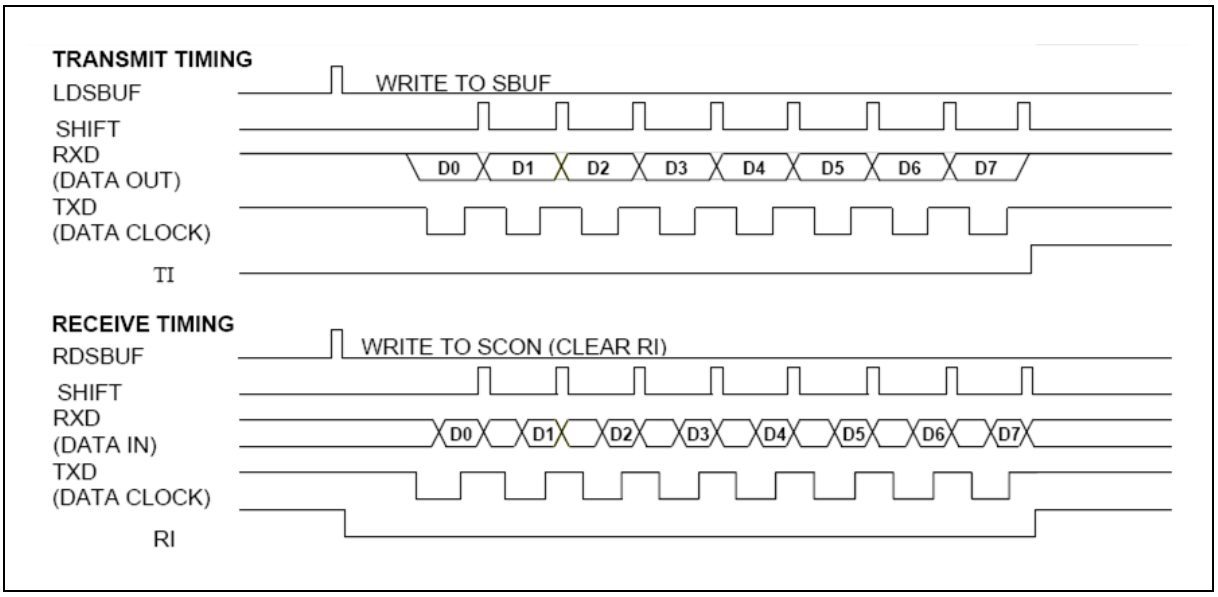


Figure 6.10-1 Serial Port Mode 0 Timing Diagram

As shown there is one bi-directional data line (RXD) and one shift clock line (TXD). The shift clocks are used to shift data in or out of the serial port controller bit by bit for a serial communication. Data bits enter or emit LSB first. The band rate is equal to the shift clock frequency.

Transmission is initiated by any instruction writes to SBUF. The control block will then shift out the clocks and begin to transfer data until all 8 bits are complete. Then the transmitted flag TI (SCON.1) will be set 1 to indicate one byte transmitting complete.

Reception is initiated by the condition REN (SCON.4) = 1 and RI (SCON.0) = 0. This condition tells the serial port controller that there is data to be shifted in. This process will continue until 8 bits have been received. Then the received flag RI will be set as 1. User can clear RI to triggering the next byte reception.

Mode 1

Mode 1 supports asynchronous, full duplex serial communication. The asynchronous mode is commonly used for communication with PCs, modems or other similar interfaces. In Mode 1, 10 bits are transmitted through TXD or received through RXD including a start bit (logic 0), 8 data bits (LSB first) and a stop bit (logic 1). The baud rate is determined by the Timer 1. SMOD (PCON.7) setting 1 makes the baud rate double. Figure 6.10-2 shows the associated timings of the serial port in Mode 1 for transmitting and receiving.

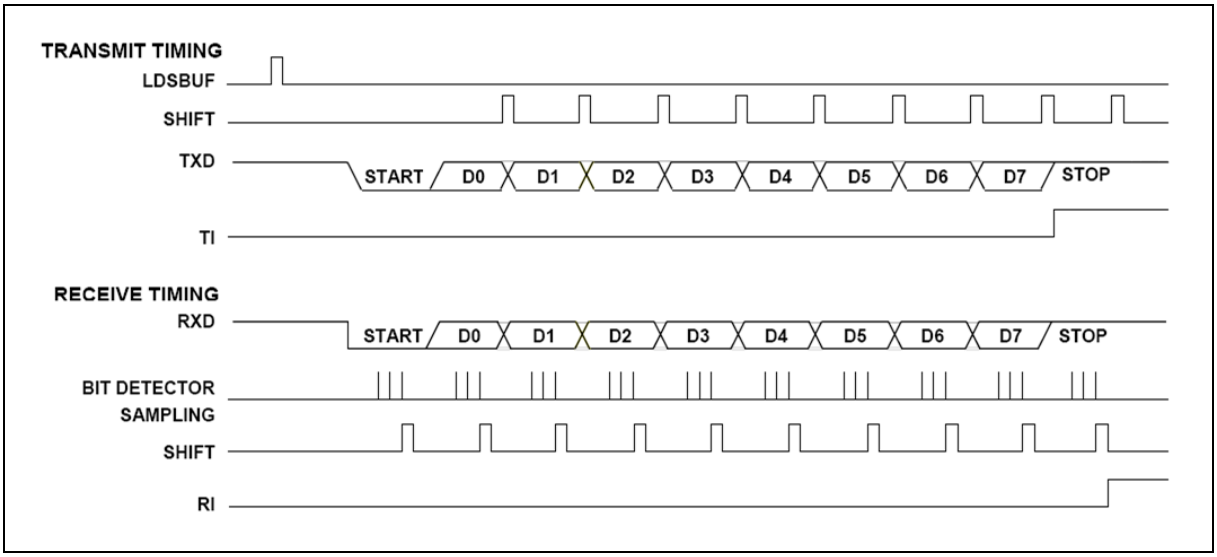


Figure 6.10-2 Serial Port Mode 1 Timing Diagram

Transmission is initiated by any writing instructions to SBUF. Transmission takes place on TXD pin. First the start bit comes out, the 8-bit data follows to be shifted out and then ends with a stop bit. After the stop bit appears, TI (SCON.1) will be set to indicate one byte transmission complete. All bits are shifted out depending on the rate determined by the baud rate generator.

Once the baud rate generator is activated and REN (SCON.4) is 1, the reception can begin at any time. Reception is initiated by a detected 1-to-0 transition at RXD. Data will be sampled and shifted in at the selected baud rate. In the midst of the stop bit, certain conditions should be met to load SBUF with the received data:

1. RI (SCON.0) = 0, and

2. Either SM2 (SCON.5) = 0, or the received stop bit = 1 while SM2 = 1 and the received data matches "Given" or "Broadcast" address. (For enhancement function, see Section 6.10.3.4"Multiprocessor Communication" and Section 6.10.3.5"Automatic Address Recognition".)

If these conditions are met, then the SBUF will be loaded with the received data, the RB8 (SCON.2) with stop bit, and RI will be set. If these conditions fail, there will be no data loaded and RI will remain 0. After above receiving progress, the serial control will look forward another 1-to-0 transition on RXD pin to start next data reception.

Mode 2

Mode 2 supports asynchronous, full duplex serial communication. Different from Mode1, there are 11 bits to be transmitted or received. They are a start bit (logic 0), 8 data bits (LSB first), a programmable 9th bit TB8 or RB8 bit and a stop bit (logic 1). The most common use of 9th bit is to put the parity bit in it or to label address or data frame for multiprocessor communication. The baud rate is fixed as 1/32 or 1/64 the system clock frequency depending on SMOD (PCON.7) bit. Figure 6.10-3 shows the associated timings of the serial port in Mode 2 for transmitting and receiving.

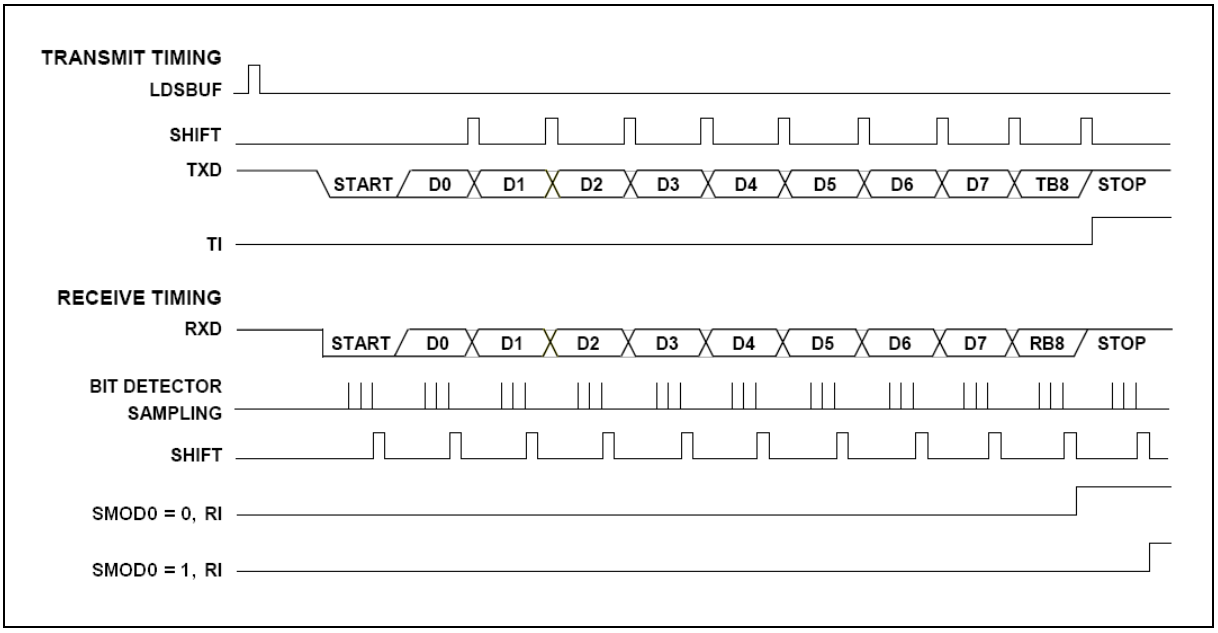


Figure 6.10-3 Serial Port Mode 2 and 3 Timing Diagram

Transmission is initiated by any writing instructions to SBUF. Transmission takes place on TXD pin. First the start bit comes out, the 8-bit data and bit TB8 (SCON.3) follows to be shifted out and then ends with a stop bit. After the stop bit appears, TI will be set to indicate the transmission complete.

While REN is set, the reception is allowed at any time. A falling edge of a start bit on RXD will initiate the reception progress. Data will be sampled and shifted in at the selected baud rate. In the midst of the stop bit, certain conditions should be met to load SBUF with the received data:

1. RI (SCON.0) = 0, and

2. Either SM2 (SCON.5) = 0, or the received 9th bit = 1 while SM2 = 1 and the received data matches “Given” or “Broadcast” address. (For enhancement function, see Section 6.10.3.4“Multiprocessor Communication” and Section 6.10.3.5“Automatic Address Recognition”.)

If these conditions are met, the SBUF will be loaded with the received data, the RB8(SCON.2) with the received 9th bit and RI will be set. If these conditions fail, there will be no data loaded and RI will remain 0. After above receiving progress, the serial control will look forward another 1-to-0 transition on RXD pin to start next data reception.

Mode 3

Mode 3 has the same operation as Mode 2, except its baud rate clock source uses Timer 1 overflows as its baud rate clocks. See Figure 6.10-3 for timing diagram of Mode 3. It has no difference from Mode 2.

6.10.3.2 Baud Rate

The baud rate source and speed for different modes of serial port is quite different from one another. All cases are listed in Table 6.10-1 Serial Port 0 Mode / baud rate DescriptionThe user should calculate the baud rate according to their system configuration.

In Mode 1 or 3, the baud rate clock source of UART0 can be selected from Timer 1 or Timer 3. User can select the baud rate clock source by BRCK (T3CON.5). For UART1, its baud rate clock comes only from Timer 3 as its unique clock source.

When using Timer 1 as the baud rate clock source, note that the Timer 1 interrupt should be disabled. Timer 1 itself can be configured for either “Timer” or “Counter” operation. It can be in any of its three running modes. However, in the most typical applications, it is configured for “Timer” operation, in the auto-reload mode (Mode 2). If using Timer 3 as the baud rate generator, its interrupt should also be

disabled.

Following shows all UART mode and baudrate fomula:

<table>
<tr><th>Mode</th><th>SM0 / SM1 (SCON[7:6])</th><th>SM2 (SCON[5])</th><th>SMOD (PCON[7])</th><th>Frame Bits</th><th colspan="2">Baud Rate</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">00</td><td>0</td><td rowspan="2">-</td><td rowspan="2">8</td><td colspan="2">FSYS divided by 12</td></tr>
<tr><td>1</td><td colspan="2">FSYS divided by 2</td></tr>
<tr><td rowspan="6">1</td><td rowspan="6">01</td><td rowspan="6">-</td><td rowspan="3">0</td><td rowspan="6">10</td><td>Time1
TM1 (CKCON[3]) = 0</td><td>$\frac{1}{32}\times\frac{F_{SYS}}{12\times(256-TH1)}$</td></tr>
<tr><td>Time1
TM1 (CKCON[3]) = 1</td><td>$\frac{1}{32}\times\frac{F_{SYS}}{(256-TH1)}$</td></tr>
<tr><td>Timer 3</td><td>$\frac{1}{32}\times\frac{F_{SYS}}{Pre\text{-}scale\times(65536-(256\times RH3+RL3))}$</td></tr>
<tr><td rowspan="3">1</td><td>Time1
TM1 (CKCON[3]) = 0</td><td>$\frac{1}{16}\times\frac{F_{SYS}}{12\times(256-TH1)}$</td></tr>
<tr><td>Time1
TM1 (CKCON[3]) = 1</td><td>$\frac{1}{16}\times\frac{F_{SYS}}{(256-TH1)}$</td></tr>
<tr><td>Timer 3</td><td>$\frac{1}{16}\times\frac{F_{SYS}}{Pre\text{-}scale\times(65536-(256\times RH3+RL3))}$</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">10</td><td rowspan="2">-</td><td>0</td><td rowspan="2">11</td><td colspan="2">FSYS divided by 64</td></tr>
<tr><td>1</td><td colspan="2">FSYS divided by 32</td></tr>
<tr><td rowspan="6">3</td><td rowspan="6">11</td><td rowspan="6">-</td><td rowspan="3">0</td><td rowspan="6">11</td><td>Time 1
TM1 (CKCON[3]) = 0</td><td>$\frac{1}{32}\times\frac{F_{SYS}}{12\times(256-TH1)}$</td></tr>
<tr><td>Time 1
TM1 (CKCON[3]) = 1</td><td>$\frac{1}{32}\times\frac{F_{SYS}}{(256-TH1)}$</td></tr>
<tr><td>Timer 3</td><td>$\frac{1}{32}\times\frac{F_{SYS}}{Pre\text{-}scale\times(65536-(256\times RH3+RL3))}$</td></tr>
<tr><td rowspan="3">1</td><td>Time1
TM1 (CKCON[3]) = 0</td><td>$\frac{1}{16}\times\frac{F_{SYS}}{12\times(256-TH1)}$</td></tr>
<tr><td>Time1
TM1 (CKCON[3]) = 1</td><td>$\frac{1}{16}\times\frac{F_{SYS}}{(256-TH1)}$</td></tr>
<tr><td>Timer 3</td><td>$\frac{1}{16}\times\frac{F_{SYS}}{Pre\text{-}scale\times(65536-(256\times RH3+RL3))}$</td></tr>
<tr><td colspan="7">Note: Timer 1 should configured as a timer in auto-reload mode (Mode 2).</td></tr>
</table>

Table 6.10-1 Serial Port 0 Mode / baud rate Description

Mode	SM0_1 / SM1_1 (S1CON[7:6])	SMOD_1 (T3CON[7])	Frame Bits	Baud Rate	
0	00	-	8	F_{SYS} divided by 12	
1	01	0	10	**Timer 3**	$\frac{1}{32}\times\frac{F_{SYS}}{\text{Pre-scale}\times(65536-(256\times RH3+RL3))}$
		1		**Timer 3**	$\frac{1}{16}\times\frac{F_{SYS}}{\text{Pre-scale}\times(65536-(256\times RH3+RL3))}$
2	10	0	11	F_{SYS} divided by 64	
		1		F_{SYS} divided by 32	
3	11	0	11	**Timer 3**	$\frac{1}{32}\times\frac{F_{SYS}}{\text{Pre-scale}\times(65536-(256\times RH3+RL3))}$
		1		**Timer 3**	$\frac{1}{16}\times\frac{F_{SYS}}{\text{Pre-scale}\times(65536-(256\times RH3+RL3))}$

Table 6.10-2 Serial Port 1 Mode / baud rate Description

Sample code: we list the most popular UART setting Mode 1 initial step as following：

Serial port 0 (**UART0**) use **timer 1** as baudrate generator: Formula is $\frac{1}{16}\times\frac{F_{SYS}}{(256-TH1)}$

```
SCON = 0x50;        //UART0 Mode1,REN=1,TI=1
TMOD |= 0x20;       //Timer1 set to Mode2 auto reload mode (must)
PCON |= 0x80;       //UART0 Double Rate Enable
CKCON |= 0x10;      //Timer 1 as clock source
T3CON &= 0xDF;      //Timer1 as UART0 clock source
TH1 = value;
TR1=1;
```

Serial port 0 (**UART0**) use **timer 3** as baudrate generator: Formula is
$\frac{1}{16}\times\frac{F_{SYS}}{\text{Pre-scale}\times(65536-(256\times RH3+RL3))}$

```
SCON = 0x50;        //UART0 Mode1,REN=1,TI=1
PCON |= 0x80;       //UART0 Double Rate Enable
T3CON &= 0xF8;      //(Prescale=1)
T3CON |= 0x20;      //UART0 baud rate clock source = Timer3
RH3 = value high byte
RL3 = value low byte
T3CON|= 0x08;           //Trigger Timer3
```

Serial port 1 (UART1) use Timer 3 as baudrate generator: Fomula is $\frac{1}{16} \times \frac{F_{SYS}}{\text{Pre-scale} \times (65536-(256 \times RH3+RL3))}$

```
SCON_1 = 0x52;     //UART1 Mode1,REN_1=1,TI_1=1
T3CON = 0xF8;      //T3PS2=0,T3PS1=0,T3PS0=0(Prescale=1),
RH3 = value high byte
RL3 = value low byte
T3CON|= 0x08;
```

Following list some popular baudrate value base on different Fsys and the deviation value:

Fsys Value	Baud Rate	TH1 Value (Hex)	RH3,RL3 Value (Hex)	Baudrate Deviation
24000000	4800	64 (SMOD=0)	FEC8	0.160256%
	9600	64	FF64	0.160256%
	19200	B2	FFB2	0.160256%
	38400	D9	FFD9	0.160256%
	57600	E6	FFE6	0.160256%
	115200	F3	FFF3	0.160256%
	150000	F6	FFF6	0.000000%
	166666	F7	FFF7	0.000400%
	187500	F8	FFF8	0.000000%
	214285	F9	FFF9	0.000333%
	250000	FA	FFFA	0.000000%
	300000	FB	FFFB	0.000000%
	375000	FC	FFFC	0.000000%
	500000	FD	FFFD	0.000000%
	750000	FE	FFFE	0.000000%
	1500000	FF	FFFF	0.000000%
22118400	4800	70 (SMOD=0)	FEE0	0.000000%
	9600	70	FF70	0.000000%
	19200	B8	FFB8	0.000000%
	38400	DC	FFDC	0.000000%
	57600	E8	FFE8	0.000000%
	115200	F4	FFF4	0.000000%
	230400	FA	FFFA	0.000000%
	276480	FB	FFFB	0.000000%
	345600	FC	FFFC	0.000000%

Fsys Value	Baud Rate	TH1 Value (Hex)	RH3,RL3 Value (Hex)	Baudrate Deviation
	460800	FD	FFFD	0.000000%
	691200	FE	FFFE	0.000000%
	1382400	FF	FFFF	0.000000%
16600000	4800	28	FF28	0.067515%
	9600	94	FF94	0.067515%
	19200	CA	FFCA	0.067515%
	38400	E5	FFE5	0.067515%
	57600	EE	FFEE	0.067515%
	115200	F7	FFF7	0.067515%
16000000	4800	30	FF30	0.160256%
	9600	98	FF98	0.160256%
	19200	CC	FFCC	0.160256%
	38400	E6	FFE6	0.160256%
	57600	EF	FFEF	2.124183%
	115200	F7	FFF7	-3.549383%
	200000	FB	FFFB	0.000000%
	250000	FC	FFFC	0.000000%
	333333	FD	FFFD	0.000100%
	500000	FE	FFFE	0.000000%
	1000000	FF	FFFF	0.000000%
11059200	4800	70	FF70	0.000000%
	9600	B8	FFB8	0.000000%
	19200	DC	FFDC	0.000000%
	38400	EE	FFEE	0.000000%
	57600	F4	FFF4	0.000000%
	115200	FA	FFFA	0.000000%
	230400	FD	FFFD	0.000000%
	345600	FE	FFFE	0.000000%
	691200	FF	FFFF	0.000000%

6.10.3.3 Framing Error Detection

Framing error detection is provided for asynchronous modes. (Mode 1, 2, or 3.) The framing error occurs when a valid stop bit is not detected due to the bus noise or contention. The UART can detect a framing error and notify the software.

The framing error bit, FE, is located in SCON.7. This bit normally serves as SM0. While the framing error accessing enable bit SMOD0 (PCON.6) is set 1, it serves as FE flag. Actually, SM0 and FE locate in different registers.

The FE bit will be set 1 via hardware while a framing error occurs. FE can be checked in UART interrupt service routine if necessary. Note that SMOD0 should be 1 while reading or writing to FE. If FE is set, any following frames received without frame error will not clear the FE flag. The clearing has to be done via software.

6.10.3.4 Multiprocessor Communication

The ML51/ML54/ML56 Series multiprocessor communication feature lets a master device send a multiple frame serial message to a slave device in a multi-slave configuration. It does this without interrupting other slave devices that may be on the same serial line. This feature can be used only in UART Mode 2 or 3. User can enable this function by setting SM2 (SCON.5) as logic 1 so that when a byte of frame is received, the serial interrupt will be generated only if the 9th bit is 1. (For Mode 2, the 9th bit is the stop bit.) When the SM2 bit is 1, serial data frames that are received with the 9th bit as 0 do not generate an interrupt. In this case, the 9th bit simply separates the slave address from the serial data.

When the master device wants to transmit a block of data to one of several slaves on a serial line, it first sends out an address byte to identify the target slave. Note that in this case, an address byte differs from a data byte. In an address byte, the 9th bit is 1 and in a data byte, it is 0. The address byte interrupts all slaves so that each slave can examine the received byte and see if it is addressed by its own slave address. The addressed slave then clears its SM2 bit and prepares to receive incoming data bytes. The SM2 bits of slaves that were not addressed remain set, and they continue operating normally while ignoring the incoming data bytes.

Follow the steps below to configure multiprocessor communications:

1. Set all devices (masters and slaves) to UART Mode 2 or 3.

2. Write the SM2 bit of all the slave devices to 1.

3. The master device's transmission protocol is:

First byte: the address, identifying the target slave device, (9th bit = 1).

Next bytes: data, (9th bit = 0).

4. When the target slave receives the first byte, all of the slaves are interrupted because the 9th data bit is 1. The targeted slave compares the address byte to its own address and then clears its SM2 bit to receiving incoming data. The other slaves continue operating normally.

5. After all data bytes have been received, set SM2 back to 1 to wait for next address.

SM2 has no effect in Mode 0, and in Mode 1 can be used to check the validity of the stop bit. For Mode 1 reception, if SM2 is 1, the receiving interrupt will not be issue unless a valid stop bit is received.

6.10.3.5 Automatic Address Recognition

The automatic address recognition is a feature, which enhances the multiprocessor communication feature by allowing the UART to recognize certain addresses in the serial bit stream by using hardware to make the comparisons. This feature saves a great deal of software overhead by eliminating the need for the software to examine every serial address, which passes by the serial port. Only when the serial port recognizes its own address, the receiver sets RI bit to request an interrupt. The automatic address recognition feature is enabled when the multiprocessor communication feature is enabled, SM2 is set.

If desired, user may enable the automatic address recognition feature in Mode 1. In this configuration, the stop bit takes the place of the ninth data bit. RI is set only when the received command frame

address matches the device's address and is terminated by a valid stop bit.

Using the automatic address recognition feature allows a master to selectively communicate with one or more slaves by invoking the "Given" slave address or addresses. All of the slaves may be contacted by using the "Broadcast" address. Two SFR are used to define the slave address, SADDR, and the slave address mask, SADEN. SADEN is used to define which bits in the SADDR are to be used and which bits are "don't care". The SADEN mask can be logically ANDed with the SADDR to create the "Given" address, which the master will use for addressing each of the slaves. Use of the "Given" address allows multiple slaves to be recognized while excluding others.

The following examples will help to show the versatility of this scheme.

Example 1, slave 0:

SADDR = 11000000b

SADEN = 11111101b

Given = 110000X0b

Example 2, slave 1:

SADDR = 11000000b

SADEN = 11111110b

Given = 1100000Xb

In the above example SADDR is the same and the SADEN data is used to differentiate between the two slaves. Slave 0 requires 0 in bit 0 and it ignores bit 1. Slave 1 requires 0 in bit 1 and bit 0 is ignored. A unique address for Slave 0 would be 1100 0010 since slave 1 requires 0 in bit 1. A unique address for slave 1 would be 11000001b since 1 in bit 0 will exclude slave 0. Both slaves can be selected at the same time by an address, which has bit 0 = 0 (for slave 0) and bit 1 = 0 (for slave 1). Thus, both could be addressed with 11000000b as their "Broadcast" address.

In a more complex system the following could be used to select slaves 1 and 2 while excluding slave 0:

Example 1, slave 0:

SADDR = 11000000b

SADEN = 11111001b

Given = 11000XX0b

Example 2, slave 1:

SADDR = 11100000b

SADEN = 11111010b

Given = 11100X0Xb

Example 3, slave 2:

SADDR = 11000000b

SADEN = 11111100b

Given = 110000XXb

In the above example the differentiation among the 3 slaves is in the lower 3 address bits. Slave 0 requires that bit 0 = 0 and it can be uniquely addressed by 11100110b. Slave 1 requires that bit 1 = 0 and it can be uniquely addressed by 11100101b. Slave 2 requires that bit 2 = 0 and its unique address is 11100011b. To select Slaves 0 and 1 and exclude Slave 2 use address 11100100b, since it is necessary to make bit 2 = 1 to exclude slave 2.

The “Broadcast” address for each slave is created by taking the logical OR of SADDR and SADEN. Zeros in this result are treated as “don’t-cares”, e.g.:

SADDR = 01010110b

SADEN = 11111100b

Broadcast = 1111111Xb

The use of don’t-care bits provides flexibility in defining the Broadcast address, however in most applications, interpreting the “don’t-cares” as all ones, the broadcast address will be FFH.

On reset, SADDR and SADEN are initialized to 00H. This produces a “Given” address of all “don’t cares” as well as a “Broadcast” address of all XXXXXXXXb (all “don’t care” bits). This ensures that the serial port will reply to any address, and so that it is backwards compatible with the standard 80C51 Serial Port Register Description

6.10.4 Register Description

SCON – Serial Port Control

Register	SFR Address	Reset Value
SCON	98H, All pages, Bit addressable	0000_0000 b

7	6	5	4	3	2	1	0
SM0/FE	**SM1**	**SM2**	**REN**	**TB8**	**RB8**	**TI**	**RI**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**SM0/FE**	**Serial Port Mode Select** SMOD0 (PCON.6) = 0: See Table 6.10-1 Serial Port 0 Mode / baud rate Description for details. SMOD0 (PCON.6) = 1: SM0/FE bit is used as frame error (FE) status flag. It is cleared by software. 0 = Frame error (FE) did not occur. 1 = Frame error (FE) occurred and detected.
[6]	**SM1**	Check with bit 7 description.
[5]	**SM2**	**Multiprocessor Communication Mode Enable** The function of this bit is dependent on the serial port 0 mode. Mode 0: This bit select the baud rate between FSYS/12 and FSYS/2. 0 = The clock runs at FSYS/12 baud rate. It maintains standard 8051compatibility. 1 = The clock runs at FSYS/2 baud rate for faster serial communication. Mode 1: This bit checks valid stop bit. 0 = Reception is always valid no matter the logic level of stop bit. 1 = Reception is valid only when the received stop bit is logic 1 and the received data matches “Given” or “Broadcast” address. Mode 2 or 3: For multiprocessor communication. 0 = Reception is always valid no matter the logic level of the 9th bit. 1 = Reception is valid only when the received 9th bit is logic 1 and the received data matches “Given” or “Broadcast” address.
[4]	**REN**	**Receiving Enable** 0 = Serial port 0 reception Disabled. 1 = Serial port 0 reception Enabled in Mode 1,2, or 3. In Mode 0, reception is initiated by the condition REN = 1 and RI = 0.
[3]	**TB8**	**9th Transmitted Bit** This bit defines the state of the 9th transmission bit in serial port 0 Mode 2 or 3. It is not used in Mode 0 or 1.

Bit	Name	Description
[2]	**RB8**	**9th Received Bit** The bit identifies the logic level of the 9th received bit in serial port 0 Mode 2 or 3. In Mode 1, RB8 is the logic level of the received stop bit. SM2 bit as logic 1 has restriction for exception. RB8 is not used in Mode 0.
[1]	**TI**	**Transmission Interrupt Flag** This flag is set by hardware when a data frame has been transmitted by the serial port 0 after the 8th bit in Mode 0 or the last data bit in other modes. When the serial port 0 interrupt is enabled, setting this bit causes the CPU to execute the serial port 0 interrupt service routine. This bit should be cleared manually via software.
[0]	**RI**	**Receiving Interrupt Flag** This flag is set via hardware when a data frame has been received by the serial port 0 after the 8th bit in Mode 0 or after sampling the stop bit in Mode 1, 2, or 3. SM2 bit as logic 1 has restriction for exception. When the serial port 0 interrupt is enabled, setting this bit causes the CPU to execute to the serial port 0 interrupt service routine. This bit should be cleared manually via software.

S1CON – Serial Port 1 Control

Register	SFR Address	Reset Value
S1CON	F8H, All pages, Bit addressable	0000_0000 b

7	6	5	4	3	2	1	0
SM0_1/FE_1	**SM1_1**	**SM2_1**	**REN_1**	**TB8_1**	**RB8_1**	**TI_1**	**RI_1**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**SM0_1/FE_1**	**Serial Port 1 Mode Select** SMOD0_1 (T3CON.6) = 0: See Table 6.10-2 Serial Port 1 Mode / baud rate Description for details. SMOD0_1 (T3CON.6) = 1: SM0_1/FE_1 bit is used as frame error (FE) status flag. It is cleared by software. 0 = Frame error (FE) did not occur. 1 = Frame error (FE) occurred and detected.
[6]	**SM1_1**	Check with bit 7 description.
[5]	**SM2_1**	**Multiprocessor Communication Mode Enable** The function of this bit is dependent on the serial port 1 mode. Mode 0: No effect. Mode 1: This bit checks valid stop bit. 0 = Reception is always valid no matter the logic level of stop bit. 1 = Reception is valid only when the received stop bit is logic 1 and the received data matches “Given” or “Broadcast” address. Mode 2 or 3: For multiprocessor communication. 0 = Reception is always valid no matter the logic level of the 9th bit. 1 = Reception is valid only when the received 9th bit is logic 1 and the received data matches “Given” or “Broadcast” address.
[4]	**REN_1**	**Receiving Enable** 0 = Serial port 1 reception Disabled. 1 = Serial port 1 reception Enabled in Mode 1,2, or 3. In Mode 0, reception is initiated by the condition REN_1 = 1 and RI_1 = 0.
[3]	**TB8_1**	**9th Transmitted Bit** This bit defines the state of the 9th transmission bit in serial port 1 Mode 2 or 3. It is not used in Mode 0 or 1.

Bit	Name	Description
[2]	**RB8_1**	**9th Received Bit** The bit identifies the logic level of the 9th received bit in serial port 1 Mode 2 or 3. In Mode 1, RB8_1 is the logic level of the received stop bit. SM2_1 bit as logic 1 has restriction for exception. RB8_1 is not used in Mode 0.
[1]	**TI_1**	**Transmission Interrupt Flag** This flag is set by hardware when a data frame has been transmitted by the serial port 1 after the 8th bit in Mode 0 or the last data bit in other modes. When the serial port 1 interrupt is enabled, setting this bit causes the CPU to execute the serial port 1 interrupt service routine. This bit must be cleared manually via software.
[0]	**RI_1**	**Receiving Interrupt Flag** This flag is set via hardware when a data frame has been received by the serial port 1 after the 8th bit in Mode 0 or after sampling the stop bit in Mode 1, 2, or 3. SM2_1 bit as logic 1 has restriction for exception. When the serial port 1 interrupt is enabled, setting this bit causes the CPU to execute to the serial port 1 interrupt service routine. This bit must be cleared manually via software.

PCON – Power Control

Register	SFR Address	Reset Value
PCON	87H, All pages	POR: 0001_0000b Others: 000U _0000b

7	6	5	4	3	2	1	0
SMOD	**SMOD0**	**LPR**	**POF**	**GF1**	**GF0**	**PD**	**IDL**
R/W	R/W	RW	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**SMOD**	**Serial Port 0 Double Baud Rate Enable** Setting this bit doubles the serial port baud rate when UART0 is in Mode 2 or when Timer 1 overflow is used as the baud rate source of UART0 Mode 1 or 3. See Table 6.10-1 Serial Port 0 Mode / baud rate Description for details.
[6]	**SMOD0**	**Serial Port 0 Framing Error Flag Access Enable** 0 = SCON.7 accesses to SM0 bit. 1 = SCON.7 accesses to FE bit.

T3CON – Timer 3 Control

Register	SFR Address	Reset Value
T3CON	C4H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
SMOD_1	SMOD0_1	BRCK	TF3	TR3	T3PS[2:0]		
R/W	R/W	R/W	R/W	R/W	R/W		

Bit	Name	Description
[7]	SMOD_1	**Serial Port 1 Double Baud Rate Enable** Setting this bit doubles the serial port baud rate when UART1 is in Mode 2. See Table 6.10-2 Serial Port 1 Mode / baud rate Description for details.
[6]	SMOD0_1	**Serial Port 1 Framing Error Access Enable** 0 = S1CON.7 accesses to SM0_1 bit. 1 = S1CON.7 accesses to FE_1 bit.

SBUF – Serial Port 0 Data Buffer

Register	SFR Address	Reset Value
SBUF	99H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
SBUF[7:0]							
R/W							

Bit	Name	Description
[7:0]	**SBUF[7:0]**	**Serial Port 0 Data Buffer** This byte actually consists two separate registers. One is the receiving resister, and the other is the transmitting buffer. When data is moved to SBUF, it goes to the transmitting buffer and is shifted for serial transmission. When data is moved from SBUF, it comes from the receiving register. The transmission is initiated through giving data to SBUF.

SBUF1 – Serial Port 1 Data Buffer

Register	SFR Address	Reset Value
SBUF1	9AH, Page 0	0000 _0000 b

7	6	5	4	3	2	1	0
SBUF1[7:0]							
R/W							

Bit	Name	Description
[7:0]	**SBUF1[7:0]**	**Serial Port 1 Data Buffer** This byte actually consists two separate registers. One is the receiving resister, and the other is the transmitting buffer. When data is moved to SBUF1, it goes to the transmitting buffer and is shifted for serial transmission. When data is moved from SBUF1, it comes from the receiving register. The transmission is initiated through giving data to SBUF1.

IE – Interrupt Enable (Bit-addressable)

Register	SFR Address	Reset Value
IE	A8H, All pages, Bit addressable	0000 _0000 b

7	6	5	4	3	2	1	0
EA	EADC	EBOD	ES	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[4]	ES	**Enable Serial Port 0 Interrupt** 0 = Serial port 0 interrupt Disabled. 1 = Interrupt generated by TI (SCON.1) or RI (SCON.0) Enabled.

EIE1 – Extensive Interrupt Enable 1

Register	SFR Address	Reset Value
EIE1	9CH, Page 0	0000 _0000 b

7	6	5	4	3	2	1	0
-	EPWM123	EI2C1	ESPI1	EHFI	EWKT	ET3	ES1
-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[0]	ES1	**Enable Serial Port 1 Interrupt** 0 = Serial port 1 interrupt Disabled. 1 = Serial port 1Interrupt Enable. When interrupt generated TI_1 (S1CON.1) or RI_1 (S1CON.0) set 1.

SADDR0 – Slave 0 Address

Register	SFR Address	Reset Value
SADDR0	A9H, Page 0	0000 _0000 b

7	6	5	4	3	2	1	0
SADDR0[7:0]							
R/W							

Bit	Name	Description
[7:0]	**SADDR0[7:0]**	**Slave 0 Address** This byte specifies the microcontroller's own slave address for UATR0 multi-processor communication.

SADEN0 – Slave 0 Address Mask

Register	SFR Address	Reset Value
SADEN0	B9H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
SADEN0[7:0]							
R/W							

Bit	Name	Description
[7:0]	**SADEN0[7:0]**	**Slave 0 Address Mask** This byte is a mask byte of UART0 that contains “don’t-care” bits (defined by zeros) to form the device’s “Given” address. The don’t-care bits provide the flexibility to address one or more slaves at a time.

SADDR1 – Slave 1 Address

Register	SFR Address	Reset Value
SADDR1	BBH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
SADDR1[7:0]							
R/W							

Bit	Name	Description
[7:0]	**SADDR1[7:0]**	**Slave 1 Address** This byte specifies the microcontroller's own slave address for UART1 multi-processor communication.

SADEN1 – Slave 1 Address Mask

Register	SFR Address	Reset Value
SADEN1	BAH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
SADEN1[7:0]							
R/W							

Bit	Name	Description
[7:0]	**SADEN1[7:0]**	**Slave 1 Address Mask** This byte is a mask byte of UART1 that contains “don’t-care” bits (defined by zeros) to form the device’s “Given” address. The don’t-care bits provide the flexibility to address one or more slaves at a time.

AUXR1 – Auxiliary Register 1

Register	SFR Address	Reset Value
AUXR1	C9H, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	UART3PX	UART2PX	UART1PX	UART0PX
-	-	-	-	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:4]	-	Reserved
[3]	UART3PX	**Serial Port 3 RX (SMC1 DATA) /TX (SMC1 CLK) Pin Exchange** 0 = Assign UART3 RXD (SMC1 DATA) to multiple I/O pin RXD UART3 TXD (SMC CLK) to multiple I/O pin TXD 1 = Assign UART3 RXD (SMC1 DATA) to multiple I/O pin TXD UART3 TXD (SMC CLK) to multiple I/O pin RXD Note : that Pin direction is controlled by I/O type of relative pin. RXD/TXD will exchange immediately once setting or clearing this bit. User should take care of not exchanging pins during transmission or receiving. Or it may cause unpredictable situation and no warning alarms.
[2]	UART2PX	**Serial Port 2 RX (SMC0 DATA) /TX (SMC0 CLK) Pin Exchange** 0 = Assign UART2 RXD (SMC0 DATA) to multiple I/O pin RXD UART2 TXD (SMC CLK) to multiple I/O pin TXD 1 = Assign UART2 RXD (SMC0 DATA) to multiple I/O pin TXD UART2 TXD (SMC CLK) to multiple I/O pin RXD Note : that Pin direction is controlled by I/O type of relative pin. RXD/TXD will exchange immediately once setting or clearing this bit. User should take care of not exchanging pins during transmission or receiving. Or it may cause unpredictable situation and no warning alarms.
[1]	UART1PX	**Serial Port 1 RX/TX Pin Exchange** 0 = Assign UART1 RXD to multiple I/O pin RXD UART1 TXD to multiple I/O pin TXD 1 = Assign UART1 RXD to multiple I/O pin TXD UART1 TXD to multiple I/O pin RXD **Note:** that Pin direction is controlled by I/O type of relative pin. RXD/TXD will exchange immediately once setting or clearing this bit. User should take care of not exchanging pins during transmission or receiving. Or it may cause unpredictable situation and no warning alarms.

Bit	Name	Description
[0]	**UART0PX**	**Serial Port 0 RX/TX Pin Exchange** 0 = Assign UART0 RXD to multiple I/O pin RXD UART0 TXD to multiple I/O pin TXD 1 = Assign UART0 RXD to multiple I/O pin TXD UART0 TXD to multiple I/O pin RXD **Note:** that Pin direction is controlled by I/O type of relative pin. RXD/TXD will exchange immediately once setting or clearing this bit. User should take care of not exchanging pins during transmission or receiving. Or it may cause unpredictable situation and no warning alarms.

6.11 Smart Card Interface (SC)

6.11.1 Overview

The ML51/ML54/ML56 Series provides Smart Card Interface controller (SC controller) with asynchronous protocal based on ISO/IEC 7816-3 standard. Software controls GPIO pins as the smartcard reset function and card detection function. This controller also provides UART emulation for high precision baud rate communication.

6.11.2 Features

- ISO 7816-3 T = 0, T = 1 compliant
- Programmable transmission clock frequency
- Programmable extra guard time selection
- Supports auto inverse convention function
- Supports UART mode
  - Full duplex, asynchronous communications
  - Supports programmable baud rate generator for each channel
  - Programmable transmitting data delay time between the last stop bit leaving the TX-FIFO and the de-assertion by setting SCnEGT register
  - Programmable even, odd or no parity bit generation and detection
  - Programmable stop bit, 1 or 2 stop bit generation

6.11.3 Block Diagram

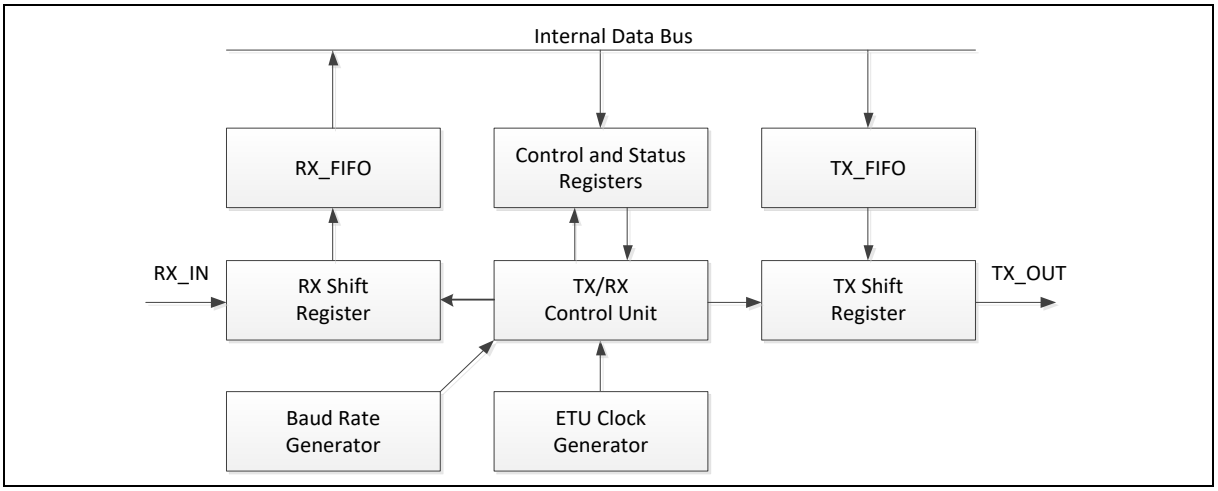


Figure 6.11-1 SC Controller Block Diagram

6.11.4 Operating Modes

6.11.4.1 Smart Card Mode

The Smart Card Interface controller supports activation, cold reset, warm reset and deactivation sequence by software control. The activation, cold reset, warm reset and deactivation and sequence are shown as follows.

SC Interface Connection

The SC interface connection is shown in Figure 15.3-1

1. SC_CLK / UART2_TXD : SC clock pin (output from MCU)
2. SC_DAT / UART2_RXD : SC data pin (bi-directional)
3. SC_RST: SC reset pin (output from MCU, firmware assigned GPIO)
4. SC_PWR: SC power pin (output from MCU, firmware assigned GPIO)
5. *: SC_PWR is used for power control function to turn ON/OFF the power for Smart Card. **Do not** use SC_PWR as the direct power supply for Smart Card.
6. SC_CD: SC card detect pin (input to MCU, detect card by a card insert mechanism)

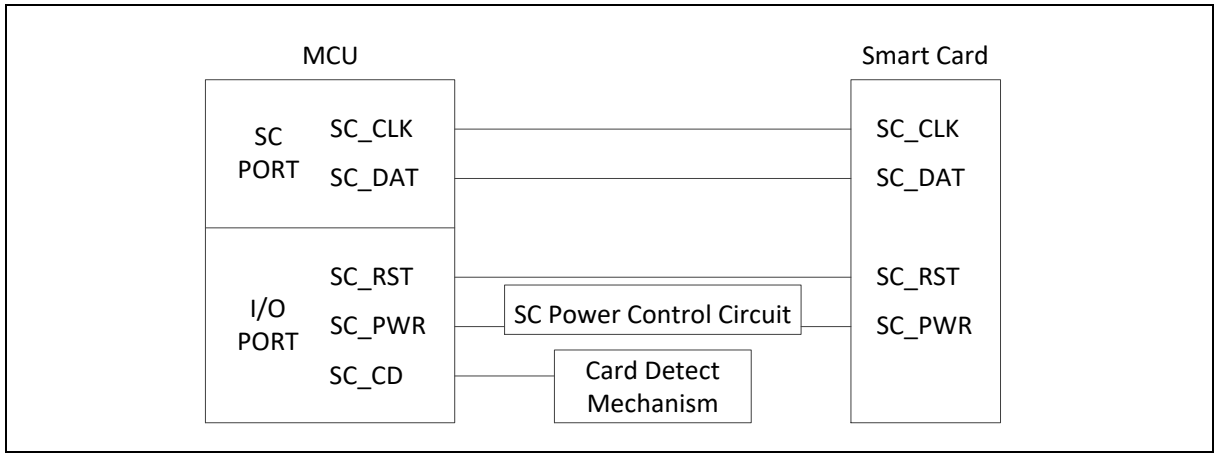
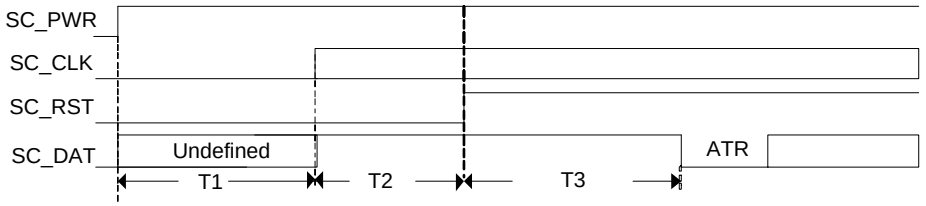


Figure 6.11-2 SC Interface Connection

Activation and Cold Reset

The activation and cold reset sequence is shown in Figure 15.3-2

1. Set SC_RST to low by software programming to ‘0’
2. Set SC_PWR at high level by software programming to ‘1’ before timing T1 and SC_DAT at high level (reception mode) by software programming to ‘1’ period of timing T1.
3. Enable SC_CLK clock by programming CLKKEEP (SCCR2[1]) to ‘1’ after timing T1.
4. De-assert SC_RST to high by software programming to ‘1’ after timing T2.
5. Smart Card host controller read the card ATR period of timing T3.



Time	Comment
T1	SC_PWR to SC_CLK Start
T2	SC_CLK Start to SC_RST Assert
T3	SC_RST Start to ATR Appear

Suggestion timing (Unit SC Clock)

T1	T2
83.5	491
133	537
165	569
165	42060

400 <= T3 <= 40000

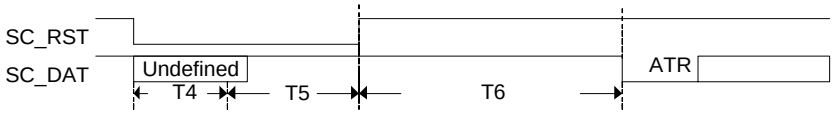
Note : The values are measured by chip I / O pin and the real value will depend on system design

Table 6.11-1 SC Activation and Cold Reset Sequence

Warm Reset

The warm reset sequence is showed in Figure 15.3-3

1. Set SC_RST to low by software programming to '0' before timing T4.
2. Set SC_DAT to high by software programming to '1' period of timing T4.
3. Set SC_RST to high by software programming to '1' after timing T5.
4. Smart Card host controller read the card ATR period of timing T6.



Time	Comment
T4	SC_RST to SC_DAT Reception Mode
T5	SC_DAT Reception Mode to SC_RST Assert
T6	SC_RST start to ATR Appear

Suggesting timing (Unit SC Clock)

T4	T5
81	483
129	531
161	563
161	42106

400 <= T6 <= 40000

Note: This value is measured by chip IO pin, the real value will depended on system design

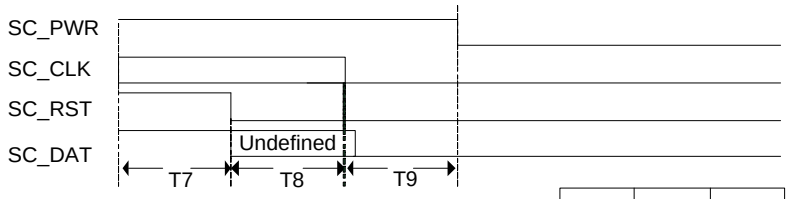
Table 6.11-2 SC Warm Reset Sequence

Deactivation

The deactivation sequence is showedin Figure 15.3-4

1. Set SC_RST to low by software programming to '0' period of timing T7.

2. Stop SC_CLK by programming CLKKEEP (SCCR2[1]) to ‘0’ period of timing T8.
3. Set SC_DAT to low by software programming to ‘0’ period of timing T8.
4. Deactivate SC_PWR by software programming to ‘0’ period of timing T9.



Time	Comment
T7	Deactivation Trigger to SC_RST Low
T8	SMC_ RST Low to Stop SC _CLK
T9	Stop SC_CLK to Stop SC_PWR

Unit SC Clock	T7	T8	T9
	97	83	87
	145	131	135
	177	163	167
	177	163	167

Table 6.11-3 SC Deactivation Sequence

6.11.4.2 UART Mode

When the UARTEN (SCCR2[0]) bit is set, the Smart Card Interface controller can also be used as basic UART function. The following is the program example for UART mode.

Programming example:

1. Set UARTEN (SCCR2[0]) bit to enter UART mode.
2. Fill “0” to CONSEL (SCnCR1[4]) and AUTOCEN (SCnCR1[3]) field. (In UART mode, those fields must be “0”)
3. Select the UART baud rate by setting ETURDIV[11:0] ({SCnETURD1[3:0]:SCnETURD0[7:0]}) fields. For example, if smartcard module clock is 12 MHz and target baud rate is 115200bps, ETURDIV should fill with (12000000 / 115200 – 1).
4. Select the data format include data length (by setting WLS (SCCR2[5:4]), parity format (by setting OPE (SCCR2[7]) and PBOFF (SCCR2[6])) and stop bit length (by setting NSB (SCnCR1[7])).
5. Write the SCnDR (SCnDR[7:0]) (TX) register or read the SCnDR (SCnDR[7:0]) (RX) register can perform UART function.

6.11.5 Smart Card Data Transfer

Basically, the smart card interface acts as a half-duplex asynchronous communication port and its data format is composed of ten consecutive bits which is showman.

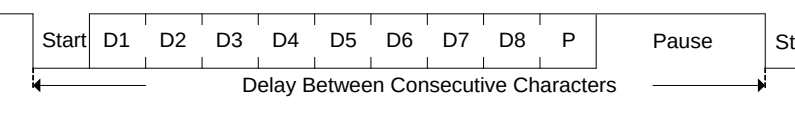


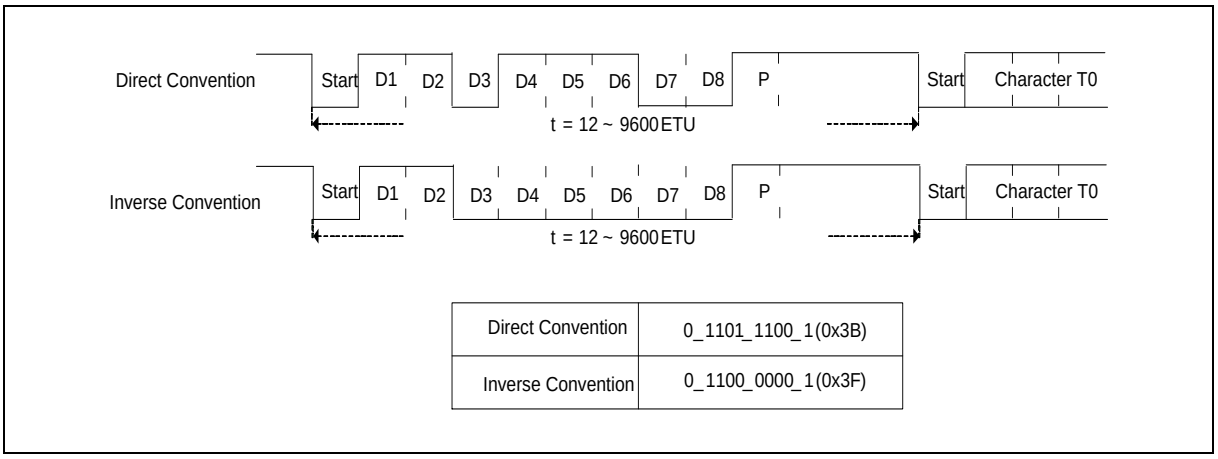
Figure 6.11-3 SC Data Character

6.11.5.1 Initial Character TS

According to 7816-3, the initial character TS has two possible patterns shown in Figure 6.11-4 Initial

Character TS. If the TS pattern is 1100_0000, it is inverse convention. When decoded by inverse convention, the conveyed byte is equal to 0x3F. If the TS pattern is 1101_1100, it is direct convention. When decoded by direct convention, the conveyed byte is equal to 0x3B.Software can set AUTOCEN (SCnCR1[3]) and then the operating convention will be decided by hardware. Software can also set the CONSEL (SCnCR1[4]) register (set to '0' or '1') to change the operating convention after SC received TS of answer to request (ATR).

If auto convention function is enabled by setting AUTOCEN (SCnCR1[3]) register, the setting step must be done before Answer to Request state and the first data must be 0x3B or 0x3F. After hardware received first data and stored it at buffer, the hardware will decided the convention and change the CONSEL (SCnCR1[4]) register automatically. If the first data is neither 0x3B nor 0x3F, the hardware will generate an interrupt (if ACERRIEN (ScnIE[4] = '1') to CPU.



Direct Convention	0_1101_1100_1(0x3B)
Inverse Convention	0_1100_0000_1(0x3F)

Figure 6.11-4 Initial Character TS

6.11.5.2 Error Signal and Character Repetition

According to ISO7816-3 T=0 mode description, as shown in Figure 6.11-5,if the receiver receives a wrong parity bit, it will pull the SC_DAT to low by 1.5 bit period to inform the transmitter parity error. Then the transmitter will retransmit the character. The SC interface controller supports hardware error detection function(SC0TSR[4]) in receiver, SC controller will generate a transfer error interrupt(if TERRIEN(ScnIE[2] = '1') to CPU.

When in T=1 mode, the receiver will not pull the SC_DAT to low by 1.5 bit period to inform the transmitter parity error.

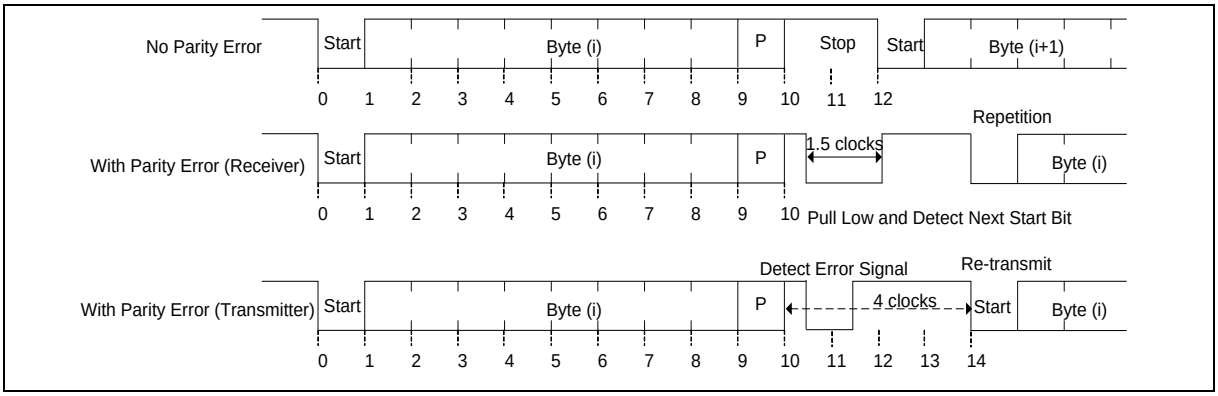


Figure 6.11-5 SC Error Signal

6.11.5.3 Block Guard Time and Extra Guard Time

Block guard time means the minimum bit length between the leading edges of two consecutive characters between different transfer directions. This field indicates the counter for the bit length of block

guard time.

According to ISO7816-3, in T = 0 mode, software must fill T bit = 0 (real block guard time = 16.5) to this field; in T = 1 mode, software must fill T bit = 1 (real block guard time = 22.5) to it.

In transmit direction, the smart card sends data to smart card host controller, first. After the period is greater than (16.5 or 22.5, by T bit setting), the smart card host controller begin to send the data.

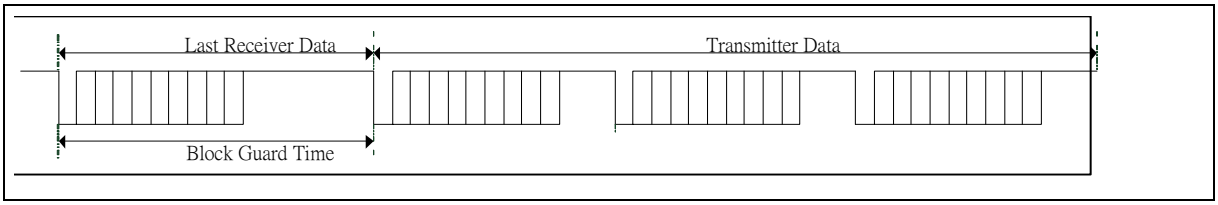


Figure 6.11-6 Transmit Direction Block Guard Time Operation

In receive direction, the smart card host controller sends data to smart card, first. If the smart card sends data to smart card host controller at the time which is less than (16.5 or 22.5, by T bit setting),the block guard time interrupt BGTIF (ScnIS[3]) is generated when RXBGTEN (SCnCR1[5]) and BGTIEN (ScnIE[3]) are enabled.

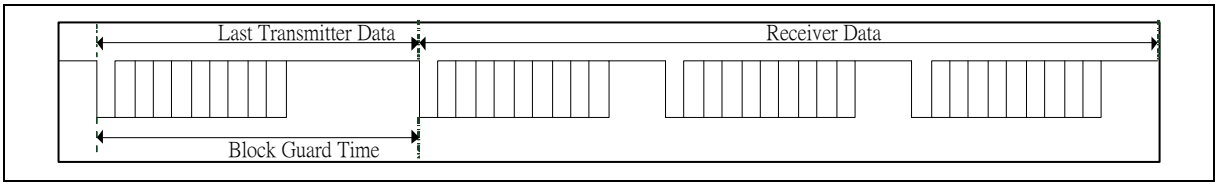


Figure 6.11-7 Receive Direction Block Guard Time Operation

Extra Guard Time is EGT (SCnEGT[7:0]),it only affects the data transmitted by smart card interface, the format is shown as Figure 6.11-8.

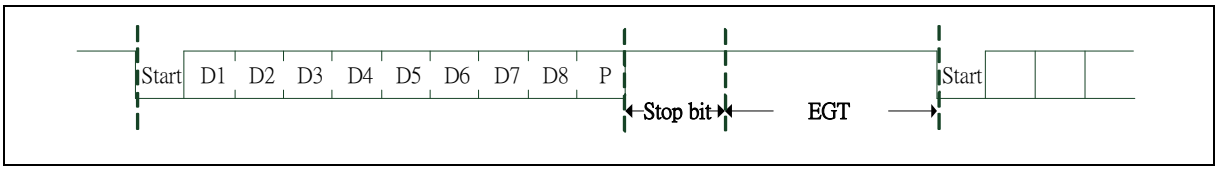


Figure 6.11-8 Extra Guard Time Operation

6.11.6 Register Description

SCnCR0 – SC Control Register 0

Register	SFR Address	Reset Value
SC0CR0	D6H, Page 0	0000_0000 b
SC1CR0	E6H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
NSB	**T**	**RXBGTEN**	**CONSEL**	**AUTOCEN**	**TXOFF**	**RXOFF**	**SCEN**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**NSB**	**Stop Bit Length** This field indicates the length of stop bit. 0 = The stop bit length is 2 ETU. 1= The stop bit length is 1 ETU. **Note:** The default stop bit length is 2. SC and UART adopt NSB to program the stop bit length.
[6]	**T**	**T Mode** 0 = T0 (ISO7816-3 T = 0 mode). 1 = T1 (ISO7816-3 T = 1 mode). The T mode controls the BGT (Block Guard Time). Block guard time means the minimum bit length between the leading edges of two consecutive characters between different transfer directions. This field indicates the counter for the bit length of block guard time. According to ISO7816-3, in T = 0 mode, the software must clear T bit to 0 for real block guard time = 16.5. In T = 1 mode, the software must set T bit to 1 for real block guard time = 22.5. **Note:** In T = 0 mode, the receiver will generate the parity flag on PEF(SC0TSR[4]), if parity error is detected and also drive the parity error signal to transceiver. In T = 1 mode, the receiver will generate the parity flag on PEF(SC0TSR[4]), if parity error detected, but doesn't drive the parity error signal to transceiver. **Note:** The description please see section 6.11.5.2 Error Signal and Character Repetition
[5]	**RXBGTEN**	**Receiver Block Guard Time Function Enable Bit** 0 = Receiver block guard time function Disabled. 1 = Receiver block guard time function Enabled.
[4]	**CONSEL**	**Convention Selection** 0 = Direct convention. 1 = Inverse convention. **Note 1:** This bit is auto clear to "0", if AUTOCEN(SCnCR0[3]) is writing "1" **Note 2:** If AUTOCEN(SCnCR0[3]) is enabled, hardware will decide the convention and change the CONSEL (SCnCR0[4]) bits automatically after SCEN (SCnCR0[0]) ="1".

Bit	Name	Description
[3]	**AUTOCEN**	**Auto Convention Enable Bit** 0 = Auto-convention Disabled. 1 = Auto-convention Enabled. When hardware receives TS in answer to reset state and the TS is direct convention, CONSEL(SCnCR0[4]) will be set to 0 automatically, otherwise if the TS is inverse convention, and CONSEL (SCnCR0[4]) will be set to 1. **Note:** If software enables auto convention function, the setting step must be done before Answer to Reset state and the first data must be 0x3B or 0x3F. After hardware received first data and stored it at buffer, hardware will decided the convention and change the CONSEL (SCnCR0[4]) bits automatically.
[2]	**TXOFF**	**TX Transition Disable Bit** 0 = The transceiver Enabled. 1 = The transceiver Disabled.
[1]	**RXOFF**	**RX Transition Disable Bit** 0 = The receiver Enabled. 1 = The receiver Disabled. **Note:** If AUTOCEN (SCnCR0[3])is enabled, these fields must be ignored.
[0]	**SCEN**	**SC Engine Enable Bit** Set this bit to 1 to enable SC operation. If this bit is cleared, SC will force all transition to IDLE state **Note:** SCEN must be set to 1 before filling in other registers, or smart card will not work properly.

SCnCR1 – SC Control Register

Register	SFR Address	Reset Value
SC0CR1	D7H, Page 0	0000_0000 b
SC1CR1	E7H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
OPE	PBOFF	WLS[1:0]		TXDMAEN	RXDMAEN	CLKKEEP	UARTEN
R/W	R/W	R/W		R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	OPE	**Odd Parity Enable Bit** 0 = Even number of logic 1's are transmitted or check the data word and parity bits in receiving mode. 1 = Odd number of logic 1's are transmitted or check the data word and parity bits in receiving mode. **Note:** This bit has effect only when PBOFF bit is '0'.
[6]	PBOFF	**Parity Bit Disable Control** 0 = Parity bit is generated or checked between the "last data word bit" and "stop bit" of the serial data. 1 = Parity bit is not generated (transmitting data) or checked (receiving data) during transfer. **Note:** In smart card mode, this field must be '0' (default setting is with parity bit)
[5:4]	WLS[1:0]	**Word Length Selection** 00 = Word length is 8 bits. 01 = Word length is 7 bits. 10 = Word length is 6 bits. 11 = Word length is 5 bits. **Note:** In smart card mode, this WLS must be '00'
[3]	TXDMAEN	**SC/UART TX DMA Enable** This bit enables the SC/UART TX operating by through PDMA transfer, TX data needs to be ready in XRAM before SC/UART TX starting. 0 = SPI TX DMA Disabled 1 = SPI TX DMA Enabled
[2]	RXDMAEN	**SC/UART RX DMA Enable** This bit enables the SC/UART RX operating by through PDMA transfer, RX data are saved in XRAM after SC/UART RX operation. 0 = SC/UART RX DMA Disabled 1 = SC/UART RX DMA Enabled
[1]	CLKKEEP	**SC Clock Enable Bit** 0 = SC clock generation Disabled. 1 = SC clock always keeps free running.

Bit	Name	Description
[0]	**UARTEN**	**UART Mode Enable Bit** 0 = Smart Card mode. 1 = UART mode. **Note 1:**When operating in UART mode, user must set CONSEL (SCnCR0[4]) = 0 and AUTOCEN(SCnCR0[3]) = 0. **Note 2:**When operating in Smart Card mode, user must set UARTEN(SCnCR1 [0]) = 0. **Note 3:**When UART is enabled, hardware will generate a reset to reset FIFO and internal state machine.

SCnDR – SC Data Register

Register	SFR Address	Reset Value
SC0DR	D9H, Page 0	0000_0000 b
SC1DR	D9H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
SCnDR[7:0]							
R/W							

Bit	Name	Description
[7:0]	**SCnDR[7:0]**	**SC / UART Buffer Data** This byte is used for transmitting or receiving data on SC / UART bus. A write of this byte is a write to the shift register. A read of this byte is actually a read of the read data buffer. **Note:** If SCEN(SCnCR0[0]) is not enabled, SCnDR cannot be programmed.

SCnEGT – SC0~1 Extra Guard Time Register

Register	SFR Address	Reset Value
SC0EGT	DAH, Page 0	0000_0000 b
SC1EGT	DAH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
SCnEGT[7:0]							
R/W							

Bit	Name	Description
[7:0]	**SCnEGT[7:0]**	**SC Extra Guard Time** This field indicates the extra guard timer value. **Note:** The counter is ETU base .

SCnETURD0 – SCn ETU Rate Divider Register

Register	SFR Address	Reset Value
SC0ETURD0	DBH, Page 0	0111_0011 b
SC1ETURD0	DBH, Page 2	0111_0011 b

7	6	5	4	3	2	1	0
ETURDIV[7:0]							
R/W							

Bit	Name	Description
[7:0]	**ETURDIV[7:0]**	**LSB Bits of ETU Rate Divider** The field indicates the LSB of clock rate divider. The real ETU is ETURDIV[11:0] + 1. **Note 1:** ETURDIV[11:0] must be greater than 0x004. **Note 2:** SCnETURD0 has to program first, then SCnETUDR2.

SCnETURD1 –SC ETU Rate Divider Register

Register	SFR Address	Reset Value
SC0ETURD1	DCH, Page 0	0011_0001 b
SC1ETURD1	DCH, Page 2	0011_0001 b

7	6	5	4	3	2	1	0
-	SCDIV[2:0]			ETURDIV[11:8]			
-	R/W			R/W			

Bit	Name	Description
[7]	-	Reserved
[6:4]	SCDIV[2:0]	**SC Clock Divider** 000 = F_{SC} is F_{SYS}/1. 001 = F_{SC} is F_{SYS}/2. 010 = F_{SC} is F_{SYS}/4. 011 = F_{SC} is F_{SYS}/8. (By default.) 100 = F_{SC} is F_{SYS}/16. 101 = F_{SC} is F_{SYS}/16. 110 = F_{SC} is F_{SYS}/16. 111 = F_{SC} is F_{SYS}/16. **Note:** that the F_{SC} clock should be 1Mhz ~ 5Mhz for ISO/IEC 7816-3 standard
[3:0]	ETURDIV[11:8]	**MSB Bits of ETU Rate Divider** The field indicates the MSB of clock rate divider. The real ETU is ETURDIV[11:0] + 1. **Note 1:** ETURDIV[11:0] must be greater than 0x004. **Note 2:** SCnETURD0 has to program first, then SCnETUDR1 .

ScnIE – SC Interrupt Enable Control Register

Register	SFR Address	Reset Value
SC0IE	DDH, Page 0	0000_0000 b
SC1IE	DDH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	ACERRIEN	BGTIEN	TERRIEN	TBEIEN	RDAIEN
-	-	-	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:5]	-	Reserved
[4]	ACERRIEN	**Auto Convention Error Interrupt Enable Bit** This field is used to enable auto-convention error interrupt. 0 = Auto-convention error interrupt Disabled. 1 = Auto-convention error interrupt Enabled.
[3]	BGTIEN	**Block Guard Time Interrupt Enable Bit** This field is used to enable block guard time interrupt. 0 = Block guard time interrupt Disabled. 1 = Block guard time interrupt Enabled.
[2]	TERRIEN	**Transfer Error Interrupt Enable Bit** This field is used to enable transfer error interrupt. The transfer error states is at SC0TSR register which includes receiver break error BEF(SC0TSR[6]), frame error FEF(SC0TSR[5]), parity error PEF(SC0TSR[4]), receiver buffer overflow error RXOV(SC0TSR[0]) and transmit buffer overflow error TXOV(SC0TSR[2]). 0 = Transfer error interrupt Disabled. 1 = Transfer error interrupt Enabled.
[1]	TBEIEN	**Transmit Buffer Empty Interrupt Enable Bit** This field is used to enable transmit buffer empty interrupt. 0 = Transmit buffer empty interrupt Disabled. 1 = Transmit buffer empty interrupt Enabled.
[0]	RDAIEN	**Receive Data Reach Interrupt Enable Bit** This field is used to enable received data interrupt. 0 = Receive data interrupt Disabled. 1 = Receive data interrupt Enabled.

SCnIS – SC Interrupt Status Register

Register	SFR Address	Reset Value
SC0IS	DEH, Page 0	0000_0010 b
SC1IS	DEH, Page 2	0000_0010 b

7	6	5	4	3	2	1	0
-		Tx_Er	ACERRIF	BGTIF	TERRIF	TBEIF	RDAIF
-		R/W	R/W	R/W	R	R	R

Bit	Name	Description
[7:6]	-	Reserved.
[5]	Tx_Er	TX transmit error flag
[4]	ACERRIF	**Auto Convention Error Interrupt Status Flag (Read Only)** This field indicates auto convention sequence error. If the received TS at ATR state is neither 0x3B nor 0x3F, this bit will be set. **Note:** This bit is read only, but it can be cleared by writing “0” to it.
[3]	BGTIF	**Block Guard Time Interrupt Status Flag (Read Only)** This field is used for block guard time interrupt status flag. **Note 1:** This bit is valid when RXBGTEN (SCnCR0[5]) is enabled. **Note 2:** This bit is read only, but it can be cleared by writing “0” to it.
[2]	TERRIF	**Transfer Error Interrupt Status Flag (Read Only)** This field is used for transfer error interrupt status flag. The transfer error states is at SC0TSR register which includes receiver break error BEF(SC0TSR[6]), frame error FEF(SC0TSR[5]), parity error PEF(SC0TSR[4]) and receiver buffer overflow error RXOV(SC0TSR[0]) and transmit buffer overflow error TXOV(SC0TSR[2]). **Note:** This field is the status flag of BEF(SC0TSR[6]), FEF(SC0TSR[5]), PEF(SC0TSR[4]), RXOV(SC0TSR[0]) and TXOV(SC0TSR[2]). So, if software wants to clear this bit, software must write “0” to each field.
[1]	TBEIF	**Transmit Buffer Empty Interrupt Status Flag (Read Only)** This field is used for transmit buffer empty interrupt status flag. **Note:** This field is the status flag of transmit buffer empty state. If software wants to clear this bit, software must write data to DAT(SCnDR[7:0]) buffer and then this bit will be cleared automatically.
[0]	RDAIF	**Receive Data Reach Interrupt Status Flag (Read Only)** This field is used for received data interrupt status flag. **Note:** This field is the status flag of received data. If software reads data from SC_DAT pin, this bit will be cleared automatically.

SCnTSR – SC Transfer Status Register

Register	SFR Address	Reset Value
SC0TSR	DFH, Page 0	0000_1010 b
SC1TSR	DFH, Page 2	0000_1010 b

7	6	5	4	3	2	1	0
ACT	BEF	FEF	PEF	TXEMPTY	TXOV	RXEMPTY	RXOV
R	R/W	R/W	R/W	R	R/W	R	R/W

Bit	Name	Description
[7]	ACT	**Transmit /Receive in Active Status Flag (Read Only)** 0 = This bit is cleared automatically when TX/RX transfer is finished 1 = This bit is set by hardware when TX/RX transfer is in active.
[6]	BEF	**Receiver Break Error Status Flag (Read Only)** This bit is set to logic 1 whenever the received data input (RX) held in the “spacing state” (logic 0) is longer than a full word transmission time (that is, the total time of “start bit” + data bits + parity + stop bits). . **Note:** This bit is read only, but it can be cleared by writing 0 to it.
[5]	FET	**Receiver Frame Error Status Flag (Read Only)** This bit is set to logic 1 whenever the received character does not have a valid “stop bit” (that is, the stop bit following the last data bit or parity bit is detected as logic 0). **Note:** This bit is read only, but it can be cleared by writing 0 to it.
[4]	PEF	**Receiver Parity Error Status Flag (Read Only)** This bit is set to logic 1 whenever the received character does not have a valid “parity bit”. **Note:** This bit is read only, but it can be cleared by writing 0 to it.
[3]	TXEMPTY	**Transmit Buffer Empty Status Flag (Read Only)** This bit indicates TX buffer empty or not. **Note:** When TX buffer has been transferred to Transmitter Shift Register, hardware sets this bit high. It will be cleared when writing data into DAT(SCnDR[7:0]) (TX buffer not empty).
[2]	TXOV	**TX Overflow Error Interrupt Status Flag (Read Only)** If TX buffer is full, an additional write to DAT(SCnDR[7:0]) will cause this bit be set to “1” by hardware. **Note:** This bit is read only, but it can be cleared by writing 0 to it.
[1]	RXEMPTY	**Receiver Buffer Empty Status Flag(Read Only)** This bit indicates RX buffer empty or not. **Note:** When Rx buffer has been read by CPU, hardware sets this bit high. It will be cleared when SC receives any new data.

Bit	Name	Description
[0]	**RXOV**	**RX Overflow Error Status Flag (Read Only)** This bit is set when RX buffer overflow. **Note:** This bit is read only, but it can be cleared by writing 0 to it.

6.12 Serial Peripheral Interface (SPI)

6.12.1 Overview

The ML51/ML54/ML56 Series provides two Serial Peripheral Interface (SPI) block to support high-speed serial communication. SPI is a full-duplex, high-speed, synchronous communication bus between microcontrollers or other peripheral devices such as serial EEPROM, LCD driver, or D/A converter. It provides either Master or Slave mode, high-speed rate up to $F_{SYS}/4$, transfer complete and write collision flag. For a multi-master system, SPI supports Master Mode Fault to protect a multi-master conflict.

6.12.2 Features

- 2 sets of SPI devices
- Supports Master or Slave mode operation
- Supports MSB first or LSB first transfer sequence
- Slave mode up to 12 Mhz

6.12.3 Block Diagram

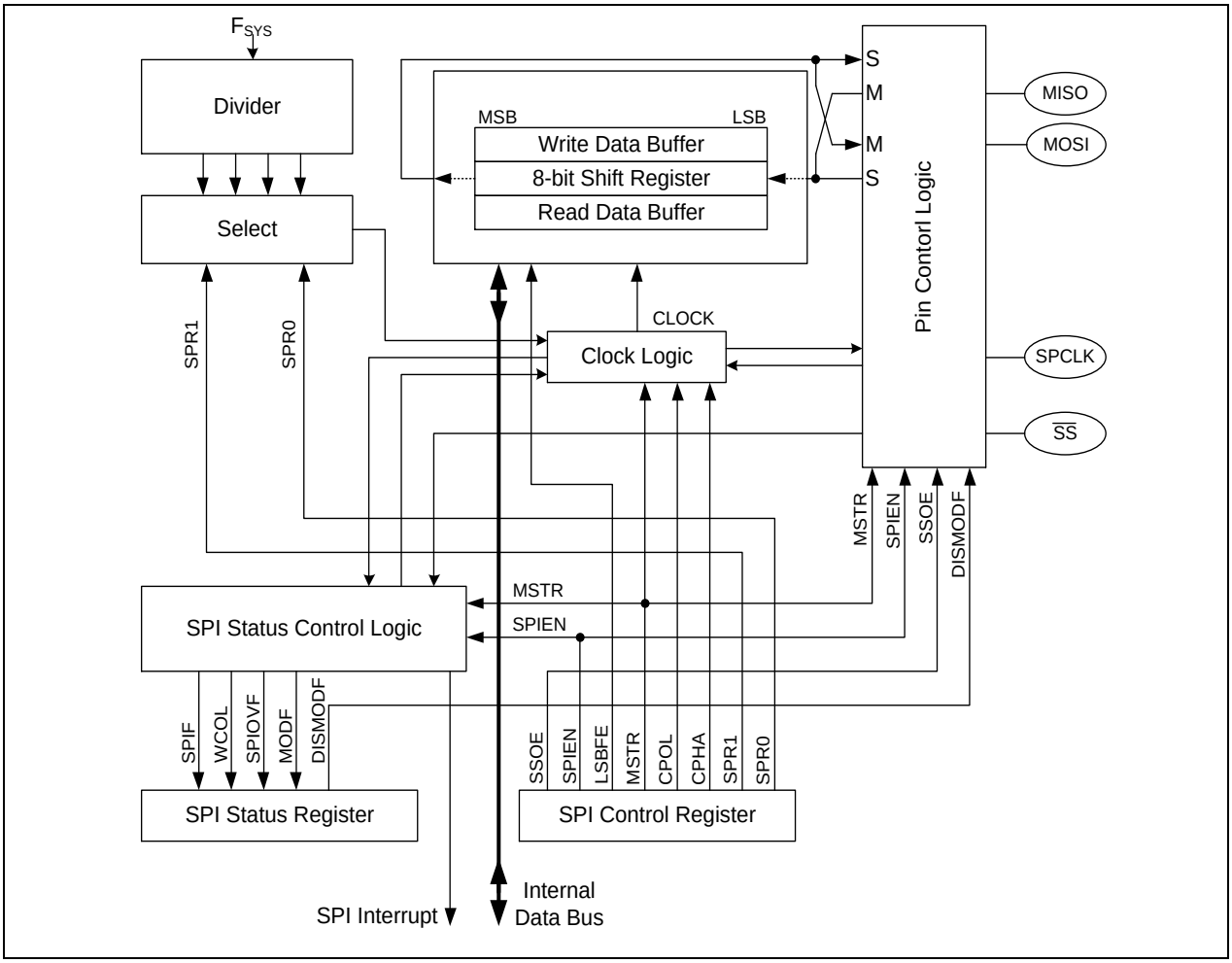


Figure 6.12-1 SPI Block Diagram

6.12.4 Functional Description

SPI block diagram provides an overview of SPI architecture in this device. The main blocks of SPI are the SPI control register logic, SPI status logic, clock rate control logic, and pin control logic. For a serial data transfer or receiving, The SPI block exists a shift register and a read data buffer. It is double buffered in the receiving and transmit directions. Transmit data cannot be written to the shifter until the previous transfer is complete. Receiving logic consists of parallel read data buffer so the shift register is free to accept a second data, as the first received data will be transferred to the read data buffer.

The 2 set, 16 pins of SPI interface and 4 pins of SPI0 interface which are Master-In/Slave-Out (MISO), Master-Out/Slave-In (MOSI), Shift Clock (SPCLK), and Slave Select ($\overline{SS}$). The MOSI pin is used to transfer a 8-bit data in series from the Master to the Slave. Therefore, MOSI is an output pin for Master device and an input for Slave. Respectively, the MISO is used to receive a serial data from the Slave to the Master.

The SPCLK pin is the clock output in Master mode, but is the clock input in Slave mode. The shift clock is used to synchronize the data movement both in and out of the devices through their MOSI and MISO pins. The shift clock is driven by the Master mode device for eight clock cycles. Eight clocks exchange one byte data on the serial lines. For the shift clock is always produced out of the Master device, the system should never exist more than one device in Master mode for avoiding device conflict.

Each Slave peripheral is selected by one Slave Select pin ($\overline{SS}$). The signal should stay low for any Slave access. When $\overline{SS}$ is driven high, the Slave device will be inactivated. If the system is multi-slave, there should be only one Slave device selected at the same time. In the Master mode MCU, the $\overline{SS}$ pin does not function and it can be configured as a general purpose I/O. However, $\overline{SS}$ can be used as Master Mode Fault detection (see Section 6.12.4.6"Mode Fault Detection") via software setting if multi-master environment exists. The ML51/ML54/ML56 Series also provides auto-activating function to toggle $\overline{SS}$ between each byte-transfer.

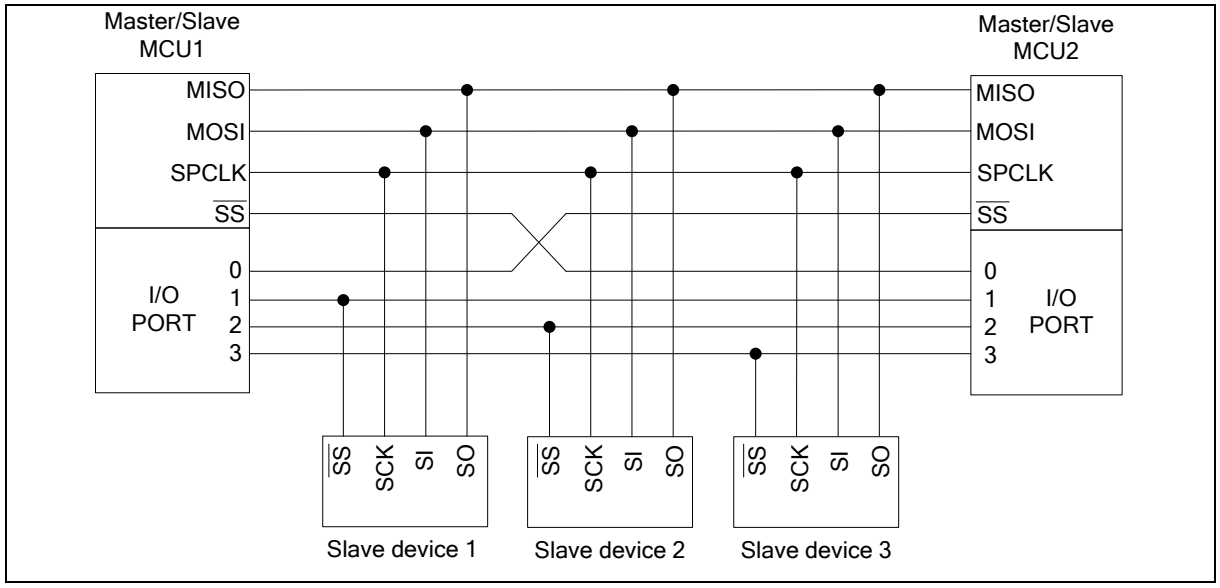


Figure 6.12-2 SPI Multi-Master, Multi-Slave Interconnection

Figure 6.12-2 shows a typical interconnection of SPI devices. The bus generally connects devices together through three signal wires, MOSI to MOSI, MISO to MISO, and SPCLK to SPCLK. The Master devices select the individual Slave devices by using four pins of a parallel port to control the four $\overline{SS}$ pins. MCU1 and MCU2 play either Master or Slave mode. The $\overline{SS}$ should be configured as Master Mode Fault detection to avoid multi-master conflict.

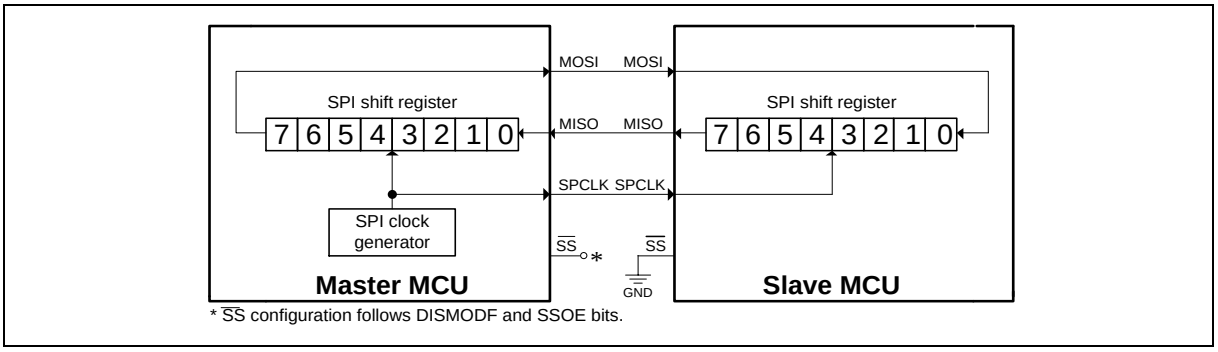


Figure 6.12-3 SPI Single-Master / Single-Slave Interconnection

Figure 6.12-3 SPI Single-Master / Single-Slave Interconnection shows the simplest SPI system interconnection, single-master and signal-slave. During a transfer, the Master shifts data out to the Slave via MOSI line. While simultaneously, the Master shifts data in from the Slave via MISO line. The two shift registers in the Master MCU and the Slave MCU can be considered as one 16-bit circular shift register. Therefore, while a transfer data pushed from Master into Slave, the data in Slave will also be pulled in Master device respectively. The transfer effectively exchanges the data, which was in the SPI shift registers of the two MCUs.

LSB/MSB First

By default, SPI data is transferred MSB first. If the LSBFE (SPInCR.5) is set, SPI data shifts LSB first. This bit does not affect the position of the MSB and LSB in the data register. Note that all the following description and figures are under the condition of LSBFE logic 0. MSB is transmitted and received first.

6.12.4.2 Operating Modes

Master Mode

The SPI can operate in Master mode while MSTR (SPInCR.4) is set as 1. Only one Master SPI device can initiate transmissions. A transmission always begins by Master through writing to SPInDR. The byte written to SPInDR begins shifting out on MOSI pin under the control of SPCLK. Simultaneously, another byte shifts in from the Slave on the MISO pin. After 8-bit data transfer complete, SPIF (SPInSR.7) will automatically set via hardware to indicate one byte data transfer complete. At the same time, the data received from the Slave is also transferred in SPInDR. User can clear SPIF and read data out of SPInDR.

Slave Mode

When MSTR is 0, the SPI operates in Slave mode. The SPCLK pin becomes input and it will be clocked by another Master SPI device. The $\overline{SS}$ pin also becomes input. The Master device cannot exchange data with the Slave device until the $\overline{SS}$ pin of the Slave device is externally pulled low. Before data transmissions occurs, the $\overline{SS}$ of the Slave device should be pulled and remain low until the transmission is complete. If $\overline{SS}$ goes high, the SPI is forced into idle state. If the $\overline{SS}$ is forced to high at the middle of transmission, the transmission will be aborted and the rest bits of the receiving shifter buffer will be high and goes into idle state.

In Slave mode, data flows from the Master to the Slave on MOSI pin and flows from the Slave to the Master on MISO pin. The data enters the shift register under the control of the SPCLK from the Master device. After one byte is received in the shift register, it is immediately moved into the read data buffer and the SPIF bit is set. A read of the SPInDR is actually a read of the read data buffer. To prevent an overrun and the loss of the byte that caused by the overrun, the Slave should read SPInDR out and the first SPIF should be cleared before a second transfer of data from the Master device comes in the read data buffer.

6.12.4.3 SPI Master Clock Configuration

These four bits select four grades of SPI clock divider. The clock rates below are illustrated under FSYS = 24 MHz condition.SPR[3:0] are valid only under Master mode (MSTR = 1). If under Slave mode, the clock will automatically synchronize with the external clock on SPICLK pin from Master device up to FSYS/4 communication speed. Following is the SPI clock rate define table

SPR3	SPR2	SPR1	SPR0	Divider of system	SPI Clock Rate
0	0	0	0	2	12M bit/s
0	0	0	1	4	6M bit/s
0	0	1	0	8	3M bit/s
0	0	1	1	16	1.5M bit/s
0	1	0	0	32	750k bit/s
0	1	0	1	64	375k bit/s
0	1	1	0	128	187k bit/s
0	1	1	1	256	93.7k bit/s
1	0	0	0	3	8M bit/s
1	0	0	1	6	4M bit/s
1	0	1	0	12	2M bit/s
1	0	1	1	24	1M bit/s
1	1	0	0	48	500k bit/s
1	1	0	1	96	250k bit/s
1	1	1	0	192	125k bit/s
1	1	1	1	384	62.5k bit/s

Table 6.12-1 SPI Master Clock Rate Define Table

SPIS[1:0] (SPI0CR1 [1:0]) provides a configurable suspend interval, 0.5 ~ 2.5 SPI clock periods, between two successive transaction words in Master mode. The definition of the suspend interval is the interval between the last clock edge of the preceding transaction word and the first clock edge of the following transaction word. Following shows the suspend interval select define.

CPHA	SPIS1	SPIS0	SPI clock
0	0	0	0.0
0	0	1	0.5
0	1	0	1.5
0	1	1	2.0
1	0	0	0.0
1	0	1	1.0
1	1	0	2.0
1	1	1	2.5

Table 6.12-2 SPI Clock Suspend Interval Select

6.12.4.4 Clock Formats and Data Transfer

To accommodate a wide variety of synchronous serial peripherals, the SPI has a clock polarity bit CPOL (SPInCR.3) and a clock phase bit CPHA (SPInCR.2). Figure 6.12-4 SPI Clock Formats shows that CPOL and CPHA compose four different clock formats. The CPOL bit denotes the SPCLK line level in its idle state. The CPHA bit defines the edge on which the MOSI and MISO lines are sampled. The CPOL and CPHA should be identical for the Master and Slave devices on the same system. To Communicate in different data formats with one another will result undetermined result.

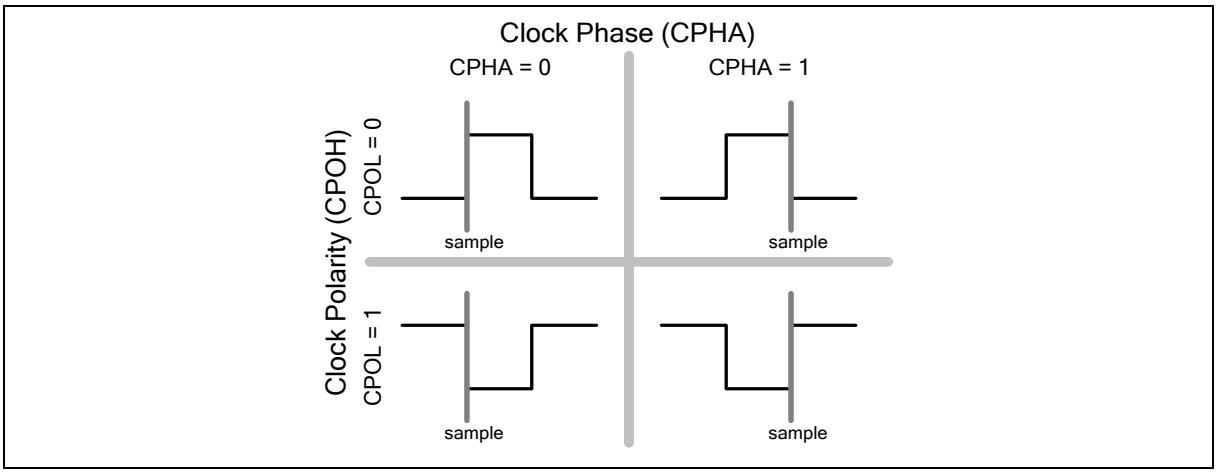


Figure 6.12-4 SPI Clock Formats

In SPI, a Master device always initiates the transfer. If SPI is selected as Master mode (MSTR = 1) and enabled (SPIEN = 1), writing to the SPI data register (SPInDR) by the Master device starts the SPI clock and data transfer. After shifting one byte out and receiving one byte in, the SPI clock stops and SPIF (SPInSR.7) is set in both Master and Slave. If SPI interrupt enable bit is set 1 and global interrupt is enabled (EA = 1), the interrupt service routine (ISR) of SPI will be executed.

Concerning the Slave mode, the $\overline{SS}$ signal needs to be taken care. As shown in Figure 6.12-4 SPI Clock Formats, when CPHA = 0, the first SPCLK edge is the sampling strobe of MSB (for an example of LSBFE = 0, MSB first). Therefore, the Slave should shift its MSB data before the first SPCLK edge. The falling edge of $\overline{SS}$ is used for preparing the MSB on MISO line. The $\overline{SS}$ pin therefore should toggle high and then low between each successive serial byte. Furthermore, if the slave writes data to the SPI data register (SPInDR) while $\overline{SS}$ is low, a write collision error occurs.

When CPHA = 1, the sampling edge thus locates on the second edge of SPCLK clock. The Slave uses the first SPCLK clock to shift MSB out rather than the $\overline{SS}$ falling edge. Therefore, the $\overline{SS}$ line can remain low between successive transfers. This format may be preferred in systems having single fixed Master and single fixed Slave. The $\overline{SS}$ line of the unique Slave device can be tied to GND as long as only CPHA = 1 clock mode is used.

The SPI should be configured before it is enabled (SPIEN = 1), or a change of LSBFE, MSTR, CPOL, CPHA and SPR[1:0] will abort a transmission in progress and force the SPI system into idle state. Prior to any configuration bit changed, SPIEN must be disabled first.

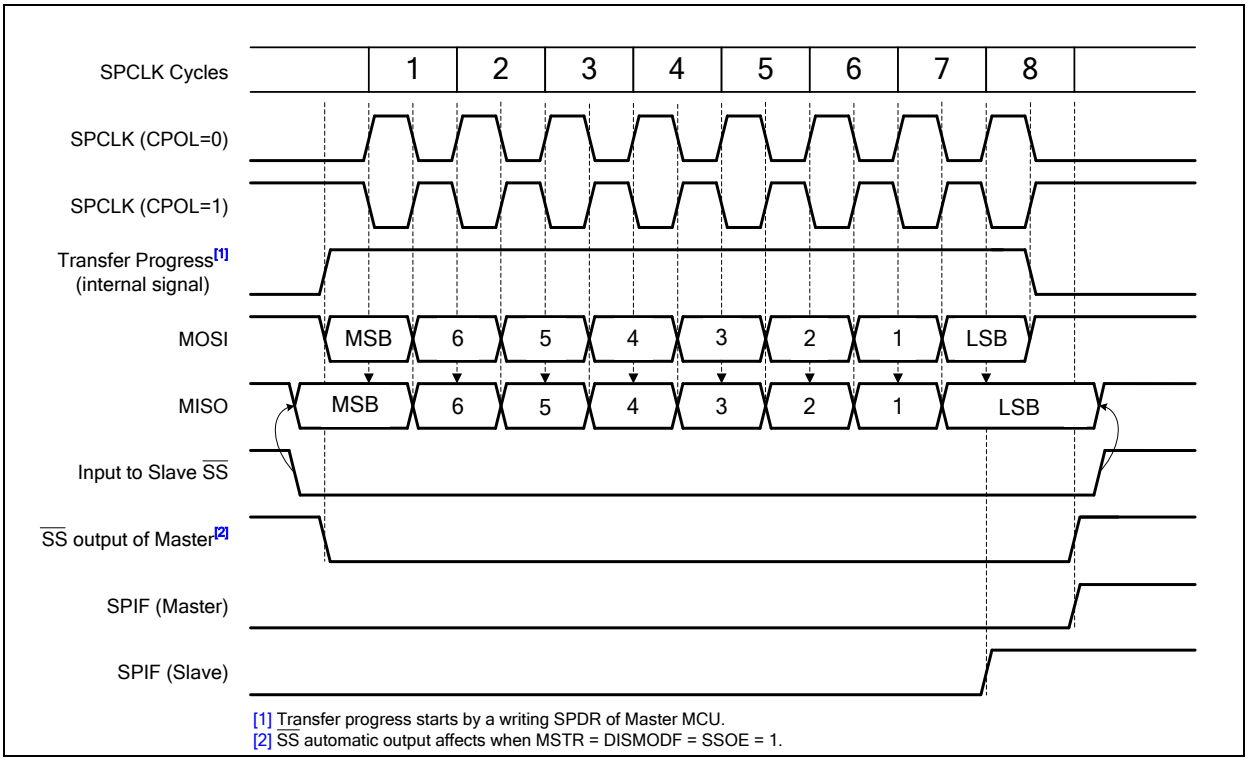


Figure 6.12-5 SPI Clock and Data Format with CPHA = 0

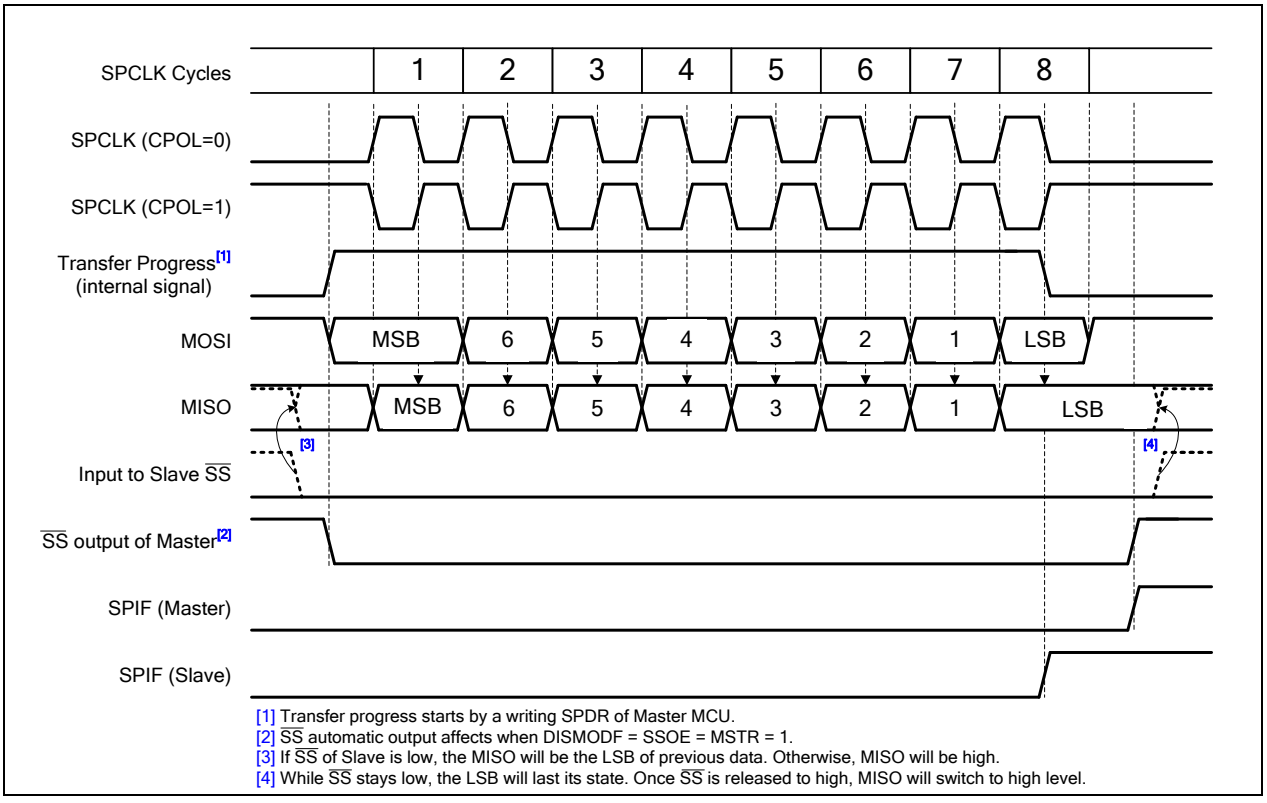


Figure 6.12-6 SPI Clock and Data Format with CPHA = 1

6.12.4.5 Slave Select Pin Configuration

The ML51/ML54/ML56 Series SPI gives a flexible $\overline{SS}$ pin feature for different system requirements. When the SPI operates as a Slave, $\overline{SS}$ pin always rules as Slave select input. When the Master mode is enabled, $\overline{SS}$ has three different functions according to DISMODF (SPInSR.3) and SSOE (SPInCR.7). By default, DISMODF is 0. It means that the Mode Fault detection activates. $\overline{SS}$ is configured as a input pin to check if the Mode Fault appears. On the contrary, if DISMODF is 1, Mode Fault is inactivated and the SSOE bit takes over to control the function of the $\overline{SS}$ pin. While SSOE is 1, it means the Slave select signal will generate automatically to select a Slave device. The $\overline{SS}$ as output pin of the Master usually connects with the $\overline{SS}$ input pin of the Slave device. The $\overline{SS}$ output automatically goes low for each transmission when selecting external Slave device and goes high during each idle state to de-select the Slave device. While SSOE is 0 and DISMODF is 1, $\overline{SS}$ is no more used by the SPI and reverts to be a general purpose I/O pin.

DISMODF	SSOE	Master Mode (MSTR = 1)	Slave Mode (MSTR = 0)
0	X	$\overline{SS}$ input for Mode Fault	$\overline{SS}$ Input for Slave select
1	0	General purpose I/O	
1	1	Automatic $\overline{SS}$ output	

Table 6.12-3 Slave Select Pin Configurations

6.12.4.6 Mode Fault Detection

The Mode Fault detection is useful in a system where more than one SPI devices might become Masters at the same time. It may induce data contention. When the SPI device is configured as a Master and the $\overline{SS}$ input line is configured for Mode Fault input depending on SPInCR0, a Mode Fault error occurs once the $\overline{SS}$ is pulled low by others. It indicates that some other SPI device is trying to address this Master as if it is a Slave. Instantly the MSTR and SPIEN control bits in the SPInCR are cleared via hardware to disable SPI, Mode Fault flag MODF (SPInSR.4) is set and an interrupt is generated if ESPI and EA are enabled.

Write Collision Error

The SPI is signal buffered in the transfer direction and double buffered in the receiving and transmit direction. New data for transmission cannot be written to the shift register until the previous transaction is complete. Write collision occurs while SPInDR be written more than once while a transfer was in progress. SPInDR is double buffered in the transmit direction. Any writing to SPInDR cause data to be written directly into the SPI shift register. Once a write collision error is generated, WCOL (SPInSR.6) will be set as 1 via hardware to indicate a write collision. In this case, the current transferring data continues its transmission. However the new data that caused the collision will be lost. Although the SPI logic can detect write collisions in both Master and Slave modes, a write collision is normally a Slave error because a Slave has no indicator when a Master initiates a transfer. During the receiving of Slave, a write to SPInDR causes a write collision in Slave mode. WCOL flag needs to be cleared via software.

Overrun Error

For receiving data, the SPI is double buffered in the receiving direction. The received data is transferred into a parallel read data buffer so the shifter is free to accept a second serial byte. However, the received data should be read from SPInDR before the next data has been completely shifted in. As long as the first byte is read out of the read data buffer and SPIF is cleared before the next byte is ready to be transferred, no overrun error condition occurs. Otherwise the overrun error occurs. In this condition, the second byte data will not be successfully received into the read data register and the previous data will remains. If overrun occur, SPIOVF (SPInSR.5) will be set via hardware. An SPIOVF setting will also require an interrupt if enabled. Figure 6.12-7 SPI Overrun Waveform shows the relationship between the data receiving and the overrun error.

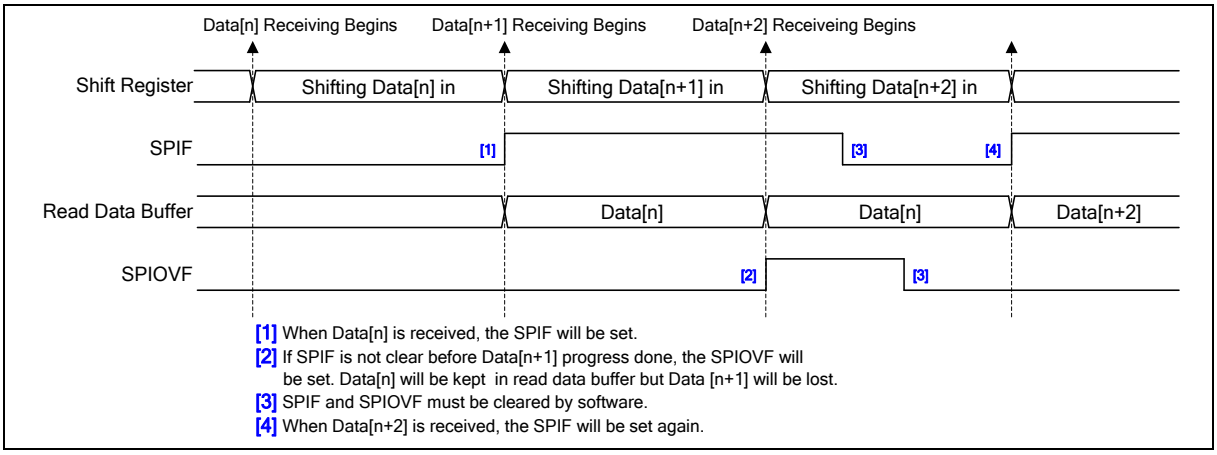


Figure 6.12-7 SPI Overrun Waveform

6.12.4.7 SPI Interrupt

Three SPI status flags, SPIF, MODF, and SPIOVF, can generate an SPI event interrupt requests. All of them locate in SPInSR. SPIF will be set after completion of data transfer with external device or a new data have been received and copied to SPInDR. MODF becomes set to indicate a low level on $\overline{SS}$ causing the Mode Fault state. SPIOVF denotes a receiving overrun error. If SPI interrupt mask is enabled via setting ESPI and EA is 1, CPU will executes the SPI interrupt service routine once any of these three flags is set. User needs to check flags to determine what event caused the interrupt. These three flags are software cleared.

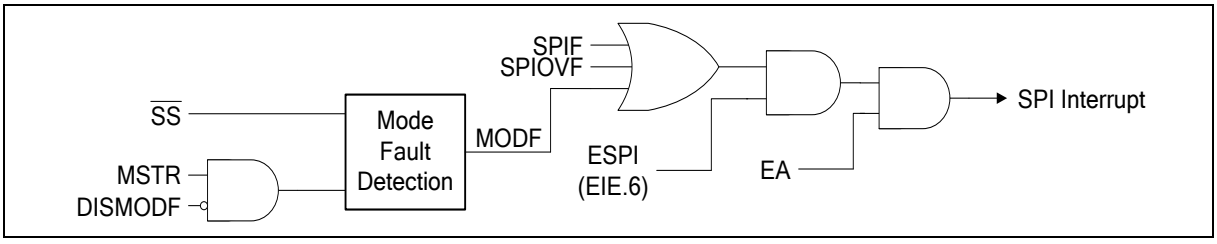


Figure 6.12-8 SPI Interrupt Request

6.12.5 Register Description

SPInCR0 – Serial Peripheral Control Register0

Register	SFR Address	Reset Value
SPI0CR0	F3H, Page 0	0000_0000 b
SPI1CR0	F9H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
SSOE	SPIEN	LSBFE	MSTR	CPOL	CPHA	SPR1	SPR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	SSOE	**Slave Select Output Enable** This bit is used in combination with the DISMODF (SPInSR.3) bit to determine the feature of $\overline{SS}$ pin as shown inTable 6.12-3 Slave Select Pin Configurations. This bit takes effect only under MSTR = 1 and DISMODF = 1 condition. 0 = $\overline{SS}$ functions as a general purpose I/O pin. 1 = $\overline{SS}$ automatically goes low for each transmission when selecting external Slave device and goes high during each idle state to de-select the Slave device.
[6]	SPIEN	**SPI Enable** 0 = SPI function Disabled. 1 = SPI function Enabled.
[5]	LSBFE	**LSB First Enable** 0 = The SPI data is transferred MSB first. 1 = The SPI data is transferred LSB first.
[4]	MSTR	**Master Mode Enable** This bit switches the SPI operating between Master and Slave modes. 0 = The SPI is configured as Slave mode. 1 = The SPI is configured as Master mode.
[3]	CPOL	**SPI Clock Polarity Select** CPOL bit determines the idle state level of the SPI clock. See Figure 6.12-4 SPI Clock Formats 0 = The SPI clock is low in idle state. 1 = The SPI clock is high in idle state.
[2]	CPHA	**SPI Clock Phase Select** CPHA bit determines the data sampling edge of the SPI clock. See Figure 6.12-4 SPI Clock Formats 0 = The data is sampled on the first edge of the SPI clock. 1 = The data is sampled on the second edge of the SPI clock.

Bit	Name	Description
[1:0]	**SPR[1:0]**	**SPI Clock Rate Select** These four bits select four grades of SPI clock divider. The clock rates below are illustrated under F_{SYS} = 24 MHz condition. See Table 6.12-1 SPI Master Clock Rate Define Table SPR[3:0] are valid only under Master mode (MSTR = 1). If under Slave mode, the clock will automatically synchronize with the external clock on SPICLK pin from Master device up to F_{SYS}/4 communication speed.

SPInCR1 – Serial Peripheral Control Register1

Register	SFR Address	Reset Value
SPI0CR1	F3H, Page 1	0000_0000 b
SPI1CR1	FAH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
-		SPR3	SPR2	TXDMAEN	RXDMAEN	SPIS1	SPIS0
-		R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:6]	-	Reserved.
[5:4]	SPR[3:2]	**SPI Clock Rate Select** These two bits select four grades of SPI clock divider. The clock rates below are illustrated under F_{SYS} = 24 MHz condition. Table 6.12-1 SPI Master Clock Rate Define Table SPR[3:0] are valid only under Master mode (MSTR = 1). If under Slave mode, the clock will automatically synchronize with the external clock on SPICLK pin from Master device up to F_{SYS}/4 communication speed.
[3]	TXDMAEN	**SPI TX DMA Enable** This bit enables the SPI TX operating by through PDMA transfer, TX data needs to be ready in XRAM before SPI TX starting. 0 = SPI TX DMA Disabled 1 = SPI TX DMA Enabled
[2]	RXDMAEN	**SPI RX DMA Enable** This bit enables the SPI RX operating by through PDMA transfer, RX data are saved in XRAM after SPI RX operation. 0 = SPI RX DMA Disabled 1 = SPI RX DMA Enabled
[1:0]	SPIS[1:0]	**SPI Interval Time Selection Between Adjacent Bytes** SPIS[1:0] and CPHA select eight grades of SPI interval time selection between adjacent bytes. As see Table 6.12-2 SPI Clock Suspend Interval Select SPIS[1:0] are valid only under Master mode (MSTR = 1).

SPInSR – Serial Peripheral Status Register

Register	SFR Address	Reset Value
SPI0SR	F4H, Page 0	0000_0000 b
SPI1SR	FBH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
SPIF	**WCOL**	**SPIOVF**	**MODF**	**DISMODF**	**DISSPIF**	**TXBFF**	**-**
R/W	R/W	R/W	R/W	R/W	R/W	R	-

Bit	Name	Description
[7]	**SPIF**	**SPI Complete Flag** This bit is set to logic 1 via hardware while an SPI data transfer is complete or an receiving data has been moved into the SPI read buffer. If ESPI and EA are enabled, an SPI interrupt will be required. This bit should be cleared via software. Attempting to write to SPInDR is inhibited if SPIF is set.
[6]	**WCOL**	**Write Collision Error Flag** This bit indicates a write collision event. Once a write collision event occurs, this bit will be set. It should be cleared via software.
[5]	**SPIOVF**	**SPI Overrun Error Flag** This bit indicates an overrun event. Once an overrun event occurs, this bit will be set. If ESPI and EA are enabled, an SPI interrupt will be required. This bit should be cleared via software.
[4]	**MODF**	**Mode Fault Error Flag** This bit indicates a Mode Fault error event. If $\overline{SS}$ pin is configured as Mode Fault input (MSTR = 1 and DISMODF = 0) and $\overline{SS}$ is pulled low by external devices, a Mode Fault error occurs. Instantly MODF will be set as logic 1. If ESPI and EA are enabled, an SPI interrupt will be required. This bit should be cleared via software.
[3]	**DISMODF**	**Disable Mode Fault Error Detection** This bit is used in combination with the SSOE (SPInCR.7) bit to determine the feature of $\overline{SS}$ pin as shown in Table 6.12-3 Slave Select Pin Configurations. DISMODF is valid only in Master mode (MSTR = 1). 0 = Mode Fault detection Enabled. $\overline{SS}$ serves as input pin for Mode Fault detection disregard of SSOE. 1 = Mode Fault detection Disabled. The feature of $\overline{SS}$ follows SSOE bit.
[2]	**DISSPIF**	**Disable SPI Complete Interrupt** This bit is used to disable SPI complete interrupt while an SPI data transfer is complete or an receiving data has been moved into the SPI read buffer. Especially in SPI PDMA operation. 0 = SPI Complete Interrupt Enabled while ESPI and EA are enabled, 1 = SPI Complete Interrupt Disabled
[1]	**TXBFF**	**SPI TX Buffer Full Flag** 0 = SPI TX buffer is empty 1 = SPI TX buffer is full

SPInDR – Serial Peripheral Data Register

Register	SFR Address	Reset Value
SPI0DR	F5H, Page 0	0000_0000 b
SPI1DR	FCH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
SPInDR[7:0]							
R/W							

Bit	Name	Description
[7:0]	**SPInDR[7:0]**	**Serial Peripheral Data** This byte is used for transmitting or receiving data on SPI bus. A write of this byte is a write to the shift register. A read of this byte is actually a read of the read data buffer. In Master mode, a write to this register initiates transmission and reception of a byte simultaneously.

6.13 Inter-Integrated Circuit (I^2C)

6.13.1 Overview

The ML51/ML54/ML56 Series provides two Inter-Integrated Circuit (I^2C) bus to serves as an serial interface between the microcontrollers and the I^2C devices such as EEPROM, LCD module, temperature sensor, and so on. The I^2C bus used two wires design (a serial data line SDA and a serial clock line SCL) to transfer information between devices.

The I^2C bus uses bi-directional data transfer between masters and slaves. There is no central master and the multi-master system is allowed by arbitration between simultaneously transmitting masters. The serial clock synchronization allows devices with different bit rates to communicate via one serial bus. The I^2C bus supports four transfer modes including master transmitter, master receiver, slave receiver, and slave transmitter. The I^2C interface only supports 7-bit addressing mode. A special mode General Call is also available. The I^2C can meet both standard (up to 100kbps) and fast (up to 400k bps) speeds.

6.13.2 Features

- 2 sets of I^2C devices
- Master/Slave mode
- Bidirectional data transfer between masters and slaves
- Multi-master bus (no central master)
- 7-bit addressing mode
- Standard mode (100 kbps) and Fast mode (400 kbps).
- Supports 8-bit time-out counter requesting the I^2C interrupt if the I^2C bus hangs up and timer-out counter overflows
- Multiple address recognition (four slave addresses with mask option)
- Supports hold time programmable

6.13.3 Functional Description

For a bi-directional transfer operation, the SDA and SCL pins should be open-drain pads. This implements a wired-AND function, which is essential to the operation of the interface. A low level on a I²C bus line is generated when one or more I²C devices output a “0”. A high level is generated when all I²C devices output “1”, allowing the pull-up resistors to pull the line high. In ML51, user should set output latches of SCL and SDA. As logic 1 before enabling the I²C function by setting I2CEN.

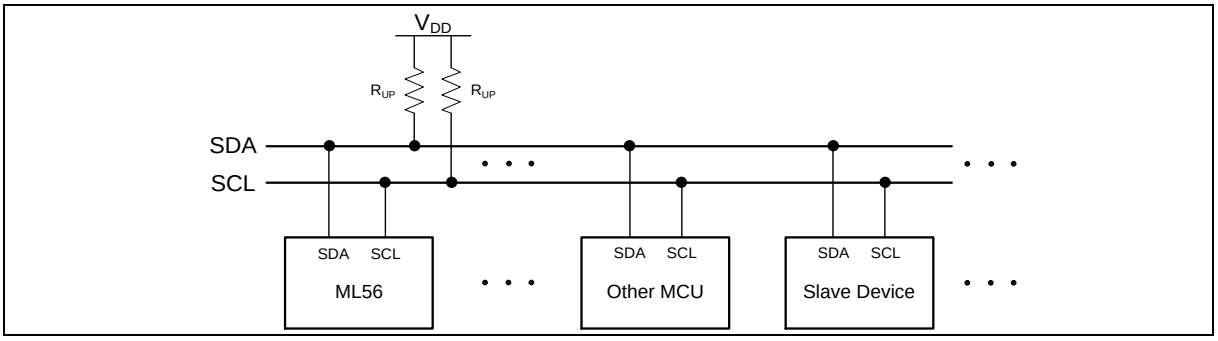


Figure 6.13-1 I²C Bus Interconnection

The I²C is considered free when both lines are high. Meanwhile, any device, which can operate as a master can occupy the bus and generate one transfer after generating a START condition. The bus now is considered busy before the transfer ends by sending a STOP condition. The master generates all of the serial clock pulses and the START and STOP condition. However if there is no START condition on the bus, all devices serve as not addressed slave. The hardware looks for its own slave address or a General Call address. (The General Call address detection may be enabled or disabled by GC (I2CnADDRx.0).) If the matched address is received, an interrupt is requested.

Every transaction on the I²C bus is 9 bits long, consisting of 8 data bits (MSB first) and a single acknowledge bit. The number of bytes per transfer (defined as the time between a valid START and STOP condition) is unrestricted but each byte has to be followed by an acknowledge bit. The master device generates 8 clock pulse to send the 8-bit data. After the 8th falling edge of the SCL line, the device outputting data on the SDA changes that pin to an input and reads in an acknowledge value on the 9th clock pulse. After 9th clock pulse, the data receiving device can hold SCL line stretched low if next receiving is not prepared ready. It forces the next byte transaction suspended. The data transaction continues when the receiver releases the SCL line.

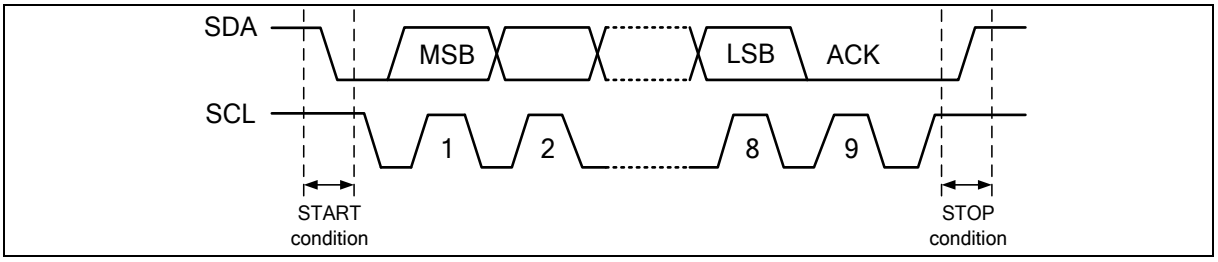


Figure 6.13-2 I²C Bus Protocol

6.13.3.1 START and STOP Condition

The protocol of the I²C bus defines two states to begin and end a transfer, START (S) and STOP (P) conditions. A START condition is defined as a high-to-low transition on the SDA line while SCL line is high. The STOP condition is defined as a low-to-high transition on the SDA line while SCL line is high. A START or a STOP condition is always generated by the master and I²C bus is considered busy after a START condition and free after a STOP condition. After issuing the STOP condition successful, the original master device will release the control authority and turn back as a not addressed slave.

Consequently, the original addressed slave will become a not addressed slave. The I^{2}C bus is free and listens to next START condition of next transfer.

A data transfer is always terminated by a STOP condition generated by the master. However, if a master still wishes to communicate on the bus, it can generate a repeated START (Sr) condition and address the pervious or another slave without first generating a STOP condition. Various combinations of read/write formats are then possible within such a transfer.

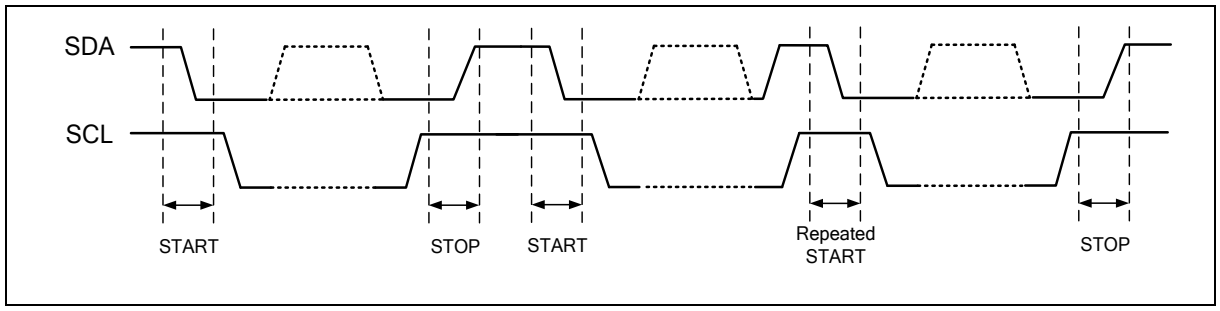


Figure 6.13-3 START, Repeated START, and STOP Conditions

6.13.3.2 7-Bit Address with Data Format

Following the START condition is generated, one byte of special data should be transmitted by the master. It includes a 7-bit long slave address (SLA) following by an 8th bit, which is a data direction bit (R/W), to address the target slave device and determine the direction of data flow. If R/W bit is 0, it indicates that the master will write information to a selected slave. Also, if R/W bit is 1, it indicates that the master will read information from the addressed slave. An address packet consisting of a slave address and a read I or a write (W) bit is called SLA+R or SLA+W, respectively. A transmission basically consists of a START condition, a SLA+W/R, one or more data packets and a STOP condition. After the specified slave is addressed by SLA+W/R, the second and following 8-bit data bytes issue by the master or the slave devices according to the R/W bit configuration.

Figure 6.13-4 Master Transmits Data to Slave by 7-bit shows a master transmits data to slave by 7-bit. A master addresses a slave with a 7-bit address and 1-bit write index to denote that the master wants to transmit data to the slave. The master keeps transmitting data after the slave returns acknowledge to the master.

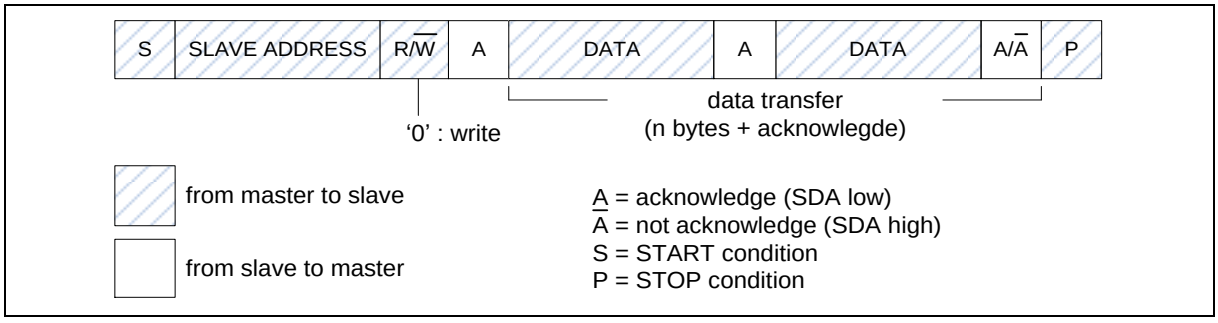


Figure 6.13-4 Master Transmits Data to Slave by 7-bit

Figure 6.13-5 Master Reads Data from Slave by 7-bit shows a master read data from slave by 7-bit. A master addresses a slave with a 7-bit address and 1-bit read index to denote that the master wants to read data from the slave. The slave will start transmitting data after the slave returns acknowledge to the master.

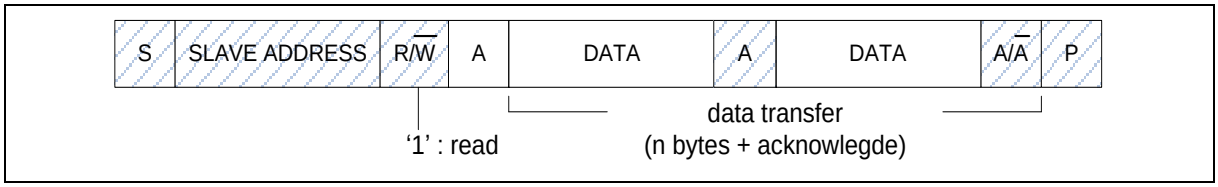


Figure 6.13-5 Master Reads Data from Slave by 7-bit

There is an exception called "General Call" address, which can address all devices by giving the first byte of data all 0. A General Call is used when a master wishes to transmit the same message to several slaves in the system. When this address is used, other devices may respond with an acknowledge or ignore it according to individual software configuration. If a device response the General Call, it operates as like in the slave-receiver mode. Note that the address 0x00 is reserved for General Call and cannot be used as a slave address, therefore, in theory, a 7-bit addressing I^2C bus accepts 127 devices with their slave addresses 1 to 127.

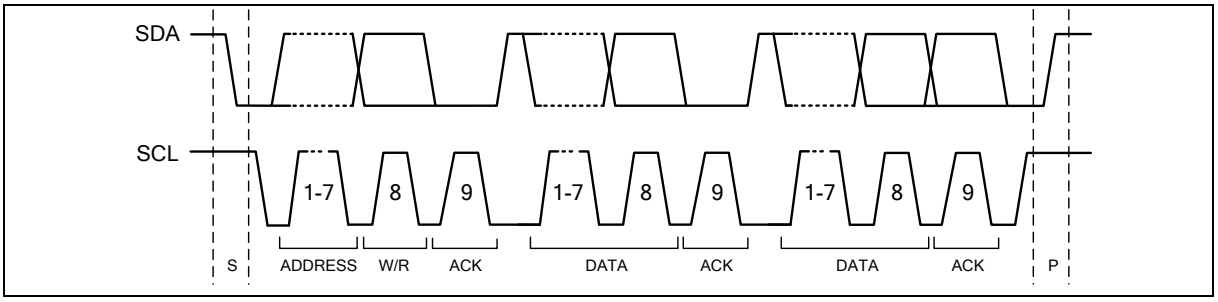


Figure 6.13-6 Data Format of One I^2C Transfer

During the data transaction period, the data on the SDA line should be stable during the high period of the clock, and the data line can only change when SCL is low.

6.13.3.3 Acknowledge

The 9th SCL pulse for any transferred byte is dedicated as an Acknowledge (ACK). It allows receiving devices (which can be the master or slave) to respond back to the transmitter (which also can be the master or slave) by pulling the SDA line low. The acknowledge-related clock pulse is generated by the master. The transmitter should release control of SDA line during the acknowledge clock pulse. The ACK is an active-low signal, pulling the SDA line low during the clock pulse high duty, indicates to the transmitter that the device has received the transmitted data. Commonly, a receiver, which has been addressed is requested to generate an ACK after each byte has been received. When a slave receiver does not acknowledge (NACK) the slave address, the SDA line should be left high by the slave so that the mater can generate a STOP or a repeated START condition.

If a slave-receiver does acknowledge the slave address, it switches itself to not addressed slave mode and cannot receive any more data bytes. This slave leaves the SDA line high. The master should generate a STOP or a repeated START condition.

If a master-receiver is involved in a transfer, because the master controls the number of bytes in the transfer, it should signal the end of data to the slave-transmitter by not generating an acknowledge on the last byte. The slave-transmitter then switches to not addressed mode and releases the SDA line to allow the master to generate a STOP or a repeated START condition.

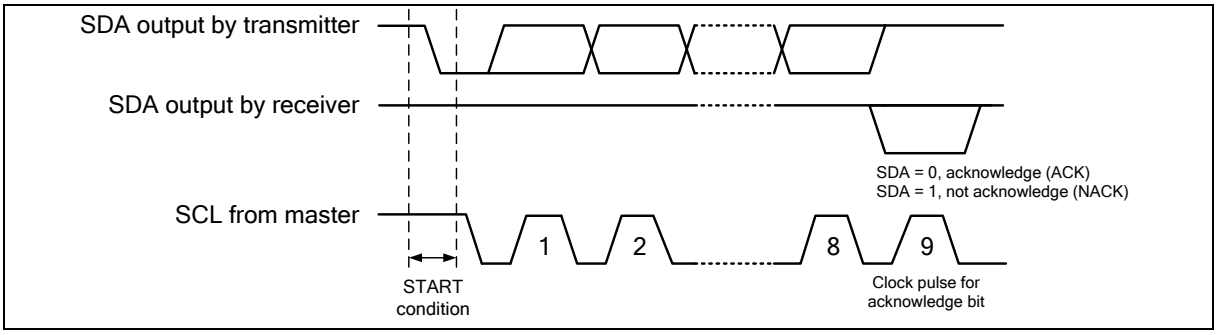


Figure 6.13-7 Acknowledge Bit

6.13.3.4 Arbitration

A master may start a transfer only if the bus is free. It is possible for two or more masters to generate a START condition. In these situations, an arbitration scheme takes place on the SDA line, while SCL is high. During arbitration, the first of the competing master devices to place'a''1' (high) on SDA while another master transmits'a''0' (low) switches off its data output stage because the level on the bus does not match its own level. The arbitration lost master switches to the not addressed slave immediately to detect its own slave address in the same serial transfer whether it is being addressed by the winning master. It also releases SDA line to high level for not affecting the data transfer continued by the winning master. However, the arbitration lost master continues generating clock pulses on SCL line until the end of the byte in which it loses the arbitration.

Arbitration is carried out by all masters continuously monitoring the SDA line after outputting data. If the value read from the SDA line does not match the value that the master has to output, it has lost the arbitration. Note that a master can only lose arbitration when it outputs a high SDA value while another master outputs a low value. Arbitration will continue until only one master remains, and this may take many bits. Its first stage is a comparison of address bits, and if both masters are trying to address the same device, arbitration continues on to the comparison of data bits or acknowledge bit.

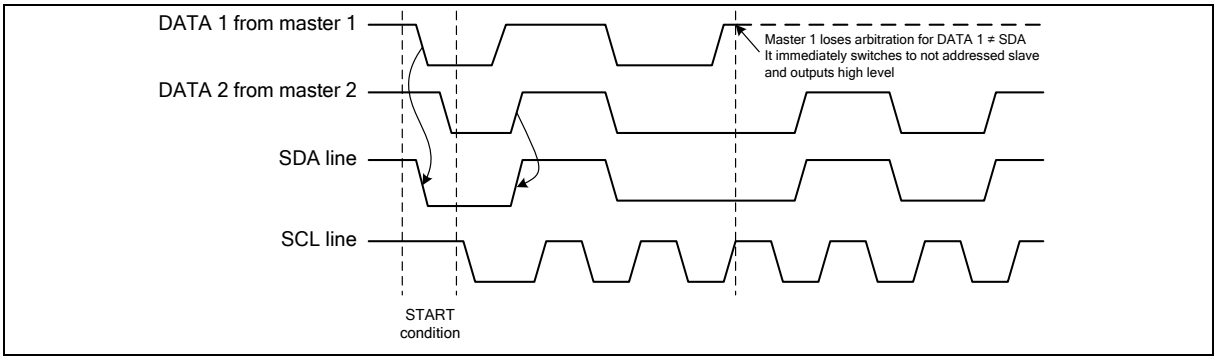


Figure 6.13-8 Arbitration Procedure of Two Masters

Since control of the I^2C bus is decided solely on the address or master code and data sent by competing masters, there is no central master, nor any order of priority on the bus. Slaves are not involved in the arbitration procedure.

6.13.3.5 Operation Modes

The on-chip I2C ports support three operation modes, Master, Slave, and General Call Mode.

In a given application, I2C port may operate as a master or as a slave. In Slave mode, the I2C port hardware looks for its own slave address and the general call address. If one of these addresses is detected, and if the slave is willing to receive or transmit data from/to master(by setting the AA bit), acknowledge pulse will be transmitted out on the 9th clock, hence an interrupt is requested on both master and slave devices if interrupt is enabled. When the microcontroller wishes to become the bus

master, hardware waits until the bus is free before entering Master mode so that a possible slave action is not be interrupted. If bus arbitration is lost in Master mode, I2C port switches to Slave mode immediately and can detect its own slave address in the same serial transfer.

To control the I2C bus transfer in each mode, user needs to set I2C_CTL0, I2C_DAT registers according to current status code of I2C_STATUS0 register. In other words, for each I2C bus action, user needs to check current status by I2C_STATUS0 register, and then set I2C_CTL0, I2C_DAT registers to take bus action. Finally, check the response status by I2C_STATUS0.

The bits, STA, STO and AA in I2C_CTL0 register are used to control the next state of the I2C hardware after SI flag of I2C_CTL0 [3] register is cleared. Upon completion of the new action, a new status code will be updated in I2C_STATUS0 register and the SI flag of I2C_CTL0 register will be set. But the SI flag will not be set when I2C STOP. If the I2C interrupt control bit INTEN (I2C_CTL0 [7]) is set, appropriate action or software branch of the new status code can be performed in the Interrupt service routine.

Figure 6.13-9 Control I2C Bus according to the Current I2C Status shows the current I2C status code is 0x08, and then set I2C_DATA=SLA+W and (STA,STO,SI,AA) = (0,0,1,x) to send the address to I2C bus. If a slave on the bus matches the address and response ACK, the I2C_STATUS0 will be updated by status code 0x18.

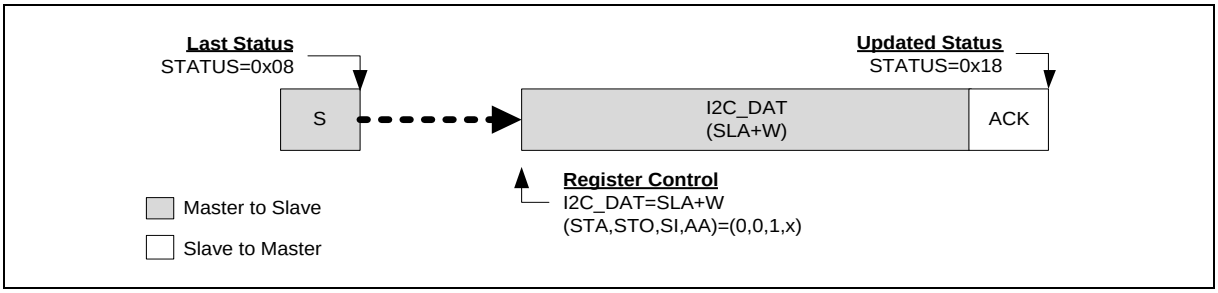


Figure 6.13-9 Control I^2C Bus according to the Current I^2C Status

Master Transmitter Mode

In the master transmitter mode, several bytes of data are transmitted to a slave receiver. The master should prepare by setting desired clock rate in I2CnCLK. The master transmitter mode may now be entered by setting STA (I2CnCON.5) bit as 1. The hardware will test the bus and generate a START condition as soon as the bus becomes free. After a START condition is successfully produced, the SI flag (I2CnCON.3) will be set and the status code in I2CnSTAT show 08H. The progress is continued by loading I2CnDAT with the target slave address and the data direction bit “write” (SLA+W). The SI bit should then be cleared to commence SLA+W transaction.

After the SLA+W byte has been transmitted and an acknowledge (ACK) has been returned by the addressed slave device, the SI flag is set again and I2CnSTAT is read as 18H. The appropriate action to be taken follows user defined communication protocol by sending data continuously. After all data is transmitted, the master can send a STOP condition by setting STO (I2CnCON.4) and then clearing SI to terminate the transmission. A repeated START condition can also be generated without sending STOP condition to immediately initial another transmission.

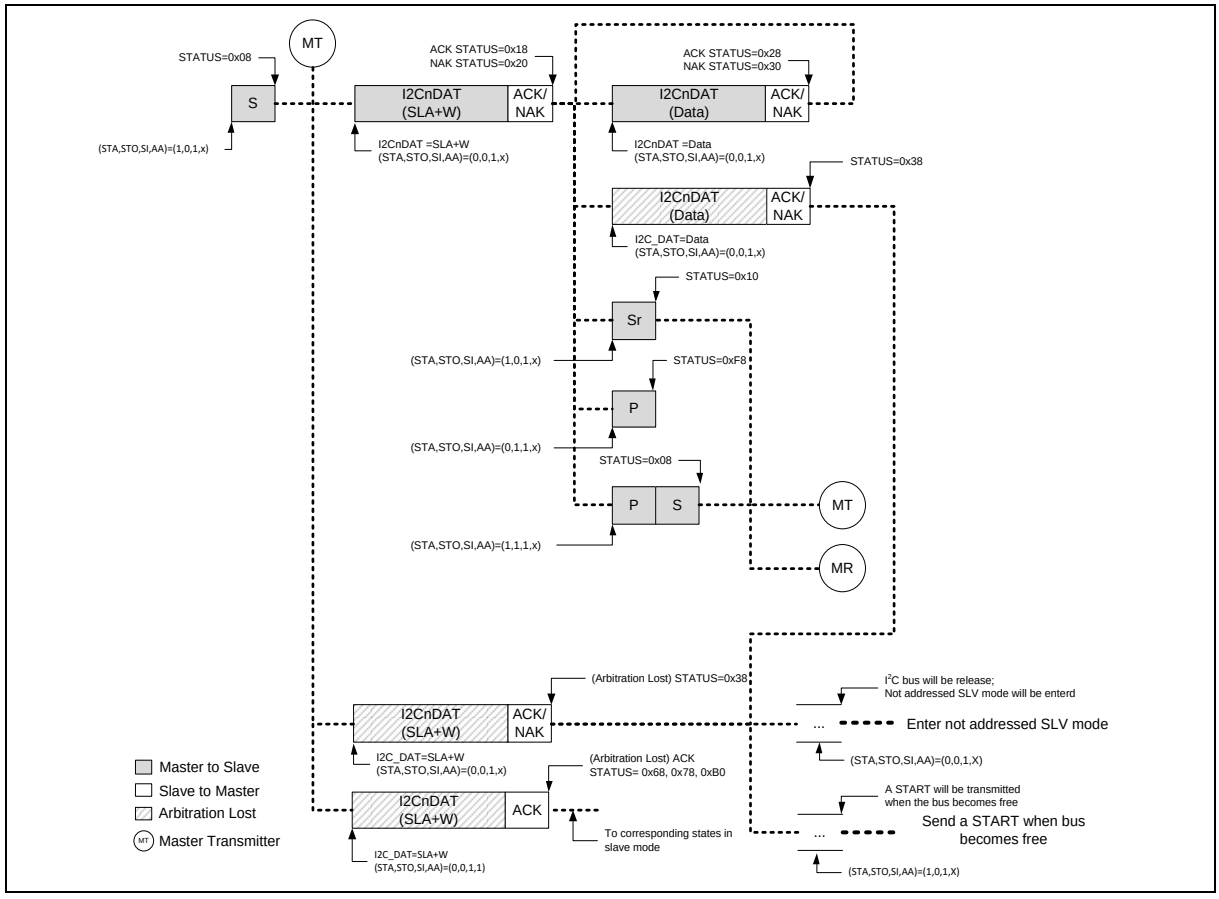


Figure 6.13-10 Flow and Status of Master Transmitter Mode

Master Receiver Mode

In the master receiver mode, several bytes of data are received from a slave transmitter. The transaction is initialized just as the master transmitter mode. Following the START condition, I2CnDAT should be loaded with the target slave address and the data direction bit “read” (SLA+R). After the SLA+R byte is transmitted and an acknowledge bit has been returned, the SI flag is set again and I2CnSTAT is read as 40H. SI flag then should be cleared to receive data from the slave transmitter. If AA flag (I2CnCON.2) is set, the master receiver will acknowledge the slave transmitter. If AA is cleared, the master receiver will not acknowledge the slave and release the slave transmitter as a not addressed slave. After that, the master can generate a STOP condition or a repeated START condition to terminate the transmission or initial another one.

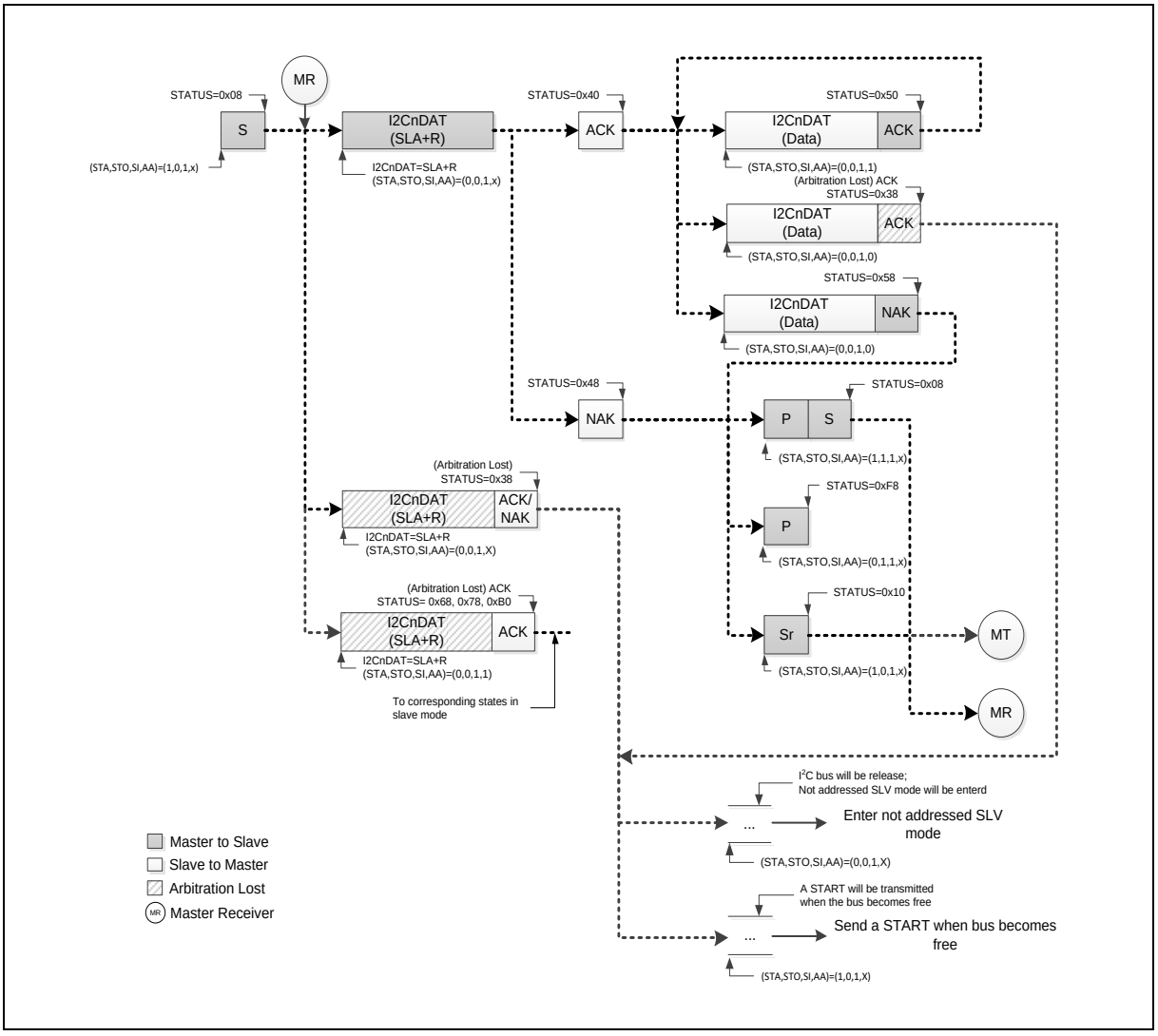


Figure 6.13-11 Flow and Status of Master Receiver Mode

Slave Receiver

In the slave receiver mode, several bytes of data are received form a master transmitter. Before a transmission is commenced, I2CnADDRx should be loaded with the address to which the device will respond when addressed by a master. I2CnCLK does not affect in slave mode. The AA bit should be set to enable acknowledging its own slave address. After the initialization above, the I^2C idles until it is addressed by its own address with the data direction bit “write” (SLA+W). The slave receiver mode may also be entered if arbitration is lost.

After the slave is addressed by SLA+W, it should clear its SI flag to receive the data from the master transmitter. If the AA bit is 0 during a transaction, the slave will return a non-acknowledge after the next received data byte. The slave will also become not addressed and isolate with the master. It cannot receive any byte of data with I2CnDAT remaining the previous byte of data, which is just received.

Slave Transmitter

The I^2C port is equipped with four slave address registers, I2CnADDRx (x=0~3). The contents of the register are irrelevant when I^2C is in Master mode. In the slave transmitter mode, several bytes of data are transmitted to a master receiver. After I2CnADDRx and I2CnCON values are given, the I^2C wait until it is addressed by its own address with the data direction bit “read” (SLA+R). The slave transmitter mode may also be entered if arbitration is lost.

After the slave is addressed by SLA+R, it should clear its SI flag to transmit the data to the master receiver. Normally the master receiver will return an acknowledge after every bytes of data is transmitted by the slave. If the acknowledge is not received, it will transmit all “1” data if it continues the transaction. It becomes a not addressed slave. If the AA flag is cleared during a transaction, the slave transmits the last byte of data. The next transmitting data will be all “1” and the slave becomes not addressed.

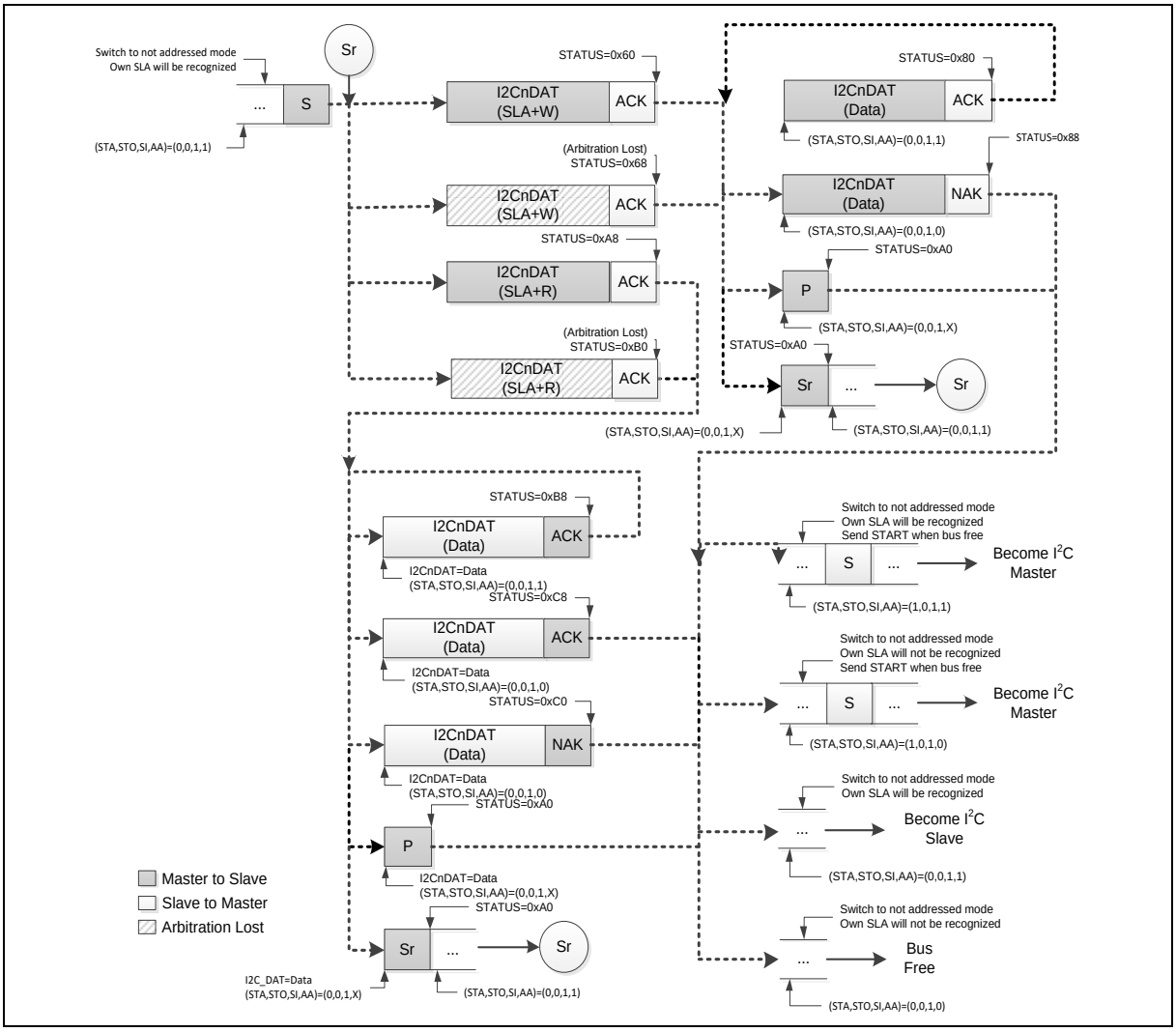


Figure 6.13-12 Flow and Status of Slave Receiver Mode

General Call

The General Call is a special condition of slave receiver mode by been addressed with all “0” data in slave address with data direction bit. Both GC (I2CnADDRx.0) bit and AA bit should be set as 1 to enable acknowledging General Calls. The slave addressed by a General Call has different status code in I2CnSTAT with normal slave receiver mode. The General Call may also be produced if arbitration is lost.

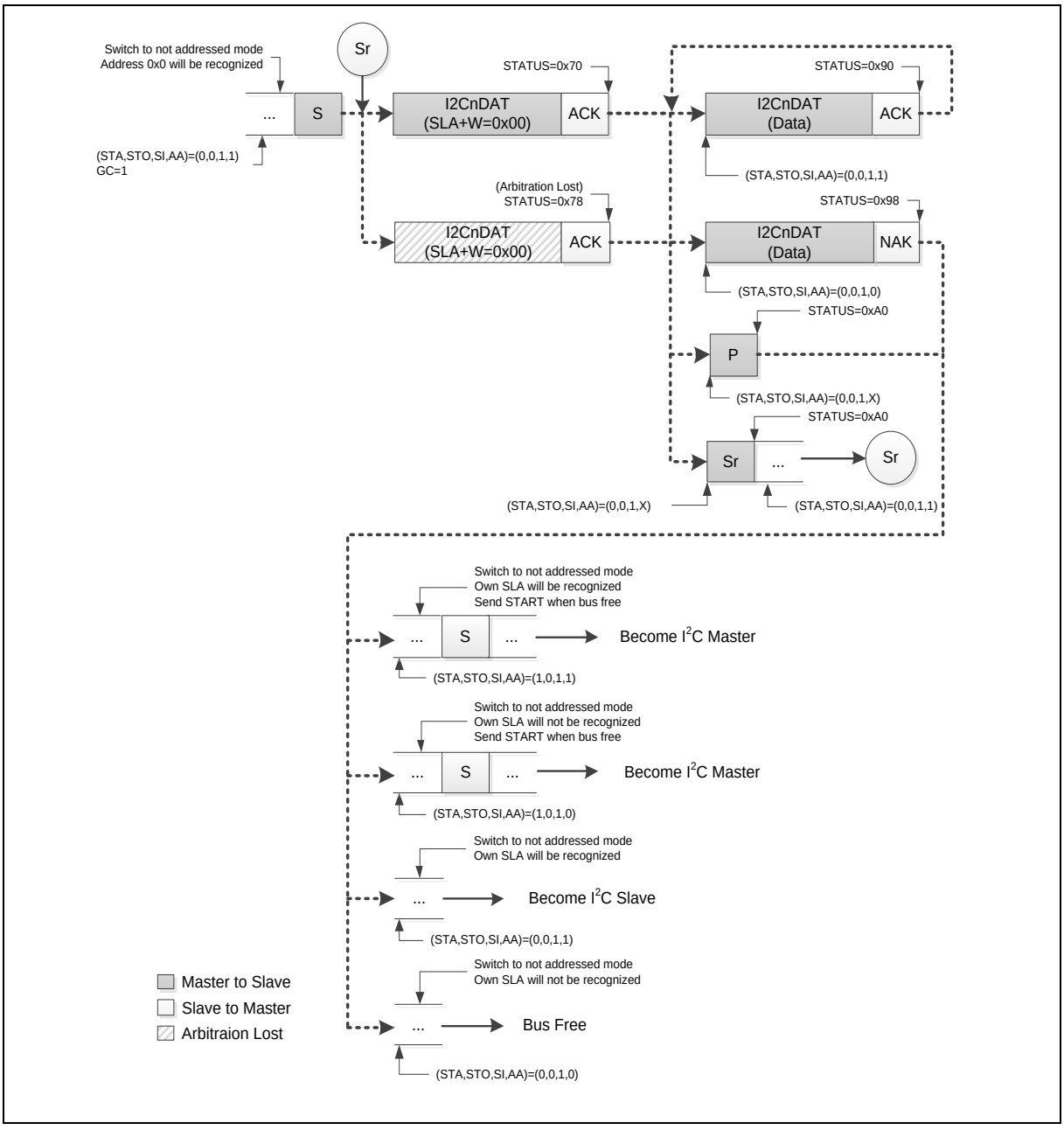


Figure 6.13-13 Flow and Status of General Call Mode

6.13.3.6 Miscellaneous States

There are two I2CnSTAT status codes that do not correspond to the 25 defined states, The first status code F8H indicates that no relevant information is available during each transaction. Meanwhile, the SI flag is 0 and no I^2C interrupt is required. The other status code 00H means a bus error has occurred during a transaction. A bus error is caused by a START or STOP condition appearing temporally at an illegal position such as the second through eighth bits of an address or a data byte, and the acknowledge bit. When a bus error occurs, the SI flag is set immediately. When a bus error is detected on the I^2C bus, the operating device immediately switches to the not addressed salve mode, releases SDA and SCL lines, sets the SI flag, and loads I2CnSTAT as 00H. To recover from a bus error, the STO bit should be set and then SI should be cleared. After that, STO is cleared by hardware and release the I^2C bus without issuing a real STOP condition waveform on I^2C bus.

There is a special case if a START or a repeated START condition is not successfully generated for I^2C bus is obstructed by a low level on SDA line e.g. a slave device out of bit synchronization, the problem can be solved by transmitting additional clock pulses on the SCL line. The I^2C hardware transmits additional clock pulses when the STA bit is set, but no START condition can be generated because the SDA line is pulled low. When the SDA line is eventually released, a normal START condition is transmitted, state 08H is entered, and the serial transaction continues. If a repeated START condition is transmitted while SDA is obstructed low, the I^2C hardware also performs the same action as above. In this case, state 08H is entered instead of 10H after a successful START condition is transmitted. Note that the software is not involved in solving these bus problems.

The following table is show the status display in I2STAT register of I^2C number and description:

Master Mode		**Slave Mode**	
STATUS	Description	STATUS	Description
0x08	Start	0xA0	Slave Transmit Repeat Start or Stop
0x10	Master Repeat Start	0xA8	Slave Transmit Address ACK
0x18	Master Transmit Address Ack	0xB0	Slave Transmit Arbitration Lost
0x20	Master Transmit Address Nack	0xB8	Slave Transmit Data ACK
0x28	Master Transmit Data Ack	0xC0	Slave Transmit Data NACK
0x30	Master Transmit Data Nack	0xC8	Slave Transmit Last Data ACK
0x38	Master Arbitration Lost	0x60	Slave Receive Address ACK
0x40	Master Receive Address Ack	0x68	Slave Receive Arbitration Lost
0x48	Master Receive Address Nack	0x80	Slave Receive Data ACK
0x50	Master Receive Data Ack	0x88	Slave Receive Data NACK
0x58	Master Receive Data Nack	0x70	GC mode Address ACK
0x00	Bus Error	0x78	GC mode Arbitration Lost
		0x90	GC mode Data ACK
		0x98	GC mode Data NACK
0xF8	Bus Released **Note:** Status “0xF8” exists in both master/slave modes, and it won’t raise interrupt.		

Figure 6.13-14 Status Display In I2STAT Register

6.13.3.7 I²C Time-Out

There is a 14-bit time-out counter, which can be used to deal with the I²C bus hang-up. If the time-out counter is enabled, the counter starts up counting until it overflows. Meanwhile I2TOF will be set by hardware and requests I²C interrupt. When time-out counter is enabled, setting flag SI to high will reset counter and restart counting up after SI is cleared. If the I²C bus hangs up, it causes the SI flag not set for a period. The 14-bit time-out counter will overflow and require the interrupt service.

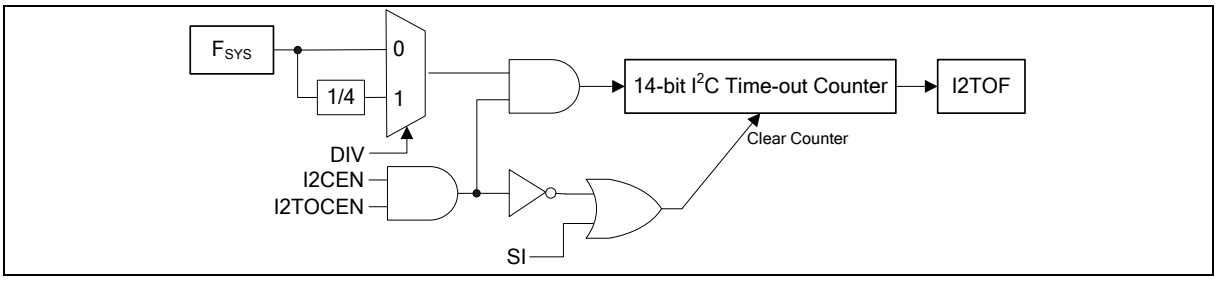


Figure 6.13-15 I²C Time-Out Counter

I2CnTOC – I²C Time-out Counter

Register	SFR Address	Reset Value
I2C0TOC	BFH, Page 0	0000_0000 b
I2C1TOC	B6H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	I2TOCEN	DIV	I2TOF
-	-	-	-	-	R/W	R/W	R/W

Bit	Name	Description
[7:3]	-	Reserved
[2]	I2TOCEN	**I²C0 Time-Out Counter Enable** 0 = I²C time-out counter Disabled. 1 = I²C time-out counter Enabled.
[1]	DIV	**I²C0 Time-Out Counter Clock Divider** 0 = The clock of I²C time-out counter is F_{SYS}/1. 1 = The clock of I²C time-out counter is F_{SYS}/4.

6.13.3.8 I²C Interrupt

There are two I²C flags, SI and I2TOF. Both of them can generate an I²C event interrupt requests. If I²C interrupt mask is enabled via setting EI2C and EA as 1, CPU will execute the I²C interrupt service routine once any of these two flags is set. User needs to check flags to determine what event caused the interrupt. Both of I²C flags are cleared by software.

6.13.4 Register Description

There are five Register Description to interface the I²C bus including I2CnCON, I2CnSTAT, I2CnDAT, I2CnADDRx, and I2CnCLK. These registers provide protocol control, status, data transmitting and receiving functions, and clock rate configuration. For application flexibility. The following registers relate to I²C function.

I2CnCON – I²C Control (Bit-addressable)

Register	SFR Address	Reset Value
I2C0CON	C0H, All pages	0000_0000 b
I2C1CON	E8H, All p Pages ages	0000_0000 b

7	6	5	4	3	2	1	0
I	**I2CEN**	**STA**	**STO**	**SI**	**AA**	**-**	**-**
R/W	R/W	R/W	R/W	R/W	R/W	-	-

Bit	Name	Description
[7]	**I**	**I²Cn Hold Time Extend Enable** 0 = I²C DATA to SCL hold time extend disabled 1 = I²C DATA to SCL hold time extend enabled, extend 8 system clock
[6]	**I2CEN**	**I²Cn Bus Enable** 0 = I²C bus Disabled. 1 = I²C bus Enabled. Before enabling the I²C, SCL and SDA port latches should be set to logic 1.
[5]	**STA**	**START Flag** When STA is set, the I²C generates a START condition if the bus is free. If the bus is busy, the I²C waits for a STOP condition and generates a START condition following. If STA is set while the I²C is already in the master mode and one or more bytes have been transmitted or received, the I²C generates a repeated START condition. Note that STA can be set anytime even in a slave mode, but STA is not hardware automatically cleared after START or repeated START condition has been detected. User should take care of it by clearing STA manually.
[4]	**STO**	**STOP Flag** When STO is set if the I²C is in the master mode, a STOP condition is transmitted to the bus. STO is automatically cleared by hardware once the STOP condition has been detected on the bus. The STO flag setting is also used to recover the I²C device from the bus error state (I2CnSTAT as 00H). In this case, no STOP condition is transmitted to the I²C bus. If the STA and STO bits are both set and the device is original in the master mode, the I²C bus will generate a STOP condition and immediately follow a START condition. If the device is in slave mode, STA and STO simultaneous setting should be avoid from issuing illegal I²C frames.

Bit	Name	Description
[3]	**SI**	**I2Cn Interrupt Flag** SI flag is set by hardware when one of 26 possible I2C status (besides F8H status) is entered. After SI is set, the software should read I2CnSTAT register to determine which step has been passed and take actions for next step. SI is cleared by software. Before the SI is cleared, the low period of SCL line is stretched. The transaction is suspended. It is useful for the slave device to deal with previous data bytes until ready for receiving the next byte. The serial transaction is suspended until SI is cleared by software. After SI is cleared, I2C bus will continue to generate START or repeated START condition, STOP condition, 8-bit data, or so on depending on the software configuration of controlling byte or bits. Therefore, user should take care of it by preparing suitable setting of registers before SI is software cleared.
[2]	**AA**	**Acknowledge Assert Flag** If the AA flag is set, an ACK (low level on SDA) will be returned during the acknowledge clock pulse of the SCL line while the I2C device is a receiver or an own-address-matching slave. If the AA flag is cleared, a NACK (high level on SDA) will be returned during the acknowledge clock pulse of the SCL line while the I2C device is a receiver or an own-address-matching slave. A device with its own AA flag cleared will ignore its own salve address and the General Call. Consequently, SI will note be asserted and no interrupt is requested. Note that if an addressed slave does not return an ACK under slave receiver mode or not receive an ACK under slave transmitter mode, the slave device will become a not addressed slave. It cannot receive any data until its AA flag is set and a master addresses it again. There is a special case of I2CnSTAT value C8H occurs under slave transmitter mode. Before the slave device transmit the last data byte to the master, AA flag can be cleared as 0. Then after the last data byte transmitted, the slave device will actively switch to not addressed slave mode of disconnecting with the master. The further reading by the master will be all FFH.
[1:0]	**-**	Reserved

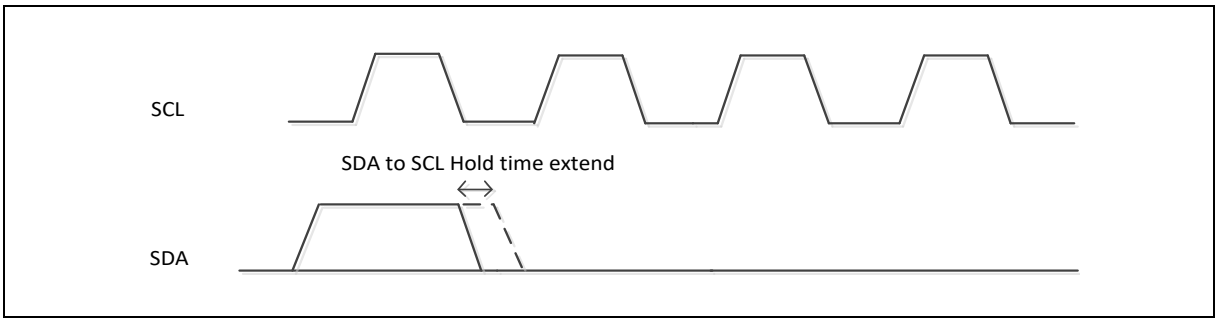


Figure 6.13 Hold Time extend enable

I2CnSTAT – I²C Status

Register	SFR Address	Reset Value
I2C0STAT	BDH, Page 0	1111_1000 b
I2C1STAT	B4H, Page 0	1111_1000 b

7	6	5	4	3	2	1	0
I2CnSTAT[7:3]					**0**	**0**	**0**
R					R	R	R

Bit	Name	Description
[7:3]	**I2CnSTAT[7:3]**	**I²Cn Status Code** The MSB five bits of I2CnSTAT contains the status code. There are 27 possible status codes. When I2CnSTAT is F8H, no relevant state information is available and SI flag keeps 0. All other 26 status codes correspond to the I²C states. When each of these status is entered, SI will be set as logic 1 and a interrupt is requested.
[2:0]	**0**	**Reserved** The least significant three bits of I2CnSTAT are always read as 0.

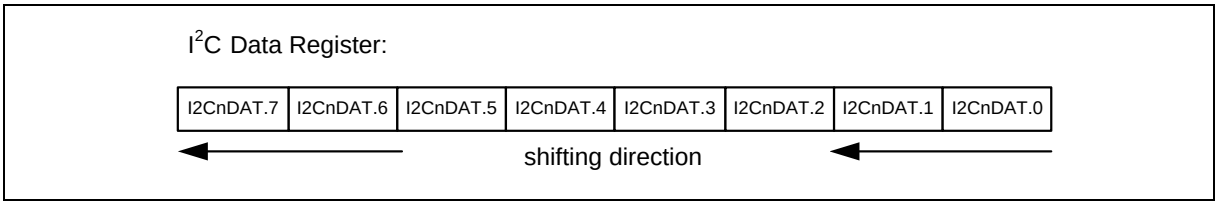
I2CnDAT – I²C Data

Register	SFR Address	Reset Value
I2C0DAT	BCH, Page 0	0000_0000 b
I2C1DAT	B3H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
I2CnDAT[7:0]							
R/W							

Bit	Name	Description
[7:0]	**I2CnDAT[7:0]**	**I²Cn Data** I2CnDAT contains a byte of the I²C data to be transmitted or a byte, which has just received. Data in I2CnDAT remains as long as SI is logic 1. The result of reading or writing I2CnDAT during I²C transceiver progress is unpredicted. While data in I2CnDAT is shifted out, data on the bus is simultaneously being shifted in to update I2CnDAT. I2CnDAT always shows the last byte that presented on the I²C bus. Thus the event of lost arbitration, the original value of I2CnDAT changes after the transaction.

I²C Data Shifting Direction.



I2CnADDRx – I2Cn Own Slave Address

Register	SFR Address	Reset Value
I2C0ADDR0	C1H, Page 0	0000_0000 b
I2C0ADDR1	A1H, Page 2	0000_0000 b
I2C0ADDR2	A2H, Page 2	0000_0000 b
I2C0ADDR3	A3H, Page 2	0000_0000 b
I2C1ADDR0	B2H, Page 0	0000_0000 b
I2C1ADDR1	A4H, Page 2	0000_0000 b
I2C1ADDR2	A5H, Page 2	0000_0000 b
I2C1ADDR3	A6H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
I2CnADDRx[7:1]							**GC**
R/W							R/W

Bit	Name	Description
[7:1]	**I2CnADDRx[7:1]**	**I2Cn Device's Own Slave Address** In master mode: These bits have no effect. In slave mode: These 7 bits define the slave address of this I2C device by user. The master should address I2C device by sending the same address in the first byte data after a START or a repeated START condition. If the AA flag is set, this I2C device will acknowledge the master after receiving its own address and become an addressed slave. Otherwise, the addressing from the master will be ignored. Note that I2CnADDRx[7:1] should not remain its default value of all 0, because address 0x00 is reserved for General Call.
[0]	**GC**	**General Call Bit** In master mode: This bit has no effect. In slave mode: 0 = The General Call is always ignored. 1 = The General Call is recognized if AA flag is 1; otherwise, it is ignored if AA is 0.

I2CnCLK – I²C Clock

Register	SFR Address	Reset Value
I2C0CLK	BEH, Page 0	0000_1001 b
I2C1CLK	B5H, Page 0	0000_1001 b

7	6	5	4	3	2	1	0
I2CnCLK[7:0]							
R/W							

Bit	Name	Description
[7:0]	**I2CnCLK[7:0]**	**I2Cn Clock Setting** In master mode: This register determines the clock rate of I²C bus when the device is in a master mode. The clock rate follows the equation, $\frac{F_{SYS}}{4 \times (I2CLK + 1)}$. Note that the I2CnCLK value of 00H and 01H are not valid. This is an implement limitation. In slave mode: This byte has no effect. In slave mode, the I²C device will automatically synchronize with any given clock rate up to 400k bps.

6.13.5 Typical Structure of I²C Interrupt Service Routine

The following software example in C language for KEIL™ C51 compiler shows the typical structure of the I²C interrupt service routine including the 26 state service routines and may be used as a base for user applications. User can follow or modify it for their own application. If one or more of the five modes are not used, the associated state service routines may be removed, but care should be taken that a deleted routine can never be invoked.

```
Void I2C_ISR (void) interrupt 6
{
   switch (I2STAT)
   {
      //================================================
      //Bus Error, always put in ISR for noise handling
      //================================================
   case 0x00:         /*00H, bus error occurs*/
        STO = 1;        //recover from bus error
        break;
      //============
      //Master Mode
      //============
   case 0x08:         /*08H, a START transmitted*/
        STA = 0;        //STA bit should be cleared by software
        I2DAT = SLA_ADDR1;   //load SLA+W/R
        break;
   case 0x10:         /*10H, a repeated START transmitted*/
        STA = 0;
        I2DAT = SLA_ADDR2;
        break;
      //========================
      //Master Transmitter Mode
      //========================
   case 0x18:         /*18H, SLA+W transmitted, ACK received*/
        I2DAT = NEXT_SEND_DATA1;  //load DATA
        break;
   case 0x20:         /*20H, SLA+W transmitted, NACK received*/
        STO = 1;        //transmit STOP
        AA = 1;      //ready for ACK own SLA+W/R or General Call
        break;
   case 0x28:         /*28H, DATA transmitted, ACK received*/
        if (Conti_TX_Data)   //if continuing to send DATA
        I2DAT = NEXT_SEND_DATA2;
        else         //if no DATA to be sent
        {
           STO = 1;
           AA = 1;
        }
        break;
   case 0x30:         /*30H, DATA transmitted, NACK received*/
        STO = 1;
        AA = 1;
        break;
      //============
      //Master Mode
      //============
```

```
case 0x38:         /*38H, arbitration lost*/
     STA = 1;        //retry to transmit START if bus free
     break;
   //====================
   //Master Receiver Mode
   //====================
case 0x40:         /*40H, SLA+R transmitted, ACK received*/
     AA = 1;       //ACK next received DATA
     break;
case 0x48:         /*48H, SLA+R transmitted, NACK received*/
     STO = 1;
     AA = 1;
     break;
case 0x50:         /*50H, DATA received, ACK transmitted*/
     DATA_RECEIVED1 = I2DAT; //store received DATA
     if (To_RX_Last_Data1)   //if last DATA will be received
        AA = 0;    //not ACK next received DATA
     else          //if continuing receiving DATA
        AA = 1;
     break;
case 0x58:         /*58H, DATA received, NACK transmitted*/
     DATA_RECEIVED_LAST1 = I2DAT;
     STO = 1;
     AA = 1;
     break;
   //====================================
   //Slave Receiver and General Call Mode
   //====================================
case 0x60:         /*60H, own SLA+W received, ACK returned*/
     AA = 1;
     break;
case 0x68:         /*68H, arbitration lost in SLA+W/R
                  own SLA+W received, ACK returned */
     AA = 0;       //not ACK next received DATA after
                 //arbitration lost
     STA = 1;        //retry to transmit START if bus free
     break;
    case 0x70:                                  //*70H, General Call received, ACK
                                                returned
     AA = 1;
     break;
case 0x78:           /*78H, arbitration lost in SLA+W/R
                  General Call received, ACK returned*/
     AA = 0;
     STA = 1;
     break;
case 0x80:         /*80H, previous own SLA+W, DATA received,
                  ACK returned*/
     DATA_RECEIVED2 = I2DAT;
     if (To_RX_Last_Data2)
        AA = 0;
     else
        AA = 1;
     break;
```

```
case 0x88:          /*88H, previous own SLA+W, DATA received,
                    NACK returned, not addressed SLAVE mode
                    entered*/
     DATA_RECEIVED_LAST2 = I2DAT;
     AA = 1;        //wait for ACK next Master addressing
     break;
case 0x90:          /*90H, previous General Call, DATA received,
                    ACK returned*/
     DATA_RECEIVED3 = I2DAT;
     if (To_RX_Last_Data3)
       AA = 0;
     else
       AA = 1;
     break;
case 0x98:          /*98H, previous General Call, DATA received,
                    NACK returned, not addressed SLAVE mode
                    entered*/
     DATA_RECEIVED_LAST3 = I2DAT;
     AA = 1;
     break;
   //==========
   //Slave Mode
   //==========
case 0Xa0:          /*A0H, STOP or repeated START received while
                    still addressed SLAVE mode*/
     AA = 1;
     break;
   //======================
   //Slave Transmitter Mode
   //======================
case 0Xa8:             /*A8H, own SLA+R received, ACK returned*/
     I2DAT = NEXT_SEND_DATA3;
     AA = 1;        //when AA is "1", not last data to be
                 //transmitted
     break;
case 0Xb0:          /*B0H, arbitration lost in SLA+W/R
                    own SLA+R received, ACK returned */
     I2DAT = DUMMY_DATA;
     AA = 0;        //when AA is "0", last data to be
                 //transmitted
     STA = 1;          //retry to transmit START if bus free
     break;
case 0Xb8:          /*B8H, previous own SLA+R, DATA transmitted,
                    ACK received*/
     I2DAT = NEXT_SEND_DATA4;
     if (To_TX_Last_Data) //if last DATA will be transmitted
       AA = 0;
     else
       AA = 1;
     break;
case 0Xc0:          /*C0H, previous own SLA+R, DATA transmitted,
                    NACK received, not addressed SLAVE mode
                    entered*/
     AA = 1;
```

```
        break;
   case 0Xc8:           /*C8H, previous own SLA+R, last DATA trans-
                      mitted, ACK received, not addressed SLAVE
        AA = 1;         mode entered*/
        break;
   }//end of switch (I2STAT)

   SI = 0;            //SI should be the last command of I2C ISR
   while(STO);             //wait for STOP transmitted or bus error
                    //free, STO is cleared by hardware
}//end of I2C_ISR
```

6.14 12-bit Analog-to-digital Converter (ADC)

6.14.1 Overview

The ML51/ML54/ML56 Series is embedded with a 12-bit SAR ADC. The ADC (analog-to-digital converter) allows conversion of an analog input signal to a 12-bit binary representation of that signal. The ML51/ML54/ML56 Series is selected as 8-channel inputs in single end mode. The internal band-gap voltage 0.814 V also can be the internal ADC input. The analog input, multiplexed into one sample and hold circuit, charges a sample and hold capacitor. The output of the sample and hold capacitor is the input into the converter. The converter then generates a digital result of this analog level via successive approximation and stores the result in the result registers. The ADC controller also supports DMA (direct memory access) function for ADC continuous conversion and storage result data into XRAM no need special enable PDMA module.

6.14.2 Features

- Analog input voltage range: 0 ~ AV_{DD}.
- External or internal voltage reference input selectable.
- 12-bit resolution and 10-bit accuracy is guaranteed.
- Up to 16 single-end analog input channels
- 1 internal channels, they are band-gap voltage (VBG).
- Maximum ADC peripheral clock frequency is 1 MHz.
- Up to 500 KSPS sampling rate.
- Software Write 1 to ADCS bit to trig ADC start.
- External pin (STADC) trigger start
- PWM trigger start

6.14.3 Block Diagram

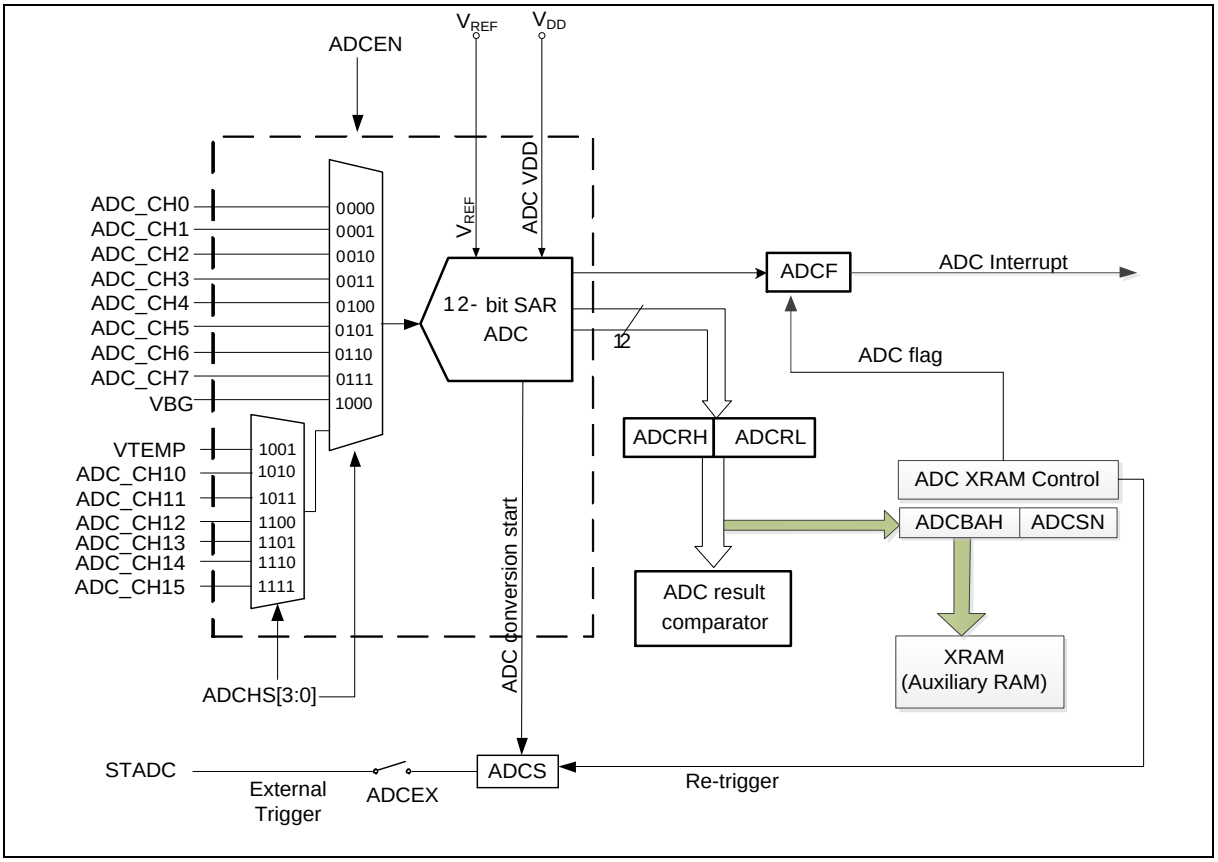


Figure 6.14-1 12-bit ADC Block Diagram

6.14.4 Functional Description

6.14.4.1 ADC Operation

Before ADC operation, the ADC circuit should be enabled by setting ADCEN (ADCCON1.0). This makes ADC circuit active. It consume extra power. Once ADC is not used, clearing ADCEN to turn off ADC circuit saves power.

The ADC analog input pin should be specially considered. ADCHS[2:0] are channel selection bits that control which channel is connected to the sample and hold circuit. User needs to configure selected ADC input pins as input-only (high impedance) mode via respective bits in PxMn registers. This configuration disconnects the digital output circuit of each selected ADC input pin. But the digital input circuit still works. Digital input may cause the input buffer to induce leakage current. To disable the digital input buffer, the respective bits in AINDIDS should be set. Configuration above makes selected ADC analog input pins pure analog inputs to allow external feeding of the analog voltage signals. Also, the ADC clock rate needs to be considered carefully. The ADC maximum clock frequency is listed in ADC Analog Electrical Characteristics. Clock above the maximum clock frequency degrades ADC performance unpredictably.

An A/D conversion is initiated by setting the ADCS bit (ADCCON0.6). When the conversion is complete, the hardware will clear ADCS automatically, set ADCF (ADCCON0.7) and generate an interrupt if enabled. The new conversion result will also be stored in ADCRH (most significant 8 bits) and ADCRL (least significant 4 bits). The 12-bit ADC result value is $4095 \times \frac{V_{AIN}}{V_{REF}}$

By the way, digital circuitry inside and outside the device generates noise which might affect the accuracy of ADC measurements. If conversion accuracy is critical, the noise level can be reduced by applying the following techniques:

1. Keep analog signal paths as short as possible. Make sure to run analog signals tracks well away from high-speed digital tracks.

2. Place the device in Idle mode during a conversion.

3. If any ADC_CHn pins are used as digital outputs, it is essential that these do not switch while a conversion is in progress.

6.14.4.2 ADC Conversion Triggered by External Source

Besides setting ADCS via software, the ML51/ML54/ML56 Series is enhanced by supporting hardware triggering method to start an A/D conversion. If ADCEX (ADCCON1.1) is set, edges or period points on selected PWM channel or edges of STADC pin will automatically trigger an A/D conversion. (The hardware trigger also sets ADCS by hardware.)

The effective condition is selected by ETGSEL (ADCCON0[5:4]) and ETGTYP (ADCCON1[3:2]). A trigger delay can also be inserted between external trigger point and A/D conversion. The external trigging ADC hardware with controllable trigger delay makes the ML51/ML54/ML56 Series feasible for high performance motor control. Note that during ADC is busy in converting (ADCS = 1), any conversion triggered by software or hardware will be ignored and there is no warning presented.

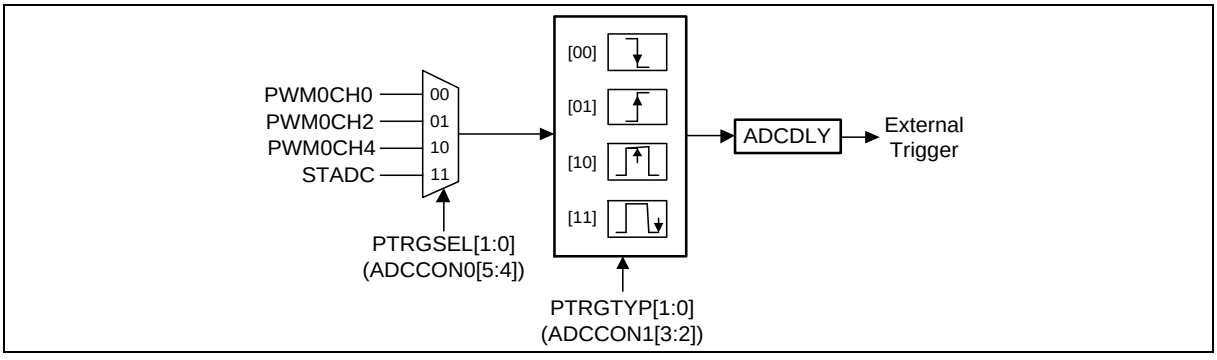


Figure 6.14-2 External Triggering ADC Circuit

6.14.4.3 ADC Conversion Result Comparator

The ML51/ML54/ML56 Series ADC has a digital comparator, which compares the A/D conversion result with a 12-bit constant value given in ACMPH and ACMPL registers. The ADC comparator is enabled by setting ADCMPEN (ADCCON2.5) and each compare will be done on every A/D conversion complete moment. ADCMPO (ADCCON2.4) shows the compare result according to its output polarity setting bit ADCMPOP (ADCCON2.6). The ADC comparing result can trigger a PWM Fault Brake output directly. This function is enabled when ADFBEN (ADCCON2.7). When ADCMPO is set, it generates a ADC compare event and asserts Fault Brake. Please also see Sector 18.1.5“Fault Brake”.

Note: After enabling the result compare function, the ADCF register changes to 1 only when ADC comparing result matches the condition and then enters interrupt vector if ADC interrupt is enabled. After this bit is enabled and ADC start is triggered, the ADC keeps converting. The register ADCRH and ADCRL value will change based on the result of ADC setting and can also be read out from the register. This process only stops after ADCF is set to 1.

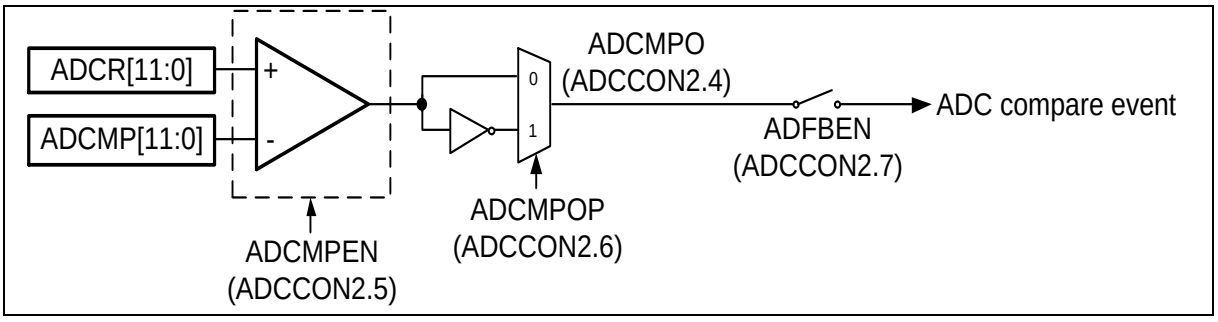


Figure 6.14-3 ADC Result Comparator

6.14.4.4 ADC Continues Conversion

The ADC controller supports DMA function, which auto store continues the A/D conversion result. The ADC DMA mode can store 12-bit ADC result into XRAM buffer, and 12-bit ADC data will auto-divide 8-bit high byte and 4-bit low nibble data two part. For reduce XRAM memory size, two 4-bit nibble data (continuing ADC conversion results) are automatically combine into one byte size and stored in XRAM.

The store method as below illustrate. It will store 8-bit of conversion high byte data first, then store combine data, the split point is ADC continue conversion length which define by ADCCN.

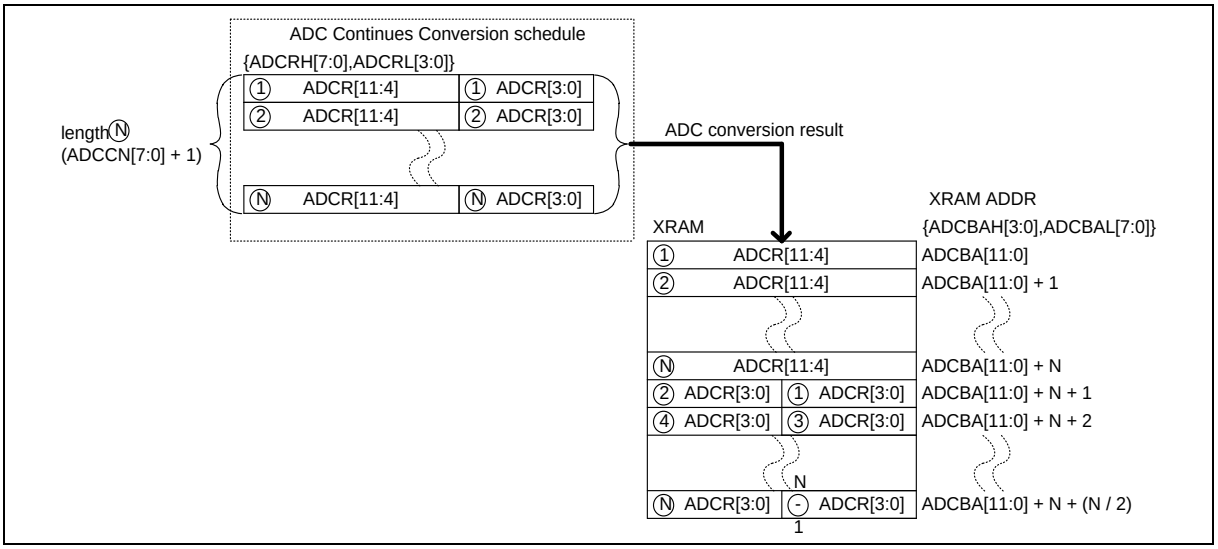


Figure 6.14-4 ADC Continues mode with DMA

A programing sequence is described below.

1 Set ADC channel and enable ADC as same as normal ADC setting method.

2 Set CONT (ADCCON1.4)to one for set ADC into continues conversion mode.

3 Set ADCBAH and ADCBAL registers to configure store address of conversion result.

4 Set ADCCN register to configure ADC conversion count.

5 Set HIE/FIE (ADCCON1[5]) to enable ADC conversion half done interrupt. (optional)

6 Start ADC RUN by software trigger (ADCS=1) or external trigger (ADCEX=1).

6.14.5 Register Description

ADCCON0 – ADC Control 0

Register	SFR Address	Reset Value
ADCCON0	A1H, Page 0	0000_0000b

7	6	5	4	3	2	1	0
ADCF	**ADCS**	**ETGSEL1**	**ETGSEL0**	**ADCHS3**	**ADCHS2**	**ADCHS1**	**ADCHS0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**ADCF**	**ADC Flag** This flag is set when an A/D conversion is completed in single sampling mode, final sampling complete in continue sampling mode or comparing hit if result comparator is enabled. The ADC result can be read. While this flag is 1, ADC cannot start a new converting. This bit is cleared by software.
[6]	**ADCS**	**A/D Converting Software Start Trigger** Setting this bit 1 triggers an A/D conversion. This bit remains logic 1 during A/D converting time and is automatically cleared via hardware right after conversion complete. The meaning of writing and reading ADCS bit is different. Writing: 0 = No effect. 1 = Start an A/D converting. Reading: 0 = ADC is in idle state. 1 = ADC is busy in converting.
[5:4]	**ETGSEL[1:0]**	**External Trigger Source Select** When ADCEX (ADCCON1.1) is set, these bits select which pin output triggers ADC conversion. 00 = PWM0CH0. 01 = PWM0CH2. 10 = PWM0CH4. 11 = STADC pin.

[3:0]	**ADCHS[3:0]**	**A/D Converting Channel Select** This filed selects the activating analog input source of ADC. If ADCEN is 0, all inputs are disconnected. 0000 = ADC_CH0 0001 = ADC_CH1. 0010 = ADC_CH2. 0011 = ADC_CH3. 0100 = ADC_CH4. 0101 = ADC_CH5. 0110 = ADC_CH6. 0111 = ADC_CH7. 1000 = VBG (Internal band-gap voltage 1.22V). 1001 = VTEMP. (Temperature Sensor). 1010 = ADC_CH10. 1011 = ADC_CH11. 1100 = ADC_CH12. 1101 = ADC_CH13. 1110 = ADC_CH14. 1111 = ADC_CH15.

ADCCON1 – ADC Control 1

Register	SFR Address	Reset Value
ADCCON1	E1H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	HIE	CONT	ETGTYP[1:0]		ADCEX	ADCEN
-	-	R/W	R/W	R/W		R/W	R/W

Bit	Name	Description
[7:6]	-	Reserved
[5]	HIE	**ADC Half Done Interrupt Enable** 0 = ADC interrupt is not set while half of A/D conversions are complete in continue mode 1 = ADC interrupt is set while half of A/D conversions are complete in continue mode
[4]	CONT	**ADC Continue Sampling Select** 0 = ADC single sampling, ADC interrupt is set while an A/D conversion is completed 1 = ADC continue sampling. ADC interrupt is set while total A/D conversions are completed
[3:2]	ETGTYP[1:0]	**External Trigger Type Select** When ADCEX (ADCCON1.1) is set, these bits select which condition triggers ADC conversion. 00 = Falling edge on PWM0/2/4 or STADC pin. 01 = Rising edge on PWM0/2/4 or STADC pin. 10 = Central point of a PWM period. 11 = End point of a PWM period. Note that the central point interrupt or the period point interrupt is only available for PWM center-aligned type.
[1]	ADCEX	**ADC External Conversion Trigger Select** This bit to select the methods of triggering an A/D conversion. 0 = A/D conversion is started only via setting ADCS bit. 1 = A/D conversion is started via setting ADCS bit or by external trigger source depending on ETGSEL[1:0] and ETGTYP[1:0]. Note that while ADCS is 1 (busy in converting), the ADC will ignore the following external trigger until ADCS is hardware cleared.
[0]	ADCEN	**ADC Enable** 0 = ADC circuit off. 1 = ADC circuit on.

ADCCON2 – ADC Control 2

Register	SFR Address	Reset Value
ADCCON2	E2H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
ADFBEN	ADCMPOP	ADCMPEN	ADCMPO	ADCAQT[2:0]			ADCDLY.8
R/W	R/W	R/W	R	R/W			R/W

Bit	Name	Description
[7]	**ADFBEN**	**ADC Compare Result Asserting Fault Brake Enable** 0 = ADC asserting Fault Brake Disabled. 1 = ADC asserting Fault Brake Enabled. Fault Brake is asserted once its compare result ADCMPO is 1. Meanwhile, PWM channels output Fault Brake data. PWMRUN (PWMnCON0.7) will also be automatically cleared by hardware. The PWM output resumes when PWMRUN is set again.
[6]	**ADCMPOP**	**ADC Comparator Output Polarity** 0 = ADCMPO is 1 if ADCR[11:0] is greater than or equal to ADCMP[11:0]. 1 = ADCMPO is 1 if ADCR[11:0] is less than ADCMP[11:0].
[5]	**ADCMPEN**	**ADC Result Comparator Enable.** ADC result comparator to trig ADCF enable bit. Only when comparator value match the condition of ADC compare value defined ADCF will be set to 1. This condition base on ADCMPH, ADCMPL and ADCMPOP register define. The ADCF register changes to 1 only when ADC comparing result matches the condition and then enters interrupt vector if ADC interrupt is enabled. 0 = ADC result comparator trig ADCF Disabled. 1 = ADC result comparator trig ADCF Enabled. **Note:** After this bit is enabled and ADC start is triggered, the ADC keeps converting. The register ADCRH and ADCRL value will change based on the result of ADC setting and can also be read out from the register. This process only stops after ADCF is set to 1
[4]	**ADCMPO**	**ADC Comparator Output Value** This bit is the output value of ADC result comparator based on the setting of ACMPOP. This bit updates after every A/D conversion complete.
[3:1]	**ADCAQT[2:0]**	**ADC Acquisition Time** This 3-bit field decides the acquisition time for ADC sampling, following by equation below: ADC acquisition time = $\frac{4*ADCAQT+10}{F_{ADC}}$. F_{ADC} base on ADCAQTDIV setting The default and minimum acquisition time is 10 ADC clock cycles. Note that this field should not be changed when ADC is in converting.
[0]	**ADCDLY.8**	**ADC External Trigger Delay Counter Bit 8** See ADCDLY register.

ADCDLY – ADC Trigger Delay Counter

Register	SFR Address	Reset Value
ADCDLY	E3H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
ADCDLY[7:0]							
R/W							

Bit	Name	Description
[7:0]	**ADCDLY[7:0]**	**ADC External Trigger Delay Counter Low Byte** This 8-bit field combined with ADCCON2.0 forms a 9-bit counter. This counter inserts a delay after detecting the external trigger. An A/D converting starts after this period of delay. External trigger delay time = $\frac{ADCDLY}{F_{ADC}}$. Note that this field is valid only when ADCEX (ADCCON1.1) is set. User should not modify ADCDLY during PWM run time if selecting PWM output as the external ADC trigger source.

AINDIDS0 – ADC Channel Digital Input Disconnect

Register	SFR Address	Reset Value
AINDIDS0	CEH, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
AIN7DIDS	**AIN6DIDS**	**AIN5DIDS**	**AIN4DIDS**	**AIN3DIDS**	**AIN2DIDS**	**AIN1DIDS**	**AIN0DIDS**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:0]	**AINnDIDS**	**ADC Channel Digital Input Disable** 0 = Enabled digital input at ADC channel n. 1 = Disabled digital input at ADC channel n . ADC channel n is read always 0.

AINDIDS1 – ADC Channel Digital Input Disconnect

Register	SFR Address	Reset Value
AINDIDS1	CEH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
AIN15DIDS	**AIN14DIDS**	**AIN13DIDS**	**AIN12DIDS**	**AIN11DIDS**	**AIN10DIDS**	**-**	**-**
R/W	R/W	R/W	R/W	R/W	R/W		

Bit	Name	Description
[7:0]	**AINnDIDS**	**ADC Channel Digital Input Disable** 0 = Enabled digital input at ADC channel n. 1 = Disabled digital input at ADC channel n . ADC channel n is read always 0.

ADCRH – ADC Result High Byte

Register	SFR Address	Reset Value
ADCRH	C3H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
ADCR[11:4]							
R							

Bit	Name	Description
[7:0]	**ADCR[11:4]**	**ADC Result High Byte** The most significant 8 bits of the ADC result stored in this register.

ADCRL – ADC Result Low Byte

Register	SFR Address	Reset Value
ADCRL	C2H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	ADCR[3:0]			
-	-	-	-	R			

Bit	Name	Description
[7:4]	-	Reserved
[3:0]	ADCR[3:0]	**ADC Result Low Byte** The least significant 4 bits of the ADC result stored in this register.

ADCMPH – ADC Compare High Byte

Register	SFR Address	Reset Value
ADCMPH	CFH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
ADCMP[11:4]							
W/R							

Bit	Name	Description
[7:0]	**ADCMP[11:4]**	**ADC Compare High Byte** The most significant 8 bits of the ADC compare value stores in this register.

ADCMPL – ADC Compare Low Byte

Register	SFR Address	Reset Value
ADCMPL	CEH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	ADCMP[3:0]			
-	-	-	-	W/R			

Bit	Name	Description
[7:4]	-	Reserved
[3:0]	**ADCMP[3:0]**	**ADC Compare Low Byte** The least significant 4 bits of the ADC compare value stores in this register.

ADCBAH – ADC RAM Base Address High Byte

Register	SFR Address	Reset Value
ADCBAH	E4H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
-				ADCBA[3:0]			
-				R/W			

Bit	Name	Description
[7:4]	-	Reserved
[3:0]	**ADCBA[3:0]**	**ADC RAM Base Address (High Byte)** The most significant 4 bits of RAM base address to store ADC continue sampling data. RAM base address ADCBA[11:0] = {ADCBAH[3:0], ADCBAL[7:0]}

ADCBAL – ADC RAM Base Address Low Byte

Register	SFR Address	Reset Value
ADCBAL	CBH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
ADCBA[7:0]							
R/W							

Address: CBH, Page:0 Reset value: 0000 0000b

Bit	Name	Description
[7:0]	**ADCBA[7:0]**	**ADC RAM Base Address (Low Byte)** The least significant 8 bits of RAM base address to store ADC continue sampling data. RAM base address ADCBA[11:0] = { ADCBAH[3:0], ADCBAL[7:0]}

ADCSN – ADC Sampling Number

Register	SFR Address	Reset Value
ADCSN	E5H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
ADCSN[7:0]							
R/W							

Bit	Name	Description
[7:0]	**ADCSN[7:0]**	**ADC Sampling Number** The total sampling numbers for ADC continue sampling select. Total sampling number= ADCSN[7:0] + 1

ADCCN – ADC Current Sampling Number

Register	SFR Address	Reset Value
ADCCN	E6H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
ADCCN[7:0]							
R							

Bit	Name	Description
[7:0]	**ADCCN[7:0]**	**ADC Current Sampling Number** The current sampling numbers for ADC continue sampling select. The current sampling number= ADCCN[7:0] + 1

ADCSR – ADC Status Register

Register	SFR Address	Reset Value
ADCSR	E7H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
-	ADCDIV[2:0]			-	CMPHIT	HDONE	FDONE
0	R/W			-	R/W	R/W	R/W

Bit	Name	Description
[7]	-	**This bit should be keep 0.**
[6:4]	ADCDIV[2:0]	**ADC Clock Divider (This Clock only for ADCAQT and ADCDLY)** 000 = F_{ADC} is F_{SYS}/1. 001 = F_{ADC} is F_{SYS} /2. 010 = F_{ADC} is F_{SYS} /4. 011 = F_{ADC} is F_{SYS} /8. 100 = F_{ADC} is F_{SYS} /16. 101 = F_{ADC} is F_{SYS} /32. 110 = F_{ADC} is F_{SYS} /64. 111 = F_{ADC} is F_{SYS} /128.
[3]	-	Reserved
[2]	CMPHIT	**ADC Comparator Hit Flag** This bit is set by hardware when ADCMPO (ADCCON2.4) flag rising **Note:** This bit can be cleared by writing 0 to it.
[1]	HDONE	**A/D Conversion Half Done Flag** This bit is set by hardware when half of ADCSN A/D conversions are complete in continue mode. **Note:** This bit can be cleared by writing 0 to it
[0]	FDONE	**A/D Conversion Full Done Flag** This bit is set by hardware when all of ADCSN A/D conversions are complete in continue mode or single conversion in single mode. **Note:** This bit can be cleared by writing 0 to it..

6.15 Voltage Reference (V_{REF})

6.15.1 External Voltage Reference

The V_{REF} pin is for analog multiplexer, such as ADC, ACMP. It default be used as an external source(set ENVRF = 0).

6.15.2 Internal Volrage Reference

It also could be configurable as on-chip reference voltage generator (V_{REF_IN}) by setting ENVRF = 1 (see Figure 6.15-1 VREF Block Diagram). The output voltage is selectable by setting VRFSEL[2:0]. The maximum load of the V_{REF_IN} must be less than 200 uA to AV_{SS}. Set pre-load is to reduce stable time of V_{REF_IN}. At first enable V_{REF_IN} and turn on pre-load at the same time, the minimum stable time of pre-load on the V_{REF_IN} must be greater than 3 ms. After the V_{REF_IN} stable, user should be turn off pre-load to avoid any interference on analog multiplexer. Pre-load is only for internal V_{REF} use. For detailed electrical characteristics, refer to the Table36-10. Internal V_{REF}.

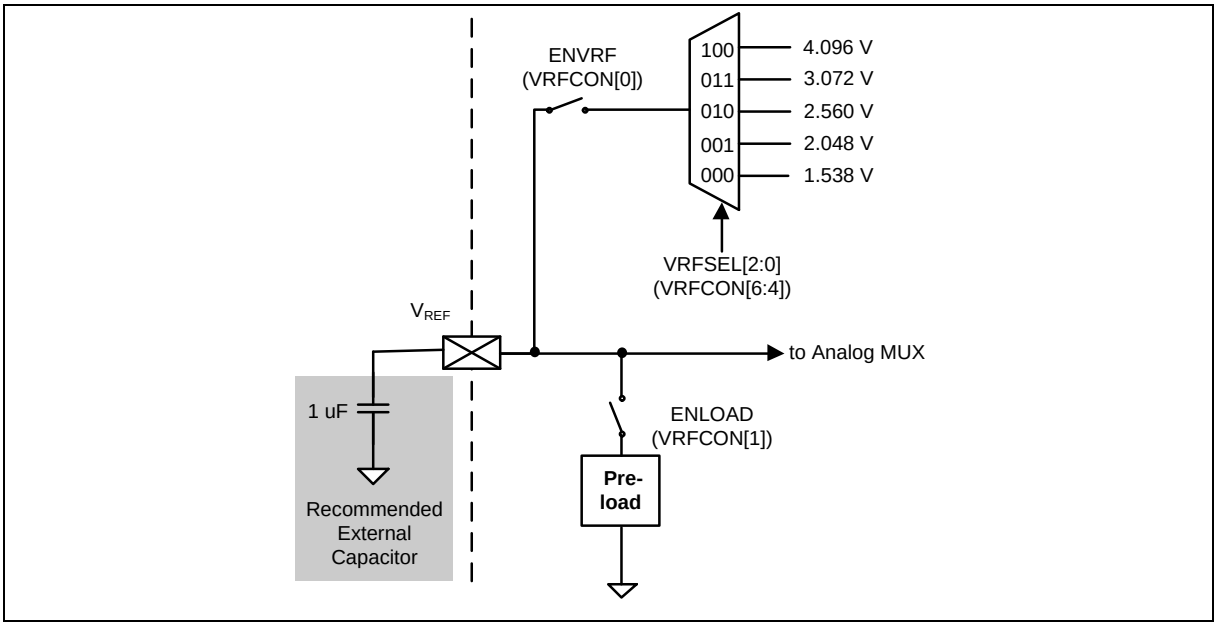


Figure 6.15-1 V_{REF} Block Diagram

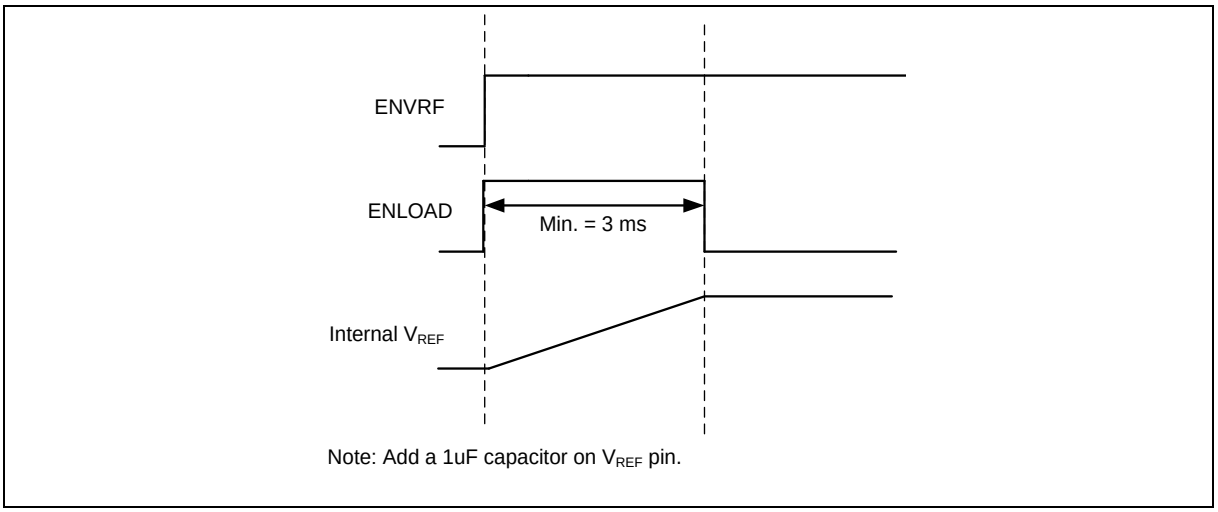


Figure 6.15-2 Pre-load Timing

VRFCON – Internal V_{REF} Control (TA Protected)

Register	SFR Address	Reset Value
VRFCON	A9H,Page 1, TA protected	0000_0000 b

7	6	5	4	3	2	1	0
-	VRFSEL[2:0]			-	-	ENLOAD	ENVRF
-	R/W			-	-	R/W	R/W

Bit	Name	Description
[7]	-	Reserved
[6:4]	VRFSEL[2:0]	**Internal V_{REF} Output Voltage Select** This field selects V_{REF} output voltage. 000 = 1.538V , when V_{DD} > 2.0V 001 = 2.048V , when V_{DD} > 2.4V 010 = 2.560V , when V_{DD} > 2.9V 011 = 3.072V , when V_{DD} > 3.4V 100 = 4.096V , when V_{DD} > 4.5V 101 = reserved 110 = reserved 111 = reserved
[3:2]	-	Reserved
[1]	ENLOAD	**Internal V_{REF} Pre-Load Enable** 1 = Internal V_{REF} Pre-load Enabled. 0 = Internal V_{REF} Pre-load Disabled
[0]	ENVRF	**Internal V_{REF} Enable** 1 = Internal V_{REF} Enabled, 0 = Internal V_{REF} Disabled Note that a 1 uF has to add on V_{REF} pin while internal V_{REF} is enabled.

6.16 Analog Comparator Controller (ACMP)

6.16.1 Overview

The ML51/ML54/ML56 Series contains two comparators. The comparator output is logic 1 when positive input is greater than negative input; otherwise, the output is 0. The comparator can be configured to generate an interrupt when the comparator output value changes.

6.16.2 Feature

- Analog input voltage range: 0 ~ AV_{DD}(voltage of AV_{DD} pin)
- Supports hysteresis function
- Supports wake-up function
- Selectable input sources of negative input
- Comparator ACMP0 supports
  - 4 positive source
    - P2.5 (ACMPn_P0)
    - P2.3 (ACMPn_P1)
    - P2.1 (ACMPn_P2)
    - P3.1 (ACMPn_P3)
  - 4 negative sources
    - P2.4 (ACMP0_N0)
    - Comparator Reference Voltage (CRV)
    - VBG (BAND-GAP voltage)
    - P2.0 (ACMP0_N1)
- Comparator ACMP1 supports
  - 4 positive source
    - P2.5 (ACMPn_P0)
    - P2.3 (ACMPn_P1)
    - P2.1 (ACMPn_P2)
    - P3.1 (ACMPn_P3)
  - 4 negative sources
    - P2.2 (ACMP1_N0)
    - Comparator Reference Voltage (CRV)
    - VBG (BAND-GAP voltage)
    - P3.2 (ACMP1_N1)

6.16.3 Block Diagram

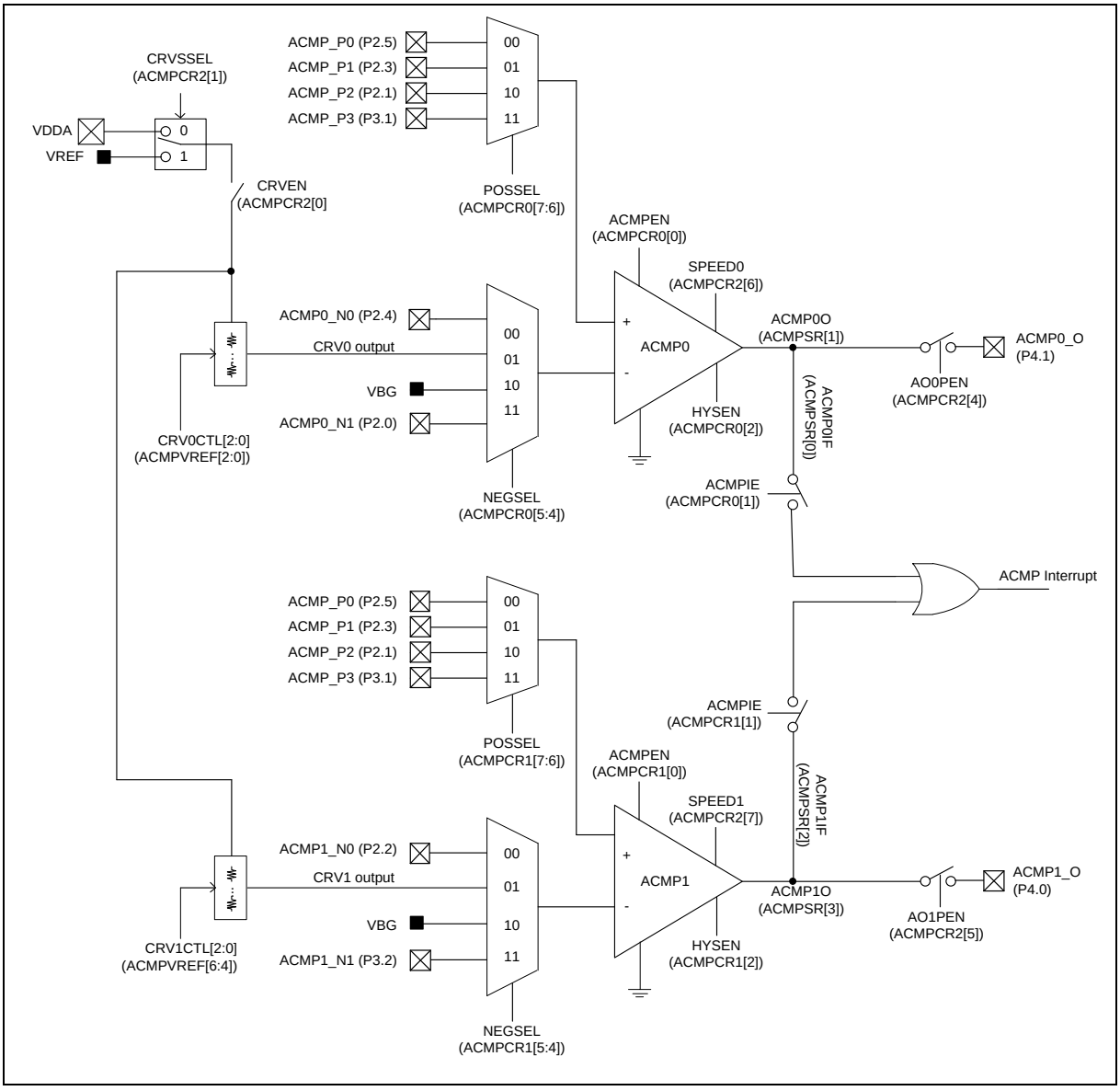


Figure 6.16-1 Analog Comparator Block Diagram

6.16.4 Functional Description

6.16.4.1 Hysteresis Function

The analog comparator provides the hysteresis function to make the comparator to have a stable output transition. If comparator output is 0, it will not be changed to 1 until the positive input voltage exceeds the negative input voltage by a high threshold voltage. Similarly, if comparator output is 1, it will not be changed to 0 until the positive input voltage drops below the negative input voltage by a low threshold voltage.

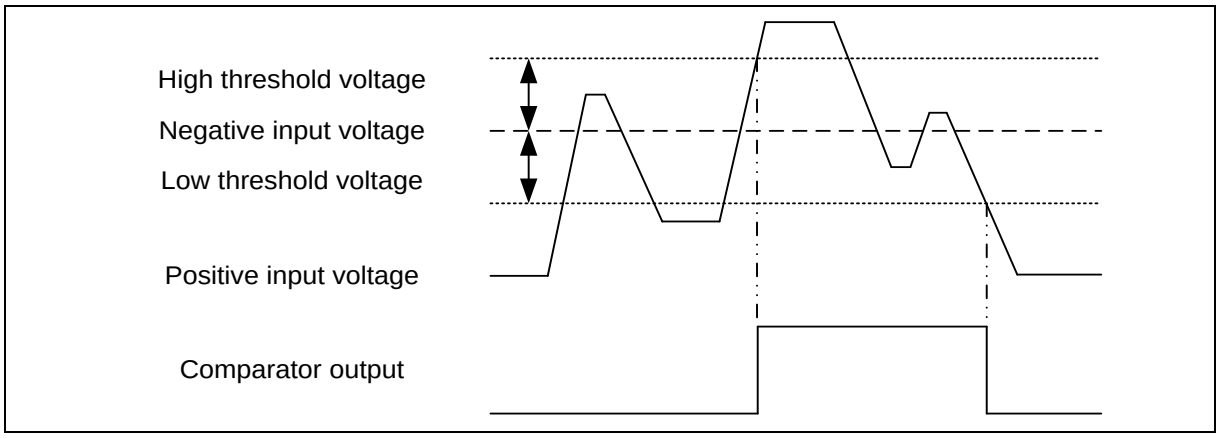


Figure 6.16-2 Comparator Hysteresis Function

6.16.4.2 Comparator Reference Voltage (CRV)

The comparator reference voltage (CRV) module is responsible for generating reference voltage for comparators. The CRV module consists of resisters ladder and analog switch. User can set the CRV output voltage by setting the CRVnCTL($ACMPV_{REF}$). The CRV output voltage can be selected as the negative input of comparator by setting NEGSEL (ACMPCR0[5:4]).

Features:

1. User selectable references voltage source by setting the CRVSSEL(ACMPCR2[1]) register.
2. User selectable references voltage by setting the CRVnCTL($ACMPV_{REF}$) register.

Comparator reference voltage = V_{IN} * (1/6+CRVnCTL/12); V_{IN} = AV_{DD} or $V_{REF.}$

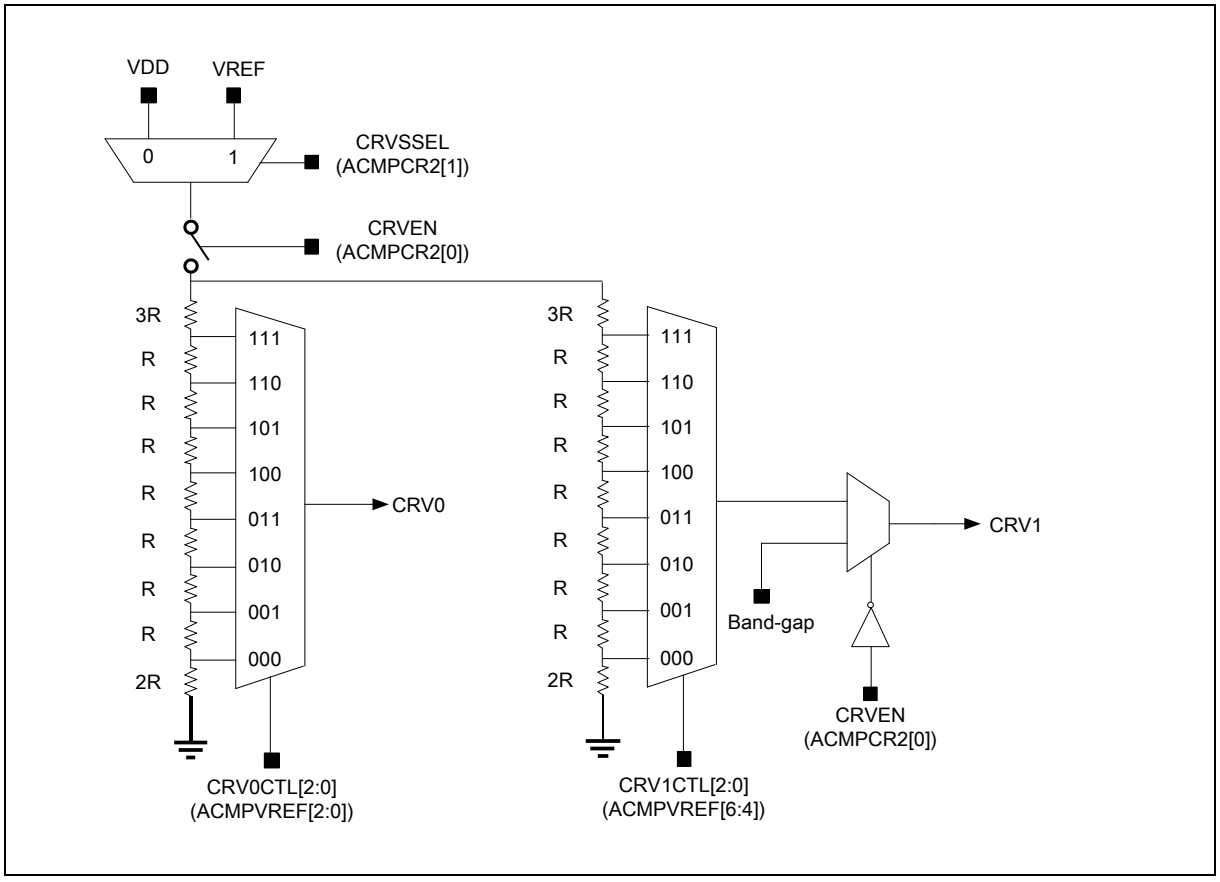


Figure 6.16-3 Comparator Reference Voltage Block Diagram

Note that If CRVEN = 0, CRV0 is equal to 0 and CRV1 is equal to Band-gap.

6.16.4.3 Interrupt Sources

The comparator generates an output ACMPnO (ACMPSR). If the ACMPIE (ACMPCR0[1]) bit in ACMPCR0 is set, a state change on the comparator output ACMPnO (ACMPSR) will cause comparator flag ACMPnIF (ACMPSR) be set and the comparator interrupt requested. User can write 1 to ACMPnIF (ACMPSR) through software to stop interrupt request.

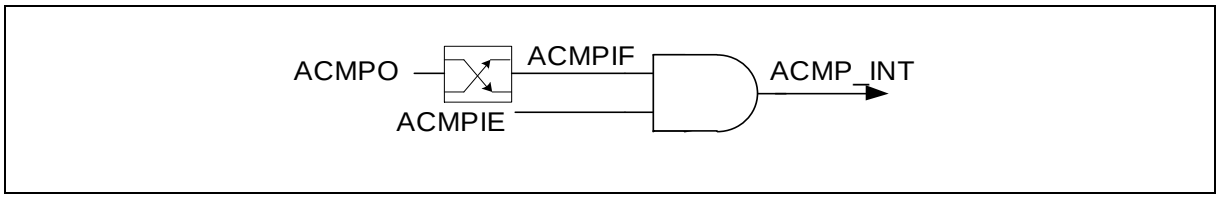


Figure 6.16-4 Analog Comparator Interrupt Sources

6.16.5 Register Description

ACMPCR0 – Analog Comparator Control Register 0

Register	SFR Address	Reset Value
ACMPCR0	D2H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
POSSEL		NEGSEL		WKEN	HYSEN	ACMPIE	ACMPEN
R/W		R/W		R/W	R/W	R/W	R/W

Bit	Name	Description
[7:6]	**POSSEL**	**Comparator 0 Positive Input Selection** 00 = ACMP0_P0 (P2.5) pin. 01 = ACMP0_P1 (P2.3) pin. 10 = ACMP0_P2 (P2.1) pin. 11 = ACMP0_P3 (P3.1) pin.
[5:4]	**NEGSEL**	**Comparator 0 Negative Input Selection** 00 = ACMP0_N0 (P2.4) pin. 01 = Internal comparator reference voltage (CRV). 10 = VBG (Band-gap). 11 = ACMP0_N1 (P2.0)pin.
[3]	**WKEN**	**Comparator 0 Power-Down Wake-Up Enable Bit** 0 = Comparator 0 Wake-up function Disabled. 1 = Comparator 0 Wake-up function Enabled.
[2]	**HYSEN**	**Comparator 0 Hysteresis Enable Bit** 0 = Comparator 0 hysteresis Disabled. 1 = Comparator 0 hysteresis Enabled.
[1]	**ACMPIE**	**Comparator 0 Interrupt Enable Bit** 0 = Comparator 0 interrupt Disabled. 1 = Comparator 0 interrupt Enabled. If WKEN (ACMPCR1[3]) is set to 1, the wake-up interrupt function will be enabled as well.
[0]	**ACMPEN**	**Comparator 0 Enable Bit** 0 = Comparator 0 Disabled. 1 = Comparator 0 Enabled.

ACMPCR1 – Analog Comparator Control Register 1

Register	SFR Address	Reset Value
ACMPCR1	D3H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
POSSEL		NEGSEL		WKEN	HYSEN	ACMPIE	ACMPEN
R/W		R/W		R/W	R/W	R/W	R/W

Bit	Name	Description
[7:6]	**POSSEL**	**Comparator 1 Positive Input Selection** 00 = ACMP1_P0 (P2.5) pin. 01 = ACMP1_P1 (P2.3) pin. 10 = ACMP1_P2 (P2.1) pin. 11 = ACMP1_P3 (P3.1) pin.
[5:4]	**NEGSEL**	**Comparator 1 Negative Input Selection** 00 = ACMP1_N0 (P2.2) pin. 01 = Internal comparator reference voltage (CRV). 10 = VBG (Band-gap). 11 = ACMP1_N1 (P3.2)pin.
[3]	**WKEN**	**Comparator 1 Power-Down Wake-Up Enable Bit** 0 = Comparator 1 Wake-up function Disabled. 1 = Comparator 1 Wake-up function Enabled.
[2]	**HYSEN**	**Comparator 1 Hysteresis Enable Bit** 0 = Comparator 1 hysteresis Disabled. 1 = Comparator 1 hysteresis Enabled.
[1]	**ACMPIE**	**Comparator 1 Interrupt Enable Bit** 0 = Comparator 1 interrupt Disabled. 1 = Comparator 1 interrupt Enabled. If WKEN (ACMPCR2[3]) is set to 1, the wake-up interrupt function will be enabled as well.
[0]	**ACMPEN**	**Comparator 1 Enable Bit** 0 = Comparator 1 Disabled. 1 = Comparator 1 Enabled.

ACMPCR2 – Analog Comparator Control Register 2

Register	SFR Address	Reset Value
ACMPCR2	ABH, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
SPEED1		POE1	POE0	SPEED0		CRVSSEL	CRVEN
R/W		R/W	R/W	R/W		R/W	R/W

Bit	Name	Description
[7:6]	**SPEED1**	**Analog Comparator 1 Speed Control** 00 = slow speed, propagation delay : 4.5us, 1.2uA (typ.) 01 = slow+ speed, propagation delay : 2.0us, 3uA (typ.) 10 = fast speed, propagation delay : 0.6us, 10uA (typ.) 11 = fast+ speed, propagation delay : 0.2us, 75uA (typ.)
[5]	**POE1**	**Analog Comparator 1 Polarity Output Enable** 0 = ACMP1 output directly. 1 = ACMP1 output inversely.
[4]	**POE0**	**Analog Comparator 0 Polarity Output Enable** 0 = ACMP0 outputs directly. 1 = ACMP0 outputs inversely.
[3:2]	**SPEED0**	**Analog Comparator 0 Speed Control** 00 = slow speed, propagation delay : 4.5us, 1.2uA (typ.) 01 = slow+ speed, propagation delay : 2.0us, 3uA (typ.) 10 = fast speed, propagation delay : 0.6us, 10uA (typ.) 11 = fast+ speed, propagation delay : 0.2us, 75uA (typ.)
[1]	**CRVSSEL**	**CRV Source Voltage Selection** 0 = V_{DD} is selected as CRV source voltage. 1 = The reference voltage (V_{REF}) is selected as CRV source voltage.
[0]	**CRVEN**	**CRV Enable Bit** 0 = CRV Disabled. 1 = CRV Enabled.

ACMPSR – Analog Comparator Status Register

Register	SFR Address	Reset Value
ACMPSR	D4H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
-				ACMP1O	ACMP1IF	ACMP0O	ACMP0IF
-				R	R/W	R	R/W

Bit	Name	Description
[7:4]	-	Reserved
[3]	ACMP1O	**Comparator 1 Output** Synchronized to the PCLK to allow reading by software. Cleared when the comparator 1 is disabled, i.e. ACMPEN (ACMPCR1[0]) is cleared to 0. **Note:** This bit is read only.
[2]	ACMP1IF	**Comparator 1 Interrupt Flag** This bit is set by hardware whenever the comparator 1 output changes state. This will generate an interrupt if ACMPIE (ACMPCR1[1]) is set to 1 **Note:** Write “0” to clear this bit to 0.
[1]	ACMP0O	**Comparator 0 Output** Synchronized to the PCLK to allow reading by software. Cleared when the comparator 0 is disabled, i.e. ACMPEN (ACMPCR0[0]) is cleared to 0. **Note:** This bit is read only.
[0]	ACMP0IF	**Comparator 0 Interrupt Flag** This bit is set by hardware whenever the comparator 0 output changes state. This will generate an interrupt if ACMPIE (ACMPCR0[1]) is set to 1 **Note:** Write “0” to clear this bit to 0.

ACMPV$_{REF}$ – ACMP Reference Voltage Control Register

Register	SFR Address	Reset Value
ACMPV$_{REF}$	D5H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
-	CRV1CTL[2:0]			-	CRV0CTL[2:0]		
-	R/W			-	R/W		

Bit	Name	Description
[7]	-	Reserved
[6:4]	**CRV1CTL[2:0]**	**Comparator 1 Reference Voltage Setting** CRV1 = CRV source voltage * (2/12+CRV1CTL/12).
[3]	-	Reserved
[2:0]	**CRV0CTL[2:0]**	**Comparator 0 Reference Voltage Setting** CRV0 = CRV source voltage * (2/12+CRV0CTL/12).

6.17 PDMA Controller (PDMA)

6.17.1 Overview

The ML51/ML54/ML56 Series provides peripheral direct memory access (PDMA) controller. The PDMA controller is used to provide high-speed data transfer between memory and peripherals or between memory and memory. The PDMA controller can transfer data from one address to another without CPU intervention. This has the benefit of reducing the workload of CPU and keeps CPU resources free for other applications.

6.17.2 Feature

- Supports transfer data width of 8 bits
- Supports software and SPI and SMC/UART request
- Supports source and destination address increment size can be byte
- Supports transfer done and half done interrupt
- Supports using PDMA to write data to perform CRC operation

6.17.3 Block Diagram

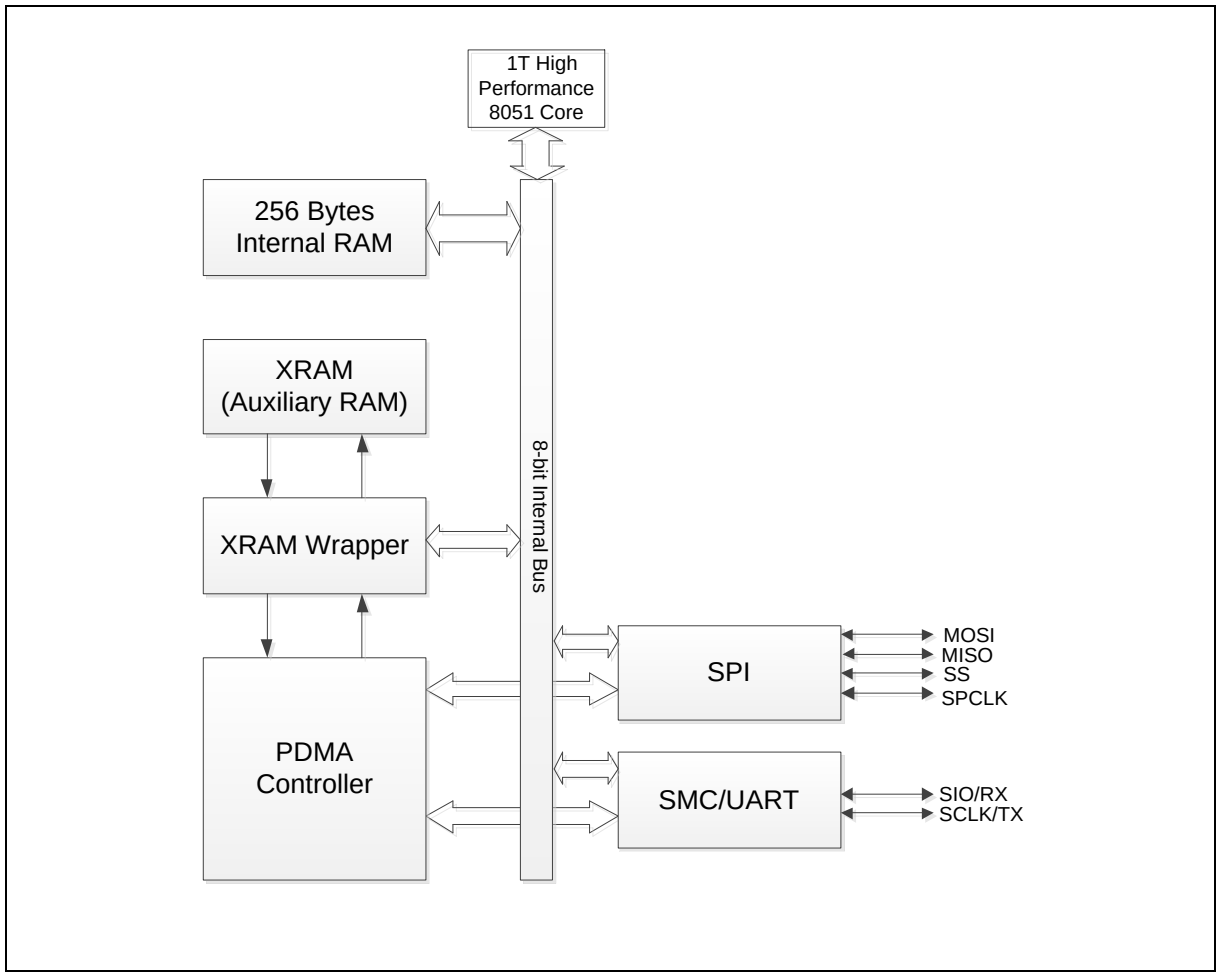


Figure 6.17-1 PDMA Interface Diagram

6.17.4 Functional Description

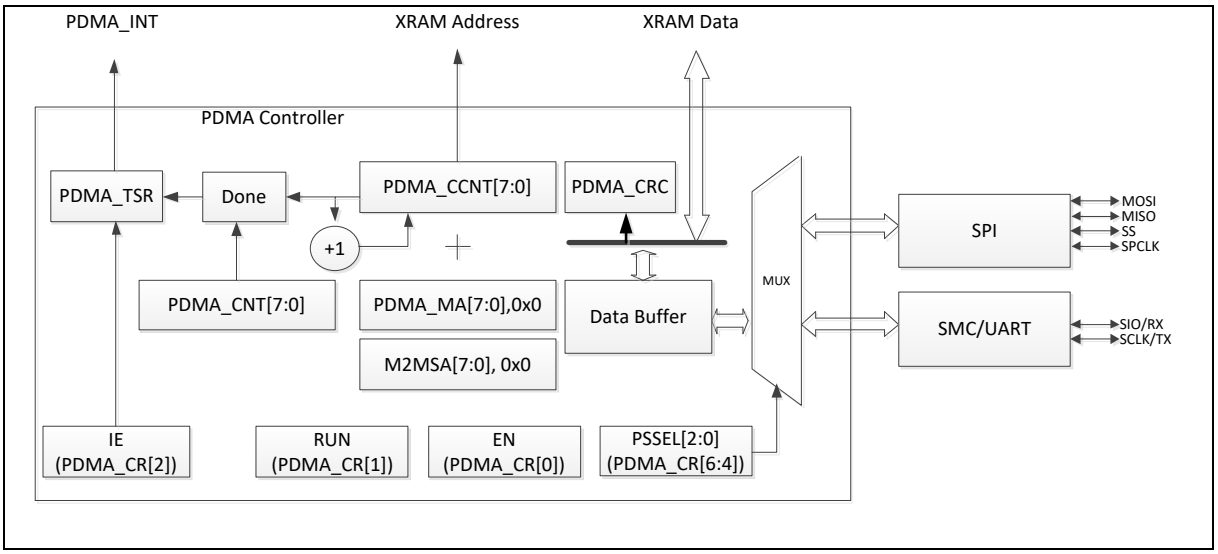


Figure 6.17-2 PDMA Controller Block Diagram

6.17.4.1 Operating Modes

Each PDMA channel behavior is not pre-defined, user must configure the channel service settings of PSSEL[3:0] registers before starting the related PDMA channel operation.

User must set EN DMAnCR[0] bit to enable PDMA channel. Then write a valid source address to the DMAnMA and DMAnBAH[3:0] register, a destination address to the MTMnDA and DMAnBAH[7:4]register if use memory to memory, and a transfer count to the DMAnCNT register. Next, trigger the RUN DMAnCR[1]. If the source address and destination are not in wrap around mode, the PDMA will continue the transfer until DMAnCCNT counts down to 0. In wrap around mode, when DMAnCCNT counts down to 0, the PDMA will reload DMAnCCNT and work around until user clears EN DMAnCR[0] bit to disable PDMA channel.

A programing sequence example is described below.

SPI peripheral to XRAM memory

1. Configure DMAnCR register to set EN DMAnCR[0] bit to enable PDMA channel.
2. Set PSSEL[3:0] = 0001 SPI0 RX (, 0011 SPI1 RX, 0101 SPI0 TX or 0111 SPI1 TX) (DMAnCR (n-1~2)) register to configure the channel service setting.
3. Set DMAnMA/DMAnBAH[3:0] registers to configure destination address.
4. Set DMAnCNT register to configure PDMA transfer count.
5. Set HIE/FIE DMAnCR[3]/[2] to enable PDMA transfer done interrupt (optional) then enable global interrupt.
6. Set RUN DMAnCR[0] bit to enable PDMA transfer.
7. Write “0” to HDONE and FDONE DMAnTSR[1][0] bit to clear interrupt flag if PDMA transfer done interrupt is generated.
8. Set RUN DMAnCR[0] bit to enable next PDMA transfer.
9. If an error occurs during the PDMA operation, the channel operation stops until user clears the error condition, , and then clears the EN DMAnCR[0] bit to disable the PDMA channel, then sets EN DMAnCR[0] bit and RUN DMAnCR[1] bit to start operation again.

SMC/UART peripheral to XRAM memory

1. Configure DMAnCR register to set EN DMAnCR[0] bit to enable PDMA channel.
2. Set PSSEL[3:0] = 0010 SMC/UART RX (or 0110 SMC/UART TX) (DMAnCR (n-1~2)) register to configure the channel service setting.
3. Set DMAnMA/DMAnBAH[3:0] registers to configure destination address.
4. Set DMAnCNT register to configure PDMA transfer count.
5. Set HIE/FIE DMAnCR[3]/[2] to enable PDMA transfer done interrupt (optional) then enable global interrupt.
6. Set RUN DMAnCR[0] bit to enable PDMA transfer.
7. Write “0” to HDONE and FDONE DMAnTSR[1][0] bit to clear interrupt flag if PDMA transfer done interrupt is generated.
8. Set RUN DMAnCR[0] bit to enable next PDMA transfer.
9. If an error occurs during the PDMA operation, the channel operation stops until user clears the error condition, , and then clears the EN DMAnCR[0] bit to disable the PDMA channel, then sets EN DMAnCR[0] bit and RUN DMAnCR[1] bit to start operation again.

Memory to Memory (XRAM) Transfer

1. Configure DMAnCR register to set EN DMAnCR[0] bit to enable PDMA channel.
2. Set PSSEL[3:0] = 0000 (XRAM to XRAM) (DMAnCR (n-1~2)) register to configure the channel service setting.
3. Set DMAnMA/DMAnBAH[3:0] registers to configure source address.
4. Set DMAnDA/DMAnBAH[7:4] registers to configure destination address.
5. Set DMAnCNT register to configure PDMA transfer count.
6. Set HIE/FIE DMAnCR[3]/[2] to enable PDMA transfer done interrupt (optional) then enable global interrupt.
7. Set RUN DMAnCR[0] bit to enable PDMA transfer.
8. Write “0” to HDONE and FDONE DMAnTSR[1][0] bit to clear interrupt flag if PDMA transfer done interrupt is generated.
9. Set RUN DMAnCR[0] bit to enable next PDMA transfer.
10. If an error occurs during the PDMA operation, the channel operation stops until user clears the error condition and then clears the EN DMAnCR[0] bit to disable the PDMA channel, then sets EN DMAnCR[0] bit and RUN DMAnCR[1] bit to start operation again.

6.17.4.2 CRC-8 Function for PDMA

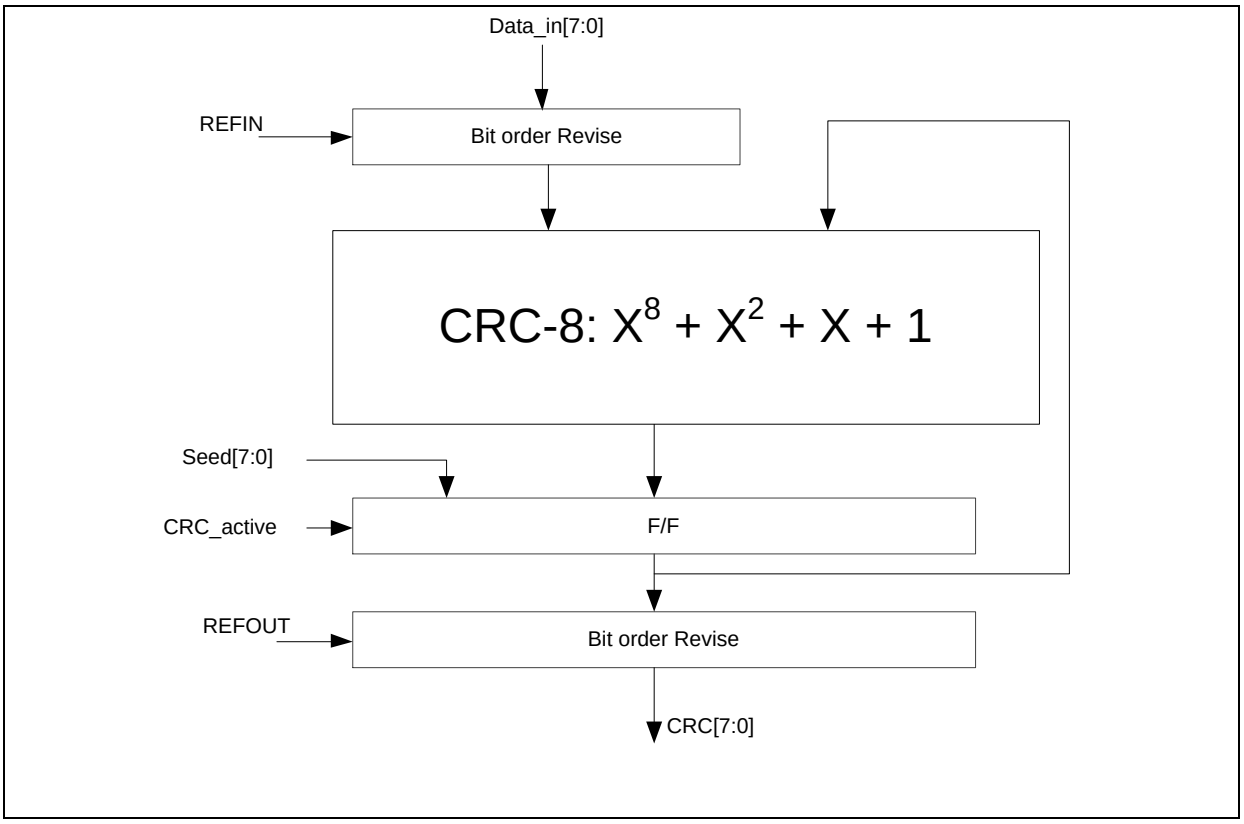


Figure 6.17-3 CRC-8 Block Diagram

6.17.5 Register Description

DMAnCR – PDMAn Control Register

Register	SFR Address	Reset Value
DMA0CR0	92H, Page 0	0000_0000 b
DMA1CR0	EBH, Page 0	0000_0000 b
DMA2CR0	B3H, Page 2	0000_0000 b
DMA3CR0	ABH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PSSEL[3:0]				HIE	FIE	RUN	EN
R/W				R/W	R/W	R/W	R/W

Bit	Name	Description
[7:4]	**PSSEL[3:0]**	**Peripheral Source Select** 0000 = XRAM to XRAM 0001 = SPI0 RX 0010 = SMC0/UART2 RX. 0011 = SPI1 RX 0100 = Reserved, No peripheral source select 0101 = SPI0 TX 0110 = SMC0/UART2 TX. 0111 = SPI1 TX 1010 = SMC1/UART3 RX. 1110 = SMC1/UART3 TX. The others are reserved, no peripheral source selected **Note:** 0001~0011,1010 : peripheral devices to XRAM memory 0101~0111,1110 : XRAM memory to peripheral devices
[3]	**HIE**	**PDMA HALFTransfer Done Interrupt Enable Bit** 0 = Interrupt Disabled when PDMA half transfer is done. 1 = Interrupt Enabled when PDMA half transfer is done.
[2]	**FIE**	**PDMA Full Transfer Done Interrupt Enable Bit** 0 = Interrupt Disabled when PDMA full transfer is done. 1 = Interrupt Enabled when PDMA full transfer is done.
[1]	**RUN**	**Trigger Enable Bit** 0 = No effect. 1 = PDMA data transfer Enabled. **Note 1:** When PDMA transfer completed, this bit will be cleared automatically.
[0]	**EN**	**PDMA Enable Bit** Setting this bit to 1 enables PDMA operation. If this bit is cleared, PDMA will ignore all PDMA request and Reset the internal state machine, pointers and internal buffer. The contents of all Register Description will not be cleared.

DMAnMAL – PDMA XRAM Base Address Low Byte

Register	SFR Address	Reset Value
DMA0MAL	93H, Page 0	0000_0000 b
DMA1MAL	ECH, Page 0	0000_0000 b
DMA2MAL	B4H, Page 2	0000_0000 b
DMA3MAL	ACH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
MAL[7:0]							
R/W							

Bit	Name	Description
[7:0]	**MAL[7:0]**	**PDMA XRAM Base Address (Low Byte)** The least significant 8 bits of XRAM address to store or read for the peripheral source data; in memory to memory transfer, this register is the source address. XRAM address = {MAH[3:0],MAL[7:0]}

DMAnBAH – PDMAn XRAM Base Address and Memory to Memory Destination Address High Byte

Register	SFR Address	Reset Value
DMA0BAH	F6H, Page 0	0000_0000 b
DMA1BAH	FDH, Page 0	0000_0000 b
DMA2BAH	B2H, Page 2	0000_0000 b
DMA3BAH	AAH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
MTMDA[7:4]				XRAMA[7:4]			
R/W				R/W			

Bit	Name	Description
[7:4]	**MTMDA[7:4]**	**Memory to Memory Destination Address (High Byte)** The most significant 4 bits of XRAM address are used for memory to memory destination address. XRAM destination address = {MDAH[3:0], MDAL[7:0]}
[3:0]	**XRAMA[7:4]**	**PDMA XRAM Base Address (High Byte)** The most significant 4 bits of XRAM address to store or read for the peripheral source data; in memory to memory transfer, this register is the destination address. XRAM address = {MAH[3:0], MAL[7:0]}

DMAnCNT – PDMA Transfer Count

Register	SFR Address	Reset Value
DMA0CNT	94H, Page 0	0000_0000 b
DMA1CNT	EDH, Page 0	0000_0000 b
DMA2CNT	B5H, Page 2	0000_0000 b
DMA3CNT	ADH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
DMAnCNT[7:0]							
R/W							

Bit	Name	Description
[7:0]	**DMAnCNT[7:0]**	**PDMA Transfer Count** The total transfer count for PDMA request operation. Total transfer count = CNT[7:0] + 1

DMAnCCNT – PDMA Current Transfer Count

Register	SFR Address	Reset Value
DMA0CCNT	95H, Page 0	0000_0000 b
DMA1CCNT	EEH, Page 0	0000_0000 b
DMA2CCNT	B6H, Page 2	0000_0000 b
DMA3CCNT	AEH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
DMAnCCNT[7:0]							
R							

Bit	Name	Description
[7:0]	**DMAnCCNT[7:0]**	**PDMA Current Transfer Count** The current transfer count for PDMA request operation. Current transfer count = CCNT[7:0] **Note:** while DMAnCNT=0xFF (total transfer count = 256) and DMAnCCNT = 0x00 , If PDMA FDONE flag (DMAnTSR[0])=0, that means, 1'st byte data is not complete.If PDMA FDONE flag (DMAnTSR[0])=1, that means, all of data are transferred..

DMAnTSR – PDMAn Transfer Status Register

Register	SFR Address	Reset Value
DMA0TSR	E9H, Page 0	0000_0000 b
DMA1TSR	F1H, Page 0	0000_0000 b
DMA2TSR	B1H, Page 2	0000_0000 b
DMA3TSR	A9H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
-					ACT	HDONE	FDONE
-					R	R/W	R/W

Bit	Name	Description
[7:3]	-	Reserved
[2]	ACT	**PDMA in Active Status Flag (Read Only)** 0 = This bit is cleared automatically when PDMA transfer is done or disabled. 1 = This bit is set by hardware when PDMA transfer is in active.
[1]	HDONE	**PDMA Half Transfer Done Flag** This bit is set by hardware when PDMA half transfer is done. **Note:** This bit can be cleared by writing 0 to it.
[0]	FDONE	**PDMA Full Transfer Done Flag** This bit is set by hardware when PDMA full transfer is done. **Note:** This bit can be cleared by writing 0 to it.

MTMnDA – Memory to Memory Destination Address Low Byte

Register	SFR Address	Reset Value
MTM0DA	EAH, Page 0	0000_0000 b
MTM1DA	F2H, Page 0	0000_0000 b
MTM2DA	B7H, Page 2	0000_0000 b
MTM3DA	AFH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
MTMnDA[7:0]							
R/W							

Bit	Name	Description
[7:0]	**MTMnDA[7:0]**	**Memory to Memory Destination Address (Low Byte)** The least significant 8 bits of XRAM address are used for memory to memory destination address. XRAM destination address = {MDAH[3:0], MDAL[7:0]}

DMAnCR1 – PDMAn Control 1 Register

Register	SFR Address	Reset Value
DMA0CR1	8AH, Page 3	0000_0000 b
DMA1CR1	8BH, Page 3	0000_0000 b
DMA2CR1	8CH, Page 3	0000_0000 b
DMA3CR1	8DH, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	XOROUT	REFOUT	REFIN	CRCEN
-	-	-	-	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:4]	-	Reserved
[3]	XOROUT	**PDMA CRC OUT Reflect Enable Bit** 0 = CRC OUT exclusive-ored Disabled when PDMA is running. 1 = CRC OUT exclusive-ored Enabled when PDMA is running, the final value is exclusive-ored with 0x55
[2]	REFOUT	**PDMA CRC OUT Reflect Enable Bit** 0 = CRC OUT reflect Disabled when PDMA is running. 1 = CRC OUT reflect Enabled when PDMA is running, the output data will be bit order revised
[1]	REFIN	**PDMA CRC IN Reflect Enable Bit** 0 = CRC IN reflect Disabled when PDMA is running. 1 = CRC IN reflect Enabled when PDMA is running, the input data will be bit order revised
[0]	CRCEN	**PDMA CRC Checksum Enable Bit** 0 = CRC checksum Disabled when PDMA is running, DMAnCRC[7:0] is set to 0x00 1 = CRC checksum Enabled when PDMA is running.

DMAnCRC – PDMA CRC Checksum

Register	SFR Address	Reset Value
DMA0CRC	92H, Page 3	0000_0000 b
DMA1CRC	93H, Page 3	0000_0000 b
DMA2CRC	94H, Page 3	0000_0000 b
DMA3CRC	95H, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
CRC[7:0]							
R/W							

Bit	Name	Description
[7:0]	**CRC[7:0]**	**PDMA CRC Checksum** The checksum of the Cyclic Redundancy Check (CRC-8) calculation The CRC-8 polynomial is below CRC-8: X8 + X2 + X + 1

DMAnSEED – PDMA CRC SEED

Register	SFR Address	Reset Value
DMA0SEED	9AH, Page 3	0000_0000 b
DMA1SEED	9BH, Page 3	0000_0000 b
DMA2SEED	9CH, Page 3	0000_0000 b
DMA3SEED	9DH, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
SEED[7:0]							
R/W							

Bit	Name	Description
[7:0]	**SEED[7:0]**	**PDMA CRC SEED** The seed of the Cyclic Redundancy Check (CRC-8) calculation The CRC-8 polynomial is below CRC-8: $X^8 + X^2 + X + 1$

Note :

	XOROUT	REFOUT	REFIN	SEED
CRC-8	0	0	0	0x00
CRC-8/ITU	1	0	0	0x00
CRC-8/ROHC	0	1	1	0xFF

6.18 LCD Driver

6.18.1 Overview

The Liquid Crystal Displays (LCD) panel is widely used to meet the display need in applications. The ML54/ML56 series is equipped with LCD driver that can directly drive the LCD panel with 4 COM x 32 SEG , 6 COM x 30 SEG or 8 COM x 28 SEG. Use the corresponding COM and SEM according to the definition of multiple function pin. The LCD driver supports 1/4 duty, 1/6 duty, or 1/8 duty. The driving voltage supports 1/2 bias, 1/3 bias or 1/4 bias with waveform type A or Type B. The source of LCD clock is based on the choice of LIRC or LXT. The LCD display can keep display on or off during chip in power-down mode. The LCD power supply VLCD source is selectable from internal charge pump, external VLCD pin or analog power AV_{DD}.

6.18.2 Features

- 1.8V to 5.5V LCD operating voltage.
- Selectable LCD clock source from LIRC or LXT
- 1/2, 1/3, 1/4 bias selectable
- Maximum 4 COM x 32 SE.G. 6 COM x 30 SE.G. 8 COM x 28 SEG
- Supports buffer mode for high current driving
- Support enhaced resistor mode for low power application
- Support external VLCD source or AV_{DD} as LCD voltage source.
- Support programmable internal charge pump circuit for LCD voltage level is higher or lower than V_{DD} application.
- Support blink function.
- Suppot display on or off during chip in Power-down mode

6.18.3 Block Diagram

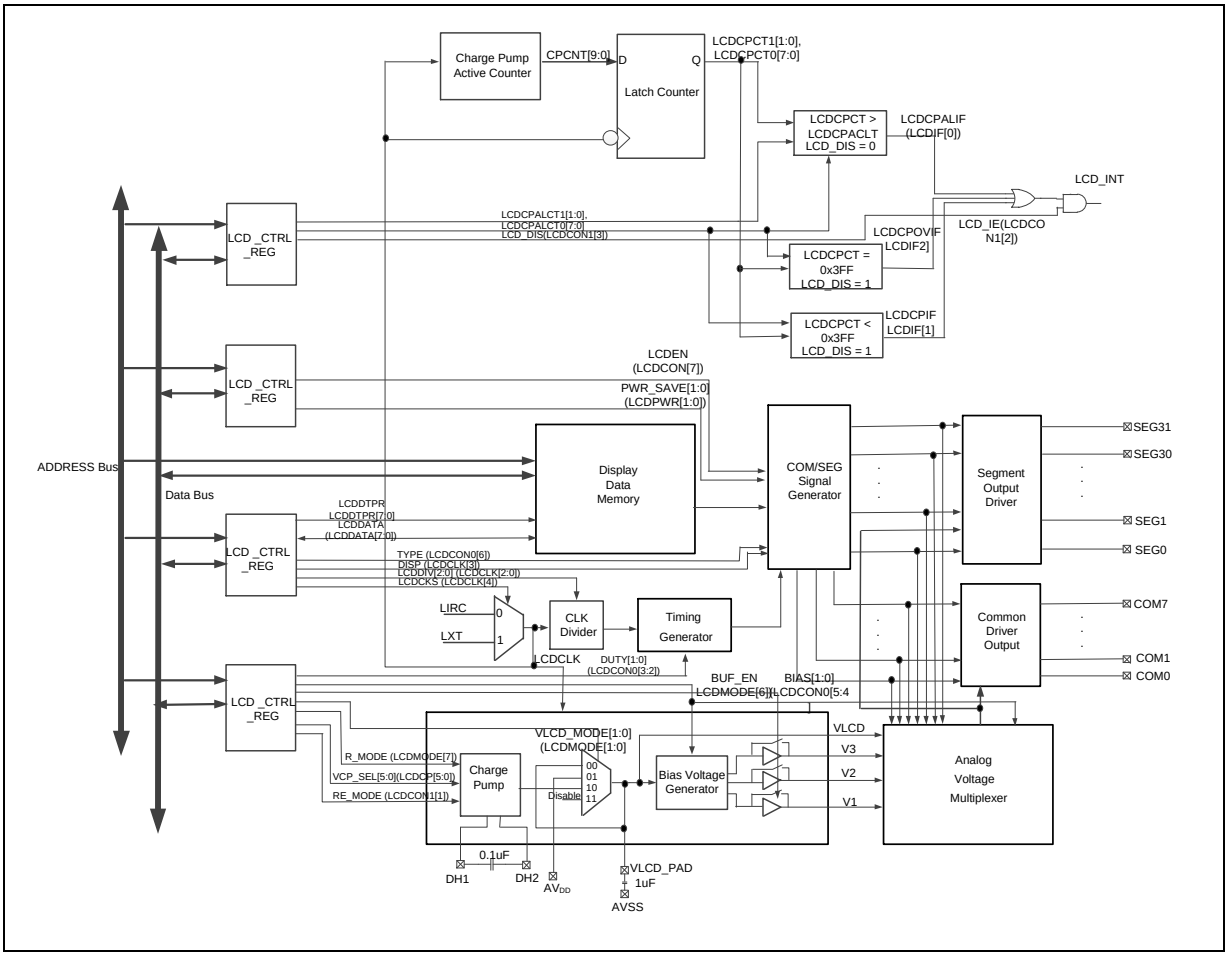


Figure 6.18-1 LCD Block Diagram

6.18.4 Functional Description

6.18.4.1 Control Logic

Setting LCDEN (LCDCON.7) as 1 turns on the LCD circuit. If LCDEN is enabled, COM pins and SEG pins driver signals and display the LCD panel according to the registers. The duty and bias can be setting by DUTY[1:0] (LCDCON[3:2]) and BIAS[1:0] (LCDCON[5:4]) bits individually. **Note:** that user should not change DUTY and BIAS settings while the LCD driver is enabled, else the output waveform will be unpredictable and may lead to cause a DC-component for one LCD frame.

The LCD clock source is selected by LCDCKS (LCDCLK[4]) and its frequency is configured by LCDDIV[2:0] (LCDCLK[2:0]). It is important to select the correct frame rate for the LCD display. Normally, the frame rate is recommended to be 30 Hz to 100 Hz. Less than 30Hz frame rate may introduce flickering, while over 100Hz frame rate may lead ghosting and unnecessary high power consumption. Giving a proper frame rate according to the LCD panel requirement obtains a good display quality. The LCD frequency frame rate follows equations below:

$F_{LCD} = \frac{F_{LIRC}\ OR\ F_{LXT}}{2^4} \times\ LCDDIV[2:0]\ \times Duty$ and F_{LIRC} or F_{LXT} is base on LCDCKS defined

COM	4	6	8
DUTY	$\frac{1}{4}$	$\frac{1}{6}$	$\frac{1}{8}$

When 1/4, 1/6 or 1/8 duty is selected that means 4 COM, 6 COM or 8 COM pins are used, and the SEG pins will be used according to the definition of multiple function pin setting. When those SEG pins were used as COM pins, these SEG bits are unavailable.

Each COM pin signal corresponds to its LCDDAT register. Once LCD data bit is enabled in the registers are transferred synchronously with the F_{LCD} and in turn automatically generate the necessary LCD driving signals to each COM pin without programming control. When the bit value is 1, the corresponding LCD pixel is darkened, when the bit value is 0, the pixel is cleared.

There are 32 segment for the LCD data storage by LCDPTR determines which SEG can be accessed. Each time defined in LCDPTR byte value controls this segment number pin cross with different COM pins. That means only one segment and most 8 corresponding common can be modified status at one time. To display LCD panel, needs to be written the proper data into the LCDDAT after LCDPTR is given.

The LCD pin is defined by Multi-function pin register. For example, one pin can be define as common or segment, the output waveform is based on MFP defined. The unused common or segment pin can be allocated as GPIO. Once the MFP define the common or segment pin as GPIO, even giving data intot relation SEG or COM reigester, GPIO is not going to output the waverform.

Figure 6.18-2. LCD Register Map shows some of the pixel enabled when setting in diffenent status. Please note that not all segment show in this figure.

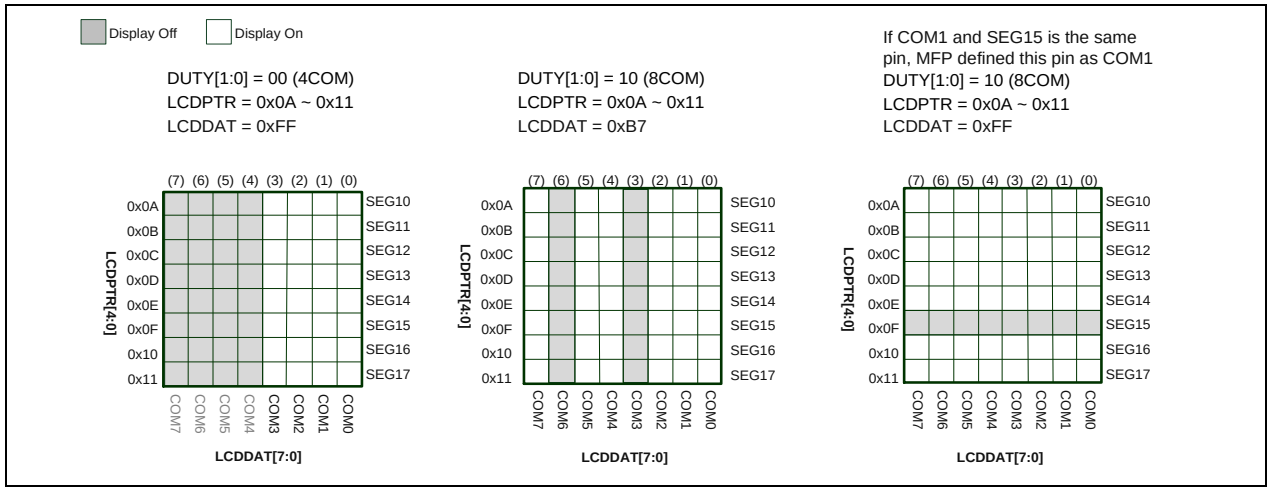


Figure 6.18-2. LCD Register Map Example

The LCD must only be driven with AC voltages. A DC voltage deteriorates the liquid crystal fluid and hence cannot be energized. Therefore, DC voltage applied to LCD electrodes may harm and destroy the LCD. The LCD driver waveforms are designed to create 0-Vdd potential across all LCD segments. ML51/ML54/ML56 Series can drive two types of the LCD waverforms which call TYPE A and TYPE B by setting LCDCON0[6] bit

If the segment waveform equals an inverted common waveform, then the segment is in ON state.Following Figure 6.18-3 One Frame of LCD Energized shows the two type of the LCD driver one COM and one SEG waveform ON / OFF state and the enegized zone.

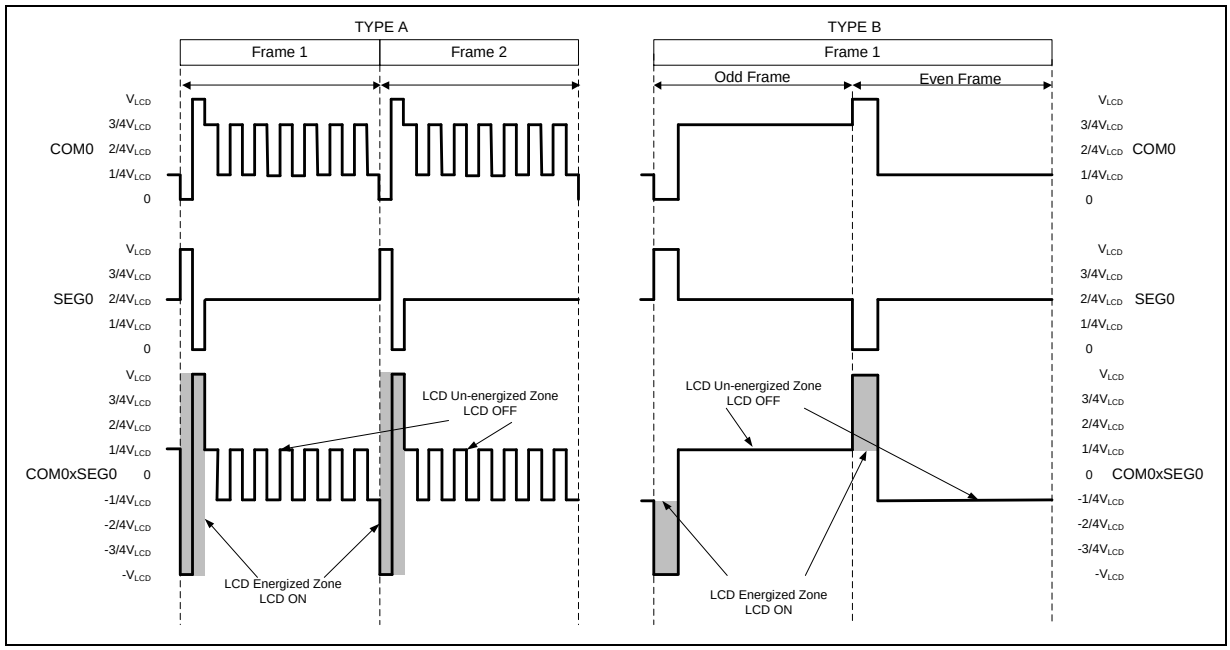


Figure 6.18-3 One Frame of LCD Energized

The accompanying timing diagrams depict the display driver signals generated by the microcontroller for various values of bias. Diagrams show the default 1/8 duty waveforms for illustration.

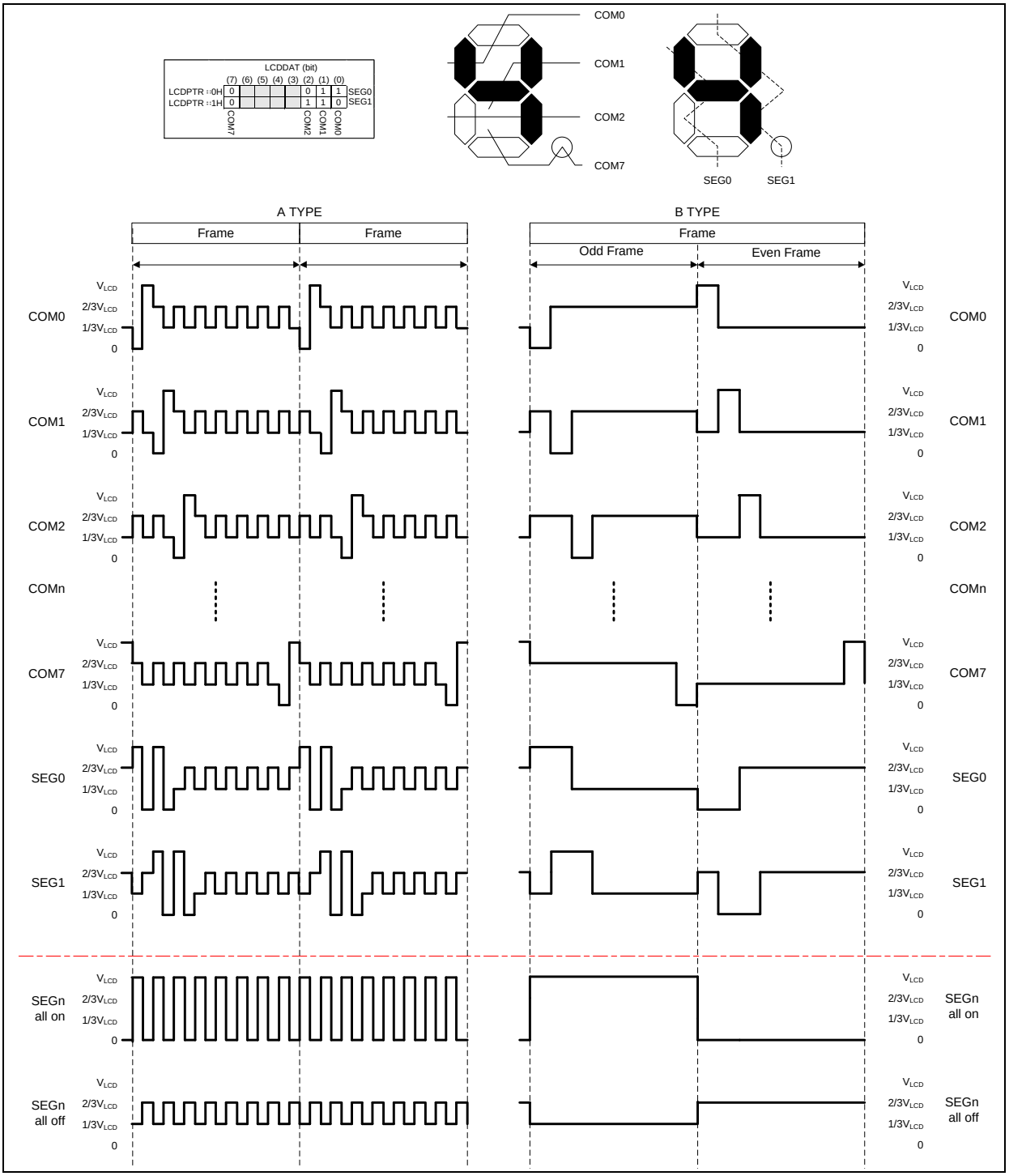


Figure 6.18-4. Example of Type A and Type B 8 COM and SEG Driving Signals of 1/3 Bias

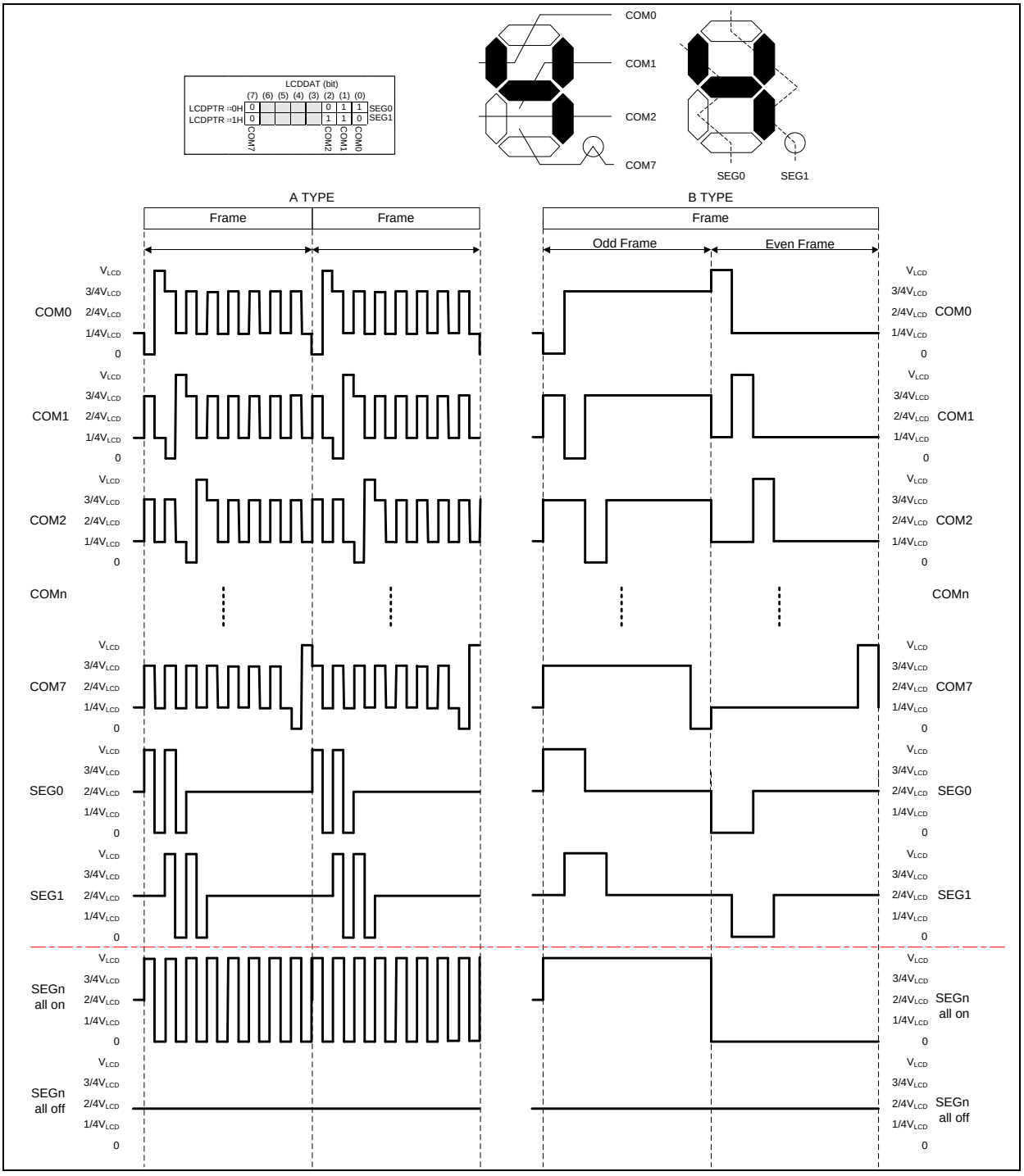


Figure 6.18-5. Example of Type A and Type B 8 COM and SEG Driving Signals of 1/4 Bias

6.18.4.2 LCD Driving Voltage Select

Followoing table shows the three types of LCD voltage source and how to select.

	VLCD_MODE[1:0] (LCDMODE[1:0])	V_{LCD} Range	VLCD / LCD_DH1 / LCD_DH2 pin	Note
External VLCD	00	1.8 ~ 5.5 V	VLCD pin connect with external voltage LCD_DH1 and LCD_DH2 can be used as normal GPIO	$V_{LCD} \neq V_{DD}$ is acceptable
AV_{DD}	01	1.8 ~ 3.6 V	VLCD pin floating or connect 0.1uF capactor to V_{SS} LCD_DH1 and LCD_DH2 can be used as normal GPIO	$V_{LCD} = V_{DD}$
Internal Charge pump	10	2.8 ~ 5.4 V	VLCD pin must connect a capactor to V_{SS}. The remomend value is 1uF. a capactor must connect between LCD_DH1 and LCD_DH2 pin. The remomend value is 0.1uF.	$V_{LCD} \neq V_{DD}$ is acceptable **LCD pump up voltage with limitation** **$V_{LCD(max.)} = 1.8*V_{DD}$.** LCD pump down voltage without this limitation.
Disable	11	-	-	

Table 6.18-1 VLCD Source Selection Table

6.18.4.3 LCD Driving Current Mode Select And Power Consumption

Foillowing table shows all LCD driving mode. Table also list the LCD driving current and the relation power consumption.

LCD Driving Mode	Control Register				LCD Driving Curruent[1]	Power Consumption[2]
	R_MODE (LCDMODE[7])	RE_MODE (LCDCON1 [1])	BUF_MODE (LCDMODE[6])	PWR_SAVING [1:0] (LCDPWR)		
Buffer Mode	0	0	1	00	Level 1 (Max.)	Level 3
Resistor Mode	1	0	0	00	Level 2	Level 1 (Max.)
Buffer Power Saving Mode	0	0	1	01/10/11	Level 3	Level 4
Resistor Power Saving Mode	1	0	0	01/10/11	Level 4	Level 2
Resistor Enhance Mode	1	1	0	-	Level 5(Min.)	Level 5 (Min.)
Note 1: Level 1 means the LCD driving current is the largest in these modes. **Note 2:** Level 1 means the current of LCD driver module power consumption is the largest in these modes.						

Table 6.18-2 LCD Driving Mode Regiter Setting

6.18.4.4 LCD Interrupt

There are Three LCD flags, LCDCPIF , LCDCPOVIF and LCDCPALIF. ALL of them can generate an LCD event interrupt requests. If LCD interrupt Enable bit LCDIE and interrupt source select pin LCDIS is select. After EA as 1, CPU will execute the LCD interrupt service routine once any of these three flags is set. User needs to check flags to determine what event caused the interrupt. Both of I^2C flags are cleared by software.

Following table shows the interrupt event condition

	Control Register		Interrupt FLAG	Interrupt Condition
	LCDIE LCDCON0[5]	LCDIS LCDCON0[6]		
LCD Charge Pump Alarm Couter Reach Interrupt	1	0	LCDCPALIF	When LCD charge pump counter is over LCDCPALCT0 and LCDCPALCT1 setting value LCD interrupt will be happen.
LCD Charge Pump Voltage Reach Interrupt	1	1	LCDCPIF	When LCD charge pump voltage reach LCDCPUMPdefine value. LCD interrupt wil be happen, and charge pump counter value will be storage in LCDCPCT0 and LCDCPCT1.
LCD Charge Pump Voltage Overflow Interrupt	1	1	LCDCPOVIF	When LCD charge pump voltage can note reach LCDCPUMPdefine value and the charge pump counter value overflow the maximaun 0x3FF. LCD interrupt wil be happen

6.18.5 Register Description

LCDCON – LCD Control

Register	SFR Address	Reset Value
LCDCON	F9H, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
LCDEN	**TYPE**	**BIAS[1:0]**		**DUTY[1:0]**		**-**	**-**
R/W	R/W	R/W		R/W		-	-

Bit	Name	Description
[7]	**LCDEN**	**LCD Enable** 0 = LCD circuit OFF. Each COM and SEG pin functions as general purpose I/O and its multi-functions other than LCD. 1 = LCD circuit ON. COM and enabled SEG pins generate the LCD driving waveform.
[6]	**TYPE**	**Display Type** 0 = Type A 1 = Type B (Power saving mode)
[5:4]	**BIAS[1:0]**	**LCD Bias** 00 = Reserved. 01 = 1/2 bias. 10 = 1/3 bias. 11 = 1/4 bias
[3:2]	**DUTY[1:0]**	**LCD Duty** 00 = 1/4 duty. 01 = 1/6 duty. 10 = 1/8 duty. 11 = Reserved. Note that when 1/4 duty is selected, only COM0 to COM3 are used for LCD driving. When 1/6 or 1/8 duty is selected that means 6 COM or 8 COM pins are used, and the SEG pins will be used according to the definition of multiple function pin setting. When those SEG pins were used as COM pins, these SEG bits are unavailable.
[1:0]	**-**	Reserved

LCDCLK – LCD Clock Control

Register	SFR Address	Reset Value
LCDCLK	FAH, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-	**-**	**-**	**LCDCKS**	**DISP**	**LCDDIV[2:0]**		
-	-	-	R/W	R/W	R/W		

Bit	Name	Description
[7:5]	**-**	Reserved
[4]	**LCDCKS**	**LCD Clock Source Select** 0 = LIRC/2^4. 1 = LXT/2^4.
[3]	**DISP**	**DISP** The LCD display keeps display on or display off during chip power-down mode. If LXT is used as the LCD clock source, user should turn on LXT first by software. 0 = Display off. 1 = Display on.
[2:0]	**LCDDIV[2:0]**	**LCD Clock Divider** 000 = 1/1. 001 = 1/2. 010 = 1/4. 011 = 1/8. 100 = 1/16. 101 = 1/32. Others = Reserved.

LCDDPTR – LCD Data Pointer

Register	SFR Address	Reset Value
LCDPTR	FBH, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	LCDPTR[4:0]				
-	-	-	R/W				

Bit	Name	Description
[7:5]	-	Reserved
[4:0]	LCDPTR[4:0]	**LCD Data Pointer** This field determines which LCD display data register is accessed by LCDDAT. It's also means the LCD segment address.the value is from 0 ~31. User should fill the target pointer value in LCDPTR before accessing LCDDAT. After LCD display data is written to LCDDAT register, the LCDPTR value increases 1 automatically.

LCDDAT – LCD Data

Register	SFR Address	Reset Value
LCDDAT	FCH, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
LCDDAT[7:0]							
R/W							

Bit	Name	Description
[7:0]	**LCDDAT[7:0]**	**LCD Data** This byte is defined which COM pin should be enabled. Bit 0 means COM 0 and bit 7 means COM 7. When value 1 written into this register will be enable the corresponding COM pins and which pixel display is based on LCD SEG defined by LCDPTR. 0 = LCD pixel is cleared. 1 = LCD pixel is darkened.

LCDPWR – LCD Power Saving Mode

Register	SFR Address	Reset Value
LCDPWR	FDH, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	-	PWR_SAVE[1:0]	
-	-	-	-	-	-	R/W	

Bit	Name	Description
[7:2]	-	Reserved
[1:0]	PWR_SAVE[1:0]	**LCD Power Save Mode Select** LCD driving cycle select, turn on timing decide the driving current. 00 = always ON. No power saving. 01 = Turns on 1/4 frame cycle (most power saving condition) 10 = Turns on 2/4 frame cycle 11 = Turns on 3/4 frame cycle

LCDBL – LCD Blink

Register	SFR Address	Reset Value
LCDBL	FEH, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-				BLINK	BLF[2:0]		
-				R/W	R/W		

Address: FEH, Page 3 Reset value: 0000 0000b

Bit	Name	Description
[7:4]	-	Reserved
[3]	BLINK	**LCD BLINK** 0 = LCD always on 1 = LCD blinking. The blinking frequency is based on BLF[2:0] define.
[2:0]	BLF[2:0]	**BLINK Frequency** Blinking frequency define value is from 0~7 When LCDCKS =1, $F_{BLINK} = F_{LXT}/2^{11+ BL\_Time[2:0]}$ When LCDCKS =0, $F_{BLINK} = F_{LIRC}/2^{11+ BL\_Time[2:0]}$

LCDMODE – LCD Mode

Register	SFR Address	Reset Value
LCDMODE	FFH, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
R_MODE	**BUF_MODE**	**-**				**VLCD_MODE[1:0]**	
R/W	R/W	-				R/W	

Bit	Name	Description
[7]	**R_MODE**	**Resister Mode Enable** This bit going to define LCD driver as Resistor Mode or Resistor Enhance Mode About R_MODE, RE_MODE and BUF_MODE define and the LCD driving current please reference Table 6.18-2 LCD Driving Mode . 0 = Disable 1 = Enable. **Note:** When R_MODE is enabled, BUF_MODE should be disabled.
[6]	**BUF_MODE**	**Buffer Mode Enable** This bit going to define LCD driver as Buffer Mode. About R_MODE, RE_MODE and BUF_MODE define and the LCD driving current please reference Table 6.18-2 LCD Driving Mode . 0 = Disable 1 = Enable **Note:** When BUF_MODE is enabled, R_MODE and RE_MODE should be disabled.
[5:2]	**-**	Reserved
[1:0]	**VLCD_MODE[1:0]**	**VLCD Source Mode Select.** This bits defined VLCD voltage source VLCD_MODE / VLCD source 00 / Ext. VLCD 01 / AV_{DD} 10 / VCP (Charge Pump) 11 / Disable Note : VCP value base on LCDCPUMP define.

LCDCPUMP – LCD Charge Pump Voltage Set Value

Register	SFR Address	Reset Value
LCDCPUMP	F1H, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-	-	VCP_SEL[5:0]					
-	-	R/W					

Address: F1H, Page 3 Reset value: 0000 0000b

Bit	Name	Description
[7:6]	-	Reserved
[5:0]	VCP_SEL[5:0]	**Charge Pump Voltage Set Value** 000000 = 5.4V 000101 = 5.2V 001010 = 5.0V 001110 = 4.8V 010011 = 4.6V 011000 = 4.4V 011101 = 4.2V 100010 = 4.0V 100111 = 3.8V 101100 = 3.6V 110000 = 3.4V 110101 = 3.2V 111010 = 3.0V 111111 = 2.8V **Note:** LCD pump up voltage with limitation $V_{LCD(max.)}$ = 1.8 * $V_{DD.}$ LCD pump down voltage without this limitation.

LCDCON1– LCD Control Register 1 (TA Protected)

Register	SFR Address	Reset Value
LCDCON1	F4H, Page 3, TA protected	0000_0000 b

7	6	5	4	3	2	1	0
-				LCDIS	LCDIE	RE_MODE	-
-				R/W	R/W	R/W	-

Bit	Name	Description
[7:4]	-	Reserved
[3]	LCDIS	**LCD Interrupt Source Select** 0 = LCD charge pump counter alarm interrupt When LCDIE is enabled and this bit is 0, only when LCD charge pump counter value over LCDCPALCT0 and LCDCPALCT1 defined value, the LCD will go intot LCD interrupt.The LCDCPALIF(LCDIF[0]) will be set to 1. 1 = LCD charge pump active counter read interrupt When bit set as 1. If **LCDCPIF=1** means LCD module successful to drivring LCD pixel. The charge pump value will be write into LCDCPCT0 and LCDCPCT1 and the LCD interrupt happen.If **LCDCPOVIF** = 1means LCD module charge pump value is match the maximum 0x3FF
[2]	LCDIE	**LCD Interrupt Enable** The Interrupt flag show in LCDIS. 0 = Disable 1 = Enable
[1]	RE_MODE	**LCD Resistor Enhance Mode Enable** This bit going to define LCD as resistor enhance mode and only invalid when R_MODE(LCDMODE[7]) bit is enabled. About R_MODE, RE_MODE and BUF_MODE define and the LCD driving current please reference Table 6.18-2 LCD Driving Mode . 0 = Disable 1 = Enable **Note:** when R_MODE and RE_MODE is enabled, BUF_MODE should be disabled.
[0]	-	Reserved

LCDCPALCT0 – LCD Charge Pump Alarm Counter Value Low Byte (TA Protected)

Register	SFR Address	Reset Value
LCDCPALCT0	F5H, Page 3, TA protected	0000_0000 b

7	6	5	4	3	2	1	0
LCDCPOVCT[7:0]							
R/W							

Bit	Name	Description
[7:0]	**LCDCPOVCT[7:0]**	**LCD Chage Pump Counter Alarm Value for Trig Interrupt Value .** Low byte of Trig LCD interrupt alarm counter value for user setting.

LCDCPALCT1 – LCD Charge Pump Alarm Counter Value High Byte (TA Protected)

Register	SFR Address	Reset Value
LCDCPALCT1	E9H, Page 3, TA protected	0000_0000 b

7	6	5	4	3	2	1	0
-						LCDCPOVCT [9:8]	
-						R/W	

Bit	Name	Description
[7:2]	-	**Reserved.**
[1:0]	**LCDCPOVCT [9:8]**	**LCD Chage Pump Counter Overflow Trig Interrupt Value .** High byte of Trig LCD interrupt alarm counter value for user setting.

LCDCPCT0 – LCD Charge Pump Counter Value Low Byte

Register	SFR Address	Reset Value
LCDCPCT0	F6H, page3	0000_0000 b

7	6	5	4	3	2	1	0
LCDCPCT[7:0]							
R							

Bit	Name	Description
[7:0]	**LCDCPCT[7:0]**	**LCD Current Frame Chage Pump Counter Value Low Byte** Each time after interrupt this byte reload LCD current frame charge pump value low byte. This byte is read only.

LCDCPCT1 – LCD Charge Pump Counter Value High Byte

Register	SFR Address	Reset Value
LCDCPCT1	EAH, page3	0000_0000 b

7	6	5	4	3	2	1	0
-						LCDCPCT[9:8]	
-						R/W	

Bit	Name	Description
[7:2]	-	**Reserved.**
[1:0]	**LCDCPCT [9:8]**	**LCD Current Frame Chage Pump Counter Value High Byte** Each time after interrupt this byte reload LCD current frame charge pump value low byte. This byte is read only.

LCDIF – LCD Interrupt Flag

Register	SFR Address	Reset Value
LCDIF	F7H, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-					LCDCPOVIF	LCDCPIF	LCDCPALIF
-					R/W		R/W

Bit	Name	Description
[7:3]	-	
[2]	LCDCPOVIF	**LCD Charge Pump Counter Value Overflow Flag** This Flag check LCDCPCT0 and LCDCPCT1 counter value. When LCDIS (LCDCON1.3) = 1 and LCDIE (LCDCON1.2) = 1. When LCD module charge pump value is match the maximum 0x3FF this bit will be set to 1 and the LCD interrupt happen. 0 = without interrupt 1 = with interrupt This bit is set 1 by hardware and should be cleard by write “0” to this bit. When this bit is set as 1 user should be noticed for the pixel now is displaying. The circuit short or LCD pixel damage may cause thie result. Since as normal the charge pump counter value shoud not reach 0x3FF.
[1]	LCDCPIF	**LCD Charge Pump Interrupt Flag** This Flag check LCDCPCT0 and LCDCPCT1 counter value. if LCDIS (LCDCON1.3) = 1 and LCDIE (LCDCON1.2) = 1. When LCD module successful to drivring LCD pixel. The charge pump value will be write into LCDCPCT0 and LCDCPCT1 and the LCD interrupt happen. User can read from this two register byte to find the charge pump reference value. 0 = without interrupt 1 = with interrupt This bit is set 1 by hardware and should be cleard by write “0” to this bit.
[0]	LCDCPALIF	**LCD Charge Pump Alarm Value Match Interrupt Flag** This Flag check LCDCPALCT0 and LCDCPALCT1 counter value. if LCDIS (LCDCON1.3) = 0 and LCDIE (LCDCON1.2) = 1. Only when LCD charge pump counter is over LCDCPALCT0 and LCDCPALCT1 defined alarm value the LCD interrupt will be happen. 0 = without interrupt 1 = with interrupt This bit is set 1 by hardware and should be cleard by write “0” to this bit.

6.18.6 LCD Program Flow

Before LCD driver is enabled, the I/O state shared with used COM and SEG pins should be carefully considered. All used COM and SEG pins generate analog output waveforms.

First, user needs to determine the VLCD, duty, and bias selections according to the target LCD panel. A suitable frame rate, normally 30 Hz to 100 Hz, is also important and needs to carefully give the LCDCLK register a proper value. If LXT is used as the LCD clock source, user should turn on LXT first by software. Otherwise, user should turn on LIRC by software. Proper considerations above will obtain a good display quality.

After the steps above, user can enable all used SEG pins by setting the corresponding SEGnEN bits in registers LCDSEG0 to LCDSEG3. The last step is to enable the LCD driver by setting LCDEN (LCDCON.7) bit as 1. This step generates LCD driving waveforms via all used CON and SEG pins. User can determine the LCDPTR and set LCDDAT to darken or clear each pixel on LCD panel afterward.

6.19 Real Time Clock (RTC)

6.19.1 Overview

The Real Time Clock (RTC) controller provides the real time and calendar message. The RTC offers programmable time tick and alarm match interrupts. The data format of time and calendar messages are expressed in BCD format. A digital frequency compensation feature is available to compensate external crystal oscillator frequency accuracy.

6.19.2 Features

- Supports real time counter and calendar counter for RTC time and calendar check.
- Supports alarm time and calendar settings
- Supports alarm time and calendar mask enable settings.
- Selectable 12-hour or 24-hour time scale setting.
- Supports Leap Year indication setting.
- Supports Day of the Week counter setting.
- Frequency of RTC clock source compensate by RTCFREQADJ0/1 register.
- All time and calendar message expressed in BCD format.
- Supports periodic RTC Time Tick interrupt with 8 period interval options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second.
- Supports RTC Time Tick and Alarm Match interrupt.
- Supports chip wake-up from Idle or Power-down mode while a RTC interrupt signal is generated.

6.19.3 Block Diagram

The RTC block diagram is shown below.

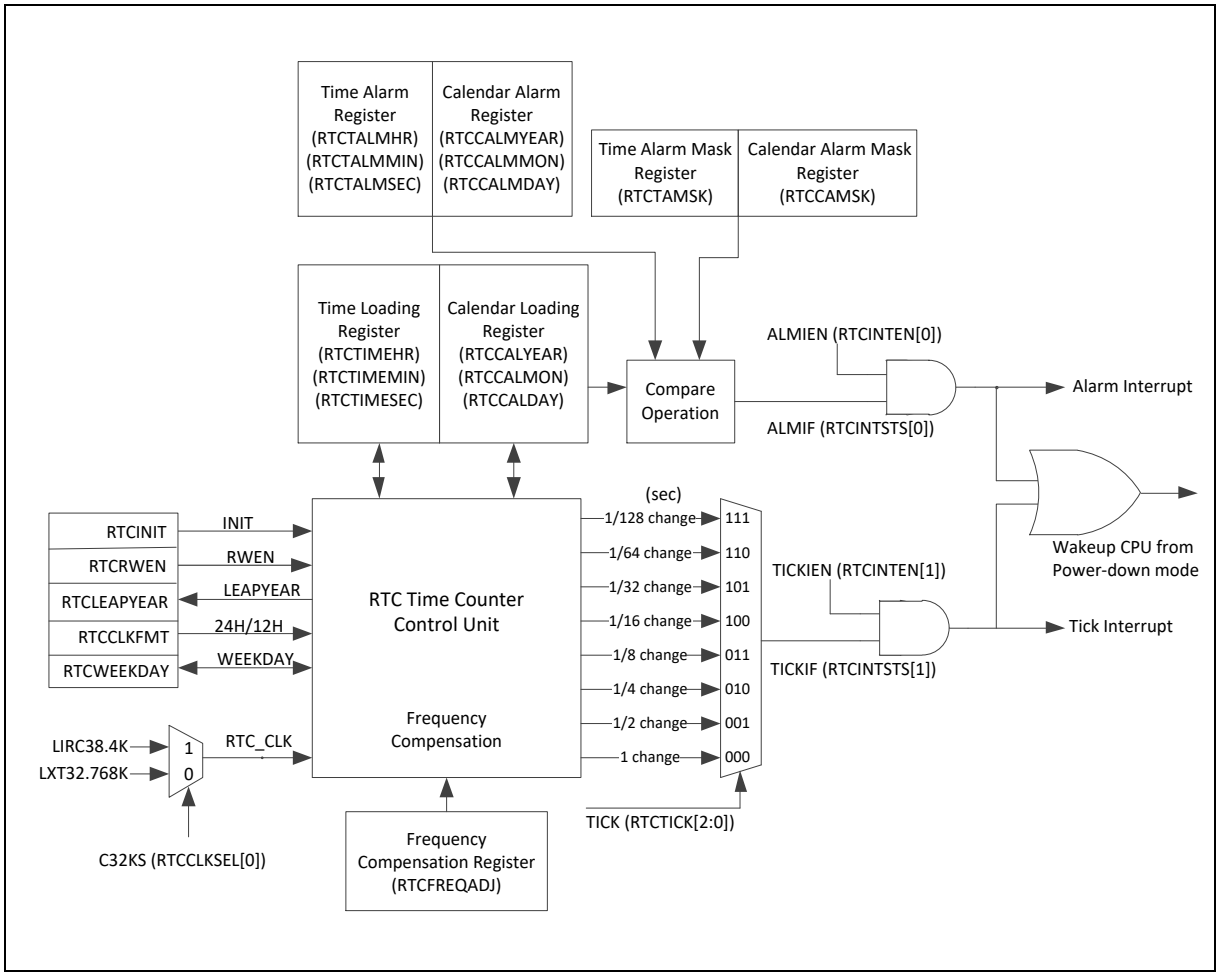


Figure 6.19-1 RTC Block Diagram

6.19.4 Functional Description

6.19.4.1 RTC Initiation

When a RTC block is powered on, RTC is at reset state. User has to write a number 0x57 to RTC initial register INIT(RTCINIT[7:0]) to make RTC leave reset state. Once the INIT(RTCINIT[7:0]) register is written as 0x57, the RTC will be in normal active state permanently. User can read INIT[0](RTCINIT[0]) to check the RTC is at normal active state or reset state. In normal active state, user write a number 0x5F to RTCINIT[7:0] to pause the counting of RTC time and calendar. Then user write a number 0x57 to RTCINIT[7:0] to continue the RTC counting.

6.19.4.2 RTC Read/Write Enable

If RWENF (RTCRWEN[0]) bit read is as 1, it means the RTC registers are read/write accessible. When executing write RTC register command, RWENF (RTCRWEN[0]) will be clear to 0. The RTC Register Description access attribute when RWENF is 1 and 0 are shown as below.

Register	INIR = 0	RWENF = 1	RWENF = 0
RTCINIT	available	R/W	R/W
RTCRWEN	available	R/W	R/W
RTCFREQADJ0	available	R/W	R/W
RTCFREQADJ1	available	R/W	R/W
RTC_INTEN	available	R/W	R/W
RTC_INTSTS	available	R/W	R/W
RTCTIMESEC	Not available	R/W	R
RTCTIMEMIN	Not available	R/W	R
RTCTIMEHR	Not available	R/W	R
RTCCALDAY	Not available	R/W	R
RTCCALMON	Not available	R/W	R
RTCCALYEAR	Not available	R/W	R
RTCCLKFMT	Not available	R/W	R
RTCRTCTEST	Not available	R/W	R
RTCWEEKDAY	Not available	R/W	R
RTCTALMSEC	Not available	R/W	R
RTCTALMMIN	Not available	R/W	R
RTCTALMHR	Not available	R/W	R
RTCCALMDAY	Not available	R/W	R
RTCCALMMON	Not available	R/W	R
RTCCALMYEAR	Not available	R/W	R
RTCLEAPYEAR	Not available	R	R
RTCTICK	Not available	R/W	R
RTCTAMSK	Not available	R/W	R

Register	INIR = 0	RWENF = 1	RWENF = 0
RTCCAMSK	Not available	R/W	R

Table 6.19-1 RTC Read/Write Enable

6.19.4.3 Frequency Compensation

The RTCFREQADJ0 and RTCFREQADJ1 registers allow user to make digital compensation to a clock input. Please follow the example and formula below to write the actual frequency of 32k crystal to RTCFREQADJ0 and RTCFREQADJ1 registers. Following are the compensation examples for higher or lower than 32768 Hz.

Example 1:

Frequency counter measurement : 32773.65 Hz

Integer Part: 32773 => RTCFREQADJ1[4:0] = 0x15, Refer the INTEGER(RTCFREQADJ1[4:0]) to get detail setting value.

Fraction Part: 0.65 X 64 = 41.6(0x2A) => RTCFREQADJ0[5:0]=0x2A

Example 2:

Frequency counter measurement : 32763.25 Hz

Integer part: 32763=> RTCFREQADJ1[4:0] = 0x0B, Refer the INTEGER(RTC_FREQADJ1[4:0]) to get detail setting value.

Fraction part: 0.25 X 64 = 16(0x10) => RTCFREQADJ0[5:0] = 0x10

Note: The value of RTCFREQADJ1 register will be the default value (0x10) and RTCFREQADJ0 will be (0x00) while the compensation is not executed. User can utilize a frequency counter to measure RTC clock source via clock output function in manufacturing. In the meanwhile, user can use clock output function to check the result of RTC frequency compensation.

6.19.4.4 Time and Calendar Counter

RTCTIMESEC, RTCTIMEMIN, RTCTIMEHR, RTCCALDAY, RTCCALMON and RTCCALYEAR are used to load the real time and calendar. RTCTALMSEC, RTCTALMMIN, RTCTALMHR, RTCCALMDAY RTCCALMMON and RTCCALMYEAR are used for setup alarm time and calendar.

6.19.4.5 12/24 hour Time Scale Selection

The 12/24 hour time scale selection depends on 24HEN (RTCCLKFMT[0]). When RTC runs as 12-hour time scale mode, RTCTIMEHR[5] (the high bit of TENHR[1:0]) means AM/PM indication, if RTCTIMEHR[5] is 1, it indicates PM time message and RTCTIMEHR[5] is 0 indicates AM time message.) shows RTCTIMEHR mapping table of 12/24 hour time scale selection.

Note: The Hour Value Write Into RTCTIMEHR[5:0], Messages Are Expressed In BCD Format.			
24-Hour Time Scale (24HEN = 1)		12-Hour Time Scale (PM Time + 0x20) (24HEN = 0) (PM Time + 0x20)	
0x00 (AM12)	0x12 (PM12)	0x12 (AM12)	0x32 (PM12)
0x01 (AM01)	0x13 (PM01)	0x01 (AM01)	0x21 (PM01)
0x02 (AM02)	0x14 (PM02)	0x02 (AM02)	0x22 (PM02)
0x03 (AM03)	0x15 (PM03)	0x03 (AM03)	0x23 (PM03)
0x04 (AM04)	0x16 (PM04)	0x04 (AM04)	0x24 (PM04)
0x05 (AM05)	0x17 (PM05)	0x05 (AM05)	0x25 (PM05)

Note: The Hour Value Write Into RTCTIMEHR[5:0], Messages Are Expressed In BCD Format.			
24-Hour Time Scale (24HEN = 1)		12-Hour Time Scale (PM Time + 0x20) (24HEN = 0) (PM Time + 0x20)	
0x06 (AM06)	0x18 (PM06)	0x06 (AM06)	0x26 (PM06)
0x07 (AM07)	0x19 (PM07)	0x07 (AM07)	0x27 (PM07)
0x08 (AM08)	0x20 (PM08)	0x08 (AM08)	0x28 (PM08)
0x09 (AM09)	0x21 (PM09)	0x09 (AM09)	0x29 (PM09)
0x10 (AM10)	0x22 (PM10)	0x10 (AM10)	0x30 (PM10)
0x11 (AM11)	0x23 (PM11)	0x11 (AM11)	0x31 (PM11)

Table 6.19-212/24 hour Time Scale Selection

6.19.4.6 Day of the Week Counter

The RTC controller provides day of week in WEEKDAY bits (RTCWEEKDAY[2:0]). The value is defined from 0 to 6 to represent Sunday to Saturday respectively.

6.19.4.7 Periodic Time Tick Interrupt

The periodic time tick interrupt has 8 period interval options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second that are selected by TICK bits (RTCTICK[2:0]). When Periodic Time Tick interrupt is enabled by setting TICKIEN (RTCINTEN[1]) to 1, the Periodic Time Tick interrupt is requested periodically in the period selected by RTCTICK[2:0] settings.

6.19.4.8 Alarm Interrupt

When the real time and calendar message in RTCTIMESEC, RTCTIMEMIN, RTCTIMEHR, RTCCALDAY, RTCCALMON and RTCCALYEAR registers are equal to alarm time and calendar values in RTCTALMSEC, RTCTALMMIN, RTCTALMHR, RTCCALMDAY, RTCCALMMON and RTCCALMYEAR registers, the RTC alarm interrupt flag ALMIF (RTCINTSTS[0]) is set to 1 and the RTC alarm interrupt signal assert if the alarm interrupt enable ALMIEN (RTCINTEN[0]) is enabled.

The RTC controller provides time alarm mask register (RTCTAMSK register) and Calendar Alarm Mask Register (RTCCAMSK register) to mask the specified digit and generate periodic interrupt without changing the alarm match condition in RTCTALM and RTCCALM registers in each alarm interrupt service routine.

6.19.4.9 Application Note

1. All data in RTCTALMSEC, RTCTALMMIN, RTCTALMHR,RTCCALMDAY, RTCCALMMON, RTCCALMYEAR, RTCTIMESEC, RTCTIMEMIN, RTCTIMEHR, RTCCALDAY, RTCCALMON and RTCCALYEAR registers are all expressed in BCD format.
2. User has to make sure that the loaded values are reasonable. For example, Load RTCCALYEAR as 201a (year), RTCCALMON as 13 (month), RTCCALDAY as 00 (day), or RTCCAL (year, month, day) does not match with RTCWEEKDAY, etc.
3. In RTCCALYEAR and RTCCALMYEAR, only 2 BCD digits are used to express “year”. The 2 BCD digits of xy means 20xy, rather than 19xy or 21xy.
4. Example of 12-Hour Time Setting
   If current RTC time is PM12:59:30 in 12-Hour Time Scale mode, the RTCTIME(hour, minute, second) setting as:

   HOUR:
   RTCTIMEHR[5:0]: 0x32 (0x12+0x20) combined by TENHR (RTCTIMEHR[5:4]) is 0x3, HR

(RTCTIMEHR[3:0]) is 0x2.

MIN:
RTCTIMEMIN[6:0]: 0x59 combined by TENMIN (RTCTIME[6:4]) is 0x5, MIN (RTCTIMEMIN[3:0]) is 0x9.

SEC:
RTCTIMESEC[6:0]: 0x30 combined by TENSEC (RTCTIMESEC[6:4]) is 0x3, SEC (RTCTIMESEC[3:0]) is 0x0.

5. The below table shows registers value after both core power and battery power are first powered on.

Register	Reset State
RTCINIT	0
RTCRWEN	0x01
RTCCALYEAR	15 (year)
RTCCALMON	8 (month)
RTCCALDAY	8 (day)
RTCTIMEHR	00 (hour)
RTCTIMEMIN	00 (minute)
RTCTIMESEC	00 (second)
RTCCALMYEAR	00 (year)
RTCCALMMON	00 (month)
RTCCALMDAY	00 (day)
RTCTALMHR	00 (hour)
RTCTALMMIN	00 (minute)
RTCTALMSEC	00 (second)
RTCCLKFMT	1 (24-hour mode)
RTCWEEKDAY	6 (Saturday)
RTCINTEN	0
RTCINTSTS	0
RTCLEAPYEAR	0
RTCTICK	0

Table 6.19-3 Registers Value after Powered On

6.19.5 Register Description

RTCINIT – RTC Initiation Register (TA Protected)

Register	SFR Address	Reset Value
RTCINIT	A1H, Page 3, TA protected	0000_0000 b

7	6	5	4	3	2	1	0
INIT[7:2]						INIT[1]/HOLD	INIT/[0]ACTIVE
R/W						R/W	R/W

Bit	Name	Description
[7:2]	**INIT[7:2]**	**RTC Initiation (Write Only)** When RTC block is powered on, RTC is at reset state. User has to write a number (0x 57) to INIT to make RTC leave reset state. Once the INIT[7:0] is written as 0x57, the RTC will be in un-reset state permanently. Once the INIT[7:0] is written as 0x5F, the RTC will be in hold and un-reset state permanently. This Byte is TA protected. The INIT is a write-only field and read value will be always 0.
[1]	**INIT[1]/HOLD**	**RTC HOLD Flag (Read Only)** 0 = RTC hold flag is inavtive. 1 = RTC hold flag is active.
[0]	**INIT[0]/ACTIVE**	**RTC Active Status (Read Only)** 0 = RTC is at reset state. 1 = RTC is at normal active state.

RTCRWEN – RTC Access Enable Register

Register	SFR Address	Reset Value
RTCRWEN	A2H, Page 3	0000_0001 b

7	6	5	4	3	2	1	0
-						FADJTG	RWENF
-						R/W	R

Bit	Name	Description
[7:2]	-	Reserved.
[1]	FADJTG	**RTC Counter Update (Read and Write)** Set this bit = 1 by software. It will update to RTC counter from RTCFREQADJ1/0. After RTC counter updated, this bit will auto recover to 0 by hardware.
[0]	RWENF	**RTC Register Access Enable Flag (Read Only)** 0 = RTC register read/write Disabled. 1 = RTC register read/write Enabled. **Note:** RWENF will be masked to 0 during RTC busy.

RTCCLKSEL – RTC Clock Select Register

Register	SFR Address	Reset Value
RTCCLKSEL	A3H, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-							**C32KS**
-							R/W

Bit	Name	Description
[7:1]	-	Reserved.
[0]	**C32KS**	**Clock 32K Source Selection:** 0 = Internal 32K clock is from 32.786 kHz crystal . 1 = Internal 32K clock is from LIRC38.4 kHz.

RTCFREQADJ0 – RTC Frequency Compensation 0 Register

Register	SFR Address	Reset Value
RTCFREQADJ0	A4H, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-		FRACTION					
-		R/W					

Bit	Name	Description
[7:6]	-	Reserved.
[5:0]	**FRACTION**	**Fraction Part** Formula: FRACTION = (fraction part of detected value) X 64. **Note:** Digit in FCR must be expressed as hexadecimal number.
Note: FREQADJ's counter will be reset for start to Compensatie when write RTCFREQADJ0/1 , RTCTIME, RTCCAL, RTCWEEKDAY. Imply RTC Time will be restart .		

RTCFREQADJ1 – RTC Frequency Compensation 1 Register

Register	SFR Address	Reset Value
RTCFREQADJ1	A5H, Page 3	0001_0000 b

7	6	5	4	3	2	1	0
-			INTEGER				
-			R/W				

Bit	Name	Description
[7:5]	-	Reserved.
[4:0]	**INTEGER**	**Integer Part** 00000 = Integer part of detected value is 32752. 00001 = Integer part of detected value is 32753. 00010 = Integer part of detected value is 32754. 00011 = Integer part of detected value is 32755. 00100 = Integer part of detected value is 32756. 00101 = Integer part of detected value is 32757. 00110 = Integer part of detected value is 32758. 00111 = Integer part of detected value is 32759. 01000 = Integer part of detected value is 32760. 01001 = Integer part of detected value is 32761. 01010 = Integer part of detected value is 32762. 01011 = Integer part of detected value is 32763. 01100 = Integer part of detected value is 32764. 01101 = Integer part of detected value is 32765. 01110 = Integer part of detected value is 32766. 01111 = Integer part of detected value is 32767. 10000 = Integer part of detected value is 32768. 10001 = Integer part of detected value is 32769. 10010 = Integer part of detected value is 32770. 10011 = Integer part of detected value is 32771. 10100 = Integer part of detected value is 32772. 10101 = Integer part of detected value is 32773. 10110 = Integer part of detected value is 32774. 10111 = Integer part of detected value is 32775. 11000 = Integer part of detected value is 32776. 11001 = Integer part of detected value is 32777. 11010 = Integer part of detected value is 32778. 11011 = Integer part of detected value is 32779. 11100 = Integer part of detected value is 32780. 11101 = Integer part of detected value is 32781. 11110 = Integer part of detected value is 32782. 11111 = Integer part of detected value is 32783.

Bit	Name	Description
Note:FREQADJ's counter will be reset for start to Compensatie when write RTCFREQADJ0/1 , RTCTIME, RTCCAL, RTCWEEKDAY. Imply RTC Time will be restart .		

RTCINTEN – RTC Interrupt Enable Register

Register	SFR Address	Reset Value
RTCINTEN	A6H, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-						TICKIEN	ALMIEN
-						R/W	R/W

Bit	Name	Description
[7:2]	-	Reserved.
[1]	TICKIEN	**Time Tick Interrupt Enable Bit** Set TICKIEN to 1 can also enable chip wake-up function when RTC tick interrupt event is generated. 0 = RTC Time Tick interrupt Disabled. 1 = RTC Time Tick interrupt Enabled.
[0]	ALMIEN	**Alarm Interrupt Enable Bit** Set ALMIEN to 1 can also enable chip wake-up function when RTC alarm interrupt event is generated. 0 = RTC Alarm interrupt Disabled. 1 = RTC Alarm interrupt Enabled.

RTCINTSTS – RTC Interrupt Status Register

Register	SFR Address	Reset Value
RTCINTSTS	A7H, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-						TICKIF	ALMIF
-						R/W	R/W

Bit	Name	Description
[7:2]	-	Reserved.
[1]	TICKIF	**RTC Time Tick Interrupt Flag** 0 = Tick condition does not occur. 1 = Tick condition occur.
[0]	ALMIF	**RTC Alarm Interrupt Flag** 0 = Alarm condition is not matched. 1 = Alarm condition is matched.

RTCTIMESEC – RTC Time-second Loading Register

Register	SFR Address	Reset Value
RTCTIMESEC	A9H, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-	TENSEC[2:0]			SEC[3:0]			
-	R/W			R/W			

Bit	Name	Description
[7]	-	Reserved.
[6:4]	**TENSEC[2:0]**	10-Sec Time Digit (0~5)
[3:0]	**SEC[3:0]**	1-Sec Time Digit (0~9)

Note:

1. RTCTIMESEC is a BCD digit counter and RTC will not check loaded data.
2. The reasonable value range is listed in the parenthesis.

RTCTIMEMIN – RTC Time-minute Loading Register

Register	SFR Address	Reset Value
RTCTIMEMIN	AAH, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-	TENMIN[2:0]			MIN[3:0]			
-	R/W			R/W			

Bit	Name	Description
[7]	-	Reserved.
[6:4]	**TENMIN[2:0]**	10-Min Time Digit (0~5)
[3:0]	**MIN[3:0]**	1-Min Time Digit (0~9)

Note:
1. RTCTIMEMIN is a BCD digit counter and RTC will not check loaded data.
2. The reasonable value range is listed in the parenthesis.

RTCTIMEHR – RTC Time-hour Loading Register

Register	SFR Address	Reset Value
RTCTIMEHR	ABH, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-		TENHR[1:0]		HR[3:0]			
-		R/W		R/W			

Bit	Name	Description
[7:6]	-	Reserved.
[5:4]	**TENHR[1:0]**	10-Hour Time Digit (0~2) When RTC runs as 12-hour time scale mode, RTCTIMEHR[5] (the high bit of TENHR[1:0]) means AM/PM indication (If RTCTIMEHR[5] is 1, it indicates PM time message.)
[3:0]	**HR[3:0]**	1-Hour Time Digit (0~9)

Note:

1. RTCTIMEHR is a BCD digit counter and RTC will not check loaded data.
2. The reasonable value range is listed in the parenthesis.

RTCCALDAY – RTC Calendar-day Loading Register

Register	SFR Address	Reset Value
RTCCALDAY	ADH, Page 3	0000_1000 b

7	6	5	4	3	2	1	0
-		TENDAY		DAY			
-		R/W		R/W			

Bit	Name	Description
[7:6]	-	Reserved.
[5:4]	**TENDAY**	10-Day Calendar Digit (0~3)
[3:0]	**DAY**	1-Day Calendar Digit (0~9)

Note:

1. RTCCALDAY is a BCD digit counter and RTC will not check loaded data.
2. The reasonable value range is listed in the parenthesis.

RTCCALMON – RTC Calendar-month Loading Register

Register	SFR Address	Reset Value
RTCCALMON	AEH, Page 3	0000_1000 b

7	6	5	4	3	2	1	0
-			TENMON	MON[3:0]			
-			R/W	R/W			

Bit	Name	Description
[7:5]	-	Reserved.
[4]	TENMON	10-Month Calendar Digit (0~1)
[3:0]	MON[3:0]	1-Month Calendar Digit (0~9)

Note:

1. RTCCALMON is a BCD digit counter and RTC will not check loaded data.
2. The reasonable value range is listed in the parenthesis.

RTCCALYEAR – RTC Calendar-year Loading Register

Register	SFR Address	Reset Value
RTCCALYEAR	AFH, Page 3	0001_0101 b

7	6	5	4	3	2	1	0
TENYEAR				YEAR			
R/W				R/W			

Bit	Name	Description
[7:4]	**TENYEAR**	10-Year Calendar Digit (0~9)
[3:0]	**YEAR**	1-Year Calendar Digit (0~9)

Note:

1. RTCCALYEAR is a BCD digit counter and RTC will not check loaded data.
2. The reasonable value range is listed in the parenthesis.

RTCTALMSEC – RTC Time Alarm-second Register

Register	SFR Address	Reset Value
RTCTALMSEC	B1H, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-	TENSEC[2:0]			SEC[3:0]			
-	R/W			R/W			

Bit	Name	Description
[7]	-	Reserved.
[6:4]	**TENSEC[2:0]**	10-Sec Time Digit of Alarm Setting (0~5)
[3:0]	**SEC[3:0]**	1-Sec Time Digit of Alarm Setting (0~9)

Note:

1. RTCTALMSEC is a BCD digit counter.
2. The reasonable value range is listed in the parenthesis.
3. This register can be read back after the RTC register access enable bit RWENF (RTCRWEN[0]) is active.

RTCTALMMIN – RTC Time Alarm-minute Register

Register	SFR Address	Reset Value
RTCTALMMIN	B2H, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-	TENMIN[2:0]			MIN[3:0]			
-	R/W			R/W			

Bit	Name	Description
[7]	-	Reserved.
[6:4]	**TENMIN[2:0]**	10-Min Time Digit of Alarm Setting (0~5)
[3:0]	**MIN[3:0]**	1-Min Time Digit of Alarm Setting (0~9)

Note:

1. RTCTALMMIN is a BCD digit counter.
2. The reasonable value range is listed in the parenthesis.
3. This register can be read back after the RTC register access enable bit RWENF (RTCRWEN[0]) is active.

RTCTALMHR – RTC Time Alarm-hour Register

Register	SFR Address	Reset Value
RTCTALMHR	B3H, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-		TENHR[1:0]		HR[3:0]			
-		R/W		R/W			

Bit	Name	Description
[7:6]	-	Reserved.
[5:4]	**TENHR[1:0]**	10-Hour Time Digit of Alarm Setting (0~2) When RTC runs as 12-hour time scale mode, RTCTIMEHR[5] (the high bit of TENHR[1:0]) means AM/PM indication (If RTCTIMEHR[5] is 1, it indicates PM time message.)
[3:0]	**HR[3:0]**	1-Hour Time Digit of Alarm Setting (0~9)

Note:

1. RTCTALMHR is a BCD digit counter.
2. The reasonable value range is listed in the parenthesis.
3. This register can be read back after the RTC register access enable bit RWENF (RTCRWEN[0]) is active.

RTCCALMDAY – RTC Calendar Alarm-day Register

Register	SFR Address	Reset Value
RTCCALMDAY	B5H, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-		TENDAY[1:0]		DAY[3:0]			
-		R/W		R/W			

Bit	Name	Description
[7:6]	-	Reserved.
[5:4]	**TENDAY[1:0]**	10-Day Calendar Digit of Alarm Setting (0~3)
[3:0]	**DAY[3:0]**	1-Day Calendar Digit of Alarm Setting (0~9)

Note:

1. RTCCALMDAY is a BCD digit counter.
2. The reasonable value range is listed in the parenthesis.
3. This register can be read back after the RTC register access enable bit RWENF (RTCRWEN[0]) is active.

RTCCALMMON – RTC Calendar Alarm-month Register

Register	SFR Address	Reset Value
RTCCALMMON	B6H, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-			TENMON	MON[3:0]			
-			R/W	R/W			

Bit	Name	Description
[7:5]	-	Reserved.
[4]	TENMON	10-Month Calendar Digit of Alarm Setting (0~1)
[3:0]	MON[3:0]	1-Month Calendar Digit of Alarm Setting (0~9)

Note:

1. RTCCALMMON is a BCD digit counter.
2. The reasonable value range is listed in the parenthesis.
3. This register can be read back after the RTC register access enable bit RWENF (RTCRWEN[0]) is active.

RTCWEEKDAY – RTC Calendar Alarm-year Register

Register	SFR Address	Reset Value
RTCCALMYEAR	B7H, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
TENYEAR[3:0]				YEAR[3:0]			
R/W				R/W			

Bit	Name	Description
[7:4]	**TENYEAR[3:0]**	10-Year Calendar Digit of Alarm Setting (0~9)
[3:0]	**YEAR[3:0]**	1-Year Calendar Digit of Alarm Setting (0~9)

Note:

1. RTCCALMYEAR is a BCD digit counter.
2. The reasonable value range is listed in the parenthesis.
3. This register can be read back after the RTC register access enable bit RWENF (RTCRWEN[0]) is active.

RTCCLKFMT – RTC Time Scale Selection Register

Register	SFR Address	Reset Value
RTCCLKFMT	B9H, Page 3	0000_0001 b

7	6	5	4	3	2	1	0
-							24HEN
-							R/W

Bit	Name	Description
[7:1]	-	Reserved.
[0]	24HEN	**24-Hour / 12-Hour Time Scale Selection** Indicates that RTCTIMEHR and RTCTALMHR are in 24-hour time scale or 12-hour time scale 0 = 12-hour time scale with AM and PM indication selected. 1 = 24-hour time scale selected.

RTCWEEKDAY – RTC Day of the Week Register

Register	SFR Address	Reset Value
RTCWEEKDAY	BBH, Page 3	0000_0110 b

7	6	5	4	3	2	1	0
-					WEEKDAY		
-					R/W		

Bit	Name	Description
[7:3]	-	Reserved.
[2:0]	WEEKDAY	**Day of the Week Register** 000 = Sunday. 001 = Monday. 010 = Tuesday. 011 = Wednesday. 100 = Thursday. 101 = Friday. 110 = Saturday. 111 = Reserved.

RTCLEAPYEAR – RTC Leap Year Indication Register

Register	SFR Address	Reset Value
RTCLEAPYEAR	BCH, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-							LEAPYEAR
-							R

Bit	Name	Description
[7:1]	-	Reserved.
[0]	LEAPYEAR	**Leap Year Indication Register (Read Only)** 0 = This year is not a leap year. 1 = This year is leap year.

RTCTICK – RTC Time Tick Register

Register	SFR Address	Reset Value
RTCTICK	BDH, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-					TICK[2:0]		
-					R/W		

Bit	Name	Description
[7:3]	-	Reserved.
[2:0]	TICK[2:0]	**Time Tick Register** These bits are used to select RTC time tick period for Periodic Time Tick Interrupt request. 000 = Time tick is 1 second. 001 = Time tick is 1/2 second. 010 = Time tick is 1/4 second. 011 = Time tick is 1/8 second. 100 = Time tick is 1/16 second. 101 = Time tick is 1/32 second. 110 = Time tick is 1/64 second. 111 = Time tick is 1/128 second. **Note:** This register can be read back after the RTC register access enable bit RWENF (RTCRWEN[0]) is active.

RTCTAMSK – RTC Time Alarm MASK Register

Register	SFR Address	Reset Value
RTCTAMSK	BEH, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-		MTENHR	MHR	MTENMIN	MMIN	MTENSEC	MSEC
-		R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:6]	-	Reserved.
[5]	**MTENHR**	Mask 10-Hour Time Digit of Alarm Setting (0~2)
[4]	**MHR**	Mask 1-Hour Time Digit of Alarm Setting (0~9)
[3]	**MTENMIN**	Mask 10-Min Time Digit of Alarm Setting (0~5)
[2]	**MMIN**	Mask 1-Min Time Digit of Alarm Setting (0~9)
[1]	**MTENSEC**	Mask 10-Sec Time Digit of Alarm Setting (0~5)
[0]	**MSEC**	Mask 1-Sec Time Digit of Alarm Setting (0~9)

Note:

1. RTCTALM is a BCD digit counter and RTC will not check loaded data.
2. The reasonable value range is listed in the parenthesis.
3. MTENHR/MHR base on 24 hour Time Scale.

RTCCAMSK – RTC Calendar Alarm MASK Register

Register	SFR Address	Reset Value
RTCCAMSK	BFH, Page 3	0000_0000 b

7	6	5	4	3	2	1	0
-		MTENYEAR	MYEAR	MTENMON	MMON	MTENDAY	MDAY
-		R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:6]	-	Reserved.
[5]	**MTENYEAR**	Mask 10-Year Calendar Digit of Alarm Setting (0~9)
[4]	**MYEAR**	Mask 1-Year Calendar Digit of Alarm Setting (0~9)
[3]	**MTENMON**	Mask 10-Month Calendar Digit of Alarm Setting (0~1)
[2]	**MMON**	Mask 1-Month Calendar Digit of Alarm Setting (0~9)
[1]	**MTENDAY**	Mask 10-Day Calendar Digit of Alarm Setting (0~3)
[0]	**MDAY**	Mask 1-Day Calendar Digit of Alarm Setting (0~9)

Note:
1. RTCCALM is a BCD digit counter and RTC will not check loaded data.
2. The reasonable value range is listed in the parenthesis.

6.20 Touch Key (TK)

6.20.1 Overview

The capacitive touch key sensing controller supports several programmable sensitivity levels for different applications to detect the finger touched or near the electrode covered by dielectric. It supports total 14 keys with single scan or programmable periodic key scans, and system can be waked up by any key for low power applications.

6.20.2 Features

- Supports up to 14 Touch Keys + Reference Pad + Shielding Electrode
- Supports any TK pin as reference pad and any one of CLKO pin as shielding electrode.
- Programmable sensitivity levels for each channel.
- Programmable scanning speed for different applications.
- Supports any touch key wake up for low power applications.
- Supports single key scan and programmable periodic key scan.
- Programmable interrupt options for key scan complete withardwareithout threshold control.

6.20.3 Basic Configuration

The Touch Key pins select source

Function Pin Name	Number of pins	Select Range
Reference	1	One of TK0 ~ TK14
Shielding	1	One of CLKO pin
Touch Key	By user	TK0 ~ TK14 except reference

Table 6.20-1 All Touch Key Pins Select Source List

6.20.4 Block Diagram

The block diagram of Touch Key controller is depicted as following:

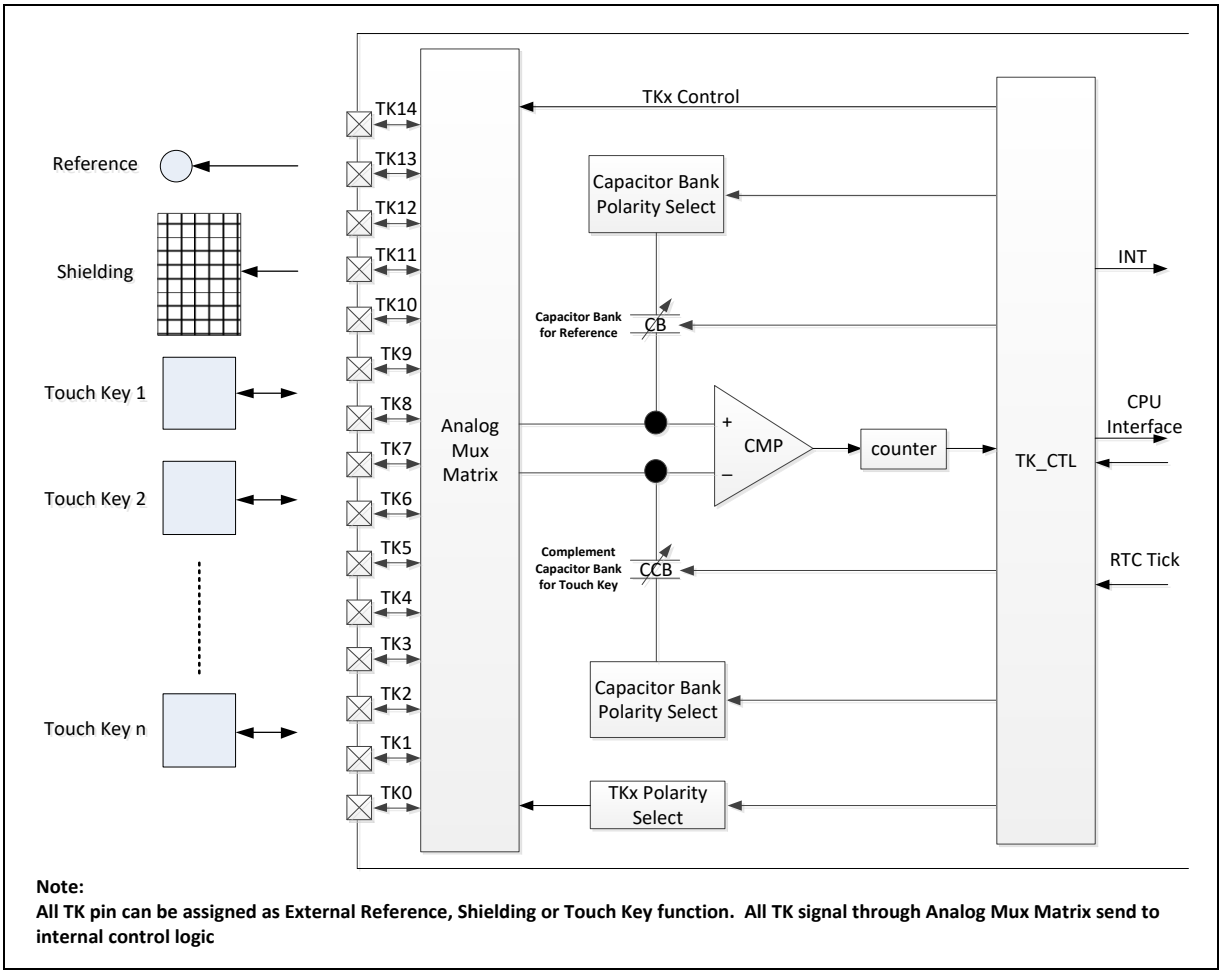


Figure 6.20-1Touch Key block diagram

6.20.5 Functional Description

When a finger touches on a key pad, capacitance of the key pad sensed by Touch Key controller will be bigger than it is not touched. The capacitance is measured by Touch Key controller analog front end circuit, user can read the sensed capacitance value to distinguish if a finger touch event occurs or not.

There is a sensing threshold control for each channel, it supports keys scanning automatically, until any threshold setting condition is met. Thus processor can keep its work or sleep state to save power consumption without interrupt.

The sensitivity level of Touch Key controller is programmable. The key scanning time is programmable for higher sensitivity or scanning speed.

Any key wake up function and proximity detection are also supported.

In this chapter the letter x within register name denotes channel 0 to 14.

Capacitive Touch Key Fundamentals

The capacitance of the touch key without a finger touch is called as “parasitic capacitance”, C_P. Parasitic capacitance results from the electric field between the touch key (including the sensor pad, traces, and vias) and other conductors in the system such as the ground planes, traces, any metal in the product’s chassis or enclosure, etc. The touch key pin and internal capacitances of NuMicro® ML56 series also contribute to the parasitic capacitance.

C_F is known as the finger capacitance. When a finger touches the touch key, the parasitic capacitance C_P and finger capacitance C_F are parallel to each other because both represent the capacitance between the touch key pin and ground. Therefore, the total capacitance C_T of the sensor is the sum of C_P and C_F.

$$C_T = C_P + C_F$$

The touch key controller converts the capacitance C_T into equivalent digital counts called raw data. Because a finger touch increases the total capacitance of the touch key, an increase in raw data indicates a finger touch.

Capacitive Touch Key Sensing Method

Touch key sensing method as shown in Figure 6.20-2, it implements two switching capacitor banks for injecting charges to C_P (or C_T) and C_R which is the parasitic capacitance of reference channel.

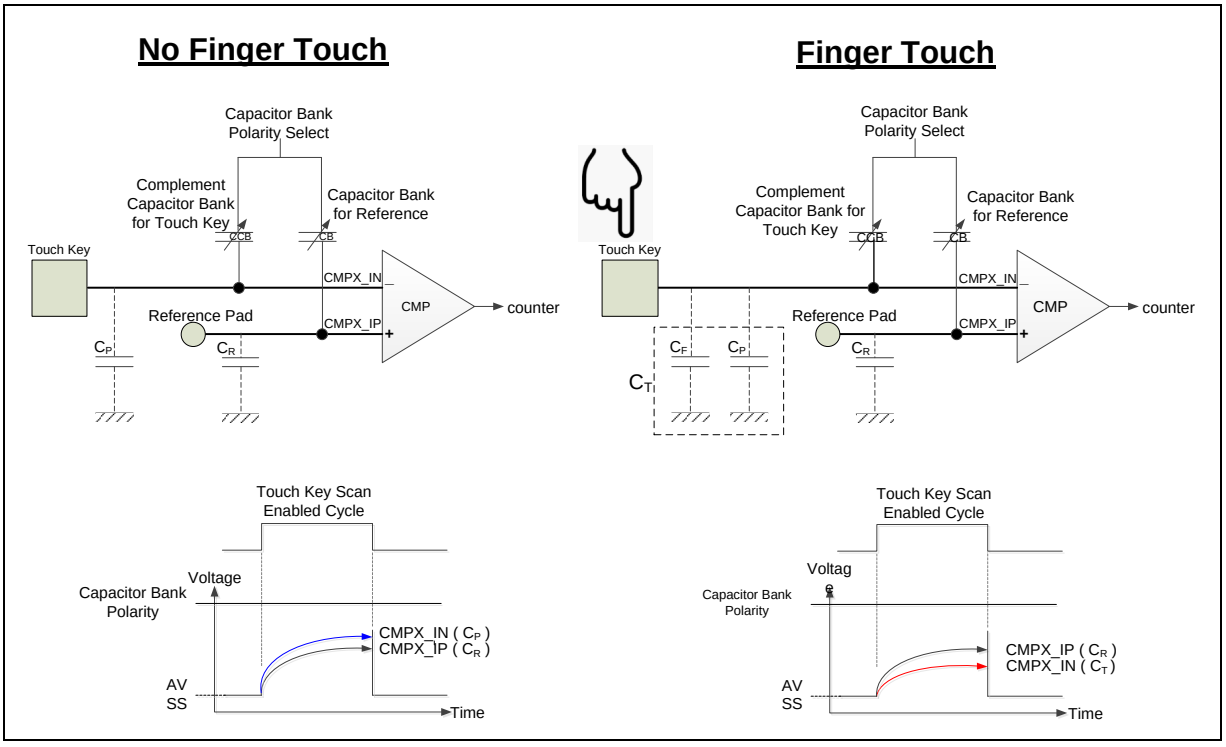


Figure 6.20-2 Touch Key Sensing Method

After touch key calibration, C_P and C_R are balanced with C_B and C_{CB}. A finger touches presents on sensing touch key results in C_T ($C_T = C_P + C_F$), makes negative input terminal voltage of the comparator is lower than positive side and comparator output is “high”. This means C_T and C_R are not balanced, and the touch key controller will increase C_{CB} to C_{CB}’ to make C_T and C_R are balanced again. A finger touch can be detected by checking the difference of C_{CB} and C_{CB}’, please refer to Figure 6.20-3.

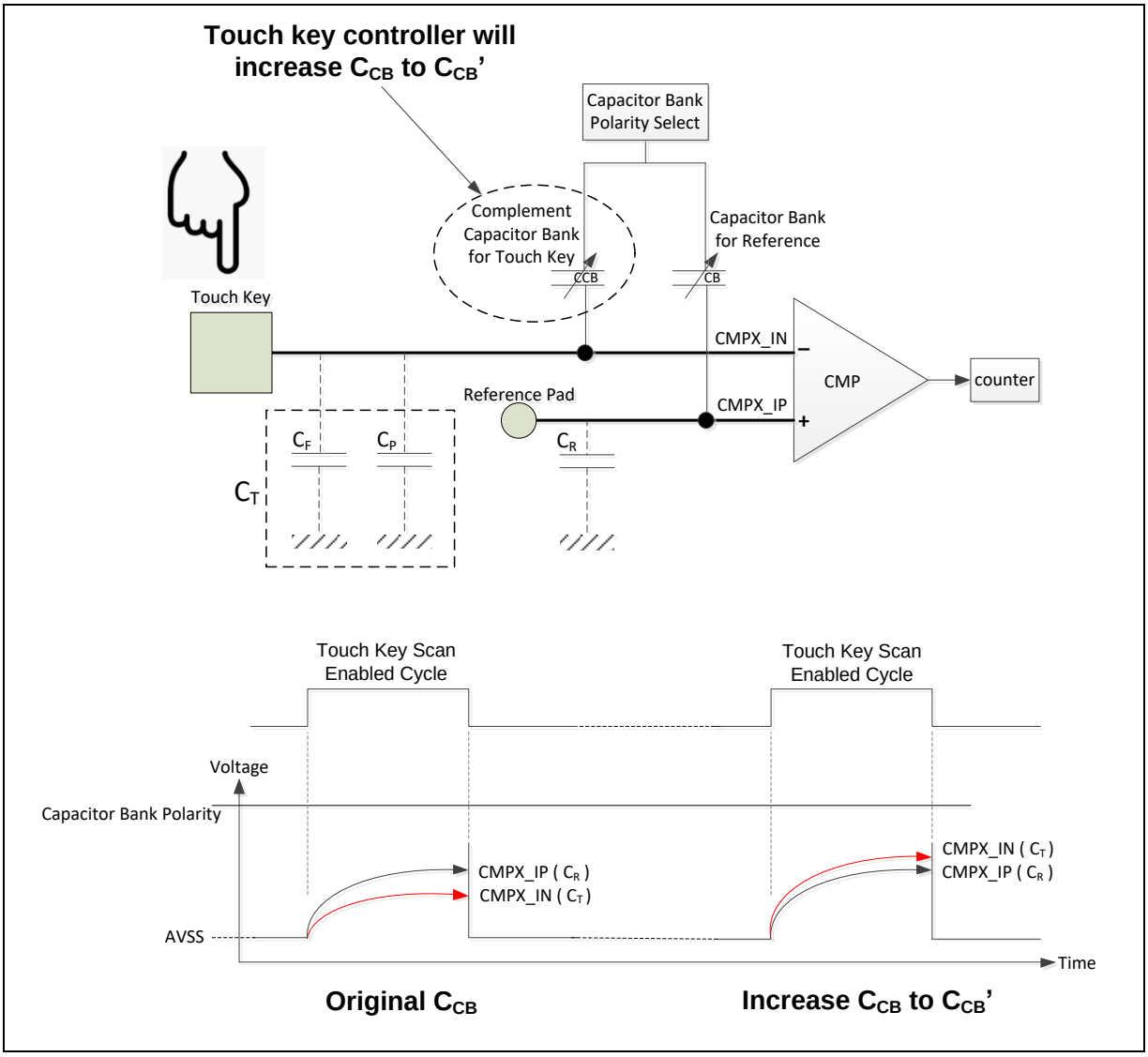


Figure 6.20-3 Finger Touch Detection Method

6.20.5.2 Touch Key Scan Mode

Single Scan Mode

In this mode, user must set TK_EN (TKCON0[2]) and SCAN (TKCON0[0]) to high and other bits of TKSENx~TKRENx must be set according to application requirements for sensitivity setting. Those channels enabled by corresponding bits of TKSENx register will be scanned successively when scan initiate. Once channel scanning starts, TKBUSY (TKSTA0[0]) will be set until the scanning is complete. The TKSCIF (TKSTA0[1]) register will be set when channel scanning completed and the sensed data in TKDATx registers is valid for reading, according to those channels enabled in TKSENx, x denotes number 0 to 14. TKSCIF (TKSTA0[1]) can generate interrupt by setting TKSCIE (TKINTEN[1]) register to high.

Periodic Scan Mode

This mode supports automatic scanning periodically. User can set TK_EN (TKCON0[2]) and TMRTRG_EN (TKCON0[1]) to high. In addition to set other bits of TKSENx~TKRENx according to application requirements, user must set RTC Time Stick properly for determining period between scanning. Besides, user can apply wake-up function to key scanning in a low power system. By setting threshold control properly, periodic scanning keeps even system is in sleep state, and until the system waked-up by interrupt which means whether a finger-touch is met the threshold or not. Key scanning stops until the interrupt flags reset.

6.20.5.3 Reference Channel for Touch Key Scanning

There must be one channel at least as reference channel which PCB layout has special concern. TK14 is assigned as default reference channel automatically if there is no reference channel internl source assigned by user. Touch Key controller malfunctions without assigning physical reference channel(s) to pin(s) in the application.

6.20.5.4 Idle State and Polarity Control

Those keys-pads are not being sensed always keep in Idle-State, their output level can be pre-determined in IDLSx(TKIDLPOLx[1:0]). For another sensing configuration, their output level is separated to two states: Idle-state and Polarity-State if their polarity control is activated in POLENx(TKIDLPOLx[4]). Their output level is pre-determined in POLx(TKIDLPOLCx[3:2]).

6.20.5.5 Sensing Time

Sensing Time is the time needed for each key-pad sensing, it is composed of PULSET (TKCON1[6:4]) and SENSET (TKCON1[2:0]). Shorter Sensing Time comes with poor sensitivity and less power-consumption, vice versa.

Sensing Time = SENSET x PULSET

6.20.5.6 Sensitivity Configuration

Sensitivity can be adjusted by setting Sensing Time properly. Besides, select capacitor bank polarity source properly which pre-determined in POL_CAP (TKCON2[1:0]) also affects sensitivity. User can choose $AV_{DD}H$ as capacitor bank polarity source to have more sensitivity choice by programming $AV_{DD}H$ to proper level, which is pre-determined in $AV_{DD}H\_S$ (TKCON0[7:4]).

6.20.5.7 Scan Interrupt Type

Scan Complete without Threshold Control Interrupt

TKSCIF (TKSTA0[1]) is always set when key-scan is complete. An interrupt is generated if TKSCIE (TKINTEN[1]) is set.

Scan Complete with Threshold Control Interrupt

Differ to TKSCIF (TKSTA0[1]), TKIFx (TKSTA1[7:0], TKSTA2[6:0]) is set if and only if corresponding scan result met its threshold control requirement. Besides, set TKSCTHIE (TKINTEN[0]) rather than TKSCIE (TKINTEN[1]) to generate interrupt when a potential key touch/release is detected. The high

threshold control for each channel can be pre-determined in TKHTHx registers, x denotes adjacent channels' number, for example, TKHTH0.

TKIFx is set if TKDATx is greater than TKHTHx. TKDATx greater than TKHTHx means a potential key touch occurs. As shown in Figure 6.20-4 Touch Key Controller Interrupt Modes For Threshold Control.

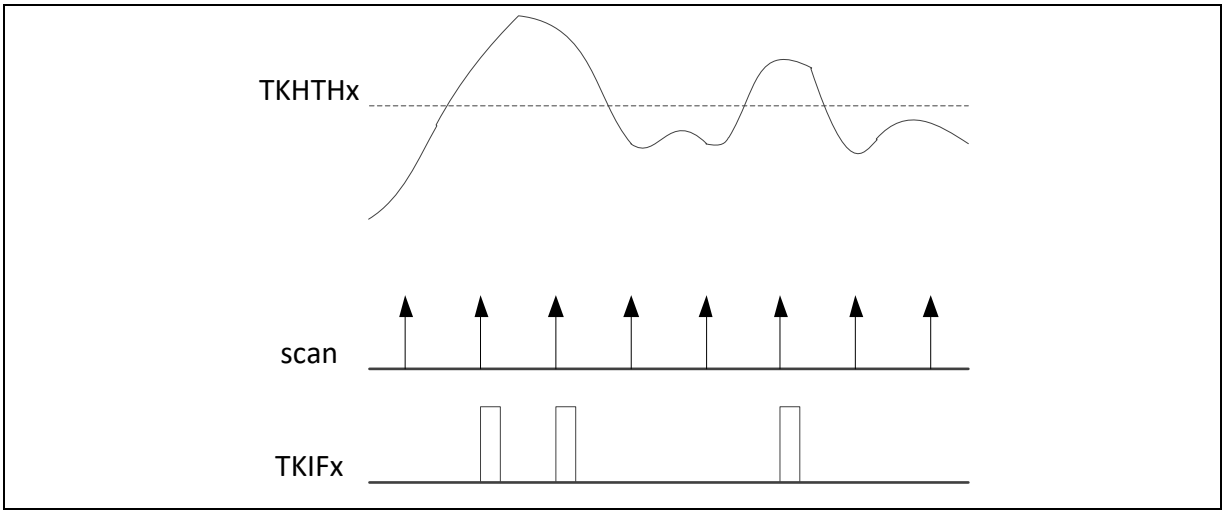


Figure 6.20-4 Touch Key Controller Interrupt Modes For Threshold Control.

6.20.5.8 Low Power Consumption Solution

Non-stop key scanning in a low power system can be easily achieved. User can use RTC to wake-up Touch Key controller for key scanning periodically. Touch Key controller requests **HIRC** for key scanning only when waked-up, and keeps CPU in power-down state. Interrupts generated when a potential key scanning event occurs, otherwise, Touch Key controller will terminate key scanning without interrupt and makes itself in power down state.

Wake-up by Key Touch / Release

To enable threshold control for interrupt generation, system keeps in low power until any potential key touch/release detected.

Wake-up by Any Key Touch

To save system more power consumption, user can use Any Key Wake-up function by setting SCAN_ALL (TKCON0[3]). All channels enabled but not assigned as reference channel are scanned, and scanning data is valid in TKDATALL. The CCBDALL may be different from normal and need to be calibrated individually. Proximity detection is also achievable by using this mode.

6.20.6 Register Description

TKCON0 – Touch Key Control 0 Register

Register	Memory Address	Reset Value
TKCON0	8000H	0111_000 b

7	6	5	4	3	2	1	0
$AV_{DD}H\_S$				SCAN_ALL	TK_EN	TMRTRG_EN	SCAN
R/W				R/W	R/W	R/W	R/W

Bit	Name	Description
[7:4]	$AV_{DD}H\_S$	**$AV_{DD}H$ Voltage Select** 0000 = 2/32 V_{DD}. 0001 = 4/32 V_{DD}. 0010 = 6/32 V_{DD}. 0011 = 8/32 V_{DD}. 0100 = 10/32 V_{DD}. 0101 = 12/32 V_{DD}. 0110 = 14/32 V_{DD}. 0111 = 16/32 V_{DD}. 1000 = 1/32 V_{DD}. 1001 = 2/32 V_{DD}. 1010 = 3/32 V_{DD}. 1011 = 4/32 V_{DD}. 1100 = 5/32 V_{DD}. 1101 = 6/32 V_{DD}. 1110 = 7/32 V_{DD}. 1111 = 8/32 V_{DD}.
[3]	SCAN_ALL	**All Keys Scan Enable** This function is used for low power key scanning operation. TKDATALL is the only one valid data when key scan is complete. 0 = Disable All Keys Scan function. 1 = Enable All Keys Scan function.
[2]	TK_EN	**Touch Key Scan Enable Bit** 0 = Disable Touch Key Function. 1 = Enable Touch Key Function.
[1]	TMRTRG_EN	**Timer Trigger Enable** 0 = Disable timer to trigger key scan. 1 = Enable timer triggers key scan periodically. Key scan will be initiated by timer periodically.
[0]	SCAN	**Scan** Write a “1’ to this bit will immediately initiate key scan on all channels which are enabled. This bit will be self-cleared after key scan started.

TKCON1 – Touch Key Control 1 Register

Register	Memory Address	Reset Value
TKCON1	8001H	0000_0000 b

7	6	5	4	3	2	1	0
-	PULSET			-	SENSET		
-	R/W			-	R/W		

Bit	Name	Description
[7]	-	Reserved.
[6:4]	PULSET	**Touch Key Sensing Pulse Width Time Control** 000 = 500ns. 001 = 1us. 010 = 2us. 011 = 4us. 100 = 8us. 101 = 8us. 110 = 8us 111 = 8us.
[3]	-	Reserved.
[2:0]	SENSET	**Touch Key Sensing Time Control** 000 = 16 x PULSET. 001 = 32 x PULSET. 010 = 64 x PULSET. 011 = 128 x PULSET. 100 = 255 x PULSET. 101 = 511 x PULSET. 110 = 1023 x PULSET. 111 = 1023 x PULSET.

TKCON2 – Touch Key Control 2 Register

Register	Memory Address	Reset Value
TKCON2	8002H	0000_0001 b

7	6	5	4	3	2	1	0
-					POL_INIT	POL_CAP	
-					R/W	R/W	

Bit	Name	Description
[7:3]	-	Reserved.
[2]	POL_INIT	**Touch Key Sensing Initial Potential Control** 0 = Key pad is connected to Gnd before sensing. 1 = Key pad is connected to $AV_{DD}H$ before sensing.
[1:0]	POL_CAP	**Capacitor Bank Polarity Select** 00 = Gnd. 01 = $AV_{DD}H$. 10 = Reserved. 11 = V_{DD}.

TKSEN0 – Touch Key Scan Enable 0 Register

Register	Memory Address	Reset Value
TKSEN0	8003H	0000_0000 b

7	6	5	4	3	2	1	0
TK7SEN	TK6SEN	TK5SEN	TK4SEN	TK3SEN	TK2SEN	TK1SEN	TK0SEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**TK7SEN**	**TK7 Scan Enable** This bit is ignored if TK7REN (TKREN0[7]) is “1”. 0 = TKDAT7 is invalid. 1 = TK7 is always enable for Touch Key scan. TKDAT7 is valid.
[6]	**TK6SEN**	**TK6 Scan Enable** This bit is ignored if TK6REN (TKREN0[6]) is “1”. 0 = TKDAT6 is invalid. 1 = TK6 is always enable for Touch Key scan. TKDAT6 is valid.
[5]	**TK5SEN**	**TK5 Scan Enable** This bit is ignored if TK5REN (TKREN0[5]) is “1”. 0 = TKDAT5 is invalid. 1 = TK5 is always enable for Touch Key scan. TKDAT5 is valid.
[4]	**TK4SEN**	**TK4 Scan Enable** This bit is ignored if TK4REN (TKREN0[4]) is “1”. 0 = TKDAT4 is invalid. 1 = TK4 is always enable for Touch Key scan. TKDAT4 is valid.
[3]	**TK3SEN**	**TK3 Scan Enable** This bit is ignored if TK3REN (TKREN0[3]) is “1”. 0 = TKDAT3 is invalid. 1 = TK3 is always enable for Touch Key scan. TKDAT3 is valid.
[2]	**TK2SEN**	**TK2 Scan Enable** This bit is ignored if TK2REN (TKREN0[2]) is “1”. 0 = TKDAT2 is invalid. 1 = TK2 is always enable for Touch Key scan. TKDAT2 is valid.
[1]	**TK1SEN**	**TK1 Scan Enable** This bit is ignored if TK1REN (TKREN0[1]) is “1”. 0 = TKDAT1 is invalid. 1 = TK1 is always enable for Touch Key scan. TKDAT1 is valid.
[0]	**TK0SEN**	**TK0 Scan Enable** This bit is ignored if TK0REN (TKREN0[0]) is “1” 0 = TKDAT0 is invalid. 1 = TK0 is always enable for Touch Key scan. TKDAT0 is valid.

TKSEN1 – Touch Key Scan Enable 1 Register

Register	Memory Address	Reset Value
TKSEN1	8004H	0000_0000 b

7	6	5	4	3	2	1	0
-	TK14SEN	TK13SEN	TK12SEN	TK11SEN	TK10SEN	TK9SEN	TK8SEN
-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	-	Reserved.
[6]	TK14SEN	**TK14 Scan Enable** This bit is ignored if TK14REN (TKREN1[6]) is “1”. 0 = TKDAT14 is invalid. 1 = TK14 is always enabled for key scan. TKDAT14 is valid.
[5]	TK13SEN	**TK13 Scan Enable** This bit is ignored if TK13REN (TKREN1[5]) is “1”. 0 = TKDAT13 is invalid. 1 = TK13 is always enable for key scan. TKDAT13 is valid.
[4]	TK12SEN	**TK12 Scan Enable** This bit is ignored if TK12REN (TKREN[4]) is “1”. 0 = TKDAT12 is invalid. 1 = TK12 is always enable for Touch Key scan. TKDAT12 is valid.
[3]	TK11SEN	**TK11 Scan Enable** This bit is ignored if TK11REN (TKREN1[3]) is “1”. 0 = TKDAT11 is invalid. 1 = TK11 is always enable for Touch Key scan. TKDAT11 is valid.
[2]	TK10SEN	**TK10 Scan Enable** This bit is ignored if TK10REN (TKREN1[2]) is “1”. 0 = TKDAT10 is invalid. 1 = TK10 is always enable for Touch Key scan. TKDAT10 is valid.
[1]	TK9SEN	**TK9 Scan Enable** This bit is ignored if TK9REN (TK_REN1[1]) is “1”. 0 = TKDAT9 is invalid. 1 = TK9 is always enable for Touch Key scan. TKDAT9 is valid.
[0]	TK8SEN	**TK8 Scan Enable** This bit is ignored if TK8REN (TK_REN1[0]) is “1”. 0 = TKDAT8 is invalid. 1 = TK8 is always enable for Touch Key scan. TKDAT8 is valid.

TKREN0 – Touch Key Reference Enable 0 Register

Register	Memory Address	Reset Value
TKREN0	8005H	0000_0000 b

7	6	5	4	3	2	1	0
TK7REN	TK6REN	TK5REN	TK4REN	TK3REN	TK2REN	TK1REN	TK0REN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	TK7REN	**TK7 Reference Enable** 0 = TK7 is not reference. 1 = TK7 is set as reference, and TKDAT7 is invalid.
[6]	TK6REN	**TK6 Reference Enable** 0 = TK6 is not reference. 1 = TK6 is set as reference, and TKDAT6 is invalid.
[5]	TK5REN	**TK5 Reference Enable** 0 = TK5 is not reference. 1 = TK5 is set as reference, and TKDAT5 is invalid.
[4]	TK4REN	**TK4 Reference Enable** 0 = TK4 is not reference. 1 = TK4 is set as reference, and TKDAT4 is invalid.
[3]	TK3REN	**TK3 Reference Enable** 0 = TK3 is not reference. 1 = TK3 is set as reference, and TKDAT3 is invalid.
[2]	TK2REN	**TK2 Reference Enable** 0 = TK2 is not reference. 1 = TK2 is set as reference, and TKDAT2 is invalid.
[1]	TK1REN	**TK1 Reference Enable** 0 = TK1 is not reference. 1 = TK1 is set as reference, and TKDAT1 is invalid.
[0]	TK0REN	**TK0 Reference Enable** 0 = TK0 is not reference. 1 = TK0 is set as reference, and TKDAT0 is invalid.

TKREN1 – Touch Key Reference Enable 1 Register

Register	Memory Address	Reset Value
TKREN1	8006H	0100_0000 b

7	6	5	4	3	2	1	0
-	TK14REN	TK13REN	TK12REN	TK11REN	TK10REN	TK9REN	TK8REN
-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	-	Reserved.
[6]	TK14REN	**TK14 Reference Enable** 0 = TK14 is not reference. 1 = TK14 is set as reference, and TKDAT14 is invalid.
[5]	TK13REN	**TK13 Reference Enable** 0 = TK13 is not reference. 1 = TK13 is set as reference, and TKDAT13 is invalid.
[4]	TK12REN	**TK12 Reference Enable** 0 = TK12 is not reference. 1 = TK12 is set as reference, and TKDAT12 is invalid.
[3]	TK11REN	**TK11 Reference Enable** 0 = TK11 is not reference. 1 = TK11 is set as reference, and TKDAT11 is invalid.
[2]	TK10REN	**TK10 Reference Enable** 0 = TK10 is not reference. 1 = TK10 is set as reference, and TKDAT10 is invalid.
[1]	TK9REN	**TK9 Reference Enable** 0 = TK9 is not reference. 1 = TK9 is set as reference, and TKDAT9 is invalid.
[0]	TK8REN	**TK8 Reference Enable** 0 = TK8 is not reference. 1 = TK8 is set as reference, and TKDAT8 is invalid.

TKINTEN – Touch Key Interrupt Enable Register

Register	Memory Address	Reset Value
TKINTEN	8007H	0000_0000 b

7	6	5	4	3	2	1	0
-						TKSCIE	TKSCTHIE
-						R/W	R/W

Bit	Name	Description
[7:2]	-	Reserved.
[1]	TKSCIE	**Touch Key Scan Complete Interrupt Enable** 0 = Key scan complete without threshold control interrupt is disable. 1 = Key scan complete without threshold control interrupt is enable.
[0]	TKSCTHIE	**Touch Key Scan Complete with High/Low Threshold Control Interrupt Enable** 0 = Key scan complete with threshold control interrupt is disable. 1 = Key scan complete with threshold control interrupt is enable.

TKSTA0 – Touch Key Status 0 Register

Register	Memory Address	Reset Value
TKSTA0	8008H	0000_0000 b

7	6	5	4	3	2	1	0
-				TKIF_ALL	TKIF	TKSCIF	TKBUSY
-				R/W	R/W	R/W	R/W

Bit	Name	Description
[7:4]	-	Reserved.
[3]	TKIF_ALL	**All Keys Scan Interrupt Flag** 0 = No threshold control event with All Keys Scan. 1 = Threshold control event occurs with All Keys Scan. This bit will be set 1 only when SCAN_ALL bit enabled. and into TK interrupt if TKSCTHIE (TKINTEN.0) bit is enabled.
[2]	TKIF	**Key Scan Interrupt Flag (Read Only)** 0 = No threshold control event with each Key Scan. 1 = Threshold control event occurs with each Keys Scan. This bit is 1 while any one of TKIF0~TKIF14 is 1.
[1]	TKSCIF	**Touch Key Scan Complete Interrupt Flag** 0 = Key scan is proceeding and data is not ready for read. 1 = Key scan is complete and data is ready for read in TKDATx registers. **Note 1:** The Touch Key interrupt asserts if TKSCIE bit of TKINTEN register is set. **Note 2:** The Touch Key interrupt also asserts if STHIE bit of TKINTEN register is set and any channel data value is greater/less than its threshold setting.
[0]	TKBUSY	**Touch Key Busy (Read Only)** 0 = Key scan is complete or stopped. 1 = Key scan is proceeding.

TKSTA1 – Touch Key Status 1 Register

Register	Memory Address	Reset Value
TKSTA1	8009H	0000_0000 b

7	6	5	4	3	2	1	0
TKIF7	**TKIF6**	**TKIF5**	**TKIF4**	**TKIF3**	**TKIF2**	**TKIF1**	**TKIF0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	**TKIF7**	**TK7 Interrupt Flag** 0 = No threshold control event with TK7. 1 = Threshold control event occurs with TK7.
[6]	**TKIF6**	**TK6 Interrupt Flag** 0 = No threshold control event with TK6. 1 = Threshold control event occurs with TK6.
[5]	**TKIF5**	**TK5 Interrupt Flag** 0 = No threshold control event with TK5. 1 = Threshold control event occurs with TK5.
[4]	**TKIF4**	**TK4 Interrupt Flag** 0 = No threshold control event with TK4. 1 = Threshold control event occurs with TK4.
[3]	**TKIF3**	**TK3 Interrupt Flag** 0 = No threshold control event with TK3. 1 = Threshold control event occurs with TK3.
[2]	**TKIF2**	**TK2 Interrupt Flag** 0 = No threshold control event with TK2. 1 = Threshold control event occurs with TK2.
[1]	**TKIF1**	**TK1 Interrupt Flag** 0 = No threshold control event with TK1. 1 = Threshold control event occurs with TK1.
[0]	**TKIF0**	**TK0 Interrupt Flag** 0 = No threshold control event with TK0. 1 = Threshold control event occurs with TK0.

TKSTA2 – Touch Key Status 2 Register

Register	Memory Address	Reset Value
TKSTA2	800AH	0000_0000 b

7	6	5	4	3	2	1	0
-	TKIF14	TKIF13	TKIF12	TKIF11	TKIF10	TKIF9	TKIF8
-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7]	-	Reserved.
[6]	TKIF14	**TK14 Interrupt Flag** 0 = No threshold control event with TK14. 1 = Threshold control event occurs with TK14.
[5]	TKIF13	**TK13 Interrupt Flag** 0 = No threshold control event with TK13. 1 = Threshold control event occurs with TK13.
[4]	TKIF12	**TK12 Interrupt Flag** 0 = No threshold control event with TK12. 1 = Threshold control event occurs with TK12.
[3]	TKIF11	**TK11 Interrupt Flag** 0 = No threshold control event with TK11. 1 = Threshold control event occurs with TK11.
[2]	TKIF10	**TK10 Interrupt Flag** 0 = No threshold control event with TK10. 1 = Threshold control event occurs with TK10.
[1]	TKIF9	**TK9 Interrupt Flag** 0 = No threshold control event with TK9. 1 = Threshold control event occurs with TK9.
[0]	TKIF8	**TK8 Interrupt Flag** 0 = No threshold control event with TK8. 1 = Threshold control event occurs with TK8.

TKCCBD0~14 – Touch Key Complement Capacitor Bank Data Register

Register	Memory Address	Reset Value
TKCCBD0	8010H	0000_0000 b
TKCCBD1	8011H	0000_0000 b
TKCCBD2	8012H	0000_0000 b
TKCCBD3	8013H	0000_0000 b
TKCCBD4	8014H	0000_0000 b
TKCCBD5	8015H	0000_0000 b
TKCCBD6	8016H	0000_0000 b
TKCCBD7	8017H	0000_0000 b
TKCCBD8	8018H	0000_0000 b
TKCCBD9	8019H	0000_0000 b
TKCCBD10	801AH	0000_0000 b
TKCCBD11	801BH	0000_0000 b
TKCCBD12	801CH	0000_0000 b
TKCCBD13	801DH	0000_0000 b
TKCCBD14	801EH	0000_0000 b

7	6	5	4	3	2	1	0
CCBDx							
R/W							

Bit	Name	Description
[7:0]	**CCBDx**	**TKx Complement CB Data** This is register is used for TKx sensitivity adjustment, where x = 0 ~14

TKCCBDALL – Touch Key Complement Capacitor Bank Data Register

Register	Memory Address	Reset Value
TKCCBDALL	801FH	0000_0000 b

7	6	5	4	3	2	1	0
CCBDALL							
R/W							

Bit	Name	Description
[7:0]	**CCBDALL**	**TK All Scans Complement CB Data** This is register is used for TK all scans sensitivity adjustment

REFCBD0~14 – Reference Capacitor Bank Data Register

Register	Memory Address	Reset Value
REFCBD0	8020H	0000_0000 b
REFCBD1	8021H	0000_0000 b
REFCBD2	8022H	0000_0000 b
REFCBD3	8023H	0000_0000 b
REFCBD4	8024H	0000_0000 b
REFCBD5	8025H	0000_0000 b
REFCBD6	8026H	0000_0000 b
REFCBD7	8027H	0000_0000 b
REFCBD8	8028H	0000_0000 b
REFCBD9	8029H	0000_0000 b
REFCBD10	802AH	0000_0000 b
REFCBD11	802BH	0000_0000 b
REFCBD12	802CH	0000_0000 b
REFCBD13	802DH	0000_0000 b
REFCBD14	802EH	0000_0000 b

7	6	5	4	3	2	1	0
REFCBDx							
R/W							

Bit	Name	Description
[7:0]	**REFCBDx**	**Touch Key x Reference CB Data** This is register is used for Touch Key x Reference sensitivity adjustment, where x = 0 ~14

REFCBDALL – Reference Capacitor Bank Data Register

Register	Memory Address	Reset Value
REFCBDALL	802FH	0000_0000 b

7	6	5	4	3	2	1	0
REFCBDALL							
R/W							

Bit	Name	Description
[7:0]	**REFCBDALL**	**Touch Key All Scans Reference CB Data** This is register is used for Touch Key All Scans Reference sensitivity adjustment.

TKIDLPOL0~14 – Touch Key Idle State / Polarity Select Register

Register	Memory Address	Reset Value
TKIDLPOL0	8030H	1100_0000 b
TKIDLPOL1	8031H	1100_0000 b
TKIDLPOL2	8032H	1100_0000 b
TKIDLPOL3	8033H	1100_0000 b
TKIDLPOL4	8034H	1100_0000 b
TKIDLPOL5	8035H	1100_0000 b
TKIDLPOL6	8036H	1100_0000 b
TKIDLPOL7	8037H	1100_0000 b
TKIDLPOL8	8038H	1100_0000 b
TKIDLPOL9	8039H	1100_0000 b
TKIDLPOL10	803AH	1100_0000 b
TKIDLPOL11	803BH	1100_0000 b
TKIDLPOL12	803CH	1100_0000 b
TKIDLPOL13	803DH	1100_0000 b
TKIDLPOL14	803EH	1100_0000 b

7	6	5	4	3	2	1	0
-			POLENx	POLx		IDLSx	
-			R/W	R/W		R/W	

Bit	Name	Description
[7:5]	-	Reserved.
[4]	POLENx	**TKx Polarity Function Enable** 0 = Disable. 1 = Enable. Where x = 0~14
[3:2]	POLx	**TKx Polarity Select** This register is ignored if POLENx is “0” or TKxREN is “1”. 00 = TKx connected to Gnd. 01 = TKx connected to $AV_{DD}H$. 10 = Reserved. 11 = TKx connected to V_{DD}. Where x = 0~14

Bit	Name	Description
[1:0]	**IDLSx**	**TKx Idle State Control** This register is ignored if both TKxSEN and POLENx are “0” or TKxREN is “1”. 00 = TKx connected to Gnd. 01 = TKx connected to $AV_{DD}H$. 10 = Reserved. 11 = TKx connected to V_{DD}. Where x = 0~14

TKDATx – Touch Key x Data Register

Register	Memory Address	Reset Value
TKDAT0	8040H	0000_0000 b
TKDAT1	8041H	0000_0000 b
TKDAT2	8042H	0000_0000 b
TKDAT3	8043H	0000_0000 b
TKDAT4	8044H	0000_0000 b
TKDAT5	8045H	0000_0000 b
TKDAT6	8046H	0000_0000 b
TKDAT7	8047H	0000_0000 b
TKDAT8	8048H	0000_0000 b
TKDAT9	8049H	0000_0000 b
TKDAT10	804AH	0000_0000 b
TKDAT11	804BH	0000_0000 b
TKDAT12	804CH	0000_0000 b
TKDAT13	804DH	0000_0000 b
TKDAT14	804EH	0000_0000 b

7	6	5	4	3	2	1	0
TKDATx							
R							

Bit	Name	Description
[7:0]	**TKDATx**	**TKx Sensing Result Data (Read Only)** This data is invalid if TKxSEN is “0” or TKxREN is “1” except SCAN_ALL (TKCON0[3]) is “1”. Where x = 0~14

TKDATALL – Touch Key x Data Register

Register	Memory Address	Reset Value
TKDATALL	804FH	0000_0000 b

7	6	5	4	3	2	1	0
TKDATALL							
R							

Bit	Name	Description
[7:0]	**TKDATALL**	**All Keys Scan Sensing Result Data (Read Only)** This data is invalid if SCAN_ALL (TKCON0[3]) is “0”.

TKHTHx – Touch Key x High Threshold Register

Register	Memory Address	Reset Value
TKHTH0	8050H	1111_1111 b
TKHTH1	8051H	1111_1111 b
TKHTH2	8052H	1111_1111 b
TKHTH3	8053H	1111_1111 b
TKHTH4	8054H	1111_1111 b
TKHTH5	8055H	1111_1111 b
TKHTH6	8056H	1111_1111 b
TKHTH7	8057H	1111_1111 b
TKHTH8	8058H	1111_1111 b
TKHTH9	8059H	1111_1111 b
TKHTH10	805AH	1111_1111 b
TKHTH11	805BH	1111_1111 b
TKHTH12	805CH	1111_1111 b
TKHTH13	805DH	1111_1111 b
TKHTH14	805EH	1111_1111 b

7	6	5	4	3	2	1	0
HTHx							
R/W							

Bit	Name	Description
[7:0]	**HTHx**	**High Threshold of TKx** High level for TKx threshold control. Where x= 0~14

TKHTHALL – Touch Key x High Threshold Register

Register	Memory Address	Reset Value
TKHTHALL	805FH	1111_1111 b

7	6	5	4	3	2	1	0
HTH_ALL							
R/W							

Bit	Name	Description
[7:0]	**HTH_ALL**	**High Threshold of All Keys Scan** High level for All Keys Scan threshold control.

6.21 Instruction Set

6.21.1 Addressing Modes

The ML51/ML54/ML56 Series executes all the instructions of the standard 80C51 family fully compatible with MCS-51. However, the timing of each instruction is different for it uses high performance 1T 8051 core. The architecture eliminates redundant bus states and implements parallel execution of fetching, decode, and execution phases. The ML51/ML54/ML56 Series uses one clock per machine-cycle. It leads to performance improvement of rate 8.1 (in terms of MIPS) with respect to traditional 12T 80C51 device working at the same clock frequency. However, the real speed improvement seen in any system will depend on the instruction mix.

All instructions are coded within an 8-bit field called an OPCODE. This single byte should be fetched from Program Memory. The OPCODE is decoded by the CPU. It determines what action the CPU will take and whether more operation data is needed from memory. If no other data is needed, then only one byte was required. Thus the instruction is called a one byte instruction. In some cases, more data is needed, which is two or three byte instructions.

Following lists all instructions for details. The note of the instruction set and addressing modes are shown below.

Rn (N = 0~7)	Register R0 To R7 Of The Currently Selected Register Bank.
Direct	8-bit internal data location's address. It could be an internal data RAM location (00H to 7FH) or an SFR (80H to FFH).
@RI (I = 0, 1)	8-bit internal data RAM location (00H to FFH) addressed indirectly through register R0 or R1.
#data	8-bit constant included in the instruction.
#data16	16-bit constant included in the instruction.
Addr16	16-bit destination address. Used by LCALL and LJMP. A branch can be any-where within the Program Memory address space.
Addr11	11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2K-Byte page of Program Memory as the first byte of the following instruction.
Rel	Signed (2's complement) 8-bit offset Byte. Used by SJMP and all conditional branches. The range is -128 to +127 bytes relative to first byte of the following instruction.
Bit	Direct addressed bit in internal data RAM or SFR.

Table 6.21-1 Instruction Set And Addressing Modes

Instruction	CY	OV	AC	Instruction	CY	OV	AC
ADD	X[1]	X	X	CLR C	0		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C, bit	X		
MUL	0	X		ANL C, /bit	X		
DIV	0	X		ORL C, bit	X		
DA A	X			ORL C, /bit	X		
RRC A	X			MOV C, bit	X		
RLC A	X			CJNE	X		
SETB C	1						
Note: X indicates the modification depends on the result of the instruction.							

Table 6.21-2 Instructions Affect Flag Settings

6.21.2 Read-Modify-Write Instructions

Instructions that read a byte from SFR or internal RAM, modify it, and rewrite it back, are called “Read-Modify-Write” instructions. When the destination is an I/O port or a port bit, these instructions read the internal output latch rather than the external pin state. This kind of instructions read the port SFR value, modify it and write back to the port SFR. All “Read-Modify-Write” instructions are listed as follows.

Instruction	Description
ANL	Logical AND. (ANL direct, A and ANL direct, #data)
ORL	Logical OR. (ORL direct, A and ORL direct, #data)
XRL	Logical exclusive OR. (XRL direct, A and XRL direct, #data)
JBC	Jump if bit = 1 and clear it. (JBC bit, rel)
CPL	Complement bit. (CPL bit)
INC	Increment. (INC direct)
DEC	Decrement. (DEC direct)
DJNZ	Decrement and jump if not zero. (DJNZ direct, rel)
MOV	bit, C Move carry to bit. (MOV bit, C)
CLR	bit Clear bit. (CLR bit)
SETB	bit Set bit. (SETB bit)

The last three seem not obviously “Read-Modify-Write” instructions but actually they are. They read the entire port latch value, modify the changed bit, and then write the new value back to the port latch.

6.21.3 Instruction Set List Table

Instruction	OPCODE	Bytes	Clock Cycles	ML51/ML54/ML56 Series VS. Tradition 80C51 Speed Ratio
NOP	00	1	1	12
ADD A, Rn	28~2F	1	2	6
ADD A, direct	25	2	3	4
ADD A, @Ri	26, 27	1	4	3
ADD A, #data	24	2	2	6
ADDC A, Rn	38~3F	1	2	6
ADDC A, direct	35	2	3	4
ADDC A, @Ri	36, 37	1	4	3
ADDC A, #data	34	2	2	6
SUBB A, Rn	98~9F	1	2	6
SUBB A, direct	95	2	3	4
SUBB A, @Ri	96, 97	1	4	3
SUBB A, #data	94	2	2	6
INCA	04	1	1	12
INCRn	08~0F	1	3	4

Instruction	OPCODE	Bytes	Clock Cycles	ML51/ML54/ML56 Series VS. Tradition 80C51 Speed Ratio
INC direct	05	2	4	3
INC @Ri	06, 07	1	5	2.4
INC DPTR	A3	1	1	24
DEC A	14	1	1	12
DEC Rn	18~1F	1	3	4
DEC direct	15	2	4	3
DEC @Ri	16, 17	1	5	2.4
MUL AB	A4	1	4	12
DIV AB	84	1	4	12
DA A	D4	1	1	12
ANL A, Rn	58~5F	1	2	6
ANL A, direct	55	2	3	4
ANL A, @Ri	56, 57	1	4	3
ANL A, #data	54	2	2	6
ANL direct, A	52	2	4	3
ANL direct, #data	53	3	4	6
ORL A, Rn	48~4F	1	2	6
ORL A, direct	45	2	3	4
ORL A, @Ri	46, 47	1	4	3
ORL A, #data	44	2	2	6
ORL direct, A	42	2	4	3
ORL direct, #data	43	3	4	6
XRL A, Rn	68~6F	1	2	6
XRL A, direct	65	2	3	4
XRL A, @Ri	66, 67	1	4	3
XRL A, #data	64	2	2	6
XRL direct, A	62	2	4	3
XRL direct, #data	63	3	4	6
CLR A	E4	1	1	12
CPL A	F4	1	1	12
RL A	23	1	1	12
RLC A	33	1	1	12
RR A	03	1	1	12
RRC A	13	1	1	12
SWAP A	C4	1	1	12
MOV A, Rn	E8~EF	1	1	12
MOV A, direct	E5	2	3	4
MOV A, @Ri	E6, E7	1	4	3
MOV A, #data	74	2	2	6
MOV Rn, A	F8~FF	1	1	12
MOV Rn, direct	A8~AF	2	4	6

Instruction	OPCODE	Bytes	Clock Cycles	ML51/ML54/ML56 Series VS. Tradition 80C51 Speed Ratio
MOV Rn, #data	78~7F	2	2	6
MOV direct, A	F5	2	2	6
MOV direct, Rn	88~8F	2	3	8
MOV direct, direct	85	3	4	6
MOV direct, @Ri	86, 87	2	5	4.8
MOV direct, #data	75	3	3	8
MOV @Ri, A	F6, F7	1	3	4
MOV @Ri, direct	A6, A7	2	4	6
MOV @Ri, #data	76, 77	2	3	6
MOV DPTR, #data16	90	3	3	8
MOVC A, @A+DPTR	93	1	4	6
MOVC A, @A+PC	83	1	4	6
MOVX A, @Ri[1]	E2, E3	1	5	4.8
MOVX A, @DPTR[1]	E0	1	4	6
MOVX @Ri, A[1]	F2, F3	1	6	4
MOVX @DPTR, A[1]	F0	1	5	4.8
PUSH direct	C0	2	4	6
POP direct	D0	2	3	8
XCH A, Rn	C8~CF	1	2	6
XCH A, direct	C5	2	3	4
XCH A, @Ri	C6, C7	1	4	3
XCHD A, @Ri	D6, D7	1	5	2.4
CLR C	C3	1	1	12
CLR bit	C2	2	4	3
SETB C	D3	1	1	12
SETB bit	D2	2	4	3
CPL C	B3	1	1	12
CPL bit	B2	2	4	3
ANL C, bit	82	2	3	8
ANL C, /bit	B0	2	3	8
ORL C, bit	72	2	3	8
ORL C, /bit	A0	2	3	8
MOV C, bit	A2	2	3	4
MOV bit, C	92	2	4	6
ACALL addr11	11, 31, 51, 71, 91, B1, D1, F1[2]	2	4	6
LCALL addr16	12	3	4	6
RET	22	1	5	4.8
RETI	32	1	5	4.8
AJMP addr11	01, 21, 41, 61, 81, A1, C1, E1[3]	2	3	8
LJMP addr16	02	3	4	6

Instruction	OPCODE	Bytes	Clock Cycles	ML51/ML54/ML56 Series VS. Tradition 80C51 Speed Ratio
SJMP rel	80	2	3	8
JMP @A+DPTR	73	1	3	8
JZ rel	60	2	3	8
JNZ rel	70	2	3	8
JC rel	40	2	3	8
JNC rel	50	2	3	8
JB bit, rel	20	3	5	4.8
JNB bit, rel	30	3	5	4.8
JBC bit, rel	10	3	5	4.8
CJNE A, direct, rel	B5	3	5	4.8
CJNE A, #data, rel	B4	3	4	6
CJNE Rn, #data, rel	B8~BF	3	4	6
CJNE @Ri, #data, rel	B6, B7	3	6	4
DJNZ Rn, rel	D8~DF	2	4	6
DJNZ direct, rel	D5	3	5	4.8

Note:
1. The ML51/ML54/ML56 Series does not have external memory bus. MOVX instructions are used to access internal XRAM.
2. The most three significant bits in the 11-bit address [A10:A8] decide the ACALL hex code. The code will be [A10, A9, A8, 1, 0, 0, 0, 1].
3. The most three significant bits in the 11-bit address [A10:A8] decide the AJMP hex code. The code will be [A10, A9, A8, 0, 0, 0, 0, 1].

Table 6.21-3 Instruction Set

7 APPLICATION CIRCUIT

7.1 Power Supply Scheme

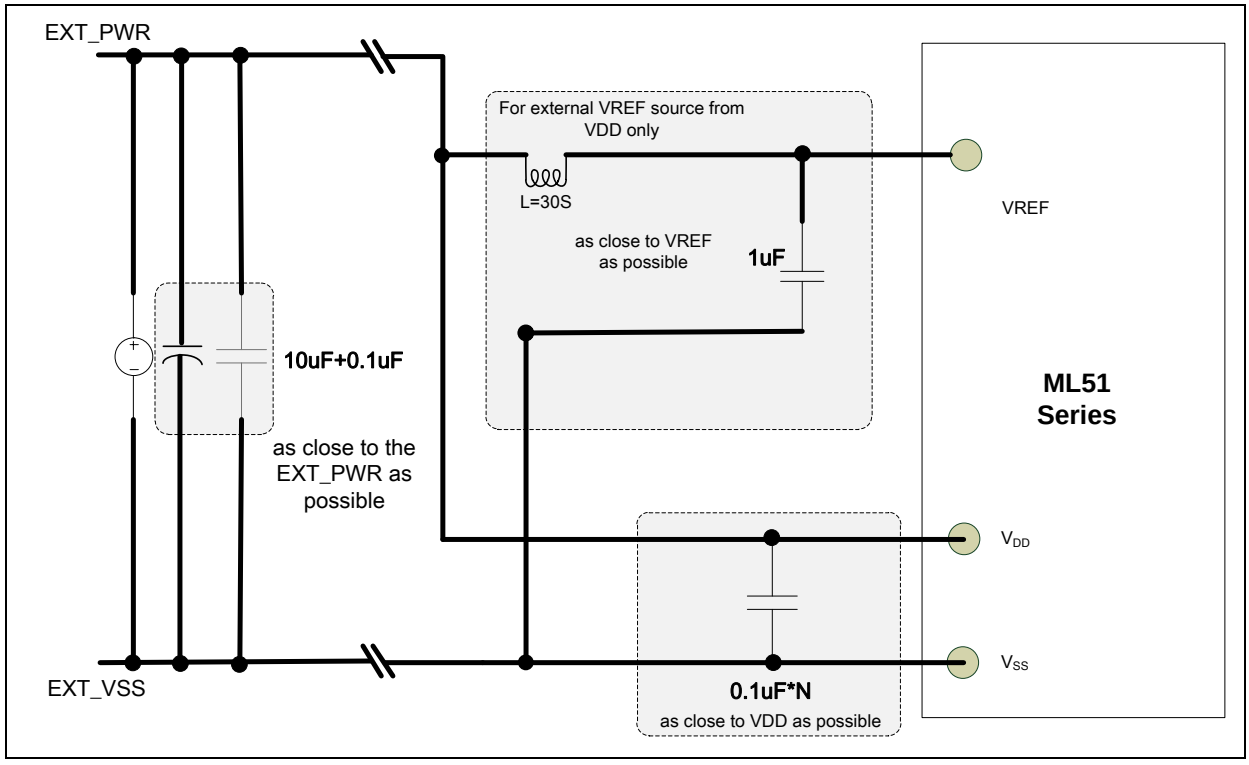
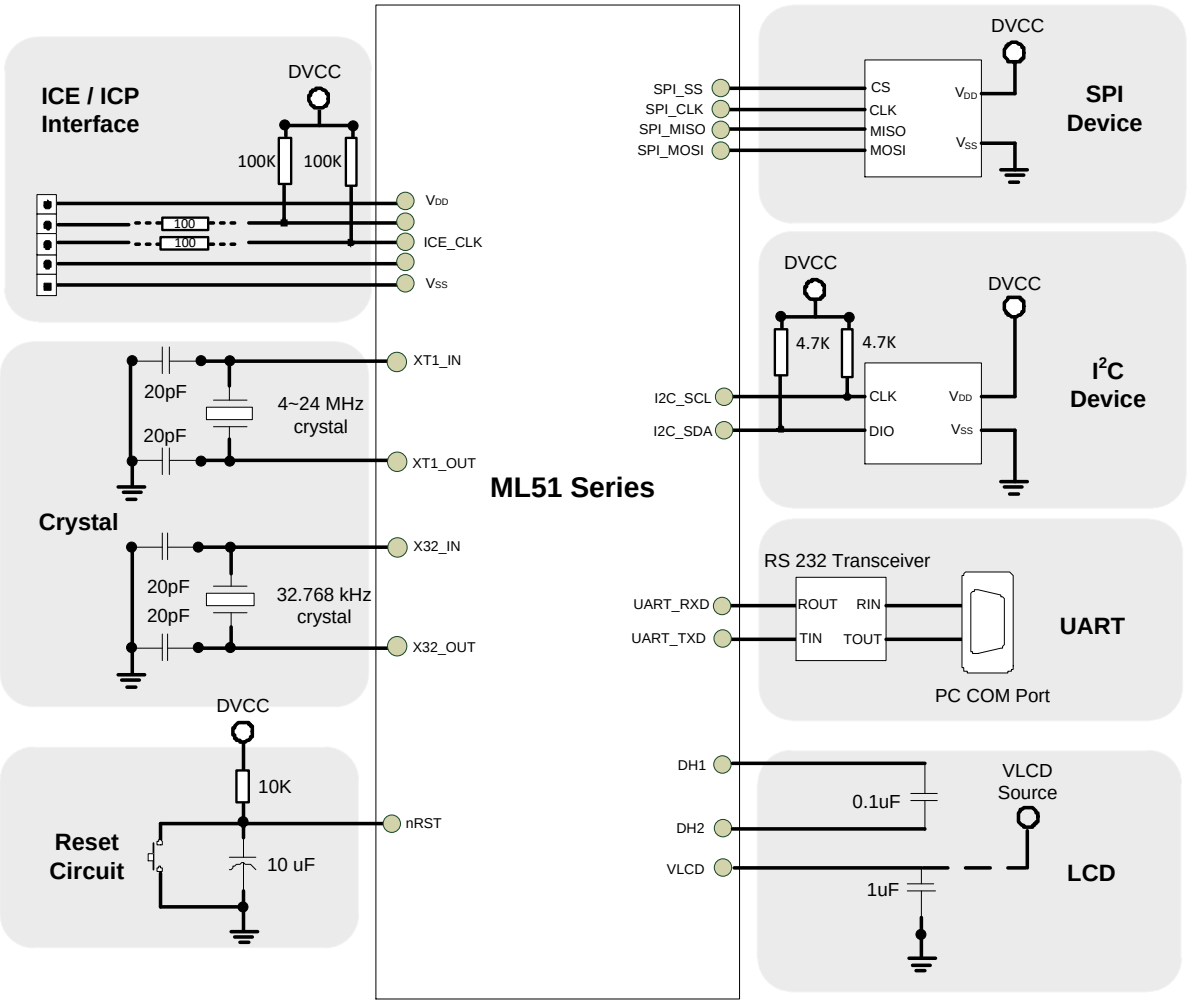


Figure 7.1 NuMicro® ML51/ML54/ML56 Series Power supply circuit

7.2 Peripheral Application Scheme



Note 1: It is recommended to use 100 kΩ pull-up resistor on both ICE_DAT and ICE_CLK pin.

Note 2: It is recommended to use 10 kΩ pull-up resistor and 10 uF capacitor on nRESET pin.

Note 3: It is suggest add 100ohm series resistor between ICE_DAT/ICE_CLK and writer pin to filter the disturb of noise on the circuit.

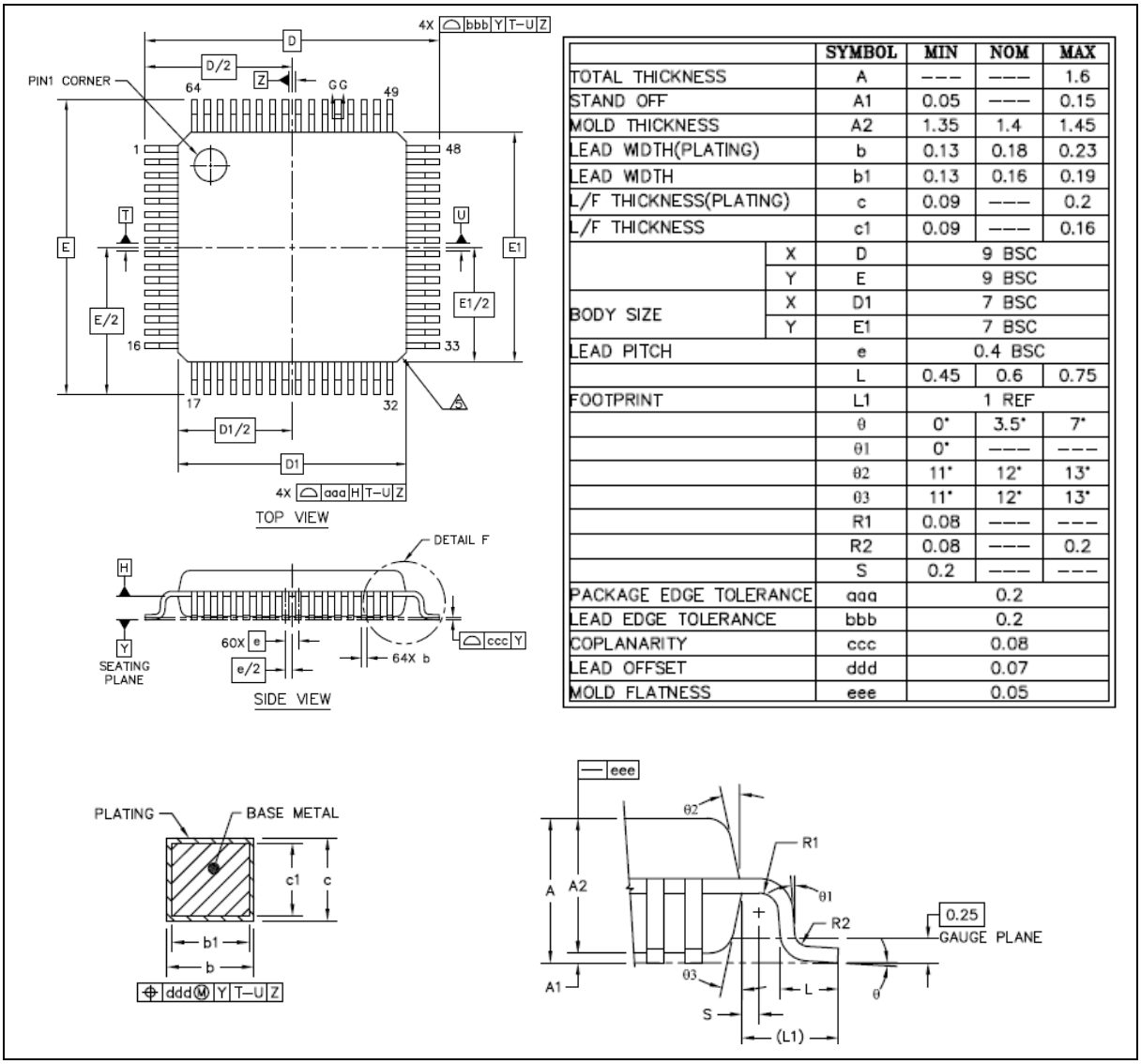
Figure 7.2 NuMicro® ML51/ML54/ML56 Series Peripheral interface circuit

8 ELECTRICAL CHARACTERISTICS

Please refer to the relative Datasheet for detailed information about the ML51/ML54/ML56 Series electrical characteristics.

9 PACKAGE DIMENSIONS

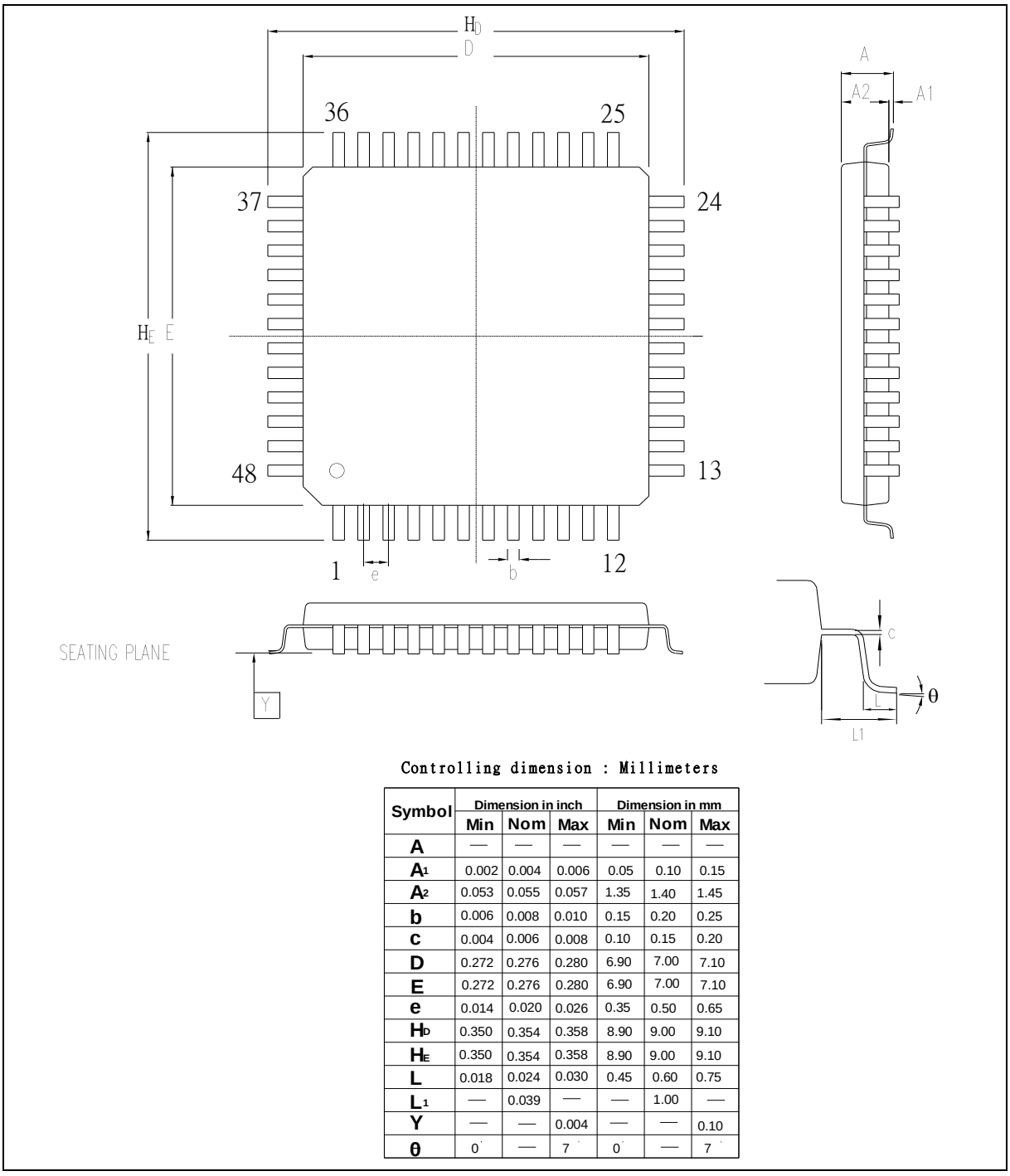
9.1 LQFP 64L-pin (7.0 x 7.0 x 1.4 mm)



		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	---	---	1.6
STAND OFF		A1	0.05	---	0.15
MOLD THICKNESS		A2	1.35	1.4	1.45
LEAD WIDTH(PLATING)		b	0.13	0.18	0.23
LEAD WIDTH		b1	0.13	0.16	0.19
L/F THICKNESS(PLATING)		c	0.09	---	0.2
L/F THICKNESS		c1	0.09	---	0.16
BODY SIZE	X	D	9 BSC		
	Y	E	9 BSC		
	X	D1	7 BSC		
	Y	E1	7 BSC		
LEAD PITCH		e	0.4 BSC		
		L	0.45	0.6	0.75
FOOTPRINT		L1	1 REF		
		θ	0°	3.5°	7°
		θ1	0°	---	---
		θ2	11°	12°	13°
		θ3	11°	12°	13°
		R1	0.08	---	---
		R2	0.08	---	0.2
		S	0.2	---	---
PACKAGE EDGE TOLERANCE		aaa	0.2		
LEAD EDGE TOLERANCE		bbb	0.2		
COPLANARITY		ccc	0.08		
LEAD OFFSET		ddd	0.07		
MOLD FLATNESS		eee	0.05		

Figure 9.1-1 LQFP 64L Package Dimension

9.2 LQFP 48-pin (7.0 x 7.0 x 1.4 mm)

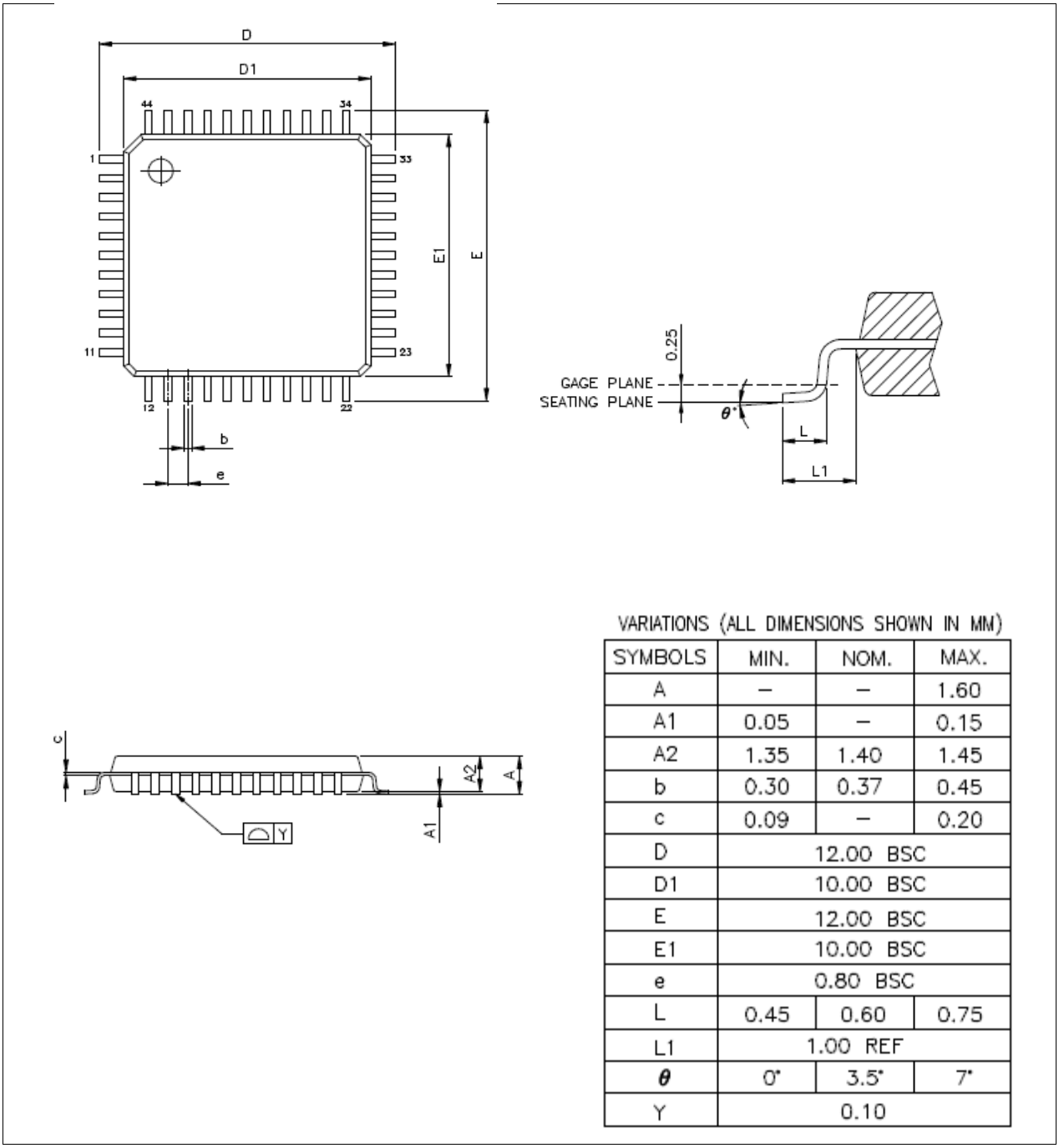


Controlling dimension : Millimeters

Symbol	Dimension in inch			Dimension in mm		
	Min	Nom	Max	Min	Nom	Max
A	—	—	—	—	—	—
A_1	0.002	0.004	0.006	0.05	0.10	0.15
A_2	0.053	0.055	0.057	1.35	1.40	1.45
b	0.006	0.008	0.010	0.15	0.20	0.25
c	0.004	0.006	0.008	0.10	0.15	0.20
D	0.272	0.276	0.280	6.90	7.00	7.10
E	0.272	0.276	0.280	6.90	7.00	7.10
e	0.014	0.020	0.026	0.35	0.50	0.65
H_D	0.350	0.354	0.358	8.90	9.00	9.10
H_E	0.350	0.354	0.358	8.90	9.00	9.10
L	0.018	0.024	0.030	0.45	0.60	0.75
L_1	—	0.039	—	—	1.00	—
Y	—	—	0.004	—	—	0.10
θ	0°	—	7°	0°	—	7°

Figure 9.2-1 LQFP-48 Package Dimension

9.3 LQFP 44-pin (10 x 10 x 1.4mm

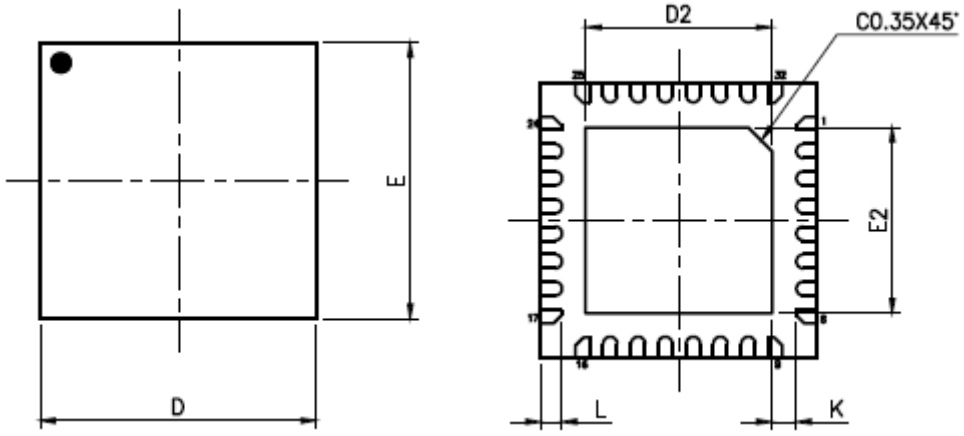


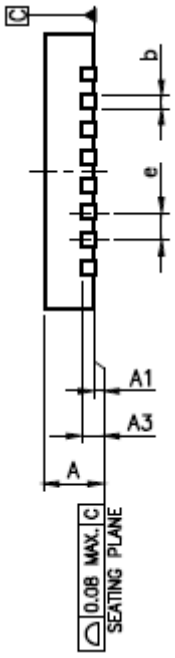
VARIATIONS (ALL DIMENSIONS SHOWN IN MM)

SYMBOLS	MIN.	NOM.	MAX.
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
b	0.30	0.37	0.45
c	0.09	–	0.20
D	12.00 BSC		
D1	10.00 BSC		
E	12.00 BSC		
E1	10.00 BSC		
e	0.80 BSC		
L	0.45	0.60	0.75
L1	1.00 REF		
θ	0°	3.5°	7°
Y	0.10		

Figure 9.3-1 LFP44 Package Dimension

9.4 QFN 33-pin (4.0 x 4.0 x 0.8 mm)

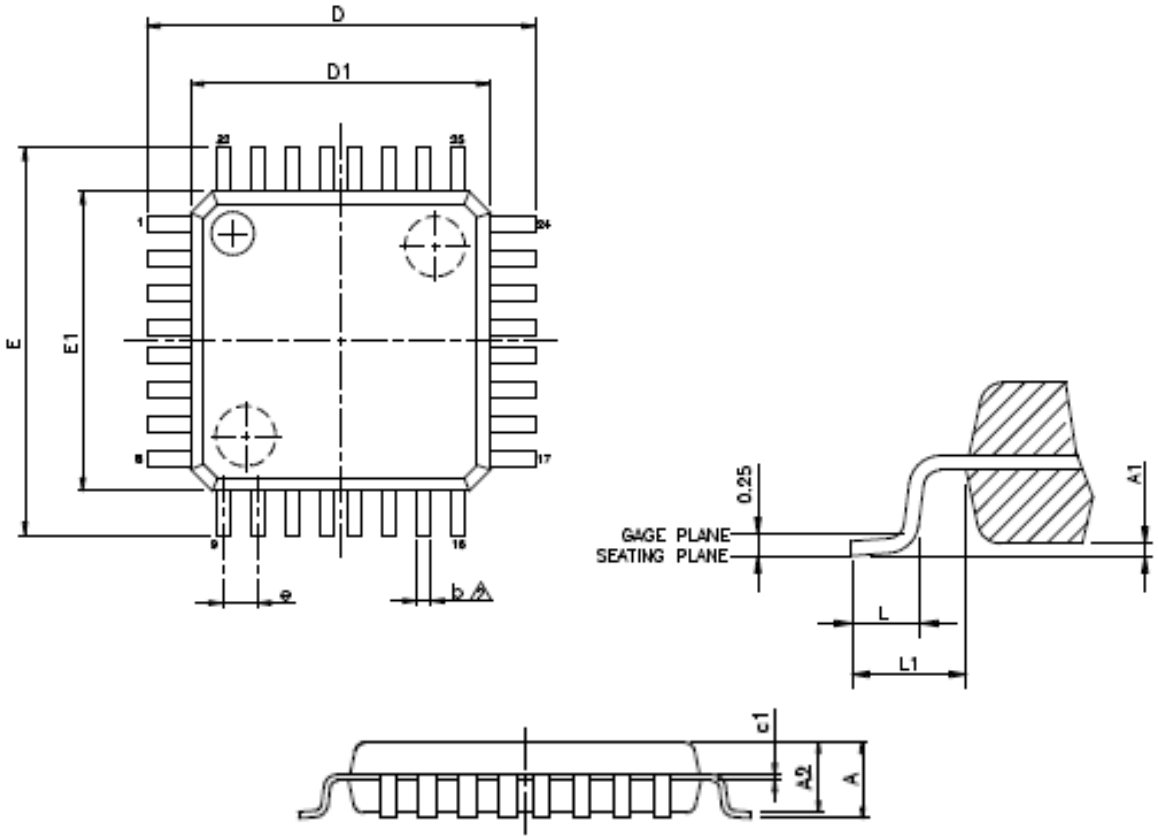




SYMBOLS	MIN.	NOM.	MAX.
A	0.70	0.75	0.80
A1	0.00	0.02	0.05
A3		0.203 REF.	
b	0.15	0.20	0.25
D		4.00 BSC	
E		4.00 BSC	
e		0.40 BSC	
L	0.25	0.30	0.35
D2	2.60	2.70	2.80
E2	2.60	2.70	2.80
K	0.20	—	—

Figure 9.4-1 QFN-33 Package Dimension

9.5 LQFP 32-pin (7.0 x 7.0 x 1.4 mm)

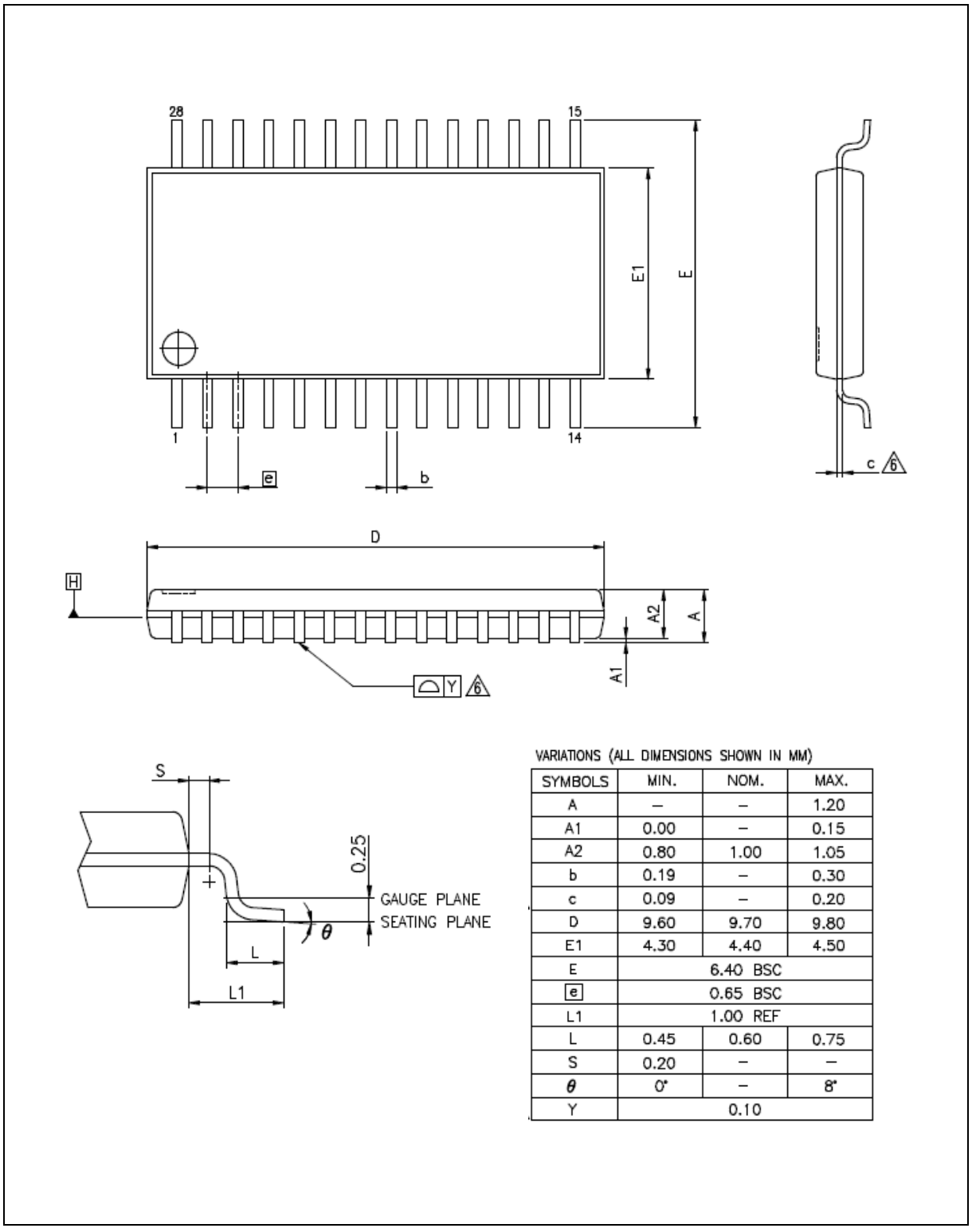


VARIATIONS (ALL DIMENSIONS SHOWN IN MM)

SYMBOLS	MIN.	MAX.
A	--	1.6
A1	0.05	0.15
A2	1.35	1.45
c1	0.09	0.16
D	9.00 BSC	
D1	7.00 BSC	
E	9.00 BSC	
E1	7.00 BSC	
e	0.8 BSC	
b	0.30	0.45
L	0.45	0.75
L1	1 REF	

Figure 9.5-1 LQFP-32 Package Dimension

9.6 TSSOP 28-pin (4.4 x 9.7 x 1.0 mm)

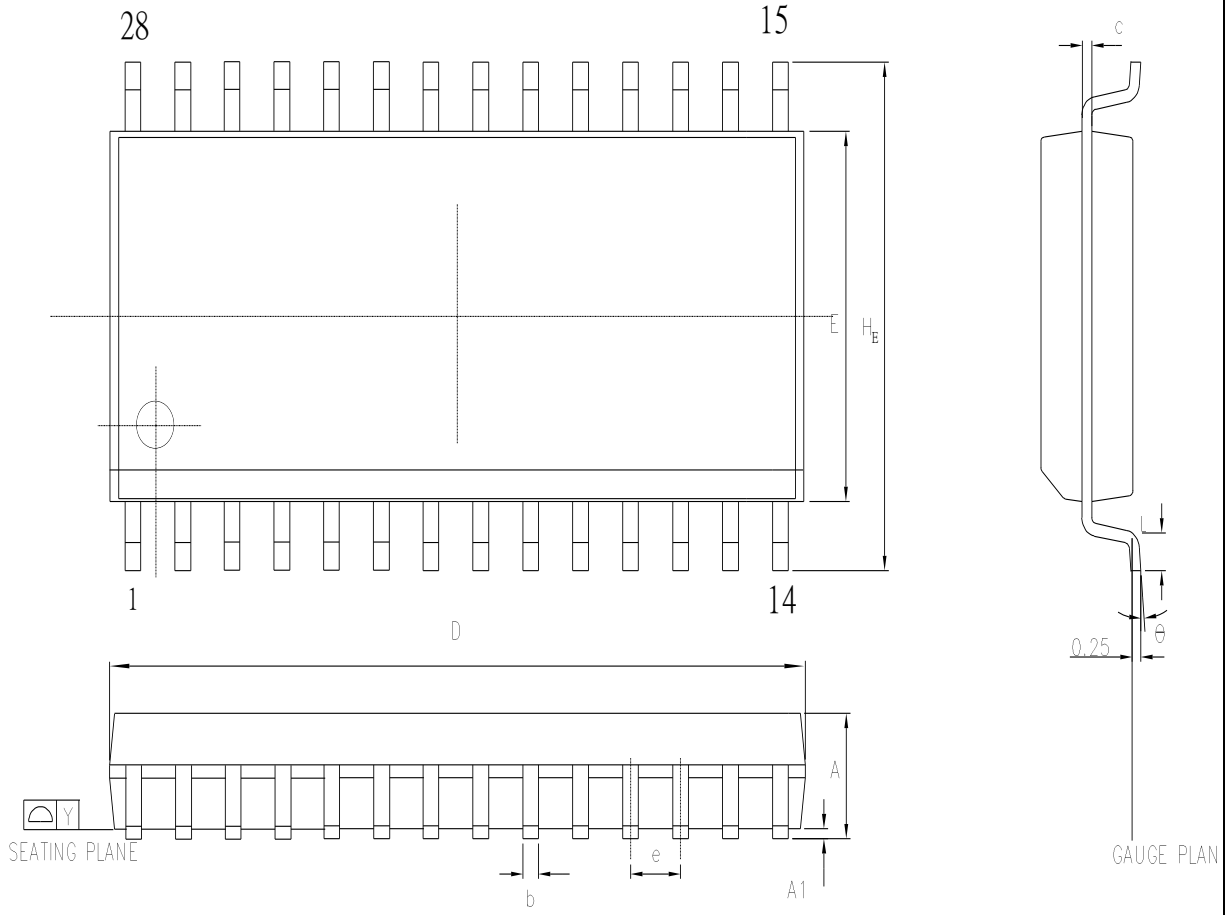


VARIATIONS (ALL DIMENSIONS SHOWN IN MM)

SYMBOLS	MIN.	NOM.	MAX.
A	–	–	1.20
A1	0.00	–	0.15
A2	0.80	1.00	1.05
b	0.19	–	0.30
c	0.09	–	0.20
D	9.60	9.70	9.80
E1	4.30	4.40	4.50
E	6.40 BSC		
e	0.65 BSC		
L1	1.00 REF		
L	0.45	0.60	0.75
S	0.20	–	–
θ	0°	–	8°
Y	0.10		

Figure 9.6-1 TSSOP-28 Package Dimension

9.7 SOP 28-pin (300mil)

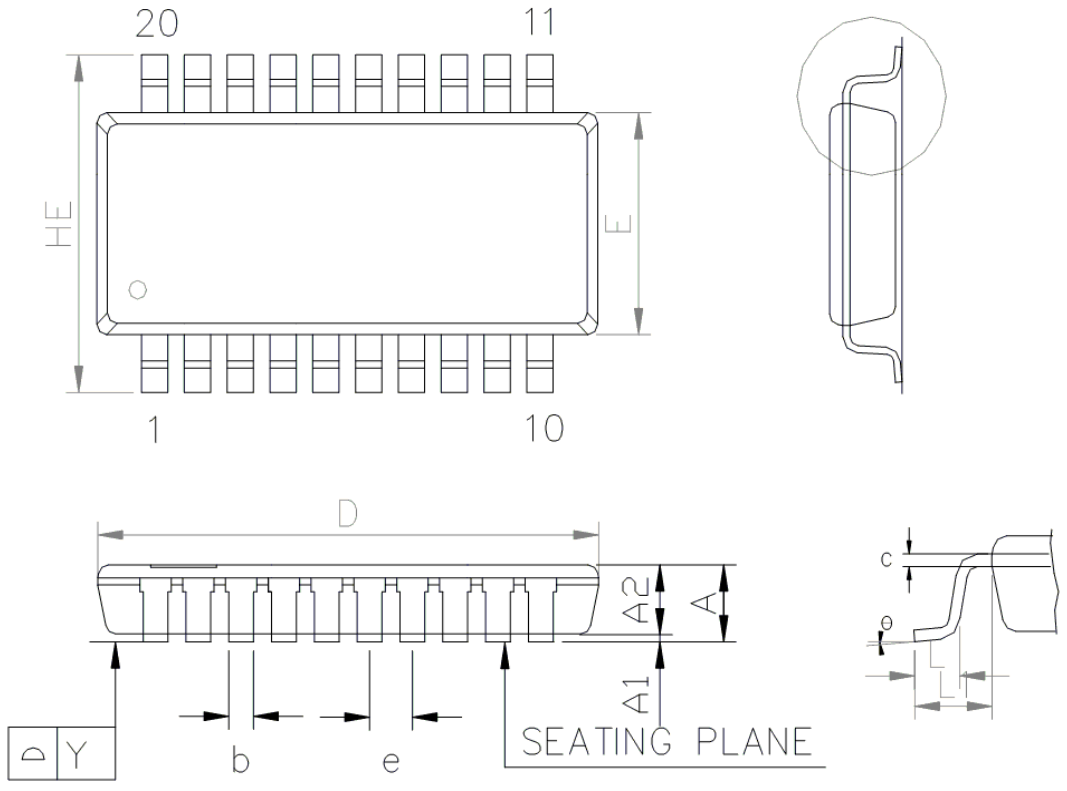


Control demensions are in milmeters .

SYMBOL	DIMENSION IN MM		DIMENSION IN INCH	
	MIN.	MAX.	MIN.	MAX.
A	2.35	2.65	0.093	0.104
A1	0.10	0.30	0.004	0.012
b	0.33	0.51	0.013	0.020
c	0.23	0.32	0.009	0.013
E	7.40	7.60	0.291	0.299
D	17.70	18.10	0.697	0.713
e	1.27 BSC		0.050 BSC	
H_E	10.00	10.65	0.394	0.419
Y	—	0.10	—	0.004
L	0.40	1.27	0.016	0.050
θ	0	8	0	8

Figure 9.7-1 SOP-28 Package Dimension

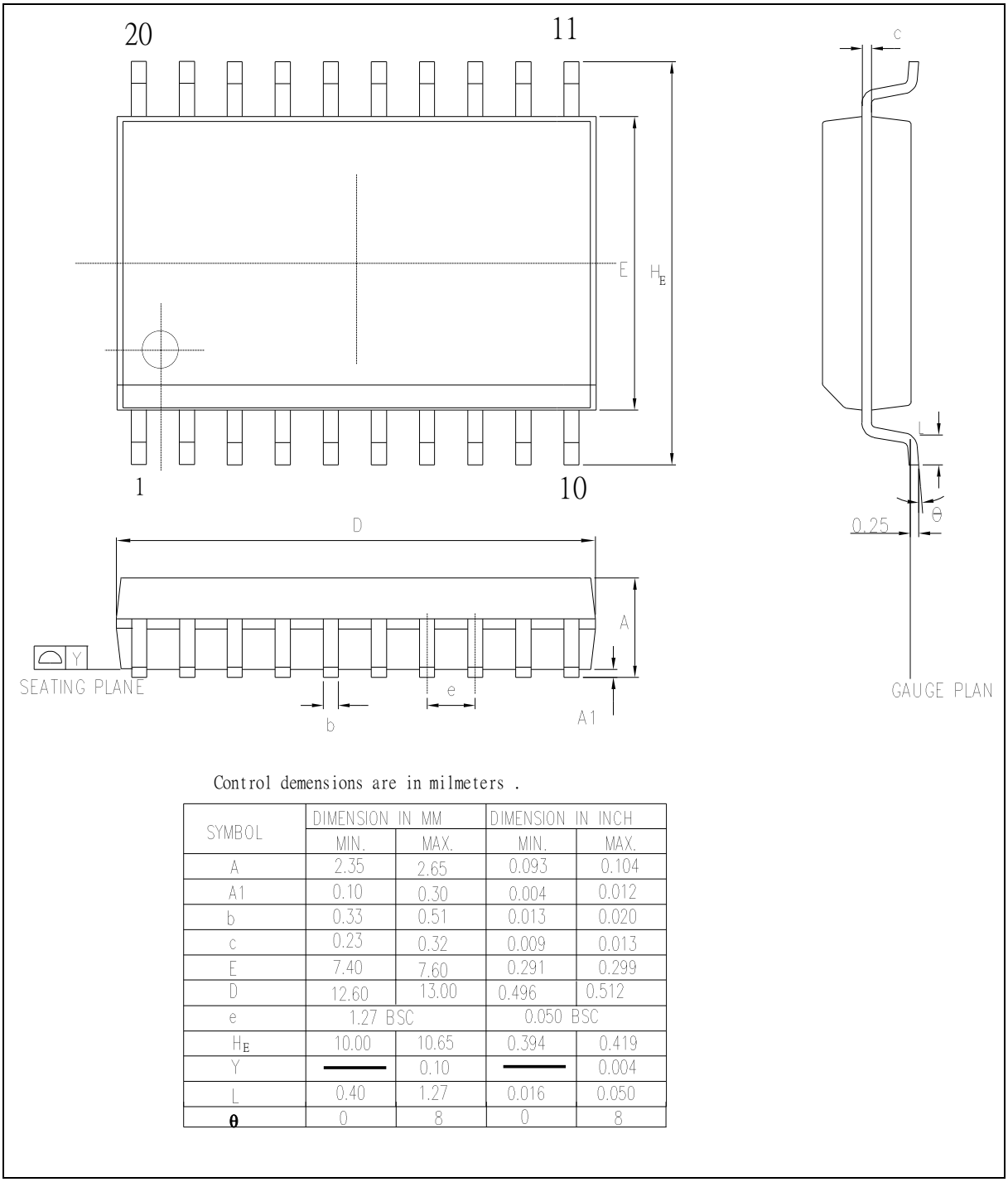
9.8 TSSOP 20-pin (4.4 x 6.5 x 0.9 mm)



SYMBOL	DIMENSION (MM)			DIMENSION (INCH)		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	-	-	1,20	-	-	0.047
A1	0,05	-	0,15	0.002	-	0.006
A2	0.80	0.90	1,05	0.031	0.035	0.041
E	4.30	4.40	4.50	0.169	0.173	0.177
HE	6.40 BSC			0.252 BSC		
D	6.40	6.50	6.60	0.252	0.256	0.260
L	0.50	0.60	0.75	0.020	0.024	0.030
L1	1.00 REF			0.039 REF		
b	0.19	-	0.30	0.007	-	0.012
e	0.65 BSC			0.026 BSC		
c	0.09	-	0.20	0.004	-	0.008
θ	0°	-	8°	0°	-	8°
Y	0.10 BASIC			0.004 BASIC		

Figure 9.8-1 TSSOP-20 Package Dimension

9.9 SOP 20-pin (300 mil)

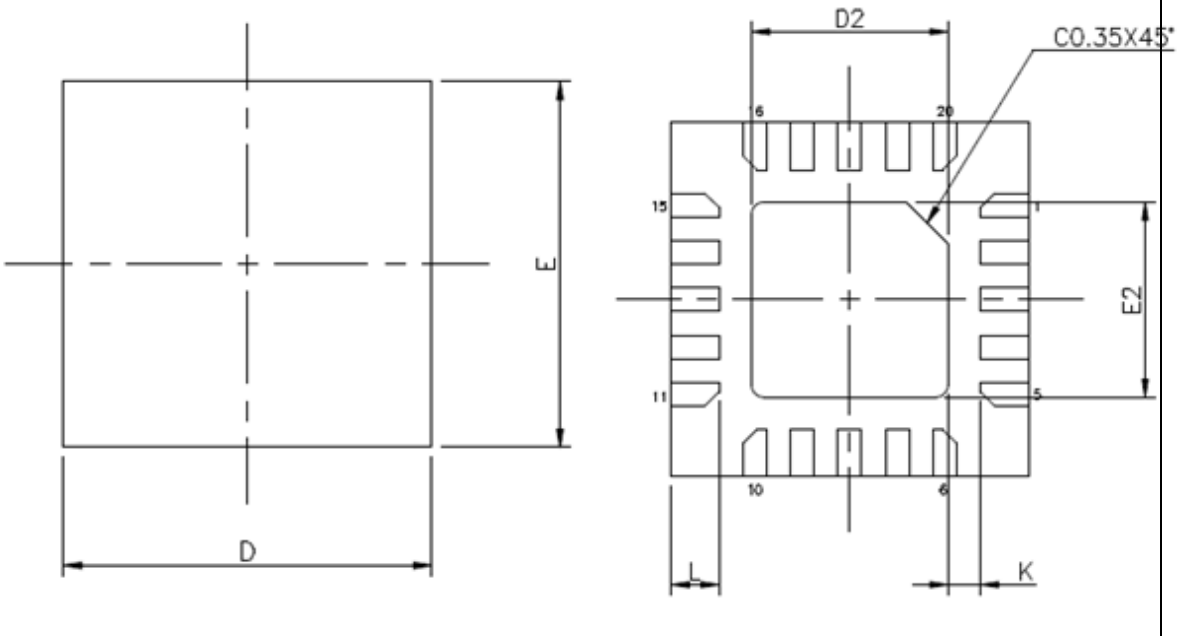


Control demensions are in milmeters .

SYMBOL	DIMENSION IN MM		DIMENSION IN INCH	
	MIN.	MAX.	MIN.	MAX.
A	2.35	2.65	0.093	0.104
A1	0.10	0.30	0.004	0.012
b	0.33	0.51	0.013	0.020
c	0.23	0.32	0.009	0.013
E	7.40	7.60	0.291	0.299
D	12.60	13.00	0.496	0.512
e	1.27 BSC		0.050 BSC	
H_E	10.00	10.65	0.394	0.419
Y	—	0.10	—	0.004
L	0.40	1.27	0.016	0.050
θ	0	8	0	8

Figure 9.9-1 SOP-20 Package Dimension

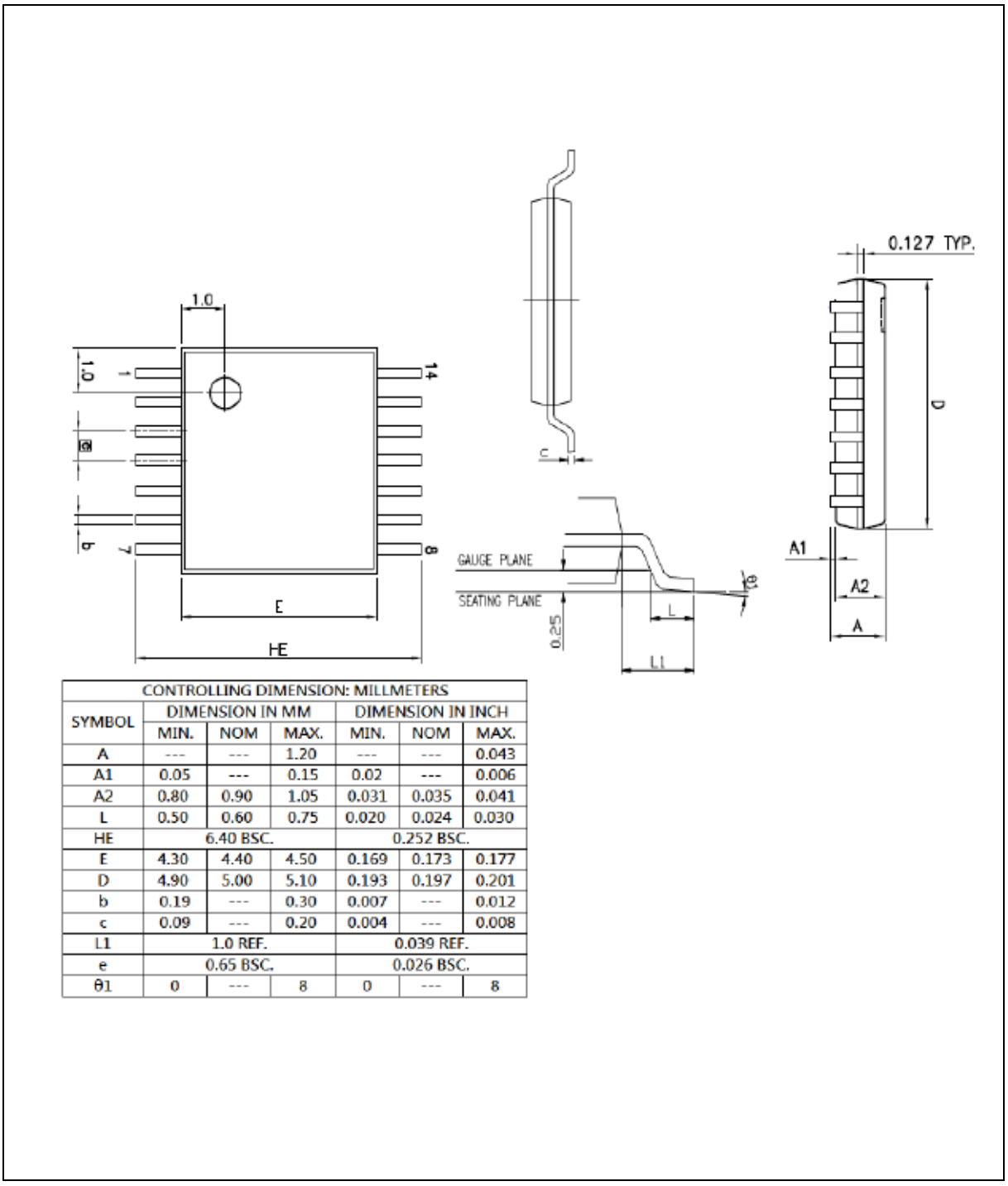
9.10 QFN 20-pin (3.0 x 3.0 x 0.8 mm)



SYMBOLS	MIN.	NOM.	MAX.
A	0.70	0.75	0.80
A1	0.00	0.02	0.05
A3		0.203 REF.	
b	0.15	0.20	0.25
D		3.00 BSC	
E		3.00 BSC	
e		0.40 BSC	
K	0.20	—	—
E2	1.60	1.65	1.70
D2	1.60	1.65	1.70
L	0.30	0.40	0.50

Figure 9.10-1 QFN-20 Package Dimension

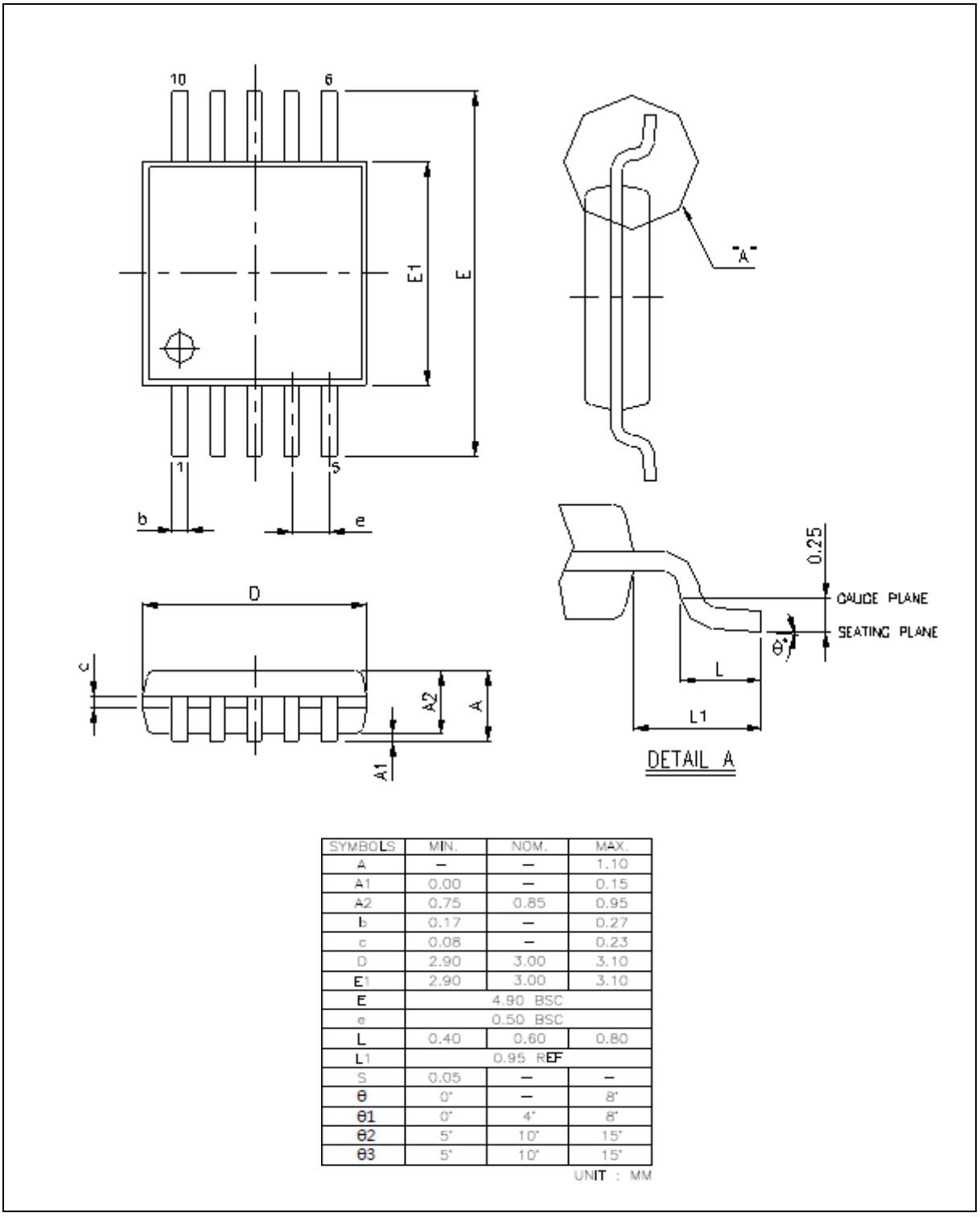
9.11 TSSOP 14-pin (4.4 x 5.0 x 0.9 mm)



CONTROLLING DIMENSION: MILLMETERS						
SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN.	NOM	MAX.	MIN.	NOM	MAX.
A	---	---	1.20	---	---	0.043
A1	0.05	---	0.15	0.02	---	0.006
A2	0.80	0.90	1.05	0.031	0.035	0.041
L	0.50	0.60	0.75	0.020	0.024	0.030
HE	6.40 BSC.			0.252 BSC.		
E	4.30	4.40	4.50	0.169	0.173	0.177
D	4.90	5.00	5.10	0.193	0.197	0.201
b	0.19	---	0.30	0.007	---	0.012
c	0.09	---	0.20	0.004	---	0.008
L1	1.0 REF.			0.039 REF.		
e	0.65 BSC.			0.026 BSC.		
θ1	0	---	8	0	---	8

Figure 9.11-1 TSSOP-14 Package Dimension

9.12 MSOP 10-pin (3.0 x 3.0 x 0.85 mm)



SYMBOLS	MIN.	NOM.	MAX.
A	–	–	1.10
A1	0.00	–	0.15
A2	0.75	0.85	0.95
b	0.17	–	0.27
c	0.08	–	0.23
D	2.90	3.00	3.10
E1	2.90	3.00	3.10
E	4.90 BSC		
e	0.50 BSC		
L	0.40	0.60	0.80
L1	0.95 REF		
S	0.05	–	–
θ	0°	–	8°
θ1	0°	4°	8°
θ2	5°	10°	15°
θ3	5°	10°	15°

UNIT : MM

Figure 9.12-1 MSOP-10 Package Dimension

10 ABBREVIATIONS

10.1 Abbreviations

Acronym	Description
ADC	Analog-to-Digital Converter
BOD	Brown-out Detection
GPIO	General-Purpose Input/Output
Fsys	Frequency of system clock
HIRC	12 MHz Internal High Speed RC Oscillator
IAP	In Application Programming
ICP	In Circuit Programming
ISP	In System Programming
LDO	Low Dropout Regulator
LIRC	10 kHz internal low speed RC oscillator (LIRC)
LVR	Low Voltage $eset
PDMA	Peripheral Direct Memory Access
POR	Power On Reset
PWM	Pulse Width Modulation
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter
UCID	Unique Customer ID
WKT	Wakeup Timer
WDT	Watchdog Timer

Table 10.1-1 List of Abbreviations

11 REVISION HISTORY

Date	Revision	Section	Description
2018.12.05	1.00		Initial release.
2019.3.18	1.01	Section 3.1	Added package type table.
		Section 4.2.2	Added Multi-function summary table
		Section 7.2	Added description that all about PWM1 register is only for 64K flash body product.
		Section 24.3	Added PDMA support in different part number.
		Section 37.6	Modified TSSOP20 package value.
2020.09.01	2.00		Added ML51 64KB/ML54/ML56 Produnct information.
2021.08.10	2.01	Section 2	Added WKT counter descitpion. For ML51 32KB/16KB series is 8bit. For ML56/ML54/ML51 64KB series is 16bit.
		Section 6.2	Added ML51 32KB /16KB flash sereis SFR full table
		Section 6.8.1	Added WKT Block diagram and register RWKH/CWKH is only for ML56/ML54/ML51 64KB flash series.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.