

1T 8051
8-bit Microcontroller

NuMicro[®] Family
MS51 Series

MS51FC0AE

MS51XC0BE

MS51EC0AE

MS51TC0AE

MS51PC0AE

Technical Reference Manual

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro[®] microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

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

www.nuvoton.com

TABLE OF CONTENTS

1 GENERAL DESCRIPTION 10

2 FEATURES 11

3 PARTS INFORMATION 15

 3.1 Package Type 15

 3.2 MS51 Series Selection Guide 15

 3.3 MS51 Series Selection Code..... 16

4 PIN CONFIGURATION 17

 4.1 MS51 32KB Series Multi Function Pin Diagram 17

 4.1.1 QFN 33-pin Package Pin Diagram17

 4.1.2 LQFP 32-pin Package Pin Diagram17

 4.1.3 TSSOP 28-pin Package Pin Diagram19

 4.1.4 TSSOP 20-pin Package Pin Diagram19

 4.1.5 QFN 20-pin Package Pin Diagram20

 4.2 MS51 32KB Series Pin Description 21

5 BLOCK DIAGRAM..... 25

 5.1 MS51 32KB Series Block Diagram 25

6 FUNCTIONAL DESCRIPTION..... 26

 6.1 Memory Organization..... 26

 6.1.1 Overview26

 6.1.2 Program Memory (APROM / LDR0M).....26

 6.1.3 Data Flash Memory.....28

 6.1.4 Security Protection Memory (SPROM)28

 6.1.5 CONFIG bytes28

 6.1.6 Data Memory (RAM)34

 6.1.7 Special Function Register (SFR).....37

 6.2 System Manager 209

 6.2.1 Clock System209

 6.2.2 Power Management.....219

 6.2.3 Power Monitoring and Reset221

 6.2.4 Interrupt System233

 6.3 Flash Memory Control 250

 6.3.1 In-Application-Programming (IAP)250

 6.3.2 In-Circuit-Programming (ICP)266

 6.3.3 On-Chip-Debugger (OCD)266

 6.3.4 96-Bit Unique Code268

 6.4 General Purpose I/O (GPIO) 269

 6.4.1 Gpio Mode269

- 6.4.2 External Interrupt Pins (INT0 and INT1)271
- 6.4.3 Pin Interrupt (PIT).....273
- 6.4.4 Control Registers of GPIO274
- 6.5 Timer..... 283
 - 6.5.1 Timer/Counter 0 And 1283
 - 6.5.2 Timer2 And Input Capture290
 - 6.5.3 TIMER 3.....307
- 6.6 Watchdog Timer (WDT)..... 311
 - 6.6.1 Overview311
 - 6.6.2 Time-Out Reset Timer.....311
 - 6.6.3 General Purpose Timer312
 - 6.6.4 Control Registers of Watchdog Timer314
- 6.7 Self Wake-Up Timer (WKT) 316
 - 6.7.1 Overview316
 - 6.7.2 Control register of WKT317
- 6.8 Pulse Width Modulated (PWM)..... 320
 - 6.8.1 Overview320
 - 6.8.2 Basic Configuration320
 - 6.8.3 Block Diagram.....322
 - 6.8.4 Functional Description323
 - 6.8.5 Control Register of PWM.....329
- 6.9 Serial Port (UART0 & UART1) 350
 - 6.9.1 Overview350
 - 6.9.2 Operating Modes350
 - 6.9.3 Baud Rate.....352
 - 6.9.4 Framing Error Detection356
 - 6.9.5 Multiprocessor Communication356
 - 6.9.6 Automatic Address Recognition356
 - 6.9.7 Control Register of Serial Port.....359
- 6.10 ISO 7816-3 Interface (SC0~2 & UART2 ~ 4)..... 375
 - 6.10.1 Overview375
 - 6.10.2 Features375
 - 6.10.3 Basic Configuration375
 - 6.10.4 Block Diagram.....376
 - 6.10.5 Operating Modes376
 - 6.10.6 ISO 7816-3 Data Transfer380
 - 6.10.7 Initial Character TS.....380
 - 6.10.8 Error Signal and Character Repetition381
 - 6.10.9 Block Guard Time and Extra Guard Time381
 - 6.10.10 Control Registers of SC Controller383
- 6.11 Inter-Integrated Circuit (I²C)..... 395

6.11.1 Overview	395
6.11.2 Functional Description	395
6.11.3 Typical Structure of I ² C Interrupt Service Routine	405
6.11.4 I ² C Time-Out	408
6.11.5 I ² C Interrupt	408
6.11.6 Control Registers of I ² C	409
6.12 Serial Peripheral Interface (SPI)	415
6.12.1 Overview	415
6.12.2 Operating Modes	417
6.12.3 Clock Formats and Data Transfer	417
6.12.4 Slave Select Pin Configuration	419
6.12.5 Mode Fault Detection	420
6.12.6 Write Collision Error	420
6.12.7 Overrun Error	420
6.12.8 SPI Interrupt	421
6.12.9 Control Register of SPI	422
6.13 12-Bit Analog-To-Digital Converter (ADC)	427
6.13.1 Overview	427
6.13.2 Functional Description	427
6.13.3 Control Registers of ADC	433
6.14 Auxiliary Features	450
6.14.1 Dual DPTRs	450
7 APPLICATION CIRCUIT	454
7.1 Power Supply Scheme	454
7.2 Peripheral Application Scheme	455
8 ELECTRICAL CHARACTERISTICS	456
9 PACKAGE DIMENSIONS	457
9.1 QFN 33-pin (4.0 x 4.0 x 0.8 mm)	457
9.2 LQFP 32-pin (7.0 x 7.0 x 1.4 mm)	458
9.3 TSSOP 28-pin (4.4 x 9.7 x 1.0 mm)	459
9.4 TSSOP 20-pin (4.4 x 6.5 x 0.9 mm)	460
9.5 QFN 20-pin (3.0 x 3.0 x 0.5mm)	461
10 ABBREVIATIONS	462
10.1 Abbreviations List	462
11 INSTRUCTION SET	463
11.1 Instruction Set And Addressing Modes	463
11.2 Read-Modify-Write Instructions	464

11.3 Instruction Set 465

12 REVISION HISTORY 468

LIST OF FIGURES

Figure 4.1-1 Pin Assignment of LQFP-32 Package 17

Figure 4.1-2 Pin Assignment of LQFP-32 Package 18

Figure 4.1-3 Pin Assignment of TSSOP28 Package 19

Figure 4.1-4 Pin Assignment of TSSOP20 Package 19

Figure 4.1-5 Pin Assignment of QFN20 Package 20

Figure 5.1-1 Functional Block Diagram..... 25

Figure 6.1-1 MS51 32KB Series Program Memory Map 27

Figure 6.1-2 SPROM Memory Mapping And SPROM Security Mode 28

Figure 6.1-3 CONFIG0 Any Reset Reloading 29

Figure 6.1-4 CONFIG2 Power-On Reset Reloading 31

Figure 6.1-5 Data Memory Map 34

Figure 6.1-6 Internal 256 Bytes RAM Addressing..... 35

Figure 6.2-1 Clock System Block Diagram 209

Figure 6.2-2 Brown-out Detection Block Diagram..... 223

Figure 6.2-3 nRESET Reset Waveform 227

Figure 6.2-4 Boot Selecting Diagram 231

Figure 6.4-1 Quasi-Bidirectional Mode Structure 270

Figure 6.4-2 Push-Pull Mode Structure 270

Figure 6.4-3 Input-Only Mode Structure 271

Figure 6.4-4 Open-Drain Mode Structure 271

Figure 6.4-5 Pin Interface Block Diagram 273

Figure 6.5-1 Timer/Counters 0 and 1 in Mode 0 283

Figure 6.5-2 Timer/Counters 0 and 1 in Mode 1 284

Figure 6.5-3 Timer/Counters 0 and 1 in Mode 2 284

Figure 6.5-4 Timer/Counter 0 in Mode 3 285

Figure 6.5-5 Timer 2 Block Diagram 290

Figure 6.5-6 Timer 2 Auto-Reload Mode and Input Capture Module Functional Block Diagram 291

Figure 6.5-7 Timer 2 Compare Mode and Input Capture Module Functional Block Diagram 292

Figure 6.5-8 Timer 3 Block Diagram 307

Figure 6.7-1 WDT as A Time-Out Reset Timer..... 312

Figure 6.7-2 Watchdog Timer Block Diagram 312

Figure 6.8-1 Self Wake-Up Timer Block Diagram 316

Figure 6.8-1 PWM0 Block Diagram 322

Figure 6.6-2 PWM1/2/3 Block Diagram 323

Figure 6.6-3 PWM0 and Fault Brake Output Control Block Diagram 324

Figure 6.6-4 PWM1/2/3 Control Block Diagram..... 325

Figure 6.6-5 PWM Edge-aligned Type Waveform 325

Figure 6.6-6 PWM Center-aligned Type Waveform..... 326

Figure 6.6-7 PWM0 Complementary Mode with Dead-time Insertion 327

Figure 6.6-8 Fault Brake Function Block Diagram 328

Figure 6.6-9 PWM Interrupt Type..... 328

Figure 6.9-1 Serial Port Mode 0 Timing Diagram 350

Figure 6.9-2 Serial Port Mode 1 Timing Diagram 351

Figure 6.9-3 Serial Port Mode 2 and 3 Timing Diagram 352

Figure 6.10-1 SC Controller Block Diagram..... 376

Figure 6.10-2 SC Interface Connection 376

Figure 6.10-3 SC Activation and Cold Reset Sequence 377

Figure 6.10-4 SC Warm Reset Sequence 378

Figure 6.10-5 SC Deactivation Sequence..... 379

Figure 6.10-6 SC Data Character 380

Figure 6.10-7 Initial Character TS..... 381

Figure 6.10-8 SC Error Signal..... 381

Figure 6.10-9 Transmit Direction Block Guard Time Operation..... 382

Figure 6.10-10 Receive Direction Block Guard Time Operation..... 382

Figure 6.10-11 Extra Guard Time Operation 382

Figure 6.11-1 I²C Bus Interconnection 395

Figure 6.11-2 I²C Bus Protocol 396

Figure 6.11-3 START, Repeated START, and STOP Conditions 396

Figure 6.11-4 Master Transmits Data to Slave by 7-bit 397

Figure 6.11-5 Master Reads Data from Slave by 7-bit 397

Figure 6.11-6 Data Format of One I²C Transfer 397

Figure 6.11-7 Acknowledge Bit 398

Figure 6.11-8 Arbitration Procedure of Two Masters 398

Figure 6.11-9 Control I²C Bus according to the Current I²C Status 399

Figure 6.11-10 Flow and Status of Master Transmitter Mode 400

Figure 6.11-11 Flow and Status of Master Receiver Mode 401

Figure 6.11-12 Flow and Status of Slave Receiver Mode 402

Figure 6.11-13 Flow and Status of General Call Mode..... 403

Figure 6.11-14 I²C Time-Out Counter 408

Figure 6.11-15 Hold Time extend enable..... 410

Figure 6.12-1 SPI Block Diagram..... 415

Figure 6.12-2 SPI Multi-Master, Multi-Slave Interconnection 416

Figure 6.12-3 SPI Single-Master, Single-Slave Interconnection..... 416

Figure 6.12-4 SPI Clock Formats..... 418

Figure 6.12-5 SPI Clock and Data Format with CPHA = 0 419

Figure 6.12-6 SPI Clock and Data Format with CPHA = 1 419

Figure 6.12-7 SPI Overrun Waveform 421

Figure 6.12-8 SPI Interrupt Request 421

Figure 6.13-1 12-bit ADC Block Diagram..... 427

Figure 6.13-2 External Triggering ADC Circuit 430

Figure 6.13-3 ADC Result Comparator 430

Figure 6.13-4 ADC Continues Mode 431

Figure 7.1-1 NuMicro® MS51 Power supply circuit 454

Figure 7.2-1 NuMicro® MS51 Peripheral interface circuit 455

Figure 9.1-1 QFN-33 Package Dimension 457

Figure 9.2-1 LQFP-32 Package Dimension 458

Figure 9.3-1 TSSOP-28 Package Dimension 459

Figure 9.4-1 TSSOP-20 Package Dimension 460

Figure 9.5-1 QFN-20 Package Dimension 461

List of Tables

Table 6.1-1 Special Function Register Memory Map 42

Table 6.1-2 SFR Definitions and Reset Values 50

Table 6.2-1 BOF Reset Value 224

Table 6.2-2 Minimum Brown-out Detect Pulse Width 227

Table 6.2-3 Interrupt Vectors 233

Table 6.2-4 Interrupt Priority Level Setting 239

Table 6.2-5 Characteristics of Each Interrupt Source 240

Table 6.3-1 IAP Modes and Command Codes 251

Table 6.4-1 Configuration for Different I/O Modes 269

Table 6.7-1 Watchdog Timer-out Interval Under Different Pre-scalars..... 311

Table 6.6-1 PWM Pin Define And Enable Control Register..... 321

Table 6.9-1 Serial Port UART0 Mode / baud rate Description..... 353

Table 6.9-2 Serial Port UART1 Mode / baud rate Description..... 354

Table 6.10-1 Smart Card or UART Pin Define And Enable Control Register 375

Table 6.12-1 Slave Select Pin Configurations 424

Table 10.1-1 List of Abbreviations..... 462

Table 11.1-1 Instruction set and addressing modes 463

Table 11.1-2 Instructions Affect Flag Settings 463

Table 11.3-1 Instruction Set 467

1 GENERAL DESCRIPTION

The MS51 is an embedded Flash type, 8-bit high performance 1T 8051-based microcontroller. The instruction set is fully compatible with the standard 80C51 and performance enhanced.

The MS51 contains 32 Kbytes of main Flash called APROM, in which the contents of User Code resides. The MS51 Flash supports In-Application-Programming (IAP) function, which enables on-chip firmware updates. IAP also makes it possible to configure any block of User Code array to be used as non-volatile data storage, which is written by IAP and read by IAP or MOVC instruction. There is an additional Flash called LDROM, in which the Boot Code normally resides for carrying out In-System-Programming (ISP). The LDROM size is configurable with a maximum of 4 Kbytes. To facilitate programming and verification, the Flash allows to be programmed and read electronically by parallel Writer or In-Circuit-Programming (ICP). Once the code is confirmed, user can lock the code for security.

The MS51 32 KB series provides rich peripherals including 256 bytes of SRAM, 2 Kbytes of auxiliary RAM (XRAM), Up to 29 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, three ISO 7816-3 interfaces, one SPI, one I²C, six basic PWM output channels, six enhanced PWM output channels, eight-channel shared pin interrupt for all I/O, and one 12-bit ADC. The peripherals are equipped with 24 sources with 4-level-priority interrupts capability.

The MS51 32 KB series is equipped with three clock sources and supports switching on-the-fly via software. The three clock sources include external clock input, 10 kHz internal oscillator, and one 16 MHz internal precise oscillator that is factory trimmed to $\pm 1\%$ at room temperature. The MS51 provides additional power monitoring detection such as power-on reset and 4-level brown-out detection, which stabilizes the power-on/off sequence for a high reliability system design.

The MS51 32 KB series microcontroller operation consumes a very low power with two economic power modes to reduce power consumption — Idle and Power-down mode, which are software selectable. Idle mode turns off the CPU clock but allows continuing peripheral operation. Power-down mode stops the whole system clock for minimum power consumption. The system clock of the MS51 can also be slowed down by software clock divider, which allows for a flexibility between execution performance and power consumption.

With high performance CPU core and rich well-designed peripherals, the MS51 benefits to meet a general purpose, home appliances, or motor control system accomplishment.

2 FEATURES

Core and System	
8051	<ul style="list-style-type: none"> 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)	<ul style="list-style-type: none"> POR with 1.15V threshold voltage level
Brown-out Detector (BOD)	<ul style="list-style-type: none"> 4-level selection, with brown-out interrupt and reset option. (4.4V / 3.7V / 2.7V / 2.2V)
Low Voltage Reset (LVR)	<ul style="list-style-type: none"> LVR with 2.0V threshold voltage level
Security	<ul style="list-style-type: none"> 96-bit Unique ID (UID) 128-bit Unique Customer ID (UCID) 128-bytes security protection memory SPROM
Memories	
Flash	<ul style="list-style-type: none"> Up to 32 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	<ul style="list-style-type: none"> 256 Bytes on-chip RAM. Additional 2 KBytes on-chip auxiliary RAM (XRAM) accessed by MOVX instruction.
Clocks	
External Clock Source	<ul style="list-style-type: none"> 4~24 MHz High-speed external crystal oscillator (HXT) for precise timing operation
Internal Clock Source	<ul style="list-style-type: none"> Default 16 MHz high-speed internal oscillator (HIRC) trimmed to $\pm 1\%$ (accuracy at 25 °C, 3.3 V), $\pm 2\%$ in -20~105°C. Selectable 24 MHz high-speed internal oscillator (HIRC).

- 10 kHz low-speed internal oscillator (LIRC) calibrating to $\pm 1\%$ 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 Timer2 with three-channel input capture module and 9 input pin can be selected.
- One 16-bit auto-reload Timer3, which can be the baud rate clock source of UART0 and UART1.

Watchdog

- 6-bit free running up counter for WDT time-out interval.
- Selectable time-out interval is 6.40 ms ~ 1.638s since WDT_CLK = 10 kHz (LIRC).
- Able to wake up from Power-down or Idle mode
- Interrupt or reset selectable on watchdog time-out

Wake-up Timer

- 16-bit free running up counter for time-out interval.
- Clock sources from LIRC
- 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 Four PWM modules, PWM0 module provides 6 output channels. PWM1 ~ PWM3 module provides 2 output channels.
- Supports independent mode for PWM output
- Supports complementary mode for 3 complementary paired PWM output channels
- Supports 16-bit resolution PWM counter
- Supports mask function and tri-state enable for each PWM pin
- PWM0 module support Dead-time insertion with 8-bit resolution
- PWM0 module Supports brake function
- PWM0 module Supports trigger ADC on the following events

Analog Interfaces

Analog-to-Digital Converter (ADC)

- Analog input voltage range: 0 ~ AV_{DD}.
- 12-bit resolution and 10-bit accuracy is guaranteed.
- Up to 15 single-end analog input channels

- 1 internal channels, they are band-gap voltage (VBG).
- Up to 500 KSPS sampling rate.
- Software Write 1 to ADCS bit.
- External pin (STADC) trigger
- PWM trigger.
- Support continues convert function auto store the A/D conversion result in XRAM.

Communication Interfaces

UART

- Supports up to 2 UARTs: UART0, UART1,
- Up to three sets ISO 7816-3 device configuration as UART
- 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

- 1 sets of I²C devices
- Master/Slave mode
- Bidirectional data transfer between masters and slaves
- 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
- Supports hold time programmable

SPI

- 1 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

- Up to three 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 impedance mode
- Schmitt trigger input / TTL mode selectable.
- Each I/O pin configured as interrupt source with edge/level

trigger setting

- Standard interrupt pins $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$.
- 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

ESD & EFT

ESD	<ul style="list-style-type: none"> • HBM pass 8 kV
EFT	<ul style="list-style-type: none"> • > ± 4.4 kV
Latch-up	<ul style="list-style-type: none"> • 150 mA pass

3 PARTS INFORMATION

3.1 Package Type

	MSOP10	TSSOP14	TSSOP20	QFN20	TSSOP28	LQFP32	QFN33
Part No.	MS51BA9AE	MS51DA9AE	MS51FB9AE MS51FC0AE	MS51XB9AE MS51XB9BE MS51XC0BE	MS51EC0AE	MS51PC0AE	MS51TC0AE

3.2 MS51 Series Selection Guide

Part Number	Flash (KB)	SRAM (KB)	LDROM (KB) ^[1]	I/O	Timer/	PWM	Connectivity				ADC(12-Bit)	Package
							ISO 7816-3 ^[2]	UART	SPI	I2C		
MS51BA9AE	8	1	4	8	4	5	-	2	-	1	5-ch	MSOP10
MS51DA9AE	8	1	4	12	4	5	-	2	1	1	7-ch	TSSOP14
MS51XB9AE	16	1	4	18	4	6	-	2	1	1	8-ch	QFN20 ^[3]
MS51XB9BE	16	1	4	18	4	6	-	2	1	1	8-ch	QFN20 ^[3]
MS51FB9AE	16	1	4	18	4	6	-	2	1	1	8-ch	TSSOP20
MS51FC0AE	32	2	4	18	4	8	2	2	1	1	10-ch	TSSOP20
MS51XC0BE	32	2	4	18	4	8	2	2	1	1	10-ch	QFN20
MS51EC0AE	32	2	4	26	4	10	3	2	1	1	15-ch	TSSOP28
MS51PC0AE	32	2	4	30	4	12	3	2	2	1	15-ch	LQFP32
MS51TC0AE	32	2	4	30	4	12	3	2	2	1	15-ch	QFN33

Note:

1. ISP ROM programmable 1K/2K/3K/4KB Flash for user program loader (LDROM) share from ARPOM.
2. ISO 7816-3 configurable as UART function, GPIO defined as UART2 ~ UART4.
3. Detailed package information please refer to Chapter 9 PACKAGE DIMENSIONS
4. This document is only for 32KB flash size part number product

3.3 MS51 Series Selection Code

MS	51	F	B	9	A	E
Core	Line	Package	Flash	SRAM	Reserve	Temperature
1T 8051 Industry	51: Base	B: MSOP10 (3x3 mm) D: TSSOP14 (4.4x5.0 mm) E: TSSOP28 (4.4x9.7 mm) F: TSSOP20 (4.4x6.5 mm) I: SOP8 (4x5 mm) O: SOP20 (300 mil) P: LQFP32 (7x7 mm) T: QFN33 (4x4 mm) U: SOP28 (300 mil) X: QFN20 (3x3mm)	A: 8 KB B: 16 KB C: 32 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

Users can find pin configuration informations 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 MS51 32KB Series Multi Function Pin Diagram

4.1.1 QFN 33-pin Package Pin Diagram

Corresponding Part Number: MS51TC0AE

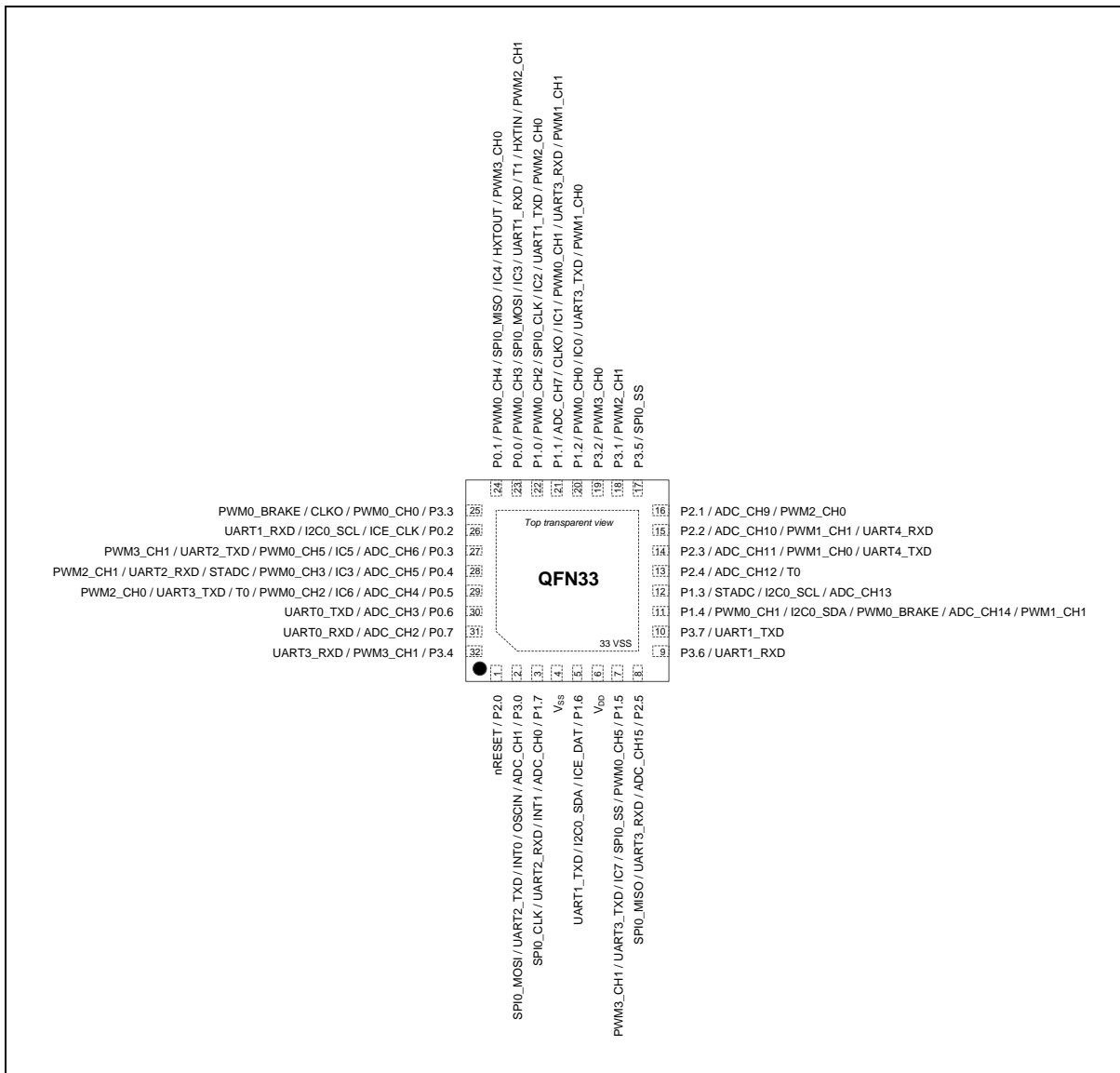


Figure 4.1-1 Pin Assignment of LQFP-32 Package

4.1.2 LQFP 32-pin Package Pin Diagram

Corresponding Part Number: MS51PC0AE

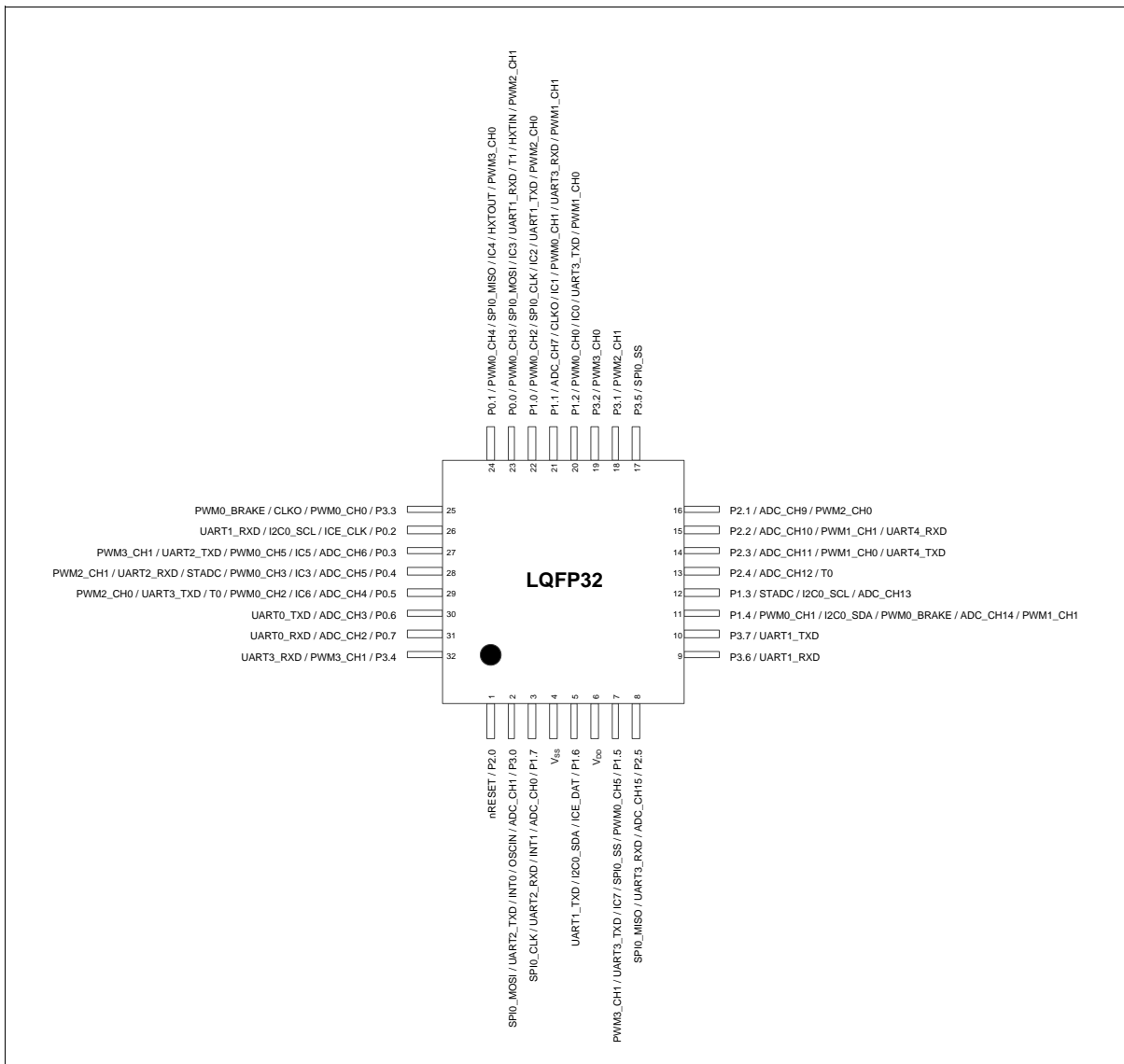


Figure 4.1-2 Pin Assignment of LQFP-32 Package

4.1.3 TSSOP 28-pin Package Pin Diagram

Corresponding Part Number: MS51EC0AE

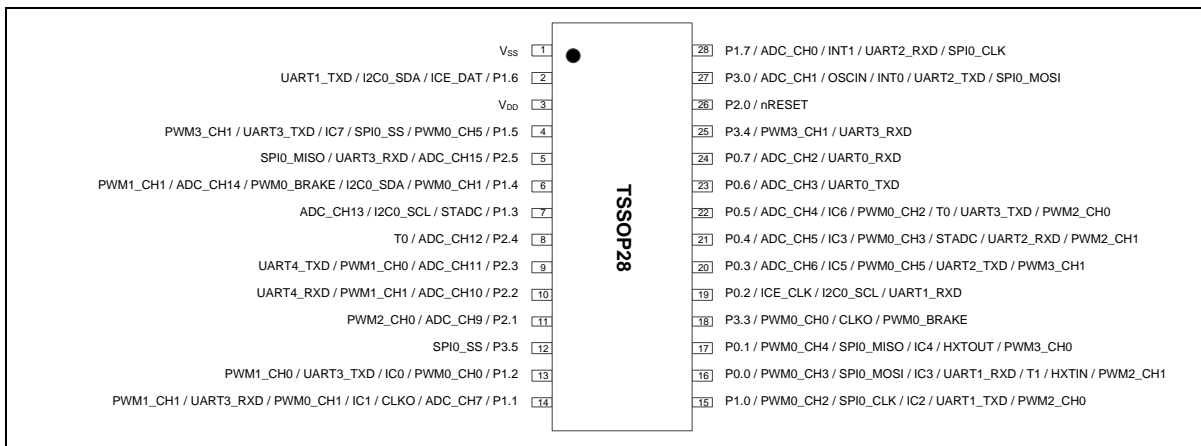


Figure 4.1-3 Pin Assignment of TSSOP28 Package

4.1.4 TSSOP 20-pin Package Pin Diagram

Corresponding Part Number: MS51FC0AE

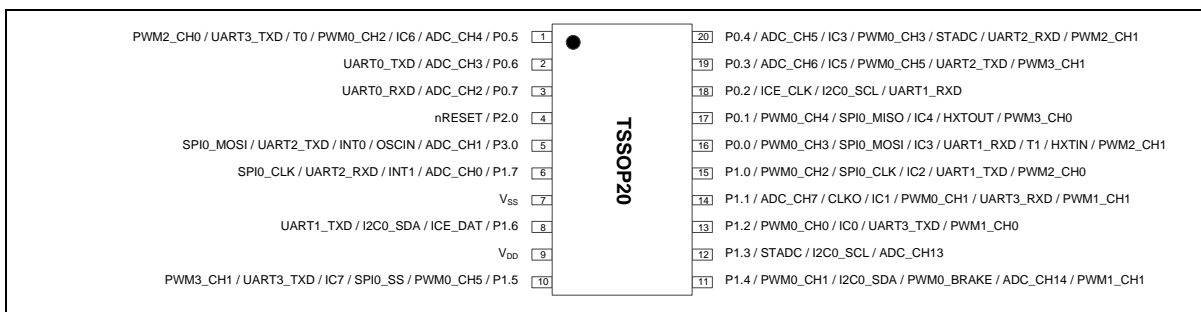


Figure 4.1-4 Pin Assignment of TSSOP20 Package

4.1.5 QFN 20-pin Package Pin Diagram

Corresponding Part Number: MS51XC0BE

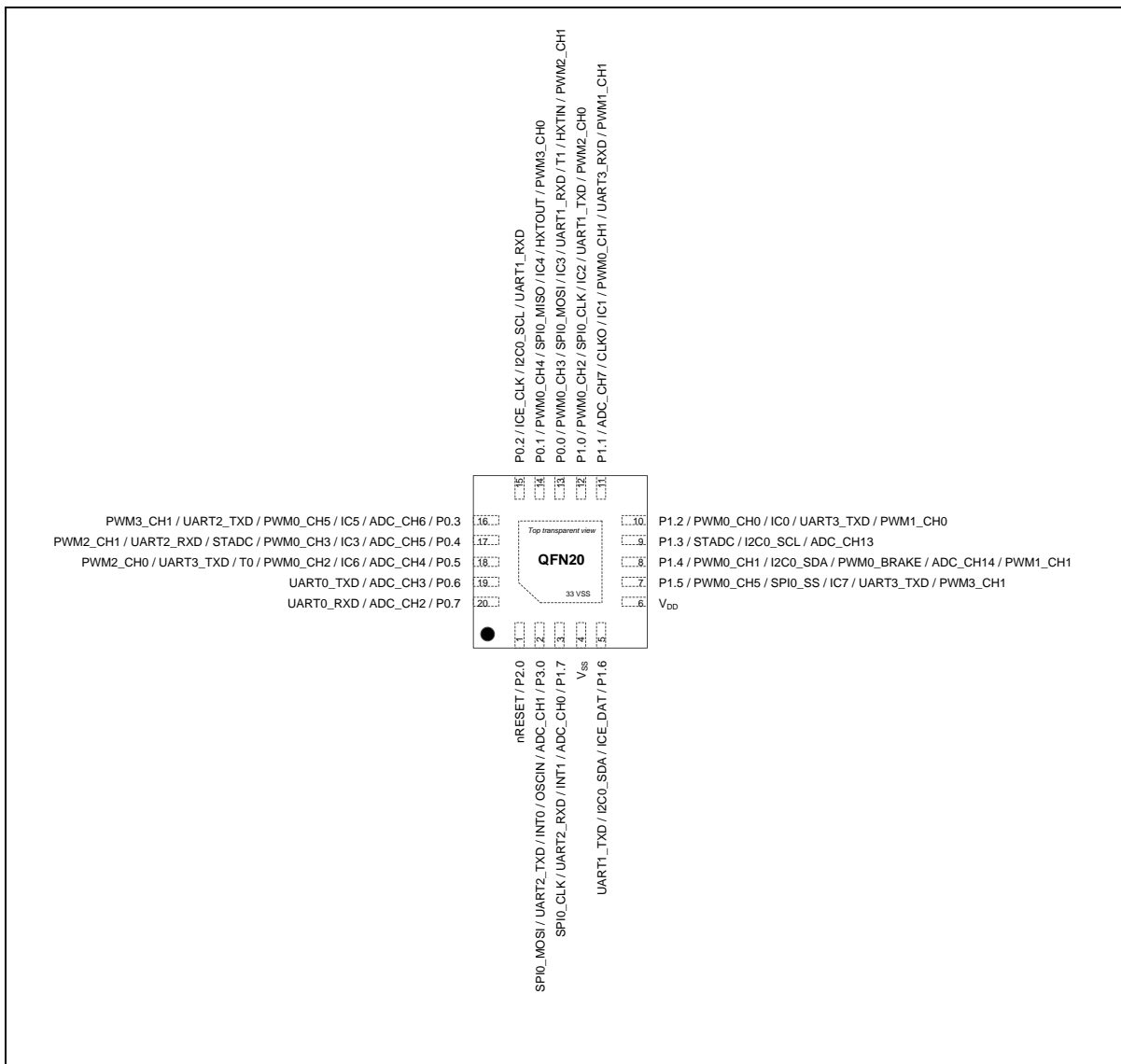


Figure 4.1-5 Pin Assignment of QFN20 Package

4.2 MS51 32KB Series Pin Description

Pin Number				Symbol	Multi-Function Description ^[1]
MS51XC0BE QFN 20	MS51FC0A E TSSOP20	MS51EC0A E TSSOP28	MS51PC0A E LQFP 32 MS51TC0A E QFN 33		
6	9	3	6	VDD	Supply voltage V _{DD} for operation.
4	7	1	4	VSS	Ground potential.
13	16	16	23	P0.0	Port 0 bit 0.
				PWM0_CH3	PWM0 output channel 3.
				PWM2_CH1	PWM2 output channel 1.
				IC3	Input capture channel 3.
				SPI0_MOSI	SPI master output/slave input.
				UART1_RXD	UART1 receive input.
				XT1_IN	External 4~24 MHz (high speed) crystal input pin.
				OSCIN	If the EXTEN[1:0] = 10b, OSCIN is the external clock input pin.
14	17	17	24	T1	External count input to Timer/Counter 1 or its toggle output.
				P0.1	Port 0 bit 1.
				PWM0_CH4	PWM0 output channel 4.
				PWM3_CH0	PWM3 output channel 0.
				IC4	Input capture channel 4.
15	18	19	26	SPI0_MISO	SPI master input/slave output.
				XT1_OUT	External 4~24 MHz (high speed) crystal output pin.
				P0.2	Port 0 bit 2.
				I2C0_SCL	I ² C clock.
16	19	20	27	UART1_RXD	UART1 receive input.
				ICE_CLK	ICE / ICP clock input.
				P0.3	Port 0 bit 3.
				ADC_CH6	ADC input channel 6.
				PWM0_CH5	PWM0 output channel5
				PWM3_CH1	PWM3 output channel1
17	20	21	28	IC5	Input capture channel 5.
				UART2_TXD	UART2 transmit data output.
				SC0_CLK	Smart Card 0 clock pin
				P0.4	Port 0 bit 4.
				ADC_CH5	ADC input channel 5.
				PWM0_CH3	PWM0 output channel 3.
				PWM2_CH1	PWM2 output channel 1.
				IC3	Input capture channel 3.
18	1	22	29	UART2_RXD	UART2 receive input.
				SC0_DAT	Smart Card 0 data pin
				STADC	External start ADC trigger
				P0.5	Port 0 bit 5.
				ADC_CH4	ADC input channel 4.
				PWM0_CH2	PWM0 output channel 2.
				PWM2_CH0	PWM2 output channel 0.
19	2	23	30	IC6	Input capture channel 6.
				UART3_TXD	UART3 transmit data output.
				SC1_CLK	Smart card clock pin.
				T0	External count input to Timer/Counter 0 or its toggle output.
				P0.6	Port 0 bit 6.

Pin Number				Symbol	Multi-Function Description ^[1]
MS51XC0BE QFN 20	MS51FC0A E TSSOP20	MS51EC0A E TSSOP28	MS51PC0A E LQFP 32 MS51TC0A E QFN 33		
				ADC_CH3	ADC input channel 3.
				UART0_TXD	UART0 transmit data output.
20	3	24	31	P0.7	Port 0 bit 7.
				ADC_CH2	ADC input channel 2.
				UART0_RXD	UART0 transmit data output.
				P1.0	Port 1 bit 0.
12	15	15	22	PWM0_CH2	PWM0 output channel 2.
				PWM2_CH0	PWM2 output channel 0.
				IC2	Input capture channel 2.
				SPI0_CLK	SPI0 clock.
				UART1_TXD	UART1 receive input.
				P1.1	Port 1 bit 1
11	14	14	21	ADC_CH7	ADC input channel 7.
				PWM0_CH1	PWM0 output channel 1.
				PWM1_CH1	PWM1 output channel 1.
				IC1	Input capture channel 1.
				UART3_RXD	UART3 receive input.
				SC1_DAT	Smart Card 1 data pin.
				CLKO	System clock output.
10	13	13	20	P1.2	Port 1 bit 2.
				PWM0_CH0	PWM0 output channel 0.
				PWM1_CH0	PWM1 output channel 0.
				IC0	Input capture channel 0.
				UART3_TXD	UART3 transmit data output.
9	12	7	12	SC1_CLK	Smart Card 1 clock pin.
				P1.3	Port 1 bit 3.
				ADC_CH13	ADC input channel 13.
8	11	6	11	I2C0_SCL	I ² C0 clock.
				STADC	External start ADC trigger
				P1.4	Port 1 bit 4.
				ADC_CH14	ADC input channel 14.
7	10	4	7	PWM0_CH1	PWM0 output channel 1.
				PWM1_CH1	PWM1 output channel 1.
				I2C0_SDA	I ² C0 data.
				PWM0_BRAKE	PWM0 Fault Brake input.
				P1.5	Port 1 bit 5.
5	8	2	5	PWM0_CH5	PWM0 output channel 5.
				PWM3_CH1	PWM3 output channel 1.
				IC7	Input capture channel 7.
				SPI0_SS	SPI0 slave select input.
				UART3_TXD	UART3 transmit data output.
				SC1_CLK	Smart card 2 clock pin
3	6	28	3	P1.6	Port 1 bit 6.
				I2C0_SDA	I ² C0 data.
				UART1_TXD	UART1 transmit data output.
				ICE_DAT	ICE data input or output.
				P1.7	Port 1 bit 7.
				ADC_CH0	ADC input channel 0.
				PWM3_CH0	PWM3 output channel 0.
				SPI0_CLK	SPI0 clock.

Pin Number				Symbol	Multi-Function Description ^[1]
MS51XC0BE QFN 20	MS51FC0A E TSSOP20	MS51EC0A E TSSOP28	MS51PC0A E LQFP 32 MS51TC0A E QFN 33		
				UART2_RXD	UART2 receive input.
				SC0_DAT	Smart Card 0 data pin
				INT1	External interrupt 1 input.
1	4	26	1	P2.0	Port 2 bit 0 input pin available when RPD (CONFIG0.2) is programmed as 0.
				nRESET	It is a Schmitt trigger input pin for hardware device reset. A low on this pin resets the device. nRESET pin has an internal pull-up resistor allowing power-on reset by simply connecting an external capacitor to VSS.
-	-	11	16	P2.1	Port 2 bit 1.
				ADC_CH9	ADC input channel 9.
				PWM2_CH0	PWM2 output channel 0.
-	-	10	15	P2.2	Port 2 bit 2.
				ADC_CH10	ADC input channel 10.
				PWM1_CH1	PWM1 output channel 1.
				UART4_RXD	UART4 receive input.
				SC2_DAT6	Smart card 2 data pin
-	-	9	14	P2.3	Port 2 bit 3.
				ADC_CH11	ADC input channel 11.
				PWM1_CH0	PWM1 output channel 0.
				UART4_TXD	UART4 transmit data output.
				SC2_CLK6	Smart card 2 clock pin
-	-	8	13	P2.4	Port 2 bit 4.
				ADC_CH12	ADC input channel 12.
				T0	External count input to Timer/Counter 0 or its toggle output.
-	-	5	8	P2.5	Port 2 bit 5.
				ADC_CH15	ADC input channel 15.
				SPI0_MISO	SPI master input/slave output.
				UART3_RXD	UART3 receive input.
				SC1_DAT6	Smart card 1 data pin
2	5	27	2	P3.0	Port 3 bit 0.
				ADC_CH1	ADC input channel 1.
				PWM2_CH1	PWM2 output channel 1.
				SPI0_MOSI	SPI master output/slave input.
				UART2_TXD	UART2 transmit data output.
				SC0_CLK6	Smart card 0 clock pin
				INT0	External interrupt 0 input.
				OSCIN	If the EXTEN[1:0] = 11b, OSCIN is the external clock input pin.
-	-	-	18	P3.1	Port 3 bit 1.
				PWM2_CH1	PWM2 output channel 1.
-	-	-	19	P3.2	Port 3 bit 2.
				PWM3_CH0	PWM3 output channel 0.
-	-	18	25	P3.3	Port 3 bit 3.
				PWM0_CH0	PWM3 output channel 0.
				CLK_OUT	System clock output.
				PWM0_BRAKE	PWM0 Fault Brake input.
-	-	25	32	P3.4	Port 3 bit 4.
				PWM3_CH1	PWM3 output channel 0.
				UART3_RXD	UART3 receive input.

Pin Number				Symbol	Multi-Function Description ^[1]
MS51XC0BE QFN 20	MS51FC0A E TSSOP20	MS51EC0A E TSSOP28	MS51PC0A E LQFP 32 MS51TC0A E QFN 33		
				SC1_DAT	Smart card 0 data pin
-	-	12	17	P3.5	Port 3 bit 5.
				SPI0_SS	SPI0 slave select input.
-	-	-	9	P3.6	Port 3 bit 6.
				UART1_TXD	UART1 transmit data output.
-	-	-	10	P3.7	Port 3 bit 7.
				UART1_RXD	UART1 receive input.

Note:

4.2.1 1. All I/O pins can be configured as a interrupt pin. This feature is not listed in multi-function description. See Chapter 6.4.2 External Interrupt Pins (INT0 and INT1)

The external interrupt $\overline{INT0}$ and $\overline{INT1}$ 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 $\overline{INT0}$ or $\overline{INT1}$ 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 $\overline{INT0}$ and $\overline{INT1}$ 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 $\overline{INT0}$ and $\overline{INT1}$ can wake up the device from the Power-down 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
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 $\overline{INT1}$ input signal's logic level. Software cannot control it.
2	IT1	External interrupt 1 type select This bit selects by which type that $\overline{INT1}$ is triggered. 0 = $\overline{INT1}$ is low level triggered. 1 = $\overline{INT1}$ 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.

Pin Number				Symbol	Multi-Function Description ^[1]
MS51XC0BE QFN 20	MS51FC0A E TSSOP20	MS51EC0A E TSSOP28	MS51PC0A E LQFP 32 MS51TC0A E QFN 33		
		If IT0 = 0 (low level trigger), this flag follows the inverse of the $\overline{INT0}$ input signal's logic level. Software cannot control it.			
0	IT0	External interrupt 0 type select This bit selects by which type that $\overline{INT0}$ is triggered. 0 = $\overline{INT0}$ is low level triggered. 1 = $\overline{INT0}$ is falling edge triggered.			
Pin Interrupt (PIT) 2. UART0_TXD and UART0_RXD pins are software exchangeable by UART0PX (AUXR1.2). 3. [I2C] alternate function remapping option. I2C pins is software switched by I2CPX (I2CON.0). 4. [STADC] alternate function remapping option. STADC pin is software switched by STADCPX(ADCCON1.6). 5. PIOx register decides which pins are PWM or GPIO. 6. UART2_TXD and UART2_RXD pin is defined by AUXR2 register. UART3_TXD, UART3_RXD, UART4_TXD and UART4_RXD pin defined by AUXR3 register.					

5 BLOCK DIAGRAM

5.1 MS51 32KB Series Block Diagram

Figure 5.1-1 Functional Block Diagram shows the MS51 functional block diagram and gives the outline of the device. User can find all the peripheral functions of the device in the diagram.

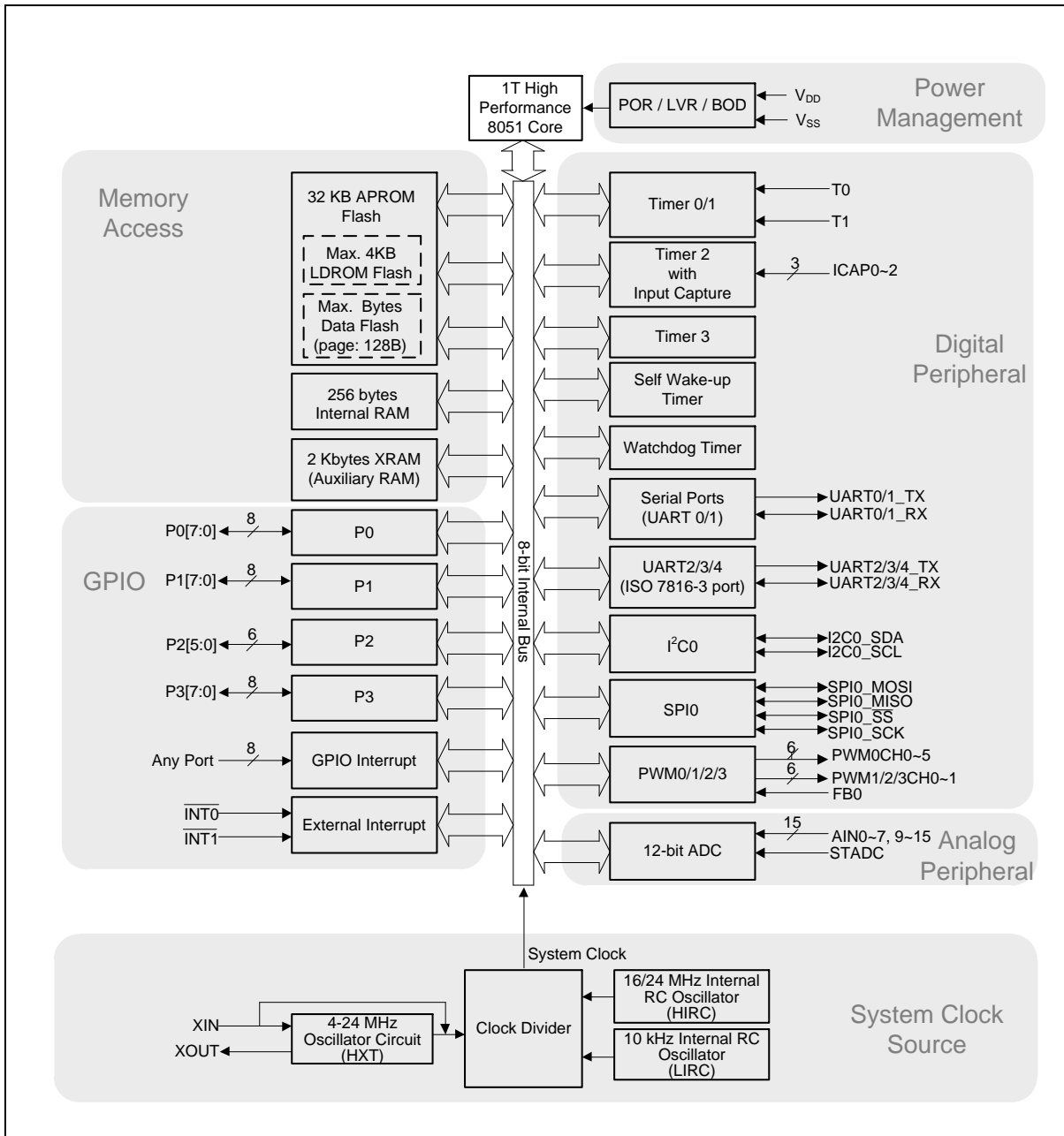


Figure 5.1-1 Functional Block Diagram

6 FUNCTIONAL DESCRIPTION

6.1 Memory Organization

6.1.1 Overview

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 MS51, there are 256 bytes of internal scratch-pad RAM. For many applications those need more internal RAM, the MS51 provides another on-chip 2 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.

6.1.2 Program Memory (APROM / LDROM)

The Program Memory stores the program codes to execute as shown in Figure 6.1-1 MS51 32KB 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 MS51 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 MS51 can be up to 32 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 6.3.1.5 "In-System-Programming \(ISP\)" on page 262](#). 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 This field selects the size of LDROM. 111 = No LDROM. APROM is 32 Kbytes. 110 = LDROM is 1 Kbytes. APROM is 31 Kbytes. 101 = LDROM is 2 Kbytes. APROM is 30 Kbytes. 100 = LDROM is 3 Kbytes. APROM is 29 Kbytes. 0xx = LDROM is 4 Kbytes. APROM is 28 Kbytes.

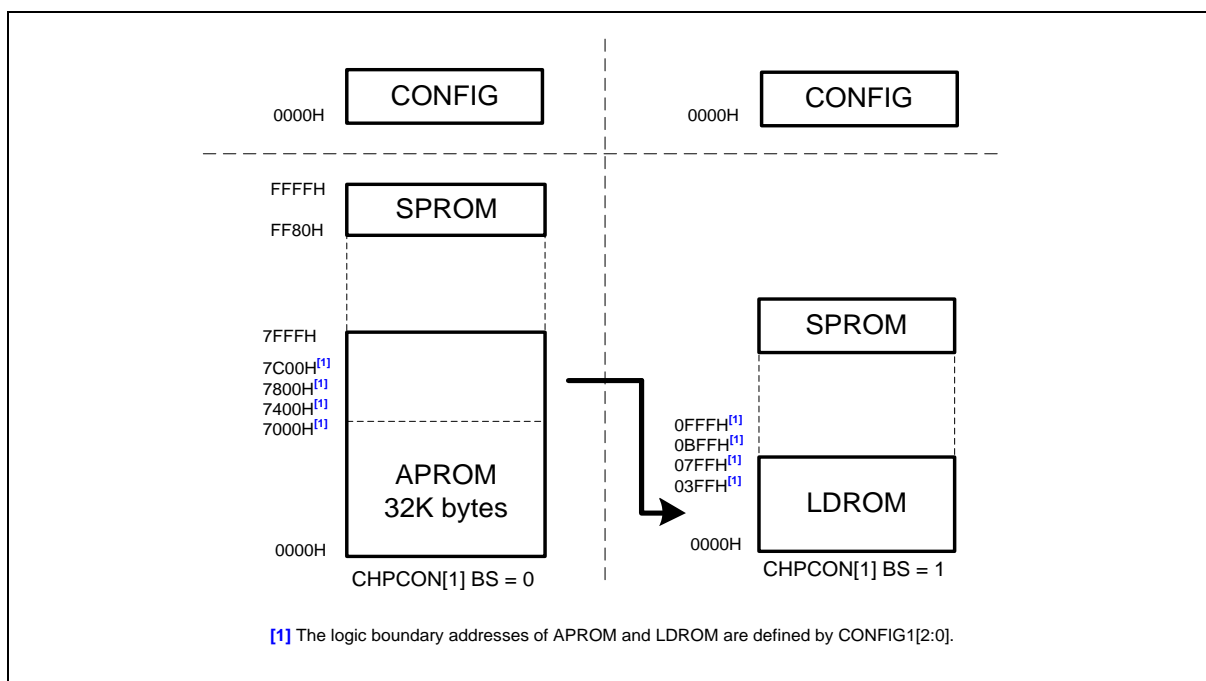


Figure 6.1-1 MS51 32KB Series Program Memory Map

6.1.3 Data Flash Memory

MS51 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 6.3.1 In-Application-Programming (IAP). All of embedded Flash memory is 128 bytes per page erased.

6.1.4 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.

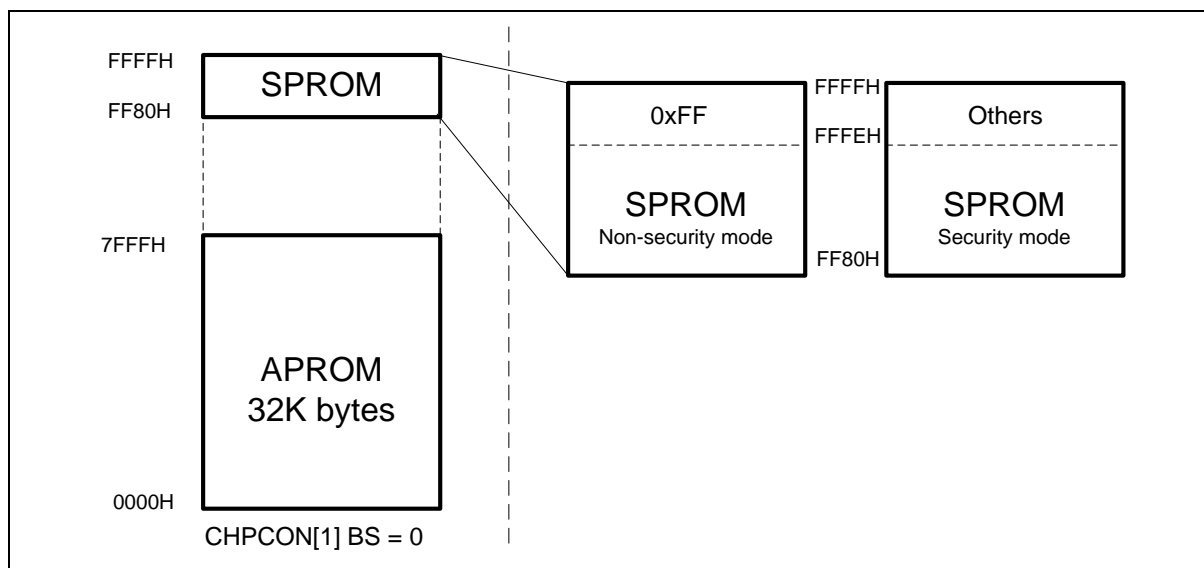


Figure 6.1-2 SPROM Memory Mapping And SPROM Security Mode

(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.

6.1.5 CONFIG bytes

The MS51 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	-	OCDPWM	OCDEN	-	RPD	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 LDR0M after resets except software reset.
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. Note that this bit is valid only when the corresponding PIO bit of PWM channel is set as 1.
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.
3	-	Reserved
2	RPD	Reset pin disable 1 = The reset function of P2.0/nRST pin Enabled. P2.0/nRST functions as the external reset pin. 0 = The reset function of P2.0/nRST pin Disabled. P2.0/nRST functions as an input-only pin P2.0.
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.

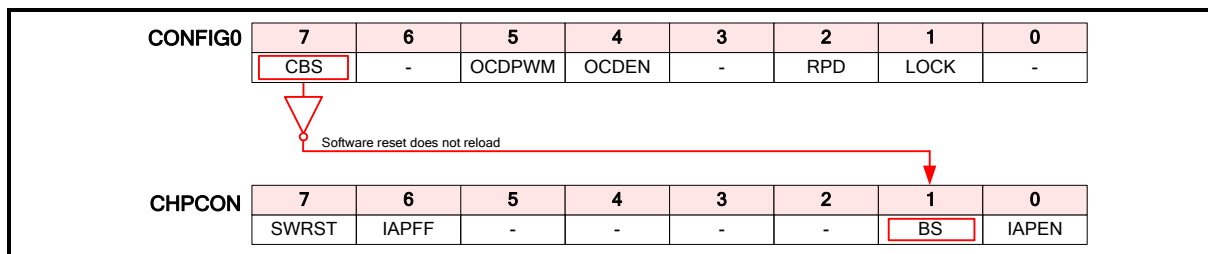


Figure 6.1-3 CONFIG0 Any Reset Reloading

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 Part number is MS51: 111 = No LDROM. APROM is 32 Kbytes. 110 = LDROM is 1 Kbytes. APROM is 31 Kbytes. 101 = LDROM is 2 Kbytes. APROM is 30 Kbytes. 100 = LDROM is 3 Kbytes. APROM is 29 Kbytes. 0xx = LDROM is 4 Kbytes. APROM is 28 Kbytes.

CONFIG2

7	6	5	4	3	2	1	0
CBODEN	-	CBOV[1:0]		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 on. 0 = Brown-out detection circuit off.
6	-	Reserved
5:4	CBOV[1:0]	CONFIG brown-out voltage select 11 = V_{BOD} is 2.2V. 10 = V_{BOD} is 2.7V. 01 = V_{BOD} is 3.7V. 00 = 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.

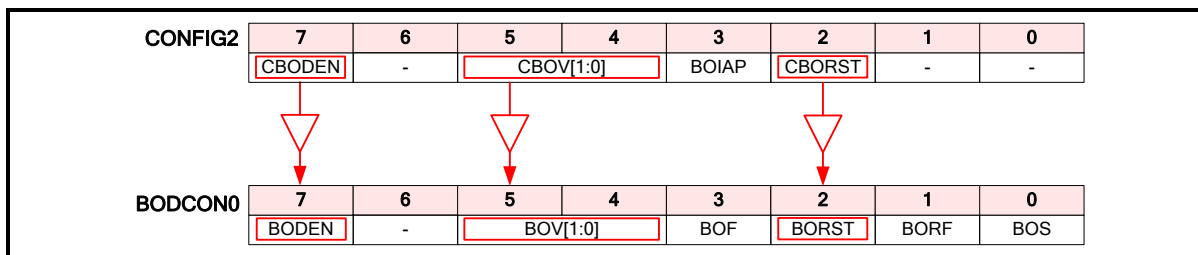


Figure 6.1-4 CONFIG2 Power-On Reset Reloading

CONFIG3

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

Factory default value: 1111 1111b

Bit	Name	Description
7:6	CKFS[1:0]	Clock filter time select Enable clock filter. It increases noise immunity and EMC capacity. 11 = 15 ns. 10 = 15 ns. 01 = 15 ns. 00 = Clock filter Disabled.
5:0	-	Reserved

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]	<p>WDT enable</p> <p>This field configures the WDT behavior after MCU execution.</p> <p>1111 = WDT is Disabled. WDT can be used as a general purpose timer via software control.</p> <p>0101 = WDT is Enabled as a time-out reset timer and it stops running during Idle or Power-down mode.</p> <p>Others = WDT is Enabled as a time-out reset timer and it keeps running during Idle or Power-down mode.</p>
3:0	-	Reserved

6.1.6 Data Memory (RAM)

6.1.6.1 internal Data Memory

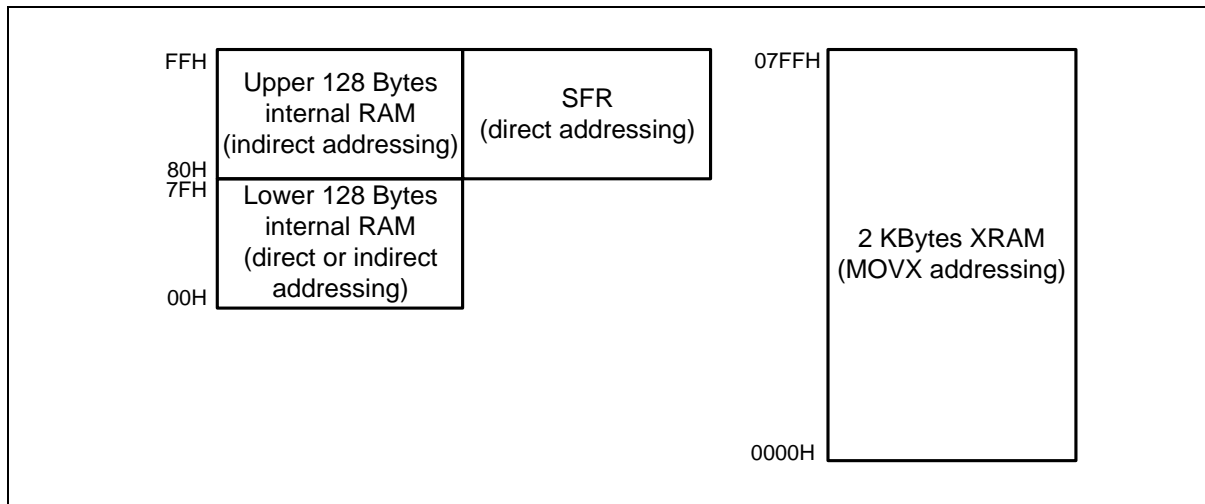


Figure 6.1-5 Data Memory Map

Figure 6.1-5 Data Memory Map shows the internal Data Memory spaces available on MS51. 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.

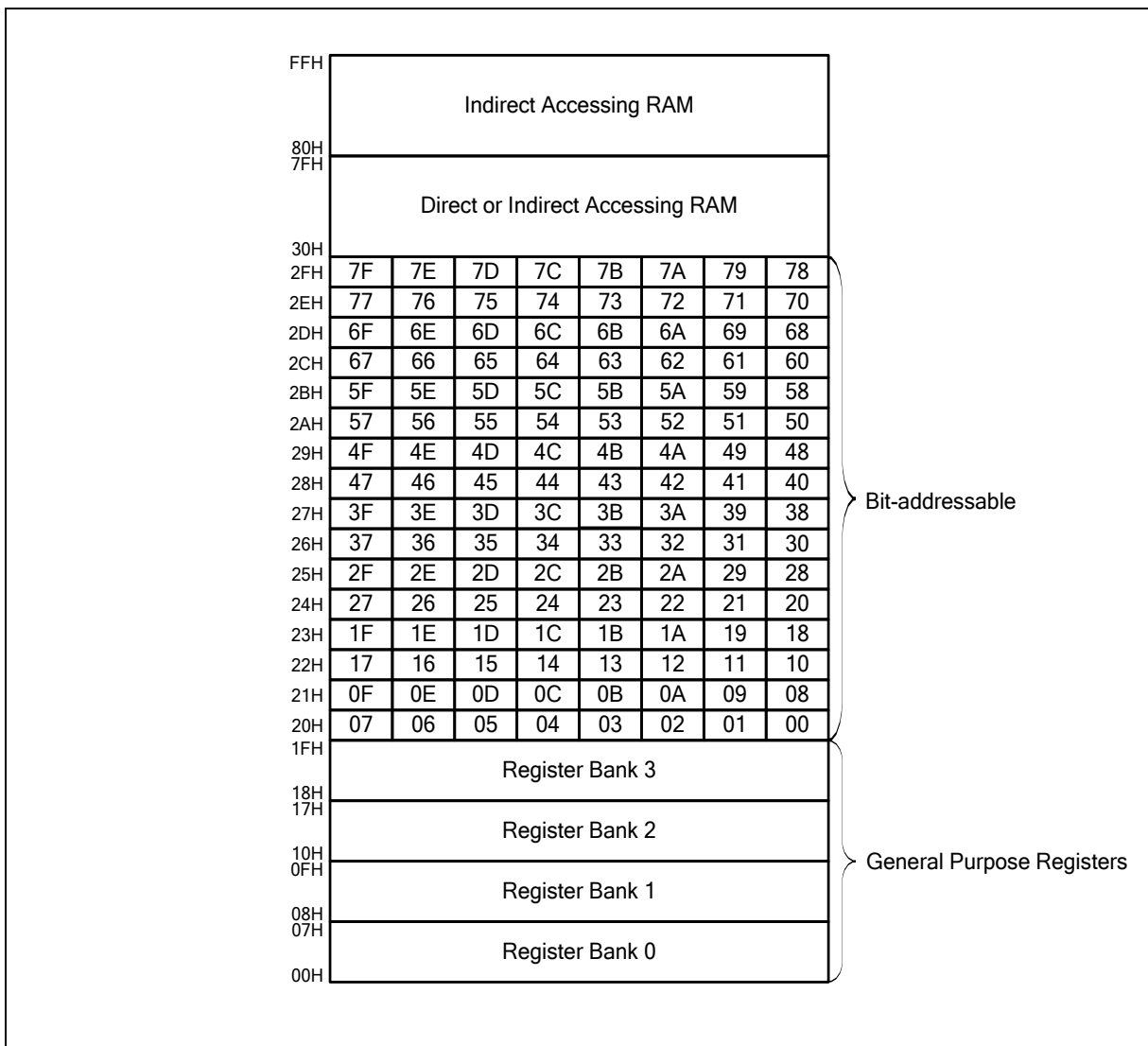


Figure 6.1-6 Internal 256 Bytes RAM Addressing

6.1.6.2 On-Chip XRAM

The MS51 provides additional on-chip 2 Kbytes auxiliary RAM called XRAM to enlarge the RAM space. It occupies the address space from 00H through FFH. The 2 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
    
```

6.1.7 Special Function Register (SFR)

The MS51 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 MS51 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 1/2 may need to be accessed. The register SFRS is used to switch SFR addressing page.

Note: In particular, since the interrupt is unpredictable, it is recommended to add following action in each interrupt subroutine. Push SFRS in the stack at the beginning of the interrupt vector program and pop SFRS when exiting the interrupt.

SFRS – SFR Page Selection

Register	SFR Address	Reset Value
SFRS	91H, All page	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	-	SFRPAGE1	SFRPAGE0
-	-	-	-	-	-	R/W	R/W

Bit	Name	Description
1:0	SFRPAGE[1:0]	SFR page select 0 = Instructions access SFR page 0. 1 = Instructions access SFR page 1. 2 = Instructions access SFR page 2.

Switch SFR page demo code:

```
MOV    SFRS, #01H
MOV    SFRS, #00H
```


6.1.7.2 Timed Access Protection (TA)

The MS51 has several features such as WDT and Brown-out detection that are crucial to proper operation of the system. If leaving these control registers unprotected, errant code may write undetermined value into them and results in incorrect operation and loss of control. To prevent this risk, the MS51 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 page	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
temporarily
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.

EIPH – Extensive Interrupt Priority High^[4]

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

7	6	5	4	3	2	1	0
PT2H	PSPIH	PFBH	PWDTH	PPWMH	PCAPH	PPIH	PI2CH
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	PSPIH	SPI interrupt priority high bit
5	PFBH	Fault Brake interrupt priority high bit
4	PWDTH	WDT interrupt priority high bit
3	PPWMH	PWM interrupt priority high bit
2	PCAPH	Input capture interrupt priority high bit
1	PPIH	Pin interrupt priority high bit
0	PI2CH	I ² C interrupt priority high bit

Note: EIPH is used in combination with the EIP to determine the priority of each interrupt source. See [Table 6.2-4 Interrupt Priority Level Setting](#) for correct interrupt priority configuration.

6.1.7.3 SFR Memory Map

Page	Addr	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
0	F8	SCON_1	PWM0DTEN	PWM0DTCNT	PWM0MEN	PWM0MD	PORDIS	EIP1	EIPH1	
1			-	-	-	-	-	-	LVRDIS	
2			PIPS0	PIPS1	PIPS2	PIPS3	PIPS4	PIPS5	PIPS6	
0	F0	B	CAPCON3	CAPCON4	SPCR	SPSR	SPDR	AINDIDS0	EIPH	
1			-	-	-	-	-	-	-	
2			SC0CR0	SC0CR1	SC1CR0	SC1CR1	SC2CR0	SC2CR1	PIPS7	
0	E8	ADCCON0	PICON	PINEN	PIPEN	PIF	C2L	C2H	EIP	
1			-	-	-	-	-	-	-	
2			SC2DR	SC2EGT	SC2ETURD0	SC2ETURD1	SC2IE	SC2IS	SC2TSR	
0	E0	ACC	ADCCON1	ADCCON2	ADCDLY	C0L	C0H	C1L	C1H	
1			-	-	-	-	-	-	-	
2			SC1DR	SC1EGT	SC1ETURD0	SC1ETURD1	SC1IE	SC1IS	SC1TSR	
0	D8	PWM0CON0	PWM0PL	PWM0C0L	PWM0C1L	PWM0C2L	PWM0C3L	PIOCON0	PWM0CON1	
1			-	-	-	-	-	-	-	
2			SC0DR	SC0EGT	SC0ETURD0	SC0ETURD1	SC0IE	SC0IS	SC0TSR	
0	D0	PSW	PWM0PH	PWM0C0H	PWM0C1H	PWM0C2H	PWM0C3H	PNP	PWM0FBD	
1			-	-	-	-	-	-	-	
2			PWM3PL	PWM3_CH0L	PWM3_CH1L	PWM3CON0	PWM3CON1	PWM3INTC	XTLCON	
0	C8	T2CON	T2MOD	RCMP2L	RCMP2H	TL2	TH2	ADCMPL	ADCMPH	
1			-	-	-	PWM0C4L	PWM0_CH5L	-	-	
2			PWM3PH	PWM3_CH0H	PWM3_CH1H	PWM3MD	PWM3MEN	EIP2	EIPH2	
0	C0	I2CON	I2ADDR	ADCRL	ADCRH	T3CON	RL3	RH3	TA	
1			-	-	-	PWM0C4H	PWM0_CH5H	PIOCON1		
2			PWM2PL	PWM2_CH0L	PWM2_CH1L	PWM2CON0	PWM2CON1	PWM2INTC		
0	B8	IP	SADEN	SADEN_1	SADDR_1	I2DAT	I2STAT	I2CLK	I2TOC	
1			-	-	-	-	-	-	-	
2			PWM2PH	PWM2_CH0H	PWM2_CH1H	PWM2MD	PWM2MEN	-	-	
0	B0	P3	P0M1	P0M2	P1M1	P1M2	TOE	-	IPH	
1			P0S	P0SR	P1S	P1SR	-	-	-	PWM0INTC
2			PWM1PL	PWM1_CH0L	PWM1_CH1L	PWM1CON0	PWM1CON1	PWM1INTC	PIOCON2	
0	A8	IE	SADDR	WDCON	BODCON1	P3M1	P3M2	IAPFD	IAPCN	
1			-	-	-	P3S	P3SR	-	PWM0INTC	
2			PWM1PH	PWM1_CH0H	PWM1_CH1H	PWM1MD	PWM1MEN	-	-	
0	A0	P2	-	AUXR1	BODCON0	IAPTRG	IAPUEN	IAPAL	IAPAH	
1			-	-	-	-	-	-	-	
2			AUXR2	AUXR3	AUXR4	AUXR5	AUXR6	AUXR7	AUXR8	
0	98	SCON	SBUF	SBUF_1	EIE	EIE1	RSR	-	CHPCON	
1			-	-	-	-	-	-	-	
2			AINDIDS1	P0DW	P1DW	P2DW	P3DW	-	-	
0	90	P1	SFRS	CAPCON0	CAPCON1	CAPCON2	CKDIV	CKSWT	CKEN	
1			-	-	-	-	-	-	-	
2			P0UP	P1UP	P2UP	P3UP	-	-	RWKH	
0	88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	WKCON	
1			-	-	-	-	-	-	-	
2			P2M1	P2M2	P2SR	P2S	ADCSN	ADCCN	ADCSR	
0	80	P0	SP	DPL	DPH	RCTRIM0	RCTRIM1	RWK	PCON	
1			-	-	-	-	-	-		
2			-	-	-	ADCBAL	ADCBAL	ADCCON3		

Note:

- 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.
- In bytes means bit addressable SFR all page accessible.
- In means this SFR all page accessible.

Table 6.1-1 Special Function Register Memory Map

6.1.7.4 SFR Description and Reset Values

Register	Definition	Address	Page	MSB 7	6	5	4	3	2	1	LSB ^[1] 0	Reset Value ^[2]	TA
EIPH1	Extensive interrupt priority high 1	FFH	0	-	-	PPWM3H	PPWM2H	PPWM1H	PWKTH	PT3H	PSH_1	0000 0000b	
LVRDIS	LVR Disable	FFH	1	LVRDIS[7:0]								0000 0000b	Y
PIPS6	Pin Interrupt Control 6	FFH	2	-	BSEL2	BSEL1	BSEL0	-	BSEL2	BSEL1	BSEL0	0000 0000b	
-	-	FEH	0	-	-	-	-	-	-	-	-	-	
EIP1	Extensive interrupt priority 1	FEH	1	-	-	PPWM3	PPWM2	PPWM1	PWKT	PT3	PS_1	0000 0000b	
PIPS5	Pin Interrupt Control 5	FEH	2	-	PSEL2	PSEL1	PSEL0	-	BSEL2	BSEL1	BSEL0	0000 0000b	
PORDIS	POR disable	FDH	0	PORDIS[7:0]								0000 0000b	Y
-	-	FDH	1	-								-	-
PIPS4	Pin Interrupt Control 4	FDH	2	-	PSEL2	PSEL1	PSEL0	-	BSEL2	BSEL1	BSEL0	0000 0000b	
PWM0MD	PWM0 mask data	FCH	0	-	-	PMD5	PMD4	PMD3	PMD2	PMD1	PMD0	0000 0000b	
-	-	FCH	1	-	-	-	-	-	-	-	-	-	
PIPS3	Pin Interrupt Control 3	FCH	2	-	PSEL2	PSEL1	PSEL0	-	BSEL2	BSEL1	BSEL0	0000 0000b	
PWM0MEN	PWM0 mask enable	FBH	0	-	-	PMEN5	PMEN4	PMEN3	PMEN2	PMEN1	PMEN0	0000 0000b	
-	-	FBH	1	-	-	-	-	-	-	-	-	-	
PIPS2	Pin Interrupt Control 2	FBH	2	-	PSEL2	PSEL1	PSEL0	-	BSEL2	BSEL1	BSEL0	0000 0000b	
PWM0DTCNT	PWM0 dead-time counter	FAH	0	PWM0DTCNT[7:0]								0000 0000b	Y
-	-	FAH	1	-	-	-	-	-	-	-	-	-	
PIPS1	Pin Interrupt Control 1	FAH	2	-	PSEL2	PSEL1	PSEL0	-	BSEL2	BSEL1	BSEL0	0000 0000b	
PWM0DTCNEN	PWM0 dead-time enable	F9H	0	-	-	-	PWM0DTCNT.8	-	PDT45EN	PDT23EN	PDT01EN	0000 0000b	Y
-	-	F9H	1	-	-	-	-	-	-	-	-	-	
PIPS0	Pin Interrupt Control 0	F9H	2	-	PSEL2	PSEL1	PSEL0	-	BSEL2	BSEL1	BSEL0	0000 0000b	
SCON_1	Serial port 1 control	F8H	A	SM0_1 FE_1	SM1_1	SM2_1	REN_1	TB8_1	RB8_1	TL_1	RI_1	0000 0000b	
EIPH	Extensive interrupt priority high	F7H	0	PT2H	PSPIH	PFBH	PWDTH	PPWMH	PCAPH	PIIH	PI2CH	0000 0000b	
-	-	F7H	1	-	-	-	-	-	-	-	-	-	
PIPS7	Pin Interrupt Control 7	F7H	2	-	PSEL2	PSEL1	PSEL0	-	BSEL2	BSEL1	BSEL0	0000 0000b	
AINDIDS0	ADC channel digital input disable	F6H	0	P11DIDS	P03DIDS	P04DIDS	P05DIDS	P06DIDS	P07DIDS	P30DIDS	P17DIDS	0000 0000b	
-	-	F6H	1	-	-	-	-	-	-	-	-	-	
SC2CR1	SC2 Control Register 1	F6H	2	OPE	PBOFF	WLS1	WLS0	TXDMAEN	RXDMAEN	CLKKEEP	UARTEN		
SPDR	SPI data	F5H	0	SPDR[7:0]								0000 0000b	
-	-	F5H	1	-								-	-
SC2CR0	SC2 Control Register 0	F5H	2	NSB	T	RXBGTEN	CONSEL	AUTOCEN	TXOFF	RXOFF	SCEN	0000 0000b	
SPSR	SPI status	F4H	0	SPIF	WCOL	SPIOVF	MODF	DISMODF	TXBUF	-	-	0000 0000b	
-	-	F4H	1	-	-	-	-	-	-	-	-	-	
SC1CR1	SC1 Control Register 1	F4H	2	OPE	PBOFF	WLS1	WLS0	TXDMAEN	RXDMAEN	CLKKEEP	UARTEN	0000 0000b	
SPCR	SPI control	F3H	0	SSOE	SPIEN	LSBFE	MSTR	CPOL	CPHA	SPR1	SPR0	0000 0000b	
SPCR2	SPI control 2	F3H	1	-	-	-	-	-	-	SPIS1	SPIS0	0000 0000b	
SC1CR0	SC1 Control Register 0	F3H	2	NSB	T	RXBGTEN	CONSEL	AUTOCEN	TXOFF	RXOFF	SCEN	0000 0000b	
CAPCON4	Input capture control 4	F2H	0	-	-	-	-	CAP23	CAP22	CAP21	CAP20	0000 0000b	
-	-	F2H	1	-	-	-	-	-	-	-	-	-	
SC0CR1	SC0 Control Register 1	F2H	2	OPE	PBOFF	WLS1	WLS0	TXDMAEN	RXDMAEN	CLKKEEP	UARTEN	0000 0000b	
CAPCON3	Input capture control 3	F1H	0	CAP13	CAP12	CAP11	CAP10	CAP03	CAP02	CAP01	CAP00	0000 0000b	
-	-	F1H	1	-	-	-	-	-	-	-	-	-	
SC0CR0	SC0 Control Register 0	F1H	2	NSB	T	RXBGTEN	CONSEL	AUTOCEN	TXOFF	RXOFF	SCEN	0000 0000b	
B	B register	F0H	A	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	0000 0000b	
EIP	Extensive interrupt priority	EFH	0	PT2	PSPI	PFB	PWDT	PPWM	PCAP	PPI	PI2C	0000 0000b	
-	-	EFH	1	-								-	-

Register	Definition	Address	Page	MSB 7	6	5	4	3	2	1	LSB ^[1] 0	Reset Value ^[2]	TA
SC2TSR	SC2 Transfer Status Register	EFH	2	ACT	BEF	FEF	PEF	TXEMPTY	TXOV	RXEMPTY	RXOV	0000 1010b	
C2H	Input capture 2 high byte	EEH	0	C2H[7:0]								0000 0000b	
-	-	EEH	1	-	-	-	-	-	-	-	-	-	
SC2IS	SC2 Interrupt Status Register	EEH	2	-	-	-	ACERRIF	BGTIF	TERRIF	TBEIF	RDAIF	0000 0010b	
C2L	Input capture 2 low byte	EDH	0	C2L[7:0]								0000 0000b	
-	-	EDH	1	-	-	-	-	-	-	-	-	-	
SC2IE	SC2 Interrupt Enable Control Register	EDH	2	-	-	-	ACERRIEN	BGTIEN	TERRIEN	TBEIEN	RDAIEN	0000 0000b	
PIF	Pin interrupt flag	ECH	0	PIF7	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	0000 0000b	
-	-	ECH	1	-	-	-	-	-	-	-	-	-	
SC2ETUR D1	SC2 ETU Rate Divider Register	ECH	2	-	SCDIV2	SCDIV1	SCDIV0	ETURDIV1 1	ETURDIV1 0	ETURDIV9	ETURDIV8	0000 0000b	
PIPEN	Pin interrupt high level/rising edge enable	EBH	0	PIPEN7	PIPEN6	PIPEN5	PIPEN4	PIPEN3	PIPEN2	PIPEN1	PIPEN0	0000 0000b	
-	-	EBH	1	-	-	-	-	-	-	-	-	-	
SC2ETUR D0	SC2 ETU Rate Divider Register	EBH	2	ETURDIV[7:0]								0111 0011b	
PINEN	Pin interrupt low level/falling edge enable	EAH	0	PINEN7	PINEN6	PINEN5	PINEN4	PINEN3	PINEN2	PINEN1	PINEN0	0000 0000b	
-	-	EAH	1	-	-	-	-	-	-	-	-	-	
SC2EGT	SC2 Extra Guard Time Register	EAH	2	SC2EGT[7:0]								0000 0000b	
PICON	Pin interrupt control	E9H	0	PIT7	PIT6	PIT5	PIT4	PIT3	PIT2	PIT1	PIT0	0000 0000b	
-	-	E9H	1	-	-	-	-	-	-	-	-	-	
SC2DR	SC2 Data Register	E9H	2	SC2DR[7:0]								0000 0000b	
ADCCON0	ADC control 0	E8H	A	ADCF	ADCS	ETGSEL1	ETGSEL0	ADCHS3	ADCHS2	ADCHS1	ADCHS0	0000 0000b	
C1H	Input capture 1 high byte	E7H	0	C1H[7:0]								0000 0000b	
-	-	E7H	1	-	-	-	-	-	-	-	-	-	
SC1TSR	SC1 Transfer Status Register	E7H	2	ACT	BEF	FEF	PEF	TXEMPTY	TXOV	RXEMPTY	RXOV	0000 1010b	
C1L	Input capture 1 low byte	E6H	0	C1L[7:0]								0000 0000b	
-	-	E6H	1	-	-	-	-	-	-	-	-	-	
SC1IS	SC1 Interrupt Status Register	E6H	2	-	-	-	ACERRIF	BGTIF	TERRIF	TBEIF	RDAIF	0000 0010b	
C0H	Input capture 0 high byte	E5H	0	C0H[7:0]								0000 0000b	
-	-	E5H	1	-	-	-	-	-	-	-	-	-	
SC1IE	SC1 Interrupt Enable Control Register	E5H	2	-	-	-	ACERRIEN	BGTIEN	TERRIEN	TBEIEN	RDAIEN	0000 0000b	
C0L	Input capture 0 low byte	E4H	0	C0L[7:0]								0000 0000b	
-	-	E4H	1	-	-	-	-	-	-	-	-	-	
SC1ETUR D1	SC1 ETU Rate Divider Register	E4H	2	-	SCDIV2	SCDIV1	SCDIV0	ETURDIV1 1	ETURDIV1 0	ETURDIV9	ETURDIV8	0011 0001b	
ADCDLY	ADC trigger delay	E3H	0	ADCDLY[7:0]								0000 0000b	
-	-	E3H	1	-	-	-	-	-	-	-	-	-	
SC1ETUR D0	SC1 ETU Rate Divider Register	E3H	2	ETURDIV[7:0]								0111 0011b	
ADCCON2	ADC control 2	E2H	0	ADFBEN	ADCMPOP	ADCM PEN	ADCMPO	ADCAQT2	ADCAQT1	ADCAQT0	ADCDLY.8	0000 0000b	
-	-	E2H	1	-	-	-	-	-	-	-	-	-	
SC1EGT	SC1 Extra Guard Time Register	E2H	2	SC1EGT[7:0]								0000 0000b	
ADCCON1	ADC control 1	E1H	0	OCEN	STADCPX	ADCDIV1	ADCDIV0	ETGTYP1	ETGTYP0	ADCEX	ADCEN	0000 0000b	
-	-	E1H	1	-	-	-	-	-	-	-	-	-	
SC1DR	SC1 Data Register	E1H	2	SC1DR[7:0]								0000 0000b	
ACC	Accumulator	E0H	A	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	0000 0000b	
PWM0CO N1	PWM0 control 1	DFH	0	PWMMOD1	PWMMOD0	GP	PWMTYP	FBINEN	PWMDIV2	PWMDIV1	PWMDIV0	0000 0000b	
-	-	DFH	1	-	-	-	-	-	-	-	-	-	
SC0TSR	SC0 Transfer Status Register	DFH	2	ACT	BEF	FEF	PEF	TXEMPTY	TXOV	RXEMPTY	RXOV	0000 1010b	
PIOCON0	PWM0 I/O switch 0	DEH	0	-	-	PIO03	PIO01	PIO00	PIO10	PIO11	PIO12	0000 0000b	
-	-	DEH	1	-	-	-	-	-	-	-	-	-	
SC0IS	SC0 Interrupt Status Register	DEH	2	-	-	-	ACERRIF	BGTIF	TERRIF	TBEIF	RDAIF	0000 0010b	

Register	Definition	Address	Page	MSB 7	6	5	4	3	2	1	LSB ^[1] 0	Reset Value ^[2]	TA
PWM0C3L	PWM0 channel 3 duty low byte	DDH	0	PWM0_CH3[7:0]								0000 0000b	
-	-	DDH	1	-	-	-	-	-	-	-	-	-	
SC0IE	SC0 Interrupt Enable Control Register	DDH	2	-	-	-	ACERRIEN	BGTIEN	TERRIEN	TBEIEN	RDAIEN	0000 0000b	
PWM0C2L	PWM0 channel 2 duty low byte	DCH	0	PWM0_CH2[7:0]								0000 0000b	
-	-	DCH	1	-	-	-	-	-	-	-	-	-	
SC0ETURD1	SC0 ETU Rate Divider Register	DCH	2	-	SCDIV2	SCDIV1	SCDIV0	ETURDIV1_1	ETURDIV1_0	ETURDIV9	ETURDIV8	0011 0001b	
PWM0C1L	PWM0 channel 1 duty low byte	DBH	0	PWM0_CH1[7:0]								0000 0000b	
-	-	DBH	1	-	-	-	-	-	-	-	-	-	
SC0ETURD0	SC0 ETU Rate Divider Register	DBH	2	ETURDIV[7:0]								0111 0011b	
PWM0C0L	PWM0 channel 0 duty low byte	DAH	0	PWM0C0[7:0]								0000 0000b	
-	-	DAH	1	-	-	-	-	-	-	-	-	-	
SC0EGT	SC0Cn0 Extra Guard Time Register	DAH	2	SC0EGT[7:0]								0000 0000b	
PWM0PL	PWM0 period low byte	D9H	0	PWM0P[7:0]								0000 0000b	
-	-	D9H	1	-	-	-	-	-	-	-	-	-	
SC0DR	SC0 Data Register	D9H	2	SC0DR[7:0]								0000 0000b	
PWM0CON0	PWM0 control register 0	D8H	0	PWM0RUN	LOAD	PWMF	CLRPWM	PWM3RUN	PWM2RUN	PWM1RUN	P33FBINEN	0000 0000b	
PWM0FBD	Brake data	D7H	0	FBF	FBINLS	FBD5	FBD4	FBD3	FBD2	FBD1	R/W	0000 0000b	
-	-	D7H	1	-	-	-	-	-	-	-	-	-	
XTLCON	XTL Clock Control	D7H	2	-	HXSG2	HXSG1	HXSG0	-	-	-	-	0111 0111b	Y
PNP	PWM0 negative polarity	D6H	0	-	-	PNP5	PNP4	PNP3	PNP2	PNP1	PNP0	0000 0000b	
-	-	D6H	1	-	-	-	-	-	-	-	-	-	
PWM3INTC	PWM3 Interrupt Control	D6H	2	-	-	INTTYP1	INTTYP0	-	INTSEL2	INTSEL1	INTSEL0	0000 0000b	
PWM0C3H	PWM0 channel 3 duty high byte	D5H	0	PWM0_CH3[15:8]								0000 0000b	
-	-	D5H	1	-	-	-	-	-	-	-	-	-	
PWM3CON1	PWM3 control 1	D5H	2	PWMMOD1	PWMMOD0	PWMTYP	-	-	PWMDIV2	PWMDIV1	PWMDIV0	0000 0000b	
PWM0C2H	PWM0 channel 2 duty high byte	D4H	0	PWM0_CH2[15:8]								0000 0000b	
-	-	D4H	1	-	-	-	-	-	-	-	-	-	
PWM3CON0	PWM3 control 0	D4H	2	PWMRUN	LOAD	PWMF	CLRPWM	-	-	-	-	0000 0000b	
PWM0C1H	PWM0 channel 1 duty high byte	D3H	0	PWM0_CH1[15:8]								0000 0000b	
-	-	D3H	1	-	-	-	-	-	-	-	-	-	
PWM3C1L	PWM3 channel 1 duty low byte	D3H	2	PWM3_CH1[7:0]								0000 0000b	
PWM0C0H	PWM0 channel 0 duty high byte	D2H	0	PWM0C0[15:8]								0000 0000b	
-	-	D2H	1	-	-	-	-	-	-	-	-	-	
PWM3C0L	PWM3 channel 0 duty low byte	D2H	2	PWM3_CH0[7:0]								0000 0000b	
PWM0PH	PWM0 period high byte	D1H	0	PWM0P[15:8]								0000 0000b	
-	-	D1H	1	-	-	-	-	-	-	-	-	-	
PWM3PL	PWM3 period low byte	D1H	2	PWM3P[7:0]								0000 0000b	
PSW	Program status word	D0H	A	CY	AC	F0	RS1	RS0	OV	-	P	0000 0000b	
ADCMPLH	ADC compare high byte	CFH	0	ADCMPL[11:4]								0000 0000b	
-	-	CFH	1	-	-	-	-	-	-	-	-	-	
EIP2H	Extensive Interrupt Priority High 2	CFH	2	-	-	-	-	-	PUART4H	PUART3H	PUART2H	0000 0000b	
ADCMPL	ADC compare low byte	CEH	0	-	-	-	-	ADCMP3	ADCMP2	ADCMP1	ADCMP0	0000 0000b	
-	-	CEH	1	-	-	-	-	-	-	-	-	-	
EIP2L	Extensive interrupt priority 1	CEH	2	-	-	-	-	-	PUART4	PUART3	PUART2	0000 0000b	
TH2	Timer 2 high byte	CDH	0	TH2[7:0]								0000 0000b	
PWM0C5L	PWM0 channel 5 duty low byte	CDH	1	PWM0_CH5[7:0]								0000 0000b	

Register	Definition	Address	Page	MSB 7	6	5	4	3	2	1	LSB ^[1] 0	Reset Value ^[2]	TA
PWM3MEN	PWM3 mask enable	CDH	2	-	-	-	-	-	-	PMEN1	PMEN0	0000 0000b	
TL2	Timer 2 low byte	CCH	0	TL2[7:0]								0000 0000b	
PWM0C4L	PWM0 channel 4 duty low byte	CCH	1	PWM0_CH4[7:0]								0000 0000b	
PWM3MD	PWM3 mask data	CCH	2	-	-	-	-	-	-	PMD1	PMD0	0000 0000b	
RCMP2H	Timer 2 compare high byte	CBH	0	RCMP2H[7:0]								0000 0000b	
-	-	CBH	1	-	-	-	-	-	-	-	-	-	
PWM3C1H	PWM3 channel 1 duty high byte	CBH	2	PWM3_CH1[15:8]								0000 0000b	
RCMP2L	Timer 2 compare low byte	CAH	0	RCMP2L[7:0]								0000 0000b	
-	-	CAH	1	-	-	-	-	-	-	-	-	-	
PWM3C0H	PWM3 channel 0 duty high byte	CAH	2	PWM3_CH0[15:8]								0000 0000b	
T2MOD	Timer 2 mode	C9H	0	LDEN	T2DIV2	T2DIV1	T2DIV0	CAPCR	CMPCR	LDS1	LDS0	0000 0000b	
-	-	C9H	1	-	-	-	-	-	-	-	-	-	
PWM3PH	PWM3 period high byte	C9H	2	PWM3P[15:8]								0000 0000b	
T2CON	Timer 2 control	C8H	A	TF2	-	-	-	-	TR2	-	CM/RL2	0000 0000b	
TA	Timed access protection	C7H	A	TA[7:0]								0000 0000b	
RH3	Timer 3 reload high byte	C6H	0	RH3[7:0]								0000 0000b	
PIOCON1	PWM0 I/O switch 1	C6H	1	PIO17	-	PIO15	-	PIO04	PIO05	PIO14	-	-	
PWM2INTC	PWM2 Interrupt Control	C6H	2	-	-	INTTPY1	INTTPY0	-	INTSEL2	INTSEL1	INTSEL0	0000 0000b	
RL3	Timer 3 reload low byte	C5H	0	RL3[7:0]								0000 0000b	
PWM0C5H	PWM0 channel 5 duty high byte	C5H	1	PWM0_CH5[15:8]								0000 0000b	
PWM2CON1	PWM2 control 1	C5H	2	PWMMOD1	PWMMOD0	PWMTYP	-	-	PWMDIV2	PWMDIV1	PWMDIV0	0000 0000b	
T3CON	Timer 3 control	C4H	0	SMOD_1	SMOD_0	BRCK	TF3	TR3	T3PS2	T3PS1	T3PS0	0000 0000b	-
PWM0C4H	PWM0C4duty high byte	C4H	1	PWM0_CH4[15:8]								0000 0000b	
PWM2CON0	PWM2 control 1	C4H	2	PWMRUN	LOAD	PWMF	CLRPWM	-	-	-	-	0000 0000b	
ADCRH	ADC result high byte	C3H	0	ADCR[11:4]								0000 0000b	
-	-	C3H	1	-	-	-	-	-	-	-	-	-	
PWM2C1L	PWM2 channel 1 duty low byte	C3H	2	PWM2_CH1[7:0]								0000 0000b	
ADCRL	ADC result low byte	C2H	0	-	-	-	-	ADCR3	ADCR2	ADCR1	ADCR0	0000 0000b	
-	-	C2H	1	-	-	-	-	-	-	-	-	-	
PWM2C0L	PWM2 channel 0 duty low byte	C2H	2	PWM2_CH0[7:0]								0000 0000b	
I2ADDR	I2C own slave address	C1H	0	I2ADDR7	I2ADDR6	I2ADDR5	I2ADDR4	I2ADDR3	I2ADDR2	I2ADDR1	GC	0000 0000b	
-	-	C1H	1	-	-	-	-	-	-	-	-	-	
PWM2PL	PWM2 period low byte	C1H	2	PWM2P[7:0]								0000 0000b	
I2CON	I2C control	C0H	A	-	I2CEN	STA	STO	SI	AA	-	I2CPX	0000 0000b	
I2TOC	I2C time-out counter	BFH	0	-	-	-	-	-	I2TOCEN	DIV	I2TOF	0000 0000b	
-	-	BFH	1	-	-	-	-	-	-	-	-	-	
PWM2MEN	PWM2 mask enable	BFH	2	-	-	-	-	-	-	PMEN1	PMEN0	0000 0000b	
I2CLK	I2C clock	BEH	0	I2CLK[7:0]								0000 1001b	
-	-	BEH	1	-	-	-	-	-	-	-	-	-	
PWM2MD	PWM2 mask data	BEH	2	-	-	-	-	-	-	PMD1	PMD0	0000 0000b	
I2STAT	I2C status	BDH	0	I2STAT7	I2STAT6	I2STAT5	I2STAT4	I2STAT3	0	0	0	1111 1000b	
-	-	BDH	1	-	-	-	-	-	-	-	-	-	
PWM2C1H	PWM2 channel 1 duty high byte	BDH	2	PWM2_CH1[15:8]								0000 0000b	
I2DAT	I2C data	BCH	0	I2DAT[7:0]								0000 0000b	
-	-	BCH	1	-	-	-	-	-	-	-	-	-	
PWM2C0H	PWM2 channel 0 duty high byte	BCH	2	PWM2_CH0[15:8]								0000 0000b	

Register	Definition	Address	Page	MSB 7	6	5	4	3	2	1	LSB ^[1] 0	Reset Value ^[2]	TA
SADDR_1	Slave 1 address	BBH	0	SADDR_1[7:0]								0000 0000b	
-	-	BBH	1	-	-	-	-	-	-	-	-	-	
PWM2PH	PWM2 period high byte	BBH	2	PWM2P[15:8]								0000 0000b	
SADEN_1	Slave 1 address mask	BAH	0	SADEN_1[7:0]								0000 0000b	
-	-	BAH	1	-	-	-	-	-	-	-	-	-	
-	-	BAH	2	-	-	-	-	-	-	-	-	-	
SADEN	Slave 0 address mask	B9H	0	SADEN[7:0]								0000 0000b	
-	-	B9H	1	-	-	-	-	-	-	-	-	-	
-	-	B9H	2	-	-	-	-	-	-	-	-	-	
IP	Interrupt priority	B8H	A	-	PADC	PBOD	PS	PT1	PX1	PT0	PX0	0000 0000b	
IPH	Interrupt priority high	B7H	0	-	PADCH	PBODH	PSH	PT1H	PX1H	PT0H	PX0H	0000 0000b	
-	-	B7H	1	-	-	-	-	-	-	-	-	-	
PIOCON2	PWM or I/O Select 2	B7H	2	PIO34	PIO33	PIO32	PIO31	-	PIO23	PIO22	PIO21	0000 0000b	
-	-	B6H	0	-	-	-	-	-	-	-	-	-	
-	-	B6H	1	-	-	-	-	-	-	-	-	-	
PWM1INTC	PWM1 Interrupt Control	B6H	2	-	-	INTTYP1	INTTYP0	-	INTSEL2	INTSEL1	INTSEL0	0000 0000b	
TOE	Timer01 output enable	B5H	0	-	-	-	-	T1OE	T0OE	-	-	0000 0000b	
-	-	B5H	1	-	-	-	-	-	-	-	-	-	
PWM1CON1	PWM1 control 1	B5H	2	PWMMOD1	PWMMOD0	-	PWMTYP	-	PWMDIV2	PWMDIV1	PWMDIV0	0000 0000b	
P1M2	P1 mode select 2	B4H	0	P1M2.7	P1M2.6	P1M2.5	P1M2.4	P1M2.3	P1M2.2	P1M2.1	P1M2.0	0000 0000b	
P1SR	P1 slew rate	B4H	1	P1SR.7	P1SR.6	P1SR.5	P1SR.4	P1SR.3	P1SR.2	P1SR.1	P1SR.0	0000 0000b	
PWM1CON0	PWM1 control 0	B4H	2	PWMRUN	LOAD	PWMF	CLRPWM	-	-	-	-	0000 0000b	
P1M1	P1 mode select 1	B3H	0	P1M1.7	P1M1.6	P1M1.5	P1M1.4	P1M1.3	P1M1.2	P1M1.1	P1M1.0	1111 1111b	
P1S	P1 Schmitt trigger input	B3H	1	P1S.7	P1S.6	P1S.5	P1S.4	P1S.3	P1S.2	P1S.1	P1S.0	0000 0000b	
PWM1C1L	PWM1 channel 1 duty low byte	B3H	2	PWM1_CH1[7:0]								0000 0000b	
P0M2	P0 mode select 2	B2H	0	P0M2.7	P0M2.6	P0M2.5	P0M2.4	P0M2.3	P0M2.2	P0M2.1	P0M2.0	0000 0000b	
P0SR	P0 slew rate	B2H	1	P0SR.7	P0SR.6	P0SR.5	P0SR.4	P0SR.3	P0SR.2	P0SR.1	P0SR.0	0000 0000b	
PWM1C0L	PWM1 channel 0 duty low byte	B2H	2	PWM1_CH0[7:0]								0000 0000b	
P0M1	P0 mode select 1	B1H	0	P0M1.7	P0M1.6	P0M1.5	P0M1.4	P0M1.3	P0M1.2	P0M1.1	P0M1.0	1111 1111b	
P0S	P0 Schmitt trigger input	B1H	1	P0S.7	P0S.6	P0S.5	P0S.4	P0S.3	P0S.2	P0S.1	P0S.0	0000 0000b	
PWM1PL	PWM1 period low byte	B1H	2	PWM1P[7:0]								0000 0000b	
P3	Port 3	B0H	A	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	Output latch, 1111 1111b Input xxxx xxxxb ^[3]	
IAPCN	IAP control	AFH	0	IAPA17	IAPA16	FOEN	FCEN	FCTRL3	FCTRL2	FCTRL1	FCTRL0	0011 0000b	
PWM0INTC	PWM0 Interrupt Control	AFH	1	-	-	INTTYP1	INTTYP0	-	INTSEL2	INTSEL1	INTSEL0	0000 0000b	
-	-	AFH	2	-	-	-	-	-	-	-	-	-	
IAPFD	IAP flash data	AEH	0	IAPFD[7:0]								0000 0000b	
-	-	AEH	1	-	-	-	-	-	-	-	-	-	
PWM1MEN	PWM1 mask enable	AEH	2	-	-	-	-	-	-	PMEN1	PMEN0	0000 0000b	
P3M2	P3 mode select 2	ADH	0	P3M2.7	P3M2.6	P3M2.5	P3M2.4	P3M2.3	P3M2.2	P3M2.1	P3M2.0	0000 0000b	
P3SR	P3 slew rate	ADH	1	P3SR.7	P3SR.6	P3SR.5	P3SR.4	P3SR.3	P3SR.2	P3SR.1	P3SR.0	0000 0000b	
PWM1MD	PWM1 mask data	ADH	2	-	-	-	-	-	-	PMD1	PMD0	0000 0000b	
P3M1	P3 mode select 1	ACH	0	P3M1.7	P3M1.6	P3M1.5	P3M1.4	P3M1.3	P3M1.2	P3M1.1	P3M1.0	1111 1111b	
P3S	P3 Schmitt trigger input	ACH	1	P3S.7	P3S.6	P3S.5	P3S.4	P3S.3	P3S.2	P3S.1	P3S.0	0000 0000b	
PWM1C1H	PWM1 channel 1 duty high byte	ACH	2	PWM1_CH1[15:8]								0000 0000b	
BODCON1	Brown-out detection control 1	ABH	0	-	-	-	-	-	LPBOD1	LPBOD0	BODFLT	POR, 0000 0001b Others, 0000 0000b	Y
-	-	ABH	1	-	-	-	-	-	-	-	-	-	

Register	Definition	Address	Page	MSB 7	6	5	4	3	2	1	LSB ^[1] 0	Reset Value ^[2]	TA
PWM1C0H	PWM1 channel 0 duty high byte	ABH	2	PWM1_CH0[15:8]								0000 0000b	
WDCON	Watchdog Timer control	AAH	0	WDTR	WDCLR	WDTF	WIDPD	WDTRF	WDPS2	WDPS1	WDPS0	POR, 0000 0111b WDT, 0000 1UUUUb Others, 0000 UUUUUb	Y
-	-	AAH	1	-	-	-	-	-	-	-	-	-	
PWM1PH	PWM1 period high byte	AAH	2	PWM1P[15:8]								0000 0000b	
SADDR	Slave 0 address	A9H	0	SADDR[7:0]								0000 0000b	
-	-	A9H	1	-	-	-	-	-	-	-	-	-	
PWM1PH	PWM1 period high byte	A9H	2	PWM1P[15:8]								0000 0000b	
IE	Interrupt enable	A8H	A	EA	EADC	EBOD	ES	ET1	EX1	ET0	EX0	0000 0000b	
IAPAH	IAP address high byte	A7H	0	IAPA[15:8]								0000 0000b	
-	-	A7H	1	-	-	-	-	-	-	-	-	-	
AUXR8	Auxiliary Register 8	A7H	2	CLODIV3	CLODIV2	CLODIV1	CLODIV0	CKTESTOE N.3	CKTESTOE N.2	CKTESTOE N.1	CKTESTOE N.0	0000 0000b	Y
IAPAL	IAP address low byte	A6H	0	IAPA[7:0]								0000 0000b	
-	-	A6H	1	-	-	-	-	-	-	-	-	-	
AUXR7	Auxiliary Register 7	A6H	2	-	-	-	SPI0NSSP 1	SPI0NSSP 0	SPI0MOSIP	SPI0MISOP	SPI0CKP	0000 0000b	
IAPUEN	IAP update enable	A5H	0	-	-	-	SPMEN	SPUEN	CFUEN	LDUEN	APUEN	0000 0000b	Y
-	-	A5H	1	-	-	-	-	-	-	-	-	-	
AUXR6	Auxiliary Register 6	A5H	2	-	-	-	UART4DG	UART3DG	UART2DG	UART1DG	UART0DG	0000 0000b	
IAPTRG	IAP trigger	A4H	0	-	-	-	-	-	-	-	IAPGO	0000 0000b	Y
-	-	A4H	1	-	-	-	-	-	-	-	-	-	
AUXR5	Auxiliary Register 5	A4H	2	CLOP	T0P	-	-	PWM3CH1 P1	PWM3CH1 P0	PWM3CH0 P1	PWM3CH0 P0	0000 0000b	
BODCON0	Brown-out detection control 0	A3H	0	BODEN	-	BOV1	BOV0	BOF	BORST	BORF	BOS	POR, CCCC XC0Xb BOD, UUUU XU1Xb Others, UUUU XUUXb	Y
-	-	A3H	1	-	-	-	-	-	-	-	-	-	
AUXR4	Auxiliary Register 4	A3H	2	PWM2CH1 P1	PWM2CH1 P0	PWM2CH0 P1	PWM2CH0 P0	PWM1CH1 P1	PWM1CH1 P0	PWM1CH0 P1	PWM1CH0 P0	0000 0000b	
AUXR1	Auxiliary register 1	A2H	0	SWRF	RSTPINF	HardF	SLOW	GF2	UART0PX	0	DPS	POR, 0000 0000b Software, 1U00 0000b nRESET pin, U100 0000b Others, UUUU 0000b	
-	-	A2H	1	-	-	-	-	-	-	-	-	-	
AUXR3	Auxiliary Register 3	A2H	2	UART4TXP 1	UART4TXP 0	UART4RXP 1	UART4RXP 0	UART3TXP 1	UART3TXP 0	UART3RXP 1	UART3RXP 0	0000 0000b	
-	-	A1H	0	-	-	-	-	-	-	-	-	-	
-	-	A1H	1	-	-	-	-	-	-	-	-	-	
AUXR2	Auxiliary Register 2	A1H	2	UART2TXP 1	UART2TXP 0	UART2RXP 1	UART2RXP 0	UART1TXP 1	UART1TXP 0	UART1RXP 1	UART1RXP 0	0000 0000b	
P2	Port 2	A0H	A	-	-	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	Output latch, 0000 000Xb Input, 0000 000Xb	
CHPCON	Chip control	9FH	0	SWRST	IAPFF	-	-	-	-	BS[5]	IAPEN	Software, 0000 00U0b Others, 0000 00C0b	Y
-	-	9FH	1	-	-	-	-	-	-	-	-	-	
-	-	9FH	2	-	-	-	-	-	-	-	-	-	
-	-	9EH	0	-	-	-	-	-	-	-	-	-	
-	-	9EH	1	-	-	-	-	-	-	-	-	-	
-	-	9EH	2	-	-	-	-	-	-	-	-	-	
RSR	Reset Flag Register	9DH	0	-	-	HardF(mirrored from AUXR1.5)	POF (mirrored from PCON.4)	RSTPINF (mirrored from AUXR1.6)	BORF (mirrored from BODCON.1)	WDTRF (mirrored from WDCON.3)	SWRF (mirrored from AUXR1.7)	00XX XXXXb	
-	-	9DH	1	-	-	-	-	-	-	-	-	-	

Register	Definition	Address	Page	MSB 7	6	5	4	3	2	1	LSB ^[1] 0	Reset Value ^[2]	TA
P3DW	Port 3 Pull-Down resistor control	9DH	2	P3DW.7	P3DW.6	P3DW.5	P3DW.4	P3DW.3	P3DW.2	P3DW.1	P3DW.0	0000 0000b	
EIE1	Extensive interrupt enable 1	9CH	0	-	-	EPWM3	EPWM2	EPWM1	EWKT	ET3	ES_1	0000 0000b	
-	-	9CH	1	-	-	-	-	-	-	-	-	-	
P2DW	Port 2 Pull-Down resistor control	9CH	2	-	-	P2DW.5	P2DW.4	P2DW.3	P2DW.2	P2DW.1	P2DW.0	0000 0000b	
EIE	Extensive interrupt enable	9BH	0	ET2	ESPI	EFB	EWDT	EPWM0	ECAP	EPI	EI2C	0000 0000b	
-	-	9BH	1	-	-	-	-	-	-	-	-	-	
P1DW	Port 1 Pull-Down resistor control	9BH	2	P1DW.7	P1DW.6	P1DW.5	P1DW.4	P1DW.3	P1DW.2	P1DW.1	P1DW.0	0000 0000b	
SBUF_1	Serial port 1 data buffer	9AH	0	SBUF_1[7:0]								0000 0000b	
-	-	9AH	1	-	-	-	-	-	-	-	-	-	
P0DW	Port 0 Pull-Down resistor control	9AH	2	P0DW.7	P0DW.6	P0DW.5	P0DW.4	P0DW.3	P0DW.2	P0DW.1	P0DW.0	0000 0000b	
SBUF	Serial port 0 data buffer	99H	0	SBUF[7:0]								0000 0000b	
-	-	99H	1	-	-	-	-	-	-	-	-	-	
AINDIDS1	ADC channel digital input disable	99H	2	P25DIDS	P14DIDS	P13DIDS	P24DIDS	P23DIDS	P22DIDS	P21DIDS	-	0000 0000b	
SCON	Serial port 0 control	98H	A	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	0000 0000b	
CKEN	Clock enable	97H	0	EXTEN1	EXTEN0	HIRCEN	LIRCEN	-	-	-	CKSWTF	0011 0000b	Y
-	-	97H	1	-	-	-	-	-	-	-	-	-	
RWKH	Self Wake-up Timer Reload High Byte	97H	2	RWK[15:8]								0000 0000b	
CKSWT	Clock switch	96H	0	HXTST	ECKP00ST	HIRCST	-	ECKP30ST	OSC1	OSC0	-	0011 0000b	Y
-	-	96H	1	-	-	-	-	-	-	-	-	-	
-	-	96H	2	-	-	-	-	-	-	-	-	0000 0000b	
CKDIV	Clock divider	95H	0	CKDIV[7:0]								0000 0000b	
-	-	95H	1	-	-	-	-	-	-	-	-	-	
P3UP	Port 3 Pull-Up resistor control	95H	2	P3UP.7	P3UP.6	P3UP.5	P3UP.4	P3UP.3	P3UP.2	P3UP.1	P3UP.0	0000 0000b	
CAPCON2	Input capture control 2	94H	0	-	ENF2	ENF1	ENF0	-	-	-	-	0000 0000b	
-	-	94H	1	-	-	-	-	-	-	-	-	-	
P2UP	Port 2 Pull-Up resistor control	94H	2	-	-	P2UP.5	P2UP.4	P2UP.3	P2UP.2	P2UP.1	P2UP.0	0000 0000b	
CAPCON1	Input capture control 1	93H	0	-	-	CAP2LS1	CAP2LS0	CAP1LS1	CAP1LS0	CAP0LS1	CAP0LS0	0000 0000b	
-	-	93H	1	-	-	-	-	-	-	-	-	-	
P1UP	Port 1 Pull-Up resistor control	93H	2	P1UP.7	P1UP.6	P1UP.5	P1UP.4	P1UP.3	P1UP.2	P1UP.1	P1UP.0	0000 0000b	
CAPCON0	Input capture control 0	92H	0	-	CAPEN2	CAPEN1	CAPEN0	-	CAPF2	CAPF1	CAPF0	0000 0000b	
-	-	92H	1	-	-	-	-	-	-	-	-	-	
P0UP	Port 0 Pull-Up resistor control	92H	2	P0UP.7	P0UP.6	P0UP.5	P0UP.4	P0UP.3	P0UP.2	P0UP.1	P0UP.0	0000 0000b	
SFRS	SFR page selection	91H	A	-	-	-	-	-	-	SFRPSEL1	SFRPSEL0	0000 0000b	
P1	Port 1	90H	A	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	Output latch, 1111 1111b Input XXXX XXXXb	
WKCON	Self Wake-up Timer control	8FH	0	-	-	-	WKTF	WKTR	WKPS2	WKPS1	WKPS0	0000 0000b	
-	-	8FH	1	-	-	-	-	-	-	-	-	-	
ADCSR	ADC Status Register	8FH	2	-	-	-	-	-	CMPHIT	HDONE	FDONE	0000 0000b	
CKCON	Clock control	8EH	0	-	PWMCKS	-	T1M	T0M	-	CLOEN	-	0000 0000b	
-	-	8EH	1	-	-	-	-	-	-	-	-	-	
ADCCN	ADC Current Sampling Number	8EH	2	ADCCN[7:0]								0000 0000b	
TH1	Timer 1 high byte	8DH	0	TH1[7:0]								0000 0000b	
-	-	8DH	1	-	-	-	-	-	-	-	-	-	
ADCSN	ADC Current Sampling Number	8DH	2	ADCSN[7:0]								0000 0000b	
TH0	Timer 0 high byte	8CH	0	TH0[7:0]								0000 0000b	
-	-	8CH	1	-	-	-	-	-	-	-	-	-	
P2S	Port 2 Schmitt Triggered Input	8CH	2	P2S.7	P2S.6	P2S.5	P2S.4	P2S.3	P2S.2	P2S.1	P2S.0	0000 0000b	
TL1	Timer 1 low byte	8BH	0	TL1[7:0]								0000 0000b	
-	-	8BH	1	-	-	-	-	-	-	-	-	-	

Register	Definition	Address	Page	MSB 7	6	5	4	3	2	1	LSB ^[1] 0	Reset Value ^[2]	TA
P2SR	Port 2 Slew Rate Control	8BH	2	P2SR.7	P2SR.6	P2SR.5	P2SR.4	P2SR.3	P2SR.2	P2SR.1	P2SR.0	0000 0000b	
TL0	Timer 0 low byte	8AH	0	TL1[7:0]								0000 0000b	
-	-	8AH	1	-	-	-	-	-	-	-	-	-	
P2M2	P2 Mode Select 2	8AH	2	P2M2.7	P2M2.6	P2M2.5	P2M2.4	P2M2.3	P2M2.2	P2M2.1	P2M2.0	0000 0000b	
TMOD	Timer 0 and 1 Mode	89H	0	GATE	C/T	M1	M0	GATE	C/T	M1	M0	0000 0000b	
-	-	89H	1	-	-	-	-	-	-	-	-	-	
P2M1	P2 Mode Select 1	89H	2	P2M1.7	P2M1.6	P2M1.5	P2M1.4	P2M1.3	P2M1.2	P2M1.1	P2M1.0	0000 0000b	
TCON	Timer 0 and 1 control	88H	A	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	0000 0000b	
PCON	Power control	87H	A	SMOD	SMOD0	-	POF	GF1	GF0	PD	IDL	POR, 0001 0000b Others, 000U 0000b	
RWKL	Self Wake-up Timer reload low byte	86H	0	RWK[7:0]								0000 0000b	
-	-	86H	1	-	-	-	-	-	-	-	-	-	
ADCCON3	ADC Control 3	86H	2	-	-	HIE	CONT	ADCAQT2	ADCAQT1	ADCAQT0	SLOW	0000 0000b	
RCTRIM1	Internal RC trim value low byte	85H	0	-	-	-	HIRC24	-	-	-	HIRCTRIM.0	0000 0000b	Y
-	-	85H	1	-	-	-	-	-	-	-	-	-	
ADCBAH	ADC RAM Base Address High byte	85H	2	-	-	-	-	ADCBA3	ADCBA2	ADCBA1	ADCBA0	0000 0000b	
RCTRIM0	Internal RC trim value high byte	84H	0	HIRCTRIM[8:1]								0000 0000b	Y
-	-	84H	1	-	-	-	-	-	-	-	-	-	
ADCBAL	ADC RAM Base Address Low Byte	84H	2	ADCBAL[7:0]								0000 0000b	
DPH	Data pointer high byte	83H	A	DPTR[15:8]								0000 0000b	
DPL	Data pointer low byte	82H	A	DPTR[7:0]								0000 0000b	
SP	Stack pointer	81H	A	SP[7:0]								0000 0111b	
P0	Port 0	80H	A	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	Output latch, 1111 1111b Input, XXXX XXXXb	
<p>Note:</p> <ol style="list-style-type: none"> () item means the bit address in bit-addressable SFRs. Reset value symbol description. 0: logic 0; 1: logic 1; U: unchanged; C: see [5]; X: see [3], [6], and [7]. All I/O pins are default input-only mode (floating) after reset. Reading back P2.0 is always 0 if RPD (CONFIG0.2) remains un-programmed 1. After reset ICE_DAT and ICE_CLK pin will keep quasi mode with pull high resistor 600 LIRC clock before change to input mode. These SFRs have TA protected writing. These SFRs have bits those are initialized according to CONFIG values after specified resets. BOF reset value depends on different setting of CONFIG2 and V_{DD} voltage level. Please check BOS is a read-only flag decided by V_{DD} level while brown-out detection is enabled. 													

Table 6.1-2 SFR Definitions and Reset Values

6.1.7.5 All SFR Description

Note:

1. All SFRs reset value show as following means U-unchanged; C-initialized by CONFIG; X- base on real chip status.

Pn – Port (Bit-addressable)

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	0011_1111 b
P3	B0H, All pages, Bit-addressable	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]	<p>Stack pointer</p> <p>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.</p>

DPL – Data Pointer Low Byte

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

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

Bit	Name	Description
7:0	DPL[7:0]	<p>Data pointer low byte</p> <p>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.</p>

DPH – Data Pointer High Byte

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

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

Bit	Name	Description
7:0	DPH[7:0]	<p>Data pointer high byte</p> <p>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.</p>

ADCBAH – ADC RAM Base Address High Byte

Register	SFR Address	Reset Value
ADCBAH	84H, Page 2	0000_0000 b

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

Bit	Name	Description
7:4	-	Reserved
3:0	ADCBAH[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	85H, Page 2	0000_0000 b

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

Bit	Name	Description
7:0	ADCBAL[7:0]	<p>ADC RAM base address (Low byte)</p> <p>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]}</p>

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]	<p>WKT reload low byte</p> <p>The RWKL register is the low byte of the 16-bit reload value of WKT.</p> <p>Note that RWK should not be FFFFH if the pre-scale is 1/1 for implement limitation.</p>

RWKH – Self Wake-up Timer Reload High Byte

Register	SFR Address	Reset Value
RWKH	97H, 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]	<p>WKT reload high byte</p> <p>The RWKH register is the low byte of the 16-bit reload value of WKT.</p> <p>Note that RWK should not be FFFFH if the pre-scale is 1/1 for implement limitation.</p>

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	-	POF	GF1	GF0	PD	IDL
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 6.9-1 Serial Port UART0 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.
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.
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 $\overline{INT1}$ input signal's logic level. Software cannot control it.
2	IT1	External interrupt 1 type select This bit selects by which type that $\overline{INT1}$ is triggered. 0 = $\overline{INT1}$ is low level triggered. 1 = $\overline{INT1}$ 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 $\overline{INT0}$ input signal's logic level. Software cannot control it.

Bit	Name	Description
0	IT0	<p>External interrupt 0 type select</p> <p>This bit selects by which type that $\overline{INT0}$ is triggered.</p> <p>0 = $\overline{INT0}$ is low level triggered.</p> <p>1 = $\overline{INT0}$ is falling edge triggered.</p>

TMOD – Timer 0 and 1 Mode

Register	SFR Address	Reset Value
TMOD	89H, Page0	0000_0000b

7	6	5	4	3	2	1	0
GATE	C/ \bar{T}	M1	M0	GATE	C/ \bar{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 $\overline{INT1}$ logic level. 1 = Timer 1 will clock only when TR1 is 1 and $\overline{INT1}$ is logic 1.															
6	C/ \bar{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 <table border="1"> <thead> <tr> <th>M1</th> <th>M0</th> <th>Timer 1 Mode</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Mode 0: 13-bit Timer/Counter</td> </tr> <tr> <td>0</td> <td>1</td> <td>Mode 1: 16-bit Timer/Counter</td> </tr> <tr> <td>1</td> <td>0</td> <td>Mode 2: 8-bit Timer/Counter with auto-reload from TH1</td> </tr> <tr> <td>1</td> <td>1</td> <td>Mode 3: Timer 1 halted</td> </tr> </tbody> </table>	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
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																
3	GATE	Timer 0 gate control 0 = Timer 0 will clock when TR0 is 1 regardless of $\overline{INT0}$ logic level. 1 = Timer 0 will clock only when TR0 is 1 and $\overline{INT0}$ is logic 1.															
2	C/ \bar{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 <table border="1"> <thead> <tr> <th>M1</th> <th>M0</th> <th>Timer 0 Mode</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Mode 0: 13-bit Timer/Counter</td> </tr> <tr> <td>0</td> <td>1</td> <td>Mode 1: 16-bit Timer/Counter</td> </tr> <tr> <td>1</td> <td>0</td> <td>Mode 2: 8-bit Timer/Counter with auto-reload from TH0</td> </tr> <tr> <td>1</td> <td>1</td> <td>Mode 3: TL0 as a 8-bit Timer/Counter and TH0 as a 8-bit Timer</td> </tr> </tbody> </table>	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
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																

TL0 – Timer 0 Low Byte

Register	SFR Address	Reset Value
TL0	8AH, Page0	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.

TL1 – Timer 1 Low Byte

Register	SFR Address	Reset Value
TL1	8BH, Page0	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, Page0	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, Page0	0000_0000b

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

Bit	Name	Description
7:0	TH1[7:0]	<p>Timer 1 high byte</p> <p>The TH1 register is the high byte of the 16-bit counting register of Timer 1.</p>

CKCON – Clock Control

Register	SFR Address	Reset Value
CKCON	8EH, Page0	0000_0000b

7	6	5	4	3	2	1	0
-	PWMCKS	-	T1M	T0M	-	CLOEN	-
-	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 F_{SYS} . 1 = The clock source of PWM is the overflow of Timer 1.
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.
1	CLOEN	System clock output enable 0 = System clock output Disabled. 1 = System clock output Enabled from CLO pin Note: CLO pin decide by AUXR5.7.

ADCCN – ADC Current Sampling Number

Register	SFR Address	Reset Value
ADCCN	8EH, Page 2	0000_0000 b

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

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

WKCON – Self Wake-up Timer Control

Register	SFR Address	Reset Value
WKCON	8FH, Page0	0000_0000b

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

Bit	Name	Description
4	WKTF	<p>WKT overflow flag</p> <p>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.</p>
3	WKTR	<p>WKT run control</p> <p>0 = WKT is halted. 1 = WKT starts running.</p> <p>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.</p>
2:0	WKPS[2:0]	<p>WKT pre-scalar</p> <p>These bits determine the pre-scale of WKT clock.</p> <p>000 = 1/1. 001 = 1/4. 010 = 1/16. 011 = 1/64. 100 = 1/256. 101 = 1/512. 110 = 1/1024. 111 = 1/2048.</p>

ADCSR – ADC Status Register

Register	SFR Address	Reset Value
ADCSR	8FH, Page 2	0000_0000 b

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

Bit	Description	
[7:3]	Reserved	Reserved
[2]	CMPHIT	<p>ADC comparator Hit Flag</p> <p>This bit is set by hardware when ADCMPO (ADCCON2.4) flag rising</p> <p>Note: This bit can be cleared by writing 0 to it.</p>
[1]	HDONE	<p>A/D Conversion Half Done Flag</p> <p>This bit is set by hardware when half of ADCSN A/D conversions are complete in continue mode.</p> <p>Note: This bit can be cleared by writing 0 to it</p>
[0]	FDONE	<p>A/D Conversion Full Done Flag</p> <p>This bit is set by hardware when all of ADCSN A/D conversions are complete in continue mode or single conversion in single mode.</p> <p>Note: This bit can be cleared by writing 0 to it..</p>

SFRS – SFR Page Selection

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

7	6	5	4	3	2	1	0
-	-	-	-	-	-	SFRPAGE1	SFRPAGE0
-	-	-	-	-	-	R/W	R/W

Bit	Name	Description
1:0	SFRPAGE[1:0]	SFR page select 0 = Instructions access SFR page 0. 1 = Instructions access SFR page 1. 2 = Instructions access SFR page 2.

CAPCON0 – Input Capture Control 0

Register	SFR Address	Reset Value
CAPCON0	92H, Page 0	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
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.
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.

PnUP – Port n Pull-up Resister Control

Register	SFR Address	Reset Value
P0UP	92H, page 2	0000_0000 b
P1UP	93H, page 2	0000_0000 b
P2UP	94H, page 2	0000_0000 b
P3UP	95H, page 2	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.x	Pn.x pull-up enable 0 = Pn.x pull-up Disabled. 1 = Pn.x pull-up Enabled.

CAPCON1 – Input Capture Control 1

Register	SFR Address	Reset Value
CAPCON1	93H, Page 0	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
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	94H, Page0	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 filter 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 filter 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 filter on input capture 0 0 = Noise filter on input capture channel 0 Disabled. 1 = Noise filter on input capture channel 0 Enabled.

CKDIV – Clock Divider

Register	SFR Address	Reset Value
CKDIV	95H, Page0	0000_0000b

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

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

CKSWT – Clock Switch

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

7	6	5	4	3	2	1	0
HXTST	ECKP00ST	HIRCST	LIRCST	ECKP30ST	OSC[1:0]		-
R	R	R	R	R	W		-

Bit	Name	Description
7	HXTST	High-speed external crystal 4 MHz to 24 MHz status 0 = High-speed external crystal is not stable or disabled. 1 = High-speed external crystal is enabled and stable.
6	ECKP00ST	External clock P00 input status 0 = External clock P00 input is not stable or disabled. 1 = External clock P00 input is enabled and stable.
5	HIRCST	High-speed internal oscillator 16 MHz status 0 = High-speed internal oscillator is not stable or disabled. 1 = High-speed internal oscillator is enabled and stable.
-	-	Reserved
3	ECKP30ST	External clock P30 input status 0 = External clock P30 input is not stable or disabled. 1 = External clock P30 input is enabled and stable.
2:1	OSC[1:0]	Oscillator selection bits This field selects the system clock source. 00 = Internal 16 MHz oscillator. 01 = External clock source according to EXTEN[1:0] (CKEN[7:6]) setting. 10 = Internal 10 kHz oscillator. 11 = Reserved. 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_0000 b

7	6	5	4	3	2	1	0
EXTEN[1:0]		HIRCEN	-	-	-	-	CKSWTF
R/W		R/W	-	-	-	-	R

Bit	Name	Description
7:6	EXTEN[1:0]	<p>External clock source enable</p> <p>11 = External clock input via OSCIN P3.0 Enabled 10 = External clock input via OSCIN P0.0 Enabled 01 = External crystal 4~24 MHz Enabled 00 = external clock input is disable. P30/P00/P01 work as general purpose I/O or other functions.</p>
5	HIRCEN	<p>High-speed internal oscillator 16 MHz enable</p> <p>0 = The high-speed internal oscillator Disabled. 1 = The high-speed internal oscillator Enabled.</p> <p>Note that once IAP is enabled by setting IAPEN (CHPCON.0), the high-speed internal 16 MHz oscillator will be enabled automatically. The hardware will also set HIRCEN and HIRCST bits. After IAPEN is cleared, HIRCEN and EHCST resume the original values.</p>
4:1	-	Reserved
0	CKSWTF	<p>Clock switch fault flag</p> <p>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.</p>

SCON – Serial Port Control

Register	SFR Address	Reset Value
SCON	98H, All page, 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
6	SM1	<p><u>SMOD0 (PCON.6) = 0:</u> See Table 6.9-1 Serial Port UART0 Mode / baud rate Description for details.</p> <p><u>SMOD0 (PCON.6) = 1:</u> 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.</p>
5	SM2	<p>Multiprocessor communication mode enable The function of this bit is dependent on the serial port 0 mode.</p> <p><u>Mode 0:</u> This bit select the baud rate between $F_{SYS}/12$ and $F_{SYS}/2$. 0 = The clock runs at $F_{SYS}/12$ baud rate. It maintains standard 8051 compatibility. 1 = The clock runs at $F_{SYS}/2$ baud rate for faster serial communication.</p> <p><u>Mode 1:</u> 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.</p> <p><u>Mode 2 or 3:</u> 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.</p>
4	REN	<p>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.</p>
3	TB8	<p>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</p>

Bit	Name	Description
2	RB8	<p>9th received bit</p> <p>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.</p>
1	TI	<p>Transmission interrupt flag</p> <p>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.</p>
0	RI	<p>Receiving interrupt flag</p> <p>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.</p>

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]							

Bit	Name	Description
7:0	SBUF[7:0]	<p>Serial port 0 data buffer</p> <p>This byte actually consists two separate registers. One is the receiving register, 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.</p> <p>The transmission is initiated through giving data to SBUF.</p>

SBUF_1 – Serial Port 1 Data Buffer

Register	SFR Address	Reset Value
SBUF_1	9AH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
SBUF_1[7:0]							
R/W							

Bit	Name	Description
7:0	SBUF_1[7:0]	<p>Serial port 1 data buffer</p> <p>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_1, it goes to the transmitting buffer and is shifted for serial transmission. When data is moved from SBUF_1, it comes from the receiving register.</p> <p>The transmission is initiated through giving data to SBUF_1.</p>

PnDW – Port n Pull-down Resister Control

Register	SFR Address	Reset Value
P0DW	9AH, page 2	0000_0000 b
P1DW	9BH, page 2	0000_0000 b
P2DW	9CH, page 2	0000_0000 b
P3DW	9DH, 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.x	Pn.x pull- down enable 0 = Pn.x pull-down Disabled. 1 = Pn.x pull- down Enabled.

EIE – Extensive Interrupt Enable

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

7	6	5	4	3	2	1	0
ET2	ESPI	EFB	EWDT	EPWM0	ECAP	EPI	EI ² C
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 = Interrupt generated by TF2 (T2CON.7) Enabled.
6	ESPI	Enable SPI interrupt 0 = SPI interrupt Disabled. 1 = Interrupt generated by SPIF (SPSR.7), SPIOVF (SPSR.5), or MODF (SPSR.4) Enable.
5	EFB	Enable Fault Brake interrupt 0 = Fault Brake interrupt Disabled. 1 = Interrupt generated by FBF (PWM0FBD.7) Enabled.
4	EWDT	Enable WDT interrupt 0 = WDT interrupt Disabled. 1 = Interrupt generated by WDTF (WDCON.5) Enabled.
3	EPWM0	Enable PWM0 interrupt 0 = PWM0 interrupt Disabled. 1 = Interrupt generated by PWMF (PWM0CON0.5) Enabled.
2	ECAP	Enable input capture interrupt 0 = Input capture interrupt Disabled. 1 = Interrupt generated by any flags of CAPF[2:0] (CAPCON0[2:0]) Enabled.
1	EPI	Enable pin interrupt 0 = Pin interrupt Disabled. 1 = Interrupt generated by any flags in PIF register Enabled.
0	EI ² C	Enable I ² C interrupt 0 = I ² C interrupt Disabled. 1 = Interrupt generated by SI (I2CON.3) or I2TOF (I2TOC.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
-	-	EPWM3	EPWM2	EPWM1	EWKT	ET3	ES_1
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
5	EPWM3	Enable PWM3 interrupt 0 = PWM3 interrupt Disabled. 1 = Interrupt generated by PWM3F (PWM3CON0.5) Enabled.
4	EPWM2	Enable PWM2 interrupt 0 = PWM2 interrupt Disabled. 1 = Interrupt generated by PWM2F (PWM2CON0.5) Enabled.
3	EPWM1	Enable PWM1 interrupt 0 = PWM1 interrupt Disabled. 1 = Interrupt generated by PWM1F (PWM1CON0.5) Enabled.
2	EWKT	Enable WKT interrupt 0 = WKT interrupt Disabled. 1 = Interrupt generated by WKTF (WKCON.4) Enabled.
1	ET3	Enable Timer 3 interrupt 0 = Timer 3 interrupt Disabled. 1 = Interrupt generated by TF3 (T3CON.4) Enabled.
0	ES_1	Enable serial port 1 interrupt 0 = Serial port 1 interrupt Disabled. 1 = Interrupt generated by TI_1 (SCON_1.1) or RI_1 (SCON_1.0) Enabled.

RSR – Reset Flag Register

Register	SFR Address	Reset Value
RSR	9DH, Page 0	00XX_XXXX b

7	6	5	4	3	2	1	0
-	-	HardF	POF	RSTPINF	BORF	WDTRF	SWRF
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
7:6	-	Reserved
5	HardF	mirrored from AUXR1.5 Clear this bit by write AUXR1.5=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 AUXR1.6 Clear this bit by write AUXR1.6=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 AUXR1.7 Clear this bit by write AUXR1.7=0 or RSR.0=0

CHPCON – Chip Control

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 fla The hardware will set this bit after IAPGO (ISPTRG.0) is set if any of the following condition is met: (1) The accessing address is oversize. (2) IAPCN commanis invalid. (3) IAP erases or programs updating un-enabled block. (4) IAP erasing or programming operates under V _{BOD} 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.
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.

AUXR1 – Auxiliary Register 1

Register	SFR Address	Reset Value
AUXR1	A2H , Page 0	POR: 0000 0000b, Software reset: 1U00 0000b, nRESET pin: U100 0000b, Others: UUUU 0000b

7	6	5	4	3	2	1	0
SWRF	RSTPINF	HardF	SLOW	GF2	UART0PX	0	DPS
R/W	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 pin, this bit will be set via hardware. It is recommended that the flag be cleared via software.
5	HardF	Hard Fault reset flag Once Program Counter (PC) is over Flash size, MCU will be 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 be disabled and only HardF flag be asserted.
4	SLOW	ADC Slow Speed Selection This bit is used to select ADC low speed. 0 = ADC convert time is high speed 1400ns, $F_{ADC} = 714$ ksp/s 1 = ADC convert time is low speed 4750ns, $F_{ADC} = 215$ ksp/s, Note1 : slow speed ADC convert time will improvement ADC convert performance when V_{DD} is under 2.5V. Note2: AUXR1.4 (SLOW) or ADCCON3 .0(SLOW) is set to 1, ADC enter Slow speed mode.
3	GF2	General purpose flag 2 The general purpose flag that can be set or cleared by the user via software.
2	UART0PX	Serial port 0 pin exchange 0 = Assign RXD to P0.7 and TXD to P0.6 by default. 1 = Exchange RXD to P0.6 and TXD to P0.7. Note that TXD and RXD 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	0	Reserved This bit is always read as 0.

Bit	Name	Description
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.

AUXR2 – Auxiliary Register 2

Register	SFR Address	Reset Value
AUXR2	A1H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
UART2TXP		UART2RXP		UART1TXP		UART1RXP	
R/W		R/W		R/W		R/W	

Bit	Name	Description
7:6	UART2TXP	UART2 TX pin select 00 = Reserved by default 01 = Assign UART2 TX to P0.3 10 = Assign UART2 TX to P3.0 11 = Reserved
5:4	UART2RXP	UART2 RX pin select 00 = Reserved by default 01 = Assign UART2 RX to P0.4 10 = Assign UART2 RX to P1.7 11 = Reserved
3:2	UART1TXP	UART1 TX pin select 00 = Reserved by default 01 = Assign UART1 TX to P1.6 10 = Assign UART1 TX to P3.6 11 = Assign UART1 TX to P1.0
1:0	UART1RXP	UART1 RX pin select 00 = Reserved by default 01 = Assign UART1 RX to P0.2 10 = Assign UART1 RX to P3.7 11 = Assign UART1 RX to P0.0

AUXR3 – Auxiliary Register 3

Register	SFR Address	Reset Value
AUXR3	A2H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
UART4TXP		UART4RXP		UART3TXP		UART3RXP	
R/W		R/W		R/W		R/W	

Bit	Name	Description
7:6	UART4TXP	UART4 TX pin select 00 = Reserved by default 01 = Assign UART4 TX to P2.3 10 = Reserved 11 = Reserved
5:4	UART4RXP	UART4 RX pin select 00 = Reserved by default 01 = Assign UART4 RX to P2.2 10 = Reserved 11 = Reserved
3:2	UART3TXP	UART3 TX pin select 00 = Reserved by default 01 = Assign UART3 TX to P1.2 10 = Assign UART3 TX to P1.5 11 = Assign UART3 TX to P0.5
1:0	UART3RXP	UART3 RX pin select 00 = Reserved by default 01 = Assign UART3 RX to P1.1 10 = Assign UART3 RX to P2.5 11 = Assign UART3 RX to P3.4

AUXR4 – Auxiliary Register 4

Register	SFR Address	Reset Value
AUXR4	A3H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PWM2CH1P		PWM2CH0P		PWM1CH1P		PWM1CH0P	
R/W		R/W		R/W		R/W	

Bit	Name	Description
7:6	PWM2CH1P	PWM2 channel 1 pin select 00 = Assign PWM2_CH1 to P3.0 01 = Assign PWM2_CH1 to P3.1 10 = Assign PWM2_CH1 to P0.0 11 = Assign PWM2_CH1 to P0.4
5:4	PWM2CH0P	PWM2 channel 0 pin select 00 = Reserved by default 01 = Assign PWM2_CH0 to P2.1 10 = Assign PWM2_CH0 to P1.0 11 = Assign PWM2_CH0 to P0.5
3:2	PWM1CH1P	PWM1 channel 1 pin select 00 = Reserved by default 01 = Assign PWM1_CH1 to P2.2 10 = Assign PWM1_CH1 to P1.4 11 = Assign PWM1_CH1 to P1.1
1:0	PWM1CH0P	PWM1 channel 0 pin select 00 = Reserved by default 01 = Assign PWM1_CH0 to P2.3 10 = Assign PWM1_CH0 to P1.2 11 = Reserved

AUXR5 – Auxiliary Register 5

Register	SFR Address	Reset Value
AUXR5	A4H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
CLOP	T0P	-		PWM3CH1P		PWM3CH0P	
R/W	R/W	-		R/W		R/W	

Bit	Name	Description
7	CLOP	CLK Output pin select 0 = Assign CLO to P1.1 while CLO output enabled 1 = Assign CLO to P3.3 while CLO output enabled
6	T0P	T0 pin select 0 = Assign T0 to P0.5 1 = Assign T0 to P2.4
5:4	-	Reserved
3:2	PWM3CH1P	PWM3 channel 1 pin select 00 = Reserved by default 01 = Assign PWM3_CH1 to P3.4 10 = Assign PWM3_CH1 to P1.5 11 = Assign PWM3_CH1 to P0.3
1:0	PWM3CH0P	PWM3 channel 0 pin select 00 = Reserved by default 01 = Assign PWM3_CH0 to P3.2 10 = Assign PWM3_CH0 to P0.1 11 = Reserved

AUXR6 – Auxiliary Register 6

Register	SFR Address	Reset Value
AUXR6	A5H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	UART4DG	UART3DG	UART2DG	UART1DG	UART0DG
-	-	-	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
4	UART4DG	UART4 RX Deglitch Control 1: Deglitch is Enabled 0: Deglitch is Disabled
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

AUXR7 – Auxiliary Register 7

Register	SFR Address	Reset Value
AUXR7	A6H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	SPI0NSSP		SPI0MOSIP	SPI0MISOP	SPI0CKP
-	-	-	R/W		R/W	R/W	R/W

Bit	Name	Description
4:3	SPI0NSSP	SPI0_SS pin select 00 = Assign SPI0_SS to P1.5 01 = Reserved 10 = Assign SPI0_SS to P3.5 11 = Reserved
2	SPI0MOSIP	SPI0_MOSI pin select 0 = Assign SPI0_MOSI to P0.0 1 = Assign SPI0_MOSI to P3.0
1	SPI0MISOP	SPI_MISO pin select 0 = Assign SPI_MISO to P0.1 1 = Assign SPI_MISO to P2.5
0	SPI0CKP	SPI_CLK pin select 0 = Assign SPI_CLK to P1.0 1 = Assign SPI_CLK to P1.7

AUXR8 – Auxiliary Register 8

Register	SFR Address	Reset Value
AUXR8	A7H, Page 2, TA protected	0000_0000 b

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

Bit	Name	Description										
7:4	CLODIV[3:0]	<p>Clock output divider</p> <p>The system clock output follows the equation below according to CLODIV value.</p> $F_{CLO} = F_{SYS}, \text{ while CLODIV} = 00H, \text{ and}$ $F_{CLO} = \frac{F_{SYS}}{2^x \text{ CLODIV}} \text{ while CLODIV} = 01H \text{ to } 0FH.$										
3-0	CKTESTOEN[3:0]	<p>Clock macro source output to CKO</p> <table border="0"> <thead> <tr> <th>CKTESTOEN</th> <th>Function</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>HIRC (24 MHz) output to CKO function pin</td> </tr> <tr> <td>0001</td> <td>OSCIN output to CKO function pin</td> </tr> <tr> <td>0010</td> <td>LIRC (10 kHz) output to CKO function pin</td> </tr> <tr> <td>0011</td> <td>HXT (4~24 MHz) output to CKO function pin</td> </tr> </tbody> </table>	CKTESTOEN	Function	0000	HIRC (24 MHz) output to CKO function pin	0001	OSCIN output to CKO function pin	0010	LIRC (10 kHz) output to CKO function pin	0011	HXT (4~24 MHz) output to CKO function pin
CKTESTOEN	Function											
0000	HIRC (24 MHz) output to CKO function pin											
0001	OSCIN output to CKO function pin											
0010	LIRC (10 kHz) output to CKO function pin											
0011	HXT (4~24 MHz) output to CKO function pin											

BODCON0 – Brown-out Detection Control 0

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[1:0] ^[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 off. 1 = Brown-out detection circuit on. Note that BOD output is not available until 2~3 LIRC clocks after enabling.
6:4	BOV[1:0]	Brown-out voltage select 11 = V _{BOD} is 2.2V. 10 = V _{BOD} is 2.7V. 01 = V _{BOD} is 3.7V. 00 = V _{BOD} 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 V _{BOD} event occurs. If both EBOD (EIE.2) and EA (IE.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 V _{BOD} . 0 = Brown-out reset when V _{DD} drops below V _{BOD} Disabled. 1 = Brown-out reset when V _{DD} drops below V _{BOD} 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 V _{BOD} while BOD circuit is enabled. It keeps 0 if BOD is not enabled. 0 = V _{DD} voltage level is higher than V _{BOD} or BOD is disabled. 1 = V _{DD} voltage level is lower than V _{BOD} . Note that this bit is read-only.
<p>Note:</p> <ol style="list-style-type: none"> BODEN, BOV[1:0], and BORST are initialized by being directly loaded from CONFIG2 bit 7, [6:4], and 2 after all resets. BOF reset value depends on different setting of CONFIG2 and V_{DD} voltage level. Please check Table 6.2-1 BOF Reset Value 		

IAPTRG – IAP Trigger

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
0	IAPGO	<p>IAP go</p> <p>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.</p> <p>Before triggering an IAP action, interrupts (if enabled) should be temporary disabled for hardware limitation.</p>

IAPUEN – IAP Updating Enable

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

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

Bit	Name	Description
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 = Interrupt generated by ADCF (ADCCON0.7) Enabled.
5	EBOD	Enable brown-out interrupt 0 = Brown-out detection interrupt Disabled. 1 = Interrupt generated by BOF (BODCON0.3) Enabled.
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.
3	ET1	Enable Timer 1 interrupt 0 = Timer 1 interrupt Disabled. 1 = Interrupt generated by TF1 (TCON.7) Enabled.
2	EX1	Enable external interrupt 1 0 = External interrupt 1 Disabled. 1 = Interrupt generated by $\overline{INT1}$ pin (P1.7) Enabled.
1	ET0	Enable Timer 0 interrupt 0 = Timer 0 interrupt Disabled. 1 = Interrupt generated by TF0 (TCON.5) Enabled.
0	EX0	Enable external interrupt 0 0 = External interrupt 0 Disabled. 1 = Interrupt generated by $\overline{INT0}$ pin (P3.0) Enabled.

SADDR – Slave 0 Address

Register	SFR Address	Reset Value
SADDR	A9H, Page 0	0000_0000 b

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

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

WDCON – Watchdog Timer Control

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 WDTE[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. <u>Writing:</u> 0 = No effect. 1 = Clearing WDT counter. <u>Reading:</u> 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 WDTE[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 MCU is reset by WDT time-out event, this bit will be set via hardware. It is recommended that the flag be cleared via software.
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.

Note:

1. WDTRF will be cleared after power-on reset, be set after WDT reset, and remains unchanged after any other resets.
2. WDPS[2:0] are all set after power-on reset and keep unchanged after any reset other than power-on reset.

BODCON1 – Brown-out Detection Control 1

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.)

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 Table 6.3-1 IAP Modes and Command Codes .
5	FOEN	
4	FCEN	
3:0	FCTRL[3:0]	

PnM1 – Port n Mode Select 1

Register	SFR Address	Reset Value
P0M1	B1H, Page 0	1111_1111 b
P1M1	B3H, Page 0	1111_1111 b
P2M1	89H, Page 2	0011_1111 b
P3M1	ACH, Page 0	1111_1111 b

7	6	5	4	3	2	1	0
PnM1.7	PnM1.6	PnM1.5	PnM1.4	PnM1.3	PnM1.2	PnM1.1	PnM1.0
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 0 mode select 1

PnM2 – Port Mode Select 2

Register	SFR Address	Reset Value
P0M2	B2H, Page 0	0000_0000 b
P1M2	B4H, Page 0	0000_0000 b
P2M2	8AH, Page 2	0000_0000 b
P3M2	ADH, Page 0	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: PxM1 and PxM2 are used in combination to determine the I/O mode of each pin of Port. See Table 6.4-1 Configuration for Different I/O Modes .

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	B1H, Page 1	0000_0000 b
P1S	B3H, Page 1	0000_0000 b
P2S	8CH, Page 2	0000_0000 b
P3S	ACH, 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.

TOE – Timer01 Output Enable

Register	SFR Address	Reset Value
TOE	B5H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	T1OE	T0OE	-	-
-	-	-	-	R/W	R/W	-	-

Bit	Name	Description
3	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.
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.

IPH – Interrupt Priority High^[2]

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
6	PADC	ADC interrupt priority high bit
5	PBOD	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

Note:IPH is used in combination with the IP respectively to determine the priority of each interrupt source. See [Table 6.2-4 Interrupt Priority Level Setting](#) for correct interrupt priority configuration.

PWMnINTC – PWM Interrupt Control

Register	SFR Address	Reset Value
PWM0INTC	B7H, page 1	0000_0000 b
PWM1INTC	B6H, 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	INTSELO
-	-	R/W	R/W	-	R/W	R/W	R/W

Bit	Name	Description
5:4	INTTYP[1:0]	<p>PWM interrupt type select</p> <p>These bit select PWM interrupt type.</p> <p>00 = Falling edge on PWM channel 0/1/2/3/4/5 pin.</p> <p>01 = Rising edge on PWM channel 0/1/2/3/4/5 pin.</p> <p>10 = Central point of a PWM period.</p> <p>11 = End point of a PWM period.</p> <p>Note that the central point interrupt or the end point interrupt is only available while PWM operates in center-aligned type.</p>
2:0	INTSEL[2:0]	<p>PWM interrupt pair select</p> <p>These bits select which PWM channel asserts PWM interrupt when PWM interrupt type is selected as falling or rising edge on PWM channel 0~5 pin..</p> <p>000 = PWM_CH0.</p> <p>001 = PWM_CH1.</p> <p>010 = PWM_CH2.</p> <p>011 = PWM_CH3.</p> <p>100 = PWM_CH4.</p> <p>101 = PWM_CH5.</p> <p>Others = PWM_CH0.</p>

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
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: IP is used in combination with the IPH to determine the priority of each interrupt source. See [Table 6.2-4 Interrupt Priority Level Setting](#) for correct interrupt priority configuration.

SADEN – Slave 0 Address Mask

Register	SFR Address	Reset Value
SADEN	B9H, Page 0	0000_0000 b

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

Bit	Name	Description
7:0	SADEN[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.

SADEN 1 – Slave 1 Address Mask

Register	SFR Address	Reset Value
SADEN_1	BAH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
SADEN_1[7:0]							
R/W							

Bit	Name	Description
7:0	SADEN_1[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.

SADDR_1 – Slave 1 Address

Register	SFR Address	Reset Value
SADDR_1	BBH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
SADDR_1[7:0]							
R/W							

Bit	Name	Description
7:0	SADDR_1[7:0]	Slave 1 address This byte specifies the microcontroller's own slave address for UART1 multi-processor communication.

I2DAT – I²C Data

Register	SFR Address	Reset Value
I2DAT	BCH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
I2DAT[7:0]							
R/W							

Bit	Name	Description
7:0	I2DAT[7:0]	<p>I²C data</p> <p>I2DAT contains a byte of the I²C data to be transmitted or a byte, which has just received. Data in I2DAT remains as long as SI is logic 1. The result of reading or writing I2DAT during I²C transceiving progress is unpredicted.</p> <p>While data in I2DAT is shifted out, data on the bus is simultaneously being shifted in to update I2DAT. I2DAT always shows the last byte that presented on the I²C bus. Thus the event of lost arbitration, the original value of I2DAT changes after the transaction.</p>

I2STAT – I²C Status

Register	SFR Address	Reset Value
I2STAT	BDH, Page 0	1111 1000 b

7	6	5	4	3	2	1	0
I2STAT[7:3]					0	0	0
R					R	R	R

Bit	Name	Description
7:3	I2STAT[7:3]	I ² C status code The MSB five bits of I2STAT contains the status code. There are 27 possible status codes. When I2STAT is F8H, no relevant state information is available and SI flag keeps 0. All other 26 status codes correspond to the I ² C 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 I2STAT are always read as 0.

I2CLK – I²C Clock

Register	SFR Address	Reset Value
I2CLK	BEH, Page 0	0000_1001 b

7	6	5	4	3	2	1	0
I2CLK[7:0]							
R/W							

Bit	Name	Description
7:0	I2CLK[7:0]	<p>I²C clock setting</p> <p><u>In master mode:</u></p> <p>This register determines the clock rate of I²C bus when the device is in a master mode. The clock rate follows the equation,</p> $\frac{F_{SYS}}{4 \times (I2CLK + 1)}$ <p>The default value will make the clock rate of I²C bus 400k bps if the peripheral clock is 16 MHz. Note that the I2CLK value of 00H and 01H are not valid. This is an implement limitation.</p> <p><u>In slave mode:</u></p> <p>This byte has no effect. In slave mode, the I²C device will automatically synchronize with any given clock rate up to 400k bps.</p>

I2TOC – I²C Time-out Counter

Register	SFR Address	Reset Value
I2TOC	BFH, 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
2	I2TOCEN	I ² C time-out counter enable 0 = I ² C time-out counter Disabled. 1 = I ² C time-out counter Enabled.
1	DIV	I ² C time-out counter clock divider 0 = The clock of I ² C time-out counter is F _{sys} /1. 1 = The clock of I ² C time-out counter is F _{sys} /4.
0	I2TOF	I ² C time-out flag This flag is set by hardware if 14-bit I ² C time-out counter overflows. It is cleared by software.

I2CON – I²C Control

Register	SFR Address	Reset Value
I2CON	C0H, All pages, Bit addressable	0000_0000 b

7	6	5	4	3	2	1	0
-	I2CEN	STA	STO	SI	AA	-	I2CPX
-	R/W	R/W	R/W	R/W	R/W	-	R/W

Bit	Name	Description
6	I2CEN	I ² C bus enable 0 = I ² C bus Disabled. 1 = I ² C bus Enabled. Before enabling the I ² C, I2C0_SCL and I2C0_SDA port latches should be set to logic 1.
5	STA	START flag When STA is set, the I ² C generates a START condition if the bus is free. If the bus is busy, the I ² C waits for a STOP condition and generates a START condition following. If STA is set while the I ² C is already in the master mode and one or more bytes have been transmitted or received, the I ² C 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 I ² C 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 I ² C device from the bus error state (I2STAT as 00H). In this case, no STOP condition is transmitted to the I ² C bus. If the STA and STO bits are both set and the device is original in the master mode, the I ² C 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 I ² C frames.
3	SI	I ² C interrupt flag SI flag is set by hardware when one of 26 possible I ² C status (besides F8H status) is entered. After SI is set, the software should read I2STAT 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 I2C0_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, I ² C 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.

Bit	Name	Description
2	AA	<p>Acknowledge assert flag</p> <p>If the AA flag is set, an ACK (low level on I2C0_SDA) will be returned during the acknowledge clock pulse of the I2C0_SCL line while the I²C device is a receiver or an own-address-matching slave.</p> <p>If the AA flag is cleared, a NACK (high level on I2C0_SDA) will be returned during the acknowledge clock pulse of the I2C0_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 slave address and the General Call. Consequently, SI will not be asserted and no interrupt is requested.</p> <p>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.</p> <p>There is a special case of I2STAT 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.</p>
0	I2CPX	<p>I²C pins select</p> <p>0 = Assign I2C0_SCL to P1.3 and I2C0_SDA to P1.4.</p> <p>1 = Assign I2C0_SCL to P0.2 and I2C0_SDA to P1.6.</p> <p>Note that I²C pins will exchange immediately once setting or clearing this bit.</p>

I2ADDR – I²C Own Slave Address

Register	SFR Address	Reset Value
I2ADDR	C1H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
I2ADDR[7:1]							GC
R/W							R/W

Bit	Name	Description
7:1	I2ADDR[7:1]	<p>I²C device's own slave address</p> <p><u>In master mode:</u> These bits have no effect.</p> <p><u>In slave mode:</u> 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.</p> <p>Note that I2ADDR[7:1] should not remain its default value of all 0, because address 0x00 is reserved for General Call.</p>
6	GC	<p>General Call bit</p> <p><u>In master mode:</u> This bit has no effect.</p> <p><u>In slave mode:</u> 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.</p>

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
3:0	ADCR[3:0]	ADC result low byte The least significant 4 bits of the ADC result stored in this register.

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 6.9-2 Serial Port UART1 Mode / baud rate Description for details.
6	SMOD0_1	Serial port 1 framing error access enable 0 = SCON_1.7 accesses to SM0_1 bit. 1 = SCON_1.7 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[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.

PIOCON1 – PWM or I/O Select

Register	SFR Address	Reset Value
PIOCON1	C6H, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
PIO17	-	PIO15	-	PIO04	PIO05	PIO14	-
R/W	-	R/W	-	R/W	R/W	R/W	-

Bit	Name	Description
7	PIO17	P1.7/PWM pin function select 0 = P1.7/PWM3_CH0 pin functions as GPIO P1.7. 1 = P1.7/PWM3_CH0 pin functions as PWM3_CH0 output.
6	-	Reserved
5	PIO15	P1.5/PWM pin function select 0 = P1.5/PWM pin functions as P1.5. 1 = P1.5/PWM pin functions as PWM0_CH5 / PWM3_CH1 output. When AUXR5[3:2].PWM3CH1P = 00, PIO15 = 1, P1.5 defined as PWM0_CH5. AUXR5[3:2].PWM3CH1P = 11, PIO15 = 1, P1.5 defined as PWM3_CH1.
4	-	Reserved
3	PIO04	P0.4/PWM pin function select 0 = P0.4/PWM pin functions as P0.4. 1 = P0.4/PWM pin functions as PWM0_CH3 / PWM2_CH1 output. When AUXR4[7:6].PWM2CH1P = 00, PIO04 = 1, P0.4 defined as PWM0_CH3. AUXR4[7:6].PWM2CH1P = 11, PIO04 = 1, P0.4 defined as PWM2_CH1.
2	PIO05	P0.5/PWM pin function select 0 = P0.5/PWM pin functions as P0.5. 1 = P0.5/PWM pin functions as PWM0_CH2 / PWM2_CH0 output. When AUXR4[5:4].PWM2CH1P = 00, PIO05 = 1, P0.5 defined as PWM0_CH2. AUXR4[5:4].PWM2CH1P = 11, PIO05 = 1, P0.5 defined as PWM2_CH0.
1	PIO14	P1.4/PWM pin function select 0 = P1.4/PWM pin functions as P1.4. 1 = P1.4/PWM pin functions as PWM0_CH1 / PWM1_CH1 output. When AUXR4[3:2]. PWM1CH1P = 00, PIO14 = 1, P1.4 defined as PWM0_CH1. AUXR4[3:2]. PWM1CH1P = 10, PIO14 = 1, P1.4 defined as PWM1_CH1.
0	-	Reserved

PIOCON2 – PWM or I/O Select

Register	SFR Address	Reset Value
PIOCON2	B7H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PIO34	PIO33	PIO32	PIO31	PIO30	PIO23	PIO22	PIO21
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
7	PIO34	P3.4/PWM pin function select 0 = P3.4/PWM pin functions as P3.4. 1 = P3.4/PWM pin functions as PWM3_CH1 output.
6	PIO33	P3.3/PWM pin function select 0 = P3.3/PWM pin functions as P3.3. 1 = P3.3/PWM pin functions as PWM0_CH0 output.
5	PIO32	P3.2/PWM pin function select 0 = P3.2/PWM pin functions as P3.2. 1 = P3.2/PWM pin functions as PWM3_CH0 output.
4	PIO31	P3.1/PWM pin function select 0 = P3.1/PWM pin functions as P3.1. 1 = P3.1/PWM pin functions as PWM2_CH1 output.
3	PIO30	P3.0/PWM pin function select 0 = P3.0/PWM pin functions as P3.0 1 = P3.0/PWM pin functions as PWM2_CH1 output.
2	PIO23	P23/PWM pin function select 0 = P2.3/PWM pin functions as P2.3. 1 = P2.3/PWM pin functions as PWM1_CH0 output.
1	PIO22	P2.2/PWM pin function select 0 = P2.2/PWM pin functions as P2.2. 1 = P2.2/PWM pin functions as PWM1_CH1 output.
0	PIO21	P2.1/PWM pin function select 0 = P2.1/PWM pin functions as P2.1. 1 = P2.1/PWM pin functions as PWM2_CH0 output.

TA – Timed Access

Register	SFR Address	Reset Value
TA	C7H, Page 0	0000_0000 b

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

Bit	Name	Description
7:0	TA[7:0]	<p>Timed access</p> <p>The timed access register controls the access to protected SFRs. 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 SFRs.</p>

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.
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.
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.

RCMP2L – Timer 2 Reload/Compare Low Byte

Register	SFR Address	Reset Value
RCMP2L	CAH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
RCMP2L[7:0]							
R/W							

Bit	Name	Description
7:0	RCMP2L[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.

RCMP2H – Timer 2 Reload/Compare High Byte

Register	SFR Address	Reset Value
RCMP2H	CBH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
RCMP2H[7:0]							
R/W							

Bit	Name	Description
7:0	RCMP2H[7:0]	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
TL2[7:0]							
R/W							

Bit	Name	Description
7:0	TL2[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
TH2[7:0]							
R/W							

Bit	Name	Description
7:0	TH2[7:0]	Timer 2 high byte The TH2 register is the high byte of the 16-bit counting register of Timer 2.

ADCMP[3:0] – ADC Compare Low Byte

Register	SFR Address	Reset Value
ADCMP[3:0]	CEH, Page 0	0000_0000 b

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

Bit	Name	Description
3:0	ADCMP[3:0]	ADC compare low byte The least significant 4 bits of the ADC compare value stores in this register.

EIP2 – Extensive Interrupt Priority 2

Register	SFR Address	Reset Value
EIP2	CEH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	PUART4	PUART3	PUART2
-	-	-	-	-	R/W	R/W	R/W

Bit	Name	Description
2	PUART4	UART4 interrupt priority low bit
1	PUART3	UART3 interrupt priority low bit
0	PUART2	UART2 interrupt priority low bit

Note: EIP2 is used in combination with the EIPH2 to determine the priority of each interrupt source. See Table 6.2-4 Interrupt Priority Level Setting for correct interrupt priority configuration.

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.

EIPH2 – Extensive Interrupt Priority High 2

Register	SFR Address	Reset Value
EIPH2	CFH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	PSC2H	PSC1H	PSC0H
-	-	-	-	-	R/W	R/W	R/W

Bit	Name	Description
2	PSC2H	SC2 / UART4 interrupt priority high bit
1	PSC1H	SC1 / UART3 interrupt priority high bit
0	PSC0H	SC0 / UART2 interrupt priority high bit

Note: EIPH2 is used in combination with the EIP2 to determine the priority of each interrupt source. See Table 6.2-4 Interrupt Priority Level Setting for correct interrupt priority configuration.

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	C	Auxiliary carry Set when the previous operation resulted in a carry-out from or a borrow-in to the 4 th bit of the low order nibble, otherwise cleared.																				
5	0	User flag 0 The general purpose flag that can be set or cleared by user.																				
4	RS1	Register bank selection bits																				
3	RS0	These two bits select one of four banks in which R0 to R7 locate. <table border="1" style="margin-left: 40px;"> <thead> <tr> <th>RS1</th> <th>RS0</th> <th>Register Bank</th> <th>RAM Address</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>00H to 07H</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>08H to 0FH</td> </tr> <tr> <td>1</td> <td>0</td> <td>2</td> <td>10H to 17H</td> </tr> <tr> <td>1</td> <td>1</td> <td>3</td> <td>18H to 1FH</td> </tr> </tbody> </table>	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
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																			
2	V	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.																				
1	1	User flag 1 The general purpose flag that can be set or cleared by user via software.																				

Bit	Name	Description
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.

PWMnPH – PWM Period High Byte, n = 0,1,2,3

Register	SFR Address	Reset Value
PWM0PH	D1H, Page 0	0000_0000 b
PWM1PH	A9H, 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
PWM0P[15:8]							
R/W							

Bit	Name	Description
7:0	PWM0P[15:8]	PWM period high byte This byte with PWM0PL controls the period of the PWM generator signal.

PNP – PWM Negative Polarity

Register	SFR Address	Reset Value
PNP	D6H, Page 0	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
n	PNPn	PWM0_CHn negative polarity output enable 0 = PWM0_CHn signal outputs directly on PWM0_CHn pin. 1 = PWM0_CHn signal outputs inversely on PWM0_CHn pin.

PWM0FBD – PWM Fault Brake Data

Register	SFR Address	Reset Value
PWM0FBD	D7H, Page 0	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.
N	FBDn	PWM0_CHn Fault Brake data 0 = PWM0_CHn signal is overwritten by 0 once Fault Brake asserted. 1 = PWM0_CHn signal is overwritten by 1 once Fault Brake asserted.

XTLCON – XLT Clock Control (TA Protected)

Register	SFR Address	Reset Value
XTLCON	D7H, Page 2	0111_0111 b

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

Bit	Name	Description
7	-	Reserved
6:4	HXSG	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)
3:0	-	Reserved

SCnDR – SC Data Register

Register	SFR Address	Reset Value
SC0DR	D9H, Page 2	0000_0000 b
SC1DR	E1H, Page 2	0000_0000 b
SC2DR	E9H, 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]	<p>SC / UART buffer data</p> <p>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.</p> <p>Note: If SCEN(SCnCR0[0]) is not enabled, SCnDR cannot be programmed.</p>

SCnEGT – SC Extra Guard Time Register

Register	SFR Address	Reset Value
SC0EGT	DAH, Page 2	0000_0000 b
SC1EGT	E2H, Page 2	0000_0000 b
SC2EGT	EAH, 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 2	0111_0011 b
SC1ETURD0	E3H, Page 2	0111_0011 b
SC2ETURD0	EBH, 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 2	0011_0001 b
SC1ETURD1	E4H, Page 2	0011_0001 b
SC2ETURD1	ECH, 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]	<p>SC clock divider</p> <p>000 = F_{SC} is $F_{SYS}/1$.</p> <p>001 = F_{SC} is $F_{SYS}/2$.</p> <p>010 = F_{SC} is $F_{SYS}/4$.</p> <p>011 = F_{SC} is $F_{SYS}/8$. (By default.)</p> <p>100 = F_{SC} is $F_{SYS}/16$.</p> <p>101 = F_{SC} is $F_{SYS}/16$.</p> <p>110 = F_{SC} is $F_{SYS}/16$.</p> <p>111 = F_{SC} is $F_{SYS}/16$.</p> <p>Note: that the F_{SC} clock should be 1Mhz ~ 5Mhz for ISO/IEC 7816-3 standard</p>
3:0	ETURDIV [11:8]	<p>MSB bits of ETU Rate Divider</p> <p>The field indicates the MSB of clock rate divider.</p> <p>The real ETU is $ETURDIV[11:0] + 1$.</p> <p>Note 1: $ETURDIV[11:0]$ must be greater than 0x004.</p> <p>Note 2: SCnETURD0 has to program first, then SCnETUDR1 .</p>

ScnIE – SC Interrupt Enable Control Register

Register	SFR Address	Reset Value
SC0IE	DDH, Page 2	0000_0000 b
SC1IE	E5H, Page 2	0000_0000 b
SC2IE	EDH, 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
7:4	-	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 2	0000_0010 b
SC1IS	E6H, Page 2	0000_0010 b
SC2IS	EEH, Page 2	0000_0010 b

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

Bit	Name	Description
7:5	-	Reserved
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 2	0000_1010 b
SC1TSR	E7H, Page 2	0000_1010 b
SC2TSR	EFH, 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	TXOVF	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.

PWMnCON0 – PWMn Control 0, n = 1,2,3

Register	SFR Address	Reset Value
PWM1CON0	B4H, 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
PWMRUN	LOAD	PWMF	CLRPWM	-	-	-	-
R/W	R/W	R/W	R/W	-	-	-	-

Bit	Name	Description
7	PWMRUN	PWM 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 control registers 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. <u>Writing:</u> 0 = No effect. 1 = Load new period and duty in their buffers while a PWM period is completed. <u>Reading:</u> 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 PWM0INTC. 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. <u>Writing:</u> 0 = No effect. 1 = Clearing PWM 16-bit counter. <u>Reading:</u> 0 = PWM 16-bit counter is completely cleared. 1 = PWM 16-bit counter is not yet cleared.

PWM0CON0 – PWM0 Control 0 (Bit-addressable)

Register	SFR Address	Reset Value
PWM0CON0	D8H, Page0	0000_0000 b

7	6	5	4	3	2	1	0
PWM0RUN	LOAD	PWMF	CLRPWM	PWM3RUN	PWM2RUN	PWM1RUN	P33FBINEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
7	PWM0RUN	<p>PWM0 run enable</p> <p>0 = PWM0 stays in idle.</p> <p>1 = PWM0 starts running.</p> <p>Note: This bit is only for PWM0CON0.</p>
6	LOAD	<p>PWM new period and duty load</p> <p>This bit is used to load period and duty control registers 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.</p> <p><u>Writing:</u></p> <p>0 = No effect.</p> <p>1 = Load new period and duty in their buffers while a PWM period is completed.</p> <p><u>Reading:</u></p> <p>0 = A loading of new period and duty is finished.</p> <p>1 = A loading of new period and duty is not yet finished.</p>
5	PWMF	<p>PWM flag</p> <p>This flag is set according to definitions of INTSEL[2:0] and INTTYP[1:0] in PWM0INTC. This bit is cleared by software.</p>
4	CLRPWM	<p>Clear PWM counter</p> <p>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.</p> <p><u>Writing:</u></p> <p>0 = No effect.</p> <p>1 = Clearing PWM 16-bit counter.</p> <p><u>Reading:</u></p> <p>0 = PWM 16-bit counter is completely cleared.</p> <p>1 = PWM 16-bit counter is not yet cleared.</p>
3	PWM3RUN	<p>PWM3 run enable</p> <p>0 = PWM3 stays in idle.</p> <p>1 = PWM3 starts running.</p> <p>Note: This bit is only for PWM0CON0.</p>

Bit	Name	Description
2	PWM2RUN	<p>PWM2 run enable</p> <p>0 = PWM2 stays in idle.</p> <p>1 = PWM2 starts running.</p> <p>Note: This bit is only for PWM0CON0.</p>
1	PWM1RUN	<p>PWM1 run enable</p> <p>0 = PWM1 stays in idle.</p> <p>1 = PWM1 starts running.</p> <p>Note: This bit is only for PWM0CON0.</p>
0	P33FBINEN	<p>P33 FB pin input enable</p> <p>0 = PWM0 output Fault Braked by P33 FB pin input Disabled.</p> <p>1 = PWM0 output Fault Braked by P33 FB pin input Enabled. Once an edge, which matches FBINLS (FBD.6) selection, occurs on FB pin, PWM0C0~C5 output Fault Brake data in FBD register. PWMRUN (PWM0CON0.7) will also be automatically cleared by hardware. The PWM output resumes when PWMRUN is set again.</p> <p>Note: This bit is only for PWM0CON0.</p>

PWMnPL – PWM Period Low Byte, n = 0,1,2,3

Register	SFR Address	Reset Value
PWM0PL	D9H, Page 0	0000_0000 b
PWM1PL	B1H, 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]	PWM period low byte This byte with PWMnPH controls the period of the PWM generator signal.

PWMnCnH – PWMn Channel x Duty High Byte, n = 0,1,2,3; x = 0,1,2,3,4,5

Register	SFR Address	Description	Reset Value
PWM0C0H	D2H, Page 0	PWM0 Channel 0 Duty High Byte	0000_0000 b
PWM0C1H	D3H, Page 0	PWM0 Channel 1 Duty High Byte	0000_0000 b
PWM0C2H	D4H, Page 0	PWM0 Channel 2 Duty High Byte	0000_0000 b
PWM0C3H	D5H, Page 0	PWM0 Channel 3 Duty High Byte	0000_0000 b
PWM0C4H	C4H, Page 1	PWM0 Channel 4 Duty High Byte	0000_0000 b
PWM0C5H	C5H, Page 1	PWM0 Channel 5 Duty High Byte	0000_0000 b
PWM1C0H	AAH, Page 2	PWM1 Channel 0 Duty High Byte	0000_0000 b
PWM1C1H	ABH, Page 2	PWM1 Channel 1 Duty High Byte	0000_0000 b
PWM2C0H	BAH, Page 2	PWM2 Channel 0 Duty High Byte	0000_0000 b
PWM2C1H	BBH, Page 2	PWM2 Channel 1 Duty High Byte	0000_0000 b
PWM3C0H	CAH, Page 2	PWM3 Channel 0 Duty High Byte	0000_0000 b
PWM3C1H	CBH, Page 2	PWM3 Channel 1 Duty High Byte	0000_0000 b

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

Bit	Name	Description
7:0	PWMnCx [15:8]	PWMnCx duty high byte This byte with PWMnCxL controls the duty of the output signal PGx from PWM generator.

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

Register	SFR Address	Description	Reset Value
PWM0C0L	DAH, Page 0	PWM0 Channel 0 Duty Low Byte	0000_0000 b
PWM0C1L	DBH, Page 0	PWM0 Channel 1 Duty Low Byte	0000_0000 b
PWM0C2L	DCH, Page 0	PWM0 Channel 2 Duty Low Byte	0000_0000 b
PWM0C3L	DDH, Page 0	PWM0 Channel 3 Duty Low Byte	0000_0000 b
PWM0C4L	CCH, Page 1	PWM0 Channel 4 Duty Low Byte	0000_0000 b
PWM0_CH5L	CDH, Page 1	PWM0 Channel 5 Duty Low Byte	0000_0000 b
PWM1_CH0L	B2H, Page 2	PWM1 Channel 0 Duty Low Byte	0000_0000 b
PWM1_CH1L	B3H, Page 2	PWM1 Channel 1 Duty Low Byte	0000_0000 b
PWM2_CH0L	C2H, Page 2	PWM2 Channel 0 Duty Low Byte	0000_0000 b
PWM2_CH1L	C3H, Page 2	PWM2 Channel 1 Duty Low Byte	0000_0000 b
PWM3_CH0L	D2H, Page 2	PWM3 Channel 0 Duty Low Byte	0000_0000 b
PWM3_CH1L	D3H, Page 2	PWM3 Channel 1 Duty Low Byte	0000_0000 b

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

Bit	Name	Description
7:0	PWMnCx [7:0]	PWMnCx duty Low byte This byte with PWMnCxH controls the duty of the output signal PGx from PWM generator.

PIOCON0 – PWM or I/O Select

Register	SFR Address	Reset Value
PIOCON0	DEH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	PIO03	PIO01	PIO00	PIO10	PIO11	PIO12
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
7:6	-	Reserved.
5	PIO03	P0.3/PWM pin function select 0 = P0.3/PWM pin functions as P0.3. 1 = P0.3/PWM pin functions as PWM0_CH5 / PWM3_CH1 output. When AUXR5[7:6]. PWM3CH1P = 00, PIO03 = 1, P0.3 defined as PWM0_CH5. AUXR5[7:6]. PWM3CH1P = 11, PIO03 = 1, P0.3 defined as PWM3_CH1.
4	PIO01	P0.1/PWM pin function select 0 = P0.1/PWM pin functions as P0.1. 1 = P0.1/PWM pin functions as PWM0_CH4 / PWM3_CH0 output. When AUXR5[5:4]. PWM3CH0P = 00, PIO01 = 1, P0.3 defined as PWM0_CH4. AUXR5[5:4]. PWM3CH0P = 10, PIO01 = 1, P0.3 defined as PWM3_CH0.
3	PIO00	P0.0/PWM pin function select 0 = P0.0/PWM pin functions as P0.0. 1 = P0.0/PWM pin functions as PWM0_CH3 / PWM2_CH1 output. When AUXR4[7:6]. PWM2CH1P = 00, PIO00 = 1, P0.0 defined as PWM0_CH3. AUXR4[7:6]. PWM2CH1P = 10, PIO00 = 1, P0.0 defined as PWM2_CH1.
2	PIO10	P1.0/PWM pin function select 0 = P1.0/PWM pin functions as P1.0. 1 = P1.0/PWM pin functions as PWM0_CH2 / PWM2_CH0 output. When AUXR4[5:4]. PWM2CH0P = 00, PIO10 = 1, P1.0 defined as PWM0_CH2. AUXR4[5:4]. PWM2CH0P = 10, PIO10 = 1, P1.0 defined as PWM2_CH0.
1	PIO11	P1.1/PWM pin function select 0 = P1.1/PWM pin functions as P1.1. 1 = P1.1/PWM pin functions as PWM0_CH1 / PWM1_CH1 output. When AUXR4[3:2]. PWM1CH1P = 00, PIO11 = 1, P1.1 defined as PWM0_CH1. AUXR4[3:2]. PWM1CH1P = 11, PIO11 = 1, P1.1 defined as PWM1_CH1.
0	PIO12	P1.2/PWM pin function select 0 = P1.2/PWM pin functions as P1.2. 1 = P1.2/PWM pin functions as PWM0_CH0 / PWM1_CH0 output. When AUXR4[1:0]. PWM1CH0P = 00, PIO12 = 1, P1.2 defined as PWM0_CH1. AUXR4[1:0]. PWM1CH0P = 10, PIO12 = 1, P1.2 defined as PWM1_CH0.

PWMnCON1 – PWM Control 1

Register	SFR Address	Reset Value
PWM0CON1	DFH, Page 0	0000_0000 b
PWM1CON1	B5H, 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 control registers. 0 = Group mode Disabled. 1 = Group mode Enabled. Note: This bit is only valid in PWM0
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, PWM0 channel 0–5 output Fault Brake data in PWM0FBD register. PWM0RUN (PWM0CON0.7) will also be automatically cleared by hardware. The PWM0 output resumes when PWM0RUN is set again. Note: This bit is only valid 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

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
OCEN	STADCPX	ADCDIV[1:0]		ETGTYP[1:0]		ADCEX	ADCEN
R/W	R/W	R/W		R/W		R/W	R/W

Bit	Name	Description
7	OCEN	ADC Offset Calibration Enable register This field is used to enable offset calibration function. 0: ADC Offset Calibration is enabled, auto-calibration by ADC hardware. 1: ADC Offset Calibration is disabled
6	STADCPX	External start ADC trigger pin select 0 = Assign STADC to P0.4. 1 = Assign STADC to P1.3. Note that STADC will exchange immediately once setting or clearing this bit.
5:4	ADCDIV[1:0]	ADC clock divider 00 = ADC clock source = FSYS/1. 01 = ADC clock source = FSYS/2. 10 = ADC clock source = FSYS/4. 11 = ADC clock source = FSYS/8.
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 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	ADCM PEN	ADCMPO	ADCAQT1[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. PWM0RUN (PWM0CON0.7) will also be automatically cleared by hardware. The PWM output resumes when PWM0RUN 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	ADCM PEN	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	ADCAQT1	ADC acquisition time This 3-bit field decides the acquisition time for ADC AIN0~AIN7 and BG sampling, following by equation below: $\text{ADC sampling time} = \frac{4 * \text{ADCAQT} + 6}{F_{\text{ADCAQT}}}$, F_{ADCAQT} is defined in ADCDIV (ADCCON2[3:1]). As default $F_{\text{ADCAQT}} = F_{\text{SYS}}$ (ADCDIV=0), Since the minima sampling time must over 370ns that means when $F_{\text{sys}} = 24\text{MHz}$, ADCAQT should be defined as 1 at least. The default and minimum acquisition time is 6 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.

ADCCON3 – ADC Control 3

Register	SFR Address	Reset Value
ADCCON3	86H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	HIE	CONT	ADCAQT2[2:0]			SLOW
-	-	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
3:1	ADCAQT2	ADC acquisition time 1 This 3-bit field decides the acquisition sampling time for ADC AIN9–AIN15 sampling, following by equation below: $\text{ADC sampling time} = \frac{4 * \text{ADCAQT} + 6}{F_{\text{ADCAQT}}}$, F_{ADCAQT} is defined in ADCDIV (ADCCON2[3:1]). As default $F_{\text{ADCAQT}} = F_{\text{SYS}}$ (ADCDIV=0), Since the minima sampling time must over 370ns that means when $F_{\text{sys}} = 24\text{MHz}$, ADCAQT should be defined as 1 at least. The minima sampling time must over 370ns, when $F_{\text{sys}} = 24\text{MHz}$, ADCAQT should be defined as 1 at least. The default and minimum acquisition time is 6 ADC clock cycles. Note that this field should not be changed when ADC is in converting.
0	SLOW	ADC Slow Speed Selection This bit is used to select ADC low speed. 0 = ADC convert time is high speed 1400ns, $F_{\text{ADC}} = 714 \text{ kpsps}$ 1 = ADC convert time is low speed 4750ns, $F_{\text{ADC}} = 215 \text{ kpsps}$, Note1 : slow speed ADC convert time will improvement ADC convert performance when VDD is under 2.5V. Note2: AUXR1.4 (SLOW) or ADCCON3 .0(SLOW) is set to 1, ADC enter Slow speed mode.

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]	<p>ADC external trigger delay counter low byte</p> <p>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.</p> $\text{External trigger delay time} = \frac{\text{ADCDLY}}{F_{\text{ADC}}}$ <p>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.</p>

CnL – Capture Low Byte, n = 0,1,2

Register	SFR Address	Reset Value
C0L	E4H, Page 0	0000_0000 b
C1L	E6H, Page 0	0000_0000 b
C2L	EDH, Page 0	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 0	0000_0000 b
C1H	E7H, Page 0	0000_0000 b
C2H	EEH, Page 0	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.

ADCCON0 – ADC Control 0 (Bit-addressable)

Register	SFR Address	Reset Value
ADCCON0	E8H, all page	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. 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. <u>Writing:</u> 0 = No effect. 1 = Start an A/D converting. <u>Reading:</u> 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 = PWM0. 01 = PWM2. 10 = PWM4 11 = STADC pin.

Bit	Name	Description
3:0	ADCHS[3:0]	<p>A/D converting channel select</p> <p>This field selects the activating analog input source of ADC. If ADCEN is 0, all inputs are disconnected.</p> <p>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). 1001 = ADC_CH9. 1010 = ADC_CH10. 1011 = ADC_CH11. 1100 = ADC_CH12. 1101 = ADC_CH13. 1110 = ADC_CH14. 1111 = ADC_CH15.</p>

PICON – Pin Interrupt Control

Register	SFR Address	Reset Value
PICON	E9H, PAGE 0,	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.

PINEN – Pin Interrupt Negative Polarity Enable.

Register	SFR Address	Reset Value
PINEN	EAH, Page 0	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
n	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 0	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
n	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	ECH, 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
n	PIFn	<p>Pin interrupt channel n flag</p> <p>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.</p> <p>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.</p>

EIP – Extensive Interrupt Priority^[3]

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

7	6	5	4	3	2	1	0
PT2	PSPI	PFB	PWDT	PPWM	PCAP	PPI	PI ² C
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	PSPI	SPI interrupt priority low bit
5	PFB	Fault Brake interrupt priority low bit
4	PWDT	WDT interrupt priority low bit
3	PPWM	PWM interrupt priority low bit
2	PCAP	Input capture interrupt priority low bit
1	PPI	Pin interrupt priority low bit
0	PI ² C	I ² C interrupt priority low bit

Note: EIP is used in combination with the EIPH to determine the priority of each interrupt source. See Table 6.2-4 Interrupt Priority Level Setting for correct interrupt priority configuration.

LDOTRIM – LDO Trim

Register	SFR Address	Reset Value
LDOTRIM	EFH, Page 1	0000_0000 b

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

Bit	Name	Description
7:5	Uldotrim[2:0]	Suspend LDO output voltage trim This field is used to adjust the output voltage of suspend LDO. 000 = Maximum voltage. 001 . . . 110 = Minimum voltage. 111 = 100.
4:0	LDOTRIM[4:0]	Main LDO output voltage trim This field is used to adjust the output voltage of main LDO. 0 0000 = Maximum voltage. 0 0001 1 1110 = Minimum voltage. 1 1111 = 1 0000.

B – B Register

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.

CAPCON3 – Input Capture Control 3

Register	SFR Address	Reset Value
CAPCON3	F1H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
CAP13	CAP12	CAP11	CAP10	CAP03	CAP02	CAP01	CAP00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:4]	CAP1[3:0]	Input capture channel 0 input pin select 0000 = P1.2/IC0 0001 = P1.1/IC1 0010 = P1.0/IC2 0011 = P0.0/IC3 0100 = P0.4/IC3 0101 = P0.1/IC4 0110 = P0.3/IC5 0111 = P0.5/IC6 1000 = P1.5/IC7 others = P1.2/IC0
[3:0]	CAP0[3:0]	Input capture channel 0 input pin select 0000 = P1.2/IC0 0001 = P1.1/IC1 0010 = P1.0/IC2 0011 = P0.0/IC3 0100 = P0.4/IC3 0101 = P0.1/IC4 0110 = P0.3/IC5 0111 = P0.5/IC6 1000 = P1.5/IC7 others = P1.2/IC0

SCnCR0 – SC Control Register 0

Register	SFR Address	Reset Value
SC0CR0	F1H, Page 2	0000_0000 b
SC1CR0	F3H, Page 2	0000_0000 b
SC2CR0	F5H, Page 2	0000_0000 b

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

Bit	Name	Description
7	NSB	<p>Stop Bit Length</p> <p>This field indicates the length of stop bit.</p> <p>0 = The stop bit length is 2 ETU.</p> <p>1 = The stop bit length is 1 ETU.</p> <p>Note: The default stop bit length is 2. SC and UART adopt NSB to program the stop bit length.</p>
6	T	<p>T Mode</p> <p>0 = T0 (ISO 7816-3 T = 0 mode).</p> <p>1 = T1 (ISO 7816-3 T = 1 mode).</p> <p>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 ISO 7816-3, 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.</p> <p>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.</p> <p>Note: The description please see section 6.10.8 Error Signal and Character Repetition</p>
5	RXBGTEN	<p>Receiver Block Guard Time Function Enable Bit</p> <p>0 = Receiver block guard time function Disabled.</p> <p>1 = Receiver block guard time function Enabled.</p>
4	CONSEL	<p>Convention Selection</p> <p>0 = Direct convention.</p> <p>1 = Inverse convention.</p> <p>Note 1: This bit is auto clear to "0", if AUTOGEN(SCnCR0[3]) is writing "1"</p> <p>Note 2: If AUTOGEN(SCnCR0[3]) is enabled, hardware will decide the convention and change the CONSEL (SCnCR0[4]) bits automatically after SCEN (SCnCR0[0]) ="1".</p>

Bit	Name	Description
3	AUTOZEN	<p>Auto Convention Enable Bit</p> <p>0 = Auto-convention Disabled.</p> <p>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.</p> <p>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.</p>
2	TXOFF	<p>TX Transition Disable Bit</p> <p>0 = The transceiver Enabled.</p> <p>1 = The transceiver Disabled.</p>
1	RXOFF	<p>RX Transition Disable Bit</p> <p>0 = The receiver Enabled.</p> <p>1 = The receiver Disabled.</p> <p>Note: If AUTOZEN (SCnCR0[3])is enabled, these fields must be ignored.</p>
0	SCEN	<p>SC Engine Enable Bit</p> <p>Set this bit to 1 to enable SC operation. If this bit is cleared, SC will force all transition to IDLE state</p> <p>Note: SCEN must be set to 1 before filling in other registers, or ISO 7816-3 will not work properly.</p>

CAPCON4 – Input Capture Control 4

Register	SFR Address	Reset Value
CAPCON4	F2H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	CAP23	CAP22	CAP21	CAP20
-	-	-	-	R/W	R/W	R/W	R/W

Bit	Name	Description
[3:0]	CAP2[3:0]	Input capture channel 0 input pin select 0000 = P1.2/IC0 0001 = P1.1/IC1 0010 = P1.0/IC2 0011 = P0.0/IC3 0100 = P0.4/IC3 0101 = P0.1/IC4 0110 = P0.3/IC5 0111 = P0.5/IC6 1000 = P1.5/IC7 others = P1.2/IC0

SCnCR1 – SC Control Register

Register	SFR Address	Reset Value
SC0CR1	F2H, Page 2	0000_0000 b
SC1CR1	F4H, Page 2	0000_0000 b
SC2CR1	F6H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
OPE	PBOFF	WLS		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 ISO 7816-3 mode, this field must be '0' (default setting is with parity bit)
5:4	WLS	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 ISO 7816-3 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 = ISO 7816-3 mode. 1 = UART mode. Note 1: When operating in UART mode, user must set CONSEL (SCnCR0[4]) = 0 and AUTOZEN(SCnCR0[3]) = 0. Note 2: When operating in ISO 7816-3 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.

SPCR – Serial Peripheral Control Register

Register	SFR Address	Reset Value
SPCR	F3H, 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 (SPSR.3) bit to determine the feature of \overline{SS} pin as shown in 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]	<p>SPI clock rate select</p> <p>These two bits select four grades of SPI clock divider. The clock rates below are illustrated under $F_{SYS} = 16$ MHz condition.</p> <p>$F_{sys} = 16$ MHz</p> <table border="1"> <thead> <tr> <th>SPR1</th> <th>SPR0</th> <th>Divider</th> <th>SPI clock rate</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>2</td> <td>8M bit/s</td> </tr> <tr> <td>0</td> <td>1</td> <td>4</td> <td>4M bit/s</td> </tr> <tr> <td>1</td> <td>0</td> <td>8</td> <td>WM bit/s</td> </tr> <tr> <td>1</td> <td>1</td> <td>16</td> <td>1 M bit/s</td> </tr> </tbody> </table> <p>$F_{sys} = 24$ MHz</p> <table border="1"> <thead> <tr> <th>SPR1</th> <th>SPR0</th> <th>Divider</th> <th>SPI clock rate</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>2</td> <td>12M bit/s</td> </tr> <tr> <td>0</td> <td>1</td> <td>4</td> <td>6M bit/s</td> </tr> <tr> <td>1</td> <td>0</td> <td>8</td> <td>3M bit/s</td> </tr> <tr> <td>1</td> <td>1</td> <td>16</td> <td>1.5M bit/s</td> </tr> </tbody> </table> <p>SPR[1:0] 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 $F_{SYS}/2$ communication speed.</p>	SPR1	SPR0	Divider	SPI clock rate	0	0	2	8M bit/s	0	1	4	4M bit/s	1	0	8	WM bit/s	1	1	16	1 M bit/s	SPR1	SPR0	Divider	SPI clock rate	0	0	2	12M bit/s	0	1	4	6M bit/s	1	0	8	3M bit/s	1	1	16	1.5M bit/s
SPR1	SPR0	Divider	SPI clock rate																																							
0	0	2	8M bit/s																																							
0	1	4	4M bit/s																																							
1	0	8	WM bit/s																																							
1	1	16	1 M bit/s																																							
SPR1	SPR0	Divider	SPI clock rate																																							
0	0	2	12M bit/s																																							
0	1	4	6M bit/s																																							
1	0	8	3M bit/s																																							
1	1	16	1.5M bit/s																																							

SPCR2 – Serial Peripheral Control Register 2

Register	SFR Address	Reset Value
SPCR2	F3H, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	-	SPIS1	SPIS0
-	-	-	-	-	-	R/W	R/W

Bit	Name	Description																																				
7:2	-	Reserved																																				
0	SPIS[1:0]	<p>SPI Interval time selection between adjacent bytes</p> <p>SPIS[1:0] and CPHA select eight grades of SPI interval time selection between adjacent bytes. As below table:</p> <table border="1"> <thead> <tr> <th>CPHA</th> <th>SPIS1</th> <th>SPIS0</th> <th>SPI clock</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0.0</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0.5</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>1.5</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>2.0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0.0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>1.0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>2.0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>2.5</td> </tr> </tbody> </table> <p>SPIS[1:0] are valid only under Master mode (MSTR = 1).</p>	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
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																																			

SPSR – Serial Peripheral Status Register

Register	SFR Address	Reset Value
SPSR	F4H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
SPIF	WCOL	SPIOVF	MODF	DISMODF	TXBUF	-	-
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 (EIE .0) and EA are enabled, an SPI interrupt will be required. This bit should be cleared via software. Attempting to write to SPDR 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 (SPCR.7) bit to determine the feature of \overline{SS} pin as shown in Table 6.12-1 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	TXBUF	SPI writer data buffer status This bit indicates the SPI transmit buffer status. 0 = SPI writer data buffer is empty 1 = SPI writer data buffer is full.

SPDR – Serial Peripheral Data Register

Register	SFR Address	Reset Value
SPDR	F5H, Page 0	0000_0000 b

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

Bit	Name	Description
7:0	SPDR[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.

AINDIDS0 – ADC Channel Digital Input Disconnect

Register	SFR Address	Reset Value
AINDIDS0	F6H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
P11DIDS	P03DIDS	P04DIDS	P05DIDS	P06DIDS	P07DIDS	P30DIDS	P17DIDS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
n	PnnDIDS	ADC Channel digital input disable 0 = ADC channel n digital input Enabled. 1 = ADC channel n digital input Disabled. ADC channel n is read always 0.

PIPSn – Pin Interrupt Control

Register	SFR Address	Reset Value
PIPS7	F7H, page 2	0000_0000 b
PIPS6	FFH, page 2	0000_0000 b
PIPS5	FEH, page 2	0000_0000 b
PIPS4	FDH, page 2	0000_0000 b
PIPS3	FCH, page 2	0000_0000 b
PIPS2	FBH, page 2	0000_0000 b
PIPS1	FAH, page 2	0000_0000 b
PIPS0	F9H, page 2	0000_0000 b

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

Bit	Name	Description
6:4	PSEL[2:0]	Pin interrupt channel Port select 000 = P0 PORT. 001 = P1 PORT. 010 = P2 PORT. 011 = P3 PORT. 100 = Reserved. 101 = Reserved. 110 = Reserved. 111 = 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].

AINDIDS1 – ADC Channel Digital Input Disconnect

Register	SFR Address	Reset Value
AINDIDS1	99H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
P25DIDS	P14DIDS	P13DIDS	P24DIDS	P23DIDS	P22DIDS	P21DIDS	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
n	PnnDIDS	ADC Channel digital input disable 0 = ADC channel n digital input Enabled. 1 = ADC channel n digital input Disabled. ADC channel n is read always 0.

SCON_1 – Serial Port 1 Control

Register	SFR Address	Reset Value
SCON_1	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 <u>SMOD0_1 (T3CON.6) = 0:</u>
6	SM1_1	See Table 6.9-2 Serial Port UART1 Mode / baud rate Description for details. <u>SMOD0_1 (T3CON.6) = 1:</u> 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.
5	SM2_1	Multiprocessor communication mode enable The function of this bit is dependent on the serial port 1 mode. <u>Mode 0:</u> No effect. <u>Mode 1:</u> 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. <u>Mode 2 or 3:</u> For multiprocessor communication. 0 = Reception is always valid no matter the logic level of the 9 th bit. 1 = Reception is valid only when the received 9 th 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	9 th transmitted bit This bit defines the state of the 9 th transmission bit in serial port 1 Mode 2 or 3. It is not used in Mode 0 or 1.
2	RB8_1	9 th received bit The bit identifies the logic level of the 9 th 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.

Bit	Name	Description
1	TL_1	<p>Transmission interrupt flag</p> <p>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.</p>
0	RI_1	<p>Receiving interrupt flag</p> <p>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.</p>

PWM0DTEN – PWM Dead-time Enable

Register	SFR Address	Reset Value
PWM0DTEN	F9H, Page 0, TA protected	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	PWM0DTCNT. 8	-	PDT45EN	PDT23EN	PDT01EN
-	-	-	R/W	-	R/W	R/W	R/W

Bit	Name	Description
4	PWM0DTCNT.8	PWM dead-time counter bit 8 See PWM0DTCNT register.
2	PDT45EN	PWM0_CH4 / PWM0_CH5 pair dead-time insertion enable This bit is valid only when PWM0_CH4 / PWM0_CH5 is under complementary mode. 0 = No delay on P0G4/P0G5 pair signals. 1 = Insert dead-time delay on the rising edge of P0G4/P0G5 pair signals.
1	PDT23EN	PWM0_CH2 / PWM0_CH3 pair dead-time insertion enable This bit is valid only when PWM0_CH2 / PWM0_CH3 is under complementary mode. 0 = No delay on P0G2/P0G3 pair signals. 1 = Insert dead-time delay on the rising edge of P0G2/P0G3 pair signals.
0	PDT01EN	PWM0_CH0 / PWM0_CH1 pair dead-time insertion enable This bit is valid only when PWM0_CH0 / PWM0_CH1 is under complementary mode. 0 = No delay on P0G0/P0G1 pair signals. 1 = Insert dead-time delay on the rising edge of P0G0/P0G1 pair signals.

PWM0DTCNT – PWM Dead-time Counter

Register	SFR Address	Reset Value
PWM0DTCNT	FAH, Page 0, 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]	<p>PWM dead-time counter low byte</p> <p>This 8-bit field combined with PWM0DTEN.4 forms a 9-bit PWM dead-time counter PWM0DTCNT. This counter is valid only when PWM is under complementary mode and the correspond PWM0DTEN bit for PWM pair is set.</p> $\text{PWM dead-time} = \frac{\text{PDTCNT} + 1}{F_{\text{SYS}}}$ <p>Note that user should not modify PWM0DTCNT during PWM run time.</p>

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

Register	SFR Address	Description	Reset Value
PWM0MEN	FBH, Page 0	PWM0_CHn Mask Enable	0000_0000 b
PWM1MEN	ADH, Page 2	PWM1_CHn Mask Enable	0000_0000 b
PWM2MEN	BDH, Page 2	PWM2_CHn Mask Enable	0000_0000 b
PWM3MEN	CDH, Page 2	PWM3_CHn Mask Enable	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
n	PMENn	PWMn_CHn mask enable 0 = PWMn_CHn signal outputs from its PWM generator. 1 = PWMn_CHn signal is masked by PMDx. Note: PMEN2~5 are only for PWM0.

PWMnMD – PWMnCx Mask Data

Register	SFR Address	Description	Reset Value
PWM0MD	FCH, Page 0	PWM0_CHn Mask Data	0000_0000 b
PWM1MD	ACH, Page 2	PWM1_CHn Mask Data	0000_0000 b
PWM2MD	BCH, Page 2	PWM2_CHn Mask Data	0000_0000 b
PWM3MD	CCH, Page 2	PWM3_CHn Mask Data	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
n	PMDn	<p>PWMn_CHn mask data</p> <p>The PWMn_CHn signal outputs mask data once its corresponding PMENx is set.</p> <p>0 = PWMn_CHn signal is masked by 0.</p> <p>1 = PWMn_CHn signal is masked by 1.</p> <p>Note: PMD2~5 are only for PWM0.</p>

PORDIS – POR Disable

Register	SFR Address	Reset Value
PORDIS	FDH, Page 0, 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 POR.

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
-	-	PPWM3	PPWM2	PPWM1	PWKT	PT3	PS_1
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
5	PPWM3	PWM3 interrupt priority low bit
4	PPWM2	PWM2 interrupt priority low bit
3	PPWM1	PWM1 interrupt priority low bit
2	PWKT	WKT interrupt priority low bit
1	PT3	Timer 3 interrupt priority low bit
0	PS_1	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 6.2-4 Interrupt Priority Level Setting](#) for correct interrupt priority configuration.

EIPH1 – Extensive Interrupt Priority High 1^[6]

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

7	6	5	4	3	2	1	0
-	-	PPWM3H	PPWM2H	PPWM1H	PWKTH	PT3H	PSH_1
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
5	PPWM3H	PWM3 interrupt priority high bit
4	PPWM2H	PWM2 interrupt priority high bit
3	PPWM1H	PWM1 interrupt priority high bit
2	PWKTH	WKT interrupt priority high bit
1	PT3H	Timer 3 interrupt priority high bit
0	PSH_1	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 6.2-4 Interrupt Priority Level Setting](#) for correct interrupt priority configuration.

LVRDIS – LVR disable

Register	SFR Address	Reset Value
LVRDIS	FFH, Page 1, 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.

6.2 System Manager

6.2.1 Clock System

6.2.1.1 Overview

The MS51 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 MS51 provides three options of the system clock sources including internal oscillator, or external clock from X_{IN} pin via software. The MS51 is embedded with two internal oscillators: one 10 kHz low-speed and one 16 MHz or 24 MHz high-speed, which is factory trimmed to ±2% under all conditions. A clock divider CKDIV is also available on MS51 for adjustment of the flexibility between power consumption and operating performance.

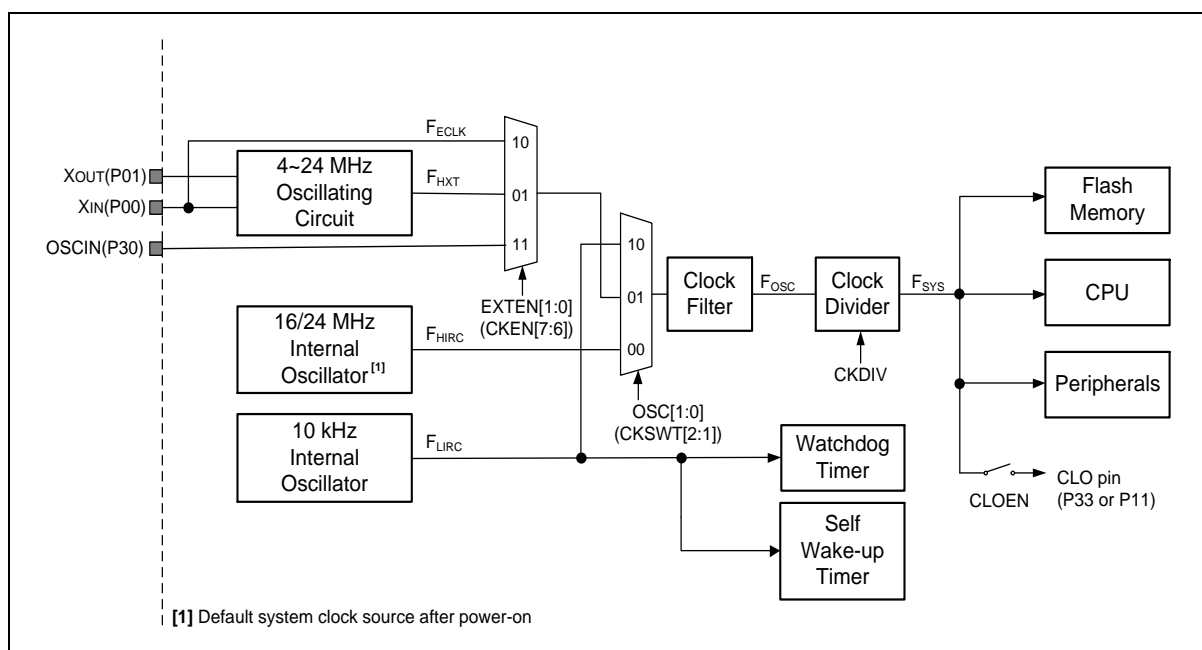


Figure 6.2-1 Clock System Block Diagram

6.2.1.2 System Clock Sources

There are a total of three system clock sources selectable in the MS51 including high-speed internal oscillator, low-speed internal oscillator and external clock input. Each of them can be the system clock source in the MS51. Different active system clock sources also affect multi-function of P0.0/X_{IN}.

6.2.1.3 Internal Oscillators

There are two internal oscillators in the MS51 – one high-speed internal oscillator (HIRC) 16 MHz or 24 MHz selectable base on HIRC24 (RCTRIM1.4) define. And one 10 kHz 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[1:0] (CKSWT [2:1]) as [1,1] to select the HIRC as the system clock. By setting OSC[1:0] as [1,0], LIRC will be selected as the system clock. Note that after the MS51 is powered, HIRC and LIRC will be both enabled and HIRC is default selected as the system clock source. While using internal oscillators, X_{IN} automatically switch as one general purpose I/O P0.0 to expend the number of general purpose I/O. The I/O output mode of P0.0 can be selected by configuring P0M1 and P0M2 registers.

Since HIRC value defined by a 9-bit data, user can adjust SFR to modify HIRC value, each bit of this value is about modify 40 kHz. Decrease this value will cause HIRC more quick and add one means HIRC speed slower about 40 kHz.

Following shows two Byte combine the 9-bit internal RC trim value. And HIRC 24 MHz define bit.

RCTRIM0 –High Speed Internal Oscillator Trim 0

Register	SFR Address	Reset Value
RCTRIM0	84H, Page0, TA protected	0000_0000b

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

RCTRIM1 –High Speed Internal Oscillator Trim 1

Register	SFR Address	Reset Value
RCTRIM1	85H, All pages, TA protected	0000_0000b

7	6	5	4	3	2	1	0
-	-	-	HIRC24	-	-	-	HIRCTRIM.0
-	-	-	R/W	-	-	-	R/W

Note: since default RCTRIM0 and RCTRIM1 value is base on 16 MHz, if base on this value then modify HIRC24(RCTIM1.4) to enable 24 MHz HIRC mode, the real HIRC deviation will more than 1%. Suggest reload the 24 MHz HIRC value from after UID, and check value bit 4 to confirm this value is suit for 24 MHz application.

6.2.1.4 External Crystal/Resonator or Clock Input

There are three possible clock source options of external clock sources – 4 MHz to 24 MHz high-speed crystal/resonator (HXT), and the external clock input (ECLK) through Xin pin. User should set OSC[1:0] as [1,0] then setting EXTEN[1:0] (CKEN[7:6]) to select external clock source. To select HXT as input source, define EXTEN[1:0] as [0,1]. To select ECLK as the system clock, when EXTEN[1:0] as [1,1], select external clock input via P3.0. Or as [1,0] input via P0.0.

When HXT is used as the system clock, X_{IN} and X_{OUT} are the input and output, respectively, of an internal inverting amplifier. A crystal or resonator should be connected between X_{IN} and X_{OUT} pins.

When enabling and selecting ECLK as the system clock source, the system clock is supplied via the X_{IN} pin. The common application is to drive X_{IN} with an active oscillator or clocks from another host device. Be aware that user should never feed any clock signal larger than voltage 2.4V to X_{IN} and give a DC voltage to X_{OUT} pin which value is half of X_{IN}, when ECLK mode is selected. Otherwise, it may break the device.

XTLCON – XLT Clock Control (TA Protected)

Register	SFR Address	Reset Value
XTLCON	D7H, Page 2	0111_0111 b

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

Bit	Name	Description
7	-	Reserved
6:4	HXSG	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)
3:0	-	Reserved

6.2.1.5 System Clock Switching

The MS51 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 SFRs 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[1:0] (CKSWT[2:1]). 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[1:0] value, OSC[1:0] 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

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

7	6	5	4	3	2	1	0
HXTST	ECKP00ST	HIRCST	LIRCST	ECKP30ST	OSC[1:0]		-
R	R	R	R	R	W		-

Bit	Name	Description
7	HXTST	High-speed external crystal 4 MHz to 24 MHz status 0 = High-speed external crystal is not stable or disabled. 1 = High-speed external crystal is enabled and stable.
6	ECKP00ST	External clock P00 input status 0 = External clock P00 input is not stable or disabled. 1 = External clock P00 input is enabled and stable.
5	HIRCST	High-speed internal oscillator 16 MHz status 0 = High-speed internal oscillator is not stable or disabled. 1 = High-speed internal oscillator is enabled and stable.
-	-	Reserved
3	ECKP30ST	External clock P30 input status 0 = External clock P30 input is not stable or disabled. 1 = External clock P30 input is enabled and stable.
2:1	OSC[1:0]	Oscillator selection bits This field selects the system clock source. 00 = Internal 16 MHz oscillator. 01 = External clock source according to EXTEN[1:0] (CKEN[7:6]) setting. 10 = Internal 10 kHz oscillator. 11 = Reserved. 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
CKSWT	97H, PAGE 0, TA protected	0011_0000 b

7	6	5	4	3	2	1	0
EXTEN[1:0]		HIRCEN	LIRCEN	-	-	-	CKSWTF
R/W		R/W	R/W	-	-	-	R

Bit	Name	Description
7:6	EXTEN[1:0]	External clock source enable 11 = External clock input via OSCIN (P30) Enabled 10 = External clock input via HXTIN (P00) Enabled 01 = External crystal 4~24 MHz Enabled 00 = external clock input is disable. P30/P00/P01 work as general purpose I/O or other functions.
5	HIRCEN	High-speed internal oscillator 16 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 16 MHz oscillator will be enabled automatically. The hardware will also set HIRCEN and HIRCST bits. After IAPEN is cleared, HIRCEN and EHCST resume the original values.
4	LIRCEN	Low speed internal oscillator 10 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 10 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 10 kHz oscillator is always enabled.
3:1	-	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.

6.2.1.6 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	95H, Page0	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, and $F_{SYS} = \frac{F_{osc}}{2 \times CKDIV}$, while CKDIV = 01H to FFH.

6.2.1.7 System Clock Output

The MS51 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, Page0	0000_0000b

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

Bit	Name	Description
1	CLOEN	System clock output enable 0 = System clock output Disabled. 1 = System clock output Enabled from CLO pin). Note: CLO pin decide by AUXR5.7.

6.2.2 Power Management

6.2.2.1 Overview

The MS51 has several features that help user to control the power consumption of the device. The power reduced feature has two option modes: Idle mode and Power-down mode, 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. If P2.0 is configured as a input-only pin, it should have an external pull-up or pull-low, or enable its internal pull-up by setting P30UP (P3UP.0).

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	Reserved	POF	GF1	GF0	PD	IDL
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W

Bit	Description	
1	PD	<p>Power-down mode</p> <p>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.</p> <p>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.</p>
0	IDL	<p>Idle mode</p> <p>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.</p>

6.2.2.2 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.

6.2.2.3 Power-Down Mode

Power-down mode is the lowest power state that the MS51 can enter. It remain the power consumption as A "Ma" 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 MS51 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 MS51 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 $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$, pin interrupt, WDT interrupt, WKT interrupt, and brown-out interrupt.

6.2.3 Power Monitoring and Reset

6.2.3.1 Overview

To prevent incorrect execution during power up and power drop, The MS51 provide three power monitor functions include power-on reset, low voltage reset and brown-out detection

The MS51 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.

6.2.3.2 Power-On Reset and Low Voltage Reset

The MS51 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

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.

PORDIS – POR Disable

Register	SFR Address	Reset Value
PORDIS	FDH, Page 0, 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 POR.

LVRDIS – LVR disable

Register	SFR Address	Reset Value
EIPH1	FFH, Page 1, 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.

6.2.3.3 Brown-Out Detection and Reset

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 4 nominal levels are 2.2V, 2.7V, 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).

The MS51 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 MS51 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)

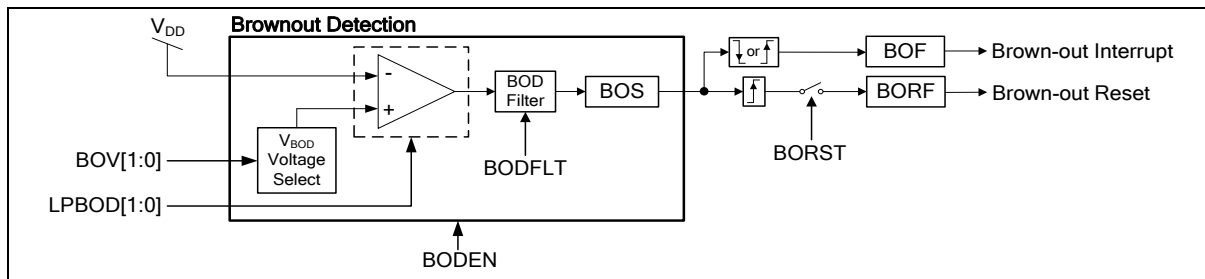


Figure 6.2-2 Brown-out Detection Block Diagram

CONFIG2

7	6	5	4	3	2	1	0
CBODEN	-	CBOV[1:0]		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 on. 0 = Brown-out detection circuit off.
5:4	CBOV[1:0]	CONFIG brown-out voltage select 11 = V_{BOD} is 2.2V. 10 = V_{BOD} is 2.7V. 01 = V_{BOD} is 3.7V. 00 = V_{BOD} is 4.4V.
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.

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

Table 6.2-1 BOF Reset Value

BODCON0 – Brown-out Detection Control 0

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 off. 1 = Brown-out detection circuit on. Note that BOD output is not available until 2~3 LIRC clocks after enabling.
6:4	BOV[1:0]	Brown-out voltage select 11 = V _{BOD} is 2.2V. 10 = V _{BOD} is 2.7V. 01 = V _{BOD} is 3.7V. 00 = V _{BOD} 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 V _{BOD} event occurs. If both EBOD (EIE.2) and EA (IE.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 V _{BOD} . 0 = Brown-out reset when V _{DD} drops below V _{BOD} Disabled. 1 = Brown-out reset when V _{DD} drops below V _{BOD} 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 V _{BOD} while BOD circuit is enabled. It keeps 0 if BOD is not enabled. 0 = V _{DD} voltage level is higher than V _{BOD} or BOD is disabled. 1 = V _{DD} voltage level is lower than V _{BOD} . Note that this bit is read-only.

NOTE:

- BODEN, BOV[2:0], and BORST are initialized by being directly loaded from CONFIG2 bit 7, [6:4], and 2 after all resets.
- BOF reset value depends on different setting of CONFIG2 and V_{DD} voltage level. Please check Table 6.2-1 BOF Reset Value

BODCON1 – Brown-out Detection Control 1

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 FSYS 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.)

BODFLT (BODCON1.1)	BOD Operation Mode	System Clock Source	Minimum Brown-Out Detect Pulse Width
0	Normal mode (LPBOD[1:0] = [0,0])	Any clock source	Typ. 1µs
	Low power mode 1 (LPBOD[1:0] = [0,1])	Any clock source	16 (1/FLIRC)
	Low power mode 2 (LPBOD[1:0] = [1,0])	Any clock source	64 (1/FLIRC)
	Low power mode 3 (LPBOD[1:0] = [1,1])	Any clock source	256 (1/ FLIRC)
1	Normal mode (LPBOD[1:0] = [0,0])	HIRC/ECLK	Normal operation: 32 (1/FSYS) Idle mode: 32 (1/FSYS) Power-down mode: 2 (1/FLIRC)
		LIRC	2 (1/FLIRC)
	Low power mode 1 (LPBOD[1:0] = [0,1])	Any clock source	18 (1/FLIRC)

BODFLT (BODCON1.1)	BOD Operation Mode	System Clock Source	Minimum Brown-Out Detect Pulse Width
	Low power mode 2 (LPBOD[1:0] = [1,0])	Any clock source	66 (1/FLIRC)
	Low power mode 3 (LPBOD[1:0] = [1,1])	Any clock source	258 (1/ FLIRC)

Table 6.2-2 Minimum Brown-out Detect Pulse Width

6.2.3.4 External Reset and Hard Fault Reset

The external reset pin nRESET is an input with a Schmitt trigger. An external reset is accomplished by holding the nRESET 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.

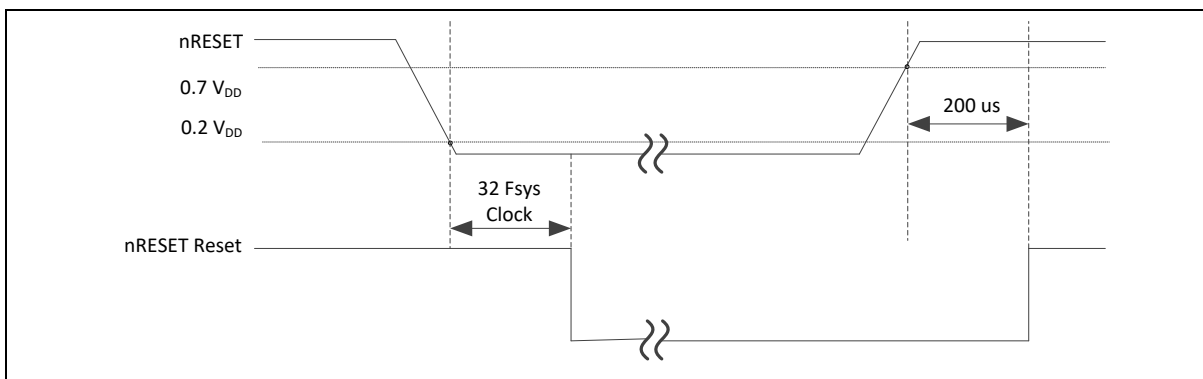


Figure 6.2-3 nRESET Reset Waveform

Once the device is in reset condition, it will remain as long as nRESET pin is low. After the nRESET 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 (AUXR1.6) 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 (AUXR1.5) 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.

AUXR1 – Auxiliary Register 1

Register	SFR Address	Reset Value
AUXR1	A2H , Page 0	POR: 0000 0000b, Software reset: 1U00 0000b, nRESET pin: U100 0000b, Others: UUUU 0000b

7	6	5	4	3	2	1	0
SWRF	RSTPINF	HardF	SLOW	GF2	UART0PX	0	DPS
R/W	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 EHF1 (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.

6.2.3.5 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

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
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.

6.2.3.6 Software Reset

The MS51 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 (AUXR1.7) 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

Register	SFR Address	Reset Value
CHPCON	9FH, All page, 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.

AUXR1 – Auxiliary Register 1

Register	SFR Address	Reset Value
AUXR1	A2H , Page 0	POR: 0000 0000b, Software reset: 1U00 0000b, nRESET pin: U100 0000b, Others: UUUU 0000b

7	6	5	4	3	2	1	0
SWRF	RSTPINF	HardF	SLOW	GF2	UART0PX	0	DPS
R/W	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  AUXR1, #01111111b ;software reset flag clear
CLR  EA
MOV  TA, #0Aah
MOV  TA, #55h
ORL  CHPCON, #10000000b ;software reset
    
```


6.2.3.7 Boot Select

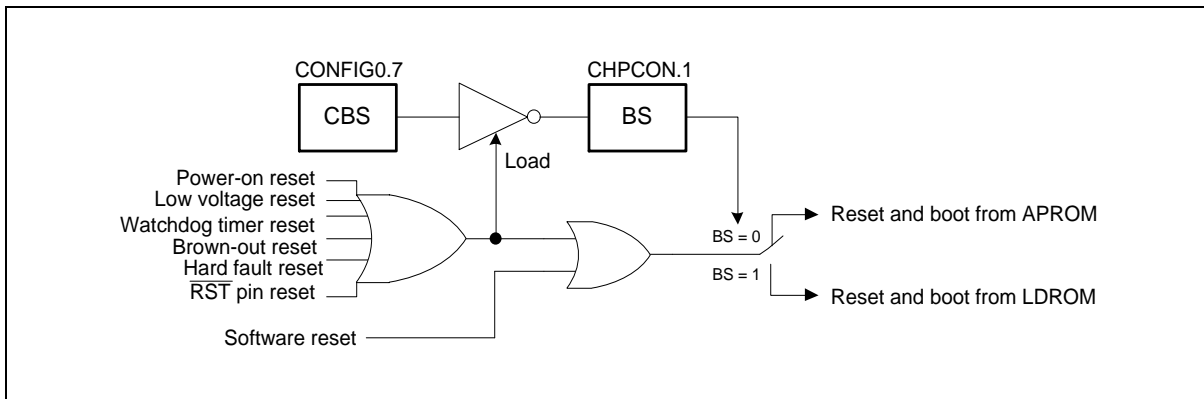


Figure 6.2-4 Boot Selecting Diagram

The MS51 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	-	OCDPWM	OCDEN	-	-	LOCK	-
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

Register	SFR Address	Reset Value
CHPCON	9FH, All page, 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

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.
Note: 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.

6.2.3.8 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.

6.2.4 Interrupt System

6.2.4.1 Overview

The purpose of the interrupt is to make the software deal with unscheduled or asynchronous events. The MS51 has a four-priority-level interrupt structure with 24 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 6.2-3 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 Address	Vector Number	Source	Vector Address	Vector Number
Reset	0000H	-	Input capture interrupt	0063H	12
External interrupt 0	0003H	0	PWM0 interrupt	006BH	13
Timer 0 overflow	000BH	1	Fault Brake interrupt	0073H	14
External interrupt 1	0013H	2	Serial port 1 interrupt	007BH	15
Timer 1 overflow	001BH	3	Timer 3 overflow	0083H	16
Serial port 0 interrupt	0023H	4	Self Wake-up Timer interrupt	008BH	17
Timer 2 event	002BH	5	PWM1 interrupt	0093H	18
I ² C status/timer-out interrupt	0033H	6	PWM2 interrupt	009BH	19
Pin interrupt	003BH	7	PWM3 interrupt	00A3H	20
Brown-out detection interrupt	0043H	8	SC0 interrupt	00ABH	21
SPI interrupt	004BH	9	SC1 interrupt	00B3H	22
WDT interrupt	0053H	10	SC2 interrupt	00BBH	23
ADC interrupt	005BH	11			

Table 6.2-3 Interrupt Vectors

6.2.4.2 *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 (IE.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 = Interrupt generated by ADCF (ADCCON0.7) Enabled.
5	EBOD	Enable brown-out interrupt 0 = Brown-out detection interrupt Disabled. 1 = Interrupt generated by BOF (BODCON0.3) Enabled.
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.
3	ET1	Enable Timer 1 interrupt 0 = Timer 1 interrupt Disabled. 1 = Interrupt generated by TF1 (TCON.7) Enabled.
2	EX1	Enable external interrupt 1 0 = External interrupt 1 Disabled. 1 = Interrupt generated by $\overline{INT1}$ pin (P1.7) Enabled.
1	ET0	Enable Timer 0 interrupt 0 = Timer 0 interrupt Disabled. 1 = Interrupt generated by TF0 (TCON.5) Enabled.
0	EX0	Enable external interrupt 0 0 = External interrupt 0 Disabled. 1 = Interrupt generated by $\overline{INT0}$ pin (P3.0) Enabled.

EIE – Extensive Interrupt Enable

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

7	6	5	4	3	2	1	0
ET2	ESPI	EFB	EWDT	EPWM0	ECAP	EPI	EI ² C
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 = Interrupt generated by TF2 (T2CON.7) Enabled.
6	ESPI	Enable SPI interrupt 0 = SPI interrupt Disabled. 1 = Interrupt generated by SPIF (SPSR.7), SPIOVF (SPSR.5), or MODF (SPSR.4) Enable.
5	EFB	Enable Fault Brake interrupt 0 = Fault Brake interrupt Disabled. 1 = Interrupt generated by FBF (PWM0FBD.7) Enabled.
4	EWDT	Enable WDT interrupt 0 = WDT interrupt Disabled. 1 = Interrupt generated by WDTF (WDCON.5) Enabled.
3	EPWM0	Enable PWM0 interrupt 0 = PWM interrupt Disabled. 1 = Interrupt generated by PWMF (PWM0CON0.5) Enabled.
2	ECAP	Enable input capture interrupt 0 = Input capture interrupt Disabled. 1 = Interrupt generated by any flags of CAPF[2:0] (CAPCON0[2:0]) Enabled.
1	EPI	Enable pin interrupt 0 = Pin interrupt Disabled. 1 = Interrupt generated by any flags in PIF register Enabled.
0	EI ² C	Enable I ² C interrupt 0 = I ² C interrupt Disabled. 1 = Interrupt generated by SI (I2CON.3) or I2TOF (I2TOC.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
-	-	EPWM3	EPWM2	EPWM1	EWKT	ET3	ES_1
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
5	EPWM3	Enable PWM3 interrupt 0 = PWM3 interrupt Disabled. 1 = Interrupt generated by PWM3F (PWM3CON0.5) Enabled.
4	EPWM2	Enable PWM2 interrupt 0 = PWM2 interrupt Disabled. 1 = Interrupt generated by PWM2F (PWM2CON0.5) Enabled.
3	EPWM1	Enable PWM1 interrupt 0 = PWM1 interrupt Disabled. 1 = Interrupt generated by PWM1F (PWM1CON0.5) Enabled.
2	EWKT	Enable WKT interrupt 0 = WKT interrupt Disabled. 1 = Interrupt generated by WKTF (WKCON.4) Enabled.
1	ET3	Enable Timer 3 interrupt 0 = Timer 3 interrupt Disabled. 1 = Interrupt generated by TF3 (T3CON.4) Enabled.
0	ES_1	Enable serial port 1 interrupt 0 = Serial port 1 interrupt Disabled. 1 = Interrupt generated by TI_1 (SCON_1.1) or RI_1 (SCON_1.0) Enabled.

6.2.4.3 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 6.2-4](#) 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.

This natural priority is defined as shown on [Table 6.2-4 Interrupt Priority Level Setting](#)

It 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 6.2.2.3 "Power-Down Mode"](#)

Interrupt Priority Control Bits		Interrupt Priority Level
IPH / EIPH / EIPH1	IP / EIP / EIP2	
0	0	Level 0 (lowest)
0	1	Level 1
1	0	Level 2
1	1	Level 3 (highest)

Table 6.2-4 Interrupt Priority Level Setting

Interrupt Source	Vector Address	Interrupt Flag(S)	Enable Bit	Natural Priority	Priority Control Bits	Power-Down Wake-Up
Reset	0000H	-	Always Enabled	Highest	-	Yes
External interrupt 0	0003H	IE0[1]	EX0	1	PX0, PX0H	Yes
Brown-out	0043H	BOF (BODCON0.3)	EBOD	2	PBOD, PBODH	Yes
Watchdog Timer	0053H	WDTF (WDCON.5)	EWDT	3	PWDT, PWDTH	Yes
Timer 0	000BH	TF0[2]	ET0	4	PT0, PT0H	No
I2C status/time-out	0033h	SI + I2TOF (I2TOC.0)	EI2C	5	PI2C, PI2CH	No
ADC	005Bh	ADCF	EADC	6	PADC, PADCH	No
External interrupt 1	0013H	IE1[1]	EX1	7	PX1, PX1H	Yes
Pin interrupt	003BH	PIF0 to PIF7 (PIF)[3]	EPI	8	PPI, PPIH	Yes
Timer 1	001BH	TF1[2]	ET1	9	PT1, PT1H	No
Serial port 0	0023H	RI + TI	ES	10	PS, PSH	No
Fault Brake event	0073h	FBF (PWM0FBD.7)	EFB	11	PFB, PFBH	No
SPI	004Bh	SPIF (SPSR.7) + MODF (SPSR.4) + SPIOVF (SPSR.5)	ESPI	12	PSPI, PSPIH	No
Timer 2	002BH	TF2[2]	ET2	13	PT2, PT2H	No
Input capture	0063H	CAPF[2:0] (CAPCON0[2:0])	ECAP	14	PCAP, PCAPH	No
PWM0 interrupt	006BH	PWMF	EPWM0	15	PPWM, PPWMH	No
Serial port 1	007BH	RI_1 + TI_1	ES_1	16	PS_1, PSH_1	No
Timer 3	0083H	TF3[2] (T3CON.4)	ET3	17	PT3, PT3H	No
Self Wake-up Timer	008BH	WKTF (WKCON.4)	EWKT	18	PWKT, PWKTH	Yes
PWM1 interrupt	0093H	PWM1F	EPWM1	19	PPWM1, PPWM1H	No
PWM2 interrupt	009BH	PWM2F	EPWM2	20	PPWM2, PPWM2H	No
PWM3 interrupt	00A3H	PWM3F	EPWM3	21	PPWM3, PPWM3H	No
SC0 interrupt	00ABH	ACERR+BGT+TERR+TBE+ RDA (SC0IS[4:0])	SC0IE	22	PSC0, PSC0H	No
SC1 interrupt	00B3H	ACERR+BGT+TERR+TBE+ RDA (SC1IS[4:0])	SC1IE	23	PSC1, PSC1H	No
SC2 interrupt	00BBH	ACERR+BGT+TERR+TBE+ RDA (SC2IS[4:0])	SC2IE	24	PSC2, PSC2H	No

Note:

1. While the external interrupt pin is set as edge triggered (Itx = 1), its own flag lex will be automatically cleared if the interrupt service routine (ISR) is executed. While as level triggered (Itx = 0), lex 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 6.2-5 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
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: IP is used in combination with the IPH to determine the priority of each interrupt source. See [Table 6.2-4 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
6	PADC	ADC interrupt priority high bit
5	PBOD	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

Note: IPH is used in combination with the IP respectively to determine the priority of each interrupt source. See Table 6.2-4 Interrupt Priority Level Setting for correct interrupt priority configuration.

EIP – Extensive Interrupt Priority

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

7	6	5	4	3	2	1	0
PT2	PSPI	PFB	PWDT	PPWM	PCAP	PPI	PI ² C
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	PSPI	SPI interrupt priority low bit
5	PFB	Fault Brake interrupt priority low bit
4	PWDT	WDT interrupt priority low bit
3	PPWM	PWM interrupt priority low bit
2	PCAP	Input capture interrupt priority low bit
1	PPI	Pin interrupt priority low bit
0	PI ² C	I ² C interrupt priority low bit

Note : EIP is used in combination with the EIPH to determine the priority of each interrupt source. See Table 6.2-4 Interrupt Priority Level Setting for correct interrupt priority configuration.

EIPH – Extensive Interrupt Priority High^[4]

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

7	6	5	4	3	2	1	0
PT2H	PSPIH	PFBH	PWDTH	PPWMH	PCAPH	PPIH	PI2CH
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	PSPIH	SPI interrupt priority high bit
5	PFBH	Fault Brake interrupt priority high bit
4	PWDTH	WDT interrupt priority high bit
3	PPWMH	PWM interrupt priority high bit
2	PCAPH	Input capture interrupt priority high bit
1	PPIH	Pin interrupt priority high bit
0	PI2CH	I ² C interrupt priority high bit

Note : EIPH is used in combination with the EIP to determine the priority of each interrupt source. See Table 6.2-4 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
-	-	PPWM3	PPWM2	PPWM1	PWKT	PT3	PS_1
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
5	PPWM3	PWM3 interrupt priority low bit
4	PPWM2	PWM2 interrupt priority low bit
3	PPWM1	PWM1 interrupt priority low bit
2	PWKT	WKT interrupt priority low bit
1	PT3	Timer 3 interrupt priority low bit
0	PS_1	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 6.2-4 Interrupt Priority Level Setting for correct interrupt priority configuration.

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
-	-	PPWM3H	PPWM2H	PPWM1H	PWKTH	PT3H	PSH_1
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
5	PPWM3H	PWM3 interrupt priority high bit
4	PPWM2H	PWM2 interrupt priority high bit
3	PPWM1H	PWM1 interrupt priority high bit
2	PWKTH	WKT interrupt priority high bit
1	PT3H	Timer 3 interrupt priority high bit
0	PSH_1	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 6.2-4 Interrupt Priority Level Setting for correct interrupt priority configuration.

EIP2 – Extensive Interrupt Priority 2

Register	SFR Address	Reset Value
EIP2	CEH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	PSC2	PSC1	PSC0
-	-	-	-	-	R/W	R/W	R/W

Bit	Name	Description
2	PSC2	SC2 / UART4 interrupt priority low bit
1	PSC1	SC1 / UART3 interrupt priority low bit
0	PSC0	SC0 / UART2 interrupt priority low bit

Note: EIP2 is used in combination with the EIPH2 to determine the priority of each interrupt source. See Table 6.2-4 Interrupt Priority Level Setting for correct interrupt priority configuration.

EIPH2 – Extensive Interrupt Priority High 2

Register	SFR Address	Reset Value
EIPH2	CFH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	PUART4H	PUART3H	PUART2H
-	-	-	-	-	R/W	R/W	R/W

Bit	Name	Description
2	PUART4H	UART4 interrupt priority high bit
1	PUART3H	UART3 interrupt priority high bit
0	PUART2H	UART2 interrupt priority high bit

Note: EIPH2 is used in combination with the EIP2 to determine the priority of each interrupt source. See Table 6.2-4 Interrupt Priority Level Setting for correct interrupt priority configuration.

6.2.4.4 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.

6.2.4.5 *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.

6.3 Flash Memory Control

6.3.1 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 MS51 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 μ s. 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.

6.3.1.1 IAP Commands

The MS51 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
96-bit Unique Code read	XX	0	0	0100	0000H to 000BH	Data out
APROM page-erase	00	1	0	0010	Address in ^[2]	FFH
LDROM page-erase	01	1	0	0010	Address in ^[2]	FFH
APROM byte-program	00	1	0	0001	Address in	Data in
LDROM byte-program	01	1	0	0001	Address in	Data in
APROM byte-read	00	0	0	0000	Address in	Data out
LDROM byte-read	01	0	0	0000	Address in	Data out
All CONFIG bytes erase	11	1	0	0010	0000H	FFH

IAP Mode	IAPCN				IAPA[15:0] {IAPAH, IAPAL}	IAPFD[7:0]
	IAPB [1:0]	FOEN	FCEN	FCTRL [3:0]		
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
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						

Table 6.3-1 IAP Modes and Command Codes

6.3.1.2 Control register of IAP

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

Register	SFR Address	Reset Value
CHPCON	9FH, All page, 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	<p>IAP fault flag</p> <p>The hardware will set this bit after IAPGO (IAPTRG.0) is set if any of the following condition is met:</p> <p>(1) The accessing address is oversize.</p> <p>(2) IAPCN command is invalid.</p> <p>(3) IAP erases or programs updating un-enabled block.</p> <p>(4) IAP erasing or programming operates under V_{BOD} while BOIAP (CONFIG2.5) remains un-programmed 1 with BODEN (BODCON0.7) as 1 and BORST (BODCON0.2) as 0.</p> <p>This bit should be cleared via software.</p>
0	IAPEN	<p>IAP enable</p> <p>0 = IAP function Disabled.</p> <p>1 = IAP function Enabled.</p> <p>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.</p>

IAPUEN – IAP Updating Enable

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 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.

IAPCN – IAP Control

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

Bit	Name	Description
7:6	IAPB[1:0]	IAP control This byte is used for IAP command. For details, see Table 6.3-1 IAP Modes and Command Codes.
5	FOEN	
4	FCEN	
3:0	FCTRL[3:0]	

IAPAH – IAP Address High Byte

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

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

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	<p>IAP go</p> <p>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.</p> <p>Before triggering an IAP action, interrupts (if enabled) should be temporary disabled for hardware limitation.</p> <p>The program process should follows below.</p> <pre> CLR EA MOV TA,#0AAH MOV TA,#55H ORL IAPTRG,#01H (SETB EA) </pre>

6.3.1.3 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 temporarily 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.

6.3.1.4 Using Flash Memory as Data Flash

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 MS51 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 :

```

;*****
; 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 MOVX. Data Flash can be defined as
128-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);
}

```

6.3.1.5 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 MS51. It makes user quite easy perform ISP through UART port. Please visit Nuvoton 8-bit Microcontroller website: [Nuvoton 80C51 Microcontroller Technical Support](http://www.nuvoton.com.tw/Products/8-bit/Microcontroller/MS51/Microcontroller_Technical_Support.aspx). 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,#1111101b  ;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,#0000001b  ;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,#0000001b  ;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

```

```

S JMP $

;*****
;      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 "S JMP  $"

END
    
```

6.3.2 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} , ICP_DAT, and ICP_CLK, involved in ICP function. \overline{RST} is used to enter or exit ICP mode. ICP_DAT is the data input and output pin. ICP_CLK 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 V_{SS} pins on the circuit board to make ICP possible.

Nuvoton provides ICP tool for MS51, 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](#).

6.3.3 On-Chip-Debugger (OCD)

6.3.3.1 Overview

The MS51 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, ICE_DAT and ICE_CLK, to establish communication between the target device and the controlling debugger host. ICE_DAT is an input/output pin for debug data transfer and ICE_CLK is an input pin for synchronization with ICE_DAT data. The P2.0/nRESET pin is also necessary for OCD mode entry and exit. The MS51 supports OCD with Flash Memory control path by ICP writer mode, which shares the same three pins of OCD interface.

The MS51 uses ICE_DAT, ICE_CLK, and P2.0/nRESET 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. If P2.0/nRESET is configured as external reset pin, it cannot be connected directly to V_{DD} and

any external capacitors connected must be removed.

2. If P2.0/nRESET is configured as input pin P2.0, any external input source must be isolated.
3. All external reset sources must be disconnected.
4. Any external component connected on ICE_DAT and ICE_CLK must be isolated.

6.3.3.2 Limitation of OCD

The MS51 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 P2.0/nRESET pin needs to be used for OCD mode selection. Therefore, neither P2.0 input nor an external reset source can be emulated.
2. The ICE_DAT pin is physically located on the same pin P1.6. Therefore, neither its I/O function nor shared multi-functions can be emulated.
3. The ICE_CLK pin is physically located on the same pin as P0.2. 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 MS51 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 control registers 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	-	OCDPWM	OCDEN	-	RPD	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. Note that this bit is valid only when the corresponding PIO bit of PWM channel is set as 1.
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 be disabled and only HardF flag be asserted.

6.3.4 96-Bit Unique Code

Before shipping out, each MS51 chip was factory pre-programmed with a 96-bit width serial number, which is guaranteed to be unique. The serial number is called Unique Code. The user can read the Unique Code only by IAP command. Please see Chapter 6.3.1.1IAP Commands.

6.4 General Purpose I/O (GPIO)

6.4.1 Gpio Mode

The MS51 has a maximum of 30 general purpose I/O pins which 29 bit-addressable general I/O pins grouped as 4 ports, P0 to P3, and 1 input only pin as P20. 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, whereas 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 1: n = 0~3, X = 0~7

Table 6.4-1 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 control registers 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
```

6.4.1.1 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.

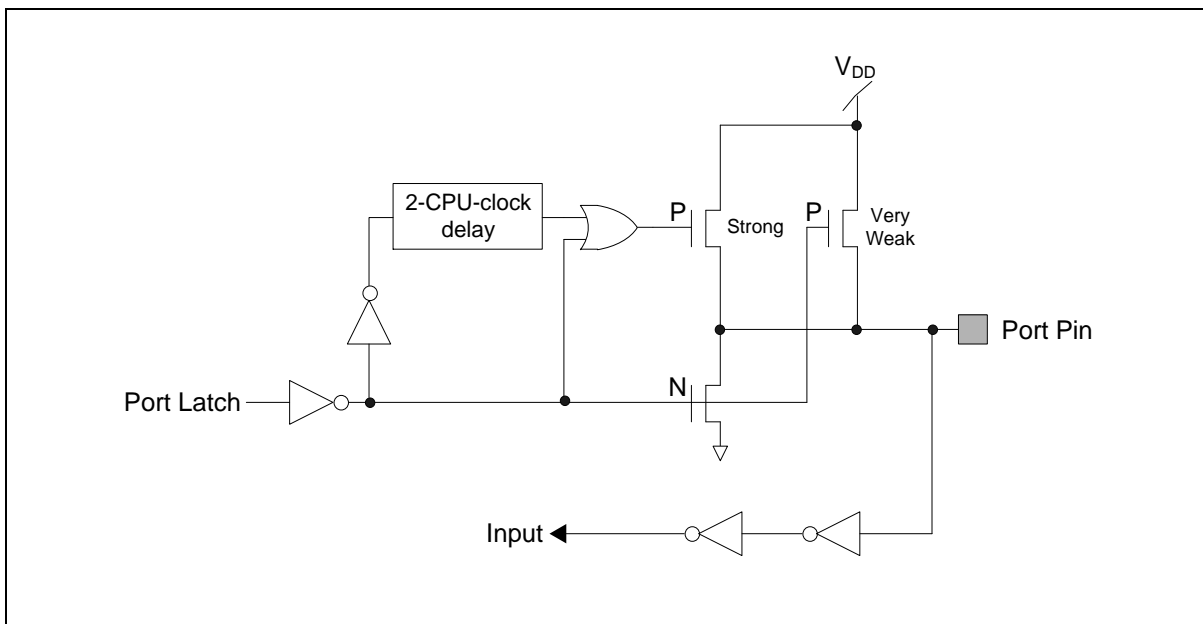


Figure 6.4-1 Quasi-Bidirectional Mode Structure

6.4.1.2 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.

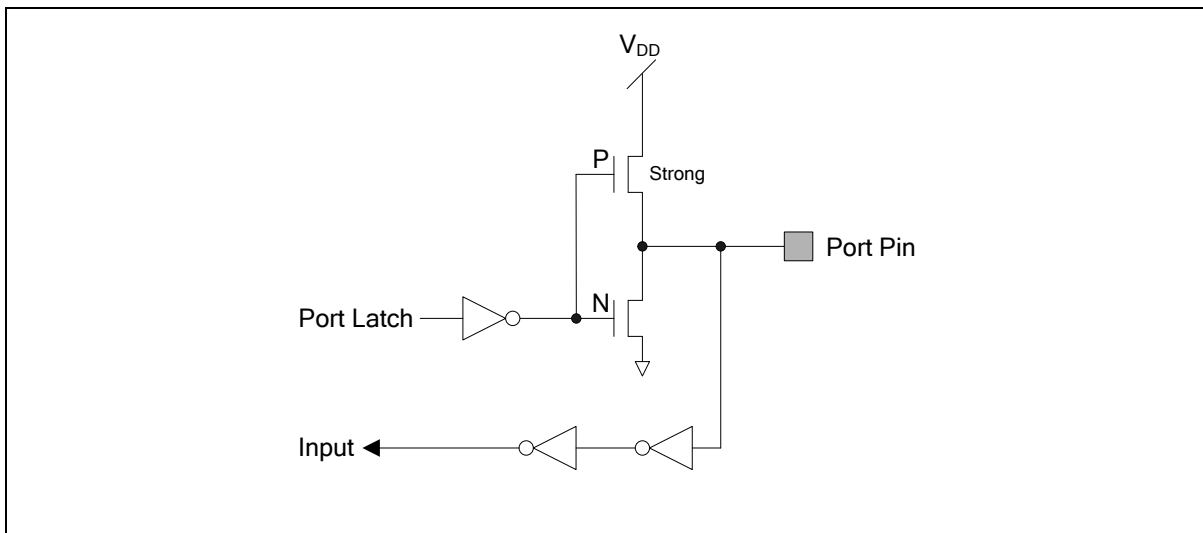


Figure 6.4-2 Push-Pull Mode Structure

6.4.1.3 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.

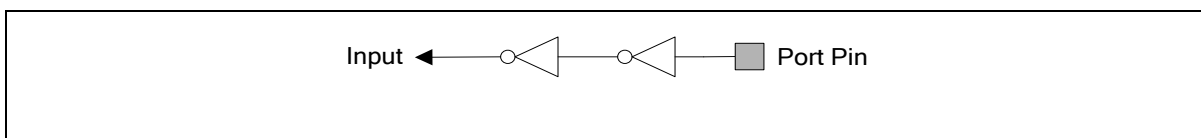


Figure 6.4-3 Input-Only Mode Structure

6.4.1.4 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 I²C 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.

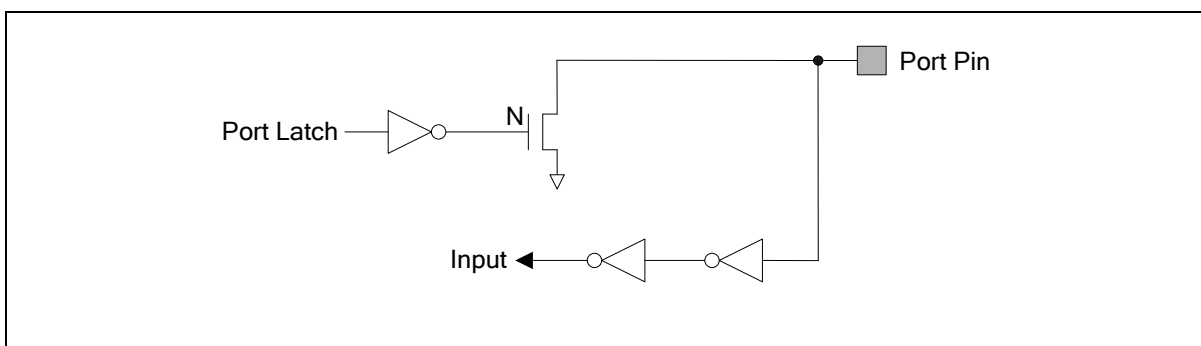


Figure 6.4-4 Open-Drain Mode Structure

6.4.2 External Interrupt Pins (INT0 and INT1)

The external interrupt $\overline{INT0}$ and $\overline{INT1}$ 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 $\overline{INT0}$ or $\overline{INT1}$ 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 $\overline{INT0}$ and $\overline{INT1}$ 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 $\overline{INT0}$ and $\overline{INT1}$ can wake up the device from the Power-down 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
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 $\overline{INT1}$ input signal's logic level. Software cannot control it.
2	IT1	External interrupt 1 type select This bit selects by which type that $\overline{INT1}$ is triggered. 0 = $\overline{INT1}$ is low level triggered. 1 = $\overline{INT1}$ 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 $\overline{INT0}$ input signal's logic level. Software cannot control it.
0	IT0	External interrupt 0 type select This bit selects by which type that $\overline{INT0}$ is triggered. 0 = $\overline{INT0}$ is low level triggered. 1 = $\overline{INT0}$ is falling edge triggered.

6.4.3 Pin Interrupt (PIT)

The MS51 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 us– d – 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.

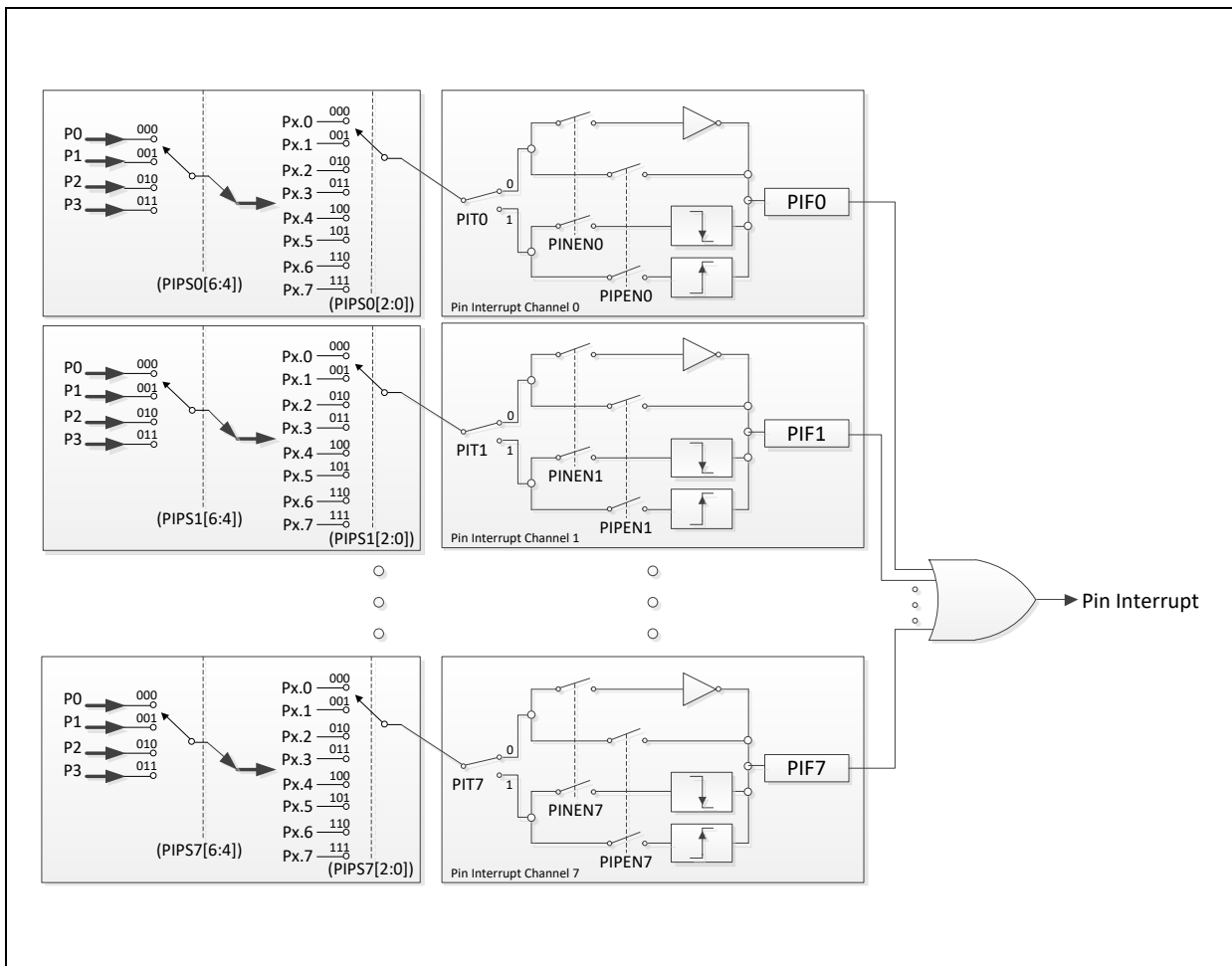


Figure 6.4-5 Pin Interface Block Diagram

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.

6.4.4 Control Registers of GPIO

The MS51 has a lot of I/O control registers to provide flexibility in all kinds of applications. The SFR related with I/O ports can be categorized into four groups: input and output control, output mode control, input type and sink current control, and output slew rate control. All of SFR are listed as follows.

6.4.4.1 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 (Bit-addressable)

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	0011_1111 b
P3	B0H, All pages, Bit-addressable	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.

6.4.4.2 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.X	PnM2.X	I/O Type
0	0	Quasi-bidirectional
0	1	Push-pull
1	0	Input-only (high-impedance)
1	1	Open-drain

PnM1 – Port Mode Select 1

Register	SFR Address	Reset Value
P0M1	B1H, Page 0	1111_1111 b
P1M1	B3H, Page 0	1111_1111 b
P2M1	89H, Page 2	0011_1111 b
P3M1	ACH, Page 0	1111_1111 b

7	6	5	4	3	2	1	0
PnM1.7	PnM1.6	PnM1.5	PnM1.4	PnM1.3	PnM1.2	PnM1.1	PnM1.0
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 0 mode select 1

PnM2 – Port Mode Select 2

Register	SFR Address	Reset Value
P0M2	B2H, Page 0	0000_0000 b
P1M2	B4H, Page 0	0000_0000 b
P2M2	8AH, Page 2	0000_0000 b
P3M2	ADH, Page 0	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: PxM1 and PxM2 are used in combination to determine the I/O mode of each pin of Port. See Table 6.4-1 Configuration for Different I/O Modes .

6.4.4.3 Input Type Select

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	B1H, Page 1	0000_0000 b
P1S	B3H, Page 1	0000_0000 b
P2S	8CH, Page 2	0000_0000 b
P3S	ACH, 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.

6.4.4.4 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	B2H, Page 1	0000_0000 b
P1SR	B4H, Page 1	0000_0000 b
P2SR	8BH, Page 2	0000_0000 b
P3SR	ADH, Page 1	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 slew rate 0 = Pn.x normal output slew rate. 1 = Pn.x high-speed output slew rate.

PICON – Pin Interrupt Control

Register	SFR Address	Reset Value
PICON	E9H, PAGE 0,	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.
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 0	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
n	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 0	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
n	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	ECH, 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
n	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
PIPS7	F7H, page 2	0000_0000 b
PIPS6	FFH, page 2	0000_0000 b
PIPS5	FEH, page 2	0000_0000 b
PIPS4	FDH, page 2	0000_0000 b
PIPS3	FCH, page 2	0000_0000 b
PIPS2	FBH, page 2	0000_0000 b
PIPS1	FAH, page 2	0000_0000 b
PIPS0	F9H, page 2	0000_0000 b

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

Bit	Name	Description
6:4	PSEL[2:0]	Pin interrupt channel Port select 000 = P0 PORT. 001 = P1 PORT. 010 = P2 PORT. 011 = P3 PORT. 100 = Reserved. 101 = Reserved. 110 = Reserved. 111 = 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].

6.5 Timer

6.5.1 Timer/Counter 0 And 1

6.5.1.1 Overview

Timer/Counter 0 and 1 on MS51 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/\bar{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 TOM (CKCON.3) bit for Timer 0 and T1M (CKCON.4) bit for Timer 1. In the “Counter” mode, the counter 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/\bar{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 temporarily 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.

6.5.1.2 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 $\bar{INT}0$ ($\bar{INT}1$) is 1. Gate setting as 1 allows the Timer to calculate the pulse width on external input pin $\bar{INT}0$ ($\bar{INT}1$). When the 13-bit value moves from 1FFFH to 0000H, the Timer overflow flag TF0 (TF1) is set and an interrupt occurs if enabled.

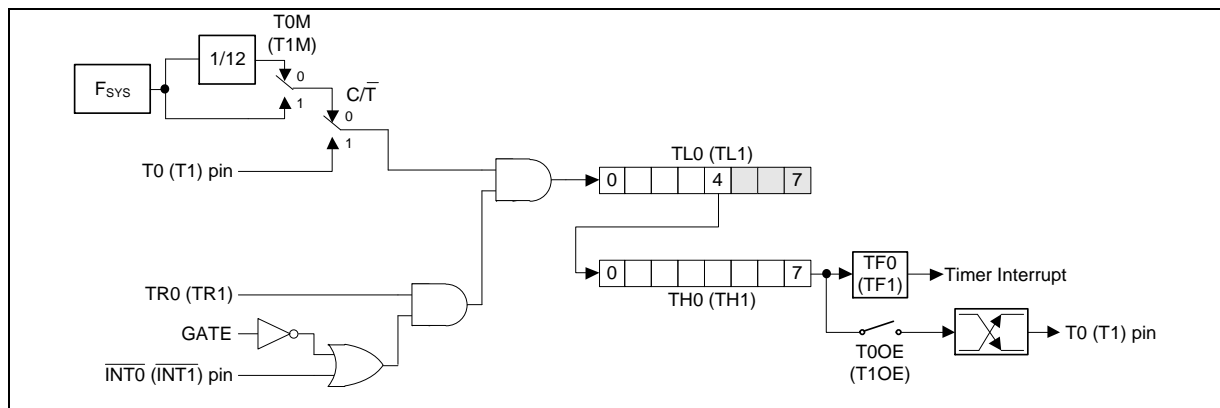


Figure 6.5-1 Timer/Counters 0 and 1 in Mode 0

6.5.1.3 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.

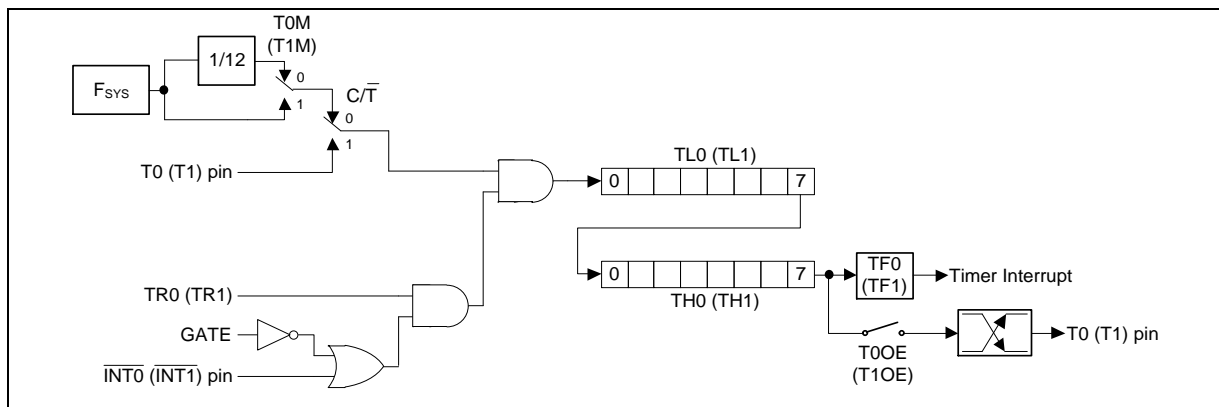


Figure 6.5-2 Timer/Counters 0 and 1 in Mode 1

6.5.1.4 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 $\overline{\text{INT0}}$ ($\overline{\text{INT1}}$) pins. The functions of GATE and $\overline{\text{INT0}}$ ($\overline{\text{INT1}}$) pins are just the same as Mode 0 and 1.

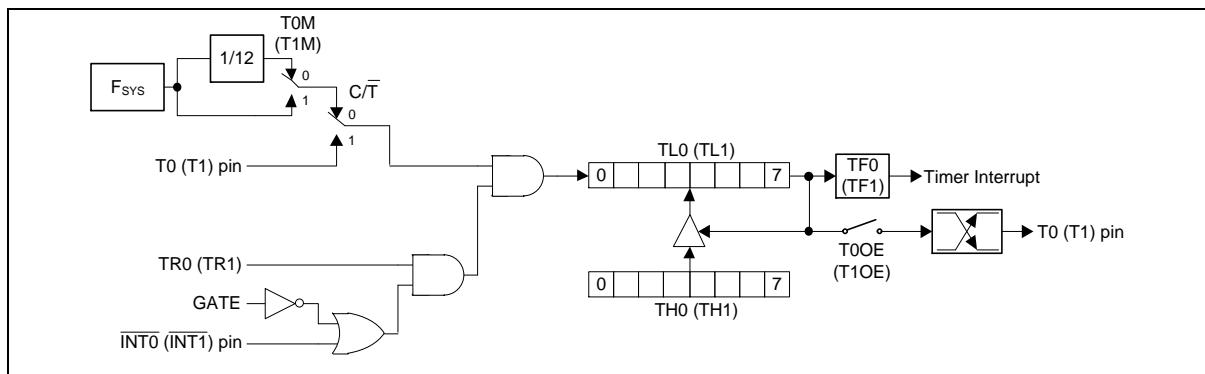


Figure 6.5-3 Timer/Counters 0 and 1 in Mode 2

6.5.1.5 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/\overline{T} , GATE, TR0, $\overline{\text{INT0}}$, and TF0. The TL0 also can be used as a 1-to-0 transition counter on pin T0 as determined by C/\overline{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, $\overline{\text{INT1}}$ pin and T1M. It can be used as a baud rate generator for the serial port or other application not requiring an interrupt.

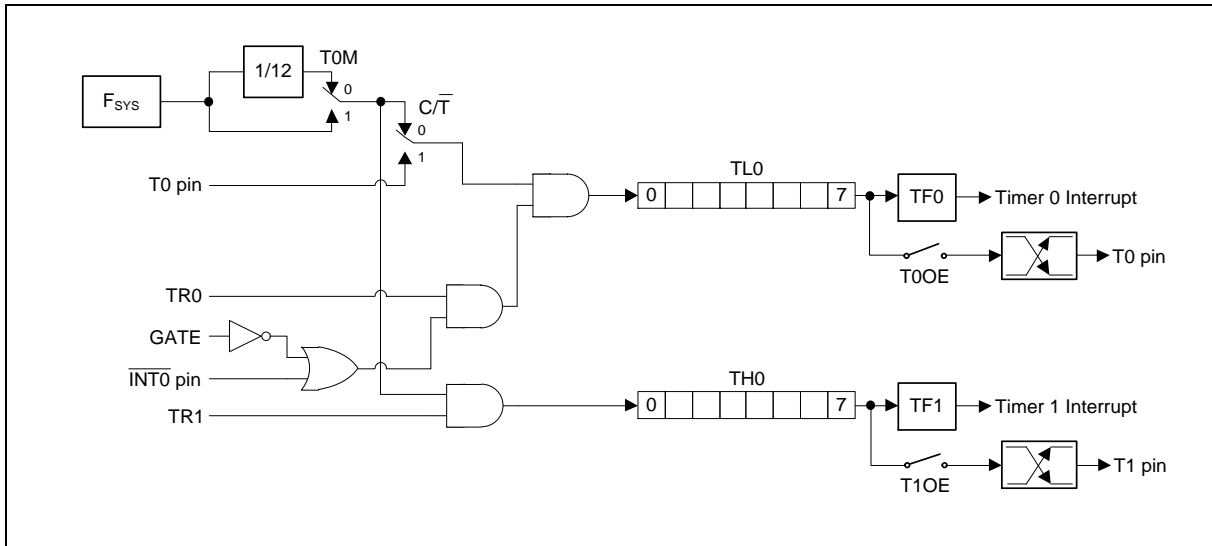


Figure 6.5-4 Timer/Counter 0 in Mode 3

6.5.1.6 Control Registers of I/O Ports

TMOD – Timer 0 and 1 Mode

Register	SFR Address	Reset Value
TMOD	89H, Page0	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 $\overline{INT1}$ logic level. 1 = Timer 1 will clock only when TR1 is 1 and $\overline{INT1}$ 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 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>M1</th> <th>M0</th> <th>Timer 1 Mode</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Mode 0: 13-bit Timer/Counter</td> </tr> <tr> <td>0</td> <td>1</td> <td>Mode 1: 16-bit Timer/Counter</td> </tr> <tr> <td>1</td> <td>0</td> <td>Mode 2: 8-bit Timer/Counter with auto-reload from TH1</td> </tr> <tr> <td>1</td> <td>1</td> <td>Mode 3: Timer 1 halted</td> </tr> </tbody> </table>	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
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																
3	GATE	Timer 0 gate control 0 = Timer 0 will clock when TR0 is 1 regardless of $\overline{INT0}$ logic level. 1 = Timer 0 will clock only when TR0 is 1 and $\overline{INT0}$ 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 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>M1</th> <th>M0</th> <th>Timer 0 Mode</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Mode 0: 13-bit Timer/Counter</td> </tr> <tr> <td>0</td> <td>1</td> <td>Mode 1: 16-bit Timer/Counter</td> </tr> <tr> <td>1</td> <td>0</td> <td>Mode 2: 8-bit Timer/Counter with auto-reload from TH0</td> </tr> <tr> <td>1</td> <td>1</td> <td>Mode 3: TL0 as a 8-bit Timer/Counter and TH0 as a 8-bit Timer</td> </tr> </tbody> </table>	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
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																

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 $\overline{INT1}$ input signal's logic level. Software cannot control it.
2	IT1	External interrupt 1 type select This bit selects by which type that $\overline{INT1}$ is triggered. 0 = $\overline{INT1}$ is low level triggered. 1 = $\overline{INT1}$ 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 $\overline{INT0}$ input signal's logic level. Software cannot control it.
0	IT0	External interrupt 0 type select This bit selects by which type that $\overline{INT0}$ is triggered. 0 = $\overline{INT0}$ is low level triggered. 1 = $\overline{INT0}$ is falling edge triggered.

TL0 – Timer 0 Low Byte

Register	SFR Address	Reset Value
TL0	8AH, Page0	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, Page0	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, Page0	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, Page0	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, Page0	0000_0000b

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

Bit	Name	Description
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.

6.5.2 Timer2 And Input Capture

6.5.2.1 Overview

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/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.

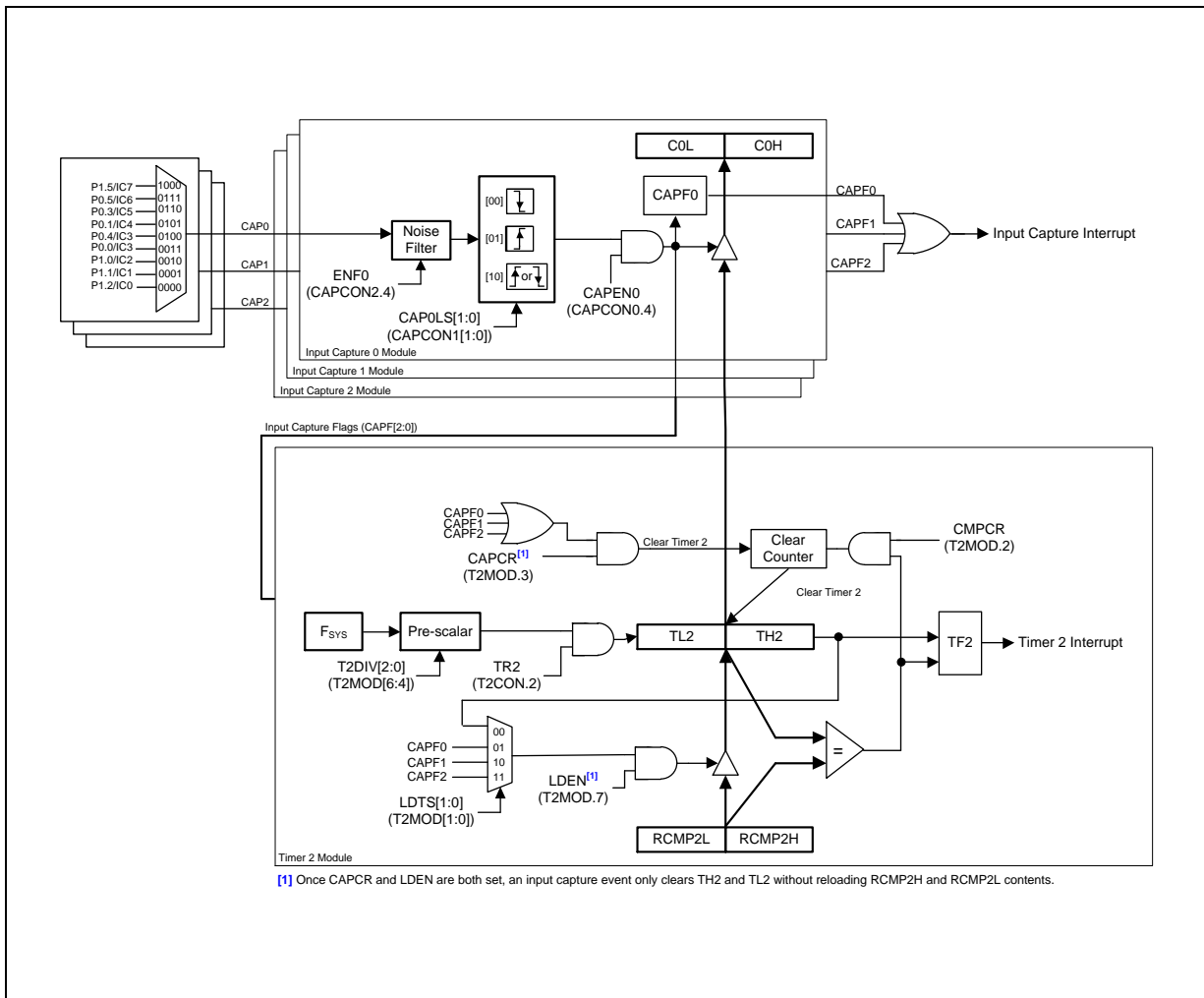


Figure 6.5-5 Timer 2 Block Diagram

6.5.2.2 Auto-Reload Mode

The Timer 2 is configured as auto-reload mode by clearing CM/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.

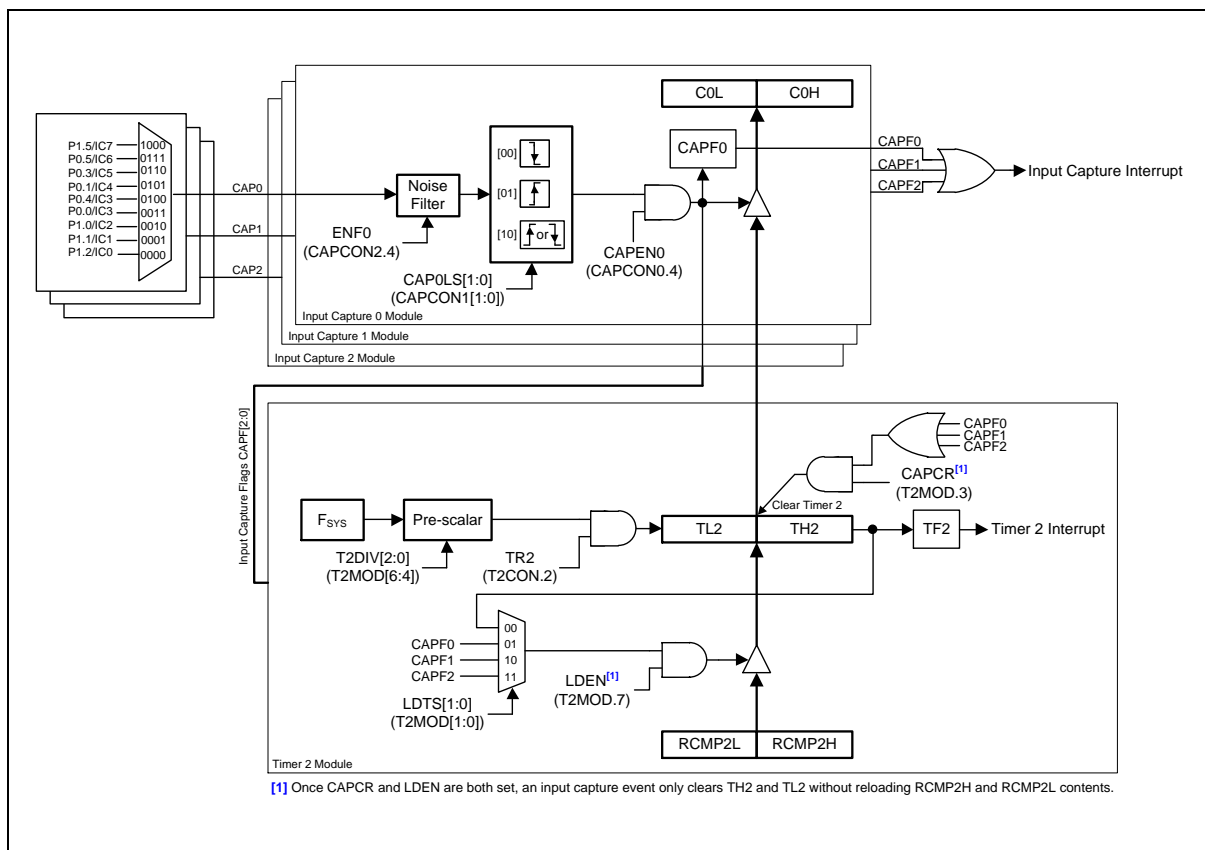


Figure 6.5-6 Timer 2 Auto-Reload Mode and Input Capture Module Functional Block Diagram

6.5.2.3 Compare Mode

Timer 2 can also be configured as the compare mode by setting CM/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.

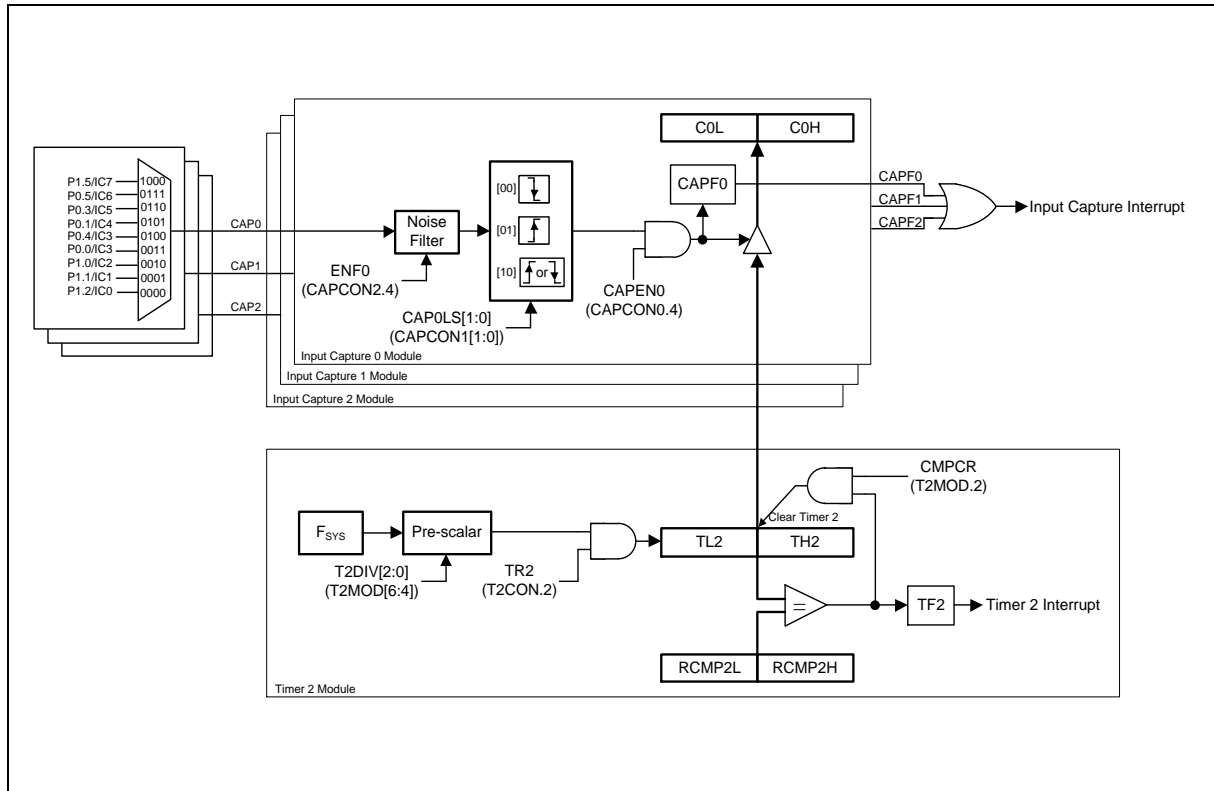


Figure 6.5-7 Timer 2 Compare Mode and Input Capture Module Functional Block Diagram

6.5.2.4 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.

6.5.2.5 Control Registers of Timer 2

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.
5:3	-	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.

RCMP2L– Timer 2 Reload/Compare Low Byte

Register	SFR Address	Reset Value
RCMP2L	CAH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
RCMP2L[7:0]							
R/W							

Bit	Name	Description
7:0	RCMP2L[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.

RCMP2H – Timer 2 Reload/Compare High Byte

Register	SFR Address	Reset Value
RCMP2H	CBH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
RCMP2H[7:0]							
R/W							

Bit	Name	Description
7:0	RCMP2H[7:0]	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
TL2[7:0]							
R/W							

Bit	Name	Description
7:0	TL2[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
TH2[7:0]							
R/W							

Bit	Name	Description
7:0	TH2[7:0]	Timer 2 high byte The TH2 register is the high byte of the 16-bit counting register of Timer 2.

Note that the TH2 and TL2 are accessed separately. It is strongly recommended that user stops Timer 2 temporarily by clearing TR2 bit before reading from or writing to TH2 and TL2. The free-running reading or writing may cause unpredictable result.

CAPCON0 – Input Capture Control 0

Register	SFR Address	Reset Value
CAPCON0	92H, Page 0	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 be 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 be 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 be cleared by software.

CAPCON1 – Input Capture Control 1

Register	SFR Address	Reset Value
CAPCON1	93H, Page 0	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.

CAPCON2 – Input Capture Control 2

Register	SFR Address	Reset Value
CAPCON2	94H, Page0	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 filter 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 filter 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 filter 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 0	0000_0000 b
C1L	E6H, Page 0	0000_0000 b
C2L	EDH, Page 0	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 0	0000_0000 b
C1H	E7H, Page 0	0000_0000 b
C2H	EEH, Page 0	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.

CAPCON3 – Input Capture Control 3

Register	SFR Address	Reset Value
CAPCON3	F1H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
CAP13	CAP12	CAP11	CAP10	CAP03	CAP02	CAP01	CAP00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
[7:4]	CAP1[3:0]	Input capture channel 1 input pin select 0000 = P1.2/IC0 0001 = P1.1/IC1 0010 = P1.0/IC2 0011 = P0.0/IC3 0100 = P0.4/IC3 0101 = P0.1/IC4 0110 = P0.3/IC5 0111 = P0.5/IC6 1000 = P1.5/IC7 others = P1.2/IC0
[3:0]	CAP0[3:0]	Input capture channel 0 input pin select 0000 = P1.2/IC0 0001 = P1.1/IC1 0010 = P1.0/IC2 0011 = P0.0/IC3 0100 = P0.4/IC3 0101 = P0.1/IC4 0110 = P0.3/IC5 0111 = P0.5/IC6 1000 = P1.5/IC7 others = P1.2/IC0

CAPCON4 – Input Capture Control 4

Register	SFR Address	Reset Value
CAPCON4	F2H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	CAP23	CAP22	CAP21	CAP20
-	-	-	-	R/W	R/W	R/W	R/W

Bit	Name	Description
[3:0]	CAP2[3:0]	Input capture channel 2 input pin select 0000 = P1.2/IC0 0001 = P1.1/IC1 0010 = P1.0/IC2 0011 = P0.0/IC3 0100 = P0.4/IC3 0101 = P0.1/IC4 0110 = P0.3/IC5 0111 = P0.5/IC6 1000 = P1.5/IC7 others = P1.2/IC0

6.5.3 TIMER 3

6.5.3.1 Overview

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 6.9.3“Baud Rate”](#).

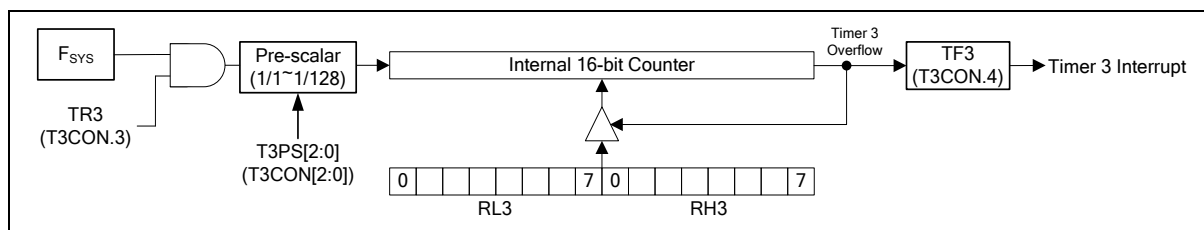


Figure 6.5-8 Timer 3 Block Diagram

6.5.3.2 Control Registers of Timer 3

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.

Bit	Name	Description
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.

6.6 Watchdog Timer (WDT)

6.6.1 Overview

The MS51 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 10 kHz oscillator. The following table shows an example of the Watchdog time-out interval with different pre-scalers.

WDPS.2	WDPS.1	WDPS.0	Clock Divider Scale	WDT Time-Out Timing ⁽¹⁾
0	0	0	1/1	6.40 ms
0	0	1	1/4	25.60 ms
0	1	0	1/8	51.20 ms
0	1	1	1/16	102.40 ms
1	0	0	1/32	204.80 ms
1	0	1	1/64	409.60 ms
1	1	0	1/128	819.20 ms
1	1	1	1/256	1.638 s

Note: This is an approximate value since the deviation of LIRC.

Table 6.6-1 Watchdog Timer-out Interval Under Different Pre-scalars

Since the limitation of the maxima vaule of WDT timer delay. To up MS51 from idle mode or power down mode suggest use WKT function see Chapter 6.6.4 Control Registers of Watchdog Timer

6.6.2 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.

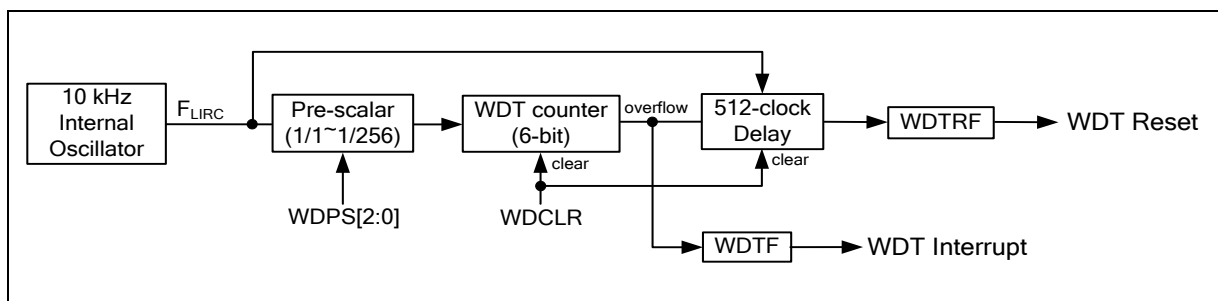


Figure 6.6-1 WDT as A Time-Out Reset Timer

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.

6.6.3 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.

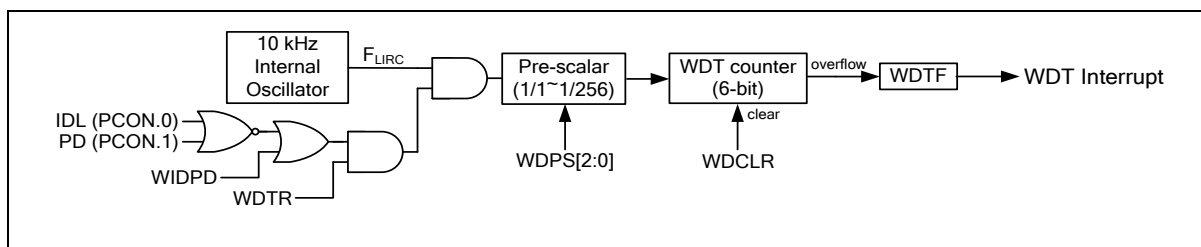


Figure 6.6-2 Watchdog Timer Block Diagram

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.

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 MS51 is equipped with this useful function by WDT waking up. It provides a very low power internal oscillator 10 kHz 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

```

6.6.4 Control Registers of Watchdog Timer

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]	<p>WDT enable</p> <p>This field configures the WDT behavior after MCU execution.</p> <p>1111 = WDT is Disabled. WDT can be used as a general purpose timer via software control.</p> <p>0101 = WDT is Enabled as a time-out reset timer and it stops running during Idle or Power-down mode.</p> <p>Others = WDT is Enabled as a time-out reset timer and it keeps running during Idle or Power-down mode.</p>

The WDT is implemented with a set of divider that divides the low-speed internal oscillator clock nominal 10 kHz. 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.

WDCON – Watchdog Timer Control

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 WDTE[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. <u>Writing:</u> 0 = No effect. 1 = Clearing WDT counter. <u>Reading:</u> 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 WDTE[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 6.6-1 . The default is the maximum pre-scale value.

Note:

1. WDTRF will be cleared after power-on reset, be set after WDT reset, and remains unchanged after any other resets.
2. WDPS[2:0] are all set after power-on reset and keep unchanged after any reset other than power-on reset.

6.7 Self Wake-Up Timer (WKT)

6.7.1 Overview

The MS51 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 one clock source, internal 10 kHz. 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 8-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 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.

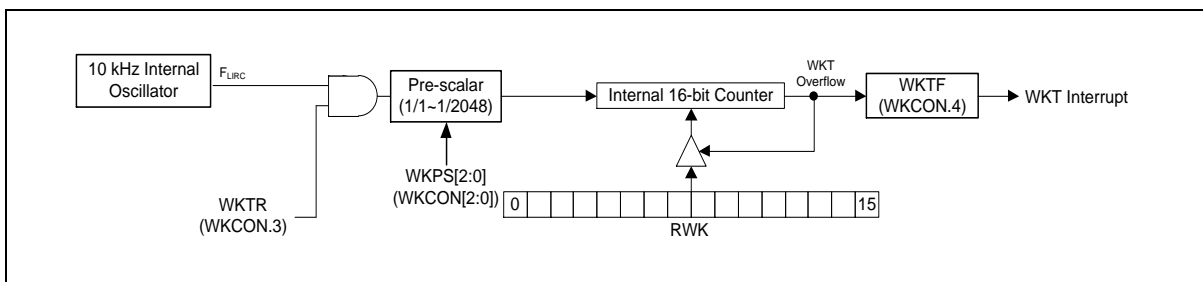


Figure 6.7-1 Self Wake-Up Timer Block Diagram

6.7.2 Control register of WKT

WKCON – Self Wake-up Timer Control

Register	SFR Address	Reset Value
WKCON	8FH, Page0	0000_0000b

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

Bit	Name	Description
7:5	-	Reserved
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.

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 The RWKL register is the low byte of the 16-bit reload value of WKT. Note that RWK should not be FFFFH if the pre-scale is 1/1 for implement limitation.

RWKH – Self Wake-up Timer Reload High Byte

Register	SFR Address	Reset Value
RWKH	97H, 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 The RWKH register is the low byte of the 16-bit reload value of WKT. Note that RWK should not be FFFFH if the pre-scale is 1/1 for implement limitation.

6.8 Pulse Width Modulated (PWM)

6.8.1 Overview

The PWM (Pulse Width Modulation) signal is a useful control solution in wide application field. It can be 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 MS51 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 MS51 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.

6.8.2 Basic Configuration

PWM output pin define and enable control register table.

PWM Channel	Output Pin	Control register 1			Control register2		
		SFR Byte Name	Bit name	Value	SFR Byte Name	Bit name	Value
PWM0_CH0	P1.2	PIOCON0	PIO12	1	AUXR4[1:0]	PWM1C0P	00
	P3.3	PIOCON2	PIO33	1	-	-	-
PWM0_CH1	P1.1	PIOCON0	PIO11	1	AUXR4[3:2]	PWM1C1P	00
	P1.4	PIOCON1	PIO14	1	AUXR4[3:2]	PWM1C1P	00
PWM0_CH2	P0.5	PIOCON1	PIO05	1	AUXR4[5:4]	PWM2C0P	00
	P1.0	PIOCON0	PIO10	1	-	-	-
PWM0_CH3	P0.4	PIOCON1	PIO04	1	AUXR4[7:6]	PWM2C1P	00
	P0.0	PIOCON0	PIO00	1	AUXR4[7:6]	PWM2C1P	00
PWM0_CH4	P0.1	PIOCON0	PIO01	1	AUXR5[1:0]	PWM3C0P	00
PWM0_CH5	P0.3	PIOCON0	PIO03	1	AUXR5[3:2]	PWM3C1P	00
	P1.5	PIOCON1	PIO15	1	AUXR5[3:2]	PWM3C1P	00
PWM1_CH0	P2.3	PIOCON2	PIO23	1	AUXR4[1:0]	PWM1C0P	01
	P1.2	PIOCON0	PIO12	1			10
PWM1_CH1	P2.2	PIOCON2	PIO22	1	AUXR4[3:2]	PWM1C1P	01
	P1.4	PIOCON1	PIO14	1			10
	P1.1	PIOCON0	PIO11	1			11
PWM2_CH0	P2.1	PIOCON2	PIO21	1	AUXR4[5:4]	PWM2C0P	00
	P1.0	PIOCON0	PIO10	1			01
	P0.5	PIOCON1	PIO05	1			10
PWM2_CH1	P3.0	PIOCON2	PIO30	1	AUXR4[7:6]	PWM2C1P	00
	P3.1	PIOCON2	PIO31	1			01
	P0.0	PIOCON0	PIO00	1			10

PWM Channel	Output Pin	Control register 1			Control register2		
		SFR Byte Name	Bit name	Value	SFR Byte Name	Bit name	Value
	P0.4	PIOCON1	PIO04	1			11
PWM3_CH0	P3.2	PIOCON2	PIO32	1	AUXR5[1:0]	PWM3C0P	01
	P0.1	PIOCON0	PIO01	1			10
	P1.7	PIOCON1	PIO17	1			-
PWM3_CH1	P3.4	PIOCON2	PIO34	1	AUXR5[3:2]	PWM3C1P	01
	P1.5	PIOCON1	PIO15	1			10
	P0.3	PIOCON0	PIO03	1			11

Table 6.8-1 PWM Pin Define And Enable Control Register

6.8.4 Functional Description

6.8.4.1 PWM Generator

The PWM generator is clocked by the system clock or Timer 1 overflow divided by a PWM clock pre-scaler 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 {PWM0_CH5H, PWM0_CH5L} registers have no effect. This mean is {PWM0C2H, PWM0C2L} and {PWM0C4H, PWM0C4L} both as same as {PWM0C0H, PWM0C0L}. Also {PWM0C3H, PWM0C3L} and {PWM0_CH5H, PWM0_CH5L} 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.

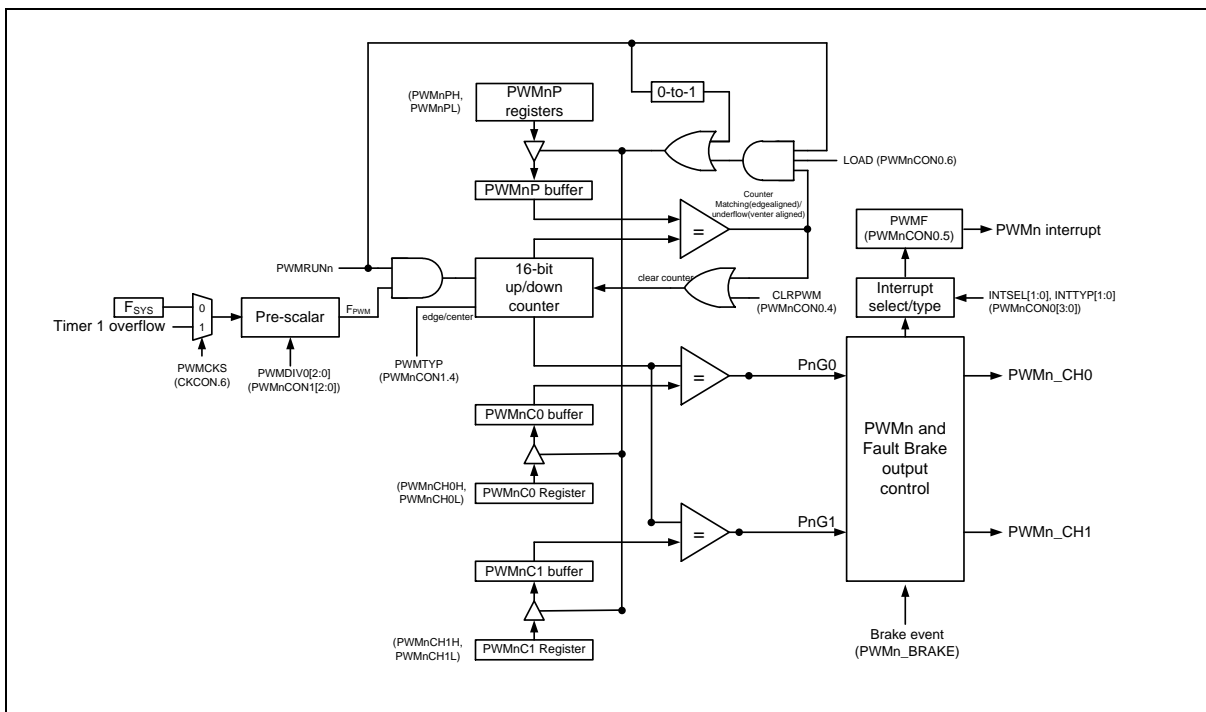


Figure 6.8-2 PWM1/2/3 Block Diagram

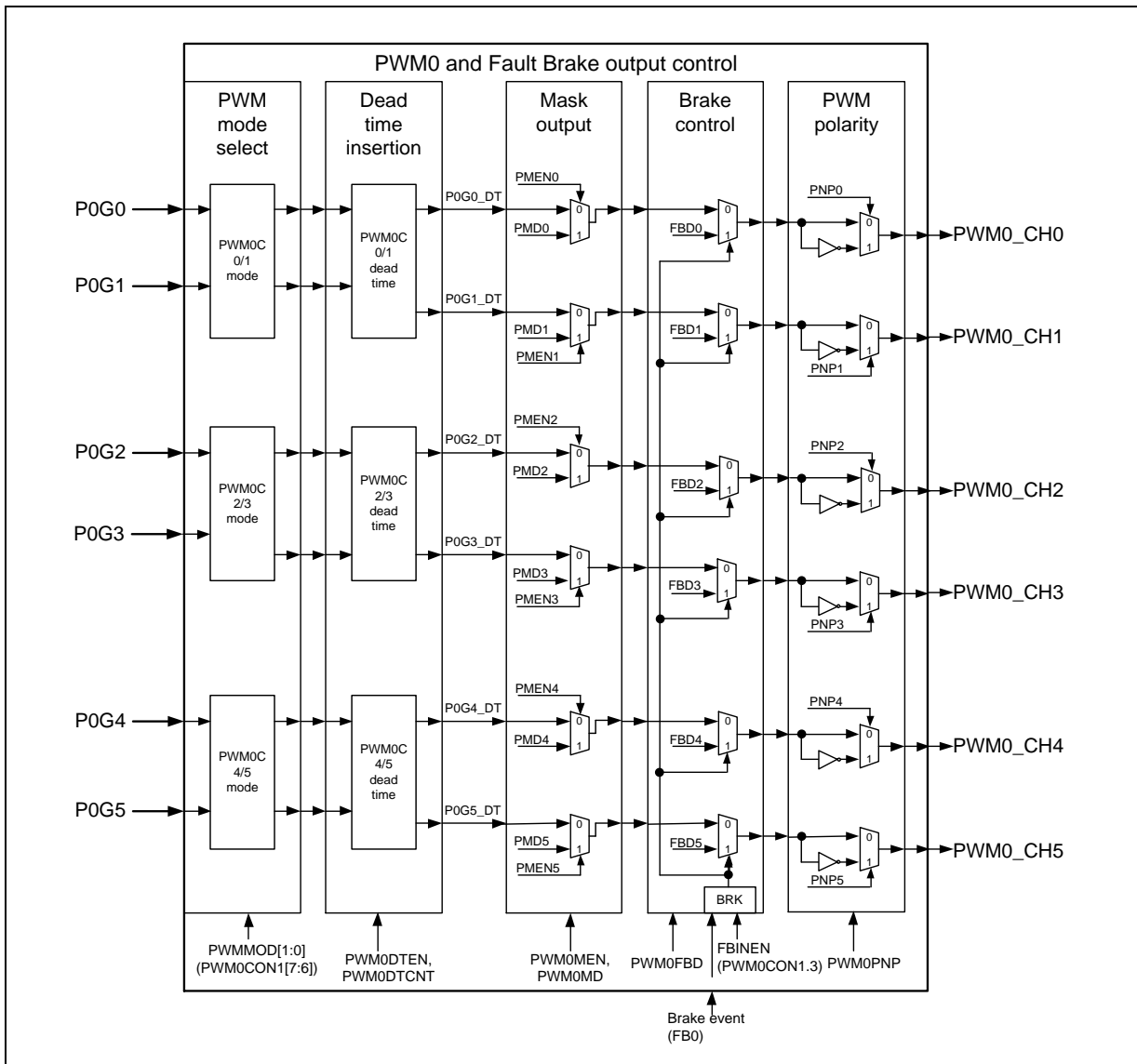


Figure 6.8-3 PWM0 and Fault Brake Output Control Block Diagram

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 {PWMnCxH, PWMnCxL} registers, PWMnRUN 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 control registers are double buffered designed. Therefore, {PWMnPH, PWMnPL} and all {PWMnCxH, PWMnCxL} 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.

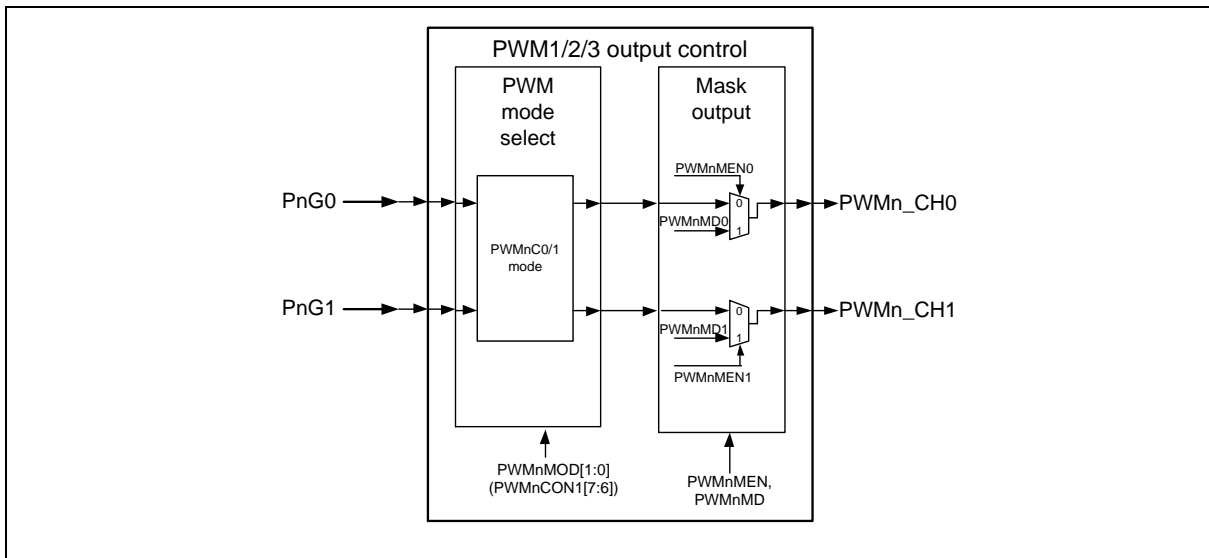


Figure 6.8-4 PWM1/2/3 Control Block Diagram

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 control registers during LOAD bit as logic 1 will cause unpredictable output.

6.8.4.2 PWM Types

The PWM generator provides two PWM types: edge-aligned or center-aligned. PWM type is selected by PWMTYP (PWMnCON1.4).

6.8.4.3 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 (PnGx before PWM and Fault Brake output control) is cleared on the compare match of 16-bit counter and the duty register {PWMnCxH, PWMnCxL} and set at the 16-bit counter is 0000H. The result PWM output waveform is left-edge aligned.

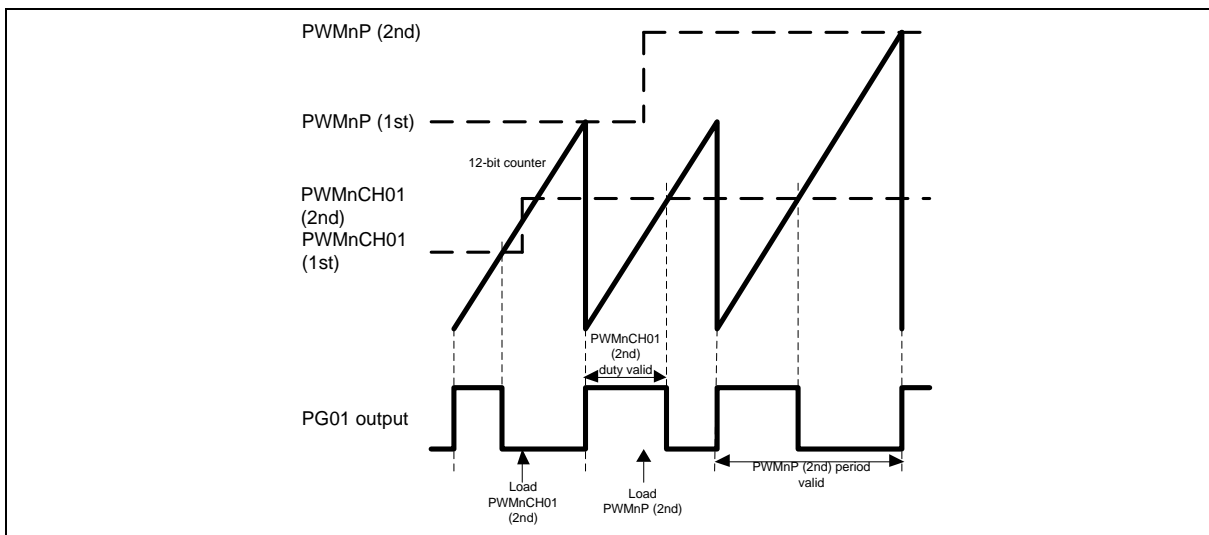


Figure 6.8-5 PWM Edge-aligned Type Waveform

The output frequency and duty cycle for edge-aligned PWM are given by following equations:

$$PWM \text{ frequency} = \frac{F_{PWM}}{\{PWMnPH, PWMnPL\} + 1}$$

(F_{PWM} is the PWM clock source frequency divided by PWMDIV).

$$PWM \text{ high level duty} = \frac{\{PWMnCHxH, PWMnCHxL\}}{\{PWMnPH, PWMnPL\} + 1}$$

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 PnGx signal is cleared on the up-count compare match of 16-bit counter and the duty register {PWMnCxH, PWMnCxL} and set on the down-count compare match. Center-aligned PWM may be used to generate non-overlapping waveforms.

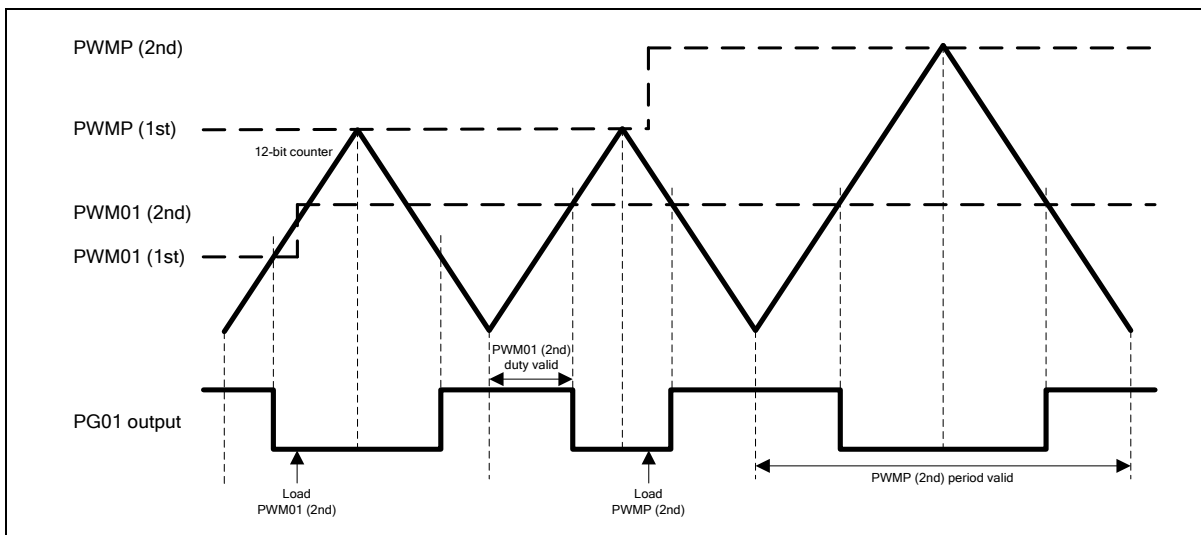


Figure 6.8-6 PWM Center-aligned Type Waveform

The output frequency and duty cycle for center-aligned PWM are given by following equations:

$$PWM \text{ frequency} = \frac{F_{PWM}}{2 \times \{PWMnPH, PWMnPL\}}$$

(F_{PWM} is the PWM clock source frequency divided by PWMDIV).

$$PWM \text{ high level duty} = \frac{\{PWMnCHxH, PWMnCHxL\}}{\{PWMnPH, PWMnPL\}}$$

6.8.4.4 Operation Modes

After PnGx signals pass through the first stage of the PWM. The PWM mode selection circuit generates different kind of PWM output modes. It supports independent mode, complementary mode, and synchronous mode.

Independent Mode

Independent mode is enabled when PWMMOD[1:0] (PWMnCON1[7:6]) is [0:0]. It is the default mode of PWM. PnGx output PWM signals independently.

Complementary Mode with Dead-Time Insertion

Complementary mode is enabled when PWMMOD[1:0] = [0:1]. In this mode, PnG0/2/4 output PWM signals the same as the independent mode. However, PnG1/3/5 output the out-phase PWM signals of PnG0/2/4 correspondingly, and ignore PnG1/3/5 Duty register {PWMnH, PWMnL} (n:1/3/5). This mode makes PnG0/PnG1 a PWM complementary pair and so on P0G2/P0G3 and P0G4/P0G5.

In a real motor application, a complementary PWM0 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 MS51 PWM0, each PWM0 pair share a 9-bit dead-time down-counter PWM0DTCNT used to produce the off time between two PWM0 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 P0G0/P0G1 pair. Pairs of P0G2/P0G3 and P0G4/P0G5 have the same dead-time circuit. Each pair has its own dead-time enabling bit in the field of PWM0DTEN [3:0].

Note that the PWM0DTCNT and PWM0DTEN registers are all TA write protection. The dead-time control are also valid only when the PWM0 is configured in its complementary mode. Dead-Time Insertion is only valid in PWM0.

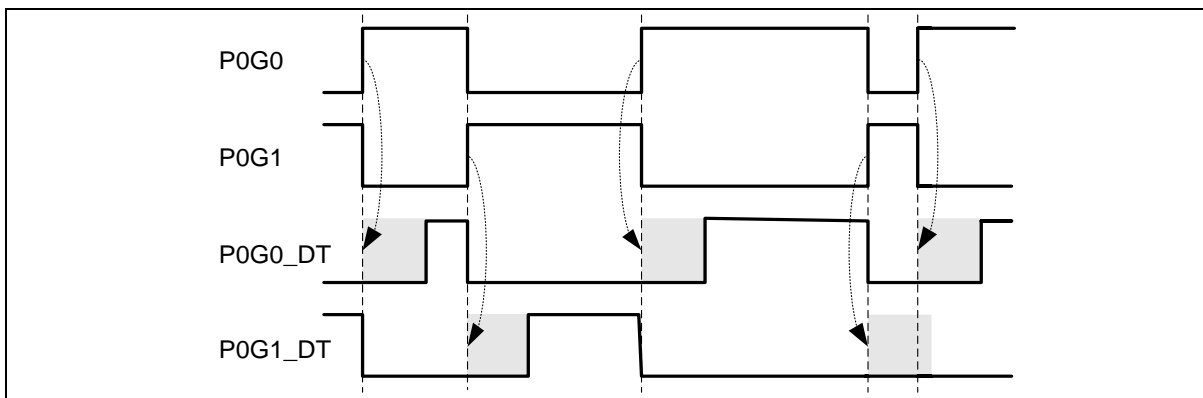


Figure 6.8-7 PWM0 Complementary Mode with Dead-time Insertion

Synchronous Mode

Synchronous mode is enabled when PWMMOD[1:0] = [1:0]. In this mode, PnG0/2/4 output PWM signals the same as the independent mode. PnG1/3/5 output just the same in-phase PWM signals of PnG0/2/4 correspondingly.

6.8.4.5 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 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.

6.8.4.6 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. PWM0RUN (PWM0CON0.7) 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 PWM0RUN 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.

Note that the Fault Brake function are only valid in PWM0.

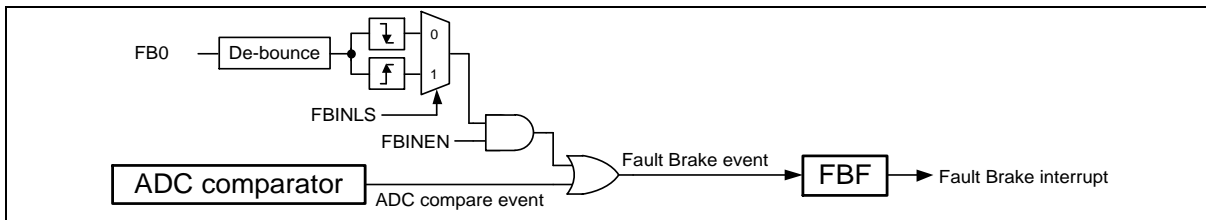


Figure 6.8-8 Fault Brake Function Block Diagram

6.8.4.7 Polarity Control

Each PWM0 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 PNP bit to make PWM0 actually outputs according to the negative logic.

Note that the polarity control function are only valid in PWM0.

6.8.4.8 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.

The PWM interrupt related with PWM waveform is shown as figure below.

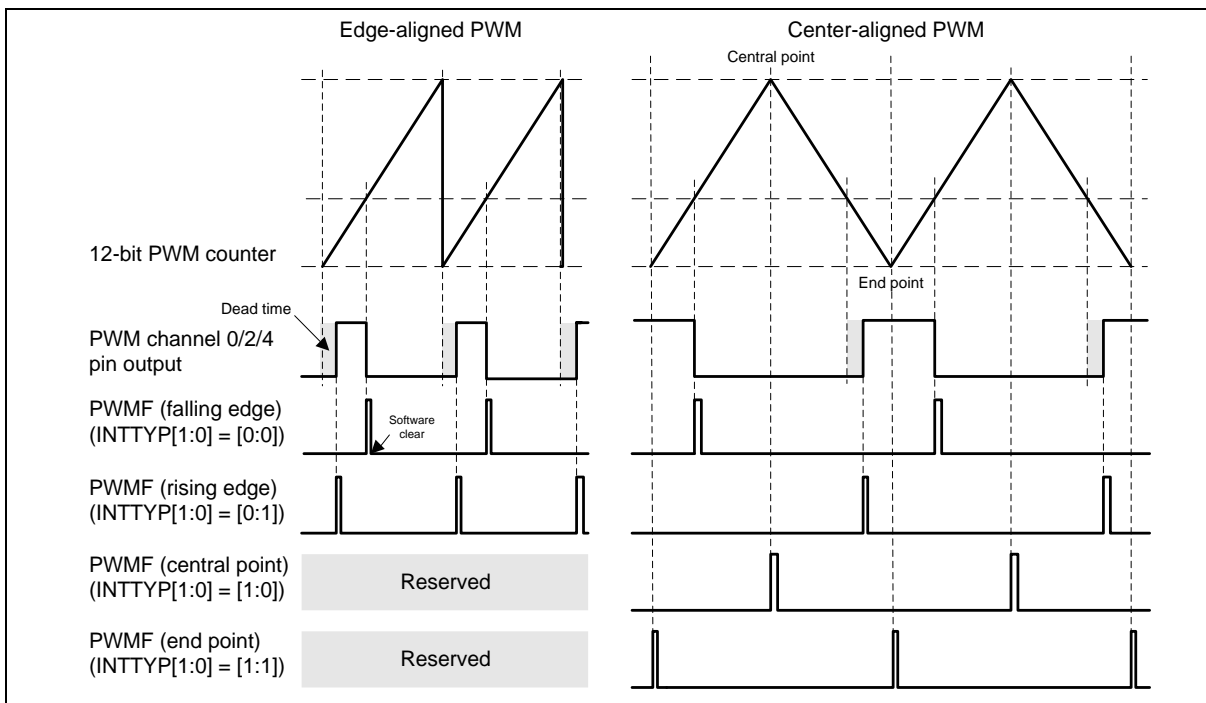


Figure 6.8-9 PWM Interrupt Type

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 (PWM0FBD.7) will be set by hardware. It generates Fault Brake interrupt if enabled. The Fault Brake interrupt enable bit is EFBO (EIE0.5). FBF is cleared via software.

6.8.5 Control Register of PWM

PWM0CON0 – PWM0 Control 0

Register	SFR Address	Reset Value
PWM0CON0	D8H, Page0, Bit-addressable	0000_0000 b

7	6	5	4	3	2	1	0
PWM0RUN	LOAD	PWMF	CLRPWM	PWM3RUN	PWM2RUN	PWM1RUN	P33FBINEN
R/W	R/W	R/W	R/W	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. Note: This bit is only for PWM0CON0.
6	LOAD	PWM new period and duty load This bit is used to load period and duty control registers 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. <u>Writing:</u> 0 = No effect. 1 = Load new period and duty in their buffers while a PWM period is completed. <u>Reading:</u> 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 PWM0INTC. 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. <u>Writing:</u> 0 = No effect. 1 = Clearing PWM 16-bit counter. <u>Reading:</u> 0 = PWM 16-bit counter is completely cleared. 1 = PWM 16-bit counter is not yet cleared.
3	PWM3RUN	PWM3 run enable 0 = PWM3 stays in idle. 1 = PWM3 starts running. Note: This bit is only for PWM0CON0.
2	PWM2RUN	PWM2 run enable 0 = PWM2 stays in idle. 1 = PWM2 starts running. Note: This bit is only for PWM0CON0.

Bit	Name	Description
1	PWM1RUN	<p>PWM1 run enable 0 = PWM1 stays in idle. 1 = PWM1 starts running. Note: This bit is only for PWM0CON0.</p>
0	P33FBINEN	<p>P33 FB pin input enable 0 = PWM0 output Fault Braked by P33 FB pin input Disabled. 1 = PWM0 output Fault Braked by P33 FB pin input Enabled. Once an edge, which matches FBINLS (FBD.6) selection, occurs on FB pin, PWM0C0~C5 output Fault Brake data in FBD register. PWMRUN (PWM0CON0.7) will also be automatically cleared by hardware. The PWM output resumes when PWMRUN is set again. Note: This bit is only for PWM0CON0.</p>

PWMnCON0 – PWMn Control 0, n = 1,2,3

Register	SFR Address	Reset Value
PWM1CON0	B4H, 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
PWMRUN	LOAD	PWMF	CLRPWM	-	-	-	-
R/W	R/W	R/W	R/W	-	-	-	-

Bit	Name	Description
7	PWMRUN	PWM 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 control registers 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. <u>Writing:</u> 0 = No effect. 1 = Load new period and duty in their buffers while a PWM period is completed. <u>Reading:</u> 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 PWM0INTC. 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. <u>Writing:</u> 0 = No effect. 1 = Clearing PWM 16-bit counter. <u>Reading:</u> 0 = PWM 16-bit counter is completely cleared. 1 = PWM 16-bit counter is not yet cleared.

PWMnCON1 – PWM Control 1

Register	SFR Address	Reset Value
PWM0CON1	DFH, page 0	0000_0000 b
PWM1CON1	B5H, 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]	<p>PWM mode select</p> <p>00 = Independent mode.</p> <p>01 = Complementary mode.</p> <p>10 = Synchronized mode.</p> <p>11 = Reserved.</p>
5	GP	<p>Group mode enable</p> <p>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 control registers.</p> <p>0 = Group mode Disabled.</p> <p>1 = Group mode Enabled.</p> <p>Note: This bit is only valid in PWM0</p>
4	PWMTYP	<p>PWM type select</p> <p>0 = Edge-aligned PWM.</p> <p>1 = Center-aligned PWM.</p>
3	FBINEN	<p>FB pin input enable</p> <p>0 = PWM0 output Fault Braked by FB pin input Disabled.</p> <p>1 = PWM0 output Fault Braked by FB pin input Enabled. Once an edge, which matches FBINLS (PWM0FBD.6) selection, occurs on FB pin, PWM0 channel 0–5 output Fault Brake data in PWM0FBD register. PWM0RUN (PWM0CON0.7) will also be automatically cleared by hardware. The PWM0 output resumes when PWM0RUN is set again.</p> <p>Note: This bit is only valid in PWM0.</p>
2:0	PWMDIV[2:0]	<p>PWM clock divider</p> <p>This field decides the pre-scale of PWM clock source.</p> <p>000 = 1/1.</p> <p>001 = 1/2</p> <p>010 = 1/4.</p> <p>011 = 1/8.</p> <p>100 = 1/16.</p> <p>101 = 1/32.</p> <p>110 = 1/64.</p> <p>111 = 1/128.</p>

CKCON – Clock Control

Register	SFR Address	Reset Value
CKCON	8EH, Page0	0000_0000b

7	6	5	4	3	2	1	0
-	PWMCKS	-	T1M	T0M	T0OE	CLOEN	-
-	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, n = 0,1,2,3

Register	SFR Address	Reset Value
PWM0PL	D9H, Page 0	0000_0000 b
PWM1PL	B1H, 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
PWM0P[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, n = 0,1,2,3

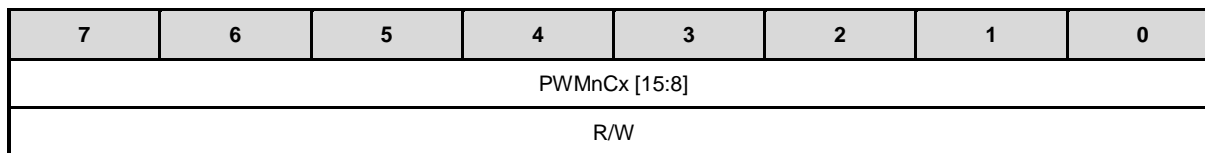
Register	SFR Address	Reset Value
PWM0PH	D1H, Page 0	0000_0000 b
PWM1PH	A9H, Page 2	0000_0000 b
PWM2PH	B9H, Page 2	0000_0000 b
PWM3PH	C9H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PWM0P[15:8]							
R/W							

Bit	Name	Description
7:0	PWM0P[15:8]	PWM period high byte This byte with PWM0PL controls the period of the PWM generator signal.

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

Register	SFR Address	Description	Reset Value
PWM0C0H	D2H, Page 0	PWM0 Channel 0 Duty High Byte	0000_0000 b
PWM0C1H	D3H, Page 0	PWM0 Channel 1 Duty High Byte	0000_0000 b
PWM0C2H	D4H, Page 0	PWM0 Channel 2 Duty High Byte	0000_0000 b
PWM0C3H	D5H, Page 0	PWM0 Channel 3 Duty High Byte	0000_0000 b
PWM0C4H	C4H, Page 1	PWM0 Channel 4 Duty High Byte	0000_0000 b
PWM0C5H	C5H, Page 1	PWM0 Channel 5 Duty High Byte	0000_0000 b
PWM1C0H	AAH, Page 2	PWM1 Channel 0 Duty High Byte	0000_0000 b
PWM1C1H	ABH, Page 2	PWM1 Channel 1 Duty High Byte	0000_0000 b
PWM2C0H	BAH, Page 2	PWM2 Channel 0 Duty High Byte	0000_0000 b
PWM2C1H	BBH, Page 2	PWM2 Channel 1 Duty High Byte	0000_0000 b
PWM3C0H	CAH, Page 2	PWM3 Channel 0 Duty High Byte	0000_0000 b
PWM3C1H	CBH, Page 2	PWM3 Channel 1 Duty High Byte	0000_0000 b



Bit	Name	Description
7:0	PWMnCx [15:8]	PWMnCx duty high byte This byte with PWMnCxL controls the duty of the output signal PGx from PWM generator.

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

Register	SFR Address	Description	Reset Value
PWM0C0L	DAH, Page 0	PWM0 Channel 0 Duty Low Byte	0000_0000 b
PWM0C1L	DBH, Page 0	PWM0 Channel 1 Duty Low Byte	0000_0000 b
PWM0C2L	DCH, Page 0	PWM0 Channel 2 Duty Low Byte	0000_0000 b
PWM0C3L	DDH, Page 0	PWM0 Channel 3 Duty Low Byte	0000_0000 b
PWM0C4L	CCH, Page 1	PWM0 Channel 4 Duty Low Byte	0000_0000 b
PWM0C5L	CDH, Page 1	PWM0 Channel 5 Duty Low Byte	0000_0000 b
PWM1C0L	B2H, Page 2	PWM1 Channel 0 Duty Low Byte	0000_0000 b
PWM1C1L	B3H, Page 2	PWM1 Channel 1 Duty Low Byte	0000_0000 b
PWM2C0L	C2H, Page 2	PWM2 Channel 0 Duty Low Byte	0000_0000 b
PWM2C1L	C3H, Page 2	PWM2 Channel 1 Duty Low Byte	0000_0000 b
PWM3C0L	D2H, Page 2	PWM3 Channel 0 Duty Low Byte	0000_0000 b
PWM3C1L	D3H, Page 2	PWM3 Channel 1 Duty Low Byte	0000_0000 b

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

Bit	Name	Description
7:0	PWMnCx [7:0]	PWMnCx duty Low byte This byte with PWMnCxH controls the duty of the output signal PGx from PWM generator.

PWM0DTEN – PWM Dead-time Enable

Register	SFR Address	Reset Value
PWM0DTEN	F9H, Page 0, TA protected	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	PWM0DTCNT. 8	-	PDT45EN	PDT23EN	PDT01EN
-	-	-	R/W	-	R/W	R/W	R/W

Bit	Name	Description
4	PWM0DTCNT.8	PWM dead-time counter bit 8 See PWM0DTCNT register.
2	PDT45EN	PWM0_CH4 / PWM0_CH5 pair dead-time insertion enable This bit is valid only when PWM0_CH4 / PWM0_CH5 is under complementary mode. 0 = No delay on P0G4/P0G5 pair signals. 1 = Insert dead-time delay on the rising edge of P0G4/P0G5 pair signals.
1	PDT23EN	PWM0_CH2 / PWM0_CH3 pair dead-time insertion enable This bit is valid only when PWM0_CH2 / PWM0_CH3 is under complementary mode. 0 = No delay on P0G2/P0G3 pair signals. 1 = Insert dead-time delay on the rising edge of P0G2/P0G3 pair signals.
0	PDT01EN	PWM0_CH0 / PWM0_CH1 pair dead-time insertion enable This bit is valid only when PWM0_CH0 / PWM0_CH1 is under complementary mode. 0 = No delay on P0G0/P0G1 pair signals. 1 = Insert dead-time delay on the rising edge of P0G0/P0G1 pair signals.

PWM0DTCNT – PWM Dead-time Counter

Register	SFR Address	Reset Value
PWM0DTCNT	FAH, Page 0, 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]	<p>PWM0 dead-time counter low byte</p> <p>This 8-bit field combined with PWM0DTCNT.4 forms a 9-bit PWM0 dead-time counter PWM0DTCNT. This counter is valid only when PWM0 is under complementary mode and the correspond PWM0DTCNT bit for PWM pair is set.</p> $\text{PWM0 dead-time} = \frac{\text{PDTCNT} + 1}{F_{\text{SYS}}}$ <p>Note that user should not modify PWM0DTCNT during PWM0 run time.</p>

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

Register	SFR Address	Description	Reset Value
PWM0MEN	FBH, Page 0	PWM0_CHn Mask Enable	0000_0000 b
PWM1MEN	ADH, Page 2	PWM1_CHn Mask Enable	0000_0000 b
PWM2MEN	BDH, Page 2	PWM2_CHn Mask Enable	0000_0000 b
PWM3MEN	CDH, Page 2	PWM3_CHn Mask Enable	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
n	PMENn	PWMn_CHn mask enable 0 = PWMn_CHn signal outputs from its PWM generator. 1 = PWMn_CHn signal is masked by PMDx. Note: PMEN2~5 are only for PWM0.

PWMnMD –PWMnCx Mask Data

Register	SFR Address	Description	Reset Value
PWM0MD	FCH, Page 0	PWM0_CHn Mask Data	0000_0000 b
PWM1MD	ACH, Page 2	PWM1_CHn Mask Data	0000_0000 b
PWM2MD	BCH, Page 2	PWM2_CHn Mask Data	0000_0000 b
PWM3MD	CCH, Page 2	PWM3_CHn Mask Data	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
n	PMDn	<p>PWMn_CHn mask data</p> <p>The PWMn_CHn signal outputs mask data once its corresponding PMENx is set.</p> <p>0 = PWMn_CHn signal is masked by 0.</p> <p>1 = PWMn_CHn signal is masked by 1.</p> <p>Note: PMD2~5 are only for PWM0.</p>

PWM0FBD – PWM Fault Brake Data

Register	SFR Address	Reset Value
PWM0FBD	D7H, Page 0	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	FB pin input level selection 0 = Falling edge. 1 = Rising edge.
5:0	FBDx	PWM0 Fault Brake data 0 = PWM0_CHn signal is overwritten by 0 once Fault Brake asserted. 1 = PWM0_CHn signal is overwritten by 1 once Fault Brake asserted.

PNP – PWM Negative Polarity

Register	SFR Address	Reset Value
PNP	D6H, Page 0	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	PWM0_CHn negative polarity output enable 0 = PWM0_CHn signal outputs directly on PWM0_CHn pin. 1 = PWM0_CHn signal outputs inversely on PWM0_CHn pin.

PIOCON0 – PWM or I/O Select

Register	SFR Address	Reset Value
PIOCON0	DEH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	PIO05	PIO04	PIO03	PIO02	PIO01	PIO00
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
7:6	-	Reserved.
5	PIO03	P0.3/PWM pin function select 0 = P0.3/PWM pin functions as P0.3. 1 = P0.3/PWM pin functions as PWM0_CH5 / PWM3_CH1 output. When AUXR5[7:6]. PWM3CH1P = 00, PIO03 = 1, P0.3 defined as PWM0_CH5. AUXR5[7:6]. PWM3CH1P = 11, PIO03 = 1, P0.3 defined as PWM3_CH1.
4	PIO01	P0.1/PWM pin function select 0 = P0.1/PWM pin functions as P0.1. 1 = P0.1/PWM pin functions as PWM0_CH4 / PWM3_CH0 output. When AUXR5[5:4]. PWM3CH0P = 00, PIO01 = 1, P0.3 defined as PWM0_CH4. AUXR5[5:4]. PWM3CH0P = 10, PIO01 = 1, P0.3 defined as PWM3_CH0.
3	PIO00	P0.0/PWM pin function select 0 = P0.0/PWM pin functions as P0.0. 1 = P0.0/PWM pin functions as PWM0_CH3 / PWM2_CH1 output. When AUXR4[7:6]. PWM2CH1P = 00, PIO00 = 1, P0.0 defined as PWM0_CH3. AUXR4[7:6]. PWM2CH1P = 10, PIO00 = 1, P0.0 defined as PWM2_CH1.
2	PIO10	P1.0/PWM pin function select 0 = P1.0/PWM pin functions as P1.0. 1 = P1.0/PWM pin functions as PWM0_CH2 / PWM2_CH0 output. When AUXR4[5:4]. PWM2CH0P = 00, PIO10 = 1, P1.0 defined as PWM0_CH2. AUXR4[5:4]. PWM2CH0P = 10, PIO10 = 1, P1.0 defined as PWM2_CH0.
1	PIO11	P1.1/PWM pin function select 0 = P1.1/PWM pin functions as P1.1. 1 = P1.1/PWM pin functions as PWM0_CH1 / PWM1_CH1 output. When AUXR4[3:2]. PWM1CH1P = 00, PIO11 = 1, P1.1 defined as PWM0_CH1. AUXR4[3:2]. PWM1CH1P = 11, PIO11 = 1, P1.1 defined as PWM1_CH1.
0	PIO12	P1.2/PWM pin function select 0 = P1.2/PWM pin functions as P1.2. 1 = P1.2/PWM pin functions as PWM0C0 / PWM1_CH0 output. When AUXR4[1:0]. PWM1CH0P = 00, PIO12 = 1, P1.2 defined as PWM0_CH1. AUXR4[1:0]. PWM1CH0P = 10, PIO12 = 1, P1.2 defined as PWM1_CH0.

PIOCON1 – PWM or I/O Select

Register	SFR Address	Reset Value
PIOCON1	C6H, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
PIO17	-	PIO15	-	PIO04	PIO05	PIO14	-
R/W	-	R/W	-	R/W	R/W	R/W	-

Bit	Name	Description
7	PIO17	P1.7/PWM pin function select 0 = P1.7/PWM3_CH0 pin functions as GPIO P1.7. 1 = P1.7/PWM3_CH0 pin functions as PWM3_CH0 output.
6	-	Reserved
5	PIO15	P1.5/PWM pin function select 0 = P1.5/PWM pin functions as P1.5. 1 = P1.5/PWM pin functions as PWM0_CH5 / PWM3_CH1 output. When AUXR5[3:2].PWM3CH1P = 00, PIO15 = 1, P1.5 defined as PWM0_CH5. AUXR5[3:2].PWM3CH1P = 11, PIO15 = 1, P1.5 defined as PWM3_CH1.
4	-	Reserved
3	PIO04	P0.4/PWM pin function select 0 = P0.4/PWM pin functions as P0.4. 1 = P0.4/PWM pin functions as PWM0_CH3 / PWM2_CH1 output. When AUXR4[7:6].PWM2CH1P = 00, PIO04 = 1, P0.4 defined as PWM0_CH3. AUXR4[7:6].PWM2CH1P = 11, PIO04 = 1, P0.4 defined as PWM2_CH1.
2	PIO05	P0.5/PWM pin function select 0 = P0.5/PWM pin functions as P0.5. 1 = P0.5/PWM pin functions as PWM0_CH2 / PWM2_CH0 output. When AUXR4[5:4].PWM2CH1P = 00, PIO05 = 1, P0.5 defined as PWM0_CH2. AUXR4[5:4].PWM2CH1P = 11, PIO05 = 1, P0.5 defined as PWM2_CH0.
1	PIO14	P1.4/PWM pin function select 0 = P1.4/PWM pin functions as P1.4. 1 = P1.4/PWM pin functions as PWM0_CH1 / PWM1_CH1 output. When AUXR4[3:2]. PWM1CH1P = 00, PIO14 = 1, P1.4 defined as PWM0_CH1. AUXR4[3:2]. PWM1CH1P = 10, PIO14 = 1, P1.4 defined as PWM1_CH1.
0	-	Reserved

PIOCON2 – PWM or I/O Select

Register	SFR Address	Reset Value
PIOCON2	B7H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PIO34	PIO33	PIO32	PIO31	PIO30	PIO23	PIO22	PIO21
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
7	PIO34	P3.4/PWM pin function select 0 = P3.4/PWM pin functions as P3.4. 1 = P3.4/PWM pin functions as PWM3_CH1 output.
6	PIO33	P3.3/PWM pin function select 0 = P3.3/PWM pin functions as P3.3. 1 = P3.3/PWM pin functions as PWM0_CH0 output.
5	PIO32	P3.2/PWM pin function select 0 = P3.2/PWM pin functions as P3.2. 1 = P3.2/PWM pin functions as PWM3_CH0 output.
4	PIO31	P3.1/PWM pin function select 0 = P3.1/PWM pin functions as P3.1. 1 = P3.1/PWM pin functions as PWM2_CH1 output.
3	PIO30	P3.0/PWM pin function select 0 = P3.0/PWM pin functions as P3.0 1 = P3.0/PWM pin functions as PWM2_CH1 output.
2	PIO23	P23/PWM pin function select 0 = P2.3/PWM pin functions as P2.3. 1 = P2.3/PWM pin functions as PWM1_CH0 output.
1	PIO22	P2.2/PWM pin function select 0 = P2.2/PWM pin functions as P2.2. 1 = P2.2/PWM pin functions as PWM1_CH1 output.
0	PIO21	P2.1/PWM pin function select 0 = P2.1/PWM pin functions as P2.1. 1 = P2.1/PWM pin functions as PWM2_CH0 output.

AUXR4 – Auxiliary Register 4

Register	SFR Address	Reset Value
AUXR4	A3H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
PWM2CH1P		PWM2CH0P		PWM1CH1P		PWM1CH0P	
R/W		R/W		R/W		R/W	

Bit	Name	Description
7:6	PWM2CH1P	PWM2 channel 1 pin select 00 = Assign PWM2_CH1 to P3.0 01 = Assign PWM2_CH1 to P3.1 10 = Assign PWM2_CH1 to P0.0 11 = Assign PWM2_CH1 to P0.4
5:4	PWM2CH0P	PWM2 channel 0 pin select 00 = Reserved by default 01 = Assign PWM2_CH0 to P2.1 10 = Assign PWM2_CH0 to P1.0 11 = Assign PWM2_CH0 to P0.5
3:2	PWM1CH1P	PWM1 channel 1 pin select 00 = Reserved by default 01 = Assign PWM1_CH1 to P2.2 10 = Assign PWM1_CH1 to P1.4 11 = Assign PWM1_CH1 to P1.1
1:0	PWM1CH0P	PWM1 channel 0 pin select 00 = Reserved by default 01 = Assign PWM1_CH0 to P2.3 10 = Assign PWM1_CH0 to P1.2 11 = Reserved

AUXR5 – Auxiliary Register 5

Register	SFR Address	Reset Value
AUXR5	A4H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
CLOP	T0P	-	-	PWM3CH1P		PWM3CH0P	
R/W	R/W	-	-	R/W		R/W	

Bit	Name	Description
3:2	PWM3CH1P	PWM3 channel 1 pin select 00 = Reserved by default 01 = Assign PWM3_CH1 to P3.4 10 = Assign PWM3_CH1 to P1.5 11 = Assign PWM3_CH1 to P0.3
1:0	PWM3CH0P	PWM3 channel 0 pin select 00 = Reserved by default 01 = Assign PWM3_CH0 to P3.2 10 = Assign PWM3_CH0 to P0.1 11 = Reserved

PWMnINTC – PWM Interrupt Control

Register	SFR Address	Reset Value
PWM0INTC	B7H, page 1	0000_0000 b
PWM1INTC	B6H, 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	INTSELO
-	-	R/W	R/W	-	R/W	R/W	R/W

Bit	Name	Description
5:4	INTTYP[1:0]	<p>PWM interrupt type select</p> <p>These bit select PWM interrupt type.</p> <p>00 = Falling edge on PWM channel 0/1/2/3/4/5 pin.</p> <p>01 = Rising edge on PWM channel 0/1/2/3/4/5 pin.</p> <p>10 = Central point of a PWM period.</p> <p>11 = End point of a PWM period.</p> <p>Note that the central point interrupt or the end point interrupt is only available while PWM operates in center-aligned type.</p>
2:0	INTSEL[2:0]	<p>PWM interrupt pair select</p> <p>These bits select which PWM channel asserts PWM interrupt when PWM interrupt type is selected as falling or rising edge on PWM channel 0~5 pin..</p> <p>000 = PWM_CH0.</p> <p>001 = PWM_CH1.</p> <p>010 = PWM_CH2.</p> <p>011 = PWM_CH3.</p> <p>100 = PWM_CH4.</p> <p>101 = PWM_CH5.</p> <p>Others = PWM_CH0.</p>

6.9 Serial Port (UART0 & UART1)

6.9.1 Overview

The MS51 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.

6.9.2 Operating Modes

6.9.2.1 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 6.9-1](#) shows the associated timing of the serial port in Mode 0.

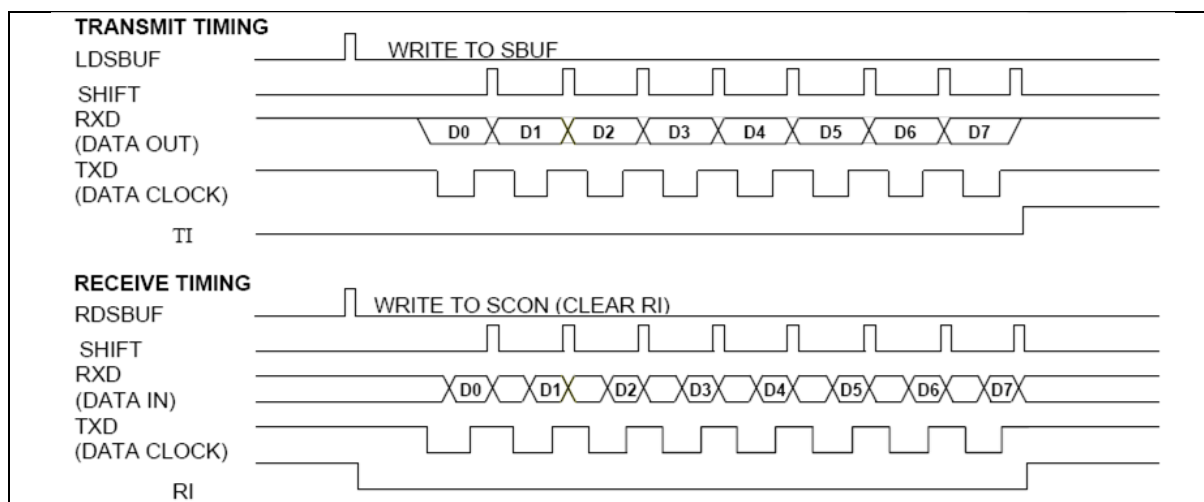


Figure 6.9-1 Serial Port Mode 0 Timing Diagram

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

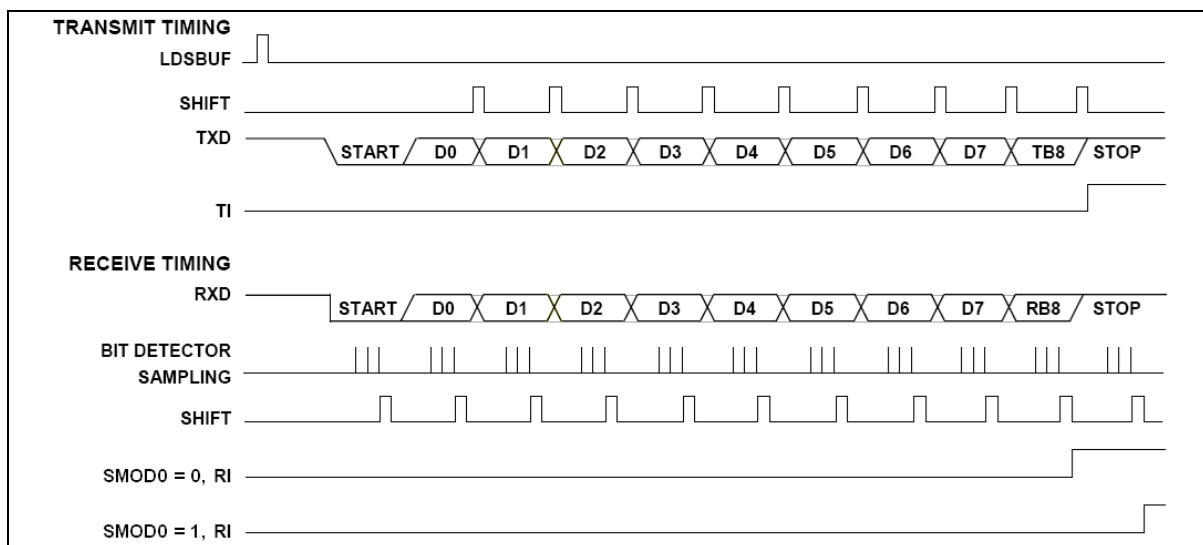


Figure 6.9-3 Serial Port Mode 2 and 3 Timing Diagram

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 6.9.5“Multiprocessor Communication” and Section 6.9.6“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.

6.9.2.4 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 6.9-3 for timing diagram of Mode 3. It has no difference from Mode 2.

6.9.3 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 6.9-1 Serial Port UART0 Mode / baud rate Description. The 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.

Mode	Frame Bits	SM0 / SM1 (SCON[7:6])	SM2 (SCON[5])	SMOD (PCON[7])	Baud Rate	
0	8	00	0	-	FSYS divided by 12	
			1		FSYS divided by 2	
1	10	01	-	0	Time1 T1M CKCON[4] = 0	$\frac{1}{32} \times \frac{F_{SYS}}{12 \times (256 - TH1)}$
					Time1 T1M CKCON[4] = 1	$\frac{1}{32} \times \frac{F_{SYS}}{(256 - TH1)}$
					Timer 3	$\frac{1}{32} \times \frac{F_{SYS}}{\text{Pre-scale} \times (65536 - (256 \times RH3 + RL3))}$
				1	Time1 T1M CKCON[4] = 0	$\frac{1}{16} \times \frac{F_{SYS}}{12 \times (256 - TH1)}$
					Time1 T1M CKCON[4] = 1	$\frac{1}{16} \times \frac{F_{SYS}}{(256 - TH1)}$
					Timer 3	$\frac{1}{16} \times \frac{F_{SYS}}{\text{Pre-scale} \times (65536 - (256 \times RH3 + RL3))}$
2	11	10	-	0	FSYS divided by 64	
				1	FSYS divided by 32	
3	11	11	-	0	Time 1 T1M CKCON[4] = 0	$\frac{1}{32} \times \frac{F_{SYS}}{12 \times (256 - TH1)}$
					Time 1 T1M CKCON[4] = 1	$\frac{1}{32} \times \frac{F_{SYS}}{(256 - TH1)}$
					Timer 3	$\frac{1}{32} \times \frac{F_{SYS}}{\text{Pre-scale} \times (65536 - (256 \times RH3 + RL3))}$
				1	Time1 T1M CKCON[4] = 0	$\frac{1}{16} \times \frac{F_{SYS}}{12 \times (256 - TH1)}$
					Time1 T1M CKCON[4] = 1	$\frac{1}{16} \times \frac{F_{SYS}}{(256 - TH1)}$
					Timer 3	$\frac{1}{16} \times \frac{F_{SYS}}{\text{Pre-scale} \times (65536 - (256 \times RH3 + RL3))}$
Note: Timer 1 should configured as a timer in auto-reload mode (Mode 2).						

Table 6.9-1 Serial Port UART0 Mode / baud rate Description

Mode	Frame Bits	SM0_1 / SM1_1 (S1CON[7:6])	SMOD_1 (T3CON[7])	Baud Rate	
0	8	00	-	FSYS divided by 12	
1	10	01	0	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	11	10	0	FSYS divided by 64	
			1	FSYS divided by 32	
3	11	11	0	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 6.9-2 Serial Port UART1 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

$$\frac{1}{16} \times \frac{F_{sys}}{\text{Pre-scale} \times (65536 - (256 \times \text{RH3} + \text{RL3}))}$$

```

SCON_1 = 0x52;    //UART1 Model,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
24 MHz	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%	
16 MHz	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%

6.9.4 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.

6.9.5 Multiprocessor Communication

The MS51 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.
6. 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.

6.9.6 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 = 110000Xb
```

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 = 1100XX0b
```

Example 2, slave 1:

```
SADDR = 11100000b
SADEN = 11111010b
Given = 11100X0b
```

Example 3, slave 2:

SADDR = 11000000b

SADEN = 11111100b

Given = 11000XXb

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 microcontrollers that do not support automatic address recognition.

6.9.7 Control Register of Serial Port

SCON – Serial Port Control

Register	SFR Address	Reset Value
SCON	98H, All page, 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
6	SM1	<p>SMOD0 (PCON.6) = 0: See Table 6.9-1 Serial Port UART0 Mode / baud rate Description for details.</p> <p>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.</p>
5	SM2	<p>Multiprocessor communication mode enable The function of this bit is dependent on the serial port 0 mode.</p> <p>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 8051 compatibility. 1 = The clock runs at FSYS/2 baud rate for faster serial communication.</p> <p>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.</p> <p>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.</p>
4	REN	<p>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.</p>
3	TB8	<p>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.</p>

Bit	Name	Description
2	RB8	<p>9th received bit</p> <p>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.</p>
1	TI	<p>Transmission interrupt flag</p> <p>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.</p>
0	RI	<p>Receiving interrupt flag</p> <p>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.</p>

SCON_1 – Serial Port 1 Control

Register	SFR Address	Reset Value
SCON_1	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
6	SM1_1	<p><u>SMOD0_1 (T3CON.6) = 0:</u> See Table 6.9-2 Serial Port UART1 Mode / baud rate Description for details.</p> <p><u>SMOD0_1 (T3CON.6) = 1:</u> 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.</p>
5	SM2_1	<p>Multiprocessor communication mode enable The function of this bit is dependent on the serial port 1 mode.</p> <p><u>Mode 0:</u> No effect.</p> <p><u>Mode 1:</u> 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.</p> <p><u>Mode 2 or 3:</u> 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.</p>
4	REN_1	<p>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.</p>
3	TB8_1	<p>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.</p>
2	RB8_1	<p>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.</p>
1	TI_1	<p>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.</p>

Bit	Name	Description
0	RI_1	<p>Receiving interrupt flag</p> <p>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.</p>

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 6.9-1 Serial Port UART0 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 6.9-1 Serial Port UART0 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]	<p>Serial port 0 data buffer</p> <p>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.</p> <p>The transmission is initiated through giving data to SBUF.</p>

SBUF_1 – Serial Port 1 Data Buffer

Register	SFR Address	Reset Value
SBUF_1	9AH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
SBUF_1[7:0]							
R/W							

Bit	Name	Description
7:0	SBUF_1[7:0]	<p>Serial port 1 data buffer</p> <p>This byte actually consists two separate registers. One is the receiving register, and the other is the transmitting buffer. When data is moved to SBUF_1, it goes to the transmitting buffer and is shifted for serial transmission. When data is moved from SBUF_1, it comes from the receiving register.</p> <p>The transmission is initiated through giving data to SBUF_1.</p>

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
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
-	-	EPWM3	EPWM2	EPWM1	EWKT	ET3	ES_1
-	-	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.

SADDR – Slave 0 Address

Register	SFR Address	Reset Value
SADDR	A9H, Page 0	0000_0000 b

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

Bit	Name	Description
7:0	SADDR[7:0]	Slave 0 address This byte specifies the microcontroller’s own slave address for UATR0 multi-processor communication.

SADEN – Slave 0 Address Mask

Register	SFR Address	Reset Value
SADEN	B9H, Page 0	0000_0000 b

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

Bit	Name	Description
7:0	SADEN[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.

SADDR_1 – Slave 1 Address

Register	SFR Address	Reset Value
SADDR_1	BBH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
SADDR_1[7:0]							
R/W							

Bit	Name	Description
7:0	SADDR_1[7:0]	Slave 1 address This byte specifies the microcontroller's own slave address for UART1 multi-processor communication.

SADEN 1 – Slave 1 Address Mask

Register	SFR Address	Reset Value
SADEN_1	BAH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
SADEN_1[7:0]							
R/W							

Bit	Name	Description
7:0	SADEN_1[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	A2H , Page 0	POR: 0000 0000b, Software reset: 1U00 0000b, nRESET pin: U100 0000b, Others: UUU0 0000b

7	6	5	4	3	2	1	0
SWRF	RSTPINF	HardF	SLOW	GF2	UART0PX	0	DPS
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

Bit	Name	Description
2	UART0PX	Serial port 0 pin exchange 0 = Assign RXD to P0.7 and TXD to P0.6 by default. 1 = Exchange RXD to P0.6 and TXD to P0.7. Note that TXD and RXD 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.

AUXR2 – Auxiliary Register 2

Register	SFR Address	Reset Value
AUXR2	A1H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
UART2TXP		UART2RXP		UART1TXP		UART1RXP	
R/W		R/W		R/W		R/W	

Bit	Name	Description
3:2	UART1TXP	UART1 TX pin select 00 = Reserved by default 01 = Assign UART1 TX to P1.6 10 = Assign UART1 TX to P3.6 11 = Assign UART1 TX to P1.0
1:0	UART1RXP	UART1 RX pin select 00 = Reserved by default 01 = Assign UART1 RX to P0.2 10 = Assign UART1 RX to P3.7 11 = Assign UART1 RX to P0.0

6.10 ISO 7816-3 Interface (SC0~2 & UART2 ~ 4)

6.10.1 Overview

The MS51 32K series provides ISO 7816-3 Interface controller (SC controller) with asynchronous protocol 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.

6.10.2 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

6.10.3 Basic Configuration

Following is the ISO 7816-3 multi function pin define

URAT Pin	SC Pin	Pin Name	SFR Define		
			SFR Byte Name	SFR Bit Name	Value
UART2_TXD	SC0_CLK	P0.3	AUXR2[7:6]	UART2TXP	01
		P3.0			10
UART2_RXD	SC0_DAT	P0.4	AUXR2[5:4]	UART2RXP	01
		P1.7			10
UART3_TXD	SC1_CLK	P1.2	AUXR3[3:2]	UART3TXP	01
		P1.5			10
		P0.5			11
UART3_RXD	SC1_DAT	P1.1	AUXR3[1:0]	UART3RXP	01
		P2.5			10
		P3.4			11
UART4_TXD	SC2_CLK	P2.3	AUXR3[7:6]	UART4TXP	01
UART4_RXD	SC2_DAT	P2.2	AUXR3[5:4]	UART4RXP	01

Table 6.10-1 Smart Card or UART Pin Define And Enable Control Register

6.10.4 Block Diagram

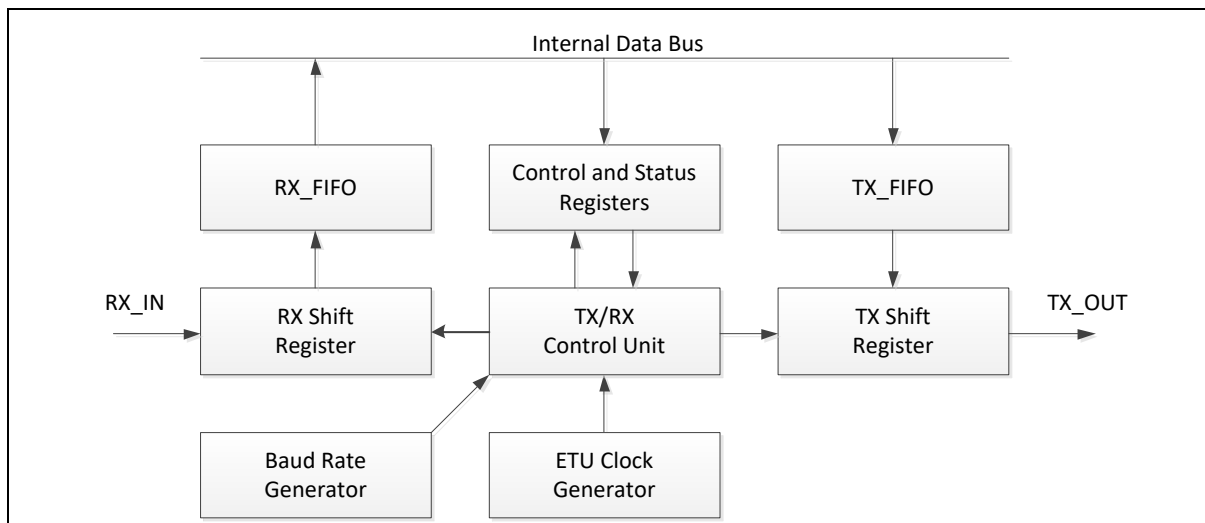


Figure 6.10-1 SC Controller Block Diagram

6.10.5 Operating Modes

6.10.5.1 ISO 7816-3 Mode

The ISO 7816-3 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

1. SCn_CLK / UARTn_TXD : SC clock pin (output from MCU)
2. SCn_DAT / UARTn_RXD : SC data pin (bi-directional)
3. SCn_RST: SC reset pin (output from MCU, firmware assigned GPIO)
4. SCn_PWR: SC power pin (output from MCU, firmware assigned GPIO)
5. *: SCn_PWR is used for power control function to turn ON/OFF the power for ISO 7816-3. **Do not** use SCn_PWR as the direct power supply for ISO 7816-3.
6. SCn_CD: SC card detect pin (input to MCU, detect card by a card insert mechanism)

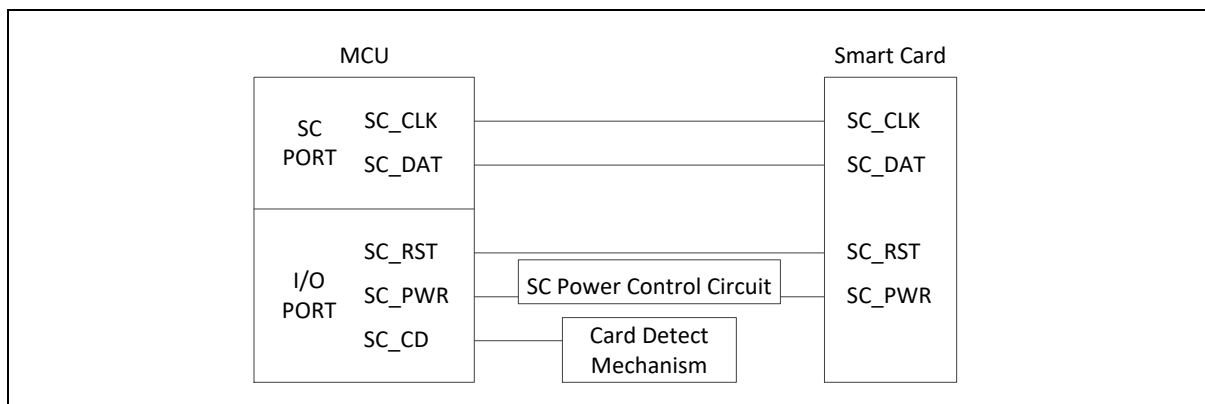


Figure 6.10-2 SC Interface Connection

Activation and Cold Reset

The activation and cold reset sequence is shown in Figure 15.3-2

1. Set SCn_RST to low by software programming to '0'
2. Set SCn_PWR at high level by software programming to '1' before timing T1 and SCn_DAT at high level (reception mode) by software programming to '1' period of timing T1.
3. Enable SCn_CLK clock by programming CLKKEEP (SCCR2[1]) to '1' after timing T1.
4. De-assert SCn_RST to high by software programming to '1' after timing T2.
5. ISO 7816-3 host controller read the card ATR period of timing T3.

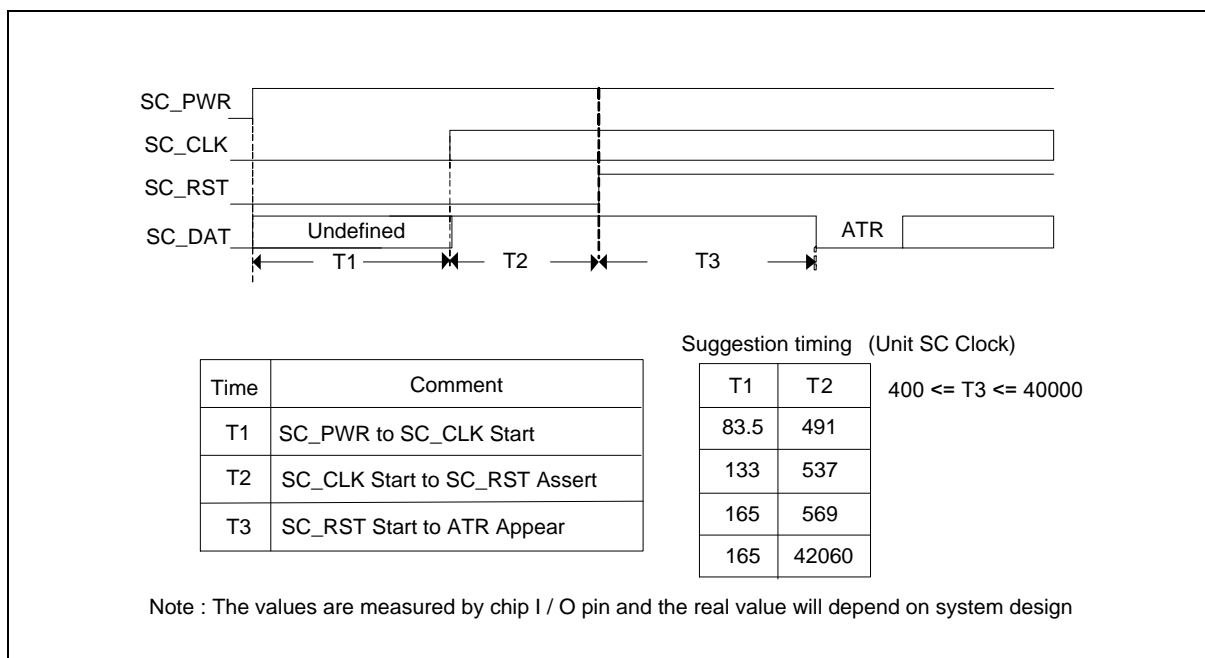


Figure 6.10-3 SC Activation and Cold Reset Sequence

Warm Reset

The warm reset sequence is showed in Figure 15.3-3

1. Set SCn_RST to low by software programming to '0' before timing T4.
2. Set SCn_DAT to high by software programming to '1' period of timing T4.
3. Set SCn_RST to high by software programming to '1' after timing T5.
4. ISO 7816-3 host controller read the card ATR period of timing T6.

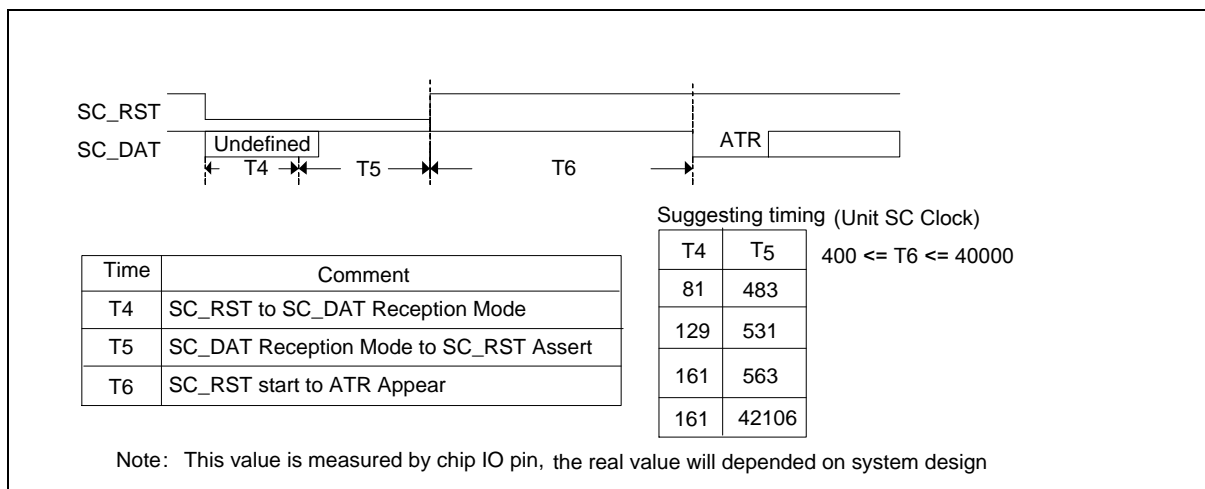


Figure 6.10-4 SC Warm Reset Sequence

Deactivation

The deactivation sequence is shown in Figure 15.3-4

1. Set SCn_RST to low by software programming to '0' period of timing T7.
2. Stop SCn_CLK by programming CLKKEEP (SCCR2[1]) to '0' period of timing T8.
3. Set SCn_DAT to low by software programming to '0' period of timing T8.
4. Deactivate SCn_PWR by software programming to '0' period of timing T9.

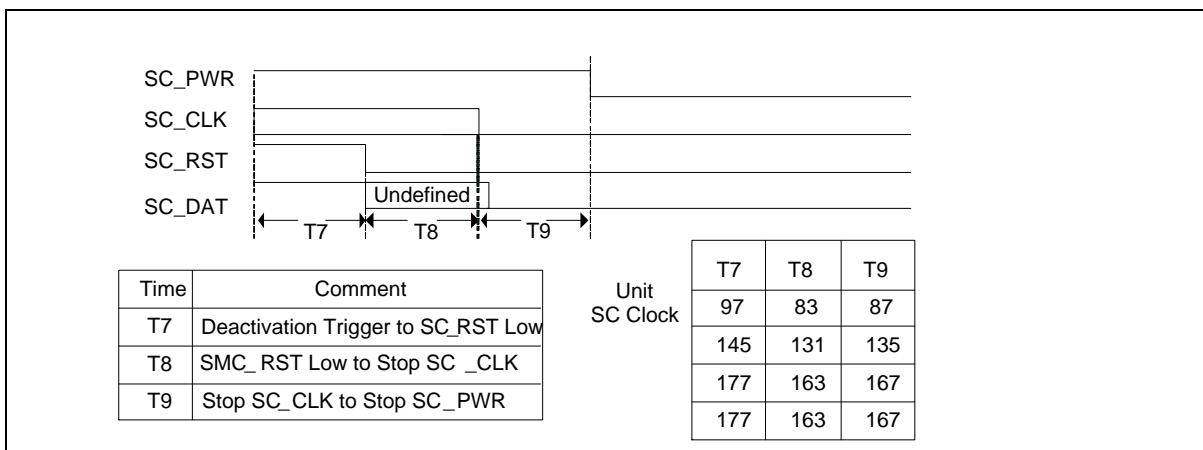


Figure 6.10-5 SC Deactivation Sequence

6.10.5.2 UART Mode

When the UARTEN (SCnCR2[0]) bit is set, the ISO 7816-3 Interface controller can also be used as basic UART function. The following is the program example for UART mode.

Programming example:

1. Set UARTEN (SCnCR2[0]) bit to enter UART mode.
2. Fill "0" to CONSEL (SCnCR1[4]) and AUTOSEN (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.F

6.10.6 ISO 7816-3 Data Transfer

Basically, the ISO 7816-3 interface acts as a half-duplex asynchronous communication port and its data format is composed of ten consecutive bits which is showman.

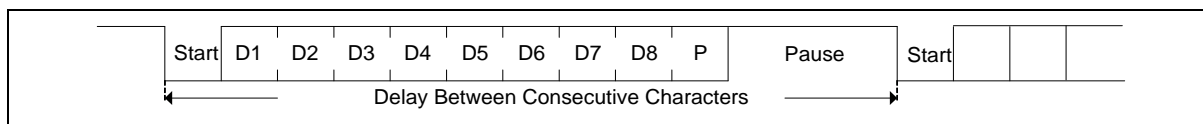


Figure 6.10-6 SC Data Character

6.10.7 Initial Character TS

According to 7816-3, the initial character TS has two possible patterns shown in Figure 6.10-7 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 AUTOSEN (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 AUTOSEN (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.

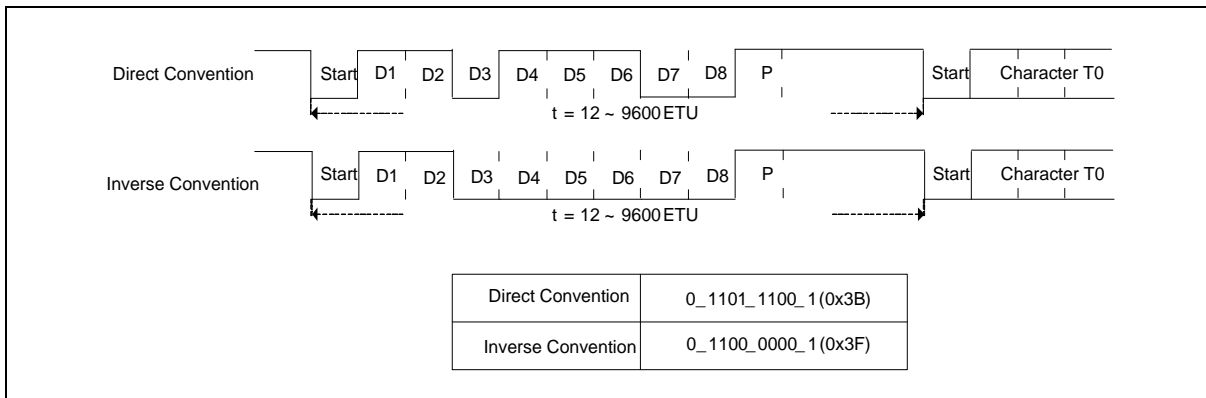


Figure 6.10-7 Initial Character TS

6.10.8 Error Signal and Character Repetition

According to ISO 7816-3 T=0 mode description, as shown in **Error! Reference source not found.**, 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.

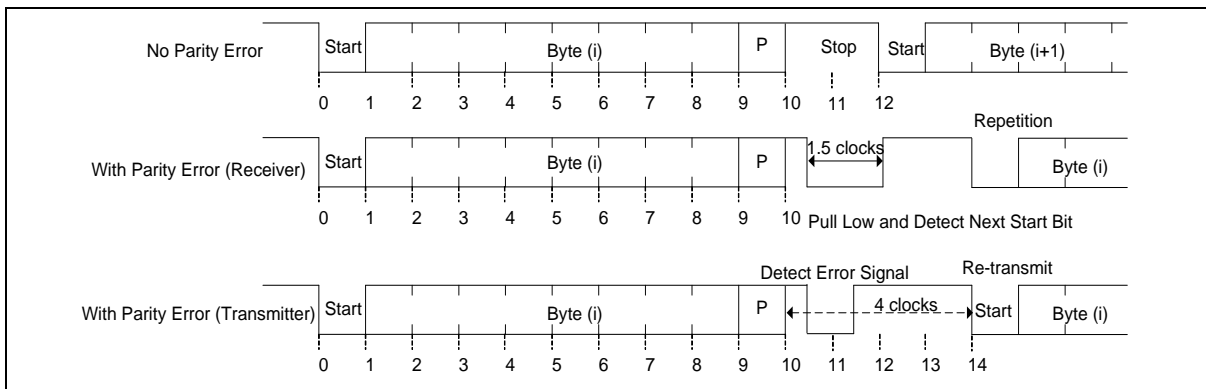


Figure 6.10-8 SC Error Signal

6.10.9 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 ISO 7816-3, 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 ISO 7816 sends data to ISO 7816-3 host controller, first. After the period is greater than (16.5 or 22.5, by T bit setting), the ISO 7816-3 host controller begin to send the data.

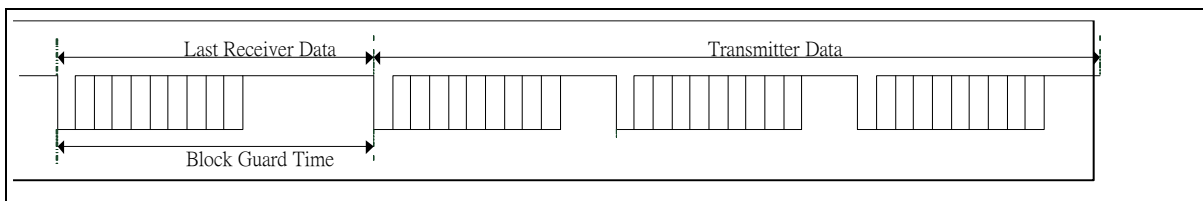


Figure 6.10-9 Transmit Direction Block Guard Time Operation

In receive direction, the ISO 7816-3 host controller sends data to ISO 7816-3, first. If the ISO 7816-3 sends data to ISO 7816-3 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.

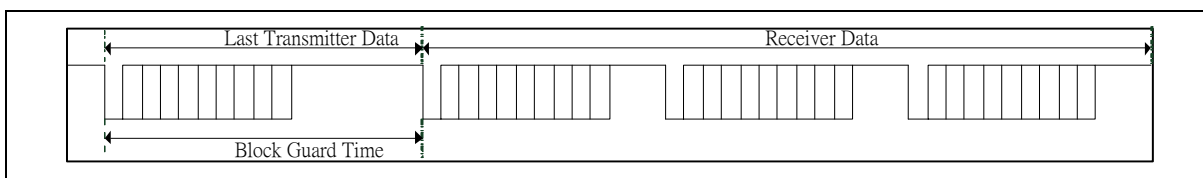


Figure 6.10-10 Receive Direction Block Guard Time Operation

Extra Guard Time is EGT (SCnEGT[7:0]), it only affects the data transmitted by ISO 7816-3 interface, the format is shown as Figure 6.10-11.

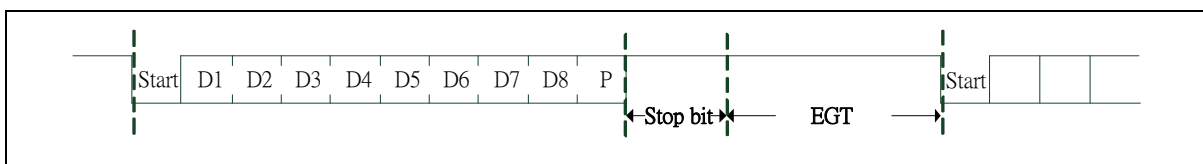


Figure 6.10-11 Extra Guard Time Operation

6.10.10 Control Registers of SC Controller

SCnCR0 – SC Control Register 0

Register	SFR Address	Reset Value
SC0CR0	F1H, Page 2	0000_0000 b
SC1CR0	F3H, Page 2	0000_0000 b
SC2CR0	F5H, Page 2	0000_0000 b

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

Bit	Name	Description
7	NSB	<p>Stop Bit Length</p> <p>This field indicates the length of stop bit.</p> <p>0 = The stop bit length is 2 ETU.</p> <p>1 = The stop bit length is 1 ETU.</p> <p>Note: The default stop bit length is 2. SC and UART adopt NSB to program the stop bit length.</p>
6	T	<p>T Mode</p> <p>0 = T0 (ISO 7816-3 T = 0 mode).</p> <p>1 = T1 (ISO 7816-3 T = 1 mode).</p> <p>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 ISO 7816-3, 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.</p> <p>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.</p> <p>Note: The description please see section 6.10.8 Error Signal and Character Repetition</p>
5	RXBGTEN	<p>Receiver Block Guard Time Function Enable Bit</p> <p>0 = Receiver block guard time function Disabled.</p> <p>1 = Receiver block guard time function Enabled.</p>
4	CONSEL	<p>Convention Selection</p> <p>0 = Direct convention.</p> <p>1 = Inverse convention.</p> <p>Note 1: This bit is auto clear to "0", if AUTOGEN(SCnCR0[3]) is writing "1"</p> <p>Note 2: If AUTOGEN(SCnCR0[3]) is enabled, hardware will decide the convention and change the CONSEL (SCnCR0[4]) bits automatically after SCEN (SCnCR0[0]) = "1".</p>

Bit	Name	Description
3	AUTOZEN	<p>Auto Convention Enable Bit</p> <p>0 = Auto-convention Disabled.</p> <p>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.</p> <p>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.</p>
2	TXOFF	<p>TX Transition Disable Bit</p> <p>0 = The transceiver Enabled.</p> <p>1 = The transceiver Disabled.</p>
1	RXOFF	<p>RX Transition Disable Bit</p> <p>0 = The receiver Enabled.</p> <p>1 = The receiver Disabled.</p> <p>Note: If AUTOZEN (SCnCR0[3])is enabled, these fields must be ignored.</p>
0	SCEN	<p>SC Engine Enable Bit</p> <p>Set this bit to 1 to enable SC operation. If this bit is cleared, SC will force all transition to IDLE state</p> <p>Note: SCEN must be set to 1 before filling in other registers, or ISO 7816-3 will not work properly.</p>

SCnCR1 – SC Control Register

Register	SFR Address	Reset Value
SC0CR1	F2H, Page 2	0000_0000 b
SC1CR1	F4H, Page 2	0000_0000 b
SC2CR1	F6H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
OPE	PBOFF	WLS		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 ISO 7816-3 mode, this field must be '0' (default setting is with parity bit)
5:4	WLS	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 ISO 7816-3 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.
0	UARTEN	UART Mode Enable Bit 0 = ISO 7816-3 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 ISO 7816-3 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 2	0000_0000 b
SC1DR	E1H, Page 2	0000_0000 b
SC2DR	E9H, 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]	<p>SC / UART buffer data</p> <p>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.</p> <p>Note: If SCEN(SCnCR0[0]) is not enabled, SCnDR cannot be programmed.</p>

SCnEGT – SC Extra Guard Time Register

Register	SFR Address	Reset Value
SC0EGT	DAH, Page 2	0000_0000 b
SC1EGT	E2H, Page 2	0000_0000 b
SC2EGT	EAH, 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 2	0111_0011 b
SC1ETURD0	E3H, Page 2	0111_0011 b
SC2ETURD0	EBH, 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 2	0011_0001 b
SC1ETURD1	E4H, Page 2	0011_0001 b
SC2ETURD1	ECH, 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]	<p>SC clock divider</p> <p>000 = F_{SC} is $F_{SYS}/1$.</p> <p>001 = F_{SC} is $F_{SYS}/2$.</p> <p>010 = F_{SC} is $F_{SYS}/4$.</p> <p>011 = F_{SC} is $F_{SYS}/8$. (By default.)</p> <p>100 = F_{SC} is $F_{SYS}/16$.</p> <p>101 = F_{SC} is $F_{SYS}/16$.</p> <p>110 = F_{SC} is $F_{SYS}/16$.</p> <p>111 = F_{SC} is $F_{SYS}/16$.</p> <p>Note: that the F_{SC} clock should be 1Mhz ~ 5Mhz for ISO/IEC 7816-3 standard</p>
3:0	ETURDIV [11:8]	<p>MSB bits of ETU Rate Divider</p> <p>The field indicates the MSB of clock rate divider.</p> <p>The real ETU is $ETURDIV[11:0] + 1$.</p> <p>Note 1: ETURDIV[11:0] must be greater than 0x004.</p> <p>Note 2: SCnETURD0 has to program first, then SCnETUDR1 .</p>

ScnIE – SC Interrupt Enable Control Register

Register	SFR Address	Reset Value
SC0IE	DCH, Page 2	0000_0000 b
SC1IE	E4H, Page 2	0000_0000 b
SC2IE	ECH, 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
7:4	-	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 2	0000_0010 b
SC1IS	E6H, Page 2	0000_0010 b
SC2IS	EEH, Page 2	0000_0010 b

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

Bit	Name	Description
7:5	-	Reserved
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 2	0000_1010 b
SC1TSR	E7H, Page 2	0000_1010 b
SC2TSR	EFH, 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	TXOVF	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.
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.

AUXR2 – Auxiliary Register 2

Register	SFR Address	Reset Value
AUXR2	A1H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
UART2TXP		UART2RXP		UART1TXP		UART1RXP	
R/W		R/W		R/W		R/W	

Bit	Name	Description
7:6	UART2TXP	UART2 TX pin select 00 = Reserved by default 01 = Assign UART2 TX to P0.3 10 = Assign UART2 TX to P3.0 11 = Reserved
5:4	UART2RXP	UART2 RX pin select 00 = Reserved by default 01 = Assign UART2 RX to P0.4 10 = Assign UART2 RX to P1.7 11 = Reserved

AUXR3 – Auxiliary Register 3

Register	SFR Address	Reset Value
AUXR3	A2H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
UART4TXP		UART4RXP		UART3TXP		UART3RXP	
R/W		R/W		R/W		R/W	

Bit	Name	Description
7:6	UART4TXP	UART4 TX pin select 00 = Reserved by default 01 = Assign UART4 TX to P2.3 10 = Reserved 11 = Reserved
5:4	UART4RXP	UART4 RX pin select 00 = Reserved by default 01 = Assign UART4 RX to P2.2 10 = Reserved 11 = Reserved
3:2	UART3TXP	UART3 TX pin select 00 = Reserved by default 01 = Assign UART3 TX to P1.2 10 = Assign UART3 TX to P1.5 11 = Assign UART3 TX to P0.5
1:0	UART3RXP	UART3 RX pin select 00 = Reserved by default 01 = Assign UART3 RX to P1.1 10 = Assign UART3 RX to P2.5 11 = Assign UART3 RX to P3.4

6.11 Inter-Integrated Circuit (I²C)

6.11.1 Overview

The MS51 provides two Inter-Integrated Circuit (I²C) bus to serves as an serial interface between the microcontrollers and the I²C devices such as EEPROM, LCD module, temperature sensor, and so on. The I²C bus used two wires design (a serial data line I2C0_SDA and a serial clock line I2C0_SCL) to transfer information between devices.

The I²C 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 I²C bus supports four transfer modes including master transmitter, master receiver, slave receiver, and slave transmitter. The I²C interface only supports 7-bit addressing mode. A special mode General Call is also available. The I²C can meet both standard (up to 100kbps) and fast (up to 400k bps) speeds.

6.11.2 Functional Description

For a bi-directional transfer operation, the I2C0_SDA and I2C0_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 MS51, user should set output latches of I2C0_SCL and I2C0_SDA. As logic 1 before enabling the I²C function by setting I2CEN.

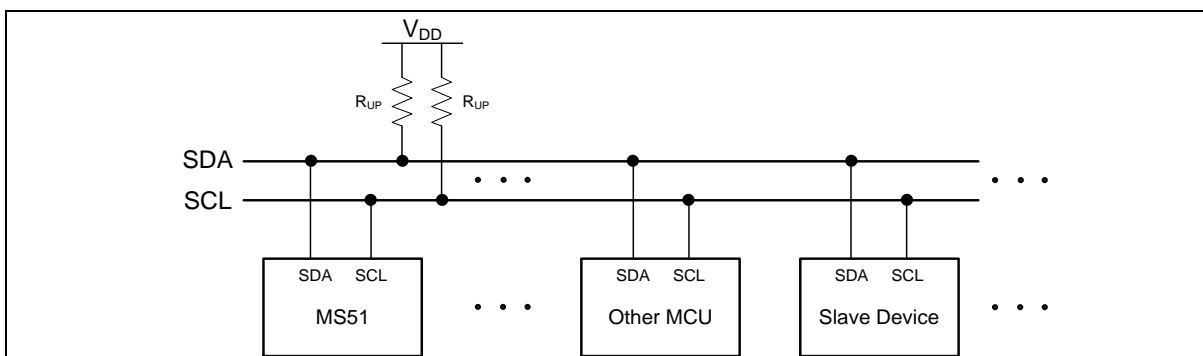


Figure 6.11-1 I²C Bus Interconnection

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 I2C0_SCL line, the device outputting data on the I2C0_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 I2C0_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 I2C0_SCL line.

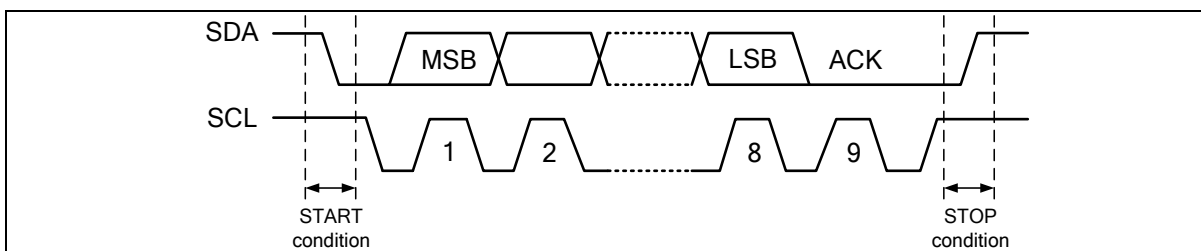


Figure 6.11-2 I²C Bus Protocol

6.11.2.1 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 I2C0_SDA line while I2C0_SCL line is high. The STOP condition is defined as a low-to-high transition on the I2C0_SDA line while I2C0_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 I²C 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.

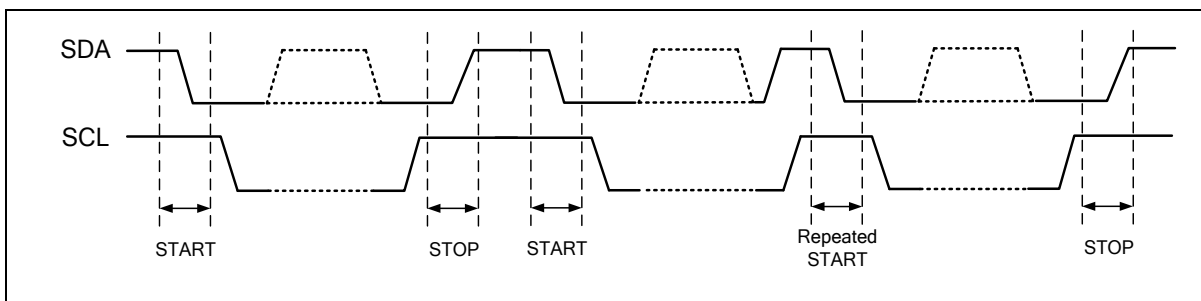


Figure 6.11-3 START, Repeated START, and STOP Conditions

6.11.2.2 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 (R) 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 6.11-4 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.

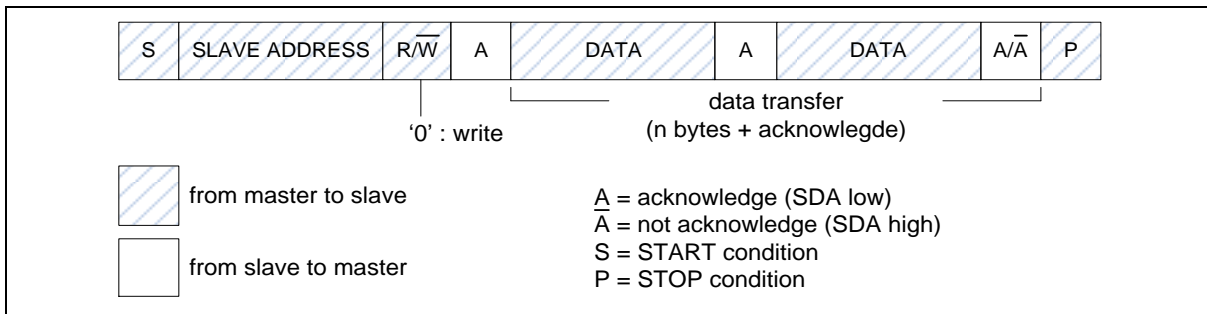


Figure 6.11-4 Master Transmits Data to Slave by 7-bit

Figure 6.11-5 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.

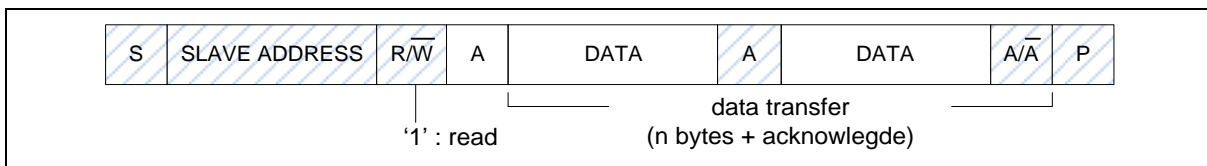


Figure 6.11-5 Master Reads Data from Slave by 7-bit

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 I²C bus accepts 127 devices with their slave addresses 1 to 127.

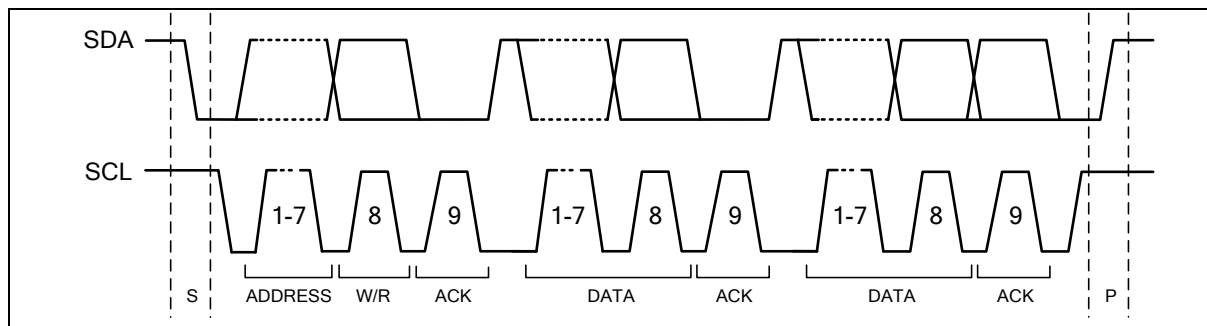


Figure 6.11-6 Data Format of One I²C Transfer

During the data transaction period, the data on the I2C0_SDA line should be stable during the high period of the clock, and the data line can only change when I2C0_SCL is low.

6.11.2.3 Acknowledge

The 9th I2C0_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 I2C0_SDA line low. The acknowledge-related clock pulse is generated by the master. The transmitter should release control of I2C0_SDA line during the acknowledge clock pulse. The ACK is an active-low signal, pulling the I2C0_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 I2C0_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 I2C0_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 I2C0_SDA line to allow the master to generate a STOP or a repeated START condition.

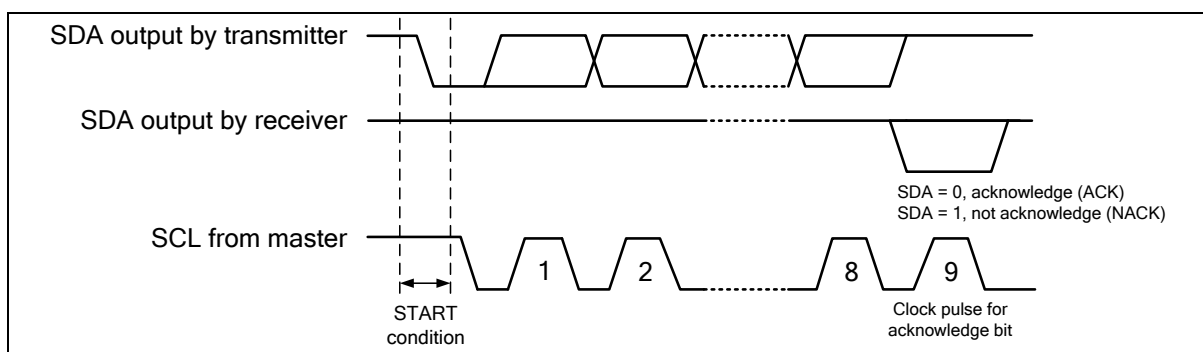


Figure 6.11-7 Acknowledge Bit

6.11.2.4 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 I2C0_SDA line, while I2C0_SCL is high. During arbitration, the first of the competing master devices to place a '1' (high) on I2C0_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 I2C0_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 I2C0_SCL line until the end of the byte in which it loses the arbitration.

Arbitration is carried out by all masters continuously monitoring the I2C0_SDA line after outputting data. If the value read from the I2C0_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 I2C0_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.

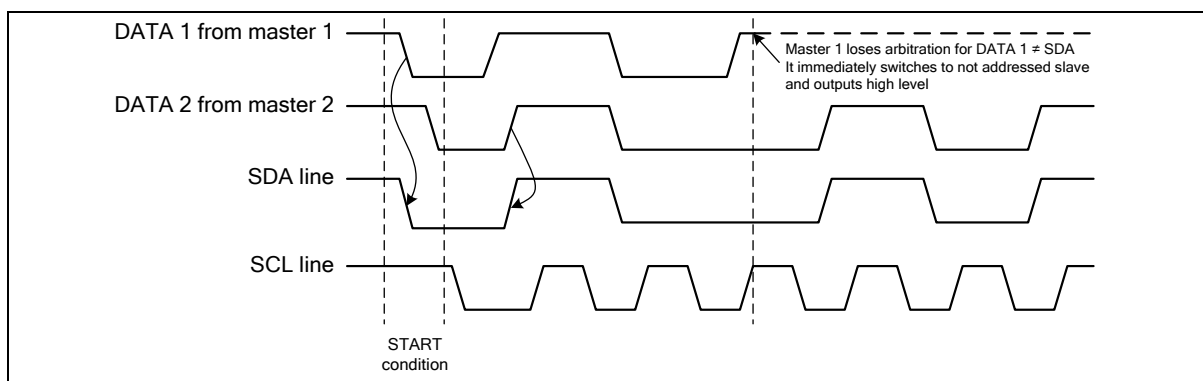


Figure 6.11-8 Arbitration Procedure of Two Masters

Since control of the I²C 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.

6.11.2.5 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 6.11-9 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.

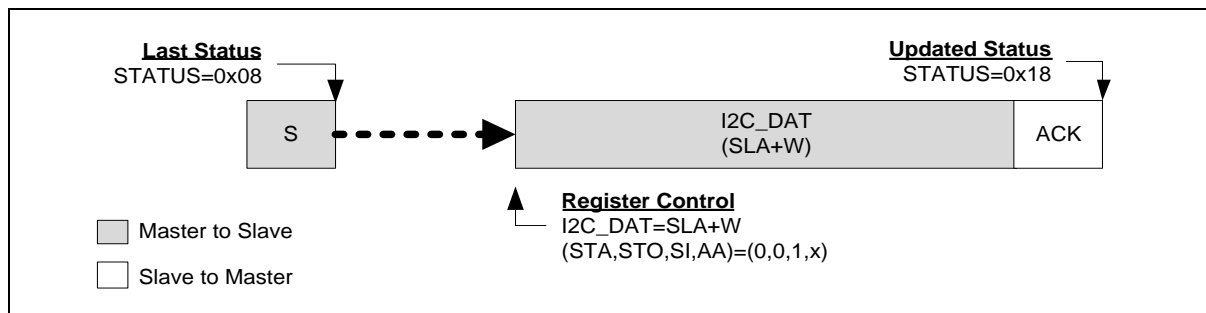


Figure 6.11-9 Control I²C Bus according to the Current I²C Status

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.

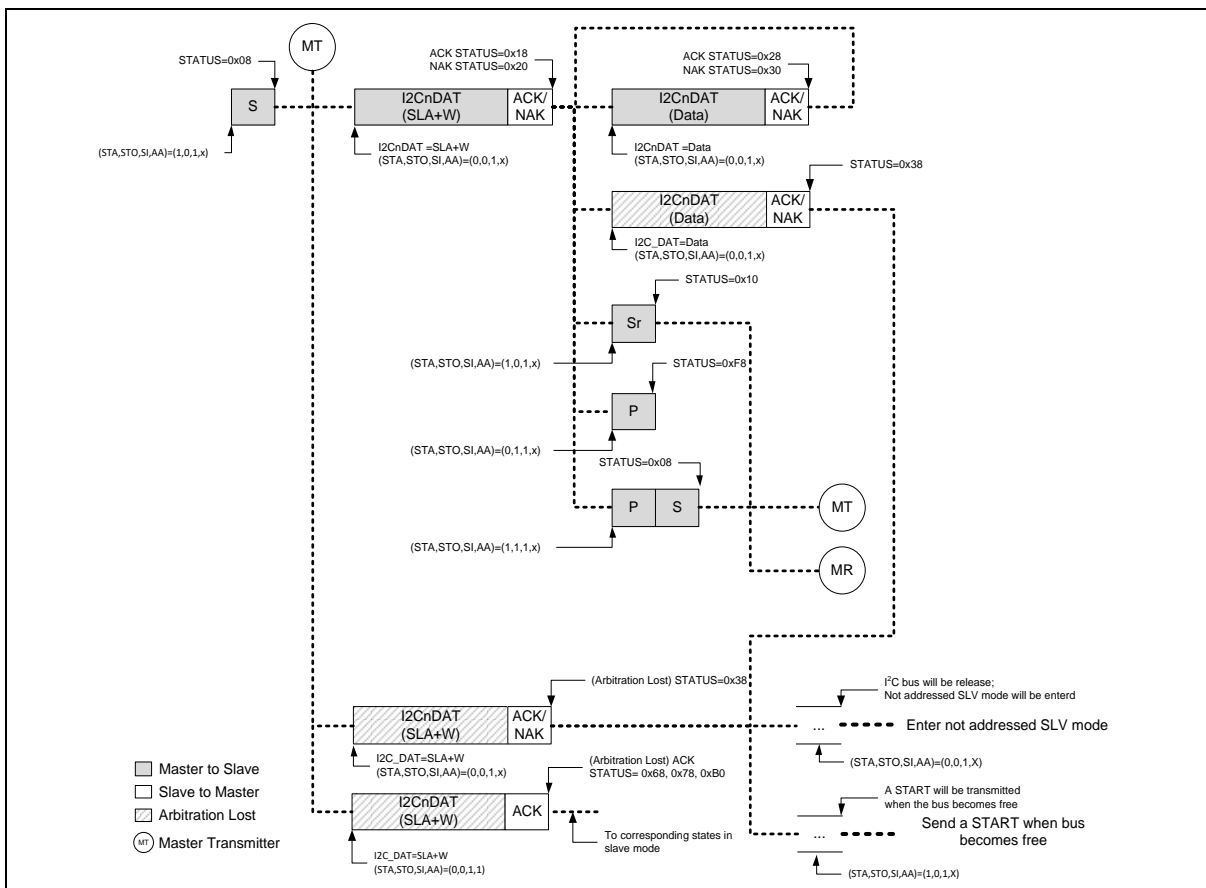


Figure 6.11-10 Flow and Status of Master Transmitter Mode

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.

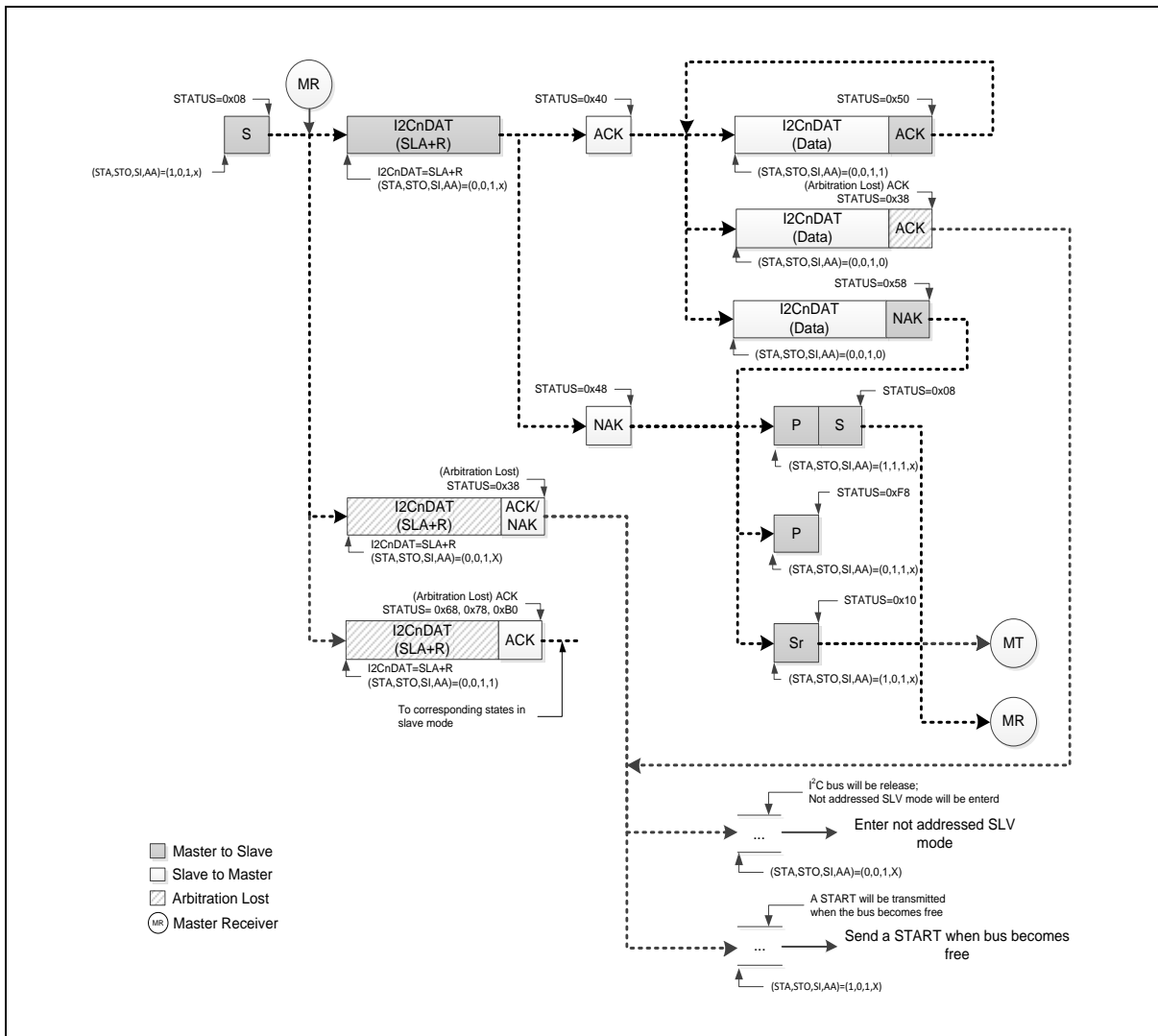


Figure 6.11-11 Flow and Status of Master Receiver Mode

Slave Receiver

In the slave receiver mode, several bytes of data are received from 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 I²C 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 I²C port is equipped with four slave address registers, I2CnADDRx (x=0~3). The contents of the register are irrelevant when I²C 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 I²C 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.

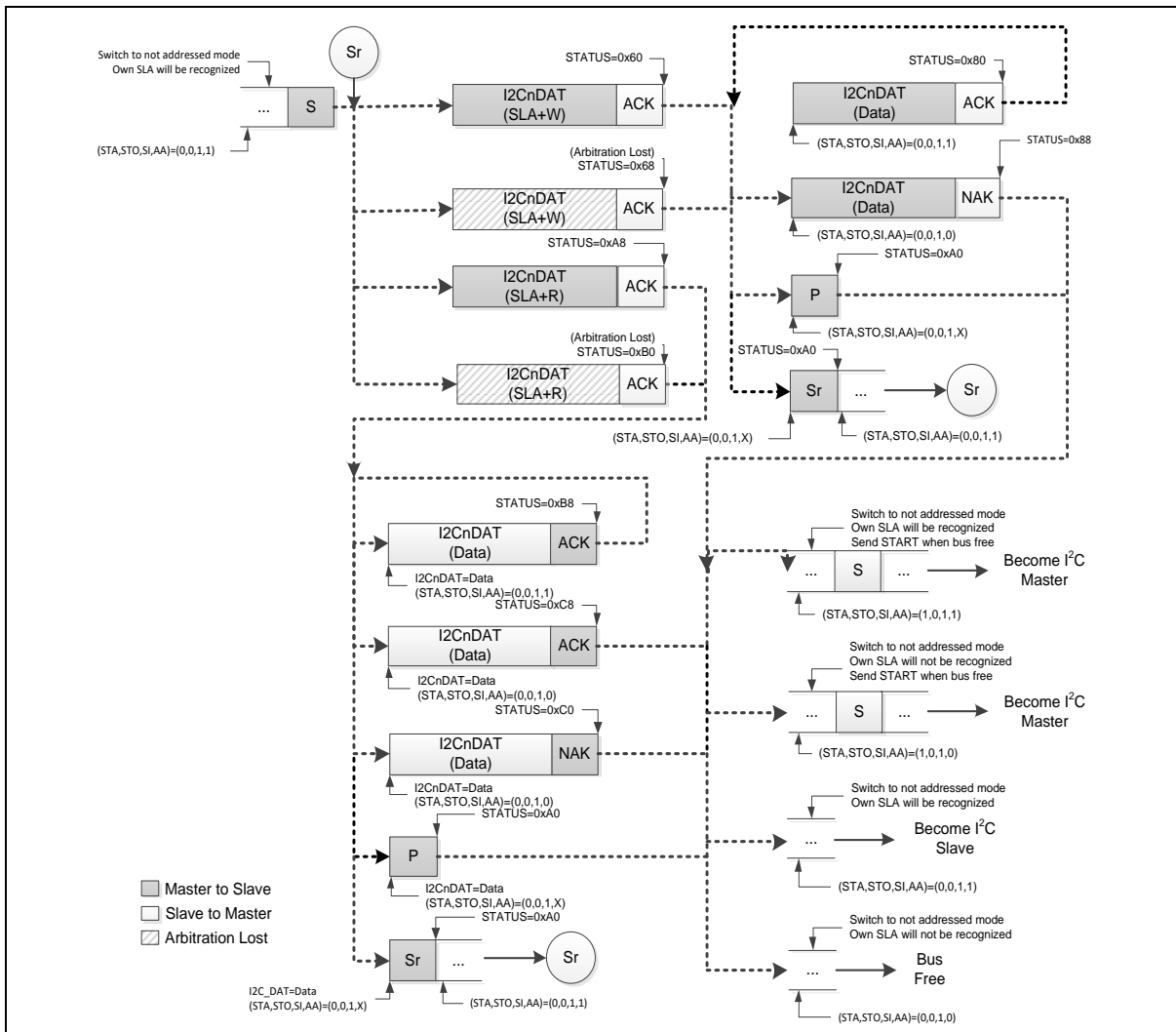


Figure 6.11-12 Flow and Status of Slave Receiver Mode

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.

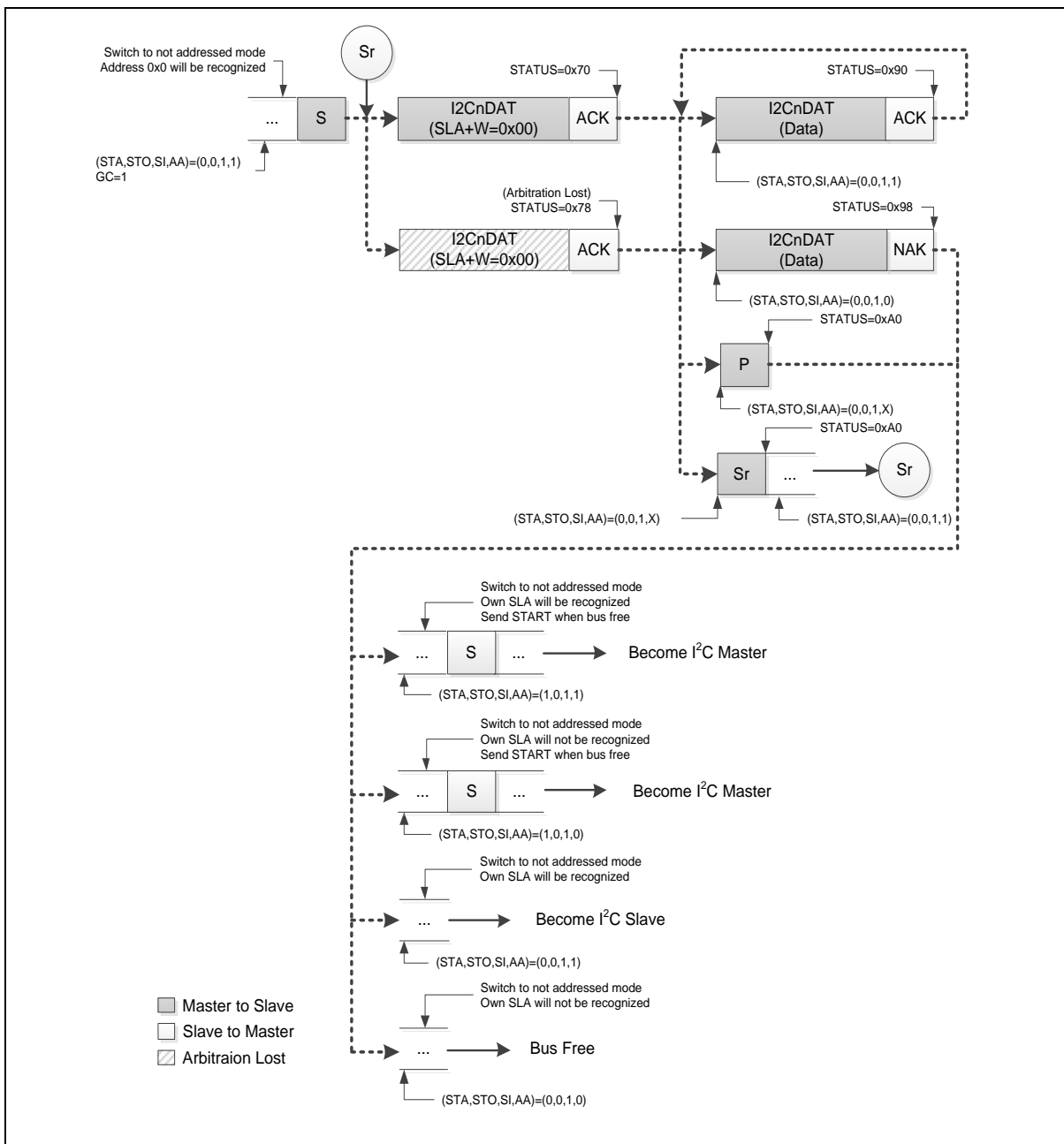


Figure 6.11-13 Flow and Status of General Call Mode

6.11.2.6 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 I²C 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 I²C bus, the operating device immediately switches to the not addressed salve mode, releases I2C0_SDA and I2C0_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 I²C bus without issuing a real STOP condition waveform on I²C bus.

There is a special case if a START or a repeated START condition is not successfully generated for I²C bus is obstructed by a low level on I2C0_SDA line e.g. a slave device out of bit synchronization, the problem can be solved by transmitting additional clock pulses on the I2C0_SCL line. The I²C hardware transmits additional clock pulses when the STA bit is set, but no START condition can be generated because the I2C0_SDA line is pulled low. When the I2C0_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 I2C0_SDA is obstructed low, the I²C 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 I²C 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.		

6.11.3 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*/
           //ACK next received DATA
           AA = 1;
           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:
        received,
        mode
        /*98H, previous General Call, DATA
        NACK returned, not addressed SLAVE
        entered*/
        DATA_RECEIVED_LAST3 = I2DAT;
        AA = 1;
        break;
        //=====
        //Slave Mode
        //=====
    case 0xA0:
        while
        /*A0H, STOP or repeated START received
        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:
        transmitted,
        /*B8H, previous own SLA+R, DATA
        ACK received*/
        I2DAT = NEXT_SEND_DATA4;
        if (To_TX_Last_Data)
            AA = 0;
        else
            AA = 1;
        break;
    case 0xC0:
        transmitted,
        mode
        /*C0H, previous own SLA+R, DATA
        NACK received, not addressed SLAVE
        entered*/
        AA = 1;
        break;
    case 0xC8:
        SLAVE
        /*C8H, previous own SLA+R, last DATA trans-
        mitted, ACK received, not addressed
        mode entered*/
        AA = 1;
        break;
    } //end of switch (I2STAT)

    SI = 0;
    I2C_ISR
    while(STO);
    error
    //SI should be the last command of
    //wait for STOP transmitted or bus
    //free, STO is cleared by hardware
}

```

6.11.4 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.

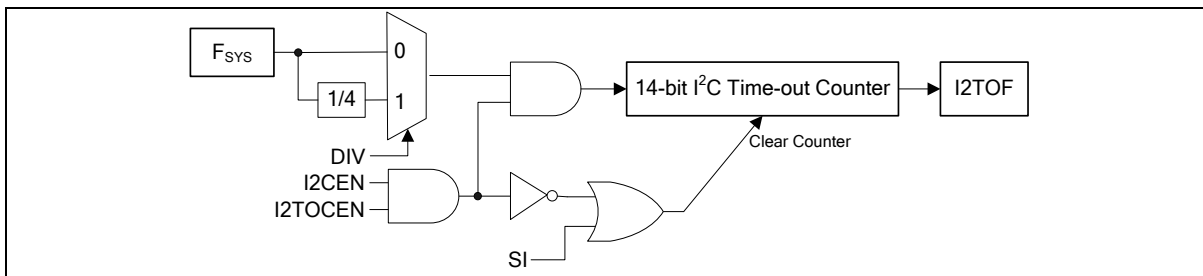


Figure 6.11-14 I²C Time-Out Counter

6.11.5 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.

6.11.6 Control Registers of I²C

There are five control registers to interface the I²C bus including I2CON, I2STAT, I2DAT, I2ADDR, and I2CLK. These registers provide protocol control, status, data transmitting and receiving functions, and clock rate configuration. For application flexibility, I2C0_SDA and I2C0_SCL pins can be exchanged by I2CPX (I2CON.0). The following registers relate to I²C function.

I2CON – I²C Control

Register	SFR Address	Reset Value
I2CON	C0H, All pages, Bit addressable	0000_0000 b

7	6	5	4	3	2	1	0
I	I2CEN	STA	STO	SI	AA	-	I2CPX
R/W	R/W	R/W	R/W	R/W	R/W	-	R/W

Bit	Name	Description
7	I	I2C0 hold time extend enable 0 = I ² C DATA to I2C0_SCL hold time extend disabled 1 = I ² C DATA to I2C0_SCL hold time extend enabled, extend 8 system clock
6	I2CEN	I ² C bus enable 0 = I ² C bus Disabled. 1 = I ² C bus Enabled. Before enabling the I ² C, I2C0_SCL and I2C0_SDA port latches should be set to logic 1.
5	STA	START flag When STA is set, the I ² C generates a START condition if the bus is free. If the bus is busy, the I ² C waits for a STOP condition and generates a START condition following. If STA is set while the I ² C is already in the master mode and one or more bytes have been transmitted or received, the I ² C 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 I ² C 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 I ² C device from the bus error state (I2STAT as 00H). In this case, no STOP condition is transmitted to the I ² C bus. If the STA and STO bits are both set and the device is original in the master mode, the I ² C 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 avoided from issuing illegal I ² C frames.

Bit	Name	Description
3	SI	<p>I²C interrupt flag</p> <p>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 I2STAT register to determine which step has been passed and take actions for next step.</p> <p>SI is cleared by software. Before the SI is cleared, the low period of I2C0_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.</p> <p>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.</p>
2	AA	<p>Acknowledge assert flag</p> <p>If the AA flag is set, an ACK (low level on I2C0_SDA) will be returned during the acknowledge clock pulse of the I2C0_SCL line while the I²C device is a receiver or an own-address-matching slave.</p> <p>If the AA flag is cleared, a NACK (high level on I2C0_SDA) will be returned during the acknowledge clock pulse of the I2C0_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 slave address and the General Call. Consequently, SI will not be asserted and no interrupt is requested.</p> <p>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.</p> <p>There is a special case of I2STAT 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.</p>
1	-	Reserved
0	I2CPX	<p>I²C pins select</p> <p>0 = Assign I2C0_SCL to P1.3 and I2C0_SDA to P1.4.</p> <p>1 = Assign I2C0_SCL to P0.2 and I2C0_SDA to P1.6.</p> <p>Note that I²C pins will exchange immediately once setting or clearing this bit.</p>

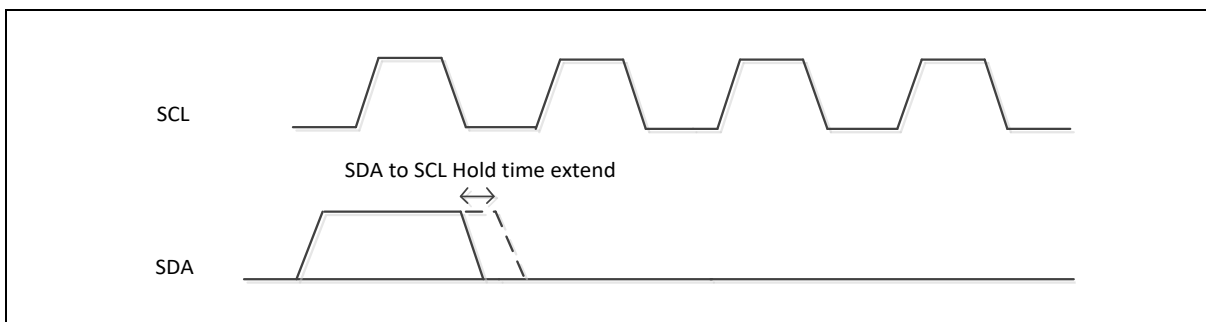


Figure 6.11-15 Hold Time extend enable

I2STAT – I²C Status

Register	SFR Address	Reset Value
I2STAT	BDH, Page 0	1111 1000 b

7	6	5	4	3	2	1	0
I2STAT[7:3]					0	0	0
R					R	R	R

Bit	Name	Description
7:3	I2STAT[7:3]	I ² C status code The MSB five bits of I2STAT contains the status code. There are 27 possible status codes. When I2STAT is F8H, no relevant state information is available and SI flag keeps 0. All other 26 status codes correspond to the I ² C 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 I2STAT are always read as 0.

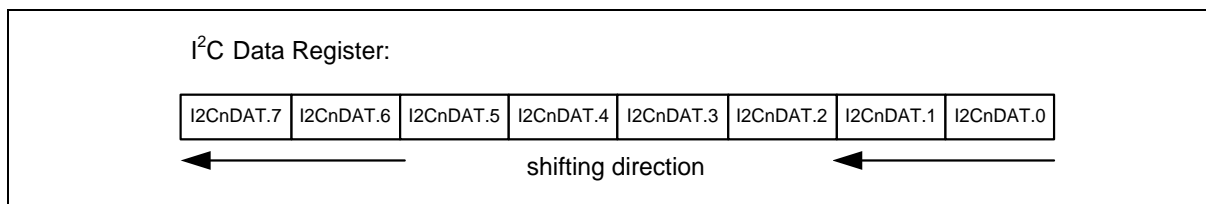
I2DAT – I²C Data

Register	SFR Address	Reset Value
I2DAT	BCH, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
I2DAT[7:0]							
R/W							

Bit	Name	Description
7:0	I2DAT[7:0]	I ² C data I2DAT contains a byte of the I ² C data to be transmitted or a byte, which has just received. Data in I2DAT remains as long as SI is logic 1. The result of reading or writing I2DAT during I ² C transceiving progress is unpredicted. While data in I2DAT is shifted out, data on the bus is simultaneously being shifted in to update I2DAT. I2DAT always shows the last byte that presented on the I ² C bus. Thus the event of lost arbitration, the original value of I2DAT changes after the transaction.

I²C Data Shifting Direction.



I2ADDR – I²C Own Slave Address

Register	SFR Address	Reset Value
I2ADDR	C1H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
I2ADDR[7:1]							GC
R/W							R/W

Bit	Name	Description
7:1	I2ADDR[7:1]	<p>I²C device's own slave address</p> <p><u>In master mode:</u> These bits have no effect.</p> <p><u>In slave mode:</u> 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.</p> <p>Note that I2ADDR[7:1] should not remain its default value of all 0, because address 0x00 is reserved for General Call.</p>
0	GC	<p>General Call bit</p> <p><u>In master mode:</u> This bit has no effect.</p> <p><u>In slave mode:</u> 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.</p>

I2CLK – I²C Clock

Register	SFR Address	Reset Value
I2CLK	BEH, Page 0	0000_1001 b

7	6	5	4	3	2	1	0
I2CLK[7:0]							
R/W							

Bit	Name	Description
7:0	I2CLK[7:0]	<p>I²C clock setting</p> <p><u>In master mode:</u></p> <p>This register determines the clock rate of I²C bus when the device is in a master mode. The clock rate follows the equation,</p> $\frac{F_{SYS}}{4 \times (I2CLK + 1)}$ <p>The default value will make the clock rate of I²C bus 400k bps if the peripheral clock is 16 MHz. Note that the I2CLK value of 00H and 01H are not valid. This is an implement limitation.</p> <p><u>In slave mode:</u></p> <p>This byte has no effect. In slave mode, the I²C device will automatically synchronize with any given clock rate up to 400k bps.</p>

I2TOC – I²C Time-out Counter

Register	SFR Address	Reset Value
I2TOC	BFH, 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
2	I2TOCEN	I ² C time-out counter enable 0 = I ² C time-out counter Disabled. 1 = I ² C time-out counter Enabled. Note: please always enable I ² C interrupt when enable I ² C time-out counter function
1	DIV	I ² C time-out counter clock divider 0 = The clock of I ² C time-out counter is F _{sys} /1. 1 = The clock of I ² C time-out counter is F _{sys} /4.
0	I2TOF	I ² C time-out flag This flag is set by hardware if 14-bit I ² C time-out counter overflows. It is cleared by software.

6.12 Serial Peripheral Interface (SPI)

6.12.1 Overview

The MS51 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 $F_{SYS}/2$, transfer complete and write collision flag. For a multi-master system, SPI supports Master Mode Fault to protect a multi-master conflict.

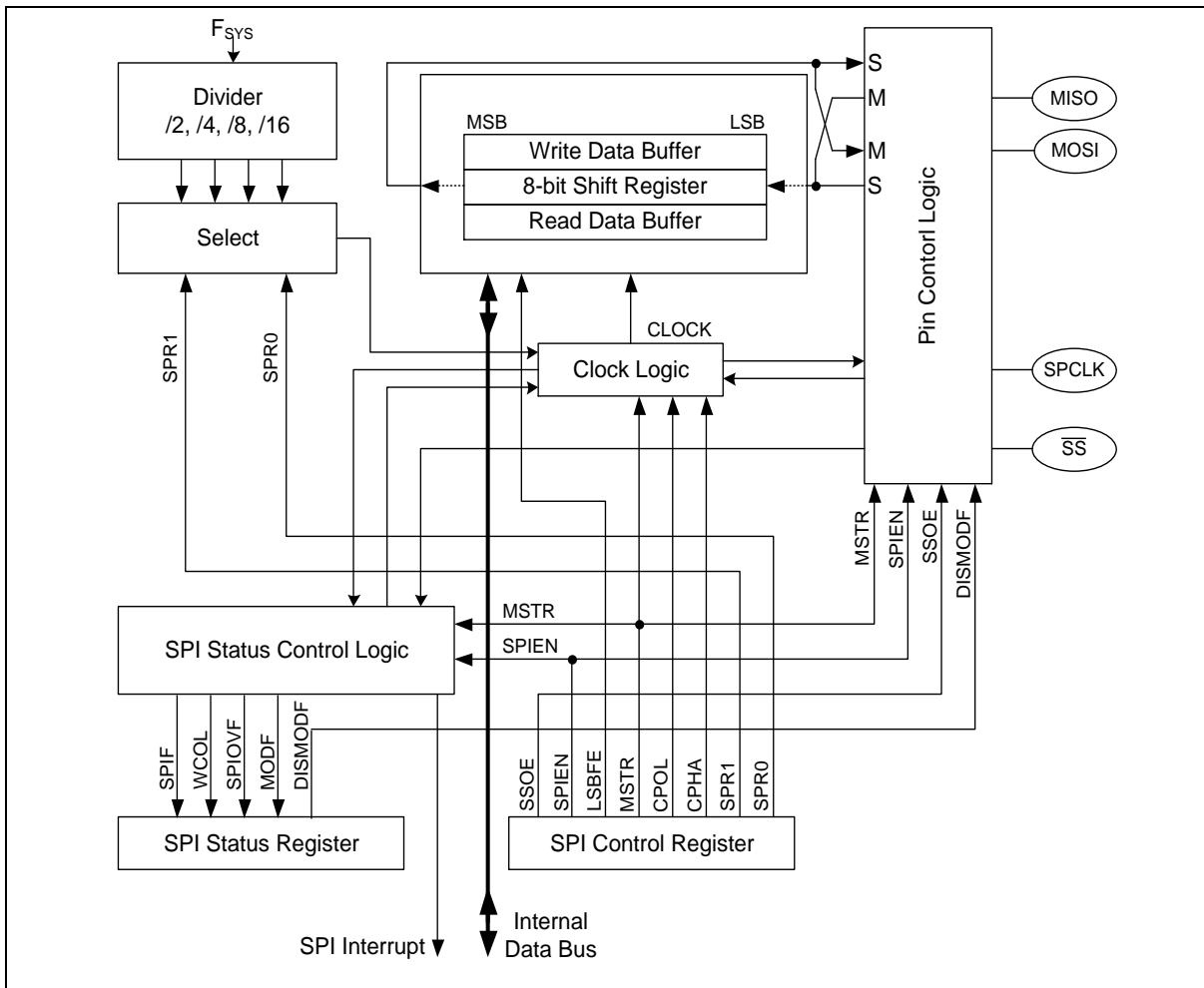


Figure 6.12-1 SPI Block Diagram

Figure 6.12-1 SPI Block Diagram shows SPI block diagram. It 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 write data buffer, a shift out register and a read data buffer. It is double buffered in the receiving and transmit directions. Transmit data can be written to the shifter until when the previous transfer is not 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 four pins of SPI interface 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 chapter 6.12.5 Mode Fault Detection) via software setting if multi-master environment exists. The MS51 also provides auto-activating function to toggle \overline{SS} between each byte-transfer.

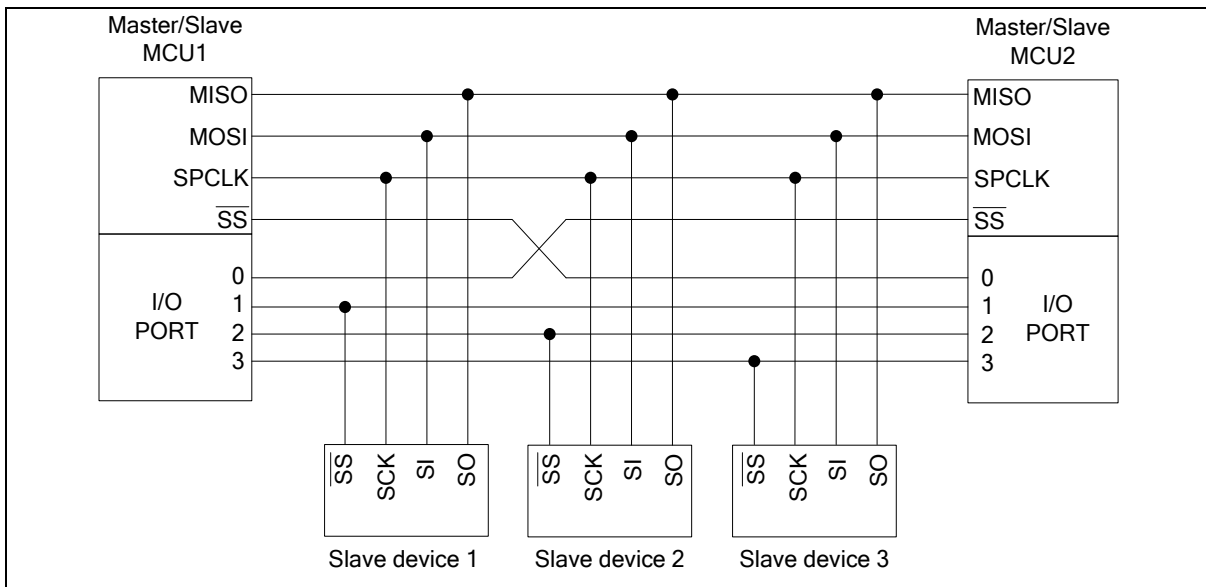


Figure 6.12-2 SPI Multi-Master, Multi-Slave Interconnection

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.

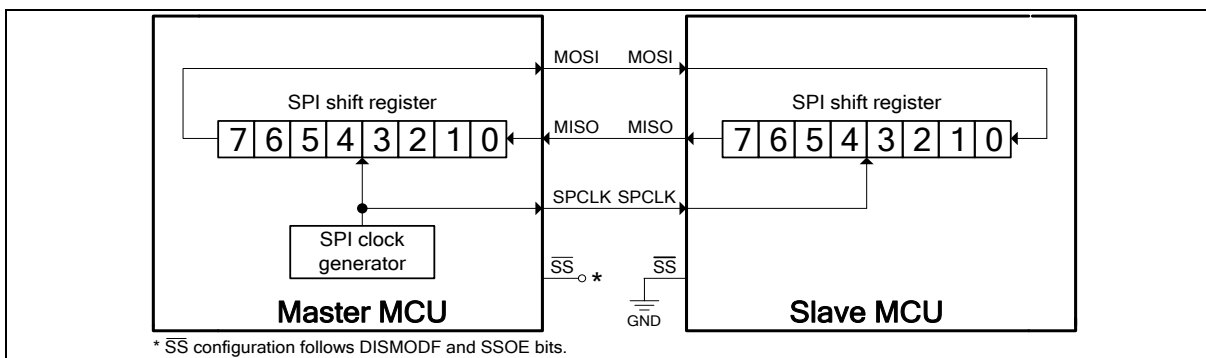


Figure 6.12-3 SPI Single-Master, Single-Slave Interconnection

Figure 6.12-3 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.

By default, SPI data is transferred MSB first. If the LSBFE (SPCR.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.

There are three SPI registers to support its operations, including SPI control register (SPCR), SPI status register (SPSR), and SPI data register (SPDR). These registers provide control, status, data storage functions, and clock rate selection. The following registers relate to SPI function.

6.12.2 Operating Modes

6.12.2.1 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.

6.12.2.2 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.

6.12.3 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.

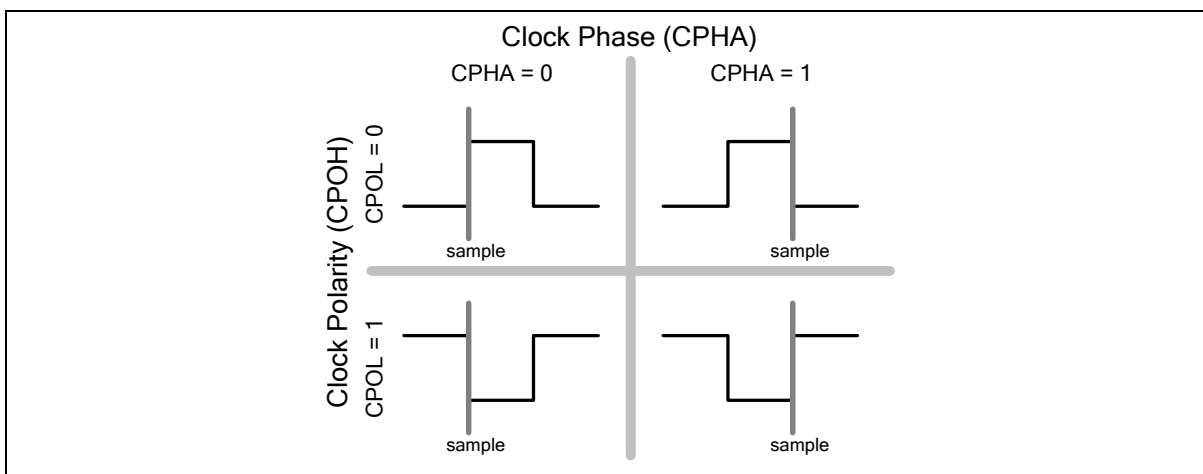


Figure 6.12-4 SPI Clock Formats

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 VSS as long as only CPHA = 1 clock mode is used.

Note: 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.

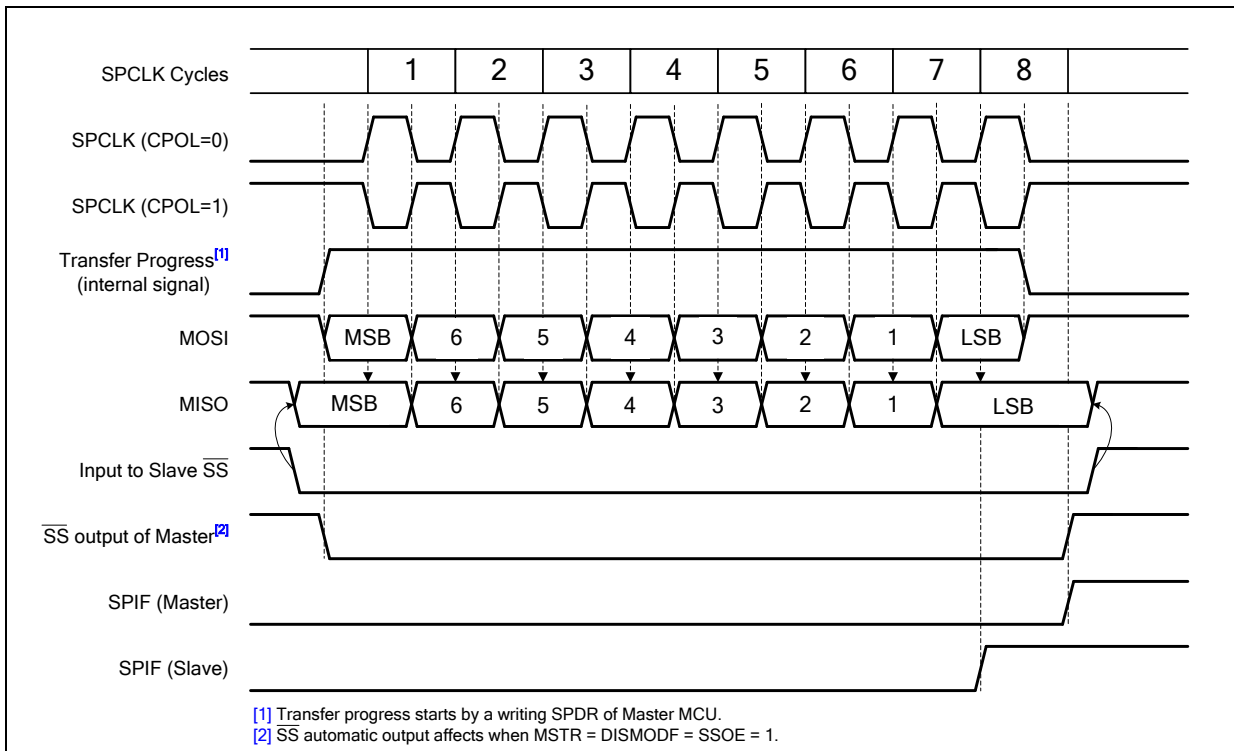


Figure 6.12-5 SPI Clock and Data Format with CPHA = 0

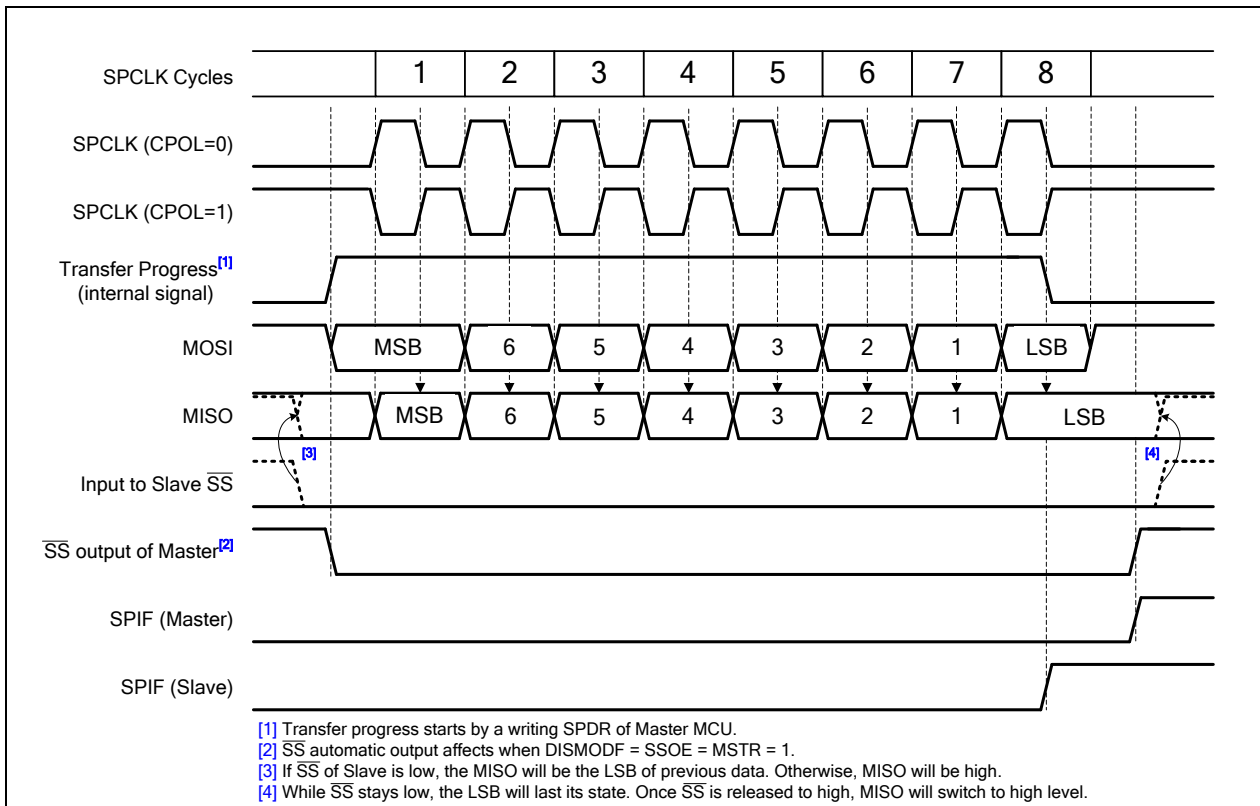


Figure 6.12-6 SPI Clock and Data Format with CPHA = 1

6.12.4 Slave Select Pin Configuration

The MS51 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 an 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.

6.12.5 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 Table 6.12-1 Slave Select Pin Configurations, 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.

6.12.6 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.

6.12.7 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 remain. 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.

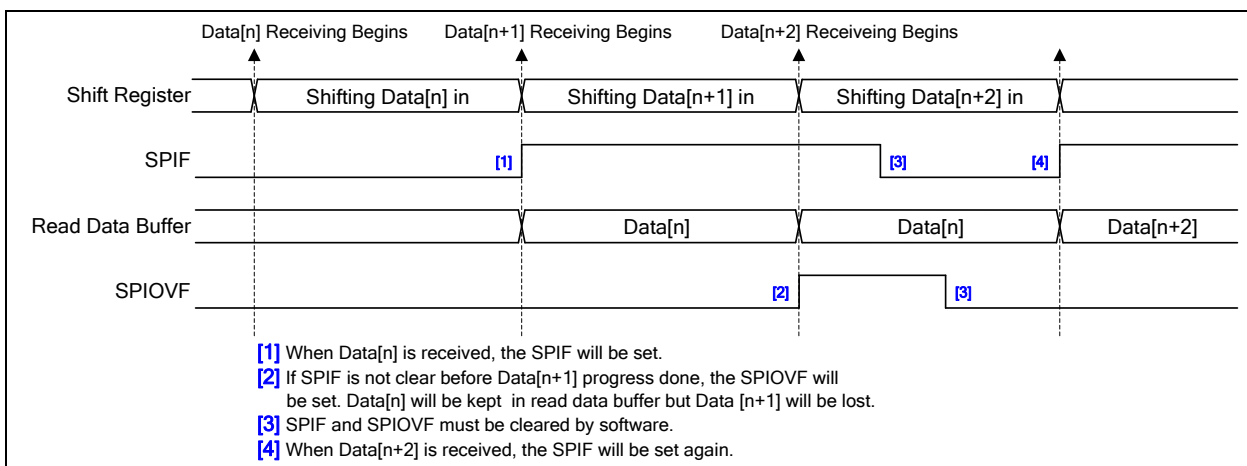


Figure 6.12-7 SPI Overrun Waveform

6.12.8 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.

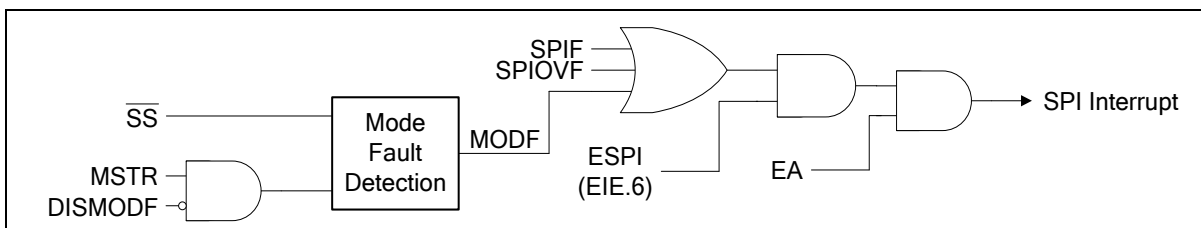


Figure 6.12-8 SPI Interrupt Request

6.12.9 Control Register of SPI

SPCR – Serial Peripheral Control Register

Register	SFR Address	Reset Value
SPCR	F3H, 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	SOE	Slave select output enable This bit is used in combination with the DISMODF (SPSR.3) bit to determine the feature of \overline{SS} pin as shown in Table 6.12-1 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	PIEN	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	STR	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	OL	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]	<p>SPI clock rate select</p> <p>These two bits select four grades of SPI clock divider. The clock rates below are illustrated under $F_{SYS} = 16$ MHz condition.</p> <p>$F_{sys} = 16$ MHz</p> <table border="1"> <thead> <tr> <th>SPR1</th> <th>SPR0</th> <th>Divider</th> <th>SPI clock rate</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>2</td> <td>8M bit/s</td> </tr> <tr> <td>0</td> <td>1</td> <td>4</td> <td>4M bit/s</td> </tr> <tr> <td>1</td> <td>0</td> <td>8</td> <td>2M bit/s</td> </tr> <tr> <td>1</td> <td>1</td> <td>16</td> <td>1 M bit/s</td> </tr> </tbody> </table> <p>$F_{sys} = 24$ MHz</p> <table border="1"> <thead> <tr> <th>SPR1</th> <th>SPR0</th> <th>Divider</th> <th>SPI clock rate</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>2</td> <td>12M bit/s</td> </tr> <tr> <td>0</td> <td>1</td> <td>4</td> <td>6M bit/s</td> </tr> <tr> <td>1</td> <td>0</td> <td>8</td> <td>3M bit/s</td> </tr> <tr> <td>1</td> <td>1</td> <td>16</td> <td>1.5M bit/s</td> </tr> </tbody> </table> <p>SPR[1:0] 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 $F_{SYS}/2$ communication speed.</p>	SPR1	SPR0	Divider	SPI clock rate	0	0	2	8M bit/s	0	1	4	4M bit/s	1	0	8	2M bit/s	1	1	16	1 M bit/s	SPR1	SPR0	Divider	SPI clock rate	0	0	2	12M bit/s	0	1	4	6M bit/s	1	0	8	3M bit/s	1	1	16	1.5M bit/s
SPR1	SPR0	Divider	SPI clock rate																																							
0	0	2	8M bit/s																																							
0	1	4	4M bit/s																																							
1	0	8	2M bit/s																																							
1	1	16	1 M bit/s																																							
SPR1	SPR0	Divider	SPI clock rate																																							
0	0	2	12M bit/s																																							
0	1	4	6M bit/s																																							
1	0	8	3M bit/s																																							
1	1	16	1.5M bit/s																																							

SPCR2 – Serial Peripheral Control Register 2

Register	SFR Address	Reset Value
SPCR2	F3H, Page 1	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	-	SPIS1	SPIS0
-	-	-	-	-	-	R/W	R/W

Bit	Name	Description																																				
7:2	-	Reserved																																				
1:0	SPIS[1:0]	<p>SPI Interval time selection between adjacent bytes SPIS[1:0] and CPHA select eight grades of SPI interval time selection between adjacent bytes. As below table:</p> <table border="1"> <thead> <tr> <th>CPHA</th> <th>SPIS1</th> <th>SPIS0</th> <th>SPI clock</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0.5</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1.0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>1.5</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>2.0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1.0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>1.5</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>2.0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>2.5</td> </tr> </tbody> </table> <p>SPIS[1:0] are valid only under Master mode (MSTR = 1).</p>	CPHA	SPIS1	SPIS0	SPI clock	0	0	0	0.5	0	0	1	1.0	0	1	0	1.5	0	1	1	2.0	1	0	0	1.0	1	0	1	1.5	1	1	0	2.0	1	1	1	2.5
CPHA	SPIS1	SPIS0	SPI clock																																			
0	0	0	0.5																																			
0	0	1	1.0																																			
0	1	0	1.5																																			
0	1	1	2.0																																			
1	0	0	1.0																																			
1	0	1	1.5																																			
1	1	0	2.0																																			
1	1	1	2.5																																			

Table 6.12-1 Slave Select Pin Configurations

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	

SPSR – Serial Peripheral Status Register

Register	SFR Address	Reset Value
SPSR	F4H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
SPIF	WCOL	SPIOVF	MODF	DISMODF	TXBUF	-	-
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 (EIE .0) and EA are enabled, an SPI interrupt will be required. This bit should be cleared via software. Attempting to write to SPDR 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 (SPCR.7) bit to determine the feature of \overline{SS} pin as shown in Table 6.12-1 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	TXBUF	SPI writer data buffer status This bit indicates the SPI transmit buffer status. 0 = SPI writer data buffer is empty 1 = SPI writer data buffer is full.

SPDR – Serial Peripheral Data Register

Register	SFR Address	Reset Value
SPDR	F5H, Page 0	0000_0000 b

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

Bit	Name	Description
7:0	SPDR[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 12-Bit Analog-To-Digital Converter (ADC)

6.13.1 Overview

The MS51 32K 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 MS51 32K series is selected as 8-channel inputs in single end mode. The internal band-gap voltage 1.22 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 continuous conversion and storage result data into XRAM.

6.13.2 Functional Description

6.13.2.1 ADC Operation

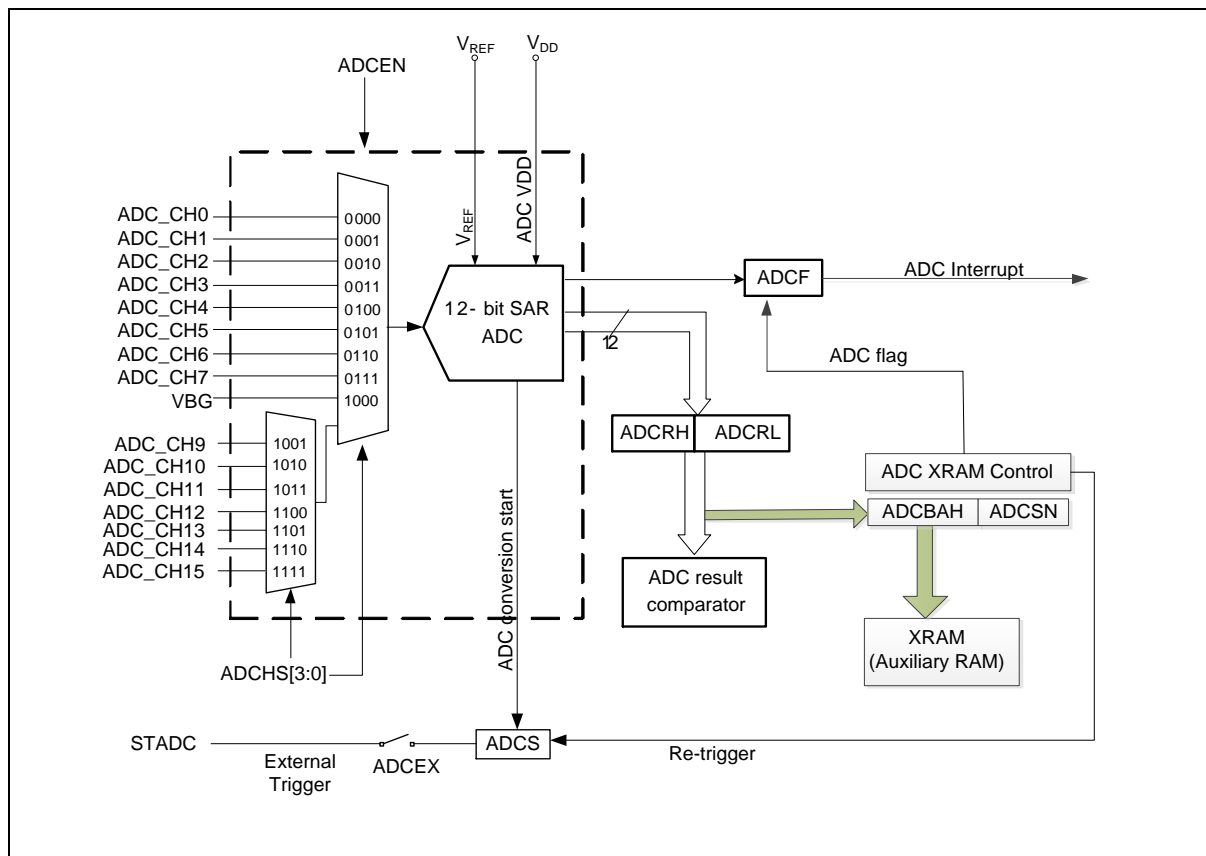


Figure 6.13-1 12-bit ADC Block Diagram

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 AINDIDS0 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.

Following shows the multi function define of ADC.

Group	Pin Name	GPIO	Description
ADC	ADC_CH0	P1.7	ADC channel 0 analog input.
	ADC_CH1	P3.0	ADC channel 1 analog input.
	ADC_CH2	P0.7	ADC channel 2 analog input.
	ADC_CH3	P0.6	ADC channel 3 analog input.
	ADC_CH4	P0.5	ADC channel 4 analog input.
	ADC_CH5	P0.4	ADC channel 5 analog input.
	ADC_CH6	P0.3	ADC channel 6 analog input.
	ADC_CH7	P1.1	ADC channel 7 analog input.
	ADC_CH9	P2.1	ADC channel 8 analog input.
	ADC_CH10	P2.2	ADC channel 10 analog input.
	ADC_CH11	P2.3	ADC channel 11 analog input.
	ADC_CH12	P2.4	ADC channel 12 analog input.
	ADC_CH13	P1.3	ADC channel 13 analog input.
	ADC_CH14	P1.4	ADC channel 14 analog input.
	ADC_CH15	P2.5	ADC channel 15 analog input.

6.13.2.2 ADC Conversion Triggered by External Source

Besides setting ADCS via software, the MS51 32K 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 triggering ADC hardware with controllable trigger delay makes the MS51 32K 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.

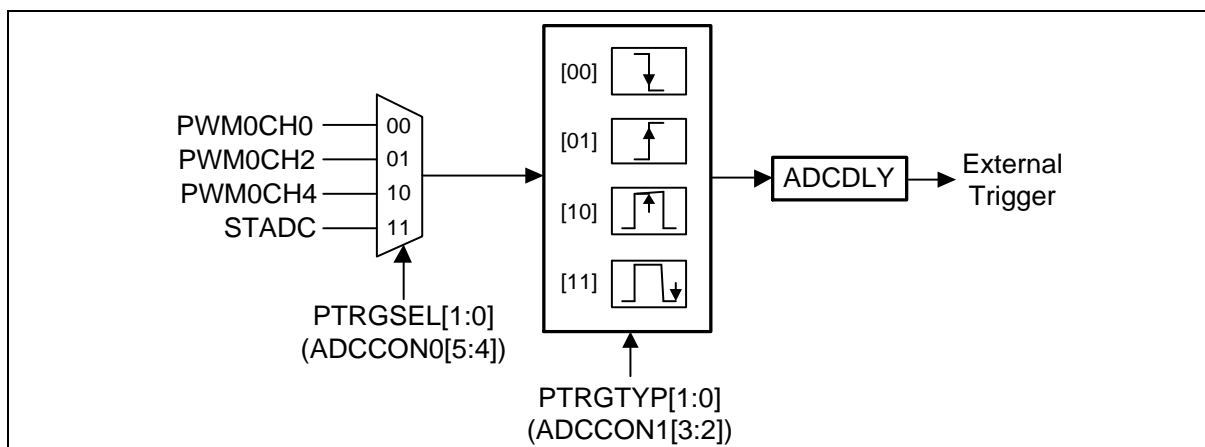


Figure 6.13-2 External Triggering ADC Circuit

6.13.2.3 ADC Conversion Result Comparator

The MS51 32K series ADC has a digital comparator, which compares the A/D conversion result with a 12-bit constant value given in ACMPL and ADCMPH 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: For the MS51 series, 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.

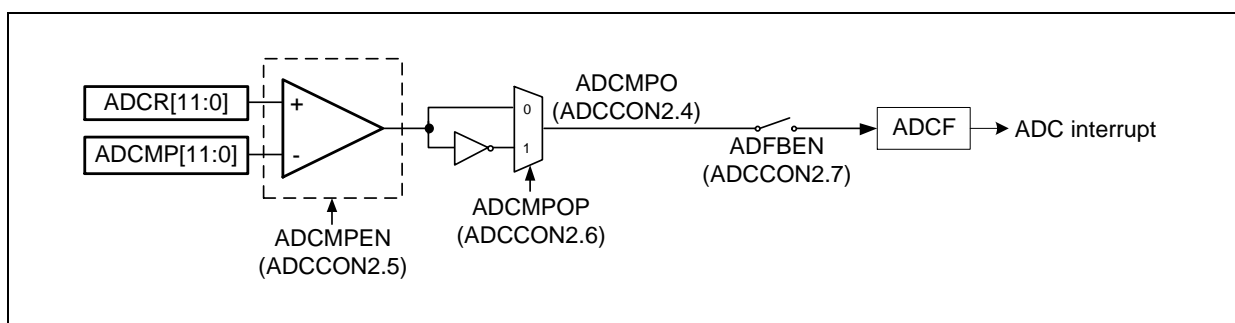


Figure 6.13-3 ADC Result Comparator

6.13.2.4 Internal Band-gap

At room temperature, all MS51 band-gap voltage values will be calibrated within the range of 1.17V to 1.30V. If you want to get the actual band-gap value for MS51, read the 2 bytes value after the UID address and the actually valid bit is 12. The first byte is the upper 8 bits, and the lower 4 bits of the second byte are the lower 4 bits of the 12 bit.

Reading and calculation steps:

- Read a band-gap value with IAP by reading UID;
- Merge the upper 8 bits and the lower 4 bits define as mark READ_BANDGAP;
- Use the following formula to convert to an actual voltage value.

Formula as following

$$Bandga_Voltage = \frac{READ_BANDGAP \times 3072}{4096} (mV)$$

For example:

Read the 2 bytes value after the UID address, wherein the first byte value is 0x64, and the second byte value is 0x0E, merged as 0x64E = 1614. The conversion result is as follows:

$$Bandgap_Voltage = \frac{1614 \times 3072}{4096} = 1210.5(mV)$$

Band-gap as ADC input to calculate the V_{DD} value:

MS51 internal embedded band-gap voltage also can be the internal ADC input. This input is useful to measure Vref value then means can know the V_{DD} value from ADC convert result

$$V_{DD} = \frac{4096}{ADC_Bandga_Result} \times Bandgap_Voltage \text{ or } V_{DD} = \frac{READ_BANDGAP \times 3072}{ADC_Bandgap_Result} (mV)$$

6.13.2.5 ADC Continues Conversion

The ADC controller supports continues convert function, which auto store the A/D conversion result. The ADC continues 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. Considering to 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.

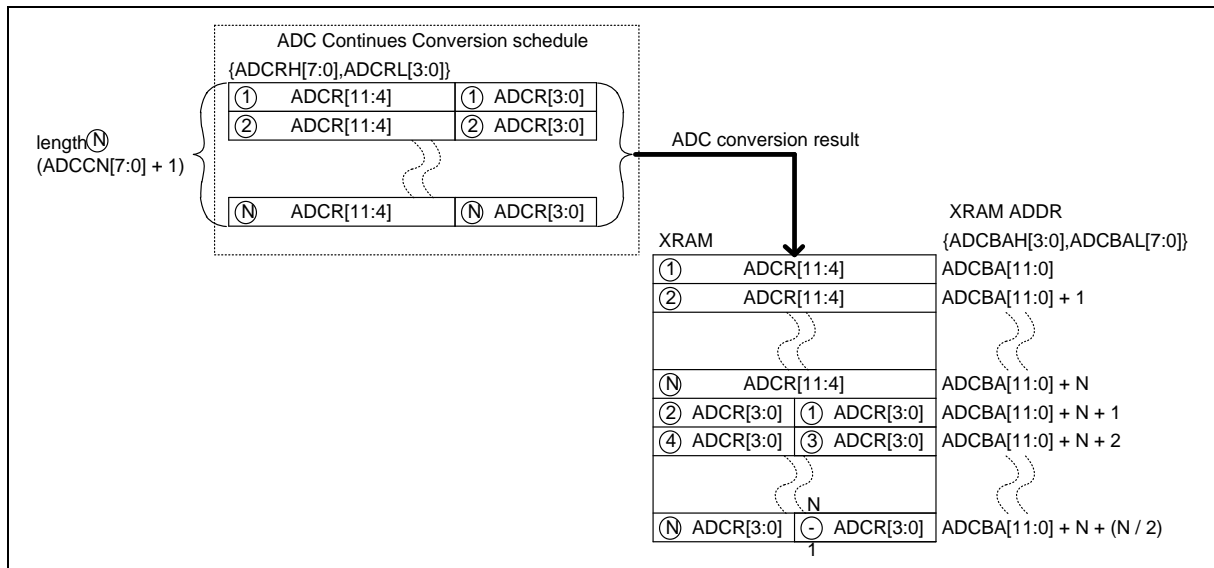


Figure 6.13-4 ADC Continues Mode

A programming 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).

6.13.3 Control Registers of ADC

ADCCON0 – ADC Control 0 (Bit-addressable)

Register	SFR Address	Reset Value
ADCCON0	E8H, all page	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. 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. <u>Writing:</u> 0 = No effect. 1 = Start an A/D converting. <u>Reading:</u> 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 = PWM0. 01 = PWM2. 10 = PWM4 11 = STADC pin.
3:0	ADCHS[3:0]	A/D converting channel select This field 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). 1001 = ADC_CH9. 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
OCEN	STADCPX	ADCDIV[1:0]		ETGTYP[1:0]		ADCEX	ADCEN
R/W	R/W	R/W		R/W		R/W	R/W

Bit	Name	Description
7	OCEN	ADC Offset Calibration Enable register This field is used to enable offset calibration function. 0: ADC Offset Calibration is enabled, auto-calibration by ADC hardware. 1: ADC Offset Calibration is disabled
6	STADCPX	External start ADC trigger pin select 0 = Assign STADC to P0.4. 1 = Assign STADC to P1.3. Note that STADC will exchange immediately once setting or clearing this bit.
5:4	ADCDIV[1:0]	ADC ADCAQT clock divider This field define sampoing acquisition time clock divider value 00 = $F_{ADCAQT} = F_{SYS}/1$. 01 = $F_{ADCAQT} = F_{SYS}/2$. 10 = $F_{ADCAQT} = F_{SYS}/4$. 11 = $F_{ADCAQT} = F_{SYS}/8$.
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 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	ADCAQT0[2:0]			ADCDLY.8
R/W	R/W	R/W	R	R/W			R/W

Bit	Name	Description
7	ADFBEN	<p>ADC compare result asserting Fault Brake enable</p> <p>0 = ADC asserting Fault Brake Disabled.</p> <p>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.</p>
6	ADCMPOP	<p>ADC comparator output polarity</p> <p>0 = ADCMPO is 1 if ADCR[11:0] is greater than or equal to ADCMP[11:0].</p> <p>1 = ADCMPO is 1 if ADCR[11:0] is less than ADCMP[11:0].</p>
5	ADCMPEN	<p>ADC result comparator enable.</p> <p>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.</p> <p>The ADCF register changes to 1 only when ADC comparing result matches the condition and then enters interrupt vector if ADC interrupt is enabled.</p> <p>0 = ADC result comparator trig ADCF Disabled.</p> <p>1 = ADC result comparator trig ADCF Enabled.</p> <p>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</p>
4	ADCMPO	<p>ADC comparator output value</p> <p>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.</p>
3:1	ADCAQT0[2:0]	<p>ADC acquisition time 1</p> <p>This 3-bit field decides the acquisition time for ADC AIN0~AIN7 and BG sampling, following by equation below:</p> $\text{ADC acquisition time} = \frac{4 * \text{ADCAQT0} + 6}{F_{\text{ADCAQT}}}$ <p>F_{ADCAQT} define is in ADCDIV (ADCCON2[3:1])</p> <p>The default and minimum acquisition time is 6 ADC clock cycles. Note that this field should not be changed when ADC is in converting.</p>
0	ADCDLY.8	<p>ADC external trigger delay counter bit 8</p> <p>See ADCDLY register.</p>

ADCCON3 – ADC Control 3

Register	SFR Address	Reset Value
ADCCON3	86H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	HIE	CONT	ADCAQT1[2:0]			SLOW
-	-	R/W	R/W	R/W			R/W

Bit	Name	Description
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
3:1	ADCAQT1	ADC acquisition time 1 This 3-bit field decides the acquisition time for ADC AIN9~AIN15 sampling, following by equation below: $\text{ADC acquisition time} = \frac{4 * \text{ADCAQT1} + 6}{F_{\text{ADCAQT}}}$, F_{ADCAQT} is defined in ADCDIV (ADCCON2[3:1]) The default and minimum acquisition time is 6 ADC clock cycles. Note that this field should not be changed when ADC is in converting.
0	SLOW	ADC Slow Speed Selection This bit is used to select ADC low speed. 0 = ADC convert time is high speed 1400ns, $F_{\text{ADC}} = 714$ ksps 1 = ADC convert time is low speed 4750ns, $F_{\text{ADC}} = 215$ ksps, Note1 : slow speed ADC convert time will improvement ADC convert performance when VDD is under 2.5V. Note2: AUXR1.4 (SLOW) or ADCCON3 .0(SLOW) is set to 1, ADC enter Slow speed mode.

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]	<p>ADC external trigger delay counter low byte</p> <p>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.</p> <p>External trigger delay time = $\frac{ADCDLY}{F_{ADC}}$.</p> <p>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.</p>

AINDIDS0 – ADC Channel Digital Input Disconnect

Register	SFR Address	Reset Value
AINDIDS0	F6H, Page 0	0000_0000 b

7	6	5	4	3	2	1	0
P11DIDS	P03DIDS	P04DIDS	P05DIDS	P06DIDS	P07DIDS	P30DIDS	P17DIDS
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	99H, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
P25DIDS	P14DIDS	P13DIDS	P24DIDS	P23DIDS	P22DIDS	P21DIDS	-

Bit	Name	Description
n	PnnDIDS	ADC Channel digital input disable 0 = ADC channel n digital input Enabled. 1 = ADC channel n digital input Disabled. 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.

ADCMP[7:0] – ADC Compare Low Byte

Register	SFR Address	Reset Value
ADCMP[7:0]	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	84H, Page 2	0000_0000 b

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

Bit	Name	Description
7:4	-	Reserved
3:0	ADCBAH[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	85H, Page 2	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]}

ADCSN – ADC Sampling Number

Register	SFR Address	Reset Value
ADCSN	8DH, Page 2	0000_0000 b

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

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

ADCCN – ADC Current Sampling Number

Register	SFR Address	Reset Value
ADCCN	8EH, Page 2	0000_0000 b

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

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

ADCSR – ADC Status Register

Register	SFR Address	Reset Value
ADCSR	8FH, Page 2	0000_0000 b

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

Bit	Description	
[7:3]	Reserved	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..

AUXR1 – Auxiliary Register 1

Register	SFR Address	Reset Value
AUXR1	A2H , Page 0	POR: 0000 0000b, Software reset: 1U00 0000b, nRESET pin: U100 0000b, Others: UUUU 0000b

7	6	5	4	3	2	1	0
SWRF	RSTPINF	HardF	SLOW	GF2	UART0PX	0	DPS
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

Bit	Name	Description
4	SLOW	<p>ADC Slow Speed Selection</p> <p>This bit is used to select ADC low speed.</p> <p>0 = ADC convert time is high speed 1400ns, $F_{ADC} = 714$ ksp/s</p> <p>1 = ADC convert time is low speed 4750ns, $F_{ADC} = 215$ ksp/s,</p> <p>Note1 : slow speed ADC convert time will improvement ADC convert performance when VDD is under 2.5V.</p> <p>Note2: AUXR1.4 (SLOW) or ADCCON3 .0(SLOW) is set to 1, ADC enter Slow speed mode.</p>

6.14 Auxiliary Features

6.14.1 Dual DPTRs

The original 8051 contains one DPTR (data pointer) only. With single DPTR, it is difficult to move data from one address to another with wasting code size and low performance. The MS51 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 (AUXR1.0) 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 AUXR1` instruction is the shortest (2 bytes) instruction to accomplish DPTR toggling rather than `ORL` or `ANL`. For AUXR1.1 contains a hard-wired 0, it allows toggling of the DPS bit by incrementing AUXR1 without interfering with other bits in the register.

```

MOV    R0, #64           ;number of bytes to move
MOV    DPTR, #D_Addr    ;load destination address
INC    AUXR1            ;change active DPTR
MOV    DPTR, #S_Addr    ;load source address
LOOP:
MOVX   A, @DPTR         ;read source data byte
INC    AUXR1            ;change DPTR to destination
MOVX   @DPTR, A         ;write data to destination
INC    DPTR             ;next destination address
INC    AUXR1            ;change DPTR to source
INC    DPTR             ;next source address
DJNZ   R0, LOOP
INC    AUXR1            ;(optional) restore DPS
    
```

AUXR1 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
DPL[7:0]							
R/W							

Bit	Name	Description
7:0	DPL[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 Byte

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

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

Bit	Name	Description
7:0	DPH[7:0]	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.

AUXR1 – Auxiliary Register 1

Register	SFR Address	Reset Value
AUXR1	A2H , Page 0	POR: 0000 0000b, Software reset: 1U00 0000b, nRESET pin: U100 0000b, Others: UUUU 0000b

7	6	5	4	3	2	1	0
SWRF	RSTPINF	HardF	SLOW	GF2	UART0PX	0	DPS
R/W	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.
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.

7 APPLICATION CIRCUIT

7.1 Power Supply Scheme

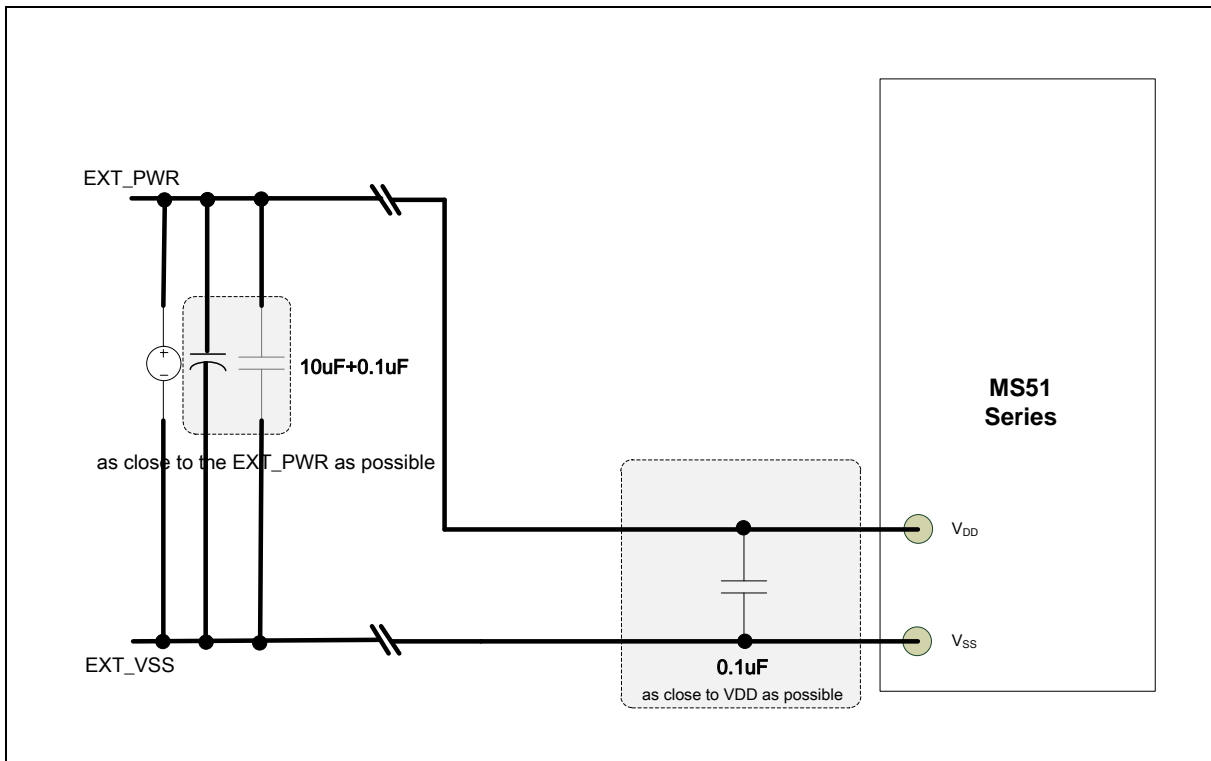


Figure 7.1-1 NuMicro® MS51 Power supply circuit

7.2 Peripheral Application Scheme

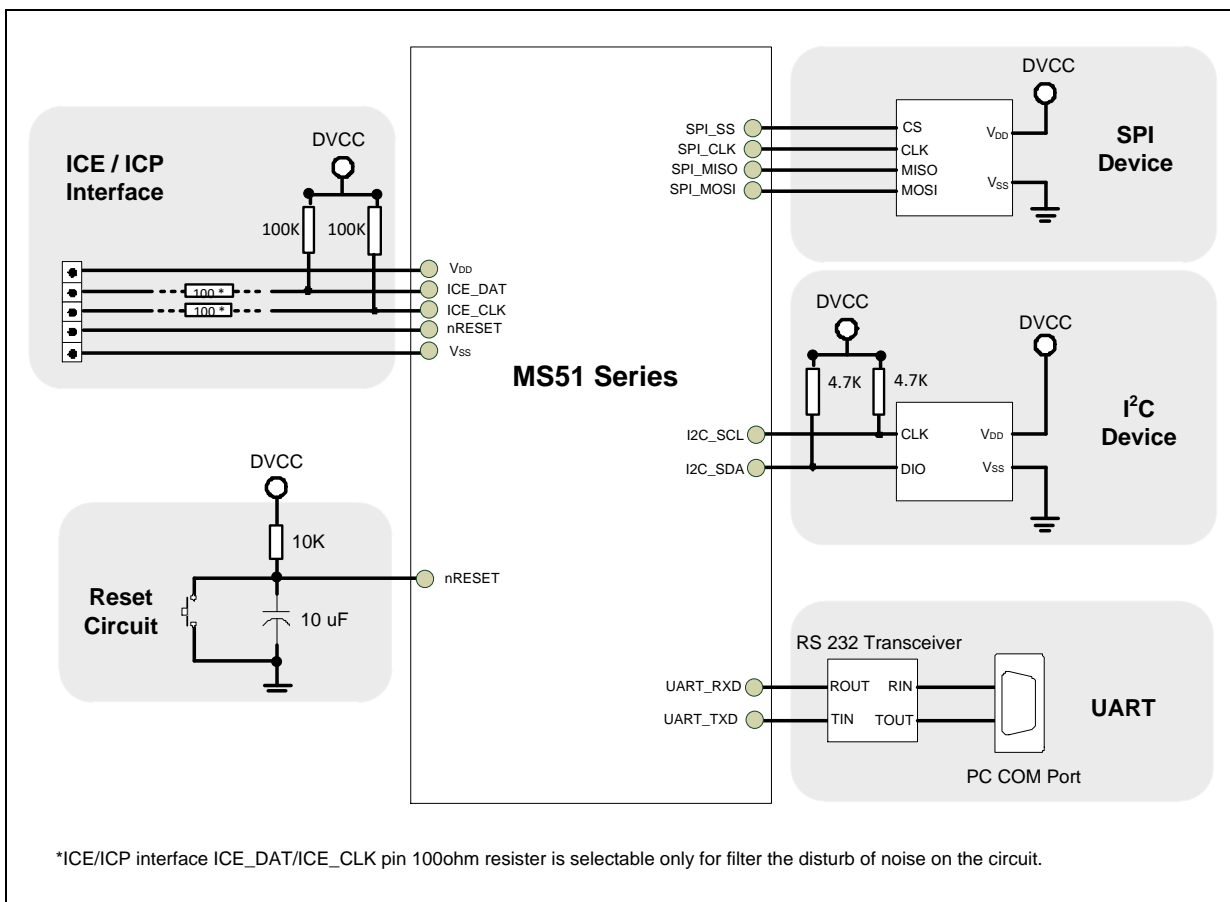


Figure 7.2-1 NuMicro® MS51 Peripheral interface circuit

8 ELECTRICAL CHARACTERISTICS

Please refer to the relative Datasheet for detailed information about the MS51 electrical characteristics.

9 PACKAGE DIMENSIONS

9.1 QFN 33-pin (4.0 x 4.0 x 0.8 mm)

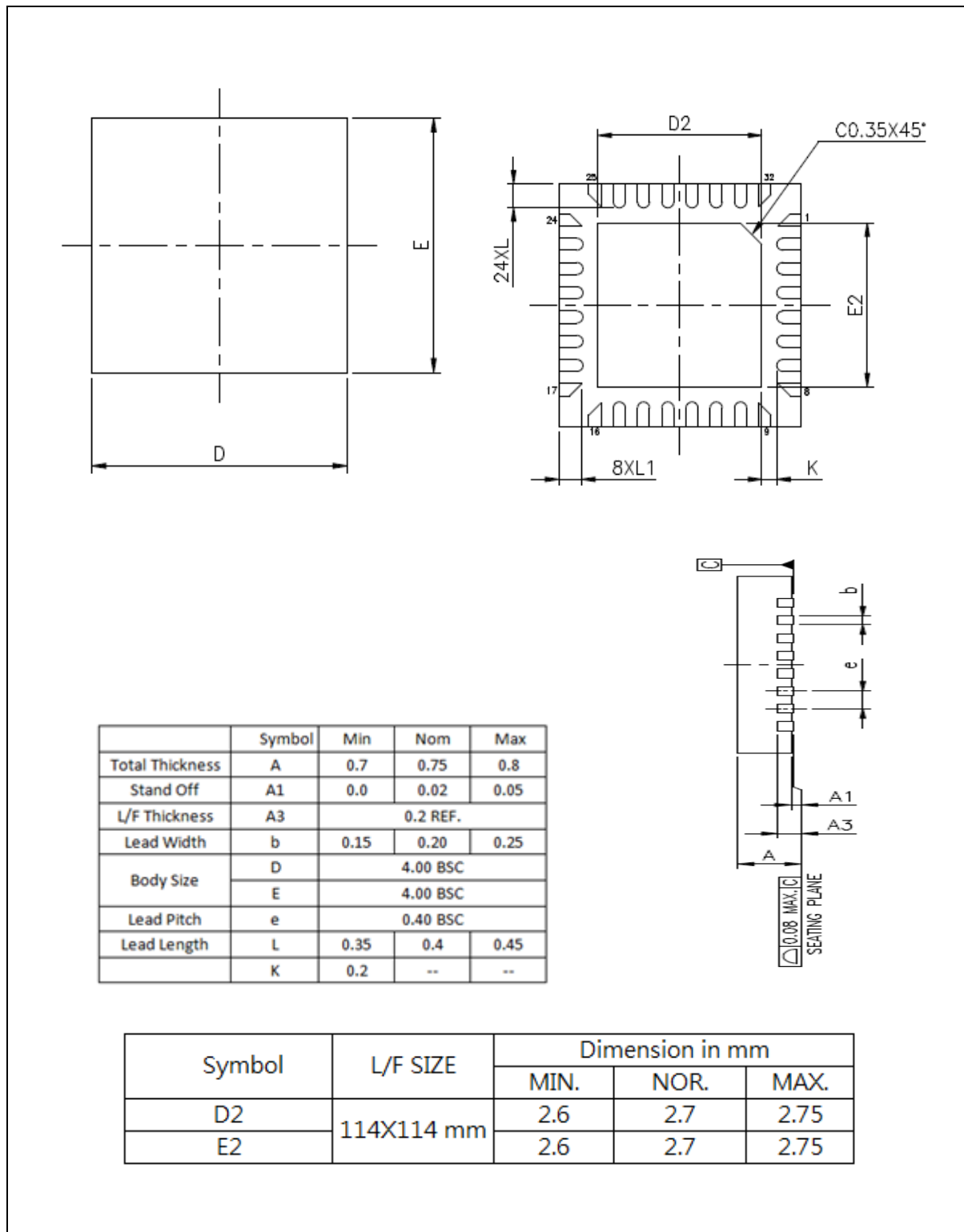


Figure 9.1-1 QFN-33 Package Dimension

9.2 LQFP 32-pin (7.0 x 7.0 x 1.4 mm)

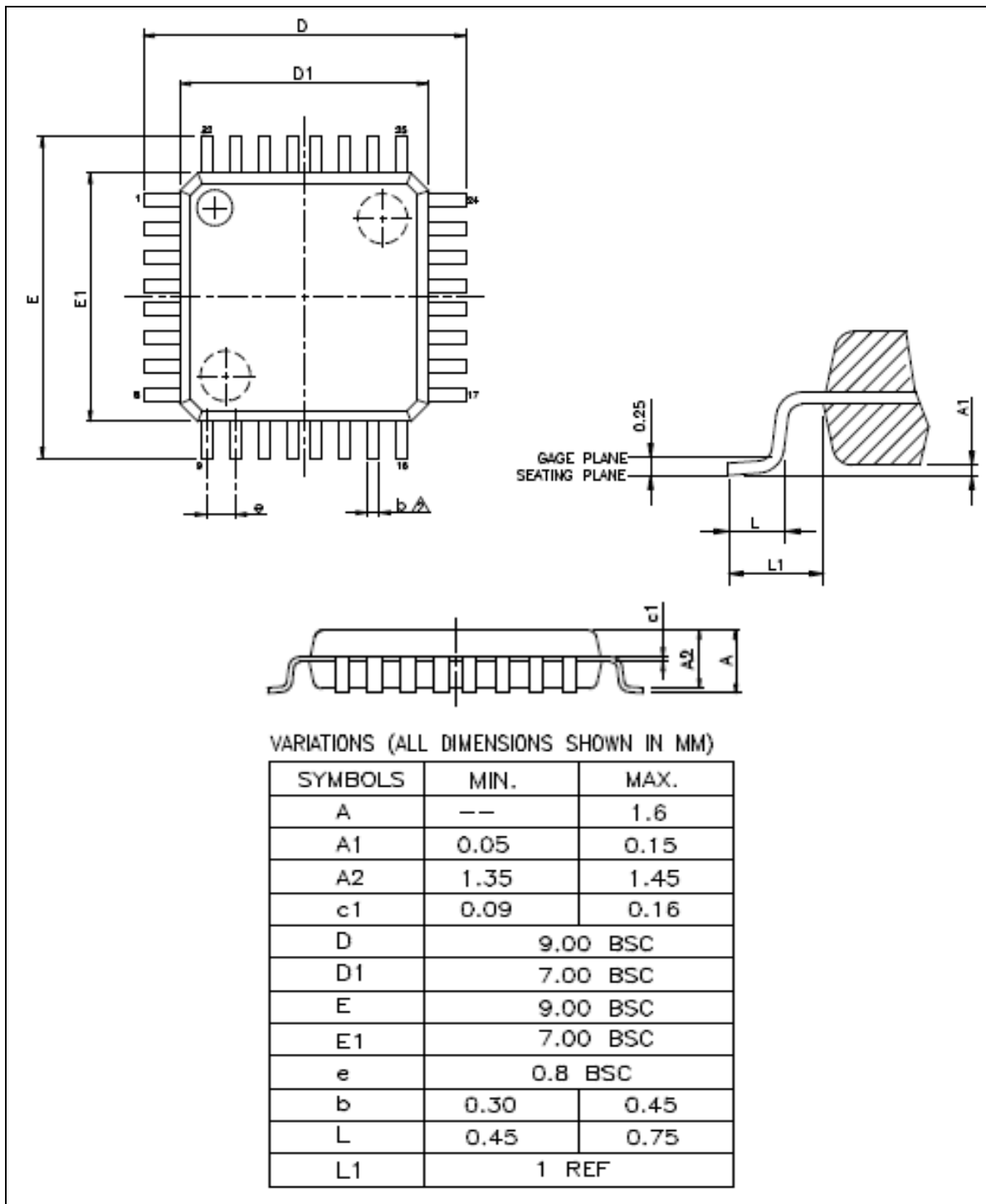


Figure 9.2-1 LQFP-32 Package Dimension

9.3 TSSOP 28-pin (4.4 x 9.7 x 1.0 mm)

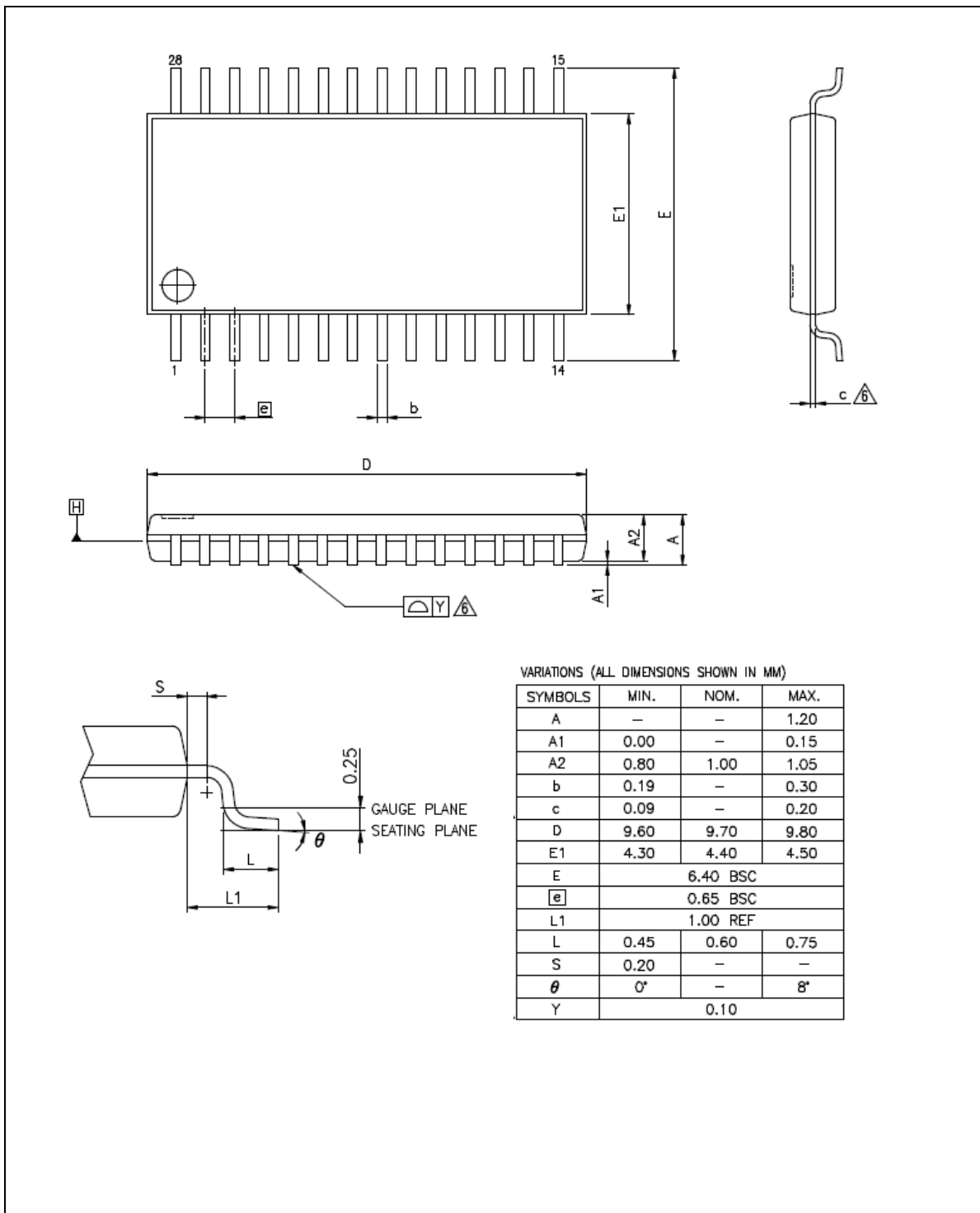


Figure 9.3-1 TSSOP-28 Package Dimension

9.4 TSSOP 20-pin (4.4 x 6.5 x 0.9 mm)

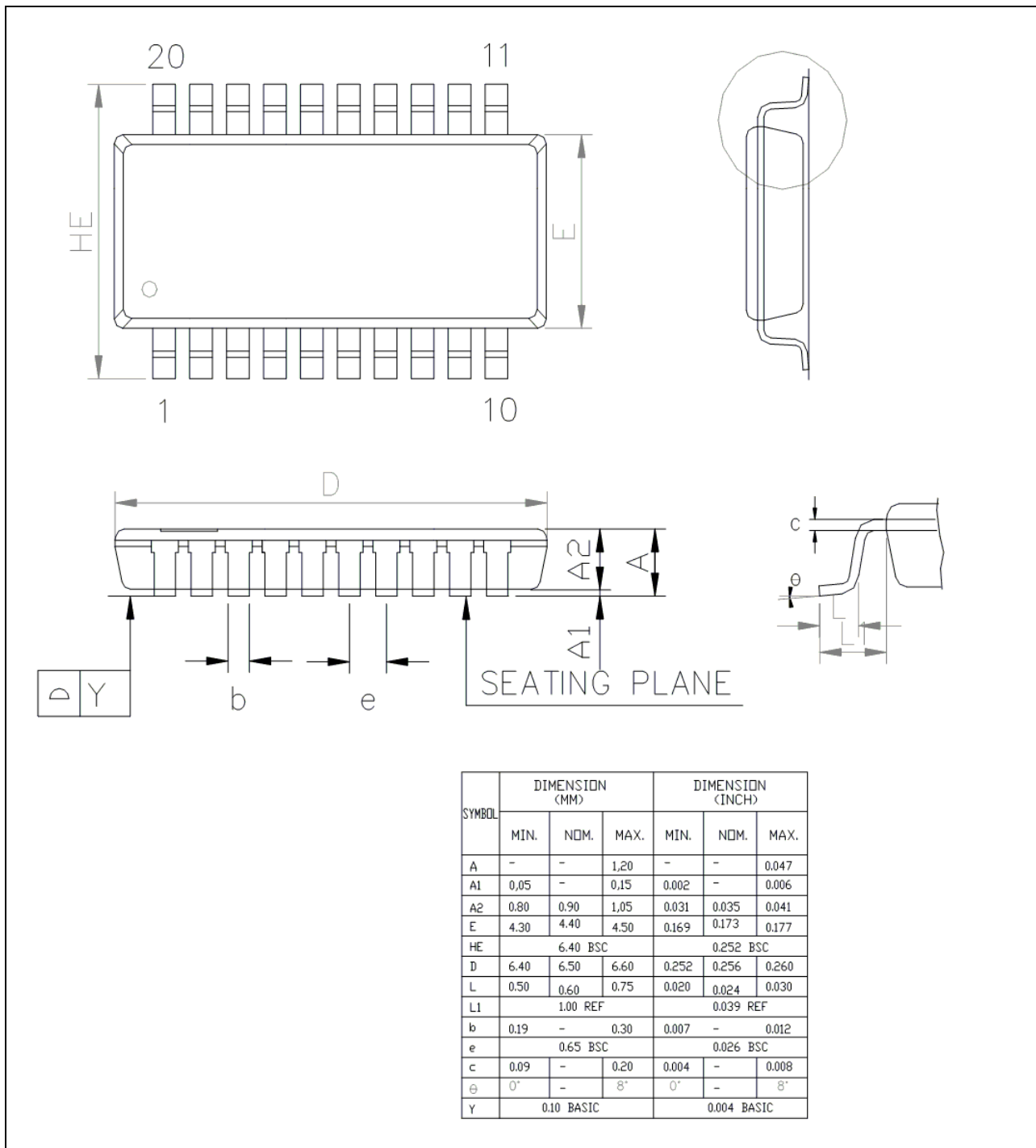


Figure 9.4-1 TSSOP-20 Package Dimension

9.5 QFN 20-pin (3.0 x 3.0 x 0.5mm)

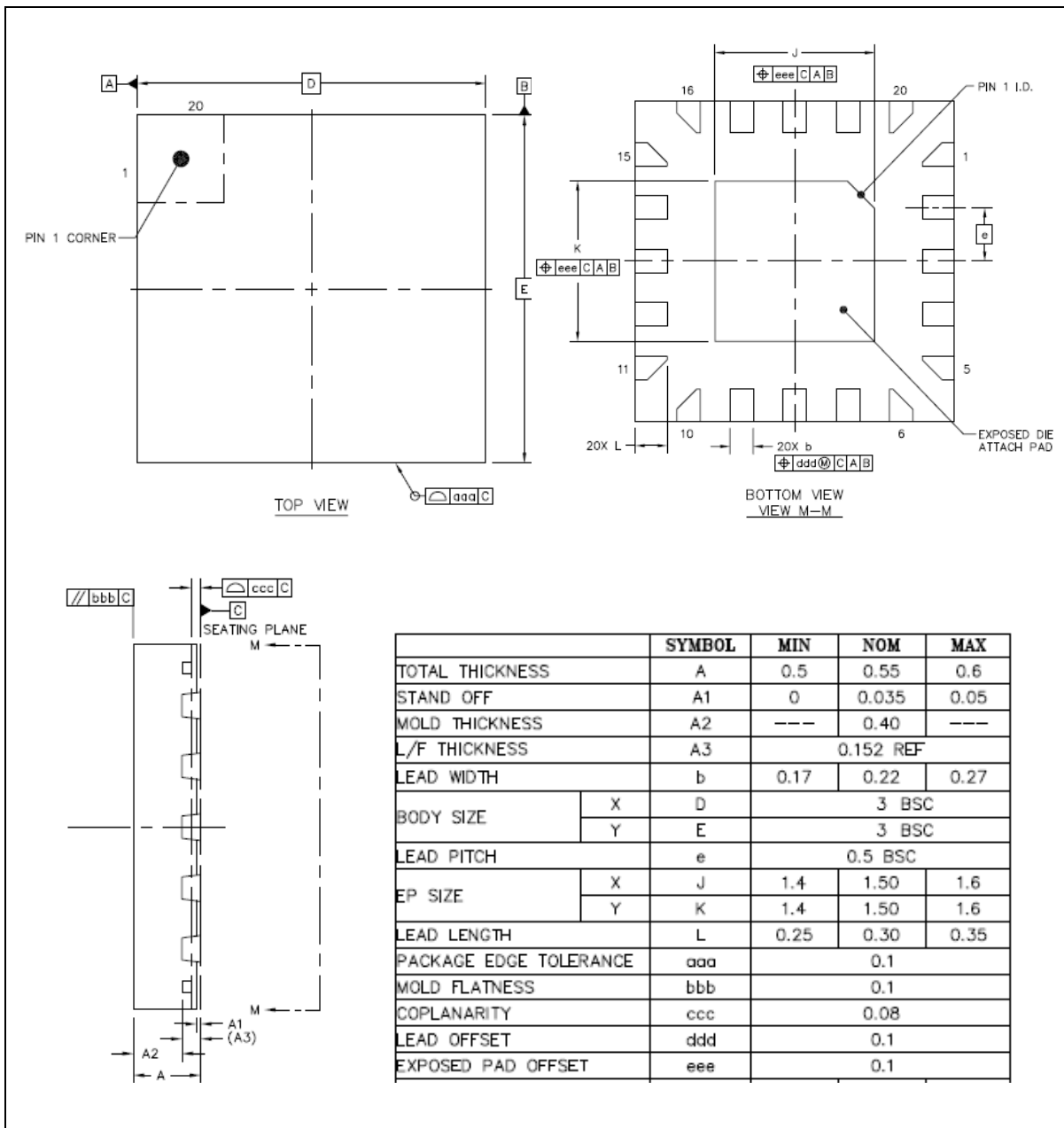


Figure 9.5-1 QFN-20 Package Dimension

10 ABBREVIATIONS

10.1 Abbreviations List

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 INSTRUCTION SET

The MS51 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 MS51 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.

11.1 Instruction Set And Addressing Modes

Table 11.3-1 Instruction Set lists all instructions for details. The note of the instruction set and addressing modes are shown below.

Instruction	Addressing Modes
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 11.1-1 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 11.1-2 Instructions Affect Flag Settings

11.2 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.

<u>Instruction</u>	<u>Description</u>
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.

11.3 Instruction Set

Instruction	OPCODE	Bytes	Clock Cycles	MS51 V.S. 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
INC A	04	1	1	12
INC Rn	08-0F	1	3	4
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

Instruction	OPCODE	Bytes	Clock Cycles	MS51 V.S. Tradition 80C51 Speed Ratio
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
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

Instruction	OPCODE	Bytes	Clock Cycles	MS51 V.S. Tradition 80C51 Speed Ratio
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
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, #data, 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 MS51 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 11.3-1 Instruction Set

12 REVISION HISTORY

Date	Revision	Description
2019.11.28	1.00	Initial release
2020.02.21	1.01	Section 3.2 Modified selection guide table MS51FC0AE/MS51XC0BE ISO 7816-3 number to 2 Section 4.2 Added P3.0 PWM2_CH1 pin define in Pin Description table. Section 6.1.7 Moved Read-Modify-Write Instructions to section 6.1.7.3. Section 6.6.4 Modified the PWM0~PWM3 GPIO control register description. Section 6.9.3 Modified table T1M description to CKCON[4].

MS51

MS51FC0AE/ MS51XC0BE/ MS51EC0AE/ MS51TC0AE/ MS51PC0AE Technical Reference Manual (TRM)

MS51FC0AE/ MS51XC0BE/ MS51EC0AE/ MS51TC0AE/ MS51PC0AE 技术参考手册

NuMaker-ML51PC Gerber & Schematic PCB

NuTiny-ML51EB Gerber & Schematic PCB File

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.

*Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.*